z/OS Connect Open API 3

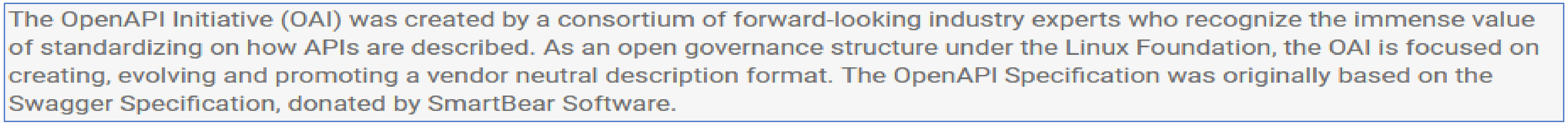
Designer and z/OS Native server
Experiences and Observations

Mitch Johnson
mitchj@us.ibm.com
Washington Systems Center

IBM Z
Wildfire Team –
Washington System Center

The significance of OpenAPI Specification

The industry standard framework for describing REST APIs


The OpenAPI Initiative (OAI) was created by a consortium of forward-looking industry experts who recognize the immense value of standardizing on how APIs are described. As an open governance structure under the Linux Foundation, the OAI is focused on creating, evolving and promoting a vendor neutral description format. The OpenAPI Specification was originally based on the Swagger Specification, donated by SmartBear Software.

- **z/OS Connect and Open API Specification 2 (Initially supported by z/OS Connect)**
  - Where the interactions with the z/OS resources were driven by the layout of the CICS COMMAREA or CONTAINER, the IMS or MQ messages or the Db2 REST service.
  - The z/OS resource interactions **determined** the contents of the API request and response messages and produced the specification document.
- **z/OS Connect and Open API Specification 3 (Supported by z/OS Connect as of the March 2022 service)**
  - As companies mature their API strategy, they begin to introduce API governance boards to drive consistency in their API design
  - As more public APIs are created, government and industry standards bodies begin to regulate and drive for standardization
  - This drives the need for “API first” functional mapping capabilities within the integration platform
  - The external API design **determines** the contents of the API request and response messages provided by the specification documents. This document is consumed by z/OS Connect to describe the z/OS resource interactions

Both Open API2 and Open API3 are supported by z/OS Connect concurrently*

Quick and easy

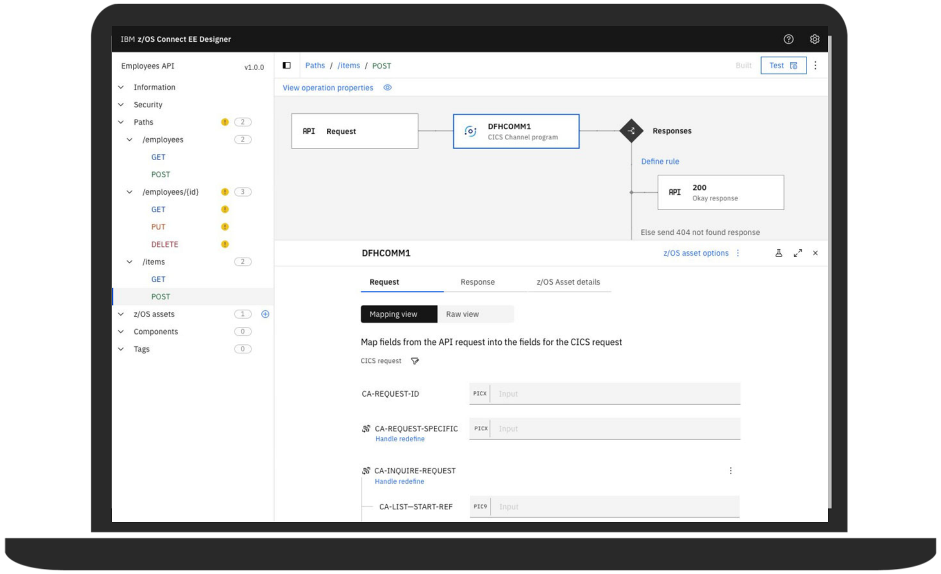
- A web-based user interface, provides a no code approach to create APIs in minutes
- Removes any dependency on Z platform development skills
- Rapid development of APIs using modern DevOps processes

```
GET http://www.acme.com/customers/12345
```

```
01 INQCUST
02 ID PIC 9 (5).
02 NAME PIC x (64).
02 ADDRESS PIC x (128)
02 TEL PIC 9 (11).
```

```
RESPONSE: HTTP 200 OK
BODY  {  “ID” : “12345”,
         “name” : “Joe Bloggs”,
         “address” : “10 OldStreet”,
         “tel” : “01234123456”  }
```

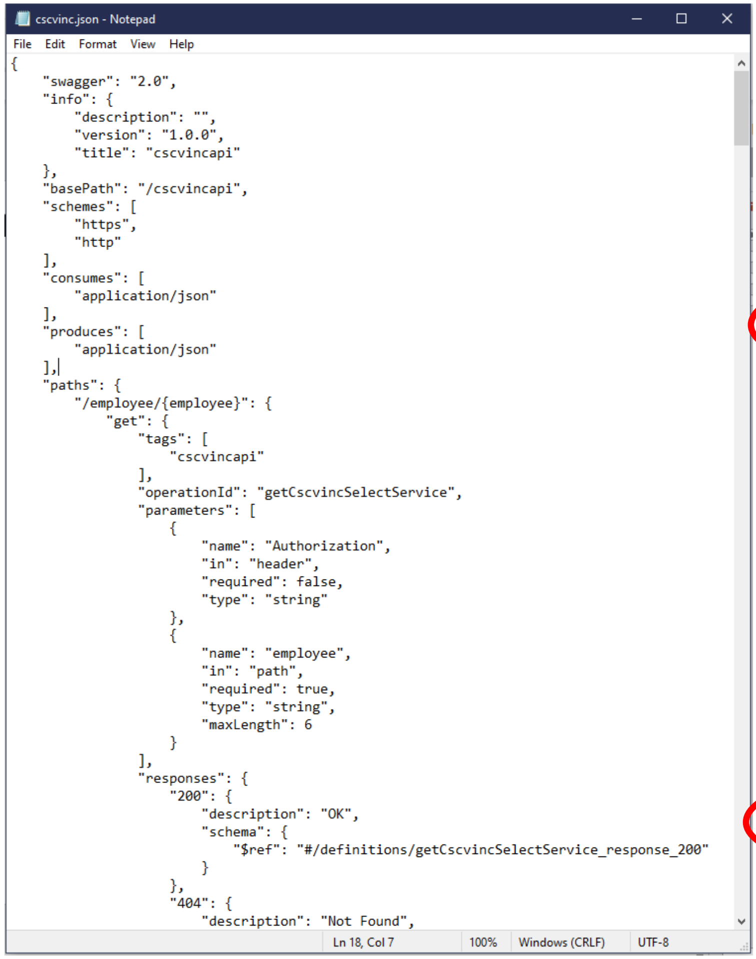
```
01 INQCUST
02 ID PIC 9 (5).
02 NAME PIC x (64).
02 ADDRESS PIC x (128)
02 TEL PIC 9 (11).
```



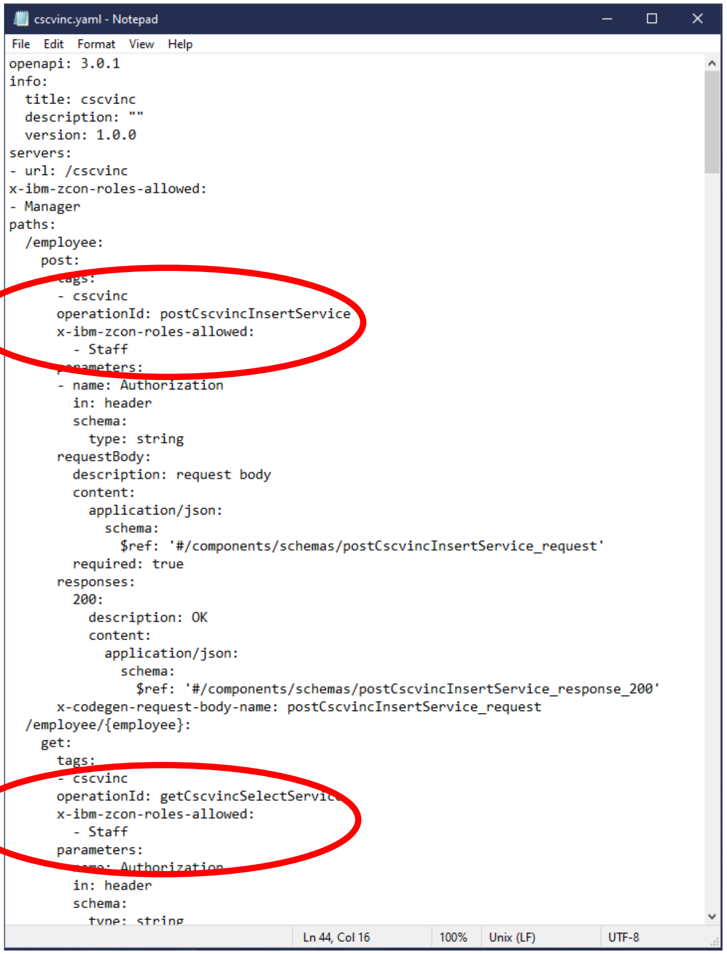
Map fields from the API request into the fields for the CICS request

Contrast the OpenAPI 2 /OpenAPI 3 specification

z/OS Connect **produces** an OpenAPI 2 specification document, where the details of the request/response messages are driven by the details of the z/OS resource (JSON Format)



```
{
    "swagger": "2.0",
    "info": {
        "description": "",
        "version": "1.0.0",
        "title": "cscvincapi"
    },
    "basePath": "/cscvincapi",
    "schemes": [
        "https",
        "http"
    ],
    "consumes": [
        "application/json"
    ],
    "produces": [
        "application/json"
    ],
    "paths": {
        "/employee/{employee}": {
            "get": {
                "tags": [
                    "cscvincapi"
                ],
                "operationId": "getCscvincSelectService",
                "parameters": [
                    {
                        "name": "Authorization",
                        "in": "header",
                        "required": false,
                        "type": "string"
                    },
                    {
                        "name": "employee",
                        "in": "path",
                        "required": true,
                        "type": "string",
                        "maxLength": 6
                    }
                ],
                "responses": {
                    "200": {
                        "description": "OK",
                        "schema": {
                            "$ref": "#/definitions/getCscvincSelectService_response_200"
                        }
                    },
                    "404": {
                        "description": "Not Found",
```



```
openapi: 3.0.1
info:
  title: cscvinc
  description: ""
  version: 1.0.0
servers:
- url: /cscvinc
x-ibm-zcon-roles-allowed:
- Manager
paths:
  /employee:
    post:
      tags:
      - cscvinc
      operationId: postCscvincInsertService
      x-ibm-zcon-roles-allowed:
        - Staff
      parameters:
      - name: Authorization
        in: header
        schema:
          type: string
      requestBody:
        description: request body
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/postCscvincInsertService_request'
        required: true
      responses:
        200:
          description: OK
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/postCscvincInsertService_response_200'
      x-codegen-request-body-name: postCscvincInsertService_request
  /employee/{employee}:
    get:
      tags:
      - cscvinc
      operationId: getCscvincSelectServic
      x-ibm-zcon-roles-allowed:
        - Staff
      parameters:
      - name: Authorization
        in: header
        schema:
          type: string
```

z/OS Connect **consumes** an OpenAPI specification document and is driven by the design of the specification of the API (YAML Format*)

mitchj@us.ibm.com

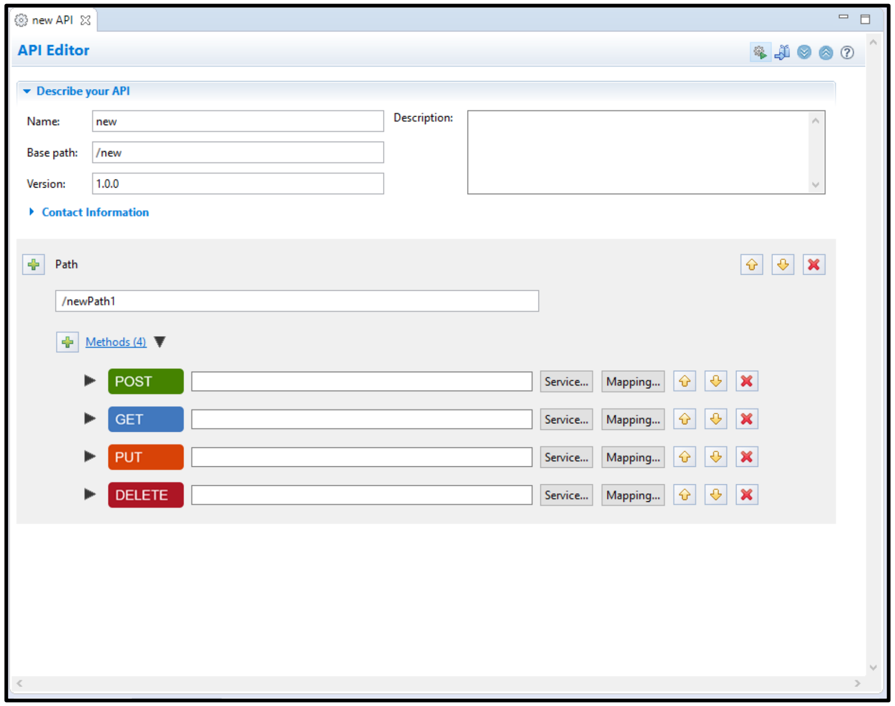
https://swagger.io/blog/api-strategy/difference-between-swagger-and-openapi/

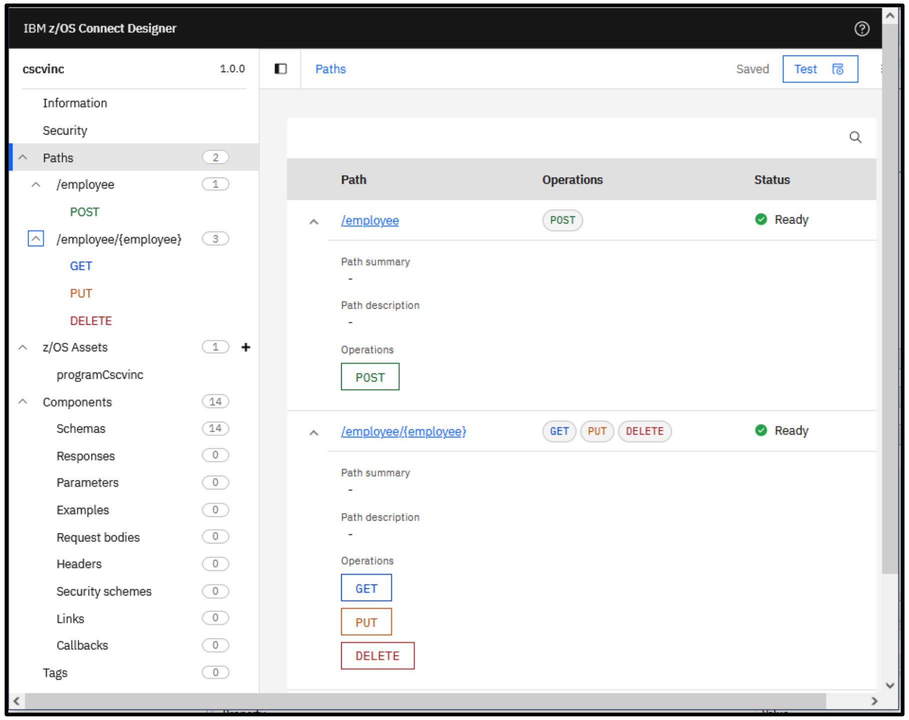
*Yet Another Markup Language

zCEE - OpenAPI 2 Palette versus the OpenAPI 3 API Designer

z/OS Connect API Toolkit (Eclipse)

z/OS Connect Designer (Designer Container)

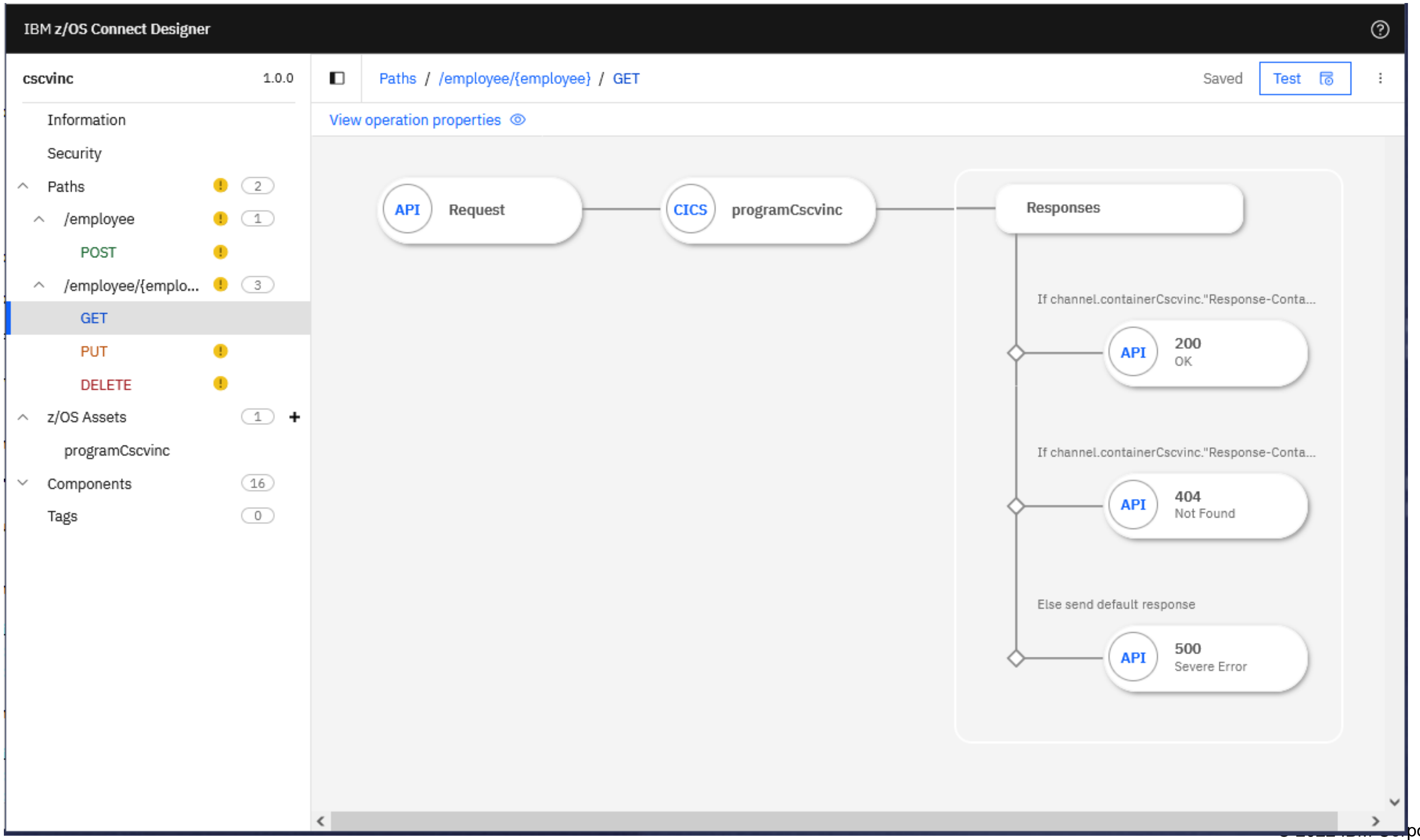




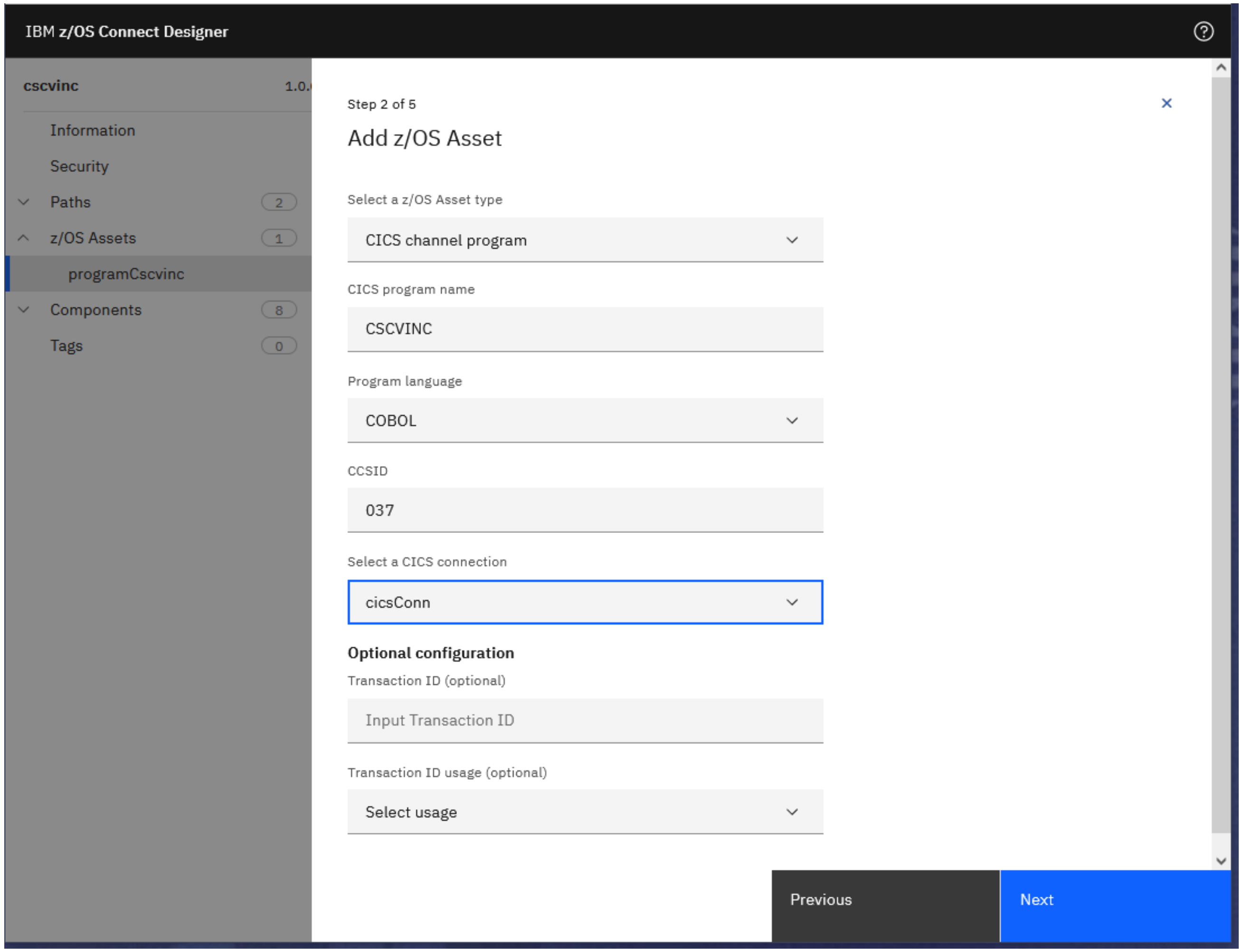
The API toolkit is used to define the URI paths and methods.

The API specification provides predefined URI Paths and methods.

