Course Exercises Guide

IBM Cloud Pak for Automation, Installation and Administration

Course code WB318 / ZB318 ERC 1.0

IBM Training

October 2019 edition

Notices

This information was developed for products and services offered in the US.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
United States of America

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at “Copyright and trademark information” at www.ibm.com/legal/copytrade.shtml.

Contents

Trademarks . . . . . **iv**

Exercises description . . . . . **v**

Exercise 1. Deploying the IBM Operational Decision Manager (ODM) container **1-1**
Part 1: Work with the development environment 1-5
Part 2: Load the ODM docker images for Cloud Pak for Automation 1-14
Part 3: Prepare the required ODM database 1-18
Part 4: Deploy the ODM container 1-20
Part 5: Verify successful ODM deployment 1-22
Part 6: Troubleshoot failed deployment. 1-24
Part 7: Access the running ODM containers 1-26

Exercise 2. Deploying the IBM FileNet P8 Content Platform Engine (CPE) container **2-1**
Part 1: Work with the development environment 2-5
Part 2: Create the Persistent Volumes and Persistent Volume Claims that are required by Content Platform Engine 2-14
Part 3: Prepare the database required by Content Platform Engine 2-18
Part 4: Prepare LDAP required by Content Platform Engine 2-22
Part 5: Load the Content Platform Engine docker images for Cloud Pak for Automation 2-25
Part 6: Deploy the Content Platform Engine container 2-31
Part 7: Verify successful Content Platform Engine deployment 2-36
Part 8: Troubleshoot failed deployment. 2-38
Part 9: Access the running Content Platform Engine container 2-40

Exercise 3. Administering the IBM Cloud Pak for Automation containers **3-1**
Part 1: Explore the OpenShift web console for container management 3-3
Part 2: Examine the available open source monitoring options 3-27
Part 3: Scale an application 3-40
Part 4: Monitor applications with probes 3-45

Appendix A. Installing Red Hat OpenShift Container Platform V3.1.1 to create the lab environment **A-1**
Part 1: Configure Environment for installing Red Hat OpenShift Container Platform 3.1.1 . . . . . A-1
Part 2: Install Red Hat OpenShift A-4

Trademarks

The reader should recognize that the following terms, which appear in the content of this training document, are official trademarks of IBM or other companies:

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide.

The following are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide:

Bluemix®	Cognitive Era™	Cognos®
SPSS®	Watson Avatar®	Worklight®

Windows is a trademark of Microsoft Corporation in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other product and service names might be trademarks of IBM or other companies.

Exercises description

This course includes the following exercises:

- Deploying the IBM Operational Decision Manager (ODM) container
- Deploying the IBM FileNet P8 Content Platform Engine (CPE) container
- Administering the IBM Cloud Pak for Automation containers

In the exercise instructions, you can check off the line before each step as you complete it to track your progress.

Important

The exercises in this course use a set of lab files that might include scripts, applications, files, solution files, PI files, and others. The course lab files can be found in the following directory:

`/root/labfiles`, for Linux

The exercises point you to the lab files as you need them.

User accounts

	Type	User ID	Password
RHEL	Operating system	root	passw0rd
OpenShift	Container Platform	admin	passw0rd

Important

Online course material updates might exist for this course. To check for updates, see the Instructor wiki at http://ibm.biz/CloudEduCourses.

Attention

Read this section if you want to do the deployment exercises in a different order or skip a deployment

The order of the labs in this book is IBM Operational Decision Manager (ODM) container deployment in Exercise 1, followed by the IBM Content Platform Engine (CPE) container deployment in Exercise 2. Then, in Exercise 3 you do some container management using Red Hat OpenShift Container Platform.

If you want to learn the deployment of both, the ODM and Content containers, then you can do both Exercise 1 and Exercise 2 and follow the sequence in this book.

However, if you do not want to work on both the deployments and are interested in just one of them - ODM or Content, you have that flexibility. Feel free to skip either Exercise 1, or Exercise 2 if you want to deploy only one container - ODM or Content. You can also choose to change the order of the deployment by doing Exercise 2 first and then Exercise 1, if you want.

Keep in mind, you must complete at least Exercise 1 or Exercise 2 or both before working on Exercise 3 on container management. Furthermore, if you skip Exercise 1 or Exercise 2, several screen captures in Exercise 3 might not match exactly with your environment and you need to modify few steps to match your project.

If you skip Exercise 1 and do Exercise 2 only, then you can use the cpe-lab project to do all your work in this Exercise 3. You can also use the container from the cpe-lab project instead of the containers that are deployed in the odm-lab project. Several screen captures and references might not match exactly with your environment so you need to modify your steps to match your environment.

If you skip Exercise 2 and do Exercise 1 only, then you can use the odm-lab project to do all your work in this exercise. Few screen captures and references might not match exactly with your environment so you need to modify your steps slightly to match your environment.

Information

Both the deployment exercises - Exercise 1 and Exercise 2 are allocated 5 hours each for completion. It provides plenty of time to troubleshoot and resolve any mistakes or errors in your labs. Make the best use of the time and do not rush through the labs as that can cause unnecessary typographical errors or user mistakes.

Exercise 1. Deploying the IBM Operational Decision Manager (ODM) container

Estimated time

05:00

Overview

In this exercise, you complete the required steps to prepare, configure, and install IBM Operational Decision Manager (ODM) V8.10.2, which is part of IBM Cloud Pak for Automation V9.0.1 on top of a Red Hat OpenShift container platform (RHOCP) V3.11.

Objectives

After completing this exercise, you should be able to:

- Start and shut the lab environment
- Connect to Red Hat OpenShift Container Platform (RHOCP)
- Load the ODM docker images for Cloud Pak for Automation (CP4A)
- Create and secure the ODM database
- Deploy the ODM container on RHOCP
- Verify the successful ODM deployment
- Troubleshoot the deployment
- Connect to the ODM containers and successfully log in to the Decision Center console and Rule Execution Server console

Introduction

The IBM Cloud Pak for Automation (CP4A) offers a software platform to develop, deploy, run, and manage your digital business automation projects by using its capabilities. In this exercise, you deploy the ODM containers.

Requirements

Availability of the lab environment that consists of three Red Hat Enterprise Linux V7.7 virtual machines (VMs) with Db2 Enterprise V11.1.1.1 and Red Hat OpenShift (RHOCP) 3.11 installed. The required IBM Cloud For Automation (CP4A) software images are already downloaded on a VM.

Virtual machine configuration

Each virtual machine is configured with the following specification:

Table 1.

OS	CPUs	RAM	Disk
Red Hat Enterprise Linux (RHEL) 7.7 (64-bit)	8	16 GB	300 GB

List of Servers with Roles

The virtual machines are listed below with their respective roles:

Table 2.

VM	RHCOCP Node type	IP address	Hostname
VM1- OCP master	Master	`10.0.0.1`	`master.cp4a.com`
VM2- OCP compute1	Compute	`10.0.0.2`	`compute1.cp4a.com`
VM3- OCP compute 2	Compute	`10.0.0.3`	`compute2.cp4a.com`

Software Requirements

The following packages are downloaded to the **Master** node:

Table 3.

Software	Folder name
IBM Cloud Pak for Automation images	`/root/labfiles/cp4a`

User IDs and Passwords

The following table contains a list of User ID and password information for this exercise:

Table 4.

Entry Point	User ID	Password
OpenShift web console: `https://master.cp4a.com:8443` or `https://console.cp4a.com:8443`	`admin`	`passw0rd`
Red Hat Linux VM	`root`	`passw0rd`
Db2 Enterprise	`db2admin`	`passw0rd`
Decision Center console: `https://master.cp4a.com:< Decision Center console NodePort>`	`odmAdmin`	`passw0rd`
Rule Execution Server console: `https://master.cp4a.com:< Rule Execution Server console NodePort>`	`odmAdmin`	`passw0rd`

Stop

Course updates and errata

A Course Corrections document might be available for this course.

If you are taking the class with an instructor, the instructor can provide this document to you.

If you are taking the course in a self-paced environment, the course corrections document is provided with the other manuals.

To check whether a Course Corrections document exists for this course:

1. Go to the following URL: http://www.ibm.com/developerworks/connect/middleware_edu
2. On the web page, locate and click the **Course Information** category.
3. Find your course in the list and click the link.
4. Click the **Attachments** tab to see whether an errata document exists with updated instructions.
5. To save the file to your computer, click the document link and follow the dialog box prompts.

Exercise instructions

Part 1: Work with the development environment

Before you start working with this exercise, it is important to get familiar with the lab environment. In this section, you learn how to work with the environment. Give special attention to the steps you need to take when you go for a long break or return to the labs after extended period (such as the start of the next day).

Read the following three sections below carefully before starting your environment and connecting to your VM in step 1:

- Understand the various states of the lab environment
- Decide on using either RDP or Browser to connect to your VM
- Copy and paste code in the environments

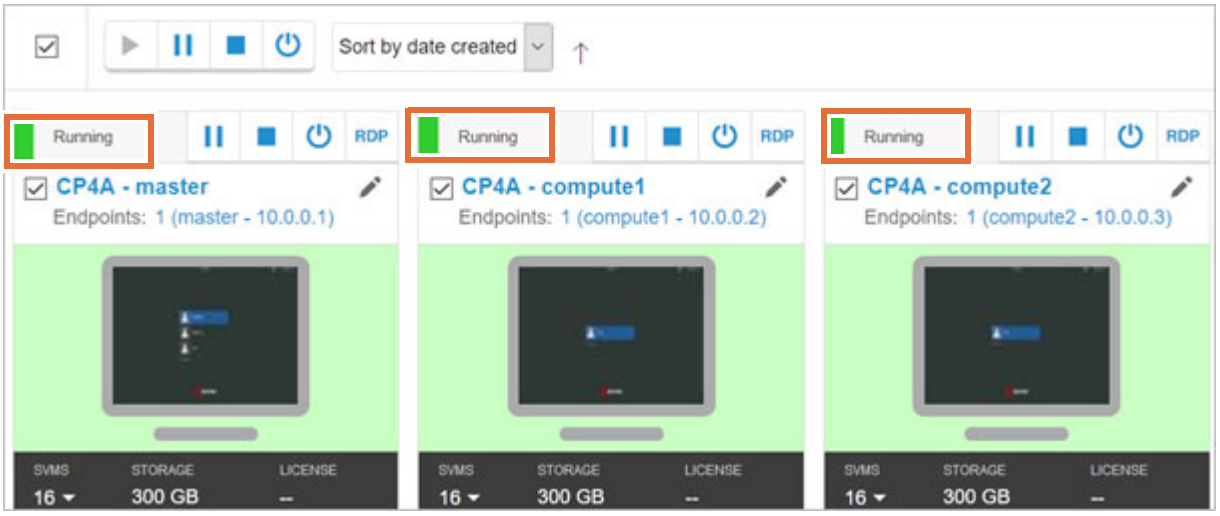
Understand the various states of the lab environment

The instructions below describe the states of the lab environment. A good understanding ensures a pleasant user experience and minimizes any unpredictability in the lab environment.

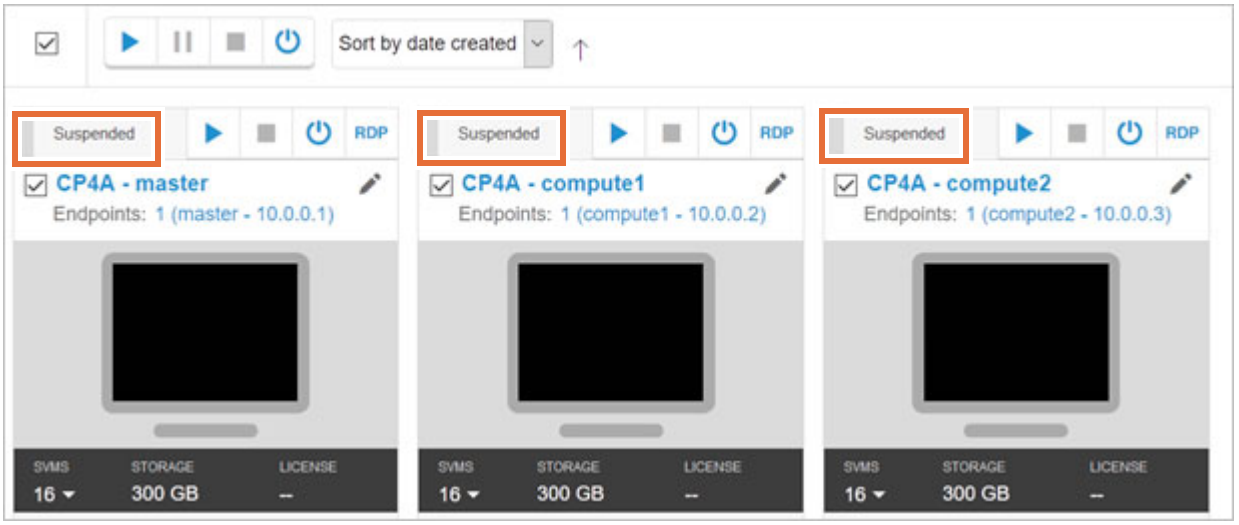
Three RHEL VMs make up your lab environment. Red Hat OpenShift (RHOCP) 3.11 is already installed. The RHOCP cluster consists of one master and two compute nodes. VM1 is the master. This master manages the two compute nodes - compute1 and compute2. It also schedules pods to run on those compute nodes.

Each square box represents one VM. For the three VMs, three square boxes are displayed next to each other in one row. The environment can be in any of the three states - Running, Suspended, or Powered off.

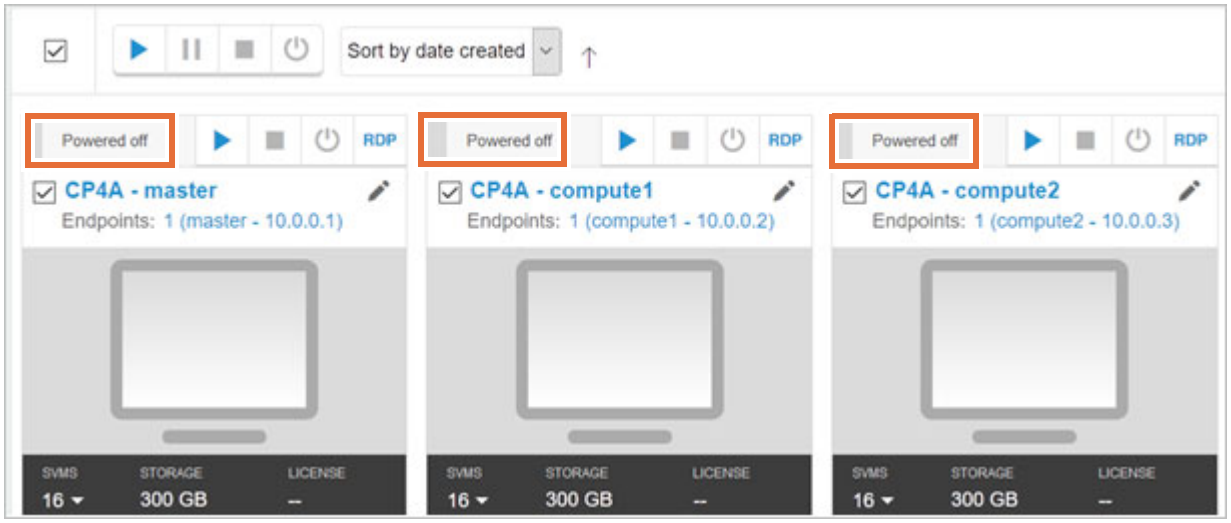
Running state: The environment is running if it displays **Running** for each VM box. To work with the lab environment, all the three VMs must be in a Running state. Before you can work with the labs, always verify that all the three VMs are in a Running state.



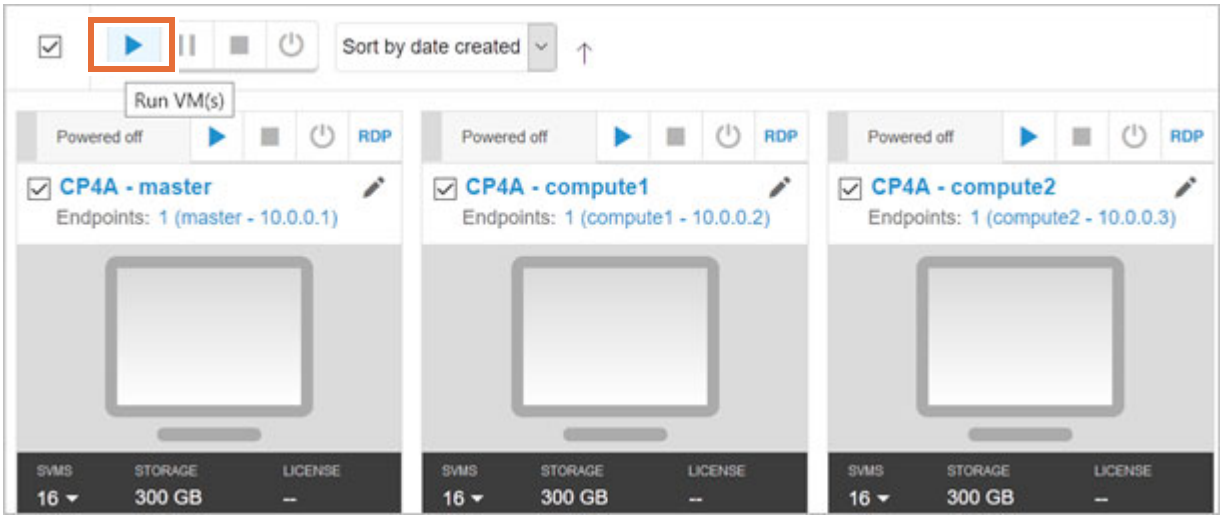
Suspended state: The environment is suspended when it displays **Suspended** for any single VM box. After **2 hours** of inactivity, the environment automatically suspends to conserve resources. Usually all the three VMs are suspended together. You cannot work with the lab environment even if a single VM is in a suspended state. That is because all the three VMs need to communicate with each other.



Powered off state: The environment is powered off or shuts down when it displays **Powered off** for any single VM box. Usually all the three VMs are powered off together. You cannot work with the lab environment even if a single VM is in a powered off state since all the three VMs need to communicate with each other.

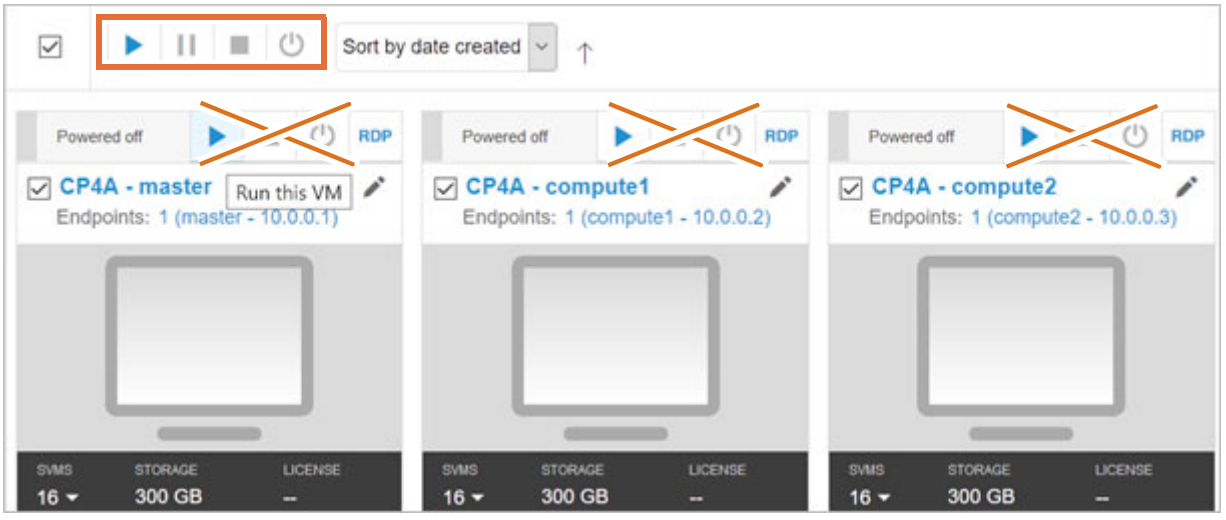


How to switch from a Powered off state to a Running state? If the environment is Powered off and you are ready to work on the labs, then you need to bring the environment to a Running state. You do that by clicking the **Run VM(s)** icon that is displayed at the top to start the environment. Wait about 10 minutes for all the VMs and the services to come up before you are ready to log in.

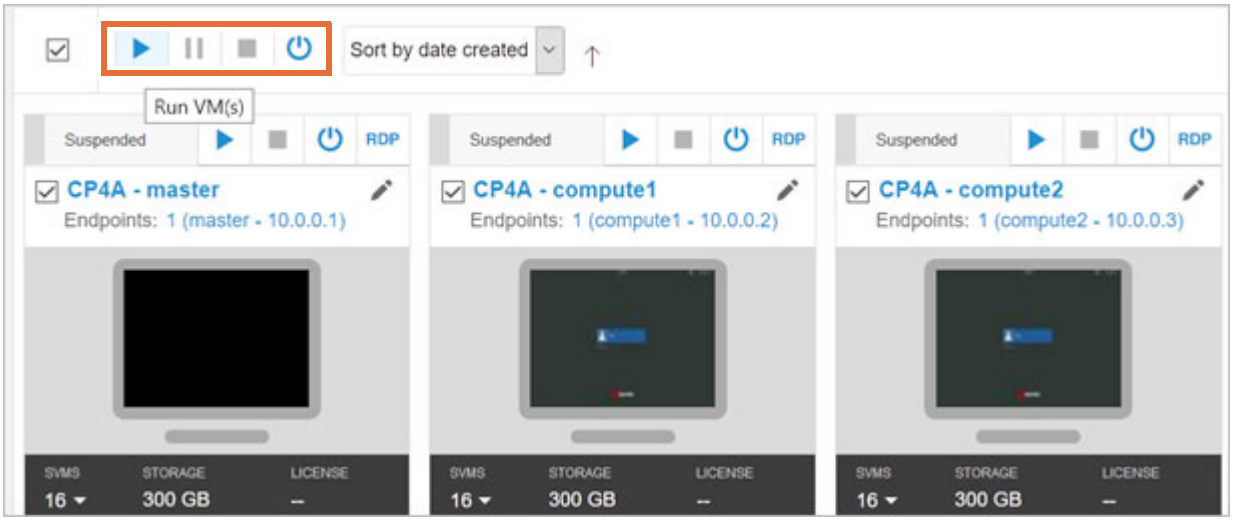


Important

When starting the environment from the Powered off state, make sure to click the **Run VM(s)** icon that is displayed above the three VMs. This step starts all the three VMs properly with a single click. Do **NOT** click **Run this VM** next to each VM to start them one at a time.



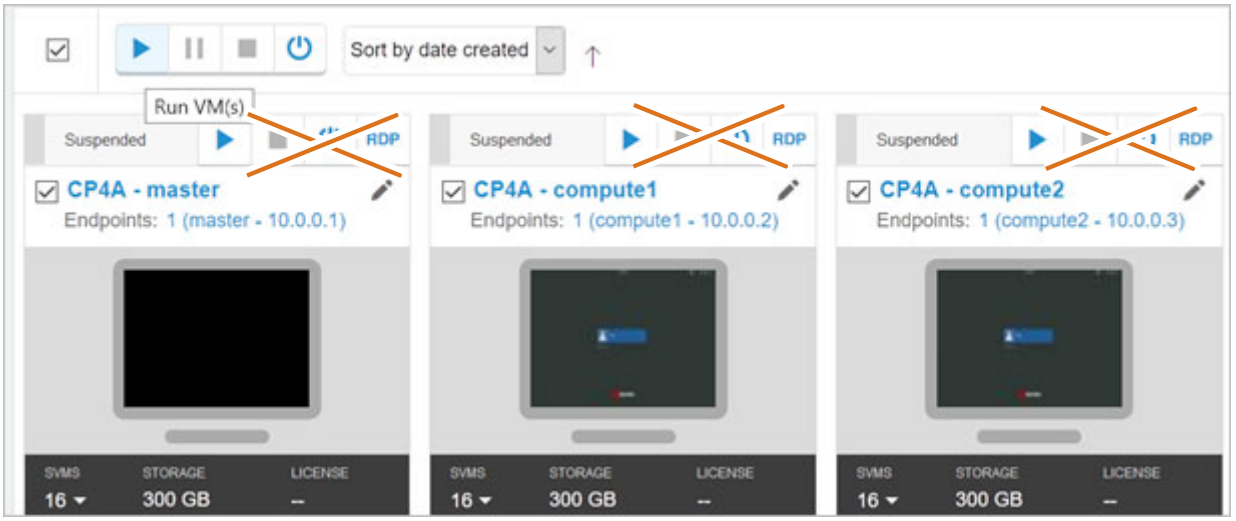
How to switch from a Suspended state to a Running state? When the environment is suspended and you are ready to work on the labs, then you need to bring the environment to a **Running** state. You do that by clicking the **Run VM(s)** icon to start the environment. Wait at least 10 minutes for all the VMs and the services to come up before you are ready to log in.



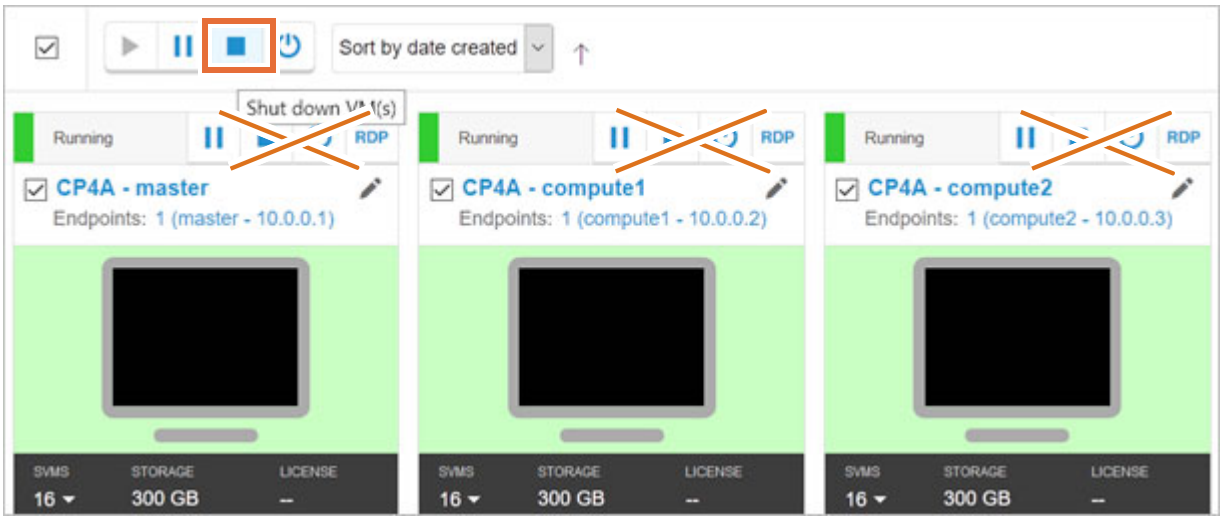
If you log in too quickly, sometimes the VM might not be responsive for a while when starting a VM from a suspended state. **Even though the status changes to Running, give it another 5 minutes before connecting. This wait ensures that all services are up and the VMs are communicating with each other.**

Important

When starting the environment from the Suspended state, make sure to click the **Run VM(s)** icon that is displayed above all the three VMs. This step starts all the three VMs properly with a single click. Do **NOT** click the **Run this VM** next to each VM to start them one at a time.



What steps to take before you go for a break of more than 2 hours or are done for the day? It is a good idea to shut down the environment when you plan to be away for over 2 hours. It prevents the environment from going into the Suspended state. Power off the entire environment by clicking the **Shut down VM(s)** icon that is displayed above all the three VMs. This step stops all the three VMs properly with a single click. Do **NOT** click the **Shutdown this VM** next to each VM in an attempt to stop them one at a time as that can cause a VM (usually a compute VM) to hang.



What steps to take after you return to labs from a break of more than 2 hours? When you return after your long break or start work the next day and if you manually shut down the environment as described earlier, you can click the Run VM(s) icon that is displayed at the top of the environment. It starts all the VMs together with a single click. Wait at least 10 minutes before logging in to the VM and resuming your lab exercise.

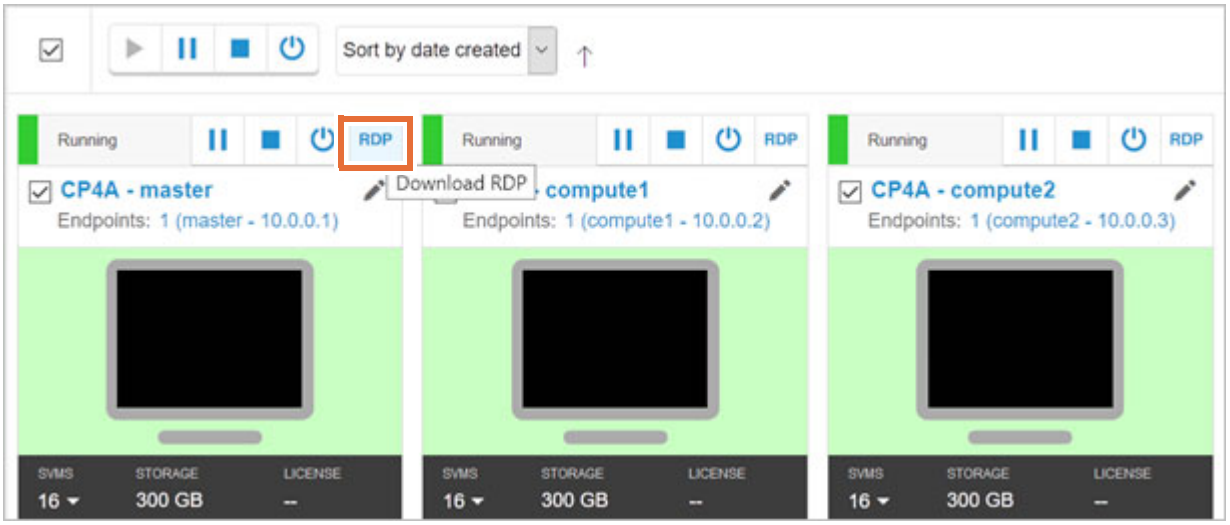
If you forgot to shut down the environment before going for a long break, then your environment state changes to **Suspended** after 2 hours. To bring it to Running state, you click the **Run VM(s)** icon that is displayed at the top of the environment and that starts all the VMs together. Wait at least 10 minutes before logging in to the VM. It is possible that the Master VM is unresponsive or unexpected results and behavior occurs after logging in. If that happens, go back to your earlier steps that you were doing before the break and verify that they are working correctly first before continuing with your lab. If things still do not appear to be working correctly, then shut down the entire environment by clicking the **Shut down VM(s)** icon that is displayed above all the three VMs. After you shut down the environment, you can then start the environment by clicking **Run VM(s)**. which starts all the three VMs together. This process ensures a clean start of the VMs. Continue working with the labs where you left off before.

Decide on using either RDP or Browser to connect to your VM

You can connect to any of the three VMs by either using your browser or Remote Desktop Protocol (RDP). If you open RDP or browser sessions for each of the three VMs, make sure to close the ones not being used to avoid working with the wrong VM. You work with the Master VM only in the exercise.

Accessing the VM by using the browser: A browser is the quickest and easiest way to access and use your VM's desktop. You can start the browser by clicking the thumbnail image of the VM in the square box.

Accessing the VM by using RDP: By default, RDP is enabled in all of your VMs. To use RDP, click the RDP icon in the upper right corner of the image thumbnail as displayed below.



The first time that you click the RDP icon for a VM, a window opens asking what Firefox should do with this file. Leave the **Open With Remote Desktop Connection default** option selected and click **OK** to open the VM desktop.

Information

If you use a browser to access the VM, you might notice a possible mouse lag sometimes. You can try both ways to connect to a VM and choose whichever one works best for you

Copy and paste code in the environment

Copy and paste is tricky in the environment. It is a good idea to enter all the commands manually in the command line as instructed the exercise. It ensures correct entry of characters.

When entering commands in the Linux Terminal, always enter the commands in a single line.

However, if you want to copy and paste some of the longer commands, you need to copy that command first into a text file, such as Notepad on your local desktop. **If the command displays in multiple lines in Notepad, make sure to format it to a single line.** Then, select and copy from Notepad before pasting into the Linux Terminal. If the paste does not work, then copy from Notepad file one more time and the second time the paste should work. During testing, it was found that repeating this step twice usually worked.

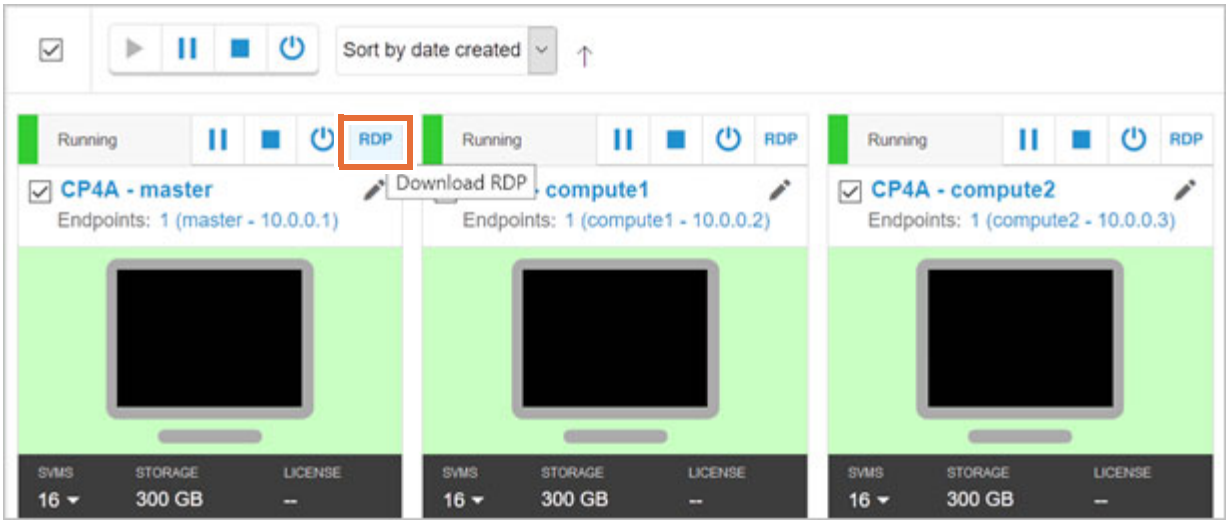
Connect to the Master VM in the environment

Now that you read the description of various states of the lab environment, you are ready to start and work with it.

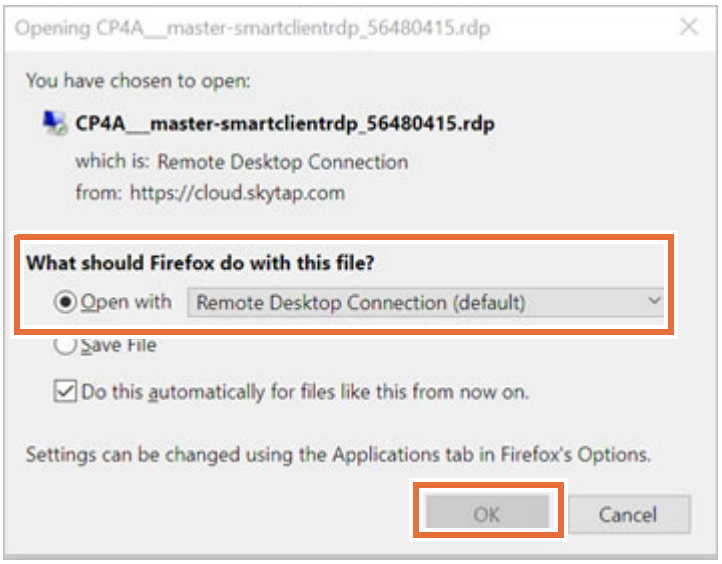
__ 1. Start your lab environment as mentioned in the previous instructions and make sure that it is in a Running state. All three VMs must be in the Running state.

__ 2. Use RDP or Browser to connect to the **Master** VM.

__ a. Connect with the **Master** VM by either clicking the thumbnail image of the VM to start in the browser or click the RDP icon in the upper right corner of the **Master** image thumbnail as displayed below.



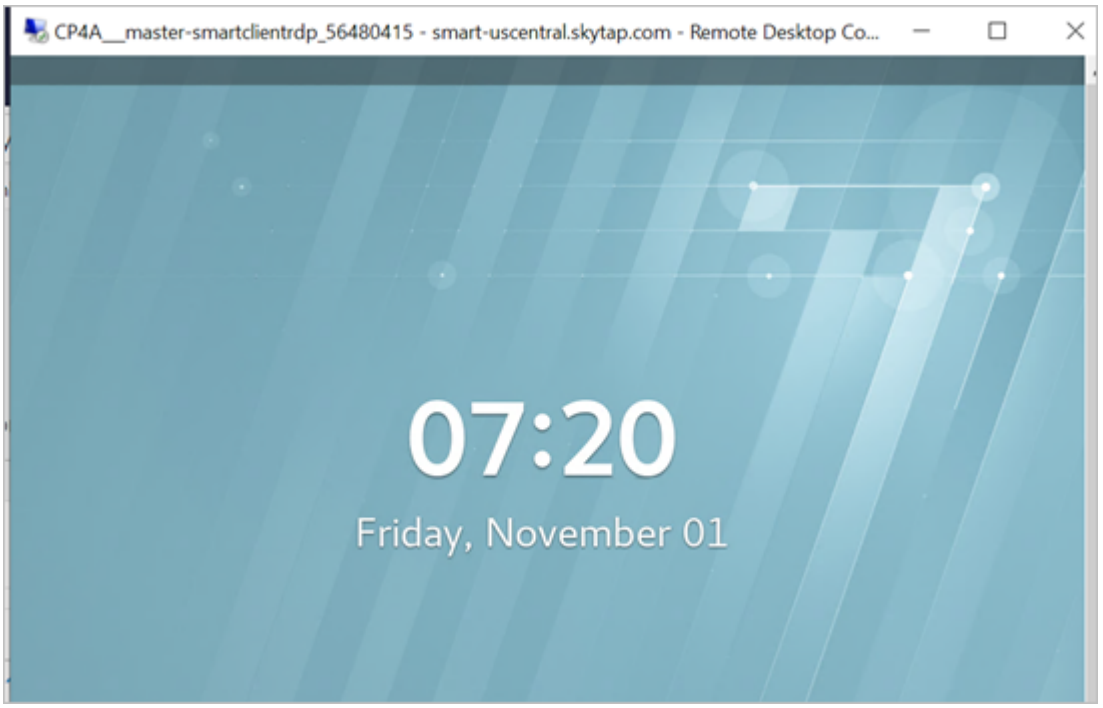
__ b. The first time that you click the RDP icon for the VM, a window opens asking what Firefox should do with this file. Leave the **Open With Remote Desktop Connection default** option selected and click **OK** to open the VM desktop.



You are now ready to work with the Master VM. Go to the next step to learn how to log in to the VM.

__ 3. Log in to the **Master** VM. Remember, that you work with the **Master** VM throughout the exercise. All the commands that you run in this exercise, you run on the **Master** VM. Optionally, you can connect to the compute VMs to explore.

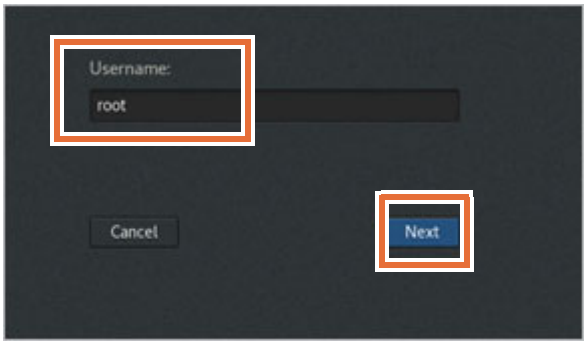
__ a. Regardless of the way you connect to the Master VM (either through the Browser or RDP), the time and date. is displayed on the desktop the first time you connect. Click the Enter key anywhere on the desktop to display the login screen.



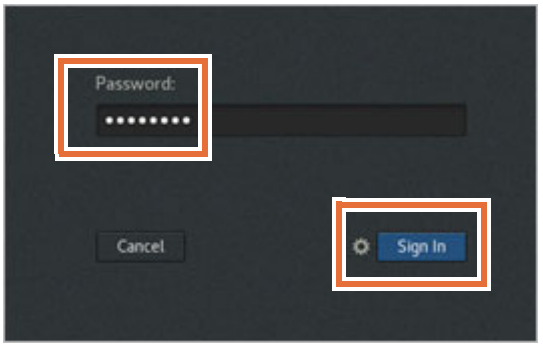
__ b. Click the Enter key anywhere on the desktop to display the login screen.

__ c. Click **Not listed**.

__ d. Enter `root` in the Username field and click **Next**.



__ e. Enter `passw0rd` in the Password field and click **Sign In**.



__ f. You are now successfully logged in to the VM.

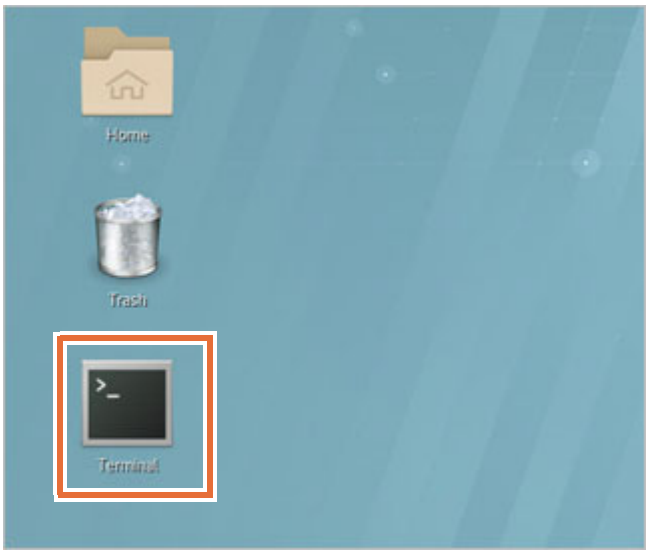
Part 2: Load the ODM docker images for Cloud Pak for Automation

The installation of Cloud Pak for Automation requires downloading and deploying the appropriate docker images. The CP4A images needed for the labs are already downloaded and available under the `/root/labfiles/cp4a` folder. In this section, you load the required ODM images for CP4A.

__ 1. Verify that you are connected to the Master VM as instructed in the previous section.

__ 2. Log in to Red Hat OpenShift by using the command line

__ a. Double-click the Terminal shortcut on the desktop to open it.

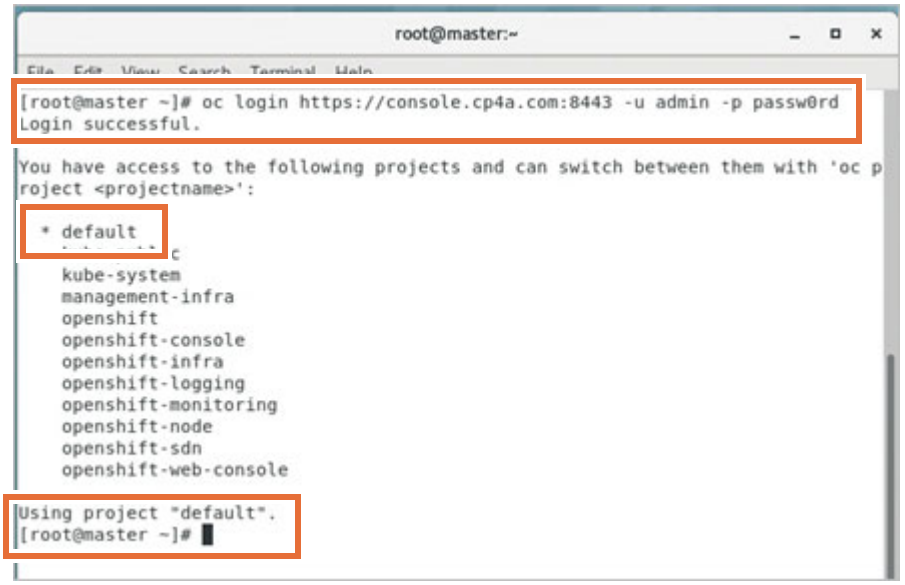


__ b. In the terminal that opens, login to the OpenShift cluster. The OpenShift console URL is `https://console.cp4a.com:8443` and the username/password `is admin/passw0rd`.

```
oc login https://console.cp4a.com:8443 -u admin -p passw0rd
```

__ c. Verify that the Login successful message is displayed.

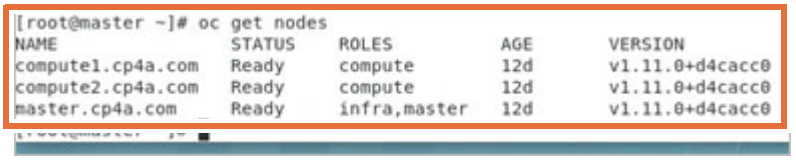
__ d. Examine the list of projects that is displayed. The asterisk * next to a project name indicates the project that is being used in the current session. Verify that default project is being used. You do not work with this default project but create a new one in a later step.



__ 3. Check the status of the nodes.

__ a. Verify that the master and the two compute nodes are in the ready state.

```
oc get nodes
```



__ b. Verify that three nodes are listed and the status is Ready for all the nodes.

__ 4. Create an OpenShift project for ODM

__ a. Create a project called **odm-lab** for ODM

```
oc new-project odm-lab
```

You are now ready to use this new project.

