

The background features abstract, textured shapes in shades of brown, grey, and red. There are also several small, horizontal orange bars scattered across the slide.

ORACLE

Oracle Container Engine for Kubernetes

Level 200

Jamal Arif

Oracle Cloud Infrastructure

October, 2019

Safe harbor statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions.

The development, release, timing, and pricing of any features or functionality described for Oracle's products may change and remains at the sole discretion of Oracle Corporation.

Objectives

After completing this lesson, you should be able to understand:

- OKE Custom Create
- Bastion Host for Private Clusters
- Using OCI LB with OKE
- Persistent Storage for OKE
 - Using OCI Block Volume
 - Using OCI File System Storage

Pre-requisites: Container Engine (OKE) 100 Level



OKE 'Custom Create' in OCI

Pre-requisites for creating a K8s Cluster

- Monthly universal Credits have limit of 3 clusters per OCI region with 1000 nodes in a cluster and Pay-as-you-go or Promo accounts have a limit for One Cluster (by default).
- Must also have compute Instance Quota (Required) – to launch k8s worker nodes in an AD or across ADs for HA
- Required Policy in the root compartment of your tenancy
allow service OKE to manage all-resources in tenancy
- To launch a K8s cluster, user must be either part of the Admin group or a group to which a policy grants the appropriate Container Engine for Kubernetes permissions.
- Policies can be created for users which are not part of the admin group
- For Example: To enable users in group 'dev-team' to perform any operation on cluster-related resources →
allow group dev-team to manage cluster-family in tenancy

Note: Policies must also grant the group 'dev-team' Networking permissions of VCN_READ and VCN_CREATE, SUBNET_READ and SUBNET_CREATE, COMPARTMENT_INSPECT, INTERNET_GATEWAY_CREATE, NAT_GATEWAY_CREATE, ROUTE_TABLE_UPDATE, SECURITY_LIST_CREATE: Details [here](#)

Custom Create Pre-reqs: Basic Virtual Cloud Network Config

- An Existing VCN with following
 - Internet Gateway
 - Nat Gateway (Private Worker Nodes)
 - Route table –
 - with default route to IGW for Load balancer subnets and public worker node subnets
 - with default route to NAT GW for private worker node subnets
 - K8s worker node subnets –
 - three public AD specific subnets in different ADs or a regional subnet for *High Availability* of public workers
 - three private AD specific subnets in different ADs or a regional subnet for *High Availability* of private workers
 - LBs Subnets – A regional public subnet for OCI public load balancer
 - Separate Security Lists for K8s worker nodes subnets and LB subnets

Custom Create Pre-reqs: Basic Virtual Cloud Network Config

Security Lists for K8s worker nodes subnets

- Stateless ingress and egress rules that allow all traffic between the different worker node subnets
- stateless ingress and egress rules that allow all traffic between worker node subnets and load balancer subnets
- an egress rule that allows all outbound traffic to the internet
- *For PUBLIC worker nodes only* - ingress rules to allow Container Engine for Kubernetes to access worker nodes on port 22 from CIDR blocks mentioned [here](#).

OKE Custom Create

Step 1: Navigate to Menu → Developer Services → Container Clusters (OKE) → Create Cluster

Developer Services

Container Clusters (OKE)

Marketplace

Registry (OCIR)

ORACLE Cloud

us-ashburn-1

Containers

Clusters

Registry

List Scope

COMPARTMENT

tutorials

jamalarif (root)/tutorials

Clusters *in tutorials Compartment*

Clusters Requirements:

NOTE: In order to use all features of this service, you must have the following minimum required privileges:

- List, Get, and Create VCNs
- List, Get, and Create Subnets
- List Availability Domains
- Create Internet Gateways
- Create NAT Gateways
- Update Route Tables
- Create Security Lists

Your tenancy must also have the following required policy statement defined in the *root compartment* of your tenancy [here](#) by a user with administrative privileges:

- allow service OKE to manage all-resources in tenancy

Without the above privileges and policies, various errors will be presented and the cluster service will not function correctly.

[Learn more here](#)

Create Cluster

Name	Status	Node Pools	VCN	Version	Created	
No clusters exist. Create one to get started.						
Showing 0 Item(s)						



OKE Custom Create

Step 2: Cluster Creation

Name of the Cluster

The version of Kubernetes to run on the master nodes and worker nodes of the cluster. Either accept the default version or select a version of your choice. Amongst other things, the Kubernetes version you select determines the default set of admission controllers that are turned on in the created cluster (the set follows the recommendation given in the [Kubernetes documentation](#) for that version).

