

Using IBM z/OS Provisioning Toolkit

Version 1 Release 1



Using IBM z/OS Provisioning Toolkit

Version 1 Release 1

Note

Before using this information and the product it supports, read the information in “Notices” on page 171.

This edition applies to the IBM z/OS Provisioning Toolkit, Version 1 Release 0 (product number 5655-CI1) and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright IBM Corporation 2017, 2019.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

About this book	v	Link scenarios supported by z/OS Provisioning Toolkit	54
Chapter 1. What is z/OS Provisioning Toolkit?	1	Managing z/OS Provisioning Toolkit images	55
How does z/OS Provisioning Toolkit work?	3		
What documentation is available for z/OS Provisioning Toolkit?	5		
Chapter 2. z/OS Provisioning Toolkit change history.	7		
Chapter 3. Prerequisites for using z/OS Provisioning Toolkit	17		
Chapter 4. Installing z/OS Provisioning Toolkit	19		
Chapter 5. Upgrading z/OS Provisioning Toolkit	21		
Chapter 6. Configuring z/OS Provisioning Toolkit	23		
Chapter 7. Security planning for z/OS Provisioning Toolkit with IBM Cloud Provisioning and Management for z/OS	25		
Chapter 8. Getting started with z/OS Provisioning Toolkit	29		
Introduction to getting started with z/OS Provisioning Toolkit	29		
Getting started with CICS by using z/OSMF workflows.	30		
Getting started with CICS by using IBM Cloud Provisioning and Management for z/OS.	34		
Getting started with other middleware	39		
Chapter 9. Preparing to use z/OS Provisioning Toolkit with IBM Cloud Provisioning and Management for z/OS	41		
Chapter 10. Adding an IBM Cloud Provisioning and Management for z/OS template for use with z/OS Provisioning Toolkit	45		
Chapter 11. z/OS Provisioning Toolkit images.	47		
Building an image with z/OS Provisioning Toolkit	49		
Running a z/OS Provisioning Toolkit image	50		
		Chapter 12. Provisioning CICS with z/OS Provisioning Toolkit.	57
		Configuration prerequisites for using z/OS Provisioning Toolkit with CICS.	57
		Overview of supported CICS types	58
		Adding a CICS template by using IBM Cloud Provisioning and Management for z/OS.	59
		Enabling faster provisioning by using parallel processing.	63
		Using dynamic port allocation in a template	64
		Including resources in an image	66
		Including a CICS bundle in your image	67
		Including program binary files in your image	69
		Including CICS resource definitions in an image	70
		Security requirements for provisioning CICS with z/OS Provisioning Toolkit	72
		Modifying the CICS workflows to support dynamic security.	73
		Deprovisioning guidelines for regions in a CICSplex	74
		Configuration properties for CICS images	75
		Output properties from provisioning CICS	92
		Modifying the set of CICS configuration properties that can be overridden.	93
		Job entry subsystem (JES) job names	94
		Troubleshooting CICS provisioning with z/OS Provisioning Toolkit	97
		Chapter 13. Provisioning z/OS Connect Enterprise Edition with z/OS Provisioning Toolkit	99
		Configuration prerequisites for using z/OS Provisioning Toolkit with z/OS Connect Enterprise Edition	99
		Adding a z/OS Connect Enterprise Edition template by using IBM Cloud Provisioning and Management for z/OS	100
		Security requirements for provisioning instances of z/OS Connect EE with z/OS Provisioning Toolkit	103
		Modifying the z/OS Connect Enterprise Edition workflows to support dynamic security	105
		Configuration properties for z/OS Connect Enterprise Edition images	106
		Output properties from provisioning IBM z/OS Connect Enterprise Edition	111

**Chapter 14. Managing containers
created by z/OS Provisioning Toolkit . . . 113**

**Chapter 15. Registering existing
subsystems in z/OSMF 117**

CICS registration information 119

IBM Db2 registration information. 120

IBM MQ registration information. 121

**Chapter 16. z/OS Provisioning Toolkit
And z/OSMF composite templates . . . 123**

**Chapter 17. Automating provisioning
with z/OS Provisioning Toolkit 127**

**Chapter 18. Troubleshooting z/OS
Provisioning Toolkit 129**

**Chapter 19. The zospt command
syntax 135**

**Chapter 20. z/OS Provisioning Toolkit
supplied templates 139**

**Chapter 21. Samples provided with
z/OS Provisioning Toolkit 141**

**Chapter 22. Provisioning a CICS
region through IBM Cloud
Provisioning and Management for
z/OS 147**

Configuration properties for CICS templates . . . 148

**Chapter 23. Provisioning an instance
of z/OS Connect Enterprise Edition
through IBM Cloud Provisioning and
Management for z/OS 165**

Configuration properties for z/OS Connect

Enterprise Edition templates 166

Notices 171

About this book

This PDF is the product documentation for IBM® z/OS® Provisioning Toolkit. The same documentation is provided in IBM Knowledge Center.

Date of this PDF

This PDF was created on July 1 2019.

Chapter 1. What is z/OS Provisioning Toolkit?

z/OS Provisioning Toolkit (z/OS PT) is a command line utility for the rapid provisioning of development runtime environments. System programmers can set up the environments that can be provisioned in this way, authorize access to individuals or teams, and set provisioning controls and limits. Application developers can use a command line interface to provision and deprovision z/OS development environments quickly and independently, without needing mainframe administration skills or authority.

DevOps tools and processes allow development and operations teams to collaborate on reducing time to delivery. Automation is critical to an efficient development process. z/OS PT helps to automate the development process by providing application developers with immediate access to the environments that they need, while the operations team retains overall command and control of the provisioning process.

z/OS Provisioning Toolkit has three main elements:

Command line utility

z/OS PT provides commands for an application developer to build an application image, run that image, provision the underlying middleware environment, and, later, deprovision that environment. Commands provided include:

- **zospt build** Build an application image.
- **zospt run** Run an application image and automatically provision and start the associated z/OS middleware.
- **zospt ps** List environments that are provisioned.
- **zospt rm** Deprovision an environment.

zosptfiles and associated images

zosptfile text files define the environments that can be provisioned. z/OS PT provides sample files and images to make it quicker to set up the foundation of commonly used environments such as CICS® or IBM z/OS Connect Enterprise Edition.

z/OSMF workflows

The toolkit uses two features of z/OS to enforce control over provisioned environments. You can define workflows in IBM z/OS Management Facility (z/OSMF) that can provision runtime environments. You can manage and control these workflows by using IBM Cloud Provisioning and Management for z/OS. The workflows can be customized to local naming standards and operating procedures. z/OS PT drives these workflows.

- For a list of workflows that are supplied with z/OS PT, see Chapter 20, “z/OS Provisioning Toolkit supplied templates,” on page 139.
- For an overview of provisioning and management of z/OS middleware and how z/OSMF fits into the solution, see IBM Cloud Provisioning and Management for z/OS: An Introduction.

To go a little deeper into the architecture and components, see “How does z/OS Provisioning Toolkit work?” on page 3.

Build, run, manage

Systems programmers can define software templates in z/OSMF. These templates specify the z/OSMF workflows, and definitions to be used by the workflows, to provision middleware systems. The workflows can be configured to support local naming standards and operating procedures. Appropriate provisioning controls and limits can be set by using IBM Cloud Provisioning and Management for z/OS, including how many environments can be provisioned through z/OS PT. An application developer user ID, or a group of user IDs, can be granted access to run these templates within defined controls and limits.

After the operations team configures the z/OSMF templates, application developers can use the z/OS PT command line utility to issue simple commands to provision z/OS environments such as a CICS region or an instance of z/OS Connect Enterprise Edition. Application developers can share a single installation of z/OS PT, including images.

z/OS PT provisioning support

z/OS PT supports provisioning of a range of environments, by providing example scripts and images and z/OSMF workflows. The flexible nature of a z/OSMF workflow means that more tools and third-party products that the environments need can be automatically configured and provisioned at the same time.

z/OS PT include images to provision the following environments:

- CICS Transaction Server, with embedded IBM z/OS Connect Enterprise Edition
- CICS Transaction Server, with a Liberty JVM server
- A standard CICS TS environment for traditional COBOL 3270 development
- z/OS Connect Enterprise Edition V2.0 stand-alone server
- z/OS Connect Enterprise Edition V3.0 stand-alone server

Try z/OS PT for yourself

Download z/OS PT without charge from IBM z/OS Provisioning Toolkit product page.

z/OS PT provides a getting started scenario that enables CICS system programmers to quickly evaluate the CICS workflows that are provided for provisioning and deprovisioning a CICS region before they try z/OS PT with Cloud Provisioning. For more information, see “Getting started with CICS by using z/OSMF workflows” on page 30.

How does z/OS Provisioning Toolkit work?

z/OS PT interacts with z/OSMF to provision containers that are ready for use, by using images. z/OS PT has many artifacts and components, and these components are used in a certain sequence.

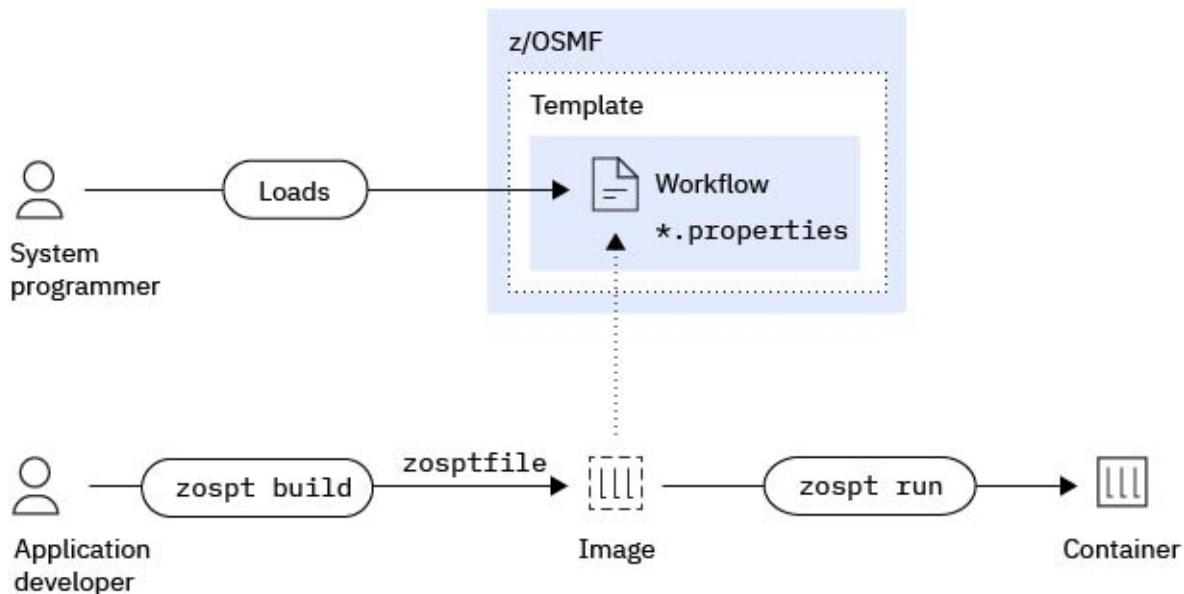


Figure 1. Introducing the z/OS Provisioning Toolkit

The components of z/OS PT

z/OSMF

IBM z/OS Management Facility, with the IBM Cloud Provisioning and Management for z/OS capability and z/OSMF workflows, are the tools that z/OS PT uses to provision containers.

Workflow

z/OSMF includes a Workflow Engine. A workflow in z/OSMF is an XML document that consists of one or more steps, where each step typically makes a REST API call or runs JCL. Workflows are used to perform a specific task. The z/OSMF workflows can be customized to local naming standards and operating procedures by editing a `*.properties` file.

A workflow step can:

- Submit JCL jobs and check the result of the job.
- Run a REXX script.
- Run a Shell script.
- Call a REST service.

`*.properties` file

This text file sets values for the variables that are used by the z/OSMF workflow, for example, the configuration options that you want to set in a provisioned CICS region.

Template

The template is an artifact in z/OSMF. The template describes how to provision a middleware instance that uses a z/OSMF workflow, and describes any definitions to be associated with that workflow. z/OS PT provides templates for provisioning IBM middleware such as CICS and IBM MQ. Templates must be approved, usually by a system programmer.

zosptfile text file

This file specifies what to include in an image when z/OS PT builds an image, and associates the image (directly or indirectly) with a template. See “Building an image with z/OS Provisioning Toolkit” on page 49.

Image An image is a preconfigured environment that can be provisioned rapidly to create one or more runtime environments. An image is a binary object that encapsulates a set of files and configuration, as defined by the `zosptfile`. Each image is associated with a z/OSMF template. The set of files and configuration in the image is used, with the associated z/OSMF template, to provision one or more containers with the same capabilities. An image can have a dependency on another image - a base image - that provides the association with a z/OSMF template. Images can provide flexibility without requiring reapproval of the template. For more information about the value of images, see Chapter 11, “z/OS Provisioning Toolkit images,” on page 47.

Container

The result of provisioning with z/OS PT is a container, with all the capabilities of an associated image. The container encapsulates an application and its dependencies, making an environment that is ready for you to work in.

The z/OSMF product documentation refers generally to software provisioned from a template as an *instance*.

The flow of z/OS PT

z/OS PT provides a getting started scenario that enables CICS system programmers to quickly evaluate the CICS workflows that are provided for provisioning and deprovisioning a CICS region before they try z/OS PT with Cloud Provisioning. For more information, see “Getting started with CICS by using z/OSMF workflows” on page 30.

For a full, dynamic scenario, a system programmer defines a template in z/OSMF, associates it with a workflow and specifies (in a `*.properties` file) any values to be used for variables in the workflow.

In z/OS, application developers use the **zospt build** command to build an image. They specify a configuration file that is called a `zosptfile` to associate the image with a z/OSMF template and provide any additional configuration. Then, they use the **zospt run** command to create a container from the image. They each can work in a container, and manage its lifecycle with z/OS PT commands to start, stop, and deprovision.

What to do next?

To start to explore z/OS PT, go to Chapter 4, “Installing z/OS Provisioning Toolkit,” on page 19. To understand more about how z/OSMF works, see z/OSMF product documentation.

What documentation is available for z/OS Provisioning Toolkit?

z/OS PT provides product documentation (in IBM Knowledge Center and as PDF), and other sources of information. Command help is provided in z/OS PT itself and a readme file is included in the z/OS PT download package. You can also find articles and examples in Developer Centers and tips in dW Answers.

Product documentation in IBM Knowledge Center and PDF

If you are reading this document in PDF, you can find z/OS PT documentation in IBM Knowledge Center at z/OS Provisioning Toolkit in IBM Knowledge Center. This documentation is updated with every release of z/OS PT, and might occasionally be updated between releases to address issues.

z/OSMF product documentation is also available in IBM Knowledge Center.

If you are reading this document in IBM Knowledge Center, you can find a PDF file of documentation for download alongside the z/OS PT package. This PDF contains the same documentation that you find in IBM Knowledge Center, in a different format.

Readme file and command help

z/OS PT contains command help. To access all the command help, type `zospt --help`. For help about a specific command, type `zospt command --help`.

A readme file is provided in the download package with initial instructions and links to the documentation that you need to work with z/OS PT.

Other sources of information

Developer Centers contain articles and examples to extend the documentation about z/OS PT:

- IBM z/OS Provisioning Toolkit product page gives you links to the latest news and to download z/OS PT.
- Mainframe Developer Center includes news about z/OS PT as part of the range of IBM products that support DevOps.
- CICS Developer Center covers the role of z/OS PT in provisioning and managing CICS environments.
- z/OS Connect Enterprise Edition Developer Center includes information about provisioning z/OS Connect EE instances by using z/OS PT.
- IBM Developer Answers is an open forum for questions and answers about IBM products. Questions about z/OS PT are tagged with **zospt**.

Tell the z/OS PT team what you think

The z/OS PT team welcome your comments on the z/OS PT product documentation. The documentation is refreshed regularly in response to feedback, so get in touch through the **Contact us** option in IBM Knowledge Center, or in Mainframe Developer Center.

Conventions used in this documentation

The following conventions are used to document z/OS PT syntax:

Notation	Explanation
zospt	A command. Square brackets indicate an optional parameter, for example: [-option]
<i>variable</i>	Variables appear in lowercase. They represent names or values that you supply.

Accessibility

Documentation that is provided with z/OS PT is accessible. Information about the accessibility features of IBM Knowledge Center is in the IBM Knowledge Center Release Notes. In addition to the accessibility of the online framework, the content of the z/OS PT documentation itself is adapted to be accessible; all images that convey additional information have alternative text.

Chapter 2. z/OS Provisioning Toolkit change history

Use this information to discover the functions and properties that are added or altered, in each version, release, or modification of IBM z/OS Provisioning Toolkit.

What's new in version 1.1.5

Release date: May 15 2019

Table 1. Function and property updates in version 1.1.5

Update	Description
New function	<ul style="list-style-type: none">• Provision an IMS DB/TM V13 or V14 system. See “Getting started with other middleware” on page 39.• Simplified configuration through the option to provision a CICS region by using IBM Cloud Provisioning and Management for z/OS without needing to configure or use the z/OSMF z/OS console services. This function depends on the value of the CONSOLE_COMMAND_TYPE configuration property. See “Workflow configuration variables” on page 89.
New configuration properties for CICS	<ul style="list-style-type: none">• CONSOLE_COMMAND_TYPE sets how the provisioning process performs tasks that include detecting successful start and stop of a CICS region and deleting log streams. Valid options are SCRIPT (uses REXX scripts to issue console commands) or APICALL (use the API that the z/OSMF z/OS console services provide). If you specify SCRIPT, you do not need to configure z/OS console services. <p>See “Configuration properties for CICS images” on page 75.</p>
Updated configuration properties for CICS	<ul style="list-style-type: none">• DFH_REGION_LOGSTREAM_HLQ is changed to support the following format for the log stream prefix: <i>DFH_REGION_APPLID.*</i>• The GRPLIST parameter in DFH_REGION_SITPARMS is changed to allow an increased maximum of three entries and to no longer require the inclusion of the CICS groups typically in DFHLIST. The CICS groups typically in DFHLIST are now automatically included in the generated list of provisioning-specific resource definitions. This generated list is now included as the first entry in the group list so that the provisioning-specific resource definitions are installed before the resources specified with the GRPLIST parameter. <p>See “Configuration properties for CICS images” on page 75.</p>
Updated sample templates	The sample templates to provision a CICS region are updated to include the new CONSOLE_COMMAND_TYPE=SCRIPT configuration property.
Updated JES job names	The names of the JES jobs that are used for steps in z/OS PT processes for provisioning and managing CICS regions are changed to ensure that job names start with a valid character. You might need to ensure that these job names meet the security requirements for your system. See “Job entry subsystem (JES) job names” on page 94.

What's new in version 1.1.4

Release date: 31 January 2019

Table 2. Function and property updates in version 1.1.4

Update	Description
New function	<ul style="list-style-type: none">• Support for provisioning CICS Transaction Server for z/OS V5.5 regions.• Support for provisioning a CICS region that obtains dynamically allocated ports from different port ranges, depending on the use of the requested port. See “Using dynamic port allocation in a template” on page 64.
New and updated sample templates	<ul style="list-style-type: none">• The following new templates are provided:<ul style="list-style-type: none">– cics_55 to provision a CICS Transaction Server for z/OS V5.5 region.– cmas_55 to provision a CICSplex® SM address space (CMAS) V5.5 region.– wui_55 to provision a Web User Interface (WUI) V5.5 region. <p>See Chapter 21, “Samples provided with z/OS Provisioning Toolkit,” on page 141.</p>
New and updated sample images	<ul style="list-style-type: none">• The cics_55_nodejs_ivp image is renamed from cics_55_open_beta_nodejs_ivp.• The cics_async_api_sample is renamed from cics_54_async_api_sample and updated to enable provisioning of a CICS Transaction Server for z/OS V5.4 or later region.• The following new images are provided to provision and, where relevant, extend a CICS Transaction Server for z/OS V5.5 region:<ul style="list-style-type: none">– cics_55– cics_55_json– cics_55_liberty– cics_55_soap– cics_55_zosconnect_v3r0– cmas_55– wui_55 <p>Corresponding sample zosptfile files are also updated.</p> <p>See “Sample images and the source artifacts that are used to build them” on page 142.</p>

Table 2. Function and property updates in version 1.1.4 (continued)

Update	Description
Removed sample templates, images and zosptfile files	<ul style="list-style-type: none"> The following templates are removed: <ul style="list-style-type: none"> – cmas_54 – wui_54 The following images and corresponding zosptfile files are removed: <ul style="list-style-type: none"> – cics_54 – cics_54_async_api_sample – cics_54_json – cics_54_liberty – cics_54_soap – cics_54_zosconnect_v3r0 – cmas_54 – wui_54
New configuration properties for CICS	<ul style="list-style-type: none"> DFH_JVM_DEBUG_USAGETYPE to set the usage type for the port that a Java™ debugger can be connected to. DFH_REGION_CMCIPTORT_USAGETYPE to set the usage type for a CMCI port. DFH_REGION_HTTP_USAGETYPE to set the usage type for an HTTP port. DFH_REGION_HTTPS_USAGETYPE to set the usage type for an HTTPS port. DFH_REGION_IPIC_USAGETYPE to set the usage type for an IPIC port. DFH_REGION_TCPIP_USAGETYPE to set the usage type for the CICSplex SM WUI port. <p>Ports can then be dynamically allocated from the port allocation range that each usage type identifies.</p> <p>See “Configuration properties for CICS images” on page 75.</p>
Updated configuration properties for CICS	<ul style="list-style-type: none"> DFH_ZOS_VSAM_VOLUME is changed to specify that the volume parameter is used for PDS creation as well as VSAM. If you specify SMS, the VOLUME parameter is omitted from PDS creation (for example trace and dump) as well as VSAM. <p>See “Configuration properties for CICS images” on page 75.</p>

What's new in version 1.1.3

Release date: 15 November 2018

Table 3. Function and property updates in version 1.1.3

Update	Description
New function	<ul style="list-style-type: none">Faster provisioning and deprovisioning for CICS templates that use Cloud Provisioning to provision a CICS region. You can enable parallel processing in the workflows for CICS templates that use Cloud Provisioning so that provisioning and deprovisioning complete faster. Parallel processing is supported with the zospt run and zospt rm commands and the relevant templates. This function requires the PTFs for APARs PI92305 and PI99364 applied to z/OSMF V2R3, or the PTFs for APARs PI95358 and PI99364 applied to z/OSMF V2R2. See “Enabling faster provisioning by using parallel processing” on page 63.Support to include program binary files, such as compiled COBOL or PL/I, in an image. You can use this support to provision a CICS region with a LIBRARY resource installed in it that references a library that contains those program binary files. See “Including program binary files in your image” on page 69.Support to include CICS resource definitions in an image by providing input files for the CICS system definition utility program DFHCSDUP in an image. You can use this to provision a CICS region with the resource definitions defined in its system definition data set (CSD). See “Including CICS resource definitions in an image” on page 70.
New sample image	<ul style="list-style-type: none">The <code>cics_54_async_api_sample</code> sample image provides an example of including program binary files and CICS resource definitions in an image. The image provisions a CICS Transaction Server for z/OS V5.4 region that runs a sample CICS Asynchronous API application. See Chapter 21, “Samples provided with z/OS Provisioning Toolkit,” on page 141.
Updated JES job names	The names of the JES jobs that are used for steps in z/OS PT processes for provisioning and managing CICS regions are changed so that each step uses a unique job name. You might need to ensure that these job names meet the security requirements for your system. See “Job entry subsystem (JES) job names” on page 94.

What's new in version 1.1.2

Release date: 31 July 2018

Table 4. Function and property updates in version 1.1.2

Update	Description
New function	<ul style="list-style-type: none">Support for Node.js applications in a CICS Transaction Server for z/OS V5.5 region.Support to use existing CICS log streams during provisioning.Support to use dummy CICS log streams during provisioning.

Table 4. Function and property updates in version 1.1.2 (continued)

Update	Description
New and updated configuration properties for CICS	<ul style="list-style-type: none"> DFH_REGION_LOGSTREAM indicates how the log streams that the provisioned CICS region requires are created. Valid options are DYNAMIC (the default), EXISTING, and DUMMY. <p>See “Configuration properties for CICS images” on page 75.</p>
Removed configuration properties for CICS	<ul style="list-style-type: none"> TEMP_DIR is no longer required and is ignored if a value is provided.

What's new in version 1.1.1

Release date: 27 April 2018

Table 5. Function updates in version 1.1.1.

Update	Description
New function	<ul style="list-style-type: none"> New getting started scenarios, with new samples and workflows to provision CICS with minimal prerequisites to get started with provisioning. For more information about getting started, see Chapter 8, “Getting started with z/OS Provisioning Toolkit,” on page 29. Support for starting and stopping the Composite_Parent container by using the zospt start and zospt stop commands. Starting or stopping the Composite_Parent container starts or stops the Composite_Child containers in the correct order.

What's new in version 1.1

Release date: 26 January 2018

Table 6. Function and property updates in version 1.1

Update	Description
New function	<ul style="list-style-type: none"> Support for provisioning additional CICS region types: CICSplex SM address spaces (CMASs) and Web User Interface (WUI) regions. See “Overview of supported CICS types” on page 58. Support to enable automation of provisioning and deprovisioning containers through the new --quiet option on the zospt run command. See Chapter 17, “Automating provisioning with z/OS Provisioning Toolkit,” on page 127. Support to build, run, and manage containers that are provisioned from composite templates that are created in z/OSMF 2.2 and later. See Chapter 16, “z/OS Provisioning Toolkit And z/OSMF composite templates,” on page 123. Support for starting and stopping multiple containers by using the zospt start and zospt stop commands. Support for specifying the <i>containerName</i> of a provisioned container on the zospt run command. Support for allowing a subset of configuration properties that are set in the template to be overridden in the z/OS PT image. For more information about overriding CICS properties, see “Modifying the set of CICS configuration properties that can be overridden” on page 93.

Table 6. Function and property updates in version 1.1 (continued)

Update	Description
New and updated configuration properties for CICS	<ul style="list-style-type: none"> • DFH_CICS_TYPE now supports new values CMAS and WUI (in addition to the current values of MAS, SMSS and Unmanaged). See “Overview of supported CICS types” on page 58. • DFH_DELETE_LOGSTREAM_TIMEOUT customizes how long a workflow waits to detect successful deletion of the CICS log streams. • DFH_ZOS_PROCLIB now specifies the complete data set name of a procedure library (PROCLIB). The workflow no longer appends .PROCLIB to the end of this value. • DFH_ZOS_STCJOBS now specifies the complete data set name of a data set into which started task JCL jobs can be created. The workflow no longer appends .STCJOBS to the end of this value. <p>See “Configuration properties for CICS images” on page 75.</p>
Removed configuration properties for CICS	<ul style="list-style-type: none"> • DFH_CONSOLE_SEARCH_SPACES is no longer required and is ignored if a value is provided.

What's new in version 1.0.3

Release date: 10 October 2017

Table 7. Function and property updates in version 1.0.3

Update	Description
New function	<ul style="list-style-type: none"> • Link a provisioned CICS image to a registered Db2[®] container, a registered IBM MQ container, or both. • Link a provisioned z/OS Connect EE image to a provisioned or registered CICS container. <p>See “Link scenarios supported by z/OS Provisioning Toolkit” on page 54.</p> <ul style="list-style-type: none"> • Register existing subsystems in z/OSMF. See Chapter 15, “Registering existing subsystems in z/OSMF,” on page 117. • Provision a CICS region from an image that includes a CICS bundle. See “Including a CICS bundle in your image” on page 67. • Provision a CICS region with a JVM that is configured for remote debug. • Support for filtering the list of containers in the zospt ps command. • Support for removing images by using the zospt rmi command. • Support for removing multiple containers by using the zospt rm command. • Define a job in an STCJOBS data set for the started task of the provisioned CICS region. • Define a job in an STCJOBS data set for the started task of the provisioned instance of z/OS Connect EE. • Enhanced validation of the user ID that is specified for the DFH_ADMIN_CONSOLE property to determine whether the z/OS console services (an API) can be used to detect unsolicited messages.

Table 7. Function and property updates in version 1.0.3 (continued)

Update	Description
New and updated configuration properties for CICS	<ul style="list-style-type: none"> • DFH_JVM_DEBUG • DFH_REGION_LOGSTREAM_HLQ • DFH_STC_JOB_CARD • DFH_ZFS_DATACLASS • DFH_ZFS_GROUP • DFH_ZOS_STCJOBS <p>See “Configuration properties for CICS images” on page 75.</p>
New and updated configuration properties for z/OS Connect EE	<ul style="list-style-type: none"> • ZCON_STC_JOB_CARD • ZCON_ZOS_STCJOBS <p>See “Configuration properties for z/OS Connect Enterprise Edition images” on page 106.</p>

What's new in version 1.0.2

Release date 17 July 2017

Table 8. Function and property updates in version 1.0.2

Version number	Function and properties updates
New function	<ul style="list-style-type: none"> • Provision a CICS region with a shared CSD by using supplied workflows. See “Shared CSD guidance” on page 91. • CICS samples are updated to support CICS Transaction Server for z/OS V5.4. • Support for provisioning z/OS Connect Enterprise Edition V3.0 by using supplied workflows. • Provision a CICS region with an IPIC port (for example to enable a stand-alone instance of z/OS Connect EE V3.0 to connect to a provisioned CICS region by using the new CICS service provider). • Provision z/OS software subsystems and services on any LPAR in a Parallel Sysplex®. • The z/OS PT documentation PDF is now available directly from this IBM Knowledge Center. See z/OS Provisioning Toolkit PDF.
New configuration properties for IBM z/OS Provisioning Toolkit	<ul style="list-style-type: none"> • New zospt_pw environment variable can be set to enable z/OS PT to be used in a script without prompting for a password. • New logging.properties file for configuring z/OS PT log size and count that contains the following properties: <ul style="list-style-type: none"> – java.util.logging.FileHandler.limit – java.util.logging.FileHandler.count <p>See Chapter 6, “Configuring z/OS Provisioning Toolkit,” on page 23.</p>

Table 8. Function and property updates in version 1.0.2 (continued)

Version number	Function and properties updates
New and updated configuration properties for CICS	<ul style="list-style-type: none"> • DFH_REGION_CSD is renamed DFH_REGION_CSD_INPUT • DFH_REGION_CSD_TYPE • DFH_REGION_IPIC • DFH_REGION_SHARED_CSD_NAME • TEMP_DIR • VALIDATE_PARAMETERS <p>See “Configuration properties for CICS images” on page 75.</p>
New configuration properties for z/OS Connect EE	<ul style="list-style-type: none"> • TEMP_DIR • VALIDATE_PARAMETERS <p>See “Configuration properties for z/OS Connect Enterprise Edition images” on page 106.</p>

What's new in version 1.0.1

Release date 11 April 2017

Table 9. Function and property updates in version 1.0.1

Update	Description
New function	<ul style="list-style-type: none"> • Provision stand-alone z/OS Connect EE servers by using supplied workflows. • Provision Liberty servers. For more information, and for sample workflows, see WebSphere Liberty workflow in GitHub. • Enhanced zospt inspect command to get information on provisioned containers. See Chapter 19, “The zospt command syntax,” on page 135. • Truststore support for connecting to z/OSMF from the command line utility.
New configuration properties for IBM z/OS Provisioning Toolkit	<ul style="list-style-type: none"> • Added to the <code>zosmf.properties</code> file: <ul style="list-style-type: none"> – truststore – truststore_password <p>See Chapter 6, “Configuring z/OS Provisioning Toolkit,” on page 23.</p>
New configuration properties for CICS	<ul style="list-style-type: none"> • DFH_CONSOLE_SEARCH_SPACES • DFH_START_TIMEOUT • DFH_STOP_TIMEOUT <p>See “Configuration properties for CICS images” on page 75.</p>

What's new in version 1.0

Release date: 10 January 2017

Table 10. New function in version 1.0

	Description
New function	<ul style="list-style-type: none">• z/OS PT is a simple command line utility for the rapid provisioning of z/OS development environments. Use z/OS PT to build an image, run an image to create a container, and then manage that container by using simple commands.• z/OS PT supports the following commands:<ul style="list-style-type: none">– zospt build– zospt run– zospt rm– zospt inspect– zospt images– zospt ps• Provision a CICS region by using supplied workflows.• Provision an IBM MQ queue manager (workflows are included with IBM MQ V9.0.1 for z/OS).• Examples for different CICS applications, including provisioning embedded z/OS Connect EE, Liberty, and web services.

Chapter 3. Prerequisites for using z/OS Provisioning Toolkit

Before you use z/OS PT, ensure that your system has the correct configuration. For z/OS PT prerequisites, and any subsystem-specific requirements, use the links that are provided here.

For system requirements to use z/OS PT, see the z/OS Provisioning Toolkit Software Compatibility Report.

Prerequisites for getting started by using z/OSMF workflows

Before you can provision a container by using z/OSMF workflows, complete the following checks:

- Ensure that z/OSMF is installed and configured. See Configuration in the z/OSMF Configuration Guide.
z/OSMF performs a system management role, and is accessed through a web interface. z/OSMF is a single point of control that allows you to:
 - View, define, and update policies that affect system behavior.
 - Monitor the performance of the systems in your enterprise.
 - Manage your z/OS software.
 - Perform data management tasks.
 - Consolidate your z/OS management tools.
 - Run workflows to help you perform a common activity on z/OS.
- Ensure that your user ID has the appropriate authority to run workflows, by granting it the following access:
 - READ access to the ZMFAPLA class *SAF-prefixZOSMF.WORKFLOW.WORKFLOWS*.
 - READ access to EJBROLE *SAF-prefixIzuManagementFacilityWorkflow.izuUsers*

For more information, see Help with security setup in IBM Z solutions product documentation.

- Install z/OS PT by following the instructions here: Chapter 4, “Installing z/OS Provisioning Toolkit,” on page 19.

Before you can provision a container, ensure that your user ID has the following authority:

- Authority to run an instance of the subsystem that you are provisioning.
- Sufficient authority in a test environment for the following tasks:
 - Allocating data sets.
 - Creating and deleting log streams.
 - Adding a procedure into a PROCLIB.
 - Issuing console commands.

These prerequisites enable you to get started and provision a simple single CICS container. For more information, see “Getting started with CICS by using z/OSMF workflows” on page 30.

Dynamic Cloud Provisioning prerequisites

To get started with provisioning other subsystems, or to provision a CICS region dynamically (by allocating ports and APPLIDs from a pool of resources), see Chapter 9, “Preparing to use z/OS Provisioning Toolkit with IBM Cloud Provisioning and Management for z/OS,” on page 41.

Chapter 4. Installing z/OS Provisioning Toolkit

z/OS PT is packaged as a compressed file that you can download without charge. Install z/OS PT, ready for subsequent configuration.

Before you begin

Before you can install z/OS PT, ensure that you meet the system requirements that are listed in Prerequisites for using z/OS Provisioning Toolkit.

Procedure

1. Find z/OS PT for download on IBM z/OS Provisioning Toolkit product page.
2. Extract the .zip file.
3. Transfer the .pax file in binary format to the system where you want to run z/OS PT.

4. Issue the following command to extract the .pax file:

```
pax -rf zospt_v#####.pax
```

The zospt installation directory is created.

5. Ensure that the user ID under which z/OSMF runs (the default user ID is IZUSVR), has both read and execute permissions for all directories and subdirectories in the zospt path. To change the permissions, issue this command:

```
chmod -R 755 zospt_install_dir
```

6. Ensure all user IDs under which you run **zospt** have the necessary permissions:

- Read and run permissions for all directories in the zospt path.
- Read, write, and run permissions for the images and logs directories in the zospt path.

7. Add zospt to the PATH. For quick access to the toolkit, issue the following command, where *zospt_directory* is the full path of the z/OS PT installation directory:

```
export PATH=$PATH:zospt_directory/bin
```

To persist the PATH declaration between sessions, add this command to your .profile file on z/OS UNIX System Services. For more information, see Customizing your .profile in the z/OS UNIX System Services User's Guide.

8. Optional: If you do not want to use the default domain and tenant in z/OSMF, add the following environment variables:

```
zospt_domain=zospt_domain_name  
zospt_tenant=zospt_tenant_name
```

One way to do this is to update your .profile in z/OS UNIX System Services. For further details, see Chapter 6, “Configuring z/OS Provisioning Toolkit,” on page 23.

9. Enter the following command from the zospt directory:

```
zospt --help
```

If the installation is successful, the command help prints to the command line.

Results

You successfully installed z/OS PT.

What to do next

You can now:

- Return to “Getting started with CICS by using z/OSMF workflows” on page 30 and complete the scenario.
- Prepare z/OSMF and z/OS PT to work together, by following the instructions in Chapter 6, “Configuring z/OS Provisioning Toolkit,” on page 23.

Chapter 5. Upgrading z/OS Provisioning Toolkit

Find out more about how to upgrade from one version of z/OS Provisioning Toolkit to another.

Before you begin

Check the prerequisites of the new version of z/OS PT and any subsystems that you use with z/OS PT. See Prerequisites for using z/OS Provisioning Toolkit.

Procedure

1. Install the latest version of z/OS PT to a unique directory location. Use a unique location to ensure that you do not overwrite any template properties files, or other customizations you made to your environment, including modifications to the dynamic security scripts, and any custom images you built. Follow the installation instructions here Chapter 4, “Installing z/OS Provisioning Toolkit,” on page 19. Return to this upgrading procedure when you complete the installation instructions.
2. Examine any XML files, properties files, or REXX files that you modified for your previous version of z/OS PT. You might need to merge or copy these changes into the new installation directory. Specifically, if you modified files in the extensions directory for the CICS or z/OS Connect EE workflows, you might need to copy those changes to the new installation directory, or merge those changes with updated versions of those files in the new installation directory.
3. Check for details of any new or updated properties that you might need to include in your properties files to use the new templates by referring to Chapter 2, “z/OS Provisioning Toolkit change history,” on page 7.
4. Deprovision any systems that are provisioned from product-specific z/OSMF templates in Draft approved state before you refresh the templates to the new version.
5. Refresh or load a new version of any product-specific templates from your previous version of z/OS PT. For more information about how to load a z/OSMF template for use with z/OS PT, see Chapter 10, “Adding an IBM Cloud Provisioning and Management for z/OS template for use with z/OS Provisioning Toolkit,” on page 45.
6. Change the PATH to point to the new version of z/OS PT.
7. Rebuild any custom images with the new version of z/OS PT. For more information about building images, see “Building an image with z/OS Provisioning Toolkit” on page 49. Reflect any changes that are made to the sample `zosptfile` text files in any `zosptfile` text files you have that are based on the samples.

What to do next

The new version of z/OS PT is ready to use. For more information about upgrading if you are using z/OS PT to provision CICS regions, see “Configuration prerequisites for using z/OS Provisioning Toolkit with CICS” on page 57.

Chapter 6. Configuring z/OS Provisioning Toolkit

To connect z/OS PT to z/OSMF, you must configure z/OS PT with basic connectivity information for the z/OSMF server.

Procedure

1. Edit the z/OS PT `zosmf.properties` UTF-8 file in the `/zospt/config` directory. The `zosmf.properties` file contains properties that are used to connect to z/OSMF:

hostname

The host name, or IP address, of the system where z/OSMF is installed. The default is `localhost`.

port

The HTTPS port that z/OSMF listens on. Ask your z/OSMF administrator for this information.

truststore = YES|NO

Using a truststore enables z/OS PT to check that it trusts the certificate of the z/OSMF server to which it attempts to connect.

- The default of `truststore=NO` means that when z/OS PT attempts to connect to z/OSMF, z/OS PT always trusts the certificate that is presented by the z/OSMF server during the SSL handshake.
- Specifying `truststore=YES` means that z/OS PT checks that the certificate (or a certificate authority that signed the z/OSMF server's certificate) is present in its truststore. If the certificate is invalid, or is not trusted, no connection is made.

Note: z/OS PT does not verify that the host name of the z/OSMF server matches the host name that is identified in the z/OSMF server's certificate.

A preconfigured truststore can be provided to z/OS PT. The truststore must be of type JKS (Java keystore) and must be provided at the following location: `zospt/config/zosmf.jks`. If no pre-configured truststore is available, z/OS PT will ask the user whether to trust the z/OSMF server's certificate and will create a truststore for future use. The user must review the certificate's details and determine whether it is acceptable for z/OS PT to trust the z/OSMF server. If the user agrees, the certificate that identifies the root certificate authority that signed the z/OSMF servers certificate (or the z/OSMF servers certificate if it is self-signed) is added to the truststore.

truststore_password

Specifies the password that is used to access the truststore.

systemNickname

Specifies the nickname of the system. The nickname is used only when you provision a container by using z/OSMF workflows, as described in "Getting started with CICS by using z/OSMF workflows" on page 30, and is ignored by IBM Cloud Provisioning and Management for z/OS. Ask your system programmer for this information, or, if your user ID is in the security group for the z/OSMF users (IZUUSER, by default), you can see a list of system nicknames by running z/OS PT command **zospt ps -a**.

2. Check that z/OS PT connects to z/OSMF. Issue **zospt ps --all**. z/OS PT prompts for a password and then reports back basic information about the z/OSMF server.
3. Optional: Set z/OS PT environment variables to further refine the connection to z/OSMF.

You can set the following environment variables in the `.profile` in z/OS UNIX System Services, or as part of a shell script that runs z/OS PT:

zospt_domain

Specifies the z/OSMF domain with which to connect to run a template. If not specified, the default domain is assumed.

zospt_tenant

Specifies the z/OSMF tenant with which to connect to run a template. If not specified, the default domain is assumed.

You can set the **zospt_pw** environment variable to the password of the user ID that runs z/OS PT but this is not generally recommended unless the profile is protected from read access for all users. If this environment variable is not set, the password is prompted for when z/OS PT is run and needs to connect to z/OSMF.

4. Optional: Edit the z/OS PT `logging.properties` UTF-8 file in the `/zospt/config` directory. The `logging.properties` file controls the size and number of log files that are created by z/OS PT. It is only necessary to change these parameters if the default logging configuration is unsuitable. The `logging.properties` file contains properties that control the z/OS PT log size and count:

java.util.logging.FileHandler.limit

Specifies an approximate maximum amount to write (in bytes) to any one file. The valid range is 0 - 2147483647 bytes. If it is set to zero, the number of bytes that can be written has no limit. The default is 1000000 bytes.

java.util.logging.FileHandler.count

Specifies the number of log files to cycle through per user. The valid range is 0 - 2147483647 bytes. The default is 20. The logs are in the `/logs` directory. The most recent log file is named `log_<user ID running zospt>.0`.

What to do next

Load a template into z/OSMF ready for use by the **zospt** commands, by following the instructions in Chapter 10, “Adding an IBM Cloud Provisioning and Management for z/OS template for use with z/OS Provisioning Toolkit,” on page 45.

Chapter 7. Security planning for z/OS Provisioning Toolkit with IBM Cloud Provisioning and Management for z/OS

The security model for z/OS PT (with the use of Cloud Provisioning) provides complete control over the provisioning process to ensure that only authorized users can provision environments.

Roles and authorities for setting up z/OSMF and IBM Cloud Provisioning and Management for z/OS

z/OSMF has a security model with a number of roles that can perform different tasks to enable Cloud Provisioning.

Table 11. z/OSMF Security roles

Role	Function
Network administrator	Defines the systems, TCP/IP stack, and pools of resources for Cloud Provisioning. These pools include ports, APPLIDs, and IP addresses. These pools are used by tenants and must be in place before a template can be run.
Security administrator	Configures the security for all the IDs that are needed to complete the setup and running of Cloud Provisioning.
Landlord	Defines the domain, domain administrators, and the system resources for the domain.
Domain administrator	Defines tenants, and associates templates with the tenants. Adds consumers to the tenants.
Provisioning approver	Has the authority to approve the IDs that are used in the workflows to provision environments.
Consumer	Has the authority to run templates in a tenant to provision environments. Consumers do not usually need a high level of authority because the templates use provisioning IDs with the authority to create an environment from scratch.

