z/OS
2.4

IBM z/OS Container Extensions Guide

Note

Before using this information and the product it supports, read the information in “Notices” on page 113.

This edition applies to Version 2 Release 4 of z/OS (5650-ZOS) and to all subsequent releases and modifications until otherwise indicated in new editions.

Last updated: 2021-06-22

Contents

Figures

Tables

About the z/OS Container Extensions content solution

Purpose of this information

This is a collection of the information needed to understand and use IBM z/OS Container Extensions (zCX). Some of this information contained in this content solution also appears elsewhere in the z/OS library.

Who should read this information

This document is for anyone who is planning, setting up, or using z/OS Container Extensions.

Related information

To find the complete z/OS® library, go to IBM Documentation (www.ibm.com/docs/en/zos).

How to send your comments to IBM

We invite you to submit comments about the z/OS product documentation. Your valuable feedback helps to ensure accurate and high-quality information.

Important: If your comment regards a technical question or problem, see instead “If you have a technical problem” on page xi.

Submit your feedback by using the appropriate method for your type of comment or question:

Feedback on z/OS function
If your comment or question is about z/OS itself, submit a request through the IBM RFE Community (www.ibm.com/developerworks/rfe/).

Feedback on IBM® Documentation function
If your comment or question is about the IBM Documentation functionality, for example search capabilities or how to arrange the browser view, send a detailed email to IBM Documentation Support at ibmdocs@us.ibm.com.

Feedback on the z/OS product documentation and content
If your comment is about the information that is provided in the z/OS product documentation library, send a detailed email to mhvrcfs@us.ibm.com. We welcome any feedback that you have, including comments on the clarity, accuracy, or completeness of the information.

To help us better process your submission, include the following information:

- Your name, company/university/institution name, and email address
- The section title of the specific information to which your comment relates
- The comprehensive content collection title: Container Extensions
- The text of your comment.

When you send comments to IBM, you grant IBM a nonexclusive authority to use or distribute the comments in any way appropriate without incurring any obligation to you.

IBM or any other organizations use the personal information that you supply to contact you only about the issues that you submit.

If you have a technical problem

If you have a technical problem or question, do not use the feedback methods that are provided for sending documentation comments. Instead, take one or more of the following actions:

- Go to the IBM Support Portal (support.ibm.com).
- Contact your IBM service representative.
- Call IBM technical support.

Summary of changes

z/OS V2R4 updates (6/24/21)

- The abbreviation ILMT for IBM License Metric Tool is no longer used.

z/OS V2R4 updates (6/1/21)

- Updates for support for IBM Container Hosting Foundation for z/OS, which delivers Monthly License Charge (MLC) pricing to satisfy the requirement for the Z hardware feature code 0104. See "Requirements for zCX" on page 7 and Chapter 3, "zCX Trial," on page 5.

z/OS V2R4 updates (3/28/21)

- Update to the example LDAP configuration file in "Preparing an LDAP configuration file" on page 27
- New section on considerations for allocating large disk skies for zCX instances in "Storage (DASD/disk) considerations" on page 13

z/OS V2R4 updates (12/10/20)

- New sections added to Memory considerations
- New instructions for "Parallel workflow step automation" on page 53
- Updated variable description in "zCX instance swap data storage" on page 47
- Updated variable description in "zCX instance user data storage" on page 50

z/OS V2R4 updates (11/19/20)

- Updates for IPv6 support

z/OS V2R4 updates (9/24/20)

- New section Linux resource monitoring in Diagnostics for zCX
- Updates to Memory considerations
- New section "Modifying host ports to a container" on page 66 in Logging on to the Docker CLI and using zCX
- New section "Considerations for the number of containers and zCX instances" on page 18 replacing and enhancing Deploying multiple zCX instances

z/OS V2R4 updates (8/27/20)

- Updates to Memory considerations

z/OS V2R4 updates (5/28/20)

- Updates to Memory considerations
- Updates to "Preparing an LDAP configuration file" on page 27

z/OS V2R4 updates (4/1/20)

- zCX Trial is enabled for z/OS users with a z14 processor or higher

Table 1. Required z/OS 2.4 APAR/PTF sets for the zCX Trial	
APAR	**PTF**
OA58969	UJ02411

- Updates to “Preparing an LDAP configuration file” on page 27
- Updates to “Docker proxy support for zCX instances” on page 34
- Updates to messages: GLZB015I, GLZB019I, GLZB020I, & GLZB021I
- Updates to “Reason codes for zCX termination message” on page 106

z/OS V2R4 updates (1/30/20)

- Updates to Memory Considerations section

z/OS V2R4 updates (12/19/19)

- IBM Licence Metric Tool (ILMT) is enabled for zCX with the following APAR & PFT sets:

Table 2. Required z/OS 2.4 APAR/PTF sets for IBM License Metric Tool enablement with zCX	
APAR	**PTF**
oa58587	UJ01572
oa58601	UJ01571
oa58621	UJ01576
oa58598	UJ01574
oa58599	UJ01575
oa58600	UJ01577

- Documentation updates to Workload management considerations for zCX
- New messages: GLZB018I & GLZB019I

Chapter 1. What is z/OS Container Extensions?

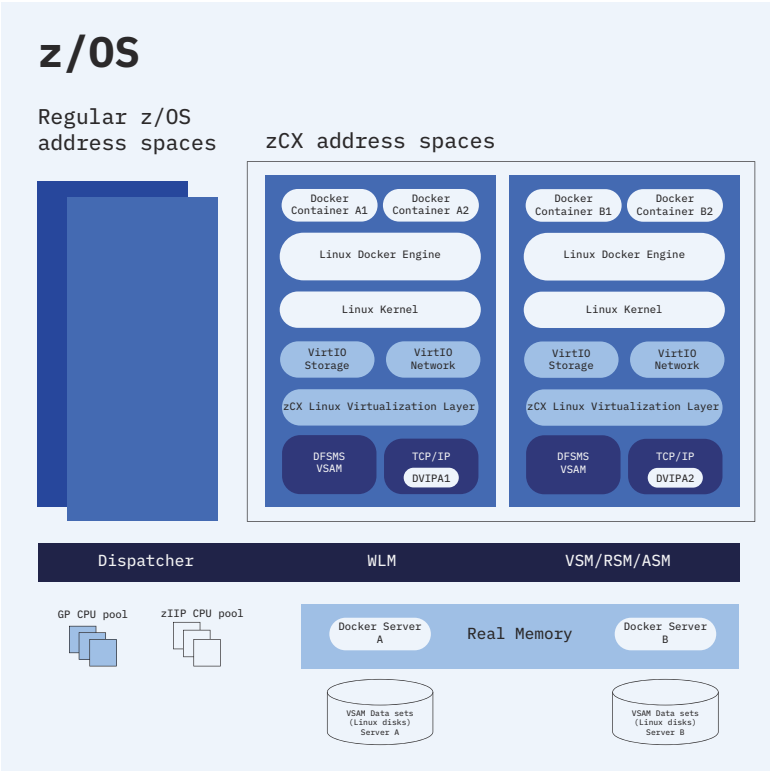
IBM z/OS Container Extensions (zCX) is a new z/OS 2.4 feature that enables clients to deploy Linux applications as Docker containers on z/OS as part of a z/OS workload. This maintains operational control of the Linux environment within z/OS, brings z/OS qualities of service to the application deployment, and does not require the provisioning of separate LPARs or system images.

IBM zCX expands and modernizes the z/OS software ecosystem by allowing applications and workloads built for Linux on Z and packaged into a Docker image to run on z/OS.

Docker is a framework that bundles an application, along with all of its dependencies and libraries,into one deployable package. It simplifies the configuration and installation of modern software. Dockerhub provides a registry and repository of Docker images that the community of Docker users has created over time. Any Docker image on Dockerhub with a tag of 's390x' or 'IBM Z' has been built for IBM Z, and can deploy either in a Linux on Z, or an IBM zCX environment.

In addition to the open source packages available on Dockerhub, software from IBM and independent software vendors may be available. You can also build your own images using the Docker infrastructure and command line provided by IBM zCX, just as you would on any other platform.

IBM zCX instances are provisioned and managed through the z/OS Management Facility (z/OSMF). z/OSMF workflows guide system administrators and automate the processes necessary to manage the lifecycle of a zCX instance. Once provisioned, Docker users and application developers can operate within a zCX instance as they would in any other Linux-based Docker environment.



For an overview of zCX and a variety of resources, see z/OS Container Extensions.

How is zCX similar to and different from Linux?

IBM zCX is a virtual Docker software appliance that includes all the necessary software components to allow a user to deploy and manage Linux on Z Docker images (e.g. s390x architecture) inside z/OS. The software appliance includes a commercial Linux kernel supported and maintained by IBM and provides

the Docker command line interface (CLI) to implement a fully functional Docker environment. Users log in to the zCX instance through Secure Shell (SSH), and run in a Bash shell as they would on any Linux platform.

This Linux environment is tailored to support the Docker CLI, and is not intended to be a general purpose operating environment. Users are prevented from performing most root operations, so that the kernel remains properly configured for running Docker. If users need a fully functional Linux environment, they should create a Docker image with whatever Linux distribution they need,and run it through the Docker CLI.

One exception to the root authority restriction is the ability of the primary zCX instance administrator (admin) to create, delete, and modify additional users and their passwords. In this way, these primary instance administrators can create the population of Docker administrators and users that fit their needs.

What can you do with zCX?

IBM zCX allows you to bring "Dockerized" Linux applications as close as possible to your enterprise z/OS resources without first having to port lots of open source infrastructure to z/OS. For instance,applications that communicate over TCP/IP with z/OS data or services can take advantage of z/OS's high speed SAMEHOST networking that is only available to processes running on the same LPAR.

Since IBM zCX instances run in a regular z/OS address space, they inherit many of the same qualities of service as other z/OS processes. All of the capabilities z/OS provides through VSAM and the TCP/IP stack to enable encryption, disaster recovery, and dynamic workload relocation, can all be applied to the zCX instance without disruption of the Linux applications running inside. In this way, IBM zCX enhances the Linux deployment environment in ways that other platforms can't.

From a Linux perspective, IBM zCX supports a large population of Docker images that allow z/OS to integrate more tightly with common pipelines and workflows that the open source community has created over the last several years. This allows z/OS to participate in common devops and automation environments through projects like Jenkins and Github, or for data science and analytics purposes through Jupyter and Jupyterhub. There are also assortments of languages, build frameworks,and web servers that can be built into Docker images and deployed on IBM zCX to enable much fuller z/OS participation in the open source arena with the agility that modern enterprises require.

Chapter 2. Terms you should know

By bringing new Docker container capabilities to z/OS, zCX requires familiarity with concepts and terminology surrounding Docker.

Docker image

A Docker image is a file composed of multiple layers that is used to execute code. Each layer has an associated JSON structure describing its basic information.

Docker registry

A Docker registry is a storage and distribution system for Docker images. Docker hub is the default public registry for finding and sharing Docker images. Private or local Docker registries can also be built inside your environment.

Docker container

A Docker container is a running instance of a Docker image.

Docker client

The Docker client is a command line interface (CLI) that allows users to interact with Docker and Docker containers. You can issue application commands such as build, run, and stop through the Docker client.

zCX Docker CLI SSH Container

The zCX-provided SSH environment that can be used to execute Docker commands.

zCX instance

A zCX instance is a z/OS Container Extensions address space, which is running one particular Linux operating system image. There can be multiple zCX instances in one z/OS image running in an LPAR. There can be multiple Docker containers running in a zCX instance.

Docker daemon

The Docker daemon is part of Docker and lives on the host operating system. It listen for requests and manages Docker objects.

Chapter 3. zCX Trial

You can try z/OS Container Extensions using the 90-day zCX Trial before deciding whether to buy the feature code. An IBM z14 or higher processor is required to enable the trial. You will have 90 days from the day the trial begins to use and evaluate the function. The trial extends to all zCX instances running on the sysplex. After 90 elapsed days, you can purchase the Container Hosting Foundation (feature code 0104) to continue using zCX.

Setting up the zCX Trial

Activating the zCX Trial is a simple process that does not require assistance from IBM. Use the following steps to begin your trial.

Edit the dynamic element parmlib member

1. Create a copy of `IFAPRD00` or your currently active `IFAPRDxx` member to enable the trial.
2. Update the new member with the following definitions:

```
PRODUCT OWNER('IBM CORP')
NAME('z/OS')
ID(5650-ZOS)
VERSION(*)
RELEASE(*)
MOD(*)
FEATURENAME('zCX TRIAL90')
STATE(ENABLED)
```

3. Issue the `SET PROD` command to specify the new member as active on the system.
4. Update `IEASYSxx` to point to the new member.

Grant file system access

To start the trial, each userID associated with zCX started task must have write access in addition to search access to the zCX instance registry directory in the z/OS UNIX System Services file system.

1. Issue the `ls -ld` *`REGISTRY_DIR`* command in the z/OS UNIX System Services OMVS shell to find the group name associated with the zCX instance registry directory.
2. Use the `CONNECT RACF TSO` command to associate the userID with the group name.

Trial duration

Once set up, the trial can run for up to 90 days. Every time a zCX server is started it will display a z/OS console message indicating the number of days left in the trial. No disablement is required when the trial ends. zCX will simply not start and a console message will be issued.

If you wish to end the trial before the 90 days have elapsed, use the following steps to prevent zCX from starting:

1. Edit the IFAPRDxx member that enabled the trial to remove the entire section associated with `FEATURENAME('zCX Trial90')`.
   - Alternatively, `STATE(ENABLED)` can be adjusted to `STATE(DISABLED)`.
2. Adjust the file system to remove write access granted for the trial.

While these steps are not required, they are necessary if you wish to prevent zCX from being restarted during the 90-day duration. They may be optionally conducted for system hygiene after the 90-day full duration as well.

The trial cannot be paused. The original start time will be used to calculate the 90-day interval.

If you mistakenly remove or disable the IFAPRDxx statement during the trial, the 90-day duration continues. Replace or enable the IFAPRDxx statement as soon as possible to continue the trial.

When the trial is over

To continue using zCX beyond the trial, you must purchase and install feature code 0104 (IBM Container Hosting Foundation) through eConfig Fulfillment Center. IBM Container Hosting Foundation for z/OS delivers Monthly License Charge (MLC) pricing to satisfy the requirement for the Z hardware feature code 0104. For more information, see Program Directory forIBM Container Hosting Foundation for z/OS (publibfp.dhe.ibm.com/epubs/pdf/c3157030.pdf).

Encountering zCX Trial errors

Error messages are issued in response to failure during setup. They include return and reason codes to identify and correct the errors. For UNIX System Service errors, the return and reason codes are errno and errnojrs. If data in the file system cannot be accessed, the trial probably did not start. It is, however, possible to have errors even when the trial has started. These errors should be dealt with promptly, as the trial start time and date are not adjusted.

Chapter 4. Planning for zCX

This chapter describes zCX requirements and considerations for your system. These sections assume a proficiency in z/OS and provide recommendations to prepare you to deploy a zCX instance.

Requirements for zCX

Hardware requirements

z/OS Container Extensions requires an IBM z14 or higher based server with Container Hosting Foundation (hardware feature code 0104). This feature code includes service and support for the underlying Linux kernel and zCX appliance, as well as unlimited zCX usage in the central processor complex (CPC) across all LPARs. This hardware feature can be ordered from the eConfig Fulfillment System. Monthly License Charge (MLC) pricing is available with IBM Container Hosting Foundation for z/OS. See "Software requirements" on page 7. If you have a z14 or higher processor but do not have feature code 0104, you can enable the zCX trial and experiment with zCX for 90 days. Details can be found in zCX Trial.

Virtualization layer

The virtualization layer required for zCX is shipped in the z/OS BCP base HBB77C0 as new parts and has the new Comp ID 5752-SCCON. The virtualization layer has the 3-character prefix GLZ.

Software requirements

zCX is shipped as a new element of z/OS in the new FMID: HZDC7C0. This includes the Linux kernel and Docker Engine (5752-SCCDE) as well as the z/OSMF Workflow (5752-SCCWF). The software has the 3-character prefix AZD. All systems in the sysplex on which you will deploy zCX must be at the z/OS V2R4 level. IBM Container Hosting Foundation for z/OS delivers Monthly License Charge (MLC) pricing to satisfy the requirement for the Z hardware feature code 0104.

z/OS Management Facility (z/OSMF) for zCX

The z/OS Management Facility (z/OSMF) interface is used to provision, deprovision, and otherwise maintain zCX instances. Specifically, zCX uses z/OSMF workflows to guide provisioning and maintenance of a zCX instance. Therefore, z/OSMF must be active in a system on the sysplex where you are deploying zCX. All systems in this sysplex must be defined in z/OSMF. To define a system in z/OSMF, go to **z/OSMF Settings**, then **Systems Option**. Any user who will be provisioning zCX instances needs access to both z/OSMF and z/OSMF workflows.

If you do not yet have z/OSMF configured on your system, refer to the z/OSMF Configuration Guide in the IBM Documentation.

The first step of the zCX provisioning workflow collects all the necessary variable values. The values can be provided manually or by specifying a z/OSMF workflow variables input properties file (more information on this below). A workflow variables input properties file can also be used for some of the variables, with manual changes to the values from the file directly in z/OSMF. After providing the necessary values, the workflow steps can be automated. The final step of the zCX provisioning workflow provides the z/OS console START command that can be used to start the zCX instance.

z/OSMF workflow variables input properties file

The workflow variables input properties file stores all the input values. The workflow extracts the values from the file, therefore saving the time otherwise required to manually input each one. A sample workflow variables input properties file can be found in the following location:

```
/usr/lpp/zcx_zos/properties/workflow_variables.properties
```

This sample file provides the same default values as the z/OSMF interface. It can be copied and customized according to your installation requirements.

CPU planning considerations

You must plan for the number of virtual CPUs to allocate to each zCX instance. One zCX instance can host multiple Docker container instances. The number of virtual CPUs you want to allocate for a zCX instance and its respective containers is specified during provisioning. Each virtual CPU is a z/OS dispatchable task within the zCX address space. This allows z/OS to dispatch the zCX virtual processors on the available z/OS processors, as with any other z/OS workload based on Workload Management policies and priorities. zCX virtual processors can be dispatched on zIIPs or general purpose CPUs. For more details, refer to **Workload Management considerations** in this chapter. For the best performance, IBM recommends that the number of virtual CPUs across all zCX instances expecting heavy and simultaneous usage should be no greater than the number of zIIPs or general purpose processors available to the z/OS system.

When Simultaneous Multi-Threading (SMT) is enabled, zCX can exploit the additional zIIP CPU threads.

The number of virtual CPUs allocated to a zCX instance can be modified using the reconfiguration workflow in z/OSMF. However, you must recycle the zCX instance to put the modification into effect.

Memory considerations

The amount of real and swap memory needed for a zCX instance depends on the memory requirements of the containers and applications that will run inside it. The combined virtual memory requirements of the applications that will run concurrently should not exceed double the amount of real memory provisioned to the zCX instance (maximum ratio of 2:1 virtual to real). You can use Docker and Linux monitors to determine container and Linux memory use.

Avoid over-defining the memory size, as Linux uses excess memory for file and buffer caches of potentially low-value content. While these buffers can help workloads avoid I/O on a stand-alone system, you should account for them in your sizing to maximize memory being used for the guest.

One option to 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.

Calculating real and swap memory allocation for your zCX instance

The total memory available to the zCX instance (both real and swap) must be at least 1 GB greater than the total virtual memory needed for all concurrently running applications.

The appropriate proportion of real to swap memory depends on the amount of real memory provisioned to the zCX instance. For zCX instances with less than 8 GB of real memory, use a 1:1 ratio of real to swap memory. For zCX instances greater than or equal to 8 GB, use a 2:1 ratio of real to swap memory.

For example, if you wish to concurrently run five applications that each require 4 GB of virtual memory, you would want at least 14 GB of real memory and 7 GB of swap memory:

```
5 applications * 4 GB each = 20 GB
+ 1 GB for zCX = 21 GB total

The general rule of a 1:1 ratio of real to swap memory yields 11 GB each (rounded up). This is
greater than 8 GB, so the 2:1 real to swap memory ratio (66% real memory) should be used.

21 GB * 0.66 = 14 GB (rounded up) of real memory
21 GB – 14 GB = 7 GB of swap memory.
```

Failure to allocate sufficient real or total memory can cause a zCX instance to run out of memory and reboot. (This would issue message GLZB017I to the zCX instance job log). If this occurs, re-examine the

virtual storage requirements of your applications and the subsequent real and swap memory calculations for your zCX instance.

Sizing and other considerations for the z/OS memory storage pool and your zCX instance

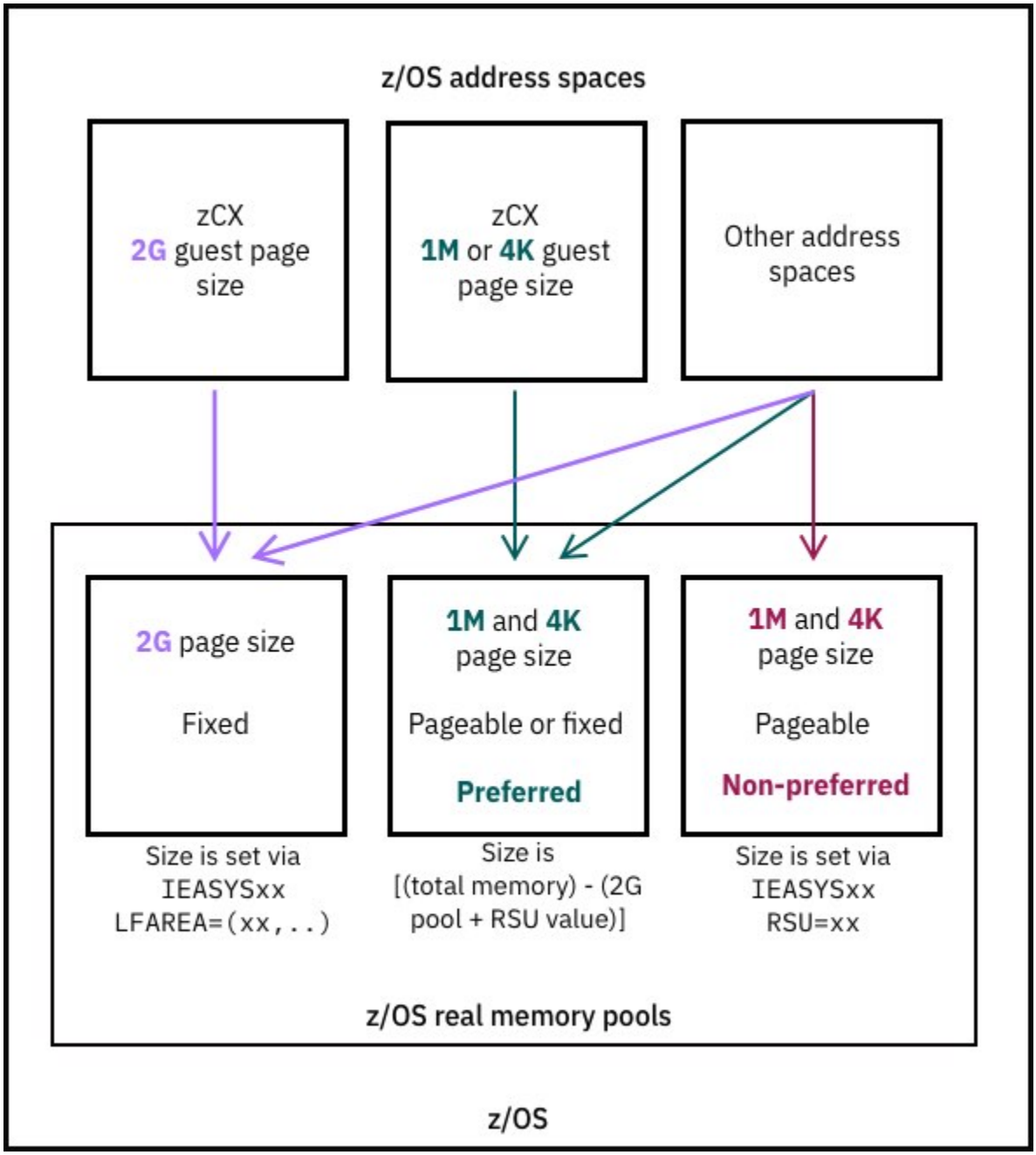


Figure 1. z/OS Storage Pools

Figure 1 on page 9 illustrates the different z/OS storage pools and use based on page size and preferred (non-reconfigurable) or non-preferred (reconfigurable) storage. Storage is separated into the following pools:

- 2G fixed
- 1M and 4K preferred
- 1M and 4K non-preferred

zCX APAR OA59573 enables use of 2G or 1M fixed pages to back guest memory. Using 2G pages provides the best performance. Both 2G and 1M pages save page table storage compared to 4K pages. A 2G page size for guest memory is not supported when z/OS is a z/VM guest. Neither 2G nor 1M page sizes are supported on zPDT.

The preferred 2G fixed pool size is defined at IPL by the IEASYSxx 2G LFAREA value and cannot be changed after IPL. The remaining memory is split between the preferred and non-preferred 1M and 4K areas.

The non-preferred storage pool size is determined by the IEASYSxx RSU (Reconfigurable Storage Units) value and can be dynamically changed after IPL. The remaining memory determines the size of the preferred 1M and 4K storage pool size.

The DISPLAY M=STOR command displays the non-preferred (reconfigurable) and preferred (non-reconfigurable) storage values for the system. The F AXR,IAXDMEM and DISPLAY M=STOR commands display the following metrics:

- LFAREA-defined total sizes (LFAREA values may be restricted to a subset of the overall available size)
- In-use allocation
- Maximum in-use allocation for 1M fixed and 2G pages
- RSU value

The system uses the fixed storage threshold control IEAOPTxx MCCFXTPR to limit how much of the 1M and 4K pool areas are fixed at any given time. When system reaches the threshold, it will swap out address spaces that are using fixed storage in order to protect itself.

More information on z/OS storage pools and page sizes is available in the z/OS MVS Initialization and Tuning Guide and z/OS MVS Initialization and Tuning Reference.

Choosing page sizes for zCX

The chosen page size determines the z/OS storage pool from which the guest memory comes. zCX address spaces are non-swappable and storage containing fixed frames is not reconfigurable. Therefore, a 1M or 4K page size uses storage from the 1M and 4K preferred storage pool and a 2G page size uses storage from the 2G fixed storage pool. The storage cannot come from non-preferred (reconfigurable) storage.

For example, suppose an LPAR has 260 GB of real storage defined as follows:

- 60 GB of storage defined as 2G pages
- 30 GB of LFAREA- defined total storage as 1M fixed pages
- 40 GB of RSU storage via the RSU parameter

The memory available for zCX would depend on the selected page frame size.

- For a 2G page size, there would be 60 GB minus what is required for other workloads.
- For a 1M page size, there would be 30 GB minus what is required for other workloads.
- For a 4k page size, there would be 160 GB minus what is required for other workloads

The amount of the 160 GB that can be fixed before a pageable storage shortage occurs is controlled by the IEAOPTxx MCCFXTPR parameter.

You can configure a zCX instance to use a single page size or a range of page sizes, which the system will attempt to use in descending size order. Only one page size is used for the guest storage. Select a single page size when you want zCX to use only that size. Selecting multiple page sizes provide higher availability and is useful for systems that have different memory requirements. zCX will terminate the instance if it cannot use any of the selected page sizes. You can automate message GLZ0024I for notifications when zCX cannot use a selected page size.

If you are using a 1M or 4K page size, consider altering your IEAOPTxx OPT IRA405I(2) and MCCFXTPR parameters. These determine the threshold of the fixed storage system and the percentages of preferred and non-preferred storage that may be fixed before a pageable storage shortage, respectively.

Alterations should depend on possible changes to the ratio of fixed and available real storage. If these parameters are not appropriately adjusted, it can lead to severe system impacts. The MVS Initialization and Tuning Reference includes more information on IEAOPTxx and the Workload Management considerations about limiting real storage usage.

Impact of page sizes

You should plan for the impact of zCX on the z/OS storage pool resources based on the selected page size(s). This includes sysplex failure environments, where a zCX instance may restart on a different system. You may need to add more real memory or related auxiliary storage to the system. You can add more 1M or 4K preferred real storage either by adding more physical memory or by reducing the amount of non-preferred storage. Adding more 2G fixed memory requires an IPL.

zCX protects the system from storage impacts during initialization when using 1M or 4K pages. It queries the system using the SRM SYSEVENT STGTEST API to determine the impact of the guest fixed real memory on system performance and availability. If the impact is critical, zCX terminates. If the impact is minor, initialization continues. This check is not required when using 2G fixed pages, as they are fixed by definition. You can find more information on SYSEVENT STGTEST in z/OS MVS Programming: Authorized Assembler Services Reference SET-WTOSYSEVENT — System event Obtain system measurement information (STGTEST).

