IBM Wazi as a Service

Tables of Contents

IBM Wazi as a Service 1

Overview 1

Release notes 2

Use cases 4
- Use case 1: Standard virtualized z/OS with IBM Wazi for VS Code 6
- Use case 2: Standard virtualized z/OS with IBM Wazi for Eclipse 7
- Use case 3: Standard virtualized z/OS with web-based IBM Wazi for Dev Spaces 8
- Use case 4: Customized and virtualized z/OS with IBM Wazi for VS Code 9
- Use case 5: Customized and virtualized z/OS with IBM Wazi for Eclipse 11
- Use case 6: Customized and virtualized z/OS with web-based IBM Wazi for Dev Spaces 12

Bringing your own image with Wazi Image Builder 13
- Architecture overview 14
- Checklist 15
- Prerequisites 15
- Downloading installation packages 17
- Validating the signed installers 17
- Installing Wazi Image Builder 17
  - Installing the web server 18
  - Installing a signed certificate on the web server 19
  - Setting up a storage server 19
  - Installing ZD&T z/OS Extraction Utilities 20
  - Authentication for the Wazi Image Builder application 21
    - Default authentication and authorization 21
    - Basic authentication 21
    - Lightweight Directory Access Protocol (LDAP) 22
  - Starting and stopping the web server 22
- Configuring settings on the web server UI 23
  - Connecting to your storage server 23
  - Managing users and roles 23
  - Adding source environments 24
    - Adding a source environment from a z/OS system on IBM Z physical hardware 24
    - Adding a source environment from an existing z/OS instance run by ZD&T 25
  - Adding Wazi as a Service target environments 25
- Upgrading the web server 26
- Setting up networking for on-premises z/OS environment 26
- Creating components 29
  - Creating a volume component 29
  - Creating a volume component from an existing z/OS instance that ZD&T runs on Linux 30
- Monitoring and managing the created components 31
- Creating z/OS images 33
- Monitoring and managing the created images 34
- Deploying images to a Wazi as a Service target environment 35
- Monitoring and managing the deployments 36
- Using REST APIs 36
  - Creating and managing Wazi as a Service target environments with REST APIs 37
  - Creating components with REST APIs 38
    - Creating volume components from z/OS system with REST APIs 38
    - Creating volume components from existing ZD&T or instances 39
  - Checking the status of the created components with REST APIs 40
  - Editing created components with REST APIs 40
  - Creating images from z/OS volumes with REST APIs 41
  - Checking the status of the created images with REST APIs 41
  - Editing the created images with REST APIs 42
  - Deploying images to a Wazi as a Service target environment with REST APIs 42
  - REST API error codes 43
- Logs 54
  - Auditing the web server 54

Finding logs 54
Uninstalling the web server 55
Sharing feedback 55

Creating z/OS VSIs in IBM Cloud VPC 55
Change log for z/OS stock images 55
Configurations in z/OS stock images 59
Viewing products and services in z/OS stock images 62
IBM Z Hypervisor as a Service (zHYPaaS) overview 63
Setting up your IBM Cloud account 64
Getting started with IBM Virtual Private Cloud (VPC) 64
Granting user permissions for VPC resources 65
Available regions 68
Managing custom images 68
Creating dedicated hosts and groups 68
Using the IBM Cloud console to create z/OS virtual server instances 69
Creating your SSH key 69
Creating a VPC and subnet 70
Configuring the ACL 71
Creating a z/OS virtual server instance 71
Creating and attaching a block storage volume 73
Configuring the security group for the instance 73
Creating a client-to-site VPN server for VPC 73
Connecting to z/OS virtual server instances 74
Operating with the serial console 77
Managing z/OS virtual server instances 79
Monitoring z/OS virtual server instances 81
Using IBM Cloud DNS services with VPC 81
Installing software products for z/OS virtual server instances 81
Installing APARs or PTFs for z/OS virtual server instances 84
Configuring additional disk volumes for z/OS virtual server instances 85
Changing active I/O software configuration for z/OS virtual server instances 87
Initiating a stand-alone dump with z/OS virtual server instances 88
Creating snapshots of boot and data volumes 88
Regenerating RACF certificates for z/OS virtual server instances 89
Establishing service to service authorizations 89
Applying backup policies to resources using tags 90
Planning your file shares 91
Creating file shares and mount targets 91
Mounting file shares on z/OS 92
Shutting down z/OS virtual server instances 94
Security best practices for z/OS virtual server instances 95
Troubleshooting z/OS virtual server instances 96

Getting started with IBM Wazi as a Service 98
Prerequisites 98
Setting up an OpenVPN server to control access to the VSI 100
Configuring an OpenVPN client to enable access to the VSI 101
Configuring access to services 101
Developing and testing with IBM Wazi for VS Code 102
Connecting IBM Wazi for VS Code 102
Setting up Ansible for IBM Wazi 102
Configuring your development workspace with Ansible 104
Configuring your development workspace manually and shell scripts 106
Running Dependency-Based User Builds from Z Open Editor 107
Running a Debug session from Z Open Editor 108
Developing and testing with IBM Wazi for Eclipse or IBM Developer for z/OS 109
Connecting IBM Wazi for Eclipse or IBM Developer for z/OS 109
Running Dependency-Based User Builds from IBM Wazi for Eclipse or IBM Developer for z/OS 110
Running a Debug session from IBM Wazi for Eclipse or IBM Developer for z/OS 111
Deploying and using IBM Wazi for Dev Spaces from a Red Hat OpenShift cluster in your VPC 111
Creating a Red Hat OpenShift cluster 112
Deploying and configuring IBM Wazi for Dev Spaces 112
Creating a development workspace 112

Known limitations 113
Known problems 115
Getting support 118
Glossary 118

IBM Z® Wazi as a Service

This page provides an entry point to product information about IBM Wazi as a Service. Use the links in the navigation tree on the left to find documentation for specific topics. On the right, you can find links to various helpful resources. That list is updated frequently. Below, you can find entry points to sites that help you learn more about this product and other IBM® products. It provides links to sources for support and troubleshooting information.

Getting Started

- Product overview
- Bringing your own image with Wazi Image Builder
- Creating a dev&test z/OS VSI in IBM Cloud VPC
- Developing and testing with a dev&test z/OS VSI in IBM Cloud VPC

Learn more

- Product page
- IBM Cloud VPC

Contacts

- Product manager

Find support

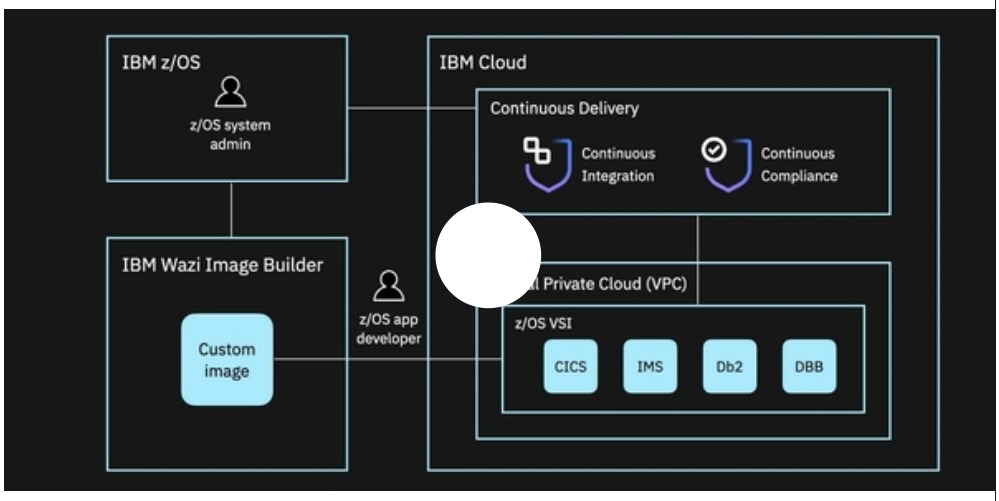
- IBM Support
- IBM Software Support home page

Overview of IBM Wazi as a Service

IBM® Wazi as a Service is a solution that enables a cloud native development and test experience for z/OS in IBM Cloud Virtual Private Cloud (VPC). With Wazi as a Service, application developers can develop and test z/OS application components in a virtual infrastructure "as a Service" environment running on IBM Z architecture on IBM Public Cloud. It gives you cloud security, with the ability to scale dynamically by providing on-demand access to z/OS resources where and when you need them, integrated with public cloud services and configured for integration with your existing DevOps tools.

IBM Wazi as a Service is now generally available in Japan (Tokyo), Brazil (São Paulo), Canada (Toronto), United Kingdom (London), US East (Washington DC), US South (Dallas), and Spain (Madrid) regions in IBM Cloud VPC.

IBM Wazi as a Service Overview



0:00 / 3:55 1x

Check out the video playlist on IBM MediaCenter to learn more.

Key capabilities

IBM Wazi as a Service provides the following key capabilities:

- z/OS development and test stock image: You can use the default z/OS stock image to creates and configure a z/OS virtual server instance for development and test in IBM Cloud VPC from the IBM Cloud console.
- IBM Wazi Image Builder: You can create and manage custom images from an existing Z platform by using Wazi Image Builder. It includes a web UI with role-based access and REST APIs to streamline the creation process. You can then deploy the custom images to IBM Cloud VPC to create a virtual server instance.

Then, you can use your industry standard integrated development environment (IDE) to build, test, and debug your applications.

For more information about using z/OS virtual server instances for development and test, see Use cases.

Benefits

IBM Wazi as a Service brings z/OS environments to IBM Cloud and can be integrated into existing development practices. It facilitates the transition to the latest development practices and frameworks like SAFe, Agile, Lean, and DevOps.

This solution provides the following benefits:

- Bridges the gap between the developer experience on distributed and mainframe platform during development of hybrid applications that contain z/OS application components.
- Delivers cloud native development experience for z/OS that is optimized to run on IBM Cloud.
- Allows developers to develop and test z/OS applications by using an IBM Cloud environment.
- Helps developers unfamiliar with z/OS and IBM Z get up to speed quickly and become productive in developing and delivering applications.
- Supports edit, build, and debug through integration with industry standard IDEs such as VS Code or Eclipse .
- Helps developers rapidly assess the impact of the changes that they want to make as often as needed.
- Rebalances entitlement over time flexibly based on your business needs.

Release notes

Stay up to date with the new features and enhancements of IBM® Wazi as a Service. Also, see Known Problems for existing problems and their resolutions.

24 April 2024

The z/OS development and test stock image in the Cloud UI is now available based on z/OS 3.1.0. For the latest updates on the z/OS development and test stock image, see Change log for z/OS stock images.

15 March 2024

IBM Wazi Image Builder 1.4.2 Fix Pack is available. To download and install the fix pack, go to IBM Passport Advantage or IBM Fix Central. To update to Wazi Image Builder 1.4.2, see Upgrading the web server.

Before creating your custom image, do the following:

- Verify the steps in the z/OS system requirements section under prerequisites.
- Rerun the networking setup for your on-premises z/OS environment and validate the configuration. For more information, see Setting up networking for on-premises z/OS environment.

For the latest updates on the z/OS stock image, see Change log for z/OS stock images.

15 December 2023

IBM Wazi Image Builder 1.4.1 Fix Pack is available. To download the fix pack, go to IBM Fix Central and select **`wib-install-1.4.1-fixpack`**.

You can now create dedicated hosts in the Spain (Madrid) and US South (Dallas) regions to carve out a single-tenant compute node and create virtual server instances according to your needs. Only users within your account that have the required permissions can create instances on the host. For more information, see Creating dedicated hosts and groups.

For the latest updates of z/OS dev and test stock images, see Change log for z/OS stock images.

10 November 2023

IBM Wazi as a Service is now available in the US South (Dallas) region in IBM Cloud Virtual Private Cloud (VPC).

For the latest updates of z/OS dev and test stock images, see Change log for z/OS stock images.

13 October 2023

IBM Wazi Image Builder 1.4 is available for download at IBM Passport Advantage. You can follow the instructions in Upgrading the web server to update to Wazi Image Builder 1.4.

Fixed APARs:

APAR	Description
PH55992	LPARNAME (XX) filter used in IEASYMXX members will not resolve correctly to the specified value of SYSTEM_NAME.

22 September 2023

IBM Wazi as a Service is now available in the Spain (Madrid) region in IBM Cloud Virtual Private Cloud (VPC).

For the latest updates of z/OS dev and test stock images, see Change log for z/OS stock images.

16 June 2023

IBM Wazi Image Builder 1.3 is available for download at IBM Passport Advantage. You can follow the instructions in Upgrading the web server to update to Wazi Image Builder 1.3.

Now when you create a new volume component that contains system residence files or edit such an existing volume component, you can enable automated volume discovery to identify and extract a minimum set of disk volumes that are needed to get your z/OS up and running. For more information, see Creating a volume component.

You can now specify the ZD&T z/OS Extraction Utilities installation directory on the web server, instead of using the **`/usr/lpp/IBM/zdt`** default path. For more information, see Installing z/OS Extraction Utilities.

You are welcome to share your feedback on Wazi Image Builder by participating in surveys or by granting us permission to collect your anonymous data. For more information, see Sharing feedback.

The z/OS dev and test stock image in the Cloud UI is now based on z/OS 2.5. For the latest updates of the z/OS stock images, see Change log for z/OS stock images.

11 May 2023

IBM Wazi Image Builder 1.2.2 Fix Pack, with updated versions of Java and Liberty, is available. To download the fix pack, go to IBM Fix Central and select **`wib-1.2.2-fixpack`**.

To address Java and Liberty vulnerabilities, the installer has been upgraded from version 1.2.1 to 1.2.2. However, as the web application remains unchanged, the version numbers displayed in the web UI and log files still show as 1.2.1. If you have downloaded the installer for version 1.2.1, update to version 1.2.2 to ensure that the Java and Liberty vulnerabilities are addressed.

10 March 2023

You can now download and install IBM Wazi Image Builder 1.2 from IBM Passport Advantage, and follow the instructions in Upgrading the web server to update to Wazi Image Builder 1.2.

You can use the **`zOS-cloud-prepare`** tool to set up the networking for your on-premises z/OS environment.

When you add a target environment, you can specify a VPC region to store your custom image. After you deploy a custom image, it is automatically converted for use in the region that you selected for the target environment.

You can go to the **Deployments** page on the Wazi Image Builder UI to monitor the deployments.

You can also use REST APIs to create and deploy custom images with Wazi Image Builder. For more information, see Using REST APIS.

For the latest updates of z/OS dev and test stock images, see Change log for z/OS stock images.

16 December 2022

IBM Wazi Image Builder 1.1.1 Fix Pack is available. To download the fix pack, go to IBM Fix Central and select **`wib-1.1.1-fixpack`**.

For the latest updates of z/OS dev and test stock images, see Change log for z/OS stock images.

23 September 2022

IBM Wazi as a Service is now generally available in Japan (Tokyo), Brazil (São Paulo), Canada (Toronto), United Kingdom (London), and US East (Washington DC) regions in IBM Cloud Virtual Private Cloud (VPC).

You can now download and install IBM Wazi Image Builder 1.1 from IBM Passport Advantage. To update from Wazi Image Builder 1.0 to 1.1, follow the instructions in the Update checklist.

For the latest updates of z/OS dev and test stock images, see Change log for z/OS stock images.

18 July 2022

IBM Wazi as a Service is now available in the US East (Washington DC) region in IBM Cloud Virtual Private Cloud (VPC).

30 June 2022

IBM Wazi as a Service is offered as an allow-listed service and is available in Japan (Tokyo), Brazil (São Paulo), Canada (Toronto), and United Kingdom (London) regions in IBM Cloud Virtual Private Cloud (VPC).

Use cases

IBM® Wazi as a Service is a cloud-ready z/OS environment that fully integrates into any enterprise-wide standard DevOps pipeline. You can mix and match different tools and services as you need with multiple use cases.

Depending on the use case that you select, the connection, configuration, and usage tasks vary. To get started rapidly, select one of the following use cases and complete the required tasks.

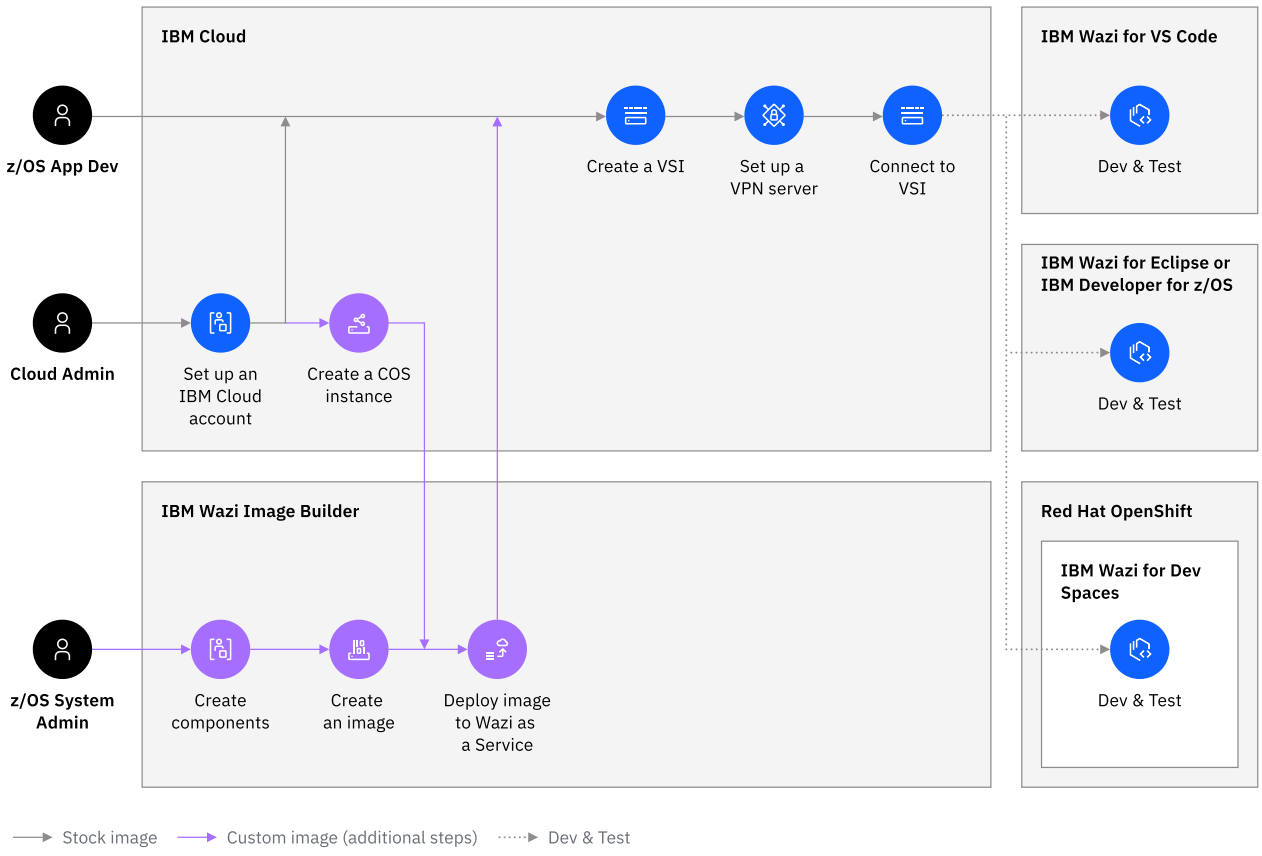


Figure 1. Wazi as a Service journey map

Use case 1: Standard virtualized z/OS with IBM Wazi for VS Code

- **Create a z/OS VSI in IBM Cloud VPC:** In IBM Cloud, create a z/OS VSI with a standard z/OS dev & test stock image to initiate your z/OS environment.
- **Develop and test with IBM Wazi for VS Code:** On your desktop, set up the VS Code extensions of IBM Wazi for VS Code, and connect to the z/OS VSI to debug, code, and build your applications.

To adopt this use case, follow the required tasks.

Use case 2: Standard virtualized z/OS with IBM Wazi for Eclipse

- **Create a z/OS VSI in IBM Cloud VPC:** In IBM Cloud, create a z/OS VSI with standard z/OS dev & test stock image to initiate your z/OS environment.
- **Develop and test with IBM Wazi for Eclipse:** On your desktop, set up IBM Wazi for Eclipse, and connect to the sandbox instance to debug, code, and build your applications.

To adopt this use case, follow the required tasks.

Use case 3: Standard virtualized z/OS with web-based IBM Wazi

- **Create a z/OS VSI in IBM Cloud VPC:** In IBM Cloud, create a z/OS VSI with standard z/OS dev & test stock image to initiate your z/OS environment.
- **Develop and test with IBM Wazi for Dev Spaces:** In Red Hat OpenShift, create a development workspace and connect to the z/OS VSI to debug, code, and build your applications.

To adopt this use case, follow the required tasks.

Use case 4: Customized and virtualized z/OS with IBM Wazi for VS Code

- **Create a custom z/OS image with IBM Wazi Image Builder:** Bring your own z/OS image with customized z/OS components and data sets, and deploy the image to IBM Cloud.
- **Create a z/OS VSI in IBM Cloud VPC:** In IBM Cloud, create a z/OS VSI with deployed z/OS custom image to initiate your z/OS environment.
- **Develop and test with IBM Wazi for VS Code:** On your desktop, set up the VS Code extensions of IBM Wazi for VS Code, and connect to the z/OS VSI to debug, code, and build your applications.

To adopt this use case, follow the required tasks.

Use case 5: Customized and virtualized z/OS with IBM Wazi for Eclipse

- **Create a custom z/OS image with IBM Wazi Image Builder:** Bring your own z/OS image with customized z/OS components and data sets, and deploy the image to IBM Cloud.
- **Create a z/OS VSI in IBM Cloud VPC:** In IBM Cloud, create a z/OS VSI with deployed z/OS custom image to initiate your z/OS environment.
- **Develop and test with IBM Wazi for Eclipse:** On your desktop, set up IBM Wazi for Eclipse, and connect to the sandbox instance to debug, code, and build your applications.

To adopt this use case, follow the required tasks.

Use case 6: Customized and virtualized z/OS with web-based IBM Wazi

- **Create a custom z/OS image with IBM Wazi Image Builder:** Bring your own z/OS image with customized z/OS components and data sets, and deploy the image to IBM Cloud.
- **Create a z/OS VSI in IBM Cloud VPC:** In IBM Cloud, create a z/OS VSI with deployed z/OS custom image to initiate your z/OS environment.
- **Develop and test with IBM Wazi for Dev Spaces:** In Red Hat OpenShift, create a development workspace and connect to the z/OS VSI to debug, code, and build your applications.

To adopt this use case, follow the required tasks.

Use case 1: Standard virtualized z/OS with IBM Wazi for VS Code

If you choose to use standard z/OS® VSI from IBM® Cloud VPC and IBM Wazi for VS Code , follow this role-based roadmap to set up and use IBM Wazi as a Service.

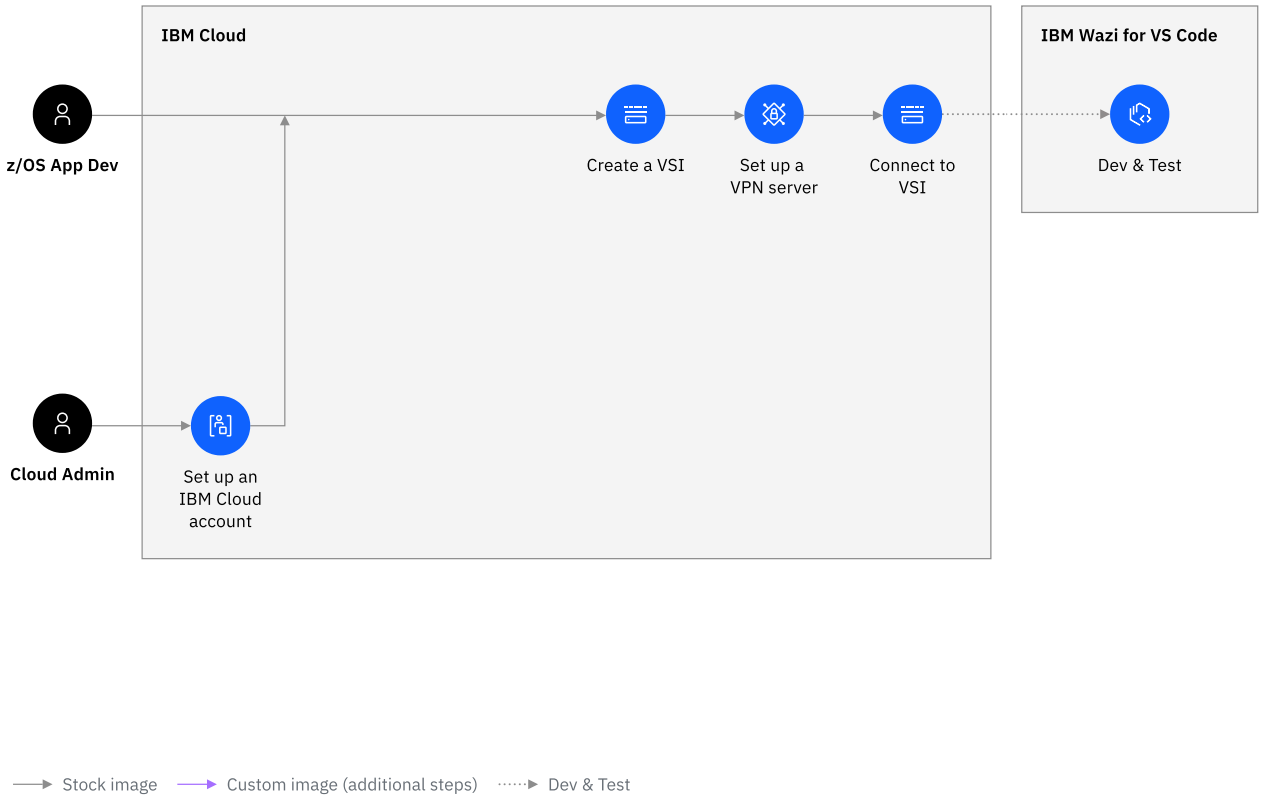


Figure 1. Wazi as a Service use case 1

	Cloud Admin	z/OS System Admin	z/OS Application Developer
z/OS VSI for IBM Cloud VPC	Set up an IBM Cloud Enterprise account	N/A	1. Create a virtual server instance in IBM Cloud VPC 2. Set up a VPN server 3. Connect to z/OS virtual server instance
IBM Wazi for VS Code	N/A	N/A	Develop and test with IBM Wazi for VS Code

Table 1. Role-based tasks for use case 1

Use case 2: Standard virtualized z/OS with IBM Wazi for Eclipse

If you choose to use standard z/OS® VSI from IBM® Cloud VPC and IBM Wazi for Eclipse , follow this role-based roadmap to set up and use IBM Wazi as a Service.

Figure 1. Wazi as a Service use case 2

	Cloud Admin	z/OS System Admin	z/OS Application Developer
z/OS VSI for IBM Cloud VPC	Set up an IBM Cloud Enterprise account	N/A	1. Create a virtual server instance in IBM Cloud VPC 2. Set up a VPN server 3. Connect to z/OS virtual server instance
IBM Wazi for VS Code	N/A	N/A	Develop and test with IBM Wazi for Eclipse

Table 1. Role-based tasks for use case 2

Use case 3: Standard virtualized z/OS with web-based IBM Wazi for Dev Spaces

If you choose to use standard z/OS® VSI from IBM® Cloud VPC and web-based IBM Wazi for Dev Spaces , follow this role-based roadmap to set up and use IBM Wazi as a Service.

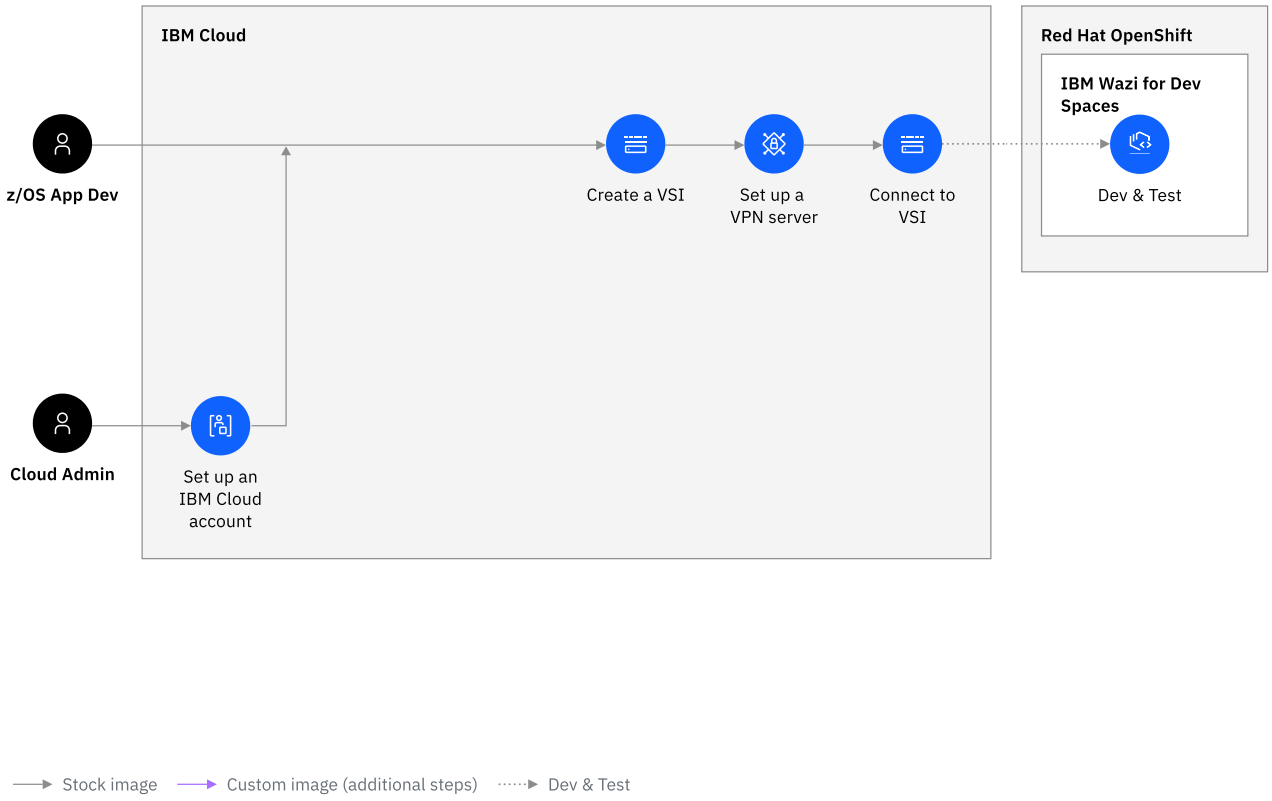


Figure 1. Wazi as a Service use case 3

	Cloud Admin	z/OS System Admin	z/OS Application Developer
z/OS VSI for IBM Cloud VPC	Set up an IBM Cloud Enterprise account	N/A	1. Create a virtual server instance in IBM Cloud VPC 2. Set up a VPN server 3. Connect to z/OS virtual server instance
IBM Wazi for VS Code	N/A	N/A	Deploy, develop, and test with IBM Wazi for Dev Spaces

Table 1. Role-based tasks for use case 3

Use case 4: Customized and virtualized z/OS with IBM Wazi for VS Code

If you choose to use IBM Wazi Image Builder and IBM Wazi for VS Code , follow this role-based roadmap to set up and use IBM Wazi as a Service.

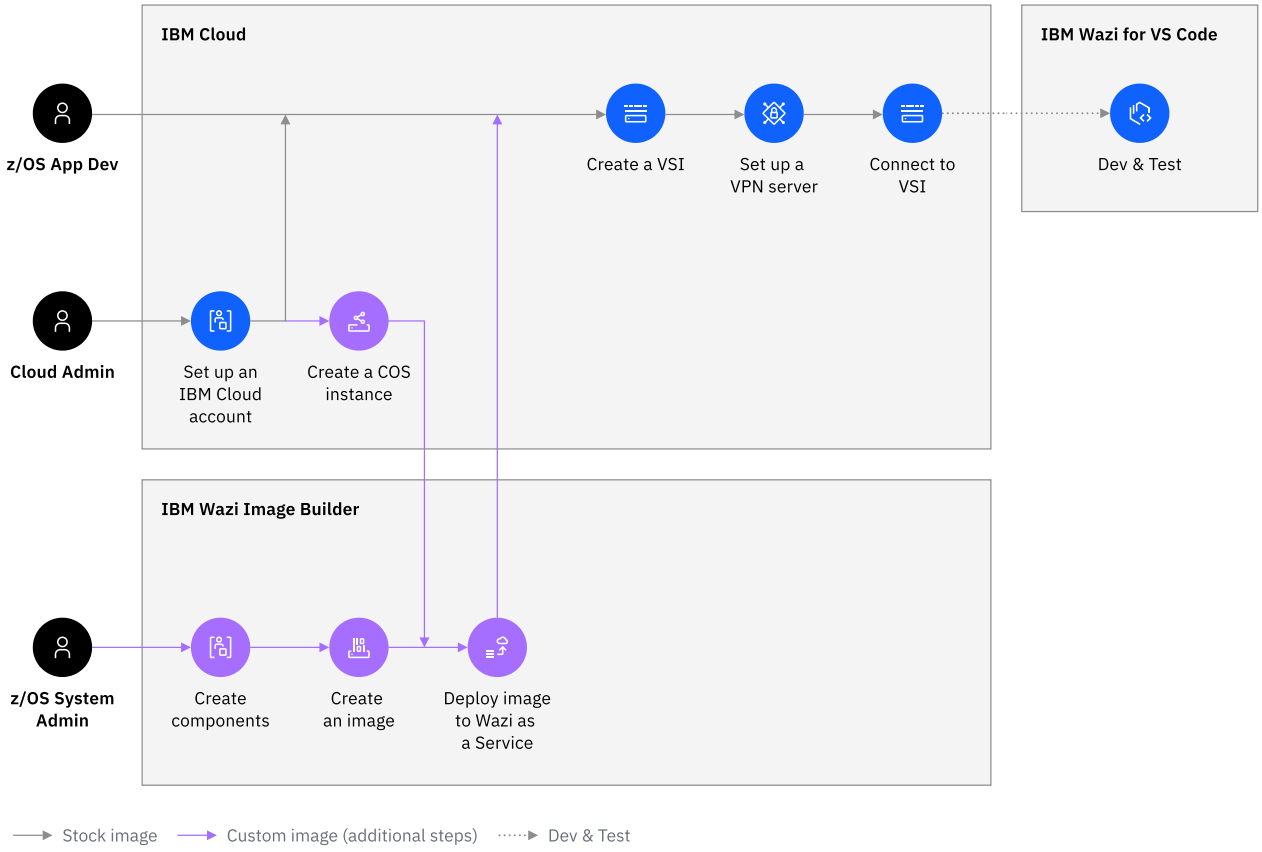


Figure 1. Wazi as a Service use case 4

	Cloud Admin	z/OS System Admin	z/OS Application Developer
IBM Cloud Object Storage (COS)	1. Set up an IBM Cloud Enterprise account 2. Create a COS instance	N/A	N/A
IBM Wazi Image Builder	N/A	1. Install and configure Wazi Image Builder 2. Create components 3. Create custom images 4. Deploy custom images	N/A
z/OS VSI for IBM Cloud VPC	N/A	N/A	1. Create a virtual server instance with custom image 2. Set up a VPN server 3. Connect to z/OS virtual server instance **Note:** You can also complete these steps with automation. For more information about automation samples and instructions, see LinuxONE and z/OS automation.
IBM Wazi for VS Code	N/A	N/A	Develop and test with IBM Wazi for VS Code

Table 1. Role-based tasks for use case 4

Use case 5: Customized and virtualized z/OS with IBM Wazi for Eclipse

If you choose to use IBM Wazi Image Builder and IBM Wazi for Eclipse , follow this role-based roadmap to set up and use IBM Wazi as a Service.

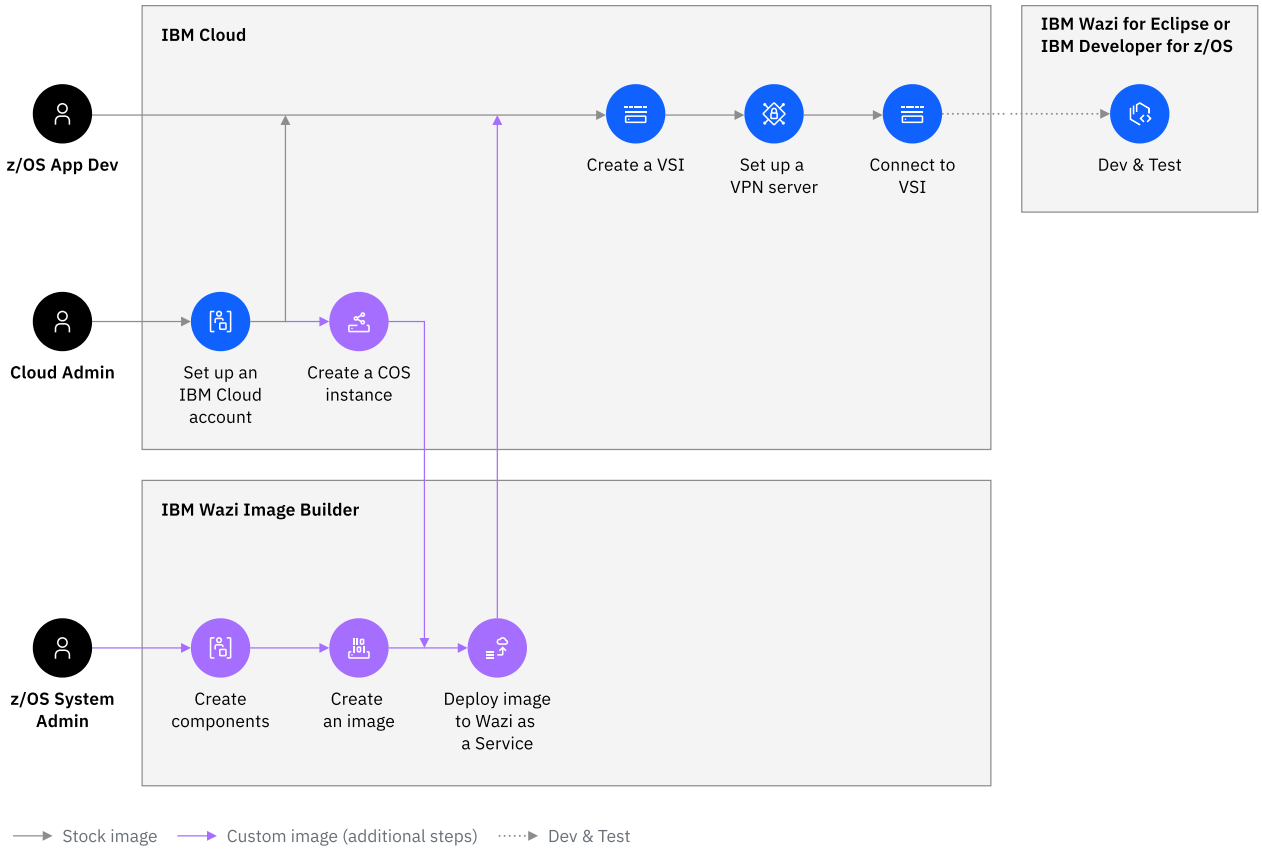


Figure 1. Wazi as a Service use case 5

	Cloud Admin	z/OS System Admin	z/OS Application Developer
IBM Cloud Object Storage (COS)	1. Set up an IBM Cloud Enterprise account 2. Create a COS instance	N/A	N/A
IBM Wazi Image Builder	N/A	1. Install and configure Wazi Image Builder 2. Create components 3. Create custom images 4. Deploy custom images	N/A

	Cloud Admin	z/OS System Admin	z/OS Application Developer
z/OS VSI for IBM Cloud VPC	N/A	N/A	1. Create a virtual server instance with custom image 2. Set up a VPN server 3. Connect to z/OS virtual server instance **Note:** You can also complete these steps with automation. For more information about automation samples and instructions, see LinuxONE and z/OS automation.
IBM Wazi for Eclipse	N/A	N/A	Develop and test with IBM Wazi for Eclipse

Table 1. Role-based tasks for use case 5

Use case 6: Customized and virtualized z/OS with web-based IBM Wazi for Dev Spaces

If you choose to use IBM Wazi Image Builder and web-based IBM Wazi for Dev Spaces , follow this role-based roadmap to set up and use IBM Wazi as a Service.

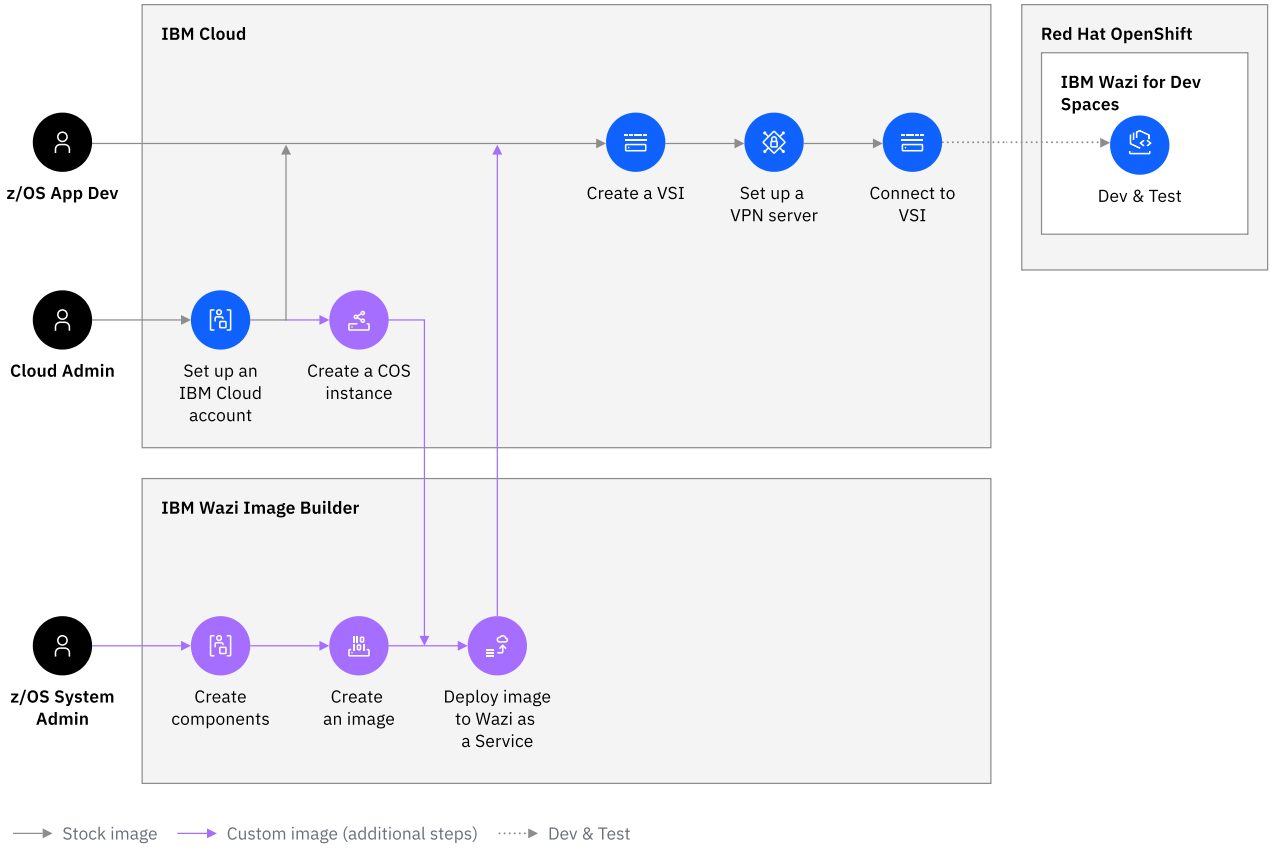


Figure 1. Wazi as a Service use case 6

	Cloud Admin	z/OS System Admin	z/OS Application Developer
IBM Cloud Object Storage (COS)	1. Set up an IBM Cloud Enterprise account 2. Create a COS instance	N/A	N/A

	Cloud Admin	z/OS System Admin	z/OS Application Developer
IBM Wazi Image Builder	N/A	1. Install and configure Wazi Image Builder 2. Create components 3. Create custom images 4. Deploy custom images	N/A
z/OS VSI for IBM Cloud VPC	N/A	N/A	1. Create a virtual server instance with custom image 2. Set up a VPN server 3. Connect to z/OS virtual server instance **Note:** You can also complete these steps with automation. For more information about automation samples and instructions, see LinuxONE and z/OS automation.
IBM Wazi for Dev Spaces	N/A	N/A	Deploy, develop, and test with IBM Wazi for Dev Spaces

Table 1. Role-based tasks for use case 6

Bringing your own image with Wazi Image Builder

You can use IBM® Wazi Image Builder to create and manage custom images from an existing Z platform. It includes a web UI with role-based access and REST APIs to streamline the creation process. You can then deploy the custom images to IBM Cloud® VPC to create a virtual server instance.