To work with z/OS PT, you need the following authorities:

- Domain administrator authority to upload templates.
- Consumer authority in z/OSMF to run templates.
- All provisioning IDs and approval IDs specified in the templates to be authorized by the provisioning approver.

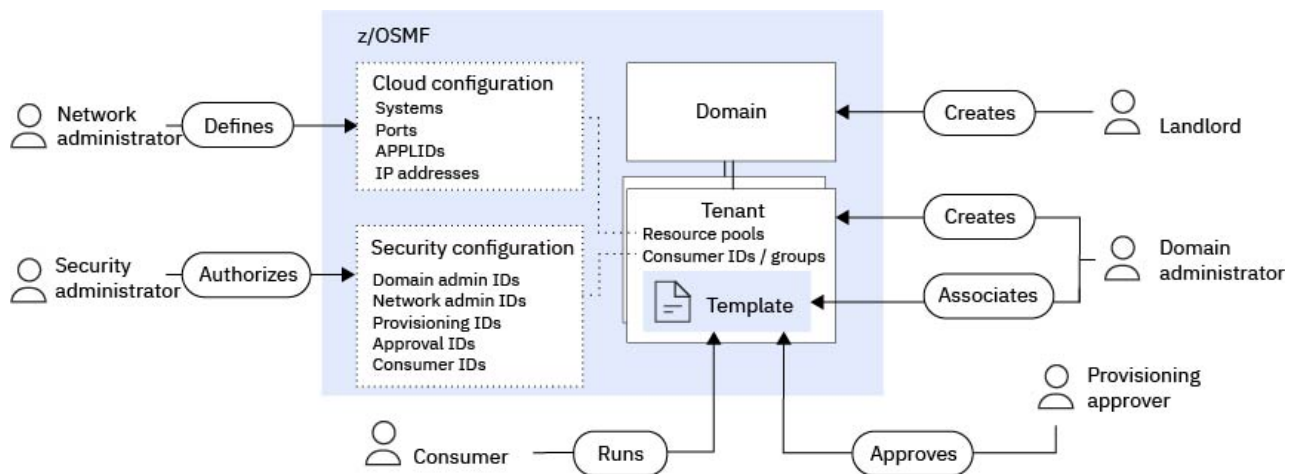


Figure 2. Roles and authorities for setting up z/OSMF and Cloud Provisioning

Authority to run the z/OS PT command line interface

The command line interface is available on UNIX System Services and every user who wants to use the toolkit needs permissions to access the z/OS PT installation directory. The permissions are described in Chapter 4, “Installing z/OS Provisioning Toolkit,” on page 19.

Certain commands use REST API calls to communicate with z/OSMF; for example, the commands to run an image or list all the containers. Each user is prompted to enter a user ID and password to authenticate with z/OSMF before the command is run in z/OSMF.

Authority to create an environment

You can set specific user IDs to have authority for each workflow in the template. You set this in the properties file for the template (for example `../zospt/templates/template_name/template_type.properties`). This approach helps to retain control over which ID is used to perform steps in the workflow, for example, to mount zFS directories or create data sets.

Cloud Provisioning introduces the z/OS console services; an API that enables workflows to issue MVS™ console commands. Workflows use the API during the provisioning and deprovisioning process, and when containers are started and stopped. Create or identify a user ID that the workflows can use to submit API requests. Specify this user ID in the properties file for the template and ensure that the user ID has the appropriate security configuration, as described in the property.

When the workflow is uploaded to z/OSMF as part of a template, the named approver for each step must give their approval through the z/OSMF web interface. A template cannot be published and run until all the steps are approved.

This approach means that developers can provision environments with a low-level authority because other IDs perform the steps in the workflow to provision an environment. All owners of user IDs used in a workflow have visibility of how the IDs are being used and approve their usage.

Authority to create and run workflows

System programmers who want to upload templates require extra authority in z/OSMF. Becoming a Domain administrator provides this level of authority. System programmers require permissions to do the following tasks:

- Upload templates.
- Associate templates with tenants.
- Refresh templates.
- Delete templates.
- Run workflows in all appropriate tenants.
- View all template instances that are running.

Application developers require only Consumer authority to run workflows in a tenant and query the environments that they provision. Typically, you can add a group of user IDs to a tenant, for example a development team and supporting operations team. All users in that tenant can run the associated workflows within the designated limits.

Configuring a truststore

The connection between z/OS PT and z/OSMF is made over HTTPS. z/OS PT can be configured to use a truststore, enabling it to verify the identity of the z/OSMF server before a secure connection is established. For more information, see Chapter 6, “Configuring z/OS Provisioning Toolkit,” on page 23.

Other permissions

Depending on the environment, some workflows might require you to configure extra security or permissions before they can be run by toolkit users:

- For CICS permissions, see “Security requirements for provisioning CICS with z/OS Provisioning Toolkit” on page 72.
- For z/OS Connect EE permissions, see “Security requirements for provisioning instances of z/OS Connect EE with z/OS Provisioning Toolkit” on page 103.

Workflows can also include security configuration, so you can comply with your company standards around securing development environments.

Chapter 8. Getting started with z/OS Provisioning Toolkit

z/OS PT provides different ways to get started, depending on the kind of middleware that is used. Use this topic as a guide to find out what information and samples are provided with z/OS PT.

Introduction to getting started with z/OS Provisioning Toolkit

Use this topic to see what z/OS PT provides to help you get started with different kinds of middleware such as CICS or z/OS Connect EE.

Getting started with CICS

This diagram shows the progression between the different CICS scenarios that are available. These scenarios enable a system programmer to assess the technology before making z/OS PT available to application developers for self-service provisioning.

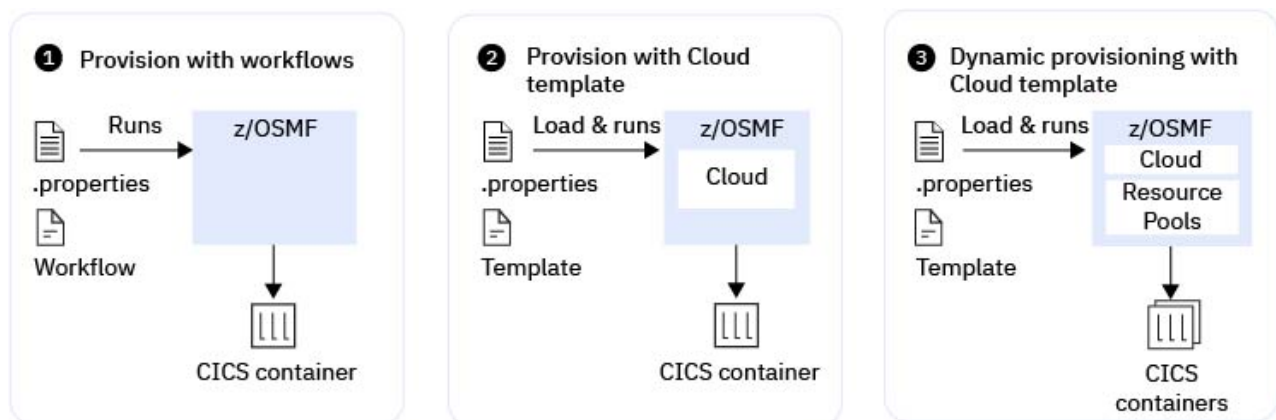


Figure 3. Progression between CICS scenarios

z/OS PT provides a simple getting started scenario that enables CICS system programmers to quickly evaluate the CICS workflows that are provided for provisioning and deprovisioning a CICS region before trying z/OS PT with Cloud Provisioning. By using the “Getting started with CICS by using z/OSMF workflows” on page 30 scenario, you run a workflow to provision a CICS container. The CICS container is not registered with the Cloud Provisioning software services registry because Cloud Provisioning isn't enabled.

A CICS system programmer can then progress to “Getting started with CICS by using IBM Cloud Provisioning and Management for z/OS” on page 34, which uses Cloud Provisioning, but with a pre-defined CICS region APPLID. This scenario uses the same CICS configuration as the previous scenario, which allows the system programmer to try Cloud Provisioning. The CICS container is registered with the Cloud Provisioning software services registry because Cloud Provisioning is enabled.

z/OS PT provides a set of sample templates that can be used with the full, dynamic, Cloud Provisioning CICS scenario, as described in “Adding a CICS

template by using IBM Cloud Provisioning and Management for z/OS” on page 59. This scenario allows an application developer to provision CICS containers in a self-service way, without requiring any z/OS-specific administration skills. The provisioning process runs under one or more automation user IDs, and APPLIDs and ports are allocated dynamically when required.

Getting started with z/OS Connect EE

z/OS PT provides a set of sample templates that can be used with the full, dynamic, Cloud Provisioning z/OS Connect EE scenario, as described in “Adding a z/OS Connect Enterprise Edition template by using IBM Cloud Provisioning and Management for z/OS” on page 100. This scenario allows an application developer to provision instances of z/OS Connect EE in a self-service way, without requiring any z/OS-specific administration skills. The provisioning process runs under one or more automation user IDs, and ports are allocated dynamically when required.

Getting started with other middleware

z/OS PT supports workflows to provision the following products:

- IBM MQ V9.0. The IBM MQ V9.0 workflow is supplied with IBM MQ V9.0. For more information, see Using IBM z/OSMF to automate IBM MQ.
- WebSphere® Liberty. To download the WebSphere Liberty workflow, follow this link: [WebSphere Liberty workflow in GitHub](#).
- IMS DB/TM system. To download the IMS workflow, follow this link: [IMS workflow in GitHub](#).

For more information about how to load these templates into z/OSMF, see Chapter 10, “Adding an IBM Cloud Provisioning and Management for z/OS template for use with z/OS Provisioning Toolkit,” on page 45.

Getting started with CICS by using z/OSMF workflows

This scenario takes a CICS system programmer through the steps to provision a simple single CICS container, by using z/OS PT to run the z/OSMF workflows.

Use these instructions to provision a container rapidly before you try z/OS PT with IBM Cloud Provisioning and Management for z/OS. You can become familiar with the artifacts you need to provision a container, the z/OS PT command line, and the provisioning lifecycle.

Before you begin

Before you, a CICS system programmer, provision a container by using z/OSMF workflows, complete the following checks.

- Ensure that z/OSMF is installed and configured. See Configuration in the z/OSMF Configuration Guide.
z/OSMF performs a system management role, and is accessed through a web interface. z/OSMF is a single point of control that allows you to:
 - View, define, and update policies that affect system behavior.
 - Monitor the performance of the systems in your enterprise.
 - Manage your z/OS software.
 - Perform data management tasks.
 - Consolidate your z/OS management tools.

- Run workflows to help you perform a common activity on z/OS.
- Ensure that your user ID has the appropriate authority to run workflows, by granting it the following access:
 - READ access to the ZMFAPLA class *SAF-prefixZOSMF.WORKFLOW.WORKFLOWS*.
 - READ access to EJBROLE *SAF-prefixIzuManagementFacilityWorkflow.izuUsers*

For more information, see Help with security setup in IBM Z solutions product documentation.

- Install z/OS PT by following the instructions here: Chapter 4, “Installing z/OS Provisioning Toolkit,” on page 19.

Before you can provision a CICS container, ensure that your user ID has the following authority:

- Authority to start a CICS region as a started task.
- Sufficient authority in a test environment for the following tasks:
 - Allocating data sets.
 - Creating and deleting log streams.
 - Adding a procedure into a PROCLIB.
 - Issuing console commands.

About this task

This task walks you through provisioning a simple single CICS container by using the z/OSMF workflows. Every step in this task runs under your user ID. A Job Entry System (JES) job is submitted for every step, and you can see the output from the JES jobs, and the source JCL. You can view the source and the output to help you to become familiar with how the workflow provisioning process works.

The following diagram shows how you can configure and run a supplied image to provision a CICS container by using the z/OSMF workflows.

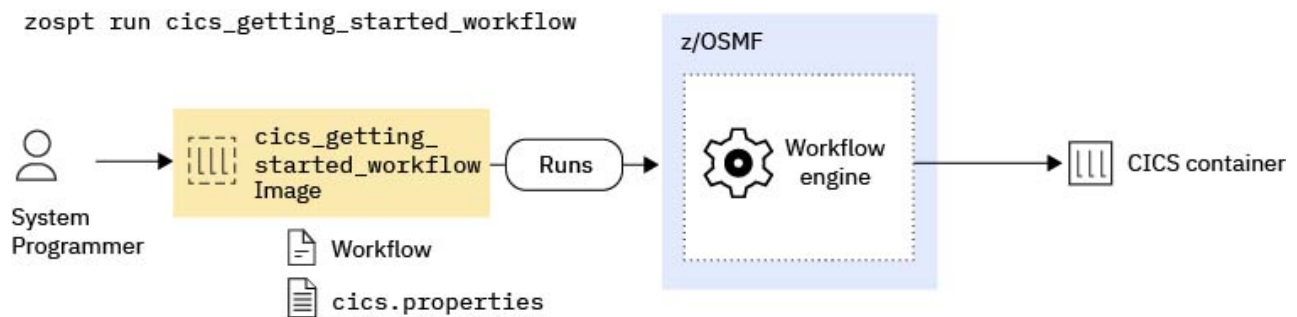


Figure 4. Provisioning a CICS container by using z/OSMF workflows

Procedure

1. Connect z/OS PT to z/OSMF.
 - a. Edit the z/OS PT `zosmf.properties` UTF-8 file in the `/zospt/config` directory. Update the following properties to use the z/OSMF workflows:

port

The HTTPS port that z/OSMF listens on. Ask your z/OSMF administrator for this information.

hostname

If you are running z/OS PT on the same LPAR that the z/OSMF server is running on, set **hostname=localhost**. To use a host name of localhost, configure z/OSMF to listen for requests on localhost by specifying **HOSTNAME(*)** in the **IZUPRMxx** parmlib member that contains configuration values for your z/OSMF server. If you do not know which LPAR the z/OSMF server is running on, ask your z/OS administrator. Otherwise, specify the host name.

systemNickname

Specify this property to set the nickname of the system on which the provisioning workflows are to run. If no value is specified, z/OS PT attempts to determine a default system nickname, which runs the workflows on the LPAR on which the z/OSMF server is running. If this default cannot be determined, or is not appropriate, ask your z/OSMF system programmer for the nickname of the LPAR on which you want to provision CICS regions. Alternatively, if your user ID has authority to see them, the **zospt ps -a** command reports available system nicknames. You must specify the host name and port of the z/OSMF server before you run the **zospt ps -a** command.

- b. Check that z/OS PT connects to z/OSMF by issuing the following command:

```
zospt ps -a
```

z/OS PT reports back basic information about the z/OSMF server.

```
2018-04-21 17:03:01 IBM z/OS Provisioning Toolkit V1.1.1
2018-04-21 17:03:05 Connecting to z/OSMF on host system01.hursley.ibm.com port 32000.
2018-04-21 17:03:06 z/OS level is V2R3.
2018-04-21 17:03:06 z/OS Management Facility level is V2R3.
2018-04-21 17:03:06 z/OSMF system nicknames are MV01, MV02, MV03, MV04, MV05, MV06, MV07, MV08, MV09.
2018-04-21 17:03:06 Getting started workflows will run on the system with nickname MV01.
NAME      IMAGE      OWNER      CREATED      STATE      TEMPLATE      SYSTEM      CONTAINER      TYPE
```

2. Edit the **cics.properties** file in the **zospt/templates/cics_getting_started** directory to configure the CICS region for the release you require (V5.1 or later) and to meet your naming standards. You must edit any property values that are surrounded by **<>** characters.

The **cics.properties** file contains the minimum set of CICS properties that is needed for you to get started, and information about each property that needs to be set. This list of properties is a subset of the full list of CICS properties.

3. Run the **cics_getting_started_workflow** image by issuing the following command:

```
zospt run cics_getting_started_workflow
```

If the provisioning is successful, output similar to the following example is displayed:

```
2018-04-23 16:02:35 IBM z/OS Provisioning Toolkit V1.1.1
2018-04-23 16:02:35 Running image cics_getting_started_workflow.
2018-04-23 16:02:35 Connecting to z/OSMF on host system01.hursley.ibm.com port 32000.
2018-04-23 16:02:37 Creating container USERTEST with id nzmay83vng8o4k9vkklaaxk8z4onwpdy on system MV01.
2018-04-23 16:02:43 Validating allowed properties for CICS getting started workflows - Complete
...
2018-04-23 16:03:57 Starting the CICS region - Complete.
2018-04-23 16:03:58 DFH_REGION_APPLID : USERTEST
2018-04-23 16:03:58 DFH_CICS_TYPE : Unmanaged
2018-04-23 16:03:58 Created container USERTEST with id nzmay83vng8o4k9vkklaaxk8z4onwpdy on system MV01.
2018-04-23 16:03:58 Container USERTEST has been started.
```


You can also run the following command to view your provisioned container, and to see that it has a container type of Workflow:

```
zospt ps
```

```
2018-04-26 11:41:21 IBM z/OS Provisioning Toolkit V1.1.1
2018-04-26 11:41:24 Connecting to z/OSMF on host localhost port 32000.
NAME      IMAGE      OWNER      CREATED      STATE      TEMPLATE  SYSTEM  CONTAINER TYPE
USERTEST  cics_getting_started_workflow  user01      2018-04-26T11:34:56  provisioned  N/A      MV01      Workflow
```

If the provision fails, the output contains the following diagnosis information; most importantly, the job information:

```
...
2018-04-23 16:38:56 ERROR: The request failed to complete.
2018-04-23 16:38:56 IZUNF0150E: The job that was submitted in step "Validating access to CICS data sets" failed during automation processing. Return code: "CC 0004" .
2018-04-23 16:38:56 Workflow Name      : provision_USERTEST_1523543918825
2018-04-23 16:38:56 Job Information   : JOBNAME: IZUNFJB, JOBID: JOB25101, CC 0004
2018-04-23 16:38:56 Current Step Title : Validating access to CICS data sets
...
```

The workflow steps submit JES jobs under your user ID. If the run fails, you can use the information that is returned in the failure to examine the job output and find the problem.

- a. View the JES job to diagnose the error. Find the job and view the output for the failing job step. You can also view the source JCL for any of the jobs, because the jobs all run under your user ID. Use this information to diagnose and fix the error.
- b. Deprovision the failed container by using the following command:

```
zospt rm -f containerName
```

In this example, *containerName* is USERTEST. The output shows a successful deprovision.

```
2018-04-23 17:00:22 IBM z/OS Provisioning Toolkit V1.1.1
2018-04-23 17:00:22 Performing deprovision on container USERTEST.
2018-04-23 17:00:28 Connecting to z/OSMF on host system01.hursley.ibm.com port 32000.
2018-04-23 17:00:35 Container USERTEST has been deleted.
2018-04-23 17:00:35 Deletion of environment completed successfully.
2018-04-23 17:00:35 Removal of all containers completed successfully.
```

The deprovision process removes everything that was created during the provisioning process, so no manual deletion is needed.

- c. Rerun the **zospt run** command. Issue the following z/OS PT command:

```
zospt run cics_getting_started_workflow
```

Results

You successfully connected z/OS PT to z/OSMF, customized the CICS region, and used the z/OS PT command line interface to provision and deprovision a CICS container.

What to do next

Before you move on to trying a scenario that uses Cloud Provisioning, you can try exploring further, for example, by adding further customization to the `cics.properties` file. To do so, follow these steps:

1. Try out the other **zospt** commands. See Chapter 19, “The **zospt** command syntax,” on page 135.
2. Deprovision your CICS container by using the following command, where *containerName* is USERTEST in this example:

```
zospt rm -f containerName
```

3. Change some properties in the `cics.properties` file to configure your CICS region, for example, setting SIT parameters or DFHRPL concatenation. For more information, see “Optional CICS properties” on page 85.
4. Reissue the **zospt run** command.

When you are ready, you can move on to “Getting started with CICS by using IBM Cloud Provisioning and Management for z/OS,” which uses the same workflows and configuration, but introduces Cloud Provisioning. Before you start this scenario, you must deprovision your CICS container by using the following command, where *containerName* is USERTEST in this example:

```
zospt rm -f containerName
```

Getting started with CICS by using IBM Cloud Provisioning and Management for z/OS

This scenario takes a system programmer through the steps to provision a simple single CICS container by using z/OS PT and IBM Cloud Provisioning and Management for z/OS.

This scenario uses the same set of workflows and properties as the scenario “Getting started with CICS by using z/OSMF workflows” on page 30, but introduces the use of Cloud Provisioning. System programmers can use this scenario to become familiar with Cloud Provisioning before they use the capability to enable self-service for application developers. For more information about the capabilities of Cloud Provisioning, see Chapter 1, “What is z/OS Provisioning Toolkit?,” on page 1 and Preparing to use Cloud Provisioning in the z/OSMF Configuration Guide.

Before you begin

Before you provision a container by using Cloud Provisioning, complete the following tasks:

- Follow the steps in “Getting started with CICS by using z/OSMF workflows” on page 30 to customize the properties for the workflows, and to configure connectivity from z/OS PT to z/OSMF.
- Ensure that IBM Cloud Provisioning and Management for z/OS is installed and configured. See Preparing to use Cloud Provisioning in the z/OSMF Configuration Guide.
- Request access from your z/OSMF system administrator (Landlord) to a domain to which your template can be added. This scenario uses the default domain that is created during the Cloud Provisioning installation process. For more information, see Chapter 7, “Security planning for z/OS Provisioning Toolkit with IBM Cloud Provisioning and Management for z/OS,” on page 25.
- Ensure that the system programmer's user ID is a domain administrator in z/OSMF.
- Ensure that you deprovisioned the CICS container that was used in “Getting started with CICS by using z/OSMF workflows” on page 30, because you need the APPLID for this scenario. Deprovision your CICS container by using the following command, where *containerName* is USERTEST in this example:

```
zospt rm -f containerName
```

About this task

In this task, you add a Cloud Provisioning template in z/OSMF from which a CICS region can be provisioned. The provisioning process runs under the system programmer's user ID.

The following diagram shows the flow of how a CICS container is provisioned by using Cloud Provisioning. This scenario uses a predefined CICS region APPLID, and the same CICS configuration as “Getting started with CICS by using z/OSMF workflows” on page 30.

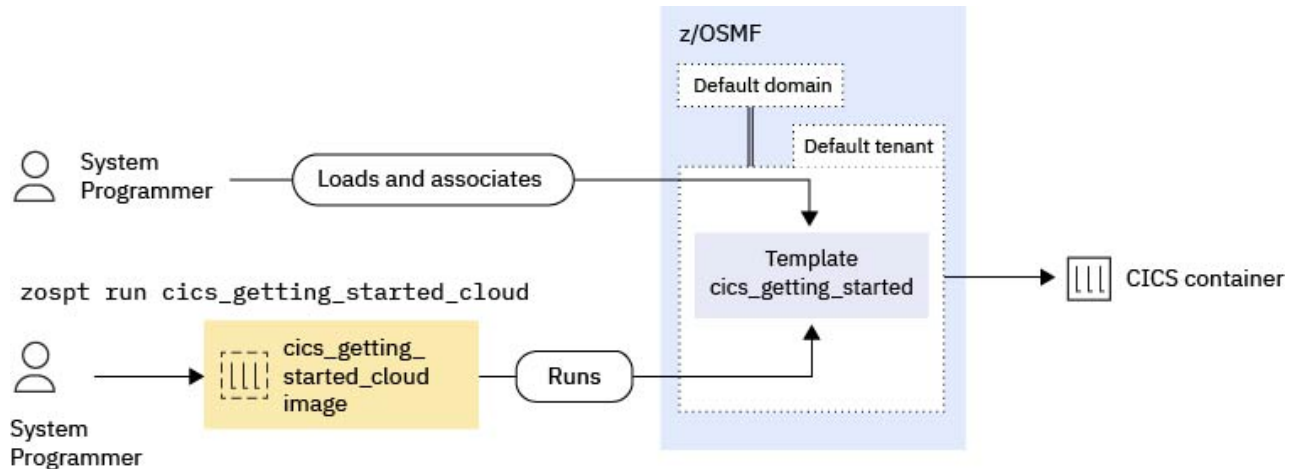
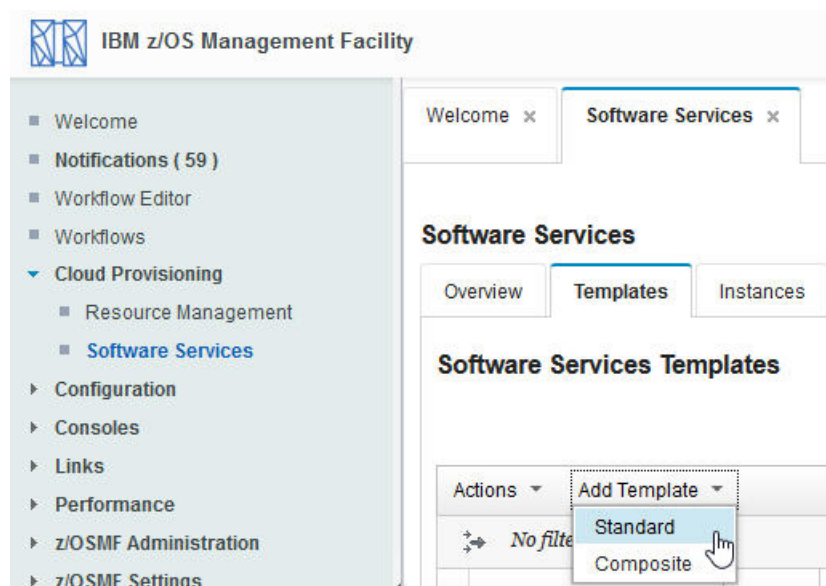


Figure 5. Provision a CICS container by using Cloud Provisioning

Procedure

1. Add the Cloud Provisioning template into z/OSMF. From a web browser, connect to the z/OSMF web interface.
 - a. Click **Cloud Provisioning** > **Software Services** > **Templates**. Click **Add Template** > **Standard**.



- b. In the **Template Source file** field in the Add Standard Template window, enter the full path of the manifest file for the CICS getting started template: `zospt/templates/cics_getting_started/cics.mf`. z/OSMF must be able to read from this directory tree.
- c. Click **Load**. The **Workflow file**, **Actions file**, and **Workflow variables input file** fields are populated automatically.
- d. Enter the name `cics_getting_started` in the **Template Name** field.
Use this name for your template so that you can use it with the sample z/OS PT image that is provided.

Software Services

Overview
Templates
Instances

Software Services Templates
Add Standard Template

Add Standard Template

Template source file:
Specify the file to populate this form.

/zospt/templates/cics_getting_started/cics.mf
Load

* Target domain:

default

* Template name: ?

cics_gettings_started

* Workflow file:

/zospt/workflows/cics/provision.xml

* Actions file:

/zospt/templates/cics_getting_started/cics_actions.xml
Create New

Workflow variables input file:

/zospt/templates/cics_getting_started/cics.properties

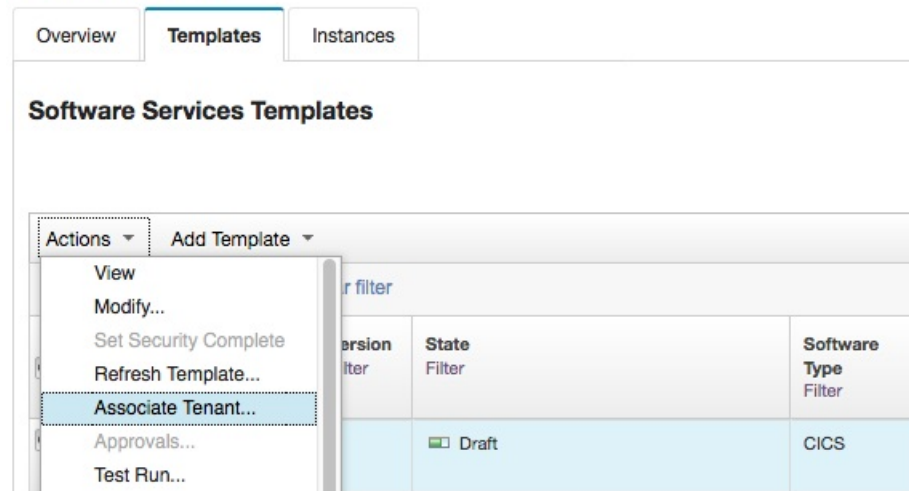
OK
Cancel

- e. Click **OK**.

The template is loaded into z/OSMF in Draft state.

2. Associate the template with a tenant in the domain to determine the set of users that can run the template:
 - a. Select the `cics_getting_started` template. From the **Actions** menu, select **Associate Tenant**.

Software Services



- b. Select the tenant that your z/OSMF administrator specifies.
- c. Click **OK**.
- d. In the Add Template and Resource Pool for Tenant view, specify the following properties and values:

Property	Setting
Software services instance name prefix	Specify a general name prefix, for example, "CGS" (for CICS getting started).
Maximum number of software services instances	Specify 1 (one). The properties file for this template contains the CICS APPLID, so you can have one CICS region active at a time. (When you run a fuller configuration as described in "Adding a CICS template by using IBM Cloud Provisioning and Management for z/OS" on page 59, this option controls the maximum number of regions that are allowed).
Maximum number of software services instances for a user	Specify 1 (one). The properties file for this template contains the CICS APPLID, so you can have one CICS region active per user at a time. (When you run a fuller configuration as described in "Adding a CICS template by using IBM Cloud Provisioning and Management for z/OS" on page 59, this option controls the maximum number of regions that are allowed per person).
Specify customized JOB statement JCL	Select this option. If you need to, you can customize the JOB statement. The job statement is applied to all JES jobs submitted as part of the provisioning process.
System selection for provisioning	Choose the system in the sysplex into which you want to provision the CICS regions. The default is to provision into the same system that is running z/OSMF. For more information about configuring z/OSMF for provisioning across different systems, see z/OSMF product documentation. Note: To use this template with z/OS PT, do not select the Prompt user for system option.

Add Template and Resource Pool for Tenant: default

Select a template name, then supply values that will be used to create a resource pool for the template. If required template is not in the list, go to the Software Services task to add a template.

* Select template:

cics_getting_started ▼

Template type:

Standard

* Software services instance name prefix:

☐ Use SNA APPLID

Specify general name prefix: ?

CGS

* Maximum number of software services instances (1-1296):

1

Maximum number of software services instances for a user:

1

☒ Allow account information to be modified when the template is provisioned

JOB statement JCL: ?

☒ Specify customized JOB statement JCL

```
//${_workflow-softwareServiceInstanceName} JOB (ACCTINFO),CLASS=A,MSGCLASS=0,
//      MSGLEVEL=(1,1),REGION=0M,NOTIFY=${_step-stepOwnerUpper}
```

e. Click **OK**.

The template is now ready for use with z/OS PT. The template is in draft mode, which means that only the creator can run the template.

3. Optional: If you are not using the default domain and tenant, you must set the following environment variables to set up the environment for z/OS PT:

- `export zospt_domain=domain_name`
- `export zospt_tenant=tenant_name`

4. Run the template in draft mode to test the template before you publish it and allow other users to run it. To run the template in draft mode, issue the following command:

```
zospt run cics_getting_started_cloud --draft
```

Example output is as follows:

```
2018-04-24 15:08:16 IBM z/OS Provisioning Toolkit V1.1.1
2018-04-24 15:08:16 Running image cics_getting_started_cloud.
2018-04-24 15:08:16 The z/OSMF template used is cics_getting_started.
2018-04-24 15:08:16 The z/OSMF domain is default.
2018-04-24 15:08:16 The z/OSMF tenant is default.
2018-04-24 15:08:16 Connecting to z/OSMF on host localhost port 32000.
2018-04-24 15:08:17 The z/OSMF template cics_getting_started was created by USER01 at date:time. The template type is standard. It was last modified by USER01 at date:time.
2018-04-24 15:08:19 Creating container CICS_CG500 with id 4fcfd7f6-f377-458a-922e-61ae3365ccf5 on system MV01.
2018-04-24 15:08:24 Validating allowed properties for CICS getting started workflows - Complete.
...
2018-04-24 15:09:39 Starting the CICS region - Complete.
2018-04-24 15:09:40 DFH CICS TYPE : Unmanaged
2018-04-24 15:09:40 DFH_REGION APPLID : IYKZLVR
2018-04-24 15:09:41 Created container CICS_CG500 with id 4fcfd7f6-f377-458a-922e-61ae3365ccf5 on system MV01.
```

You can also run the following command to view your provisioned container:

```
zospt ps
```

Example output is as follows:

```
2018-04-26 13:57:29 IBM z/OS Provisioning Toolkit V1.1.1
2018-04-26 13:57:29 Connecting to z/OSMF on host localhost port 32000.
NAME      IMAGE      OWNER    CREATED      STATE    TEMPLATE      SYSTEM    CONTAINER TYPE
USERTEST  cics_getting_started_cloud  user01    2018-04-26T12:56:03  provisioned  cics_getting_started  MV01      Standard
```

Results

You successfully added a Cloud Provisioning template in z/OSMF, associated the `cics_getting_started` template with a tenant in the domain, and ran the template.

What to do next

1. Deprovision your CICS container by using the command `zospt rm -f containerName`.
2. Build your z/OS PT knowledge by moving on to the next section of the documentation, Chapter 7, “Security planning for z/OS Provisioning Toolkit with IBM Cloud Provisioning and Management for z/OS,” on page 25, and continuing through the following sections.
3. When you are ready, try a full, dynamic scenario “Adding a CICS template by using IBM Cloud Provisioning and Management for z/OS” on page 59, which uses the provided sample template `cics_55`.

Getting started with other middleware

In addition to CICS, you can use z/OS PT to provision other types of middleware. Some templates are provided with z/OS PT, and others with the relevant IBM products.

Install z/OS PT

Before you can provision any middleware, you must install z/OS PT. See Chapter 4, “Installing z/OS Provisioning Toolkit,” on page 19.

z/OS PT templates for other middleware

z/OS PT provides templates for z/OS Connect EE.

- For more information about templates for z/OS Connect EE, see Chapter 21, “Samples provided with z/OS Provisioning Toolkit,” on page 141.
- For more information about how to load z/OS Connect EE templates in z/OSMF, see “Adding a z/OS Connect Enterprise Edition template by using IBM Cloud Provisioning and Management for z/OS” on page 100.

Available workflows for other middleware

z/OS PT supports workflows to provision the following products:

- IBM MQ V9.0. The IBM MQ V9.0 workflow is supplied with IBM MQ V9.0. For more information, see Using IBM z/OSMF to automate IBM MQ.
- WebSphere Liberty. To download the WebSphere Liberty workflow, follow this link: [WebSphere Liberty workflow in GitHub](#).
- IMS DB/TM system. To download the IMS workflow, follow this link: [IMS workflow in GitHub](#).

For more information about how to load templates for these workflows into z/OSMF, see Chapter 10, “Adding an IBM Cloud Provisioning and Management for z/OS template for use with z/OS Provisioning Toolkit,” on page 45.

Chapter 9. Preparing to use z/OS Provisioning Toolkit with IBM Cloud Provisioning and Management for z/OS

Before you use z/OS PT for dynamic provisioning, ensure that your system is set up correctly, and that you understand the capabilities of each of the components you need. With dynamic provisioning, you can dynamically allocate ports and other resources from a pool, and add and remove security profiles for provisioned environments.

The following information explains the components that you need to install or configure to be able to dynamically provision a subsystem by using Cloud Provisioning and z/OS PT.

Security planning

Before you enable Cloud Provisioning, you must understand the Cloud Provisioning security model. See Chapter 7, “Security planning for z/OS Provisioning Toolkit with IBM Cloud Provisioning and Management for z/OS,” on page 25.

Install z/OS Management Facility (z/OSMF) and IBM Cloud Provisioning and Management for z/OS

For information about how to install and configure z/OSMF, see Configuration in the z/OSMF Configuration Guide.

For a summary of the function that z/OSMF provides, and initial prerequisites, see “Prerequisites for getting started by using z/OSMF workflows” on page 17.

The following plug-ins are included in the z/OSMF core function and are relevant to using z/OS PT for dynamic provisioning:

- Resource Management. With this capability, you can:
 - Create and manage domains.
 - Create and manage tenants.
 - Add resource pools and software services templates to tenants.
- z/OS console services. With this capability, you can:
 - Use a Representational State Transfer (REST) API to issue console commands.
 - Manage all provisioned systems through a single console.
 - Detect unsolicited messages to determine the status of a provisioned system, such as whether a CICS region has successfully started or stopped. For example, look for a Control has been given to CICS message to know when CICS startup is complete.

For provisioning a CICS region by using IBM Cloud Provisioning and Management for z/OS, use of z/OS console services is optional and depends on the value of the **CONSOLE_COMMAND_TYPE** configuration property. See “Workflow configuration variables” on page 89.

To use z/OS console services, you must configure z/OSMF z/OS console services and configure the relevant authorization for the userid that the **DFH_ADMIN_CONSOLE** configuration property specifies. See z/OS console services in

the z/OSMF Programming Guide and “Security requirements for provisioning CICS with z/OS Provisioning Toolkit” on page 72.

If you do not use z/OS console services, the equivalent functions are provided by REXX scripts, in same way as for the scenario “Getting started with CICS by using z/OSMF workflows” on page 30. A console is opened for each provisioned CICS region, using the APPLID of the CICS region as the console name.

To provision an instance of a subsystem dynamically, you also need to install and configure IBM Cloud Provisioning and Management for z/OS (Cloud Provisioning). See Preparing to use Cloud Provisioning in the z/OSMF Configuration Guide.

Functions that Cloud Provisioning provides include:

- Creating reusable templates for provisioning subsystems (for example, CICS or z/OS Connect EE).
- Running steps in the provisioning process under different automation user IDs.
- Controlling the users that are allowed to use each template.
- Limiting the number of provisioned subsystems each user can have.
- Accessing a software registry of provisioned (or registered) systems.

Configure IBM Cloud Provisioning and Management for z/OS

Before you use Cloud Provisioning, ensure that the following relevant actions are complete:

- A z/OSMF administrator (Landlord) must define a tenant with which the template can be associated, and the domain to which that tenant belongs. z/OSMF comes with a default domain and default tenant.
- A z/OSMF administrator (Landlord) must give you domain administrator authority to enable you to add templates into the domain. If you cannot use the supplied default domain, ensure that your user ID is added as a domain administrator to a domain that you can use.
- You must ensure that you have user IDs with sufficient authority to complete the workflow steps. You must have a user ID that can use the z/OSMF z/OS console services (an API) to issue MVS commands.
- When APAR PI77388 is applied to z/OSMF, z/OSMF can provision environments across different systems in a sysplex. When you use this capability, you must ensure that data set high-level qualifiers and zFS directories that are used during provisioning are available on all the systems to which you want to provision environments. For more information about configuring z/OSMF for provisioning across different systems, see Cloud provisioning services in the z/OSMF Programming Guide.

Install the Configuration Assistant plug-in

This plug-in is called the Configuration Assistant for z/OS Communications Server plug-in (z/OSMF V2R2) or the Network Configuration Assistant plug-in (z/OSMF V2R3). For installation information, see Prerequisite plug-ins for Cloud Provisioning in the z/OSMF Configuration Guide.

Functions that this plug-in provides include:

- Defining ranges of subsystem names (for example APPLIDs or storage subsystem identifiers (SSIDs)), ports, and IP addresses for use in provisioning.

- Managing network resource pools for cloud provisioning templates.
- Providing subsystem names (for example APPLIDs or SSIDs), ports, and IP addresses during provisioning (and returning this information during deprovisioning).
- Updating the TCPIP stacks with information about which systems have been allocated which ports.

For more information about the Configuration Assistant, see Network Configuration Assistant task overview in the z/OSMF Configuration Guide. For information about configuring TCPIP, see TCPIP: Getting started in z/OSMF Online Help.

Try out dynamic provisioning

You are now ready to try dynamically provisioning a subsystem.

- To dynamically provision a CICS region, see “Adding a CICS template by using IBM Cloud Provisioning and Management for z/OS” on page 59.
- To dynamically provision an instance of z/OS Connect EE, see “Adding a z/OS Connect Enterprise Edition template by using IBM Cloud Provisioning and Management for z/OS” on page 100.
- To dynamically provision another subsystem (for example, IBM MQ, or Liberty), see Chapter 10, “Adding an IBM Cloud Provisioning and Management for z/OS template for use with z/OS Provisioning Toolkit,” on page 45.

Chapter 10. Adding an IBM Cloud Provisioning and Management for z/OS template for use with z/OS Provisioning Toolkit

The foundation of provisioning with z/OS PT is a z/OSMF template. The template pulls in the z/OSMF workflows and associated actions and variables that define both the environment to be provisioned, and how you want to provision it. You load the template into z/OSMF by using the z/OSMF web interface. After you load a template, you can use z/OS PT to build an image based on that template.

Before you begin

Before you add a Cloud Provisioning template, follow the steps here: Chapter 9, “Preparing to use z/OS Provisioning Toolkit with IBM Cloud Provisioning and Management for z/OS,” on page 41.

About this task

For a reminder of how templates work with z/OS PT, see Chapter 1, “What is z/OS Provisioning Toolkit?,” on page 1.

The following procedure summarizes how to add a template to z/OSMF and prepare it to use with z/OS PT. For more information about how to complete each step, see z/OSMF product documentation.

For a walk-through of the process by using either a CICS or z/OS Connect EE template, see “Adding a CICS template by using IBM Cloud Provisioning and Management for z/OS” on page 59 or “Adding a z/OS Connect Enterprise Edition template by using IBM Cloud Provisioning and Management for z/OS” on page 100.

Procedure

1. Prepare a template for use with z/OSMF. Use the *.properties file to customize the workflow to create an environment that meets your company standards.
2. Log on to the z/OSMF web interface and add the template. Go to **Cloud Provisioning > Software Services** and select the **Templates** tab. Select **Add Template** to add the template.
3. Ensure that all owners of the user IDs in the workflow approve the template.
4. Associate the template with a tenant. During this step, you can set limits for how many times the container can be provisioned and how many containers each user can provision. The association automatically creates a resource pool for the template and the network administrator is notified.
5. Make sure that the necessary network resources are allocated to the resource pool for the template. This step must be completed by a network administrator.
6. After the template is approved and the network resources are allocated, test the template before you publish it to users in the tenant. To test the template, build an image that references the template name and run the **zospt run** command with the **--draft** option. If you did not use the default domain and tenant, you must update your .profile in z/OS UNIX System Services to add the following environment variables:

```
zospt_domain=zospt_domain_name  
zospt_tenant=zospt_tenant_name
```

For more information about building an image to run a template, see “Building an image with z/OS Provisioning Toolkit” on page 49.

7. After your testing is complete, use the z/OSMF web interface to publish the template for all users in the tenant to use.

What to do next

Use z/OS PT to build images from the template that you loaded. See “Building an image with z/OS Provisioning Toolkit” on page 49 for details.

Chapter 11. z/OS Provisioning Toolkit images

A z/OS PT image is a binary object that contains the configuration, application code, and environment variables to rapidly provision a preconfigured environment that is tailored to a specific user's requirements. An image can also name a template that z/OS PT uses to provision the middleware that is specified in the template. z/OS PT provides a set of sample CICS and z/OS Connect EE images to help you get started.

Overview of z/OS PT images

A z/OS PT image is a set of files and configuration information that is built from a `zosptfile`, which contains syntax statements about what to include in an image, and what the image needs to run in a container. After an image is built, you can use z/OS PT to run it and provision a container with the image contents. If you run the same image multiple times, you provision identical containers. A z/OS PT image extends the configuration of an environment that is defined by a template that you load into z/OSMF. By providing configuration in images, you have more flexibility to tailor the environment to a user's needs, and can reduce the number of templates that need to be loaded in z/OSMF.

The template contains a base set of configuration that can be overridden in the images. The environment variables that you can set in an image vary by subsystem. For CICS, you can choose which environment variables you want to allow an image to override. See “Modifying the set of CICS configuration properties that can be overridden” on page 93.

