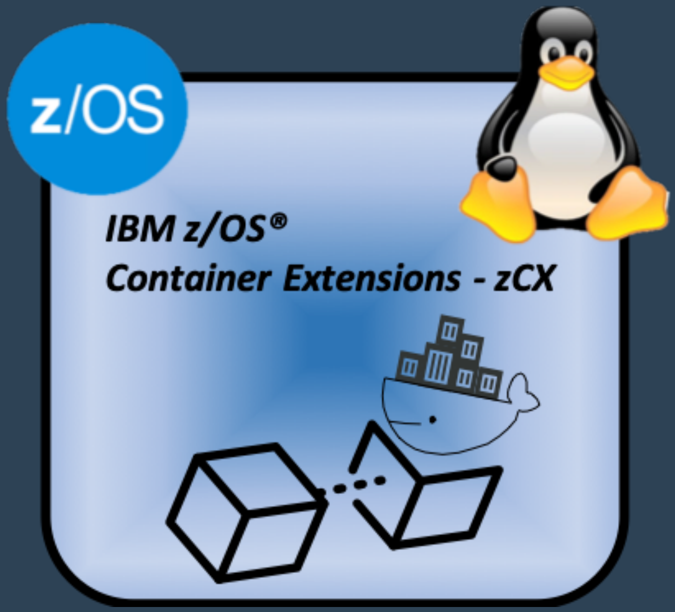
Introduction to z/OS Container Extensions (zCX) and Security Considerations

RACF User Groups
May 2020



Gus Kassimis
IBM Distinguished Engineer
z/OS Architect, Networking and zCX
kassimis@us.ibm.com

Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.

BigInsights
BlueMix
CICS*
COGNOS*
DB2*
DFSMSdfp
DFSMSdss
DFSMShsm
DFSORT
DS6000*
DS8000*
FICON*
GDPS*
HyperSwap
IBM*
IBM (logo)*
IMS
Language Environment*
MQSeries*
Parallel Sysplex*
PartnerWorld*
RACF*
Rational*
Redbooks*
REXX
SmartCloud*
System z10*
Tivoli*
UrbanCode
WebSphere*
z13
zEnterprise*
z/OS*
zSecure
z Systems
z/VM*

* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.
Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.
ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.
Java and all Java based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.
Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
OpenStack is a trademark of OpenStack LLC. The OpenStack trademark policy is available on the OpenStack website.
TEALEAF is a registered trademark of Tealeaf, an IBM Company.
Windows Server and the Windows logo are trademarks of the Microsoft group of countries.
Worklight is a trademark or registered trademark of Worklight, an IBM Company.
UNIX is a registered trademark of The Open Group in the United States and other countries.
Docker and the Docker logo are trademarks or registered trademarks of Docker, Inc. in the United States and/or other countries. Docker, Inc. and other parties may also have trademark rights in other terms used herein.
* Other product and service names might be trademarks of IBM or other companies.

Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

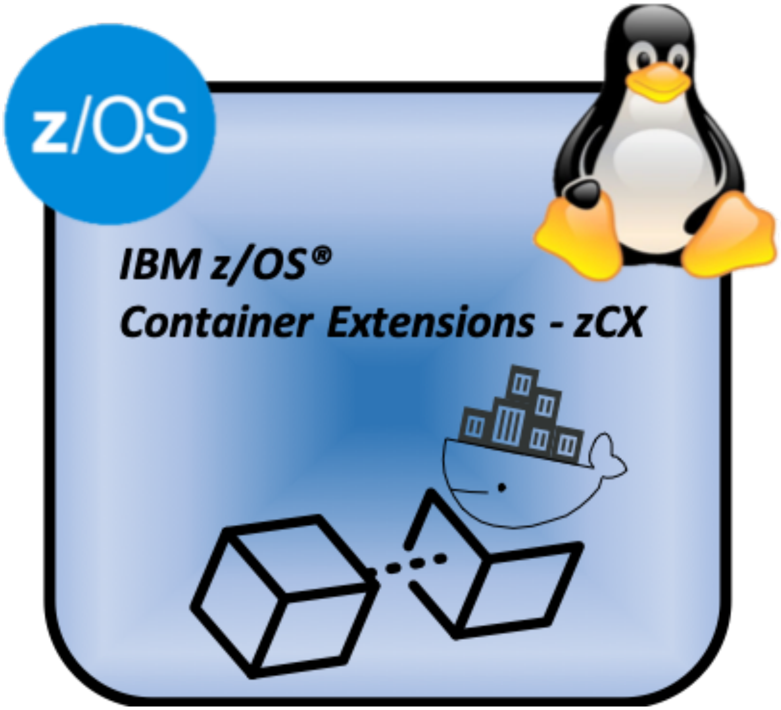
Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

This information provides only general descriptions of the types and portions of workloads that are eligible for execution on Specialty Engines (e.g, zIIPs, zAAPs, and IFLs) ("SEs"). IBM authorizes customers to use IBM SE only to execute the processing of Eligible Workloads of specific Programs expressly authorized by IBM as specified in the “Authorized Use Table for IBM Machines” provided at www.ibm.com/systems/support/machine_warranties/machine_code/aut.html (“AUT”). No other workload processing is authorized for execution on an SE. IBM offers SE at a lower price than General Processors/Central Processors because customers are authorized to use SEs only to process certain types and/or amounts of workloads as specified by IBM in the AUT.

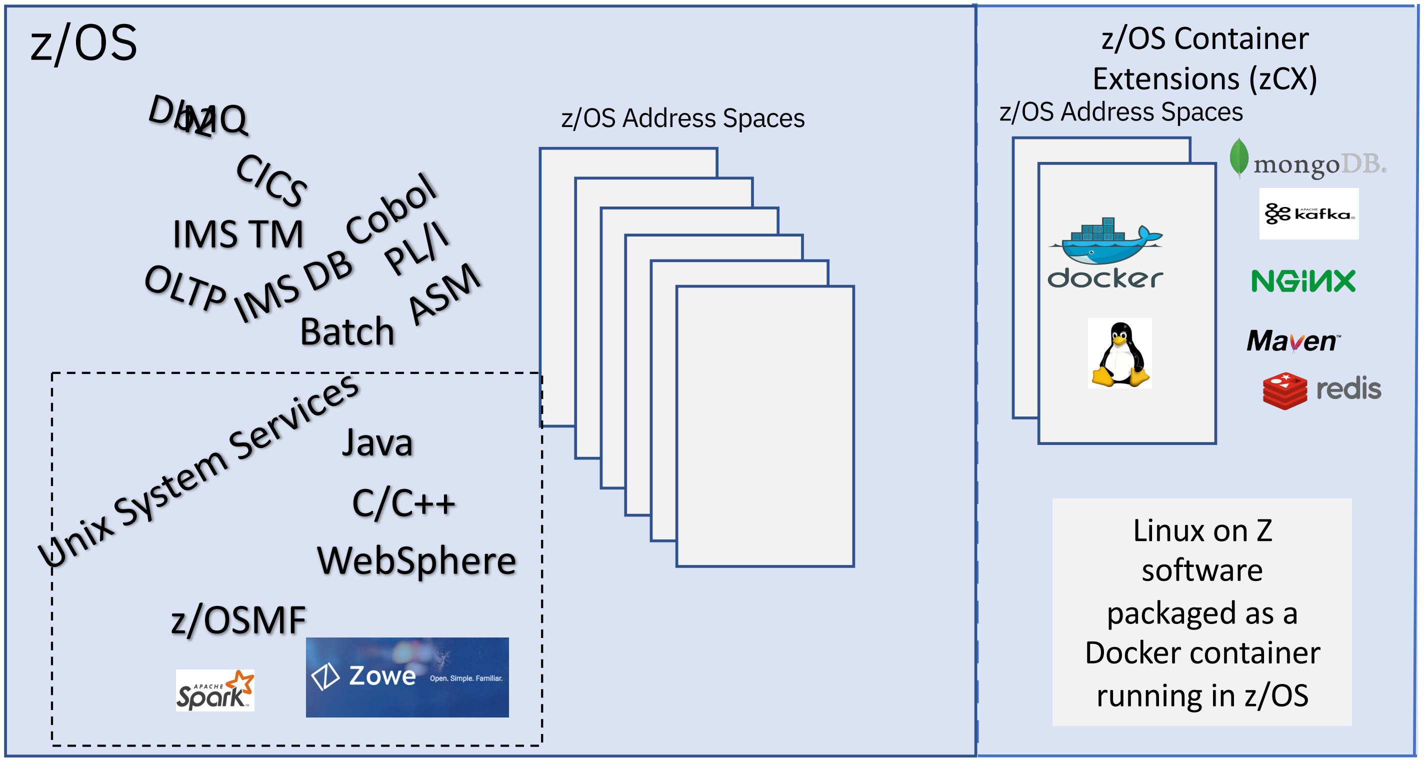
IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion. Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision. The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features or functionality described for our products remain at our sole discretion.

Agenda

- *What is z/OS Container Extensions (zCX)?*
- *What does it enable you to do?*
- *Review of zCX architecture*
- *Security considerations for zCX and containers*



Expanding the z/OS Software Ecosystem



z/OS

DB2

MQ

CICS

IMS TM

Cobol

OLTP

IMS DB

PL/I

Batch

ASM

Unix System Services

Java

C/C++

WebSphere

z/OSMF

z/OS Address Spaces

z/OS Container Extensions (zCX)

z/OS Address Spaces

Linux on Z software packaged as a Docker container running in z/OS

- Traditional z/OS workloads, middleware, subsystems and programming languages
- Unix System Services provided z/OS with a Unix personality enabling porting of Unix applications and new programming languages to the platform
- z/OS Container Extensions (zCX) provides the next big evolution – unmodified Linux on Z Docker images running inside z/OS

What Is IBM z/OS Container Extensions (zCX)?

New function in z/OS 2.4 that enables clients to:

✓ Deploy Linux on Z software components as Docker Containers in a z/OS system, in direct support of z/OS workloads

✓ Without requiring a separately provisioned Linux server

✓ While maintaining overall solution operational control within z/OS and with z/OS Qualities of Service

✓ Requires IBM z14 or z15 server with Container Hosting Foundation (feature code 0104)**

Design Thinking Hill Statement:

A **solution architect** can **create a solution to be deployed on z/OS based on components available as Docker containers** in the Linux on Z ecosystem transparently exploiting z/OS QoS, **without requiring z/OS development skills**.

zCX presents many opportunities, but it is not a replacement for…

Linux on Z Environments

- Native Linux on Z LPARs
- Linux under z/VM
- Linux under KVM on Z
- LinuxONE offerings
- IBM Secure Services Container

Native z/OS Environments & Software

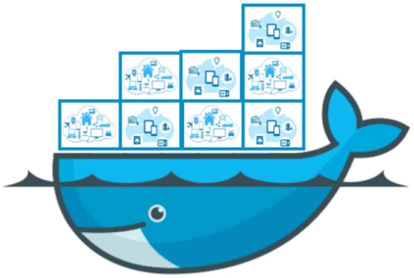
- z/OS UNIX System Services
- Java on z/OS
- Running software natively on z/OS

What is Docker?