Cluster Creation [help](#) [close](#)

CLUSTER COMPARTMENT
Kubernetes

NAME
oke_custom_create

KUBERNETES VERSION
v1.12.7
Kubernetes version installed on your master nodes

☐ **QUICK CREATE**
Quickly create a cluster with default settings, also creates a dedicated network

☒ **CUSTOM CREATE**
Create a cluster with custom settings, assumes an existing network

OKE Custom Create (contd.)

Step 2: Cluster Creation

- Provide VCN you have created/already available for OKE.
- Provide LB subnets (public subnets)
- Service CIDR blocks and PODs CIDR block are optional and chosen by default.

- Specify whether to encrypt Kubernetes secrets at rest in the etcd key-value store for the cluster using the Key Management service.

Network Selection

NETWORK COMPARTMENT

Kubernetes

jamalarif (root)/Kubernetes

VCN

OKE-VCN

An existing VCN to provision your cluster in. A /16 CIDR is sufficient for the majority of cases. An Internet Gateway is required as well as a default route to the gateway.

KUBERNETES SERVICE LB SUBNETS ⓘ

LB_AD2 x LB_AD1 x

If automatic Load Balancer integration is desired, two subnets must be provided to host Load Balancers created by Kubernetes in your tenancy. These subnets should be different from those subnets used by node pools that you create for this cluster (cluster nodes).

KUBERNETES SERVICE CIDR BLOCK OPTIONAL

Defaults to 10.96.0.0/16

This is the CIDR range used by exposed Kubernetes services (ClusterIPs). This CIDR should not overlap with the VCN CIDR range.

PODS CIDR BLOCK OPTIONAL

Defaults to 10.244.0.0/16

This is the CIDR range used for IP addresses by your pods. A /16 CIDR is generally sufficient. This CIDR should not overlap with any subnet range in the VCN (it can also be outside the VCN CIDR range).

Encryption

- ☒ **NO ENCRYPTION**
Uses standard block storage encryption
- ☐ **ENCRYPT USING CUSTOMER-MANAGED KEYS**
Requires access to a valid Key Management System. [Learn more](#)

OKE Custom Create (contd.)

Step 2: Cluster Creation

Version: The version of Kubernetes to run on each worker node in the node pool. By default, the version of Kubernetes specified for the master nodes is selected.

Image: The image to use on each node in the node pool. An image is a template of a virtual hard drive that determines the operating system and other software for the node.

Shape: The compute shape to use for each node in the node pool.

Quantity per Subnet: The number of worker nodes to create for the node pool in each private subnet.

Node Pool

NAME

custom_pool1

VERSION

v1.12.7

The version of Kubernetes installed on all nodes in the node pool.

IMAGE

Oracle-Linux-7.4

The OS/image installed on all nodes in the node pool.

SHAPE

VM.Standard1.4

The shape for all nodes in the node pool.

SUBNETS ⓘ

worker_AD_3 (Private) × worker_AD_2 (Private) × worker_AD1 (Private) ×

The subnets used for node instances in the node pool. These subnets should be different from the Cluster Kubernetes Service LB subnets. A subnet per Availability Domain (AD) is typical.

QUANTITY PER SUBNET

1

The number of nodes per subnet.

OKE Custom Create (contd.)

Step 2: Cluster Creation

Public SSH Key: (Optional) The public key is installed on all worker nodes in the cluster, and you can use this key to access the worker nodes (Connect via Bastion Host since worker nodes are in Private subnets)

Kubernetes Labels: One or more labels (in addition to a default label) to add to worker nodes in the node pool to enable the targeting of workloads at specific node pools.

PUBLIC SSH KEY

OPTIONAL

A
O
n
p
z

In order to access your private nodes with a public SSH key you will need to set up a bastion host (a.k.a. jump box). [Learn more about setting up a bastion host](#)

Kubernetes Labels

KEY

VALUE

name

custom_pool1

nodes

private

Nodes added to this node pool will automatically get one or more Kubernetes labels applied, enabling users to target Kubernetes workloads in a specific pool

+ Another Pair



OKE Custom Create (contd.)