Regardless of page size, SVC dumps of zCX instances with large memory footprints can put pressure on real and auxiliary storage resources. APAR OA59573 defaults that the guest memory will not be captured during a standalone dump.

Summary of frame size options

Consider all systems on which zCX may run.

- 2G
  - Choose this option for the best performance and CPU use when you have 2G page space available. The 2G page space must be available on all systems on which zCX may run.
  - Choose this option when storage from 1M and 4K preferred pool would impact the system because the pool would not have enough available fixed space.
- 2G, 4K:
  - Choose this option for the best performance and highest availability. The 4k pages provide higher availability if 2G pages are unexpectedly unavailable or if zCX temporarily restarts on another system that does not have 2G storage available.
  - Choosing **First Fit 2G, 1M, 4K** can also be considered as 1M and 4K pages come out of the same preferred storage pool. However, keep in mind that using 1M fixed pages without a large enough LFAREA maximum for 1M may impact other 1M fixed users who start after the appliance. As such, planning for 2G pages with 4K pages as a backup might be the better choice.
- 1M, 4K:
  - Choose this option when running z/OS as a z/VM guest. Since 2G pages are not supported as a z/VM guest, 1M pages provide the best performance on z/VM.
  - This option allows storage used by zCX available to other workloads that do not support 2G pages when the zCX is not running and thus not using storage.
  - This option provides high availability when 1M fixed pages are unavailable due to fragmentation of the preferred 1M and 4K storage pool.
- 4K:
  - Choose this option for highest compatibility. It has the lowest performance.
  - This option is only recommended when waiting for the next IPL to increase the LFAREA or when you do not want to dedicate 2G or 1M pages since zCX is not in production or performance sensitive.

Implementing storage limits for zCX instances

You can also limit the amount of real memory that one or more zCX instances can consume by specifying a memory limit (`MEMLIMIT`) in the WLM resource group associated with zCX instances, as described in the **Workload management considerations** section in this chapter.

The virtual memory used above the bar, however, is not limited by the memory limit (`MEMLIMIT`) control. Regardless of the page size used, the real storage used by zCX can be limited via WLM resource groups.

Memory demands that reduce available memory for Docker containers

- The memory reserved by zCX for handling Linux kernel crashes
- The memory reserved by the Linux kernel for its own use
- The memory used by processes running in the zCX instance in support of Docker
- The memory used by the Linux kernel to manage disk devices

Linux kernel crash memory reservation

Table 3. Memory reserved for Linux crash handling according to zCX instance memory size

zCX instance memory size	Memory reserved for Linux crash handling
2G to 255G	512M
256G to 1023G	1G
1024G	2G

Estimating Linux reserved memory

Use the following formula to estimate how much memory the Linux kernel reserves for its own use.

```
reserved-in-M = ((18000 * instance-mem-in-G) + 40000) / 1024
```

Estimating baseline memory usage

zCX instances have processes (Docker daemon, zCX Docker CLI SSH container, and more) that use memory. These processes are always running and their memory cannot be freed for use by other Docker containers. There is no formula for determining how much memory is used for these processes. You can use the Linux `free` command (`free -h`) to display the currently available memory for a newly provisioned zCX instance. This provides a starting estimate for the memory available for Docker containers.

Using a large number of data/swap disks

As additional data and/or swap disks are added to a zCX instance, the Linux kernel uses more memory to manage these devices. If your zCX instance has many (over 100) data and/or swap devices, you may need to increase your memory allocation to account for the additional memory usage.

Example of evaluating memory of a zCX instance

For zCX instance provisioned with 4G of memory:

- 512M of memory is reserved for handling Linux kernel crashes
- Approximately 110M of memory will be reserved by the Linux kernel for its own use
  - `((18000 * 4) + 40000) / 1024 = 110M`

In total, 622M of memory are reserved, leaving 3474M (3.4G) for Linux non-kernel processes.

The Linux `free` command (`free -h`) can be used in the zCX Docker CLI SSH container to view how much memory is available for Docker containers:

Table 4. Example Linux free command output

	Total	**Used**	**Free**	**Shared**	**Buff/cache**	**Available**
Mem:	3.4G	699M	2.3G	5.4M	467M	2.6G
Swap:	997M	0B	997M			

In the output, the **Available** column shows the amount of memory available for Docker containers (2.6G). Therefore, approximately 1.4G of the original 4G specification is used for other purposes and is not available for Docker containers.

Storage (DASD/disk) considerations

Each zCX instance requires multiple VSAM linear data sets (LDS) to be allocated for its exclusive use. You must define a High Level Qualifier (HLQ) that the provisioning workflow uses to allocate these VSAM LDS data sets. The HLQ value, which is specified in the provisioning workflow, can be multiple qualifiers, but the length must be less than or equal to to 28 characters (including periods). The VSAM LDS are fully allocated during provisioning as primary extents only. Secondary extents are not supported. Consider having unique HLQs for each set of zCX appliance instances that have unique users or purposes, such as PROD, DEV, and TEST.

zCX VSAM data set requirements

- To allow allocation of data sets greater than 4 GB, use a *DATACLAS* with Extended Format and Extended Addressability.
- Use a *MGMTCLAS* that specifies the Partial Release attribute as NO to prevent unused space from being released when a data set is closed.

For SMS-managed data sets, attributes such as *STORCLAS*, *DATACLAS*, and *MGMTCLAS* must be assigned either by specifying the values in the provisioning workflow or by SMS Automatic Class Selection (ACS) routines. A valid storage group must be assigned by your ACS routines, and the volumes in that storage group should have the same performance attributes as the STORCLAS assigned to the data set.

For non SMS-managed data sets, a VOLSER must be specified in the provisioning workflow.

zCX VSAM data set permissions

- Each zCX provisioning user needs a security profile which permits the allocation of VSAM data sets.
- Each zCX provisioning user requires ALTER access to the zCX instance's VSAM data sets.
- Each user ID assigned to a zCX started task will require CONTROL access to the zCX instance's VSAM data sets.

zFS authority requirements

A Z file system (zFS) is allocated and mounted for each zCX instance. You can use the same HLQ as used for VSAM data set allocation, or specify a different HLQ to allocate the zFS. Primary extents and secondary extents are both supported on zFS allocation. The same *DATACLAS*, *STORCLAS*, and *MGMTCLAS* requirements apply to zFS allocation and VSAM data set allocation.

z/OS provisioning users needs permission to allocate, mount, unmount, and delete a zFS on the zCX instance directory. Permission is granted by setting up non-privileged user mount `UNIXPRIV` class resource, and providing users access to the `SUPERUSER.FILESYS.USERMOUNT` resource.

Table 5. Data sets for zCX			
Data set type	**Purpose**	**Size requirement**	**Can additional space be added?**
Root disk	Linux root file system	Greater than or equal to 4 GB	No, only through deprovisioning and reprovisioning
Configuration disk	Holds configuration data for zCX appliance instances	Greater than or equal to 2 MB	No, only through deprovisioning and reprovisioning
User Data disk	Holds all docker images, containers, logs, and volumes	Greater than or equal to 20 GB recommended (workload dependent)	Yes, additional disks can be added to the data pool (zCX recycle required)
Swap Data disks	Optionally used by Linux kernel for paging/ swapping when virtual memory exceeds real memory	If used, greater than or equal to 2 GB (workload dependent)	Yes, additional disks can be added to the data pool (zCX recycle required)
Diagnostics and Logs (DLOGS) Data disk	Holds diagnostic data, logs, and FFDC information	Greater than or equal to 1 GB	No, only through deprovisioning and reprovisioning
Instance Directory zFS	Holds the zCX appliance image, configuration file, and FFDC information.	Greater than or equal to 4 GB	Can be expanded by secondary extents

Considerations for allocating large disk sizes for zCX instances

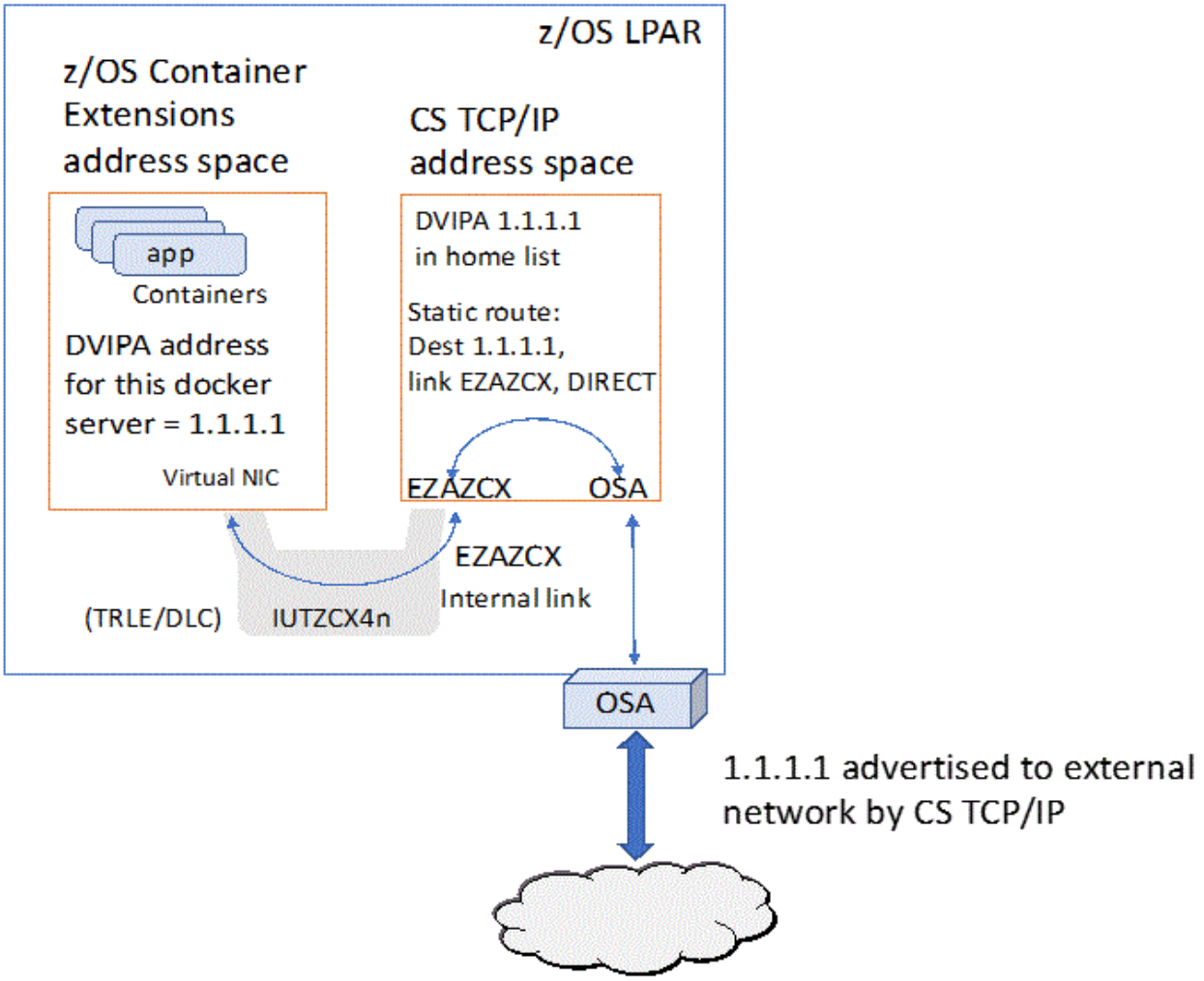
Beginning with OA60920 service level, zCX z/OSMF `provision` and `add_data_disks` workflows provide the capability to allocate large disk sizes up to 1024 GB for zCX instances. Large disks require additional time to format the disks for zCX use. Formatting of the disks (VSAM linear data sets) is performed during the first start of the zCX instance. As a result, there will be longer startup times for the first boot of zCX instances when large disks are specified.

IBM z/OS Container Extensions network overview

IBM z/OS Container Extensions (zCX) provides an execution environment allowing z/OS to host applications based on Linux® that are managed with docker containers. Each instance of zCX (unique zCX job) is provisioned within a unique z/OS address space. The zCX address space represents a virtual server which hosts applications managed in docker containers. The virtual server does not require operational tasks from the z/OS environment or the z/OS administrator. From an operational perspective, the virtual server is transparent to the z/OS environment. The zCX environment is configured using z/OSMF. For information about configuring the zCX environment, see *IBM z/OS Management Facility Online Help* for Configuration Workflow.

z/OS Communications Server provides network communications and network related services for the zCX workloads. The Linux virtual server is represented by a unique type of application instance DVIPA called a zCX DVIPA. The VIPARange statement is used with the ZCX keyword to create zCX DVIPAs. Defining zCX DVIPAs is the primary IP configuration task and in many instances it will be the only required configuration task.

The following figure provides an overview of the z/OS Communications Server support provided for the zCX environment.



Users define zCX DVIPAs to TCP/IP using VIPARange with the ZCX parameter using Network Configuration Assistant or directly in their TCP/IP profile.

The remaining steps are automatic:

1. EZAZCX interface is automatically created[1] by TCP/IP and connected to a dynamically created internal network represented by a dynamically created TRLE called IUTZCX4n. A unique instance of the IUTZCX4n DLC is created for each TCP/IP stack connecting to zCX servers. The n represents the instance of each IUTZCX4n DLC created.
2. When the zCX job is started by the user:
   a. this instance of zCX binds to the DVIPA and connects to TCP/IP over EZAZCX
   b. DVIPA (zCX) is activated / added to the home list
   c. TCP/IP creates a static route to the zCX DVIPAs for its own internal use. This route is local to and controlled by the stack that owns the zCX and is not advertised using dynamic routing.
3. When using dynamic routing the zCX DVIPAs are treated by OMPROUTE as application-instance DVIPAs and are advertised accordingly.[2]

[1] The EZAZCX interface is automatically created when the IUTSAMEH interface is created (either by DYNAMICXCF or static DEVICE/LINK/HOME for IUTSAMEH) and at least one VIPARANGE ZCX is configured. The EZAZCX interface transitions to ready when the first zCX DVIPA is activated. The EZAZCX interface is created for zCX instances using an IPv4 DVIPA. If the zCX instance is using an IPv6 DVIPA, the interface created is EZ6ZCX and the TRLE created is IUTZCX6n.

[2] When using dynamic routing, you must define the zCX DVIPAs to OMPRoute just like any other application-instance DVIPA. OMPROUTE will then advertise the zCX DVIPA when it activates on the host. When the

4. CS TCP/IP forwards packets for 1.1.1.1 over the zCX IP route and the EZAZCX Interface.

Note: IP filters can be configured and applied during IP forwarding. IPSec tunnels can be applied to the external IP routes. If you have IP filters defined, updates to your IP filter rules are required. You must ensure that you permit ROUTED and LOCAL (EITHER) traffic for the zCX DVIPAs.

z/OS Container Extensions IPv6 network overview

z/OS supports zCX instances with both IPv4 and IPv6 connectivity. A zCX instance can support just IPv4 or both IPv4 and IPv6 connectivity.

Defining zCX DVIPAs using VIPARange remains as the primary IP configuration task that controls and enables each zCX instance for each specific IP version. When the zCX job is started, the appropriate zCX interfaces will be started.

Configuring and enabling zCX IPv6 requires that your zCX instance is already enabled for IPv4 (see the above IPv4 steps) and then the required IPv6 steps are summarized below for the following two use cases:

- **z/OS TCP/IP users who already have enabled z/OS for IPv6 (key steps):**
  1. Define IPv6 zCX DVIPAs using the VIPARange statement.

     Note:

     a. A separate IPv6 VIPARange statement should be created for each zCX instance that requires IPv6 connectivity.

     b. For supporting high availability for zCX, your IPv6 VIPARange statements should be configured in all z/OS TCP/IP instances within the sysplex eligible to host this zCX IPv6 instance.
  2. Specify the IPv6 DVIPA in the z/OSMF zCX workflows (provisioning or reconfiguration workflows). Optionally specify any other IPv6 addresses or hostnames for any zCX configuration options, such as DNS, Registry, Proxy, or LDAP addresses using the z/OSMF zCX workflows.
  3. Common IPv6 configuration steps:

     Most existing z/OS IPv6 users will have already completed the following common IPv6 steps, but here are some key steps to consider:

     a. External IPv6 Interfaces:

        To enable IPv6 communications with hosts external to this z/OS TCP/IP instance you must enable the associated z/OS IPv6 interfaces, such as OSA or HiperSockets.

     b. IPv6 Dynamic XCF must be enabled.[3]

     c. If using dynamic routing, define your IPv6 DVIPAs to OMPRoute.

        Note: IP filters can be configured and applied during IP forwarding. IPSec tunnels can be applied to the external IPv6 routes. If you have IP filters defined, updates to your IP filter rules are required. You must ensure that you permit ROUTED and LOCAL (EITHER) traffic for the zCX DVIPAs.
- **z/OS TCP/IP users who have not enabled z/OS for IPv6:**
  1. This type of user must first complete the z/OS IPv6 migration or enablement. This type of user must start with the z/OS Communications Server: IPv6 Network and Appl Design Guide.

DVIPA is moved, external hosts will automatically (with dynamic routing updates) find the new location of the DVIPA and the related zCX applications. If OMPROUTE is not being used, you must ensure the zCX DVIPAs can be reached by other hosts by defining static routes on other hosts that need to reach the zCX instance.

[3] The EZAZCX interface is automatically created when the IUTSAMEH interface is created (either by IPCONFIG(6) DYNAMICXCF or static DEVICE/LINK/HOME for IUTSAMEH) and at least one VIPARANGE ZCX is configured. The EZAZCX interface transitions to ready when the first zCX DVIPA is activated. The EZAZCX interface is created for zCX instances using an IPv4 DVIPA. If the zCX instance is using an IPv6 DVIPA, the interface created is EZ6ZCX and the TRLE created is IUTZCX6n.

2. Once your z/OS TCP/IP stack is enabled for IPv6, review the first list above for the specific zCX IPv6 steps.

Configuring application-instance DVIPAs for IBM z/OS Container Extensions (zCX)

IBM z/OS Container Extensions (zCX) provides an execution environment allowing z/OS to host applications based on Linux docker containers. Each instance of zCX (unique zCX job) is provisioned within a unique z/OS address space. The zCX address space represents a virtual server. The virtual server does not require operational tasks from the z/OS administrator. From an operational perspective, the virtual server is transparent to the z/OS environment. z/OS Communications Server TCP/IP provides network communications for the zCX workloads.

The zCX server (address space) is represented by a unique z/OS application instance Dynamic VIPA. Both IPv4 and IPv6 zCX DVIPAs are defined using VIPARANGE with the ZCX keyword. Each zCX address space can connect to the network using a single IPv4 DVIPA or both a single IPv4 and a single IPv6 DVIPA.

Configuring the zCX server IP address

The zCX job for the docker environment must be configured using z/OSMF workflows. The zCX configuration requires defining various job related parameters, including some network related parameters which includes the virtual server's IP address. The IP address specified in the zCX configuration (workflows) must match (be coordinated with) your z/OS TCP/IP configuration for VIPARANGE ZCX. For more information about using z/OSMF workflows for zCX, see *IBM z/OS Management Facility Online Help* for Configuration Workflow.

zCX DVIPA operational characteristics and considerations

- Dynamic VIPAs created from the VIPARANGE subnet with ZCX will not move from one stack to another stack. The MOVEable parameter is ignored.
- The value of ZCX for a VIPARANGE statement cannot be changed without first removing the existing VIPARANGE statement and then redefining it.
- The SAF parameter can be dynamically changed.
- The creation of zCX DVIPAs is protected by the same authorization functions provided for all VIPARANGE DVIPAs. The zCX instance and the associated user ID of the zCX started task must adhere the VIPARANGE authorization requirements. Refer to the z/OSMF workflows for configuring your zCX job.
- zCX DVIPAs are controlled by internal processing related to the zCX workload. When the zCX server is started, the DVIPA will automatically be activated and added to the home list. TCP/IP creates a static route to the zCX DVIPAs for its own internal use. This static route is local to and controlled by the stack that owns the zCX instance and it is not advertised using dynamic routing. Native z/OS applications cannot bind to a zCX DVIPA and IOCTLs cannot change the state of a zCX DVIPA.
- If IP filters are defined, updates to the IP filter rules are required for all zCX DVIPAs. Two IPSec rules are required for all of the zCX DVIPAs, one rule defining the zCX DVIPAs as the source and another rule for the zCX DVIPAs as the destination. Both rules must be defined with ROUTING EITHER, permitting both ROUTED and LOCAL traffic for all of the zCX DVIPAs.

zCX DVIPA resiliency

There are usage scenarios where the zCX job (workload) will need to be relocated to another instance of z/OS. For this reason, the zCX DVIPA should be defined in all z/OS systems that could become eligible to host this zCX workload. The zCX DVIPA can only be active (used) within a single instance of z/OS at one time. However, using the z/OS Dynamic VIPA support, the zCX job could terminate (planned or unplanned) and then be restarted in another z/OS instance without reconfiguring the zCX job or z/OS TCP/IP.

When using dynamic routing, the zCX DVIPAs must be added to the OMPRoute configuration. The DVIPAs are treated by OMPROUTE as normal application-instance DVIPAs where the DVIPAs are advertised accordingly.

When using static routes you must ensure the zCX DVIPAs can be reached by other hosts by defining static routes on other hosts that need to reach the zCX instance. If a zCX DVIPA is moved to another z/OS, updates to the static route definitions will be required.

Considerations for the number of containers and zCX instances

Maximum number of active Docker containers per zCX instance

A zCX instance can have up to 1000 active Docker containers. Prior to PTF OA59943, zCX instances could support only up to 200 active Docker Containers.

The zCX Docker CLI SSH container is always active and is included in the total number of active containers in a zCX instance.

Deploying multiple zCX instances

You may want to consider deploying multiple zCX instances in your z/OS system if:

- You want complete isolation of applications
- You have applications or want instances with different business and performance priorities, such as unique WLM service classes
- You would like to cap resources such as CPU and memory allocated for related workloads

Each zCX instance has individually assigned storage, network, and memory resources. However, they share CPU resources across instances and other z/OS workloads. Resource access can be adjusted with configuration and WLM policy controls. The maximum number of active zCX instances in a z/OS system is 64.

Workload management considerations

zCX instances are managed by the z/OS Workload Manager (WLM) based on policy service class goal definitions. Optionally, you can cap CPU and real memory resources using WLM resource group association. CPU and memory resource capping for zCX may require additional capacity planning and performance analysis. In addition to this section, use the z/OS MVS Initialization and Turning Guide for information on planning real and auxiliary storage resources.

Defining WLM service classes for zCX address spaces

One or more service classes should be created for your zCX instances. The service class should have a single performance period specified with a velocity goal. Response time goals are not appropriate for zCX instances, as WLM is not aware of transactions executing inside an instance.

Multiple zCX instances can share the same service class if they have similar performance objectives and host similar workloads, such as Docker containers. The service class should have a specified importance level to indicate the importance of meeting the service class goals compared to other workloads and service classes running on the same z/OS system or sysplex. The performance goals and importance levels may need to be adjusted over time to ensure that the zCX instances have sufficient access to CPU resources.

Exploiting zIIP and general purpose processors for zCX workloads

If sufficient zIIP processors are available (greater than or equal to the number of zCX virtual processors), the majority of zCX instance processing can execute on zIIPs. However, if the zIIP processors are heavily used or constrained, zCX virtual processors may be dispatched on general purpose processors depending

on the configuration of the z/OS system. Dynamic changes to IIPHONORPRIORITY at the system or service class level will immediately take effect for zCX.

zCX workloads, like all others, require capacity planning for memory (fixed and auxiliary), standard CPU, and zIIP CPU to ensure that the system-wide performance, availability, and resource goals are met. More information on using zIIP processors can be found in the **Using System z Integrated Information Processor (zIIP)** section of **z/OS MVS Planning: Workload Management** in **z/OS MVS**.

System level option:

- Parameter IIPHONORPRIORITY in parmlib member IEAOPTxx controls whether zIIP-eligible work is allowed to overflow to standard processors when there is insufficient capacity on zIIPs. The default is YES. Specifying NO means standard processors will not process zIIP-eligible work unless they are required to resolve contention for resources. zIIPs that are overworked may struggle to offload work to standard processors in a timely manner.

Service class level:

- Specifying NO in the Honor Priority field when defining a service class in the WLM service definition explicitly prevents the overflow of zIIP-eligible work to standard processors. There is an exception if contention for resources with standard processor work must be resolved. Specifying DEFAULT indicates that the current value of the IIPHONORPRIORITY parameter in parmlib member IEAOPTxx is used for the zCX work in the service class when there is insufficient capacity on zIIPs.

Resource capping for zCX address spaces

You can optionally limit the amount of CPU and real memory that zCX instances can consume through the use of WLM resource groups or tenant resource groups. When limiting CPU consumption, consider whether the limit should apply to standard processors only or include specialty processors as well. You can include specialty processor usage by specifying the "Include Specialty Processor Consumption" attribute while defining resource groups or tenant resource groups for zCX. If overflow to standard processors is enabled for zCX workloads and you only want to limit the amount of overflow onto standard processors, you can accomplish this by specify a limit and indicating that specialty processor usage is not included. If you are enabling resource groups or tenant resource groups for sub-capacity licensing of IBM software deployed in zCX there are some additional considerations, see Using the IBM License Metric Tool for sub-capacity pricing.

You can cap real memory (auxiliary storage cannot be capped) by specifying the amount of real storage that service classes associated with a resource group can consume at any given time. zCX instances can potentially consume a lot of memory, so capping should be strongly considered. Instances are capped by specifying the memory pool size of the related service classes. When the resource group reaches a threshold slightly below the specified size, the system pages out real storage to limit usage. Therefore, the limit should be above what was configured for the zCX instances. During initialization of a zCX address space, zCX determines if the allocation of fixed real memory will impact the system or resource group. A determined critical impact will prevent the zCX instance from initializing.

- For resource groups: The service class associated with the zCX instance must specify the name of the associated resource group.
- For tenant resource groups: An associated tenant report class must be defined when specifying the tenant resource group. WLM classification rules must specify the tenant report class association for the zCX instance.

Classification rules for zCX

WLM classification rules are needed to associate the zCX instances with a service class and optionally a report class. zCX instances can be classified under Subsystem type STC. Qualifiers *job_name* or *userid* can be used to identify zCX instances.

Using the IBM License Metric Tool for sub-capacity pricing

You can use the IBM License Metric Tool for sub-capacity pricing of Linux on Z IBM SW (Passport Advantage products) that are deployed in zCX. There is an IBM License Metric Tool disconnected scanner embedded in zCX that can be enabled alongside an IBM License Metric Tool server to scan and generate a report on a zCX instance. You can set up a new IBM License Metric Tool server or use an existing one for zCX use.

For general information on sub-capacity licensing, refer to: https://www.ibm.com/software/passportadvantage/subcaplicensing.html%C2%A0

For General Counting rules for sub-capacity licensing, refer to: https://www.ibm.com/software/passportadvantage/Counting_Software_licenses_using_specific_virtualization_technologies.html

Enabling IBM License Metric Tool using zCX workflows

You can enable IBM License Metric Tool when provisioning a new zCX instance or by reconfiguring an existing one. When running either the provisioning or reconfiguration workflow, set the z/OSMF variable `ZCX_ILMT_ENABLE` to TRUE. The default value is FALSE, and can be reset to FALSE to disable an active scanner.

IBM License Metric Tool Scanning

Once enabled, the IBM License Metric Tool disconnected scanner will periodically scan for hardware configuration and software licenses in the zCX instance. During these scans it will collect the following information about any processors available to the zCX instance:

- Number of zIIPs and standard processors available to the zCX instance
- Number of processors available to the z/OS system
- Additional virtualization layers (e.g. z/VM if running z/OS as a guest)
- Centralized Processor Complex (CPC)

This information is used to determine the Processor Value Units (PVU) for licensing the IBM software running in zCX. The hardware scan and software scan will first occur when IBM License Metric Tool is enabled, as well as at any subsequent start of a zCX instance. The hardware scans will then run twice every hour at minute 0 and minute 30. The software scans will run every 7 days at approximately the time of the first scan of the zCX instance (for example, if the zCX instance was first started on a Monday at 8 AM, a software scan will run every Monday at 8 AM).

Restriction: If you deploy and remove an instance of IBM software in zCX within a single software scan interval, the IBM License Metric Tool disconnected scanner will not collect or report information on it.

For more details on the IBM License Metric Tool disconnected scanner, refer to:https://www.ibm.com/support/knowledgecenter/SS8JFY_9.2.0/com.ibm.lmt.doc/Inventory/planinconf/c_disc_sys_main.html

Capping processor usage for licensing purposes with z/OS Workload Manager (WLM)

z/OS Workload Manager (WLM) resource group or tenant resource group maximum processor limits can be used to cap processor usage for one or more zCX instances. To specify maximum usage, specify the resource group type 3 limits. These limits express the number of processors as a numeric value that is 100 times the true number of processors (e.g. 100 represents 1 processor, 200 represents 2 processors, and so on). If the maximum is specified under any other resource group type, WLM will enforce the limits but this information will not be reported to IBM License Metric Tool by the scanner.