- A Packaging standard for software
  - Think of it like a shipping container
  - Makes moving, stacking, unstacking of compliant software easier
  - Common in the application world on Linux and cloud

- Dockerhub
  - Contains many popular docker packages
  - s390x packages support Linux on z
  - https://hub.docker.com/search?q=&type=image&architecture=s390x

- By focusing on Docker
  - We reduce the complexity of installation and configuration for the user
  - We reduce the service footprint on Linux to what Docker supports
  - We gain access to a large number of packages out of the box



zCX – A turn-key Virtual Docker Server Software Appliance

Pre-packaged Linux Docker appliance

- Provided and maintained by IBM
- Provisioned using z/OSMF workflows

Provides standard Docker interfaces

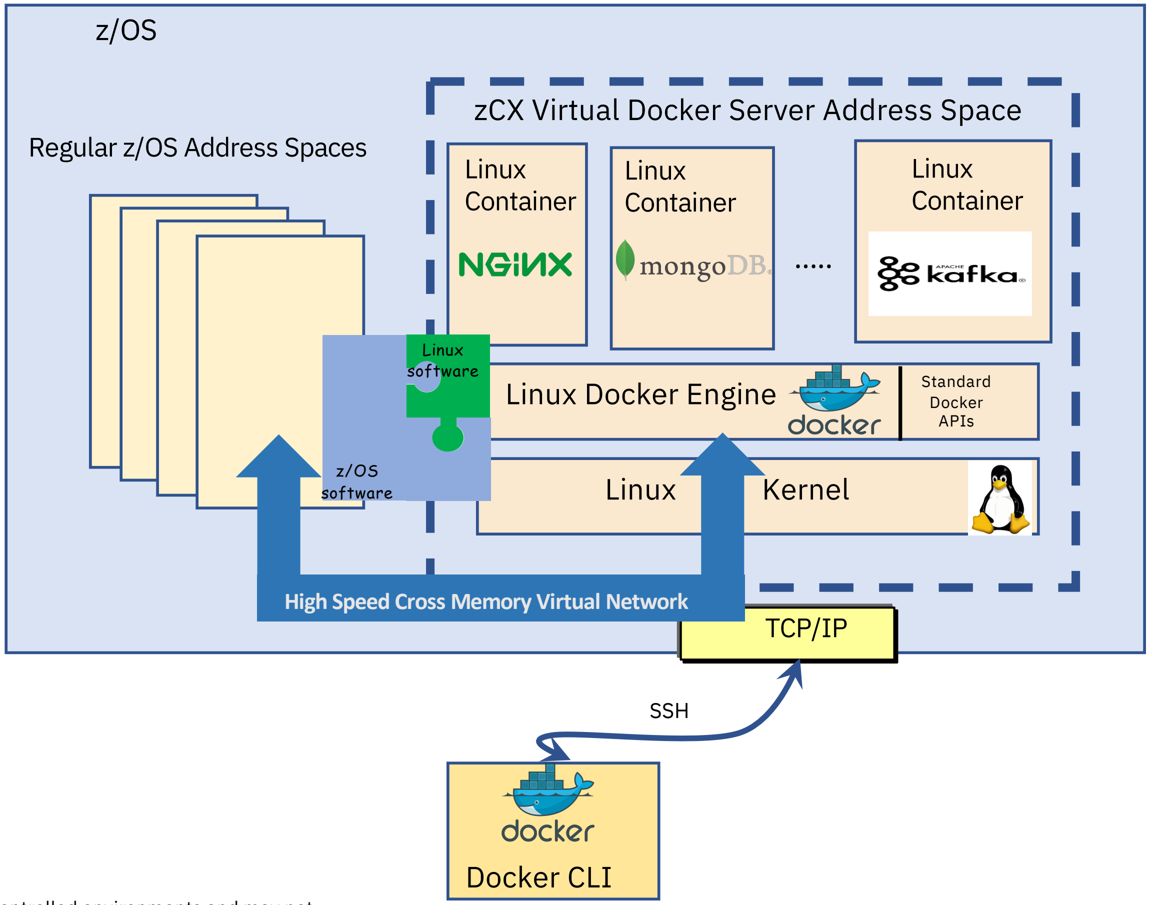
- Supports deployment of any software available as a Docker image for Linux on Z
- Communications with native z/OS applications over high speed virtual IP network
- No z/OS skills required to develop and deploy Docker Containers

No Linux system administration skills required

- Interfaces limited to Docker CLI
- No direct access to underlying Linux kernel

Managed as a z/OS process

- Multiple instances can be deployed in a z/OS system
- Managed using z/OS Operational Procedures
- zCX workloads are zIIP eligible
  - IBM zCX achieved 98% or higher zIIP eligibility for selective zCX container workloads.**



**This is based on IBM internal benchmarks performed in controlled environments and may not representative of the results you will see for your specific zCX workloads in your environment.

IBM zCX - Goals & Qualities of Service

Integrated Disaster Recovery & Planned Outage Coordination

Using z/OS DR/GDPS to cover storage used by Linux automatically, integrated restart capabilities for site failures, etc.

Integrated Planned Outage Coordination

No need to coordinate with non-z/OS administrators when planning a maintenance window, moving workloads to alternate CECs, sites, etc.

z/OS Storage Resilience

Eliminate single points of failure

Exploit z/OS VSAM which offers transparent encryption, and failure detection with HyperSwap

Configuration validation, I/O health checks,

Automatic exploitation of future z/OS Storage enhancements

z/OS Networking Virtualization, Security & Availability

Support for VIPAs, Dynamic VIPAs allowing for non-disruptive changes, failover, and dynamic movement of the workload.

High speed and secure communications with Cross-Memory Virtual Network Interface (SAMEHOST)

z/OS Workload Management, Capacity Planning & Chargeback

WLM: Service Class goals, Business Importance levels, ability to cap resource consumption (CPU and memory)

Capacity Provisioning Manager (CPM) support

SMF support for accounting and chargeback

Use Cases

Expanding the z/OS software ecosystem for z/OS applications

- Latest Microservices (logstash, Etcd, Wordpress, etc.)
- Non-SQL databases (MongoDB, IBM Cloudant, etc.)
- Analytics frameworks (e.g. expanding the z/OS Spark ecosystem)
- Messaging frameworks (example: Apache Kafka, IBM MQ Client Concentrator)
- IBM App Connect Enterprise
- Web server proxies (example: nginx)
- Emerging Programming languages and environments

System Management components

- System management components in support of z/OS that are not available on z/OS
- Centralized data bases for management
- Centralized UI portals for management products – Example:
  - IBM Service Management Unite (SMU)
    - IBM Service Management Unite Suite V1.6 (PID 5698-AAF) is available as a docker image for use with zCX today.

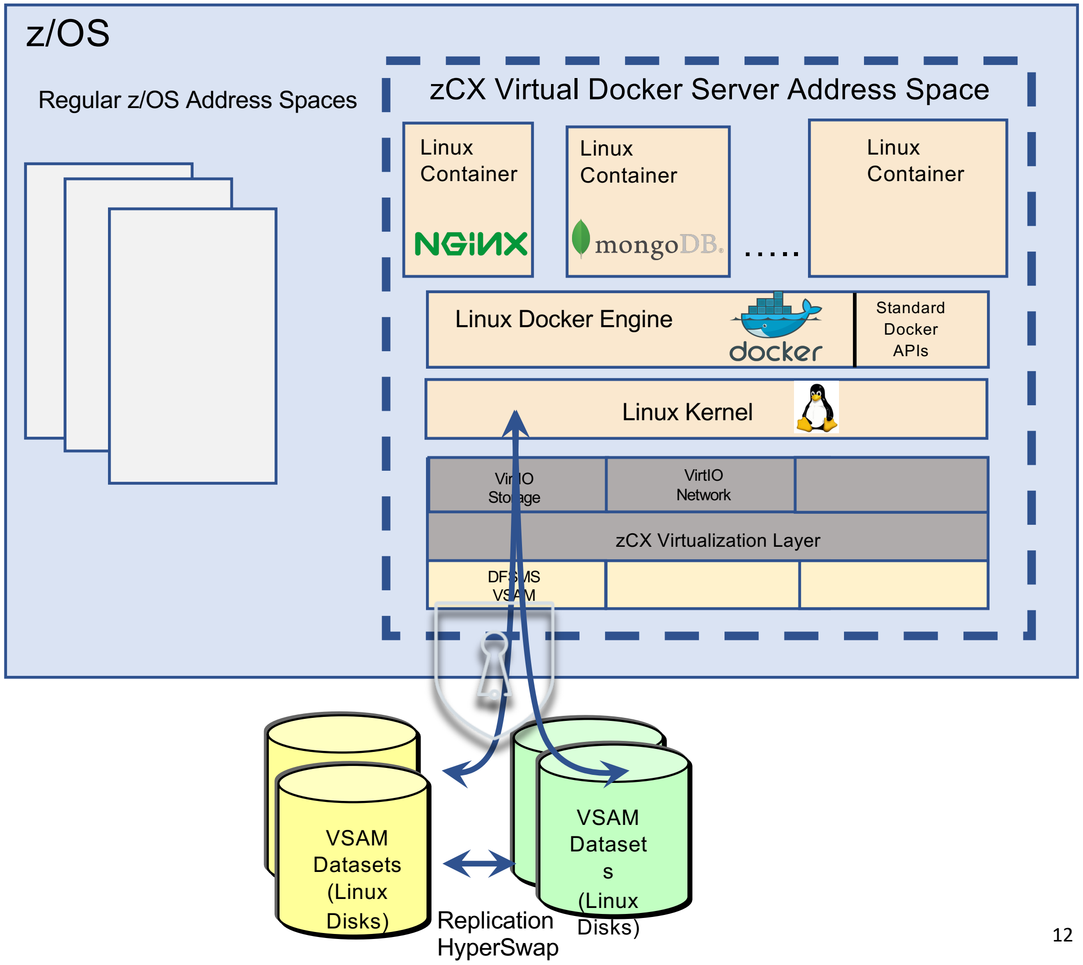
Open Source Application Development Utilities

- Complement existing z/OS ecosystem and Zowe and DevOps tooling
- Gitlab/Github server
- Linux based development tools
- Linux Shell environments
- Apache Ant, Apache Maven

Note: The use cases depicted reflect the types of software that could be deployed in IBM zCX in the future. They are not a commitment or statement of software availability for IBM zCX

IBM zCX – z/OS Storage Integration

- z/OS Linux Virtualization Layer:
  - Allows virtual access to z/OS Storage, Network
  - Using virtio Linux interfaces
  - Allows us to support unmodified, open source Linux for Z
- Linux storage/disk access (via z/OS owned and managed VSAM datasets)
  - Leverages latest I/O enhancements
  - Built-in host-based encryption
  - Replication and HyperSwap technologies for Continuous Availability and Disaster Recovery

