ORACLE
Cloud Infrastructure

Container Engine for Kubernetes (OKE)

Level 200

Jamal Arif

Dec 2018

ORACLE

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, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

ORACLE

Objectives

After completing this lesson, you should be able to:

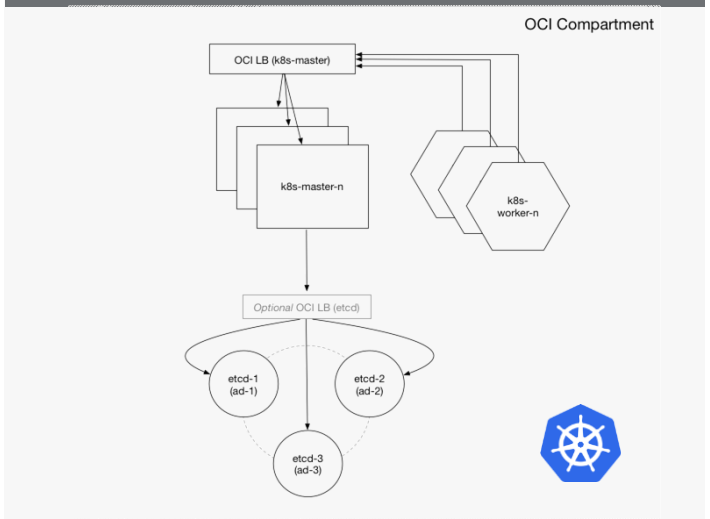
- Describe the OCI Container Engine for Kubernetes
- Managing a Kubernetes Cluster on OCI
- **Pre-requisites: Docker and Kubernetes basic understanding**

Oracle Cloud Infrastructure and Kubernetes
Roll Your Own, Pre-Built Installer, Managed Service

