**OpenShift on IBM Z**

**Lab 001**

**Exploring the OpenShift GUI**

Lab Version Date: April 14, 2020

Table of Contents

[Lab 001 – Exploring the OpenShift GUI 3](#_Toc37771159)

[Connect to OCP and Authenticate 3](#_Toc37771160)

[The Administrator Perspective 5](#_Toc37771161)

[The Developer Perspective 7](#_Toc37771162)

[Deploy from Container Image 9](#_Toc37771163)

[View Workload from Administrator Perspective 15](#_Toc37771164)

# Lab 001 – Exploring the OpenShift GUI

## Connect to OCP and Authenticate

* On your desktop, open a browser (Firefox or Chrome)
* In the address bar, navigate to the following URL:

**<https://console-openshift-console.apps.atsocpd1.dmz/>**

* + You might see a security challenge if the cluster has not yet been accessed from your workstation. This is due to default SSL certificate being “self-signed” and not recognized by the browser. Accept the challenge to continue:

|  |  |
| --- | --- |
| *Chrome* | *Firefox* |
| 1. Click on "Advanced" 2. Click on "Proceed to ... | 1. Click on "Advanced" 2. Click on "Add exception" |

You should now see the OpenShift login page:

A screenshot of a cell phone

Description automatically generated

* Log into the cluster using the following credentials
  + Username: **userNN** (where NN is your user number)
  + Password: **p@ssw0rd**

Take a moment to notice the following elements of the user interface:

The **Menu** button. This provides a drop-down menu of other options.



The **User Profile** button for other options related to logging in/out.

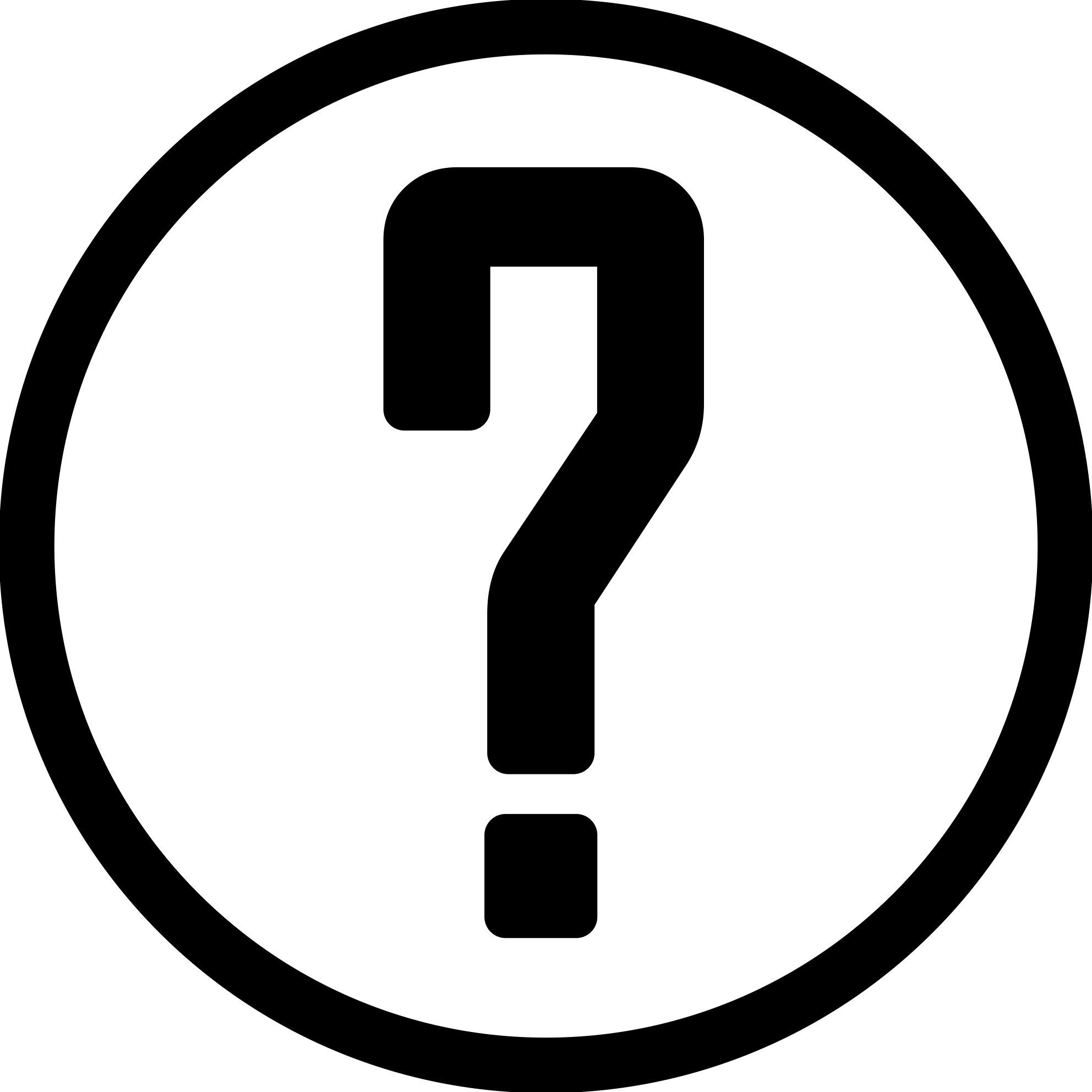
The **Import** button. This opens a window to enter YAML or JSON to create various types of new objects (deployments, services, operators, etc.)