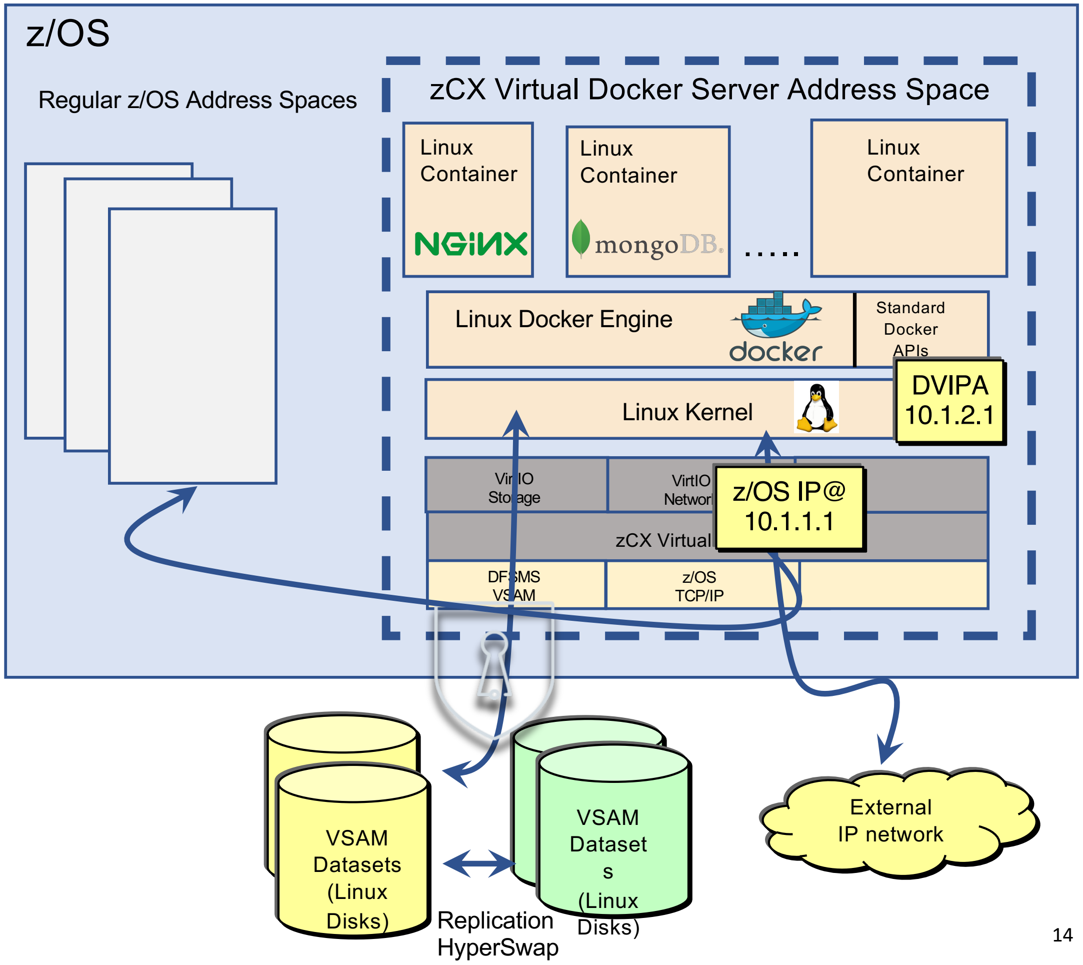


IBM zCX – z/OS Storage Integration – Pervasive Encryption

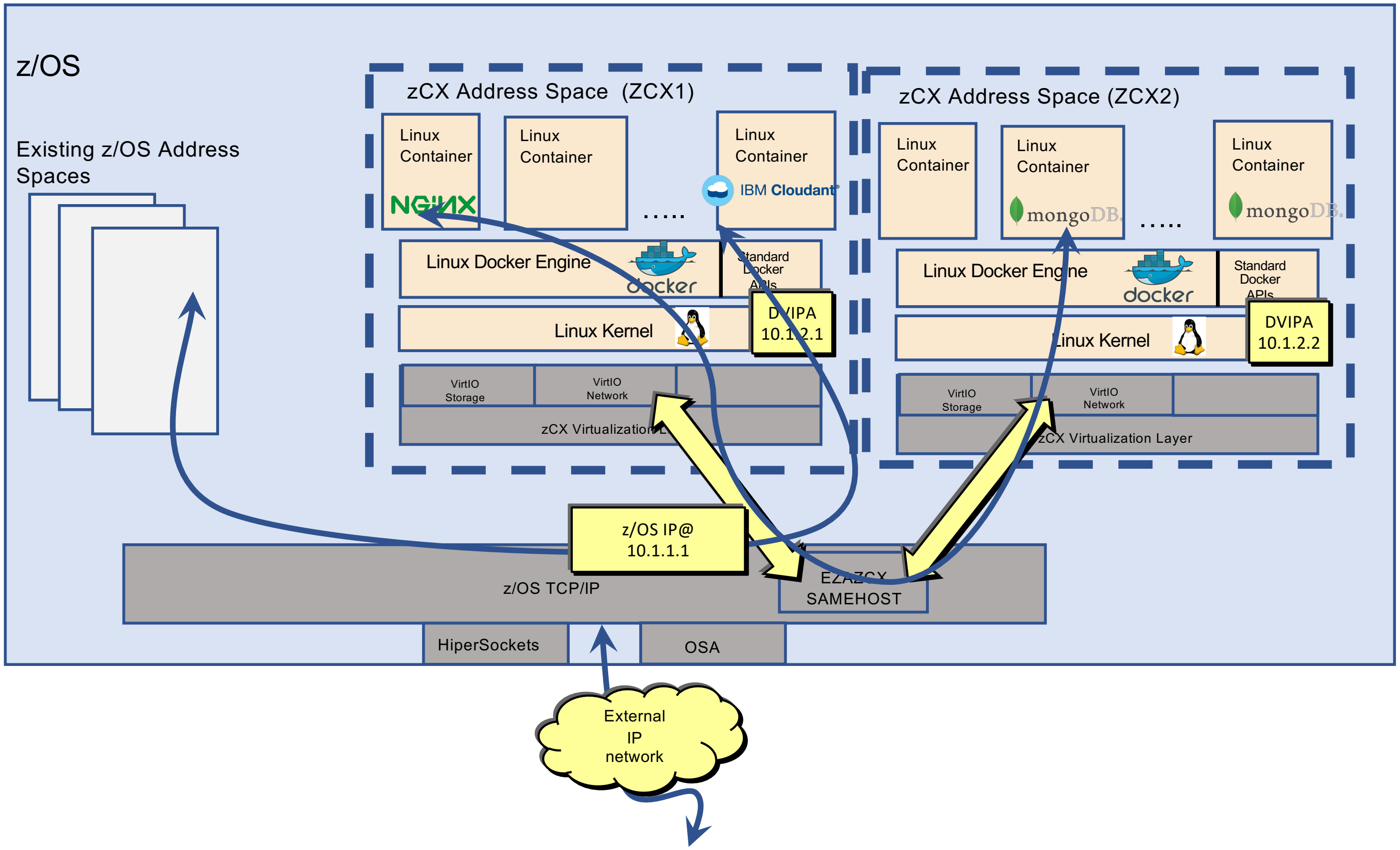
- The zCX Linux root and swap datasets are automatically encrypted by Linux using LUKS encryption.
  - Pervasive encryption is not recommended for the above VSAM linear data sets.
- Pervasive encryption is recommended for the configuration, user data, and diagnostics data VSAM LDS, and for the zCX instance directory zFS file system using VSAM encryption support provided by DFSMS.
  - You can associate an encryption key label with the above data sets either by adding they key label to the DFP segment of the data set's security profile, or by adding the key label to the data set's SMS data class.
- For more information refer to the z/OS Container Extensions Guide: https://www.ibm.om/support/knowledgecenter/en/SSLTBW_2.4.0/com.ibm.zos.v2r4.izso100/izso100_setupsecurity.htm

IBM zCX – z/OS Network Integration

- z/OS Linux Virtualization Layer:
  - Allows virtual access to z/OS Storage, Network and Console
  - Using virtio Linux interfaces
    - Stable, well defined interfaces used to virtualize Linux
  - Allows us to support unmodified, open source Linux for z kernels
- Linux network access via high speed virtual *SAMEHOST* link to z/OS TCP/IP protocol stack
  - Each Linux Docker Server represented by a z/OS owned, managed and advertised Dynamic VIPA (DVIPA)
    - Allows restart of a CX instance in another system in the sysplex
  - Provide high performance network access across z/OS applications and Linux Docker containers – leveraging cross memory
    - All communications between zCX containers and z/OS applications over TCP/IP
    - Support for zCX exploitation of Inbound Workload Queuing (IWQ) now available (APARs PH16581/OA58300)
  - External network access via z/OS TCP/IP
    - z/OS IP filters to restrict external access



Getting Started with zCX: Networking
IBM zCX – High Speed Virtual IP Network – SAMEHOST (EZAZCX)



- z/OS TCP/IP acts as a router for all traffic in/out of zCX
- IP Security filters can be used to permit/deny traffic to/from zCX instances
- z/OS TCP/IP does not have awareness of TCP/UDP ports being used in zCX containers (i.e. no port reservation statements in TCP/IP profile)
- zERT (z/OS Encryption Readiness Technology) is not aware and does not report on zCX traffic unless the remote endpoint is also on z/OS

zCX Network Configuration Steps

1. zCX Network information that will be needed for each zCX instance (inputs to z/OSMF zCX provisioning workflow):
   - zCX Server IP address – an IPv4 zCX DVIPA,
   - DNS Server IP Addresses (up to 2 for resiliency)
   - DNS Search Domain – example: pok.ibm.com, ibm.com
   - MTU (optional, default = 1492, suitable for most environments)
   - TCP/IP Stack name (only needed if multiple TCP/IP stacks are configured/active on the z/OS system)

2. z/OS TCP/IP profile:
   - zCX DVIPA(s) - Using VIPARANGE statements, configure zCX DVIPAs (IPv4 and optionally IPv6).
     The DVIPA must match zCX server configuration! (Must match the z/OSMF Workflow configuration, step 1 above)
   - *Note:* The same VIPARANGE statements should be replicated across all systems in the Sysplex that you wish to start this zCX instance on.

3. OMPROUTE profile:
   - Updates for zCX Dynamic VIPAs being used (Same as other DVIPAs – Use wildcarding where possible to simplify configuration)
   - And remember to propagate these to all other systems in the Sysplex that this zCX instance may be started on

4. *IPSec Policy:*
   - *If you have IP Filters defined you need to ensure that to ensure that you permit ROUTED and LOCAL traffic for these DVIPAs*

VIPARange ZCX (syntax and definition)

```
•               .-DEFINE-.    .-MOVEable NONDISRUPTive--.
• >>-VIPARange--+--------+--+-+-------------------------+--address_mask--ipv4_addr------+--+----------
-+---->
•               '-DELEte-'  | '-MOVEable DISRUPTive-----'               'SAF resname'   |
•                           | .-MOVEable NONDISRUPTive--.                               |
•                           '--+-------------------------+--ipv6_intfname--ipv6_addr/prefix_len--'
•
• >------+--------+----><
•        '--ZCX---'
```

Notes:

- zCX DVIPAs are defined with VIPARANGE with a new keyword “ZCX”.
- The MOVEABLE keyword is ignored on a ZCX VIPARANGE (i.e. a zCX DVIPA can’t be activated if already active).
- An expected use case is that a zCX VIPARANGE may exist on multiple hosts so it should remain in sysplex VIPARange configuration.

- Support will also be available in the z/OSMF Network Configuration Assistant for defining zCX DVIPAs under the Configure Sysplex Networking actions

zCX DVIPAs – Security Considerations

VIPARANGE DVIPA creation can be controlled through two SERVAUTH profiles:

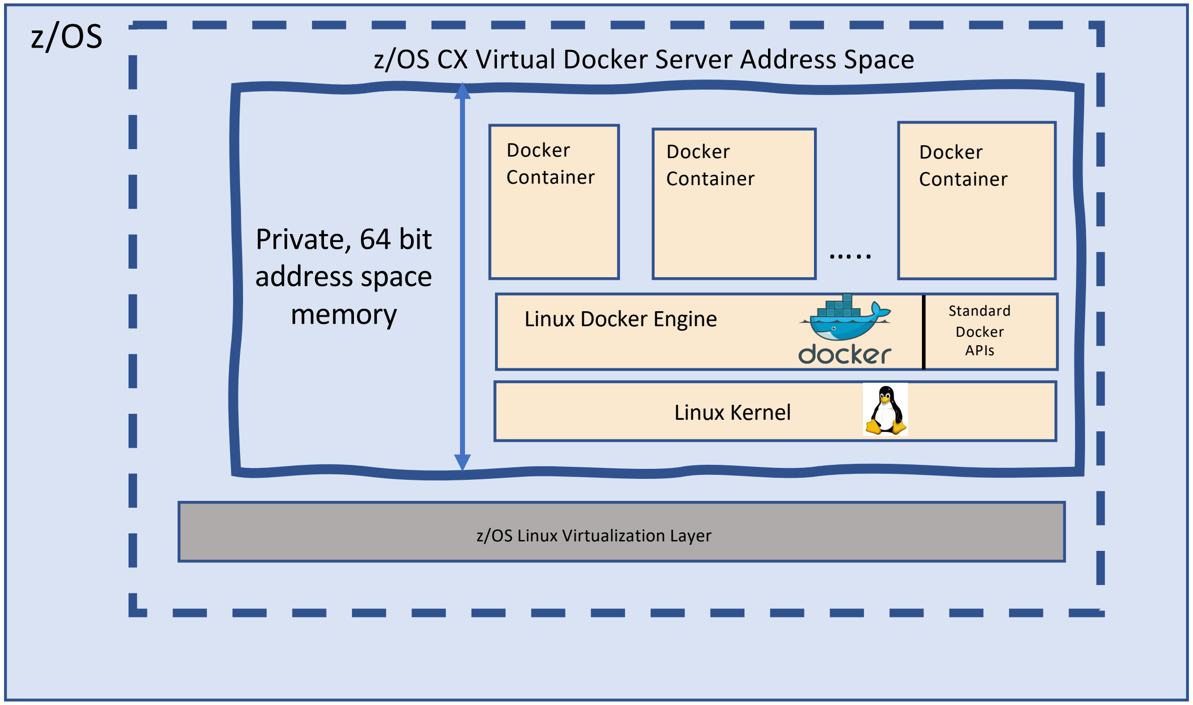
- EZB.MODDVIPA.*sysname*.*tcpname*
  - Limits who can create a VIPARANGE DVIPA in general
- EZB.MODDVIPA.*sysname*.*tcpname*.*resname*
  - Limit who can create a specific VIPARANGE DVIPA:
    - VIPARANGE DEFINE 255.255.255.255 10.10.10.1 *SAF APPL1* *ZCX*
    - Profile: EZB.MODDVIPA.*sysname*.*tcpname*.*APPL1*

If either of these 2 profiles are enabled then the userid associated with the zCX Started task will require READ access to these profiles

- If these profiles are not enabled then the userid associated with the zCX Started task must be UID(0) or have READ access to BPX.SUPERUSER FACILITY class profile

