Lab 02 Explore RedHat OpenShift Container Platform

Contents

LAB 02 EXPLORE REDHAT OPENSHIFT CONTAINER PLATFORM 6
2.1 INTRODUCTION 6
2.2 APPLICATION MODERNIZATION WITH IBM CLOUD PAK FOR APPLICATIONS – PROOF OF TECHNOLOGY 6
2.3 WHAT IS RED HAT OPEN SHIFT CONTAINER PLATFORM 7
2.4 AUDIENCE FOR THIS PROOF OF TECHNOLOGY 7
2.5 LAB ENVIRONMENT 8
2.6 USER ID AND PASSWORDS 8
2.7 INSTALLATION OF IBM CLOUD PAK FOR APPLICATIONS 8
2.8 LINUX TIPS 8
2.9 REVIEW SCRIPT BEFORE RUNNING 9
2.10 LET'S GET STARTED 10
APPENDIX: SKYTAP TIPS FOR LABS 23
2.1 HOW TO USE COPY / PASTE BETWEEN LOCAL DESKTOP AND SKYTAP VM? 23

2.1 Introduction

This is "**Lab 02 -Explore RedHat OpenShift Container Platform**" from an IBM Cloud Pak for Applications & App Modernization Proof of technology (PoT). The labs are not required to be executed in order. And, you may skip labs, and only perform the labs that suit your desired learning objectives.

<u>The full set of labs in the PoT are:</u>

Lab01 - Getting started with Docker

Lab02 - Explore RedHat OpenShift Container Platform

Lab03 - Getting started with Kubernetes

Lab04 – Liberty application deployment using Operators

Lab05 – IBM Cloud Pak for Applications - App Modernization using Transformation Advisor

Lab06 – App Modernization with Java EE Microservices and Liberty

Lab07 – Using Tekton pipelines for CI/CD of microservices to RedHat OpenShift Container Platform

2.2 Application Modernization with IBM Cloud Pak for Applications – Proof of Technology

The goal of this lab is to provide a quick introduction to Red Hat OpenShift Container Platform (RHOCP) which is the Kubernetes implementation used in IBM Cloud Pak for Applications (CP4Apps). Let's start by learning some basic concepts.

2.3 What is Red Hat Open Shift Container Platform

The Red Hat OpenShift Container Platform offers full access to an enterprise ready Kubernetes. RHOCP includes a Kubernetes distribution that has undergone an extensive compatibility test matrix with many of the software elements you will find in your datacenter.

As with the rest of the Red Hat software portfolio RHOCP includes service level agreements for support, bug fixes, and CVE protection. RHOCP provides default security context constraints, pod security policies, best practice network and storage settings, service account configuration, SELinux integration, HAproxy edge routing configuration, and other out of the box protections needed for an enterprise deployment.

Redhat OpenShift Container Platform (RHOCP) offers an integrated monitoring solution, based on Prometheus, that offer deep coverage and alerting of common Kubernetes' issues.

2.4 Audience for this Proof of Technology

This Proof of Technology is mainly designed for database administrators (DBAs), Linux system administrators, application architects, solution designers and infrastructure professionals. IBM Cloud Pak for Applications is a conglomeration of many different technologies and not every aspect can be covered in a single Proof of Technology.

In this session, we cover certain aspects that will be useful to a specific role. For example:

Role	Capabilities
Database administrators	Learn a new paradigm of database software deployment and database management.
Application architects	Deploy application servers using a push-button approach and evaluate existing applications suitability to deploy in new paradigm of using cloud capabilities.
Solution designers	Learn new greenfield architecture of microservices and implement those on your platform.
Infrastructure professionals	Learn about Kubernetes and Docker containers orchestration using IBM Cloud Pak for Apps on OpenShift Container Platform in your own environment.

2.5 Lab environment

We are using a single virtual machine (VM) to demonstrate IBM Cloud Pak for Apps. the lab environment was built to the following specifications:

- The operating system is Red Hat Enterprise Linux Server 7.7
- The kernel version is: Linux 3.10.0-1062.4.1.el7.x86_64 x86_64
- The docker version installed is: Docker version 1.13.1, build 4ef4b30/1.13.1
- The Red Hat OpenShift Container Platform version is: RHOCP 3.11.153
- The recommended host memory to run the VM is 20 GB RAM
- The minimum SSD space required to build and run the labs is 100 GB of free space

2.6 User ID and passwords

The logon credentials for the primary Operating System user is:

Username: ibmdemo

Password: `passw0rd` (with a zero), which is also the `sudo` password

.

2.7 Installation of IBM Cloud Pak for Applications

IBM Cloud Pak for Applications, which pre-reqs Red Hat OCP and Docker, is already installed on an IBM-provided laptop.

2.8 Linux tips

If you are new to the Linux environment, the following tips should help.

- As opposed to the Microsoft Windows, you must click in a Linux GUI VM window to get it to focus, so you can type the commands.
- You can type the `clear` command from the command line window to clear the contents.
- When you are typing in a command shell, after few characters, click the **tab** key to autocomplete instead of typing the whole command. This will save some typing.
- To close a command line window, you can either type `exit` followed by `Enter` or press `CTRL-D`.
- The command that you need to type are bold faced such as:
  ```
  $ ls -l
  ```
- If the command shown in this lab documents starts with **`$`**, please assume that you are running that command as `ibmdemo` user.
- You should not need to run as root. The **ibmdemo** user should have sufficient rights. However, you may need to prefix the command with sudo to execute as a privileged (root) user. For example: `sudo kubectl (command goes here).`
- It is assumed that you will click `Enter` after typing the commands

2.9 Review script before running

__1. While doing the lab exercises, we ask you to type the name of the script to run. We recommend that you view the contents of the script before running it.

__2. You can use a method of your choice to view the contents of the script, such as one of the following commands.
`$ more <script name>` → See the contents one screen at a time
`$ cat <script name>` → Equivalent of `Windows` type command.

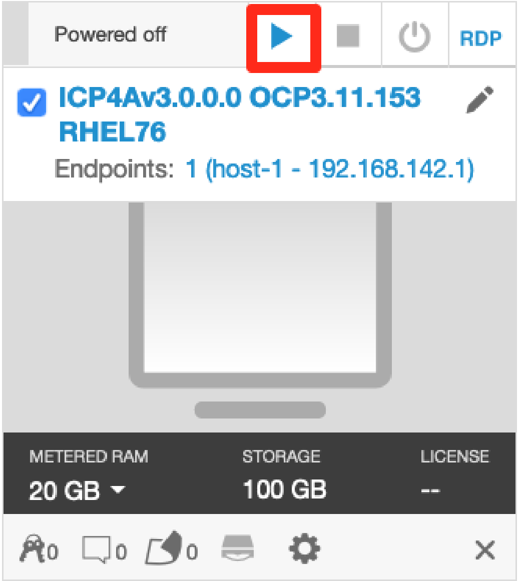
__3. The lab exercises do not have explicit instructions requesting you to view the contents of the script. It is good practice to view the script before you run it.

__4. If you like the scripts and want to use or customize them for your use, please feel free to ask the instructor to get them for you.

