IBM Hyper Protect Virtual Servers User’s Guide - Version 1.2.x

Tables of Contents

About this documentation

This documentation describes how to use the IBM® Hyper Protect Virtual Servers to deploy and manage Docker-based workloads on IBM Z and LinuxONE servers in your environment.

The documentation is structured based on the following major workflow:-

- How to configure and start a Secure Service Container partition.
- How to install Hyper Protect Virtual Servers hosting appliance by using the Secure Service Container user interface.
- How to configure and start IBM Hyper Protect Virtual Servers on IBM Z and LinuxONE servers in your cloud environment.
- How to securely build and deploy your containerized workload into the Hyper Protect Virtual Server containers.

This documentation describes the version of IBM Hyper Protect Virtual Servers that is available for deployment with:

- Hardware Management Console (HMC) / Support Element (SE) Version 2.14.0 (z14, z14 ZR1, LinuxONE Emperor II or LinuxONE Rockhopper II). For more information about Secure Service Container, see Secure Service Container User's Guide, SC28-6978-02a.
- Hardware Management Console (HMC) / Support Element (SE) Version 2.15.0 (z15, LinuxONE III). For more information about Secure Service Container, see Secure Service Container User's Guide, SC28-7005-01.

Figures that are included in this document illustrate concepts and are not necessarily accurate in content, appearance, or specific behavior.

Important: The PDF version of this product document is a snapshot of the content on IBM Documentation on 2021-09-23. To read the up-to-date documentation about IBM Hyper Protect Virtual Servers, see IBM Documentation.

Intended audience

The primary audience for this documentation is developers wanting to securely build their applications, and administrators who are responsible for installing, and managing containerized applications in the secured cloud environment. Those containerized applications can be hosted within Hyper Protect Virtual Server containers on an IBM Z or LinuxONE server.

This documentation distinguishes the following types of user roles:

- Cloud admin
- Appliance admin
- System admin
- ISV or App developer

The different tasks that are described in this documentation are associated to one of these user roles. A single user can have a single role or multiple roles. For more information about the roles, see User roles.

Prerequisite and related information

To deploy and manage containerized workloads within Hyper Protect Virtual Server containers on IBM Z or LinuxONE servers, in addition to this documentation, system administrators also need to access the following reference to accomplish specific tasks.

- For more information about IBM Secure Service Container deployment to z14, z14 ZR1, LinuxONE Emperor II or LinuxONE Rockhopper II, see Secure Service Container User's Guide, SC28-6978-02a.
- For more information about IBM Secure Service Container deployment to z15 or LinuxONE III, see Secure Service Container User's Guide, SC28-7005-00b.

Release notes

- What's new
- Known issues and limitations
- Accessibility features

What's new in version 1.2.4

Get a quick overview of what's added, changed, improved, or deprecated in this release.

IBM® Hyper Protect Virtual Servers Version 1.2.4 introduces the following new features and enhancements:

High availability and disaster recovery

You can setup backup and recovery procedures for IBM Hyper Protect Virtual Servers. For more information, see the following topics.

- High availability and disaster recovery
  - Backing up and recovering SSH images
  - Backing up and recovering non-SSH images
  - Backing up and recovering non-SSH images by using BYOI

Repository registration files changes

Repository registration files that were generated by using Hyper Protect Virtual Servers Version 1.2.3, or earlier, can no longer be used to register the repository. You must regenerate the registration files for Hyper Protect Virtual Servers Version 1.2.4.

What's new in version 1.2.3

Get a quick overview of what's added, changed, improved, or deprecated in this release.

IBM Hyper Protect Virtual Servers Version 1.2.3 introduces the following new features and enhancements:

The imagecache parameter

You can specify the **`imagecache`** parameter in the configuration yaml file that is used to create a virtual server by using the **`hpvs deploy`** command. When the value of the **`imagecache`** parameter is set to **`true`**, then the image from the cache is used during the deploy operation, and when the value is set to **`false`**, the deploy operation pulls the images and register the repositories every time the deploy operation is run. For more information, see the following topics.

- Creating a Hyper Protect Virtual Server instance
- Building your application with the Secure Build virtual server
- Deploying your applications securely
- Working with Monitoring virtual servers
- Working with GREP11 virtual servers
- Configuration files of IBM Hyper Protect Virtual Servers

License acceptance

The license must be accepted for executing the setup script. For more information, see Setting up the environment by using the setup script.

Non-default SSH port

A non-default SSH port can be specified in the "github url" parameter. For more information, see Building your application with the Secure Build virtual server.

New CLI commands

The **`hpvs host show`**, **`hpvs host unset`**, and **`hpvs network update`** commands are now supported. For more information, see Commands in IBM Hyper Protect Virtual Servers.

Networking with Hipersockets

You can leverage technologies that are available within the Z architecture like internal communications to drive performance, scale, and optimized use of hardware resources. IBM Z Architecture internal logical partition (LPAR) to LPAR communications technology using Hipersockets is now supported, thereby reducing additional hardware requirement and increasing performance. For more information, see the following topics.

- System requirements
- Configuring the network on the Secure Service Container partition

Added Linux capabilities

Added support for the **`cap_add`** parameter. For more information, see the following topics.

- Building your application with the Secure Build virtual server
- Deploying your applications securely.

Schnorr signature support

The Schnorr signature is a digital signature produced by the Schnorr signature algorithm and is known for its simplicity, efficiency, and generates short signatures. For more information, see the following topics.

- Working with GREP11 virtual servers
- Testing the GREP11 virtual server

What's new in version 1.2.2.1

Get a quick overview of what's added, changed, improved, or deprecated in this release.

IBM Hyper Protect Virtual Servers Version 1.2.2.1 introduces the following new features and enhancements:

Fix Packs are available only on IBM Fix Central

The installation package of IBM Hyper Protect Virtual Servers version 1.2.2.1 is available only on IBM Fix Central.

For more information on how to download the Fix Pack, see Downloading the Fix Pack installation packages.

BIP32 support

Address path (BIP32) defines how to derive private and public keys of a wallet from a binary master seed (m) and an ordered set of indices. This feature is now supported. For more information, see the following topics.

- Working with GREP11 virtual servers
- Testing the GREP11 virtual server

SLIP-0010 support

SLIP-0010 describes how to derive private and public key pairs for curve types different from secp256k1. Secp256k1 refers to the parameters of the elliptic curve used in Bitcoin's public-key cryptography, and is defined in the Standards for Efficient Cryptography (SEC). This feature is now supported. For more information, see the following topics.

- Working with GREP11 virtual servers
- Testing the GREP11 virtual server

Upgrade IBM Hyper Protect Virtual Servers

You can upgrade IBM Hyper Protect Virtual Servers from version 1.2.2 to version 1.2.2.1 For more information, see Upgrading IBM Hyper Protect Virtual Servers.

What's new in version 1.2.2

Get a quick overview of what's added, changed, improved, or deprecated in this release.

IBM Hyper Protect Virtual Servers Version 1.2.2 introduces the following new features and enhancements:

JSON format for the output of the Command Line Interface (CLI)

You can use the --json flag when you want the output to be displayed in json format. For more information, see Commands in IBM Hyper Protect Virtual Servers.

The hpvs undeploy command

You can use the `hpvs undeploy` command to delete existing virtual server instances along with resources like networks, and quotagroups, that were allocated to that virtual server. For more information, see Undeploying virtual servers.

Git Large File Storage (LFS) support

You can use Git LFS with the Secure Build virtual server to build your source code stored in the GitHub repository, deploy it into the IBM Hyper Protect Virtual Servers as a Hyper Protect Virtual Server instance, and publish the built image to the remote Docker repository. For more information, see Building your application with the Secure Build Virtual Server.

Ed25519 support

Ed25519 is a public-key signature system with several attractive features and is now supported. Only the **CEX7P** card is supported with ED25519. For more information, see the following topics.

- Working with GREP11 virtual servers
- Testing the GREP11 virtual server

Updated hpvs deploy command

You can update the resources or configuration of a virtual server after the completion of the deploy operation by using the **-u**, or the **--update** flag of the **hpvs deploy** command. For more information, see the following topics.

- Creating a Hyper Protect Virtual Server instance
- Building your application with the Secure Build virtual server
- Deploying your applications securely
- Working with Monitoring virtual servers
- Working with GREP11 virtual servers

Updated the topic on GREP11 virtual servers

The example of the configuration yaml file has been updated with a new variable and the example of the json file is updated with the changes required for the new GREP11 image. For more information, see Working with GREP11 virtual servers.

Base64 format for the SSH key

The following topics have been updated with changes for the base64 format for the SSH key.

- Creating a Hyper Protect Virtual Server instance
- Building your application with the Secure Build virtual server
- Virtual server configuration file

Setup script changes

The setup.sh script can be executed by a root or a non-root user. For more information, see Setting up the environment by using the setup script.

RUNQ_ROOTDISK

A dedicated root-disk can be assigned to a virtual server in the environment variables by using the mount_id of disks (mounts) that are assigned to a virtual server from the available quotagroup. You can reset this root-disk by using the **--update** flag of the **hpvs deploy** command and setting the value of the **reset_root** parameter to *true* in **mount** section of the configuration file. RUNQ_ROOTDISK works for both passthrough and non passthrough quotagroups. The parameter **reset_root:true** works only for non passthrough quotagroups.

The following topics have been updated for this feature.

- Creating a Hyper Protect Virtual Server instance
- Building your application with the Secure Build virtual server
- Virtual server configuration file

Upgrade IBM Hyper Protect Virtual Servers

You can upgrade IBM Hyper Protect Virtual Servers from version 1.2.1.1, or 1.2.1 to version 1.2.2. For more information, see Upgrading IBM Hyper Protect Virtual Servers from version 1.2.1.1, or 1.2.1 to version 1.2.2.

Enabling ports

When you are using IBM Hyper Protect Virtual Servers version 1.2.2, or later, before you build a docker image by using the Hyper Protect base images, you must open the required ports for your application. For more information, see Enabling ports.

What's new in version 1.2.1.1

Get a quick overview of what's added in this Fix Pack 1 release.

Fix Packs are available only on IBM Fix Central

The installation package of IBM Hyper Protect Virtual Servers version 1.2.1.1 is available only on IBM Fix Central.

For more information on how to download the Fix Pack, see Downloading the Fix Pack installation packages.

Changes in the images for Hyper Protect Virtual Servers

- IBM Hyper Protect Virtual Servers Version 1.2.1.1 contains updated signing keys which enable the deployment of new images on the IBM Hyper Protect Virtual Servers platform.
- The signing keys that were shipped with the IBM Hyper Protect Virtual Servers Version 1.2.1 (July 2020) have been refreshed in IBM Hyper Protect Virtual Servers Version 1.2.1.1. It is highly recommended that you migrate to IBM Hyper Protect Virtual Servers Version 1.2.1.1 to continue using the version with refreshed keys for the provided images.
- IBM Hyper Protect Virtual Servers Version 1.2.1.1 does not introduce any new features nor does it change the functionality of existing features supported by IBM Hyper Protect Servers Version 1.2.1.

Fix Pack installation instructions

To install the fix pack for IBM Hyper Protect Virtual Servers Version 1.2.1.1, delete the */usr/local/bin/hpvs* directory and then follow the instructions from step 4 of the topic Downloading the installation package.

What's new in version 1.2.1

Get a quick overview of what's added, changed, improved, or deprecated in this release.

IBM Hyper Protect Virtual Servers Version 1.2.1 introduces the following new features and enhancements:

The IBM Hyper Protect Virtual Servers Command Line Interface (CLI)

The IBM Hyper Protect Virtual Servers environment can be setup by using a new set of CLI commands that simplifies the process of running various commands to create the virtual servers, deploy your workloads, monitoring, and GREP11 library. This includes the `hpvs deploy` command that simplifies the creation of the IBM Hyper Protect Virtual Servers and deployment. For more information, see Commands in IBM Hyper Protect Virtual Servers.

Setup script to set up the environment

The `setup.sh` script performs an initial environment check on the Linux Management server. The script also performs the following actions:

- Checks if Docker, OpenSSL and GPG are installed. This is required to use IBM Hyper Protect Virtual Servers.
- Sets up the initial infrastructure required for the new IBM Hyper Protect Virtual Servers CLI. For more information, see Setting up the environment by using the setup script.

The mustgather script

The IBM Hyper Protect Virtual Servers Version 1.2.1 provides an automated procedure to gather useful information when you want to open a support ticket. For more information, see Gathering Information for IBM Support.

Known issues and limitations

This topic lists some of the known issues and limitations of IBM Hyper Protect Virtual Servers.

Known issues and limitations with IBM Hyper Protect Virtual Servers Version 1.2.4.

- When you are running applications on virtual servers that are using non-passthrough quotagroups, it is recommended that you monitor the available datapool size by using the `hpvs quotagroup show` command, and update the size by 5 GB when the size of the datapool is less than 1 GB. You can use the `hpvs quotagroup update` command to increase the size of the datapool.
- The snapshots of a Hyper Protect Virtual Server container can be created only on the same Secure Service Container partition that the server instance resides.
- You can use IBM Hyper Protect Virtual Servers only with Docker Hub or IBM Container Registry.
- You must restart the Hyper Protect Virtual Server container after you revert a snapshot of the Hyper Protect Virtual Server container.
- Secure Build requires that the private key, used to secure access to the source Github repository, does not have a passphrase.
- IBM Cloud Object Storage service is supported only for archiving application manifest files.
- Backup and restore of encrypted credentials used by the Secure Build container can only be supported by using Hosting Appliance snapshots.
- The monitoring infrastructure collects metrics only from the Hyper Protect Virtual Servers hosting appliance and Secure Service Container partition.
- When using ep11 for text file encryption, the text file size architectural limit is 4MB.
- You must not delete the contents of the installation directory after you have run the `setup.sh` script. The `setup.sh` creates a working directory that contains images which are symbolic links pointing to the images in

the extracted directory. If you delete the contents of the installation directory, you cannot run the **`hpvs`** commands and must repeat the process of downloading and extracting the images.
- You cannot create a snapshot of virtual servers that are configured with passthrough quotagroups.
- You cannot retrieve snapshots from virtual servers that have been deleted.
- You cannot create snapshots of a virtual server that has multiple quotagroups attached when any of them is a passthrough quotagroup.
- If you create a snapshot for a virtual server with passthrough and non-passthrough quotagroups, that results in an error, then you cannot delete the snapshot and the virtual server.
- When specifying the size for the quotagroup, you must not use decimal notation. For example, use 1800 MB instead of 1.8GB.
- If you had used quotagroup parameters when creating a virtual server, then you must provide those values as parameters when you want to update the virtual server.
- If you update a virtual server without specifying the volume details, those volumes are detached from the virtual server that was used during virtual server creation and the virtual server will be in the restarting state. You can delete the virtual server but you cannot delete the quotagroup, from which the volumes were assigned to the virtual server.
- The snapshots of a Hyper Protect virtual server container can be created only on the same Secure Service Container partition that the server instance resides in.
- The Secure Service Container for IBM Cloud Private feature and IBM Hyper Protect Virtual Servers feature cannot co-exist on the same Secure Service Container partitions or Linux master/management servers. Each feature must use separate, dedicated Secure Service Container partitions and Linux master/management servers.

Known issues and limitations with IBM Hyper Protect Virtual Servers version 1.2.3 and 1.2.2.

- When you are running applications on virtual servers that are using non-passthrough quotagroups, it is recommended that you monitor the available datapool size by using the **`hpvs quotagroup show`** command, and update the size by 5 GB when the size of the datapool is less than 1 GB. You can use the **`hpvs quotagroup update`** command to increase the size of the datapool.
- The snapshots of a Hyper Protect Virtual Server container can be created only on the same Secure Service Container partition that the server instance resides.
- You can use IBM Hyper Protect Virtual Servers only with Docker Hub or IBM Container Registry.
- You must restart the Hyper Protect Virtual Server container after you revert a snapshot of the Hyper Protect Virtual Server container.
- Secure Build requires that the private key, used to secure access to the source Github repository, does not have a passphrase.
- IBM Cloud Object Storage service is supported only for archiving application manifest files.
- Backup and restore of encrypted credentials used by the Secure Build container can only be supported by using Hosting Appliance snapshots.
- The monitoring infrastructure collects metrics only from the Hyper Protect Virtual Servers hosting appliance and Secure Service Container partition.
- When using ep11 for text file encryption, the text file size architectural limit is 4MB.
- You must not delete the contents of the installation directory after you have run the **`setup.sh`** script. The **`setup.sh`** creates a working directory that contains images which are symbolic links pointing to the images in the extracted directory. If you delete the contents of the installation directory, you cannot run the **`hpvs`** commands and must repeat the process of downloading and extracting the images.
- You cannot create a snapshot of virtual servers that are configured with passthrough quotagroups.
- You cannot retrieve snapshots from virtual servers that have been deleted.
- You cannot create snapshots of a virtual server that has multiple quotagroups attached when any of them is a passthrough quotagroup.
- If you create a snapshot for a virtual server with passthrough and non-passthrough quotagroups, that results in an error, then you cannot delete the snapshot and the virtual server.

- When specifying the size for the quotagroup, you must not use decimal notation. For example, use 1800 MB instead of 1.8GB.
- If you had used quotagroup parameters when creating a virtual server, then you must provide those values as parameters when you want to update the virtual server.
- If you update a virtual server without specifying the volume details, those volumes are detached from the virtual server that was used during virtual server creation and the virtual server will be in the restarting state. You can delete the virtual server but you cannot delete the quotagroup, from which the volumes were assigned to the virtual server.
- The snapshots of a Hyper Protect virtual server container can be created only on the same Secure Service Container partition that the server instance resides in.
- The Secure Service Container for IBM Cloud Private feature and IBM Hyper Protect Virtual Servers feature cannot co-exist on the same Secure Service Container partitions or Linux master/management servers. Each feature must use separate, dedicated Secure Service Container partitions and Linux master/management servers.

Known issues and limitations with IBM Hyper Protect Virtual Servers Version 1.2.1.

- The snapshots of a Hyper Protect Virtual Server container can be created only on the same Secure Service Container partition that the server instance resides.
- You can use IBM Hyper Protect Virtual Servers only with Docker Hub or IBM Container Registry.
- You must restart the Hyper Protect Virtual Server container after you revert a snapshot of the Hyper Protect Virtual Server container.
- A `/newroot` mount point is initiated by default to bootstrap the Hyper Protect Virtual Server container.
- Secure Build does not support Git Large File Storage (LFS).
- Secure Build requires that the private key, used to secure access to the source Github repository, does not have a passphrase.
- IBM Cloud Object Storage service is supported only for archiving application manifest files.
- Backup and restore of encrypted credentials used by the Secure Build container can only be supported by using Hosting Appliance snapshots.
- The monitoring infrastructure collects metrics only from the Hyper Protect Virtual Servers hosting appliance and Secure Service Container partition.
- When using ep11 for text file encryption, the text file size architectural limit is 4MB.
- You must not delete the contents of the installation directory after you have run the `setup.sh` script. The `setup.sh` creates a working directory that contains images which are symbolic links pointing to the images in the extracted directory. If you delete the contents of the installation directory, you cannot run the `hpvs` commands and must repeat the process of downloading and extracting the images.
- You cannot create a snapshot of virtual servers that are configured with passthrough quotagroups.
- You cannot retrieve snapshots from virtual servers that have been deleted.
- You cannot create snapshots of a virtual server that has multiple quotagroups attached when any of them is a passthrough quotagroup.
- If you create a snapshot for a virtual server with passthrough and non-passthrough quotagroups, that results in an error, then you cannot delete the snapshot and the virtual server.
- When specifying the size for the quotagroup, you must not use decimal notation. For example, use 1800 MB instead of 1.8GB.
- If you had used quotagroup parameters when creating a virtual server, then you must provide those values as parameters when you want to update the virtual server.
- If you delete a quotagroup of a virtual server that has been deleted, then you cannot remove the volume.
- The snapshots of a Hyper Protect virtual server container can be created only on the same Secure Service Container partition that the server instance resides in.
- The Secure Service Container for IBM Cloud Private feature and IBM Hyper Protect Virtual Servers feature cannot co-exist on the same Secure Service Container partitions or Linux master/management servers. Each feature must use separate, dedicated Secure Service Container partitions and Linux master/management servers.

Accessibility features for IBM® Hyper Protect Virtual Servers

Accessibility features assist users who have a disability, such as restricted mobility or limited vision, to use information technology content successfully.

Overview

IBM Hyper Protect Virtual Servers includes the following major accessibility features:

- Keyboard-only operations
- Screen reader operations
- Command line interface (CLI) to configure the IBM Hyper Protect Virtual Servers offering

IBM Hyper Protect Virtual Servers uses the latest W3C Standard, WAI-ARIA 1.0, to ensure compliance with Section 508 Standards for Electronic and Information Technology and Web Content Accessibility Guidelines (WCAG) 2.0. To take advantage of accessibility features, use the latest release of your screen reader and the latest web browser that is supported by IBM Hyper Protect Virtual Servers.

The IBM Hyper Protect Virtual Servers online product documentation in IBM Knowledge Center is enabled for accessibility. For general accessibility information, see Accessibility in IBM.

Keyboard navigation

IBM Hyper Protect Virtual Servers uses standard navigation keys.

IBM Hyper Protect Virtual Servers uses the following keyboard shortcuts.

Action	Shortcut for Internet Explorer	Shortcut for Firefox
Move to the Contents View frame	Alt+C, then press Enter and Shift+F6	Shift+Alt+C and Shift+F6

Vendor software

IBM Cloud Private includes certain vendor software that is not covered under the IBM license agreement. IBM makes no representation about the accessibility features of these products. Contact the vendor for accessibility information about its products.

Related accessibility information

In addition to standard IBM help desk and support websites, IBM has a TTY telephone service for use by deaf or hard of hearing customers to access sales and support services:

TTY service
800-IBM-3383 (800-426-3383)
(within North America)

For more information about the commitment that IBM has to accessibility, see IBM Accessibility.

Introduction

To understand how IBM Hyper Protect Virtual Servers works, you can start with the following topics.

Overview

Many technologies need to protect applications in production, leveraging encryption technologies; however, security threats can also surface during the development, pre-production phases. Additionally, during deployment and production, insiders who manage the infrastructure that hosts critical applications, may pose a threat given their super-user credentials and level of access to secrets or encryption keys.

Organizations need to incorporate secure design practices in their development operations and embrace DevSecOps to ensure the protection of their applications from the vulnerabilities and threat vectors that can compromise their data and potentially threaten their business.

IBM® Hyper Protect Virtual Servers protects Linux workloads on IBM Z and LinuxONE throughout their lifecycle build management and deployment. This solution delivers the security needed to protect mission critical applications in hybrid multi-cloud deployments.

IBM Hyper Protect Virtual Servers enables:

- Developers to securely build their applications in a trusted environment with integrity.
- IT infrastructure providers to manage the servers and virtualized environment where the applications are deployed without having access to those applications or their sensitive data
- Application users to validate that those securely built applications originate from a trusted source by integrating this validation into their own auditing processes.
- Chief Information Security Officers (CISOs) to be confident that their data is both protected and private from internal and external threats.

IBM Hyper Protect Virtual Servers solutions delivers security measures to address threat vectors that appear at different phases of an application’s lifecycle: build, deployment, and management. It is designed to uniquely protect such workloads that are deployed on IBM Z and LinuxONE servers in hybrid, multi-cloud environments.

IBM Security Threat Management solutions help you thrive in the face of cyber uncertainty. As part of IBM's best practices towards securing keys and key management, IBM Hyper Protect Virtual Servers version 1.2.3, 1.2.2.1 and 1.2.2 images are signed with keys that are embedded into the product, enabling image validation during the deployment process. The signing keys for IBM Hyper Protect Virtual Servers} version 1.2.4 are valid until 31 December 2022, while the signing keys for IBM Hyper Protect Virtual Servers version 1.2.3 are valid until 02 December 2021.

These signing keys are periodically refreshed and the updated keys will be made available with adequate notice before the expiration of the existing signing keys. It is recommend that you upgrade the IBM Hyper Protect Virtual Servers product regularly and stay protected against internal and external threats.

Advantages

Running Hyper Protect Virtual Server containers on the Secure Service Container partitions provides you the following advantages in terms of security and integrity.

- System administrators do not need the access to the application data, memory, logs, secrets, applications or the operating system in the Hyper Protect Virtual Server containers.
- Application developers do not need the secret to the production environment, and managing the Hyper Protect Virtual Server containers does not require access to the application secrets.
- The containerized application images are signed with GPG keys when publishing, and verified again when being deployed. The signing keys are generated within the Secure Build process and your private keys are never revealed. Only the images generated by using the Secure Build procedure can be uploaded to your docker repository and installed to the Secure Service Container partitions.
- The Secure Build generates a signed manifest indicating the origin of the image for future audit. The manifest contains a copy of the Github project that was cloned by the Secure Build server container, and a copy of the build log (build.log) and overall build status result (build.json). The manifest tar ball is signed with the manifest private key. The user can download the manifest public key and use it to verify a manifest. You can optionally store the manifest in the IBM Cloud Object Storage (COS) by using the Secure Build.
- You can integrate IBM Hyper Protect Virtual Servers into your own Continuous Integration and Continuous Delivery (CICD) pipeline to fully adopt the security advantages provided by the offering.

Technology at a glance

IBM Hyper Protect Virtual Servers is a software solution built on the IBM Secure Service Container framework, which enables users to run containerized Linux workloads in the secure virtual server containers on IBM Z and LinuxONE.

IBM Hyper Protect Virtual Servers provides an encrypted environment (data at rest, data in flight), with peer to peer and peer to host isolation protecting container applications from access via privileged credentials, whether access is accidental or malicious, internal or external to an organization.

IBM Hyper Protect Virtual Servers ensures your applications can be deployed and managed from trusted sources without the infrastructure team being able to access the data, secrets or application.

IBM Secure Service Container

IBM Secure Service Container is a software appliance infrastructure that combines an operating system, middleware, and application components into a single software image. Software images deployed to a Secure Service Container partition can exploit the underlying security capabilities of the IBM Z and LinuxONE infrastructure.

By focusing on ease of management, ease of deployment, and security, the Secure Service Container is delivered in a virtual software appliance form factor, which can also isolate the running workload and deliver protections around the access of the environment.

In the Secure Service Container, a specialized Docker runtime environment called **`runq`** is used to spawn a dedicated qemu virtual server (VS) instance for each Docker image, including a guest operating system (OS) kernel for each qemu virtual server, and during deployment, a runtime environment of the workloads.

All these components are packaged together as the hosting appliance, and can be deployed on a partition of an IBM Z and LinuxONE server in a single step.

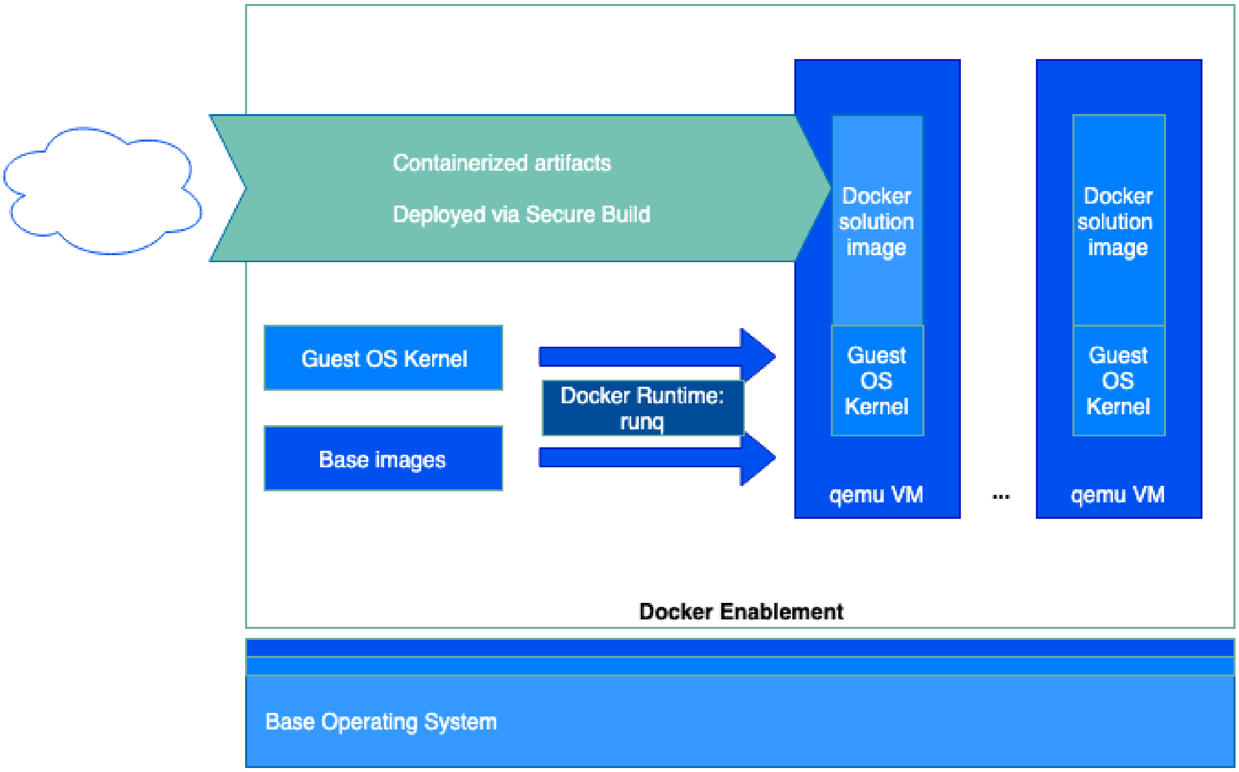

Figure 1. IBM Secure Service Container framework - Docker Enablement

Architecture overview

When using IBM Hyper Protect Virtual Servers, you need to prepare a management server (x86 or Linux on IBM Z or LinuxONE, for example, s390x) to run the commands and manage the components of the offering.

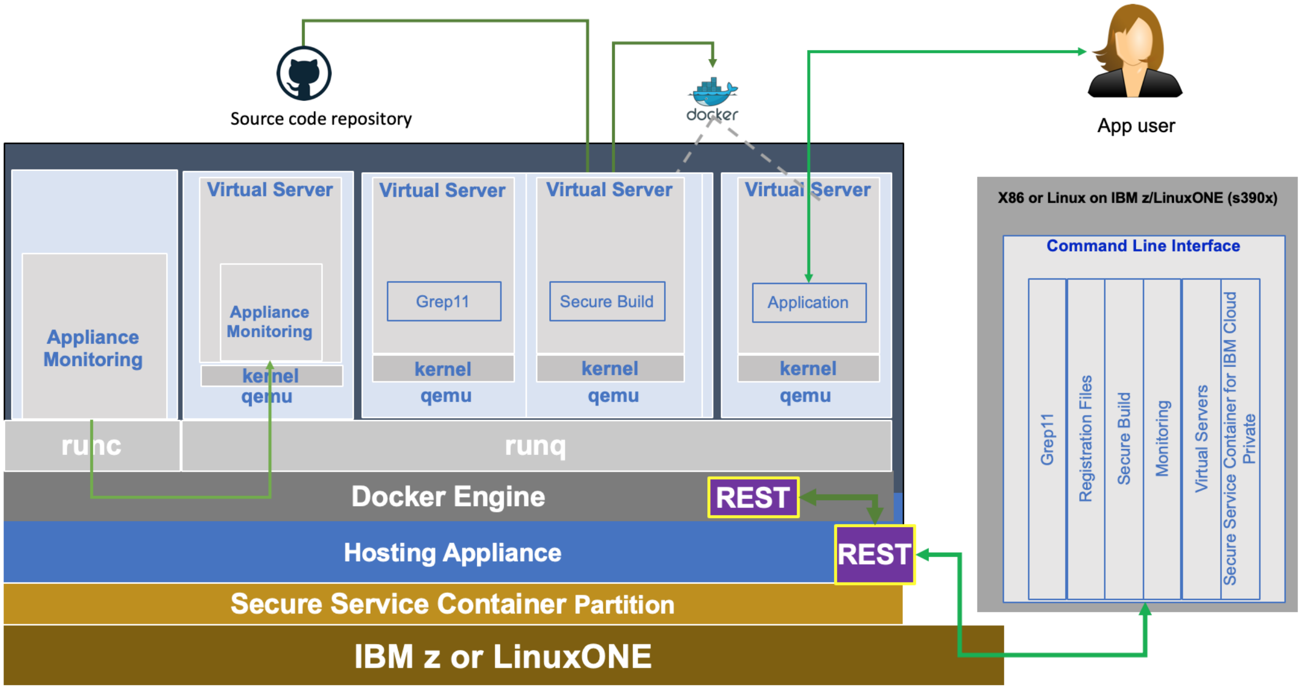

Figure 2. IBM Hyper Protect Virtual Servers - Architecture

The IBM Hyper Protect Virtual Servers offering provides a list of commands with the following capabilities across the application lifecycle phases:

- Build
  - Build user-provided source code (located in a git repository) into Linux on IBM Z / LinuxONE (i.e. s390x) compatible workloads
  - Create Hyper Protect Virtual Server containers on the Secure Service Container partition based on images in the git repository
- Register
  - Download a repository definition file template from the hosting appliance
  - Encrypt a repository definition file with security keys
- Deploy
  - Deploy workloads into Hyper Protect Virtual Server containers on the Secure Service Container partitions
- Manage
  - Manage Hyper Protect Virtual Server container images
- Monitor
  - Monitor IBM Hyper Protect appliance health such as the usage of CPU, memory, disk, and uptime.
- Crypto
  - Provide Enterprise PKCS #11 (EP11) interfaces for crypto operations such as key generation, encryption, decryption, data wrapping and unwrapping in EP11 over gRPC (grep11) client applications.

IBM Hyper Protect Virtual Servers also leverages Docker Content Trust (DCT), which uses digital signatures for data sent to and received from remote Docker registries on the Secure Service Container partitions. For more information about the DCT, see Content trust in Docker.

By using IBM Hyper Protect Virtual Servers, your repository and containerized images are protected with different keys on different stages.

Key Name	Originator / Owner	Location	Function	Lifecycle Phase
IBM Key Pair	IBM	• Public key or certificate: CLI tool • Private key: Hosting appliance	• Public key or certificate: Encrypt repository definition files • Private key: Decrypt repository definition files	• Public key or certificate: Application registration • Private key: Application deployment
Repository signing key pair	IBM	• Public key or certificate: Remote Docker repository • Private key: Hosting appliance	• Public key or certificate: Verify images built by Secure Build • Private key: Sign images built by Secure Build	Application build (First time)

Key Name	Originator / Owner	Location	Function	Lifecycle Phase
Image signing key pair	ISV or app developer	• Public key or certificate: Sent to cloud admin(dev) • Private key: Hosting appliance	• Public key or certificate: Cloud admin verifies signature of the repository definition file and images built by the Secure Build • Private key: Sign the repository definition file and images built by Secure Build	• Public key or certificate: Application registration • Private key: Application Registration
Secure Build initialization key pair	• ISV or app developer • Cloud admin	• Public key or certificate: Sent to cloud admin(dev) • Private key: Local file system	• Public key or certificate: Creates the Secure Build container on the Secure Service Container partition • Private key: initialize the Secure Build container so that the Secure Build container only accepts the API calls encrypted with this private key.	• Public key or certificate: Secure Build initialization • Private key: Secure Build invocation
Secure Build manifest key pair	Secure Build container	• Public key or certificate: Sent to audit(dev) • Private key: Hosting appliance	• Public key or certificate: can be retrieved from the Secure Build container to audit the manifest • Private key: Sign the manifest during the Secure Build	• Public key or certificate: Manifest audit • Private key: Manifest signature
Monitoring infrastructure (server-side) key pair	Cloud admin	• Public key or certificate: Local file system • Private key: Local file system	• Public key or certificate: Enable TLS encryption for monitoring infrastructure • Private key: Enable TLS encryption for monitoring infrastructure	• Public key or certificate: Collecting monitoring metrics • Private key: N/A

Key Name	Originator / Owner	Location	Function	Lifecycle Phase
Monitoring client key pair	Cloud admin	• Public key or certificate: Local file system • Private key: Local file system	• Public key or certificate: Enable mutual TLS communication • Private key: Enable TLS encryption for the client tool	• Public key or certificate: Collecting monitoring metrics only if client authentication is enabled • Private key: N/A
GREP11 container key pair	Cloud admin	• Public key or certificate: Hosting appliance • Private key: Local file system	• Public key or certificate: Authenticate secure communication between GREP11 container and client apps • Private key: Encrypt the requests from GREP11 client apps	• Public key or certificate: Invoking GREP11 calls • Private key: N/A

Components

IBM Hyper Protect Virtual Servers consists of the following components:

- A hosting appliance that is based on the IBM Secure Service Container framework, which can host containerized workloads with focus on superior data security in the cloud and on-premise.
- Base images of Hyper Protect Virtual Server container (**`hpvsop-base`** and **`hpvsop-base-ssh`**), which can be used to host your application code. The **`hpvsop-base-ssh`** base image provides additional SSH daemon for debugging and testing.
- A base image of the Secure Build container (**`secure-docker-build`**), which can be provisioned on the Secure Service Container partition and bound to build your application code exclusively.
- Base images of the monitoring infrastructure (**`collectd-host`** and **`monitoring-host`**), which can be used to collect metrics from Secure Service Container framework.
- A base image of the Enterprise PKCS #11 (EP11) over gRPC (Grep11) container (**`grep11-container`**), which can communicate with Hardware Security Module (HSM) and generates asymmetric (public and private) key pairs.
- A set of command line tools that are used to:
  - Create and manage the Hyper Protect Virtual Server instances
  - Securely build and publish your applications as containerized workloads
  - Deploy your containerized workloads to the Secure Service Container framework
  - Monitor IBM Hyper Protect appliance health such as the usage of CPU, memory, disk, and uptime.
  - Provide Enterprise PKCS #11 (EP11) interfaces for crypto operations such as key generation, encryption, decryption, data wrapping and unwrapping in EP11 over gRPC (grep11) client applications.

Table 1. IBM Hyper Protect Virtual Servers components

Component	In 1.2.4	In 1.2.3
Hosting appliance	4.3.5	3.17.0

Component	In 1.2.4	In 1.2.3
`hpvsop-base` and `hpvsop-base-ssh`	1.2.4	1.2.3
`secure-docker-build`	1.2.4	1.2.3
`collectd-host` and `monitoring-host`	1.2.3, and 1.2.4	1.2.3, and 0.9.1
`grep11-container`	1.2.4	2.3.0

The table shows the command line tool modules:

CLI modules	Available in 1.2.1, or later	Available in 1.2.0.1 or 1.2.0
Crypto commands	Yes	Yes
Deploy commands	Yes	Yes
Host commands	Yes	Yes
Images commands	Yes	Yes
Quotagroup commands	Yes	Yes
Regfile commands	Yes	Yes
Registry commands	Yes	Yes
Repository commands	Yes	Yes
Secure Build commands	Yes	Yes
Snapshot commands	Yes	Yes
Secure Build commands	Yes	Yes
Virtual server commands	Yes	Yes
Monitoring commands	No	Yes
GREP11 commands	No	Yes

See Downloading the installation package for the information about how to get these components.

System requirements

Software, hardware, and system configuration settings that are required for setting up a Hyper Protect Virtual Server offering.

Hardware requirements for the Linux management server

The x86 or Linux on IBM Z/LinuxONE (i.e., s390x architecture) management server is used to download the Hyper Protect Virtual Server installation binary, and install IBM Hyper Protect Virtual Servers CLI tool.

Table 1. 64-bit x86 or Linux on IBM Z/LinuxONE (i.e., s390x architecture) management server requirements

Minimal requirement
2 or more x86 Linux cores with at least 2.4 GHz, or 1 Integrated Facility (IFL) on mainframe
8 GB RAM
150 GB disk space

Hardware requirements for Secure Service Container partition

You can configure Secure Service Container partitions on the following IBM Z and LinuxONE systems.

- IBM z15 (z15) (machine type 8561 or 8562)
- IBM z14 (z14) (machine type 3906 or 3907)
- IBM LinuxONE III (LinuxONE III)
- IBM LinuxONE Emperor II (Emperor II), or IBM LinuxONE Rockhopper II (Rockhopper II)

The suggested practice is to use the latest available firmware for Secure Service Container, which is identified by the engineering changes (ECs) in the following table. To find the latest available EC microcode control levels (MCLs) for Secure Service Container, use the instructions for hardware updates in "Prerequisites for using Secure Service Container" after you download Secure Service Container User's Guide from the About topic.

Table 2. Engineering changes by machine type

Machine Type	Version / Driver	Bundle	Engineering Changes
8561 or 8562	Version 2.15.0 Driver 41	S49a or later	• SE-BCBASE P46639 • SE-BCBOOT P46640 • SE-BCINST P46655
3906	Version 2.14.1 Driver 36	S64b or later	• SE-BCBASE P41454 • SE-BCBOOT P41454 • SE-BCINST P41467
3907	Version 2.14.1 Driver 36	S53 or later	• SE-BCBASE P41453 • SE-BCBOOT P41454 • SE-BCINST P41467

The following table shows the minimal requirement for one Secure Service Container partition.

Table 3. Secure Service Container partition requirements

Minimal (one Hyper Protect Virtual Server container + one Secure Build container)
2 IFLs
12 GB RAM
190 GB storage (50 GB for the hosting appliance, 100 GB in the storage pool for one Hyper Protect Virtual Server container, and 40 GB for one Secure Build container)

Note:

- The actual resources required on the Secure Service Container partition depends on the resource consumption of your workload to be deployed into the Hyper Protect Virtual Server container.
- If you plan to have multiple Hyper Protect Virtual Server containers or Secure Build containers communicating with each other on the Secure Service Container partition, and assign IP addresses to each of them, you need to use at least 1 Open System Adapter (OSA) card to create multiple virtual devices for data traffic. If you plan to have internal network communication established between Hyper Protect Virtual Servers on two Secure Service Container partitions, you can have Hipersockets configured in layer 2=1 mode. Hipersockets are supported in IBM Hyper Protect Virtual Servers version 1.2.3, or later.
- If you want to use Enterprise PKCS #11 over gRPC (GREP11) containers in IBM Hyper Protect Virtual Servers, you must prepare a Trusted Key Entry (TKE) workstation and Crypto Express cards, such as IBM Crypto Express6s (CEX6S) and IBM Crypto Express7s (CEX7S). The Crypto Express will differ by machine generation (CEX6S for the z14 generation, CEX7S for the z15 generation).

Software requirements

- IBM PCIe Cryptographic Coprocessor Version 3 (PCIeCC3) software, which includes IBM Common Cryptographic Architecture (CCA) and Enterprise PKCS #11 (EP11), and be ordered from Cryptocards software-package selection page.

Supported operating systems and platforms

The operating system for running the containers on the Secure Service Container partitions is Ubuntu 18.04, which is provided by the hosting appliance.

However, you must configure the x86 or Linux on IBM Z/LinuxONE (i.e., s390x architecture) management server with the supported operating system in the following table.

Table 4. Supported operating system and platform

Platform	Operating system
Linux 64-bit	Ubuntu 16.04 LTS and 18.04 LTS

Note:

- Redhat Linux distribution is a compatible operating system for the management server, but it has not been tested with IBM Hyper Protect Virtual Servers. Use the operating system at your own risk.
- Linux Unified Key Setup (LUKS) hardware encryption on the x86 or Linux on IBM Z/LinuxONE (i.e., s390x architecture) management server can protect the hardware from faulty access. When installing Ubuntu onto the x86 or Linux on Z server, select the **Encrypt the new Ubuntu installation for Security** option to encrypt the hard disk.

Networking

IBM Hyper Protect Virtual Servers requires two levels of network to work properly.

- Network among Hyper Protect Virtual Server containers by using the internal IP addresses
- Network for external requests to the services inside the workload deployed in the Hyper Protect Virtual Server container

Table 5. Supported network interfaces on the Secure Service Container partitions

Interface	Layer 2 network	Layer 3 network
Ethernet	Yes	Yes
VLAN	Yes	Yes

On the x86 or Linux on IBM Z/LinuxONE (i.e., s390x architecture) management server, network connection must be available to the Secure Service Container partition by using its IP address or host name.

Note:

- The default network driver is **`bridge`** and sufficient for communication among Hyper Protect Virtual Server containers.
- If you plan to access Hyper Protect Virtual Server containers from your underlying network or the containers being accessed by external workload, use the network driver **`macvlan`** and assign IP addresses to those containers, or configure the port mapping for the container on the Secure Service Container partition. When you are using port mapping, you must use the Secure Service Container management IP and mapped host port to access the virtual server application.
- If you plan to access Hyper Protect Virtual Server containers on another Secure Service Container partition, use the network driver **`macvlan`** and assign IP address to the containers on both partitions.
- If you plan to have multiple Hyper Protect Virtual Server containers or Secure Build containers communicating with each other on the Secure Service Container partition, and assign IP addresses to each of them, you need to use at least 1 Open System Adapter (OSA) card to create multiple virtual devices for data traffic. If you plan to have internal network communication established between Hyper Protect Virtual Servers on two Secure Service Container partitions, you can have Hipersockets configured in layer 2=1 mode. Hipersockets are supported in IBM Hyper Protect Virtual Servers version 1.2.3, or later.
- For more information about networking requirements in IBM Hyper Protect Virtual Servers, see Network requirements for Hyper Protect Virtual Server.
- For more information, see Networking overview for Docker containers.

Supported Docker versions

You must install the supported Docker version on the x86 or Linux on IBM Z/LinuxONE (i.e., s390x architecture) management server.

- For the x86 management server, the minimum Docker version required by IBM Hyper Protect Virtual Servers is V19.03.2 or above.
- For the Linux on IBM Z/LinuxONE (i.e., s390x architecture), the minimum Docker version required by IBM Hyper Protect Virtual Servers is v18.06.3-ce or above.
- For Docker installation, see Get Docker Engine - Community for Ubuntu or Get Docker Engine - Enterprise for Ubuntu.

Note: You can only use IBM Hyper Protect Virtual Servers with Docker Hub or IBM Container Registry.

Required ports

If you use port mapping for Secure Build container, Monitoring infrastructure, and GREP11 container, ensure that the following ports or configured mapping ports are available on the Secure Service Container partition. Otherwise, You need to request IP address for each container on the Secure Service Container partition.

Table 6. Required ports on the Secure Service Container partition

Port No.	Required by Module
`443`	Hosting Appliance REST API
`443`	Secure Build Server or bring your own image, with macvlan
Any non-reserved port	Secure Build Server
`8443`	Monitoring infrastructure
`9876`	GREP11 container

Note: You can map multiple ports on the Secure Service Container partaition with ports on your Hyper Protect Virtual Servers virtual server container for your workload. For example, the configuration such as `{"80":"8080","22":"220"}` for a Hyper Protect Virtual Servers container means port 80 on the container is mapped to port 8080 on the partition, and port 22 on the container to port 220 on the partition. For more information, see the Virtual server configuration file.

User roles

IBM Hyper Protect Virtual Servers distinguishes between different types of administrators and users to perform tasks:

- Application developer or ISV

  The application developer or independent software vendor (ISV) is responsible for developing and publishing containerized applications or solutions to the private cloud environment, and to ensure the security, integrity, and audit requirements of the software.

- Private cloud operations administrator

  The private cloud operations administrator (cloud admin) is responsible for infrastructure, security, and management of the on-premises, shared, and multi-tenant private cloud environment for containerized applications.

- Appliance administrator

  The appliance administrator (appliance admin) is responsible for deploying and managing the hosting appliances, that runs in a logical partition (LPAR) on an IBM Z or IBM LinuxONE server.

- IBM Z or LinuxONE system administrator

  The IBM Z or LinuxONE system admin (Z system admin) is responsible for creating and managing Secure Service Container partitions by performing tasks on the HMC of the IBM Z or IBM LinuxONE server.

Those roles might also interact with:

- IBM Z or LinuxONE storage admin
- Network admin
- Solution admin
- Containerized application user

FAQs

What is IBM Hyper Protect Virtual Servers?

IBM Hyper Protect Virtual Servers provides a secure virtualized infrastructure for private cloud deployments - protecting the entire lifecycle of critical Linux workloads during their build, deployment and management.

As an application developer or ISV, how can I benefit from IBM Hyper Protect Virtual Servers?

Application developers and ISVs can securely build applications with integrity.

As a cloud administrator or system administrator, how can I benefit from IBM Hyper Protect Virtual Servers?

Cloud administrators or system administrators can help manage their layer of the IT infrastructure without having access to the higher level applications and sensitive data.

As a solution end-user, how can I benefit from IBM Hyper Protect Virtual Servers?

Solution end-users can ensure the provenance of the applications being deployed by validating that applications originate from a trusted source.

What is Secure Service Container Framework?

Secure Service Container framework provides the base infrastructure for an integration of operating system, middleware, and software components into an appliance with extra security. In addition to extra security based on the `runq` container environment, the host operating system itself is also extremely secure. The Secure Service Container framework works autonomously and provides core services and infrastructure focusing on consumability and security.

What is a hosting appliance?

A hosting appliance is a software appliance built with the Secure Service Container Framework, and adds the capability to securely run containerized workloads.

What is a software appliance?

A software appliance is an integrated software containing an operating system, libraries, and so on to fulfill a single purpose, which can be installed as an appliance image on IBM Z or LinuxONE servers.

Can I deploy an application as is or is containerizing my application required to use IBM Hyper Protect Virtual Servers?

As long as your applications are developed based on Open Container Initiative (OCI) specification, you can use them in IBM Hyper Protect Virtual Servers.

Can I use my own private key to sign the images for the Docker Content Trust?

Yes. You can either use the **`docker trust key generate`** command to generate the signing key pair, or use the **`docker trust key load`** command to load an existing key for signing. However, passing in an existing key pair would invalidate the Secure Build concept as the private trust key exists(existed) outside of the Secure Build, and therefore someone else could use that key to push a bad image to the same docker repo.

What happens when I run the `docker push` command against a DCT-enabled repository?

The **`docker push`** command establishes trust at the time the first push to the docker repo is done. The command uses DOCKER_CONTENT_TRUST environment variables to determine where to establish the trust with.

Where is the Secure Build container?

The Secure Build container is created on the hosting appliance when you run the **`securebuild create`** command.

When is the docker repo key pair generated?

The docker repo key pair is generated on the first build when the **`docker push`** command is executed.

What are manifest signing keys?

The manifest signing keys are generated inside the Secure Build server container on first creation of a manifest by a Secure Build instance. It then uses **`gpg`** to sign the manifest tar file and will optionally push that signed tar to an external Cloud Object Store. The manifest and public key to validate the signature can also be retrieved from the Secure Build using the cli.

Can the Secure Build server container be used to build an existing docker image on the Docker Hub?

Yes. The newly built image must have a new name so that DCT can be established by the Secure Build server container.

Planning for the environment

You can use a PLANNING FOR YOUR IBM HYPER PROTECT VIRTUAL SERVERS WORKSHEET or the tables listed on this topic to get an overall understanding of what information you will need to run the offering, and where to get such information.

Before you begin

- Ensure that you have the required hardware, software, network devices, and ports ready as listed on the System requirements.

Management server

The following table shows the required information for the x86 or Linux on IBM Z/LinuxONE (i.e., s390x architecture) management server.

Table 1. Management server checklist

#	Resource	The actual value	Example	Where to get
1	Architecture	x86 or s390x Linux	s390x	System administrator
2	Memory		4 GB	System administrator
3	vcpu/cores		2	System administrator
4	Disk size		50 GB	System administrator
5	Host name		**`management_server`**	**`hostname`**
6	Password for the user		**`root_user_password`** or **`sudo_user_password`**	System administrator
7	Internal IP address		**`192.168.40.251`**	Network administrator
8	Remote docker registry server		**`docker.io`**	Cloud administrator
9	Remote docker registry user name to register the base images		**`docker_base_user`**	Cloud administrator
10	Remote docker registry user password to register the base images		**`docker_base_passw0rd`**	Cloud administrator

To configure multiple aliases to one network interface controller (NIC) on the management server, see IP-Aliasing.

Secure Service Container partitions

The following table shows the required information you will need when configuring Secure Service Container storage.

Table 2. Secure Service Container partition checklist

#	Resource	The actual value	Example	Where to get
1	Partition IP address		`10.152.151.105`	System administrator
2	Master ID		`ssc_master_user`	System administrator
3	Master password		`ssc_master_password`	System administrator
4	Storage disks for quotagroups resizing		`3600507630affc4270000000000 02000` (FCP) or `0.0.78CA` (FICON DASD)	System administrator

Note: If you plan to use multiple Secure Service Container partitions, make sure you have a checklist for each partition.

A Hyper Protect Virtual Server instance with SSH daemon

The following table shows the required information you will need to create a Hyper Protect Virtual Server with SSH daemon on the Secure Service Container Partition.

Table 5. A Hyper Protect Virtual Server container checklist

#	Resource	The actual value	Example	Where to get
1	Partition IP address		`10.152.151.105`	System administrator
2	External network name		`encf900`	Cloud administrator
3	Container external IP address		`10.20.4.20`	cloud administrator
4	Internal network name		`encf900_internal_net work`	Cloud administrator
5	Internal IP address		`192.168.40.23`	Cloud administrator
6	Parent device		`encf900`	Appliance administrator
7	Gateway		`192.168.40.1`	Cloud administrator
8	Subnet		`192.168.40.0/24`	Cloud administrator
9	Repository name		`HpvsopBaseSSH`	Cloud administrator
10	Image tag		`1.2.4`	Cloud administrator
11	Virtual CPU number (vcpu)		`2`	Cloud administrator
12	Memory size (MB)		`2048`	Cloud administrator
13	Quotagroup name		qg_hpvsopbasessh	Cloud administrator
14	Quotagroup size (GB)		`20G`	Cloud administrator

For more information, see Creating a Hyper Protect Virtual Server instance. You can also build your application into a s390x-compatible container image, and deploy it into a Hyper Protect Virtual Server instance. For more information, see Deploying your applications securely.

A Secure Build virtual server

The following table shows the required information you will need to create a Secure Build virtual server on the Secure Service Container partition.

Table 3. A Secure Build container checklist

#	Resource	The actual value	Example	Where to get
1	Partition IP address		`10.152.151.105`	System administrator
2	Secure Build container name		`securebuildserver`	Cloud administrator
3	Virtual CPU number (vcpu)		`2`	System administrator
4	Memory (MB)		`2048`	System administrator
5	Storage for the Secure Build server application (GB)		`10`	System administrator
6	Storage for the Docker images built by Secure Build (GB)		`16`	System administrator
7	Storage for logs configuration data for the Secure Build Container (GB)		`2`	System administrator
8	Quotagroup of Secure Build server		`securebuild_qg`	Cloud administrator
9	Connection method (port-mapping/IP)		`IP`	System administrator
10	Internal network name (Only needed if an IP address is being used.)		`encf900`	Cloud administrator
11	External IP address		`10.20.4.12`	System administrator
12	Repository ID of the Secure Build server image		`SecureDockerBuild`	Cloud administrator
13	Tag of the Secure Build server image		`1.2.4`	Cloud administrator
14	Repository ID for your apps		`MyDockerAppImage`	Cloud administrator
15	Source code repository URL		`github.com:MyOrg/my-docker-app.git`	App developers or ISV
16	Source code branch		`master`	App developers or ISV
17	Private key for Source code repository		`/root/git_key`	App developers or ISV
18	Remote docker registry server		`docker.io`	Cloud administrator
19	Remote docker repository name for built images		`docker_writable_user/MyDockerAppImage`	Cloud administrator
20	Remote docker registry user name to push the images		`docker_writable_user`	Cloud administrator

#	Resource	The actual value	Example	Where to get
21	Remote docker registry user password to push the images		`docker_writeable_passw0rd`	Cloud administrator

For more information, see Building your application with the Secure Build virtual server Build.

Monitoring

The following table shows the required information you will need to set up the monitoring infrastructure for IBM Hyper Protect Virtual Servers.

Table 6. Monitoring infrastructure checklist

#	Resource	The actual value	Example	Where to get
1	Partition IP address		`10.152.151.105`	System administrator
2	Domain suffix		`first`	System administrator
3	DNS name		`example.com`	System administrator
4	Connecting port on partition (port-mapping)		`8443 and 25826`	System administrator
5	Private key for the monitoring infrastructure		`server.key`	`openssl` utility
6	Certificate for the monitoring infrastructure		`server-certificate.crt`	`openssl` utility
7	Certificates for the monitoring client		`myrootCA.crt`	`openssl` utility

For more information, see Working with Monitoring virtual servers.

Grep11

The following table shows the required information you will need to set up the GREP11 container for IBM Hyper Protect Virtual Servers.

Table 7. A GREP11 container checklist

#	Resource	The actual value	Example	Where to get
1	Partition IP address		`10.152.151.105`	System administrator
2	Crypto domain name		`07.0007`	System administrator
3	External IP address		`10.20.4.12`	System administrator
8	TLS key and certificate		`server.pem`, `server-key.pem`	`openssl` utility
9	CA certificate for mutual_TLS (Optional)		`ca.pem`	`openssl` utility

For more information, see Working with GREP11 virtual servers.

Next

You can download the IBM Hyper Protect Virtual Servers installation package by following the instructions on the Downloading the installation package topic.

Downloading IBM Hyper Protect Virtual Servers

You can get IBM Hyper Protect Virtual Servers software package from the IBM Passport Advantage.

Note: To download the Fix pack of IBM Hyper Protect Virtual Servers, see Downloading IBM Hyper Protect Virtual Servers Fix Pack.

This procedure is intended for users with the role *Cloud administrator*.

Before you begin

- Ensure you have the management server ready with one of the supported architectures as required in the Hardware requirements for management server section.
- Ensure that you install the OpenSSL or similar tool on the x86 or Linux on IBM Z/LinuxONE (i.e., s390x architecture) management server.

Procedure

On your x86 or Linux on IBM Z/LinuxONE (i.e., s390x architecture) management server, complete the following steps with root user authority.

1. Log in to IBM Passport Advantage website by using your IBM account ID and password. Contact your sales representative if you do not have one.

2. Go to **My Programs**, and then select the **IBM Hyper Protect Virtual Servers** program.

3. Download IBM Hyper Protect Virtual Servers image `<part_number>.tar.gz` to your x86 or Linux on IBM Z/LinuxONE (i.e., s390x architecture) management server. Note that `<part_number>` is the package name of Hyper Protect Virtual Servers. You need to replace the `<part_number>` with the actual value in the following steps accordingly.

    - For IBM Hyper Protect Virtual Servers version 1.2.3, the `<part_number>` is `M02VFEN`.
    - For IBM Hyper Protect Virtual Servers version 1.2.3, the `<part_number>` is `G00GWZX`.
    - For IBM Hyper Protect Virtual Servers version 1.2.2.1, or 1.2.2, the `<part_number>` is `CC7L3EN`.
    - For IBM Hyper Protect Virtual Servers version 1.2.1.1, or 1.2.1, the `<part_number>` is `CC75CEN`.
    - For IBM Hyper Protect Virtual Servers version 1.2.0.1, or 1.2.0, the `<part_number>` is `CC37UEN`.
4. On the x86 or Linux on IBM Z/LinuxONE (i.e., s390x architecture) management server with root user authority, create an installation directory to store IBM Hyper Protect Virtual Servers image and configuration files.

    ```
    mkdir /opt/<installation_directory>
    ```

5. Change to the installation directory, and extract the compressed file on the x86 or Linux on Z server.

    ```
    cd /opt/<installation_directory>
    gunzip <part_number>.tar.gz
    tar -xvf <part_number>.tar
    ```

    You will get the following files in the current directory:

    - `<part_number>.tar.gz`, the offering image tar file.

    - `<part_number>.sig`, the signature file for the offering image.
    - `<part_number>.pub`, the public key issued by IBM for the offering image.
6. To verify the integrity of IBM Hyper Protect Virtual Servers image tar file, run the following example command by using the signature file with the `.sig` suffix and the public key issued by IBM with the suffix `.pub` along with the image tar file.

    ```
    openssl dgst -sha256 -verify <part_number>.pub -signature <part_number>.sig
    <part_number>.tar.gz
    ```

7. Extract the compressed tar file on the x86 or Linux on IBM Z/LinuxONE (i.e., s390x architecture) management server.

    ```
    cd /opt/<installation_directory>
    tar -xvzf <part_number>.tar.gz
    ```

    Note: Some of the extracted files are in `*.gz format`, and they should be used as is and should not be extracted once again.

Result

After extracting the installation package image, you can see the similar layout of files and directories under the `<installation_directory>` directory.

- When you are using IBM Hyper Protect Virtual Servers Version 1.2.1:

```
.
├── License
│   ├── LA_cs
│   ├── LA_de
│   ├── LA_el
│   ├── LA_en
│   ├── LA_es
│   ├── LA_fr
│   ├── LA_in
│   ├── LA_it
│   ├── LA_ja
│   ├── LA_ko
│   ├── LA_lt
│   ├── LA_pl
│   ├── LA_pt
│   ├── LA_ru
│   ├── LA_sl
│   ├── LA_tr
│   ├── LA_zh
│   ├── LA_zh_TW
│   ├── LI_cs
│   ├── LI_de
│   ├── LI_el
│   ├── LI_en
│   ├── LI_es
│   ├── LI_fr
│   ├── LI_in
│   ├── LI_it
│   ├── LI_ja
│   ├── LI_ko
│   ├── LI_lt
│   ├── LI_pl
│   ├── LI_pt
│   ├── LI_ru
│   ├── LI_sl
│   ├── LI_tr
│   ├── LI_zh
│   ├── LI_zh_TW
│   ├── non_ibm_license
```

```
│   └── notices
├── M02VFEN.tar.gz
├── bin
│   ├── hpvs_s390x
│   └── hpvs_x86
├── config
│   ├── templates
│   │   ├── virtualserver.template.readme.yml
│   │   └── virtualserver.template.yml
│   └── yaml
│       ├── secure_build.yml.example
│       ├── secure_create.yml.example
│       ├── vs_configfile_readme.yml
│       ├── vs_grep11.yml
│       ├── vs_hpvsopbase.yml
│       ├── vs_hpvsopbasessh.yml
│       ├── vs_monitoring.yml
│       ├── vs_regfiledeployexample.yml
│       └── vs_securebuild.yml
├── envcheck.sh
├── images
│   ├── CollectdHost.tar.gz
│   ├── HpvsopBase.tar.gz
│   ├── HpvsopBaseSSH.tar.gz
│   ├── Monitoring.tar.gz
│   ├── SecureDockerBuild.tar.gz
│   └── hpcsKpGrep11_runq.tar.gz
├── mustgather.sh
├── readme.txt
├── secure-service-container-for-hpvs.appliance.4.3.5.img.gz
├── setup.sh
├── swidtag
│   └── ibm.com_IBM_Hyper_Protect_Virtual_Servers-1.2.4.swidtag
└── version
```

Note

- `readme.txt`, which is the general README file for IBM Hyper Protect Virtual Servers.
- `License`, a directory that contains the license files of IBM Hyper Protect Virtual Servers.
- `version`, which states the current version of IBM Hyper Protect Virtual Servers.
- `./secure-service-container-for-hpvs.appliance.4.3.5.img.gz`, which is the hosting appliance to be installed on the IBM Z or LinuxONE system.
- `./images/HpvsopBase.tar.gz`, which is the base image of a Hyper Protect Virtual Server container without the secure shell (SSH) access.
- `./images/HpvsopBaseSSH.tar.gz`, which is the base image of a Hyper Protect Virtual Server container with the secure shell (SSH) access.
- `./images/CollectdHost.tar.gz`, which is the base image of collectd-host container of the monitoring infrastructure.
- `./images/SecureDockerBuild.tar.gz`, which is the docker image of the Secure Build container.
- `./images/Monitoring.tar.gz`, which is the base image of monitoring-host container of the monitoring infrastructure.
- `./images/hpcsKpGrep11_runq.tar.gz`, which is the base image of the GREP11 container.
- `./config/templates/virtualserver.template.yml`, which is the template example of network, quotagroup, and resource definitions for the virtual server.
- `./config/yaml/`, a directory that contains configuration example files for the Hyper Protect Virtual Server containers.
- `/envcheck.sh`, the shell script that automates checking the prerequisites for Linux management server for setting up the IBM Hyper Protect Virtual Servers environment.
- `/setup.sh`, the shell script that automates setting up the IBM Hyper Protect Virtual Servers environment.
- `./mustgather.sh`, an automated script to collect debug information when you want to open a support ticket (This is applicable for IBM Hyper Protect Virtual Servers Version 1.2.2). For IBM Hyper Protect Virtual Servers Version 1.2.1, the script is available in the `./config` folder.

Next

The IBM Hyper Protect Virtual Servers CLI is installed as part of the shell script. For more information, see Setting up the environment by using the setup script topic.

Downloading IBM Hyper Protect Virtual Servers Fix Pack

You can download the Fix Pack of IBM Hyper Protect Virtual Servers from IBM Fix Central.

This procedure is intended for users with the role *Cloud administrator*.

Before you begin

- Ensure you have the management server ready with one of the supported architectures as required in the Hardware requirements for management server section.
- Ensure that you install the OpenSSL or similar tool on the x86 or Linux on IBM Z/LinuxONE (i.e., s390x architecture) management server.

Procedure

On your x86 or Linux on IBM Z/LinuxONE (i.e., s390x architecture) management server, complete the following steps with root user authority.

1. Go to IBM Fix Central website.

2. Locate IBM Hyper Protect Virtual Servers Fix packs either by entering **IBM Hyper Protect Virtual Servers** on the **Find product** panel, or select **IBM Hyper Protect Virtual Servers** under the **z Systems** product group on the **Select product** panel.

3. Select the version and platform, and then click **Continue**.

4. Select the fix pack from the list on the **Select fixes** page, and then click **Continue**.

5. Log in to IBM Fix Central site with your IBM ID and password as prompted, and then download the selected Fix Pack installation image to your x86 or Linux on IBM Z/LinuxONE (i.e., s390x architecture) management server.

Next

- To install the fix pack for IBM Hyper Protect Virtual Servers Version 1.2.2.1, delete the */usr/local/bin/hpvs* directory and follow the instructions in the topic Upgrading IBM Hyper Protect Virtual Servers.
- To install the fix pack for IBM Hyper Protect Virtual Servers Version 1.2.1.1, delete the */usr/local/bin/hpvs* directory and follow the instructions from step 4 of the topic Downloading IBM Hyper Protect Virtual Servers.

Note:
For information about the files and directory structure, see File and directory structure of IBM Hyper Protect Virtual Servers.

Setting up the Secure Service Container Partition

The following topics shows how to setup Secure Service Container Partitions when using IBM Hyper Protect Virtual Servers.

1. Creating the Secure Service Container partition
2. Installing the Hyper Protect Virtual Servers hosting appliance
3. Configuring the storage on the Secure Service Container partition
4. Configuring the network on the Secure Service Container partition

Creating the Secure Service Container partition

Use this procedure to configure a Secure Service Container partition on a host system and later install IBM Hyper Protect Virtual Servers on that partition.

This procedure is intended for users with the role *system administrator*.

Before you begin

- Refer to the checklist that you prepared on this topic Planning for the environment.
- Ensure that the hosting system is one of supported servers as required in the Hardware requirements for Secure Service Container partition section.
- Check that you download Secure Service Container User's Guide, SC28-6978-02a.

Procedure

To create and manage Secure Service Container partitions, you can use specific tasks on the Hardware Management Console (HMC) for a host system running either in standard mode (that is, with Processor Resource/System Manager or PR/SM), or with Dynamic Partition Manager (DPM) enabled. For more information about the host system, see the appropriate overview on the IBM® Redbooks® website at http://www.redbooks.ibm.com/. For example, for the z15, see the IBM z15 Technical Introduction, SG24-8850.

- For a host system (CPC) running in standard mode

  1. Open the **Customize/Delete Activation Profiles** task, and then select **SSC** mode on the **Customize Image Profiles** page.

  2. Configure the processor requirements on the **Processor** page, specify partition security options on the **Security** page, and specify the amount of storage required on the **Storage** page.