IBM zCX – Memory Architecture and Isolation Characteristics

- Linux guest has addressability to private, 64-bit memory in the zCX address space
  - This contiguous virtual memory range represents the total real memory that Linux has access to
  - *Linux cannot access any memory outside of that private memory range*
    - ***Cannot*** access Common Storage (above or below the bar)
    - ***Cannot*** access memory for other address spaces (no cross-memory support)
    - ***Cannot*** any other memory inside the zCX address space
  - *Linux cannot perform any direct execution of code outside its private memory range*
    - ***Cannot*** branch, PC or SVC to native z/OS code outside of its memory object
    - The only way to communicate with other z/OS processes is using standard TCP/IP sockets
  - Impacts from misbehaving containers running inside zCX are confined to that zCX instance

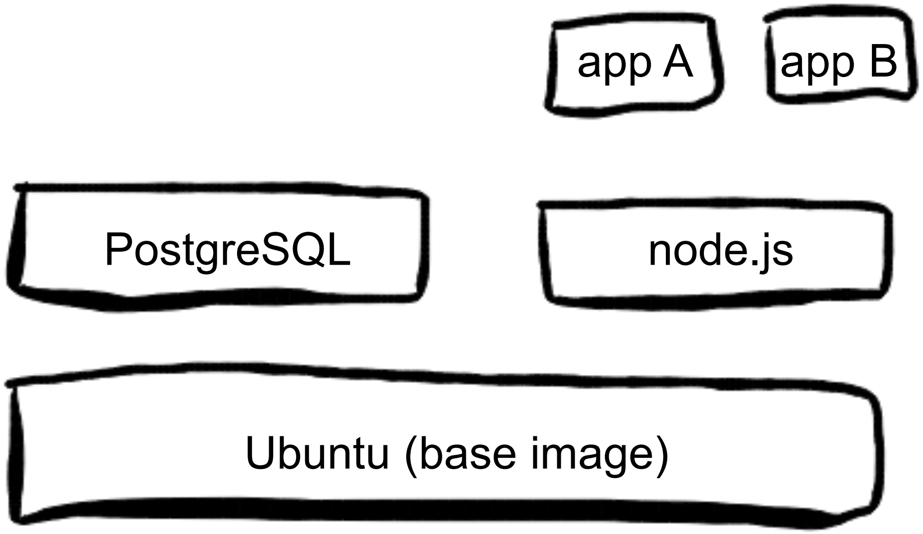


Docker Containers: “Build, Ship, and Run Any App, Anywhere”

- One implementation of a container solution
- Powerful tool to build, modify, deploy, run, manage containers
  - Extreme focus on efficiency, fast response times
  - Stores incremental differences and caching whenever possible
- Registries serve as central places for images
  - Efficient distribution, versioning
  - Internal enterprise trusted registries typically deployed to provide governance over trusted images
- Terminology
  - image: a self contained set of files, base for a container
  - container: runnable instance, based on an image
- Maintained by Docker, Inc.

Typical Container Layering

- Images are built using layers
  - Each layer has an associated JSON structure describing its basic information
  - Layers are stored on Docker Registry as gzipped tar files(.tar.gz)
  - Layers are stacked into a Docker Image via ‘pointer to parent layer’
  - Allows to build on common infrastructure
- Only differences are stored and pushed
  - Memory efficiency and density
- Change in underlying layer requires rebuilding all depending images
  - Will generate a new image (with new ID) for app A
  - Having both versions of app a allows for simple migration and rollback
- Difference between Docker Image and Docker Container?
  - A Docker Image is a immutable snapshot of a live Docker Container
  - A Docker Container is a running instance of a Docker Image



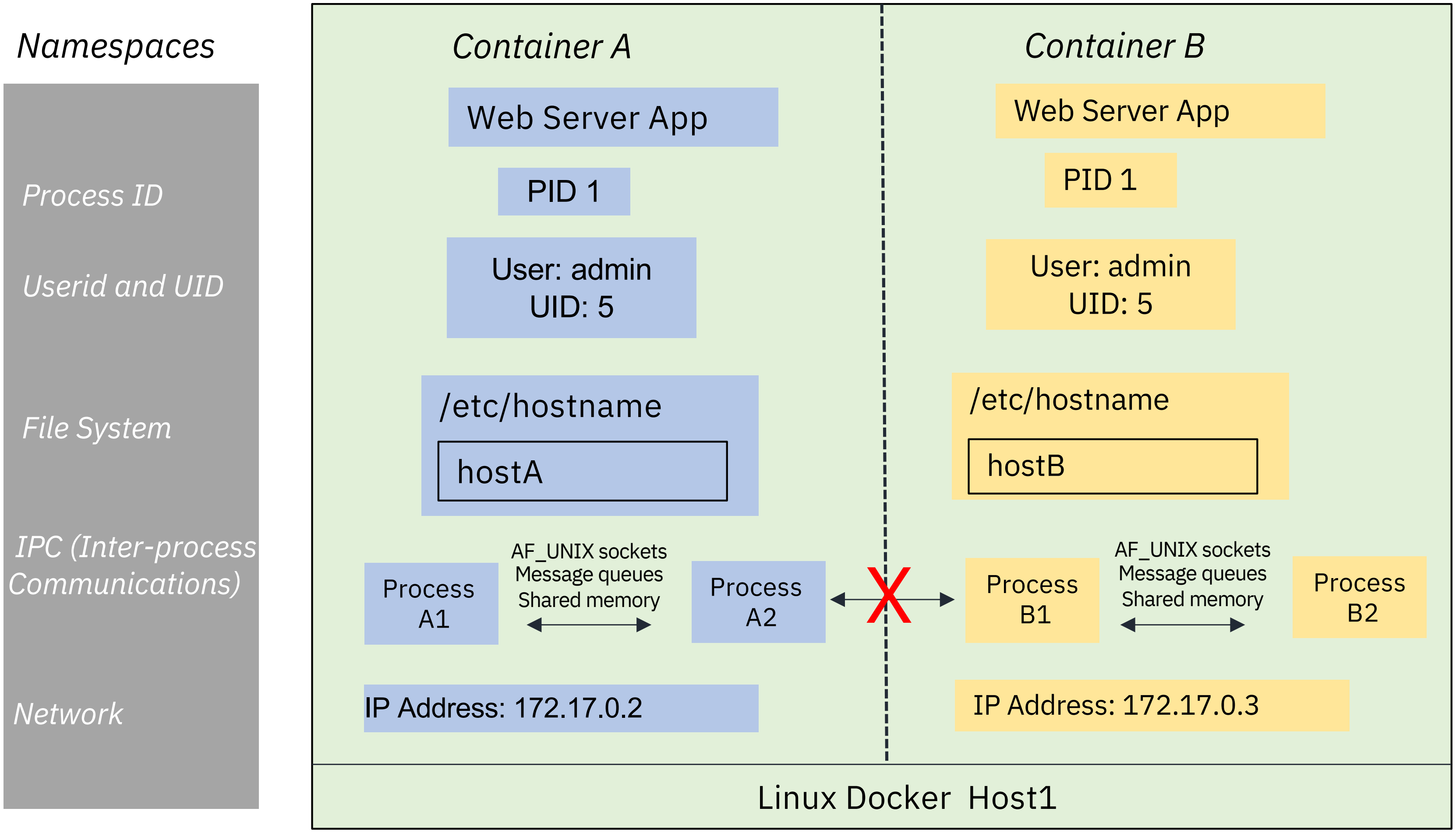
Key Container Characteristic:
Resource Isolation

Containers isolate applications' execution environments from each another and largely from the underlying OS.

They use controlled portions of the host operating system's resources; many applications share the same OS kernel, in a highly managed way. They provide the ability to govern the isolation and usage of system resources, such as CPU, memory, I/O, networking, etc for a group of processes.

With containers on Linux this is managed through Cgroups (Control Groups) and Namespace isolation at the network, file system, security (uid and userid) and IPC layers.

Containers and Name Space Isolation – Multiple containers of the same image



Containers:
Escaping outside of the container name space

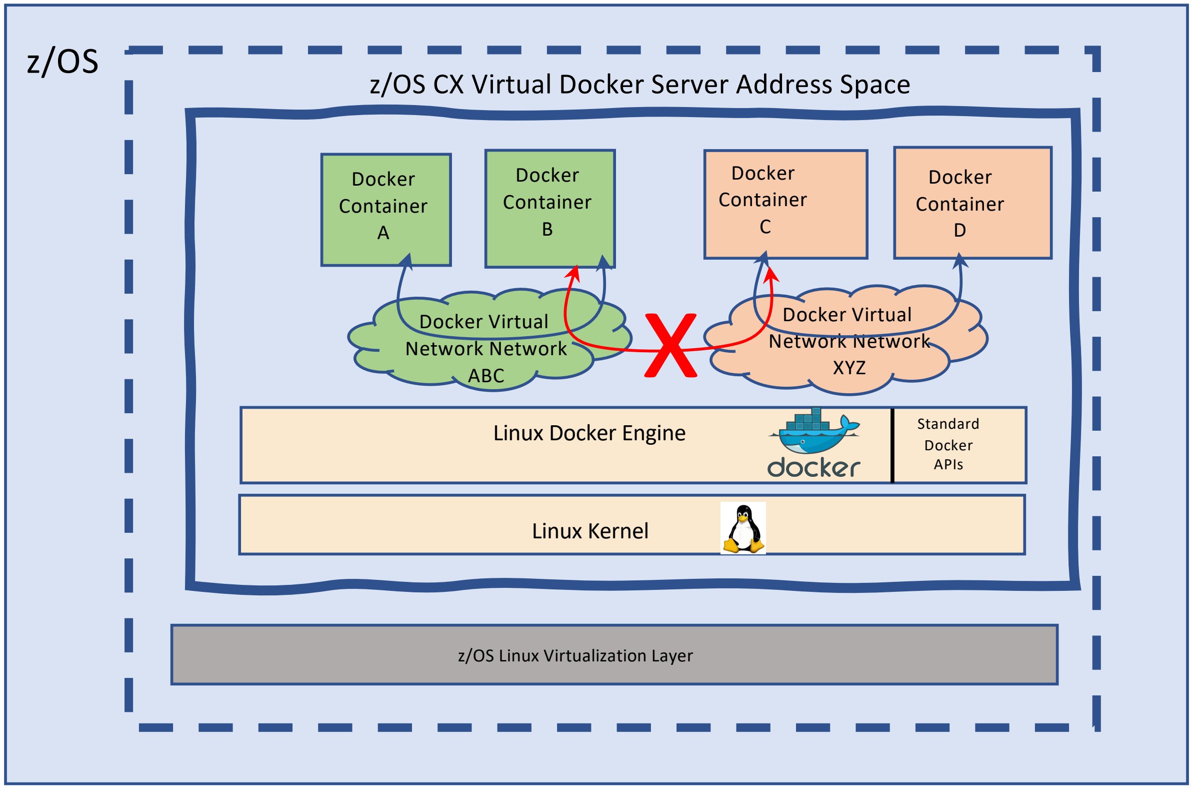
Containers can escape out of the virtual container name space using the --privileged option on the Docker run command:

```
            docker run --privileged ubuntu bash
```

zCX does not allow deployment of privileged containers – Docker commands specifying --privileged or --network=host options