Why use a z/OS PT image?

z/OS PT enables users to build images to configure a middleware instance that is provisioned from a z/OSMF template. Some of the benefits of using z/OS PT images are as follows:

- Images enable a single z/OSMF template to be used to provision systems with different configurations.
- An image can contain zFS artifacts that are deployed to the file system and used by the provisioned subsystem.
- Images are reusable and can be run multiple times.
- Images are easy to build and do not require reapproval of the template.
- An application developer can build images upon an image that is supplied by the system programmer.

The following diagram shows that the CICS system programmer provides the base image for any developer who needs a CICS TS V5.5 container. A developer can choose to have a Java Liberty environment or a JSON web services environment. By basing an image upon the `cics_55` image, all three use cases can use the same CICS TS V5.5 template in z/OSMF. Each image only needs to provide a few environment variables to customize the CICS region to meet a developer's needs. The application developer runs the `cics_55_json` image, which uses the base image that the system programmer supplied, and the `cics_55` template in z/OSMF to provision a CICS container that provisions an Axis2 JVM server for JSON web services.

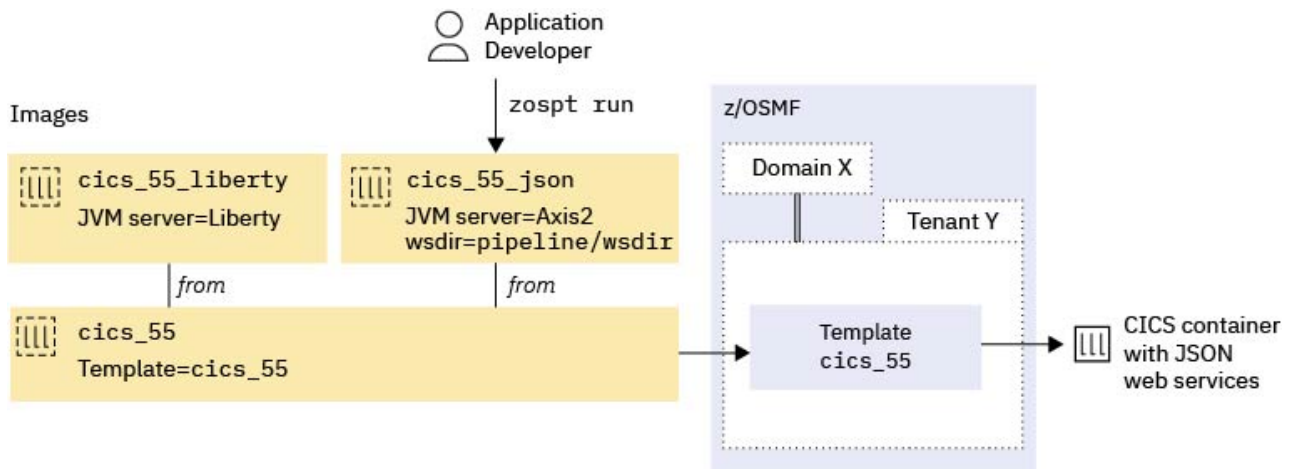


Figure 6. Images that are built from a base image

Using a simple **from** statement in the zosptfile, you can build dependencies between images until you have a stack of images. A base image has **from scratch** as the first line. It has no dependencies and must contain the name of a template in z/OSMF. You can choose how granular to make your images. The following example shows three images in a stack, where the base image, `cics_55`, can be run to provision a simple CICS region. The next image, `cics_55_liberty`, contains configuration such that when it is run, it provisions a CICS region with a Liberty server. The first image, `my_java_app`, built by an application developer, builds a Java application. If an application developer runs the image, a CICS region is provisioned with Liberty and with the Java application running inside.

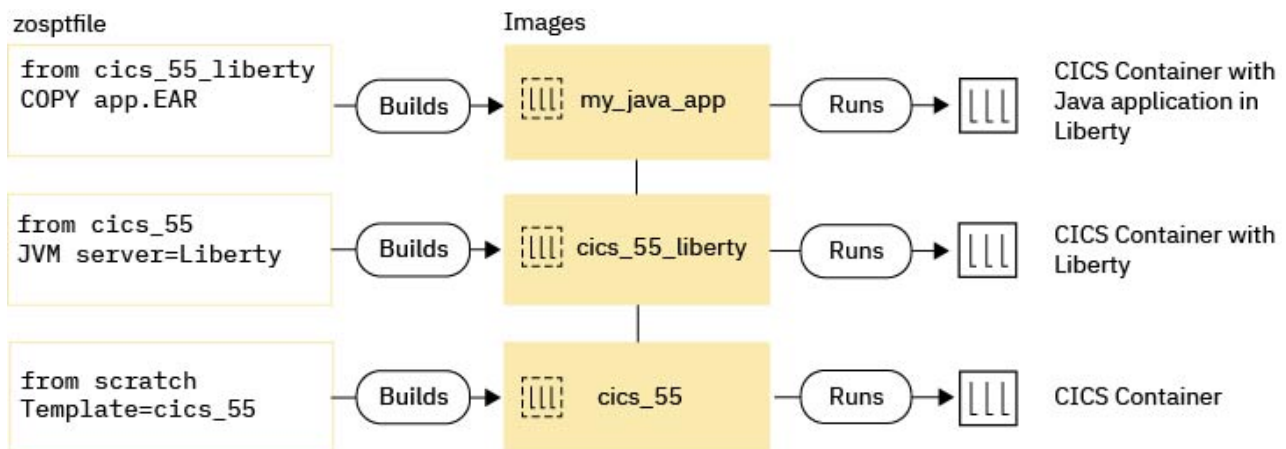


Figure 7. A stack of z/OS PT images

z/OS PT sample images

z/OS PT provides sample images in the `zospt/images` directory, which are built from the corresponding sample zosptfile from the `zospt/samples` directory. For more information about the sample images that are provided with z/OS PT, see “Sample images and the source artifacts that are used to build them” on page 142.

Building an image with z/OS Provisioning Toolkit

After templates are set up in z/OSMF, you use the **zospt build** command to build an image. This image is a set of files and configuration information that is used with an associated z/OSMF template to provision one or more containers with the same capability. You need to set up the **zosptfile**, then run the **zospt build** command to build the image.

About this task

A z/OS PT image is created by using the **zospt build** command. You can create images from scratch, or base an image on an existing one. In this way, you can create stacks of images that together build different containers. For a reminder of how images work with z/OS PT, see Chapter 1, “What is z/OS Provisioning Toolkit?,” on page 1.

The build command takes the contents of a **zosptfile** and uses that to determine what is put into an image. A **zosptfile** is a text document in UTF-8 encoding that defines the instruction set that is used to assemble an image. Valid instructions are shown in the following table:

Instruction	Description
FROM <i>imageName</i>	Defines the image on which this image is based. If you specify FROM scratch, this image is the base image. It is not based on another image. scratch is a reserved base image that is the starting point for building environments. scratch cannot be directly built or run. Note: If you create an image by building on an existing image, the environment variables in the original image persist.
ENV ZOSMF_TEMPLATE = <i>template name</i>	Associate this image with a z/OSMF template.
ENV <i>key=value</i>	Define an environment variable for this image. Some templates, such as CICS, allow environment variables to be specified to further customize the configuration of a provisioned environment. You can customize some of the properties of a provisioned environment by adding environment variable definitions to the zosptfile for the image, which can override values that are specified in the properties file of the template that is named in the image. For more information about which properties can be added to the zosptfile for an image, see the Configuration properties file for the product you are provisioning. Environment variables that are defined in a zosptfile for <i>image_1</i> can also be referenced by other instructions in the same zosptfile , or by zosptfile instructions of other images built FROM <i>image_1</i> , by referencing the environment variable as <i>\$(key)</i> . For example, you might define an environment variable to represent the directory into which certain files are to be copied, and the environment variable might then be used in COPY instructions in the zosptfile to ensure that the files are copied into the correct directory. To show that an environment variable is only referenced by other instructions in the zosptfile , prefix the environment variable name with IMAGE_ . The sample images that are provided with z/OS PT give examples of environment variables. For more information about sample images, see Chapter 21, “Samples provided with z/OS Provisioning Toolkit,” on page 141.
COPY <i>fromFile toFile</i>	Copy a file or directory into this image. The file must be in the same directory as the zosptfile , or one of its subdirectories.

For examples of **zosptfile** text files, see Chapter 21, “Samples provided with z/OS Provisioning Toolkit,” on page 141.

Procedure

1. Edit the `zosptfile`:
 - a. Define the image on which this image is based:
 - To build this image from nothing, specify `FROM scratch`.
 - To build this image from an existing one, specify `FROM imageName`.
 - To see available images, use the **zospt images** command to print a list of the images that are available to provision, and the **zospt inspect** command to see more details about a specified image.
 - b. Optional: If the image is a base image, specify the template name that you want to call in z/OSMF when the image is run.
 - c. Optional: Set **environment variables**.
 - d. Optional: Copy files or directories into this image.
2. Run the **zospt build** command to build the image.

```
zospt build -t imageName path
```

This command builds an image that is called *imageName* by using the `zosptfile` found in the directory that is specified by *path*, and adds it to the local images directory.

This command operates on local resources, and runs without connecting to z/OSMF.

For a complete syntax description, and for information about how to remove an image, see Chapter 19, “The **zospt** command syntax,” on page 135.

If you set all the properties successfully, a success message is displayed, as shown in the following example console output:

```
Successfully built /u/user/zospt/samples/cics_55_liberty/zosptfile into  
image cics_55_liberty.
```

What to do next

The image is ready to use for provisioning with the **zospt run** command, as described in “Running a z/OS Provisioning Toolkit image.”

Running a z/OS Provisioning Toolkit image

Use the **zospt run** command to provision a container from an image. z/OS PT uses the z/OSMF REST API to run the appropriate template and provision a container that uses the information in the image and any dependent images. In just a few minutes, a running environment is available for development. You can also use the `link` option on the **zospt run** command to provision a new container with a link to an existing container.

Running an image

Ensure that z/OS PT is configured and can correctly identify the z/OSMF domain that contains the template to be run and the z/OSMF tenant that your user ID is assigned to. For a reminder of how to configure z/OS PT, see Chapter 10, “Adding an IBM Cloud Provisioning and Management for z/OS template for use with z/OS Provisioning Toolkit,” on page 45.

Run the **zospt run** command to create a container.

```
zospt run imageName
```

This command creates a container from the image called *imageName*.

If you are testing a template that is in the *Draft approved* state, specify the **--draft** option on the **zospt run** command. See the following example:

```
zospt run imageName --draft
```

The **--draft** option is useful when you want to validate that the z/OSMF template is correctly configured before you publish it for wider use.

For a complete syntax description, see Chapter 19, “The **zospt** command syntax,” on page 135.

For the command **zospt run cics_54**, the console output returns progress on each step in the workflow as the container is provisioned, including displaying some public properties, as shown in the following example:

```
2018-04-05 15:04:51 IBM z/OS Provisioning Toolkit V1.1.0
2018-04-05 15:04:51 Running image cics_54.
2018-04-05 15:04:51 The z/OSMF template used is cics_54.
2018-04-05 15:04:51 The z/OSMF domain is default.
2018-04-05 15:04:51 The z/OSMF tenant is default.
2018-04-05 15:04:51 Connecting to z/OSMF on host system01.ibm.com port 27820.
2018-04-05 15:05:10 The z/OSMF template cics_54 was created by USER01 at 2018-01-05T07:40:34.394Z.
It was last modified by USER01 at 2018-01-06T07:41:11.761Z.
2018-04-05 15:05:29 Creating container CICS_CICPJ00G with id cf91dd2f-9848-4672-af91-5aea7f80a02e on system MV01.
2018-04-05 15:05:34 Getting dynamic applid - Complete.
2018-04-05 15:05:36 Validating access to CICS data sets - Complete.
2018-04-05 15:05:41 Validating access to zFS directories - Complete.
2018-04-05 15:05:43 Allocating CMCI port - Complete.
2018-04-05 15:05:45 Allocating http port - Complete.
2018-04-05 15:05:47 Allocating https port - Complete.
2018-04-05 15:05:52 Creating CICS security configuration - Complete.
2018-04-05 15:06:14 Creating the zFS directory for the CICS region - Complete.
2018-04-05 15:06:45 Creating CICS region data sets - Complete.
2018-04-05 15:06:48 Formatting CICS region data sets - Complete.
2018-04-05 15:06:52 Creating CICS log stream model - Complete.
2018-04-05 15:07:23 Creating CICS CSD definitions - Complete.
2018-04-05 15:07:28 Adding the image into the provisioned file system - Complete.
2018-04-05 15:07:34 Creating the CICS Region JCL - Complete.
2018-04-05 15:07:35 DFH_REGION_CMCIPORT : 32001
2018-04-05 15:07:35 DFH_REGION_HTTP : 28465
2018-04-05 15:07:35 DFH_REGION_APPLID : CICPJ00G
2018-04-05 15:07:35 DFH_CICS_TYPE : SMSS
2018-04-05 15:07:35 DFH_JVM_DEBUG : NO
2018-04-05 15:07:35 DFH_REGION_ZFS_DIRECTORY : /u/cicprov/mount/CICPJ00G
2018-04-05 15:07:35 DFH_REGION_HTTPS : 28471
2018-04-05 15:07:37 Created container CICS_CICPJ00G with id cf91dd2f-9848-4672-af91-5aea7f80a02e on system MV01.
2018-04-05 15:07:38 Performing start on container CICS_CICPJ00G.
2018-04-05 15:07:42 Checking zFS is mounted - Complete.
2018-04-05 15:07:53 Starting the CICS Region - Complete.
2018-04-05 15:07:54 Started container CICS_CICPJ00G.
```

If parallel provisioning is enabled in the workflow for CICS templates that use Cloud Provisioning to provision a CICS region, some steps run in parallel so that provisioning completes faster. Output includes the message *This workflow contains parallel steps*. For example:

```
2018-10-18 04:12:32 IBM z/OS Provisioning Toolkit V1.1.3
2018-10-18 04:12:32 Running image cics_54.
2018-10-18 04:12:32 The z/OSMF template used is cics_54.
...
2018-10-18 04:13:37 This workflow contains parallel steps.
...
2018-10-18 04:15:18 Creating the zFS directory for the CICS region - Complete.
2018-10-18 04:15:18 Creating CICS region data sets - Complete.
2018-10-18 04:15:18 Formatting CICS region data sets - Complete.
2018-10-18 04:15:18 Creating CICS log stream model - Complete.
2018-10-18 04:15:18 Creating CICS system definition - Complete.
2018-10-18 04:15:21 Setting directory permissions - Complete.
2018-10-18 04:15:21 Creating CICS CSD - Complete.
2018-10-18 04:15:21 Creating CICS CSD definitions - Complete.
2018-10-18 04:15:21 Creating the CICS region JCL - Complete.
...
2018-10-18 04:15:40 Starting the CICS region - Complete.
2018-10-18 04:16:45 Container CICS_CICPD000 has been started.
```

For details, see “Enabling faster provisioning by using parallel processing” on page 63.

If the properties are all correct, the output from the **zospt run** command ends with a message that says the container is started. Your provisioned container is ready to use. The output also contains the details of how to access the container to use it for development and testing. If you have any problems when you provision a container, see Chapter 18, “Troubleshooting z/OS Provisioning Toolkit,” on page 129.

Using the **--name** option

Use the **--name** option on the **zospt run** command to assign a meaningful name to a container, instead of having an automatically assigned container name:

```
zospt run imageName --name containerName
```

This command provisions a container with name *containerName* (where *containerName* must be unique). The maximum name length is 32 characters. The acceptable characters are A-Z a-z 0-9 _ -.

Using the **--link** option

Use the **link** option on the **zospt run** command to provision a new container with a link to an existing container:

```
zospt run imageName --link containerName|containerId:<alias>
```

This command creates a new container from the image that is called *imageName* and connects it to an existing container (identified by either *containerName* or *containerId*), which can be given an *alias*. Existing subsystems must already be provisioned by using z/OSMF or z/OS PT or registered as containers in z/OSMF so that they can be linked to by other containers.

You can specify the **--link** option multiple times in a single command, for example:

```
zospt run cics_54_liberty --link REG-DB2_TDB201:db2 --link REG-MQ_TMQ00:mq
```

For more information about registering existing subsystems in z/OSMF, see Chapter 15, “Registering existing subsystems in z/OSMF,” on page 117. For more information about supported link scenarios, see “Link scenarios supported by z/OS Provisioning Toolkit” on page 54.

For command `zospt run cics_54_test --link REG-DB2_DB200`, the console output returns progress on each step in the workflow as the container is provisioned and reports back properties about the provisioned container, for example, its port numbers, which is shown in the following example:

```

2018-10-03 11:58:55 IBM z/OS Provisioning Toolkit V1.1.0
2018-10-03 11:58:55 Running image cics_54_test.
2018-10-03 11:58:55 The z/OSMF template used is cics_54.
2018-10-03 11:58:55 The z/OSMF domain is default.
2018-10-03 11:58:55 The z/OSMF tenant is default.
2018-10-03 11:58:55 Connecting to z/OSMF on host system01.hursley.ibm.com port 32000.
2018-10-03 11:58:56 The z/OSMF template cics_54 was created by USER01 at 2018-01-04T14:32:52.199Z.
It was last modified by USER01 at 2018-01-05T13:11:54.307Z.
2018-10-03 11:59:02 Creating container CICS_CICPD002 with id 4743267a-5321-4420-86bb-a7e8040a1674 on system MV01.
2018-10-03 11:59:12 Getting DB2 object id - Complete.
2018-10-03 11:59:12 Getting necessary DB2 data - Complete.
2018-10-03 11:59:12 Getting dynamic applid - Complete.
2018-10-03 11:59:12 Validating access to CICS data sets - Complete.
2018-10-03 11:59:20 Validating access to zFS directories - Complete.
2018-10-03 11:59:20 Validating console commands can be issued - Complete.
2018-10-03 11:59:23 Hardening state indicating template validation steps passed. - Complete.
2018-10-03 11:59:26 Allocating CMCI port - Complete.
2018-10-03 11:59:29 Allocating HTTP port - Complete.
2018-10-03 11:59:32 Allocating HTTPS port - Complete.
2018-10-03 11:59:39 Obtaining hostname for the lpar - Complete.
2018-10-03 11:59:45 Creating the zFS directory for the CICS region - Complete.
2018-10-03 11:59:51 Creating CICS region data sets - Complete.
2018-10-03 11:59:54 Formatting CICS region data sets - Complete.
2018-10-03 11:59:57 Creating CICS log stream model - Complete.
2018-10-03 12:00:01 Adding the image into the provisioned file system - Complete.
2018-10-03 12:00:07 Set directory permissions - Complete.
2018-10-03 12:00:36 Creating CICS CSD definitions - Complete.
2018-10-03 12:00:39 Hardening state indicating successful CSD update - Complete.
2018-10-03 12:00:45 Creating the CICS Region JCL - Complete.
2018-10-03 12:00:48 Removing temporary files from zFS - Complete.
2018-10-03 12:00:49 DFH_REGION_CMCIPOPT : 32001
2018-10-03 12:00:49 DFH_REGION_HTTP : 26020
2018-10-03 12:00:49 DFH_REGION_APPLID : CICPD002
2018-10-03 12:00:49 DFH_CICS_TYPE : SMSS
2018-10-03 12:00:49 DFH_CICSGRP : DEFAULT
2018-10-03 12:00:49 DFH_REGION_IPIC : NO
2018-10-03 12:00:49 DFH_JVM_DEBUG : NO
2018-10-03 12:00:49 DFH_REGION_ZFS_DIRECTORY : /u/user01/mount/CICPD002
2018-10-03 12:00:49 DFH_REGION_HOSTNAME : SYSTEM01.IBM.COM
2018-10-03 12:00:49 DB2_REGISTRY_NAME : REG-DB2_TDB200
2018-10-03 12:00:49 DFH_REGION_HTTPS : 26022
2018-10-03 12:00:49 Created container CICS_CICPD002 with id 4743267a-5321-4420-86bb-a7e8040a1674 on system MV01.
2018-10-03 12:00:51 Performing start on container CICS_CICPD002.
2018-10-03 12:00:54 Checking zFS is mounted - Complete.
2018-10-03 12:01:01 Starting the CICS Region - Complete.
2018-10-03 12:01:02 Started container CICS_CICPD002.

```

If the properties are all correct, the output from the **zospt run** command ends with a message that says the container is started. Your provisioned container is ready to use. The output also contains the details of the name of the registry subsystem that is specified in the **link** option. An IBM Db2, IBM MQ, or IPIC connection definition is installed on the provisioned system (depending on the subsystem type of the linking container) but no checks are made to ensure that the connection is acquired or successful. To resolve any problems when you provision a container, see Chapter 18, “Troubleshooting z/OS Provisioning Toolkit,” on page 129.

Using the --quiet option

The **--quiet** option is provided to make it easier to automate provisioning when you are using z/OS PT. By specifying the **--quiet** option, only the container name is written to STDOUT during the **zospt run** command. The container name can then be stored for use in further z/OS PT commands. For more information about using z/OS PT in automation, see Chapter 17, “Automating provisioning with z/OS Provisioning Toolkit,” on page 127.

```
zospt run imageName --quiet
```

This command runs an image in quiet mode.

Exploring further

Your provisioned container is ready to use. By using z/OS PT commands, you can inspect, stop, restart, and deprovision your container. For more information, see

Chapter 14, “Managing containers created by z/OS Provisioning Toolkit,” on page 113. For a complete syntax description, see Chapter 19, “The **zospt** command syntax,” on page 135.

Link scenarios supported by z/OS Provisioning Toolkit

A range of link scenarios that define and install a connection to another subsystem are supported by z/OS PT. Use the **link** option on the **zospt run** command to try these scenarios.

Linking containers

For more information about the **zospt run --link** command itself, see “**--link** containerName:alias | containerId:alias” on page 137.

For more information about how to run the **zospt run --link** command, and for some example output from the command, see “Using the **--link** option” on page 52.

Linking a provisioned CICS image to a registered IBM Db2 container

- A Db2 connection is defined and installed in the provisioned CICS container.
- If the provisioned CICS container includes a Liberty JVM server, a JDBC type 2 connection is established.

Linking a provisioned CICS image to a registered IBM MQ container

- An IBM MQ connection is defined and installed in the provisioned CICS container.

Linking a provisioned CICS image to a registered IBM Db2 container, and a registered IBM MQ container

- A Db2 connection and an IBM MQ connection are defined and installed in the provisioned CICS container.
- If the provisioned CICS container includes a Liberty JVM server, a JDBC type 2 connection is established.

Linking a provisioned instance of z/OS Connect EE to a provisioned or registered CICS container

- The template that is used for provisioning the CICS container must specify the following properties:
 - **DFH_REGION_IPIC=YES** so that CICS is listening on an IPIC port and the provisioned z/OS Connect EE container can determine the IPIC port to which it needs to connect.
 - **DFH_REGION_SEC=NO**, which means security for the CICS region is disabled, and all other security options are ignored.
- An IPIC Connection is defined and installed in the provisioned z/OS Connect EE container (by using the CICS Service Provider).
- If you specify an *alias* on the **link** command, the *alias* name (for example, *cicsConn*) must match the value of the *connectionRef* specified in z/OS Connect EE for building a service archive:

```
<zosconnect_cicsIpicConnection id="cicsConn"
host="SYSTEM01.HURSLEY.IBM.COM"
port="$${instance-ZCON_CICS_IPIC_PORT}" />
```

Managing z/OS Provisioning Toolkit images

After you build an image with the **zospt build** command, you can manage them using other z/OS PT commands. You can see which images are available, or run or remove an image.

List images

Use the **zospt images** command to list all images:

```
2019-01-14 15:23:53 IBM z/OS Provisioning Toolkit V1.1.4
IMAGE NAME          CREATED          SIZE
cics_52             2019-01-14T06:45:51 10240B
cics_53             2019-01-14T06:45:51 10240B
cics_54             2019-01-14T06:45:51 10240B
cics_55             2019-01-14T06:45:51 10240B
cics_55_json        2019-01-14T06:45:51 10240B
cics_55_liberty     2019-01-14T06:45:51 10240B
cics_55_nodejs_ivp  2019-01-14T06:45:51 20480B
cics_55_soap        2019-01-14T06:45:51 10240B
cics_55_zosconnect_v3r0 2019-01-14T06:45:51 10240B
cics_async_api_sample 2019-01-14T06:45:51 235520B
cics_getting_started_cloud 2019-01-14T06:45:51 10240B
cics_getting_started_workflow 2019-01-14T06:45:51 10240B
cmas_55             2019-01-14T06:45:51 10240B
wui_55              2019-01-14T06:45:51 10240B
zosconnect_v2r0     2019-01-14T06:45:51 10240B
zosconnect_v3r0     2019-01-14T06:45:51 10240B
```

Inspect images

Use the **zospt inspect** command to see more details about the image you built:

```
zospt inspect imageName
```

The following text is some example output from the **zospt inspect** command:

```
{
  "name": "cics_55",
  "created": "Wed Jan 14 06:45:51 GMT 2019",
  "size": "10240B",
  "FROM": "scratch",
  "ENV": {"ZOSMF_TEMPLATE": "cics_55"}
}
```

For more information about the **size**, **FROM**, and **ENV** property names, see “Building an image with z/OS Provisioning Toolkit” on page 49.

Remove images

When one or more images are no longer needed, use the **zospt rmi** command to remove them. This example command shows how to remove two images:

```
zospt rmi imageName1 imageName2
```

The following output is returned:

```
2017-09-12 13:29:04 IBM z/OS Provisioning Toolkit V1.1.0
2017-09-12 13:29:04 Deleted image  imageName1.
2017-09-12 13:29:04 Deleted image  imageName2.
```

Exploring further

- For a complete syntax description and any conditions of use, see Chapter 19, “The **zospt** command syntax,” on page 135.
- For more information about managing containers that are created by z/OS PT, see Chapter 14, “Managing containers created by z/OS Provisioning Toolkit,” on page 113.

Chapter 12. Provisioning CICS with z/OS Provisioning Toolkit

With z/OS PT, you can rapidly provision regions of CICS Transaction Server for z/OS. Images for provisioning commonly used CICS environments, such as CICS TS with embedded z/OS Connect Enterprise Edition, CICS TS with a Liberty JVM server, or standard CICS TS for traditional COBOL development, are provided.

Configuration prerequisites for using z/OS Provisioning Toolkit with CICS

In addition to the requirements for z/OS PT, some system configuration is needed to enable z/OS PT to provision CICS regions.

System requirements

For any specific system requirements to be able to provision CICS regions by using z/OS PT, see the z/OS Provisioning Toolkit Software Compatibility Report.

Configuration requirements

Ensure that a data class that is compatible with creating zFS directories exists. The following table shows an example of a valid data class configuration:

Override Space	No
Volume Count	15
Data set Name Type	EXTENDED
If Extended	PREFERRED
Extended Addressability	Yes
Record Access Bias	User
Space Constraint Relief	No
Reuse	No
Initial Load	Recovery
RLS CF Cache Value	All
RLS Above the 2-GB Bar	No
Extent Constraint Removal	No
CA Reclaim	Yes
Log Replicate	No
System Determined Blocksize	No

Options for issuing console commands

To use REXX scripts to issue console commands to start and stop CICS, check its status and delete logstreams, no additional configuration is required.

To use the API that the z/OSMF z/OS console services provide to issue commands and detect unsolicited messages, see the configuration information for z/OS console services in Chapter 9, “Preparing to use z/OS Provisioning Toolkit with

IBM Cloud Provisioning and Management for z/OS,” on page 41.

Message suppression

If you are using z/OS console services, ensure that message suppression is switched off for the following messages:

- DFHSI1517: Control is being given to CICS.
- EYUNL0099I: LRT initialization complete.
- EYUXL0010I: CMAS initialization complete .
- DFHKE1799: TERMINATION OF CICS IS COMPLETE.
- IXG661I: SETLOGR FORCE DELETE PROCESSED SUCCESSFULLY FOR LOGSTREAM=resourcename.

These messages are used by the CICS workflow. They must be available to send to an extended multiple console support (MCS) console that the workflow uses to determine whether CICS is started or stopped, and whether a CICS region's log streams were deleted successfully. For more information about suppressing messages, see Suppressing messages in z/OS MVS Planning: Operations.

User ID authorization

User IDs that run the different steps of the provisioning process must have the correct security authorizations. For more information, see “Security requirements for provisioning CICS with z/OS Provisioning Toolkit” on page 72.

Overview of supported CICS types

You can use configuration properties to customize the workflows that are provided with z/OS PT for provisioning CICS regions. To simplify the process of customizing the CICS regions for specific roles, you can set a property that describes the type of CICS region you want to provision. To set the CICS type, specify the DFH_CICS_TYPE property when you configure the template.

For more information about other configuration properties, see “Configuration properties for CICS images” on page 75.

CMAS

The template is used to provision CICSplex SM address spaces (CMASs). Each provision from the template creates a CMAS, and defines a CICSplex in the data repository of the CMAS. The name of the CICSplex can be provided, or can be generated as part of the provisioning process. After you provision a CMAS, it is then possible to use z/OS PT to provision one or more WUI or MAS regions to the defined CICSplex. If you specify DFH_CICS_TYPE=CMAS, extra properties can be set to customize the CMAS configuration, as described in “CICSplex SM properties” on page 87. It is not possible to add a provisioned CMAS to an existing CICSplex or create a CMAS to CMAS link from one CMAS to another by using z/OS PT.

MAS

The template is used to provision managed application system (MAS) regions. A MAS region is managed by CICSplex SM and belongs to a CICSplex. Each provision creates a MAS region and connects it into a CICSplex. The name of the CICSplex to connect the MAS into is specified as part of the template configuration, as described in “CICSplex SM properties” on page 87. The MAS can

either connect to a CICSplex defined in a CMAS that is provisioned by z/OS PT, or can be connected to an existing CICSplex that was defined outside of z/OS PT.

SMSS

The template is used to provision CICS regions that do not participate in a CICSplex, but that can be managed through CICS Explorer®. To enable CICS Explorer to be connected to a provisioned CICS region, a CMCI port is allocated to the CICS region. Also, the CICS resources that are required to enable the CMCI port are defined in the system definition data set (CSD) of the region and are installed during the start of the CICS region.

Unmanaged

The template is used to provision CICS regions that do not participate in a CICSplex and cannot be managed through CICS Explorer. No ports are allocated to the CICS region unless they are specified as required through other configuration properties on the template.

WUI

The template is used to provision Web User Interface (WUI) regions. A WUI region enables a CICSplex to be managed through CICS Explorer or through a Web User Interface that the region provides. Each WUI region is managed by CICSplex SM and participates in a CICSplex. The name of the CICSplex to connect the WUI into is specified as part of the template configuration, as described in “CICSplex SM properties” on page 87. A WUI region must be connected to a CMAS region at the same CICS version. The WUI can connect either to a CICSplex defined in a CMAS that is provisioned by using z/OS PT, or to an existing CICSplex that was defined without using z/OS PT.

To enable CICS Explorer to be connected to a provisioned WUI region, a CMCI port is allocated to the WUI region, and the CICS resources that are required to enable the CMCI port are defined in the CSD of the region. The CICS resources are then installed during the start of the region. A second port is allocated to the WUI region to enable the WUI to be provided for managing the CICSplex.

Adding a CICS template by using IBM Cloud Provisioning and Management for z/OS

This scenario walks through the steps to add a CICS Cloud Provisioning template to z/OSMF and test it.

This scenario uses a richer set of z/OSMF capabilities than the simple Cloud Provisioning scenario “Getting started with CICS by using IBM Cloud Provisioning and Management for z/OS” on page 34. You set up the template to use the following functionality in z/OSMF:

- Run workflows under a set of authorized provisioning user IDs, rather than only a single user ID.
- Allocate APPLIDs dynamically from resource pools.
- Allocate ports dynamically from port ranges.
- Add and remove security profiles dynamically to set up security for provisioned environments such as CICS.

Before you begin

See the requirements in Chapter 10, “Adding an IBM Cloud Provisioning and Management for z/OS template for use with z/OS Provisioning Toolkit,” on page 45. You also need the following resources:

- A range of APPLIDs in a resource pool. These APPLIDs can then be allocated dynamically when you run z/OS PT.
- A set of user IDs (agreed with your security administrator) that have sufficient authority to provision a CICS environment. For details, see “Security requirements for provisioning CICS with z/OS Provisioning Toolkit” on page 72.
- A range of ports defined in the z/OSMF Network Configuration Assistant for use with templates. Ports can then be allocated to the provisioned CICS regions. For details, see Create a port allocation range in z/OSMF Online Help.

For an introduction to the steps to add and verify a template, and to the interfaces used, refer to the simple Cloud Provisioning scenario, as described in “Getting started with CICS by using IBM Cloud Provisioning and Management for z/OS” on page 34.

About this task

This scenario provisions a CICS TS V5.5 region from a template called `cics_55`. The following steps summarize the procedure:

- Optionally, copy the template (if you want more than one variant of the template).
- Configure the properties that z/OSMF uses to process the template.
- Add the Cloud Provisioning template into z/OSMF.
- Approve the template for use.
- Associate the template with a domain and tenant.
- Configure the network resource pool.
- Set up the environment for z/OS PT.
- Use z/OS PT to provision CICS.
- Publish the template.

Procedure

1. Optional: If you want more than one variant of the template, copy the directory for the CICS template. For example, copy the `/cics_55` directory in the `/templates` directory. Give the new directory a different name but keep it in the `/templates` directory.
2. Configure the `cics.properties` file that z/OSMF uses to process the template:
 - z/OS PT includes sample `cics.properties` files for provisioning CICS TS V5.2 to CICS TS V5.5 regions (`zospt/templates/cics_5n/cics.properties`). The files are in code page ISO-8859-1.
 - Add any additional or alternative properties. See “Configuration properties for CICS images” on page 75.
 - Make sure that you include properties for the set of user IDs that has sufficient authority to provision a CICS environment.
 - Do not set `DFH_REGION_APPLID` because this property prevents z/OS PT from accessing the pool of APPLID resources.
3. Add the template to z/OSMF through the z/OSMF web interface.

For a similar step that includes screen captures, see “Getting started with CICS by using IBM Cloud Provisioning and Management for z/OS” on page 34.

- a. Enter the full path of the manifest file for the template in **Template Source file**, for example, `zospt/templates/cics_55/cics.mf`. z/OSMF must be able to read from this directory tree.
- b. Enter the template name in **Template Name**, for example, `cics_55`.

Note: The template name is important. It must match the name that is used in an image that is built with **zospt**.

After you click **OK**, the template is loaded into z/OSMF.

4. Optional: If a “Pending Security Update” message is displayed, click **Actions > Refresh Template**.
5. Ensure that approvals are completed by all users that are named as approvers for the template, or for steps in the template. For example, you might require approvals from the following users:
 - A security administrator, if provisioning dynamically creates SAF resources.
 - A z/OS systems programmer, if provisioning dynamically creates log streams.
 - A capacity planner.
6. Associate the template with a tenant in the domain in the z/OSMF web interface.

For a similar step that includes screen captures, see “Getting started with CICS by using IBM Cloud Provisioning and Management for z/OS” on page 34. Select and specify the following properties and values:

Property	Setting
Software services instance name prefix	Select the Use SNA APPLID check box (which automatically selects the Create network resource pool check box). Note: To use this template with z/OS PT, do not select the Create workload management pool check box.
Maximum number of software services instances.	Specify the maximum number of software services instances that can be allocated.
Maximum number of software services instances for a user	Specify the maximum number of software services instances that can be allocated by an individual user.
System selection for provisioning	Choose the system in the sysplex into which you want to provision the CICS regions. The default is to provision into the same system that is running z/OSMF. For more information about configuring z/OSMF for provisioning across different systems, see z/OSMF product documentation. Note: To use this template with z/OS PT, do not select the Prompt user for system option.

7. Configure the network resource pool by using z/OSMF.
 - a. In the Configuration Assistant, select **Manage z/OS Cloud Configuration** then click **Proceed**.
 - b. Select **Cloud Domain** and click **Proceed**.
 - c. Select the pool for your template in the Network Resource Pools table (the pool is named `domainname.tenantname.templateName`).
 - d. Click **Actions > Modify**.
 - e. On the **Attributes** tab, select **Is complete**.

- f. On the **Port Allocation** tab, select the range of ports that was agreed with your network administrator for use in this template.
 - g. On the **SNA Application Name Range** tab, select the range of APPLIDs that was agreed with your network administrator.
 - h. Click **Save**. Your network resource pool is configured. For more information, see z/OSMF product documentation.
8. Ensure that you have access to the pools of ports and APPLIDs. The template is now ready to test with the **zospt** utility.
 9. Set up the environment for z/OS PT. Set the following environment variables:
 - `export zospt_domain=domain_name`
 - `export zospt_tenant=tenant_name`
 10. Use z/OS PT and the **zospt build** command to build an image that references the template. See “Building an image with z/OS Provisioning Toolkit” on page 49.
You might set *imageName* to be the same as *templateName*, for example `cics_55`.
 11. Use z/OS PT and the **zospt run** command to provision a CICS container from an image that uses your template. Issue the following command, using the `--draft` option to run the template in *Draft approved* state, and where *imageName* is the name of the image you built:
`zospt run imageName --draft`

For example:

```
zospt run cics_55 --draft
```

If parallel provisioning is enabled in the workflow for CICS templates that use Cloud Provisioning to provision a CICS region, some steps run in parallel so that provisioning completes faster. Output includes the message `This workflow contains parallel steps`. See “Enabling faster provisioning by using parallel processing” on page 63.

When provisioning completes, a message that the container is started is displayed. Your provisioned container is ready to use.

12. Optional: If an error is reported, see “Troubleshooting CICS provisioning with z/OS Provisioning Toolkit” on page 97 and Chapter 18, “Troubleshooting z/OS Provisioning Toolkit,” on page 129. If you need to change the properties for the workflow to resolve the error, adjust the properties, refresh the template and retry:
 - a. Update the properties file. For example, update `zospt/templates/cics_55/cics.properties`.
 - b. Refresh the template from the z/OSMF web interface. For details of this step, including screen captures, see “Getting started with CICS by using IBM Cloud Provisioning and Management for z/OS” on page 34).
 - c. Reapprove the template for use.
 - d. Issue the following command to remove the container that failed to provision correctly.
`zospt rm --force containerName`
If parallel provisioning is enabled in the workflow for CICS templates that use Cloud Provisioning to deprovision a CICS region, some steps run in parallel so that deprovisioning completes faster. Output includes the message `This workflow contains parallel steps`. See “Enabling faster provisioning by using parallel processing” on page 63.
 - e. Retry the **zospt run** command, as described in step 11.

13. Optional: Publish the template for every user in the tenant to use.
After the template is published, you use the **zospt run** command without the **--draft** option.

What to do next

Use z/OS PT to build images from the template that you loaded. See “Building an image with z/OS Provisioning Toolkit” on page 49.

Enabling faster provisioning by using parallel processing

For faster provisioning and deprovisioning, you can enable parallel processing for relevant steps in the z/OSMF workflow. In z/OS PT, such parallel processing is available for the workflows that are provided with the CICS templates that use Cloud Provisioning to provision a CICS region.

Before you begin

- For z/OSMF V2R3, the PTFs for APARs PI92305 and PI99364 must be applied.
- For z/OSMF V2R2, the PTFs for APARs PI95358 and PI99364 must be applied.

See PI92305: Z/OSMF Workflow supports parallel workflow, PI99364: Collection of fixes for z/OSMF workflows, and PI95358: Z/OSMF Workflow enhancement.

About this task

If the prerequisite PTFs are applied to z/OSMF, you can enable parallel processing in the workflow for CICS templates that use Cloud Provisioning to provision a CICS region. Suitable steps in the workflow can run in parallel so that provisioning or deprovisioning runs faster. Steps are suitable if they do not need to complete in a specified sequence. Parallel processing is supported with the **zospt run** and **zospt rm** commands and templates that use the CICS workflow.

When these z/OS PT commands run, the console output includes the message This workflow contains parallel steps.

You can enable parallel processing at any time. However, if it is not enabled when you create and test a template, the console output is easier to interpret if an error occurs. Therefore, you might enable parallel processing after you create and test a template and are ready to publish it.

For more information about parallel processing in z/OSMF, see Enabling automated steps for parallel processing in the z/OSMF Programming Guide.

If you have created and tested a template and it is ready to publish, you can also enable faster provisioning by turning off the validation. See the `VALIDATE_PARAMETERS` property in “Provisioning process customizations” on page 88.

Procedure

1. Locate the extension file for parallel processing in the `zospt/workflows/cics/extensions/` directory: `zospt/workflows/cics/extensions/run_steps_in_parallel.xml`
2. Edit the file and uncomment the `<parallelSteps>true</parallelSteps>` statement.

3. Refresh the template from the z/OSMF web interface. For details of this step, including screen captures, see “Getting started with CICS by using IBM Cloud Provisioning and Management for z/OS” on page 34).
4. Reapprove the template for use.

What to do next

Run z/OS PT to provision or deprovision a CICS container. See “Adding a CICS template by using IBM Cloud Provisioning and Management for z/OS” on page 59.

Using dynamic port allocation in a template

You can specify that a template uses dynamic port allocation, and control which range of ports are used for dynamic allocation by setting a usage type. You can use this to provision a CICS region that obtains dynamically allocated ports from different port ranges, depending on the use of the requested port.

About this task

When you add a CICS template by using IBM Cloud Provisioning and Management for z/OS, you can use the z/OSMF ability to allocate ports dynamically from port ranges. Typically, a network administrator uses z/OSMF to define a range of ports for use with templates and supplies this information to the system administrator who creates the templates.

By default, when a container is provisioned, any available port is allocated from the port allocation ranges that were defined for use with a template.

However, you can control which range of ports are used for dynamic allocation by setting a usage type. Typically, the network administrator uses z/OSMF to set a usage type when they define each port allocation range for use with templates. When you use z/OS PT to create a template, you use a configuration property to specify the required usage type for a type of port. When a container that requires such a port is provisioned, any available port is allocated from the port allocation range that this usage type identifies.

z/OS PT supports the following configuration properties for port usage types.

- `DFH_JVM_DEBUG_USAGETYPE`. Sets the usage type for the port that a Java debugger can be connected to.
- `DFH_REGION_CMCIPTPORT_USAGETYPE`. Sets the usage type for a CMCI port (used for connection to CICS Explorer).
- `DFH_REGION_HTTP_USAGETYPE`. Sets the usage type for an HTTP port.
- `DFH_REGION_HTTPS_USAGETYPE`. Sets the usage type for an HTTPS port.
- `DFH_REGION_IPIC_USAGETYPE`. Sets the usage type for an IPIC port.
- `DFH_REGION_TCPIP_USAGETYPE`. Sets the usage type for the CICSplex SM WUI port.

You can use specific port ranges for different purposes, for example, you might use different port ranges for different categories of CICS applications.

You can use several approaches to use specific port ranges for different purposes:

- Create a number of templates, where each one is configured for a different usage type. Remember that with this approach, each template requires a setup and approval process.

- Create a template that is configured for multiple usage types, then configure and build different images for the different usage types.
- Create a template that is configured for one or more usage types. Either configure and build different images for the different usage types, or configure and build different images that override the usage type. You need to specify that the relevant usage type configuration properties can be overridden in an image. See “Modifying the set of CICS configuration properties that can be overridden” on page 93.

The following procedure outlines how to use specific port ranges for different purposes by creating a template that is configured for multiple usage types.

