Integrating IBM Cloud Private (ICP) and IBM Cloud Automation Manager (CAM)

Delivering multi-cloud environment with containers and virtual machines

Stuart Cunliffe – IBM System Group Lab Services
@StuCunliffe
Agenda

- High-level IBM Cloud Private review
- Where does IBM Cloud Automation Manager fit?
- Installing IBM CAM
- Connecting to multiple cloud environments
- IBM CAM templates
  - Demo 1
- IBM CAM services
  - Demo 2
  - Demo 3
IBM Cloud Private - review
IBM Cloud Private is a private cloud platform for developing and running containerized workloads locally on premise.

IBM Cloud Private is an application platform for developing and managing on-premises, containerized applications.

It is an integrated environment for managing containers that includes the container orchestrator Kubernetes, a private image repository, a management console, and monitoring frameworks.

It includes a graphical user interface which provides a centralized location from where you can deploy, manage, monitor, and scale your applications.
Docker performs operating system level virtualization also known as containerization.

A container image is a lightweight, stand-alone, executable package of a piece of software that includes everything needed to run it: code, runtime, system tools, system libraries, settings.

Docker simplifies dependency matrix, Docker will ensure they are cross environments through docker image.

Docker is ready for PaaS Cloud, as it eliminates the traditional one environment at a time develop, deploy, update and migrate.

Docker is available in two editions
  • Community Edition (CE)
  • Enterprise Edition (EE)
Kubernetes

- Kubernetes is a portable, extensible open-source platform for managing containerized workloads and services.

- It is a **container orchestrator**, which **runs and manages containers**.

- **Manages Application** and not machines

- Rich **ecosystem of plug-ins** for scheduling, storage and networking,

- It provides much of the **simplicity of Platform as a Service (PaaS)** with the **flexibility of Infrastructure as a Service (IaaS)**, and enables **portability across infrastructure providers**.

- Some features like **intelligent scheduling**, **self healing**, **horizontal scaling**, **service discovery**, **automated rollouts and rollback** and **configuration management**
Kubernetes

- **Pods**
  - It is **unit of execution** of Kubernetes.
  - **Collection of containers** that can share storage and network resources.
  - It contains **one or more application containers** which are relatively tightly coupled.

- **Service**
  - Defined a set of pods and a means by which to access them, such as single stable IP address and corresponding DNS name.

- **Helm Charts**
  - A chart is a collection of files that describe a related set of Kubernetes resources. This can be simple deployments with single pods or complex full stack deployments.
ICP – Node Classes

- ICP has five classes of nodes
  - **Boot Node**
    - Aka bootstrap node
    - Used for *running installation, configuration*, node scaling and cluster updates.
    - You can use a single node for both master and boot.
  - **Master Node**
    - Provides *management services and controls the worker nodes* in the cluster.
    - Host processes that are responsible for resource allocation, state maintenance, scheduling and monitoring.
  - **Worker Node**
    - Provides a *containerized environment for running tasks*.
    - More worker nodes can be added to improve performance and efficiency.
  - **Proxy Node**
    - Transmits *external request to the service* created inside your cluster.
    - Multiple proxy nodes can be deployed in a high availability environment.
  - **Management Node (Optional Node)**
    - Only hosts management services like *monitoring, metering and logging*.
    - Can prevent the master node from becoming overloaded.
### ICP – Node Classes

- **ICP supported platform by node type**

<table>
<thead>
<tr>
<th>Node type</th>
<th>Linux 64-bit (x86_84)</th>
<th>Linux on POWER 64-bit Little Endian (LE) (ppc64le)</th>
<th>IBM Z (s390x)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boot</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>Master</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>Management</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>Proxy</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>Worker</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>
ICP – London environment

Power Servers:

- Lancashire: Master & Boot Node
- Merseyside: Worker Node 1
- Yorkshire: Worker Node 2
IBM Cloud Automation Manager
IBM PowerVC - London environment

Power Servers:

Lancashire
- Master & Boot Node
- AIX Oracle VM

Merseyside
- Worker Node 1
- RHEL Mongo VM

Yorkshire
- Worker Node 2
- SLES App VM
Multi-cloud environment

IBM Cloud

Amazon Web Services

IBM PowerVC
Virtual Machines, LPARs

vSphere
Virtual Machines

IBM Power Servers

X86 Servers

IBM Cloud Private
Containerized applications

off-premise

on-premise
CAM Overview

What is CAM?

Marketing

IBM® Cloud Automation Manager is a multi-cloud, self-service management platform running on IBM Cloud Private that empowers developers and administrators to meet business demands.