Step 2: Cluster Creation

Cluster Creation - Review

[help](#) [close](#)

Existing resources to be used

Network

COMPARTMENT: Kubernetes

VCN: OKE-VCN

SERVICE LB SUBNETS: LB_AD2, LB_AD1

KUBERNETES CIDR BLOCK: 10.96.0.0/16

PODS CIDR BLOCK: 10.244.0.0/16

[edit](#)

Resources to be created

Cluster

NAME: cluster2

COMPARTMENT: Kubernetes

VERSION: v1.12.7

ADD ONS: Kubernetes Dashboard, Tiller

[edit](#)

Node Pool

NAME: custom_pool1

COMPARTMENT: Kubernetes

VERSION: v1.12.7

IMAGE: Oracle-Linux-7.4

SHAPE: VM.Standard1.4

NUMBER OF NODES: 1

SUBNETS: worker_AD_2 (Private), worker_AD1 (Private), worker_AD_3 (Private)

[edit](#)

☒ VIEW DETAIL PAGE AFTER THIS CLUSTER IS REQUESTED

Create

Back

Cancel

cluster2	 Creating	1	OKE-VCN	v1.12.7	7/8/2019
--------------------------	--	---	---------	---------	----------





Connecting to Private Worker Nodes via Bastion Host

How to connect to private worker nodes via a bastion host

- Create a regional public subnet in OKE virtual cloud network
- Create a Bastion Host in the regional subnet (Oracle Linux)
- Verify bastion host subnet has security list rules to connect to worker nodes subnet
- Use ssh proxy command to connect to the worker nodes
- ```
$ ssh -t -o ProxyCommand='ssh -i /pathtosshprivatekey/ opc@<Bastion Host public IP> -W %h:%p %r' -i /pathtosshprivatekey/ opc@<private instance IP>
```
- This command allows us to “tunnel” through the bastion host to our private instance, while maintaining SSH keys locally on our laptop. Storing private SSH keys on a public server such as a Bastion host is not recommended.



# Running a basic app on OKE & exposing it externally using an OCI Load Balancer



# Connecting to Cluster using Kubectl

Step 1: Navigate to Menu → Developer Services → Container Clusters (OKE) → 'Your Cluster'

## Resources

Node Pools

Work Requests

Getting Started

## Quick Start: Deploy Sample App

### 1 Access Kubeconfig File

To get started, learn how to download the `kubeconfig` file for this cluster by clicking below. This file will contain a series of authentication mechanisms and cluster connection information.

Access Kubeconfig

### How to Access Kubeconfig

[help](#) [close](#)

You must have [downloaded and installed](#) the OCI CLI and [configured](#) it for use.

To access the kubeconfig for your cluster, run the following commands:

```
1. mkdir -p $HOME/.kube
2. oci ce cluster create-kubeconfig --cluster-id
 ocid1.cluster.oc1.iad.aaaaaaaaztcnzuga2tsmrwgq3gczbhe4tcjmvmu4gemrrhcziyug4zt --file
 $HOME/.kube/config --region us-ashburn-1
```

To set your KUBECONFIG environment variable to the file for this cluster, use:

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

If you wish to save your new kubeconfig to a different location, please change the `--file` argument in the CLI command above with the new location path. You may also have to set or update your KUBECONFIG environment variable with this new location path. To persist the environment variable, your shell initiation script may have to be updated as well.

For more information on managing kubeconfig files, please refer to the official [Kubernetes documentation](#). More information on the available commands for OCI's Container Engine for Kubernetes CLI can be found [here](#).

Close

```
sararif-mac:~ sararif$ kubectl cluster-info
Kubernetes master is running at https://czdiyzug4zt.us-ashburn-1.clusters.oci.o
raclecloud.com:6443
KubeDNS is running at https://czdiyzug4zt.us-ashburn-1.clusters.oci.oraclecloud
.com:6443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy
```