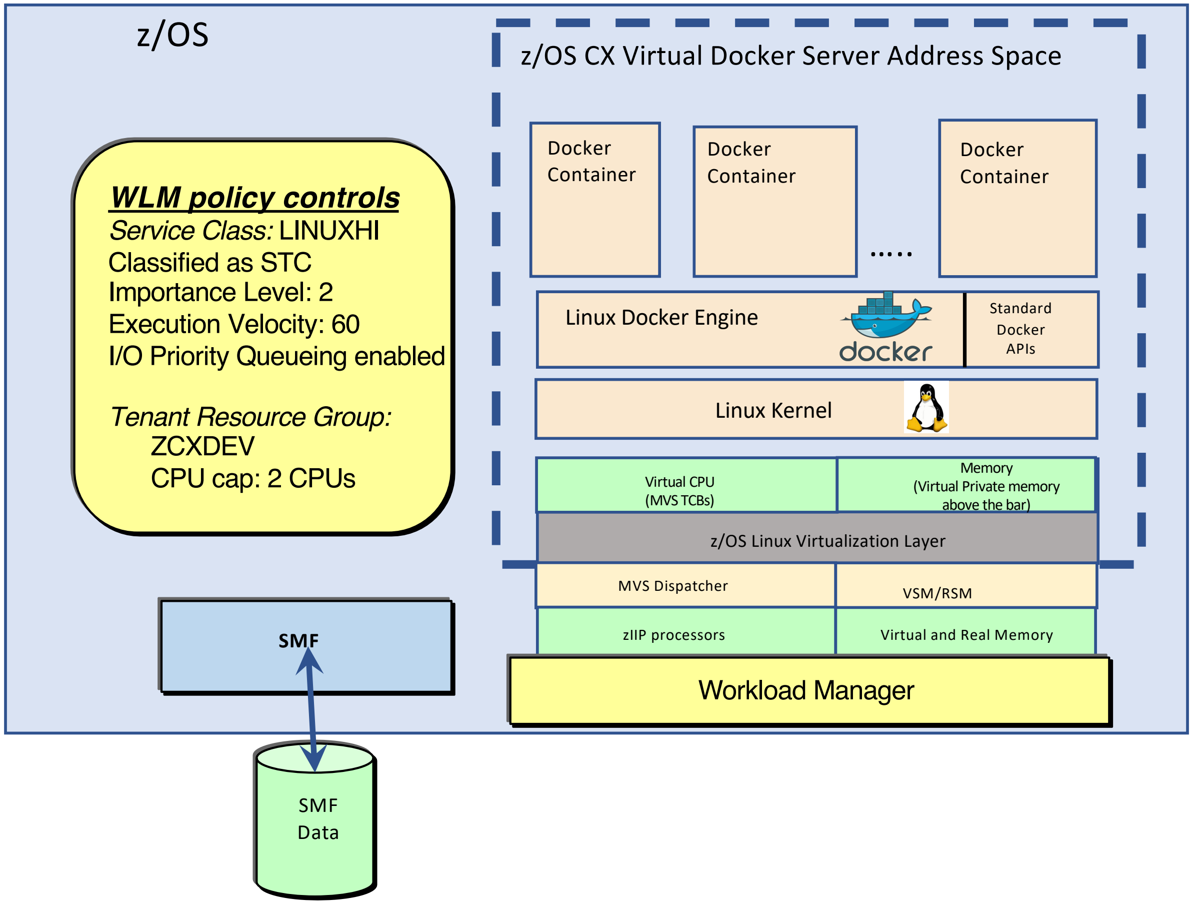
IBM zCX – Docker Virtual Networks

- By default, Docker creates a virtual network interface for each container deployed
  - The default Docker bridge network is defined automatically
- Administrators can define additional private Docker networks within a Docker server
  - Containers can be attached to one or more private networks during deployment
  - Containers in the same Docker virtual network can communicate directly with each other
  - Containers in different virtual networks can be isolated from each other
- Docker also supports other virtual networking options
  - Host network option (allow a container to use the underlying Linux host network interfaces directly) – this option is disallowed by zCX
  - Overlay networks – for virtual networks across a Docker Swarm cluster


z/OS
z/OS CX Virtual Docker Server Address Space
Docker Container A
Docker Container B
Docker Container C
Docker Container D
Docker Virtual Network Network ABC
X
Docker Virtual Network Network XYZ
Linux Docker Engine
docker
Standard Docker APIs
Linux Kernel
z/OS Linux Virtualization Layer

IBM zCX - CPU, Memory and Workload Management

- Memory Management
  - Provisioned per zCX Docker Server address space
  - Private, above the 2GB bar Fixed Memory
  - Managed by VSM, RSM
- CPU Management
  - Virtual CPUs provisioned to each zCX Docker Server address space
    - Each virtual CPU is a dispatchable thread (i.e. MVS TCB) within the address space
    - zIIP CPU access via MVS dispatcher
  - A zCX instance can host multiple Docker Container instances
- Normal WLM policy and resource controls extend to zCX Docker Server address spaces
  - Service Class association, goals and Importance levels
  - Tenant Resource Group association
    - Optional caps for CPU and real memory
- Normal SMF data available
  - SMF type 30, 72, etc.
  - Enables z/OS performance management and capacity planning



z/OS

z/OS CX Virtual Docker Server Address Space

Docker Container

Docker Container

.....

Docker Container

Linux Docker Engine

docker

Standard Docker APIs

Linux Kernel

Virtual CPU (MVS TCBs)

Memory (Virtual Private memory above the bar)

z/OS Linux Virtualization Layer

MVS Dispatcher

VSM/RSM

zIIP processors

Virtual and Real Memory

Workload Manager

WLM policy controls
Service Class: LINUXHI
Classified as STC
Importance Level: 2
Execution Velocity: 60
I/O Priority Queueing enabled

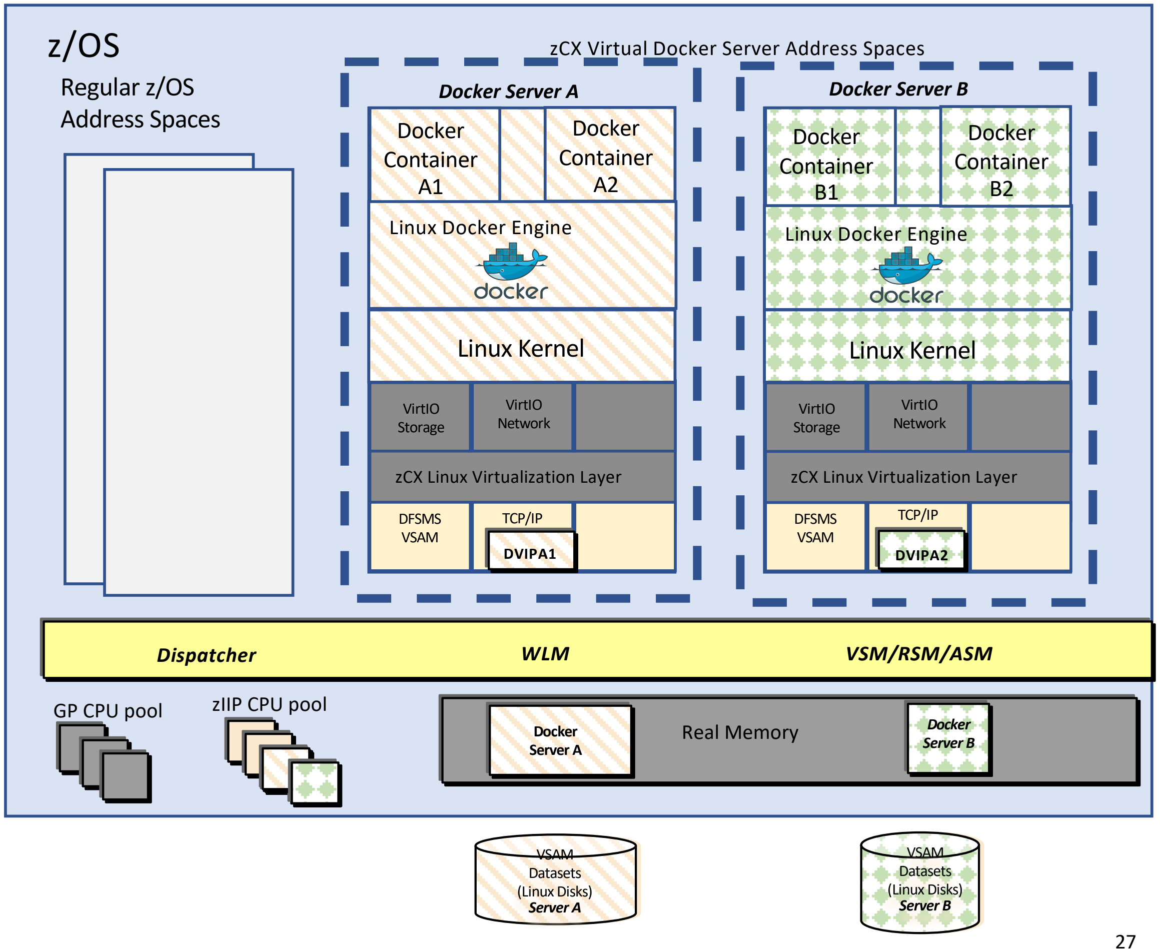
Tenant Resource Group:
ZCXDEV
CPU cap: 2 CPUs

SMF

SMF Data

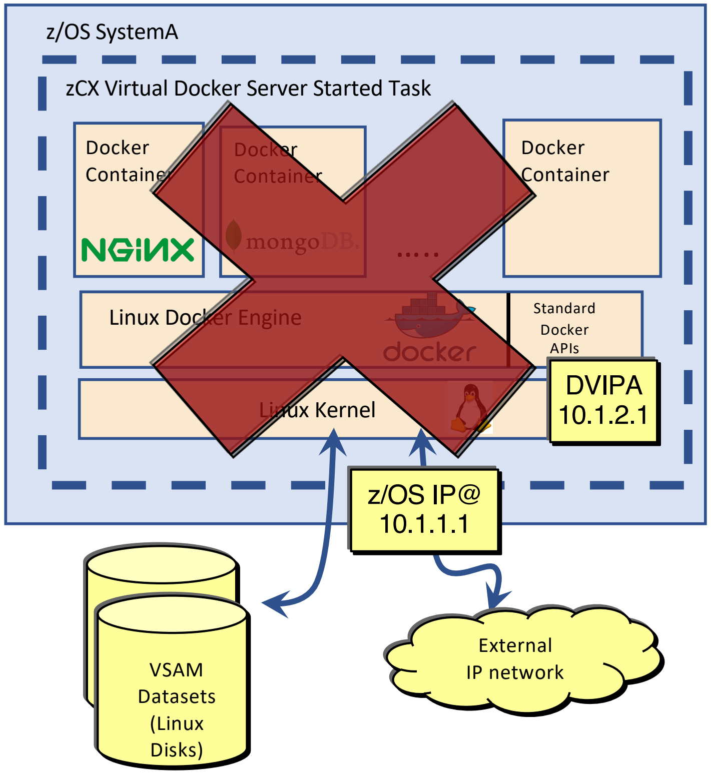
Deploying Multiple zCX Virtual Docker Server Instances

- Multiple zCX instances can be deployed within a z/OS system:
  - Isolation of applications (containers)
  - Different business/performance priorities (i.e. unique WLM service classes)
  - Capping of resources allocated for related workload (CPU, memory, disk, etc.)
- Each zCX address space:
  - Has specific assigned storage, network and memory resources
  - Shares CPU resources with other address spaces
    - But can influence resource access via configuration and WLM policy controls
- A new Hypervisor built using existing z/OS capabilities
  - The z/OS Dispatcher, WLM and VSM/RSM components manage access to CPU and memory
  - The zCX virtualization layer manages Storage, Network and Console access
    - Using dedicated resources
    - There is no communications across z/OS Linux virtualization layer instances
- Integrated z/OS Capacity Provisioning and Management
  - WLM, CPM, adding/removing CPU and Memory resources