More information on capping processor usage with WLM can be found here: https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.4.0/com.ibm.zos.v2r4.ieaw100/rgr.htm

Managing IBM License Metric Tool scanner data

Once the IBM License Metric Tool scanner data is retrieved and uploaded to the IBM License Metric Tool server, it should be removed from the zCX SSH CLI container shared directory. You can remove it manually or automatically. To remove the files manually, periodically log onto the zCX shell using SSH and copy the files with an SSH File Transfer Protocol (SFTP). You can provide your own automated process to perform these steps, or use the IBM License Metric Tool-provided sample automation based on Ansible. The Ansible samples will upload the scan results from the endpoints to the IBM License Metric Tool server and then delete the IBM License Metric Tool disconnected scanner data files stores in the zCX SSH CLI container shared directory.

More information about these samples, which are provided for convenience and not formally part of the IBM License Metric Tool product, can be found here: https://github.com/IBM/ansible-automation-for-lmt

Local user access to IBM License Metric Tool scanner data

A zCX user must have access to the IBM License Metric Tool disconnected scanner to retrieve and manage the data. If you are using local user management for zCX users, you can use an existing zCX admin's credentials or create a new user specifically for this purpose. The user will need to access the zCX SSH CLI container that listens on port 8022. You can use the zCX admin credentials to create additional zCX users with the password option. You can also upload the public SSH keys so the zCX user can use them to authenticate into the zCX SSH CLI container and retrieve the data.

LDAP user access to IBM License Metric Tool scanner data

A zCX user must have access to the IBM License Metric Tool disconnected scanner to retrieve and manage the data. If you are using LDAP user management for zCX users, you can use either an existing zCX administrator's credentials or another zCX user defined with the password or SSH key option to retrieve the data.

Interpreting the IBM License Metric Tool scanner data

IBM License Metric Tool disconnected scanner results are located in the zCX SSH CLI at `/media/azd_shared_volume/IBM/ILMT`.

There are one or more compressed .tar archive files for software scan results with the format: `YYYYMMDDHHMM-hostname-timestamp.tar.gz`

- *YYYYMMDDHHMM* is the date/time stamp of the scan
- *hostname* is the zCX instance name
- *timestamp* is the Linux timestamp of the IBM License Metric Tool installation
- For example: `201911111449-zcxrlk-1573462032.tar.gz`

There are three logs available about the IBM License Metric Tool disconnected scanner:

- `zcx_ilmt_hw_scan.log` is the log of the last IBM License Metric Tool hardware scan.
- `zcx_ilmt_setup.log` is the log of the IBM License Metric Tool disconnected scanner installation.
- `zcx_ilmt_sw_scan.log` is the log of the last IBM License Metric Tool software scan.

The following files and directories are available for help debugging IBM License Metric Tool issues:

- `zcx_ilmt_qc_test.log` is the log that contains the output from the last Query Capacity (qclib) qc_test program invocation.
- IBM License Metric Tool disconnected scanner "config" directory (read-only access)
- IBM License Metric Tool disconnected scanner "logs" directory (read-only access)
- IBM License Metric Tool disconnected scanner "work" directory (read-only access)

Prerequisites and Dependencies for IBM License Metric Tool

Table 6. Required z/OS 2.4 APAR/PTF sets for IBM License Metric Tool enablement with zCX

APAR	PTF
oa58587	UJ01572
oa58601	UJ01571
oa58621	UJ01576
oa58598	UJ01574
oa58599	UJ01575
oa58600	UJ01577

IBM License Metric Tool enablement for zCX also requires an IBM License Metric Tool server at version 9.2.18 or higher.

If z/OS and zCX instances are running as a z/VM guest:

- z/VM 7.1 is required
- PTF UM35568 for APAR VM66329 is required

Chapter 5. Security for zCX

Started task ID

Define a GLZ started procedure in the PROCLIB concatenation. zCX provides a sample GLZ procedure in *SYS1.PROCLIB*. You can have multiple procedures to eliminate the need for *CONF=* and *JOBNAME=* parameters on each START GLZ command.

Update the security server:

1. Define a z/OS user ID under which the zCX instances will run, and permit the user ID to the GLZ started procedure(s). This can be a single user ID for all zCX instances or a set of user IDs.
2. Permit the user ID to create zCX Dynamic VIPAs (EZB.MODDVIPA.*.*). This is required in the following cases:
   - If the EZB.MODDVIPA.*.* SERVAUTH class profile is defined to restrict access to all VIPARANGE DVIPAs. If an existing user id is being used for this zCX instance, then no additional definitions are needed.
   - If the SAF keyword was specified when creating the VIPARANGE zCX statement in the TCP/IP profile. If the SAF keyword specifies a new resource name, then you may need to also create a unique profile if there is not a generic profile already covering that resource name.

If SAF-based security is not enabled for DVIPA creation, then the userID associated with the zCX started task will require one of the following:

- A UID(0) specification in its OMVS segment
- READ access to a BPX.SUPERUSER profile if that is defined on the system

Local and LDAP user management

User management for the zCX Docker CLI can be optionally integrated with your z/OS defined users using LDAP-based authentication. There is also the option for user management through a local registry. LDAP-based authentication can be integrated with RACF or other compliant security manager products by using the IBM® Tivoli® Directory Server for z/OS®. You should decide what method of user management you will use for zCX prior to provisioning, although you can switch between the types after implementation. More information can be found in the User Management chapter.

Setting up pervasive encryption for zCX data sets

By default, zCX instance root file systems and swap data volumes backed by VSAM linear data sets are 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.

VSAM linear data sets will use the encryption key labels if specified with one of the above methods before the data set is created.

In addition to providing the encryption key label, the zCX instance directory zFS file system must be enabled for encryption in one of three ways:

- Use the global `format_encryption=on` option in the IOEFSPRM configuration.
- Set the zCX z/OSMF variable *ZCX_ZFS_ENCRYPT* to TRUE in the zCX provisioning workflow.
- Manually issue the `zfsadm encrypt` command after successfully provisioning the zCX instance.

Only VSAM data sets defined with the extended format option are eligible to be encrypted. While zFS does not require aggregates to be defined with the extended format option to be encryption-eligible, zCX does. For zFS filesystem encryption, all members of the system's sysplex must be z/OS V2R3 or higher.

Use the following resource for more information on getting started with encryption:

- Redbooks: Getting Started with z/OS Data Set Encryption
  - https://www.redbooks.ibm.com/redbooks/pdfs/sg248410.pdf
- zFS Administration Guide in the IBM Documentation
- Using the z/OS data set encryption enhancements in the IBM Documentation
  - https://www.ibm.com/support/knowledgecenter/SSLTBW_2.3.0/com.ibm.zos.v2r3.idak100/encryption23.htm

Chapter 6. User management

User access to the Docker CLI can be managed using either local user management or central LDAP server-based user management. You can also change the user management technique of a provisioned zCX instance using the “Reconfiguration workflow” on page 57

Using a local user management within a zCX instance

Using this method, a Docker administrator user ID is specified during provisioning of a zCX instance. The Docker administrator has access to the Docker CLI, as well as the ability to define and delete additional zCX users. This approach is simple, and therefore useful when conducting initial testing in zCX or when a limited number of zCX instances are deployed. However, it requires that all authorized users are defined and maintained on each individual zCX instance; there is no sharing of user access across zCX instances.

Using an LDAP server for authorization across zCX instances

Using this method, the z/OS system programmer that is executing the z/OSMF provisioning workflow specifies the target LDAP server that should be used for authorization and authentication of the zCX Docker CLI users. This approach allows for consolidated user management for the zCX Docker CLI in your enterprise. It also allows you to optionally integrate your zCX Docker CLI user management with your z/OS defined users by providing two options for LDAP based authentication:

- IBM® Tivoli® Directory Server for z/OS®. This LDAP server allows optional integration with RACF or other compliant security manager products using the SDBM so that you can authorize users to zCX using their existing z/OS users and credentials.
- An LDAP server, such as OpenLPAP, in your enterprise.

The remainder of this chapter provides instructions to proceed with either user management method.

Local user management

zCX instance Docker administrator

The zCX instance Docker administrator is responsible for managing Docker users within a zCX instance. The user information for the Docker administrator must be given during provisioning. The Docker administrator does not necessarily need to be a z/OS user or zCX appliance administrator.

The Docker administrator user ID is represented by z/OSMF workflow variable *ZCX_DOCKER_ADMIN*. The Docker administrator user ID's ssh key must be give during provisioning and is represented by z/OSMF workflow variable *ZCX_DOCKER_ADMIN_SSH_KEY*. You can use `ssh-keygen` utility to generate public/private rsa key pairs. By default, the keys are stored in your home directory under the .ssh directory. You can cut and paste the contents of public ssh key (id_rsa.pub) in the *ZCX_DOCKER_ADMIN_SSH_KEY* variable field.

- For example, you can use the following command to generate a private/public key pair on your client system:

  ```
  ssh-keygen -t rsa -b 4096 -C "your_email@domain.com"
  ```

  You can optionally provide a passphrase to further secure private/public key access.

If you experience login issues from ssh client, then use `ssh -v option` for additional debugging information.

If you have previously used a different private/public key pair or a deprecated password authentication option with the same zCX appliance instance IP address, then you must remove the entry from the `known_hosts` file under the .ssh directory.

Adding Docker Users

We suggest allowing the administrator to create Docker users. These Docker users have their own home directories, user names, and passwords.

Docker users have only the ability to run Docker commands. This is because they are added to the Docker user group. Docker users do not have Sudo, the program that allows users to use elevated privileges. Therefore, they cannot create or modify other Docker users.

The `useradd` and `usermod` commands are not supported on by zCX.

To create a new Docker user with user name *username* and add the new user to the existing group Docker, the administrator runs the command:

```
sudo adduser --ingroup docker username
```

and receives the following output:

```
Adding user 'username' ...
Adding new user 'username' (1004) with group `docker' ...
Creating home directory /home/username' ...
Copying files from `/etc/skel' ...
```

The administrator is then prompted with the following:

```
Enter new UNIX password:
Retype new UNIX password:
```

Here, the administrator assigns the new user a password. The new user can change the password if desired. Once the password is entered, the administrator receives the output:

```
passwd: password updated successfully
Changing the user information for username
```

The administrator is then prompted to provide the following information about the new user. This is typical requested information for creating a new user on a system:

```
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] y
```

The administrator can force a new Docker user to change the user password at log in using the following command:

```
$sudo passwd -e username
```

Once created, the new Docker user can SSH into the container and use Docker with the command:

```
ssh username@ip_address -p 8022
```

If the administrator forces the new Docker user to set a new password, the new Docker user will be prompted to do so after login in. A Docker user can also change his or her password at any time with the following command:

```
username@6c279b325214:~$ passwd
```

Which will give the output:

```
Changing password for username.
(current) UNIX password:
Enter new UNIX password:
Retype new UNIX password:
```

Followed by:

```
passwd: password updated successfully
```

Adding Docker administrators

Additional Docker administrators can be created by an existing Docker administrator adding the users to the groups "Docker" and "sudo". Begin by issuing the following command to add a new Docker user:

```
$ sudo adduser --ingroup docker username
```

Then add the user to the "sudo" group to grant administrative access:

```
sudo adduser username sudo
```

The administrator can then switch to the new user using the command:

```
$ su username
```

And confirm the user is in the appropriate 'docker' and 'sudo' groups:

```
 username@host:/home/username$ groups
```

Which should provide the output:

```
docker sudo
```

Removing users from groups

A user can be removed from a group by issuing the following command:

```
sudo deluser username groupname
```

Administrator privileges can be revoked by removing a user from the "sudo" group.

Deleting Docker users

A user and the corresponding home directory can be deleted by issuing the following command:

```
sudo userdel -r username
```

To view additional options, use:

```
$ sudo userdel -h
```

Preparing an LDAP configuration file

The following describes how to prepare an LDAP client configuration file using posixGroup LDAP groups and posixAccount user accounts. If using LDAP user management, the configuration file is needed when provisioning a zCX instance.

An LDAP client configuration file should define the LDAP server URI using the *uri* variable as `ldap://host:port`. If no port is specified, 389 will be used by default.

- The distinguished name of the search base is defined by the variable *base*.
- If server has TLS support, *TLS_CACERT file_name* specifies the file that contains certificates for all of the certificate authorities. During provisioning, zCX copies the certificate file to the directory `/etc/ldap` and renames it to `ldap-ca.crt`. Therefore, the *TLS_CACERT* setting in the LDAP configuration file should be `/etc/ldap/ldap-ca.crt`.

- The LDAP group to which log in access will be restricted is defined by the variables *pam_groupdn* and *pam_member_attribute*
- The TLS_CACERT file must be in EBCDIC code page on z/OS. This file must be located on the same z/OS system as the zCX instance and must be in a UNIX file system accessible to the provisioning user through z/OSMF.

`BINDDN <distinguished_name>` specifies a user name with read and search access to the LDAP directory. The `BINDDN` parameter specifies the distinguished name (DN) to use when performing LDAP operations. If not specified, LDAP operations are performed using an anonymous identity or the identity of each zCX user that is allowed to log in using LDAP authentication. Only specify `BINDDN` if the LDAP server does not permit anonymous search for users in the directory. Most LDAP servers allow anonymous access by default. If you do need to use the BINDDN parameter, create a read-only account with read and search access on LDAP server. Use the newly created *distinguished_name* in the `ldap.conf` file.

`BINDPW` specifies the password to use when performing LDAP operations. This is typically used in conjunction with the `BINDDN` parameter. The password is in clear text in the `ldap.conf` file.

Only one LDAP group can be given access to a zCX instance. If a user needs access to more than one zCX instance, that user should be added to each corresponding group.

Example LDAP configuration file

The following is a basic example of how to define an LDAP client configuration file:

```
# Here is a sample ldap.conf file to allow a zCX appliance to authenticate users
# by using an LDAP server.

uri ldap://hostname
base dc=example,dc=com
ldap_version 3

# LDAP searches will be performed using the anonymous identity by default

# The file that contains certificates for all of the certificate authorities
tls_cacertfile /etc/ldap/ldap-ca.crt
# The LDAP group to allow access to this zCX instance
pam_groupdn cn=zcxldapgroup,ou=Groups,dc=example,dc=com
pam_member_attribute memberUid
```

The above sample restricts access to members of LDAP posixGroup *zcxldapgroup*, whose members are specified in the *memberUid* attribute. See additional attributes you can define in an LDAP configuration file: http://manpages.ubuntu.com/manpages/trusty/man5/pam_ldap.5.html

Example of setting IPv6 LDAP support

Here is a sample ldap.conf file to allow a zCX instance to authenticate users by using an LDAP server with an IPv6 address. If IPv6 address is used, it should be in square brackets. LDAP searches will be performed using the anonymous identity by default. The file that contains certificates for all of the certificate authorities is `tls_cacert /etc/ldap/ldap-ca.crt`.

```
uri ldap://[IPv6_address]:port
base dc=example,dc=com
ldap_version 3
```

Example of LDAP user management for zCX

This example demonstrates how to set up a posixGroups and posixAccounts for LDAP user management. This scenario is based on OpenLDAP.

Assume you have two zCX instances (*zcxappliance1* and *zcxappliance2*) and three zCX users (*zcxuser1*, *zcxuser2*, and *zcxadmin*). You want to give *zcxuser1* access to only *zcxappliance1* and *zcxuser2* access to only *zcxappliance2*. You want to give *zcxadmin* access to both *zcxappliance1* and *zcxappliance2*.

The OpenLDAP tree structure for this scenario is as follows:

1. dc=example,dc=com
    a. ou=Groups
        i) cn=zcxgroup1
        ii) cn=zcxgroup2
    b. ou=People
        i) uid=zcxadmin
        ii) uid=zcxuser1
        iii) uid=zcxuser2

Here, *zcxgroup1* and *zcxgroup2* are posixGroups set up for access to *zcxappliance1* and *zcxappliance2*, respectively. User *zcxuser1* will be a member of *zcxgroup1* and *zcxuser2* will be a member of *zcxgroup2*. Since *zcxadmin* should be allowed to access both *zcxappliance1* and *zcxappliance2*, *zcxadmin* will be a member of both *zcxgroup1* and *zcxgroup2*.

Below is an example of an LDAP entry of object class posixGroup that adds *zcxuser1* and *zcxadmin* to *zcxgroup1*:

```
dn: cn=zcxgroup1,ou=groups,dc=example,dc=com
objectClass: top
objectClass: posixGroup
cn: zcxgroup1
gidNumber: 1001
memberUid: zcxuser1
memberUid: zcxadmin
```

- *gidNumber*: the Linux GID number of the group. All gidNumbers must be unique and in the range of 1000-59999.
- *cn*: the name of the group
- *memberUid*: identifies users of the group

Similarly, *zcxgroup2* can be defined as:

```
dn: cn=zcxgroup2,ou=groups,dc=example,dc=com
objectClass: top
objectClass: posixGroup
cn: zcxgroup2
gidNumber: 1002
memberUid: zcxuser2
memberUid: zcxadmin
```

The entries above can be added to the LDAP server by saving the group definitions as LDIF files and using the `ldapadd` command.

The posixAccount entries for each user can be similarly created and added. Here is an example definition for *zcxuser1*:

```
dn: uid=zcxuser1,ou=People, dc=example,dc=com
objectClass: shadowAccount
objectClass: posixAccount
objectClass: inetOrgPerson
cn: zcxuser1
gidNumber: 1001
homeDirectory: /home/zcxuser1
sn: zcxuser1
uid: zcxuser1
uidNumber: 2001
displayName: zcxuser1
gecos: zcxuser1
givenName: zcxuser1
loginShell: /bin/bash
```

- uid: user's Linux login ID
- uidNumber: the user's Linux UID. All uidNumbers must be unique and in the range of 1000-59999.
- gidNumber: the primary group of this user

- homeDirectory: the location of the home directory that will be created the first time the user logs in to the zCX instance.

The entries above can be added to the LDAP server by saving the user definitions as LDIF files and using the `ldapadd` command. Use the `ldappasswd` command to set the password for an LDAP user.

Restricting LDAP login by group

All users defined in the search based on an LDAP configuration file will have access to a zCX instance unless otherwise defined by the *pam_groupdn* attribute. Continuing with the previous example, two lines can be added to the *zcxappliance1* configuration file to allow only members of *zcxgroup1* access:

```
pam_groupdn cn=zcxgroup1,ou=Groups,dc=example,dc=com
pam_member_attribute memberUid
```

Similarly, the following lines can be added to the *zcxappliance2* configuration file to allow only the users of *zcxgroup2* access:

```
pam_groupdn cn=zcxgroup2,ou=Groups,dc=example,dc=com
pam_member_attribute memberUid
```

IBM Tivoli Directory Server for z/OS

For more information, see the IBM Tivoli Directory Server for z/OS with RACF back-end in the IBM Documentation.

User management reconfiguration

You can change user management control between local and LDAP-based user management or from one LDAP server to another using the “Reconfiguration workflow” on page 57. This section provides guidance for the different reconfiguration options.

Switching from local user management to LDAP-based user management

You can use the reconfiguration workflow to switch from local to central LDAP-based user management for a provisioned zCX instance. To preserve access and data for the existing local users (including the default administrator), you should define users and groups with corresponding attributes in the LDAP server.

The login access of the default administrator user will be removed when switching from local to LDAP-based user management because the local administrator role and capabilities are not needed to manage users centrally. However, the home directory of the administrator will remain intact. If the zCX instance has no other users, you can follow the instructions in Example of LDAP user management for zCX to define groups and users in your LDAP server.

If you want preserve the access and data of additional users and groups, you should recreate them in the LDAP server with the same attributes they had locally. (LDAP groups should have the same names and group IDs. LDAP users should have the same user names, LDAP *uidNumber*s corresponding to local *uid*s, primary group assignments, and home directory settings.)

For example, imagine a zCX instance switching to OpenLDAP with existing local group `appliance1` and users `user1` and `user2`. The users and group have the following attributes:

- Local group `appliance1` has a group ID (*gid*) of `1004`
- Local user `user1` has a user ID (*uid*) of `5001`, primary group of `appliance1`, and home directory of `/home/user1`
- Local user `user2` has a user ID (*uid*) of `5002`, primary group of `appliance1`, and home directory of `/home/user2`

Ignoring all other groups and users, the directory tree structure in the LDAP server is:

- dc=example,dc=come
  - ou=Groups
    - cn=appliance1
  - ou=People
    - uid=user1
    - uid=user2

The posixGroup `appliance1` can be defined in the LDAP server as follows:

```
dn: cn=appliance1,ou=groups,dc=example,dc=com
objectClass: top
objectClass: posixGroup
cn: appliance1
gidNumber: 1004
memberUid: user1
memberUid: user2
```

Note that the *gidNumber* matches that of the *gid* number of the local group `appliance1`.

The posixAccount `user1` can be defined in the LDAP server as follows:

```
dn: uid=user1,ou=People, dc=example,dc=com
objectClass: shadowAccount
objectClass: posixAccount
objectClass: inetOrgPerson
cn: user1
gidNumber: 1004
homeDirectory: /home/user1
sn: user1
uid: user1
uidNumber: 5001
displayName: user1
gecos: user1
givenName: user1
loginShell: /bin/bash
```

The attribute *uid* corresponds to Linux login id while the *uidNumber* is user's Linux uid. The *uidNumber* should match that of local user `user1` (`5001` in this example). The *gidNumber* corresponds to this user's primary group (`1004` in this example). The attribute *homeDirectory* should also match the local home directory of the user.

User entries can be added to LDAP server by saving the user definitions in LDIF files and using the `lparadd` command. Passwords for LDAP users can be set using the `ldappasswd` command.

Switching from LDAP-based user management to local user management

You can use the reconfiguration workflow to switch from LDAP-based to local user management.

The home directories of the LDAP users who had access to the zCX instance will remain intact. For the same users to have access to the reconfigured zCX instance, the administrator must individually create local user accounts with corresponding attributes to those defined on the LDAP server.

For example, imagine a user with account name `user1` defined in the LDAP with the following attributes:

- `user1` has a user ID of `5001`
- `user1` has the LDAP server home directory `/home/user1`
- `user1` is part of the group `appliance1` with a group ID of `1004`

The administrator should execute the following steps to ensure that the user can log into the reconfigured zCX instance with access to the home directory created during their first LDAP-based login:

1. Create the local group `appliance1` with groups ID (*gid*) of `1004`

   ```
   sudo groupadd  -g 1004 appliance1
   ```

2. Create the local user `user1` with the attributes identical to those defined in the LDAP server:

```
sudo adduser --uid 5005 --ingroup docker --home /home/user1 user1
sudo adduser user1appliance1
```

An additional step is required when a zCX instance is reconfigured from local to LDAP-based user management, and then back to local user management. The passwords of the original local users are deleted when the zCX instance is reconfigured from local to RPL-based user management. Therefore, when it is switched back to local user management, an administrator must set the passwords for the local users.

Switching between LDAP servers

You can use the reconfiguration workflow to switch user management from one LDAP server to another.

If the new LDAP server uses the exact same directory structure as the existing server, then all user groups will be maintained and home directories will remain accessible, as in a failover scenario. However, if the new LDAP server has a different directory structure, then current users and groups do not exist in the new server and will be unable log in after reconfiguration. The home directories of these users will remain intact. For access to the new server, the groups and users should be recreated in the new server with the same LDAP entry attributes. If only some of the users are defined on the new LDAP server, only they will have to access the reconfigured zCX instance.

Chapter 7. Docker registries

A Docker registry is a storage and distribution system for Docker images. Users must connect to a registry in order to load a Docker image onto a zCX instance. zCX supports one secure and one insecure registry.

Docker Hub is an external registry for finding and sharing container images. zCX can support base images in Docker hub that are tagged as "IBM Z" or "s390x".

You can also host a private, or local, Docker registry to account for security, network availability, and performance.

Insecure Docker registry

When connected to an insecure private registry, the Docker daemon in a zCX instance will disregard security for your private registry. Communication to an insecure Docker registry is not protected. Therefore, it is only recommended that an insecure Docker registry be used for performing isolated tests.

Secure Docker registry

Connection of a zCX instance to a secure Docker registry is established by connecting to a registry with TLS authentication. To configure the connection, you must provide the domain name or IP address of the registry and (if the registry requires access through a specific port) the port number during provisioning. The workflow also requires the TLS certificate authority (CA) certification. These must be located on the same z/OS system as the zCX instance and must be in a UNIX file system accessible to the provisioning user through z/OSMF.

Table 7. z/OSMF workflow variables for a secure Docker registry

Variable	Description
ZCX_SECURE_DOCKER_REGISTRY_ENABLE	This variable indicates whether the secure Docker registry with TLS authentication is enabled. When set to TRUE, the IBM zCX appliance instance will be configured to use TLS CA authentication for secure Docker registry communication.
ZCX_SECURE_DOCKER_REGISTRY_IP	This is the domain name or IP address of the secure private Docker registry. This configures the IBM zCX appliance instance to use the TLS authentication provided for communication with the secure private Docker registry. TLS authentication will protect communication to the secure private Docker registry. Check with the Docker registry administrator to obtain the information. To use IPv6 addressing capability, the variable *ZCX_GUESTIPV6* must be set. Specify a domain name that resolves to IPv6 address.
ZCX_SECURE_DOCKER_REGISTRY_PORT	This is the port number that will be used to access the private Docker registry. This variable is only necessary if the private Docker registry requires a specific port to be used. The Docker daemon running in the IBM zCX appliance will use the domain name or IP address and port specified to connect to the secure Docker registry.

Table 7. z/OSMF workflow variables for a secure Docker registry (continued)	
Variable	**Description**
ZCX_DOCKER_REGISTRY_TLS_CA_CERT	This is the absolute path to the secure Docker registry TLS VA Certificate. This should be the same CA certificate that signed the secure Docker registry server certificate. The TLS CA certificate file must be in EBCDIC code page on z/OS. This file must be located on the same z/OS system as the IBM zCX appliance and must be in a UNIX file system accessible to the provisioning user through z/OSMF.

If you are using a universal binary repository manager to manage Docker registries, then make sure you provide the Docker registry endpoint as a secure registry IP/Domain name rather than the web interface of the binary repository manager.

Building a private registry

Since the official Docker registry image does not run on the IBM Z platform, you can build one using a Docker file customized for Z. The following instructions describe how to build a private Docker registry image for zCX.

1. Create a directory within your home directory on zCX. We recommend naming the directory `myregistry`.
2. Download the registry image Docker build file to the directory created in step 1. The build file is available at https://github.com/linux-on-ibm-z/dockerfile-examples/blob/master/DockerDistribution/Dockerfile.
3. Run the build command:

   ```
   docker build -t myregistry .
   ```

   When the build is complete you will see "*Successfully tagged myregistry:latest*".
4. Run

   ```
   docker images
   ```

   and you should see the Docker image *myregistry* in the list of images.

Run the following command to start your private registry server that listens to port 5000:

```
docker run --name myregistry-server -p 5000:5000 -d myregistry
```

You can run the `docker ps` command to check that the registry is up and running.

You can find more details on running and restricting access to a secure private registry in the Docker registry resources: https://docs.docker.com/registry/

Docker proxy support for zCX instances

This section describes how to configure your zCX instance to use a proxy server. Both the Docker daemon and the Docker client will require independent proxy configurations in order to use the proxy services.

Docker daemon proxy configuration

Use this setup if your Docker daemon needs to access external websites or locations (such as external Docker hub) that may be protected or disabled by corporate firewalls. The environment variables *HTTP_PROXY*, *HTTPS_PROXY*, and *NO_PROXY* are used to configure proxy services for the Docker daemon. These variables are set during provisioning and can be changed through the reconfiguration workflow.

Table 8. Workflow variables for Docker proxy configuration			
Variable	**Description**	**Required?**	**Default (if applicable)**
ZCX_CONFIGURE_PROXY	This variable must be specified as TRUE to configure proxy for the Docker daemon. If the value is FALSE, the proxy configuration workflow variables *ZCX_HTTP_PROXY*, *ZCX_HTTPS_PROXY*, and *ZCX_NO_PROXY* are ignored.	Yes	FALSE
ZCX_HTTP_PROXY	This variable specifies the value of the environment variable *HTTP_PROXY*. The input value is the address and port of the proxy server. For example: `http://10.1.130.50:3001`An IPv6 address can be specified only when *ZCX_GUESTIPV6* variable is specified. For example: `http://[1001:fe5::1:5]:8001`	No, but if *ZCX_CONFIGURE_PROXY* is set to TRUE, then either *ZCX_HTTP_PROXY* or *ZCX_HTTPS_PROXY* must be specified.	
ZCX_HTTPS_PROXY	This variable to specifies the value for the environment variable *HTTPS_PROXY*. This input value is the address and port of the proxy server. For example: `http://10.1.130.50:3001`An IPv6 address can be specified only when *ZCX_GUESTIPV6* variable is specified. For example: `http://[1001:fe5::1:5]:8001`	No, but if *ZCX_CONFIGURE_PROXY* is set to TRUE, then either *ZCX_HTTP_PROXY* or *ZCX_HTTPS_PROXY* must be specified.	
ZCX_NO_PROXY	This variable specifies the value for the environment variable *NO_PROXY*. This input value is hosts or ranges to be excluded from the proxy. For example: `*.test.example.com,.example2.com`	No	

Table 8. Workflow variables for Docker proxy configuration (continued)			
Variable	**Description**	**Required?**	**Default (if applicable)**
ZCX_HTTPS_PROXY_CA_CERT	This variable specifies the location of a CA certificate to use with HTTPS Docker proxy configuration. It must reside in a z/OS UNIX System Services directory accessible to z/OSMF and the user running the workflow. This must be on the z/OS system where the zCX instance is or will be provisioned in EBCDIC encoded format.	No	

Once the zCX instance is started, log in with the admin user ID and issue the `docker info` command to verify the proxy settings. Changes can be made through the zCX Proxy Configuration category of the reconfiguration workflow, which allows you to enable, disable, or update the proxy configuration.