# Running a basic nginx on OKE

- Create a yaml file 'nginx.yaml' with deployment details of a basic nginx app

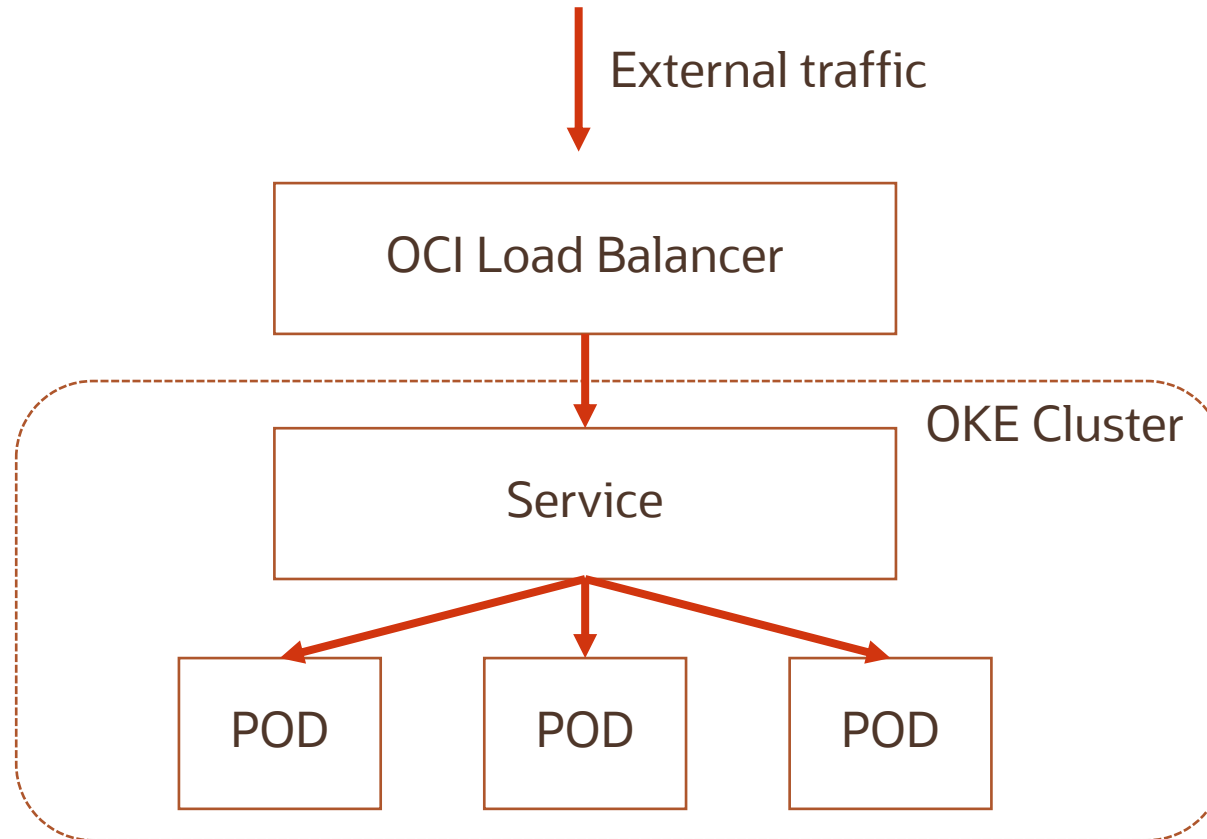
```
1 apiVersion: apps/v1beta1
2 kind: Deployment
3 metadata:
4 name: nginx-deployment
5 spec:
6 replicas: 2 # tells deployment to run 2 pods matching the template
7 template: # create pods using pod definition in this template
8 metadata:
9 # unlike pod-nginx.yaml, the name is not included in the meta data as a unique name is
10 # generated from the deployment name
11 labels:
12 app: nginx
13 spec:
14 containers:
15 - name: nginx
16 image: latest
17 ports:
18 - containerPort: 80
```

- `$kubectl apply -f nginx.yaml`
- It creates a deployment with 2 replica pods each running an nginx container with the image specified in the yaml file

```
[sararif-mac:wordpress sararif$ kubectl get deployments
NAME DESIRED CURRENT UP-TO-DATE AVAILABLE AGE
nginx-deployment 2 2 2 2 110s
[sararif-mac:wordpress sararif$ kubectl get pods
NAME READY STATUS RESTARTS AGE
nginx-deployment-5c689d88bb-wll8b 1/1 Running 0 114s
nginx-deployment-5c689d88bb-xxcm2 1/1 Running 0 114s
[sararif-mac:wordpress sararif$
```

# Using OCI Load Balancers to Load Balance traffic

- The nginx app created in last step is running on OKE cluster, however we are unable to access the app from outside the OKE cluster.
- Kubernetes 'service' can be used to expose the app. While exposing the service, we can use the type 'LoadBalancer' to make use of OCI Load Balancer to load balance the traffic across the two pods.



