Practice: Deploy an OKE Cluster Using Cloud Shell

Try this hands-on lab with the Oracle Cloud 30-day Free Trial account or your own tenancy. If you do not have a free account, click here to get one.

Overview

You will perform the following tasks in this practice:

- Launch Cloud Shell
- Generate an ssh Key
- Create a Kubernetes Cluster
- Deploy a Sample Nginx App on the Cluster Using kubectl

Practice: Launching Cloud Shell

Overview

In this practice, you will launch a Cloud Shell session for the OCI user account assigned to you in preparation for the upcoming practices.

Cloud Shell

Oracle Cloud Infrastructure (OCI) Cloud Shell is a web browser–based terminal accessible from the Oracle Cloud Console. Cloud Shell is free to use (within monthly tenancy limits), and provides access to a Linux shell with a pre-authenticated OCI CLI and other useful tools.

It provides:

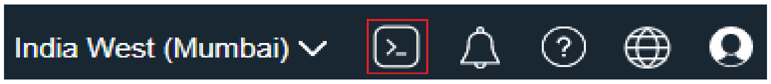
- An ephemeral machine to use as a host for a Linux shell, preconfigured with the latest version of the OCI CLI and several useful tools
- 5 GB of storage for your home directory
- A persistent frame of the Console which stays active as you navigate to different pages of the Console

Note:

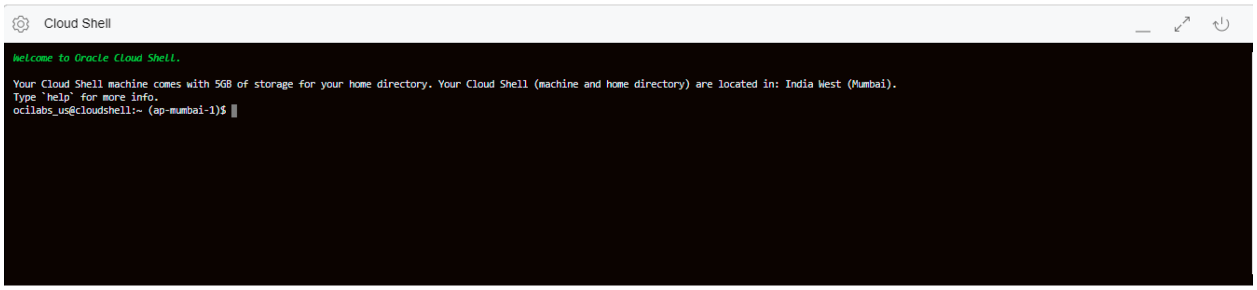
- The OCI CLI will execute commands against the region selected in the Console's Region selection menu when the Cloud Shell is started. Changing the region selection in the Console will not change the region for existing Cloud Shell instances; you will need to open a new Cloud Shell instance to change regions.
- Cloud Shell sessions have a maximum length of 24 hours, and time out after 20 minutes of inactivity. However, this should not impact this practice.

Tasks

1. Log in to your Oracle Cloud Account.
2. Log in as administrator.
3. Click the **Cloud Shell** icon in the OCI Console header, highlighted in the following screenshot:



4. This will launch the Cloud Shell in a “drawer” at the bottom of the Console. When it is ready, you will see the terminal.



```
Welcome to Oracle Cloud Shell.

Your Cloud Shell machine comes with 5GB of storage for your home directory. Your Cloud Shell (machine and home directory) are located in: India West (Mumbai).
Type `help` for more info.
ocilabs_us@cloudshell:~ (ap-mumbai-1)$
```

5. You can use the icons in the upper-right corner of the Cloud Shell window to minimize, maximize, and close your Cloud Shell session.

   You can also use the menu icon in the upper-left corner of the Cloud Shell window to upload or download files, restart the Console, and use different setting options.

6. For clipboard operations:
   - Windows users can use `Ctrl-C` or `Ctrl-Insert` to copy, and `Ctrl-V` or `Shift-Insert` to paste.
   - For Mac OS users, use `Cmd-C` to copy and `Cmd-V` to paste.