Procedure

1. Use z/OSMF to create the required port allocation ranges and set a usage type for each range. In z/OSMF, Usage Type is an optional field, but you must set it here so that you can use it in the template configuration. You would usually set a unique usage type for each port allocation range. Typically, a network administrator completes this step. For more information, see Create a port allocation range in z/OSMF Online Help.
2. Use z/OS PT to create a new template, as described in “Adding a CICS template by using IBM Cloud Provisioning and Management for z/OS” on page 59.
 - a. When you configure the `cics.properties` file, add one or more of the following configuration properties and specify the usage type for each one. The usage type name must match the Usage Type field set in z/OSMF for the port allocation range that you require.
 - `DFH_JVM_DEBUG_USAGETYPE`. Sets the usage type for the port that a Java debugger can be connected to.
 - `DFH_REGION_CMCIPTORT_USAGETYPE`. Sets the usage type for a CMCI port (used for connection to CICS Explorer).
 - `DFH_REGION_HTTP_USAGETYPE`. Sets the usage type for an HTTP port.
 - `DFH_REGION_HTTPS_USAGETYPE`. Sets the usage type for an HTTPS port.
 - `DFH_REGION_IPIC_USAGETYPE`. Sets the usage type for an IPIC port.
 - `DFH_REGION_TCPIP_USAGETYPE`. Sets the usage type for the CICSplex SM WUI port.

A valid usage type name is 1-32 characters and the following valid characters: A-Z a-z 0-9 For example:

```
# Set networking variables
DFH_REGION_HTTP=YES
DFH_REGION_HTTP_USAGETYPE=http

DFH_REGION_HTTPS=YES
DFH_REGION_HTTPS_USAGETYPE=https

DFH_JVM_DEBUG=YES
DFH_JVM_DEBUG_USAGETYPE=jvm

DFH_REGION_TCPIP=YES
DFH_REGION_TCPIP_USAGETYPE=WUIport

DFH_REGION_CMCIPTORT_USAGETYPE=CMCIport

DFH_REGION_IPIC=YES
DFH_REGION_IPIC_USAGETYPE=ipic
```

When you use the z/OSMF web interface to add the template to z/OSMF and associate the template with a tenant in the domain, a network resource pool is generated.

3. Use the Network Configuration Assistant in z/OSMF to modify the network resource pool to specify the port allocation ranges that are associated with the tenant and the template. Typically, a network administrator completes this step. For more information, see *Modify the network resource pool in z/OSMF Online Help*.
4. Use z/OS PT to build and run the image. See “Building an image with z/OS Provisioning Toolkit” on page 49.

If the following message is displayed during provisioning, ensure that you used the correct name for the usage type in the `cics.properties` file and that this name is set for the port allocation range in the Usage Type field in z/OSMF.

Unable to find free port

Be aware that the same message might be displayed if a port allocation range is not associated with the template, or simply because all the ports in the valid port ranges are in use.

5. Repeat step 4 to create more images that specify other usage types, and so use different port allocation ranges.

Including resources in an image

With z/OS PT, you can include CICS bundles, a JVM server resource, CICS resource definitions, program binary files, and zFS files in an image.

CICS bundles

A CICS bundle is a directory that contains artifacts and a manifest that describes the bundle and its dependencies. CICS bundles provide a way to group and manage related resources. CICS bundles also provide versioning for the management of resource updates, and can declare dependencies on other resources outside the bundle. Application developers can use CICS bundles for application packaging and deployment, business events, and services. System programmers can use CICS bundles for policies. For more information about defining CICS bundles, see *Defining CICS bundles in CICS TS product documentation*.

CICS bundles can contain policies, OSGi, or Liberty Java application code and resource definitions, including LIBRARY resources for local applications. By including a CICS bundle in an image, you can provision a CICS region with a specific configuration, or with specific applications already available.

JVM server resource

You can either define the type of JVM server you want in the `zosptfile` by using the **DFH_REGION_JVMSEVER** environment property, or define a JVMSEVER resource in the CICS bundle. If you choose to use the **DFH_REGION_JVMSEVER** environment property, ports can be dynamically allocated for the JVM server during provisioning. For more information about the possible values of **DFH_REGION_JVMSEVER** in the `zosptfile`, see “JVM configuration options” on page 80.

CICS resource definitions

With z/OS PT, you can include CICS resource definitions in a CICS bundle, or include them by providing DFHCSDUP input files in an image. When you include CICS resource definitions in an image, the provisioned CICS region will have the resource definitions defined in its system definition data set (CSD).

If you include CICS resource definitions in a CICS bundle, the bundle is installed automatically into the provisioned CICS region. However, if you include CICS resource definitions by providing DFHCSDUP input files in an image, to install the resulting CSD group during CICS startup, you must add the group to an appropriate group list.

CICS application program binary files

With z/OS PT, you can include program binary files for application programming languages that CICS supports, such as compiled COBOL or PL/I. When you include program binary files in an image, you can provision a CICS region with a LIBRARY resource installed in it that references a library that contains those program binary files. You might use this feature in a development environment to provision a CICS region with a unit test library to test code.

z/OS PT supports load modules and program objects. During provisioning, a PDSE is created. Any load modules or program objects in the image are copied into the PDSE. Program objects are created automatically when load modules are copied into a PDSE.

zFS files

You can copy a file or directory in zFS to an image. The file must be in the same directory as the zosptfile, or in one of its subdirectories. For example, you might copy in a pipeline configuration file to provide a service provider pipeline. For more information, see “Building an image with z/OS Provisioning Toolkit” on page 49.

Setting environment variables

If the system programmer allows users of the templates to override any CICS configuration properties, you can specify environment variables in the image to configure the provisioned environment. You can customize some of the properties of a provisioned environment by adding environment variable definitions to the zosptfile for the image, which can override values that are specified in the properties file of the template that is named in the image. For example, you can specify the **DFH_REGION_JVMSEVER** property to provision a Liberty JVM server. For more information about which properties you can add to the zosptfile for an image, see “Configuration properties for CICS images” on page 75.

Including a CICS bundle in your image

With z/OS PT, you can include a CICS bundle, a JVM server resource, or both, in your image. CICS bundles can contain policies, OSGi, or Liberty Java application code and resource definitions, including LIBRARY resources for local applications. By including a CICS bundle in an image, you can provision a CICS region with a specific configuration, or with specific applications already available.

CICS bundles

A CICS bundle is a directory that contains artifacts and a manifest that describes the bundle and its dependencies. CICS bundles provide a way to group and manage related resources. CICS bundles also provide versioning for the management of resource updates, and can declare dependencies on other resources outside the bundle. Application developers can use CICS bundles for application packaging and deployment, business events, and services. System programmers can use CICS bundles for policies. For more information about defining CICS bundles, see *Defining CICS bundles in CICS TS product documentation*.

How to include a CICS bundle in your image

You can include a CICS bundle in your image by updating the `zosptfile` to include `COPY` statements that list the CICS bundles to add. All the CICS bundle artifacts must be copied to the image source directory, which must contain a `zosptfile` so that you can build the new image by using the **zospt build** command. For more information about how to build an image with z/OS PT, see “Building an image with z/OS Provisioning Toolkit” on page 49.

CICS bundle installation

All CICS bundles that are installed are named as `DFH$Bn`, where *n* is a number in the range 0 - 999.

JVM server creation

You can either define the type of JVM server you want in the `zosptfile` by using the `DFH_REGION_JVMSEVER` environment property, or define a `JVMSEVER` resource in the CICS bundle. If you choose to use the `DFH_REGION_JVMSEVER` environment property, ports can be dynamically allocated for the JVM server during provisioning. For more information about the possible values of `DFH_REGION_JVMSEVER` in the `zosptfile`, see “JVM configuration options” on page 80.

Include a CICS bundle to provision a CICS region with a Liberty application

1. Create your Liberty application (for example, a CICS TSQ sample).
2. Create a CICS bundle to include the Liberty application you created.
3. Export the CICS bundle to a UNIX System Services directory.
4. Create a `zosptfile` in the same directory as the exported CICS bundle (for example, the `bundleLiberty` directory). Update the `zosptfile` to copy all required CICS bundles into the image to a `bundles` directory. The bundles must be in a directory called `bundles`, otherwise they will not be installed in the provisioned CICS regions. This example shows how your `zosptfile` might look:

```
FROM scratch
ENV ZOSMF_TEMPLATE=cics_55
ENV DFH_REGION_JVMSEVER=Liberty
```

```
COPY com.ibm.server.examples.wlp.tsq.bundle bundles/com.ibm.server.examples.wlp.tsq.bundle
```

5. Build the `zosptfile` into an image called (in this example) *myTSQSampleApp*, by using the following command:

```
zospt build -t myTSQSampleApp bundleLiberty
```
6. Run the image to create a container by using the following command:

```
zospt run myTSQSampleApp
```

If the properties are all correct, the CICS region is provisioned with a configured JVM server named DFHWLP, a CICS bundle resource installed, and the running TSQ sample in the JVM server.

Including program binary files in your image

With z/OS PT, you can include program binary files for application programming languages that CICS supports, such as compiled COBOL or PL/I. When you include program binary files in an image, you can provision a CICS region with a LIBRARY resource installed in it that references a library that contains those program binary files.

About this task

You might include program binary files in an image in a development environment to provision a CICS region with a unit test library to test code.

The examples in this procedure use the supplied sample image `cics_async_api_sample`, which is based on the `cics_getting_started_workflow` image. The `samples/cics_async_api_sample` directory includes compiled source code for a CICS asynchronous API application. Running the sample image provisions a CICS region with this application.

For more information about the sample application, see *IBM CICS Asynchronous API: Concurrent Processing Made Simple*. For the application source, see *CICS Asynchronous API Redbooks Examples in GitHub*.

Procedure

1. Compile your programs, for example your COBOL programs.
2. Either continue with steps 3 to 6 on page 70, or use the `copy_build_run.sh` script that is supplied in `samples/cics_async_api_sample`. To use the script, follow the instructions in the comments. This script copies program binary files to a suitable zFS directory, builds them into an image, then runs that image.
3. Copy the compiled program binary files to the zFS directory where you intend to create your `zosptfile`, or a subdirectory of that directory. For example, to copy the members of a data set named `USER.ASYNCAPI.LOAD` to the `/u/userid/myimage/application_binaries` directory, use the following command:

```
cp -X "'/USER.ASYNCAPI.LOAD'" /u/userid/myimage/application_binaries
```

For more information about copying files to zFS, see *Copying executable modules between MVS data sets and the z/OS UNIX file system* in the *z/OS UNIX System Services User's Guide*.

4. Create a `zosptfile` in the zFS directory that you planned to use, for example `/u/userid/myimage`. You can copy the example `zosptfile` in `samples/cics_async_api_sample`.
 - a. Update the `zosptfile` to define the image on which this image is based. For example, specify `FROM cics_getting_started_workflow`.
 - b. Update the `zosptfile` to copy any program objects into the image to a `binaries` directory. The program binary files must be in a directory called `binaries`, otherwise they will not be installed in the provisioned CICS regions.

This example shows how your zosptfile might look:

```
FROM cics_getting_started_workflow
```

```
COPY application_binaries binaries
```

5. From the zFS directory that contains the zosptfile for your image, use the **zospt build** command to build the image. For example, to build an image named `cics_async_api_sample`, issue the following command from the `/u/userid/myimage` directory:

```
zospt build -t cics_async_api_sample .
```

6. Use the **zospt run** command to run the image and create a container. For example, use the following command:

```
zospt run cics_async_api_sample
```

If the properties are all correct, the CICS region is provisioned with a LIBRARY resource installed in it that references a library that contains the program binary files. During provisioning, a PDSE data set is created. The data set ranking places this data set before the DFHRPL library.

Including CICS resource definitions in an image

With z/OS PT, you can include CICS resource definitions in a CICS bundle, or include them by providing DFHCSDUP input files in an image. When you include CICS resource definitions in an image, the provisioned CICS region will have the resource definitions defined in its system definition data set (CSD).

About this task

For information about including CICS resource definitions in a CICS bundle, see “Including a CICS bundle in your image” on page 67.

To include CICS resource definitions in an image, you specify resource definitions in files that are suitable input for the CICS system definition utility program DFHCSDUP. You might include CICS resource definitions by using DFHCSDUP input files in an image when you want to install more CICS resource definitions in the provisioned CICS regions and you do not want to use CICS bundles.

Note: To include CICS resource definitions by using DFHCSDUP input files in an image, each provisioned CICS region must have its own unique CSD created during provisioning. DFHCSDUP files in an image are not supported if CICS regions share a CSD.

The examples in this procedure use the supplied sample image `cics_async_api_sample`, which is based on the `cics_getting_started_workflow` image. The `samples/cics_async_api_sample` directory includes compiled source code for a CICS asynchronous API application. Running the sample image provisions a CICS region with this application.

The sample directory also includes a sample DFHCSDUP input file in `csd_definitions/appdefs`, for resource definitions defined into a CSD group named `ASYNCAPI`.

For more information about the sample application, see *IBM CICS Asynchronous API: Concurrent Processing Made Simple*. For the application source, see *CICS Asynchronous API Redbooks Examples* in GitHub.

Procedure

1. Specify the CICS resource definitions in one or more files, in EBCDIC format, that are suitable for use with the DFHCSDUP utility program. These files can be files on zFS or partitioned data set members. See DFHCSDUP: syntax and command rules in CICS TS product documentation.
2. Copy the files to the zFS directory where you intend to create your zosptfile, or a subdirectory of that directory. For example, to copy a data set member named USER.ASYNCAPI.DEFS(APPDEFS) to the /u/userid/myimage/csd_definitions directory, use the following command:

```
cp "'/USER.ASYNCAPI.DEFS(APPDEFS)'" /u/userid/myimage/csd_definitions
```

For more information about copying files to zFS, see Copying a PDS or PDSE to a z/OS UNIX directory in the z/OS UNIX System Services User's Guide.

3. Create a zosptfile in the zFS directory that you planned to use, for example /u/userid/myimage. You can copy the example zosptfile in samples/cics_async_api_sample.
 - a. Update the zosptfile to define the image on which this image is based. For example, specify FROM cics_getting_started_workflow.
 - b. Update the zosptfile to copy the CICS resource definition files into the image to a csddefs directory. The files must be in a directory named csddefs, otherwise they will not be added to the CSDs for the provisioned CICS regions.

This example shows how your zosptfile might look:

```
FROM cics_getting_started_workflow
```

```
COPY csd_definitions csddefs
```

4. From the zFS directory that contains the zosptfile for your image, use the **zospt build** command to build the image. For example, to build an image named cics_async_api_sample, issue the following command from the /u/userid/myimage directory:

```
zospt build -t cics_async_api_sample .
```
5. Use the **zospt run** command to run the image and create a container. For example, use the following command:

```
zospt run cics_async_api_sample
```

If the properties are all correct, the CICS region is provisioned with the resource definitions defined in the CSD. If you used the sample DFHCSDUP input file in csd_definitions/appdefs, the resource definitions are defined into a CSD group named ASYNCAPI.

What to do next

To install the CSD group during CICS startup, you must add the group to an appropriate group list.

Security requirements for provisioning CICS with z/OS Provisioning Toolkit

The user IDs that z/OSMF uses to provision regions with z/OS PT require certain permissions to run steps of the provisioning process.

User ID	Permissions	Use
DFH_ADMIN_TSO	<ul style="list-style-type: none"> • READ DFH_CICS_HLQ.* • READ DFH_CPSM_HLQ.* • READ DFH_LE_HLQ.* • READ DFH_CICS_LICENSE_DATASET • ALTER (create/delete/read/write/update) DFH_REGION_HLQ.*. • Authority to run TSO LISTCAT command. • ALTER DFH_LOG_HLQ.* (if specified). • Authority to define log streams. • UPDATE DFH_ZOS_PROCLIB (add delete member). • READ DFH_ZOS_PROCLIB <i>name</i>.STCJOBS (if you are using STC jobs). • Authority to define MAS to the CMAS by using EYU9XDBT. • Authority to delete a MAS from the CMAS by using EYU9XDBT. 	<ul style="list-style-type: none"> • Validate that CICS data sets exist. • Create CICS region data sets. • Create the CICS log stream model. • Create CICS system definition data set (CSD) definitions. • Create the CICS region JCL. • Delete standard CICS data sets. • Create the MAS definition. • Delete the MAS definition.
DFH_ADMIN_ZFS	<ul style="list-style-type: none"> • r_x on DFH_CICS_USSHOME. • r_x on DFH_JAVA_HOME. • rwx DFH_ZFS_MOUNTPOINT. • r__ image location. • r_x on DFH_CICS_USSCONFIG. • ALTER (create/delete/read/write/update) DFH_REGION_HLQ.*. • Authority to run TSO MOUNT command. • Authority to run TSO UNMOUNT command. 	<ul style="list-style-type: none"> • Validate that z/OS File System (zFS) directories exist. • Define zFS. • Create JVM profile. • Create Liberty server.xml. • Add the image to zFS. • Check that zFS is mounted. • Delete zFS.
DFH_ADMIN_SECURITY	<ul style="list-style-type: none"> • Permission to run the REXX scripts for dynamic security and the commands that are issued by these REXX scripts. By default, the scripts exit immediately with return code zero and do not modify any security profiles. This default behavior allows for predefined security profiles to be used. The scripts can be modified to issue commands to dynamically create, update, and delete the security profiles that are needed for the provisioned CICS region. For more information, see “Modifying the CICS workflows to support dynamic security” on page 73. 	<ul style="list-style-type: none"> • Set up CICS security. • Delete CICS security configuration.

User ID	Permissions	Use
DFH_ADMIN_CONSOLE	<ul style="list-style-type: none"> Authority to open a console. When REXX scripts are used to issue console commands, the console names are the same as the APPLID of the provisioned CICS regions. When z/OS console services are used, the console name is XXXXXCN, where XXXXX is the first six characters of the provisioning user ID. When REXX scripts are used to issue console commands, authority to issue the TSO commands that these scripts require. When z/OS console services are used to issue console commands, authority to use z/OS console services (new user ID XXXXXCN where XXXXX are the first six characters of the provisioning user ID). See z/OS console services in the z/OSMF Programming Guide. Authority to issue a console START command. Authority to issue modify commands for CEMT P SHUT and CEMT P SHUT IMM. Authority to issue cancel commands. Authority to issue force commands. Authority to delete log streams. Authority to issue display commands (to display whether a CICS region is active, and to display details of the console that is used during provisioning to validate it is set up correctly to receive unsolicited messages). <p>For information about use of REXX scripts or z/OS console services, see “Workflow configuration variables” on page 89.</p>	<ul style="list-style-type: none"> Start the CICS region. Stop the CICS region. Delete the CICS log stream model.

Modifying the CICS workflows to support dynamic security

You can use REXX scripts in the workflow to enable the provision and deprovision process to dynamically configure the security profiles that are needed for a CICS region.

The CICS template provides workflows to provision and deprovision a CICS region. A step in the workflows runs a REXX script, which by default does not make any dynamic security changes. You can modify the REXX scripts so that security profiles can be added, modified, and deleted dynamically. In this way, the REXX scripts enable the provision and deprovision process to dynamically configure the security profiles that are needed for that CICS region.

About this task

Two example REXX scripts are provided with the workflows for provisioning and deprovisioning CICS regions. The scripts show example Resource Access Control Facility (RACF®) commands for configuring the necessary security profiles.

- `defineSecurity.rexx` contains example RACF definitions to dynamically create and configure the profiles that are needed by a CICS region.
- `deleteSecurity.rexx` contains example RACF definitions to delete the profiles that were dynamically created by the `defineSecurity.rexx` script.

The example REXX scripts exit immediately without issuing any commands. To use the scripts, the required security calls must be made before the exit statement.

Note:

Check the return code from each security command for an appropriate response in the `defineSecurity.rexx` script. If the REXX script exits with a nonzero return code, provisioning of the CICS region fails.

Edit `deleteSecurity.rexx` to undo any changes that are made by `defineSecurity.rexx`.

A failure in provisioning a CICS region might occur before, during, or after any dynamic security configuration process. The `deleteSecurity.rexx` script is always called during deprovisioning, so ensure that this script can complete successfully if it attempts to delete security profiles that do not exist. This situation might occur if the provision process failed before a profile was created, or if the deprovision process is retried and an earlier attempt already removed the security profiles.

Procedure

1. Locate the following REXX extensions for dynamic security in the `zospt/workflows/cics/extensions` directory:
 - `defineSecurity.rexx` - run during provision of a CICS region.
 - `deleteSecurity.rexx` - run during deprovision of a CICS region.
2. Locate the line at the start of each REXX script that issues `exit 0`. Make sure that any required security calls are made before this command.
3. Customize the scripts for your environment by using the sample REXX scripts as a guide.

Deprovisioning guidelines for regions in a CICSplex

You can provision MAS and WUI regions into an existing CICSplex. When you deprovision such regions, use the following guidelines to ensure that those CICS regions are deprovisioned successfully.

You can use z/OS PT to provision additional managed application system (MAS) regions or Web User Interface (WUI) regions into an existing CICSplex. This CICSplex might be a CICSplex SM address space (CMAS) with a CICSplex that was provisioned earlier, or an existing CICSplex that was not provisioned. When you are ready to deprovision the MAS or WUI regions, use the following guidelines to ensure that those CICS regions are deprovisioned successfully.

- Before you attempt to deprovision the region, ensure that the CICS system definition name is not specified as a SCOPE in a Business Application Services (BAS) definition in the CICSplex. If you attempt to deprovision a region that is

associated with BAS definitions, deprovisioning cannot complete because the CICS system definition (CSYSDEF) cannot be removed while it is in use.

- Ensure that the CMAS that the MAS or WUI region is connected to is active so that the CICS system definition (CSYSDEF) for the region can be removed during deprovisioning.
- Before you deprovision a CMAS that manages a CICSplex that MAS and WUI regions were added to, deprovision the MAS and WUI regions first. This enables the CICS system definition (CSYSDEF) for each MAS and WUI region to be removed during deprovisioning of those regions.

Note: If the CMAS cannot be active when a MAS or WUI region is deprovisioned (for example, it is already deprovisioned), you can amend the z/OSMF workflow that runs the deprovisioning process. You amend the workflow to skip the step that attempts to remove the CSYSDEF for the region. Follow steps 1 - 4 in Chapter 18, “Troubleshooting z/OS Provisioning Toolkit,” on page 129 to locate the failing step in the workflow. Select the check box for the failing step and click **Actions > Skip**. Clear the check box for the failing step, select the next Ready step, click **Actions > Perform**, then click **OK**. The deprovisioning process can now continue.

Configuration properties for CICS images

You can configure a number of properties to customize the way that CICS is provisioned through z/OS PT. Some properties are set when you configure a template. Other properties are used to configure an image. Properties for a template are specified in a `cics.properties` ISO8859-1 file, ready to load into z/OSMF. Properties for an image are specified in a `zosptfile`, ready for use by the z/OS PT commands that build, run, and manage provisioned systems.

Type of CICS region

Property	Description
DFH_CICS_TYPE	<p>To set your CICS region type, use one of the following available options:</p> <ul style="list-style-type: none"> • CMAS - A single CICSplex SM address space (CMAS) is provisioned, and a CICSplex is defined to the data repository of the CMAS. You must set extra properties for this region type, as described in “CICSplex SM properties” on page 87. • MAS - A managed application system (MAS) region is managed by CICSplex SM and participates in a CICSplex. You must set extra properties for this region type, as described in “CICSplex SM properties” on page 87. • SMSS - A stand-alone CICS region (SMSS) that can be managed through CMCI. Although the region has CICSplex SM libraries, it does not require a CMAS or participate in a CICSplex. • Unmanaged - A stand-alone CICS region that cannot be managed through CMCI. • WUI - A Web User Interface (WUI) region must be connected to a CMAS region of the same version. You must set extra properties for this region type, as described in “CICSplex SM properties” on page 87.

High-level qualifiers for your installation

Property	Description	Where set
DFH_CICS_HLQ	High-level qualifier (HLQ) of the CICS installation location. DFH_CICS_HLQ is used to define the STEPLIB and DFHRPL concatenations for provisioned CICS regions. If the CICS region needs to access modules in the MVS link pack area (LPA), specify the SIT parameter LPA=YES in the compiled SIT named by the DFH_REGION_SIT property, or in the DFH_REGION_SITPARMS property itself.	Template
DFH_CPSM_HLQ	High-level qualifier (HLQ) of the CICSplex SM installation location.	Template
DFH_LE_HLQ	High-level qualifier (HLQ) of Language Environment®.	Template

User IDs for the configuration

For more information about the steps for which each user ID is used, see “Security requirements for provisioning CICS with z/OS Provisioning Toolkit” on page 72.

Property	Description	Where set
DFH_ADMIN_TSO	TSO user ID that is used for steps that require a greater level of authority; for example, creating data sets.	Template
DFH_ADMIN_CONSOLE	<p>User ID that is used to issue MVS console commands. Commands are issued by using REXX scripts or z/OSMF z/OS console services, depending on the CONSOLE_COMMAND_TYPE property. See “Workflow configuration variables” on page 89.</p> <p>The authorizations that the DFH_ADMIN_CONSOLE user ID requires depend on whether REXX scripts or z/OSMF z/OS console services are used. See “Security requirements for provisioning CICS with z/OS Provisioning Toolkit” on page 72 and z/OS console services in the z/OSMF Programming Guide.</p> <p>When z/OS console services are used, an MVS console is created and used to submit console commands. The MVS console name is the first six characters of the DFH_ADMIN_CONSOLE user ID, followed by the letters CN. Ensure that system messages are routed to the console so that the start and stop of the CICS region can be detected.</p> <p>When the z/OS console services are used, ensure that messages DFHSI1517, EYUNL0099I, EYUXL0010I, DFHKE1799, and IXG661I are not suppressed by the Message Processing Facility (MPF). These messages are used to detect when different types of CICS regions start and stop, and when a CICS region's log streams are deleted successfully.</p> <p>If multiple z/OSMF servers are installed on different LPARs in a sysplex, the generated MVS console name (the first six characters of the DFH_ADMIN_CONSOLE user ID) must be unique on each LPAR.</p>	Template
DFH_ADMIN_ZFS	User ID that has zFS authority to mount and unmount zFS file systems.	Template

Property	Description	Where set
DFH_ADMIN_SECURITY	User ID that has authorization to make dynamic security profiles changes in the Security Manager. By default, no dynamic security updates are made, because the REXX scripts that are run exit immediately with return code zero. This default behavior allows for predefined security profiles to be used. The REXX scripts can be modified to issue commands to dynamically create, update, and delete the security profiles that are needed for the provisioned CICS region. For more information, see “Modifying the CICS workflows to support dynamic security” on page 73.	Template
DFH_APPROVER_TSO	Approval user ID for TSO.	Template
DFH_APPROVER_CONSOLE	Approval user ID for the DFH_ADMIN_CONSOLE user ID to be used.	Template
DFH_APPROVER_SECURITY	Approval user ID for dynamic security configuration.	Template
DFH_APPROVER_ZFS	Approval user ID for zFS.	Template

CICS region configuration

Property	Description	Where set
DFH_REGION_HLQ	High-level qualifier (HLQ) for all CICS region data sets.	Template
DFH_STC_ID	The user ID with which CICS starts. CICS runs as a started task and the job name matches the APPLID of the CICS region. A SAF security profile must exist that ensures the started task runs under the DFH_STC_ID user ID. This profile can be either predefined, or created dynamically as part of the provisioning process. For more information, see “Modifying the CICS workflows to support dynamic security” on page 73. The provisioned CICS region creates log streams under the DFH_STC_ID user ID.	Template
DFH_REGION_LOGSTREAM	Indicates how the log streams that the provisioned CICS region requires are created. Valid options are: <ul style="list-style-type: none"> • DYNAMIC: The log streams that the CICS region requires are created during the provisioning process, and deleted during the deprovisioning process. DYNAMIC is the default value. • EXISTING: The log streams (or the appropriate log stream models) that the CICS region requires must already exist. If JOURNALMODEL resources are required, they must be defined already on a shared system definition data set (CSD), or they can be added to a newly created CSD by using the DFH_REGION_CSD_INPUT property. • DUMMY: The CICS region does not use any log streams, and the CICS region must be started by using an initial start. 	Template

Property	Description	Where set
DFH_REGION_LOGSTREAM_HLQ	<p>A high-level qualifier (up to 8 characters) for the CICS log stream names. This property is optional.</p> <ul style="list-style-type: none"> If you specify a value, the log stream prefix has the following format: <i>DFH_REGION_LOGSTREAM_HLQ.DFH_REGION_APPLID.*</i> If you specify the property, but with no value (for example, <i>DFH_REGION_LOGSTREAM_HLQ=</i>), the log stream prefix has the following format: <i>DFH_REGION_APPLID.*</i> If you do not specify the property, the user ID of the running CICS region, which is specified by the <i>DFH_STC_ID</i> property, is used in the log stream prefix. For example, <i>DFH_STC_ID.DFH_REGION_APPLID.*</i> 	Template
DFH_LOG_HLQ	The high-level qualifier for log stream data set names. This property is optional. If the property is not specified, the high-level qualifier IXGLOGR is used.	Template
DFH_REGION_CICSSVC	The CICS SVC number that is installed on the MVS LPAR.	Template
DFH_REGION_DFLTUSER	User ID used as the default ID for the CICS region.	Template
DFH_REGION_VTAMNODE	Name of the VTAM® node to activate when the CICS region starts up.	Template
DFH_REGION_REGION	JCL region size for the CICS JCL, in the format <i>nnnnM</i> (for example, 400M).	Template
DFH_REGION_MEMLIMIT	JCL MEMLIMIT size for the CICS JCL, in the format <i>nnG</i> (for example, 16G).	Template
DFH_CICS_LICENSE_DATASET	Location of the CICS license data set.	Template

z/OS configuration options

Property	Description	Where set
DFH_ZOS_PROCLIB	<p>Names the PROCLIB data set into which the workflow can create a procedure for the CICS region. The CICS region runs as a started task. If you specify a PROCLIB that is a PDSE, it enables concurrent addition of new procedures to the PROCLIB. If a PDS is used and multiple concurrent updates to the PROCLIB are attempted, z/OS abends them with a 213-30 abend code, to avoid data corruption.</p> <p>When a procedure is created in the PROCLIB, its name will match the APPLID of the CICS region that is provisioned. Therefore, use a suitable naming convention for the CICS region APPLIDs to ensure that the procedure names are unique and do not clash with any existing procedure names. If the procedure name is not unique, another started task with the same name might be stopped during deprovisioning of the CICS region. Ensure that the named PROCLIB data set is included in the PROCLIB concatenation on your system so that it can be found when the command to start the CICS region is issued.</p>	Template

Property	Description	Where set
DFH_ZOS_STCJOBS	Names the data set into which the workflow can create the source JCL that defines the job level characteristics of the started task for the provisioned CICS region. Used only when DFH_STC_JOB_CARD is also specified. If you specify a data set of type PDSE, it enables concurrent addition of new STCJOBS to the data set. If a data set of type PDS is used and multiple concurrent updates to the data set are attempted, z/OS produces a 213-30 abend code, to avoid data corruption.	Template
DFH_STC_JOB_CARD	Specifies the job card to use to define the job level characteristics of the started task for the provisioned CICS region. When specified, DFH_ZOS_STCJOBS must also be used to indicate the data set for STCJOBS.	Template
DFH_ZOS_VSAM_UNIT	UNIT value that is used in creation of data sets, for example: SYSDA, 3390.	Template
DFH_ZOS_VSAM_VOLUME	VOLUME value that is used in the creation of PDS and VSAM data sets, for example, SYSDA, trace, or dump. To use SMS (Storage Management Subsystem) support, specify SMS. In this situation, the VOLUME parameter is omitted from the PDS or VSAM creation.	Template

zFS configuration options

Property	Description	Where set
DFH_ZFS_DATACLASS	DATACLASS (a list of data set allocation attributes and their values) defined on z/OS that is used to create the zFS data sets. For more information about setup details, see “Configuration prerequisites for using z/OS Provisioning Toolkit with CICS” on page 57. For non-SMS-managed systems, either remove the DFH_ZFS_DATACLASS property, or do not give the property a value. Specify a value for DFH_ZOS_VSAM_VOLUME instead.	Template
DFH_ZFS_GROUP	Specifies the name of a SAF group, which is granted read, write, and execute authority to the CICS region zFS directory and contents.	Template
DFH_ZFS_MOUNTPOINT	zFS directory that is used to mount the zFS directories for a CICS region. This property is optional. If omitted, no zFS directories are created for the provisioned CICS region.	Template
DFH_CICS_USSHOME	Location of the CICS USSHOME installation.	Template

Property	Description	Where set
DFH_CICS_USSCONFIG	<p>Specifies the directory in which z/OS UNIX configuration files for the CICS region are stored. When DFH_ZFS_MOUNTPOINT is specified, the contents of this directory are copied into the provisioned CICS region's zFS directory and the USSCONFIG SIT parameter for each provisioned CICS region is set.</p> <p>If the region will include any TCP/IP ports that are secured by using TLS, an SSL cipher suite specification file called <code>strongciphers.xml</code> must exist in the zFS subdirectory <code><DFH_CICS_USSCONFIG>/security/ciphers</code>. Sample cipher suite files are provided in the <code><DFH_CICS_USSHOME>/security/ciphers</code> directory. For more information about cipher suite files, see <i>Cipher suites and cipher suite specification files in CICS TS product documentation</i>.</p> <p>This property is optional. If omitted when DFH_ZFS_MOUNTPOINT is specified, a default USSCONFIG directory is created inside the provisioned CICS region's zFS directory. If omitted when DFH_ZFS_MOUNTPOINT is not specified, the USSCONFIG SIT parameter is not set.</p>	Template
DFH_ZCEE_INSTALL_DIR	The runtime directory in the installation directory of z/OS Connect EE; for example, <code>/usr/lpp/IBM/zosconnect/v3r0/runtime</code> . This property is relevant only for provisioning images for CICS and z/OS Connect EE.	Template

JVM configuration options

Property	Description	Where set
DFH_JAVA_HOME	Location of the Java installation	Template
DFH_REGION_JVMSEVER	<p>The type of JVM server o provision. The following values are allowed:</p> <ul style="list-style-type: none"> Liberty - provisions a Liberty JVM server named DFHWLP. OSGi - provisions an OSGi JVM server named DFHOSGI. Axis2 - provisions an Axis2 JVM server named DFHJVMAX. <i>None</i> - does not provision a JVM server. 	Template or image. If set on the image, this value overrides the value set in the template by the <code>cics.properties</code> file.
DFH_JVM_DEBUG	<p>Dynamically allocate a port for the remote debugging of a JVM server. If DFH_REGION_JVMSEVER is set to Liberty, OSGi, or Axis2, the JVM server is configured to use the allocated port. Valid values are:</p> <ul style="list-style-type: none"> YES NO <p>If not set, no JVM debug port is allocated.</p>	Template
DFH_ZOSCONNECT_USERID	The default user ID for z/OS Connect EE.	Template or image. If set on the image, this value overrides the value set in the template by the <code>cics.properties</code> file.

Node.js configuration options

Support for Node.js requires CICS Transaction Server for z/OS V5.5 or later.

Property	Description	Where set
DFH_NODE_HOME	<p>Location of the Node.js installation directory on zFS.</p> <p>When this property is set, a <code>nodejsprofiles/general.profile</code> file is created in the directory specified by the <code>DFH_CICS_USSCONFIG</code> property for use by the Node.js application. The <code>general.profile</code> file contains entries for the <code>WORK_DIR</code> and <code>NODE_HOME</code> parameters. See Node.js profile and command line options in CICS TS product documentation.</p>	Template

Networking variables

Property	Description	Where set
DFH_JVM_DEBUG	<p>Dynamically allocate a port for the remote debugging of a JVM server. If <code>DFH_REGION_JVMSEVER</code> is set to <code>Liberty</code>, <code>OSGi</code>, or <code>Axis2</code>, the JVM server is configured to use the allocated port. Valid values are:</p> <ul style="list-style-type: none">• YES• NO <p>If not set, no JVM debug port is allocated.</p>	Template
DFH_JVM_DEBUG_USAGETYPE	<p>Optional property to set the usage type for the port that a Java debugger can be connected to, so that the port is dynamically allocated from the port allocation range that this usage type identifies.</p> <p>A valid usage type name is 1-32 characters and the following valid characters: A-Z a-z 0-9</p> <p>Typically, a network administrator uses z/OSMF to configure a port allocation range and specify the usage type for that range.</p> <p>This property is used if <code>DFH_JVM_DEBUG</code> is set to YES.</p>	Template
DFH_REGION_CMCIPOUT_USAGETYPE	<p>Optional property to set the usage type for a CMCI port (used for connection to CICS Explorer) so that the port is dynamically allocated from the port allocation range that this usage type identifies.</p> <p>A valid usage type name is 1-32 characters and the following valid characters: A-Z a-z 0-9</p> <p>Typically, a network administrator uses z/OSMF to configure a port allocation range and specify the usage type for that range.</p>	Template
DFH_REGION_HTTP	<p>Dynamically allocate an HTTP port during provision. Valid values are:</p> <ul style="list-style-type: none">• YES• NO <p>If not set, no HTTP port is allocated.</p>	Template

Property	Description	Where set
DFH_REGION_HTTP_USAGETYPE	<p>Optional property to set the usage type for an HTTP port so that the port is dynamically allocated from the port allocation range that this usage type identifies.</p> <p>A valid usage type name is 1-32 characters and the following valid characters: A-Z a-z 0-9</p> <p>Typically, a network administrator uses z/OSMF to configure a port allocation range and specify the usage type for that range.</p> <p>This property is used if DFH_REGION_HTTP is set to YES.</p>	Template
DFH_REGION_HTTPS	<p>Dynamically allocate an HTTPS port during provision. Valid values are:</p> <ul style="list-style-type: none"> • YES • NO <p>If not set, no HTTPS port is allocated.</p>	Template
DFH_REGION_HTTPS_USAGETYPE	<p>Optional property to set the usage type for an HTTPS port so that the port is dynamically allocated from the port allocation range that this usage type identifies.</p> <p>A valid usage type name is 1-32 characters and the following valid characters: A-Z a-z 0-9</p> <p>Typically, a network administrator uses z/OSMF to configure a port allocation range and specify the usage type for that range.</p> <p>This property is used if DFH_REGION_HTTPS is set to YES.</p>	Template
DFH_REGION_IPIC	<p>Dynamically allocate an IPIC port during provision. Valid values are:</p> <ul style="list-style-type: none"> • YES <p>A TCPIPService is configured for the CICS region to listen for inbound IPIC requests. The default autoinstall program for IPCONN resources (DFHISAIP) is enabled so that outbound IPCONN connections can be automatically created when an inbound request is received.</p> <p>If the property DFH_REGION_SEC=NO is specified, the IPIC port is unsecured. If the property DFH_REGION_SEC=YES is specified, the IPIC port is secured by using SSL with client authentication. The default autoinstall program DFHISAIP creates IPCONN resources that specify SSL(NO) as one of their parameters, meaning that outbound IPIC connections are not secured by SSL.</p> • NO <p>No IPIC port is allocated.</p> <p>If not set, no IPIC port is allocated.</p>	Template

Property	Description	Where set
DFH_REGION_IPIC_USAGETYPE	<p>Optional property to set the usage type for an IPIC port so that the port is dynamically allocated from the port allocation range that this usage type identifies.</p> <p>A valid usage type name is 1-32 characters and the following valid characters: A-Z a-z 0-9</p> <p>Typically, a network administrator uses z/OSMF to configure a port allocation range and specify the usage type for that range.</p> <p>This property is used if DFH_REGION_IPIC is set to YES.</p>	Template
DFH_REGION_TCPIP_USAGETYPE	<p>Optional property to set the usage type for the CICSplex SM WUI port so that the port is dynamically allocated from the port allocation range that this usage type identifies.</p> <p>A valid usage type name is 1-32 characters and the following valid characters: A-Z a-z 0-9</p> <p>Typically, a network administrator uses z/OSMF to configure a port allocation range and specify the usage type for that range.</p>	Template

Security options

Property	Description	Where set
DFH_REGION_KEYRING	<p>Name of a key ring. Used for SSL or TLS connections to CICS. You must specify this property only when:</p> <ul style="list-style-type: none"> DFH_REGION_SEC is set to YES, and DFH_CICS_TYPE is set to SMSS, meaning that a secure CMCI port is created. DFH_REGION_JVMSEVER is set to None, OSGI or Axis2, and DFH_REGION_HTTPS is set to YES, meaning that an HTTPS port is created. <p>If no value is specified and the property is needed, the APPLID of the CICS region is used. The key ring can be dynamically created as part of the provisioning process. For more information, see “Modifying the CICS workflows to support dynamic security” on page 73.</p>	Template
DFH_REGION_SEC	<p>Specify whether security is enabled for the CICS region.</p> <p>Note: If the value is specified as NO, all other security options are ignored.</p>	Template
DFH_REGION_SECPREFX	<p>Specify whether to use SECPREFX. If no value is specified, the APPLID of the CICS region is used. Valid values are:</p> <ul style="list-style-type: none"> NO YES <i>prefix</i> 	Template
DFH_REGION_XAPPC	<p>Specify whether Resource Access Control Facility (RACF) session security can be used with APPC connections. Valid values are:</p> <ul style="list-style-type: none"> NO YES 	Template

Property	Description	Where set
DFH_REGION_XCMD	Specify whether CICS command security is enabled. Valid values are: <ul style="list-style-type: none"> • NO • YES • <i>classname</i> 	Template
DFH_REGION_XDB2	Specify whether DB2ENTRY security checking is enabled. Valid values are: <ul style="list-style-type: none"> • NO • <i>classname</i> 	Template
DFH_REGION_XDCT	Specify whether transient data queue security is enabled. Valid values are: <ul style="list-style-type: none"> • NO • YES • <i>classname</i> 	Template
DFH_REGION_XFCT	Specify whether file security is enabled. Valid values are: <ul style="list-style-type: none"> • NO • YES • <i>classname</i> 	Template
DFH_REGION_XHFS	Specify whether zFS file security is enabled. Valid values are: <ul style="list-style-type: none"> • NO • YES • <i>classname</i> 	Template
DFH_REGION_XJCT	Specify whether journal security is enabled. Valid values are: <ul style="list-style-type: none"> • NO • YES • <i>classname</i> 	Template
DFH_REGION_XPCT	Specify whether started transaction security is enabled. Valid values are: <ul style="list-style-type: none"> • NO • YES • <i>classname</i> 	Template
DFH_REGION_XPPT	Specify whether program security is enabled. Valid values are: <ul style="list-style-type: none"> • NO • YES • <i>classname</i> 	Template
DFH_REGION_XPSB	Specify whether program specification block security is enabled. Valid values are: <ul style="list-style-type: none"> • NO • YES • <i>classname</i> 	Template
DFH_REGION_XPTKT	Specify whether PassTicket security is enabled. XPTKT is a new security option that requires a CICS APAR to be applied. Valid values are: <ul style="list-style-type: none"> • NO • YES 	Template

Property	Description	Where set
DFH_REGION_XRES	Specify whether CICS resource security is enabled. Valid values are: <ul style="list-style-type: none"> • NO • YES • <i>classname</i> 	Template
DFH_REGION_XTRAN	Specify whether transaction attach security is enabled. Valid values are: <ul style="list-style-type: none"> • NO • YES • <i>classname</i> 	Template
DFH_REGION_XTST	Specify whether temporary storage queue security is enabled. Valid values are: <ul style="list-style-type: none"> • NO • YES • <i>classname</i> 	Template
DFH_REGION_XUSER	Specify whether surrogate user checking is enabled. Valid values are: <ul style="list-style-type: none"> • NO • YES 	Template

Registration properties

For a list of properties to register a CICS workflow, see Table 19 on page 119.

Optional CICS properties

The following optional properties are used to insert SIT parameters and data sets into the provisioned region.

