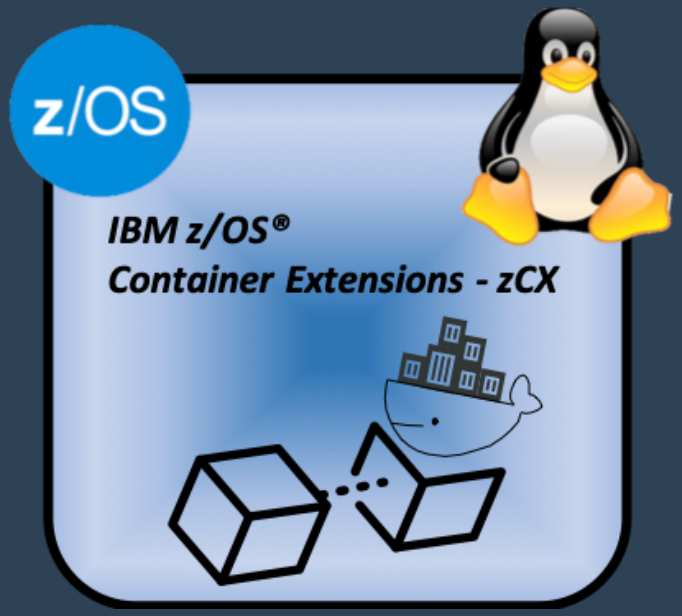
z/OS V2.4 Preview: z/OS Container Extensions

Running Linux on Z Docker containers inside z/OS



Gus Kassimis
IBM Distinguished Engineer
kassimis@us.ibm.com

Trademarks

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

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

* 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.

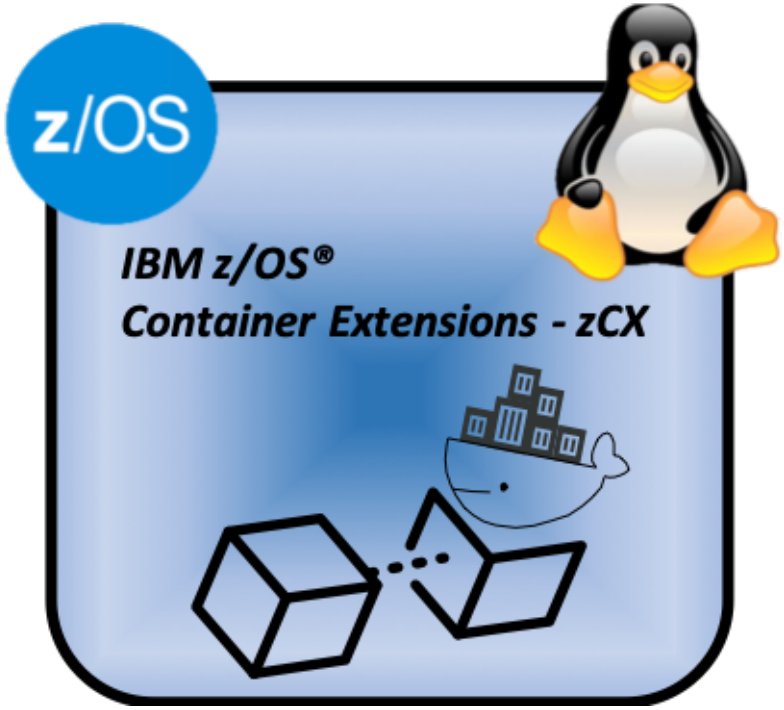
Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products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

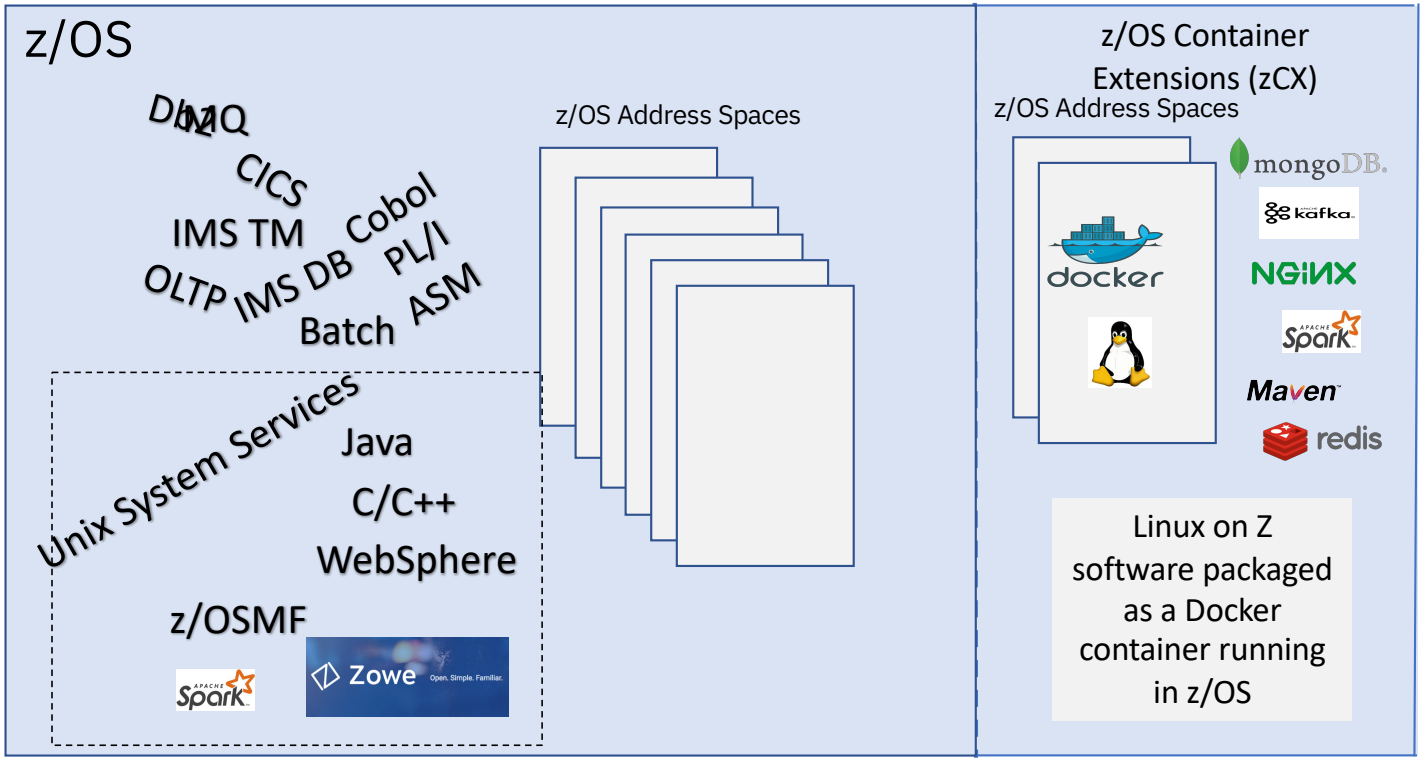
This information provides only general descriptions of the types and portions of workloads that are eligible for execution on Specialty Engines (e.g, zIIPs, zAAPs, and IFLs) ("SEs"). IBM authorizes customers to use IBM SE only to execute the processing of Eligible Workloads of specific Programs expressly 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.