Technical

A container based application running on ICP that can be configured to orchestrate VM/Service deployments into one or more clouds (private and/or public).
Cloud Automation Manager in IBM Cloud Private

- Containerized cloud native application
- Installed into an IBM Cloud Private worker node with a Helm chart
- Leverages IBM Cloud Private services for enterprise capabilities
IBM Cloud Automation Manager
Full stack automation and service orchestration

- **Automated provisioning** – Automated provisioning of infrastructure and applications with workflow orchestration
- **Self-service** - Self-service access to cloud infrastructure and application services
- **Manage and govern** – Manage and govern workloads across multiple and hybrid clouds
- **Built with open technology** to avoid vendor lock-in
CAM Overview

Key Features

• Simple installation via a provided ICP Helm Chart

• Connects to multiple types of clouds via its embedded use of Terraform
  - OpenStack (PowerVC), Google, AWS, Azure .. and many more

• Allows customization of templates, inputs, and outputs
  - VMs, APIs, application content, emails
  - Internal or Github/Gitlab repositories

• Build a “Service” workflow that orchestrates consumption of multiple clouds and templates
  - ex. deploy Websphere VM in IBM Cloud and a MongoDB in PowerVC in one step

• Allow publishing of “Service” to the ICP Catalog for deployment within ICP GUI
  - service can also be used by other consumers such as ServiceNow
Why Terraform

Declarative Cloud Automation

Common approach in all clouds
• All clouds and all application architectures

Open source
• 10,000+ GitHub stars; 1,153 contributors
• Supported by major cloud vendors

Large & growing ecosystem
• Terraform Module Registry
• Many OEM providers and provisioners

Enterprise hardened by IBM
• Secrets management
• Role based provisioning
• Team development
Terraform Content

Now Available on CAMHub as open source [https://ibm.biz/BdZfLs](https://ibm.biz/BdZfLs)

---

### Open Source Library

- **MEAN stack** – VMware, IBM Cloud, Azure, AWS
- **LAMP stack** – VMware, IBM Cloud, Azure, AWS
- **Node.js** – VMware, IBM Cloud
- **Strongloop** – VMware, IBM Cloud
- **MariaDB** – VMware
- **MongoDB** – VMware, IBM Cloud
- **MongoDB Strongloop 3 tier** – VMware, IBM Cloud
- **Virtual Servers with SSH key** – IBM Cloud, AWS
- **Apache HTTP Server** – VMware, IBM Cloud, AWS
- **Apache Tomcat** – VMware, IBM Cloud, AWS

---

### Enterprise Library

- **IBM DB2 EE** (v10.5 & v11.1)
- **IBM MQ** (v8 & v9)
- **IBM WebSphere Application Server ND** (v8.5.5, v9.0)
- **IBM WebSphere Liberty** (v17)
- **IBM HTTP Server** (v8.5.5, v9)
- **Oracle DB Enterprise** (v12c)
- **Oracle MySQL** (v5.7)

All deployable to VMware, IBM Cloud, AWS

---

Automation content available with IBM Cloud Private purchase. Product licenses must be purchased separately or BYOL. See pricing and packaging for more information.

---

**+Community Templates** (Terraform Module Registry)

**+Bring your own** (Self written, IBM Cloud Schematics, ...)

---

Supported when deployed through CAM with current S&S
CAM installation
CAM Installation

CAM 3.1.2 Planning and Requirements

IBM Cloud Private 3.1.0 or above installed and running

- IBM Cloud Private is certified on KVM, ESX, Nutanix Acropolis, and IBM PowerVM hypervisors

CAM 3.1.2 Helm Chart and Docker Images

- CAM Helm Chart deploys to worker nodes which can be x86 or ppc64le (IBM Z is not supported)
- Community Edition available from via ibm-charts helm repository and docker hub
- Enterprise Edition available from Passport Advantage
CAM Installation
CAM 3.1.2 Planning and Requirements

ICP Persistent Storage
• Requires setup of 4 ICP Persistent Volumes

ICP Storage class can be
• ESS (Spectrum Connect)
• NFS
• GlusterFS
• PowerVC Flex Volume Driver
# CAM Installation

## CAM 3.1.2 Planning and Requirements

<table>
<thead>
<tr>
<th>Persistent storage requirement</th>
<th>Size (GB)</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>cam-mongo-pv</td>
<td>20 GB</td>
<td>Additionally, the size grows to 30G for every 15k Deployments</td>
</tr>
<tr>
<td>cam-logs-pv</td>
<td>10 GB</td>
<td>Static</td>
</tr>
<tr>
<td>cam-terraform-pv</td>
<td>15 GB</td>
<td>Usage can grow or shrink</td>
</tr>
<tr>
<td>cam-bpd-appdata-pv</td>
<td>20 GB</td>
<td>The size grows based on the number of templates in local repository</td>
</tr>
</tbody>
</table>
CAM Installation