Property	Description	Where set
DFH_REGION_STEPLIB	Comma-separated list of variables to append to the standard CICS and Language Environment STEPLIB data sets.	Template
DFH_REGION_STEPLIB_TOP	Comma-separated list of variables to add before the standard CICS and Language Environment STEPLIB data sets.	Template
DFH_REGION_RPL	Comma-separated list of variables to append to the standard CICS and Language Environment DFHRPL data sets.	Template
DFH_REGION_RPL_TOP	Comma-separated list of variables to add before the standard CICS and Language Environment DFHRPL data sets.	Template

Property	Description	Where set
DFH_REGION_CSD_TYPE	<p>Specify whether a new CSD needs to be created for each CICS region, or whether an existing CSD can be shared. For more information about sharing a CSD, see “Shared CSD guidance” on page 91. The default is to create a new CSD, which is then deleted when the CICS region is deprovisioned.</p> <p>The provisioning process adds region-specific resources to the CSD. During deprovision those resources get removed from the CSD if it is shared. When a CSD is shared, the CSD must be accessed in a way that allows the successful provision and deprovision of other CICS regions.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • NEW • SHAREDREADONLY <p>Provision CICS regions that access a non record-level sharing (RLS) CSD in read-only mode. Provision and deprovision fail if region-specific resources cannot be added or removed because the CSD is in use.</p> <ul style="list-style-type: none"> • SHAREDRLS <p>Provision CICS regions that need to access the CSD in update mode.</p> <p>CSD-specific CICS system initialization parameters are automatically set based on the value of this property and cannot be overridden. For more information about CICS system initialization parameters, see “Specifying more CICS system initialization parameters” on page 89.</p>	Template
DFH_REGION_CSD_NAME	Specify the name of an existing CSD to be shared by the provisioned CICS regions. This option applies only when DFH_REGION_CSD_TYPE is set to SHAREDREADONLY or SHAREDRLS.	Template
DFH_REGION_CSD_INPUT	Comma-separated list of variables that allow data sets to be inserted into the CSD upgrade job, for example: DFH_REGION_CSD_INPUT=CICPROV.CSD(EQACSD13). This option applies only when DFH_REGION_CSD_TYPE is set to NEW.	Template
DFH_REGION_SIT	The two-character name of the compiled SIT.	Template
DFH_REGION_SITPARMS	<p>A provisioned CICS region is automatically configured with a set of system initialization (SIT) parameters. DFH_REGION_SITPARMS enables more SIT parameters to be specified for further customization of the region. The parameters must be comma-separated, for example: DFH_REGION_SITPARMS=DEBUGTOOL=YES,AUXTR=ON</p> <p>For more information about which SIT parameters are configured automatically and how, see “Specifying more CICS system initialization parameters” on page 89.</p>	Template

Property	Description	Where set
DFH_REGION_DD_LIST	<p>Comma-separated list of DD cards to add to the CICS region JCL. Commas can be included in the DD card by prefixing them with two backslashes \\. Do not include // at the beginning of the line.</p> <p>For example: DFH_REGION_DD_LIST=EXMPCAT DD DSN=TEST.EXMPLAPP.EXMPCAT\\,DISP=SHR,EXMPCONF DD DSN=TEST.EXMPLAPP.EXMPCONF\\,DISP=SHR</p> <p>For example in an image: ENV {DFH_REGION_DD_LIST=EXMPCAT DD DSN=TEST.EXMPLAPP.EXMPCAT\\,DISP=SHR,EXMPCONF DD DSN=TEST.EXMPLAPP.EXMPCONF\\,DISP=SHR}</p>	Template
DFH_CICS_LLQ	<p>CICS installation data set low-level qualifier, for example CICSLLQ is returned:</p> <p>CTS550.CICS720.SDFHAUTH.CICSLLQ CTS550.CICS720.SDFHLOAD.CICSLLQ CTS550.CICS720.SDFJAUTH.CICSLLQ</p>	Template
DFH_CPSM_LLQ	<p>CICSplex SM installation data set low-level qualifier, for example CPSMLLQ is returned:</p> <p>CTS550.CPSM550.SEYUAUTH.CPSMLLQ CTS550.CPSM550.SEYULOAD.CPSMLLQ</p>	Template

CICSplex SM properties

The following optional properties are used to configure CICS regions that participate in a CICSplex.

Property	Description	DFH_CICS_TYPE	Where set
DFH_CICSPLEX_DEF	Use this property to set the name of a CICSplex to be created during provision of a CMAS. If no value is specified, the CICSplex name is generated dynamically as part of the provisioning process.	CMAS	Template or image
DFH_CICSPLEX	The name of the CICSplex that the MAS or WUI is associated with.	MAS or WUI	Template
DFH_CMAS_APPLID	The VTAM APPLID of the CMAS that the MAS or WUI is attached to. This property is optional.	MAS or WUI	Template
CMAS_REGISTRY_NAME	This property can be set when you provision a WUI or a MAS region. The provisioning workflow looks for the CMAS in the z/OSMF registry to get the information needed for the workflow to connect to the CMAS. This property is optional and can be specified as an alternative to DFH_CICSPLEX and optional properties DFH_CMAS_APPLID and DFH_CMAS_SYSID.	MAS or WUI	Template
DFH_CMAS_SYSID	The SYSID of the CMAS that the MAS or WUI is attached to. This property is optional.	MAS or WUI	Template

Property	Description	DFH_CICS_TYPE	Where set
DFH_CICSGRP	The CICS system group that the MAS or WUI is attached to. The CICS system group must already be defined in the CICSplex. This property is optional.	MAS or WUI	Template
DFH_REGION_EYUPARMS	Define EYUPARM settings for this region in a comma-separated list. This property is optional.	CMAS or MAS or WUI	Template
DFH_REGION_WUIPARMS	Define WUI server initialization parameters in a comma-separated list. This property is optional.	WUI	Template

Provisioning process customizations

Property	Description	Where set
DFH_START_TIMEOUT	The time to wait for the CICS region to start before it reports an error. The valid range is 30 - 600 seconds. The default is 30 seconds.	Template
DFH_STOP_TIMEOUT	The amount of time to wait for the CICS region to stop before it attempts a more forceful stop. The stop order is CEMT P SHUT, CEMT P SHUT IMMEDIATE, CANCEL. The valid range is 10 - 600 seconds. The default is 30 seconds.	Template
VALIDATE_PARAMETERS	During provisioning of a CICS region, several steps validate the properties of the configured template. When the template is configured correctly, you can omit these validation steps so that provisioning each CICS region takes less time. Valid values are as follows: <ul style="list-style-type: none"> • NO. Do not run validation steps. • YES. Run validation steps. This value is the default. 	Template
DFH_DELETE_LOGSTREAM_TIMEOUT	The amount of time the workflow waits to detect successful deletion of the CICS region's log streams during deprovision. The valid range is 10 - 600 seconds. The default is 30 seconds.	Template

Workflow configuration variables

Property	Description	Where set
CONSOLE_COMMAND_TYPE	<p>Sets how the provisioning process issues MVS commands and retrieves command responses for tasks such as starting and stopping a CICS region, checking the status of a CICS region, and deleting log streams. Valid values are as follows:</p> <ul style="list-style-type: none">• SCRIPT. Use REXX scripts to issue console commands. A console is opened for each provisioned CICS region, using the APPLID of the CICS region as the console name. <p>For templates for a <code>cics_getting_started</code> workflow, SCRIPT is the only valid value.</p> <p>If you use this value with templates that provision a CICS region by using IBM Cloud Provisioning and Management for z/OS, it reduces the configuration required.</p> <ul style="list-style-type: none">• APICALL. Use z/OSMF z/OS console services (a REST API) to issue console commands and detect unsolicited messages. A single console manages all provisioned CICS regions. To use this value, you must configure z/OS console services and the relevant additional authorization for the userid that the <code>DFH_ADMIN_CONSOLE</code> property specifies. See “User IDs for the configuration” on page 76 and Chapter 9, “Preparing to use z/OS Provisioning Toolkit with IBM Cloud Provisioning and Management for z/OS,” on page 41. <p>This value is not valid for templates for a <code>cics_getting_started</code> workflow.</p> <p>If this value is not specified, templates for a <code>cics_getting_started</code> workflow use REXX scripts, and templates for workflows that use IBM Cloud Provisioning and Management for z/OS use z/OS console services.</p>	Template

Specifying more CICS system initialization parameters

CICS regions are provisioned with system initialization (SIT) parameters that are configured automatically to support the properties that are defined in the template. Use the `DFH_REGION_SITPARMS` property for further customization of SIT parameters for CICS regions. Any SIT parameter can be specified by using `DFH_REGION_SITPARMS`, but any conflicting SIT parameters are overridden by the automatically configured SIT parameters (conflicts are commented out in the generated started task procedure for the CICS region). The following table lists SIT parameters that are automatically configured and whether they can be overridden. SIT parameters not in the table can all be specified on the `DFH_REGION_SITPARMS` property.

SIT parameter	How it is configured	Can be overridden
APPLID	Set to the dynamically obtained APPLID for the CICS region.	No
CICSSVC	Set to the value of the <code>DFH_REGION_CICSSVC</code> property.	No

SIT parameter	How it is configured	Can be overridden
CPSMCONN	Set to LMAS if the value of DFH_CICS_TYPE is set to MAS.	No
CSDACC	Set to READONLY or READWRITE, depending on the DFH_REGION_CSD_TYPE property.	No
CSDRLS	Set to YES if DFH_REGION_CSD_TYPE is set to SHAREDRLS, otherwise set to NO.	No
DFLTUSER	Set to the value of the DFH_REGION_DFLTUSER property.	No
GMTEXT	Set to a string that includes the dynamic APPLID of the provisioned region and the type of region that is being provisioned.	Yes
GMTRAN	Set to CESN if DFH_REGION_SEC is set to YES.	Yes
GRPLIST	Set to a list that contains the specific resource definitions for the provisioned CICS region.	Yes, but the provided GRPLIST must specify a maximum of three entries. These are concatenated onto a first entry that is generated during provisioning to add specific resource definitions for the provisioned CICS region. This generated list also includes the CICS groups typically in DFHLIST, so DFHLIST is not required as a separate entry in your GRPLIST.
IRCSTRT	Set to YES.	Yes
ISC	Set to YES.	Yes
JVMPROFILEDIR	Set to a dynamically provisioned zFS directory for the CICS region.	No
KEYRING	Set to the value of the DFH_REGION_KEYRING property, or the dynamically provisioned APPLID of the CICS region, if DFH_REGION_KEYRING is not specified. Set when DFH_REGION_SEC is set to YES.	No
RLS	Set to YES only if DFH_REGION_CSD_TYPE is set to SHAREDRLS.	Only when DFH_REGION_CSD_TYPE is not SHAREDRLS.
SEC	Set to the value of the DFH_REGION_SEC property.	No
SECPRFX	Set to the value of the DFH_REGION_SECPRFX property, or the dynamically provisioned APPLID of the CICS region if the DFH_REGION_SECPRFX property is not set.	No
SIT	Set to the value of the DFH_REGION_SIT property.	No
START	Set to AUTO.	Yes
SYSIDNT	Set to a dynamic system ID based on the dynamic APPLID for the CICS region.	No
TCPIP	Always set to YES to enable the CICS region to open TCPIP SERVICES for web or IPIC traffic.	No
TRANISO	Set to YES.	Yes

SIT parameter	How it is configured	Can be overridden
USSHOME	Set to the value of the DFH_CICS_USSHOME property.	No
USSCONFIG	Set to the value of the DFH_CICS_USSCONFIG property.	No
XAPPC	Set to the value of the DFH_REGION_XAPPC property.	No
XCMD	Set to the value of the DFH_REGION_XCMD property.	No
XDB2	Set to the value of the DFH_REGION_XDB2 property.	No
XDCT	Set to the value of the DFH_REGION_XDCT property.	No
XFCT	Set to the value of the DFH_REGION_XFCT property.	No
XHFS	Set to the value of the DFH_REGION_XHFS property.	No
XJCT	Set to the value of the DFH_REGION_XJCT property.	No
XPCT	Set to the value of the DFH_REGION_XPCT property.	No
XPPT	Set to the value of the DFH_REGION_XPPT property.	No
XPSB	Set to the value of the DFH_REGION_XPSB property.	No
XPTKT	Set to the value of the DFH_REGION_XPTKT property.	No
XRES	Set to the value of the DFH_REGION_XRES property.	No
XTRAN	Set to the value of the DFH_REGION_XTRAN property.	No
XTST	Set to the value of the DFH_REGION_XTST property.	No
XUSER	Set to the value of the DFH_REGION_XUSER property.	No

Shared CSD guidance

By default, when a CICS region is provisioned, a new CSD is defined for that CICS region. When the CICS region is deprovisioned, the CSD is deleted. This ensures that each CICS region has a unique CSD, and that any changes that are made to that CSD do not affect other CICS regions.

You can change the value of the DFH_REGION_CSD_TYPE parameter to provision CICS regions that share an existing CSD. Using a shared CSD ensures that updates to the CSD are available to all CICS regions that share the CSD. If a CSD is shared, it must not be held open by a CICS region. A CSD that remains open can cause provision and deprovision failures.

The shared CSD can be used for regions that are provisioned and deprovisioned by using z/OSMF to avoid situations where a CSD held open in a CICS region can cause provisioning and deprovisioning failures. One approach is to clone an existing CSD, then use that clone as the shared CSD for provisioning.

Accessing the CSD by using record-level sharing (RLS) provides the highest level of sharing. If the CICS regions you provision need to update the CSD, specify DFH_REGION_CSD_TYPE=SHAREDRLS. For more information about considerations for creating a CSD, see Setting up shared data sets, CSD and SYSIN in CICS TS product documentation.

If the CICS regions you provision do not need to update the CSD, a non-RLS CSD can be shared. The CICS regions open the CSD in read-only mode, which avoids problems with the CSD being locked when DFHCSDUP jobs are run during the provisioning and deprovisioning workflows. This option is specified by the property DFH_REGION_CSD_TYPE=SHAREDREADONLY.

If you need to update a non-RLS shared CSD, ensure that the updates do not cause provisioning and deprovisioning failures. DFHCSDUP jobs that the provisioning workflows submit include an extra DD statement that opens a data set exclusively. This data set is used as a serialization mechanism to ensure that the DFHCSDUP jobs do not cause each other to fail because the CSD is in use when multiple provisioning or deprovisioning processes are running concurrently. You can specify the same DD card in your own DFHCSDUP jobs to serialize your updates with updates that are made during the provisioning and deprovisioning processes. The DD card is as follows, where the values of DFH_REGION_HLQ and DFH_ZOS_VSAM_UNIT are properties that are described in tables “CICS region configuration ” on page 77 and “z/OS configuration options” on page 78.

```
//CSDLOCK DD DSN=<DFH_REGION_HLQ>.CSDLOCK,
// DISP=(MOD,CATLG),
// UNIT=<DFH_ZOS_VSAM_UNIT>,SPACE=(1,1)
```

Output properties from provisioning CICS

The properties that can be output when you provision a CICS region are listed. These properties can be output during provisioning when you provision a CICS region by using z/OS PT and IBM Cloud Provisioning and Management for z/OS.

Table 12. Output properties from provisioning a CICS region

Property	Description
DB2_REGISTRY_NAME	The z/OSMF Service Component Registry external name for the IBM Db2 system that the CICS region is connected to.
DB2_REGISTRY_ALIAS	A user-specified alias for the Db2 system that the CICS region is connected to. This alias is used in the description of the DB2CONN resource that is installed in the CICS region.
DFH_CICSGRP	The name of the CICS system group that the CICS region is added to.
DFH_CICSplex	The name of the CICSplex that the CICS region is connected to.
DFH_CICSplex_DEF	When a CMAS is provisioned, the name of the CICSplex definition (CplexDEF) that is created.
DFH_CICS_TYPE	The CICS type: <ul style="list-style-type: none"> • CMAS. CICSplex SM address space. • MAS. Managed application system. • SMSS. Stand-alone CICS region. • Unmanaged. Unmanaged CICS region. • WUI. Web User Interface region. For more information about CICS region types, see “Overview of supported CICS types” on page 58.
DFH_CMAS_APPLID	The APPLID of the CMAS that manages the CICS region.
DFH_CMAS_SYSID	The SYSID of the CMAS that manages the CICS region.
DFH_JVM_DEBUG	The port number that a Java debugger can be connected to.
DFH_REGION_APPLID	The APPLID for the CICS region.
DFH_REGION_CMCIPT	The port number that CICS Explorer can be connected to.
DFH_REGION_HOSTNAME	Host name that the CICS region runs on.
DFH_REGION_HTTP	The port number on which the CICS region listens for HTTP requests. If the CICS region is running a Liberty JVM server, Liberty manages the port. Otherwise, the CICS region itself has the port open.

Table 12. Output properties from provisioning a CICS region (continued)

Property	Description
DFH_REGION_HTTPS	The port number on which the CICS region listens for HTTPS requests. If the CICS region is running a Liberty JVM server, Liberty manages the port. Otherwise, the CICS region itself has the port open.
DFH_REGION_IPIC	The port number on which the CICS region listens for IPIC connections.
DFH_REGION_TCPIPSPORT	The CICSplex SM WUI port number.
DFH_REGION_ZFS_DIRECTORY	The zFS directory for the CICS region. Use this directory to store any files that should be deleted when the CICS region is deprovisioned.
MQ_REGISTRY_NAME	The z/OSMF Service Component Registry external name for the IBM MQ queue manager this CICS region is connected to.
MQ_REGISTRY_ALIAS	A user-specified alias for the IBM MQ queue manager that this CICS region is connected to. The alias is used in the description of the MQCONN resource that is installed in the CICS region.

Modifying the set of CICS configuration properties that can be overridden

The CICS template provides workflows to provision and deprovision a CICS region. A set of properties determine the configuration of the provisioned CICS region. You set these properties before you add the template to z/OSMF. To change them in the template, you must refresh the template in z/OSMF, or create a new version of the template. You can enable different configurations of provisioned CICS regions that use the same template in z/OSMF, by overriding the values of a subset of the properties that are set in the template. To do this, you specify these properties in a z/OS PT image. The subset of properties that can be overridden in an image can be customized for your environment.

Before you begin

Consider how much flexibility you would like in customizing the CICS regions that are provisioned from each z/OSMF template. Templates added into z/OSMF have an approval process, where users across the organization can approve the automation steps that occur during the provisioning process. The values set for the template properties affect these automation steps (for example, you can specify the high-level qualifier for data sets that are created during the provisioning of each CICS region).

Properties that affect the provisioning process, for example, data set high-level qualifiers, would need to be reviewed as part of the template approval process, so these properties cannot be overridden in a z/OS PT image.

Properties that change the configuration of the CICS region, for example, its SIT parameters, and DFHRPL concatenation, might be appropriate to be overridden in a z/OS PT image so that the same template can be used to provision CICS regions with different configurations. Other examples of properties that can be overridden in the z/OS PT images include properties that allow provisioned MAS or WUI regions to connect into different CICSplexes or CICS groups, and properties to set port allocation ranges for specific port types.

Overriding properties in this way can minimize the number of templates that need to be maintained and approved in z/OSMF. To ensure that only users with the

correct permissions can build images, you can restrict the permissions of the images directory in z/OS PT.

Procedure

1. Locate the `image_properties.xml` file: `zospt/workflows/cics/extensions/image_properties.xml`. This file describes which template properties can be overridden in a z/OS PT image.
2. To determine whether the set of template properties that can be overridden is appropriate, review the descriptions of the properties in the `image_properties.xml` file.
3. To add more properties that can be overridden, uncomment or add extra `atCreate` XML elements to the file. To limit the set of properties that can be overridden, comment out or delete the appropriate `atCreate` XML elements from the file. For example, to override the `CMAS_REGISTRY_NAME`, uncomment, or add, the following line to `image_properties.xml`:

```
<atCreate name="CMAS_REGISTRY_NAME" required="false" prompt="false"/>
```

The full set of CICS properties is described in “Configuration properties for CICS images” on page 75, but check `image_properties.xml` for the properties that can be overridden.

Job entry subsystem (JES) job names

The names of the JES jobs that are used for steps in z/OS PT processes for provisioning and managing a CICS region are listed. You might need to review these job names and ensure that they meet the security requirements for your system.

Provisioning

Table 13 shows the steps and the name of the associated JES job when you use z/OS PT to provision a CICS region.

Table 13. Provisioning steps and names of JES jobs

Step	Description	Name of JES job in z/OS PT 1.1.5
validateCICSVariables	Validate access to CICS data sets	Z\${instance-DFH_REGION_SYSID}CDS
validateZFSVariables	Validate access to z/OS File System (zFS) directories	Z\${instance-DFH_REGION_SYSID}VZV
validateSharedCSD	Validate the shared CICS system definition (CSD)	Z\${instance-DFH_REGION_SYSID}VDS
validateConsoleRouting	Validate the extended multiple console support (MCS) console configuration	
validateConsoleRoutingRexx	Validate that console commands can be issued	Z\${instance-DFH_REGION_SYSID}VCA
obtainCICSVersion	Obtain the CICS version and hostname for the logical partition (LPAR)	Z\${instance-DFH_REGION_SYSID}OBV
CreateCICSDatasets	Create CICS region data sets	Z\${instance-DFH_REGION_SYSID}DS
DefineZFS	Define the zFS file system	Z\${instance-DFH_REGION_SYSID}DZF
MountAndSetupZFS	Mount the zFS file system and create required directories	Z\${instance-DFH_REGION_SYSID}ZFS

Table 13. Provisioning steps and names of JES jobs (continued)

Step	Description	Name of JES job in z/OS PT 1.1.5
FormatCICSDatasets	Format CICS region data sets	Z\${instance-DFH_REGION_SYSID}FMC
CreateCICSLogstream	Create the CICS log stream model	Z\${instance-DFH_REGION_SYSID}CLS
CreateDataRepository	Create standard CICSplex SM address space (CMAS) data sets	Z\${instance-DFH_REGION_SYSID}DSC
CreateWUIDatasets	Create Web User Interface (WUI) region data sets	Z\${instance-DFH_REGION_SYSID}DSW
setupCICSsecurity	Create the CICS security configuration	
DefineMAStoCICSplex	Create the CICS system definition	Z\${instance-DFH_REGION_SYSID}DMS
setDirectoryPermissions	Set directory permissions	Z\${instance-DFH_REGION_SYSID}SPM
CreateNewCSD	Creating CICS CSD	Z\${instance-DFH_REGION_SYSID}NCS
CreateCSDDefinitions	Create CICS CSD definitions	Z\${instance-DFH_REGION_SYSID}CSD
CreateCICSJCL	Create the CICS region job control language (JCL)	Z\${instance-DFH_REGION_SYSID}CJ
CreateCMASJcl	Create the CMAS CICS region	Z\${instance-DFH_REGION_SYSID}CCJ
CreateWUIJCL	Create the WUI region	Z\${instance-DFH_REGION_SYSID}CWJ
ProcessStartCICSRexx	Start the CICS region	Z\${instance-DFH_REGION_SYSID}STE
CheckStartup	Check the REST status from the start command	
removeTempWUIDef	Remove the temporary region definition from the CICSplex	Z\${instance-DFH_REGION_SYSID}DWD

Deprovisioning

Table 14 shows the steps and the name of the associated JES job when you use z/OS PT to deprovision a CICS region.

Table 14. Deprovisioning steps and names of JES jobs

Step	Description	Name of JES job in z/OS PT 1.1.5
ValidateVariables	Check that variables are valid for deprovisioning. If they are not, manual correction is required.	
CheckCICSStoppedRexx	Stop the CICS region	Z\${instance-DFH_REGION_SYSID}STP
ShutdownFailed	Check the REST status from the cancel command	Z\${instance-DFH_REGION_SYSID}RTN
deleteStandardDatasets	Delete standard CICS data sets	Z\${instance-DFH_REGION_SYSID}DDS
deleteCICSLogstreamRexx	Delete the CICS log stream model	Z\${instance-DFH_REGION_SYSID}DLG
checkDeleteLogstreamResponse	Check the delete log stream response	Z\${instance-DFH_REGION_SYSID}CKR
DeleteZFS	Delete the zFS directory for the CICS region	Z\${instance-DFH_REGION_SYSID}DLZ

Table 14. Deprovisioning steps and names of JES jobs (continued)

Step	Description	Name of JES job in z/OS PT 1.1.5
deleteCICSsecurity	Delete the CICS security configuration	
deleteCMASDatasets	Delete CMAS data sets	Z\${instance-DFH_REGION_SYSID}DMD
deleteWUIDatasets	Delete WUI region data sets	Z\${instance-DFH_REGION_SYSID}DW
DeleteMASDefine	Delete the CICS system definition	Z\${instance-DFH_REGION_SYSID}DMS
deleteCSDDefinitions	Delete definitions from the shared CSD	Z\${instance-DFH_REGION_SYSID}DCS

Starting a container

Table 15 shows the steps and the name of the associated JES job when you use z/OS PT to start a CICS region.

Table 15. Steps to start a CICS region and names of JES jobs

Step	Description	Name of JES job in z/OS PT 1.1.5
RemountIfRequired	Check that zFS is mounted	Z\${instance-DFH_REGION_SYSID}RMT
ProcessStartCICSrexx	Start the CICS region	Z\${instance-DFH_REGION_SYSID}STE
CheckStartup	Check the REST status from the start command	Z\${instance-DFH_REGION_SYSID}RTE
removeTempWUIDef	Remove the temporary region definition from the CICSplex	Z\${instance-DFH_REGION_SYSID}DWD

Stopping a container

Table 16 shows the steps and the name of the associated JES job when you use z/OS PT to stop a CICS region.

Table 16. Steps to stop a CICS region and names of JES jobs

Step	Description	Name of JES job in z/OS PT 1.1.5
CheckCICSStoppedRexx	Stop the CICS region	Z\${instance-DFH_REGION_SYSID}STP
ShutdownFailed	Check the REST status from the cancel command	Z\${instance-DFH_REGION_SYSID}RTN

Checking the status of a provisioned container

Table 17 shows the step and the name of the associated JES job when you use z/OS PT to check the status of a CICS region. Examples of when this step is invoked include the **zospt inspect**, **start**, or **stop** commands.

Table 17. Step to check the status of a CICS region and the name of the JES job

Step	Description	Name of JES job in z/OS PT 1.1.5
checkStatusRexx	Check whether the CICS region is active	Z\${instance-DFH_REGION_SYSID}CKC

Troubleshooting CICS provisioning with z/OS Provisioning Toolkit

Use this information to help diagnose common problems that are found when CICS regions are provisioned by z/OS PT.

Checking the rest status from the start command step fails in the `zospt run` command output

When you use the **zospt run** command to provision a CICS container from an image, the workflow step Checking the rest status from the start command might fail with a return code of 8. This failure is because the previous step failed to detect a console message to indicate that CICS completed startup processing successfully. The message to detect depends on the type of CICS region that is being provisioned:

- When DFH_CICS_TYPE is Unmanaged or SMSS, the message number is DFHSI1517.
- When DFH_CICS_TYPE is MAS or WUI, the message number is EYUNL0099I.
- When DFH_CICS_TYPE is CMAS, the message number is EYUXL0010I.

Check the CICS job log to find out whether the appropriate console message was issued. If the console message was not issued, check the job log further for the reason. If the message was issued but not detected, check the DFH_ADMIN_CONSOLE property. Ensure that the user ID being used for console commands is configured correctly to be able to find unsolicited messages from an extended MCS console. Ensure that the console messages for detection are not suppressed by the Message Processing Facility (MPF). It is also possible to adjust how long the workflow waits to detect the console message after it issues the command to start the CICS region. Check the DFH_START_TIMEOUT property to understand the default wait time. If the console message was not issued in that time period, consider increasing the value of the property. For more information about the DFH_ADMIN_CONSOLE and DFH_START_TIMEOUT properties, see “Configuration properties for CICS images” on page 75.

Troubleshooting problems when parallel processing is enabled

When parallel processing is enabled for a workflow, if a failure occurs, z/OS PT returns the failed step, and might return successful or running steps. To identify and diagnose the failed step, use the Workflow Steps table in the z/OSMF web interface, as described in Chapter 18, “Troubleshooting z/OS Provisioning Toolkit,” on page 129.

Delete the CICS system definition step fails in the `zospt rm` command for a MAS or WUI region in a CICSplex

When you use the **zospt rm** command to deprovision a MAS or WUI region in a CICSplex, the workflow step Delete the CICS system definition might fail with a return code of 8. This failure can occur in the following situations:

- The CMAS that manages the CICSplex is not active. For example, the CMAS was deprovisioned before the MAS or WUI region.
- The CICS system definition (CSYSDEF) for the MAS or WUI region is still associated with one or more Business Application Services (BAS) resources that are defined in the CICSplex.

To resolve this problem, you can amend the z/OSMF workflow that runs the deprovisioning process. You amend the workflow to skip the step that attempts to remove the CSYSDEF for the region.

1. Follow steps 1 - 4 in Chapter 18, “Troubleshooting z/OS Provisioning Toolkit,” on page 129 to locate the failed step in the workflow.
2. Select the check box for the failed step and click **Actions > Skip**.
3. Clear the check box for the failed step, select the next Ready step, click **Actions > Perform**, then click **OK**.

The deprovisioning process can now continue.

See also “Deprovisioning guidelines for regions in a CICSplex” on page 74.

Unable to find free port message occurs

When you provision a region from a template that uses dynamic port allocation, this message can occur if all the ports in the valid port ranges are in use. This situation does not indicate any error.

However, if you provision a region from a template that uses dynamic port allocation and multiple port ranges, this message can occur if you specify a usage type configuration property in the `cics.properties` file, but no corresponding usage type value is set for the port allocation range in the Usage Type field in z/OSMF. To resolve this problem:

- Ensure that you used the correct name for the usage type in the `cics.properties` file. Typically, your network administrator supplies this name.
- Ensure that the Usage Type field for the port allocation range is set with the same name in z/OSMF. Typically, your network administrator sets this.

Diagnosing other z/OS PT problems

For more information about troubleshooting z/OS PT, see Chapter 18, “Troubleshooting z/OS Provisioning Toolkit,” on page 129.

Chapter 13. Provisioning z/OS Connect Enterprise Edition with z/OS Provisioning Toolkit

You can use z/OS PT to provision a stand-alone instance of an IBM z/OS Connect Enterprise Edition server, or an instance of z/OS Connect EE embedded in CICS. You can provision IBM z/OS Connect Enterprise Edition V2.0 and V3.0. Find out more about how to provision a stand-alone instance of a z/OS Connect EE server.

Configuration prerequisites for using z/OS Provisioning Toolkit with z/OS Connect Enterprise Edition

In addition to the requirements for z/OS PT, some system configuration is needed to enable z/OS PT to provision instances of z/OS Connect EE.

System requirements

For any specific system requirements to be able to provision instances of z/OS Connect EE by using z/OS PT, see the z/OS Provisioning Toolkit Software Compatibility Report.

Configuration requirements

Ensure that a data class that is compatible with creating zFS directories exists. This table shows an example of a valid data class configuration:

Override Space	No
Volume Count	15
Data set Name Type	EXTENDED
If Extended	PREFERRED
Extended Addressability	Yes
Record Access Bias	User
Space Constraint Relief	No
Reuse	No
Initial Load	Recovery
RLS CF Cache Value	All
RLS Above the 2-GB Bar	No
Extent Constraint Removal	No
CA Reclaim	Yes
Log Replicate	No
System Determined Blocksize	No

Message suppression

Ensure that message suppression is switched off for the following message:

- CWWKF0011I: z/OS Connect EE is started.

This message is used by the z/OS Connect EE workflow and needs to be available to be sent to an extended MCS console that the workflow uses to determine whether z/OS Connect EE is started. For more information about suppressing messages, see Suppressing messages in z/OS MVS Planning: Operations.

User ID authorization

Ensure that the user IDs that run the various steps of the provisioning process have the appropriate security authorizations. For more information, see “Security requirements for provisioning instances of z/OS Connect EE with z/OS Provisioning Toolkit” on page 103.

Adding a z/OS Connect Enterprise Edition template by using IBM Cloud Provisioning and Management for z/OS

This scenario walks through the steps to add a z/OS Connect EE Cloud Provisioning template to z/OSMF and test it.

This scenario sets up the template to use the z/OSMF ability to run workflows under a set of authorized provisioning user IDs. The template also uses the z/OSMF ability to allocate ports dynamically from a pool of resources, and add and remove security profiles dynamically to set up security for provisioned environments such as z/OS Connect EE.

Before you begin

See the requirements in Chapter 10, “Adding an IBM Cloud Provisioning and Management for z/OS template for use with z/OS Provisioning Toolkit,” on page 45. You also need the following resources:

- A range of ports in a resource pool. These ports can then be allocated dynamically when you run z/OS PT.
- A set of user IDs (agreed with your security administrator) that has sufficient authority to provision an instance of z/OS Connect EE. See “Security requirements for provisioning instances of z/OS Connect EE with z/OS Provisioning Toolkit” on page 103 for details.

About this task

This scenario provisions an instance of z/OS Connect Enterprise Edition V2.0 from a template called `zosconnect_v3r0`. The following steps summarize the procedure:

- Optionally, copy the template (if you want more than one variant of the template).
- Configure the properties that z/OSMF uses to process the template.
- Add the Cloud Provisioning template to z/OSMF.
- Approve the template for use.
- Associate the template with a domain and tenant.
- Configure the network resource pool.
- Set up the environment for z/OS PT.
- Use z/OS PT to provision an instance of z/OS Connect EE.
- Resolve any provisioning errors.
- Publish the template.

Procedure

1. Optional: If you want more than one variant of the template, copy the directory for the z/OS Connect EE. For example, copy the /zosconnect_v3r0 directory in the /templates directory. Give the new directory a different name but keep it in the /templates directory.
2. Configure the zosconnect.properties file that z/OSMF uses to process the template:
 - z/OS PT includes sample zosconnect.properties files for provisioning instances of z/OS Connect EE (zospt/templates/zosconnect_vnrn/zosconnect.properties). The files are in code page ISO-8859-1.
 - Add any additional or alternative properties. See “Configuration properties for z/OS Connect Enterprise Edition images” on page 106.
 - Make sure that you include properties for the set of user IDs that has sufficient authority to provision a z/OS Connect EE instance.
3. Add the template to z/OSMF through the z/OSMF web interface.
 - a. From a web browser, connect to the z/OSMF web interface.
 - b. Click **Cloud Provisioning > Software Services > Templates**. Click **Add Template > Standard**.
 - c. Enter the full path of the manifest file for the template in **Template Source file**, for example, zospt/templates/zosconnect_v3r0/zosconnect.mf. z/OSMF must be able to read from this directory tree.
 - d. Enter the template name in **Template Name**, for example, zosconnect_v3r0.

Note: The template name is important. It must match the name that is used in an image that is built with **zospt**.

After you click **OK**, the template is loaded into z/OSMF.

4. Optional: If a “Pending Security Update” message is displayed, click **Actions > Refresh Template**.
5. Ensure that approvals are completed by all users that are named as approvers for the template, or for steps in the template. For example, you might require approvals from the following users:
 - A security administrator, if provisioning dynamically creates SAF resources.
 - A z/OS systems programmer, to allow issuing of console commands to start and stop the server.
 - A capacity planner.
6. Associate the template with a tenant in the domain in the z/OSMF web interface.
 - a. Select your template, then click **Actions > Associate Tenant**.
 - b. Select the tenant that your z/OSMF administrator specifies.
 - c. Click **OK**.
 - d. In the Add Template and Resource Pool for Tenant view, select or specify the following properties and values:

Property	Setting
Software services instance name prefix	Choose a prefix of up to 5 characters; this prefix is used as the first 5 characters of the provisioned instance's job name. Note: To use this template with z/OS PT, do not select the Create workload management pool check box.

Property	Setting
Maximum number of software services instances.	Specify the maximum number of software services instances that can be allocated.
Maximum number of software services instances for a user	Specify the maximum number of software services instances that can be allocated by an individual user.
System selection for provisioning	Choose the system in the sysplex into which to provision the z/OS Connect EE instance. The default is to provision into the same system on which z/OSMF is running. For more information about configuring z/OSMF for provisioning across different systems, see z/OSMF product documentation. Note: To use this template with z/OS PT, do not select the Prompt user for system option.

- e. Click **OK**.
7. Configure the network resource pool by using z/OSMF.
 - a. In the Configuration Assistant, select **Manage z/OS Cloud Configuration** then click **Proceed**.
 - b. Select **Cloud Domain** and click **Proceed**.
 - c. Select the pool for your template from the **Network Resource Pools** table (the pool is named *domainname.tenantname.templatename*).
 - d. Click **Actions > Modify**.
 - e. On the **Attributes** tab, select **Is complete**.
 - f. On the **Port Allocation** tab, select the pool of ports that was agreed with your network administrator for use in this template.
 - g. Click **Save**. Your network resource pool is configured. For more information about configuring the network resource pool, see z/OSMF product documentation.
8. Ensure that you have access to the pool of ports. The template is now ready for testing with the **zospt** utility.
9. Set up the environment for z/OS PT. Set the following environment variables:
 - export zospt_domain=*domain_name*
 - export zospt_tenant=*tenant_name*
10. Use z/OS PT and the **zospt build** command to build an image that references the template. See “Building an image with z/OS Provisioning Toolkit” on page 49.
You might set *imageName* to be the same as *templateName*, for example *zosconnect_v3r0*.
11. Use z/OS PT and the **zosptrun** command to run the image to provision an instance of z/OS Connect EE that uses your template. Issue the following command, using the **--draft** option to run the template in *Draft approved* state, and where *imageName* is the name of the image you built:

```
zospt run imageName --draft
```


If the properties are all correct, output from the steps of the **zospt run** command is displayed, ending with success message, as shown in the following example:

Created container ZOSCONNECT_ZCON01 with id 031fa6c4-91c4-4035-989e-c898a5cc8049

12. If an error is reported, see Chapter 18, “Troubleshooting z/OS Provisioning Toolkit,” on page 129. If you need to change the properties for the workflow to resolve an error, adjust the properties, refresh the template and retry:

- a. Update the properties file. For example, update `zospt/templates/zosconnect_v3r0/zosconnect.properties`.
 - b. Refresh the template from the z/OSMF web interface.
 - c. Reapprove the template for use.
 - d. Issue the following command to remove the container that failed to provision correctly.

```
zospt rm --force containerName
```
 - e. Retry the **zospt run** command, as described in step 11 on page 102.
13. Optional: Publish the template for every user in the tenant to use.
 After the template is published, you use the **zospt run** command without the **--draft** option.
- If you have created and tested a template and it is ready to publish, you can also enable faster provisioning by turning off the validation. See the `VALIDATE_PARAMETERS` property in “Provisioning process customizations” on page 110.

What to do next

Use z/OS PT to build images from the template that you loaded. See “Building an image with z/OS Provisioning Toolkit” on page 49 for details.

Security requirements for provisioning instances of z/OS Connect EE with z/OS Provisioning Toolkit

The user IDs that are used by z/OSMF to provision instances of z/OS Connect EE with z/OS PT require certain permissions to run steps of the provisioning process.

User ID	Permissions	Use
ZCON_ADMIN_CONSOLE	<ul style="list-style-type: none"> Authority to open an MVS console that is named XXXXXCN, where XXXXX is the first six characters of the ZCON_ADMIN_CONSOLE user ID. Authority to issue a console START command. Authority to issue a console STOP command. Ability to issue a console command to display whether a job is active. Authority to issue a console CANCEL command. 	<ul style="list-style-type: none"> Start the z/OS Connect EE instance started task. Stop the z/OS Connect EE instance started task. Check whether the z/OS Connect EE instance is running.

User ID	Permissions	Use
ZCON_ADMIN_SECURITY	<ul style="list-style-type: none"> Permission to run the REXX scripts for dynamic security and the commands that are issued by these REXX scripts. By default, the scripts exit immediately with return code zero and do not modify any security profiles. This default behavior allows for predefined security profiles to be used. The scripts can be modified to issue commands to dynamically create, update, and delete the security profiles that are needed for the provisioned instance of z/OS Connect EE. For more information, see “Modifying the z/OS Connect Enterprise Edition workflows to support dynamic security” on page 105. 	<ul style="list-style-type: none"> Dynamically create and delete a SAF certificate and key ring. Dynamically create and delete SAF security profiles that are required by the provisioned instance.
ZCON_ADMIN_SERVER	<ul style="list-style-type: none"> Authority to run the zosconnect create command to create z/OS Connect EE instances. Authority to issue standard UNIX Systems Services commands, for example, chmod, mkdir, iconv, cp, ctag, mv, echo, rmdir, and pax. Authority for the started task to run under this user ID. Permission to run the REXX scripts for dynamic creation and deletion of a SAF key ring. By default, the scripts exit immediately with return code zero and do not modify any security profiles. This default behavior allows for predefined security profiles to be used. The scripts can be modified to issue commands to dynamically create, update, and delete the security profiles that are needed for the provisioned instance of z/OS Connect EE. For more information, see “Modifying the z/OS Connect Enterprise Edition workflows to support dynamic security” on page 105. 	<ul style="list-style-type: none"> Create the z/OS Connect EE instance from a script Copy server configuration files. Copy application configuration from an image into the zFS file system. Delete the z/OS Connect EE instance's data directory during deprovision.
ZCON_ADMIN_TSO	<ul style="list-style-type: none"> Authority to copy a default procedure from the z/OS Connect EE installation directory on zFS. Authority to copy the configured procedure from zFS into the PROCLIB named by the ZCON_ZOS_PROCLIB variable. Authority to delete the procedure from the PROCLIB named by the ZCON_ZOS_PROCLIB variable. 	<ul style="list-style-type: none"> Configure the procedure that is used to start the z/OS Connect EE instance and store the configured procedure into the PROCLIB. Delete the procedure from the PROCLIB during deprovision.

User ID	Permissions	Use
ZCON_ADMIN_ZFS	<ul style="list-style-type: none"> ALTER (create/delete/read/write/update) ZCON_FILE_SYSTEM_HLQ.* Authority to run the TSO MOUNT command. Authority to run the /usr/sbin/unmount command. Authority to issue chown commands. 	<ul style="list-style-type: none"> Create and mount zFS data set. Unmount and delete zFS data set. Change the ownership of the zFS directory to the ZCON_ADMIN_SERVER user ID.

Modifying the z/OS Connect Enterprise Edition workflows to support dynamic security

You can use REXX scripts in the workflow to enable the provision and deprovision process to dynamically configure the security profiles that are needed for an instance of z/OS Connect EE.

The z/OS Connect EE template provides workflows to provision and deprovision an instance of z/OS Connect EE. A step in these workflows runs a REXX script, which by default does not make any dynamic security changes. You can modify the REXX scripts so that security profiles can be added, modified, and deleted dynamically. In this way, the REXX scripts enable the provision and deprovision process to dynamically configure the security profiles that are needed for that instance of z/OS Connect EE.

About this task

Four example REXX scripts are provided with the workflows for provisioning and deprovisioning instances of z/OS Connect EE. The scripts show example RACF commands for configuring the necessary security profiles. You can use them as a guide to create scripts for an alternative system authorization facility (SAF) provider.

- `defineSecurity.rexx` contains example Resource Access Control Facility (RACF) definitions to dynamically create and configure the profiles that are needed by a z/OS Connect EE server.
- `deleteSecurity.rexx` contains example RACF definitions to delete the profiles that were dynamically created by the `defineSecurity.rexx` script.
- `defineKeyring.rexx` contains example RACF definitions to dynamically create a key ring for SSL certificates that are needed by a z/OS Connect EE server.
- `deleteKeyring.rexx` contains example RACF definitions to delete the profiles that were dynamically created by the `defineKeyring.rexx` script.

The example REXX scripts exit immediately without issuing any commands. To use the scripts, the required security calls must be made before the exit statement.

Note:

Check the return code from each security command for an appropriate response in the `defineSecurity.rexx` script. If the REXX script exits with a nonzero return code, provisioning of the z/OS Connect EE instance fails.

Edit `deleteSecurity.rexx` to undo any changes that are made by `defineSecurity.rexx`.

Edit deleteKeyring.rexx to undo any changes that are made by defineKeyring.rexx.

A failure in provisioning an instance of z/OS Connect EE might occur before, during, or after any dynamic security configuration process. The deleteSecurity.rexx and deleteKeyring.rexx scripts are always called during deprovisioning, so ensure that these scripts can complete successfully if they attempt to delete security profiles that do not exist. This situation might occur if the provision process failed before a profile was created, or if the deprovision process is retried and an earlier attempt already removed the security profiles.