# Using OCI Load Balancers to Load Balance traffic

- Create a service and expose it via type LoadBalancer
  - `$kubectl expose deployment nginx-deployment --type=LoadBalancer --port=80`
- A service is created and an OCI Load Balancer is launched. External IP is OCI LB listener Public IP.
- Security lists are automatically edited to allow traffic connectivity from LB subnet to worker node subnet.

```
sararif-mac:wordpress sararif$ kubectl expose deployment nginx-deployment --type=LoadBalancer --port=80
service/nginx-deployment exposed
[sararif-mac:wordpress sararif$ kubectl get services
```

| NAME             | TYPE         | CLUSTER-IP   | EXTERNAL-IP   | PORT(S)      | AGE   |
|------------------|--------------|--------------|---------------|--------------|-------|
| kubernetes       | ClusterIP    | 10.96.0.1    | <none>        | 443/TCP      | 3h36m |
| nginx-deployment | LoadBalancer | 10.96.241.71 | 129.213.74.72 | 80:30680/TCP | 92s   |

```
[sararif-mac:wordpress sararif$
```

129.213.74.72

## Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to [nginx.org](https://nginx.org).  
Commercial support is available at [nginx.com](https://nginx.com).

*Thank you for using nginx.*



# Security lists are added...

## Load Balancers *in* Kubernetes *Compartment*

| <a href="#">Create Load Balancer</a>                 |          |                                                     |                        |                |                               |   |
|------------------------------------------------------|----------|-----------------------------------------------------|------------------------|----------------|-------------------------------|---|
| Name                                                 | State    | OCID                                                | IP Address             | Overall Health | Created                       |   |
| <a href="#">d2886e68-a1df-11e9-b50e-0a580aed4e8f</a> | ● Active | ...4vtmyq <a href="#">Show</a> <a href="#">Copy</a> | 129.213.74.72 (Public) | ? Unknown      | Tue, 09 Jul 2019 00:23:49 GMT | ⋮ |
| Showing 1 Item(s) < Page 1 >                         |          |                                                     |                        |                |                               |   |

|                          |    |                |     |     |       |                              |   |
|--------------------------|----|----------------|-----|-----|-------|------------------------------|---|
| <input type="checkbox"/> | No | 172.16.20.0/24 | TCP | All | 30680 | TCP traffic for ports: 30680 | ⋮ |
| <input type="checkbox"/> | No | 172.16.21.0/24 | TCP | All | 30680 | TCP traffic for ports: 30680 | ⋮ |
| <input type="checkbox"/> | No | 172.16.10.0/24 | TCP | All | 30680 | TCP traffic for ports: 30680 | ⋮ |
| <input type="checkbox"/> | No | 172.16.11.0/24 | TCP | All | 30680 | TCP traffic for ports: 30680 | ⋮ |
| <input type="checkbox"/> | No | 172.16.12.0/24 | TCP | All | 30680 | TCP traffic for ports: 30680 | ⋮ |

