

IBM Cloud Private 3.1.2

**Lab Exercise #1**

**IBM Cloud Private Console Walkthrough**

**Duration:** 30 mins

---

## **1 TABLE OF CONTENTS**

2	Login to ICP .....	3
2.1	Overview .....	3
2.2	Access your Master VM.....	3
2.3	Login to your ICP Console .....	3
2.4	Getting Started.....	5
2.5	ICP Version Check.....	5
3	ICP Console .....	7
3.1	Dashboard.....	7
3.2	Nodes.....	8
3.3	Namespaces.....	9
3.4	Helm Charts.....	11
3.5	Storage.....	14
3.6	Monitoring.....	16
3.7	Alerts .....	18
3.8	Deployments.....	20
3.9	StatefulSets.....	22
3.10	DaemonSets .....	24
3.11	Services.....	26
3.12	Command Line Parameters .....	28
4	Lab Summary .....	30

---

## **2Login to ICP**

---

### **2.1 Overview**

---

In this lab exercise, you explore the IBM Cloud Private Administration Console

---

### **2.2 Access your Master VM**

---

Using the **MASTER** VM in your ICP environment, log in as <username> with the password <password>

---

### **2.3 Login to your ICP Console**

---

If you are not already logged in to the ICP Admin Console from a previous exercise, open a browser and navigate to <https://<ip>:<port>>

IBM **Cloud** Private

Fast. Flexible. Intelligent.  
Open. Enterprise-grade.

---

Log in to your account

**Username**

**Password**

**Log in**

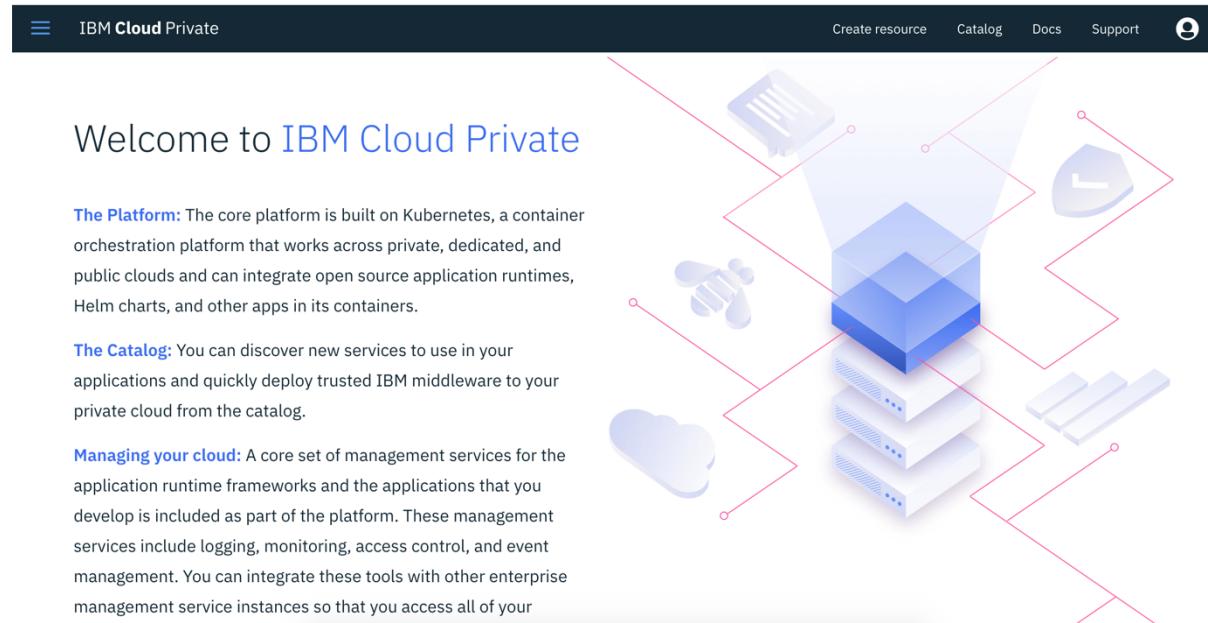
Log in by using **username: <username> and password: <password>**

---

## 2.4 Getting Started

---

The **Welcome** page displays after you successfully log in.



**Welcome to IBM Cloud Private**

**The Platform:** The core platform is built on Kubernetes, a container orchestration platform that works across private, dedicated, and public clouds and can integrate open source application runtimes, Helm charts, and other apps in its containers.

**The Catalog:** You can discover new services to use in your applications and quickly deploy trusted IBM middleware to your private cloud from the catalog.

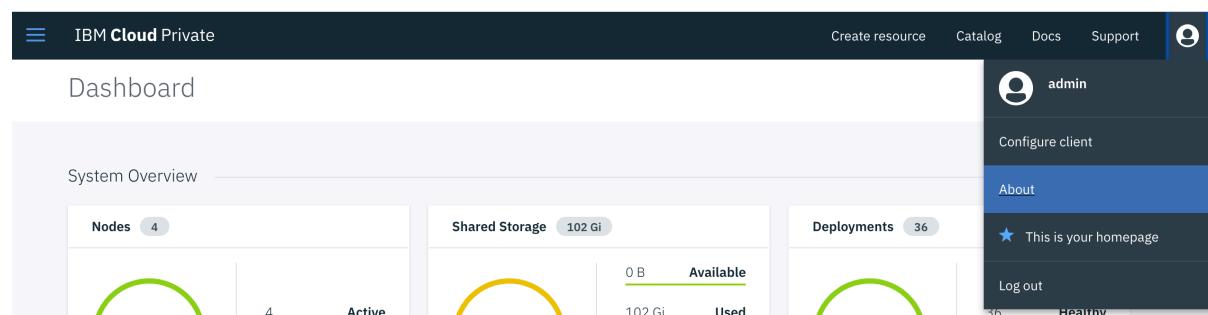
**Managing your cloud:** A core set of management services for the application runtime frameworks and the applications that you develop is included as part of the platform. These management services include logging, monitoring, access control, and event management. You can integrate these tools with other enterprise management service instances so that you access all of your

---

## 2.5 ICP Version Check

---

To check the version of the IBM Cloud Private on Console, click on the top right profile icon and click on “**About**”.



