Introduction to z/OS Container Extensions (zCX)

Steve Warren
Senior Technical Staff Member
Architect, IBM Garage for Systems
swarren@us.ibm.com

Jovanna Hadley
zClient Technical Specialist
Advanced Technology Group
jovanna.hadley@ibm.com

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

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 significantly improving network transaction rates and reducing latency
- ✓ While maintaining overall solution operational control within z/OS and with z/OS Qualities of Service
- ✓ Requires IBM z14 (or later) based server with Container Hosting Foundation (feature code 0104)
  - ✓ zCX Trial - Try and Buy capability let's you optionally kick the tires for 90 days

Design Thinking Hill Statement:

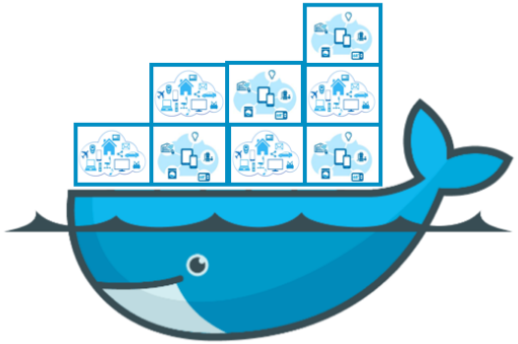
A **solution architect** can **create a solution to be deployed on z/OS based on components available as Docker containers** in the Linux on Z ecosystem transparently exploiting z/OS QoS, **without requiring z/OS development skills**.

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



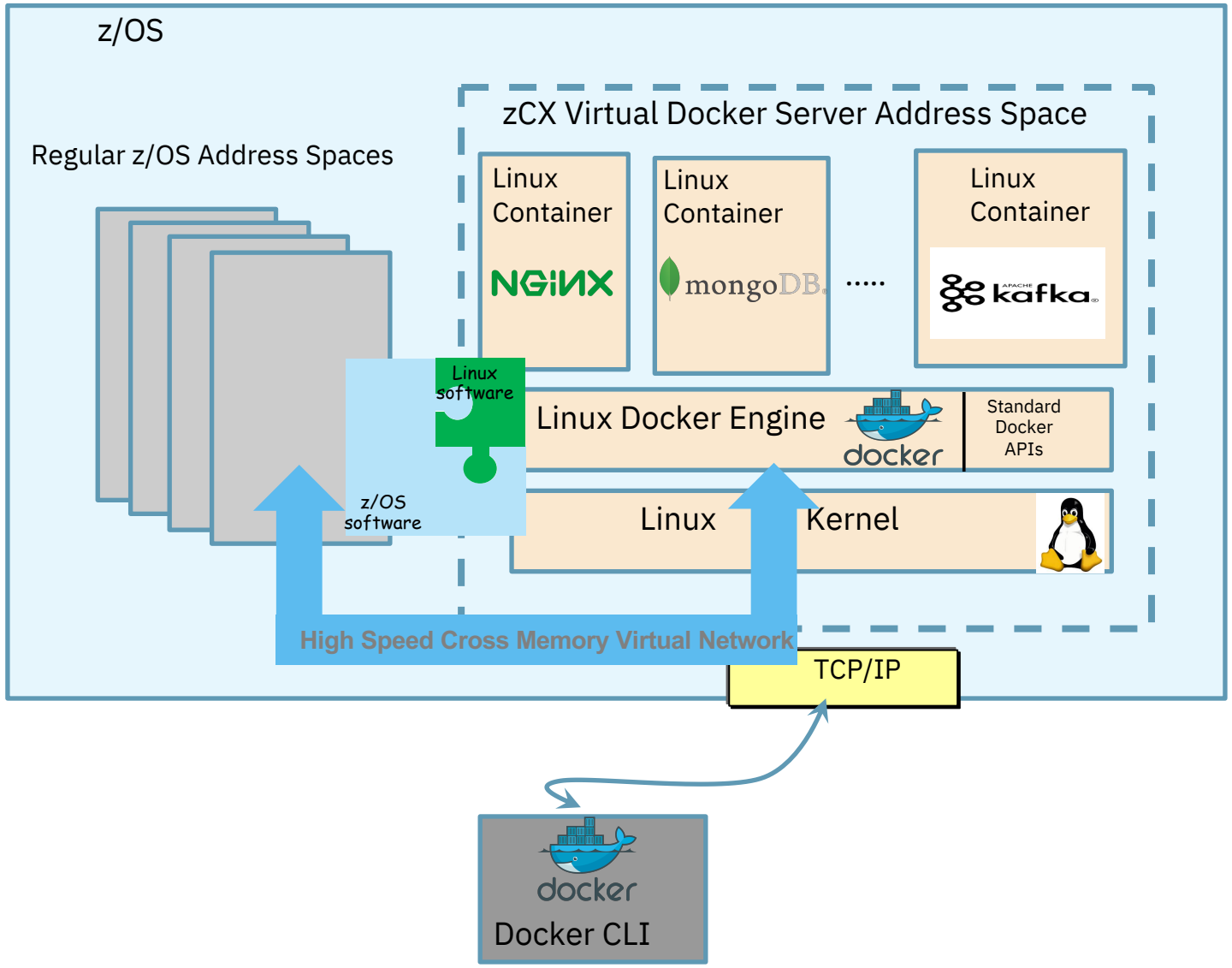
zCX – A turn-key Virtual Docker Server Software Appliance