- **Architecture overview**
  IBM Wazi Image Builder enables enterprises to extract their on-premise IBM Z® platforms and deploy it as a custom image through API or web-based graphical interface for use within the IBM Cloud Virtual Private Cloud (VPC) infrastructure. Wazi Image Builder can be installed on an x86_64 Linux® system hosted either in a cloud or an on-premise physical or virtual machine.
- **Checklist**
  To get started rapidly, refer to the following recommended role-based roadmap.
- **Prerequisites**
  Learn about hardware and software requirements for Wazi Image Builder.
- **Downloading installation packages**
  Learn about how to download installation packages.
- **Validating the signed installers**
- **Installing Wazi Image Builder**
  IBM Wazi Image Builder provides a web server that enables users to use the browser to extract volumes, transfer the volumes to the image storage server, and deploy a created image to the target environment.
- **Configuring settings on the web server UI**
  Learn how to configure prerequisite settings on the web user interface to use Wazi Image Builder features.
- **Upgrading the web server**
  To upgrade the Wazi Image Builder web server, you must run the installer with the root user ID.
- **Setting up networking for on-premises z/OS environment**
  In IBM Wazi Image Builder, when you initiate the process to deploy a custom image to a target environment, you are asked to specify the PARMLIB and IEASYM suffix for the member that contains the TCP/IP symbol configuration.
- **Creating components**
  A component is a collection of assets from an IBM Z mainframe. A component can be reusable with other components from the same IBM Z mainframe in a mix and match fashion. You can use the created components to create images, and deploy the image as a custom image to your IBM Cloud VPC.
- **Monitoring and managing the created components**
  After you created a component, you can check the status of the extraction, and the details of your created components.
- **Creating z/OS images**
  After you create components from IBM Z mainframe system or existing z/OS instances, you can create z/OS® images by combining the created components into a single package. Then, you can deploy the images to IBM Cloud VPC.
- **Monitoring and managing the created images**
  After you create an image, you can stop, resume, or delete the image. Also, you can retrieve logs to check the details of the image.

- **Deploying images to a Wazi as a Service target environment**
  After you create an image, you can deploy the image to a Wazi as a Service target environment in IBM Cloud VPC.
- **Monitoring and managing the deployments**
  After you deploy a custom image to a Wazi as a Service target environment, you can use the Deployments page to check the deployment status, delete, or resume the deployments.
- **Using REST APIs**
  Wazi Image Builder supports REST APIs usage. You can access all functions of Wazi Image Builder without the web user interface, including creating components and images, deploying images, and monitoring the deployments.
- **Logs**
  Learn how to see logs for Wazi Image Builder.
- **Uninstalling the web server**
- **Sharing feedback**
  You can share anonymous data on the web server to help shape the product.

Architecture overview

IBM® Wazi Image Builder enables enterprises to extract their on-premise IBM Z® platforms and deploy it as a custom image through API or web-based graphical interface for use within the IBM Cloud® Virtual Private Cloud (VPC) infrastructure. Wazi Image Builder can be installed on an x86_64 Linux® system hosted either in a cloud or an on-premise physical or virtual machine.

Wazi Image Builder provides the following features:

- An application development and testing environment that can improve development infrastructure availability and flexibility.
- Current® levels of IBM z/OS® software that give access to new runtime capabilities for development and testing for enterprises.
- Mixed workload support for enterprises, which can help reduce the development costs.
- An approachable and portable environment for education on Z for enterprises.
- A web-based interface to extract, manage, and deploy the images from existing Z packages.
- Creating and managing images from various sources.
- Deploying images for developers and testers in a self-service automated way.
- Monitoring the status and availability of all created assets and target environments.

Source environments

Wazi Image Builder can work with your source environments to extract and deploy the necessary volumes to target environments. The following source environment types are supported:

Genuine z/OS on IBM Z
: ZD&T z/OS Extraction Utilities must be installed and configured on such a source environment to extract z/OS artifacts.

Existing z/OS instances
: ZD&T z/OS Extraction Utilities must be installed and configured on such a source environment to extract z/OS artifacts.

You can specify and configure source environments on the web server. For more information, see Adding source environments.

ZD&T z/OS Extraction Utilities

ZD&T z/OS Extraction Utilities is required if you want to create components from IBM Z platforms, either the physical or emulated IBM Z platforms. It needs to be installed on one or more IBM Z platforms to allow Wazi Image Builder to use the IBM Z platforms as source environments to extract z/OS resources. For more information, see Installing ZD&T z/OS Extraction Utilities.

Target environments

A target environment is a IBM Cloud VPC environment, where you can deploy extracted z/OS artifacts as a custom image, which can then be used to enable a cloud native development and test experience for z/OS.

Web server

Wazi Image Builder web server enables users to use the browser to extract artifacts such as volumes, and transfer the artifacts to the storage server. Then, you can deploy the custom images to your target environment.

The web server provides the following features:

- Integrating with company LDAP account to enable administrators to set up accounts with minimum efforts.
- Scheduling the extraction or deployment for another date or time that you want.
- Flexible functions to select required volumes from the source machine, and transfer the volumes to the target machine.

For more information, see Installing Wazi Image Builder.

Storage server

Wazi Image Builder stores extracted information on the intermediary storage machine, for example, SFTP server. Extracted information is never deleted from the storage server until the information is manually deleted. You need to prepare such a storage server and ensure enough storage on the storage server to hold the required artifacts.

Note: The web server and storage server can be configured on the same machine. If you want to use one machine, you need to have large storage on the machine.

Checklist

To get started rapidly, refer to the following recommended role-based roadmap.

For more information about users and roles, see Managing users and roles.

Server administrator	z/OS system administrator	Builder
1. Downloading installation packages 2. Validating the signed installers 3. Installing Wazi Image Builder 4. Installing the web server 5. Setting up a storage server	1. Installing z/OS Extraction Utilities 2. Setting up networking for on-premises z/OS environment 3. Connecting to your storage server 4. Managing users and roles 5. Adding source environments 6. Adding Wazi as a Service target environments	1. Creating components 2. Monitoring and managing the created components 3. Creating z/OS images 4. Monitoring and managing the created images 5. Deploying images to a Wazi as a Service target environment 6. Monitoring and managing the deployments

Prerequisites

Learn about hardware and software requirements for Wazi Image Builder.

Wazi Image Builder can be installed on an x86_64 Linux® system hosted either in a cloud or an on-premise physical or virtual machine. For a complete list of hardware and software requirements, you can generate the report from Software Product Compatibility Reports.

For other prerequisites, see the following details.

- Storage server requirements
- Source environments: z/OS system requirements
- Target environments: Wazi as a Service requirements

Storage server requirements

To install and run Wazi Image Builder, a storage server to host the artifacts, such as z system volumes and metadata must be set up. To transfer volumes images files from the storage server or to the storage server, you must choose SFTP as the transferring method.

- **Disk space**
  - At least 14 GB of disk space is needed to install the web server.
  - Sufficient space is needed to hold numerous and potentially large files for extracted IBM® Z volumes.
- **Software requirements**
  - A running SFTP server

z/OS system requirements

If you want to set up source environments from either genuine z/OS® on IBM Z® or existing z/OS instances to extract and provision the necessary volumes or data sets, the following z/OS system prerequisites must be met.

- Supported z/OS versions: V2.4 and V2.5.
- You must install all the PTFs that are identified with the SMP/E FIXCAT of `IBM.TargetSystem-RequiredService.AlternateHypervisors`. Installation of all these PTFs must be completed on the z/OS system before you deploy the custom image. For more information, see IBM Fix Category Values and Descriptions.
- You need to install ZD&T z/OS Extraction Utilities to create components from IBM Z platforms. For more information, see Installing ZD&T z/OS Extraction Utilities.
- To run the extraction code, the following minimum requirements for the z/OS user ID must be met:
  - For 32-bit Java™, the user ID needs a minimum region of approximately 150 MB.
  - For 64-bit Java, the user ID needs a minimum region of approximately 250 MB.

  The region size can be specified in the OMVS segment (**ASSIZEMAX**) for remote login, for example, SSH, and for TSO login through the TSO segment (**SIZE** or **MAXSIZE**). If OMVS or TSO segments do not limit sizes, global settings might need to be adjusted in the BPXPRM*xx* member of PARMLIB. These values are only estimates, and your environment might need more, or you might be able to specify less.
  Example

  ```
  alu ibmuser OMVS(ASSIZEMAX(262144000))
  ```

- You need to set up the networking for your on-premises z/OS environment and validate the configuration before you create your custom image. For more information, see Setting up networking for on-premises z/OS environment.
- The TCP/IP profile must contain the following values:

  z/OS IP address
  : This value must be defined in either the HOME or INTERFACE statement. Using system symbolic parameters for the IP address is not supported.

  z/OS default route address
  : This value must be defined using the BEGINRoutes statement. Using system symbolic parameters for the default route is not supported. Routing configurations via OMPROUTE are also not supported.

- If you want to extract volumes from z/OS systems, the following requirements are needed.
  Required
  - An SSH server must be running and accessible by the system to run Wazi Image Builder.
  - The SFTP client must be able to connect to the Wazi Image Builder storage server.
  - To use SFTP, Java 1.6 or later versions must be installed, and the PATH needs to be specified in the $HOME/.profile and pointed to the bin directory of the Java installation.
  - Make sure to grant access to each volume that is extracted. For more information, see Creating a volume component.
  - Make sure to grant READ access to DFDSS program ADRDSSU.
  - At least one offline DASD device must be available on the source z/OS system before the system is extracted.

  Optional
  - Configure zEnterprise® Data Compression (zEDC) if it is available. Grant READ access to the resource FPZ.ACCELERATOR.COMPRESSION in SAF class FACILITY to the user ID that is used in the Wazi Image Builder.
  - Grant READ access to resource STGADMIN.ADR.DUMP.CNCURRNT in SAF class FACILITY.
  - To extract on-premise z/OS volumes, the user ID that runs the extraction must have the capability to use substantial CPU time, as this operation might take a considerable amount of time. To prevent potential errors, it is recommended to grant the user ID unlimited CPU time, at least during the extraction process. You can use the TSO OMVS segment to modify a user's maximum CPU time.
    Example

    ```
    ALTUSER userid OMVS(CPUTIMEMAX(86400))
    ```

- If you want to extract volumes from an existing z/OS instance, Java must be installed, and the user who creates volume components must have access to Java after login.

Wazi as a Service requirements

When you install the web server, you will be asked to install Terraform® automatically or manually from the Hashicorp Linux Repository. Terraform is required to allow Wazi Image Builder to convert the deployed images for use in IBM Cloud® VPC. You need to have access to the Internet to install Terraform. The Terraform binary will be installed to a path that is already on the OS path.

Make sure that you have an instance of IBM Cloud Object Storage available and have created a bucket in IBM Cloud Object Storage to store your images. For more information, see Granting access to IBM Cloud Object Storage to import images.

Downloading installation packages

Learn about how to download installation packages.

To download the installation package that is required for IBM® Wazi Image Builder, complete the following steps:

1. Log on to Passport Advantage®.
2. Select Software download and Media access.
3. Select Program offering and agreement number, and then click Continue.
4. Enter the part description or part number, and then click Finder.
5. Optionally, you can click the alphabetical order list to display and view the product by name.
6. Select All operating systems in the Operating system field, and All languages in the Languages field. Then, click Go.
7. If you want to download the individual media from the list, click Select individual files to expand the list.
8. Verify the e-assemblies that you want to download with the lists in the following tables.

To download Wazi Image Builder fix packs, go to IBM Fix Central.

After a package is downloaded, you can verify the integrity of the downloaded package by using the .md5 checksum file. To do a checksum on a downloaded package, use command **`md5sum -c <checksum_file_name>`**.

Table 1. IBM Wazi Image Builder V1.4

	Name	Part No.	Image Classification	File Name
1	IBM Wazi Image Builder 1.x Installation Multilingual eAssembly	G06Y9ML	Required	N/A
2	IBM Wazi Image Builder 1.4 Installation Multilingual	M0FHNML	Required	wib-install.tgz
3	IBM Wazi Image Builder 1.4 Installer checksum Multilingual	M0FHPML	Required	wib.md5
4	IBM Wazi Image Builder Quick Start Guide	M0B6KML	Optional	wib_quickstart.pdf

Validating the signed installers

The .tgz files that you download from IBM are compressed installation packages. Each Wazi Image Builder installation package contains the following signed installer and the corresponding signature file.

Installer	Signature file
Web server installer wib-install	wib-install.cosign.sig

You can validate any signed installer with its signature file by running the following command.

```
openssl dgst -sha256 -verify <PUBLICKEYNAME> -signature <signature_file> <installer_to_sign>
```

For example, run the following command from the directory that contains the extracted installer.

```
openssl dgst -sha256 -verify signature/public.pem -signature wib-install.sig wib-install
```

If the installer is valid, you can receive the message of `Verified OK`.

Installing Wazi Image Builder

IBM Wazi Image Builder provides a web server that enables users to use the browser to extract volumes, transfer the volumes to the image storage server, and deploy a created image to the target environment.

Before you begin, make sure that you meet the hardware and software requirements and download the installation packages.

- **Installing the web server**
  To install the Wazi Image Builder web server, run the installer with the root user ID.

- **Installing a signed certificate on the web server**
  To access to the Wazi Image Builder web server on your internal server, you need to install a signed certificate that is used by the web server.
- **Setting up a storage server**
  The storage is a crucial function of Wazi Image Builder. You can transfer and store extracted artifacts on the storage server by using SFTP (SSH File Transfer Protocol). You must have administrator privileges to complete this task.
- **Installing ZD&T z/OS Extraction Utilities**
  To create components from z/OS®, either genuine z/OS on IBM Z® or an existing z/OS instance that runs on Linux®, you need to install ZD&T z/OS Extraction Utilities.
- **Authentication for the Wazi Image Builder application**
  Before you use the Wazi Image Builder web server, you can modify how authentication is accomplished by Wazi Image Builder.
- **Starting and stopping the web server**
  When you installed the Wazi Image Builder web server and selected to start the web server manually, you need to run the script to start the web server and ensure that the server process runs under the user ID that is specified during the installation.

Installing the web server

To install the Wazi Image Builder web server, run the installer with the root user ID.

1. Open the directory where the installation package wib-install.tgz is stored.

   ```
   cd <directory>
   ```

2. Decompress the installation package.

   ```
   tar -xvf <package_name>
   ```

   When the decompression is complete, the installers can be found in the target directory. If you don't specify a target directory, the installers can be found in the same directory where the installation package is.

   Web server installer
   : wib-install

3. Run the installer, and then follow instructions to complete the installation.

   ```
   ./<installer_name>
   ```

   Standard installation
   : This option will install Wazi Image Builder with the following default settings:

     Default user ID for running the web server
     : wib

     Default web server installation directory
     : /opt/ibm/wib

     Default port for accessing the web server
     : 9443

     Default user ID for logging into the web server
     : wibadmin

   Custom installation
   : You can follow instructions to modify all default settings and complete the installation.

   Note: Terraform is required only when you deploy custom images to your target environment in IBM Cloud®. By running the web server installer, you can choose to automatically install the Terraform binary from HashiCorp. You can also install Terraform manually.
   Alternatively, you can run the following command to install the web server silently. Silent installation does not install any required dependencies. You need to install the dependencies before you start the installer silently. For the list of dependencies, see Hardware and software requirements.

   ```
   ./<installer_name> --install --wib --wib_port=<port_number> --wib_user=<username> --wib_path=
   <installation_path> --wib_password=<password>
   ```

4. After the installation completes, run the following command to verify whether the installation is successful.
   - RHEL operating system

     ```
     rpm -qa | grep wibapp
     ```

- Ubuntu operating system

```
dpkg -l | grep wibapp
```

If the web server is installed successfully, information, including the right version number, of the installed application is returned.

What's next

If you set the web server to start automatically, you can find the web server URL in the output. Before you log in with your user ID and password, complete the following steps:

- Installing a signed certificate on the web server
- Setting up a storage server
- Installing ZD&T z/OS Extraction Utilities

Installing a signed certificate on the web server

To access to the Wazi Image Builder web server on your internal server, you need to install a signed certificate that is used by the web server.

To generate your own pkcs12 keystore (wibkey.p12) that contains the certificate and put the encrypted password in the server.env file, follow these steps:

1. Check the installation directory of your installed web server. For example, /opt/ibm/wib is the default installation directory, but you can specify your own installation directory during the installation process.
2. Run the following command to generate wibkey.p12 and put it in the *<installation_directory>*/wib-server/resources/security.

```
openssl pkcs12 -export -out wibkey.p12 -inkey cert.key -in cert.crt -password pass:
<password_value>
```

3. Modify the encrypted key store password.
   a. Get the encryption key that is specified by wlp.password.encryption.key in the *<installation_directory>*/wib-server/bootstrap.properties; for example, /opt/ibm/wib/wib-server/bootstrap.properties
   b. Run the following command where you installed the web server.

```
<installation_directory>/Liberty/bin/securityUtility encode --encoding=aes --key=
<encryption_key_found_above> <password_value>
```

   Note: To ensure that the `securityUtility` command can be run successfully, the Java™ path must be set up. To set up the Java path, you need to add the JAVA_Home in the environment variable or add Java in the Path environment variable.
   c. Modify the *<installation_directory>*/Liberty/usr/servers/wib-server/server.env file with your encoded password value.

```
POSTGRES_SERVER=xxx
   POSTGRES_PORT=5432
   POSTGRES_NAME=xxx
   POSTGRES_USER=xxx
   POSTGRES_PASSWORD=xxx
   POSTGRES_DRIVER_DIR=xxx

   KEYSTORE_PASSWORD={aes}AG6iTGAo/v3DbfEv+7FgNH4oaoanLomL5enZr86JiS0p
```

Setting up a storage server

The storage is a crucial function of Wazi Image Builder. You can transfer and store extracted artifacts on the storage server by using SFTP (SSH File Transfer Protocol). You must have administrator privileges to complete this task.

Before you configure the storage server, you need to install SFTP.

Setting up an SFTP server

To enable an SFTP server, complete the following steps:

1. Run the following command to check whether SSH is installed.

```
rpm -qa|grep ssh
```

2. If SSH is not installed, run the following command.

```
yum install openssh-server
```

3. Open the port 22.

```
iptables -I INPUT -p tcp --dport 22 -j ACCEPT
```

4. Create a directory and change the authority.

```
mkdir -p /data/sftp
chmod 701 /data
```

5. Add a group, and a user for SFTP. Then, provide your new password.

```
groupadd sftpusers
useradd -g sftpusers -d /upload -s /sbin/nologin sftpuser
passwd sftpuser
```

6. Create a directory for the file transfer. Then, make the changes that you need.

```
mkdir -p /data/sftpuser/upload
chown -R root:sftpusers /data/sftpuser
chown -R sftpuser:sftpusers /data/sftpuser/upload
```

7. Add the following lines in the /etc/ssh/sshd_config.
   Match Group *sftpuser*
   ForceCommand internal-sftp
8. Restart the SFTP service.

```
service sshd restart
```

What's next

After you start the web server, you can connect to your storage server by following the instructions in Connecting to your storage server.

Installing ZD&T z/OS Extraction Utilities

To create components from z/OS®, either genuine z/OS on IBM Z® or an existing z/OS instance that runs on Linux®, you need to install ZD&T z/OS Extraction Utilities.

ZD&T z/OS Extraction Utilities is a component to be installed on one or more z/OS source environments to extract z/OS, CICS®, and Db2® resources.

To extract resources from an existing z/OS instance that runs on Linux, install ZD&T z/OS Extraction Utilities by transferring *<installation_directory>*/zSystem/zdtMainframeSFTP.jar to /usr/lpp/IBM/zdt of the Linux system.

To extract resources from genuine z/OS on IBM Z, install ZD&T z/OS Extraction Utilities by following the steps:
Note: If you have previously installed ZD&T z/OS Extraction Utilities and want to use the automated system volume discovery tool when you create a volume component, make sure that you complete steps 4 to 6.

1. After the installer wib-install.tgz runs on a Linux machine to install the web user interface, directory *<installation_directory>*/zSystem contains the IBM® Z components, the program directory file HALME21.pdf, and other files, including jar file zdtMainframeSFTP.jar.
2. Install the IBM Z components by following the instructions in HALME21.pdf, specifically in section 6.0 "Installation Instructions". The data set hlq.IBM.HALME21.F2 obtained at step 6.1.3 in HALME21.pdf is a usable program library. If you want to install Wazi Image Builder without SMP/E, complete the following steps:
   a. Choose a value for dsnprefix. Then, rename hlq.IBM.HALME21.F2 to dsprefix.SFEUAUTH, and rename hlq.IBM.HALME21.F3 to dsprefix.SFEUEXEC.
   b. Go to step 6.1.11 by skipping steps from 6.1.4 to 6.1.10.
3. Transfer *<installation_directory>*/zSystem/zdtMainframeSFTP.jar to the z/OS UNIX System Services folder /usr/lpp/IBM/zdt that is located on the source z/OS.
4. Transfer *<installation_directory>*/zSystem/feucvoli from the Tools Server to the z/OS UNIX System Services folder /usr/lpp/IBM/zdt that is located on the source z/OS by using Secure Copy Protocol (SCP) instead of Secure File Transfer Protocol (SFTP).

5. Ensure that /usr/lpp/IBM/zdt/feucvoli is executable by running the **chmod 755 /usr/lpp/IBM/zdt/feucvoli** command in the UNIX System Services on z/OS.
6. To extract volumes, add export zdtAuth=STEPLIB_specified_in_feuc to your z/OS UNIX System Services login profile to point to the same STEPLIB as you specified in /usr/lpp/IBM/zdt/feuc.

Note:

- Instead of using the default directory /usr/lpp/IBM/zdt, you can use a different path and specify the value as the ZD&T z/OS Extraction tools installation directory with the web server. For instructions or more information about source environment types, see section Adding source environments.
- Ensure that you have write access to the /tmp directory in the source z/OS UNIX System Services, because the utility will write a lot of logs into it. According to best practices, clean this folder periodically.

Authentication for the Wazi Image Builder application

Before you use the Wazi Image Builder web server, you can modify how authentication is accomplished by Wazi Image Builder.

- **Default authentication and authorization**
  Wazi Image Builder comes with a default user ID and password. You can also change the default password.
- **Basic authentication**
  By default, Wazi Image Builder uses a basic registry as the user registry for authentication and provides a default user ID 'wibadmin'. Also, you can add more users to this registry.
- **Lightweight Directory Access Protocol (LDAP)**
  Learn about the steps to enable LDAP authentication for Wazi Image Builder.

Default authentication and authorization

Wazi Image Builder comes with a default user ID and password. You can also change the default password.

The default user ID 'wibadmin' is defined in file *<installation_directory>*/Liberty/usr/servers/wib-server/basicauth-conf.xml, and the default password of this user ID is set during installation. This password is authorized as an administrator account on the web server. To change the default password, complete the following steps.

1. Find the encryption key that is specified by **`wlp.password.encryption.key`** in the file *<installation_directory>*/Liberty/usr/servers/wib-server/bootstrap.properties.
2. If the Java™ path is not set up, you need to set up the path before you run the next command.
3. Run the following command where you installed the web server.

   ```
   <installation_directory>/Liberty/bin/securityUtility encode --encoding=aes --key=
   <encryption_key_found_above> <new password value>
   ```

   Copy the following encoded string into file *<installation_directory>*/Liberty/usr/servers/wib-server/basicauth-conf.xml. For example, replace the text in bold with the new encoded password value.

   ```
   <!-- user authentication -->
   <basicRegistry
   id="_home_markdown_jenkins_workspace_Transform_in_SSSTT9_1.0.0_com.ibm.zsys.rdt.tools.user.guide
   .doc_topics_default_ps_wib" realm="wib">
   <user name="wibadmin"
   password="*{aes}AM1LZsnwLRNsVtYAiwqhVDO9/RL+NgYthDZXZhQgARtB*"/>
   </basicRegistry>
   ```

For more information about authorization on the web server, see Managing users and roles.

Basic authentication

By default, Wazi Image Builder uses a basic registry as the user registry for authentication and provides a default user ID 'wibadmin'. Also, you can add more users to this registry.

If you need to change the default authentication before you modify the basic authentication, see Default authentication and authorization.

To add more users to the basic registry, complete the following steps before you start the server.

1. Find the encryption key that is specified by **wlp.password.encryption.key** in file *<installation_directory>*/Liberty/usr/servers/wib-server/bootstrap.properties.
2. If the Java™ path is not set up, you need to set up the path before you run the next command.
3. Run the following command where you installed the web server.

   ```
   <installation_directory>/Liberty/bin/securityUtility encode --encoding=aes --key=
   <encryption_key_found_above> <new password value>
   ```

   Copy the following encoded string into file *<installation_directory>*/Liberty/usr/servers/wib-server/basicauth-conf.xml. For example, replace the text in bold with the new encoded password value.

   ```
   <basicRegistry
   id="_home_markdown_jenkins_workspace_Transform_in_SSSTT9_1.0.0_com.ibm.zsys.rdt.tools.user.guide
   .doc_topics_new_user_wib" realm="wib">
           <user name="wibadmin" password="{aes}AM1LZsnwLRNsVtYAiwqhVDO9/RL+NgYthDZXZhQgARtB"/>
                   <user name="new user" password="<new user password>"/>
                   <user name="another new user" password="<another new user password>"/>
       </basicRegistry>
   ```

   This step only allows the user to authenticate to the web server. For more information about authorization, see Managing users and roles.

Lightweight Directory Access Protocol (LDAP)

Learn about the steps to enable LDAP authentication for Wazi Image Builder.

The Lightweight Directory Access Protocol (LDAP) settings for the server are defined in the *<installation_directory>*/Liberty/usr/servers/wib-server/ldap-conf.xml file. To enable the support, refer to Configuring LDAP user registries in Liberty to modify the file and define your LDAP configuration before you start the server. The appSecurity-2.0, ldapRegistry-3.0, and transportSecurity-1.0 Liberty features are already included in the configuration of the server, and these steps are unnecessary. The truststore of the server is located in the *<installation_directory>*/Liberty/usr/servers/wib-server/resources/security/wibtrust.p12. The password for the wibtrust.p12 file is 'changeme', and the client needs to know the information to add their certificate to the file. If you are communicating with an SSL-enanbled LDAP server, you need to add your signer certificate for the LDAP server to this keystore file.

This step only allows the user to authenticate to the web server. For more information about authorization, see Managing users and roles.

Starting and stopping the web server

When you installed the Wazi Image Builder web server and selected to start the web server manually, you need to run the script to start the web server and ensure that the server process runs under the user ID that is specified during the installation.

Changing the user ID

You can change the user ID by modifying *WIB_USER* in the *<installation_directory>*/bin/wib_env.sh script file. You also need to change the file ownership of all the files in the directory *<installation_directory>*/Liberty/usr/servers/wib-server/ to the user ID, for example, **chmod -R newuser:root <installation_directory>/Liberty/usr/servers/wib-server/**.

If the web server is already started, you can run the following command before you restart the server by using the newly specified user ID.

```
<installation_directory>/bin/stopServer.sh
```

Starting the web server manually

To start the web server, you can run the command by using any user ID. However, the user ID must have the permission to switch to the user ID that is specified during the installation. By default, you can use the sudo command to switch to another user ID.

To start the web server, run the following command.

```
<installation_directory>/bin/startServer.sh
```

Note: If you selected to set the web server to start and stop automatically, you can go to the URL in the installation output directly. Open the browser, and enter the URL that is contained in the output. Then, enter the default user ID (wibadmin) and Password (password) to log in to the web server.

Stopping the web server manually

If you need to stop the web server, run the following command.

```
<installation_directory>/bin/stopServer.sh
```

Configuring settings on the web server UI

Learn how to configure prerequisite settings on the web user interface to use Wazi Image Builder features.

- **Connecting to your storage server**
  After you enabled an SFTP protocol, you can go to the Wazi Image Builder web server UI to connect to the storage server.
- **Managing users and roles**
  You can add and manage users the web server to control access to Wazi Image Builder.
- **Adding source environments**
  Source environments are configured to create components or images that are used in target environments for development and testing.
- **Adding Wazi as a Service target environments**
  You can use the Wazi Image Builder web user interface to add a Wazi as a Service target environment in IBM Cloud® Virtual Private Cloud (VPC). Then, you can deploy z/OS® images to IBM Cloud and monitor the deployments from the Deployments page.

Connecting to your storage server

After you enabled an SFTP protocol, you can go to the Wazi Image Builder web server UI to connect to the storage server.

To connect to an SFTP server, complete the following steps:

1. Click Storage on the home page, or click the upper-right settings button to navigate to Storage.
2. Select the protocol that you want to use.
3. Enter the qualified hostname of the storage server along with the port and directory.
4. Enter the credentials of the storage server.
5. Click Save.

Note: The credentials for the storage server are the only credentials that the web server stores in its local database and are encrypted using AES 128-bit encryption.

Managing users and roles

You can add and manage users the web server to control access to Wazi Image Builder.

Note: The Users & roles page requires administrator privileges. To configure the Users & roles page, you need to configure the administrators first before you configure the users.

Configuring a user registry

To add a user to the web server, you must ensure that the user ID exists in your connected user registry.

For more information about configuring your user registry, see Authentication for the Wazi Image Builder application.

Adding a user

To add a new user who exists in the user registry to the web server, complete the following steps:

1. Click Users & roles on the home page, or click the upper-right settings button to navigate to Users & roles.
2. Click Add user.
3. Enter a user ID that exists in the user registry, and select the corresponding roles for the user.
    - For the user to create z/OS® images or deploy a z/OS images to a target environment, select the Developer/Tester role.
    - For the user to extract z/OS volumes for use on the web server, select the Builder role. Builder role includes the capabilities of the Developer/Tester role.
4. Click Save.

Adding source environments

Source environments are configured to create components or images that are used in target environments for development and testing.

The source environment configuration can be IBM Z® mainframe systems.

Notes:

- For z/OS on IBM Z physical hardware source environments that are added, you can add, edit, or delete existing source systems. However, a source environment cannot be deleted if components created from this source environment exist.
- If you want to set up source environments from either genuine z/OS on IBM Z physical hardware or existing z/OS instances to extract and provision the necessary volumes or data sets, the z/OS system requirements must be met.
- Extracting from a z/OS instance initially provisioned from Extended ADCD is also considered as extracting from an existing z/OS instance; therefore, the z/OS system requirements apply.

- **Adding a source environment from a z/OS system on IBM Z physical hardware**
  To create components from a real IBM Z environment, you need to configure an IBM Z mainframe first. Then, you can create components by extracting, compressing, and transferring artifacts from an existing z/OS® environment on IBM Z.
- **Adding a source environment from an existing z/OS instance run by ZD&T**
  You can create volume components quickly from an existing z/OS instance that runs on a host Linux® system by adding this system as a source environment on the web server.

Adding a source environment from a z/OS system on IBM Z physical hardware

To create components from a real IBM Z® environment, you need to configure an IBM Z mainframe first. Then, you can create components by extracting, compressing, and transferring artifacts from an existing z/OS® environment on IBM Z.

To create a new IBM Z mainframe source environment, complete the following steps:

1. Click Source environments on the home page, or click the upper-right settings button to navigate to Source environments.
2. Click Add source environment.
3. Select the IBM Z platform source environment type.
4. Specify a unique source environment name that can be identified when you create components or images.
5. Enter the specified hostname and SSH port of the IBM Z mainframe system that is used to extract z/OS data.
6. Specify the ZD&T z/OS Extraction tools installation directory. The default path is /usr/lpp/IBM/zdt/.
7. Click Add source environment.

For the IBM Z mainframe systems that are added, you can edit or delete the existing source system.
Note: The configured mainframe system cannot be deleted if some components were created from the mainframe system.

Adding a source environment from an existing z/OS instance run by ZD&T

You can create volume components quickly from an existing z/OS instance that runs on a host Linux® system by adding this system as a source environment on the web server.

To add a Linux source environment, complete the following steps:

1. Click Source environments on the home page, or click the upper-right settings button to navigate to Source environments.
2. Click Add source environment.
3. Select the Linux running a ZD&T emulator source environment type.
4. Specify a unique source environment name that can be identified when you create volume components.
5. Enter the specified hostname and SSH port of the host Linux system. Then, enter the location of the ZD&T emulator devmap on the source environment.
6. If you want to define the directory where the compressed files will be stored before they are transmitted to the storage server, enable the Compression option. Then, enter a working directory that can temporarily store the compressed files. For example, /home/ibmsys1/zdt.
7. Enter the specified hostname and SSH port of the z/OS® instance on Linux.
8. Specify the ZD&T z/OS Extraction tools installation directory. The default path is /usr/lpp/IBM/zdt/.
9. Click Connect ZD&T environment.
10. Enter the credentials to the Linux environment that you are connecting to, and click Submit.

Adding Wazi as a Service target environments

You can use the Wazi Image Builder web user interface to add a Wazi as a Service target environment in IBM Cloud® Virtual Private Cloud (VPC). Then, you can deploy z/OS® images to IBM Cloud and monitor the deployments from the Deployments page.

Before you begin, make sure that you have a valid IBM Cloud account and a IBM Cloud Object Storage instance that is set up by the Cloud administrator. For more information, see Use cases.

To add a Wazi as a Service target environment, complete the following steps:

1. Click Target environments on the home page, and click the Settings button. Or, you can click the upper-right Configure settings button to navigate to Target environments.
2. Click Add target to add a Wazi as a Service target environment.
3. Enter a unique name that can be used to identify the target environment.
4. Optional: Enter some notes to describe the target environment.
5. Enter your IBM Cloud API key, and then click Establish connection.
   Note: To view or manage the IBM Cloud API keys that are associated with your user identity or the ones that you have access to manage for other users in the account, go to API keys.
6. Select a resource group from the Resource group drop-down list.
7. Select an IBM Cloud Object Storage instance from the Cloud Object Storage instance drop-down list.
   Note: This instance must be already created in IBM Cloud to be used for persisting z/OS images. To create an IBM Cloud Object Storage, go to Cloud Object Storage. To authorize the image source service, see Granting access to IBM Cloud Object Storage to import images.
8. Select a bucket within your selected IBM Cloud Object Storage instance from the Bucket drop-down list.
   Note: This bucket is a temporary storage location where z/OS images are persisted before they are registered as custom images in the VPC. To view or manage your buckets, go to IBM Cloud, select Resource List from the navigation menu, and then select the IBM Cloud Object Storage instance for persisting created images.
9. Select a region from the VPC region for custom image drop-down list. This region is where your custom image exists after it is deployed and converted for use in IBM Cloud.
   Important: Make sure that you select a region where z/OS virtual server instances are supported. See the Available regions table for a complete list of supported regions.
   Note: If you update Wazi Image Builder from a previous version, the VPC region for custom image field is empty. By default, the deployed image goes to the region that you specified for the Cloud Object Storage. You can edit the target environment to change the VPC region.
10. Optional: Enable the Advanced settings option, and modify the number of volumes that you want to extract and transfer simultaneously.
11. Click Add a target environment.

Table 1. Available regions

Location	Region	API endpoints
Brazil (São Paulo)	`br-sao`	`br-sao.iaas.cloud.ibm.com`
Canada (Toronto)	`ca-tor`	`ca-tor.iaas.cloud.ibm.com`
United Kingdom (London)	`eu-gb`	`eu-gb.iaas.cloud.ibm.com`
Japan (Tokyo)	`jp-tok`	`jp-tok.iaas.cloud.ibm.com`
US East (Washington DC)	`us-east`	`us-east.iaas.cloud.ibm.com`
US South (Dallas)	`us-south`	`us-south.iaas.cloud.ibm.com`
Spain (Madrid)	`eu-es`	`eu-es.iaas.cloud.ibm.com`

Upgrading the web server

To upgrade the Wazi Image Builder web server, you must run the installer with the root user ID.

1. Open the directory where the installation package wib-install.tgz is stored.

   ```
   cd <directory>
   ```

2. Change the authority of the installation package.

   ```
   chmod 755 <package_name>
   ```

3. Decompress the installation package.

   ```
   tar -xvf <package_name>
   ```

   When the decompression is complete, the installers can be found in the target directory. If you don't specify a target directory, the installers can be found in the same directory where the installation package is.

   Web server installer
   : wib-install

4. Run the installer, and then follow instructions to upgrade.

   ```
   ./<installer_name>
   ```

   You can upgrade the web server with the following options.
   - Select **1** to reinstall the web server with existing settings except the password of the web server.
   - Select **2** to refresh the installation of the web server by removing all existing settings and configuring new settings.

   More information about the settings can be found in topic Installing the web server.
   Alternatively, you can upgrade the web server silently by using the following command.

   ```
   ./<installer_name> --update --wib
   ```

5. After the installation completes, run the following command to verify whether the installation is successful.
   - RHEL operating system

     ```
     rpm -qa | grep wibapp
     ```

   - Ubuntu operating system

     ```
     dpkg -l | grep wibapp
     ```

   If the web server is installed successfully, information, including the right version number, of the installed application is returned.

What's next

If you set the web server to start automatically, you can find the web server URL in the output. If you selected to start the web server manually, you can follow the instructions in topic Starting and stopping the web server.

Setting up networking for on-premises z/OS environment

In IBM® Wazi Image Builder, when you initiate the process to deploy a custom image to a target environment, you are asked to specify the PARMLIB and IEASYM suffix for the member that contains the TCP/IP symbol configuration.

Some of these symbols are modified in the IBM Wazi as a Service deployment process so that the IP address that is dynamically assigned by IBM Cloud® is injected into this member and is used by your z/OS® TCP/IP profile when it runs in IBM Cloud.

You have two options to set up the networking for your on-premises z/OS environment.

Option 1: Using the zOS-cloud-prepare tool to automatically create z/OS configuration

You can automatically create a z/OS configuration (PARMLIB, PROCLIB, TCPPARMS, IPLPARM) by using the IBM-provided zOS-cloud-prepare tool. This tool runs in a running z/OS UNIX System Services environment and creates a new load suffix that contains a templates configuration so that the z/OS can run in a Wazi as a Service environment successfully.

Before you begin, check the installation directory of your web server. For example, /opt/ibm/wib is the default installation directory, but you can specify your own installation directory during the installation process.

The zOS-cloud-prepare tool is packaged with Wazi Image Builder at *<installation_directory>*/zSystem/zOS-cloud-prepare.s390x. Complete the following steps before you run this tool:

1. Copy the zOS-cloud-prepare file to the z/OS UNIX folder /usr/lpp/IBM/zdt in the source z/OS environment. You can use one of the following ways:
    - Use the Secure Copy Protocol (SCP) to transfer the file.
    - Use the Securing File Transfer Protocol (SFTP) to transfer the file with binary mode.
2. Add execution permission to the file with the following command:

    ```
    chmod 755 zOS-cloud-prepare.s390x
    ```

3. Execute the file on your source environment.

This tool creates a new load suffix that you defined (default value is *K2*) and creates three new data sets based on the new suffix.

- K2.PARMLIB
  Based on copies of an existing load suffix, where *XX* is either a discovered or a provided value:
    - IEASYMXX: Symbols that are used and can be updated during deployment.
    - MSTJCLXX: Adds a prefix to the *XX.PROCLIB* to the proclib search list.
- K2.PROCLIB
  Based on copies of existing proclibs from an existing load suffix:
    - JES2: Adds a prefix to the *XX.PROCLIB* to the proclib search list (static or dynamic).
    - RESOLVER: Sets the *XX.TCPPARMS(RESSETUP)* and adds execution step for *XX.PROCLIB(UPDRESOL)*.
    - TCPIP: Sets the *XX.TCPPARMS(PROFILE)* and *XX.TCPPARMS(TCPDATA)*.
    - UPDRESOL: REXX that updates symbols within the *XX.TCPPARMS(INODES)*.
- K2.TCPPARMS
  TCPIP templates are created based on the new load suffix (*K2*):
    - INODES: Host/domain resolution
    - PROFILE: z/OS network configuration
    - RESSETUP: Resolver configuration that points to *XX.TCPPARMS(INODES)* and *XX.TCPPARMS(TCPDATA)*
    - TCPDATA: z/OS hostname and domain configuration
- K2.MSGPARMS
    - DEVICES: Buffer for device notifications during attachment and detachment

Tip: To see the syntax of the tool, issue `zOS-cloud-prepare --help`.
Following is a list of commands in the zOS-cloud-prepare tool:

1. `-config=""`: Path to the config file.
2. `-dbs=100`: Devices buffer size.
3. `-debug=false`: Enable verbose logging.
4. `-destloadsuffix="K2"`: Destination IPL parm load suffix to be created.
5. `-dryrun=true` (optional): Skip committing changes to the system.
6. `-osaportname=""`: Specify the OSA port name if `zoautil` is unavailable for discovery.
7. `-srcloadsuffix="A0"`: Source IPL parm load suffix to be copied.

8. `-sysdsn="SYS0"`: System data set name that contains the load parameters (IPLPARM).
9. `-tmpdir="/tmp"`: Temporary directory to store artifacts. It is cleaned up if debug is not enabled.
10. `-zosutil="/usr/lpp/IBM/zoautil"`: Location of `zoautil` (used as OSA port name when not specified).
11. `-ttls=true`: Activate Application Transparent TLS (AT-TLS) in destination `TCPIP config`.
12. `-version=false`: Display the version information for the tool used.

Note:
The default value of the High Level Qualifier (HLQ) is *K2*. You can specify an HLQ other than K2 with the `"-destloadsuffix"` command line parameter.

The load parm member (LOADK2 in the example) is created from the existing load suffix that you specified with the `"-srcloadsuffix"` command line parameter.

The load parm member (LOADK2 in the example) is created under the DSN (SYS0.IPLPARM in the example) that you specified with the `"-sysdsn"` command line parameter.

To run the zOS-cloud-prepare tool, complete the following steps.

1. Run the tool in `-dryrun=true` mode, and save the output to a file. When this value is provided, no changes are committed to your system, but it prints to the console what happens if you run `-dryrun=false`. For example:

   ```
   ./zOS-cloud-prepare.s390x -dryrun=true -destloadsuffix=K2 -osaportname=CU331660 -
   srcloadsuffix=ZL -sysdsn=SYS0 > cloud-prepare.out 2>&1
   ```

2. Remove the translation tag from output file.

   ```
   chtag -r ./cloud-prepare.out
   ```

3. View the output file contents. If you see "Done" at the end, then it means that the process is completed. Otherwise, an error might have occurred and caused the tool to exit early. If this happens, then investigate and resolve the error, and run the tool in dryrun mode again, until you see the "Done" message. For example:

   ```
   ./zOS-cloud-prepare.s390x -dryrun=true -debug=true -destloadsuffix=K2 -osaportname=CU331660 -
   srcloadsuffix=ZL -sysdsn=SYS0 > cloud-prepare.out 2>&1
   ```

   Tip: When you debug errors, use the `"-debug=true"` command line parameter to get the tool to output more details.
4. When the tool runs to completion without error, then review the output file to see what data sets and data set members (and contents) the tool intends to create.
5. If everything looks good, run the tool again with the same parameters in step 1, and specify `"-dryrun=false"` this time. The tool will attempt to create the data sets and data set members. For example:

   ```
   ./zOS-cloud-prepare.s390x -dryrun=false -destloadsuffix=K2 -osaportname=CU331660 -
   srcloadsuffix=ZL -sysdsn=SYS0 > cloud-prepare.out 2>&1
   ```

   Note: Make sure that you are given the appropriate permissions to create the data sets (see the Prerequisites section). If you do not want the tool to make the updates, then use the data from the output file to manually create the data sets and data set members on the source z/OS system.
6. From ISPF, review the data set members that the tool creates, and make any adjustments that might be unique to the source z/OS system and necessary for the custom image to IPL successfully in the z/OS virtual server instance.

Also, review the following known limitations with the zOS-cloud-prepare tool, and make changes accordingly.