  3. Provide or modify any cryptographic controls as appropriate on the **Crypot** page.

  4. On the **SSC** page, ensure the **Secure Service Container installer** option is selected under **Boot selection** if you create the partition for the first time, and then provide values for the default primary user ID, password, and IP address of the network adapter for the Secure Service Container partition.

  5. Click **Save** to save the changes and wait for the partition to be created.

  6. Select the image of the Secure Service Container partition, and start the Secure Service Container partition by using the **Activate** task.

- For a host system with DPM enabled

  1. Open the HMC **New Partition** task, and then select **Secure Service Container** as the **Partition Type** from the list.

  2. Provide the values for the primary user ID and password as prompted.

3. Define the number of virtual processors for the partition on the **Processor** page, define the initial and maximum amounts of memory to be assigned to the partition on the **Memory** page. The minimal initial amount of a Secure Service Container partition is 4 GB.
4. Define all of the network interface cards (NICs) for the partition on the **Network** page. For a Secure Service Container partition, you must also specify at least one NIC for communication with the Secure Service Container web interface.
5. Attach storagegroups or create host bus adapters (HBAs) for the partition on the **Storage** page.
6. Configure the cryptographic features on the **Cryptos** page as needed.
7. On the **Boot** page, note that option set in the "Boot from" menu is **Secure Service Container**. This boot option cannot be changed unless you first change the partition type.
8. Click **Finish** to save the partition definition, and then wait for DPM creating the partition.
9. Select the image of the Secure Service Container partition, and activate the partition by selecting **Yes** on the **Start** task page.

Note:

- Write down the following values specified in the image profile (standard mode system) or the partition definition (DPM-enabled system) for the Secure Service Container partition when you configure the Secure Service Container. You will need them when configuring the appliance network and creating cluster nodes.
  - primary user ID
    - Master password
    - IP address
- For more detailed information on how to create and start the Secure Service Container partition, see Secure Service Container User's Guide, SC28-6978-02a.

Next

You can install the hosting appliance onto the Secure Service Container partition by following the instructions on Installing the Hyper Protect Virtual Servers hosting appliance.

Installing the Hyper Protect Virtual Servers hosting appliance

Use this procedure to install and start the Hyper Protect Virtual Servers hosting appliance in a Secure Service Container partition on the IBM z or LinuxONE server.

This procedure is intended for users with the role *appliance administrator*.

Note:

- The Hyper Protect Virtual Servers hosting appliance is an enhanced version of the IBM Secure Service Container software appliance.
- The Hyper Protect Virtual Servers hosting appliance displays with the name **IBM Secure Service Container** on the Secure Service Container user interface.
- The Hyper Protect Virtual Servers hosting appliance uses all of the IBM Secure Service Container documentation and techniques to install, administer, and maintain.
- The Hyper Protect Virtual Servers hosting appliance version numbering scheme is unique to the Hyper Protect Virtual Servers hosting appliance, as opposed to the general Secure Service Container verion numbering

scheme.

- Only one appliance can be installed and run in a Secure Service Container partition at any given time; this type of partition does not support running multiple appliances simultaneously. You can define more than one Secure Service Container partitions on the same system, and run instances of the same appliance in each one. In this case, each partition must use separate storage devices.

Before you begin

- Check that you have the appliance image `secure-service-container-for-hpvs.appliance.<version_number>.img.gz` in the installation directory. For instructions, see Downloading the installation package.
- Check that you have the Secure Service Container partition created to install the hosting appliance as instructed in the Creating the Secure Service Container partition topic.
- Check that you download Secure Service Container User's Guide, SC28-6978-02a.

Procedure

Complete the following tasks through the browser of your choice.

1. Log in to the Secure Service Container installer by using the primary user ID and password in your browser. For example, `https://<secure_service_container_partition_ip_address>`.

2. On the main page, click the plus (+) icon to install image files from local disk. The page display changes to the **Install Software Appliance** page.

3. On the **Install Software Appliance** page, select the **Upload image to target disk** option, and then locate the appliance image file on your local disk under the **Local Installation Image** section.

4. Under **Target Disk on Server**, select the device type **FICON DASD** or **FCP**, and then click **Apply** to upload the appliance image to the target disk on the server. **Note:**

   - You can only specify one type of disk (either DASD or FCP) during the appliance installation stage.
   - Target FCP disks must be large enough to fit the uncompressed appliance, with an additional 2 GB for the Secure Service Container installer to use.
5. Click **Reboot** on the confirmation dialog to have the installer automatically reactivate the partition. The Secure Service Container installer uploads the appliance image to the target disk, and prepares the partition to load the appliance after the next reboot. a. When the reboot process begins, the installer displays the Reboot window. b. If an IP address type other than DHCP is in use for the appliance page, the Secure Service Container installer redirects the browser to the software appliance page.

6. On the appliance page, accept the self-signed certificate for the SSL connection, and log in to the Secure Service Container user interface by using the primary user ID and password.

For more detailed instructions, see the following topic after you download Secure Service Container User's Guide, SC28-6978-02a.

- Chapter 13 - Installing a new software appliance in a Secure Service Container partition

Next

You can configure the storage on the Secure Service Container partition as instructed in the Configuring the storage on the Secure Service Container partition topic.

Configuring the storage on the Secure Service Container partition

Use this procedure to make storage devices assigned to the Secure Service Container partition available in IBM Hyper Protect Virtual Servers. These resources can then be utilized by the containerized applications running on the Secure Service Container partition.

This procedure is intended for users with the role *appliance administrator*.

Before you begin

- Refer to the checklist that you prepared in the topic Planning for the environment.
- Check with the cloud administrator the list of requirements of the containerized application to assign sufficient resources (disk space, network adapters) to IBM Hyper Protect Virtual Servers.
- Check with the IBM Z or LinuxONE system administrator that sufficient resources are assigned to the Secure Service Container partition to fit the requirements of the containerized application.
- Check with the IBM Z or LinuxONE system administrator to get the disk IDs that can be used for the Secure Service Container partition.
- Check that you have downloaded the Secure Service Container User's Guide, SC28-6978-02a.

Procedure

Storage resources are grouped into storage pools that are created when the appliance is built. A storage pool is a uniquely named collection of storage disks on which the appliance file system is mounted.

- Use the Secure Service Container user interface to manage the storage resources. On the Secure Service Container partition user interface, use the **Storage Disks by Storage Pool** page to add FICON DASD or FCP disks to a storage pool. Each storage pool must contain only one type of storage: either FICON DASD or FCP disks.

  For more detailed instructions, see *Viewing and managing storage resources* section in **Chapter 14 Using the Secure Service Container user interface** after you download Secure Service Container User's Guide, SC28-6978-02a.

Complete the following steps by using the Graphical User Interface of the Secure Service Container.

1. Login to the User Interface of the Secure Service Container by using the IP address of the Secure Service Container.

2. In the navigation pane, click the **Storage** icon.

3. In the **LV Data Pool** area, click the plus sign **(+)** to add a disk to the LV Data Pool.

4. In the **Available Devices area** select the disks that you want to add, and click the **>>** icon to move the selected disks to the **Assigned Devices** list.

5. Click **Apply**. The **Confirm Add disk** page is displayed.

6. Review your selection and click **Yes** to proceed. You can view the status of attaching and formatting of the disks. When the attach and format of the disks are complete, you can view the details of the disks that you added in the LV Data Pool area.
   Note:

   - The disks that you have selected will be formatted.

- The volumes for Hyper Protect Virtual Server containers and Secure Build containers can be created and allocated from the storage pool when those containers are created on the Secure Service Container partition.

Next

You can configure the network devices by following the instructions on Configuring the network on the Secure Service Container partition.

Configuring the network on the Secure Service Container partition

You can configure the network devices for the hosting appliance by using the Secure Service Container user interface. The containers on the Secure Service Container partitions communicate through the Ethernet-type or VLAN-type connections over the network devices bound to Open Systems Adapter-Express (OSA-Express) devices, or Hipersockets. Hipersockets are supported in IBM Hyper Protect Virtual Servers version 1.2.3, or later.

If you want the Hyper Protect Virtual Server container on the Secure Service Container partition to be accessed by external services, you must configure two network devices with one for internal communication, and another for external access. You can configure one network device to each of the OSA-Express devices on the Secure Service Container partitions, or multiple network devices on one OSA-Express device. You can also achieve internal network communication between Hyper Protect Virtual Servers within the same IBM Z system by configurating a Hipersocket device. This procedure is intended for users with the role *appliance administrator*.

Before you begin

- Refer to the checklist that you prepared on this topic Planning for the environment.
- Check that you have the connection information to each Secure Service Container partition. For more information, see Creating Secure Service Container partitions.
- Check that you install the hosting appliance by following the instructions on Installing the Hyper Protect Virtual Servers hosting appliance.

Procedure

Complete the following steps to configure the network devices.

1. Connect to the Secure Service Container partition through the browser of your choice. For example, **`https://<secure_service_container_partition_ip_address>`**.

2. On the **Login** page, enter the master use ID and password values that you supplied in the image profile (standard mode system) or the partition definition (DPM-enabled system), and click **Login**.

3. In the navigation pane, click the **Network** icon to display the network connections page.

4. Select one of the network devices to get the channel path identifier (CHPID) of the OSA-Express device. For example, **`encf900_network`** is the network device name, and **`AA`** is the CHPID. The network device can only be used for the external communication for the Hyper Protect Virtual Server container. You can choose the Hipersockets option if it is displayed as available, for internal network connection.

5. Configure another network device on the Secure Service Container partition.

    - For an ethernet-type connection:

    - Click the plus (+) icon to add a new connection, and then select **Ethernet** as the connection type.
    - Select a new network device from the drop-down list. Ensure that the CHPID in the Device Details section is different from the one in step 4. For example, the network device name is `encf900_internal_network`, and the CHPID is `AB`. This network device can only be used for the internal communication for the Hyper Protect Virtual Server container.
    - Use the default value for the **Port** field, and set the connection state to **Active**.
    - If you chose the Hipersockets option (in sub-step 2 of step 5), for the **Layer2** field, you must select a value of 1 from the list.
    - Use **Disabled** for both IPV4 and IPV6 addresses fields.
- For a VLAN-type connection, ensure that your OSA or Hipersocket device is tagged with an VLAN ID (for example, `1121`) and the OSA or Hipersocket device is connected with the trunk port of the switch.

    - Click the plus (+) icon to add a new connection, and then select **VLAN** as the connection type.
    - Select a parent device (also known as a tagged OSA or Hipersocket device) from the drop-down list. If the parent device is not available, click the plus (+) icon to create a parent device. For example, the parent device name is `encf300`.
    - Enter the VLAN ID by which the OSA or Hipersocket device is tagged. For example, `1121`.
    - Use the auto-generated connection name. For example, `vxlan0f300.1121`.
    - If the DHCP is not configured in your network, select the **Manual** checkbox on the **IPv4** tab and assign an appropriate IP address according to your network.
    - Set the connection state on the **General** tab to **Active**.
    - Click the **ADD** button to save the changes.

Note: The Secure Service Container partition requires configuration of the necessary DNS entries if you plan to explore the following features in IBM Hyper Protect Virtual Servers.

- Configure appropriate DNS entry or entries for Secure Build containers on the IBM Hyper Protect Virtual Servers partition, so that the Secure Build containers can access the github source code URLs. This DNS configuration is performed on the Hardware Management Console (HMC) as part of the Secure Service Container LPAR profile's network configuration.
- Configure a DNS entry for the monitoring infrastructure, so that the monitoring client tools can access the monitoring infrastructure on the IBM Hyper Protect Virtual Servers partition.
- Configure a DNS entry for the GREP11 container, so that the client application code can access the GREP11 container on the IBM Hyper Protect Virtual Servers partition.

For more information on how to configure DNS entries on the Secure Service Container partition, see the following topic after you download Secure Service Container User's Guide, SC28-6978-02a.

- Chapter 14, "Using the Secure Service Container user interface", section "Viewing and managing network connections"
- Chapter 3 or 7, "Configuring a Secure Service Container partition"

Next

- You can deploy your workloads by following the instructions on Building your application with the Secure Build virtual server.

Working with IBM Hyper Protect Virtual Servers

- Setting up the IBM Hyper Protect Virtual Servers environment
- Registering base images in the remote registry server
- Creating a Hyper Protect virtual server instance
- Generating the signing keys
- Enabling ports

- Building your application with the Secure Build virtual server
- Verifying the signature of the manifest file
- Rolling keys in a Secure Build container
- Deploying your applications securely
- Refreshing registered repositories with a new signing key pair
- Working with Monitoring virtual servers
  - Creating CA signed certificates for the monitoring infrastructure
- Working with GREP11 virtual servers
  - Creating OpenSSL certificates for GREP11 containers
  - Testing the GREP11 virtual server
- Backing up and restoring IBM Hyper Protect Virtual Servers
- Undeploying virtual servers
- Updating virtual servers
- Uninstalling IBM Hyper Protect Virtual Servers
  - Uninstalling the Hyper Protect Virtual Servers CLI tools
  - Uninstalling Secure Service Container partitions
- Updating Hyper Protect Virtual Server containers
- Upgrading IBM Hyper Protect Virtual Servers
- Upgrading IBM Hyper Protect Virtual Servers 1.2.1.1, or 1.2.1 to 1.2.2

Setting up the environment by using the setup script

The IBM Hyper Protect Virtual Servers Version 1.2.1, or later, provides an automated procedure that simplifies the installation and configuration of the IBM Hyper Protect Virtual Servers environment.

This procedure is intended for users with the role *cloud administrator*.

Before you begin

- Refer to the checklist that you prepared for the management server in the topic Planning for the environment.

- When you are using IBM Hyper Protect Virtual Servers version 1.2.1, or 1.2.1.1, check that you have root user privilege. For a non-root user with root privilege, use **`sudo`** where ever it is required.

- When you are using IBM Hyper Protect Virtual Servers Version 1.2.2, you can execute the setup script either as a root or non-root user. A non-root user will be prompted to provide the user password during script execution.

- Check that you have IBM Hyper Protect Virtual Servers installation binary on the x86 or Linux on IBM Z/LinuxONE (i.e., s390x architecture) management server. For more information, see Downloading IBM Hyper Protect Virtual Servers.

- Check that the x86 or Linux on IBM Z/LinuxONE (i.e., s390x architecture) management server has the following required software packages:

  - One of supported Docker versions
  - OpenSSL
  - GPG
  - The haveged utility

Procedure

On your x86 or Linux on IBM Z/LinuxONE (i.e., s390x architecture) management server, complete the following steps under the **`<installation_directory>`** directory with root user authority (applicable only for IBM Hyper Protect Virtual Servers Version 1.2.1).

1. Run the **`setup.sh`** shell script to complete the environment preparation on the management server. When you run the setup script the first time, you must accept the license in order to continue with the setup.

   ```
   ./setup.sh -e LICENSE=accept
   ```

   A message is displayed stating that the license was accepted and the setup continues.
   To view the license you can run the following command.

   ```
   ./setup.sh -e LICENSE=view -e LANG=xx
   ```

   where **`xx`** is the language code. See Available language codes for the list of available language codes. If no language code is specified, the default language is used which is English.
   You can also deny accepting the license by running the following code. However, you cannot proceed with the setup without accepting the license.

   ```
   sh setup.sh -e LICENSE=deny
   ```

   If you have already accepted the license earlier and want to run the setup script again, you can use the following command.

   ```
   ./setup.sh
   ```

   Note: If you have not accepted the license even once, then running the script results in an error and you are prompted to accept the license.

   The **`setup.sh`** shell script automates the following actions:

   - Invoke the **`envcheck.sh`** script to validate the prerequisites. The **`envcheck.sh`** shell script automates checking of the following requirements of the management server and does the following:
     - The system architecture: when the system architecture is not x86 or Linux on IBM Z/LinuxONE (i.e., s390x architecture), the script fails and a message stating that the architecture is not supported is displayed.
     - The Linux distribution: When the Linux distribution is not Ubuntu or RHEL, the script fails and a message is displayed stating that the script is supported only Ubuntu and RHEL based systems.
     - The Ubuntu or RHEL Version: When the Ubuntu Version is not 18.04 or later, or 16.04 or later, or the RHEL Version is not 7.X or later, or 8.X or later, a warning message is displayed indicating that the Ubuntu or RHEL versions are not supported and the script continues execution.
     - GPG version: When the GPG version is not 2.2.4 or later, the script fails and a message is displayed stating that the GPG version must be upgraded.
     - Docker Installation: When Docker is not installed, the script fails. Also, when Docker is not at version 19.03.2 or later for x86, and 18.06.3 or later for s390x, the script fails.
     - Number of CPU cores: When number of cores is less than 4 for x86 and 1 for x390x, a warning message is displayed that there are lesser number of cores than required and the script continues execution.
     - Amount of memory: When the memory is less than 8 GB, a warning message is displayed that the memory is less than required and the script continues execution.
     - Disk space: When the disk space is less than 150 GB, a warning message is displayed that the disk space is less than required and the script continues execution.
     - OpenSSL: When OpenSSL is not installed, the script fails. A message prompting you to install OpenSSL and retry the script is displayed.
     - The haveged utility: When haveged is not installed, the script fails. A message prompting you to install haveged and retry the script is displayed.
   - Sets the **`PATH`** for the **`hpvs`** commands
   - Creates the **`$HOME/hpvs`** (working directory) directory structure and copies all the keys, registry files, and all the required config files and creates symbolic links of the images to this folder.

   - Extracts and verifies the base images in the installation directory.
   - Loads the base images **`hpvsop-base`** and **`hpvsop-base-ssh`** into your local Docker registry.
   - Creates and updates the **`$HOME/hpvs/config/reg.json`** config file with the registry details for your remote Docker registry server, or with the IBM Cloud Registry details. The credentials will be encrypted after the script completes.
   - Updates the **`$HOME/hpvs/hosts`** config file with the Secure Service Container partition information. You need to enter the IP address of the partition, and connection credentials.
2. You are prompted to select an option for configuring the container registry. Select a value of **1** when you want to use Docker Hub (publicly hosted). Select a value of **2** when you want to use the IBM Cloud Registry. Use one of the following set of instructions depending on the option you choose for configuring the container registry.

   1. When the script is executing the setup of the Docker registry (when you chose a value of 1), you are prompted to enter the following information.

      - The Docker registry name, for example **`docker_hub`**.
      - The Docker registry Username, for example **`docker_username`**.
      - The Docker registry password. Type in the password of the Docker registry.
   2. When the script is executing the setup of the IBM Cloud Registry (when you chose a value of 2), you are prompted to enter the following information.

      - The IBM Cloud Registry name, for example **`cloud_reg`**.
      - The IBM Cloud Registry Server URL, for example **`us.icr.io`**.
      - The CONTENT_TRUST_SERVER URL, for example **`https://notary.us.icr.io:/`**
      - The IBM Cloud API key: Type in the IBM Cloud API key. (For more information, see the section Creating an IBM Cloud API Key).
3. When the script is executing the setup of the hosts **`config`** file, you are prompted to enter the following information.

   - The Secure Service Container LPAR (Host) IP address, for example **`10.20.4.23`**.
   - The Secure Service Container LPAR (Host) Name, for example **`zbcor5`**.
   - The Username of the Secure Service Container LPAR, for example **`blockchain`**.
   - The password.
4. To push base images to the container registry, refer the instructions provided in Registering base images in the remote registry server.

The following is an example of the directory structure (working directory) created by the setup script which shows the symbolic links that were created.

```
 .
 ├── config
 │   ├── grep11
 │   │   ├── images
 │   │   │   └── hpcsKpGrep11_runq.tar.gz -> /var/124-
GA/images/hpcsKpGrep11_runq.tar.gz
 │   │   ├── keys
 │   │   ├── regfiles
 │   │   └── vs_grep11.yml
 │   ├── hpvsopbase
 │   │   ├── images
 │   │   │   └── HpvsopBase.tar.gz -> /var/124-GA/images/HpvsopBase.tar.gz
 │   │   ├── keys
 │   │   ├── regfiles
 │   │   └── vs_hpvsopbase.yml
 │   ├── hpvsopbasessh
 │   │   ├── images
 │   │   │   └── HpvsopBaseSSH.tar.gz -> /var/124-GA/images/HpvsopBaseSSH.tar.gz
 │   │   ├── keys
 │   │   ├── regfiles
 │   │   └── vs_hpvsopbasessh.yml
 │   ├── monitoring
 │   │   ├── images
 │   │   │   ├── CollectdHost.tar.gz -> /var/124-GA/images/CollectdHost.tar.gz
 │   │   │   └── Monitoring.tar.gz -> /var/124-GA/images/Monitoring.tar.gz
 │   │   ├── keys
 │   │   ├── regfiles
 │   │   └── vs_monitoring.yml
 │   ├── reg.json
 │   ├── securebuild
 │   │   ├── images
 │   │   │   └── SecureDockerBuild.tar.gz -> /var/124-
GA/images/SecureDockerBuild.tar.gz
 │   │   ├── keys
 │   │   ├── regfiles
 │   │   ├── secure_build.yml.example
 │   │   ├── secure_create.yml.example
 │   │   └── vs_securebuild.yml
 │   ├── templates
 │   │   ├── virtualserver.template.readme.yml
 │   │   └── virtualserver.template.yml
 │   ├── vs_configfile_readme.yml
 │   └── vs_regfiledeployexample.yml
 ├── hosts
 └── logs
```

Where

- **`images/HpvsopBase.tar.gz`**, which is the base image of a Hyper Protect Virtual Server container without the secure shell (SSH) access.
- **`images/HpvsopBaseSSH.tar.gz`**, which is the base image of a Hyper Protect Virtual Server container with the secure shell (SSH) access.
- **`images/CollectdHost.tar.gz`**, which is the base image of collectd-host container of the monitoring infrastructure.
- **`images/SecureDockerBuild.tar.gz`**, which is the docker image of the Secure Build container.
- **`images/Monitoring.tar.gz`**, which is the base image of monitoring-host container of the monitoring infrastructure.
- **`images/hpcsKpGrep11_runq.tar.gz`**, which is the base image of the GREP11 container.
- **`config/templates/virtualserver.template.yml`**, which is the template example of network, quotagroup, and resoource definitions for the virtual server.
- **`config/yaml/`**, a directory that contains configuration example files for the Hyper Protect Virtual Server containers.
- **`config/grep11/keys`**, **`config/grep11/regfiles`**, **`config/hpvsopbase/keys`**, **`config/hpvsopbase/regfiles`**, **`config/hpvsopbasessh/keys`**, **`config/hpvsopbasessh/regfiles`**, **`config/securebuild/keys`**, **`config/securebuild/regfiles`**, **`config/monitoring/keys`**, and **`config/monitoring/regfiles`**, you can use these folders to save the keys or .enc files you generate.

  After the script completes, you can run the **`hpvs`** command locally to validate the environment is ready for use. The **`hpvs`** command shows you a list of supported actions to manage IBM Hyper Protect Virtual Servers. For more information about the **`hpvs`** command, see Commands for IBM Hyper Protect Virtual Servers.

Available language codes

Language Code	Language
cs	Slovak
en	English
in	Malay
ko	Korean
pt	Portuguese
tr	Turkish

Language Code	Language
de	German
es	Spanish
it	Italian
ru	Russian
zh	Chinese
el	Greek
fr	French
ja	Japanese
pl	Polish
sl	Slovenian
zh_TW	Chinese Traditional

Next

To configure the environment for IBM Hyper Protect Virtual Servers, follow the instructions in Creating a Hyper Protect Virtual Server instance.

Registering base images in the remote registry server

You must register the base images in the remote docker repository by using your ID and password. The remote docker repository can be Docker Hub or IBM Cloud Container Registry.

Note that the following context uses Docker Hub for demonstration. You can use the equivalent values or settings if you choose to use IBM Cloud Container Registry. For more information, see Getting started with IBM Cloud Container Registry.

The base images are the default Hyper Protect Virtual Server container images that can be used to host your application code, and include two different types of container images for your development and production environments.

- **HpvsopBaseSSH**, which packages the SSH daemon into the default Hyper Protect Virtual Server container image, so that you can log in to the Hyper Protect Virtual Server by using the secure shell and your private key for debugging and development.
- **HpvsopBase**, which excludes the SSH daemon on the default Hyper Protect Virtual Server container image, and can be used in the production environment.

This procedure is intended for users with the role *cloud administrator* or *Application developer or ISV*.

Before you begin

- Check that you have the account ID and password on the remote docker registry server to create repositories for base images. For example, **docker_base_user** is your user ID on the remote docker registry server.
- Check that you have installed the GPG command line tool on the x86 or Linux on IBM Z/LinuxONE (i.e., s390x architecture) management server. For more information, see GNU Privacy Guard.
- Check that you enable Docker Content Trust (DCT) for your remote docker registry server. For more information, see Content trust in Docker or Setting up your trusted content environment for IBM Container Registry.

```
export DOCKER_CONTENT_TRUST=1
```

- Refer to the checklist that you prepared for the Hyper Protect Virtual Server on this topic Planning for the environment.

Procedure

Complete the following steps under the **`<installation_directory>/VS/hpvs-cli/config`** directory with root user authority.

1. Install the Hyper Protect Virtual Server base images to your x86 or Linux on IBM Z/LinuxONE (i.e., s390x architecture) management server.

   a. Extract the base images into different folders.

   ```
       mkdir <destination-folder-HpvsopBase>
       mkdir <destination-folder-HpvsopBaseSSH>
       tar -xvf HpvsopBase.tar.gz -C <destination-folder-HpvsopBase>
       tar -xvf HpvsopBaseSSH.tar.gz -C <destination-folder-HpvsopBaseSSH>
   ```

   Note: You can omit this step if you have already run the setup script. See Setting up the environment by using the setup script.

   b. Create the docker loadable binary under the same directory. Note that the warning message **`gpg: Can't check signature: No public key`** can be safely ignored when running the following **`gpg`** commands.

   ```
       gpg <destination-folder-HpvsopBase>/HpvsopBase.tar.gz.sig
       gpg <destination-folder-HpvsopBaseSSH>/HpvsopBaseSSH.tar.gz.sig
   ```

   Note: You can omit this step if you have already run the setup script. See Setting up the environment by using the setup script.

   c. Log in to the remote docker repository.

   - For Docker Hub, run the **`docker login`** command. For more information, see Docker Login command.
   - For IBM Cloud Container Registry, run **`docker login -u iamapikey -p <iam_api_key> <region>.icr.io`** command. For more information, see Using Docker to authenticate with an API key.

   d. Install the base images by using the **`docker load`** commands.

   ```
       docker load -i <destination-folder-HpvsopBase>/HpvsopBase.tar.gz
       docker load -i <destination-folder-HpvsopBaseSSH>/HpvsopBaseSSH.tar.gz
   ```

   Note: You can omit this step if you have already run the setup script. See Setting up the environment by using the setup script.

   e. Run the **`docker images`** command to check whether the base images are loaded into the local registry successfully.

   ~~~~
   ```
   REPOSITORY                                                      TAG     IMAGE ID
   CREATED            SIZE
   sys-zaas-team-hpvsop-dev-docker-local.artifactory.\
   swg-devops.com/zaas/hyperpvsop-base-image                       1.2.4   c6a593192565   3
   days ago         1.04GB
   sys-zaas-team-hpvsop-dev-docker-local.artifactory.\
   swg-devops.com/zaas/hyperpvsop-base-ssh-image                   1.2.4   a6252e869355   3
   days ago         1.04GB

   ```
   ~~~~

2. Create two repositories in your namespace for both the **`hpvsop-base`** image and the **`hpvsop-base-ssh`** image on the Docker Hub. For example, **`docker_base_user/hpvsop-base`** and **`docker_base_user/hpvsop-base-ssh`**. Note that the repository name must match the image name.

3. Use the **docker tag** command to tag base images with the same ID used by the CLI tool. For example, **1.2.3** is the tag ID of the CLI tool that you can get by running the **docker images** command. Run the following commands to tag both base images.

```
docker tag sys-zaas-team-hpvsop-dev-docker-local.artifactory.swg-
devops.com/zaas/hyperpvsop-base-image:1.2.4 docker_base_user/hyperpvsop-base-
image:1.2.4
docker tag sys-zaas-team-hpvsop-dev-docker-local.artifactory.swg-
devops.com/zaas/hyperpvsop-base-ssh-image :1.2.4 docker_base_user/hpvsop-base-ssh-
image:1.2.4
```

4. Run the **docker images** command to check whether the tags for the base images are as expected.

```
REPOSITORY                                                          TAG      IMAGE
ID        CREATED        SIZE
...
docker tag sys-zaas-team-hpvsop-dev-docker-local.artifactory.\
swg-devops.com/zaas/hyperpvsop-base-image                          1.2.4
c6a593192565   3 days ago     1.04GB
docker_base_user/hyperpvsop-base-image                             1.2.4
a6252e869355   3 days ago     1.04GB
docker tag sys-zaas-team-hpvsop-dev-docker-local.artifactory.\
swg-devops.com/zaas/hyperpvsop-base-ssh-image                      1.2.4
c6a593192565   3 days ago     1.04GB
docker_base_user/hyperpvsop-base-ssh-image                         1.2.4
a6252e869355   3 days ago     1.04GB
...
```

5. Push the base images to your remote docker repositories. For example:

```
docker login
docker push docker_base_user/hyperpvsop-base-image:1.2.4
docker push docker_base_user/hyperpvsop-base-ssh-image:1.2.4
```

6. Write down the following information to be used when building your app with the Secure Build container.

   - Your Docker Hub ID account used to register the base images. For example, **docker_base_user**
   - Your Docker Hub ID password. For example, **passw0rd**

Creating a Hyper Protect Virtual Server instance

You can provision a Hyper Protect Virtual Server instance on the Secure Service Container partition by using the **hpvs-op-ssh** base image provided in the IBM Hyper Protect Virtual Servers, and later connect to the instance by using the secure shell. This is useful when you want to debug your application deployed in the Hyper Protect Virtual Server container before publishing the application into your production environment. You can also provision a Hyper Protect Virtual Server instance on the Secure Service Container partition by using the **hpvs-op** base image provided in the IBM Hyper Protect Virtual Servers when you want to deploy your application in the Hyper Protect Virtual Server container for your production environment.

This procedure is intended for users with the role *cloud administrator*.

Before you begin

- Refer to the checklist that you prepared for the Hyper Protect Virtual Server this topic in the topic Planning for the environment.

- Ensure the IBM Hyper Protect Virtual Servers CLI is ready for use. For more information, see Setting up the environment by using the setup script.

- You can use the **hpvs host list** command to verify if a host is already set. When multiple hosts are available, and you want to use a particular host, you can use the **hpvs host set** command. For more information about the **hpvs host** commands, see Commands in IBM Hyper Protect Virtual Servers.
- When the IBM Hyper Protect Virtual Servers is at Version 1.2.2, or later, use the following commands to generate and export the SSH public key as the environment variable for the instance provisioning. Setting a passphrase for the key is not supported.

  ```
  ssh-keygen -t rsa -b 4096 -C "your_email@example.com" -f
  $HOME/hpvs/config/hpvsopbasessh/id_rsa
  ```

  Run the following command to convert the .pub file to base64 format.

  ```
  echo $(cat id_rsa.pub | base64)| tr -d ' ' >>
  /$HOME/hpvs/config/hpvsopbasessh/keys/id_rsa_base64.pub
  export key=$(cat $HOME/hpvs/config/hpvsopbasessh/keys/id_rsa_base64.pub)
  ```

  Note: Applicable only for a virtual server created by using the **hpvs-op-ssh** base image.
- When the IBM Hyper Protect Virtual Servers is at Version 1.2.1, use the following commands to generate and export the SSH public key as the environment variable for the instance provisioning.

  ```
   ssh-keygen -t rsa -b 4096 -C "your_email@example.com" -f
  $HOME/hpvs/config/hpvsopbasessh/id_rsa
   export key=$(cat $HOME/hpvs/config/hpvsopbasessh/keys/id_rsa.pub)
  ```

  Note: Applicable only for a Virtual Server created by using the **hpvs-op-ssh** base image.
- When you create a virtual server, specify a virtual server name that has a maximum of 23 characters, when the version of the Hyper Protect Virtual Servers is 1.2.1, or 1.2.1.1. This restriction does not apply to Hyper Protect Virtual Servers version 1.2.2, or later.

Procedure

Choose one of the options to provision the instance:

- By using the yaml configuration file and **hpvs deploy** command.
- By using the **hpvs vs create** command.

By using the yaml configuration file and hpvs deploy command

This is the recommended option to provision the instance because of it's ease of use and is also an easier method of creating multiple instances quickly.

1. Update the template file **$HOME/hpvs/config/templates/virtualserver.template.yml** based on the networking configuration, quotagroup and resource settings of the Hyper Protect Virtual Server instance if necessary. You must specify the details for the network based on your network configurations. The **vs_hpvsopbasessh.yml** that has the configuration details for the virtual server refers to the corresponding sections of the **virtualserver.template.yml** when you run the **hpvs deploy** command. For example, the **resourcedefinition: ref** value refers to the **resourcedefinitiontemplate** definition in the template file. The **quotagroup: ref** value refers to the **quotagrouptemplates** definition in the template file. The **network: ref** value refers to the **networktemplates** definition in the template file.

   ```
   version: v1
   type: virtualserver-template
   networktemplates:
   - name: external_network
     subnet: "10.20.4.0/22"
     gateway: "10.20.4.1"
     parent: "encf900"
     driver: "macvlan"
   - name: internal_network
   ```

```
  subnet: "192.168.40.0/24"
  gateway: "192.168.40.1"
  parent: "encf900"
  driver: "bridge"
quotagrouptemplates:
# Passthrough quotagroup templates - A quotagroup will be dynamically created 
based
# on the template and attached as single volume mount point to the virtual server.
# Allowed filesystem types for the passthrough type quogagroup are btrfs, ext4, 
xfs
- name: p-small
  size: 20GB
  filesystem : ext4
  passthrough: true
- name: p-medium
  size: 50GB
  filesystem : ext4
  passthrough: true
- name: p-large
  size: 100GB
  filesystem : ext4
  passthrough: true
- name: p-xlarge
  size: 200GB
  filesystem : ext4
  passthrough: true
- name: p-xxlarge
  size: 400GB
  filesystem : ext4
  passthrough: true
# Non passthrough quotagroup definitions - This quotagroups can be shared by
# creating multiple volume mountpoints with the same virtual server or multiple
# virtual server.  A non passthrough quotagroup will be dynamically created based
# on the template and attached as volume mount points to the virtual server.
# Only brtfs filesystem is supported in non passthrough quotagroups
# mount points attached to virtual server can have filesystem btrfs, ext4, xfs
- name: np-small
  size: 20GB
  passthrough: false
- name: np-medium
  size: 50GB
  passthrough: false
- name: np-large
  size: 100GB
  passthrough: false
- name: np-xlarge
  size: 200GB
  passthrough: false
- name: np-xxlarge
  size: 400GB
  passthrough: false
resourcedefinitiontemplates:
- name: default
  cpu: 1
  memory: 1024
- name: small
  cpu: 2
  memory: 2048
- name: large
  cpu: 4
  memory: 4096
- name: xl
  cpu: 8
  memory: 8192
- name: xxl
  cpu: 12
  memory: 12288
```

For more information about the template file for a Hyper Protect Virtual Server instance, see Virtual server template file.

2. Create the configuration yaml file `$HOME/hpvs/config/hpvsopbasessh/demo_server_configfile.yml` for the instance by referring to the example file `$HOME/hpvs/config/hpvsopbasessh/vs_hpvsopbasessh.yml`. The following is an example of a vs_hpvsopbasessh.yml file.

```
version: v1
type: virtualserver
virtualservers:
- name: test-hpvsopbasessh
  host: SSC_LPAR_NAME
  hostname: hpvsopbasessh-container
  repoid: HpvsopBaseSSH
  imagetag: 1.2.4
  imagefile: HpvsopBaseSSH.tar.gz
  imagecache: true
  resourcedefinition:
     ref: small
  environment:
   - key: LOGTARGET
     value: "/dev/console"
   - key: ROOTFS_LOCK
     value: "y"
   - key: ROOT_SSH_KEY
     value: "@/root/hpvs/config/hpvsopbasessh/keys/id_rsa_base64.pub" # provide
ssh key in base64 format
   - key: RUNQ_ROOTDISK
     value: newroot
  networks:
   - ref:  external_network
     ipaddress: 10.20.4.12
  volumes:
   - name: qg_hpvsopbasessh
     ref : np-medium
     mounts:
      - mount_id: newroot
        mountpoint: /newroot
        filesystem: ext4
        size: 10GB
        reset_root: false
      - mount_id: data
        mountpoint: /data
        filesystem: ext4
        size: 10GB
```

Note:

- You must configure the mount point as `/newroot` when you deploy the HpvsopBaseSSH image.
- For creating a virtual server using the `hpvs-op` base image, use the `vs_hpvsopbase.yml` configuration file.
- `resourcedefinition: ref` value refers to the `resourcedefinitiontemplate` definition in the template file.
- `quotagroup: ref` value refers to the `quotagrouptemplates` definition in the template file.
- `network: ref` value refers to the `networktemplates` definition in the template file.
- When you specify `@` at the beginning of a file path, it indicates that the path mentioned is read as a file and the content within the file is assigned as the value.
- For more information about the configurations for a Hyper Protect Virtual Server instance, see Virtual server configuration file.
- In this example, the network definition is for an external network. For more information on other network configurations, see Network requirements for Hyper Protect Virtual Server.

- For more information about quotagroups in IBM Hyper Protect Virtual Servers, see Overview of quotagroups for IBM Hyper Protect Virtual Servers.

  The **imagecache** parameter is supported when the IBM Hyper Protect Virtual Servers is at version 1.2.3. The following parameters specified in the example **vs_hpvsopbasessh.yml** as shown above, are applicable only for IBM Hyper Protect Virtual Servers version 1.2.2, or later.

```
environment:
- key: ROOT_SSH_KEY
  value: "@/root/hpvs/config/hpvsopbasessh/keys/id_rsa_base64.pub" # provide ssh
key in base64 format
- key: RUNQ_ROOTDISK
  value: newroot
volumes:
   ref : np-medium
   mounts:
    - mount_id: newroot
      reset_root: false
    - mount_id: data
      mountpoint: /data
      filesystem: ext4
      size: 10GB
```

The following parameters specified in the example **vs_hpvsopbasessh.yml** as shown above, are applicable for IBM Hyper Protect Virtual Servers versions 1.2.1.1, and 1.2.1.

```
environment:
- key: SSH_PUBLIC_KEY
  value: "@/root/hpvs/config/hpvsopbasessh/keys/id_rsa.pub"
- key: EX_VOLUMES
  value: "/qg_passthrough"
volumes:
   mounts:
   - mountpoint: /volumes:
   - name: qg_hpvsopbasessh
     ref : np-small
   mounts:
   - mount_id: new_qg_hpvsopbasessh
   - name: qg_passthrough
    ref: p-small
   mounts:
   - mountpoint: /qg_passthrough
```

3. Create the instance by using the configurations in the yaml file.

```
hpvs deploy --config $HOME/hpvs/config/hpvsopbasessh/demo_server_configfile.yml
```

If you create a new template file and refer to the this template file from the virtual server configuration file, then you must add the **--template** parameter to specify the absolute path to the template file when running the **hpvs deploy** command.

Note:

- You can use the **hpvs undeploy** command to delete this virtual server. This command is supported in Hyper Protect Virtual Servers version 1.2.2, or later. For more information, see Undeploying virtual servers.
- You can update the resources or configuration of a virtual server after the completion of the deploy operation by using the **-u**, or the **--update** flag of the **hpvs deploy** command. For more information, see Updating virtual servers.

By using the `hpvs vs create` command

1. Load the **hpvs-op-ssh** base image to the Secure Service Container partition.

```
hpvs image load --file=$HOME/hpvs/config/hpvsopbasessh/images/HpvsopBaseSSH.tar.gz
```

Note: For creating a virtual server using the **hpvs-op** base image, use the **HpvsopBase.tar.gz** image from the **$HOME/hpvs/config/hpvsopbase/images/** directory. For creating a virtual server using the **hpvs-op-ssh** base image, use the **HpvsopBaseSSH.tar.gz** image from the **$HOME/hpvs/config/hpvsopbasessh/images/** directory.

2. Create the quotagroup for the instance. The following is an example.

   ```
   hpvs quotagroup create --name qg_hpvsopbasessh --size=40GB
   ```

   Note: If you create a non-passthrough quotagroup for the instance, ensure that you specify a value that is at least 5 GB greater than the size you require for the virtual server.
   For more information about the **hpvs quotagroup** command, see Commands in IBM Hyper Protect Virtual Servers. For more information about quotagroups in IBM Hyper Protect Virtual Servers, see Overview of quotagroups for IBM Hyper Protect Virtual Servers.

3. Create the network for the instance to be connected externally. The following is an example.

   ```
   hpvs network create --name external_net --driver macvlan --parent encf900 --subnet
   10.20.4.0/22 --gateway 10.20.4.1
   ```

   For more information about the **hpvs network** command, see Commands in IBM Hyper Protect Virtual Servers. For more information about the network in IBM Hyper Protect Virtual Servers, see Network requirements for Hyper Protect Virtual Server.

4. Create the network for the instance to be connected within your intranet. The following is an example.

   ```
   hpvs network create --name internal_net --driver bridge --parent encf900 --subnet
   192.168.40.0/24 --gateway 192.168.40.1
   ```

5. Create the instance. The following is an example for IBM Hyper Protect Virtual Servers Version 1.2.2.1, or later.

   ```
   hpvs vs create --name demo_server --repo HpvsopBaseSSH --tag 1.2.4 \
   --cpu 2 --ram 2048 --env=
   {LOGTARGET=/dev/console,ROOTFS_LOCK=y,ROOT_SSH_KEY="$key",RUNQ_ROOTDISK=new} \
   --quotagroup "{quotagroup = qg_hpvsopbasessh, mountid = new,mount = /newroot,
   filesystem = ext4, size = 30GB, reset_root = true}" \
   --network "{name = external_net, ip = 10.20.4.12}" --network "{name =
   internal_net,ip = 192.168.40.23}"
   ```

   The following is an example for IBM Hyper Protect Virtual Servers Version 1.2.2.

   ```
   hpvs vs create --name demo_server --repo HpvsopBaseSSH --tag 1.2.2.-release-
   cedc95a \
   --cpu 2 --ram 2048 --env=
   {LOGTARGET=/dev/console,ROOTFS_LOCK=y,ROOT_SSH_KEY="$key"} \
   --quotagroup "{quotagroup = qg_hpvsopbasessh, mountid = new,mount = /newroot,
   filesystem = btrfs, size = 30GB}" \
   --network "{name = external_net, ip = 10.20.4.12}" --network "{name =
   internal_net,ip = 192.168.40.23}"
   ```

   The following is an example for IBM Hyper Protect Virtual Servers Version 1.2.1.1, or 1.2.1.

   ```
   hpvs vs create --name demo_server --repo HpvsopBaseSSH --tag 1.2.1.1-release-
   481a2e1 \
   --cpu 2 --ram 2048 --env=
   {LOGTARGET=/dev/console,ROOTFS_LOCK=y,SSH_PUBLIC_KEY="$key"} \
   --quotagroup "{quotagroup = qg_hpvsopbasessh, mountid = new,mount = /newroot,
   filesystem = btrfs, size = 30GB}" \
   --network "{name = external_net, ip = 10.20.4.12}" --network "{name =
   internal_net,ip = 192.168.40.23}"
   ```

   Note:

- You must configure the mount point as `/newroot` when you deploy the HpvsopBaseSSH, or HpvsopBase image.
- For creating a virtual server using the `hpvs-op` base image, use the repo ID HpvsopBase, and for the virtual server using the `hpvs-op-ssh` base image, use the repo ID HpvsopBaseSSH.
- In this example, the network definition is for an external network and an internal network. For more information on other network configurations, see Network requirements for Hyper Protect Virtual Server.
- For more information about quotagroups in IBM Hyper Protect Virtual Servers, see Overview of quotagroups for IBM Hyper Protect Virtual Servers.

Next

You can connect to the provisioned Hyper Protect Virtual Server instance by using the secure shell and the respective private key. For example,

```
ssh root@10.20.4.12 -i $HOME/hpvs/config/hpvsopbasessh/id_rsa
```

Note: Applicable only for a virtual server created by using the `hpvs-op-ssh` base image.

Generating the signing keys

You can generate the key pair for signing the repository registration file by using the GnuPG tool.

This procedure is intended for users with the role *cloud administrator* and *app developer or ISV*.

Before you begin

- Check that you have installed the cli tool on your x86 or Linux on IBM Z/LinuxONE (i.e., s390x architecture) management server as a part of the Setting up the environment by using the setup script.

Procedure

1. List the GPG keys by running the following command.

   ```
   gpg --list-keys
   gpg --list-secret-keys
   ```

2. The following commands create a GPG key pair, export the public key `isv_user.pub` and the private key `isv_user.private`. The key pair is protected by using the passphrase `over-the-lazy-dog`. If `isv_user` is listed when you run the `gpg --list-keys` command, then you must use another name.

   ```
   export keyName=isv_user
   export passphrase=over-the-lazy-dog
   cat >isv_definition_keys <<EOF
        %echo Generating registration definition key
        Key-Type: RSA
        Key-Length: 4096
        Subkey-Type: RSA
        Subkey-Length: 4096
        Name-Real: isv_user
        Expire-Date: 0
        Passphrase: over-the-lazy-dog
        # Do a commit here, so that we can later print "done" :-)
        %commit
        %echo done
   EOF
   ```

```
gpg -a --batch --generate-key isv_definition_keys
gpg --armor --pinentry-mode=loopback --passphrase  ${passphrase} --export-secret-
keys ${keyName} > ${keyName}.private
gpg --armor --export ${keyName} > ${keyName}.pub
```

The "export keyName=isv_user" and "Name-Real: isv_user" must be unique. You cannot use the same keys to sign multiple images. You should not have multiple keys with same username, also you should not have multiple images singed with same key in a Secure Service Container.

3. Copy the generated key pair **`isv_user.pub`** and **`isv_user.private`** to the **`<$HOME/hpvs>/config`** directory on your x86 or Linux on IBM Z/LinuxONE (i.e., s390x architecture) management server.

Enabling ports

When you are using IBM Hyper Protect Virtual Servers version 1.2.2, or later, before you build a docker image by using the Hyper Protect base images, you must open the required ports for your application.

The following information shows an example of how you can open the ports before building the docker image.

```
#Here is the example on how you can re-write the iptable rules and open the required
ports
*filter
:INPUT DROP [4:180]
:FORWARD DROP [0:0]
:OUTPUT DROP [0:0]
-A INPUT -i lo -j ACCEPT
-A INPUT -s 127.0.0.0/8 -j DROP
-A INPUT -p tcp -m state --state ESTABLISHED -j ACCEPT
-A INPUT -p udp -m state --state ESTABLISHED -j ACCEPT
-A INPUT -p icmp -m state --state ESTABLISHED -j ACCEPT
#
-A OUTPUT -o lo -j ACCEPT
-A OUTPUT -p tcp -m state --state NEW,ESTABLISHED -j ACCEPT
-A OUTPUT -p udp -m state --state NEW,ESTABLISHED -j ACCEPT
-A OUTPUT -p icmp -m state --state NEW,ESTABLISHED -j ACCEPT
#
# Since by default all ports are blocked on HPVS Base image you could open the required
ports by doing the following.
# This is an example where you can open port 22 which is required for SSH access.
-A INPUT -p tcp -m tcp --dport 22 -j ACCEPT
COMMIT
```

Copy this iptables.conf in your Docker build path and add an entry in the Dockerfile to use the iptables.conf, as shown below.

```
COPY iptables.conf /etc/iptables/
```

If you are using base images as parent images, then you must initialize the **`systemd`** service by including following line in your Dockerfile:

```
CMD ["/sbin/init"]
```

After the Dockerfile is updated, you can build your docker image by using Secure Build. For more information, see Building your application with the Secure Build virtual server.

Building your application with the Secure Build virtual server

You can use the Secure Build virtual server to build your source code stored in the GitHub repository, deploy it into the IBM Hyper Protect Virtual Servers as a Hyper Protect Virtual Server instance, and publish the built image to the remote Docker repository.

During the Secure Build process, the Secure Build virtual server performs the following actions:

- Retrieve the source code from your GitHub repository, therefore your private key to access the GitHub repository is required for authentication.
- Pull the **`hpvsop-base`** or **`hpvsop-base-ssh`** base images that you choose in the Docker file from the remote Docker registry, to host your application in a Hyper Protect Virtual Server instance on the Secure Service Container partition, which uses the Docker credential stored by using the **`hpvs registry add`** command.
- Builds the image and signs the tag of the image.
- Push the built image to the remote Docker repository such as DockerHub or IBM Container Registry, which uses credentials that you added during the **`hpvs registry add`** command. It also signs the repository registration file with your own key pair so that only authorized repository registration file is allowed into Secure Service Container partition. Also it will encrypt repository registration file using IBM key.
- Optional: Archive the Secure Build manifest file for your applications in the IBM Cloud Object Storage service for audit purpose.

If you want other developers or ISVs to build their image based on your published image in the IBM Hyper Protect Virtual Servers, you can also create a dedicated user ID for them to pull your image.

When you have large files in your repository, or a lot of binaries, it is recommended to use Git LFS. You can use Git LFS when the IBM Hyper Protect Virtual Servers are at Version 1.2.2. To ensure secure communication, it is recommended that you configure the Git LFS server over HTTPS protocol only. Git Large File Storage (LFS) helps you work more efficiently with large files and binary files in your repository. An update of a binary file is seen by Git as a complete file change, rather than for example a plain text file, where only the differences to the file are stored. If you have frequent changes to binary files, then your Git repository will grow in size. After a certain amount of time, Git commands will become slower because of the growing size of your repository.

This procedure is intended for users with the role *cloud administrator* and *app developer or ISV*.

- Cloud administrator creates the Secure Build virtual server and register the repository for the App developer or ISV.
- App developer or ISV can then use the Secure Build virtual server to build and deploy applications from a remote GitHub repository.

Before you begin

- Refer to the checklist that you prepared for the Secure Build virtual server in this topic Planning for the environment.
- Ensure that you have all the user IDs and passwords to pull the base images, push the built images, and pull the built images from the remote Docker registry server.
- Check that you have installed the cli tool on your x86 or Linux on IBM Z/LinuxONE (i.e., s390x architecture) management server as a part of the Setting up the environment by using the setup script.
- When you create a virtual server, specify a virtual server name that has a maximum of 23 characters, when the version of the Hyper Protect Virtual Servers is 1.2.1, or 1.2.1.1. This restriction does not apply to Hyper Protect Virtual Servers version 1.2.2, or later.
- When you are using IBM Hyper Protect Virtual Servers version 1.2.2, or later, before you build a docker image by using the Hyper Protect base images, you must open the required ports for your application. For more information, see Enabling ports.

Procedure

On your x86 or Linux on IBM Z/LinuxONE (i.e., s390x architecture) management server, complete the following steps with root user authority.

1. Creating the certificate and key to securely communicate with secure build server.
2. Choose one of the following options to create a Secure Build virtual server.
   - Create virtual server by using the yaml configuration file and `hpvs deploy` command.
   - Create virtual server by using the `hpvs image` and `hpvs vs create` command.
3. Generating the signing keys.
4. Building the application by using the Secure Build.
5. Choose one of the following options to deploy the application.
   - Deploy application by using the yaml configuration file and `hpvs deploy` command.
   - Deploy application by using the `hpvs vs create` command.

Creating the certificate and key to securely communicate with secure build server

1. Run the following command.

   ```
   cd $HOME/hpvs/config/securebuild/keys
   ```

2. Create the certificate and key to securely communicate with secure build server.

   ```
   openssl req -newkey rsa:2048 \
   -new -nodes -x509 \
   -days 3650 \
   -out sbs.cert \
   -keyout sbs.key \
   -subj "/C=GB/O=IBM/CN=johndoe.example.com"
   ```

   Note: If you see errors like `random number generator:RAND_load_file:Cannot open file`, then run the following commands.

   ```
   openssl rand -out $HOME/.rnd -hex 256
   ```

3. Run the following command to change the certificate to base64 encoding.

   ```
   echo $(cat sbs.cert | base64) | tr -d ' ' >> sbs_base64.cert
   ```

Create virtual server by using the yaml configuration file and `hpvs deploy` command

This is the recommended option to provision the instance because of it's ease of use and is also an easier method of creating multiple instances quickly.

1. Update the template file `$HOME/hpvs/config/templates/virtualserver.template.yml` based on the networking configuration, quotagroup and resource settings of the Hyper Protect Virtual Server instance if necessary. The `vs_securebuild.yml` that has the configuration details for the virtual server refers to the corresponding sections of the `virtualserver.template.yml` when you run the `hpvs deploy` command. For example, the `resourcedefinition: ref` value refers to the `resourcedefinitiontemplate` definition in the template file. The `network: ref` value refers to the `networktemplates` definition in the template file.

   ```
   version: v1
   type: virtualserver-template
   networktemplates:
   - name: external_network
     subnet: "10.20.4.0/22"
   ```

```
  gateway: "10.20.4.1"
  parent: "encf900"
  driver: "macvlan"
- name: internal_network
  subnet: "192.168.40.0/24"
  gateway: "192.168.40.1"
  parent: "encf900"
  driver: "bridge"
quotagrouptemplates:
# Passthrough quotagroup templates - A quotagroup will be dynamically created 
based
# on the template and attached as single volume mount point to the virtual server.
# Allowed filesystem types for the passthrough type quogagroup are btrfs, ext4, 
xfs
- name: p-small
  size: 20GB
  filesystem : ext4
  passthrough: true
- name: p-medium
  size: 50GB
  filesystem : ext4
  passthrough: true
- name: p-large
  size: 100GB
  filesystem : ext4
  passthrough: true
- name: p-xlarge
  size: 200GB
  filesystem : ext4
  passthrough: true
- name: p-xxlarge
  size: 400GB
  filesystem : ext4
  passthrough: true
# Non passthrough quotagroup definitions - This quotagroups can be shared by
# creating multiple volume mountpoints with the same virtual server or multiple
# virtual server.  A non passthrough quotagroup will be dynamically created based
# on the template and attached as volume mount points to the virtual server.
# Only brtfs filesystem is supported in non passthrough quotagroups
# mount points attached to virtual server can have filesystem btrfs, ext4, xfs
- name: np-small
  size: 20GB
  passthrough: false
- name: np-medium
  size: 50GB
  passthrough: false
- name: np-large
  size: 100GB
  passthrough: false
- name: np-xlarge
  size: 200GB
  passthrough: false
- name: np-xxlarge
  size: 400GB
  passthrough: false
resourcedefinitiontemplates:
- name: default
  cpu: 1
  memory: 1024
- name: small
  cpu: 2
  memory: 2048
- name: large
  cpu: 4
  memory: 4096
- name: xl
  cpu: 8
  memory: 8192
- name: xxl
  cpu: 12
  memory: 12288
```

For more information about the template file for a Hyper Protect Virtual Server instance, see Virtual server template file.

2. Create the configuration yaml file $HOME/hpvs/config/securebuild/demo_securebuild.yml for the instance by referring to the example file $HOME/hpvs/config/securebuild/vs_securebuild.yml. The following is an example of a **`vs_securebuild.yml`** file. In this example, the network definition is for an external network. For more information on other network configurations, see Network requirements for Hyper Protect Virtual Server. For more information about quotagroups in IBM Hyper Protect Virtual Servers, see Overview of quotagroups for IBM Hyper Protect Virtual Servers.

```
version: v1
type: virtualserver
virtualservers:
- name: securebuildserver
  host: SSC_LPAR_NAME
  repoid: SecureDockerBuild
  imagetag: 1.2.4
  imagefile: SecureDockerBuild.tar.gz
  imagecache: true
  resourcedefinition:
     ref: small
  environment:
   - key: ROOTFS_LOCK
     value: "y"
   - key: CLIENT_CRT
     value: "@/root/hpvs/config/securebuild/keys/sbs_base64.cert" # provide
certificate file in base64 format
   - key: RUNQ_ROOTDISK
     value: newroot
  networks:
   - ref:  external_network
     ipaddress: 10.20.4.67
  volumes:
   - name: securebuild_qg
     ref: np-medium
     mounts:
      - mountpoint: /data
        filesystem: ext4
        size: 16GB
        mount_id: data
      - mountpoint: /docker
        filesystem: ext4
        size: 16GB
        mount_id: docker
      - mountpoint: /newroot
        filesystem: ext4
        size: 10GB
        mount_id: newroot
        reset_root: false
```

For more information about the config file for a Hyper Protect Virtual Server instance, see Virtual Server Configuration file. **Note**: You must configure the mount point as **`/newroot`** when you deploy an image that is based on the base image.

The **`imagecache`** parameter is supported when the IBM Hyper Protect Virtual Servers is at version 1.2.3. The following parameters specified in the example **`vs_securebuild.yml`** as shown above, are applicable only for IBM Hyper Protect Virtual Servers version 1.2.2, or later.

```
environment:
 - key: CLIENT_CRT
```

```
   value: "@/root/hpvs/config/securebuild/keys/sbs_base64.cert" # provide
certificate file in base64 format
 - key: RUNQ_ROOTDISK
   value: newroot
```

The following parameters specified in the example **vs_securebuild.yml** as shown above, are applicable for IBM Hyper Protect Virtual Servers versions 1.2.1.1, and 1.2.1.

```
environment:
 - key: CLIENT_CRT
   value: "@/root/hpvs/config/securebuild/keys/sbs.cert"
 - key: EX_VOLUMES
   value: "/docker,/data"
```

3. Create the instance by using the configurations in the yaml file.

   ```
   hpvs deploy --config $HOME/hpvs/config/securebuild/demo_securebuild.yml
   ```

   Note:

   - You can use the **hpvs undeploy** command to delete this virtual server. This command is supported in Hyper Protect Virtual Servers version 1.2.2, or later. For more information, see Undeploying virtual servers.
   - You can update the resources or configuration of a virtual server after the completion of the deploy operation by using the **-u**, or the **--update** flag of the **hpvs deploy** command. For more information, see Updating virtual servers.

Create virtual server by using the `hpvs image` and `hpvs vs create` commands

1. Upload the Secure Build image **SecureDockerBuild.tar.gz** to the Secure Service Container partition.

   ```
   hpvs image load --
   file=$HOME/hpvs/config/securebuild/images/SecureDockerBuild.tar.gz
   ```

2. Export the certificate as an environment variable.

   ```
   export cert=$(echo $(cat ~/hpvs/config/securebuild/keys/sbs.cert | base64) | tr -d
   ' ')
   ```

3. Create the quotagroup for the Secure Build virtual server.

   ```
   hpvs quotagroup create --name securebuild_qg --size=50GB
   ```

   Note: If you create a non-passthrough quotagroup for the Secure Build virtual server, it is recommended that you ensure that 20% of disk space is always available in order to address any I/O errors.
   For more information about the **hpvs quotagroup** command, see Commands in IBM Hyper Protect Virtual Servers. For more information about quotagroups in IBM Hyper Protect Virtual Servers, see Overview of quotagroups for IBM Hyper Protect Virtual Servers.

4. Create the external network for the Secure Build virtual server.

   ```
   hpvs network create --name external_net --driver macvlan --parent encf900 --subnet
   10.20.4.0/22 --gateway 10.20.4.1
   ```

   For more information about the **hpvs network** command, see Commands in IBM Hyper Protect Virtual Servers. For more information about the network in IBM Hyper Protect Virtual Servers, see Network requirements for Hyper Protect Virtual Server.

5. Create the Secure Build virtual server. The following is an example for IBM Hyper Protect Virtual Servers Version 1.2.2.1, or later.

   ```
   hpvs vs create --name securebuildserver --repo SecureDockerBuild \
   --tag 1.2.4 --cpu 2 --ram 2048 \
   --env=
   ```

```
{EX_VOLUMES="/docker,/data",ROOTFS_LOCK=y,CLIENT_CRT=$cert,RUNQ_ROOTDISK=new} \
--quotagroup "{quotagroup = securebuild_qg, mountid = new, mount = /newroot, 
filesystem = ext4, size = 4GB,  reset_root=true}" \
--quotagroup "{quotagroup = securebuild_qg, mountid = data, mount = /data, 
filesystem = ext4, size = 4GB}" \
--quotagroup "{quotagroup = securebuild_qg, mountid = docker, mount = /docker, 
filesystem = ext4, size = 16GB}" \
--network "{name = external_net,ip = 10.20.4.12}"
```

The following is an example for IBM Hyper Protect Virtual Servers Version 1.2.1.1, or 1.2.1.

```
hpvs vs create --name securebuildserver --repo SecureDockerBuild \
--tag  1.2.1.1-release-bf10b8e  --cpu 2 --ram 2048 \
--quotagroup "{quotagroup = securebuild_qg, mountid = new, mount = /newroot, 
filesystem = ext4, size = 4GB}" \
--quotagroup "{quotagroup = securebuild_qg, mountid = data, mount = /data, 
filesystem = ext4, size = 4GB}" \
--quotagroup "{quotagroup = securebuild_qg, mountid = docker, mount = /docker, 
filesystem = ext4, size = 16GB}" \
--env={EX_VOLUMES="/docker,/data",ROOTFS_LOCK=y,CLIENT_CRT=$cert} \
--network "{name = external_net,ip = 10.20.4.12}"
```

where

- `/newroot` storage on the quotagroup `securebuild_qg` is for the Secure Build server image. You must configure the mount point as `/newroot` when you deploy an image that is based on the base image.
- `/data` storage on the quotagroup `securebuild_qg` is for the log configuration date.
- `/docker` storage on the qutogroup `securebuild_qg` is for the applications to be built on the Secure Build server.
- `CLINT_CRT=$cert` is to ensure only authorized REST API calls from the Secure Build virtual server can be accepted by the hosting appliance in order to build Hyper Protect Virtual Server instances.
- For a full list of supported parameters and options of the `hpvs` command, see Commands in IBM Hyper Protect Virtual Servers.
- In this example, the network definition is for an external network. For more information on other network configurations, see Network requirements for Hyper Protect Virtual Server.
- For more information about quotagroups in IBM Hyper Protect Virtual Servers, see Overview of quotagroups for IBM Hyper Protect Virtual Servers.

Generating the signing keys

To generate the signing keys, follow the instructions listed in Generating the signing keys.

Building the application by using the Secure Build

1. Create the Secure Build configuration file. You can use the `$HOME/hpvs/config/securebuild/secure_build.yml.example` example file as a reference when updating the file.

```
secure_build_workers:
  sbs:
     url: '<url of the secure build service. e.g- https://10.20.4.67>'
     port: '443'
     cert_path: '<complete path of certificate.  e.g-
/root/hpvs/config/securebuild/keys/sbs_cert>'
     key_path: '<complete path of key.  e.g-
/root/hpvs/config/securebuild/keys/sbs_key>'
  regfile:
     id: '<Enter Id. It could be any name>'
  github:
     url: '<git hub url. e.g- ssh://git@github.com:<port>/MyOrg/my-docker-
app.git>'
     branch: 'master'
     ssh_private_key_path: '<complete path of key github private key. e.g - 
/root/git_key>'
     recurse_submodules: 'False'
     dockerfile_path: './Dockerfile'
     docker_build_path: '<Enter the path to the subdirectory within the Github 
project to be used as the build context for the Docker build>'
  docker:
     push_server: '<get this from hpvs registry list. e.g - docker_push>'
     base_server: '<get this from hpvs registry list. e.g - docker_base>'
     pull_server: '<get this from hpvs registry list. e.g - docker_pull>'
     repo: 'docker_user_name/docker_image_name'
     image_tag_prefix: 'latest'
     content_trust_base: 'True'
  manifest_cos:
     bucket_name: '<Enter the bucket name on the S3 object store where manifest 
files will be transferred to after each build>'
     api_key: '<Enter the API key used to authenticate with the S3 object store>'
     resource_crn: '<Enter the resource instance ID for the S3 object store>'
     auth_endpoint: '<Enter the authentication endpoint for the S3 object store>' 
(For example: `https://iam.cloud.ibm.com/identity/token`)
     endpoint: '<Enter the endpoint for the S3 object store>' (For 
example:'https://s3.us-east.cloud-object-storage.appdomain.cloud')
  # Add all allowlist environment variables that are required in your virtual 
server. If you try to create a virtual server with environment variables that are 
not added to the allowlist, then creating the virtual server fails. This is an 
optional parameter and if you do not have any environment variable for the virtual 
server, you can comment this parameter.
  env:
     allowlist: [KEY1,KEY2]
  build:
     args:
       <ARG1>: '<value1>'
       <ARG2>: '<value2>'
  signing_key:
     private_key_path: '<Enter the absolute private key path. For example, 
/root/hpvs/config/securebuild/keys/isv_user.private'
     public_key_path: '<Enter the absolute public key path. For example, 
/root/hpvs/config/securebuild/keys/isv_user.pub'

  # Add linux capabilities to hyper protect virtual server. List of Linux 
capabilities
  # are available here https://man7.org/linux/man-pages/man7/capabilities.7.html.
  # All the capabilities listed are supported except "CAP_PERFMON", "CAP_BPF", and 
CAP_CHECKPOINT_RESTORE".
  # While adding capabilities remove the prefix "CAP".
  # For example CAP_AUDIT_CONTROL will be AUDIT_CONTROL

  cap_add: [] # eg: ["NET_ADMIN","NET_RAW"], or ["ALL"]
```

Note:

- Starting with IBM Hyper Protect Virtual Servers version 1.2.4, the term "whitelist" is replaced with "allowlist". For IBM Hyper Protect Virtual Servers versions earlier than 1.2.4, you must use "whitelist" instead of "allowlist".
- If the base image in Docker file is not signed then the `base_server` parameter is not required and `content_trust_base` must be `False`.
- If you want to specify a non-default SSH port, then you can add the value of the port that you want to use in the `github url` parameter as shown above in the `secure_build.yml.example` file , when the IBM Hyper Protect Virtual Servers is at version 1.2.3. When no port is specified, the `github url` can be specified as `"git@github.com:MyOrg/my-docker-app.git"`.
- The `cap_add: []` parameter is applicable for IBM Hyper Protect Virtual Servers version 1.2.3, or later. To enable all privileges' you can use `cap_add:["ALL"]`, but as a good security practice, provide the least possible privileges' to your virtual server.

- Build parameters (**build args**) are used to give additional information as might be required for the specific application that you want to run on the virtual server.
- You must provide valid GitHub URL and also ensure that you use a **.git** extension when specifying the URL.
- It is recommended that you choose an endpoint URL that is located in the same region as your service or application, and specify this URL as the value for the **endpoint** parameter in the **manifest_cos** section of the secure_build.yml file. For more information about identifying endpoint URL, see Cloud Object Storage.

For a full list of supported parameters in the configuration file, see Secure Build configuration file.

To configure a Cloud Object Storage service to archive the application manifest files of your applications built by your Secure Build container, ensure that you have the following information about your IBM Cloud Object Storage at hand.

- The API Key to the cloud object storage service
- The object storage bucket to store the manifest
- The resource instance name of the cloud object storage service
- The authentication endpoint for the cloud object storage service
- The endpoint for the cloud object storage service

2. Build your application and upload the application manifest file to the cloud object storage by using Secure Build. You can choose either of the following options:

   - Use one command to perform all the Secure Build actions including initialization, build, and generating the encrypted repository registration file. This option is recommended if you are building the application by using the Secure Build for the first time.

     ```
     hpvs sb init --config $HOME/hpvs/config/securebuild/secure_build.yml.example
     --out $HOME/hpvs/config/MyDockerAppImageRegfile.enc --build
     ```

   - Use individual commands to perform each step of building the application by using the Secure Build virtual server. This option is recommended if you plan to build the application by using the Secure Build multiple times. In this scenario, you can run the **hpvs sb build** command for subsequent builds.

     ```
     hpvs sb build --config $HOME/hpvs/config/securebuild/secure_build.yml.example
     hpvs sb regfile --config
     $HOME/hpvs/config/securebuild/secure_build.yml.example --out
     $HOME/hpvs/config/MyDockerAppImageRegfile.enc
     ```

     You can log in to your cloud account and check the application manifest file has been transferred to its bucket in your Cloud Object Storage service after the commands complete execution.