Private CA certificate for HTTPS Docker proxy server

You can install a private CA certificate to use with HTTPS Docker proxy configuration. To install, specify the CA certificate file location in the z/OSMF variable *ZCX_HTTPS_PROXY_CA_CERT* field during provisioning or with the reconfiguration workflow on an existing zCX instance. This should be the same CA certificate that signed the HTTPS proxy server certificate. It must reside in a z/OS UNIX System Services directory accessible to z/OSMF and the user running the workflow. This must be on the z/OS system where the zCX instance is or will be provisioned in EBCDIC encoded format. For example, the certificate could reside at `/directory/proxy_CA.crt`

Remove the private CA certificate by providing an empty value for the z/OSMF variable *ZCX_HTTPS_PROXY_CA_CERT* in the zCX reconfiguration workflow.

Docker client proxy configuration for Docker containers

Use this setup if your applications running inside Docker containers need to access external websites or locations that may be protected or disabled by corporate firewalls. Docker client proxy configuration can be through either Docker user set up or environment variables. Docker provides information on both these methods at https://docs.docker.com/network/proxy/.

When configuring proxy through Docker user set up, the environment variables will persist for each new container that the user creates or starts. When configuring proxy through environment variables, the variables are set only for the corresponding Docker run instance.

Configure the Docker client using Docker user setup: Once the zCX instance is started, log into the SSHD CLI and set up the proxy configuration using the following steps:

1. Create a `.docker` directory under your user home directory.
2. Create a `config.json` file in the `.docker` directory using the following sample:

```
#Sample content for ~/.docker/config.json
{
 "proxies":
 {
 "default":
 {
     "httpProxy": "http://10.1.130.50:3001",
     "httpsProxy": "http://10.1.130.50:3001",
```

```
    "noProxy": "*.test.example.com,.example2.com"
  }
 }
}
```

The `config.json` file must specify the address and port for the proxy server. The *noProxy* environment variable can be used to exclude hosts or ranges from going through the proxy.

3. Save the `config.json` file.

The proxy variables are now determined for each new container that the user creates or starts.

Configure the Docker client using environment variables: You can specify proxy environment variables with the `--env` flag, which is specified on the `docker run` command. Environment variables can also be specified in a file (`NAME=VALUE` format), and the file can be invoked with the `--env-file` flag on the `docker run` command.

Table 9. Proxy environment variables

Variable	Description	Example
HTTP_PROXY	This variable specifies the address and port of the proxy server.	`docker run --env HTTP_PROXY="http://10.1.130.50:3001"`
HTTPS_PROXY	his variable specifies the address and port of the proxy server.	`docker run --env HTTPS_PROXY="https://10.1.130.50:3001"`
NO_PROXY	This variable specifies the hosts or ranges that are to be excluded from the proxy.	`docker run --env NO_PROXY="*.test.example.com,.example2.com"`

Proxy configuration for SSH CLI container

Use this setup if you need proxy configuration for commands issued inside the zCX SSH CLI container. Commands such as `curl` issued inside the zCX SSH CLI container may require proxy configuration to access external websites or locations that are protected or disabled by corporate firewalls.

Export the following environment variables once the zCX instance is started. You may need to export the upper and lower case environment variables.

Table 10. Environment variables for Docker proxy configuration

Variable	Example

HTTP_PROXY	To set:
	`export HTTP_PROXY="http://127.0.0.1:3001"`
	To remove:
	`unset HTTP_PROXY`

HTTPS_PROXY	To set:
	`export HTTPS_PROXY="http://127.0.0.1:3001"`
	To remove:
	`unset HTTPS_PROXY`

Table 10. Environment variables for Docker proxy configuration (continued)

Variable	Example

NO_PROXY	To set:
	`export NO_PROXY="localhost"`
	To remove:
	`unset NO_PROXY`

http_proxy	To set:
	`export http_proxy="http://127.0.0.1:3001"`
	To remove:
	`unset http_proxy`

https_proxy	To set:
	`export https_proxy="http://127.0.0.1:3001"`
	To remove:
	`unset https_proxy`

no_proxy	To set:
	`export no_proxy="localhost"`
	To remove:
	`unset no_proxy`

Chapter 8. Preparing for zCX workflows

This chapter describes important information to know before using a zCX workflow, as well as the variables used by the workflows. Many of these variables are required for provisioning, and should be specified in the workflow variables input properties file.

Workflow variables input properties file

The workflow variables input properties file contains all the variables and corresponding values required to provision an IBM zCX instance. A sample input properties file of workflow variables is provided under 'Properties Directory' inside the installation directory.

A z/OS system administrator can copy the sample file in to a read/write directory and modify it with system-specific inputs. The system administrator should consult the z/OS storage administrator, z/OS security administrator, and the z/OS network administrator to provide the proper values.

The workflow variables input properties file is optional, as all the inputs can be added in the z/OSMF provisioning workflow directly. If the variables are manually added to the workflow, they can be saved to a specified directory as a workflow variables input properties file for subsequent use. Descriptions of the variables can be found in this document or by clicking the tool tips icons in z/OSMF.

IBM zCX installation directory

The zCX installation directory is the z/OS UNIX path where all zCX code binaries and workflows are installed on your system. The default installation directory location is `/usr/lpp/zcx_zos`.

The zCX installation directory is represented by the input variable `ZCX_INSTALL_DIR`.

Table 11. zCX Appliance Installation Variables

Binary	**Location**
Default	/usr/lpp/zcx_zos
Binaries (1BL and zCX)	/usr/lpp/zcx_zos/IBM
z/OSMF zCX workflows	/usr/lpp/zcx_zos/workflows
Sample Workflows Variables Input Properties File	/usr/lpp/zcx_zos/properties
JCL and JSON Templates used in workflows	/usr/lpp/zcx_zos/workflows/templates
REXX and Shell scripts used in workflows	/usr/lpp/zcx_zos/workflows/utilities
z/OSMF zCX workflow actions	/usr/lpp/zcx_zos/workflows/actions
z/OSMF zCX workflow steps	/usr/lpp/zcx_zos/workflows/steps

zCX instance name

Each zCX instance must be given a unique name on the z/OS system. The name identifies the zCX instance when creating an instance-specific directory, building configuration files, allocating ROOT and CONFIG VSAM data sets, and more. It is also used as a unique job name (`ZCX_JOB_NAME`) to start the zCX instance.

Follow the z/OS job name guidelines when creating a unique instance name. The guidelines contain safeguards that prevent new zCX instances from overwriting existing ones during provisioning. If a workflow detects the reuse of an existing instance name, it fails the current workflow step.

The zCX instance name is represented by the z/OSMF workflow variable `ZCX_INSTNAME`.

zCX registry and zCX instance directories

zCX registry

The zCX registry is a z/OS Unix System Services file system directory that stores information on all provisioned zCX instances. When a new zCX instance is provisioned, a unique zCX instance directory will be created inside the zCX registry. The zCX registry path must be defined by a system administrator on a sysplex-aware file system as part of the installation and configuration of z/OS Container Extensions. A system administrator also has the option to mount a sysplex-aware zFS.

The zCX registry path can be up to 30 characters long. The recommended path is `/global/zcx_zos/instances`. You can create the zCX registry using `mkdir` command in z/OS UNIX System Services. For example:

```
mkdir -p /global/zcx_zos/instances
```

All zCX provisioning users must have access to this registry.

The zCX registry is represented by the z/OSMF workflow variable *ZCX_REGISTRY_DIR*.

zCX instance directories

Each new zCX instance generates a zCX instance directory inside the zCX registry. A new zFS is allocated and mounted within each unique zCX instance directory. Each zCX instance directory holds instance-specific configuration information used to define and start the zCX instance, including the generated JSON file needed to start the instance as a task. A zCX instance directory also holds any first failure data capture (FFDC) information collected during runtime. The information in a zCX instance directory is used for backup configuration, reconfiguration, restore configuration, upgrade, rollback, and deprovisioning of a zCX instance. The zFS in a zCX instance directory is unmounted and deleted during deprovisioning.

zCX instance ROOT VSAM data set allocation

The zCX provisioning workflow allocates a linear VSAM data set to hold the zCX instance root image based on values of the variables described in this section.

Table 12. z/OSMF workflow variables for ROOT VSAM data set allocation

Variable	Description	Required?	Default (if applicable)
ZCX_HLQ	This is the high level qualifier used for VSAM data set and zFS allocation. The zCX z/OSMF provisioning workflow will construct the fully-qualified VSAM data set name and zFS filesystem using this specified high level qualifier, low level qualifiers, and the unique instance name. The value can be a maximum of 28 characters.	Yes	

Table 12. z/OSMF workflow variables for ROOT VSAM data set allocation (continued)			
Variable	**Description**	**Required?**	**Default (if applicable)**
ZCX_ROOTMB	This is the allocated size of the ROOT VSAM data set. The minimum size required is 4000 MB.	Yes	4000 MB
ZCX_ROOTVOLSER	This is the volume serial used for VSAM data set allocation. This value may be required if the data set is SMS managed. Check with your storage administrator for more details.	This value may be required if the data set is SMS managed. Check with your storage administrator for more details.	
ZCX_ROOTSTORCLAS	This is the root VSAM data set STORCLAS used for SMS managed ROOT VSAM data set allocation. This is required if allocating a ROOT VSAM data set larger than 4 GB and needs extended addressability and SMS specification.	SMS specification may be required for VSAM data set allocation depending on site installation requirements. Check with your storage administrator to obtain the STORCLAS specification.	
ZCX_ROOTDATACLAS	This is the root VSAM data set DATACLAS used for data set allocation. This is required if allocating a root VSAM data set that is larger than 4 GB and needs extended addressability.	Extended addressability specification may be required for VSAM data set allocation depending on site installation requirements. Check with your storage administrator to obtain the DATACLAS specification.	
ZCX_ROOTMGMTCLAS	This is the root VSAM data set MGMTCLAS used for SMS-managed root VSAM data set allocation.	SMS managed storage specification may be required for VSAM data set allocation depending on site installation requirements.. Check with your storage administrator to obtain the MGMTCLAS specification.	

zCX instance CONF VSAM data set allocation

The zCX provisioning workflow allocates a linear VSAM data set to hold the zCX instance configuration information based on values of the variables described in this section.

Table 13. z/OSMF workflow variables for CONF VSAM data set allocation			
Variable	**Description**	**Required**	**Default (if applicable)**
ZCX_CONFIGVOLSER	This is the configuration VSAM data set volume serial used for CONFIG VSAM data set allocation.	If the system uses SMS, then the volume name/ serial based on installation requirements may be required. Check with the storage administrator for more details.	
ZCX_CONFIGMB	This is the size of the zCX Configuration VSAM data set in megabytes. The specified size will be used to allocate the CONFIG VSAM data set. The minimum value is 2 MB.		2 MB
ZCX_CONFIGDATACLAS	This is the configuration VSAM data set DATACLAS used for CONFIG VSAM data set allocation. Check with the storage administrator to obtain the DATACLAS specification.	This is required when allocating a CONFIG VSAM data set greater than 4 GB that needs extended addressability specification. Depending on site installation requirements, extended addressability specification may be required for all VSAM data set allocation.	
ZCX_CONFIGSTORCLAS	This is the configuration VSAM data set STORCLAS used for SMS-managed CONFIG VSAM data set allocation.	This is required when allocating a ROOT VSAM data set greater than 4 GB that needs extended addressability and SMS specification. Depending on site installation requirements, SMS may be required for VSAM data set allocation. Check with the storage administrator and obtain the STORCLAS specification.	

Table 13. z/OSMF workflow variables for CONF VSAM data set allocation (continued)			
Variable	**Description**	**Required**	**Default (if applicable)**
ZCX_CONFIGMGMTCLAS	This is the Config VSAM dataset MGMTCLAS to use for SMS managed storage Config VSAM dataset allocation.	Depending on your installation requirements, SMS managed storage specification may be required for VSAM dataset allocation in your site. Check with your storage administrator and obtain the MGMTCLAS specification.	

zCX instance Z file system (zFS) allocation

A new Z file system (zFS) is allocated and mounted on the newly created instance directory when executing the zCX provisioning workflow. As part of provisioning, the workflow will explode the compressed binaries and build the instance image in the zFS directory before loading the image on the ROOT VSAM data set. An uncompressed image size is around 3500 MB, and the zFS must have enough space to hold the uncompressed image before it is loaded on the ROOT VSAM data set and removed from the file system.

The recommended zFS allocation size is greater of 4095 MB, or 1.5x the zCX instance memory requirement (*ZCX_MEMGB*). In addition to the uncompressed root image, the zFS instance directory also holds the FFDC information for the zCX instance.

IBM recommends updating the BPXPRMxx member mount statements to include the zFS mount on the instance directory to persist after IPL.

Table 14. z/OSMF workflow variables for zFS allocation			
Variable	**Description**	**Required**	**Default (if applicable)**
ZCX_ZFS_FILESYSTEM_HLQ	Specify a high level qualifier to use for zFS file system allocation. The high level qualifier can be up to 28 characters including periods. The zCX workflow constructs the fully qualified zFS file system name using the specified high level qualifier and the unique instance name. The zFS file system is used to hold the zCX appliance instance specific details and FFDC information for the zCX appliance instance.	Yes	

Table 14. z/OSMF workflow variables for zFS allocation (continued)			
Variable	**Description**	**Required**	**Default (if applicable)**
ZCX_ZFS_ENCRYPT	Flag to indicate whether to encrypt the zFS file system that is mounted on the zCX instance directory. The encryption key label must be predefined using either the RACF DS profile for the zFS file system high level qualifier or with the SMS data class construct to use the encrypt option. Once the encryption key label is defined, the zFS file system VSAM linear data set is defined with an encryption key label. The encryption process uses the VSAM encryption support that is provided by DFSMS. When zFS encrypts a file system, it encrypts all security information, access control lists, symbolic link contents, and file contents.	Yes	
ZCX_ZFS_VOLUME	This is the zFS volume name that is used for allocation.	Depends on whether it is SMS managed and site installation requirements. Check with the storage administrator for more details.	
ZCX_ZFS_PRIMARY_MEGABYTES	This is the primary megabytes that are used for zFS allocation. The recommended zFS filesystem allocation size is greater of 4095MB or 1.5X of the zCX Appliance Instance memory specification (`ZCX_MEMGB`).	Yes	4095 MB
ZCX_ZFS_SECONDARY_MEGABYTES	This is the secondary megabytes for the zFS allocation.	Yes	1000 MB

Table 14. z/OSMF workflow variables for zFS allocation (continued)

Variable	Description	Required	Default (if applicable)
ZCX_ZFS_DATACLAS	This is the DATACLAS that is used for zFS allocation.	Yes, if the zFS being allocated is greater than 4 GB and requires extended addressability specification. Extended addressability specification may be required for zFS file system allocation depending on site installation requirements. Check with the storage administrator to obtain the DATACLAS specification.	
ZCX_ZFS_STORCLAS	This is the zFS STORCLAS to use for SMS-managed file system allocation.	Yes, if the zFS file system being allocated is greater than 4 GB and requires extended addressability specification. Extended addressability specification may be required for zFS allocation depending on site installation requirements. Check with the storage administrator to obtain the STORCLAS specification.	
ZCX_ZFS_GROUP_NAME	This is the SAF group that controls access to the zCX appliance instance directory. The started task user must be connected to this group.	Yes	
ZCX_ZFS_MGMTCLAS	This is the zFS file system MGMTCLAS to use for SMS managed storage zFS file system allocation.	SMS managed storage specification may be required for zFS file system allocation in your site. Check with your storage administrator and obtain the MGMTCLAS specification.	

zCX instance diagnostic data storage

The diagnostics data (dlogs) linear VSAM data set will be used to hold diagnostic data, zCX instance logs, and FFDC information. Only one dlogs data linear VSAM data set can be allocated, and it is required for provisioning a zCX instance. The data set must be specified during provisioning and cannot be added, modified, or removed later.

Table 15. z/OSMF workflow variables for diagnostic data storage.

Variable	Description	Required?	Default (if applicable)
ZCX_DLOGSMB	This is the allocated size for each Dlog data VSAM data set diagnostics and logs space use in megabytes. The minimum size required is 1000 MB.		1000 MB
ZCX_DLOGS_COUNT	This is the number of Dlogs data VSAM data sets allocated for IBM zCX appliance instance diagnostics and logs space use. Currently only one VSAM data set is supported for diagnostics and logs space use. The size of Dlogs Data VSAM data set is determined by the value specified in the *ZCX_DLOGSMB* variable field.	Yes	1
ZCX_DLOGSVOLSER	This is the name assigned to the Dlogs data VSAM data sets during allocation.	Depends on site installation requirements and whether the data set uses SMS managed storage. Check with the storage administrator for more detail.	

Table 15. z/OSMF workflow variables for diagnostic data storage. (continued)			
Variable	**Description**	**Required?**	**Default (if applicable)**
ZCX_DLOGSSTORCLAS	This is the Dlogs data VSAM data set STORCLAS for Dlogs data VSAM data set allocation.	Yes, if allocating a Dlogs data VSAM data set that is larger than 4 GB and needs extended addressability and SMS managed storage specification.SMS managed storage specification may be required for VSAM data set allocation depending on site installation requirements. Check with the storage administrator to obtain the STORCLAS specification.	
ZCX_DLOGSDATACLAS	This is the Dlogs data VSAM data set DATACLAS for Dlogs data VSAM data set allocation.	Yes, if allocating a Dlogs data VSAM data set that is larger than 4 GB and needs extended addressability. Extended addressability specification may be required for VSAM data set allocation depending on site installation requirements. Check with the storage administrator to obtain the DATACLAS specification.	
ZCX_DLOGSMGMTCLAS	This is the Dlogs data VSAM data set MGMTCLAS for Dlogs data VSAM data set allocation.	=SMS managed storage specification may be required for VSAM data set allocation depending on site installation requirements. Check with your Storage Administrator to obtain the MGMTCLAS specification.	

zCX instance swap data storage

Swap data linear VSAM data sets are allocated to use as swap space for a zCX instance. Swap data volumes allow zCX to support Docker containers when the virtual memory in use exceeds the real memory available to the zCX instance. The number and size (in megabytes) of each swap VSAM data set must be specified. At least one swap data linear VSAM data set must be allocated for swap space use. Additional swap data sets can be added to a zCX instance after provisioning by using the add data disks workflow. Allocated swap data sets cannot be removed from a zCX instance without deprovisioning.

Table 16. z/OSMF workflow variables for swap data.			
Variable	**Description**	**Required?**	**Default (if applicable)**
ZCX_SWAPMB	This is the swap data VSAM data set allocation size (in megabytes) for each VSAM data set allocation. The minimum size required for swap space use is 1000 MB. The number of swap data VSAM data sets is specified by *ZCX_SWAP_COUNT* The recommended Swap data size is greater than the maximum amount of virtual memory that will be used by the containers deployed in the zCX instance, minus the zCX instance memory specification (*ZCX_MEMGB*). For example, if the zCX instance requires 10 GB of virtual memory and is configured with 4 GB of real memory, then at least 6 GB of swap space is required.	Yes	2000 MB
ZCX_CREATE_SWAP	This indicates whether a swap data volume is created for the zCX instance. When set to TRUE, swap data linear VSAM data sets are allocated to use as swap space.	Yes	

Table 16. z/OSMF workflow variables for swap data. (continued)			
Variable	**Description**	**Required?**	**Default (if applicable)**
ZCX_SWAP_COUNT	Specify the number of swap data VSAM data sets to allocate for the zCX instance. Each swap data VSAM data set is allocated according to the size specified by *ZCX_SWAPMB*. At least one swap data linear VSAM dataset must be allocated. The maximum number of swap data sets that can be specified at a time is 5. You can add up to 245 total swap and data disks per zCX instance.		1
ZCX_SWAPSTORCLAS	This is the swap data VSAM data sets STORCLAS to use for the SMS managed storage swap data allocation.	Yes, if allocating a swap data VSAM data set greater than 4 GB that needs extended addressability and SMS managed storage specification. SMS managed storage specification may be required for VSAM data set allocation depending on site installation requirements. Check with the storage administrator to obtain the STORCLAS specification.	
ZCX_SWAPDATACLAS	This is the swap data VSAM data set DATACLAS for swap data allocation.	Yes, if allocating a swap data VSAM data set greater than 4 GB that needs extended addressability specification. Extended addressability specification may be required for VSAM data set allocation depending on site installation requirements. Check with the storage administrator to obtain the DATACLAS specification.	

Table 16. z/OSMF workflow variables for swap data. (continued)			
Variable	**Description**	**Required?**	**Default (if applicable)**
ZCX_SWAPMGMTCLAS	This is the swap data VSAM data set MGMTCLAS for SMS managed storage swap data allocation.	SMS managed storage specification may be required for VSAM data set allocation depending on site installation requirements. Check with the storage administrator to obtain the MGMTCLAS specification.	

zCX instance user data storage

User data volumes hold all Docker images, containers, and generated files in a zCX instance. This includes Docker volumes used by a container. At least one user data linear VSAM data set must be allocated. You must specify the number and size (in megabytes) of user data VSAM data sets. Additional user data volumes can be added to a zCX instance after provisioning by using the add data disks workflow. User data volumes allocated to a zCX instance cannot be removed without deprovisioning.

Table 17. z/OSMF workflow variables for user data set allocation			
Variable	**Description**	**Required?**	**Default (if applicable)**
ZCX_DATAMB	This is the user data VSAM data set allocation size in megabytes for each VSAM data set allocation. The minimum required size for user data space use is 1000 MB.	Yes	20 GB
ZCX_DATA_COUNT	This is the number of user data VSAM data sets to allocate for a zCX instance. At least one user data linear VSAM dataset must be allocated. The maximum number of user data VSAM data sets that can be specified at a time is 10. You can add up to 245 total swap and data disks per zCX instance.	Yes	1
ZCX_DATAVOLSER	This is the volume name to use for user data VSAM data set allocation.	Maybe, depending on site installation requirements and if you are using SMS managed storage. Check with the storage administrator for more details.	

Table 17. z/OSMF workflow variables for user data set allocation (continued)			
Variable	**Description**	**Required?**	**Default (if applicable)**
ZCX_DATASTORCLAS	This is the STORCLAS to use for SMS managed storage VSAM data set allocation.	Yes, if allocating a VSAM data set greater than 4 GB that needs extended addressability and SMS managed storage specification. SMS managed storage specification may be required depending on site installation requirements. Check with the storage administrator to obtain the STORCLAS specification.	
ZCX_DATADATACLAS	This is the DATACLAS to use for user data VSAM data set allocation. This is required	Yes, if allocating a VSAM data set greater than 4 GB that needs extended addressability specification. Extended addressability specification may be required depending on site installation requirements. Check with the storage administrator to obtain the DATACLAS specification.	
ZCX_DATAMGMTCLAS	This is the MGMTCLAS to use for SMS managed storage user data VSAM data set allocation.	SMS managed storage specification may be required depending on site installation requirements. Check with the storage administrator to obtain the MGMTCLAS specification.	

Network considerations

TCP/IP profile updates

The following z/OS TCP/IP profile updates must be made before provisioning and starting a zCX instance.

- DVIPAs for zCX must be defined using VIPARANGE statements with the ZCX attribute.
  - Multiple DVIPAs can be defined on a single VIPARANGE statement if they belong to the same subnet.
  - The DVIPA specified in the provisioning workflows must have a matching VIPARANGE definition in the TCP/IP profile when the zCX instance is started. Otherwise, zCX initialization will fail.

– Sysplex considerations: If you need the capability of restarting a zCX instance across other systems in the sysplex, then include the same VIPARANGE statements on all systems in the sysplex on which the zCX instance can be started.

IPv4/IPv6 dual-stack

Specifying an IPv6 address for the *ZCX_GUESTIPV6* variable enables dual-stack support. This allows use of IPv4 and IPv6 protocols with any container, service, or network. Docker can communicate with both IPv4 and IPv6 endpoints. These endpoints can include DNS servers, registries, proxies, and LDAP.

The following network information is used to provision a zCX instance:

Table 18. Unique network settings for each zCX instance

Variable	**Description**
ZCX_GUESTIPv4	This is a unique guest IP address. It must be a zCX DVIPA IPv4 address.
ZCX_HOSTNAME	This is a hostname for the zCX appliance instance.
ZCX_GUESTIPV6	This is a unique guest IPv6 address. This variable must be specified with the *ZCX_GUESTIPV4* variable. Specifying an IPv6 address enables IPv4/IPv6 dual-stack support.

Table 19. Additional network settings that may be shared across multiple zCX instances

Variable	**Description**
ZCX_HOSTDNS1 and *ZCX_HOSTDNS2*	These are DNS Server IP Addresses. Up to two IP addresses can be specified for resiliency. These IP addresses can be either IPv4 or IPv6 format. IPv6 format is only allowed if the variable *ZCX_GUESTIPV6* is assigned a value. These IP addresses may be the same as the DNS Server IP addresses configured in the z/OS Resolver (*NSINTERADDR* or *NAMESERVER* statements in TCPIP.DATA).
ZCX_HOSTDNS_SEARCH	This is the DNS Search Domain. Specify the list of domain names that are appended, in the order listed, to the host name to form the fully qualified domain name of a host being queried. This is similar to z/OS Resolver configuration SEARCH statement in TCPIP.DATA.

Table 20. Optional network configuration options that may be shared across multiple zCX instances	
Variable	**Description**
ZCX_MTU	This is the MTU size to be used for IP communications. Check with your network administrator to obtain this information. The default MTU size is 1492, which should be suitable for most environments. A larger MTU can be specified to optimize local communications inside the same z/OS system (to z/OS applications or to other zCX instances). When a larger MTU is specified, then all network peers communicating with the zCX instance must have path MTU discovery enabled across the network.
ZCX_TCPIPNAME	This is the TCP/IP Stack name that is only needed if your z/OS system has multiple TCP/IP stacks configured (CINET configuration). If multiple stacks are configured, this variable indicates the TCP/IP stack to which the zCX instance should connect. If you omit the TCP/IP stack name in a multi-stack configuration, zCX will attempt to connect to each TCP/IP stack that is active until a successful connection is made (the specified DVIPA is successfully found and activated).
ZCX_MTU_IPV6	This is the MTU size for ZCX_GUESTIPV6. This variable is only valid if the variable ZCX_GUESTIPV6 is set. It is to be used for IP communications. Check with your network administrator to obtain this information. The default MTU size is 1492, which should be suitable for most environments. You can specify a larger MTU to optimize local communications inside the same z/OS system (to z/OS applications or to other zCX instances). When a larger MTU is specified, all network peers communicating with the zCX instance must have path MTU discovery enabled across the network.

Parallel workflow step automation

Some workflows are capable of executing steps in parallel, thus reducing the amount of time needed to complete. Both the provisioning workflow (provision.xml) and the add data disks workflow (add_data_disks.xml) are capable of parallel workflow step automation.

To take advantage of parallel workflow step automation, JES2 must execute the jobs submitted by the workflows in parallel. There are two ways to allow JES2 to run workflows in parallel:

1. Configure JES2 with a job class that specifies `DUPL_JOB=NODELAY`, and configure the provisioning user's z/OSMF workflows default JOB statement by navigating to **Actions** then **Customize JOB Statement** in z/OSMF Workflow to reference the job class. This method allows all jobs with the same job name submitted by the workflow to run in parallel.
2. During provisioning or reconfiguration of a zCX instance, set *ZCX_UNIQUE_JOBNAMES* in the zCX General Configuration to TRUE. This directs the workflow to use unique job names for the longer running jobs, allowing JES2 to run those jobs in parallel. When TRUE, the *ZCX_UNIQUE_JOBNAME_PREFIX* property allows the provisioning user to set a 1-3 character prefix for the generated job names.