Agenda

- *What is z/OS Container Extensions (zCX)?*
- *What does it enable you to do?*
- *How to I get started with zCX?*
- *How do I manage and monitor zCX*



Expanding the z/OS Software Ecosystem



- 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

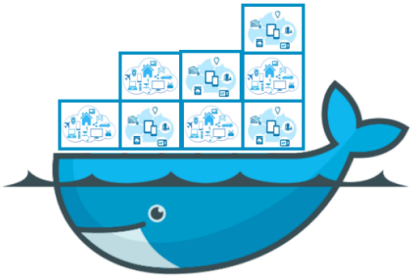
✓ Requires IBM z14 (or later) based server

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**.

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



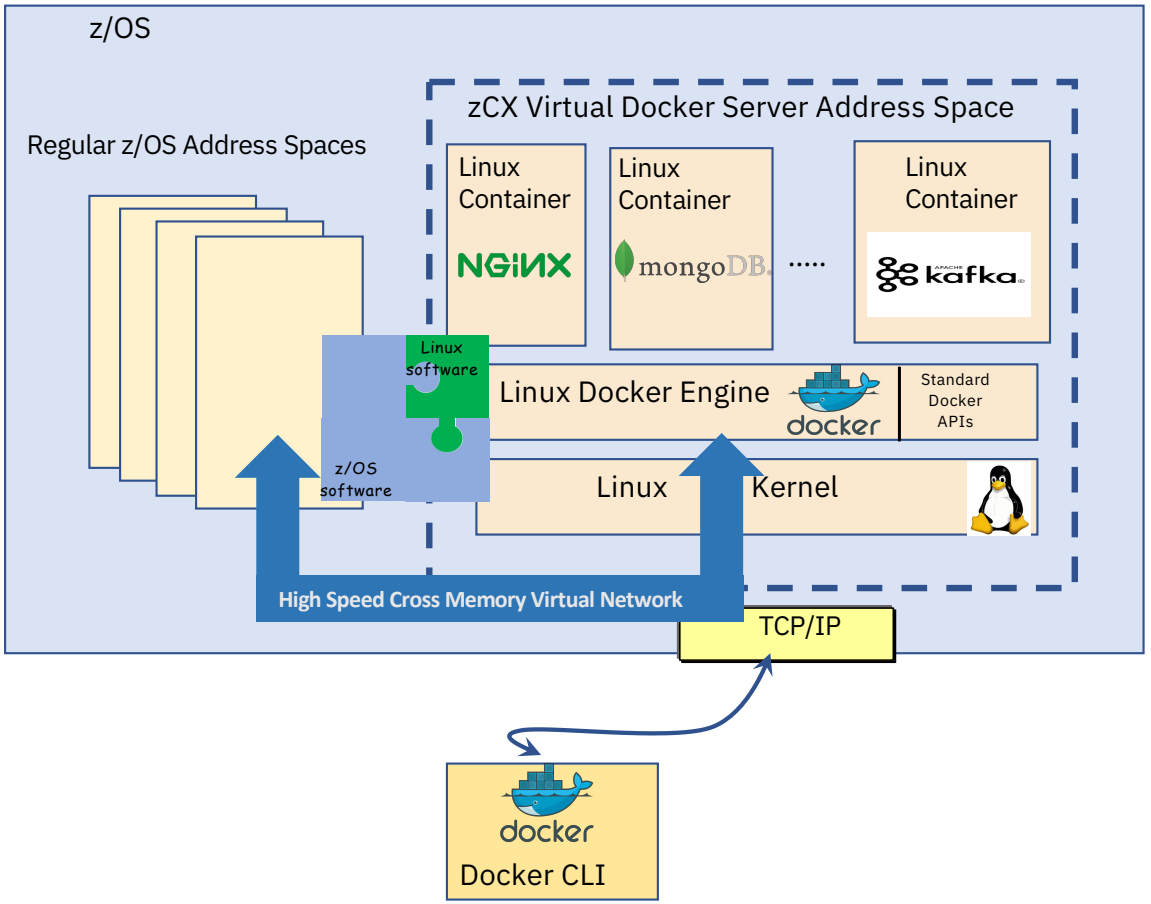
zCX – A turn-key Virtual Docker Server Software Appliance

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 - Goals & Qualities of Service

Integrated Disaster Recovery & Planned Outage Coordination	z/OS Storage Resilience	z/OS Networking Virtualization, Security & Availability	z/OS Workload Management, Capacity Planning & Chargeback
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.	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 zHyperLink and future z/OS Storage enhancements	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)	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)
- 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 – Examples:
  - Tivoli Enterprise Portal (TEPS)
  - Service Management Unite (SMU)