   You can use the **hpvs sb manifest** command to download the manifest file of the secure build.

   ```
   hpvs sb manifest --config $HOME/hpvs/config/securebuild/secure_build.yml.example -
   -name <build_name>
   ```

   where you can get the **<build_name>** by using the **hpvs sb status** after the build completes. When the command execution completes, the manifest file is downloaded to the current directory from which the **hpvs sb manifest** command was run from. To verify the signature of the manifest file, see instructions in Verifying the signature of the manifest file.

   Note:

   - If the **hpvs sb init**, **hpvs sb build**, or the **hpvs sb regfile** commands fails for any reason, for example you specified incorrect parameters, then you can use the **hpvs sb update** command to update the configuration of the Secure Build configuration and rerun the commands with the updated configuration. The **regfile[id]** and **docker[repo]** parameters cannot be updated by using this command.
   - You can use the **hpvs sb log** command to view the run time logs of the secure build process, or for troubleshooting or debugging. The logs are available when you run the **hpvs sb init**, **hpvs sb build**, or the **hpvs sb regfile** commands.

- You can use the **`hpvs sb status`** command to view the status of the last secure build process.
- You can use the **`hpvs sb clean`** command to clean the logs of the secure build process. Build artifacts from the earlier builds are deleted.
- For more information about the secure build commands, see hpvs sb.

Deploying the application by using the yaml configuration file and `hpvs deploy` command

1. Create the configuration yaml file $HOME/hpvs/config/demo_app.yml for the instance by referring to the example file $HOME/hpvs/config/vs_regfiledeployexample.yml. The following is an example of a **`vs_regfiledeployexample.yml`** file.

```
version: v1
type: virtualserver
virtualservers:
- name: testcontainer
  host: SSC_LPAR_NAME
  repoid: MyDockerRepo
  imagetag: latest
  reporegfile: /HOME/hpvs/config/MyDockerAppImageRegfile.enc
  imagecache: true
  resourcedefinition:
     ref: small
  networks:
   - ref:  external_network
     ipaddress: 10.20.4.61
  volumes:
   - name: myquotagroup
     ref : np-medium
     mounts:
      - mount_id: new
        mountpoint: /new
        filesystem: ext4
        size: 10GB
```

   The **`imagecache`** parameter is supported when the IBM Hyper Protect Virtual Servers is at version 1.2.3. In this example, the network definition is for an external network. For more information on other network configurations, see Network requirements for Hyper Protect Virtual Server.
   For more information about quotagroups in IBM Hyper Protect Virtual Servers, see Overview of quotagroups for IBM Hyper Protect Virtual Servers.
   For more information about the config file for a Hyper Protect Virtual Server instance, see Virtual server Configuration file.

2. Deploy the image by using the configurations in the yaml file.

```
hpvs deploy --config $HOME/hpvs/config/demo_app.yml
```

   Note:

   - You can use the **`hpvs undeploy`** command to delete this virtual server. This command is supported in Hyper Protect Virtual Servers version 1.2.2, or later. For more information, see Undeploying virtual servers.
   - You can update the resources or configuration of a virtual server after the completion of the deploy operation by using the **`-u`**, or the **`--update`** flag of the **`hpvs deploy`** command. For more information, see Updating virtual servers.

Deploying the application by using the `hpvs vs create` command

1. Register the repository on the Secure Service Container partition for the application image by using the generated repository registration file.

```
hpvs repository register --pgp=$HOME/hpvs/config/MyDockerAppImageRegfile.enc --
id=MyDockerRepo
```

2. Create the quotagroup of the application image on the Secure Service Container partition.

   ```
   hpvs quotagroup create --name myquotagroup --size=30GB
   ```

   Note: If you create a non-passthrough quotagroup for the Secure Build virtual server, it is recommended that you ensure that 20% of disk space is always available in order to address any I/O errors.

3. Deploy the application image into the IBM Hyper Protect Virtual Servers as a Hyper Protect Virtual Server instance.

   ```
   hpvs vs create --name testcontainer --repo MyDockerRepo --tag latest --cpu 2 --ram 
   2048 --env={env_var1=value1,env_var2=value2} --quotagroup "{quotagroup = 
   myquotagroup, mountid = new, mount = /newroot, filesystem = btrfs, size = 25GB}" -
   -network "{name = external_net,ip = 10.20.4.73}"
   ```

Verifying the signature of the manifest file

You can verify if the manifest file has been signed by using the public key.

This procedure is intended for users with the role *cloud administrator*.

Procedure

Complete the following steps to verify if the manifest file has been signed by using the public key.

To get the manifest file, complete the following steps.

1. You can get the **<BUILD_NAME>** by using the **hpvs sb status** command after the build completes.

   ```
   hpvs sb status --config $HOME/hpvs/config/securebuild/secure_build.yml.example
   ```

   Now you can use the **hpvs sb manifest** command to download the manifest file of the secure build.

   ```
   hpvs sb manifest --config $HOME/hpvs/config/securebuild/secure_build.yml.example -
   -name "${BUILD_NAME}"
   ```

2. When the command execution completes, the manifest file is downloaded to the current directory from which the **hpvs sb manifest** command was run from as **${MANIFEST}.sig.tbz**. Extract the compressed tar file by using the following command.

   ```
   tar -xjf 
   $HOME/hpvs/config/securebuild/secure_build.yml.example/manifest/manifest.${BUILD_N
   AME}.sig.tbz
   ```

To verify the signature, complete the following steps.

1. You can retrieve the pubkey using the **hpvs sb pubkey** command.

   ```
   hpvs sb pubkey --config $HOME/hpvs/config/securebuild/secure_build.yml.example --
   name <build_name>
   ```

   When the command execution completes, the pubkey is downloaded to the current directory from which the **hpvs sb pubkey** command was run from as **${PUBKEY}.pem**.

2. Convert the hex signature to binary by running the following command.

   ```
   cat "${MANIFEST}.sig" | xxd -r -p > "${MANIFEST}.sig.bin"
   ```

   For example:

```
cat manifest.docker.io.dockeruser.securebuildcontainer32.latest-a5714c9.2020-07-
01_09-21-04.706478.sig | wc --bytes 512
```

```
cat manifest.docker.io.dockeruser.securebuildcontainer32.latest-a5714c9.2020-07-
01_09-21-04.706478.sig | xxd -r -p >
manifest.docker.io.dockeruser.securebuildcontainer32.latest-a5714c9.2020-07-01_09-
21-04.706478.sig.bin
```

```
cat manifest.docker.io.dockeruser.securebuildcontainer32.latest-a5714c9.2020-07-
01_09-21-04.706478.sig.bin | wc --bytes 256
```

3. SHA256 hash the .tbz file before you provide it as an input for verifying by running the following command.

```
openssl dgst -sha256 -binary -out "${MANIFEST}.tbz.sha256" "${MANIFEST}.tbz"
```

For example

```
openssl dgst -sha256 -binary -out
manifest.docker.io.dockeruser.securebuildcontainer32.latest-a5714c9.2020-07-01_09-
21-04.706478.tbz.sha256
manifest.docker.io.dockeruser.securebuildcontainer32.latest-a5714c9.2020-07-01_09-
21-04.706478.tbz
```

4. Use the openssl verify command.

```
openssl dgst -sha256 -verify "${MAN_PUBKEY}" -signature "${MANIFEST}.sig.bin"
"${MANIFEST}.tbz.sha256"
```

For example

```
openssl dgst -sha256 -verify docker.io.dockeruser.securebuildcontainer32.latest-
a5714c9.2020-07-01_09-21-04.706478-public.pem -signature
manifest.docker.io.dockeruser.securebuildcontainer32.latest-a5714c9.2020-07-01_09-
21-04.706478.sig.bin manifest.docker.io.dockeruser.securebuildcontainer32.latest-
a5714c9.2020-07-01_09-21-04.706478.tbz.sha256
```

Rolling keys in a Secure Build container

You can roll the keys used by a Secure Build container when it is required as per your security policies, or when the private keys get compromised by malicious attacks.

For the keys that might be impacted or rolled, see List of keys used during the Secure Build.

In order to roll such keys, you must create a new Secure Build container and an updated repository registration file for your applications images to be created.

This procedure is intended for users with the role *Application developer or ISV*.

Before you begin

- Ensure you install the latest IBM Hyper Protect Virtual Servers CLI tool on your x86 or Linux on IBM Z/LinuxONE (i.e., s390x architecture) management server.

Procedure

Complete the following procedure on the management server with root user authority.

1. Create and initialize a new Secure Build container by following the instructions from the topic Building your application with the Secure Build virtual server.

2. Contact the image repository host (DockerHub or IBM Cloud Container Registry) to reset the repository state. See the **Lost keys** section on the Manage keys for content trust.

3. Update your application to use the latest Secure Build container, and then build the application by following the instructions from the topic Building your application with the Secure Build virtual server. Note that if the remote repository has been reset properly, the new images will be in the repository signed with the newly generated private key.

4. Update the repository on the Secure Service Container partition by using the new signing key. For more information, see Refreshing registered repositories with a new signing key pair.

Deploying your applications securely

You can deploy your own Linux-based container image as a Hyper Protect Virtual Server on the IBM Hyper Protect Virtual Servers. This feature is also known as Bring Your Own Image (BYOI).

This procedure is intended for users with the role *cloud administrator* and *app developer or ISV*.

- App developer or ISV prepares Linux-based container image for s390x architecture.
- Cloud administrator registers the repository for the App developer or ISV.
- App developer or ISV can later deploy the images into the IBM Hyper Protect Virtual Servers.

Before you begin

- Refer to the checklist that you prepared for the Hyper Protect Virtual Server on this topic Planning for the environment.

- Ensure your Linux-based container image are built for the IBM LinuxONE and IBM Z platform (s390x architecture), and available either on DockerHub or IBM Container Registry.

- Ensure your Linux-based container images are signed using Docker Content Trust. If not signed using Docker Content Trust , follow the steps listed in Sign your image by using Docker Content Trust.

- When you create a virtual server, specify a virtual server name that has a maximum of 23 characters, when the version of the IBM Hyper Protect Virtual Servers is 1.2.1, or 1.2.1.1. This restriction does not apply to Hyper Protect Virtual Servers version 1.2.2, or later.

- When you are using IBM Hyper Protect Virtual Servers version 1.2.4, or later, to deploy your own Linux-based container image as a Hyper Protect Virtual Server whose repository is not registered, you must regenerate the repository registration file. You can use the `hpvs regfile create` command to regenerate the repository registration file.

Procedure

Complete the following steps with root user authority.

1. Sign your image by using Docker Content Trust.
2. Adding the registry.
3. Generating the signing keys.
4. Preparing the configuration.
5. Deploy your image.

Sign your image by using Docker Content Trust

1. Run the following command to load the image from the DockerHub onto your management server.

   ```
   docker image pull <your_docker_id>/<result_image_name>:<tag>
   ```

2. Enable Docker Content Trust (DCT), specify the server for the Docker Content Trust service by running the following command.

   ```
   export DOCKER_CONTENT_TRUST=1
   export DOCKER_CONTENT_TRUST_SERVER=https://notary.docker.io
   ```

3. Re-tag your docker images by running the following command.

   ```
   docker tag <your_docker_id>/<result_image_name>:<tag>
   <your_docker_id>/<result_image_name>:<new-tag>
   ```

4. Push tagged images to the DockerHub by running the following command.

   ```
   docker push <your_docker_id>/<result_image_name>:<new-tag>
   ```

   Enter your root passphrase and repository passphrase when you are prompted to. The generated public key is stored in `~/.docker/trust/tuf/docker.io/<your_docker_id>/<result_image_name>/metadata/root.json/`

   The image will be pushed to a remote Docker repository with DCT enabled.

Adding the registry

1. Verify whether you already have a registry by running the following command.

   ```
   hpvs registry list
   ```

   If there are no registries displayed, then add a registry by running the following command.

   ```
   hpvs registry add --name registry_name --user <username> --dct
   https://notary.docker.io --url docker.io
   ```

   Where
   name - Specify a name for your registry.
   user - Docker registry username.

Generating the signing keys

To generate the signing keys, follow the instructions listed in the topic Generating the signing keys.

Preparing the configuration

1. Create the configuration yaml `secure_create.yaml` file so that the repository registration file for your image can be generated. You can use the `$HOME/hpvs/config/securebuild/secure_create.yaml.example` example file as a reference when updating the file.

   ```
   repository_registration:
      docker:
         repo: 'docker_user_name/docker_image_name'
         pull_server: '<get this from hpvs registry list. e.g - docker_pull>'
         # this root.json you will get after once you will push image to DockerHub
   using Docker Content Trust
         # optional - if you signed your image from the same management server that
   you are running the commands from, then this parameter is optional.
         # Otherwise, you must copy the
   '/root/.docker/trust/tuf/docker.io/docker_user_name/docker_image_name/metadata/roo
   t.json' to the machine you are running the commands from and provide the complete
   path to the root.
   ```

```
      content_trust_json_file_path: 
'/root/.docker/trust/tuf/docker.io/docker_user_name/docker_image_name/metadata/roo
t.json'
   # Add all allowlist environment variables that are required in your virtual 
server. You cannot create a virtual server if you try to create a virtual server 
with environment variables that are not added to the allowlist. This is an 
optional parameter and if you do not have any environment variable for the virtual 
server, you can comment this parameter.
   env:
      allowlist: ["env_var1","env_var2"]
   signing_key:
   # complete path of signing private key
      private_key_path: '/root/hpvs/config/isv_user.private'
   # complete path of signing public key
      public_key_path: '/root/hpvs/config/isv_user.pub'

   # Add linux capabilities to hyper protect virtual server. List of linux 
capabilities
   # are available here https://man7.org/linux/man-pages/man7/capabilities.7.html.
   # All the capabilities listed are supported except "CAP_PERFMON", "CAP_BPF", 
and CAP_CHECKPOINT_RESTORE".
   # While adding capabilities remove the prefix "CAP".
   # For example CAP_AUDIT_CONTROL will be AUDIT_CONTROL

   cap_add: [] # eg: ["NET_ADMIN","NET_RAW"], or ["ALL"]
```

Note:

- The `cap_add: []` parameter is applicable for IBM Hyper Protect Virtual Servers version 1.2.3, or later. To enable all privileges' you can use `cap_add:["ALL"]`, but as a good security practice, provide the least possible privileges' to your virtual server. For a complete list of supported parameters in the `secure_create.yaml` file, see Create repository registration.
- Starting with IBM Hyper Protect Virtual Servers version 1.2.4, the term "whitelist" is replaced with "allowlist". For IBM Hyper Protect Virtual Servers versions earlier than 1.2.4, you must use "whitelist" instead of "allowlist".

2. Generate the repository registration file for your image.

```
 hpvs regfile create --config $HOME/hpvs/config/securebuild/secure_create.yaml --
out $HOME/hpvs/config/encryptedRegfile.enc
```

Deploy your own image

Choose one of the following options to deploy your own image.

- By using the `hpvs deploy` command.
- By using the `hpvs vs create` command.

Complete the following steps to deploy your own image by using the `hpvs deploy` command.

1. Update the template file `$HOME/hpvs/config/templates/virtualserver.template.yml` based on the networking configuration, quotagroup and resource settings of the Hyper Protect Virtual Server instance if necessary. The `vs_regfiledeployexample.yml` that has the configuration details for the virtual server refers to the corresponding sections of the `virtualserver.template.yml` when you run the `hpvs deploy` command. For example, the `resourcedefinition: ref` value refers to the `resourcedefinitiontemplate` definition in the template file. The `network: ref` value refers to the `networktemplates` definition in the template file.

```
version: v1
type: virtualserver-template
networktemplates:
-  name: external_network
   subnet: "10.20.4.0/22"
   gateway: "10.20.4.1"
   parent: "encf900"
   driver: "macvlan"
-  name: internal_network
   subnet: "192.168.40.0/24"
   gateway: "192.168.40.1"
   parent: "encf900"
   driver: "bridge"
quotagrouptemplates:
# Passthrough quotagroup templates - A quotagroup will be dynamically created
based
# on the template and attached as single volume mount point to the virtual server.
# Allowed filesystem types for the passthrough type quogagroup are btrfs, ext4,
xfs
-  name: p-small
   size: 20GB
   filesystem : ext4
   passthrough: true
-  name: p-medium
   size: 50GB
   filesystem : ext4
   passthrough: true
-  name: p-large
   size: 100GB
   filesystem : ext4
   passthrough: true
-  name: p-xlarge
   size: 200GB
   filesystem : ext4
   passthrough: true
-  name: p-xxlarge
   size: 400GB
   filesystem : ext4
   passthrough: true
# Non passthrough quotagroup definitions - This quotagroups can be shared by
# creating multiple volume mountpoints with the same virtual server or multiple
# virtual server.  A non passthrough quotagroup will be dynamically created based
# on the template and attached as volume mount points to the virtual server.
# Only brtfs filesystem is supported in non passthrough quotagroups
# mount points attached to virtual server can have filesystem btrfs, ext4, xfs
-  name: np-small
   size: 20GB
   passthrough: false
-  name: np-medium
   size: 50GB
   passthrough: false
-  name: np-large
   size: 100GB
   passthrough: false
-  name: np-xlarge
   size: 200GB
   passthrough: false
-  name: np-xxlarge
   size: 400GB
   passthrough: false
resourcedefinitiontemplates:
-  name: default
   cpu: 1
   memory: 1024
-  name: small
   cpu: 2
   memory: 2048
-  name: large
   cpu: 4
   memory: 4096
-  name: xl
   cpu: 8
   memory: 8192
-  name: xxl
```

```
   cpu: 12
   memory: 12288
```

For more information about the template file for a Hyper Protect Virtual Server instance, see Virtual server template file.

2. Create the configuration yaml file $HOME/hpvs/config/demo_byoi.yml for the instance by referring to the example file $HOME/hpvs/config/vs_regfiledeployexample.yml. The following is an example of a `vs_regfiledeployexample.yml` file.

```
version: v1
type: virtualserver
virtualservers:
- name: testcontainer
  host: SSC_LPAR_NAME
  repoid: MyOwnRepo
  imagetag: latest
  reporegfile: /root/hpvs/config/encryptedRegfile.enc
  imagecache: true
  resourcedefinition:
     ref: small
  networks:
   - ref:  external_network
     ipaddress: 10.20.4.61
  volumes:
   - name: myquotagroup
     ref : np-medium
     mounts:
      - mount_id: new
        mountpoint: /new
        filesystem: ext4
        size: 10GB
```

The **imagecache** parameter is supported when the IBM Hyper Protect Virtual Servers is at version 1.2.3. In this example, the network definition is for an external network. For more information on other network configurations, see Network requirements for Hyper Protect Virtual Server.
For more information about quotagroups in IBM Hyper Protect Virtual Servers, see Overview of quotagroups for IBM Hyper Protect Virtual Servers.
For more information about the config file for a Hyper Protect Virtual Server instance, see Virtual server Configuration file.

3. Deploy the image by using the configurations in the yaml file.

```
hpvs deploy --config $HOME/hpvs/config/demo_byoi.yml
```

Note:

- You can use the **hpvs undeploy** command to delete this virtual server. This command is supported in Hyper Protect Virtual Servers version 1.2.2, or later. For more information, see Undeploying virtual servers.
- You can update the resources or configuration of a virtual server after the completion of the deploy operation by using the **-u**, or the **--update** flag of the **hpvs deploy** command. For more information, see Updating virtual servers.

Complete the following steps to deploy your own image by using the `hpvs vs create` command.

1. Register the repository on the Secure Service Container partition.

```
hpvs repository register --pgp=$HOME/hpvs/config/encryptedRegfile.enc --
id=MyOwnRepo
```

2. Pull the image from the registered DockerHub or IBM Cloud Registry by running the following command (it is recommended to run this command to avoid cache issues).

```
hpvs image pull --tag=latest --repo MyOwnRepo
```

3. Create the quotagroup on the Secure Service Container partition for the Hyper Protect Virtual Server that will host your own Linux-based image.

   ```
   hpvs quotagroup create --name myquotagroup --size=50GB
   ```

   Note: If you create a non-passthrough quotagroup, ensure that you specify a value that is at least 5 GB greater than the size you require for the virtual server. For more information about the `hpvs quotagroup` command, see Commands in IBM Hyper Protect Virtual Servers. For more information about quotagroups in IBM Hyper Protect Virtual Servers, see Overview of quotagroups for IBM Hyper Protect Virtual Servers.

4. Create the network on the Secure Service Container partition for the Hyper Protect Virtual Server that will host your own Linux-based image.

   ```
   hpvs network create --driver macvlan --gateway 10.20.4.1 --name external_network -
   -parent encf900 --subnet 10.20.4.0/22
   ```

   For more information about the `hpvs network` command, see Commands in IBM Hyper Protect Virtual Servers. For more information about the network in IBM Hyper Protect Virtual Servers, see Network requirements for Hyper Protect Virtual Server.

5. Deploy your image as a Hyper Protect Virtual Server.

   ```
   hpvs vs create --name testcontainer --repo MyOwnRepo --tag latest --cpu 2 --ram
   2048  --env={env_var1=value1,env_var2=value2} --quotagroup "{quotagroup =
   myquotagroup, mountid = new, mount = /new, filesystem = btrfs, size = 30GB}" \
   --network "{name = external_network, ip = 10.20.4.188}"
   ```

   Where

   - `--repo MyOwnRepo` must be consist with the repository name when registering the repository.
   - For a complete list of supported parameters and options, see Commands in IBM Hyper Protect Virtual Servers.
   - In this example, the network definition is for an external network. For more information on other network configurations, see Network requirements for Hyper Protect Virtual Server.
   - For more information about quotagroups in IBM Hyper Protect Virtual Servers, see Overview of quotagroups for IBM Hyper Protect Virtual Servers.

Refreshing registered repositories with a new signing key pair

You can update the repositories on the Secure Service Partition with a new signing key pair, and revoke the access from an existing key pair.

Note:

If this task is being performed because the original key pair was compromised, the generation and loading of the new encrypted repository file signed by the new key must be done using a trusted channel.

This procedure is intended for users with the role *Cloud administrator*.

Before you begin

- Ensure that you have the following information of the key pair to be revoked.
  - The private key file

    - The public key file
    - The passphrase of the private key
- Ensure you have a list of repositories registered by using the key to be revoked.
- Ensure that you install GnuPG or a similar tool on your x86 or Linux on IBM Z/LinuxONE (i.e., s390x architecture) management server. For more information, see The GNU Privacy Handbook.

Procedure

Complete the following steps with root user authority.

1. Generate a second set of keys by following the instructions listed in Generating the signing keys. Here, you must use a new name for the keys, for example specify the public key name as **isv_user1.pub** and the private key name as **isv_user1.private**.

2. Use one of the following procedures depending on your task.

    - Scenario: For Secure Build.

        - Re-configure the **secure_build.yml** with the new public key and private key is the earlier key.

```
signing_key:
  private_key_path: '/root/isv_user.private'
  public_key_path: '/root/isv_user1.pub'
```

        - Run the following command to get the signed and encrypted regfile.

```
hpvs sb regfile --config $HOME/hpvs/config/securebuild/secure_build.yml
--out $HOME/hpvs/config/MyDockerAppImageRegfile.enc
```

        - Run the following command to update the repository.

```
hpvs repository update --
pgp=$HOME/hpvs/config/MyDockerAppImageRegfile.enc --id=MyDockerRepo
```

        - Run the following command to re-configure **secure_build.yml** again with the new public key and the new private key.

```
signing_key:
  private_key_path: '/root/isv_user1.private'
  public_key_path: '/root/isv_user1.pub'
```

        - Run the following command to get the signed and encrypted regfile.

```
hpvs sb regfile --config $HOME/hpvs/config/securebuild/secure_build.yml
--out $HOME/hpvs/config/MyDockerAppImageRegfile.enc
```

        - Run the following command to update the repository.

```
hpvs repository update --
pgp=$HOME/hpvs/config/MyDockerAppImageRegfile.enc --id=MyDockerRepo
```

    - Scenario: For deploying your own application

        - Re-configure the **secure_create.yml** with the new public key and private key is the earlier key.

```
signing_key:
    private_key_path: '/root/isv_user.private'
    public_key_path: '/root/isv_user1.pub'
```

        - Run the following command to get the signed and encrypted regfile.

```
hpvs regfile create --config
$HOME/hpvs/config/securebuild/secure_create.yml --out
```

```
$HOME/hpvs/config/encryptedRegfile.enc
```

- Run the following command to update the repository.

  ```
  hpvs repository update --pgp=$HOME/hpvs/config/encryptedRegfile.enc --
  id=MyOwnRepo
  ```

- Run the following command to re-configure **secure_create.yml** again with the new public key and the new private key.

  ```
  signing_key:
      private_key_path: '/root/isv_user1.private'
      public_key_path: '/root/isv_user1.pub'
  ```

- Run the following command to get the signed and encrypted regfile.

  ```
  hpvs regfile create --config
  $HOME/hpvs/config/securebuild/secure_create.yml --out
  $HOME/hpvs/config/encryptedRegfile.enc
  ```

- Run the following command to update the repository.

  ```
  hpvs repository update --pgp=$HOME/hpvs/config/encryptedRegfile.enc --
  id=MyOwnRepo
  ```

Creating the monitoring Virtual Servers

You can monitor a wide range of components with the monitoring infrastructure provided by IBM Hyper Protect Virtual Servers.

Note:

- The monitoring metrics are collected from Secure Service Container partitions.
- Only Hyper Protect Virtual Servers hosting appliance and Secure Service Container partition level metrics are supported for IBM Hyper Protect Virtual Servers 1.2.x.

For more information about collection of metrics, see Metrics collected by the monitoring infrastructure.

This procedure is intended for users with the role *cloud administrator*.

Before you begin

- Refer to the checklist that you prepared for the Hyper Protect Virtual Server on this topic Planning for the environment.
- Ensure that ports **8443** and **25826** are available for the monitoring infrastructure on the Secure Service Container partition.
- Ensure the IBM Hyper Protect Virtual Servers CLI is ready for use. For more information, see Setting up the environment by using the setup script.
- Ensure that running the **setup.sh** script has created the folder structure for monitoring container deployment under **/root/hpvs/config/monitoring**.
- Ensure that you do not specify any external IP details for the monitoring or collectd containers because they use the Secure Service Container's IP with port mapping for getting Secure Service Container LPAR metrics.
- When you create a virtual server, specify a virtual server name that has a maximum of 23 characters, when the version of the Hyper Protect Virtual Servers is 1.2.1, or 1.2.1.1. This restriction does not apply to Hyper Protect Virtual Servers version 1.2.2, or later.

Procedure

On your x86 or Linux on IBM Z/LinuxONE (i.e., s390x architecture) management server, complete the following steps with root user authority.

1. Generate certificates for the secure communication between the Hyper Protect monitoring infrastructure (server) and the monitoring client. The monitor client invoke the **collectd-exporter** endpoint on the server to show the collected metrics. Note that when you generate certificates, use **collectdhost-<METRIC_DN_SUFFIX>.<COMMON_NAME>** or ***.<COMMON_NAME>** as the common name. A wild card certificate with ***.<COMMON_NAME>** common name can be used across multiple partitions. To generate CA signed certificates, see Creating CA signed certificates for the monitoring infrastructure.

2. Copy the certificate and key files for the monitoring infrastructure into the **./keys** directory. The certificate and key are used by monitoring infrastructure to encrypt the metric data in transit. If you create the client certificate to enable the client authentication, you can also copy the client certificate to the **./keys** directory.

   ```
   cp -p server.key $HOME/hpvs/config/monitoring/keys/server.key
   cp -p server-certificate.pem $HOME/hpvs/config/monitoring/keys/server-
   certificate.crt
   cp -p client-certificate.pem $HOME/hpvs/config/monitoring/keys/client-
   certificate.crt
   cp -p myrootCA.crt $HOME/hpvs/config/monitoring/keys/myrootCA.crt
   ```

3. Choose one of the options to provision the instance:

   - By using the yaml configuration file and **hpvs deploy** command.
   - By using the **hpvs vs create** command.

Using the yaml configuration files and hpvs deploy command

This is the recommended option to provision the instance because of it's ease of use and is also an easier method of creating multiple instances quickly.

1. Update the template file **$HOME/hpvs/config/templates/virtualserver.template.yml** based on the networking configuration of the Hyper Protect Virtual Server instance if necessary. The **vs_monitoring.yml** file that has the configuration details for the virtual server refers to the corresponding sections of the **virtualserver.template.yml** when you run the **hpvs deploy** command.

   ```
   version: v1
   type: virtualserver-template
   networktemplates:
   -   name: external_network
       subnet: "10.20.4.0/22"
       gateway: "10.20.4.1"
       parent: encf900
       driver: macvlan
   -   name: internal_network
       subnet: "192.168.40.0/24"
       gateway: "192.168.40.1"
       parent: encf900
       driver: bridge
   quotagrouptemplates:
   # Passthrough quotagroup templates - A quotagroup will be dynamically created
   based
   # on the template and attached as single volume mount point to the virtual server.
   # Allowed filesystem types for the passthrough type quogagroup are btrfs, ext4,
   xfs
   -   name: p-small
       size: 20GB
       filesystem : ext4
       passthrough: true
   -   name: p-medium
       size: 50GB
   ```

```
   filesystem : ext4
   passthrough: true
-  name: p-large
   size: 100GB
   filesystem : ext4
   passthrough: true
-  name: p-xlarge
   size: 200GB
   filesystem : ext4
   passthrough: true
-  name: p-xxlarge
   size: 400GB
   filesystem : ext4
   passthrough: true
# Non passthrough quotagroup definitions - This quotagroups can be shared by
# creating multiple volume mountpoints with the same virtual server or multiple
# virtual server.  A non passthrough quotagroup will be dynamically created based
# on the template and attached as volume mount points to the virtual server.
# Only brtfs filesystem is supported in non passthrough quotagroups
# mount points attached to virtual server can have filesystem btrfs, ext4, xfs
-  name: np-small
   size: 20GB
   passthrough: false
-  name: np-medium
   size: 50GB
   passthrough: false
-  name: np-large
   size: 100GB
   passthrough: false
-  name: np-xlarge
   size: 200GB
   passthrough: false
-  name: np-xxlarge
   size: 400GB
   passthrough: false
resourcedefinitiontemplates:
-  name: default
   cpu: 1
   memory: 1024
-  name: small
   cpu: 2
   memory: 2048
-  name: large
   cpu: 4
   memory: 4096
-  name: xl
   cpu: 8
   memory: 8192
-  name: xxl
   cpu: 12
   memory: 12288
```

- For more information about the template file for a Hyper Protect Virtual Server instance, see Virtual server template file.

2. Create the configuration yaml file $HOME/hpvs/config/monitoring/demo_monitoring.yml for the instance by referring to the example file $HOME/hpvs/config/monitoring/vs_monitoring.yml. The following is an example of a `vs_monitoring.yml` file.

```
version: v1
type: virtualserver
virtualservers:
- name: test-monitoring
 host: SSC_LPAR_NAME
 hostname: monitoring-host-container
 repoid: Monitoring
 imagetag: 1.2.4
 imagefile: Monitoring.tar.gz
 imagecache: true
 environment:
  - key: "PRIVATE_KEY_SERVER"
    value: "@/root/hpvs/config/monitoring/keys/server.key"
  - key: "PUBLIC_CERT_SERVER"
    value: "@/root/hpvs/config/monitoring/keys/server-certificate.crt"
  - key: "PUBLIC_CERT_CLIENT"
    value: "@/root/hpvs/config/monitoring/keys/myrootCA.crt"
  - key: "METRIC_DN_SUFFIX"
    value: "first"
  - key: "COMMON_NAME"
    value: "example.com"
 ports:
  - hostport: 8443
    protocol: tcp
    containerport: 8443
  - hostport: 25826
    protocol: udp
    containerport: 25826
- name: test-collectd
 host: SSC_LPAR_NAME
 hostname: collectd-host-container
 repoid: CollectdHost
 imagetag: 1.2.4
 imagefile: CollectdHost.tar.gz
 imagecache: true
```

The **`imagecache`** parameter is supported when the IBM Hyper Protect Virtual Servers is at version 1.2.3. **Note**: Since an external IP is not specified for the monitoring container, this container can be reached by using Secure Service Container partition's IP address over port the 8443. If you want to customize network, resources or storage settings, please refer to the parameters and examples of Virtual server configuration file. For more information on other network configurations, see Network requirements for Hyper Protect Virtual Server.

3. Create the instance by using the configurations in the yaml file.

```
hpvs deploy --config $HOME/hpvs/config/monitoring/demo_monitoring.yml
```

Note:

- You can use the **`hpvs undeploy`** command to delete this virtual server. This command is supported in Hyper Protect Virtual Servers version 1.2.2, or later. For more information, see Undeploying virtual servers.
- You can update the resources or configuration of a virtual server after the completion of the deploy operation by using the **`-u`**, or the **`--update`** flag of the **`hpvs deploy`** command. For more information, see Updating virtual servers.

By using the `hpvs vs create` command

1. Upload the collectd image to the Secure Service Container partition by using the **`hpvs image load`** command.

```
hpvs image load --file=~/hpvs/config/monitoring/images/CollectdHost.tar.gz
```

2. Upload the monitoring image to the Secure Service Container partition by using the **`hpvs image load`** command.

```
hpvs image load --file=~/hpvs/config/monitoring/images/Monitoring.tar.gz
```

3. Create the collectd container by running the **`hpvs vs create`** command.

```
hpvs vs create --name collectd-host --repo CollectdHost --tag 1.2.4 --hostname
collectd-host-container
```

4. Create the `env.json` file as shown below.

   ```
   {
   "PRIVATE_KEY_SERVER":"@/$HOME/hpvs/config/monitoring/keys/server.key",
   "PUBLIC_CERT_SERVER":"@/$HOME/hpvs/config/monitoring/keys/server-certificate.crt",
   "PUBLIC_CERT_CLIENT":"@/$HOME/hpvs/config/monitoring/keys/myrootCA.crt",
   "METRIC_DN_SUFFIX":"first",
   "COMMON_NAME":"example.com"
   }
   ```

   Note: The COMMON_NAME (CN) value should coincide with CN value used during certificate creation at step 3. For example, if you set COMMON_NAME for creating server certificate as collectdhost-first.example.com, or *.example.com, the COMMON_NAME (CN) in the env.json file must be set to "example.com".

5. Create the monitoring container by running the `hpvs vs create` command.

   ```
   hpvs vs create --name monitoring-host --repo Monitoring --tag 1.2.4 --hostname
   monitoring-host-container --ports "{containerport = 8443, protocol = tcp, hostport
   = 8443}" --ports "{containerport = 25826, protocol = udp, hostport = 25826}" --
   envjsonpath ~/hpvs/config/env.json
   ```

   Note: Since an external IP is not specified for the monitoring container, this container can be reached by using Secure Service Container partition's IP address over port the 8443. For more information on other network configurations, see Network requirements for Hyper Protect Virtual Server.

Next

You can configure any client tools that use the `collectd-exporter` endpoint to collect the monitoring metrics from the monitoring infrastructure.

- The following example file `prometheus.yml` shows how you can configure Prometheus to use the metrics collected by the monitoring infrastructure in IBM Hyper Protect Virtual Servers. Ensure that you copy the required keys and certificates to the file path mentioned in the `prometheus.yml` file.

  ```
  global:
    scrape_interval: 10s
  scrape_configs:
    - job_name: 'prometheus'
      static_configs:
              - targets:  ['collectdhost-first.example.com:8443']
      scheme: https
      tls_config:
          ca_file: /etc/prometheus/keys/server-certificate.pem
          cert_file: /etc/prometheus/keys/client-certificate.pem
          key_file:  /etc/prometheus/keys/client.key
          server_name: collectdhost-first.example.com
  ```

  Note:

  - With a properly configured `prometheus.yml` file, and properly configured, created, and running `monitoring-host` and `collectd-host` containers on the Secure Service Container partition, the targets view of the prometheus server will show the target Secure Service Container partition "State" as "UP" with a default color green.
  - To access the targets view of the Prometheus server, enter the following link with the actual IP address or the hostname of the Prometheus server in your browser.
    `http://<prometheus_server_IP_address_or_hostname>:9090/targets`
- The following example shows how you can view the current monitoring metrics for the `collectdhost-first.example.com` target Secure Service Container partition by using the `wget` utility. In this example, the `wget` command is executed from the directory containing the `prometheus.yml` file's keys, with the output written to the `metrics` file, or a derivative file if the `metrics` file already exists. Make an entry in the `/etc/hosts` file with collectdhost-first.example.com for the server IP(Secure Service Container LPAR IP).

```
wget https://collectdhost-first.example.com:8443/metrics --ca-
certificate=myrootCA.crt --certificate=client-certificate.crt --private-
key=client.key
```

Note: You can also use the **wget** utility with the **--no-check-certificate** option to skip the SSL certificate validation when retrieving the monitoring metrics from the target Secure Service Container partition.

```
wget https://collectdhost-first.example.com:8443/metrics --ca-
certificate=myrootCA.crt --certificate=client-certificate.crt --private-
key=client.key --no-check-certificate
```

Creating CA signed certificates for the monitoring infrastructure

You can generate Certificate Authority (CA) Root and CA signed certificates for the monitoring infrastructure by using the **openssl** utility or any other certificate generation tools that comply with your organization rules.

This procedure is intended for users with the role *cloud administrator*.

Before you begin

- Ensure that you install the OpenSSL or similar tool on a workstation that you can use to generate the certificates.

Procedure

Complete the following steps on your workstation with root user authority.

1. Go to the following directory on your workstation to run the **openssl** command or any similar tool.

   ```
   cd $HOME/hpvs/config/monitoring/keys/ca-certificates
   ```

2. Create CA Root certificates by using the following procedure. The root CA certificate will be used to sign CA certificates.

   a. Create the CA root private key. After the command completes, the CA root private key **myrootCA.key** is generated under the current directory. For example, **$HOME/hpvs/config/monitoring/keys/ca-certificates/myrootCA.key**.

   ```
      openssl genrsa -out myrootCA.key 4096
   ```

   b. Create the Certificate Signing Request (CSR) based on the CA root private key. After the command completes, the CSR **myrootCA.csr** is generated under the current directory. For example, **/$HOME/hpvs/config/monitoring/keys/ca-certificates/myrootCA.csr**.

   1. The command prompts you to enter values for various certificate fields, such as Organization Unit (OU), Common Name (CN), Email, Country Code, State/Province name, City, Organization or Company Name.
      a. Create the CSR file by using the following command.

      ```
         openssl req -verbose -new -key myrootCA.key -out myrootCA.csr -sha256
      ```

      b. Create the CA root certificate by using the following command.

      ```
         openssl ca  -out myrootCA.crt -keyfile myrootCA.key -verbose -selfsign -md
      sha256 -infiles myrootCA.csr
      ```

2. If you want to avoid entering each value when the command runs, you can use a OpenSSL configuration file to create the self signed CSR. For example, **`$HOME/hpvs/config/monitoring/keys/ca-certificates/myca.cnf`**. For more information about the OpenSSL configuration file, see [OpenSSL configuration examples]().

   a. Create other required configuration and OpenSSL database by using the following commands.

   ```
   cd $HOME/hpvs/config/monitoring/keys/ca-certificates/
   touch index.txt
   touch index.txt.attr
   touch serial
   mkdir crl
   mkdir newcerts
   ```

   Note:

   - Update "dir" in myca.cnf to `$HOME/hpvs/config/monitoring/keys/ca-certificates`.
   - Those files are required to successfully create a CA root certificate.
   - You must update the file `serial` and enter a number in the file. For example, `1000`. This number signifies the serial number of the certificates being created.

   b. Create the CA root certificate by using the following command. After the command completes, the CA root certificates `myrootCA.crt` is created under the current directory.

   ```
      openssl ca -config $HOME/hpvs/config/monitoring/keys/ca-
   certificates/myca.cnf -out myrootCA.crt -keyfile myrootCA.key -verbose -
   selfsign -md sha256 -infiles myrootCA.csr
   ```

   c. Validate the CA root certificate by using the following command. After the command completes, the details of the CA root certificate is printed in the output.

   ```
   openssl x509 -noout -text -in myrootCA.crt
   ```

3. Create the CSR for the CA signed server certificate or client certificate by completing the instructions.

   a. Make a note of the details to generate certificates such as the Common Name (CN) and Subject Alternative Name (SAN) that you intend to set in the certificate. For example, `example.com`, `myorg.example.com`. For more information, see [OpenSSL configuration examples]().

   b. Go to the a directory on your workstation to run the `openssl` command or any similar tool.

   ```
   cd $HOME/hpvs/config/monitoring/keys/ca-certificates
   ```

   c. Create a private key by using the following command. After the command completes, a private key will be created under the current directory.

   - For a server certificate, use the following command.

     ```
     openssl genrsa  -out server.key 4096
     ```

   - For a client certificate, use the following command.

     ```
     openssl genrsa  -out client.key 4096
     ```

4. Create a Certificate Signing Request (CSR) based on the private key you just created. You will be asked to enter values for various certificate fields such as Organization Unit (OU), Common Name (CN), Email, Country Code, State or Province name, City, Organization or Company Name. After the command completes, a CSR file is created under the current directory.

   a. If you choose to enter the values for the certificate fields as prompted, then run the following command to create a server certificate.

   ```
   openssl req -new -key server.key -out server-certificate.csr
   ```

Or run the following command to create a client certificate.

```
 openssl req -new -key client.key -out client-certificate.csr
```

b. If you choose to avoid entering these fields on command prompt in an interactive manner, then create a configuration file such as `server-certificate.cnf` and provide the list of these fields and their values as in the following the command for a server certificate.

```
 openssl req -new -config server-certificate.cnf -key server.key -out server-certificate.csr
```

Or a `client-certificate.cnf` configuration file as in the following command for a client certificate.

```
 openssl req -new -config client-certificate.cnf -key client.key -out client-certificate.csr
```

Note:

- To create a server certificate, include the entry `extendedKeyUsage=serverAuth` in the `server-certificate.cnf` file.
- To create a client certificate, include the entry `extendedKeyUsage=clientAuth` in the `client-certificate.cnf` file.
- For the sample configuration files, see OpenSSL configuration examples. After the commands complete, the CSR is created as the `$HOME/hpvs/config/monitoring/keys/ca-certificates/server-certificate.csr` file or `$HOME/hpvs/config/monitoring/keys/ca-certificates/client-certificate.csr` file.

5. Create the CA signed certificates by using the CA root certificate.

   - To create the CA signed server certificate, run the following command.

     ```
     openssl x509 -req -days 365 -in $HOME/hpvs/config/monitoring/keys/ca-certificates/server-certificate.csr -CA $HOME/hpvs/config/monitoring/keys/ca-certificates/myrootCA.crt -CAkey $HOME/hpvs/config/monitoring/keys/ca-certificates/myrootCA.key -CAcreateserial -out ./server-certificate.crt
     ```

   - To create the CA signed client certificate, run the following command.

     ```
     openssl x509 -req -days 365 -in $HOME/hpvs/config/monitoring/keys/ca-certificates/client-certificate.csr -CA $HOME/hpvs/config/monitoring/keys/ca-certificates/myrootCA.crt -CAkey $HOME/hpvs/config/monitoring/keys/ca-certificates/myrootCA.key -CAcreateserial -out ./client-certificate.crt
     ```

Next

You can configure the monitoring infrastructure by following the instructions from the topic Working with Monitoring virtual servers.

Creating the GREP11 container

The GREP11 virtual server supports the Schnorr signature when the Hyper Protect Virtual Servers is at version 1.2.3. The Schnorr algorithm can be used as a signing scheme to generate digital signatures. It is proposed as an alternative algorithm to the Elliptic Curve Digital Signature Algorithm (ECDSA) for cryptographic signatures in the Bitcoin system. The Schnorr signature is known for the simplicity and efficiency.

The GREP11 virtual server supports the Ed25519 public-key signature system when the Hyper Protect Virtual Servers is at version 1.2.2. Ed25519 provides various advantages such as fast single and batch-signature verification, signing ability, key generation, and compact signatures and keys.

The GREP11 virtual server supports BIP32 when the Hyper Protect Virtual Servers is at version 1.2.2.1. BIP32 defines how to derive private and public keys of a wallet from a binary master seed (m) and an ordered set of indices.

The GREP11 virtual server also supports SLIP-0010 when the Hyper Protect Virtual Servers is at version 1.2.2.1. SLIP-0010 describes how to derive private and public key pairs for curve types different from secp256k1.

You can connect to your (Enterprise PKCS #11) EP11 instantiation using a gRPC (GREP11) container on the Secure Service Container partition, and then use the Hardware Security Module (HSM) to perform numerous cryptographic operations, such as generating asymmetric (public and private) key pairs for digital signing and verification, or generating symmetric keys for encrypting data as needed by the deployed applications. For more information, see EP11.

This procedure is intended for users with the role *cloud administrator*.

Before you begin

- Refer to the checklist that you prepared for the Hyper Protect Virtual Server on this topic Planning for the environment.
- Check with your system administrator that the crypto express domain is configured in the EP11 mode. For more information, see **Chapter 8 - Using the Crypto Module Notebook to administer EP11 crypto modules** in the Cryptographic Services ICSF Trusted Key Entry Workstation (TKE) User's Guide.
- Check with your system administrator that the master key is initialized. For more information, see Trusted Key Entry (TKE) CCA Playlist Introduction, and the **Reviewing and changing current logical partition cryptographic controls** topic in the Processor Resource/Systems Manager Planning Guide.
- Check that you have installed the cli tool on your x86 or Linux on IBM Z/LinuxONE (i.e., s390x architecture) management server as a part of the Setting up the environment by using the setup script.
- When you create a virtual server, specify a virtual server name that has a maximum of 23 characters, when the version of the Hyper Protect Virtual Servers is 1.2.1, or 1.2.1.1. This restriction does not apply to Hyper Protect Virtual Servers version 1.2.2, or later.
- Only the `CEX7P` card supports ED25519. This is applicable for Hyper Protect Virtual Servers version 1.2.2, or later, and if you want to use ED25519 to sign or encrypt data.
- The `CEX7P` and `CEX6P`cards supports BIP32 and SLIP-0010. This is applicable for Hyper Protect Virtual Servers version 1.2.2.1, or later, and if you want to use BIP32 and SLIP-0010.
- The `CEX7P` and `CEX6P`cards supports Schnorr signature. This is applicable for Hyper Protect Virtual Servers version 1.2.3, or later, and if you want to use Schnorr signature.
- If you want to use BIP32 or SLIP-0010 features that are supported on Hyper Protect Virtual Servers version 1.2.2.1, or later, then you must complete the following configuration procedures.
  - Contact IBM support to install the EP11 firmware update on the EP11 crypto module. For the z15 systems, the MCL version is P46647.010, and the `CEX7P` card with EP11 Level 4.7.22-4. For the z14 systems the MCL version is P46645.005, and the `CEX6P` card with EP11 Level 3.7.12-2.
  - To enable the new control point (bit 66) in the absence of TKE catcher program support, you can zeroize and re-initialize the domain (or domain group). The EP11 firmware update changes the zeroized state of the new control point from `off` (disabled) to `on` (enabled).
- If you want to use Schnorr signature that is supported on Hyper Protect Virtual Servers version 1.2.3, or later, then you must complete the following configuration procedures.
  - Contact IBM support to install the EP11 firmware update on the EP11 crypto module. For the z15 systems, the MCL version is P46647.012, and the CEX7P card with EP11 Level 4.7.24-1. For the z14 systems the MCL version is P46645.007, and the CEX6P card with EP11 Level 3.7.14-1.
  - To enable the new control point (bit 67) in the absence of TKE catcher program support, you can zeroize and re-initialize the domain (or domain group). The EP11 firmware update changes the zeroized state of the new control point from `off` (disabled) to `on` (enabled).

Procedure

On your x86 or Linux on IBM Z/LinuxONE (i.e., s390x architecture) management server, complete the following steps with root user authority.

1. Generate certificates for the secure communication between the Hyper Protect Virtual Servers GREP11 container and the grep11 client. For more information on generating the certificates, see Creating OpenSSL certificates for GREP11 virtual servers. Copy the keys to the `<$HOME/hpvs>/config/grep11/keys` directory on your x86 or Linux on IBM Z/LinuxONE (i.e., s390x architecture) management server.

2. Check the available crypto domains on the HSM by using the `hpvs crypto list` command. For more information about the `crypto` commands, see Commands in IBM Hyper Protect virtual servers.

   ```
   hpvs crypto list
   ```

   The command might show the following output indicating the crypto domain status.

   ```
   +---------------+--------+
   | CRYPTO.DOMAIN | STATUS |
   +---------------+--------+
   | 07.0000       | online |
   | 07.0007       | online |
   | 07.0009       | online |
   | 09.0000       | online |
   | 09.0007       | online |
   | 09.0009       | online |
   | 09.0007       | in use |
   +---------------+--------+
   ```

   Note:

   - Use the crypto domain that is online. In this example it is "EP11SERVER_EP11CRYPTO_DOMAIN":"07.0007".

3. Choose one of the options to provision the instance:

   - By using the yaml configuration file and `hpvs deploy` command.
   - By using the `hpvs vs create` command.

By using the yaml configuration file and `hpvs deploy` command

This is the recommended option to provision the instance because of it's ease of use and is also an easier method of creating multiple instances quickly.

1. Update the template file `$HOME/hpvs/config/templates/virtualserver.template.yml` based on the networking configuration of the Hyper Protect Virtual Server instance if necessary. The `vs_grep11.yml` that has the configuration details for the virtual server refers to the corresponding sections of the `virtualserver.template.yml` when you run the `hpvs deploy` command. For example, the `network: ref` value refers to the `networktemplates` definition in the template file.

   ```
   version: v1
   type: virtualserver-template
   networktemplates:
   -  name: external_network
      subnet: "10.20.4.0/22"
      gateway: "10.20.4.1"
      parent: encf900
      driver: macvlan
   -  name: internal_network
      subnet: "192.168.40.0/24"
      gateway: "192.168.40.1"
      parent: encf900
      driver: bridge
   quotagrouptemplates:
   # Passthrough quotagroup templates - A quotagroup will be dynamically created 
   based
   # on the template and attached as single volume mount point to the virtual server.
   ```

```
# Allowed filesystem types for the passthrough type quogagroup are btrfs, ext4,
xfs
-  name: p-small
   size: 20GB
   filesystem : ext4
   passthrough: true
-  name: p-medium
   size: 50GB
   filesystem : ext4
   passthrough: true
-  name: p-large
   size: 100GB
   filesystem : ext4
   passthrough: true
-  name: p-xlarge
   size: 200GB
   filesystem : ext4
   passthrough: true
-  name: p-xxlarge
   size: 400GB
   filesystem : ext4
   passthrough: true
# Non passthrough quotagroup definitions - This quotagroups can be shared by
# creating multiple volume mountpoints with the same virtual server or multiple
# virtual server.  A non passthrough quotagroup will be dynamically created based
# on the template and attached as volume mount points to the virtual server.
# Only brtfs filesystem is supported in non passthrough quotagroups
# mount points attached to virtual server can have filesystem btrfs, ext4, xfs
-  name: np-small
   size: 20GB
   passthrough: false
-  name: np-medium
   size: 50GB
   passthrough: false
-  name: np-large
   size: 100GB
   passthrough: false
-  name: np-xlarge
   size: 200GB
   passthrough: false
-  name: np-xxlarge
   size: 400GB
   passthrough: false
resourcedefinitiontemplates:
-  name: default
   cpu: 1
   memory: 1024
-  name: small
   cpu: 2
   memory: 2048
-  name: large
   cpu: 4
   memory: 4096
-  name: xl
   cpu: 8
   memory: 8192
-  name: xxl
   cpu: 12
   memory: 12288
```

2. Create the configuration yaml file $HOME/hpvs/config/grep11/demo_grep11.yml for the instance by referring to the example file $HOME/hpvs/config/grep11/vs_grep11.yml.

   The following is an example of a vs_grep11.yml file that uses port mapping for the network.

   ```
   version: v1
   type: virtualserver
   ```

```
virtualservers:
- name: test-grep11
  host: SSC_LPAR_NAME
  repoid: hpcsKpGrep11_runq
  imagetag: 1.2.4
  hostname: grep11.example.com
  imagefile: hpcsKpGrep11_runq.tar.gz
  imagecache: true
  crypto:
     crypto_matrix:
      - 07.0007
  networks:
   - ref:  external_network
     ipaddress: 10.20.4.12
  environment:
   - key: EP11SERVER_EP11CRYPTO_DOMAIN
     value: "07.000c"
   - key: EP11SERVER_EP11CRYPTO_CONNECTION_TLS_CERTFILEBYTES
     value: "@/root/hpvs/config/grep11/keys/server.pem"
   - key: EP11SERVER_EP11CRYPTO_CONNECTION_TLS_KEYFILEBYTES
     value: "@/root/hpvs/config/grep11/keys/server-key.pem"
   - key: EP11SERVER_EP11CRYPTO_CONNECTION_TLS_CACERTBYTES
     value: "@/root/hpvs/config/grep11/keys/ca.pem"
   - key: EP11SERVER_EP11CRYPTO_CONNECTION_TLS_ENABLED
     value: "true"
   - key: EP11SERVER_EP11CRYPTO_CONNECTION_TLS_MUTUAL
     value: "true"
   - key: TLS_GRPC_CERTS_DOMAIN_CRT
     value: "\\n"
   - key: TLS_GRPC_CERTS_DOMAIN_KEY
     value: "\\n"
   - key: TLS_GRPC_CERTS_ROOTCA_CRT
     value: "\\n"
```

The `imagecache` parameter is supported when the IBM Hyper Protect Virtual Servers is at version 1.2.3. You must access the GREP11 service via port 9876. In this example, the network definition is for an external network. For more information on other network configurations, see Network requirements for Hyper Protect Virtual Server.
Note: The values *key: "EP11SERVER_EP11CRYPTO_ENABLED"*, and *value: "true"*, specified in the example vs_grep11.yml as shown above, are applicable only for IBM Hyper Protect Virtual Servers version 1.2.2, or later.

3. Create the instance by using the configurations in the yaml file.

   ```
   hpvs deploy --config $HOME/hpvs/config/grep11/demo_grep11.yml
   ```

   Note:

   - You can use the `hpvs undeploy` command to delete this virtual server. This command is supported in Hyper Protect Virtual Servers version 1.2.2, or later. For more information, see Undeploying virtual servers.
   - You can update the resources or configuration of a virtual server after the completion of the deploy operation by using the `-u`, or the `--update` flag of the `hpvs deploy` command. For more information, see Updating virtual servers.

By using the `hpvs vs create` command

1. Upload the GREP11 image to the Secure Service Container partition by using the `hpvs image load` command.

   ```
    hpvs image load --file $HOME/hpvs/config/grep11/images/hpcsKpGrep11_runq.tar.gz
   ```

2. Create the `grep11_env.json` file as shown below.

```
 {
  "EP11SERVER_EP11CRYPTO_DOMAIN":"07.0007",

"EP11SERVER_EP11CRYPTO_CONNECTION_TLS_CERTFILEBYTES":"@/$HOME/hpvs/config/grep11/k
eys/server.pem",

"EP11SERVER_EP11CRYPTO_CONNECTION_TLS_KEYFILEBYTES":"@/$HOME/hpvs/config/grep11/ke
ys/server-key.pem",

"EP11SERVER_EP11CRYPTO_CONNECTION_TLS_CACERTBYTES":"@/$HOME/hpvs/config/grep11/key
s/ca.pem",
  "EP11SERVER_EP11CRYPTO_CONNECTION_TLS_ENABLED":true,
  "EP11SERVER_EP11CRYPTO_CONNECTION_TLS_MUTUAL":true,
  "EP11SERVER_EP11CRYPTO_ENABLED":"true",
  "TLS_GRPC_CERTS_DOMAIN_CRT":"\\n",
  "TLS_GRPC_CERTS_DOMAIN_KEY":"\\n",
  "TLS_GRPC_CERTS_ROOTCA_CRT":"\\n"
 }
```

Note:

- The "server.pem", "server-key.pem", and "ca.pem" files are created as a part of the generation of certificates for the secure communication between the Hyper Protect Virtual Servers GREP11 container and the grep11 client.
- The value *key: "EP11SERVER_EP11CRYPTO_ENABLED"*, specified in the **grep11_env.json** file as shown above, is applicable only for IBM Hyper Protect Virtual Servers version 1.2.2, or later.

3. Create the external network for the GREP11 virtual server.

```
hpvs network create --name external_net --driver macvlan --parent encf900 --subnet
10.20.4.0/22 --gateway 10.20.4.1
```

For more information about the **hpvs network** command, see Commands in IBM Hyper Protect Virtual Servers. For more information about the network in IBM Hyper Protect Virtual Servers, see Network requirements for Hyper Protect Virtual Server.

4. Create the GREP11 container by running the **hpvs vs create** command.

```
hpvs vs create --name grep11container --repo hpcsKpGrep11_runq --tag 1.2.4 --
crypto_matrix=07.0007 --cpu 2 --ram 2048 --envjsonpath
/Users/username/hpvs_config/crypto/grep11_env.json --network "{name =
external_network, ip = 10.20.4.12}"
```

For more information about the TKE, check out the video on YouTube - TKE Introduction Videos 1 Introduction to TKE.

Next

You can update your application to use the asymmetric key pairs provided by the GREP11 containers. For more information about how to verify if the GREP11 virtual server is working as expected, refer to Testing the GREP11 virtual server.

Creating OpenSSL certificates for GREP11 Virtual Servers

You can generate Certificate Authority (CA) signed certificates for the Grep11 infrastructure by using the **openssl** utility.

This procedure is intended for users with the role *cloud administrator*.

Before you begin

- Ensure that you install the OpenSSL on a workstation that you can use to generate the certificates.

Procedure

Complete the following steps on your workstation with root user authority.

1. Generate the CA key by running the following command.

   ```
   openssl genrsa -out ca.key 2048
   ```

2. Create the CA certificate by running the following command.

   ```
   openssl req -new -x509 -key ca.key -days 730 -out ca.pem
   ```

3. Generate the Server key by running the following command.

   ```
   openssl genrsa -out server-key.pem 2048
   ```

4. Export the COMMON_NAME (fully qualified domain name), path length, and Subject Alternative Name (to indicate all of the domain names and IP addresses that are secured by the certificate) by running the following commands. These values will be used to generate the server certificate.

   ```
   export COMMON_NAME=grep11.example.com
   export PATHLEN=CA:true
   export SUBJECT_ALT_NAME=DNS:<domain-name:port>,IP:<ip>
   e.g. export SUBJECT_ALT_NAME=DNS:grep11.example.com:9876,IP:10.20.6.62
   ```

5. Create the **openssl.cnf** file and copy the content given below.

   ```
    # OpenSSL configuration file.
    #

    # Establish working directory.

    dir   = .

    [ ca ]
    default_ca  = CA_default

    [ CA_default ]
    serial   = $dir/serial
    #database  = ${ENV::DIR}/index.txt
    #new_certs_dir  = $dir/newcerts
    #private_key       = $dir/ca.key
    #certificate       = $dir/ca.cer
    default_days  = 730
    default_md  = sha256
    preserve  = no
    email_in_dn  = no
    nameopt   = default_ca
    certopt   = default_ca
    default_crl_days = 45
    policy   = policy_match

    [ policy_match ]
    countryName  = match
    stateOrProvinceName = optional
    organizationName = match
    organizationalUnitName = optional
    commonName  = supplied
    emailAddress  = optional
   ```

```
 [ req ]
 default_md  = sha256
 distinguished_name = req_distinguished_name
 prompt             = yes

 [ req_distinguished_name ]
 #countryName = Country
 #countryName_default = US
 #countryName_min = 2
 #countryName_max = 2
 #localityName = Locality
 #localityName_default = Los Angeles
 #organizationName = Organization
 #organizationName_default = IBM
 #commonName = Common Name
 #commonName_max = 64

 C  = US
 ST = California
 L  = Los Angeles
 O  = IBM
 CN = ${ENV::COMMON_NAME}

 [ certauth ]
 subjectKeyIdentifier = hash
 authorityKeyIdentifier = keyid:always,issuer:always
 keyUsage = digitalSignature, keyEncipherment, dataEncipherment, keyCertSign, 
cRLSign
 keyUsage = digitalSignature, keyEncipherment, dataEncipherment, keyCertSign, 
cRLSign
 basicConstraints = ${ENV::PATHLEN}
 #crlDistributionPoints = @crl

 [ server ]
 basicConstraints = CA:FALSE
 keyUsage = digitalSignature, keyEncipherment, dataEncipherment
 extendedKeyUsage = serverAuth
 nsCertType = server
 crlDistributionPoints = @crl
 subjectAltName = ${ENV::SUBJECT_ALT_NAME}

 [ client ]
 basicConstraints = CA:FALSE
 keyUsage = digitalSignature, keyEncipherment, dataEncipherment
 extendedKeyUsage = clientAuth,msSmartcardLogin
 nsCertType = client
 crlDistributionPoints = @crl
 authorityInfoAccess = @ocsp_section
 subjectAltName = @alt_names

 [ selfSignedServer ]
 subjectKeyIdentifier = hash
 authorityKeyIdentifier = keyid:always,issuer:always
 keyUsage = digitalSignature, keyEncipherment, dataEncipherment
 basicConstraints = CA:FALSE
 subjectAltName = ${ENV::SUBJECT_ALT_NAME}
 extendedKeyUsage = serverAuth

 [ selfSignedClient ]
 subjectKeyIdentifier = hash
 authorityKeyIdentifier = keyid:always,issuer:always
 keyUsage = digitalSignature, keyEncipherment, dataEncipherment
 basicConstraints = CA:FALSE
 subjectAltName = @alt_names
 extendedKeyUsage = clientAuth

 [ server_client ]
 subjectKeyIdentifier = hash
 keyUsage = digitalSignature, keyEncipherment, dataEncipherment
 basicConstraints = CA:FALSE
 subjectAltName = ${ENV::SUBJECT_ALT_NAME}
 crlDistributionPoints = @crl
 extendedKeyUsage = serverAuth,clientAuth

 [ v3_intermediate_ca ]
 # Extensions for a typical intermediate CA (`man x509v3_config`).
 subjectKeyIdentifier = hash
 authorityKeyIdentifier = keyid:always,issuer
 basicConstraints = critical, ${ENV::PATHLEN}
 keyUsage = critical, digitalSignature, cRLSign, keyCertSign
 crlDistributionPoints = @crl
 authorityInfoAccess = @ocsp_section

 [ crl ]
 URI=http://localhost/ca.crl

 [ ocsp_section ]
 OCSP;URI.0 = http://localhost:2560/ocsp

 [ ocsp ]
 # Extension for OCSP signing certificates (`man ocsp`).
 basicConstraints = CA:FALSE
 subjectKeyIdentifier = hash
 authorityKeyIdentifier = keyid,issuer
 keyUsage = critical, digitalSignature
 extendedKeyUsage = critical, OCSPSigning

 [alt_names]
 # email= ${ENV::SUBJECT_ALT_NAME}
 otherName=msUPN;UTF8:${ENV::SUBJECT_ALT_NAME}

 [v3_conf]
 keyUsage = digitalSignature, keyEncipherment, dataEncipherment, keyCertSign,
cRLSign
 basicConstraints = CA:FALSE
```

6. Create the server certificate signing request by running the following command.

   ```
   openssl req -new -key server-key.pem -out server.csr
   ```

7. Create the server certificate by running the following command.

   ```
   openssl x509 -sha256 -req -in server.csr -CA ca.pem -CAkey ca.key -set_serial 8086
   -extfile openssl.cnf -extensions server -days 730 -outform PEM -out server.pem
   ```

8. Create the client key by running the following command.

   ```
   openssl genrsa -out client-key.pem 2048
   ```

9. Create the client certificate signing request by running the following command.

   ```
   openssl req -new -key client-key.pem -out client.csr
   ```

10. Create the client certificate by running the following command.

    ```
    openssl x509 -req -days 730 -in client.csr -CA ca.pem -CAcreateserial -CAkey
    ca.key -out client.pem
    ```

Next

You can proceed with configuring of the GREP11 infrastructure as instructed in the Working with GREP11 virtual servers

Testing the GREP11 virtual server

You can use the Enterprise PKCS #11 (EP11) API over gRPC (also referred to as GREP11 API) to remotely access the GREP11 container on the Secure Service Container partition for data encryption and management.

This procedure is intended for users with the role *cloud administrator*.

Before you begin

- Ensure that you have the connection information to the GREP11 container on the Secure Service Container partition. For example, `grep11.example.com:9876`.
- Ensure that you have the client certificates to authenticate with the GREP11 container.
- For mutual TLS communication, you need the client private key `client-key.pem`, the root certificate `ca.pem`, and the public certificate `client.pem`
- Only the `CEX7P` card is supported with ED25519. This is applicable for Hyper Protect Virtual Servers version 1.2.2, or later, and if you want to use ED25519 to sign or encrypt data.
- The `CEX7P` and `CEX6P`cards are supported with BIP32 and SLIP-0010. This is applicable for Hyper Protect Virtual Servers version 1.2.2.1, or later, and if you want to use BIP32 and SLIP-0010.
- The `CEX7P` and `CEX6P`cards supports Schnorr signature. This is applicable for Hyper Protect Virtual Servers version 1.2.3, or later, and if you want to use Schnorr signature.

Procedure

1. Install Golang by following the instructions provided here: https://golang.org/doc/install

2. Set the `PATH` for "go" by running the following commands.

   ```
   export GOROOT=/usr/local/go
   export GOPATH=$HOME/go
   export PATH=$GOPATH/bin:$GOROOT/bin:$PATH
   ```

3. Run the following commands.

   ```
   go get gopkg.in/yaml.v2
   go get "github.com/gogo/googleapis/google/api"
   go get "github.com/stretchr/testify/assert"
   ```

4. Complete the following steps to obtain the Golang example code that is used to test the GREP11 virtual server.

   ```
   mkdir $HOME/go/src/github.com/ibm-developer
   cd $HOME/go/src/github.com/ibm-developer
   git clone -b onprem https://github.com/pranjank/ibm-cloud-hyperprotectcrypto.git
   cd ibm-cloud-hyperprotectcrypto/golang/examples
   ```

5. Configure credential.yaml. The following is an example.

   ```
   url: "grep11.example.com:9876"
   cert_path: "/root/hpvs/config/grep11/keys/client.pem"
   key_path: "/root/hpvs/config/grep11/keys/client-key.pem"
   cacert_path: "/root/hpvs/config/grep11/keys/ca.pem"
   ```

6. Run the following command to test the GREP11 virtual server.

   ```
   go test -v server_test.go
   ```

   On successful completion, you might see a display similar to the following.

```
=== RUN   Example_getMechanismInfo
--- PASS: Example_getMechanismInfo (0.11s)
=== RUN   Example_encryptAndDecrypt
--- PASS: Example_encryptAndDecrypt (0.26s)
=== RUN   Example_digest
--- PASS: Example_digest (0.19s)
=== RUN   Example_signAndVerifyUsingRSAKeyPair
--- PASS: Example_signAndVerifyUsingRSAKeyPair (0.20s)
=== RUN   Example_signAndVerifyUsingECDSAKeyPair
--- PASS: Example_signAndVerifyUsingECDSAKeyPair (0.16s)
=== RUN   Example_signAndVerifyUsingECDSAKeyPairWithSchnorr
--- PASS: Example_signAndVerifyUsingECDSAKeyPairWithSchnorr (0.14s)
=== RUN   Example_signAndVerifyToTestErrorHandling
--- PASS: Example_signAndVerifyToTestErrorHandling (0.17s)
=== RUN   Example_wrapAndUnwrapKey
--- PASS: Example_wrapAndUnwrapKey (0.18s)
PASS
ok      command-line-arguments  1.557s
```

7. Run the following command to test ED25519 on the GREP11 virtual server (**Note**: This step is applicable only for IBM Hyper Protect Virtual Servers version 1.2.2, or later).

   ```
   go test -v ed25519_test.go
   ```

   On successful completion, you might see a display similar to the following.