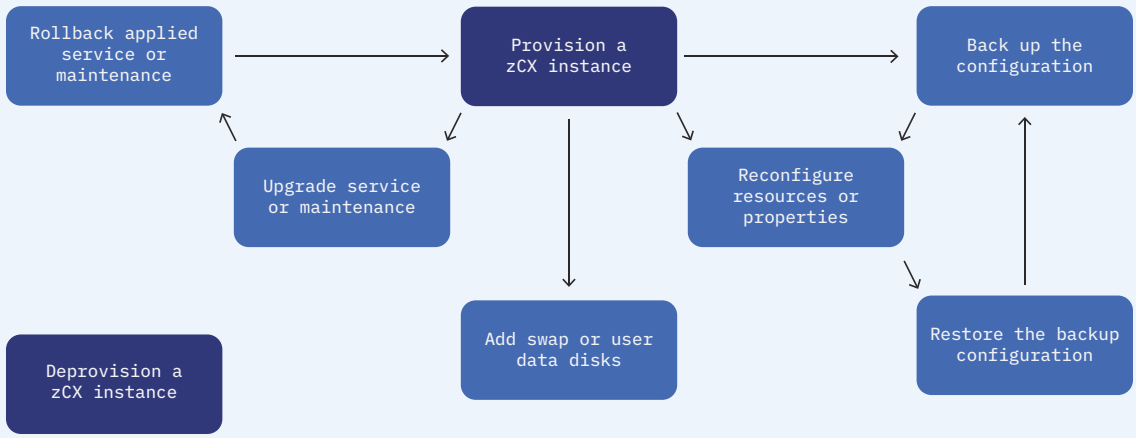
Additionally, the provisioning user must run the workflow in parallel after performing the initial manual step or steps. The provisioning user can run the workflow in parallel by navigating to **Actions** then **Start Parallel Automation** in z/OSMF Workflow.

Chapter 9. Running zCX Workflows

zCX uses z/OSMF workflows provision, deprovision, and otherwise manage zCX instances. The available workflows are:

- Provisioning Workflow
- Backup Configuration Workflow
- Reconfiguration Workflow
- Restore Configuration Workflow
- Add Data Disks Workflow
- Upgrade Workflow
- Rollback Workflow
- Deprovisioning Workflow

Lifecycle management of a zCX instance using workflows



Workflow variable input properties files

The workflow variables input properties file contains all the variables and corresponding values required to provision an IBM zCX instance. A sample input properties file of workflow variables is provided under 'Properties Directory' inside the installation directory.

A z/OS system administrator can copy the sample file in to a read/write directory and modify it with system-specific inputs. The system administrator should consult the z/OS storage administrator, z/OS security administrator, and the z/OS network administrator to provide the proper values.

The workflow variables input properties file is optional, as all the inputs can be added in the z/OSMF provisioning workflow directly. If the variables are manually added to the workflow, they can be saved to a specified directory as a workflow variables input properties file for subsequent use. Descriptions of the variables can be found in this document or by clicking the tool tips icons in z/OSMF.

How to use a zCX workflow

Once logged into z/OSMF, use the following instructions to create and begin any of the workflows. Specific information on each workflow can be found in the respective section of this chapter.

1. Click **Workflows** in the left navigation panel.
2. Click **Actions**, then **Create Workflows**.
3. In the **Create Workflows** window, provide the path of the workflow you wish to run (the default locations of each workflow are in their respective sections of this chapter).
4. Provide the path to the workflow variables input properties file, or manually input the values.
5. Select the system that will provision the zCX instance.
6. Click **Next**.
7. Click **Assign all steps to owner user ID** at the bottom of the **Create Workflow** window to give the owner the authority to execute them.
8. Click **Finish**.The workflow task UI panel will open, and all the steps will be listed in the panel. Start with step 1 then continue through sequentially. To begin, click a step, then click **Actions** and select **Perform**. After step 1, all following steps can be automated by selecting the option to Automatically perform the selected step, and all subsequent automated steps, according to their declared step dependencies, until one of the following occurs:
   - all workflow steps have been completed
   - a non-automated, non-Complete step, is reached, or
   - an error occurs.

Modify the JOB card

When using a zCX workflow, ensure that the SYSOUT class used by your jobs does not purge the job output before z/OSMF can retrieve it. Should this happen, z/OSMF will report that the step failed. Rerunning the step can lead to unpredictable results in that step of the workflow, or in a subsequent step. You can tailor the default JOB card from the **Workflows** menu by performing the following steps:

1. Select **Actions**, then **Customize JOB Statement**.
2. Highlight **Modify the default JOB statement** and select **OK**.
3. Ensure that the MSGCLASS specified uses a SYSOUT class that is not quickly purged. Consult the JES system programmer to determine the proper SYSOUT class.
4. Select **OK** to save the JOB card.

Provisioning workflow

Provisioning workflow path: `/usr/lpp/zcx_zos/workflows/provision.xml`

Running automated steps of provisioning in parallel

The automated steps of the provisioning workflow can be executed in parallel, thus reducing the amount of time it takes for the workflow to complete. Instructions can be found in "Parallel workflow step automation" on page 53. First manually execute the non-automated steps at the beginning of the workflow, and then execute the automated workflow steps by navigating to **Actions** then **Start Parallel Automation**.

Starting a zCX instance

Once a zCX instance is successfully provisioned, you can start the zCX instance by issuing a z/OS START command. This command will be provided during the last step of the provisioning workflow. The zCX instance will then run as a started task.

Backup configuration workflow

Backup configuration workflow path: `/usr/lpp/zcx_zos/workflows/backup_config.xml`

This workflow will backup the existing configuration of a IBM zCX instance. This can be a useful step before using the reconfiguration workflow, as it allows you to save the current working zCX instance. The backup configuration workflow will update the zCX instance configuration disk backed by the configuration VSAM linear data set.

Only a single backup configuration is supported, and every execution of the backup configuration workflow overwrites a previously backed up configuration. The backed up configuration can be reapplied to the zCX instance by running the Restore Configuration workflow.

The backup configuration workflow asks you to provide:

1. The unique name of the zCX instance to back up
2. A description of your backup

Reconfiguration workflow

Reconfiguration workflow path: `/usr/lpp/zcx_zos/workflows/reconfigure.xml`

The reconfiguration workflow will update a zCX instance configuration. The steps in this workflow walk through each of variable that you may reconfigure, allowing you to assign a new value. This can be done against an active zCX appliance, but the changes will not take effect until the appliance is restarted. This workflow cannot alter any properties related to disk management. Disk management is handled by the add data disks workflow.

If you would like to save your previous configurations, you should run the back up configuration workflow. This will allow you to restore your previous configuration using the restore configuration workflow should you encounter any problems after reconfiguration.

The reconfiguration workflow asks you to provide:

1. The unique name of the zCX instance to reconfigure
2. The zCX instance registry path, if not specified in the workflow variables input file.

Restore configuration workflow

Restore configuration workflow path: `/usr/lpp/zcx_zos/workflows/restore_config.xml`

This workflow will restore a previously backed up zCX instance configuration. You can back up a zCX instance by running the back up configuration workflow. Only one backup can exist at any given time, and the restore configuration workflow will display the both description of the backup and the date it was created.

The restore configuration workflow asks you to provide:

1. The unique name of the zCX instance to restore
2. The zCX instance registry path, if not specified in the workflow variables input file.

The following table provides descriptions of the variables needed to restore configuration:

Table 21. Restore configuration workflow variables

Variable	Description
ZCX_BACKUP_INFO	This is a user-specified description that will be associated with the IBM zCX backup configuration using the workflow. This description will be displayed in the restore configuration workflow to communicate the contents of the backup configuration.

Table 21. Restore configuration workflow variables (continued)	
Variable	**Description**
ZCX_BACKUP_DATE	This is the date when the appliance instance configuration was backed up by the backup configuration workflow. This variable will be displayed when using the IBM zCX restore configuration workflow.

Add data disks workflow

Add data disks workflow path: `/usr/lpp/zcx_zos/workflows/add_data_disks.xml`

This workflow allows you to add swap data and user data disks to a zCX instance. The workflow will display current zCX swap data and user data storage configurations, then provide the capability to add more. New VSAM linear data sets will be allocated and formatted to hold the new swap data and user data disks. When this workflow is run against an active zCX instance, the changes do not take effect until the appliance is restarted.

You cannot remove swap data and user data disks once they are added to a zCX instance. You also cannot restore a backup configuration once additional disks are added. A new backup configuration should be created once the zCX instance is restarted with the additional disks.

The add data disks workflow asks you to provide:

1. The unique name of the zCX instance to which you want to add swap data storage and use data storage, if not specified in the workflow variables input file
2. The zCX instance registry path, if not specified in the workflow variables input file
3. The size of additional swap data storage megabytes and the number of additional VSAM data sets to allocate
4. Allocation parameters for swap data VSAM data sets (Volume Serial, *STORCLAS*, *DATACLAS* and *MGMTCLAS*)
5. The size of additional user data storage in megabytes and the number of additional VSAM data sets to allocate
6. Allocation parameters for user data VSAM data sets (Volume Serial, *STORCLAS*, *DATACLAS* and *MGMTCLAS*)

After the above information is provided in steps 1 and 2, the remainder of the workflow is automated.

Upgrade workflow

Upgrade workflow path: `/usr/lpp/zcx_zos/workflows/upgrade.xml`

This workflow applies service and maintenance to an existing zCX instance. The workflow will display the current version of the zCX instance and ask for confirmation of the target version before upgrading.

The upgrade workflow will allocate an alternate ROOT VSAM data set to hold the new service binaries of the zCX instance. It will use the same *VOLUME*, *DATACLAS*, *STORCLAS* and *MGMTCLAS* that were previously used when provisioning the zCX instance. Before upgrading, ensure there is enough storage in the *VOLUME* and SMS managed *STORCLAS* to allocate another ROOT VSAM data set. Currently, the size of this additional data set is 4 GB.

You can perform the upgrade workflow while the target instance is up and running, but you must recycle (start and stop) the instance for the upgrade to take effect. This allows you to schedule service windows for Docker applications running on a zCX instance.

The upgrade workflow will backup one copy of your existing configuration files and ROOT VSAM data set. If there is a failure when executing the upgrade workflow, manually run step 13 ("On error run this step to restore backed up files") to undo the workflow. You can use your installation-specific backup tools to back

up all the VSAM data sets associated with the zCX instance (including the instance directory zFS) if you would like additional copies.

If you want to return to the previous service level after upgrading, you can use the rollback workflow. This will rollback your zCX instance service level to the previously saved version.

The upgrade workflow asks you to provide:

1. The unique name of the zCX instance to upgrade
2. The zCX instance registry path, if not specified in the workflow variables input file
3. The zCX install directory path where the zCX binaries are installed. The default location is `/usr/lpp/zcx_zos`.

When the upgrade workflow asks you to confirm the desired upgrade version of the zCX instance, it will display the:

- Current ROOT binary information
- Target ROOT binary infromation
- Install directory that was used to apply the service to the zCX instance. The default location is `/usr/lpp/zcx_zos`.

Rollback workflow

Rollback workflow path: `/usr/lpp/zcx_zos/workflows/rollback.xml`

This workflow allows you to roll back an upgraded service and maintenance level of an existing zCX instance. It restores the ROOT VSAM data set and configuration files that were saved prior to upgrading.

The rollback workflow will only rollback the service level that was applied to an existing zCX instance, not the user data such as Docker images and containers.

The rollback workflow asks you to provide:

1. The unique name of the zCX instance to rollback service and maintenance level
2. The zCX instance registry path, if not specified in the workflow variables input file.

Deprovisioning workflow

Deprovisioning workflow path: `/usr/lpp/zcx_zos/workflows/deprovision.xml`

The deprovisioning workflow asks you to provide:

1. The unique name of the zCX appliance instance to deprovision
2. The zCX registry directory path
   - This can be found in the zCX z/OSMF provisioning workflow step 1 or in the workflow variable input properties file used for provisioning. If you still cannot locate it, try the recommended path: `/global/zcx_zos/instances`

The zCX deprovisioning workflow uses the instance name and location to retrieve the variable values used for provisioning. It will then use the values to return the resources used by the zCX instance.

Remove resources and security controls

If you are permanently deprovisioning a zCX instance you should consider releasing any reserved resources (such as zCX Dynamic VIPA) and removing any security controls associated with this specific zCX instance (such as unique zCX userIDs and related profiles).

Chapter 10. Managing a zCX started task

This chapter explains how to start a zCX instance, automate startup of a zCX instance, apply Automatic Restart Management (ARM), use MODIFY commands, and stop a zCX instance.

Starting a zCX instance

After successfully provisioning a zCX instance, you can start the zCX started task. Th zCX started task and can be started using the z/OS console START command. You can use either of the following 2 options:

- Cut and paste the START command text from the last step in the z/OSMF provisioning workflow. The step is labeled as "Use the provided start command to bring up the zCX appliance instance on z/OS". **Select** and **Perform** the workflow step and to see a custom START command for your zCX instance. For example:

  ```
  S GLZ,JOBNAME=ZCXNAME,CONF='/pathname/ZCXNAME/start.json'
  ```

  In this example, the zCX job name is *ZCXNAME* and */pathname* reflects the zCX instance directory path name in your environment.

- Create a custom started task procedure for a zCX instance. To do this, use the GLZ proc available in SYS1.PROCLIB as a sample, create a new member using the Started task name for this zCX instance, and update the JCL to point to your zCX appliance configuration file:

  ```
  //GLZ      PROC CONF='/pathname/ZCXNAME/
  start.json'
  //* START OF SPECIFICATIONS ****************************************
  //
  *
                               *
  //* MEMBER NAME:
  GLZ                                                                    *
  //
  *
                               *
  //*  DESCRIPTIVE
  NAME:                                                                      *
  //*    Procedure used to start z/OS Container Extensions instance             *
  ```

  You can then issue a START console command without any additional parameters:

  ```
  S ZCXNAME
  ```

Automating start up of a zCX appliance during system IPL

You can automate the start up of your zCX instances during the z/OS IPL process using your automation processes and tools. You will need to ensure that zCX is started after the z/OS TCP/IP stack has been fully initialized. The z/OS Console START command can be used to start zCX. zCX initialization is complete when the following message is issued:

```
GLZM004I zCX Docker services for instance ZCXNAME are available.
```

You can also automate restart of a failed zCX instance using your automation processes and tools, or by exploiting z/OS Automatic Restart Management (ARM).

Automatic Restart Management (ARM)

Automatic Restart Management (ARM) can be used to automatically restart a failed zCX instance. A zCX instance registers with ARM during initialization as an ARM element named *SYSGLZ_jobname*.

TERMTYPE=ELEMTERM is specified when registering a zCX instance with ARM. ARM policy defaults will restart this element if the zCX instance terminates, but not if the system terminates. An ARM policy must be activated to override the default for this element with TERMTYPE(ALLTERM) to also allow ARM to restart when the system terminates. The following is an example of ARM policy JCL that will enable ARM to restart zCX instances for element and system termination:

```
//IXCMIAPU EXEC PGM=IXCMIAPU,REGION=2M
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DATA TYPE(ARM)
DEFINE POLICY NAME(ARMPOL1) REPLACE(YES)
  RESTART_GROUP(GRPZCX1)
   ELEMENT(SYSGLZ_*)
    TERMTYPE(ALLTERM)
/*
```

Enabling ARM to restart a zCX instance when the system terminates enables ARM cross-system restart for the element. Eligible target systems must be properly configured to start the zCX instance. This includes TCP/IP setup, security setup, access to applicable VSAM data sets, zFS, and PROCLIB.

Stopping a zCX instance:

Stopping a zCX instance:

- Use the STOP command to stop the zCX instance without restarting it. When the zCX instance accepts the STOP command, it will de-register from ARM.
- Use the FORCE command with ARM,ARMRESTART to cancel the zCX instance but allow ARM to restart it.

No *ELEMTYPE* is specified during registration. A *RESTART_ORDER* should be specified in the policy if the default of SYSLVL2 is not appropriate for the zCX instances. For example, no change to the *ELEMTYPE* is required for zCX instances that require only DB2, TCP/IP, IMS, or VTAM, as they are at level 1 by default. However, zCX instances that require CICS (level 2 by default), would need to be moved to a level above the CICS instance required, assuming there is not other automation to orchestrate an ARM restart.
The following is an example of ARM policy JCL that will disable ARM:

```
//IXCMIAPU EXEC PGM=IXCMIAPU,REGION=2M
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DATA TYPE(ARM)
DEFINE POLICY NAME(ARMPOL2) REPLACE(YES)
  RESTART_GROUP(GRPZCX1)
   ELEMENT(SYSGLZ_*)
    RESTART_ATTEMPTS(0)
/*
```

If your are using ARM to orchestrate restarts across systems, then you may need to provide a ELEMENT_TYPE(element_type) or LEVEL(level) to the previous example. This ensures that the zCX instance is restarted in the appropriate cross system restart order in relation to other ARM elements.

ARM couple data sets

Couple data sets required for ARM must be defined. They must also be online and in use before starting a zCX instance for which you want ARM support.

- zCX ARM registration fails if ARM is not in use during zCX initialization. When zCX fails for this reason, initialization continues under the assumption that you do not want ARM support.
- If ARM loses access to the couple data sets, zCX registration is lost. In this event, ARM cannot restart a failed zCX instance.

For more information about automatic restart management parameters, see z/OS MVS Setting Up a Sysplex.

MODIFY commands for zCX

zCX accepts the following MVS™ MODIFY commands, in which *job_name* refers to the zCX instance name that receives the command:

MODIFY *job_name*,DISPLAY,CONFIG
Display zCX configuration information.

Output:

```
GLZC003I Configuration information for zCX instance ZCXPROD
File Path: /oc4z/shared/zcx_instances/ZCXPROD/start.json
FFDC Path: /oc4z/shared/zcx_instances/ZCXPROD/FFDC
Dump Path: /oc4z/shared/zcx_instances/ZCXPROD/FFDC/zcx-guest.dmp
Memory size:           8GB
Number of CPUs:        4
Number of Disks:       5
Number of Networks:    1
CTRACE Parmlib Member: CTIGLZ00
Memory Pages:          2
Memory Page Size:      2G
```

MODIFY *job_name*,DISPLAY,DISKVER
Display VSAM data sets in use by zCX and version information.

Output:

```
GLZC008I Disk Version information for zCX instance ZCXPROD
DevNo  Data Set Name                                Version
   1   OMVSSPA.ZCX.ZCXPROD.ROOT                     20190730T123523Z
         3.5.1           1.7.1           HZDC7C0         oa58015
   2   OMVSSPA.ZCX.ZCXPROD.CONF                     20190809T000147Z
   3   OMVSSPA.ZCX.ZCXPROD.SWAP1                    20190809T000129Z
   4   OMVSSPA.ZCX.ZCXPROD.DATA1                    20190809T000132Z
   5   OMVSSPA.ZCX.ZCXPROD.DLOG1                    20190809T000136Z
Total number of disks: 5
```

MODIFY *job_name*,DISPLAY,DISK
Display VSAM data sets in use by zCX.

Output:

```
GLZC004I Disk information for zCX instance ZCXPROD
DevNo     Size  Encrypted? Data set Name
   1       4GB     No      OMVSSPA.ZCX.ZCXPROD.ROOT
   2       3MB     No      OMVSSPA.ZCX.ZCXPROD.CONF
   3    1001MB     No      OMVSSPA.ZCX.ZCXPROD.SWAP1
   4       4GB     No      OMVSSPA.ZCX.ZCXPROD.DATA1
   5    1001MB     No      OMVSSPA.ZCX.ZCXPROD.DLOG1
Total number of disks: 5
```

MODIFY *job_name*,DISPLAY,NET
Display networks in use by zCX.

Output:

```
GLZC005I Network information for zCX instance ZCXPROD
DevNo  Stack     MTU    IP Address
   0   TCP342    1492   9.12.41.90
   1   TCP343    1492   2001::9:114:114:31
Total number of networks: 2
```

MODIFY *job_name*,DISPLAY,VERSION
Display the version information for each zCX subcomponent.

Output:

```
GLZB022I zCX instance CVER8VD9 version information
Bootloader:           HZDC7C0             oa58601
                      3.7.1               1.0
Current Appliance:    HZDC7C0             oa58598
                      3.7.1               1.9.9
```

```
                       20191202T140618Z
Available Appliance:   HZDC7C0             oa59509
                       3.7.1               1.11.4
                       20200428T182310Z
Virtualization Layer:  HBB77C0   OA58996   06/05/20
                       Started on 2020/06/12 11:45:32
Workflows Performed:
Provision:             1.0.29    OA59682   2020/06/12 10:59
Reconfigure:           1.0.18    OA59682   2020/06/12 11:35
Upgrade:               1.0.11    OA59682   2020/06/12 11:38
Add Data Disks:        1.0.16    OA59682   2020/06/12 11:45
```

MODIFY *job_name*,DUMP,GUEST

Dump zCX Linux guest information to the specified FFDC dump file.

Output:

```
GLZC010I A Dump Guest command for zCX instance ZCXPROD has been accepted.
```

Stopping a zCX instance

You can stop a zCX instance using the z/OS console STOP command:

```
STOP ZCXNAME
```

Chapter 11. Logging on to the Docker CLI and using zCX

When a zCX instance is provisioned, it automatically starts a container that provides SSH access for zCX users to use the Docker command line interface (CLI). You can see the following messages in the joblog of the zCX started task:

```
Please connect to IBM z/OS Container Extensions Docker CLI via your SSH client using port
8022
The server is listening on: 9.12.41.127
```

The SSH container is ready for use when you see the following message in the job log:

```
GLZM004I zCX Docker services for instance ZCXNAME are available
```

(where *ZCXNAME* is the name of zCX started task).

Who should have access to the zCX Docker CLI SSH container?

Through the Docker CLI, users can deploy and manage any containers in the zCX instance. This includes the ability to deploy new containers, start/stop containers, inspect container logs, delete containers, etc. In other words, an authorized user of the zCX Docker CLI acts as a zCX administrator. They have control of all containers deployed in a zCX instance. Therefore, access to the Docker CLI should be carefully granted and maintained. The **User management** chapter provides information and instruction on defining access to the zCX Docker CLI SSH container through local or LDAP user management.

Logging on to the zCX Docker CLI SSH container

You can access the zCX Docker CLI environment using an SSH client from your workstation/laptop or optionally, from your z/OS UNIX system services shell environment.

SSH from the client to the server

Run the following command from a workstation with an SSH client, assuming that the private key file is called *id_rsa* and resides in directory $KEY_PATH:

```
ssh -i $KEY_PATH/id_rsa -p 8022 admin@zcx_hostname
```

This was tested with OpenSSH_7.5p1 on Ubuntu Linux and OpenSSH_7.9p1 on Mac.

SSH using PuTTY

If you are using the PuTTY tool to SSH to a remote host, follow these instructions (for Release 0.70 on 64-bit Windows):

1. Convert your private key to the ppk format for use in PuTTY:
   - Start PuTTYgen and in the Conversions menu, click **Import key**.
   - Browse to your SSH private key, select it, then click **Open**.
   - Enter the pass phrase if there is one.
   - From the File menu, click **Save private key** to save the key in the ppk format.
2. Configure an ssh session in PuTTY:
   - In the PuTTY configuration window, type in your zCX IP address, ssh port, and a name for your PuTTY session.

- Click **Connection** then **Data** on the left navigation panel and set the **Auto-login username** to the administrator username.
- Click **Connection**, then **SSH** then **Auth** in the left navigation panel and browse to your generated private key file for authentication.
- Click **Session** then **Save** in the top left navigation panel. You are now ready to SSH into zCX. Click **Open** and proceed.

Using the zCX Docker CLI SSH container

The standard Docker CLI commands are available in this environment. To see the Docker commands available, issue the `docker --help` command. This includes the ability to connect to the Docker daemon running on zCX. Documentation for specific commands can also be obtained using `docker command --help`. The versions of the Docker client and server components can be determined by issuing the following command: `docker version`. This may be helpful if consulting Docker documentation to ensure consistency with the level of your zCX.

Once you login to the zCX Docker CLI SSH container you will have access to a home directory under `/home/`*`userid`* where *userid* is the user you are logged in as. You can use this directory to save any files required. These directories and their contents will be preserved across zCX code upgrades and reconfigurations.

The default time zone of the zCX Docker CLI SSH container is set to Etc/UTC. Each user may set a different time zone by setting the TZ environment variable: *export TZ=<time-zone-name>*. Available time zone names can be displayed in the CLI using:

```
ls /usr/share/zoneinfo
```

For example,

```
export TZ=America/New_York
```

The default time zone can be changed by editing the `.profile` for the user:

```
vi ~/.profile
```

Insert a new line at the bottom containing the above export command:

```
:wq
```

Modifying host ports to a container

As of PTF OA59943, host ports 80 and 443 on the IP address of a zCX instance can be used by a single container.

The `docker run` command `-p` option must include the specific IP address on which the port is used. For example

```
-p 1.1.1.1:80:80
```

maps port 80 on IP address 1.1.1.1 to port 80 within the container.

For a zCX instance below the PTF OA59943 level, host ports 80 and 443 on the IP address of a zCX instance are not available for use by any container.

Restrictions in the zCX Docker CLI SSH container

zCX provides a virtual Docker appliance that includes a pre-configured Linux kernel maintained and supported by IBM. Modifications to the underlying Linux file system and configuration are only supported using the IBM provided z/OSMF workflows. Any modifications to the underlying Linux host or root file system can damage your zCX appliance, result in loss of data, or void your service warranty.

Most Docker CLI commands are available to zCX users. These commands allow you to deploy well-behaved Docker applications within zCX. There are however some Docker CLI command options that have been restricted as they would allow containers unauthorized access to the underlying Linux host. The table below lists these restrictions:

Table 22. Restrictions in the zCX Docker CLI container

Restrictions	Docker Commands	Comments
Privileged containers or containers that require direct access to the resources of the Linux host	`Docker run` with `--privileged` or `--network=host`	Privileged containers are considered insecure and not supported in zCX.
Users are prevented from changing user namespace to host.	A command such as the following is blocked: `docker run --userns=host hello-world`	Direct access to the resources of the Linux host is insecure and not supported in zCX.

Users are prevented from running commands in the zCX Docker CLI SSH container as root	`docker exec [OPTIONS] --user root SSH_CONTAINER COMMAND [ARG...]`
	`docker exec [OPTIONS] --user 0 SSH_CONTAINER COMMAND [ARG...]`
	`docker container exec [OPTIONS] --user root SSH_CONTAINER COMMAND [ARG...]`
	`docker container exec [OPTIONS] --user 0 SSH_CONTAINER COMMAND [ARG...]`
		Running commands as root in the SSH container will allow access to the resources of the Linux host. It will not be allowed.

Users are blocked from pausing the zCX Docker CLI SSH container	`docker pause SSH_CONTAINER` `docker container pause SSH_CONTAINER`	Once the SSH container is paused, it cannot be unpaused, and all users will be blocked out of zCX access.
Users are blocked from updating the zCX Docker CLI SSH container	`docker update  [OPTIONS] SSH_CONTAINER` `docker container update [OPTIONS] SSH_CONTAINER`	Updating the properties of SSH container is not allowed, as it may have unpredictable effects on zCX access.
Users are prevented from renaming the zCX Docker CLI SSH container.	`docker rename SSH_CONTAINER NEW_NAME` `docker container rename SSH_CONTAINER NEW_NAME`	Renaming the SSH container is not allowed because it may adversely impact security features in place.
Users are prevented from removing the zCX Docker CLI SSH container by force	The following command is blocked: `docker rm ibm_zcx_zos_cli -f`	Removing the SSH container of a zCX appliance would prevent users from having access to that appliance.

Table 22. Restrictions in the zCX Docker CLI container (continued)		
Restrictions	**Docker Commands**	**Comments**
Users are blocked from tagging the zCX Docker CLI SSH image zCX ships.	`docker tag SSHD_IMAGE NEW_TAG`	Tagging the SSH image is not allowed because it may adversely impact security features in place.
Users are blocked from mounting Docker volumes owned by the zCX Docker CLI SSH container	Attempts to bind mount *AZD_HOME_VOLUME* and *AZD_MGMT_VOLUME* are blocked	Containers specific for SSH container use should not be accessible to other containers.
Users are blocked from deleting Docker volumes owned by the zCX Docker CLI SSH container	A command such as the following is blocked: `docker volume rm AZD_HOME_VOLUME`	SSH-owned volumes should not be deleted.
Containers or images with the name `ibm_zcx_zos_*` are restricted to IBM internal use.		

Modifying the zCX Docker CLI SSH container

The zCX Docker CLI SSH container is a special Docker container that is provided and serviced by IBM. It serves the important role of allowing authorized zCX users to login into zCX and perform Docker related container management. As a result, it must always be available and must not be modified or extended in an unauthorized manner as that can lead to unpredictable results and in a zCX appliance that can no longer be used properly or serviced by IBM. The following are guidelines on the type of activities that are not supported:

- You cannot install additional software into zCX Docker CLI SSH container. Any such updates will be overwritten when zCX maintenance is applied. If there is requirements for additional commands or functions for this container you can submit requirements to IBM using the RFE process.
- You should not stop or remove the zCX Docker CLI SSH container.
- You cannot `docker exec` into this container as root and make modifications.
- You cannot pause the zCX Docker CLI SSH container.
- You cannot tag the zCX Docker CLI SSH image.
- You cannot remove Docker volumes owned by the zCX Docker CLI container.
- Containers other than the zCX Docker CLI container cannot bind mount the volumes owned by the zCX Docker CLI container.