__ 5. Load the ODM docker images

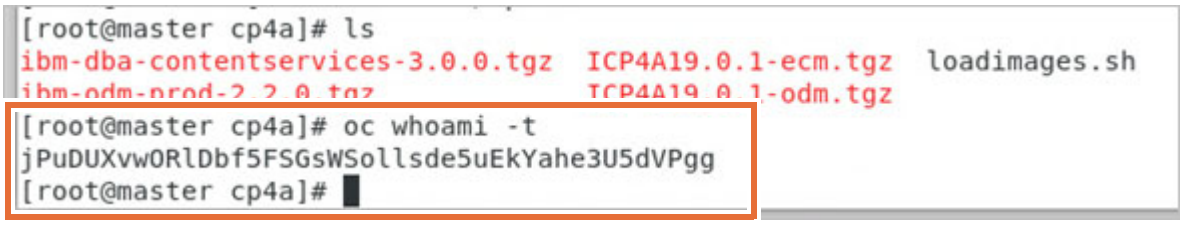
__ a. Identify the images for ICP4A that are located under `/root/labfiles/cp4a` folder

```
cd labfiles/cp4a
```

```
ls
```

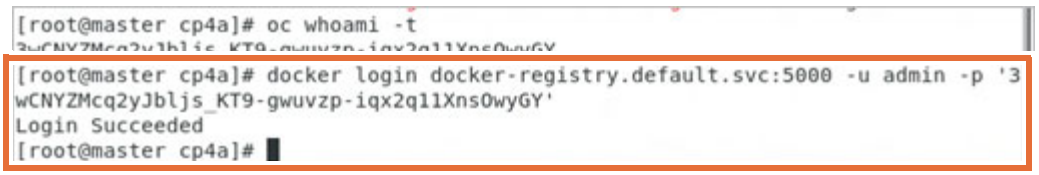
__ b. Create a login token for the internal docker registry to connect to the internal docker registry.

```
oc whoami -t
```



__ c. Log in to the internal docker registry by using an authentication token. Replace the token variable with the token that you created in the previous step. Make sure to include the token in quotation marks (single or double quotation marks). Do not copy and paste the command from the pdf because the VM does not interpret the quotation marks correctly.

```
docker login docker-registry.default.svc:5000 -u admin -p '<token>'
```



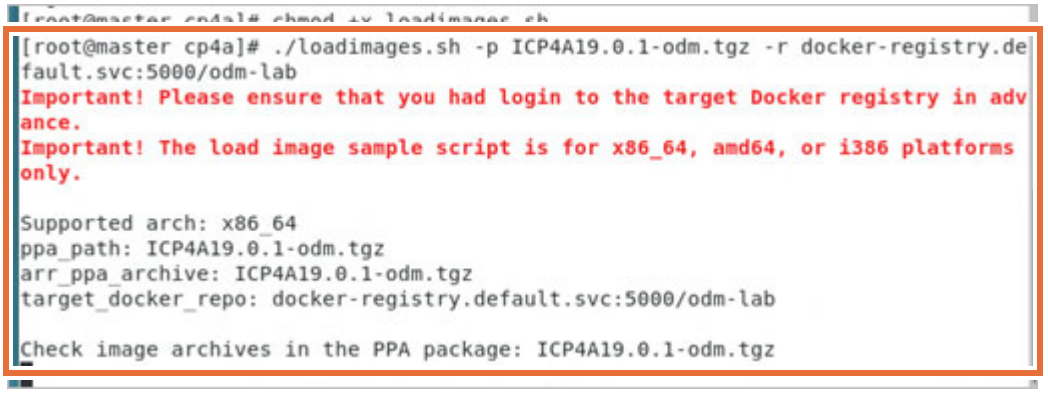
__ d. You are now ready to push the ODM images to the internal docker registry. Before you load the registry, you can examine the `loadimages.sh` shell script that is included under the `/root/labfiles/cp4a` folder. Open the `loadimages.sh` script using an editor of your choice. Optionally, you can double-click the **Home** folder on the desktop, go to `/root/labfiles/cp4a` and then right-click loadimages.sh to open with Text Editor. Examine the `loadimages.sh` script and close it when done. Do not make any changes. This script loads the images and pushes them to internal docker registry.

__ e. Change the permission of the shell script before you run it.

```
chmod +x loadimages.sh
```

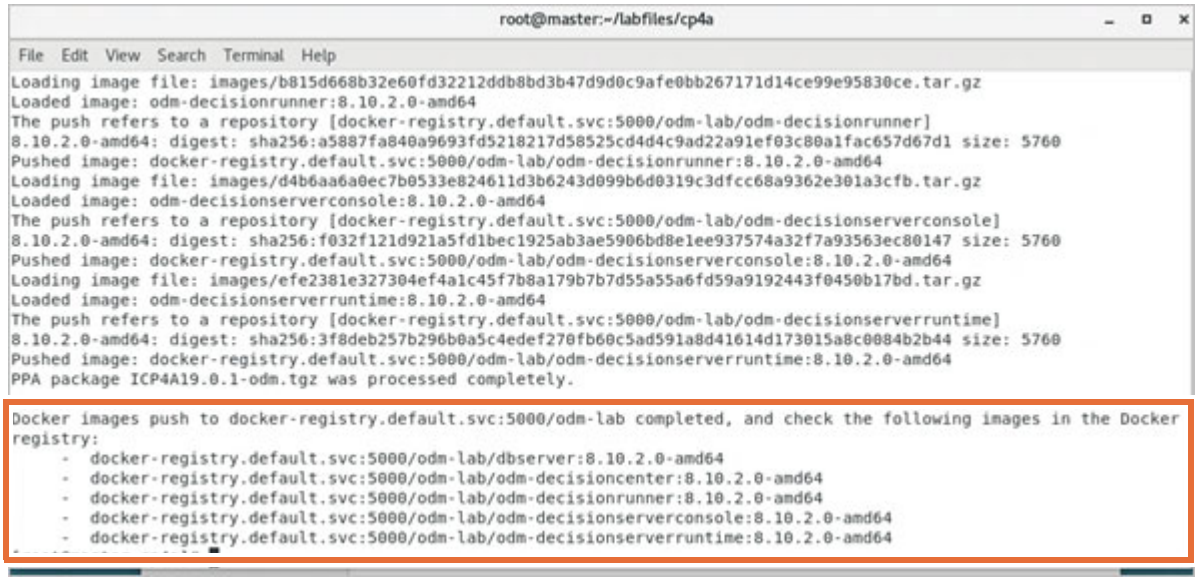
__ f. Push the ODM images (`ICP4A19.0.1-odm.tgz`) to the internal docker registry. Do not enter any space in the command:

```
./loadimages.sh -p ICP4A19.0.1-odm.tgz -r docker-registry.default.svc:5000/odm-lab
```



It takes about 10 minutes for the command to run and the images to load. Wait until the command completes.

__ g. When the command completes, verify that all the ODM images are successfully loaded. Five images are listed.



You are now ready to create the ODM database.

Part 3: Prepare the required ODM database

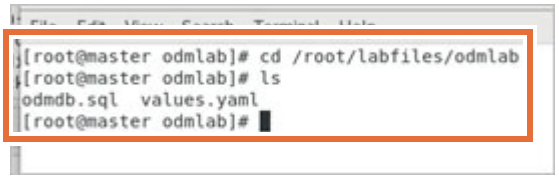
In this section, you create the required ODM database and connect to it.

__ 1. Create an ODM database

__ a. Switch to the `/root/labfiles/odmlab` folder and list the contents of the folder. The database script odmdb.sql is available there.

```
cd /root/labfiles/odmlab
```

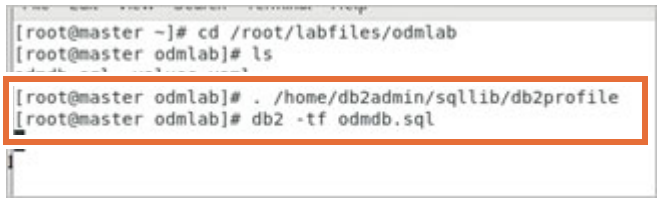
```
ls
```



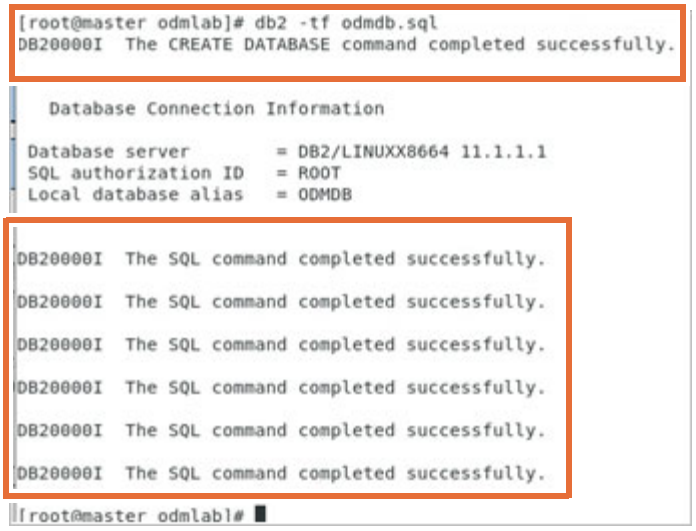
__ b. Initialize the Db2 environment variable and run the create database script.

```
. /home/db2admin/sqllib/db2profile
```

```
db2 -tf odmdb.sql
```



__ c. It takes few minutes for the database script to run. Verify that the script ran successfully and the database is created.



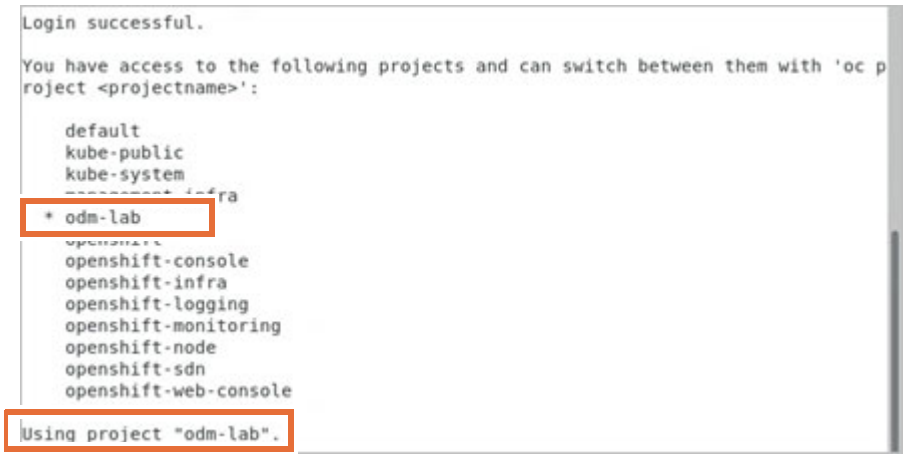
__ 2. Secure the ODM database.

To secure access to the database, a secret must be created to encrypt the database user and password before installing the helm chart.

__ d. Make sure that you are logged in to the OpenShift cluster. The OpenShift console URL is `https://console.cp4a.com:8443` and the username/password `is admin/passw0rd`.

```
oc login https://console.cp4a.com:8443 -u admin -p passw0rd
```

__ e. Verify that the Login successful message is displayed and that you are using the odm-lab project.



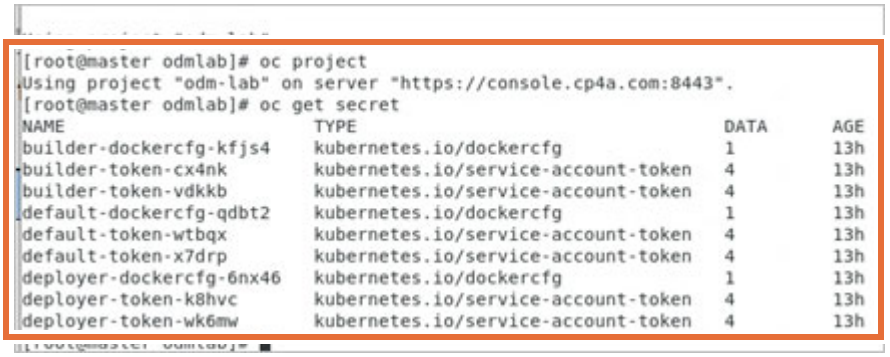
__ f. Another way to find out which project you are using is to query OpenShift through the project command.

```
oc project
```

__ g. Before creating a secret, it is a good idea to ensure that another secret with the same name does not exist in the cluster.ine.

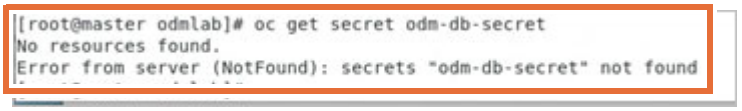
```
oc get secret
```

__ h. Check the list of secrets that are returned and make sure that the odm-db-secret is not listed.



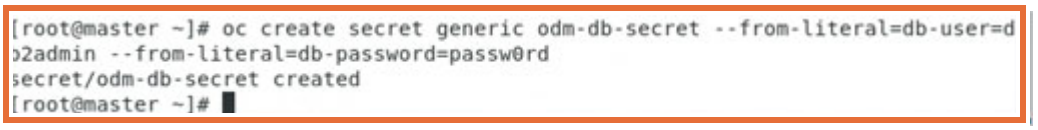
__ i. Alternatively, instead of listing all the secrets and looking for one through the list, you can also check for the odm-db-secret specifically.

```
oc get secret odm-db-secret
```



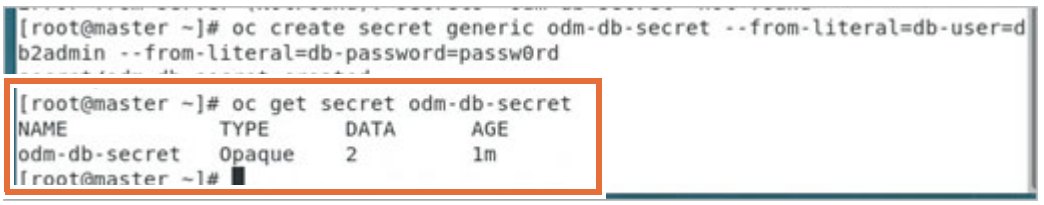
__ j. Now that you confirmed that the secret does not exist, you are ready to create it. Enter the entire command on a single line:

```
oc create secret generic odm-db-secret --from-literal=db-user=db2admin
--from-literal=db-password=passw0rd
```



__ k. Verify that odm-db-secret is created.

```
oc get secret odm-db-secret
```



You are now ready to deploy the ODM container.

Part 4: Deploy the ODM container

__ 1. Add privileges to the ODM project.

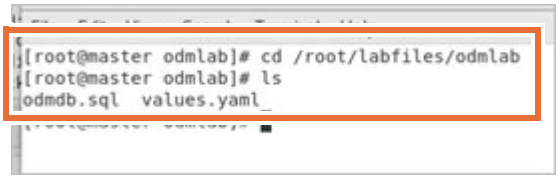
```
oc adm policy add-scc-to-user privileged -z default
```

__ 2. Install the helm chart for deployment

__ a. Review the yaml file. Make sure that you are in the `/root/labfiles/odmlab` folder to list the contents of the folder. The `values.yaml` is available there.

```
cd /root/labfiles/odmlab
```

```
ls
```



__ b. Open the `values.yaml` file by using an editor of your choice. Optionally, you can double-click the **Home** folder on the desktop, go to `/root/labfiles/odmlab` and then right-click `values.yaml` to open with Text Editor. Examine the `values.yaml` script and close it when done. Do not make any changes as this file contains the properties for the helm chart that you use in the next step.

__ 3. Run the installation.

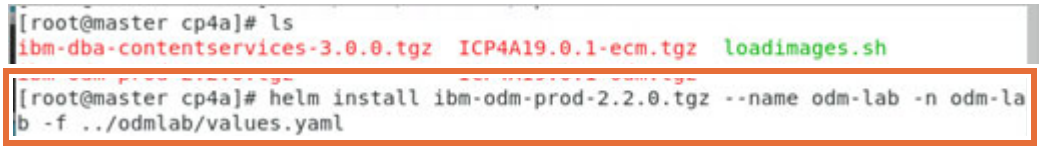
__ a. Switch to the cp4a folder and list the contents of the folder. The `ibm-odm-prod-2.2.0.tgz` helm chart is available there.

```
cd /root/labfiles/cp4a
```

```
ls
```

__ b. Install the ODM helm chart using helm command.command:

```
helm install ibm-odm-prod-2.2.0.tgz --name odm-lab -n odm-lab -f ../odmlab/values.yaml
```



__ c. The command completes quickly and returns at the command prompt.

```
)
       echo $SCHEME://$NODE_IP:$NODE_PORT_DC/teamserver

    -- Decision Runner
       export NODE_PORT_DR=$(kubectl get --namespace odm-lab -o jsonpath="{.spec.ports[0].nodePort}" services odm-l
ab-odm-decisionrunner)
       export NODE_IP=$(kubectl get nodes --namespace odm-lab -o jsonpath="{.items[0].status.addresses[0].address}"
)
       echo $SCHEME://$NODE_IP:$NODE_PORT_DR/DecisionRunner

    -- Decision Server Console
       export NODE_PORT_DSC=$(kubectl get --namespace odm-lab -o jsonpath="{.spec.ports[0].nodePort}" services odm-
lab-odm-decisionserverconsole)
       export NODE_IP=$(kubectl get nodes --namespace odm-lab -o jsonpath="{.items[0].status.addresses[0].address}"
)
       echo $SCHEME://$NODE_IP:$NODE_PORT_DSC/res

    -- Decision Server Runtime
       export NODE_PORT_DSR=$(kubectl get --namespace odm-lab -o jsonpath="{.spec.ports[0].nodePort}" services odm-
lab-odm-decisionserverruntime)
       export NODE_IP=$(kubectl get nodes --namespace odm-lab -o jsonpath="{.items[0].status.addresses[0].address}"
)
       echo $SCHEME://$NODE_IP:$NODE_PORT_DSR/DecisionService

  To learn more about the odm-lab release, try:

    $ helm status odm-lab
    $ helm get odm-lab

[root@master cp4a]#
```

If everything went well, the ODM container is deployed in few minutes. You verify that in the next section.

Part 5: Verify successful ODM deployment

In this section, you run several verification steps for successful ODM deployment

__ 1. Check the status of the pod. A list of pods returns with a status of either running, failing, starting, crashing or something else. Although it takes time for the containers to start, checking the status first gives a good idea of the state of the pod.

__ a. Check the status of the pod.

```
oc get pods
```

__ b. Verify that there are four rows for pods.

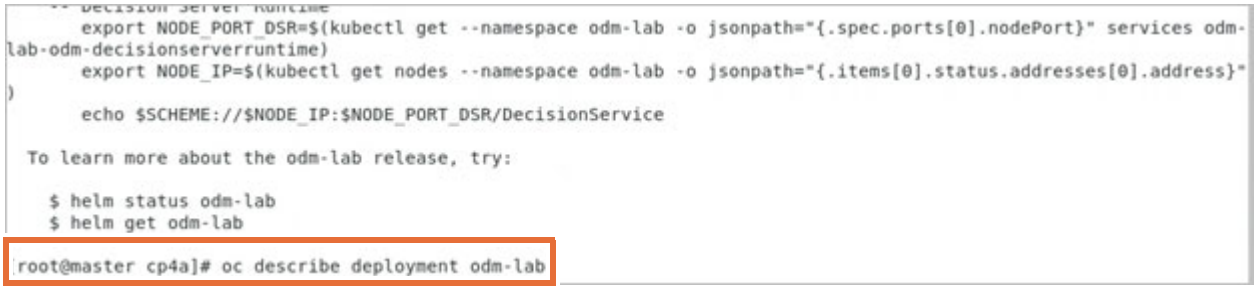
```
[root@master cp4a]# oc get pods
NAME                                                READY     STATUS              RESTARTS   AGE
odm-lab-odm-decisioncenter-7f78864cb4-kwvb7         0/1       ContainerCreating   0          22s
odm-lab-odm-decisionrunner-b6bf759fd-vccm4          0/1       ContainerCreating   0          22s
odm-lab-odm-decisionserverconsole-77c88fbb85-snrlb  0/1       ContainerCreating   0          22s
odm-lab-odm-decisionserverruntime-78c44b4547-ck776  0/1       ContainerCreating   0          22s
[root@master cp4a]#
```

__ c. Examine the status of the pods. They are being created. It takes few minutes for the pods to start and get into a Running status. It is OK if you see a Ready state of 0/1 for containers. Give it few minutes to change to state 1/1. While you are waiting for the state to change, continue to the next step for some verifications.

__ 2. Verify the deployment status. When the deployment is started, the helm chart creates among others artifacts, a deployment.

__ a. Run the command to check deployment and get detailed information about the deployment.

```
oc describe deployment odm-lab
```



__ b. Verify that a message is received that the replica set was successfully scaled to 1.

__ 3. Check the replica set.

__ a. Run the command to check the replica and return the information that a pod is created.

```
oc describe replicaset odm-lab
```

__ 4. Check the pod for messages that are generated during its creation.

__ a. Run the command to view the pod creation messages and watch for errors.

```
oc describe pod odm-lab
```

```
[root@master cp4a]# oc describe pod odm-lab
```

```
  Type    Reason     Age   From                          Message
  ----    ------     ----  ----                          -------
  Normal  Scheduled  1m    default-scheduler             Successfully assigned odm-lab/odm-lab-odm-decisionserverrunt
ime-78c44b4547-6mfcf to compute1.cp4a.com
  Normal  Pulling    1m    kubelet, compute1.cp4a.com  pulling image "docker-registry.default.svc:5000/odm-lab/odm-
decisionserverruntime:8.10.2.0-amd64"
  Normal  Pulled     1m    kubelet, compute1.cp4a.com  Successfully pulled image "docker-registry.default.svc:5000/
odm-lab/odm-decisionserverruntime:8.10.2.0-amd64"
  Normal  Created    59s   kubelet, compute1.cp4a.com  Created container
  Normal  Started    59s   kubelet, compute1.cp4a.com  Started container
```

__ 5. Check the status of the pod one more time.

__ a. Now that some time is passed, you run the verification again to make sure that the state of containers is 1/1 and that all have a Running status.

```
oc get pods
```

```
[root@master cp4a]# oc get pods
NAME                                              READY     STATUS    RESTARTS   AGE
odm-lab-odm-decisioncenter-7f78864cb4-fc6l8       1/1       Running   0          4m
odm-lab-odm-decisionrunner-b6bf759fd-9rh59        1/1       Running   0          4m
odm-lab-odm-decisionserverconsole-77c88fbb85-f7xhv   1/1       Running   0          4m
odm-lab-odm-decisionserverruntime-78c44b4547-6mfcf   1/1       Running   0          4m
```

__ b. The successful installation results in containers with a Running status. Make sure that the status is Running for all containers. It is important that the container also displays 1/1 as in Ready. If it displays 0/1, then give it few more minutes before it changes to 1/1.

Part 6: Troubleshoot failed deployment.

Several tasks are available for you troubleshoot a failed deployment. This section describes some of the steps you can take to ensure a successful deployment.

__ 1. Check for typographical errors. If your deployment fails or you see an error, then the first thing to check is to make sure that there are no typographical errors. Check the typed command to see any obvious mistakes. You can scroll up through the terminal and examine the previous commands run, if needed. You can use the up arrow key in your keyboard to recall the earlier run commands to scan through any mistakes and make corrections and run them again.

__ 2. Verify that the lab environment is up and running. If you just started a suspended environment, then make sure that you give plenty of time before working with the environment. A good verification is to log in to the OpenShift console and make sure that you are successful.

```
oc login https://console.cp4a.com:8443 -u admin -p passw0rd
```

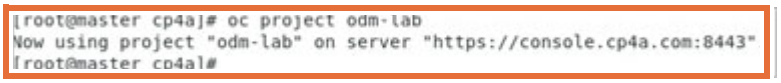
__ 3. Verify the current project.

__ a. Verify that you are connected to the correct project - odm-lab.

```
oc project
```

__ b. If the current project is the default project or some other project, then switch to the odm-lab project.

```
oc project odm-lab
```



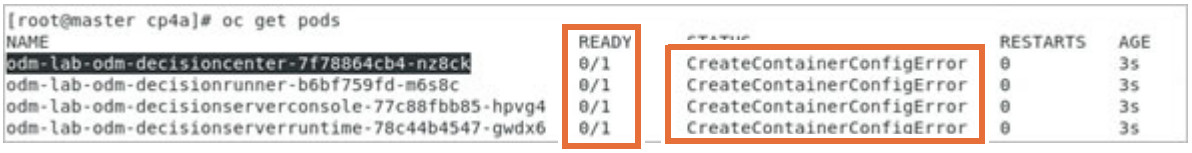
__ 4. Check the logs for the pod, fix errors, and then redeploy.

__ a. You can also check the logs for a failed or crashing pod to help troubleshoot the problem. First, check the pod status to identify the pod name that is being used in the current run.

```
oc get pods
```

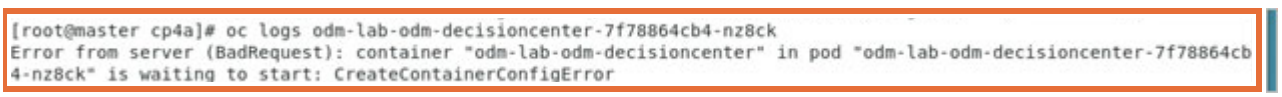
The state of a pod might be stuck at 0/1 even though it says it is Running. If it does not change to 1/1 even after a while then you might want to check the log for that pod. Copy the name of that pod.

In other failures, it is possible that a pod fails to start and gives an error. If that happens, then you can again copy the name of the failing pod (s) to check the logs.



__ b. Check the logs of a specific problematic pod by using its name.

```
oc logs odm-lab-odm-decisioncenter-7f78864cb4-nz8ck
```

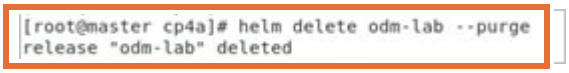


The result above indicates a problem with the request. You can go back to see whether any mistake was done or a step was skipped.

__ 5. Remove the current failed deployment, fix errors, and then redeploy. You do not need to run this step if you successfully deployed the ODM. In that case, you can review this step.

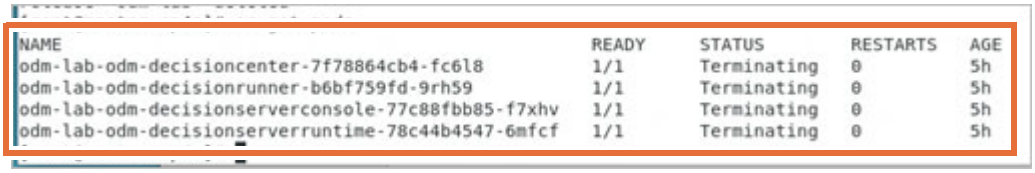
__ a. Run the command to purge the deployment. Do **NOT** run this command when the deployment is successful. Note that there are two dashes right before purge (even though it may look like a single dash) in the command. You can run this if you identified the problem, made the correction, and are ready to install the helm chart again for deployment.

```
helm delete odm-lab --purge
```

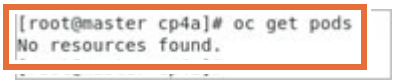


__ a. Check the pod status. The pods start terminating.

```
oc get pods
```



__ b. Run the check again to verify the pods again until resources are no longer available.



__ c. Now you are ready to install ODM helm chart again by using the helm command:

```
helm install ibm-odm-prod-2.2.0.tgz --name odm-lab -n odm-lab -f ../values.yaml
```

Part 7: Access the running ODM containers

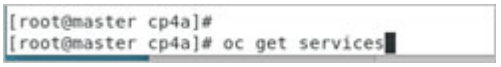
Now that ODM is deployed, you are ready to connect to it. In this section, you verify the connectivity and access to ODM. You do not do any ODM development or other work with ODM. It is the task of an ODM developer or ODM Administrator.

__ 1. Identify ports and URL for the running containers

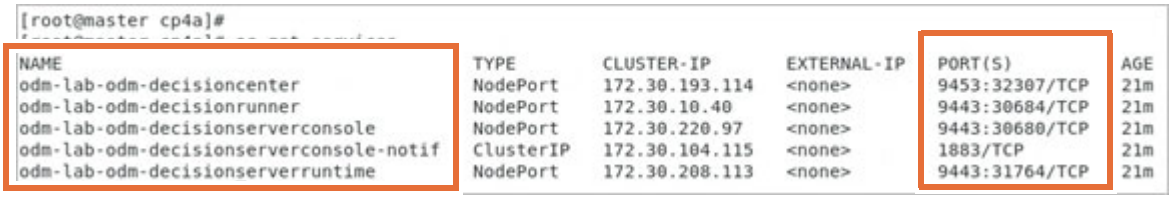
__ a. Verify that the pods are running.

__ b. Find the port numbers.

```
oc get services
```

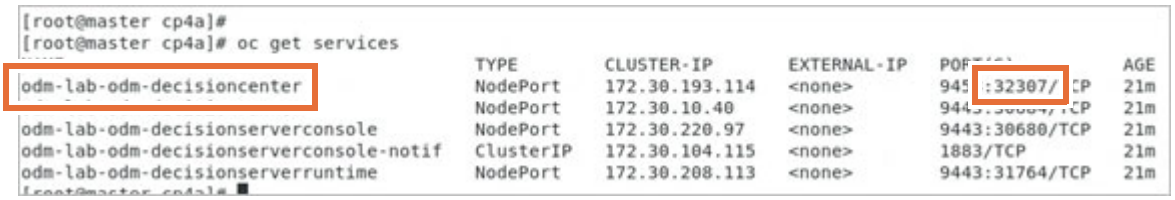


__ c. Examine the output from the command. It displays the node ports of the running services. You log in to the relevant services in the next steps.



__ 2. Connecting to the Decision Center console

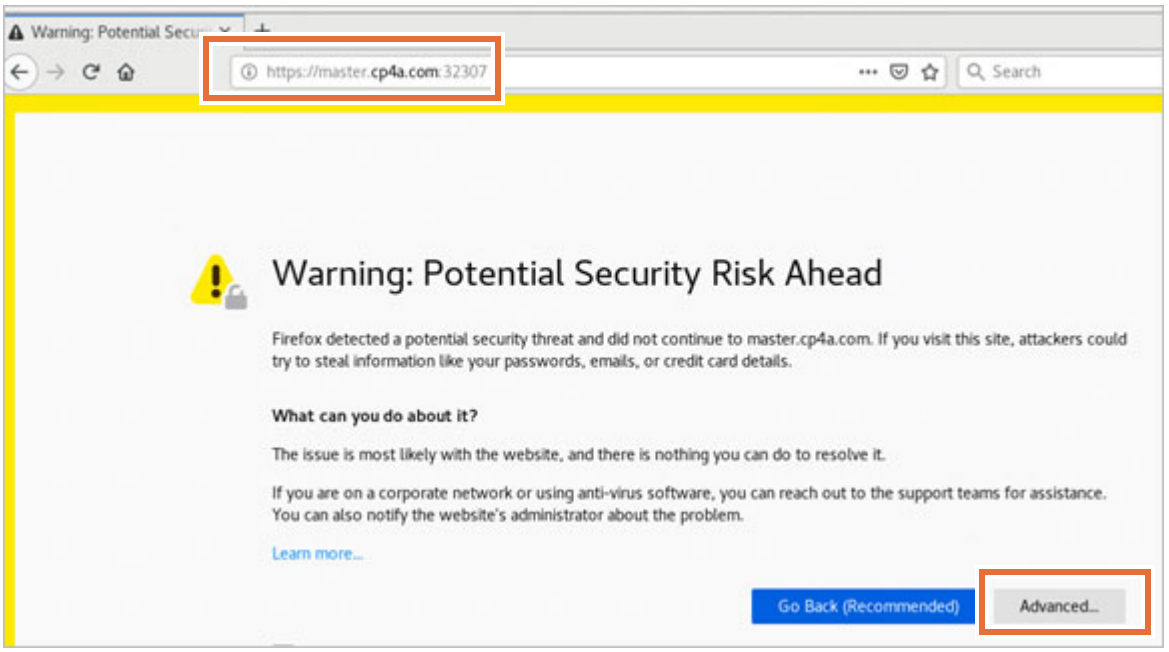
__ a. Find the node port for the decision center. The node port in the view below are the 5 digits between the colon and the forward slash. The node port in your environment is different from the node port in the view below:



__ b. Start the Decision Center console in the browser. Start Firefox and enter the following URL:

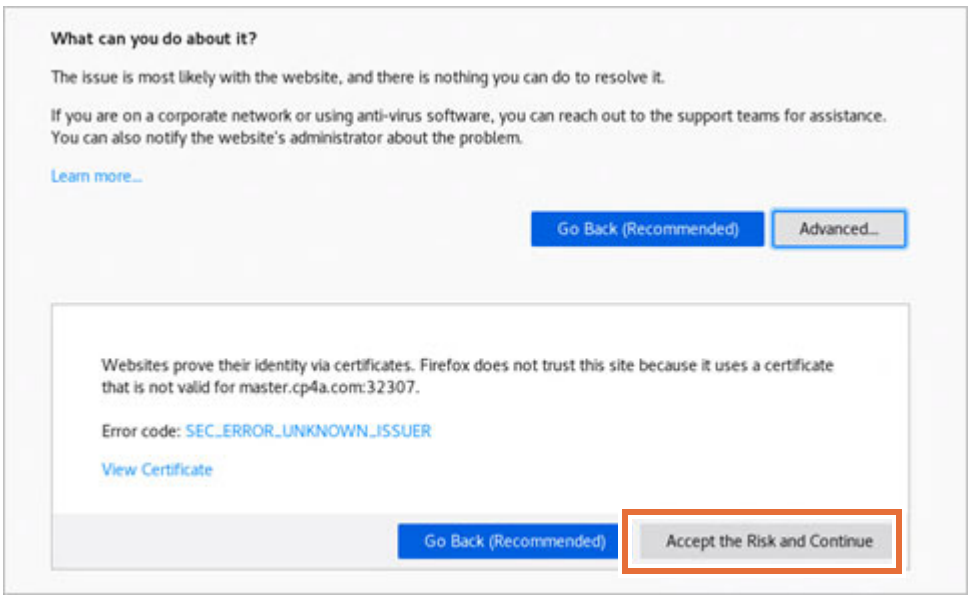
```
https://master.cp4a.com:<your Decision Center NodePort>
```

__ c. Verify that the Firefox warning page is displayed in the browser.

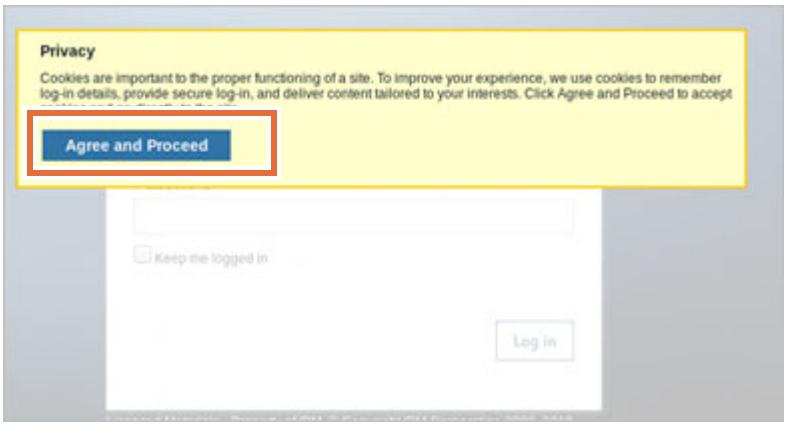


__ d. Click Advanced.

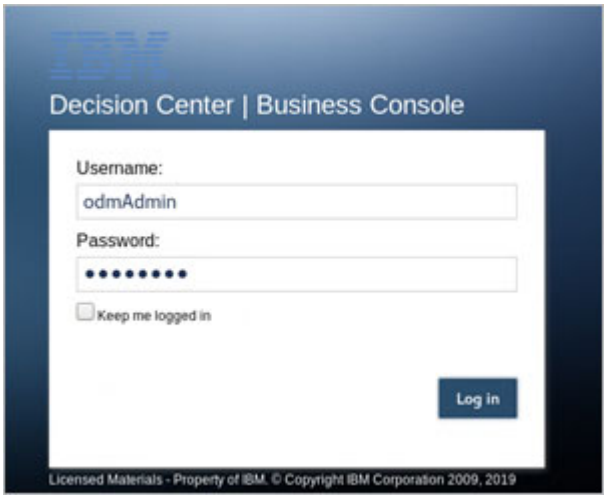
__ e. Scroll down and click Accept the Risk and Continue.



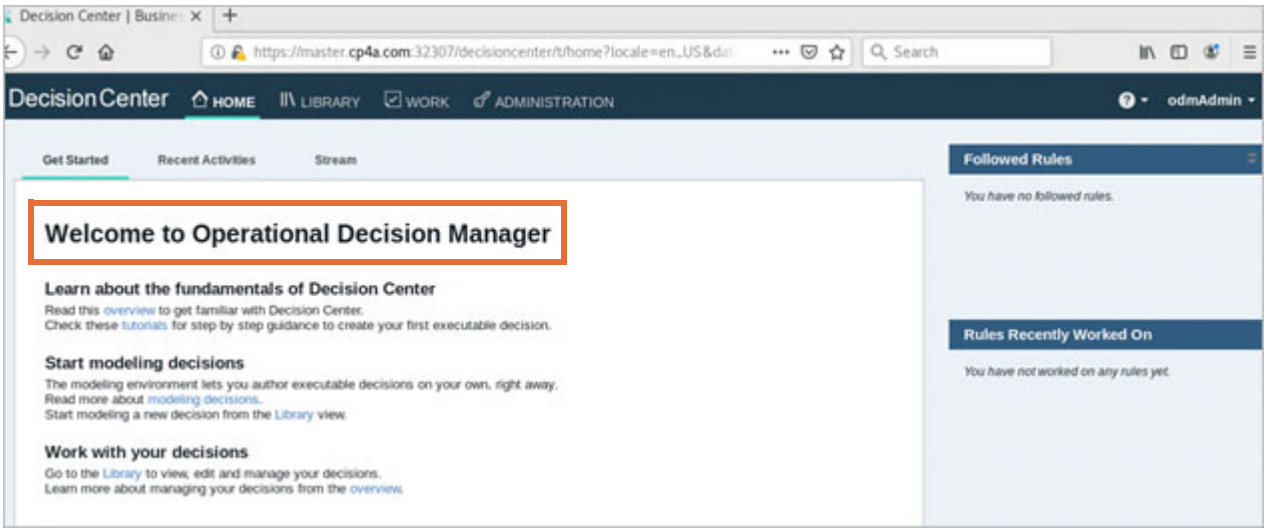
__ f. When the Privacy window is displayed, click Agree and Proceed.



__ g. You are ready to login, when the **Decision Center Business Console** login page is displayed. Log in by entering `odmAdmin` for both the **Username** and the **Password** field.



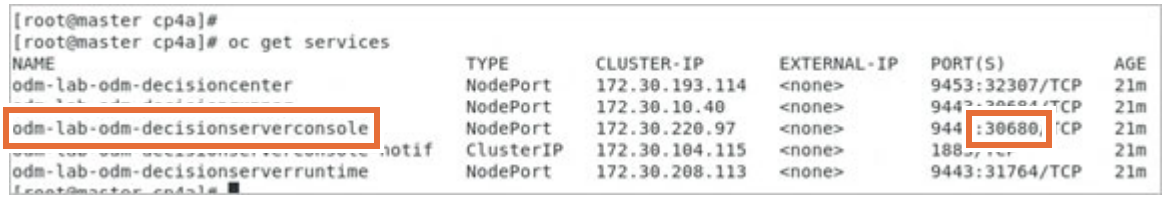
__ h. Make sure that the **Welcome to Operational Decision Manager** page is displayed.



Now that you successfully logged in, you confirmed a successful deployment of the Decision Center console. Next, you verify the other services.

__ 3. Connect to the Rule Execution Server console

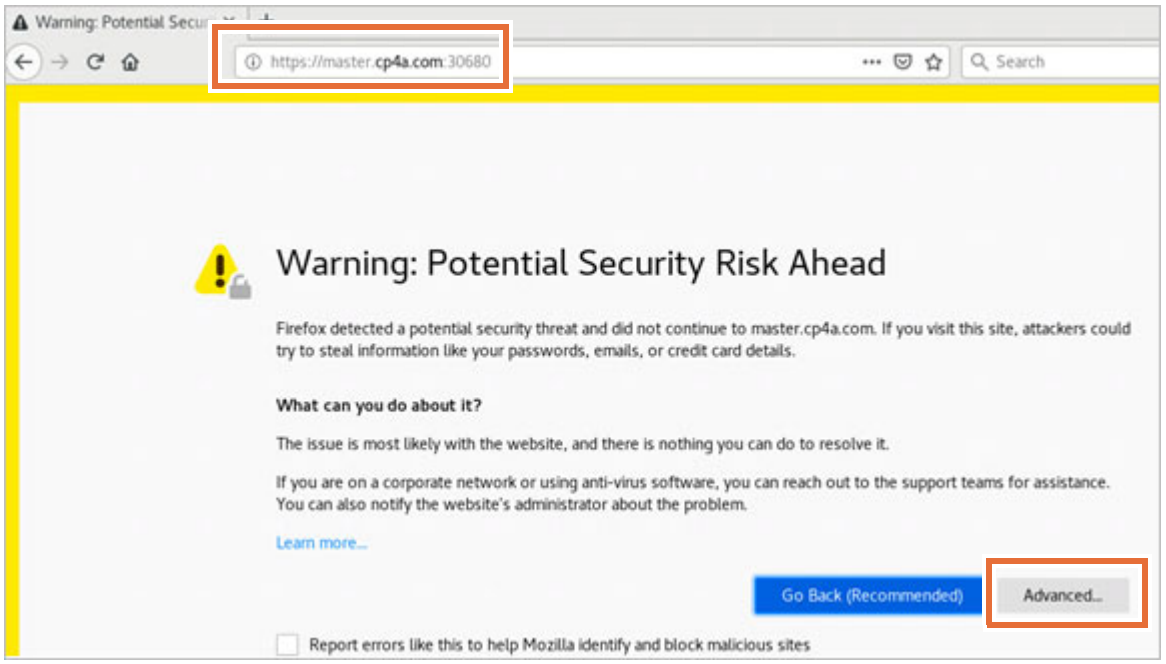
__ a. Find the node port for for the Rule Execution Server. Switch back to the Terminal and look for the name that has **decisionserverconsole** and find the node port. Note that the node port in your environment is different from the node port in the view below:



__ b. Start the Rule Execution Server console in the browser. Start Firefox and enter the following URL:

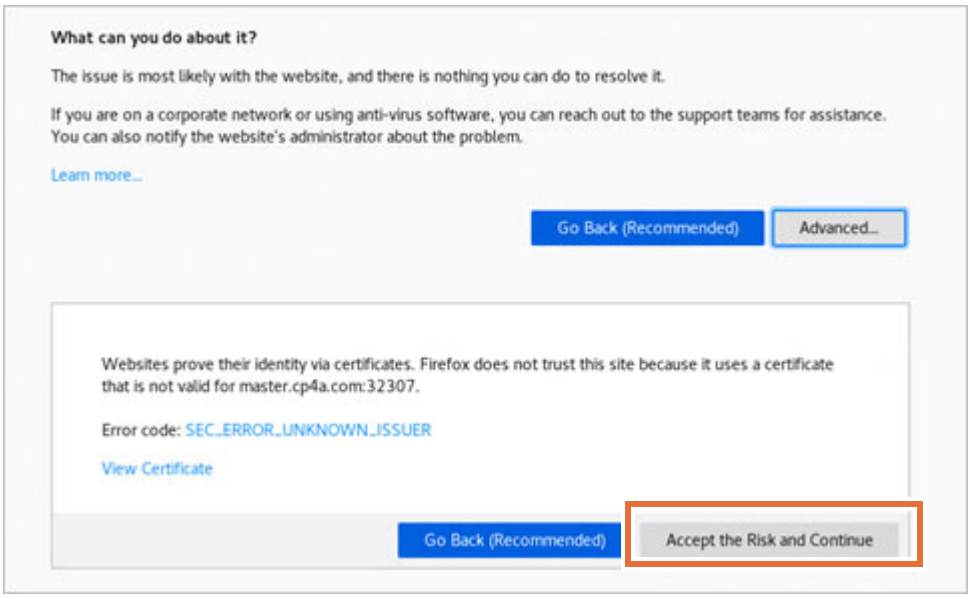
`https://master.cp4a.com:<your Rule Execution Server NodePort>`

__ c. Verify that the Firefox warning page is displayed in the browser.



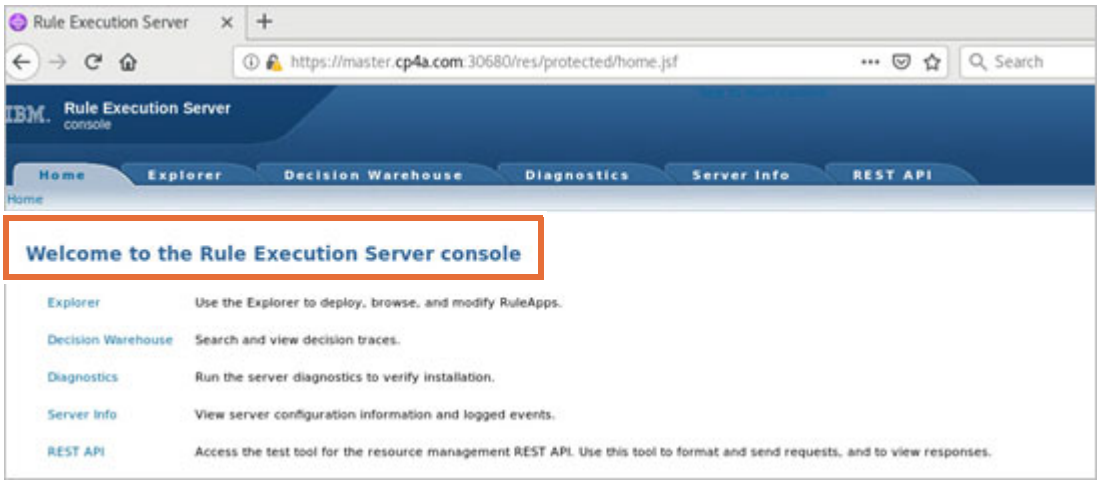
__ d. Click Advanced.