The **Help** button for documentation related to your current location in OpenShift and cluster information.

These four buttons display on each page of the OpenShift console.

By default, the menu on the left side of the page should be activated and displaying the menu. We’ll look more closely at this in the next section.

* Click on the **Import button (A close up of a logo

  Description automatically generated)**. This brings up a new window where you can enter YAML or JSON code to create objects in the cluster. Objects can be anything like Pods, Services, Deployment Configs, Secrets, etc., so if someone has one of these objects in another cluster, this button can be used to easily bring it into this cluster.
* Click on the **Help button ()**. This brings up a drop-down menu of helpful links such as the official OpenShift documentation, links to download command line tools, and information about the cluster. Also, this menu can be customized so you can add your own links if desired.
* Click on the **User Profile button** (displaying your user name). This brings up a drop-down menu to let you log out, or copy a login command that can be pasted into a terminal to easily log into the cluster from a command line.

## The Administrator Perspective

If not already visible, click on the **Menu Button** to see the drop-down menu of further options and check that the **Administrator** perspective is selected.   
A screenshot of a cell phone

Description automatically generated

The **Developer / Administrator** **toggle**. This lets you flip between which perspective you want to use.

**Home**: Provides overview of projects, resources, and events in the scope of your credentials.

**Operators**: Lets you see what operators are installed for your project(s).

**Workloads**: Expands to provide access to all deployed pods, deployments, secrets, etc.

**Networking**: Provides all services, routes, and ingress required for external access to the cluster.

**Storage**: Access to Persistent Volume Claims to request a portion of the cluster’s Persistent Volume.

**Builds**: View and create Builds – the process of transforming input parameters into resulting objects.

**Administration**: View and edit cluster settings, depending on logged in credentials.

**Service Catalog**: If enabled, view all provisioned services (such as z/OS Cloud Broker).

The **Administrator** perspective is the default view for the OpenShift console. This perspective provides visibility into options related to cluster administration, as well as a broader view of the projects associated with the currently logged-in user. Let’s create our first project now.

By default, the OpenShift console will open the **Projects** page. If you have clicked away from this page, under the **Menu button, click Home 🡪 Projects**.  
  
A screenshot of a cell phone

Description automatically generated

The rest of the page is populated by your projects. One should have been created for you, named userNN-project (where NN is your user number).

* **Click on the userNN-project hyperlink**. You can use the search bar to find it more easily.
* If no project exists, create one for yourself:
  + Name: **userNN-project**
  + Display Name: **userNN Project 001**
  + Description: **userNN’s Project for Lab 001**
* **Click Create**

You’ll now see the overview dashboard for your new cluster.

* **Scroll down the Overview tab**. This displays information about your project, such as CPU and memory usage, any alerts or crashlooping pods, and more. You won’t see much information yet, as no workloads are running in this project.
* **Click on the YAML tab** to the right of Overview. This is the code that was used to generate this project. It can be directly edited here if changes are desired.
* **Click on the Workloads tab** to the right of YAML.This page displays all of the workloads in your project, so it’s empty for now.

Workloads are typically created by developers, so let’s swap over to the developer perspective. We’ll come back to the administrator perspective later in this lab.

* Under the Menu, click the **Administrator dropdown, and select Developer**.

## The Developer Perspective

A screenshot of a cell phone

Description automatically generated

The **Developer / Administrator toggle**. This lets you flip between which perspective you want to use.

**+Add**: Clicking on this will open a prompt letting you add a workload to your current project.

**Topology**: This will show all of the deployed workloads in the currently selected project.

**Builds**: This will let you view or create Build Configurations in the currently selected project.

**Advanced**: Opens a drop-down menu of more options, including viewing all of your projects and streaming events/errors.

The **Developer** perspective provides views and workflows specific to developer use cases, while hiding many of the cluster management related options typically used by administrators. This perspective provides developers with a streamlined view of the options they typically use.

Switching to the Developer perspective takes you to the **Topology** view. If no workloads are deployed in the selected project, options to deploy a workload are displayed.

The project we viewed from the Administrator perspective.

No specific application is selected. You can create one when deploying a workload.

A screenshot of a cell phone

Description automatically generated

Multiple deployment methods available from the web console.

No workloads were found, so you were redirected from the Topology page.

There are multiple methods of deploying workloads from the OpenShift web browser.