Dashboard

System Overview

Nodes	4
Active	4

Shared Storage	102 Gi
Used	0 B
Available	102 Gi

Deployments	36
Healthy	36

admin

Configure client

**About**

This is your homepage

Log out

X

# IBM Cloud Private

VERSION **3.1.2 Enterprise Edition**

[Copyright information](#)

## TECHNOLOGIES USED

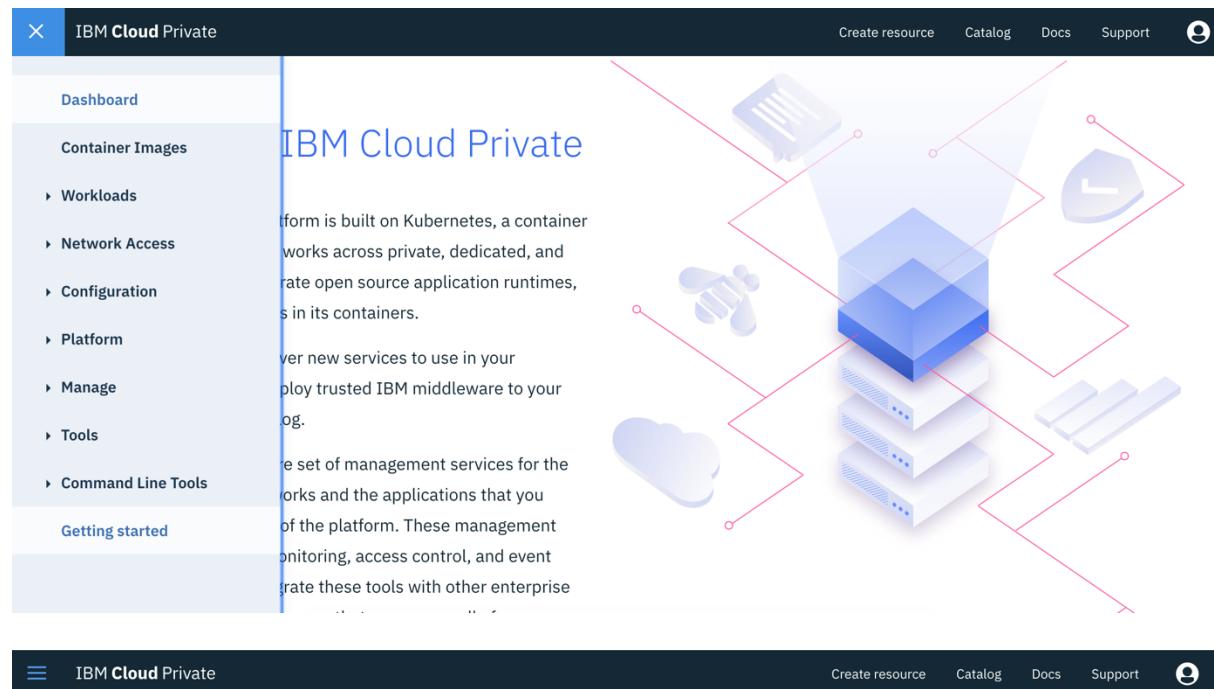


## 3 ICP Console

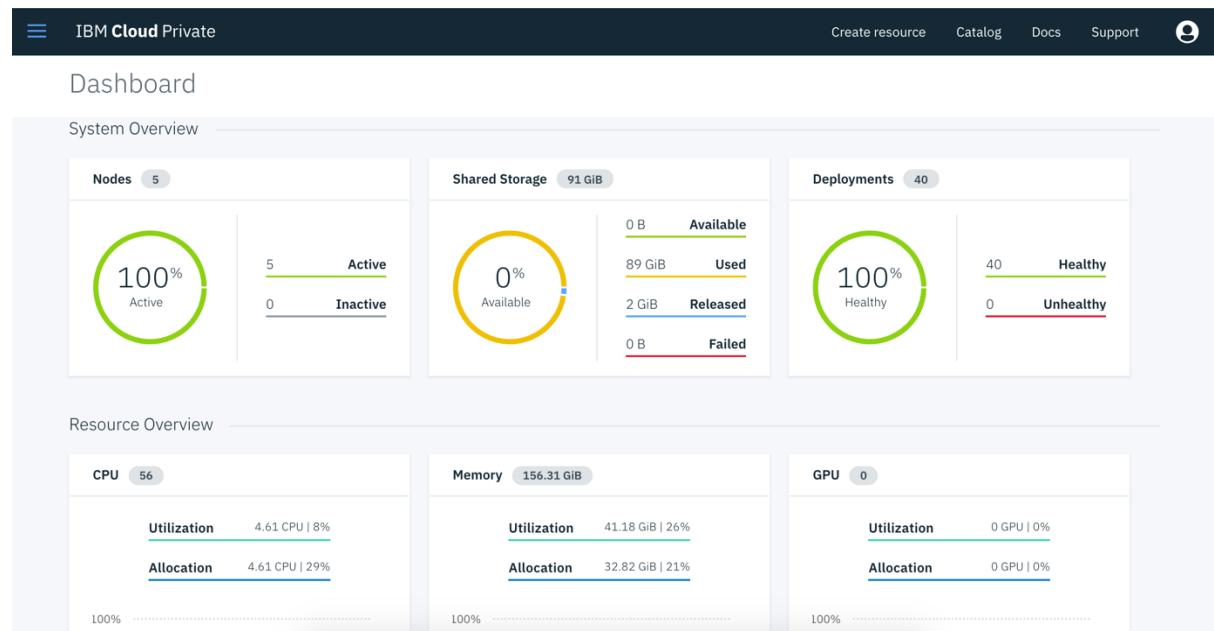
### 3.1 Dashboard

[https://www.ibm.com/support/knowledgecenter/SSBS6K\\_3.1.2/user\\_management/assign\\_role.html](https://www.ibm.com/support/knowledgecenter/SSBS6K_3.1.2/user_management/assign_role.html)

Click **Menu** in the top left corner of the page, and then select **Dashboard** to navigate to the Dashboard page. The Dashboard page provides an overview of the current status of the ICP cluster.



The screenshot shows the IBM Cloud Private dashboard. On the left is a sidebar with navigation links: Dashboard, Container Images, Workloads, Network Access, Configuration, Platform, Manage, Tools, and Command Line Tools. Below these are sections for Getting started and Help & Support. The main area features a large central cluster node with various icons representing different services and components connected to it, such as databases, storage, and monitoring tools. The overall theme is a modern, cloud-native architecture.



The screenshot shows the IBM Cloud Private dashboard with a "System Overview" section. It includes three cards: "Nodes" (5 total, 100% Active), "Shared Storage" (91 GiB total, 0% Available), and "Deployments" (40 total, 100% Healthy). Below this is a "Resource Overview" section with three cards: "CPU" (56 total, Utilization 4.61 CPU | 8%, Allocation 4.61 CPU | 29%), "Memory" (156.31 GiB total, Utilization 41.18 GiB | 26%, Allocation 32.82 GiB | 21%), and "GPU" (0 total, Utilization 0 GPU | 0%, Allocation 0 GPU | 0%). The dashboard has a clean, modern design with a dark header and light-colored cards for data presentation.

## 3.2 Nodes

Click **Menu**, and then select **Platform > Nodes** to navigate to the Nodes page. This page displays information about the nodes that are part of the ICP Cluster.

The screenshot shows the IBM Cloud Private Platform > Nodes page. On the left, a sidebar menu includes Container Images, Workloads, Network Access, Configuration, Platform (selected), Alerting, Logging, Metering, Monitoring, Network, Nodes (selected), and Storage. The main area displays four cards: Shared Storage (91 GiB total, 0% Available, 89 GiB Used, 2 GiB Released, 0 B Failed), Deployments (40 total, 100% Healthy, 40 Healthy, 0 Unhealthy), Memory (156.31 GiB total, Utilization 41.14 GiB | 26%, Allocation 32.82 GiB | 21%), and GPU (0 total, Utilization 0 GPU | 0%, Allocation 0 GPU | 0%).