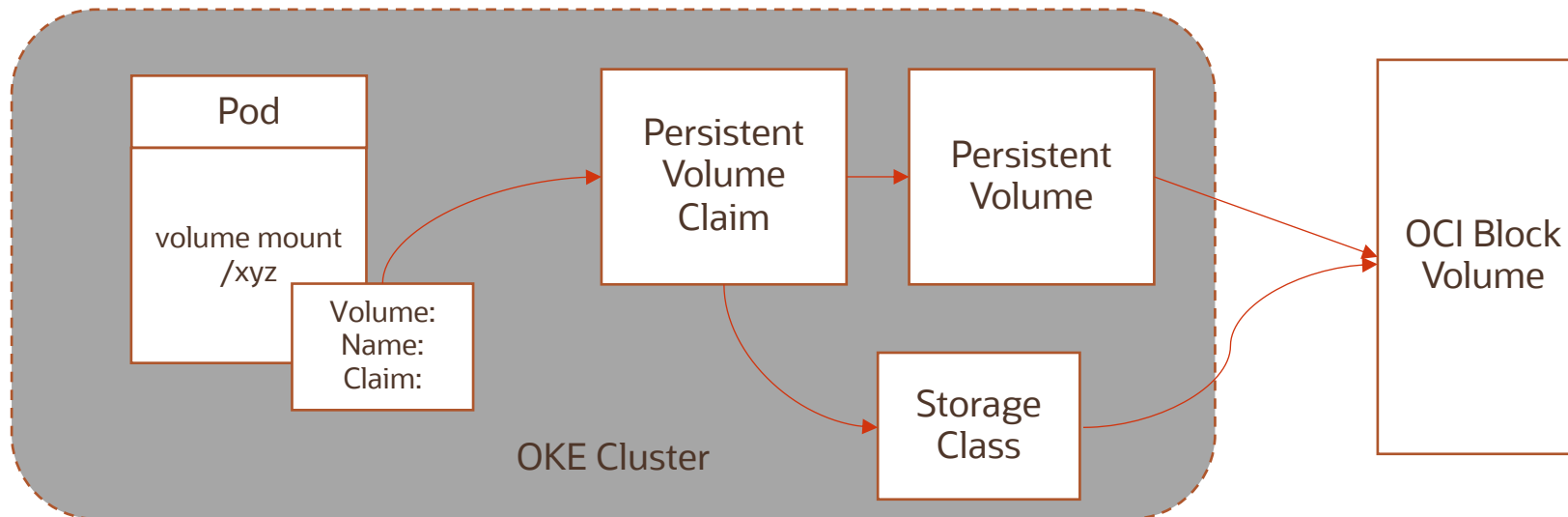




# Using OCI Block Volume as Persistent Volume Claim for Kubernetes Pods

# Why Persistent storage for containers?..

- Container storage via a container's root file system is ephemeral, and can disappear upon container deletion and creation.
- To provide a durable location to store data and prevent it from being lost, you can create and use persistent volumes to store data outside of containers
- You can define and apply a persistent volume claim to your cluster, which in turn creates a persistent volume that's bound to the claim.
- A claim is a block storage volume in the underlying IaaS provider (OCI BV in current scenario)



# Creating a PVC..

- Following yaml defines a PVC that requests 50GB of persistent storage.

```
1 apiVersion: v1
2 kind: PersistentVolumeClaim
3 metadata:
4 name: mysqlclaim
5 spec:
6 storageClassName: "oci"
7 selector:
8 matchLabels:
9 failure-domain.beta.kubernetes.io/zone: "US-ASHBURN-AD-1"
10 accessModes:
11 - ReadWriteOnce
12 resources:
13 requests:
14 storage: 50Gi
15
```

- `$ kubectl create -f`  
[https://raw.githubusercontent.com/jamalarif/oci/master/oke/wordpress/mysql\\_pvc\\_claim.yml](https://raw.githubusercontent.com/jamalarif/oci/master/oke/wordpress/mysql_pvc_claim.yml)

```
[sararif-mac:wordpressexample sararif$ kubectl get pvc
NAME STATUS VOLUME CAPACITY ACCESS MODES STORAGECLASS AGE
mysqlclaim Bound ocid1.volume.oc1.iad.abuwcljrjslqj4mv3o3ow4gkqf53wpkru7n53zaf1ju5kdiuyd 50Gi RWO oci 31s
wiyg5cir7a Bound ocid1.volume.oc1.iad.abuwcljrjslqj4mv3o3ow4gkqf53wpkru7n53zaf1ju5kdiuyd 50Gi RWO oci 31s
sararif-mac:wordpressexample sararif$
```



# Using PVC with K8s deployments..

- Use that created PVC while creating a new deployment. For instance -
- `$ kubectl create -f https://raw.githubusercontent.com/jamalarif/oci/master/oke/wordpress/mysql\_deployment.yml`

```
sararif-mac:wordpressexample sararif$ kubectl create -f https://raw.githubusercontent.com/jamalarif/oci/master/oke/wordpress/mysql_deployment.yml
deployment.extensions/mysql created
sararif-mac:wordpressexample sararif$ kubectl get deployments
```

| NAME  | DESIRED | CURRENT | UP-TO-DATE | AVAILABLE | AGE  |
|-------|---------|---------|------------|-----------|------|
| mysql | 1       | 1       | 1          | 1         | 112s |