Begin by importing the YAML description of an API into the Designer



Add the z/OS asset, e.g., a CICS program



Step 2 of 5

Add z/OS Asset

Select a z/OS Asset type: CICS channel program

CICS program name: CSCVINC

Program language: COBOL

CCSID: 037

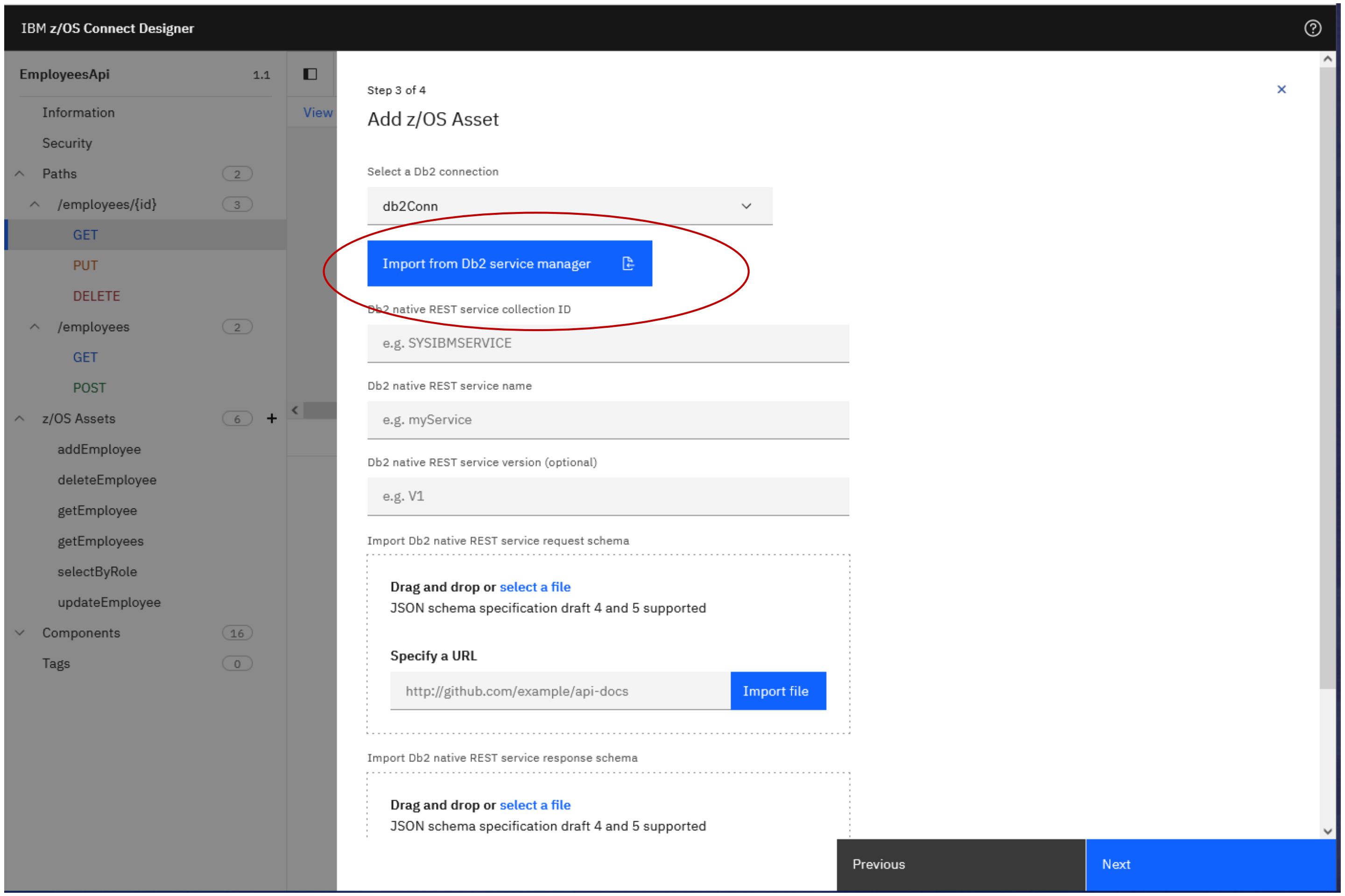
Select a CICS connection: cicsConn

Optional configuration

Transaction ID (optional): Input Transaction ID

Transaction ID usage (optional): Select usage

Or a Db2 REST service



Step 3 of 4

Add z/OS Asset

Select a Db2 connection: db2Conn

Db2 native REST service collection ID: e.g. SYSIBMSERVICE

Db2 native REST service name: e.g. myService

Db2 native REST service version (optional): e.g. V1

Import Db2 native REST service request schema

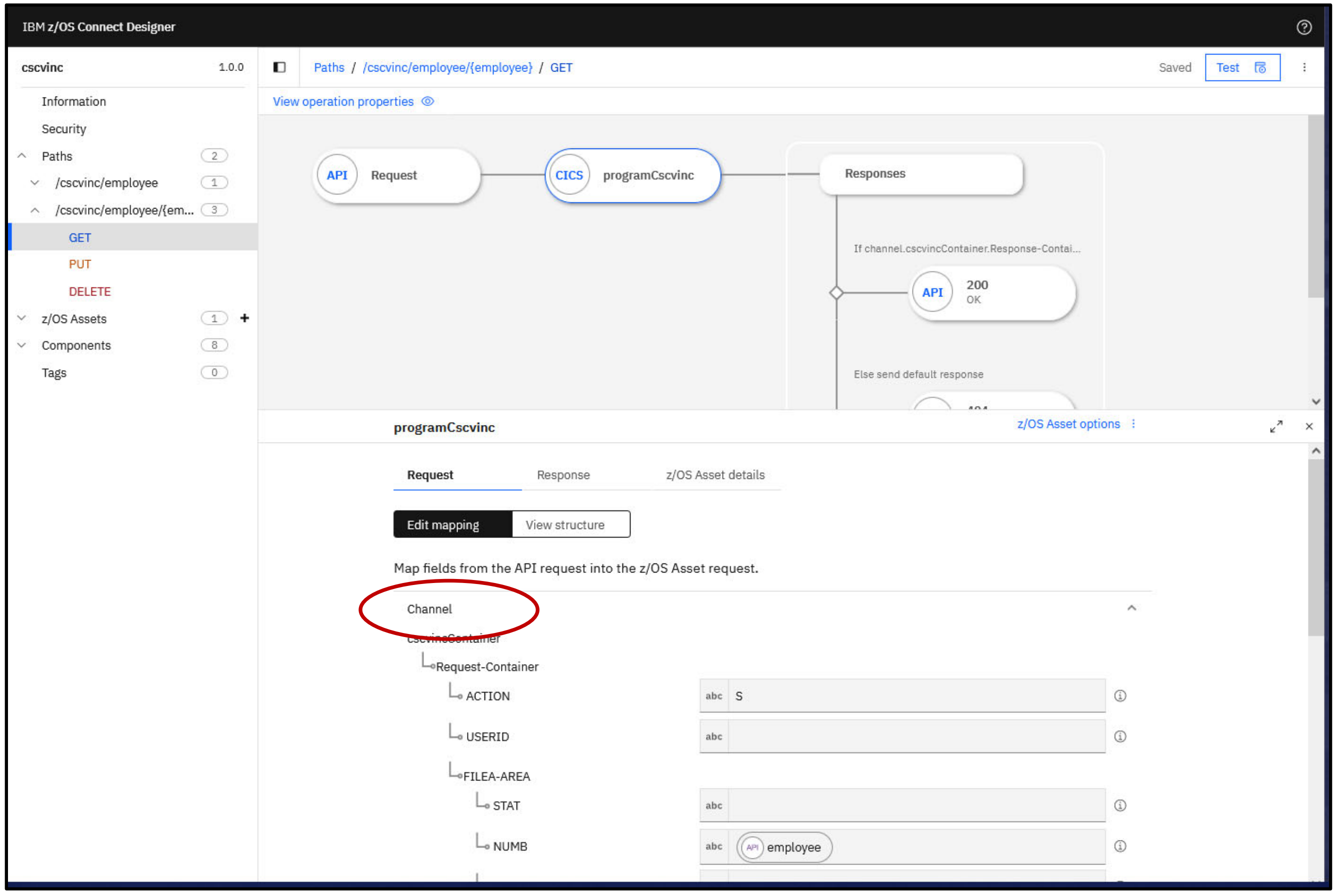
Drag and drop or select a file
JSON schema specification draft 4 and 5 supported

Specify a URL: http://github.com/example/api-docs

Import Db2 native REST service response schema

Drag and drop or select a file
JSON schema specification draft 4 and 5 supported

Map the API’s methods and request messages to the z/OS “request”



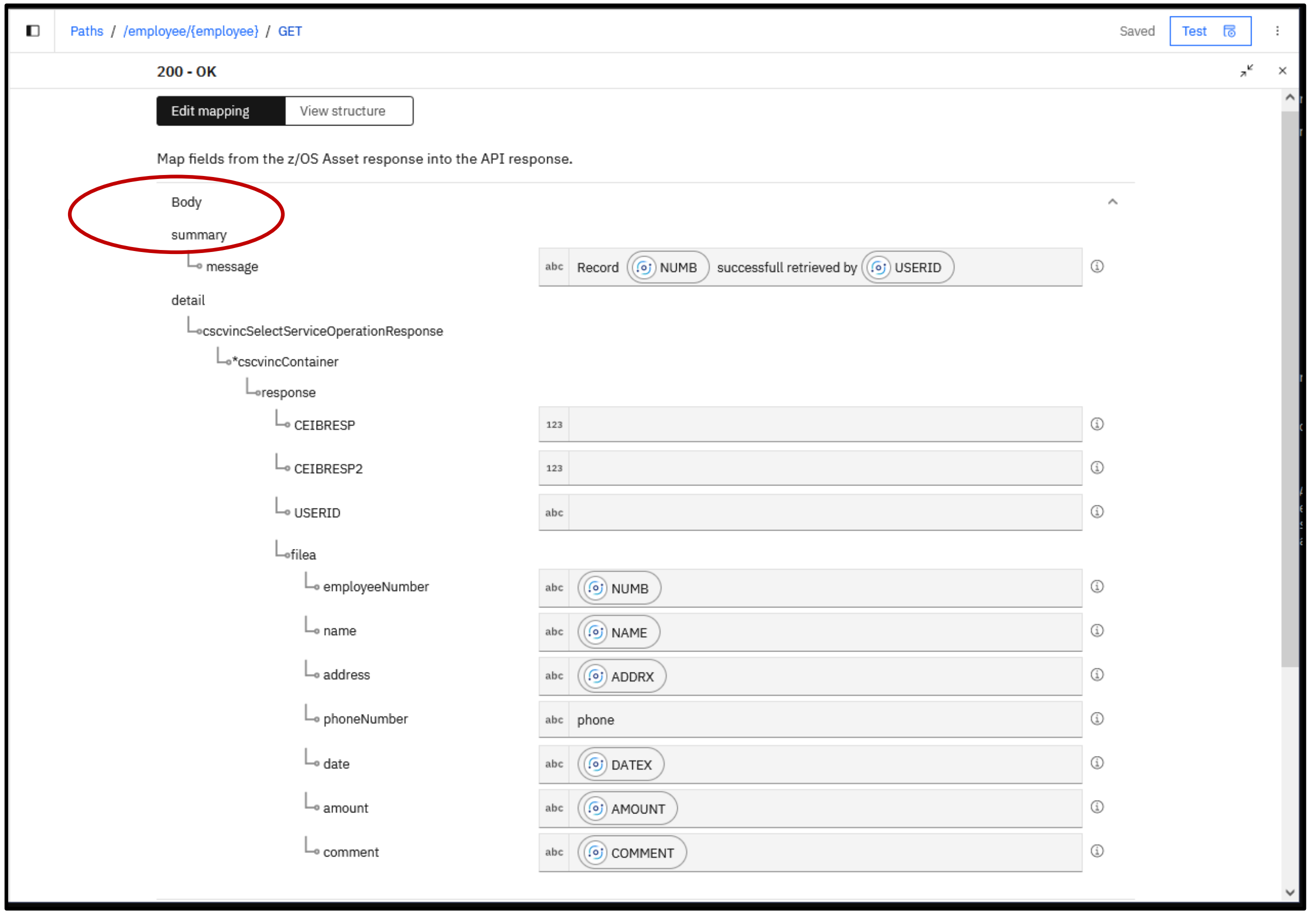
Map fields from the API request into the z/OS Asset request.

Channel
cscvincContainer
Request-Container
ACTION abc S
USERID abc
FILEA-AREA
STAT abc
NUMB abc employee

Map **(right to left)** the values provided by the API request message properties to the fields of the request “message” sent to the z/OS resources. And augment the z/OS request “message” as needed by the z/OS resource.

To: z/OS “request” ← From: the API request message body

Map the z/OS “responses” to the API’s response messages



Paths / /employee/{employee} / GET

Saved Test

200 - OK

Edit mapping | View structure

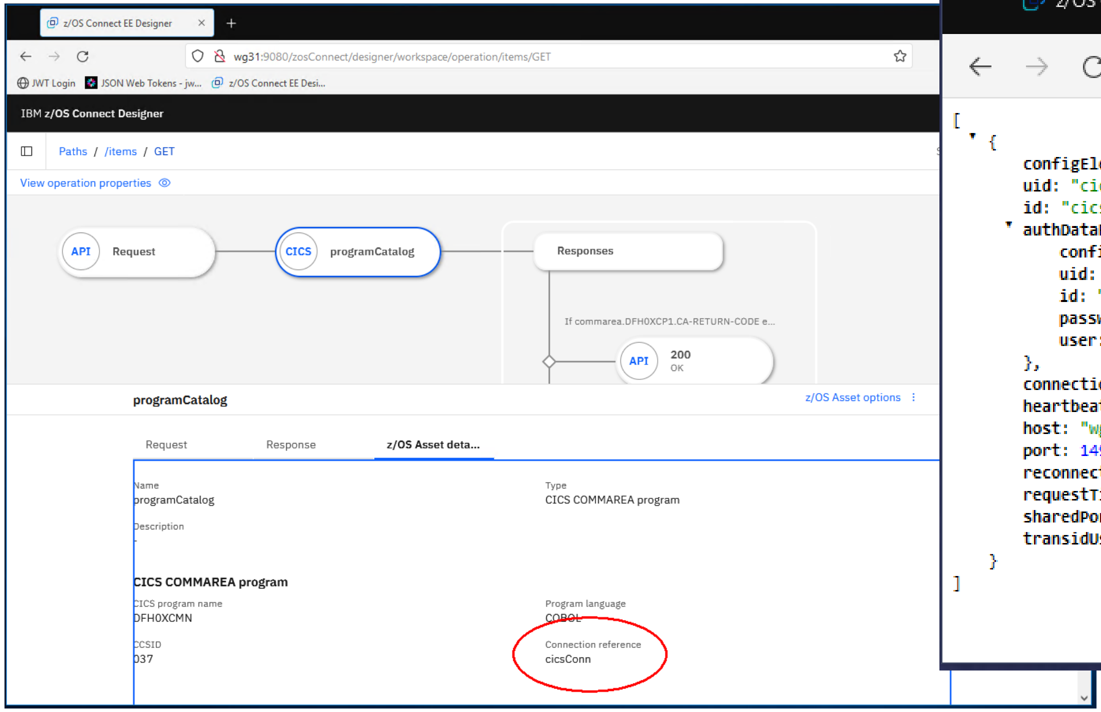
Map fields from the z/OS Asset response into the API response.

Body	
summary	
message	abc Record NUMB successfull retrieved by USERID
detail	
cscvincSelectServiceOperationResponse	
*cscvincContainer	
response	
CEIBRESP	123
CEIBRESP2	123
USERID	abc
filea	
employeeNumber	abc NUMB
name	abc NAME
address	abc ADDRX
phoneNumber	abc phone
date	abc DATEX
amount	abc AMOUNT
comment	abc COMMENT

Map **(right to left)** the values returned by the z/OS resource to the corresponding API response properties and augment other API response message fields

To: the API “response” message body ← From: the z/OS response

Accessing a CICS program using IPIC



programCatalog

Name
programCatalog

Type
CICS COMMAREA program

Description
-

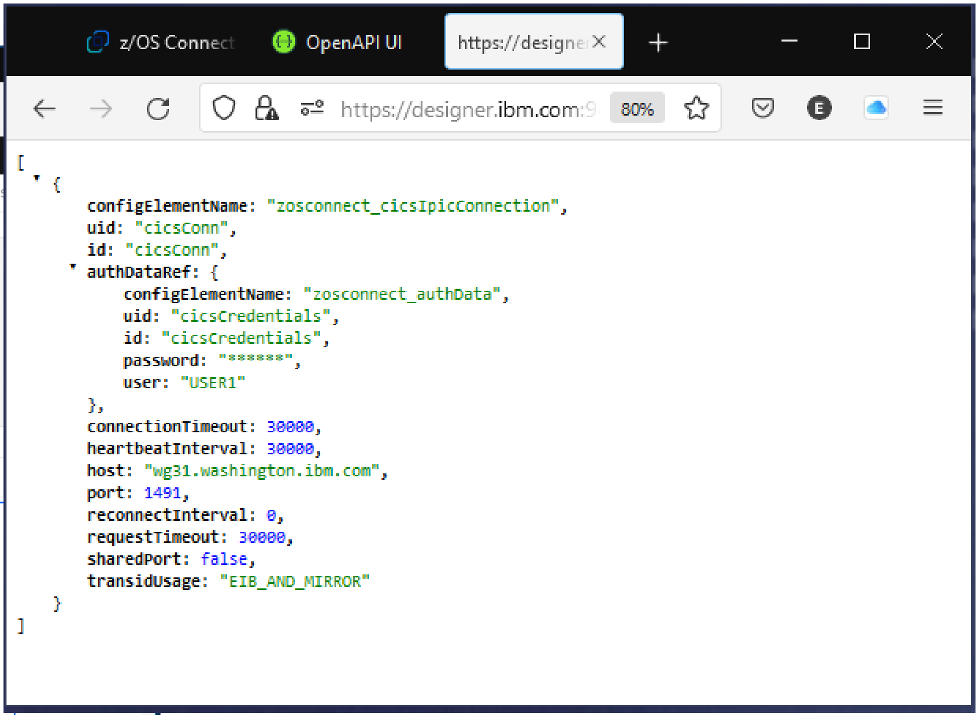
CICS COMMAREA program

CICS program name
DFHOXCMN

Program language
COBOL

CCSID
037

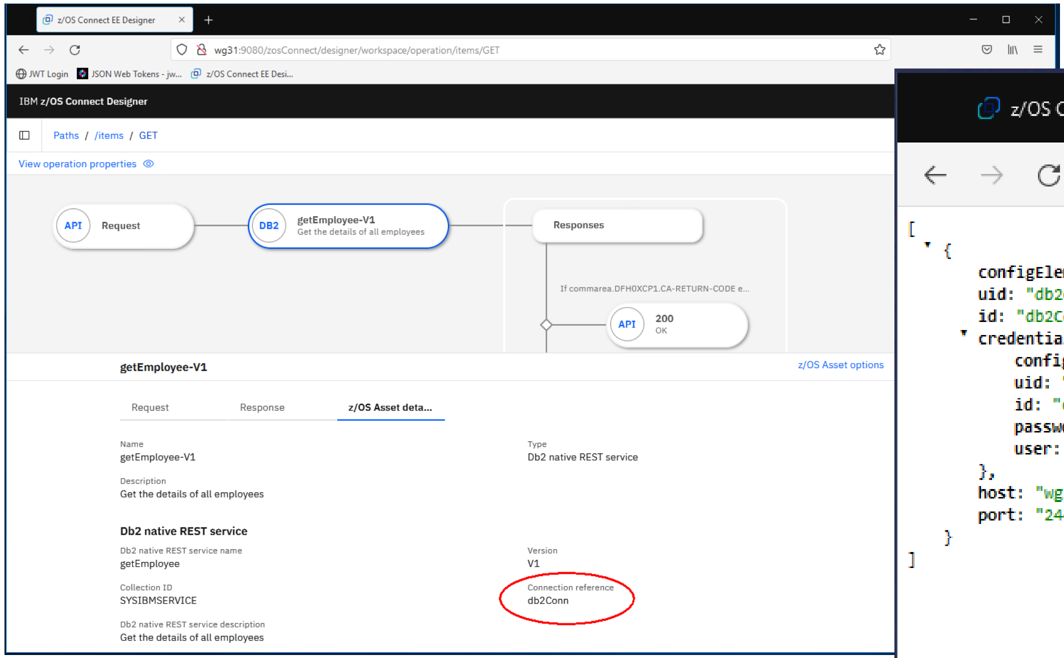
Connection reference
cicsConn



```
[
  {
      configElementName: "zosconnect_cicsIpicConnection",
      uid: "cicsConn",
      id: "cicsConn",
      authDataRef: {
          configElementName: "zosconnect_authData",
          uid: "cicsCredentials",
          id: "cicsCredentials",
          password: "******",
          user: "USER1"
      },
      connectionTimeout: 30000,
      heartbeatInterval: 30000,
      host: "wg31.washington.ibm.com",
      port: 1491,
      reconnectInterval: 0,
      requestTimeout: 30000,
      sharedPort: false,
      transidUsage: "EIB_AND_MIRROR"
  }
]
```

The connection references identifies a zosconnect_cicsIpicConnection configuration element. Which provides the connection details to a CICS region.

Accessing a Db2 REST Service Manager



z/OS Connect EE Designer
wg31:9080/zosConnect/designer/workspace/operation/items/GET
IBM z/OS Connect Designer
Paths / /items / GET
View operation properties
API Request
DB2 getEmployee-V1 Get the details of all employees
Responses
If commarea.DFH0XCP1.CA-RETURN-CODE e...
API 200 OK
z/OS Asset options
Request
Response
z/OS Asset deta...

getEmployee-V1

Name
getEmployee-V1

Type
Db2 native REST service

Description
Get the details of all employees

Db2 native REST service

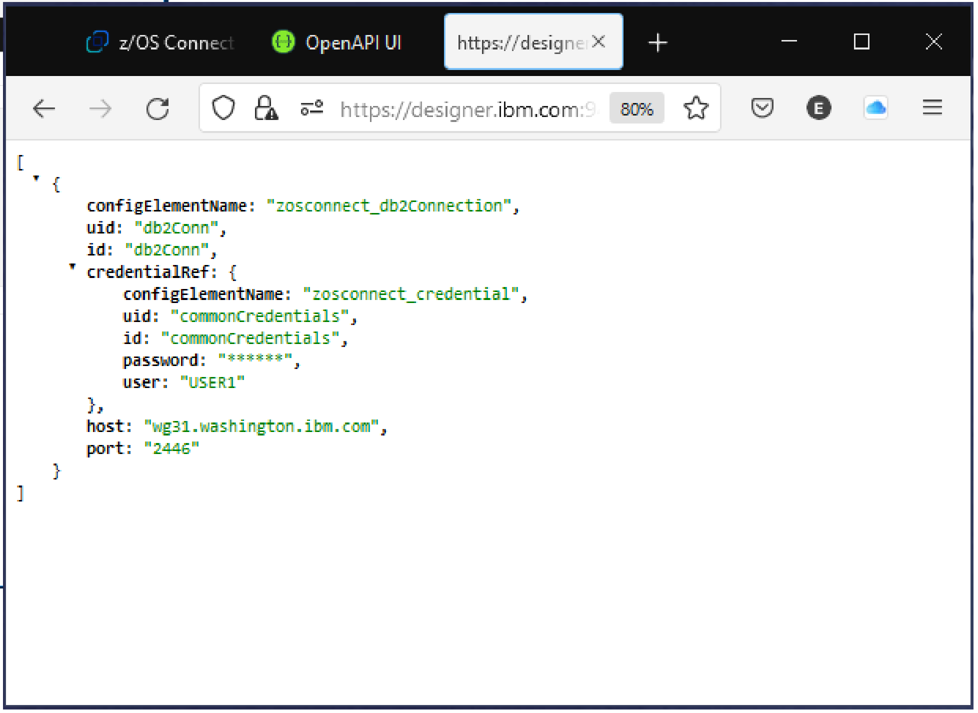
Db2 native REST service name
getEmployee

Version
V1

Collection ID
SYSIBMSERVICE

Connection reference
db2Conn

Db2 native REST service description
Get the details of all employees

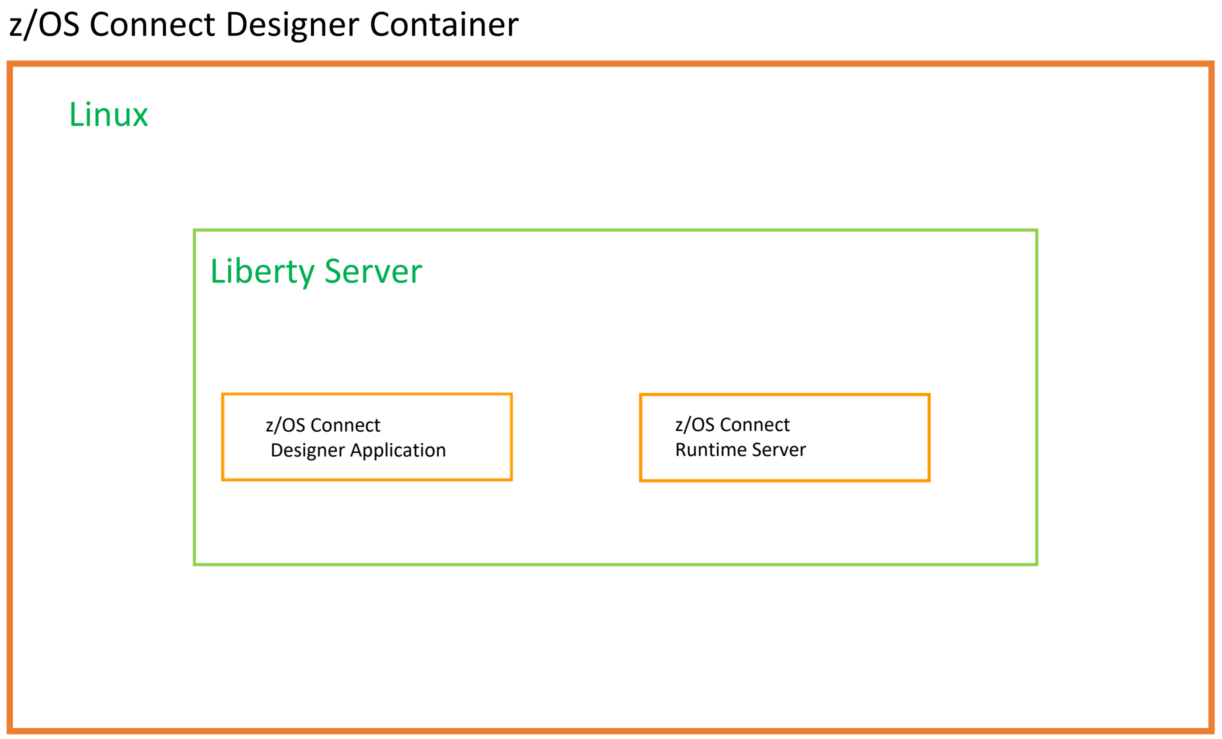


```
[
  {
    configElementName: "zosconnect_db2Connection",
    uid: "db2Conn",
    id: "db2Conn",
    credentialRef: {
        configElementName: "zosconnect_credential",
        uid: "commonCredentials",
        id: "commonCredentials",
        password: "******",
        user: "USER1"
    },
    host: "wg31.washington.ibm.com",
    port: "2446"
  }
]
```

The connection references identifies a zosconnect_db2Connection configuration element. Which provides the connection details to a DB2 DDF task.