Bind mounts and Docker volumes

Docker volumes is the recommended method for persisting Docker container data beyond the life of a container. Bind mounts are another method, but are not recommended for most situations. Bind mounts can have dependencies on the underlying host file system directory, which may change over time. Bind mounts can also expose the underlying host to updates that can damage or compromise it.

Docker bind mounts should only be used by special Docker containers that have a legitimate requirement for accessing the underlying Linux host services or data. For example, containers such as Grafana, Prometheus, NodeExporter and cAdvisor can monitor resource usage within zCX. These containers do a have a requirement to access selected data sources within the Linux host via bind mounts.

zCX allows read-only bind mounts of the following directories of the underlying Linux host:

- `/proc` and subdirectories
- `/sys` and subdirectories

- /dev and subdirectories
- /media
- /media/data
- /media/data/docker
- /var/lib/docker

zCX allows read-only bind mounts of the following files of the underlying Linux host:

- /var/run/docker.sock
- /etc/hostname

Preconfigured and mounted Docker volumes

IBM zCX provides the following Docker volumes already preconfigured and mounted to the SSH CLI Container:

Table 23. IBM-provided preconfigured and mounted Docker volumes

Docker volume	**Use**
AZD_HOME_VOLUME	To persist user data information that resides in the home directory of the running zCX instance
AZD_MGMT_VOLUME	To persist configuration information for the running zCX instance
AZD_SHARED_VOLUME	To store and share user data between Docker containers and access the data through the SSH CLI Container

AZD_HOME_VOLUME and AZD_MGMT_VOLUME are for IBM zCX use only. You should not mount, modify, or delete either of these Docker volumes. They cannot be mounted or removed by zCX Docker users.

AZD_SHARED_VOLUME is for use as a regular Docker volume that can be shared among any Docker containers running in the zCX instance. AZD_SHARED_VOLUME is mounted at /media/azd_shared_volume in the SSH CLI Container. You should be able to access this directory and store or retrieve data from the SSH CLI Container. Additionally, you should be able to mount the AZD_SHARED_VOLUME to any other Docker containers started in the zCX instance to store or retrieve data. For example:

```
docker run -it -v AZD_SHARED_VOLUME:/container/directory bash
```

All of these docker volumes will appear as Docker volumes in the running zCX instance. You can view them using the `docker volume ls` command.

Chapter 12. Diagnostics for zCX

This chapter introduces the tools and techniques that will be useful when errors or unusual execution situations arise with a zCX instance.

There are z/OS, zCX, and Linux tools that can be used for diagnosing errors. Additionally, both environments can produce First Failure Data Capture (FFDC) data, including z/OS SVC dumps produced by the zCX instance and FFDC files produced by the Linux environment.

Obtaining diagnostic information for a zCX instance

zCX has many layers of code running, each with its own versions. It is helpful to know what versions of code are running.

Display version information about the virtualization level

Check the zCX instance job log for the GLZB001I message. The code level that you see in the GLZB001I message is for the virtualization layer.

Display version information about the workflow level

When you create a workflow in z/OSMF, the workflow table will show the version.

Display version information about the appliance level

Issue the following operator command to get the appliance version:

F *job_name*,DISPLAY,DISKVER
Display VSAM data sets in use by zCX and version information.

Output:

```
GLZC008I Disk Version information for zCX instance ZCXPROD
DevNo  Data Set Name                                  Version
   1   OMVSSPA.ZCX.ZCXPROD.ROOT                      20190730T123523Z
         3.3.3           1.3.0          HZDC7C0         v1.3.0
   2   OMVSSPA.ZCX.ZCXPROD.CONF                      20190809T000147Z
   3   OMVSSPA.ZCX.ZCXPROD.SWAP1                     20190809T000129Z
   4   OMVSSPA.ZCX.ZCXPROD.DATA1                     20190809T000132Z
   5   OMVSSPA.ZCX.ZCXPROD.DLOG1                     20190809T000136Z
Total number of disks: 5
```

The following commands can also be used to obtain information about the appliance:

F *job_name*,DISPLAY,CONFIG
Display zCX configuration information.

Output:

```
GLZC003I Configuration information for zCX instance ZCXPROD
File Path: /oc4z/shared/zcx_instances/ZCXPROD/start.json
FFDC Path: /oc4z/shared/zcx_instances/ZCXPROD/FFDC
Dump Path: /oc4z/shared/zcx_instances/ZCXPROD/FFDC/zcx-guest.dmp
Memory size:           8GB
Number of CPUs:        4
Number of Disks:       5
Number of Networks:    1
CTRACE Parmlib Member: CTIGLZ00
```

F *job_name*,DISPLAY,DISK
Display disk information.

Output:

```
GLZC004I Disk information for zCX instance ZCXPROD
DevNo     Size  Encrypted? Data set Name
   1       4GB     No      OMVSSPA.ZCX.ZCXPROD.ROOT
   2       3MB     No      OMVSSPA.ZCX.ZCXPROD.CONF
   3    1001MB     No      OMVSSPA.ZCX.ZCXPROD.SWAP1
   4       4GB     No      OMVSSPA.ZCX.ZCXPROD.DATA1
   5    1001MB     No      OMVSSPA.ZCX.ZCXPROD.DLOG1
Total number of disks: 5
```

F *job_name*,DISPLAY,NET
Display networks in use by zCX.

Output:

```
GLZC005I Network information for zCX instance ZCXPROD
DevNo  Stack     MTU    IP Address
   0   TCP342    1492   9.12.41.90
Total number of networks: 1
```

Display zCX instance network information

There may be an issue with the network associated with a zCX instance. Using the F job_name,DISPLAY,NET command from above with the following network command should provide sufficient information for diagnosing a network issue.

F *job_name*,DISPLAY,NET
Displays zCX instance network information

```
F ZCXINST1,DISPLAY,NET
GLZC005I Network information for zCX instance ZCXINST1
DevNo  Stack     MTU    IP Address
  0   TCPIP     1492   9.114.33.171
Total number of networks: 1
```

D TCPIP,,N,VIPADCFG
Displays VIPA configuration

```
EZD0101I NETSTAT CS V2R4 TCPIP 410
DYNAMIC VIPA INFORMATION:
...
VIPA RANGE:
  IPADDR/PREFIXLEN: 9.114.33.171/32
    MOVEABLE: NONDISR                   FLG: C
...
```

D TCPIP,,N,VIPADYN
Displays Dynamic VIPAs

```
EZD0101I NETSTAT CS V2R4 TCPIP 133
DYNAMIC VIPA:
  IPADDR/PREFIXLEN: 9.114.33.171/32
  STATUS:  ACTIVE     ORIGIN: VIPARANGE IOCTL  ZCX:      YES
  ACTTIME: 09/19/2019 17:37:42                 JOBNAME: ZCXINST1
```

D TCPIP,,N,DEVL,INTFN=EZAZCX
Displays EZAZCX interface

```
EZD0101I NETSTAT CS V2R4 TCPIP 864
INTFNAME: EZAZCX            INTFTYPE: ZCX        INTFSTATUS: READY
   ACTMTU: 65535
   SECCLASS: 255                    MONSYSPLEX: NO
 MULTICAST SPECIFIC:
   MULTICAST CAPABILITY: NO
 INTERFACE STATISTICS:
   BYTESIN                          = 24912321
   INBOUND PACKETS                  = 419107
   INBOUND PACKETS IN ERROR         = 0
   INBOUND PACKETS DISCARDED        = 0
   INBOUND PACKETS WITH NO PROTOCOL  = 0
   BYTESOUT                         = 2062357038
   OUTBOUND PACKETS                 = 1643572
```

```
   OUTBOUND PACKETS IN ERROR         = 0
   OUTBOUND PACKETS DISCARDED        = 0
1 OF 1 RECORDS DISPLAYED
END OF THE REPORT
```

Helpful Docker commands

`docker info`

`docker version`

```
Client:
Version:           18.09.7
API version:       1.39
Go version:        go1.10.1
Git commit:        2d0083d
Built:             Wed Jul  3 12:13:47 2019
OS/Arch:           linux/s390x
Experimental:      falseServer:
Engine:
 Version:          18.09.7
 API version:      1.39 (minimum version 1.12)
 Go version:       go1.10.1
 Git commit:       2d0083d
 Built:            Mon Jul  1 19:31:12 2019
 OS/Arch:          linux/s390x
 Experimental:     false
```

`docker system df`

```
TYPE                TOTAL               ACTIVE              SIZE                RECLAIMABLE
Images              2                   1                   488MB               124.7MB (25%)
Containers          1                   1                   301.6kB             0B (0%)
Local Volumes       2                   2                   6.912kB             0B (0%)
Build Cache         0                   0                   0B                  0B
```

`df -h`
Displays file system usage. The 'overlay' and '/dev/mapper/data' lines in the output refer to usage space.

```
Filesystem                      Size  Used Avail Use% Mounted on
overlay                         2.0G  515M  1.3G  29% /
tmpfs                            64M     0   64M   0% /dev
tmpfs                           1.9G     0  1.9G   0% /sys/fs/cgroup
/dev/mapper/data_VG_24710-data  2.0G  515M  1.3G  29% /home
shm                              64M     0   64M   0% /dev/shm
tmpfs                           1.9G  372K  1.9G   1% /run/docker.sock
devtmpfs                        1.8G     0  1.8G   0% /dev/tty
tmpfs                           1.9G     0  1.9G   0% /proc/scsi
tmpfs                           1.9G     0  1.9G   0% /sys/firmware
```

`free -h`

```
              total        used        free      shared  buff/cache   available
Mem:           9.5G        690M        8.3G        8.9M        612M        8.7G
Swap:          5.8G          0B        5.8G
```

The following command may help determine the status of containers running in a zCX instance:

```
docker logs containter_ID
```

You can use

```
docker ps -a
```

to obtain the *container_ID*.

Linux resource monitoring

A monitoring process, `azd_resmon`, facilitates zCX instance tuning by checking the following metrics once per minute:

- Memory in use (MEM)
- Swap space in use (SWAP)
- ROOT disk space in use (ROOT)
- DATA disk space in use (DATA)

An entry and its level is written to the zCX job log about the MEM, SWAP, ROOT, and DATA metrics when the following occur:

- Any metric is greater than or equal to 50% (`notice` log)
- Any metric is greater than or equal to 70% (`warn` log)
- Any metric is greater than or equal to 85% (`crit` log)
- Any metric changes after a `notice`, `warn`, or `crit` entry is written.
- All metrics reduce to below 50% after a previous `notice`, `warn`, or `crit` entry is written (`info` log)

Each entry has the following format:

```
mmm dd hh:mm:ss azd_resmon: loglvl  MEM: nn SWAP: nn ROOT: nn DATA: nn
```

Where:

- *mmm* is the month
- *dd* is the day of the month
- *hh* is the hour of the day in 24-hour notation
- *mm* is the minutes of the hour
- *s* is the seconds of the minute
- *loglvl* is the log level (either `info`, `notice`, `warn`, or `crit`)
- *nn* is the percentage that is being used of a metric

For example:

```
Jun 24 22:42:01 azd_resmon: crit    MEM: 92 SWAP:  1 ROOT: 19 DATA: 25
Jun 24 22:44:01 azd_resmon: crit    MEM: 92 SWAP:  2 ROOT: 19 DATA: 25
Jun 24 22:46:01 azd_resmon: info    MEM: 43 SWAP:  1 ROOT: 19 DATA: 25
```

Here, the first log entry shows that the allocated memory (MEM) is at 92%. The second entry shows that the allocated swap space (SWAP) increased from 1% to 2%. The third entry shows that the all metrics have dropped below 50%.

Necessary documentation for IBM service

Diagnosing problems with zCX requires information from various sources. IBM Service will frequently request this information. This section describes the artifacts that will be requested by IBM Service for any zCX problems.

First, take note of the different pieces that make up zCX:

Table 24. zCX composition

zCX component	FMID	COMPID	FESN
zCX server	HZDC7C0	5752SCCDE	0509651
zCX virtual base	HBB77C0	5752SCCON	0501283
zCX workflows	HZDC7C0	5752SCCWF	0501251

SVC Dump

This dump is typically produced automatically by the system. If the code has ABENDed, this will be created and available for retrieval in the usual place as defined by the installation. zCX uses normal z/OS processing to create and handle this dump.

It may be necessary to create a console dump of the zCX address space. This also follows existing z/OS processes. zCX will automatically collect the appropriate information into the dump when the following is issued to the z/OS system console:

```
DUMP COMM=(Mydump)
R xx, JOBNAME=job_name
```

CTRACE information

The collection of CTRACE information is controlled by a zCX CTRACE parmlib member. More information on CTRACE can be found in the **Reference materials for zCX** chapter of this document.

JES joblog

zCX is a started task that will create a job in JES like any other job on the system. This joblog will be available on the jes spool. Use the procedures associated with your system to capture the joblog for the zCX instance in question.

zCX Server Instance Directory FFDC

Kernel panic errors will automatically reboot the instance and produce FFDC data. Some errors in the appliance will not automatically reboot the instance. When this is the case, FFDC data will be produced the next time the instance is instantiated. Typically, the appliance will fail to reboot the second time in those circumstances.

A GLZM008I message will be produced when the appliance has created FFDC data. An additional message GLZM009I will be produced to indicate where this data has been stored. It will be found in the server instance directory associated with the appliance and will have the file name as indicated in the message.

If IBM Service requests these logs, either FTP them from your z/OS image as binary files, or use the TSO OGET command with the BINARY parameter to export them to a data set.

Example:

```
13.02.34 S0271400  GLZB001I zCX instance ZCXRLK initialization is complete. Code date 07/18/19.
13.02.35 S0271400  GLZM008I zCX instance ZCXRLK is storing failure data for appliance  681
   681             alert:
   681             AZDL0004E At least one prior boot failed
13.02.35 S0271400  GLZM009I zCX instance ZCXRLK stored failure data:  682
   682             Size=36864
   682             Checksum=172244767
   682             File Name=NP8-2019-08-02T17_02_35-AZDL0004E-BL2_error.tar
   682             Directory=/oc4z/shared/zcx_instances/ZCXRLK/FFDC
13.02.43 S0271400  GLZB002I zCX instance ZCXRLK has ended.  683
13.02.45 S0271400  GLZB005I z/OS system service GLZBAIN failed, RC=0000000C RSN=01010016  693
   693             zCX ended due to failure.
13.02.45 S0271400  IEF404I ZCXRLK - ENDED - TIME=13.02.45
13.02.45 S0271400  $HASP395 ZCXRLK   ENDED - RC=0012
```

Example of zCX instance error

This example produced both FFDC information from the appliance and an ABEND from zCX. When reporting this type of problem, send the FFDC file, the dump, and the JES joblog to IBM Service.

```
11.04.15 S0269997  GLZB001I zCX instance ZCXRLK initialization is complete. Code date 07/18/1
11.04.17 S0269997  GLZM008I zCX instance ZCXRLK is storing failure data for appliance  815
   815             alert:
   815             AZDL0001E Unexpected error 2A5A044C, rsn1=00000060 rsn2=00000201
```

```
11.04.17 S0269997  GLZM009I zCX instance ZCXRLK stored failure data:  816
   816             Size=36864
   816             Checksum=750364863
   816             File Name=NP8-2019-08-02T15_04_17-2A5A044C-BL1_error.tar
   816             Directory=/oc4z/shared/zcx_instances/ZCXRLK/FFDC
11.04.25 S0269997  IEA794I SVC DUMP HAS CAPTURED:  823
   823             DUMPID=003 REQUESTED BY JOB (ZCXRLK  )
   823             DUMP TITLE=COMPON=GLZ,COMPID=5752SCCON,ISSUER=GLZMIREC,MODULE=G
   823                        LZBATCB+00000502,ABEND=S0C0D,REASON=8004000F
   823             DUMP CAPTURED USING OPTIMIZE=YES
11.04.25 S0269997  GLZB002I zCX instance ZCXRLK has ended.  824
11.04.26 S0269997  GLZB005I z/OS system service GLZBAIN failed, RC=0000000C RSN=01010015  841
   841             zCX ended due to failure.
11.04.27 S0269997  IEF404I ZCXRLK - ENDED - TIME=11.04.27
11.04.27 S0269997  $HASP395 ZCXRLK   ENDED - RC=0012
```

Chapter 13. Reference information for zCX

This chapter contains the reference information for zCX including:

- MODIFY commands for zCX
- SYSGLZ Component Trace
- GLZ messages
- Reason codes
- COD Abend reason codes

SYSGLZ Component Trace

The following summarizes information for requesting a SYSGLZ component trace for z/OS Container Extensions (zCX).

Table 25. zCX Information for SYSGLZ

Information Type	**SYSGLZ Specifications**
Parmlib member	CTIGLZnn (Default member is CTIGLZ00)
Default tracing	Yes
Trace request OPTIONS parameter	In CTIGLZnn and REPLY for TRACE command
Buffer	• Default: 64 MB • Range: 1 MB - 64 MB • Size set by CTIGLZnn parmlib member or TRACE CT command • Size change after IPL when restarting a trace after stopping it • Location: zCX instance address spaces
Trace records location	Address-space buffer, trace data set for external writer
Request of SVC dump	By DUMP or SLIP command
Trace formatting by IPCS	`CTRACE COMP(SYSGLZ) SUB((ASID(xxxx)))`
Trace format OPTIONS parameter	Yes

Specify options for requesting a SYSGLZ component trace in a CTIGLZxx parmlib member or on the reply for a TRACE CT command. The following table indicates the parameters in a CTIGLZnn parmlib member.

The following table indicates the parameters you can specify on a GCTIGLZnn parmlib member.

Parameter	**Allowed specification on CTIGLZnn**
ON or OFF	Yes
ASID	No
JOBNAME	No
BUFSIZE	Yes (can be changed after IPL)
OPTIONS	Yes

Parameter	Allowed specification on CTIGLZnn
MOD	No
SUB	No
PRESET	No
LIKEHEAD	No
WTR	Yes
WTRSTART or WTRSTOP	Yes

The IBM-supplied CTIGLZ00 parmlib member initializes tracing as soon as a zCX instance address space starts. The contents of CTIGLZ00 are:

```
TRACEOPTS
          ON
          BUFSIZE(64M)
          OPTIONS('DISK','NET','SIE')
```

The following tables indicate the parameters that can be specified on TRACE CT commands and a REPLY.

Table 26. Parameters for TRACE CT for Trace	
Parameter	**Allowed specification on TRACE CT for Trace**
ON, OFF, or nnnnM	One is required
nnnnK or nnnnM	Yes
SUB	Required; specified as SUB=(ASID(xxx)), where xxxx is the 4-digit hexadecimal ASID of the zCX instance the command is targeting
PARM	Yes

Table 27. Parameters for TRACE CT for Write	
Parameter	**Allowed specification on TRACE CT for Write**
WTRSTART or WTRSTOP	One is required if a writer is being used

Table 28. Parameters on REPLY for Trace	
Parameter	**Allowed specification on REPLY for Trace**
ASID	No
JOBNAME	No
OPTIONS	Yes
WTR	Yes

The values for the OPTIONS parameter in the CTIGLZxx parmlib member and reply for a TRACE command are in the following table.

Table 29. Values for the OPTIONS parameter in the CTIGLZxx parmlib member	
Value	**Meaning**
ALL	Trace everything

Table 29. Values for the OPTIONS parameter in the CTIGLZxx parmlib member (continued)	
Value	**Meaning**
`DISK`	Trace Disk events
`NET`	Trace Network events
`SIE`	Trace SIE events
`MIN`	Trace events related to zCX component recovery, abnormal conditions, and other non-mainline paths.

Format the trace with an IPCS CTRACE `COMP(SYSGLZ) SUB((ASID(xxxx)))` subcommand.

GLZ Messages

GLZB001I **zCX instance *job_name* initialization is complete. Code date `mm/dd/yy`.**

Explanation:
The zCX instance identified by job name *job_name* completed initialization and is about to begin IPL of Linux. The code date represents the zCX build level.

System action:
The zCX instance is initialized and the Linux guest started IPL.

Operator response:
None.

System programmer response:
None.

Routing code:
2,*

Descriptor code:
4, 5

GLZB002I **zCX instance *job_name* has ended.**

Explanation:
The zCX instance identified by *job_name* has begun termination processing.

System action:
None.

Operator response:
None.

System programmer response:
Restart zCX.

Routing code:
2,*

Descriptor code:
4, 5

GLZB003I **A zCX instance already exists with job name *job_name*. zCX instance names must be unique.**

Explanation:
Another zCX instance is running under job name *job_name*. zCX instance names must be unique.

System action:
The zCX instance ends.

Operator response:
Confirm that the correct job with the correct parameters was submitted. Contact the systems programmer as needed.

System programmer response:
Determine why an instance is already running. If a new zCX appliance was intended to be created, then change the zCX job name in the corresponding z/OSMF workflow.

Routing code:
10, *

Descriptor code:
5

GLZB004I **z/OS UNIX service *service* failed, RC=*return_code* RSN=*reason_code***

Explanation:
A z/OS UNIX service failed. *service* is the z/OS UNIX callable service that failed with return code *return_code* and reason code *reason_code*. If an associated path name of the file that is related to the callable service failure exists, it is included at the end of the message.

System action:
The zCX instance ends.

Operator response:
Contact the system programmer.

System programmer response:

See documentation for the corresponding return and reason codes. If necessary, contact IBM support.

Routing code:
10, *

Descriptor code:
12

GLZB005I **z/OS system service *service* failed, RC=*return_code* RSN=*reason_code* *optional_failure_explanation*.**

Explanation

A z/OS system service or an internal zCX function that is called *<srvname>* produced an expected or unexpected failure with the indicated return code *return_code* and reason code *reason_code*. An explanation of the failure is provided in *optional_failure_explanation*. Some failures are expected while others are not.

In the message text, *optional failure explanation* can be one these explanations:

Note: Not all possible explanations are listed here.

Table 30. Possible optional failure explanations

Failure explanation and actions
Unable to obtain LPAR information. **Explanation:** See related GLZ error messages. **System Action:** zCX ended due to failure indicated.
Unable to attach Timer service task. **Explanation:** See related GLZ error messages. **System Action:** zCX ended due to failure indicated.
Unable to create RESMGR for address space. **Explanation:** See related GLZ error messages. **System Action:** zCX ended due to failure indicated.
Could not ENQ for STGTEST. **Explanation:** See related GLZ error messages. **System Action:** zCX ended due to failure indicated.
Not enough 1M fixed frames available. **Explanation:** The system failed to allocate the required guest memory to be backed by 1M fixed pages. The most likely reasons are: • Enough 1M frames are not available in the Large Frame Area (LFAREA). • The job is associated with a WLM Resource Group with a Memory limit (MEMPOOL) and the total number of current and potential guest storage fixed pages would exceed the MEMPOOL's limit. • Not enough 1M pages are available even though the LFAREA maximum was not reached. For more information, see the *<srvname>* return and reason code. **System Action:** Based on selected configuration options, a 4K page size allocation might be attempted. If a 4K page size was not selected, then zCX terminates initialization. **Operator Response:** Contact the system programmer. **System Programmer Response:** See message GLZB024I as it provides the guest storage size that was attempted and what the next frame size (if any) will be attempted. Either: • See the *<srvname>* return and reason codes for more information about the error. • Verify the selected guest storage size, and adjust it if necessary. • Increase the size of the 1M LFAREA. • If appropriate, increase the size of WLM Resource Group group limit that is associated with the job to accommodate the entire guest storage size. • If appropriate, increase the amount of real storage available to the system to ensure that there are enough 1M frames. • To potentially prevent termination, add 4K pages to the list of acceptable page sizes when provisioning the appliance.

Table 30. Possible optional failure explanations (continued)

Failure explanation and actions
Not enough 2G frames available. **Explanation:** The system failed to allocate the guest memory that is required to be backed by 2G fixed pages. The most likely reasons are: • Not enough 2G frames are available in the Large Frame Area (LFAREA). • The job is associated with a WLM Resource Group with a Memory limit (MEMPOOL). The total number of in-use fixed pages plus the guest storage fixed pages would exceed the MEMPOOL's limit. For more information, see the *<srvname>* return and reason code. **System Action:** Based on selected configuration options, a 1M or a 4K page size allocation can be attempted. If neither are selected, then zCX ends initialization. **Operator Response:** Contact the system programmer. **System Programmer Response:** See message GLZB024I as it provides the guest storage size that was attempted and what the next frame size (if any) will be attempted. Either: • Verify the selected guest storage size, and adjust it if necessary. • Increase the size of the 2G LFAREA. • If appropriate, increase the size of WLM Resource Group group limit that is associated with the job to accommodate the entire guest storage size. • If appropriate, increase the amount of real storage available to the system to ensure that there are enough 2G frames. • To potentially prevent termination, add 4K pages to the list of acceptable page sizes when provisioning the appliance.
LFAREA does not have 2G or 1M pages defined. **Explanation:** The system attempted to a allocate the guest memory that is backed by 2G or 1M fixed pages, but a Large Frame Area (LFAREA) is not defined. **System Action:** Based on selected configuration options, a 4K page size might be attempted. If 4K was not selected, then zCX ends initialization. **System Programmer Response:** See message GLZB024I to determine the page size and guest storage size that was attempted. Either: • Define an appropriate LFAREA for 2G or 1M pages based on the possible page sizes that are configured for the appliance. • Correct the appliance configuration if the wrong page size was mistakenly selected.

System action:
The zCX instance ends.

Operator response:
Contact the system programmer.

System programmer response:
If the failing system service begins with the prefix GLZ, refer to Reason codes for zCX termination message in *z/OS Container Extensions* for the failure explanation text. Otherwise, consult the appropriate product documentation for the return and reason codes for the given failing system service. If necessary, contact IBM support.

Routing code:
10, *

Descriptor code:
12

GLZB007I The z/OS JSON parser could not be initialized.

Explanation:
The z/OS JSON parser needed to parse the zCX startup file could not be initialized.

System action:
The zCX instance ends.

Operator response:
Contact IBM support.

System programmer response:
Contact IBM support.

Routing code:
10, *

Descriptor code:
5

GLZB008I **The zCX startup file is missing the required option *option*.**

Explanation:
The zCX startup file is missing the required option *option*.

System action:
The zCX instance ends.

Operator response:
Contact the system programmer.

System programmer response:
Use the z/OSMF zCX workflow to properly construct the appliance instance startup file.

Routing code:
10, *

Descriptor code:
5

GLZB009I **Either the array element count for option *option* could not be extracted, or the element count value is not supported.**

Explanation:
The zCX startup file option *option* is specified as an array. Either the number of entries in the array could not be extracted, or the element count value is not supported.

System action:
The zCX instance ends.

Operator response:
Contact the system programmer.

System programmer response:
Use the z/OSMF zCX workflow to properly construct the instance startup file.

Routing code:
10, *

Descriptor code:
5

GLZB010I **The total number of disk and network entries exceeds the allowable maximum of 255 entries.**

Explanation:
The total number of disk and network entries exceeds the allowable maximum of 255 entries.

System action:
The zCX instance ends.

Operator response:
Contact the system programmer.

System programmer response:
Use the z/OSMF zCX workflow to properly construct the instance startup file.

Routing code:
10, *

Descriptor code:
5

GLZB011I **The zCX startup file could not be parsed because it does not conform to proper JSON syntax. Error description= *error_description* File path=*path_name*.**

Explanation:
The zCX configuration file has a JSON syntax error.

System action:
The zCX instance ends.

Operator response:
Contact the system programmer.

System programmer response:
Use the z/OSMF zCX workflow to properly construct the instance startup file.

Routing code:
10, *

Descriptor code:
5

GLZB012I **An IP address in the zCX instance *job_name* startup file has an incorrect format.**

Explanation:
An IP address in the zCX instance *job_name* startup file has an incorrect format.

System action:
The zCX instance ends.

Operator response:
Contact the system programmer.

System programmer response:
Use the z/OSMF zCX workflow to properly construct the instance startup file.

Routing code:
10, *

Descriptor code:
5

GLZB013I **The IP address in the zCX instance *job_name* startup file failed validation. RC=*return_code* RSN=*reason_code*.**

Explanation:
The IP address in the zCX instance *job_name* startup file failed validation.

System action:
The zCX instance ends.

Operator response:
Contact the system programmer.

System programmer response:
Use the z/OSMF zCX workflow to properly construct the instance startup file.