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
  - Running the Acme Air benchmark on zCX, up to 98% of the zCX CPU consumption was measured to be zIIP eligible.*



** * Results were extrapolated from internal IBM benchmarks performed in a controlled environment using a single z14 z/OS 2.4 LPAR with TCP/IP inbound workload queuing (IWQ) for inbound traffic and two zCX containers: one running Node.js and one running a MongoDB database. zIIP eligibility is based on the CPU consumption of the work running on the zCX address spaces and the associated work on the TCPIP and VTAM address spaces. Results may vary.

zCX – Goals and 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, IBM MQ Client Concentrator)
- 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.

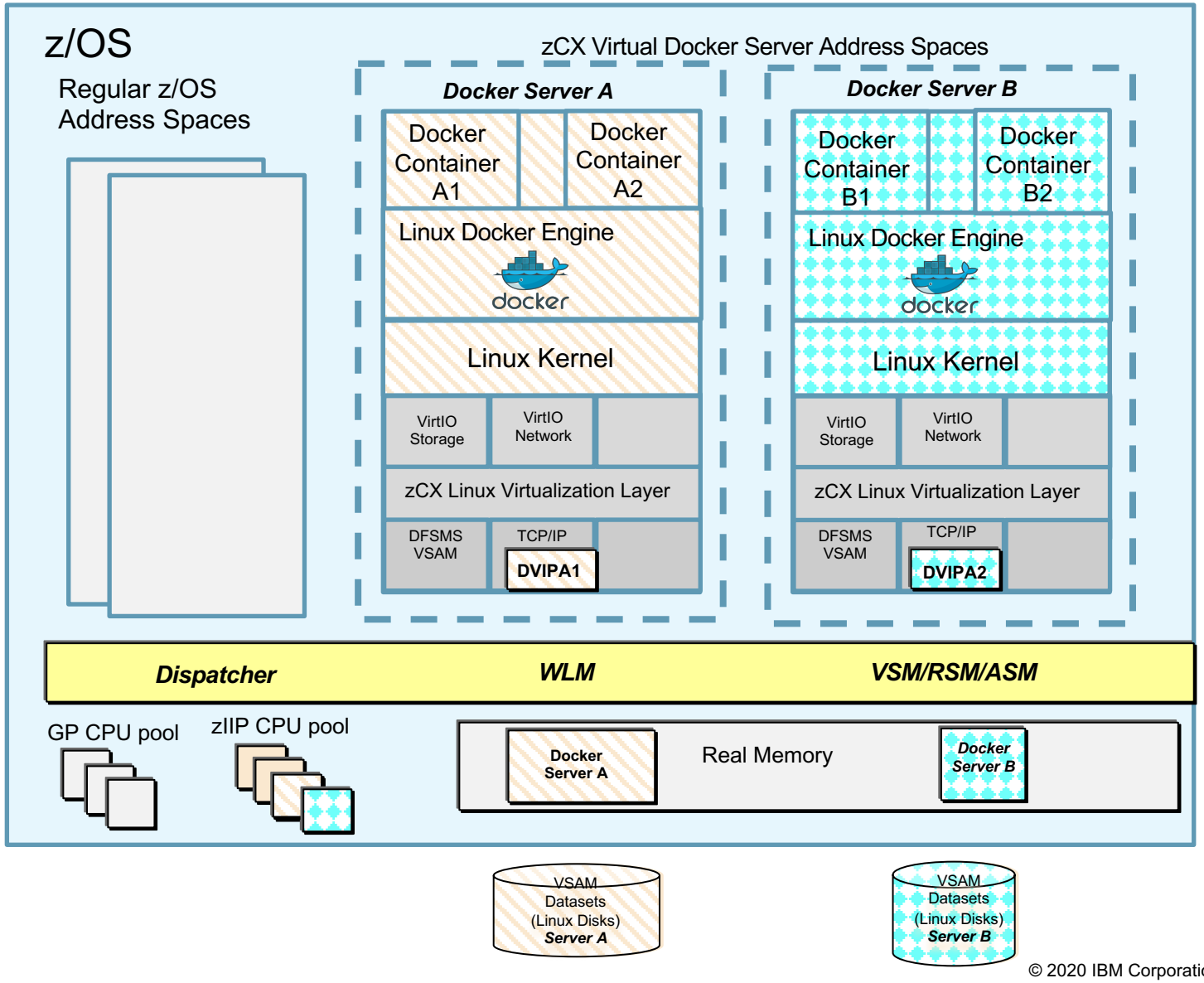
Open Source Application Development Utilities