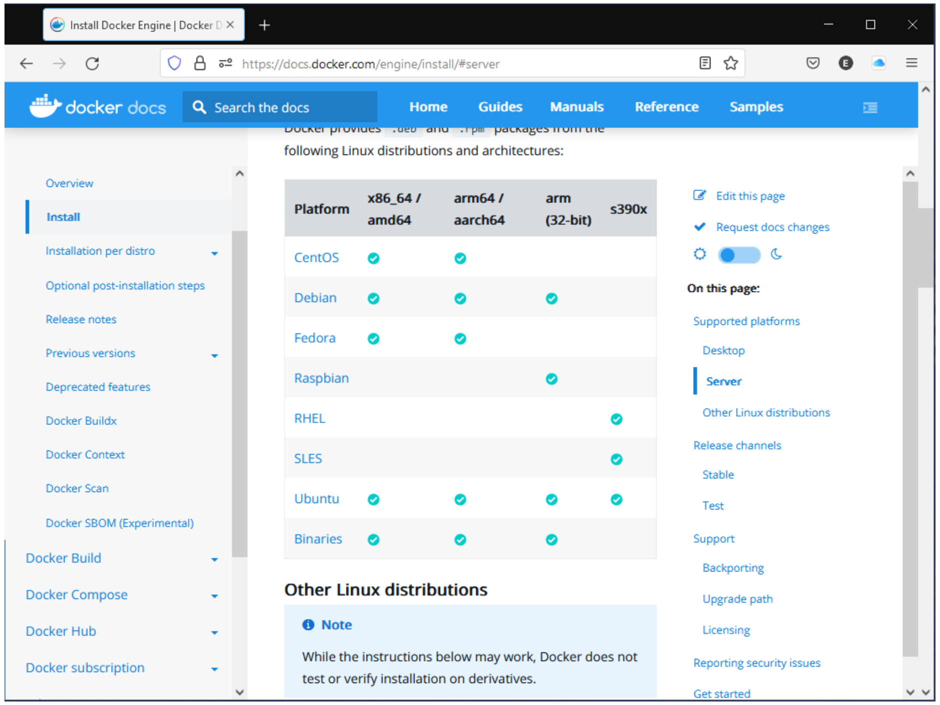
mitchj@us.ibm.com

The basic z/OS Connect Designer Container


z/OS Connect Designer Container
Linux
Liberty Server
z/OS Connect Designer Application
z/OS Connect Runtime Server

A z/OS Connect Designer container is composed of a Linux environment with a Liberty server running a z/OS Connect Designer application and a z/OS Connect runtime server

Docker Desktop Alternatives (e.g., no license required)



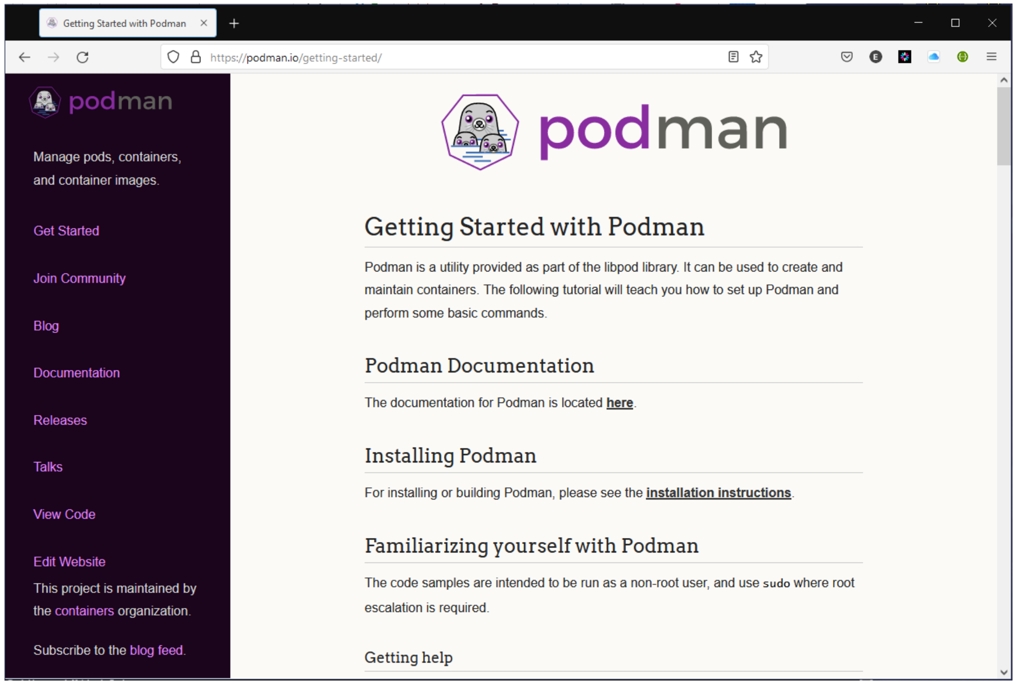
Docker provides .deb and .rpm packages from the following Linux distributions and architectures:

Platform	x86_64 / amd64	arm64 / aarch64	arm (32-bit)	s390x
CentOS	✓	✓		
Debian	✓	✓	✓	
Fedora	✓	✓		
Raspbian			✓	
RHEL				✓
SLES				✓
Ubuntu	✓	✓	✓	✓
Binaries	✓	✓	✓	

Other Linux distributions

Note

While the instructions below may work, Docker does not test or verify installation on derivatives.



Manage pods, containers, and container images.

This project is maintained by the containers organization.

Subscribe to the blog feed.

Getting Started with Podman

Podman is a utility provided as part of the libpod library. It can be used to create and maintain containers. The following tutorial will teach you how to set up Podman and perform some basic commands.

Podman Documentation

The documentation for Podman is located here.

Installing Podman

For installing or building Podman, please see the installation instructions.

Familiarizing yourself with Podman

The code samples are intended to be run as a non-root user, and use sudo where root escalation is required.

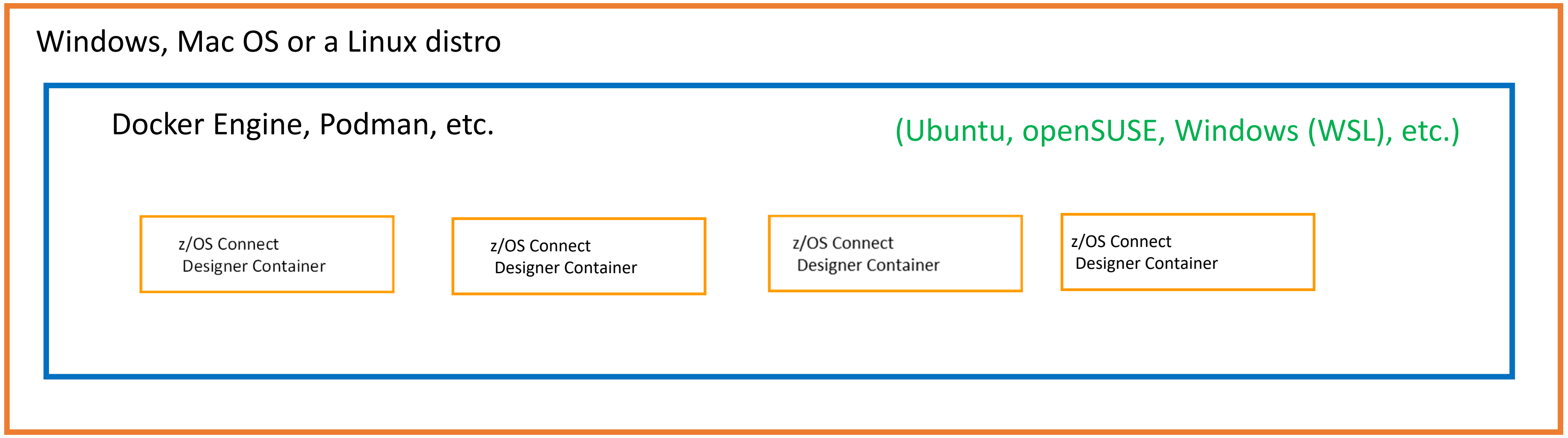
Getting help

https://docs.docker.com/engine/install/#server

https://podman.io/getting-started/

Important: The command line interface (CLI) syntax is the same between the Docker and Podman. Just change a Docker command from using the ***docker*** command to the ***podman*** command, e.g., ***docker ps -a*** when using Docker becomes ***podman ps –a*** when using Podman.

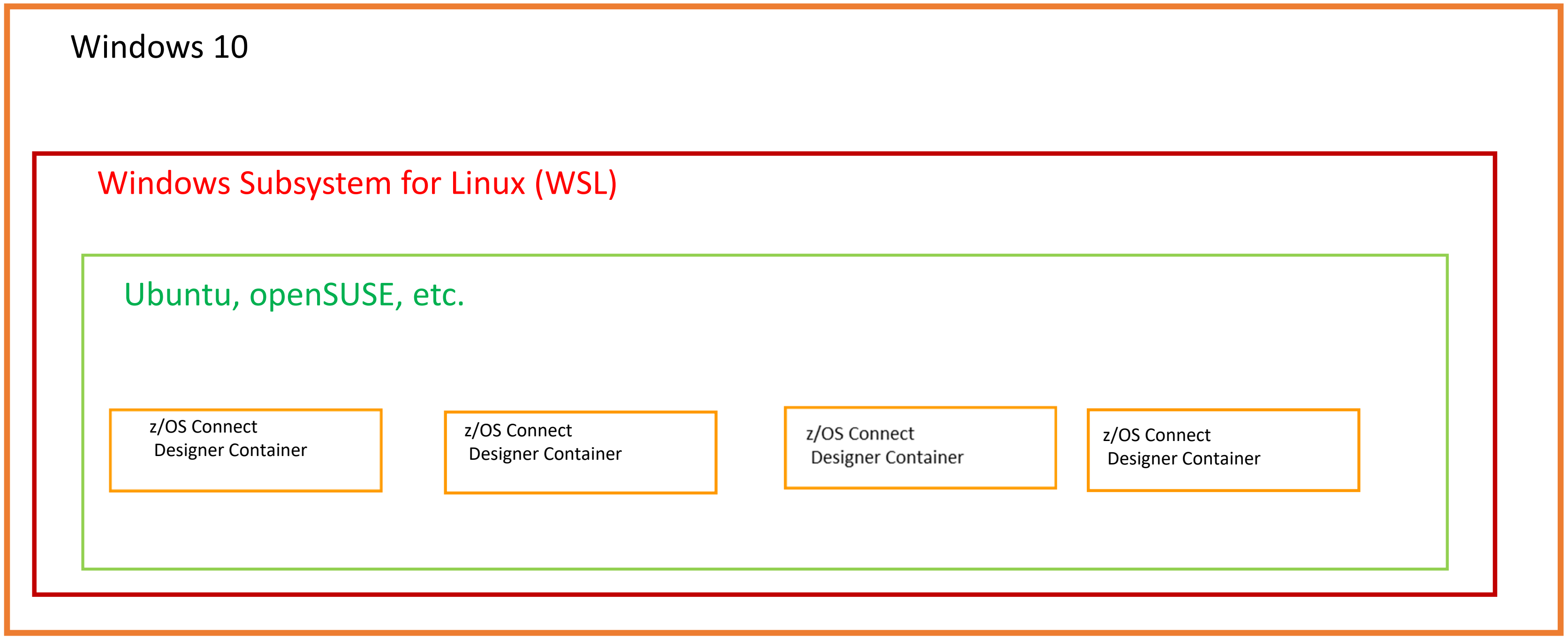
z/OS Connect Designer Container Topology


Windows, Mac OS or a Linux distro
Docker Engine, Podman, etc.
(Ubuntu, openSUSE, Windows (WSL), etc.)
z/OS Connect Designer Container
z/OS Connect Designer Container
z/OS Connect Designer Container
z/OS Connect Designer Container

Warning, Machines with Mac M1 Pro processors may be problematic.

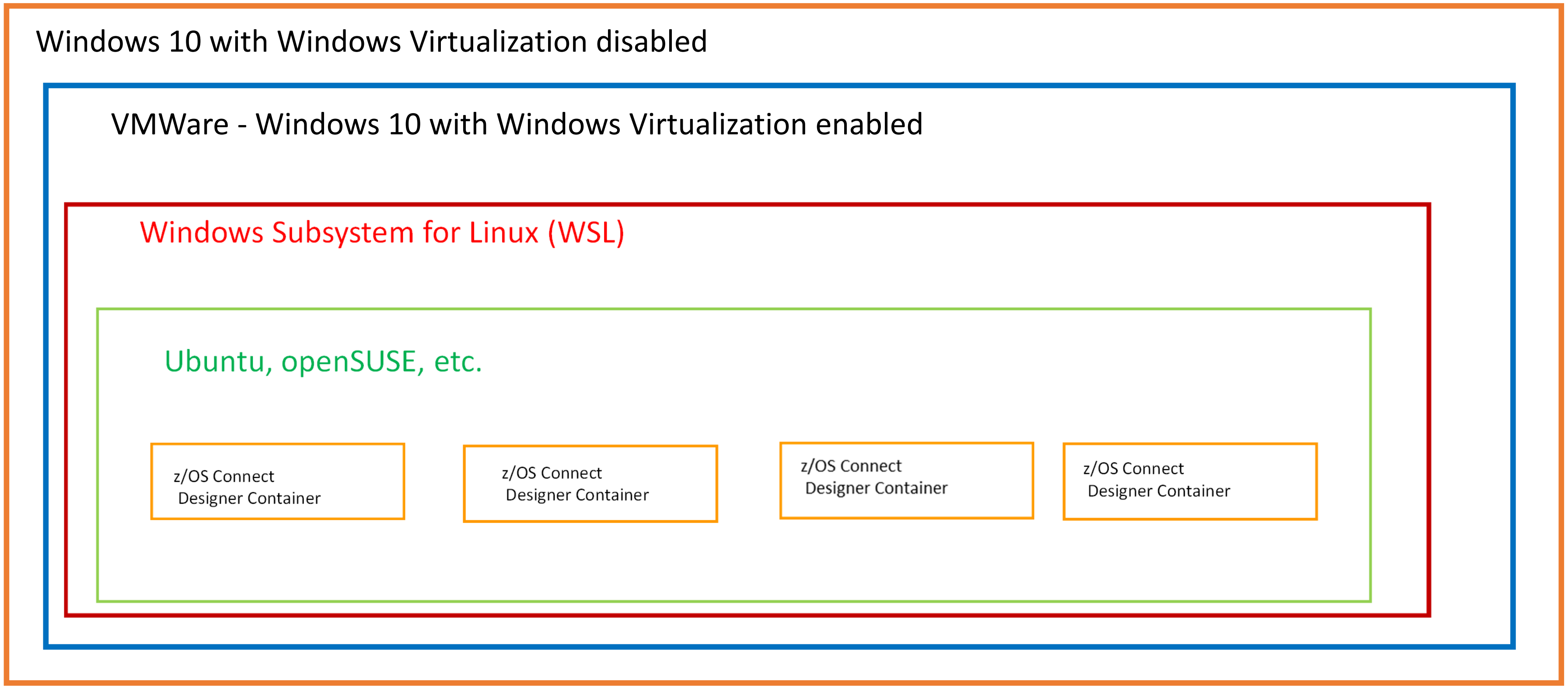
z/OS Connect Designer Container Topology – Local Windows

One container for each API

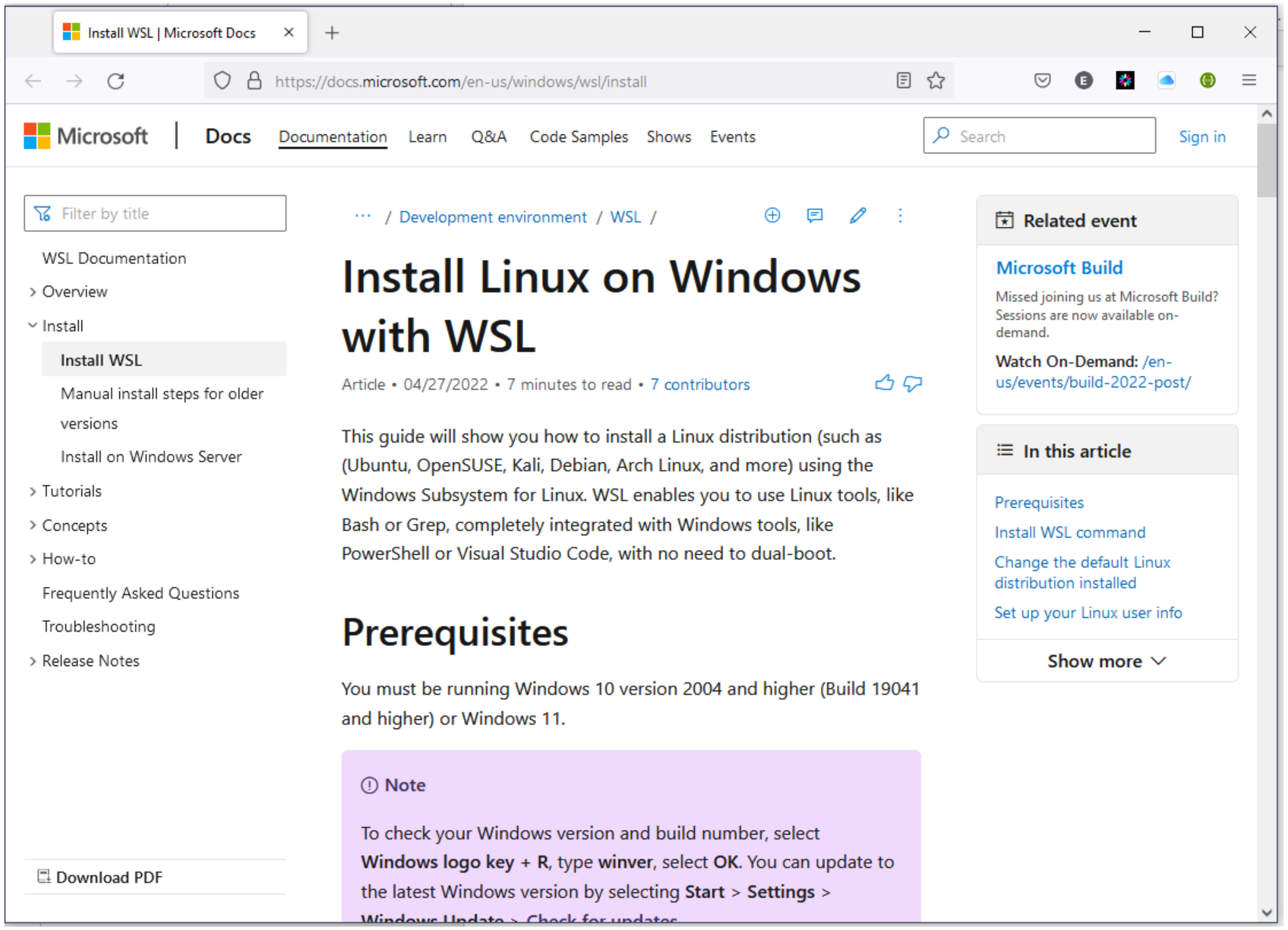

Windows 10
Windows Subsystem for Linux (WSL)
Ubuntu, openSUSE, etc.
z/OS Connect Designer Container
z/OS Connect Designer Container
z/OS Connect Designer Container
z/OS Connect Designer Container

A z/OS Connect Designer is an application running in Liberty running in Linux (the container) which is running Linux (the container runtime) running in WSL running in Windows.

z/OS Connect Designer Container Topology – VMWare Windows


Windows 10 with Windows Virtualization disabled
VMWare - Windows 10 with Windows Virtualization enabled
Windows Subsystem for Linux (WSL)
Ubuntu, openSUSE, etc.
z/OS Connect Designer Container
z/OS Connect Designer Container
z/OS Connect Designer Container
z/OS Connect Designer Container

https://docs.microsoft.com/en-us/windows/wsl/install


Install WSL | Microsoft Docs
https://docs.microsoft.com/en-us/windows/wsl/install
Microsoft
Docs
Documentation
Learn
Q&A
Code Samples
Shows
Events
Search
Sign in
Filter by title
WSL Documentation
Overview
Install
Install WSL
Manual install steps for older versions
Install on Windows Server
Tutorials
Concepts
How-to
Frequently Asked Questions
Troubleshooting
Release Notes
Download PDF
... / Development environment / WSL /

Install Linux on Windows with WSL

Article • 04/27/2022 • 7 minutes to read • 7 contributors

This guide will show you how to install a Linux distribution (such as (Ubuntu, OpenSUSE, Kali, Debian, Arch Linux, and more) using the Windows Subsystem for Linux. WSL enables you to use Linux tools, like Bash or Grep, completely integrated with Windows tools, like PowerShell or Visual Studio Code, with no need to dual-boot.

Prerequisites

You must be running Windows 10 version 2004 and higher (Build 19041 and higher) or Windows 11.

Note

To check your Windows version and build number, select **Windows logo key** + **R**, type **winver**, select **OK**. You can update to the latest Windows version by selecting **Start** > **Settings** >

Related event

Microsoft Build

Missed joining us at Microsoft Build? Sessions are now available on-demand.