__ e. Scroll down and click Accept the Risk and Continue.



__ f. You are ready to log in, when the **Rule Execution Server console l**ogin page is displayed. Log in by entering `odmAdmin` for both the **User Name** and the **Password** field and click **Sign In**.

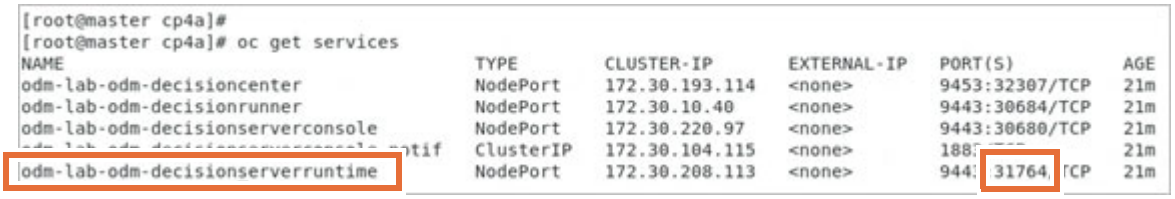
__ g. Verify that the **Welcome to the Rule Execution Server console** page is displayed.



Now that you successfully logged in, you confirmed a successful deployment of the Decision Server console. Next, you verify the other services.

__ 4. Connect to the Decision Server runtime.

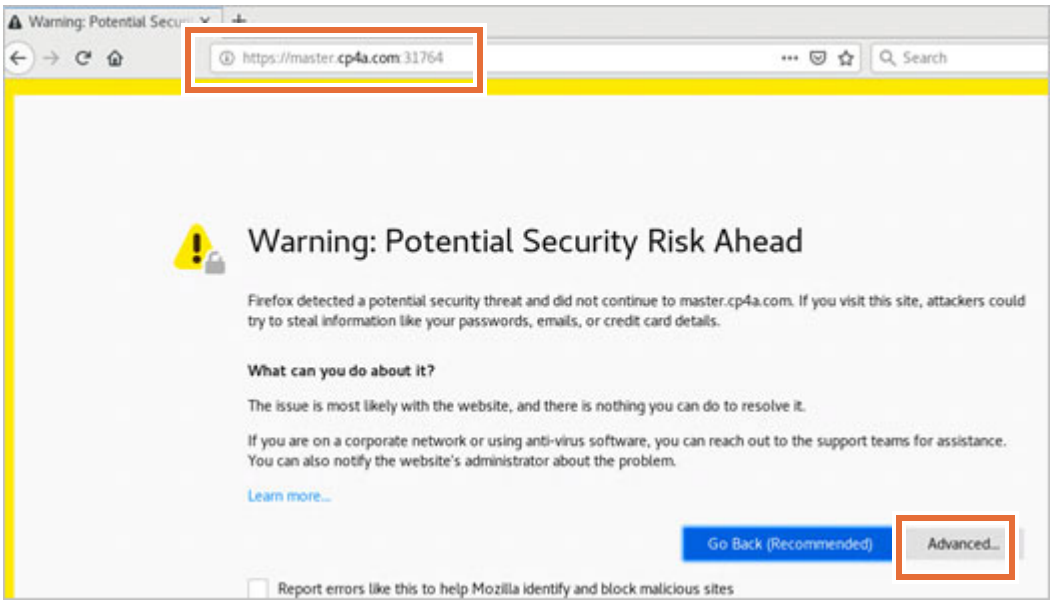
__ a. Find the node port for for the decision center runtime. Note that the node port in your environment is different from the node port in the view below:



__ b. Start the Decision Server runtime in the browser. Start Firefox and enter the following URL:

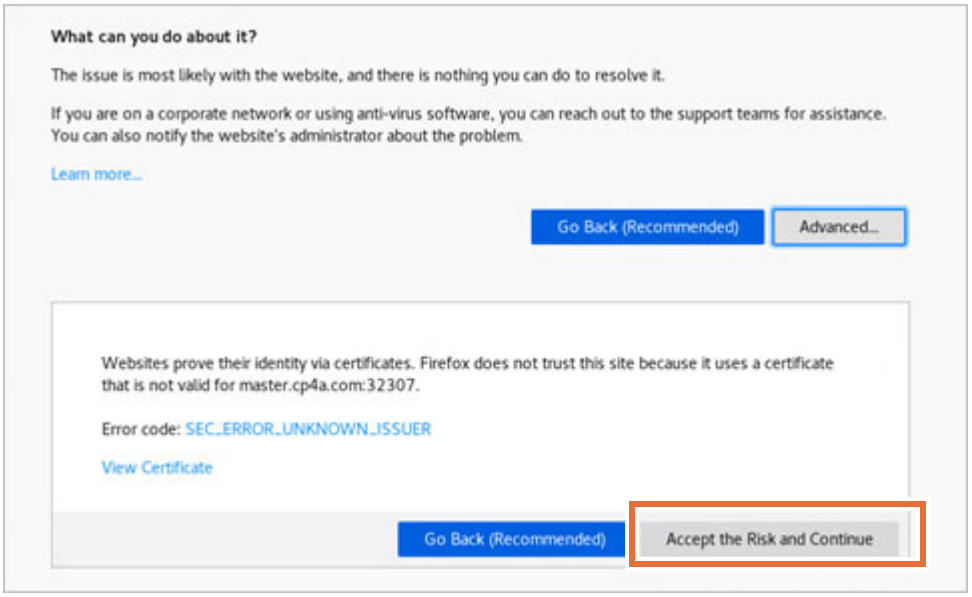
`https://master.cp4a.com:<your Decision Server runtime NodePort>`

__ c. Verify that the Firefox warning page is displayed in the browser.



__ d. Click Advanced.

__ e. Scroll down and click Accept the Risk and Continue.



__ f. The Hosted Transparent Decision Service page is displayed. Note the listed Decision Server version 8.10.2.0 and the patch level.

__ 5. Connect to the Decision Runner.

__ a. Find the node port for the Decision Runner. The node port in your environment is different from the node port in the view below:

```
[root@master cp4a]#
[root@master cp4a]# oc get services
NAME                                    TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
odm-lab-odm-decisionrunner              NodePort    172.30.10.40     <none>        9443:30684/TCP   21m
odm-lab-odm-decisionserverconsole       NodePort    172.30.220.97    <none>        9443:30080/TCP   21m
odm-lab-odm-decisionserverconsole-notif ClusterIP   172.30.104.115   <none>        1883/TCP         21m
odm-lab-odm-decisionserverruntime       NodePort    172.30.208.113   <none>        9443:31764/TCP   21m
```

__ b. Start the Decision Runner in the browser. Start Firefox and enter the following URL:

`https://master.cp4a.com:<your Decision Runner NodePort>`

__ c. Verify that the Firefox warning page is displayed in the browser.

__ d. Click Advanced.

__ e. Scroll down and click Accept the Risk and Continue.

__ f. The Decision Runner page is displayed. Note the listed Decision Runner version 8.10.2.0 and the patch level.

__ 6. Close all the open windows. Before moving to the next exercise, it is a good idea to close all open windows and sessions.

__ a. Enter exit in the terminal to close it. If you had multiple terminal windows open close all of them

__ b. Close the Firefox browser.

__ c. Close any other open windows, including any text editors.

__ d. Close the RDP console window or the browser window that was used to work with the Master VM.

You now completed the successful deployment of ODM containers and verified the connectivity and access to ODM. You do not do any ODM development or other work by using the ODM container. That is the task of an ODM developer or ODM Administrator. For ODM development related training refer to the IBM Training site. This concludes this exercise.

End of exercise

Exercise 2. Deploying the IBM FileNet P8 Content Platform Engine (CPE) container

Estimated time

05:00

Overview

In this exercise, you complete the required steps to prepare, configure, and install IBM Content Platform Engine (CPE) V5.5.3, which is part of IBM Cloud Pak for Automation V19.0.1 on top of a Red Hat OpenShift container platform (RHOCP) V3.11.

Objectives

After completing this exercise, you should be able to:

- Start and shut the lab environment
- Connect to Red Hat OpenShift Container (RHOCP)
- Configure the Persistent Volumes required by Content Platform Engine
- Prepare the database required by Content Platform Engine
- Prepare LDAP required by Content Platform Engine
- Load the Content Platform Engine docker images for Cloud Pak for Automation
- Deploy the Content Platform Engine container
- Verify successful Content Platform Engine deployment
- Connect to the Content Platform Engine container and successfully log in to the Administrative Console for Content Engine (ACCE)

Introduction

The IBM Cloud Pak for Automation (CP4A) offers a software platform to develop, deploy, run, and manage your digital business automation projects by using its capabilities. In this exercise, you deploy the Content Platform Engine (CPE) container.

Requirements

Availability of the lab environment that consists of three Red Hat Enterprise Linux V7.7 virtual machines (VMs) with Db2 Enterprise V11.1.1.1, Open LDAP, and Red Hat OpenShift (RHOCP)

3.11 installed. The required IBM Cloud For Automation (CP4A) software images are already downloaded on a VM.

The completion of Exercise 1 is not a requirement to work on this Exercise 2.

Virtual machine configuration

Each virtual machine is configured with the following specification:

Table 5.

OS	CPUs	RAM	Disk
Red Hat Enterprise Linux (RHEL) 7.7 (64-bit)	8	16 GB	300 GB

List of Servers with Roles

The virtual machines are listed below with their respective roles:

Table 6.

VM	RHCOCP Node type	IP address	Hostname
VM1- OCP master	Master	`10.0.0.1`	`master.cp4a.com`
VM2- OCP compute1	Compute	`10.0.0.2`	`compute1.cp4a.com`
VM3- OCP compute 2	Compute	`10.0.0.3`	`compute2.cp4a.com`

Software Requirements

The following packages are downloaded to the **Master** node:

Table 7.

Software	Folder name
IBM Cloud Pak for Automation images	`/root/labfiles/cp4a`

User IDs and Passwords

The following table contains a list of User ID and password information for this exercise:

Table 8.

Entry Point	User ID	Password
OpenShift web console: `https://master.cp4a.com:8443` or `https://console.cp4a.com:8443`	`admin`	`passw0rd`
Red Hat Linux VM	`root`	`passw0rd`
Db2 Enterprise	`db2admin`	`passw0rd`
LDAP console: https://master.cp4a.com:6443	`cn=admin, dc=ibm,dc =edu`	`passw0rd`
Administrative console for Content Engine (ACCE): `http://master.cp4a.com:<http NodePort>/acce`	`p8admin`	`passw0rd`

Stop

Course updates and errata

A Course Corrections document might be available for this course.

If you are taking the class with an instructor, the instructor can provide this document to you.

If you are taking the course in a self-paced environment, the course corrections document is provided with the other manuals.

To check whether a Course Corrections document exists for this course:

1. Go to the following URL: http://www.ibm.com/developerworks/connect/middleware_edu
2. On the web page, locate and click the **Course Information** category.
3. Find your course in the list and click the link.
4. Click the **Attachments** tab to see whether an errata document exists with updated instructions.
5. To save the file to your computer, click the document link and follow the dialog box prompts.

Exercise instructions

Part 1: Work with the development environment

Before you start working with this exercise, it is important to get familiar with the lab environment. In this section, you learn how to work with the environment. Give special attention to the steps you need to take when you go for a long break or return to the labs after extended period (such as the start of the next day).

Read the following three sections below carefully before starting your environment and connecting to your VM in step 1:

- Understand the various states of the lab environment
- Decide on using either RDP or Browser to connect to your VM
- Copy and paste code in the environments

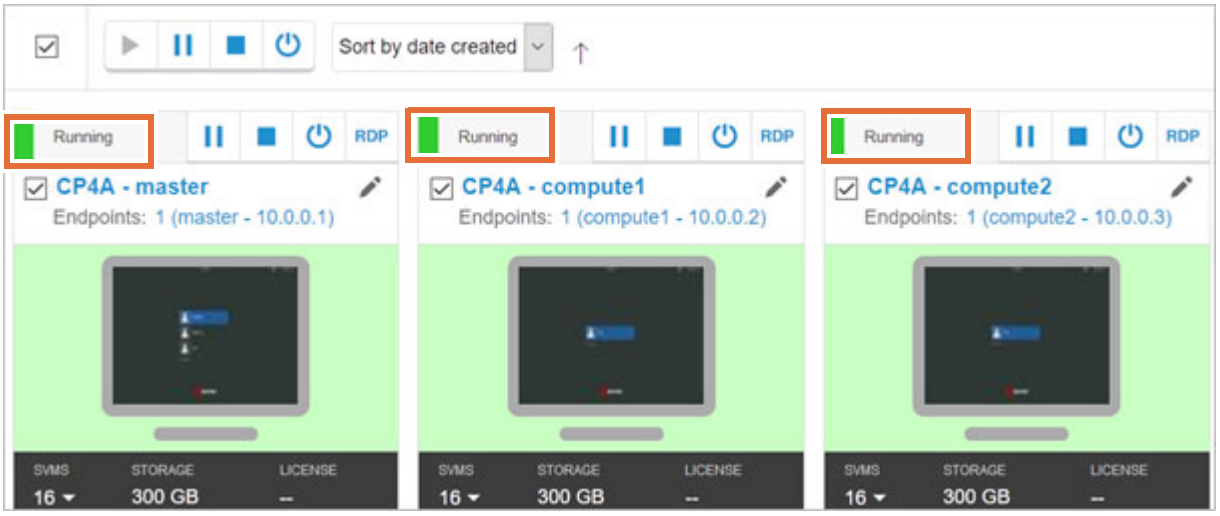
Understand the various states of the lab environment

The instructions below describe the states of the lab environment. A good understanding ensures a pleasant user experience and minimizes any unpredictability in the lab environment.

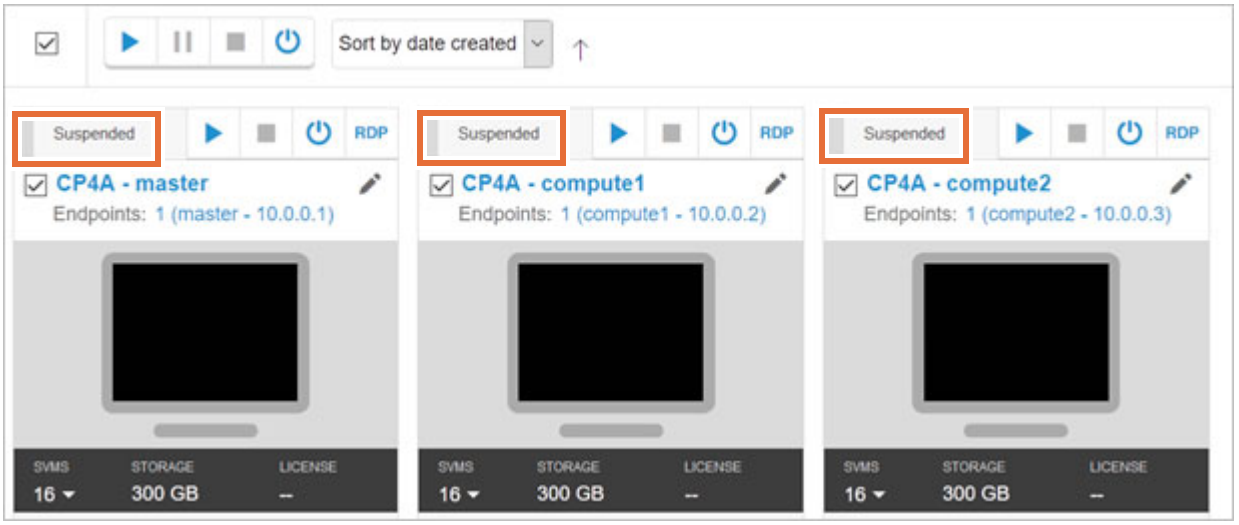
Three RHEL VMs make up your lab environment. Red Hat OpenShift (RHOCP) 3.11 is already installed. The RHOCP cluster consists of one master and two compute nodes. VM1 is the master. This master manages the two compute nodes - compute1 and compute2. It also schedules pods to run on those compute nodes.

Each square box represents one VM. For the three VMs, three square boxes are displayed next to each other in one row. When you click the link to access the lab environment for the first time, the environment can be in any of the three states - Running, Suspended, or Powered off.

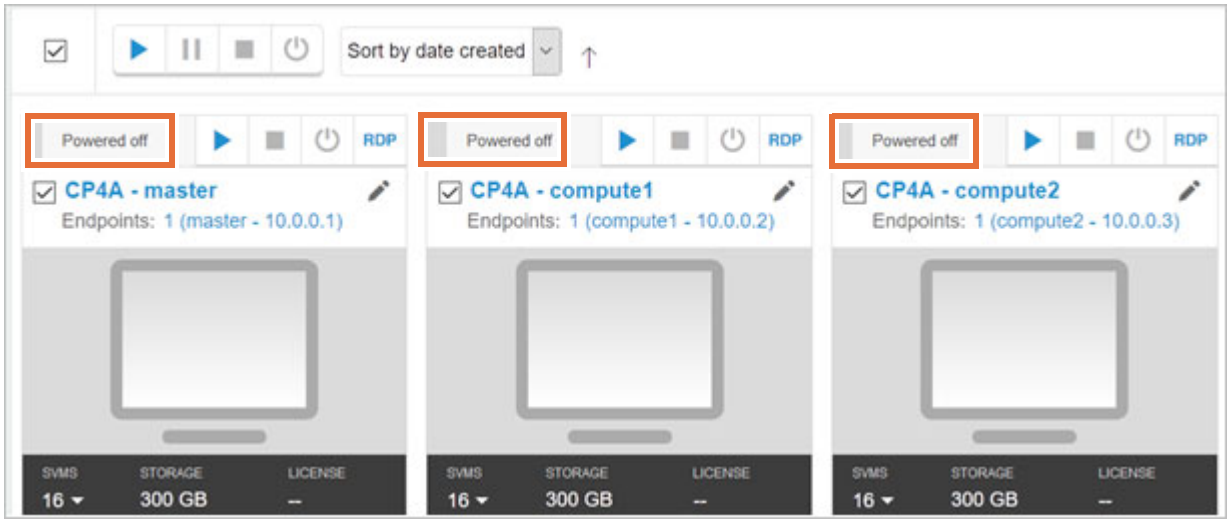
Running state: The environment is running when it displays **Running** for each VM box. To work with the lab environment, all the three VMs must be in a Running state. Before you can work with the labs, always verify that all the three VMs are in a Running state.



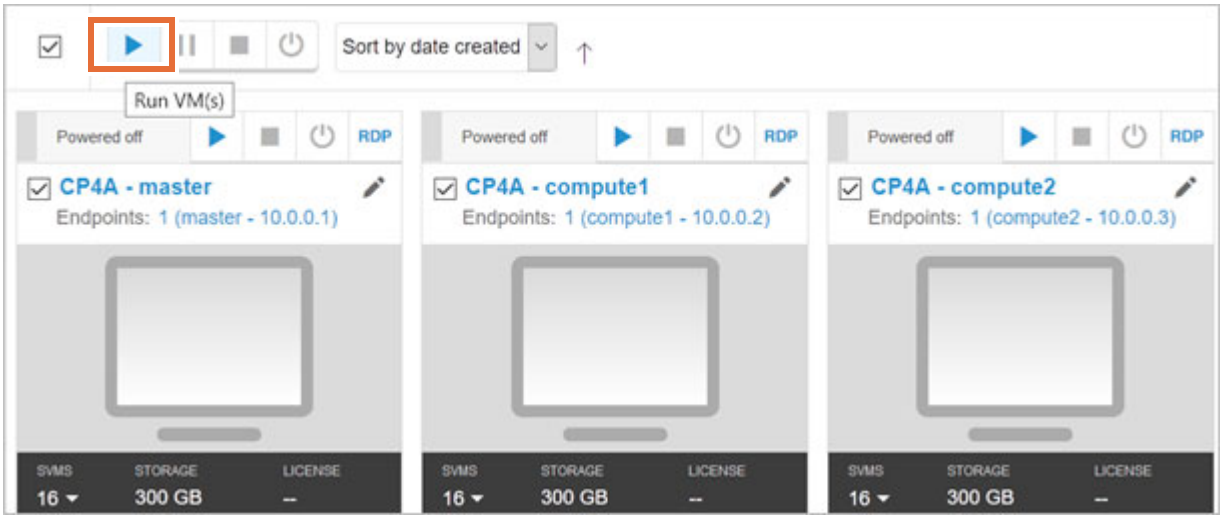
Suspended state: The environment is suspended when it displays **Suspended** for any single VM box. After **2 hours** of inactivity, the environment automatically suspends to conserve resources. Usually all the three VMs are suspended together. You cannot work with the lab environment even if a single VM is in a suspended state. It is because all the three VMs need to communicate with each other.



Powered off state: The environment is powered off or shut down when it displays **Powered off** for any single VM box. Usually all the three VMs are powered off together. You cannot work with the lab environment even if a single VM is in a powered off state since all the three VMs need to communicate with each other.

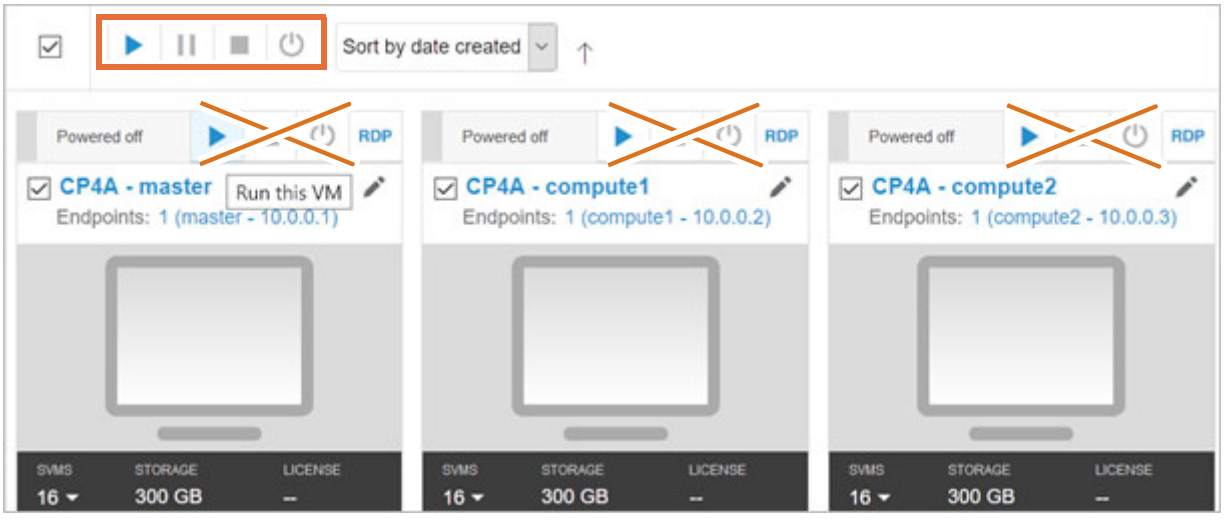


How to switch from a Powered off state to a Running state? If the environment is Powered off and you are ready to work on the labs, then you need to bring the environment to a Running state. You do that by clicking the **Run VM(s)** icon that is displayed at the top to start the environment. Wait about 10 minutes for all the VMs and the services to come up before you are ready to log in.

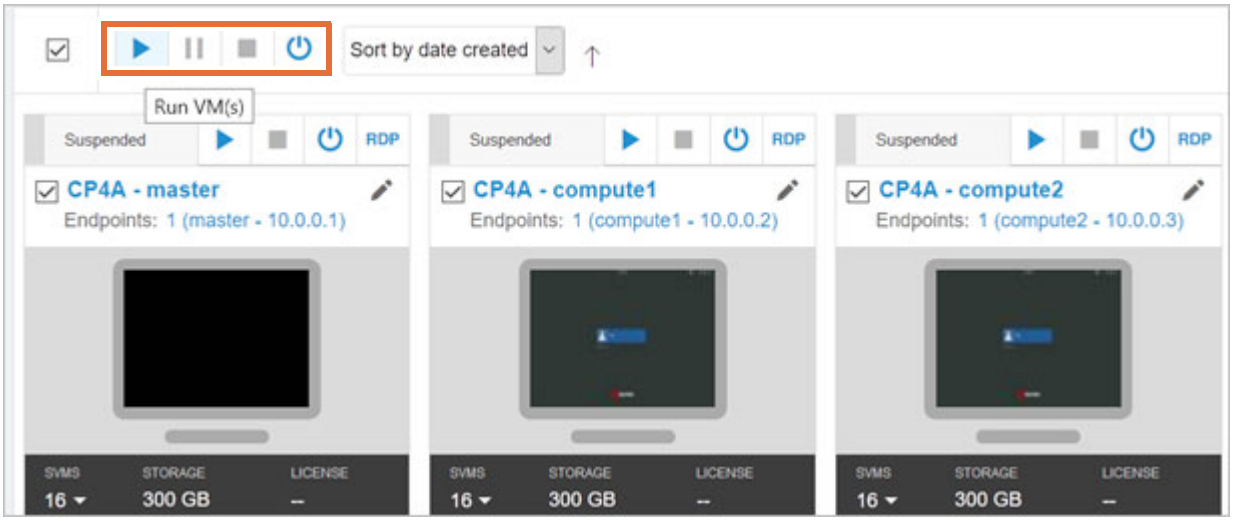


Important

When starting the environment from the Powered off state, make sure to click the **Run VM(s)** icon that is displayed above all the three VMs. This step starts all the three VMs properly with a single click. Do **NOT** click **Run this VM** next to each VM to start them one at a time.



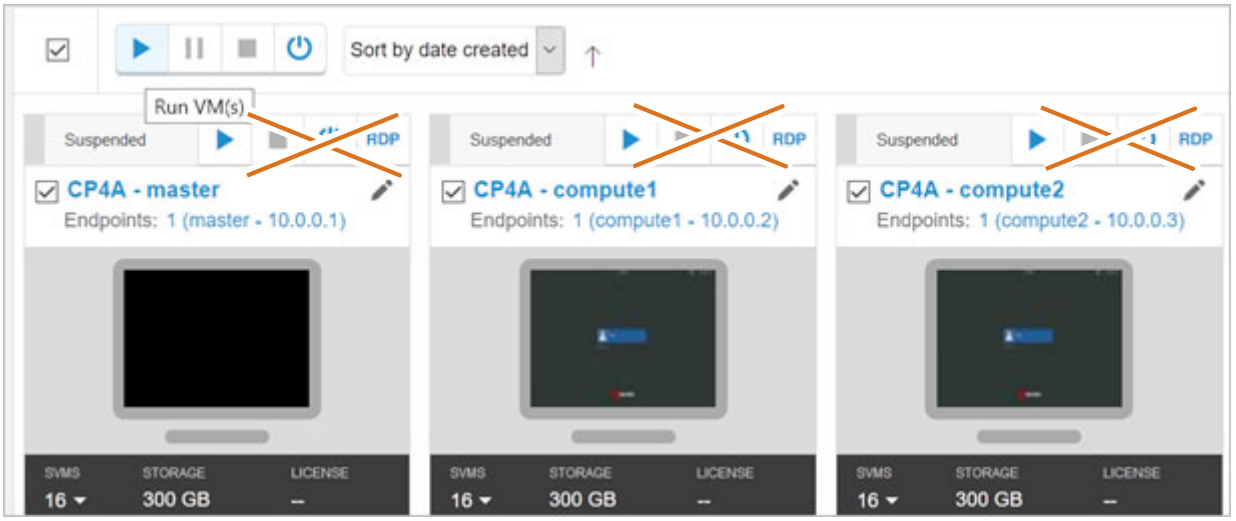
How to switch from a Suspended state to a Running state? When the environment is suspended and you are ready to work on the labs, then you need to bring the environment to a **Running** state. You do that by clicking the **Run VM(s)** icon to start the environment. Wait at least 10 minutes for all the VMs and the services to come up before you are ready to log in.



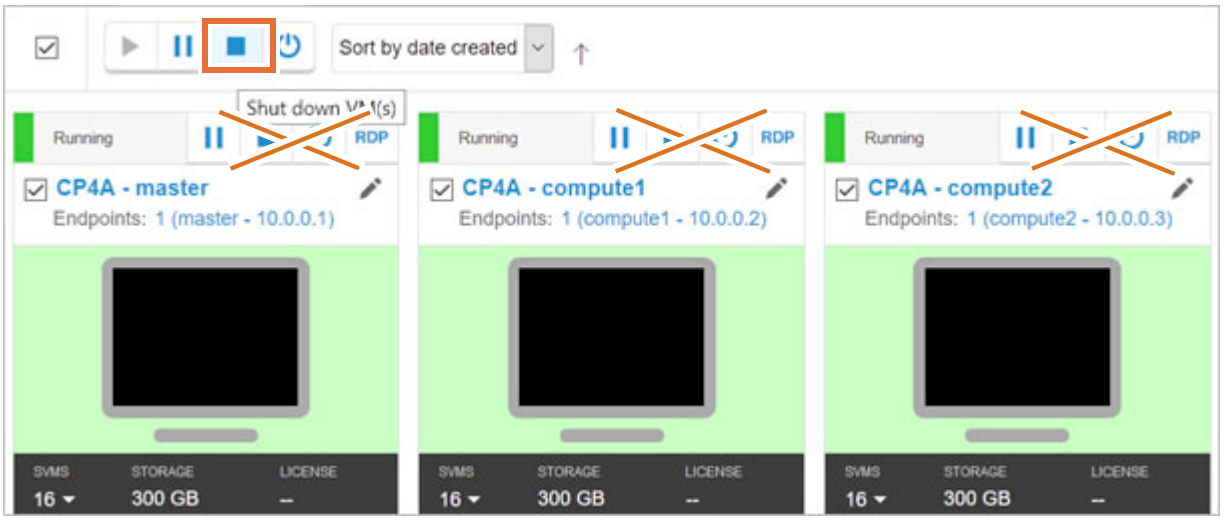
If you log in too quickly, sometimes the VM might not be responsive for a while when starting a VM from a suspended state. **Even though the status changes to Running, give it another 5 minutes before connecting. This wait ensures that all services are up and the VMs are communicating with each other.**

Important

When starting the environment from the Suspended state, make sure to click the **Run VM(s)** icon that is displayed above all the three VMs. This step starts all the three VMs properly with a single click. Do **NOT** click the **Run this VM** next to each VM to start them one at a time.



What steps to take before you go for a break of more than 2 hours or are done for the day? It is a good idea to shut down the environment when you plan to be away for over 2 hours. It prevents the environment from going into the Suspended state. Power off the entire environment by clicking the **Shut down VM(s)** icon that is displayed above all the three VMs. This step stops all the three VMs properly. Do **NOT** click the **Shutdown this VM** next to each VM in an attempt to stop them one at a time as that can cause a VM (usually a compute VM) to hang.



What steps to take after you return to labs from a break of more than 2 hours? When you return after your long break or start work the next day and if you manually shut down the environment as described earlier, you can click the Run VM(s) icon that is displayed at the top of the environment. It starts all the VMs together. Wait at least 10 minutes before logging in to the VM and resuming your lab exercise.

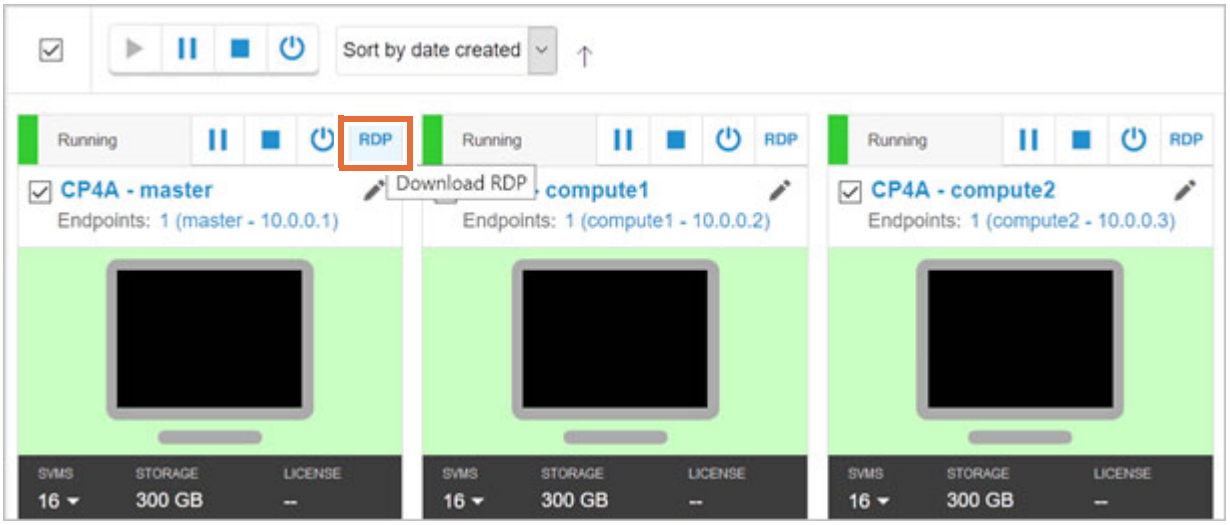
If you forgot to shut down the environment before going for a long break, then your environment state changes to **Suspended** after 2 hours. To bring it to Running state, you click the **Run VM(s)** icon that is displayed at the top of the environment and that starts all the VMs together. Wait at least 10 minutes before logging in to the VM. It is possible that the Master VM is unresponsive and displays unexpected results and behavior after logging in. If that happens, go back to your earlier steps that you were doing before the break and verify that they are working correctly first before continuing with your lab. If things still do not appear to be working correctly, then shut down the entire environment by clicking the **Shut down VM(s)** icon that is displayed above all the three VMs. After you have shut down the environment, you can then start the environment by clicking **Run VM(s)**. which starts all the three VMs together. This process ensures a clean start of the VMs. Continue working with the labs where you left off before.

Decide on using either RDP or Browser to connect to your VM

You can connect to any of the three VMs by either using your browser or Remote Desktop Protocol (RDP). If you open RDP or browser sessions for each of the three VMs, make sure to close the ones not being used to avoid working with the wrong VM. You work with the Master VM only in the exercise.

Accessing the VM by using the browser: A browser is the quickest and easiest way to access and use your VM's desktop. You can start the browser by clicking the thumbnail image of the VM in the square box.

Accessing the VM by using RDP: By default, RDP is enabled in all of your VMs. To use RDP, click the RDP icon in the upper right corner of the image thumbnail as displayed below.



The first time that you click the RDP icon for a VM, a window opens asking what Firefox should do with this file. Leave the **Open With Remote Desktop Connection default** option selected and click **OK** to open the VM desktop.

Information

If you use a browser to access the VM, you might notice a possible mouse lag sometimes. You can try both ways to connect to a VM and choose whichever one works best for you

Copy and paste code in the environment

Copy and paste is tricky in the environment. It is a good idea to enter all the commands manually in the command line as instructed the exercise. This ensures correct entry of characters.

When entering commands in the Linux Terminal, always enter the commands in a single line

However, if you want to copy and paste some of the longer commands, you need to copy that command first into a text file, such as Notepad on your local desktop. **If the command displays in multiple lines in Notepad, make sure to format it to a single line.** Then, select and copy from Notepad before pasting into the Linux Terminal. If the paste does not work, then copy from Notepad file one more time and the second time the paste should work. During testing, it was found that repeating this step twice usually worked.

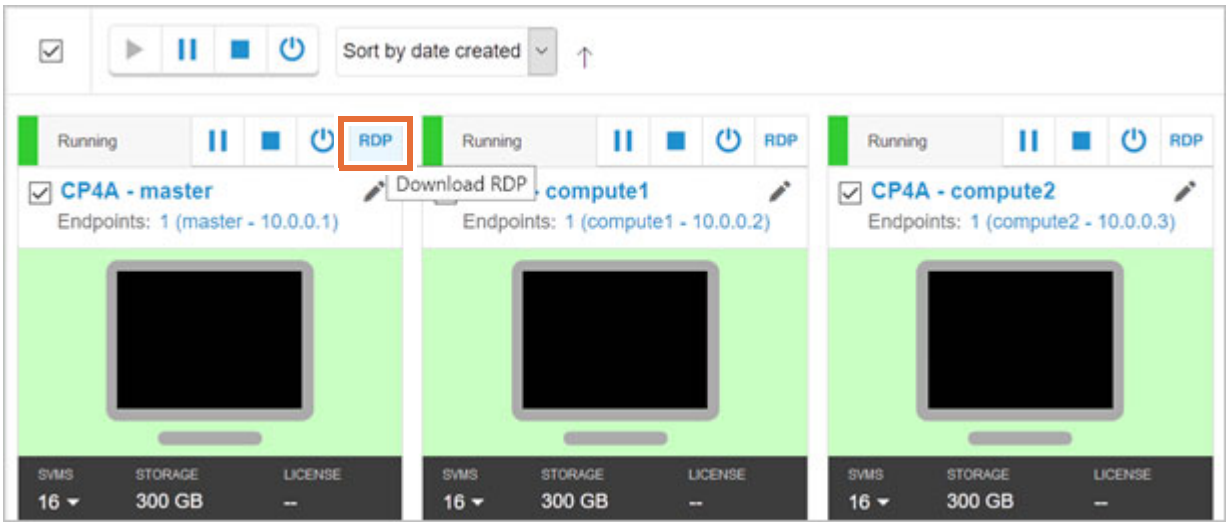
Connect to the Master VM in the environment

Now that you read the description of various states of the lab environment, you are ready to start and work with it.

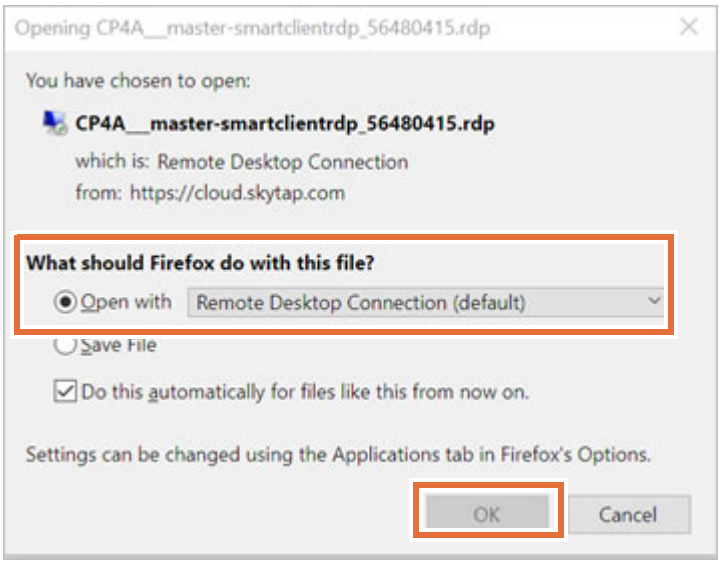
__ 1. Start your lab environment as mentioned in the previous instructions and make sure that it is in a Running state. All three VMs must be in the Running state.

__ 2. Use RDP or Browser to connect to the **Master** VM.

__ a. Connect with the **Master** VM by either clicking the thumbnail image of the VM to start in the browser or click the RDP icon in the upper right corner of the **Master** image thumbnail as displayed below.



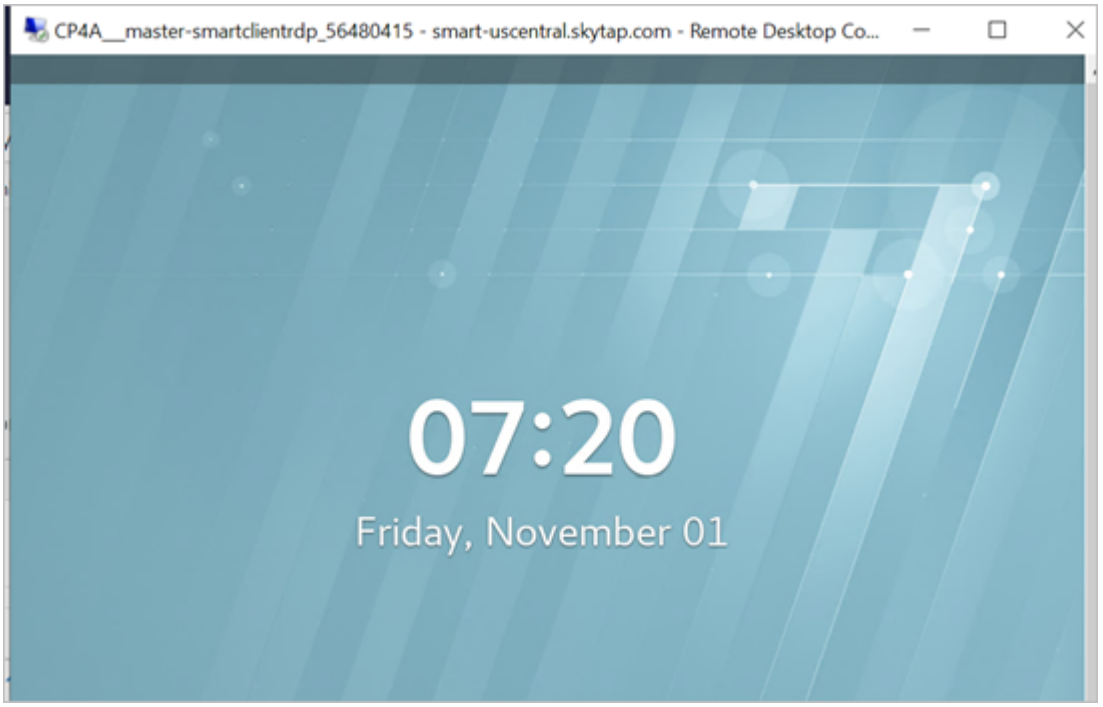
__ b. The first time that you click the RDP icon for the VM, a window opens asking what Firefox should do with this file. Leave the **Open With Remote Desktop Connection default** option selected and click **OK** to open the VM desktop.



You are now ready to work with the Master VM. Go to the next step to learn how to log in to the VM.

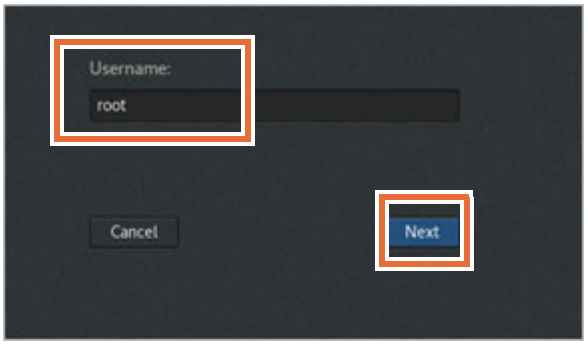
__ 3. Log in to the **Master** VM. Remember, that you work with the **Master** VM throughout the exercise. All the commands that you run in this exercise, you run on the **Master** VM. Optionally, you can connect to the compute VMs if you want to explore.

__ a. Regardless of the way you connect to the Master VM (either through the Browser or RDP), the time and date. is displayed on the desktop the first time you connect. Click the Enter key anywhere on the desktop to display the login screen.

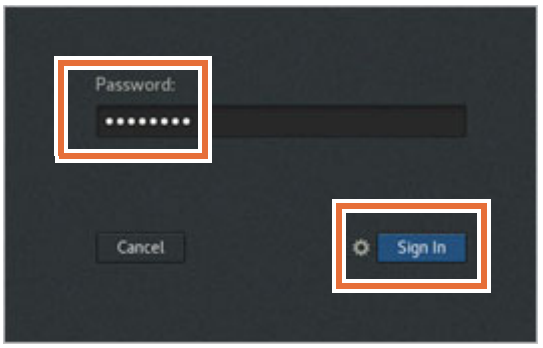


__ b. Click **Not listed**.

__ c. Enter `root` in the Username field and click **Next**.



__ d. Enter `passw0rd` in the Password field and click **Sign In**.



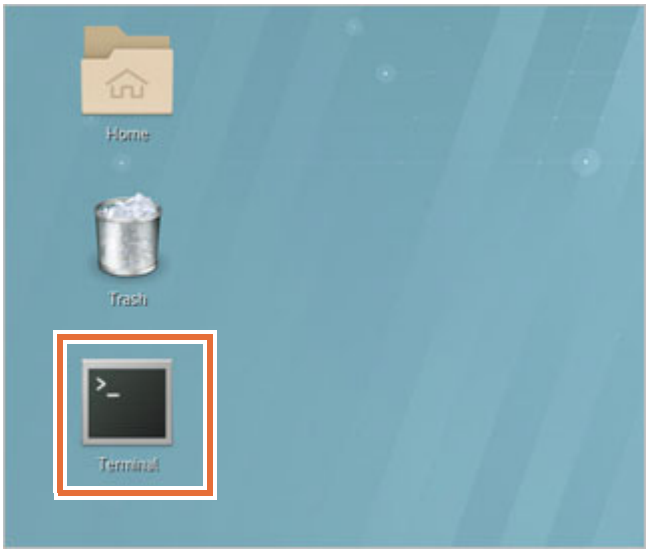
__ e. You are now successfully logged in to the Master VM.

Part 2: Create the Persistent Volumes and Persistent Volume Claims that are required by Content Platform Engine

The Content Platform Engine (CPE) container requires several persistent volume (PV) and persistent volume claims (PVC) in RHOCP. The Kubernetes persistent volume framework provides a mechanism for containers to request and use persistent storage. To avoid data loss, these services are configured to use persistent volumes. You create several PV and PVC in this section.