Routing code:
10, *

Descriptor code:
5

GLZB014I **The zCX instance startup file option *name* encountered an error. *error_reason*.**

Explanation

The zCX instance startup file option *name* encountered the error that is described by *error_reason*. The possible values for *name* and *error_reason* are:

Table 31. Possible values for name and error reason

name	***error_reason***
mtu	Valid values are from 1280 to 65535 inclusive.
ip_address	The length of the text string that specifies the IP address must be between 1-39 characters.
stack_name	The length of the text string that specifies the TCP/IP stack name must be between 1-8 characters.
cpu	It must be a value from 1-64.
mem_gb	It must be a value from 1-1024.
install_path	The length of the text string that specifies the install path must be between 1-255 characters.
dsname	The length of the text string that specifies the dsname must be between 1-44 characters.
purpose	The length of the text string that specifies the disk purpose must be between 1-6 characters.
dump	The length of the text string that specifies the dump location must be between 1-255 characters.
ctrace	The length of the text string that specifies the ctrace parm must be between 1-8 characters.
ffdc_path	The length of the text string that specifies the path name must be between 1-255 characters inclusive.
networks	There must be no more than one network with an IPv4 address and one network with an IPv6 address.
purpose	The value of the text string that specifies the disk purpose must be "root", "config", "data", "dlogs", or "swap".
wfversion	The length of the text string that specifies the workflow version must be between 1-8 characters inclusive.
wfservice	The length of the text string that specifies the workflow service must be between 1-8 characters inclusive.
wfdate	The length of the text string that specifies the workflow date must be between 1-19 characters inclusive.
pageframesize	The value of the text string that specifies the frame size must be "4K", "1M" or "2G."

System action:
The zCX instance ends.

Operator response:
Contact the system programmer.

System programmer response:
Use the z/OSMF zCX workflow to properly construct the instance startup file.

Routing code:
10, *

Descriptor code:
5

GLZB015I **Not licensed for zCX. <optional why license check failed> Diag=*xxxx*. <optional why license not being enforced>**

Explanation

This message is issued when a zCX instance is not licensed to run on the machine where it was started. "Machine type is not supported" only appears if the machine is not a z14 or higher. Diag = *xxxx* where *xxxx* is either a hexadecimal diagnostic code or the text "unavailable" if there is no diagnostic data available. This code is intended for IBM use.

<optional why license check failed> is one of the following:

- Machine type not supported.
- Trial expired.

<optional why license not being enforced> is one of the following:

- A blank line (which implies the license is being enforced)
- The text "Temporary unlicensed use expires *mm/dd/yyyy*."
- Trial Enabled.

Note: Feature Code 0104 (Container Hosting Foundation) is required by IBM z/OS Container Extensions, and can be ordered on the IBM z14® servers from the eConfig fulfillment system. You can contact your sales representatives to obtain the required access to the eConfig system. IBM Container Hosting Foundation for z/OS delivers Monthly License Charge (MLC) pricing to satisfy the requirement for the Z hardware feature code 0104.

System action:
The zCX instance terminates.

Operator response:
None.

System programmer response:
None.

Routing code:
2, 10, *

Descriptor code:
4, 5

GLZB016I **zCX instance *job_name* cannot be started. *reason*.**

Explanation

The zCX instance cannot be started due to the *reason* indicated. The possible values for *reason* are:

1. The name of the startup file must be provided.
2. The name of the startup file must be in the range of 2 - 255 characters (inclusive).

System action:
The request to start fails and zCX does not start.

Operator response:
Correct the `Start` command and retry.

System programmer response:
Ensure that a properly formatted startup file exists.

Routing code:
10

Descriptor code:
5

GLZB017I **The zCX instance *job_name* is rebooting.**

Explanation:
The zCX instance that is associated with *job_name* has received a request to reboot the appliance.

System action:
The system starts to reboot. Initialization processing is restarted.

Operator response:
None.

System programmer response:
None.

Routing code:
2

Descriptor code:
4, 5

GLZB018I **zCX instance *jobname storage check results*.**
Guest 4K frames requested= *number*
SYSEVENT STGTEST reported values:
***number1*: Level 1, no real impact.**
***number2*: Level 2, might affect performance.**

***number3*: Level 3, might substantially affect system performance.**

Explanation

The zCX instance checked the available storage on the system against the requested storage of the appliance to be started.

zCX terminates if starting will have a critical memory impact to the system or the memory pool that the zCX instance is running in. It will continue if the usage will have a minor impact that can cause paging.

Note: Regardless of the page frame sizes selected when provisioning the appliance, the values reported in this message are always in terms of 4K page frames. For appliances using a 1M page frame size, consider that there are 256 4K page frames in a 1M page frame.

In the message text:

jobname
Is the name of the job this appliance was started.

storage check results
Possible values:

- Continues but guest memory size can impact performance.
- Terminates due to guest memory size impacting system availability.

number
Is the number of 4K frames to be used for guest memory.

number1
SYSEVENT STGTEST value 1. Number of 4K frames available before affecting system performance a little, if at all.

number2
SYSEVENT STGTEST value 2. Number of 4K frames available before affecting system performance to some degree. Level where SRM can start paging.

number3
SYSEVENT STGTEST value 3. Number of 4K frames available before affecting system performance to some degree. SRM will most likely initiate paging. zCX terminates if this level would be reached.

System action:
zCX terminates if its memory usage for the guest exceeds the SYSEVENT STGTEST value 3 indicating a substantial impact to the system or memory pool. It continues if below the value but raises a warning through this message if it would be above STGTEST value 1 indicating some effect.

Operator response:
Contact the system programmer.

System programmer response:
Consider either increasing the memory available to run all the concurrent workloads to avoid a system impact, or decrease the size of the storage that is defined for the zCX instance or instances. If the appliance is running in a WLM resource group with a memory limit, that is, a memory pool, consider increasing the memory limit.

Routing code:
10

Descriptor code:
12

GLZB019I **zCX instance *jobname* terminating due to guest memory size impacting system availability. Guest 4K frames requested: *xxxxxxxx* Current number of fixed frames:*yyyyyyyy* Current fixed frame warning level: *zzzzzzzz***

Explanation

zCX terminated because it determined that it would have caused a critical memory impact to the system or memory pool due to its required amount of fixed storage.

Note: Regardless of the page frame sizes selected when provisioning the appliance, the values reported in this message are always in terms of 4K page frames. For appliances using a 1M page frame size, consider that there are 256 4K page frames in a 1M page frame.

In the message text:

jobname
Is the job name of the zCX instance.

xxxxxxxx
Is the number of 4K frames to be used for the guest memory.

yyyyyyyy
The current amount of fixed storage in 4K increments.

zzzzzzzz
The current system warning level in 4K increments. When above this level, the system takes action to prevent new work from starting and swaps out current work.

System action:
The zCX instance is terminated.

Operator response:
Contact the system programmer.

System programmer response:

Consider either increasing the memory available to run all the concurrent workloads to prevent a system impact, or decrease the size of the storage that is defined for the zCX instance or instances. If the appliance is running in a WLM resource group with a memory limit, that is, a memory pool, consider increasing the memory limit.

Routing code:
10

Descriptor code:
12

GLZB020I **The zCX instance *job_name* failed to enable trial mode.**

Explanation:
The zCX instance cannot be started. Check for accompanying messages for additional information.

System action:
The system will not start zCX.

Operator response:
None.

System programmer response:
Determine the cause of the failure, fix the problem and restart the instance.

Routing code:
10

Descriptor code:
5

GLZB021I **The zCX instance *jobname* can run in trial mode for *num_days* more day(s).**

Explanation:
zCX is running in a trial. This zCX can continue to be started in trial mode for the number of days indicated.

System action:
The system will run zCX.

Operator response:
None.

System programmer response:
None.

Routing code:
11

Descriptor code:
6

GLZB022I **zCX instance *jobname* version information**
Bootloader: *bootloader version information*
Current® Appliance: *current appliance version information*
Available Appliance: *available appliance version information*
Virtualization Layer: *virtualization layer version information*
Started on *yyyy/mm/dd hh:mm:ss*
Workflows Performed:
Provision: *provision workflow version information*
Reconfigure: *reconfigure workflow version information*
Upgrade: *upgrade workflow version information*
Add Data Disks: *add data disks workflow version information*

Explanation

zCX is reporting for instance *job_name* the version information for each of its parts. GLZB022I is issued during instance initialization, as well as in response to the MODIFY *job_name*, DISPLAY,VERSION command. This includes:

Bootloader
Version information for the first bootloader that started the appliance.

Current Appliance
Version information for the appliance level with which the current instance is provisioned.

Available Appliance
Version information for the appliance level that is installed in the active installation path. If this does not match the current appliance version, then a newer appliance is available on this system. The upgrade workflow can be used on this appliance to install the newer version.

Virtualization Layer
Version information for the virtualization layer, as well as the date and time that this instance started.

Workflows Performed
Version information for the provision, reconfigure, upgrade, and add data disks workflows that are used on the current appliance. This information represents the most recently used version of each workflow. The version information indicates "N/A" if a workflow has not been used or if version information is not available.

System action:
Processing continues.

Operator response:
None.

System programmer response:
None.

Routing code:

11/2

Descriptor code:
6/5

GLZB023I **zCX instance *< jobname>* Guest storage size *< numgb>* GB. Backed by *<numpages>* *<pagesize>* pages.**

Explanation

The zCX guest storage size of *< numgb>* GB backed by *<numpage>* pages of size *<pagesize>* was allocated successfully. However, when using a 4K page size, the system has not yet fixed the pages. In the message:

<jobname>
The name of the job for the zCX instance that is starting.

<numgb>
The guest storage size, which is the amount of memory in GB used to back the guest.

<numpages>
The number of pages of *<pagesize>* that would be required based on the *< stgsize>* and *<pagesize>*.

<pagesize>
Is one of the following:

- "2G fixed"
- "1M fixed"
- "4K pageable then fixed"

System action:
Processing continues.

Operator response:
None.

System programmer response:
None.

Routing code:
11

Descriptor code:
6

GLZB024I **zCX instance *< jobname>* attempted guest storage size *< stgsize>* GB backed by *<numpages> <pagesize>* pages. zCX guest page size restriction: *<fillin values>***

Explanation

The appliance configuration specifies that only a 2G page size can be used to back the guest storage. However, backing the guest with 2G pages is not supported when z/OS is a z/VM® guest. In the message:

<jobname>
The name of the job for the zCX instance that is starting.

<stgsize>
The guest storage size specified during provisioning.

<numpages>
The number of pages of *<pagesize>* that would be required based on the *< stgsize>* and *<pagesize>*.

<pagesize>
One of the page frame sizes selected during provisioning.

<fillin values> can be one of the following:

Table 32. Possible values for <fillin values>

Value and explanation
2G pages not supported on z/VM. Terminating. **Explanation:** The appliance configuration specifies that only a 2G page size can be used to back the guest storage. However, backing the guest with 2G pages is not supported when z/OS is a z/VM guest. **System Action:** zCX terminates. **Operator Response:** Notify the system programmer. **System Programmer Response:** Reconfigure the appliance page frame size with 1M and/or 4K pages to accommodate a z/VM environment. **Routing code:** 2 **Descriptor code:** 4, 5

Table 32. Possible values for <fillin values> (continued)

Value and explanation
2G pages not supported on z/VM. Trying 1M fixed. **Explanation:** The appliance configuration specifies that 2G as well as 1M page sizes can be used to back the guest storage. 2G pages are always attempted first. However, a 2G page size is not supported when z/OS is a z/VM guest. zCX will attempt a 1M page size. **System Action:** zCX attempts a 1M page size. **Operator Response:** None. **System Programmer Response:** If the appliance is never intended to be used on a native z/OS LPAR, then remove the 2G page frame size from the appliance configuration. Review the zCX documentation to ensure you are choosing the appropriate page frame size that fits your needs. **Routing code:** 2 **Descriptor code:** 4, 5
2G pages not supported on z/VM. Trying 4K pageable then fixed. **Explanation:** The appliance configuration specifies that a 2G or 4K page size can be used to back the guest storage. 2G is always attempted first. However, a 2G page size is not supported when z/OS is a z/VM guest. zCX will attempt a 4K page size. **System Action:** zCX attempst a 4K page size. **Operator Response:** None. **System Programmer Response:** If the appliance is never intended to be used on a native z/OS LPAR, then remove the 2G page frame size from the appliance configuration. Review the zCX documentation to ensure you are choosing the appropriate page frame size that fits your needs. **Routing code:** 2 **Descriptor code:** 4, 5

Table 32. Possible values for <fillin values> (continued)

Value and explanation
LFAREA 2G max would be exceeded. Terminating. **Explanation:** The appliance configuration specifies that only a 2G page size can be used to back the guest storage, but there is not enough space in the 2G Large Frame Area (LFAREA). **System Action:** zCX terminates. **Operator Response:** None. **System Programmer Response:** Determine why the required number of 2G pages were not available. Validate the guest storage size as it might be larger than intended. Review the zCX documentation that is related to how to choose a page size and how the LFAREA needs to be updated when a 2G page size is chosen. **Routing code:** 2 **Descriptor code:** 4, 5
LFAREA 2G max would be exceeded. Trying 1M fixed. **Explanation:** The appliance configuration specifies that at least a 2G or 1M page size can be used to back the guest storage. The system will attempt a 1M page size because there is not enough space in the 2G LFAREA. **System Action:** zCX will attempt a 1M page size. **Operator Response:** Notify the systems programmer. **System Programmer Response:** Determine why the required number of 2G pages were not available. Validate the guest storage size as it might be larger than intended. Review the zCX documentation related to how to choose a page size and how the LFAREA needs to be updated when choosing a 2G page size. **Routing code:** 2 **Descriptor code:** 4, 5

Table 32. Possible values for <fillin values> (continued)

Value and explanation
2G pages but not multiple of 2G. Terminating. **Explanation:** When a 2GB page frame size is used to back the guest, they can only be allocated in multiples of 2GB, but the guest storage is not a multiple of 2GB. No other page size was selected to be attempted. **System Action:** zCX attempts a 1M page size. **Operator Response:** Notify the systems programmer. **System Programmer Response:** Correct the guest storage size so it is a multiple of 2GB, or allow 1M or 4K page sizes. **Routing code:** 2 **Descriptor code:** 4, 5
2G pages but not multiple of 2G. Trying 1M fixed. **Explanation:** When a 2G page frame size is used to back the guest, they can only be allocated in multiples of 2GB, but the guest storage is not a multiple of 2GB. As the appliance configuration specifies that a 1M page size can also be used, the system attempts a 1M page size. **System Action:** zCX will not use a 2G page size to back the guest. A 1M page size is attempted. **Operator Response:** Notify the systems programmer. **System Programmer Response:** Correct the guest storage size so it is a multiple of 2GB. **Routing code:** 2 **Descriptor code:** 4, 5
2G pages but not multiple of 2G. Trying 4K pageable then fixed. **Explanation:** When a 2GB page frame size is used to back the guest, they can only be allocated in multiples of 2GB, but the guest storage is not a multiple of 2GB. As the appliance configuration specifies that a 4K page size can also be used, the system attempts a 4K page size. **System Action:** zCX will not use a 2GB page size to back the guest. A 4K page size is attempted. **Operator Response:** Notify the systems programmer. **System Programmer Response:** Correct the guest storage size so it is a multiple of 2GB. **Routing code:** 2 **Descriptor code:** 4, 5
LFAREA 1M max would be exceeded. Trying 4K pageable then fixed. **Explanation:** The appliance configuration specifies that at least a 1M or 4K page size can be used to back the guest storage. The system attempts a 4K page size because there is not enough space in the 1M LFAREA. **System Action:** zCX attempts a 4K page size next. **Operator Response:** Notify the systems programmer. **System Programmer Response:** Validate the guest storage size as it might be larger than intended. Review the zCX documentation that is related to how to choose a page size and how the LFAREA needs to be updated when choosing 1M pages. **Routing code:** 2 **Descriptor code:** 4, 5

Table 32. Possible values for <fillin values> (continued)

Value and explanation
LFAREA 1M max would be exceeded. Terminating. **Explanation:** The system could not use a 1M page size to back the guest because there is not enough space in the 1M LFAREA. A 4K page size was not selected when the appliance was configured. **System Action:** zCX terminates. **Operator Response:** Notify the systems programmer. **System Programmer Response:** Validate the guest storage size as it might be larger than intended. Review the zCX documentation that is related to how to choose a page size and how the LFAREA needs to be updated when 1M pages are chosen. **Routing code:** 2 **Descriptor code:** 4, 5
2G or 1M pages not supported on zPDT®. Trying 4K pages. **Explanation:** The system cannot use a 1M page size to back the guest because there is not enough space in the 1M LFAREA. A 4K page size was not selected when the appliance was configured. **System Action:** zCX attempts to allocate storage using 4K pages. **Operator Response:** Notify the systems programmer. **System Programmer Response:** Reconfigure the workflow to provision only 4K page frame size, if necessary. **Routing code:** 2 **Descriptor code:** 4, 5
2G or 1M pages not supported on zPDT. Terminating. **Explanation:** zCX does not support 2G or 1M pages on a zPDT system. **System Action:** zCX terminates.

Table 32. Possible values for <fillin values> (continued)

Value and explanation
Operator Response: Notify the systems programmer. **System Programmer Response:** Reconfigure the workflow to provision guest memory using only 4K page frame size. **Routing code:** 2 **Descriptor code:** 4, 5

System action:
Processing continues.

Operator response:
Follow the response specific to the *<fillin values>* in the above table.

System programmer response:
Follow the response specific to the *<fillin values>* in the above table. If necessary, call IBM.

Routing code:
11

Descriptor code:
6

GLZB025I **zCX instance *<jobname>*: Initialization is starting. Code date *<mm/dd/yy>*.**

Explanation

The zCX initialization process started. The date <mm/dd/yy> is the most recent compiled code date of any module in the zCX code.

System action:
The system continues zCX initialization.

Operator response:
None.

System programmer response:
None.

Routing code:
11

Descriptor code:
6

GLZB026I **zCX instance *<jobname>*: Consider reconfiguring the appliance: Guest page frame size not selected, 4K being used.**

Explanation

Since the appliance was last configured, support was added to allow the choice of the best z/OS page frame size to back the guest storage. A page frame size other than 4K might be more appropriate for your workload.

System action:
The system continues using a 4K page frame size to back the guest storage.

Operator response:
Notify the system programmer.

System programmer response:
Review the zCX documentation on how to choose a page frame size and reconfigure the appliance.

Routing code:
4, 5

Descriptor code:
2

GLZB027I **zCX instance *<jobname>*: *<initialization phase>***

Explanation

zCX proceeded to the next specified phase of the initialization process. In the message:

<initialization phase>
The phase can be one of the following:

- Allocating Linux guest memory.
- Loading Linux image.
- Starting virtual CPUs.
- IPLing guest and starting Docker services.

System action:
The system continues zCX initialization with the indicated phase.

Operator response:
None.

System programmer response:
None.

Routing code:
11

Descriptor code:
6

GLZB028I **zCX instance *<jobname>*: Guest memory size cannot be accommodated. Maximum guest memory virtual storage address would exceed: *<hardware limit>***

Explanation

The instance failed to instantiate because there is not enough contiguous high private virtual storage in the zCX instance address space to accommodate the guest memory size. The *<hardware limit>* is the highest virtual address that can be used. The instance must be re provisioned using a smaller guest memory size so it fits. In the message:

<jobname>
The name of the zCX instance.

<hardware limit>
The highest virtual address that the hardware can support for the guest memory. A value of 0 indicates that the limit might not be determined but the condition is still valid.

System action:
zCX instance terminates.

Operator response:
Consult the system programmer.

System programmer response:
The guest virtual memory must be contiguous and fit in the high virtual user region. The two sections of the high virtual user region are below and above the high virtual common and shared areas. The *<hardware limit>* determines which sections of the user region can be used. The subset of the high user region that is below the *<hardware limit>* might not be available for guest memory use due to other system usage of high virtual private in the address space. You might be able to decrease the size of HVCOMMON or HVSHARE to increase the size of the private area below the hardware virtual storage range. Doing so will accommodate a larger guest size but requires an IPL and that the storage is available. For further discussion of the address space virtual storage map and high virtual private and what system parameters that affect it, see the *z/OS MVS Initialization and Tuning Reference*.

Routing code:
2, 10, 11

Descriptor code:
5

GLZC001I **Command not recognized.**

Explanation:
The command is not recognized.

System action:
The command is not processed.

Operator response:
Correct and re-enter the command.

System programmer response:
Correct and re-enter the command.

Routing code:
*

Descriptor code:
5

GLZC002I **DUMP option is {missing|incorrect}.**

Explanation:
The DUMP option is missing or incorrect.

System action:
The command is not processed.

Operator response:
Correct and re-enter the command.

System programmer response:
Correct and re-enter the command.

Routing code:
*

Descriptor code:
5

GLZC003I **Configuration information for zCX instance *job_name*: *text*.**

Explanation

In response to a `MODIFY` command with the parameters `DISPLAY,CONFIG` directed to zCX instance *job_name*, the system displays configuration information for zCX instance *job_name* as follows:

Table 33. Configuration information as displayed by the system

Label	Variable
File Path	*file_path*
FFDC Path	*ffdc_path*
Dump Path	*dump_path*
Memory Size	*memory_size*
Number of CPUs	*number_cpus*
Number of Disks	*number_disks*
Number of Networks	*number_networks*
CTRACE Parmlib Member	*member_name*
Memory Page Size	*memory_page_size*
Memory Pages	*memory_pages*

- *job_name*: The job name of the current zCX instance
- *file_path*: The configuration file path
- *ffdc_path*: The directory where first failure data capture (FFDC) files are stored
- *dump_path*: The dump file path
- *memory_size*: The amount of memory that is provided to the Linux guest
- *number_cpus*: The number of virtual CPUs for this zCX instance
- *number_disks*: The number of disks defined for this zCX instance
- *number_networks*: The number of network connections defined for this instance
- *member_name*: The CTRACE parmlib member for this zCX instance
- *memory_page_size*: The number of memory pages defined for this instance
- *memory_pages*: The memory page size is one of 2G, 1M or 4K defined for this instance

The following example shows what the system might display for zCX instance ZCX233 when the operator enters the `MODIFY ZCX233,DISPLAY,CONFIG` command:

```
GLZC003I Configuration information for zCX
instance ZCX233
File Path: /u/ocoz/zcx_instances/ZCX233/
start.json
FFDC Path: /u/ocoz/zcx_instances/ZCX233/FFDC
Dump Path: /u/ocoz/zcx_instances/ZCX233/FFDC/
zcx-guest.dmp
Memory Size: 2GB
Number of CPUs: 4
Number of Disks: 5
Number of Networks: 1
CTRACE Parmlib Member: CTIGLZ00
Memory Pages: 2048
Memory Page Size: 1M fixed
```

System action:
None.

Operator response:
None.

System programmer response:
None.

Routing code:
*

Descriptor code:
5

GLZC004I **Disk information for zCX instance *job_name*: *text*.**

Explanation

The system displays disk information for zCX instance *job_name*as follows, with additional rows for each disk:

Table 34. Disk information as displayed by the system			
DevNo	**Size**	**Encrypted?**	**Data set name**
device_number	*disk_size*	*disk_encrypted*	*dataset_name*

Total number of disks: *number_disks*

- *job_name*: The job name of the current zCX instance
- *device_number*: The device number for this disk
- *disk_size*: The size of this disk (units follow)
- *disk_encrypted*: Whether the disk is encrypted. Possible values are Yes, if the disk is encrypted and No, if the disk is not encrypted.
- *dataset_name*: The data set name for this disk

System action:
None.

Operator response:
None.

System programmer response:
None.

Routing code:
*

Descriptor code:
5

GLZC005I **Network information for zCX instance *job_name*: *text***

Explanation

The system displays network information for zCX instance *job_name* as follows, with more rows for each device:

Table 35. Network information as displayed by the system			
DevNo	**Stack**	**MTU**	**IP Address**
device_number	*stack_name*	*max_message_size*	*ip_address*

Total number of networks: *number_networks*

- *job_name*: The job name of the current zCX instance
- *device_number*: The device number for this disk
- *stack_name*: The TCP/IP stack name for this network connection
- *max_message_size*: The maximum message size for this network
- *ip_address*: The IP address for this network connection
- *number_networks*: The number of network connections defined for the zCX instance.

System action:
None.

Operator response:
None.

System programmer response:
None.

Routing code:
*

Descriptor code:
5

GLZC006I **A stop command for zCX instance *job_name* has been accepted.**

Explanation:
The zCX instance *job_name* accepted the stop command and began stop processing.

System action:
The zCX instance begins stop processing.

Operator response:
None.

System programmer response:
None.

Routing code:
2, *

Descriptor code:
4,5

GLZC007I **A stop command is already in progress for zCX instance *job_name*.**

Explanation:
A stop command is already in progress for zCX instance *job_name*.

System action:
The stop command is ignored.

Operator response:
None.

System programmer response:
If the zCX instance does not stop normally, use the FORCE command to terminate the instance.

Routing code:
*

Descriptor code:
5

GLZC008I **Disk Version information for zCX instance *job_name*.**

Explanation

This message is issued in response to an F *job_name*,display,diskver command.

Table 36. Disk version information as displayed by the system

DevNo	Data Set Name	Version
device_number	*dataset_name*	*version*

root_info
Total number of networks: *number_networks*

- *device_number*: The device number for this disk
- *dataset_name*: The data set associated with the disk
- *version*: The version information provided in the data set
- *root_info*: Additional information provided for root disk only

System action:
The system displays the disk devices that are associated with the zCX instance. Additionally, it displays Version information. For root disks, an extra line of version information is present. This extra line is only available for the root.

Operator response:
None.

System programmer response:
None.

Routing code:
*

Descriptor code:
5

GLZC009I **Stop command rejected. zCX instance *job_name* is not ready to accept a Stop command.**

Explanation:
The Stop command was rejected. The command was issued too soon after the Start command.

System action:
The Stop command is rejected. The zCX instance continues processing.

Operator response:
Re-issue the command after message GLZM004I has been issued, or use the Force command to terminate the instance.

System programmer response:
None.

Routing code:
10, *

Descriptor code:
5

GLZC010I **A Dump Guest command for zCX instance *job_name* has been accepted.**

Explanation:
A Dump Guest command for the zCX instance indicated is accepted. A dump file is written to the dump path of the instance configuration file.

System action:
The zCX instance continues processing.

Operator response:
None.

System programmer response:
None.

Routing code:
2

Descriptor code:
4,5

GLZC011I **A complete dump of the guest memory for zCX instance *job_name* has been written to file *dumppath*.**

Explanation:
The Dump Guest command for the zCX instance is complete. The dump file is written to *dumppath*.

System action:
The zCX instance continues processing.

Operator response:
None.

System programmer response:
None.

Routing code:
2

Descriptor code:
4, 5

GLZC012I **A dump of the guest memory for zCX instance *job_name* has failed.**

Explanation:
The dump guest command for the zCX instance failed. Verify that the dump path exists and can be written to by the zCX user ID.

System action:
The zCX instance continues processing.

Operator response:
Check the operator console to understand what caused the failure.

System programmer response:
None.

Routing code:
2

Descriptor code:
4,5

GLZM001I CTRACE option *option* is not valid.

Explanation:
The option *option* is not valid.

System action:
The zCX instance does not use the specified CTRACE parmlib member.

Operator response:
None.

System programmer response:
Correct the specified option in the CTRACE parmlib member used by the zCX instance.

Routing code:
10, *

Descriptor code:
12

GLZM002I CTRACE parmlib member *member_name* could not be used. Switching to member CTIGLZ00.

Explanation:
CTRACE parmlib member *member_name* is invalid. Default member CTIGLZ00 will be used instead.

System action:
Member CTIGLZ00 will be used as the CTRACE parmlib member.

Operator response:
None.

System programmer response:
Verify the parmlib member *member_name* exists. Adjust as necessary.

Routing code:
10, *

Descriptor code:
12

GLZM003I CTRACE parmlib member CTIGLZ00 could not be used. Switching to the default settings.

Explanation:
CTRACE parmlib member CTIGLZ00 could not be used. Switching to the default settings.

System action:
The default parmlib member settings are used.

Operator response:
None.

System programmer response:
Correct the unusable CTIGLZ00 parmlib member. Reset the CTRACE using the TRACE CT command, or restart the zCX appliance instance that uses the CTIGLZ00 parmlib member or another parmlib member containing valid zCX CTRACE configuration options.

Routing code:
10, *

Descriptor code:
12

GLZM004I zCX Docker services for instance *job_name* are available.

Explanation:
zCX Docker services for instance *job_name* are available.

System action:
None.

Operator response:
None.

System programmer response:
None.

Routing code:
2/*

Descriptor code:
4/5

GLZM005I zCX Docker services for instance *job_name* are not available.

Explanation:
zCX Docker services for instance *job_name* are not available. See accompanying messages in the appliance job log. This message is expected when the STOP command is issued.

System action:
All containers that are running in the zCX instance are terminated. No Docker commands can be issued at this point.

Operator response:
If this message was not issued in response to a STOP command, issue the STOP command to the zCX instance and restart the instance.

System programmer response:
Check for accompanying messages generated from this job. Correct the problem and retry. Contact IBM service for help if necessary.