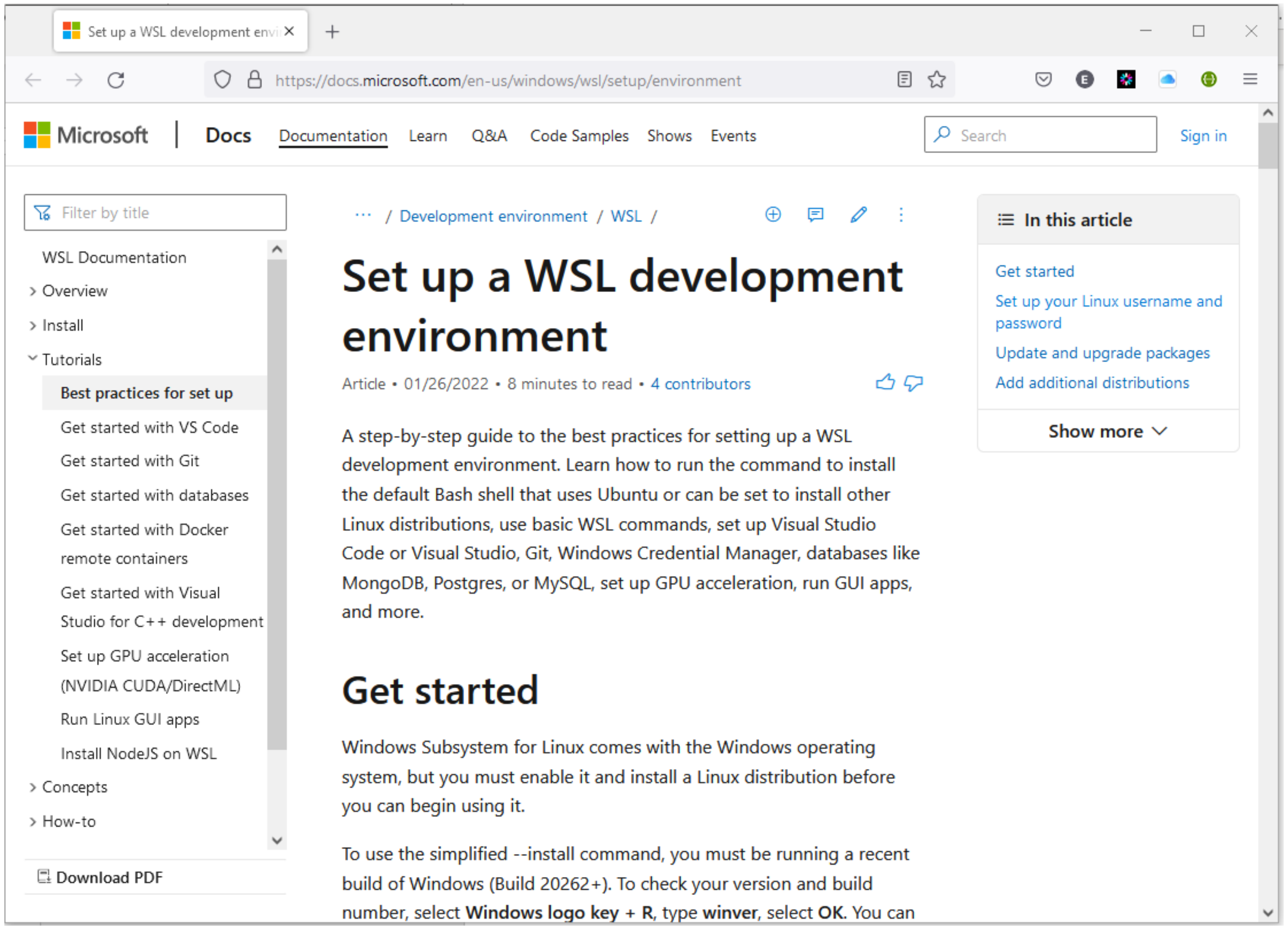
Watch On-Demand: /en-us/events/build-2022-post/

In this article

Prerequisites
Install WSL command
Change the default Linux distribution installed
Set up your Linux user info

Show more

https://docs.microsoft.com/en-us/windows/wsl/setup/environment


Set up a WSL development envi
https://docs.microsoft.com/en-us/windows/wsl/setup/environment
Microsoft | Docs
Documentation
Learn
Q&A
Code Samples
Shows
Events
Search
Sign in
Filter by title
WSL Documentation
Overview
Install
Tutorials
Best practices for set up
Get started with VS Code
Get started with Git
Get started with databases
Get started with Docker remote containers
Get started with Visual Studio for C++ development
Set up GPU acceleration (NVIDIA CUDA/DirectML)
Run Linux GUI apps
Install NodeJS on WSL
Concepts
How-to
Download PDF
… / Development environment / WSL /
In this article
Get started
Set up your Linux username and password
Update and upgrade packages
Add additional distributions
Show more

Set up a WSL development environment

Article • 01/26/2022 • 8 minutes to read • 4 contributors

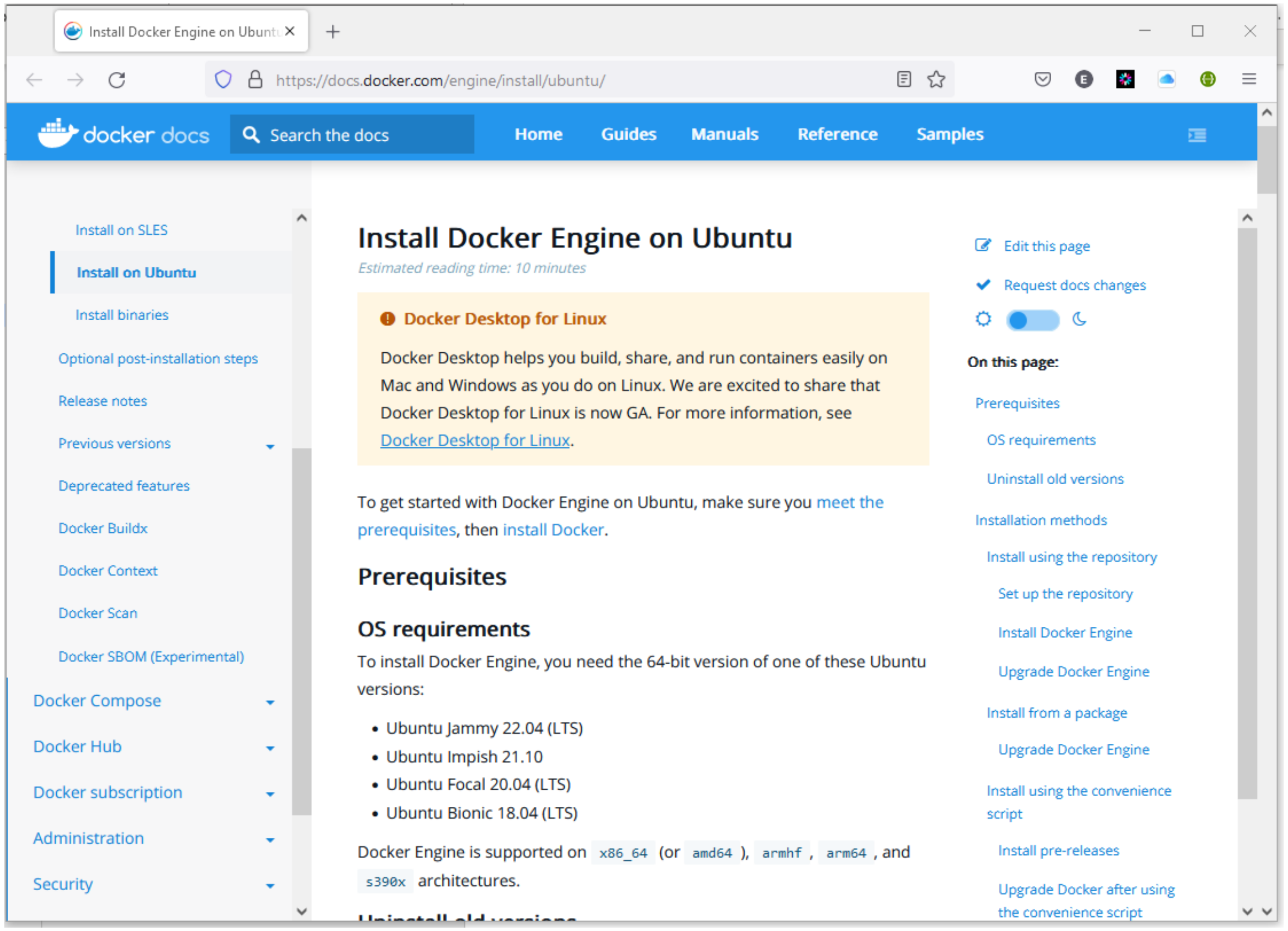
A step-by-step guide to the best practices for setting up a WSL development environment. Learn how to run the command to install the default Bash shell that uses Ubuntu or can be set to install other Linux distributions, use basic WSL commands, set up Visual Studio Code or Visual Studio, Git, Windows Credential Manager, databases like MongoDB, Postgres, or MySQL, set up GPU acceleration, run GUI apps, and more.

Get started

Windows Subsystem for Linux comes with the Windows operating system, but you must enable it and install a Linux distribution before you can begin using it.

To use the simplified --install command, you must be running a recent build of Windows (Build 20262+). To check your version and build number, select **Windows logo key + R**, type **winver**, select **OK**. You can

https://docs.docker.com/engine/install/ubuntu/



Install Docker Engine on Ubuntu
https://docs.docker.com/engine/install/ubuntu/

docker docs | Search the docs | Home | Guides | Manuals | Reference | Samples

- Install on SLES
- **Install on Ubuntu**
- Install binaries
- Optional post-installation steps
- Release notes
- Previous versions
- Deprecated features
- Docker Buildx
- Docker Context
- Docker Scan
- Docker SBOM (Experimental)
- Docker Compose
- Docker Hub
- Docker subscription
- Administration
- Security

Install Docker Engine on Ubuntu

Estimated reading time: 10 minutes

Docker Desktop for Linux

Docker Desktop helps you build, share, and run containers easily on Mac and Windows as you do on Linux. We are excited to share that Docker Desktop for Linux is now GA. For more information, see Docker Desktop for Linux.

To get started with Docker Engine on Ubuntu, make sure you meet the prerequisites, then install Docker.

Prerequisites

OS requirements

To install Docker Engine, you need the 64-bit version of one of these Ubuntu versions:

- Ubuntu Jammy 22.04 (LTS)
- Ubuntu Impish 21.10
- Ubuntu Focal 20.04 (LTS)
- Ubuntu Bionic 18.04 (LTS)

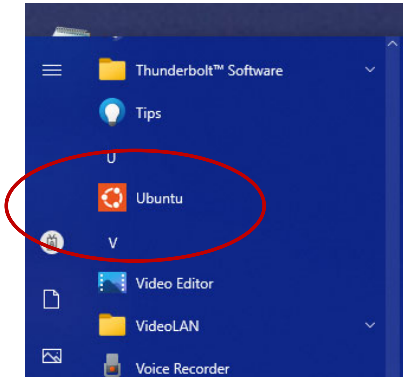
Docker Engine is supported on `x86_64` (or `amd64`), `armhf`, `arm64`, and `s390x` architectures.

Edit this page
Request docs changes

On this page:

- Prerequisites
  - OS requirements
  - Uninstall old versions
- Installation methods
  - Install using the repository
    - Set up the repository
    - Install Docker Engine
    - Upgrade Docker Engine
  - Install from a package
    - Upgrade Docker Engine
  - Install using the convenience script
    - Install pre-releases
    - Upgrade Docker after using the convenience script

Adds a Linux terminal icon to the Start menu



Windows PowerShell Commands

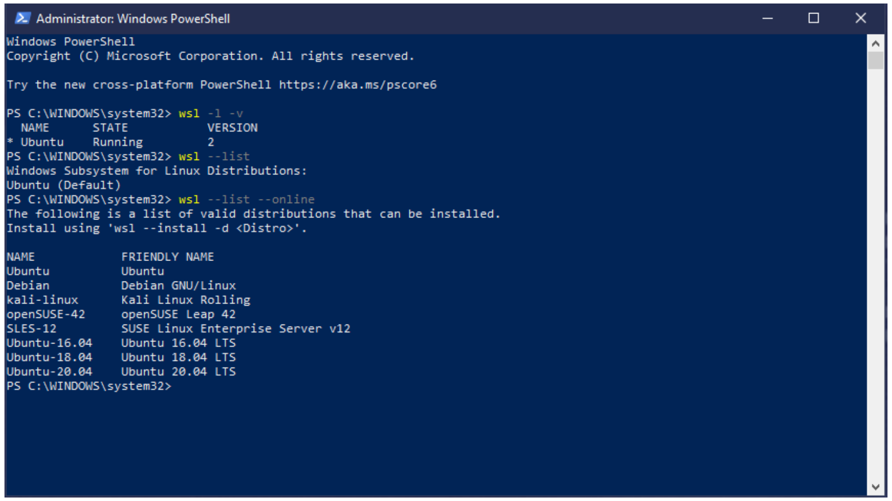
- **WSL Commands**

  List details of all distributions
  - *wsl -l –v*

  List all distributions
  - *wsl –list*

  Set the default install version for a new distribution
  - *wsl --set-default-version 2*

  Display a list of available Linux distributions
  - *wsl –list --online*



Administrator: Windows PowerShell

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\WINDOWS\system32> wsl -l -v
  NAME      STATE           VERSION
* Ubuntu    Running         2
PS C:\WINDOWS\system32> wsl --list
Windows Subsystem for Linux Distributions:
Ubuntu (Default)
PS C:\WINDOWS\system32> wsl --list --online
The following is a list of valid distributions that can be installed.
Install using 'wsl --install -d <Distro>'.

NAME            FRIENDLY NAME
Ubuntu          Ubuntu
Debian          Debian GNU/Linux
kali-linux      Kali Linux Rolling
openSUSE-42     openSUSE Leap 42
SLES-12         SUSE Linux Enterprise Server v12
Ubuntu-16.04    Ubuntu 16.04 LTS
Ubuntu-18.04    Ubuntu 18.04 LTS
Ubuntu-20.04    Ubuntu 20.04 LTS
PS C:\WINDOWS\system32>
```

- **Manage Windows Virtualization (requires Administrative authority)**

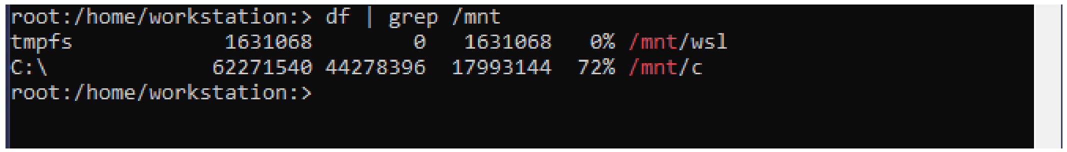
  Enables Windows virtualization (reboot required)
  - *bcdedit /set hypervisorlaunchtype auto*

  Disables Windows virtualization (reboot required)
  - *bcdedit /set hypervisorlaunchtype off*

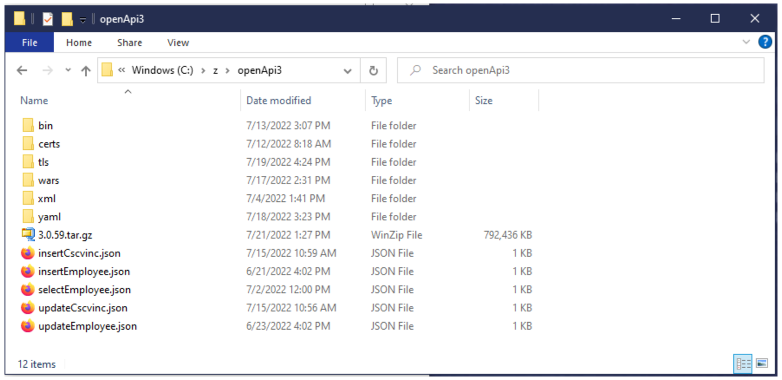
Useful Linux Commands

Display Windows related filesystems

- *df | grep /mnt*



```
root:/home/workstation:> df | grep /mnt
tmpfs              1631068        0   1631068   0% /mnt/wsl
C:\               62271540 44278396  17993144  72% /mnt/c
root:/home/workstation:>
```



openApi3

Windows (C:) › z › openApi3

Name	Date modified	Type	Size
bin	7/13/2022 3:07 PM	File folder	
certs	7/12/2022 8:18 AM	File folder	
tls	7/19/2022 4:24 PM	File folder	
wars	7/17/2022 2:31 PM	File folder	
xml	7/4/2022 1:41 PM	File folder	
yaml	7/18/2022 3:23 PM	File folder	
3.0.59.tar.gz	7/21/2022 1:27 PM	WinZip File	792,436 KB
insertCscvinc.json	7/15/2022 10:59 AM	JSON File	1 KB
insertEmployee.json	6/21/2022 4:02 PM	JSON File	1 KB
selectEmployee.json	7/2/2022 12:00 PM	JSON File	1 KB
updateCscvinc.json	7/15/2022 10:56 AM	JSON File	1 KB
updateEmployee.json	6/23/2022 4:02 PM	JSON File	1 KB

12 items

Copy files from the host to the Linux container (using the sudo command)

- *sudo cp /mnt/c/z/openApi3/xml/*.xml .*
- *sudo cp /mnt/c/z/openApi3/yaml/*.yaml .*

A dot means the current directory.

Or use the sudo command to switch to root authority

- *sudo su root*
- *cp /mnt/c/z/openApi3/xml/*.xml .*
- *cp /mnt/c/z/openApi3/yaml/*.yaml .*

A dot means the current directory.

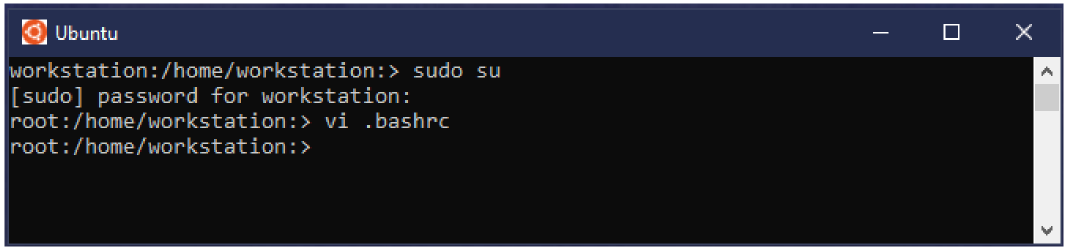
Copy files from the Linux container to the host

- *cp /home/workstation/docker/cscvinc/project/build/libs/api.war /mnt/c/z/openApi3/wars/cscvinc.war*

Customized the Linux container shell environment

- Add these lines to file ***.bashrc*** in the Linux home directory

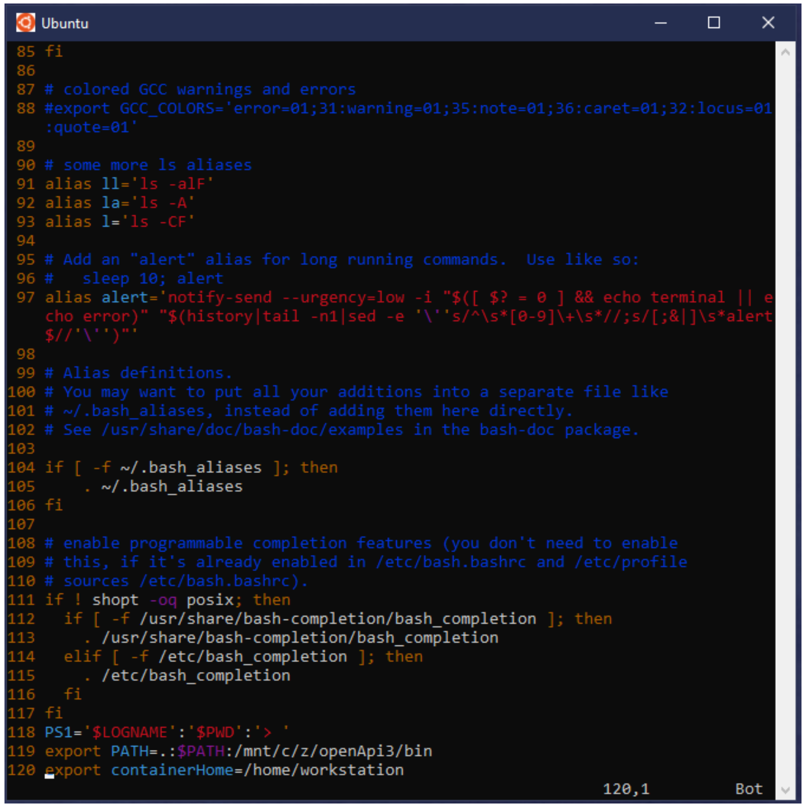
```
PS1='$LOGNAME':'$PWD':'> '
export PATH=.:$PATH:/mnt/c/z/openApi3/bin
export containerHome=/home/workstation
```



```
workstation:/home/workstation:> sudo su
[sudo] password for workstation:
root:/home/workstation:> vi .bashrc
root:/home/workstation:>
```

- Create a file named ***.exrc*** in the Linux home directory

```
set showmode
set redraw
set wrapmargin=3
set nu
```



```
fi

# colored GCC warnings and errors
#export GCC_COLORS='error=01;31:warning=01;35:note=01;36:caret=01;32:locus=01
:quote=01'

# some more ls aliases
alias ll='ls -alF'
alias la='ls -A'
alias l='ls -CF'

# Add an "alert" alias for long running commands.  Use like so:
#   sleep 10; alert
alias alert='notify-send --urgency=low -i "$([ $? = 0 ] && echo terminal || e
cho error)" "$(history|tail -n1|sed -e '\''s/^\s*[0-9]\+\s*//;s/[;&|]\s*alert
$//'\'')"'

# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi
PS1='$LOGNAME':'$PWD':'> '
export PATH=.:$PATH:/mnt/c/z/openApi3/bin
export containerHome=/home/workstation
                                                        120,1         Bot
```

The WSC recommended default *docker-compose.yaml* file

```
version: "3.2"
services:
    zosConnect:
        image: icr.io/zosconnect/ibm-zcon-designer:3.0.59
        user: root
        environment:
             - BASE_PATH=basePath
             - CICS_USER=USER1
             - CICS_PASSWORD=USER1
             - CICS_HOST=wg31.washington.ibm.com
             - CICS_PORT=1491
             - DB2_USERNAME=USER1
             - DB2_PASSWORD=USER1
             - DB2_HOST=wg31.washington.ibm.com
             - DB2_PORT=2446
             - HTTP_PORT=9080
        ports:
            - "9443:9443"
            - "9080:9080"
        volumes:
            ~~- ./CatalogManagerApi:/workspace/project~~
            - ./project:/workspace/project
            - ./logs/:/logs/
            ~~- ./certs:/config/resources/security/:ro~~
            - ./certs:/output/resources/security/
```

The container name is the combination of the project name (e.g., current directory) || zosConnect || _1

Commands related to creating and managing containers

- Start the docker daemon as a background process (note the use &), there is no equivalent with Podman
  dockerd &
- Check to see if the Docker daemon is active
  ps –ef | grep dockerd
- Start a new container or update an existing container using a *docker-compose-yaml* file
  docker-compose -f /home/workstation/docker/sandbox/docker-compose.yaml up -d
- Start a new container using docker-compose-yaml while in directory /home/workstation/docker/sandbox
  docker-compose up –d
- Stop the container using docker-compose command while in directory /home/workstation/docker/sandbox
  docker-compose down
- Start the sandbox container regardless of current directory
  docker start sandbox_zosconnect_1
- Stop the sandbox container regardless of current directory
  docker stop sandbox_zosconnect_1
- Copy server XML override files from a Windows directory into a container's directory*
  docker cp /mnt/c/z/openApi3/xml/. sandbox_zosConnect_1:/config/configDropins/overrides

Commands for managing containers

- List the active containers

  docker ps