- The tool might not create the output data sets and/or data set members when the target (output) data set is in use (even with `DISP=SHR`). For example, if ISPF is already open and browsing `SYS0.IPLPARM`, the tool will not create the new load parm member that it intends to create, and an error will not be raised. It is always a best practice to review the data set and data set members that the tool creates before you proceed.
- The tool does not currently handle source z/OS environments where symbols are only loaded for specific `HWNAME` and `LPARNAME` values. As these values cannot be set properly in the Wazi as a Service environment in IBM Cloud, it means that these symbols will not get loaded and might cause IPL problems because the configuration is expecting values for them. The workaround is to either temporarily remove the `HWNAME/LPARNAME` filtering for needed symbols, or to copy the needed symbols that are protected by the filters to the `K2.PARMLIB(IEASYMK2)` member.
- The tool does not currently copy or modify the `CONSOLEXX` member. Debugging IPL issues on Wazi as a Service could be more difficult if the CONSOLE statement for `DEVNUM(SYSCON)` does not specify `ROUTCODE(ALL)` and `ALLOWCMD(Y)`. If the parameters for the `SYSCON CONSOLE` statement are not set on the source z/OS system as described here, then the suggested workaround is to manually copy the `CONSOLEXX` member to the `K2.PARMLIB` data set that the tool creates, and edit it accordingly.

After you run the zOS-cloud-prepare tool, you can start the extraction in Wazi Image Builder by creating the components that you want, and then create an image from the components.

Option 2: Using your existing z/OS configuration

You can also use your existing z/OS configuration with the following modifications.

- The TCP/IP profile is set up so that it relies on the following symbols that are defined in your system PARMLIB (IEASYMxx).
  - IP address: HOMEIPADDRESS1
  - Default route: DEFAULTROUTEADDR
- The resolver configuration is defined in a PDS (not a PDSE) and is set up so that it relies on the following symbols that are defined in your system PARMLIB (IEASYMxx).
  - Name server: NAMESERVERADDR1

If a new load parm is produced from these configuration changes, you can create the components that you want, and then create an image from the components.

Configuring image properties

After you created an image, you need to edit the image to match the configuration by completing the following steps:

1. In the Images (from source environment) table, find the image that you created.
2. Click the Edit icon  , and then select Edit advanced image properties.
3. Under IPL properties, change the LOADxx suffix to the suffix that you produced.
4. Validate that the SYSRES device number and IODF device number are correct. The numbers are used by Wazi as a Service to build the IPL statement.
5. Click Next to check the networking settings, and then click Save.

For more information, see Monitoring and managing the created images.

Creating components

A component is a collection of assets from an IBM® Z mainframe. A component can be reusable with other components from the same IBM Z mainframe in a mix and match fashion. You can use the created components to create images, and deploy the image as a custom image to your IBM Cloud® VPC.

Wazi Image Builder supports creating volume components only.

Notes:

- Restriction: TCP/IP profile data sets that are sequential or PDSE are not supported. Only PDS is supported now.
- For any extraction from z/OS, the z/OS system requirements must be met.

- **Creating a volume component**
  Learn how to create a volume component.
- **Creating a volume component from an existing z/OS instance that ZD&T runs on Linux**
  After you configure an existing z/OS instance that ZD&T runs on a host Linux® system, you can create a volume component from the source environment with faster speed.

Creating a volume component

Learn how to create a volume component.

If you create a volume component, grant READ access to the SAF profile in class DASDVOL for the volumes that are extracted to the user ID that is used in the Wazi Image Builder. For more information about DASD volume migration, see section 15.3.2 "RACF® requirements" in the zPDT® Guide and Reference.

To create a volume component, complete the following steps:

1. Select an IBM Z® mainframe that you want to connect from the SOURCE ENVIRONMENT list.
2. Click Create components on the home page, or click the Components tab on the header and click Create component.
3. Enter the credentials of the IBM Z mainframe if the source environment is not logged in.

Note: Contact your administrator if you do not have the information of the mainframe system.

4. From the Component type list, select Volume.
5. Enter a unique name that can be used to identify the volume component.
6. Optional: If you want system residence files to be contained in the component, select the Contains the system residence file(s) checkbox, and then optionally minimize the system residence volumes by using the automated system volume discovery tool.
   Note: The automated system volume discovery tool can identify the system residence volumes that are considered necessary to IPL the base z/OS® based on critical system data sets, such as the system link list, authorized program facility list, master catalog, and z/OS UNIX System Services file systems. To use this tool, make sure that you have completed all the steps in Installing ZD&T z/OS Extraction Utilities.
   a. Click Discover system volumes.
   b. In the Enable automated system volume discovery window:
      i. Optional: Expand Discovery settings.
      ii. Optional: If you want to exclude data sets with particular high-level qualifiers (HLQs) from the discovered system volume list, select these HLQs in the High-level qualifiers table.
      iii. Optional: Select the Override the current selection with the discovered values checkbox to override the volumes that are being selected if there is any.
      iv. Click Run.
      The system residence volumes identified by the automated system volume discovery tool are highlighted in the Volumes table with an additional Reasons for selection column. In this column, you can see the corresponding system residence volume is important to the IPL of which system data sets and use it as a reference for selecting volumes that you need.
7. Select all the volumes that are needed to create the component.
8. Optional: Enable the Adjust advanced settings option, and then enter an integer in the box of Concurrent extractions to define the number of artifacts that you want to extract simultaneously.
   Note: Concurrent extractions can extract multiple volumes in parallel, and improve the efficiency of creating components with multiple volumes. However, if the number is too large, the system might be overloaded, and the overall process might be slow.
9. Optional: Enter some notes that can help you identify the component.
10. Click Show *number* items selected to verify the volumes that are selected to create the component.
11. Click Create component directly, or click the Schedule extraction icon to specify the date and time that you want to create your component.

After the volume component is created:

- All extracted artifacts will be stored on the system that you configured for storage.
- The type of the volume component that contains system residence files is System residence volume. Otherwise, the type of the volume component is Volume.

Creating a volume component from an existing z/OS instance that ZD&T runs on Linux

After you configure an existing z/OS instance that ZD&T runs on a host Linux® system, you can create a volume component from the source environment with faster speed.

Before you create a volume component from such a source environment, ensure that the following two requirements are met.

- The z/OS instance on Linux is up and running.
- The latest z/OS® Extraction Utilities is installed in the source environment. For detailed instructions, see Installing ZD&T z/OS Extraction Utilities.

To create a volume component, complete the following steps.

1. Select a Linux running a ZD&T instance environment that you want to connect to from the drop-down menu in the SOURCE ENVIRONMENT pane.
2. Click Create components on the home page. Or, click the Components tab on the header, and click Create component.
3. Enter the credentials of the existing z/OS instance on Linux.
   Note: Contact your administrator if you do not have the information.
4. Enter a unique name that can be used to recognize the volume component.
5. Optional: Select if you want the system residence files to be contained in the component.
6. Optional: Enable the Adjust advanced settings option, and enter an integer in the box of Concurrent extractions to define the number of volumes that you want to copy simultaneously.

Note: Concurrent extractions can copy multiple volumes in parallel, and improve the efficiency of creating components with multiple volumes. However, if the number is too large, the system might be overloaded, and the overall process might be slow.

7. Optional: Enter some notes that can help you identify the component.
8. Select the volumes that are needed to create your component.
9. Click Show *number* items selected to verify the volumes that are selected to create the component.
10. Click Create component directly. Or, click Schedule extraction to specify the date and time that you want to create your component.
    When you create a component, all the volumes that are copied will be stored on the system that you configured for storage.
11. Enter the credentials to log in the Linux that runs the ZD&T emulator .

Monitoring and managing the created components

After you created a component, you can check the status of the extraction, and the details of your created components.

Editing created components

After you created a component, you can edit the contents of the component by completing the following steps:

1. Click Manage components on the home page, or click the Components tab on the header.
2. In the Components table, find the component that you want to update, and then click the Edit icon.
3. Change the component name if you want.
4. Optional: Enable the Adjust advanced settings option, and then enter an integer in the box of Concurrent extractions to define the number of artifacts that you want to extract simultaneously.
5. Modify the volumes or data sets that you selected, or reselect the volumes or data sets for the component. If you are updating a volume component that contains system residence files, you can optionally minimize the system residence volumes by using the automated system volume discovery tool.
   Note: The automated system volume discovery tool can identify the system residence volumes that are considered necessary to IPL the base z/OS® based on critical system data sets, such as the system link list, authorized program facility list, master catalog, and z/OS UNIX System Services file systems.
   a. Click Discover system volumes.
   b. In the Enable automated system volume discoverywindow, complete the following steps:
      i. Optional: Expand Discovery settings.
      ii. Optional: If you want to exclude data sets with particular high-level qualifiers (HLQs) from the discovered system volume list, select these HLQs in the High-level qualifiers table.
      iii. Optional: Select the Override the current selection with the discovered values checkbox to override the volumes that are being selected if there is any.
      iv. Click Run.
      The system residence volumes identified by the automated system volume discovery tool are highlighted in the Volumes table with an additional Reasons for selection column. In this column, you can see the corresponding system residence volume is important to the IPL of which system data sets and use it as a reference for selecting volumes that you need.
6. Click Update component, or click the Schedule extraction icon to specify the date and time when you want to update the component.

Note:

- A component that is being extracted, scheduled, or provisioned cannot be edited. However, you can create a new version or duplicate the component to edit and work with it.
- The component type cannot be edited.

Resuming a failed extraction of a component

If the extraction of a component failed, you can resume the component by completing the following steps:

1. Click the open and close list of options icon, and select Resume component.
2. Verify the details of the component, and click Yes, perform extraction.

Refreshing the extraction of a component

To refresh the extraction of a component, complete the following steps:

1. Click the open and close list of options icon, and select Refresh component.
2. Verify the details of the component, and click Yes, refresh.

Creating a new version of a component

When you create a component, the version number always starts at 1, and increases by 1 when you create a new version. If you want to add new contents or refresh contents but do not want to modify the existing component, you can create a new version by completing the following steps:

1. Click the open and close list of options icon, and select Create new version.
2. Log in to the specified IBM Z® mainframe if required.
3. Change the component name if you want.
4. Enable the Adjust advanced settings option, and enter an integer in the box of Concurrent extractions to define the number of artifacts that you want to extract simultaneously.
5. Add contents, or reselect new contents that you need for the new version.
6. Click Create component. Or, click Schedule extraction.

Note:

- When you create a new version of a component, you can add, remove, or reselect the volumes or data sets for the new version.
- The New version button is available only for the newest version of a component.
- The component type cannot be edited.

Duplicating a created component

If you want to copy a created component with a separate lifecycle that starts from version 1, complete the following steps:

1. Click the open and close list of options icon, and select Duplicate component.
2. Log in to the specified IBM Z mainframe if required.
3. Create a new component name.
4. Enable the Adjust advanced settings option, and enter an integer in the box of Concurrent extractions to define the number of artifacts that you want to extract simultaneously.
5. Duplicate the same contents, or reselect the volumes or add data sets that you need for the duplicated component.
6. Click Create component. Or, click Schedule extraction.

Deleting created components

To delete a component, complete the following steps:

1. Click the open and close list of options icon.
2. Select Delete component, and click Delete.

Note:

- Only the components that are created from IBM Z can be deleted separately.
- A component that is being created cannot be deleted.
- A component that is used in an image cannot be deleted. To delete a component, you must delete all of the images that use the components first.
- If you delete a component and image, the volume files that are located at the storage server will be deleted automatically.

Canceling a scheduled extraction

If you want to cancel the extraction of a component that is scheduled to be created, complete the following steps:

1. Click the open and close list of options icon, and select Cancel scheduled component extraction.
2. Click Yes, cancel scheduled extraction to confirm cancellation.

Stopping the extraction of a component

To stop a component that is being extracted, complete the following steps:

1. Click the open and close list of options icon, and select Stop extraction.

2. Click Yes, stop extraction to confirm it.

Migrating components to the current version

If a data set component that is created from a previous version contains mini volumes, the migration icon is displayed. You need to re-extract the contents of the component, otherwise the components cannot be provisioned.

To migrate a component, complete the following steps:

1. Click the Migrate icon, and select Perform extraction.
2. Enter the credentials of the IBM Z mainframe that you create component from.
3. Click Complete.

In a migration, the contents of a component might be changed if the original contents that came from the mainframe system changed.

Downloading log

To check the details of the created components, click the open and close list of options icon. Then, select Download log.

Note: Only the components that are created from IBM Z volumes support editing, creating new versions, or duplicating functions. For the components that are created from IBM Z CICS® regions and Db2® subsystems, these functions are not supported.

Creating z/OS images

After you create components from IBM Z® mainframe system or existing z/OS instances, you can create z/OS® images by combining the created components into a single package. Then, you can deploy the images to IBM Cloud VPC.

An image is a collection of components that are packaged together from your source environment for deploying to your target environment. When you create an image, you must select to include one component of type system residence volume and any number of other components that represent your application or application data from the source environment.

To create an image, complete the following steps:

1. Select an IBM Z mainframe or an existing z/OS instance that you want to connect from the drop-down menu in the SOURCE ENVIRONMENT pane.
2. Click Create images on the home page. Or, click the Images tab on the header, and click Create image.
3. Enter a unique name that can be used to recognize the z/OS image.
4. Optional: Enter some notes that can help you identify the image.
5. Select the components and versions that you want to include in the image. If the data sets included in the selected components are related to user catalogs that are not part of the image, a warning message is displayed. Update components to ensure that the necessary user catalogs are extracted and included.
   Note: An image must contain at least one and only one component that contains your system residence volumes and any number of other components that might represent your application, application data, or any number of subsystems such as CICS® or IMS.
6. Click Show *number* items selected to verify the components that are selected to create your image. Then, click Create image.

Configuring image properties

To deploy custom images to IBM Cloud®, you need to edit the image to match the configuration by completing the following steps:

1. In the Images (from source environment) table, find the image that you created.
2. Click the Edit icon  , and then select Edit advanced image properties.
3. Under IPL properties, change the LOADxx suffix to the suffix that you produced.
4. Validate that the SYSRES device number and IODF device number are correct. The numbers are used by Wazi as a Service to build the IPL statement.
5. Click Next to check the networking settings, and then click Save.

For more information, see Monitoring and managing the created images.

Monitoring and managing the created images

After you create an image, you can stop, resume, or delete the image. Also, you can retrieve logs to check the details of the image.

Modifying selected components for a created image

After you created an image, you can edit the components that you selected for the image on the Images page by completing the following steps:

1. Click the Edit icon  , and then select Edit image.
2. Optional: Change the image name if you want.
3. Modify the components that are selected, or reselect components for the image.
4. Click Update image.

Note:

- Only the images that were created from IBM Z® mainframe can be edited.
- An image that is being extracted, scheduled, or deployed cannot be edited. However, you can create a new version or duplicate the image to edit and work with it.

Editing image properties

After you created an image, you can edit the image properties on the Images page by completing the following steps:

1. Click the Edit icon  , and then select Edit image properties.
2. Modify the system configurations, and then click Next.
3. Modify the networking configurations, and then click Save.

You can also edit image properties when modifying selected components for a created image.
Note:

- You can edit the properties of only the images that are created from z/OS environments extracted from your on-premise IBM Z systems.
- An image that is being extracted, scheduled, or deployed cannot be edited. However, you can create a new version or duplicate the image to edit and work with it.

Duplicating a created image

To copy a created image with a separate lifecycle that starts from version 1, complete the following steps:

1. On the Images page, select Duplicate image from the action menu  .
2. Log in to the specified IBM Z mainframe if required.
3. Create a new image name.
4. Duplicate the same components, or reselect the components that you want for the duplicated image.
5. Click Create image.

Creating a new version of an image

When you create an image, the version number always starts at 1, and increases by 1 when you create a new version. If you want to add new components or refresh the components that you selected but do not want to modify the existing image, you can create a new version by completing the following steps:

1. On the Images page, select Create new version from the action menu  .
2. Log in to the specified IBM Z mainframe if required.
3. Change the image name if you want.
4. Add new components, or reselect the components that you need for the new version.

5. Click Create image. If a different system residence volume component is selected in step 5, the "Systems residence volume selection" dialog is displayed after you click Create image. Confirm whether you want to keep the image properties from the previous version or import the properties from the selected component, and then click OK.

Note:

- The Create new version button is available only for the newest version of an image.
- When you create a new version of an image, you can add, remove, or reselect the components for the new version.

Deleting a created image

To delete an image that was created, complete the following step:

1. On the Images page, select Delete image from the action menu  .
2. Click Yes, delete.

Note:

- An image that is being extracted, scheduled, or deployed cannot be deleted. To delete an image that was deployed, you must remove the image from all target environments first.
- If you delete an image that was created from IBM Z components, the components that are used in the image will not be deleted. However, if you delete an image that was created from an exiting ZD&T instance, the components that are used in the image will also be deleted.

Downloading log

To check the details of a created image, go to the Images page, and then select Download log from the action menu  .

Note:

- Only the images that were created from IBM Z components supports editing, creating new versions, or duplicating functions.
- When you edit an image, create a new version, or duplicate an image, you must select at least one component that contains system residence volume.

Deploying images to a Wazi as a Service target environment

After you create an image, you can deploy the image to a Wazi as a Service target environment in IBM Cloud® VPC.

To deploy a created image to a Wazi as a Service target environment, complete the following steps:

1. From the drop-down menu in the SOURCE ENVIRONMENT pane, select an environment that contains the custom images that you want to deploy.
2. From the drop-down menu in the TARGET WAZI AS A SERVICE pane, select a target environment.
3. Go to the Deployments page.
4. Click Deploy image.
5. Enter a name in the Custom image VPC name field to identify the deployment.
6. Select the custom image and image version that you want to deploy.
7. Specify the PARMLIB and IEASYM suffix for the member that contains the TCP/IP symbol configuration.
   Note: Make sure that you set up your on-premises environment so that the TCP/IP profile is properly configured in the deployment process. For more information, see Setting up networking for on-premises z/OS environment.
8. Click Deploy.

When the deployment is successfully completed, the image is automatically converted for use in IBM Cloud in the VPC region that you selected for the target environment.

The deployment process creates two artifacts:

- A custom qcow2 image that is named 'wazi-custom-image' by default
- A data volume snapshot that is named 'wazi-custom-image-data' by default and is up to 16 TB

When you create a virtual server instance in the IBM Cloud console, you need to select the custom image and create a snapshot from the data volume. For more information, see Creating a z/OS virtual server instance.

Monitoring and managing the deployments

After you deploy a custom image to a Wazi as a Service target environment, you can use the Deployments page to check the deployment status, delete, or resume the deployments.

Checking the status of the deployment process

To check the status of a deployment process, complete the following steps.

1. Click the drop-down menu of TARGET WAZI AS A SERVICE, and select the target environment that you deployed the image to.
2. Click Manage deployments from the home page, or click Deployments on the top tab.
3. Find the image that you deployed. A status icon is displayed for each deployment.
4. You can use the Filter option to filter deployments with different status, including Completed, In progress, and Needs attention.

Resuming a failed deployment

To resume a failed deployment of an image, complete the following steps.

1. Click the action icon beside the deployment, and select Resume deployment.
2. Provide the required credentials.
3. Click Deploy.

Deleting a deployment

To delete the deployment of an image, complete the following steps.

1. Click the action icon beside the deployment, and select Delete deployment.
2. Confirm the information in the pop up window. Check the box if you want to force the deletion.
   Note: A deployment that is still in progress cannot be deleted or resumed.
3. Click Delete.

Note: Removing deployment records from the Deployments page does not remove the custom image in IBM Cloud VPC. Likewise, removing images from the Wazi as a Service target environment in IBM Cloud VPC does not remove the corresponding deployment record on the Deployments page. For more information, see Managing z/OS virtual server instances.

Using REST APIs

Wazi Image Builder supports REST APIs usage. You can access all functions of Wazi Image Builder without the web user interface, including creating components and images, deploying images, and monitoring the deployments.

To get started with REST APIs, you need to find the specification of all REST APIs that are available in the Wazi Image Builder, and know the restrictions of REST APIs usage.

To find the complete specification of all the available REST APIs provided by Wazi Image Builder after you start your web server, go to the following URL:
https://*<your-tools-server-host-name>*:9443/api/explorer/
Note: This document only provides the restrictions and some examples on how to use the REST APIs.

Error response

For the error codes that are returned from the response body, refer to REST API error codes to get the corresponding error messages. The parameters that are returned from the response body are used to complete the arguments in the error messages.

Restrictions

When you use REST APIs, the following restrictions also exist.

- All REST APIs require the basic authentication.

- Any account that can authenticate with the web user interface can be used to call any of the REST APIs. Authorization to the REST APIs is determined based on the role of the user.
- The initial password of 'wibadmin' is 'password'. To change the password, complete the following steps:
  1. Find the encryption key that is specified by `wlp.password.encryption.key` in the /opt/ibm/wib/Liberty/usr/servers/wib-server/bootstrap.properties.
  2. Run the following command where you install the web server.

     ```
     /opt/ibm/wib/Liberty/bin/securityUtility encode --encoding=aes --key=
     <encryption_key_found_above> <new password value>
     ```

     After you run the command, an encoded string is created.

  3. Put the encoded string into the server.xml file that is at /opt/ibm/wib/Liberty/usr/servers/wib-server/server.xml. Then, replace the text in **bold** with the new encoded password value.

     ```
     <!-- user authentication -->
     <basicRegistry
     id="_home_markdown_jenkins_workspace_Transform_in_SSSTT9_1.0.0_com.ibm.zsys.rdt.tools.user.
     guide.doc_topics_api_usage_wib" realm="wib">
     <user name="wibadmin"
     password="{aes}AM1LZsnwLRNsVtYAiwqhVDO9/RL+NgYthDZXZhQgARtB"/>
     </basicRegistry>
     ```

- **Creating and managing Wazi as a Service target environments with REST APIs**
  You can call REST APIs to create and manage Wazi as a Service target environments.
- **Creating components with REST APIs**
  If you want to extract data from z/OS® Systems, you must create components before you create images.
- **Checking the status of the created components with REST APIs**
  After the components are created, you can call the REST API to check the status of the created components.
- **Editing created components with REST APIs**
  After you create volume components from z/OS systems, you can edit the component, or create a new version and duplicate for the component.
- **Creating images from z/OS volumes with REST APIs**
  Learn about the steps to create images from z/OS volumes.
- **Checking the status of the created images with REST APIs**
  After the images are created, you can call the REST APIs to check the status of the created images.
- **Editing the created images with REST APIs**
  After you create images, you can edit the image, or create a new version or a duplicate of the image.
- **Deploying images to a Wazi as a Service target environment with REST APIs**
  You can deploy an image from an emulated IBM Z® hardware to a specific Wazi as a Service target environment by using the UUID.
- **REST API error codes**
  If an error occurs when you use REST APIs, refer to the response error codes for detailed description.

Creating and managing Wazi as a Service target environments with REST APIs

You can call REST APIs to create and manage Wazi as a Service target environments.

About this task

You can create a Wazi as a Service target environment with a UUID contained in the response by using the REST API. This UUID represents the Wazi as a Service target environment that you create. You can manage the target environment by using the UUID.

To create or manage Wazi as a Service target environments by calling REST APIs, you need to use your own setup variables to replace the *italic* text in the examples. For more information about the descriptions and available values of each field, see the REST API specification at the following URL:

https://*<your-tools-server-host-name>*:9443/api/explorer/

Procedure

1. Use the following REST API to create a Wazi as a Service target environment. After a Wazi as a Service target environment is created, the response contains a UUID that represents the target environment.

```
curl -X POST -k -u wibadmin:password
'https://your_tools_server_host_name:9443/wibrest/zdtrs/target-environment-services/v1/waziaas'
-H "accept: application/json;charset=utf-8"
-D '{
   "url": "cloud.ibm.com",
   "resource-group-id": "your_resource_group_id",
   "cos-bucket-name": "your_bucket_name",
   "cos-region-id": "global",
   "cos-id": "your_cos_id_crn:v1:bluemix:public:cloud-object-
storage:global:a/4094bc65badadaasd016f4651f0788bc:1959c2a4-1212-4edd-1212-2ad591b37c46::",
   "api-token": "your_API_token",
   "label": "sampleLabel",
   "description": "sampleDescription",
   "concurrent-transfers": 4,
   "vpc-region": "ca-tor",
   "vpc-region-endpoint": "ca-tor.iaas.cloud.ibm.com",
   "public-domain": true
   }'
```

2. After a Wazi as a Service target environment is created, you can use the following REST APIs to get, update, or delete the target environment.

   Getting a Wazi as a Service target environment

   UUID is required to get a Wazi as a Service target environment. To get a target environment, use the following REST API.

   ```
   curl -X GET -k -u wibadmin:password
   'https://your_tools_server_host_name:9443/wibrest/zdtrs/target-environment-
   services/v1/waziaas/your-target-system-uuid'
   -H 'accept: application/json;charset=utf-8'
   ```

   Updating a Wazi as a Service target environment

   UUID is required to update a Wazi as a Service target environment. To update a target environment, use the following REST API.

   ```
   curl -X PUT -k -u wibadmin:password
   'https://<hostname>:9443/wibrest/zdtrs/target-environment-services/v1/waziaas' \
   -H 'accept: application/json;charset=utf-8'  \
   -H 'Content-Type: application/json;charset=utf-8'  \
   -D {
     "label": "updatedLabel",
     "uuid": "your_target_system_uuid"
   }'
   ```

   Deleting a Wazi as a Service target environment

   UUID is required to delete a Wazi as a Service target environment. To delete a target environment, use the following REST API. After a target environment is deleted, the response displays 200 `(OK)` status code.

   ```
   curl -X DELETE -k -u wibadmin:password
   'https://<hostname>:9443/wibrest/zdtrs/target-environment-services/v1/waziaas/<UUID>' \
   -H 'accept: application/json;charset=utf-8'  \
   -H 'Content-Type: application/json;charset=utf-8'
   ```

Creating components with REST APIs

If you want to extract data from z/OS® Systems, you must create components before you create images.

- **Creating volume components from z/OS system with REST APIs**
  Learn about the steps to create components from IBM Z® volumes by using REST API.
- **Creating volume components from existing ZD&T or instances**
  Learn about the steps to create volume components from existing ZD&T or instances.

Creating volume components from z/OS system with REST APIs

Learn about the steps to create components from IBM Z® volumes by using REST API.

Procedure

1. Call the REST API to authenticate the SSH connection to the z/OS® system.

```
curl -k -u wibadmin:password -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{ "system-uuid": "systemuuid", "username": "zosuser", "password": "zospw"}' 'https://your_tools_server_host_name:9443/wibrest/zdtrs/auth-services/v3/authenticate-ssh'
```

2. Call the REST APIs to get the information on all volumes for the z/OS system. Put the token from above call into the authorization header.

```
curl -k -u wibadmin:password -X GET --header 'Accept: application/json' --header 'SSHAuthorization: {"token":"VVNFUjM1NTE4MTkwMzg4MzI=","hostname":"YOUR_Z_OS_HOST_NAME","port":22}' 'https://your_tools_server_host_name:9443/wibrest/zdtrs/z-services/listZosVolumes?hostName=YOUR_Z_OS_HOST_NAME&sshPort=22'
```

   The following response is displayed.

```
[{"id":0,"name":"A27AAA","extractionTime":"1960-01-01 23:03:20","deviceNumber":"048C","deviceType":"3390","progress":0,"compressed":0,"size":1892010960},{"id":0,"name":"A27CCC","extractionTime":"1960-01-01 23:03:20","deviceNumber":"04BE","deviceType":"3390","progress":0,"compressed":0,"size":2838016440},{"id":0,"name":"A45SYS","extractionTime":"1960-01-01 23:03:20","deviceNumber":"0492","deviceType":"3390","progress":0,"compressed":0,"size":8514049320},…]
```

3. Create the component by including your selected volumes that are returned from above call.

```
curl -k -u wibadmin:password -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' --header 'SSHAuthorization: {"token":"VVNFUjM1NTE4MTkwMzg4MzI=","hostname":"YOUR_Z_OS_HOST_NAME","port":22} ' -d '{ "name": "testComp-fromLinux", "version":1, "zSystem": { "hostname": "YOUR_Z_OS_HOST_NAME", "port": 22 }, "systemType": "COMP_SYSTEM_TYPE_ZSYSTEM_ENV", "description": "comp_desc", "volumes": [ { "name": "A27AAA" } , { "name": "MVS220" } ] , "schedules":{"originalScheduleTime": "2018-01-06 13:01:00.0","originalScheduleTimezoneId": "CTT"}, "concurrentExtractionNumber": 3, "ftpSystem": {"hostname": "your.image.storage.com"} }' 'https://your_tools_server_host_name:9443/wibrest/zdtrs/componentServices/v2/zosVolumeComponent'
```

Creating volume components from existing ZD&T or instances

Learn about the steps to create volume components from existing ZD&T or instances.

Procedure

1. Call the REST API to authenticate the SSH connection to the z/OS® system.

```
curl -k -u wibadmin:password -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{ "system-uuid": "systemuuid", "username": "zosuser", "password": "zospw"}' 'https://your_tools_server_host_name:9443/wibrest/zdtrs/auth-services/v3/authenticate-ssh'
```

2. Call the REST APIs to get the information on all volumes for a specific z/OS system. Put the token from above call into the authorization header.

```
curl -k -u wibadmin:password -X GET --header 'Accept: application/json' --header 'SSHAuthorization: {"token":"VVNFUjM1NTE4MTkwMzg4MzI=","hostname":"YOUR_Z_OS_HOST_NAME","port":22} ' 'https://your_tools_server_host_name:9443/wibrest/zdtrs/zdt-services/volumes?uuid=systemuuid'
```

   The following response is displayed.

```
{"zosAllVolumes": [{"id":0,"name":"A27AAA","extractionTime":"1960-01-01

23:03:20","deviceNumber":"048C","deviceType":"3390","progress":0,"compressed":0,"size":1892010960},{"id":0,"name":"A27CCC","extractionTime":"1960-01-01

23:03:20","deviceNumber":"04BE","deviceType":"3390","progress":0,"compressed":0,"size":2838016440},{"id":0,"name":"A45SYS","extractionTime":"1960-01-01
```

```
23:03:20","deviceNumber":"0492","deviceType":"3390","progress":0,"compressed":0,"size":851404932
0},…] ,
    ”zosInfo“:{ "sysinfo":{"loadsuf":"AU",...}, "disks":[...],...}
    }
```

3. Call the REST API to authenticate the SSH connection to the Linux® system that is running the ZD&T or instance.

```
curl -k -u wibadmin:password -X POST --header 'Content-Type: application/json' --header 'Accept:
application/json' -d '{ "hostname": "YOUR_LINUX_HOST_NAME", "port": 22,"username": "LinuxUSER",
"password": "LinuxPW"}' 'https://your_tools_server_host_name:9443/wibrest/zdtrs/auth-
services/v2/authenticateSSH'
```

4. Create the component by providing the authentication to the Linux system. The authentication needs to include your selected volumes, and the whole "zosInfo" structure in the "zsystemInformation" section that is returned from step 2 GET VOLUME call.

```
curl -k -u wibadmin:password -X POST --header 'Content-Type: application/json' --header 'Accept:
application/json' --header 'SSHAuthorization:
{"token":"VVNFUjM1NTE4MTkwMzg4MzI=","hostname":"YOUR_LINUX_HOST_NAME","port":22} ' -d '{ "name":
"testComp-fromLinux",   "zSystem": { "hostname": "YOUR_Z_OS_HOST_NAME",  "port": 2022   },
"systemType": "COMP_SYSTEM_TYPE_ZDT_ENV",   "description": "comp_desc",   "volumes": [   {
"name": "A27AAA" } ,     { "name": "MVS220" }    ] , "schedules":{"originalScheduleTime": "2018-
01-06 13:01:00.0","originalScheduleTimezoneId": "CTT"}, "concurrentExtractionNumber": 3,
"ftpSystem": {"hostname": "your.image.storage.com", ”zsystemInformation“:{ "sysinfo":
{"loadsuf":"AU",...}, "disks":[...],...} } }'
'https://your_tools_server_host_name:9443/wibrest/zdtrs/componentServices/zdt-volume'
```

Checking the status of the created components with REST APIs

After the components are created, you can call the REST API to check the status of the created components.

Procedure

Run the following command, and use the variables of your z/OS instances to replace the following *italic* text.

```
curl -k -u wibadmin:password -X GET --header 'Accept: application/json'
'https://your_tools_server_host_name:9443/wibrest/zdtrs/componentServices/components/YOUR_Z_OS_HOST_N
AME?sourcePort=YOUR_Z_OS_HOST_PORT'
```

Check the field in the component level. If the message `"status": "COMP_SCHEDULED"` is displayed, it means that the component is still being extracted.

When the message `"status":"COMP_EXTRACTED"` is displayed, it means that the component extraction is done, and the component is ready to be used.

Editing created components with REST APIs

After you create volume components from z/OS® systems, you can edit the component, or create a new version and duplicate for the component.

Procedure

1. Call the REST API to authenticate the ssh connection to the z/OS system.

```
curl -k -u wibadmin:password -X POST --header 'Content-Type: application/json' --header 'Accept:
application/json' -d '{ "system-uuid": "systemuuid", "username": "zosuser",  "password":
"zospw"}' 'https://your_tools_server_host_name:9443/wibrest/zdtrs/auth-services/v3/authenticate-
ssh'
```

   The following response is displayed.

```
{"expiration":1516202935577,"username":"ZOSUSER","password":"ZOSPW","token":"VVNFUjM1NTE4ODczMjY
3NjU=","port":22}
```

2. Run the following command to submit the request.

```
curl -k -u wibadmin:password -X PUT --header 'Content-Type: application/json' --header 'Accept:
application/json' --header 'SSHAuthorization:
{"token":"dXN1cjM1NTg5OTg4OTgwNA==","hostname":"YOUR_Z_OS_HOST_NAME","port":22} ' -d ' {"name":
"name of the component","zSystem": {"hostname": "myzos.domain.com","port": 22},"version":
1,"newInfo": {"name": "new name","description": "new comment","systemType":
"COMP_SYSTEM_TYPE_ZSYSTEM_ENV","concurrentExtractionNumber": 3},"addItems": ["volume to be
added"],"removeItems": ["volume to be removed"],"refreshItems": ["volume to be
refreshed"],"duplicate": false,"newVersion": false,"schedules": {"originalScheduleTime": "2018-
01-06 13:01:00.0","originalScheduleTimezoneId": "CTT"}}'
'https://your_tools_server_host_name:9443/wibrest/zdtrs/componentServices/component'
```

Creating images from z/OS volumes with REST APIs

Learn about the steps to create images from z/OS® volumes.

Procedure

1. Create the images by including some of the components that you have created earlier.

   ```
   curl -k -u wibadmin:password -X POST --header 'Content-Type: application/json' --header 'Accept:
   application/json' -d '{"name": "testApp1", "zSystem": {"hostname": "YOUR_Z_OS_HOST_NAME","port":
   22}, "version": 1, "description": "comment", "components": [{"name": "name of component",
   "zSystem": { "hostname": "myzos.domain.com", "port": 22 }, "version": 1 } ], "sysResComponent":
   { "name": " name of the system residence volume component ", "zSystem": { "hostname":
   "myzos.domain.com", "port": 22 }, "version": 1 }} '
   'https://your_tools_server_host_name:9443/wibrest/zdtrs/appServices/v2/zosApplicationImage'
   ```

   If an error code is displayed, for example:

   ```
   {
     "code": 31326,
     "message": "These data sets cannot be extracted because their containing volumes cannot be
   found: ["SYS1.A39AAA.LPALIB , SYS1.AFOMMOD1 "]",
     "level": {
       "name": "SEVERE"
     },
     "cause": "",
     "resolution": ""
   }
   ```

   The error code '31326' means that "these data sets cannot be extracted because their containing volumes cannot be found: [dataset_names]". For more information about error code, see REST API error codes.

2. Optional: Try again when an error code is displayed at step 1.

   ```
   curl -k -u wibadmin:password -X POST --header 'Content-Type: application/json' --header 'Accept:
   application/json' -d '{"name": "testApp1-fromLinux", "zSystem": {"hostname":
   "YOUR_Z_OS_HOST_NAME", "port": 22},   "ftpSystem": {"hostname":
   "your_storage_server_host_name"},   "components":[ {"name": "testComp-fromLinux"} ],
   "schedules": { "originalScheduleTime": "2018-01-06 13:01:00.0", "originalScheduleTimezoneId":
   "CTT" }, "concurrentExtractionNumber": 3 }'
   'https://your_tools_server_host_name:9443/wibrest/zdtrs/appServices/zosApplicationImage'
   ```

   Then, the generated information of images is returned.

Checking the status of the created images with REST APIs

After the images are created, you can call the REST APIs to check the status of the created images.

Procedure

Run the following command, and use the variables of your z/OS instances to replace the following *italic* text.

```
curl -k -u wibadmin:password -X GET --header 'Accept: application/json'
'https://your_tools_server_host_name:9443/wibrest/zdtrs/appServices/application/YOUR_Z_OS_HOST_NAME?
applname=testApp1-fromLinux'
```

Check the field in the application level. If the message `"status": "APPL_IN_PROGRESS"` is displayed, it means that the image is still being extracted.

When the message `"status":"APPL_AVAILABLE"` is displayed, it means that the application extraction is done, and the image is ready to be used.

Editing the created images with REST APIs

After you create images, you can edit the image, or create a new version or a duplicate of the image.

Procedure

1. To edit an image that is not deployed, run the following command.

```
curl -k -u wibadmin:password -X PUT --header 'Content-Type: application/json' --header 'Accept:
application/json' -d '{"version":3,"zSystem":
{"hostname":"myzos.domain.com","port":1022},"name":"zdt-
dev1.wholeimage.with.LOAD$1","zsystemInformationUpdateRequest":{"domain":0,"ccp":0},"newInfo":
{"name":"zdt-dev1.wholeimage.without.LOAD$1"},"addItems":[{"name":"zdt-dev1.1vol.s25","zSystem":
{"hostname":"myzos.domain.com","port":1022},"version":1}],"removeItems":[{"name":"zdt-
dev1.1vol.LOAD$1","zSystem":
{"hostname":"myzos.domain.com","port":1022},"version":1}],"duplicate":false,"newVersion":false,"
sysResComponent":{"name":"-dev1.wholeimage.with.volinit","zSystem":
{"hostname":"myzos.domain.com","port":1022},"version":3}}'
https://your_tools_server_host_name:9443/wibrest/zdtrs/appServices/application
```

2. To create a new version of an image, run the following command.

```
curl -k -u wibadmin:password -X PUT --header 'Content-Type: application/json' --header 'Accept:
application/json' -d '{"version":3,"zSystem":
{"hostname":"myzos.domain.com","port":1022},"name":"zdt-
dev1.wholeimage.with.LOAD$1","zsystemInformationUpdateRequest":{"domain":0,"ccp":0},"newInfo":
{},"addItems":[{"name":"zdt-dev1.1vol.s25","zSystem":
{"hostname":"myzos.domain.com","port":1022},"version":1}],"removeItems":[{"name":"zdt-
dev1.1vol.LOAD$1","zSystem":
{"hostname":"myzos.domain.com","port":1022},"version":1}],"duplicate":false,"newVersion":true,"s
ysResComponent":{"name":"zdt-dev1.wholeimage.with.volinit","zSystem":
{"hostname":"myzos.domain.com","port":1022},"version":3}}'
https://your_tools_server_host_name:9443/wibrest/zdtrs/appServices/application
```

3. To duplicate an image to a new image, run the following command.

```
curl -k -u wibadmin:password -X PUT --header 'Content-Type: application/json' --header 'Accept:
application/json' -d '{"version":3,"zSystem":
{"hostname":"myzos.domain.com","port":1022},"name":"zdt-
dev1.wholeimage.with.LOAD$1","zsystemInformationUpdateRequest":{"domain":0,"ccp":0},"newInfo":
{"name":"zdt-dev1.wholeimage.with.LOAD$1.duplicated"},"addItems":[{"name":"zdt-
dev1.1vol.s25","zSystem":
{"hostname":"myzos.domain.com","port":1022},"version":1}],"removeItems":[{"name":"zdt-
dev1.1vol.LOAD$1","zSystem":
{"hostname":"myzos.domain.com","port":1022},"version":1}],"duplicate":true,"newVersion":false,"s
ysResComponent":{"name":"-dev1.wholeimage.with.volinit","zSystem":
{"hostname":"myzos.domain.com","port":1022},"version":3}}'
https://your_tools_server_host_name:9443/wibrest/zdtrs/appServices/application
```

Deploying images to a Wazi as a Service target environment with REST APIs

You can deploy an image from an emulated IBM Z® hardware to a specific Wazi as a Service target environment by using the UUID.

About this task

After a Wazi as a Service target environment is created by using REST APIs, the response contains a UUID that represents the Wazi as a Service target environment. You can use this UUID to deploy images to the target environment.

Procedure

1. Create a deployment to a Wazi as a Service target environment by using the following REST API. In this example, optional fields are added to specify the IPL parameters or to specify the z/OS® user credential that is required to deploy a created image.

```
curl -X POST -k -u wibadmin:password
'https://your_tools_server_host_name:9443/wibrest/zdtrs/deploy-services/v1/waziaas/zdt-image'
-H 'accept: application/json;charset=utf-8'
-H 'Content-Type: application/json;charset=utf-8'
-d '{
   "general": {
   "label": "sampleLabel",
   "description": "sampleDescription",
   "target-uuid": "your_target_system_UUID",
   "resume": false,
   "update": false,
   "image-uuid": "your_image_UUID"}
}'
```

2. After you create a deployment, you can use the following REST API to monitor the progress.

```
curl -N --http2 -H 'accept:text/event-stream" -k -u wibadmin:password
'https://your_tools_server_host_name:9443/wibrest/zdtrs/monitor-services/v1/provision-
status/{your_provision_uuid}'
```

   If the response is `"status": "SUCCEEDED"`, the deployment is completed. To resume a failed deployment, use the following REST API:

```
curl -k -u wibadmin:password -X POST
-H 'accept: application/json;charset=utf-8'
-H 'Content-Type: application/json;charset=utf-8'
-d '{
   "emulator": {"cp":1,"ram":3,"ziip":0},"ipl":{"device-
address":"a80","iodfaddress":"0a82","load-suffix":"au"},
   "general": {"label":"docker1","provision-uuid":"11111111-aaaa-bbbb-cccc-
222222222222","resume":true},
   "zoscreds": {"username":"zosUsername","password":"zosPassword","private-key":"privateKey"}'
'https://your_tools_server_host_name:9443/wibrest/zdtrs/deploy-services/v1/waziaas/zdtimage'
```

3. If you need to delete deployment from a Wazi as a Service target environment, use the following REST API. After the deletion is completed, the response displays 200 `(OK)` status code.

```
curl -X DELETE -k -u wibadmin:password
'https://your_tools_server_host_name:9443/wibrest/zdtrs/deploy-services/v1/waziaas/zdt-
image/your_provision_uuid?force=false&resume=false&preserve=false'
-H 'accept: application/json;charset=utf-8'
```

REST API error codes

If an error occurs when you use REST APIs, refer to the response error codes for detailed description.

Note: The italic text such as *{0}*, *{1}*, *{2}* indicates that they are variables. The variables will be replaced by the contents that are displayed in the messages.

Table 1. Response error codes

Error ID	Description
403	This account is locked out for *{0}* minutes.
10001	The source system already exists.
10002	The source system does not exist.
10003	The target environment already exists.
10004	The target environment does not exist.
10005	The storage server already exists.
10006	The storage server does not exist.
10007	The Licensing system *{0}* already exists.
10008	The Licensing system *{0}* does not exist.
10009	The Token system already exists.