__ 1. Log in to Red Hat OpenShift by using the command line

__ a. Double-click the Terminal shortcut on the desktop to open it.



__ b. In the terminal that opens, login to the OpenShift cluster. The OpenShift console URL is `https://console.cp4a.com:8443` and the username/password `is admin/passw0rd`.

```
oc login https://console.cp4a.com:8443 -u admin -p passw0rd
```

__ c. Verify that the Login successful message is displayed.

__ d. Examine the list of projects that is displayed. The asterisk * next to a project name indicates the project that is being used in the current session. Depending on whether you completed Exercise 1 on ODM before working on this FileNet lab, the current project that is displayed is either odm-lab or the default project. You do not work with either the odm-lab or the default project but instead create a new one in a later step.

__ 2. Check the status of the nodes.

__ a. Verify that the master and the two compute nodes are in the ready state..

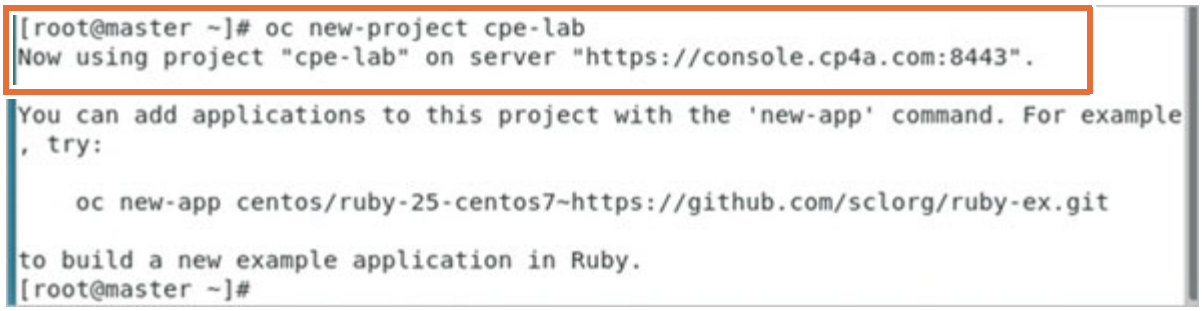
```
oc get nodes
```

__ b. Verify that three nodes are listed and the status is Ready for all the nodes.

__ 3. Create an OpenShift project for CPE

__ a. Create a project called **cpe-lab** for Content Platform Engine.

```
oc new-project cpe-lab
```



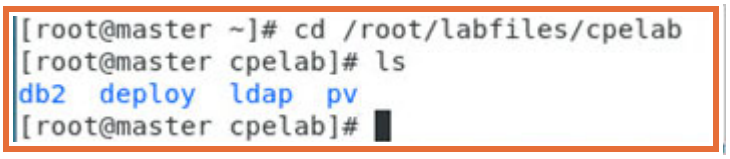
You are now ready to use this new project.

__ 4. Create the PV and PVC required by CPE.

__ a. Go to the cpelab folder that is located under `/root/labfiles/cpelab` directory.

```
cd /root/labfiles/cpelab
```

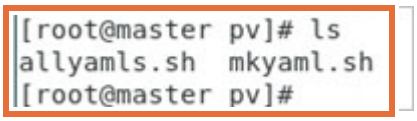
```
ls
```



__ b. The script to create the persistent volumes and persistent volume claims is located under the `pv` folder. Go to that pv folder.

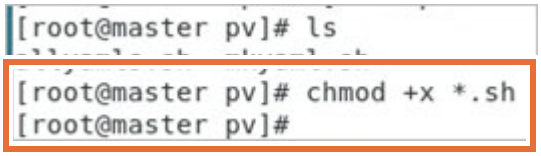
```
cd pv
```

```
ls
```



__ c. Two scripts are available under the `pv` folder. Make these two scripts executable before running them.

```
chmod +x *.sh
```



__ d. Examine all the two scripts under the `pv` folder by using an editor of your choice. Optionally, you can double-click the **Home** folder on the desktop, Go to the `/root/labfiles/cpelab/pv` and then right-click each script to open with Text Editor. The `allyamls.sh` script makes a reference to the `mkyaml.sh` script that in turn creates a new `create_pvs.yaml` file. Then, you run the newly created `create_pvs.yaml` file that creates the persistent volumes and persistent volume claims. Examine the two scripts and do not make any changes. When done examining the scripts, close them.

__ e. Run the `allyamls.sh` script.

```
./allyamls.sh
```

```
[root@master pv]# ./allyamls.sh
creating /nfs/cpe-lab/configDropins_overrides
creating /nfs/cpe-lab/log
creating /nfs/cpe-lab/FileNet
creating /nfs/cpe-lab/lib_bootstrap
creating /nfs/cpe-lab/temp
creating /nfs/cpe-lab/icmrules
creating /nfs/cpe-lab/asa
run exportfs -arv
then run oc apply -f create_pvs.yaml
[root@master pv]#
```

__ f. As indicated by the script, notify the NFS server to update the exported directories:

```
exportfs -arv
```

```
[root@master pv]# exportfs -arv
exporting *:/nfs
[root@master pv]#
```

__ g. Verify that the `create_pvs.yaml` file is created under the pv folder when you ran the `allyamls.sh` script earlier.

```
ls
```

```
[root@master pv]# ls
allyamls.sh  create_pvs.yaml  mkyaml.sh
[root@master pv]#
```

__ h. You are now ready to apply the created yaml file with the generated definitions of all persistent volumes and persistent volume claims.

```
oc apply -f create_pvs.yaml
```

```
[root@master pv]# oc apply -f create_pvs.yaml
persistentvolume/cpe-lab-icp-cfgstore-pv created
persistentvolumeclaim/cpe-lab-icp-cfgstore-pvc created
persistentvolume/cpe-lab-icp-logstore-pv created
persistentvolumeclaim/cpe-lab-icp-logstore-pvc created
persistentvolume/cpe-lab-icp-fnlogstore-pv created
persistentvolumeclaim/cpe-lab-icp-fnlogstore-pvc created
persistentvolume/cpe-lab-icp-bootstrapstore-pv created
persistentvolumeclaim/cpe-lab-icp-bootstrapstore-pvc created
persistentvolume/cpe-lab-icp-textextstore-pv created
persistentvolumeclaim/cpe-lab-icp-textextstore-pvc created
persistentvolume/cpe-lab-icp-icmrulesstore-pv created
persistentvolumeclaim/cpe-lab-icp-icmrulesstore-pvc created
persistentvolume/cpe-lab-icp-filestore-pv created
persistentvolumeclaim/cpe-lab-icp-filestore-pvc created
[root@master pv]#
```

__ i. Query the persistent volume claims to verify that several pv and pvc are created.

```
oc get pvc
```

__ j. Examine the several columns in the output. The `NAME` column lists all the persistent volume claims that are created. The `VOLUME` column displays all the persistent volumes created. The `STATUS` column shows that they are all bound. The remaining columns describe the other configuration that is required for the deployment.

```
[root@master pv]# oc get pvc
NAME                                STATUS   VOLUME                              CAPACITY   ACCESS MODES   STORAGECLASS                    AGE
cpe-lab-icp-bootstrapstore-pvc      Bound    cpe-lab-icp-bootstrapstore-pv       1Gi        RWX            cpe-lab-icp-bootstrapstore      4m
cpe-lab-icp-cfgstore-pvc            Bound    cpe-lab-icp-cfgstore-pv             1Gi        RWX            cpe-lab-icp-cfgstore            4m
cpe-lab-icp-filestore-pvc           Bound    cpe-lab-icp-filestore-pv            10Gi       RWX            cpe-lab-icp-filestore           4m
cpe-lab-icp-fnlogstore-pvc          Bound    cpe-lab-icp-fnlogstore-pv           5Gi        RWX            cpe-lab-icp-fnlogstore          4m
cpe-lab-icp-icmrulesstore-pvc       Bound    cpe-lab-icp-icmrulesstore-pv        1Gi        RWX            cpe-lab-icp-icmrulesstore       4m
cpe-lab-icp-logstore-pvc            Bound    cpe-lab-icp-logstore-pv             5Gi        RWX            cpe-lab-icp-logstore            4m
cpe-lab-icp-textextstore-pvc        Bound    cpe-lab-icp-textextstore-pv         1Gi        RWX            cpe-lab-icp-textextstore        4m
```

You now completed the successful creation of PV and PVCs

Part 3: Prepare the database required by Content Platform Engine

In this section, you create the required database and configure them for storing the Global Configuration Database (GCD) and the Object Store data.

__ 1. Create the required database

__ a. Switch to the `/root/labfiles/cpelab/db2` folder and list the contents of the folder. Several scripts to create databases are available there.

```
cd /root/labfiles/cpelab/db2
```

```
ls
```

```
[root@master pv]# cd /root/labfiles/cpelab/db2
[root@master db2]# ls
create_gcddb.sh  DB2JCCDriver.xml       GCDDB.xml  setfilenetworkload.sh
create_os1db.sh  db2_list_databases.sh  OS1DB.xml
[root@master db2]#
```

__ b. Make the database scripts executable, and copy them to the home directory of the Db2 instance user, who can access the database.

```
chmod +x *.sh
```

```
cp *.sh /home/db2admin
```

```
[root@master db2]# chmod +x *.sh
[root@master db2]# cp *.sh /home/db2admin
[root@master db2]#
```

__ c. Switch to the db2admin directory and list the contents to verify that the scripts were copied successfully.

```
cd /home/db2admin
```

```
ls
```

```
[root@master db2]# cd /home/db2admin
[root@master db2admin]# ls
create_gcddb.sh  db2admin               setfilenetworkload.sh
create_os1db.sh  db2_list_databases.sh  sqllib
[root@master db2admin]#
```

__ d. Run the `setfilenetworkload.sh` script for the Db2 configuration.

```
su - db2admin ./setfilenetworkload.sh
```

```
[root@master db2admin]# su - db2admin ./setfilenetworkload.sh
Last login: Wed Oct 23 10:27:23 PDT 2019 on pts/0
DB2_WORKLOAD=FILENET_CM
DB2_SKIPINSERTED=YES [DB2_WORKLOAD]
DB2_OPTPROFILE=YES [DB2_WORKLOAD]
DB2_USE_ALTERNATE_PAGE_CLEANING=YES [DB2_WORKLOAD]
DB2_MINIMIZE_LISTPREFETCH=YES [DB2_WORKLOAD]
DB2_EVALUNCOMMITTED=YES [DB2_WORKLOAD]
DB2COMM=TCPIP
DB2AUTOSTART=YES
[root@master db2admin]#
```

__ e. Create the GCDDB database by running the `create_gcddb.sh` script.

```
su - db2admin ./create_gcddb.sh
```

```
[root@master db2admin]# su - db2admin ./create_gcddb.sh
Last login: Sun Nov 10 12:55:48 PST 2019 on pts/0
SQL1024N  A database connection does not exist.  SQLSTATE=08003
SQL1013N  The database alias name or database name "GCDDB    " could not be
found.  SQLSTATE=42705
Creating the database, patience please
DB20000I  The CREATE DATABASE command completed successfully.

   Database Connection Information

 Database server        = DB2/LINUXX8664 11.1.1.1
 SQL authorization ID   = DB2ADMIN
 Local database alias   = GCDDB

DB20000I  The SQL command completed successfully.
DB20000I  The UPDATE DATABASE CONFIGURATION command completed successfully.
DB20000I  The UPDATE DATABASE CONFIGURATION command completed successfully.
DB20000I  The SQL command completed successfully.
[root@master db2admin]#
```

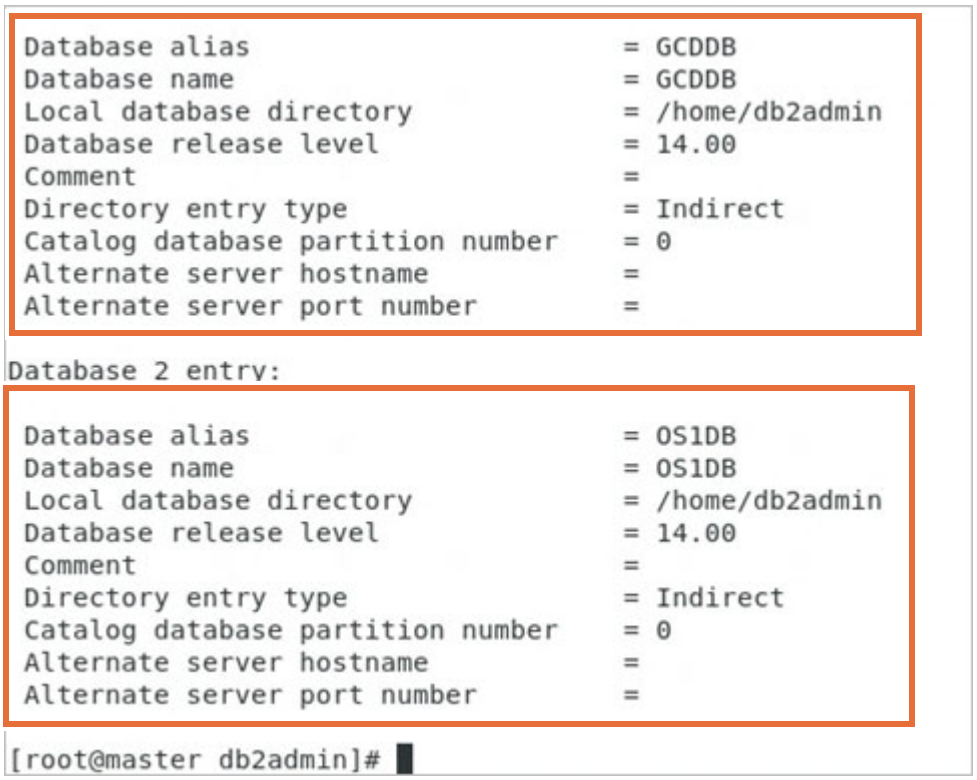
__ f. Create the OS1DB database by running the `create_os1db.sh` script.

```
su - db2admin ./create_os1db.sh
```

```
root@master db2admin]# su - db2admin ./create_os1db.sh
ast login: Sun Nov 10 12:56:50 PST 2019 on pts/0
QL1024N  A database connection does not exist.  SQLSTATE=08003
QL1013N  The database alias name or database name "OS1DB    " could not be
ound.  SQLSTATE=42705
reating the database, patience please
320000I  The CREATE DATABASE command completed successfully.

  Database Connection Information

Database server        = DB2/LINUXX8664 11.1.1.1
SQL authorization ID   = DB2ADMIN
Local database alias   = OS1DB

320000I  The SQL command completed successfully.
320000I  The UPDATE DATABASE CONFIGURATION command completed successfully.
320000I  The UPDATE DATABASE CONFIGURATION command completed successfully.
320000I  The SQL command completed successfully.
root@master db2admin]#
```

__ g. For verification, run the last `db2_list_databases.sh` script. Verify that the GCDDB and the OS1DB databases are listed. If you also completed Exercise 1, then the ODMDB database is also listed.

```
su - db2admin ./db2_list_databases.sh
```



You completed the creation of the database.

__ 2. Add the java archives for accessing the Db2 server.

__ a. Copy the Db2 Client files into the persistent volume directory configDropins_overrides.

```
cp /opt/ibm/db2/V11.1/java/db2jcc* /nfs/cpe-lab/configDropins_overrides
```

__ b. Switch back to the `/root/labfiles/cpelab/db2` folder.

```
cd /root/labfiles/cpelab/db2
```

__ c. Copy the XML files to the configDropins_overrides directory. Review the XML files to examine the contents.

```
cp *.xml /nfs/cpe-lab/configDropins_overrides
```

You are now ready to complete the LDAP configuration.

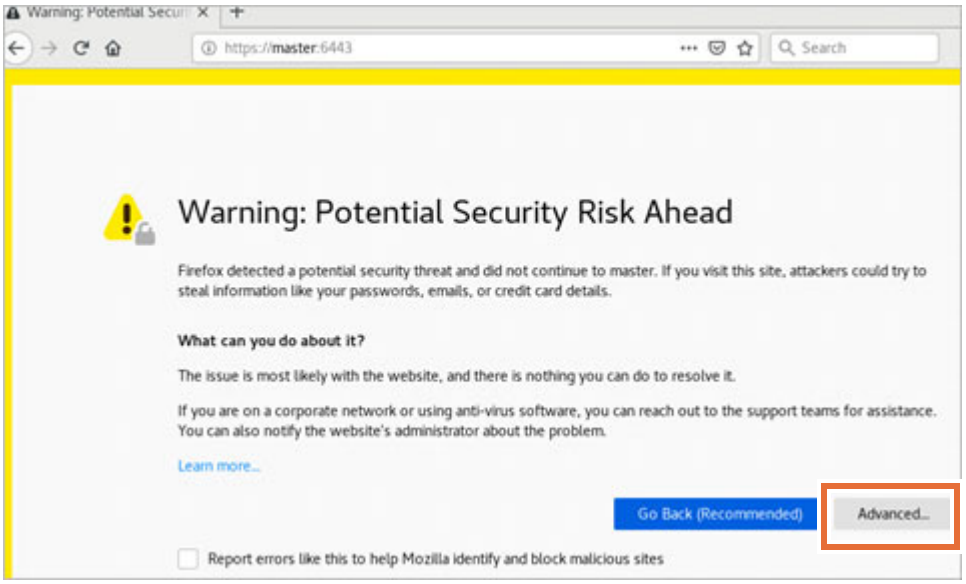
Part 4: Prepare LDAP required by Content Platform Engine

In this section, you examine LDAP users and then make the XML available to the CPE container. The openLDAP docker container is already installed and configured on the Master VM.

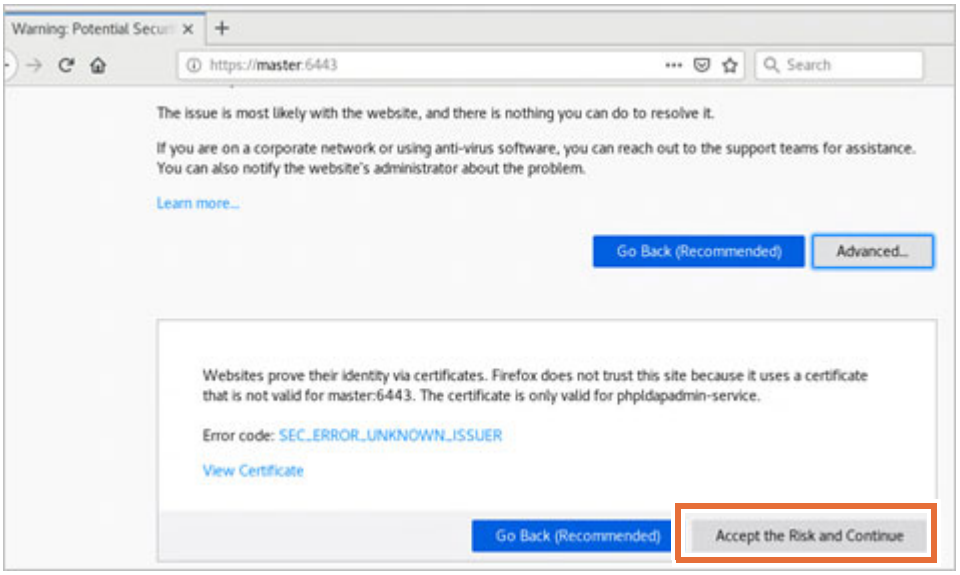
__ 1. Log in to the OpenLDAP Administration console.

__ a. Start a Firefox session and click the OpenLDAP administration console shortcut. Optionally, you can enter `https://console.cp4a.com:6443` in the URL field and press the Enter key.

__ b. When the Security warning message is displayed, click **Advanced**.

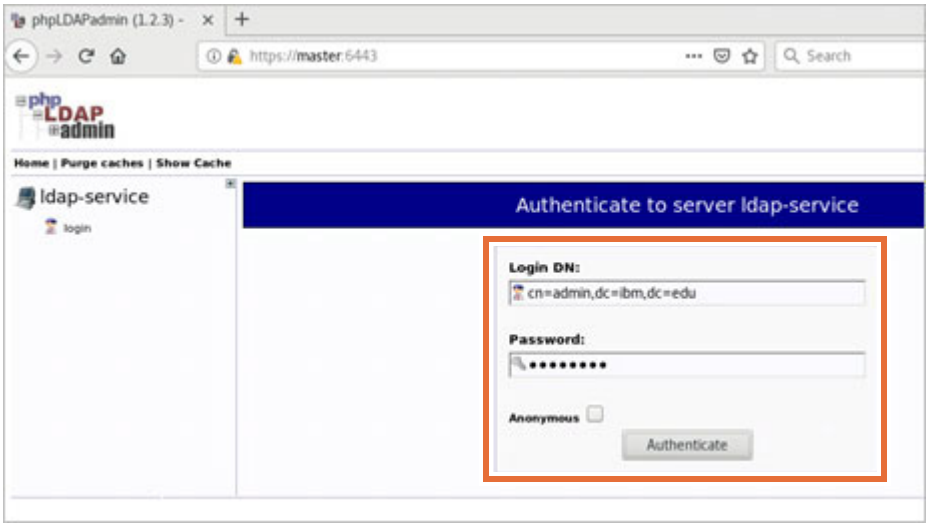


__ c. Scroll down and click **Accept Risk and Continue**.

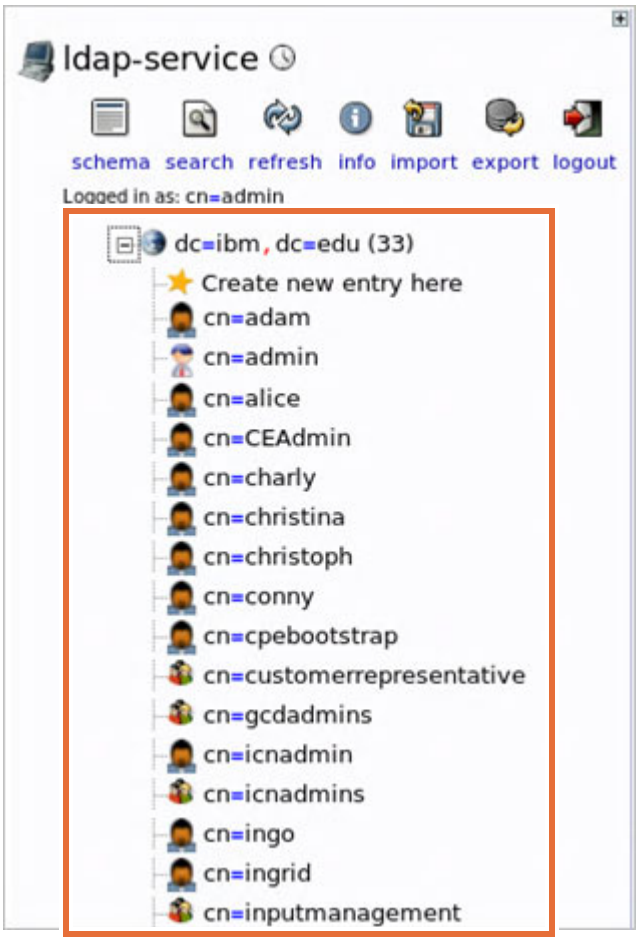


__ d. Click **login**.

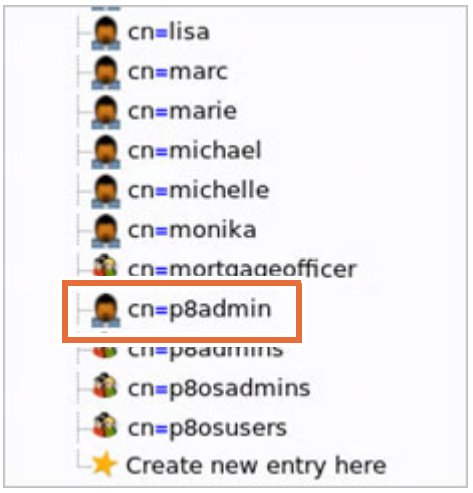
__ e. Enter `cn=admin,dc=ibm,dc=edu` in the **Login DN** field and enter `passw0rd` in the **Password** field. Click **Authenticate**.



__ f. Expand the users on the left pane to examine the list of users.



__ g. Scroll down the list and click the **p8admin** user.



__ h. Note that the description field in the right pane for the user p8admin. It defines the user as the primary p8 administrator. The FileNet developer uses this p8admin user to work with the cpe container. You log in to the deployed container later in the exercise.

__ i. To make the LDAP available to WebSphere Liberty in the CPE Container, an XML configuration file is needed in the persistent volume referring to the directory configDropins_overrides. Switch back to the open terminal and go to the `/root/labfiles/cpelab/ldap` folder. Then, list the contents of the folder.

```
cd /root/labfiles/cpelab/ldap
```

```
ls
```

```
[root@master db2]# cd /root/labfiles/cpelab/ldap
[root@master ldap]# ls
ldap_TDS.xml  p8users.ldif
[root@master ldap]#
```

__ j. The p8users.ldif file consists of LDAP users that were already populated in openLDAP for this exercise. The ldap_TDS.xml configuration file makes the users available to the CPE container. Feel free to examine the two files. Do not make any changes to the XML file.

__ k. Copy the XML to the configDropins_overrides directory.

```
cp ldap_TDS.xml /nfs/cpe-lab/configDropins_overrides
```

```
[root@master ldap]# cp ldap_TDS.xml /nfs/cpe-lab/configDropins_overrides
[root@master ldap]#
```

Part 5: Load the Content Platform Engine docker images for Cloud Pak for Automation

The installation of Cloud Pak for Automation requires downloading and deploying the appropriate docker images. The CP4A images that are needed for the labs are already downloaded and available under the `/root/labfiles/cp4a` folder. In this section, you load the required CPE images for CP4A.

__ 1. Load the CPE docker images

__ a. Identify the images for ICP4A that are located under `/root/labfiles/cp4a` folder

```
cd /root/labfiles/cp4a
```

```
ls
```

```
[root@master ldap]# cd /root/labfiles/cp4a
[root@master cp4a]# ls
ibm-dba-contentservices-3.0.0.tgz  ICP4A19.0.1-ecm.tgz  loadimages.sh
ibm-odm-prod-2.2.0.tgz             ICP4A19.0.1-odm.tgz
[root@master cp4a]#
```

__ b. Create a directory under `/root/labfiles/cp4a` and call it `ecm`.

```
mkdir ecm
```

__ c. Go to the new ecm directory.

```
cd ecm
```

__ d. Unpack the CP4A distribution file for the containers of the Enterprise Content Management system. The file name is `ICP4A19.0.1-ecm.tgz`.

```
tar xvfz ../ICP4A19.0.1-ecm.tgz
```

__ e. It takes few minutes for all the images to be extracted. The last file is a large archive and takes a little extra time. Verify that 5 images are extracted to the ecm folder.

```
[root@master ecm]# tar xvfz ../ICP4A19.0.1-ecm.tgz
images/11f11b46d1bb531fa5f40938aba118fa42052c6f4e655d0c3733a33f990e2e27.tar.gz
tar: images/11f11b46d1bb531fa5f40938aba118fa42052c6f4e655d0c3733a33f990e2e27.tar.gz: impla
usibly old time stamp 1969-12-31 16:00:00
images/ea952b6d8dd2af79b4212910a26a8e6d3b2aead89f2e2e4816ce56e3fe0ffe56.tar.gz
tar: images/ea952b6d8dd2af79b4212910a26a8e6d3b2aead89f2e2e4816ce56e3fe0ffe56.tar.gz: impla
usibly old time stamp 1969-12-31 16:00:00
images/1953557e582185f5d01cc4b56c9eac3fdfc41d7173f470daea11a67c0c198e89.tar.gz
tar: images/1953557e582185f5d01cc4b56c9eac3fdfc41d7173f470daea11a67c0c198e89.tar.gz: impla
usibly old time stamp 1969-12-31 16:00:00
images/cdc6ff5b83a9e777d72344e3a57e0997968325f7c2f4153954be6dd000d08ff1.tar.gz
tar: images/cdc6ff5b83a9e777d72344e3a57e0997968325f7c2f4153954be6dd000d08ff1.tar.gz: impla
usibly old time stamp 1969-12-31 16:00:00
images/13d98681077f3e92ad0cfb05a4d2cc8e981907aa7d5f5082e37ad0d09221bfa7.tar.gz
tar: images/13d98681077f3e92ad0cfb05a4d2cc8e981907aa7d5f5082e37ad0d09221bfa7.tar.gz: impla
usibly old time stamp 1969-12-31 16:00:00
manifest.json
tar: manifest.json: implausibly old time stamp 1969-12-31 16:00:00
manifest.yaml
tar: manifest.yaml: implausibly old time stamp 1969-12-31 16:00:00
[root@master ecm]#
```

__ a. Examine to the contents of the ecm folder and then go to the images folder to verify that five images are located under that images folder.

```
ls
```

```
cd images
```

```
ls
```

```
[root@master ecm]# ls
images  manifest.json  manifest.yaml
[root@master ecm]# cd images
[root@master images]# ls
11f11b46d1bb531fa5f40938aba118fa42052c6f4e655d0c3733a33f990e2e27.tar.gz
13d98681077f3e92ad0cfb05a4d2cc8e981907aa7d5f5082e37ad0d09221bfa7.tar.gz
1953557e582185f5d01cc4b56c9eac3fdfc41d7173f470daea11a67c0c198e89.tar.gz
cdc6ff5b83a9e777d72344e3a57e0997968325f7c2f4153954be6dd000d08ff1.tar.gz
ea952b6d8dd2af79b4212910a26a8e6d3b2aead89f2e2e4816ce56e3fe0ffe56.tar.gz
[root@master images]#
```

__ b. Switch back to the ecm folder.

```
cd ..
```

```
[root@master images]# cd ..
[root@master ecm]#
```

__ c. Create a login token for the internal docker registry to connect to the internal docker registry.

```
oc whoami -t
```

```
[root@master ecm]# oc whoami -t
cumYLdlVHA-K16v2vJskL3HVUoM7rHFo6dyfiBVhWjE
[root@master ecm]#
```

__ d. Log in to the internal docker registry by using an authentication token. Replace the token variable with the token that you created in the previous step. Make sure to include the token in quotation marks (single or double quotation marks). Do not copy and paste the command from the pdf because the remote VM does not interpret the quotation marks correctly.

```
docker login docker-registry.default.svc:5000 -u admin -p '<token>'
```

```
[root@master ecm]# docker login docker-registry.default.svc:5000 -u admin -p 'cumYLdlVHA-K
16v2vJskL3HVUoM7rHFo6dyfiBVhWjE'
Login Succeeded
[root@master ecm]#
```

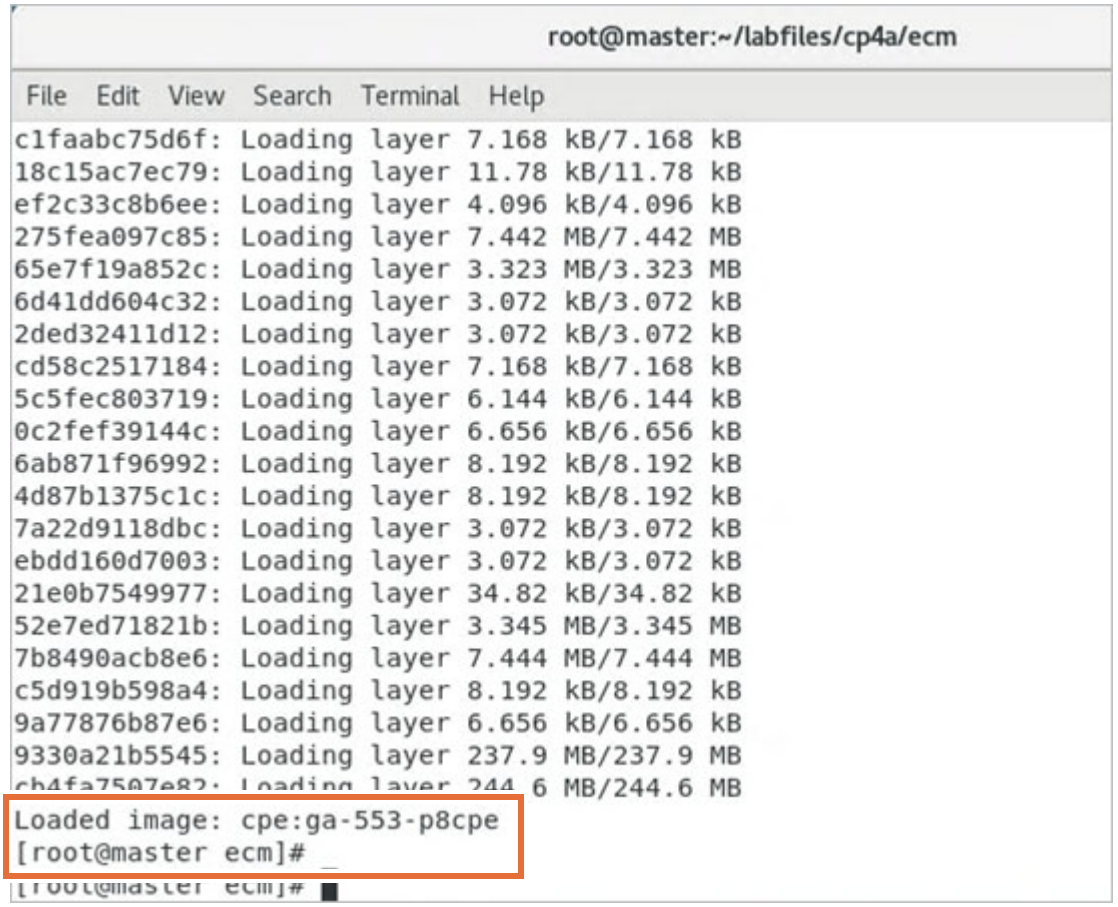
__ e. You are now ready to push the CPE images to the internal docker registry. Before you load the registry, you examine the `manifest.json` file that is included under the `/root/labfiles/cp4a/ecm` folder. Open the `manifest.json` file using an editor of your choice. Optionally, you can double-click the **Home** folder on the desktop, go to `/root/labfiles/cp4a/ecm` and then right-click `manifest.json` to open with Text Editor. Examine the file and find the name of the container images for the Content Platform Engine. Copy the image name between the quotation marks (Do not copy the quotation marks). Do not make any changes. You load that image to the docker registry in the next step.


```
{
  "manifest-revision": "1.0",
  "charts": null,
  "images": [
    {
      "image": "cpe",
      "tag": "ga-
      "archive": images/11f11b46d1bb531fa5f40938aba118fa42052c6f4e655d0c3733a33f990e2e27.tar.gz
    },
    {
      "image": "css",
      "tag": "ga-553-p8css",
      "archive": "images/ea952b6d8dd2af79b4212910a26a8e6d3b2aead89f2e2e4816ce56e3fe0ffe56.tar.gz"
    },
    {
      "image": "cmis",
      "tag": "ga-304-cmis-if007",
      "archive": "images/1953557e582185f5d01cc4b56c9eac3fdfc41d7173f470daea11a67c0c198e89.tar.gz"
    },
    {
      "image": "extshare",
      "tag": "ga-306-es",
      "archive": "images/cdc6ff5b83a9e777d72344e3a57e0997968325f7c2f4153954be6dd000d08ff1.tar.gz"
    },
    {
      "image": "crs",
      "tag": "ga-553-p8cgql",
      "archive": "images/13d98681077f3e92ad0cfb05a4d2cc8e981907aa7d5f5082e37ad0d09221bfa7.tar.gz"
    }
  ]
}
```


__ f. Load the CPE image and push it to the internal docker registry.

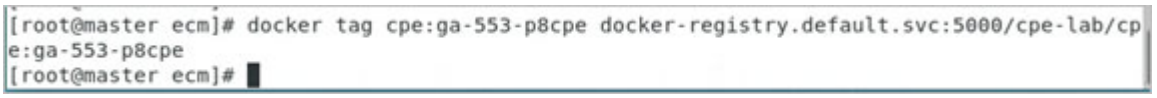
```
docker load < images/11f11b46d1bb531fa5f40938aba118fa42052c6f4e655d0c3733a33f990e2e27.tar.gz
```

__ g. Wait until the image is completely loaded into the docker registry. The command outputs the name of the image which it loads.



__ h. Using the docker tag command, give the container an additional tag pointing to the RHOCP docker repository.

```
docker tag cpe:ga-553-p8cpe docker-registry.default.svc:5000/cpe-lab/cpe:ga-553-p8cpe
```



__ i. Push the image to the RHOCP docker registry. It takes several minutes for the images to load.

```
docker push docker-registry.default.svc:5000/cpe-lab/cpe:ga-553-p8cpe
```

```
[root@master ecm]# docker push docker-registry.default.svc:5000/cpe-lab/cpe:ga-553-p8cpe
The push refers to a repository [docker-registry.default.svc:5000/cpe-lab/cpe]
cb4fa7507e82: Preparing
9330a21b5545: Preparing
9a77876b87e6: Preparing
c5d919b598a4: Preparing
7b8490acb8e6: Preparing
52e7ed71821b: Preparing
21e0b7549977: Preparing
ebdd160d7003: Preparing
7a22d9118dbc: Preparing
4d87b1375c1c: Preparing
6ab871f96992: Preparing
0c2fef39144c: Preparing
5c5fec803719: Preparing
cd58c2517184: Preparing
cb4fa7507e82: Pushed
6d41dd604c32: Pushed
65e7f19a852c: Pushed
275fea097c85: Pushed
ef2c33c8b6ee: Pushed
```

__ j. To finish the configuration of the files in the configDropins_overrides directory, assign the files to the user id 50001, and the group id 50000.

```
cd /nfs/cpe-lab/configDropins_overrides/
```

```
chown 50001:50000 *
```

```
chmod 755 *
```

```
e813ccb40331: Pushed
35265ad107f5: Pushed
4765a2d3f85d: Pushed
ga-553-p8cpe: digest: sha256:123e4e25581331da651ab2b552fa6f27a18f2752dfdd8c49842c71c7bd404
[root@master ecm]# cd /nfs/cpe-lab/configDropins_overrides/
[root@master configDropins_overrides]# chown 50001:50000 *
[root@master configDropins_overrides]# chmod 755 *
[root@master configDropins_overrides]#
```

__ k. Verify the permission and group change displays correctly.

```
ls -l
```

```
[root@master configDropins_overrides]# ls -l
total 7420
-rwxr-xr-x. 1 50001 50000 3905812 Nov 10 13:05 db2jcc4.jar
-rwxr-xr-x. 1 50001 50000     212 Nov 10 13:08 DB2JCCDriver.xml
-rwxr-xr-x. 1 50001 50000 3668626 Nov 10 13:05 db2jcc.jar
-rwxr-xr-x. 1 50001 50000    1534 Nov 10 13:05 db2jcc_license_cu.jar
-rwxr-xr-x. 1 50001 50000    1015 Nov 10 13:08 GCDDB.xml
-rwxr-xr-x. 1 50001 50000     617 Nov 10 13:26 ldap_TDS.xml
-rwxr-xr-x. 1 50001 50000    1048 Nov 10 13:08 OS1DB.xml
[root@master configDropins_overrides]#
```

You are now ready to deploy the CPE container.

Part 6: Deploy the Content Platform Engine container

For deployment of the CPE container, two deployment methods are supported, deployment using a yaml file, and deployment using a helm chart. In this exercise, you deploy by using the Helm chart. It is available in the `cpelab/deploy` folder.

An archive with the helm chart is also available on `https://github.com/icp4a/cert-kubernetes/blob/19.0.1/CONTENT/helm-charts/ibm-dba-contentservices-3.0.0.tgz`.

__ 1. Prepare for deployment.

__ a. Go to the deploy folder and view its content.

```
cd /root/labfiles/cpelab/deploy
```

```
ls
```

```
[root@master deploy]# ls
ibm-dba-contentservices-3.0.0.tgz  values.yaml
[root@master deploy]#
```

__ b. The archive with the helm chart in github also contains the values.yaml file, which is the only file of the Helm chart that needs to be customized. To make deployment easier, a version of the `values.yaml` file with the required changes is available in the deploy directory. Examine the values.yaml file by opening it using an editor of your choice. Optionally, you can double-click the **Home** folder on the desktop, go to `/root/labfiles/cpelab/deploy` folder and then right-click `values.yaml` to open with Text Editor.

__ c. Check the image tag in the vlaues.yaml file. The image tag contains details for accessing the CPE container. Find the value of the pullPolicy. The pullPolicy in the image section has been set to IfNotPresent to make sure the container is only loaded when needed.

```
# Default values for ibm-ecm-icp.
# This is a YAML-formatted file.
# Declare variables to be passed into your templates.
arch:
  amd64: "3 - Most preferred"

replicaCount: 1
image:
  repository: docker-registry.default.svc:5000/cpe-lab/cpe
  tag: ga-553-p8cpe
  pullPolicy: IfNotPresent
```

__ d. Check the `dataVolume` section. The `dataVolume` specification has been adapted to match the names of the persistent volume claims, which were created by the script before.

```
## global persistence settings
persistence:
  enabled: true
#  useDynamicProvisioning: false

## Persistence parameters for /database
dataVolume:
## Persistence parameters for CPE
  nameforCPECfgstore: "cpe-cfg-stor"
  nameforCPELogstore: "cpe-log-stor"
  nameforFilestore: "file-stor"
  nameforICMrulestore: "icmrule-stor"
  nameforTextextstore: "textext-stor"
  nameforBootstrapstore: "bootstrap-stor"
  nameforFNLogstore: "fnlog-stor"
```

__ e. Close the yaml when you are done examining it. Do not make any changes.

__ 2. Create a secret. The deployment needs access to the docker registry, which is granted through the secret named admin.registrykey. If it had been existing before, recreate it, by first deleting an old version of it.

__ a. Delete old secret, if present.

```
oc delete secret admin.registrykey -n cpe-lab
```

```
[root@master deploy]# oc delete secret admin.registrykey -n cpe-lab
Error from server (NotFound): secrets "admin.registrykey" not found
[root@master deploy]#
```

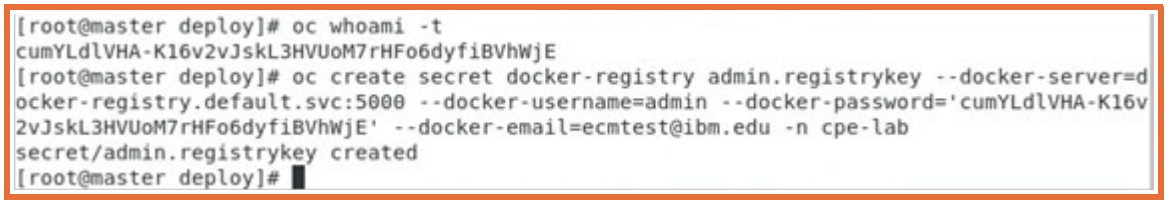
__ b. Create an authentication token for the internal docker registry to connect to the internal docker registry.

```
oc whoami -t
```

```
[root@master deploy]# oc whoami -t
cumYLdlVHA-K16v2vJskL3HVUoM7rHFo6dyfiBVhWjE
[root@master deploy]#
```

__ c. Create a new secret. Log in to the internal docker registry by using an authentication token. Replace the token variable with the token that you created in the previous step. Make sure to include the token in quotation marks (single or double quotation marks). Do not copy and paste the command from the pdf as the quotation marks are not interpreted correctly by the remote VM. Make sure to enter the entire command in a single line.

```
oc create secret docker-registry admin.registrykey
--docker-server=docker-registry.default.svc:5000 --docker-username=admin
--docker-password='<token>' --docker-email=ecmtest@ibm.edu -n cpe-lab
```



__ d. Verify correct installation of the admin.registrykey. Check the output and confirm that the password returned is the same token value that you entered before.

```
oc get secret admin.registrykey --output="jsonpath={.data.\.dockerconfigjson}" |
base64 --decode
```

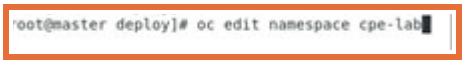
__ e. If the password that is displayed is incorrect, delete the admin.regsitrykey and try re-creating it.