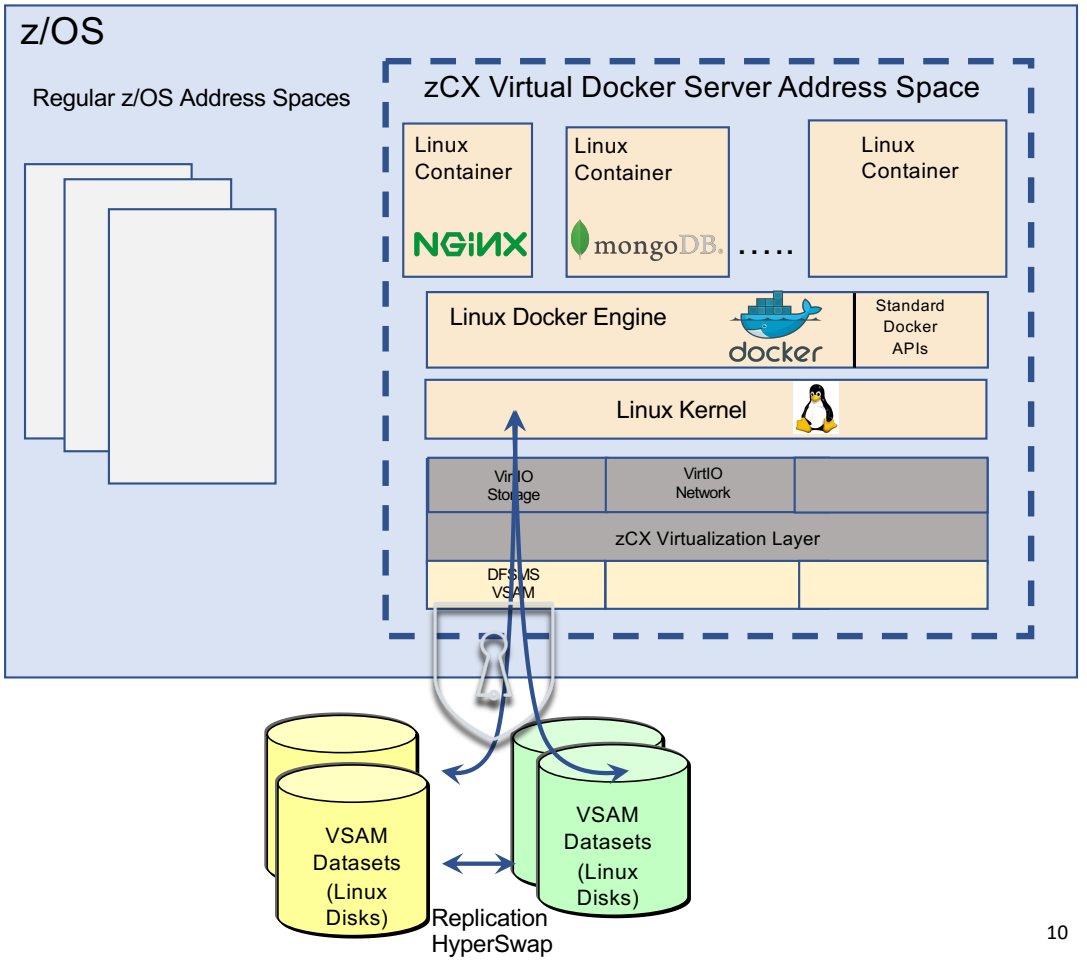
Open Source Application Development Utilities

- 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

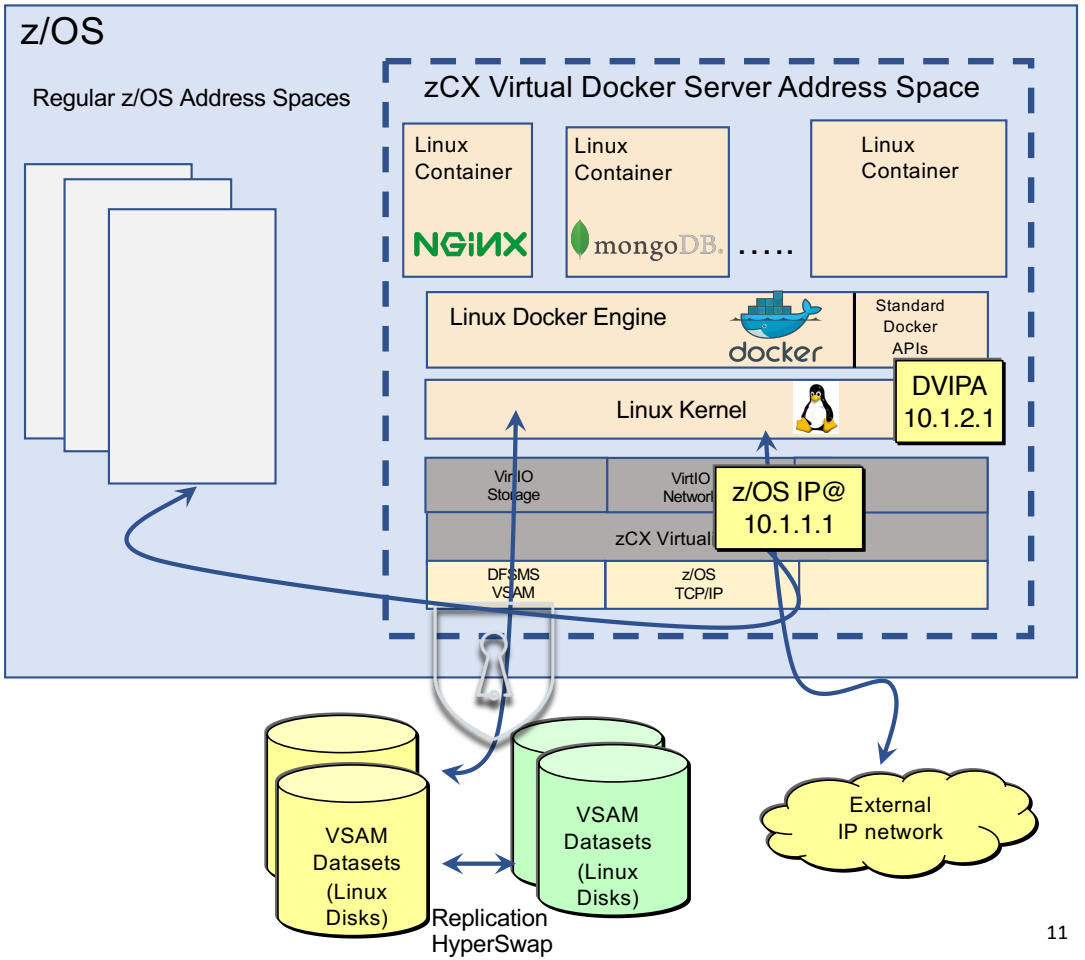
IBM zCX – z/OS Storage Integration

- 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 (e.g. zHyperLinks, I/O fabric diagnostics, etc.)
  - Built-in host-based encryption
  - Replication technologies and HyperSwap