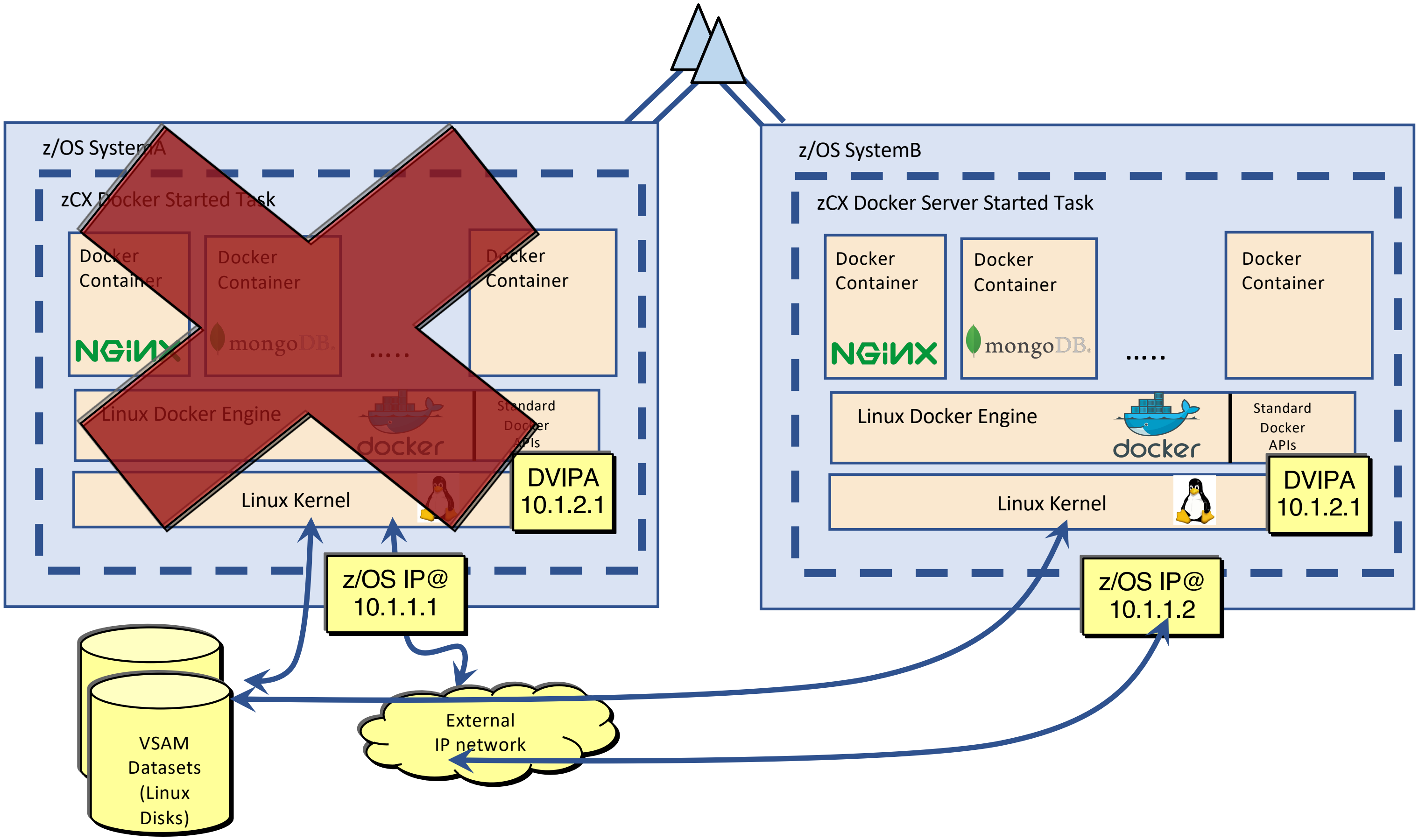
z/OS Container Extensions
Operations and Disaster Recovery Integration

- Started using z/OS Start Command
  - –Support for Start, Stop, Modify
- Automated Operations using z/OS facilities
  - –System Automation
  - –Automatic Restart Manager (ARM)
  - –Other z/OS Automation framework/product
- Planned and Unplanned Outage and Disaster Recovery coordination
  - –zCX Docker Server failure (restart in place)



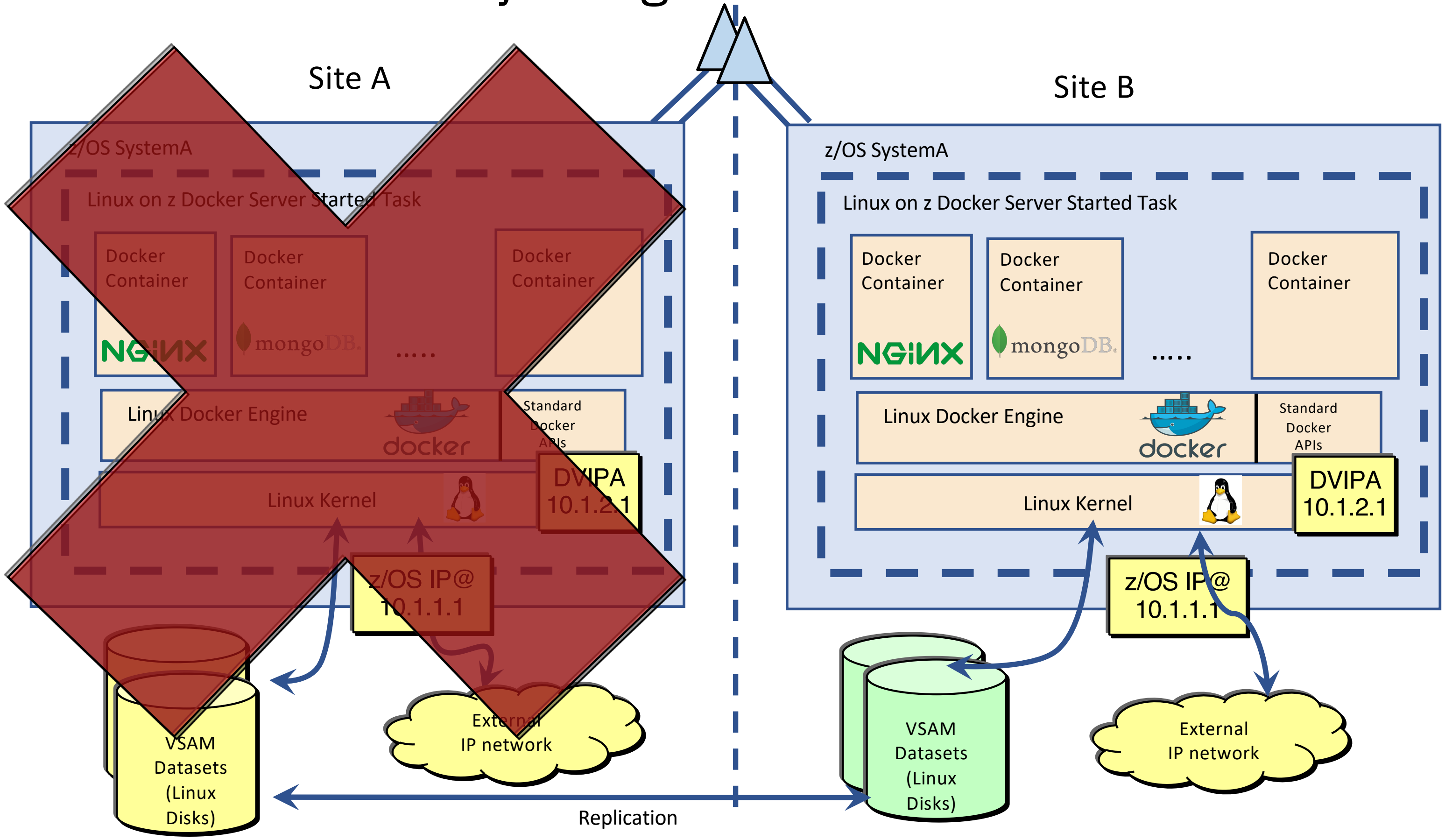
z/OS Container Extensions
Operations and Disaster Recovery Integration

- Started using z/OS Start Command
  - Support for Start, Stop, Modify
- Automated Operations using z/OS facilities
  - System Automation
  - Automatic Restart Manager (ARM)
  - Other z/OS Automation framework/product
- Planned and Unplanned Outage and Disaster Recovery coordination
  - zCX Docker Server failure (restart in place)
  - LPAR failure (restart on other LPAR in the sysplex)



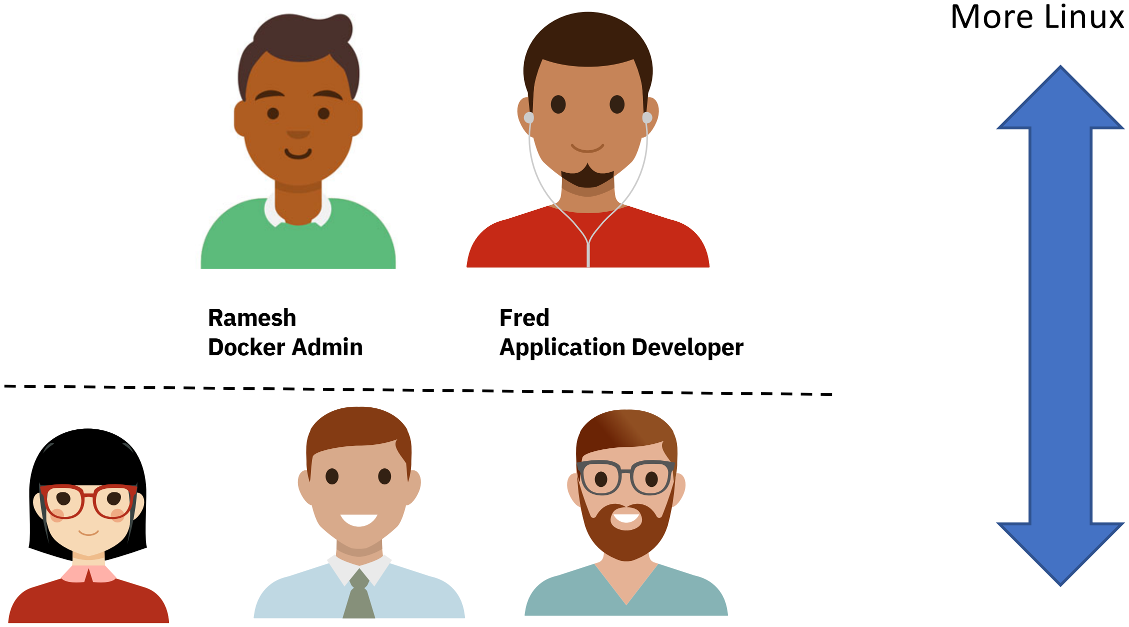
z/OS Container Extensions
Operations and Disaster Recovery Integration

- Started using z/OS Start Command
  - Support for Start, Stop, Modify
- Automated Operations using z/OS facilities
  - System Automation
  - Other z/OS Automation framework/product
- Planned and Unplanned Outage and Disaster Recovery coordination
  - z/OS Container Extensions Docker Server failure (restart in place)
  - LPAR failure (restart on other LPAR in the sysplex)
  - Site failure (restart on alternate site) – GDPS or other automated DR framework



Personas

Personas



More Linux Skill

Ramesh
Docker Admin

Fred
Application Developer

Shichi
IT Architect

Omar
Solution Architect

Zach
z/OS Systems Programmer
(includes Networking,
Storage, Security, WLM, etc.
Admins)

More z/OS Skill

The Experiences Today

DISCOVER, TRY, BUY
How do I get it?

GET STARTED
How do I get value?

EVERYDAY USE
How do I get my job done?

MANAGE AND UPGRADE
How do I keep it running?

LEVERAGE AND EXTEND
How do I build on it?

SUPPORT
How do I get unstuck?

The Experiences Today

DISCOVER, TRY, BUY
How do I get it?

GET STARTED
How do I get value?

EVERYDAY USE
How do I get my job done?

MANAGE AND UPGRADE
How do I keep it running?

LEVERAGE AND EXTEND
How do I build on it?

SUPPORT
How do I get unstuck?

Experimenting with zCX

z/OS Container Extensions trial
z/OS Container Extensions (zCX) is now available on a trial [1] basis. Clients may try zCX for up to 90 days without having to purchase hardware FC 0104. When the 90-day trial period has ended, zCX instances will no longer function unless FC 0104 has been installed.

zCX trial can be applied with the PTF for APAR OA58969. With the trial APAR, it is intended that clients will have a full zCX user experience for the 90 days once zCX is enabled. For more details:
IBM Announcement Link

z/OS V2.4 and z14 or z15 processor required!

Zach
Systems Programmer

[1] 90-day trial is free subject to normal hardware and software consumption when adding a workload to z/OS.

The Experiences Today

DISCOVER, TRY, BUY
How do I get it?

GET STARTED
How do I get value?

EVERYDAY USE
How do I get my job done?

MANAGE AND UPGRADE
How do I keep it running?

LEVERAGE AND EXTEND
How do I build on it?

SUPPORT
How do I get unstuck?

Provisioning

Zach can provision one or more z/OS Container Extensions instances in a z/OS system, each with custom:

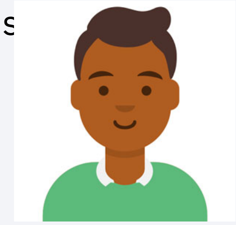
- Resource allocation
  - Number of virtual CPUs, memory, network connectivity and storage
- Docker Configuration settings
- Definition of z/OS Container Extensions appliance admin user and Docker admin user

Resource Allocation:

- zIIP eligible CPUs, resource capping possible via WLM Resource Groups or Tenant Resource Groups
- Support for Fixed z/OS Memory (not pageable), estimated 1GB minimum
- Support for Dynamic VIPA (DVIPA support)
- z/OS VSAM LDS for storage with support for encryption and replication

Docker Configuration Options

- Registry to be used
- Logging options
- Other (tbd)



Ramesh
Docker Admin

Zach
Systems Programmer

The Experiences Today

DISCOVER, TRY, BUY
How do I get it?