1. **From Git**: Use this option to import an existing codebase in a Git repository to create, build, and deploy an application.
2. **Container Image**: Use existing images from an image stream or registry to deploy it.
3. **From Catalog**: Explore the Developer Catalog to select the required applications, services, or source to image builders and add it to your project.
4. **From Dockerfile**: Import a dockerfile from your Git repository to build and deploy an application.
5. **YAML**: Use the editor to add YAML or JSON definitions to create and modify resources.
6. **Add Database**: Filters the Developer Catalog to display only the databases it contains.

Next, let’s deploy a workload from a Container Image.

## Deploy from Container Image

In this section, we will be building and deploying a container using a container image. A container image holds a set of software that is ready to run, while a container is a running instance of a container image. Images can be hosted in registries, such as the Red Hat registry, Docker Hub, or a private registry of your own.

* **Click on the Container Image tile**

This brings up a new page which prompts you for an image name and further configurable parameters further down the page. Only the image name is required, and the rest will automatically populate for you.

* In the search bar for Image Name, type **open-liberty** and press enter or the search button.

A close up of a logo

Description automatically generated

* Notice that the remaining fields on this screen automatically populate

A screenshot of a cell phone

Description automatically generated

* **Click on the links** for Routing, Deployment Configuration, etc. at the bottom of the page to see the available advanced options. We won’t be changing any of these.

A close up of a logo

Description automatically generated

* **Click the Create button**

You will now be taken to the topology view, where you will see an icon for your new Open Liberty application.

* **Click on the icon** for the open-liberty application. This will bring up a window on the right-hand side of the screen with information about your deployment.

Depending on how quickly you clicked on this icon, it will display either *1 pod*, or *0 scaling to 1*. Either way, it will become *1 pod* after a few seconds.

A screenshot of a cell phone

Description automatically generated

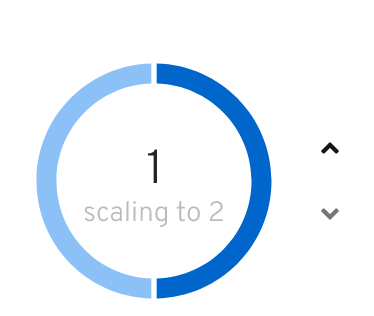
Here you’ll see information about your deployment. You’ll notice that many of the fields such as Labels, Update Strategy, and Timeout have been populated with default values. These can be modified.

* **Click on the Actions dropdown**.

A screenshot of a cell phone

Description automatically generated  
Many of the options can be modified from this menu, along with other tasks such as starting or pausing a rollout, or deleting the deployment configuration.

* **Click on the up arrow** next to the blue circle.

  
  
This will scale your application from one pod to two pods.   
A close up of a logo

Description automatically generated

This is a simple demonstration of horizontal scaling with Kubernetes. You now have two instances of your pod running in the OpenShift cluster. Traffic to the Liberty server will now be distributed to all of the pods, and if for some reason a pod is lost, that traffic will be redistributed to the remaining pods until a Kubernetes spins up another. If a whole worker node is lost, Kubernetes will move the pods to different worker nodes.

OpenShift and Kubernetes also support autoscaling of pods based on CPU/memory/etc. consumption per node, but that is outside the scope of this lab.

* **Click on the Resources tab**.

A screenshot of a cell phone

Description automatically generated

Notice the two pods associated with your Liberty application. On this page, you’ll see more information about your pods, any build configurations currently running (there shouldn’t be any), and the services/ports associated with the pod.

* **Click on the route address** beginning with http://

A close up of a sign

Description automatically generated

If you see the page above, your Open Liberty application is up and running! You have just deployed an Open Liberty container from a container image pulled from Docker Hub into OpenShift running on an IBM Z server.

Let’s go back to the Administrator perspective to see an overview of our project, now with a workload running.

## View Workload from Administrator Perspective

* From the menu, **switch back to the Administrator perspective**.
* **Click on your project**.

The overview page now displays data about the CPU and Memory Usage. Recall that this was empty earlier.

A screenshot of a cell phone

Description automatically generated

* **Click Events** under the Home dropdown.

A screenshot of a cell phone

Description automatically generated

This page is populated with all of the events associated with your project, including errors, container creation messages, pod scaling and deletion, and much more. You can filter by type, category, or by searching for keywords.

A screenshot of a cell phone

Description automatically generated

Feel free to click through a few more pages from the menu. You’ll notice a few of them have objects created as a part of the Open Liberty application, such as Workloads 🡪 Pods, Networking 🡪 Services and Routes, Builds 🡪 Image Streams. These were all created as part of the container image package.

* **Click on the Projects tab** again, and select your project.
* **Click on the workloads tab** to open the application overview on the right-hand side of the page.

* Click on the open-libery workload, click the Actions dropdown, and select **Delete Deployment Config**.

A screenshot of a cell phone

Description automatically generated

* Similarly, navigate to Networking 🡪 Services, click on the three dots to the far right, and **delete your service**.

A screenshot of a cell phone

Description automatically generated

* Navigate to Networking 🡪 Routes, and **delete your route**.

A screenshot of a cell phone

Description automatically generated

* Navigate to Builds 🡪 Image Streams and **delete your image stream**.

A screenshot of a cell phone

Description automatically generated

**End of Lab**