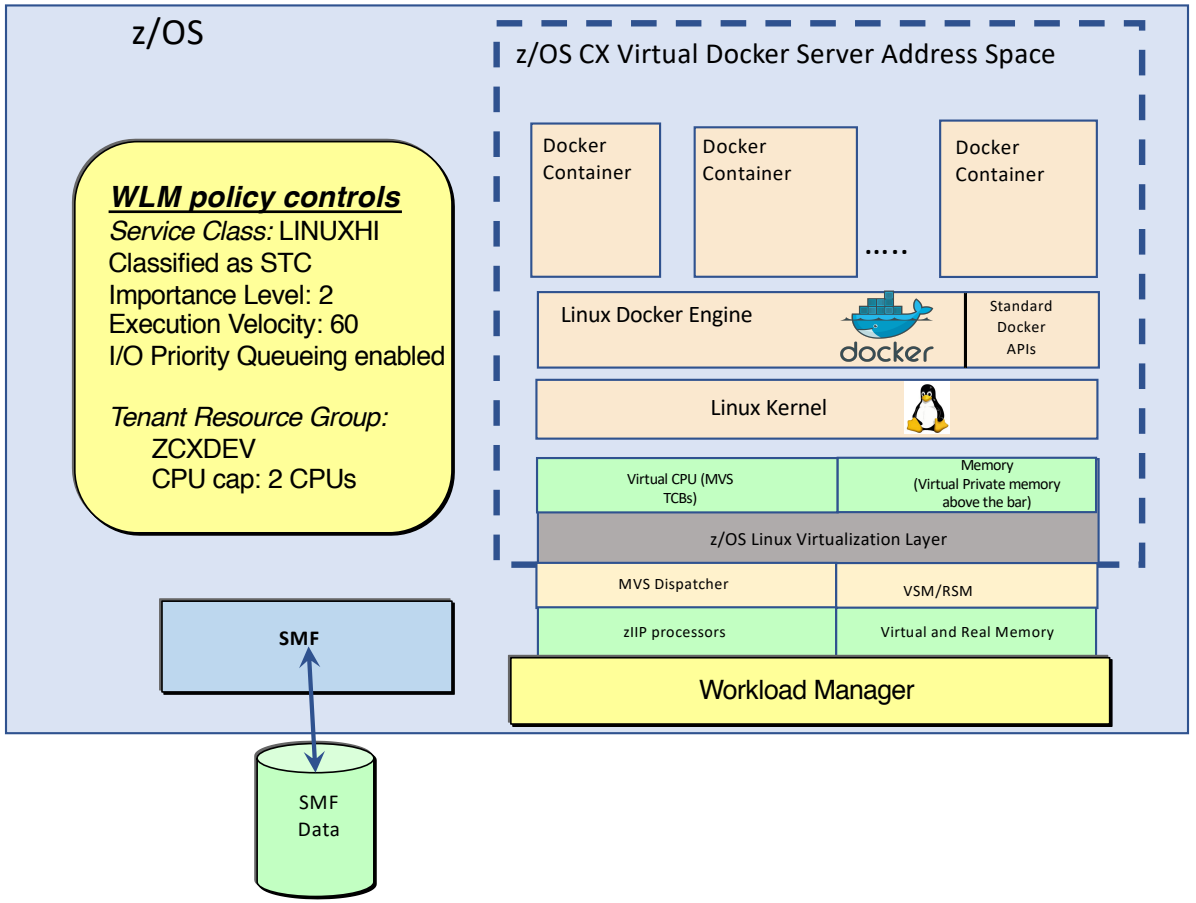
IBM zCX – z/OS Network Integration

- 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
  - External network access via z/OS TCP/IP
    - z/OS IP filters to restrict external access



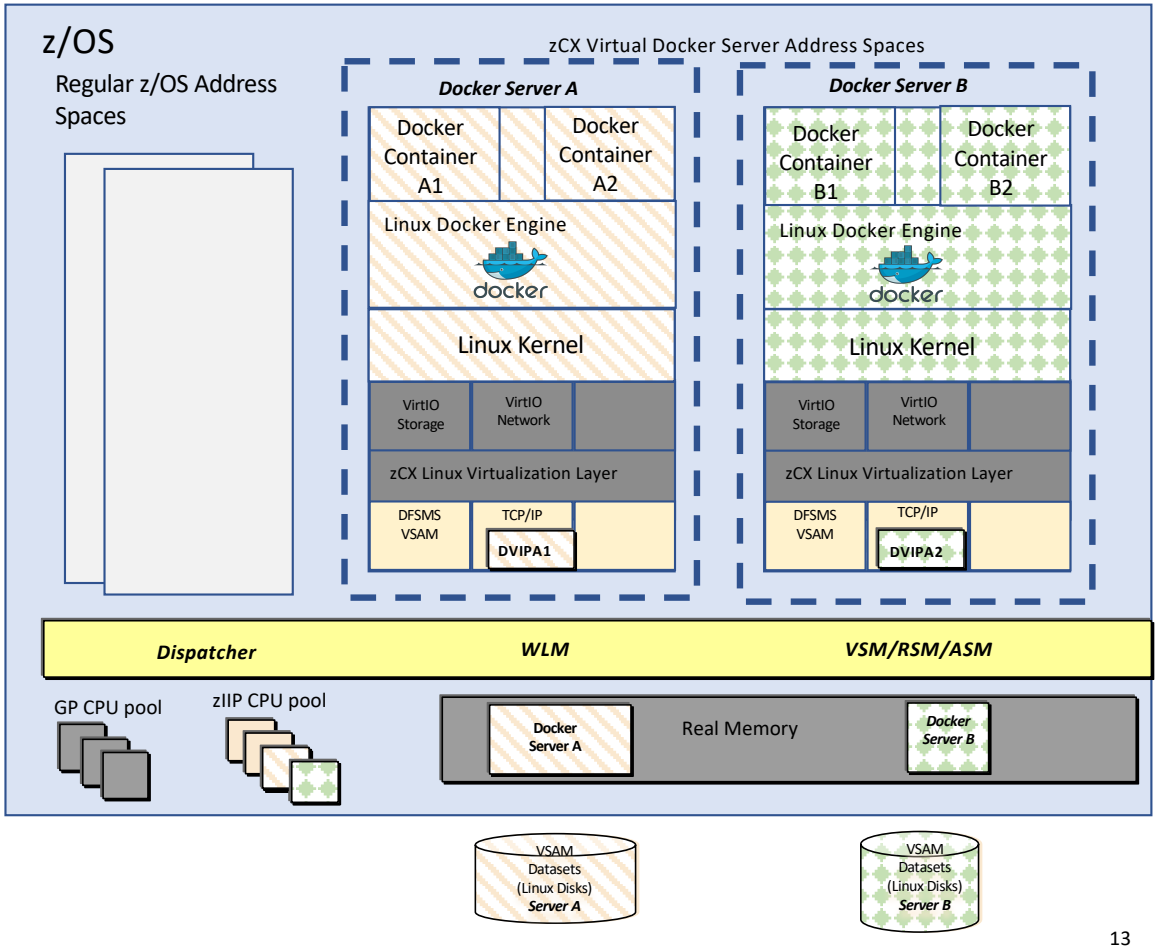
IBM zCX - CPU, Memory and Workload Management