Error ID	Description
10010	The Token system does not exist.
10011	The IBM® Application Discovery system already exists.
10012	The IBM Application Discovery system does not exist.
10013	The application image already exists.
10014	The application image does not exist.
10015	The component already exists.
10016	The component does not exist.
10017	The volume already exists.
10018	The volume does not exist.
10019	The cloud platform already exists.
10020	The cloud platform does not exist.
10027	Target environment hostname *{0}* is not reachable.
10028	Create entity *{0}* failed.
10029	Retrieve entity *{0}* failed.
10030	Update entity *{0}* failed.
10031	Delete entity *{0}* failed.
10101	Failed to connect to the remote host *{0}* on port *{1}*.
10102	Token expired.
10200	Failed to connect to the storage server *{0}* on port *{1}*.
10201	Connection refused.
10202	Invalid user ID or password.
10203	Invalid base directory *{0}* specified for server *{1}* on port *{2}*.
10204	Invalid hostname for server *{0}*.
10205	Connection closed without indication.
10206	Failed to set the file transfer type.
10207	A failure occurred when storing content on the storage server.
10208	A failure occurred when retrieving content from the storage server.
10209	Failed to create directory *{0}* on the storage server.
10210	Unable to access the directory *{0}* on the storage server.
10211	Failed to delete the file *{0}* from the image storage server.
10213	Internal Error.
10214	The old base directory is needed.
10215	This feature is supported only for SFTP storage system.
10216	This source system cannot be deleted because there exists components or applications related to it.
10300	Failed to connect to the IBM Application Discovery server *{0}* on port *{1}*.
11001	Installation files download failed.
11003	Deploy request parameter not passed.
11007	Invalid Linux® platform of target system.
11011	zPDT® installation script error: Parameters invalid.
11012	zPDT installation script error: Primary licensing server is required.
11013	zPDT installation script error: zPDT package name is required.
11014	zPDT installation script error: zPDT installation failed.
11015	zPDT installation script error: Init gen2 failed.
11016	zPDT installation script error: Client Config failed.
11020	Deploy script error: Parameters invalid.
11021	Deploy script error: Devmap required.
11022	Deploy script error: Devmap does not exist.
11023	Deploy script error: Parameter loadparam required.
11024	Deploy script error: Parameter IODF address required.
11025	Deploy script error: Emulator startup message not found.
11026	Deploy script error: Parameter device number required.
11027	Deploy script error: Z is not on.
11028	Application images cannot be deployed to the same system that is running the web server.
11029	Target server is not available for deployment.

Error ID	Description
11030	Failed to create the directory *{0}* on the target system.
11033	IBM Z Development and Test Environment did not start successfully.
11034	The IPL of z/OS® running on IBM Z Development and Test Environment was not successful.
11035	Internal script error when installing with missing path.
11036	Patch installation failed when installing .
11037	The file failed to be transferred to the target environment during deployment.
11100	Some volume image files failed to be deployed.
11101	Changing the owner of volume image files failed.
11102	Executing command *{0}* on target environment *{1}* failed
11103	Generating devmap failed.
11104	Changing owner of devmap failed.
11105	A failure occurred when trying to start the emulator.
11106	A failure occurred when trying to stop the emulator.
11107	A failure occurred when trying to IPL with the given parameters.
11108	A failure occurred when trying to clean the target system with the given parameters.
11109	A failure occurred when trying to start an emulator on an undeployed target system.
11110	A failure occurred when trying to remove an image on the target system because the supplied credentials do not match the credentials used to start the deployment.
11111	A failure occurred when trying to obtain licensing information from the target system: *{0}*.
11112	Updating the emulator is not authorized.
11113	Downloading the z-volume files failed.
11114	Unzipping the z-volume files failed.
11115	User has no root or sudo access, please change the target system settings to bypass the emulator installation.
11116	User has no root or sudo access, please change the target system settings to bypass the dependencies installation.
11117	User has no root or sudo access, please change the target system settings to bypass the network configuration.
11118	User has no root or sudo access, please change the target system settings to bypass the user configuration.
11119	User has no root or sudo access, please change the target system settings to bypass the user configuration.
12000	A failure occurred when reading the devmap file: *{0}*.
12001	A failure occurred when parsing the devmap file.
12002	A failure occurred when writing the devmap file.
12003	Create application failed.
12004	The devmap file does not exist in the specified location: *{0}*.
12005	You do not have the permissions to access the devmap.
12006	You do not have the permissions to access the volume file.
12007	The IPL command for importing a image was not provided.
12008	The compression directory does not exist in the specified location: *{0}*.
12009	The user *{0}* does not have write access to the compression directory *{1}*.
12010	Should not specify the compression directory *{0}* because the compression option is off.
20001	Authorization token has expired.
20002	Authorization token was not supplied.
20003	User already exists.
20004	User does not exist.
21001	The machine running the web server experienced a network error.
21002	The ssh command on *{0}* failed to be executed.
21003	The sftp command on *{0}* failed to be executed.
21004	Remote command execution at *{0}* has been running for too long.
21005	Sending/receiving files to/from *{0}* failed when *{1}*.
21006	The target environment TCP port routing rules must contain 2022>22.
21007	The target environment port routing rules have syntax error.
21008	The target environment internal IP address has syntax error.
30000	Source system hostname is required.
30001	Source system hostname is invalid.
30002	Application name is required.
30003	Application name is invalid.

Error ID	Description
30004	Authorization is required.
30005	Authorization is invalid.
30006	Application is required.
30007	Application is invalid.
30008	The component schedule date is invalid.
30009	The component schedule time is invalid.
30010	The component schedule date is required.
30011	The component schedule time is required.
30012	The component schedule time zone is invalid.
30013	The component schedule time zone is required.
30014	Only one component that contains the system residence volume can be selected.
30015	The extraction cannot be resumed as the application image is already extracted.
30016	Only an application created from an IBM Z® mainframe supports extraction being resumed or stopped.
30017	The number of simultaneous volume extractions cannot be less than 1.
30018	The extraction cannot be stopped as the application is not being extracted.
30019	The z/OS user does not match the user who created the application.
30020	Component name is required.
30021	Component name is invalid.
30022	Volumes are required.
30023	Volumes are invalid.
30024	A data set name filter string is required.
30025	The data set name filter string cannot be * or **.
30026	The selected data set list is required.
30027	Only z/OS components can be deleted.
30028	This application component is currently being deleted.
30029	This application component cannot be deleted as it is being used by the application image *{0}*.
30030	This application component is currently being extracted.
30031	The data set name filter string contains invalid characters.
30032	The application component type cannot be changed from *{0}* to *{1}*.
30033	Only application components containing volumes can have their type changed.
30034	The version of the component is invalid.
30035	Only components created from an IBM Z mainframe support editing.
30036	The same volume or data set cannot be processed in multiple requests.
30037	The extraction cannot be stopped as the component is not being extracted.
30038	The extraction cannot be resumed as the component is already extracted.
30039	Creating a new version or duplication cannot occur at the same time.
30040	Deploy preference is required.
30041	Deploy preference is invalid.
30042	The image storage information is required to be defined on the *<a href*
30043	Component is required.
30044	The emulator licensing information is required to be defined on the *<a href*
30045	Target environment is required.
30046	Number of *<abbr title*.
30047	Memory size is invalid.
30048	Deployment label is required.
30049	Not enough *<abbr title*
30050	Not enough memory to start the emulator.
30051	Type of storage server is required (for example, SFTP).
30052	The cloud platform label is required.
30053	The user name who created the cloud platform configuration is required.
30054	The user name to log in the cloud is required.
30055	The password of the user to log in the cloud is required.
30056	The scope of the cloud is required.
30057	The deployment label is already existed.

Error ID	Description
30058	The deployment label is not found.
30059	The scope of the cloud is invalid.
30060	The request to import an existing image failed due to missing required input.
30061	The request to validation an existing image failed due to missing required input.
30063	The compression directory is required.
30064	The comment is longer than the allotted maximum of 225 characters.
30065	The component's system type is invalid.
30066	You cannot create *{0}* component on the source system of type *{1}*.
30067	Valid z/OS system information is required.
30068	Please login *{0}* instead of *{1}*.
30080	Target environment is required.
30081	Target environment is invalid.
30082	A source system is required.
30083	The source system is invalid.
30084	A storage server is required.
30085	The storage server is invalid.
30086	The IBM Application Discovery system is required.
30087	The IBM Application Discovery system information is invalid or empty.
30088	The port is invalid.
30089	A cloud platform is required.
30090	The endpoint URL of the cloud platform is required.
30091	The scope of the cloud platform is required.
30092	The domain name URL of the cloud platform is required.
30093	The project name URL of the cloud platform is required.
30094	The type of the cloud platform is required.
30095	The username of the cloud platform is required.
30096	The password of the cloud platform is required.
30097	No catalog found on the cloud platform.
30098	Authentication of the cloud platform has failed.
30099	Query on the cloud platform has failed.
30100	Credential is required.
30101	Credential is invalid.
30102	Token is required.
30103	Token is invalid.
30104	Target environment hostname is required.
30105	Target environment hostname is invalid.
30106	SSH port is required.
30107	SSH port is invalid.
30108	A User ID is required.
30109	Password is required.
30110	Your password has expired.
30111	Target environment label is required.
30112	Target environment label is invalid.
30113	A private key file is required.
30114	Saving private key for user *{0}* failed.
30115	The file size cannot exceed 50KB.
30116	All IPL parameters need to be entered, this includes the device address, IODF address, and device suffix.
30120	MVSDSALLOCATE is required.
30121	MVSDSALLOCATE is invalid.
30122	Parameter JSON string is required.
30123	Parameter JSON string is invalid.
30124	Storage server hostname is required.
30125	Storage server hostname is invalid.
30140	User is required.

Error ID	Description
30141	User is invalid.
30142	Username is required.
30143	Username is invalid.
30144	ADCD name is required.
30145	ADCD name already exists.
30146	User role is required.
30147	User role is invalid.
30148	User role does not exist.
30149	The version of the application is invalid.
30150	The number of crypto co-processors must be between 1 to 16.
30151	The domain that is defined must be a positive number.
30152	The UUID parameter for this API is required.
31000	This application image is being processed, please wait for a while.
31001	This application image is being extracted, please confirm to delete it.
31002	This application image is being deployed to *{0}*.
31003	This application image is being deprovisioned from *{0}*, please wait for it to finish.
31004	This application image is being deployed to *{0}* but failed, please confirm to delete it.
31005	This selected application component *{0}* is not in this application.
31006	This selected application component *{0}* is found in other applications as shown below: *{1}*.
31007	This application image is being deleted.
31008	This application image is not available.
31009	You must choose at least one component that contains the system residence volume.
31010	Only data set application components from another IBM Z mainframe can be added to this application.
31011	The data set application component *{0}* cannot be added to this application image as it has not been extracted.
31012	This application image cannot be deleted as it deployed to the following target environments: *{0}*.
31013	Only an ADCD or Z system application needs to be migrated.
31014	This *{0}* does not need migration.
31015	This *{0}* needs migration before you can deploy it.
31016	The contents of an application image cannot be edited as the image has been deployed to a target environment.
31017	One component cannot be processed in multiple requests.
31018	The name of an application image cannot be changed when creating a new version.
31019	The name for the new application image must be changed when duplicating an application image.
31020	The system residence volume component must be selected when creating an application image.
31021	This application image contains duplicate volume *{0}*.
31022	This application image contains duplicate data set *{0}*.
31023	This ADCD version doesn’t support including data set or CICS® component extracted from other z/OS.
31024	This ADCD image must include at least one CICS component in order to include CICS components extracted from other z/OS.
31025	This ADCD version doesn't support including Db2® components that are extracted from other z/OS.
31026	This ADCD image cannot include more than one Db2 runtime.
31027	This ADCD image must include at least one Db2 component to include Db2 components that are extracted from other z/OS.
31028	Duplicate table name *{1}* is found in schema *{0}*.
31029	Duplicate database name *{0}* is found.
31030	Only one ADCD Db2 component can be selected.
31031	The version of Db2 from component *{0}* does not match the version of ADCD Db2 component *{1}*.
31032	The Db2 files failed to be transferred.
31033	The specific ADCD package could not be found.
31034	The image of the specific ADCD package had been deployed.
31035	The SYSRES component *{0}* needs to be migrated.
31050	Wait for the application component *{0}* to finish processing.
31051	The application component *{0}* has been deleted.
31052	Only application components containing data sets created from an IBM Z mainframe support extraction being stopped or resumed.

Error ID	Description
31053	The extraction of this application component cannot be resumed as it was previously extracted successfully.
31054	The contents of this application component cannot be edited as the component has been deployed.
31055	The name of an application component cannot be changed when creating a new version.
31056	The name for the new application component must be changed when duplicating an application component.
31057	A new version can only be created from the latest version.
31058	This component contains duplicate data set *{0}*.
31059	This component doesn't need to be migrated.
31100	The related records for the given application image and target environment cannot be found.
31101	IPL parameters creation failed.
31102	Read properties file failed.
31103	Input stream contains a malformed Unicode escape sequence.
31104	Internal server error when attempting to read ADCD property file.
31105	No ADCD files.
31106	Wrong arguments passed to the configuring ADCD script.
31107	The specified log *{0}* is not found.
31108	The application image size is 0 and could not be deployed to cloud platform.
31109	Deployment to this target system has already been started.
31110	The application image cannot be deployed to cloud by using this REST API.
31111	A target cloud instance cannot be deprovisioned while it is being provisioned.
31112	A target cloud instance cannot be deprovisioned while it is being deployed.
31113	Deprovision of *{0}* from target cloud platform *{1}* failed with error return code *{2}*.
31114	A target cloud instance cannot be deprovisioned while it is being deprovisioned.
31115	An application image cannot be deployed to a target environment when an image has already been deployed to it.
31116	Specify the credentials to log in to z/OS for deploying *{0}*.
31117	The password for user *{0}* to log in to z/OS *{1}* has expired and needs to be changed.
31118	Cannot deploy to a target environment that is being deprovisioned.
31119	This application cannot be deployed because its data set component *{0}* has not been extracted.
31120	The user ID must not be longer than 8 characters.
31121	Authentication failed.
31122	The Linux group zpdt does not exist.
31123	TheLinux user ID ibmsys1 does not exist.
31124	Unsupported platform.
31125	*{0}* is not a supported CPU architecture.
31126	The user performing the deployment does not belong to Linux group zpdt.
31127	The IBM is already running on the target environment.
31128	A software-based license server with an IP address containing 255 in the fourth octet is not supported.
31129	The emulator licensing server hostname *{0}* could not resolve to a valid IP address from the system *{1}*.
31130	The connection to hostname *{0}* on port *{1}* from the system *{2}* failed.
31131	The deployment directory *{0}* does not exist on the target environment.
31132	Write access is required to the deployment directory *{0}*.
31133	The deployment directory does not have enough available disk space.
31143	The required software 'sftp' is not installed on the system *{0}*.
31144	The required 32-bit library 'stdlib' is not installed on the system *{0}*.
31145	The required software 'perl' is not installed on the system *{0}*.
31146	The required software 'zip' is not installed on the system *{0}*.
31147	The required software 'unzip' is not installed on the system *{0}*.
31148	The required software 'gzip' is not installed on the system *{0}*.
31149	The required software 'bc' is not installed on the system *{0}*.
31150	The required library 'libasound2' is not installed on the system *{0}*.
31151	You do not have permission to run the command *{0}* on the system *{1}*.
31152	You do not have permission to change the owner of file *{0}* on the system *{1}*.
31153	Failed to list all of the privilege management commands.
31154	*{0}* is not a supported Linux distribution.
31155	The IBM needs to be running in order to deploy data sets.

Error ID	Description
31159	Cannot run command 'nc' to validate the connectivity to hostname *{0}* on port *{1}* from the system *{2}*.
31160	Cannot run command 'getent' to resolve hostname *{0}* from the system *{1}*.
31162	This application cannot be deployed because it contains component *{0}* that needs to be migrated.
31163	This application cannot be deployed because it needs to allocate at least 1 CP to the image, with a default recommendation of 3.
31164	This application cannot be deployed because it can only allocate up to eight processors to the image.
31165	This application cannot be deployed because it cannot allocate more zIIPs than CPs.
31166	Specify the image that has been deployed to the target environment.
31167	Specify the image that has been deployed to the target environment.
31168	The application *{0}* with version *{1}* is not found to be deployed on host *{2}*.
31169	Reading devmap failed.
31170	No spare 3390 volume available for restoring data set.
31171	Volume cannot be found at *{0}*.
31172	The emulator information cannot be retrieved from the target environment *{0}*.
31173	A failure occurred when trying to update the emulator.
31174	IBMUSER cannot be set as the password for ADCD Images.
31175	The RAM that is set for the emulator cannot exceed 18 GB.
31300	Failed to list volumes.
31301	Missing or invalid parameters when executing command *{0}*, with output *{1}*.
31302	An error occurred when allocating a file when executing command *{0}*, with output *{1}*.
31303	The z/OS SMS settings cannot be acquired.
31304	The z/OS spool volumes cannot be acquired.
31305	The z/OS proclibs cannot be acquired.
31306	The z/OS APF-authorized list cannot be acquired when executing command *{0}* with output *{1}*.
31307	The z/OS TCP/IP settings cannot be acquired when executing command *{0}*, with output *{1}*.
31308	No SYSPRINT/stdout when executing command *{0}*, with output *{1}*.
31309	The load library for the z/OS extraction tool is not APF-authorized.
31310	READ access is not granted to the SAF profile in the class DASDVOL.
31311	The Volume image file cannot be opened: *{0}*.
31312	Compression initialization failed - internal error with output *{0}*.
31313	Cannot open the volume - internal error without *{0}*.
31314	Command *{0}* invoked cannot execute.
31315	Linux command *{0}* was not found.
31316	Invalid argument given to the exit statement when executing command *{0}*.
31317	Fatal error signal with return code *{0}* when executing command *{1}*.
31318	Data sets allocated failed.
31319	Volume *{0}* extraction failed.
31320	Failed to list data set.
31321	A failure occurred while retrieving IBM Z mainframe information, return code: *{0}*
31322	Retrieving volume information for data sets failed.
31323	No volume information retrieved for selected data sets.
31324	An ABEND occurred on the IBM Z mainframe with the following output.
31325	These data sets cannot be extracted because your user ID has insufficient access to read them: *{0}*.
31326	These data sets cannot be extracted because their containing volumes cannot be found: *{0}*.
31327	These data sets cannot be extracted because of unknown errors: *{0}*.
31328	APF authorization failed.
31329	The CICS region information is required.
31330	The CICS region name is required.
31331	The CICS region *{0}* was already defined.
31332	The CICS region *{0}* was not found.
31333	The LOADLIB of REXX script in command *{0}* was not found.
31334	Parsing CICS CSD got format error.
31335	You must select some GROUPs defined in the CICS CSD.
31336	Got unexpected exception when trying to retrieve CICS CSD definition for *{0}*.

Error ID	Description
31337	The input LIST name *{0}* is not a valid CSD LIST name in *{1}*.
31338	The input GROUP name *{0}* is not a valid CSD GROUP name in *{1}*.
31339	Components are created using the CICS region *{0}*. Need to delete those CICS components first.
31340	Vary Online created volumes failed with error code *{0}*.
31341	The Db2 subsystem information is required.
31342	The Db2 subsystem ID is required.
31343	The Db2 subsystem ID *{0}* was already defined.
31344	The Db2 subsystem ID *{0}* was not found.
31345	An error occurred when allocating required work files for call to DSNTEP2.
31346	The expected DDL file is missing from the tmp directory.
31347	An error occurred when creating target Db2 objects from DSNTEP2.
31348	DDL generation warning.
31349	An error occurred when reading or writing DDL file for update with FEUXPUNA.
31350	An error occurred when executing DSNUTILU or Db2 Load ended with RC > 4.
31351	No tables can be unloaded because input table list is empty or validation resulted in an empty file.
31352	The number of tables exceeds the maximum number of tables that can be unloaded.
31353	An error occurred when setting up DSNREXX Environment (RXSUBCOM).
31354	An error occurred when connecting to Db2 SSID.
31355	An error occurred when connecting to Db2 Location.
31356	An error occurred when defining Db2 Cursor.
31357	An error occurred when opening Db2 Cursor.
31358	An error occurred during Db2 Prepare.
31359	Invalid parameter passed to FEUXSQLP resulting in error for SQL command setup.
31360	Max Rows returned from SQL Call, max rows are set to $1.
31361	Insufficient parameters or error reading parameter file.
31362	No tables were found in the table lists or an error occurred when reading the table list file.
31363	Db2 unload failed.
31364	An error occurred when updating Db2 Load parameters.
31365	The expected PIPE or file was not found.
31366	An error occurred when allocating the required work files for Db2 Admin Tool DDL generation.
31367	Utility Statement exceeds the maximum length that is allowed.
31368	The schema parameter string is required.
31369	Db2 Admin Tool DDL generation error.
31370	Db2 subsystem is in use.
31371	Unable to connect to the Db2 subsystem with the given parameters.
31600	Parameter object required.
31601	Image service is not available on the cloud.
31603	The label of the cloud platform is required.
31604	The label of the cloud platform already exists.
31605	Network is required.
31606	The selection of flavor is required.
31607	The selection of server image is required.
31608	Compute service is not available on the cloud.
31609	The specific flavor cannot be found.
31612	ID is invalid.
31649	The default user ID is required.
31650	Cannot find the uuid of the device.
31652	Availability zone is required.
31653	Floating IP pool is required.
31654	The image is not active.
31655	The image is not found.
31656	The flavor is not public accessed.
31657	The flavor is not found.
31700	The target environment could not be deleted as an application image is deployed to it.

Error ID	Description
31701	Required parameters are missing for deleting a target environment.
31750	Privilege command option already exists.
31751	The privilege command syntax is required to be defined on the *<a href*.
31752	ID is invalid.
31753	Privilege command option does not exist.
31754	Cannot find network interfaces from server *{0}\:{1}*.
31755	The deployment uuid is required.
31756	The deployment does not exist.
31757	The deployment has not finished yet.
50010	Configuring TCPIP on the wazi-sandbox container failed.
50011	Invalid original z/OS IP address.
50012	Invalid TCP volume request parameter.
50013	Invalid data set name.
50014	Invalid TCP member name.
50015	Invalid z/OS IP.
50016	Invalid gateway IP.
50017	Configuring ADCD IPL on the wazi-sandbox container failed.
50018	Invalid RSU request parameter.
50019	Invalid ADCD IPL request parameter.
50026	Querying license on the wazi-sandbox container failed.
50027	Generating the parm file for the IEASYM script failed.
50028	File name for IEASYM parm file is either invalid or empty.
50029	Symbolic information for IEASYM parm file is either invalid or empty.
50031	The licensing server request is required.
50032	The licensing server hostname is required.
50033	The licensing server UUID is required.
50034	The token server request is required.
50035	The token server hostname is required.
50036	The token server UUID is required.
50037	The UUID is of an invalid format.
50038	A label is required.
50039	The label must be under 44 characters long.
50040	The label is already in use by another system.
50041	The label cannot be empty.
50042	There already exists a primary licensing system.
50081	Running command of getting the number of ZPD files failed.
50082	Decryption is not done yet in the container.
50100	The application UUID is required.
50101	The application UUID is invalid.
50102	The target system UUID is required.
50103	The target system UUID is invalid.
50104	The provision UUID is required.
50105	The provision UUID is invalid.
50106	The user name is more than 8 characters.
50107	The general properties are required.
50108	The credential of the target z/OS is required.
50109	The emulator properties are required.
50110	The IPL properties are required.
50111	The provision request string is invalid.
50112	The target system does not exist.
50113	The provision object does not exist.
50114	The list of ports contains invalid value.
50115	Failed to read the content of /etc/resolve.
50116	Failed to read the content of /etc/hosts file in container.

Error ID	Description
50117	Could not resume provision for *{0}* because it is not a failed provision.
50118	The *{0}* cannot be deprovisioned with status *{1}*.
50119	The target system cannot exceed the maximum number of allowed provisions.
50120	The number of containers that a target system has must be between 1 and 5.
50121	Failed to send temp file to storage server location *{0}*.
50122	Failed to read the content of /zdt/zsystem/FEUCUTIL file in container.
500128	Failed to create PVC *{0}*.
500129	Failed to create custom resource of volume copy *{0}*.
500130	Container does not exist for the given provision UUID *{0}* and container ID *{1}*.
50900	Unable to establish an SSH connection to the SSH service listening on port *{1}* of host *{0}*.
50920	Unable to establish an HTTPS connection to the Docker daemon listening on port *{1}* of host *{0}*.
50940	Unable to establish an HTTPS connection to the instance controller service listening on port *{1}* of host *{0}*.
50941	A failure occurred while executing an Instance Controller API.
50942	An unsupported Instance Controller HTTP method was called.
50943	An attempt was made to call an Instance Controller API with invalid credentials (HTTP status code 401).
50944	An attempt was made to call an Instance Controller API for which the user is not authorized to access (HTTP status code 403).
50945	An attempt was made to call an Instance Controller API that does not exist (HTTP status code 404).
50946	An attempt was made to call an Instance Controller API with an incorrect HTTP method (HTTP status code 405).
50947	An attempt was made to call an Instance Controller API that is not capable of generating a response for the request Accept headers (HTTP status code 406).
50948	An attempt was made to call an Instance Controller API using a media type that is not supported by this API (HTTP status code 415).
50949	An attempt was made to call an Instance Controller API due to a conflict (HTTP status code 409).
50950	An attempt was made to call an Instance Controller API that no longer exists (HTTP status code 410).
50951	An attempt was made to call an Instance Controller API and caused an unknown error.
50980	An oprmsg command failed.
50981	The requested oprmsg command is invalid.
50982	The requested oprmsg command is empty.
60000	This target environment has not been provisioned.
60001	This entity *{0}* is being used by another user, please retry after some time.
60215	The cloning option is disabled on target environment *{0}*.
60218	Failed to clone Persistent Volume Claim *{0}* in namespace *{1}*.
60220	Failed to create ConfigMap *{0}* in namespace *{1}*.
60221	Failed to create Secret *{0}* in namespace *{1}*.
60222	Failed to delete ConfigMap *{0}* in namespace *{1}*.
60223	Failed to delete Secret *{0}* in namespace *{1}*.
60224	Failed to delete Persistent Volume Claim *{0}* in namespace *{1}*.
60225	Failed to update Persistent Volume Claim *{0}* in namespace *{1}*.
60228	Persistent Volume Claim *{0}* already exists in namespace *{1}*.
70001	The extraction utility (/usr/lpp/IBM/zdt/feuc) on this z/OS returned an unexpected result.
70991	Checking *{0}* version for the extraction utility SFTP module zDTMainframeSFTP.
70992	A Java™ runtime environment is either not installed on this z/OS or is not added to the z/OS UNIX System Services PATH environment variable for this user.
70993	The extraction utility SFTP module (/usr/lpp/IBM/zdt/zDTMainframeSFTP.jar) on this z/OS is either not installed or corrupted.
70994	The extraction utility SFTP module (/usr/lpp/IBM/zdt/zDTMainframeSFTP.jar) on this z/OS is older than the current version. Please upgrade to the latest version.
70995	The extraction utility (/usr/lpp/IBM/zdt/feuc) has not been installed.
70996	The extraction utility version file could not be found.
70997	The extraction utility (/usr/lpp/IBM/zdt/feuc) on this z/OS is older than 12.0.1. Please upgrade to the latest version.
70998	The extraction utility (/usr/lpp/IBM/zdt/feuc) on this z/OS is older than the current version.
70999	Internal error acquiring information from z/OS.
79999	An error occurred when connecting to z/OS: *{0}*.
99998	Unexpected exception.

Error ID	Description
99999	Unknown error.

Logs

Learn how to see logs for Wazi Image Builder.

- **Auditing the web server**
  Learn about the steps to enable audit log for Wazi Image Builder.
- **Finding logs**
  You can refer to log files for troubleshooting. This topic provides an overview of the key log files that are associated with Wazi Image Builder.

Auditing the web server

Learn about the steps to enable audit log for Wazi Image Builder.

The audit log for the web server is defined in the *<installation_directory>*/Liberty/usr/servers/wib-server/audit.xml file.

To enable the support, follow the instruction that is stated in the audit.XML file.

For more information, see Default Audit File Handler.

Finding logs

You can refer to log files for troubleshooting. This topic provides an overview of the key log files that are associated with Wazi Image Builder.

On the Linux system where the web server is installed

/var/log/wib*.log
: Records events and activities of the web server installer.

<installation_path>/wib/wib-server/logging.xml
: Specifies the logging level configurations of the web server.

<installation_path>/wib/wib-server/logs/*.log
: Records events and activities within the web server.

<installation_path>/wib/wib-server/postgresql/logfile
: Records database-related issues that might occur when attempting to start the web server.

<installation_path>/wib/wib-server/postgresql/zdt_db_conf.log
: Records database-related issues that might occur when attempting to start the web server.

On the Linux system where the license server is installed

/home/ibmsys1/*.log
/home/ibmsys1/UIMserver/
/var/hasplm/error.log
/var/hasplm/log/

On the source z/OS system where ZD&T z/OS Extraction Utilities are installed

/tmp/feucdebug_log.HH.MM.SS.ssssss or /tmpDir/feucdebug_log.HH.MM.SS.ssssss
: Records events and activities of the volume discovery tool.

Uninstalling the web server

To uninstall the web server, complete the following steps:

1. Open the directory where the installation package wib-install.tgz is stored.

   ```
   cd <directory>
   ```

2. Change the authority of the installation package.

   ```
   chmod 755 <package_name>
   ```

3. Decompress the installation package.

   ```
   tar -xvf <package_name>
   ```

   When the decompression is complete, the installers can be found in the target directory. If you don't specify a target directory, the installers can be found in the same directory where the installation package is.

   Web server installer
   : wib-install

4. Run the installer with the root user ID, and then follow instructions to uninstall.

   ```
   ./<installer_name>
   ```

   Alternatively, you can uninstall the web server silently by using the following command:

   ```
   ./<installer_name> --uninstall --wib
   ```

Sharing feedback

You can share anonymous data on the web server to help shape the product.

Your feedback matters! To provide feedback, log in to the web server, and then click the upper-right Information menu  > Feedback.

Creating z/OS VSIs in IBM Cloud VPC

Learn how to create and configure IBM z/OS virtual server instances in IBM Cloud Virtual Private Cloud (VPC) from the IBM Cloud console.

For the latest updates of z/OS dev and test stock images, see Change log for z/OS stock images and Configurations in z/OS stock images.

If you experience difficulties with your instances, review the possible causes in Troubleshooting z/OS virtual server instances.

Change log for z/OS stock images

In this change log, you can learn about the latest changes, improvements, and updates for the z/OS development and test stock image for use with IBM Wazi as a Service.

It is recommended that you use the new image for your running virtual server instances.

For a list of software stack available in the latest stock image, see Configurations in z/OS stock images.

2024

April 2024

New

- The z/OS dev and test stock image in the Cloud UI is now based on z/OS 3.1.0 and named as **`ibm-zos-3-1-s390x-dev-test-wazi-1`**.
- New disk layout - CICS/MQ/IMS/DB2 and SMPE target/dlibs are on two separate disk volumes.

Changed (in comparison with z/OS 2.5 Image)

- New Program Temporary Fixes (PTFs) are installed on the z/OS system and programs.
- IBM Developer for z/OS (IDz) is upgraded from v16.0.2 to v16.0.4.
- IBM Debug for z/OS is upgraded from v16.0.2 to v16.0.4.
- IBM Explorer for z/OS is upgraded from v3.3.2 to v3.3.4.
- IBM Dependency Based Build (DBB) is upgraded from v2.0.1 to v2.0.3.
- RSE API is upgraded from v1.1.2 to v1.1.4 with the JWT support enabled.
- IBM Z Open Automation Utilities (ZOAU) is upgraded from v1.2.3 to v1.2.5.
- Db2 for z/OS remains at v13.1.503 and V12.1 is removed.
- Java v17 is added and v11 is the default version. Java v8 is available in the image.
- IBM Open Enterprise SDK for Go v1.21 is added to the image and is set as default by the environment variables. Go v1.20 is available in the image and v1.19 is removed.
- IBM Open Enterprise SDK for Node.js v20 is added and v18.0 is available in the image.
- IBM Z Workload Scheduler is upgraded from v10.1.0.2 to v10.1.0.3.
- IBM z/OS Connect is upgraded from v3.0.73 to v3.0.77.
- The z/OS 2.5 image with the name **`ibm-zos-2-5-s390x-dev-test-wazi-7`** is available in the UI.

March 2024

Changed

- The name of the z/OS 2.5 dev and test stock image is **`ibm-zos-2-5-s390x-dev-test-wazi-7`**. IBM MQ is upgraded from v9.3.3 to v9.3.5 with prerequisite/co-requisite New Program Temporary Fixes (PTFs) for the MQ installed.
- The z/OS 2.4 images with the names **`ibm-zos-2-4-s390x-dev-test-wazi-*`** are no longer available in the UI or APIs.

2023

December 2023

Changed

- The name of the z/OS 2.5 dev and test stock image is **`ibm-zos-2-5-s390x-dev-test-wazi-6`**. New Program Temporary Fixes (PTFs) are installed on the z/OS system and programs.
- IBM Z Workload Scheduler is upgraded from v10.1.0.1 to v10.1.0.2.
- IBM z/OS Connect is upgraded from v3.0.72 to v3.0.73.
- Python v3.9.5 is removed from the image. The default version is Python v3.11.
- Java 8 is upgraded from v8.0.8.8 to v8.0.8.15. Both Java v8 and Java 11 are available in the image.
- The script for RACF certificate regeneration is updated.

October 2023

New

- IBM Wazi Deploy is added to the z/OS 2.5 dev and test stock image.

Changed

- The name of the z/OS 2.5 dev and test stock image is **`ibm-zos-2-5-s390x-dev-test-wazi-5`**. New Program Temporary Fixes (PTFs) are installed on the z/OS system and programs.
- IBM Z Open Automation Utilities (ZOAU) is upgraded from v1.2.3 to v 1.2.4.5.
- IBM z/OS Connect is upgraded from v3.0.70 to v3.0.72.

August 2023

Changed

- The name of the z/OS 2.5 dev and test stock image is **`ibm-zos-2-5-s390x-dev-test-wazi-4`**. New Program Temporary Fixes (PTFs) are installed on the z/OS system and programs.
- IBM Developer for z/OS (IDz) is upgraded from v16.0.1 to v16.0.2.

- IBM Debug for z/OS is upgraded from v16.0.1 to v16.0.2.
- IBM Explorer for z/OS is upgraded from v3.3.1 to v3.3.2.
- IBM IMS is upgraded from v15.3 to v15.4.
- IBM MQ is upgraded from v9.3.2 to v9.3.3.
- IBM z/OS Connect is upgraded from v3.0.68 to v3.0.70.
- IBM Open Enterprise SDK for Go v1.20 is added to the image and is set as default by the environment variables. Go v1.19 is still available in the image.
- RSE API is upgraded from v1.1.1 to v1.1.2.
- Extended address volume (EAV) is now enabled by default.

July 2023

Changed

- The name of the z/OS 2.5 dev and test stock image is **`ibm-zos-2-5-s390x-dev-test-wazi-3`**. New Program Temporary Fixes (PTFs) are installed on the z/OS system and programs.
- IBM Dependency Based Build (DBB) is upgraded from v2.0 to v2.0.1.
- IBM z/OS Connect is upgraded from v3.0.67 to v3.0.68.
- RSE API is updated to use RACF Certificates and key rings instead of keytool and keystores.
- Some Hash-based Message Authentication Codes (HMAC) are removed from the SSH server configuration file **`/etc/ssh/sshd_config`**.
- The ".ibm.com" domain was previously appearing as the domain suffix on the output of the hostname command. It is now removed from this output.

June 2023

New

- The build date and time is added to the initial screen that you see when you connect to the instance via TN3270.
- A new TSO logon procedure **`DB12PROC`** is available if you want to use Db2 v12 SPUFI. The default TSO logon procedure for Db2 v13 SPUFI is **`DB13PROC`**.

Changed

- The name of the z/OS 2.5 dev and test stock image is **`ibm-zos-2-5-s390x-dev-test-wazi-2`**. New Program Temporary Fixes (PTFs) are installed on the z/OS system and programs.
- Db2 v12 is added back to the image. Both v12 and v13.1.503 will be started automatically by default.
- The size of the message queue data set **`IMS.LGMSG`** is increased with more available space.

May 2023

New

- The z/OS dev and test stock image in the Cloud UI is now based on z/OS 2.5 and is named **`ibm-zos-2-5-s390x-dev-test-wazi-1`**.
- Data Set File System (DSFS) is added to the image.
- IBM_DB is added to enable Python access to Db2. See sample script **`/u/ibmuser/runibmdb`**.

Changed

- New Program Temporary Fixes (PTFs) are installed on the z/OS system and programs.
- IBM Developer for z/OS (IDz) is upgraded from v16.0.0 to v16.0.1.
- IBM Debug for z/OS is upgraded from v16.0.0 to v16.0.1.
- Db2 for z/OS is upgraded from v13.1 to v13.1.503.
- IBM z/OS Connect is upgraded from v3.0.66 to v3.0.67.
- RSE API is upgraded from v1.1.0 to v1.1.1 and is switched to use Java 11.
- IBM Dependency Based Build (DBB) is switched to use Java 11.
- IBM Explorer for z/OS is upgraded from v3.3.0 to v3.3.1.
- Python v3.11 is set as default. Python v3.9.5 is still supported in the image.
- IBM Z Open Automation Utilities (ZOAU) is upgraded from v1.2.2 to v1.2.3. ZOAU is configured for Python 3.11 and 3.9.
- zUNIT port is switched to SSL port.
- The z/OS 2.4 image with the name **`ibm-zos-2-4-s390x-dev-test-wazi-10`** is also available in the UI and is updated with security fixes only.

April 2023

Changed

- The name of the z/OS 2.4 dev and test stock image is **`ibm-zos-2-4-s390x-dev-test-wazi-9`**. New Program Temporary Fixes (PTFs) are installed on the z/OS system and programs.
- IBM MQ is upgraded from v9.3.1 to v9.3.2.
- IBM Open Enterprise SDK for Go v1.18 is removed from the image. Only v1.19 is supported.
- IBM z/OS Connect is upgraded from v3.0.65 to v3.0.66.

March 2023

Changed

- The name of the z/OS 2.4 dev and test stock image is **`ibm-zos-2-4-s390x-dev-test-wazi-8`**. New Program Temporary Fixes (PTFs) are installed on the z/OS system and programs.
- IBM Open Enterprise SDK for Python is upgraded from v3.10 to v3.11. Python v3.9.5 is still supported in the image.
- IBM Open Enterprise SDK for Node.js is upgraded from v16.0 to v18.0.
- IBM z/OS Connect is upgraded from v3.0.64 to v3.0.65.
- RACF is changed to use **`PHRASE`** instead of **`PASSWORD`** for user authentication. A password phrase is a character string that consists of mixed-case letters, numbers, and special characters including blanks. For more information, see Passwords and password phrases and Specify user password or password phrase.

February 2023

New

- IBM Compiler for REXX v1.4 is added to the z/OS 2.4 dev and test stock image.
- IBM Open Enterprise SDK for Go v1.19 is added to the image and is set as default by the environment variables. Go v1.18 is still available in the image.

Changed

- The name of the z/OS 2.4 dev and test stock image is **`ibm-zos-2-4-s390x-dev-test-wazi-7`**. New Program Temporary Fixes (PTFs) are installed on the z/OS system and programs.
- IBM MQ is upgraded from v9.3 to v9.3.1 and will be started by default beginning with the new image.
- IBM IMS is upgraded from v15.1 to v15.3 and will be started by default beginning with the new image. Continuous enhancements are released as PTFs, which can include one or more new IMS functions or support for new technologies or products. For more information, see IMS enhancement PTFs.
- IBM Z Open Automation Utilities (ZOAU) in IBM Developer for z/OS (IDz) v16 is upgraded from v1.2.0 to v1.2.2. ZOAU v1.1.1 is still available in the image.
- IBM Z Workload Scheduler is upgraded from v10.1 to v10.1.0.1.
- IBM z/OS Connect is upgraded from v3.0.61 to v3.0.64.

2022

December 2022

Changed

- The name of the z/OS 2.4 dev and test stock image is **`ibm-zos-2-4-s390x-dev-test-wazi-6`**. New Program Temporary Fixes (PTFs) are installed on the z/OS system and programs.
- IBM Developer for z/OS (IDz) is upgraded from v15.0.5 to v16.0.0.
- IBM Debug for z/OS is upgraded from v15.0.5 to v16.0.0.
- IBM Explorer for z/OS is upgraded from v3.2.0.20 to v3.3.0.
- RSE API is upgraded from v1.0.10 to v1.1.0.
- IBM Dependency Based Build (DBB) is upgraded from v1.1.3 to v2.0.
- Z Open Automation Utilities (ZOAU) is upgraded from v1.1.1 to v1.2.0. ZOAU v1.1.1 is still available in the image.
- Java is upgraded from v8 to v11. Java v8 is still available in the image.
- IBM Open Enterprise Python for z/OS is upgraded from v3.9 to v3.10. Both v3.9.5 and v3.10 are supported in the image.
- The unsecured Db2 DRDA TCP/IP port 8100 is closed. All DB2 requests should use the SSL port 8102. The DB2 CLP sample is removed as it does not work over SSL Db2 Port.
- The unsecured z/OS Connect port 9080 is closed. All z/OS Connect requests should use the SSL port 9443.
- SSL is enabled for Debug Manager port 8135 via z/OS Policy Agent.
- SSL is enabled for CICS CS IP Socket port 8120 via z/OS Policy Agent.
- Weak ciphers are removed from port 992, 10443, 8137, 8154, 8194, and 8195 to improve configuration security.
- I/O definition file (IODF) on the source system is updated to support 100 storage volumes.

November 2022

New

- OMVS is added to the ISPF user panel (ISR@390U) Option 15.

Changed

- The name of the z/OS 2.4 dev and test stock image is `ibm-zos-2-4-s390x-dev-test-wazi-5`. New Program Temporary Fixes (PTFs) are installed on the z/OS system and programs.
- IBM z/OS Connect is upgraded from v3.0.60 to v3.0.61.
- Netview Timer is adjusted to execute every 6 hours to ensure that BPXAS spool entries are cleaned up automatically each day.
- JES2 Initialization is adjusted to increase the limit of job output elements (JOEs) and job queue elements (JQEs).

October 2022

New

- Program product libraries for IMS v15.1 are provided on volume PRDVS1 in the z/OS 2.4 dev and test stock image.
- Program product libraries for IBM MQ v9.3 Continuous Delivery are provided on volume PRDVS1 in the z/OS 2.4 dev and test stock image.

Changed

- The name of the z/OS 2.4 dev and test stock image is `ibm-zos-2-4-s390x-dev-test-wazi-4`. New Program Temporary Fixes (PTFs) are installed on the z/OS system and programs.
- RSE API is upgraded from v1.0.9 to v1.0.10.
- IBM Explorer for z/OS is upgraded from v3.2.0.19 to v3.2.0.20.
- IBM z/OS Connect is upgraded from v3.0.58 to v3.0.60.

August 2022

Changed

- The name of the z/OS 2.4 dev and test stock image is `ibm-zos-2-4-s390x-dev-test-wazi-3`. New Program Temporary Fixes (PTFs) are installed on the z/OS system and programs.
- CICS Transaction Server for z/OS is upgraded from v5.6 to v6.1.
- CICS subsystem is upgraded from CICSTS56 to CICSTS61.
- CICS software data sets are upgraded from `CICSTS56.*` to `CICS.V6R1M0.**`.
- Db2 for z/OS is upgraded from v12.1 to v13.1.
- Db2 subsystem is upgraded from DBC1 to DBD1.
- Db2 software data sets are upgraded from `DB2.V12R1M0.*` to `DB2.V13R1M0.**`.
- IBM Z NetView is upgraded from v6.3 to v6.4.
- IBM z/OS Connect is upgraded from v3.0.57 to v3.0.58.

July 2022

New

- Tape devices are supported in the z/OS 2.4 dev and test stock image.

Changed

- The name of the z/OS 2.4 dev and test stock image is `ibm-zos-2-4-s390x-dev-test-wazi-2`. New Program Temporary Fixes (PTFs) are installed on the z/OS system and programs.
- Enterprise COBOL for z/OS is upgraded from 6.3 to 6.4.
- Enterprise PL/I for z/OS is upgraded from v5.3 to v6.1.
- IBM Open Enterprise SDK for Go is upgraded from v1.17 to v1.18.
- IBM z/OS Connect is upgraded from v3.0.56 to v3.0.57.
- IBM Developer for z/OS (IDz) is upgraded from v15.0.4 to v15.0.5.
- IBM Debug for z/OS is upgraded from v15.0.4 to v15.0.5.
- RSE API is upgraded from v1.0.8 to v1.0.9.
- IBM Explorer for z/OS is upgraded from v3.2.0.18 to v3.2.0.19.

Configurations in z/OS stock images

IBM provides the z/OS development and test stock image for you to create virtual server instances. The stock image is refreshed on a regular basis.

You can use z/OS dev and test virtual servers only for development, testing, employee education, or demonstration of your applications that run on z/OS. Licensee must not use the Program for production workloads of any kind, nor robust development workloads, including without limitation, production module builds, pre-production testing, stress testing, or performance testing.