   ```
   === RUN   TestED25519NewMechanism
   --- PASS: TestED25519NewMechanism (0.96s)
       ed25519_test.go:55: Testing GetMechanismList(), Checking 
   [CKM_IBM_ED25519_SHA512]: OK
       ed25519_test.go:69: Testing GetMechanismInfo(), Checking 
   [CKM_IBM_ED25519_SHA512]: OK
   === RUN   TestED25519SignVerify
   --- PASS: TestED25519SignVerify (1.72s)
      ed25519_test.go:160: Testing GenerateKeyPair(), Generated ED25519 PKCS key pair
      ed25519_test.go:218: Testing SignInit() and Sign(), Data signed by using 
   SignInit() and Sign() with ED25519 PKCS key pair
      ed25519_test.go:277: Testing VerifyInit() and Verify(), ED25519 signature 
   verified by using VerifyInit() and Verify()
   === RUN   TestED25519SignVerifyMulti
   --- PASS: TestED25519SignVerifyMulti (1.66s)
       ed25519_test.go:160: Testing GenerateKeyPair(), Generated ED25519 PKCS key 
   pair
   === RUN   TestED25519SignVerifySingle
   --- PASS: TestED25519SignVerifySingle (1.25s)
       ed25519_test.go:160: Testing GenerateKeyPair(), Generated ED25519 PKCS key 
   pair
       ed25519_test.go:175: Testing SignSingle(), Data signed by using SignSingle() 
   with ED25519 PKCS key pair
       ed25519_test.go:196: Testing VerifySingle(), ED25519 signature verified by 
   using VerifySingle()
   === RUN   TestED25519SignVerifyCrosstest
   --- PASS: TestED25519SignVerifyCrosstest (2.22s)
       ed25519_test.go:160: Testing GenerateKeyPair(), Generated ED25519 PKCS key 
   pair
       ed25519_test.go:175: Testing SignSingle(), Data signed by using SignSingle() 
   with ED25519 PKCS key pair
       ed25519_test.go:277: Testing VerifyInit() and Verify(), ED25519 signature 
   verified by using VerifyInit() and Verify()
       ed25519_test.go:218: Testing SignInit() and Sign(), Data signed by using 
   SignInit() and Sign() with ED25519 PKCS key pair
       ed25519_test.go:196: Testing VerifySingle(), ED25519 signature verified by 
   using VerifySingle()
   === RUN   TestED25519InvalidKeyType
   --- PASS: TestED25519InvalidKeyType (2.75s)
       ed25519_test.go:369: Testing GenerateKeyPair(), Generated ECDSA PKCS key pair 
   for negative test
   ```

```
    ed25519_test.go:160: Testing GenerateKeyPair(), Generated ED25519 PKCS key 
pair
    ed25519_test.go:175: Testing SignSingle(), Data signed by using SignSingle() 
with ED25519 PKCS key pair
=== RUN   TestED25519InvalidKeys
--- PASS: TestED25519InvalidKeys (8.36s)
    ed25519_test.go:160: Testing GenerateKeyPair(), Generated ED25519 PKCS key 
pair
    ed25519_test.go:175: Testing SignSingle(), Data signed by using SignSingle() 
with ED25519 PKCS key pair
    ed25519_test.go:160: Testing GenerateKeyPair(), Generated ED25519 PKCS key 
pair
    ed25519_test.go:175: Testing SignSingle(), Data signed by using SignSingle() 
with ED25519 PKCS key pair
    ed25519_test.go:160: Testing GenerateKeyPair(), Generated ED25519 PKCS key 
pair
    ed25519_test.go:175: Testing SignSingle(), Data signed by using SignSingle() 
with ED25519 PKCS key pair
    ed25519_test.go:160: Testing GenerateKeyPair(), Generated ED25519 PKCS key 
pair
    ed25519_test.go:175: Testing SignSingle(), Data signed by using SignSingle() 
with ED25519 PKCS key pair
=== RUN   TestED25519InvalidSignature
--- PASS: TestED25519InvalidSignature (7.65s)
    ed25519_test.go:160: Testing GenerateKeyPair(), Generated ED25519 PKCS key 
pair
    ed25519_test.go:175: Testing SignSingle(), Data signed by using SignSingle() 
with ED25519 PKCS key pair
    ed25519_test.go:369: Testing GenerateKeyPair(), Generated ECDSA PKCS key pair 
for negative test
    ed25519_test.go:385: Testing SignSingle(), Data signed by using SignSingle() 
with ECDSA PKCS key pair for  negative test
=== RUN   TestED25519ParallelGenerateKey
--- PASS: TestED25519ParallelGenerateKey (1.00s)
    ed25519_test.go:160: Testing GenerateKeyPair(), Generated ED25519 PKCS key 
pair
    ed25519_test.go:160: Testing GenerateKeyPair(), Generated ED25519 PKCS key 
pair
    ed25519_test.go:160: Testing GenerateKeyPair(), Generated ED25519 PKCS key 
pair
    ed25519_test.go:160: Testing GenerateKeyPair(), Generated ED25519 PKCS key 
pair
    ed25519_test.go:160: Testing GenerateKeyPair(), Generated ED25519 PKCS key 
pair
    ed25519_test.go:160: Testing GenerateKeyPair(), Generated ED25519 PKCS key 
pair
    ed25519_test.go:160: Testing GenerateKeyPair(), Generated ED25519 PKCS key 
pair
    ed25519_test.go:160: Testing GenerateKeyPair(), Generated ED25519 PKCS key 
pair
    ed25519_test.go:160: Testing GenerateKeyPair(), Generated ED25519 PKCS key 
pair
    ed25519_test.go:160: Testing GenerateKeyPair(), Generated ED25519 PKCS key 
pair
=== RUN   TestED25519ParallelSignVerify
--- PASS: TestED25519ParallelSignVerify (2.54s)
    ed25519_test.go:160: Testing GenerateKeyPair(), Generated ED25519 PKCS key 
pair
    ed25519_test.go:218: Testing SignInit() and Sign(), Data signed by using 
SignInit() and Sign() with ED25519 PKCS key pair
    ed25519_test.go:218: Testing SignInit() and Sign(), Data signed by using 
SignInit() and Sign() with ED25519 PKCS key pair
    ed25519_test.go:218: Testing SignInit() and Sign(), Data signed by using 
SignInit() and Sign() with ED25519 PKCS key pair
    ed25519_test.go:218: Testing SignInit() and Sign(), Data signed by using 
SignInit() and Sign() with ED25519 PKCS key pair
    ed25519_test.go:218: Testing SignInit() and Sign(), Data signed by using 
```

```
SignInit() and Sign() with ED25519 PKCS key pair
    ed25519_test.go:218: Testing SignInit() and Sign(), Data signed by using 
SignInit() and Sign() with ED25519 PKCS key pair
    ed25519_test.go:218: Testing SignInit() and Sign(), Data signed by using 
SignInit() and Sign() with ED25519 PKCS key pair
    ed25519_test.go:218: Testing SignInit() and Sign(), Data signed by using 
SignInit() and Sign() with ED25519 PKCS key pair
    ed25519_test.go:218: Testing SignInit() and Sign(), Data signed by using 
SignInit() and Sign() with ED25519 PKCS key pair
    ed25519_test.go:218: Testing SignInit() and Sign(), Data signed by using 
SignInit() and Sign() with ED25519 PKCS key pair
    ed25519_test.go:218: Testing SignInit() and Sign(), Data signed by using 
SignInit() and Sign() with ED25519 PKCS key pair
    ed25519_test.go:277: Testing VerifyInit() and Verify(), ED25519 signature 
verified by using VerifyInit() and Verify()
    ed25519_test.go:277: Testing VerifyInit() and Verify(), ED25519 signature 
verified by using VerifyInit() and Verify()
    ed25519_test.go:277: Testing VerifyInit() and Verify(), ED25519 signature 
verified by using VerifyInit() and Verify()
    ed25519_test.go:277: Testing VerifyInit() and Verify(), ED25519 signature 
verified by using VerifyInit() and Verify()
    ed25519_test.go:277: Testing VerifyInit() and Verify(), ED25519 signature 
verified by using VerifyInit() and Verify()
    ed25519_test.go:277: Testing VerifyInit() and Verify(), ED25519 signature 
verified by using VerifyInit() and Verify()
    ed25519_test.go:277: Testing VerifyInit() and Verify(), ED25519 signature 
verified by using VerifyInit() and Verify()
    ed25519_test.go:277: Testing VerifyInit() and Verify(), ED25519 signature 
verified by using VerifyInit() and Verify()
    ed25519_test.go:277: Testing VerifyInit() and Verify(), ED25519 signature 
verified by using VerifyInit() and Verify()
    ed25519_test.go:277: Testing VerifyInit() and Verify(), ED25519 signature 
verified by using VerifyInit() and Verify()
=== RUN   TestED25519ParallelSignVerifySingle
--- PASS: TestED25519ParallelSignVerifySingle (2.10s)
    ed25519_test.go:160: Testing GenerateKeyPair(), Generated ED25519 PKCS key 
pair
    ed25519_test.go:175: Testing SignSingle(), Data signed by using SignSingle() 
with ED25519 PKCS key pair
    ed25519_test.go:175: Testing SignSingle(), Data signed by using SignSingle() 
with ED25519 PKCS key pair
    ed25519_test.go:175: Testing SignSingle(), Data signed by using SignSingle() 
with ED25519 PKCS key pair
    ed25519_test.go:175: Testing SignSingle(), Data signed by using SignSingle() 
with ED25519 PKCS key pair
    ed25519_test.go:175: Testing SignSingle(), Data signed by using SignSingle() 
with ED25519 PKCS key pair
    ed25519_test.go:175: Testing SignSingle(), Data signed by using SignSingle() 
with ED25519 PKCS key pair
    ed25519_test.go:175: Testing SignSingle(), Data signed by using SignSingle() 
with ED25519 PKCS key pair
    ed25519_test.go:175: Testing SignSingle(), Data signed by using SignSingle() 
with ED25519 PKCS key pair
    ed25519_test.go:175: Testing SignSingle(), Data signed by using SignSingle() 
with ED25519 PKCS key pair
    ed25519_test.go:175: Testing SignSingle(), Data signed by using SignSingle() 
with ED25519 PKCS key pair
    ed25519_test.go:175: Testing SignSingle(), Data signed by using SignSingle() 
with ED25519 PKCS key pair
    ed25519_test.go:196: Testing VerifySingle(), ED25519 signature verified by 
using VerifySingle()
    ed25519_test.go:196: Testing VerifySingle(), ED25519 signature verified by 
using VerifySingle()
    ed25519_test.go:196: Testing VerifySingle(), ED25519 signature verified by 
using VerifySingle()
    ed25519_test.go:196: Testing VerifySingle(), ED25519 signature verified by 
using VerifySingle()
```

```
    ed25519_test.go:196: Testing VerifySingle(), ED25519 signature verified by 
using VerifySingle()
    ed25519_test.go:196: Testing VerifySingle(), ED25519 signature verified by 
using VerifySingle()
    ed25519_test.go:196: Testing VerifySingle(), ED25519 signature verified by 
using VerifySingle()
    ed25519_test.go:196: Testing VerifySingle(), ED25519 signature verified by 
using VerifySingle()
    ed25519_test.go:196: Testing VerifySingle(), ED25519 signature verified by 
using VerifySingle()
    ed25519_test.go:196: Testing VerifySingle(), ED25519 signature verified by 
using VerifySingle()
PASS
ok      command-line-arguments    37.947s
```

8. Run the following command to test BIP32 on the GREP11 virtual server (**Note**: This step is applicable only for IBM Hyper Protect Virtual Servers version 1.2.2.1, or later).

   ```
   go test -v bip32_test.go
   ```

   On successful completion, you might see a display similar to the following.

   ```
   === RUN   Example_bip32DeriveKey
   --- PASS: Example_bip32DeriveKey (1.15s)
   === RUN   Example_bip32_Base
   --- PASS: Example_bip32_Base (0.82s)
   === RUN   Example_bip32_KeyDerivation
   --- PASS: Example_bip32_KeyDerivation (0.21s)
   === RUN   Example_bip32_Cross_SignVerify
   --- PASS: Example_bip32_Cross_SignVerify (0.72s)
   PASS
   ok      command-line-arguments  3.068s
   ```

9. Run the following command to test SLIP10 on the GREP11 virtual server (**Note**: This step is applicable only for IBM Hyper Protect Virtual Servers version 1.2.2.1, or later).

   ```
   go test -v slip10_test.go
   ```

   On successful completion, you might see a display similar to the following.

   ```
   === RUN   Example_slip10DeriveKey
   --- PASS: Example_slip10DeriveKey (3.05s)
   === RUN   Example_slip10_invalid_signAndVerify
   --- PASS: Example_slip10_invalid_signAndVerify (0.58s)
   === RUN   Example_slip10_cross_signAndVerify
   --- PASS: Example_slip10_cross_signAndVerify (0.75s)
   PASS
   ok      command-line-arguments  4.503s
   ```

Backing up and restoring IBM Hyper Protect Virtual Servers

You can create backups and restore from those backups as part of your disaster recovery plan.

Note: Backup and restore feature is supported by the following components:

- Hyper Protect Virtual Servers hosting appliance
- Hyper Protect virtual server containers

Procedure

To back up and restore IBM Hyper Protect Virtual Servers, complete the following procedure according to your role.

- As a system or appliance administrator, back up and restore the hosting appliance by using the Secure Service Container user interface. For more information, download Secure Service Container User's Guide, SC28-6978-02a.
    - To create a backup, use the **`Export`** button on the navigation pane. The configuration file **`export.data`** will be stored on your local file system.
    - To restore the hosting appliance from a backup, use the **`Import`** button on the navigation pane, and then upload the exported configuration file **`export.data`** as instructed. After the restore is completed, the Hyper Protect Virtual Servers hosting appliance will be restarted.
- As an application developer or ISV, you can back up and restore the Hyper Protect Virtual Server containers with your application code by using the **`hpvs snapshot`** commands. For more information, see hpvs snapshot.
    - To create a backup for a Hyper Protect Virtual Server container with your application, use the **`hpvs snapshot`** command as in the following command example.

      ```
      hpvs snapshot create --name hpvs_snapshot1 --vs testcontainer
      ```

      Note: The snapshots of the Hyper Protect Virtual Server containers are stored on the Secure Service Container partition.

    - To restore the Hyper Protect Virtual Server container from a snapshot, use the **`hpvs snapshot restore`** command as in the following command example. You must restart the Hyper Protect Virtual Server container after it is restored from a snapshot.

      ```
      hpvs snapshot restore --name hpvs_snapshot1 --vs testcontainer
      ```

      Note: This command restores all the quotagroups associated with the virtual server. To restore a specific quotagroup, run the following command. In the following example, only the **`myquotagroup`** is restored.

      ```
      hpvs snapshot restore --name hpvs_snapshot1 --vs testcontainer --quotagroup myquotagroup
      ```

Limitations of the `hpvs snapshot` commands

- You must delete a container only after you have deleted its snapshots. You cannot retrieve the snapshots if you fail to do so.
- You cannot create snapshots of a virtual server that has passthrough quotagroups.
- You cannot create snapshots of a virtual server that has multiple quotagroups attached when any of them is a passthrough quotagroup.
- If you do not specify any quotagroups when creating a virtual server, the snapshot command fails.
- Snapshot restore from Hosting Appliance Version 1.1.18 to Hosting Appliance Version 3.11.4 is not supported because of changes related to the using keys in the Hosting Appliance Version 3.11.4.

Undeploying virtual servers

When the IBM Hyper Protect virtual Servers is at Version 1.2.2, or later, you can use the **`hpvs undeploy`** command to delete existing virtual server instances along with resources like networks, and quotagroups, that were allocated to that virtual server. Only resources that are not shared with other virtual servers are deleted. This command also deletes all images, and repositories that are not shared with other virtual servers.

This procedure is intended for users with the role *cloud administrator*.

Before you begin

- Ensure the IBM Hyper Protect Virtual Servers CLI is ready for use. For more information, see Setting up the environment by using the setup script.

Procedure

On your x86 or Linux on IBM Z/LinuxONE (i.e., s390x architecture) management server, complete the following steps with root user authority.

1. You can use the configuration yaml file **`$HOME/hpvs/config/yaml/vs_hpvsopbasessh.yml`** that you used for creating the virtual server instance. The following is an example of the **`vs_hpvsopbasessh.yml`** file.

```
version: v1
type: virtualserver
virtualservers:
- name: test-hpvsopbasessh
  host: SSC_LPAR_NAME
  hostname: hpvsopbasessh-container
  repoid: HpvsopBaseSSH
  imagetag: 1.2.1-abcdefg
  imagefile: HpvsopBaseSSH.tar.gz
  resourcedefinition:
     ref: small
  environment:
   - key: LOGTARGET
     value: "/dev/console"
   - key: ROOTFS_LOCK
     value: "y"
   - key: SSH_PUBLIC_KEY
     value: "@/root/hpvs/config/hpvsopbasessh/keys/id_rsa.pub"
  networks:
   - ref:  external_network
     ipaddress: 10.20.4.12
  volumes:
   - name: qg_hpvsopbasessh
     ref : np-small
     mounts:
      - mount_id: new_qg_hpvsopbasessh
        mountpoint: /newroot
        filesystem: ext4
        size: 10GB
   - name: qg_passthrough
     ref: p-small
     mounts:
      - mountpoint: /qg_passthrough
```

   Note: The configuration yaml file should specify either the **`imagefile`** parameter, or the **`reporegfile`** parameter, but not both. For more information about the configurations for a Hyper Protect Virtual Server instance, see Virtual server configuration file.

2. You can use the **`hpvs vs list`** command to view the list of virtual servers. You can undeploy only those virtual servers (and the resources associated with the virtual server) that were created by using the **`hpvs deploy`** command, from this list.

3. Run the following command to undeploy the virtual server.

```
hpvs undeploy --config $HOME/hpvs/config/yaml/vs_hpvsopbasessh.yml
```

   A message is displayed stating that virtual server(s) and associated networks, storage, images, and repository will be deleted. You are prompted to enter either **`Yes`** or **`No`**. If you enter **`Yes`**, the command execution continues, otherwise it exits the command execution.
   When you have a large number of virtual servers to undeploy, you can use the following flags to simplify the undeploy operation.

    - **`--exclude`**: To exclude virtual servers from the undeploy operation. You can specify a single virtual server, or a comma separated list of virtual servers.
    - **`--include`**: To include the virtual servers from the undeploy operation. You can specify a single virtual server, or a comma separated list of virtual servers.
    - If you do not use the **`--exclude`** or **`--include`** flag, all virtual servers that are listed in the configuration yaml file are undeployed. The **`--exclude`** or **`--include`** flags are mutually exclusive and you must specify only one of them when you run the **`hpvs undeploy`** command.
4. You can run the following commands to verify if the resources associated with the virtual server or virtual servers are deleted.

    - **`hpvs image list`** command to verify if the images associated with the virtual server(s) are deleted.
    - **`hpvs network list`** command to verify if the networks associated with the virtual server(s) are deleted.
    - **`hpvs quotagroup list`** command to verify if the quotagroups associated with the virtual server(s) are deleted.
    - **`hpvs repository list`** command to verify if the repositories associated with the virtual server(s) are deleted.

    Note: The resources that are shared by other virtual servers are not deleted and an appropriate message is displayed.

Updating virtual servers

You can update the resources or configuration of a virtual server after the completion of the deploy operation by using the **`-u`**, or the **`--update`** flag of the **`hpvs deploy`** command. This flag is applicable for Hyper Protect Virtual Servers version 1.2.2, or later.

This procedure is intended for users with the role *cloud administrator* and *app developer or ISV*.

Procedure

The information about the parameters to be updated are read from the configuration yaml file. You can edit the configuration file with the details of the update you want to perform and use this configuration file to run the command. This flag is applicable for Hyper Protect Virtual Servers version 1.2.2, or later. Run the following command to update the virtual server instance.

```
hpvs deploy --update --config $HOME/hpvs/config/demo_byoi.yml
```

Note: It is recommended that you back up the Hyper Protect Virtual Server container by using the **`hpvs snapshot`** command before you run the **`hpvs deploy update`** command. For more information about the **`hpvs`** commands, see Commands in IBM Hyper Protect Virtual Server.

When you have a large number of virtual servers to update, you can use the following flags to simplify the deploy update operation.

- **`--exclude`**: To exclude virtual servers from the deploy update operation. You can specify a single virtual server, or a comma separated list of virtual servers.
- **`--include`**: To include the virtual servers from the deploy update operation. You can specify a single virtual server, or a comma separated list of virtual servers.
- If you do not use the **`--exclude`** or **`--include`** flag, all virtual servers that are listed in the configuration yaml file are updated. The **`--exclude`** or **`--include`** flags are mutually exclusive and you must specify only one of them when you run the **`hpvs deploy`** command along with the **`--update`** flag.

You can use the **`--update`** flag of the **`hpvs deploy`** command in the following scenarios:

- Increase the size of the mountpoint (you might need to increase the size of the quotagroup to accommodate the increase in mountpoint size).
- Update the repository definition file.
- Update the network by modifying the network config section in configuration yaml file. If the network not exist, a new network can be created with the specified details. Similarly, you can change an existing IP address.

You cannot use the `--update` flag of the `hpvs deploy` command in the following scenarios:

- Add a new mount ID, reduce the size of the mountpoint or reduce the size of the quotagroup.
- Detach a quotagroup (you cannot detach a quotagtoup by using the `hpvs vs update` command as well). Doing so might cause errors or lead to an irrecoverable state of the quotagroup and the virtual server.

For more information about the parameters that can be updated, see Updating the parameters of IBM Hyper Protect Virtual Servers.

Note:

- Networks that are detached when you run the `hpvs deploy` command by specifying the `--update` flag, are deleted if they not used by any other virtual server.
- You cannot update the settings of an existing network in the virtual server template file.

Uninstalling IBM Hyper Protect Virtual Servers

You can uninstall IBM Hyper Protect Virtual Servers. Note that you must back up your own workload first.

1. Uninstalling IBM Hyper Protect Virtual Servers CLI tools
2. Uninstalling Secure Service Container partitions

Uninstalling IBM Hyper Protect Virtual Servers CLI tools

You can uninstall IBM Hyper Protect Virtual Servers CLI tools and the modules included in the CLI tools.

This procedure is intended for users with the role *cloud administrator*.

Before you begin

- Ensure that you back up the Hyper Protect Virtual Server container by using the `hpvs snapshot` command. For more information about the `hpvs` command, see Commands in IBM Hyper Protect Virtual Server.

Procedure

Complete the following steps with root user authority.

1. To remove the virtual servers, run the `hpvs vs list` command to view the available virtual servers. Then you can delete each of the virtual servers by running the `hpvs vs delete` command.

2. To remove the networks, run the `hpvs network list` command to view the available networks. Then you can delete each of the networks by running the `hpvs network delete` command.

3. To remove the quotagroups, run the `hpvs quotagroup list` command to view the available quotagroups. Then you can delete each of the quotagroups by running the `hpvs quotagroup delete` command.

4. To remove the repositories, run the **`hpvs repository list`** command to view the available repositories. Then you can delete each of the repositories by running the **`hpvs repository delete`** command.

5. To remove the registries, run the **`hpvs registry list`** command to view the available registries. Then you can delete each of the registries by running the **`hpvs registry delete`** command.

6. To remove the images, run the **`hpvs image list`** command to view the available images. Then you can delete each of the images by running the **`hpvs image delete`** command.

7. To remove the hosts, run the **`hpvs host list`** command to view the available hosts. Then you can delete each of the hosts by running the **`hpvs host delete`** command.

Uninstalling Secure Service Container partitions

You can stop, deactivate, or delete the Secure Service Container partitions on the IBM Z or LinuxONE machine.

This procedure is intended for users with the role *system administrator*.

Before you begin

- Check whether your host system is running in standard mode (that is, with Processor Resource/System Manager or PR/SM) or has Dynamic Partition Manager (DPM) enabled.
- Check that you download Secure Service Container User's Guide, SC28-6978-02a.

Procedure

1. (Optional) Export the Secure Service Container configuration by following the description in section *Exporting or importing appliance configuration data* of chapter 14 "Using the Secure Service Container user interface".

2. Stop/deactivate or delete the Secure Service Container partition:

   - On a standard mode system, chapter 7 - Deactivating or deleting a Secure Service Container partition on a standard mode system.
   - On a DPM-enabled system, chapter 12 - Stopping or deleting a Secure Service Container partition on a DPM-enabled system.

Updating Hyper Protect Virtual Server containers

You can update IBM Hyper Protect Virtual Servers containers to use different resource settings, such as CPU, memory, or a different image tag.

This procedure is intended for users with the role *cloud administrator*.

Before you begin

- Ensure that you back up the IBM Hyper Protect Virtual Servers container by using the **`hpvs snapshot`** command. For more information about the **`hpvs`** commands, see Commands in IBM Hyper Protect Virtual Server.
- You cannot update the **`cryptoControl`**, and **`cryptoMatrix`** crypto parameters.
- You cannot update the name of the virtual server.

- For all parameters that accept "array" type, you must specify all the parameters when you run the **`hpvs vs update`** command. This happens because the other parameters that were specified during the **`hpvs vs create`** command are overwritten when you specify only one value when running **`hpvs vs update`** command.
- Running the **`hpvs vs update`** command will stop and restart the virtual server.
- To execute the **`hpvs vs update`** command, the **`mountID`** that is specified as the RUNQ_ROOTDISK must have the parameter **`reset_root = true`** set in its configuration. This applies to IBM Hyper Protect Virtual Servers version 1.2.2, or later.
- When you are using IBM Hyper Protect Virtual Servers version 1.2.4, or later, a new repository registration file must be created if the repositories have not yet been registered on the Secure Service Container LPAR. You can use the **`hpvs regfile create`** command to create the repository registration file.

Procedure

Complete the following steps with root user authority.

1. Get the IBM Hyper Protect Virtual Servers container name from the result of the **`hpvs vs list`** command.

2. You can update CPU, memory, or both settings for the IBM Hyper Protect Virtual Servers container. The following command example updates the Hyper Protect Virtual Server container **`testcontainer`** to use 4 CPU threads and 1024 MB memory.

   ```
   hpvs vs update --name testcontainer --repo HpvsopBase --cpu 4 --ram 1024 --tag
   1.2.2.1-release-abcdef
   ```

   Note: The following are applicable for versions of IBM Hyper Protect Virtual Servers earlier than 1.2.4.

   - Specify only the parameters that you want to update in the command.
   - If you had used quotagroup or environment variable (either as --env or --envjsonpath) parameters when creating the virtual server, then you must provide those values as parameters to the **`hpvs vs update`** command. For Hyper Protect Virtual Server version 1.2.2, or later, if you create a virtual server by specifying the **`--env`** parameter **`ROOT_SSH_KEY`**, and if you do not specify the **`--env`** variable during an update operation, you will not be able to connect to the virtual server instance by using the secure shell (SSH).
3. You can update the parameters of the IBM Hyper Protect Virtual Servers container. The following example updates the image tag to use a different image tag, for example the Secure Build virtual server image tag is updated to **`1.2.2.1-release-abcdef`**

   ```
   hpvs vs update --name securecontainer --repo SecureDockerBuild --tag 1.2.2.1-
   release-abcdef
   ```

   Note: The following are applicable for versions of IBM Hyper Protect Virtual Servers earlier than 1.2.4.

   - Specify only the parameters that you want to update in the command.
   - If you had used quotagroup or environment variable (either as --env or --envjsonpath) parameters when creating the virtual server, then you must provide those values as parameters to the **`hpvs vs update`** command. For Hyper Protect Virtual Server version 1.2.2, or later, if you create a virtual server by specifying the **`--env`** parameter **`ROOT_SSH_KEY`**, and if you do not specify the **`--env`** variable during an update operation, you will not be able to connect to the virtual server instance by using the secure shell (SSH).
4. You can update the network. The following example updates the IP address to 192.168.72.3.

   ```
   hpvs vs update --name testcontainer --repo HpvsopBaseSSH --tag 1.2.2.1-release-
   abcdef \
   --cpu 2 --ram 4096 --env={LOGTARGET=/dev/console,ROOTFS_LOCK=y,ROOT_SSH_KEY"$key"}
   \
   --quotagroup  "{quotagroup = myquotagroup , mountid = new,mount = /newroot,
   filesystem = ext4, size = 30GB}" \
   --network "{name = internal_net, ip = 192.168.72.3}"
   ```

Note: The environment key value is *key: ROOT_SSH_KEY* for IBM Hyper Protect Virtual Servers version 1.2.2, or later, and *key: SSH_PUBLIC_KEY* for IBM Hyper Protect Virtual Servers version 1.2.1.1, and 1.2.1.

5. You can specify the optional parameters **`RUNQ_ROOTDISK`** and **`reset_root`** for IBM Hyper Protect Virtual Servers Version 1.2.2, or later. A dedicated root-disk can be assigned to a virtual server in the environment variables by using the mount_id of disks (mounts) that are assigned to a virtual server from the available quotagroup. RUNQ_ROOTDISK works for both passthrough and non passthrough quotagroups. However, the parameter **`"reset_root=true"`** is supported only for non passthrough quotagroups.

   ```
   hpvs vs update --name testcontainer --repo HpvsopBase --cpu 4 --ram 1024 --tag
   1.2.2.1-release-abcdef --env=
   {LOGTARGET=/dev/console,ROOTFS_LOCK=y,RUNQ_ROOTDISK=new} --quotagroup "{quotagroup
   = secondvol, mountid = new,mount = /newroot, filesystem = ext4, size = 3GB,
   reset_root = true}"
   ```

   For more information about the **`hpvs vs list`** command, or the **`hpvs vs update`** command, see Commands in IBM Hyper Protect Virtual Servers. For more information about the parameters that can be updated, see Updating the parameters of IBM Hyper Protect Virtual Servers.

Upgrading IBM Hyper Protect Virtual Servers

You can upgrade IBM Hyper Protect Virtual Servers from earlier versions to version 1.2.4 by downloading the latest version from IBM Passport Advantage or IBM Fix Central. You can follow the procedure detailed in this topic, by using the latest images available in the version 1.2.4.

This procedure is intended for users with the role *system administrator*.

Before you begin

- Ensure that you back up all the workload and configuration data. It is recommended that you take a snapshot of the virtual servers so that you can revert to them in case you face any issues in the future. For more information, see Backing up and restoring IBM Hyper Protect Virtual Servers.
- Ensure that you have stopped all the running Hyper Protect Virtual Server containers. For more information, see Commands for Hyper Protect Virtual Server Containers.
- When you are using IBM Hyper Protect Virtual Servers version 1.2.4, or later, a new repository registration file must be created if the repositories have not yet been registered on the Secure Service Container LPAR. You can use the **`hpvs regfile create`** command to create the repository registration file.

Procedure

Complete the following steps on your x86 or Linux on IBM Z/LinuxONE (i.e., s390x architecture) management server.

1. Download the latest IBM Hyper Protect Virtual Servers. For more information, see Downloading IBM Hyper Protect Virtual Servers.

2. You should move the existing **`$HOME/hpvs`** directory to a new location. Then, delete the /usr/local/bin/hpvs directory. Ensure that you take a backup of all the configuration yaml files, keys, and enc files. Execute the setup script which provides an automated procedure that simplifies the installation and configuration of the IBM Hyper Protect Virtual Servers environment. For more information, see Setting up the IBM Hyper Protect Virtual Servers environment

3. To backup data from IBM Hyper Protect Virtual Servers, complete the following steps.

   1. Login to the User Interface of the Secure Service Container by using the IP address of the Secure Service Container.

    2. In the left navigation pane, click the **Ex-/Import** icon.
    3. Click the **Export** button and save the configuration (metadata) to your workstation.
       Note:
       - This data is critical and should be stored carefully.
       - You must ensure that the Secure Service Container is not used for any other purposes during the upgrade process.
4. To upgrade the Hosting Appliance to 4.3.5 (the same procedure can be followed to upgrade the Hosting Appliance from any lower version to a higher version), complete the following steps.

    1. Login to the User Interface of the Secure Service Container by using the IP address of the Secure Service Container.
    2. In the left navigation pane, click the **Maintenance** icon.
    3. To move the Hosting Appliance into the installer mode, click the **Installer** button.
    4. The Hosting Appliance reboots into the installer mode and prompts you to upload the Hosting Appliance image.
    5. Upload the "secure-service-virtual server-for-hpvs.appliance.4.3.5.img.gz" from the installation folder. When the Hosting Appliance completes installation and reboots, it will be upgraded to version 4.3.5.
       Note: You can skip this step if you are upgrading from version 1.2.2 to 1.2.2.1.
5. To restore the data that was backed up in step 3, to the Hosting Appliance, complete the following steps.

    1. Login to the User Interface of the Secure Service Container by using the IP address of the Secure Service Container.
    2. In the left navigation pane, click the **Ex-/Import** icon.
    3. Click the **Upload** button and upload the configuration (metadata) to the Hosting Appliance. This restores all your configuration details such as networks, quotagroups and virtual servers back onto the Hosting Appliance.
    4. It is recommended that you take a snapshot of the virtual servers so that you can revert to them in case you face any issues in the future.
6. To validate the upgrade, complete the following steps on your x86 or Linux on IBM Z/LinuxONE (i.e., s390x architecture) management server.

    1. Start all virtual servers that you had stopped before the upgrade task.
    2. Verify whether all the virtual servers are online and in the running state by using the `hpvs vs list` command.
    3. Verify whether the Hosting Appliance Configuration is displayed as version 4.3.5 (depends on the Hosting Appliance version you are upgrading to) in the User Interface of the Secure Service Container.
    4. Verify whether the configuration details of the image, repository, registry, and quotagroup are as you expect them to be. You can verify this by running the `hpvs image list`, `hpvs repository list`, `hpvs registry list`, and `hpvs quotagroup list` commands, in that order.
    5. Verify whether you can access the virtual servers and run workloads.
7. After you validate that the virtual servers are online and in the running state, to upgrade the virtual servers, you can update the base images that are used by the virtual servers. See the section "Updating the virtual servers" for information about updating the images used by the different virtual servers.

8. After you complete the update of the virtual servers, you can take a backup and remove the folders that you do not require.

9. If there are failures during the upgrade process, see the section "Rollback" in case of update failures.

Updating the virtual servers images

You can use the `hpvs deploy` command and specify the `-u` or the `--update` flag and the configuration yaml file to update the images of the virtual servers. You can edit the configuration yaml file to specify the new image tag to be used during the virtual server deploy update process and use this configuration file to run the command.

Note:

- When using IBM Hyper Protect Virtual Servers, it is recommended that you have a RUNQ_ROOTDISK with ext4 filesystem for deploying and updating virtual servers. If your virtual server requires a dedicated root disk, then before you update the virtual server, you must define the RUNQ_ROOTDISK variable in environment value.
- If your virtual server is not using the RUNQ_ROOTDISK feature, then the root disk is created as an overlay on the existing appliance_data quotagroup. As the default size of the appliance_data quotagroup is 10GB, it is recommended that you extend the size of this quotagroup based on your virtual server workload, to ensure proper functioning of your workloads.
- When you update the virtual server with images that are at version 1.2.2, or later, by using a dedicated root disk (RUNQ_ROOTDISK/reset_root feature), you should use `reset_root: true` so that the virtual server can be reinitialized from the provided image tag. However, before you upgrade it is recommended that you add your data or applications to another mountpoint of the virtual server such as `/data` and not use `/newroot` as the mountpoint. Otherwise, you might lose data that was created in '/' when you set "reset_root: true" in the configuration yaml file that you use for the virtual server update operation.

Updating the HpvsopBaseSSH virtual server image

Complete the following steps on your x86 or Linux on IBM Z/LinuxONE (i.e., s390x architecture) management server with root user authority.

1. Copy the configuration files used for the virtual server deployment from your back up folder to your hpvs directory (/root/hpvs). For example, if you moved */$HOME/hpvs* to */$HOME/hpvs_123* before executing setup.sh in step 2, after executing setup.sh copy the configuration file and the virtualserver template file by running the following commands. Also, you must copy the ssh key-pair to same folder (according to the key location as indicated in the `vs_hpvsopbasessh.yml` file) so that you can log in to the virtual server after the update.

   ```
   cp /$HOME/hpvs_123/config/hpvsopbasessh/vs_hpvsopbasessh.yml 
   /$HOME/hpvs/config/hpvsopbasessh/vs_hpvsopbasessh_124.yml
   cp /$HOME/hpvs_123/config/templates/virtualserver.template.yml 
   /$HOME/hpvs/config/templates/virtualserver.template.yml
   ```

   Note: For versions of IBM Hyper Protect Virtual Servers earlier than 1.2.4, you cannot use the old hosts file and must create a new one by using the `hpvs host add` command.

2. Choose one of the options to perform the update.

   - Update by using a dedicated root disk. This method is recommended.

     - Update or append the configuration yaml file to specify the `imagetag`, and `imagefile` parameters with the 1.2.4 image. For example:

       ```
       imagetag: 1.2.4-release-d0651e4
       imagefile: HpvsopBaseSSH.tar.gz
       environment:
       - key: RUNQ_ROOTDISK
         value: newroot    // Value is the mount_id of the root disk of your 
       virtual server
       ```

       For more information about the configuration file and updating the virtual server instance, see Creating a Hyper Protect Virtual Server instance.

     - Run the following command to update the virtual server instance.

       ```
       hpvs deploy --update --config 
       /$HOME/hpvs/config/hpvsopbasessh/vs_hpvsopbasessh_124.yml
       ```

       After this command completes execution, the virtual server is the running state with the root disk parameter set in the virtual server profile (the virtual server is still running with the earlier base image).

- Update or append the environment variable for *ROOT_SSH_KEY* and set the value *reset_root: true* in the mounts section of the root disk of the virtual server. For example:

  ```
  environment:
    - key: RUNQ_ROOTDISK
      value: newroot
    - key: ROOT_SSH_KEY
      value: "@/root/hpvs/config/hpvsopbasessh/keys/id_rsapub_base64.cert"
  volumes:
    - name: qg_hpvsopbasessh
      ref : np-medium
      mounts:
      - mount_id: newroot
        mountpoint: /newroot
        filesystem: ext4
        size: 10GB
        reset_root: true        #This flag will reset the newroot root
  disk
  ```

  Note: In IBM Hyper Protect Virtual Servers 1.2.2, the SSH key of the base image is changed from *SSH_PUBLIC_KEY* to *ROOT_SSH_KEY*, and the value must be passed in the base64 format.

- Run the following command to update the virtual server instance.

  ```
  hpvs deploy --update --config
  /$HOME/hpvs/config/hpvsopbasessh/vs_hpvsopbasessh_124.yml
  ```

  After this command completes execution, the virtual server is the running state with the root disk parameter set in the virtual server profile. The virtual server is now running with the 1.2.4 base image.

- To verify whether the virtual server instance is upgraded (after a successful upgrade, the virtual server image must point to the latest version), run the following command.

  ```
  hpvs vs list
  ```

- Update without the root disk.

  - Update or append the configuration yaml file to specify the **`imagetag`**, and **`imagefile`** parameters with the 1.2.4 image, and the key parameter of the environment variable as *ROOT_SSH_KEY*. For example:

    ```
    imagetag: 1.2.4-release-d0651e4
    imagefile: HpvsopBaseSSH.tar.gz
    environment:
    - key: ROOT_SSH_KEY
      value: "@/root/hpvs/config/hpvsopbasessh/keys/id_rsapub_base64.cert"
    ```

    Note: In IBM Hyper Protect Virtual Servers 1.2.2, the SSH key of the base image is changed from *SSH_PUBLIC_KEY* to *ROOT_SSH_KEY*, and the value must be passed in the base64 format.

  - Run the following command to update the virtual server instance.

    ```
    hpvs deploy --update --config
    /$HOME/hpvs/config/hpvsopbasessh/vs_hpvsopbasessh_124.yml
    ```

    After this command completes execution, the virtual server is in the running state with the root disk parameter set in the virtual server profile. The virtual server is now running with the 1.2.4 base image).

  - To verify whether the virtual server instance is upgraded (after a successful upgrade, the virtual server image must point to the latest version), run the following command.

    ```
    hpvs vs list
    ```

Updating the HpvsopBase virtual server image

Complete the following steps on your x86 or Linux on IBM Z/LinuxONE (i.e., s390x architecture) management server with root user authority.

1. Copy the configuration files used for the virtual server deployment from your back up folder to your hpvs directory (/root/hpvs). For example, if you moved */$HOME/hpvs* to */$HOME/hpvs_123* before executing setup.sh in step 2, after executing setup.sh copy the configuration file and the virtualserver template file by running the following commands.

   ```
   cp /$HOME/hpvs_123/config/hpvsopbase/vs_hpvsopbase.yml 
   /$HOME/hpvs/config/hpvsopbase/vs_hpvsopbase_124.yml
   cp /$HOME/hpvs_123/config/templates/virtualserver.template.yml 
   /$HOME/hpvs/config/templates/virtualserver.template.yml
   ```

   Note: For versions of IBM Hyper Protect Virtual Servers earlier than 1.2.4, you cannot use the old hosts file and must create a new one by using the `hpvs host add` command.

2. Choose one of the options to perform the update.

   - Update by using a dedicated root disk. This method is recommended.

     - Update or append the configuration yaml file to specify the `imagetag`, and `imagefile` parameters with the 1.2.4 image For example:

       ```
       imagetag: 1.2.4-release-d0651e4
       imagefile: HpvsopBase.tar.gz
       environment:
       - key: RUNQ_ROOTDISK
         value: newroot    // Value is the mount_id of the root disk of your 
       virtual server
       ```

       For more information about the configuration file and updating the virtual server instance, see [Creating a Hyper Protect Virtual Server instance](#).

     - Run the following command to update the virtual server instance.

       ```
       hpvs deploy --update --config 
       /$HOME/hpvs/config/hpvsopbase/vs_hpvsopbase_124.yml
       ```

       After this command completes execution, the virtual server is the running state with the root disk parameter set in the virtual server profile (the virtual server is still running with the earlier base image).

     - Set the value *reset_root: true* in the mounts section of the root disk of the virtual server. For example:

       ```
       environment:
       - key: RUNQ_ROOTDISK
         value: newroot
       volumes:
        - name: qg_hpvsopbasessh
          ref : np-medium
          mounts:
            - mount_id: newroot
              mountpoint: /newroot
              filesystem: ext4
              size: 10GB
              reset_root: true        #This flag will reset the newroot root 
       disk
       ```

     - Run the following command to update the virtual server instance.

       ```
       hpvs deploy --update --config 
       /$HOME/hpvs/config/hpvsopbasessh/vs_hpvsopbasessh_124.yml
       ```

    After this command completes execution, the virtual server is the running state with the root disk parameter set in the virtual server profile. The virtual server is now running with the 1.2.4 base image.

  - To verify whether the virtual server instance is upgraded (after a successful upgrade, the virtual server image must point to the latest version), run the following command.

    ```
    hpvs vs list
    ```

- Update without the root disk.

  - Update or append the configuration yaml file to specify the **`imagetag`**, and **`imagefile`** parameters with the 1.2.4 image. For example:

    ```
    imagetag: 1.2.4-release-d0651e4
    imagefile: HpvsopBase.tar.gz
    ```

  - Run the following command to update the virtual server instance.

    ```
    hpvs deploy --update --config
    /$HOME/hpvs/config/hpvsopbase/vs_hpvsopbase_124.yml
    ```

    After this command completes execution, the virtual server is in the running state with the root disk parameter set in the virtual server profile. The virtual server is now running with the 1.2.4 base image.

  - To verify whether the virtual server instance is upgraded (after a successful upgrade, the virtual server image must point to the latest version), run the following command.

    ```
    hpvs vs list
    ```

Updating the Secure Build virtual server image

Complete the following steps on your x86 or Linux on IBM Z/LinuxONE (i.e., s390x architecture) management server with root user authority.

1. Copy the configuration file used for the virtual server deployment from your back up folder to your hpvs directory (/root/hpvs). For example, if you moved */$HOME/hpvs* to */$HOME/hpvs_123* before executing setup.sh in step 2, after executing setup.sh copy the configuration file and the virtualserver template file by running the following commands. Also, you must copy the SecureBuild certificate and key to the same folder (according to the key location as indicated in the **`vs_securebuild.yml`** file) so that you can securely communicate with Secure Build server after the update.

   ```
   cp /$HOME/hpvs_123/config/securebuild/vs_securebuild.yml
   /$HOME/hpvs/config/securebuild/vs_securebuild_124.yml
   cp /$HOME/hpvs_123/config/templates/virtualserver.template.yml
   /$HOME/hpvs/config/templates/virtualserver.template.yml
   ```

   Note: For versions of IBM Hyper Protect Virtual Servers earlier than 1.2.4, you cannot use the old hosts file and must create a new one by using the **`hpvs host add`** command.

2. Choose one of the options to perform the update.

   - Update by using a dedicated root disk. This method is recommended.

     - Update or append the configuration yaml file to specify the **`imagetag`**, and **`imagefile`** parameters with the 1.2.4 image. For example:

       ```
         imagetag: 1.2.4-release-f78a642
         imagefile: SecureDockerBuild.tar.gz
         environment:
         - key: RUNQ_ROOTDISK
       ```

```
    value: newroot    // Value is the mount_id of the root disk of your
virtual server
```

For more information about the configuration file and updating the Secure Build virtual server instance, see Building your application with the Secure Build virtual server.

- Run the following command to update the Secure Build virtual server.

  ```
    hpvs deploy --update --config
  /$HOME/hpvs/config/securebuild/vs_securebuild_124.yml
  ```

  After this command completes execution, the virtual server is the running state with the root disk parameter set in the virtual server profile (the virtual server is still running with the earlier base image).

- Set the value *reset_root: true* in the mounts section of the root disk of the virtual server. For example:

  ```
    environment:
    - key: RUNQ_ROOTDISK
      value: newroot
    volumes:
    - name: securebuild_qg
      ref : np-medium
      mounts:
       - mount_id: newroot
         mountpoint: /newroot
         filesystem: ext4
         size: 10GB
         reset_root: true        #This flag will reset the newroot root
  disk
  ```

- Run the following command to update the Secure Build virtual server.

  ```
    hpvs deploy --update --config
  /$HOME/hpvs/config/securebuild/vs_securebuild_124.yml
  ```

  After this command completes execution, the Secure Build virtual server is the running state with the root disk parameter set in the virtual server profile. The Secure Build virtual server is now running with the 1.2.4 base image).

- To verify whether the Secure Build virtual server is upgraded (after a successful upgrade, the virtual server image must point to the latest version), run the following command.

  ```
    hpvs vs list
  ```

- Update without the root disk.

  - Update or append the configuration yaml file to specify the **`imagetag`**, and **`imagefile`** parameters with the 1.2.4 image. For example:

    ```
    imagetag: 1.2.4-release-f78a642
    imagefile: SecureDockerBuild.tar.gz
    ```

  - Run the following command to update the Secure Build virtual server.

    ```
    hpvs deploy --update --config
    /$HOME/hpvs/config/securebuild/vs_securebuild_124.yml
    ```

    After this command completes execution, the Secure Build virtual server is in the running state with the root disk parameter set in the virtual server profile. The Secure Build virtual server is now running with the 1.2.4 base image.

- To verify whether the Secure Build virtual server is upgraded (after a successful upgrade, the virtual server image must point to the latest version), run the following command.

  ```
  hpvs vs list
  ```

3. When you upgrade from IBM Hyper Protect Virtual Servers version 1.2.3 to IBM Hyper Protect Virtual Servers version 1.2.4, complete the following steps.

   - Run the following command to update the Secure Build virtual server. Starting with IBM Hyper Protect Virtual Servers version 1.2.4, the term "whitelist" is replaced with "allowlist", therefore you must update the **secure_build.yml** file before you run the following command.

     ```
     hpvs sb update --config secure_build.yml
     ```

   - Run the following command to rebuild the Secure Build virtual server.

     ```
     hpvs sb build --config secure_build.yml
     ```

   - Run the following command to regenerate the repository registration file.

     ```
     hpvs sb regfile --config secure_build.yaml --out
     /root/hpvs/encryptedRegfile.enc
     ```

   - Run the following command to update the repository.

     ```
     hpvs repository update --pgp=/root/hpvs/encryptedRegfile.enc --id=<repo id>
     ```

Updating the monitoring and collectd virtual server images

Complete the following steps on your x86 or Linux on IBM Z/LinuxONE (i.e., s390x architecture) management server with root user authority.

1. Copy the configuration files used for the virtual server deployment from your back up folder to your hpvs directory (/root/hpvs). For example, if you moved */$HOME/hpvs* to */$HOME/hpvs_123* before executing setup.sh in step 2, after executing setup.sh copy the configuration file and the virtualserver template file by running the following commands. Also, you must copy the keys and certificates which are required to the same folder (according to the certificate location as indicated in the **vs_monitoring.yml** file).

   ```
   cp /$HOME/hpvs_123/config/monitoring/vs_monitoring.yml
   /$HOME/hpvs/config/monitoring/vs_monitoring_124.yml
   cp /$HOME/hpvs_123/config/templates/virtualserver.template.yml
   /$HOME/hpvs/config/templates/virtualserver.template.yml
   ```

   Note: For versions of IBM Hyper Protect Virtual Servers earlier than 1.2.4, you cannot use the old hosts file and must create a new one by using the **hpvs host add** command.

2. Update or append the configuration yaml file to specify the **imagetag**, and **imagefile** parameters with the 1.2.4 image. For example:

   ```
   - name: test-monitoring
     imagetag: 1.2.4
     imagefile: Monitoring.tar.gz
     .
     .
   - name: test-collectd
       imagetag: 1.2.4
       imagefile: CollectdHost.tar.gz
   ```

   For more information about the configuration file updating the collectd and monitoring virtual servers, see Working with Monitoring virtual servers.

3. Run the following command to update the collectd and monitoring virtual servers.

   ```
   hpvs deploy --update --config /$HOME/hpvs/config/monitoring/vs_monitoring_124.yml
   ```

After this command completes execution, the monitoring and collectd virtual servers are in the running state with the latest image.

4. To verify whether the collectd and monitoring virtual server are upgraded (after a successful upgrade, the virtual server image must point to the latest version), run the following command.

   ```
   hpvs vs list
   ```

Updating the GREP11 virtual server image

Complete the following steps on your x86 or Linux on IBM Z/LinuxONE (i.e., s390x architecture) management server with root user authority.

1. Copy the configuration files used for the virtual server deployment from your back up folder to your hpvs directory (/root/hpvs). For example, if you moved */$HOME/hpvs* to */$HOME/hpvs_123* before executing setup.sh in step 2, after executing setup.sh copy the configuration file and the virtualserver template file by running the following commands. Also, you must copy the keys and certificates which are required to the same folder (according to the certificate location as indicated in the **vs_grep11.yml** file).

   ```
   cp /$HOME/hpvs_123/config/grep11/vs_grep11.yml
   /$HOME/hpvs/config/grep11/vs_grep11_124.yml
   cp /$HOME/hpvs_123/config/templates/virtualserver.template.yml
   /$HOME/hpvs/config/templates/virtualserver.template.yml
   ```

   Note: For versions of IBM Hyper Protect Virtual Servers earlier than 1.2.4, you cannot use the old hosts file and must create a new one by using the **hpvs host add** command.

2. Update or append the configuration yaml file to specify the **imagetag**, and **imagefile** parameters with the 1.2.4 image. For example:

   ```
   - name: test-grep11
     imagetag: 1.2.4
     imagefile: hpcsKpGrep11_runq.tar.gz
     environment:
     ...
      - key: "EP11SERVER_EP11CRYPTO_ENABLED"       # add this key in 1.2.2, or later
        value: "true"
     ...
   ```

   For more information about the configuration file updating the GREP11 virtual servers, see Working with GREP11 virtual servers.

3. Run the following command to update the GREP11 virtual server.

   ```
   hpvs deploy --update --config /$HOME/hpvs/config/grep11/vs_grep11_124.yml
   ```

   After this command completes execution, the GREP11 virtual server is in the running state with the latest image.

4. To verify whether the GREP11 virtual server is upgraded (after a successful upgrade, the virtual server image must point to the latest version), run the following command.

   ```
   hpvs vs list
   ```

   You can also verify the upgrade by using the **hpvs vs log** command. The following is an example to verify the upgrade.

   ```
   hpvs vs log --name test-grep11
   ```

   You might see a display similar to the following in the log file.

   ```
   1.2.2 -> Starting GREP11 server [v2.3.0]   module=entry
   1.2.2.1 -> Starting GREP11 server [v2.3.0-19-ga5827e0a]  module=entry
   1.2.3 -> Starting GREP11 server [v2.3.83]              module=entry
   ```

The following are some limitations related to upgrading virtual servers

- Upgrading a virtual server with a RUNQ_ROOTDISK, by defining a dedicated root disk, is supported only for the ext4 filesystem.
- Upgrading virtual servers from version 1.2.0 to 1.2.2 is not supported.

Rollback in case of update failure

1. Complete the following steps on your x86 or Linux on IBM Z/LinuxONE (i.e., s390x architecture) management server, to roll back in case of an update failure.

    1. Take a back up of the current working CLI directory. For example, */root/hpvs* is backed up to */root/hpvs_124*.
    2. Move the old working CLI directory back to original state. For example, move */root/hpvs_123* to */root/hpvs*.
    3. You can later choose to delete the folder */root/hpvs_124*.
2. To roll back the Hosting Appliance from version 4.3.5 to version 3.17.0 (you can follow the same procedure for earlier versions of the Hosting Appliance), complete the following steps.

    1. Login to the User Interface of the Secure Service Container by using the IP address of the Secure Service Container.
    2. In the left navigation pane, click the **Maintenance** icon.
    3. To move the Hosting Appliance into the installer mode, click the **Installer** button.
    4. The Hosting Appliance reboots into the installer mode and prompts you to upload the Hosting Appliance image.
    5. Upload the "secure-service-virtual server-for-hpvs.appliance.3.17.0.img.gz" from the installation folder. When the Hosting Appliance completes installation and reboots, it will be rolled back to version 3.17.0.
    Note: You can skip this step if you are upgrading from version 1.2.2 to 1.2.2.1.
3. a. To restore the data that was backed up from the IBM Hyper Protect Virtual Servers, step 3, complete the following steps.

    1. Login to the User Interface of the Secure Service Container by using the IP address of the Secure Service Container.
    2. In the left navigation pane, click the **Ex-/Import** icon.
    3. Click the **Upload** button and upload the configuration to the Hosting Appliance.

    b. To restore the data that was backed up from the IBM Hyper Protect Virtual Servers, complete the following steps.

    1. Update the configuration yaml file with the image tag you want to revert to (previous image version) and run the `hpvs deploy --update` command with the updated configuration file.
    2. (Optional) Use the `hpvs snapshot restore` command to restore the Virtual Servers with the previous data (note that the virtual server will now be running with the reverted or rolled back image version).

Note: Rollback is successful only if the contents of virtual server are not changed after the upgrade, and if the snapshots taken for backup are not deleted at any point of time. You could choose to take a backup of your virtual server disks before the upgrade, as this could help you restore the data in case the data changed. For more information, see Backing up and restoring IBM Hyper Protect Virtual Servers.

Upgrading IBM Hyper Protect Virtual Servers from 1.2.0 or 1.2.0.1 to 1.2.1

You can upgrade IBM Hyper Protect Virtual Servers from Version 1.2.0 or 1.2.0.1 to Version 1.2.1 by downloading the latest version from IBM Passport Advantage or IBM Fix Central.

This procedure is intended for users with the role *system administrator*.

Before you begin

- Ensure that you back up all the workload and configuration data. It is recommended that you take a snapshot of the virtual servers so that you can revert to them in case you face any issues in the future. For more information, see Backing up and restoring IBM Hyper Protect Virtual Servers.
- Ensure that you have stopped all the running Hyper Protect Virtual Server containers. For more information, see Commands for Hyper Protect Virtual Server Containers.

Procedure

Complete the following steps on your x86 or Linux on IBM Z/LinuxONE (i.e., s390x architecture) management server with root user authority.

1. Download the latest IBM Hyper Protect Virtual Servers. For more information, see Downloading IBM Hyper Protect Virtual Servers.

2. Execute the setup script which provides an automated procedure that simplifies the installation and configuration of the IBM Hyper Protect Virtual Servers environment. For more information. The setup script creates the **`$HOME/hpvs`** directory structure and copies all the keys, and all the required config files and creates symbolic links of the images to this directory. For more information, see Setting up the IBM Hyper Protect Virtual Servers environment

3. To backup data from IBM Hyper Protect Virtual Servers 1.2.0.1, or 1.2.0, complete the following steps.

    1. Login to the User Interface of the Secure Service Container by using the IP address of the Secure Service Container.
    2. In the left navigation pane, click the **Ex-/Import** icon.
    3. Click the **Export** button and save the configuration (metadata) to your workstation. (**Note**: This data is critical and should be stored carefully.)
4. To upgrade the Hosting Appliance from Version 1.1.18 to Version 3.11.4, complete the following steps.

    1. Login to the User Interface of the Secure Service Container by using the IP address of the Secure Service Container. .
    2. In the left navigation pane, click the **Maintenance** icon.
    3. To move the Hosting Appliance into the installer mode, click the **Installer** button.
    4. The Hosting Appliance reboots into the installer mode and prompts you to upload the Hosting Appliance image.
    5. Upload the "secure-service-virtual server-for-hpvs.appliance.3.11.4.img.gz" from the installation folder. When the Hosting Appliance completes installation and reboots, it will be upgraded to Version 3.11.4.

    Note: If the upgrade of the Hosting Appliance from Version 1.1.18 to Version 3.11.4 fails for any reason, to roll back to the earlier state, install Hosting Appliance Version 1.1.18 by following the instructions in Installing the Hyper Protect Virtual Servers hosting appliance. Then import the data that you had exported and saved in step 3.

5. To restore the data that was backed up in step 3, to the Hosting Appliance, complete the following steps.

    1. Login to the User Interface of the Secure Service Container by using the IP address of the Secure Service Container.
    2. In the left navigation pane, click the **Ex-/Import** icon.
    3. Click the **Upload** button and upload the configuration (metadata) to the Hosting Appliance. This restores all your configuration details such as networks, quotagroups and virtual servers back onto the

Hosting Appliance.

4. It is recommended that you take a snapshot of the virtual servers so that you can revert to them in case you face any issues in the future.

Note: After you successfully completed the upgrade of the Hosting Appliance to Version 3.11.4 (in step 4), if restoring the data that was backed up in step 3 to the Hosting Appliance fails for any reason, and you have not made any changes like creating new virtual servers in Hosting Appliance Version 3.11.4, to roll back to the earlier state, install Hosting Appliance Version 1.1.18 by following the instructions in Installing the Hyper Protect Virtual Servers hosting appliance. Then import the data that you had exported and saved in step 3.

6. To validate the upgrade, complete the following steps

   1. Start all virtual servers that you stopped before you started the upgrade task.
   2. Verify whether all the virtual servers are online and running by using the `hpvs vs list` command.
   3. Verify whether the Hosting Appliance Configuration is displayed as Version 3.11.4 in the User Interface of the Secure Service Container.
   4. Verify whether the image, repository, registry, and quotagroup configuration details of the virtual servers are as you expect them to be by running the `hpvs image list`, `hpvs repository list`, `hpvs registry list`, and `hpvs quotagroup list` commands.
   5. Verify whether you can access the virtual servers and can execute commands.

The following are some limitations related to upgrading virtual servers

- The repository registration files that were created with IBM Hyper Protect Virtual Servers Version 1.2.0, or 1.2.0.1 are not supported on the Hosting Appliance Version 3.11.4 because of design changes. All old repository registration files must be deleted and you must create new repository registration files and register them with the Hosting Appliance 3.11.4 in IBM Hyper Protect Virtual Servers Version 1.2.1.
- virtual servers that were created by using the HpvsopBase or HpvsopBaseSSH as parent images, and deployed on IBM Hyper Protect Virtual Servers version 1.2.0, or 1.2.0.1 cannot be upgraded to IBM Hyper Protect Virtual Servers version 1.2.1.
- Upgrading Secure Build virtual servers from IBM Hyper Protect Virtual Servers Version 1.2.0, or 1.2.0.1 to IBM Hyper Protect Virtual Servers Version 1.2.1 is not supported. You must delete the Secure Build Server and its repository. Then follow the instructions in Building your application with the Secure Build virtual server to create and deploy Secure Build virtual servers.
- Upgrading GREP11 virtual servers from IBM Hyper Protect Virtual Servers Version 1.2.0, or 1.2.0.1 to IBM Hyper Protect Virtual Servers Version 1.2.1 is not supported. You must delete the GREP11 virtual servers, and then follow the instructions in Creating the GREP11 container to create and deploy GREP11 virtual servers.

Updating the collectd and monitoring virtual servers images

You can use the `hpvs vs update` command to update the images of the collectd and monitoring virtual servers.

1. Upload the collectd image to the Secure Service Container partition by using the `hpvs image load` command.

   ```
   hpvs image load --file=~/hpvs/config/monitoring/images/CollectdHost.tar.gz
   ```

2. Upload the monitoring image to the Secure Service Container partition by using the `hpvs image load` command.

   ```
   hpvs image load --file=~/hpvs/config/monitoring/images/Monitoring.tar.gz
   ```

3. Update the collectd container by using the `hpvs vs update` command.

   ```
   hpvs vs update --name collectd-host --repo CollectdHost --tag 1.2.1 --hostname
   collectd-host-container
   ```

Note: In IBM Hyper Protect Virtual Server Version 1.2.1, there is no change in the collectd image and therefore an upgrade is not required. Generally, between Hyper Protect Virtual Server releases, if the image of a virtual server does not change, then an upgrade is not required for that virtual server.

4. Update the monitoring container by using the **hpvs vs update** command.

```
hpvs vs update --name monitoring-host --repo Monitoring --tag 1.2.1 --hostname
monitoring-host-container --ports "{containerport = 8443, protocol = tcp, hostport
= 8443}" --ports "{containerport = 25826, protocol = udp, hostport = 25826}" --
envjsonpath ~/hpvs/config/env.json
```

You can use the same certificates and the **env.json** file that you used earlier when you created the virtual server.
For more information on updating virtual servers, see Updating Hyper Protect Virtual Server containers

Troubleshooting IBM Hyper Protect Virtual Servers

Refer to the following information for troubleshooting issues with IBM Hyper Protect Virtual Servers version 1.2.1, or later.

- For Secure Service Container partitions, use the Secure Service Container user interface to view the logs.
- For components such as Secure Build containers, use the **hpvs sb log** command to retrieve the build logs, or **hpvs sb status** command to check the progress of the Secure Build.
- For the command line tools provided by the product, add **--debug** to view the detailed log.
- All output from the command line is recorded in **$HOME/hpvs/logs/**.
- For the commands that require a yaml configuration file, check the formatting of the yaml file by comparing with the example yaml file in the **$HOME/hpvs/config** directory of each command.
- To know more about the errors you might encounter, see Error messages in Hyper Protect Virtual Servers.
- Use the mustgather script to collect debugging information when you want to open a support ticket. For more information, see Gathering Information for IBM Support.

Known issues with IBM Hyper Protect Virtual Servers 1.2.3

ERROR: Failed to pull image

If you see this error when running the **hpvs image pull** command or during the deployment of virtual servers, it might be because there is insufficient space in the **appliance_data** quotagroup. You can check the available size of the **appliance_data** quotagroup by using the command **hpvs quotagroup show --name appliance_data**. Ensure that the available size is larger than the image size. Increase the available size of the **appliance_data** quotagroup by using the command **hpvs quotagroup update --size**

ERROR: Failed to SSH to the HPVS container when both external and private networks exists

If you see this error, it might be because the default gateway is set to the private network. When you are setting up the network configuration and to setup the default gateway to the external network, specify the external network name as the first entry in the lexicographic order. Otherwise, access the container by using SSH fails. For example:

```
external
a_encf100_network
```

```
internal
b_encf960_network
```

ERROR: HVS-VSUD003 Update virtual server failed due to wrong quotagroup configuration

If you see this error, it might be because an invalid parameter was specified in the quotagroup configuration. You can provide valid quotagroup configuration details and retry the command. Note: The parameter **`reset_root`** flag is not supported in IBM Hyper Protect Virtual Servers version 1.2.2, for updating a virtual server by using the **`hpvs vs update`** command. You can use the **`hpvs deploy -u`** command to update the virtual server with the parameter **`reset_root:true`**.

HPVS container hangs or secure shell (SSH) access fails

If a running container hangs or SSH access to the container fails, check if the non-passthrough quotagroup that is used for the root disk (RUNQ_ROOTDISK) has enough available size. The available size should be set to more than 5 GB. You can use the **`hpvs quotagroup update`** command to increase the available size.

The "hpvs vs `**`" command failed

If you see the following issue:

```
$ hpvs vs * * command is throwing error like
ERROR: HVS-<Error code> API Error: .......: no space left on device 500
```

Then, check the **`appliance_data`** quotagroup size.

```
$ sudo hpvs quotagroup show --name appliance_data
+-------------+----------------+
| PROPERTIES  | VALUES         |
+-------------+----------------+
| name        | appliance_data |
| filesystem  | btrfs          |
| passthrough | false          |
| pool_id     | lv_data_pool   |
| size        | 10GB           |
| available   | 0B             |
| containers  | None           |
+-------------+----------------+
```

If the available size is displayed as 0 GB, then increase the size of the **`appliance_data`** quotagroup by running the following command.

```
hpvs quotagroup update --name appliance_data --size 80GB
```

The data pool is not ready

If you see this in an error message: "The data pool is not ready", ensure that you add storage disks to the datapool by using the User Interface of the Secure Service Container.

standard_init_linux.go: : exec user process caused "exec format error"

You might notice one of the following errors when running the IBM Hyper Protect Virtual Servers CLI tool.

- standard_init_linux.go:211: exec user process caused "exec format error", or

- standard_init_linux.go:190: exec user process caused "exec format error"

The problem is most likely caused by the CLI commands for x86 architecture being executed on the Linux on IBM Z/LinuxONE (i.e., s390x architecture) management server, or the s390x architecture CLI commands being executed on the x86 Linux system.

To workaround the problem, ensure that you use the CLI tool with the correct tag for the management server.

- The IBM Hyper Protect Virtual Servers CLI tagged with 1.2.x.s390x.s390x must be executed on the s390x architecture management server.
- The IBM Hyper Protect Virtual Servers CLI tagged with 1.2.x.s390x must be executed on the x86 architecture management server.

gpg: Invalid option errors when generating the GPG key pair

You might encounter an error messages such as **`gpg: Invalid option "--pinentry-mode=loopback"`** or **`gpg: Invalide opiton "--generate-key"`** when generating the GPG key pair on the s390x Linux management server.

The problem is most likely caused by a different version of GnuPG tools that you have installed, such as **`gnupg 1.4.20-1ubuntu3.3`**, or **`gnupg2 2.1.11-6ubuntu2.1`**.

To resolve the problem, use **`--gen-key`** option instead of **`--generate-key`** in the command, or upgrade GnuPG to a later version such as 2.2.17.

GPG hangs or "Not enough random bytes available." error when generating the GPG key pair

You might experience GPG hangs, or encounter an error messages such as **`Not enough random bytes available. Please do some other work to give the OS a chance to collect more entropy! (Need 188 more bytes)`** when generating the GPG key pair on the s390x Linux management server.

The problem is most likely caused by a missing utility **`haveged`** on the Linux management server. For more information, see Stackoverflow.

To resolve the problem, install the **`haveged`** utility with the following command:

```
apt-get install -y haveged
```

Secure Build failed to clone the Github repository if a passphrase is associated with the private key

You might encounter the following error message or a similar one when the Secure Build tries to build the source code from a Github repository by using the private key with a passphrase.

```
Could not read from remote repository.
```

The problem is most likely caused by a known limitation that the Secure Build requires the private key used to secure access to the source Github repository does not have a passphrase.

To workaround the problem, consider one of the following options.

- Generate a new SSH key pair with the **`-N`** parameter and an empty passphrase as in the following command example. Note that both private key and public key are generated. The **`-m pem`** parameter is optional and ensures the private key is generated with a **`RSA PRIVATE KEY`** comment line.

```
ssh-keygen -t rsa -b 4096 -f /tmp/id_rsa -N "" -m pem
```

- Overwrite the private key with an empty passphrase as in the following command example. Note that the public key is not changed.

```
openssl rsa -in id_rsa -out id_rsa
```

After you generate the new key pair or overwrite the private key, ensure that you update the **github:key** value with the new private key, and then run the **securebuild update** command to apply the changes. For more information, see Updating the configuration of a running Secure Build container.