- 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



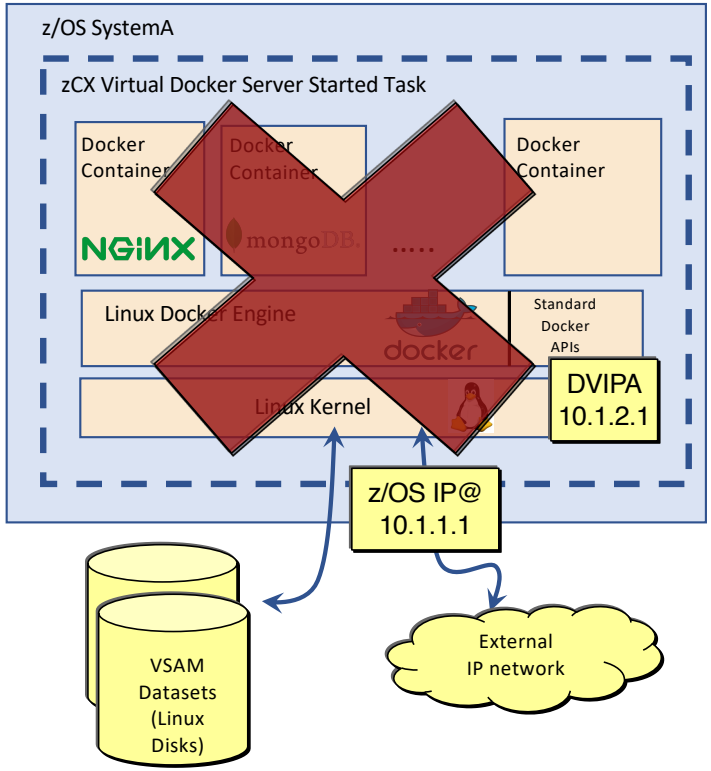
Deploying Multiple zCX Virtual Docker Server Instances

- 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



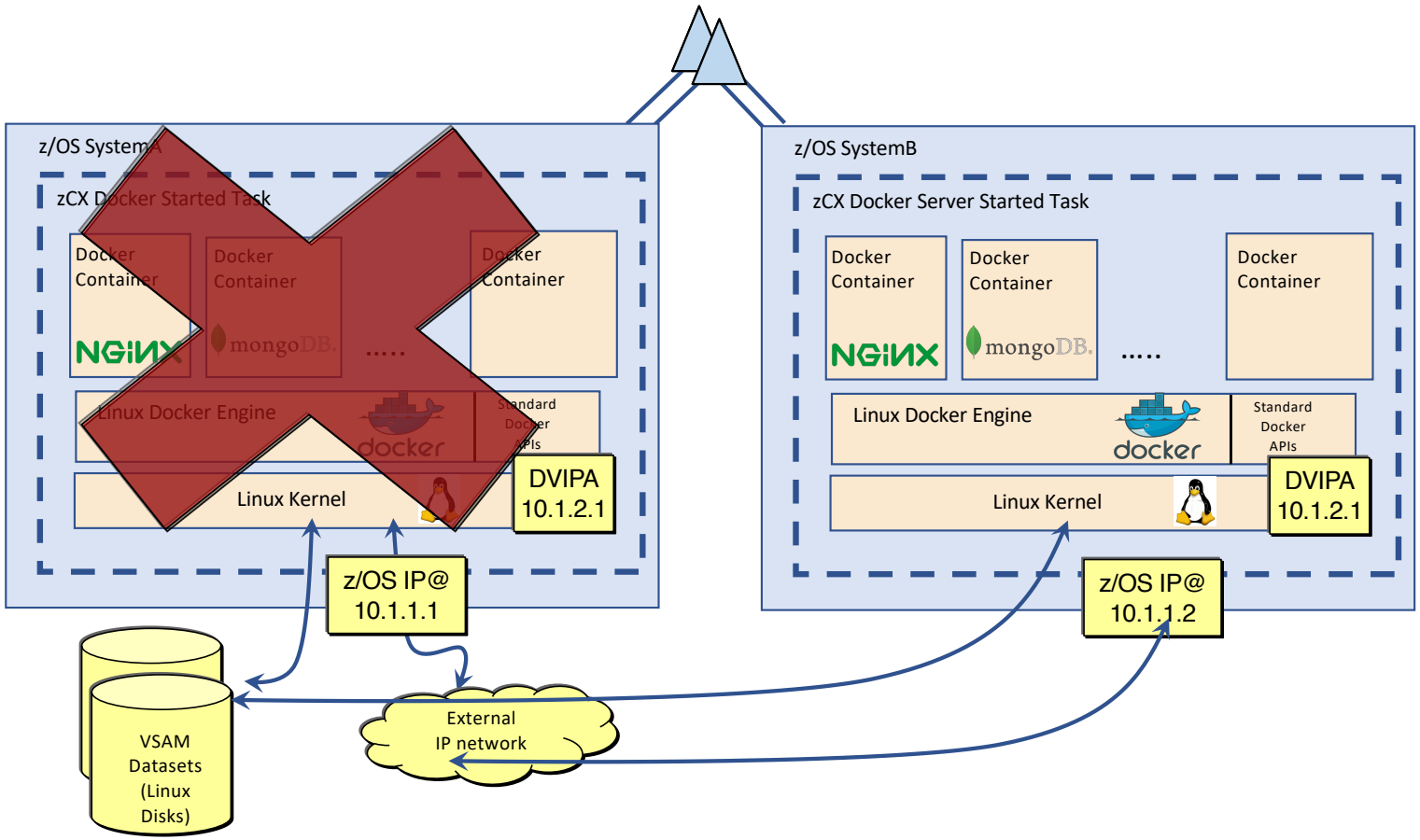
z/OS Container Extensions
Operations and Disaster Recovery Integration