OCI

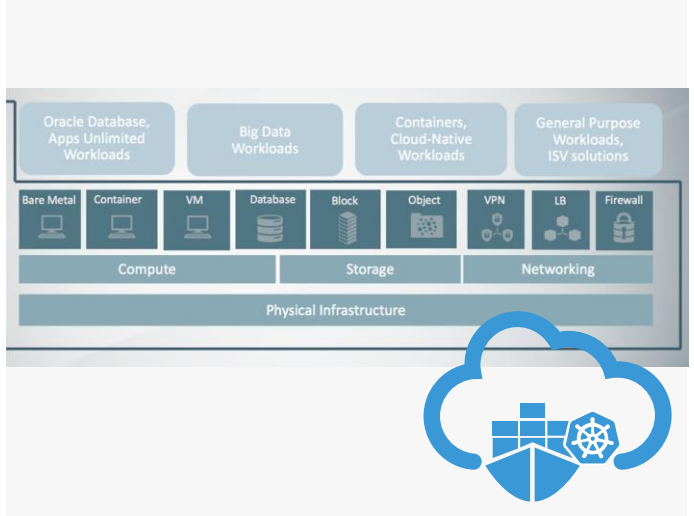
DIY Container Management

Quickstart Experience
OSS Terraform Installer on GitHub



Self Managed Kubernetes Service

OCI Container Engine for Kubernetes



Enterprise Class Managed Kubernetes Service

IaaS

CaaS

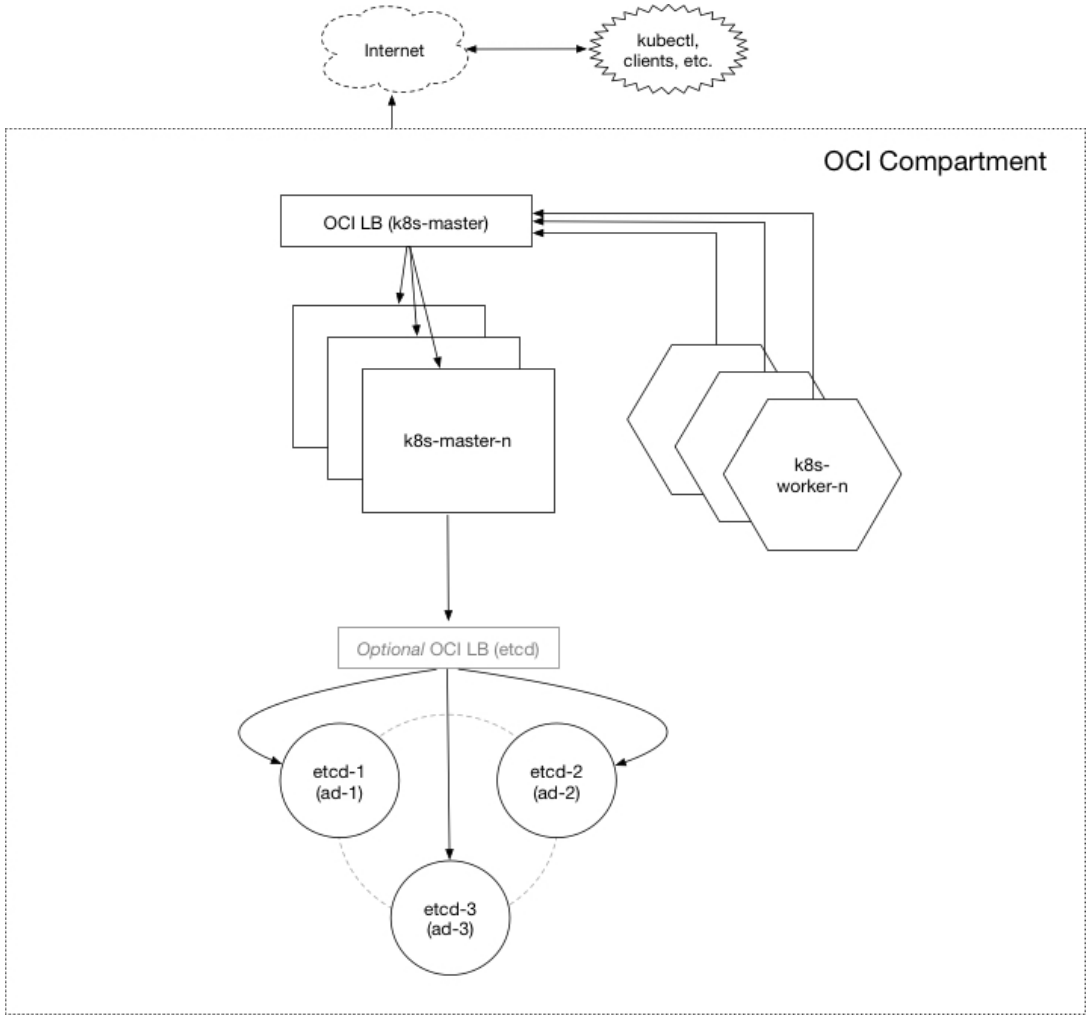
DIY - Terraform Kubernetes Installer for OCI

Open Source OCI Kubernetes installer, based on Terraform

- Oracle developed for Kubernetes on OCI
- Available now on Github - https://github.com/oracle/terraform-kubernetes-installer

Key Highlights

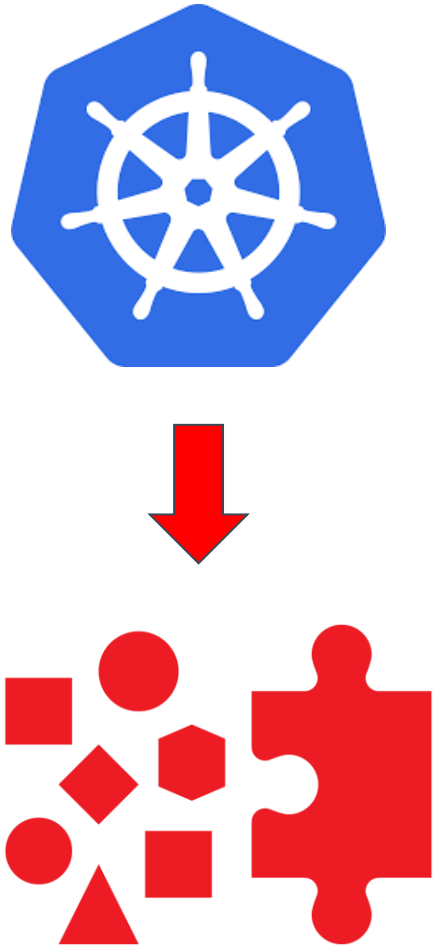
- Highly available Kubernetes cluster configured in your OCI tenancy and compartment
- Creates VCN, subnets, LBs and instances for control plane
- Specify number and shape of nodes for your cluster
- Scale your cluster as needed



Available on Oracle Github!