Products and services

The z/OS development and test stock image preinstalls the products and services as listed in the following table.

z/OS base components
• IBM z/OS Liberty Embedded • z/OS Container Extensions (zCX) • z/OS File System (was Distributed File Service) • Base Control Program (BCP) • Bulk Data Transfer (BDT) • Common Information Model (CIM) • Communications Server • Cryptographic Services • DFSORT • DFSMSdfp • ESCON Director Support • First Failure Support Technology/MVS (FFST/MVS) • Hardware Configuration Definition (HCD) • IBM HTTP Server powered by Apache • IBM Documentation for z/OS • IBM z/OS Management Facility (z/OSMF) • Integrated Security Services • ISPF • JES2 • Language Environment® • Metal C Runtime Library • MICR/OCR • Network File System (NFS) • Resource Access Control Facility (RACF) • Runtime Library Extensions • SMP/E • System Display and Search Facility (SDSF) • Terminal Input Output Controller (TIOC) • Time Sharing Option/Extensions (TSO/E) • z/OS Font Collection • z/OS OpenSSH • z/OS UNIX System Services • 3270 PC File Transfer Program • Environmental Record Editing & Printing Program (EREP) • GDDM • High Level Assembler (HLASM) • ICKDSF (Device Support Facility) • XL Open C/C++ • Hardware Configuration Manager (HCM) • IBM Knowledge Center for z/OS • IBM Tivoli® Directory Server for z/OS (IBM TDS for z/OS) • Infoprint Server • IBM Z® Deep Neural Network Library (zDNN) • Resource Measurement Facility • z/OS Security Level 3 • Communications Server Security Level 3 • XML Toolkit for z/OS
z/OS additional components

z/OS additional components
• IBM Compiler & Library for REXX/370 V1.4 • IBM Enterprise COBOL for z/OS V6.4 • IBM Enterprise PL/I for z/OS V6.1 • IBM CICS Transaction Server for z/OS V6.1 • IBM Db2 for z/OS V12.1 (applicable for z/OS 2.5 only) • IBM Db2 for z/OS V13.1 • IBM Information Management System V15.4 • IBM Z NetView V6.4 • IBM System Automation for z/OS V4.3 • IBM Z Open Automation Utilities V1.1.1 • IBM MQ for z/OS CD V9.3.5 • IBM Open Enterprise SDK for Python V3.9 • IBM Open Enterprise SDK for Python V3.11 • IBM Open Enterprise SDK for Go V1.19 (*applicable for z/OS 2.5 only*) • IBM Open Enterprise SDK for Go V1.20 • IBM Open Enterprise SDK for Go V1.21 (*applicable for z/OS 3.1 only*) • IBM Open Enterprise SDK for Node.js V18 • IBM Open Enterprise SDK for Node.js V20 (*applicable for z/OS 3.1 only*) • IBM SDK for z/OS, Java 2 Technology Edition, V8 • IBM Semeru Runtime Certified Edition for z/OS V11 • IBM Semeru Runtime Certified Edition for z/OS V17 (*applicable for z/OS 3.1 only*) • IBM Developer for z/OS Enterprise Edition V16.0.2 - includes IBM Debug 16.0.2, z/OS Explorer 3.3.2, DBB 2.0.1, RSE API 1.1.2, ZOAU 1.2.3, ZUNIT, Wazi Deploy and so on (*applicable for z/OS 2.5 only*) • IBM Developer for z/OS Enterprise Edition V16.0.4 - includes IBM Debug 16.0.4, z/OS Explorer 3.3.4, DBB 2.0.3, RSE API 1.1.4, ZOAU 1.2.5, ZUNIT, Wazi Deploy 3.0.2 and so on (*applicable for z/OS 3.1 only*) • IBM Z Workload Scheduler V10.1.0.2 (*applicable for z/OS 2.5 only*) • IBM Z Workload Scheduler V10.1.0.3 (*applicable for z/OS 3.1 only*) • IBM z/OS Connect Enterprise Edition V3.0.73 (*applicable for z/OS 2.5 only*) • IBM z/OS Connect Enterprise Edition V3.0.77 (*applicable for z/OS 3.1 only*)

For this release, z/OS Container Extensions (zCX) is not supported and is restricted for use.

Reserved configurations

The following configurations are reserved in the z/OS dev and test stock image.

Components/Tasks	Configurations
FTP	Port: 21
SSH	Port: 22
TN3270	Secure port: 992
ZOS CONNECT	Port: 9443
z/OSMF	Port: 10443
Resource pool	Port: 12000-12029
IBM z/OS Management Facility (z/OSMF) Plug-ins	• Software Management and Software Update • Security Configuration Assistant • ISPF (account number IZUACCT, logon proc IZUFPROC) • Network Configuration Assistant • SDSF (imported task from **`/usr/lpp/sdsf/zosmf/sdsf.properties`**, defined RACF class **`ZMFAPLA`** profiles **`IZUDFLT.ZOSMF.IBMSDSF.SDSF.JOBS`** and **`IZUDFLT.ZOSMF.IBMSDSF.SDSF.SETTINGS`**) • Resource Monitoring • z/OS Operator Consoles (default EMCS console name iserVS01)
Db2	• Software data sets: **`DB2.V13R1M0.**`** • Subsystem: **`DBD1`** • Subsystem data sets: **`DBD1.*`** • Db2 SYSADM = IBMUSER • Resync port: 8101 • Secure port: 8102

Components/Tasks	Configurations
CICS	• Software data sets: **CICSTS.V6R1M0.** • Subsystem: **CICSTS61** • Subsystem data sets: **CICSTS61.*** • TCP/IP Listener and Enhanced Listener ports: 8120, 8121 • Debug DTCN non-SSL port: 8150 • Debug DTCN SSL port: 8153 • CICS Management Client Interface (CMCI) port: 8154 • zUnit SSL port: 8155 • Sample: General insurance application (GENAPP)
Debug	• Dependency Based Build (DBB) Shared Daemon port: 8180 • Debug Manager (DBGMGR) port: 8135 • EQAPROF Debug Profile Service HTTP port: 8191 • EQAPROF Debug Profile Service HTTPS port: 8192 • EQARMTD Remote debug Daemon HTTPS port: 8194
z/OS Explorer	• z/OS Explorer (RSED) port: 8137, 8138, 8139 • z/OS Explorer JES2 Monitor (JMON) port: 8115 • z/OS Explorer API HTTPS (RSEAPI) port: 8195

Table 2. Reserved configurations in the z/OS dev and test stock image

Reserved devices

Each device unit is represented by a unit control block (UCB) in the z/OS image. The following table lists the device numbers that are reserved in the stock image. A device group is shown as **device number,range**. For example, the entry **0060,2 3277-2** indicates a device group of two devices of type 3277-2 with consecutive device numbers from 0060 through 0061.

Device number	I/O device
0060,2	3277-2
1500,16	OSA
DE00,48	3390

Table 3. Devices in the z/OS dev and test stock image

Viewing products and services in z/OS stock images

You can use the System Modification Program/Extended (SMP/E) database, which is also called the consolidated software inventory (CSI), to query preinstalled products and services in the z/OS stock images.

A CSI can be divided into multiple partitions through the VSAM key structure. Each partition is referred as zone. Target zones are used to record information about the status and structure of the z/OS stock images. The following table shows three CSIs and its target zones for z/OS stock images:

CSI name	Description	Target zones
SMPE.ZOS24.CSI	Products and services in base z/OS	**TGT24**
SMPE.PROGPROD.CSI	Middleware and programming languages	**CICS61T**, **COB640T**, **DB2V13T**, **JAVA80T**, **PLI610T**, **PY390T**, **SA430T**, **TNV640T**, **ZOA110T**
SMPEDEV.PROGPROD.CSI	Additional contents in the z/OS dev and test stock image	**DZEE15T**, **GO118T**, **NODE16T**, **TWS101T**, **ZCEE30T**

Table 1. CSIs and target zones for z/OS stock images

You can use the following methods to view products and services in z/OS stock images:

- Use the **LIST** command: To view entries and check each CSI data set, you can use the **LIST PRODUCT** or **LIST SYSMOD** commands. For more information about SMP/E command syntax and capability, see z/OS SMP/E Commands.
- Use the Interactive System Productivity Facility (ISPF): SMP/E can be run through dialogs under ISPF. You must access the SMP/E dialog from ISPF and use option **12.10**. For more information, see z/OS SMP/E User's Guide.
- Use the z/OSMF user interface: To define the z/OS stock image to z/OSMF, you must use z/OSMF software instances of the installed software. You need to complete the following steps:

    1. Create up to three software instances for each CSI data set. For more information, see Adding software instances.

2. Select pull-down list of the **Action** menu, you can view the contents and Program Temporary Fixes (PTFs) that are installed across all these software instances.

You can query multiple software instances at the same time.

- Use the prepared SMP/E reports: You can view products and services in z/OS stock images by using the data set **`$INFO.SYSMODS`**, which exist in the z/OS stock image already. You can check each prepared report by specifying their volumes and access those volumes through the ISPF or z/OSMF.
    - **`D24VS1`** (IPL volume)
    - **`PRDVS1`** (Product volume)
    - **`DEVVS1`** (Development and test volume)

IBM Z Hypervisor as a Service (zHYPaaS) overview

The zHYPaaS host control program creates a virtual machine (VM) runtime environment for each guest operating system (OS) and can host many guests at the same time.

The IBM z/Architecture provides a means for the guest operating system to determine the available hardware facilities, including both actual and virtualized hardware. The hardware information can be obtained through the Store System Information (STSI), Store Facility List (STFL), Store Facility List Extended (STFLE), Store CPU ID (STIDP) instructions, or any related interfaces of the hypervisor and the guest operating system. However, some hardware facilities or features might not be available to the guest operating system because the zHYPaaS chose not to support them. In addition, zHYPaaS might hide or alter certain aspects of the underlying actual hardware. For more information about the hardware environment, see How to determine the hardware environment.

Understanding and determining when zHYPaaS is the host

zHYPaaS presents a STIDP environment code that indicates an alternate level 3 configuration and uses the STSI instruction to provide three levels of the machine configuration information, including the central processor complex (CPC), the logical partition (LPAR), and the virtual machine in which the guest operating system is running. Under the virtual machine environment, the STSI instruction returns a System-Information Block (SYSIB) 3.2.2 with the VM Control-Program Identifier, which can be used to identify that zHYPaaS is the host control program. Both the STIDP environment code and the STSI instruction must be used together to determine that zHYPaaS is the host.

STIDP instruction

zHYPaaS returns STIDP information as follows:

- The STIDP environmental code is set to **`x'FD'`** to indicate an alternate level 3 configuration.
- Configuration Identification: Bit positions **`8-31`** contains six hexadecimal digits as described in the following order:
    - Two digits containing the LPAR user-partition identifier (UPID) that is assigned to the virtual server instance. It is the same value that is returned in the LPAR-number contained in STSI SYSIB 2.2.2. For more information, see STSI instruction.
    - Four digits are the rightmost 4 hex digits of the CPC sequence code. For more information about the sequence code, see STSI altered by zHYPaaS.
- The Machine-Type Number in bit positions **`32-47`** matches a machine type related to the virtual server instance hardware profile that is selected. It represents the actual IBM Z hardware machine type of the IBM Z hardware, which the virtual server instance is running on.

STSI instruction

Typically, zHYPaaS is running in an LPAR on a machine with many other LPARs. In some environments, the STSI instruction might not alter the actual LPAR and CPC information. However, due to the nature of IBM Cloud, the zHYPaaS alters some values for the CPC, LPAR, and VM to make each virtual server instance appear to be running in its own LPAR and CPC. Therefore, the LPAR and CPC do not appear to have more resources than the VM.

STSI VM SYSIB 3.2.2

The STSI VM information is unique to zHYPaaS virtual server instances and the key information is presented as follows:

- The first token of the Control-Program Identifier is set to Extended Binary-Coded Decimal Interchange Code (EBCDIC) **`zHYPaaS`** and it ends at the first blank. Extra identification information might or might not be provided after the first token. The first token is sufficient to determine whether zHYPaaS is the host or not.
- The Universally-Unique Identification (UUID) is unique for each virtual server instance.

- The EBCDIC Virtual-Machine Name is set to the first 8 characters of the Extended Virtual Machine Name. The extended name must be used as the VM name is greater than 8 characters.
- The EBCDIC Extended VM Virtual-Machine name is set to the cloud instance identifier (ID). The ID is static and persists for the lifecycle of the virtual server instance until that virtual server instance is deleted. All lowercase letters are capitalized and the virtual server instance's ID can also be retrieved through the cloud provided instance details. For more information about how to retrieve the identifier based on different operating systems, see Retrieving the virtual server instance identifier.

STSI altered by zHYPaaS

For zHYPaaS, some STSI values are altered:

- The Machine-Type Number matches a machine type related to the virtual server instance hardware profile that is selected. It represents the actual IBM Z hardware machine type of the IBM Z hardware, which the virtual server instance is running on.
- The first character of the four-character Plant of Manufacture (POM) is set to EBCDIC character `C`. For zHYPaaS, the POM is set to `C1` with trailing blanks, but the second character or the remaining 2 trailing blank characters might change in the future to denote specific attributes.
- The Logical-CPU ID in SYSIB 2.2.1 and the Logical-Partition Number in SYSIB 2.2.2 are set to a virtual LPAR number that has no relationship with the actual LPAR that zHYPaaS is running in. The virtual LPAR number is not unique for each zHYPaaS instance even within the same CPC.
- The EBCDIC CPC sequence code is unique for each virtual server instance. However, it has no relationship with the actual machine sequence code or actual CPU ID. The CPU ID that is retrieved from the STIDP instruction includes the rightmost 4 hex digits of the configuration-identification field.
- The machine Model-Capacity Identifier is EBCDIC character `A00`.
- The Model-Permanent-Capacity Identifier (P) and Model-Temporary-Capacity Identifier (T) are always equal to the Model-Capacity Identifier value.
- The Model-Capacity Rating (CR) and other model capacity ratings are set to `0`.

For more information, see z/Architecture Principles of Operation.

Setting up your IBM Cloud account

Before you get started with IBM Cloud Virtual Private Cloud (VPC), you need to set up a valid IBM Cloud account.

The following guides and resources can help you set up account authentication, manage your account settings, effectively organize resources in your account, and control access to resources.

- Account types
- Setting up your IBM Cloud account
- Assigning access to resources by using access groups

Getting started with IBM Virtual Private Cloud (VPC)

With IBM Cloud Virtual Private Cloud (VPC), you can use the UI, CLI, and API to quickly provision virtual server instances with high network performance. VPC infrastructure contains a number of Infrastructure-as-a-Service (IaaS) offerings, including Virtual Servers for VPC.

Before you begin, set up your IBM Cloud account to access VPC. Make sure that your account is upgraded to a paid account.

Following are the overall steps to create and configure your VPC and other attached resources. For more detailed instructions, see Using the IBM Cloud console to create z/OS virtual server instances.

1. Create your SSH key.
2. Create a VPC.
3. Create subnets in one or more zones. You can create subnets in suggested prefix ranges or in your own IP ranges that you bring to IBM Cloud.
4. Attach a public gateway if you want to allow all resources in a subnet to communicate with the public internet.
5. Configure an access control list (ACL) to limit the subnet's inbound and outbound traffic.
6. Create virtual server instances with the vCPU, RAM, and instance storage (optional) that's right for your workload. When you're planning to provision virtual server instances for IBM Cloud VPC, refer to Planning for instances. Note that zIIP (z Integrated Information Processor) processors are not supported, and vCPUs for z/OS virtual server instances are general-purpose processors only.

7. If you want more storage, create block storage volumes and attach them to your instances.
8. To define the inbound and outbound traffic that is allowed for instances, configure their security groups.
9. To enable your VPC to connect securely to another private network, such as your on-premises network or another VPC, create a virtual private network (VPN).
10. If you want instances to be reachable from the internet, reserve and associate floating IP addresses.

For a general overview of the VPC infrastructure and related compute, networking, and storage concepts, see the following topics in IBM Cloud docs:

- What is Virtual Private Cloud?
- Architecture Center VPC Resources
- VPC overview
- About networking for VPC
- About virtual server instances for VPC
- About storage for VPC
- Understanding IaaS basics
- IBM Cloud Monitoring

Granting user permissions for VPC resources

IBM Cloud™ Virtual Private Cloud uses role-based access control that enables account administrators to control their users' access to VPC resources. Access can be assigned to individual users or to groups of users by using IBM Cloud Identity and Access Management (IAM).

- For more information about IAM roles, see IAM roles and actions.
- For more information about the minimum IAM roles that a user or service ID need to interact with an IBM Cloud™ Virtual Private Cloud (VPC) infrastructure resource, see Required permissions.

This document shows examples of how the account administrator can use the IBM Cloud console to grant the correct permissions for managing VPC infrastructure resources. It covers the following scenarios:

- **Full-access scenario:** Assign an access policy so a new user can create and use all VPC infrastructure resources (including VPCs).
- **Limited access scenario:** Assign an access policy so an existing user can create and use only virtual server instances.
- **Team access scenario:** Set up resource groups and access groups to allow two separate teams to create and use the VPC resources that are assigned to their team.

You can also manage permissions through the CLI or API. For more information, see How do I use IBM Cloud IAM.

Full-access scenario

This scenario shows how to invite a new IBM Cloud user to your account and give them access to VPC infrastructure. With access, they can view, create, and update all VPC resources in the Default resource group.

To give a new user access to all VPC infrastructure resources:

1. Go to the IAM Users page in the IBM Cloud console and click **Invite users**.
2. Enter the email addresses of the users that you want to invite in the **Enter Email addresses** section.
3. In the **Assign users additional access** section, select **IAM services** and complete the following tasks:
    1. From the **What type of access do you want to assign?** list, select **VPC Infrastructure Services**.
    2. From the **Resource type** list, select **All resource types**.
    3. In the **Platform access** area, select **Editor**.
    4. In the **Resource group access** area, select **Viewer**.
    5. In the **Service access** area, select **Console Administrator**.
    6. Scroll to the end of the page and click **Add**.
    7. In the **Access summary** side panel, review the details and click **Invite**.

Limited access scenario

This scenario shows how to give an existing user permission to create and manage only virtual server instances in the Default resource group. Before the user can create an instance and associate a floating IP, the user also needs access to related resources, such as the VPC and subnet in which the instance will be created.

1. Go to the IAM Users page in the IBM Cloud console and select the user whose access you want to configure.
2. On the **Access policies** tab, click **Assign access**.
3. In the **Assign users additional access** section, select **IAM services** and complete the following tasks:
    1. From the **What type of access do you want to assign?** list, select **VPC Infrastructure Services**.
    2. From the **in** list, select **Resource group: default**.
    3. From the **Resource type** list, select **All resource types**.
    4. In the **Platform access** area, select **Editor**.
    5. Make sure that the **Resource group access** option is set to **Viewer**.
    6. In the **Service access** area, select **Console Administrator**.
    7. Scroll to the end of the page and click **Add**.
    8. Review the **Access summary** side pane, and click **Assign**.

Team access scenario

This scenario shows how an account administrator can assign authorization so that different teams have access to separate VPC resources. The example uses *resource groups* to set up separate resource access for two teams. For the purposes of this example, resources are not shared across teams.

The example takes you through the process of creating resource groups, creating access groups, and assigning the appropriate policies to provide your teams with access to separate VPC resources.

In this scenario, you're setting up two different project teams to use separate VPCs. You will assign access so that each team has access to their teams' VPC resources only.

- Your first team is a test team. You have decided to assign them access to VPCs in a resource group named `test_vpcs`.
- The second team is your production team. They are assigned access to VPCs in a resource group named `production_vpcs`.

This strategy can be used to assign separate VPC resources to any number of teams. However, all resources share the same VPC quotas for the account. For more information about quotas and limits, see VPC quotas.

Step 1: Create resource groups

Create resource groups that contain each of your teams' VPC resources.

1. Create a resource group called `test_team`.
2. Create a resource group called `production_team`.

For more information about how to create resource groups, see Managing resource groups.

By default, account administrators can create new resource groups. Other users must be assigned the **Editor** role for **All Account Management Services**, which allows them to create resource groups.

Step 2: Create access groups

Resource access can be assigned to groups of users. Groups of users with the same access permissions are called *access groups*. In this scenario, the account administrator creates an access group to represent each grouping of team members who require a specific type of VPC access, a total of four unique access groups.

Create four access groups with the following names, and assign the appropriate users to each access group:

- `test_team_manage_vpcs`
- `test_team_view_vpcs`
- `production_team_manage_vpcs`
- `production_team_view_vpcs`

For more information about how to create access groups and assign users to the access groups, see Create access groups.

Step 3: Add IAM policies to the access groups

Add the necessary VPC access policies for each access group. For example, add a policy so members of the `test_team_manage_vpcs` access group can create, update, and delete all VPC resources in the `test_team` resource group.

1. Go to the IAM Group UI in the IBM Cloud console.
2. Select an access group. Let us start with the `test_team_manage_vpcs` access group.
3. On the **Access policies** tab, click **Assign access**.
4. In the **Assign access group additional access** section, select **IAM services**
5. From the **What type of access do you want to assign?** list, select **VPC Infrastructure Services**.
6. From the **in** list, select **Resource group: test_team**.
7. From the **Resource type** list, select **All resource types**.
8. In the **Platform access** area, select **Editor**.
9. In the **Resource group access** area, select **Viewer**.
10. In the **Service access** area, select **Console Administrator**.
11. Scroll to the end of the page and click **Add**.
12. In the **Access summary** side panel, review the details and click **Assign**.

Because floating IP resources and the boot volume that is automatically attached to an instance are created in the Default resource group, you must also add access policies for the Default resource group.

Access group	Resource group	Resource type	Platform access role	Service access role
test_team_manage_vpcs	Default	Block Storage for VPC	Editor	
test_team_manage_vpcs	Default	Floating IP for VPC	Editor	

Table 1. Access policies for the default resource group

Repeat the previous steps to add access policies for the remaining three access groups.

Access group	Resource group	Resource type	Platform access role	Service access role
test_team_view_vpcs	test_team	All resource types	Viewer	
test_team_view_vpcs	Default	Block Storage for VPC	Viewer	
test_team_view_vpcs	Default	Floating IP for VPC	Viewer	
production_team_manage_vpcs	production_team	All resource types	Editor	Console Administrator
production_team_manage_vpcs	Default	Block Storage for VPC	Editor	
production_team_manage_vpcs	Default	Floating IP for VPC	Editor	
production_team_view_vpcs	production_team	All resource types	Viewer	
production_team_view_vpcs	Default	Block Storage for VPC	Viewer	
production_team_view_vpcs	Default	Floating IP for VPC	Viewer	

Table 2. Access policies for the remaining access groups

The teams are now set up to use VPCs. Members of the `test_team_manage_vpcs` and `production_team_manage_vpcs` access groups can now create VPCs in their assigned resource groups (that is, in the `test_team` and `production_team` resource groups).

When you create a VPC or other resources, make sure that you specify the resource group in which to create the resource. If you don't specify a resource group, the resource is created in the Default resource group.

Viewing user's permissions

Policies can be viewed in the user's **Access policies** tab.

Use the following CLI commands to validate the resource group permissions that are assigned to your user, by policy or by access group:

- Validate by policy:

  ```
  ibmcloud iam user-policies <username>
  ```

- Validate by access group:

  ```
  ibmcloud iam access-groups -u <username>
  ```

Changes to IAM access policies for VPC can take up to 10 minutes to take effect.

To learn more about the access management on IBM Cloud, continue your reading in IBM Cloud docs:

- Managing identity and access
- Managing users and access
- What is IAM

Available regions

A region is a specific geographical location where you can deploy apps, services, and other IBM Cloud™ resources.

Available regions for z/OS virtual server instances

You can create z/OS virtual server instances in the following IBM Cloud regions:

Location	Region	z/OS stock images and custom images	API endpoints
Brazil (São Paulo)	`br-sao`	✔	`br-sao.iaas.cloud.ibm.com`
Canada (Toronto)	`ca-tor`	✔	`ca-tor.iaas.cloud.ibm.com`
United Kingdom (London)	`eu-gb`	✔	`eu-gb.iaas.cloud.ibm.com`
Japan (Tokyo)	`jp-tok`	✔	`jp-tok.iaas.cloud.ibm.com`
US East (Washington DC)	`us-east`	✔	`us-east.iaas.cloud.ibm.com`
US South (Dallas)	`us-south`	✔	`us-south.iaas.cloud.ibm.com`
Spain (Madrid)	`eu-es`	✔	`eu-es.iaas.cloud.ibm.com`

Table 1. IBM Cloud regions for z/OS virtual server instances

Available regions for dedicated hosts and groups

You can create s390x dedicated hosts and dedicated host groups in the following IBM Cloud regions:

Location	Region	s390x dedicated hosts	API endpoints
Spain (Madrid)	`eu-es`	✔	`eu-es.iaas.cloud.ibm.com`
US South (Dallas)	`us-south`	✔	`us-south.iaas.cloud.ibm.com`

Table 2. IBM Cloud regions for s390x dedicated hosts

For more information about IBM Cloud regions, see Region and data center locations for resource deployment.

Managing custom images

After you import custom images, you can manage the images from the **Custom images** page in the IBM Cloud console.

You can rename a custom image, create a new virtual server from the image, copy the UUID of the image, or delete the image.

When you delete an image from IBM Cloud VPC, the corresponding deployment record on the **Deployments** page in the Wazi Image Builder UI is not removed.

To manage a custom image, complete the following steps:

1. In IBM Cloud console, navigate to **Menu icon** > **VPC Infrastructure > Compute > Images**.
2. On the **Custom images** tab, click the Actions icon for a specific image and select from the available options. Encrypted custom images are identified by a lock icon after the image name.

To create z/OS virtual server instances with your custom image, see Using the IBM Cloud console to create z/OS virtual server instances.

For more information, see Managing custom images in IBM Cloud VPC.

Creating dedicated hosts and groups

You can create one or more dedicated hosts with associated dedicated host groups in your IBM Cloud™ VPC by using the IBM Cloud console, the CLI, or the API. Dedicated Host for VPC is fully integrated into IBM Cloud.

You can create a dedicated host to carve out a single-tenant compute node, free from users outside of your organization. Within that dedicated space, you can create virtual server instances according to your needs. Additionally, you can create dedicated host groups that contain dedicated hosts for a specific purpose. Because a dedicated host is a single-tenant space, only users within your account that have the required permissions can create instances on the host.

When you create a dedicated host, you are billed by the usage of the host on an hourly basis. You are not billed for the vCPU and RAM associated with instances that are running on the host. The host is owned and managed by IBM.

For Wazi as a Service, dedicated hosts are currently available in the Spain (Madrid) and US South (Dallas) region only.

For more information, see the following documentation:

- s390x dedicated host profiles
- Creating dedicated hosts and groups
- Managing dedicated hosts and groups

Using the IBM Cloud console to create z/OS virtual server instances

You can use create and configure an IBM Cloud™ Virtual Private Cloud (VPC) in the IBM Cloud console and then create a z/OS virtual server instance.

If you want to complete the following tasks by using the CLI or REST APIs, see Setting up your API and CLI environment.

Objectives

To create and configure your VPC and other attached resources, complete the following steps:

1. Create your SSH key.
2. Create a VPC and subnet to define the network. When you create your subnet, attach a public gateway if you want to allow all resources in the subnet to communicate with the public internet.
3. To limit the subnet's inbound and outbound traffic, you can configure an access control list (ACL). By default, all traffic is allowed.
4. Create a virtual server instance. By default, a 250 GB boot volume is attached to the z/OS virtual server instance.
5. If you want more storage, create a block storage volume and attach it to your instance.
6. To define the inbound and outbound traffic that is allowed for the instance, configure its security group.
7. If you want your instance to be reachable from the internet, you can create a VPN server as a gateway in the VPC or use the floating IP address for direct access. For more information, see Client-to-site VPN servers and Reserving a floating IP address.

After you enter data on the provisioning pages, select the **Get sample API call** button to view the sequence of API requests that correspond to your settings. Viewing the API calls is a good way to learn about the API and understand actions and their dependencies.

Creating your SSH key

Make sure that you have an SSH key. The key is used to connect to the virtual server instance. For example, generate an SSH key on your Linux server or Mac system by running the following command:

```
ssh-keygen -t rsa -C "user_ID"
```

This command generates two files. The generated public key is in the **`id_rsa.pub`** file under an **`.ssh`** directory in your home directory, for example, **`/Users/<USERNAME>/.ssh/id_rsa.pub`**.

For Windows systems, you can use PuTTYgen to generate an SSH key.

For more information, see SSH keys.

After creating the SSH keys, you must upload your created SSH keys to the IBM Cloud console. To add or delete SSH keys in the IBM Cloud console, complete the following steps:

1. In IBM Cloud console, navigate to **Menu icon ☰ > VPC Infrastructure > Compute > SSH keys**. Any existing SSH keys are displayed. You can use the **...** menu to copy the UUID of an SSH key or delete an SSH key.

2. To add an SSH key, click **Add SSH Key**.
3. On the Add SSH key page, enter a name for your SSH key, select a resource group, and select a region.

   You are shown a list of the available regions for your specific resource group.
4. Locate your public SSH key. It might be in an `.ssh` directory within your home directory, for example, `/Users/<USERNAME>/.ssh/id_rsa.pub`.

   The directory might contain two files with the same file name. The "public" SSH key contains the extension `.pub`. The content of the public SSH key file typically begins with `ssh-rsa` and ends with your username.
5. You can open the public SSH key file with a text editor. Then, copy and paste the entire contents of the SSH file into the **Public key** space on the form.
6. Click **Add SSH key** to create your SSH key in the IBM Cloud console. It now displays in **VPC Infrastructure > Compute > SSH keys**.

For more information, see Managing SSH keys.

Creating a VPC and subnet

To create a VPC and subnet, complete the following steps:

1. Open IBM Cloud console.
2. Click **Menu icon** > **VPC Infrastructure > Network > VPCs** and click **Create**.
3. Enter a name for the VPC, such as `my-vpc`.

   Make sure that you're creating a VPC with generation 2 compute resources.
4. Select a resource group for the VPC. Use resource groups to organize your account resources for access control and billing purposes. For more information, see Best practices for organizing resources in a resource group.
5. *Optional:* Enter tags to help you organize and find your resources. You can add more tags later. For more information, see Working with tags.
6. The process assigns a default ACL. Later in this tutorial we'll configure rules for the ACL.
7. Select whether the default security group allows inbound SSH and ping traffic to virtual server instances in this VPC. We'll configure more rules for the default security group later.
8. *Optional:* Select whether you want to enable your VPC to access classic infrastructure resources. For more information, see Setting up access to classic infrastructure.
9. *Optional:* Clear the **Default address prefixes** option if you don't want to assign default address prefixes to each zone in your VPC. After you create your VPC, you can go to its details page and set your own address prefixes.
10. By default the create VPC process defines three subnets. If you need to edit the properties that are defined for the subnet, click the pencil icon for the subnet that you want to edit. You can also remove a subnet that is pre-defined by clicking the minus icon. If you need to add a subnet, complete the following steps.
11. Click **Add subnet** and enter a name for the new subnet in your VPC, such as `my-subnet`.
12. Select a location for the subnet. The location consists of a region and a zone.

    The region that you select is used as the region of the VPC. All additional resources that you create in this VPC are created in the selected region.
13. Select a resource group for the subnet.
14. Enter an IP range for the subnet in CIDR notation, for example: `10.240.0.0/24`. In most cases, you can use the default IP range. If you want to specify a custom IP range, you can use the IP range calculator to select a different address prefix or change the number of addresses.

    A subnet cannot be resized after it is created.
15. Attach a public gateway to the subnet if you want to allow all attached resources to communicate with the public internet.

You can also attach the public gateway after you create the subnet.

16. Click **Create virtual private cloud**.

To create additional subnets in this VPC, click the **Subnets** tab and click **New subnet**. When you define the subnet, make sure to select `my_vpc` in the **Virtual private cloud** field.

Configuring the access control lists (ACL)

You can configure the ACL to limit inbound and outbound traffic to the subnet. By default, all traffic is allowed.

Each subnet can be attached to only one ACL. However, each ACL can be attached to multiple subnets.

To configure the ACL, complete the following steps:

1. In the navigation pane, click **Network > Subnets**.
2. Click the subnet that you created.
3. In the **Subnet details** area, click the name of the ACL.
4. Click **Add rule** to configure inbound and outbound rules that define what traffic is allowed in or out of the subnet. For each rule, specify the following information:
   - Select whether to allow or deny the specified traffic.
   - Select the protocol to which the rule applies.
   - For the source and destination of the rule, specify the IP range and ports for which the rule applies. For example, if you want all inbound traffic to be allowed to the IP range `192.168.0.0/24` in your subnet, specify **Any** as the source and `192.168.0.0/24` as the destination. But if you want to allow inbound traffic only from `169.168.0.0/24` to your entire subnet, specify `169.168.0.0/24` as the source and **Any** as the destination for the rule.
   - Specify the rule's priority. Rules with lower numbers are evaluated first and override rules with higher numbers. For example, if a rule with priority 2 allows HTTP traffic and a rule with priority 5 denies all traffic, HTTP traffic is still allowed.

   See Configuring rules for an example.
5. When you finish creating rules, click the **Access control lists** breadcrumb at the beginning of the page.

For more information, see About access control lists.

Creating a z/OS virtual server instance

You can create one or more virtual server instances in your IBM Cloud VPC by using the IBM Cloud console.

To create a virtual server instance, start with following steps:

1. In the IBM Cloud console, go to **Menu icon** ☰ **> VPC Infrastructure > Compute > Virtual server instances**.
2. Click **Create**.
3. Select or enter the information in Table 1.

Field	Value
Location	Locations are composed of regions (specific geographic areas) and zones (fault-tolerant data centers within a region). Select an available region where you want to create your virtual server instance.
Name	A name is required for your virtual server instance.
Resource group	Select a resource group for the instance.
Tags	You can assign a user tag to the instance so that you can easily filter instance resources in your resource list. For more information, see Working with tags.
Access management tags	Access management tags help you apply flexible access policies on specific resources. For more information, see the Controlling access to resources by using tags UI tutorial.
Table 1. Selections to begin instance creation	

Then, complete the following steps:

1. Click **Change image**, and select an image (that is, operating system and version) for the instance. If the geographic location where you are creating an instance supports it, you have the option to select *s390x* architecture.

You can use the *z/OS stock image* with the latest program temporary fixes (PTFs). You can also select *Custom images* to use the custom image and data volume snapshot that you created with Wazi Image Builder. The image is called **`wazi-custom-image`** by default.

All operating system images use cloud-init that you can use to enter user metadata that is associated with the instance for post provisioning scripts. Metadata isn't supported for z/OS virtual server instances.

2. Click **Change profile** to select the vCPU and RAM combinations for the instance. For more information, see s390x instance profiles.

   z/OS virtual server instances require a minimum profile of 2 vCPUs x 16 GB RAM (2x16). One vCPU of the selected profile is reserved for running the service. When you select the profile for any z/OS stock images with RAM smaller than 8 GB, you might encounter the IAR057D message.

3. Select an existing SSH key or add an SSH key that is to be used to access the virtual server instance. To add an SSH key, click **New key** and name the key. After you enter your previously generated public key value, click **Add SSH key**. You can select multiple SSH keys if you want to allow those SSH key owners to access your instance.

   When you create a z/OS virtual server instance from z/OS stock image or a Wazi aaS custom image, the SSH keys that are selected in this step are ignored. When the instance is running and z/OS is successfully IPLed, you can SSH into the VSI by using an SSH key that was configured into the snapshot or custom image from which the new instance was created.

4. Check the boot volume. In the current release, 250 GB is allotted for the boot volume for a z/OS virtual server instance. You can click the pencil icon to edit the details of the boot volume and use Hyper Protect Crypto Services for the Encryption option. When you enable Auto-delete for the boot volume, the boot volume is deleted automatically if the instance is deleted.

5. *Optional:* In the **Data volumes** section, click **Create** to attach a block storage volume to your instance if you want more storage.

   When you create a z/OS virtual server instance with the custom image, you must add a data volume by clicking **Create** and selecting **`Import from Snapshot`**. The snapshot is called **`wazi-custom-image-data`** by default. You can select **Hyper Protect Crypto Services** for the Encryption option.

6. In the **Networking** section, specify the IBM Cloud VPC where you want to create your instance.

7. *Optional:* In the **Network interfaces** section, you can click the pencil icon to edit the details of the network interface. You can click **Create** to include more network interfaces depending on the vCPU count that is included in the instance profile.

   You can also select which security groups to attach to each interface. By default, the VPC's default security group is attached. The default security group allows inbound SSH and ping traffic, all outbound traffic, and all traffic between instances in the group. All other traffic is blocked. You can configure rules to allow more traffic. If you later edit the rules of the default security group, those updated rules will apply to all current and future instances in the group.

   Each interface has a maximum network bandwidth of 16 Gbps. If the profile you selected for this instance has a maximum network bandwidth greater than 16 Gbps, you might want to create more interfaces to optimize network performance.

8. In the **Advanced options** section, you can choose to complete more instance configurations in Table 2.

Field	Value
User data	You can add user data that automatically performs common configuration tasks or runs scripts. User data is not supported for z/OS virtual server instances.
Metadata	Disabled by default. This setting informs the instance to collect the instance configuration information and user data. Metadata is not supported for z/OS virtual server instances.
Add to dedicated host	This setting is disabled by default. To create the virtual server instance in a single-tenant space, click the toggle to enable the dedicated host. For more information, see Creating dedicated hosts and groups.
Add to placement group	Placement groups are disabled by default. Click the toggle to enable placement groups. Then, select or create a placement group for the instance. If you add a placement group, the instance is placed according to the placement group policy. For more information, see About placement groups.
Host failure auto restart	This setting is enabled by default. To disable host failure auto restart, click the toggle. For more information, see Host failure recovery policies.
Table 2. Advanced options	

Finally, click **Create virtual server instance**. The status of the instance starts as *Pending*, changes to *Stopped*, and then *Running*. You might need to refresh the page to see the change in status.

For more information, see Creating virtual server instances.

Creating and attaching a block storage volume

You can create a block storage volume and attach it to your virtual server instance if you want more storage.

To create and attach a block storage volume, see the following topics:

- Creating Block Storage for VPC volumes
- Attaching a Block Storage for VPC volume

To verify the newly attached block storage volume with its address assigned, you can find the information on your z/OS virtual server instance console via a broadcast message sent to you with the affected device address. See the following message as an example.

```
``` {: screen}
IKJ56455I IBMUSER LOGON IN PROGRESS AT 15:19:05 ON JUL
Preparing attached block storage vde of size 55G
Attached block storage vde on address DD60
READY
```
```

Configuring the security group for the instance

You can configure the security group to define the inbound and outbound traffic that is allowed for the instance. For example, after you configure ACL rules for the subnet based on your company's security policies, you can further restrict traffic for specific instances depending on their workloads.

To configure the security group, complete the following steps:

1. In the navigation pane, click **Compute > Virtual server instances**.
2. Click your instance to view its details.
3. In the **Network interfaces** section, click the name of the security group.
4. On the **Rules** tab of the security group, click **Manage rules**.
5. Click **Create** to configure inbound and outbound rules that define what type of traffic is allowed to and from the instance.

   For each rule, specify the following information:

   - Specify a CIDR block or IP address for the permitted traffic. Alternatively, you can specify a security group in the same VPC to allow traffic to or from all instances that are attached to the selected security group.
   - Select the protocols and ports to which the rule applies.

   Tips:

   - All rules are evaluated, regardless of the order in which they're added.
   - Rules are stateful, which means that return traffic in response to allowed traffic is automatically permitted. For example, you created a rule that allows inbound TCP traffic on port `80`. That rule also allows replying outbound TCP traffic on port `80` back to the originating host, without the need for another rule.
   - If your z/OS virtual server instance is created by using the z/OS dev and test stock image, you can refer to Reserved configurations when adding additional ports into the security group of your instance.
6. *Optional:* To view interfaces that are attached to the security group, click **Attached resources** in the navigation pane.
7. When you finish creating rules, click the **Security groups** breadcrumb at the beginning of the page.

For more information, see About security groups.

Creating a client-to-site VPN server for VPC

You can create a VPN server from the internet to connect to the z/OS virtual server instance in the VPC network, such as a laptop. This solution helps you to connect to IBM Cloud resources through secure and encrypted connections.

To create the VPN server and set up a client VPN environment for the z/OS virtual server instance in the VPC, follow these steps:

1. Review Planning considerations for VPN servers.
2. Complete all prerequisites in Before you begin.

3. Provision a stand-alone VPN server in a subnet, or provision a high availability VPN server in the two subnets. For instructions, see Creating a VPN server.
4. Create VPN routes.
5. Set up a VPN client environment and connect to the VPN server.

Connecting to z/OS virtual server instances

A z/OS virtual server instance is typically a private backend infrastructure component that must never be directly accessed from the outside world. Even environments that are used for development, testing, or demonstration must follow good practices for logical isolation.

To learn how to use subnets to securely isolate and control traffic to backend infrastructure such as a z/OS virtual server instance, you can refer to the following tutorials:

- Public front end and private back end in a Virtual Private Cloud
- Securely access remote instances with a bastion host
- About client-to-site VPN servers

After you connected to the VPC network by using the client-to-site VPN server, you can access the z/OS virtual server instance by using the private IP address.

Before you begin

Make sure that you locate the private IP address before you connect to the z/OS instances. Private IP addresses are IP addresses that are provided by the system and are only reachable within the VPC network. You can find the private IP address of your z/OS virtual server instance under the *Reserved IP address* column on the console after the instance is created successfully.

Step 1. Configuring the password

Before you can log in to a z/OS virtual server instance, you must change the password for the default user ID **ibmuser**.

If you are using your z/OS Wazi aaS custom images, you do not need to configure the password. You can safely ignore this step and then connect to the z/OS virtual server instance instead.

1. Log in to the z/OS UNIX shell environment by using your SSH private key and the default user ID. The **vsi ip address** in the following code snippets represent the private IP address of your z/OS virtual server instance.

   ```
   ssh -i <path to your private key file> ibmuser@<vsi ip address>
   ```

   You receive a response similar to the following example. When prompted to continue connecting, type **yes**.

   ```
   The authenticity of host 'xxx.xxx.xxx.xxx (xxx.xxx.xxx.xxx)' can't be established.
   ECDSA key fingerprint is SHA256:xxxxxxxxxxxxxxxxxxxxxx.
   Are you sure you want to continue connecting (yes/no)? yes
   Warning: Permanently added 'xxx.xxx.xxx.xxx' (ECDSA) to the list of known hosts.
   ```

2. Use the **tsocmd** command to configure password phrases for the z/OS instance. Replace **YOUR PASSWORD PHRASE** with your own password phrase.

   ```
   tsocmd "ALTUSER IBMUSER PHRASE('YOUR PASSWORD PHRASE') NOEXPIRE RESUME"
   ```

   You must follow the syntax rules for password phrases that are listed in Assigning password phrases.

   For more information, see the following topics:

   - tsocmd - Run a TSO/E command from the shell (including authorized commands)
   - ALTUSER (Alter user profile)

Step 2. Getting connected

You can interact with your z/OS virtual server instance with the following approaches.

Using TN3270 terminal emulator

You can use the TN3270 terminal emulator to log on to the Time Sharing Option/Extensions (TSO/E). The default port for the secured TN3270 is **992** and you need to provide a self-signed certificate-authority (CA) certificate as a parameter to have a secured connection to the 3270 emulation program. You can find your self-signed CA certificate in a file that is called **common_cacert** in the home directory of the **IBMUSER** user ID in z/OS UNIX System Services.