- 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)



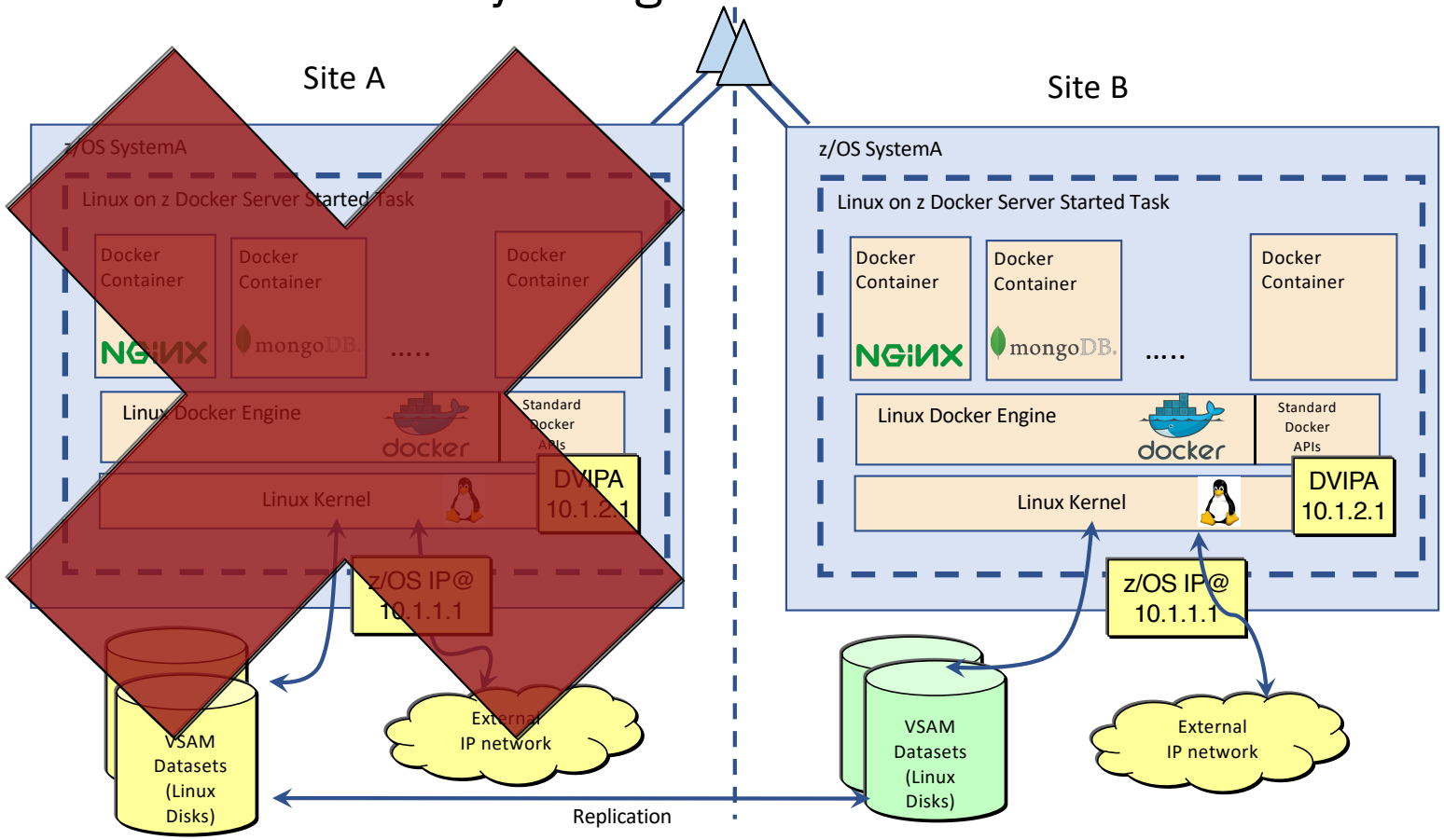
z/OS Container Extensions
Operations and Disaster Recovery Integration

- 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)