GET STARTED
How do I get value?

EVERYDAY USE
How do I get my job done?

MANAGE AND UPGRADE
How do I keep it running?

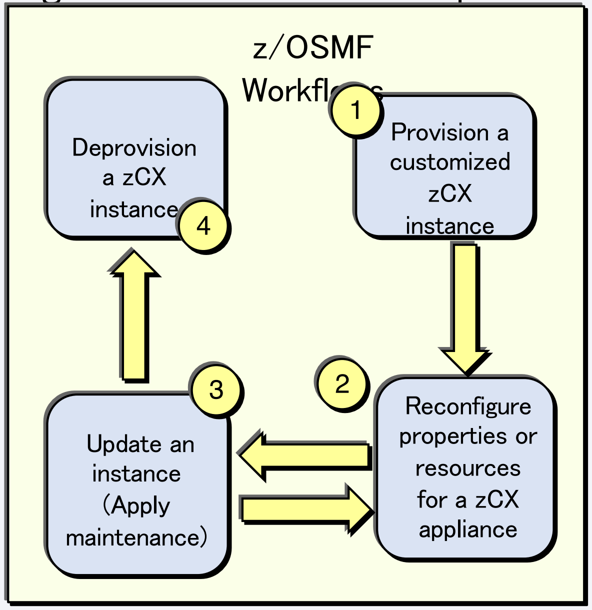
LEVERAGE AND EXTEND
How do I build on it?

SUPPORT
How do I get unstuck?

Provisioning (continued)

Provisioning and deprovisioning and lifecycle management via provided z/OSMF workflows

- Automates many of the steps of provisioning a Container Extensions instance
  - You can provision a zCX instance in a few minutes
- Provides guidance for out of band steps (RACF/SAF resources, TCP/IP network definitions, WLM definitions, DFSMS setup)
- Runs as Started Task, can be started/stopped via operator commands and integrated into automated operations procedures


z/OSMF Workflows
1
Provision a customized zCX instance
2
Reconfigure properties or resources for a zCX appliance
3
Update an instance (Apply maintenance)
4
Deprovision a zCX instance

Zach
Systems Programmer

The Experiences Today

DISCOVER, TRY, BUY
How do I get it?

GET STARTED
How do I get value?

EVERYDAY USE
How do I get my job done?

MANAGE AND UPGRADE
How do I keep it running?

LEVERAGE AND EXTEND
How do I build on it?

SUPPORT
How do I get unstuck?

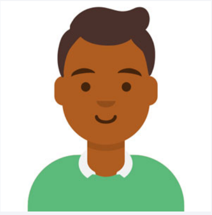
Docker administrators and permitted Docker users can deploy any Linux on Z docker container image using standard Docker CLI

Access to Docker CLI by remote access into IBM provided and controlled SSHD container environment (included and active in each z/OS Container Extensions instance)

Remote Docker CLI access will not be supported

SSH access to underlying Linux kernel will not be supported

Zach
Systems Programmer



Ramesh
Docker Admin

Fred
Application Developer

Omar
Solution Architect

The Experiences Today

DISCOVER, TRY, BUY
How do I get it?

GET STARTED
How do I get value?

EVERYDAY USE
How do I get my job done?

MANAGE AND UPGRADE
How do I keep it running?

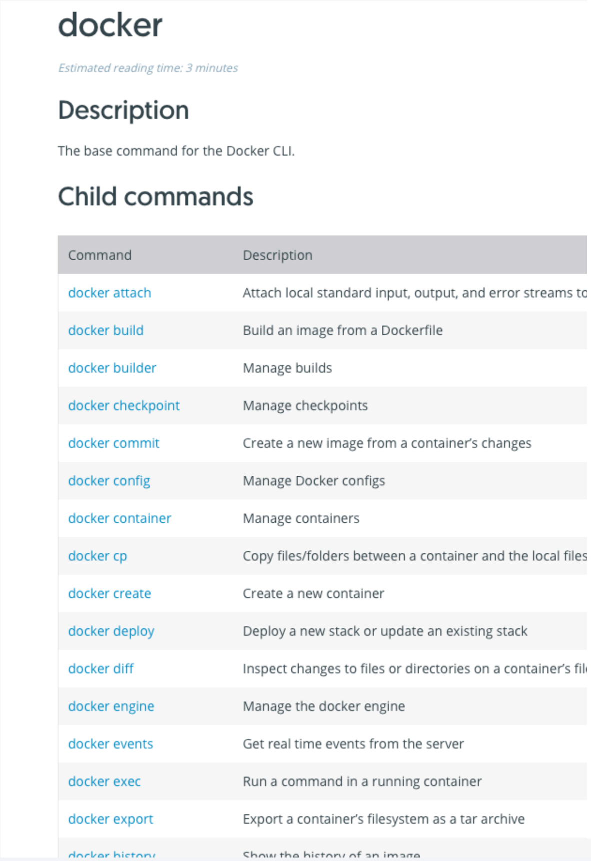
LEVERAGE AND EXTEND
How do I build on it?

SUPPORT
How do I get unstuck?

Docker CLI (Command Line Interface)

https://docs.docker.com/engine/reference/commandline/docker/

Standard Docker CE command line interface



docker

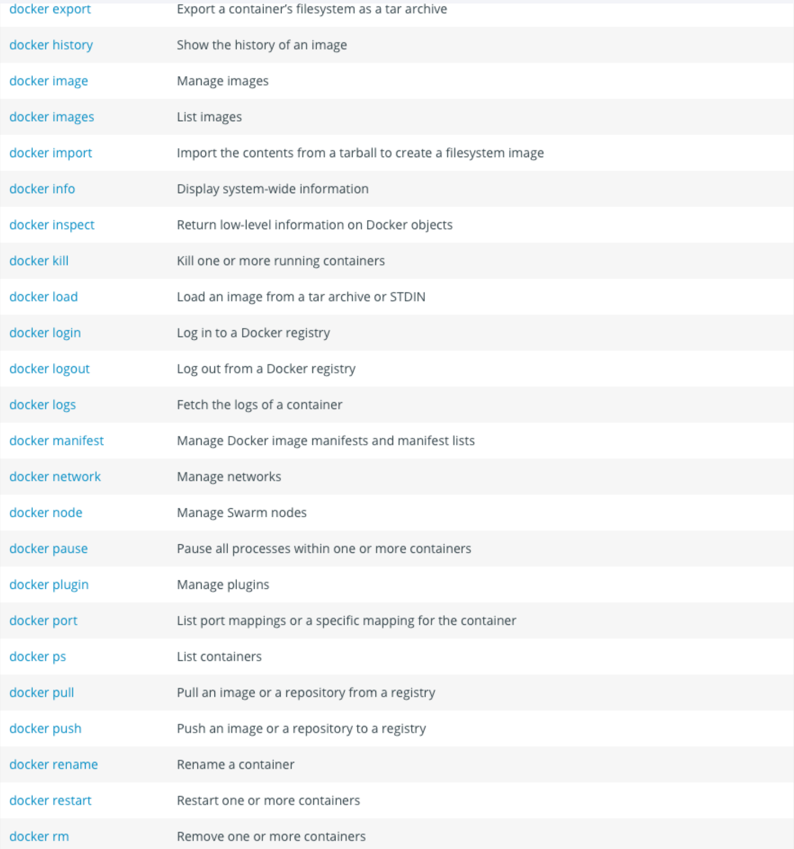
Estimated reading time: 3 minutes

Description

The base command for the Docker CLI.

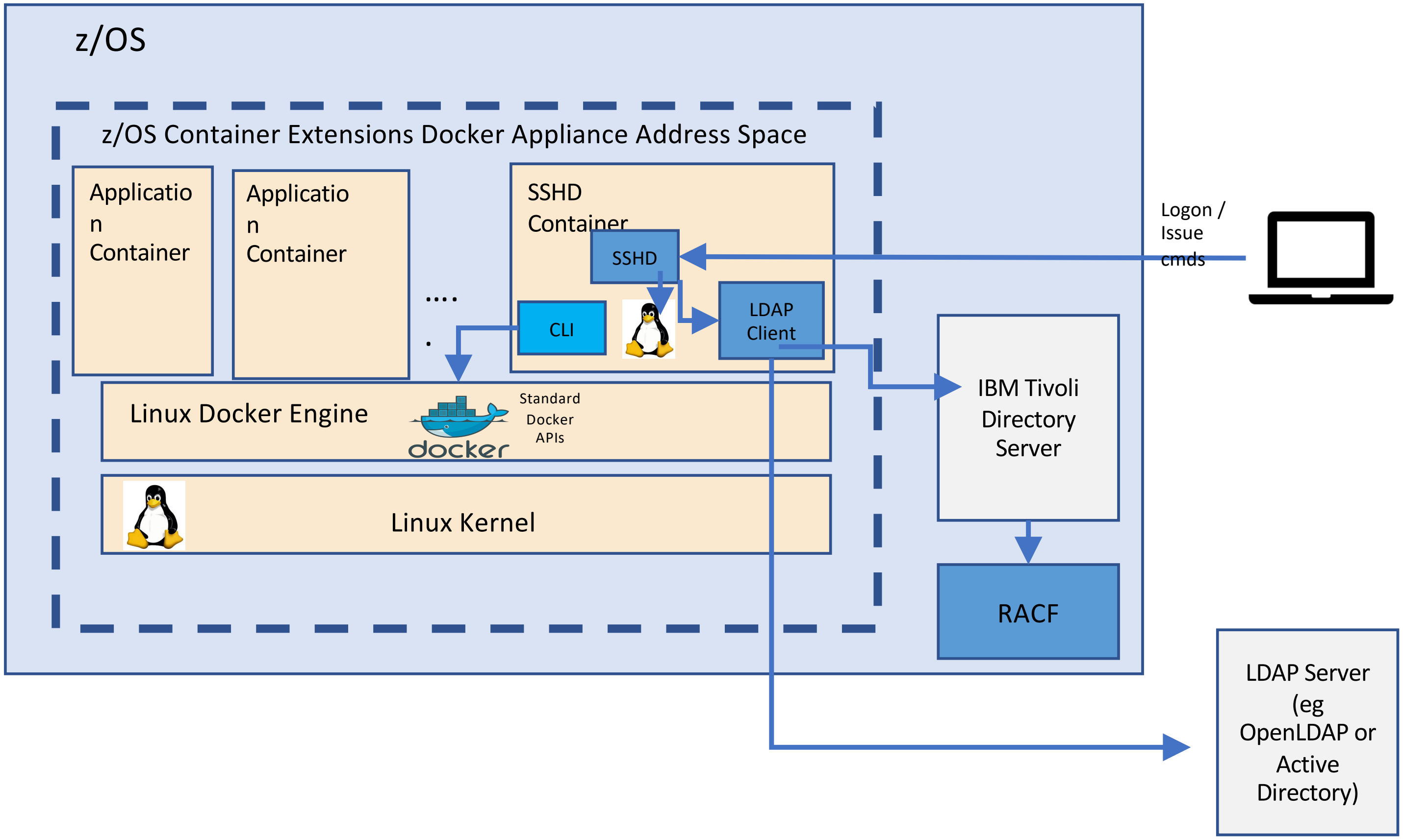
Child commands

Command	Description
docker attach	Attach local standard input, output, and error streams tc
docker build	Build an image from a Dockerfile
docker builder	Manage builds
docker checkpoint	Manage checkpoints
docker commit	Create a new image from a container's changes
docker config	Manage Docker configs
docker container	Manage containers
docker cp	Copy files/folders between a container and the local files
docker create	Create a new container
docker deploy	Deploy a new stack or update an existing stack
docker diff	Inspect changes to files or directories on a container's fil
docker engine	Manage the docker engine
docker events	Get real time events from the server
docker exec	Run a command in a running container
docker export	Export a container's filesystem as a tar archive
docker history	Show the history of an image



docker export	Export a container's filesystem as a tar archive
docker history	Show the history of an image
docker image	Manage images
docker images	List images
docker import	Import the contents from a tarball to create a filesystem image
docker info	Display system-wide information
docker inspect	Return low-level information on Docker objects
docker kill	Kill one or more running containers
docker load	Load an image from a tar archive or STDIN
docker login	Log in to a Docker registry
docker logout	Log out from a Docker registry
docker logs	Fetch the logs of a container
docker manifest	Manage Docker image manifests and manifest lists
docker network	Manage networks
docker node	Manage Swarm nodes
docker pause	Pause all processes within one or more containers
docker plugin	Manage plugins
docker port	List port mappings or a specific mapping for the container
docker ps	List containers
docker pull	Pull an image or a repository from a registry
docker push	Push an image or a repository to a registry
docker rename	Rename a container
docker restart	Restart one or more containers
docker rm	Remove one or more containers

User Management and Authentication



3 Options for User management and authentication:

1. **Local appliance registry (SSH keys or password)**
2. **z/OS LDAP Server (IBM Tivoli Directory Server) with RACF integration**
3. **Remote LDAP server (e.g. OpenLDAP, Active Directory, etc.)**

The Experiences Today

DISCOVER, TRY, BUY
How do I get it?