CAM 3.1.2 Installation Steps

Login to the ICP GUI as administrator and deploy the ibm-cam Helm chart
CAM Configuration & Use
CAM Configuration & Use

- Cloud Connections
  - Private Clouds (like PowerVC, OpenStack, vmWare, etc)
  - Public Clouds (like AWS, Azure, IBM, etc)

- CAM Templates
  - Deployment specifications

- CAM Services
  - Plans, Workflows, Exposure of select variables
CAM Configuration & Use

Create Cloud Connection for PowerVC

Select Cloud Provider:

A large variety of clouds are available

Choose “OpenStack” for PowerVC
CAM Configuration & Use

Create Cloud Connection for PowerVC

Select Namespace:
Global allows use in all namespaces

Provide Connection Name and Description
CAM Configuration & Use

Create Cloud Connection

Obtain connection values from the powervcrc file

Authorization URL:

Username: <powervc admin>

Domain Name: Default

Region: RegionOne

Project Name: ibm-default (or any custom project)
CAM Configuration & Use

Create Cloud Connection

- Network Connection will be validated during save
- Menu option to re-test connection at any later time
CAM Templates and Services

You create and configure Templates and Services from CAM UI

• A **CAM Template** is source code and input/output variables that define a terraform deployment

• A CAM Template can be deployed directly from CAM GUI and/or used as a component within a “Service”

• A **CAM Service** defines a workflow that consumes one or more CAM templates.

• A CAM Service can be deployed from CAM GUI and/or published to the ICP Catalog to allow deployment from ICP
CAM Template Configuration & Use

Demonstration 1:

Create and deploy a simple AIX VM template
Create a new version/release of the template
Modify existing deployment (plan and apply)
CAM Service Configuration & Use

Demonstration 2:

Create a multi-cloud environment

Deploy multi-cloud environment across containers within ICP, Linux VMs in public cloud and AIX VM in private cloud
Demonstration 3:

Create a service with a decision tree

Deploy decision tree service to a chosen cloud provider

Publish the service to ICP and deploy from there
Multi-cloud environment

- Github
- Amazon Web Services
- IBM Cloud
- IBM Cloud Automation Manager

IBM Cloud Private
- Containerized applications

IBM PowerVC
- Virtual Machines, LPARs

vSphere
- Virtual Machines

IBM Power Servers

X86 Servers

Off-premise
On-premise
CAM Template Configuration & Use

Provided Templates

- Templates are provided by CAM for all kinds of clouds
- These can be used as is or as a starting point for your own template
- OpenStack -> IBM PowerVC Single Virtual Machine Example will be used for illustration
CAM Template Configuration & Use

Deploy PowerVC Single VM Example Template

- Note this example template is maintained on a GIT hub site
- You could create your own GIT hub repository for your own custom templates
CAM Template Configuration & Use

Deploy PowerVC Single VM Example Template

- Fill in Deployment Parameters using the Openstack information from your specific PowerVC resources:
  - Openstack Image ID
  - Openstack Flavor ID
CAM Template Configuration & Use

Deploy PowerVC Single VM Example Template

- Fill in Deployment Parameters using the Openstack ID information from your specific PowerVC resources:
  - Openstack Network Name
  - Openstack Image User ID
  - Openstack Image Password
CAM Template Configuration & Use
Deploy PowerVC Single VM Example Template

- Deployment shows in-progress
- Follows the Terraform process:
  - Plan
  - Apply
  - Modify
- This process allows for post error modification of a parameter and retry
CAM Template Configuration & Use

Deploy PowerVC Single VM Example Template

Log File:
- Shows Terraform Parameters
- Shows Status during deploy
- Shows Resultant State
- Shows Terraform error messages if “apply” fails
CAM Template Configuration & Use
Deploy PowerVC Single VM Example Template

PowerVC deployment will show a hard coded VM name from the example template.