- For macOS, Linux, and Windows system where the Secure Copy Protocol (SCP) client is installed, you need to transfer the **common_cacert** file before you connect to the secured telnet port and run the following commands:

  ```
  scp ibmuser@<vsi ip address>:/u/ibmuser/common_cacert <my local dir>/common_cacert
  c3270 -cafile <my local dir>/common_cacert -port 992 <vsi ip address>
  ```

  Ensure that the contents of the **common_cacert** file transferred from the z/OS VSI to the VSI client are in PEM format, a base64 ASCII encoded certificate with a header and footer that reads as follows:

  ```
  "-----BEGIN CERTIFICATE-----" and "-----END CERTIFICATE-----"
  ```

  If the contents of the file are not in the preceding format, then transfer the file to the z/OS VSI client by using the following **scp** command with the -O option issued from the client machine:

  ```
  scp -O ibmuser@<vsi ip address>:/u/ibmuser/common_cacert <my local dir>/common_cacert
  ```

- For Windows, you need to import the self-signed CA cert into Windows first and then create the IBM Personal Communications (PCOMM) to read the CA cert from the truststore in Windows. Complete the following steps:

  1. Transfer the **common_cacert** file from the z/OS system to your workstation.
  2. Open **Internet Options** in your workstation.
  3. Under **Content**, select **Certificates**.
  4. Select **Import** to open the Certificate Import Wizard page and click **Next**.
  5. Click **Browse** to select the **common_cacert** file and click **Next**.
  6. Under the Certificate Store page, select **Place all certificates in the following store** and click **browse** to select the **Trusted Root Certificate Authorities**, and then click **Next**.
  7. On the Completing the Certificate Import Wizard page, click **Finish**.
  8. After you transfer the CA cert to the truststore successfully, you need to create a secure session in PCOMM. Under the **Host Definition** tab of the **Link Parameters** configuration, enter the IP address or the hostname of your z/OS virtual server instance with port **992**.
  9. Under the **Security Setup** tab of the **Link Parameters** configuration, check the **Enable Security** box.

  The default user ID is **IBMUSER** and the password is the one you configured in the previous step. Then, you can interact with the z/OS in the TSO native mode, by using the Interactive System Productivity Facility (ISPF), or by using z/OS UNIX shell and utilities. For more information, see Interacting with z/OS: TSO, ISPF, and z/OS UNIX interfaces.

Important:

- The unsecured port **23** for 3270 connection is closed. You must use the secured port **992**.
- The VSI server certificate only contains the private IP address information of the z/OS virtual server instance.
- Optionally, you can use the IP address of the VSI as part of the **accepthostname** argument when connecting over a floating IP address. For example:

  ```
  c3270 -cafile <my local dir>/common_cacert -port 992 -accepthostname <vsi ip address>  <floating ip>
  ```

Using IBM Host On-Demand

If you want to import the CA certificate by using IBM Host On-Demand, run the following commands:

1. Download the certificate file from the z/OS system.

   ```
   scp ibmuser@<vsi ip address>:/u/ibmuser/common_cacert ./Downloads/common_cacert
   ```

2. Import the downloaded certificate. Use a recognizable alias.

```
keytool -importcert -alias <alias> -file ./Downloads/common_cacert -keystore
/Applications/HostOnDemand/lib/CustomizedCAs.jks -storepass hodpwd
```

3. Check what certificates have been imported.

```
keytool -list -keystore /Applications/HostOnDemand/lib/CustomizedCAs.jks -storepass hodpwd
```

Using the web browser to access z/OSMF

You can use the web browser to access the IBM z/OS Management Facility (z/OSMF). For example, access the url **`https://<vsi ip address>:10443/zosmf/LogOnPanel.jsp`**.

For more information about the z/OSMF, see IBM z/OS Management Facility.

When you launch z/OSMF, browser security warnings are displayed because z/OS virtual server instances are created with TLS certificates that are signed by an internal self-signed root certificate.

Using serial console from IBM Cloud UI

You can use the IBM Cloud UI to connect to a serial console and monitor the progress of the IPL within the console.

To connect to the serial console, you need to be assigned **`Operator`** (or greater) and **`Console Administrator`** roles for the virtual server instance in IBM Cloud Identity and Access Management (IAM). Otherwise, the serial console is disabled.

Follow these steps to connect to a console by using IBM Cloud UI.

1. In the IBM Cloud console, go to **Menu icon** ☰ **> VPC Infrastructure > Compute > Virtual server instance**.
2. In the **Virtual server instances for VPC** list, click the overflow button of the instance that you need to access, then click **Open serial console**. Alternatively, on the instance details page, click **Action** on the upper right then click **Open serial console**.
3. If the serial console is being used, you will be prompted to confirm whether to force open a session. This will disconnect the other user's session.
4. Enter the credentials following the prompts to log in to your instances.
5. Use the **Ctrl + L** key combination to open a new z/OS Master Console, where you can issue four commands **`start`**, **`stop`**, **`ipl`**, and **`oprmsg`** for z/OS virtual server instance operations.
6. You can then issue **`<command> help`** to get more help for each of the commands. For more instructions on the commands, see Operating with the serial console.

For more information, see Accessing virtual server instances by using VNC or serial consoles. The VNC console is not supported on z/OS virtual server instances.

Using SSH private key through a floating IP address

You can use the SSH private key through a floating IP address to connect the z/OS UNIX shell environment.

However, you might want to unbind your floating IP from your z/OS virtual server instance for security considerations and use a bastion host or client-to-site VPN server for all connections. The default secured port is 22.

For more information about the floating IP address, see Reserving a floating IP address

If you want to access another z/OS virtual server instance that is created by using a different SSH key, you must ask the instance owner to add your SSH public key into the **`authorized_keys`** file on that z/OS virtual server instance. For more information about the **`authorized_keys`** file, see Format of the authorized_keys file and Configuring new users to access your z/OS virtual server instance.

If your z/OS virtual server instance is created by using the z/OS dev and test stock image, you can refer to the Reserved configurations table when adding additional ports into the security group of your instance.

Step 3. Configuring new users to access your z/OS virtual server instance

To allow others to access your z/OS virtual server instance by using their own z/OS user IDs and own SSH keys after the instance is created, you must create each user profile and add each SSH public key into the **`authorized_keys`** file on the z/OS virtual server instance.

1. You can use one of the following methods to create a new z/OS user ID in the RACF (Resource access control facility):

- Issuing the **ADDUSER** command.
- Enrolling the user through the TSO/E Information Center Facility (ICF) panels. For more information about administering the Information Center Facility, see z/OS TSO/E Administration.

Here is an example of using the **ADDUSER** command to create a user profile. Suppose you want to create a user profile for user Steve H., a member of Department A. You want to assign the following values:

- **STEVEH** for the user ID
- **DEPTA** for the default connect group
- **DEPTA** for the owner of the STEVEH user profile
- **R3I5VQX** for the initial password
- **Steve H.** for the user's name

Steve H. does not require any of the user profile segments except TSO. The TSO segment values that you want to set to start with are **123456** for the account number and **PROC01** for the logon procedure.

To create a user profile with these values, you can run the following commands:

```
ADDUSER STEVEH DFLTGRP(DEPTA) OWNER(DEPTA) NAME('Steve H.')
    PASSWORD(R315VQX) TSO(ACCTNUM(123456) PROC(PROC01))
```

2. Then, you can use one of the following options to add new SSH public keys into the **authorized_keys** file.

   - Use the linux command such as **ssh-copy-id -i <new-ssh-public-key-file> ibmuser@<vsi ip address>** to upload new SSH public keys on your local file system into the z/OS virtual server instance. For more information, see ssh-copy-id for copying SSH keys to servers
   - Log in to the z/OS virtual server instance, use the z/OS Unix System shell command such as **cat <new-ssh-public-key-file> >> ~/.ssh/authorized_keys** to concatenate the new SSH public key into the **authorized_keys** file. For more information, see cat - Concatenate or display files
   - Use the Zowe CLI to connect to the z/OS virtual server instance, then create a CLI profile and add new SSH public keys into the **authorized_keys** file. For more information about the Zowe CLI profiles, see Creating Zowe CLI profiles

For more information about the **authorized_keys** file and z/OS user profiles, see Format of the authorized_keys file and Summary of steps for defining users.

Next steps

After you connected to your z/OS virtual server instance, you can manage your instance and monitor your instance.

Operating with the serial console

You can interact with your z/OS virtual server instance by using the serial console from the IBM Cloud UI.

To connect to the serial console, you need to be assigned **Operator** (or greater) and **Console Administrator** roles for the virtual server instance in IBM Cloud Identity and Access Management (IAM). Otherwise, the serial console is disabled.

Follow the instructions in Connecting to z/OS virtual server instances to open the serial console.

Then, you can enter the **help** command to see the commands that you can issue and the keyboard guidance.

You can enter the following commands to modify z/OS workloads.

Command	Function
getlog	Compresses and uploads the logs from VSI to Cloud Object Storage (COS)
ipl	Loads a specific z/OS workload to main memory
oprmsg	Issues the operator communication
start	Starts the Control Program (CP) that was in the stopped state
stop	Stops the Control Program (CP)

Table 1: Available commands

You can then issue **<command> help** to get more help for each of the commands.

The `getlog` command

The **getlog** command compresses and uploads the logs from VSI to COS.

Here is the command syntax:

```
getlog [access_key_id] [secret_access_key] [endpoint_url] [bucket_name]
```

- **access_key_id** is the COS hash-based message authentication code access key id created in service credentials.
- **secret_access_kdy** is the COS hash-based message authentication code access key created in service credentials.
- **endpoint_url** is the URL endopoint for COS.
- **bucket_name** is the name of the bucket where the file is uploaded.

The **getlog** command returns a success message for the file upload or an error message explaining the file upload failure.

The ipl command

The **ipl** command starts the process of loading an operating system or a stand-alone utility program.

Here is the command syntax:

```
ipl device-number [parm parm-value] [gprparm xxxx] [clear]
```

- **device-number** refers to a device number (“address”) in the devmap that contains the initial load program for the operating system.
- **parm-value** is a string of up to 8 characters that provides more information about the operating system that is loaded.
- **gprparm** provides a string of characters that are inserted into the general-purpose registers (as EBCDIC characters that use 32-bit registers), starting with register 0, placing 4 characters in each register. The keyword can be entered as gpr_parm. The gprparm function was carried forward from much earlier systems and has no known use today.
- **clear** causes IBM Z system memory to be zeroed before loading the operating system.

The return values are as follows:

- 0: IPL function started.
- 16: Unable to initialize the manual operations interface.
- 99: The device number is not valid.

The oprmsg command

The **oprmsg** command provides input to IBM Z operating systems through the SCLP operator message interface. (This interface is also known as the HMC or the hardware console.)

Here is the command syntax:

```
oprmsg {text}
```

{text} is the message that is sent to the IBM Z operating system. If it contains any special characters (such as parentheses), the message should be enclosed in single quotation marks.

This command is used when you shut down a z/OS virtual server instance.

You can also issue **oprmsg 'D A,L'** to get the status of the z/OS subsystems.

The return values are as follows:

- 0: The message was sent to the SCLP operator interface.
- 12: No input text was found.
- 16: Unable to initialize the manual operation interface.
- 32: Unable to initialize the SCLP message interface.

The start command

The **start** command starts a CP that was in the stopped state (due to an earlier **stop** command). ISV zPDT must be operational when using this command.

Here is the command syntax:

```
start [CP-number]
```

CP-number is the target CP number.

The return values are as follows:

- 0: Operation is complete.
- 12: CP number is invalid.
- 16: Unable to initialize the manual operation interface.

The `stop` command

The **`stop`** command places a CP in the stopped state. The CP can be restarted with a **`start`** command or a reset function.

Here is the command syntax:

```
stop [CP-number]
```

CP-number is the target CP number.

Generally, a CP is stopped to display register or memory contents. In rare cases, it might be stopped to halt the process of an application or operating system function.

The return values are as follows:

- 0: Operation is complete.
- 12: CP number is invalid.
- 16: Unable to initialize the manual operation interface.

Managing z/OS virtual server instances

You can view and manage your z/OS virtual server instances from the **Virtual server instances** page in IBM Cloud console.

You can stop, start, reboot, and delete the z/OS virtual server instances.

To manage your instances, complete the following steps:

1. In IBM Cloud console, navigate to **Menu icon** > **VPC Infrastructure** > **Compute** > **Virtual server instances**.
2. On the **Virtual server instances** page, click the Actions icon for the instance that you want to manage. You can select from the following actions:

Action	Description
Rename	Change the name of the instance.
Stop	Stop the instance.
Start	Start an instance that is stopped. This action is not available if the instance status is Running.
Reboot	Immediately powers off a running instance and then powers it back on again.
Resize	Vertically scale virtual server instances to any supported profile size.
Delete	To delete an instance, the instance must have a powered off status. If the instance has a floating IP address, it must be unassociated or released before the instance is deleted. The delete action permanently removes an instance and its connected vNIC, boot volume, and data from your account.

Table 1. Actions available for z/OS virtual server instances

Renaming a z/OS virtual server instance

From the **Virtual server instances** page in IBM Cloud console, click **Rename**. Enter the new name for the z/OS virtual server instance. Click **Rename**.

Stopping and starting a z/OS virtual server instance

First, you must shut down all the subsystems in the z/OS system and then the virtual server instance is halted. For more information, see Shutting down z/OS virtual server instances.

This change frees the instance resources that were being consumed. The virtual server instance is transitioned to the Stop state. If the instance is stopped, the instance remains in the stopped state and must be started manually. Billing is suspended for some compute

resources while the instance is stopped. You cannot interact with an instance if it is stopped, but volumes remain provisioned. If the instance is started, normal interaction and billing continue.

A Force stop action triggers a power cycle reset of the virtual server instance.

The start action starts a z/OS virtual server instance that is in a stopped state.

From the **Virtual server instances** page in IBM Cloud console, click **Stop** or **Start**.

Rebooting a z/OS virtual server instance

The reboot action triggers a guest operating system reboot. The z/OS virtual server instance remains in a running state while the guest operating system is rebooting. Billing continues.

A Force reboot action triggers a power cycle reset of the z/OS virtual server instance.

From the **Virtual server instances** page in IBM Cloud console, click **Reboot**.

Creating an image from a z/OS virtual server instance

You can create an image from a virtual server instance boot volume. The image is a full copy of the source volume. It includes the operating system and any user data. Data volumes are not supported.

From the **Virtual server instances** page in IBM Cloud console, click **Create image**.

To create an image of the boot volume associated with an instance, the virtual server instance must be stopped.

For more information, see About creating an image from a volume and Creating an image from a volume.

Resizing a virtual server instance

You can increase or decrease the amount of vCPU and RAM available for greater flexibility in workload management to address resource requirement changes, optimize cost or workload performance. Once resizing an instance is complete, you are billed the hourly rate of the new instance profile selected.

z/OS virtual server instances require a minimum profile of 2 vCPUs x 16 GB RAM (2x16). One vCPU of the selected profile is reserved for running the service.

To resize a virtual server instance, the instance must be stopped. For more information, see Resizing a virtual server instance.

Deleting a z/OS virtual server instance

From the **Virtual server instances** page in IBM Cloud console, click **Delete**.

The delete action permanently removes an instance and its connected vNIC, and data from your account. The instance boot volume is also deleted if it was configured to be deleted when the attached instance is deleted. If the instance has one or more attached data volumes, those volumes are preserved unless the default setting is changed to auto-delete. After you confirm the delete action, the process to delete the instance and its associated vNIC, boot volume, and data begins. The delete action can take up to 30 minutes, but when the process is complete, the instance no longer appears on the virtual server instances page.

Viewing instance details

You can view a summary of all instances on the **Virtual server instances** page. You can access the details page for an instance clicking an individual instance name to view details and make changes.

From the instance details page, you can also view the associated network interface, access its subnet, toggle the auto-delete setting, and reserve or delete a floating IP address.

Retrieving the instance identifier

When a virtual server instance is created, the instance is automatically assigned an instance identifier (ID), which includes the SMBIOS system-uuid as a portion of the ID. The ID can be up to 64 bytes and it consists of digits, lowercase letters, underscores, and dashes.

IDs are immutable, globally unique, and never reused, so the ID uniquely identifies a particular instantiation of a virtual server instance across all of IBM Cloud. The ID, including the SMBIOS system-uuid portion, is static and persists for the lifecycle of the virtual server instance until that virtual server instance is deleted.

From the instance details page, you can view and copy the **Virtual server instance ID**.

You can also SSH into the instance, and then run the **`DISPLAY IPLINFO`** command. When the IEE254I message is displayed, the ID is included in the **`VM EXT NAME`**.

For more information, see Managing virtual server instances.

Monitoring your z/OS virtual server instance

You can monitor the CPU, volume, memory, and network usage of your instance over time. Metrics are retained in the system for two weeks.

To monitor your instance, complete the following steps:

1. Click **Virtual server instance** in the navigation pane.
2. Click the name of your instance.
3. Click **Monitoring** in the navigation pane.

Memory percentage of the virtual server instance represents the full allocation of memory that is dedicated to z/OS, rather than the actual memory usage within z/OS.

Because the monitoring data is stored in IBM Cloud Monitoring, you must be authenticated to an instance of IBM Cloud Monitoring in your account. For more information, see IBM Cloud Monitoring.

Using IBM Cloud DNS services with VPC

IBM Cloud™ DNS Services provides private DNS to Virtual Private Cloud (VPC) users. Private DNS zones are resolvable only on IBM Cloud, and only from explicitly permitted networks in an account. To get started, create a DNS Services instance by using the IBM Cloud console.

DNS Services allow you to:

- Create private DNS zones that are collections for holding domain names.
- Create DNS resource records under these DNS zones.
- Specify access controls used for the DNS resolution of resource records on a zone-wide level.

DNS Services also maintains its own worldwide set of DNS resolvers. Instances that are provisioned under IBM Cloud on an IBM Cloud network can use resource records that are configured through IBM Cloud DNS Services by querying DNS Services resolvers.

Resource records and zones that are configured through DNS Services are:

- Separated from the wider, public DNS and their publicly accessible records.
- Hidden from machines outside of and not part of the IBM Cloud private network.
- Accessible only from machines that you authorize on the IBM Cloud private network.
- Resolvable only via the resolvers provided by the service.

To use DNS Services, you must have at least one virtual server instance in a VPC in the IBM Cloud. If you do not have one, learn how to get started with IBM Virtual Private Cloud.

While the private DNS resolvers are required to resolve private DNS names, they also resolve public DNS names if the request is for a name that is not defined to be in a private DNS zone.

For more information, see Getting started with DNS Services.

Installing software products for z/OS virtual server instances

You can install software products for your z/OS virtual server instances. The general software installation process for the z/OS virtual server instance is the same as the on-premises process. However, you need to upload specific certificates from your workstation to the z/OS virtual server instance and import them into Resource Access Control Facility (RACF) before you follow the on-premises standard installation process.

Before you begin

1. Make sure that you have the Shopz customer number to request the service certificate. For more information, see Shopz.
2. Make sure that you can connect to your z/OS virtual server instance through the public outbound connectivity. For example, you can use the SSH private key through a floating IP address to connect the z/OS UNIX shell environment or enable the public gateway for the VPC subnet. For more information, see Connect to z/OS virtual server instances and Creating a client-to-site VPN server for VPC.

Procedures

To install software products for the z/OS virtual server instance, complete the following steps:

1. Request a service certificate from Shopz and download the certificate to your workstation. For more information, see Obtaining a user certificate.

   You need to specify the password for the certificate.

2. Download the **DigiCert Global Root CA** to your workstation.

3. After you download these certificates to your workstation, you need to upload them to the target z/OS virtual server instance. For example, you can follow the commands by using the SSH File Transfer Protocol (SFTP).

```
sftp ibmuser@<vsi ip address>
put DigiCertGlobalRootCA.crt /u/ibmuser/digicert_ca
put ShopzCert.pfx /u/ibmuser/smpe_cert
```

4. Import these certificates to a new key ring in the RACF database. You can use the following settings and must replace the following **temp4pass** with the password that is specified on Shopz.

```
//SMPECERT JOB CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID.,
//         REGION=0M
//ALLOC    EXEC PGM=IEFBR14
//ROOT     DD DSN=IBMUSER.DIGICERT.ROOT.CERT,
//         DISP=(NEW,CATLG),UNIT=SYSDA,
//         SPACE=(TRK,(1,1)),
//         DCB=(RECFM=VB,LRECL=256)
//CLIENT   DD DSN=IBMUSER.SMPE.CLIENT.CERT,
//         DISP=(NEW,CATLG),UNIT=SYSDA,
//         SPACE=(TRK,(1,1)),
//         DCB=(RECFM=VB,LRECL=256)
//*
//OGET     EXEC PGM=IKJEFT01
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
  OGET '/u/ibmuser/digicert_ca' 'IBMUSER.DIGICERT.ROOT.CERT' BINARY
  OGET '/u/ibmuser/smpe_cert'   'IBMUSER.SMPE.CLIENT.CERT'   BINARY
/*
//ADDCERT  EXEC PGM=IKJEFT01
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
  RACDCERT ADD('IBMUSER.DIGICERT.ROOT.CERT') +
    CERTAUTH TRUST +
    WITHLABEL('DigiCert Global Root CA')

  RACDCERT ADD('IBMUSER.SMPE.CLIENT.CERT') +
    ID(SYSTEM) TRUST +
    WITHLABEL('SMPE Client Certificate') +
    PASSWORD('temp4pass')
/*
//KEYRING  EXEC PGM=IKJEFT01
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
  RACDCERT ADDRING(SHOPZ_RING) ID(SYSTEM)
```

```
  RACDCERT ID(SYSTEM) CONNECT(CERTAUTH +
    LABEL('DigiCert Global Root CA') RING(SHOPZ_RING) +
    USAGE(CERTAUTH))

  RACDCERT ID(SYSTEM) CONNECT(ID(SYSTEM) +
    LABEL('SMPE Client Certificate') RING(SHOPZ_RING) +
    USAGE(CERTAUTH))

  SETROPTS REFRESH RACLIST(DIGTCERT DIGTRING)
/*
```

5. Then, you can download software products from Shopz by using **HTTPS**. You need to replace the **SERVINFO DD** block statement with the sample job that is provided by Shopz.

```
//HTTPSGET JOB CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID.,
//         REGION=0M,TIME=1440
//SMPER1   EXEC PGM=GIMSMP,PARM='PROCESS=WAIT'
//SMPCSI   DD DISP=SHR,DSN=SMPE.PROGPROD.CSI
//SMPNTS   DD PATHDISP=KEEP,PATH='/tmp/'
//SMPOUT   DD SYSOUT=*
//SMPRPT   DD SYSOUT=*
//SMPLIST  DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SMPCNTL  DD *
  SET BOUNDARY (GLOBAL) .
  RECEIVE
        FROMNETWORK(
          SERVER(SERVINFO)
      /* TRANSFERONLY */
          CLIENT(MYCLIENT)
          )
          .
/*
//SERVINFO DD *
      <SERVER
        host="deliverycb-bld.dhe.ibm.com"
        user="<userid>"
        pw="<password>"
        >
        <PACKAGE
          file="<server_dir_name>/PROD/GIMPAF.XML"
            hash="<hash_value>"
          id="<order_id>"
          >
        </PACKAGE>
      </SERVER>
/*
//MYCLIENT DD *
  <CLIENT
    downloadmethod="https"
    downloadkeyring="javatruststore"
    javahome="/usr/lpp/java/current"
    classpath="/usr/lpp/smp/classes"
    >
  </CLIENT>
/*
```

6. Now you can follow the on-premises installation process for different software products in their IBM Program Directories. For example, for COBOL V6.4, you can access the sample installation jobs by performing an SMP/E RECEIVE and then copy the jobs from the RELFILES to a work data set for editing and submission. The following sample installation jobs are provided as part of the product:

    - IGYWSMPE: SMP/E zone definition
    - IGYWALLOC: Product data sets allocation
    - IGYWZFS: Product zFS allocation
    - IGYISMKD: zFS directory allocation
    - IGYWDDEF: SMP/E DDDEFs
    - IGYWAPLY: SMP/E APPLY
    - IGYWIVP1: IVP #1
    - IGYWIVP2: IVP #2
    - IGYWACPT: SMP/E ACCEPT
7. If the installation verification procedure (IVP) runs correctly, the software product installation is completed.

Installing APARs or PTFs for z/OS virtual server instances

Your z/OS virtual server instance is capable to be refreshed with an update that is provided for an Authorized Program Analysis Report (APAR) or Program Temporary Fix (PTF). An update for an APAR is a temporary fix that is created for individual customers. PTFs can be used to correct related experienced problems or to avoid possible problems even if you have not experienced the related problems.

How to apply service to your z/OS virtual server instance depends on how you acquire your instance.

- If you obtain your z/OS virtual server instance from Wazi Image Builder in your own z/OS system environment, you can service the custom image by installing necessary APARs or PTFs on the source system with standard on-premises procedures. Then, you need to rebuild the custom image by using Wazi Image Builder and create a new z/OS virtual server instance.
- If you obtain your z/OS virtual server instance from the IBM-provided dev and test stock image, you can select one of the following approaches:
    - The IBM-provided z/OS stock image is updated each month with all the currently available PTFs. You can wait for next month's stock image to create a new z/OS virtual server instance.
    - If you need corrective service for your z/OS dev and test stock image and do not want to wait until the next refresh, you can install the APAR or PTF by yourself.

      You need to do the acquisition of the APAR or PTF yourself through another supported service channel if you have one. If you only have the license for the z/OS dev and test stock image, you cannot acquire the latest PTFs.

After you have acquired necessary updates, you can use System Modification Program Extended (SMP/E) or IBM z/OS Management Facility (z/OSMF) Software Update to install APARs or PTFs by yourself.

Updating your z/OS virtual server instance with individual fixes

Individual z/OS components or middleware products might need to be restarted if indicated by the **++HOLD** information for the maintenance that is being installed. It is also suggested that you make a backup before you install any maintenance. Any maintenance that updates core z/OS system libraries (**SYS1.LINKLIB**, **SYS1.LPALIB**, **SYS1.MIGLIB**, and other standard system libraries) must not be installed on a live system unless **++HOLD** information provides specific instructions.

You can complete the following instructions to apply the APAR or PTF for the z/OS dev and test stock image:

1. To transfer the APAR or PTF, you can use the following settings:

```
//TERSE    JOB CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID.,
//         REGION=4M
//STEP1    EXEC PGM=AMATERSE,PARM=PACK
//SYSPRINT DD SYSOUT=*
//SYSUT1   DD DISP=OLD,DSN=IBMUSER.UJ90039
//SYSUT2   DD DISP=(NEW,CATLG),DCB=(RECFM=FB,LRECL=1024,BLKSIZE=0),
//            DSN=IBMUSER.UJ90039.TRS,
//            SPACE=(CYL,(1000,1),RLSE),UNIT=3390,VOL=SER=STG001
```

2. To upload the APAR or PTF to the z/OS virtual server instance, you can use the following settings through sftp:

```
sftp ibmuser@<vsi ip address>
put UA#####.TRS /u/ibmuser/UA#####.TRS
```

3. To unterse the APAR or PTF on your z/OS virtual server instance, you can use the following settings:

```
//UNTERSE  JOB CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID.,
//         REGION=0M
//ALLOC    EXEC PGM=IEFBR14
//DD1      DD DSN=IBMUSER.UA#####.TRS,
//         DISP=(NEW,CATLG),UNIT=SYSDA,
//         SPACE=(CYL,(100,1)),
//         DCB=(RECFM=FB,LRECL=1024,BLKSIZE=27648)
//*
//OGET     EXEC PGM=IKJEFT01
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
  OGET '/u/ibmuser/UA#####.TRS' 'IBMUSER.UA#####.TRS' BINARY
/*
//UNTERSE  EXEC PGM=AMATERSE,PARM=UNPACK
```

```
//SYSPRINT DD SYSOUT=*
//SYSUT1   DD DISP=OLD,DSN=IBMUSER.UA#####.TRS
//SYSUT2   DD DISP=(NEW,CATLG),DSN=IBMUSER.UJ90039,
//         UNIT=SYSDA,VOL=SER=STGVS1,
//         SPACE=(CYL,(1000,1),RLSE)
```

4. Then, you can follow the standard on-premises processes to install the APAR or PTF. You can select one of the following approaches:
   - If you want to install the PTF through SMP/E, see SMP/E commands.
   - If you want to install the PTF through z/OSMF, see Software Update with z/OSMF.

Configuring additional disk volumes for z/OS virtual server instances

You can add an additional volume to a z/OS virtual server instance and designate it for use in z/OS. For example, adding an SVC dump volume to provide a representation of the virtual storage for the system when an error occurs.

Before you begin

Complete the following prerequisites:

1. Create a z/OS virtual server instance in the IBM Cloud Virtual Private Cloud (VPC) environment and make sure that the instance is accessible through 3270 connection. For more information, see Creating virtual server instances and Connecting to z/OS virtual server instances.
2. Make sure that you can edit the z/OS virtual server instance through the IBM Cloud console. For more information, see Managing z/OS virtual server instances.

Adding disk volumes

Complete the following steps to add disk volumes:

1. Attach a new block storage data volume to your z/OS virtual server instance. For more information, see Attaching a block storage volume.

   Important:

   - If your z/OS instance is created by using one of the IBM-provided z/OS stock images, the size of the new block storage volume must not exceed 55 GB. The storage volumes larger than 55 GB are considered to be Extended Address Volumes (EAV). To work around this restriction and use larger volumes, you must run the command **`SETSMS USEEAV(YES)`** on the z/OS instance before you attach a larger block storage device. If you are using EAV volumes, you must also ensure the intended use case for the additional volume supports EAV.
   - In addition to running the command **`SETSMS USEEAV(YES)`**, you also need to harden this new system setting in system **`parmlib`** in case the z/OS system is shut down and re-IPLed. Otherwise, the setting might be reverted to the default **`USEEAV(NO)`**.
2. Verify that the newly attached storage volume is attached to and ready to be used by the z/OS virtual server instance.

   A message is sent to the broadcast data set with the affected device address. See the following message as an example.

   ```
   IKJ56455I IBMUSER LOGON IN PROGRESS AT 15:19:05 ON JUL
   Preparing attached block storage vde of size 55G
   Attached block storage vde on address DD60
   READY
   ```

   You can also issue **`OPER`** and **`SE LIST`** commands sequentially to display your device address information.
3. Initialize the volume through an ICKDSF job and vary it online.
   1. If your z/OS instance is not created by using one of the IBM-provided z/OS stock images, use ISPF (3.2) and create a new data set (for example, **`IBMUSER.JCL`**) to store the INITVOL JCL. For example, the data set can be created with the following settings:

```
Space units . . . . . TRKS
Primary quantity. . . . 10
Secondary quantity. . . 1
Directory blocks. . . . 5
Record format . . . . . FB
Record length . . . . . 80
Block size  . . . . . . 3120
```

If your z/OS instance is created by using one of the IBM-provided z/OS stock images, skip this step, and use the existing **IBMUSER.JCL** data set for the JCL member to be created in the next step.

2. Using ISPF (2 or 3.4), create a new data set member (**INITVOL**) with the following JCL that references the address of the new disk (for example, **DD60**).

   - You can use the following settings for a non-SMS (Storage Management Subsystem) managed volume:

     ```
     //INITVOL  JOB CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID.,
     //         REGION=0M
     //STEP1    EXEC PGM=ICKDSF
     //SYSPRINT DD SYSOUT=*
     //SYSIN    DD *
         INIT UNITADDRESS(DD60) VOLID(SDUMP1) VERIFY(*NONE*) +
         VTOC(0,1,225) NODSEXIST
     /*
     ```

     In this example, a **VOLID** of **SDUMP1** is used because **VOLID(SDUMP1)** is one of the three volume IDs (**SDUMP1**, **SDUMP2**, and **SDUMP3**) that are preconfigured in the SDUMP settings on the standard z/OS stock image.

     If you are using the z/OS dev and test stock image, then initializing a volume specifically for SDUMP is not recommended, and you need to initialize an SMS-managed volume instead.

     The size of the VTOC depends on the number of data sets that you intend to place on the volume. A larger VTOC size might be needed if you want to allocate extended address volumes (EAVs). You can adjust the size of the VTOC depending on your use case.

   - To initialize an SMS-managed volume, you can use the following settings:

     ```
     //INITVOL  JOB CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID.,
     //         REGION=0M
     //STEP1    EXEC PGM=ICKDSF
     //SYSPRINT DD SYSOUT=*
     //SYSIN    DD *
         INIT UNITADDRESS(DD60) VOLID(USER00) NOVERIFY +
         VTOC(0,1,225) NODSEXIST STGR
     /*
     ```

     In this example, **VOLID(USERxx)**, xx can be 00 to 0F.

     Several SMS storage groups in the dev and test stock image can be used. **USERxx** volumes are placed in the default SMS pool that is associated with Storage Class SCBASE and Storage Group SGBASE.

     The size of the VTOC depends on the number of data sets that you intend to place on the volume. A larger VTOC size might be needed if you want to allocate extended address volumes (EAVs). You can adjust the size of the VTOC depending on your use case.

3. To submit the job that you are editing with the ISPF editor, first save any changes that you made, then enter the **SUBMIT** command on the command line of the edit panel.

4. Using ISPF (S or 13.14), open SDSF (Spool Search and Display Facility).

5. Using SDSF (ST), view the status of jobs. An active job that is called **INITVOL** must be running and is waiting for confirmation to initialize the volume.

6. Enter **s** next to the **INITVOL** job to view the job output log. Toward the top of the log, there is a prompt waiting for confirmation to initialize the volume.

7. Using SDSF, reply to the prompt to confirm and initialize the volume. For example,

   ```
   /R <num>,U
   ```

   The **INITVOL** job must complete and no longer be an active job. Check the job log again to confirm that the job completed successfully.

8. Using SDSF, vary online the new disk device based on the address. For example,

   ```
   /V DD60,ONLINE
   ```

   If an error message **VARY ONLINE - CU AUTHORIZATION FAILED** shows, you can safely ignore it, because z/OS virtual server instances are in an emulated environment and there are no control units.

9. Using SDSF, verify that the device is online and ready.

   ```
   /D U,DASD,ONLINE
   ```

4. *Optional:* If your z/OS instance is not created by using one of the IBM-provided z/OS stock images, run the appropriate TSO command in the z/OS instance to add the volume ID of the newly formatted volume to the list of target volumes for SDUMP if this is your intended use case. For example, SDUMP1 is the volume ID of the newly formatted volume.

   ```
   DD ADD,VOL=(SDUMP1)
   ```

   If your z/OS instance is created by using one of the IBM-provided z/OS stock images, you can skip this step because SDUMP is pre-configured to look for volumes with volume IDs: **SDUMP1**, **SDUMP2**, or **SDUMP3**. In addition, the dev and test stock image uses SMS and new system DUMPs are placed on SMS volumes by default. The **ADD** command does not persist the change. For this change to persist over future system IPLs, make the corresponding changes to the appropriate parmlib (for example, **SYS1.PARMLIB(COMMND00)**).

5. Create a z/OS UNIX System Services file system and mount it to a new directory. A new volume is not necessarily required to add a file system for a new directory because any existing space can be used if there is sufficient amount for your requirements.

   The following example assumes the use of an SMS-managed placement for the new file system. Other options of the **zfsadm** command can also be used to direct the allocation to specific volumes.

   1. You can use the following **zfsadm** command to run from the OMVS shell and define a new 5 MB zFS file system named **OMVS.IBMUSER.TEST**.

      ```
      IBMUSER :/u/ibmuser : > zfsadm define -aggregate omvs.ibmuser.test -megabytes 5 IOEZ00248I
      VSAM linear dataset omvs.ibmuser.test successfully created.
      ```

   2. To format the file system as a zFS compatibility mode aggregate, you can use the following settings:

      ```
      IBMUSER : /u/ibmuser : > zfsadm formate -aggregate omvs.ibmuser.test
      IOEZ00077I HFS-compatibility aggregate OMVS.IBMUSER.TEST has been successfully created.
      ```

   3. To mount the file system to a newly created directory, you can use the following settings:

      ```
      IBMUSER : /u/ibmuser : > mkdir /u/ibmuser/newdir
      IBMUSER : /u/ibmuser : > mount -t zfs -f OMVS.IBMUSER.TEST /u/ibmuser/newdir
      ```

   4. To ensure that this new file system is mounted again after you restart a system, you must also update the appropriate **BPXPRMxx** system parmlib member.

6. *Optional:* If your new volume is a non-SMS managed volume and you want to allocate new data sets directly to it, you need to specify **SCNOSMS** as the storage class when you allocate data sets through ISPF (3.2).

Changing active I/O software configuration for z/OS virtual server instances

The Hardware Configuration Definition (HCD) component of z/OS consolidates the hardware and software I/O configuration processes under a single interactive user interface. The output of HCD is an I/O definition file (IODF), which contains I/O configuration data. An IODF is used to define multiple software configurations to the z/OS virtual server instance. The IODF **PROV.IODF00** exists in the z/OS stock image. The corresponding data sets are on volume **OPEVS1** with device **0DE28**.

If you want to change the active I/O configuration, you need to use one of the following methods to activate the IODF.

- Using the HCD, see z/OS HCD User's Guide.
- Using MVS ACTIVATE command, see Activate command.

z/OS stock images always IPL with **LOADxx** member **LOADK2** found in **SYS0.IPLPARM**. The IODF statement identifies the I/O definition file that contains information about the I/O configuration, which means you need to specify the IODF statement so that the

corresponding I/O configurations will be used at IPL. You must follow the IODF statement policies for the z/OS virtual server instance:

- The IODF suffix is part of the IODF name. The range value is a two-digit hexadecimal number (X'00-FF').
- Asterisks (**), pluses (++), minuses (--), and equals (=) in IODF suffix are not supported for z/OS virtual server instances.
- The IODF high-level qualifier needs to be specified explicitly.
- The subchannel set indicator needs to be specified as **0**.
- The specified IODF needs to reside on the IODF disk device **0DE28** with volume **OPEVS1**.

For more information, see Statements and parameters for LOADxx.

Initiating a stand-alone dump with z/OS virtual server instances

You can initiate a stand-alone dump on the serial console from the IBM Cloud UI. The dump program is placed on the volume **DE26** and the target for the dump output is placed on the volume **DE2B**.

Before you begin

Make sure that you are connected to the z/OS virtual server instance through the serial console and open a new z/OS serial console. For more information, see Getting connected.

The z/OS serial console supports four operands: **start**, **stop**, **ipl**, **oprmsg**. For more information, see Operating with the serial console.

Procedures

To initiate a stand-alone dump by using the z/OS serial console, you can follow these steps:

1. Use the tab key to switch between the z/OS serial console and the command line.
2. To stop control processors, enter **stop all**, and press Enter.
3. To start the volume **DE2B**, enter **ipl DE2B**, and press Enter.
4. To initiate a stand-alone dump, enter **oprmsg 'device address'**, and press Enter.
5. To verify that the stand-alone dump is initiated successfully, see the following message as an example:

   ```
   oprmsg <device address> STAND-ALONE DUMP PROCESSING COMPLETED
   ```

Creating snapshots of boot and data volumes

Snapshots are a point-in-time copy of your Block Storage for VPC boot or data volume that you manually create. To create a snapshot, the original volume must be attached to a running virtual server instance. The first snapshot is a full backup of the volume. Subsequent snapshots of the same volume record only the changes since the last snapshot.

You can access a snapshot of a boot volume and use it to provision a new boot volume. You can create a data volume from a snapshot (called restoring a volume) and attach it to an instance. Snapshots are persistent; they have a lifecycle that is independent from the original volume.

Creating snapshots of a custom image data volume

You can create a snapshot or create an image from a boot volume on a virtual server instance. The image is a full copy of the source volume. It includes the operating system and any user data.

However, you cannot create an image from a data volume. To make a copy of a data volume, you need to create a snapshot of the volume.

When you create a virtual server instance from a custom image that you produced with IBM Wazi Image Builder, it contains both a boot volume and a data volume. The z/OS Wazi aaS custom image is more dependent on the data volume.

If you want to make a copy of this virtual server instance, make sure to create a snapshot of the data volume as well. You can then create new instances from the images and snapshots.

For more information, see the following topics in IBM Cloud documentation:

- Planning for snapshot creation and use
- Creating snapshots
- FAQs for snapshots

Regenerating RACF certificates for z/OS virtual server instances

If you have a long-running z/OS virtual server instance, or use previous z/OS stock images, you might want to regenerate the RACF certificates to extend the expiry date.

You can complete the following process to regenerate RACF certificates for your instance.

This process applies to z/OS virtual server instances that are created with z/OS stock images only.

1. Create a snapshot of the existing instance as a backup.
2. In TSO/ISPF, update data set **`SYS1.PARMLIB(IEASYM00)`** and change the date value of **`&CERTEXPIRE`** to the value that you want. Make sure that the syntax of this update is correct, otherwise you might fail to restart your z/OS system. The statement has the following syntax:

   ```
   SYMDEF(&CERTEXPIRE='YYYY-MM-DD')
   ```

3. Run the following TSO COMMAND from TSO/ISPF Option 6:

   ```
   RACDCERT CERTAUTH ALTER(LABEL('VSICA')) NEWLABEL('VSICA_REGEN')
   ```

4. Run the following TSO COMMAND to verify that the CA certificate has the new label:

   ```
   RACDCERT CERTAUTH LIST(LABEL('VSICA_REGEN'))
   ```

   If you receive the message: **`IRRD107I No matching certificate was found for this user, then the command in Step 3 did NOT work`**, then retry step 3.

5. Restart your z/OS system.
6. Verify that the CA certificate is re-created properly.

   ```
   RACDCERT CERTAUTH LIST(LABEL('VSICA'))
   ```

Establishing service to service authorizations

Before you can create backup policies, you need to establish service-to-service authorizations and specify user roles. This authorization enables the Backup for VPC service to detect volume tags and create backups.

Overview

Every user that accesses VPC infrastructure services resources must be assigned one or more access policies that define their IAM roles. These policies determine the actions that a user can perform within the context of the service or instance that you select. The allowable actions are customized and defined by the IBM Cloud service as operations that are allowed to be performed on the service. The actions are then mapped to IAM user roles.

To use Backup for VPC to create backups of block storage volumes, you must create service-to-service authorizations and user roles:

- IBM Cloud Backup for VPC (source) to Block Storage for VPC (target) as *Operator*
- IBM Cloud Backup for VPC (source) to Block Storage Snapshots for VPC (target) as *Editor*
- IBM Cloud Backup for VPC (source) to Virtual Server for VPC (target) as *Operator*

For more information about managing access to Virtual Private Cloud (VPC) resources, see Getting Started with IAM.

Procedure

From the UI, follow this procedure to create three authorization policies:

1. In the IBM Cloud console, go to **Manage > Access (IAM)**. The **Manage access and users** page is displayed.
2. From the side panel, select **Authorizations**.
3. On the **Manage authorizations** page, click **Create**. Then, create three separate service authorizations by repeating steps 4 - 6.
4. On the **Grant a service authorization** page, select **VPC infrastructure service** from the menu for both the source and target service. Identify the source and target by checking **Resources based on selected attributes** and **Resource type**. Table 1 lists the resource types for the source and target services, and user roles.
5. Under **Authorize dependent services**, select the role based on the following table.

Source service - resource type	Target service - resource type	Dependent service user role
IBM Cloud Backup for VPC	Block Storage for VPC	Operator
IBM Cloud Backup for VPC	Block Storage Snapshots for VPC	Editor
IBM Cloud Backup for VPC	Virtual Server for VPC	Operator

Table 1. Service-to-service authorizations

6. Click **Authorize**.

Applying backup policies to resources by using tags

Apply backup policies by adding tags to new or existing block storage volumes. When these tags match a backup policy tag, a backup is created.

General procedure

1. Create a backup policy and plan.
2. Apply backup policy tags to your target block storage volumes by using the UI. Go to your block storage volume that you want to backup and add at least one tag to it.
3. Verify that your block storage volume is associated with a backup policy. For more information, see View a list of volumes that have a backup policy.

Applying tags to block storage volumes

Apply tags to new or existing block storage volumes in any of these ways:

- Add tags to volumes directly from the block volumes list view by using the tags column.
- Add tags from the volume details page.
- Click **Apply** for backup policies on the volume details page to associate backup policies by using tags

Add tags from the block storage volumes list view

1. Go to the list of block storage volumes.
2. Locate an *available* volume. In the **Tags** column, volumes with tags show a number that indicates tags that are already applied. You can view the tags by clicking the number link. Volumes without tags have an **Add tags** link.
3. Click **Add tags**.
4. In the new window, type a tag in the **User tags** text box.
5. Click **Save**.

Add tags from the volume details page

1. Go to the list of block storage volumes.

2. Click the name of a volume in the list.
3. On the volume details page, click the pencil icon next to the block storage volume name.
4. In the new window, type a tag in the **User tags** text box.
5. Click **Save**.

Planning your file shares

When you're planning to create file shares on your VPC, you might find this checklist helpful to set up and use the file service.

IBM Cloud® File Storage for VPC is available for customers with special approval to preview this service in the Frankfurt, London, Dallas, Toronto, Washington, Sao Paulo, Sydney, Tokyo, and Osaka regions. Contact your IBM Sales representative if you are interested in getting access.