7. To get started with Cloud Shell, you can run the following OCI CLI command. Your Cloud Shell comes with the OCI CLI pre-authenticated, so there is no setup to perform before you can start using it.

   This command will display the namespace of your OCI Tenant.

```
$ oci os ns get
{
  "data": "bm6rwnfgnfbj"
}
```

This completes the task of launching Cloud Shell. Keep this session active for the next practice.

Practice: Generating an ssh Key

Overview

Instances use an ssh key pair instead of a password to authenticate a remote user. A key pair file contains a private key and a public key. You keep the private key on your computer and provide the public key every time you launch an instance. In this practice, you will generate an ssh key to be used later while launching an instance.

Tasks

1. Launch the Cloud Shell session as described in the previous practice.
2. Execute the following commands to generate an ssh-key, which will be used to create a Compute instance. As long as an **`id_rsa`** and **`id_rsa.pub`** keypair is present, they can be reused. By default, these are stored in **`~/.ssh/`** directly.

```
$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key
(/home/ocilabs_us/.ssh/id_rsa):
Created directory '/home/ocilabs_us/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in
/home/ocilabs_us/.ssh/id_rsa.
Your public key has been saved in
/home/ocilabs_us/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:zxQogNbEvQJIAbJb+3x1r8QEJN8ZWlUYHHyZge5lZ10
ocilabs_us@804708cc8ef6
The key's randomart image is:
+---[RSA 2048]----+
|=+.=o.. . o++*o+ |
|o.+ o..+ = o= + E|
|....  ..= +. .  o|
| o .. .. . .. o +|
|. .  .  S +. o o |
|   o   . B ..    |
|    o .   = .    |
|     .   . .     |
|          .      |
+----[SHA256]-----+
```

3. Make sure permissions are restricted because sometimes ssh fails if private keys have permissive file permissions.

```
$ chmod 0700 ~/.ssh
$ chmod 0600 ~/.ssh/id_rsa
$ chmod 0644 ~/.ssh/id_rsa.pub
$ ls -l ~/.ssh
total 8
-rw-------. 1 ocilabs_us oci 1675 Jun 29 17:10 id_rsa
-rw-r--r--. 1 ocilabs_us oci  405 Jun 29 17:10 id_rsa.pub
```

4. Copy the contents of `~/.ssh/id_rsa.pub` to Notepad. This is your ssh key to connect to the instances.

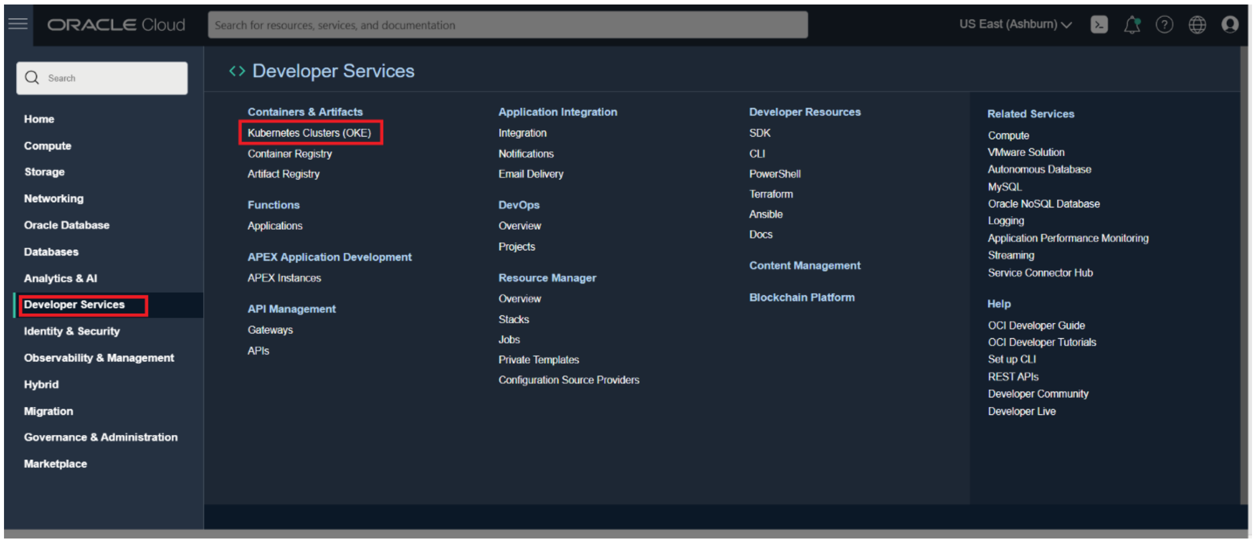
```
$ cat ~/.ssh/id_rsa.pub
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDEUsgq5R/5PcdS1+Mws2Y6vli0HcCw9g3l
uI0x/yFDwE+stlnfyzv4c73+uS35VD6kgFMo5izZKx3fV0JpqhUPjwtwuyigP9jc
6cgJmWjYhkbCHD8r8bFvrdVv0KuUPi+oKQUI4Zr4EtuTao3kkLywWz6aEJgS6GY2
19JSXqBH27QjgGk4l4sdeb9VuTuQ07Z7VyzyAUfKK5oqlJfLC6a/JhdfTLYnv++W
y3lnVZUojEQK57bOD7jVDTTErs0PSWXzMedretrEXtsBU+Tm1DZBe7QWoqghMTkI
a3hegu1qIwVxujfy7xDNPE1FHR/LG0978CyJwAfRShjXAYQtSwMF
ocilabs_us@804708cc8ef6
```