Procedure

1. Locate the following REXX extensions for dynamic security in the `zospt/workflows/zosconnect/extensions` directory:
 - `defineKeyring.rexx` - run during provision of an instance of z/OS Connect EE.
 - `defineSecurity.rexx` - run during provision of an instance of z/OS Connect EE.
 - `deleteKeyring.rexx` - run during deprovision of an instance of z/OS Connect EE
 - `deleteSecurity.rexx` - run during deprovision of an instance of z/OS Connect EE
2. Locate the line at the start of each REXX script that issues `exit 0`. Make sure that any security calls are made before this command.
3. Use the sample REXX scripts as a guide to customize the scripts for your environment.

Configuration properties for z/OS Connect Enterprise Edition images

You can configure a number of properties to customize the way that instances of z/OS Connect EE are provisioned through z/OS PT. Properties for a template are specified in a `zosconnect.properties` ISO8859-1 file, ready to load into z/OSMF.

All the configuration properties for an instance of z/OS Connect Enterprise Edition are set at the template level.

High-level qualifiers for your installation

Property	Description	Where set
ZCON_INSTALL_DIR	The path to the z/OS Connect EE installation directory on zFS.	Template

User IDs for the configuration

For more information about the steps for which each user ID is used, see “Security requirements for provisioning instances of z/OS Connect EE with z/OS Provisioning Toolkit” on page 103.

Property	Description	Where set
ZCON_ADMIN_CONSOLE	<p>User ID that is used for CONSOLE API commands and needs to have special OPERPARM configuration.</p> <p>When the CONSOLE API is used, an MVS console is created and used to submit console commands. The MVS console name is the first six characters of the ZCON_ADMIN_CONSOLE user ID followed by the letters CN. If multiple z/OSMF servers are installed on different LPARs in a sysplex, the generated MVS console name (the first six characters of the DFH_ADMIN_CONSOLE user ID) must be unique on each LPAR.</p> <p>For more information about authorizations and configuration of the z/OS console services, see z/OS console services in the z/OSMF Programming Guide.</p> <p>Also ensure that message CWWKF0011I is not suppressed by the Message Processing Facility (MPF), because this message is used to detect when z/OS Connect EE starts.</p>	Template
ZCON_ADMIN_SECURITY	<p>User ID under which dynamic SAF security updates are made during the provisioning process.</p> <p>By default, no dynamic security updates are made, because the REXX scripts that are run exit immediately with return code zero. This default behavior allows for predefined security profiles to be used. The REXX scripts can be modified to issue commands to dynamically create, update, and delete the security profiles that are needed for the provisioned instances of z/OS Connect EE. For more information, see “Modifying the z/OS Connect Enterprise Edition workflows to support dynamic security” on page 105.</p>	Template
ZCON_ADMIN_SERVER	<p>User ID under which the provisioned instance of z/OS Connect EE runs. This user ID owns the data directory on zFS of the instance of z/OS Connect EE.</p> <p>z/OS Connect EE runs as a started task and its job name matches the instance name of the provisioned z/OS Connect EE instance in z/OSMF. A SAF security profile must exist that ensures the started task runs under the ZCON_ADMIN_SERVER user ID. This profile can be predefined, or created dynamically as part of the provisioning process. For more information, see “Modifying the z/OS Connect Enterprise Edition workflows to support dynamic security” on page 105.</p>	Template
ZCON_ADMIN_TSO	User ID that can be used in the step to create the procedure in the PROCLIB named by the ZCON_ZOS_PROCLIB variable for the instance of z/OS Connect EE.	Template
ZCON_ADMIN_ZFS	User ID that has zFS authority to mount and unmount zFS file systems that are used by the z/OS Connect EE instance.	Template
ZCON_APPROVER_CONSOLE	Approval user ID of the ZCON_ADMIN_CONSOLE user ID for issuing MVS console commands.	Template
ZCON_APPROVER_SECURITY	Approval user ID of the ZCON_ADMIN_SECURITY user ID for making dynamic SAF security changes.	Template
ZCON_APPROVER_SERVER	Approval user ID of the ZCON_ADMIN_SERVER user ID for running the instance of z/OS Connect EE.	Template

Property	Description	Where set
ZCON_APPROVER_TSO	Approval user ID of the ZCON_ADMIN_TSO user ID for creating the procedure that is used to start the instance of z/OS Connect EE.	Template
ZCON_APPROVER_ZFS	Approval user ID of the ZCON_ADMIN_ZFS user ID for creating and mounting directories in zFS.	Template

z/OS Connect EE instance configuration

Property	Description	Where set
ZCON_FILE_SYSTEM_HLQ	Specifies the high-level qualifier for the zFS data sets that are defined during the provisioning of the z/OS Connect EE instance.	Template

z/OS configuration options

Property	Description	Where set
ZCON_ZOS_PROCLIB	<p>Names a PROCLIB data set into which the workflow can create a procedure for the z/OS Connect EE instance. The z/OS Connect EE instance runs as a started task. If you specify a PROCLIB that is a PDSE, it enables concurrent addition of new procedures to the PROCLIB. If a PDS is used and multiple updates to the PROCLIB are attempted concurrently, z/OS abends them with a 213-30 abend code, to avoid data corruption.</p> <p>When a procedure is created in the PROCLIB, its name will match the instance name of the z/OS Connect EE instance that is provisioned. Therefore, use a suitable naming convention for the instances z/OS Connect EE to ensure that the procedure names are unique and do not clash with any existing procedure names. If the procedure name is not unique, another started task with the same name might be stopped during deprovisioning of the z/OS Connect EE instance. Ensure that the named PROCLIB data set is included in the PROCLIB concatenation on your system so that it can be found when the command to start the z/OS Connect EE instance is issued.</p>	Template
ZCON_ZOS_STCJOBS	Names the data set into which the workflow can create the source JCL that defines the job level characteristics of the started task for the provisioned instance of z/OS Connect EE. Used only when ZCON_STC_JOB_CARD is also specified. If you specify a data set of type PDSE, it enables concurrent addition of new STCJOBS to the data set. If a PDS is used and multiple concurrent updates to the data set are attempted, z/OS produces a 213-30 abend code, to avoid data corruption.	Template
ZCON_STC_JOB_CARD	Specifies the job card to use for the started task of the provisioned instance of z/OS Connect EE. When specified, ZCON_ZOS_STCJOBS must also be used to indicate the data set prefix for STCJOBS.	Template
ZCON_ZOS_VSAM_VOLUME	VOLUME value that is used in the creation of the zFS data sets, for example, SYSDA. If the value is specified as SMS (Storage Management Subsystem), the VOLUME parameter is omitted from the VSAM creation.	Template

zFS configuration options

Property	Description	Where set
ZCON_ZFS_DATACLASS	<p>Specifies a data class that is used to define a zFS data set for the z/OS Connect EE instance. A data set is defined, formatted for zFS, and mounted into a directory created in the ZCON_ZFS_MOUNTPOINT location.</p> <p>DATACLASS (a list of data set allocation attributes and their values) defined on z/OS that is used to create the zFS data sets. For more information about setup details, see “Configuration prerequisites for using z/OS Provisioning Toolkit with z/OS Connect Enterprise Edition” on page 99.</p> <p>For non-SMS-managed systems, either remove the ZCON_ZFS_DATACLASS property, or do not give the property a value. Specify a value for ZCON_ZOS_VSAM_VOLUME instead.</p>	Template
ZCON_ZFS_MOUNTPOINT	Specifies a zFS directory into which a data directory for the provisioned instance of z/OS Connect EE can be defined.	Template

JVM configuration options

Property	Description	Where set
JAVA_HOME	Location of the Java installation	Template

Security options

Property	Description	Where set
ZCON_SAF_AUTHENTICATION	<p>Specify whether to use SAF authentication. Valid values are:</p> <ul style="list-style-type: none">• FALSE. If you specify FALSE, you must manually configure an alternative authentication mechanism.• TRUE	Template

Property	Description	Where set
ZCON_TLS	<p>Specify whether to use transport layer security. Valid values are:</p> <ul style="list-style-type: none"> • JKS. An HTTPS port is needed for the instance of z/OS Connect EE and a Java keystore is used for transport layer security. The Java keystore is created dynamically by z/OS Connect EE during startup. • NO. The instance of z/OS Connect EE accepts requests over an unsecured HTTP port. • SAF. An HTTPS port is needed for the instance of z/OS Connect EE and a SAF key ring is used for transport layer security. The value of the ZCON_SERVER_KEYRING parameter determines whether a predefined key ring is used. <p>If ZCON_SERVER_KEYRING is specified, it is assumed to be the name of a predefined key ring.</p> <p>If ZCON_SERVER_KEYRING is not specified, the key ring is created dynamically by running the defineKeyring.rexx script that is found in the workflows/extensions folder. To enable the key ring to be created dynamically, you must modify defineKeyring.rexx (the default is to exit immediately with return code zero and not to make any dynamic profile updates). The corresponding deleteKeyring.rexx script must also be modified for your environment.</p>	Template
ZCON_SERVER_KEYRING	Names an existing SAF key ring to use when ZCON_TLS=SAF.	Template
ZCON_ZFS_GROUP	Specifies the name of a SAF group, which is granted read, write and run authority to the z/OS Connect EE instances data directory. Users of the group can add APIs and services into the file system of the z/OS Connect EE instance.	Template

Provisioning process customizations

Property	Description	Where set
ZCON_STOP_TIMEOUT	The time to wait for the z/OS Connect EE instance to stop before a CANCEL command is issued. The valid range is 10 - 600 seconds. The default is 30 seconds.	Template
TEMP_DIR	The name of a temporary directory into which the ZCON_ADMIN_ZFS user ID can create, update, and delete files. The default is /tmp. When you provision across systems in a sysplex, this directory must be accessible from the system that is running z/OSMF and the system on which the environment is being provisioned.	Template
VALIDATE_PARAMETERS	<p>During provisioning of an instance of z/OS Connect EE, a step validates some properties of the configured template. When the template is configured correctly, you can omit this validation step so that provisioning each instance of z/OS Connect EE takes less time. Valid values are as follows:</p> <ul style="list-style-type: none"> • NO. Do not run the validation step. • YES. Run the validation step. This value is the default. 	Template

Output properties from provisioning IBM z/OS Connect Enterprise Edition

The properties that can be output when you provision an instance of IBM z/OS Connect Enterprise Edition are listed. These properties can be output during provisioning when you provision an instance of z/OS Connect EE by using z/OS PT and IBM Cloud Provisioning and Management for z/OS.

Table 18. Output properties from provisioning an instance of IBM z/OS Connect Enterprise Edition

Property	Description
CICS_REGISTRY_ALIAS	A user-specified alias for the CICS region that z/OS Connect EE is connected to over IPIC. The alias is used as the id of the zosconnect_cicsIPICConnection configuration element.
CICS_REGISTRY_NAME	The z/OSMF Service Component Registry external name for the CICS region that z/OS Connect EE is connected to.
ZCON_HOSTNAME	Host name that the z/OS Connect EE server is running on.
ZCON_HTTP_PORT	The HTTP port for the z/OS Connect EE server.
ZCON_HTTPS_PORT	The HTTPS port for the z/OS Connect EE server.
ZCON_INSTALL_DIR	The installation directory path for z/OS Connect EE.

Chapter 14. Managing containers created by z/OS Provisioning Toolkit

After you provision containers by using the **zospt run** command, you can manage them by using other z/OS PT commands. You can see which containers are running, start and stop them, and deprovision them when they are no longer needed.

Listing running containers

To list running containers, use the **zospt ps** command:

```
zospt ps
```

This command lists details for any running containers that you are authorized to view, and all running registered containers. You can view running containers that you created, and you might be able to view running containers that were created by other members of the same tenant. If you are a domain administrator, you can view more containers. Any user can view all registered containers.

The command shows the container name, the image name, the owner, the date the container was created, the current state, the template that was used to create the container, the system the container was created on, and the container type of each running container. The container name of each registered container is prefixed with REG.

```
2018-12-12 16:24:35 Connecting to z/OSMF on host system01.hursley.ibm.com port 32000.
NAME          IMAGE      OWNER  CREATED      STATE      TEMPLATE      SYSTEM  CONTAINER TYPE
CICS_CICPD000 cics_55   user01 2018-12-09T10:58:31 provisioned cics_55      MV01    Standard
ZOSCONNECT_ZCON008 testApi   user01 2018-12-10T02:27:34 provisioned zosconnect_v3r0 MV02    Standard
CICS_CICPD000 cics_55_liberty user02 2018-12-11T10:44:31 provisioned cics_55      MV01    Standard
REG-DB2_DB2000 N/A       user02 2018-12-12T15:01:55 provisioned db2_registration MV01    Standard
```

Listing all containers

To list all containers, enter the following command:

```
zospt ps -a
```

This command lists details for any running containers that you are authorized to view, including deprovisioned and deprovisioning-failed containers, and all registered containers. You can view running containers that you created, and you might be able to view running containers that were created by other members of the same tenant. If you are a domain administrator, you can view more containers. Any user can view all registered containers.

The command shows the version of z/OS and z/OSMF on the running LPAR. It then shows the container name, the image name, the owner, the date the container was created, the current state, the template that was used to create the container, the system the container was created on, and the container type of each container. The container name of each registered container is prefixed with REG.

```

2018-12-12 16:34:35 Connecting to z/OSMF on host system01.hursley.ibm.com port 32000.
2018-12-12 16:34:35 z/OS level is V2R3.
2018-12-12 16:34:35 z/OS Management Facility level is V2R3.
2018-12-12 16:34:35 z/OSMF system nicknames are MV01, MV02.
2018-12-12 16:34:35 Getting started workflows will run on the system with nickname MV01.
2018-12-12 16:34:35 zospt installation directory : /u/cicsuser/zospt

```

NAME	IMAGE	OWNER	CREATED	STATE	TEMPLATE	SYSTEM	CONTAINER	TYPE
CICS_CICPD000	cics_55	user01	2018-12-09T10:58:31	provisioned	cics_55	MV01		Standard
ZOSCONNECT_ZCON008	testApi	user01	2018-12-10T02:27:34	provisioned	zosconnect_v3r0	MV02		Standard
CICS_CICPD000	cics_55_liberty	user02	2018-12-11T10:44:31	provisioned	cics_55	MV01		Standard
REG-DB2_DB2000	N/A	user02	2018-12-12T15:01:55	provisioned	db2_registration	MV01		Standard
CICS_CICPD00C	cics_55_app	user02	2018-12-12T15:43:34	being-deprovisioned	cics_55	MV01		Standard
REG-MQ_CICPJ00A	N/A	user03	2018-12-12T16:01:03	deprovisioned	mq_registration	MV01		Standard

Filtering the list of containers

To filter the results of the **zospt ps** or **zospt ps -a** commands by **OWNER**, **STATE**, or **SYSTEM**, you can add **-f** or **--filter** to the command. You can specify the **-f** option multiple times in a single command. For example, to filter the results by **OWNER** and **STATE**, enter the following command:

```
zospt ps -f owner=user02 -f state=provisioned
```

Output from this example command is as follows:

```

2018-12-12 16:44:35 Connecting to z/OSMF on host system01.hursley.ibm.com port 32000.

```

NAME	IMAGE	OWNER	CREATED	STATE	TEMPLATE	SYSTEM	CONTAINER	TYPE
CICS_CICPD000	cics_55_liberty	user02	2018-12-11T10:44:31	provisioned	cics_55	MV01		Standard
REG-DB2_DB2000	N/A	user02	2018-12-12T15:01:55	provisioned	db2_registration	MV01		Standard

For a complete syntax description and any conditions of use, see Chapter 19, “The **zospt** command syntax,” on page 135.

Inspecting containers

To see more details about the container you created, including the status of the container, and to see a list of any linked-to containers, use the **zospt inspect** command:

```
zospt inspect containerName | containerID
```

Example output from the command **zospt inspect CICS_CICPJ00Z** is as follows:

```
{
  "container-name": "CICS_CICPJ00Z",
  "container-id": "5209fa00-fadc-4d1e-aa37-e2c4b7160f30",
  "instance-name": "CICS_CICPJ00Z",
  "template": "cics_55",
  "image_name": "cics_55",
  "created-time": "2018-12-09T12:31:14.425Z",
  "type": "CICS",
  "system": "MV01",
  "sysplex": "PLEX01",
  "owner": "user123",
  "state": "provisioned",
  "tenant-name": "default",
  "domain-name": "default",
  "public_properties": {
    "DFH_CICS_TYPE": "SMSS",
    "DFH_REGION_APPLID": "CICPJ00Z",
    "DFH_REGION_CMCIPORT": "32001",
    "DFH_REGION_HOSTNAME": "HOST",
    "DFH_REGION_HTTP": "28499",
    "DFH_REGION_ZFS_DIRECTORY": "/u/zospt/mount/CICPJ00Z",
    "INSTANCE_STATUS": "active"
  },
  "connected-containers": [{"container-name": "DB2_DB2J001"}]
}
```

You can also use the **zospt inspect** command to see more details about a registered container that is created by any user:

```
{
  "container-name": "REG-DB2_CICPJ000",
  "container-id": "89b5481b-acbb-4a6c-9ffd-a3139e0b504f",
  "instance-name": "REG-DB2_CICPJ000",
  "template": "db2_registration",
  "image_name": "N/A",
  "created-time": "2018-12-09T15:01:55.034Z",
  "type": "REG-DB2",
  "system": "MV01",
  "sysplex": "PLEX01",
  "owner": "user234",
  "state": "provisioned",
  "tenant-name": "default",
  "domain-name": "default",
  "public_properties": {
    "DSN_HLQ": "SYS2.DB2.V11",
    "DSN_SSID": "DJ29",
    "DSN_ZFS_HLQ": "/db2/db2v11"
  },
  "connected-containers": []
}
```

The **INSTANCE_STATUS** property that is returned shows the current status of the container. If the container was stopped in any way, the **INSTANCE_STATUS** shows as inactive.

The **connected-containers** property that is returned shows the container or containers that the inspected container is linked to.

For a complete syntax description and any conditions of use, see Chapter 19, “The **zospt** command syntax,” on page 135.

Starting and stopping containers

Containers are started in the running state when you provision them. You can use the **zospt stop** command to stop one or more running containers and a **zospt**

start command to restart one or more containers. You can use *containerName* or *containerID* to specify a target container, and you can specify multiple container names or IDs, each separated by a space. For example, you might use the following commands to stop and start two containers and use their name and ID interchangeably:

```
zospt stop CICS_CICPP003 05dc7424-3178-4e98-a179-1d7f95b749a2
...
zospt start d5975552-711d-437e-ba86-63d97108523e CICS_CICPD001
```

For a complete syntax description and any conditions of use, see Chapter 19, “The **zospt** command syntax,” on page 135.

Removing and deprovisioning containers

When one or more containers are no longer needed, use the **zospt rm** command to remove them. This process deletes the data sets and zFS directories that are associated with the environment. This example command shows how to remove two containers:

```
zospt rm containerName1 containerName2
```

If one or more of the specified containers are running, the **zospt rm** command fails. In this case, you can either stop any running containers, or use the **--force** option on the command to deprovision the environment. If the container you want to remove is a registered container, you must use the **--force** option.

You can use *containerName* or *containerID* to specify a target container, and you can specify multiple container names or IDs, each separated by a space.

For a complete syntax description and any conditions of use, see Chapter 19, “The **zospt** command syntax,” on page 135.

Exploring further

Now that you are working with z/OS PT, stay in touch through the IBM developerCenters. There, you can find scenarios and examples of z/OS PT in action:

- CICS Developer Center for scenarios that involve CICS.
- z/OS Connect Enterprise Edition Developer Center for scenarios that involve z/OS Connect EE.

Chapter 15. Registering existing subsystems in z/OSMF

Use the provided registration workflows to register existing instances of CICS, IBM Db2, and IBM MQ in the z/OSMF web interface. Registered subsystems are then available to be dynamically linked to during the provision of other containers.

z/OSMF maintains a registry of provisioned and registered subsystems. You can provision a container that automatically links to a registered subsystem by using the **link** option on the **zospt run** command. For more information about how to use the **link** option, see “Using the **--link** option” on page 52.

The following diagram shows how to dynamically connect to a registered subsystem by using the **--link** command.

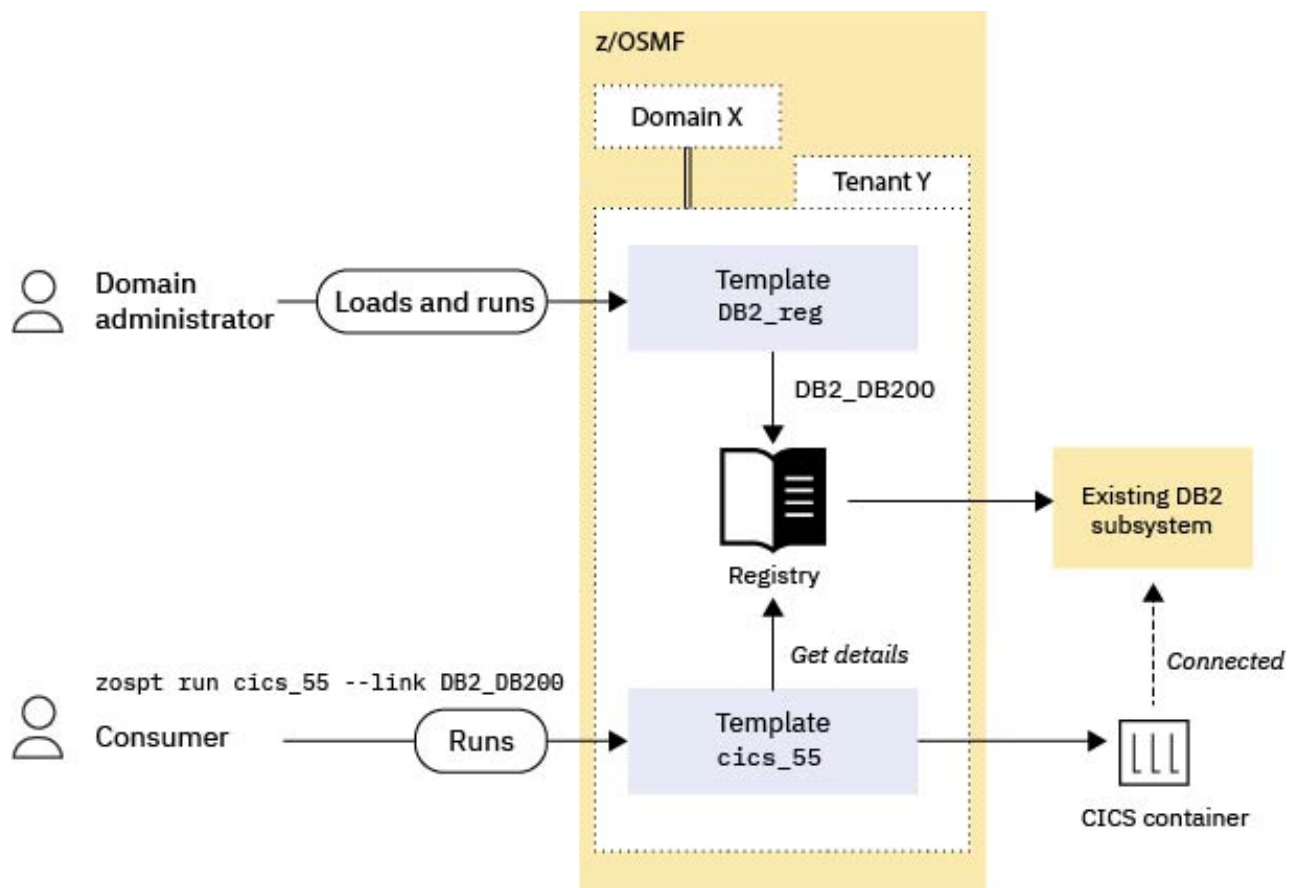


Figure 8. Dynamically connect to a registered subsystem by using the **--link** command

Registering subsystems allows systems that are being provisioned by z/OS PT to connect to existing subsystems:

- With minimal configuration.
- Without the need to install the correct resources in the registered subsystem.

How to register an existing subsystem

A z/OSMF administrator must give you domain administrator authority to enable you to add templates into the domain.

The following diagram shows how to register a subsystem instance in the z/OSMF web interface.

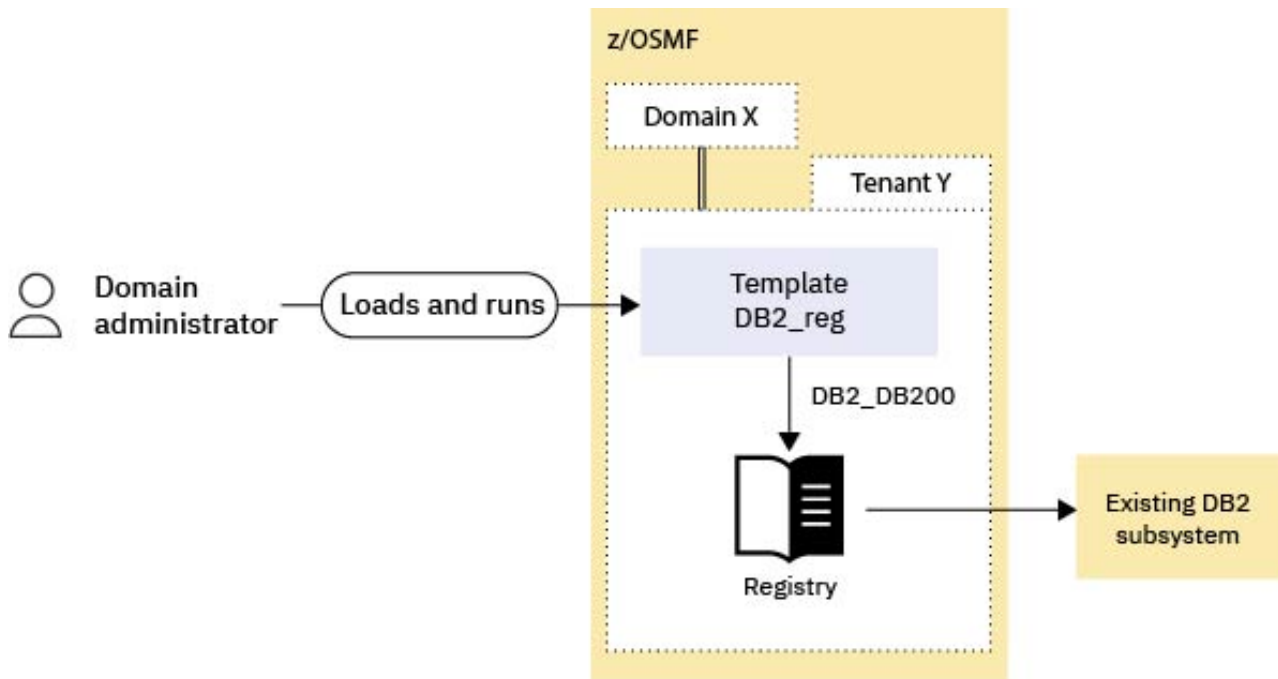


Figure 9. Register a subsystem instance in the z/OSMF web interface

For more information, see Chapter 10, “Adding an IBM Cloud Provisioning and Management for z/OS template for use with z/OS Provisioning Toolkit,” on page 45. Upload and publish the registration workflow by using the z/OSMF web interface:

1. Add the template to z/OSMF by going to **Cloud Provisioning > Software Services** and selecting the **Templates** tab. Click **Add Template** and specify the manifest file to load in the **Template source file** field (this file is the .mf file, found in the `../zospt/workflows/registration/subsystem` folder). Click **Load**. The fields for **Workflow file**, **Actions file**, and **Workflow variables input** are populated automatically. Give your template a name, and click **OK**.
2. Associate the template with a tenant. When you make the association, you can set limits for how many times the container can be provisioned and how many each user is allowed to provision. The association automatically creates a resource pool for the template and the network administrator is notified.

Note: For users of IBM z/OS V2.3, to ensure that a container is visible to other members of the tenant, select the **Allow other members of the tenant to access and run actions for software services instances** check box.

3. If you are ready, you can publish the template by selecting **Software Services** and selecting the **Templates** tab. Select the template that you want to publish, and select **Actions > Publish**. No approvals are needed to complete this step. If you do not publish the template at this stage, the template remains in draft state.

Running the template

You can run the template in either of the following ways:

- Use z/OS PT to build the image (by using the **zospt build** command), then run the image (by using the **zospt run** command). If the template is in draft state, use the **--draft** option on the **zospt run** command. Add the subsystem properties to the `zosptfile`. For more information about building and running images, see “Building an image with z/OS Provisioning Toolkit” on page 49 and “Running a z/OS Provisioning Toolkit image” on page 50.
- Use the z/OSMF web interface to run the template by selecting **Software Services** and selecting the **Templates** tab. Select the template that you want to run, and select **Actions > Run > .** If your template is in draft state, select **Actions > Test Run**. Select the **Associated tenant**, and enter the subsystem properties. Properties that are marked with an asterisk are mandatory.

Supported z/OS PT commands on a registered subsystem

The following **zospt** commands are supported on registered subsystems:

- **zospt run**
- **zospt rm -f(--force)**
- **zospt inspect**
- **zospt ps**

Further information

- For specific registration information for CICS, see “CICS registration information.”
- For specific registration information for Db2, see “IBM Db2 registration information” on page 120.
- For specific registration information for IBM MQ, see “IBM MQ registration information” on page 121.

CICS registration information

Use the provided CICS registration workflows (found in `../zospt/workflows/registration/subsystem`) to register existing CICS regions in z/OSMF.

Supplying CICS properties

When you register an existing CICS system in z/OSMF, you need to supply certain information about the CICS region, for example, its APPLID. If you run the template directly from the z/OSMF Web UI, you are prompted to complete the necessary fields. Alternatively, you can build a z/OS PT image with the information that is specified in the `zosptfile`. The built image can then be run. The following table shows the CICS properties that you need to add to register an existing CICS region by using z/OS PT:

Table 19. CICS properties to supply to the z/OSMF template

Property	Description	Required?
DFH_REGION_APPLID	Specify the APPLID of the CICS region that you want to register.	Yes

Table 19. CICS properties to supply to the z/OSMF template (continued)

Property	Description	Required?
DFH_REGION_HOSTNAME	Specify the host name of the CICS region that you want to provision.	Yes
DFH_REGION_IPIC	Dynamically allocates an IPIC port during provision. For more information about this property, see “Configuration properties for CICS images” on page 75.	Yes

The properties can be put in a zosptfile (see the following example), which is used to build an image:

FROM scratch

```
# Define the z/OSMF template to drive for provisioning
ENV ZOSMF_TEMPLATE=cics_registration_template
ENV DFH_REGION_APPLID=<regionApplid>
ENV DFH_REGION_HOSTNAME=<regionHostname>
ENV DFH_REGION_IPIC=<portNumber>
```

IBM Db2 registration information

Use the provided Db2 registration workflows (found in ../zospt/workflows/registration/subsystem) to register existing Db2 systems in z/OSMF.

Supplying Db2 properties

When you register an existing Db2 system in z/OSMF, you need to supply certain information, for example the Db2 subsystem ID. If you run the template directly from the z/OSMF web interface, you are prompted to complete the necessary fields. Alternatively, you can build a z/OS PT image with the information that is specified in the zosptfile. The built image can then be run. The following table shows the Db2 properties:

Table 20. Db2 properties to supply to the z/OSMF template

Property	Description	Required?
DSN_HLQ	Location of the Db2 libraries in MVS.	Yes
DSN_SSID	The Db2 subsystem ID.	No
DSN_GROUPID	The Db2 group ID.	No
DSN_ZFS_HLQ	Location of Db2 directories in UNIX System Services.	Yes

The properties can be put in a zosptfile (see the following example), which is used to build an image:

FROM scratch

```
# Define the z/OSMF template to drive for provisioning
ENV ZOSMF_TEMPLATE=Db2_registration_template
ENV DSN_HLQ=<DB2_hlq>
ENV DSN_SSID=<DB2_ssid>
ENV DSN_ZFS_HLQ=<zfs_hlq>
```

IBM MQ registration information

Use the provided IBM MQ registration workflows (found in `../zospt/workflows/registration/subsystem`) to register existing IBM MQ queue managers in z/OSMF.

Supplying IBM MQ properties

When you register an existing IBM MQ queue manager in z/OSMF, you need to supply certain information about the IBM MQ queue manager, for example, its subsystem ID. If you run the template directory from the z/OSMF Web UI, you are prompted to complete the necessary fields. Alternatively, you can build a z/OS PT image with the information that is specified in the `zosptfile`. The built image can then be run. The following table shows the IBM MQ properties that you need to add to register an existing IBM MQ queue manager by using z/OS PT:

Table 21. IBM MQ properties to supply to the z/OSMF template

Property	Description	Required?
CSQ_SSID	The IBM MQ subsystem ID. Can be set to the name of a queue-sharing group or the name of a queue manager.	Yes
CSQ_TARG_LIB_HLQ	Location of the IBM MQ libraries.	Yes

The properties can be put in a `zosptfile` (see the following example), which is used to build an image:

```
FROM scratch
```

```
# Define the z/OSMF template to drive for provisioning
ENV ZOSMF_TEMPLATE=mq_registration_template
ENV CSQ_SSID=<mq_ssid>
ENV CSQ_TARG_LIB_HLQ=<mq_hlq>
```

Chapter 16. z/OS Provisioning Toolkit And z/OSMF composite templates

You can use a z/OSMF composite template to provision or deprovision multiple related containers by using a single z/OS PT command. The connection between the containers is defined in z/OSMF when you create the composite template. Containers that need to be provisioned and deprovisioned together can be managed in z/OSMF, instead of running multiple **zospt run** commands and using the **--link** option to connect them together.

Composite templates

Composite templates are available with z/OSMF 2.2 and later. A composite template is constructed by using z/OSMF to define which set of templates make up the composite template, the order in which the templates are provisioned, and how they are connected together.

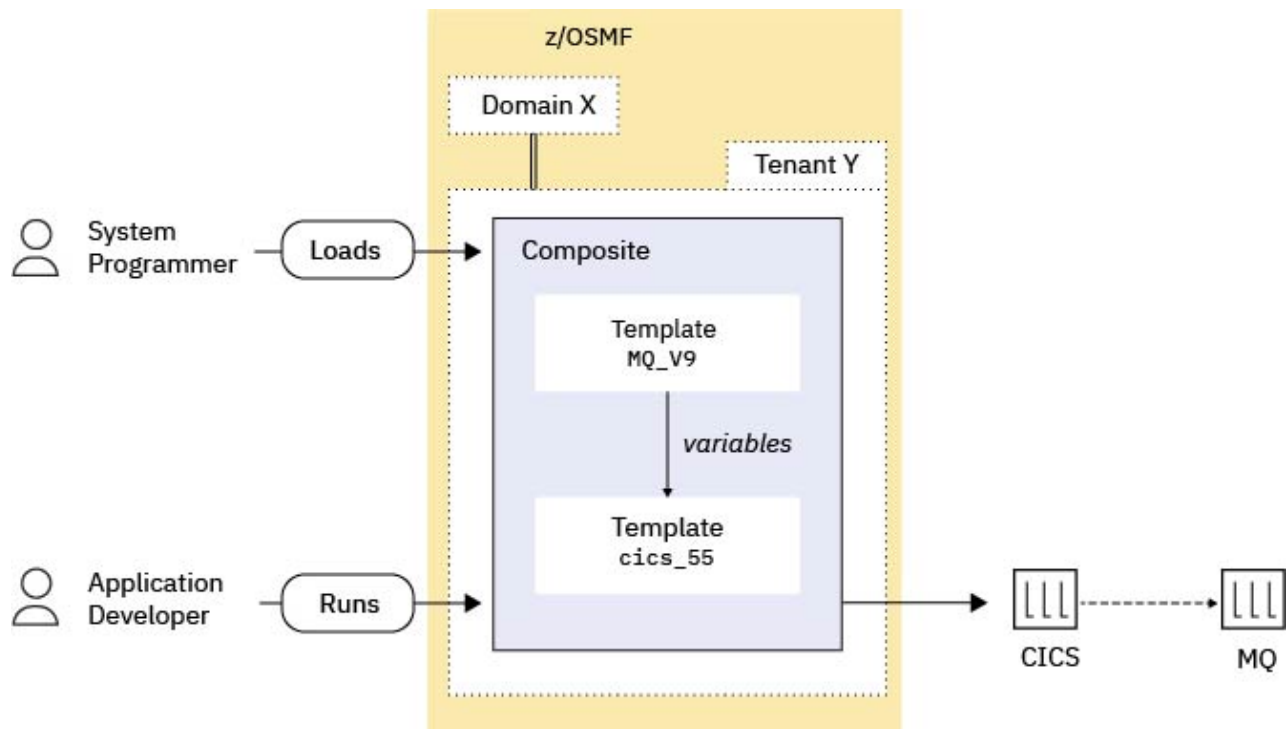


Figure 10. An IBM MQ and CICS composite template

You can name a composite template in a z/OS PT image and you can use the **zospt run** command to provision the composite template. Try out the templates that make up the composite template individually before you use them in a composite template.

For more information about composite templates, see Composite templates in z/OSMF Online Help.

When to use a composite template

You can provision a connected set of containers by using a composite template, or by provisioning separate containers and connecting them together by using multiple z/OS PT commands. A composite template might be suitable in the following situations:

- You need to provision multiple containers that must be provisioned and deprovisioned together.
- You want to define and manage the connected set of containers in z/OSMF, and present a single composite template that users can provision and deprovision.

You cannot use composite templates in the following situations:

- You are using z/OSMF 2.1.
- The IBM Cloud Provisioning and Management for z/OS 1.2 APARs are not applied.
- You want to build a z/OS PT image that contains files that need to be copied to the zFS file system for use by a provisioned container. You cannot use COPY instructions when you build an image for a composite template.

In these situations, consider issuing multiple z/OS PT commands to provision a connected set of linked containers, each of which uses a different z/OS PT image. For more information about how to link containers together, see “Using the --link option” on page 52. For more information about link scenarios that are supported, see “Link scenarios supported by z/OS Provisioning Toolkit” on page 54.

Creating a composite template

You can create a composite template by using z/OSMF, in a similar way to a standard (non-composite) template.

- For more information about how to create a composite template, see Add a template in z/OSMF Online Help.
- For more information about how to create a standard template in preparation for using z/OS PT, see Chapter 10, “Adding an IBM Cloud Provisioning and Management for z/OS template for use with z/OS Provisioning Toolkit,” on page 45.

Building an image that names a composite template

You build an image that names a composite template in a similar way to building an image that names a standard (non-composite) template. You add instructions to the `zosptfile`, then you use the **zospt build** command. The following example shows how you specify the composite template name in the `zosptfile` to build an image that names a composite template:

```
ENV ZOSMF_TEMPLATE=<composite_template_name>
```

To specify environment variables in the image and target them at individual templates in the composite template, prefix the environment variable name with the individual template name. For example, you created a composite template that provisions a CICS region that uses template `cics_55` and a z/OS Connect EE instance that uses template `zosconnect_v3r0`. If you need the CICS region to be provisioned with a Liberty JVM server, you can set the following environment variable in the image:

```
ENV cics_55.DFH_REGION_JVMSEVER=Liberty
```

Setting the environment variable in this way ensures that the **DFH_REGION_JVMSEVER** variable is passed into the **cics_55** template when the CICS region is provisioned.

You cannot use **COPY** instructions to copy files or directories into an image for a composite template. You can specify only the template name and environment variables.

For more information about building an image by using z/OS PT, see “Building an image with z/OS Provisioning Toolkit” on page 49.

Running an image that names a composite template

You can run an image that names a composite template in the same way that you can run an image that names a standard (non-composite) template, by using the **zospt run** command. The **--name** and **--quiet** options are supported on **zospt run** commands for an image that names a composite template. Differences in the output that is generated from the **zospt run** command are shown in the following example output:

```
/u/user/cics_demo:>zospt run test
2018-01-11 07:51:46 IBM z/OS Provisioning Toolkit V1.1.0
2018-01-11 07:51:46 Running image test.
2018-01-11 07:51:46 The z/OSMF template used is cics_zcee.
2018-01-11 07:51:46 The z/OSMF domain is default.
2018-01-11 07:51:46 The z/OSMF tenant is default.
2018-01-11 07:51:46 Connecting to z/OSMF on host system01.hursley.ibm.com port 32000.
2018-01-11 07:51:47 The z/OSMF template cics_zcee was created by USER01 at 2017-11-08T07:49:19.567Z.
The template type is composite. It was last modified by USER01 at 2017-11-08T07:49:19.567Z.
2018-01-11 07:51:47 The z/OSMF composite child template cics_user_test was created by USER01 at 2017-10-30T08:15:31.829Z.
Its position in the provisioning sequence is 1. It was last modified by USER01 at 2017-11-08T07:20:47.386Z.
2018-01-11 07:51:47 The z/OSMF composite child template zosconnect_nosec_user was created by USER01 at 2017-08-08T09:06:20.822Z.
Its position in the provisioning sequence is 2. It was last modified by USER01 at 2017-11-08T07:20:28.208Z.
2018-01-11 07:51:56 Creating composite container CZ_CZ00 with id cecaacb2-fcf0-4c6f-b7d1-8e246e82d76e on system MV01.
.
.
2018-01-11 07:54:54 Created container ZOSCONNECT_CICPD00K with id 6fcd76d-e9e1-4fc7-abfd-4e15c80cf48f on system MV01.
2018-01-11 07:54:54 Created composite container CZ_CZ00 with id cecaacb2-fcf0-4c6f-b7d1-8e246e82d76e on system MV01.
```

For more information about running an image by using z/OS PT, see “Running a z/OS Provisioning Toolkit image” on page 50.

Managing containers created from a composite template

You can manage containers that are created from a composite template in the same way that you manage containers that are created from a standard (non-composite) template. That is, you can see which containers are running, start and stop them, and deprovision them when they are no longer needed.

- Listing running containers.

Use the **zospt ps** command. The output of the command indicates whether a container is a **Composite_Parent**, a **Composite_Child**, or a **Standard** container, as shown in the following example **zospt ps** command output:

NAME	IMAGE	OWNER	CREATED	STATE	TEMPLATE	SYSTEM	CONTAINER TYPE
CZ_CZ00	test	user01	2018-01-11T07:51:47	provisioned	cics_zcee	MV01	Composite_Parent
CICS_CICPD00I	N/A	user01	2018-01-11T07:51:47	provisioned	cics_zcee:cics_user_test	MV01	Composite_Child
ZOSCONNECT_CICPD00K	N/A	user01	2018-01-11T07:51:47	provisioned	cics_zcee:zosconnect_nosec_user	MV01	Composite_Child

- Inspecting containers.

Use the **zospt inspect** command. The command output includes information about the relationship between the container and its parent or child containers, as shown in the following example:

```

/u/user:>zospt inspect ZOSCONNECT_CICPD00K
{
  "container-name": "ZOSCONNECT_CICPD00K",
  "container-id": "6fdb76d-e9e1-4fc7-abfd-4e15c80cf48f",
  "instance-name": "ZOSCONNECT_CICPD00K",
  "template": "zosconnect_nosec_user",
  "image_name": "N/A",
  "created-time": "2018-01-11T07:51:47",
  "type": "ZOSCONNECT",
  "system": "MV01",
  "owner": "user01",
  "state": "provisioned",
  "public_properties": {
    "INSTANCE_STATUS": "inactive",
    "ZCON_HTTP_PORT": "28665",
    "ZCON_INSTALL_DIR": "/usr/lpp/IBM/zosconnect/v3r0",
    "ZCON_RECOMMENDATION": "Export the environment variable WLP_USER_DIR=/u/user/mount/CICPD00K before you issue zosconnect commands against the server",
    "ZCON_SERVER_DIRECTORY": "/u/user/mount/CICPD00K/servers/CICPD00K"
  },
  "connected-containers": [{"container-name": "CICS_CICPD001"}],
  "composite-parent": "CZ_CZ00",
  "provisioning-sequence": "2"
}

```

- Starting and stopping containers.

Use the **zospt start** and **zospt stop** commands. You can specify Composite_Parent containers or Composite_Child containers. When you specify a Composite_Parent container, all containers are started or stopped in the correct sequence.

- Removing and deprovisioning containers.