2.10 Let’s get started

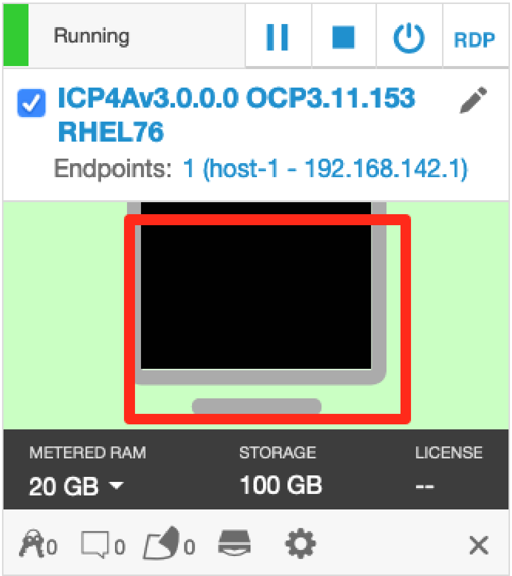
On your laptop/workstation, locate the `ICP4Av3.0.0.0 OCP3.11.153 RHEL76` virtual machine

__1. The VM should already be running. If not, Launch the Lab environment by clicking the **Run this VM** icon.



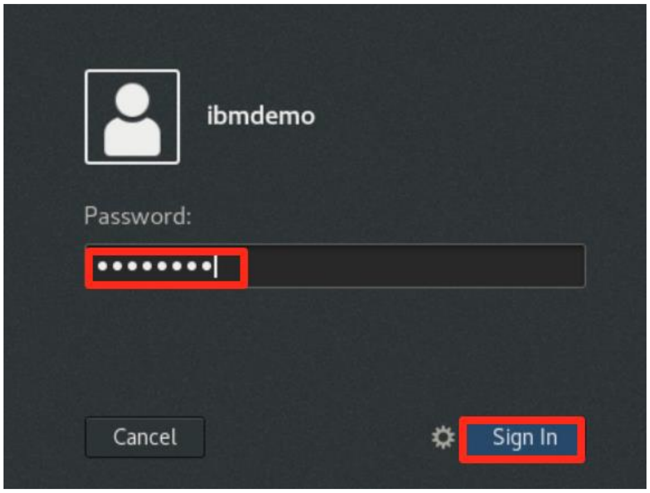
Powered off
RDP
ICP4Av3.0.0.0 OCP3.11.153 RHEL76
Endpoints: 1 (host-1 - 192.168.142.1)
METERED RAM
20 GB
STORAGE
100 GB
LICENSE
--
0
0
0

__2. After the VM is running, click its icon to access the VM’s desktop.

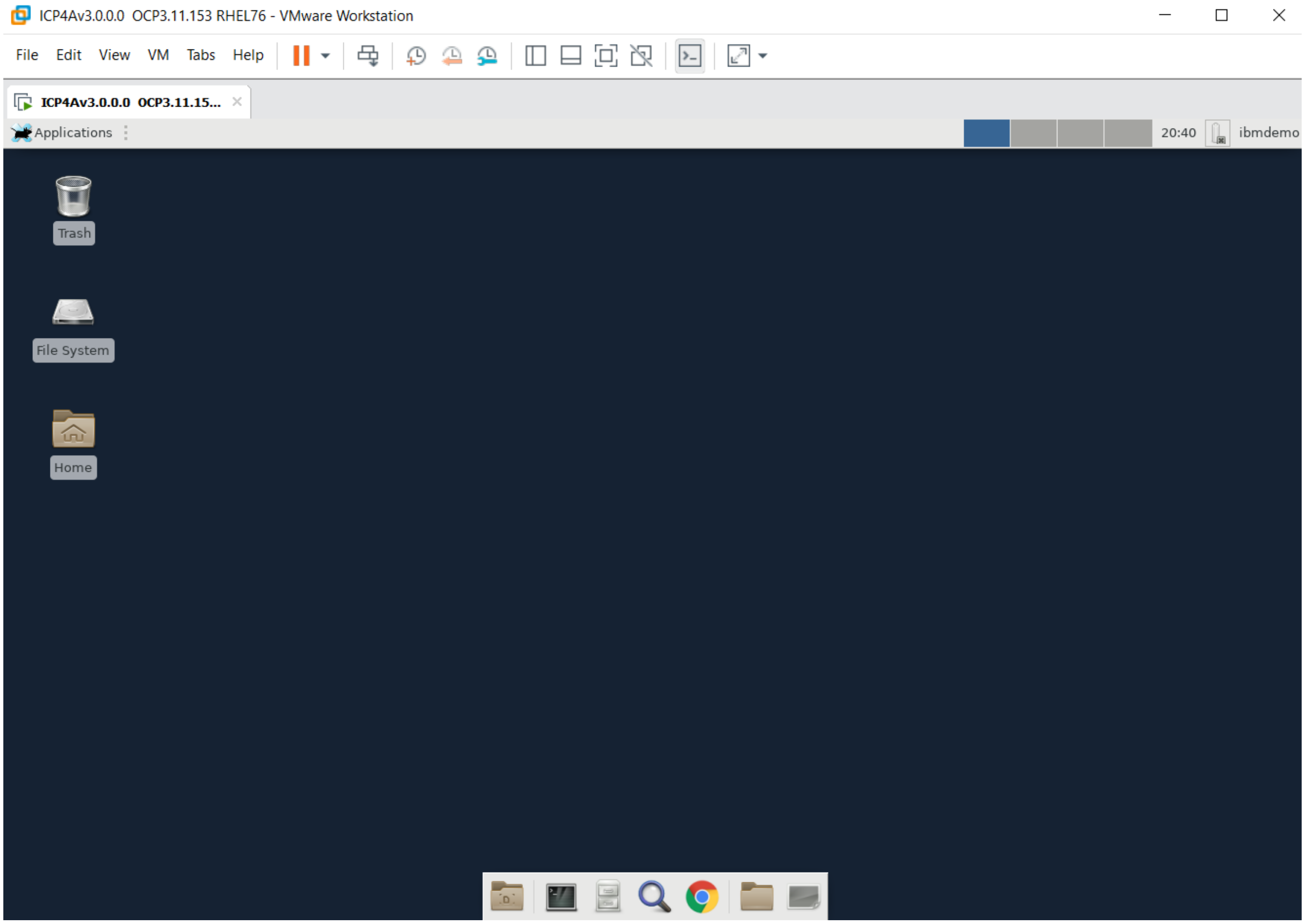


Running
RDP
ICP4Av3.0.0.0 OCP3.11.153 RHEL76
Endpoints: 1 (host-1 - 192.168.142.1)
METERED RAM
20 GB
STORAGE
100 GB
LICENSE
--
0
0
0

__3. After the VM machine powers on, log with the `ibmdemo` user using the password `passw0rd`

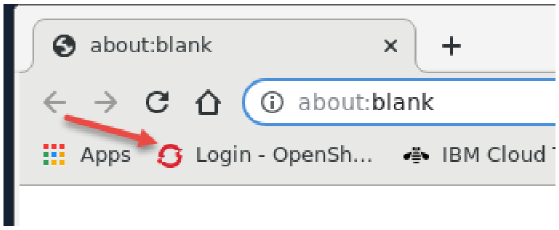


The `ICP4Av3.0.0.0 OCP3.11.153 RHEL76` virtual machine running and its Desktop is displayed in a web browser window.


ICP4Av3.0.0.0 OCP3.11.153 RHEL76 - VMware Workstation
File
Edit
View
VM
Tabs
Help
ICP4Av3.0.0.0 OCP3.11.15...
Applications
20:40
ibmdemo
Trash
File System
Home