__ f. In OpenShift, the security context constraint on allowed user and group ids in a container can cause deployment errors. To prevent it, you update the namespace to include the constraints for the components that you want to deploy. But first, run the command to make sure that you can use gedit.

```
export EDITOR=gedit
```

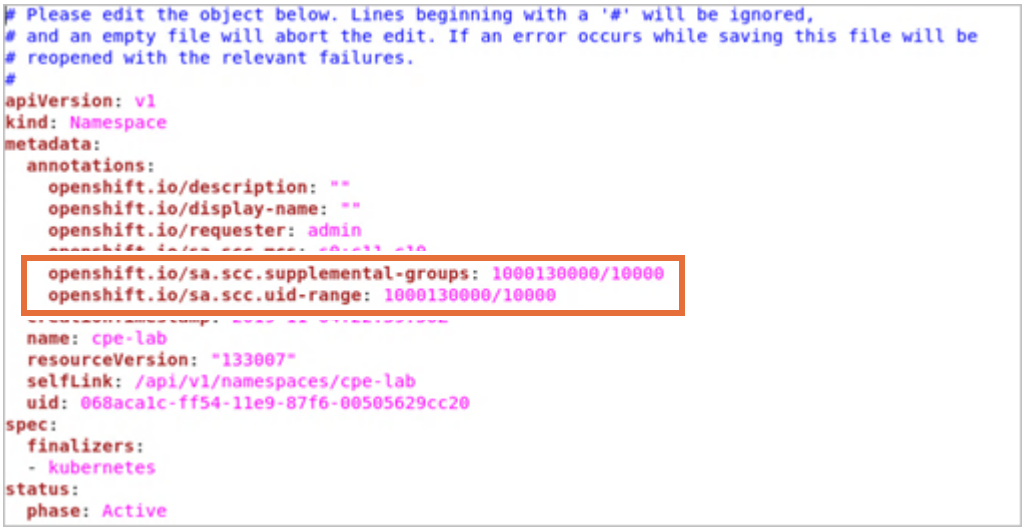
__ g. Now you are ready to use gedit. Run the command to update the namespace.

```
oc edit namespace cpe-lab
```



__ h. In the editor, find the following two lines:

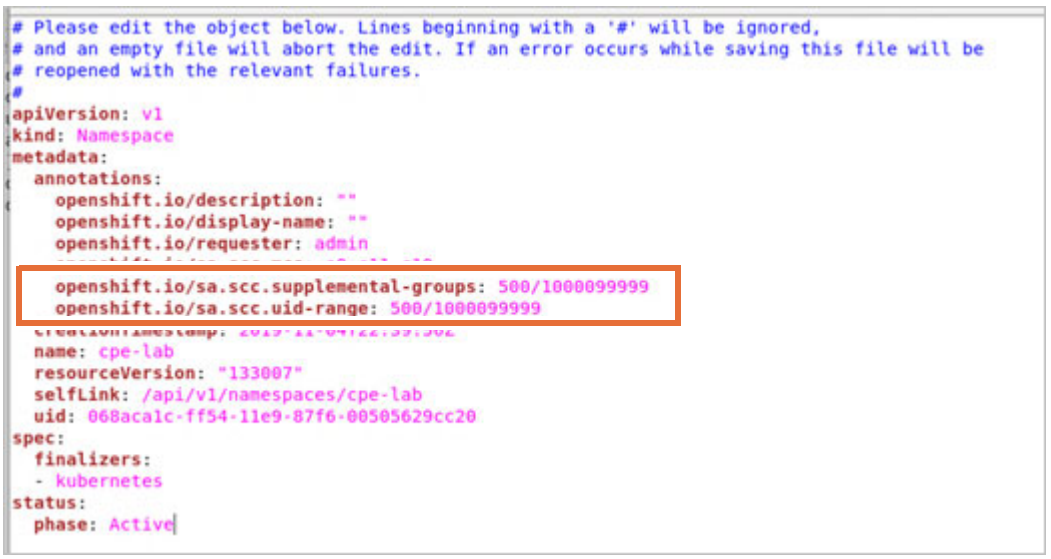
```
openshift.io/sa.scc.supplemental-groups: 1000130000/10000
openshift.io/sa.scc.uid-range: 1000130000/10000
```



__ i. Change the `1000130000/10000` value to `500/1000099999` for both the lines. The two lines are changed to the following values:

```
openshift.io/sa.scc.supplemental-groups: 500/1000099999
openshift.io/sa.scc.uid-range: 500/1000099999
```

__ j. Verify that the changes are done correctly.



__ k. Click **Save** at the top and then close the file. You might have to maximize the editor to view the Save button.

__ l. Add privileges to the CPE project.

```
oc adm policy add-scc-to-user privileged -z default
```

```
[root@master deploy]# oc adm policy add-scc-to-user privileged -z default
scc "privileged" added to: ["system:serviceaccount:cpe-lab:default"]
[root@master deploy]#
```

__ 3. Install the helm chart for deployment

__ a. Verify that you are in the `/root/labfiles/cpelab/deploy` folder.

```
pwd
```

__ a. List the contents of the deploy folder and verify that `ibm-dba-contentservices-3.0.0.tgz helm chart` is listed.

```
ls
```

```
[root@master deploy]# pwd
/root/labfiles/cpelab/deploy
ibm-dba-contentservices-3.0.0.tgz  values.yaml
```

__ b. Install the CPE helm chart by using the helm command. Make sure to enter the entire command in a single line.

```
helm install ibm-dba-contentservices-3.0.0.tgz --name cpe-lab --namespace
cpe-lab --values values.yaml
```

```
[root@master deploy]#
[root@master deploy]# helm install ibm-dba-contentservices-3.0.0.tgz --name cpe-lab --namespace cpe-lab --values values.yaml
```

__ c. The command completes quickly and returns at the command prompt.

```
NOTES:
1.Get the application URL with ClusterIP and Port to configure Navigator.

# Get the ClusterIP for the service

  export CLUSTER_IP=$(kubectl get --namespace cpe-lab -o jsonpath="{.spec.clusterIP}" services cpe-lab-ibm-dba-contentservices
)

# Get the Port for the service

  export PORT=$(kubectl get --namespace cpe-lab  -o jsonpath="{.spec.ports[0].port}" services cpe-lab-ibm-dba-contentservices)

  echo "Using this Cluster IP address and Port to configure CSS server in CPE: $CLUSTER_IP:$PORT"

2.Get the application URL with Node IP and Port by running these commands:

  export NODE_PORT=$(kubectl get --namespace cpe-lab -o jsonpath="{.spec.ports[0].nodePort}" services cpe-lab-ibm-dba-contents
ervices)

# Get the HTTPS Port for the service
# Replace this with actual deployment service name for <Service Name>

  export NODE_PORT_HTTPS=$(kubectl get --namespace cpe-lab -o jsonpath="{.spec.ports[1].nodePort}" services cpe-lab-ibm-dba-co
ntentservices)

# Get the proxy IP for the cluster
  export NODE_IP=$(kubectl get nodes | grep proxy | awk '{print $1}')

  echo http://$NODE_IP:$NODE_PORT

  echo https://$NODE_IP:$NODE_PORT_HTTPS
```

If everything went well, the CPE container is deployed in few minutes. You verify that in the next section.

Part 7: Verify successful Content Platform Engine deployment

In this section, you run several verification steps for successful CPE deployment

__ 1. Check the status of the pod. The status of the pod is a good indicator of whether the containers are either running, failed, started, crashed, and so on. Although it takes time for the containers to start, checking the status first gives a good idea of the state of the pod.

__ a. Check the status of the pod.

```
oc get pods
```

__ b. Verify that there is one row for the cpe pod.

__ c. Examine the status of the pod as it is being created. It takes few minutes for the pod to start and get into a Running status. It is OK if you see a Ready state of 0/1 for pod. Give it some time to change to state 1/1. While you are waiting for state to change, continue with the following verifications. You run this command again later. If the state is already 1/1 then you can optionally, review the remaining verification steps.

```
[root@master deploy]# oc get pods
NAME                                          READY     STATUS              RESTARTS   AGE
cpe-lab-ibm-dba-contentservices-5dc55857bb-2pn57   0/1       ContainerCreating   0          25s
[root@master deploy]#
```

__ 2. Verify the deployment status. When the deployment is started, the helm chart creates among others artifacts, a deployment.

__ a. Run the command to check deployment and get detailed information about the deployment.

```
oc describe deployment cpe-lab
```

```
[root@master deploy]#
[root@master deploy]# oc describe deployment cpe-lab
```

__ b. Verify that a message is received that the replica set was successfully scaled to 1.

```
    ClaimName:  cpe-lab-icp-icmrulesstore-pvc
    ReadOnly:   false
  textext-stor:
    Type:       PersistentVolumeClaim (a reference to a PersistentVolumeClaim in the same namespace)
    ClaimName:  cpe-lab-icp-textextstore-pvc
    ReadOnly:   false
  bootstrap-stor:
    Type:       PersistentVolumeClaim (a reference to a PersistentVolumeClaim in the same namespace)
    ClaimName:  cpe-lab-icp-bootstrapstore-pvc
    ReadOnly:   false
  fnlog-stor:
    Type:       PersistentVolumeClaim (a reference to a PersistentVolumeClaim in the same namespace)
    ClaimName:  cpe-lab-icp-fnlogstore-pvc
    ReadOnly:   false
  file-stor:
    Type:       PersistentVolumeClaim (a reference to a PersistentVolumeClaim in the same namespace)
    ClaimName:  cpe-lab-icp-filestore-pvc
    ReadOnly:   false
Conditions:
  Type           Status  Reason
  ----           ------  ------
  Available      True    MinimumReplicasAvailable
  Progressing    True    NewReplicaSetAvailable
OldReplicaSets:  cpe-lab-ibm-dba-contentservices-5dc55857bb (1/1 replicas created)
NewReplicaSet:   <none>
Events:
  Type    Reason             Age   From                   Message
  ----    ------             ----  ----                   -------
  Normal  ScalingReplicaSet  6m    deployment-controller  Scaled up replica set cpe-lab-ibm-dba-contentservices-5dc55857bb to
1
[root@master deploy]#
```

__ 3. Check the replica set.

__ a. Run the command to check the replica and return the information that a pod is created.

```
oc describe replicaset cpe-lab
```

```
[root@master deploy]# oc describe replicaset cpe-lab
```

```
    ReadOnly:   false
Events:
  Type    Reason            Age   From                   Message
  ----    ------            ----  ----                   -------
  Normal  SuccessfulCreate  10m   replicaset-controlle   Created pod: cpe-lab-ibm-dba-contentservices-5dc55857bb-2pn57

[root@master deploy]#
```

__ 4. Check the pod for messages that are generated during its creation.

__ a. Run the command to view the pod creation messages and watch for errors.

```
oc describe pod cpe-lab
```

```
[root@master deploy]# oc describe pod cpe-lab
```

```
  Type    Reason     Age   From                          Message
  ----    ------     ----  ----                          -------
  Normal  Scheduled  13m   default-scheduler             Successfully assigned cpe-lab/cpe-lab-ibm-dba-contentservices-5dc55857b
b-2pn57 to compute1.cp4a.com
  Normal  Pulling    13m   kubelet, compute1.cp4a.com  pulling image "docker-registry.default.svc:5000/cpe-lab/cpe:ga-553-p8cp
e"
  Normal  Pulled     11m   kubelet, compute1.cp4a.com  Successfully pulled image "docker-registry.default.svc:5000/cpe-lab/cpe
:ga-553-p8cpe"
  Normal  Created    11m   kubelet, compute1.cp4a.com  Created container
  Normal  Started    11m   kubelet, compute1.cp4a.com  Started container
```

__ 5. Check the status of the pod one more time.

__ a. Now that some time is passed, you run the verification again to make sure that the state of the CPE pod is 1/1 and that it has a Running status. If the state is still 0/1, then wait few more minutes and check again.

```
oc get pods
```

```
[root@master deploy]# oc get pods
NAME                                              READY     STATUS    RESTARTS   AGE
cpe-lab-ibm-dba-contentservices-5dc55857bb-2pn57  1/1       Running   0          16m
[root@master deploy]#
```

__ b. The successful installation results in a running CPE container. Make sure that the status is Running for the pod. It is important that the container also displays 1/1 as in Ready. If it displays 0/1, then give it few more minutes before it changes to 1/1.

Part 8: Troubleshoot failed deployment.

Several tasks are available for you troubleshoot a failed deployment. This section describes some of the steps you can take to ensure a successful deployment.

__ 1. Check for typographical errors. If your deployment fails or you see an error, then the first thing to check is to make sure that no typographical errors exist. Check the typed command to see any obvious mistakes. You can scroll up through the terminal and examine the previous commands, if needed. You can use the up arrow key in your keyboard to recall the earlier run commands to scan through any mistakes and make corrections.

__ 2. Verify that the lab environment is up and running. If you started a suspended environment, then make sure that you give plenty of time before working with the environment. A good verification is to log in to the OpenShift console and make sure that you are successful.

```
oc login https://console.cp4a.com:8443 -u admin -p passw0rd
```

__ 3. Verify the current project.

__ a. Verify that you are connected to the correct project cpe-lab.

```
oc project
```

__ b. If the current project is the default project or some other project, then switch to the cpe-lab project.

```
oc project cpe-lab
```

__ 4. Check container logs for the container, fix errors and then redeploy.

__ a. You can also check the logs for a failed or crashing container to help troubleshoot the problem. First, check the pod status to identify the container name that is being used in the current run.

```
oc get pods
```

The state of a container might be stuck at 0/1 even though it says it is Running. If it does not change to 1/1 even after a while then you might want to check the log for that container. Copy the name of that container.

In other failures, it is possible that a container fails to start and gives an error. If that happens, then you can again copy the name of the failing container (s) to check the logs.

__ b. Check the logs of a specific problematic container by using its name.

```
oc logs ocpe-lab-ibm-dba-contentservices-5dc55857bb
```

If the results indicate a problem, then you can go back to see whether any mistake was done or a step was skipped.

__ 5. Remove the current failed deployment, fix errors, and then redeploy. You do not need to run this step if you successfully deployed CPE. In that case, you can just review this step.

__ a. Run the command to purge the deployment. Do **not** run this command when the deployment is successful. Note that there are two dashes right before purge (even though it can look like a single dash) in the command. You can run this when you identified the problem, made the correction, and are ready to install the helm chart again for deployment.

```
helm delete cpe-lab --purge
```

__ a. Check the pod status. The containers start terminating.

```
oc get pods
```

__ b. Run the check again to verify the pods again until resources are no longer available.

__ c. Now you are ready to install CPE helm chart again using the helm command:

Part 9: Access the running Content Platform Engine container

Now that CPE is deployed, you are ready to connect to it. In this section, you verify the connectivity and access to CPE. You do not do any CPE development. That is the task of a CPE developer or CPE Administrator.

__ 1. Identify ports and URL for the running containers

__ a. Verify that containers are running.

__ b. Find the port numbers.

```
oc describe service cpe-lab
```

__ c. Examine the output from the command. It displays the node ports of the running services. You log in to the relevant services in the next steps.

Note the node port. There are two of them. One for an http request and the other for an https request.

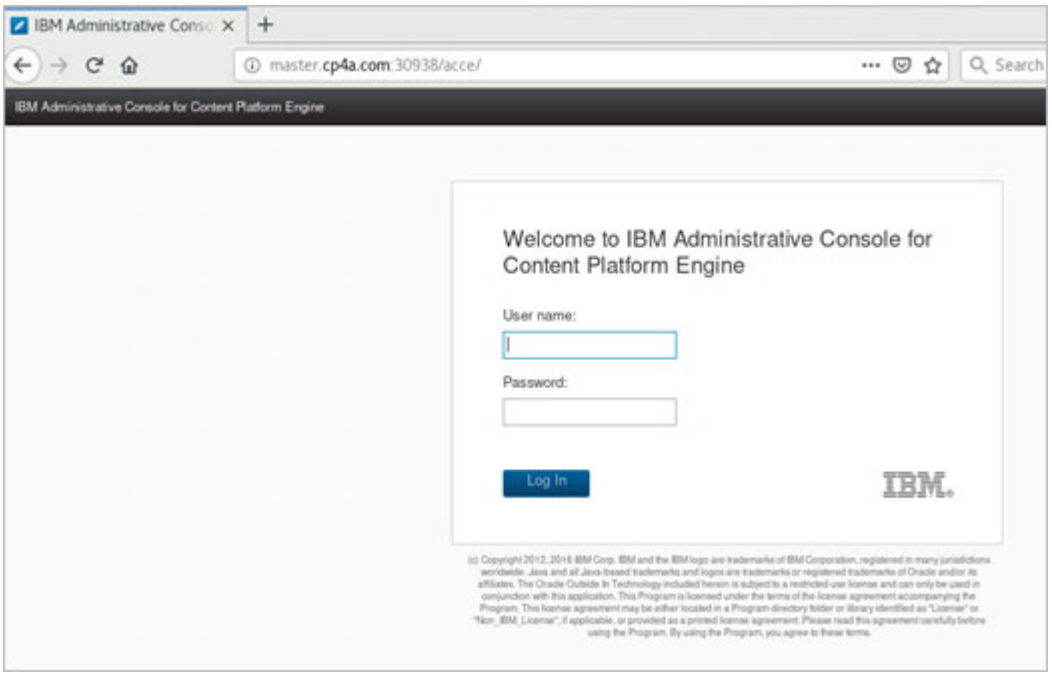
__ 2. Connect to the Administrative Console for Content Engine (ACCE)

__ a. Find the node port for the CPE container. For ACCE URL, you can use either `http://master.cp4a.com:<http node port>/acce` or `https://master.cp4a.com:<https node port>/acce`. The node port in your environment is different from the node port in the view below:

__ b. Start Firefox and enter the following URL:

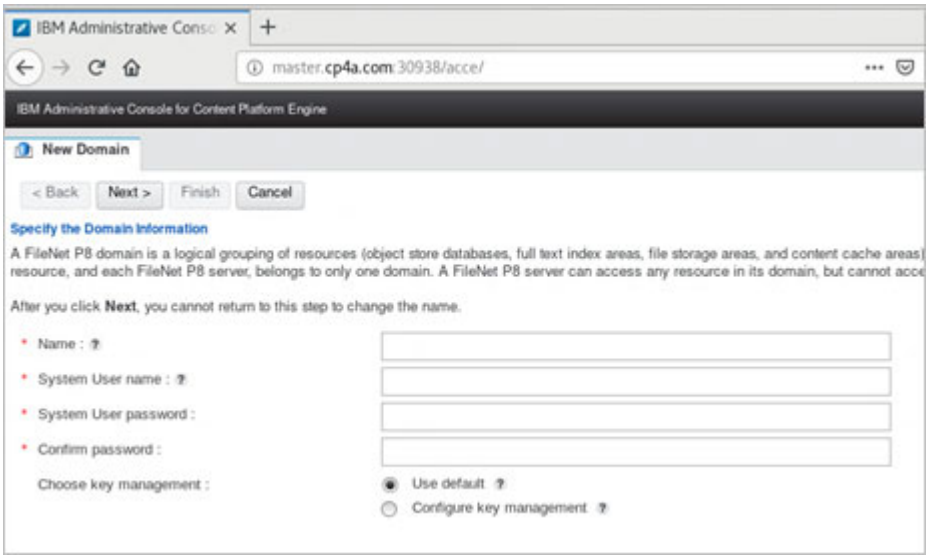
`https://master.cp4a.com:<https NodePort>/acce`

__ c. Verify that the ACCE login page is displayed in the browser.



__ d. Log in by using `p8admin` for User name and `passw0rd` for Password. Click Log In.

__ e. After a successful login to the Administrative console, verify that the Domain configuration page is displayed. This confirms the successful ACCE deployment.



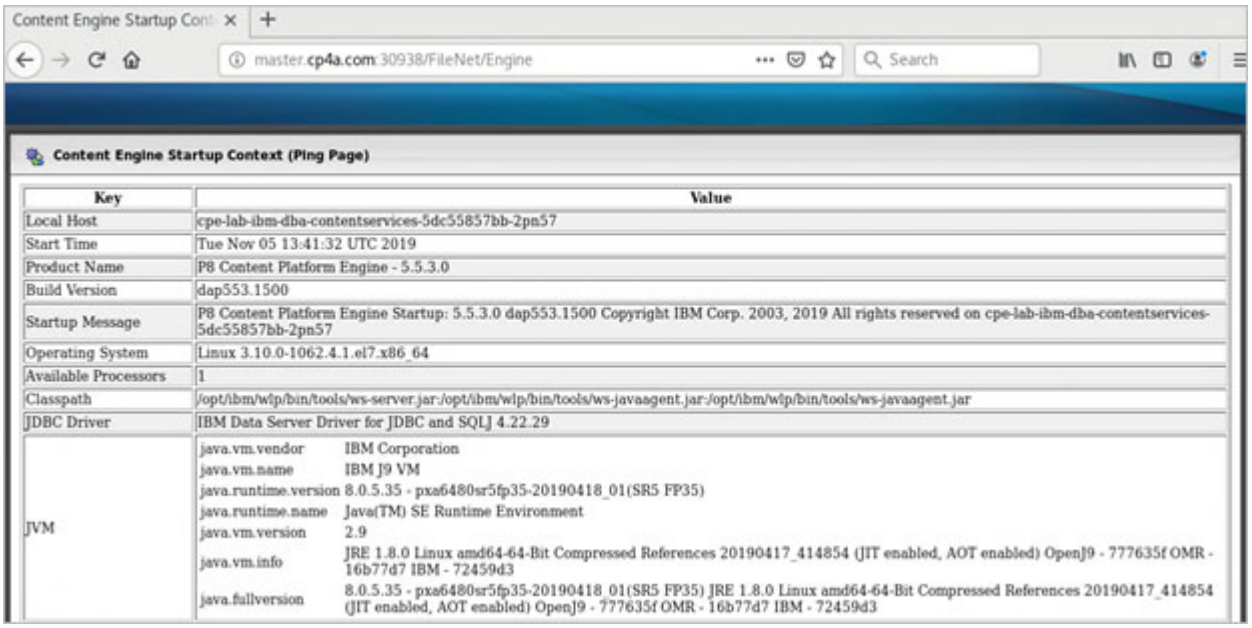
__ 3. Connect to the Content Engine Ping Page (CE Ping page)

__ a. Use the same the node port for the CPE container that you used before. For CE Ping Page URL, you can use either `http://master.cp4a.com:<http node port>/FileNet/Engine` or `https://master.cp4a.com:<https node port>/FileNet/Engine`.

__ b. In the browser, enter the following URL:

```
https://master.cp4a.com:<https NodePort>/FileNet/Engine
```

__ c. Verify that the CE Ping Page is displayed in the browser.



Key	Value
Local Host	cpe-lab-ibm-dba-contentservices-5dc55857bb-2pn57
Start Time	Tue Nov 05 13:41:32 UTC 2019
Product Name	P8 Content Platform Engine - 5.5.3.0
Build Version	dap553.1500
Startup Message	P8 Content Platform Engine Startup: 5.5.3.0 dap553.1500 Copyright IBM Corp. 2003, 2019 All rights reserved on cpe-lab-ibm-dba-contentservices-5dc55857bb-2pn57
Operating System	Linux 3.10.0-1062.4.1.el7.x86_64
Available Processors	1
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
JDBC Driver	IBM Data Server Driver for JDBC and SQLJ 4.22.29
JVM	java.vm.vendor: IBM Corporation java.vm.name: IBM J9 VM java.runtime.version: 8.0.5.35 - pxa6480sr5fp35-20190418_01(SR5 FP35) java.runtime.name: Java(TM) SE Runtime Environment java.vm.version: 2.9 java.vm.info: JRE 1.8.0 Linux amd64-64-Bit Compressed References 20190417_414854 (JIT enabled, AOT enabled) OpenJ9 - 777635f OMR - 16b77d7 IBM - 72459d3 java.fullversion: 8.0.5.35 - pxa6480sr5fp35-20190418_01(SR5 FP35) JRE 1.8.0 Linux amd64-64-Bit Compressed References 20190417_414854 (JIT enabled, AOT enabled) OpenJ9 - 777635f OMR - 16b77d7 IBM - 72459d3

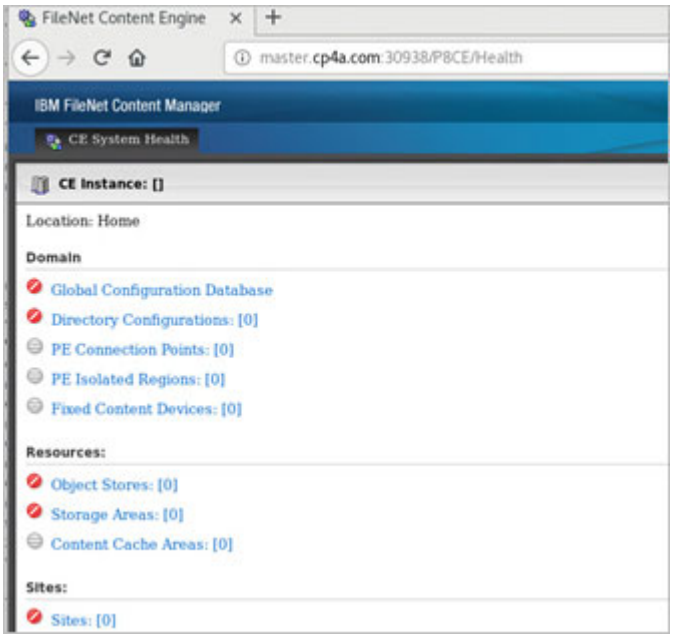
__ 4. Connect to the Content Engine Health Page (CE Health page)

__ a. Use the same the node port for the CPE container that you used before. For CE Health Page URL, you can use either `http://master.cp4a.com:<http node port>/P8CE/Health` or `https://master.cp4a.com:<https node port>/P8CE/Health`.

__ b. In the browser, enter the following URL:

```
https://master.cp4a.com:<https NodePort>/P8CE/Health
```

__ c. Verify that the CE Health Page is displayed in the browser.



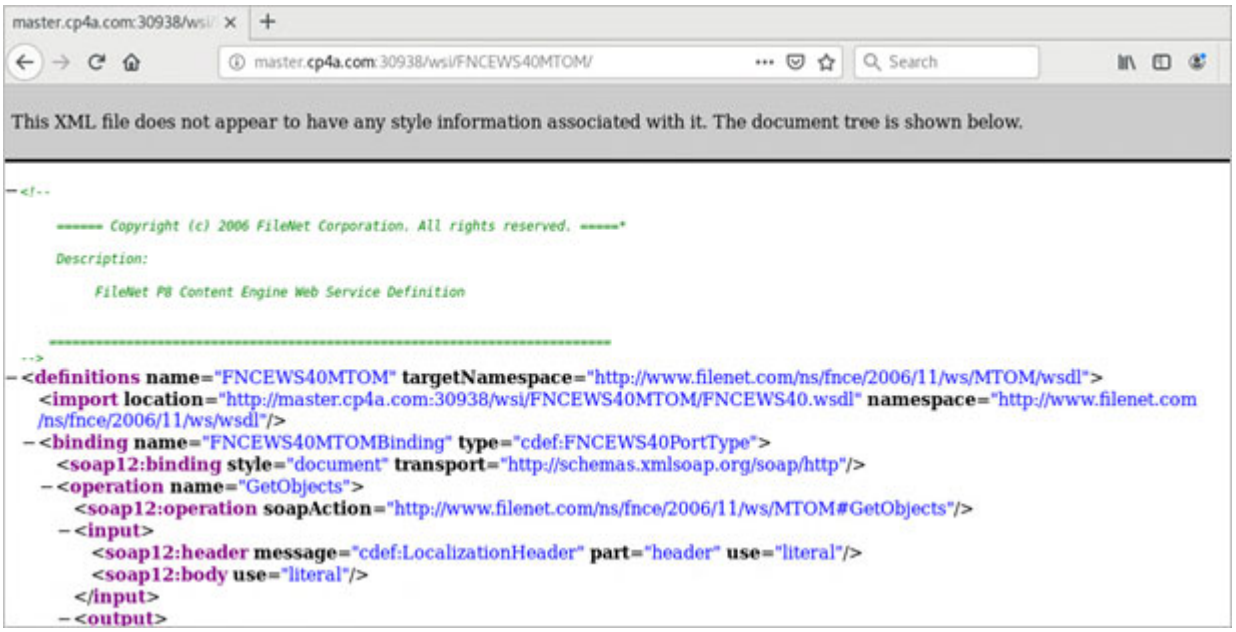
__ 5. Connect to the Content Engine Web Service (CE WS)

__ a. Use the same the node port for the CPE container that you used before. For CE WS URL, you can use either `http://master.cp4a.com:<http node port>/wsi/FNCEWS40MTOM/` or `https://master.cp4a.com:<https node port>/wsi/FNCEWS40MTOM/`.

__ b. In the browser, enter the following URL:

`https://master.cp4a.com:<https NodePort>/wsi/FNCEWS40MTOM/`

__ c. Verify that the CE WS is displayed as an XML in the browser.



__ 6. Close all open windows. Before moving to the next exercise, it is a good idea to close all open windows and sessions.

__ a. Enter exit in the terminal to close it. If you had multiple terminal windows open close all of them

__ b. Close the Firefox browser.

__ c. Close any other open windows, including any text editors.

__ d. Close the RDP console window or the browser window that was used to work with the Master VM.

You now completed the successful deployment of the CPE container and verified the connectivity and access to CPE. You do not do any CPE development or any other work by using the CPE container. That is the task of a FIleNet P8 Content Platform Engine developer or FIleNet P8 Content Platform Engine Administrator. For FIleNet P8 Content Platform Engine development related training refer to the IBM Training site.This concludes this exercise.

End of exercise

Exercise 3. Administering the IBM Cloud Pak for Automation containers

Estimated time

02:00

Overview

In this exercise, you do some basic administration and management of the installed containerized IBM Cloud Pak for Automation products that are deployed on the Red Hat OpenShift environment.

Objectives

After completing this exercise, you should be able to:

- Explore the Red Hat OpenShift container (RHOCP) web console for container management
- Examine the available open source monitoring options - metrics, alerts, and dashboards
- Scale an application deployment by using OpenShift
- Monitor containers by using probes

Introduction

The IBM Cloud Pak for Automation (CP4A) offers a software platform to develop, deploy, run, and manage your digital business automation projects by using its capabilities. In this exercise, you administer the containers.

Management of containerized products is an extensive area. In this exercise, you get introduced to some important administration areas of container management by using OpenShift and learn some key basic skills that are required by an administrator.

Requirements

- Availability of the lab environment that consists of three Red Hat Enterprise Linux V7.7 virtual machines (VMs) with Red Hat OpenShift (RHOCP) 3.11 installed.
- Completion of either Exercise 1 or Exercise 2 or both.

Virtual machine configuration

Each virtual machine is configured with the following specification:

Table 9.

OS	CPUs	RAM	Disk
Red Hat Enterprise Linux (RHEL) 7.7 (64-bit)	8	16 GB	300 GB

List of Servers with Roles

The virtual machines are listed below with their respective roles:

Table 10.

VM	RHCOCP Node type	IP address	Hostname
VM1- OCP master	Master	`10.0.0.1`	`master.cp4a.com`
VM2- OCP compute1	Compute	`10.0.0.2`	`compute1.cp4a.com`
VM3- OCP compute 2	Compute	`10.0.0.3`	`compute2.cp4a.com`

Software Requirements

The following packages are downloaded to the **Master** node:

Table 11.

Software	Folder name
IBM Cloud Pak for Automation images	`/root/labfiles/cp4a`

User IDs and Passwords

The following table contains a list of User ID and password information that is required for this exercise:

Table 12.

Entry Point	User ID	Password
OpenShift web console: `https://master.cp4a.com:8443` or `https://console.cp4a.com:8443`	`admin`	`passw0rd`
Red Hat Linux VM	`root`	`passw0rd`

Information

In this exercise, you work with the existing odm-lab and cpe-lab projects that you created in Exercise 1 and Exercise 2. You also work with containers from the odm-lab project that you deployed in Exercise 1.

If you skipped Exercise 1 and did Exercise 2 only, then you can use the cpe-lab project to do all your work in this Exercise 3. You can also use the container from the cpe-lab project instead of the containers that are deployed in the odm-lab project. Several screen captures and references might not match exactly with your environment so you need to modify your steps to match your environment.

If you skipped Exercise 2 and did Exercise 1 only, then you can use the odm-lab project to do all your work in this exercise. Few screen captures and references might not match exactly with your environment so you need to modify your steps slightly to match your environment.

The basic management behavior is the same regardless of the project or pod type.

Part 1: Explore the OpenShift web console for container management

Until now, you worked with the command-line interface (CLI) to interact with RHOCP. In this exercise, in addition to using the CLI, you also work with the OpenShift web console.

The OpenShift web console is a user interface accessible from a web browser. It is a convenient way to manage and monitor applications. Although CLI can be used to manage the lifecycle of applications, the web console presents several extra benefits, such as the state of a deployment, pod, service, and other resources, and providing information about system-wide events.

The web console runs as a pod on the master.

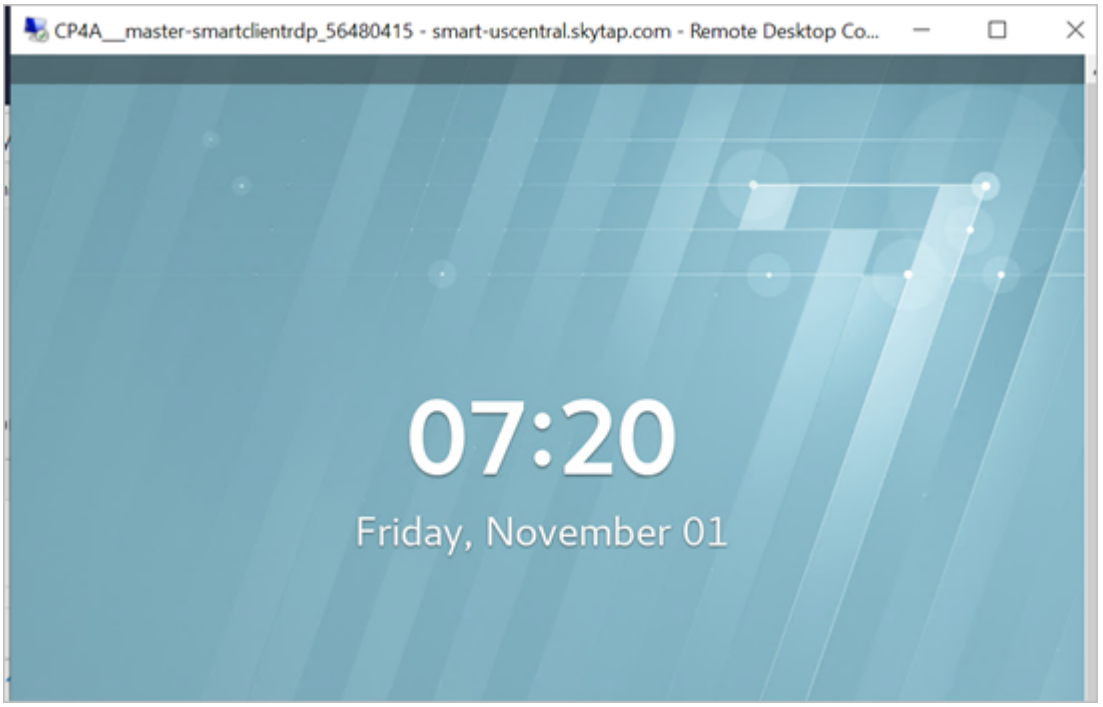
You can use the web console to monitor critical properties in your infrastructure, which include:

- The readiness or the status of a pod
- The availability of a volume
- The availability of monitoring - metrics, alerts and dashboard
- The availability of an application by using probes

__ 1. Verify that the lab environment is up and running as explained in the previous Exercises. If you completed either Exercise 1 or Exercise 2, then you are already familiar with the steps to take when your environment is in a Suspended or Powered Off state. If needed, you can also review those instructions again to make sure that your environment is in a Running state before continuing with this exercise.

__ 2. Log in to the **Master** VM. Remember, that you work with the **Master** VM throughout the exercise. All the commands that you run in this exercise, you run on the **Master** VM.

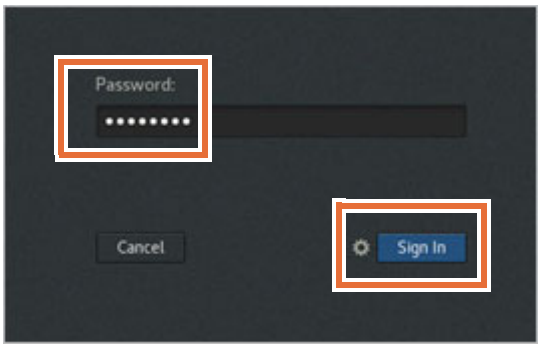
__ a. If the time, and date displays on the desktop, click the Enter key anywhere on the desktop for the login screen to come up.



__ b. Click **Not listed**.

__ c. Enter `root` in the Username field and click **Next**.

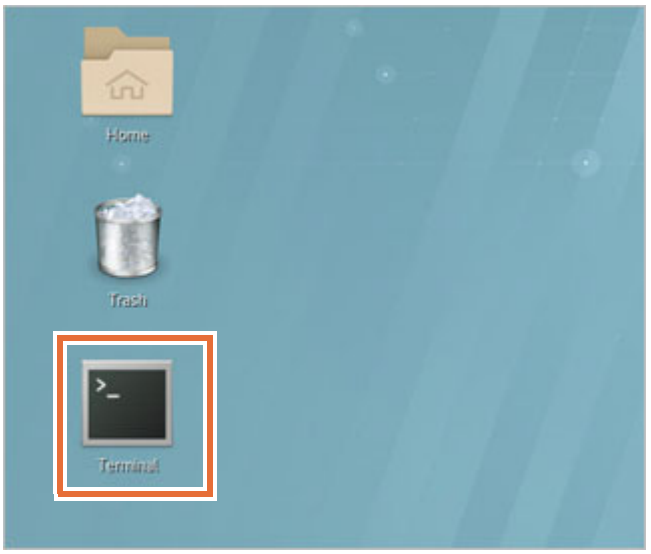
__ d. Enter `passw0rd` in the Password field and click **Sign In**.



__ e. You are now successfully logged in to the Master VM.

__ 3. Log in to the Red Hat OpenShift cluster by using the command line

__ a. Double-click the Terminal shortcut on the desktop to open it.



__ b. In the terminal that opens, log in to the OpenShift cluster. The OpenShift console URL is `https://console.cp4a.com:8443` and the username/password `is admin/passw0rd`.

```
oc login https://console.cp4a.com:8443 -u admin -p passw0rd
```

__ c. Verify that the Login successful message is displayed. It does not matter which project you are connected to right now. Next, you verify that things are running fine in the environment.

__ 4. Check the status of the nodes.

__ d. Verify that the master and the two compute nodes are in the ready state.

```
oc get nodes
```

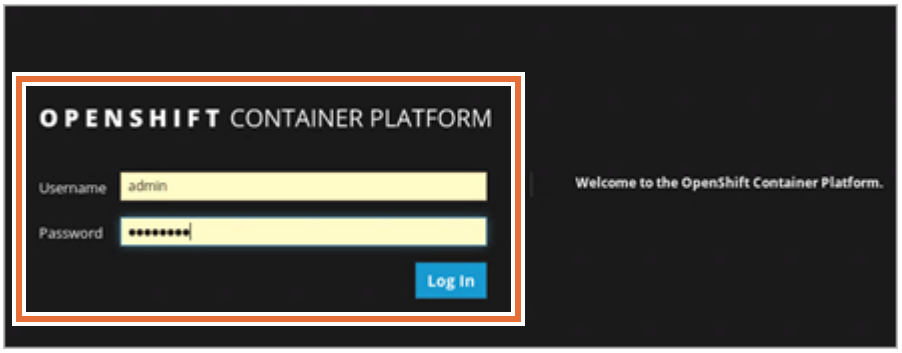
```
[root@master ~]# oc get nodes
NAME                STATUS    ROLES          AGE       VERSION
compute1.cp4a.com   Ready     compute        12d       v1.11.0+d4cacc0
compute2.cp4a.com   Ready     compute        12d       v1.11.0+d4cacc0
master.cp4a.com     Ready     infra,master   12d       v1.11.0+d4cacc0
```

__ e. Verify that three nodes are listed and the status is Ready for all the nodes.

__ 5. Log in to the OpenShift web console

__ a. Start Firefox and click the OpenShift web console shortcut. Optionally, you can enter `https://console.cp4a.com:8443` in the URL field and press the Enter key.

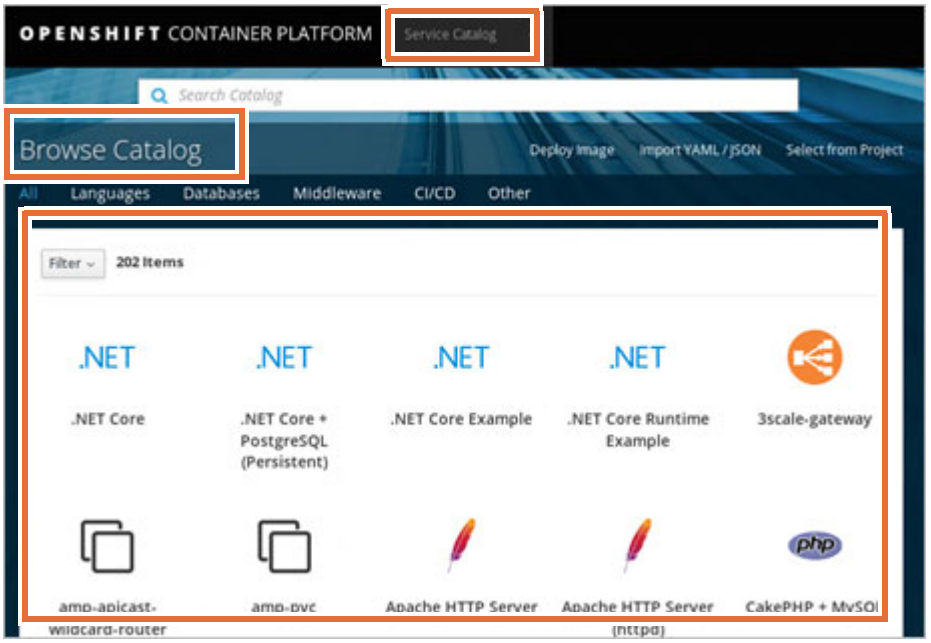
__ b. Enter `admin` in the **Username** field and `passw0rd` in the **password** field. Click **Log In**.



You are now ready to work with the web console.

__ 6. Explore the Service Catalog view.

__ a. The first time you log in to the web console you are in the Service Catalog view. Verify that the Browse Catalog page is open with several service templates displayed.



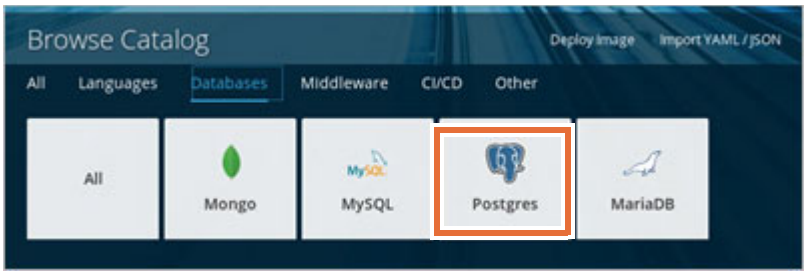
OpenShift Container Platform includes a service catalog, which is an implementation of the Open Service Broker API (OSB API) for Kubernetes. This allows users to connect

any of their applications that are deployed in OpenShift Container Platform to a wide variety of templates.

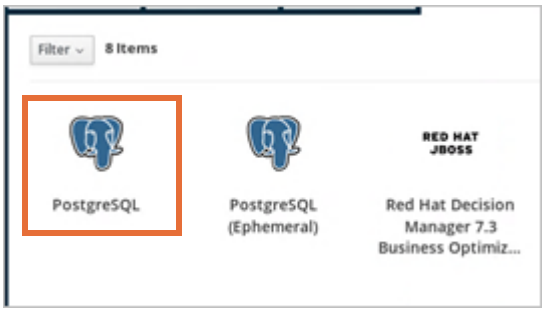
A template describes a set of objects that can be parameterized and processed to produce a list of objects for creation by OpenShift Container Platform. A template can be processed to create anything you have permission to create within a project, for example services, build configurations, and deployment configurations.

__ b. These Catalog templates are grouped by categories such as Languages, Databases, Middleware, CICD, and other. Click the **Databases** tab.

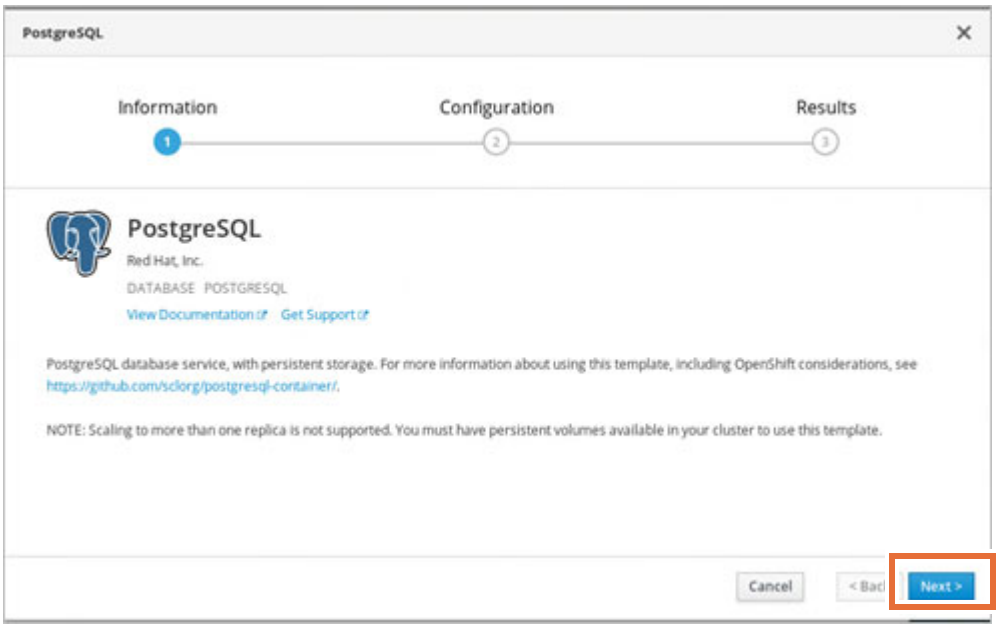
__ c. Several database options are displayed. Click **Postgres**.