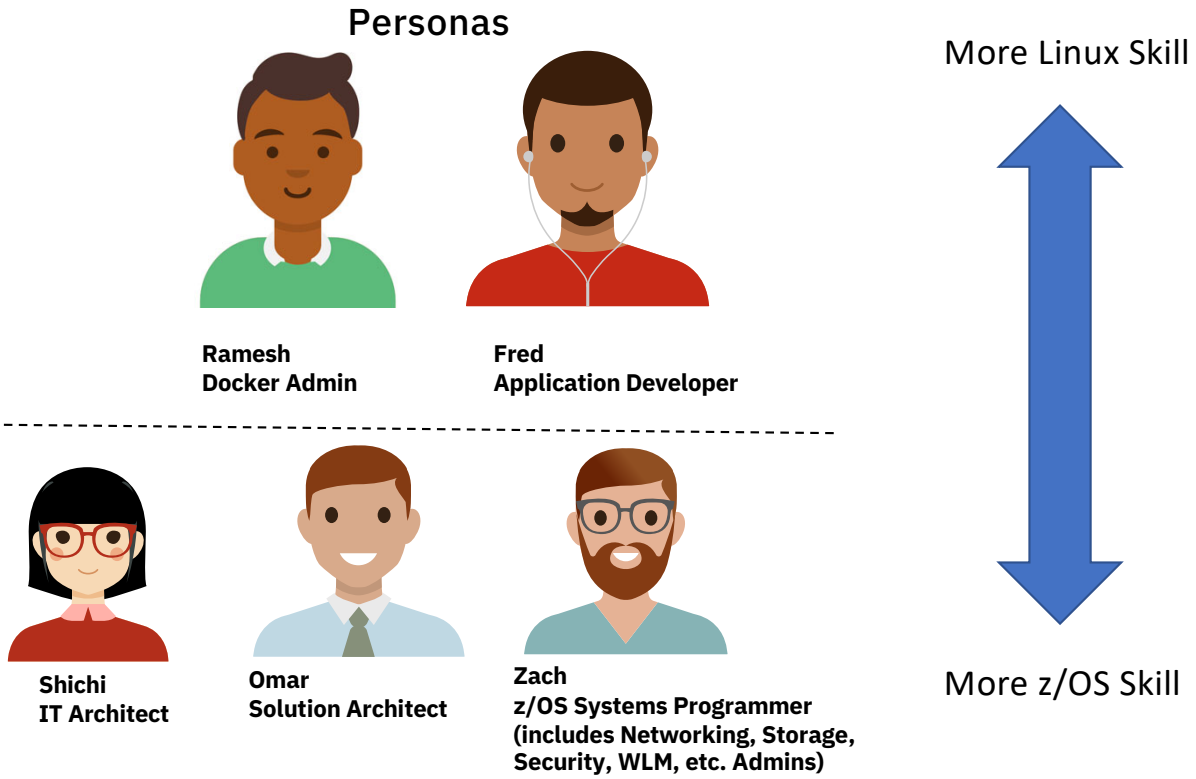


z/OS Container Extensions
Operations and Disaster Recovery Integration

- 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



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?

Provisioning

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

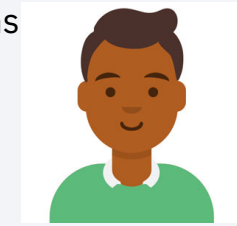
- Resource allocation
  - Number of virtual CPUs, memory, network connectivity and storage
- Docker Configuration settings
- Definition of z/OS Container Extensions appliance admin user and Docker admin user

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)



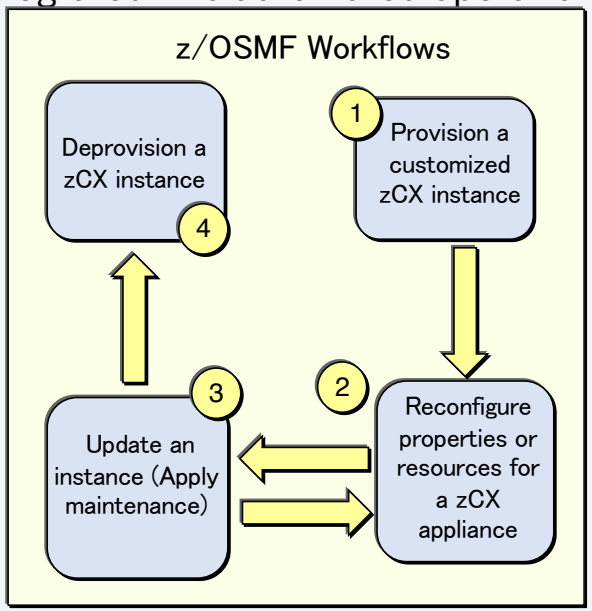
Ramesh
Docker Admin

Zach
Systems Programmer

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



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

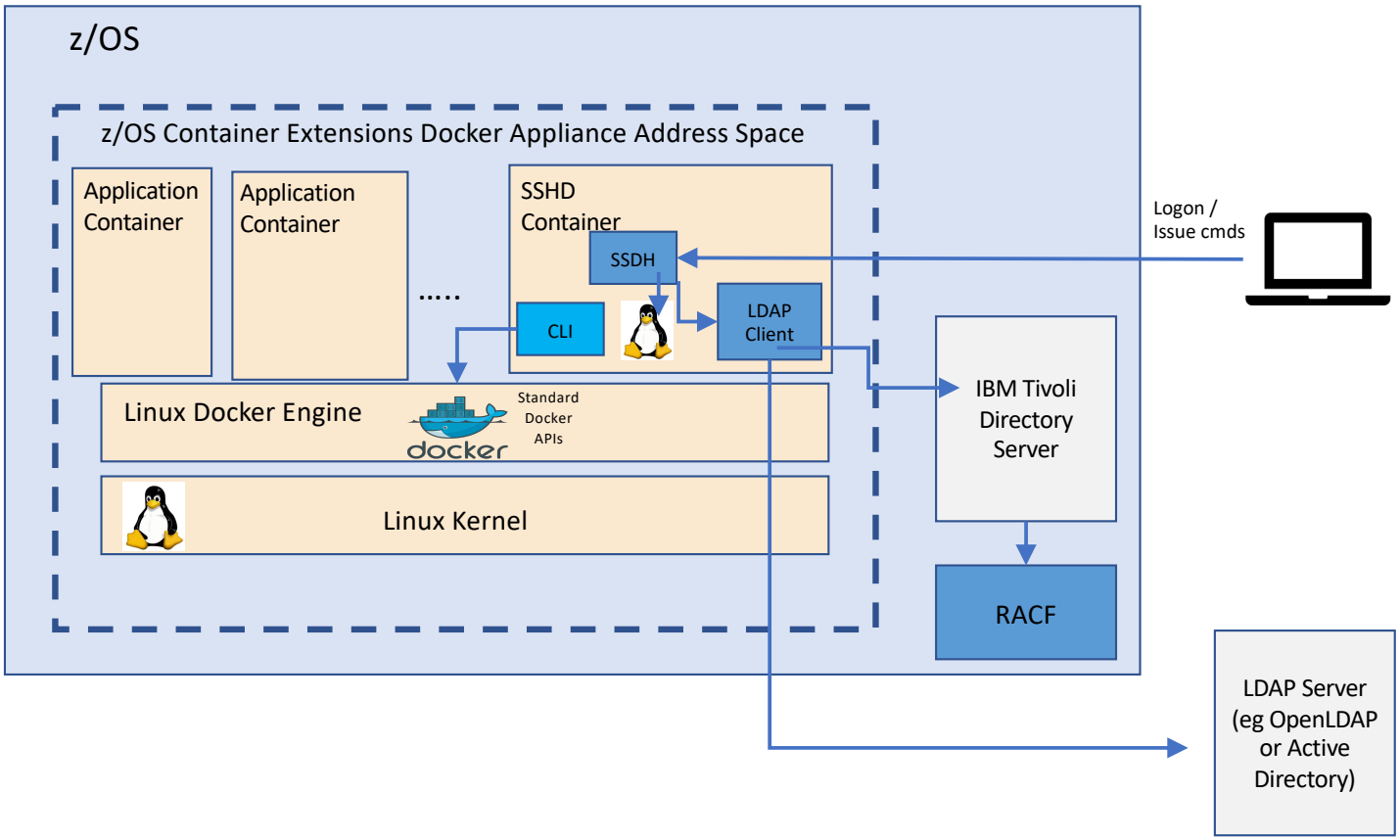
Description

The base command for the Docker CLI.