The screenshot shows the IBM Cloud Private Nodes page. At the top, there is a search bar labeled "Search". Below it is a table with columns: Name, Role, Architecture, Status, Schedulable, and Created. The table lists five nodes:

Name	Role	Architecture	Status	Schedulable	Created
<a href="#">10.41.14.42</a>	worker	amd64	Active	Schedulable	10 days ago
<a href="#">10.41.12.238</a>	management	amd64	Active	Schedulable	10 days ago
<a href="#">10.41.12.241</a>	master, etcd	amd64	Active	Schedulable	10 days ago
<a href="#">10.41.13.130</a>	proxy	amd64	Active	Schedulable	10 days ago
<a href="#">10.41.14.41</a>	worker	amd64	Active	Schedulable	10 days ago

At the bottom, there is a pagination bar with "items per page: 20 ▾ | 1-5 of 5 items" and "1 of 1 pages" with navigation arrows.

Click the **Name** of a node to *drill down* and see more information about the node.

The screenshot shows the IBM Cloud Private interface. At the top, there's a dark header bar with the IBM Cloud Private logo, a search bar containing 'Nodes / 10.41.12.241 /', and navigation links for 'Create resource', 'Catalog', 'Docs', 'Support', and a user icon. Below the header, the IP address '10.41.12.241' is displayed. A 'Overview' section is selected, indicated by an underline. Underneath is a table titled 'Node details' with the following data:

Type	Detail
Hostname	10.41.12.241
Unschedulable	Schedulable
Status	Active
Address	10.41.12.241
Labels	beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,etcld=true,kubernetes.io/hostname=10.41.12.241,master=true,node-role.kubernetes.io/etcd=true,node-role.kubernetes.io/master=true,role=master
CPU	8
Memory	31.26 GiB

### 3.3 Namespaces

Click **Menu** and then select **Manage > Namespaces** to navigate to the Namespaces page.



## ▶ Workloads

## ▶ Network Access

## ▶ Configuration

## ▶ Platform

## ▼ Manage

Identity & Access

Resource Security

Service Brokers

Helm Repositories

Namespaces

The screenshot shows the 'Namespaces' section of the IBM Cloud Private interface. At the top right are navigation links: 'Create resource', 'Catalog', 'Docs', 'Support', and a user icon. Below the header is a search bar and a 'Create Namespace' button. The main area displays a table of namespaces:

Name	Status	Pod Security Policy
cert-manager	Active	ibm-restricted-psp
default	Active	ibm-anyuid-hostaccess-psp, ibm-anyuid-hostpath-psp, ibm-anyuid-psp, ibm-privileged-psp, ibm-restricted-psp
ibmcom	Active	ibm-restricted-psp
istio-system	Active	ibm-privileged-psp, ibm-restricted-psp
kube-public	Active	ibm-restricted-psp
kube-system	Active	ibm-restricted-psp

Users are assigned to organizational units called namespaces. Namespaces are also known as tenants or accounts. In IBM Cloud Private, users are assigned to teams. You can assign multiple namespaces to a team. Users of a team are members of the team's namespaces.