This completes the task of creating the ssh key.

Practice: Creating a Kubernetes Cluster

Tasks

1. From the OCI Services menu, click **Kubernetes Clusters (OKE)** under Developer Services.



2. Under **List Scope**, select the compartment in which you would like to create a cluster.
3. Click **Create Cluster**. Select **Quick Create** and click **Launch Workflow**.

   NAME: Provide a name (oke-cluster in this example)

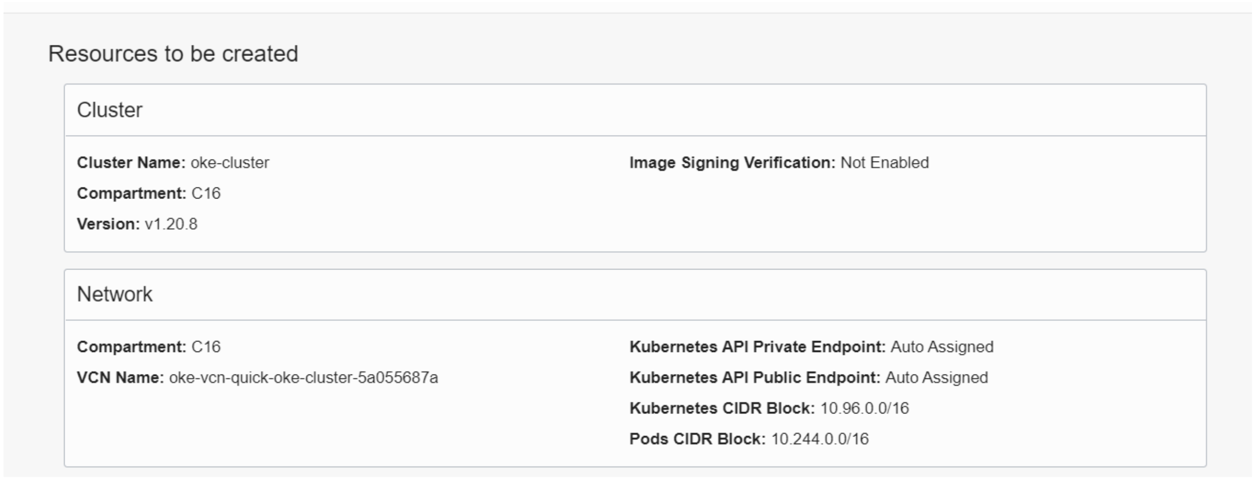
   COMPARTMENT: Choose your compartment

   Kubernetes Endpoint: Public Endpoint

   Kubernetes Worker Nodes: Public Workers

   SHAPE: Choose a VM shape

   NUMBER OF NODES: 1



Resources to be created

Cluster

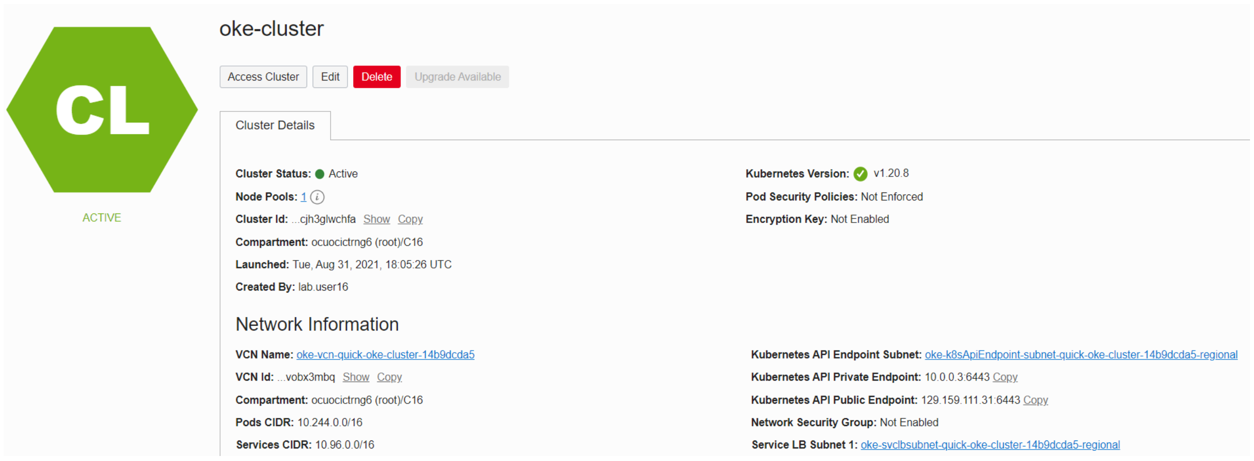
Cluster Name: oke-cluster
Compartment: C16
Version: v1.20.8
Image Signing Verification: Not Enabled

Network

Compartment: C16
VCN Name: oke-vcn-quick-oke-cluster-5a055687a
Kubernetes API Private Endpoint: Auto Assigned
Kubernetes API Public Endpoint: Auto Assigned
Kubernetes CIDR Block: 10.96.0.0/16