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

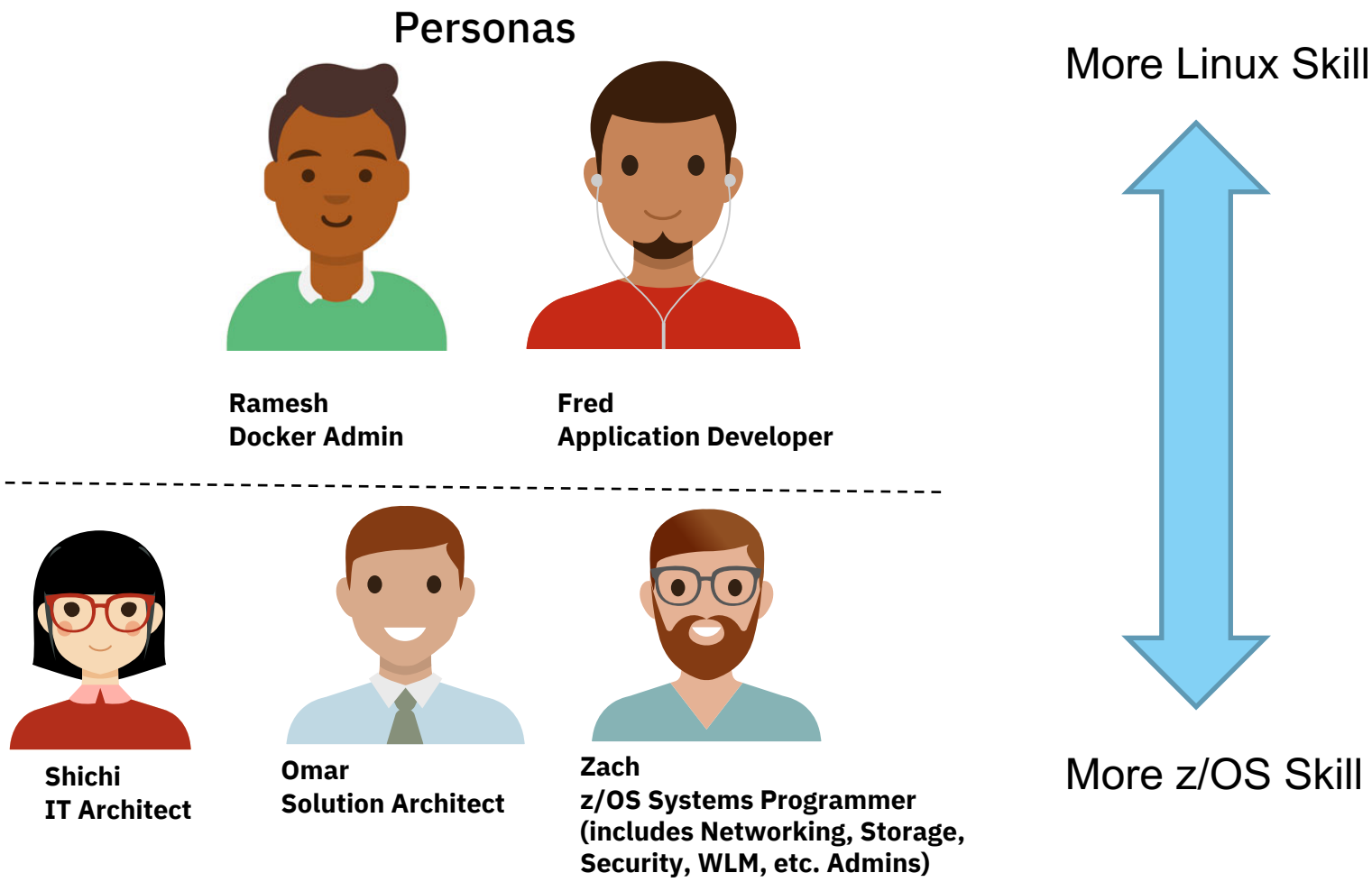
Note: The use cases depicted reflect the types of software that could be deployed in IBM zCX in the future. They are not a commitment or statement of software availability for IBM zCX