## 3.4 Helm Charts

Click **Catalog** on the navigation bar to navigate to the Helm Chart Catalog page.

The screenshot shows the 'Catalog' section of the IBM Cloud Private interface. At the top right are navigation links: 'Create resource', 'Catalog', 'Docs', 'Support', and a user icon. Below the header is a search bar. The main area displays a sidebar with 'All Categories' and a list of categories: DevOps, Operations, Security, Network, Data Science & Analytics, Data, Integration, Storage, and Blockchain. To the right, there is a 'Helm Charts' section with four cards:

- artifactory-ha** (ibm-community-charts) - Universal Repository Manager supporting all major packaging formats, build tools and CI servers.
- audit-logging** (mgmt-charts) - Audit logging storage and search management solution.
- auth-apikeys** (mgmt-charts) - ICP IAM Token Service.
- auth-idp** (mgmt-charts) - ICP Security Authentication Provider.

By using the Catalog, you can browse and install packages in your IBM Cloud Private cluster from Helm charts.

The Catalog displays Helm charts, which contain application packages that can run as Kubernetes services. The packages are stored in repositories. The Catalog in IBM Cloud Private contains connections to recommended repositories by default, but you can connect to other repositories. After you connect to a repository, you can access its charts from the

---

Catalog. Application developers can also develop applications and publish them in the Catalog so that other users can easily access and install the applications.

\*\* If your catalog page is empty, click `Menu --> Manage > Helm Repositories`, click `Sync Repositories` and after a couple of minutes return to the Catalog page \*\*

Note: Click on the Helm Chart name to view the readme file.

Click **Menu** and then select **Manage > Helm Repositories** to navigate to the list of configured Helm repositories page.



IBM Cloud Private

## Container Images

- ▶ Workloads
  - ▶ Network Access
  - ▶ Configuration
  - ▶ Platform
- 
- ▼ Manage

Identity & Access

Resource Security

Service Brokers

Helm Repositories

Namespaces



## Repositories

Search repositories		Sync repositories	Add repository
NAME	URL	ACTION	
local-charts	https://ibmblore.icp:8443/helm-repo/charts	⋮	⋮
ibm-charts	https://raw.githubusercontent.com/IBM/charts/master/repo/stable/	⋮	⋮
mgmt-charts	https://ibmblore.icp:8443/mgmt-repo/charts	⋮	⋮
ibm-community-charts	https://raw.githubusercontent.com/IBM/charts/master/repo/community/	⋮	⋮
ppc64le-isv-charts	https://raw.githubusercontent.com/ppc64le/charts/master/repo/stable/	⋮	⋮
ibm-charts-public	https://registry.bluemix.net/helm/ibm/	⋮	⋮
check	https://raw.githubusercontent.com/IBM/charts/master/repo/stable	⋮	⋮
items per page	20 ▾   1-7 of 7 items	1 of 1 pages	< 1 ▾ >

Note: If you want to connect to other repositories, you can add them here.

## 3.5 Storage

Click **Menu** and then select **Platform > Storage** to navigate to the Storage page.



IBM Cloud Private

▶ Workloads

▶ Network Access

▶ Configuration

▼ Platform

Alerting

Logging

Metering

Monitoring

Network Policies

Nodes

Storage

## Storage

[PersistentVolume](#)

[PersistentVolumeClaim](#)

Search								<a href="#">Create PersistentVolume</a> +
Name	Type	Capacity	Access mode	Reclaim policy	Status	Claim		Created
pvc-0906d572-61ba-11e9-a85a-00163e01d917	Glusterfs	10Gi	RWO	Delete	Bound	default/data-mylogs-ibm-icplogging-data-0		16 days ago
pvc-090c8496-61ba-11e9-a85a-00163e01d917	Glusterfs	10Gi	RWO	Delete	Bound	default/data-mylogs-ibm-icplogging-data-1		16 days ago
pvc-1246a1cf-601c-11e9-a85a-00163e01d917	Glusterfs	2Gi	RWO	Delete	Bound	default/checkflag		18 days ago
mgmt-repo-pv	Hostpath	5Gi	RWO	Delete	Bound	kube-system/mgmt-repo-pvc		27 days ago
helm-repo-pv	Hostpath	5Gi	RWO	Delete	Bound	kube-system/helm-repo-pvc		27 days ago

Data storage in a Kubernetes cluster is handled by using volumes. For Kubernetes, a PersistentVolume (PV) is a piece of networked storage in a cluster that is provisioned by an administrator. A PersistentVolumeClaim (PVC) is a request for storage that is made by a user.

In an IBM Cloud Private cluster, administrators can create PersistentVolumes that are available to all projects in the cluster. Users can then create PersistentVolumeClaims to request this resource for their application. All PersistentVolume types that are supported by

Kubernetes are also supported by IBM Cloud Private.