```
CONTAINER ID   IMAGE                                           COMMAND                  CREATED          STATUS
PORTS                                                                                     NAMES
97756ede6692   icr.io/zosconnect/ibm-zcon-designer:3.0.55   "/opt/ibm/helpers/ru…"   26 hours ago    Up 26 hours
0.0.0.0:9088->9080/tcp, :::9088->9080/tcp, 0.0.0.0:9429->9443/tcp, :::9429->9443/tcp   employees_zosConnect_1
642f17a4063a   icr.io/zosconnect/ibm-zcon-designer:3.0.55   "/opt/ibm/helpers/ru…"   47 hours ago    Up 20 hours
0.0.0.0:9082->9080/tcp, :::9082->9080/tcp, 0.0.0.0:9445->9443/tcp, :::9445->9443/tcp   sandbox_zosConnect_1
```

- List all active and stopped containers

  docker ps -a

- Remove a container by name or container ID

  docker rm sandbox_zosconnect_1

  or

  docker rm **642f17a4063a**

- Invoke a command in the container

  docker exec -it sandbox_zosConnect_1 bash

Creating a new container in Linux

- Make new Linux directory (project) for the container

  mkdir sandbox
- Change location to the new directory

  cd sandbox
- Make a “configuration” path directory

  mkdir -p project/src/main/liberty/config
- Copy server XML configuration file from the Windows to the container’s “configuration” directory

  cp /mnt/c/z/openApi3/xml/* project/src/main/liberty/config
- Make the certs and logs subdirectories

  mkdir certs

  mkdir logs
- Copy the base docker-compose.yaml file from Windows into the current directory

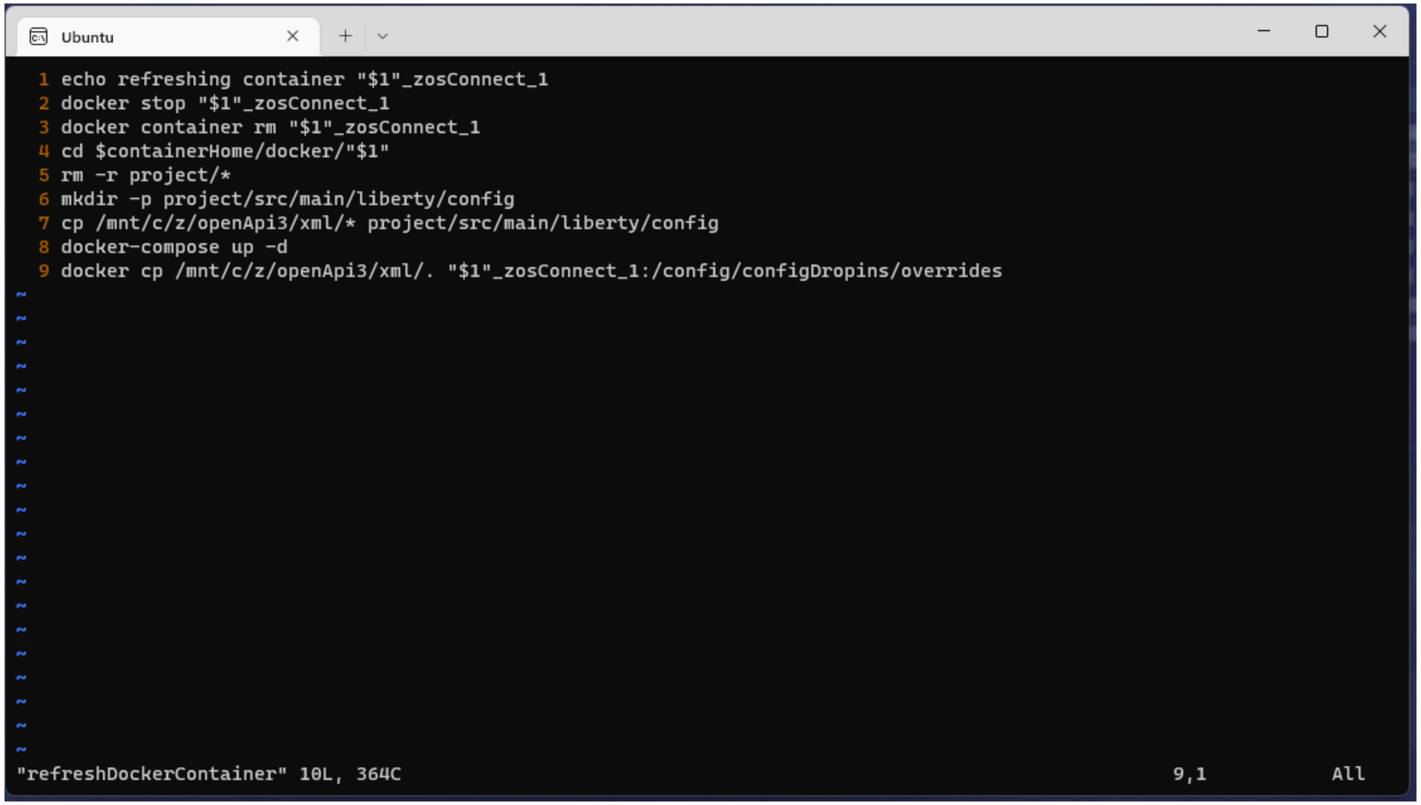
  cp /mnt/c/z/openApi3/yaml/docker-compose.yaml .
- Edit docker-compose.yaml file and make the ports unique

  vi docker-compose.yaml
- Start the container

  docker -compose up –d
- Copy a server’s default XML override files from Windows into a container’s directory*

  docker cp /mnt/c/z/openApi3/xml/. sandbox_zosConnect_1:/config/configDropins/overrides

Linux script refreshDockerContainer



Ubuntu

```
echo refreshing container "$1"_zosConnect_1
docker stop "$1"_zosConnect_1
docker container rm "$1"_zosConnect_1
cd $containerHome/docker/"$1"
rm -r project/*
mkdir -p project/src/main/liberty/config
cp /mnt/c/z/openApi3/xml/* project/src/main/liberty/config
docker-compose up -d
docker cp /mnt/c/z/openApi3/xml/. "$1"_zosConnect_1:/config/configDropins/overrides
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"refreshDockerContainer" 10L, 364C                                                    9,1           All
```

refreshDockerContainer myContainer

Commands to refresh a container

- Remove the container

```
  docker rm sandbox_zosConnect_1
```

- Set location to the container's Linux directory

```
  cd /home/workstation/docker/sandbox
```

- Remove the subdirectories from under the project directory

```
  rm -r project/*
```

- Create the project directory subdirectory structure

```
  mkdir -p project/src/main/liberty/config
```

- Copy the server XML files into the Linux config directory

```
  cp /mnt/c/z/openApi3/xml/*  project/src/main/liberty/config
```

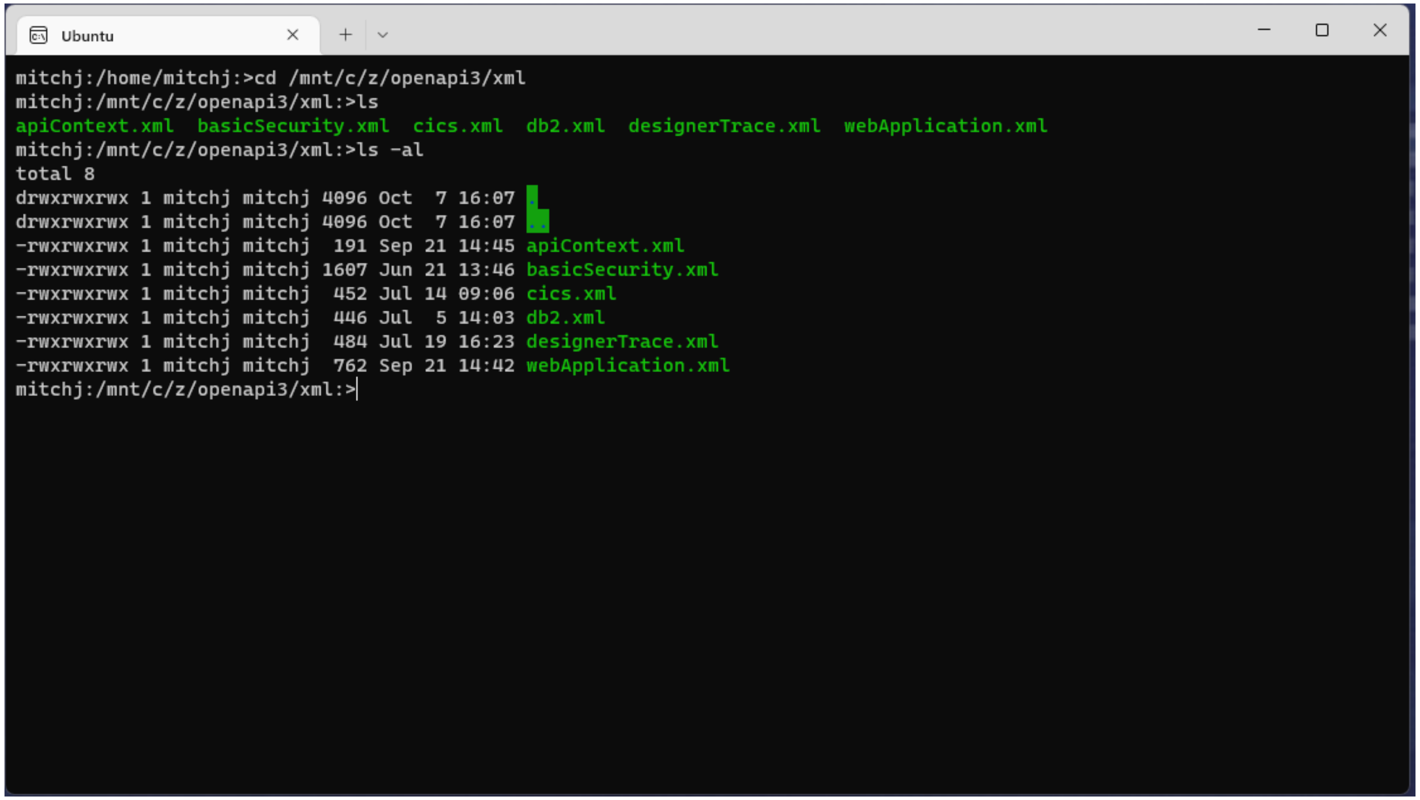
- Start the container

```
  docker-compose up -d
```

- Copy a server's default XML files into the container's config overrides directory

```
  docker cp /mnt/c/z/openApi3/xml/.
            sandbox_zosConnect_1:/config/configDropins/overrides
```

Contents of /mnt/c/z/openApi3/xml



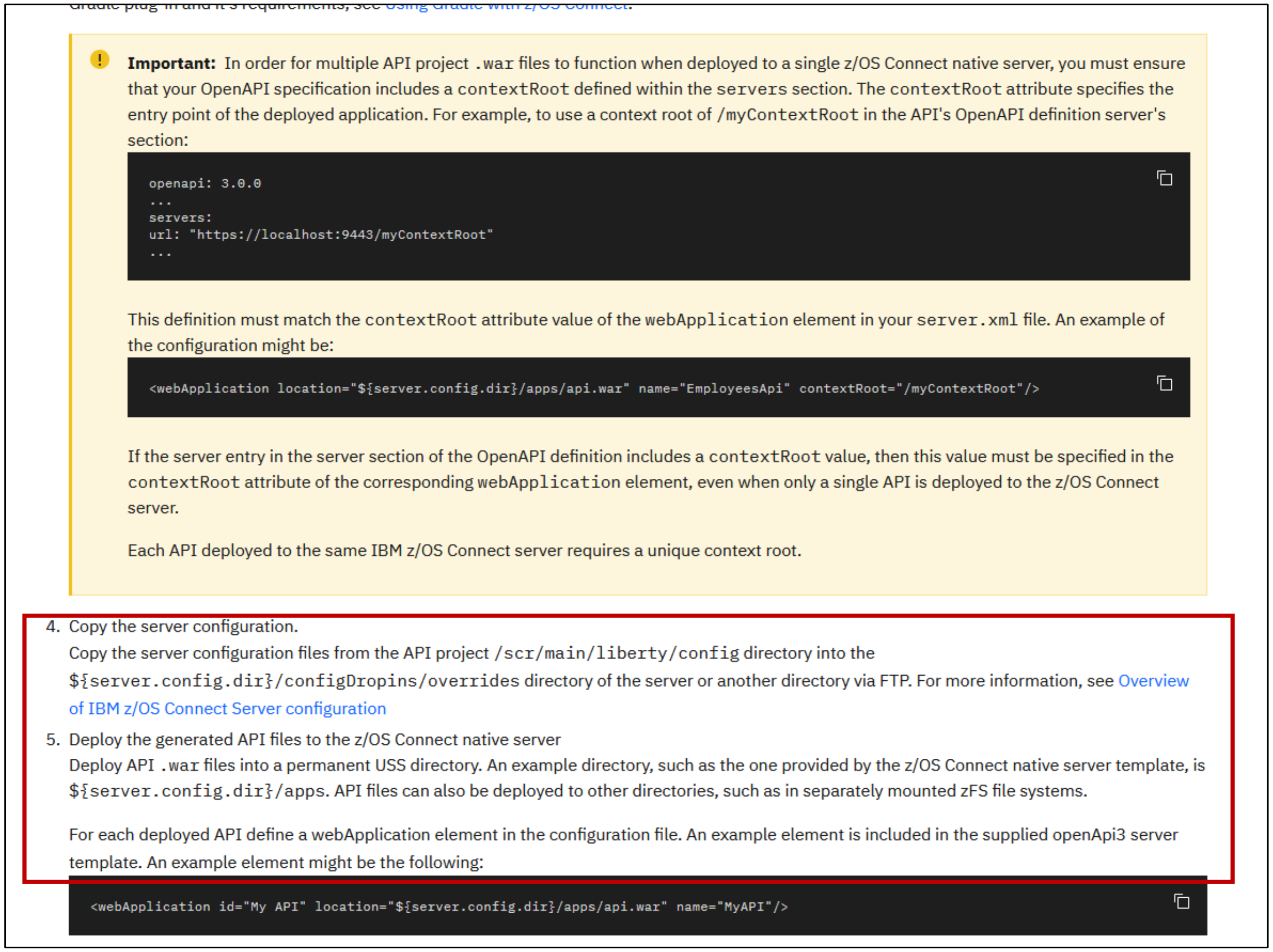
```
mitchj:/home/mitchj:>cd /mnt/c/z/openapi3/xml
mitchj:/mnt/c/z/openapi3/xml:>ls
apiContext.xml  basicSecurity.xml  cics.xml  db2.xml  designerTrace.xml  webApplication.xml
mitchj:/mnt/c/z/openapi3/xml:>ls -al
total 8
drwxrwxrwx 1 mitchj mitchj 4096 Oct  7 16:07 .
drwxrwxrwx 1 mitchj mitchj 4096 Oct  7 16:07 ..
-rwxrwxrwx 1 mitchj mitchj  191 Sep 21 14:45 apiContext.xml
-rwxrwxrwx 1 mitchj mitchj 1607 Jun 21 13:46 basicSecurity.xml
-rwxrwxrwx 1 mitchj mitchj  452 Jul 14 09:06 cics.xml
-rwxrwxrwx 1 mitchj mitchj  446 Jul  5 14:03 db2.xml
-rwxrwxrwx 1 mitchj mitchj  484 Jul 19 16:23 designerTrace.xml
-rwxrwxrwx 1 mitchj mitchj  762 Sep 21 14:42 webApplication.xml
mitchj:/mnt/c/z/openapi3/xml:>
```

Contents of /mnt/c/z/openApi3/xml/apiContext.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<server>

<webApplication id="myApi" name="${BASE_PATH}"
contextRoot="/${BASE_PATH}"
 location="${server.config.dir}dropins/api.war" />

</server>
```

z/OS Server Issues and Considerations – API Deployment

https://www.ibm.com/docs/en/zos-connect/zos-connect/3.0?topic=server-devops-overview



Gradle plug-in and it's requirements, see Using Gradle with z/OS Connect.

Important: In order for multiple API project `.war` files to function when deployed to a single z/OS Connect native server, you must ensure that your OpenAPI specification includes a `contextRoot` defined within the `servers` section. The `contextRoot` attribute specifies the entry point of the deployed application. For example, to use a context root of `/myContextRoot` in the API's OpenAPI definition server's section:

```
openapi: 3.0.0
...
servers:
url: "https://localhost:9443/myContextRoot"
...
```

This definition must match the `contextRoot` attribute value of the `webApplication` element in your `server.xml` file. An example of the configuration might be:

```
<webApplication location="${server.config.dir}/apps/api.war" name="EmployeesApi" contextRoot="/myContextRoot"/>
```

If the server entry in the server section of the OpenAPI definition includes a `contextRoot` value, then this value must be specified in the `contextRoot` attribute of the corresponding `webApplication` element, even when only a single API is deployed to the z/OS Connect server.

Each API deployed to the same IBM z/OS Connect server requires a unique context root.

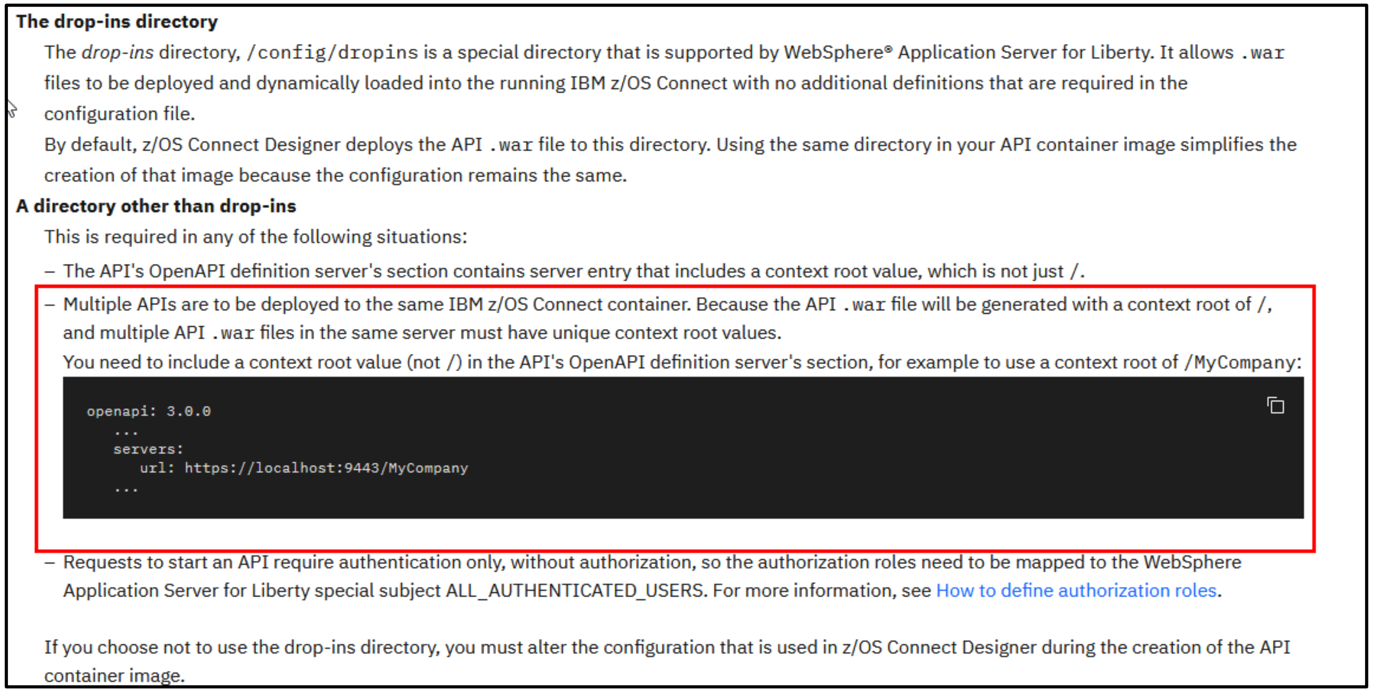
4. Copy the server configuration.
   Copy the server configuration files from the API project `/scr/main/liberty/config` directory into the `${server.config.dir}/configDropins/overrides` directory of the server or another directory via FTP. For more information, see Overview of IBM z/OS Connect Server configuration
5. Deploy the generated API files to the z/OS Connect native server
   Deploy API `.war` files into a permanent USS directory. An example directory, such as the one provided by the z/OS Connect native server template, is `${server.config.dir}/apps`. API files can also be deployed to other directories, such as in separately mounted zFS file systems.

   For each deployed API define a webApplication element in the configuration file. An example element is included in the supplied openApi3 server template. An example element might be the following:

```
<webApplication id="My API" location="${server.config.dir}/apps/api.war" name="MyAPI"/>
```

z/OS Server Issues and Considerations – Context Root

https://www.ibm.com/docs/en/zos-connect/zos-connect/3.0?topic=image-devops-overview



The drop-ins directory

The *drop-ins* directory, `/config/dropins` is a special directory that is supported by WebSphere® Application Server for Liberty. It allows `.war` files to be deployed and dynamically loaded into the running IBM z/OS Connect with no additional definitions that are required in the configuration file.

By default, z/OS Connect Designer deploys the API `.war` file to this directory. Using the same directory in your API container image simplifies the creation of that image because the configuration remains the same.

A directory other than drop-ins

This is required in any of the following situations:

– The API's OpenAPI definition server's section contains server entry that includes a context root value, which is not just `/`.

– Multiple APIs are to be deployed to the same IBM z/OS Connect container. Because the API `.war` file will be generated with a context root of `/`, and multiple API `.war` files in the same server must have unique context root values.
You need to include a context root value (not `/`) in the API's OpenAPI definition server's section, for example to use a context root of `/MyCompany`:

```
openapi: 3.0.0
   ...
   servers:
      url: https://localhost:9443/MyCompany
   ...
```

– Requests to start an API require authentication only, without authorization, so the authorization roles need to be mapped to the WebSphere Application Server for Liberty special subject ALL_AUTHENTICATED_USERS. For more information, see How to define authorization roles.

If you choose not to use the drop-ins directory, you must alter the configuration that is used in z/OS Connect Designer during the creation of the API container image.

- **Required to define applications and add context root**

```
<webApplication id="cics" contextRoot="/cics"  name="cicsAPI"
      location="${server.config.dir}apps/cscvinc.war"/>
<webApplication id="db2" contextRoot="/db2"  name="db2API"
      location="${server.config.dir}apps/employees.war"/>
```

Contents of /mnt/c/z/openApi3/xml/cics.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<server description="IPIC connection to CICS">
    <featureManager>
        <feature>zosconnect:cics-1.0</feature>
    </featureManager>

    <zosconnect_cicsIpicConnection id="cicsConn"
        host="${CICS_HOST}"
        port="${CICS_PORT}"
        authDataRef="cicsCredentials" />

    <zosconnect_authData id="cicsCredentials"
        user="${CICS_USER}"
        password="${CICS_PASSWORD}" />

</server>
```

Contents of /mnt/c/z/openApi3/xml/db2.xml

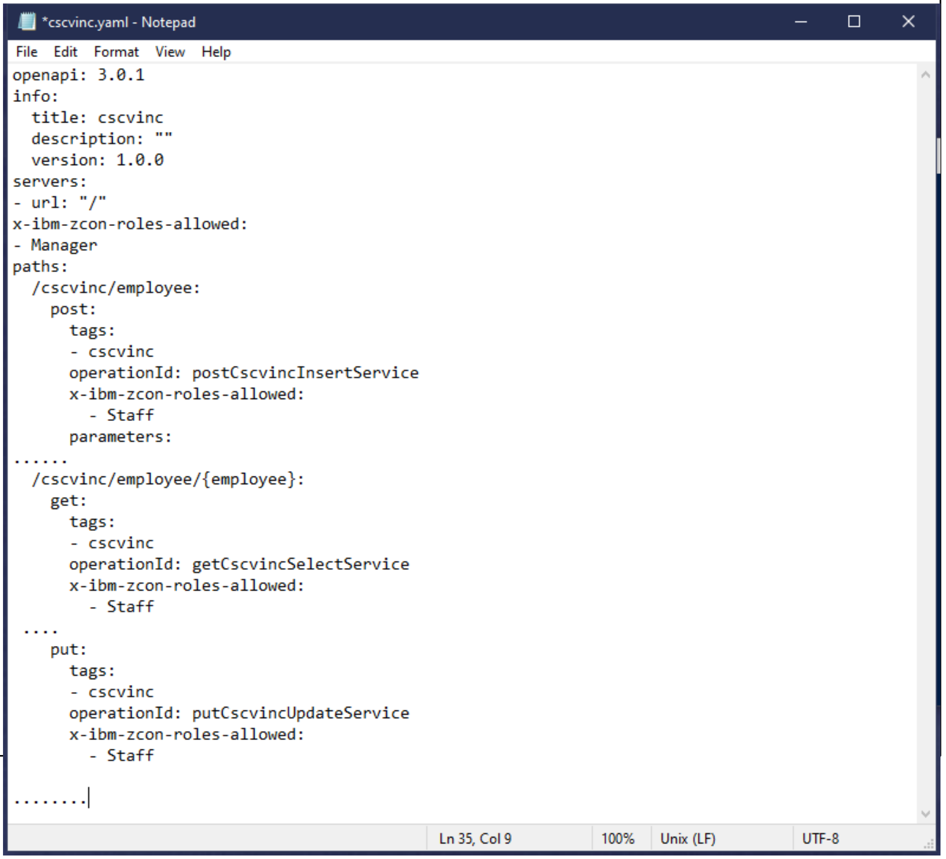
```
<?xml version="1.0" encoding="UTF-8"?>
<server description="Db2 zosconnect_db2Connection ">
    <featureManager>
        <feature>zosconnect:db2-1.0</feature>
    </featureManager>