Routing code:
2/*

Descriptor code:
4/5

GLZM006I **zCX instance *job_name* memory utilization exceeded threshold. Utilization is *percent*%.**

Explanation:
The zCX instance *job_name* is using *percent*% of the memory that is configured for it. If the memory utilization reaches 100%, the zCX instance might restart.

System action:
The zCX instance continues processing.

Operator response:
Report this problem to the zCX appliance instance docker administrator.

System programmer response:
None.

Routing code:
2, 10, *

Descriptor code:
4

Docker Administrator Response:
Reduce memory utilization. Some possible short-term actions are: Stop one or more containers that are running in the zCX instance. Stop and restart the zCX instance. Long-term actions might include: Reduce the number of containers running in the zCX instance. Reconfigure the zCX instance to have more memory. Reconfigure the zCX instance to have more SWAP disk volumes.

GLZM007I **zCX instance *job_name* memory utilization is below threshold. Utilization is *percent*%.**

Explanation:
The zCX instance memory utilization exceeded the threshold, but is now below the threshold. The memory utilization fell to *percent*%.

System action:
The zCX instance continues processing.

Operator response:
None.

System programmer response:
None.

Routing code:
2,*

Descriptor code:
4

GLZM008I **zCX instance *job_name* is starting failure data for appliance alert: *alert_message*.**

Explanation:
The zCX instance *job_name* is about to copy failure data from the appliance to the z/OS UNIX file system. The data is stored in the directory specified for the *ffdc-path* option in the startup file. The *alert_message* describes the reason why the data was collected.

System action:
The zCX instance continues processing.

Operator response:
None.

System programmer response:
None.

Routing code:
2

Descriptor code:
4

GLZM009I **zCX instance *job_name* stored failure data:**
Size=*file_size*
Checksum=*file_checksum*
File name=*file_name*
Directory=*ffdc-path*

Explanation:
The zCX instance *job_name* has copied failure data from the appliance to the z/OS UNIX file system. In the directory *ffdc-path*, the file *file_name* was stored. The size of the file is *file_size*. The checksum for the file is *file_checksum*. The size of the file and the checksum can be used to verify that no data is lost when the file is moved to another location.

System action:
The zCX instance continues processing.

Operator response:
Contact the system programmer.

System programmer response:
Review the alert described in message GLZM008I. Contact IBM support for help if necessary.

Routing code:
2

Descriptor code:
4

GLZM010I **zCX instance *job_name* failed to copy failure data from the DLOG device to the z/OS UNIX file system.**
reason_text

Explanation
The zCX instance *job_name* failed to copy failure data from the appliance to the z/OS UNIX file system. The possible values for *reason_text* are:

- The length of the alert message was 0 or greater than 1024.
- The DLOG device was not found.
- FFDC path name was not specified in the startup file.
- The length of the file name was 0 or greater than 255.
- The length of the output file path was greater than 1023.
- Storage could not be obtained.
- The output file could not be opened.
- The DLOG device could not be read.
- The output file could not be written.

System action:
The zCX instance continues processing.

Operator response:
Contact the system programmer.

System programmer response:
Check the job log for more messages. GLZB004I or GLZB005I is issued in some cases. For output file problems, such as running out of space or no authorization, correct the problem. Contact IBM support for help if necessary.

Routing code:
2

Descriptor code:
4

GLZM011I **zCX instance *job_name* alert: *resource severity*. Utilization at *percent*%.**

Explanation

In the message:

job_name

The job name of the zCX instance.

resource

The resource being monitored as indicated in the alert message. This can be: "memory", "swap disk", "root disk", or "data disk."

severity

The new severity of the resource being monitored. This can be: "critical", "warning"', or "normal."

***percent*%**

The current utilization percentage of the monitored resource. This is an integer.

zCX is reporting for instance *job_name* that the utilization of resource *resource* has significantly changed. The message shows the resource name and current utilization as a percentage. If the resource is exhausted, the zCX instance may unexpectedly fail. To prevent this, review the System Programmer Actions below.

The *severity* and *percent* are related in the following ways:

- Severity "normal" has utilization of 0%-69%
- Severity "warning" has utilization of 70%-84%
- Severity "critical" has utilization of 85%-100%

A message of type "normal" will be issued if:

- Utilization is < 70% and the last severity was "critical"

OR

- Utilization is <= 65% and the last severity was "warning"

A message of type "warning" will be issued if:

- Utilization is >= 70% and < 85% and the last severity was "normal"

OR

- Utilization is <= 80% and the last severity was "critical"

A message of type "critical" will be issued if:

- utilization is >= 85% and the last severity was "warning" or "normal"

System action:
The zCX instance continues processing.

Operator response:
Notify the systems programmer.

System programmer response

If these messages cause concern, please engage your Docker administrator/Linux system programmer to select a Linux monitoring tool of choice to perform further analysis.

Possible remedial actions for critical and warning severity messages are dependent on the resource involved:

- For Memory: Stop running containers. Add memory via "reconfigure" workflow.
- Swap disk: Stop running containers. Add swap disk(s) via "add disk" workflow.
- Root disk: Resize root disk via "upgrade" workflow.
- Data disk: Remove data files. Prune stale container images. Remove unneeded docker volumes. Add data disks via "add disk" workflow.

Note that running out of memory, swap disk, or root disk space may cause the zCX instance to fail. Running

out of data disk space may cause containers within the instance to fail.

Routing code:
2,11

Descriptor code:
6

GLZN001I ***network_type*** **network at IP@=*ip_address* failed to initialize RC=*return_code* RSN=*reason_code* for zCX instance *job_name*.**

Explanation:
The *network_type* of IPv4 or IPv6 at IP address *ip_address* failed to initialize for zCX instance *job_name*.

System action:
Network services are not available for this IP address. If no networks are available, the instance terminates.

Operator response:
See Chapter 3 in the *z/OS Communications Server: IP and SNA Codes* for an explanation of the codes. Data link control (DLC) status codes explain the meaning of the return codes and reason codes in the message. Contact your system programmer for additional assistance as necessary.

System programmer response:
See Chapter 3 in the *z/OS Communications Server: IP and SNA Codes* for an explanation of the codes. Data link control (DLC) status codes explain the meaning of the return codes and reason codes in the message.

Routing code:
10, *

Descriptor code:
5

GLZN002I ***network_type*** **network at IP@=*ip_address* failed to confirm a connection after multiple attempts for zCX instance *job_name*.**

Explanation:
The *network_type* of IPv4 or IPv6 at IP address *ip_address* failed to confirm that the zCX instance *job_name* is connected to the network.

System action:
Network services are not available for this IP address. If no networks are available, the instance terminates.

Operator response:
None.

System programmer response:
See Chapter 3 in the *z/OS Communications Server: IP and SNA Codes* for an explanation of the codes. Data link control (DLC) status codes explain the meaning of the return codes and reason codes in the message. Check for possible messages from Communications Server. Restart the zCX instance if necessary.

Routing code:
10,*

Descriptor code:
5

GLZN003I ***network_type*** **network at IP@=*ip_address* disconnected for zCX instance *job_name*.**

Explanation:
The *network_type* of IPv4 or IPv6 at IP address *ip_address* failed to (re)connect for *job_name*.

System action:
The network remains disconnected.

Operator response:
None.

System programmer response:
Determine and correct the source of the network problem. See Chapter 3 in the *z/OS Communications Server: IP and SNA Codes* for an explanation of the codes. Re-establish network connectivity and then re-start the server instance.

Routing code:
10

Descriptor code:
4

GLZN004I **zCX instance *job_name* failed network connection at IP@=*ip_address* Attempt *attempt* of *total_attempts* will be retried.**

Explanation:
The server instance that is associated with *job_name* is attempting to reconnect to the network. It will retry the total number of times indicated.

System action:
The server instance tries to reconnect the network. There is a delay between attempts to allow the network to recover network issues itself.

Operator response:
None.

System programmer response:
No action is necessary. It might be helpful to understand the reason for the network disconnect.

Routing code:
10

Descriptor code:
4

GLZN005I ***network_type*** **network at IP@=*ip_address* successfully**

reconnected for zCX instance *job_name*.

Explanation:
The *network_type* of IPv4 or IPv6 at IP address *ip_address* has been reconnected.

System action:
The system continues.

Operator response:
None.

System programmer response:
None.

Routing code:
10

Descriptor code:
4

GLZS001I **zCX instance *job_name* entered a disabled wait state. PSW=*psw0-3 psw4-7 psw8-11 psw12-15***

Explanation

zCX instance *job_name* entered a disabled wait state. The PSW information is as follows:

- *psw0-3*: Bytes 0 - 3 of the program status word (PSW)
- *psw4-7* : Bytes 4 - 7 of the program status word (PSW)
- *psw8-11*: Bytes 8 - 11 of the program status word (PSW)
- *psw12-15*: Bytes 12 - 15 of the program status word (PSW)

System action:
The zCX instance terminates.

Operator response:
Attempt to restart the zCX instance.

System programmer response:
Call IBM Service.

Routing code:
10, *

Descriptor code:
4

GLZV001I **MMSRV CONNECT service failed RC=*return_code*, DSN=*dataset_name*.**

Explanation:
The MMSRV CONNECT service failed with the indicated return code and data set name.

System action:
The zCX instance terminates.

Operator response:
None.

System programmer response:
Look for accompanying DFSMS messages. Investigate the cause of the MMSRV failure that uses the return code. Adjust accordingly. Check for possible IEC161I message for a VSAM OPEN error.

Routing code:
10,*

Descriptor code:
5

GLZV002I **zCX instance *job_name* is formatting available space for use by DSN=*dataset_name*.**

Explanation:
The system is formatting unused space in the named data set to make the full capacity available.

System action:
None.

Operator response:
None.

System programmer response:
None.

Routing code:
2,*

Descriptor code:
4.5

GLZV003I **zCX instance *job_name* formatting complete for DSN=*dataset_name*.**

Explanation:
The system completed formatting of the named data set.

System action:
None.

Operator response:
None.

System programmer response:
None.

Routing code:
2, *

Descriptor code:
4, 5

GLZV004I **zCX instance *job_name* cannot be started. VSAM data set *dataset_name* is corrupted.**

Explanation:
zCX was unable to use the data set requested in the appliance startup file. The data set was found, but zCX

was unable to recognize the contents of the data set as its own.

System action:
The zCX instance that is identified by *job_name* terminates.

Operator response:
None.

System programmer response:
Correctly provision the VSAM data set that uses the zCX workflows.

Routing code:
10, *

Descriptor code:
5

GLZV005I **zCX instance *job_name: mm_sr* service failed RC=*return_code*, DSN=*dataset_name*.**

Explanation:
The Media Manager service *mm_srv* failed with the indicated return code and data set name.

System action:
The zCX instance terminates.

Operator response:
None.

System programmer response:
Look for possible accompanying DFSMS messages. Investigate the cause of the service failure that uses the return code.

Routing code:
10,*

Descriptor code:
5

AZD messages

AZDB0001E **Error *errcode* copying configuration disk contents**

Explanation

An error occurred copying the configuration disk contents into the zCX instance. In the message text:

errcode is an internal error number.

System action

The zCX instance continues processing using the current configuration.

Operator response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

Programmer response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

AZDB0002E **Error installing ILMT scanner, rc=*retcode***

Explanation

An error occurred installing the ILMT scanner. In the message text:

retcode is the ILMT scanner installer return code.

System action

The zCX instance continues processing without ILMT active.

Operator response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

Programmer response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

AZDB0003E **Error uninstalling ILMT scanner, rc=*retcode***

Explanation

An error occurred uninstalling the ILMT scanner. In the message text:

retcode is the ILMT scanner uninstaller return code.

System action

The zCX instance continues processing.

Operator response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

Programmer response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

AZDB0004E **Error creating ILMT scanner configuration, rc=*retcode***

Explanation

An error occurred configuring the ILMT scanner. In the message text:

retcode is the ILMT scanner configure return code.

System action

The zCX instance continues processing without ILMT active.

Operator response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

Programmer response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

AZDB0005E **Error creating ILMT output directory, rc=*retcode***

Explanation

An error occurred configuring the ILMT scanner. In the message text:

retcode is an internal error code indicating the failure.

System action

The zCX instance continues processing without ILMT active.

Operator response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

Programmer response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

AZDB0006E **Error updating ILMT scanner configuration, rc=*retcode***

Explanation

An error occurred configuring the ILMT scanner. In the message text:

retcode is the ILMT scanner configure return code.

System action

The zCX instance continues processing without ILMT active.

Operator response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

Programmer response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

AZDB0007E **Error creating ILMT &mountpoint mountpoint, rc=*retcode***

Explanation

An error occurred configuring the ILMT scanner. In the message text:

retcode is an internal error code indicating the failure.

System action

The zCX instance continues processing without ILMT active.

Operator response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

Programmer response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

AZDB0008E **Error mounting ILMT &mountpoint mountpoint, rc=*retcode***

Explanation

An error occurred configuring the ILMT scanner. In the message text:

retcode is an internal error code indicating the failure.

System action

The zCX instance continues processing without ILMT active.

Operator response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

Programmer response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

AZDB0009E ROOT disk critical free space shortage, *inuse* percent used

Explanation

The zCX instance ROOT disk is nearly full. Once the ROOT disk is full, the zCX instance will be unable to start. In the message text:

inuse is the percentage of the ROOT disk in use.

System action

The zCX instance continues processing.

Operator response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

Programmer response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

If using the journaled Docker logging driver, consider switching to the json-file Docker logging driver. This will move the Docker logs from the ROOT disk to the DATA disk(s).

If the zCX instance is unable to be started, use the zCX upgrade workflow to overwrite the ROOT disk with a fresh appliance image, optionally increasing the size of the ROOT disk to provide more space.

AZDD0001E Failure *errcode* locating custom image container

Explanation

An internal error occurred searching for the zCX CLI SSH container. In the message text:

errcode is the internal error code.

System action

The zCX instance continues processing with the prior zCX CLI SSH container.

Operator response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

Programmer response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

AZDD0002E Failure *errcode* running custom image container

Explanation

An internal error occurred on the docker run command for the zCX CLI SSH container. In the message text:

errcode is the internal error code.

System action

The zCX instance continues processing with the prior zCX CLI SSH container.

Operator response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

Programmer response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

AZDD0003E Failure *errcode* starting custom image container

Explanation

An internal error occurred on the docker start command for the zCX CLI SSH container. In the message text:

errcode is the internal error code.

System action

The zCX instance continues processing with the prior zCX CLI SSH container.

Operator response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

Programmer response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

AZDD0004E **Failure *errcode* building custom image**

Explanation

An internal error occurred on the docker build command for the zCX CLI SSH container. In the message text:

errcode is the internal error code.

System action

The zCX instance continues processing with the prior zCX CLI SSH container.

Operator response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

Programmer response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

AZDD0005E **Failed to create Docker Registry directory: *dirname***

Explanation

An internal error occurred installing the Docker proxy private CA certificate. In the message text:

dirname is the name of the directory that was not created .

System action

The zCX instance continues processing without the certificate.

Operator response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

Programmer response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

AZDD0006E **Failure *errcode* installing Docker proxy private CA certificate**

Explanation

An internal error occurred installing the Docker proxy private CA certificate. In the message text:

errcode is the internal error code.

System action

The zCX instance continues processing without the certificate.

Operator response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

Programmer response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

AZDD0007E **Failure *errcode* check-summing Docker proxy private CA certificate**

Explanation

An internal error occurred validating the Docker proxy private CA certificate. In the message text:

errcode is the internal error code.

System action

The zCX instance continues processing without the certificate.

Operator response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

Programmer response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

AZDD0008E **Failure *errcode* removing Docker proxy private CA certificate**

Explanation

An internal error occurred removing a Docker proxy private CA certificate. In the message text:

errcode is the internal error code.

System action

The zCX instance continues processing without removing the certificate.

Operator response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

Programmer response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

AZDD0009E Failure *errcode* updating ca-certificate.crt

Explanation

An internal error occurred updating a CA certificate. In the message text:

errcode is the internal error code.

System action

The zCX instance continues processing with the prior certificate.

Operator response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

Programmer response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

AZDL0001E Unexpected error *errcode*, rsn1=*rsn1code* rsn2=*rsn2code*

Explanation

An unexpected error occurred during start of the zCX instance. In the message text:

errcode is the error code .
rsn1code and *rsn2code* are the reason codes.

System action

The zCX instance ends.

Operator response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

Programmer response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

AZDL0002E No purpose=root device found

Explanation

zCX could not find the ROOT disk.

System action

The zCX instance ends.

Operator response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

Programmer response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

AZDL0003E No purpose=config device found

Explanation

zCX could not find the CONFIG disk.

System action

The zCX instance ends.

Operator response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

Programmer response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

AZDL0004E At least one prior boot failed

Explanation

One or more prior starts of the zCX instance failed.

System action

The zCX instance continues processing.

Operator response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

Programmer response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

AZDM0001E **Unexpected error *errcode* performing server maintenance**

Explanation

An internal error occurred performing an automated maintenance process. In the message text:

errcode is the internal error code.

System action

The zCX instance continues processing.

Operator response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

Programmer response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

AZDN0001E **Failure *errcode* configuring network time protocol**

Explanation

An internal error occurred configuring the Linux guest network time protocol. In the message text:

errcode is the internal error code.

System action

The zCX instance continues processing without changing the network time protocol settings.

Operator response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

Programmer response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

AZDN0002E **Failure *errcode* configuring DNS search domains**

Explanation

An internal error occurred configuring the DNS search domains provided by the zCX admin. In the message text:

errcode is the internal error code.

System action

The zCX instance continues processing with the prior DNS search domain settings.

Operator response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

Programmer response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

AZDN0004E **Failure *errcode* configuring IPv4 address**

Explanation

An internal error occurred configuring the IPv4 address. In the message text:

errcode is the internal error code.

System action

The zCX instance continues processing with the prior IPv4 address.

Operator response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

Programmer response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

AZDP0001E **Unexpected error *errcode* configuring data disks**

Explanation

An internal error occurred configuring one or more data disks. In the message text:

errcode is the internal error code.

System action

The zCX instance continues processing with the prior data disks configuration.

Operator response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

Programmer response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

AZI messages

AZIF0144E **An unexpected error occurred in LPAR *lparname***

Explanation

The Linux guest abnormally ended. In the message text:

lparname is the LPAR containing the zCX instance

System action

The zCX instance's Linux guest is restarted.

Operator response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

Programmer response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

AZIF0146E **An unexpected error occurred in LPAR *lparname***

Explanation

A user program abnormally ended. The failing program may be running in a Docker container or in the Linux guest. In the message text:

lparname is the LPAR containing the zCX instance

System action

The zCX instance continues processing.

Operator response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

Programmer response

Refer to message GLZM009I in *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

If the issuance of this message is related to the running of a Docker container, investigate the container's logs.

Reason codes for zCX termination message

zCX uses the following reason codes to further describe the reasons for termination.

Table 37. Reason codes for zCX termination message

Return Code	Reason Code	Condition	Description
'4'X	'0115004'X	Warning.	The trial period has ended.
'8'X	'01010005'X	Parameter error.	The length of the required startup file name is not valid. If zero, the startup file name was not specified as required. If nonzero, it was either too short to be a valid file name or too long. The file name must have between 2 and 255 characters (inclusive).

Table 37. Reason codes for zCX termination message (continued)

Return Code	Reason Code	Condition	Description
'8'X	'01010008'X	System service error.	ISGENQ returned an unexpected return and reason code. Unable to obtain the ENQ intended to ensure that every zCX instance in the sysplex has a unique job name. Since zCX cannot guarantee uniqueness of job names, it terminates.
'8'X	'0101000B'X	Parameter error.	The zCX startup file could not be processed by GLZBAPCF (which should have arranged for messages to explain the issues that were encountered).
'8'X	'01150006'X	Parameter error.	The zCX registry path could not be determined from the CONF parameter to the GLZ PROC. The path name specified had no slashes. Correct the CONF parameter and start again.
'8'X	'01150007'X	Parameter error.	The zCX registry path could not be determined from the CONF parameter to the GLZ PROC. The path name specified had only one slash. Correct the CONF parameter and start again.
'C'X	'01010006'X	Environmental error.	Not licensed to run zCX. Message GLZB015I was issued to describe the error.
'C'X	'01010007'X	Environmental error.	There is another zCX instance running in the sysplex with the same job name. zCX terminates because each zCX instance running in the sysplex must have a unique job name.

Table 37. Reason codes for zCX termination message (continued)

Return Code	Reason Code	Condition	Description
'C'X	'0101000C'X	Environmental error.	Unable to create and initialize the control blocks for the virtual CPUs.
'C'X	'0101000D'X	Environmental error.	Unable to create a latch set to be used for SDEV serialization.
'C'X	'0101000E'X	Environmental error.	Unable to create a latch set to be used for VDEV serialization.
'C'X	'0101000F'X	Environmental error.	z/OS UNIX System Services could not dub the zCX address space.
'C'X	'01010010'X	Environmental error.	Unable to load the DNIC load module.
'C'X	'01010011'X	Environmental error.	Unable to establish ESTAE recovery environment.
'C'X	'01010012'X	Environmental error.	Unable to load the Linux image into guest memory.
'C'X	'01010013'X	Environmental error.	Unable to attach a task for each of the virtual CPUs.
'C'X	'01010014'X	System error.	The guest successfully initialized but Docker was not shut down when the guest terminated.
'C'X	'01010015'X	System error.	The first bootloader failed. Check the joblog for messages that explain the failure.
'C'X	'01010016'X	System error.	The second bootloader failed. Check the joblog for messages that explain the failure.
'C'X	'01010017'X	System error.	The guest terminated because of a shutdown or halt request from an unknown source.
'C'X	'01010018'X	System error.	The guest terminated before it could complete its initialization.
'C'X	'01010019'X	System error.	The guest terminated without reporting any status.

Table 37. Reason codes for zCX termination message (continued)

Return Code	Reason Code	Condition	Description
'C'X	'01010020'X	System error.	The guest reported an unknown status before it terminated.
'C'X	'01150001'X	System error.	Time conversion processing overflowed. This is an unexpected system error.
'C'X	'0115002'X	Environmental Error	The trial start time was in the future relative to the current time. This is an unexpected error.
'C'X	'01150005'X	System error.	The length of the path name for the file that keeps track of the trial is greater than the size of the buffer where the path name will be stored. This is an internal error.
'C'X	'01150008'X	Environmental error.	The user ID for the zCX instance does not have permission to search one of directories in the path for the zCX registry. Message GLZB004I was issued. It has the path name.
'C'X	'01150009'X	Environmental error.	The path name for the zCX trial was found but it is not a regular file.
'C'X	'0115000A'X	System error.	The BPX4STA service failed. Message GLZB004I was issued. It has the error information and path name.
'C'X	'0115000B'X	Environmental error.	The file that is used to keep track of the trial already exists. This error could occur if multiple zCX instances are started at the same time when the trial has not yet begun. Retry the START command.

Table 37. Reason codes for zCX termination message (continued)			
Return Code	**Reason Code**	**Condition**	**Description**
'C'X	'0115000C'X	Environmental error.	The user ID for the zCX instance does not have permission to create a file in the zCX registry. Message GLZB004I was issued. It has the error information and path name.
'C'X	'0115000D'X	Environmental error.	The BPX4OPN service failed. Message GLZB004I was issued. It has the error information and path name.

C0D ABEND Reason Codes

Container Extensions reason codes

80040001
Bad RC from VSM Locate macro.

80040002
Bad RC from LOAD or NUCLKUP macros.

80040003
Unable to create a latch set needed for resource serialization. Message GLZB005I was issued to describe the failure. It has the return code from the service ISGLCRT and the name of the latch set.

80040004
A device for the zCX instance could not be initialized. The total number of disk and network devices would exceed the maximum of 255.

80040005
The number of virtqueues for a virtual disk or network device has been exceeded.

80040006
A disk device for the zCX instance could not be initialized. The total number of disk and network devices would exceed the maximum of 255.

80040007
The initial program for the zCX instance did not receive control as a started task. zCX only supports the started task environment. The START command must be used to start a zCX instance. This is a user error.

80040008
The guest used an indirect descriptor to provide buffers for incoming network traffic. This is not supported by the virtualization layer.

80040009
The guest used a chain of descriptors for an outgoing network request that were not in the expected format.

8004000A
The address of the prefix area (PSA) for one of guest's virtual CPUs was not within the guest memory.

8004000B
The guest used a chain of descriptors for a disk request that were not in the expected format.

8004000C
The guest used a chain of descriptors for a disk request that were not in the expected format.

8004000D
The virtualization layer received a request from the guest that it does not support. It responded with an operation exception. This should be reported to IBM.

8004000E
The guest sent a request that is not supported by the virtualization layer to a network device.

8004000F
The guest stopped unexpectedly. This was not a shutdown initiated by the STOP command. It may indicate that the guest encountered a problem.

80040010
The request to format the VSAM linear data set for one of the disks failed. Message GLZV002I was issued before the format. It contains the name of the data set.

80040011
The request to format the VSAM linear data set for one of the disks failed. Message GLZV002I was issued before the format. It contains the name of the data set.

80040012
The request to update the high used control interval of the VSAM linear data set for one of the disks failed. Message GLZV003I was issued before the update. It has the name of the data set.

80040013
The virtualization layer was unable to connect to the network. Time out occurred while waiting for a response to the connect request. Check the job log for messages about network problems. Restart the zCX instance after network problems have been resolved.

80040014
The state of the lock word, used by the virtualization layer to serialize with the guest, is not valid.

80040015
The task that handles communication with the z/OS console has terminated abnormally. Check for earlier messages or abends that may have led to the termination.

80040016
The first control interval of the VSAM linear data set for one of the disks did not contain the expected data. Message GLZV004I was issued to describe the failure. It has the name of data set.

80040017
The request to read the first control interval of the VSAM linear data set for one of the disks failed.

80040018
The request to read the first control interval of the VSAM linear data set for one of the disks failed.

80040019
A virtual CPU has encountered an unrecoverable error. The entire zCX instance has been terminated.

8004001A
The task that performs the connect and reconnects of the network(s) associated with this server instance has failed. This is an internal error. If no fix exists, contact the IBM Support Center.

8004001B
After multiple attempts to connect to the network, the server instance is unable to do so.

8004001C
The server is unable to find the disk needed to perform re-initialization for reboot processing. The entire zCX instance will terminate.

8004001D
Errors have been encountered trying to re-initialize a disk during reboot processing. The entire zCX instance has been terminated.

8004001F
Bad ALCPET encountered.

80040020
Bad Pause Pet.

80040021
Bad pause/release element.

80040022
Timer request to set when the timer is still set.

80040023
Invalid timer request.

80040024
Timer is not able to cancel the DIE.

80040025
Unexpected IEAVXTSWRC encountered.

80040026
TDS is requested to be set when still set.

80040027
Unexpected IEAVXTDSRC encountered.

80040028
Invalid TDS Request.

80040029
Internal error. If no fix exists, contact the IBM Support Center.

8004FFFF
Overflow of the virtio descriptor array.

Notices

This information was developed for products and services that are offered in the USA or elsewhere.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
United States of America

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

This information could include missing, incorrect, or broken hyperlinks. Hyperlinks are maintained in only the HTML plug-in output for IBM Documentation. Use of hyperlinks in other output formats of this information is at your own risk.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Site Counsel
2455 South Road

Poughkeepsie, NY 12601-5400
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of 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.

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

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or

reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

IBM Online Privacy Statement

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

Depending upon the configurations deployed, this Software Offering may use session cookies that collect each user's name, email address, phone number, or other personally identifiable information for purposes of enhanced user usability and single sign-on configuration. These cookies can be disabled, but disabling them will also eliminate the functionality they enable.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at ibm.com/privacy and IBM's Online Privacy Statement at ibm.com/privacy/details in the section entitled "Cookies, Web Beacons and Other Technologies," and the "IBM Software Products and Software-as-a-Service Privacy Statement" at ibm.com/software/info/product-privacy.

Policy for unsupported hardware

Various z/OS elements, such as DFSMSdfp, JES2, JES3, and MVS, contain code that supports specific hardware servers or devices. In some cases, this device-related element support remains in the product even after the hardware devices pass their announced End of Service date. z/OS may continue to service element code; however, it will not provide service related to unsupported hardware devices. Software problems related to these devices will not be accepted for service, and current service activity will cease if a problem is determined to be associated with out-of-support devices. In such cases, fixes will not be issued.

Minimum supported hardware

The minimum supported hardware for z/OS releases identified in z/OS announcements can subsequently change when service for particular servers or devices is withdrawn. Likewise, the levels of other software products supported on a particular release of z/OS are subject to the service support lifecycle of those

products. Therefore, z/OS and its product publications (for example, panels, samples, messages, and product documentation) can include references to hardware and software that is no longer supported.

- For information about software support lifecycle, see: IBM Lifecycle Support for z/OS (www.ibm.com/software/support/systemsz/lifecycle)
- For information about currently-supported IBM hardware, contact your IBM representative.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at Copyright and Trademark information (www.ibm.com/legal/copytrade.shtml).

Index

A

F

M

S

T

Z

Product Number: 5650-ZOS