Click **PersistentVolumeClaim** to see the current Persistent Volume Claims in the ICP Cluster

## Storage

[PersistentVolume](#)

[PersistentVolumeClaim](#)

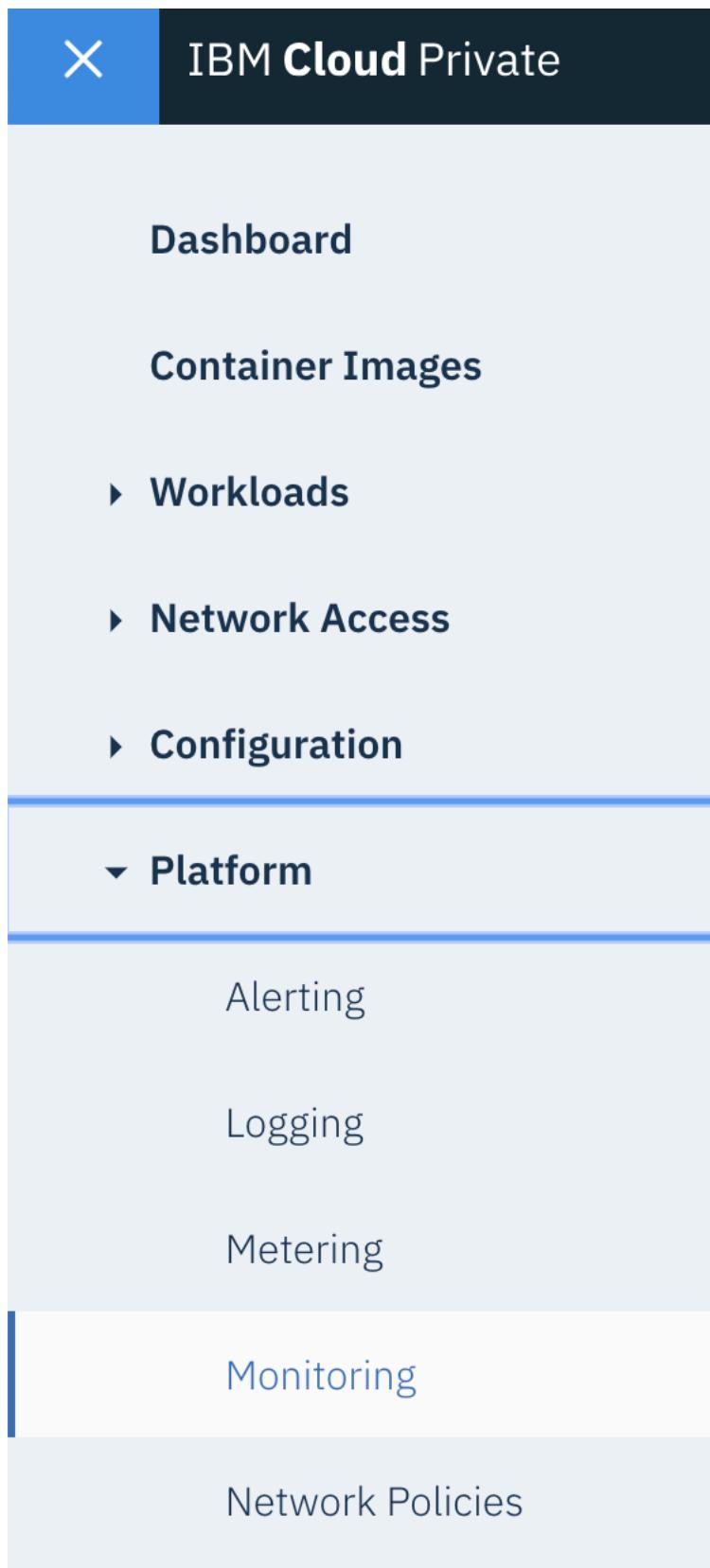
All namespaces ▾

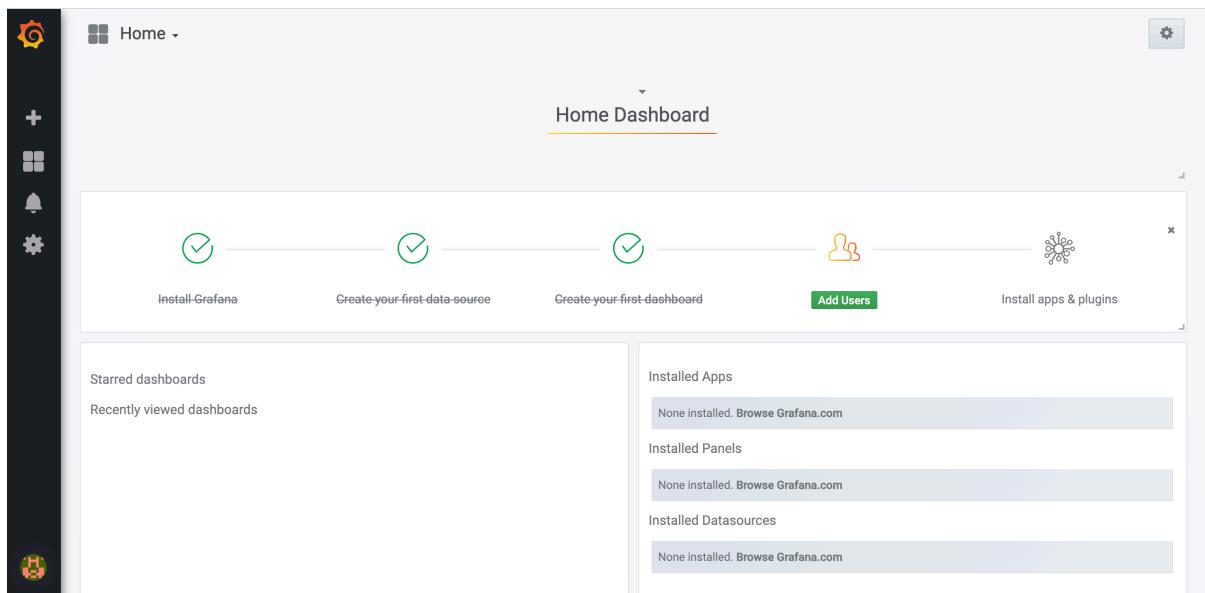
Search							<a href="#">Create PersistentVolumeClaim</a> +
Name	Namespace	Status	PersistentVolume	Requests	Access mode	Created	
data-mylogs-ibm-icplogging-data-0	default	Bound	pvc-0906d572-61ba-11e9-a85a-00163e01d917	10Gi	RWO	16 days ago	
data-mylogs-ibm-icplogging-data-1	default	Bound	pvc-090c8496-61ba-11e9-a85a-00163e01d917	10Gi	RWO	16 days ago	
checkflag	default	Bound	pvc-1246a1cf-601c-11e9-a85a-00163e01d917	2Gi	RWO	18 days ago	
mgmt-repo-pvc	kube-system	Bound	mgmt-repo-pv	5Gi	RWO	27 days ago	
helm-repo-pv	kube-system	Bound	helm-repo-pv	5Gi	RWO	27 days ago	