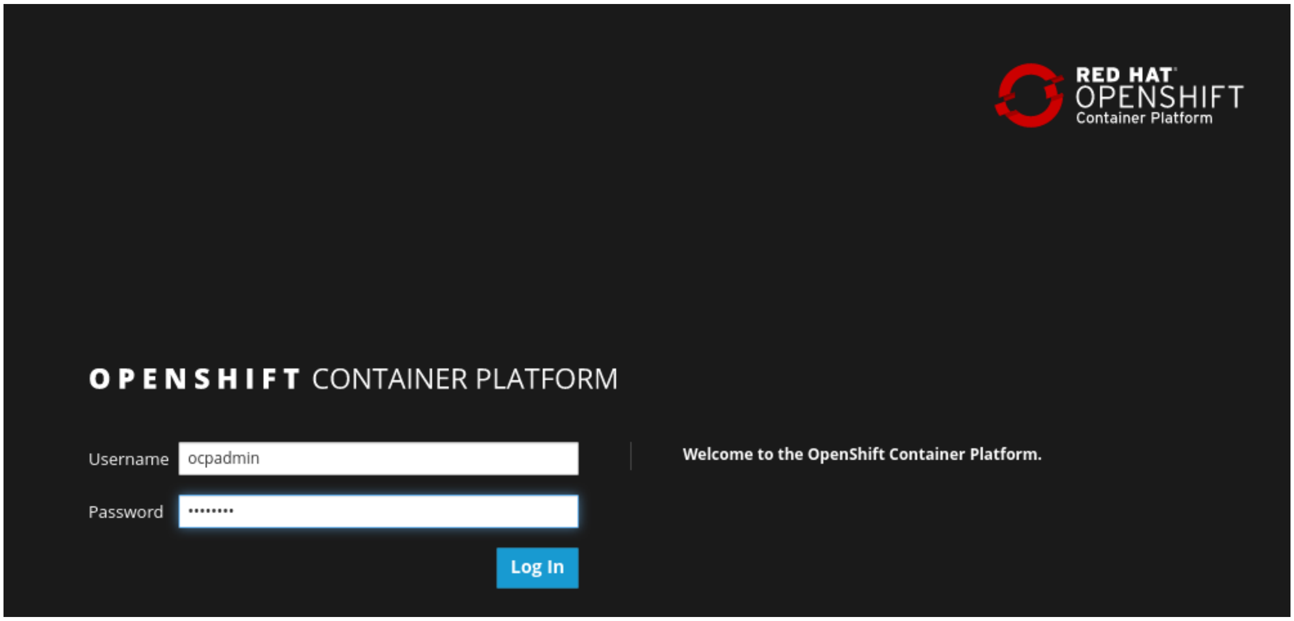
Note: Refer to the **Appendix** in this lab guide for details for using Copy / Paste between the lab guide and the lab environment.

__4. Launch the `Chrome` browser and click on the `Login - OpenShift Container Platform` bookmark



__5. Scroll down to the `username` and `password` fields and type `ocpadmin` in the username and `ocpadm1n` (note the "1") in the password and click `Log in`

Note: ocpadmin is a RHOCP cluster admin, this is the equivalent of the Unix/Linux root user. Use extreme care in this PoT to avoid making unwanted and potentially unrecoverable changes to the RHOCP cluster



OPENSHIFT CONTAINER PLATFORM

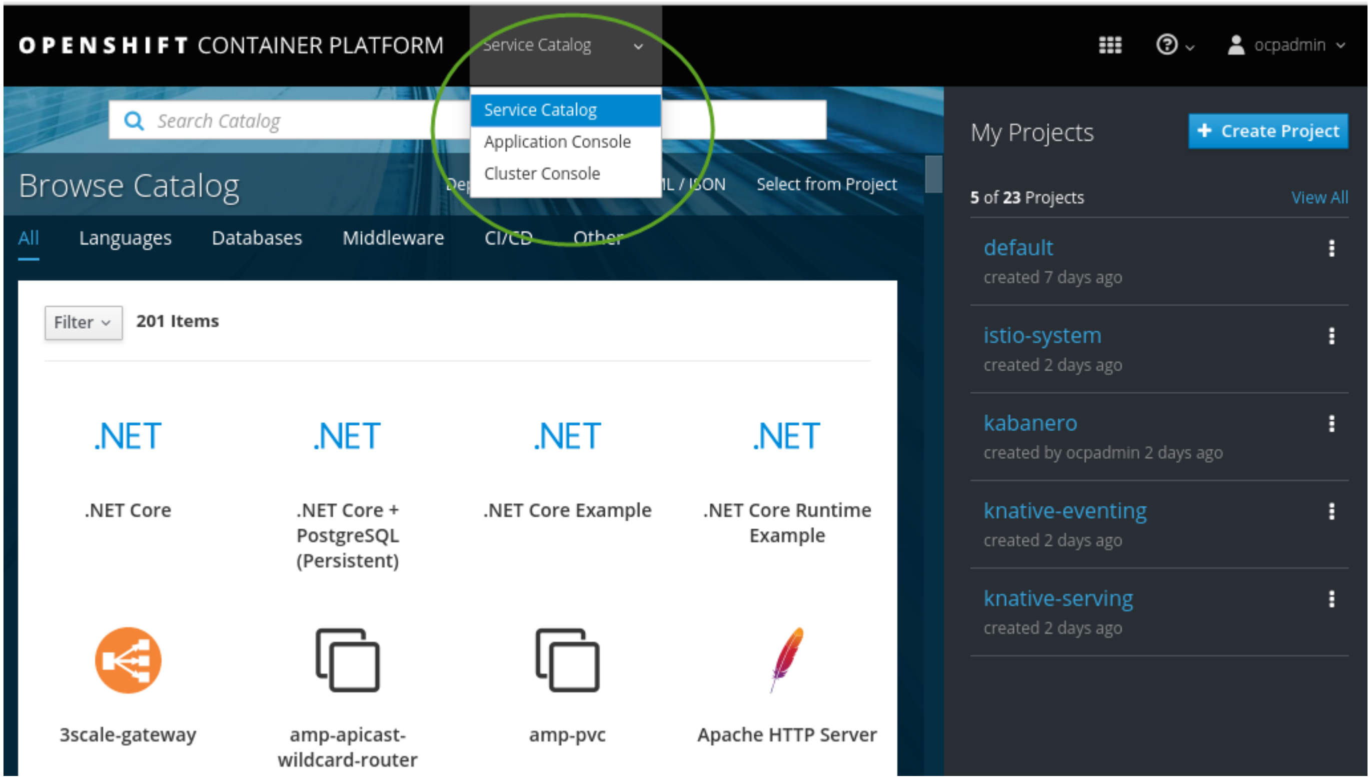
Username: ocpadmin

Password: ••••••••

Welcome to the OpenShift Container Platform.

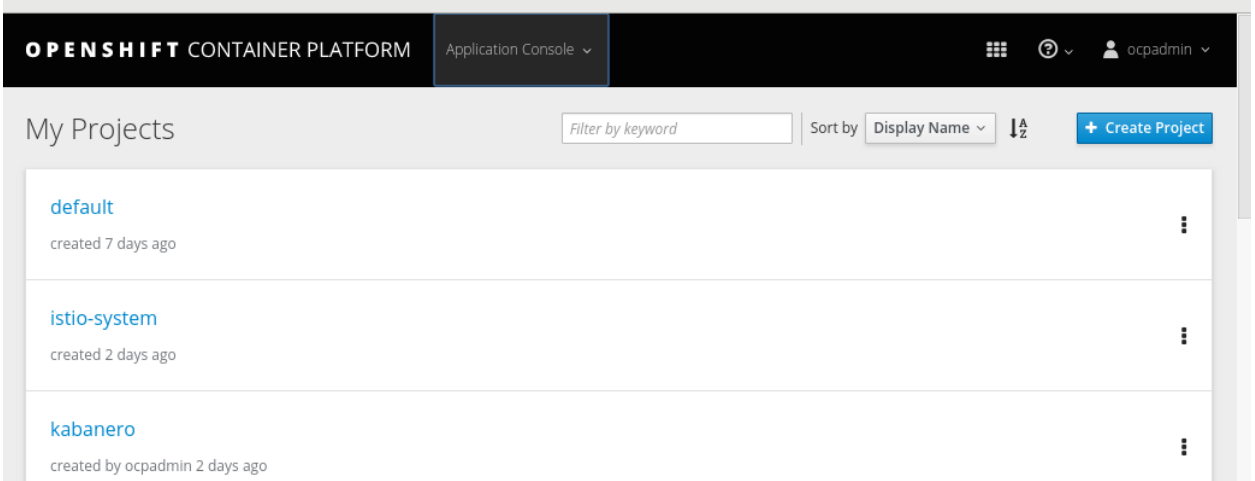
Note: If you receive a "Warning: Potential Security Risk Ahead" page, this is because OCP uses a self-signed certificate. Click `Advanced`, then scroll down and click `Accept the Risk and Continue` which will import the certificate into the browser