Pods CIDR Block: 10.244.0.0/16

4. Click **Next** and **Create Cluster**.

   You now have an OKE cluster with one node and a Virtual Cloud Network with all the necessary resources and configuration needed.



oke-cluster

Cluster Status: ● Active
Node Pools: 1
Cluster Id: ...cjh3glwchfa Show Copy
Compartment: ocuocictrng6 (root)/C16
Launched: Tue, Aug 31, 2021, 18:05:26 UTC
Created By: lab.user16

Kubernetes Version: ✓ v1.20.8
Pod Security Policies: Not Enforced
Encryption Key: Not Enabled

Network Information

VCN Name: oke-vcn-quick-oke-cluster-14b9dcda5
VCN Id: ...vobx3mbq Show Copy
Compartment: ocuocictrng6 (root)/C16
Pods CIDR: 10.244.0.0/16
Services CIDR: 10.96.0.0/16

Kubernetes API Endpoint Subnet: oke-k8sApiEndpoint-subnet-quick-oke-cluster-14b9dcda5-regional
Kubernetes API Private Endpoint: 10.0.0.3:6443 Copy
Kubernetes API Public Endpoint: 129.159.111.31:6443 Copy
Network Security Group: Not Enabled
Service LB Subnet 1: oke-svclbsubnet-quick-oke-cluster-14b9dcda5-regional