## 3.6 Monitoring

---

Click **Menu** and then select **Platform > Monitoring** to open Grafana in a new browser window.





**NOTE** If the page doesn't display correctly it may be due to prometheus not starting correctly in your lab environment. Execute the following steps to restart any failed prometheus pods:

- open a terminal session and enter the following command. When prompted log in as *user*: <username> with *password*: <password> and select the default namespace

```
cloudctl login -a https://<ip>:<port>
```

- enter the following command to find the failed pod

```
kubectl get pods -n kube-system | grep monitoring-prometheus
```

- take the pod name of the failed pod (in `crashloopbackoff` state) and execute the following command

```
kubectl delete pod -n kube-system <podname>
```

e.g.: `kubectl delete pod -n kube-system monitoring-prometheus-fdb49f66b-jp6dw`

- after a few moments, refresh the grafana page

Close the Grafana browser tab.

---

### 3.7 Alerts

---

Click **Menu** and then select **Platform > Alerting** to open the ICP Alert Manager in a new browser tab.



Dashboard

Container Images

▶ Workloads

▶ Network Access

▶ Configuration

▼ Platform

Alerting

Logging

Metering

Monitoring

Network Policies

The screenshot shows the Alertmanager interface with the following details:

- Header:** Alertmanager, Alerts, Silences, Status, New Silence
- Filter/Group:** Filter (selected), Group
- Receiver:** All, Silenced, Inhibited
- Search Bar:** Custom matcher, e.g. `env="production"`
- Alert Entries:**
  - Alert 1:** alertername="HighCPUUsage" (highlighted), +  
06:36:23, 2019-05-04 + Info Source Silence  
instance="172.16.217.55:8445" +
  - Alert 2:** 06:36:23, 2019-05-04 + Info Source Silence  
instance="172.16.217.23:8445" +

Close the ICP Alert Manager tab.

## 3.8 Deployments

### Deployments

Click **Menu** and then select **Workloads > Deployments** to navigate to the Deployments page.



IBM Cloud Private

**Dashboard**

**Container Images**

▼ **Workloads**

Brokered Services

DaemonSets

Deployments

Helm Releases

Jobs

StatefulSets

Click

Name	Namespace	Desired	Current	Ready	Available	Created	
monitoring-grafana	kube-system	1	1	1	1	2 days ago	<a href="#">Launch</a>
monitoring-prometheus	kube-system	1	1	1	1	2 days ago	<a href="#">Launch</a>
monitoring-prometheus-alertmanager	kube-system	1	1	1	1	2 days ago	<a href="#">Launch</a>
monitoring-prometheus-collectdexporter	kube-system	1	1	1	1	2 days ago	<a href="#">Launch</a>
monitoring-prometheus-elasticsearchexporter	kube-system	1	1	1	1	2 days ago	<a href="#">Launch</a>
monitoring-prometheus-kubestatemetrics	kube-system	1	1	1	1	2 days ago	<a href="#">Launch</a>

A **Deployment controller** provides declarative updates for **Pods and ReplicaSets**. You describe a desired state in a Deployment object, and the Deployment controller changes the actual state to the desired state at a controlled rate. You can define Deployments to create new ReplicaSets, or to remove existing Deployments and adopt all their resources with new Deployments.

Notes:

1. Click on a Deployment Name to *drill down* and see more information, such as the **Pods** that are part of the Deployment. Scroll down, and click a Pod to find out information about the Container and Logs.
2. Use the **Namespace** drop-down menu in the top right corner of the page to change the Deployments that are displayed.

## 3.9 StatefulSets

Click **Menu** and then select **Workloads > StatefulSets** to navigate to the StatefulSets page.



IBM Cloud Private

Dashboard

Container Images

▼ Workloads

Brokered Services

DaemonSets

Deployments

Helm Releases

Jobs

StatefulSets

Name	Namespace	Desired	Current	Created
logging-elk-data	kube-system	1	1	5 days ago
mariadb	kube-system	1	1	27 days ago
icp-mongodb	kube-system	1	1	27 days ago
image-manager	kube-system	1	1	27 days ago

Like a Deployment, a StatefulSet manages Pods that are based on an identical container spec. Unlike a Deployment, a StatefulSet maintains a sticky identity for each of their Pods. These pods are created from the same spec, but are not interchangeable: each has a persistent identifier that it maintains across any rescheduling.

## 3.10 DaemonSets

Click **Menu**, and then select **Workloads > DaemonSets** to navigate to the DaemonSets page.