Kubernetes Challenges

- Managing, maintaining, upgrading Kubernetes Control Plane
  - API Server, etcd, scheduler etc….
- Managing, maintaining, upgrading Kubernetes Data Plane
  - In place upgrades, deploy parallel cluster etc….
- Figuring out container networking & storage
  - Overlays, persistent storage etc… - it should just work
- Managing Teams
  - How do I manage & control team access to my clusters?
- CI/CD Integration
  - How do I drive automated testing and conditional release into my application lifecycle?



Introducing Container Engine for Kubernetes - OKE

What is It?

- Managed Kubernetes container service to deploy and run your own container based apps
- Tooling to create, scale, manage & control your own standard Kubernetes clusters instantly

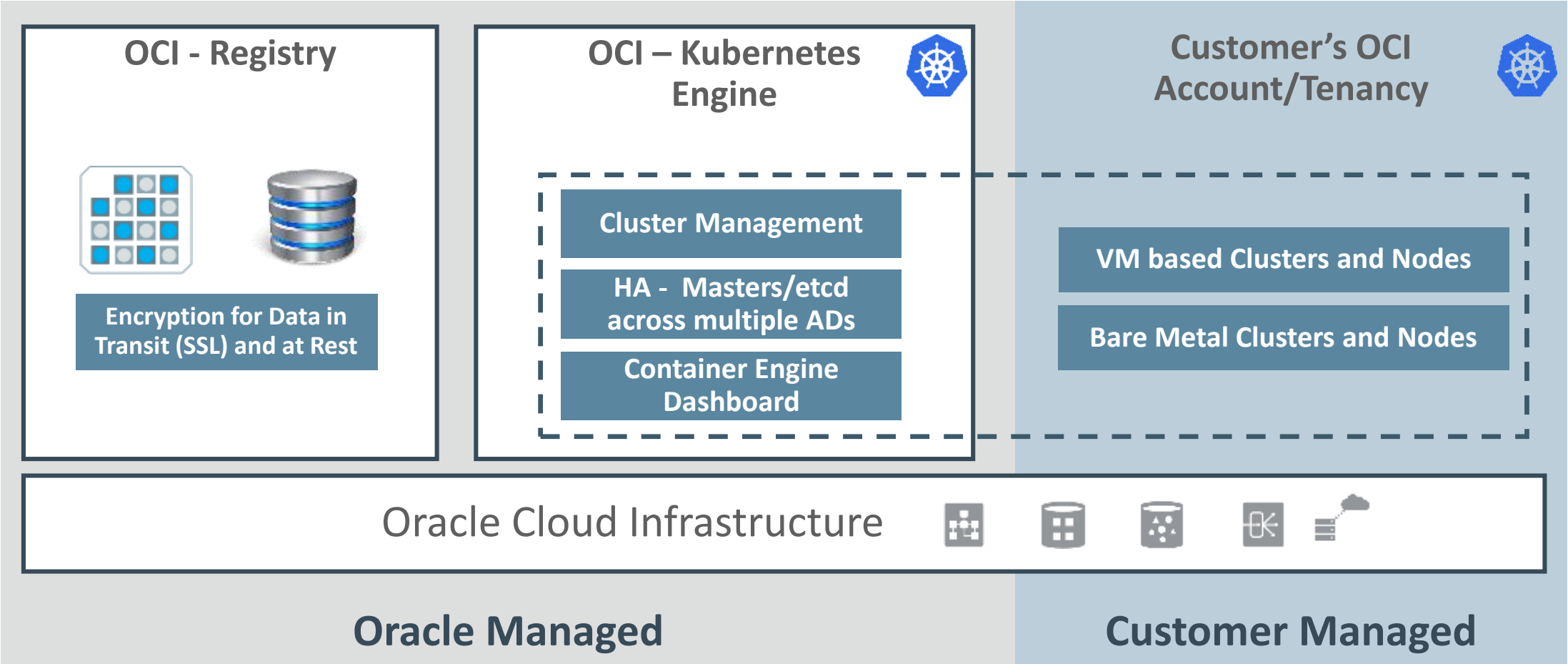
What Problems Does it Solve?