Planning for creating file shares

Consider the following prerequisites before you set up File Storage for VPC.

- Evaluate your IAM access permissions to create file shares.
- Evaluate the capacity and performance that you need by selecting a File Storage profile. You can select an IOPS tier profile or use Custom IOPS to tailor capacity and IOPS performance to meet your needs.
- Choose the UI, CLI, or API for creating and managing your file shares.
- Decide whether to set up replication when creating the new share. The replica file share is in a different zone from the primary file share. Replication is a good way to recover by having a read-only copy of the share in a different zone in your region. You can fail over to the replica site if the primary site becomes unavailable or compromised.
- Evaluate how you plan to use your file shares.
- Make sure that you have a unique name for your file shares that easily identify the file share as your list of shares grows. Associate it with a resource group in your IBM Cloud customer account.
- When creating a file share, choose the encryption type that best suits your needs. By default, your file shares are encrypted with IBM-managed encryption. For greater control, consider using customer-managed encryption. Your data is protected at rest with your own root keys. For information about this option, see Creating file shares with customer-managed encryption.
- When creating a file share, decide when to create a mount target. You can create additional mount targets separate from file share creation. Also, make sure that you created a VPC.|
- Estimate the size of the file share that you require for your current needs. You can later increase the size of the file share, depending on what the file share profile allows.
- To mount a file share on a target, you must create at least one mount target.
- To create a mount target, you must have a VPC. Identify an existing VPC or create a new one.

Creating file shares and mount targets

You can create file shares and mount targets in your IBM Cloud VPC by using the IBM Cloud console.

IBM Cloud® File Storage for VPC is available for customers with special approval to preview this service in the Frankfurt, London, Dallas, Toronto, Washington, Sao Paulo, Sydney, Tokyo, and Osaka regions. Contact your IBM Sales representative if you are interested in getting access.

Create a file share

To create a file share, complete the following steps:

1. In the IBM Cloud console, go to **Menu icon** > **VPC Infrastructure** > **Storage** > **File Shares**. A list of file shares displays.
2. On the File shares for VPC page, click **Create**. The **Create** tab is selected by default.
3. Enter the information described in the following table.
4. When you create a file share, specify the information for the mount target as well.
5. When finished, click **Create file share**. You're returned to the File Storage for VPC page, where a message indicates that the file share is provisioning. When completed, the status changes to **Active**.

Field	Value
Location	Choose the geography, region, and zone where you want to create the file share. Location information is inherited from the VPC, for example, North America, Dallas, Dallas 2.
Name	Specify a meaningful name for your file share. The file share name can be up to 63 lowercase alpha-numeric characters and include the hyphen (-), and must begin with a lowercase letter. You can later edit the name if you want.
Resource Group	Use the default resource group or specify a resource group. Resource groups help organize your account resources for access control and billing purposes.
Tags	Enter user tags to associate with this file share. As you type, existing tags appear that you can select. For more information about tags, see Add user tags to a file share.
Access Management Tags	Enter access management tags that you created in IAM to apply them to this file share. For more information about access management tags, see Access management tags for file shares.
Mount targets (Optional)	Click **Create** to create a new mount target for the file share. You can create one mount target per VPC per file share. Provide a name for the mount target and select a VPC in that zone. You can add as many mount targets are you have VPCs. If you don't have one, first create a VPC. (To use the API, see Creating a VPC with the REST APIs.) For more information about creating mount targets as a separate operation, see Create a mount target.
Profile	Select an IOPS tier or Custom IOPS for file share. The profile that you select determines the input/output performance of a file share. For more information about file storage IOPS tier and Custom profiles, see File storage profiles.
Size	Specify the size for the file share. You can later increase this size, depending on the file share profile.
Encryption	Encryption with IBM-managed keys is enabled by default when you create a new file share. You can also choose **Customer Managed** and use your own encryption key. For more information about creating encrypted file shares, see Creating file shares with customer-managed encryption. If you create a replica share while provisioning a new file share, the encryption is inherited. You can't encrypt a replica with a different key. If you change the encryption on the source share, the replica is updated.
Asynchronous replica	Informational field that says that you can create a replica after the file share is created.

Table 1. Values for creating a file share and mount target

To see the REST API call, click the **Create with REST API </>** link. Viewing the API calls is a good way to learn about the API and understand actions and their dependencies.

Create a mount target

You can create one or several mount targets for an existing file share by completing the following steps:

1. In the IBM Cloud console, go to **Menu icon** > **VPC Infrastructure** > **Storage** > **File Shares**.
2. Select a file share from the list.
3. On the **File shares details** page, under Mount targets, click **Create**.

   You must have at least one VPC to create a mount target. If you don't have one, first create a VPC.
4. In the **Create mount target** window, provide a name for the mount target.
5. Select a VPC from the list and then click **Create**.

For more information, see Creating file shares and mount targets.

Mounting file shares on z/OS

Use these instructions to connect a z/OS-based IBM Cloud Compute Instance to a Network File System (NFS) file share.

IBM Cloud® File Storage for VPC is available for customers with special approval to preview this service in the Frankfurt, London, Dallas, Toronto, Washington, Sao Paulo, Sydney, Tokyo, and Osaka regions. Contact your IBM Sales representative if you are interested in getting access.

Before you begin

Before you mount File Storage for VPC file shares, make sure you meet the following requirements:

1. Verify that the virtual server instance where you want to mount the share is in the same zone as the file share.
2. Confirm that a mount target for the share exists for the VPC that the instance resides in. If a new mount target is needed, follow the instructions in Creating file shares and mount targets.
3. Get the mount path of the file share from the mount target. Mount path information can be obtained from the File share details page in the UI, from the CLI, with the API or Terraform.

Mount the file share

Follow these steps to mount a file share on a z/OS host:

1. Create a **/u/ibmuser** directory.

   ```
   mkdir /u/ibmuser/test
   ```

2. Start the NFS client. You can issue commands **D OMVS,P** to verify whether the NFS client is started or not.

   - If the entry for NFS file system is in the **sys1.parmlib.bpxprm00**, you can safely skip this step.
   - If the entry for NFS file system is not in the **sys1.parmlib.bpxprm00**, you need to complete the following steps to start the NFS client:

     a. To set up the NFS client, run the following job on your z/OS virtual server instance.

     ```
     //NFSCLNT  JOB CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID.,
     //         REGION=0M
     //*********************************************************************
     //* Configuration for the NFS client.                                 *
     //*********************************************************************
     //ALIAS    EXEC PGM=IDCAMS
     //SYSPRINT DD SYSOUT=*
     //SYSIN    DD *
       DEF ALIAS(NAME(MVSCLNT) RELATE(CATALOG.VS01.SYS)) -
           CAT(CATALOG.VS01.MASTER)
     /*
     //DEFLOG1  EXEC PGM=IEFBR14
     //SYSPRINT DD SYSOUT=*
     //LOG1     DD DSN=MVSCLNT.LOG1,
     //         DISP=(NEW,CATLG),UNIT=3390,SPACE=(CYL,(2,5),RLSE),
     //         RECFM=VB,LRECL=137,BLKSIZE=6144,VOL=SER=OPEVS1
     //*
     //DEFLOG2  EXEC PGM=IEFBR14
     //SYSPRINT DD SYSOUT=*
     //LOG2     DD DSN=MVSCLNT.LOG2,
     //         DISP=(NEW,CATLG),UNIT=3390,SPACE=(CYL,(2,5),RLSE),
     //         RECFM=VB,LRECL=137,BLKSIZE=6144,VOL=SER=OPEVS1
     //*
     //DEFRACF  EXEC PGM=IKJEFT01
     //SYSPRINT DD SYSOUT=*
     //SYSTSPRT DD SYSOUT=*
     //SYSTSIN  DD *
       RDEFINE STARTED MVSNFSC.* UACC(NONE) +
         STDATA(USER(SYSTEM) GROUP(SYS1) TRUSTED(YES))
       SETR REFRESH RACLIST(STARTED)
     /*
     //COPYSAMP EXEC PGM=IEBCOPY
     //SYSPRINT DD SYSOUT=*
     //SYSUT1   DD UNIT=SYSDA,SPACE=(TRK,(5,1))
     //SYSUT2   DD UNIT=SYSDA,SPACE=(TRK,(5,1))
     //SYSUT3   DD UNIT=SYSDA,SPACE=(TRK,(5,1))
     //SYSUT4   DD UNIT=SYSDA,SPACE=(TRK,(5,1))
     ```

```
//INDD     DD DSN=SYS1.NFSSAMP,DISP=SHR
//OUTDD    DD DSN=SYS1.PROCLIB,DISP=SHR
//SYSIN    DD *
  COPY I=INDD,O=OUTDD
  SELECT MEMBER=((GFSCPROC,MVSNFSC))
/*
```

b. Add the following command to **SYS1.PARMLIB(BPXPRM00)** after **AUTOMNT**.

```
FILESYSTYPE TYPE(NFS) ENTRYPOINT(GFSCINIT) ASNAME(MVSNFSC)
```

c. To start the NFS client address space **MVSNFSC**, issue the following commands from the console **/SDSF Log**.

```
set OMVS=00
```

d. To verify that the NFS client is started, issue the commands **D OMVS,P**.

If the principal **MVSNFSC** is in the list, the NFS client is started successfully.

3. Run the following commands from TSO to mount the file share on z/OS in ASCII format:

```
mount type(NFS) filesystem(nfs02) mountpoint('/u/ibmuser/test') parm('<mount
path>,xlat(y),vers(4)')
```

You can also run the following commands to mount the file share on z/OS in binary format:

```
mount type(NFS) filesystem(nfs02) mountpoint('/u/ibmuser/test') parm('<mount path>,vers(4)')
```

For more information, see Additional mount command examples.

4. To verify that mounting file shares to z/OS is successfully completed, you can run command **df** to display the directory **/u/ibmuser/test**.

Shutting down z/OS virtual server instances

Before you stop or reboot a z/OS virtual server instance, you must shut down all the subsystems in the z/OS system.

Similar to deactivating a logical partition (LPAR) on a native Z system, the z/OS system must be shut down manually before it is stopped or rebooted. If the z/OS system is not properly shut down, it will result in inconsistent or corrupted data, and the system will have trouble IPLing the next time.

Before you begin

Before you shut down the z/OS virtual server instance, complete the following prerequisites:

1. Make sure that you connect to the z/OS console by using the serial console.
2. Notify users that the system is being shut down and ask them to log off. For example, use the operator **SEND** command to send a note to all TSO/E users and inform them that the system will be shut down.

```
send 'The system is being shut down in five minutes. Please log off.',NOW
```

Procedures

To shut down z/OS virtual server instances that are running on the previous version of the hypervisor, complete the following steps:

1. To stop all the z/OS subsystems and the z/OS system, you can select one of the following approaches:
   - Use the all-in-one command on the z/OS serial console. If your z/OS virtual server instance is created from the z/OS dev and test stock image, use the following command:

     ```
     oprmsg '%netv shutsys'
     ```

   - Use separate commands to shut down each z/OS subsystem and the z/OS system in sequence.
     - To stop the various z/OS subsystems, you can run the following operator commands sequentially. For more information, see OPERATOR command.

```
P IZUSVR1
SETRRS SHUTDOWN
C SSHD4
P FTPD1
P IZUANG1
P SDSF
P TN3270
P TSO
P PAGENT
P CSF
P VLF
P LLA
<STOP
P EQAPROF
C EQARMTD
P RSED
P JMON
C RSEAPI
C DBB
C DBBS
P DBGMGR
P TCPIP
Z NET,QUICK
F OMVS,SHUTDOWN
$PJES2
```

- To stop the z/OS system, you can run the following commands after virtual telecommunications access method (VTAM), Job entry subsystem 2 (JES2), and the OMVS shell are successfully stopped.

  ```
  oprmsg 'V XCF,VS01,OFFLINE'
  ```

- To reply to the resulting write-to-operator-with-reply (WTOR) message, you must enter a response with the following command. The **xx** in the following command example is the reply number.

  ```
  oprmsg 'R NN,SYSNAME=VS01'
  ```

2. You can stop the z/OS virtual server instance by using the IBM Cloud console now.

Security best practices for z/OS virtual server instances

To secure your z/OS virtual server instance and identify any security vulnerabilities, you must refer to the following security information and industry standard security reports.

Products and services in the z/OS stock images are periodically reviewed and updated. IBM continues to follow the standard security guidance and provides differentiating technologies in security and data privacy, focusing on but not limited to the following areas:

- Security patch management
- Firmware currency
- Setup of suggested products and services
- Configuration of hardware and software (operating systems, middleware, third-party applications, open source, and network cards)
- Integration and monitoring of software, hardware, and end-points
- Cybersecurity:
  - Least access privilege
  - Separation of duties
  - Defense-in-depth
  - Authentication strength
  - End-to-end encryption

Make sure that you follow the security best practices for your z/OS virtual server instance:

- If you want to apply the latest service for your instance, you can wait until the next month's stock image to create a new instance.
- Ensure that you follow the password policy, see Configuring the password.
- For more information, see the following references:
  - What is zero trust?

- NIST Special Publication Security & Privacy Controls for Information Systems & Organizations
- IBM X-Force Threat Intelligence Report
- IBM on Enterprise Security
- PCI DSS v4.0
- CIS Benchmarks for IBM z Systems

Troubleshooting z/OS virtual server instances

This document covers common difficulties that you might encounter when you use z/OS virtual server instances, and offer helpful tips on how to resolve them.

Why can't I remove broadcast messages when working with a z/OS virtual server instance?

When you attach a block storage to your z/OS virtual server instance, a broadcast message is displayed on your display workstation. This happens every time when you log on to the instance by using a TN3270 connection.

```
Preparing attached block storage vde of size 15G
Attached block storage vde on address DE00
```

You can remove the broadcast message by using the following procedure:

1. Log on to the z/OS instance by using a TN3270 terminal emulator.
2. Enter **OPERATOR** on the command line and press Enter. The **OPERATOR** command along with its subcommands are used to regulate and maintain TSO/E from a terminal. For more details, see OPERATOR command.
3. Enter the subcommand **SE LIST** and press Enter. The **SE LIST** subcommand lists all the broadcast messages.

   You might have to press Enter twice so that the subcommand takes effect.
4. Enter the subcommand **SE NNNN,DELETE** to delete a particular message where **NNNN** is the message number of the broadcast message you want to delete. For example, **SE 002, DELETE**.
5. Enter the subcommand **SE LIST** to verify that the message is deleted successfully.
6. Enter the subcommand **END** to terminate operation of the OPERATOR command.

Why can't I connect to the z/OS virtual server instance?

What's happening

- You open a TN3270 terminal emulator with your user ID to log on to the TSO/E. However, you cannot connect to the z/OS virtual server instance and you might see the **logon proceeding** or **logon rejected** messages appear on your display terminal.
- You cannot access the z/OS virtual server instance by using your user ID on the IBM Cloud UI or CLI.

Why it's happening

The potential cause that you cannot connect to the z/OS virtual server instance is the resource shortage in the job entry subsystem 2 (JES2). For example, if you find the **$HASP050** messages in your serial console on the IBM Cloud UI or CLI, the event occurs because of the resource shortages. For more information about **HASP050** messages, see $HASP050 message for resource shortages.

How to fix it

You can use the following methods to resolve the resource shortage:

1. Increase the quantity of the JES2 resource with a **$T** command or decrease the demand for the resource by purging held output.
   - To increase the limit of job queue elements (JQEs), which is one of the resource types in JES2, you can use the following instructions:

a. Access the z/OS virtual server instance serial console to submit operator commands. For more details, see OPERATOR command.

b. Enter the subcommand **$DJOBDEF, JOBNUM** to display the current maximum number of JQEs. For example, **1000**.

c. Enter the subcommand **$TJOBDEF, JOBNUM=x** to increase the limit of JQEs, where **x** must be higher than the maximum number in the previous step. For example, **$TJOBDEF, JOBNUM=5000**.

- To decrease the demand for the resource, see Purge a job's output.

2. Then, you can enter **$S** to reinitialize the JES2 and reconnect to the z/OS virtual server instance successfully.

   If you are currently logged in to the z/OS virtual server instance, enter **c u=userid** to cancel the current session and reconnect to the z/OS virtual server instance. For more information, see OPERATOR-CANCEL subcommand.

Why can't I access z/OSMF through the Chrome browser?

What's happening

When you try to connect to IBM z/OS Management Facility (z/OSMF) on the z/OS virtual server instance through the Chrome browser, you might receive error messages **NET::ERR_CERT_INVALID** or **ERR_CERT_AUTHORITY_INVALID** depending on the operating system on your local workstation.

Why it's happening

The connection to the z/OS virtual server instance is using a self-signed CA certificate, which is not trusted by the Chrome browser.

How to fix it

- For Windows, you can click **Advanced** and select **Proceed anyway**.
- For macOS, Linux, and Windows system where the Secure Copy Protocol (SCP) client is installed, you need to take the following steps to import and trust the self-signed CA certificate:

  1. Copy the self-signed common CA certificate from the z/OS virtual server instance to your local workstation. The self-signed CA certificate is stored in a file that is called **common_cacert**, which is in the home directory of the **IBMUSER** user ID in z/OS UNIX System Services.

     For example,

     ```
     scp ibmuser@<vsi ip address>:/u/ibmuser/common_cacert
     ```

  2. Open your **Keychain Access** app and import the self-signed CA certificate into the system keychain.
  3. After the self-signed CA certificate is imported into your keychain, double-click the self-signed CA certificate and expand the **Trust** list. Under the **When using this certificate**, select **Always Trust** from the list.
  4. Reload the URL to access z/OSMF through the Chrome browser.

     If you receive a warning message **NET::ERR_CERT_COMMON_NAME_INVALID** after reloading the z/OSMF URL, click **Advanced** and select **Proceed anyway**.

Why does my z/OS virtual server instance crash with memory shortage?

When you submit a job on z/OS, it might experience memory shortages, resulting in S0C4 abends or invalid return codes.

To increase the amount of available memory to your jobs, you can specify the **MAXSHARE** parameter in the **SMFLIMxx** parmlib member.

To do this, complete the following steps:

1. Navigate to ISPF, enter the 3.4 panel to use the Data Set List Utility.
2. Search for **SYS1.PARMLIB**.
3. Create a data set member named **SMFLIMxx**, where xx is either two digits or two letters. For example, **SMFLIM00**, **SMFLIMXZ** are valid names.
4. Edit the **SMFLIMxx** member to include the following line:

```
REGION MAXSHARE(2000)
```

This step defines 2000 pages, and each page is a 4 KB block of virtual storage. You can increase the page count by a factor of 10 if there is a large amount memory allocated to the instance. Note that it increases the amount of memory that is used by the instance.

5. Issue the operator command **`set smflim=(xz)`**, and you can see the following output:

```
SET SMFLIM=(XZ)
IEE536I SMFLIM   VALUE XZ NOW IN EFFECT
```

6. Then, verify the command that you went through with the operator command **`d smflim,r`**.

```
D SMFLIM,R
IFA900I 09.52.31 SMF LIMIT DATA 191
Member and rule number SMFLIMXZ 0001
```

Once this is done, you can resubmit your job.

Getting started with IBM Wazi as a Service

In this tutorial, you can learn how to develop and test your applications with your virtual server instance by using IBM's development tools portfolio.

You can learn how to use tools such as VS Code and Eclipse, understand Cloud concepts and tools and how they add value to z/OS DevOps.

You can also learn new technologies such as Ansible for automation, DBB and Groovy for build, GitLab and Tekton for serverless pipelines.

The tutorial contains the following topics:

- Checklist for the prerequisites
- How to configure access to the available services in the VSI
- How to develop and test with IBM Wazi for VS Code
- How to develop and test with IBM Wazi for Eclipse or IBM Developer for z/OS
- How to deploy and use IBM Wazi for Dev Spaces from a Red Hat OpenShift cluster in IBM Cloud VPC

To use the tools, such as IBM Z Open Editor, Open Debug, or IBM Wazi for Dev Spaces, IBM Wazi for Eclipse, you need to agree to different license agreements or make a purchase before you install.

Intended audience

This tutorial is for developers who want to develop and test their z/OS applications from a cloud environment.

This tutorial distinguishes the following types of user roles:

- Cloud admin
- System admin
- System programmer
- z/OS App developer

The different tasks that are described here are associated to one of these user roles. A user can have one or multiple roles.

Prerequisites

Before you start to develop and test your applications with your virtual server instance in IBM Cloud Virtual Private Cloud (VPC), make sure that you meet the following prerequisites.

IBM Cloud Virtual Private Cloud configuration prerequisites

Ensure you have access to a VSI that you set up in an IBM Cloud VPC with the following characteristics:

- The VSI uses the z/OS development and test stock image (**ibm-zos-2-5-s390x-dev-test-wazi**) or a custom image that you created by using Wazi Image Builder.
- The VPC is configured for access securely with a Virtual Private Network that you connected to. Do not use a floating IP because it might expose your image ports to the internet.
    - See Setting up OpenVPN to access the VSI for instructions on how to install a VPN server.
    - See Configuring an OpenVPN client to access the VSI for instructions for users to download and install a VPN client.
- (Optional) The VSI runs on a subnet that access to the internet via public gateway. For more information, see External connectivity. This is required only for convenience with the tutorials and scripts that are provided in this guide to allow cloning GitHub repositories with sample code directly on the z/OS UNIX System Services of the z/OS VSI. You can also clone the repo on your local development client and copy it to the VSI. To create a public gateway, complete the following steps:
    1. Log in to IBM Cloud.
    2. In the VPC Infrastructure Web UI, select **Public gateways**.
    3. Click **Create** to create a gateway for the VPC in the same location and resource group as your z/OS virtual server instanceI.
    4. Associate the gateway with the subnet in which your instance runs. You can do this in the web UI of the gateway or the subnet in IBM Cloud.
- The instance is created with at least one SSH public key for access. The creator of the instance can add more public keys by using SSH and adding them to the **~/.ssh/authorized_keys** file in the home directory of the z/OS users that they enabled, for example in **/u/ibmuser/.ssh/authorized_keys**.
- All z/OS users are disabled by default when you create a VSI. You must enable at least one user to complete the tutorial steps. The creator of the VSI can log on with the SSH key and execute TSO commands on z/OS UNIX to enable the user with a password phrase. The password phrase can then be used for client tools such as Zowe CLI profiles.

```
ssh ibmuser@<vsi ip address>
tsocmd "ALTUSER IBMUSER PHRASE('YOUR PASSWORD PHRASE') NOEXPIRE RESUME"
```

Development client prerequisites

To use the VSI with the IBM development tools of your choice, ensure that you have fully installed and configured at least one of the following development clients:

- VS Code with the latest extensions and Zowe CLI components. Go to Downloads and click **VSCode** in the IBM Wazi Code row of the IBM Z and Cloud Modernization Stack table to get a **.zip** file. For more information, see Getting started with IBM Z Open Editor.
- IBM Wazi for Eclipse 2.4.0 or IBM Developer for z/OS 16.0.0
- IBM Wazi for Dev Spaces as documented in details in Deploying and using IBM Wazi for Dev Spaces from a Red Hat OpenShift cluster in your VPC

Optionally, you can use Ansible playbooks to simplify developer setup. To use Ansible playbooks, you need to run predefined playbooks and review them, but not edit them, and have it configured on your development machine. If you do not have Ansible installed, you need to run commands that are included in the playbooks manually.

You can run the following two commands on MacOS with Brew to install Ansible and the Red Hat Ansible Certified Content for IBM Z.

```
brew install ansible
ansible-galaxy collection install ibm.ibm_zos_core
```

For an overview of all the prerequisites and installation alternatives for Ansible, check out Requirements.

The Git repository contains example programs and scripts to use in these tutorials. Ensure that you have access to GitHub and have installed Git on your development machine. If you are working on Windows, you need to install and configure Git for Windows. Then, you can also execute the Bash shell scripts that are provided with the examples.

Available services

The z/OS development and test stock image provides the following services.

Services	Ports

Services	Ports
SSH	22
FTP	21
TN 3270 secure	992
RSE API	8195
RSE Daemon	8137
RSE Server	8138
IBM Z Open Debug: Debug profile service Local debug port remote debug service port	 8192 8193 8194
z/OSMF	10443

Table 1. Key services and ports

For more information about port numbers, installed libraries and modules, and other available software packages, see **devtest1.yml** and Configurations in z/OS stock images.

Setting up OpenVPN to access the VSI

To set up a VPN for VPC service, configure the VPN by using a dedicated VSI that you create and configure manually with an OpenVPN distribution.

If you are assigned the role of Security Focal for the VPC, complete the following steps:

1. Go to the IBM Cloud console.
2. Create a Linux VSI (x86, CentOS, or Ubuntu) to act as VPN Server. Provide only the SSH Key of the VPC's Security Focal during creation. The Security Focal is responsible for all setup and maintenance of the VPN server, including the adding of new user access.
3. Create an IBM Cloud Security Group with an ***Inbound*** rule allowing the OpenVPN access using the **UDP** protocol via port **1194** and an ***Outbound*** rule allowing all data to pass through (*Protocol:* **ALL**, *Destination type:* **Any**). For more information about security groups, see About security groups.
4. SSH is needed to access the VPN server to configure it and to add new users. You can either add **TCP** port **22** to the new security group or add a new security group that only allows SSH access, enabling and disabling it as needed.
5. Assign the new security group to the VPN Server.
6. Assign a floating IP to the VPN Server.

When the VPN VSI is running, install and configure OpenVPN with the following steps:

1. Get the helper script by running the following commands on the VPN server:

```
curl -O https://raw.githubusercontent.com/angristan/openvpn-install/master/openvpn-install.sh
chmod +x openvpn-install.sh
```

2. Run the helper script:

```
./openvpn-install.sh
```

Accept default settings for all configuration options except the following: - For the IPv4 address of the network interface you want OpenVPN to listen to, specify the floating IP address that is assigned to the VPN Server.
- For DNS, choose option 1 (**Current system resolvers (from /etc/resolv.conf)**).

1. Edit the **/etc/openvpn/server.conf** script to change the default configuration. Remove or comment out the lines that contain the existing **push** directives, for example, **push "dhcp-option DNS ..."** and **push "redirect-gateway ..."**, then add the following directive to push the route to your network. Adjust the **10.242** to match the network configuration of your VPC—for example. For the London region, use example **10.242.0.0**; for Tokyo, replace that with **10.244.0.0**:

```
push "route 10.242.0.0 255.255.0.0"
```

2. Restart the OpenVPN service: **systemctl restart openvpn**

When the VPN server is configured and running, complete the following steps to generate a configuration for each user:

1. Re-execute the **`openvpn-install.sh`** script, select **Add a new user**, and specify the username in **`client name`**.
2. Distribute the **`<client name>.ovpn`** file to the users so that they can import this file into their OpenVPN client. The z/OS VSI is then accessible through its reserved (non-public) IP address, for example, **`10.244.64.5`**.
3. Optionally, after the configuration of the VPN server is completed and all user configuration files are created, you can block port **`22`** (ssh) through the IBM Cloud Dashboard and leave only UDP port **`1194`** active. You can reopen port **`22`** temporarily when a new user configuration file needs to be generated.

Configuring an OpenVPN client to access the VSI

You need an OpenVPN server to protect your access to virtual server instances. Follow the instructions to set up your OpenVPN client to access the VSI that is protected by the VPN.

After the administrator configures an OpenVPN server, you can get your own **`.ovpn`** configuration file to enable your access to the VSI, and the IP address of the VSI.

To install and configure an OpenVPN client, you can get the official OpenVPN client from OpenVPN, or an OpenVPN-compatible client for macOS from TunnelBlick.

After you install the OpenVPN client, import your configuration file to configure the client.

To configure the OpenVPN client by using the official OpenVPN client, complete the following steps:

1. From the menu, select **Import Profile**.
2. Choose **File**.
3. Drag the configuration file onto the client's window, or browse and find the file.

Your client is configured. You can connect and access the VSI by using the IP address that you get from your administrator.

Configuring access to services

If you are using IBM Wazi for VS Code or Z Open Editor or IBM Wazi for Eclipse development clients, you can run the following commands to clone the sample repository in a directory of your choice that provides scripts and code samples for exploring working with the z/OS VSI. If you are using IBM Wazi for Dev Spaces, you can use the "IBM Wazi with sample" stack to create a workspace, which automatically clones and loads this sample repository.

```
git clone git@github.com:IBM/zopeneditor-sample.git
git checkout wazi-main
```

1. Click **File > Open Folders** to open this repository in your VS Code editor.
2. In IBM Wazi for Eclipse, use the Git perspective to add the cloned folder as an existing Git repository, and then the context menu for this repository with the **Create z/OS Project** command to add it to your **z/OS Project Explorer**.
3. Review the **`README.md`** file in the **`wazi-main`** branch for an overview of the code and configuration scripts for building and running the sample applications.
4. Verify that you have received the **`10.x.x.x`** IP address and connected to the VPN for accessing your instance. Ensure that the admin who created the VSI has added your public SSH key to the instance as documented in Creating your SSH key.
5. You can test your access by logging in with **`ssh ibmuser@<vsi ip address>`**. Password is not needed.
6. If you receive a **`Permission denied (publickey).`** error message, it means that your public key is missing. Follow the instructions in Configuring your development workspace manually and shell scripts to connect to the VSI by using Zowe Explorer and add your SSH key.

If you are using IBM Wazi for Eclipse or IBM Developer for Z, you can also add your SSH key by using these tools.

Developing and testing with IBM Wazi for VS Code

Learn how to develop and test with IBM Wazi for VS Code.

- Connecting IBM Wazi for VS Code

  You can connect IBM Z Open Editor with Zowe Explorer in VS Code. Before you begin, ensure that you set up Z Open Editor with Zowe CLI and are familiar with the basic operations.

- Running Dependency-Based User Builds from Z Open Editor

  Z Open Editor user build allows you to run builds directly out of the editor remotely on z/OS UNIX with IBM Dependency Based Build.

- Running a Debug session from Z Open Editor

  To start a Debug session in VS Code, you need to install the two IBM Z Open Debug extensions.

Connecting IBM Wazi for VS Code

You can connect IBM Z Open Editor with Zowe Explorer in VS Code. Before you begin, ensure that you set up Z Open Editor with Zowe CLI and are familiar with the basic operations.

Open the **`zopeneditor-sample`** you cloned as a workspace and make sure that you are in the **`wazi-main`** branch. Then, follow these steps:

- Setting up Ansible for IBM Wazi

  Ansible and the Red Hat Ansible Certified Content for IBM Z collections provide important capabilities that help administrators with enterprise automation scenarios and help developers automate their development setup.

- Configuring your development workspace with Ansible

  With Ansible available in your development workspace, you can run the configuration scripts that are provided in the **`ansible`** folder.

- Configuring your development workspace manually and shell scripts

  If you do not have Ansible available on your development machine or no public ssh key is configured for the VSI, you can still connect to the VSI by using VS Code with Zowe Explorer.

Setting up Ansible for IBM Wazi

Ansible and the Red Hat Ansible Certified Content for IBM Z collections provide important capabilities that help administrators with enterprise automation scenarios and help developers automate their development setup.

In particular, you can use Ansible to prepare a remote z/OS system such as IBM Wazi Sandbox for the development prerequisites that are necessary for building and running applications right from your IDE.

You can edit and run Ansible playbooks directly in IBM Wazi by using editing capabilities such as the Red Hat Ansible language server, as well as running playbooks that use the Ansible Certified Content for IBM Z.

To familiarize yourself with these Ansible collections, check out these resources:

- Red Hat Ansible Certified Content for IBM Z overview
- Red Hat Ansible Certified Content for IBM Z documentation
- GitHub repository with examples for using Red Hat Ansible Certified Content for IBM Z for many more scenarios.

IBM Wazi for Dev Spaces has all of these capabilities preinstalled and configured when you use the IBM Wazi for Dev Spaces stack and container image. When you create your workspace, it has Ansible and its prerequisites installed and ready to use, including Python, Red

Hat Ansible Certified Content for IBM Z, the VS Code extension, and the language server for Ansible editing that offers features such as code completion and linking.

If you are using other editors, such as IBM Wazi for VS Code or IBM Wazi for Eclipse, you need to set up Ansible on your development desktop machine.

Setting up Ansible in IBM Wazi for Dev Spaces

You can create or open an Ansible playbook file with the extension **`.yml`** or **`.yaml`**, and select **`Ansible`** in the language selector of the editor in the status bar. (The default language is **`YAML`**, which does not provide the right language capabilities.)

After you select **`Ansible`** as the language, the editor provides features such as rich hovers with help text, code completion through ctrl-space, and linking warnings and errors that are added or removed when you save a file.

To enable these features for the preinstalled Red Hat Ansible Certified Content for IBM Z, add the following code to your playbook:

```
  collections:
    - ibm.ibm_zos_core
```

To learn more about the Ansible editor and additional configuration options, see Ansible VS Code Extension by Red Hat.

To run Ansible playbooks directly out of IBM Wazi for Dev Spaces, open the **`wazi-terminal`** from the Terminal menu and run the **`ansible`** or **`ansible-playbook`** commands. Check out Configuring your development workspace with Ansible for running playbooks that support COBOL development.

Setting up Ansible on MacOS

To use the same capabilities that are described previously for IBM Wazi for Dev Spaces, you need to install Ansible on your Mac, and then install and configure the VS Code extension.

You can install Ansible by using the Brew package manager, which also takes care of all prerequisites, such as Python.

```
brew install ansible
brew install ansible-lint
```

If the package manager is not available, see the Ansible Documentation that outlines alternative methods.

After the installation is completed, you can add the z/OS collections by running the following command:

```
ansible-galaxy collection install ibm.ibm_zos_core ibm.ibm_zos_ims ibm.ibm_zos_cics ibm.ibm_zhmc
ibm.ibm_zos_sysauto
```

Check out Installation instructions for more details.

You can now run Ansible collections from any command-line tool, such as the Mac Terminal program or the terminal view in VS Code.

If you use IBM Wazi for Eclipse, the MacOS terminal is the preferred option.

If you use IBM Wazi for VS Code, to edit Ansible playbooks with the exact same features for IBM Wazi for Dev Spaces, install the Red Hat Ansible VS Code extension from the VS Code Marketplace.

Setting up Ansible on Windows

To install and run Ansible on Windows, you can use the Windows Subsystem for Linux (WSL). Follow these basic steps:

1. Follow the instructions for installing WSL. Run the following command from a Powershell:

   ```
   wsl --install
   ```

   The default WSL is installed by using Ubuntu. Review the configuration documentation for various configuration options. You need to create a username and a password. The Microsoft documentation also has a special page for using WSL with VS Code to review. This document outlines two basic modes of working with WSL:

   a. Keeping the source code on Windows and accessing it from WSL for running commands such as playbooks.

   b. Moving the source code into WSL and opening it in VS Code as a remote workspace. For that model, you can install the WSL VS Code extension from Microsoft that manages such a remote access for you. You can also use the second approach to clone Git repos into WSL and open VS Code folders on WSL.

2. Install Ansible into WSL/Ubuntu. See the Ansible Documentation for more information.

```
sudo apt update
sudo apt install software-properties-common
sudo add-apt-repository --yes --update ppa:ansible/ansible
sudo apt install ansible
sudo apt install ansible-lint
```

3. Install Red Hat Ansible Certified Content for IBM Z Collection. Check out Installation instructions for more information.

```
ansible-galaxy collection install ibm.ibm_zos_core
```

4. Set up passwordless SSH from the WSL/Ubuntu shell to your z/OS system:

```
ssh-keygen -t rsa -b 4096 -C "username@your-org.com"
ssh-copy-id username@zos_ip_address
```

   Now you are ready to run Ansible playbooks from the WSL Ubuntu command shell from within VS Code or a separate window.

5. Finally, install the Ansible VS Code extension for Ansible to get VS Code editing support. Depending on whether you are working with your source code from Windows or the WSL/Ubuntu environment, you need to install the extension locally or remotely through the WSL extension. The same applies to the Z Open Editor and Zowe Explorer extension.

Advanced WSL setup steps for Zowe CLI

If you are working with your Git repo from WSL/Ubuntu, you can also install Zowe CLI for additional automation scripts, which requires Node.js.

```
sudo apt install nodejs npm
```

To avoid a conflict with the Windows path that automatically gets appended to your Ubuntu path, create the file **`/etc/wsl.conf`** with the following content:

```
[interop]
appendWindowsPath = false
```

After you save the file, restart WSL with **`wsl --shutdown`** from a Windows Powershell. Restart WSL/Ubuntu and run the following commands:

```
mkdir ~/.npm-global
npm config set prefix '~/.npm-global'
```

Edit your ~/.profile file to add the following code at the end:

```
export PATH=~/.npm-global/bin:$PATH
```

Then, run the following command:

```
source ~/.profile
npm install -g @zowe/cli@zowe-v1-lts
```

Now you can install additional plug-ins that are needed for your work.

Configuring your development workspace with Ansible

With Ansible available in your development workspace, you can run the configuration scripts that are provided in the **`ansible`** folder.

Prerequisites

If you do not have Ansible, see Configuring your development workspace manually and shell scripts for instructions on how to manually create and run some of the commands. You can review the Ansible scripts to see what interactions are scripted there.

Ensure that you have configured Ansible and installed the Red Hat Ansible Certified Content for IBM Z. For more information, see Red Hat Ansible Certified Content for IBM Z.

Configuring the connection details for your z/OS system

Edit the file **ansible/inventories/inventory.yml** in the repository that you cloned by replacing the **ansible_host** IP address with the IP address of your system. If you configured a different user than **ibmuser**, replace it too.

You can edit the file directly in the repository folders. Or you can copy the **inventories** folder to your home or **etc** directory. If you do that, replace the path to that location for the -i parameter in all the following command examples.

If you use IBM Wazi as a Service Development and Test VSI, edit the entry **devtest1**.

In the following steps of this tutorial, we will use the hostname **devtest1** for all commands and examples, and you need to replace it with the hostname you chose to use.

Verify that you can use Ansible by executing a ping command:

```
cd ansible
ansible -i inventories devtest1 -m ping
```

Using an Ansible playbook to initialize your local development workspace

When the command succeeds, run your first playbook with the following command:

```
ansible-playbook -i inventories --extra-vars "host=devtest1" initialize-local-files.yaml
```

1. The command prompts you for the password and generates several files. Enter the password of the user that you specified in the previous step. The default password for **IBMUSER** is **sys1**.
2. For the version of Zowe CLI profiles that you want to create, confirm the default value **v7**.
3. For the editor that you are using, enter **che** if you are in Wazi for Dev Spaces.
4. Finally, it prompts if you want to overwrite previously created files. If you modified Zowe CLI profiles in a playbook, enter **yes** to replace the playbook, or enter **no** to generate a temporary file that you can rename.

When the playbook finished executing, scroll back up and carefully review the output because it contains instructions for the following tasks to complete the setup:

- Finish the Zowe CLI setup depending on how you answered the questions above. If you chose to not automatically overwrite, then it will provide the copy command that you need to run to do so now. For Zowe CLI **v7**, make sure that the playbook generated profiles for your host **devtest1** and placed them for you in **~/.zowe/zowe.config.json**. You must run the **zowe config update-schemas** command provided to complete the Zowe configuration.
- Complete the setup of the IBM Z Open Debug launch that was created.
- Print JSON values that you need to add to your IBM Wazi for Dev Spaces or VS Code user settings if you want to use the debugger.

Switch to Zowe Explorer to interact with z/OS:

1. Click the Refresh icon () from the **Data Sets** view.
2. Click the plus (+) icon to add the profile **devtest1.rseapi** that is created by the script to the view.
3. Click the Search icon next to the new profile entry, and click **Create a new filter**.
4. Enter **IBMUSER** or the username you use for this tutorial as the filter name to see existing data sets. Note that it might be an empty list.
5. Right-click and create a data set such as a PDS **IBMUSER.TEST**. Then, you can add files, edit, and save the files.

For more information about using Zowe Explorer, see Interact with Zowe Explorer.

Using Ansible to prepare and build an application

Run the following playbook to create data sets, copy source files, and build the SAM application.

```
ansible-playbook -i inventories --extra-vars "host=devtest1" dbb-sam-build.yml
```

When you are prompted for the editor that is used for Debug, enter **vscode** for IBM Wazi for Dev Spaces and VS Code.

Review the playbooks and ensure that it performs the following tasks by using templates for JCL and Groovy files:

- Executes the JCL to allocate data sets.

- Copies program files, JCL for running and debugging, and Groovy build scripts from the local directions to z/OS UNIX and data sets.
- Runs a build by using IBM Dependency Based Build with a Groovy build script.

Refresh the Zowe Explorer view to see the data sets and members that are created for the **SAM1** and **SAM2** sample COBOL applications. For example, expand the data set **IBMUSER.SAMPLE.COBOL** and click **SAM1** to open the program in the COBOL editor of Z Open Editor.

You can find the same source for this application in the local folder **COBOL**. The folder **COPYBOOK** is also used by the script. Open **SAM1.cbl** in the editor and explore the Z Open Editor editing capabilities in Making COBOL and PL/I code changes with either the local or remote copy of the program.

Using Ansible to copy test data and run an application

Run the following command to upload test data and run the application:

```
ansible-playbook -i inventories --extra-vars "host=devtest1" dbb-sam-run.yml
```

After the test data is uploaded, you can run the application again from Zowe Explorer by expanding the PDS **IBMUSER.SAMPLE.JCL** and right-clicking the **RUN** JCL, and selecting **Submit Job** from the menu.

A notification is displayed in IBM Wazi for Dev Spaces or VS Code, and you can click to see the job inside the Zowe Explorer **Jobs** view. Expand the view to review the spool files that are generated by the job. **IBMUSER.SAMPLE.JCL** also contains a **DEBUG** JCL that you can use when you run a Debug session from Z Open Editor.

For more information, see Tutorial: Improve your development productivity with Ansible.

Configuring your development workspace manually and shell scripts

If you do not have Ansible available on your development machine or no public ssh key is configured for the VSI, you can still connect to the VSI by using VS Code with Zowe Explorer.

Creating RSE profiles

To create an RSE API profile, find the script **zowe/zowecli-create-profiles.sh**, and modify it with the IP address and port values that you obtain to run commands for creating z/OSMF and SSH profiles.

If you are working on a Windows machine, you need to use a shell that can run bash files, such as Git Bash, to run this script and the other scripts inside the **zowe** folder.

Zowe v2 team configuration file

1. Issue the command **zowe config init --global-config**. After you fill in the information, the **zowe.config.json** file will be added to your **~/.zowe** directory. If you enter a username and password for a mainframe service, your credentials are stored securely on your computer by default.
2. Edit the **zowe.config.json** file in a text editor that supports JSON files to update the host and port values and other information needed in the profiles that are listed within the config file.
3. If the user and password were entered in step 1, you can use your profiles to access Data Sets, z/OS UNIX files and directories, and Jobs. If those fields were entered and accepted during the **init** command, you will be prompted the first time you run CLI commands. If **autoStore:true** is in the config file, the credentials are stored securely on your computer.

Zowe v1 profiles

If a team configuration file is in place, v1 profile creation and use is not available.

Run the following Zowe CLI command to create a Zowe profile to access the z/OS virtual server instance. Make sure to replace the IP address with the address of your instance.

```
zowe profiles create rse-profile devtest1.rseapi --host 10.x.x.x --port 8195 --user ibmuser --pass
sys1 --bp rseapi --protocol https --reject-unauthorized false --ow
```

For more information, see Using command line to interact with z/OS.

Using Zowe Explorer to interact with z/OS

After you create an RSE API profile, you can switch to Zowe Explorer and open the z/OS UNIX folder **/u/ibmuser**. Inside that folder, go to the subfolder **.ssh** and open the file **authorized_keys**.

If you cannot access the VSI through SSH earlier, you can edit this file and append you public SSH to it. Then, try the ssh command again to ensure you can log on.

1. Click the Refresh icon () from the **Data Sets** view.
2. Click the plus (+) icon to add the profile **devtest1.rseapi** that is created by the script to the view.
3. Click the Search icon next to the new profile entry, and click **Create a new filter**.
4. Enter **IBMUSER** or the username that you use for this tutorial as the filter name to see existing data sets. Note that it might be an empty list.
5. Right-click and create a data set such as a PDS **IBMUSER.TEST**. Then, you can add files, edit, and save the files.

For more information about using Zowe Explorer, see Using the Zowe Explorer views.

The folder **groovy** contains scripts that you can use to run a full build of the COBOL application that is called **SAM1** and run it. To run the COBOL application, make sure that you run the **zowe/zowecli-create-profiles.sh** script to create a required SSH profile.