Hyper Protect Virtual Server instance restarting continuously when running `hpvs vs list` command

You might notice that the Hyper Protect Virtual Server container has been in the restarting state continuously if you run the **hpvs vs list** command.

The problem is most likely caused by the excessive memory setting assigned to the Hyper Protect Virtual Server container. The memory size allocated to the Hyper Protect Virtual Server container cannot exceed the available memory resource on the Secure Service Container partition.

To workaround the problem, consider one of the following options

- Ask the appliance or system administrator to allocate sufficient memory on the Secure Service Container partition before creating or updating the Hyper Protect Virtual Server container.
- Change the memory setting of the Hyper Protect Virtual Server container to a valid value, and then use the **hpvs vs update** command to update the container.

Known issues with IBM Hyper Protect Virtual Servers 1.2.2

ERROR: HVS-VSUD003 Update virtual server failed due to wrong quotagroup configuration

If you see this error, it might be because an invalid parameter was specified in the quotagroup configuration. You can provide valid quotagroup configuration details and retry the command. Note: The parameter **reset_root** flag is not supported in IBM Hyper Protect Virtual Servers version 1.2.2, for updating a virtual server by using the **hpvs vs update** command. You can use the **hpvs deploy -u** command to update the virtual server with the parameter **reset_root:true**.

HPVS container hangs or secure shell (SSH) access fails

If a running container hangs or SSH access to the container fails, check if the non-passthrough quotagroup that is used for the root disk (RUNQ_ROOTDISK) has enough available size. The available size should be set to more than 5 GB. You can use the **hpvs quotagroup update** command to increase the available size.

The "hpvs vs `**`" command failed

If you see the following issue:

```
$ hpvs vs * * command is throwing error like
ERROR: HVS-<Error code> API Error: .......: no space left on device 500
```

Then, check the **appliance_data** quotagroup size.

```
$ sudo hpvs quotagroup show --name appliance_data
+-------------+----------------+
| PROPERTIES  | VALUES         |
+-------------+----------------+
| name        | appliance_data |
| filesystem  | btrfs          |
| passthrough | false          |
| pool_id     | lv_data_pool   |
| size        | 10GB           |
| available   | 0B             |
| containers  | None           |
+-------------+----------------+
```

If the available size is displayed as 0 GB, then increase the size of the `appliance_data` quotagroup by running the following command.

```
hpvs quotagroup update --name appliance_data --size 80GB
```

The data pool is not ready

If you see this in an error message: "The data pool is not ready", ensure that you add storage disks to the datapool by using the User Interface of the Secure Service Container.

standard_init_linux.go: : exec user process caused "exec format error"

You might notice one of the following errors when running the IBM Hyper Protect Virtual Servers CLI tool.

- standard_init_linux.go:211: exec user process caused "exec format error", or
- standard_init_linux.go:190: exec user process caused "exec format error"

The problem is most likely caused by the CLI commands for x86 architecture being executed on the Linux on IBM Z/LinuxONE (i.e., s390x architecture) management server, or the s390x architecture CLI commands being executed on the x86 Linux system.

To workaround the problem, ensure that you use the CLI tool with the correct tag for the management server.

- The IBM Hyper Protect Virtual Servers CLI tagged with 1.2.x.s390x.s390x must be executed on the s390x architecture management server.
- The IBM Hyper Protect Virtual Servers CLI tagged with 1.2.x.s390x must be executed on the x86 architecture management server.

gpg: Invalid option errors when generating the GPG key pair

You might encounter an error messages such as `gpg: Invalid option "--pinentry-mode=loopback"` or `gpg: Invalide opiton "--generate-key"` when generating the GPG key pair on the s390x Linux management server.

The problem is most likely caused by a different version of GnuPG tools that you have installed, such as `gnupg 1.4.20-1ubuntu3.3`, or `gnupg2 2.1.11-6ubuntu2.1`.

To resolve the problem, use `--gen-key` option instead of `--generate-key` in the command, or upgrade GnuPG to a later version such as 2.2.17.

GPG hangs or "Not enough random bytes available." error when generating the GPG key pair

You might experience GPG hangs, or encounter an error messages such as **`Not enough random bytes available. Please do some other work to give the OS a chance to collect more entropy! (Need 188 more bytes)`** when generating the GPG key pair on the s390x Linux management server.

The problem is most likely caused by a missing utility **`haveged`** on the Linux management server. For more information, see Stackoverflow.

To resolve the problem, install the **`haveged`** utility with the following command:

```
apt-get install -y haveged
```

Secure Build failed to clone the Github repository if a passphrase is associated with the private key

You might encounter the following error message or a similar one when the Secure Build tries to build the source code from a Github repository by using the private key with a passphrase.

```
Could not read from remote repository.
```

The problem is most likely caused by a known limitation that the Secure Build requires the private key used to secure access to the source Github repository does not have a passphrase.

To workaround the problem, consider one of the following options.

- Generate a new SSH key pair with the **`-N`** parameter and an empty passphrase as in the following command example. Note that both private key and public key are generated. The **`-m pem`** parameter is optional and ensures the private key is generated with a **`RSA PRIVATE KEY`** comment line.

  ```
  ssh-keygen -t rsa -b 4096 -f /tmp/id_rsa -N "" -m pem
  ```

- Overwrite the private key with an empty passphrase as in the following command example. Note that the public key is not changed.

  ```
  openssl rsa -in id_rsa -out id_rsa
  ```

After you generate the new key pair or overwrite the private key, ensure that you update the **`github:key`** value with the new private key, and then run the **`securebuild update`** command to apply the changes. For more information, see Updating the configuration of a running Secure Build container.

Hyper Protect Virtual Server instance restarting continuously when running `hpvs vs list` command

You might notice that the Hyper Protect Virtual Server container has been in the restarting state continuously if you run the **`hpvs vs list`** command.

The problem is most likely caused by the excessive memory setting assigned to the Hyper Protect Virtual Server container. The memory size allocated to the Hyper Protect Virtual Server container cannot exceed the available memory resource on the Secure Service Container partition.

To workaround the problem, consider one of the following options

- Ask the appliance or system administrator to allocate sufficient memory on the Secure Service Container partition before creating or updating the Hyper Protect Virtual Server container.
- Change the memory setting of the Hyper Protect Virtual Server container to a valid value, and then use the **`hpvs vs update`** command to update the container.

ERROR: HVS-VSUD003 Update virtual server failed due to wrong quotagroup configuration

If you see this error, it might be because an invalid parameter was specified in the quotagroup configuration. You can provide valid quotagroup configuration details and retry the command. Note: The parameter `reset_root` flag is not supported in IBM Hyper Protect Virtual Servers version 1.2.2, for updating a virtual server by using the `hpvs vs update` command. You can use the `hpvs deploy -u` command to update the virtual server with the parameter `reset_root:true`.

HPVS container hangs or secure shell (SSH) access fails

If a running container hangs or SSH access to the container fails, check if the non-passthrough quotagroup that is used for the root disk (RUNQ_ROOTDISK) has enough available size. The available size should be set to more than 5 GB. You can use the `hpvs quotagroup update` command to increase the available size.

The "hpvs vs `**`" command failed

If you see the following issue:

```
$ hpvs vs * * command is throwing error like
ERROR: HVS-<Error code> API Error: .......: no space left on device 500
```

Then, check the `appliance_data` quotagroup size.

```
$ sudo hpvs quotagroup show --name appliance_data
+-------------+----------------+
| PROPERTIES  | VALUES         |
+-------------+----------------+
| name        | appliance_data |
| filesystem  | btrfs          |
| passthrough | false          |
| pool_id     | lv_data_pool   |
| size        | 10GB           |
| available   | 0B             |
| containers  | None           |
+-------------+----------------+
```

If the available size is displayed as 0 GB, then increase the size of the `appliance_data` quotagroup by running the following command.

```
hpvs quotagroup update --name appliance_data --size 80GB
```

The data pool is not ready

If you see this in an error message: "The data pool is not ready", ensure that you add storage disks to the datapool by using the User Interface of the Secure Service Container.

standard_init_linux.go: : exec user process caused "exec format error"

You might notice one of the following errors when running the IBM Hyper Protect Virtual Servers CLI tool.

- standard_init_linux.go:211: exec user process caused "exec format error", or
- standard_init_linux.go:190: exec user process caused "exec format error"

The problem is most likely caused by the CLI commands for x86 architecture being executed on the Linux on IBM Z/LinuxONE (i.e., s390x architecture) management server, or the s390x architecture CLI commands being executed on the x86 Linux system.

To workaround the problem, ensure that you use the CLI tool with the correct tag for the management server.

- The IBM Hyper Protect Virtual Servers CLI tagged with 1.2.x.s390x.s390x must be executed on the s390x architecture management server.
- The IBM Hyper Protect Virtual Servers CLI tagged with 1.2.x.s390x must be executed on the x86 architecture management server.

gpg: Invalid option errors when generating the GPG key pair

You might encounter an error messages such as `gpg: Invalid option "--pinentry-mode=loopback"` or `gpg: Invalide opiton "--generate-key"` when generating the GPG key pair on the s390x Linux management server.

The problem is most likely caused by a different version of GnuPG tools that you have installed, such as `gnupg 1.4.20-1ubuntu3.3`, or `gnupg2 2.1.11-6ubuntu2.1`.

To resolve the problem, use `--gen-key` option instead of `--generate-key` in the command, or upgrade GnuPG to a later version such as 2.2.17.

GPG hangs or "Not enough random bytes available." error when generating the GPG key pair

You might experience GPG hangs, or encounter an error messages such as `Not enough random bytes available. Please do some other work to give the OS a chance to collect more entropy! (Need 188 more bytes)` when generating the GPG key pair on the s390x Linux management server.

The problem is most likely caused by a missing utility `haveged` on the Linux management server. For more information, see Stackoverflow.

To resolve the problem, install the `haveged` utility with the following command:

```
apt-get install -y haveged
```

Secure Build failed to clone the Github repository if a passphrase is associated with the private key

You might encounter the following error message or a similar one when the Secure Build tries to build the source code from a Github repository by using the private key with a passphrase.

```
Could not read from remote repository.
```

The problem is most likely caused by a known limitation that the Secure Build requires the private key used to secure access to the source Github repository does not have a passphrase.

To workaround the problem, consider one of the following options.

- Generate a new SSH key pair with the `-N` parameter and an empty passphrase as in the following command example. Note that both private key and public key are generated. The `-m pem` parameter is optional and ensures the private key is generated with a `RSA PRIVATE KEY` comment line.

  ```
  ssh-keygen -t rsa -b 4096 -f /tmp/id_rsa -N "" -m pem
  ```

- Overwrite the private key with an empty passphrase as in the following command example. Note that the public key is not changed.

  ```
  openssl rsa -in id_rsa -out id_rsa
  ```

After you generate the new key pair or overwrite the private key, ensure that you update the **`github:key`** value with the new private key, and then run the **`securebuild update`** command to apply the changes. For more information, see [Updating the configuration of a running Secure Build container](#).

Hyper Protect Virtual Server instance restarting continuously when running `hpvs vs list` command

You might notice that the Hyper Protect Virtual Server container has been in the restarting state continuously if you run the **`hpvs vs list`** command.

The problem is most likely caused by the excessive memory setting assigned to the Hyper Protect Virtual Server container. The memory size allocated to the Hyper Protect Virtual Server container cannot exceed the available memory resource on the Secure Service Container partition.

To workaround the problem, consider one of the following options

- Ask the appliance or system administrator to allocate sufficient memory on the Secure Service Container partition before creating or updating the Hyper Protect Virtual Server container.
- Change the memory setting of the Hyper Protect Virtual Server container to a valid value, and then use the **`hpvs vs update`** command to update the container.

Known issues with IBM Hyper Protect Virtual Servers 1.2.1.1, or 1.2.1

The "hpvs vs `**`" command failed

If you see the following issue:

```
$ hpvs vs * * command is throwing error like
ERROR: HVS-<Error code> API Error: .......: no space left on device 500
```

Then, check the **`appliance_data`** quotagroup size.

```
$ sudo hpvs quotagroup show --name appliance_data
+-------------+----------------+
| PROPERTIES  | VALUES         |
+-------------+----------------+
| name        | appliance_data |
| filesystem  | btrfs          |
| passthrough | false          |
| pool_id     | lv_data_pool   |
| size        | 10GB           |
| available   | 0B             |
| containers  | None           |
+-------------+----------------+
```

If the available size is displayed as 0 GB, then increase the size of the **`appliance_data`** quotagroup by running the following command.

```
hpvs quotagroup update --name appliance_data --size 80GB
```

The data pool is not ready

If you see this in an error message: "The data pool is not ready", ensure that you add storage disks to the datapool by using the User Interface of the Secure Service Container.

standard_init_linux.go: : exec user process caused "exec format error"

You might notice one of the following errors when running the IBM Hyper Protect Virtual Servers CLI tool.

- standard_init_linux.go:211: exec user process caused "exec format error", or
- standard_init_linux.go:190: exec user process caused "exec format error"

The problem is most likely caused by the CLI commands for x86 architecture being executed on the Linux on IBM Z/LinuxONE (i.e., s390x architecture) management server, or the s390x architecture CLI commands being executed on the x86 Linux system.

To workaround the problem, ensure that you use the CLI tool with the correct tag for the management server.

- The IBM Hyper Protect Virtual Servers CLI tagged with 1.2.x.s390x.s390x must be executed on the s390x architecture management server.
- The IBM Hyper Protect Virtual Servers CLI tagged with 1.2.x.s390x must be executed on the x86 architecture management server.

gpg: Invalid option errors when generating the GPG key pair

You might encounter an error messages such as **`gpg: Invalid option "--pinentry-mode=loopback"`** or **`gpg: Invalide opiton "--generate-key"`** when generating the GPG key pair on the s390x Linux management server.

The problem is most likely caused by a different version of GnuPG tools that you have installed, such as **`gnupg 1.4.20-1ubuntu3.3`**, or **`gnupg2 2.1.11-6ubuntu2.1`**.

To resolve the problem, use **`--gen-key`** option instead of **`--generate-key`** in the command, or upgrade GnuPG to a later version such as 2.2.17.

GPG hangs or "Not enough random bytes available." error when generating the GPG key pair

You might experience GPG hangs, or encounter an error messages such as **`Not enough random bytes available. Please do some other work to give the OS a chance to collect more entropy! (Need 188 more bytes)`** when generating the GPG key pair on the s390x Linux management server.

The problem is most likely caused by a missing utility **`haveged`** on the Linux management server. For more information, see Stackoverflow.

To resolve the problem, install the **`haveged`** utility with the following command:

```
apt-get install -y haveged
```

Secure Build failed to clone the Github repository if a passphrase is associated with the private key

You might encounter the following error message or a similar one when the Secure Build tries to build the source code from a Github repository by using the private key with a passphrase.

```
Could not read from remote repository.
```

The problem is most likely caused by a known limitation that the Secure Build requires the private key used to secure access to the source Github repository does not have a passphrase.

To workaround the problem, consider one of the following options.

- Generate a new SSH key pair with the `-N` parameter and an empty passphrase as in the following command example. Note that both private key and public key are generated. The `-m pem` parameter is optional and ensures the private key is generated with a `RSA PRIVATE KEY` comment line.

  ```
  ssh-keygen -t rsa -b 4096 -f /tmp/id_rsa -N "" -m pem
  ```

- Overwrite the private key with an empty passphrase as in the following command example. Note that the public key is not changed.

  ```
  openssl rsa -in id_rsa -out id_rsa
  ```

After you generate the new key pair or overwrite the private key, ensure that you update the `github:key` value with the new private key, and then run the `securebuild update` command to apply the changes. For more information, see Updating the configuration of a running Secure Build container.

Hyper Protect Virtual Server instance restarting continuously when running `hpvs vs list` command

You might notice that the Hyper Protect Virtual Server container has been in the restarting state continuously if you run the `hpvs vs list` command.

The problem is most likely caused by the excessive memory setting assigned to the Hyper Protect Virtual Server container. The memory size allocated to the Hyper Protect Virtual Server container cannot exceed the available memory resource on the Secure Service Container partition.

To workaround the problem, consider one of the following options

- Ask the appliance or system administrator to allocate sufficient memory on the Secure Service Container partition before creating or updating the Hyper Protect Virtual Server container.
- Change the memory setting of the Hyper Protect Virtual Server container to a valid value, and then use the `hpvs vs update` command to update the container.

References

Refer to the following topics when you use IBM Hyper Protect Virtual Servers.

- File and directory structure of IBM Hyper Protect Virtual Servers Version 1.2.1
- Commands in IBM Hyper Protect Virtual Servers
- Configuration files in IBM Hyper Protect Virtual Servers
- Network requirements for Hyper Protect Virtual Server
- Overview of quotagroups for IBM Hyper Protect Virtual Servers
- Updating the parameters of IBM Hyper Protect Virtual Servers
- High availability and disaster recovery
  - Backing up and recovering SSH images
  - Backing up and recovering non-SSH images

File and directory structure of IBM Hyper Protect Virtual Servers

After you download and extract the IBM Hyper Protect Virtual Servers image file, you can see the similar layout of files and directories under the **`<installation_directory>`** directory.

Note: For the sample layout of files and directories in IBM Hyper Protect Virtual Servers 1.2.0 or 1.2.0.1, see this topic.

For more information about how to get the IBM Hyper Protect Virtual Servers image file, see Downloading the installation package.

```
.
├── License
│   ├── LA_cs
│   ├── LA_de
│   ├── LA_el
│   ├── LA_en
│   ├── LA_es
│   ├── LA_fr
│   ├── LA_in
│   ├── LA_it
│   ├── LA_ja
│   ├── LA_ko
│   ├── LA_lt
│   ├── LA_pl
│   ├── LA_pt
│   ├── LA_ru
│   ├── LA_sl
│   ├── LA_tr
│   ├── LA_zh
│   ├── LA_zh_TW
│   ├── LI_cs
│   ├── LI_de
│   ├── LI_el
│   ├── LI_en
│   ├── LI_es
│   ├── LI_fr
│   ├── LI_in
│   ├── LI_it
│   ├── LI_ja
│   ├── LI_ko
│   ├── LI_lt
│   ├── LI_pl
│   ├── LI_pt
│   ├── LI_ru
│   ├── LI_sl
│   ├── LI_tr
│   ├── LI_zh
│   ├── LI_zh_TW
│   ├── non_ibm_license
│   └── notices
├── M02VFEN.tar.gz
├── bin
│   ├── hpvs_s390x
│   └── hpvs_x86
```

```
├── config
│   ├── templates
│   │   ├── virtualserver.template.readme.yml
│   │   └── virtualserver.template.yml
│   └── yaml
│       ├── secure_build.yml.example
│       ├── secure_create.yml.example
│       ├── vs_configfile_readme.yml
│       ├── vs_grep11.yml
│       ├── vs_hpvsopbase.yml
│       ├── vs_hpvsopbasessh.yml
│       ├── vs_monitoring.yml
│       ├── vs_regfiledeployexample.yml
│       └── vs_securebuild.yml
├── envcheck.sh
├── images
│   ├── CollectdHost.tar.gz
│   ├── HpvsopBase.tar.gz
│   ├── HpvsopBaseSSH.tar.gz
│   ├── Monitoring.tar.gz
│   ├── SecureDockerBuild.tar.gz
│   └── hpcsKpGrep11_runq.tar.gz
├── mustgather.sh
├── readme.txt
├── secure-service-container-for-hpvs.appliance.4.3.5.img.gz
├── setup.sh
├── swidtag
│   └── ibm.com_IBM_Hyper_Protect_Virtual_Servers-1.2.4.swidtag
└── version
```

Note

- `readme.txt`, which is the general README file for IBM Hyper Protect Virtual Servers.
- `License`, a directory that contains the license files of IBM Hyper Protect Virtual Servers.
- `version`, which states the current version of IBM Hyper Protect Virtual Servers.
- `./secure-service-container-for-hpvs.appliance.3.17.0.img.gz`, which is the hosting appliance to be installed on the IBM Z or LinuxONE system.
- `./images/HpvsopBase.tar.gz`, which is the base image of a Hyper Protect Virtual Server container without the secure shell (SSH) access.
- `./images/HpvsopBaseSSH.tar.gz`, which is the base image of a Hyper Protect Virtual Server container with the secure shell (SSH) access.
- `./images/CollectdHost.tar.gz`, which is the base image of collectd-host container of the monitoring infrastructure.
- `./images/SecureDockerBuild.tar.gz`, which is the docker image of the Secure Build container.
- `./images/Monitoring.tar.gz`, which is the base image of monitoring-host container of the monitoring infrastructure.
- `./images/hpcsKpGrep11_runq.tar.gz`, which is the base image of the GREP11 container.
- `./config/templates/virtualserver.template.yml`, which is the template example of network, quotagroup, and resoource definitions for the virtual server.
- `./config/yaml/`, a directory that contains configuration example files for the Hyper Protect Virtual Server containers.
- `/envcheck.sh`, the shell script that automates checking the prerequisites for Linux management server for setting up the IBM Hyper Protect Virtual Servers environment.
- `/setup.sh`, the shell script that automates setting up the IBM Hyper Protect Virtual Servers environment.
- `./config/mustgather.sh`, an automated script to collect debug information when you want to open a support ticket.

Commands in IBM Hyper Protect Virtual Servers

Learn about the **hpvs** commands that you can run to manage your IBM Hyper Protect Virtual Servers. If the IBM Hyper Protect Virtual Servers Version is 1.2.2, or later, you can use the --json flag when you want the output to be displayed in json format. You can also redirect this output to a file that you specify.

Commands

hpvs crypto

List crypto domains.

Example:

```
hpvs crypto --help
List crypto

Usage:
  hpvs crypto [command]

Available Commands:
  list        List crypto

Flags:
  -h, --help      Help for crypto
  --host string   Host LPAR name (This flag is applicable only for Hyper Protect 
Virtual Servers version 1.2.2, or later)
  --json       if --json flag is passed, the output will be in json format (This flag 
is applicable for Hyper Protect Virtual Servers version 1.2.2, or later)

Global Flags:
      --debug                   If --debug is passed, it will enable debug logs
      --host string             Host LPAR name (This Global flag is applicable only for
Hyper Protect Virtual Servers version 1.2.1)
      --log-output-dir string   Set log output directory

Use "hpvs crypto [command] --help" for more information about a command.
```

hpvs crypto list

List the crypto card information.

Example:

```
hpvs crypto list --help
List crypto card information

Usage:
  hpvs crypto list [flags]

Flags:
  -h, --help   Help for list

Global Flags:
      --debug                   If --debug is passed, it will enable debug logs
      --host string             Host LPAR name
      --json                    if --json flag is passed, the output will be in json 
format (This flag is applicable for Hyper Protect Virtual Servers version 1.2.2, or 
later)
      --log-output-dir string   Set log output directory
```

hpvs deploy

Deploy a Hyper Protect Virtual Server instance.

```
hpvs deploy --help
Deploy virtual servers

Usage:
  hpvs deploy [flags]

Flags:
      --config string         YAML configuration file for the virtual server deployment
      --exclude strings       Virtual servers e.g vs1,vs2; to be excluded from 
deploying, other vs will be included for deployment, by default all vs will be deployed 
(This flag is applicable for Hyper Protect Virtual Servers version 1.2.2, or later)
  -h, --help                  Help for deploy
      --include strings       Virtual servers e.g vs1,vs2; to be included for 
deploying, other vs will be excluded from deployment by default all vs will be deployed 
(This flag is applicable for Hyper Protect Virtual Servers version 1.2.2, or later)
      --templatefile string   YAML resource template file for the virtual server 
deployment
       -u, --update           If -u is passed virtual server deployment setup is 
updated (This flag is applicable for Hyper Protect Virtual Servers version 1.2.2, or 
later)

Global Flags:
      --debug                   If --debug is passed, it will enable debug logs
      --host string             Host LPAR name (This Global flag is applicable only for 
Hyper Protect Virtual Servers version 1.2.1)
      --log-output-dir string   Set log output directory
```

hpvs help

Display the help information.

Example:

```
hpvs --help
IBM® Hyper Protect Virtual Servers, the evolution of the
  IBM® Secure Service Container for IBM® Cloud Private offering,
  protects Linux workloads on IBM Z and LinuxONE throughout their
  lifecycle build management and deployment.
  This solution delivers the security needed to protect
  mission critical applications in hybrid multi-cloud deployments.

Usage:
  hpvs [command]

Available Commands:
  crypto      Crypto command
  deploy      Deploy command
  help        Help about any command
  host        Host command
  image       Image Command
  network     Network command
  quotagroup  Quotagroup command
  regfile     Generate encrypted repository registration file. If you have already 
image build on s390x arch
  registry    Registry command
  repository  Repository command
  sb          SecureBuild command
  snapshot    Snapshot command
  version     Print hpvs version
  vs          Virtual Server command

Flags:
      --debug                   If --debug is passed, it will enable debug logs
  -h, --help                    Help for hpvs
      --host string             Host LPAR name
      --json                    if --json flag is passed, the output will be in json 
```

```
format (This flag is applicable for Hyper Protect Virtual Servers version 1.2.2, or
later)
      --log-output-dir string   Set log output directory

Use "hpvs [command] --help" for more information about a command.
```

hpvs host

Add, delete, update, list, unset, or set the Secure Service Container partition information in the **`hosts`** file.

Example:

```
hpvs host --help
add, delete, update, list, unset, show, set Host

Usage:
  hpvs host [command]

Available Commands:
  add         Add host
  delete      Delete host
  list        List host
  set         Set host
  show        Show host
  unset       Unset host
  update      Update host

Flags:
  -h, --help   Help for host

Global Flags:
      --debug                   If --debug is passed, it will enable debug logs
      --host string             Host LPAR name (This Global flag is applicable only for
Hyper Protect Virtual Servers version 1.2.1)
      --log-output-dir string   Set log output directory

Use "hpvs host [command] --help" for more information about a command.
```

hpvs host add

Add the connection information to a Secure Service Container partition into the **`hosts`** file.

Example:

```
hpvs host add --help
Add host

Usage:
  hpvs host add [flags]

Flags:
  -h, --help          Help for add
      --ip string     IP address of the secure service container host(LPAR)
      --name string   Name of the secure service container host(LPAR)
      --user string   REST user name of the secure service container host(LPAR)

Global Flags:
      --debug                   If --debug is passed, it will enable debug logs
      --host string             Host LPAR name (This Global flag is applicable only for
Hyper Protect Virtual Servers version 1.2.1)
      --log-output-dir string   Set log output directory
```

Note: For more information about specifications for the username, and host(LPAR) name validation, see Chapter 3 - Configuring a Secure Service Container partition on a standard mode system in Secure Service Container User's Guide.

hpvs host delete

Delete an entry from the **host** file. If there is only one host in the list, it is taken as the default host

Example:

```
hpvs host delete --help
Delete host

Usage:
  hpvs host delete [flags]

Flags:
  -h, --help          Help for delete
      --name string   Name of the secure service container host(LPAR)

Global Flags:
      --debug                   If --debug is passed, it will enable debug logs
      --host string             Host LPAR name (This Global flag is applicable only for
Hyper Protect Virtual Servers version 1.2.1)
      --log-output-dir string   Set log output directory
```

hpvs host list

List the entries in the **hosts** file.

Example:

```
hpvs host list --help
List host

Usage:
  hpvs host list [flags]

Flags:
  -h, --help   Help for list

Global Flags:
      --debug                   If --debug is passed, it will enable debug logs
      --host string             Host LPAR name (This Global flag is applicable only for
Hyper Protect Virtual Servers version 1.2.1)
      --log-output-dir string   Set log output directory
```

hpvs host set

Set the Secure Service Container partition to work on.

Example:

```
hpvs host set --help
Set host

Usage:
  hpvs host set [flags]

Flags:
  -h, --help          Help for set
      --name string   Name of the secure service container host(LPAR)

Global Flags:
      --debug                   If --debug is passed, it will enable debug logs
      --host string             Host LPAR name (This Global flag is applicable only for
Hyper Protect Virtual Servers version 1.2.1)
      --log-output-dir string   Set log output directory
```

hpvs host show

Show the host details (This command is supported in Hyper Protect Virtual Servers version 1.2.3, or later)

Example:

```
hpvs host show --help
Show host

Usage:
  hpvs host show [flags]
Flags:
  -h, --help          Help for show
      --json          if --json flag is passed, the output will be in json format
      --name string   Name of the secure service container host(LPAR)

Global Flags:
      --debug                   If --debug is passed, it will enable debug logs
      --host string             Host LPAR name (This Global flag is applicable only for
Hyper Protect Virtual Servers version 1.2.1)
      --log-output-dir string   Set log output directory
```

hpvs host unset

Unset host (This command is supported in Hyper Protect Virtual Servers version 1.2.3, or later)

Example:

```
hpvs host unset --help
Unset host

Usage:
  hpvs host unset [flags]
Flags:
  -h, --help          Help for unset
      --json          if --json flag is passed, the output will be in json format
      --name string   Name of the secure service container host(LPAR)
Global Flags:
      --debug                   If --debug is passed, it will enable debug logs
      --host string             Host LPAR name (This Global flag is applicable only for
Hyper Protect Virtual Servers version 1.2.1)
      --log-output-dir string   Set log output directory
```

hpvs host update

Update the password for an entry in the **hosts** file.

Example:

```
hpvs host update --help
Update host

Usage:
  hpvs host update [flags]

Flags:
  -h, --help          Help for update
      --name string   Name of the secure service container host(LPAR)

Global Flags:
      --debug                   If --debug is passed, it will enable debug logs
      --host string             Host LPAR name (This Global flag is applicable only for
Hyper Protect Virtual Servers version 1.2.1)
      --log-output-dir string   Set log output directory
```

hpvs image

List, delete, show, load, or pull Images

Example:

```
hpvs image --help
list, delete, show, load, pull Image

Usage:
  hpvs image [command]

Available Commands:
  delete      Delete image
  list        List image
  load        Upload image
  pull        Pull image
  show        Show image

Flags:
  -h, --help          Help for image
      --host string   Host LPAR name (This flag is applicable only for Hyper Protect 
Virtual Servers version 1.2.2, or later)
      --json          if --json flag is passed, the output will be in json format (This 
flag is applicable for Hyper Protect Virtual Servers version 1.2.2, or later)

Global Flags:
      --debug                   If --debug is passed, it will enable debug logs
      --host string             Host LPAR name (This Global flag is applicable only for 
Hyper Protect Virtual Servers version 1.2.1)
      --log-output-dir string   Set log output directory

Use "hpvs image [command] --help" for more information about a command.
```

hpvs image delete

Delete an image from the Secure Service Container partition.

Example:

```
hpvs image delete --help
Delete image

Usage:
  hpvs image delete [flags]

Flags:
  -h, --help        Help for delete
      --id string   Image id

Global Flags:
      --debug                   If --debug is passed, it will enable debug logs
      --host string             Host LPAR name
      --json                    if --json flag is passed, the output will be in json 
format (This flag is applicable for Hyper Protect Virtual Servers version 1.2.2, or 
later)
      --log-output-dir string   Set log output directory
```

hpvs image list

List images on the Secure Service Container partition.

Example

```
hpvs image list --help
List image

Usage:
  hpvs image list [flags]

Flags:
  -h, --help   Help for list

Global Flags:
      --debug                   If --debug is passed, it will enable debug logs
      --host string             Host LPAR name
      --json                    if --json flag is passed, the output will be in json 
format (This flag is applicable for Hyper Protect Virtual Servers version 1.2.2, or 
later)
      --log-output-dir string   Set log output directory
```

hpvs image load

Load an image into the Secure Service Container partition.

Example:

```
hpvs image load --help
Upload image using tar bundle

Usage:
  hpvs image load [flags]

Flags:
      --file string   Image file path. Eg: --file=/home/user/img.tar.gz
  -h, --help          Help for load

Global Flags:
      --debug                   If --debug is passed, it will enable debug logs
      --host string             Host LPAR name
      --json                    if --json flag is passed, the output will be in json 
format (This flag is applicable for Hyper Protect Virtual Servers version 1.2.2, or 
later)
      --log-output-dir string   Set log output directory
```

hpvs image pull

Pull an image from a repository defined in the **`registry`** file.

```
hpvs image pull --help
Pull image from docker hub

Usage:
  hpvs image pull [flags]

Flags:
  -h, --help          Help for pull
      --repo string   Repository id
      --tag string    Image tag

Global Flags:
      --debug                   If --debug is passed, it will enable debug logs
      --host string             Host LPAR name
      --json                    if --json flag is passed, the output will be in json 
format (This flag is applicable for Hyper Protect Virtual Servers version 1.2.2, or 
later)
      --log-output-dir string   Set log output directory
```

hpvs image show

Show the details information of all images on a Secure Service Container partition.

Example:

```
hpvs image show --help
Show image details

Usage:
  hpvs image show [flags]

Flags:
  -h, --help        Help for show
      --id string   Image id

Global Flags:
      --debug                      If --debug is passed, it will enable debug logs
      --host string                Host LPAR name
      --json                       if --json flag is passed, the output will be in json 
format (This flag is applicable for Hyper Protect Virtual Servers version 1.2.2, or 
later)
      --log-output-dir string   Set log output directory
```

hpvs network

List, create, delete, or display the network information in the IBM Hyper Protect Virtual Servers.

Example:

```
hpvs network --help
list, create, delete, show Network

Usage:
  hpvs network [command]

Available Commands:
  create      Create network
  delete      Delete network
  list        List network
  show        Show network

Flags:
  -h, --help      Help for network
  --host string   Host LPAR name (This flag is applicable only for Hyper Protect 
Virtual Servers version 1.2.2, or later)
  --json          if --json flag is passed, the output will be in json format (This 
flag is applicable for Hyper Protect Virtual Servers version 1.2.2, or later)

Global Flags:
      --debug                      If --debug is passed, it will enable debug logs
      --host string                Host LPAR name (This Global flag is applicable only for
Hyper Protect Virtual Servers version 1.2.1)
      --log-output-dir string   Set log output directory

Use "hpvs network [command] --help" for more information about a command.
```

hpvs network create

Create a network in the IBM Hyper Protect Virtual Servers.

Example:

```
hpvs network create --help
Create network

Usage:
  hpvs network create [flags]

Flags:
      --driver string    Network driver name. bridge or macvlan. (default "bridge")
      --gateway string   Gateway IP
  -h, --help             Help for create
      --name string      Network name
      --parent string    Parent network interface name
      --range string     IP address range for DHCP vs assignment. ex- 192.168.0.0/30
      --subnet string    Subnet address. ex- 192.168.1.0/24

Global Flags:
      --debug                   If --debug is passed, it will enable debug logs
      --host string             Host LPAR name
      --json                    if --json flag is passed, the output will be in json 
format (This flag is applicable for Hyper Protect Virtual Servers version 1.2.2, or 
later)
      --log-output-dir string   Set log output directory
```

hpvs network delete

Delete a network from IBM Hyper Protect Virtual Servers.

Example:

```
hpvs network delete --help
Delete network

Usage:
  hpvs network delete [flags]

Flags:
  -h, --help          Help for delete
      --name string   Network name

Global Flags:
      --debug                   If --debug is passed, it will enable debug logs
      --host string             Host LPAR name
      --json                    if --json flag is passed, the output will be in json 
format (This flag is applicable for Hyper Protect Virtual Servers version 1.2.2, or 
later)
      --log-output-dir string   Set log output directory
```

hpvs network list

List all the networks in IBM Hyper Protect Virtual Servers.

Example:

```
hpvs network list --help
List network

Usage:
  hpvs network list [flags]

Flags:
  -h, --help   Help for list

Global Flags:
      --debug                   If --debug is passed, it will enable debug logs
      --host string             Host LPAR name
      --json                    if --json flag is passed, the output will be in json 
format (This flag is applicable for Hyper Protect Virtual Servers version 1.2.2, or 
later)
      --log-output-dir string   Set log output directory
```

hpvs network show

Show the network details in IBM Hyper Protect Virtual Servers.

Example:

```
hpvs network show --help
Show network

Usage:
  hpvs network show [flags]

Flags:
      --JSON          If --JSON flag is passed it will give JSON response body (This 
flag is applicable only for Hyper Protect Virtual Servers version 1.2.1)
  -h, --help          Help for show
      --name string   Network name

Global Flags:
      --debug                   If --debug is passed, it will enable debug logs
      --host string             Host LPAR name
      --json                    if --json flag is passed, the output will be in json 
format (This flag is applicable for Hyper Protect Virtual Servers version 1.2.2, or 
later)
      --log-output-dir string   Set log output directory
```

hpvs network update

Update a network from IBM Hyper Protect Virtual Servers. This command updates the default docker network or bridge network (This command is supported in Hyper Protect Virtual Servers version 1.2.3, or later).

Example:

```
hpvs network update --help
Update network

Usage:
  hpvs network update [flags]

Flags:
  -h, --help            Help for update
      --name string     Network name, current feature only supports default docker 
network update. (default "bridge")
      --subnet string   Network subnet, complete default docker network subnet address 
e.g 172.31.0.1/16.

Global Flags:
      --debug                   If --debug is passed, it will enable debug logs
      --host string             Host LPAR name
      --json                    if --json flag is passed , the output will be in json 
format
      --log-output-dir string   Set log output directory
```

Note: The **`hpvs network update`** command is disruptive, and all running virtual servers will be restarted when you run this command.

hpvs quotagroup

create, delete, list, show, or update quotagroups in IBM Hyper Protect Virtual Servers.

Example:

```
hpvs quotagroup --help
create, delete, list, show, update Quotagroup

Usage:
  hpvs quotagroup [command]
```

```
Available Commands:
  create      Create quotagroup
  delete      Delete quotagroup
  list        List quotagroup
  show        Show quotagroup
  update      Update quotagroup

Flags:
  -h, --help      Help for quotagroup
  --host string   Host LPAR name
  --json          if --json flag is passed, the output will be in json format (This 
flag is applicable for Hyper Protect Virtual Servers version 1.2.2, or later)

Global Flags:
      --debug                   If --debug is passed, it will enable debug logs
      --host string             Host LPAR name (This Global flag is applicable only for 
Hyper Protect Virtual Servers version 1.2.1)
      --log-output-dir string   Set log output directory

Use "hpvs quotagroup [command] --help" for more information about a command.
```

hpvs quotagroup create

Create a quotagroup in IBM Hyper Protect Virtual Servers.

Example:

```
hpvs quotagroup create --help
Create quotagroup

Usage:
  hpvs quotagroup create [flags]

Flags:
      --filesystem string   Quotagroup file system. supported - btrfs, ext4, xfs, none.
  -h, --help                Help for create
      --name string         Quotagroup name
      --passthrough         If --passthrough flag is passed, passthrough quotagroup 
will be created. By default non passthrough quotagroup is created.
      --size string         Quotagroup size in GB or MB e.g. 30GB or 300MB

Global Flags:
      --debug                   If --debug is passed, it will enable debug logs
      --host string             Host LPAR name
      --json                    if --json flag is passed, the output will be in json 
format (This flag is applicable for Hyper Protect Virtual Servers version 1.2.2, or 
later)
      --log-output-dir string   Set log output directory
```

hpvs quotagroup delete

Delete a quotagroup in IBM Hyper Protect Virtual Servers.

Example:

```
hpvs quotagroup delete --help
Delete quotagroup

Usage:
  hpvs quotagroup delete [flags]

Flags:
  -h, --help          Help for delete
      --name string   Quotagroup name

Global Flags:
      --debug                      If --debug is passed, it will enable debug logs
      --host string                Host LPAR name
      --json                       if --json flag is passed, the output will be in json 
format (This flag is applicable for Hyper Protect Virtual Servers version 1.2.2, or 
later)
      --log-output-dir string   Set log output directory
```

hpvs quotagroup list

List all quotagroups in the IBM Hyper Protect Virtual Servers.

Example:

```
hpvs quotagroup list --help
List quotagroup

Usage:
  hpvs quotagroup list [flags]

Flags:
  -h, --help   Help for list

Global Flags:
      --debug                      If --debug is passed, it will enable debug logs
      --host string                Host LPAR name
      --json                       if --json flag is passed, the output will be in json 
format (This flag is applicable for Hyper Protect Virtual Servers version 1.2.2, or 
later)
      --log-output-dir string   Set log output directory
```

hpvs quotagroup show

Show the detail information of a quotagroup.

Example:

```
hpvs quotagroup show --help
Show quotagroup

Usage:
  hpvs quotagroup show [flags]

Flags:
  -h, --help          Help for show
      --name string   Quotagroup name

Global Flags:
      --debug                      If --debug is passed, it will enable debug logs
      --host string                Host LPAR name
      --json                       if --json flag is passed, the output will be in json 
format (This flag is applicable for Hyper Protect Virtual Servers version 1.2.2, or 
later)
      --log-output-dir string   Set log output directory
```

hpvs quotagroup update

Update a quotagroup with new configuration.

Example:

```
hpvs quotagroup update --help
Update quotagroup

Usage:
  hpvs quotagroup update [flags]
```

```
Flags:
  -h, --help          Help for update
      --name string   Quotagroup name
      --size string   Quotagroup size in GB or MB e.g. 30GB or 40MB

Global Flags:
      --debug                   If --debug is passed, it will enable debug logs
      --host string             Host LPAR name
      --json                    if --json flag is passed, the output will be in json
format (This flag is applicable for Hyper Protect Virtual Servers version 1.2.2, or
later)
      --log-output-dir string   Set log output directory
```

hpvs regfile

Administer the repository registration files.

Example:

```
hpvs regfile --help
Generate encrypted repository registration file. If you have already image build on
s390x arch

Usage:
  hpvs regfile [command]

Available Commands:
  create      Create encrypted repository registration file

Flags:
  -h, --help   Help for regfile

Global Flags:
      --debug                   If --debug is passed, it will enable debug logs
      --host string             Host LPAR name (This Global flag is applicable only for
Hyper Protect Virtual Servers version 1.2.1)
      --log-output-dir string   Set log output directory

Use "hpvs regfile [command] --help" for more information about a command.
```

hpvs regfile create

Create an encrypted repository registration file.

Example:

```
hpvs regfile create --help
Create encrypted repository registration file

Usage:
  hpvs regfile create [flags]

Flags:
      --config string   Config file path
  -h, --help            Help for create
      --out string      Output path for encrypted regfile. Default will be generated in
current directory

Global Flags:
      --debug                   If --debug is passed, it will enable debug logs
      --host string             Host LPAR name (This Global flag is applicable only for
Hyper Protect Virtual Servers version 1.2.1)
      --log-output-dir string   Set log output directory
```

hpvs registry

Add, delete, update, list, or show registry configurations.

Example:

```
hpvs registry --help
add, delete, update, list, show Registry

Usage:
  hpvs registry [command]

Available Commands:
  add         Add registry
  delete      Delete registry
  list        List registry
  show        Show registry
  update      Update registry

Flags:
  -h, --help   Help for registry
  --json       if --json flag is passed, the output will be in json format (This flag 
is applicable for Hyper Protect Virtual Servers version 1.2.2, or later)

Global Flags:
      --debug                   If --debug is passed, it will enable debug logs
      --host string             Host LPAR name (This Global flag is applicable only for 
Hyper Protect Virtual Servers version 1.2.1)
      --log-output-dir string   Set log output directory

Use "hpvs registry [command] --help" for more information about a command.
```

hpvs registry add

Add a registry configuration. When you add a registry, refer the following topics for more information about password rules:

- Docker registry Password Rules.
- IBM Cloud Registry.

Example:

```
hpvs registry add --help
Add registry

Usage:
  hpvs registry add [flags]

Flags:
      --dct string    Docker content-trust-server server url (default 
"https://notary.docker.io")
  -h, --help          Help for add
      --name string   Name of registry. use any name e.g - docker_pull or docker_push 
etc
      --url string    Docker server url (default "docker.io")
      --user string   User ID of docker registry

Global Flags:
      --debug                   If --debug is passed, it will enable debug logs
      --host string             Host LPAR name (This Global flag is applicable only for 
Hyper Protect Virtual Servers version 1.2.1)
      --json                    if --json flag is passed, the output will be in json 
format (This flag is applicable for Hyper Protect Virtual Servers version 1.2.2, or 
later)
      --log-output-dir string   Set log output directory
```

Note: When you are configuring the IBM Cloud registry, the `dct` parameter is required. You must set the following parameters as shown below:

- The `--dct` parameter must be specified as `https://notary.<server_url>`, for example "https://notary.us.icr.io", or "https://notary.de.icr.io". For more information about the value of `<region>`, see Using Docker to authenticate with an API key.
- The `--user` parameter must be specified as `iamapikey`. For more information about the API key, see Automating access to IBM Cloud Container Registry.
- The `--url` parameter must be specified as `<region>.icr.io`. For more information about the value of `<region>`, see Using Docker to authenticate with an API key.
- The value of `<region>` specified in the `--dct` and `--url` parameters must be the same.

hpvs registry delete

Delete a registry configuration.

Example:

```
hpvs registry delete --help
Delete registry

Usage:
  hpvs registry delete [flags]

Flags:
  -h, --help          Help for delete
      --name string   Name of registry

Global Flags:
      --debug                   If --debug is passed, it will enable debug logs
      --host string             Host LPAR name (This Global flag is applicable only for
Hyper Protect Virtual Servers version 1.2.1)
      --json                    if --json flag is passed, the output will be in json
format (This flag is applicable for Hyper Protect Virtual Servers version 1.2.2, or
later)
      --log-output-dir string   Set log output directory
```

hpvs registry list

List all registry configurations.

Example:

```
hpvs registry list --help
List registry

Usage:
  hpvs registry list [flags]

Flags:
  -h, --help   Help for list

Global Flags:
      --debug                   If --debug is passed, it will enable debug logs
      --host string             Host LPAR name (This Global flag is applicable only for
Hyper Protect Virtual Servers version 1.2.1)
      --json                    if --json flag is passed, the output will be in json
format (This flag is applicable for Hyper Protect Virtual Servers version 1.2.2, or
later)
      --log-output-dir string   Set log output directory
```

hpvs registry show

Show the detail information of a registry.

Example:

```
hpvs registry show --help
Show registry

Usage:
  hpvs registry show [flags]

Flags:
  -h, --help          Help for show
      --name string   Name of registry

Global Flags:
      --debug                    If --debug is passed, it will enable debug logs
      --host string              Host LPAR name (This Global flag is applicable only for 
Hyper Protect Virtual Servers version 1.2.1)
      --json                     if --json flag is passed, the output will be in json 
format (This flag is applicable for Hyper Protect Virtual Servers version 1.2.2, or 
later)
      --log-output-dir string   Set log output directory
```

hpvs registry update

Update a registry configuration.

Example:

```
hpvs registry update --help
Update registry

Usage:
  hpvs registry update [flags]

Flags:
      --dct string    Docker content-trust-server server url Ex: 
https://notary.docker.io
  -h, --help          Help for update
      --name string   Name of registry
      --url string    Docker server url Ex: docker.io
      --user string   User ID of docker registry

Global Flags:
      --debug                    If --debug is passed, it will enable debug logs
      --host string              Host LPAR name (This Global flag is applicable only for 
Hyper Protect Virtual Servers version 1.2.1)
      --json                     if --json flag is passed, the output will be in json 
format (This flag is applicable for Hyper Protect Virtual Servers version 1.2.2, or 
later)
      --log-output-dir string   Set log output directory
```

hpvs repository

list, register, delete, show, or update the repository configuration.

Example:

```
hpvs repository --help
list, register, delete, show, update Repository

Usage:
  hpvs repository [command]

Available Commands:
  delete      Delete repository
  list        List repository
  register    Register repository
  show        Show repository
  update      Update repository

Flags:
  -h, --help   Help for repository

Global Flags:
      --debug                    If --debug is passed, it will enable debug logs
      --host string              Host LPAR name
      --json                     if --json flag is passed, the output will be in json 
format (This flag is applicable for Hyper Protect Virtual Servers version 1.2.2, or 
later)
      --log-output-dir string   Set log output directory

Use "hpvs repository [command] --help" for more information about a command.
```

hpvs repository delete

Delete a repository configuration.

Example:

```
hpvs repository delete --help
Delete repository

Usage:
  hpvs repository delete [flags]

Flags:
      --force       If --force flag is passed, will delete repository along with 
associated images and virtual servers. This operation is irreversible.
  -h, --help        Help for delete
      --id string   Repository id

Global Flags:
      --debug                    If --debug is passed, it will enable debug logs
      --host string              Host LPAR name
      --json                     if --json flag is passed, the output will be in json 
format (This flag is applicable for Hyper Protect Virtual Servers version 1.2.2, or 
later)
      --log-output-dir string   Set log output directory
```

hpvs repository list

List all repository configurations.

Example:

```
hpvs repository list --help
List repository

Usage:
  hpvs repository list [flags]

Flags:
  -h, --help   Help for list

Global Flags:
      --debug                    If --debug is passed, it will enable debug logs
      --host string              Host LPAR name
      --json                     if --json flag is passed, the output will be in json 
format (This flag is applicable for Hyper Protect Virtual Servers version 1.2.2, or 
later)
      --log-output-dir string   Set log output directory
```

hpvs repository register

Register a repository configuration.

Example:

```
hpvs repository register --help
Register repository

Usage:
  hpvs repository register [flags]

Flags:
  -h, --help         Help for register
      --id string    Repository id
      --pgp string   PGP file path

Global Flags:
      --debug                   If --debug is passed, it will enable debug logs
      --host string             Host LPAR name
      --json                    if --json flag is passed, the output will be in json 
format (This flag is applicable for Hyper Protect Virtual Servers version 1.2.2, or 
later)
      --log-output-dir string   Set log output directory
```

hpvs repository show

Show the repository configuration.

Example:

```
hpvs repository show --help
Show repository

Usage:
  hpvs repository show [flags]

Flags:
  -h, --help        Help for show
      --id string   Repository id

Global Flags:
      --debug                   If --debug is passed, it will enable debug logs
      --host string             Host LPAR name
      --json                    if --json flag is passed, the output will be in json 
format (This flag is applicable for Hyper Protect Virtual Servers version 1.2.2, or 
later)
      --log-output-dir string   Set log output directory
```

hpvs repository update

Update the repository configuration.

Example:

```
hpvs repository update --help
Update repository

Usage:
  hpvs repository update [flags]

Flags:
  -h, --help         Help for update
      --id string    Repository id
      --pgp string   PGP file path
```

```
Global Flags:
      --debug                   If --debug is passed, it will enable debug logs
      --host string             Host LPAR name
      --json                    if --json flag is passed, the output will be in json 
format (This flag is applicable for Hyper Protect Virtual Servers version 1.2.2, or 
later)
      --log-output-dir string   Set log output directory
```

hpvs sb

Administer Secure Build Virtual Servers.

Example:

```
hpvs sb --help
SecureBuild command

Usage:
  hpvs sb [command]

Available Commands:
  build       Securely build your image
  clean       Secure build clean. It will clean vs data eg - logs
  init        Initialize secure build configuration
  log         Get logs
  manifest    Get manifest file
  pubkey      Get manifest public key
  regfile     Get encrypted repository registration file
  status      Get secure build status
  update      Update secure build environment

Flags:
  -h, --help   Help for sb

Global Flags:
      --debug                   If --debug is passed, it will enable debug logs
      --host string             Host LPAR name (This Global flag is applicable only for 
Hyper Protect Virtual Servers version 1.2.1)
      --log-output-dir string   Set log output directory

Use "hpvs sb [command] --help" for more information about a command.
```

hpvs sb build

Securely build your image.

Example:

```
hpvs sb build --help
Securely build your image

Usage:
  hpvs sb build [flags]

Flags:
      --config string   Config file path
  -h, --help            Help for build
      --timeout int     Build timeout in minutes (default 10)

Global Flags:
      --debug                   If --debug is passed, it will enable debug logs
      --host string             Host LPAR name (This Global flag is applicable only for 
Hyper Protect Virtual Servers version 1.2.1)
      --log-output-dir string   Set log output directory
```

hpvs sb clean

Clean up the data on the Secure Build Virtual Server.

Example:

```
hpvs sb clean --help
Secure build clean. It will clean vs data eg - logs

Usage:
  hpvs sb clean [flags]

Flags:
      --config string   Config file path
  -h, --help            Help for clean

Global Flags:
      --debug                   If --debug is passed, it will enable debug logs
      --host string             Host LPAR name (This Global flag is applicable only for 
Hyper Protect Virtual Servers version 1.2.1)
      --log-output-dir string   Set log output directory
```

hpvs sb init

Initialize the Secure Build Virtual Server, Securely build your image, and generate the encrypted repository registration file.

Exmple:

```
hpvs sb init --help
Initialize secure build environment, securely build the image and get the encrypted 
repository registration file

Usage:
  hpvs sb init [flags]

Flags:
      --build           If --build is passed, it will init and build
      --config string   Config file path
  -h, --help            Help for init
      --out string      Output path for encrypted regfile. Default will be generated in 
current directory
      --timeout int     Build timeout in minutes (default 10)

Global Flags:
      --debug                   If --debug is passed, it will enable debug logs
      --host string             Host LPAR name (This Global flag is applicable only for 
Hyper Protect Virtual Servers version 1.2.1)
      --log-output-dir string   Set log output directory
```

hpvs sb log

View the audit, build, or system output log information of the Secure Build server. The default log is the build log.

Example:

```
hpvs sb log --help
Get secure build logs

Usage:
  hpvs sb log [flags]

Flags:
      --config string   Config file path
  -h, --help            Help for log
      --type string     There are three type ex:- audit, build, syslog (default 
"build")

Global Flags:
      --debug                   If --debug is passed, it will enable debug logs
      --host string             Host LPAR name (This Global flag is applicable only for
Hyper Protect Virtual Servers version 1.2.1)
      --log-output-dir string   Set log output directory
```

hpvs sb manifest

Retrieve the manifest file from the Secure Build server.

Exmaple:

```
hpvs sb manifest --help
Get manifest file

Usage:
  hpvs sb manifest [flags]

Flags:
      --config string   Config file path
  -h, --help            Help for manifest
      --name string     Build name. you can get build name using <hpvs sb status> after
build

Global Flags:
      --debug                   If --debug is passed, it will enable debug logs
      --host string             Host LPAR name (This Global flag is applicable only for
Hyper Protect Virtual Servers version 1.2.1)
      --log-output-dir string   Set log output directory
```

hpvs sb pubkey

Retrieve the public key to encrypt the manifest file.

Example:

```
hpvs sb pubkey --help
Get manifest public key

Usage:
  hpvs sb pubkey [flags]

Flags:
      --config string   Config file path
  -h, --help            Help for pubkey
      --name string     Build name. you can get build name using <hpvs sb status> after
build

Global Flags:
      --debug                   If --debug is passed, it will enable debug logs
      --host string             Host LPAR name (This Global flag is applicable only for
Hyper Protect Virtual Servers version 1.2.1)
      --log-output-dir string   Set log output directory
```

hpvs sb regfile

Retrieve the encrypted repository registration file based on the Secure Build configuration file.

Example:

```
hpvs sb regfile --help
Get encrypted repository registration file

Usage:
  hpvs sb regfile [flags]

Flags:
      --config string   Config file path
  -h, --help            Help for regfile
      --out string      Output path for encrypted regfile. Default will be generated in 
current directory

Global Flags:
      --debug                    If --debug is passed, it will enable debug logs
      --host string              Host LPAR name (This Global flag is applicable only for 
Hyper Protect Virtual Servers version 1.2.1)
      --log-output-dir string   Set log output directory
```

hpvs sb status

Show the status of the Secure Build servers based on the Secure Build configuration file.

Example:

```
hpvs sb status --help
Get secure build status

Usage:
  hpvs sb status [flags]

Flags:
      --config string   Config file path
  -h, --help            Help for status

Global Flags:
      --debug                    If --debug is passed, it will enable debug logs
      --host string              Host LPAR name
      --json                     if --json flag is passed, the output will be in json 
format (This flag is applicable for Hyper Protect Virtual Servers version 1.2.2, or 
later)
      --log-output-dir string   Set log output directory
```

hpvs sb update

Update the Secure Build servers based on the Secure Build configuration file.

Example:

```
hpvs sb update --help
Update secure build environment

Usage:
  hpvs sb update [flags]

Flags:
      --config string   Config file path
  -h, --help            Help for update

Global Flags:
      --debug                    If --debug is passed, it will enable debug logs
      --host string              Host LPAR name (This Global flag is applicable only for 
Hyper Protect Virtual Servers version 1.2.1)
      --log-output-dir string   Set log output directory
```

hpvs snapshot

List, create, delete, or restore a snapshot for a Hyper Protect Virtual Server instance.

Example:

```
hpvs snapshot --help
list, create, delete, restore Snapshot

Usage:
  hpvs snapshot [command]

Available Commands:
  create      Create snapshot
  delete      Delete snapshot
  list        List snapshots
  restore     Restore snapshot

Flags:
  -h, --help   Help for snapshot

Global Flags:
      --debug                   If --debug is passed, it will enable debug logs
      --host string             Host LPAR name
      --json                    if --json flag is passed, the output will be in json 
format (This flag is applicable for Hyper Protect Virtual Servers version 1.2.2, or 
later)
      --log-output-dir string   Set log output directory

Use "hpvs snapshot [command] --help" for more information about a command.
```

hpvs snapshot create

Create a snapshot for a Hyper Protect Virtual Server instance.

Example:

```
hpvs snapshot create --help
Create snapshot of a given vs

Usage:
  hpvs snapshot create [flags]

Flags:
  -h, --help          Help for create
      --name string   Snapshot name
      --vs string     VS name

Global Flags:
      --debug                   If --debug is passed, it will enable debug logs
      --host string             Host LPAR name
      --json                    if --json flag is passed, the output will be in json 
format (This flag is applicable for Hyper Protect Virtual Servers version 1.2.2, or 
later)
      --log-output-dir string   Set log output directory
```

hpvs snaptshot delete

Delete a snapshot.

Example:

```
hpvs snapshot delete --help
Delete snapshot of a given vs

Usage:
  hpvs snapshot delete [flags]

Flags:
  -h, --help          Help for delete
      --name string   Snapshot name
      --vs string     VS name
```

```
Global Flags:
      --debug                   If --debug is passed, it will enable debug logs
      --host string             Host LPAR name
      --json                    if --json flag is passed, the output will be in json 
format (This flag is applicable for Hyper Protect Virtual Servers version 1.2.2, or 
later)
      --log-output-dir string   Set log output directory
```

hpvs snapshot list

List all the snapshots.

Example:

```
hpvs snapshot list --help
List snapshots of a given vs

Usage:
  hpvs snapshot list [flags]

Flags:
  -h, --help        Help for list
      --vs string   VS name

Global Flags:
      --debug                   If --debug is passed, it will enable debug logs
      --host string             Host LPAR name
      --json                    if --json flag is passed, the output will be in json 
format (This flag is applicable for Hyper Protect Virtual Servers version 1.2.2, or 
later)
      --log-output-dir string   Set log output directory
```

hpvs snapshot restore

Restore a snapshot to a Virtual Server instance.

Example:

```
hpvs snapshot restore--help
Restore snapshot of a given vs

Usage:
  hpvs snapshot restore [flags]

Flags:
  -h, --help                Help for restore
      --name string         Snapshot name
      --quotagroup string   Quotagroup name
      --vs string           VS name

Global Flags:
      --debug                   If --debug is passed, it will enable debug logs
      --host string             Host LPAR name
      --json                    if --json flag is passed, the output will be in json 
format (This flag is applicable for Hyper Protect Virtual Servers version 1.2.2, or 
later)
      --log-output-dir string   Set log output directory
```

hpvs undeploy

Undeploy virtual servers (This command is supported in Hyper Protect Virtual Servers version 1.2.2, or later)

Example:

```
hpvs undeploy --help

Usage:
  hpvs undeploy [flags]

Flags:
      --config string      YAML configuration file used for the virtual server 
deployment
      --exclude strings    Virtual servers e.g vs1,vs2; to be excluded from 
undeploying, other vs will be included for undeployment, by default all vs will be 
undeployed
  -h, --help               Help for undeploy
      --include strings    Virtual servers e.g vs1,vs2; to be included for undeploying, 
other vs will be excluded from undeployment, by default all vs will be undeployed

Global Flags:
      --debug                    If --debug is passed, it will enable debug logs
      --host string              Host LPAR name
      --json                     if --json flag is passed , the output will be in json 
format
      --log-output-dir string    Set log output directory
```

hpvs vs

Administer Virtual Serer instances.

Example:

```
hpvs vs --help
create, delete, list, log, restart, show, start, stop VS

Usage:
  hpvs vs [command]

Available Commands:
  create      Create virtual server
  delete      Delete virtual server
  list        List virtual servers
  log         Get virtual server log
  restart     Restart virtual server
  show        Show virtual server
  start       Start virtual server
  stop        Stop virtual server
  update      Update virtual server

Flags:
  -h, --help      Help for vs
  --host string   Host LPAR name (This flag is applicable only for Hyper Protect 
Virtual Servers version 1.2.2, or later)
  --json          if --json flag is passed, the output will be in json format (This 
flag is applicable for Hyper Protect Virtual Servers version 1.2.2, or later)

Global Flags:
      --debug                    If --debug is passed, it will enable debug logs
      --host string              Host LPAR name (This Global flag is applicable only for 
Hyper Protect Virtual Servers version 1.2.1)
      --log-output-dir string    Set log output directory

Use "hpvs vs [command] --help" for more information about a command.
```

hpvs vs create

Create a Virtual Server instance.

Example:

```
hpvs vs create --help
Create virtual server

Usage:
  hpvs vs create [flags]

Flags:
      --cpu string                  Number of cpu (default "1")
      --crypto_control              If --crypto_control flag is passed, domain is of 
control type default is usage type.
      --crypto_matrix stringArray   List of crypto domain E.g --
crypto_matrix=9.0001,9.0002 , if --crypto_control is not passed then usage type with 
one crypto domain. E.g --crypto_matrix=9.0001
      --domainName string           Domain name. Ex- example.com
      --env stringToString          Environment variable of virtual server. E.g --env=
{key1=value1,key2=value2} (default [])
      --envjsonpath string          JSON environment variable path
      --extraHosts stringArray      Extra hosts. eg:- 
{"host1.example.com:192.168.0.2","host2.example.com:192.168.0.3"}
  -h, --help                        Help for create
      --hostname string             Hostname
      --labels stringArray          Labels
      --name string                 Name of virtual server
      --network stringArray         List of networks. E.g --network "
{name=example_network, ip=192.168.0.2}"
      --ports stringArray           List of ports. E.g --ports "{containerport = 443, 
protocol = tcp, hostport = 21443}"
      --quotagroup stringArray      List of quotagroup configurations. E.g --quotagroup 
"{quotagroup = volume-name, mountid = new, mount = /newroot, filesystem = ext4, size = 
4GB}"
      --ram string                  RAM in MB (default "1024")
      --repo string                 Repository id
      --tag string                  Image tag

Global Flags:
      --debug                   If --debug is passed, it will enable debug logs
      --host string             Host LPAR name
      --json                    if --json flag is passed, the output will be in json 
format (This flag is applicable for Hyper Protect Virtual Servers version 1.2.2, or 
later)
      --log-output-dir string   Set log output directory
```

hpvs vs delete

Delete a Virtual Server instance.

Example:

```
hpvs vs delete --help
Delete virtual server

Usage:
  hpvs vs delete [flags]

Flags:
  -h, --help          Help for delete
      --name string   Name of virtual server

Global Flags:
      --debug                   If --debug is passed, it will enable debug logs
      --host string             Host LPAR name
      --json                    if --json flag is passed, the output will be in json 
format (This flag is applicable for Hyper Protect Virtual Servers version 1.2.2, or 
later)
      --log-output-dir string   Set log output directory
```

hpvs vs list

List all the Virtual Server instances on the Secure Service Container partition.

Example:

```
hpvs vs list --help
List virtual servers

Usage:
  hpvs vs list [flags]

Flags:
  -h, --help   Help for list

Global Flags:
      --debug                   If --debug is passed, it will enable debug logs
      --host string             Host LPAR name
      --json                    if --json flag is passed, the output will be in json 
format (This flag is applicable for Hyper Protect Virtual Servers version 1.2.2, or 
later)
      --log-output-dir string   Set log output directory
```

hpvs vs log

Retrieve the log information of a Virtual Server instance.

Example:

```
hpvs vs log --help
Get virtual server log

Usage:
  hpvs vs log [flags]

Flags:
  -h, --help          Help for log
      --name string   Name of virtual server

Global Flags:
      --debug                   If --debug is passed, it will enable debug logs
      --host string             Host LPAR name
      --json                    if --json flag is passed, the output will be in json 
format (This flag is applicable for Hyper Protect Virtual Servers version 1.2.2, or 
later)
      --log-output-dir string   Set log output directory
```

hpvs vs restart

Restart a Virtual Server instance.

Example:

```
hpvs vs restart --help
Restart virtual server

Usage:
  hpvs vs restart [flags]

Flags:
  -h, --help          Help for restart
      --name string   Name of virtual server

Global Flags:
      --debug                   If --debug is passed, it will enable debug logs
      --host string             Host LPAR name
      --json                       if --json flag is passed, the output will be in json 
format (This flag is applicable for Hyper Protect Virtual Servers version 1.2.2, or 
later)
      --log-output-dir string   Set log output directory
```

hpvs vs show

Show the configuration of a Virtual Server instance.

Example:

```
hpvs vs show --help
Show virtual server

Usage:
  hpvs vs show [flags]

Flags:
      --JSON      If --JSON flag is passed it will give JSON response body (This flag 
is applicable only for Hyper Protect Virtual Servers version 1.2.1)
  -h, --help      Help for show
      --name string   Name of virtual server

Global Flags:
      --debug                   If --debug is passed, it will enable debug logs
      --host string             Host LPAR name
      --json                    if --json flag is passed, the output will be in json 
format (This flag is applicable for Hyper Protect Virtual Servers version 1.2.2, or 
later)
      --log-output-dir string   Set log output directory
```

hpvs vs start

Start a Virtual Server instance.

Example:

```
hpvs vs start --help
Start virtual server

Usage:
  hpvs vs start [flags]

Flags:
  -h, --help          Help for start
      --name string   Name of virtual server

Global Flags:
      --debug                   If --debug is passed, it will enable debug logs
      --host string             Host LPAR name
      --json                    if --json flag is passed, the output will be in json 
format (This flag is applicable for Hyper Protect Virtual Servers version 1.2.2, or 
later)
      --log-output-dir string   Set log output directory
```

hpvs vs stop

Stop a running Virtual Server instance.

Example:

```
hpvs vs stop --help
Stop virtual server

Usage:
  hpvs vs stop [flags]
```

```
Flags:
  -h, --help          Help for stop
      --name string   Name of virtual server

Global Flags:
      --debug                     If --debug is passed, it will enable debug logs
      --host string               Host LPAR name
      --json                      if --json flag is passed, the output will be in json 
format (This flag is applicable for Hyper Protect Virtual Servers version 1.2.2, or 
later)
      --log-output-dir string   Set log output directory
```

hpvs vs update

Update a Virtual Server instance.

Example:

```
hpvs vs update --help
Update virtual server

Usage:
  hpvs vs update [flags]

Flags:
      --cpu string                Number of cpu
      --domainName string         Domain name. Ex- example.com
      --env stringToString        Environment variable of virtual server. E.g --env=
{key1=value1,key2=value2} (default [])
      --envjsonpath string        JSON environment variable path
      --extraHosts stringArray    Extra hosts. eg:- 
{"host1.example.com:192.168.0.2","host2.example.com:192.168.0.3"}
  -h, --help                      Help for update
      --hostname string           Hostname
      --labels stringArray        Labels
      --name string               Name of virtual server
      --network stringArray       List of networks. E.g --network "
{name=example_network, ip=192.168.0.2}"
      --ports stringArray         List of ports. E.g --ports "{containerport = 443, 
protocol = tcp, hostport = 21443}"
      --quotagroup stringArray    List of quotagroup configurations. E.g --quotagroup "
{quotagroup = volume-name, mountid = new, mount = /newroot, filesystem = ext4, size = 
4GB}"
      --ram string                RAM in MB
      --repo string               Repository id
      --tag string                Image tag

Global Flags:
      --debug                     If --debug is passed, it will enable debug logs
      --host string               Host LPAR name
      --json                      if --json flag is passed, the output will be in json 
format (This flag is applicable for Hyper Protect Virtual Servers version 1.2.2, or 
later)
      --log-output-dir string   Set log output directory
```

Configuration files of IBM Hyper Protect Virtual Servers

Learn about the configuration files that you can use to manage your IBM Hyper Protect Virtual Servers.

hosts

The **hosts** file stores the connection information to Secure Service Container partitions in the following format **<LPAR_NAME>, <User_Name>, <LPAR_IP>,<Encrypted_Password>**. By default, the **hosts** file is created under the **$HOME/hpvs>** folder when you run **hpvs host add** command for the first time.