__6. The default view is the `Service Catalog.` There's also an `Application Console` and a `Cluster Console.`

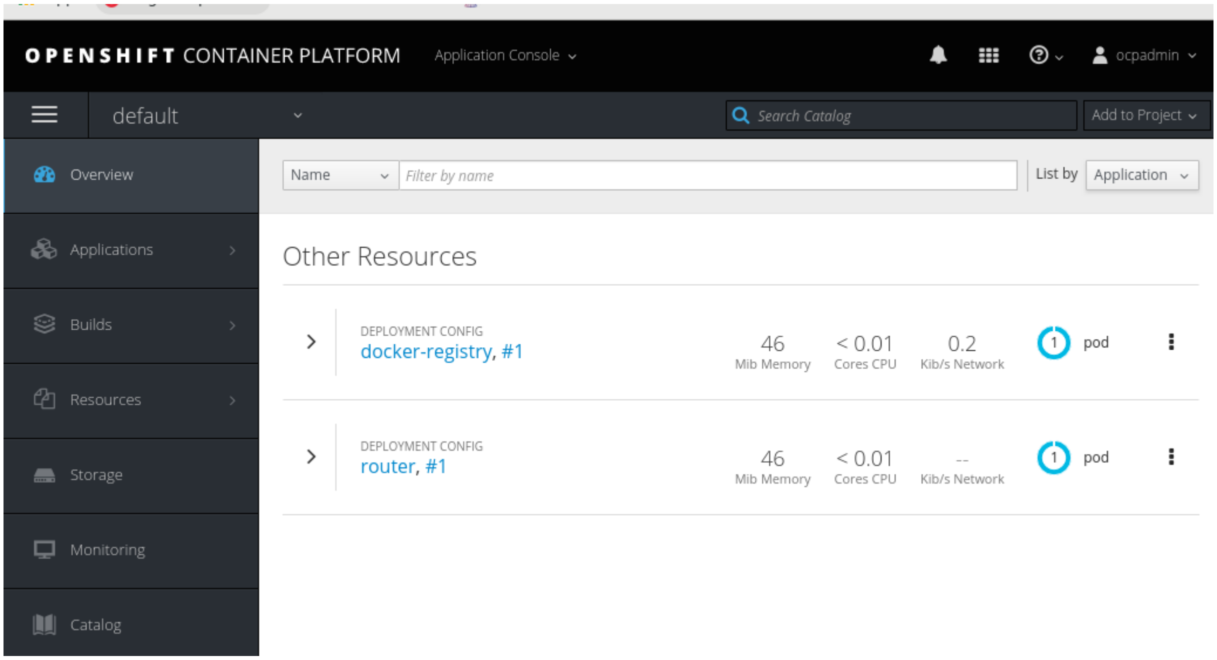


The OpenShift Container Platform includes a *service catalog*, an implementation of the Open Service Broker API (OSB API) for Kubernetes. This allows users to connect any of their applications deployed in OpenShift Container Platform to a wide variety of service brokers.

__7. Click on `Application Console`



__8. The Application Console organizes deployments by Project. A Project extends a Kubernetes namespace by providing additional annotations and provides an easy way to observe and change all the resources such as applications, storage, etc associated with a Project. As a cluster administrator the user ocpadmin has access to all projects and all resources



Other Resources

DEPLOYMENT CONFIG
docker-registry, #1
46 Mib Memory
< 0.01 Cores CPU
0.2 Kib/s Network
1 pod

DEPLOYMENT CONFIG
router, #1
46 Mib Memory
< 0.01 Cores CPU
-- Kib/s Network
1 pod

Kubernetes namespaces provide:

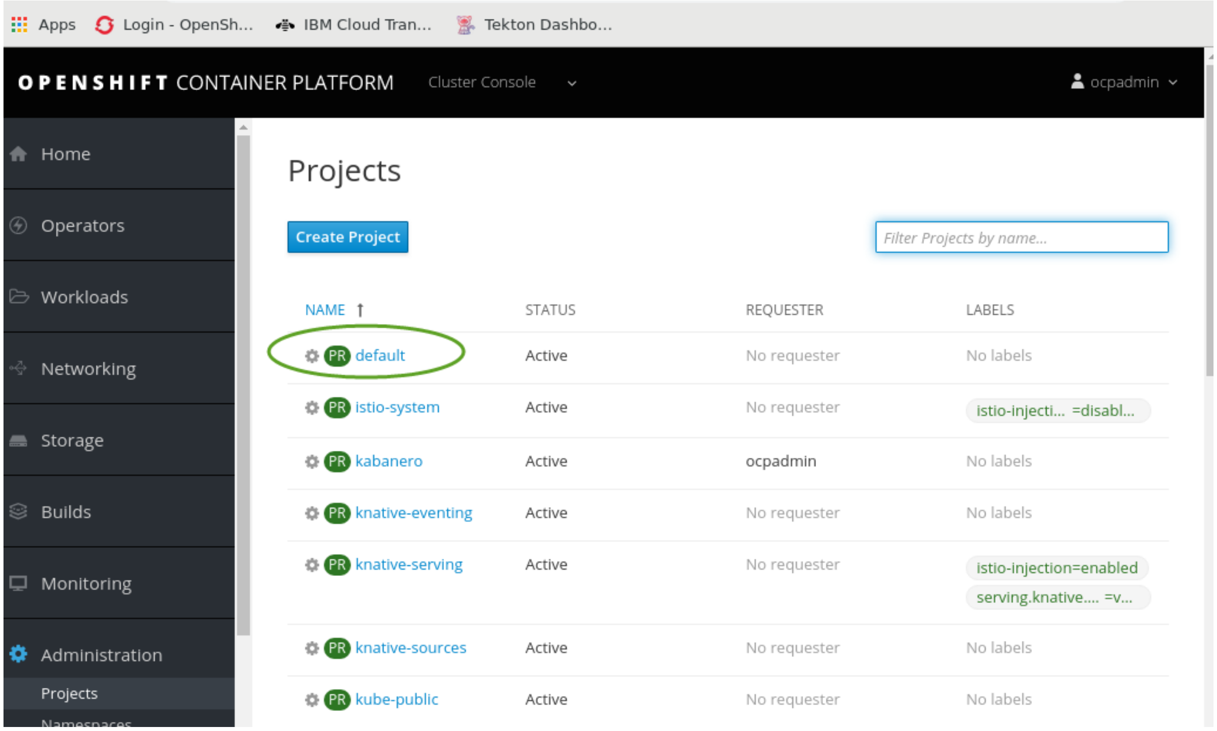
__a. A unique scope to named resources to avoid naming collisions

__b. Delegated management for trusted users

__c. The ability to limit resource consumption

Most objects in the system are scoped by namespace, but some are excepted and have no namespace, including nodes and users.