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



Personas



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), at least 2 GB minimum
- Support for Dynamic VIPA (DVIPA support)
- z/OS VSAM LDS for storage with support for encryption and replication

Zach
Systems Programmer

Docker Configuration Options:

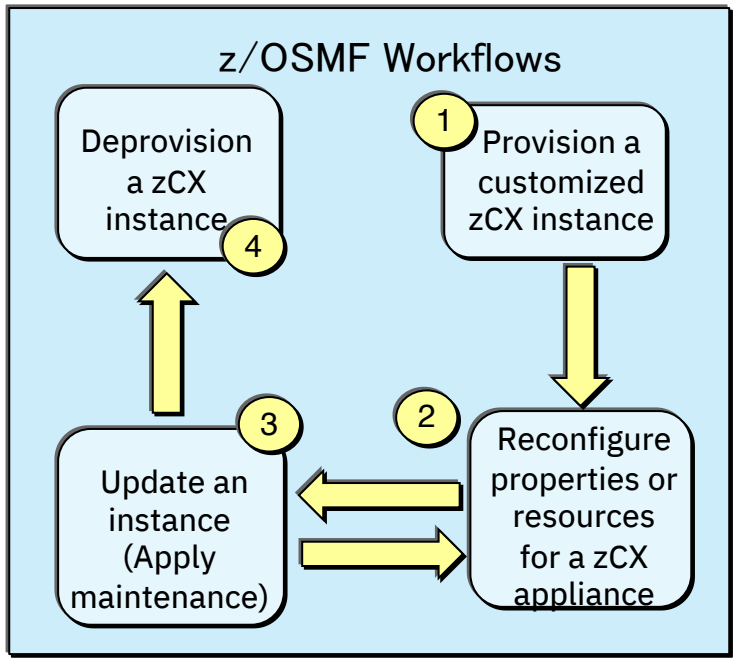
- Registry to be used
- Logging options
- Other

Ramesh
Docker Admin

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



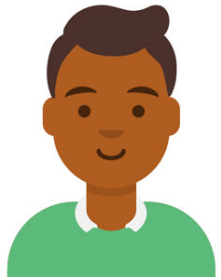
Zach
Systems Programmer

Docker Interface

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 is not supported

Zach
Systems Programmer



Ramesh
Docker Admin

Fred
Application Developer

Omar
Solution Architect

Docker Interface (continued)