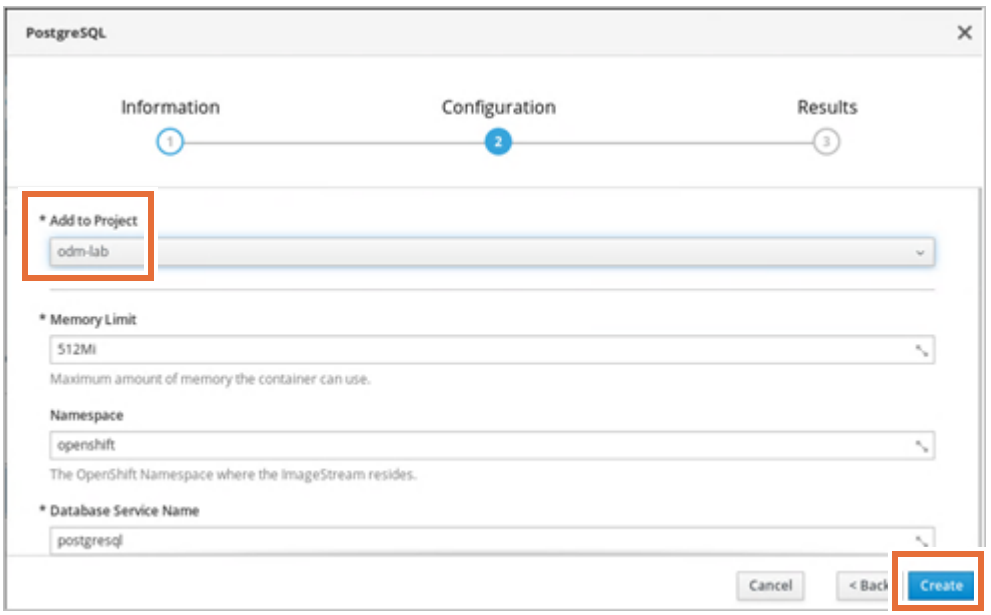
__ d. Click PostgreSQL.



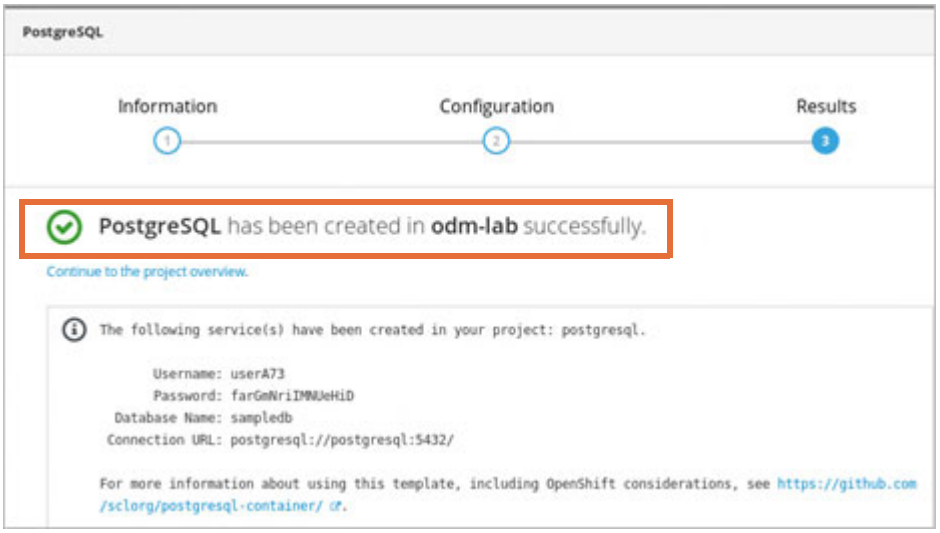
__ e. The PostgreSQL configuration page is displayed. OpenShift Container Platform provides a container image for running PostgreSQL. This image can provide database services based on username, password, and database name settings provided by configuration. You can use this configuration in your existing projects. Click **Next**.



__ f. In the template configuration page, select the odm-lab project if you did Exercise 1. Otherwise, select cpe-lab project. Leave the remaining defaults and click **Create**.



__ g. A message is displayed that PostgreSQL is created successfully. While this portion of the creation is successful, it fails, as you see later. The purpose of these steps is to demonstrate how easy it is to add a new template from the service catalog and then show how to investigate the failed pod. In a later step, you view the failed status and explore the events that are generated for the PostgreSql pod. Click **Close**.



__ h. Feel free to explore the other templates available in the service catalog.

__ 7. Verify that the PostgreSql database pod throws an error.

__ a. Switch back to the terminal and check the current project for that session.

```
oc project
```

__ b. Make sure that the current project is the one that you used earlier for the configuration of the Postgresql service. Switch to that project if you need to. If you completed both Exercise 1 and Exercise, then you used odm-lab project in the earlier step.

```
oc project odm-lab
```

__ c. Check the pods for your project.

```
oc get pods
```

If you are in the odm-lab project, you see four pods for odm and two pods for Postgresql.

If it has been less than 10 minutes since you added Postgresql database to your project, then one of the Postgresql pods is in Ready 1/1 state. It is the pod that is attempting the deployment of the database. The second pod is in the pending state since it is still waiting for deployment to occur. That is why the state is still 0/1 and not 1/1 for the second Postgresql pod. The AGE column displays the time that lapsed since you added the Postgresql service. If it displays less than 10 minutes, then the state continues to be 0/1. Wait for the pod to fail with an error. For that to happen, the AGE column needs to get to at least 10 minutes for the Postgresql row.

```
[root@master ~]# oc get pods
NAME                                              READY     STATUS    RESTARTS   AGE
odm-lab-odm-decisioncenter-7f78864cb4-n2qzn       1/1       Running   1          1h
odm-lab-odm-decisionrunner-b6bf759fd-nqzkw        1/1       Running   1          1h
odm-lab-odm-decisionserverconsole-77c88fbb85-2h2h9 1/1      Running   1          1h
odm-lab-odm-decisionserverruntime-78c44b4547-mqv76 1/1      Running   1          1h
postgresql-1-deploy                               1/1       Running   0          1m
postgresql-1-kvhpw                                0/1       Pending   0          1m
[root@master ~]#
```

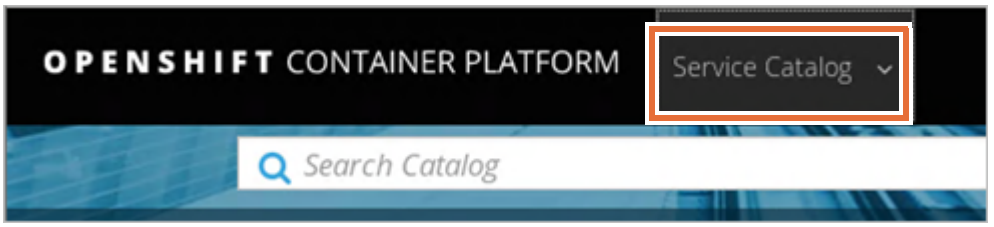
__ d. Check the pods one more time to see whether 10 minutes passed. The timeout value for this deployment is 600 seconds or 10 minutes. It takes 10 minutes from the time you added the service for the error to appear. You know 10 minutes passed when the Age displays at 10m or more. When that happens, only one Postgredql pod is listed and since it failed deployment the status is Error. You can now continue to the next step.

```
[root@master ~]# oc get pods
NAME                                              READY     STATUS    RESTARTS   AGE
odm-lab-odm-decisioncenter-7f78864cb4-n2qzn       1/1       Running   1          1h
odm-lab-odm-decisionrunner-b6bf759fd-nqzkw        1/1       Running   1          1h
odm-lab-odm-decisionserverconsole-77c88fbb85-2h2h9 1/1      Running   1          1h
postgresql-1-deploy                               0/1       Error     0          11m
[root@master ~]#
```

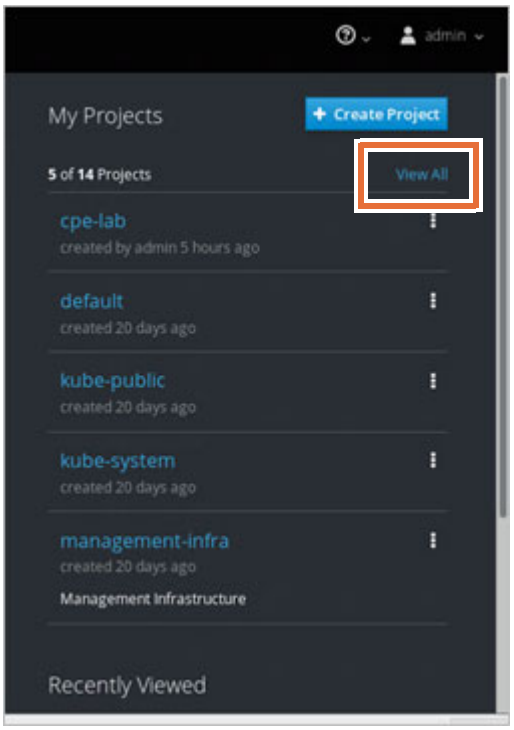
__ e. Leave the terminal open and switch to the web console.

__ 8. Explore the Service Catalog view

__ f. Verify that you are still in the **Service Catalog** view of the web console.



__ g. The right pane displays the projects that you have permission to work with. Since admin is the cluster administrator, all projects are listed. Click **View All** to view all the projects in this cluster.



__ h. In the My Projects page that opens, scroll down and click **odm-lab**. If you did not complete the odm-lab project in Exercise 1, then open the cpe-lab project that you created in Exercise 2.

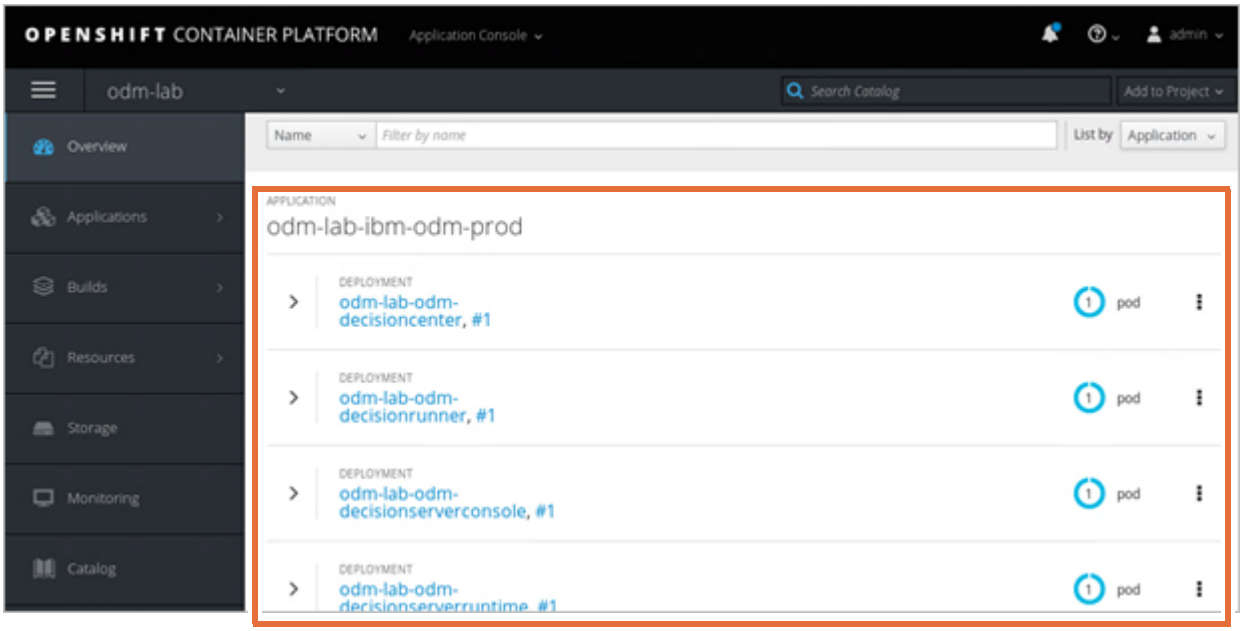
__ i. Verify that the current view switches from the **Service Catalog** view to the **Application Console** view.

Reminder

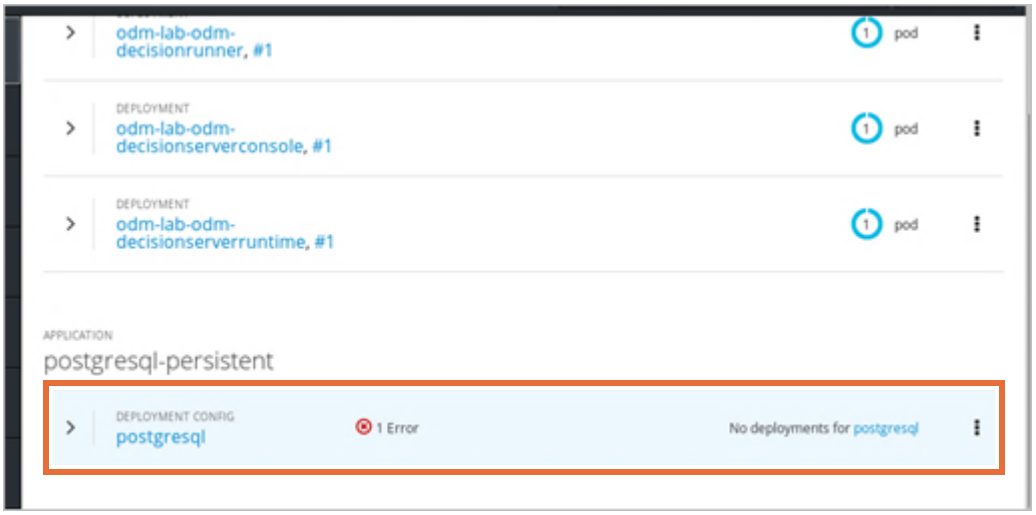
The steps below show the pods for the odm-lab project. If you use the cpe-lab project, then the screen captures do not match exactly with your environment. You need to modify the steps to match your project.

__ 9. Explore the **Application Console** view in the web console.

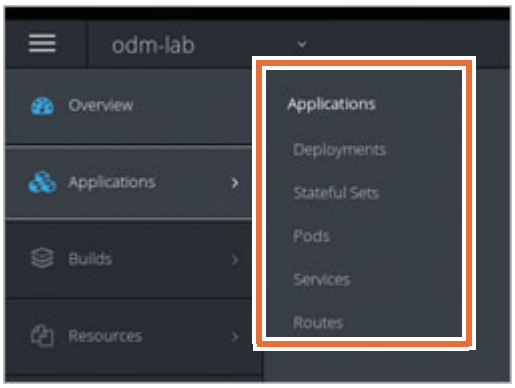
__ a. When you click a project, the web console provides an overview of your project. The **Overview** tab is selected by default and provides a high-level view of the current project. It displays the name of the services and their associated pods that are running in the project. Verify that the ODM pods are displayed. (If you opened the cpe-lab project, you see only the CPE pod). You might need to scroll down the browser to view all the pods.



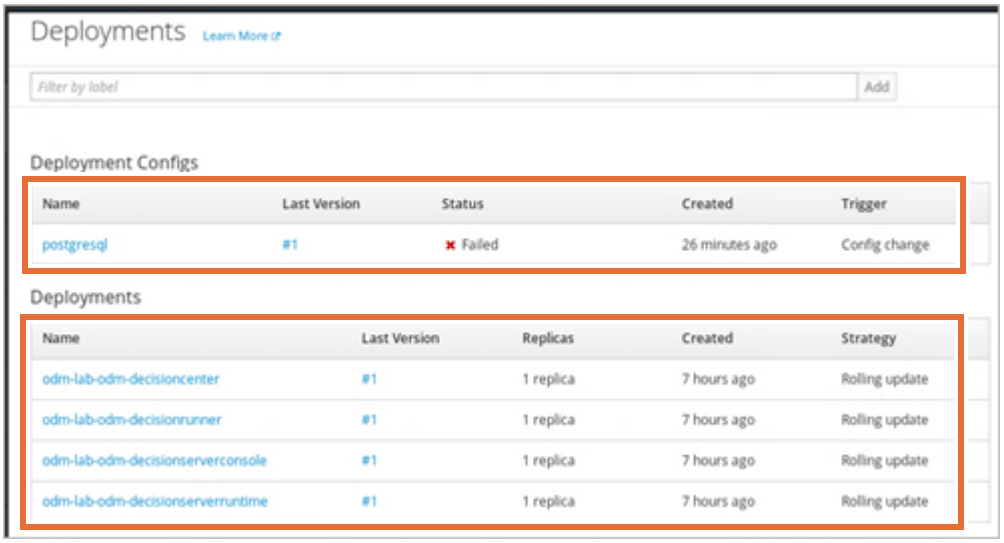
__ b. If you scroll further down, you see the failed postgresql pod that you just added to the project. If you do not see the error for postgresql, then it is trying to re-create the deployment. The timeout value for this deployment is 600 seconds or 10 minutes. It takes 10 minutes from the time you added the service for the error to appear.



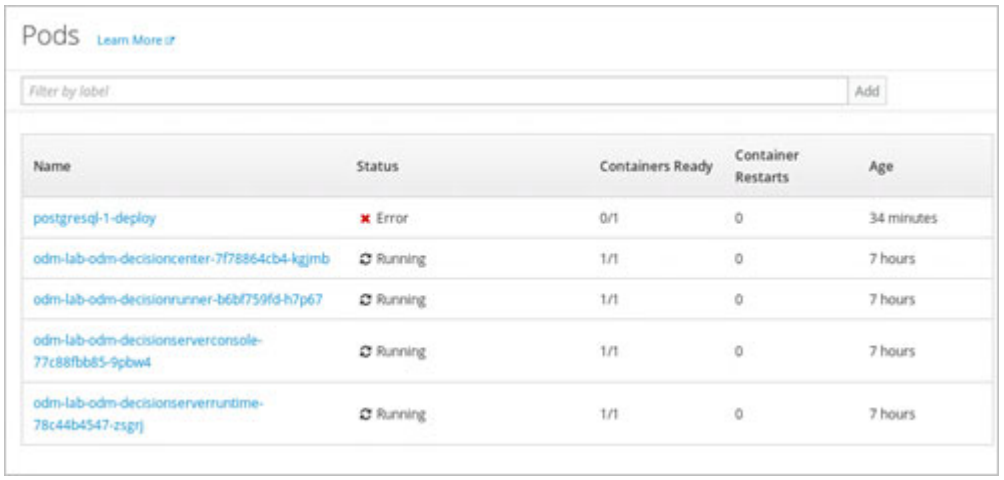
__ c. Click the **Applications** tab. The Applications tab provides access to deployments, pods, services, and routes. It also gives access to Stateful Sets, a Kubernetes feature that provides a unique identity to pods for managing the ordering of deployments.



__ d. Click **Application** > **Deployments** to view the deployments for the odm-lab project. Once again, you see the four rows for ODM container deployment and one for the failed postgresql deployment.

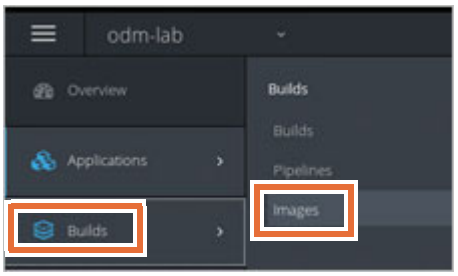


__ e. Click **Application** > **Pods** to view the Pods for the odm-lab project. All the pods have a Running status except for the postgresql pod. All the ODM pods are in a Ready state as indicated by a value of 1/1. As an Administrator, it is a good idea to check the pods and to confirm their status and state. If any pods are failing or the ready state is 0/1, as in the postgresql pod, then you know that something might be broken and needs your attention.

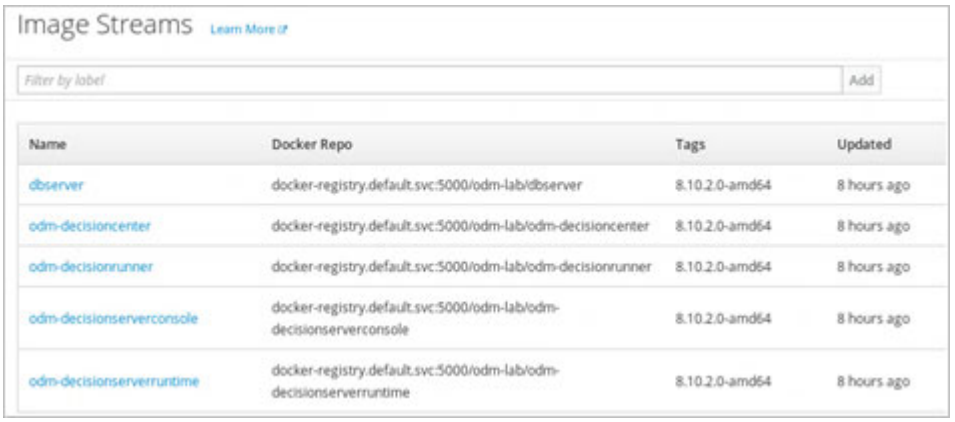


Recall the CLI command **oc get pods** that you ran to check the status of the pods. The output of that command is similar to the one from the web console.

__ f. Click **Builds > Images**.

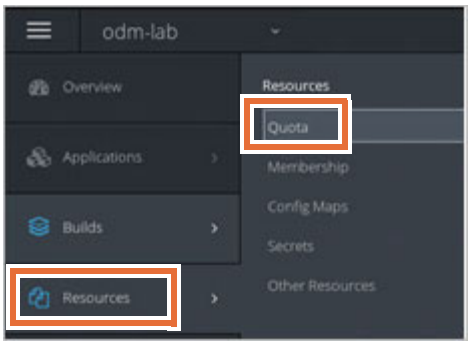


__ g. Verify that five Image Streams are listed. If you recall, in Exercise 1, you ran the docker push command to push the ODM images to the OCP docker repository. Similarly, you also did a docker push for the CPE image in Exercise 2. That image stream is in the cpe-lab project and is not visible here in the odm-lab project. An image stream and its associated tags provide an abstraction for referencing container images from within the OpenShift Container Platform. By using the image stream and its tags, you can see what images are available and ensure that you are using the specific image that you need even if the image in the repository changes. Image streams do not contain actual image data, but present a single virtual view of related images, similar to an image repository.

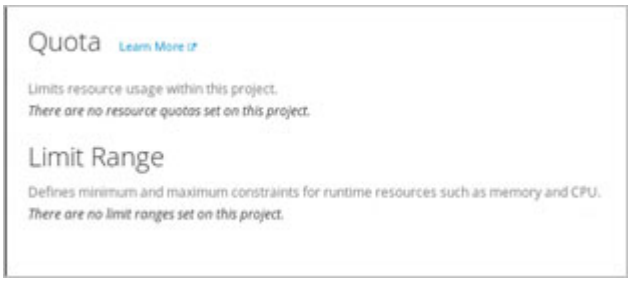


If you used the cpe-lab project, then you see one image stream.

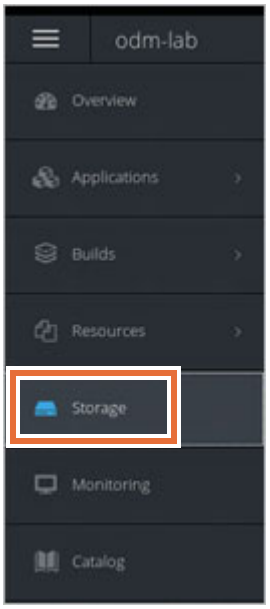
__ h. Click **Resources > Quota**.



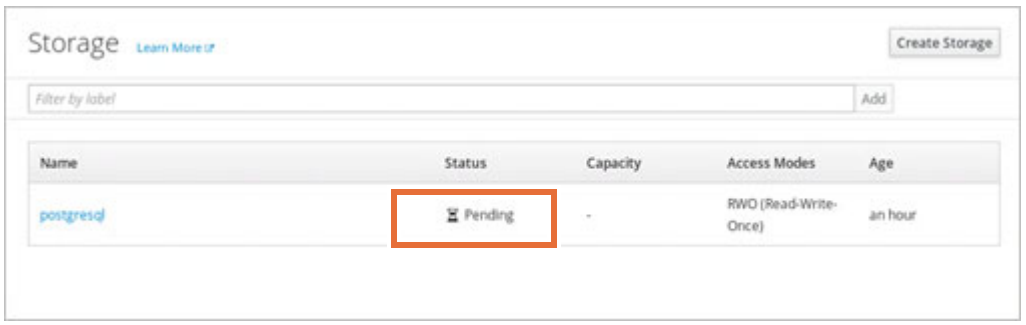
__ i. The resource quotas and limit ranges for the project are displayed. No quotas or limit ranges are defined for this project. A resource quota provides constraints that limit aggregate resource consumption per project. It can limit the quantity of objects that can be created in a project by type, and the total amount of compute resources that may be consumed by resources in that project. A limit range enumerates compute resource constraints in a project at the pod and container level, and specifies the amount of resources that a pod or container can consume. Both of these constraints are defined by the cluster administrator to manage project resources and is done in the cluster console that you explore later.



__ j. Click the **Storage** tab.

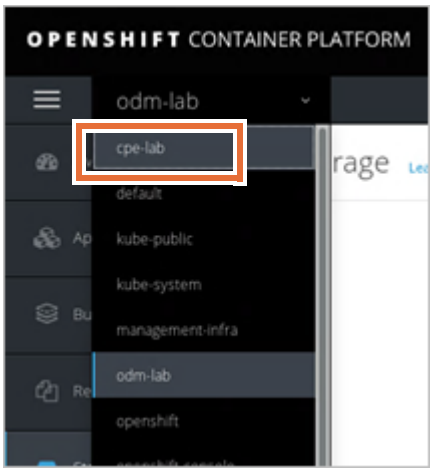


__ k. The Storage tab provides access to storage requests. There is one entry of the persistent volume claim (PVC) for the Postgresql database that you added before. However, the status is in pending status which indicates a possible issue with the PVC.

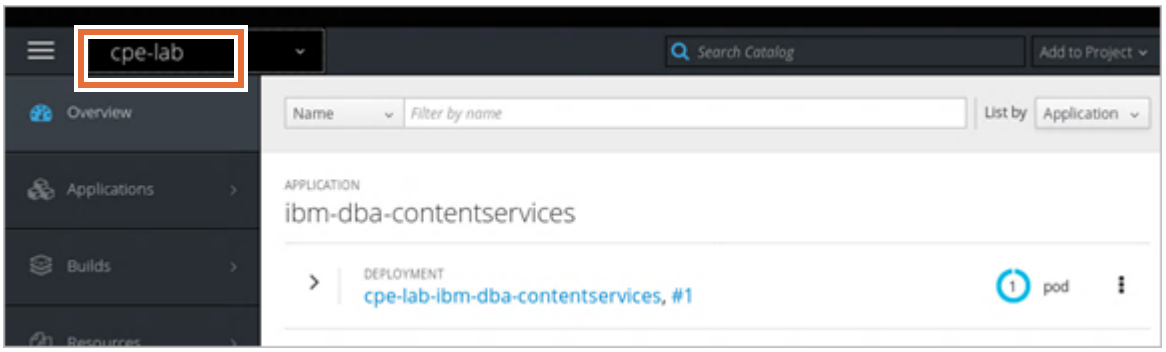


If you used the cpe-lab project, then you see several PVs since you created them in Exercise 2.

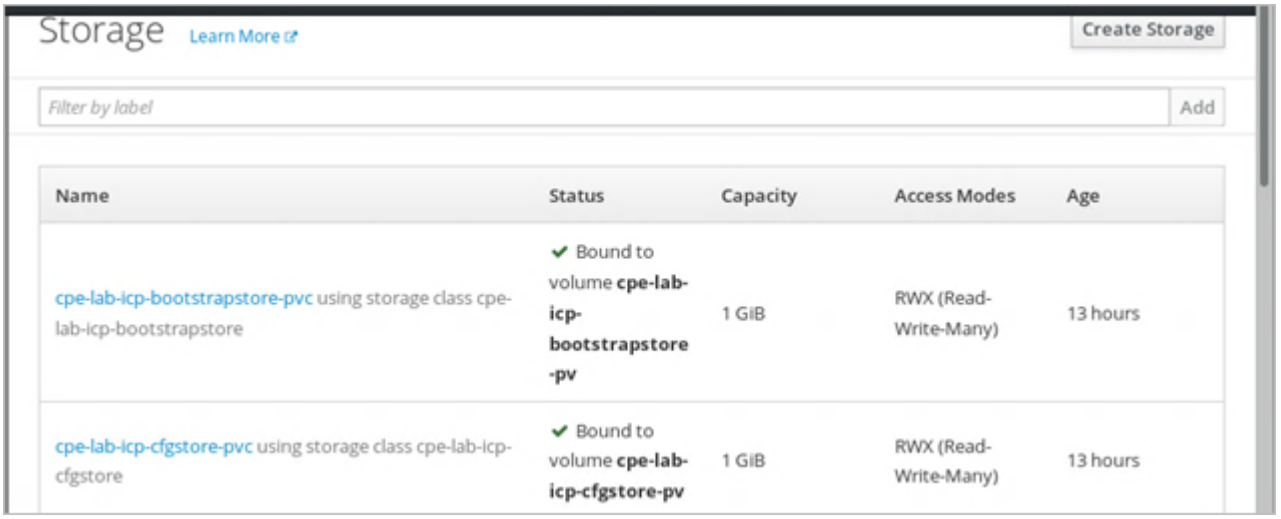
__ l. Recall that you created several persistent volume claims (PVC) for the cpe-lab project. To verify that you can view those in this console, switch to the cpe-lab project by clicking the project list and selecting cpe-lab.



__ m. As you saw earlier, clicking a project, displays the Application overview page by default and lists the single CPE pod.

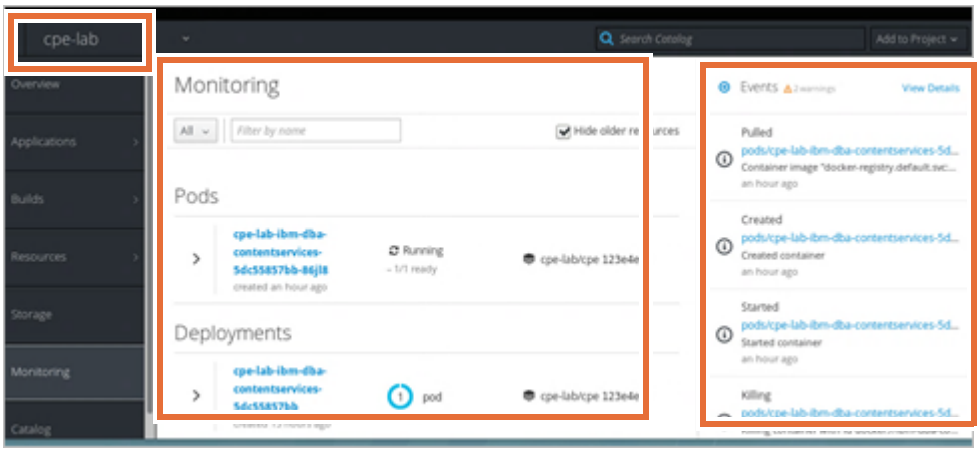


__ n. Click the **Storage** tab for the cpe-lab project. This time you notice several persistent volumes claims listed. You created these PVC by using the command line in the CPE exercise. Since you also created persistent volumes that time, the PVC are bounded and ready for use.



You can create new PVC by clicking Create Storage in the web console instead of using the CLI. It is a matter of personal preference and expertise.

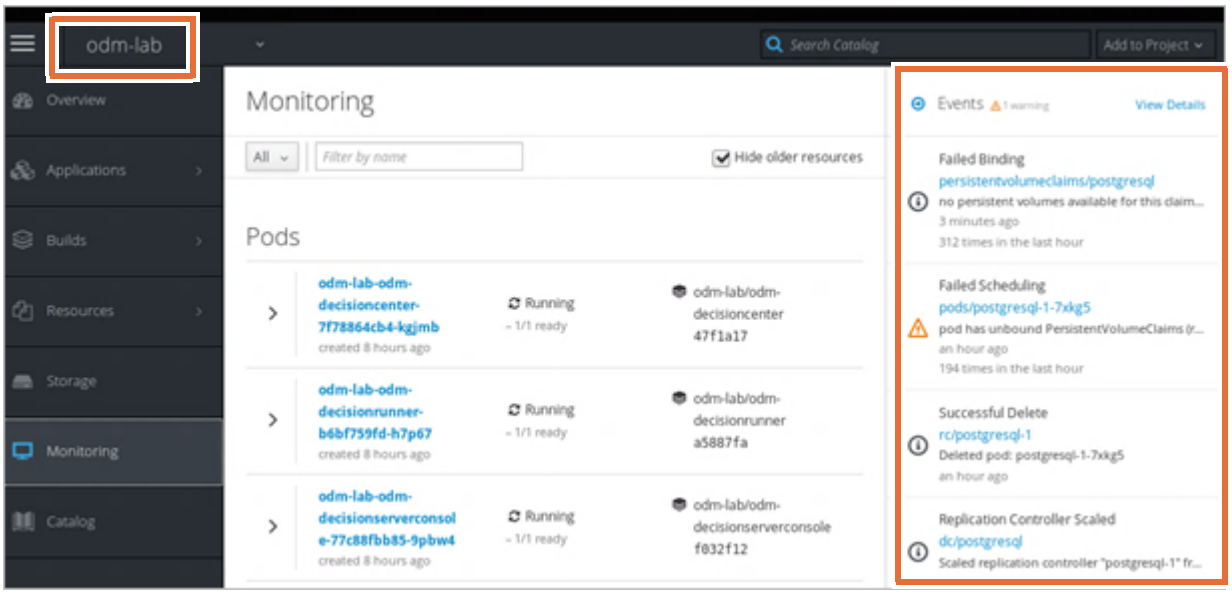
__ o. Click the Monitoring tab. The Monitoring tab provides access to build, deployment, and pod logs. It also provides access to event notifications for the various objects in the project.



__ p. Examine the Events page that opens to the right of the page. Depending on the state of the pods and your environment you might see some events. If you recently started the environment, the cpe and odm containers would be started recently too and there might be some events indicating that. Depending on which project you used to create the PostgreSql service, the events vary between the projects. If you created the Postgresql service in the odm-lab project, then you do not see the error event here in the cpe-lab project. No errors for either the odm or cpe containers exist. Many objects, such as pods and deployments, have their own Events tab as well, which shows events that are related to that object. You explore those events later.

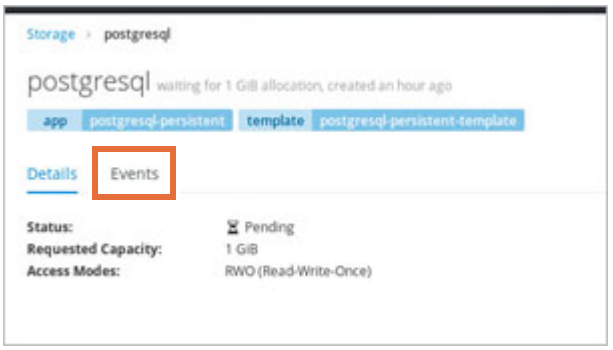
A comprehensive list of events in the OpenShift Container Platform 3.11 is available at `https://docs.openshift.com/container-platform/3.11/dev_guide/events.html`

__ q. Switch back to the odm-lab project and click Monitoring again. This time notice several events in the Events section that are related to the Postgresql service.

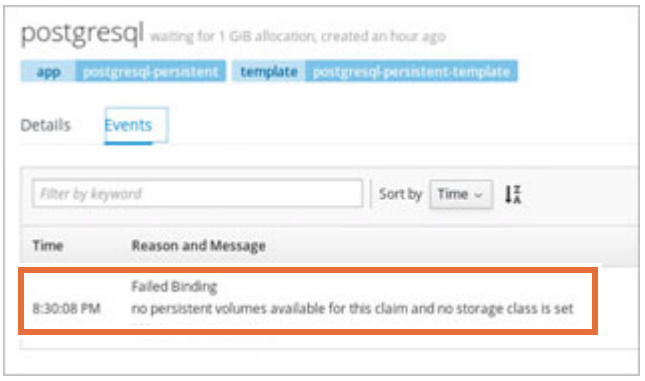


__ r. The postgresql failed binding events are listed in the project in which postgresql service is added. Depending on which project you used to create the Postgresql service, you see the failed event only in that project. These failed events are caused by the failed postgresql deployment, which is part of the project you used. Click the **Failed Binding** event. Optionally, you can also click events to see the details.

__ s. The details show the pending status of postgresql. Click the Events tab.



__ t. The Events tab mentions the reason for the failed binding, which is missing the PV.

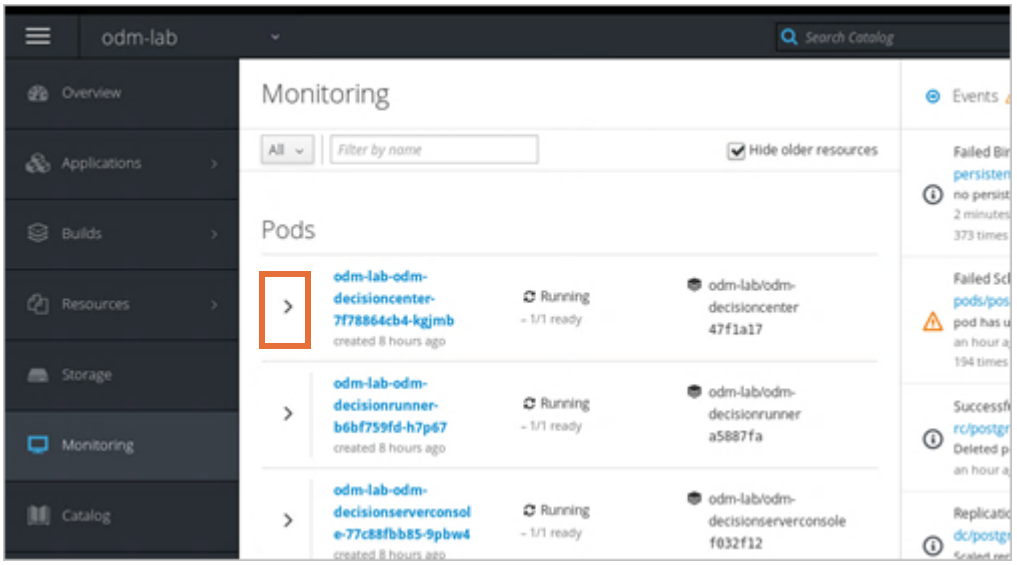


Reading the message, the cluster administrator knows how to resolve the problem. Since the administrator did not create any persistent volume, the persistent volume claim for PostgreSQL remains unbound. As a result, the PostgreSql pod stays in a pending status and eventually fails. To solve this error, the cluster administrator can create PVs using either the CLI or the cluster console that you explore next.

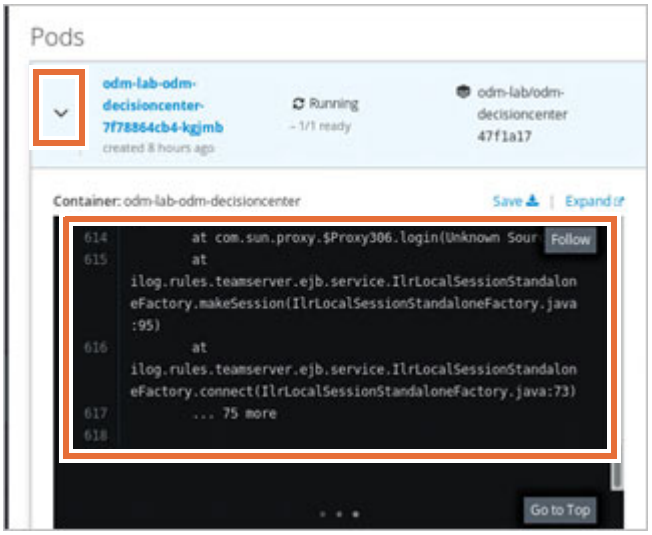
__ 10. Explore the events for the odm pod. (If you skipped the ODM exercise, then you can work with the cpe project and use the cpe pod for the steps below)

__ a. Click the Monitoring tab.

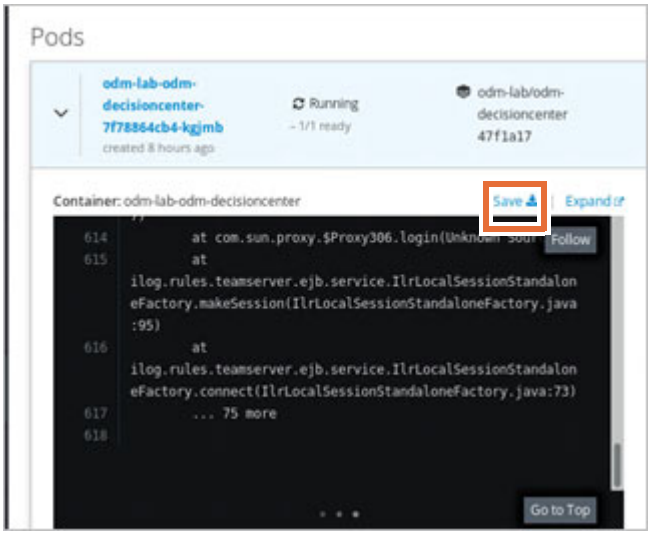
__ b. Click the right arrow to the left of the decision center pod (It is the first one).



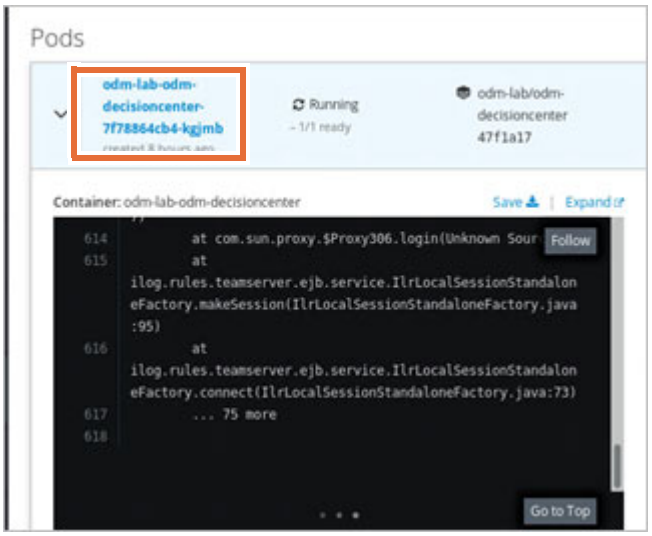
__ c. Clicking the right arrow, turns the direction of the arrow down and opens the decision center container log view.



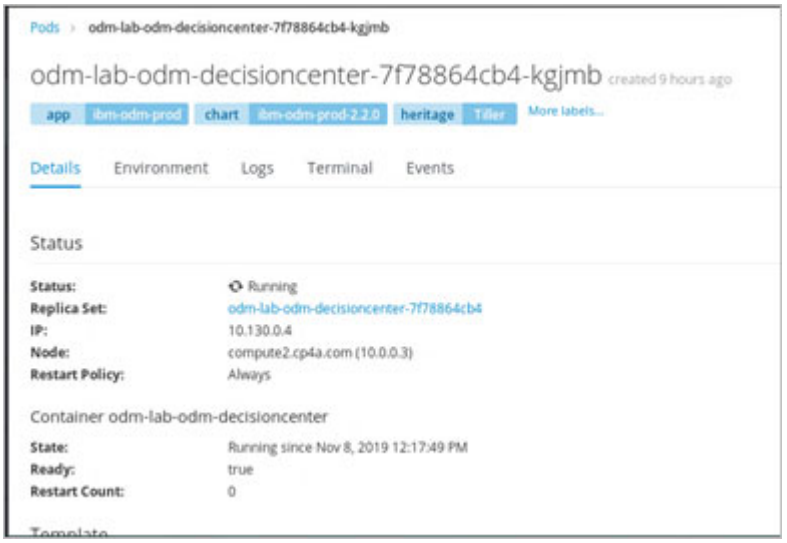
__ d. You can scroll up and down through the logs and examine them. You can also save the logs to your VM when you want to go through them in details by clicking the **Save** link.



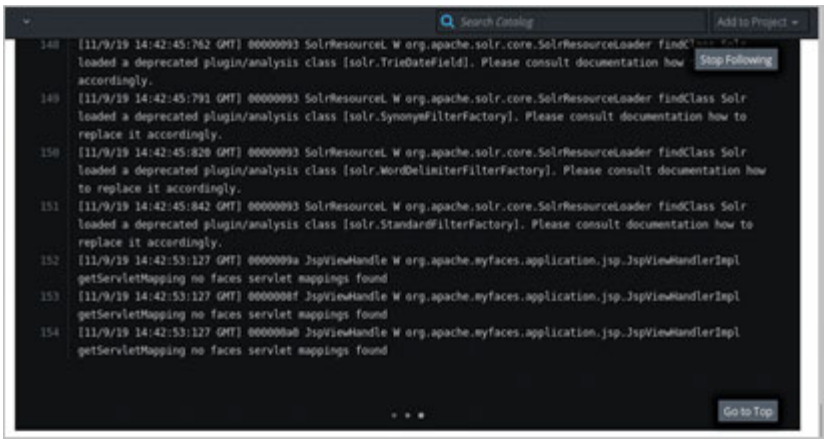
__ e. Click the **odm-lab-decisioncenter** link.