Use the **zospt rm** or **zospt rm -f** command and specify a Composite_Parent container:

- The **zospt rm** command checks that all the Composite_Child containers are stopped. If all child containers are stopped, the command attempts to deprovision all containers. If any child containers are still active, the **zospt rm** command fails.
- The **zospt rm -f** command deprovisions all Composite_Child containers that are owned by the Composite_Parent container in the correct sequence.

The **zospt rm** and **zospt rm -f** commands are not supported for Composite_Child containers.

For more information about managing containers that are created by z/OS PT from standard templates, see Chapter 14, “Managing containers created by z/OS Provisioning Toolkit,” on page 113.

Exploring further

Stay in touch and look out for more information at CICS Developer Center. For more information about composite and standard templates, see Software Services in z/OSMF Online Help.

Chapter 17. Automating provisioning with z/OS Provisioning Toolkit

To allow the provision and deprovision of containers to be automated, z/OS PT can be configured so that no user input is needed at run time. You can specify options on the **zospt run** command to make it easier to retrieve the name of the provisioned containers for use in further automation steps.

Configuring z/OS PT for use in automation

Configure the following z/OS PT environment variables:

- Set the **zospt_pw** environment variable to the password of the user ID that runs the automation. The user ID that runs the automation must have the appropriate authority in z/OSMF to run the z/OS PT commands.
- If z/OS PT is configured to use a truststore (**truststore** environment variable), the truststore must already contain the certificate authorities that are needed to trust the z/OSMF server's certificate.

For more information about setting the **zospt_pw** and **truststore** environment variables, see Chapter 6, “Configuring z/OS Provisioning Toolkit,” on page 23. For information about the container name, including the maximum length and a list of acceptable characters, see the **zospt run** command in Chapter 19, “The **zospt** command syntax,” on page 135.

Obtaining the container name from a zospt run command

By default, the **zospt run** command reports the progress of the provisioning process by writing messages to STDOUT, which can make it difficult to capture the container name that is allocated dynamically to the provisioned container. To make it easier to obtain the container name, specify the **--quiet** option on the **zospt run** command:

```
/u/cicsusr:>zospt run cics_55 --quiet
CICS_CICPJ001
```

If the command is successful, only the container name is written to STDOUT. If the command fails, error information is written to STDERR.

Example script to retrieve the z/OS PT container name

In the following example, the script provisions a container and uses a variable that is called *mycontainer* to obtain the container name. In the following example, if the provision completes successfully with a return code zero, the script then inspects the provisioned container by using the obtained name:

```
#!/bin/sh
mycontainer=$(zospt run myimage --quiet)
if [ $? -eq 0 ]
then
    zospt inspect $mycontainer
fi
```

Specifying the container name on a **zospt run** command

Instead of receiving a dynamic container name from a **zospt run** command, you can specify the container name to use. For example, use the following command:

```
zospt run imageName --name containerName
```

If the automated script provisions multiple containers, the automation needs a way to generate a unique container name for each container that gets provisioned.

Example script that sets the z/OS PT container name

In the following example, the script provisions a container that is named *mycics* and then inspects the container:

```
#!/bin/sh
zospt run myimage --name mycics
zospt inspect mycics
```

Logging

z/OS PT commands emit informational messages to STDOUT and error messages to STDERR.

z/OS PT return codes

z/OS PT commands return the following exit codes:

- 0 = Success.
- 12 = An error occurred. If the error occurs during a **zospt run** command, the container is created and must be removed. If the error occurs on other commands, it can be possible to retry those commands. The container must be removed when it is no longer needed.
- 16 = An error occurred during a **zospt run** command before the container was created. The container does not need to be removed.
- 20 = An internal error occurred. For more information, see the z/OS PT logs and STDERR.

Chapter 18. Troubleshooting z/OS Provisioning Toolkit

Use these resources to find diagnostic information for z/OS PT with IBM Cloud Provisioning and Management for z/OS, and a link to dW Answers for answers to specific issues. If you cannot resolve a problem and need to contact IBM, see the following topic for how to get the information that is needed by IBM Support (also known as “MustGather” information).

Diagnosing problems with the getting started scenario

To diagnose problems when you use the getting started scenario, that is, use z/OS PT and z/OSMF workflows, see “Getting started with CICS by using z/OSMF workflows” on page 30.

The information that follows here helps you to diagnose problems when you use z/OS PT and IBM Cloud Provisioning and Management for z/OS.

Finding diagnostic information

Look in the following places for information about how z/OS PT ran:

- The console after you enter a **zospt** command. The console output indicates the result of the command.
- z/OS PT logs. The logs are in the /logs directory in the z/OS PT installation location and contain more debugging information, such as the exchanges between z/OS PT and z/OSMF. The most recent log file is named log_<user ID running zospt>.0.
- The z/OSMF web interface. z/OSMF maintains workflow information for failed workflows and this information is the primary source to identify why a workflow failed. Messages in the interface indicate the processing status. z/OSMF can surface messages that originate from elsewhere, such as z/OS itself or one of its components. z/OSMF messages always start with the characters “IZU”.
- JES job output of jobs that are submitted by z/OSMF as part of a workflow.

An example of how to debug a problem

In this example, the `cics.properties` file that is used to provision a CICS region contains a deliberate error. The template that contains the error was loaded into z/OSMF, approved, and associated with a tenant so that it was in a “Draft approved” state and ready to be run. A z/OS PT image, `cics_debug`, was built that named the z/OSMF template to use for provisioning.

To try this example, set the `DFH_LE_HLQ` property to an invalid value in the `cics.properties` file and load the template into z/OSMF.

You can use the following steps to diagnose the cause of the error.

1. Run the `cics_debug` image in draft mode by running z/OS PT command **zospt run cics_debug --draft**.

The output from this command shows the deliberate error and gives more information, including the container name (which is shown in bold in the code output):

```

ERROR: The request failed to complete.
IZUMF0150E: The job that was submitted in step "Validating access to CICS data sets" failed during
automation processing. Return code: "CC 0008" .
Instance Name          : CICS_USR01
Workflow Name          : CICS_USR01provision1515683080153
Current Step           : validateCICSVariables
Current Step Title      : Validating access to CICS data sets
Current Step Description : This step runs a JES job to validate as many as possible of the template
properties that are related to datasets. The job runs under the user ID specified by the DFH_ADMIN_TSO
property and includes checks for the existence of datasets required during provision, the authority to
access the datasets required during provision, and that there are not already datasets for the
provisioned region. If this step fails, see the JES job output to understand which properties or security
permissions need correcting. The source JCL for the JES job contains comments for each step in the job,
which describe what the step is checking and how to resolve any problems that are identified. If you are
viewing this description from within the workflows view in the z/OSMF UI, then you can view the JES job
by clicking on the Status tab for the workflow step.
DFH_CICS_TYPE : Unmanaged
DFH_REGION_APPLID : CICPUZZ0
Created container CICS_USR01 with id cfdd0d7a-7993-456e-ba18-a033c2f2567d on system MV29 in a failed
state. Use 'zospt rm --force CICS_USR01' to remove it.
Failed to run image cics_debug.

```

2. In z/OSMF, use the container name (CICS_USR01) to find the failed provision request. Log in to z/OSMF. Click **Cloud Provisioning** > **Software Services** and select the **Instances** tab. Enter the container name in the **Find** field to find the failed container instance.
3. Click the instance name (the name of the failed container, CICS_USR01) to show more information. Click the link to the workflow that was submitted to run the provision (shown in the following image).

IBM z/OS Management Facility

Software Services

Overview Templates **Instances**

Software Services Instances ▸ CICS_IVP01

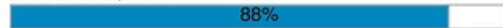
View CICS_IVP01

Instance Details	Type:	CICS
Metadata	Current system:	PLEX2.MV29 (MV29)
Variables	Provisioned system:	PLEX2.MV29 (MV29)
Actions	State:	Provisioning-Failed
History	Domain:	default
	Tenant:	default
	Template Owner:	tester
	Workflow:	CICS_IVP01provision1515683080153

4. The Workflow Steps table shows what the workflow was doing, and how far it got, before it failed. Look for the first **Failed** step. This step might be a parent step in the table, or a substep of an **In Progress** step.
If a step has substeps, you can use the expand icon (a plus symbol) next to the step title to see those substeps.

CICS_IVP01provision1515689348037

Description:
Procedure to provision a CICS Region
Percent complete:



Owner:
millwoo
Steps complete:
23 of 45

System:
PLEX2.MV29
Status:
 In Progress

Workflow Steps

Actions ▾					
⚙️ No filter applied					
<input type="checkbox"/>	State Filter	No. Filter	Title Filter	CalledWorkflow Filter	Automated Filter
<input type="checkbox"/>	✓ Skipped	1	■ Getting dynamic CICSplex name		Yes
<input type="checkbox"/>	✓ Skipped	2	+ Obtain CICSplex Information for the MAS CICS Region		
<input type="checkbox"/>	✓ Skipped	3	+ Defining MQ to the CICS region		
<input type="checkbox"/>	✓ Skipped	4	+ Defining DB2 to the CICS region		
<input type="checkbox"/>	✓ Skipped	5	■ Registering with WLM		Yes
<input type="checkbox"/>	✓ Skipped	6	■ Getting dynamic applid		Yes
<input type="checkbox"/>	✗ Failed	7	■ Validating access to CICS data sets		Yes
<input type="checkbox"/>	➡ Ready	8	■ Validating access to zFS directories		Yes

- In this example, the “Validating access to CICS data sets” step failed. Click the step title to view the Step Properties page, then click the Status tab to see what happened when the step ran.
- In this example, the “Validating access to CICS data sets” step ran some JCL. Steps might submit JCL, or might create REST requests to other components of z/OSMF.

Note: When workflow steps submit JCL, you can view the job output on the JES spool as an alternative to debugging by using z/OSMF.

- The JCL output in the following example shows that a Language Environment data set was not found.

IBM z/OS Management Facility

Welcome x Software Services x Resource Manage... x Workflows x

Workflows > CICS_IVP01provision1515683080153 > 7. Validating access to CICS data sets

Properties for Workflow Step 7. Validating access to CICS data sets

General Details Dependencies Notes Perform Status Input Variables Feedback

Name: IVP01 ID: JOB97089 Class: A Type: JOB Status: OUTPUT Return code: CC 0004

JESMSGLG JESJCL JESYSMSG SYSPLINT SYSPLINT **SYSPLINT** SYSPLINT SYSPLINT

DD name: SYSPLINT Step name: LEHLQ Procedure step name: Dataset ID: 109 Class: 0 Record count: 26

Output (1.252KB of 1.252KB shown)

```

IDCAMS SYSTEM SERVICES                                TIME: 15:04:47      01/11/18      PAGE
0
LISTCAT ENT('CEEBC.SCEECICS')
IDC3012I ENTRY CEEBC.SCEECICS NOT FOUND
IDC3009I ** VSAM CATALOG RETURN CODE IS 8 - REASON CODE IS IGG0CLEG-42
IDC1566I ** CEEBC.SCEECICS NOT LISTED
IDCAMS SYSTEM SERVICES                                TIME: 15:04:47      01/11/18      PAGE
0
THE NUMBER OF ENTRIES PROCESSED WAS:
0
AIX -----0
ALIAS -----0
CLUSTER -----0
DATA -----0
GDG -----0
INDEX -----0
NONVSAM -----0
PAGESPACE -----0
PATH -----0
SPACE -----0
USERCATALOG -----0

```

In this example, the marked text shows that the high-level qualifier (HLQ) for Language Environment data sets is incorrect. To resolve the problem, edit the `cics.properties` file and replace the HLQ for Language Environment data sets with the correct value.

8. You must refresh the template to use the changed `cics.properties` file. For instructions, see “Adding a CICS template by using IBM Cloud Provisioning and Management for z/OS” on page 59.
9. The failed provision attempt counts towards any limits set in z/OSMF for the number of instances that a user can provision. Therefore, ensure that containers that failed to provision are deprovisioned. In this example, to deprovision and remove the container, issue the z/OS PT command `zospt rm -f CICS_USR01`. This command generates some output, with the final message, in this example, Deleted container CICS_USR01.

Resolving problems with provisioning CICS regions

For help to resolve problems when CICS regions are provisioned by z/OS PT, see “Troubleshooting CICS provisioning with z/OS Provisioning Toolkit” on page 97.

Getting answers to specific questions on dW Answers

dW Answers is an open forum for questions and answers about IBM products. All questions about z/OS PT are tagged with **zospt.z/OS PT** questions might have more tags if they relate to a specific environment, for example **cics** if they are about CICS with z/OS PT. To see existing answers, or to ask a new question, go to IBM Developer Answers. To submit a new question, click **Ask a question**.

Contacting IBM Support

If you cannot resolve the problem yourself by using the diagnostic information that is provided or suggestions in IBM Developer Answers, contact IBM. The IBM

Software Support Handbook gives information about how to contact and work with IBM Support. The following information is helpful to IBM Support:

- The console output from a failed request.
- The logs that z/OS PT writes.
- The `zosmf.properties` file in the `/config` directory in the z/OS PT installation location.

Depending upon the type of error, IBM Support might ask for more information, such as:

- z/OSMF configuration settings.
- z/OSMF workflow output.
- The template that you used.
- Template properties.
- The z/OS PT image that you used.
- z/OS PT image contents.
- The logs that are written by the system that is being provisioned.

Chapter 19. The **zospt** command syntax

The **zospt** commands that are used by the z/OS Provisioning Toolkit, and the available command options, are listed.

Syntax

Usage: **zospt** [OPTIONS] COMMAND [arg...]

Options:

--version : Displays the command line version.
-h (--help) : Displays the command line help.

Commands:

build	PATH [-h (--help)] -t (--tag) <imageName>	Build an image
images	[-h (--help)]	List all images
inspect	<imageName> <containerName> <containerId> [-h (--help)]	Inspect an image or a container
rm	<containerName> <containerId> ... [-f (--force)] [-h (--help)]	Remove one or more containers
rmi	<imageName> ... [-h (--help)]	Remove one or more images
run	<imageName> [--draft] [--link <containerName> <containerId>:<alias>] [--name <containerName>] [-q (--quiet)] [-h (--help)]	Run an image in a new container
start	<containerName> <containerId> ... [-h (--help)]	Start one or more containers
stop	<containerName> <containerId> ... [-h (--help)]	Stop one or more containers
ps	[-a (--all)] [-f (--filter) <filter>] [-h (--help)]	List containers

Run '**zospt** COMMAND --help' for more information on a command.

Description

To show the syntax of the command, run **zospt command --help**. If a command fails, use the error messages in the console output and Chapter 18, “Troubleshooting z/OS Provisioning Toolkit,” on page 129 to resolve any issues.

build

Build an image that is used by the **zospt run** command to provision containers. The **build** command has the following parameters:

PATH

The path to the directory that contains the **zosptfile** to use in this build. This parameter is mandatory.

-t (--tag)

The name of the image that this command builds. You cannot use the following reserved characters:

: @ / \

This parameter is mandatory.

imageName

The name of the image to use as input to this build. This parameter is mandatory.

When you run the **zospt build** command, z/OS PT operates on local resources, and runs without connecting to z/OSMF.

images

List the images that are available to run, with their name, size, and date of creation at the command prompt. This command has no additional parameters.

When you run the **zospt images** command, z/OS PT operates on local resources, and runs without connecting to z/OSMF.

inspect

Return details, in JSON format, about a specified image or container. The **inspect** command has the following parameters:

imageName

The name of the image to be inspected. When you run the **zospt inspect imageName** command, z/OS PT operates on local resources, and runs without connecting to z/OSMF.

containerName

The name of the container to be inspected. When you run the **zospt inspect containerName** command, z/OS PT operates on remote resources and connects to z/OSMF.

containerId

The ID of the container to be inspected. When you run the **zospt inspect containerId** command, z/OS PT operates on remote resources and connects to z/OSMF.

It is mandatory to specify one of these parameters.

The following example shows the details that are returned for an image:

```
{
  "name": "cics_55_liberty",
  "created": "Wed Dec 12 06:45:51 GMT 2018",
  "size": "10240B",
  "FROM": "cics_55",
  "ENV": {
    "ZOSMF_TEMPLATE": "cics_55",
    "DFH_REGION_JVMSEVER": "Liberty",
    "IMAGE_LIBERTY_DIR": "workdir/DFHWLP/wlp/usr/servers/defaultServer"
  }
}
```

ps List the provisioned containers and containers for which provisioning failed. The **ps** command has the following parameters:

-a (--all)

Lists all the containers in any of the following states: being provisioned, provisioned, provisioning failed, being deprovisioned, deprovisioned, and deprovisioning failed. This parameter is optional.

-f (--filter) filter

Lists the returned containers, filtered by the owner, the state of the container, the system on which the container is running, or a combination of filters. The format of *filter* is:

[OWNER|STATE|SYSTEM]=value

value is the parameter value that you want to filter by.

For multiple filters, specify **-f** before each filter, for example:

```
zospt ps -f owner=user02 -f state=provisioned
```

When you run the **zospt ps** command, you are prompted to enter your password on the console to authorize the connection to z/OSMF.

rm Remove (or deprovision) one or more containers that were provisioned by using the **zospt run** command. The **rm** command has the following parameters:

containerName|containerId

The name or ID of the container that was returned as a result of a **zospt run** command. It is mandatory to specify one of these parameters. To remove more than one container, specify multiple container names or IDs, each separated by a space.

-f (--force)

This parameter is relevant only to provisioning a CICS container. Shuts down the CICS region. This parameter is optional.

When you run the **zospt rm** command, you are prompted to enter your password on the console to authorize the connection to z/OSMF. After authorization, z/OS PT uses z/OSMF workflows to remove or deprovision the specified container or containers. The command output returns the progress of the workflows while it runs in z/OSMF. Multiple jobs run as part of each command, so they might take several minutes to complete.

If parallel provisioning is enabled in the workflow for CICS templates that use Cloud Provisioning to deprovision a CICS region, some steps run in parallel so that deprovisioning completes faster. Output includes the message *This workflow contains parallel steps*. See “Enabling faster provisioning by using parallel processing” on page 63.

If you are deprovisioning a MAS or WUI region that was provisioned into an existing CICSplex, see “Deprovisioning guidelines for regions in a CICSplex” on page 74.

rmi

Remove one or more images. The **rmi** command has the following parameter:

imageName

The name of the image to use as input. To remove more than one image, specify multiple image names, each separated by a space. This parameter is mandatory.

run

Run an image that was built by using the **zospt build** command to create a new container. The **run** command has the following parameters:

imageName

The name of the image to run. This parameter is mandatory.

--draft

Run the z/OSMF template in *Draft approved* state. This parameter is optional. If you omit this parameter, the z/OSMF template must be in *Published* state.

--link containerName:alias | containerId:alias

Provision a container with a connection to one or more other containers. Specify the name or ID of the container to link to. Each container can have an optional alias, which can be used to identify the connection in the provisioned container.

You can specify the **--link** option multiple times in a single command.

--name containerName

The name of the container to run. This name must be unique. You can use this parameter to assign a meaningful name to a container, rather than

having a container name that is assigned automatically. The maximum name length is 32 characters. The acceptable characters are A-Z a-z 0-9 _ -. This parameter is optional.

-q (--quiet)

Do not write information messages to the console during the run. The only output that is written to stdout is the name of the provisioned container. If an error occurs, the error information is written to stderr.

When you run the **zospt run** command, you are prompted to enter your password on the console to authorize the connection to z/OSMF. After authorization, z/OS PT uses z/OSMF workflows to run the specified image. The command output returns the progress of the workflows while it runs in z/OSMF. Multiple jobs run as part of each command, so they might take several minutes to complete.

If parallel provisioning is enabled in the workflow for CICS templates that use Cloud Provisioning to provision a CICS TS unmanaged region, some steps run in parallel so that provisioning completes faster. Output includes the message This workflow contains parallel steps. See “Enabling faster provisioning by using parallel processing” on page 63.

start

Start one or more containers that were created by the **zospt run** command. The **start** command has the following parameters:

containerName|containerId

The name or ID of the container that was returned as a result of the **zospt run** command. Only one of these parameters is needed. To start more than one container, specify multiple container names or IDs, each separated by a space.

When you run the **zospt start** command, you are prompted to enter your password on the console to authorize the connection to z/OSMF. After authorization, z/OS PT uses z/OSMF workflows to start the specified container. The command line returns the progress of the workflows while it runs in z/OSMF. Multiple jobs run as part of each command, so they might take several minutes to complete.

stop

Stops one or more containers that were created by the **zospt run** command. The **stop** command has the following parameters:

containerName|containerId

The name or ID of the container that was returned as a result of the **zospt run** command. Only one of these parameters is needed. To stop more than one container, specify multiple container names or IDs, each separated by a space.

When you run the **zospt stop** command, you are prompted to enter your password on the console to authorize the connection to z/OSMF. After authorization, z/OS PT uses z/OSMF workflows to stop the specified container. The command output returns the progress of the workflows while it runs in z/OSMF. Multiple jobs run as part of each command, so they might take several minutes to complete.

Chapter 20. z/OS Provisioning Toolkit supplied templates

z/OS PT includes templates for provisioning CICS and z/OS Connect Enterprise Edition.

You can load templates into z/OSMF to enable the provisioning of runtime environments. Each template consists of a set of workflows that are run to provision an environment, or to act on a provisioned environment, for example, starting, stopping, or deprovisioning an environment. You manage templates in the IBM Cloud Provisioning and Management for z/OS component of z/OSMF. You can run templates by using z/OS PT.

z/OS PT supplied templates and workflows

z/OS PT supplies and maintains the templates and workflows to provision and deprovision the following products:

- CICS Transaction Server unmanaged region
- CICS system-managed single server (SMSS)
- CICSplex SM managed application system (MAS)
- CICSplex SM Web User Interface (WUI)
- CICSplex SM address space (CMAS)
- z/OS Connect Enterprise Edition V2 or V3 server

The templates and workflows can be customized for specific customer environments by setting properties in a properties file. Before you use a z/OS PT supplied template, review the list of customization properties:

- For CICS, see “Configuration properties for CICS images” on page 75.
- For z/OS Connect EE, see “Configuration properties for z/OS Connect Enterprise Edition images” on page 106.

If you cannot find the customization that you need, make a request by using the following link [Request a feature](#).

How to use the supplied templates

Consider the following things when you use the z/OS PT supplied templates:

- The supplied templates and workflows are written and maintained to provide reliable provisioning and deprovisioning of a subsystem. It is important to always run the deprovision action on a provisioned subsystem, regardless of whether the provisioning succeeded or failed. Deprovisioning provisioned subsystems ensures that the artifacts that were created during provision of the subsystem are deleted during deprovision, and that the LPAR is ready for further provisions to run.
- If you use the supplied templates and workflows without any modification, it is easier to upgrade to later versions when they become available. If you have any workflow modification suggestions or requests, make a request by using the following link [Request a feature](#).

Other workflows supported by z/OS PT

z/OS PT supports workflows to provision the following products:

- IBM MQ V9.0. The IBM MQ V9.0 workflow is supplied with IBM MQ V9.0. For more information, see [Using IBM z/OSMF to automate IBM MQ](#).
- WebSphere Liberty. To download the WebSphere Liberty workflow, follow this link: [WebSphere Liberty workflow in GitHub](#).
- IMS DB/TM system. To download the IMS workflow, follow this link: [IMS workflow in GitHub](#).

Chapter 21. Samples provided with z/OS Provisioning Toolkit

To give you a foundation for configuring templates and building images, z/OS PT provides examples of the configuration for commonly used environments, such as CICS Transaction Server for z/OS. The download package for z/OS PT provides sample templates and images, and the corresponding artifacts that z/OS PT uses to build these images.

Sample templates

Templates are configured by a properties file and added into z/OSMF to enable the provisioning of systems based on the templates. Example templates for provisioning CICS regions and instances of z/OS Connect EE are provided in the `zospt/templates` directory. Each example contains a `*.properties` file, which you must customize for your environment. To create a new template, copy the entire template directory to another directory in the `zospt/templates` directory, then update the properties file for the new template.

Many samples provision a CICS region at a specific release, for example CICS Transaction Server for z/OS V5.5. To use a sample with a different CICS release, change the following properties in the `cics.properties` file to use the release that you require:

- `DFH_CICS_HLQ`
- `DFH_CICS_LICENSE_DATASET`
- `DFH_CICS_USSHOME`

For details, see “Configuration properties for CICS images” on page 75.

Table 22. Sample CICS templates provided with z/OS PT

Sample template name	Details
<code>cics_getting_started</code>	A z/OSMF template that is configured with a minimal set of properties to enable the provisioning of a simple, single CICS Transaction Server for z/OS region. For a walk-through that uses the <code>cics_getting_started</code> template, see “Getting started with CICS by using z/OSMF workflows” on page 30.
<code>cics_52</code>	A z/OSMF template that is configured to provision a CICS Transaction Server for z/OS V5.2 region.
<code>cics_53</code>	A z/OSMF template that is configured to provision a CICS Transaction Server for z/OS V5.3 region.
<code>cics_54</code>	A z/OSMF template that is configured to provision a CICS Transaction Server for z/OS V5.4 region.
<code>cics_55</code>	A z/OSMF template that is configured to provision a CICS Transaction Server for z/OS V5.5 region.
<code>cmas_55</code>	A z/OSMF template that is configured to provision a CICSplex SM address space (CMAS) V5.5 region. ¹
<code>wui_55</code>	A z/OSMF template that is configured to provision a Web User Interface (WUI) V5.5 region. ¹
<code>zosconnect_v2r0</code>	A z/OSMF template that is configured to provision an instance of z/OS Connect Enterprise Edition V2.0.
<code>zosconnect_v3r0</code>	A z/OSMF template that is configured to provision an instance of z/OS Connect Enterprise Edition V3.0.

Note:

1. For details about the order in which to provision regions in a CICSplex, and how provisioned regions can connect to a CICSplex, see “Overview of supported CICS types” on page 58.

Sample images and the source artifacts that are used to build them

An image is a binary object that contains the configuration, application code, and environment variables to provision an environment. Properties for an image are specified in a `zosptfile` configuration file. Building a `zosptfile` creates a binary image that can then be used to provision applications and systems.

- Sample `zosptfile` configuration files are provided in the `zospt/samples` directory.
- Some of these samples are built into images that are provided with z/OS PT in the `zospt/images` directory.

To list the images in the `zospt/images` directory, you can use the **zospt images** command. Before you use a provided image, look at its `zosptfile` in the corresponding directory in `zospt/samples` to see how suitable it is for your use.

Table 23. Sample `zosptfile` configuration files provided with z/OS PT in the `zospt/samples` directory

Sample name	Details
<code>cics_getting_started_workflow</code>	A <code>zosptfile</code> to build an image to test the “Getting started with CICS by using z/OSMF workflows” on page 30 scenario. The image provisions the <code>cics_getting_started</code> z/OSMF template (which is configured with a minimal set of properties to enable provisioning of a CICS Transaction Server for z/OS region).
<code>cics_getting_started_cloud</code>	A <code>zosptfile</code> to build an image to test the “Getting started with CICS by using IBM Cloud Provisioning and Management for z/OS” on page 34 scenario. The image provisions the <code>cics_getting_started</code> z/OSMF template (which is configured with a minimal set of properties to enable provisioning of a CICS Transaction Server for z/OS region).
<code>cics_async_api_sample</code>	<p>A <code>zosptfile</code> that shows how to build an image that includes program binary files and CICS resource definitions for a sample CICS Asynchronous API application. The image requires a CICS Transaction Server for z/OS V5.4 or later region configured in the <code>cics.properties</code> file that is used in the <code>cics_getting_started_workflow</code> image.</p> <p>The sample includes CICS resource definitions in a DFHCSDUP input file. These definitions are added into the system definition data set (CSD) of each provisioned CICS region. The sample also includes program binary files. During CICS provisioning, these files are copied into a newly allocated PDSE and made available in the CICS region through an installed LIBRARY resource.</p>
<code>cics_52</code>	A <code>zosptfile</code> that identifies the name of the z/OSMF template (<code>cics_52</code>) to drive to provision a region that runs CICS Transaction Server for z/OS V5.2.
<code>cics_53</code>	A <code>zosptfile</code> that identifies the name of the z/OSMF template (<code>cics_53</code>) to drive to provision a region that runs CICS Transaction Server for z/OS V5.3.
<code>cics_54</code>	A <code>zosptfile</code> that identifies the name of a z/OSMF template (<code>cics_54</code>) to drive to provision a region that runs CICS Transaction Server for z/OS V5.4.
<code>cics_55</code>	A <code>zosptfile</code> that identifies the name of a z/OSMF template (<code>cics_55</code>) to drive to provision a region that runs CICS Transaction Server for z/OS V5.5.

Table 23. Sample *zosptfile* configuration files provided with z/OS PT in the *zospt/samples* directory (continued)

Sample name	Details
cics_55_json	A <i>zosptfile</i> that defines an Axis2 JVM server and associated service provider pipeline based on CICS Transaction Server for z/OS V5.5. The environment variable <code>IMAGE_WSDIR</code> is defined, which can be used to copy WSBind files into the WSBind pickup directory for the service provider pipeline.
cics_55_json_microservice	A <i>zosptfile</i> that shows how to build an image that contains a JSON microservice that uses an Axis2 JVM server and associated service provider pipeline based on CICS Transaction Server for z/OS V5.5. The sample uses the environment variable <code>IMAGE_WSDIR</code> to copy WSBind files into the WSBind pickup directory for the service provider pipeline. The environment variable is defined in the <code>cics_55_json</code> image that this sample extends.
cics_55_liberty	A <i>zosptfile</i> that defines a CICS Transaction Server for z/OS V5.5 region with an embedded Liberty JVM server. The environment variable <code>IMAGE_LIBERTY_DIR</code> is defined, which can be used to copy configuration files for the Liberty JVM server into the correct directory.
cics_55_nodejs_ivp	<p>A <i>zosptfile</i> that shows how to build an image that includes a CICS bundle containing a Node.js IVP application. The image requires a CICS Transaction Server for z/OS V5.5 region configured in the <code>cics.properties</code> file that is used in the <code>cics_getting_started_workflow</code> image. By default, the Node.js IVP application listens for requests on port 9080. If this port number is unsuitable, you can change the port number before building an image from this sample.</p> <p>To change the port number, either edit the file <code>zospt/samples/cics_55_nodejs_ivp/bundles/nodejsivp/profiles/ivp_sample.profile</code> or, if the file is read-only, copy the <code>zospt/samples/cics_55_nodejs_ivp</code> directory and edit the copied <code>ivp_sample.profile</code> file.</p>
cics_55_soap	A <i>zosptfile</i> that defines a service provider pipeline that is configured for SOAP web services based on CICS Transaction Server for z/OS V5.5. The environment variable <code>IMAGE_WSDIR</code> is defined, which can be used to copy WSBind files into the WSBind pickup directory for the service provider pipeline.
cics_55_soap_microservice	A <i>zosptfile</i> that shows how to build an image that contains a SOAP microservice that uses a service provider pipeline that is configured for SOAP web services based on CICS Transaction Server for z/OS V5.5. The sample uses the environment variable <code>IMAGE_WSDIR</code> to copy WSBind files into the WSBind pickup directory for the service provider pipeline. The environment variable is defined in the <code>cics_55_soap</code> image that this sample extends.
cics_55_zosconnect_v3r0	A <i>zosptfile</i> that defines a CICS Transaction Server for z/OS V5.5 region with embedded z/OS Connect Enterprise Edition V3.0. The sample sets the environment variable <code>DFH_ZOSCONNECT_USERID</code> to provide a default user ID for z/OS Connect EE tasks, which run in CICS. This environment variable replaces any value that is specified for the <code>DFH_ZOSCONNECT_USERID</code> property in the z/OSMF template that provisions the underlying CICS region. The sample defines two further environment variables, <code>IMAGE_APIS</code> and <code>IMAGE_SERVICES</code> , which can be used to copy z/OS Connect EE API archive files (AAR files) and service archive files (SAR files) to the appropriate directories so that they are installed when the z/OS Connect EE instance is started.
cics_55_zosconnect_v3r0_api	A <i>zosptfile</i> that shows how to build an image that contains z/OS Connect EE V3.0 API archive files (AAR files) that are deployed into a CICS Transaction Server for z/OS V5.5 region with embedded z/OS Connect EE V3.0. The sample uses the environment variable <code>IMAGE_APIS</code> to copy AAR files into the appropriate directory so that they are installed when the z/OS Connect EE instance is started. The environment variable is defined in the <code>cics_55_zosconnect_v3r0</code> image that this sample extends.

Table 23. Sample *zosptfile* configuration files provided with z/OS PT in the *zospt/samples* directory (continued)

Sample name	Details
cics_55_zosconnect_v3r0_services	A <i>zosptfile</i> that shows how to build an image that contains z/OS Connect EE V3.0 service archive files (SAR files) that are deployed into a CICS Transaction Server for z/OS V5.5 region with embedded z/OS Connect EE V3.0. The sample uses the environment variable <code>IMAGE_SERVICES</code> to copy SAR files into the appropriate directory so that they are installed when the z/OS Connect EE instance is started. The environment variable is defined in the <code>cics_55_zosconnect_v3r0</code> image that this sample extends.
cmas_55	A <i>zosptfile</i> that identifies the name of a z/OSMF template (<code>cmas_55</code>) to drive to provision a CICSplex SM address space (CMAS) V5.5 region.
wui_55	A <i>zosptfile</i> that identifies the name of a z/OSMF template (<code>wui_55</code>) to drive to provision a Web User Interface (WUI) V5.5 region.
zosconnect_v2r0	A <i>zosptfile</i> that identifies the name of a z/OSMF template (<code>zosconnect_v2r0</code>) to drive to provision an instance of z/OS Connect EE V2.0. The sample defines an environment variable <code>IMAGE_APIS</code> , which can be used to copy z/OS Connect EE API archive files (AAR files) to the appropriate directory so that they are installed when the z/OS Connect EE instance is started.
zosconnect_v3r0	A <i>zosptfile</i> that identifies the name of a z/OSMF template (<code>zosconnect_v3r0</code>) to drive to provision an instance of z/OS Connect EE V3.0. The sample defines two environment variables, <code>IMAGE_APIS</code> and <code>IMAGE_SERVICES</code> , which can be used to copy z/OS Connect EE API archive files (AAR files) and service archive files (SAR files) to the appropriate directories so that they are installed when the z/OS Connect EE instance is started.

Table 24. Sample images provided with z/OS PT in the *zospt/images* directory

Sample image name	Details
cics_getting_started_workflow	An example image to try out the “Getting started with CICS by using z/OSMF workflows” on page 30 scenario, by provisioning the <code>cics_getting_started</code> z/OSMF template through z/OS PT.
cics_getting_started_cloud	An example image to try out the “Getting started with CICS by using IBM Cloud Provisioning and Management for z/OS” on page 34 scenario, by provisioning the <code>cics_getting_started</code> z/OSMF template through z/OS PT.
cics_async_api_sample	<p>Uses the configuration from the <code>cics_getting_started_workflow</code> image and extends it to provision a CICS Transaction Server for z/OS V5.4 or later region that runs a sample CICS Asynchronous API application.</p> <p>The sample includes CICS resource definitions in a DFHCSDUP input file. These definitions are added into the system definition data set (CSD) of each provisioned CICS region. The sample also includes program binary files. During CICS provisioning, these files are copied into a newly allocated PDSE and made available in the CICS region through an installed LIBRARY resource.</p> <p>For more information about the sample application, see IBM CICS Asynchronous API: Concurrent Processing Made Simple. For the application source, see CICS Asynchronous API Redbooks Examples in GitHub.</p> <p>The image requires a CICS Transaction Server for z/OS V5.4 or later region configured in the <code>cics.properties</code> file that is used in the <code>cics_getting_started_workflow</code> image.</p>
cics_52	Defines the z/OSMF template (<code>cics_52</code>) to drive to provision a region that runs CICS Transaction Server for z/OS V5.2.
cics_53	Defines the z/OSMF template (<code>cics_53</code>) to drive to provision a region that runs CICS Transaction Server for z/OS V5.3.

Table 24. Sample images provided with z/OS PT in the `zospt/images` directory (continued)

Sample image name	Details
cics_54	Defines the z/OSMF template (cics_54) to drive to provision a region that runs CICS Transaction Server for z/OS V5.4.
cics_55	Defines the z/OSMF template (cics_55) to drive to provision a region that runs CICS Transaction Server for z/OS V5.5.
cics_55_json	Uses the configuration from the cics_55 image and extends it with an Axis2 JVM server and associated service provider pipeline. The environment variable IMAGE_WSDIR is defined in the image, which can be used in images that are built from the cics_55_json image, to copy WSBind files into the WSBind pickup directory for the service provider pipeline.
cics_55_liberty	Uses the configuration from the cics_55 image and extends it with an embedded Liberty JVM server. The environment variable IMAGE_LIBERTY_DIR is defined in the image, which can be used in images that are built from the cics_55_liberty image, to copy configuration files for the Liberty JVM server into the correct directory.
cics_55_nodejs_ivp	Inherits the configuration from the cics_getting_started_workflow image and extends the configuration by including a CICS bundle that contains a Node.js IVP application that listens on port 9080. The bundle is installed automatically during startup of the CICS region. The image requires a CICS Transaction Server for z/OS V5.5 region configured in the cics.properties file that is used in the cics_getting_started_workflow image.
cics_55_soap	Inherits the configuration from the cics_55 image and extends it with a service provider pipeline that is configured for SOAP web services. The environment variable IMAGE_WSDIR is defined in the image, which can be used in images that are built from the cics_55_soap image, to copy WSBind files into the WSBind pickup directory for the service provider pipeline.
cics_55_zosconnect_v3r0	Inherits the configuration from the cics_55_liberty image and extends it by configuring the embedded Liberty JVM server to run z/OS Connect EE. The image includes the environment variable DFH_ZOSCONNECT_USERID, which provides a default user ID for z/OS Connect EE tasks that run in CICS. This environment variable replaces any value that is specified for the DFH_ZOSCONNECT_USERID property in the z/OSMF template (cics_55) that provisions the underlying CICS region. The image includes two further environment variables, IMAGE_APIS and IMAGE_SERVICES, which can be used in images that are built from the cics_55_zosconnect_v3r0 image, to copy z/OS Connect EE API archive files (AAR files) and service archive files (SAR files) to the appropriate directories so that they are installed when the z/OS Connect EE instance is started.
cmas_55	Defines the z/OSMF template (cmas_55) to drive to provision a CICSplex SM address space (CMAS) V5.5 region.
wui_55	Defines the z/OSMF template (wui_55) to drive to provision a Web User Interface (WUI) V5.5 region.
zosconnect_v2r0	Defines the z/OSMF template (zosconnect_v2r0) to drive to provision an instance of z/OS Connect EE V2.0. The image defines the environment variable IMAGE_APIS, which can be used in images that are built from the zosconnect_v2r0 image, to copy z/OS Connect EE API archive files (AAR files) to the appropriate directory so that they are installed when the z/OS Connect EE instance is started.

Table 24. Sample images provided with z/OS PT in the *zospt/images* directory (continued)

Sample image name	Details
zosconnect_v3r0	Defines the z/OSMF template (zosconnect_v3r0) to drive to provision an instance of z/OS Connect EE V3.0. The image defines two environment variables, IMAGE_APIS and IMAGE_SERVICES, which can be used in images that are built from the zosconnect_v3r0 image, to copy z/OS Connect EE API archive files (AAR files) and service archive files (SAR files) to the appropriate directories so that they are installed when the z/OS Connect EE instance is started.

Chapter 22. Provisioning a CICS region through IBM Cloud Provisioning and Management for z/OS

You can use the templates that are supplied with z/OS PT to provision a CICS region through IBM Cloud Provisioning and Management for z/OS but without running z/OS PT. For example, you might use this approach if you are working with IBM z/OS Cloud Broker.

Before you begin

Before you can provision a CICS region by using a template that is supplied with z/OS PT, complete the following checks:

- Ensure that z/OSMF is installed and configured. See Configuration in the z/OSMF Configuration Guide.

For summary information about z/OSMF, see Chapter 3, “Prerequisites for using z/OS Provisioning Toolkit,” on page 17.

- Ensure that your user ID has sufficient authority to create an environment, load templates into z/OSMF, and run workflows. See Chapter 7, “Security planning for z/OS Provisioning Toolkit with IBM Cloud Provisioning and Management for z/OS,” on page 25 and “Security requirements for provisioning CICS with z/OS Provisioning Toolkit” on page 72.

For more information about setting up security for IBM Cloud Provisioning and Management for z/OS, see Help with security setup in IBM Z solutions product documentation.

- Ensure that IBM Cloud Provisioning and Management for z/OS is installed and configured. See Preparing to use Cloud Provisioning in the z/OSMF Configuration Guide.

About this task

z/OS PT is packaged as a compressed file that you can download without charge. You can access the templates that are supplied with z/OS PT without fully installing it.

For information about the templates, see Chapter 20, “z/OS Provisioning Toolkit supplied templates,” on page 139. For a list of templates that are supplied with z/OS PT, see the *Sample templates* section of Chapter 21, “Samples provided with z/OS Provisioning Toolkit,” on page 141.

You can customize the templates to local naming standards and operating procedures by using the configuration properties that are supplied with the templates. You can manage and control the templates and the workflows that they drive by using IBM Cloud Provisioning and Management for z/OS.

Procedure

1. Find z/OS PT for download on IBM z/OS Provisioning Toolkit product page.
2. Extract the .zip file.
3. Transfer the .pax file in binary format to the z/OS location where you want to store your templates.
4. Issue the following command to extract the .pax file:

```
pax -rf zospt_v#####.pax
```

The zospt installation directory is created.

5. Identify the template that you require in the zospt/templates directory, for example zospt/templates/cics_55. For a description of each template, see the *Sample templates* section of Chapter 21, “Samples provided with z/OS Provisioning Toolkit,” on page 141.

To create a new template, copy the entire template directory to another directory in the zospt/templates directory.

6. Edit the cics.properties file in the directory for your chosen template to configure the CICS region for the release you require (V5.1 or later) and to meet your naming standards. See “Configuration properties for CICS templates.”
7. Add the template to z/OSMF through the z/OSMF web interface. For details, see “Adding a CICS template by using IBM Cloud Provisioning and Management for z/OS” on page 59, and for example steps with screenshots, see “Getting started with CICS by using IBM Cloud Provisioning and Management for z/OS” on page 34.
8. Optional: You can use z/OS PT to run and evaluate your template before you publish it for wider use. To do this, complete all steps in Chapter 4, “Installing z/OS Provisioning Toolkit,” on page 19 and see “Adding a CICS template by using IBM Cloud Provisioning and Management for z/OS” on page 59.

What to do next

After you add the template to z/OSMF and it is ready to publish, you can use a suitable method to run the template, for example IBM z/OS Cloud Broker.

Configuration properties for CICS templates

You can configure some properties to customize the way that CICS is provisioned through IBM Cloud Provisioning and Management for z/OS. These properties are set when you configure a template, and are specified in a cics.properties ISO8859-1 file, ready to load into z/OSMF.

If you provision CICS by using z/OS PT, see “Configuration properties for CICS images” on page 75, because more configuration options might apply.

Type of CICS region

Property	Description
DFH_CICS_TYPE	<p>To set your CICS region type, use one of the following available options:</p> <ul style="list-style-type: none">• CMAS - A single CICSplex SM address space (CMAS) is provisioned, and a CICSplex is defined to the data repository of the CMAS. You must set extra properties for this region type, as described in “CICSplex SM properties” on page 159.• MAS - A managed application system (MAS) region is managed by CICSplex SM and participates in a CICSplex. You must set extra properties for this region type, as described in “CICSplex SM properties” on page 159.• SMSS - A stand-alone CICS region (SMSS) that can be managed through CMCI. Although the region has CICSplex SM libraries, it does not require a CMAS or participate in a CICSplex.• Unmanaged - A stand-alone CICS region that cannot be managed through CMCI.• WUI - A Web User Interface (WUI) region must be connected to a CMAS region of the same version. You must set extra properties for this region type, as described in “CICSplex SM properties” on page 159.