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

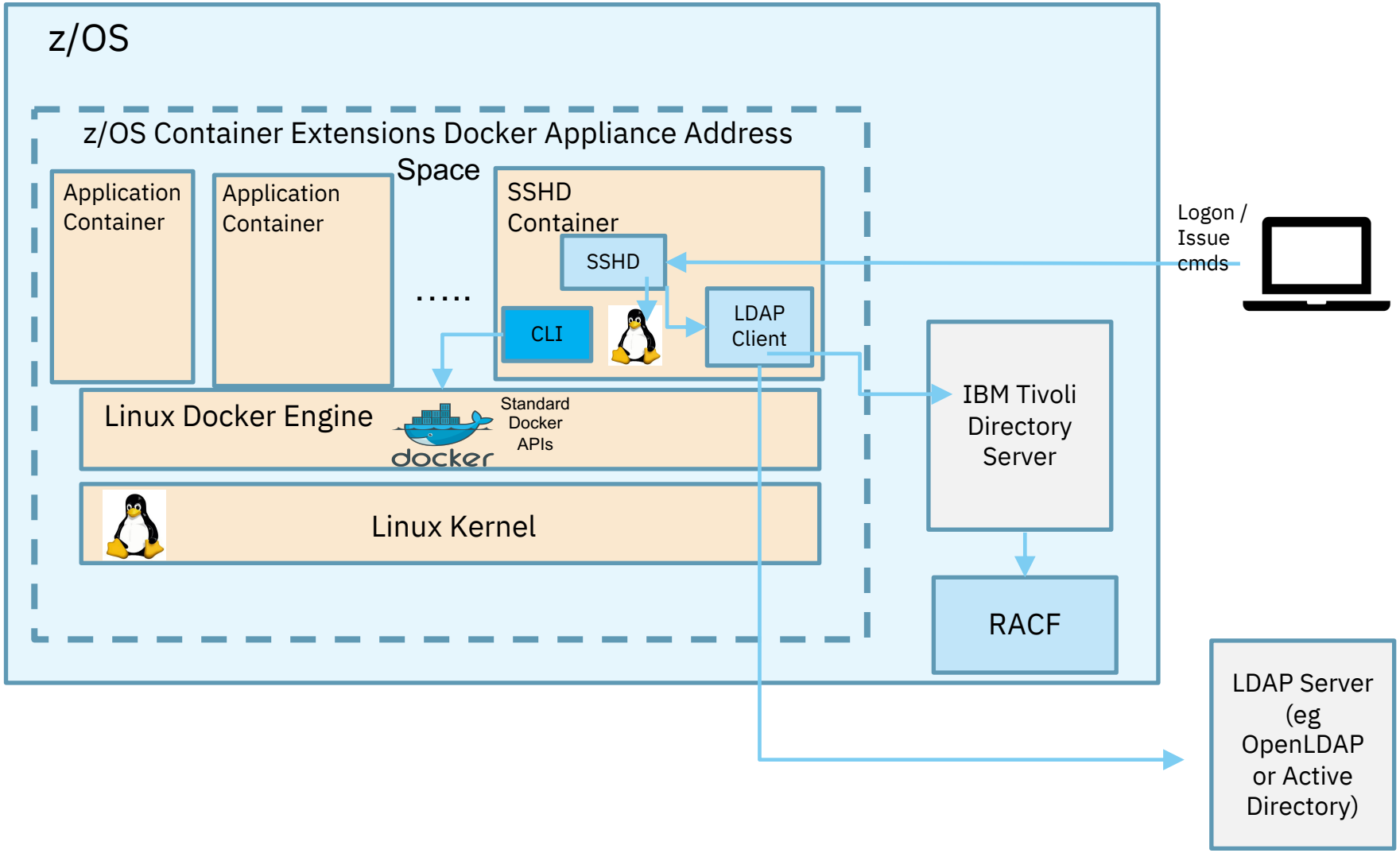
Description

The base command for the Docker CLI.

Child commands

Command	Description
docker attach	Attach local standard input, output, and error streams to a running container
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 filesystem
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 filesystem
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