__ f. You are in the Pod Details view. Examine the details listed.

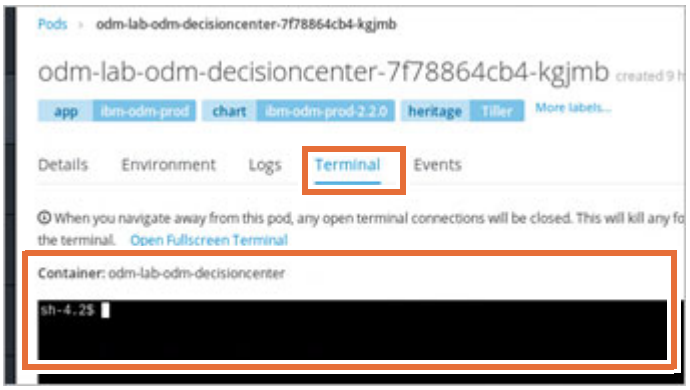


__ g. Click Logs. It displays the same logs that you saw before.

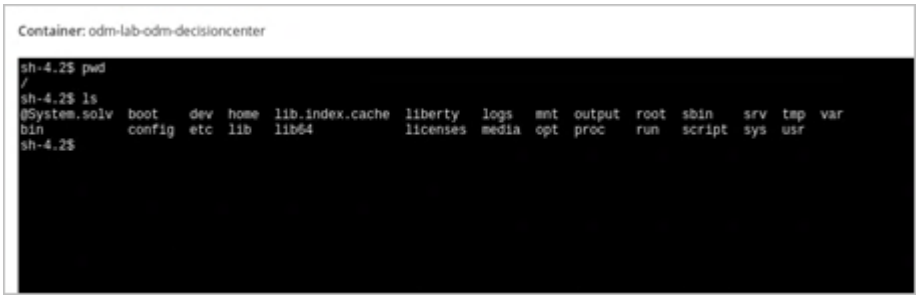


__ h. Click **Go to Top** at the bottom of the logs or scroll up to view the Terminal tab.

__ i. Click Terminal. A shell to the running Decision Center container starts.

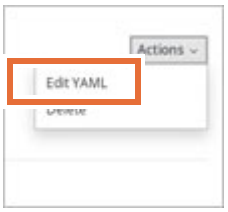


__ j. Feel free to type few sample commands in the shell as a test.

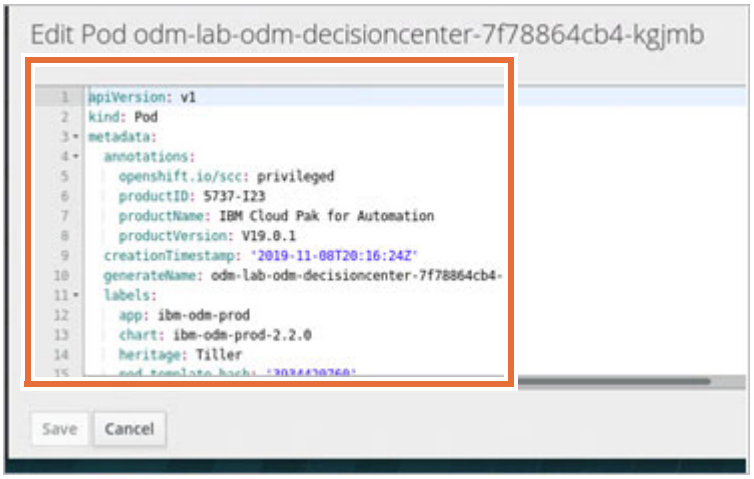


__ k. Click **Details** again and this time scroll down to explore the template.

__ l. Click **Actions > Edit YAML**.



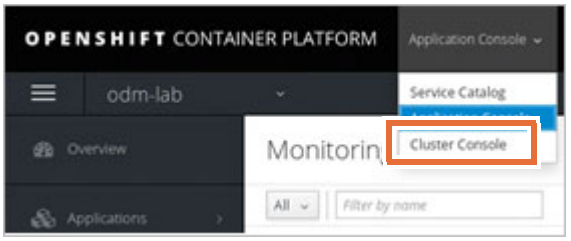
__ m. The yaml file opens. If runtime changes are needed, they can be made here. Feel free to explore. Do not make any changes and click Cancel to close it.



__ 11. Explore the **Cluster Console** view in the web console.

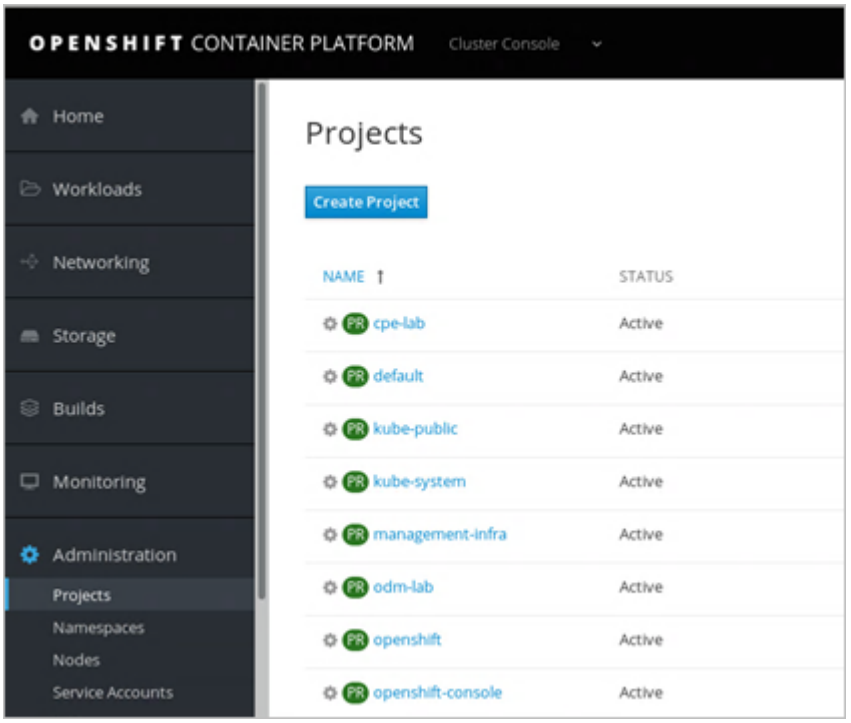
You must be a cluster administrator to access the Cluster Console. Since the admin user is the cluster administrator, you can log in to the Cluster Console.

__ a. Click the down arrow next to Application Console and then click the Cluster Console link.

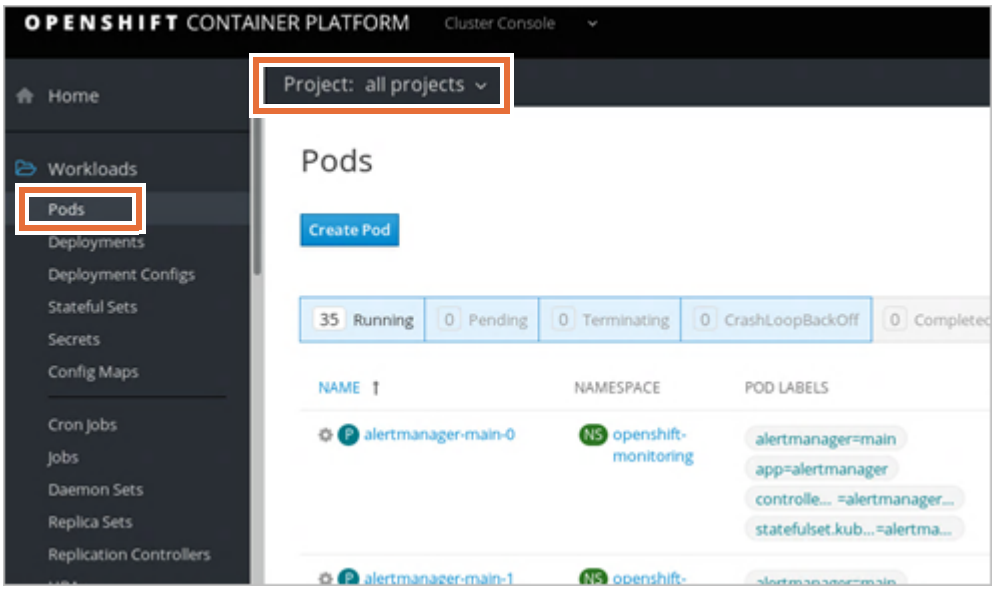


__ b. You might be prompted to log in to the Cluster Console since it is secured for a cluster admin access only. If prompted to log in, use admin / passw0rd credentials to log in.

__ c. Verify that a list of all the Projects for the cluster is displayed. Remember, that since you are in the Cluster console, you can view all the artifacts in the cluster, so all the projects are listed, not just the odm-lab project, or the cpe-lab project.

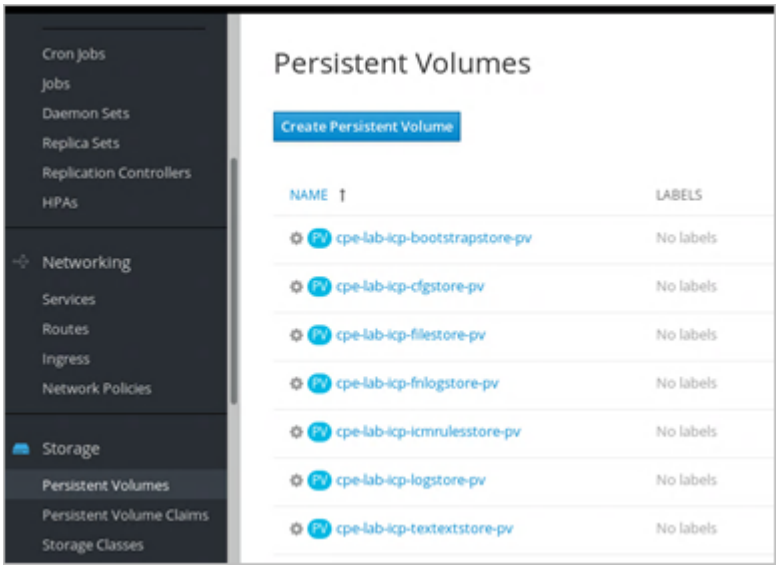


__ d. Click **Workloads > Pods** and then select **all Projects** under the **Project** list to view all the pods for the cluster. The administrator can get an overview of the status of all the pods that are either running, terminating, crashing and so on in the environment and then take any appropriate corrective steps.



__ e. Feel free to click the Deployments, Deployment Configs, Stateful Sets, Secrets, and ConfigMaps under **Workloads** and examine the content that is listed. Also, note the scope of the Project that is displayed at the top; it says all projects. This means that you have a view on every object and artifact in the cluster

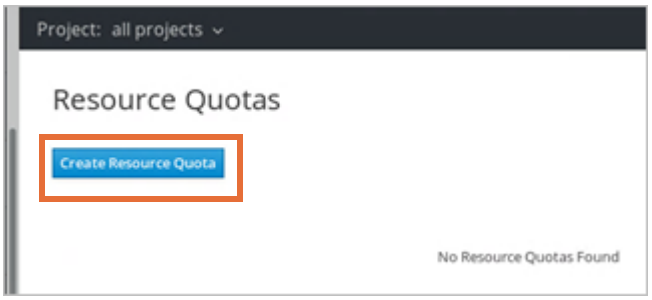
__ f. Scroll down on the left and click **Storage > Persistent Volumes**. The list displays the persistent volumes that you created in the CPE exercise. OpenShift provides both static and dynamic storage management for container data by using the Kubernetes concepts of Persistent Volumes and Persistent Volume Claims. Persistent volumes are OpenShift resources that are created and destroyed only by an OpenShift administrator. A persistent volume resource represents network-attached storage accessible to all OpenShift nodes. The PVs listed here can be requested and used by Persistent Volume Claims of several projects. If any PV is missing or needs edit, the administrator can then create a new PV by clicking Create Persistent Volume.



__ g. Clicking Monitoring displays links for Alerts, Metrics, and Dashboards that can be configured in the OpenShift Cluster. Do not click any of these links as you explore them separately in the next section.

__ h. Expand Administration and click Resource Quotas.

__ i. No Resource Quotas are defined. In an earlier step, you explored the Quota and Limit Ranges for the odm-lab project and it did not display any values. It is because no Resource Quotas were defined by the administrator. Click Create Resource Quota.



__ j. The resource quota can be created for a specific project only and not for all project at the same time. When you click Create Resource Quota, the Project changes to default. You can optionally select which project you want to restrain. In the Create Resource Quota that opens, explore the values that you can configure and limit.

__ k. Click Cancel when done, do not save any changes

Part 2: Examine the available open source monitoring options

The OpenShift metrics subsystem enables the capture and long-term storage of performance metrics for an OpenShift cluster. Metrics are collected for nodes and for all containers that are running in each node.

The metrics subsystem is deployed as a set of containers based on the following open source projects:

Heapster

Collects metrics from all nodes in a Kubernetes cluster and forwards them to a storage engine for long-term storage. Red Hat OpenShift Container Platform uses Hawkular as the storage engine for Heapster. The Heapster project was incubated by the Kubernetes community to provide a way for third-party applications to capture performance data from a Kubernetes cluster.

Hawkular Metrics

Provides a REST API for storing and querying time-series data. The Hawkular Metrics component is part of the larger Hawkular project. Hawkular Metrics uses Cassandra as its data store. Hawkular was created as the successor to the RHQ Project and is a key piece of the middleware management capabilities of the Red Hat CloudForms product.

Hawkular Agent

Collects custom performance metrics from applications and forwards them to Hawkular Metrics for storage. The application provides metrics to the Hawkular Agent. The Hawkular OpenShift Agent (HOSA) is a technology preview features and is not installed by default Red Hat does not provide support for technology preview features and does not recommend using them for production.

Cassandra

Stores time-series data in a non-relational, distributed database.

Prometheus and Grafana

In addition to the open source projects mentioned above, OpenShift Container Platform ships with a pre-configured and self-updating monitoring stack that is based on the Prometheus open source project and its wider ecosystem. It provides monitoring of cluster components and ships with a set

of alerts to immediately notify the cluster administrator about any occurring problems and a set of Grafana dashboards. The Prometheus Operator (PO) creates, configures, and manages Prometheus and Alertmanager instances. It also automatically generates monitoring target configurations based on familiar Kubernetes label queries. OpenShift Container Platform Monitoring ships with a Prometheus instance for cluster monitoring and a central Alert manager cluster.

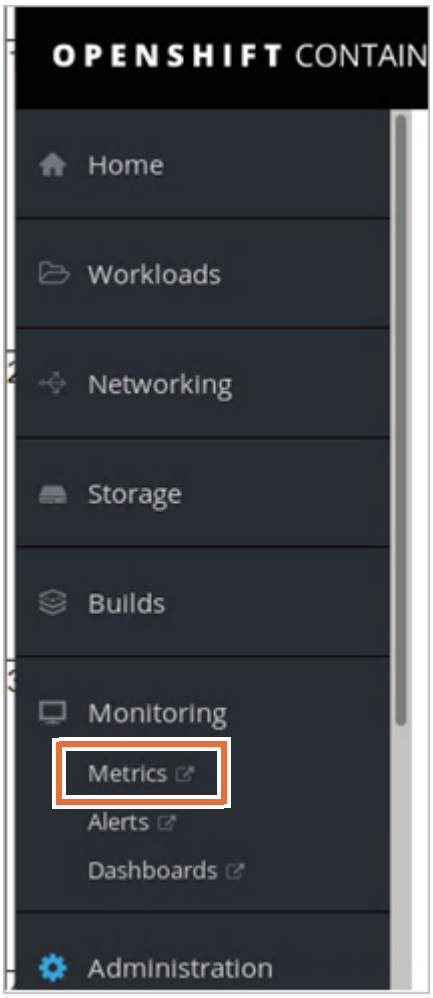
The OpenShift metrics subsystem works independently of other OpenShift components. Red Hat OpenShift Container Platform does not force an organization to deploy the full metrics subsystem. If an organization already has a monitoring system and wants to use it to manage an OpenShift cluster, there is the option of deploying only the Heapster component and to delegate long-term storage of metrics to the external monitoring system.

Configuring these metrics is beyond the scope of this exercise. As an Administrator, one needs to be careful in how these monitoring capabilities are configured as it can impact the performance of the cluster.

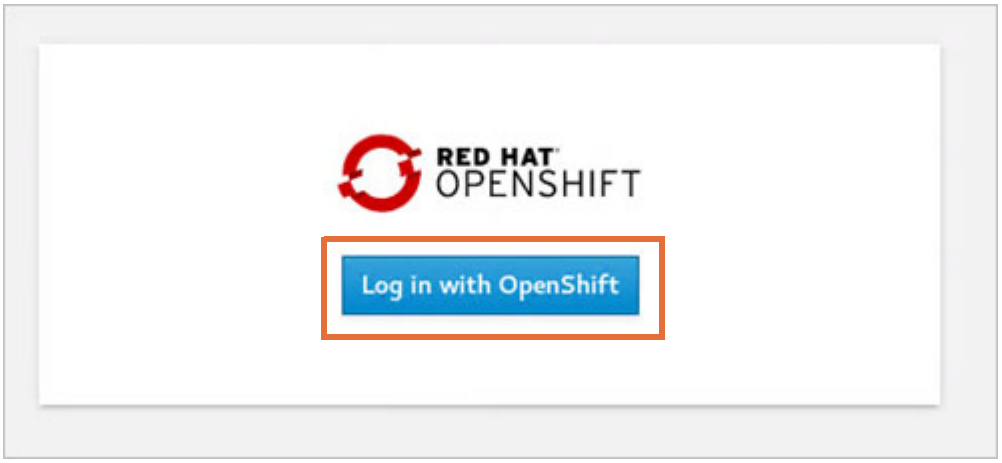
__ 1. Explore Monitoring metrics.

__ a. Make sure that you are in the Cluster Console view of the web console.

__ b. Click **Monitoring > Metrics**.

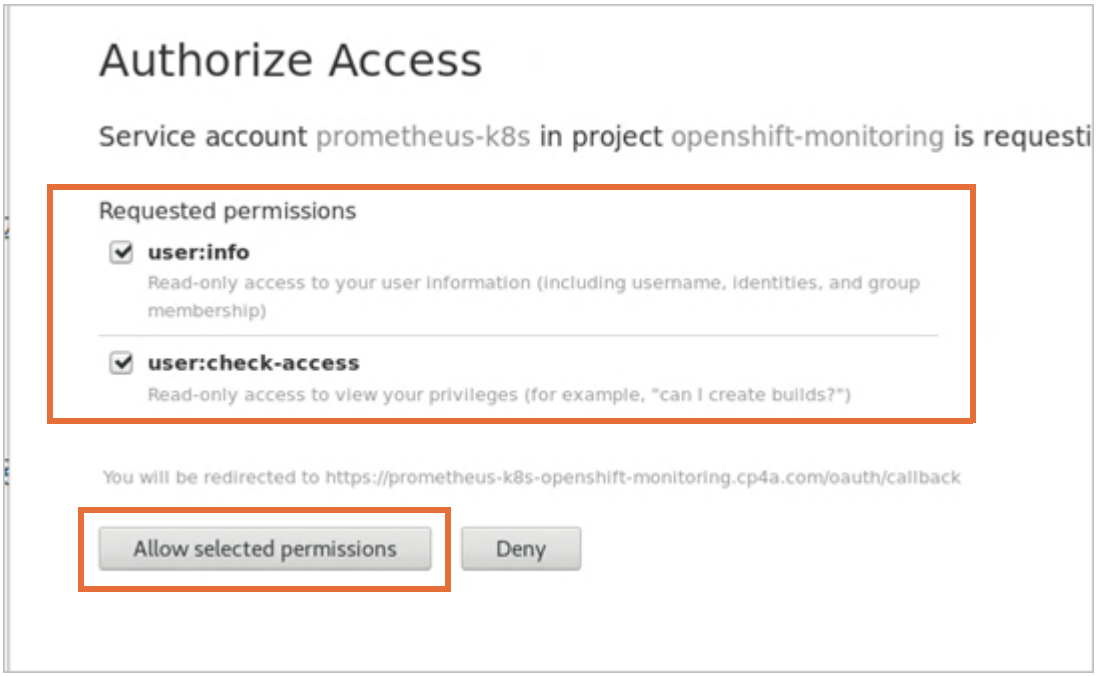


__ c. In the new browser tab that opens, click **Log in with OpenShift**.

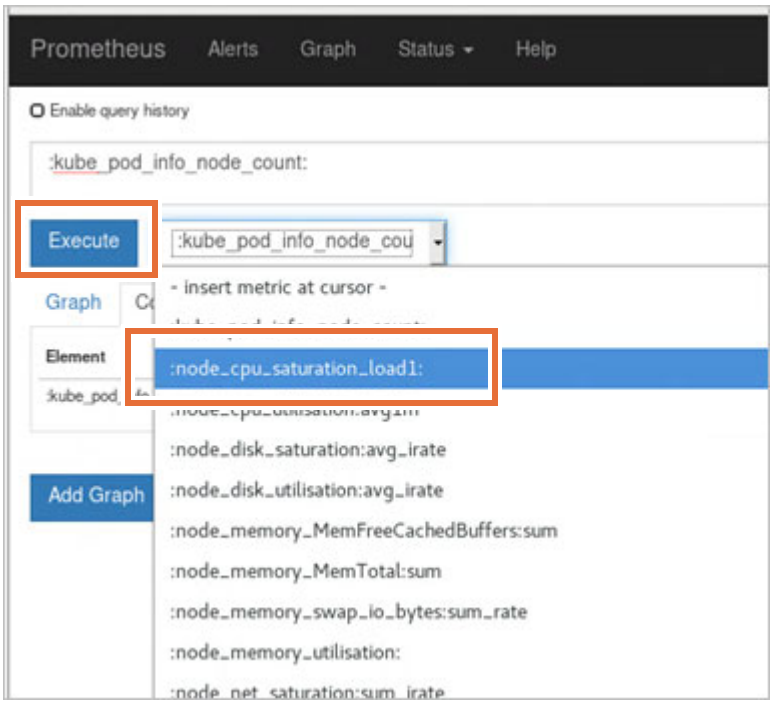


__ d. Enter `admin` in the **Username** field and enter `passw0rd` in the **Password** field. Click **Log In**.

__ e. Leave the default options selected and click **Allow selected permissions** to give access to the admin account.

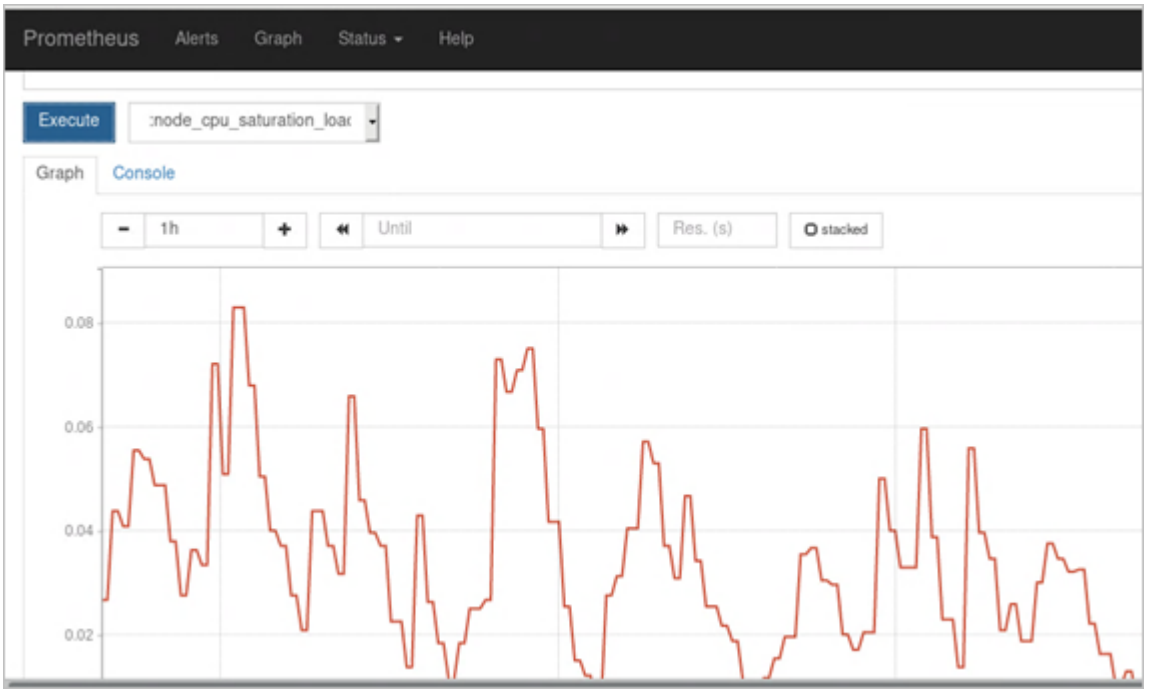


__ f. In the page that opens, select `node.cpu_saturation_load1` and click **Execute**.



__ g. Click the Graph tab.

__ h. Examine the graph that opens in the Graph tab.

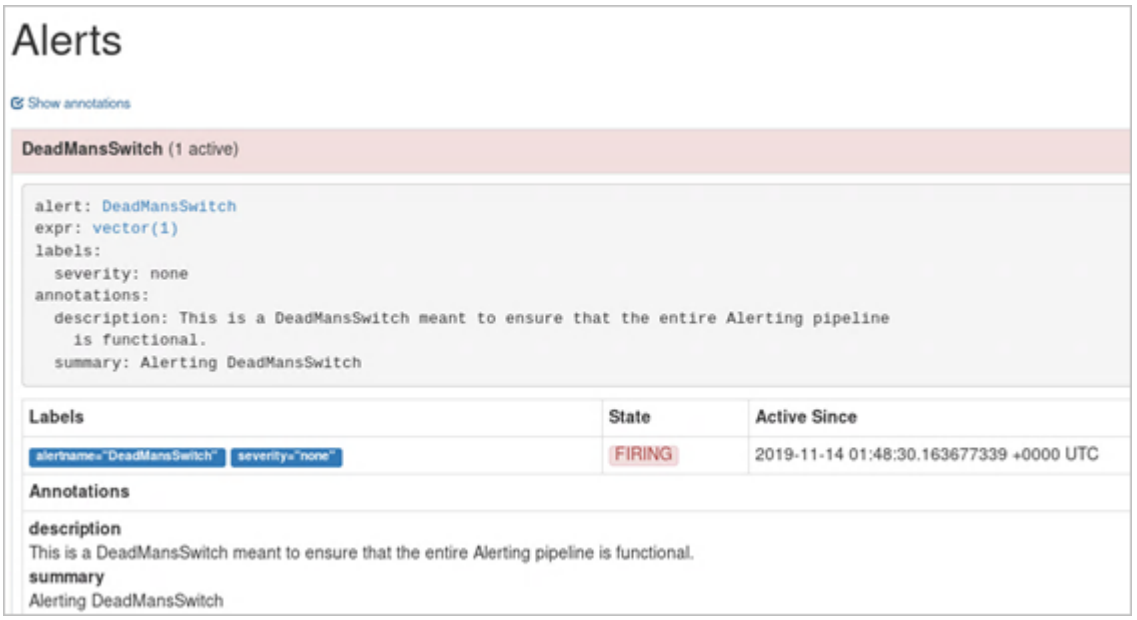


__ i. Click any specific point on the graph to view the CPU saturation load for that moment.

__ j. Next, select `node.cpu_utilization_avg1` from the list and click Execute. See how the graph changes for the average utilization of the CPU.

__ k. Click Alerts from the menu at the top.

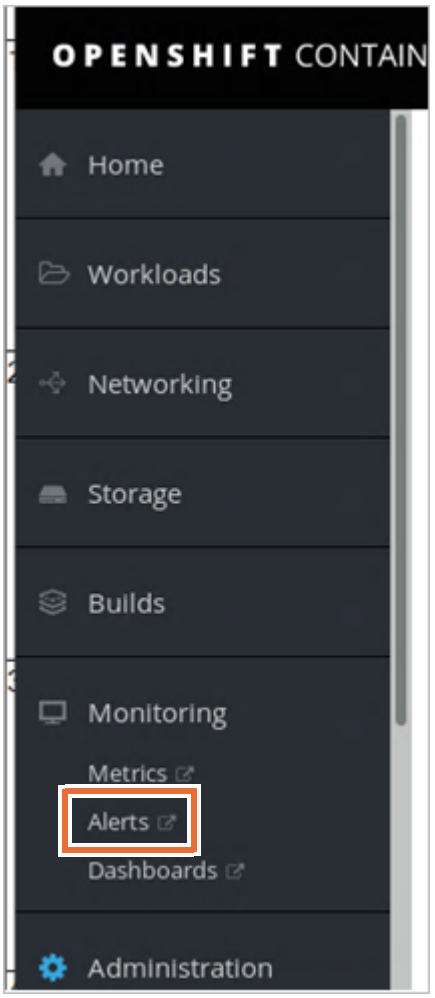
__ l. The Alert manager manages incoming alerts; this includes silencing, inhibition, aggregation, and sending out notifications through methods such as email, PagerDuty, and HipChat. Click the **DeadMansSwitch** under alerts. OpenShift Container Platform Monitoring ships with a dead man's switch alert to ensure the availability of the monitoring infrastructure. The **DeadMansSwitch** is a Prometheus alerting rule that always triggers. The Alert manager continuously sends notifications for the dead man's switch to the notification provider that supports this function. This also ensures that communication between the Alert manager and the notification provider is working. Several alerts are preinstalled by default.



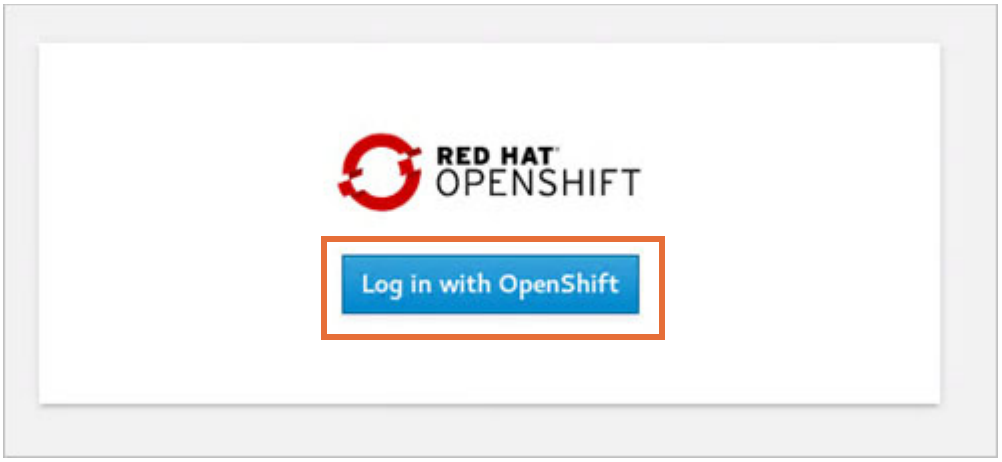
__ m. Feel free to click the other menu options. For an explanation of all the alert rules, see Red Hat OpenShift documentation.

__ 2. Explore Monitoring alerts.

__ a. Switch back to the cluster console view tab and click **Monitoring > Alerts**.

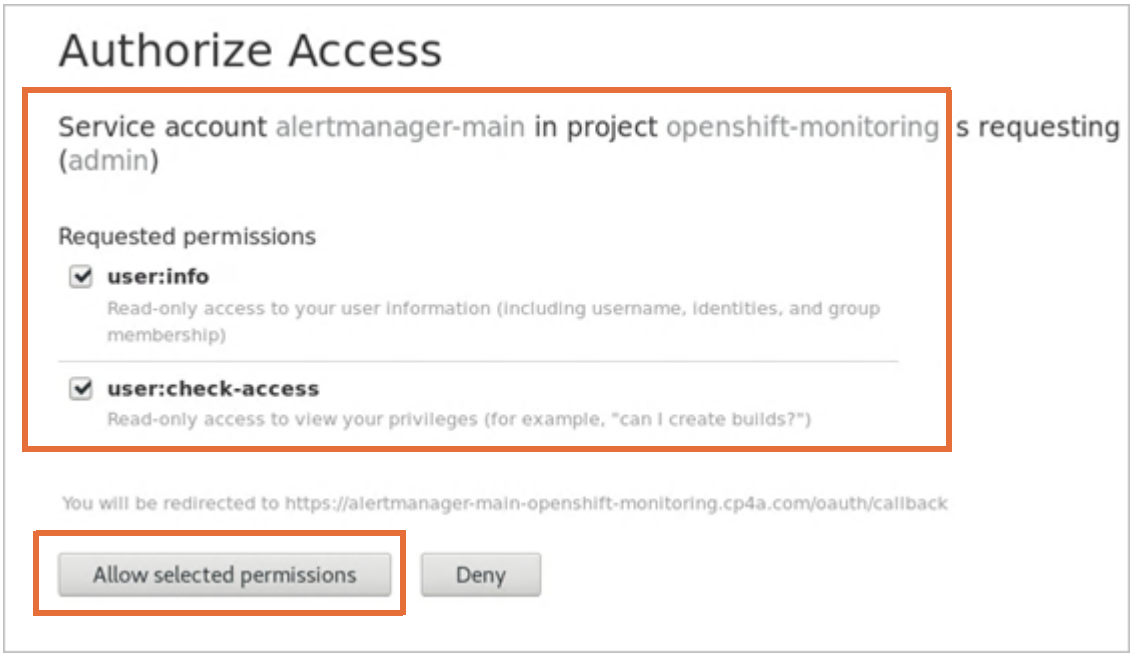


__ b. In the new browser tab that opens, click **Log in with OpenShift**.



__ c. When the login page opens, enter `admin` in the **Username** field and enter `passw0rd` in the **Password** field. Click **Log In**.

__ d. Leave the default options selected and click **Allow selected permissions** to give access to the admin account.



__ e. In the page that opens, there are three tabs - Alerts, Silences, and Status.

The Alert manager handles alerts that are sent by client applications such as the Prometheus server. It takes care of duplicating, grouping, and routing them to the correct receiver integration such as email, PagerDuty, or OpsGenie. It also takes care of silencing and inhibition of alerts.

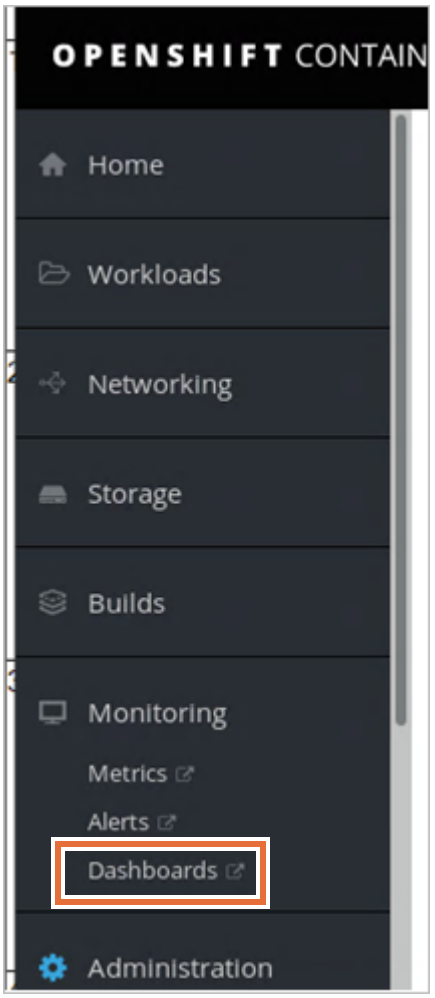
Silences are configured in the web interface of the Alert manager. Silences are a straightforward way to mute alerts for a given time. A silence is configured based on matchers, just like the routing tree. Incoming alerts are checked whether they match all the equality or regular expression matchers of an active silence. If they do, no notifications are sent out for that alert.

The Status displays the targets that are being monitored.

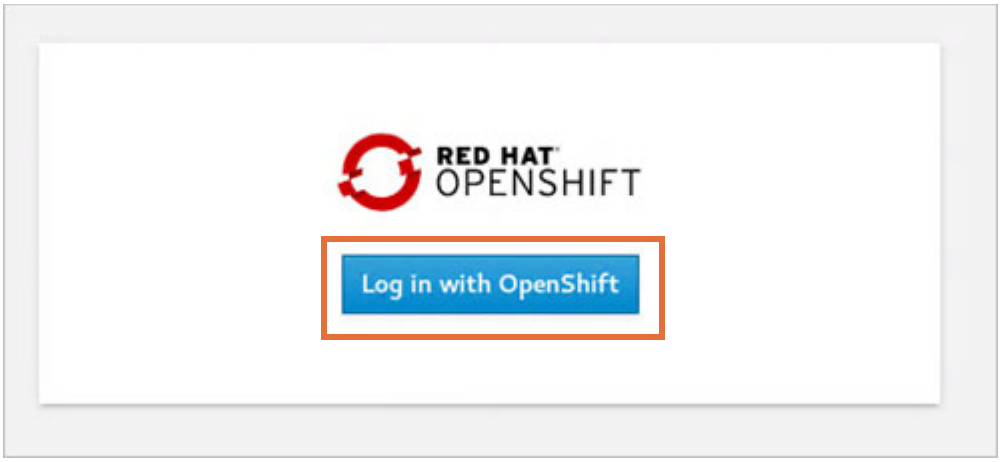
You do not do any Alert manager configuration.

__ 3. Explore Monitoring Dashboards.

__ a. Switch back to the cluster console view tab and click **Monitoring > Dashboards**.

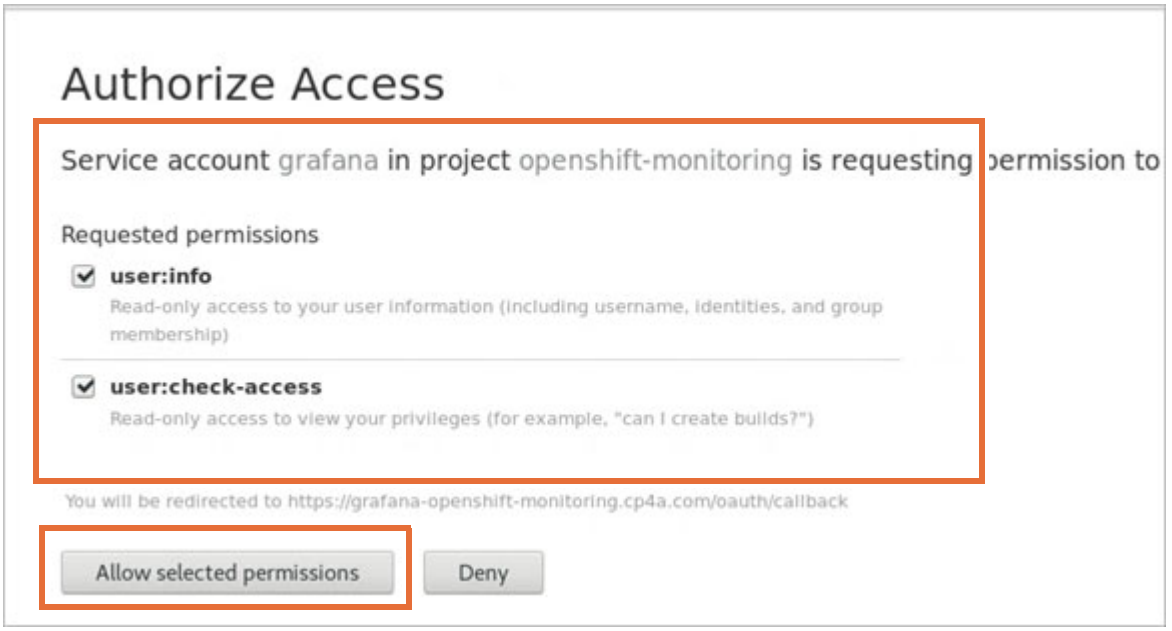


__ b. In the new browser tab that opens, click **Log in with OpenShift**.

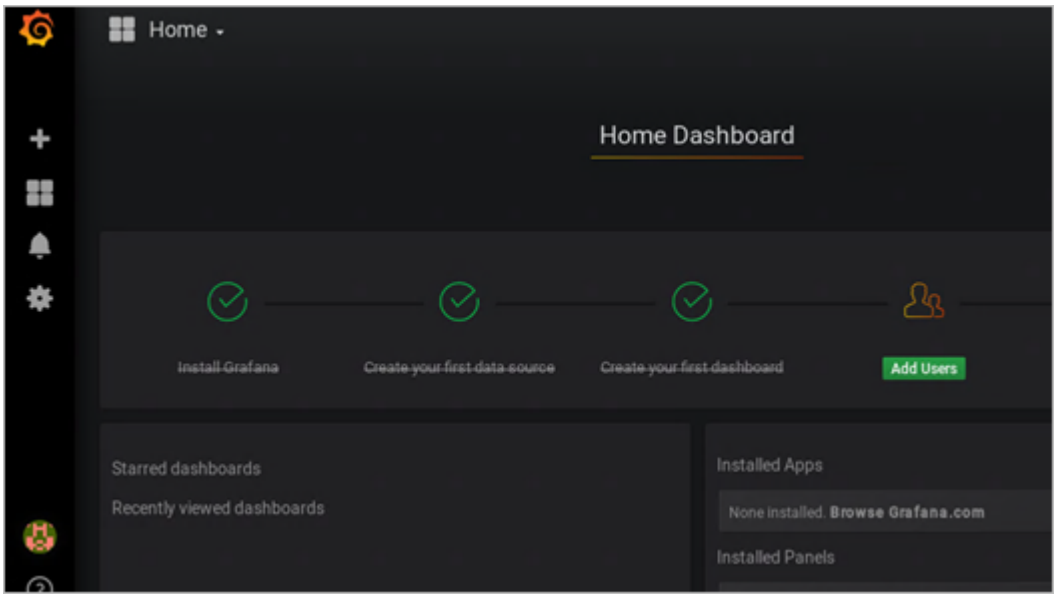


__ c. When the login page opens, enter `admin` in the **Username** field and enter `passw0rd` in the **Password** field. Click **Log In**.

__ d. Leave the default options selected and click **Allow selected permissions** to give access to the admin account.

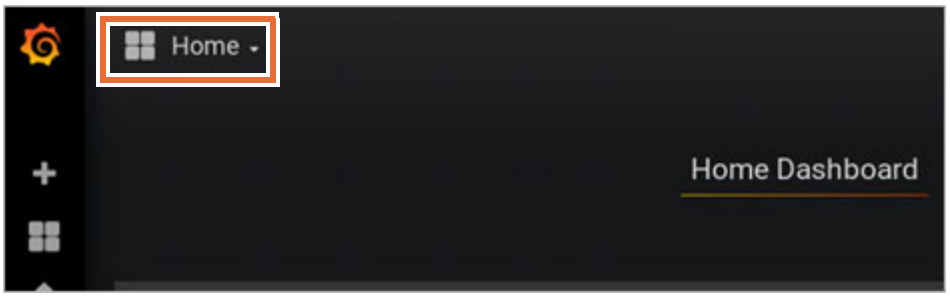


__ e. Verify that the Home Dashboard page opens.

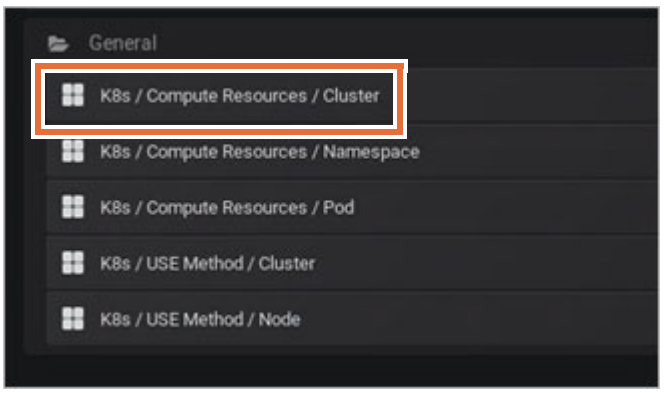


In addition to Prometheus and Alert manager, OpenShift Container Platform Monitoring also includes a Grafana instance and pre-built dashboards for cluster monitoring troubleshooting. The Grafana instance that is provided with the monitoring stack, along with its dashboards, is read-only.

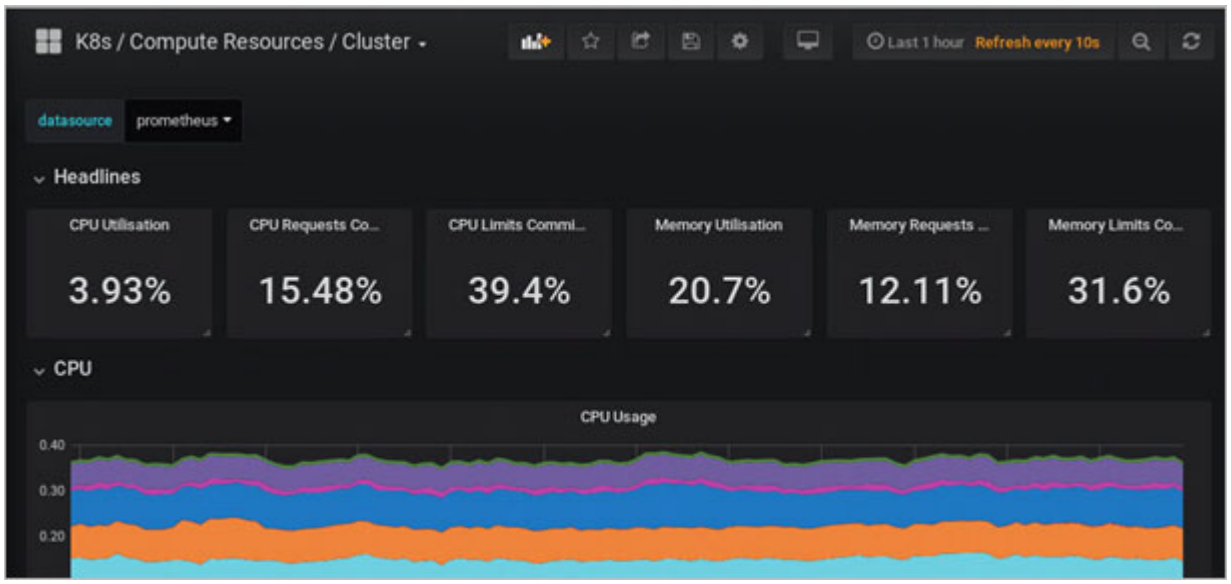
__ f. Click **Home** at the upper left of the page.