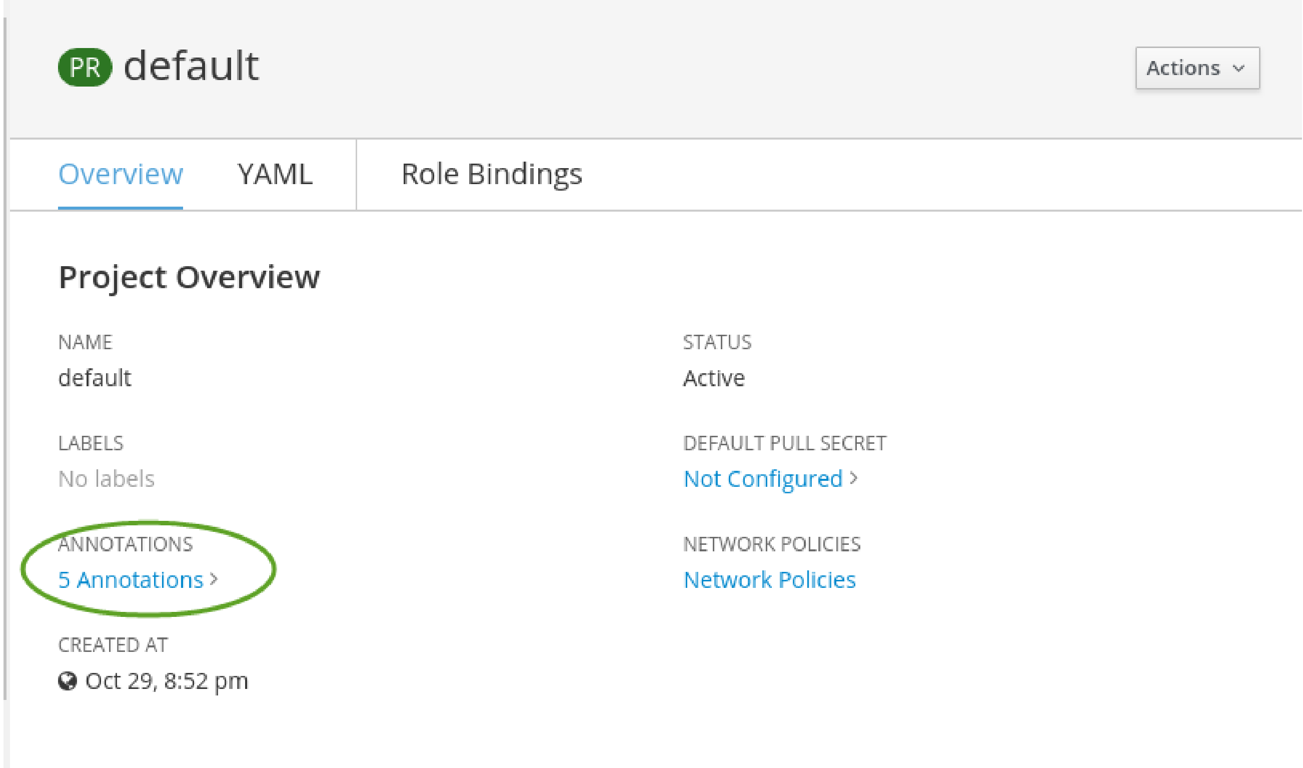
__9. Click on `Cluster Console` which provides a view of all artifacts in the cluster, including sensitive information such as `Role Bindings` and resources by project or the entire cluster.



Projects

NAME	STATUS	REQUESTER	LABELS
PR default	Active	No requester	No labels
PR istio-system	Active	No requester	istio-injecti… =disabl…
PR kabanero	Active	ocpadmin	No labels
PR knative-eventing	Active	No requester	No labels
PR knative-serving	Active	No requester	istio-injection=enabled serving.knative.… =v…
PR knative-sources	Active	No requester	No labels
PR kube-public	Active	No requester	No labels

__10. Click on the `default` project, then click on `Annotations`



Project Overview

NAME
default

STATUS
Active

LABELS
No labels

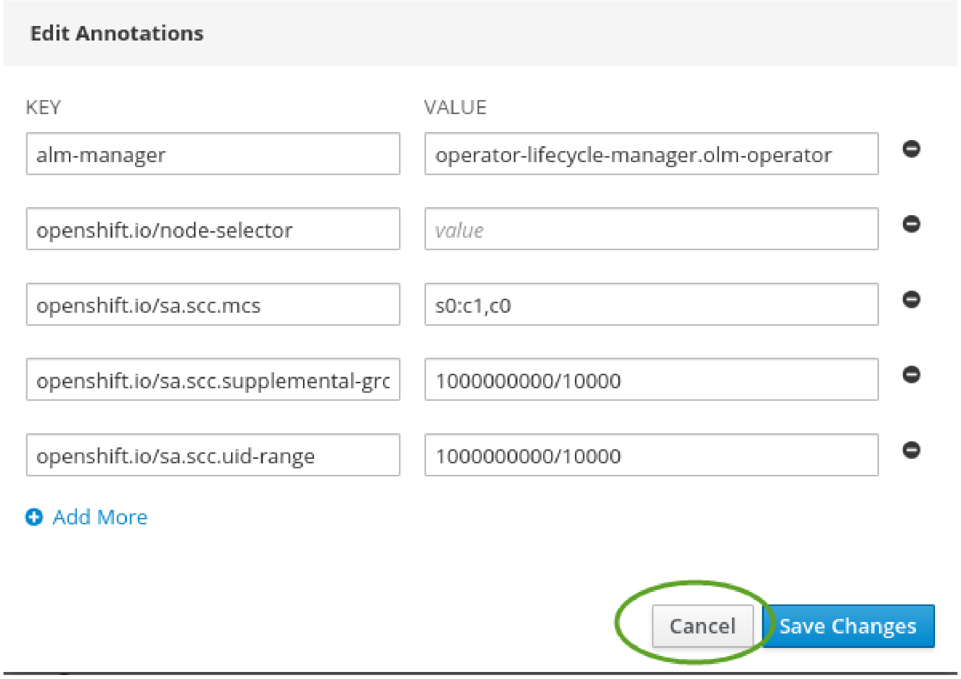
DEFAULT PULL SECRET
Not Configured

ANNOTATIONS
5 Annotations

NETWORK POLICIES
Network Policies

CREATED AT
Oct 29, 8:52 pm

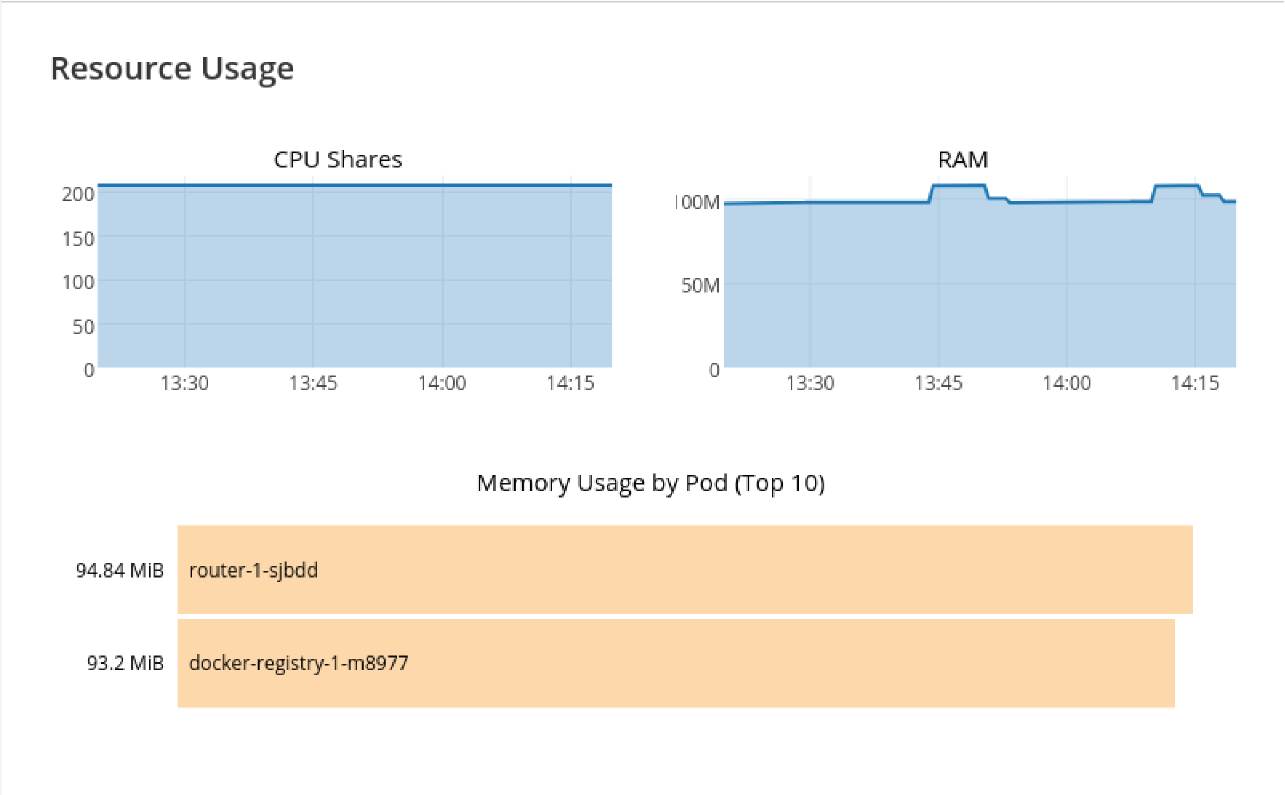
__11. As previously stated, an OCP `Project` extends a Kubernetes `namespace` with additional annotations and associates Kubernetes artifacts to the Project. Click `cancel` to exit this view,