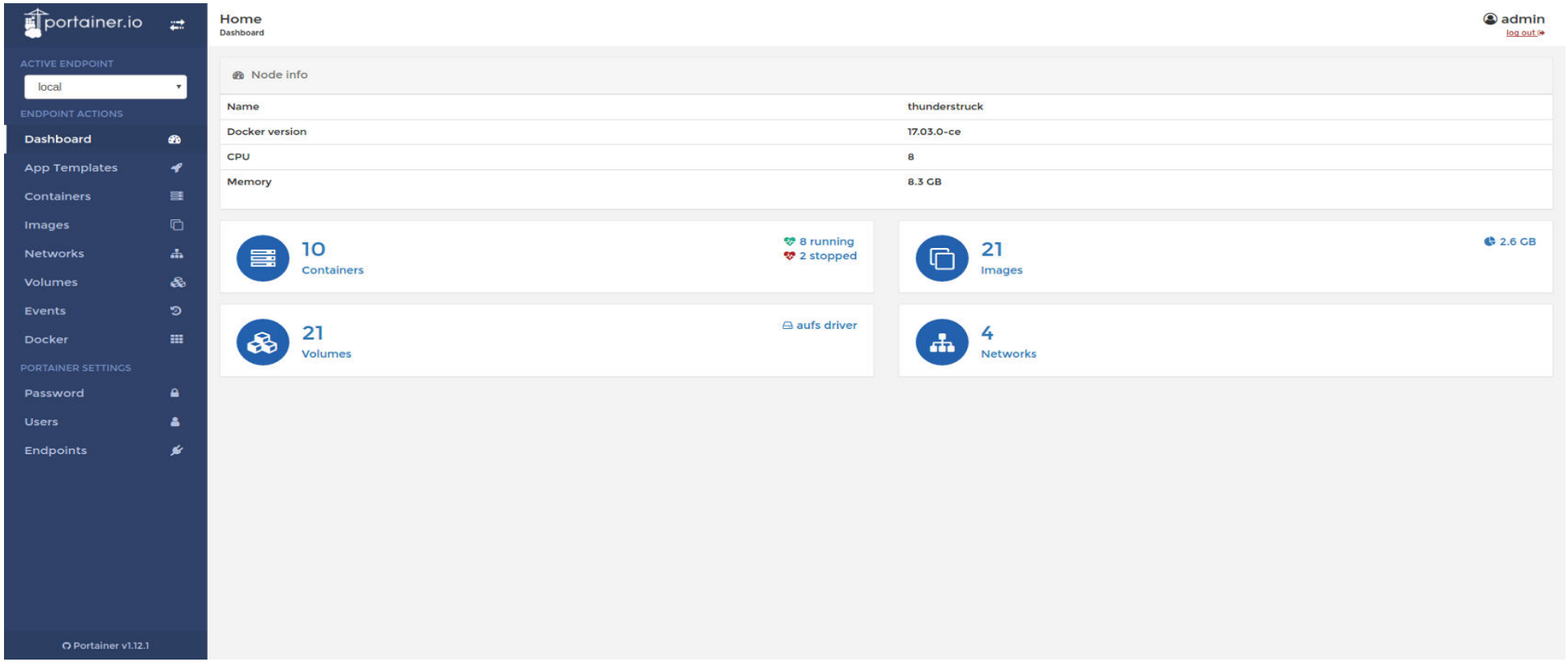


3 Options for User management and authentication:

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.)

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



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

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

PLANNING

Capacity Planning - Memory

- Memory for each zCX instance
  - Specified at provisioning time
  - Fixed private memory above the bar
  - Amount of memory needed will depend on the type of containers/software you are deploying
  - Minimum: 2GB; Recommended: at least 4GBs – especially if you are not memory constrained
    - ***Be aware that the whole memory is allocated as fixed memory by zCX at startup!***
    - *Allow for 1GB for zCX itself on top of your memory estimates*
    - *You need to ensure that this memory is available on your system and its use by zCX will not constrain your system*
  - Avoid Linux swapping if possible
    - If the containers deployed in zCX drive virtual memory usage above the amount of fixed memory specified, Linux will begin to page to its swap disks, significantly impacting performance and driving requirements for larger swap datasets.
    - Use real to swap memory ratio – **2:1** *(when zCX instances of 8GB or higher)*
  - Suggestion to tune the memory size
    - Determine the appropriate memory size is to keep Linux from swapping. Lower the memory size until Linux begins to swap, then increase the size to the next largest increment that does not impact performance.

Capacity Planning – Memory (Example)

- 3 applications running in your zCX
  - Each require an average of 5GB of virtual memory
- (3 applications * 5GB each) + (1GB for zCX) = 16GB total
- 16GB is greater than 8GB, therefore we should use 2:1 real to swap memory ratio
- 16GB * .66 = 10.66 = **11GB** (rounded up) of real memory
- 16GB – 11GB = **5GB** of swap memory (swap disk space)

Capacity Planning – Memory…

- How is the zCX appliance memory backed?
  - 4k pages (preferred storage pool (non-reconfigurable))
  - 1M pages (preferred storage poll (non-reconfigurable))*
  - 2G pages (fixed storage pool)*
- The bigger the page, the less z/OS overhead
  - Smaller translation table sizes, less Translation Lookaside Buffer misses