Child commands

Command	Description
docker attach	Attach local standard input, output, and error streams to
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 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
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?

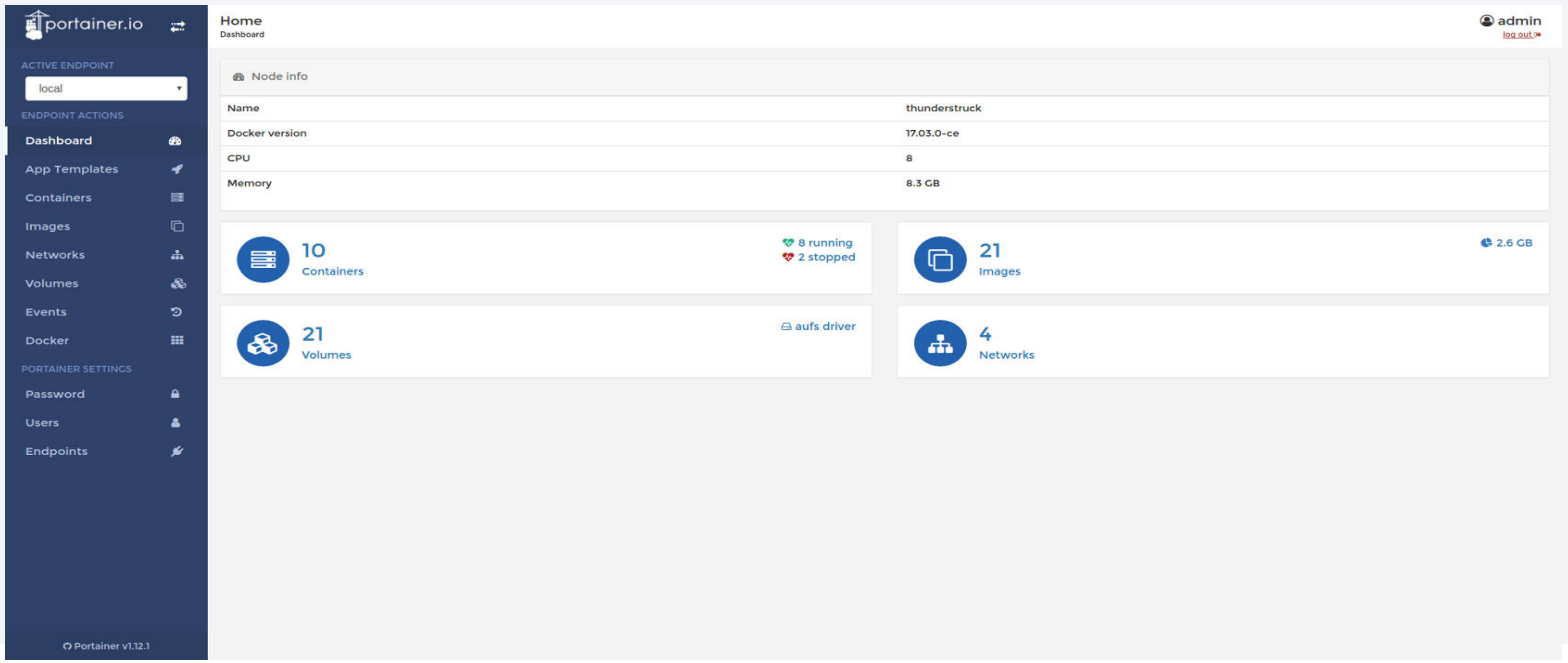
SUPPORT
How do I get unstuck?

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

portainer.io

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



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?

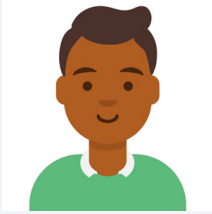
Monitoring z/OS Container Extensions instances

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

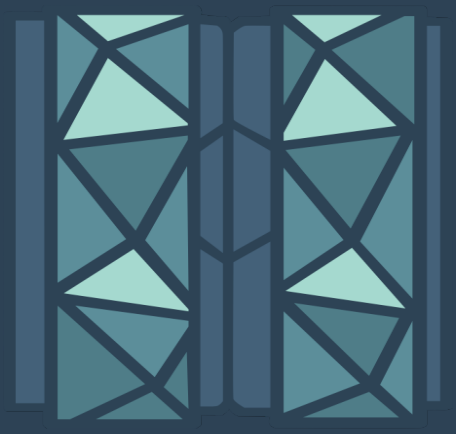
Modernize and Extend your z/OS® Applications with

IBM z/OS® Container Extensions(zCX)

Discover more on our content solutions page by visiting
http://ibm.biz/zOSContainerExtensions

Scan the QR code or visit
http://ibm.biz/zCXSurvey
To take a survey on zCX

Thank you!



Backup

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 and Managing z/OS Container Extensions Address Spaces

System Programmers, Operators and Performance Analysts can use standard available z/OS Monitoring tools and facilities to monitor z/OS Container Extensions address spaces

zCX address spaces are started tasks that have standard address space metrics and SMF records available that can be used for monitoring and chargeback

- Type 30 and type 72 records are available
- Other SMF records related to z/OS Container Extensions address space resource utilization also available

z/OS WLM policies and facilities can be used to dynamically adjust service class goals and importance levels for zCX appliances

Zach
Systems Programmer
And Operations Staff

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?

Diagnosing problems with z/OS Container Extensions

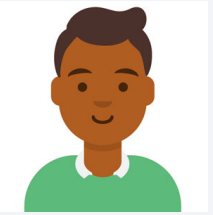
- IBM z/OS support process can be used to help diagnose and address problems with the underlying z/OS Container Extensions implementation, including problems in:
  - Virtualization layer
  - Docker appliance and Linux kernel layers
- Problems with software deployed as containers pursued using existing channels (IBM, ISV, Open Source, etc.)

Facilities provided:

- First Failure Data Capture (FFDC)
- Facilities to allow on demand capture of z/OS Container Extensions appliance or virtualization diagnostic data
- CTRACE and z/OS Dump support for virtualization layer
- Ability to extract z/OS Container Extensions appliance diagnostics and dump data

Zach
Systems Programmer

And z/OS Operations staff



Ramesh
Docker Admin