Example:

```
lpar,blockchain,10.20.5.216,YmwwY2tjaGExbg==
```

You can use **hpvs host** command to manage the content of this file. For more information, see hpvs host.

registry

The **registry** file stores the connection information to remote registry servers in the following format **<Docker_server_name>, <User_Name>, <Docker_server_URL>,<Encrypted_Password>**. By default, the **registry** file is created under the **$HOME/hpvs/config** folder as **reg.json** when you run the **hpvs registry add** command for the first time. You can use **hpvs registry** command to manage the content of this file. For more information, see hpvs registry.

repository

The **repository** file stores the connection information to remote registry servers in the following format **to-do**. By default, the encrypted **repository** file is created under the user defined path when you run the **hpvs repository register** command for the first time. You can use **hpvs repository** command to manage the content of this file. For more information, see hpvs repository.

Virtual server template file

The template file contains the definitions of the resources, volumes, environment templates, and networks that are required to create a virtual server. You edit and customize this template to suit the configuration of the virtual servers you want to create, or you can use your own templates.

```
version: '<Placeholder for virtualserver template yaml version. This will be v1>'
type: '<Placeholder to define yaml file type. This will be virtualserver-template>'
networktemplates: '<Define the list of network resource templates. This defintions will
be used to create the network defined in vs config file before the virtual server is
created.>'
# public network sample
-   name: '<Enter the network name to be created. For example, "external_network">'
    subnet: '<Enter the public network subnet. For example, 10.20.4.0/22 >'
    gateway: '<Enter the public network gateway. For example,10.20.4.1>'
    parent: '<Enter the network parent device name to be used for public network
connection. For example, encf900>'
    driver: '<Enter the network driver. For example, macvlan>'
# internal network sample
-   name: internal_network
    subnet: '<Enter the internal network subnet. For example, 192.168.40.0/24>'
    gateway: '<Enter the internal network gateway. For example, 192.168.40.1>'
    parent: '<Enter the network parent device name to be used for internal network
connection. For example, encf900>'
    driver: '<Enter the network driver. For example, bridge>'
quotagrouptemplates: '<Define the list of quotagroups resource templates. This
defintions will be used to create the quotagroup defined in vs config file before
mounting them to virtual server.>'
# Non passthrough quotagroup definitions - This quotagroups can be shared by
# creating multiple volume mountpoints with the same virtual server or multiple
# virtual server.  A non passthrough quotagroup will be dynamically created based
# on the template and attached as volume mount points to the virtual server.
# Only brtfs filesystem is supported in non passthrough quotagroups
# mount points attached to virtual server can have filesystem btrfs, ext4, xfs
-   name: '<Enter the quotagroup name to be created. For example, "qg_default".
```

```
qg_default will be created with btrfs filesystem >'
   size: '<Enter the non passthrough quotagroup size and unit to be created. Supported 
unit is MB and GB.  For example, "20GB">'
   passthrough: '<Set to false to create non pasthrough quotagroup. >'
# Passthrough quotagroup templates - A quotagroup will be dynamically created based 
# on the template and attached as single volume mount point to the virtual server.
# Allowed filesystem types for the passthrough type quogagroup are btrfs, ext4, xfs
-  name: '<Enter the quotagroup name to be created. For example, "qg_passthrough". 
qg_passthrough will be created with defined filesystem.>'
   size: '<Enter the passthrough quotagroup size and unit to be created. Supported unit 
is MB and GB.  For example, "20GB">'
   filesystem : '<Enter the filesystem for the quotagroup. The value can be btrfs, 
ext4, or xfs>'
   passthrough: '<Set to true to create pasthrough quotagroup and passed directly to a 
virtual server as a disk device. For example, true. By default the value is false which 
creates non passthrough quotagroup>'
resourcedefinitiontemplates: '<Define the list of resources. This definition will be 
used to create the virtual server with defined cpu and memory.>'
-  name: '<Enter the resource name to be used during virtual server creation. For 
example, "small".>'
   cpu: '<Enter the number of CPUs. For example, "2" >'
   memory: '<Enter the memory defined in MB. For example, "6192" >'
```

Virtual server configuration file

The configuration file contains the resources, volumes, environment templates, and networks that you want to specify when you create a virtual server. You edit and customize this template to suit the configuration of the virtual servers you want to create.

```
version:'<Placeholder for virtualserver yaml version. This will be v1>'
type: '<Placeholder to define yaml file type. This will be virtualserver>'
virtualservers:
- name: '<Enter the name of the virtual server>'
  order: '<Enter the order in which this virtual server is to be deployed. It takes int 
type>'
  host: '<Enter the  Secure Service Container partition  identifier name>'
  hostname: '<Enter the optional host name for the virtual server. For example, 
"testhostname">'
  domainname : '<Enter the optional domain name for the virtual server. For example, 
"example.com">'
  extrahosts: '<Define the optional list of host entries that need to be present in 
virtual server hosts file. >'
   - '<Enter the hosts entry in string format. For example, 
"host1.example.com:192.168.0.2" > '
  repoid: '<Enter the registered id of repository to create virtual server from>'
  imagecache: '<Set to true if image from cache to be used. Default is false which will 
pull the images and register repositories freshly everytime>'
  imagetag: '<Enter the image tag to create virtual server from>'
  reporegfile : '<Enter the repository registration file to register Secure Service 
Container partition with dockerhub or IBM Cloud registry'>
  imagefile: '<Enter the image file to be loaded to Secure Service Container 
partition>'
  resourcedefinition: '<Define the optional resourcedefinition section. Optional if 
default cpu and memory to be used>'
     ref: '<ref refers to name in the resourcedefinitiontemplates under template files 
to be used for cpu and memory allocation for virtual server>'
  environment: '<Define the list of optional environment section for the virtual server 
instance. Optional if no environment variables to be set>'
   - key: '<Enter the environment key for ssh connection. For example, "ROOT_SSH_KEY" 
>'
     value: '<Enter the environment value. For example, 
"@/root/hpvs/config/hpvsopbasessh/keys/id_rsa_base64.pub" If value starts with @, the 
content of the file will be assigned as value to the key. Here the file is expected to 
have ssh key in base64 format> '
   - key: '<Enter the environment key. For example, "RUNQ_ROOTDISK" . Specify the root 
```

```
disk of virtual server using this option. >'
     value: '<Enter the environment value. For example, newroot. This is value of
mountid which is to be assigned as rootdisk">'
   - key: '<Enter the environment key. For example, "ROOTFS_LOCK" >'
     value: '<Enter the environment value. For example, y">'
  networks: '<Define the list of optional networks section for the virtual server.
Optional if no networks to be set>'
   - ref: '<ref refers to network name in template file. Enter the network name to be
used by virtual server. For example, "external_network">'
     ipaddress: '<Enter the network IP address to access the virtual server. For
example, "10.20.4.111">'
  volumes: '<Define the list of quotagroups to be created and volume mount points from
same to be assigned for the virtual server. Optional if no volumes to be assigned>'
   - name: '<Enter the name of quotagroup to be created. For example, qg_securebuild>'
     ref: '<ref refers to name in quotagrouptemplates under template file. Using the
definition the quotagroup is created. For example, np-medium >'
     mounts: '<Define the list of mounts to be attached to virtual server from the
quotagroup. If its passthrough quotagroup only one mount point to be defined. >'
     - mount_id: '<Enter the mountid for the mount on  Secure Service Container
partition host. If mount id is not given starting prefix '/' is removed from mountpoint
and assigned as mountid. For example, mymountidentifier'
       mountpoint: '<Enter the mount point inside the virtual server where the
quotagroup is mounted. For example, /newroot>'
       filesystem: '<Enter the filesystem used for the mount inside the virtual server.
Required only for non passthrough quotagroup. The value can be btrfs, ext4, or xfs>'
       size: '<Enter the size of volume mount point to be used from quotagroup and
mount to virtual server.  Required only for non passthrough quotagroup. For example,
20GB>'
       reset_root: <This optional variable is used as a flag to reset the rootdisk of
the virtual server during update. Set to true, to reset the specified RUNQ_ROOTDISK
during update. The reset_root option is applicable only for non passthrough quotagroup
and ext4 filesystem. The default value of reset_root is false. >
  ports: '<Define the list of ports for the virtual server. Optional if no ports to be
assigned>'
   - hostport: '<Enter the port exposed from Secure Service Container host>'
     protocol: '<Enter the protocol to be used for port communcication. tcp or udp is
supported. Default is tcp>'
     containerport: '<Enter the port exposed within the virtual server>'
```

The following parameters specified in the example vs_configfile_readme_yml as shown above, are applicable only for IBM Hyper Protect Virtual Servers version 1.2.2, or later.

```
environment:
- key: '<Enter the environment key for ssh connection. For example, "ROOT_SSH_KEY" >'
  value: '<Enter the environment value. For example,
"@/root/hpvs/config/hpvsopbasessh/keys/id_rsa_base64.pub" If value starts with @, the
content of the file will be assigned as value to the key. Here the file is expected to
have ssh key in base64 format> '
- key: '<Enter the environment key. For example, "RUNQ_ROOTDISK" . Specify the root
disk of virtual server using this option. >'
  value: '<Enter the environment value. For example, newroot. This is value of mountid
which is to be assigned as rootdisk">'
  volumes:
     mounts:
       reset_root: <This optional variable is used as a flag to reset the rootdisk of
the virtual server during update. Set to true, to reset the specified RUNQ_ROOTDISK
during update. The default value is false. This feature is applicable only for non
passthrough quotagroup and ext4 filesystem. >
```

Note: The environment parameter **`key: '<Enter the environment key for ssh connection. For example, "ROOT_SSH_KEY" >'`** is applicable only for a virtual server created by using the **`hpvs-op-ssh`** base image.

The following parameters specified in the example vs_configfile_readme_yml as shown above, are applicable for IBM Hyper Protect Virtual Servers versions 1.2.1.1, and 1.2.1.

```
environment:
- key: '<Enter the environment key. For example, "SSH_PUBLIC_KEY" >'
  value: '<Enter the environment value. For example,
"@/root/hpvs/config/hpvsopbasessh/keys/id_rsa.pub" If value starts with @, the content
of the file will be assigned as value to the key.>'
```

Note: The environment parameter `key: '<Enter the environment key for ssh connection. For example, "SSH_PUBLIC_KEY" >'` is applicable only for a virtual server created by using the `hpvs-op-ssh` base image.

If you want to customize network, resources or storage settings, you can edit this file and add the corresponding definitions in the virtual server template file.

- The following example shows a customized the resource definition.

Configuration file entry:

```
resourcedefinition:
   ref: user_own
```

The corresponding template entry:

```
resourcedefinitiontemplates:
-  name: user_own
   cpu: 1
   memory: 4096
```

- The following example shows a customized the network definition

Configuration file entry:

```
networks:
   - ref:  userdefinednetwork
     ipaddress: 192.168.40.111
```

The corresponding template entry:

```
-  name: userdefinednetwork
   subnet: "192.168.40.0/24"
   gateway: "192.168.40.1"
   parent: encf900
   driver: bridge
```

- The following example shows a customized the storage definition

Configuration file entry:

```
volumes:
   - name: userdefnonpassqg
     ref : user-npsmall
     mounts:
      - mount_id: new_qg_hpvsopbasessh
        mountpoint: /newroot
        filesystem: ext4
        size: 10GB
```

The corresponding template entry:

```
-  name: user-npsmall
   size: 20GB
   passthrough: false
```

Secure Build configuration

The securebuild.yaml file defines the configuration of each Secure Build container. The following example is a yaml template, with descriptions of each parameter, that you can fill out with the configuration for your own Secure Build containers.

```
secure_build_workers:
   sbs:
      url: '<url of the secure build service. e.g- https://10.20.4.72>'
      port: '443'
      cert_path: '<complete path of certificate.  e.g- /root/sbs_cert>'
      key_path: '<complete path of key.  e.g- /root/sbs_key'
   regfile:
      id: '<Enter Id. It could be any name>'
   github:
      url: '<git hub url. e.g- git@github.com:MyOrg/my-docker-app.git>'
      branch: 'master'
      ssh_private_key_path: '<complete path of key github private key. e.g - 
/root/git_key>'
      recurse_submodules: 'False'
      dockerfile_path: './Dockerfile'
      docker_build_path: '<Enter the path to the subdirectory within the Github project 
to be used as the build context for the Docker build>'
   docker:
      push_server: '<get this from hpvs registry list. e.g - docker_push>'
      base_server: '<get this from hpvs registry list. e.g - docker_base>'
      pull_server: '<get this from hpvs registry list. e.g - docker_pull>'
      repo: 'docker_user_name/docker_image_name'
      image_tag_prefix: 'latest'
      content_trust_base: 'True'
   manifest_cos:
      bucket_name: '<Enter the bucket name on the S3 object store where manifest files 
will be transferred to after each build>'
      api_key: '<Enter the API key used to authenticate with the S3 object store>'
      resource_crn: '<Enter the resource instance ID for the S3 object store>'
      auth_endpoint: '<Enter the authentication endpoint for the S3 object store>'
      endpoint: '<Enter the endpoint for the S3 object store>'
   env:
      allowlist: []
   build:
      args: []
   signing_key:
      private_key_path: '/root/isv_user.private'
      public_key_path: '/root/isv_user.pub'

   # Add linux capabilities to hyper protect virtual server. List of linux capabilities
   # are available here https://man7.org/linux/man-pages/man7/capabilities.7.html.
   # All the capabilities listed are supported except "CAP_PERFMON", "CAP_BPF", and 
CAP_CHECKPOINT_RESTORE".
   # While adding capabilities remove the prefix "CAP".
   # For example CAP_AUDIT_CONTROL will be AUDIT_CONTROL

   cap_add: [] # eg: ["NET_ADMIN","NET_RAW"], or ["ALL"]
```

Note:

- The `cap_add: ["ALL"]` parameter is applicable for IBM Hyper Protect Virtual Servers version 1.2.3, or later. To enable all privileges' you can use `cap_add:["ALL"]`, but as a good security practice, provide the least possible privileges' to your virtual server.
- Starting with IBM Hyper Protect Virtual Servers version 1.2.4, the term "whitelist" is replaced with "allowlist". For IBM Hyper Protect Virtual Servers versions earlier than 1.2.4, you must use "whitelist" instead of "allowlist".

Create repository registration

```
repository_registration:
   docker:
      repo: 'docker_user_name/docker_image_name'
      pull_server: '<get this from hpvs registry list. e.g - docker_pull>'
      # this root.json you will get after once you will push image to dokcer hub using 
docker content trust
      content_trust_json_file_path: 
'/root/.docker/trust/tuf/docker.io/docker_user_name/docker_image_name/metadata/root.jso
n'
    env:
      allowlist: []
    signing_key:
      private_key_path: '/root/isv_user.private'
      public_key_path: '/root/isv_user.pub'
```

Note: Starting with IBM Hyper Protect Virtual Servers version 1.2.4, the term "whitelist" is replaced with "allowlist". For IBM Hyper Protect Virtual Servers versions earlier than 1.2.4, you must use "whitelist" instead of "allowlist".

Network requirements for IBM Hyper Protect Virtual Servers

Network configuration settings that are required for setting up a Hyper Protect Virtual Server instance.

Bridge types supported on IBM Hyper Protect Virtual Servers

There are two types of bridge network.

- Bridge - This type of bridge network used when virtual servers need to communicate between them on the same Secure Service Container partition.
- Macvlan - Macvlan type of bridge is used when virtual servers need to communicate between them on different Secure Service Container partitions by using the underlying network. It is also used when you require external connectivity.

Internal or external network configuration scenarios

Based on the requirements of your application and the type of connectivity required, following are some of the possible scenarios.

Scenario: Internal network

You can create an internal network using either of the following options.

- By using the **`hpvs deploy`** command.

    1. The following is an example of a virtual server template file, showing options of macvlan and bridge. You can use either macvlan or bridge when creating your internal network.

    ```
    version: v1
    type: virtualserver-template
    networktemplates:
    - name: internal_network
      subnet: "192.168.40.0/24"
      gateway: "192.168.40.1"
      parent: encf500
      driver: bridge
    -  name: internal_network
       subnet: "192.168.56.0/24"
    ```

```
   gateway: "192.168.56.1"
   parent: "encf700"
   driver: "macvlan"
```

2. The following is an example of a virtual server configuration yaml (vs_config.yml) file.

```
networks:
  - ref:  internal_network
    ipaddress: 192.168.40.2
```

3. Create the virtual server by using the configuration in the yaml file.

```
hpvs deploy --config <$path_to_configfile>/vs_config.yml
```

- By using the **`hpvs vs create`** command.

1. Create the network.

```
hpvs network create --name internal_network --driver bridge --parent encf500 
\
--subnet 192.168.40.0/24 --gateway 192.168.40.1
```

2. Create the virtual server.

```
hpvs vs create --name testcontainer  --network "{name = internal_network, ip 
= 192.168.40.2}"
```

Scenario: External network

You can create an external network using either of the following options.

- By using the **`hpvs deploy`** command.

1. The following is an example of a virtual server template file.

```
version: v1
type: virtualserver-template
networktemplates:
- name: external_network
  subnet: "10.20.4.0/22"
  gateway: "10.20.4.1"
  parent: encf500
  driver: macvlan
```

2. The following is an example of a virtual server configuration yaml (vs_config.yml) file.

```
networks:
  - ref:  external_network
    ipaddress: 10.20.4.2
```

3. Create the virtual server by using the configuration in the yaml file.

```
hpvs deploy --config <$path_to_configfile>/vs_config.yml
```

- By using the **`hpvs vs create`** command.

1. Create the network.

```
hpvs network create --name external_network --driver macvlan --parent encf900
\
--subnet 10.20.4.0/22 --gateway 10.20.4.1
```

2. Create the virtual server.

```
hpvs vs create --name testcontainer  --network "{name = external_network, ip 
= 10.20.4.2}"
```

Scenario: External network using port mapping

You can create an external network using port mapping using either of the following options.

- By using the **hpvs deploy** command.

    1. The following is an example of a virtual server configuration yaml (vs_config.yml) file.

       ```
       ports:
         - hostport: 21443
           protocol: tcp
           containerport: 443
       ```

    2. Create the virtual server by using the configuration in the yaml file.

       ```
       hpvs deploy --config <$path_to_configfile>/vs_config.yml
       ```

- By using the **hpvs vs create** command.

    1. Create the virtual server.

       ```
       hpvs vs create --name testcontainer  --ports "{containerport = 443, protocol
       = tcp, hostport = 21443}"
       ```

If you use port mapping for Secure Build virtual server, Monitoring infrastructure, and GREP11 virtual server, ensure that the following ports or configured mapping ports are available on the Secure Service Container partition. Otherwise, you need to request IP address for each virtual server using external network on the Secure Service Container partition.

The following table shows the required ports on the Secure Service Container partition

Port No.	Required by Module
`443`	Hosting Appliance REST API
`443`	Secure Build Server or bring your own image with macvlan
Any non-reserved port	Secure Build Server
`8443`	To access monitoring by Prometheus
`25826`	Used by collectd host
`9876`	GREP11 container

Overview of quotagroups for IBM Hyper Protect Virtual Servers

This topic provides information about the types quotagroups that are supported by Hyper Protect Virtual Servers.

Quotagroup types supported on IBM Hyper Protect Virtual Servers

There are two types of quotagroups.

- Passthrough quotagroup - In a passthrough quotagroup, the quotagroup is attached directly to virtual server. In this case one quotagroup can be attached to one virtual server. Passthrough quotagroups support three types of filesystems: btrfs, ext4, and xfs. Passthrough quotagroups also offer better performance.

- Non-passthrough quotagroup - In a non-passthrough quotagroup, the quotagroup is not directly attached to a virtual server. It is similar to a nested storage capability. The non-passthrough quotagroup is always created with the btrfs filesystem as default. During the creation of a virtual server, the non-passthrough quotagroup is re-formatted according to any of the three options that are available: btrfs, ext4, or xfs. In this case one quotagroup can be shared by multiple virtual servers. So, If you want to create one quotagroup and use it across multiple virtual servers, use non-passthrough quotagroup.

Creating a virtual server using a passthrough quotagroup

1. The following is an example of creating a passthrough quotagroup.

```
hpvs quotagroup create --name pass_vol --size 20GB --passthrough
```

2. The following is an example of creating a virtual server that uses a passthrough quotagroup.

```
hpvs vs create --name vs_pass --repo test_repo --quotagroup "{quotagroup = 
pass_vol, mountid = new,mount = /newroot}"
```

Note: Since a passthrough quotagroup is directly attached to the virtual server, you do not have to specify the size and filesystem type when you want to create a virtual server using a passthrough quotagroup.

Creating a virtual server using non-passthrough quotagroup

1. The following is an example of creating a passthrough quotagroup.

```
hpvs quotagroup create --name nonpass_vol --size 20GB
```

2. The following is an example of creating a virtual server that uses a passthrough quotagroup.

```
hpvs vs create --name vs_nonpass --repo test_repo --quotagroup "{quotagroup = 
nonpass_vol, mountid = new,mount = /newroot, filesystem = ext4, size = 10GB}"
```

Note: When you create a non-passthrough quotagroup, for example of size 20 GB, the available space might be displayed approximately as 18 GB. If you create a virtual server using the same quotagroup (of 18GB), it might fail if the size of quotagroup reduces (which might occur sometimes due race conditions). It is recommended that you use the following command to get details about the available space, and retry creating the virtual server with available space as displayed by the `hpvs quotagroup show` command.

```
hpvs quotagroup show --name=<quotagroup name>
```

Comparison of volumes between pass through and non-passthrough quotagroups

Type	Supported filesystem types	Creating a virtual server	Deleting a virtual server
Passthrough	All filesystem types	The entire volume is mounted	Deletes attached volume
Non-passthrough	Only btrfs	Qutoagroup can be sliced and mounted as raw volumes on a virtual server	Storage is returned to the Qutoagroup

Note:

- The `appliance_data` quotagroup is a default quotagroup of type non-passthrough which is pre existing on the Secure Service Container partition. The default size of this quotagroup is 10GB.
- If you create any virtual server without specifying any quotagroup, then it uses the `appliance_data` quotagroup by default. For example the following command did not specify any quotagroup, so the `appliance_data` quotagroup will be used to create the virtual server.

```
hpvs vs create --name test_VS --repo test_repo
```

- When you create a virtual server by specifying a quotagroup that you created, the root file system of the virtual server uses the `appliance_data` quotagroup. In the following example, the `my_VS` virtual server uses two quotagroups, `my_qg` mounted on /data, and `appliance_data` which is mounted on '/'.

```
hpvs quotagroup create --name my_qg --size 15GB --passthrough
hpvs vs create --name my_VS --repo my_repo --quotagroup "{quotagroup = my_qg, 
mount = /data}"
```

- The default size of the `appliance_data` quotagroup is 10GB, and the virtual server uses all of it for the root file system. Therefore it is recommended that you increase the size of the `appliance_data` quotagroup by running the following command.

  ```
  hpvs quotagroup update --name appliance_data --size 80GB
  ```

- You must monitor the size of the quotagroup that is being consumed by the virtual servers and ensure that there is sufficient available space for the optimal functioning of the virtual servers.

Updating the parameters of IBM Hyper Protect Virtual Servers

This topic provides information about the parameters of virtual servers that can be updated after deployment, by using either the **`hpvs vs update`** command or the **`--update`** flag of the **`hpvs deploy`** command.

Parameters of virtual servers that can be updated

Parameter	Can be updated	Notes on the update	Value of reset_root
cpu	Yes		false
crypto	No		not applicable
domainname	Yes		true
env	Yes		true if the environment parameter is related a content change in the disk
extra_hosts	Yes		true
hostname	Yes		true
labels	Yes		false
log_config	No		not applicable
memory (ram)	Yes		false
name	No		not applicable
networks	Yes		true for updating the IP, false for attaching and detaching networks
ports	Yes		false
quotagroup (volume)	Yes	Can only increase and not decrease the size only by using the **`hpvs deploy`** command with **`-u`** or **`--update`** flag	false
repo (repoid)	Yes		true
repo definition file (reporegfile)	Yes	Can be updated only by using the **`hpvs deploy`** command with **`-u`** or **`--update`** flag	false
tag (imagetag)	Yes	Re-assigning the same tag to different iterations of the same Docker image causes failures and is not a good practice	true

High availability and disaster recovery

The IBM® Hyper Protect Virtual Servers protects workloads that are deployed on IBM Z/LinuxONE (i.e., s390x architecture) servers in hybrid, and multi-cloud environments. The IBM Hyper Protect Virtual Servers might experience an outage during a disaster scenario. Therefore, you can deploy your workload in an active-active mode across multiple Hyper Protect Virtual Servers instances. The active-active mode ensures an operable workload with fault tolerant virtual servers.

Example workloads that you can deploy in an active-active mode are databases (PostgreSQL, MongoDB, or MySQL), or applications with no local state.

If the latency requirements or types of workload does not allow running in an active-active mode, you can perform regular backups from one virtual server to another instance in a different data center. When a disaster occurs, the amount of data that is lost depends on the frequency of the backups, and the time taken to restore a backup.

Backing up and recovering SSH images

IBM Hyper Protect Virtual Servers can backup and recover SSH images. There are primary and recovery virtual server images for each of the application types. The primary and recovery virtual servers reside on separate IBM Z/LinuxONE (i.e., s390x architecture) servers.

Setting up the backup and recovery environment is explained here by using an example, where the Digital Bank and MongoDB applications are built using SSH images by using the Secure Build process, and deployed as virtual server instances.

After the primary and recovery MongoDB instances are created, you must login to the two instances by using SSH to configure the 'rsync over ssh' protocol. You must create and run a cron job to execute **`mongodump`** (dumps data), use **`rsync`** to transfer the dump data from the primary virtual server to the recovery virtual server, and use **`mongorestore`** to restore data on the recovery virtual server.

A load balancer is utilized to alter which virtual server (primary or recovery) a client can access. An example is the IBM Cloud Internet Services (CIS). This provides a unique URL to clients that can be modified to point to the primary public IP address or recovery public IP address. When the primary virtual server goes down, the digital bank application is started on the recovery Secure Service Container LPAR. This is supported by using passthrough or non-passthrough quotagroups.

This procedure is intended for users with the role *cloud administrator* and *infrastructure administrator*.

Before you begin

- Generate and exchange the SSH keys for the virtual server. For more information about generating SSH keys, see Generating SSH keys.
- Deploy the primary and secondary virtual servers by running the **`hpvs deploy`** command. For more information about deploying the virtual servers, see Creating a Hyper Protect Virtual Server instance.
- Install the application on the both primary and recovery virtual servers.
- Acquire the public and private IP addresses of the the primary and secondary virtual servers. The public and internal IP addresses are assigned during the virtual server creation. For more information about how to acquire details about the virtual server, see hpvs vs show command.

Procedure for backup

Complete the following steps.

1. Install **`rsync`** on the primary virtual server. This is a two step process.

- Synchronize the package index files from their sources. Package index files are like local metadata files providing information to the system. The sources for the package index files are defined within the virtual server. Only official Ubuntu repositories are defined by default. It enables retrieving information about available packages, versions, and dependencies.
    - To fetch and update the metadata, run the following command on the primary virtual server (from root):

      ```
      apt-get update
      ```

- Install `rsync` on the primary virtual server.
- Run the following command on the primary virtual server (from root):

  ```
  apt-get install rsync
  ```

2. If required, quiesce the relevant application before the backup operation. This step is optional.

3. To setup a cron job (when this file is created, it will automatically initiate the backup), complete the following tasks.

   - You can schedule a job to copy the contents of the disk to the recovery virtual server instance.
   - Create a cron backup script (for example: cron_backup), and specify the desired frequency of the backup. For example: hourly (max data loss = 1 hour). The following is an example of the cron_backup script.

     ```
     #!/bin/sh
     rsync -a /data <public or internal backup HPVS IP>:/data
     ```

     NOTE:

     - Once the file is created, it is automatically executed by the operating system.
     - The Hyper protect Virtual Server IP (as shown in the cron_backup file) is the public or internal IP address of the Hyper Protect Virtual Server and you must reference it in the cron backup script. Also, provide the primary virtual server’s internal address if the primary and backup virtual servers are connected over the internal network, or provide the primary virtual server’s public address if the primary and backup virtual servers are connected over the internet, in the cron backup script.
4. This is an optional step. To maintain multiple backups external to the virtual server instances created with Hyper Protect Virtual Servers, complete the following tasks.

   - Package the data by using the `tar` command (tar -cvf FILENAME.tar DIRECTORY/).
   - Encrypt the data by using GnuPG (gpg –encrypt).
   - Store the data, for example in the IBM Cloud Object Storage. For more information, see Upload data.
5. Always access the application that should be recoverable by using a URL that points to the virtual server IP address, and never access the IP address directly. You can then adjust the URL to point to the recovery virtual server. The access point is input from a load balancer (example CIS, Cloudflare, F5) or a Domain Name System (DNS).

Procedure for recovery

To recover from a disaster using the backup environment as described in the section above, complete the following steps.

1. Connect to the recovery virtual server instance.
2. Start the application and test whether the application starts successfully.
3. Reconfigure the load balancer or the DNS to map the application URL to the recovery virtual server.
4. Test whether the application is accessible externally, as expected.
5. Test the recovery procedure periodically to ensure its effectiveness.

Backing up and recovering non-SSH images

IBM Hyper Protect Virtual Servers can backup and recover non-SSH images. The primary and recovery virtual servers reside on different Secure Service Container LPARs on the same IBM Z/LinuxONE (i.e., s390x architecture) management server.

Setting up the backup and recovery environment is explained here with an example where the Digital Bank and MongoDB applications are built using non-SSH images by using the Secure Build process, and deployed as virtual server instances. For the digital banking application, deploy the primary and the secondary instance with same image. For the MongoDB application, you must synchronize the application data from the primary to the secondary instance, and because the configurations are different, you must configure two different images for the MongoDB application.

An external load balancer sends a request to the primary virtual server and executes **`mongodump`** (dumps data), **`rsync`** (synchronizes data with the recovery MongoDB instance), and **`mongorestore`**. As this example uses non-ssh base images, **`rsync`** is implemented via a manual deployment of the Dockerfile, so that the appropriate port can be exposed for the Docker container (update the IP table or firewall configuration).

A load balancer is utilized to alter which virtual server (primary or recovery) a client can access. An example is the IBM Cloud Internet Services (CIS). This provides a unique URL to clients that can be modified to point to the primary public IP address or recovery public IP address. When the primary virtual server goes down, the digital bank application is started on the recovery Secure Service Container LPAR. This is supported by using passthrough or non-passthrough quotagroups.

This procedure is intended for users with the role *cloud administrator* and *infrastructure administrator*.

Before you begin

- Deploy two Digital banking application instances that are built by using the Secure Build process and deployed as virtual server instances.
- The cloud administrator acquires the public and private IP addresses of the primary and backup virtual servers from the infrastructure administrator. These are required when you create the virtual server.

Backup procedure

Complete the following steps.

1. Create a Dockerfile for deploying the application. The following is an example Dockerfile for the primary MongoDB server.

```
FROM test4hpvsop/hpvsop-base:1.2.3-release-d0651e4

COPY --chown=root:root config/iptables_rsync.conf /etc/iptables/iptables.conf
COPY --chown=root:root rsync_cron_pri.sh /etc/cron.hourly/rsync_cron_pri
COPY start_rsync_client.sh /root/start.sh
COPY config/rc.local /etc/rc.local

RUN apt-get update && \
    apt-get install -y \
    gnupg \
    rysnc \
    cron \
    wget && \
    wget -qO - https://www.mongodb.org/static/pgp/server-4.4.asc | apt-key add -
&& \
    mkdir -p /etc/apt/sources.list.d && \
    echo "deb [ arch=s390x,s390x ] https://repo.mongodb.org/apt/ubuntu
```

```
bionic/mongodb-org/4.4 multiverse" | tee /etc/apt/sources.list.d/mongodb-org-
4.4.list && \
    apt-get update && \
    ln -s /bin/true /usr/local/bin/systemctl && \
    apt-get install -y -q \
    mongodb-org-server \
    mongodb-org-shell \
    mongodb-org-mongos \
    mongodb-org-tools \
    mongodb-org && \
    /usr/local/bin/systemctl enable mongod && \
    chmod +x /root/start.sh && \
    chmod +x /etc/rc.local && \
    chmod +x /etc/cron.hourly/rsync_cron_pri && \
    rm -f /usr/local/bin/systemctl

COPY config/mongod.conf /etc/mongod.conf
ENV SEC_MONGO_IP ${SEC_MONGO_IP}
ENV ADMIN_PASSWD ${ADMIN_PASSWD}
ENV USER_PASSWD ${USER_PASSWD}
ENV RSYNC_PASSWD ${ RSYNC_PASSWD}

ENTRYPOINT ["/root/start.sh"]
```

The following is an example Dockerfile for the secondary MongoDB server.

```
FROM test4hpvsop/hpvsop-base2:1.2.3-release-d0651e4

COPY --chown=root:root config/iptables_rsync.conf /etc/iptables/iptables.conf
COPY --chown=root:root config/rsyncd.conf /etc/
COPY --chown=root:root rsync_cron_sec.sh /etc/cron.hourly/rsync_cron_sec
COPY start_rsync_server.sh /root/start.sh
COPY config/rc.local /etc/rc.local

RUN apt-get update && \
    apt-get install -y \
    gnupg \
    wget && \
    wget -qO - https://www.mongodb.org/static/pgp/server-4.4.asc | apt-key add -
&& \
    mkdir -p /etc/apt/sources.list.d && \
    echo "deb [ arch=s390x,s390x ] https://repo.mongodb.org/apt/ubuntu
bionic/mongodb-org/4.4 multiverse" | tee /etc/apt/sources.list.d/ mongodb-org-
4.4.list && \
    apt-get update && \
    ln -s /bin/true /usr/local/bin/systemctl && \
    apt-get install -y -q \
    mongodb-org-server \
    mongodb-org-shell \
    mongodb-org-mongos \
    mongodb-org-tools \
    mongodb-org && \
    /usr/local/bin/systemctl enable mongod && \
    chmod +x /root/start.sh && \
    chmod +x /etc/rc.local && \
    chmod +x /etc/cron.hourly/rsync_cron_sec && \
    rm -f /usr/local/bin/systemctl

COPY config/mongod.conf /etc/mongod.conf
ENV PRI_MONGO_IP ${PRI_MONGO_IP}
ENV ADMIN_PASSWD ${ADMIN_PASSWD}
ENV USER_PASSWD ${USER_PASSWD}
ENV RSYNC_PASSWD ${RSYNC_PASSWD}

ENTRYPOINT ["/root/start.sh"]
```

2. Create two scripts for the installation and configuration of MongoDB. The following are examples of script configurations that you can use to create your scripts, based on your environment and the Dockerfile of your application.

   For the rsync client that resides in the primary server, the following is an example script configuration.

```
#!/bin/bash
#configure mongo
echo "start mongo db"
/usr/bin/mongod -f /etc/mongod.conf --fork
echo "add mongodb user"
mongo admin --eval "db.createUser({user:\"admin\", pwd:\"$ADMIN_PASSWD\", roles:
[{role:\"userAdminAnyDatabase\", db:\"admin\"}]})"
mongo test --eval "db.createUser({user:\"test\", pwd:\"$USER_PASSWD\", roles:
[{role:\"readWrite\", db:\"test\"}]})"
echo "stop mongo db"
/usr/bin/mongod -f /etc/mongod.conf --shutdown
sed -i "s/#security/security/" /etc/mongod.conf
sed -i "s/#  authorization/  authorization/" /etc/mongod.conf

#start cron service
service cron start

# configure rsync client with rsync protocal
sed -i "s/<server_ip>/$SEC_MONGO_IP/" /etc/cron.hourly/rsync_cron_pri
echo "$RSYNC_PASSWD" > /data/rsync_passwd
chmod 600 /data/rsync_passwd
mkdir /data/dump
#start system services including mongo
exec /sbin/init
```

   Note: The MongoDB name is 'test' in the example.

   For the rsync server that resides in the recovery server, the following is an example script configuration.

```
#!/bin/bash
#configure mongo
echo "start mongo db"
/usr/bin/mongod -f /etc/mongod.conf --fork
echo "add mongodb user"
mongo admin --eval "db.createUser({user:\"admin\", pwd:\"$ADMIN_PASSWD\", roles:
[{role:\"userAdminAnyDatabase\", db:\"admin\"}]})"
mongo test --eval "db.createUser({user:\"test\", pwd:\"$USER_PASSWD\", roles:
[{role:\"readWrite\", db:\"test\"}]})"
echo "stop mongo db"
/usr/bin/mongod -f /etc/mongod.conf --shutdown
sed -i "s/#security/security/" /etc/mongod.conf
sed -i "s/#  authorization/  authorization/" /etc/mongod.conf

# start cron service
service cron start

# configure rsync server with rsync protocal
sed -i "s/<pri_mongo_ip>/$PRI_MONGO_IP/" /etc/rsyncd.conf
sed -i "s/<user_passwd>/$USER_PASSWD/" /etc/cron.hourly/rsync_cron_sec
echo "rsync_backup:$RSYNC_PASSWD" > /data/rsync.password
chmod 600 /data/rsync.password
mkdir /data/dump
#start system services including mongo
exec /sbin/init
```

   Note: The MongoDB name is 'test' in the example.

   The following is an example of the rsync server configuration file.

```
uid = root
gid = root
use chroot = no
max connections = 1
timeout = 300
pid file = /data/rsyncd.pid
lock file = /data/rsync.lock
log file = /data/rsyncd.log
ignore errors
read only = false
list = false
hosts allow = <pri_mongo_ip>/24
hosts deny = 0.0.0.0/32
auth users = rsync_backup
secrets file = /data/rsync.password
[backup]
comment = "backup db files"
path = /data/dump
```

3. To setup a cron job (when this file is created, it automatically initiate the backup), complete the following tasks.

    - You can schedule a job to copy the contents of the disk to the recovery virtual server instance.
    - Create a cron backup script (for example: cron_backup.sh), and specify the desired frequency of the backup. For example: hourly (max data loss = 1 hour).

    NOTE: Once the file is created, it is automatically executed by the operating system.

    The following is an example of the cron job on the primary Mongo instance.

    ```
    #!/bin/sh
    mongodump --host 127.0.0.1 --port 27017 -u test -p <USER_PASSWORD> -d test -o
    /data/dump/
    rsync -avz -P /data/dump/  rsync_backup@<server_ip>::backup --password-
    file=/data/rsync_passwd
    ```

    The following is an example of the cron job on the secondary Mongo instance.

    ```
    #!/bin/sh
    mongorestore --host 127.0.0.1 --port 27017 -u test -p <user_passwd> -d test
    /data/dump/test
    ```

    NOTE: The Hyper protect Virtual Server IP (as shown in the cron job example of the primary Mongo instance) is the public or internal IP address of the secondary Hyper Protect Virtual Server. In the cron script, provide the primary virtual server's internal address if the primary and backup virtual servers are connected over the internal network, or provide the primary virtual server's public address if the primary and backup virtual servers are connected over the internet.

4. Create the configuration files for MongoDB deployment and configuration. It is recommended that you place the MongoDB data in '/data' which is mounted in an external quotagorup. The following is an example of a MongoDB configuration file.

    ```
    # mongod.conf
    # for documentation of all options, see:
    #   http://docs.mongodb.org/manual/reference/configuration-options/
    # Where and how to store data.
    storage:
      dbPath: /data
      journal:
        enabled: true
    #  engine:
    #  mmapv1:
    #  wiredTiger:
    # where to write logging data.
    systemLog:
      destination: file
    ```

```
  logAppend: true
  path: /data/mongod.log
# network interfaces
net:
  port: 27017
  bindIp: 0.0.0.0
# how the process runs
processManagement:
  timeZoneInfo: /usr/share/zoneinfo
#security:
#  authorization: enabled
#operationProfiling:
#replication:
#sharding:
## Enterprise-Only Options:
#auditLog:
#snmp:
```

The following is an example of the `rc.local` file that is used to configure MongoDB as a system service.

```
#!/bin/bash -e
/usr/bin/mongod -f /etc/mongod.conf
```

The following is an example of the file that is used for iptables for the firewall setting (expose only MongoDB and rsync ports).

```
# originally generated by iptables-save
# modifications for basic networking protection while maintaining typical access avenues
*filter
:INPUT DROP [4:180]
:FORWARD DROP [0:0]
:OUTPUT DROP [0:0]
-A INPUT -i lo -j ACCEPT
-A INPUT -s 127.0.0.0/8 -j DROP
-A INPUT -p tcp -m state --state ESTABLISHED -j ACCEPT
-A INPUT -p udp -m state --state ESTABLISHED -j ACCEPT
-A INPUT -p icmp -m state --state ESTABLISHED -j ACCEPT
#
-A OUTPUT -o lo -j ACCEPT
-A OUTPUT -p tcp -m state --state NEW,ESTABLISHED -j ACCEPT
-A OUTPUT -p udp -m state --state NEW,ESTABLISHED -j ACCEPT
-A OUTPUT -p icmp -m state --state NEW,ESTABLISHED -j ACCEPT
# Open mongodb Port (27017)
-A INPUT -p tcp -m tcp --dport 27017 -j ACCEPT
# Open rsync Port (873)
-A INPUT -p tcp -m tcp --dport 873 -j ACCEPT
#
COMMIT
```

5. Create two separate Secure Build Server instances for the primary and secondary MongoDB servers. For more information, see steps 1 and 2 from the topic Building your application with the Secure Build virtual server.

6. Build two images for the primary and the secondary application servers by using the **`hpvs sbs init`** and **`hpvs sbs build`** commands. For more information, see steps 3 and 4 from the topic Building your application with the Secure Build virtual server.

7. Deploy the primary and recovery virtual servers by using the **`hpvs deploy`** command. For more information, see step 5 from the topic Building your application with the Secure Build virtual server.

8. Always access the application that should be recoverable by using a URL that points to the virtual server IP address, and never access the IP address directly. You can then adjust the URL to point to the recovery virtual server. The access point is input from a load balancer (example CIS, Cloudflare, F5) or a Domain Name System (DNS).

Procedure for recovery

To recover from a disaster using the backup environment as described in the section above, complete the following steps.

1. Connect to the recovery virtual server instance and verify whether the application is up and the backup data is available.
2. Reconfigure the DNS to map the application URL to the recovery virtual server via the load balancer.
3. Test whether the application is accessible externally, as expected.
4. Test the recovery procedure periodically to ensure its effectiveness.

Backing up and recovering non-SSH images by using BYOI

IBM Hyper Protect Virtual Servers can backup and recover non-SSH images by using the Bring Your Own Image (BYOI) function. The primary and recovery virtual servers reside on different Secure Service Container LPARs on the same IBM Z/LinuxONE (i.e., s390x architecture) management server.

Setting up the backup and recovery environment is explained here with an example application and PostgreSQL database deployed by using BYOI. PostgreSQL has its own backup tool, for example Bucardo, and does not require a separate file synchronize function like **rsync**. Bucardo can monitor the primary virtual server and synchronize with the recovery virtual server as needed. A separate virtual server built by using the 'hpvs-op-ssh' image is deployed in the recovery Secure Service Container LPAR to host Bucardo.

A client connects to the primary virtual server application, and the primary application uses the application URL to connect to the load balancer, and the load balancer redirects the PostgreSQL IP to the primary PostgreSQL virtual server. If the primary PostgreSQL is down, the app URL will disconnect and reconnect to the recovery PostgreSQL virtual server. This is supported by using passthrough or non-passthrough quotagroups.

This procedure is intended for users with the role *cloud administrator* and *infrastructure administrator*.

Before you begin

- The cloud administrator acquires the public and private IP addresses of the primary and backup virtual servers from the infrastructure administrator. These are required when you create the virtual server.
- Deploy the application server.
- The client is able to connect to the application server.
- The application server configures and connects to the Postgre database via the URL provided by the load balancer.

Backup procedure

Complete the following steps.

1. Create a Dockerfile which includes Bucardo related configuration. The following is an example Dockerfile.

```
FROM test4hpvsop/hpvsop-base:1.2.3-release-d0651e4

COPY --chown=root:root scripts/start.sh /usr/bin/start.sh
COPY --chown=root:root config/iptables.conf /etc/iptables/
COPY --chown=root:root scripts/initdb.sql /etc/
RUN apt-get update && \
    apt-get install -y postgresql-10 postgresql-contrib libpq-dev postgresql-
server-dev-10 postgresql-plperl-10 \
    libdbix-safe-perl libtest-simple-perl libboolean-perl libextutils-makemaker-
cpanfile-perl \
    libextutils-modulemaker-perl libcgi*-perl libdbd-pg-perl libencode-locale-perl 
libpod-parser-perl \
    libsys-syslog-perl vim sudo iputils-ping net-tools netcat bucardo && \
    echo "listen_addresses = '*'" >> /etc/postgresql/10/main/postgresql.conf && \
    cat /etc/postgresql/10/main/postgresql.conf && \
    echo "host all bucardo 0.0.0.0/0 trust" >> /etc/postgresql/10/main/pg_hba.conf 
&& \
    cat /etc/postgresql/10/main/pg_hba.conf && \
    mkdir -p /var/run/bucardo && mkdir -p /var/log/bucardo && \
    echo '*:5432:*:bucardo:bucardobucardo' > /root/.pgpass && chmod 600 
/root/.pgpass && \
    chmod a+x /usr/bin/start.sh
ENV PASSWD ${PASSWD}
ENV BUCARDO_PASSWD ${BUCARDO_PASSWD}

CMD ["/usr/bin/start.sh"]
```

The following is an example of the **start.sh** file.

```
#!/bin/bash
/etc/init.d/postgresql restart
echo "ALTER USER postgres WITH PASSWORD '$PASSWD';" | sudo -u postgres psql
sudo passwd -d postgres
echo -e "$PASSWD\n$PASSWD" | sudo -u postgres passwd
echo "Init Account..."
sudo -i -u postgres psql -c "create user bucardo with superuser password 
'$BUCARDO_PASSWD';"
echo "Init Database..."
sudo -i -u postgres psql -c "create database bucardodb with owner = bucardo;"
echo "Init Table"
sudo -i -u postgres psql -d bucardodb -f /tmp/initdb.sql
echo "Verify Database..."
sudo -i -u postgres psql -d bucardodb -c "select * from tmp_t0;"
exec /sbin/init
```

The following **initdb.sql** file is an example initialization of the Postgre database

```
create table tmp_t0(c0 bigint,c1 varchar(100));
alter table tmp_t0 add primary key(c0);
insert into tmp_t0
select id, md5(id::varchar) from generate_series(1,10) as id;
```

The following is an example for configuring iptables.

```
# originally generated by iptables-save
# modifications for basic networking protection while maintaining typical access 
avenues
*filter
:INPUT DROP [4:180]
:FORWARD DROP [0:0]
:OUTPUT DROP [0:0]
-A INPUT -i lo -j ACCEPT
-A INPUT -s 127.0.0.0/8 -j DROP
-A INPUT -p tcp -m state --state ESTABLISHED -j ACCEPT
-A INPUT -p udp -m state --state ESTABLISHED -j ACCEPT
-A INPUT -p icmp -m state --state ESTABLISHED -j ACCEPT
#
-A OUTPUT -o lo -j ACCEPT
-A OUTPUT -p tcp -m state --state NEW,ESTABLISHED -j ACCEPT
-A OUTPUT -p udp -m state --state NEW,ESTABLISHED -j ACCEPT
-A OUTPUT -p icmp -m state --state NEW,ESTABLISHED -j ACCEPT

# Open postgreSQL Port (5432)
-A INPUT -p tcp -m tcp --dport 5432 -j ACCEPT

COMMIT
```

2. Build an image for Postgre database and deploy two IBM Hyper Protect Virtual Servers instances by using the image that you built. For more information, see Deploying your applications securely.

3. Create a IBM Hyper Protect Virtual Servers instance by using the 'hpvs-op-ssh' image. For more information, see Creating a Hyper Protect Virtual Server instance.

4. Access the IBM Hyper Protect Virtual Servers instance you created in the previous step via an SSH terminal, and deploy Bucardo on the IBM Hyper Protect Virtual Servers instance. For more information, see Bucardo.

5. Always access the application that should be recoverable by using a URL that points to the virtual server IP address, and never access the IP address directly. You can then adjust the URL to point to the recovery virtual server. The access point is input from a load balancer (example CIS, Cloudflare, F5) or a Domain Name System (DNS).

Procedure for recovery

To recover from a disaster using the backup environment as described in the section above, complete the following steps.

1. Connect to the recovery virtual server instance and verify whether the database application is up and the backup data is available.
2. Reconfigure the DNS to map the application URL to the recovery Postgre database instance via the load balancer.
3. Test whether the application is accessible externally, as expected.
4. Test the recovery procedure periodically to ensure its effectiveness.

Gathering Information for IBM Support

The IBM Hyper Protect Virtual Servers Version 1.2.1, or later, provides an automated script to collect debug information when you want to open a support ticket. The script gathers useful information like logs, configuration files and Secure Service Container concurrent dump and creates an archive file to upload on the IBM support portal. This archive file helps to debug the issue.

This procedure is intended for users with the role *cloud administrator*.

Before you begin

- Check that you have IBM Hyper Protect Virtual Servers installation binary on the x86 or Linux on IBM Z/LinuxONE (i.e., s390x architecture) management server. For more information, see Downloading IBM Hyper Protect Virtual Servers.
- Ensure that you set the correct host before you run the mustgather script.

Procedure

On your x86 or Linux on IBM Z/LinuxONE (i.e., s390x architecture) management server, complete the following steps with root user authority.

1. Run the `mustgather.sh` shell script to gather information that is useful when you want to open a support ticket.

   ```
   #sh mustgather.sh
   ```

2. You are prompted to enter the following information.

- The Secure Service Container LPAR IP address, for example `10.20.4.23`.
- The Secure Service Container LPAR Username, for example **`blockchain`**.
- The Secure Service Container LPAR password.

When the script is running, you might see a display that is similar to the following.

```
*****************************************************************************

HPVS details, network configuration and storage configurations are collected
LPAR concurrent dump is triggered. Please wait...
LPAR dump is getting downloaded. It may take few minutes for processing.
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                             Dload  Upload   Total   Spent    Left  Speed
100 17.1M    0 17.1M    0     0  94.1M      0 --:--:-- --:--:-- --:--:--
94.1M

*****************************************************************************

Please upload /root/hpvs/hpvs_logs_2020_06_27.tar.gz file on IBM support
portal.
```

Next

The generated log file must be uploaded to the IBM Support Site when you raise a support ticket. See IBM Support.

OpenSSL configuration examples

You can use the following example files with the **`openssl`** command if you want to avoid entering the values for each parameter required when creating certificates.

Note: You must update the configuration files with the actual values for your environment. For more information, see Creating CA signed certificates.

The sample configuration file to generate the Root CA certificate

```
[ ca ]
default_ca = CA_LOC

[ CA_LOC ]
prompt            = no
dir               = /home/myuser/ca
certs             = $dir/certs
crl_dir           = $dir/crl
new_certs_dir     = $dir/newcerts
database          = $dir/index.txt
serial            = $dir/serial
RANDFILE          = $dir/private/.rand
private_key       = $dir/private/myrootCA.key
certificate       = $dir/certs/myrootCA.crt
crlnumber         = $dir/crlnum
crl               = $dir/crl/mycrl.pem
default_crl_days  = 30
preserve          = no
policy            = policy
default_days      = 365

[ policy ]
commonName              = supplied
stateOrProvinceName     = supplied
```

```
countryName             = supplied
emailAddress            = supplied
organizationName        = supplied
organizationalUnitName  = supplied

[ req ]
default_bits        = 4096
distinguished_name  = req_distinguished_name

string_mask         = utf8only
default_md          = sha256
x509_extensions     = v3_ca

[ req_distinguished_name ]
countryName                     = AB
stateOrProvinceName             = CD
localityName                    = EF_GH
organizationName              = myorg
organizationalUnitName          = myorgunit
commonName                      = mycn
emailAddress                    = myemail@example.com

[ v3_ca ]
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid:always,issuer
basicConstraints = critical, CA:true
keyUsage = critical, digitalSignature
```

The sample configuration file to generate the CSR for a server certificate

```
[ req ]
prompt                 = no
days                   = 365
distinguished_name     = req_distinguished_name
req_extensions         = v3_req


[ req_distinguished_name ]
countryName            = AB
stateOrProvinceName    = CD
localityName           = EFG_HIJ
organizationName       = MyOrg
organizationalUnitName = MyOrgUnit
commonName             = mycommname.com
emailAddress           = emailaddress@myemail.com

[ v3_req ]
basicConstraints       = CA:false
extendedKeyUsage       = serverAuth
subjectAltName         = @sans

[ sans ]
DNS.0 = localhost
DNS.1 = myexampleserver.com
```

The sample configuration file to generate the CSR for a Client certificate

```
[ req ]
```

```
prompt                 = no
days                   = 365
distinguished_name     = req_distinguished_name
req_extensions         = v3_req


[ req_distinguished_name ]
countryName            = AB
stateOrProvinceName    = CD
localityName           = EFG_HIJ
organizationName       = MyOrg
organizationalUnitName = MyOrgUnit
commonName             = mycommname.com
emailAddress           = emailaddress@myemail.com

[ v3_req ]
basicConstraints       = CA:false
extendedKeyUsage       = clientAuth
subjectAltName         = @sans

[ sans ]
DNS.0 = localhost
DNS.1 = myexampleclient.com
```

Others

The following topics list other miscellaneous topics that you can refer to when using IBM Hyper Protect Virtual Servers.

- Security of IBM Hyper Protect Virtual Servers
- List of docker images in the IBM Hyper Protect Virtual Servers
- List of keys used during the Secure Build
- Metrics collected by the monitoring infrastructure

Security of IBM Hyper Protect Virtual Servers

IBM Hyper Protect Virtual Servers provides various security advantages by using the IBM Secure Service Container as the hosting environment.

IBM Secure Service Container is designed to support the deployment of software container technology without requiring application changes to leverage the security capabilities. This is especially useful considering the regulatory focus on protecting critical data from internal and external threats. For example:

- The infrastructure and data are protected against access and abuse by root users, system administrator credentials and other privileged user access.
- Infrastructure management organizations can manage the physical IT infrastructure without having visibility to the end-user's applications and customer data.

As a system or appliance administrator who manages the underlying infrastructure, you can simply download the appliance, deploy it, and then make it available on your system for your developers.

As a developer, you can focus on creating your dockerized solution and deploy it into this environment, and still know that your docker solution is not visible to the system or appliance administrator.

Security mechanisms

Various security mechanisms are also applied to protect the data in the IBM Hyper Protect Virtual Servers.

- Persistence data is encrypted by using the automatic file system encryption technology Linux Unified Key Setup (LUKS). The encryption keys are stored within appliances and not accessible to administrators, and keys are managed based on the appliance lifecyle. The Docker container data mounted to disk is also encrypted.
- In-flight data is encrypted by using the automatic network encryption technology Transport Layer Security (TLS). Data is transferred through encrypted management REST API interfaces among Secure Service Container partitions.
- Diagnostic data is encrypted, which includes first-failure data capture (FFDC) data required to fix problems, Dump data including log message buffers, and so on. Such data is only accessible to the service team.
- Operating system access to the underlying Hyper Protect Virtual Servers hosting appliance is prohibited, and back doors to this host level are eliminated because SSH is disabled on the Secure Service Container partitions by default. Access to the cluster nodes are via SSH keys that are protected by the cloud administrator. Users traditionally with OS access are not allowed to access application data and customer data.

Encryption algorithms

Encryption algorithms used for storage and data transport are provided by the IBM Secure Service Container in the offering.

The web server of IBM Secure Service Container is nginx. The following table contains the utilized subset (default) of cryptographic capabilities of the Secure Service Container web server.

Table 1. Cryptographic capabilities of the Secure Service Container web server

openSSL ciphers	Protocol	Key Exchange	Authentication	Encryption	MAC
ECDHE-RSA-AES256-GCM-SHA384	TLSv1.2	Kx=ECDH	Au=RSA	Enc=AESGCM(256)	Mac=AEAD
ECDHE-ECDSA-AES256-GCM-SHA384	TLSv1.2	Kx=ECDH	Au=ECDSA	Enc=AESGCM(256)	Mac=AEAD
ECDHE-RSA-AES256-SHA384	TLSv1.2	Kx=ECDH	Au=RSA	Enc=AES(256)	Mac=SHA384
ECDHE-ECDSA-AES256-SHA384	TLSv1.2	Kx=ECDH	Au=ECDSA	Enc=AES(256)	Mac=SHA384
DHE-RSA-AES256-GCM-SHA384	TLSv1.2	Kx=DH	Au=RSA	Enc=AESGCM(256)	Mac=AEAD
DHE-RSA-AES256-SHA256	TLSv1.2	Kx=DH	Au=RSA	Enc=AES(256)	Mac=SHA256
ECDH-RSA-AES256-GCM-SHA384	TLSv1.2	Kx=ECDH/RSA	Au=ECDH	Enc=AESGCM(256)	Mac=AEAD
ECDH-ECDSA-AES256-GCM-SHA384	TLSv1.2	Kx=ECDH/ECDSA	Au=ECDH	Enc=AESGCM(256)	Mac=AEAD
ECDH-RSA-AES256-SHA384	TLSv1.2	Kx=ECDH/RSA	Au=ECDH	Enc=AES(256)	Mac=SHA384
ECDH-ECDSA-AES256-SHA384	TLSv1.2	Kx=ECDH/ECDSA	Au=ECDH	Enc=AES(256)	Mac=SHA384
AES256-GCM-SHA384	TLSv1.2	Kx=RSA	Au=RSA	Enc=AESGCM(256)	Mac=AEAD
AES256-SHA256	TLSv1.2	Kx=RSA	Au=RSA	Enc=AES(256)	Mac=SHA256
ECDHE-RSA-AES128-GCM-SHA256	TLSv1.2	Kx=ECDH	Au=RSA	Enc=AESGCM(128)	Mac=AEAD
ECDHE-ECDSA-AES128-GCM-SHA256	TLSv1.2	Kx=ECDH	Au=ECDSA	Enc=AESGCM(128)	Mac=AEAD

openSSL ciphers	Protocol	Key Exchange	Authentication	Encryption	MAC
ECDHE-RSA-AES128-SHA256	TLSv1.2	Kx=ECDH	Au=RSA	Enc=AES(128)	Mac=SHA256
ECDHE-ECDSA-AES128-SHA256	TLSv1.2	Kx=ECDH	Au=ECDSA	Enc=AES(128)	Mac=SHA256
DHE-RSA-AES128-GCM-SHA256	TLSv1.2	Kx=DH	Au=RSA	Enc=AESGCM(128)	Mac=AEAD
DHE-RSA-AES128-SHA256	TLSv1.2	Kx=DH	Au=RSA	Enc=AES(128)	Mac=SHA256
ECDH-RSA-AES128-GCM-SHA256	TLSv1.2	Kx=ECDH/RSA	Au=ECDH	Enc=AESGCM(128)	Mac=AEAD
ECDH-ECDSA-AES128-GCM-SHA256	TLSv1.2	Kx=ECDH/ECDSA	Au=ECDH	Enc=AESGCM(128)	Mac=AEAD
ECDH-RSA-AES128-SHA256	TLSv1.2	Kx=ECDH/RSA	Au=ECDH	Enc=AES(128)	Mac=SHA256
ECDH-ECDSA-AES128-SHA256	TLSv1.2	Kx=ECDH/ECDSA	Au=ECDH	Enc=AES(128)	Mac=SHA256
AES128-GCM-SHA256	TLSv1.2	Kx=RSA	Au=RSA	Enc=AESGCM(128)	Mac=AEAD
AES128-SHA256	TLSv1.2	Kx=RSA	Au=RSA	Enc=AES(128)	Mac=SHA256

Note:

Authenticated Encryption with Associated Data (AEAD) is not a hash function. AEAD is an implicit integrity check in AEAD ciphers (for example, AESGCM). Therefore you can declare AESGCM ciphers as:

- Algorithm Application: Data Encryption, Integrity Check
- Type: Encryption Algorithm

Table 2. AEAD algorithm application and type

Purpose	Protocol	Algorithm Application	Type	Name	Value
SSL (secure data transmission)	TLS V1.2	Data Encryption, Integrity Check	Encryption Algorithm	AES-GCM	256

Appliance Component Communication

This table only lists the utilized subset of cryptographic capabilities supported by GnuPG. See The GNU Privacy Guard for more information about GnuPG.

Table 3. Subset of cryptographic capabilities supported by GnuPG

Purpose	Protocol	Algorithm Application	Type	Name	Value
Data Encryption (GnuPG)	OpenPGP	Data Encryption	Encryption Algorithm	AES	256
Data Encryption (GnuPG)	OpenPGP	Key Exchange	Encryption Algorithm	RSA	4096
Data Encryption (GnuPG)	OpenPGP	Authenticity	Encryption Algorithm	RSA	4096
Data Encryption (GnuPG)	OpenPGP	Integrity Check	Hash Function	MD5	128
Data Encryption (GnuPG)	OpenPGP	Integrity Check	Hash Function	SHA-1	160

Purpose	Protocol	Algorithm Application	Type	Name	Value
Data Encryption (GnuPG)	OpenPGP	Integrity Check	Hash Function	SHA-2	512

Additional Information: The currently used cipher for AES under GnuPG is CFB.

Filesystem Encryption

This table only lists the utilized subset of cryptographic capabilities supported by **`cryptsetup`** or **`dm-crypt`** system.

Table 4. Subset of cryptographic capabilities supported by **`cryptsetup`** or **`dm-crypt`**

Purpose	Protocol	Algorithm Application	Type	Name	Value
Filesystem Encryption	LUKS	Data Encryption	Encryption Algorithm	AES	256
Filesystem Encryption	OpenPGP	Passphrase Exchange	Encryption Algorithm	RSA	4096

List of docker images in the IBM Hyper Protect Virtual Servers

Table 1. The full list of the docker image files in the IBM Hyper Protect Virtual Servers

Image file Name	Location	Used by which command(s)	Description	More information
`HpvsopBaseSSH.tar.gz`	**`<installation_directory>/VS/hpvs-cli/config/<destination-folder-HpvsopBaseSSH>`**	**`docker load`**	Base image for the Hyper Protect Virtual Server container with SSH daemon	Setting up the environment by using the setup script or Registering base images in the remote registry server
`HpvsopBase.tar.gz`	**`<installation_directory>/VS/hpvs-cli/config/<destination-folder-HpvsopBase>`**	**`docker load`**	Base image for the Hyper Protect Virtual Server container without SSH daemon	Setting up the environment by using the setup script or Registering base images in the remote registry server
`SecureDockerBuild.tar.gz`	**`<installation_directory>/VS/securebuild-cli/config`**	**`docker run`** + **`image load`**	Base image for the Secure Build container	Building your application with the Secure Build virtual server
`CollectdHost.tar.gz`	**`<installation_directory>/VS/monitoring-cli/config`**	**`docker run`** + **`monitoring create`**	Base image for the **`collect-host`** container of the monitoring infrastructure	Creating the monitoring virtual servers
`Monitoring.tar.gz`	**`<installation_directory>/VS/monitoring-cli/config`**	**`docker run`** + **`monitoring create`**	Base image for the **`monitoring-host`** container of the monitoring infrastructure	Creating the monitoring virtual servers

Image file Name	Location	Used by which command(s)	Description	More information
`hpcsKpGrep11_runq.tar.gz`	`<installation_directory>/VS/grep11-cli/config`	`docker run` + `grep11 create`	Base image for the grep11 container	[Creating the GREP11 container]()

List of keys used during the Secure Build

Table 1. The full list of the keys used during the Secure Build and BYOI lifecycle.

Key Name	Key Function	Private Key Location	How Created	Owned by Whom
Image Signing Key	Pushing Docker images to a Docker repository	Encrypted volume on the Secure Build container	Created by the remote registry server on first push to the remote repository, and written to Secure Build container	ISV or application developer
Manifest Signing Key	Signing a manifest created by Secure Build	Encrypted volume on the Secure Build container	Created by the Secure Build container when an image is built	ISV or application developer
Client certificate and Key	Used by the cloud administrator to securely interact with the Secure Build REST API, contains certificate and private key	Client	Created on creation of the Secure Build container and provided to the client as the file specified in their CLIENT_CRT_KEY setting	Cloud administrator

Metrics collected by the monitoring infrastructure

The following table shows the metrics collected by the monitoring infrastructure provided in IBM Hyper Protect Virtual Servers .

Table 1. Monitoring metrics collected

#	Plugin Name	Metrics Name	Labels	Description
01	[collectd]()	collectd_collectd_cache_size	collectd="cache", instance	The number of elements in the metric cache.
02	[collectd]()	collectd_collectd_derive_total	collectd="write_queue", type="dropped", instance	The number of metrics dropped due to a queue length limitation.
03	[collectd]()	collectd_collectd_queue_length	collectd="write_queue", instance	The number of metrics currently in the write queue.
04	[cpu]()	collectd_cpu_percent	cpu="idle", instance	Percentage of time that the CPU or CPUs were idle and the system did not have an outstanding disk I/O request.

#	Plugin Name	Metrics Name	Labels	Description
05	cpu	collectd_cpu_percent	cpu="interrupt", instance	Percentage of time spent by the CPU or CPUs to service hardware interrupts.
06	cpu	collectd_cpu_percent	cpu="nice", instance	Percentage of time spent by the CPU or CPUs to run a niced guest. Nice is when the CPU is executing a user task having below-normal priority.
07	cpu	collectd_cpu_percent	cpu="softirq", instance	Percentage of time spent by the CPU or CPUs to service software interrupts.
08	cpu	collectd_cpu_percent	cpu="steal", instance	Percentage of time spent in involuntary wait by the virtual CPU or CPUs while the hypervisor was servicing another virtual processor.
09	cpu	collectd_cpu_percent	cpu="system", instance	Percentage of CPU utilization while the CPU is running kernel code. This includes device drivers and kernel modules.
10	cpu	collectd_cpu_percent	cpu="user", instance	Percentage of CPU utilization while the CPU is running code in user-mode. This includes your application code.
11	cpu	collectd_cpu_percent	cpu="wait", instance	Percentage of time when the CPU or CPUs were waiting for an I/O operation to complete, and the CPU can’t be used for anything else.
12	df	collectd_df_percent_bytes	df= **`<MountPoint>`**, type="free", instance	Free disk space on the file system, expressed as a percentage. MountPoints: root, /hostfs/var/lib/quotagroups/lv_data_pool/appliance_data
13	df	collectd_df_percent_bytes	df= **`<MountPoint>`**, type="reserved", instance	Reserved disk space on the filesystem, expressed as a percentage. MountPoints: root, /hostfs/var/lib/quotagroups/lv_data_pool/appliance_data
14	df	collectd_df_percent_bytes	df= **`<MountPoint>`**, type="used", instance	Used disk space on the file system, expressed as a percentage. MountPoints: root, /hostfs/var/lib/quotagroups/lv_data_pool/appliance_data
15	load	collectd_load_longterm	load="relative", instance	The average system load over a period of the last 15 minutes.
16	load	collectd_load_midterm	load="relative", instance	The average system load over a period of the last 5 minutes.
17	load	collectd_load_shortterm	load="relative", instance	The average system load over a period of 1 minute.
18	memory	collectd_memory	memory="buffered", instance	Amount of memory used for buffering, mostly for I/O operations.
19	memory	collectd_memory	memory="cached", instance	Memory used for caching disk data for reads, memory-mapped files or tmpfs data.
20	memory	collects_memory	memory="free", instance	Total amount of unused memory.