Page Size	% ITR Range	% ETR Range
2G	1- 13	>0 – 6
1M	1 - 10	>0 - 5
4k	>0 - 1	0 - 2

*requires OA59573

Capacity Planning – Memory…

- Which size page to choose:
  - **2G fixed pages**
    - best performance (1 page = over 524K 4k pages)
    - least flexible – storage pre-allocated at IPL and can't be used for other storage pools
  - **1M fixed pages**
    - middle performance (1 page = 256 4k pages)
    - good flexibility – LFAREA specification only a max value (storage can be reused by 4k pages if needed)
  - **4k pages**
    - worst performance
    - best availability
- How to select
  - Allocate the storage pools desired or increase their sizes
    - Combination of RSU and LFAREA parms in IEASYSxx
    - Choose storage pool selection criteria for each appliance when provisioning

See zCX operations workshop module for more details

WLM Policies

- Ensure that your WLM policies are updated to specify a Service Class for the zCX STC instance(s)
  - Classified under Subsystem=STC, can use Jobname or Userid qualifiers to identify zCX instances
  - Unique Service Classes (and optionally Report Classes) should be created
    - A single period should be created with an execution velocity goal– may need to experiment on exact value – will depend on the priority of this workload vs other workloads on the system, zIIP/CPU utilization, etc.
    - A WLM Importance Level that describes the overall priority of the zCX workload vs other workloads in the system
- zIIPs and spillover to general purpose processors
  - SRM/WLM provides options that allow you to control how work is assigned between zIIPs and standard CPs. zIIP-eligible work may also execute on general purpose processors in order to achieve workload goals.
  - System Level Option: Parmlib member IEAOPTxx contains the IIPHONORPRIORITY statement which controls the workflow to zIIPs.
    - Yes (Default) allows spillover to general purpose processors when needed. No indicates no spillover allowed (other than to resolve resource contention)
  - Service class level: You can specify YES/NO for Honor Priority when defining a service class
    - Overrides the IIPHONORPRIORITY setting in IEAOPTxx

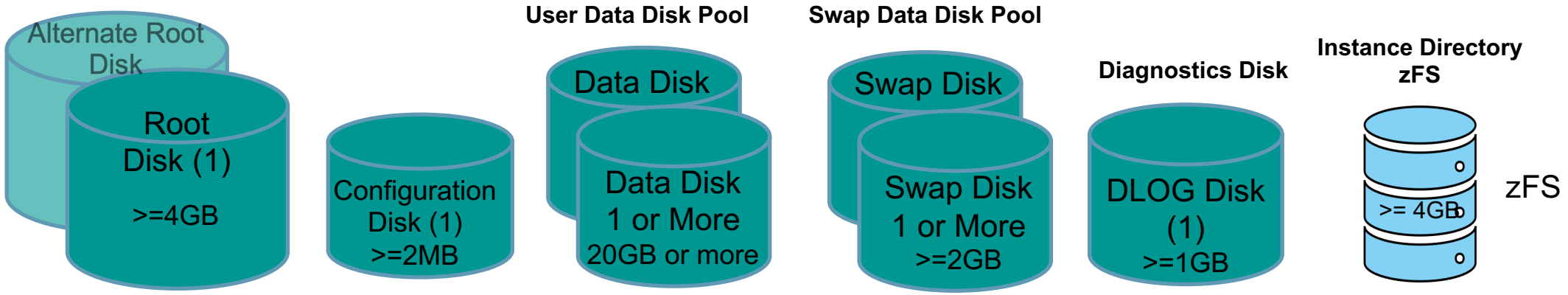
WLM Policies (Cont.)