<zosconnect_db2Connection id="db2Conn"
    host="${DB2_HOST}"
    port="${DB2_PORT}"
    credentialRef="commonCredentials" />

<zosconnect_credential id="commonCredentials"
    user="${DB2_USERNAME}"
    password="${DB2_PASSWORD}" />

</server>
```

Contents of /mnt/c/z/openApi3/xml/basicSecurity.xml (1 of 2)

```
<server description="basic security">

    <!-- Enable features -->
    <featureManager>
        <feature>appSecurity-2.0</feature>
        <feature>restConnector-2.0</feature>
    </featureManager>

    <webAppSecurity allowFailOverToBasicAuth="true" />

    <basicRegistry id="basic" realm="zosConnect">
        <user name="Fred" password="fredpwd" />
        <user name="user1" password="user1" />
        <user name="user2" password="user2" />
        <user name="user3" password="user3" />
        <group name="Manager">
           <member name="Fred"/>
        </group>
        <group name="Staff">
           <member name="Fred"/>
           <member name="user1"/>
           <member name="user2"/>
</group>
  </basicRegistry>
```

Contents of /mnt/c/z/openApi3/xml/basicSecurity.xml (2 of 2)

```
<administrator-role>
       <user>Fred</user>
       <group>staffGroup</group>
   </administrator-role>

<authorization-roles id="Manager">
        <security-role name="Manager">
            <group name="managerGroup"/>
        </security-role>
     </authorization-roles>
     <authorization-roles id="Staff">
        <security-role name="Staff">
            <group name="staffGroup"/>
        </security-role>
    </authorization-roles>

</server>
```

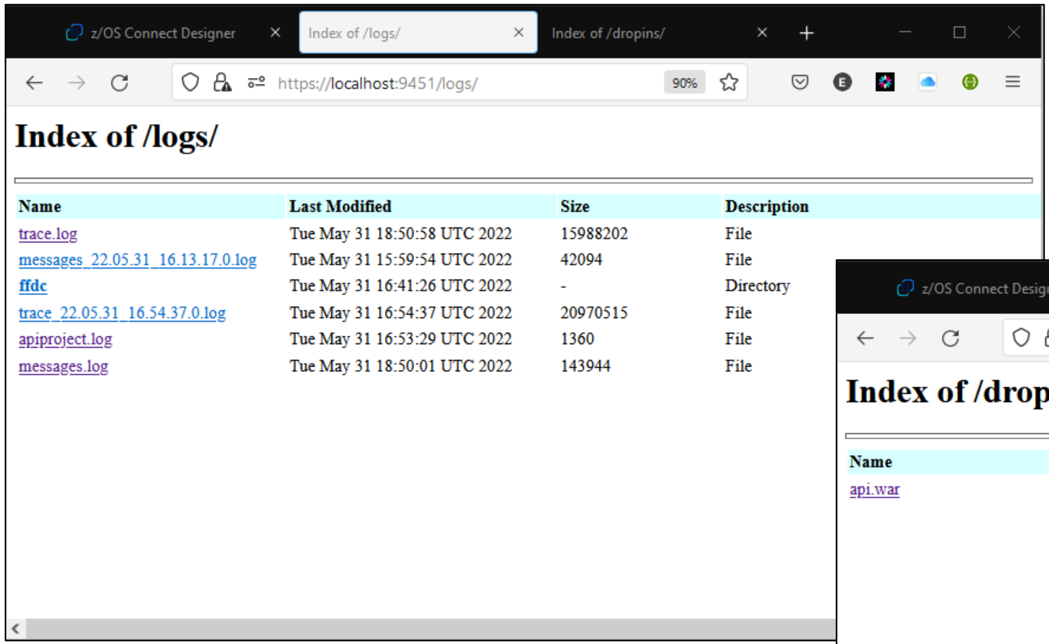


```
openapi: 3.0.1
info:
  title: cscvinc
  description: ""
  version: 1.0.0
servers:
- url: "/"
x-ibm-zcon-roles-allowed:
- Manager
paths:
  /cscvinc/employee:
    post:
      tags:
      - cscvinc
      operationId: postCscvincInsertService
      x-ibm-zcon-roles-allowed:
        - Staff
      parameters:
......
  /cscvinc/employee/{employee}:
    get:
      tags:
      - cscvinc
      operationId: getCscvincSelectService
      x-ibm-zcon-roles-allowed:
        - Staff
 ....
    put:
      tags:
      - cscvinc
      operationId: putCscvincUpdateService
      x-ibm-zcon-roles-allowed:
        - Staff

........
```

Contents of /mnt/c/z/openApi3/xml/webApplication.xml

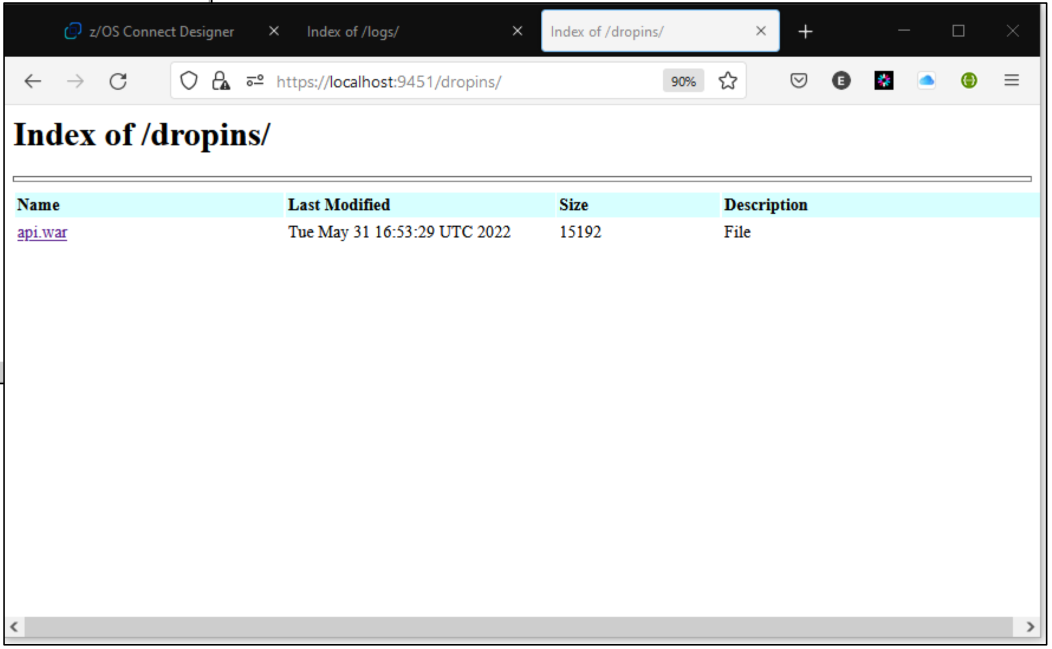
```
<?xml version="1.0" encoding="UTF-8"?>
<server description="Default server">
<webApplication id="resources-dropins" name="dropins"
   location="/opt/ibm/wlp/usr/servers/defaultServer/dropins">
    <web-ext context-root="dropins"
      enable-file-serving="true" enable-directory-browsing="true">
      <file-servering-attribute name="enxtendDocumentRoot"
       value="/opt/ibm/wlp/usr/servers/defaultServer/dropins" />
    </web-ext>
</webApplication> >
<webApplication id="resources-logs" name="logs"
   location="/logs">
    <web-ext context-root="logs"
      enable-file-serving="true" enable-directory-browsing="true">
      <file-servering-attribute name="enxtendDocumentRoot"
       value="/logs" />
    </web-ext>
</webApplication> >
</server>
```

Provides access to the logs and traces as well as the WAR file



Index of /logs/

Name	Last Modified	Size	Description
trace.log	Tue May 31 18:50:58 UTC 2022	15988202	File
messages_22.05.31_16.13.17.0.log	Tue May 31 15:59:54 UTC 2022	42094	File
ffdc	Tue May 31 16:41:26 UTC 2022	-	Directory
trace_22.05.31_16.54.37.0.log	Tue May 31 16:54:37 UTC 2022	20970515	File
apiproject.log	Tue May 31 16:53:29 UTC 2022	1360	File
messages.log	Tue May 31 18:50:01 UTC 2022	143944	File



Index of /dropins/

Name	Last Modified	Size	Description
api.war	Tue May 31 16:53:29 UTC 2022	15192	File

Contents of /mnt/c/z/openApi3/xml/designerTrace.xml

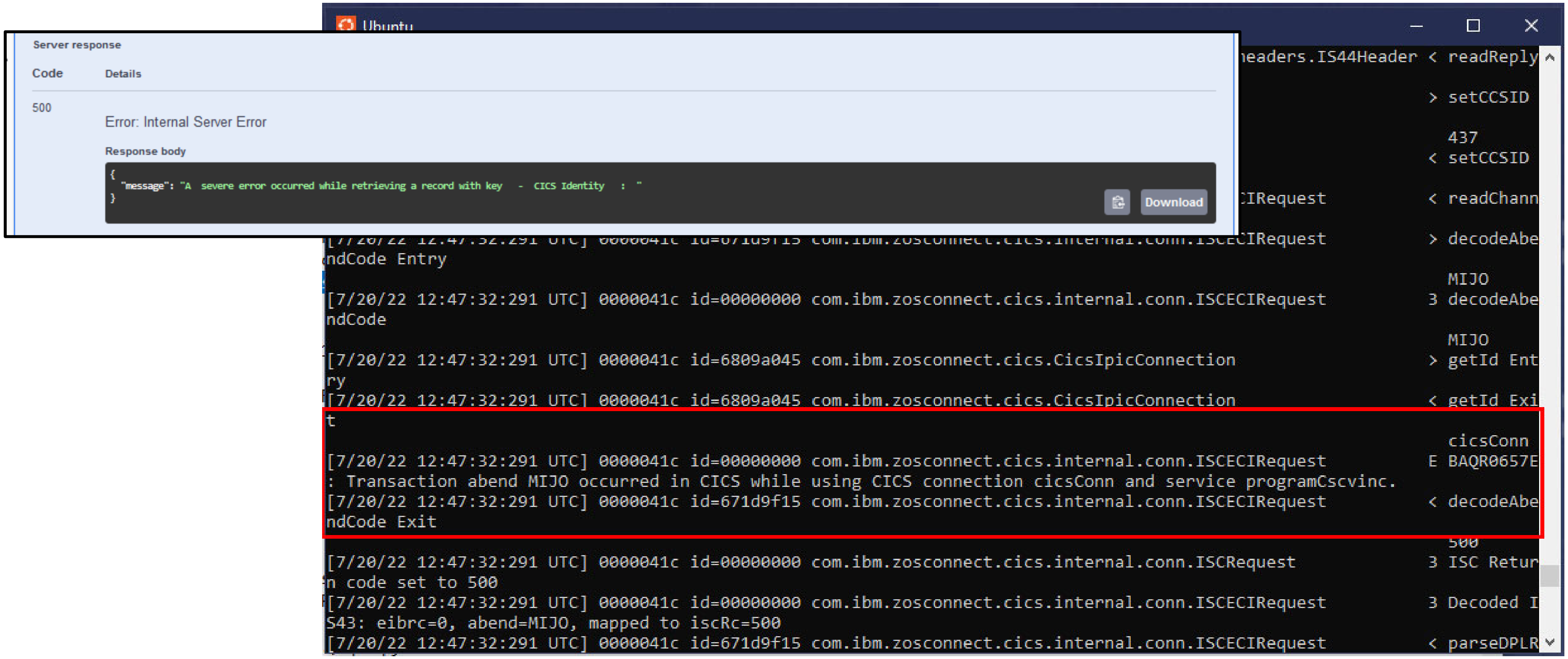
```
<?xml version="1.0" encoding="UTF-8"?>
<server description="trace specifications">

<logging traceSpecification="
         com.ibm.ws.security.*=all:
         zosConnectCics=all:
         zosConnectDb2=all:
         SSLChannel=all:
         SSL=all:
         "/>
</server>
```

Contents of /mnt/c/z/openApi3/trace/cicsDb2Trace.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<server description="trace specifications">
<logging traceSpecification="
          zosConnectCics=all:
          zosConnectDb2=all
         "/>
</server>
```

- Reset trace specification → ***docker cp /mnt/c/z/openApi3/tls/cicsDb2Trace.xml cscvinc_zosConnect_1:/config/configDropins/overrides/designerTrace.xml***

Tracing the CICS connection using Linux command *tail –f logs/trace.out*



Server response

Code Details

500

Error: Internal Server Error

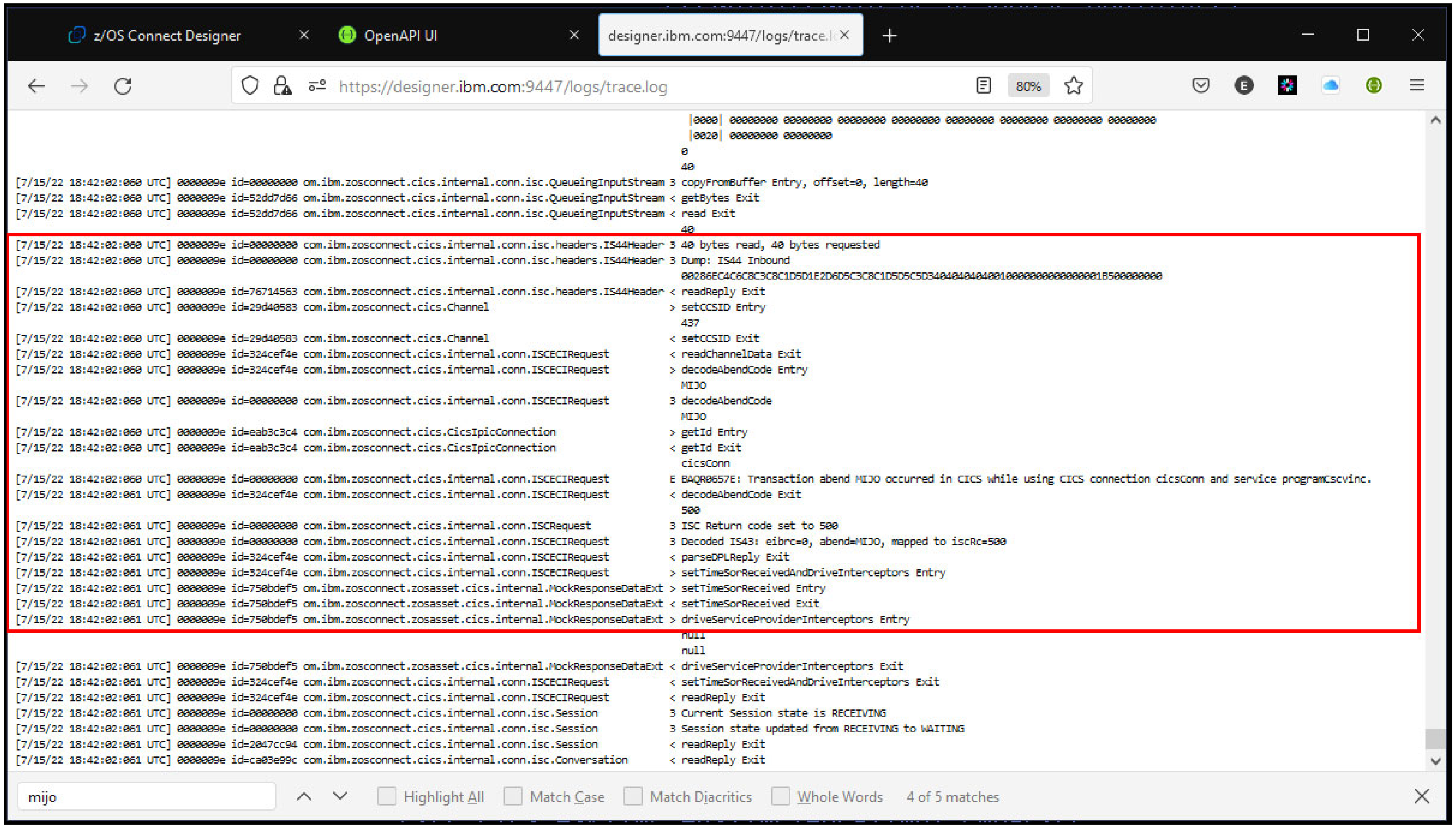
Response body

```
{
  "message": "A  severe error occurred while retrieving a record with key   -  CICS Identity   :  "
}
```

```
eaders.IS44Header < readReply
                  > setCCSID
                    437
                  < setCCSID
CIRequest         < readChann
[7/20/22 12:47:32:291 UTC] 0000041c id=671d9f15 com.ibm.zosconnect.cics.internal.conn.ISCECIRequest > decodeAbe
ndCode Entry
                                                                                                       MIJO
[7/20/22 12:47:32:291 UTC] 0000041c id=00000000 com.ibm.zosconnect.cics.internal.conn.ISCECIRequest 3 decodeAbe
ndCode
                                                                                                       MIJO
[7/20/22 12:47:32:291 UTC] 0000041c id=6809a045 com.ibm.zosconnect.cics.CicsIpicConnection           > getId Ent
ry
[7/20/22 12:47:32:291 UTC] 0000041c id=6809a045 com.ibm.zosconnect.cics.CicsIpicConnection           < getId Exi
t
                                                                                                       cicsConn
[7/20/22 12:47:32:291 UTC] 0000041c id=00000000 com.ibm.zosconnect.cics.internal.conn.ISCECIRequest E BAQR0657E
: Transaction abend MIJO occurred in CICS while using CICS connection cicsConn and service programCscvinc.
[7/20/22 12:47:32:291 UTC] 0000041c id=671d9f15 com.ibm.zosconnect.cics.internal.conn.ISCECIRequest < decodeAbe
ndCode Exit
                                                                                                       500
[7/20/22 12:47:32:291 UTC] 0000041c id=00000000 com.ibm.zosconnect.cics.internal.conn.ISCRequest     3 ISC Retur
n code set to 500
[7/20/22 12:47:32:291 UTC] 0000041c id=00000000 com.ibm.zosconnect.cics.internal.conn.ISCECIRequest 3 Decoded I
S43: eibrc=0, abend=MIJO, mapped to iscRc=500
[7/20/22 12:47:32:291 UTC] 0000041c id=671d9f15 com.ibm.zosconnect.cics.internal.conn.ISCECIRequest < parseDPLR
```

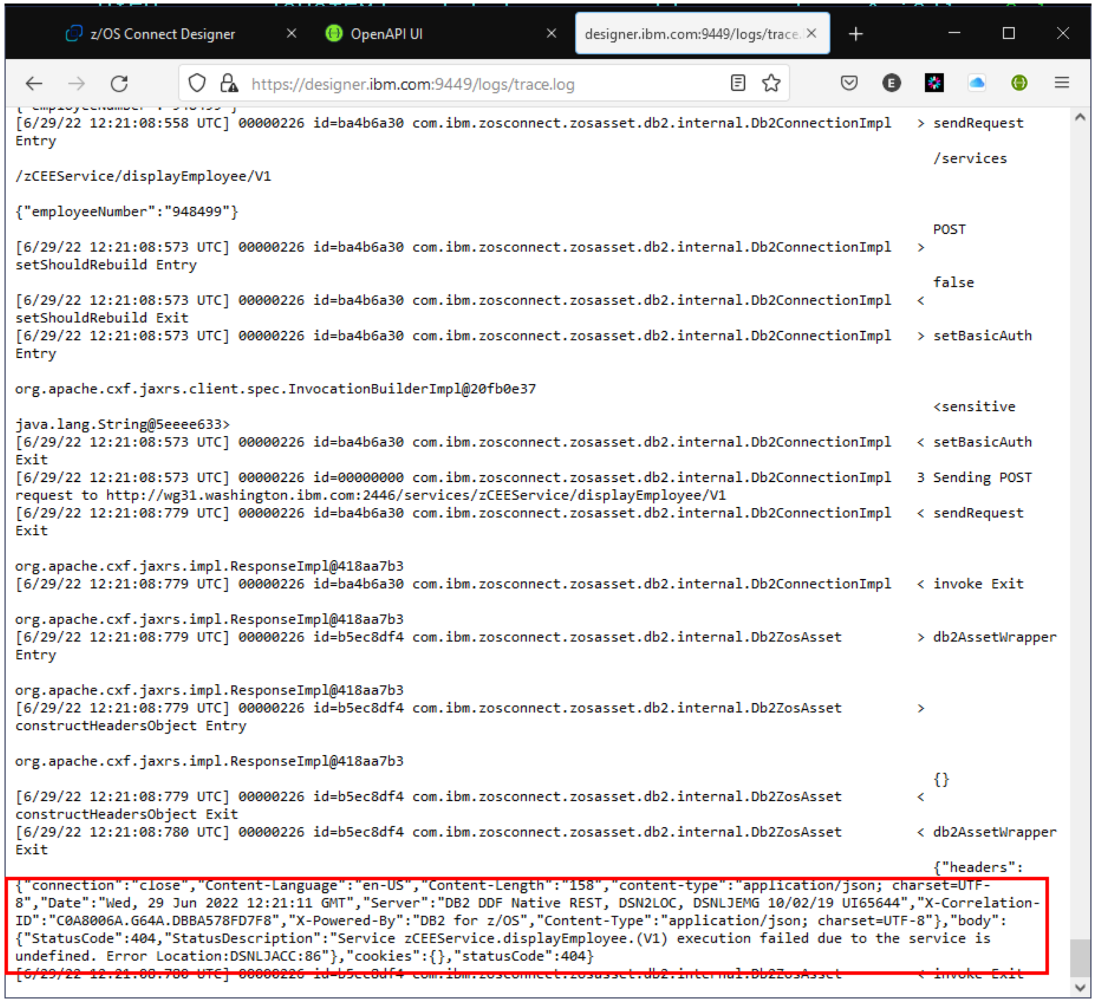
- Real time monitoring of the trace output → ***tail –f logs/trace.out***