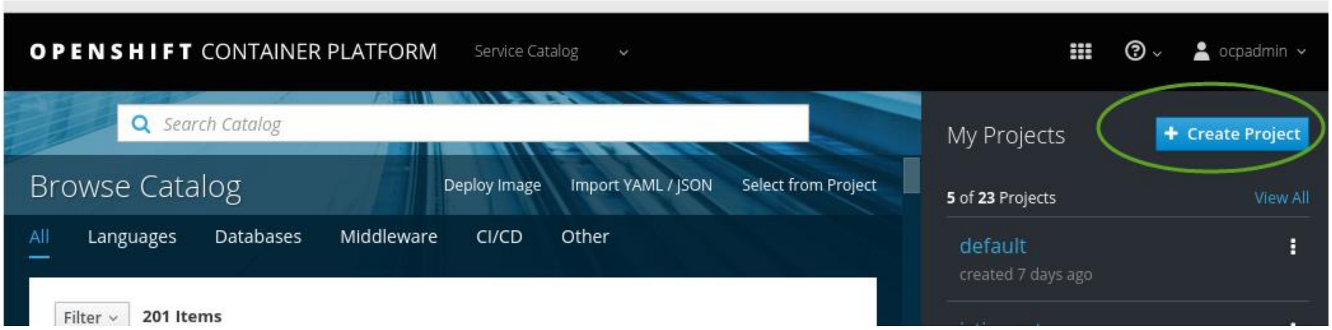
Edit Annotations

KEY	VALUE
alm-manager	operator-lifecycle-manager.olm-operator
openshift.io/node-selector	*value*
openshift.io/sa.scc.mcs	s0:c1,c0
openshift.io/sa.scc.supplemental-grc	1000000000/10000
openshift.io/sa.scc.uid-range	1000000000/10000

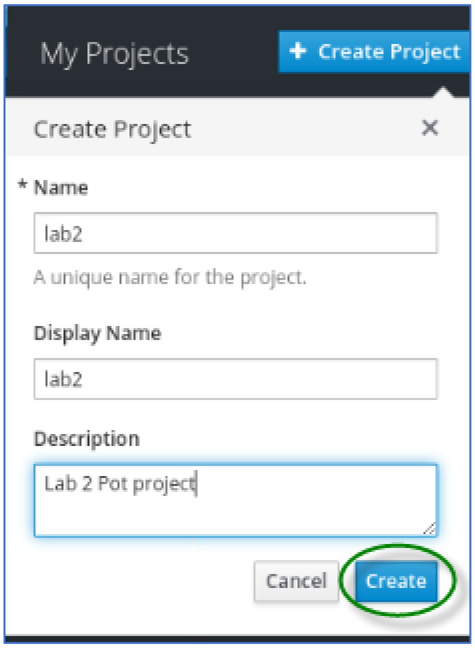
__12. Scroll down to view the `Resource Usage` for the project.



__13. Create a new project. Click on Service Catalog on the top of the page to return to the default view, and click on Create Project



__14. Enter lab2 for both the Name and Display Name and Lab 2 Pot project for the Description then click Create



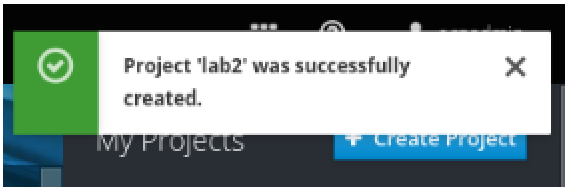
* Name: lab2

A unique name for the project.

Display Name: lab2

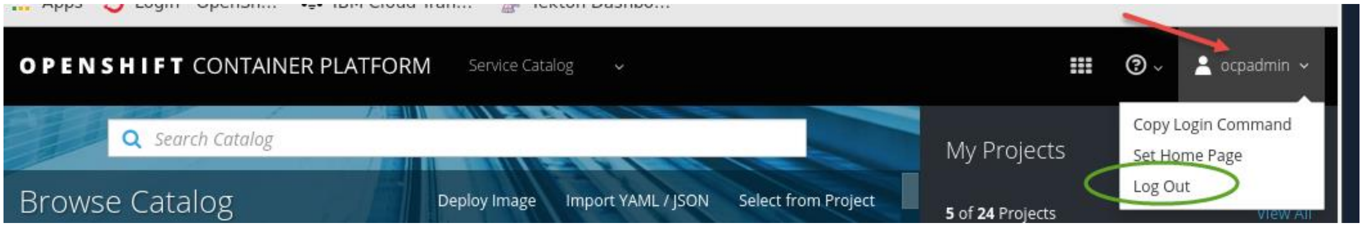
Description: Lab 2 Pot project

The project named “lab2” is successfully created.


Project 'lab2' was successfully created.

A project can also be created and managed from the command line.

__15. Log out of the console by clicking the `caret` next to `ocpadmin,` then clicking `Log Out. Then` close the browser.



__16. **Open a terminal** and change to the "**student**" directory by typing cd student

__17. **From the terminal window,** type `oc login` entering `ocpadmin` and `ocpadm1n` as username and password

These are the same credentials as you used when you logged into the console.

oc is the OpenShift Container Platform command line tool.

Once logged in, you see the list of projects that you have access to. The Asterisk (*) identifies the project that you are currently in. In this case, the "**default**" project.

```
[ibmdemo@icp4a student]$ oc login

Authentication required for https://icp4a.pot.com:8443 (openshift)
Username: ocpadmin
Password:
Login successful.

You have access to the following projects and can switch between them with 'oc project <projectname>':

  * default
    istio-system
    kabanero
    knative-eventing
    knative-serving
    knative-sources
    kube-public
    kube-service-catalog
    kube-system
    management-infra
    openshift
    openshift-console
    openshift-infra
    openshift-logging
    openshift-metrics-server
    openshift-monitoring
    openshift-node
    openshift-node-problem-detector
    openshift-pipelines
    openshift-sdn
    openshift-web-console
    operator-lifecycle-manager
    ta
Using project "default".
```

oc extends **kubectl**, the Kubernetes command line.

oc offers the same capabilities as the kubectl but it is further extended to natively support OpenShift Container Platform features, such as OpenShift resources such as DeploymentConfigs, BuildConfigs, Routes, ImageStreams, and ImageStreamTags which are specific to OpenShift distributions, and not available in standard Kubernetes.

__18. Enter `oc projects` which will list the available projects and inform you of the project you are currently using.

```
[ibmdemo@icp4a student]$ oc projects

* default
    istio-system
    kabanero
    knative-eventing
    knative-serving
    knative-sources
    kube-public
    kube-service-catalog
    kube-system
    lab2
    management-infra
    openshift
    openshift-console
    openshift-infra
    openshift-logging
    openshift-metrics-server
    openshift-monitoring
    openshift-node
    openshift-node-problem-detector
    openshift-pipelines
    openshift-sdn
    openshift-web-console
    operator-lifecycle-manager
    ta

Using project "default" on server "https://icp4a.pot.com:8443".
```

__19. Enter `oc get nodes` which will list the nodes, node in this case, in our cluster

```
[ibmdemo@icp4a student]$ oc get nodes

NAME            STATUS    ROLES                  AGE       VERSION
icp4a.pot.com   Ready     compute,infra,master   6d        v1.11.0+d4cacc0
```

__20. Now type `kubectl get nodes` which also provides the same list of nodes

```
[ibmdemo@icp4a student]$ kubectl get nodes

NAME            STATUS    ROLES                  AGE       VERSION
icp4a.pot.com   Ready     compute,infra,master   6d        v1.11.0+d4cacc0
```