- Too complex, costly and time consuming to build & maintain Kubernetes environments
- Too hard to integrate Kubernetes with a registry and build process for container lifecycle management
- Too difficult to manage and control team access to production clusters

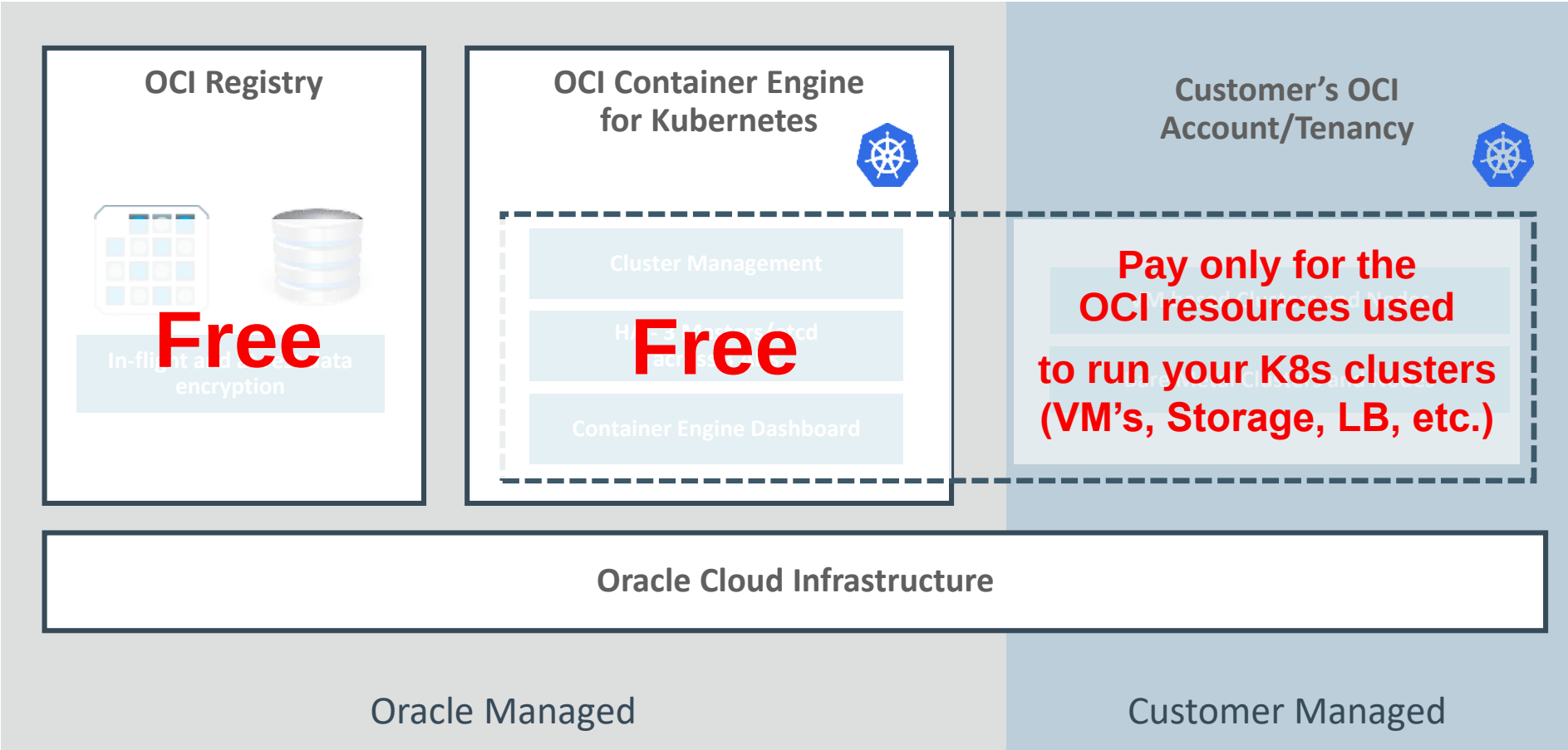
Key Benefits

- Enables developers to get started and deploy containers quickly. Gives DevOps teams visibility and control for Kubernetes management.
- Combines production grade container orchestration of open Kubernetes, with control, security, IAM, and high predictable performance of Oracle's next generation cloud infrastructure