Then, you can open, edit parameters, and run the following scripts in the **groovy** folder:

- **dbb-sam-prepare-and-build.sh** configures and runs the build
- **dbb-same-run.sh** runs the application.

Running Dependency-Based user builds from Z Open Editor

Z Open Editor user build allows you to run builds directly out of the editor remotely on z/OS UNIX System Services with IBM Dependency Based Build.

1. Run Ansible automation to remotely set up user build on the z/OS UNIX system of the VSI by running the playbook:

   ```
   cd ansible
   ansible-playbook -i inventories --extra-vars "host=devtest1" dbb-prepare-userbuild.yml
   ```

2. Review the script and its output. You can also set up user build manually as documented in Setting up the user build.

   The script clones a Git repository in **/u/ibmuser/projects** and copies a configuration file into that repository. This configuration is specific to the z/OS system that you are using. All the required values are provided for the z/OS development and test stock image in the Ansible inventory by the file **ansible/inventories/host_vars/devtest1.yml**.

   The output of the **dbb-prepare-userbuild.yml** playbook provides JSON to be copied into your VS Code user settings. They contain **user-** and **system-specific** values for Z Open Editor to start a build.

3. Open the **zapp.yaml** file and review the **dbb-userbuild** entry.

   1. If you are not using IBM Wazi Sandbox, remove the value **-DBB_PERSONAL_DAEMON** from the command.
   2. Check the **buildScriptArgs** and ensure that the **application** value matches the name of the sample repository folder that you cloned and opened in VS Code at the beginning of this tutorial. The default value is **zopeneditor-sample**.

4. Open the **SAM1** program in the editor and right-click **Prepare IBM User Build**. The **Output** view will show the progress of creating folders and copying over properties from the local **application-conf** directory, which needs to be done only once or after the settings are changed.
5. After the operation succeeds, right-click **Run IBM User Build** to build the program **SAM1** and download the logs file.

You can now make program changes with Z Open Editor and build the application with User Build. To run the application, you can also use the Ansible playbook that copies required test data over to MVS. The Ansible playbook then executes a JCL that is also available in the **IBMUSER.SAMPLE.JCL(RUN)** partitioned data set. After you have the test data in place, you can repeatedly right-click to execute the **RUN** JCL in Zowe Explorer.

Running a Debug session from Z Open Editor

To start a Debug session in VS Code, you need to first install the two IBM Z Open Debug extensions.

When you run the playbook **`initialize-local-files.yaml`**, it prints three JSON properties that you need to add to your VS Code user settings. Replace **`10.x.x.x`** with your the address of your instance:

```
"zopendebug.connection.connectionName": "10.x.x.x",
"zopendebug.connection.connectionPort": 8192,
"zopendebug.connection.userName": "ibmuser",
```

The z/OS development and test stock image runs the REST services that are required by the Debug client by using a self-signed SSL certificate. You need to enable VS Code to accept such a certificate as valid. You can restart VS Code by using command line with a special parameter, for example:

```
code --ignore-certificate-errors /User/username/zopeneditor-sample
```

The **`initialize-local-files.yaml`** also creates VS Code launches that provide the IP address and special ports that are needed to connect to a debug session that is running on z/OS. Move the generated file **`~/.zowe/launch.json`** to the **`.zowe`** folder in your Git repository with the following command:

```
mv ~/.zowe/launch.json ../.vscode/launch.json
```

If you did not run the Ansible script, you can create the launch manually by editing and copying the file **`ansible/templates/debug-launch.j2`** and replacing values for username, IP address, and port (default: 8194).

The **`dbb-sam-build.yml`** Ansible playbook that you run earlier also generates and copied a JCL to the data set **`IBMUSER.SAMPLE.JCL`** called DEBUG that can be used for building and running a debug session of the SAM1 COBOL application on z/OS.

Similarly, you can manually create the JCL by filling in the variables in **`ansible/templates/DEBUG.j2`** with the values list in the file **`ansible/inventories/host_vars/devtest1.yml`**.

When you have the VS Code user settings **`.vscode/launch.json`**, and **`IBMUSER.SAMPLE.JCL(DEBUG)`** in place, you can create a Debug Profile in VS Code with the **`cmd-shift-p`** (Mac) or **`ctrl-shift-p`** (Windows) command, and type or select **Debug: IBM Z Open Debug Profiles** view.

1. Close the Welcome screen, click **Create Profile**, and select the Batch option. A form with the connection details based on your user settings is displayed.
2. Check that a blue checkmark icon appears next to the form with the label **`Connected`**.

   Note: If the checkmark icon does not appear, make sure that you start VS Code with the **`--ignore-certificate-errors`** parameter, that you are connected to your VPN, and that the VS Code user settings are entered correctly.
3. Click **Create**. A new profile in a table with a slider button in the **Activate** column is displayed.
4. Click the slider and wait until you see **Active** and a green checkmark next to the entry in the **Connection** column.

You can now start and connect to a z/OS Debug session.

To start a Debug session on z/OS, you can use the DEBUG JCL that is generated and copied to **`IBMUSER.SAMPLE.JCL(DEBUG)`** by the **`dbb-sam-build.yml`** Ansible playbook, or start the session manually.

1. Switch to the Zowe Explorer activity group in VS Code.
2. In the **Data Sets** view, repeat the steps from earlier to get the data sets from **`IBMUSER`**.
3. Expand the **`PDS IBMUSER.SAMPLE.JCL`**.
4. Right-click the file **`DEBUG`** and select **Submit Job**. A notification displays the ID of the new job.
5. Click the ID to show the job in the Zowe Explorer **Jobs** view and open its spool files. The job should remain in Active state.

Then, use the VS Code Debug activity group to connect to the Debug session.

1. From **Run and Debug**, select **List parked IBM Z Open Debug sessions** and click **Play**.
2. Enter the password of the user **`IBMUSER`**. The default value is **`sys1`**.

3. The **Debug Console** view is displayed in VS Code that shows you whether it finds a parked debug session. It shows a session ID and a state of Parked.
4. If a parked session is not available, run the launch again. If it is not successful, investigate the Job's spool files again to see what might be wrong.
5. When you see a parked session, select **Connect to parked IBM Z Open Debug session** from **Run and Debug**, and click **Play**.
6. Enter the password again.
7. The expanded source gets opened in a VS Code editor tab with the first line highlighted.
8. The session is paused and you can now set breakpoints. For example, set a breakpoint in line 442 in paragraph "100-PROCESS-TRANSACTIONS" by clicking next to the line number until a red dot appears.
9. When you see a hovering toolbar, you can click **Continue** to jump the toolbar to the breakpoint.
10. Use the **Variables** view to inspect the values of Locals and Registers.
11. Click **Step over** and **Step into** to step through the program.
12. To finish the session, press **Continue** until the program ends, or click **Stop**.

Check the Zowe Explorer jobs view to confirm that the Debug session has ended. You can edit the program now, run a User Build, or build Ansible playbook and Debug again.

Developing and testing with IBM Wazi for Eclipse or IBM Developer for z/OS

Learn how to develop and test with IBM Wazi for Eclipse or IBM Developer for z/OS.

- Connecting IBM Wazi for Eclipse or IBM Developer for z/OS

  Ensure that you are in the z/OS projects perspective to complete the steps.

- Running Dependency-Based User Builds from IBM Wazi for Eclipse or IBM Developer for z/OS

  Find instructions on how to run dependency-based user builds from IBM Developer for z/OS.

- Running a Debug session from IBM Wazi for Eclipse or IBM Developer for z/OS

  Find instructions on how to run a Debus session from IBM Wazi for Eclipse or IBM Developer for z/OS.

Connecting IBM Wazi for Eclipse or IBM Developer for z/OS

Ensure that you are in the z/OS Projects perspective to complete the steps.

To connect with the Eclipse clients, you need to create a connection.

1. When you see a window for setting up the perspective, select the options for dark or light editor, choose the Eclipse editor, and click **Finish**.
2. The side panel shows the **Remote Systems** view. You can create a connection in one of the following ways:
   - Expand **New Connections**, and right-click the **z/OS** option.
   - Select the icon for new connections. Then, select **z/OS** and click **Next** in the wizard dialog.
3. Enter the **`Host Name`**, and click **Next**. The default connection name is the hostname, but you can change it to better remember what the connection is for. Optionally, you can add a description to it.
4. Add the Daemon Port **`8137`**, and click **Finish**.
5. Right-click the connection that you created, and select **Properties**.
   1. Choose **Port Overrides**. Make sure that the override RSE server port is enabled and uses the **`Port Override: 8138`**.

2. Click **Apply and Close**.
6. Check that the connection works. Right-click and click **Connect** to log in.

Go to **Git Repositories** to import the `zopeneditor-sample` repository.

- If you haven't cloned the repository or want a new clone, complete the following steps:
    1. Click **Clone a Git repository**.
    2. Enter `https://github.com/IBM/zopeneditor-sample.git` and click **Next**.
    3. Select only `wazi-main` and click **Finish**.
    4. Right-click the imported repository in the **Git Repository** view and select **Create z/OS Project**.
    5. Switch back to **z/OS Project** to see the project.
- If you already have a clone, complete the following steps:
    1. Click **Add a Git repository**.
    2. Search for your existing clone in the directories.
    3. Click **Add**.

Open the file `COBOL/SAM1.cbl` to edit the program.

Running Dependency Based User Builds from IBM Wazi for Eclipse or IBM Developer for z/OS

In this tutorial, you can find instructions on how to run dependency based user builds from IBM Developer for z/OS. For more information, see Starting a User Build

To run a Dependency Based User Build, complete the following steps:

1. In the z/OS Projects perspective, click the project and select **Dependency Based Build** > **Generate Property Group**.

   For this exercise, select **Local COBOL Settings** and click **Finish**.

2. Right-click `SAM1.cbl` in the z/OS Projects perspective, or in the editor, and select **Dependency Based Build** and then **Configure User Build**. A wizard comes up to help configure the DBB User Build run.

3. Complete the **Configure User Build Operation** page, and then click **Next**.
    1. Select the zOS system.
    2. Select the build script.
    3. Browse the remote system `/u/ibmuser/projects/dbb-zappbuild/build.groovy`.
    4. Enter the build sandbox folder `/u/ibmuser/projects`.
    5. Enter the build destination HLQ `IBMUSER.SAMPLE`.
4. In the **CFile Attributes** page, use the default options, and then click **Next**.

5. In the **Log File** page, use the default options, and then click **Next**.

6. Specify additional parameters in the **Script Parameters** page, and then click **Next**.
    - Add a Dependency Based Build command
        - Option: --application
        - Value: zopeneditor-sample
7. In the **Additional File to Load** page, expand the project `zopeneditor-sample`, and select the `application-conf` folder. And then, click **Next**.

8. View the summary of User Build Operation, and then click **Finish**.
    1. If `SAM1.cbl` already exists, select the option **Overwrite** and click **OK**.
    2. If resources already exist, click **Overwrite All**.
    3. Overwrite the temporary logs for previous builds, click **OK**.

4. DBB User Build Information is displayed, click **OK**.
5. You can see the following information in the console log:
    - Build State: CLEAN
    - Total files processed: 1

Running a Debug session from IBM Wazi for Eclipse or IBM Developer for z/OS

To start a Debug session on z/OS, you can edit the file **`JCL\DEBUG.jcl`** by commenting out the line **`SET TESTOPT='/TEST(,,,DBMDT:*)'`** with the comment that it should be used with Eclipse and commenting the respective line that defines **`TESTOPT`** for VS Code's Z Open Debug. Then, use the Remote System explorer to copy the file to a data set such as **`IBMUSER.SAMPLE.JCL(DEBUG)`**.

If you used the Ansible playbook **`ansible/dbb-sam-build.yml`** and selected **`eclipse`** as the editor that you use for debugging, then it was already done for you.

1. Go to **Remote Systems** and expand the connection **MVS Files > My Data Sets**.
2. Expand the PDS **`IBMUSER.SAMPLE.JCL`**.
3. Right-click the file **`DEBUG`** and select **Submit Job**.
4. In the **Job Submission Confirmation** window, click **Notify**.
5. The **Remote Console** view is updated with the information of the job that was submitted. Click the job number to check that the job remains in Active state.

If the **Confirm Perspective Switch** window is displayed, click **Switch**, otherwise the Debug perspective opens automatically.

1. The source gets opened with the first line highlighted.
2. The session is paused and you can now set breakpoints. For example, set a breakpoint in line 425 in paragraph **`100-PROCESS-TRANSACTIONS`** by right-clicking on that line and selecting the **Add Breakpoint** option until a dot appears.
3. In the **Debug** toolbar, click **Resume** to jump to the breakpoint.
4. Use the **Variables** view to inspect the values.
5. Click **Step over** and **Step into** to step through the program.
6. To finish the session, click **Resume** until the program ends, or click **Terminate**.
7. If a **Debug Engine Message** window is displayed, click **OK**.

Go back to the z/OS Projects perspective and check the job submitted to confirm that the Debug session has ended. You can edit the program now, run a User Build, or build Ansible playbook and Debug again.

Deploying and using IBM Wazi for Dev Spaces from a Red Hat OpenShift cluster in your VPC

You can use IBM Wazi for Dev Spaces to move all your development into the IBM Cloud as an alternative to using VS Code or Eclipse from your development system.

The advantage is that all development tools are installed and configured in the IBM Cloud. Developers do not need to install any tools, but need only a browser to develop applications.

- Creating a Red Hat OpenShift cluster

  The images that are required for Wazi are in the public IBM and Red Hat catalogs that need to be pulled over the internet. A public gateway makes the internet reachable for your subnets for outgoing traffic, but it does not expose your z/OS VSI to it.

- Deploying and configuring IBM Wazi for Dev Spaces

The IBM Operator Catalog is the recommended approach to install Wazi for Dev Spaces because it is periodically updated with the latest set of IBM Operators, which includes Wazi for Dev Spaces and more.

- Creating a development workspace

  To build and run applications, you need a workspace that not only has the IBM Wazi tools, but also Ansible for running the playbooks provided with the sample application.

Creating a Red Hat OpenShift cluster

The images that are required for Wazi are in the public IBM and Red Hat catalogs that need to be pulled over the internet. A public gateway makes the internet reachable for your subnets for outgoing traffic, but it does not expose your z/OS VSI to it.

If you do not have a public gateway, complete the following steps to create one:

1. Log in to IBM Cloud.
2. Go to **Public gateways for VPC**, and click **Create** to create a public gateway in your VPC.
3. Go to **Subnets**, drill into each subnet, and attach the public gateway with the slider.
4. Go to Red Hat OpenShift cluster to create the cluster. Use the default settings except for the following fields:
   - OCP: `version 4.10 or 4.11`
   - Infrastructure: click **VPC**. Choose the same VPC that runs your instance and the same Cloud Object Storage.
   - Location - Resource group: Choose the same group that you use for your other resources such as VSI. Choose the worker zone where your VPC and VSI are running.
5. Click **Create**.
6. When the deployment finishes, you can click `OpenShift web console` on the cluster screen to open OpenShift.

Deploying and configuring IBM Wazi for Dev Spaces

The IBM® Operator Catalog is the recommended approach to install Wazi for Dev Spaces because it is periodically updated with the latest set of IBM Operators, which includes Wazi for Dev Spaces and more.

To install the IBM Operator Catalog, you must be a cluster administrator. You can add CatalogSource objects to your cluster using either the OpenShift® web console or OpenShift CLI to enable the IBM Operator Catalog in the cluster, and then install the Wazi for Dev Spaces Operator.

Follow the instructions in the following topics:

- Deploying Wazi for Dev Spaces by using the IBM Operator Catalog
- Creating a Wazi for Dev Spaces instance via an Operator

For more information, see the Release notes and known issues for Red Hat OpenShift Dev Spaces.

Creating a development workspace

To build and run applications, you need a workspace that not only has the IBM Wazi tools, but also Ansible for running the playbooks provided with the sample application.

To create a workspace, follow the instructions in these tutorials:

- Tutorial: Creating and getting started with your workspace
- Tutorial: Improve your development productivity with Ansible

Known limitations

Before you use Wazi as a Service, make sure that you are aware of the following known limitations.

Limitations of Wazi Image Builder and z/OS custom images

Following are the known limitations of Wazi Image Builder and z/OS custom images.

- When using z/OS Extraction Utilities, specifically feucvoli, to find data sets from source environments for extraction, uncataloged data sets are not automatically detected and included. In cases where uncataloged data sets are needed, such as JES2 Checkpoint or JES2 spool, the disk volumes containing these data sets must be included manually, unless there are other cataloged data sets on these volumes that have already been identified. If additional volumes that contain uncataloged data sets are needed, the system programmer of the z/OS source environments must make necessary adjustments to the list of volumes.
- Cloud-init is not enabled in z/OS Wazi aaS custom images.
- You cannot validate the checksum of the custom images that are deployed from Wazi Image Builder.
- Cross-region bucket is not supported for custom image deployments. You need to upload to a regional bucket.
- The **`zOS-cloud-prepare`** tool does not support the syntax of substringing system symbols.

Limitations of IBM Cloud VPC and z/OS virtual server instances

Following are the known limitations of z/OS virtual server instances in the IBM Cloud Virtual Private Cloud (VPC).

- Deleting a subnet through the user interface might not work as expected when you delete an existing VPC. You can use the command **`ibmcloud is subnet-delete`** as the workaround.
- IBM z Systems Integrated Information Processor (zIIP) is not supported, and vCPUs for z/OS virtual server instances are general-purpose processors only.
- Only the following device types are supported on the z/OS virtual server instances:
    - 3270-X
    - 3279
    - 3390B
    - Open Systems Adapter (OSA)
- When you create a z/OS virtual server instance, **User data** parameters are not processed by the z/OS operating system.
- Removing block storage from a z/OS virtual server instance that is not fully initialized results in the instance failure. The instance cannot be recovered after this type of failure.
- Accessing your z/OS virtual server instance by using a VNC console on the IBM Cloud UI is not supported.
- You must store your data in an attached block storage device or volume because updating a z/OS virtual server instance dynamically is not supported. To update your z/OS virtual server instance, use one of the following alternatives:
    - Create an instance with the new version of the z/OS stock image, and then attach the block storage.
    - Create a custom z/OS image with the new fix in your on-premises environment, import the image, and then attach the block storage.
- The reports generated in the IBM Sub-Capacity Reporting Tool (SCRT) are not valid.
- z/OS Resource Measurement Facility (RMF) Data Gatherer for SMF record types 70 - 79 has the following restrictions:
    - **CPU information**
        - SMF record type 70 subtype 1 contains one physical and one logical CPU (similarly as for z/VM systems with enabled VMGUEST option). Postprocessor CPU Activity report is impacted.
        - Monitor III measurement table ERBCPCDB consists of Header Section and Home LPAR section only. Monitor III CPC Capacity report is impacted.
    - **Logical Control Unit (LCU) is not supported**
        - SMF record type 74 subtype 1 does not contain LCU data. Postprocessor Device Activity report is impacted.
        - SMF record type 78 subtype 3 cannot be gathered due to missing LCU data. Postprocessor I/O Queuing Activity report is impacted.

        - SMF record type 79 subtype 14 cannot be gathered due to missing LCU data. Monitor II I/O Queuing Activity report is impacted.
        - Monitor III measurement table IQDG3 cannot be gathered due to missing LCU data. Monitor III I/O Queuing report is impacted and shows message `No hardware data`.
    - **Retrieval of Crypto Measurement data and Crypto Domain Measurement data is not supported**
        - SMF record type 70 subtype 2 does not contain crypto card information. Postprocessor Crypto Hardware Activity report is impacted.
        - Monitor III measurement table CRYG3 does not exist. Monitor III Crypto reports CRYACC, CRYOVW, and CRYPKC are impacted.
    - **Dynamic CHPID Management (DCM) is not supported**
        - SMF record type 73 contains indicator SMF73CMH (Bit 0 of SMF73SFL) with meaning DCM supported by hardware. This indicator will be OFF. Postprocessor Channel Path Activity is impacted.
        - SMF record type 79 subtype 12 contains indicator SMF79CCMH (Bit 2 of SMF79CFLG1) with meaning DCM supported by hardware. The indicator will be OFF. Monitor II Channel Path Activity report is impacted.
        - Monitor III measurement table CPDG3 contains indicator CPDCMH (Bit 4 of CPDFlg) with meaning DCM supported by hardware. This indicator will be OFF. Monitor III Channel Path Activity report is impacted.
    - **Interrupt Delay Time facility is not supported**
        - SMF record type 79 subtype 9 contains indicator R799FID (Bit 7 of R799CNX) with meaning Interrupt Delay Time facility is provided. This indicator will be OFF. Hence the Interrupt Delay Time stored in field SMF79IDT will be zero. Monitor II DEV and DEVV reports are impacted.
        - SMF record type 74 subtype 1 contains indicator R74FID (Bit 3 of SMF74ENF) with meaning Interrupt Delay Time facility is provided. This indicator will be OFF. Therefore, the Interrupt Delay Time stored in field SMF74IDT will be zero. Postprocessor Device Activity report is impacted (AVG INT DLY is missing).
    - **Utility string facility is not supported**
        - SMF record type 73 contains indicator SMF73PNI (Bit 7 of SMF73FG4) with meaning PNET ID SMF73NT1 and SMF73NT2 are valid. This indicator will be OFF. Hence PNET IDs SMF73NT1 and SMF73NT2 will be zero. Postprocessor Channel Path Activity report is impacted.
        - SMF record type 74 subtype 9 contains indicator R749PNVA (Bit 0 of R749FLAG) with meaning PNET ID R749NET1 and R749NET2 are valid. This indicator will be OFF. Therefore, PNET IDs R749NET1 and R749NET2 will be zero. Postprocessor PCIe Activity report is impacted.
        - Monitor III measurement table CPDG3 contains indicator CPDUSF (Bit 6 of CPDCFLG) with meaning Channel path data utility string failure. This indicator will be ON. For this reason, PNET IDs CPDPnetID1 and CPDPnetID2 will be zero. Monitor III Channel Path Activity report is impacted.
        - Monitor III measurement table PCIG3 contains indicator PCINetIDVal (Bit 0 of PCIValidFlag) with meaning PNET IDs are valid. This indicator will be OFF. PNET IDs PCIPnetID1 and PCIPnetID2 will be zero. Monitor III PCIe Activity report is impacted.
    - **EADM compression measurement facility is not supported**
        - SMF record type 74 subtype 10 and Monitor III measurement table SCMG3 cannot be gathered. Postprocessor and Monitor III EADM Activity reports are impacted.
    - **Channel Path Measurement Facility (CPMF) is not supported**
        - SMF record type 73 contains indicator SMF73CMF (Bit 5 of SMF73CFL) with meaning CPMF is available. This indicator will be OFF. Postprocessor Channel Path Activity report is impacted.
        - SMF record type 79 subtype 12 contains indicator R79CCPMF (Bit 0 of SMF79CFLG1) with meaning CPMF is available. This indicator will be OFF. Monitor II Channel Path Activity report is impacted.
        - Monitor III measurement table CPDG3 contains indicator CPDCPMF (Bit 2 of CPDFlg) with meaning CPMF is available. This indicator will be OFF. Monitor III Channel Path Activity report is impacted.
    - **LIST DATA interface IDCSS01 does not return data**
        - SMF record type 74 subtype 5 contains fields R745CSC and R745CRTN, which show the error information and return code of the IDCSS01 interface. Postprocessor Cache Subsystem Activity report is impacted.
        - Monitor III measurement table CATG3 contains fields CATG3_Stat_Code and CTAG3_RC_IDCSS, which show the error information and return code of the IDCSS01 interface. Monitor III Cache Detail and Monitor III Cache Summary reports are impacted.
- Only one network interface is supported in each z/OS virtual server instance.
- The Central Processor Assist for Cryptographic Function (CPACF) only supports clear keys in z/OS stock images. Protected keys and secure keys are not supported.
- PCIe cards such as the Cryptographic Coprocessor, Acceleration, and Compression are not supported.
- Disks are not available to share between z/OS virtual server instances.
- The multisystem sysplex (MULTISYSTEM) is not supported and the system must be either in a single-system sysplex (MONOPLEX) or in XCF-local mode. You can use a system IPL parameter **`PLEXCFG=(MONOPLEX,XCFLOCAL)`** in the **`IEASYSxx`** parmlib member, which enables the system to come up in XCF-local mode when MONOPLEX mode is failed due to issues with the sysplex couple data sets (CDS). For more information, see PLEXCFG.

- The only supported GRS specification is **GRS = NONE** in the **IEASYSxx** parmlib member, which means that the system is not in a global resource serialization complex.
- The Coupling Facility Resource Management (CFRM) policy and its related couple data sets (CDS) must not be used because the coupling facilities are not supported. You cannot use the **COUPLExx** parmlib member to define the type for couple data set in the CFRM. For example, you must not issue **SETXCF COUPLE, TYPE=CFRM**.
- z/OS Container Extensions (zCX) is not supported and restricted for use. zCX is not intended for this environment.
- **HWNAME**, **LPARNAME**, and **VMUSERID** in the **IEASYMxx** parmlib member or in **LOADxx** member are not supported.
- The percentage bar graph might indicate that the memory is fully consumed. This percentage represents the full allocation of memory that is reserved for z/OS, rather than the actual memory usage within z/OS.

For other known limitations of IBM Cloud VPC, see Service limitations.

Known problems

Known problems are documented here with some suggested workaround.

Expired CA Certificate in Wazi as a Service z/OS image

The CA Certificate may expire if you are not using the latest Wazi as a Service stock image.

Symptom: Connections through TN3270 no longer work although SSH to z/OS works.

This problem does not occur in the latest z/OS image created after December 2023.

To resolve this issue, do the following:

1. SSH to z/OS Unix System Services.
2. Verify if your CA Certificate has expired by using the following command:

```
tsocmd "racdcert certauth list(label('VSICA'))"
```

3. Verify the end date of the certificate and if it is a past date, proceed with the next steps.
4. (Optional) take a snapshot of your instance.
5. Update the z/OS symbolic parm used for generating the new CA Certificate by running the following commands in order:

```
dsed "s/CERTEXPIRE='....-..-..'/CERTEXPIRE='2025-02-04'/g" 'sys1.parmlib(ieasym00)'
```

```
dsed "s/CERTEXPIRE='....-..-..'/CERTEXPIRE='2025-02-04'/g" 'k2.parmlib(ieasymk2)'
```

Ignore the error "BGYSC4904E Unable to open dataset k2.parmlib(ieasymk2) for read" that the previous command may generate.

6. Ensure that either `SYS1.PARMLIB(IEASYM00)` or `K2.PARMLIB(IEASYMK2)` is updated correctly by using the following commands, respectively:

```
dcat 'sys1.parmlib(ieasym00)' | grep CERTEXPIRE
```

```
dcat 'k2.parmlib(ieasymk2)' | grep CERTEXPIRE
```

Alternatively, if “dcat” is unavailable on the z/OS image, use the following commands in order:

```
cp "//'sys1.parmlib(ieasym00)'" '/tmp/ieasym00'
```

```
cat /tmp/ieasym00 | grep CERTEXPIRE
```

or

```
cp "//'k2.parmlib(ieasymk2)'" '/tmp/ieasymk2'
```

```
cat /tmp/ieasymk2 | grep CERTEXPIRE
```

The CERTEXPIRE symbol value is contained in either **sys1.parmlib** or **k2.parmlib**. Therefore, one of the cat commands reveals a null response.

7. Ensure that you see the following line:

```
SYMDEF(&CERTEXPIRE='2025-02-04')
```

If both the "dsed" commands fail to update **SYS1.PARMLIB(IEASYM00)**, then do not proceed with the next steps. However, you can proceed if one of them fails.

8. Alter the existing CA Certificate label to trigger an automatic regeneration by using the following command:

```
tsocmd "racdcert certauth alter(label('VSICA')) newlabel('VSICA_REGEN')"
```

In IBM Cloud:

1. Open the serial console and shutdown z/OS by using the following command:

```
oprmsg '%netv shutsys'
```

2. Wait for the shutdown to complete and then from the cloud console or CLI, perform a VSI stop followed by a VSI start.
3. Verify the CA Certificate expiration by using the following command:

```
tsocmd "racdcert certauth list(label('VSICA'))"
```

4. Ensure that the end data has changed.

Incorrect RDS port displayed in Open Debug Profiles when you switch from one z/OS machine to another

This issue happens when you have at least two different z/OS machines that are configured with different local or internal Rational Directory Server (RDS) port values.

When you run an Open Debug session in VS Code and switch connection from one machine to another machine, the Rational Directory Server port in the debug profile is not changed.

To solve this issue, edit the profile and click **Update** to update to the correct Rational Directory Server port.

Custom image deployment fails due to four-hour import limitation

When you use Wazi Image Builder to deploy custom images to IBM Cloud, the import of the image must completed within a four-hour time limit.

When you import larger custom images from Cloud Object Storage, a timeout might occur if the location of the COS instance is not close to the region that hosts the custom image.

To solve this issue, store the custom images in a Cloud Object Storage instance location that is close to the region in which the custom image is created, so that the import can finish in less than 4 hours.

Cannot IPL custom image that is built by Wazi Image Builder

This issue might happen when Wazi Image Builder fails to search and replace the IP address and routing configuration.

When you deploy the custom image to IBM Cloud VPC, the extracted system is automatically modified for the networking of an on-prem z/OS instance to work in an IBM Cloud VSI environment. The IP address and route configuration need to be located within your TCP/IP profile data set and the data set containing this information must be a partitioned data set (PDS).

To solve this issue, configure your z/OS in a way where the IP address and routing information are directly in a TCP/IP data set members and the data set is a PDS.

For more information, see Setting up networking for on-premises z/OS environment.

Custom image IPLs but cannot connect via SSH or TN3270

This issue might happen for the following reasons:

- The image is deployed by using incorrect **PARMLIB** and **IEASYM** suffix.
- The security settings of your z/OS virtual server instance do not include the required ports.
- The services on z/OS (such as SSH and TN3270) are not running.

To solve this issue, check the following settings.

1. Set up your networking for on-premises z/OS environment by following the instructions.
2. Specify the correct **PARMLIB** and **IEASYM** suffix when you deploy the custom image.
3. Configure security group to define the required ports that are allowed for the instance.
4. Make sure the services on z/OS (such as SSH and TN3270) are running.

TN3270/SSH access to z/OS VSI fails after some time

This issue happens when you use TN3270 terminal emulator and SSH key to connect to your z/OS VSI.

The access fails after some time because the JES spool is filling up.

To solve this issue, you can add the following commands to your IPL sequence to check the JES spool periodically.

```
CLEAR AUTOMATIC COMMANDS
$CA,ALL
EVERY 6 HOURS, CLEAR SMFDUMP JOBS > 6 HOURS
$TA,T=23.45,I=21600,'$PJQ,JM=SMFDUMP*,H>6'
EVERY 6 HOURS, CLEAR USS CHILD TASKS > 6 HOURS
$TA,T=23.45,I=21600,'$PJQ,JM=BPXAS,H>6,PROTECTED'
DAILY CLEAR ALL OUTPUT IN MSGCLASS T
$TA,T=23.45,I=86400,'$POJQ,Q=T,PROTECTED'
DAILY CLEAR JOB OUTPUT OLDER THAN 10 DAYS
$TA,T=23.45,I=86400,'$POJQ,ALL,A>10,PROTECTED'
```

Alternatively, you can adjust NetView Timers to ensure that BPXAS spool entries are cleaned up automatically each day. Add the following line to **NETVIEW.STOCK.CNMCLST($TIMERS)** before the **EXIT** statement.

```
'AT 00:00:00,ROUTE=AUTO1,ID=NEWTMR,$TIMERS'
```

Then, run the following z/OS console command:

```
%NETV $TIMERS
```

Other JES spool entries can cause similar issues without more robust automation. If you want to clean up all entries in the JES queue every day, you can add the following lines to **NETVIEW.STOCK.CNMCLST($TIMERS)** before the **EXIT** statement.

```
'AT 00:00:00,ROUTE=AUTO1,ID=NEWTMR,$TIMERS'
'AT 00:30,ROUTE=AUTO1,ID=CLN1,MVS $POJQ,JM=*,DAYS>1'
```

Then, run the following z/OS console command:

```
%NETV $TIMERS
```

SSH hangs with high CPU usage on z/OS stock image

When you encounter a hanging SSH session, it might be due to high resource consumption.

To address this issue, open a Telnet 3270 session to connect to the instance. After you log on to TSO, enter SDSF, and then enter the DA panel. Use the PF11 key to navigate the columns until you see the **CPU%** column. Identify any tasks that might be taking up an excessive amount of CPU, and either cancel the task, or modify the task to use less resources.

Alternatively, you can consider provisioning an instance with a larger profile, such as 4x32.

Connection to z/OS VSI fails due to inadequate profile

This issue happens when the profile that you selected when you created the VSI is not adequate. z/OS virtual server instances require a minimum of 2 vCPUs x 16 GB RAM and one vCPU is reserved for running the service.

To solve this issue, you can resize the z/OS VSI, or create a new z/OS VSI with a minimum profile of **2x16**, and then try again. For more information, see Resizing a virtual server instance.

You can check the IPL status of z/OS by opening the serial console for the VSI. When z/OS has fully IPLed by using an adequate profile, you can verify connectivity via SSH using a command that is similar to **ssh ibmuser@<vsi ip address>**.

If the profile is **2x16** or greater and the connectivity problem still exists, try the following steps:

- Issue a command such as **oprmsg D A,L** from the serial console to get the status of the z/OS subsystems.

- Check the network connectivity between client and z/OS VSI, the type of IP address that is being used (private or floating point), and the security groups.

Connection to z/OS VSI fails due to missing PTFs

When you create a VSI with your custom image, a message that starts with **IEA303W** shows up in the serial console.

This issue happens when the PTFs that are required to run in the Cloud environment, as listed in the Prerequisites, are not applied in your z/OS system.

To solve this issue, make sure that you install all the PTFs that are identified with the SMP/E FIXCAT of **IBM.TargetSystem-RequiredService.AlternateHypervisors**. Installation of all these PTFs must be completed on the z/OS system before you deploy the custom image.

For more information, see IBM Fix Category Values and Descriptions.

Custom image deployment fails due to missing DASD devices

When no offline direct access storage devices (DASD) are available in the source z/OS during extraction, although the extraction might be successful, the deployment of this extracted image fails.

When you deploy a custom image, a volume called **ZDTDPY** is created. This volume needs to be mounted on an offline DASD device in the source z/OS. Therefore, there must be at least one offline DASD device available when you deploy the image.

You can follow the steps to solve the issue:

1. Add an offline DASD device to the source z/OS volume list.
2. Create a new version of the SYSRES component that contains the offline DASD device.
3. Create a new version of the image to replace the SYSRES component with the new version created in step 2.
4. Deploy the new image.

Getting support

Learn how to get help and support for IBM® Wazi as a Service.

Before you contact support, check out Known limitations and Known problems to see whether you can find a workaround for your problem.

Getting support for IBM Cloud z/OS dev and test services

For questions about IBM Cloud z/OS dev and test services, check out Getting help and support.

Getting support for IBM Wazi Image Builder

For questions about Wazi Image Builder, contact IBM Support. Read the IBM Support Guide for contact information, guidelines, and reference materials.

To open a support case, complete the following steps:

1. Go to the IBM Support site.
2. Click **Open a case**.
3. Enter your IBMid and password, and then you can see the IBM Privacy Statement.
4. Read the statement carefully, and click **I consent** if you agree with the statement.
5. In the **Open a case** page, enter the detailed information.
6. Click **Submit case** to submit the case.

Glossary

Find terms that you might be interested to know when you use IBM® Wazi as a Service.

access control list

A list that statelessly manages inbound and outbound traffic for a subnet through the use of rules. An access control list helps provide security at the subnet level.

cloud resource name (CRN)

A globally unique identifier for a specific cloud resource. The value is segmented hierarchically by version, instance, type, location, and scope, separated by colons.

component

A collection of artifacts, such as volumes, from an IBM Z platform. A component is reusable with different components to create more than one image.

compute

Infrastructure or resources that serve as the basis for building apps in the cloud.

Custom Resource

Custom Resource (CR) objects are created from CRDs that have been added to the cluster by a cluster administrator, allowing all cluster users to add the new resource type into projects.

While only cluster administrators can create CRDs, developers can create the CR from an existing CRD if they have read and write permission to it.

Custom resource definition

A custom resource definition (CRD) object defines a new, unique object type, called a kind, in the cluster and lets the Kubernetes API server handle its entire lifecycle.

data center (DC)

The physical location of the servers that provide cloud services.

data sets

In z/OS, a named collection of related data records stored and retrieved by an assigned name. Equivalent to a file.

devfile

A portable file that describes your development environment. With the devfile, you can define a portable developmental environment, such as the source code, IDE tools, application runtimes, and predefined commands, without the need for reconfiguration.

experimental offering

An offering that IBM makes available solely for evaluation and testing purposes, and might be unstable or not compatible with previous versions. An experimental offering can be discontinued with short notice. There are no warranties, SLAs or support provided, and experimental offerings are not intended for production use.

floating IP address

A public, routable IP address that makes use of 1-to-1 network address translation (NAT) so that a server can communicate with the public internet and private subnet within a cloud environment. Floating IP addresses are associated to an instance, for example, a virtual server instance, a load balancer, or a VPN gateway, by means of a virtual network interface card (vNIC).

globally unique identifier (GUID)

An algorithmically determined number that uniquely identifies an entity within a system.

hybrid cloud

A cloud computing environment that consists of multiple public and private resources.

IBM Wazi for Eclipse

IBM Wazi for Eclipse is an Eclipse-based modern, lean integrated development environment with program understanding, edit, debug, and build automation capabilities.

IBM Wazi for VS Code

Provides application developers with a set of capabilities to edit, build, and debug their z/OS programs by using a desktop IDE through a set of VS Code extensions.

IBM Wazi for Dev Spaces

Provides application developers with a set of capabilities to edit, build, and debug their z/OS programs by using the in-browser IDE within Red Hat OpenShift Dev Spaces.

Identity and Access Management (IAM)

The process of controlling access of authorized users to data and applications, while helping companies comply with various regulatory requirements.

image

A collection of components that are packaged together as a single deliverable. An image contains the z/OS core system and user-selected z/OS applications and subsystems and is used to be deployed to your target environment.

Image storage server

A type of server that can be used to store z/OS images for Sandbox instances. Sandbox uses SSH File Transfer Protocol (SFTP) server as the image storage server, and stores ADCD images on the server.

initial program load (IPL)

An initial program load (IPL) is the act of loading a copy of the operating system from disk into the processor’s real storage and executing it.

instance

A z/OS instance that is running on an emulated IBM Z hardware from a created image.

Interactive System productivity Facility (ISPF)

The menu-drive interface that offers a wide rang of functions to assist users in working with data files on the z/OS system.

Job Control Language (JCL)

JCL is the command language in z/OS. JCL identifies the program to be executed, the inputs that are required and location of the input/output and informs the Operating System through Job control Statements. In mainframe environment, programs can be executed in batch and online modes. JCL is used for submitting a program for execution in batch mode. JCL is core to development activities, and batch workload is core to all users.

JES 2

A z/OS subsystem that receives jobs into the system, converts them to internal format, selects them for execution, processes their output, and purges them from the system. In an installation with more than one processor, each JES2 processor independently controls its job input, scheduling, and output processing.

Microsoft Visual Studio Code (VS Code)

A code editor redefined and optimized for building and debugging modern web and cloud applications. Visual Studio Code is free and available on Linux®, macOS, and Windows.

profile

- A collection of controls used to evaluate whether technology and services are configured in compliance with security policies.
- A specification of a resource's capacities and capabilities. Different profiles are optimized for different workloads and use cases. A resource's pricing model might depend on its profile.

quota

The number of resources that can be consumed at an account or service instance level.

Red Hat OpenShift Dev Spaces (formerly Red Hat CodeReady Workspaces)

Red Hat OpenShift Dev Spaces provides a powerful and familiar in-browser IDE with support for Microsoft Visual Studio Code extensions. Developers need only a machine capable of running a web browser to code, build, test, and run on OpenShift.

Red Hat OpenShift

Red Hat OpenShift is a hybrid cloud, enterprise Kubernetes application platform.

route

A route allows developers to expose services through an HTTP(S) aware load balancing and proxy layer via a public DNS entry. An administrator typically configures their router to be visible outside the cluster firewall and can also add additional security, caching, or traffic controls on the service content.

SCM

SCM refers to Software Configuration Management or Source Code Management. It refers to the task of tracking and controlling changes in the software. SCM practices include revision control and the establishment of baselines. If something goes wrong, SCM can determine what was changed and who changed it.

Secret

The Secret object type provides a mechanism to hold sensitive information such as passwords, OpenShift Container Platform client configuration files, private source repository credentials, and so on. Secrets decouple sensitive content from the pods.

security group

A resource that provides rules to filter IP traffic to resources in a virtual private cloud. Rules are stateful, such that packets in response to allowed packets are automatically permitted.

server configuration file

A file (in XML format) used by the SHK Sentinel Key Server to obtain networking and logging parameters. It is at this location: /opt/safenet_sentinel/common_files/sentinel_keys_server/Sntlconfigsrvr.xml

Software-based License Server update file

The specific license key file that is generated in the Rational License Key Center and installed on a Software-based License Server by using an update_license command. This file is used for software-based licensing.

source environment

The environment where ZD&T users extract components. Several types of source environments are supported, including existing physical IBM Z platforms and Linux systems that run a ZD&T emulator .

stock image

An IBM-provided image template that can be used to create virtual server instances in a virtual private cloud.

system console

In z/OS, a console attached to the processor controller used to initialize a z/OS system.

target environment

An Intel x86-compatible environment that is capable to deploy z/OS images.

Time Sharing Option/Extensions (TSO/E)

Time Sharing Option/Extensions (TSO/E) allows users to create an interactive session with the z/OS system. TSO provides a single-user logon capability and a basic command prompt interface to z/OS.

tile

A visual representation of a running application that provides status on a dashboard.

token

A type of IBM Rational product entitlement that allows great flexibility in the deployment and use of associated, purchased products. The product documentation uses the phrase Rational Token to refer to IBM Rational Token licensing.

user build

A developer who is working on COBOL applications can run a user build to compile and link programs before the code is ready to be exposed to the repository for others to use. With user build, you can compile your program without having to perform commits or

pushes.

UUID

A universally unique identifier. It is obtained from the Intel machine BIOS. It is used to uniquely associate a UIM serial number with a particular machine.

virtual private cloud (VPC)

A virtual network that is tied to a private user account and isolated from other networks in a public cloud. Only authorized users can access virtual private cloud resources, which include virtual servers, storage, and subnets.

virtual private network (VPN)

A private connection between two endpoints, even when the data is transferred across a public network. Usually, a VPN is used in combination with security methods, such as authentication and encryption, to provide maximum data security and privacy.

virtual server

A server that shares its resources with other servers to support applications.

virtual telecommunications access method (VTAM)

A set of programs that maintain control of the communication between terminals and application programs running under z/OS.

write-to-operator-with-reply (WTOR) message

A message sent to an operator console informing the operator of errors and system conditions that may need correcting. The operator must enter a response.

ZD&T z/OS Extraction Utilities

A component of ZD&T Enterprise Edition to be installed on one or more IBM Z platforms to allow Wazi Image Builder to use such IBM Z platforms as source environments to extract z/OS resources.

Installation of ZD&T z/OS Extraction Utilities is required if you need to create components from either the physical or emulated IBM Z platforms.

zone

An independent fault domain. A zone is an abstraction designed to assist with improved fault tolerance and decreased latency.

Zowe

Zowe is an open-source project that is created to host technologies that benefit the IBM Z platform for all members of the Z community, including Integrated Software Vendors (ISVs), System Integrators, and z/OS consumers. Zowe, like Mac or Windows, comes with a set of APIs and OS capabilities that applications build on and also includes some applications out of the box. Zowe offers modern interfaces to interact with z/OS, with which you can work with z/OS in a way that is similar to what you experience on cloud platforms today. You can use these interfaces as delivered or through plug-ins and extensions that are created by clients or third-party vendors.

Zowe CLI

Zowe CLI enables you to interact with z/OS remotely via a command line interface.

Zowe Explorer

The Zowe Explorer VS Code extension (Zowe Explorer) provides you with graphical views to explore MVS™, z/OS UNIX, and JES remotely.

Resources

For more information, see the following resources:

- Glossary of z/OS terms and abbreviations
- Glossary of IBM Wazi
- Terminology for IBM Z Development and Test Environment
- Glossary terms for IBM Cloud