__21. Type `oc get pods` which lists the pods in the `default` project (refer to the message at the end of the `oc login` command above)

```
[ibmdemo@icp4a student]$ oc get pods

NAME                       READY     STATUS    RESTARTS   AGE
docker-registry-1-m8977    1/1       Running   4          6d
router-1-sjbdd             1/1       Running   4          6d
```

__22. Type `oc get pods --all-namespaces`

```
[ibmdemo@icp4a student]$ oc get pods --all-namespaces

NAMESPACE                          NAME                                          READY
STATUS      RESTARTS   AGE
default                            docker-registry-1-m8977                       1/1
Running     4          6d
default                            router-1-sjbdd                                1/1
Running     4          6d
istio-system                       istio-ingressgateway-d897d9676-xzl42          1/1
Running     2          2d
istio-system                       istio-pilot-dcf4bd85c-vlxxp                   1/1
Running     2          2d
kabanero                           appsody-operator-549fd759c8-28gtx             1/1
Running     2          2d
kabanero                           controller-manager-0                          1/1
Running     4          2d
kabanero                           icpa-landing-768684bb7c-hv2sw                 1/1
Running     2          2d
kabanero                           kabanero-cli-654564cb49-tzz25                 1/1
Running     2          2d
kabanero                           kabanero-landing-fcf8788cc-b5gdw              1/1
Running     2          2d
kabanero                           kabanero-operator-8667c666bc-glpfg            1/1
Running     2          2d
kabanero                           knative-eventing-operator-67cdf5dc9f-ljg6x    1/1
Running     2          2d
kabanero                           knative-serving-operator-b64558bbc-4tvzh      1/1
Running     2          2d
kabanero                           openshift-pipelines-operator-66c4d787cf-dhbnm 1/1
Running     2          2d
kabanero                           tekton-dashboard-55fd66fbff-drcff             2/2
Running     4          2d
kabanero                           webhooks-extension-65d44777-hxlj6             1/1
Running     2          2d
knative-eventing                   eventing-controller-57dc75c787-lf52q          1/1
Running     2          2d
knative-eventing                   in-memory-channel-controller-d65c9bbdd-jd28s  1/1
Running     2          2d
knative-eventing                   in-memory-channel-dispatcher-5dcf5d557b-w44t7 1/1
Running     2          2d
knative-eventing                   sources-controller-5c65c58cfd-x7pqt           1/1
Running     2          2d
knative-eventing                   webhook-6cc6bbf964-trs89                      1/1
Running     2          2d
knative-serving                    activator-5647c87477-x7qrz                    1/1
Running     4          2d
knative-serving                    autoscaler-58746d858-69z88                    1/1
Running     2          2d
knative-serving                    controller-5f6d69cd4b-mlhhw                   1/1
Running     2          2d
knative-serving                    knative-openshift-ingress-6464b574f4-rvpr8    1/1
Running     2          2d
knative-serving                    networking-certmanager-677bf4f846-nrxqd       1/1
Running     2          2d
```

```
knative-serving                   networking-istio-6c885577bb-g2dt9                             1/1
Running     2          2d
knative-serving                   webhook-7645bc789f-vtrdw                                      1/1
Running     2          2d
kube-service-catalog              apiserver-jghnh                                               1/1
Running     4          6d
kube-service-catalog              controller-manager-n98hx                                      1/1
Running     17         6d
kube-system                       master-api-icp4a.pot.com                                      1/1
Running     4          6d
kube-system                       master-controllers-icp4a.pot.com                              1/1
Running     4          6d
kube-system                       master-etcd-icp4a.pot.com                                     1/1
Running     4          6d
openshift-console                 console-dd447dd5c-h2kqr                                       1/1
Running     4          6d
openshift-infra                   hawkular-cassandra-1-2k2gr                                    1/1
Running     4          6d
openshift-infra                   hawkular-metrics-schema-jgrk6                                 0/1
Completed   0          6d
openshift-infra                   hawkular-metrics-tgpbr                                        1/1
Running     4          6d
openshift-infra                   heapster-m2td8                                                1/1
Running     4          6d
openshift-metrics-server          metrics-server-56ff69fdff-snk4t                               1/1
Running     6          6d
openshift-monitoring              alertmanager-main-0                                           3/3
Running     12         6d
openshift-monitoring              alertmanager-main-1                                           3/3
Running     12         6d
openshift-monitoring              alertmanager-main-2                                           3/3
Running     12         6d
openshift-monitoring              cluster-monitoring-operator-57647f5877-qpckw                  1/1
Running     4          6d
openshift-monitoring              grafana-77cb866df7-dcbws                                      2/2
Running     8          6d
openshift-monitoring              kube-state-metrics-59c45bb4f6-b9hz9                           3/3
Running     12         6d
openshift-monitoring              node-exporter-vpsqp                                           2/2
Running     8          6d
openshift-monitoring              prometheus-k8s-0                                              4/4
Running     17         6d
openshift-monitoring              prometheus-k8s-1                                              4/4
Running     17         6d
openshift-monitoring              prometheus-operator-6bc9848445-pdbm6                          1/1
Running     4          6d
openshift-node-problem-detector   node-problem-detector-zpkqm                                   1/1
Running     4          6d
openshift-node                    sync-ndr8t                                                    1/1
Running     4          6d
openshift-pipelines               tekton-pipelines-controller-6c9778c6d4-x78sp                  1/1
Running     2          2d
openshift-pipelines               tekton-pipelines-webhook-76686fd9c7-bdpnm                     1/1
Running     2          2d
openshift-sdn                     ovs-dq9xf                                                     1/1
Running     4          6d
openshift-sdn                     sdn-xkrtz                                                     1/1
Running     4          6d
openshift-web-console             webconsole-85777774db-454rv                                   1/1
Running     6          6d
operator-lifecycle-manager        catalog-operator-96d84b9d-8bfzs                               1/1
Running     4          6d
operator-lifecycle-manager        olm-operator-89dcb84cf-qvlps                                  1/1
Running     4          6d
ta                                ta-9mwaqdq2nrskos5kkb12c9ss2-ta-rh-couchdb-5d9f74586c-md9d2   1/1
Running     2          2d
ta                                ta-9mwaqdq2nrskos5kkb12c9ss2-ta-rh-server-6d4c6c6b6c-wsxwh    1/1
Running     2          2d
```

```
ta                                  ta-9mwaqdq2nrskos5kkb12c9ss2-ta-rh-ui-75bbc6c6d6-x6f8h        1/1
Running     2          2d
ta                                  ta-operator-649b6b7c68-qs58f                                  1/1
Running     2          2d
```

__23. Pay attention that all pods are “running” or “completed” Make sure that the running containers have all the required number of ready containers e.g 1/1 or 3/3. Completed pods are pods that perform tasks during ICP startup

__24. You can learn more about Kubernetes in lab 03 “Getting Started with Kubernetes”, which is part of this proof of technology PoT.

End of Lab 02 – Explore RedHat OpenShift Container Platform

Appendix: SkyTap Tips for labs

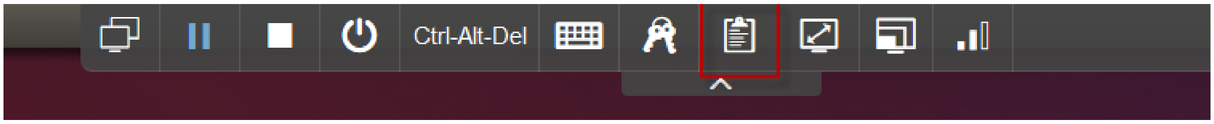
2.1 How to use Copy / Paste between local desktop and Skytap VM?

Using copy / Paste capabilities between the lab document (PDF) on your local workstation to the VM is a good approach to more efficiently work through a lab, while reducing the typing errors that often occur when manually entering data.