#	Plugin Name	Metrics Name	Labels	Description
21	memory	collectd_memory	memory="slab_recl", instance	Amount of reclaimable memory used for slab kernel allocations.
22	memory	collectd_memory	memory="slab_unrecl", instance	Amount of unreclaimable memory used for slab kernel allocations.
23	memory	collectd_memory	memory="used", instance	Total amount of memory used.
24	memory	collectd_memory_percent	memory="buffered", instance	Amount of memory used for buffering, mostly for I/O operations.
25	memory	collectd_memory_percent	memory="cached", instance	Memory used for caching disk data for reads, memory-mapped files or tmpfs data.
26	memory	collects_memory_percent	memory="free", instance	Total amount of unused memory.
27	memory	collectd_memory_percent	memory="slab_recl", instance	Amount of reclaimable memory used for slab kernel allocations.
28	memory	collectd_memory_percent	memory="slab_unrecl", instance	Amount of unreclaimable memory used for slab kernel allocations.
29	memory	collectd_memory_percent	memory="used", instance	Total amount of memory used.
30	uptime	collectd_uptime	instance	Seconds since system boot.

About error messages in Hyper Protect Virtual Servers

This topic details the error code format for IBM Hyper Protect Virtual Servers CLI.

Error Code Format for IBM Hyper Protect Virtual Servers CLI

The following is the Error Code Format IBM Hyper Protect Virtual Servers CLI. You might encounter these messages when any IBM Hyper Protect Virtual Servers CLI command fails:
HVS-XXYYZZZ
where:

Variable	Meaning of the variable
HVS	Error code reserved for IBM Hyper Protect Virtual Servers.
XX	The hpvs command. Example: hpvs vs, hpvs image, hpvs crypto.
YY	Subcommand of a given hpvs command. Example: hpvs vs create, hpvs image list. When the same error is produced by multiple subcommands, it is displayed as YY.
ZZZ	Error Number. Example: 001, 001.

For more information about various error messages that originate from the IBM Hyper Protect Virtual Servers CLI, see Error messages of IBM Hyper Protect Virtual Servers

Messages of IBM Hyper Protect Virtual Servers

This reference information provides additional information about messages you might encounter while using the IBM Hyper Protect Virtual Servers Version 1.2.1. It is organized according to the identifier of the command that produces the message.

Crypto command messages

This section lists the messages you might encounter while using the crypto commands.

- **HVS-CYLI001: The command to list crypto failed due to an internal server error**
  Explanation: An internal server error occurred resulting crypto list details.
  System Action: List Crypto command execution fails.
  User Action: Ensure that the Crypto Express cards are properly connected. If the problem persists, obtain an appliance and LMS dump and, contact IBM Support.

Deploy command messages

This section lists the messages you might encounter while running the deploy command.

- **HVS-DEPL001: Error while reading the %s config file. Provide valid details in the config file**
  Explanation: The deploy operation failed as there is an error while reading the deploy configuration files.
  System Action: Deploy command execution fails.
  User Action: Ensure that a valid virtual server configuration file or template file for deployment exists and is defined properly. Refer the configuration files section of the IBM Documentation.

- **HVS-DEPL002: Error parsing the %s config file. Provide valid details in the config file**
  Explanation: The deploy operation failed as there is an error while parsing the deploy configuration files.
  System Action: Deploy command execution fails.
  User Action: Ensure that a valid virtual server configuration file or template file for deployment is defined properly. Refer the configuration files section of the IBM Documentation.

- **HVS-DEPL003: Error in getting the size for the quotagroup. Provide the size in the correct format**
  Explanation: The deploy operation failed as there was an error while parsing the size details for a given quotagroup in the template file.
  System Action: Deploy command execution fails.
  User Action: Ensure that a valid quotagroup size is defined in quotagrouptemplates and retry the command. Refer the configuration files section of the IBM Documentation.

- **HVS-DEPL004: Mountpoint %s for the volume in the Virtual Server yaml file should start with /**
  Explanation: The deploy operation failed as there is an error while validating the mount point for the volume in vitual server yaml file.
  System Action: Deploy command execution fails.
  User Action: Ensure that mount point for the volume in the virtual server yaml file starts with '/' and retry the command.

- **HVS-DEPL005: Host %s is not accessible. Verify the host configuration**
  Explanation: The deploy operation failed as there was an error when trying to access the configured Secure Service Container LPAR.
  System Action: Deploy command execution fails.
  User Action: Ensure that the valid Secure Service Container LPAR host is configured and retry the command. If the problem still persists, obtain appliance logs, LMS dump and contact IBM Support.

- **HVS-DEPL006: Repository registration file %s does not exist**
  Explanation: The deploy operation failed as repository registration file does not exist.
  System Action: Deploy command execution fails.
  User Action: Check if the repository registration file path defined in the virtual server configuration file path exists. If not, configure the file path correctly and retry the command.

- **HVS-DEPL007: Error reading the repository registration file %s . Provide a valid registration file**
  Explanation: The deploy operation failed as there was an error in reading the repository registration file.
  System Action: Deploy command execution fails.
  User Action: Ensure that a valid repository registration file is defined in the virtual server configuration file and retry the command. Refer the repository commands section of the IBM Documentation.

- **HVS-DEPL008: Only control crypto devices can have more than one crypto matrix. Set crypto control true to have more than one crypto matrix defined**
  Explanation: The deploy operation failed because of invalid crypto card configuration.
  System Action: Deploy command execution fails.
  User Action: Crypto control boolean flag should be set to true in the virtual server configuration file if more than one crypto matrix need to be defined. Ensure the configuration is correct and retry the command.

- **HVS-DEPL009: Mountpoint for the volume %s in Virtual Server yaml file is empty or not defined**
  Explanation: The deploy operation failed as mountpoint for the volume is empty or not defined in the virtual server config file.
  System Action: Deploy command execution fails.
  User Action: Ensure the mountpoint is defined correctly in the virtual server config file. Ensure the configuration is correct and retry the command. Refer the configuration files section of the IBM Documentation.

- **HVS-DEPL010: Image file %s to be loaded to Secure Service Container partition does not exist**
  Explanation: The deploy operation failed as image to be loaded to Secure Service Container partition does not exist.
  System Action: Deploy command execution fails.
  User Action: Check if image file path defined in the virtual server configuration file exists. Ensure the configuration is correct and retry the command.

- **HVS-DEPL011: Quotagroup with reference %s to be created is not defined in Virtual Server template file**
  Explanation: The deploy operation failed as the quotagroup definition referred is not defined in the quotagrouptemplates of the virtual server template file.
  System Action: Deploy command execution fails.
  User Action: Ensure that the reference is defined correctly in the virtual server config file or defined correctly in the virtual server template file. Refer the configuration files section of the IBM Documentation.

- **HVS-DEPL012: Network name to be attached to virtual server is empty. Define network with non empty network name**
  Explanation: The deploy operation failed as the network name definition is not correct in the networktemplates section of virtual server template file.
  System Action: Deploy command execution fails.
  User Action: Ensure that a non empty network name is provided and retry the command. Refer the configuration files section of the IBM Documentation.

- **HVS-DEPL013: Network driver %s attached to virtual server is invalid. Define the driver as either macvlan or bridge**
  Explanation: The deploy operation failed as the network driver definition is not correct in the networktemplates section of the virtual server template file.
  System Action: Deploy command execution fails.
  User Action: Ensure that a valid network driver is specified (either macvlan or bridge) and retry the command. Refer the configuration files section of the IBM Documentation.

- **HVS-DEPL014: Gateway %s assigned to Virtual Server network is invalid. Provide valid IPv4 address**
  Explanation: The deploy operation failed as the Gateway definition is not correct in the networktemplates section of the virtual server template file.
  System Action: Deploy command execution fails.
  User Action: Ensure that gateway is given in standard IPv4 format and retry the command.

- **HVS-DEPL015: Subnet %s assigned to Virtual Server network is invalid. Provide valid subnet**
  Explanation: The deploy operation failed as the Subnet definition is not correct in the networktemplates section of the virtual server template file.
  System Action: Deploy command execution fails.
  User Action: Ensure that the subnet is specified in the IPv4/Prefix format and retry the command.

- **HVS-DEPL016: The parent device assigned to Virtual Server network is empty or not defined. Provide valid network parent device**
  Explanation: The deploy operation failed as the parent definition is not correct in networktemplates section of virtual server template file.
  System Action: Deploy command execution fails.
  User Action: Ensure that a valid network parent is specified and is non empty. Ensure the configuration is correct and retry the command.

- **HVS-DEPL017: Size for the volume %s in the Virtual Server yaml file is empty or not defined. Provide valid size for the volume**
  Explanation: The deploy operation failed as the size is not defined for the volume with non passthrough quotagroup in virtual server config file.
  System Action: Deploy command execution fails.
  User Action: Ensure that the size is given for the volume to be attached to virtual server and retry the command. Refer the configuration files section of the IBM Documentation.

- **HVS-DEPL018: Resource definition %s assigned to Virtual Server is not defined in virtual server template file. Define required resource definition in template file**
  Explanation: The deploy operation failed as the resource definition defined in virtual server config file is not present in resourcedefinitiontemplates section of the virtual server template file.
  System Action: Deploy command execution fails.
  User Action: Ensure that resource reference is correct in virtual server config file or defined correctly in virtual server template file. Refer the configuration files section of the IBM Documentation.

- **HVS-DEPL019: Error in getting the size for the volume %s in the virtual server yaml file. Provide a valid size for the volume**
  Explanation: The deploy operation failed as there was an error while parsing size details for given volume in virtual server config file.
  System Action: Deploy command execution fails.
  User Action: Ensure that a valid size is defined correctly in the volume section of virtual server config file and retry the command. Refer the configuration files section of the IBM Documentation.

- **HVS-DEPL020: Filesystem for the volume %s in the virtual server yaml file is empty or invalid. Provide a valid filesystem for the volume**
  Explanation: The deploy operation failed as there was error in configuration of filesystem in the virtual server config file.
  System Action: Deploy command execution fails.
  User Action: Ensure that a valid filesystem is specified which can be either ext4, btrfs, or xfs, and retry the command. Refer the configuration files section of the IBM Documentation.

- **HVS-DEPL021: Host Port value is invalid. Provide a port value in the range 1-65535**
  Explanation: The deploy operation failed as the host port configuration is not correct in the virtual server config file.
  System Action: Deploy command execution fails.
  User Action: Ensure that a valid port configuration is provided in the range of 1 to 65535, and retry the command.

- **HVS-DEPL022: Protocol value is invalid. Provide the supported protocol "tcp" or "udp"**
  Explanation: The deploy operation failed as the protocol value configuration is not correct in the virtual server config file.
  System Action: Deploy command execution fails.
  User Action: Ensure that a valid protocol is provided which should be either "tcp" or "udp", and retry the command.

- **HVS-DEPL023: Container Port value is invalid. Provide the port value in the range 1-65535**
  Explanation: The deploy operation failed as port configuration is not correct in virtual server config file for container port.
  System Action: Deploy command execution fails.
  User Action: Ensure that a valid port configuration is provided in the range 1 to 65535 and retry the command.

- **HVS-DEPL024: IP Address %s assigned to the virtual server network is invalid. Provide a valid IPv4 address**
  Explanation: The deploy operation failed as the IP address definition is not correct in the virtual server config file.
  System Action: Deploy command execution fails.
  User Action: Ensure that the IP address is given in the standard IPv4 format and retry the command.

- **HVS-DEPL025: Network %s assigned to the virtual server is not defined in the virtual server template file. Define the network in the template file**
  Explanation: The deploy operation failed as the network definition defined in the virtual server config file is not present in the networktemplates section of the virtual server template file.
  System Action: Deploy command execution fails.
  User Action: Ensure that the network reference is defined correctly in the virtual server config file or defined correctly in the virtual server template file. Refer the configuration files section of the IBM Documentation.

- **HVS-DEPL026: Quotagroup with given name %s does not exist**
  Explanation: Deploy operation failed as the specified quotagroup does not exist.
  System Action: Deploy command execution fails.
  User Action: Ensure that proper quotagroup details are provided. Refer the configuration quotagroup commands section of the IBM Documentation.

- **HVS-DEPL027: Virtual server name(s) provided in the include list are incorrect**
  Explanation: Deploy operation failed as the virtual server details in the list are incorrect.
  System Action: Deploy command execution fails.
  User Action: Ensure proper virtual server names are provided in the include list. Refer the deploy command section of the IBM Documentation.

- **HVS-DEPL028: Virtual server name(s) provided in the exclude list are incorrect**
  Explanation: Deploy operation failed as the virtual server details in the list are incorrect.
  System Action: Deploy command execution fails.
  User Action: Ensure that proper virtual server names are provided in the exclude list. Refer the deploy command section of the IBM Documentation.

- **HVS-DEPL029: All virtual servers are excluded**
  Explanation: Deploy operation failed as all the virtual servers listed in the yaml file are excluded.
  System Action: Deploy command execution fails.
  User action: Ensure that proper values are provided in the list. Refer the deploy command section of the IBM Documentation.

- **HVS-DEPL030: Virtual server with the given name %s does not exist**
  Explanation: Deploy operation failed as the virtual server with the name does not exist.
  System Action: Deploy command execution fails.
  User Action: Ensure that proper virtual server details are given. Refer the deploy command section of the IBM Documentation.

- **HVS-DEPL031: Error in updating the quotagroup as it is a passthrough quotagroup**
  Explanation: Deploy operation failed as the passthrough quotagroup cannot be updated.
  System Action: Deploy command execution fails.
  User Action: Ensure initial value of passthrough quotagroup is provided. Use the `hpvs quotgaroup show` command. Refer the quotagroup command section of the IBM Documentation.

- **HVS-DEPL032: Quotagroup %s change in passthrough parameter of quotagroup is not allowed**
  Explanation: Deploy operation failed to update the passthrough parameter of the quotagroup.
  System Action: Deploy command execution fails.
  User Action: Ensure initial value of passthrough parameter value is provided. Use the `hpvs quotgaroup show` command. Refer the quotagroup command section of the IBM Documentation.

- **HVS-DEPL033: Repository ID %s does not exist. Provide valid details and try again**
  Explanation: Deploy operation failed to update repository.
  System Action: Deploy command execution fails.
  User Action: Ensure the repository exists. Refer the deploy command section of the IBM Documentation.

- **HVS-DEPL034: Error while reading the %s ENV file. Error : %s. Provide a valid ENV file**
  Explanation: Deploy operation failed to read ENV variable files parsed.
  System Action: Deploy command execution fails.
  User Action: Ensure ENV files exists or a proper filepath is provided. Refer the deploy command section of the IBM Documentation.

- **HVS-DEPL035: Parameters reporegfile and imagefile cannot be used together and are mutually exclusive**
  Explanation: Deploy operation failed as both the reporegfile and imagefile are provided.
  System Action: Deploy command execution fails.
  User Action: Ensure that use only one parameter reporegfile, or imagefile. Refer the deploy command section of the IBM Documentation.

- **HVS-DEPL036: Parameter imagetag cannot be empty in the yaml file. Please provide valid value and retry**
  Explanation: Deploy operation failed as imagetag is not provided in the yaml file.
  System Action: Deploy command execution fails.
  User Action: Ensure that you provide imagetag in the yaml file and retry. Refer the deploy command section of the IBM Documentation.

Host command messages

This section lists the messages you might encounter while using the image commands.

HVS-HOIN00x: Messages for host initialization

This section lists the messages you might encounter while running the host initialization command.

- **HVS-HOIN001: Initialize host failed. Error while creating directory for host file. Details: %s. Refer the product documentation on host configuration details**
  Explanation: Host initialization failed as there is an error while creating host directory.
  System Action: Host init command execution fails.
  User Action: Retry the command, if the problem persists, obtain an appliance and LMS dump, and contact IBM Support.

HVS-HOAD00x: Messages for host add command

This section lists the messages you might encounter while running the host add command.

- **HVS-HOAD001: Add host failed. User already exists for the given host. Refer the product documentation on host configuration details**
  Explanation: Host add failed because user is already present for the given host. Refer to the product

documentation for more details.
System Action: Add host command execution fails.
User Action: Provide a different user and retry the command.

- **HVS-HOAD002: Add host failed. There is an internal error in writing to host file. Refer the product documentation**
  Explanation: An internal processing error occurred while writing to host file.
  System Action: Add host command execution fails.
  User Action: Retry the command. If the problem persists obtain an appliance and LMS dump, and contact IBM Support.

- **HVS-HOAD003: Add host failed. There is an internal error in creating a new file. Refer the product documentation**
  Explanation: An internal processing error occurred while creating host file.
  System Action: Add host command execution fails.
  User Action: Retry the command. If the problem persists obtain an appliance and LMS dump, and contact IBM Support.

- **HVS-HOAD004: Add host failed. There is an internal error in writing to host file. Refer the product documentation**
  Explanation: An internal processing error occurred while writing to host file.
  System Action: Add host command execution fails.
  User Action: Retry the command. If the problem persists obtain an appliance and LMS dump, and contact IBM Support.

- **HVS-HOAD005: Add host failed. One or more input parameters are invalid. Refer the product documentation**
  Explanation: Adding host failed because one or more input parameters are invalid.
  System Action: Add host command execution fails.
  User Action: Ensure that valid parameters are provided. Refer to the product documentation for more details and retry the command.

- **HVS-HOAD006: Set default host failed. Refer the product documentation**
  Explanation: An internal processing error occurred while setting the default host.
  System Action: Add host command execution fails.
  User Action: Retry the command. If the problem persists obtain an appliance and LMS dump, and contact IBM Support.

HVS-HOUD001x: Messages for host update command

This section lists the messages you might encounter while running the host update command.

- **HVS-HOUD001: Update host failed. Invalid host details provided. Provide valid host details**
  Explanation: Invalid host details are provided causing a failure to update the host.
  System Action: Update host command execution fails.
  User Action: Ensure that valid host details are provided and retry the command. Refer the host update commands section of the IBM Documentation.

- **HVS-HOUD002: Update host failed**
  Explanation: The update operation failed as there is an error while updating the host.
  System Action: Update host command execution fails.
  User Action: Ensure that valid host details are provided and retry the command. Refer the host update commands section of the IBM Documentation.

- **HVS-HOUD003: Update host failed**
  Explanation: Update operation failed as there is an error while updating the host.
  System Action: Update host command execution fails.

User Action: Ensure that valid host details are provided and retry the command. Refer the host update commands section of the IBM Documentation.

HVS-HODE00x: Messages for host delete command

This section lists the messages you might encounter while running the host update command.

- **HVS-HODE001: Delete host failed. The host provided does not exist. Provide valid host details**
  Explanation: Invalid host details are provided causing the failure to delete the host.
  System Action: Delete host command execution fails.
  User Action: Ensure that valid host detail is provided and retry the command.

- **HVS-HODE002: Delete host failed. Unable to update the host file. Refer the product documentation**
  Explanation: An internal processing error occurred while updating the host details.
  System Action: Delete host command execution fails.
  User Action: Retry the command. If the problem persists obtain an appliance and LMS dump, and contact IBM Support.

- **HVS-HODE003: Delete host failed to set default host**
  Explanation: An internal processing error occurred while setting the default host.
  System Action: Delete host command execution fails.
  User Action: Retry the command. If the problem persists obtain an appliance and LMS dump, and contact IBM Support.

- **HVS-HODE004: Delete host failed. There is an internal error. Refer the product documentation**
  Explanation: An internal processing error occurred while setting the default host.
  System Action: Delete host command execution fails.
  User Action: Retry the command. If the problem persists obtain an appliance and LMS dump, and contact IBM Support.

- **HVS-HODE005: Delete host failed. The host provided does not exist. Ensure to provide valid host details and retry command**
  Explanation: Host delete operation failed because the host provided does not exist.
  System Action: Delete host command execution fails.
  User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

- **HVS-HODE006: Delete host failed to update host file**
  Explanation: Host delete operation failed because the repository ID is invalid.
  System Action: Delete host command execution fails.
  User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

- **HVS-HODE007: Set default host failed**
  Explanation: Host delete operation failed because the repository ID is invalid.
  System Action: Delete host command execution fails.
  User Action: Ensure that valid host details are provided and try again.

HVS-HOST00x: Messages for host set command

This section lists the messages you might encounter while running the host set command.

- **HVS-HOST001: Set host failed. No user present for this host. Ensure that proper host details are added**
  Explanation: Invalid host details are provided causing the failure to set the host.
  System Action: Set host command execution fails.
  User Action: Ensure that valid host details are provided and retry the command. Refer the host delete commands section of the IBM Documentation.

- **HVS-HOST002: Set host failed. There is an internal error updating host details. Refer the product documentation**
  Explanation: An internal processing error occurred while updating the host details.
  System Action: Set host command execution fails.
  User Action: Retry the command. If the problem persists obtain an appliance and LMS dump, and contact IBM Support.

- **HVS-HOST003: Set host failed. There is an internal error in writing to file %s. Refer the product documentation**
  Explanation: An internal processing error occurred while writing in the host file.
  System Action: Set host command execution fails.
  User Action: Retry the command. If the problem persists obtain an appliance and LMS dump, and contact IBM Support.

- **HVS-HOST004: Set host failed. No user present for this host. Ensure that proper host details are added**
  Explanation: Host set operation failed because no user was present for this host.
  System Action: Set host command execution fails.
  **User Action: Ensure that valid host details are provided and try again.

- **HVS-HOST005: Set host failed. There is an internal error updating host details. Refer to the product documentation**
  Explanation: Host set operation failed because there was an internal error while updating the host details.
  System Action: Set host command execution fails.
  User Action: Retry the command, if the problem persists obtain the appliance dump, LMS logs and Contact IBM Support.

- **HVS-HOST006: Set host failed. There is an internal error updating host details. Refer to the product documentation**
  Explanation: Host set operation failed because there was an internal error while updating the host details.
  System Action: Set host command execution fails.
  User Action: Retry the command, if the problem persists obtain the appliance dump, LMS logs and Contact IBM Support.

HVS-HOSH00x: Messages for host show command

This section lists the messages you might encounter while running the host show command.

- **HVS-HOSH001: Show host failed. No user present for this host. Ensure that proper host details are added**
  Explanation: Invalid host details are provided causing the failure to show host.
  System Action: Set host command execution fails.
  User Action: Ensure that valid host details are provided and retry the command. Refer the host commands section of the IBM Documentation.

HVS-HOUS00x: Messages for host unset command

This section lists the messages you might encounter while running the host unset command.

- **HVS-HOUS001: Unset host failed. The requested host %s is not set**
  Explanation: Invalid host details are provided causing the failure to unset host.
  System Action: Set host command execution fails.
  User Action: Ensure that valid host details are provided and retry the command. Refer the host commands section of the IBM Documentation.

- **HVS-HOUS002: Unset host failed to set default host. Details: %s. Refer to product documentation for resolution**
  Explanation: Invalid host details are provided causing the failure to unset host.
  System Action: Set host command execution fails.
  User Action: Ensure that valid host details are provided and retry the command. Refer the host commands section of the IBM Documentation.

- **HVS-HOUS003: Unset host failed. No user present for this host. Ensure that proper host details are added**
  Explanation: Invalid host details are provided causing the failure to unset host.
  System Action: Set host command execution fails.
  User Action: Ensure that valid host details are provided and retry the command. Refer the host commands section of the IBM Documentation.

HVS-HOLI00x: Messages for host list command

This section lists the messages you might encounter while running the host list command.

- **HVS-HOLI001 List host failed. There is an internal error in reading file**
  Explanation: An internal processing error occurred while reading the host file.
  System Action: List host command execution fails.
  User Action: Retry the command. If the problem persists obtain an appliance and LMS dump, and contact IBM Support.

- **HVS-HOLI002: List host failed. There is an internal error in reading file**
  Explanation: Host list operation failed because there was an internal error in reading file.
  System Action: List host command execution fails.
  User Action: Retry the command, if the problem persists obtain the appliance dump, LMS logs and Contact IBM Support.

Image command messages

This section lists the messages you might encounter while using the image commands.

HVS-IMSW00x: Messages for image show command

This section lists the messages you might encounter while running the image show command.

- **HVS-IMSW001: The command to show the image failed due to an internal server error**
  Explanation: An internal server error occurred, resulting in failure of image show request.
  System Action: Show image command execution fails.
  User Action: Retry the command if the problem persists, obtain an appliance dump and LMS and Contact IBM Support.

- **HVS-IMSW002: Show Image command failed. Invalid Image hash ID**
  Explanation: The show image command failed as an invalid image hash ID was provided.
  System Action: Show image command execution fails.
  User Action: Ensure you provide a valid Image hash ID and retry the command.

- **HVS-IMSW003: Show image command failed due to server issues. Refer logs**
  Explanation: Image show operation failed because there is an internal server error.
  System Action: Show image command execution fails.
  User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

- **HVS-IMSW004: Show image command failed due to unavailability of image hash**
  Explanation: Image show operation failed because image hash was not found.
  System Action: Show image command execution fails.
  User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

HVS-IMPL00x: Messages for image pull command

This section lists the messages you might encounter while running the image pull command.

- **HVS-IMPL001: The command to pull the image failed due to an internal server error**
Explanation: An internal server error, resulting in failure of image pull request.
System Action: Pull Image command execution fails.
User Action: Retry the command if the problem persists, obtain an appliance dump and LMS and Contact IBM Support.

- **HVS-IMPL002: Pull image command failed due to error in json format. Verify if repoID and imgTag are defined correctly**
Explanation: An invalid image repoID and/or imgTag is provided causing the failure to pull the image.
System Action: Pull image command execution fails.
User Action: Ensure that a valid image repoID and imgTag are provided and retry the command. Refer the image commands section of the IBM Documentation.

- **HVS-IMPL003: Pull image command failed due to server issues. Refer logs**
Explanation: Image pull operation failed because there is an internal server error.
System Action: Pull image command execution fails.
User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

- **HVS-IMPL004: Pull image command failed due to error in json format. Refer documentation**
Explanation: Image pull operation failed because there is an error in json format.
System Action: Pull image command execution fails.
User Action: Ensure that valid image details are provided and try again.

- **HVS-IMPL005: Image pull failed. Could not find trust data for image %s. Provide valid image details and try again**
Explanation: Image pull operation failed because trust data for image was not found.
System Action: Pull image command execution fails.
User Action: Ensure that valid image details are provided and try again.

HVS-IMLD00x: Messages for image load command

This section lists the messages you might encounter while running the image load command.

- **HVS-IMLD001: The command to load the image failed due to an internal server error**
Explanation: An internal server error occurred, resulting in a request to load the image failure.
System Action: Load image command execution fails.
User action: Retry the command. If the problem persists, obtain an appliance dump and LMS, and contact IBM Support.

- **HVS-IMLD002: The command to load the image failed**
Explanation: Load image failed as there is an error while loading the given image.
System Action: Load image command execution fails.
User action: Ensure that a valid image is provided and retry the command. Refer the image commands section of the IBM Documentation.

- **HVS-IMLD003: Failed to load image for repository %s. Provide valid gpg keys and try again**
Explanation: Load image failed because the gpg keys are not valid.
System Action: Load image command execution fails.
User Action: Ensure that a valid gpg key is provided and retry the command. Refer the image commands section of the IBM Documentation.

- **HVS-IMLD004: Failed to verify the signature for the repository %s when loading the docker image bundle**
Explanation: Load image failed because signature verification of repository failed when loading the docker image bundle.
System Action: Load image command execution fails.
User Action: Ensure that the docker image bundle contains proper signatures. If the problem persists, obtain an appliance and LMS dump, and contact IBM Support.

HVS-IMDE00x: Messages for image delete command

This section lists the messages you might encounter while running the image delete command.

- **HVS-IMDE001: The command to delete the image failed due to an internal server error**
  Explanation: An internal server error, resulting in failure of delete image request.
  System Action: Delete image command execution fails.
  User Action: Retry the command if the problem persists, obtain an appliance dump and LMS and Contact IBM Support.

- **HVS-IMDE002: Delete image command failed due to unavailability of image hash**
  Explanation: Delete image command failed, as an invalid image hash Id is provided.
  System Action: Delete image command execution fails.
  User Action: Provide a valid image hash Id and retry the command. Check the image commands section of the of the IBM Documentation. Retry the command if the problem persists, obtain an appliance dump and LMS and Contact IBM Support.

- **HVS-IMDE003: Delete image command failed due to server issues. Refer logs**
  Explanation: Image delete operation failed because there is an internal server error.
  System Action: Delete image command execution fails.
  User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

- **HVS-IMDE004: Delete Image command failed due to unavailability of image hash**
  Explanation: Image delete operation failed because image hash was not found.
  System Action: Delete image command execution fails.
  User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

- **HVS-IMDE005: Image delete failed. The image %s cannot be deleted because the container(s) %s exist(s) and depend on it**
  Explanation: Image delete operation failed because the container(s) %s exist(s) and depend on it.
  System Action: Delete image command execution fails.
  User Action: Ensure that valid image details are provided and try again.

HVS-IMLI00x: Messages for image list command

This section lists the messages you might encounter while running the image delete command.

- **HVS-IMLI001: The command to list the image failed due to an internal server error**
  Explanation: An internal server error occurred, resulting in a request to list an image failure.
  System Action: List image command execution fails.
  User action: Retry the command. If the problem persists, obtain an appliance dump and LMS, and contact IBM Support.

HVS-IMYY00x: Messages for some image commands

This section lists the messages you might encounter while running some image commands.

- **HVS-IMYY001: Image %s was not found. Provide valid image details and try again**
  Explanation: Image operation failed because image details were not valid.
  System Action: Image command execution fails.
  User Action: Ensure that valid image details are provided and try again.

- **HVS-IMYY002: Could not create tar file. Obtain an appliance dump and contact IBM Support**
  Explanation: Image operation failed because there was an error to create tar file.
  System Action: Image command execution fails.
  User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

- **HVS-IMYY003: The images bundle contains invalid filetypes. Provide valid details and try again**
  Explanation: Image operation failed because the filetypes provided in image bundle are invalid.
  System Action: Image command execution fails.
  User Action: Ensure that valid image details are provided and try again.
- **HVS-IMYY004: The images bundle does not contain a .sig file for repository %s. Provide valid details and try again**
  Explanation: Image operation failed because the images bundle does not contain a .sig file for repository.
  System Action: Image command execution fails.
  User Action: Ensure that valid image details are provided and try again.
- **HVS-IMYY005: The images bundle does not contain a .enc file for repository %s. Provide valid details and try again**
  Explanation: Image operation failed because the images bundle does not contain a .enc file for repository.
  System Action: Image command execution fails.
  User Action: Ensure that valid image details are provided and try again.
- **HVS-IMYY006: Unable to open images bundle. Obtain an appliance dump and contact IBM Support**
  Explanation: Image operation failed because there was an error to open images bundle.
  System Action: Image command execution fails.
  User Action: Ensure that valid image details are provided and try again.
- **HVS-IMYY007: Unable to check if image %s is in use. Obtain an appliance dump and contact IBM Support**
  Explanation: Image operation failed because there was an error to check if image is in use.
  System Action: Image command execution fails.
  User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.
- **HVS-IMYY008: Unable to determine the size of image %s. Obtain an appliance dump and contact IBM Support**
  Explanation: Image operation failed because there was an error to determine the size of image.
  System Action: Image command execution fails.
  User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.
- **HVS-IMYY009: Unable to allocate storage of size for image file %s. size:%s offset:%s. Obtain an appliance dump and contact IBM Support**
  Explanation: Image operation failed because there was an error to allocate storage of size for image file.
  System Action: Image command execution fails.
  User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.
- **HVS-IMYY010: Unable to truncate image file %s. Obtain an appliance dump and contact IBM Support**
  Explanation Image operation failed because there was an error to truncate image file.
  System Action: Image command execution fails.
  User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.
- **HVS-IMYY011: Unable to create filesystem %s for image %s. Obtain an appliance dump and contact IBM Support**
  Explanation: Image operation failed because there was an error to create filesystem for image.
  System Action: Image command execution fails.
  User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.
- **HVS-IMYY012: Unable to resize filesystem %s for image %s. Obtain an appliance dump and contact IBM Support**

Explanation: Image operation failed because there was an error to resize filesystem for image.
System Action: Image command execution fails.
User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

Network command messages

This section lists the messages you might encounter while using the network command.

HVS-NWCR00x: Messages for network create command

This section lists the messages you might encounter while running the network create command.

- **HVS-NWCR001: The command to create network failed due to an internal server error**
  Explanation: An internal server error occurred resulting network list command failure.
  System Action: Network create command execution fails.
  User Action: Retry the command. If the problem persists, obtain an appliance and LMS dump, and contact IBM Support.

- **HVS-NWCR002: Create network failed. Error in parsing the network details. Provide valid network details**
  Explanation: An internal server error occurred resulting network list command failure.
  System Action: Network create command execution fails.
  User Action: Retry the command by providing all valid values. Refer the network delete commands section of the IBM Documentation.

- **HVS-NWCR003: Network create failed because invalid network driver name was provided. Provide valid network driver details and try again**
  Explanation: Network create operation failed because invalid network driver name was provided.
  System Action: Create network command execution fails.
  User Action: Ensure that valid network driver details are provided and try again.

- **HVS-NWCR004: Network create failed because requested gateway %s is out of range. Provide valid address and try again**
  Explanation: Network create operation failed because requested gateway is out of range.
  System Action: Create network command execution fails.
  User Action: Ensure that valid network details are provided and try again.

- **HVS-NWCR005: Network create failed because subnet should be specified with gateway details**
  Explanation: Network create operation failed because subnet was not specified with gateway details.
  System Action: Create network command execution fails.
  User Action: Ensure that valid network details are provided and try again.

- **HVS-NWCR006: Network create failed because invalid IPNetwork %s was provided. Provide valid IPNetwork and try again**
  Explanation: Network create operation failed because invalid IPNetwork was provided.
  System Action: Create network command execution fails.
  User Action: Ensure that valid network details are provided and try again.

- **HVS-NWCR007: Network create failed. Error occurred creating firewall rules for network %s**
  Explanation: Network create operation failed because error occurred while creating firewall rules for network.
  System Action: Create network command execution fails.
  User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

- **HVS-NWCR008: Network create failed. Found Error: %s. Provide valid network details and try again**
  Explanation: Network create operation failed because there was error %s to create network.

System Action: Create network command execution fails.
User Action: Ensure that valid network details are provided and try again.

HVS-NWUD00x: Messages for network update command

This section lists the messages you might encounter while running the network update command.

- **HVS-NWUD001: Network update failed because subnet %s conflicts with existing subnet %s. Provide valid subnet details and try again**
  Explanation: Network update operation failed because subnet conflicts with existing subnet.
  System Action: Update network command execution fails.
  User Action: Ensure that valid subnet details are provided and try again.

- **HVS-NWUD002: Network subnet value is invalid. Provide a valid network subnet and try again**
  Explanation: Network operation failed due to invalid subnet address provided.
  System Action: Network update command execution fails.
  User Action: Retry the command. If the problem persists, refer the network update commands section of the IBM Documentation.

- **HVS-NWUD003: Update failed for the network with given name '%s'. Provide a valid internal network name and try again**
  Explanation: Network operation failed since the current feature supports only docker default/bridge network update.
  System Action: Network update command execution fails.
  User Action: Retry the command. If the problem persists, refer the network update commands section of the IBM Documentation.

HVS-NWDE00x: Messages for network delete command

This section lists the messages you might encounter while running the network delete command.

- **HVS-NWDE001: The command to delete network failed due to an internal server error**
  Explanation: An internal server error occurred resulting network list command failure.
  System Action: Network show command execution fails.
  User Action: Obtain an appliance and LMS dump Contact IBM Support.

- **HVS-NWDE002: Delete network failed. Provide valid network name**
  Explanation: An internal server error occurred resulting network delete command failure.
  System Action: Network delete command execution fails.
  User Action: Retry the command by providing a valid name. Refer the network delete commands section of the IBM Documentation.

- **HVS-NWDE003: Delete Network %s failed as there are containers attached**
  Explanation: An internal server error occurred resulting network delete command failure.
  System Action: Network delete command execution fails.
  User Action: Retry the command after you delete the attached resources. Refer the network delete commands section of the IBM Documentation.

HVS-NWLI00x: Messages for network list command

This section lists the messages you might encounter while running the network list command.

- **HVS-NWLI001: The command to list network failed due to an internal server error**
  Explanation: An internal server error occurred resulting network list command failure.
  System Action: Network list command execution fails.
  User Action: Retry the command. If the problem persists, obtain an appliance and LMS dump, and contact IBM Support.

HVS-NWSW00x: Messages for network show command

This section lists the messages you might encounter while running the network show command.

- **HVS-NWSW001: The command to show network failed due to an internal server error**
  Explanation: An internal server error occurred resulting network show command failure.
  System Action: Network show command execution fails.
  User Action: Retry the command. If the problem persists, obtain an appliance and LMS dump and contact IBM Support.

- **HVS-NWSW002: Show network failed. Error in parsing the network details**
  Explanation: An internal server error occurred while processing the network show command output.
  System Action: Network show command execution fails.
  User Action: Retry the command. If the problem persists, obtain an appliance and LMS dump and contact IBM Support.

HVS-NWYY00x: Messages for some network commands

This section lists the messages you might encounter while running some network commands.

- **HVS-NWYY001: Network %s was not found. Provide valid network details and try again**
  Explanation: Network operation failed because network was not found.
  System Action: Network command execution fails.
  User Action: Ensure that valid network details are provided and try again.

- **HVS-NWYY002: Failed to get defined interfaces. Obtain an appliance dump and contact IBM Support**
  Explanation: Network operation failed because there was error to get defined interfaces.
  System Action: Network command execution fails.
  User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

Quotagroups command messages

This section lists the messages you might encounter while using the quotagroup commands.

HVS-QGCR00x: Messages for create quotagroup command

This section lists the messages you might encounter while running quotagroup create command.

- **HVS-QGCR001: The command to create the quotaGroup failed due to an internal server error**
  Explanation: An internal server error occurred, resulted in a failure request to create a quotagroup.
  System Action: Create quotagroup command execution fails.
  User Action: Retry the command. If problem persists collect the logs, dumps, and contact IBM Support.

- **HVS-QGCR002: Create qutotagroup failed. Error in getting the size and unit of quotagroup. Provide valid size and unit**
  Explanation: An invalid quotagroup size or/and unit is provided causing the failure to create a quotagroup.
  System Action: Create quotagroup command execution fails.
  User Action: Ensure that a valid quotagroup name, and size details are provided. Refer the quotagroup commands section of the IBM Documentation.

- **HVS-QGCR003: Create quotagroup failed. Error in parsing the quotagroup details. Provide valid quotagroup details**
  Explanation: An error occurred while processing the input parameters, resulting in the a request to create a quotagroup failure.
  System Action: Create quotagroup command execution fails.
  User Action: Ensure you specify the supported values and retry the commands. Refer the quotagroup commands section of the IBM Documentation.

- **HVS-QGCR004: Quotagroup create failed. The quotagroup %s already exists. Provide different quotagroup details and try again**
  Explanation: Quotagroup create operation failed because the quotagroup %s already exists.
  System Action: Create quotagroup command execution fails.
  User Action: Provide a different quotagroup name and try again.

- **HVS-QGCR005: Quotagroup create failed because unsupported filesystem %s for %s was provided. Provide valid filesystem details and try again**
  Explanation: Quotagroup create failed because unsupported filesystem details are provided.
  System Action: Create quotagroup command execution fails.
  User Action: Ensure that supported filesystem details are provided and try again.

- **HVS-QGCR006: Quotagroup create failed. Insufficient space to create quotagroup %s**
  Explanation: Quotagroup create failed because there is insufficient space.
  System Action: Create quotagroup command execution fails.
  User Action: Retry the command with less size of quotagroup. If the problem persists obtain the appliance dump, LMS logs, and Contact IBM Support.

- **HVS-QGCR007: Quotagroup create failed. Exception occurred while creating quotagroup %s**
  Explanation: Quotagroup create failed because there is was an exception while creating quotagroup.
  System Action: Create quotagroup command execution fails.
  User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and Contact IBM Support.

- **HVS-QGCR008 Filesystem size (%s Bytes) is lower than the minimum size of %s Bytes. Provide valid details and try again**
  Explanation: Quotagroup create failed because the filesystem details provided are lower than the minimum size.
  System Action: Create quotagroup command execution fails.
  User Action: Ensure that valid filesystem details are provided and try again.

HVS-QGUD00x: Messages for update quotagroup command

This section lists the messages you might encounter while running quotagroup update command.

- **HVS-QGUD001: The command to update the quotaGroup failed due to an internal server error**
  Explanation: An internal server error occurred, resulting in a request to update a given quotagroup failure.
  System Action: Update quotagroup command execution fails.
  User Action: Retry the command, and if the problem persists, obtain an appliance dump and LMS and contact IBM Support.

- **HVS-QGUD002: Update quotagroup failed. Error in getting the size and unit of quotagroup. Provide valid size and unit**
  Explanation: An invalid quotagroup size or/and unit is provided causing the failure to update a given quotagroup.
  System Action: Update quotagroup command execution fails.
  User Action: Ensure that a valid quotagroup size and unit details are provided. Refer the quotagroup command section of the IBM Documentation.

- **HVS-QGUD003: Failed to update quotagroup. Error in parsing the quotagroup details**
  Explanation: An error occurred while processing the input parameters, resulting in a request to update a quotagroup failure.
  System Action: Update quotagroup command execution fails.
  User Action: Obtain an appliance and LMS dump contact IBM Support.

- **HVS-QGUD004: Quotagroup update failed. Not enough space available on pool %s to extend logical volume %s**
  Explanation: Quotagroup update failed because there was is not enough space available on pool to extend

logical volume.
System Action: Update quotagroup command execution fails.
User Action: Retry the command with less size of quotagroup. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

- **HVS-QGUD005: Quotagroup update failed. Logical volume %s does not exist on pool %s**
  Explanation: Quotagroup update failed because logical volume does not exist on pool.
  System Action: Update quotagroup command execution fails.
  User Action: Ensure that valid quotagroup details are provided and try again.

- **HVS-QGUD006: Quotagroup update failed. The Quotagroup %s does not exist**
  Explanation: Quotagroup update failed because the quotagroup does not exist.
  System Action: Update quotagroup command execution fails.
  User Action: Ensure that valid quotagroup details are provided and try again.

- **HVS-QGUD007: Quotagroup update failed because quotagroup %s does not exist. Provide valid quotagroup details and try again**
  Explanation: Quotagroup update failed because the quotagroup does not exist.
  System Action: Update quotagroup command execution fails.
  User Action: Ensure that valid quotagroup details are provided and try again.

- **HVS-QGUD008: Quotagroup update failed. Exception occurred while resizing quotagroup %s**
  Explanation: Quotagroup update failed because there was an error while resizing quotagroup.
  System Action: Update quotagroup command execution fails.
  User Action: Ensure that valid quotagroup details are provided and try again.

- **HVS-QGUD009: Quotagroup update failed because shrinking of quotagroup %s from %s to %s is not allowed**
  Explanation: Quotagroup update failed because shrinking of quotagroup is not allowed.
  System Action: Update quotagroup command execution fails.
  User Action: Ensure that valid quotagroup details are provided and try again.

- **HVS-QGUD010: Quotagroup update failed. Size %s of quotagroup %s is lower than the minimum (%s Bytes). Use a size that is equal to or greater than the minimum size supported**
  Explanation: Quotagroup update failed because the size of quotagroup is lower than the minimum allowed size.
  System Action: Update quotagroup command execution fails.
  User Action: Ensure that quotagroup size is equal to or greater than the minimum size supported and try again.

- **HVS-QGUD011: Quotagroup update failed to update passthrough quotagroup %s because it is used by the running virtual server %s**
  Explanation: Quotagroup update failed because it is used by the running virtual server.
  System Action: Update quotagroup command execution fails.
  User Action: Ensure that valid quotagroup details are provided and try again.

- **HVS-QGUD012: Quotagroup update failed because quotagroup is locked and resize not allowed for poolid %s quotagroup %s**
  Explanation: Quotagroup update failed because quotagroup is locked and resize is not allowed.
  System Action: Update quotagroup command execution fails.
  User Action: Ensure that valid quotagroup details are provided and try again.

- **HVS-QGUD013: Quotagroup update failed because quotagroup modification of external key config is already in progress**
  Explanation: Quotagroup update failed because quotagroup modification of external key config is already in progress.
  System Action: Update quotagroup command execution fails.
  User Action: Ensure that valid quotagroup details are provided and try again.

- **HVS-QGUD014: Quotagroup update failed because there was no response from the external key daemon**
  Explanation: Quotagroup update failed because there was no response from the external key daemon.
  System Action: Update quotagroup command execution fails.
  User Action: Ensure that valid quotagroup details are provided and try again.

- **HVS-QGUD015: Quotagroup update failed. Reason: %s. Provide valid quotagroup details and try again**
  Explanation: Quotagroup update failed because there was an error while updating quotagroup.
  System Action: Update quotagroup command execution fails.
  User Action: Retry the command, if the problem persists obtain the appliance dump, LMS logs, and contact IBM Support.

- **HVS-QGUD016: Error while trying to remove the passthrough file %s (errno=%s). Provide valid details and try again**
  Explanation: Quotagroup update failed because there was an error while trying to remove the passthrough file.
  System Action: Update quotagroup command execution fails.
  User Action: Retry the command, if the problem persists obtain the appliance dump, LMS logs, and contact IBM Support.

HVS-QGDE00x: Messages for delete quotagroup command

This section lists the messages you might encounter while running quotagroup delete command.

- **HVS-QGDE001: The command to delete the quotaGroup failed due to an internal server error**
  Explanation: An internal server error occurred, resulting in a request to delete a quotagroup failure.
  System Action: Delete quotagroup command execution fails.
  User Action: Retry the command. If the problem persists, obtain an appliance dump and LMS, and contact IBM Support.

- **HVS-QGDE002: Quotagroup delete failed. The quotagroup %s cannot be removed because the container(s) %s are still using it**
  Explanation: Quotagroup delete failed because the containers are still using it.
  System Action: Delete quotagroup command execution fails.
  User Action: Ensure that valid quotagroup details are provided and try again.

- **HVS-QGDE003: Quotagroup delete failed to remove logical volume %s on pool %s**
  Explanation: Quotagroup delete failed because there was an error to remove logical volume.
  System Action: Delete quotagroup command execution fails.
  User Action: Retry the command, if the problem persists obtain the appliance dump, LMS logs, and contact IBM Support.

- **HVS-QGDE004: Quotagroup delete failed to remove logical volume %s on pool %s, volume group does not exist**
  Explanation: Quotagroup delete failed because the volume group does not exist.
  System Action: Delete quotagroup command execution fails.
  User Action: Ensure that valid quotagroup details are provided and try again.

- **HVS-QGDE005: Quotagroup update failed. Multiple virtual servers are owning the passthrogh quotagroup %s: %s. Provide valid quotagroup details and try again**
  Explanation: Quotagroup delete failed because multiple virtual servers are owning the passthrogh quotagroup.
  System Action: Delete quotagroup command execution fails.
  User Action: Ensure that valid quotagroup details are provided and try again.

- **HVS-QGDE006: Quotagroup delete failed because the quotagroup 'appliance_data' cannot be removed**
  Explanation: Quotagroup delete failed because the quotagroup 'appliance_data' cannot be removed.
  System Action: Delete quotagroup command execution fails.
  User Action: Ensure that valid quotagroup details are provided and try again.

- **HVS-QGDE007: Error removing directory %s : errno: %s. Obtain an appliance dump and contact IBM Support**
  Explanation: Quotagroup delete failed because there was an error in removing directory.
  System Action: Delete quotagroup command execution fails.
  User Action: Retry the command, if the problem persists obtain the appliance dump, LMS logs, and contact IBM Support.

HVS-QQGLI00x: Messages for quotagroup list command

This section lists the messages you might encounter while running quotagroup list command.

- **HVS-QGLI001: The command to list the quotaGroup failed due to an internal server error**
  Explanation: An internal server error occurred, resulting in a request to list a quotagroup failure.
  System Action: List quotagroup command execution fails.
  User Action: Retry the command if the problem persists, obtain an appliance dump and LMS and Contact IBM Support.

HVS-QGSW00x: Messages for quotagroup show command

This section lists the messages you might encounter while running quotagroup show command.

- **HVS-QGSW001: The command to show the quotaGroup failed due to an internal server error**
  Explanation: An internal server error occurred, resulting in a request to show a quotagroup failure.
  System Action: Show quotagroup command execution fails.
  User Action: Retry the command if the problem persists, obtain an appliance dump, and LMS and contact IBM Support.

HVS-QGYY00x: Messages for some quotagroup commands

This section lists the messages you might encounter while running some quotagroup commands.

- **HVS-QGYY001: Could not find quotagroup %s. Provide valid quotagroup name and try again**
  Explanation: Quotagroup operation failed because quotagroup name is invalid.
  System Action: Quotagroup command execution fails.
  User Action: Ensure that a valid quotagroup name is provided and try again.

- **HVS-QGYY002: Size = %s of quotagroup %s is not an integer multiple of extent size (%s Bytes). Provide valid details and try again**
  Explanation: Quotagroup operation failed because quotagroup size is invalid.
  System Action: Quotagroup command execution fails.
  User Action: Ensure that a valid quotagroup size is provided. Quotagroup size should be an integer multiple of extent size.

- **HVS-QGYY003: Failed to create directory for quotagroup %s errno: %s. Obtain an appliance dump and contact IBM Support**
  Explanation: Quotagroup operation failed because there was an error to create directory.
  System Action: Quotagroup command execution fails.
  User Action: Retry the command, if the problem persists obtain the appliance dump, LMS logs, and contact IBM Support.

- **HVS-QGYY004: Failed to set permissions for quotagroup %s. Obtain an appliance dump and contact IBM Support**
  Explanation: Quotagroup operation failed because there was an error in setting the permissions.
  System Action: Quotagroup command execution fails.
  User Action: Retry the command, if the problem persists obtain the appliance dump, LMS logs, and contact IBM Support.

- **HVS-QGYY005: Syntax error in passthrough file %s. Provide valid details and try again**
  Explanation: Quotagroup operation failed because there was an error in passthrough file.

System Action: Quotagroup command execution fails.
User Action: Ensure that a valid passthrough file is provided and try again.

- **HVS-QGYY006: Error creating directory %s already exists. Obtain an appliance dump and contact IBM Support**
Explanation: Quotagroup operation failed because the directory already exists.
System Action: Quotagroup command execution fails.
User Action: Retry the command, if the problem persists obtain the appliance dump, LMS logs, and contact IBM Support.

- **HVS-QGYY007: An Input/Output error occurred reading the configuration file for container %s errno: %s. Obtain an appliance dump and contact IBM Support**
Explanation: Quotagroup operation failed because there was an error while reading the configuration for container %s.
System Action: Quotagroup command execution fails.
User Action: Retry the command. If the problem persists obtain the appliance dump, LMS logs, and contact IBM Support.

- **HVS-QGYY008: Status entry already exists for pool_id %s quotagroup %s. Provide valid details and try again**
Explanation: Quotagroup operation failed because the status of pool id %s quotagroup %s already exists.
System Action: Quotagroup command execution fails.
User Action: Ensure that valid quotagroup details are provided and try again.

- **HVS-QGYY009: External key operation %s on non-external key quotagroup %s. Obtain an appliance dump and contact IBM Support**
Explanation: Quotagroup operation failed because the external key operation %s was executed on non-external key quotagroup %s.
System Action: Quotagroup command execution fails.
User Action: Ensure that valid quotagroup details are provided and try again.

- **HVS-QGYY010: Failed to set permissions for pool_id %s quotagroup %s. Obtain an appliance dump and contact IBM Support**
Explanation: Quotagroup operation failed because there was an error in setting the permissions.
System Action: Quotagroup command execution fails.
User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

- **HVS-QGYY011: The command failed because the data pool is not ready or initialized completely. Add the disks to data pool or verify the initialization is complete using Secure Service Container User Interface and retry**
Explanation: The operation failed because the disks are not added.
System Action: The command execution fails.
User Action: Add the disks to data pool or verify the initialization is complete and retry the operation. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

- **HVS-QGYY012: The operation failed because there is not enough disk space available for file %s with size=%s Bytes. Provide valid details and try again**
Explanation:: The operation failed because there is not enough disk space available for file %s with size=%s Bytes.
System Action: The command execution fails.
User Action: Ensure that valid virtual server details are provided and try again. Ensure that valid virtual server details are provided and try again.

Registry command messages

This section lists the messages you might encounter while running quotagroup show command.

HVS-RYAD00x: Messages for registry add command

This section lists the messages you might encounter while running registry add command.

- **HVS-RYAD001: Add registry failed. Error occurred while creating %s file. Refer the product documentation**
  Explanation: An internal error occurred while creating the registry file.
  System Action: Add registry command execution fails.
  User Action: Retry the command. If the problem persists obtain an appliance and LMS dump, and contact IBM Support.

- **HVS-RYAD002: Add registry failed. Error occurred while reading %s file. Refer the product documentation**
  Explanation: An internal processing error occurred while reading the registry file.
  System Action: Add registry command execution fails.
  User Action: Retry the command. If the problem persists obtain an appliance and LMS dump, and contact IBM Support.

- **HVS-RYAD003: Add registry failed. Registry is already present. Provide a different registry**
  Explanation: Registry add failed because the given registry is already present.
  System Action: Add registry command execution fails.
  User Action: Provide a different registry and retry the command. Refer the registry add commands section of the IBM Documentation.

- **HVS-RYAD004: Add registry failed. Error occurred while writing %s file. Refer the product documentation**
  Explanation: An internal processing error occurred while writing in the registry file.
  System Action: Add registry command execution fails.
  User Action: Retry the command. If the problem persists obtain an appliance and LMS dump, and contact IBM Support.

- **HVS-RYAD005: Add registry failed. Error while encrypting the message**
  Explanation: An error occurred while encrypting the message.
  System Action: Add registry command execution fails.
  User Action: Ensure that valid details are provided and retry the command. Refer the registry add commands section of the IBM Documentation.

- **HVS-RYAD006: Add registry failed to validate name. Provide name greater or equal to 1 character. Special characters such as `"_"`, `""-""`, are allowed**
  Explanation: An invalid registry name is provided causing the failure to add a registry.
  System Action: Add registry command execution fails.
  User action: Ensure that name is greater or equal to 1 character and special characters such as `"_"`, `""-""`, are allowed. Retry the command.

- **HVS-RYAD009: Add registry failed because the username, password, or URL provided was invalid**
  Explanation: An invalid username, password, or URL is provided causing the failure to add a registry.
  System Action: Add registry command execution fails.
  User Action: Ensure that username, password, or URL are valid and retry the command.

HVS-RYUD00x: Messages for registry update command

This section lists the messages you might encounter while running registry update command.

- **HVS-RYUD001: Update registry failed. Error while encrypting the message**
  Explanation: An error occurred while encrypting the message.
  System Action: Update registry command execution fails.
  User Action: Ensure that valid details are provided and retry the command. Refer the registry commands section of the IBM Documentation.

- **HVS-RYUD002: Update registry failed. Error occurred while reading %s file**
  Explanation: An internal processing error occurred while reading the registry file.
  System Action: Update registry command execution fails.
  User Action: Retry the command. If the problem persists obtain an appliance and LMS dump, and contact IBM Support.

- **HVS-RYUD003: Update registry failed. Error occurred while writing %s file**
  Explanation: An internal processing error occurred while writing in the registry file.
  System Action: Update registry command execution fails.
  User Action: Retry the command. If the problem persists obtain an appliance and LMS dump, and contact IBM Support.

- **HVS-RYUD004: Update registry failed because the username, password, or URL provided was invalid**
  Explanation: An invalid username, password, or URL is provided causing the failure to update a registry.
  System Action: Update registry command execution fails.
  User Action: Ensure that username, password, or URL are valid and retry the command.

- **HVS-RYUD005: Update registry failed. Given name does not exist in the registry list. Provide a valid name that exists**
  Explanation: An invalid name is provided causing the failure to update a registry.
  System Action: Update registry command execution fails.
  User Action: Ensure that the name provided to update registry exists and retry the command.

- **HVS-RYUD006 Update registry failed. Error while decrypting the message**
  Explanation: An error occurred while decrypting the message.
  System Action: Update registry command execution fails.
  User Action: Ensure that valid details are provided and retry the command. Refer the registry commands section of the IBM Documentation.

HVS-RYDE00x: Messages for registry delete command

This section lists the messages you might encounter while running registry delete command.

- **HVS-RYDE001: Delete registry failed. Error occurred while reading %s file. Refer the product documentation**
  Explanation: An internal processing error occurred while reading the registry file.
  System Action: Delete registry command execution fails.
  User Action: Retry the command. If the problem persists obtain an appliance and LMS dump, and contact IBM Support.

- **HVS-RYDE002: Delete registry %s not found**
  Explanation: An invalid registry is provided causing the failure to delete a registry.
  System Action: Delete registry command execution fails.
  User Action: Ensure that the registry exists and retry the command.

- **HVS-RYDE003: Delete registry failed. Error occurred while writing %s file. Refer the product documentation**
  Explanation: An internal processing error occurred while writing in the registry file.
  System Action: Delete registry command execution fails.
  User Action: Retry the command. If the problem persists obtain an appliance and LMS dump, and contact IBM Support.

HVS-RYSW00x: Messages for registry show command

This section lists the messages you might encounter while the running registry show command.

- **HVS-RYSW001: Show registry failed. Error occurred while reading %s file. Refer the product documentation**
  Explanation: An internal processing error occurred while reading the registry file.

System Action: Show registry command execution fails.
User Action: Retry the command. If the problem persists obtain an appliance and LMS dump, and contact IBM Support.

- **HVS-RYSW002: Show registry failed. Registry is not found. Provide a valid registry**
  Explanation: An invalid registry is provided causing the failure to show the registry.
  System Action: Show registry command execution fails.
  User Action: Ensure that the registry exists and retry the command.

HVS-RYLI00x: Messages for registry list command

This section lists the messages you might encounter while the running registry list command.

- **HVS-RYLI001: List registry failed. Error occurred while reading %s file. Refer the product documentation**
  Explanation: An internal processing error occurred while reading the registry file.
  System Action: List registry command execution fails.
  User Action: Retry the command. If the problem persists obtain an appliance and LMS dump, and contact IBM Support.

Regfile command messages

This section lists the messages you might encounter while running the regfile command.

- **HVS-RFCR001: Create regfile failed. Details :%s. Provide proper values**
  Explanation: Regfile create operation failed as there is an error.
  System Action: Regfile command execution fails.
  User Action: Retry the command, if the problem persists obtain an appliance and LMS dump, and contact IBM Support.

- **HVS-RFCR002: Regfile create command failed to read passphrase for signing private key. Details: %s. Retry the command with proper values**
  Explanation: Regfile create operation failed as there is an error. Ensure passphrase is valid.
  System Action: Regfile command execution fails.
  User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

- **HVS-RFCR003:: Regfile create command failed to get private key path. Details: %s. Retry the command with proper values**
  Explanation: Regfile create operation failed as there is an error. Ensure file provided to config is valid file.
  System Action: Regfile command execution fails.
  User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

- **HVS-RFCR004: Regfile create command failed to encrypt regfile. Details: %s. Retry the command with proper values**
  Explanation: Regfile create operation failed because of an error. Ensure details part of config file are valid.
  System Action: Regfile command execution fails.
  **User Action*: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

- **HVS-RFCR005: Regfile create command failed to read yaml file. Details: %s. Retry the command with valid file**
  Explanation: Regfile create operation failed as there is an error. Ensure config file is valid.
  System Action: Regfile command execution fails.
  User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

- **HVS-RFCR006: Regfile create command failed to parse config file. Details: %s. Retry the command with valid config details in file %s**
  Explanation: Regfile create operation failed as there is an error. Ensure the config file is valid.
  System Action: Regfile command execution fails.
  User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

- **HVS-RFCR007: Regfile create command failed to obtain password. Details: %s. Retry the command with proper registry details**
  Explanation: Regfile create operation failed because of an error. Ensure the registry details are valid.
  System Action: Regfile command execution fails.
  User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

- **HVS-RFCR008: Regfile create command failed to decrypt message with password. Details: %s. Retry the command with proper values**
  Explanation: Regfile create operation failed because of an error. Ensure the registry details are valid.
  System Action: Regfile command execution fails.
  User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

- **HVS-RFCR009: Regfile create command failed because repo parameter was not found in yaml. Retry the command with proper values**
  Explanation: Regfile create operation failed because of an error. Ensure the details part of config file are valid.
  System Action: Regfile command execution fails.
  User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

- **HVS-RFCR010: Regfile create command failed because %s not found. Set the 'content_trust_json_file_path' parameter**
  Explanation: Regfile create operation failed as there is an error. Ensure the 'content_trust_json_file_path' is set.
  System Action: Regfile command execution fails.
  User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

- **HVS-RFCR011: Regfile create command found invalid value: env[allowlist] = '[]' in the secure build yaml file. Retry the command with proper values**
  Explanation: Regfile create operation failed because of an error. Provide valid details for the env[allowlist] parameter.
  System Action: Regfile command execution fails.
  User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

- **HVS-RFCR012: Regfile create command failed to read the file. Details: %s. Retry the command with a valid file**
  Explanation: Regfile create operation failed because of an error. Ensure public_key_path is valid.
  System Action: Regfile command execution fails.
  User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

- **HVS-RFCR013: Regfile create command failed because the 'public_key_path' parameter was not found in the yaml. Retry the command with proper values**
  Explanation: Regfile create operation failed because of an error. Ensure the 'public_key_path' is set.
  System Action: Regfile command execution fails.
  User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

- **HVS-RFCR014: Regfile create command failed to parse. Details: %s. Retry the command with proper values**
  Explanation: Regfile create operation failed because of an error. Ensure the 'content_trust_json_file_path' is set.
  System Action: Regfile command execution fails.
  User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

Repository command messages

This section lists the messages you might encounter while running the repository commands.

HVS-RELI00x: Messages for repository list command

This section lists the messages you might encounter while the running registry list command.

- **HVS-RELI001: The command to list repository failed due to an internal server error**
  Explanation: Repository list operation failed as there is an error.
  System Action: Repository command execution fails.
  User Action: Retry the command, if the problem persists obtain an appliance and LMS dump, and contact IBM Support.

- **HVS-RELI002: Unable to execute %s. Method is not defined. Obtain an appliance dump and contact IBM Support**
  Explanation: Repository list operation failed because there was an error to execute %s.
  System Action: List repository command execution fails.
  User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

HVS-RERG00x: Messages for repository register command

This section lists the messages you might encounter while the running registry register command.

- **HVS-RERG001 The command to register repository failed due to an internal server error**
  Explanation: Repository register operation failed as there is an error.
  System Action: Repository command execution fails.
  User Action: Ensure that a valid repository configuration file was defined properly. Refer the repository register commands section of the IBM Documentation.

- **HVS-RERG002: Register repository failed. Error occurred while reading %s file. Provide valid file**
  Explanation: Repository register operation failed as there is an error.
  System Action: Repository command execution fails.
  User Action: Ensure that a valid repository configuration file exits and have defined permissions. Refer the repository register commands section of the IBM Documentation.

- **HVS-RERG003: Register repository failed. Error in parsing the repository details. Provide proper values**
  Explanation: Repository register operation failed as there is an error.
  System Action: Repository command execution fails
  User Action: Ensure that a valid repository configuration details are provided. Refer the repository register commands section of the IBM Documentation.

- **HVS-RERG004: Register repository failed. Repository definition file %s already exists. Provide different repository definition file and try again**
  Explanation: Repository register operation failed because repository definition file already exists.
  System Action: Register repository command execution fails.
  User Action: Provide different repository definition file and try again.

- **HVS-RERG005: Register repository failed. Error:%s Provide valid pgp file and try again**
Explanation: Repository register operation failed because pgp file is invalid.
System Action: Repository command execution fails.
User Action: Ensure that a valid pgp file is provided. Repository gp file must begin with `-----BEGIN PGP MESSAGE-----`.

- **HVS-RERG006: Register repository failed. Error:%s Provide valid repository ID and try again**
Explanation: Repository register operation failed because repository ID is invalid.
System Action: Repository command execution fails.
User Action: Ensure that a valid repository ID is provided. Repository ID should not start with __ and should not be more than 253 characters.

HVS-REUD00x: Messages for repository update command

This section lists the messages you might encounter while the running registry update command.

- **HVS-REUD001: The command to update repository failed due to an internal server error**
Explanation: Repository update operation failed as there is an error.
System Action: Repository command execution fails.
User Action: Retry the command, if the problem persists obtain an appliance and LMS dump, and contact IBM Support.

- **HVS-REUD002: Update repository failed. Error occurred while reading %s file. Provide valid file**
Explanation: Repository update operation failed as there is an error.
System Action: Repository command execution fails.
User Action: Ensure that a valid repository configuration file exits and have defined permissions. Refer the repository update commands section of the IBM Documentation.

- **HVS-REUD003: Update repository failed. Error in parsing the repository details. Provide proper values**
Explanation: Repository update operation failed as there is an error.
System Action: Repository command execution fails.
User Action: Ensure that a valid repository configuration details are provided. Refer the repository register commands section of the IBM Documentation.

- **HVS-REUD004: Update repository failed. Error:%s Provide valid repository ID and try again**
Explanation: Repository update operation failed because repository id is invalid.
System Action: Repository command execution fails.
User Action: Ensure that a valid repository id is provided. Repository id should not start with __ and should not be more than 253 characters.

- **HVS-REUD005: Update repository failed.Error:%s Provide valid pgp file and try again**
Explanation: Repository update operation failed because pgp file is invalid.
System Action: Repository command execution fails.
User Action: Ensure that a valid pgp file is provided. Repository gp file must begin with `-----BEGIN PGP MESSAGE-----`.

- **HVS-REUD006: Repository update failed because it is not allowed to update a %s definition file with a %s definition file**
Explanation: Repository update operation failed because it is not allowed to update definition file with a different definition file.
System Action: Update repository command execution fails.
User Action: Provide different repository definition file and try again.

HVS-REDE00x: Messages for repository delete command

This section lists the messages you might encounter while the running registry update command.

- **HVS-REDE001: The command to delete repository failed due to an internal server error**
Explanation: Repository delete operation failed as there is an error.

System Action: Repository command execution fails.
User Action: Retry the command, if the problem persists obtain an appliance and LMS dump, and contact IBM Support.

- **HVS-REDE002: Failed to delete repository as containers or images are using it. Use --force to force delete**
  Explanation: Repository delete operation failed as there is an error.
  System Action: Repository command execution fails.
  User Action: Retry the command using --force. If the problem persists obtain an appliance and LMS dump, and contact IBM Support.

- **HVS-REDE003: Delete repository failed. Error:%s Provide valid repository id and try again**
  Explanation: Repository delete operation failed because repository id is invalid.
  System Action: Repository command execution fails.
  User Action: Ensure that a valid repository id is provided. Repository id should not start with __ and should not be more than 253 characters.

HVS-RESW00x: Messages for repository show command

This section lists the messages you might encounter while running repository update commands.

- **HVS-RESW001 The command to show repository failed due to an internal server error**
  Explanation: Repository show operation failed as there is an error.
  System Action: Repository command execution fails.
  User Action: Retry the command, if the problem persists obtain an appliance and LMS dump, and contact IBM Support.

- **HVS-RESW002: Show repository failed. Error:%s Provide valid repository id and try again**
  Explanation: Repository show operation failed because repository id is invalid.
  System Action: Repository command execution fails.
  User Action: Ensure that a valid repository id is provided. Repository id should not start with __ and should not be more than 253 characters.

HVS-RESW00x: Messages for some repository commands

This section lists the messages you might encounter while the running some repository commands.

- **HVS-REYY001: Repository %s was not found. Provide valid repository details and try again**
  Explanation: Repository operation failed because repository was not found.
  System Action: Repository command execution fails.
  User Action: Provide valid repository details and try again.