GET STARTED
How do I get value?

EVERYDAY USE
How do I get my job done?

MANAGE AND UPGRADE
How do I keep it running?

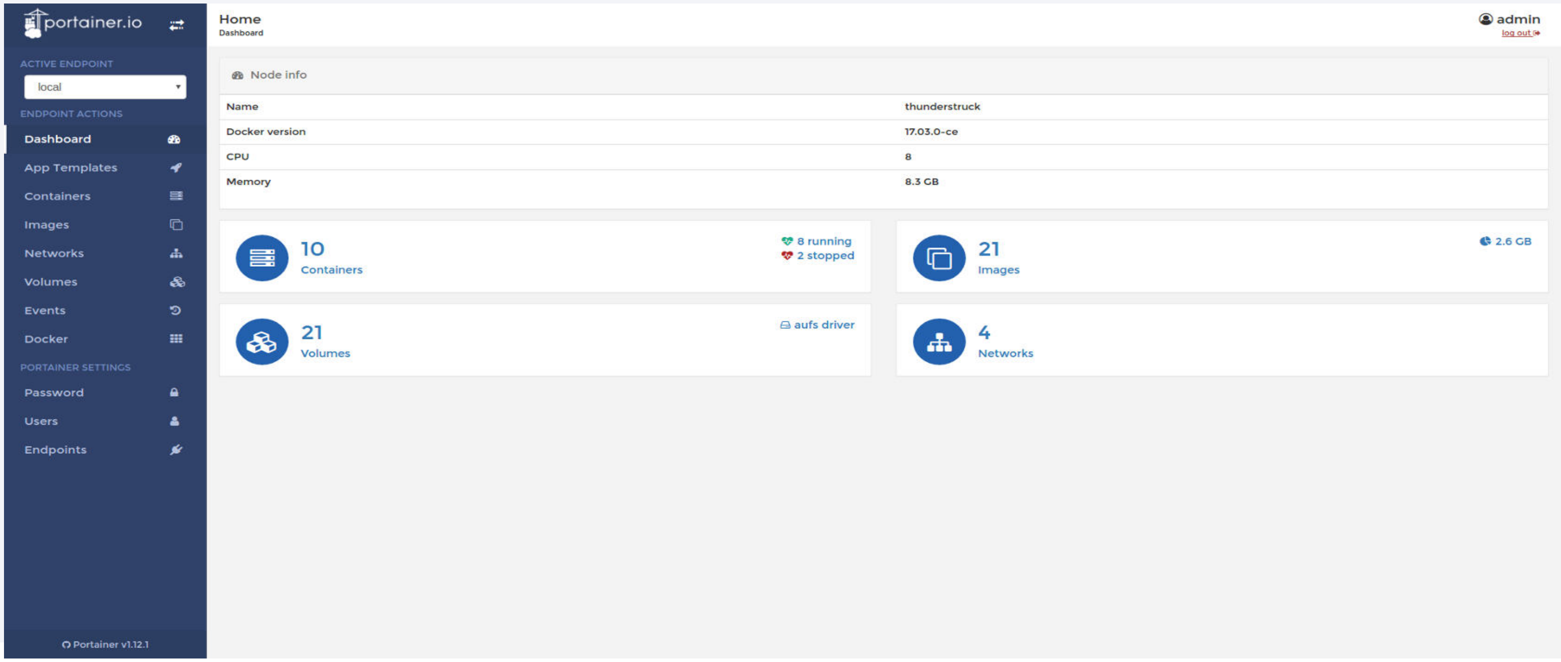
LEVERAGE AND EXTEND
How do I build on it?

SUPPORT
How do I get unstuck?

Graphical user interface access to Docker

z/OS Container Extensions Docker Administrators can deploy Portainer Daemon container for s390x (from Dockerhub) as an additional or alternative interface to the Docker CLI for specific Docker users

Permitted Portainer users can use the graphical interface to deploy and manage Docker containers in a z/OS Container Extensions instance



Node info

Name	thunderstruck
Docker version	17.03.0-ce
CPU	8
Memory	8.3 GB

10 Containers — 8 running, 2 stopped

21 Images — 2.6 GB

21 Volumes — aufs driver

4 Networks

The Experiences Today

DISCOVER, TRY, BUY
How do I get it?

GET STARTED
How do I get value?

EVERYDAY USE
How do I get my job done?

MANAGE AND UPGRADE
How do I keep it running?

LEVERAGE AND EXTEND
How do I build on it?

SUPPORT
How do I get unstuck?

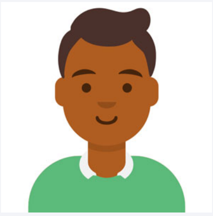
Monitoring z/OS Container Extensions instances

Docker administrators can deploy and use open source and ISV Docker Container images for Linux on Z (s390x images) to monitor overall server and container resource utilization

Examples of Open Source Docker images tested with z/OS Container Extensions

- Prometheus: Open source monitoring and alerting solution based on time series database
  - Flexible query language
  - System and application level monitoring
  - Collects metrics from instrumented targets
- Grafana: Open source metrics analytics and visualization tool
  - Support for Prometheus as a data source (among others)
  - Provides easy to build dashboards for visualizing system and application metrics
- cAdvisor: Monitors container based environments
  - Collects metrics at container and system level
  - Can act as a data source for Prometheus and provides its own UI
- Prometheus Node Exporter: Acts as a data source for system level metrics for Prometheus

Zach
Systems Programmer



Ramesh
Docker Admin

The Experiences Today

DISCOVER, TRY, BUY
How do I get it?

GET STARTED
How do I get value?

EVERYDAY USE
How do I get my job done?

MANAGE AND UPGRADE
How do I keep it running?

LEVERAGE AND EXTEND
How do I build on it?

SUPPORT
How do I get unstuck?

Clustering and Orchestration

Permitted z/OS Container Extensions Docker users create a Swarm cluster of z/OS Container Extensions instances using standard Docker CLI

Permitted z/OS Container Extensions Docker users can deploy Docker containers in a z/OS Container Extensions Swarm cluster using standard Docker CLI

Future support:

- Kubernetes clustering
- Statement of Direction issued on 5/14/2019

Shichi
IT Architect

Omar
Solution Architect

Zach
Systems Programmer

Ramesh
Docker Admin

Fred
Application Developer

Other resources

YouTube

__Getting Started videos:__

Resource Planning for zCX:

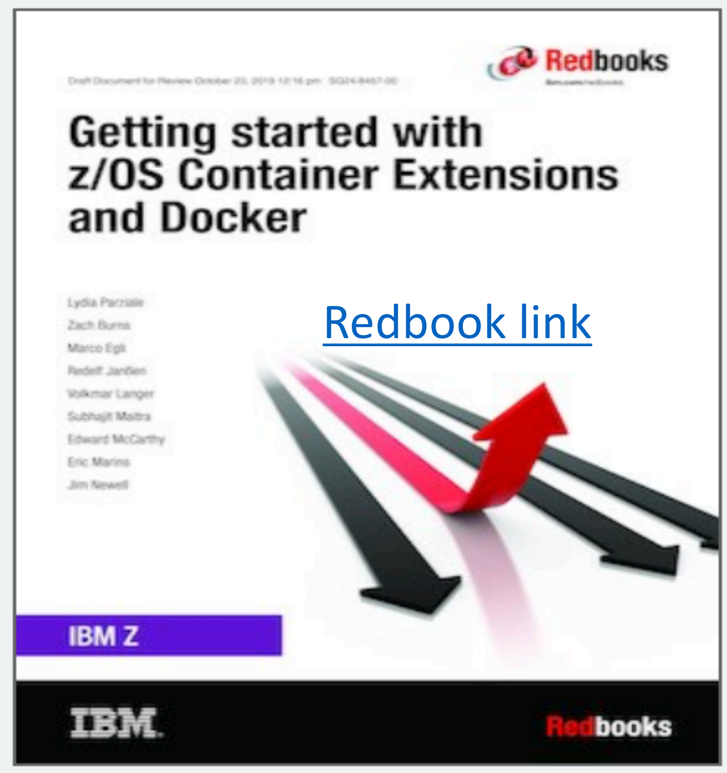
https://www.youtube.com/watch?v=5o1r2EPMMUc

Provisioning zCX using z/OSMF workflows:

https://www.youtube.com/watch?v=CPeI5KmoAw0

Getting started with Docker in zCX:

https://www.youtube.com/watch?v=9aYFzhvJVb



Redbooks

Draft Document for Review October 23, 2019 12:16 pm SG24-8457-00

Getting started with z/OS Container Extensions and Docker

Lydia Parziale
Zach Burns
Marco Egli
Redelf Janßen
Volkmar Langer
Subhajit Maitra
Edward McCarthy
Eric Marins
Jim Newell

IBM Z

IBM

Redbooks

Redbook link

OPEN **MAINFRAME** PROJECT

Ambitus

A project to enable open source software deployment on z/OS*

https://github.com/ambitus/linux-containers

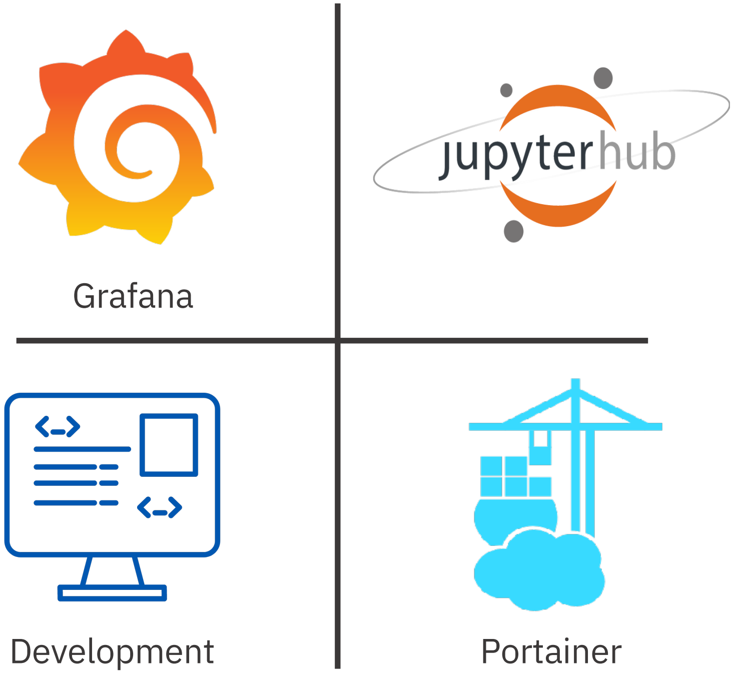
Help Linux architects and developers understand z/OS

Enable system programmers to provision resources for open source deployments

Both personas operate in the environments they understand

Samples for building common Docker images

How-to information to help with container deployment



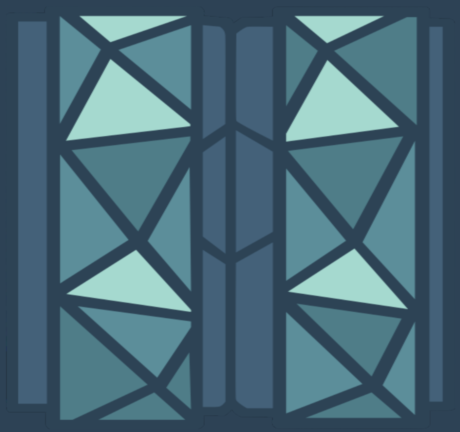
**Ambitus is Latin for "compass"*

Modernize and Extend your z/OS® Applications with

IBM z/OS® Container Extensions(zCX)

Resource	Link
Content Solutions Page	http://ibm.biz/zOSContainerExtensions
Open Z Systems Exchange	http://ibm.biz/openzsx
zCX FAQ	http://ibm.biz/zcx_FAQ

Thank you!



Backup