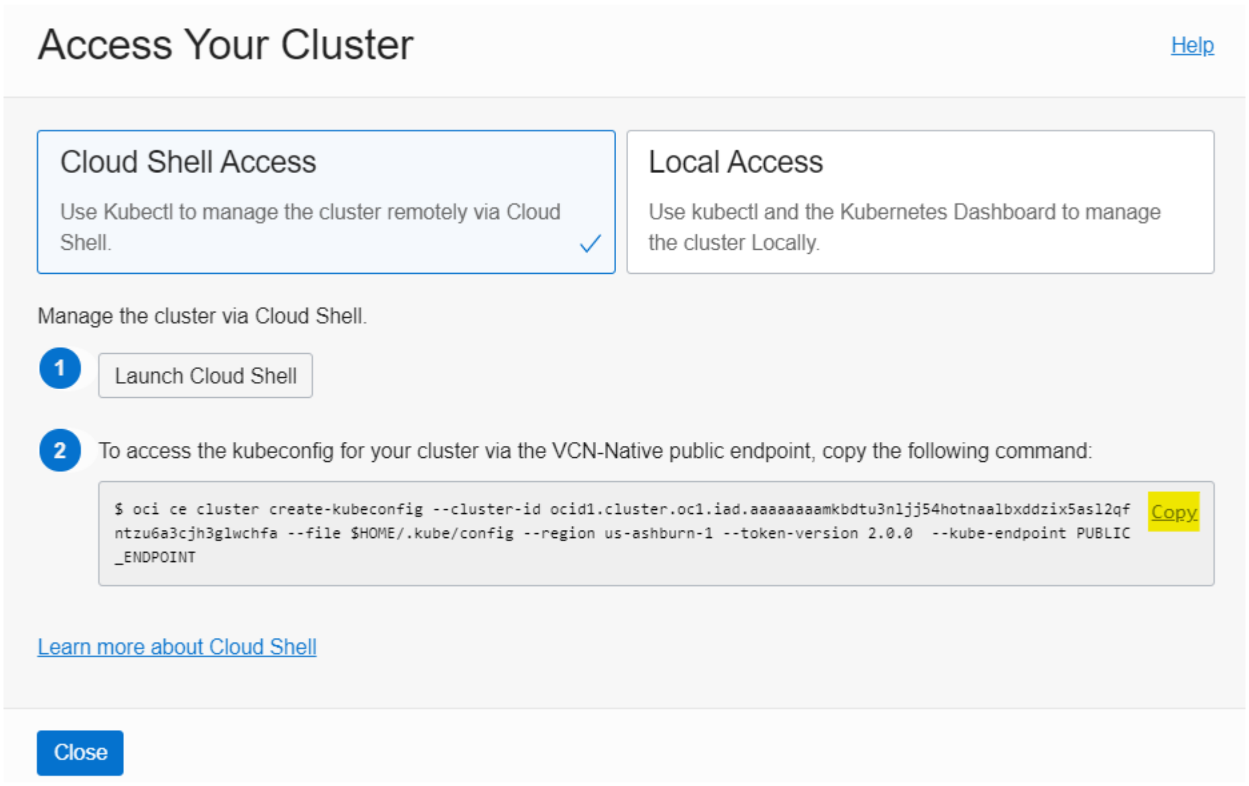
Practice: Deploying a Sample Nginx App on the Cluster Using kubectl

Overview

To access a cluster using kubectl, you have to set up a Kubernetes configuration file (commonly known as a “kubeconfig” file) for the cluster. The kubeconfig file (by default named `config` and stored in the `$HOME/.kube` directory) provides the necessary details to access the cluster. Having set up the kubeconfig file, you can start using kubectl to manage the cluster.

Tasks

1. In the Console, open the navigation menu and click **Developer Services**. Under **Containers**, click **Kubernetes Clusters (OKE)**.
2. Select a **Compartment** you have permission to work in.
3. On the **Cluster List** page, click the name of the cluster you want to access using kubectl. The **Cluster** page shows details of the cluster.
4. Click the **Access Cluster** button to display the **Access Your Cluster** dialog box.
5. Click **Cloud Shell Access**.
6. Click **Launch Cloud Shell** to display the Cloud Shell window.