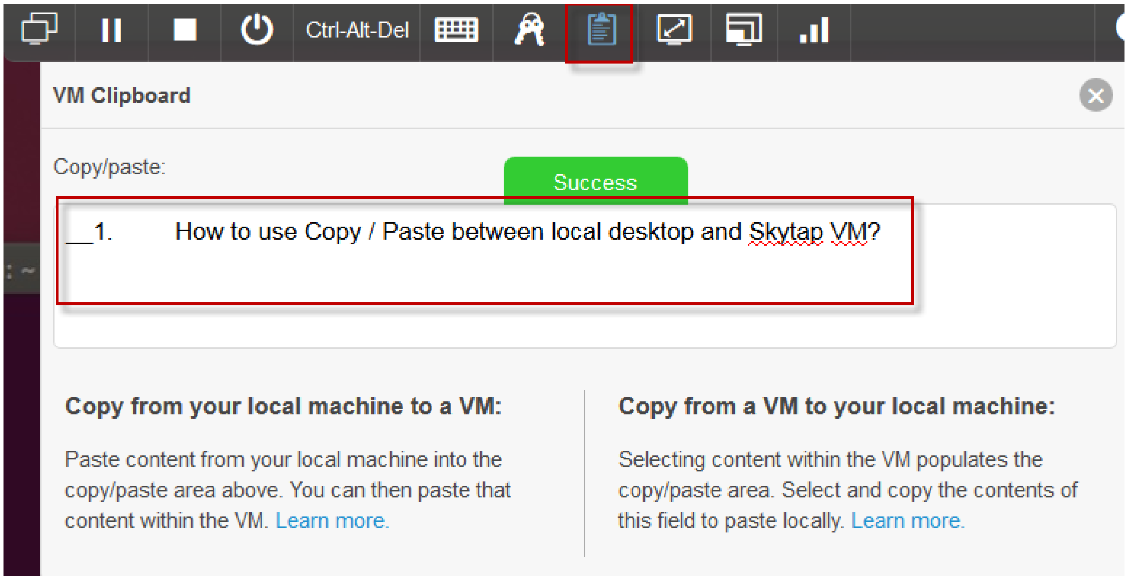
__1. In SkyTap, you will find that any text copied to the clipboard on your local workstation is not available to be pasted into the VM on SkyTap. So how can you easily accomplish this?

__a. First copy the text you intend to paste, from the lab document, to the clipboard on your local workstation, as you always have (CTRL-C)

__b. Return to the SkyTap environment and click on the Clipboard at the top of the SkyTap session window.



__c. Use **CTRL-V** to paste the content into the Copy/paste VM clipboard. Or use the **paste** menu item that is available in the dialog, when you right mouse click in the clipboard text area.



__1. How to use Copy / Paste between local desktop and Skytap VM?

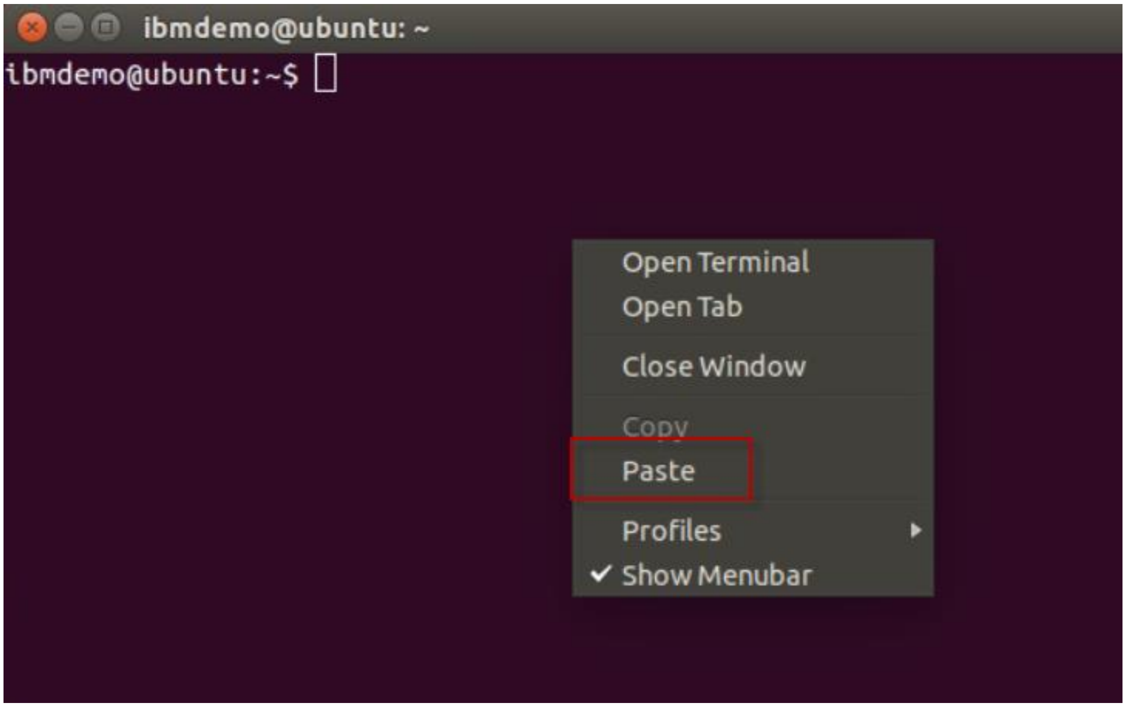
Copy from your local machine to a VM:

Paste content from your local machine into the copy/paste area above. You can then paste that content within the VM. Learn more.

Copy from a VM to your local machine:

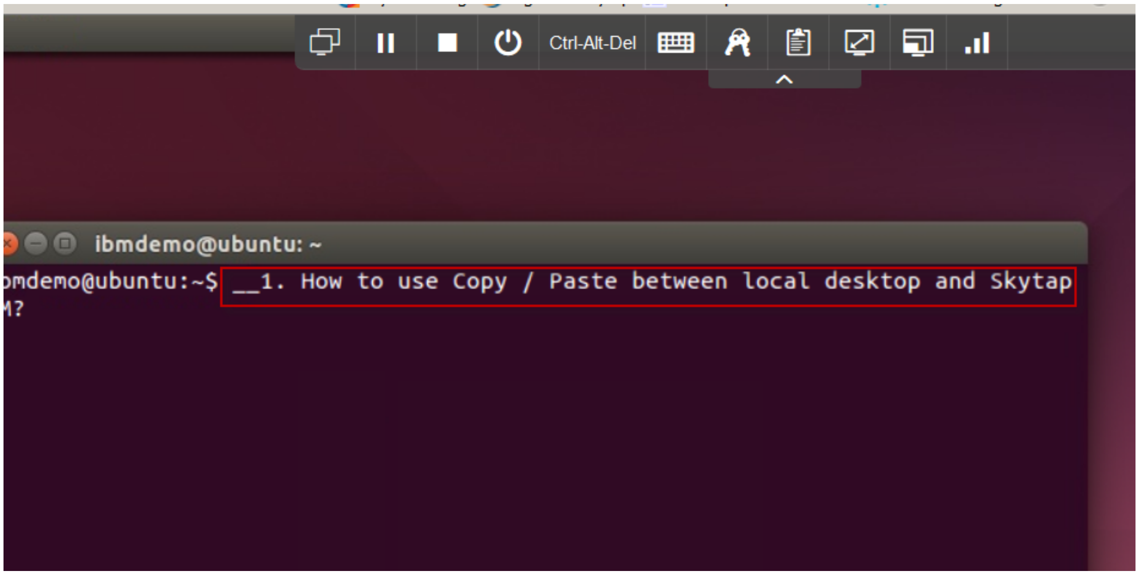
Selecting content within the VM populates the copy/paste area. Select and copy the contents of this field to paste locally. Learn more.

__d. Once the text is pasted, just navigate away to the VM window where you want to paste the content. Then, use **CTRL-C**, or right mouse click & us the **paste menu item** to paste the content.



```
ibmdemo@ubuntu:~$
```

__e. The text is pasted into the VM



```
omdemo@ubuntu:~$ __1. How to use Copy / Paste between local desktop and Skytap
M?
```

Note: The very first time you do this, if the text does not paste, you may have to paste the contents into the Skytap clipboard twice. This is a known Skytap issue. It only happens on the 1st attempt to copy / paste into Skytap.