Working with OKE and OCIR on OCI

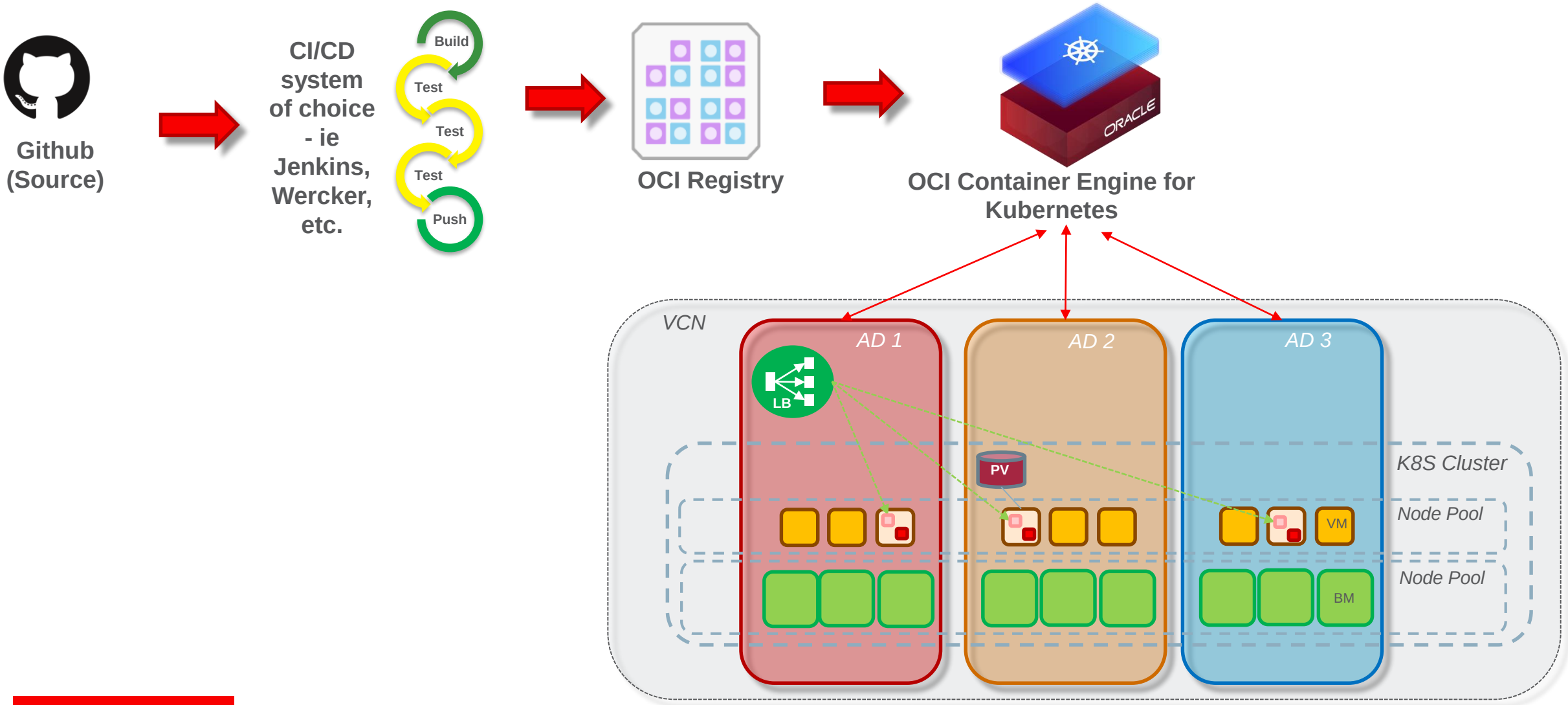


OKE/OCIR Pricing and Packaging

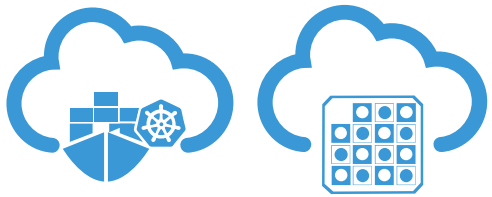


OCI Container Engine for Kubernetes and Registry

An Open, Fully-Managed Kubernetes Platform & Private Registry



Oracle Container Engine & Registry



Container Native

- **Standard Docker & Kubernetes**
  - Deploy standard & open upstream Docker and Kubernetes versions for compatibility across environments
- **Registry Integration**
  - Full Docker v2 compatible private registry to store and manage images
- **Container Engine**
  - Deploy and operate containers and clusters
- **Full integration to cloud networking and storage**
  - Leverage the enterprise class networking, load balancing and persistent storage of Oracle Cloud Infrastructure