Access Your Cluster
Help
Cloud Shell Access
Use Kubectl to manage the cluster remotely via Cloud Shell.
Local Access
Use kubectl and the Kubernetes Dashboard to manage the cluster Locally.
Launch Cloud Shell
Copy
Learn more about Cloud Shell
Close

Manage the cluster via Cloud Shell.

1. Launch Cloud Shell

2. To access the kubeconfig for your cluster via the VCN-Native public endpoint, copy the following command:

```
$ oci ce cluster create-kubeconfig --cluster-id ocid1.cluster.oc1.iad.aaaaaaaamkbdtu3nljj54hotnaalbxddzix5asl2qf
ntzu6a3cjh3glwchfa --file $HOME/.kube/config --region us-ashburn-1 --token-version 2.0.0  --kube-endpoint PUBLIC
_ENDPOINT
```

7. Run the Oracle Cloud Infrastructure CLI command to set up the kubeconfig file and save it in a location accessible to kubectl. Paste the command copied from Step 6 into Cloud Shell.

```
$ oci ce cluster create-kubeconfig --cluster-id
ocid1.cluster.oc1.iad.aaaaaaaamkbdtu3nljj54hotnaalbxddzix5asl2qf
ntzu6a3cjh3glwchfa --file $HOME/.kube/config --region us-
ashburn-1 --token-version 2.0.0  --kube-endpoint PUBLIC_ENDPOINT
New config written to the Kubeconfig file
/home/lab_user16/.kube/config
```

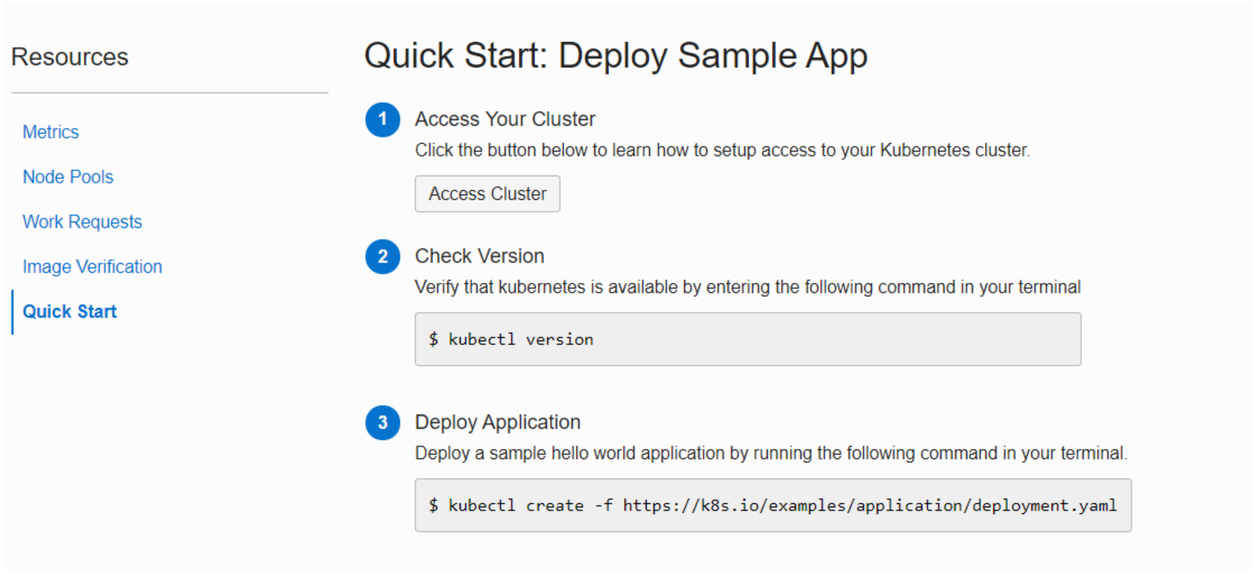
8. Set the value of the KUBECONFIG environment variable to point to the name and location of the kubeconfig file.

```
$ export KUBECONFIG=$HOME/.kube/config
```

9. Verify that kubectl can connect to the cluster.

```
$ kubectl get nodes
NAME          STATUS    ROLES    AGE      VERSION
10.0.10.19    Ready     node     6m42s    v1.20.8
```

10. Switch to the OCI Console window and navigate to your cluster. In the Cluster detail window, scroll down and click **Quick Start**, under **Resources**. Follow the steps under the **Quick Start** section.