High-level qualifiers for your installation

Property	Description
DFH_CICS_HLQ	High-level qualifier (HLQ) of the CICS installation location. DFH_CICS_HLQ is used to define the STEPLIB and DFHRPL concatenations for provisioned CICS regions. If the CICS region needs to access modules in the MVS link pack area (LPA), specify the SIT parameter LPA=YES in the compiled SIT named by the DFH_REGION_SIT property, or in the DFH_REGION_SITPARMS property itself.
DFH_CPSM_HLQ	High-level qualifier (HLQ) of the CICSplex SM installation location.
DFH_LE_HLQ	High-level qualifier (HLQ) of Language Environment.

User IDs for the configuration

For more information about the steps for which each user ID is used, see “Security requirements for provisioning CICS with z/OS Provisioning Toolkit” on page 72.

Property	Description
DFH_ADMIN_TSO	TSO user ID that is used for steps that require a greater level of authority; for example, creating data sets.

Property	Description
DFH_ADMIN_CONSOLE	<p>User ID that is used to issue MVS console commands. Commands are issued by using REXX scripts or z/OSMF z/OS console services, depending on the <code>CONSOLE_COMMAND_TYPE</code> property. See “Workflow configuration variables” on page 160.</p> <p>The authorizations that the <code>DFH_ADMIN_CONSOLE</code> user ID requires depend on whether REXX scripts or z/OSMF z/OS console services are used. See “Security requirements for provisioning CICS with z/OS Provisioning Toolkit” on page 72 and z/OS console services in the z/OSMF Programming Guide.</p> <p>When z/OS console services are used, an MVS console is created and used to submit console commands. The MVS console name is the first six characters of the <code>DFH_ADMIN_CONSOLE</code> user ID, followed by the letters CN. Ensure that system messages are routed to the console so that the start and stop of the CICS region can be detected.</p> <p>When the z/OS console services are used, ensure that messages DFHSI1517, EYUNL0099I, EYUXL0010I, DFHKE1799, and IXG661I are not suppressed by the Message Processing Facility (MPF). These messages are used to detect when different types of CICS regions start and stop, and when a CICS region's log streams are deleted successfully.</p> <p>If multiple z/OSMF servers are installed on different LPARs in a sysplex, the generated MVS console name (the first six characters of the <code>DFH_ADMIN_CONSOLE</code> user ID) must be unique on each LPAR.</p>
DFH_ADMIN_ZFS	User ID that has zFS authority to mount and unmount zFS file systems.
DFH_ADMIN_SECURITY	<p>User ID that has authorization to make dynamic security profiles changes in the Security Manager.</p> <p>By default, no dynamic security updates are made, because the REXX scripts that are run exit immediately with return code zero. This default behavior allows for predefined security profiles to be used. The REXX scripts can be modified to issue commands to dynamically create, update, and delete the security profiles that are needed for the provisioned CICS region. For more information, see “Modifying the CICS workflows to support dynamic security” on page 73.</p>
DFH_APPROVER_TSO	Approval user ID for TSO.
DFH_APPROVER_CONSOLE	Approval user ID for the <code>DFH_ADMIN_CONSOLE</code> user ID to be used.
DFH_APPROVER_SECURITY	Approval user ID for dynamic security configuration.
DFH_APPROVER_ZFS	Approval user ID for zFS.

CICS region configuration

Property	Description
DFH_REGION_HLQ	High-level qualifier (HLQ) for all CICS region data sets.

Property	Description
DFH_STC_ID	<p>The user ID with which CICS starts.</p> <p>CICS runs as a started task and the job name matches the APPLID of the CICS region. A SAF security profile must exist that ensures the started task runs under the DFH_STC_ID user ID. This profile can be either predefined, or created dynamically as part of the provisioning process. For more information, see “Modifying the CICS workflows to support dynamic security” on page 73.</p> <p>The provisioned CICS region creates log streams under the DFH_STC_ID user ID.</p>
DFH_REGION_LOGSTREAM	<p>Indicates how the log streams that the provisioned CICS region requires are created. Valid options are:</p> <ul style="list-style-type: none"> • DYNAMIC: The log streams that the CICS region requires are created during the provisioning process, and deleted during the deprovisioning process. DYNAMIC is the default value. • EXISTING: The log streams (or the appropriate log stream models) that the CICS region requires must already exist. If JOURNALMODEL resources are required, they must be defined already on a shared system definition data set (CSD), or they can be added to a newly created CSD by using the DFH_REGION_CSD_INPUT property. • DUMMY: The CICS region does not use any log streams, and the CICS region must be started by using an initial start.
DFH_REGION_LOGSTREAM_HLQ	<p>A high-level qualifier (up to 8 characters) for the CICS log stream names. This property is optional.</p> <ul style="list-style-type: none"> • If you specify a value, the log stream prefix has the following format: <i>DFH_REGION_LOGSTREAM_HLQ.DFH_REGION_APPLID.*</i> • If you specify the property, but with no value (for example, <i>DFH_REGION_LOGSTREAM_HLQ=</i>), the log stream prefix has the following format: <i>DFH_REGION_APPLID.*</i> • If you do not specify the property, the user ID of the running CICS region, which is specified by the DFH_STC_ID property, is used in the log stream prefix. For example, <i>DFH_STC_ID.DFH_REGION_APPLID.*</i>
DFH_LOG_HLQ	<p>The high-level qualifier for log stream data set names. This property is optional. If the property is not specified, the high-level qualifier IXGLOGR is used.</p>
DFH_REGION_CICSSVC	<p>The CICS SVC number that is installed on the MVS LPAR.</p>
DFH_REGION_DFLTUSER	<p>User ID used as the default ID for the CICS region.</p>
DFH_REGION_VTAMNODE	<p>Name of the VTAM node to activate when the CICS region starts up.</p>
DFH_REGION_REGION	<p>JCL region size for the CICS JCL, in the format <i>nnnnM</i> (for example, 400M).</p>
DFH_REGION_MEMLIMIT	<p>JCL MEMLIMIT size for the CICS JCL, in the format <i>nnG</i> (for example, 16G).</p>
DFH_CICS_LICENSE_DATASET	<p>Location of the CICS license data set.</p>

z/OS configuration options

Property	Description
DFH_ZOS_PROCLIB	<p>Names the PROCLIB data set into which the workflow can create a procedure for the CICS region. The CICS region runs as a started task. If you specify a PROCLIB that is a PDSE, it enables concurrent addition of new procedures to the PROCLIB. If a PDS is used and multiple concurrent updates to the PROCLIB are attempted, z/OS abends them with a 213-30 abend code, to avoid data corruption.</p> <p>When a procedure is created in the PROCLIB, its name will match the APPLID of the CICS region that is provisioned. Therefore, use a suitable naming convention for the CICS region APPLIDs to ensure that the procedure names are unique and do not clash with any existing procedure names. If the procedure name is not unique, another started task with the same name might be stopped during deprovisioning of the CICS region. Ensure that the named PROCLIB data set is included in the PROCLIB concatenation on your system so that it can be found when the command to start the CICS region is issued.</p>
DFH_ZOS_STCJOBS	Names the data set into which the workflow can create the source JCL that defines the job level characteristics of the started task for the provisioned CICS region. Used only when DFH_STC_JOB_CARD is also specified. If you specify a data set of type PDSE, it enables concurrent addition of new STCJOBS to the data set. If a data set of type PDS is used and multiple concurrent updates to the data set are attempted, z/OS produces a 213-30 abend code, to avoid data corruption.
DFH_STC_JOB_CARD	Specifies the job card to use to define the job level characteristics of the started task for the provisioned CICS region. When specified, DFH_ZOS_STCJOBS must also be used to indicate the data set for STCJOBS.
DFH_ZOS_VSAM_UNIT	UNIT value that is used in creation of data sets, for example: SYSDA, 3390.
DFH_ZOS_VSAM_VOLUME	<p>VOLUME value that is used in the creation of PDS and VSAM data sets, for example, SYSDA, trace, or dump.</p> <p>To use SMS (Storage Management Subsystem) support, specify SMS. In this situation, the VOLUME parameter is omitted from the PDS or VSAM creation.</p>

zFS configuration options

Property	Description
DFH_ZFS_DATACLASS	<p>DATACLASS (a list of data set allocation attributes and their values) defined on z/OS that is used to create the zFS data sets. For more information about setup details, see “Configuration prerequisites for using z/OS Provisioning Toolkit with CICS” on page 57.</p> <p>For non-SMS-managed systems, either remove the DFH_ZFS_DATACLASS property, or do not give the property a value. Specify a value for DFH_ZOS_VSAM_VOLUME instead.</p>
DFH_ZFS_GROUP	Specifies the name of a SAF group, which is granted read, write, and execute authority to the CICS region zFS directory and contents.
DFH_ZFS_MOUNTPOINT	zFS directory that is used to mount the zFS directories for a CICS region. This property is optional. If omitted, no zFS directories are created for the provisioned CICS region.
DFH_CICS_USSHOME	Location of the CICS USSHOME installation.
DFH_CICS_USSHOME	Location of the CICS USSHOME installation.

Property	Description
DFH_ZCEE_INSTALL_DIR	The runtime directory in the installation directory of z/OS Connect EE; for example, /usr/lpp/IBM/zosconnect/v3r0/runtime. This property is relevant only for provisioning images for CICS and z/OS Connect EE.

JVM configuration options

Property	Description
DFH_JAVA_HOME	Location of the Java installation
DFH_REGION_JVMSERVER	The type of JVM server to provision. The following values are allowed: <ul style="list-style-type: none"> Liberty - provisions a Liberty JVM server named DFHWLP. OSGi - provisions an OSGi JVM server named DFHOSGI. Axis2 - provisions an Axis2 JVM server named DFHJVMAX. None - does not provision a JVM server.
DFH_JVM_DEBUG	Dynamically allocate a port for the remote debugging of a JVM server. If DFH_REGION_JVMSERVER is set to Liberty, OSGi, or Axis2, the JVM server is configured to use the allocated port. Valid values are: <ul style="list-style-type: none"> YES NO If not set, no JVM debug port is allocated.
DFH_ZOSCONNECT_USERID	The default user ID for z/OS Connect EE.

Node.js configuration options

Support for Node.js requires CICS Transaction Server for z/OS V5.5 or later.

Property	Description
DFH_NODE_HOME	Location of the Node.js installation directory on zFS. When this property is set, a nodejsprofiles/general.profile file is created in the directory specified by the DFH_CICS_USSCONFIG property for use by the Node.js application. The general.profile file contains entries for the WORK_DIR and NODE_HOME parameters. See Node.js profile and command line options in CICS TS product documentation.

Networking variables

Property	Description
DFH_JVM_DEBUG	Dynamically allocate a port for the remote debugging of a JVM server. If DFH_REGION_JVMSERVER is set to Liberty, OSGi, or Axis2, the JVM server is configured to use the allocated port. Valid values are: <ul style="list-style-type: none"> YES NO If not set, no JVM debug port is allocated.

Property	Description
DFH_JVM_DEBUG_USAGETYPE	<p>Optional property to set the usage type for the port that a Java debugger can be connected to, so that the port is dynamically allocated from the port allocation range that this usage type identifies.</p> <p>A valid usage type name is 1-32 characters and the following valid characters: A-Z a-z 0-9</p> <p>Typically, a network administrator uses z/OSMF to configure a port allocation range and specify the usage type for that range.</p> <p>This property is used if DFH_JVM_DEBUG is set to YES.</p>
DFH_REGION_CMCIPTORT_USAGETYPE	<p>Optional property to set the usage type for a CMCI port (used for connection to CICS Explorer) so that the port is dynamically allocated from the port allocation range that this usage type identifies.</p> <p>A valid usage type name is 1-32 characters and the following valid characters: A-Z a-z 0-9</p> <p>Typically, a network administrator uses z/OSMF to configure a port allocation range and specify the usage type for that range.</p>
DFH_REGION_HTTP	<p>Dynamically allocate an HTTP port during provision. Valid values are:</p> <ul style="list-style-type: none"> • YES • NO <p>If not set, no HTTP port is allocated.</p>
DFH_REGION_HTTP_USAGETYPE	<p>Optional property to set the usage type for an HTTP port so that the port is dynamically allocated from the port allocation range that this usage type identifies.</p> <p>A valid usage type name is 1-32 characters and the following valid characters: A-Z a-z 0-9</p> <p>Typically, a network administrator uses z/OSMF to configure a port allocation range and specify the usage type for that range.</p> <p>This property is used if DFH_REGION_HTTP is set to YES.</p>
DFH_REGION_HTTPS	<p>Dynamically allocate an HTTPS port during provision. Valid values are:</p> <ul style="list-style-type: none"> • YES • NO <p>If not set, no HTTPS port is allocated.</p>
DFH_REGION_HTTPS_USAGETYPE	<p>Optional property to set the usage type for an HTTPS port so that the port is dynamically allocated from the port allocation range that this usage type identifies.</p> <p>A valid usage type name is 1-32 characters and the following valid characters: A-Z a-z 0-9</p> <p>Typically, a network administrator uses z/OSMF to configure a port allocation range and specify the usage type for that range.</p> <p>This property is used if DFH_REGION_HTTPS is set to YES.</p>

Property	Description
DFH_REGION_IPIC	<p>Dynamically allocate an IPIC port during provision. Valid values are:</p> <ul style="list-style-type: none"> • YES A TCPIPService is configured for the CICS region to listen for inbound IPIC requests. The default autoinstall program for IPCONN resources (DFHISAIP) is enabled so that outbound IPCONN connections can be automatically created when an inbound request is received. <p>If the property DFH_REGION_SEC=NO is specified, the IPIC port is unsecured. If the property DFH_REGION_SEC=YES is specified, the IPIC port is secured by using SSL with client authentication. The default autoinstall program DFHISAIP creates IPCONN resources that specify SSL(NO) as one of their parameters, meaning that outbound IPIC connections are not secured by SSL.</p> • NO No IPIC port is allocated. <p>If not set, no IPIC port is allocated.</p>
DFH_REGION_IPIC_USAGETYPE	<p>Optional property to set the usage type for an IPIC port so that the port is dynamically allocated from the port allocation range that this usage type identifies.</p> <p>A valid usage type name is 1-32 characters and the following valid characters: A-Z a-z 0-9</p> <p>Typically, a network administrator uses z/OSMF to configure a port allocation range and specify the usage type for that range.</p> <p>This property is used if DFH_REGION_IPIC is set to YES.</p>
DFH_REGION_TCPIP_USAGETYPE	<p>Optional property to set the usage type for the CICSplex SM WUI port so that the port is dynamically allocated from the port allocation range that this usage type identifies.</p> <p>A valid usage type name is 1-32 characters and the following valid characters: A-Z a-z 0-9</p> <p>Typically, a network administrator uses z/OSMF to configure a port allocation range and specify the usage type for that range.</p>

Security options

Property	Description
DFH_REGION_KEYRING	<p>Name of a key ring. Used for SSL or TLS connections to CICS. You must specify this property only when:</p> <ul style="list-style-type: none"> • DFH_REGION_SEC is set to YES, and DFH_CICS_TYPE is set to SMSS, meaning that a secure CMCI port is created. • DFH_REGION_JVMSEVER is set to None, OSGI or Axis2, and DFH_REGION_HTTPS is set to YES, meaning that an HTTPS port is created. <p>If no value is specified and the property is needed, the APPLID of the CICS region is used. The key ring can be dynamically created as part of the provisioning process. For more information, see “Modifying the CICS workflows to support dynamic security” on page 73.</p>

Property	Description
DFH_REGION_SEC	Specify whether security is enabled for the CICS region. Note: If the value is specified as NO, all other security options are ignored.
DFH_REGION_SECPREFIX	Specify whether to use SECPREFIX. If no value is specified, the APPLID of the CICS region is used. Valid values are: <ul style="list-style-type: none"> • NO • YES • <i>prefix</i>
DFH_REGION_XAPPC	Specify whether Resource Access Control Facility (RACF) session security can be used with APPC connections. Valid values are: <ul style="list-style-type: none"> • NO • YES
DFH_REGION_XCMD	Specify whether CICS command security is enabled. Valid values are: <ul style="list-style-type: none"> • NO • YES • <i>classname</i>
DFH_REGION_XDB2	Specify whether DB2ENTRY security checking is enabled. Valid values are: <ul style="list-style-type: none"> • NO • <i>classname</i>
DFH_REGION_XDCT	Specify whether transient data queue security is enabled. Valid values are: <ul style="list-style-type: none"> • NO • YES • <i>classname</i>
DFH_REGION_XFCT	Specify whether file security is enabled. Valid values are: <ul style="list-style-type: none"> • NO • YES • <i>classname</i>
DFH_REGION_XHFS	Specify whether zFS file security is enabled. Valid values are: <ul style="list-style-type: none"> • NO • YES • <i>classname</i>
DFH_REGION_XJCT	Specify whether journal security is enabled. Valid values are: <ul style="list-style-type: none"> • NO • YES • <i>classname</i>
DFH_REGION_XPCT	Specify whether started transaction security is enabled. Valid values are: <ul style="list-style-type: none"> • NO • YES • <i>classname</i>
DFH_REGION_XPPT	Specify whether program security is enabled. Valid values are: <ul style="list-style-type: none"> • NO • YES • <i>classname</i>

Property	Description
DFH_REGION_XPSB	Specify whether program specification block security is enabled. Valid values are: <ul style="list-style-type: none"> • NO • YES • <i>classname</i>
DFH_REGION_XPTKT	Specify whether PassTicket security is enabled. XPTKT is a new security option that requires a CICS APAR to be applied. Valid values are: <ul style="list-style-type: none"> • NO • YES
DFH_REGION_XRES	Specify whether CICS resource security is enabled. Valid values are: <ul style="list-style-type: none"> • NO • YES • <i>classname</i>
DFH_REGION_XTRAN	Specify whether transaction attach security is enabled. Valid values are: <ul style="list-style-type: none"> • NO • YES • <i>classname</i>
DFH_REGION_XTST	Specify whether temporary storage queue security is enabled. Valid values are: <ul style="list-style-type: none"> • NO • YES • <i>classname</i>
DFH_REGION_XUSER	Specify whether surrogate user checking is enabled. Valid values are: <ul style="list-style-type: none"> • NO • YES

Optional CICS properties

The following optional properties are used to insert SIT parameters and data sets into the provisioned region.

Property	Description
DFH_REGION_STEPLIB	Comma-separated list of variables to append to the standard CICS and Language Environment STEPLIB data sets.
DFH_REGION_STEPLIB_TOP	Comma-separated list of variables to add before the standard CICS and Language Environment STEPLIB data sets.
DFH_REGION_RPL	Comma-separated list of variables to append to the standard CICS and Language Environment DFHRPL data sets.
DFH_REGION_RPL_TOP	Comma-separated list of variables to add before the standard CICS and Language Environment DFHRPL data sets.

Property	Description
DFH_REGION_CSD_TYPE	<p>Specify whether a new CSD needs to be created for each CICS region, or whether an existing CSD can be shared. For more information about sharing a CSD, see “Shared CSD guidance” on page 162. The default is to create a new CSD, which is then deleted when the CICS region is deprovisioned.</p> <p>The provisioning process adds region-specific resources to the CSD. During deprovision those resources get removed from the CSD if it is shared. When a CSD is shared, the CSD must be accessed in a way that allows the successful provision and deprovision of other CICS regions.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • NEW • SHAREDREADONLY <p>Provision CICS regions that access a non record-level sharing (RLS) CSD in read-only mode. Provision and deprovision fail if region-specific resources cannot be added or removed because the CSD is in use.</p> <ul style="list-style-type: none"> • SHAREDRLS <p>Provision CICS regions that need to access the CSD in update mode.</p> <p>CSD-specific CICS system initialization parameters are automatically set based on the value of this property and cannot be overridden. For more information about CICS system initialization parameters, see Specifying more CICS system initialization parameters.</p>
DFH_REGION_CSD_NAME	Specify the name of an existing CSD to be shared by the provisioned CICS regions. This option applies only when DFH_REGION_CSD_TYPE is set to SHAREDREADONLY or SHAREDRLS.
DFH_REGION_CSD_INPUT	Comma-separated list of variables that allow data sets to be inserted into the CSD upgrade job, for example: DFH_REGION_CSD_INPUT=CICPROV.CSD(EQACSD13). This option applies only when DFH_REGION_CSD_TYPE is set to NEW.
DFH_REGION_SIT	The two-character name of the compiled SIT.
DFH_REGION_SITPARMS	<p>A provisioned CICS region is automatically configured with a set of system initialization (SIT) parameters. DFH_REGION_SITPARMS enables more SIT parameters to be specified for further customization of the region. The parameters must be comma-separated, for example: DFH_REGION_SITPARMS=DEBUGTOOL=YES,AUXTR=ON</p> <p>For more information about which SIT parameters are configured automatically and how, see Specifying more CICS system initialization parameters.</p>
DFH_REGION_DD_LIST	<p>Comma-separated list of DD cards to add to the CICS region JCL. Commas can be included in the DD card by prefixing them with two backslashes \\. Do not include // at the beginning of the line.</p> <p>For example: DFH_REGION_DD_LIST=EXMPCAT DD DSN=TEST.EXMPLAPP.EXMPCAT\\,DISP=SHR,EXMPCONF DD DSN=TEST.EXMPLAPP.EXMPCONF\\,DISP=SHR</p> <p>For example in an image: ENV {DFH_REGION_DD_LIST=EXMPCAT DD DSN=TEST.EXMPLAPP.EXMPCAT\\,DISP=SHR,EXMPCONF DD DSN=TEST.EXMPLAPP.EXMPCONF\\,DISP=SHR}</p>
DFH_CICS_LLQ	<p>CICS installation data set low-level qualifier, for example CICSLLQ is returned:</p> <p>CTS550.CICS720.SDFHAUTH.CICSLLQ CTS550.CICS720.SDFHLOAD.CICSLLQ CTS550.CICS720.SDFJAUTH.CICSLLQ</p>

Property	Description
DFH_CPSM_LLQ	CICSplex SM installation data set low-level qualifier, for example CPSMLLQ is returned: CTS550.CPSM550.SEYUAUTH.CPSMLLQ CTS550.CPSM550.SEYULOAD.CPSMLLQ

CICSplex SM properties

The following optional properties are used to configure CICS regions that participate in a CICSplex.

Property	Description	DFH_CICS_TYPE
DFH_CICSplex_DEF	Use this property to set the name of a CICSplex to be created during provision of a CMAS. If no value is specified, the CICSplex name is generated dynamically as part of the provisioning process.	CMAS
DFH_CICSplex	The name of the CICSplex that the MAS or WUI is associated with.	MAS or WUI
DFH_CMAS_APPLID	The VTAM APPLID of the CMAS that the MAS or WUI is attached to. This property is optional.	MAS or WUI
CMAS_REGISTRY_NAME	This property can be set when you provision a WUI or a MAS region. The provisioning workflow looks for the CMAS in the z/OSMF registry to get the information needed for the workflow to connect to the CMAS. This property is optional and can be specified as an alternative to DFH_CICSplex and optional properties DFH_CMAS_APPLID and DFH_CMAS_SYSID.	MAS or WUI
DFH_CMAS_SYSID	The SYSID of the CMAS that the MAS or WUI is attached to. This property is optional.	MAS or WUI
DFH_CICSGRP	The CICS system group that the MAS or WUI is attached to. The CICS system group must already be defined in the CICSplex. This property is optional.	MAS or WUI
DFH_REGION_EYUPARMS	Define EYUPARM settings for this region in a comma-separated list. This property is optional.	CMAS or MAS or WUI
DFH_REGION_WUIPARMS	Define WUI server initialization parameters in a comma-separated list. This property is optional.	WUI

Workflow configuration variables

Property	Description
CONSOLE_COMMAND_TYPE	<p>Sets how the provisioning process issues MVS commands and retrieves command responses for tasks such as starting and stopping a CICS region, checking the status of a CICS region, and deleting log streams. Valid values are as follows:</p> <ul style="list-style-type: none">• SCRIPT. Use REXX scripts to issue console commands. A console is opened for each provisioned CICS region, using the APPLID of the CICS region as the console name. For templates for a <code>cics_getting_started</code> workflow, SCRIPT is the only valid value. If you use this value with templates that provision a CICS region by using IBM Cloud Provisioning and Management for z/OS, it reduces the configuration required.• APICALL. Use z/OSMF z/OS console services (a REST API) to issue console commands and detect unsolicited messages. A single console manages all provisioned CICS regions. To use this value, you must configure z/OS console services and the relevant additional authorization for the userid that the <code>DFH_ADMIN_CONSOLE</code> property specifies. See “User IDs for the configuration” on page 149 and Chapter 9, “Preparing to use z/OS Provisioning Toolkit with IBM Cloud Provisioning and Management for z/OS,” on page 41. This value is not valid for templates for a <code>cics_getting_started</code> workflow. <p>If this value is not specified, templates for a <code>cics_getting_started</code> workflow use REXX scripts, and templates for workflows that use IBM Cloud Provisioning and Management for z/OS use z/OS console services.</p>

Specifying more CICS system initialization parameters

CICS regions are provisioned with system initialization (SIT) parameters that are configured automatically to support the properties that are defined in the template. Use the `DFH_REGION_SITPARMS` property for further customization of SIT parameters for CICS regions. Any SIT parameter can be specified by using `DFH_REGION_SITPARMS`, but any conflicting SIT parameters are overridden by the automatically configured SIT parameters (conflicts are commented out in the generated started task procedure for the CICS region). The following table lists SIT parameters that are automatically configured and whether they can be overridden. SIT parameters not in the table can all be specified on the `DFH_REGION_SITPARMS` property.

SIT parameter	How it is configured	Can be overridden
APPLID	Set to the dynamically obtained APPLID for the CICS region.	No
CICSSVC	Set to the value of the <code>DFH_REGION_CICSSVC</code> property.	No
CPSMCONN	Set to <code>LMAS</code> if the value of <code>DFH_CICS_TYPE</code> is set to <code>MAS</code> .	No
CSDACC	Set to <code>READONLY</code> or <code>READWRITE</code> , depending on the <code>DFH_REGION_CSD_TYPE</code> property.	No

SIT parameter	How it is configured	Can be overridden
CSDRLS	Set to YES if DFH_REGION_CSD_TYPE is set to SHAREDRLS, otherwise set to NO.	No
DFLTUSER	Set to the value of the DFH_REGION_DFLTUSER property.	No
GMTEXT	Set to a string that includes the dynamic APPLID of the provisioned region and the type of region that is being provisioned.	Yes
GMTRAN	Set to CESN if DFH_REGION_SEC is set to YES.	Yes
GRPLIST	Set to a list that contains the specific resource definitions for the provisioned CICS region.	Yes, but the provided GRPLIST must specify a maximum of three entries. These are concatenated onto a first entry that is generated during provisioning to add specific resource definitions for the provisioned CICS region. This generated list also includes the CICS groups typically in DFHLIST, so DFHLIST is not required as a separate entry in your GRPLIST.
IRCSTRT	Set to YES.	Yes
ISC	Set to YES.	Yes
JVMPROFILEDIR	Set to a dynamically provisioned zFS directory for the CICS region.	No
KEYRING	Set to the value of the DFH_REGION_KEYRING property, or the dynamically provisioned APPLID of the CICS region, if DFH_REGION_KEYRING is not specified. Set when DFH_REGION_SEC is set to YES.	No
RLS	Set to YES only if DFH_REGION_CSD_TYPE is set to SHAREDRLS.	Only when DFH_REGION_CSD_TYPE is not SHAREDRLS.
SEC	Set to the value of the DFH_REGION_SEC property.	No
SECPRFX	Set to the value of the DFH_REGION_SECPRFX property, or the dynamically provisioned APPLID of the CICS region if the DFH_REGION_SECPRFX property is not set.	No
SIT	Set to the value of the DFH_REGION_SIT property.	No
START	Set to AUTO.	Yes
SYSIDNT	Set to a dynamic system ID based on the dynamic APPLID for the CICS region.	No
TCPIP	Always set to YES to enable the CICS region to open TCPIP SERVICES for web or IPIC traffic.	No
TRANISO	Set to YES.	Yes
USSHOME	Set to the value of the DFH_CICS_USSHOME property.	No
USSCONFIG	Set to the value of the DFH_CICS_USSCONFIG property.	No
XAPPC	Set to the value of the DFH_REGION_XAPPC property.	No

SIT parameter	How it is configured	Can be overridden
XCMD	Set to the value of the DFH_REGION_XCMD property.	No
XDB2	Set to the value of the DFH_REGION_XDB2 property.	No
XDCT	Set to the value of the DFH_REGION_XDCT property.	No
XFCT	Set to the value of the DFH_REGION_XFCT property.	No
XHFS	Set to the value of the DFH_REGION_XHFS property.	No
XJCT	Set to the value of the DFH_REGION_XJCT property.	No
XPCT	Set to the value of the DFH_REGION_XPCT property.	No
XPPT	Set to the value of the DFH_REGION_XPPT property.	No
XPSB	Set to the value of the DFH_REGION_XPSB property.	No
XPTKT	Set to the value of the DFH_REGION_XPTKT property.	No
XRES	Set to the value of the DFH_REGION_XRES property.	No
XTRAN	Set to the value of the DFH_REGION_XTRAN property.	No
XTST	Set to the value of the DFH_REGION_XTST property.	No
XUSER	Set to the value of the DFH_REGION_XUSER property.	No

Shared CSD guidance

By default, when a CICS region is provisioned, a new CSD is defined for that CICS region. When the CICS region is deprovisioned, the CSD is deleted. This ensures that each CICS region has a unique CSD, and that any changes that are made to that CSD do not affect other CICS regions.

You can change the value of the DFH_REGION_CSD_TYPE parameter to provision CICS regions that share an existing CSD. Using a shared CSD ensures that updates to the CSD are available to all CICS regions that share the CSD. If a CSD is shared, it must not be held open by a CICS region. A CSD that remains open can cause provision and deprovision failures.

The shared CSD can be used for regions that are provisioned and deprovisioned by using z/OSMF to avoid situations where a CSD held open in a CICS region can cause provisioning and deprovisioning failures. One approach is to clone an existing CSD, then use that clone as the shared CSD for provisioning.

Accessing the CSD by using record-level sharing (RLS) provides the highest level of sharing. If the CICS regions you provision need to update the CSD, specify DFH_REGION_CSD_TYPE=SHAREDRLS. For more information about considerations for creating a CSD, see Setting up shared data sets, CSD and SYSIN in CICS TS product documentation.

If the CICS regions you provision do not need to update the CSD, a non-RLS CSD can be shared. The CICS regions open the CSD in read-only mode, which avoids problems with the CSD being locked when DFHCSDUP jobs are run during the provisioning and deprovisioning workflows. This option is specified by the property DFH_REGION_CSD_TYPE=SHAREDREADONLY.

If you need to update a non-RLS shared CSD, ensure that the updates do not cause provisioning and deprovisioning failures. DFHCSDUP jobs that the provisioning workflows submit include an extra DD statement that opens a data set exclusively. This data set is used as a serialization mechanism to ensure that the DFHCSDUP

jobs do not cause each other to fail because the CSD is in use when multiple provisioning or deprovisioning processes are running concurrently. You can specify the same DD card in your own DFHCSDUP jobs to serialize your updates with updates that are made during the provisioning and deprovisioning processes. The DD card is as follows, where the values of DFH_REGION_HLQ and DFH_ZOS_VSAM_UNIT are properties that are described in tables “CICS region configuration ” on page 150 and “z/OS configuration options” on page 152.

```
//CSDLOCK DD DSN=<DFH_REGION_HLQ>.CSDLOCK,  
// DISP=(MOD,CATLG),  
// UNIT=<DFH_ZOS_VSAM_UNIT>,SPACE=(1,1)
```

Chapter 23. Provisioning an instance of z/OS Connect Enterprise Edition through IBM Cloud Provisioning and Management for z/OS

You can use the templates that are supplied with z/OS PT enable you to provision an instance of z/OS Connect EE through IBM Cloud Provisioning and Management for z/OS but without running z/OS PT.

Before you begin

Before you can provision an instance of z/OS Connect EE by using a template that is supplied with z/OS PT, complete the following checks:

- Ensure that z/OSMF is installed and configured. See Configuration in the z/OSMF Configuration Guide.

For summary information about z/OSMF, see Chapter 3, “Prerequisites for using z/OS Provisioning Toolkit,” on page 17.

- Ensure that your user ID has sufficient authority to create an environment, load templates into z/OSMF, and run workflows. See Chapter 7, “Security planning for z/OS Provisioning Toolkit with IBM Cloud Provisioning and Management for z/OS,” on page 25 and “Security requirements for provisioning instances of z/OS Connect EE with z/OS Provisioning Toolkit” on page 103.

For more information about setting up security for IBM Cloud Provisioning and Management for z/OS, see Help with security setup in IBM Z solutions product documentation.

- Ensure that IBM Cloud Provisioning and Management for z/OS is installed and configured. See Preparing to use Cloud Provisioning in the z/OSMF Configuration Guide.
-

About this task

z/OS PT is packaged as a compressed file that you can download without charge. You can access the templates that are supplied with z/OS PT without fully installing it.

For information about the templates, see Chapter 20, “z/OS Provisioning Toolkit supplied templates,” on page 139. For a list of templates that are supplied with z/OS PT, see the *Sample templates* section of Chapter 21, “Samples provided with z/OS Provisioning Toolkit,” on page 141.

You can customize the templates to local naming standards and operating procedures by using the configuration properties that are supplied with the templates. You can manage and control the templates and the workflows that they drive by using IBM Cloud Provisioning and Management for z/OS.

Procedure

1. Find z/OS PT for download on IBM z/OS Provisioning Toolkit product page.
2. Extract the .zip file.
3. Transfer the .pax file in binary format to the z/OS location where you want to store your templates.

4. Issue the following command to extract the .pax file:

```
pax -rf zospt_v#####.pax
```

The zospt installation directory is created.

5. Identify the template that you require in the zospt/templates directory, for example zospt/templates/zosconnect_v3r0. For a description of each template, see the *Sample templates* section of Chapter 21, “Samples provided with z/OS Provisioning Toolkit,” on page 141.

To create a new template, copy the entire template directory to another directory in the zospt/templates directory.

6. Edit the zosconnect.properties file in the directory for your chosen template to configure the instance of z/OS Connect EE for the release you require and to meet your naming standards. See “Configuration properties for z/OS Connect Enterprise Edition templates.”
7. Add the template to z/OSMF through the z/OSMF web interface. For details, see “Adding a z/OS Connect Enterprise Edition template by using IBM Cloud Provisioning and Management for z/OS” on page 100, and for example steps with screenshots, see “Getting started with CICS by using IBM Cloud Provisioning and Management for z/OS” on page 34.
8. Optional: You can use z/OS PT to run and evaluate your template before you publish it for wider use. To do this, complete all steps in Chapter 4, “Installing z/OS Provisioning Toolkit,” on page 19 and see “Adding a z/OS Connect Enterprise Edition template by using IBM Cloud Provisioning and Management for z/OS” on page 100.

What to do next

After you add the template to z/OSMF and it is ready to publish, you can use a suitable method to run the template.

Configuration properties for z/OS Connect Enterprise Edition templates

You can configure a number of properties to customize the way that instances of z/OS Connect EE are provisioned through IBM Cloud Provisioning and Management for z/OS. Properties for a template are specified in a zosconnect.properties ISO8859-1 file, ready to load into z/OSMF.

If you provision an instance of z/OS Connect EE by using z/OS PT, see “Configuration properties for z/OS Connect Enterprise Edition images” on page 106, because more configuration options might apply.

High-level qualifiers for your installation

Property	Description	Where set
ZCON_INSTALL_DIR	The path to the z/OS Connect EE installation directory on zFS.	Template

User IDs for the configuration

For more information about the steps for which each user ID is used, see “Security requirements for provisioning instances of z/OS Connect EE with z/OS Provisioning Toolkit” on page 103.

Property	Description	Where set
ZCON_ADMIN_CONSOLE	<p>User ID that is used for CONSOLE API commands and needs to have special OPERPARM configuration.</p> <p>When the CONSOLE API is used, an MVS console is created and used to submit console commands. The MVS console name is the first six characters of the ZCON_ADMIN_CONSOLE user ID followed by the letters CN. If multiple z/OSMF servers are installed on different LPARs in a sysplex, the generated MVS console name (the first six characters of the DFH_ADMIN_CONSOLE user ID) must be unique on each LPAR.</p> <p>For more information about authorizations and configuration of the z/OS console services, see z/OS console services in the z/OSMF Programming Guide.</p> <p>Also ensure that message CWWKF0011I is not suppressed by the Message Processing Facility (MPF), because this message is used to detect when z/OS Connect EE starts.</p>	Template
ZCON_ADMIN_SECURITY	<p>User ID under which dynamic SAF security updates are made during the provisioning process.</p> <p>By default, no dynamic security updates are made, because the REXX scripts that are run exit immediately with return code zero. This default behavior allows for predefined security profiles to be used. The REXX scripts can be modified to issue commands to dynamically create, update, and delete the security profiles that are needed for the provisioned instances of z/OS Connect EE. For more information, see “Modifying the z/OS Connect Enterprise Edition workflows to support dynamic security” on page 105.</p>	Template
ZCON_ADMIN_SERVER	<p>User ID under which the provisioned instance of z/OS Connect EE runs. This user ID owns the data directory on zFS of the instance of z/OS Connect EE.</p> <p>z/OS Connect EE runs as a started task and its job name matches the instance name of the provisioned z/OS Connect EE instance in z/OSMF. A SAF security profile must exist that ensures the started task runs under the ZCON_ADMIN_SERVER user ID. This profile can be predefined, or created dynamically as part of the provisioning process. For more information, see “Modifying the z/OS Connect Enterprise Edition workflows to support dynamic security” on page 105.</p>	Template
ZCON_ADMIN_TSO	User ID that can be used in the step to create the procedure in the PROCLIB named by the ZCON_ZOS_PROCLIB variable for the instance of z/OS Connect EE.	Template
ZCON_ADMIN_ZFS	User ID that has zFS authority to mount and unmount zFS file systems that are used by the z/OS Connect EE instance.	Template
ZCON_APPROVER_CONSOLE	Approval user ID of the ZCON_ADMIN_CONSOLE user ID for issuing MVS console commands.	Template
ZCON_APPROVER_SECURITY	Approval user ID of the ZCON_ADMIN_SECURITY user ID for making dynamic SAF security changes.	Template
ZCON_APPROVER_SERVER	Approval user ID of the ZCON_ADMIN_SERVER user ID for running the instance of z/OS Connect EE.	Template

Property	Description	Where set
ZCON_APPROVER_TSO	Approval user ID of the ZCON_ADMIN_TSO user ID for creating the procedure that is used to start the instance of z/OS Connect EE.	Template
ZCON_APPROVER_ZFS	Approval user ID of the ZCON_ADMIN_ZFS user ID for creating and mounting directories in zFS.	Template

z/OS Connect EE instance configuration

Property	Description	Where set
ZCON_FILE_SYSTEM_HLQ	Specifies the high-level qualifier for the zFS data sets that are defined during the provisioning of the z/OS Connect EE instance.	Template

z/OS configuration options

Property	Description	Where set
ZCON_ZOS_PROCLIB	<p>Names a PROCLIB data set into which the workflow can create a procedure for the z/OS Connect EE instance. The z/OS Connect EE instance runs as a started task. If you specify a PROCLIB that is a PDSE, it enables concurrent addition of new procedures to the PROCLIB. If a PDS is used and multiple updates to the PROCLIB are attempted concurrently, z/OS abends them with a 213-30 abend code, to avoid data corruption.</p> <p>When a procedure is created in the PROCLIB, its name will match the instance name of the z/OS Connect EE instance that is provisioned. Therefore, use a suitable naming convention for the instances z/OS Connect EE to ensure that the procedure names are unique and do not clash with any existing procedure names. If the procedure name is not unique, another started task with the same name might be stopped during deprovisioning of the z/OS Connect EE instance. Ensure that the named PROCLIB data set is included in the PROCLIB concatenation on your system so that it can be found when the command to start the z/OS Connect EE instance is issued.</p>	Template
ZCON_ZOS_STCJOBS	Names the data set into which the workflow can create the source JCL that defines the job level characteristics of the started task for the provisioned instance of z/OS Connect EE. Used only when ZCON_STC_JOB_CARD is also specified. If you specify a data set of type PDSE, it enables concurrent addition of new STCJOBS to the data set. If a PDS is used and multiple concurrent updates to the data set are attempted, z/OS produces a 213-30 abend code, to avoid data corruption.	Template
ZCON_STC_JOB_CARD	Specifies the job card to use for the started task of the provisioned instance of z/OS Connect EE. When specified, ZCON_ZOS_STCJOBS must also be used to indicate the data set prefix for STCJOBS.	Template
ZCON_ZOS_VSAM_VOLUME	VOLUME value that is used in the creation of the zFS data sets, for example, SYSDA. If the value is specified as SMS (Storage Management Subsystem), the VOLUME parameter is omitted from the VSAM creation.	Template

zFS configuration options

Property	Description	Where set
ZCON_ZFS_DATACLASS	<p>Specifies a data class that is used to define a zFS data set for the z/OS Connect EE instance. A data set is defined, formatted for zFS, and mounted into a directory created in the ZCON_ZFS_MOUNTPOINT location.</p> <p>DATACLASS (a list of data set allocation attributes and their values) defined on z/OS that is used to create the zFS data sets. For more information about setup details, see “Configuration prerequisites for using z/OS Provisioning Toolkit with z/OS Connect Enterprise Edition” on page 99.</p> <p>For non-SMS-managed systems, either remove the ZCON_ZFS_DATACLASS property, or do not give the property a value. Specify a value for ZCON_ZOS_VSAM_VOLUME instead.</p>	Template
ZCON_ZFS_MOUNTPOINT	Specifies a zFS directory into which a data directory for the provisioned instance of z/OS Connect EE can be defined.	Template

JVM configuration options

Property	Description	Where set
JAVA_HOME	Location of the Java installation	Template

Security options

Property	Description	Where set
ZCON_SAF_AUTHENTICATION	<p>Specify whether to use SAF authentication. Valid values are:</p> <ul style="list-style-type: none">• FALSE. If you specify FALSE, you must manually configure an alternative authentication mechanism.• TRUE	Template

Property	Description	Where set
ZCON_TLS	<p>Specify whether to use transport layer security. Valid values are:</p> <ul style="list-style-type: none"> • JKS. An HTTPS port is needed for the instance of z/OS Connect EE and a Java keystore is used for transport layer security. The Java keystore is created dynamically by z/OS Connect EE during startup. • NO. The instance of z/OS Connect EE accepts requests over an unsecured HTTP port. • SAF. An HTTPS port is needed for the instance of z/OS Connect EE and a SAF key ring is used for transport layer security. The value of the ZCON_SERVER_KEYRING parameter determines whether a predefined key ring is used. <p>If ZCON_SERVER_KEYRING is specified, it is assumed to be the name of a predefined key ring.</p> <p>If ZCON_SERVER_KEYRING is not specified, the key ring is created dynamically by running the defineKeyring.rexx script that is found in the workflows/extensions folder. To enable the key ring to be created dynamically, you must modify defineKeyring.rexx (the default is to exit immediately with return code zero and not to make any dynamic profile updates). The corresponding deleteKeyring.rexx script must also be modified for your environment.</p>	Template
ZCON_SERVER_KEYRING	Names an existing SAF key ring to use when ZCON_TLS=SAF.	Template
ZCON_ZFS_GROUP	Specifies the name of a SAF group, which is granted read, write and run authority to the z/OS Connect EE instances data directory. Users of the group can add APIs and services into the file system of the z/OS Connect EE instance.	Template

Notices

This information was developed for products and services offered in the U.S.A. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

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 rights 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 give 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 (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*

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 jurisdictions 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.

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 want 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 Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
United States of America*

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 Programming License Agreement, or any equivalent agreement between us.

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.

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 actual people or business enterprises 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.

Trademarks

IBM, the IBM logo, and [ibm.com](http://www.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 at www.ibm.com/legal/copytrade.shtml.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

UNIX is a registered trademark of The Open Group in the United States and other countries.

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.



Printed in USA