- **HVS-REYY002: Failed to login to repository %s. Provide valid repository details and try again**
  Explanation: Repository operation failed because there was an error to login to repository.
  System Action: Repository command execution fails.
  User Action: Provide valid repository details and try again.

- **HVS-REYY003 Repository definition file %s does not exist. Provide valid repository details and try again**
  Explanation: Repository operation failed because the repository definition file does not exist.
  System Action: Repository command execution fails.
  User Action: Provide valid repository definition file and try again.

- **HVS-REYY004: Found invalid payload: %s. Provide valid repository details and try again**
  Explanation: Repository operation failed because invalid payload was found.
  System Action: Repository command execution fails.
  User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

- **HVS-REYY005: Signature validation and decryption failed. GPG returned rc=%s: %s. Provide valid details and try again**

Explanation: Repository operation failed because there was an error while validating signature and decrypting.
System Action: Repository command execution fails.
User Action: Provide valid repository details and try again.

- **HVS-REYY006: Verifying and decrypting using gpg failed. Shell returned rc=%s: %s. Provide valid details and try again**
  Explanation: Repository operation failed because there was an error while validating signature and decrypting using gpg.
  System Action: Repository command execution fails.
  User Action Provide valid repository details and try again.

- **HVS-REYY007: Gpg returned rc=0, but its output misses the 'Good signature' line: %s. Obtain an appliance dump and contact IBM Support**
  Explanation: Repository operation failed because output misses the 'Good signature' line.
  System Action: Repository command execution fails.
  User Action: Provide valid repository details and try again.

- **HVS-REYY008: The REST call %s does not support the version: %s. Provide valid details and try again**
  Explanation: Repository operation failed because the REST call does not support the version.
  System Action: Repository command execution fails.
  User Action: Provide valid details and try again.

- **HVS-REYY009: The repository definition has more than one class defined. Provide valid details and try again**
  Explanation: Repository operation failed because the repository definition has more than one class defined.
  System Action: Repository command execution fails.
  User Action: Provide valid repository definition details and try again.

- **HVS-REYY010: The repository definition has no classes defined. Provide valid details and try again**
  Explanation: Repository operation failed because there was error in repository definition file.
  System Action: Repository command execution fails.
  User Action: Provide valid repository definition details and try again.

- **HVS-REYY011: Failed to save the repository definition file. Obtain an appliance dump and contact IBM Support**
  Explanation: Repository operation failed because there was error to save the repository definition file.
  System Action: Repository command execution fails.
  User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

Root command messages

This section lists the messages you might encounter while running the root command.

- **HVS-ROOT001: Command Execution Error %s**
  Explanation: %s command execution fails because required flags are not set.
  System Action: %s command execution fails.
  User Action: Ensure you provide the required flag values and retry the command.

- **HVS-ROOT002: Internal Server Error. Error due to %s. Obtain an appliance dump and contact IBM Support**
  Explanation: The operation failed as there is an internal server error.
  System Action: The command execution fails.
  User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

Secure Build command messages

This section lists the messages you might encounter while running the Secure Build command.

- **HVS-SBMF001: Getting secure build manifest failed. Error in parsing the manifest details. Verify if the values are defined correctly**
  Explanation: Get secure build manifest operation failed as there is an error while validating the manifest details.
  System Action: Get secure build manifest command execution fails.
  User Action: Ensure that valid manifest details are provided and retry the command. If the problem still persists, obtain the appliance logs, LMS dump and contact IBM Support.

- **HVS-SBRF001: Executing command secure build regfile failed. Error in reading passphrase. Verify the passphrase and try again**
  Explanation: An invalid passphrase is provided causing the failure to get the secure build regfile.
  System Action: Get secure build regfile command execution fails.
  User Action: Ensure that a valid passphrase is provided and retry the command.

    - **HVS-SBRF002: Executing command secure build regfile failed. Error while parsing the json**
      Explanation: An error occurred while processing the input parameters, resulting in a request to secure build regfile failure.
      System Action: Secure build regfile fails.
      User Action: Ensure you pass the supported values and retry the command. Refer the Secure Build commands section of the IBM Documentation.

    - **HVS-SBRF003: Executing command secure build regfile failed. Error while reading the %s config file. Verify if config file is defined correctly**
      Explanation: An error occurred while reading parameters from the yaml file.
      System Action: Secure build regfile command execution fails.
      User Action: Ensure that valid details are provided in the yaml file and retry the command. If the problem persists, obtain the appliance logs, LMS dump, and contact IBM Support.

    - **HVS-SBRF004: Executing command secure build regfile failed. Error while reading the %s public key file. Verify if public key file is defined correctly**
      Explanation: An error occurred while reading parameters from the public key file.
      System Action: Secure build regfile command execution fails.
      User Action: Ensure that valid details are provided in the public key file and retry the command. If the problem persists, obtain the appliance logs, LMS dump, and contact IBM Support.

    - **HVS-SBRF005: Executing command secure build regfile failed. Error while parsing the %s config file. Verify if config file is defined correctly**
      Explanation: An error occurred while reading parameters from the yaml file.
      System Action: Secure build regfile command execution fails.
      User Action: Ensure that valid details are provided and retry the command. If the problem persists obtain the appliance logs, LMS dump, and contact IBM Support.

    - **HVS-SBRF006: Secure Build regfile failed. Error occurred during encryption. Details: %s. Provide valid passphrase and retry**
      Explanation: An error occurred while encrypting the message.
      System Action: Secure build regfile command execution fails.
      User Action: Provide a valid passphrase and retry the command. Refer the Secure Build commands section of the IBM Documentation.

- **HVS-SBGE001: Executing command secure build %s failed. Failed to load certificate %s for secure communication with Secure Build server. Verify if its a valid certificate and retry**
  Explanation: Invalid secure build server certificate is configured resulting in failure to execute the secure build command.

System Action: Secure build %s command execution fails.
User Action: Ensure that valid secure build server certificate is configured. Retry the command and if the problem still persists obtain an appliance and LMS dump, and contact IBM Support.

- **HVS-SBGE002: Executing command secure build %s failed. Error while creating http request for url %s. Verify if the URL and config details are valid**
  Explanation: An invalid URL and/or request body is provided causing the failure to execute the secure build command.
  System Action: Secure build %s command execution fails.
  User Action: Ensure that a valid URL and/or config details are provided and retry the command. Refer the Secure Build commands section of the IBM Documentation.

- **HVS-SBGE003: Executing command secure build %s failed. Secure Build Server is not reachable. Check the network connectivity and retry**
  Explanation: An internal server error occurred, resulting in failure to execute secure build command.
  System Action: Secure build %s command execution fails.
  User Action: Ensure there is proper network connectivity and retry the command. If the problem persists obtain the appliance dump and LMS logs, and contact IBM Support.

- **HVS-SBGE004: Executing command secure build %s failed. Error reading the response from secure build server**
  Explanation: Secure build command execution failed as there was error reading the response from secure build server.
  System Action: Secure build %s command execution fails.
  User Action: Retry the command if the problem persists, obtain the appliance and LMS dump, and contact IBM Support.

- **HVS-SBGE005: Executing command secure build %s failed**
  Explanation: Secure build command execution failed as there is an error while reading the response body.
  System Action: Secure build %s command execution fails.
  User Action: Ensure that valid details are provided and retry the command. If the problem still persists, obtain the appliance logs, LMS dump, and contact IBM Support.

- **HVS-SBGE006: Executing command secure build %s failed**
  Explanation: Secure build command execution failed as there is an error while validating the manifest details.
  System Action: Secure build %s command execution fails.
  User Action: Ensure that valid URL, port, cert_path, and key_path are provided and retry the command. If the problem still persists obtain the appliance logs, LMS dump and Contact IBM Support.

- **HVS-SBPO001: Executing command secure build %s failed. Failed to load certificate %s for secure communication with Secure Build server. Verify if it is a valid certificate and retry**
  Explanation: Invalid secure build server certificate is configured resulting in failure to execute the secure build command.
  System Action: Secure build %s command execution fails.
  User Action: Ensure that valid secure secure build server certificate is configured. Retry the command and if the problem still persists obtain the appliance and LMS dump contact IBM Support.

- **HVS-SBPO002: Executing command secure build %s failed. Error while creating http request for URL %s. Verify if URL and config details are valid**
  Explanation: An invalid URL and/or request body is provided causing the failure to execute the secure build command.
  System Action: Secure build %s command execution fails.
  User Action: Ensure that a valid URL and/or config details are provided and retry the command. Refer the Secure Build commands section of the IBM Documentation.

- **HVS-SBPO003: Executing command secure build %s failed. Secure Build Server is not reachable. Please check network connectivity and retry**
  Explanation: An internal server error occurred, resulting in failure to execute secure build command.

System Action: Secure build %s command execution fails.
User Action: Ensure proper network connectivity and retry the command, if the problem persists obtain the appliance dump and LMS logs and Contact IBM Support.

- **HVS-SBPO004: Executing command secure build %s failed. Error reading the response from secure build server**
  Explanation: Secure build command execution failed as there was error reading the response from secure build server.
  System Action: Secure build %s command execution fails.
  User Action: Retry the command. If the problem persists, obtain the appliance and LMS dump, and contact IBM Support.

- **HVS-SBPO005: Executing command secure build %s failed**
  Explanation: Secure build command execution failed as there is an error while reading the response body.
  System Action: Secure build %s command execution fails.
  User Action: Ensure that valid details are provided and retry the command. If the problem still persists, obtain the appliance logs, LMS dump, and contact IBM Support.

- **HVS-SBPO006: Executing command secure build %s failed**
  Explanation: Secure build command execution failed as there is an error while validating the manifest details.
  System Action: Secure build %s command execution fails.
  User Action: Ensure that valid URL, port, cert_path, and key_path are provided and retry the command. If the problem persists, obtain the appliance logs, LMS dump, and contact IBM Support.

- **HVS-SBPO007: Executing command secure build %s failed. Secure build server returned empty body**
  Explanation: Secure build command execution failed as server returned empty body.
  System Action: Secure build %s command execution fails.
  User Action: Retry the command if the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

- **HVS-SBPU001: Executing command secure build %s failed. Failed to load certificate %s for secure communication with Secure Build server. Verify if its a valid certificate and retry**
  Explanation: Invalid secure build server certificate is configured resulting in failure to execute the secure build command.
  System Action: Secure build %s command execution fails.
  User Action: Ensure that valid secure build server certificate is configured. Retry the command and if the problem still persists, obtain the appliance and LMS dump, and contact IBM Support.

- **HVS-SBPU002: Executing command secure build %s failed. Error while creating http request for URL %s. Verify if URL is valid**
  Explanation: An invalid URL is provided causing the failure to execute the secure build command.
  System Action: Secure build %s command execution fails.
  User Action: Ensure that a valid URL is provided and retry the command. Refer the Secure Build commands section of the IBM Documentation.

- **HVS-SBPU003: Executing command secure build %s failed. Secure Build Server is not reachable. Check network connectivity and retry**
  Explanation: An internal server error occurred, resulting in failure to execute secure build command.
  System Action: Secure build %s command execution fails.
  User Action: Ensure proper network connectivity and retry the command. If the problem persists, obtain the appliance dump and LMS logs, and contact IBM Support.

- **HVS-SBPU004: Executing command secure build %s failed. Error reading the response from secure build server**
  Explanation: Secure build command execution failed as there was error reading the response from secure build server.
  System Action: Secure build %s command execution fails.

User Action: Retry the command if the problem persists, obtain the appliance and LMS dump, and contact IBM Support.

- **HVS-SBPU005: Executing command secure build %s failed**
  Explanation: Secure build command execution failed as there is an error while reading the response body.
  System Action: Secure build %s command execution fails.
  User Action: Ensure that valid details are provided and retry the command. If the problem still persists obtain the appliance logs, LMS dump, and contact IBM Support.

- **HVS-SBPU006: Executing command secure build %s failed**
  Explanation: Secure build command execution failed as there is an error while validating the manifest details.
  System Action: Secure build %s command execution fails.
  User Action: Ensure that valid URL, port, cert_path, and key_path are provided and retry the command. If the problem persists, obtain an appliance logs, LMS dump, and contact IBM Support.

- **HVS-SBPU007: Executing command secure build %s failed. Secure build server not initialized yet**
  Explanation: Secure build command execution failed as secure build server is not initialized.
  System Action: Secure build %s command execution fails.
  User Action: Ensure that secure build server is initialized and retry the command. If the problem persists, obtain the appliance logs, LMS dump, and contact IBM Support.

- **HVS-SBDE001: Executing command secure build %s failed. Failed to load certificate %s for secure communication with Secure Build server. Verify if its a valid certificate and retry**
  Explanation: Invalid secure build server certificate is configured resulting in failure to execute the secure build command.
  System Action: Secure build %s command execution fails.
  User Action: Ensure that valid secure build server certificate is configured. Retry the command and if the problem persists, obtain the appliance and LMS dump, and contact IBM Support.

- **HVS-SBDE002: Executing command secure build %s failed. Error while creating http request for URL %s. Verify if URL is valid**
  Explanation: An invalid URL is provided causing the failure to execute the secure build command.
  System Action: Secure build %s command execution fails.
  User Action: Ensure that a valid URL is provided and retry the command. Refer the Secure Build commands section of the IBM Documentation.

- **HVS-SBDE003: Executing command secure build %s failed. Secure Build Server is not reachable. Check network connectivity and retry**
  Explanation: An internal server error occurred, resulting in failure to execute secure build command.
  System Action: Secure build %s command execution fails.
  User Action: Ensure proper network connectivity and retry the command. If the problem persists, obtain the appliance dump and LMS logs, and contact IBM Support.

- **HVS-SBDE004: Executing command secure build %s failed. Error reading the response from secure build server**
  Explanation: Secure build command execution failed as there was error reading the response from secure build server.
  System Action: Secure build %s command execution fails.
  User Action: Retry the command. If the problem persists, obtain the appliance and LMS dump, and contact IBM Support.

- **HVS-SBDE005: Executing command secure build %s failed**
  Explanation: Secure build command execution failed as there is an error while reading the response body.
  System Action: Secure build %s command execution fails.
  User Action: Ensure that valid details are provided and retry the command. If the problem persists obtain the appliance logs, LMS dump, and contact IBM Support

- **HVS-SBDE006: Executing command secure build %s failed**
  Explanation: Secure build command execution failed as there is an error while validating the manifest details.
  System Action: Secure build %s command execution fails.
  User Action: Ensure that valid URL, port, cert_path, and key_path are provided and retry the command. If the problem persists, obtain the appliance logs, LMS dump, and contact IBM Support.

- **HVS-SBPA001: Executing command secure build %s failed. Failed to load certificate %s for secure communication with Secure Build server. Verify if its a valid certificate and retry**
  Explanation: Invalid secure build server certificate is configured resulting in failure to execute the secure build command.
  System Action: Secure build %s command execution fails.
  User Action: Ensure that valid secure build server certificate is configured. Retry the command and if the problem persists, obtain the appliance and LMS dump, and contact IBM Support.

- **HVS-SBPA002: Executing command secure build %s failed. Error while creating http request for url %s. Verify if url and config details are valid**
  Explanation: An invalid URL and/or request body is provided causing the failure to execute the secure build command.
  System Action: Secure build %s command execution fails.
  User Action: Ensure that a valid URL and/or config details are provided and retry the command. Refer the Secure Build commands section of the IBM Documentation.

- **HVS-SBPA003: Executing command secure build %s failed. Secure Build Server is not reachable. Check network connectivity and retry**
  Explanation: An internal server error occurred, resulting in failure to execute secure build command.
  System Action: Secure build %s command execution fails.
  User Action: Ensure proper network connectivity and retry the command. If the problem persists obtain the appliance dump and LMS logs, and Contact IBM Support.

- **HVS-SBPA004: Executing command secure build %s failed. Error reading the response from secure build server**
  Explanation: Secure build command execution failed as there was error reading the response from secure build server.
  System Action: Secure build %s command execution fails.
  User Action: Retry the command. If the problem persists, obtain the appliance and LMS dump, and contact IBM Support.

- **HVS-SBPA005: Executing command secure build %s failed**
  Explanation: Secure build command execution failed as there is an error while reading the response body.
  System Action: Secure build %s command execution fails.
  User Action: Ensure that valid details are provided and retry the command. If the problem persists obtain the appliance logs, LMS dump, and contact IBM Support.

- **HVS-SBPA006: Executing command secure build %s failed**
  Explanation: Secure build command execution failed as there is an error while validating the details.
  System Action: Secure build %s command execution fails.
  User Action: Ensure that valid URL, port, cert_path, and key_path are provided and retry the command. If the problem persists, obtain the appliance logs, LMS dump, and contact IBM Support.

- **HVS-SBBL005: Sb secure build status %s**
  Explanation: A timeout error occurred, resulting in a request to build the secure build failure.
  System Action: Secure build command execution fails.
  User Action: Retry the command. If the problem persists, obtain the appliance and LMS dump, and contact IBM Support.

- **HVS-SBBL006: Sb build failed. Invalid github url and/or branch value. Provide valid details and retry**
  Explanation: Secure build command execution failed because an invalid github url and/or branch value was provided.

System Action: Secure build command execution fails.
User Action: Ensure that a valid github url and/or branch value is provided and retry.

- **HVS-SBIN001: Executing complete secure build failed. Error in reading passphrase. Verify the passphrase and try again**
  Explanation: An invalid passphrase is provided causing the failure to complete the secure build.
  System Action: Complete secure build command execution fails.
  User Action: Ensure that a valid passphrase is provided and retry the command.

- **HVS-SBLO001: Executing command secure build log failed. Log is not found as build is not triggered. Retry after build is triggered**
  Explanation: Secure build log command execution failed as build is not triggered.
  System Action: Secure build log command execution fails.
  User Action: Ensure that build is triggered and retry the command. If the problem persists, obtain the appliance logs, LMS dump and, contact IBM Support.

- **HVS-SBGC001: Getting secure build credentials from config file %s failed. Verify if file exists and is configured correctly**
  Explanation: Get secure build credentials command execution failed as there was error reading the config file.
  System Action: Get secure build credentials command execution fails.
  User Action: Ensure that the config file exists and retry the command. Refer the Secure Build commands section of the IBM Documentation.

- **HVS-SBGC002: Error parsing the %s config file. Verify if config file is defined correctly**
  Explanation: An error occurred while reading parameters from the yaml file.
  System Action: Secure build regfile command execution fails.
  User Action: Ensure that valid details are provided and retry the command. If the problem persists, obtain the appliance logs, LMS dump, and contact IBM Support.

- **HVS-SBGC003: Port value %s is invalid. Provide port value in the range 1-65535**
  Explanation: An error occurred while reading port parameter from the yaml file.
  System Action: Secure build regfile command execution fails.
  User Action: Ensure that valid port details are provided and retry the command. If the problem persists, obtain appliance logs, LMS dump, and contact IBM Support.

- **HVS-SBMF001: Getting secure build manifest failed. Error in parsing the manifest details. Verify if the values are defined correctly**
  Explanation: Get secure build manifest operation failed as there is an error while validating the manifest details.
  System Action: Get secure build manifest command execution fails.
  User Action: Ensure that valid manifest details are provided and retry the command. If the problem persists, obtain appliance logs, LMS dump, and contact IBM Support.

- **HVS-SBMF002: Executing command secure build manifest failed. Error occurred while writing the manifest information into a file**
  Explanation: An error occurred while writing the manifest information into a file.
  System Action: Secure build manifest command execution fails.
  User Action: Ensure that valid details are provided and retry the command. Refer the Secure Build commands section of the IBM Documentation.

- **HVS-SBMF003: Executing command secure build manifest failed. Error occurred while decoding the manifest in base64 format**
  Explanation: An error occurred while decoding the manifest.
  System Action: Secure build manifest command execution fails.
  User Action: Ensure that valid details are provided and retry the command. Refer the Secure Build commands section of the IBM Documentation.

- **HVS-SBPK001: Executing command secure build public key failed. Error in creating json object. Verify if the build name is given correctly and retry**
  Explanation: An invalid build name is provided causing the failure to execute the secure build public key command.
  System Action: Secure build public key command execution fails.
  User Action: Ensure that a valid build name is provided and retry the command.

- **HVS-SBPK002:: Executing command secure build public key failed. Error occurred while writing the public key file**
  Explanation: An error occurred while writing the to the public key file.
  System Action: Secure build public key command execution fails.
  User Action: Ensure that valid details are provided and retry the command. Refer the Secure Build commands section of the IBM Documentation.

- **HVS-SBYJ001: Executing command secure build %s failed. Error while reading the %s config file. Verify if config file is defined correctly**
  Explanation: An error occurred while reading parameters from the config file.
  System Action: Secure build %s command execution fails.
  User Action: Ensure that valid details are provided in the config file and retry the command. If the problem persists, obtain the appliance logs, LMS dump, and contact IBM Support.

- **HVS-SBYJ002: Executing command secure build %s failed. Error while parsing the %s config file. Verify if config file is defined correctly**
  Explanation: An error occurred while reading parameters from the config file.
  System Action: Secure build %s command execution fails.
  User Action: Ensure that valid details are provided and retry the command. If the problem persists, obtain the appliance logs, LMS dump, and contact IBM Support.

- **HVS-SBYJ003: Executing command secure build %s failed. Failed to get docker push server registry. Verify if docker registry has been added with push server information and retry**
  Explanation: An invalid docker registry is provided causing the failure to complete the secure build.
  System Action: Secure build %s command execution fails.
  User Action: Ensure that the docker registry exists and retry the command.

- **HVS-SBYJ004: Executing command secure build %s failed. Failed to decrypt message with password for push server**
  Explanation: An invalid password for push server resulted in failure to the decrypt message.
  System Action: Secure build %s command execution fails.
  User Action: Ensure that valid details are provided and retry the command. If the problem persists, obtain the appliance logs, LMS dump, and contact IBM Support.

- **HVS-SBYJ005: Executing command secure build %s failed. Docker push server name is empty in docker registry. Verify if docker registry has been added with valid push server name and retry**
  Explanation: An invalid docker registry is provided causing the failure to complete the secure build.
  System Action: Secure build %s command execution fails.
  User Action: Ensure that valid detail is provided for docker registry and retry the command.

- **HVS-SBYJ006: Executing command secure build %s failed. Failed to get docker base server registry. Verify if docker registry has been added with base server information and retry**
  Explanation: An invalid docker base server registry is provided causing the failure to complete the secure build.
  System Action: Secure build %s command execution fails.
  User Action: Ensure that the docker registry exists with valid base server information and retry the command.

- **HVS-SBYJ007: Executing command secure build %s failed. Failed to decrypt message with password for base server**
  Explanation: An invalid password for base server resulted in failure to the decrypt message.
  System Action: Secure build %s command execution fails.

User Action: Ensure that valid details are provided and retry the command. If the problem persists, obtain the appliance logs, LMS dump, and contact IBM Support.

- **HVS-SBYJ008: Executing command secure build %s failed. Failed to get docker pull server registry. Verify if docker registry has been added with pull server information and retry**
Explanation: An invalid docker pull server registry is provided causing the failure to complete the secure build.
System Action: Secure build %s command execution fails.
User Action: Ensure that the docker registry exists with valid pull server information and retry the command.

- **HVS-SBYJ009: Executing command secure build %s failed. Failed to decrypt message with password for pull server**
Explanation: An invalid password for pull server resulted in failure to the decrypt message.
System Action: Secure build %s command execution fails.
User Action: Ensure that valid details are provided and retry the command. If the problem persists, obtain the appliance logs, LMS dump, and contact IBM Support.

- **HVS-SBYJ010: Executing command secure build %s failed. Error while reading the %s github ssh private key path. Verify if github ssh private key path is defined correctly**
Explanation: An error occurred while reading github ssh private key path.
System Action: Secure build %s command execution fails.
User Action: Ensure that valid github ssh private key path is provided and retry the command. If the problem persists, obtain the appliance logs, LMS dump, and contact IBM Support.

- **HVS-SBYJ011: Executing command secure build %s failed. Error while parsing the json**
Explanation: Get secure build operation failed as there is an error while validating the json file.
System Action: Secure build %s command execution fails.
User Action: Ensure that valid manifest details are provided and retry the command. If the problem persists, obtain the appliance logs, LMS dump, and contact IBM Support.

- **HVS-SBYJ012: Executing command secure build %s failed. env allowlist in secure build yaml file is empty. Provide valid env allowlist and retry**
Explanation: An invalid env allowlist is provided causing the secure build failure.
System Action: Secure build %s command execution fails.
User Action: Ensure that valid env allowlist details are provided in secure build yaml file and retry the command. If the problem persists, obtain the appliance logs, LMS dump, and contact IBM Support.

- **HVS-SBYJ013: Executing command secure build %s failed because docker.repo should not contain any capital letters**
Explanation: Capital letter(s) were found in docker.repo name causing the secure build failure.
System Action: Secure build %s command execution fails.
User Action: Ensure that valid repo details are provided in the secure build yaml file and retry the command. If the problem persists, obtain appliance logs, LMS dump, and contact IBM Support.

- **HVS-SBYY001: Failed to restart docker daemon. Obtain an appliance dump and contact IBM Support**
Explanation: Repository show operation failed because there was an error to restart docker daemon.
System Action: Repository command execution fails.
User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

Snapshot command messages

This section lists the messages you might encounter while using the snapshot commands.

HVS-SSCR00x: Messages for snapshot create command

This section lists the messages you might encounter while running the snapshot create command.

- **HVS-SSCR001: Snapshot creation failed. Virtual server %s not found. Provide valid Snapshot/Snapshot names and retry**
  Explanation: Invalid Snapshot name provided, resulting Create Snapshot command failure.
  System Action: Create Snapshot command execution fails.
  User Action: Ensure that the a valid virtual server name is provided and retry the command. Refer the snapshot create commands section of the IBM Documentation.

- **HVS-SSCR002: Snapshot creation failed. User not authorized to perform this operation. Check if you have the required permissions to run this command**
  Explanation: User does not have the required permissions to run the command.
  System Action: Create Snapshot command execution fails.
  User Action: Ensure user has the required permissions to run the command. Refer the snapshot create commands section of the IBM Documentation.

- **HVS-SSCR003: Snapshot creation failed. Virtual Server %s not found. Provide valid Virtual Server name and retry**
  Explanation: Invalid Snapshot name provided that results in the failure of the create snapshot command.
  System Action: Create Snapshot command execution fails.
  User Action: Ensure that the a valid virtual server name is provided and retry the command. Refer the snapshot create commands section of the IBM Documentation.

- **HVS-SSCR004: Snapshot creation failed. Unable to process the request. Ensure that the pre-conditions are met. Refer the product documentation**
  Explanation: There is an internal processing error that results in the failure of the create Snapshot command.
  System Action: Create Snapshot command execution fails.
  User Action: Retry the command and if the problem persists, obtain an appliance and LMS dump, and contact IBM Support.

- **HVS-SSCR005: Snapshot creation failed. Internal error or the service is unavailable. Refer the product documentation**
  Explanation:An internal server error occurred that results in a failure of the create snapshot command.
  System Action: Create Snapshot command execution fails.
  User Action: Retry the command and if the problem persists, obtain an appliance and LMS dump, and contact IBM Support.

- **HVS-SSCR006: Snapshot creation failed. Unable to process command completely. Run the hpvs snapshot list command to verify**
  Explanation: An internal processing error occurred that results in a failure of the create snapshot command.
  System Action: Create Snapshot command execution fails.
  User Action: Retry the command. Additionally run the snapshot list command and verify if the snapshot is created successfully. If the problem persists, obtain an appliance and LMS dump and contact IBM Support.

- **HVS-SSCR007: Snapshot creation failed. Internal error or the service is unavailable. Refer to the product documentation**
  Explanation: An internal server error occurred that results in a failure of the create snapshot command.
  System Action: Create Snapshot command execution fails.
  User Action: Retry the command and if the problem persists, obtain an appliance and LMS dump, and contact IBM Support.

- **HVS-SSCR008: Snapshot Create failed. Snapshot %s already exists. Provide a different snapshot name and try again**
  Explanation: Snapshot create operation failed because snapshot name already exists.
  System Action: Create snapshot command execution fails.
  User Action: Provide different snapshot name and try again.

- **HVS-SSCR009: Snapshot create failed. Failed to create snapshot for Virtual Server %s. Provide valid quotagroup details and try again**
  Explanation: Snapshot create operation failed because there was an error to create snapshot for virtual

server.
System Action: Create snapshot command execution fails.
User Action: Ensure that valid snapshot details are provided and try again.

HVS-SSDE00x: Messages for snapshot delete command

This section lists the messages you might encounter while running the snapshot delete command.

- **HVS-SSDE001: Delete snapshot failed.Virtual Server %s or Snapshot %s not found. Provide valid Virtual Server/Snapshot names and retry**
  Explanation: An invalid snapshot name or virtual server name was provided that results in a failure of the delete snapshot command.
  System Action: Delete Snapshot command execution fails.
  User Action: Ensure that the a valid snapshot name and virtual server name is provided and retry the command. Refer the snapshot delete commands section of the IBM Documentation.

- **HVS-SSDE002: Delete snapshot failed. User is not authorized to perform this operation. Check if you have the required permissions to run this command**
  Explanation: User does not have the required permissions to run the command.
  System Action: Delete Snapshot command execution fails.
  User Action: Ensure the user has the required permissions to run the command. Refer the snapshot delete commands section of the IBM Documentation.

- **HVS-SSDE003: Delete snapshot failed. Unable to process the request. Ensure that the pre-conditions are met. Refer the product documentation**
  Explanation: There is an internal processing error that results in a failure of the delete snapshot command.
  System Action: Delete Snapshot command execution fails.
  User Action: Retry the command. Also, run snapshot list command to verify if the given snapshot is deleted. If the problem still persists, obtain an appliance and LMS dump and contact IBM Support.

- **HVS-SSDE004: Delete snapshot failed. Virtual Server %s not found. Provide valid Virtual Server/Snapshot names and retry**
  Explanation: An invalid virtual server or snapshot name was provided that results in a failure of the delete snapshot command.
  System Action: Delete Snapshot command execution fails.
  User Action: Ensure that a valid virtual server or snapshot name is provided and retry the command. Refer to the product documentation for more details on the specific details of this command. Refer the snapshot delete commands section of the IBM Documentation.

- **HVS-SSDE005: Delete snapshot failed. Internal error or the service is unavailable. Refer the product documentation**
  Explanation: An internal server error occurred that results in a failure of the create Snapshot command.
  System Action: Delete Snapshot command execution fails.
  User Action: Retry the command and if the problem persists, obtain an appliance and LMS dump and contact IBM Support.

HVS-SSLI00x: Messages for snapshot list command

This section lists the messages you might encounter while running the snapshot list command.

- **HVS-SSLI001: List snapshots failed. Virtual Server %s not found. Provide valid Snapshot/Snapshot names and retry**
  Explanation: An invalid virtual server name was provided that results in a failure of the list snapshot command.
  System Action: List Snapshot command execution fails.
  User Action: Ensure that the a valid virtual server name is provided and retry the command. Refer the snapshot list commands section of the IBM Documentation.

- **HVS-SSLI002: List snapshots failed. Virtual Server %s not found. Provide valid Virtual Server/Snapshot names and retry**
  Explanation: An invalid virtual server name was provided that results in a failure of the list snapshot command.
  System Action: List Snapshot command execution fails.
  User Action: Ensure that the a valid virtual server name is provided and retry the command. Refer the snapshot list commands section of the IBM Documentation.

- **HVS-SSLI003: List snapshots failed. Unable to process the request. Ensure that the pre-conditions are met. Refer to product documentation for resolution Explanation**: An internal processing error occurred that results in a failure of the list snapshot command.
  System Action: Create Snapshot command execution fails.
  User Action: Retry the command and if the problem persists, obtain an appliance and LMS dump, and contact IBM Support.

- **HVS-SSLI004: List snapshots failed. Internal error or the service is unavailable. Refer the product documentation**
  Explanation: An internal server error occurred that results in a failure of the list snapshot command.
  System Action: List Snapshot command execution fails.
  User Action: Retry the command and if the problem persists, obtain an appliance and LMS dump and contact IBM Support.

- **HVS-SSLI005: List snapshots failed. Internal error or the service is unavailable. Refer the product documentation**
  Explanation: An internal server error occurred that results in a failure of the list snapshot command.
  System Action: List Snapshot command execution fails.
  User Action: Retry the command and if the problem persists, obtain an appliance and LMS dump and contact IBM Support.

- **HVS-SSLI006: List snapshots failed. Unable to process the request. Retry the command or refer the product documentation**
  Explanation: An internal processing error occurred that results in a failure of the list snapshot command.
  System Action: List Snapshot command execution fails.
  User Action: Retry the command and if the problem persists, obtain an appliance and LMS dump and contact IBM Support.

HVS-SSRT00x: Messages for snapshot restore command

This section lists the messages you might encounter while running the snapshot restore command.

- **HVS-SSRT001: Restore snapshot failed. Virtual Server %s or Snapshot %s not found. Provide a valid Virtual Server/Snapshot names and retry**
  Explanation: Invalid virtual server, snapshot name, or both, that results in a failure of the restore snapshot command.
  System Action: Restore Snapshot command execution fails.
  User Action: Ensure that the a valid virtual server and snapshot names are provided and retry the command. Refer the snapshot restore commands section of the IBM Documentation.

- **HVS-SSRT002: Restore snapshot failed. User not authorized to perform this operation. Check if you have the required permissions to run this command Explanation**: User does not have the required permissions to run the command.
  System Action: Restore Snapshot command execution fails.
  User Action: Ensure user has the required permissions to run the command. Refer to the product documentation for more details on the specific details of this command. Refer the snapshot restore commands section of the IBM Documentation.

- **HVS-SSRT003: Restore snapshot failed. Virtual Server %s or Snapshot %s not found. Provide a valid Virtual Server/Snapshot names and retry**

Explanation: Invalid virtual server, snapshot name, or both was provided that results in a failure of the restore snapshot command.
System Action: Restore Snapshot command execution fails.
User Action: Ensure that the a valid virtual server and snapshot names are provided and retry the command. Refer the snapshot restore commands section of the IBM Documentation.

- **HVS-SSRT004: Restore snapshot failed. Unable to process the request. Ensure that the pre-conditions are met. Refer product documentation**
  Explanation: An internal error while processing the request that results in a failure of the restore snapshot command.
  System Action: Restore Snapshot command execution fails.
  User Action: Retry the command and if the problem persists, obtain an appliance and LMS dump and contact IBM Support.

- **HVS-SSRT005: Restore snapshot failed. Internal error or the service is unavailable. Refer the product documentation**
  Explanation: An internal server error occurred that results in a failure of the restore snapshot command.
  System Action: Restore Snapshot command execution fails.
  User Action: Retry the command and if the problem persists, obtain an appliance and LMS dump and contact IBM Support.

- **HVS-SSRT006: Restore snapshot failed. Unable to process the request. Retry the command or refer the product documentation**
  Explanation: An internal server error occurred that results in a failure of the restore snapshot command.
  System Action: Restore Snapshot command execution fails.
  User Action: Retry the command and if the problem persists, obtain an appliance and LMS dump and contact IBM Support.

- **HVS-SSRT007: Snapshot restore failed because the repository information is missing in the snapshot. Provide valid details and try again**
  Explanation: Snapshot restore operation failed because the repository information is missing in the snapshot.
  System Action: Restore snapshot command execution fails.
  User Action: Ensure that valid snapshot details are provided and try again.

- **HVS-SSRT008: Snapshot restore failed to read the repository binding information: %s. Provide valid details and try again**
  Explanation: Snapshot restore operation failed because it failed to read the repository binding information.
  System Action: Restore snapshot command execution fails.
  User Action: Ensure that valid repository binding details are provided and try again.

HVS-SSYY00x: Messages for some snapshot commands

This section lists the messages you might encounter while running some snapshot commands.

- **HVS-SSYY001: Snapshot %s does not exist for container %s. Provide valid snapshot details and try again**
  Explanation: Snapshot operation failed because snapshot does not exist for container.
  System Action: Snapshot command execution fails.
  User Action: Ensure that valid snapshot details are provided and try again.

- **HVS-SSYY002: Virtual Server name is missing in URL. Provide a valid container name and try again**
  Explanation: Snapshot operation failed because virtual server name is missing in URL.
  System Action: Snapshot command execution fails.
  User Action: Ensure that valid virtual server details are provided and try again.

- **HVS-SSYY003: Snapshot name is missing in URL. Provide a valid snapshot name and try again**
  Explanation: Snapshot operation failed because snapshot name is missing in URL.
  System Action: Snapshot command execution fails.
  User Action: Ensure that valid snapshot details are provided and try again.

- **HVS-SSYY004: Failed to set snapshot ownership for container %s. Obtain an appliance dump and contact IBM Support**
  Explanation: Snapshot operation failed because there was an error to set snapshot ownership for container.
  System Action: Snapshot command execution fails.
  User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

- **HVS-SSYY005: Failed to delete snapshot for container %s. Obtain an appliance dump and contact IBM Support**
  Explanation: Snapshot operation failed because there was an error to delete the snapshot for container.
  System Action: Snapshot command execution fails.
  User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

- **HVS-SSYY006: Failure obtaining the size of a block device. Obtain an appliance dump and contact IBM Support**
  Explanation: Snapshot operation failed because there was an error to obtain the size of a block device.
  System Action: Snapshot command execution fails.
  User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

Token operations

This section lists the messages you might encounter while running the token related operations.

- **HVS-TKCT001: Secure Service Container partition is not accessible. Add valid host configuration and retry**
  Explanation: Secure Service Container LPAR host configuration is improper, resulting in token creation failure.
  System Action: Fails to proceed with requested command execution.
  User Action: Ensure that the Secure Service Container LPAR details are properly configured and set on LMS. If the problem persists, obtain an appliance and LMS dump and contact IBM Support.

- **HVS-TKGT001: Secure Service Container partition is not accessible. %s url not found Provide valid host configuration and retry**
  Explanation: An invalid Secure Service Container LPAR host URL is configured resulting in failure to obtain token.
  System Action: Fails to proceed with the requested command execution.
  User Action: Ensure that the a valid Secure Service Container LPAR host is configured and set and retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

- **HVS-TKGT002: Secure Service Container partition is not accessible. Provide valid credentials and retry**
  Explanation: Invalid Secure Service Container LPAR host credentials are configured resulting in failure to obtain token.
  System Action: Fails to proceed with requested command execution.
  User Action: Ensure that valid Secure Service Container LPAR host credentials are configured. Retry the command and if the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

- **HVS-TKGT003: Secure Service Container partition is not accessible due to an internal server error**
  Explanation: An internal server occurred resulting in failure to obtain token resulting in failure with the command execution.
  System Action: Fails to proceed with requested command execution.
  User Action: Retry the command and if the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

Virtual Server command messages

This section lists the messages you might encounter while using the Virtual Server command.

HVS-VSCR00x: Messages for create Virtual Server command

This section lists the messages you might encounter while running the create container command.

- **HVS-VSCR001: The command to create virtual server failed due to an internal server error**
  Explanation: An internal server error occurred, resulting in failure of create Virtual Server command failed.
  System Action: Create Virtual Server command execution fails.
  User Action: Obtain an appliance and LMS dump and contact IBM Support.

- **HVS-VSCR002: Create container failed due to wrong network configuration. Provide valid network configuration**
  Explanation: Create Virtual Server instance failed as an invalid network configuration is provided.
  System Action: Create Virtual Server command execution fails.
  User Action: Ensure you specify valid network configuration details and retry the command. Refer the create Virtual Server commands section of the IBM Documentation.

- **HVS-VSCR003: Create virtual server failed due to wrong quotagroup configuration. Provide a valid quotagroup configuration**
  Explanation: Create virtual server instance failed as an invalid quotagroup configuration is provided.
  System Action: Create virtual server command execution fails.
  User action: Ensure you specify valid quotagroup configuration details and retry the command. Refer the create Virtual Server commands section of the IBM Documentation.

- **HVS-VSCR004: Create container failed due to wrong port configuration**
  Explanation: Create Virtual Server instance failed as an invalid port configuration is provided.
  System Action: Create Virtual Server command execution fails.
  User Action: Ensure you pass a valid port configuration details and retry the command. Refer the create Virtual Server commands section of the IBM Documentation.

- **HVS-VSCR005: Create container failed due to error in reading json environment details from file**
  Explanation: Create Virtual Server instance failed as there is an internal error in reading the json environment file.
  System Action: Create Virtual Server command execution fails.
  User Action: Ensure that the json environment file is present and properly constructed. The json environment file is one of the input parameters to the command. The json environment file supports pre-defined set of input values to the create virtual server command. Refer to the product documentation for more details on the json environment file. If the problem still persists, obtain an appliance and LMS dump, and contact IBM Support.

- **HVS-VSCR006: Create container failed due to error in parsing environment details**
  Explanation: Create Virtual Server instance failed as there is an error in the parsing json environment file.
  System Action: Create Virtual Server command execution fails.
  User Action: Ensure that the json environment file is present and properly constructed. The json environment file is one of the input parameters to the command. The json environment file supports pre-defined set of input values to the create virtual server command. Refer to the product documentation for more details on the json environment file. If the problem still persists, obtain an appliance and LMS dump, and contact IBM Support.

- **HVS-VSCR007: Create virtual server failed due to error in reading %s provided in env json file. Please recheck values in env json file**
  Explanation: Create Virtual Server instance failed as there an error while reading the file.
  System Action: Create Virtual Server command execution fails.
  User Action: Ensure that the file is present and properly constructed. If the problem still persists, obtain an appliance and LMS dump contact IBM Support.

- **HVS-VSCR008: Create virtual server failed due to error in parsing details**
  Explanation: Create virtual server instance failed as there an error while reading parameters.
  System Action: Create virtual server command execution fails.
  User action: Ensure all the parameters are properly specified. If the problem still persists, obtain an appliance and LMS dump contact IBM Support.

- **HVS-VSCR009: Create virtual server virtual server failed. Details: %s. Provide valid passthrough quotagroup**
  Explanation: Create virtual server instance failed as there an error as the provided quotagroup is not passthrough.
  System Action: Create virtual server command execution fails.
  User action: Ensure that a valid passthrough quotagroup is provided and retry the command. Refer the create Virtual Server commands section of the IBM Documentation.

- **HVS-VSCR010: Create virtual server failed due error in quotagroup size or unit. Details: %s. Please provide valid quotagroup size and unit**
  Explanation: Create virtual server instance failed as an invalid quotagroup size is provided.
  System Action: Create virtual server command execution fails.
  User action: Ensure that the a valid quotagroup size and unit is provided and retry the command. Refer the create Virtual Server commands section of the IBM Documentation.

- **HVS-VSCR011: Create virtual server failed to validate %s. Provide name of length 2 to 254 characters and can have `"_"`, `"."`, `"-"`, as special characters**
  Explanation: Create virtual server instance failed as an invalid virtual server name is provided.
  System Action: Create virtual server command execution fails.
  User action: Ensure that the a valid quotagroup size and unit is provided and retry the command.Refer the create Virtual Server commands section of the IBM Documentation.

- **HVS-VSCR012: Create virtual server failed retrieving %s Please enter a valid container name**
  Explanation: Create virtual server instance failed as an invalid virtual server name is provided.
  System Action: Create virtual server command execution fails.
  User action: Provide name of length 2 to 254 characters and can have `"_"`, `"."`, `"-"`, as special characters. Retry the command. Refer the create Virtual Server commands section of the IBM Documentation.

- **HVS-VSCR013: Create Virtual Server failed due to internal server issues**
  Explanation: Virtual Server create operation failed due to internal server issues.
  System Action: Create virtual server command execution fails.
  User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

- **HVS-VSCR014: Create Virtual Server failed due to wrong network configuration. %s. Refer documentation**
  Explanation: Virtual Server create operation failed because of wrong network configuration.
  System Action: Create virtual server command execution fails.
  User Action: Ensure that valid network details are provided and try again.

- **HVS-VSCR015: Create Virtual Server failed due to wrong quotagroup configuration. Refer documentation**
  Explanation: Virtual Server create operation failed due to wrong quotagroup configuration.
  System Action: Create virtual server command execution fails.
  User Action: Ensure that valid quotagroup details are provided and try again.

- **HVS-VSCR016: Create Virtual Server failed due to wrong port configuration. Refer documentation**
  Explanation: Virtual Server create operation failed due to wrong port configuration.
  System Action: Create virtual server command execution fails.
  User Action: Ensure that valid port configuration details are provided and try again.

- **HVS-VSCR017: Create Virtual Server failed due to error in reading environment details from file. Refer documentation**

Explanation: Virtual Server create operation failed due to error in reading environment details from file.
System Action: Create virtual server command execution fails.
User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

- **HVS-VSCR018: Create Virtual Server failed due to error in parsing environment details**
  Explanation: Virtual Server create operation failed due to error in parsing environment details.
  System Action: Create virtual server command execution fails.
  User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

- **HVS-VSCR019: Create Virtual Server failed due to error in reading file. Refer documentation**
  Explanation: Virtual Server create operation failed due to error in reading file.
  System Action: Create virtual server command execution fails.
  User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

- **HVS-VSCR020: Create Virtual Server failed due to error in parsing Virtual Server details. Provide valid details and retry**
  Explanation: Virtual Server create operation failed due to error in parsing Virtual Server details.
  System Action: Create virtual server command execution fails.
  User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

- **HVS-VSCR021: Create Virtual Server failed because quotagroup does not exist**
  Explanation: Virtual Server create operation failed because quotagroup does not exist.
  System Action: Create virtual server command execution fails.
  User Action: Ensure that valid quotagroup details are provided and try again.

- **HVS-VSCR022:: Create Virtual Server failed due error in volume size**
  Explanation: Virtual Server create operation failed because of error in volume size.
  System Action: Create virtual server command execution fails.
  User Action: Ensure that valid volume details are provided and try again.

- **HVS-VSCR023: Create Virtual Server failed to validate Virtual Server name. Please provide name of length 2 to 254 characters and can have `_`, `.`, `-` as special characters**
  Explanation: Virtual Server create operation failed because virtual server name is invalid.
  System Action: Create virtual server command execution fails.
  User Action: Ensure that valid virtual server name is provided and try again.

- **HVS-VSCR024: Create Virtual Server failed to retrieve Virtual Server name**
  Explanation: Virtual Server create operation failed to retrieve virtual server name.
  System Action: Create virtual server command execution fails.
  User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

- **HVS-VSCR025: Virtual Server Create failed due to wrong label configuration. Error: %s. Provide valid label configuration**
  Explanation: Virtual Server create operation failed due to wrong label configuration.
  System Action: Create virtual server command execution fails.
  User Action: Ensure that valid label configuration details are provided and try again

- **HVS-VSCR026: Create virtual server failed because invalid reset_root value is provided. Supported values are - true and false**
  Explanation: Virtual Server create operation failed because invalid reset_root value is provided. Supported values are - true and false.
  System Action: Create virtual server command execution fails.
  User Action: Ensure that valid reset_root value is provided and try again.

- **HVS-VSCR027: Create virtual server failed because the mount_id corresponding to reset_root parameter was not found in RUNQ_ROOTDISK parameter**
  Explanation: Virtual Server create operation failed because the mount_id corresponding to reset_root parameter was not found in RUNQ_ROOTDISK parameter.
  System Action: Create virtual server command execution fails.
  User Action: Ensure that valid reset_root and RUNQ_ROOTDISK values are provided and try again.

- **HVS-VSCR028: Create virtual server failed beacuse the RUNQ_ROOTDISK parameter is not set. It is mandatory to provide RUNQ_ROOTDISK in env flag when using reset_root parameter**
  Explanation: Virtual Server create operation failed because the RUNQ_ROOTDISK parameter is not set. It is mandatory to provide RUNQ_ROOTDISK in env flag when using reset_root parameter.
  System Action: Create virtual server command execution fails.
  User Action: Ensure that valid reset_root and RUNQ_ROOTDISK values are provided and try again.

HVS-VSUD00x: Messages for update Virtual Server command

This section lists the messages you might encounter while running the update container command.

- **HVS-VSUD001: Update virtual server failed due to server issues**
  Explanation: An internal server error occurred, resulting in failure of Update Virtual Server (also referred as container) command failed.
  System Action: Update Virtual Server command execution fails.
  User Action: Obtain an appliance and LMS dump and contact IBM Support.

- **HVS-VSUD002: Update virtual server failed due to wrong network configuration. Provide valid network**
  Explanation: Update Virtual Server instance failed as an invalid network configuration is provided.
  System Action: Update Virtual Server command execution fails.
  User Action: Provide valid network configuration details and retry the command. Refer the update Virtual Server commands section of the IBM Documentation.

- **HVS-VSUD003: Update virtual server failed due to wrong quotagroup configuration. Invalid parameter in quotagroup configuration. Provide valid quotagroup configuration.**
  Explanation: Update virtual server instance failed as an invalid quotagroup configuration is provided.
  System Action: Update virtual server command execution fails.
  User Action: Provide supported parameters and retry operation. **Note**: The parameter **`reset_root`** flag is not supported in IBM Hyper Protect Virtual Servers version 1.2.2, for updating a virtual server by using the **`hpvs vs update`** command. You can use the **`hpvs deploy -u`** command to update the virtual server with the parameter **`reset_root:true`**.

- **HVS-VSUD004:: Update virtual server failed due to wrong port configuration. Provide port value in the range 1-65535**
  Explanation: Update virtual server instance failed as an invalid port configuration is provided.
  System Action: Update virtual server command execution fails.
  User Action: Provide valid port configuration details and retry the command. Refer the update Virtual Server commands section of the IBM Documentation.

- **HVS-VSUD005: Update virtual server failed due to error in reading json environment details from file**
  Explanation: Update virtual server instance failed as there is an internal error in reading json environment file.
  System Action: Update virtual server command execution fails.
  User Action: Ensure that the json environment file is present and properly constructed. The json environment file is one of the input parameters to the command. The json environment file supports pre-defined set of input values to the Update virtual server command. Refer to the product documentation for more details on the json environment file. If the problem persists, obtain an appliance and LMS dump, and contact IBM Support.

- **HVS-VSUD006: Update virtual server failed due to error in parsing environment details. Provide valid environment json file**
  Explanation: Update virtual server instance failed as there is an error parsing json environment file.

System Action: Update virtual server command execution fails.
User Action: Ensure that the json environment file is present and properly constructed. The json environment file is one of the input parameters to the command. The json environment file supports pre-defined set of input values to the Update virtual server command. Refer to the product documentation for more details on the json environment file. If the problem persists, obtain an appliance and LMS dump, and contact IBM Support.

- **HVS-VSUD007: Update server failed due to error in reading %s provided in env json file. Recheck values in env json file**
  Explanation: Update virtual server instance failed as there an error while reading the env json file.
  System Action: Update virtual server command execution fails.
  User Action: Ensure that the file location specified in env json path exists and is properly specified. If the problem persists, obtain an appliance and LMS dump, and contact IBM Support.

- **HVS-VSUD008: Update virtual server failed due to error in parsing details**
  Explanation: Update virtual server instance failed as there an error in parsing details.
  System Action: Update virtual server command execution fails.
  User Action: Ensure that all the parsed values for command execution are proper. If the problem persists, obtain an appliance and LMS dump, and contact IBM Support.

- **HVS-VSUD009: Update virtual server failed. Details: %s. Provide valid passthrough quotagroup**
  Explanation: Update virtual server instance failed as there an error as the provided quotagroup is not passthrough.
  System Action: Update virtual server command execution fails.
  User Action: Ensure that a valid passthrough quotagroup is provided and retry the command. Refer the update Virtual Server commands section of the IBM Documentation.

- **HVS-VSUD010: Update virtual server failed due error in quotagroup size or unit. Details: %s. Provide valid quotagroup size and unit**
  Explanation: Update virtual server instance failed as an invalid quotagroup size is provided.
  System Action: Update virtual server command execution fails.
  User Action: Ensure that the a valid quotagroup size and unit is provided and retry the command. Refer the update Virtual Server commands section of the IBM Documentation.

- **HVS-VSUD011: Update virtual server failed. The updated virtual server name did not match the original virtual server name. Provide valid virtual server name and try again**
  Explanation: Update virtual server instance failed as virtual server name did not match the original virtual server name.
  System Action: Update virtual server command execution fails.
  User Action: Ensure that the a valid virtual server name is provided and retry the command. Refer to the product documentation for more details on the specific details of this command.

- **HVS-VSUD012: Update virtual server failed. The repository of the old virtual server is not in the list of upgradeable repositories for the new virtual server repository. Allowed are %s. Provide an allowed repository and try again**
  Explanation: Update virtual server instance failed as the repository of the old virtual server is not in the list of upgradeable repositories for the new virtual server repository.
  System Action: Update virtual server command execution fails.
  User Action: Ensure that the valid repository details are provided and retry the command. Refer to the product documentation for more details on the specific details of this command.

- **HVS-VSUD013: Update virtual server failed because virtual server crypto device allocation update is not allowed**
  Explanation: Update virtual server instance failed because virtual server crypto device allocation update is not allowed.
  System Action: Update virtual server command execution fails.
  User Action: Ensure that the valid crypto details are provided and retry the command. Refer to the product documentation for more details on the specific details of this command.

- **HVS-VSUD014: Update virtual server failed due to wrong label configuration. Error: %s. Provide valid label configuration**
  Explanation: Virtual Server update operation failed due to wrong label configuration.
  System Action: Update virtual server command execution fails.
  User Action: Ensure that valid label configuration details are provided and try again.

- **HVS-VSUD015: Update virtual server failed because invalid reset_root value is provided. Supported values are - true and false**
  Explanation: Virtual Server update operation failed because invalid reset_root value is provided. Supported values are - true and false.
  System Action: Update virtual server command execution fails.
  User Action:Ensure that a valid reset_root value is provided and try again.

- **HVS-VSUD016: Update virtual server failed because the mount_id corresponding to reset_root parameter was not found in RUNQ_ROOTDISK parameter**
  Explanation: Virtual Server update operation failed because the mount_id corresponding to reset_root parameter was not found in RUNQ_ROOTDISK parameter.
  System Action: Update virtual server command execution fails.
  User Action: Ensure that valid reset_root and RUNQ_ROOTDISK values are provided and try again.

- **HVS-VSUD017: Update virtual server failed because the RUNQ_ROOTDISK parameter is not set**
  Explanation: Virtual Server update operation failed because the RUNQ_ROOTDISK parameter is not set. It is mandatory to provide RUNQ_ROOTDISK in the env flag when using reset_root parameter.
  System Action: Update virtual server command execution fails.
  User Action: Ensure that valid reset_root and RUNQ_ROOTDISK values are provided and try again.

- **HVS-VSUD018: Update virtual server failed because detach or attach quotgaroup is not allowed**

- **Explanation**: Update virtual server failed because detach or attach quotgaroup is not allowed. Provide proper values of quotagroup given during create.
  System Action: Update virtual server command execution fails.
  User Action: Ensure that valid quotagroup values are provided and try again.

HVS-VSLI00x: Messages for list Virtual Server command

This section lists the messages you might encounter while running the list Virtual Server command.

- **HVS-VSLI001: The command to list virtual server failed due to an internal server error**
  Explanation: An internal server error occurred, resulting in failure of list virtual server command.
  System Action: List virtual server command execution fails.
  User Action: Obtain an appliance and LMS dump, and contact IBM Support.

HVS-VSDE00x: Messages for delete Virtual Server command

This section lists the messages you might encounter while running the delete Virtual Server command.

- **HVS-VSDE001: The command to delete virtual server failed due to an internal server error**
  Explanation: An internal server error occurred, resulting in failure of delete virtual server command.
  System Action: Delete virtual server command execution fails.
  User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

- **HVS-VSDE002: Delete Virtual Server for %s failed because snapshot is present. Delete snapshot before deleting virtual server**
  Explanation: Virtual Server delete operation failed because snapshot is present for the virtual server.
  System Action: Delete virtual server command execution fails.
  User Action: Ensure that valid virtual server details are provided and try again.

- **HVS-VSDE003: Virtual Server Delete failed. The configuration for the virtual server is not valid: %s. Provide valid configuration values and try again**
  Explanation: Virtual Server delete operation failed because the configuration for the virtual server is not valid.
  System Action: Delete virtual server command execution fails.
  User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

- **HVS-VSDE004: Virtual Server delete failed. Could not remove virtual server %s**
  Explanation: Virtual Server delete operation failed because there was an error while trying to remove the virtual server.
  System Action: Delete virtual server command execution fails.
  User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

- **HVS-VSDE005: Virtual Server delete failed to remove directory for virtual server %s**
  Explanation: Virtual Server delete operation failed because there was an error while trying to remove while trying to remove the virtual server directory.
  System Action: Delete virtual server command execution fails.
  User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

- **HVS-VSDE006: Virtual Server delete failed because passthrough quotagroup of container %s cannot be removed**
  Explanation: Virtual Server delete operation failed because passthrough quotagroup of container %s cannot be removed.
  System Action: Delete virtual server command execution fails.
  User Action: Ensure that valid virtual server details are provided and try again.

- **HVS-VSDE007: Delete subvolume for %s failed. Obtain an appliance dump and contact IBM Support**
  Explanation: Virtual Server delete operation failed because there was an error while trying to delete subvolume.
  System Action: Delete virtual server command execution fails.
  User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

HVS-VSSW00x: Messages for show virtual server command

This section lists the messages you might encounter while running the show virtual server command.

- **HVS-VSSW001: The command to show virtual server failed due to an internal server error**
  Explanation: An internal server error occurred, resulting in failure of show virtual server command.
  System Action: Show virtual server command execution fails.
  User Action: Obtain an appliance and LMS dump, and contact IBM Support.

HVS-VSLO00x: Messages for log virtual server command

This section lists the messages you might encounter while running the delete virtual server command.

- **HVS-VSLO001: The command to log virtual server failed due to an internal server error**
  Explanation: An internal server error occurred, resulting in failure of log virtual server command.
  System Action: Show virtual server command execution fails.
  User action: Obtain an appliance and LMS dump, and contact IBM Support.

HVS-VSYY00x: Messages for some virtual server commands

This section lists the messages you might encounter while running some virtual server commands.

- **HVS-VSYY001: Virtual Server %s was not found on the system. Specify a virtual server that exists on the system**

Explanation: Virtual Server operation failed because virtual server was not found.
System Action: Virtual Server command execution fails.
User Action: Ensure that valid virtual server details are provided and try again.

- **HVS-VSYY002: Error invoking script on host. Shell returned rc=%s: %s. Obtain an appliance dump and contact IBM Support**
Explanation: Virtual Server operation failed because there was an error while invoking script.
System Action: Virtual Server command execution fails.
User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

- **HVS-VSYY003: Error occurred while writing script: %s. Obtain an appliance dump and contact IBM Support**
Explanation: Virtual Server operation failed because there was an error while writing script.
System Action: Virtual Server command execution fails.
User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

- **HVS-VSYY004: Error occurred while changing permissions of script %s: %s. Obtain an appliance dump and contact IBM Support**
Explanation: Virtual Server operation failed because there was an error while changing permissions of script.
System Action: Virtual Server command execution fails.
User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

- **HVS-VSYY005: Error occurred while copying tmp script %s to %s: %s. Obtain an appliance dump and contact IBM Support**
Explanation: Virtual Server operation failed because there was an error while copying the script.
System Action: Virtual Server command execution fails.
User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

- **HVS-VSYY006: Error occurred while executing script. rc=%s: stdout=%s stderr=%s. Obtain an appliance dump and contact IBM Support**
Explanation: Virtual Server operation failed because there was an error while executing the script.
System Action: Virtual Server command execution fails.
User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

- **HVS-VSYY007: Found empty request body. Provide valid details and try again**
Explanation: Virtual Server operation failed because the request body was empty.
System Action: Virtual Server command execution fails.
User Action: Ensure that valid virtual server details are provided and try again.

- **HVS-VSYY00: The parameter 'network_name' is required when 'networks' is specified. Provide valid details and try again**
Explanation: Virtual Server operation failed because the parameter 'network_name' is required when 'networks' is specified.
System Action: Virtual Server command execution fails.
User Action: Ensure that valid network_name details are provided and try again.

- **HVS-VSYY009: The container %s is not in a running state to process this request. Provide valid details and try again**
Explanation: Virtual Server operation failed because the container is not in a running state.
System Action: Virtual Server command execution fails.
User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

- **HVS-VSYY010: Failed to read from sysfs, node %s. Obtain an appliance dump and contact IBM Support**
  Explanation: Virtual Server operation failed to read from sysfs.
  System Action: Virtual Server command execution fails.
  User Action: Retry the command. If the problem persists, obtain the appliance dump, LMS logs, and contact IBM Support.

- **HVS-VSYY011: Virtual Server %s attached to locked quotagroup, operation not allowed. Obtain an appliance dump and contact IBM Support**
  Explanation: Virtual Server operation failed because the virtual server is attached to locked quotagroup.
  System Action: Virtual Server command execution fails.
  User Action: Ensure that valid virtual server details are provided and try again.

- **HVS-VSYY012: Virtual Server configuration requires a quotagroup that is locked. Create operation is not allowed. Obtain an appliance dump and contact IBM Support**
  Explanation: Virtual Server operation failed because it requires a quotagroup that is locked.
  System Action: Virtual Server command execution fails.
  User Action: Ensure that valid virtual server details are provided and try again.

- **HVS-VSYY013: The host network cannot be used except by adding it to the repository definition file. Obtain an appliance dump and contact IBM Support**
  Explanation: Virtual Server operation failed because the host network cannot be used except by adding it to the repository definition file.
  System Action: Virtual Server command execution fails.
  User Action: Ensure that valid virtual server details are provided and try again.

- **HVS-VSYY014: The operation failed because invalid value: %s = %s was found. Provide valid details and try again**
  Explanation: Virtual Server operation failed because invalid value: %s = %s was found.
  System Action: Virtual Server command execution fails.
  User Action: Ensure that valid virtual server details are provided and try again.

- **HVS-VSYY015: The operation failed because there is not enough free memory (%s MB) to support the memory required by the container (%s MB)**
  Explanation: Virtual Server operation failed because there is not enough free memory (%s MB) to support the memory required by the container (%s MB).
  System Action: Virtual Server command execution fails.
  User Action: Ensure that valid virtual server details are provided and try again.

Terminology

The following list explains the terms that might be used in this documentation.

- Appliance

  IBM Secure Service Container based appliance provided by an Appliance Vendor. From Hosting Appliance perspective, it is the combination of IBM Secure Service Container and Hosting Appliance.

- Appliance Administrator

  The person administrating an appliance which includes tasks, such as configuring storage, or memory to the appliance or performing other configuration tasks through the API provided by Secure Service Container or the Hosting Appliance.

- Appliance Operational Data

Metrics, logs, appliance dump data, error logs, stack traces, kernel dump, etc.

- Appliance Protected Data

  Appliance secrets, workload data, configuration data, settings, and other internal information stored by an appliance.

- Appliance Vendor

  An internal, or external exploiter of Secure Service Container, packaging Secure Service Container into an appliance.

- BYOK

  The abbreviation of Bring Your Own Key, which allows you to import your existing keys to Hyper Protect Crypto Services service instances that protect your keys with advanced encryption.

- BYOI

  The abbreviation of **Bring Your Own Image**, which is a part of IBM Hyper Protect Virtual Servers solution to support the development and deployment of your own container images on top of the Secure Service Container framework.

- Container

  A runtime instance of an Open Container Image (OCI) compatible image.

- Datapool

  Synonyms for **Storage Pool**.

- EP11

  Enterprise PKCS #11 (EP11) is specifically designed for customers seeking support for open standards and enhanced security. The EP11 library provides an interface very similar to the industry-standard PKCS #11 API.

- GPG

  The abbreviation of Gnu Privacy Guard, which is an open standard used for signing, encrypting, and decrypting texts with public and private keys to increase the security of communications.

- GREP11

  GREP11 represents the Enterprise PKCS #11 (EP11) APIs over gRPC calls, which is designed to be a stateless interface for cryptographic operations on cloud.

- gRPC

  A modern open source high performance remote procedure call (RPC) framework that can connect services in and across data centers for load balancing, tracing, health checking, and authentication.

- Hardware security module

  A physical appliance that provides on-demand encryption, key management, and key storage as a managed service.

- Hosting appliance

  A technical component within IBM Secure Service Container based appliances, providing the enablement for running Docker-based workloads.

- Hyper Protect hosting appliance

  An enhanced version of IBM Secure Service Container software appliance.

- Image

  Images are the basis of the containers. An image is an ordered collection of root file system changes and the corresponding execution parameters for use within a container runtime.

- ISV

  The abbreviation of **Independent Software Vendor**, who provides software solutions by developing and deploying containerized applications to the Secure Service Container partitions.

- Management server

  An x86 or Linux on IBM Z or LinuxONE (i.e., s390x architecture) management server used to run the commands provided by IBM Hyper Protect Virtual Servers , and administer the offering.

- Manifest

  A manifest is generated by the Secure Build for audit purpose, which contains a copy of the github project cloned by the Secure Build container, a copy of the build log, and a **`build.json`** with the build status.

- Manifest public key

  A manifest public key is used to verify the manifest generated by the Secure Build.

- Manifest private key

  A manifest private key is used to sign the manifest during the Secure Build.

- Namespace

  A namespace such as **`ibmzcontainers`**that contains a number of unique images. For examples, the images include **`hpvsop-base`**, **`hyperpcons-worker`**, **`hyperpcons-riaas`**, and so on.

- Partition

  A partition is the logic partition (LPAR) on the mainframe, and can be created by using the logic partitioning tools such as Hardware Management Console (HMC) or other logical partitioning tools.

- PKCS #11

  The abbreviation of Public-Key Cryptography Standards #11, which defines a platform-independent API to cryptographic tokens, such as HSM and smart cards.

- Quotagroup

  The storage assigned to a workload running on an appliance. The appliance administrator assigns FCP, or ECKD based storage to an appliance, and then creates quotagroups, representing parts of the underlying storage. The administrator finally assigns quotagroups to workloads through the appliance API.

- Registry

  A Registry is a hosted service containing repositories of container images that responds to the Registry API. For example, Docker Hub.

- Repository registration files

  A cleartext Python or JSON format file, which is generated by the Secure Build container when the container is created. The JSON format repository registration file can be used as the direct input to generate an encrypted repository definition file.

- Repository definition files

  An encrypted registration file or a repository definition file is used to register the repository, for authentication or validation reasons, such that a Hosting Appliance will trust that the image, when pulled from the registry, is authentic.

- Repository

  A repository is a set of containerized images. A repository can be shared by pushing it to a registry server. Different images in the repository can be labeled using tags. For example, **`hpvsop-base`**.

- runc

  A CLI tool for spawning and running containers according to the Open Container Initiative (OCI) specification.

- runq

  An open-sourced hypervisor-based Docker runtime environment, which is based on runc to run regular containerized images in a lightweight KVM or Qemu virtual machine.

- s390x

  The underlying architecture of IBM Z or LinuxONE mainframe.

- Secure Build

  The process of building the application code from a Git-like source repository into a container image for s390x architecture, signing the image by using the authentication keys, and publishing the image to the remote repository for later integration.

- Secure Service Container

  A container framework based on the runq technology, that is supported by the IBM Z or LinuxONE servers.

- Secure Service Container partition

  A type of logic partitions (LPAR) on the mainframe that runs the Secure Service Container framework.

- SSH

  The abbreviation of Secure Shell, which is a cryptographic network protocol for operating network services securely over an unsecured network by using public and private keys.

- Storage Pool

  A storage pool is a uniquely named collection of storage disks on which the appliance file system is mounted.

- System Administrator

  This role includes the system administrator of a machine, storage administrators, and network administrators.

- tag

  A tag is used to version images in a repository. For example, **`latest`**, **`1.2.3.4-develop-a0d3aea`**, or **`s390x-develop-54a9045`**.

- Workload

  The application and data provided and generated by a (running) Workload Image.

- Workload Data

  Workload user or workload client data, workload logs, workload secrets stored in the appliance.

- Workload Image

  A container-based image, provided by the Workload Vendor. An appliance only runs workload images which have been registered with the appliance through a repository definition file.

- Workload User

  The end user of a workload.

- Workload Vendor

  The creator of a Docker image running on top of Hosting Appliance.