IBM Cloud Private

**Dashboard**

**Container Images**

▼ **Workloads**

Brokered Services

DaemonSets

Deployments

Helm Releases

Jobs

StatefulSets

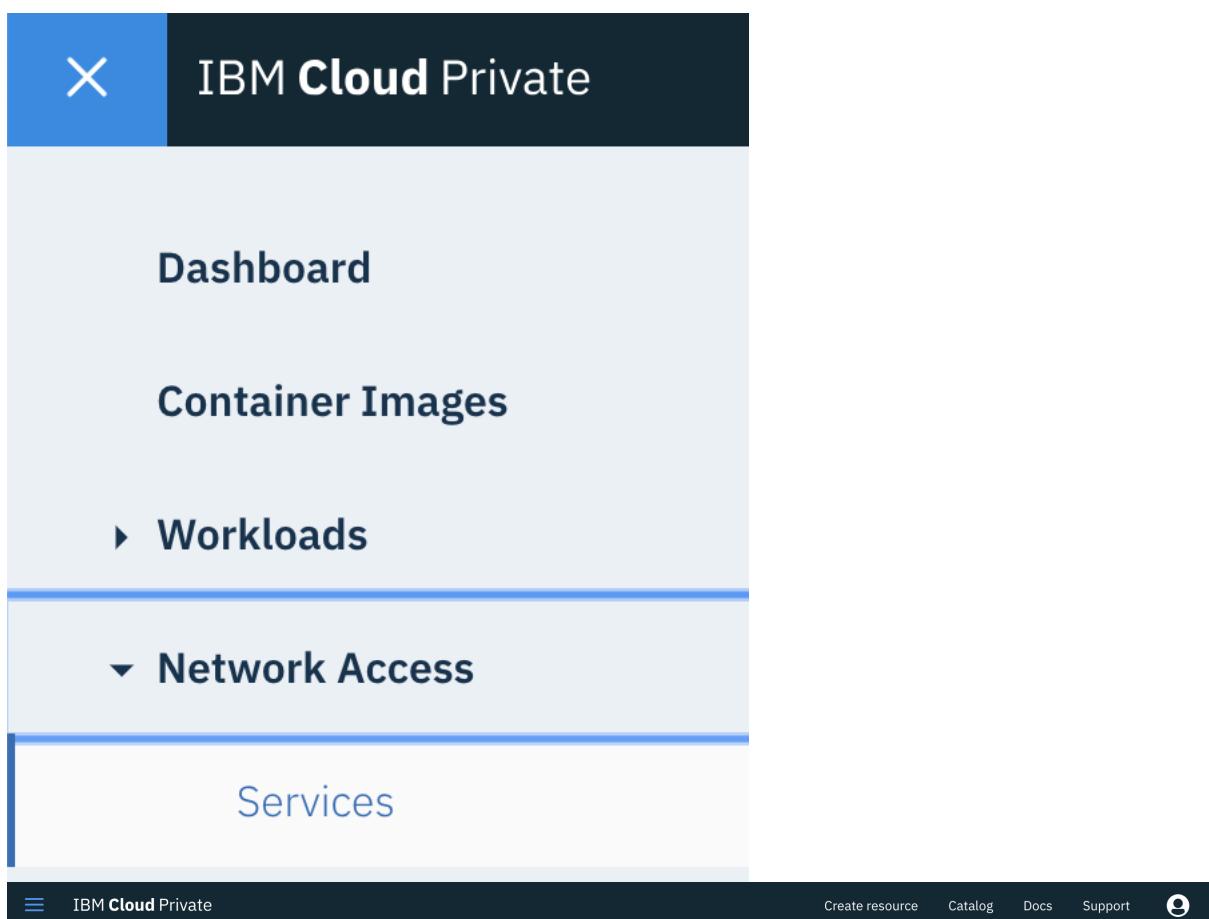
ReplicaSets

IBM Cloud Private		Create resource	Catalog	Docs	Support			
DaemonSets						All namespaces		
Search								
Name	Namespace	Desired	Current	Ready	Up-to-date	Available	Node selector	Created
monitoring-prometheus-nodeexporter	kube-system	4	4	4	4	4	-	2 days ago
rook-ceph-agent	rook-operator	2	2	2	2	2	-	2 days ago
rook-discover	rook-operator	2	2	2	2	2	-	2 days ago
logging-elk-filebeat-ds	kube-system	4	4	4	4	4	-	5 days ago
node-problem-detector	kube-system	4	4	4	4	4	-	12 days ago
audit-logging-fluentd-ds	kube-system	4	4	4	4	4	-	27 days ago

A DaemonSet ensures that all (or some) Nodes run a copy of a Pod. As nodes are added to the cluster, Pods are added to the cluster. As nodes are removed from the cluster, those Pods are garbage-collected. Deleting a DaemonSet cleans up the Pods that it created.

## 3.11 Services

Click **Menu**, and then select **Network Access > Services** to navigate to the Services page.



## Services

[Services](#)   [Ingress](#)

All namespaces ▾

Search

Create Service +

Name	Namespace	Created
monitoring-prometheus	kube-system	2 days ago
monitoring-grafana	kube-system	2 days ago
monitoring-prometheus-alertmanager	kube-system	2 days ago
monitoring-prometheus-collectdexporter	kube-system	2 days ago
monitoring-prometheus-elasticsearchexporter	kube-system	2 days ago

Kubernetes Pods are mortal; when they die, they are not resurrected. ReplicaSets in particular create and destroy Pods dynamically (when scaling up or down). While each Pod gets its own IP address, even those IP addresses cannot be relied upon to be stable over time. This leads to a problem: if some set of Pods (for example, call them backends) provides functionality to other Pods (for example, call them frontends) inside the Kubernetes cluster, how do those frontends find out and keep track of which backends are in that set?