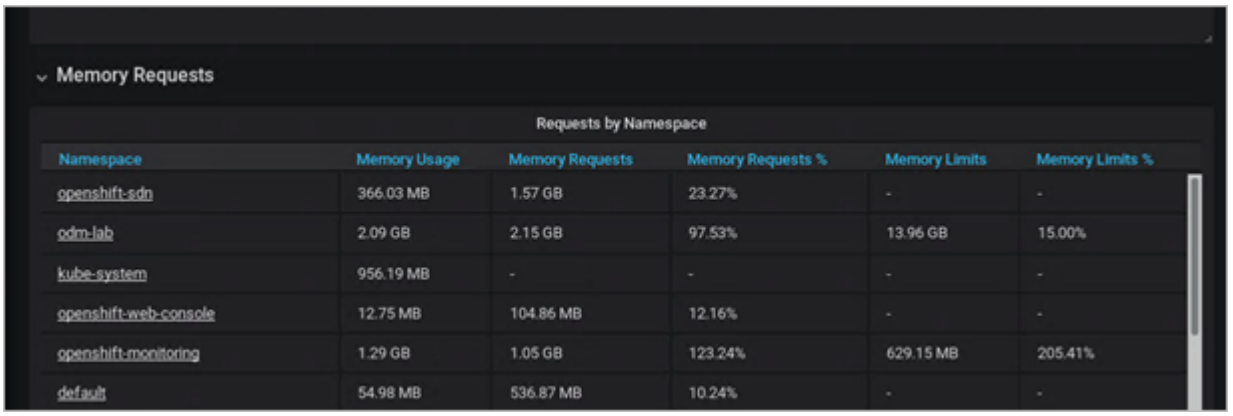
__ g. Click the link for K**8s / Compute Resources/Cluster** to view the resources for your OpenShift cluster.



__ h. The cluster resource page opens. Examine the CPU and memory requests by cluster.



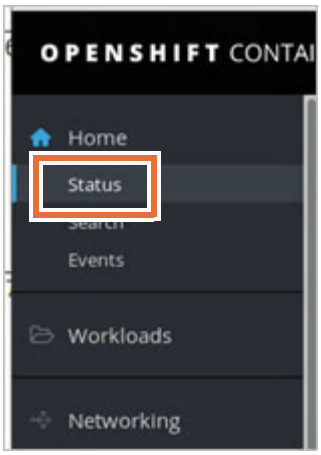
__ i. Scroll down to view the requests broken down by namespace.



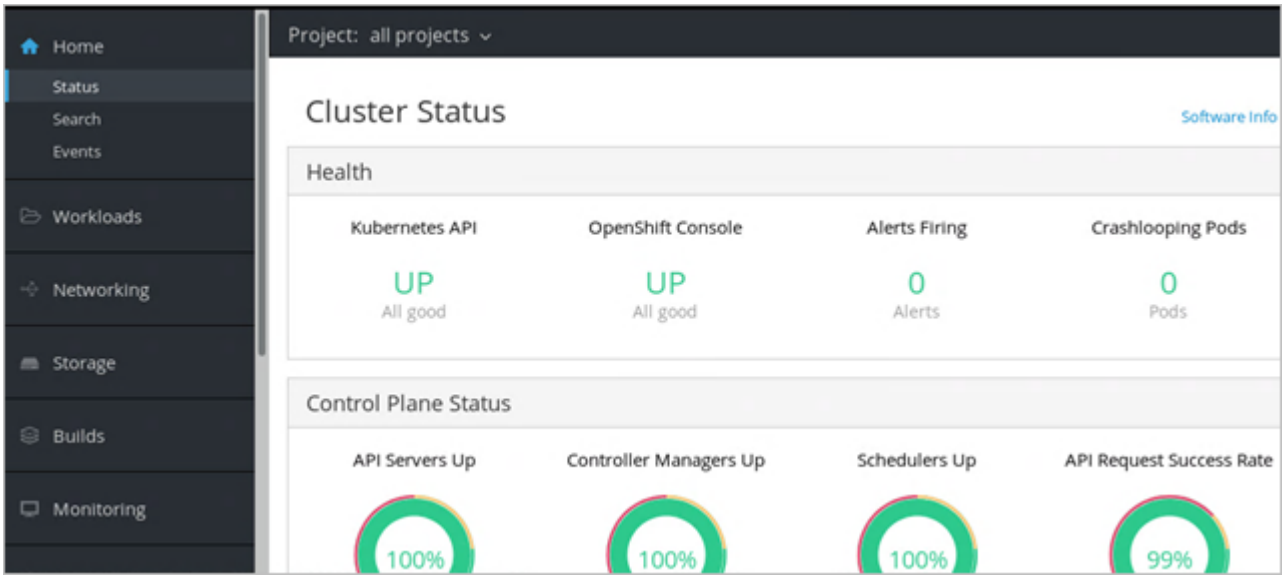
__ j. Feel free to continue to explore the grafana dashboard.

__ 4. Explore the Home dashboard. Now that you explored the Monitoring tab, there is one more dashboard in the cluster console that is of great interest to the Administrator.

__ a. Switch back to the Cluster Console tab and click **Home > Status**.

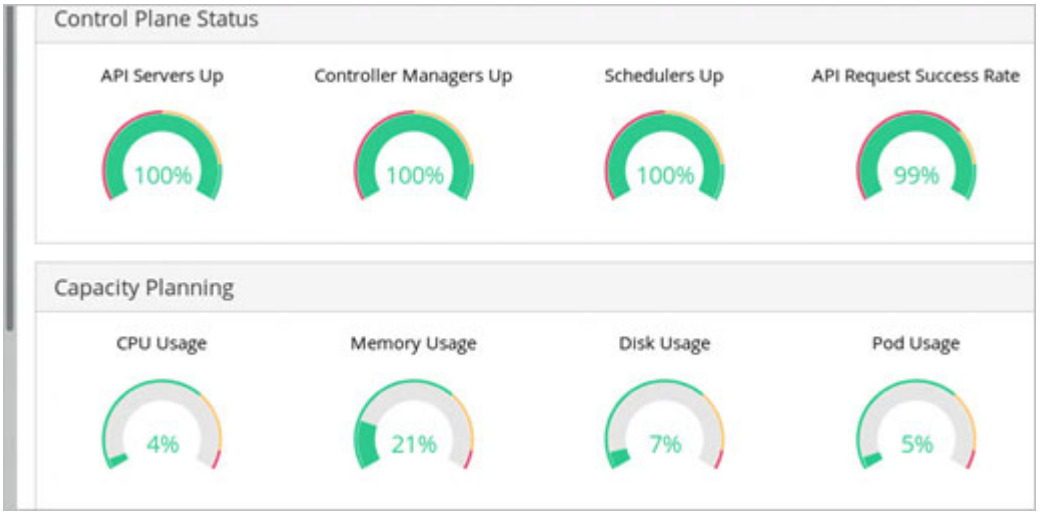


__ b. The cluster status page for all the projects is displayed and provides a high-level view of your cluster.

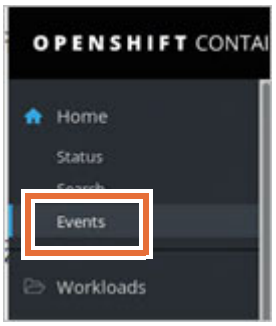


The administrator can check the health of the cluster by examining this view. It shows the API server is up and all is good. The number of alerts being fired is listed here also along with any crashing pods.

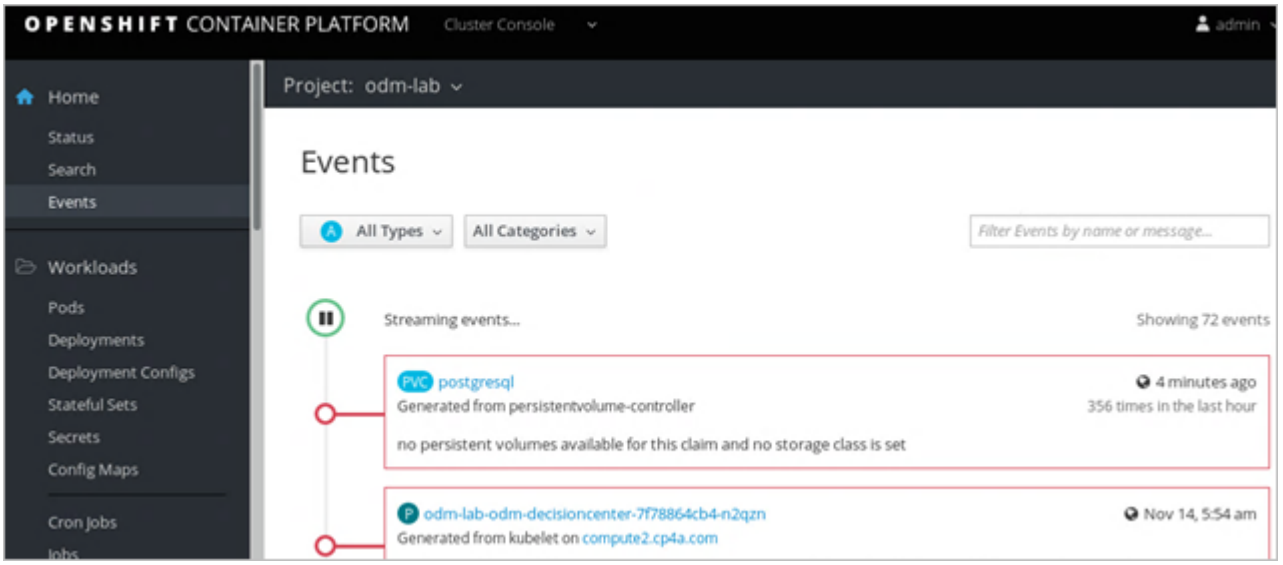
__ c. Scroll down to see the Capacity planning section. it displays the CPU Usage, Memory Usage, Disk Usage and Pod Usage. The administrator can use this data to investigate any problem areas in the cluster and address those issues.



__ d. Click **Home > Events** on the left pane.



__ e. The Events page displays events that are being generated for all projects. If you switch to the odm-lab or cpe-lab project, you can see the failed events from Postgresql.



Part 3: Scale an application

As part of managing application deployments, an administrator must be able to control the number of replicas of pods. Replication controller guarantees that the specified number of replicas of a pod are running always.

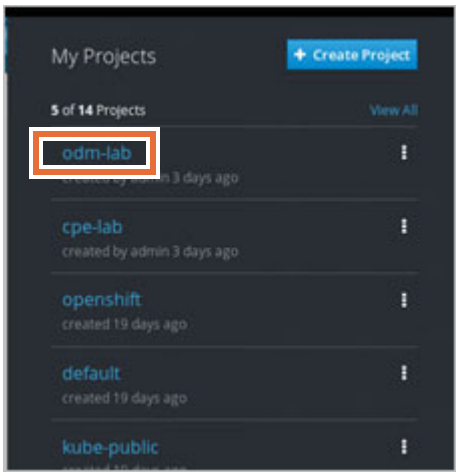
The replication controller instantiates more pods if pods are killed, or are deleted explicitly by an administrator. Similarly, it deletes pods as necessary to match the specified replica count, if there are more pods running than the desired count. Although Kubernetes administrators usually manage replication controllers directly, the recommended approach for OpenShift users is to manage a deployment configuration that creates or changes replication controllers on demand.

In this section, you use both the CLI and web console to view and manage replicas.

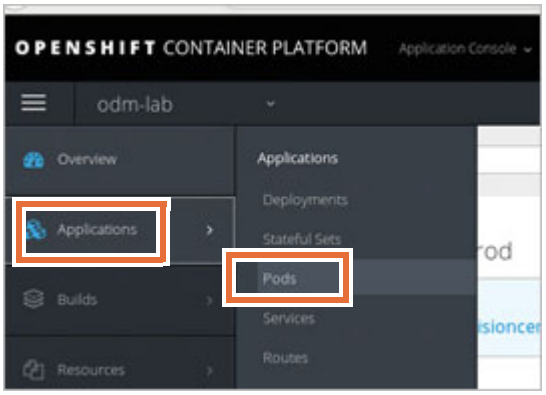
__ 1. Change the number of replicas for an application.

__ a. Switch to the **Service Catalog** view at the top.

__ b. In the My Projects on the right, click the odm-lab project.



__ c. The view switches to the Application console view. Click **Applications > Pods** to view the pods for the project.



__ d. For the dm-lab project, there are four pods running and one failed postgresql pod. If you are using the cpe-lab project, then the number of pods varies.

__ e. Leave the web console open and open a new terminal and login to the OpenShift cluster. The OpenShift console URL is https://console.cp4a.com:8443 and the username/password is admin/passw0rd.

```
oc login https://console.cp4a.com:8443 -u admin -p passw0rd
```

__ f. Check the current project being used in the cluster.

```
oc project
```

__ g. If the current project is cpe-lab, then switch to the odm-lab project.

```
oc project odm-lab
```

__ h. Check the status of the pods and compare the result with the one in the web console. The number of pods is the same.

```
oc get pods
```

```
[root@master ~]# oc get pods
NAME                                                READY     STATUS    RESTARTS   AGE
odm-lab-odm-decisioncenter-7f78864cb4-kgjmb         1/1       Running   0          9h
odm-lab-odm-decisionrunner-b6bf759fd-h7p67          1/1       Running   0          9h
odm-lab-odm-decisionserverconsole-77c88fbb85-9pbw4  1/1       Running   0          9h
odm-lab-odm-decisionserverruntime-78c44b4547-zsgrj  1/1       Running   0          9h
postgresql-1-deploy                                 0/1       Error     0          3h
[root@master ~]#
```

__ i. Run the command with the -o wide option to view the internal IP address and the node of each running pod.

```
oc get pods -o wide
```

```
[root@master ~]# oc get pods -o wide
NAME                                                READY     STATUS    RESTARTS   AGE       IP            NODE
NOMINATED NODE
odm-lab-odm-decisioncenter-7f78864cb4-kgjmb         1/1       Running   0          9h        10.130.0.4    compute2.cp4a.com
<none>
odm-lab-odm-decisionrunner-b6bf759fd-h7p67          1/1       Running   0          9h        10.129.0.4    compute1.cp4a.com
<none>
odm-lab-odm-decisionserverconsole-77c88fbb85-9pbw4  1/1       Running   0          9h        10.130.0.3    compute2.cp4a.com
<none>
odm-lab-odm-decisionserverruntime-78c44b4547-zsgrj  1/1       Running   0          9h        10.129.0.5    compute1.cp4a.com
<none>
postgresql-1-deploy                                 0/1       Error     0          3h        10.129.0.6    compute1.cp4a.com
<none>
[root@master ~]#
```

__ j. The cluster administrator may decide to remove the failed postgresql pod. Before continuing, delete that pod. To delete, you use the name of the postgresql pod.

```
oc delete pod postgresql-1-deploy
```

```
[root@master deploy]# oc get pods -o wide
NAME                                                READY     STATUS    RESTARTS
DE
odm-lab-odm-decisioncenter-7f78864cb4-kgjmb         1/1       Running   0
odm-lab-odm-decisionrunner-b6bf759fd-h7p67          1/1       Running   0
odm-lab-odm-decisionserverconsole-77c88fbb85-9pbw4  1/1       Running   0
odm-lab-odm-decisionserverruntime-78c44b4547-zsgrj  1/1       Running   0
postgresql-1-deploy                                 0/1       Error     0
[root@master deploy]# oc delete pod postgresql-1-deploy
pod "postgresql-1-deploy" deleted
```

__ k. Verify that the pod is deleted.

```
oc get pods -o wide
```

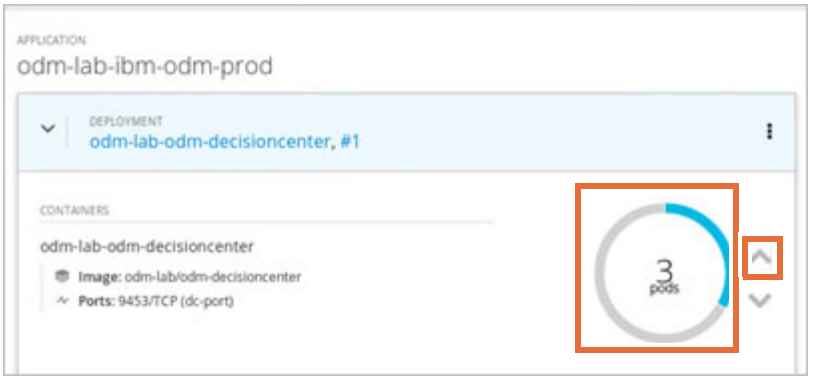
```
[root@master deploy]# oc get pods -o wide
NAME                                                READY     STATUS    RESTARTS
E
odm-lab-odm-decisioncenter-7f78864cb4-kgjmb         1/1       Running   0
odm-lab-odm-decisionrunner-b6bf759fd-h7p67          1/1       Running   0
odm-lab-odm-decisionserverconsole-77c88fbb85-9pbw4  1/1       Running   0
odm-lab-odm-decisionserverruntime-78c44b4547-zsgrj  1/1       Running   0
[root@master deploy]#
```

__ l. Leave the terminal open and switch to the web console and click the Overview tab.

__ m. In the pods that are displayed, click the right arrow next to Decision Center pod.

__ n. The view is expanded and the arrow is pointed down. The view shows the Decision Center pod that is running. Note that the number of pods is 1.

__ o. You can click the up arrow or down arrow next to the blue donut to increase or decrease the number of pods. Click the up arrow twice to scale the number of pods to 3.

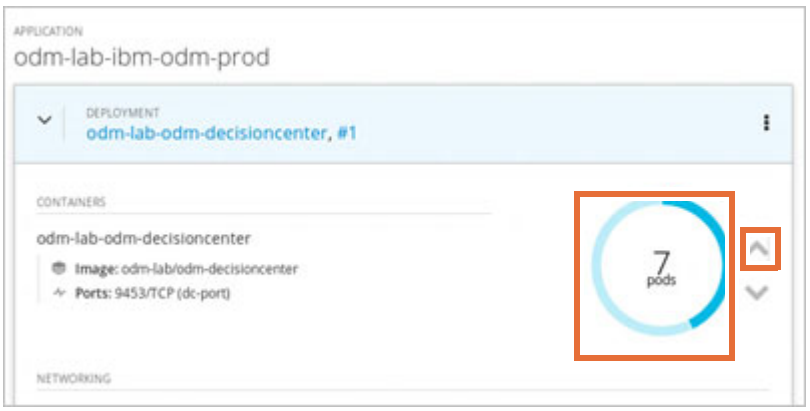


__ p. Switch back to the terminal and run the command again to view the number of pods. This time notice the two new pods start. Take a note of the compute nodes where the new pods are running. It might be either that both the new pods are running on the same node or they run on different ones.

```
oc get pods -o wide
```

```
odm-lab-odm-decisioncenter-7f78864cb4-kgjmb          1/1     Running   0          11h     10.130.0.4    compute2.cp4a.com
odm-lab-odm-decisioncenter-7f78864cb4-m5bsz          0/1     Running   0          4s      10.130.0.10   compute2.cp4a.com
odm-lab-odm-decisioncenter-7f78864cb4-z6z6c          0/1     Running   0          4s      10.129.0.18   compute1.cp4a.com
odm-lab-odm-decisionrunner-b6bf759fd-h7p67           1/1     Running   0          11h     10.129.0.4
odm-lab-odm-decisionserverconsole-77c88fbb85-9pbw4   1/1     Running   0          11h     10.130.0.3    compute2.cp4a.com
odm-lab-odm-decisionserverruntime-78c44b4547-zsgrj   1/1     Running   0          11h     10.129.0.5    compute1.cp4a.com
[root@master deploy]#
```

__ q. Go back to the Application console and click the up arrow four more times to scale the the number of pods to 7.



__ r. Switch back to the terminal and run the command again to view the number of pods. This time notice the four new pods that are starting. Count the total number of pods that running on the compute1 node and the total number of pods that are running on the compute2 node. The number of pods running on the compute1 node is close to the number of pods running in the compute2 node. This way the Administrator is assured that the router is balancing the request between the compute nodes.

```
oc get pods -o wide
```

```
NAME                                                 READY   STATUS              RESTARTS   AGE     IP            NODE
NOMINATED NODE
odm-lab-odm-decisioncenter-7f78864cb4-5ttl2          0/1     Running             0          2s      10.129.2.20   compute1.cp4a.com
<none>
odm-lab-odm-decisioncenter-7f78864cb4-cwwwg          0/1     ContainerCreating   0          2s      <none>        compute1.cp4a.com
<none>
odm-lab-odm-decisioncenter-7f78864cb4-j9p7p          0/1     Running             0          2s      10.130.2.12   compute2.cp4a.com
<none>
odm-lab-odm-decisioncenter-7f78864cb4-kgjmb          1/1     Running             0          11h     10.130.0.4    compute2.cp4a.com
<none>
odm-lab-odm-decisioncenter-7f78864cb4-m5bsz          0/1     Running             0          2m      10.130.0.10   compute2.cp4a.com
<none>
odm-lab-odm-decisioncenter-7f78864cb4-vmlgr          0/1     ContainerCreating   0          2s      <none>        compute2.cp4a.com
<none>
odm-lab-odm-decisioncenter-7f78864cb4-z6z6c          0/1     Running             0          2m      10.129.0.18   compute1.cp4a.com
<none>
odm-lab-odm-decisionrunner-b6bf759fd-h7p67           1/1     Running             0          11h     10.129.0.4    compute1.cp4a.com
<none>
odm-lab-odm-decisionserverconsole-77c88fbb85-9pbw4   1/1     Running             0          11h     10.130.0.3    compute2.cp4a.com
<none>
odm-lab-odm-decisionserverruntime-78c44b4547-zsgrj   1/1     Running             0          11h     10.129.0.5    compute1.cp4a.com
```

__ s. If a pod crashes or is deleted, a new one starts to match the replica # set, which is 7 in this instance. In the terminal, select the name of one of the pods and run the command to delete it.

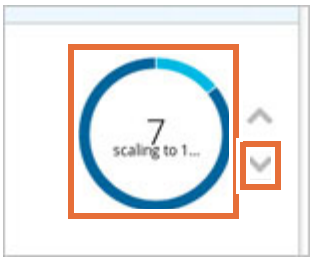
```
oc delete pod <a pod name>
```

```
[root@master deploy]# oc delete pod odm-lab-odm-decisioncenter-7f78864cb4-5q6fb
pod "odm-lab-odm-decisioncenter-7f78864cb4-5q6fb" deleted
```

__ t. One pod gets deleted but since the replica # is set to 7, a new pod is started right away to keep the number of running pods to 7 always.

```
t@master deploy]# oc get pods -o wide
                                                  READY   STATUS    RESTARTS   AGE     IP            NODE
MINATED NODE
lab-odm-decisioncenter-7f78864cb4-5qfpq           0/1     Running   0          40s     10.130.0.22   compute2.cp4a.
one>
lab-odm-decisioncenter-7f78864cb4-7vfwt           1/1     Running   0          18m     10.130.0.21   compute2.cp4a.
one>
lab-odm-decisioncenter-7f78864cb4-9b9nm           1/1     Running   0          27m     10.129.0.22   compute1.cp4a.
one>
lab-odm-decisioncenter-7f78864cb4-cqznd           1/1     Running   0          18m     10.129.0.24   compute1.cp4a.
one>
lab-odm-decisioncenter-7f78864cb4-w6f2n           1/1     Running   0          18m     10.129.0.23   compute1.cp4a.
one>
lab-odm-decisioncenter-7f78864cb4-wdcxk           1/1     Running   0          1h      10.130.0.14   compute2.cp4a.
one>
lab-odm-decisioncenter-7f78864cb4-zcgmw           1/1     Running   0          18m     10.130.0.20   compute2.cp4a.
one>
lab-odm-decisionrunner-b6bf759fd-n7mxr            1/1     Running   0          18h     10.129.0.12   compute1.cp4a.
one>
lab-odm-decisionserverconsole-77c88fbb85-xgx88    1/1     Running   0          18h     10.129.0.13   compute1.cp4a.
one>
lab-odm-decisionserverruntime-78c44b4547-dtmkq    1/1     Running   0          18h     10.129.0.14   compute1.cp4a.
one>
```

__ u. Go back to the web console and change the number of pods back to 1 as it was before. You do that by clicking the down arrow 6 times to scale the number of pods down to 1.



Part 4: Monitor applications with probes

OpenShift applications, including IBM Cloud Pak for Automation containerized applications, can become unhealthy due to issues such as temporary connectivity loss, configuration errors, or application errors. Developers can use probes to monitor their applications. A probe is a Kubernetes action that periodically performs diagnostics on a running container. Probes can be configured by using either the oc command-line client or the OpenShift web console. There are currently two types of probes that administrators can use:

Liveness Probe

A liveness probe determines whether an application that is running in a container is in a healthy state. If the liveness probe detects an unhealthy state, OpenShift kills the pod and tries to redeploy it again. Developers can set a liveness probe by configuring the **template.spec.containers.livenessprobe** paragraph of a pod configuration.

Readiness Probe

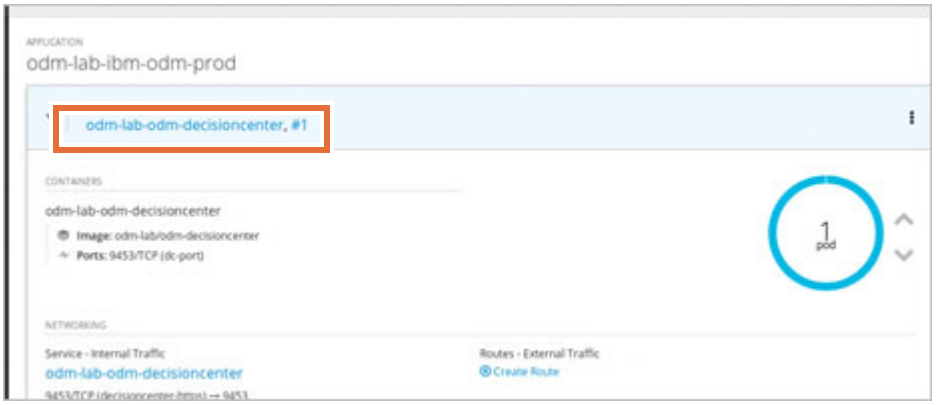
A readiness probe determines whether a container is ready to serve requests. If the readiness probe returns a failed state, OpenShift removes the container's IP address from the endpoints of all services. Developers can use readiness probes to signal to OpenShift that even though a container is running, it should not receive any traffic from a proxy. Developers can set a readiness probe by configuring the

__ 1. Check the events for the project.

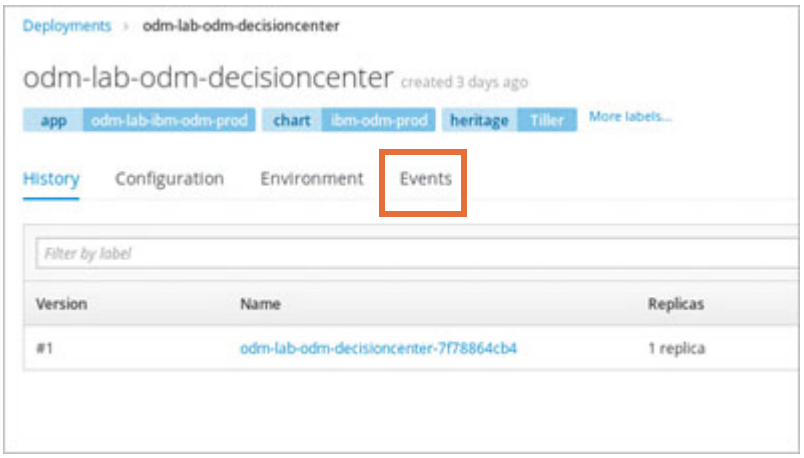
__ a. Make sure that you are in the odm-lab project in the Application console view of the web console.

__ b. In the web console, click the **Overview** tab.

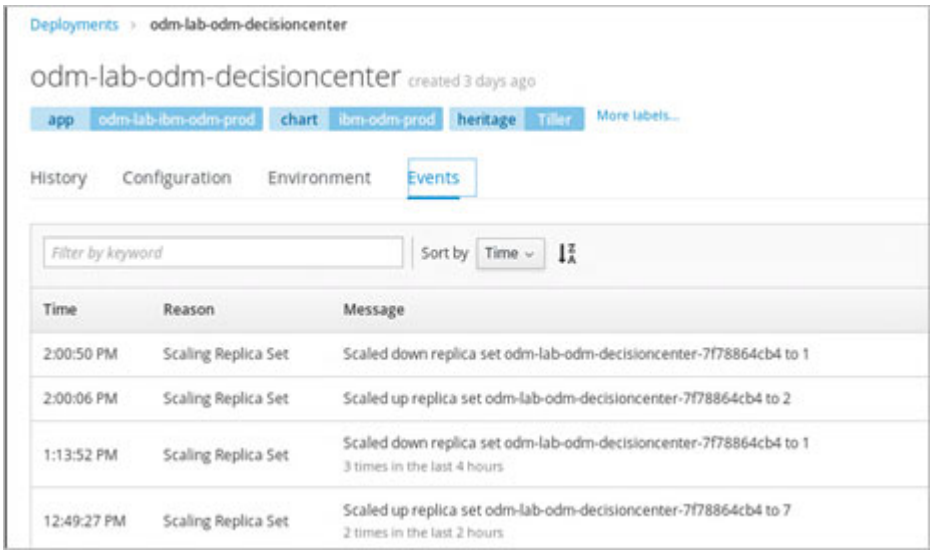
__ c. Click the Decision Center deployment.



__ d. Click the **Events** tab.



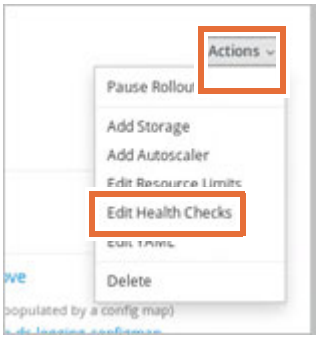
__ e. The latest generated events are listed.



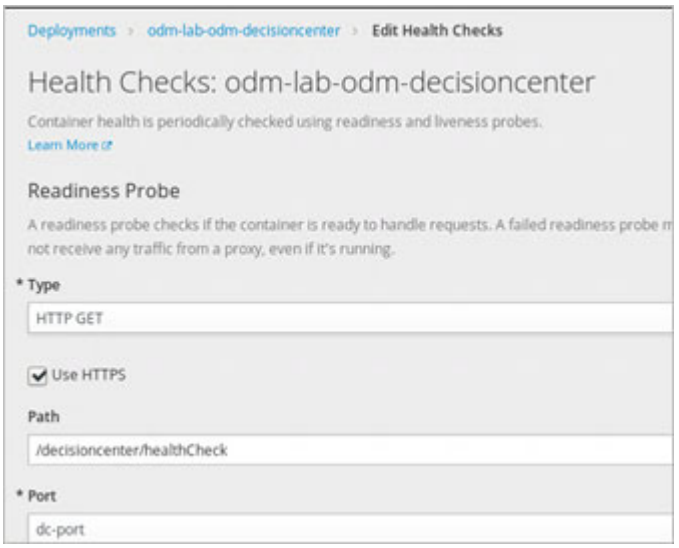
__ f. Click the **Configuration** tab. Scroll down to view the container details.

__ g. Examine the Readiness Probe and Liveness Probe settings for the container.

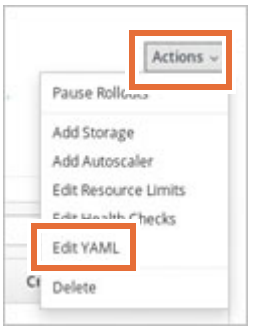
__ h. To change the settings of the Readiness Probe and Liveness Probe, you can click Actions and then click **Edit Health Checks**.



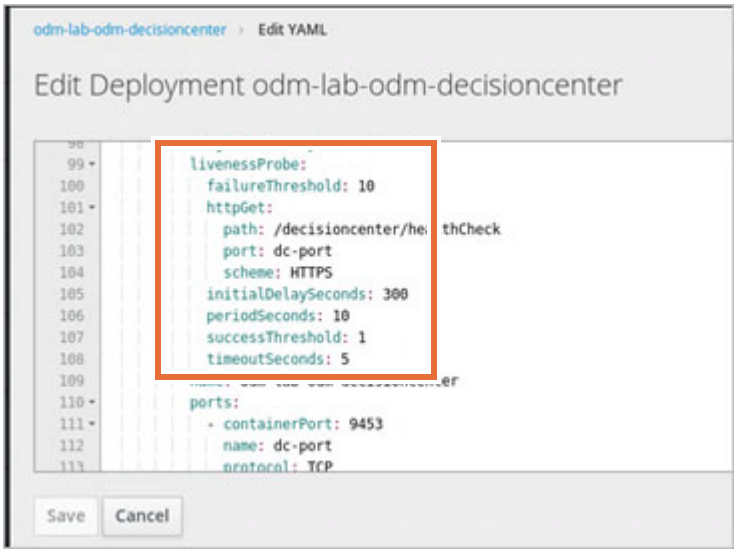
__ i. For each probe type, you can select the type, such as HTTP GET, TCP Socket, or Container Command, and specify the parameters for each type. You do not change any settings and click Cancel.



__ j. The web console can also be used to edit the YAML file that defines the deployment configuration. Click **Actions > Edit YAML**.



__ k. Upon the creation of a probe, a new entry is added to the configuration file for the deployment configuration. You can review or edit a probe by using the deployment configuration editor The live editor allows you to edit the periodSeconds, successThreshold, and failureThreshold and few other options. Scroll through the YAML file to examine the entries.



The **periodSeconds** value of 10 seconds specifies that the OpenShift performs a liveness probe every 10 seconds. The **initialDelaySeconds** value of 300 seconds tells OpenShift that it needs to wait 300 seconds before performing the first probe. To perform a probe, OpenShift sends an HTTP GET request to the server that is running in the Container and listening on port 8080. If the handler for the server's /health path returns a success code, OpenShift considers the Container to be alive and healthy. If the handler returns a failure code, the OpenShift kills the Container and restarts it.

Any code greater than or equal to 200 and less than 400 indicates success. Any other code indicates failure.

The **timeoutSeconds** value of 5 seconds is the time OpenShift waits for the probe to finish. If this time is exceeded, OpenShift Container Platform considers the probe to have failed.

The **SuccessThreshold** value of 1 is the minimum consecutive successes for the probe to be considered successful after having failed.

The **failureThreshold** value of 10 is the minimum consecutive failures for the probe to be considered failed after having succeeded.

Readiness and liveness probes can be used in parallel for the same container. Using both, an administrator, can configure the values to ensure that traffic does not reach a container that is not ready for it, and that containers are restarted when they fail

__ 2. Close all open windows.

__ a. Enter exit in the terminal to close it. If you had multiple terminal windows open close all of them

__ b. Close the Firefox browser.

__ c. Close any other open windows, including any text editors.

__ d. Close the RDP console window or the browser window that was used to work with the Master VM.

You now completed some basic administration tasks for managing containers in the OpenShift environment. This concludes this exercise.

End of exercise

Exercise review and wrap-up

The first part of the exercise ...

Appendix A. Installing Red Hat OpenShift Container Platform V3.1.1 to create the lab environment

Overview

This appendix describes how to prepare the Red Hat Enterprise Linux environment and install Red hat OpenShift Container Platform (RHOCP) V3.11.

Part 1: Configure Environment for installing Red Hat OpenShift Container Platform 3.1.1

__ 1. Login to the Master node as `root \ passw0rd`

__ 2. Verify that ports are open by running the `ss` command. Note that for this environment, ports have already been verified. As a test, you can check port 9091 that is needed by Calico. Replace the `<port_numbers>` variable with 9091. If the port is not in use, the output is empty. If the port is in use, an output is displayed. In this example, since the port is not in use, the output is empty.

```
ss -tnlp | awk '{print $4}'| egrep -w "<port_numbers>"
```

__ 3. Make sure the Python 2.6+ (2.6 – 2.9.x are supported) is installed on each node in the cluster. To verify the Python version run the following command.

```
python -V
```

Note: Python V3.x is not supported yet.

__ 4. Make sure unzip is installed.

```
unzip -v
```

Unzip version should be at 6.0 or higher.

__ 5. Make sure proper version of Firefox is installed.

Firefox must be the most recent version.

```
firefox -v
```

Version should be at 60.2.2 or higher.

__ 6. Verify the GNU Compiler Collection is installed.

```
rpm -q gcc -v
```

GCC should be at version 4.8 or higher and is used to configure Socat in a later step.

__ 7. Verify NTP is enabled.

```
timedatectl
```

__ 8. It is a good idea to synchronize the clocks in each node in the cluster. To synchronize your clocks, you can use network time protocol (NTP). Verify that the response includes “NTP enabled:: yes”

__ 9. Increase the number of mmap counts (**Node: Master and Compute nodes**)

The default operating system limits on mmap counts is likely to be too low, which may result in out of memory exceptions. Increase the number of mmap counts on each node in the cluster.

__ a. Update kernel parameters.

```
echo "kernel.sem=250 32000 32 1024" >> /etc/sysctl.conf
echo "vm.max_map_count=262144" >> /etc/sysctl.conf
```

__ b. Restart the server for changes to take effect

```
shutdown -r now
```

__ 10. Configure Password-less SSH (**Node: Master**)

Configure password-less SSH from the Master node to all other nodes in the cluster.

__ a. Create keys for Master node

```
ssh-keygen -b 4096 -f ~/.ssh/id_rsa -N ""
cat ~/.ssh/id_rsa.pub | sudo tee -a ~/.ssh/authorized_keys
```

__ b. Copy the public key to all the nodes in the cluster.

```
ssh-copy-id -i ~/.ssh/id_rsa.pub root@<WORKER_NODE_IP>
```

If you get a warning that all keys were skipped, that is expected behavior. Answer “yes” if asked to connect.

When prompted to continue to connect, enter `yes`.

When prompted to enter a password, enter `passw0rd`.

__ c. Verify that you can now use SSH from the master node to other nodes ***without supplying a password***.

```
ssh root@<WORKER_NODE_IP>
```

__ d. Once connected to the worker node, restart the sshd on each worker node.

```
systemctl restart sshd
```

Important

If you receive an error message stating “Signing failed, agent refused operation” verify the IP you are using is correct.

__ e. Enter `exit` to exit SSH session to the worker node

__ 11. Configure Secure Shell (SSH) for root user. (**Node: Master and Compute nodes**)

__ a. Before editing the host file on the Master node (next step), get the host name of each node by using the `hostname` command. Make a note of the Master and Worker host names.

```
hostname
```

__ b. Enable remote login

```
sed -i 's/prohibit-password/yes/' /etc/ssh/sshd_config
sed -i 's/PermitRootLogin no/PermitRootLogin yes/'  /etc/ssh/sshd_config
```

__ c. Restart ssh

```
systemctl restart sshd
```

__ 12. Configure `/etc/hosts` file (**Node: Master**)

Edit the `/etc/hosts` file on the Master node and add IP addresses and host names for each node in the cluster.

__ a. Update the `/etc/hosts` file on the Master node

```
127.0.0.1    localhost
<MASTER_NODE_IP>   <MASTER_NODE_HOSTNAME> #master
<compute1_NODE_IP>   <WORKER_NODE_HOSTNAME> #compute1
<compute2_NODE_IP>   <WORKER_NODE_HOSTNAME> #compute2
```

__ b. Copy `/etc/hosts` from Master node to all cluster servers

```
scp /etc/hosts root@<compute_NODE_IP>:/etc/hosts
```

The `/etc/hosts` file must be identical on all machines in the cluster. Answer “yes” if asked to connect.

Important

Enable Red Hat subscription on Master and Worker nodes. Refer to subscription document for details. It is important to complete the steps in the Subscription document before continuing or else commands fail. The exception is if you are running these steps in your own environment (not this sandbox), in which case, check with your RHEL Administrator.

__ 13. Stop and disable Firewall (**Node: Master and Compute nodes**)

```
service firewalld stop
systemctl disable firewalld
```

If you restart any of the VMs then you must stop and disable firewall again using these commands.

__ 14. Start and enable the NFS and RPC services (**Node: Master**)

__ a. Enable NFS and RPC services:

```
systemctl enable nfs-server
systemctl enable rpcbind
systemctl enable nfs-lock
systemctl enable nfs-idmap
```

__ b. Start NFS and RPC services:

```
systemctl start nfs-server
systemctl start rpcbind
systemctl start nfs-lock
systemctl start nfs-idmap
```

Part 2: Install Red Hat OpenShift

1. Update the host file on Master to add an entry for the cluster console link to work in the OpenShift web console. Cluster console view is a view that opens when you are logged into the OpenShift web console. This link does not work if the host does not have the following entry in this environment:

```
10.0.0.1 console.cp4a.com
```

2. Update the host file to add the entries for the metrics to work in the cluster console.

```
grafana-openshift-monitoring.cp4a.com
prometheus-k8s-openshift-monitoring.cp4a.com
alertmanager-main-openshift-monitoring.cp4a.com
```

__ 3. The host file looks like the following:

```
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
10.0.0.1    master.cp4a.com     console console.cp4a.com grafana-openshift-monitoring.cp4a.com
prometheus-k8s-openshift-monitoring.cp4a.com alertmanager-main-openshift-monitoring.cp4a.com
10.0.0.2    compute1.cp4a.com   compute1
10.0.0.3    compute2.cp4a.com   compute2
```

__ 4. Download the inventory file from Red Hat and customize it for the installation as needed.

__ 5. Check the prerequisites by running the command below.

```
ansible-playbook -vv playbooks/prerequisites.yml
```

__ 6. Run the following command to install OpenShift cluster.

```
ansible-playbook -vv
/usr/share/ansible/openshift-ansible/playbooks/deploy_cluster.ym
l
```

__ 7. Set the password for user admin.

```
htpasswd -cb /etc/origin/master/htpasswd admin passw0rd
```

__ 8. Add the cluster-admin role to user admin.

```
oc adm policy add-cluster-role-to-user cluster-admin admin
```

It may say that user admin could not be found. That won't have any impact, just ignore the message.

__ 9. Open `https://master.cp4a.com:8443` in Firefox browser. Verify you can login with admin/passw0rd. Bookmark the page for future use

__ 10. Verify that you can click the Cluster Console link at the top. This confirms that you can view the cluster console and do cluster admin administration and monitoring using the console. You can also go to the cluster console directly by using the link:
`https://console.cp4a.com:8443`

__ 11. Download tiller.yaml from shared box folder. Create a service account for your tiller pod and assign it to cluster-admin role.

```
kubectl apply -f /root/tiller.yaml


helm init --service-account tiller --upgrade


oc adm policy add-cluster-role-to-user cluster-admin
system:serviceaccount:kube-system:tiller
```

__ 12. Extract helm package.

```
tar zxvf *.gz
```

__ 13. Copy helm executable file to /usr/bin.

```
cp linux-amd64/helm /usr/bin
```

__ 14. Initialize helm locally.

```
helm init --client-only --skip-refresh
```

__ 15. Copy tiller docker image to both compute nodes.

```
scp tiller.tar root@compute1:/root
scp tiller.tar root@compute2:/root
```

__ 16. Use SSH to connect to both compute nodes, load the docker image.

```
ssh compute1
docker load -i tiller.tar
exit
ssh compute2
docker load -i tiller.tar
exit
```

__ 17. Make sure you quit SSH and go back to VM 1. Install tiller.

```
helm init
```

__ 18. Create a service account for your tiller pod and assign it cluster-admin role.

```
kubectl apply -f /root/tiller.yaml
helm init --service-account tiller --upgrade
oc adm policy add-cluster-role-to-user cluster-admin
system:serviceaccount:kube-system:tiller
```

Sample Inventory file:

Create an OSEv3 group that contains the masters, nodes, and etcd groups

[OSEv3:children]

masters

nodes

etcd

Set variables common for all OSEv3 hosts

[OSEv3:vars]

SSH user, this user should allow ssh based auth without requiring a password

ansible_ssh_user=root

openshift_deployment_type=openshift-enterprise

openshift_image_tag=v3.11.117

#openshift_pkg_version=-3.11.117

If ansible_ssh_user is not root, ansible_become must be set to true

#ansible_become=true

default selectors for router and registry services

openshift_router_selector='node-role.kubernetes.io/infra=true'

openshift_registry_selector='node-role.kubernetes.io/infra=true'

uncomment the following to enable htpasswd authentication; defaults to DenyAllPasswordIdentityProvider

openshift_master_identity_providers=[{'name': 'htpasswd_auth', 'login': 'true', 'challenge': 'true', 'kind': 'HTPasswdPasswordIdentityProvider'}]

openshift_master_default_subdomain=cp4a.com

openshift_disable_check=memory_availability,disk_availability,docker_storage,docker_image_availability

```
#os_sdn_network_plugin_name=redhat/openshift-ovs-multitenant

openshift_master_cluster_method=native
openshift_master_cluster_hostname=console.cp4a.com
openshift_master_cluster_public_hostname=console.cp4a.com

ansible_service_broker_install=false
openshift_enable_service_catalog=false
template_service_broker_install=false
openshift_logging_install_logging=false

# registry passwd
oreg_url=registry.redhat.io/openshift3/ose-${component}:${version}
oreg_auth_user=XXXXX
oreg_auth_password=XXXX
# openshift_examples_modify_imagestreams=true

# docker config
openshift_docker_additional_registries=registry.redhat.io
#openshift_docker_insecure_registries=rh16.cn.ibm.com:5000
#openshift_docker_blocked_registries
openshift_docker_options="--log-driver json-file --log-opt max-size=1M --log-opt max-file=3"

openshift_cluster_monitoring_operator_install=true
openshift_metrics_install_metrics=true
openshift_enable_unsupported_configurations=True
openshift_logging_es_nodeselector='node-role.kubernetes.io/infra: "true"'
openshift_logging_kibana_nodeselector='node-role.kubernetes.io/infra: "true"'

# host group for masters
[masters]
master.cp4a.com

# host group for etcd
```

```
[etcd]
master.cp4a.com

# host group for nodes, includes region info
[nodes]
master.cp4a.com openshift_node_group_name='node-config-master-infra'
compute1.cp4a.com openshift_node_group_name='node-config-compute'
compute2.cp4a.com openshift_node_group_name='node-config-compute'

#openshift_console_install=true
#openshift_console_hostname=console.cp4a.com
#openshift_master_api_port=8443
#openshift_master_console_port=8443
```

IBM Training