Note: Modifications we have made to this example template are available at end of this presentation. Modifications were to allow input of VM name, Optional IP assignment, Optional cloud-init activation input, and an appvg disk size.
CAM Template Configuration & Use

Deploy PowerVC Single VM Example Template

<table>
<thead>
<tr>
<th>Name</th>
<th>Console</th>
<th>IP Address</th>
<th>Created</th>
<th>State</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>openstack compute_instance_v2.sin..</td>
<td></td>
<td>10.160.147.17</td>
<td>01/25/2019 8:42AM</td>
<td>Running</td>
<td></td>
</tr>
</tbody>
</table>

**Lifecyle Activities include:**

- Details (allows modifications)
- Stop/Start
- Taint (forces delete and redeploy of VM upon next change)
CAM Template Configuration & Use

Removal of the deployed instance is done in two steps

1. Destroy Resource. This deletes the deployed VM in PowerVC, but does not delete the instance definition within the CAM UI. You could then just re-apply the CAM instance to recreate the VM again later.

2. Delete Instance. This removes the deployed instance from within the CAM UI. It does not touch PowerVC. So this is similar to PowerVC Unmanage VM provided you have not executed step 1.
CAM Service Configuration & Use

Once you have the template(s) working properly, you can then create a CAM service that can be deployed directly from CAM or Published to ICP Catalog and deployed from ICP.

Here we will cover

• Creation and Configuration of the CAM Service
• Publishing the CAM Service to the ICP Catalog
• Deploying the CAM Service from ICP UI
CAM Service Configuration & Use

Create New Service

- Provide a new Service Name
- Optionally add your own category

Examples:
- Generic Linux VMs
- Database Services
- Web Services
- SAP HANA Services
- IBM AI Services
CAM Service Configuration & Use
Create New Service

- Provide Description
  - Short
  - Long
  - ICON

- Information here will be presented as the initial page when selected from the ICP Catalog
CAM Service Configuration & Use

Create New Service

- Add Features
  - VM Characteristics

- Information here will be presented as the initial page when selected from the ICP Catalog
CAM Service Configuration & Use
Create New Service

- Select the “Composition” tab
- Drag and Drop the OpenStack IBM PowerVC Single VM Deploy Template onto the canvas area
- Although not illustrated, notice there are many other templates that could be included in this workflow
CAM Service Configuration & Use
Create New Service

- Click on the “Parameters” tab
- There are three sections
  - Input (User Defined)
  - Activity (From Template)
  - Output (User Defined)
CAM Service Configuration & Use

Create New Service

- We will create a service input variable named `input_flavor_id` to pass into the CAM template

  - Characteristics
    - Read-Write
    - String Type
    - Display Name
    - Description (for hover)
    - Default Value (used PVC ID for Tiny Compute Size)
CAM Service Configuration & Use

Create New Service

- Now we will link our new Input Parameter to the associated Activity Parameter
- Expand the Activity Parameters
- Select “edit” from the right side menu of the Openstack_flavor_id row
CAM Service Configuration & Use

Create New Service

- Select the “Link” tab
- Expand the Input Parameters section
- Click on the “+” sign to the right of `input_flavor_id` to link the two variables together
CAM Service Configuration & Use

Create New Service

Status after the two variables have been linked
CAM Service Configuration & Use
Create New Service

You get a "Standard" plan by default, but let’s add another “Small” Plan and modify it’s input_flavor_id to be the PowerVC Small Compute Template.
Create New Service

Add Plan

* indicates required field

Plan Name:
Small

Description:
PowerVC Small Compute Template

Add “Small” Plan with a Description
CAM Service Configuration & Use

Create New Service

Edit the `input_flavor_id` of the “Small” plan
CAM Service Configuration & Use

Create New Service

Change the default value to the PowerVC Small Compute Template Openstack ID and Save.
CAM Service Configuration & Use

Publish New Service

Now save the service, and then click on “Publish” to the ICP Catalog

Note – Once a service is published, you cannot reuse its name even if you retire the service; thus the suggestion to add –v# for version in the name. This is planned to be fixed in a future release.

Although not shown here you can test a service deploy from within CAM prior to publishing.
CAM Service Configuration & Use

Publish New Service

The new service should now be available to use from the ICP Catalog

If it is not immediately available, you may want to try Manage -> Helm Repositories -> Synchronize in ICP
CAM Service Configuration & Use

Deploy the New Service from ICP

- Click on the new service in the ICP Catalog
- Select the desired Plan
- And click on “Configure”
Deploy the New Service from ICP

- Provide a new service name for the deployment
- Choose your desired namespace for deployment
- Input Flavor ID should contain the appropriate default value for the plans compute template
- Click on Submit
CAM Service Configuration & Use

Deploy the New Service from ICP

- Order Confirmation
- “Go to Instances” navigate back to CAM to see details
- Once deployment completes you will have active status
- Details are viewed by clicking on the service name, then viewing logs.
CAM Service Configuration & Use

Deploy the New Service from ICP

For a service you will have both "Service" and "Activity" logs