Oracle Cloud console screenshot showing instance details for 'oke-czdiyug4zt-nrwmyrrgqzd-st6r6otoeua-0'. The instance is in the 'RUNNING' state. The console displays various information including Instance Information, Primary VNIC Information, Launch Options, and Attached Block Volumes. The Attached Block Volumes section shows a single volume named 'mysqlclaim' with a size of 50.0 GB, attached to the instance.

```
1 #MySQL Deployment
2 apiVersion: extensions/v1beta1
3 kind: Deployment
4 metadata:
5 name: mysql
6 labels:
7 app: mysql
8 spec:
9 replicas: 1
10 selector:
11 matchLabels:
12 app: mysql
13 template:
14 metadata:
15 labels:
16 app: mysql
17 spec:
18 containers:
19 - image: mysql:5.6
20 name: mysql
21 env:
22 - name: MYSQL_ROOT_PASSWORD
23 valueFrom:
24 secretKeyRef:
25 name: mysql
26 key: password
27 ports:
28 - containerPort: 3306
29 name: mysql
30 volumeMounts:
31 - name: mysql-persistent-storage
32 mountPath: /var/lib/mysql
33 volumes:
34 - name: mysql-persistent-storage
35 persistentVolumeClaim:
36 claimName: mysqlclaim
37
```

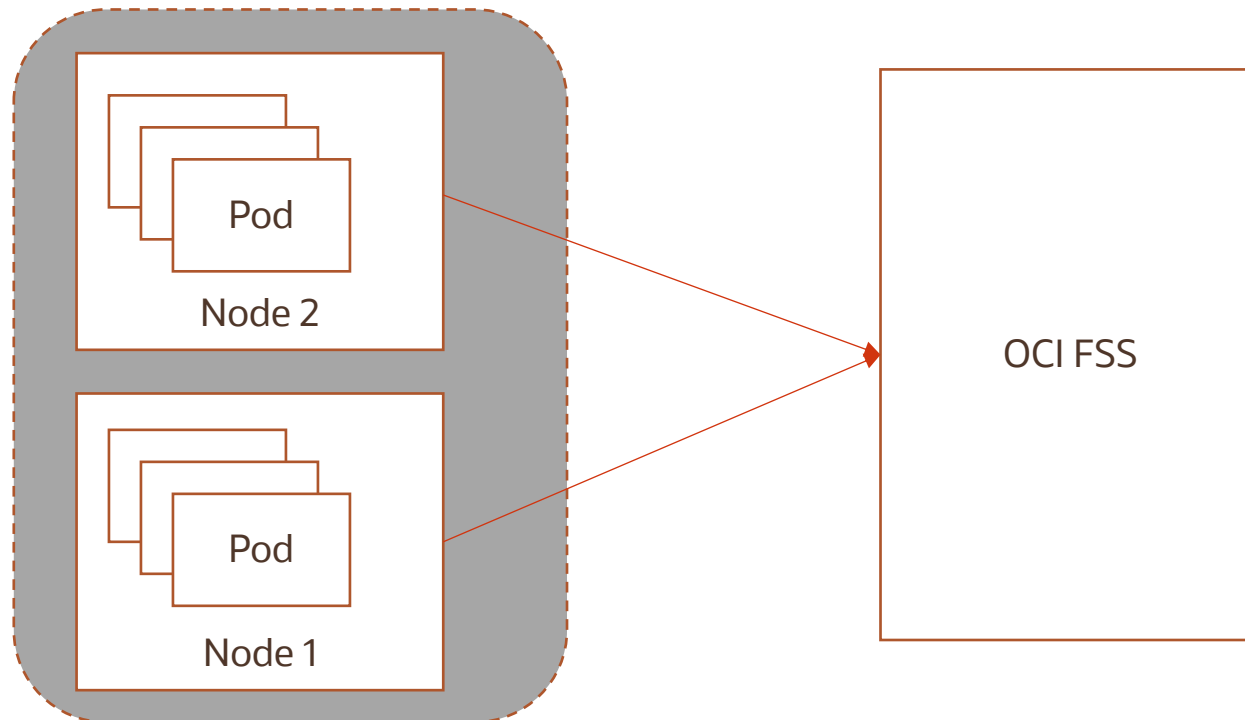




Using a network file system as persistent storage

# Why FSS as persistent storage?

- Container storage via a container's root file system is ephemeral, and can disappear upon container deletion and creation.
- Containers that need to maintain state can use OCI File System Storage
- OCI FSS can be shared across multiple worker nodes, allowing multiple PODs to have access to the same persistent storage device.