Quick Start: Deploy Sample App

1 Access Your Cluster

Click the button below to learn how to setup access to your Kubernetes cluster.

2 Check Version

Verify that kubernetes is available by entering the following command in your terminal

```
$ kubectl version
```

3 Deploy Application

Deploy a sample hello world application by running the following command in your terminal.

```
$ kubectl create -f https://k8s.io/examples/application/deployment.yaml
```

11. Follow the steps in Quick Start.

```
$ kubectl version
Client Version: version.Info{Major:"1", Minor:"19",
GitVersion:"v1.19.12",
GitCommit:"ea17fb43b3d97f55c2170b2dbaff407831dcc78e",
GitTreeState:"clean", BuildDate:"2021-07-23T10:21:50Z",
GoVersion:"go1.15.13 BoringCrypto", Compiler:"gc",
Platform:"linux/amd64"}
Server Version: version.Info{Major:"1", Minor:"20",
GitVersion:"v1.20.8",
GitCommit:"50317190d44dbdb51ae7ff430917b32ba96188b5",
GitTreeState:"clean", BuildDate:"2021-06-30T14:20:31Z",
GoVersion:"go1.15.13 BoringCrypto", Compiler:"gc",
Platform:"linux/amd64"}
$ kubectl create -f
https://k8s.io/examples/application/deployment.yaml
deployment.apps/nginx-deployment created
```

12. Get Deployment and Pods data.

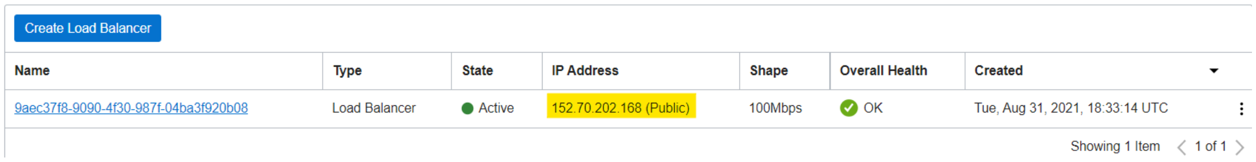
```
$ kubectl get deployments
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment    2/2     2            2           2m24s
$ kubectl get pods -o wide
NAME                                READY   STATUS    RESTARTS
AGE     IP            NODE          NOMINATED NODE   READINESS
GATES
nginx-deployment-66b6c48dd5-qg422   1/1     Running   0
2m44s   10.244.0.4    10.0.10.19    <none>           <none>
nginx-deployment-66b6c48dd5-zsb52   1/1     Running   0
2m44s   10.244.0.5    10.0.10.19    <none>           <none>
```

13. Create a service to expose the application. The cluster is integrated with OCI Cloud Controller Manager (CCM). As a result, creating a service of type --type=LoadBalancer will expose the pods to the Internet using an OCI Load Balancer.

```
$ kubectl expose deployment nginx-deployment --port=80 --
type=LoadBalancer
service/nginx-deployment exposed
```

14. Switch to the OCI Console window. From the OCI Services menu, click **Networking** and then **Load Balancers**. A new OCI LB should be getting provisioned.

15. Note down its Public IP address.



Name	Type	State	IP Address	Shape	Overall Health	Created
9aec37f8-9090-4f30-987f-04ba3f920b08	Load Balancer	● Active	152.70.202.168 (Public)	100Mbps	✓ OK	Tue, Aug 31, 2021, 18:33:14 UTC

16. Open a new browser tab and enter the URL http://`<Load-Balancer-Public-IP>` .The Nginx welcome screen should be displayed.

This completes the task of deploying a sample app using kubectl.

Terminate the Load Balancer, OKE Cluster, and VCN that are created as part of the practice.