Designer displays an HTTP 500 return code


z/OS Connect Designer
OpenAPI UI
designer.ibm.com:9447/logs/trace.l
https://designer.ibm.com:9447/logs/trace.log
80%

```
                                                                                                               |0000| 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
                                                                                                               |0020| 00000000 00000000
                                                                                                              0
                                                                                                              40
[7/15/22 18:42:02:060 UTC] 0000009e id=00000000 om.ibm.zosconnect.cics.internal.conn.isc.QueueingInputStream 3 copyFromBuffer Entry, offset=0, length=40
[7/15/22 18:42:02:060 UTC] 0000009e id=52dd7d66 om.ibm.zosconnect.cics.internal.conn.isc.QueueingInputStream < getBytes Exit
[7/15/22 18:42:02:060 UTC] 0000009e id=52dd7d66 om.ibm.zosconnect.cics.internal.conn.isc.QueueingInputStream < read Exit
                                                                                                              40
[7/15/22 18:42:02:060 UTC] 0000009e id=00000000 com.ibm.zosconnect.cics.internal.conn.isc.headers.IS44Header 3 40 bytes read, 40 bytes requested
[7/15/22 18:42:02:060 UTC] 0000009e id=00000000 com.ibm.zosconnect.cics.internal.conn.isc.headers.IS44Header 3 Dump: IS44 Inbound
                                                                                                              00286EC4C6C8C3C8C1D5D1E2D6D5C3C8C1D5D5C5D34040404040400100000000000000001B500000000
[7/15/22 18:42:02:060 UTC] 0000009e id=76714563 com.ibm.zosconnect.cics.internal.conn.isc.headers.IS44Header < readReply Exit
[7/15/22 18:42:02:060 UTC] 0000009e id=29d40583 com.ibm.zosconnect.cics.Channel                              > setCCSID Entry
                                                                                                              437
[7/15/22 18:42:02:060 UTC] 0000009e id=29d40583 com.ibm.zosconnect.cics.Channel                              < setCCSID Exit
[7/15/22 18:42:02:060 UTC] 0000009e id=324cef4e com.ibm.zosconnect.cics.internal.conn.ISCECIRequest          < readChannelData Exit
[7/15/22 18:42:02:060 UTC] 0000009e id=324cef4e com.ibm.zosconnect.cics.internal.conn.ISCECIRequest          > decodeAbendCode Entry
                                                                                                              MIJO
[7/15/22 18:42:02:060 UTC] 0000009e id=00000000 com.ibm.zosconnect.cics.internal.conn.ISCECIRequest          3 decodeAbendCode
                                                                                                              MIJO
[7/15/22 18:42:02:060 UTC] 0000009e id=eab3c3c4 com.ibm.zosconnect.cics.CicsIpicConnection                   > getId Entry
[7/15/22 18:42:02:060 UTC] 0000009e id=eab3c3c4 com.ibm.zosconnect.cics.CicsIpicConnection                   < getId Exit
                                                                                                              cicsConn
[7/15/22 18:42:02:060 UTC] 0000009e id=00000000 com.ibm.zosconnect.cics.internal.conn.ISCECIRequest          E BAQR0657E: Transaction abend MIJO occurred in CICS while using CICS connection cicsConn and service programCscvinc.
[7/15/22 18:42:02:061 UTC] 0000009e id=324cef4e com.ibm.zosconnect.cics.internal.conn.ISCECIRequest          < decodeAbendCode Exit
                                                                                                              500
[7/15/22 18:42:02:061 UTC] 0000009e id=00000000 com.ibm.zosconnect.cics.internal.conn.ISCRequest             3 ISC Return code set to 500
[7/15/22 18:42:02:061 UTC] 0000009e id=00000000 com.ibm.zosconnect.cics.internal.conn.ISCECIRequest          3 Decoded IS43: eibrc=0, abend=MIJO, mapped to iscRc=500
[7/15/22 18:42:02:061 UTC] 0000009e id=324cef4e com.ibm.zosconnect.cics.internal.conn.ISCECIRequest          < parseDPLReply Exit
[7/15/22 18:42:02:061 UTC] 0000009e id=324cef4e com.ibm.zosconnect.cics.internal.conn.ISCECIRequest          > setTimeSorReceivedAndDriveInterceptors Entry
[7/15/22 18:42:02:061 UTC] 0000009e id=750bdef5 om.ibm.zosconnect.zosasset.cics.internal.MockResponseDataExt > setTimeSorReceived Entry
[7/15/22 18:42:02:061 UTC] 0000009e id=750bdef5 om.ibm.zosconnect.zosasset.cics.internal.MockResponseDataExt < setTimeSorReceived Exit
[7/15/22 18:42:02:061 UTC] 0000009e id=750bdef5 om.ibm.zosconnect.zosasset.cics.internal.MockResponseDataExt > driveServiceProviderInterceptors Entry
                                                                                                              null
                                                                                                              null
[7/15/22 18:42:02:061 UTC] 0000009e id=750bdef5 om.ibm.zosconnect.zosasset.cics.internal.MockResponseDataExt < driveServiceProviderInterceptors Exit
[7/15/22 18:42:02:061 UTC] 0000009e id=324cef4e com.ibm.zosconnect.cics.internal.conn.ISCECIRequest          < setTimeSorReceivedAndDriveInterceptors Exit
[7/15/22 18:42:02:061 UTC] 0000009e id=324cef4e com.ibm.zosconnect.cics.internal.conn.ISCECIRequest          < readReply Exit
[7/15/22 18:42:02:061 UTC] 0000009e id=00000000 com.ibm.zosconnect.cics.internal.conn.isc.Session           3 Current Session state is RECEIVING
[7/15/22 18:42:02:061 UTC] 0000009e id=00000000 com.ibm.zosconnect.cics.internal.conn.isc.Session           3 Session state updated from RECEIVING to WAITING
[7/15/22 18:42:02:061 UTC] 0000009e id=2047cc94 com.ibm.zosconnect.cics.internal.conn.isc.Session           < readReply Exit
[7/15/22 18:42:02:061 UTC] 0000009e id=ca03e99c com.ibm.zosconnect.cics.internal.conn.isc.Conversation      < readReply Exit
```

Designer displays an HTTP 500 return code


z/OS Connect Designer
OpenAPI UI
designer.ibm.com:9449/logs/trace
https://designer.ibm.com:9449/logs/trace.log

```
[6/29/22 12:21:08:558 UTC] 00000226 id=ba4b6a30 com.ibm.zosconnect.zosasset.db2.internal.Db2ConnectionImpl    > sendRequest
Entry
                                                                                                                /services
/zCEEService/displayEmployee/V1

{"employeeNumber":"948499"}

                                                                                                                POST
[6/29/22 12:21:08:573 UTC] 00000226 id=ba4b6a30 com.ibm.zosconnect.zosasset.db2.internal.Db2ConnectionImpl    >
setShouldRebuild Entry
                                                                                                                false
[6/29/22 12:21:08:573 UTC] 00000226 id=ba4b6a30 com.ibm.zosconnect.zosasset.db2.internal.Db2ConnectionImpl    <
setShouldRebuild Exit
[6/29/22 12:21:08:573 UTC] 00000226 id=ba4b6a30 com.ibm.zosconnect.zosasset.db2.internal.Db2ConnectionImpl    > setBasicAuth
Entry

org.apache.cxf.jaxrs.client.spec.InvocationBuilderImpl@20fb0e37
                                                                                                                <sensitive
java.lang.String@5eeee633>
[6/29/22 12:21:08:573 UTC] 00000226 id=ba4b6a30 com.ibm.zosconnect.zosasset.db2.internal.Db2ConnectionImpl    < setBasicAuth
Exit
[6/29/22 12:21:08:573 UTC] 00000226 id=00000000 com.ibm.zosconnect.zosasset.db2.internal.Db2ConnectionImpl    3 Sending POST
request to http://wg31.washington.ibm.com:2446/services/zCEEService/displayEmployee/V1
[6/29/22 12:21:08:779 UTC] 00000226 id=ba4b6a30 com.ibm.zosconnect.zosasset.db2.internal.Db2ConnectionImpl    < sendRequest
Exit

org.apache.cxf.jaxrs.impl.ResponseImpl@418aa7b3
[6/29/22 12:21:08:779 UTC] 00000226 id=ba4b6a30 com.ibm.zosconnect.zosasset.db2.internal.Db2ConnectionImpl    < invoke Exit

org.apache.cxf.jaxrs.impl.ResponseImpl@418aa7b3
[6/29/22 12:21:08:779 UTC] 00000226 id=b5ec8df4 com.ibm.zosconnect.zosasset.db2.internal.Db2ZosAsset          > db2AssetWrapper
Entry

org.apache.cxf.jaxrs.impl.ResponseImpl@418aa7b3
[6/29/22 12:21:08:779 UTC] 00000226 id=b5ec8df4 com.ibm.zosconnect.zosasset.db2.internal.Db2ZosAsset          >
constructHeadersObject Entry

org.apache.cxf.jaxrs.impl.ResponseImpl@418aa7b3
                                                                                                                {}
[6/29/22 12:21:08:779 UTC] 00000226 id=b5ec8df4 com.ibm.zosconnect.zosasset.db2.internal.Db2ZosAsset          <
constructHeadersObject Exit
[6/29/22 12:21:08:780 UTC] 00000226 id=b5ec8df4 com.ibm.zosconnect.zosasset.db2.internal.Db2ZosAsset          < db2AssetWrapper
Exit
                                                                                                                {"headers":
{"connection":"close","Content-Language":"en-US","Content-Length":"158","content-type":"application/json; charset=UTF-
8","Date":"Wed, 29 Jun 2022 12:21:11 GMT","Server":"DB2 DDF Native REST, DSN2LOC, DSNLJEMG 10/02/19 UI65644","X-Correlation-
ID":"C0A8006A.G64A.DBBA578FD7F8","X-Powered-By":"DB2 for z/OS","Content-Type":"application/json; charset=UTF-8"},"body":
{"StatusCode":404,"StatusDescription":"Service zCEEService.displayEmployee.(V1) execution failed due to the service is
undefined. Error Location:DSNLJACC:86"},"cookies":{},"statusCode":404}
```

Updating and Moving a z/OS Connect Designer image to another Linux image

- Pull in a new (download) a z/OS Connect Designer image
  docker pull icr.io/zosconnect/ibm-zcon-designer:3.0.59

- Save the z/OS Connect Docker image to a file
  docker save icr.io/zosconnect/ibm-zcon-designer:3.0.59 | gzip > 3.0.59.tar.gz

- Copy the z/OS Connect Docker image file to a Windows directory location
  cp *.tar.gz /mnt/c/z/ftp

- Use FTP to move the image file from the original image to the target Linux image

- Load the z/OS Connect Docker image on the Linux image
  docker load < 3.0.57.tar.gz

Commands for managing the container certificates and key stores

- Import the CICS public certificate authority certificate into the local keystore.

docker exec -it sandbox_zosConnect_1 keytool -importcert   -file /output/resources/security/CICSCA.pem -nopromt -keystore /output/resources/security/zosConnect.jks -storetype PKCS12 –alias cicsca

- Import the Db2 public certificate authority certificate into the local keystore.

docker exec -it sandbox_zosConnect_1 keytool -importcert   -file /output/resources/security/DB2CA.pem -nopromt -keystore /output/resources/security/zosConnect.jks -storetype PKCS12 –alias db2ca

- List the contents of the local keystore

docker exec -it sandbox_zosConnect_1 keytool -v -list -keystore /output/resources/security/zosConnect.jks -storetype PKCS12

Note my use of *docker exec -it sandbox_zosConnect_1* rather than *docker run -it --rm -v /Users/<username>/Desktop/ZCWorkspace/certs:/tmp/cert/output icr.io/zosconnect/ibm-zcon-designer:3.0.56* as documented at **https://www.ibm.com/docs/en/zos-connect/zos-connect/3.0?topic=db2-configuring-connection-basic-authentication-tls**

Commands for managing the container certificates and key stores

- Create a self-signed certificate (and create a local keystore)
- *docker exec -it sandbox_zosConnect_1 keytool -keystore /output/resources/security/cicsKeyStore.jks   -storetype PKCS12 -storepass changeit -genkey -keysize 2048 -alias cicsusr -dname "CN=user1, O=IBM, C=US" -keyalg RSA -validity 365*

- Create a certificate request from the self-signed certificate

docker exec -it sandbox_zosConnect_1 keytool -keystore /output/resources/security/cicsKeyStore.jks -storetype PKCS12 -certreq -alias cicsusr -file /output/resources/security/user1.arm

- Send the certificate request to the certificate authority for signing

- Import the signed personal certificate into the local key store.
- *docker exec -it cscvinc_zosConnect_1 keytool -importcert -file /output/resources/security/user1.PEM -alias cicsusr -storetype PKCS12 --noprompt -keystore /output/resources/security/cicsKeyStore.jks*

The TLS *docker-compose.yaml* file

```
version: "3.2"
services:
    zosConnect:
        image: icr.io/zosconnect/ibm-zcon-designer:3.0.57
        user: root
        environment:
            - CICS_USER=USER1
            - CICS_PASSWORD=USER1
            - CICS_HOST=wg31.washington.ibm.com
            - CICS_PORT=1491
            - CICSTRUSTSTORE_PASSWORD=changeit
            - CICSKEYSTORE_PASSWORD=secret
            - CICSSSL_PORT=1493
            - DB2_USERNAME=USER1
            - DB2_PASSWORD=USER1
            - DB2_HOST=wg31.washington.ibm.com
            - DB2_PORT=2446
            - DB2TRUSTSTORE_PASSWORD=changeit
            - Db2KEYSTORE_PASSWORD=secret
            - DB2SSL_PORT=2445
            - HTTP_PORT=9080
        ports:
            - "9447:9443"
            - "9084:9080"
        volumes:
            - ./project:/workspace/project
            - ./logs/:/logs/
            - ./certs:/output/resources/security/
```

`docker-compose . . .`

`podman rm . . .`
`podman-compose . . .`

Contents of /mnt/c/z/openApi3/tls/cicsTLSServer.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<server description="IPIC connection to CICS">
    <featureManager>
        <feature>zosconnect:cics-1.0</feature>
    </featureManager>
    <zosconnect_cicsIpicConnection id="cicsConn"
      host="${CICS_HOST}" port="${CICSSSL_PORT}"
      authDataRef="cicsCredentials"
      sslCertsRef=cicsSSLSettings" />
   <ssl id="cicsSSLSettings"
     keyStoreRef= "cicsTrustStore"
     trustStoreRef= "cicsTrustStore" />
   <keyStore id= "cicsTrustStore"
     location="/output/resources/security/cicsTrustStore.jks"
     password="${CICSTRUSTSTORE_PASSWORD}" type="PKCS12" />
    <zosconnect_authData id="cicsCredentials"
     user="${CICS_USER}" password="${CICS_PASSWORD}" />
</server>
```

Contents of /mnt/c/z/openApi3/tls/cicsTLSMutual.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<server description="IPIC connection to CICS">
    <featureManager>
        <feature>zosconnect:cics-1.0</feature>
    </featureManager>
    <zosconnect_cicsIpicConnection id="cicsConn"
        host="${CICS_HOST}"
        port="${CICSSSL_PORT}"
        zosConnectNetworkid="DESIGNER"
        zosConnectApplid="DESIGNER"
        sslCertsRef="cicsSSLSettings" />
<ssl id="cicsSSLSettings"
     keyStoreRef= "cicsKeyStore"
     trustStoreRef= "cicsTrustStore" />
   <keyStore id= "cicsTrustStore"
     location="/output/resources/security/cicsTrustStore.jks"
     password="${CICSTRUSTSTORE_PASSWORD}" type="PKCS12" />
   <keyStore id= "cicsKeyStore"
     location="/output/resources/security/CICSUSR1.P12"
     password="${CICSKEYSTORE_PASSWORD}" type="PKCS12" />
</server>
```

Contents of /mnt/c/z/openApi3/tls/db2TLSServer.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<server description="DB2 SSL">
    <featureManager>
        <feature>zosconnect:db2-1.0</feature>
    </featureManager>
    <zosconnect_credential id="commonCredentials" />
      user="${DB2_USERNAME}" password="${DB2_PASSWORD}" />
    <zosconnect_db2Connection id="db2ConnTLS"
      host="${DB2_HOST}" port="${DB2SSL_PORT}"
      credentialRef="commonCredentials"
      sslCertsRef="db2SSLSettings" />
     <ssl id="db2SSLSettings"
      keyStoreRef="db2TrustStore"
      trustStoreRef="db2TrustStore" />
    <keyStore id="db2TrustStore"
     location="/output/resources/security/db2TrustStore.jks"
     password="${DB2TRUSTSTORE_PASSWORD}" type="PKCS12" />

</server>
```

Useful URLs for z/OS Connect Designer Container

- **Accessing the Designer**
  http://localhost:9082/zosConnect/designer

- **Review the Container's Liberty configuration**
  https://localhost:9445/ibm/api/config

- **Access the Container's API Explorer**
  https://localhost:9445/api/explorer/

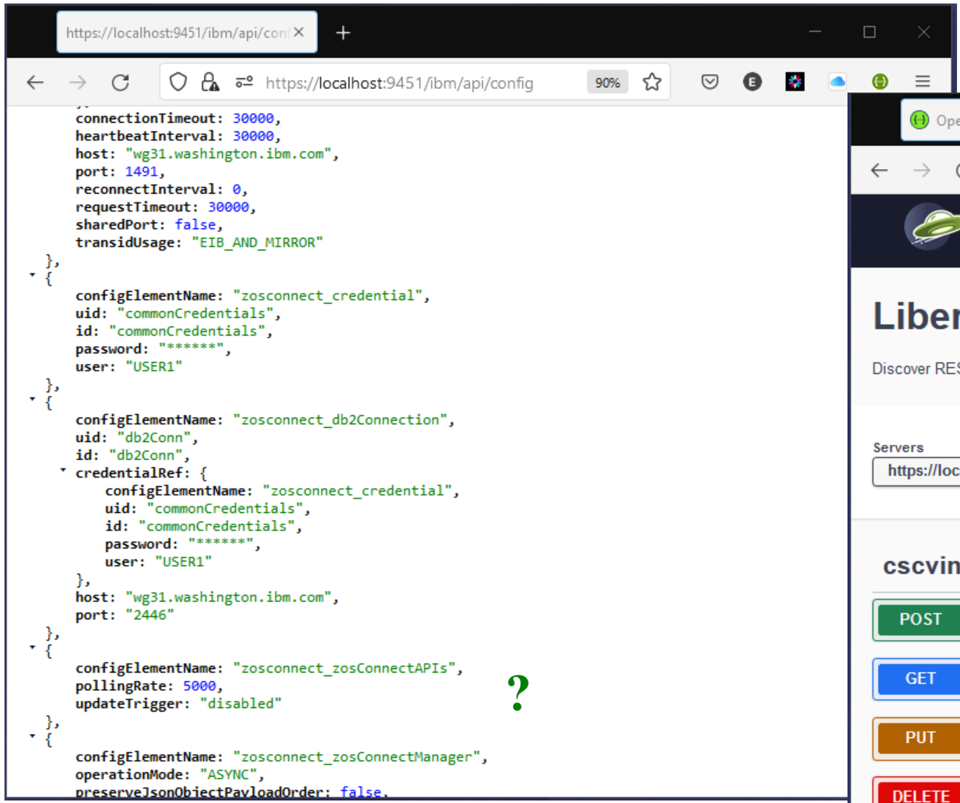
- **Access the Container's logs directory**
  https://localhost:9445/logs

- **Access and/or download the Web Archive (WAR) file**
  https://localhost:9445/dropins

- **Convert a Swagger (Open API 2) document to Open API 3**
  https://mermade.org.uk/openapi-converter

- **Validate and/or view an Open API 3 document**
  https://jsonformatter.org/yaml-viewer

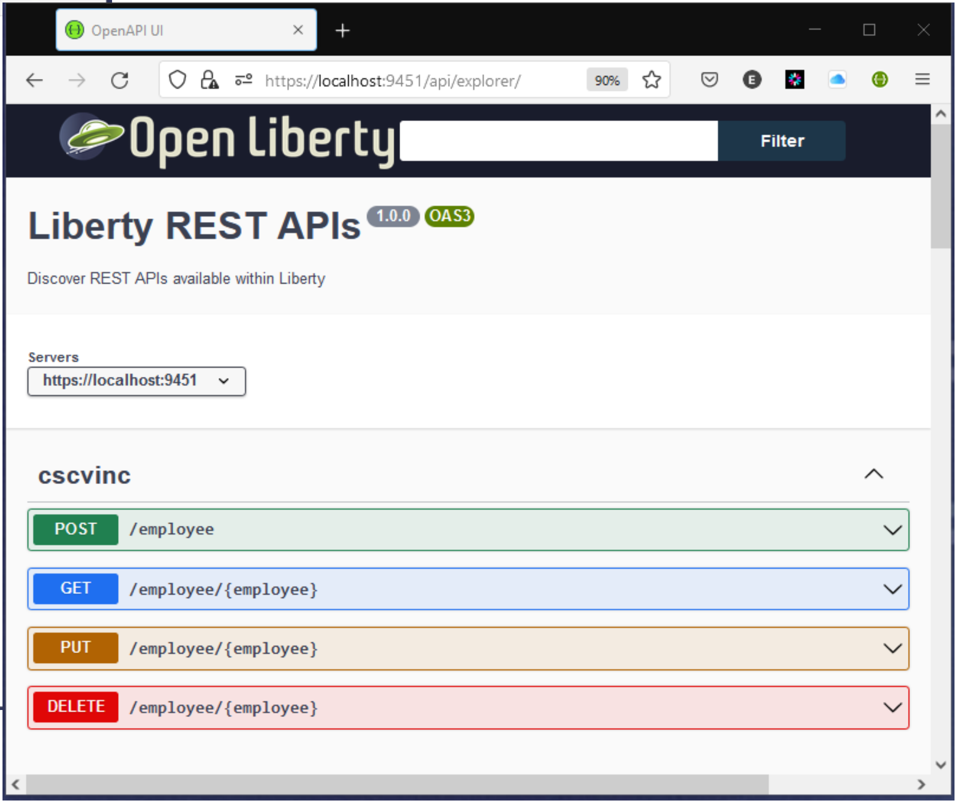
Remember the container can be accessed directly (w/o using Designer)



https://localhost:9451/ibm/api/config

```
        connectionTimeout: 30000,
        heartbeatInterval: 30000,
        host: "wg31.washington.ibm.com",
        port: 1491,
        reconnectInterval: 0,
        requestTimeout: 30000,
        sharedPort: false,
        transidUsage: "EIB_AND_MIRROR"
    },
  ▾ {
        configElementName: "zosconnect_credential",
        uid: "commonCredentials",
        id: "commonCredentials",
        password: "******",
        user: "USER1"
    },
  ▾ {
        configElementName: "zosconnect_db2Connection",
        uid: "db2Conn",
        id: "db2Conn",
      ▾ credentialRef: {
            configElementName: "zosconnect_credential",
            uid: "commonCredentials",
            id: "commonCredentials",
            password: "******",
            user: "USER1"
        },
        host: "wg31.washington.ibm.com",
        port: "2446"
    },
  ▾ {
        configElementName: "zosconnect_zosConnectAPIs",
        pollingRate: 5000,
        updateTrigger: "disabled"
    },
  ▾ {
        configElementName: "zosconnect_zosConnectManager",
        operationMode: "ASYNC",
        preserveJsonObjectPayloadOrder: false,
```

?



https://localhost:9451/api/explorer/

Liberty REST APIs 1.0.0 OAS3

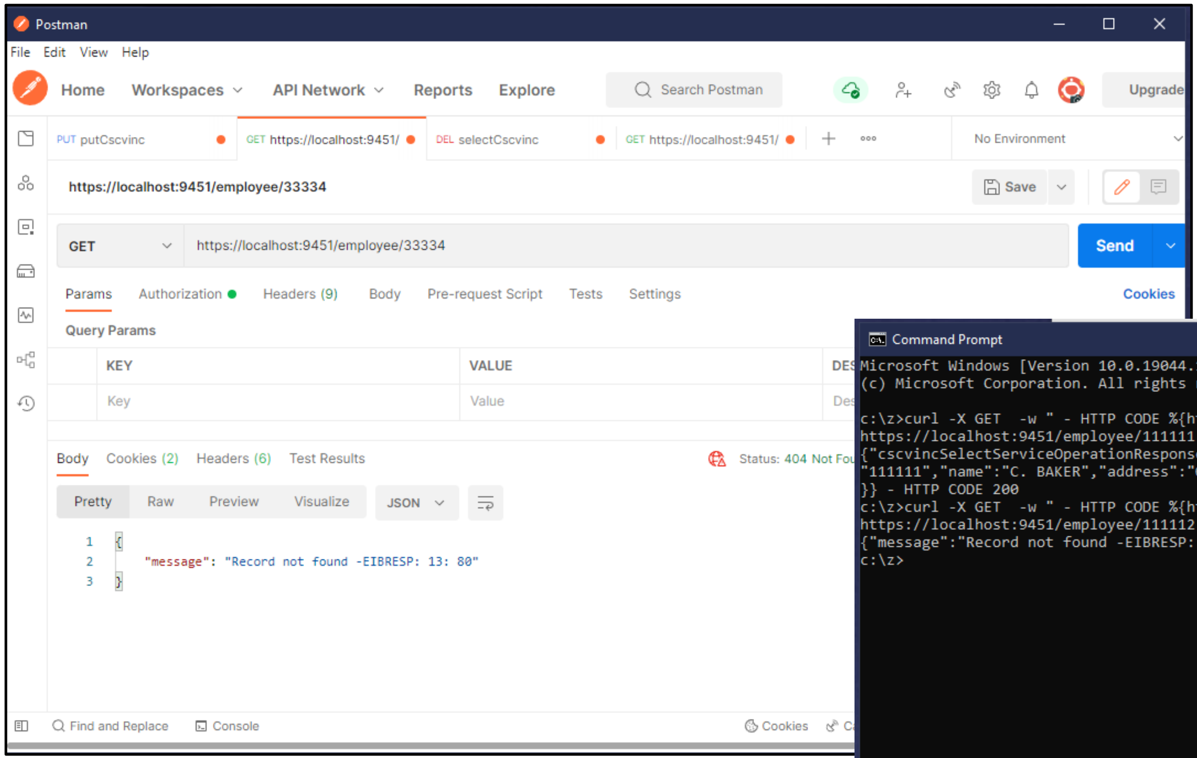
Discover REST APIs available within Liberty

Servers
https://localhost:9451

cscvinc

POST	/employee
GET	/employee/{employee}
PUT	/employee/{employee}
DELETE	/employee/{employee}

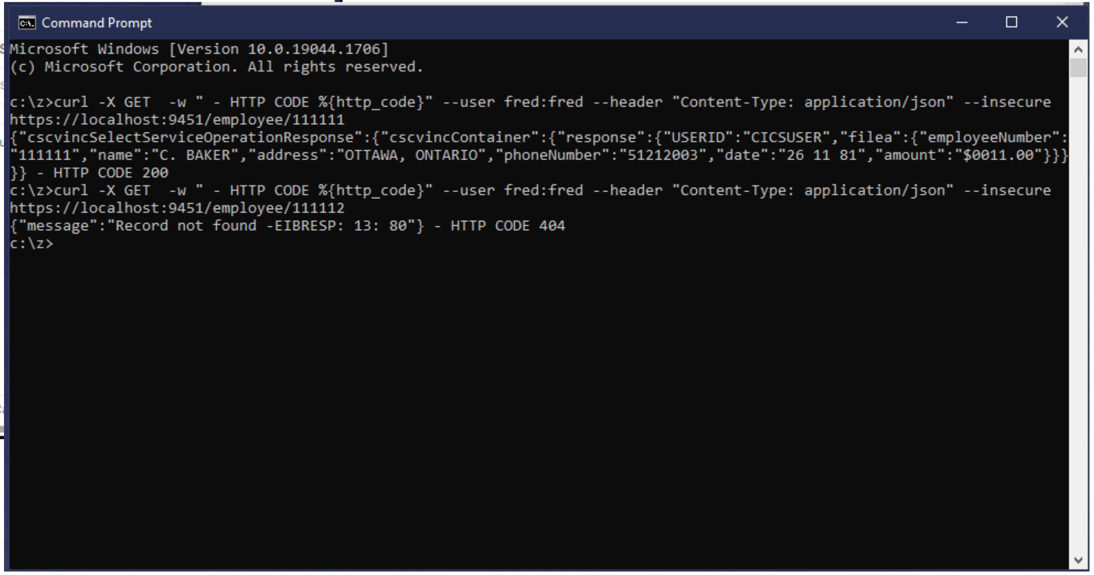
The container’s API can be invoked using Postman or curl


Postman
File Edit View Help
Home Workspaces API Network Reports Explore
Search Postman
Upgrade
PUT putCscvinc
GET https://localhost:9451/
DEL selectCscvinc
GET https://localhost:9451/
No Environment
Save
Send
Params Authorization Headers (9) Body Pre-request Script Tests Settings
Cookies
Query Params
KEY VALUE
Body Cookies (2) Headers (6) Test Results
Status: 404 Not Fou
Pretty Raw Preview Visualize JSON
Find and Replace Console Cookies

https://localhost:9451/employee/33334

GET https://localhost:9451/employee/33334