# Creating a PVC using OCI FSS..

- Create a storage Class

```
1 kind: StorageClass
2 apiVersion: storage.k8s.io/v1beta1
3 metadata:
4 name: fsstest1
5 provisioner: oracle.com/oci-fss
6 parameters:
7 # Insert mount target OCID from the FSS here
8 mntTargetId: ocid1.mounttarget.oc1.iad.aaaaaa4r
```

- Create a Persistent Volume and a persistent volume claim from this storage class.

```
apiVersion: v1
kind: PersistentVolume
metadata:
 name: oke-fsspv
spec:
 storageClassName: fsstest1
 capacity:
 storage: 100Gi
 accessModes:
 - ReadWriteMany
 mountOptions:
 - nosuid
 nfs:
 # Replace this with the IP of your FSS file system in OCI
 server: 172.16.5.4
 # Replace this with the Path of your FSS file system in OCI
 path: "/okefss-1"
 readOnly: false
```

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
 name: oke-fsspvc
spec:
 storageClassName: fsstest1
 accessModes:
 - ReadWriteMany
 resources:
 requests:
 # Although storage is provided here it is not used for FSS file systems
 storage: 100Gi
 volumeName: oke-fsspv
```

# Use the volume with a Pod..

- Create a Pod and mount to PVC
- Use node labels for launch multiple Pods across different Nodes and test persistent storage across pods.
  - \$kubectl label node 172.16.10.2 nodeName=worker1
  - \$kubectl label node 172.16.11.2 nodeName=worker2
  - Create Pods with yaml as below (Note: Volume details and nodeSelector)

```
apiVersion: v1
kind: Pod
metadata:
 name: nginx-fss
spec:
 containers:
 - name: web
 image: nginx
 volumeMounts:
 - name: nfs
 mountPath: "/usr/share/nginx/html/"
 ports:
 - containerPort: 80
 name: http
 volumes:
 - name: nfs
 persistentVolumeClaim:
 claimName: oke-fsspvc
 readOnly: false
 nodeSelector:
 nodeName: worker1
```

```
apiVersion: v1
kind: Pod
metadata:
 name: nginx-fss2
spec:
 containers:
 - name: web
 image: nginx
 volumeMounts:
 - name: nfs
 mountPath: "/usr/share/nginx/html/"
 ports:
 - containerPort: 80
 name: http
 volumes:
 - name: nfs
 persistentVolumeClaim:
 claimName: oke-fsspvc
 readOnly: false
 nodeSelector:
 nodeName: worker2
```

# Use the volume with a Pod..

- Use the following command to write some data in the shared persistent storage from both PODs and test if it is persistent and available across both pods.
  - `$kubectl exec -it nginx-fss bash`

```
sararif-mac:fss1 sararif$ kubectl exec -it nginx-fss bash
root@nginx-fss:/# cat /usr/share/nginx/html/hello.txt
HI THERE, I AM POD 1 on WORKER Node 1
Good to hear from POD1, this is POD2 from workernode 2
root@nginx-fss:/# exit
exit
sararif-mac:fss1 sararif$ kubectl exec -it nginx-fss2 bash
root@nginx-fss2:/# cat /usr/share/nginx/html/hello.txt
HI THERE, I AM POD 1 on WORKER Node 1
Good to hear from POD1, this is POD2 from workernode 2
root@nginx-fss2:/# exit
exit
sararif-mac:fss1 sararif$
```

# Summary

After completing this lesson, you should be able to understand:

- How to create an OKE cluster using Custom Create
- How to connect to worker nodes of a private cluster via a Bastion Host
- How to expose an app using k8s service and use OCI LB to load balance the traffic
- Why Persistent Storage is required for containers and how you can
  - Using OCI Block Volume as persistent storage
  - Using OCI File System Storage as persistent storage



## **Oracle Cloud always free tier:**

[oracle.com/cloud/free/](https://oracle.com/cloud/free/)

## **OCI training and certification:**

[cloud.oracle.com/en\\_US/iaas/training](https://cloud.oracle.com/en_US/iaas/training)

[cloud.oracle.com/en\\_US/iaas/training/certification](https://cloud.oracle.com/en_US/iaas/training/certification)

[education.oracle.com/oracle-certification-path/pFamily\\_647](https://education.oracle.com/oracle-certification-path/pFamily_647)

## **OCI hands-on labs:**

[ocitraining.qcloudable.com/provider/oracle](https://ocitraining.qcloudable.com/provider/oracle)

## **Oracle learning library videos on YouTube:**

[youtube.com/user/OracleLearning](https://youtube.com/user/OracleLearning)