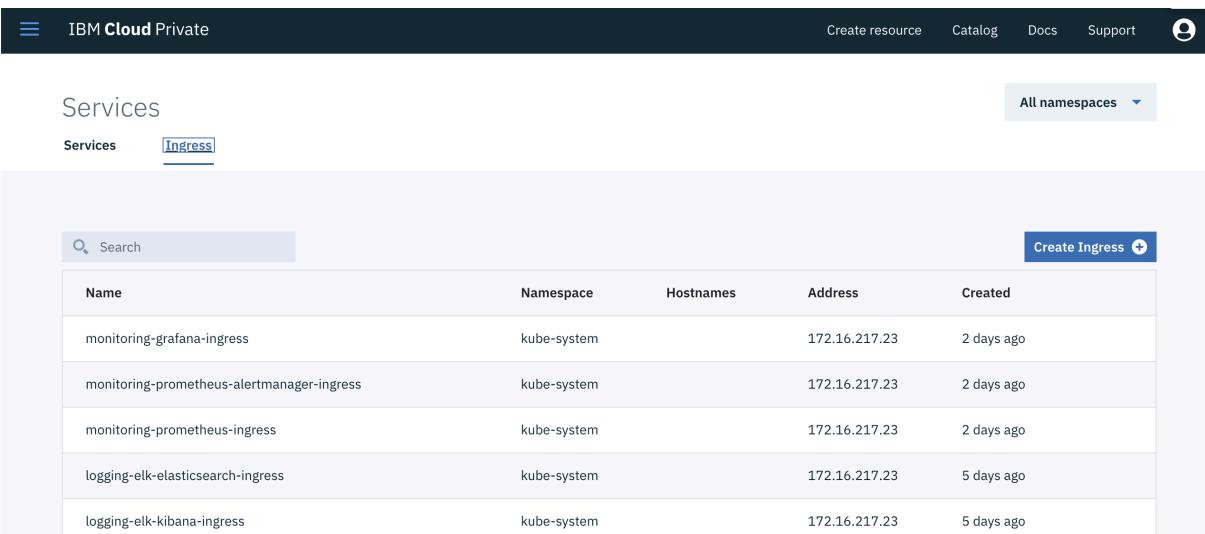
Enter Services.

---

A Kubernetes Service is an abstraction that defines a logical set of Pods and a policy by which to access them - sometimes called a microservice. The set of Pods targeted by a Service is typically determined by a Label Selector.

Note: Use the drop-down menu in the upper left corner to customize the number of items displayed per page, or click > in the upper right corner to page through a long list of services. Click a service to view more information about it. Click the Action link to view and edit the contents of a service.

Click **Ingress** to navigate to the Ingress page.



The screenshot shows the IBM Cloud Private Services dashboard. At the top, there is a navigation bar with links for 'Create resource', 'Catalog', 'Docs', and 'Support'. Below the navigation bar, there is a search bar labeled 'Search' and a dropdown menu labeled 'All namespaces'. The main area is titled 'Services' and contains two tabs: 'Services' and 'Ingress', with 'Ingress' being the active tab. A table below lists five ingress objects, each with columns for Name, Namespace, Hostnames, Address, and Created. The table includes a 'Create Ingress' button in the top right corner.

Name	Namespace	Hostnames	Address	Created
monitoring-grafana-ingress	kube-system		172.16.217.23	2 days ago
monitoring-prometheus-alertmanager-ingress	kube-system		172.16.217.23	2 days ago
monitoring-prometheus-ingress	kube-system		172.16.217.23	2 days ago
logging-elk-elasticsearch-ingress	kube-system		172.16.217.23	5 days ago
logging-elk-kibana-ingress	kube-system		172.16.217.23	5 days ago

An Ingress is an API object that manages external access to the services in a cluster, typically HTTP. Ingress can provide load balancing, SSL termination and name-based virtual hosting.

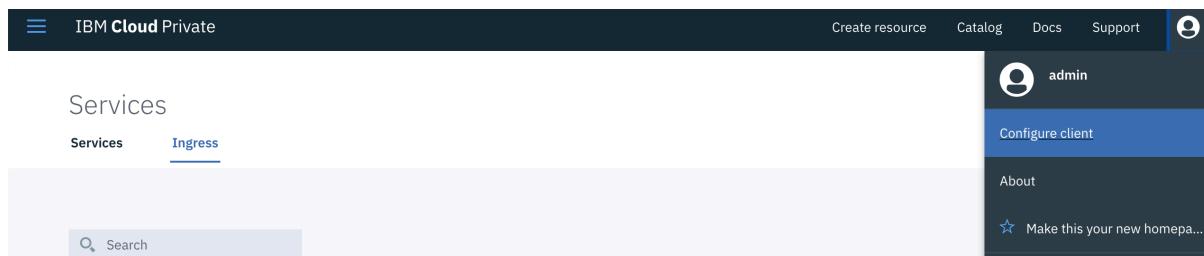
---

### 3.12 Command Line Parameters

---

---

Click the **User** icon on the navigation bar and then select **Configure Client** to display the commands that are used to configure a kubectl command line to connect to this ICP Cluster.



Close the **Configure client** dialog box.

---

## 4Lab Summary

In this lab exercise, you explored the IBM Cloud Private Administration Console. You learned about:

- The ICP Admin Console dashboard
- Nodes, Namespaces, Deployments, Stateful Sets, Daemon Sets, Services, and Ingress
- Helm Charts
- Storage, Monitoring and Alerts

---

---