- zIIPs and spillover to general purpose processors (cont)
  - Your WLM administrators and Capacity Planners will need to be consulted to determine what makes sense in your environment as they will have insights into available zIIP and standard CP capacity, etc.
- Also consider leveraging Tenant Resource Groups to cap CPU consumption
  - Associate the service classes associated with your zCX workloads with a particular group
- Your WLM and capacity planners may also need to consider whether a resource cap is needed for zIIPs and standard CP capacity one or more zCX STC instances can consume
  - This can be done using WLM Resource Groups or the new WLM Tenant Resource Groups (provide more granularity)

Recommendation: Consult with your WLM and Capacity Planning administrators to determine optimal settings for your environment

Planning for Datasets and zFS requirements

- Each zCX instance requires multiple VSAM datasets to be allocated for its exclusive use
  - zCX Appliance Linux Disks
    - These are VSAM LDS
    - Primary Extents only – datasets are fully allocated at provisioning time
    - Requires Data Class (DATACLASS) with Extended Format Required and Extended Addressability Enabled to allow >4GB datasets to be allocated
    - Define appropriate STORCLASS that will encompass the set of volumes eligible for zCX Appliance Instances
    - Define MGTCLASS as per installation practices
    - Define ACS routines to associate zCX datasets with appropriate DATACLASS, STORCLASS, MGTCLASS (per installation practices) or note down these classes for direct specification in z/OSMF zCX workflows
  - zCX Instance zFS – One per instance (supports primary and secondary extents)
    - Can use the same Dataclass/Storclass/Mgtclass specifications as the zCX Appliance Linux Disks or separate specifications



Planning for Datasets and zFS requirements (cont)

Dataset type	What is it used for	Size requirements	Can additional space be added?
Root disk	Linux root file system	>= 4GB	No – only through deprovisioning and reprovisioning
Configuration disk	Holds configuration data for zCX appliance	>=2MB	No – only through deprovisioning and reprovisioning
User Data disk	Holds all docker images, containers, logs and volumes	>=20GB recommended (workload dependent)*	Yes – additional disks can be added to data pool (zCX recycle required)
Swap Data disks	Used by Linux kernel for paging/swapping when virtual memory exceeds real memory	>=2GB minimum (workload dependent)	Yes – additional disks can be added to data pool (zCX recycle required)
Diagnostics and Logs (DLOGS) Data disk	Used to hold diagnostic data, logs and FFDC information	>=1GB minimum	No – only through deprovisioning and reprovisioning
Instance zFS	Used to hold zCX Appliance image, configuration file, etc.	>=4GB	Can be expanded by secondary extents
Alternate Root disk	When maintenance/service is applied via upgrade workflow, an alternate Linux root file system is created	>=4GB	No – only through deprovisioning and reprovisioning
Total space per zCX instance (minimum)		**37GB**	
Total space for all zCX instances		**Number of zCX instances * 37GB**	

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)
     - *Virtual SAMEHOST network not constrained to packet size limits imposed on physical network. In many cases, using a larger MTU between zCX and z/OS can reduce latency and increase throughtput significantly while reducing GPP and zIIP consumption*
   - 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.
   - EZAZCX interface is created when the EZASAMEMVS (samehost) interface is created.
     Note. When using DynamicXCF or have enabled IUTSAMEH for Enterprise Extender both EZASAMEMVS (samehost) and EZAZCX interfaces are dynamically created and started. If you're not using Dynamic XCF or Enterprise Extender, then you must manually define (dev/link/home) IUTSAMEH (which will also create EZAZCX)

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)

- 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.

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

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

>------+--------+---><
       '--ZCX---'
```

- 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*
    - 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 in the security product (e.g. RACF) must be UID(0) or have READ access to BPX.SUPERUSER FACILITY class profile

How do I get started today?

- Run on z/OS 2.4
- Have z/OSMF installed and running
- Obtain rights to use zCX
  - Purchase hardware feature code 0104, or
  - Use the zCX Trial (Try and Buy for up to 90 days)
    - Obtain APAR 0A58969
    - Display current registered products via D PROD,REG. If zCX not shown:

      Update IFAPRDxx member to add product enablement policy for zCX

      Activate the parmlib using SET PRD command.
    - Give the userid associated with the zCX server write access to the zCX instance directory
- Plan your resources (Memory, Storage, zIIPs , DVIPA, etc..)
- Provision your zCX server (instance)
- Start zCX server
- Install your docker applications

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
Getting Started with z/CX and Docker	http://www.redbooks.ibm.com/redbooks/pdfs/sg248457.pdf
IBM z/OS Container Extensions (zCX) use cases	http://www.redbooks.ibm.com/redpieces/pdfs/sg248471.pdf
Ambitus (Open Mainframe Project)	https://www.openmainframeproject.org/projects/ambitus

Other Resources

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