```
{
    "message": "Record not found -EIBRESP: 13: 80"
}
```



```
Microsoft Windows [Version 10.0.19044.1706]
(c) Microsoft Corporation. All rights reserved.

c:\z>curl -X GET  -w " - HTTP CODE %{http_code}" --user fred:fred --header "Content-Type: application/json" --insecure
https://localhost:9451/employee/111111
{"cscvincSelectServiceOperationResponse":{"cscvincContainer":{"response":{"USERID":"CICSUSER","filea":{"employeeNumber":
"111111","name":"C. BAKER","address":"OTTAWA, ONTARIO","phoneNumber":"51212003","date":"26 11 81","amount":"$0011.00"}}}
}} - HTTP CODE 200
c:\z>curl -X GET  -w " - HTTP CODE %{http_code}" --user fred:fred --header "Content-Type: application/json" --insecure
https://localhost:9451/employee/111112
{"message":"Record not found -EIBRESP: 13: 80"} - HTTP CODE 404
c:\z>
```

mitchj@us.ibm.com

Interesting Liberty defaults for the Designer

```
********************************************************************************
product = WebSphere Application Server 22.0.0.3, z/OS Connect 03.00.57 (wlp-1.0.62.cl220320220302-1100)
wlp.install.dir = /opt/ibm/wlp/
server.output.dir = /opt/ibm/wlp/output/defaultServer/
java.home = /opt/ibm/java/jre
java.version = 1.8.0_321
java.runtime = Java(TM) SE Runtime Environment (8.0.7.6 - pxa6480sr7fp6-20220330_01(SR7 FP6))
os = Linux (5.10.102.1-microsoft-standard-WSL2; amd64) (en_US)
process = 1@192.168.112.2
Classpath = /opt/ibm/wlp/bin/tools/ws-server.jar:/opt/ibm/wlp/bin/tools/ws-javaagent.jar:/opt/ibm/wlp/bin/tools/ws-javaagent.jar
Java Library path = /opt/ibm/java/jre/lib/amd64/compressedrefs:/opt/ibm/java/jre/lib/amd64:/usr/lib64:/usr/lib
********************************************************************************
[6/1/22 18:11:29:925 UTC] 00000001 com.ibm.ws.kernel.launch.internal.FrameworkManager           A CWWKE0001I: The server defaultServer has been
launched.
[6/1/22 18:11:30:827 UTC] 00000027 com.ibm.ws.config.xml.internal.ServerXMLConfiguration        A CWWKG0093A: Processing configuration drop-ins
resource: /opt/ibm/wlp/usr/servers/defaultServer/configDropins/defaults/keystore.xml
[6/1/22 18:11:30:851 UTC] 00000027 com.ibm.ws.config.xml.internal.ServerXMLConfiguration        A CWWKG0093A: Processing configuration drop-ins
resource: /opt/ibm/wlp/usr/servers/defaultServer/configDropins/overrides/http-ssl-endpoint.xml
[6/1/22 18:11:30:853 UTC] 00000027 com.ibm.ws.config.xml.internal.ServerXMLConfiguration        A CWWKG0093A: Processing configuration drop-ins
resource: /opt/ibm/wlp/usr/servers/defaultServer/configDropins/overrides/tls.xml
[6/1/22 18:11:31:051 UTC] 00000001 com.ibm.ws.kernel.launch.internal.FrameworkManager           I CWWKE0002I: The kernel started after 1.287 seconds
[6/1/22 18:11:31:272 UTC] 00000033 com.ibm.ws.kernel.feature.internal.FeatureManager            I CWWKF0007I: Feature update started.
[6/1/22 18:11:34:054 UTC] 00000027 g.apache.cxf.cxf.core.3.2:1.0.62.cl220320220302-1100(id=90)] I Aries Blueprint packages not available. So
namespaces will not be registered
[6/1/22 18:11:34:118 UTC] 00000026 com.ibm.ws.security.ready.internal.SecurityReadyServiceImpl  I CWWKS0007I: The security service is starting...
[6/1/22 18:11:34:288 UTC] 00000027 com.ibm.ws.app.manager.internal.monitor.DropinMonitor        A CWWKZ0058I: Monitoring dropins for applications.
[6/1/22 18:11:34:788 UTC] 00000027 com.ibm.ws.cache.ServerCache                                 I DYNA1001I: WebSphere Dynamic Cache instance named
baseCache initialized successfully.
[6/1/22 18:11:34:794 UTC] 00000027 com.ibm.ws.cache.ServerCache                                 I DYNA1071I: The cache provider default is being
used.
[6/1/22 18:11:34:796 UTC] 00000027 com.ibm.ws.cache.CacheServiceImpl                            I DYNA1056I: Dynamic Cache (object cache)
initialized successfully.
[6/1/22 18:11:35:004 UTC] 00000026 ibm.ws.security.authentication.internal.jaas.JAASServiceImpl I CWWKS1123I: The collective authentication plugin
with class name NullCollectiveAuthenticationPlugin has been activated.
[6/1/22 18:11:35:425 UTC] 0000004c com.ibm.ws.security.token.ltpa.internal.LTPAKeyCreateTask    I CWWKS4105I: LTPA configuration is ready after
0.455 seconds.
[6/1/22 18:11:35:594 UTC] 00000026 com.ibm.ws.session.WASSessionCore                            I SESN8501I: The session manager did not find a
persistent storage location; HttpSession objects will be stored in the local application server's memory.
[6/1/22 18:11:35:810 UTC] 00000026 .microprofile.metrics.internal.monitor.MonitorMetricsHandler I CWPMI2003I: Monitoring metrics can be retrieved
through mpMetrics.
[6/1/22 18:11:35:911 UTC] 00000026 com.ibm.ws.security.audit.file.AuditFileHandler              I CWWKS5804I: The audit file handler service is
starting.
[6/1/22 18:11:35:916 UTC] 00000026 com.ibm.ws.security.audit.source.AuditServiceImpl            I CWWKS5850I: The audit service is starting.
[6/1/22 18:11:35:924 UTC] 00000026 com.ibm.ws.security.audit.source.AuditServiceImpl            I CWWKS5851I: The audit service is ready.
[6/1/22 18:11:35:930 UTC] 00000026 com.ibm.ws.security.audit.file.AuditFileHandler              I CWWKS5805I: The audit file handler service is
ready.
[6/1/22 18:11:36:195 UTC] 0000002f com.ibm.ws.ssl.config.WSKeyStore                             I Successfully loaded default keystore:
/opt/ibm/wlp/output/defaultServer/resources/security/key.p12 of type: PKCS12
```

/opt/ibm/wlp/usr/servers/defaultServer/server.xml

Default server XML configuration files

defaults/keystore.xml

```
<server description="Default Server">
    <keyStore id="defaultKeyStore" password="IRawdFa2HxS7cte9T8M1MOvpKQzqtYnQRZwKGffp3t0=" />
```

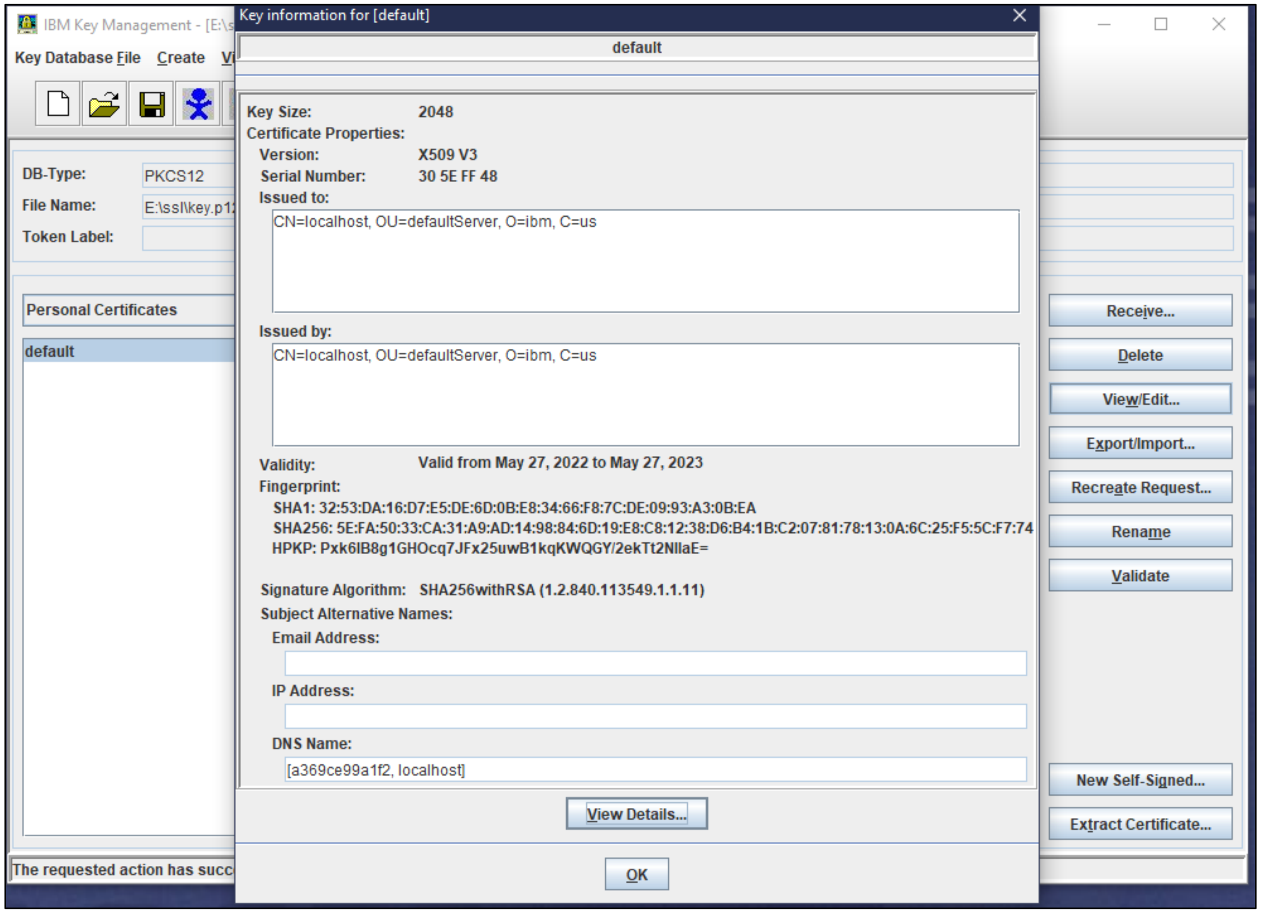
overrides/http-ssl-endpoint.xml

```
<server>
    <variable name="HTTP_PORT" defaultValue="9080" />
    <variable name="HTTPS_PORT" defaultValue="9443" />
    <httpEndpoint id="defaultHttpEndpoint" host="*" httpsPort="${HTTPS_PORT}" httpPort="${HTTP_PORT}" />
</server>
```

overrides/tls.xml

```
<server description="Default Server"
<featureManager>
        <feature>transportSecurity-1.0</features>
</featureManager>
```

Be wary of the container's default self-signed personal certificate


IBM Key Management - [E:\s

Key Database File Create Vi

DB-Type: PKCS12
File Name: E:\ssl\key.p1
Token Label:

Personal Certificates

default

The requested action has succ

Key information for [default]

default

Key Size: 2048
Certificate Properties:
Version: X509 V3
Serial Number: 30 5E FF 48
Issued to:
CN=localhost, OU=defaultServer, O=ibm, C=us
Issued by:
CN=localhost, OU=defaultServer, O=ibm, C=us
Validity: Valid from May 27, 2022 to May 27, 2023
Fingerprint:
SHA1: 32:53:DA:16:D7:E5:DE:6D:0B:E8:34:66:F8:7C:DE:09:93:A3:0B:EA
SHA256: 5E:FA:50:33:CA:31:A9:AD:14:98:84:6D:19:E8:C8:12:38:D6:B4:1B:C2:07:81:78:13:0A:6C:25:F5:5C:F7:74
HPKP: Pxk6IB8g1GHOcq7JFx25uwB1kqKWQGY/2ekTt2NllaE=

Signature Algorithm: SHA256withRSA (1.2.840.113549.1.1.11)
Subject Alternative Names:
Email Address:
IP Address:
DNS Name:
[a369ce99a1f2, localhost]

Note that the certificate expires after 1 year.

In Linux
cp ../sandbox/certs/key.p12 /mnt/c/ssl
In Windows, use ikeyman to open the keystore

Key directories in the Designer container

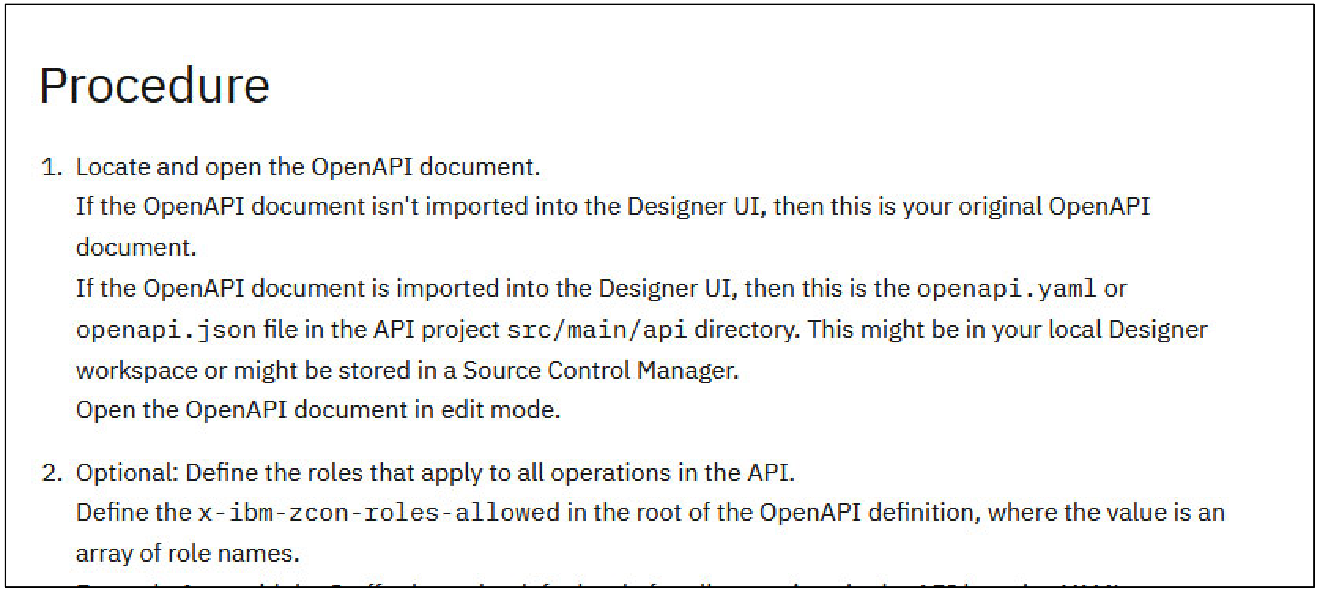
- **Key container directories**

  /workspace/project
  /opt/ibm/wlp/usr/servers/defaultServer
  /output/resources/security
  /opt/ibm/wlp/usr/servers/defaultServer/configDropins/default
  /opt/ibm/wlp/usr/servers/defaultServer/configDropins/

  /config -> /opt/ibm/wlp/usr/servers/defaultServer
  /output -> /opt/ibm/wlp/output/defaultServer
  ${server.output.dir}   ->   /output/ibm/wlp/output/defaultServer

z/OS Server Issues and Considerations – Adding Roles

https://www.ibm.com/docs/en/zos-connect/zos-connect/3.0?topic=authorization-how-define-roles



Procedure

1. Locate and open the OpenAPI document.
   If the OpenAPI document isn't imported into the Designer UI, then this is your original OpenAPI document.
   If the OpenAPI document is imported into the Designer UI, then this is the `openapi.yaml` or `openapi.json` file in the API project `src/main/api` directory. This might be in your local Designer workspace or might be stored in a Source Control Manager.
   Open the OpenAPI document in edit mode.
2. Optional: Define the roles that apply to all operations in the API.
   Define the `x-ibm-zcon-roles-allowed` in the root of the OpenAPI definition, where the value is an array of role names.

Other useful commands

- List the installed images
  docker images

```
REPOSITORY                            TAG       IMAGE ID       CREATED          SIZE
icr.io/zosconnect/ibm-zcon-designer   3.0.57    386f4ac8cbd0   25 hours ago     1.16GB
icr.io/zosconnect/ibm-zcon-designer   3.0.56    cf167f4230b5   6 weeks ago      1.57GB
icr.io/zosconnect/ibm-zcon-designer   3.0.55    be9c9101f533   2 months ago     1.52GB
hello-world                           latest    feb5d9fea6a5   8 months ago     13.3kB
```

- Remove an installed image
  docker rmi icr.io/zosconnect/ibm-zcon-designer:3.0.56

- Invoking Linux commands in the container
  docker exec -it sandbox_zosConnect_1 ls -l /templates/gradleLibs/
  docker exec -it sandbox_zosConnect_1 ls -lR /templates/gradleLibs/com/ibm/zosconnect
  docker exec -it sandbox_zosConnect_1 cd /workspace/project && gradle build --debug

- Install the Podman podman-compose command
  pip install podman-compose

Other useful container related commands

- Display the details of a container
  ```
  docker container inspect sandbox_zosConnect-1
  ```
- Create a copy of a container
  ```
  docker commit sandbox_zosConnect_1    sandbox_zosconnect_1_Next
  ```
- Copy the configuration XML override file from Linux into the container
  ```
  docker cp /mnt/c/z/openApi3/xml/. sandbox_zosConnect_1:/config/configDropins/overrides/
  ```
- Copy the war files and from the container
  ```
  docker cp sandbox_zosConnect_1:/workspace/project/build/libs/api.war
  /mnt/c/z/openApi3/wars/cscvinc.war
  ```
- Copy the configuration XML files from the container into Linux
  ```
  docker cp /mnt/c/z/openApi3/xml/.   sandbox_zosConnect_1:/config/configDropins/overrides
  ```
- Display a docker container's IP information
  ```
  docker inspect -f '{{range.NetworkSettings.Networks}}{{.IPAddress}}{{end}}'
  db2api_zosConnect_1
  ```

```
192.168.176.2
```

Contents of C:/z/openApi3/bin/createPodmanContainer

```
echo on
[ -z "$2" ] && HTTP_port=9080 || HTTP_port=$2
[ -z "$3" ] && HTTPS_port=9443 || HTTPS_port=$3
echo creating container "$1"_zosConnect_1 with HTTP_port="$HTTP_port" and
HTTPS_port="$HTTPS_port"
mkdir $containerHome/podman/"$1"
cd $containerHome/podman/"$1"
mkdir certs
mkdir logs
mkdir -p project/src/main/liberty/config
cp /mnt/c/z/openApi3/xml/* project/src/main/liberty/config
cp /mnt/c/z/openApi3/yaml/docker-compose.yaml .
sed -i "s/9080:9080/$HTTP_port:9080/" docker-compose.yaml
sed -i "s/9443:9443/$HTTPS_port:9443/" docker-compose.yaml
podman-compose up -d
podman cp /mnt/c/z/openApi3/xml/. "$1"_zosConnect_1:/config/configDropins/overrides
```

Used to create a new container, ***`createPodmanContainer   containerName`***

Contents of C:/z/openApi3/bin/refreshPodmanContainer

```
echo refreshing container "$1"_zosConnect_1
podman stop "$1"_zosConnect_1
podman container rm "$1"_zosConnect_1
cd $containerHome/podman/"$1"
rm -r project/*
mkdir -p project/src/main/liberty/config
cp /mnt/c/z/openApi3/xml/* project/src/main/liberty/config
podman-compose up -d
podman cp /mnt/c/z/openApi3/xml/. "$1"_zosConnect_1:/config/configDropins/overrides
```

Used to refresh an existing container, ***refreshPodmanContainer*** ***containerName***

Contents of C:/z/openApi3/bin/dockerBash

Used to start a Linux shell within a Docker container, ***dockerBash*** ***containerName***

```
docker exec -it "$1"_zosConnect_1 bash
```

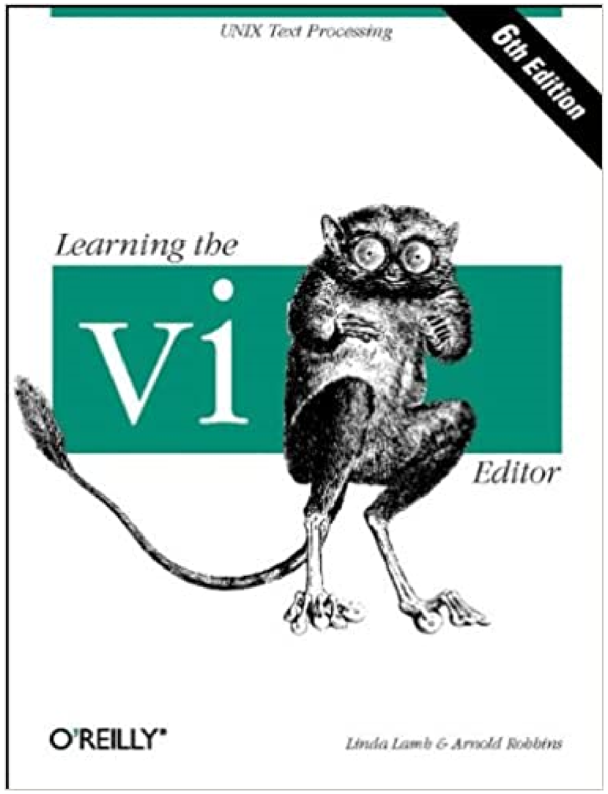
Contents of C:/z/openApi3/bin/podmanBash

Used to start a Linux shell within a Podman container, ***podmanBash*** ***containerName***

```
podman exec -it "$1"_zosConnect_1 bash
```

Visual Editor (vi) Hints and Tips

If you are going need to edit Linux files, I highly recommend this book for learning how to use the vi editor.



Learning the vi Editor (Nutshell Handbooks) Sixth Edition