Developer Friendly

- **Streamlined Workflow**
  - Use your favorite CI to push containers to the registry, then Kubernetes to deploy to clusters and manage operations
- **Full REST API**
  - Automate the workflow, create and scale clusters through full REST API
- **Built In Cluster Add-Ons**
  - Kubernetes Dashboard, DNS & Helm
- **Open Standards**
  - Docker Based Runtime
  - Worker Node SSH Access
  - Standard Kubernetes

Enterprise Ready

- **Simplified Cluster Operations**
  - Use the fully managed, highly available registry, master nodes and control plane
- **Full Bare Metal Performance and Highly Available IaaS**
  - Combine Kubernetes with bare metal shapes for raw performance
  - Deploy Kubernetes clusters across multiple Availability Domains for resilient applications
- **Team Based Access Controls**
  - Control team access and permissions to clusters
- **Autonomous Clusters**
  - Maintain cluster size and performance in face of node failures and load fluctuations

ORACLE

Pre-requisites for OKE (1) - Service Limits for tenancy

- Monthly universal Credits have limit of 3 clusters per OCI region with 1000 nodes in a cluster
- Pay-as-you-go or Promo accounts to contact support to activate clusters.
- Must also have compute Instance Quota **(Required)** – to launch k8s worker nodes in an AD or across ADs for HA
- Block Volume Quota – Only required if you want to create k8s persistent volumes
- Load Balancer Quota – Only required if you want to distribute traffic between worker nodes

Pre-requisites for OKE (2) – Required IAM policies

- 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: if users will be using the Console to create and update clusters, polices must also grant the dev-team group the Networking permissions VCN_READ and SUBNET_READ.

Pre-requisites for OKE (3) – Basic Virtual Cloud Network Config

- An Existing VCN with following
  - Internet Gateway
  - Route table with default route to IGW
  - K8s worker node subnets – at least three subnets in different ADs for High Availability of Workers
  - LBs Subnets – 2 subnets in different ADs for OCI Public Load Balancer
  - Separate Security Lists for K8s Worker Nodes Subnets and LB Subnets
  - 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
    - ingress rules to allow Container Engine for Kubernetes to access worker nodes on port 22 from 130.35.0.0/16, 138.1.0.0/17, 147.154.0.0/16, 192.29.0.0/16
    - an egress rule that allows all outbound traffic to the internet

Kubernetes Cluster Creation

- **Quick Cluster** – Uses default settings to create a 'quick cluster' with new network resources as required.
- **Custom Cluster** – custom settings to create a 'custom cluster'. This approach gives you the most control over the new cluster.

Kubernetes Cluster Creation – Quick Cluster

- **Name** – Name of the K8s Cluster
- **Version** - The version of Kubernetes to run on the master node of the cluster
- **Quick Create –** Creates following new resources
  - VCN
  - 2 LB Public Subnets
  - 3 Worker Node Subnets
  - Security List with required security list rules
  - Route table with default gateway to IGW
  - Internet Gateway
- **Node Pool** – Shape of the VM and quantity per subnet with your id_rsa.pub key
- **Kubernetes Labels (key, value) -** 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
- **Additional Add-on** – Kubernetes Dashboard, Tiller (helm)

Kubernetes Cluster Creation – Quick Cluster

Cluster Creation help close

CLUSTER COMPARTMENT

tutorials

NAME

cluster2

KUBERNETES VERSION

v1.11.1

Kubernetes version installed on your master and worker 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

Create Node Pool

NAME: pool1 COMPARTMENT: tutorials KUBERNETES VERSION: v1.11.1 IMAGE: Oracle-Linux-7.5

SHAPE

VM.Standard2.1

The shape of all nodes in the pool

QUANTITY PER SUBNET

1

The number of nodes per subnet.

PUBLIC SSH KEY *OPTIONAL*

Input SSH public key

The SSH public key to access your nodes.

Kubernetes Labels

KEY: name VALUE: pool1

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

Create Virtual Cloud Network

A new VCN network will be created for you in order to have a functioning cluster

COMPARTMENT: tutorials RESOURCE CREATION: 1 VCN, 2 service lb subnets and 3 worker node subnets

Additional Add Ons

- [x] KUBERNETES DASHBOARD ENABLED
- [x] TILLER (HELM) ENABLED

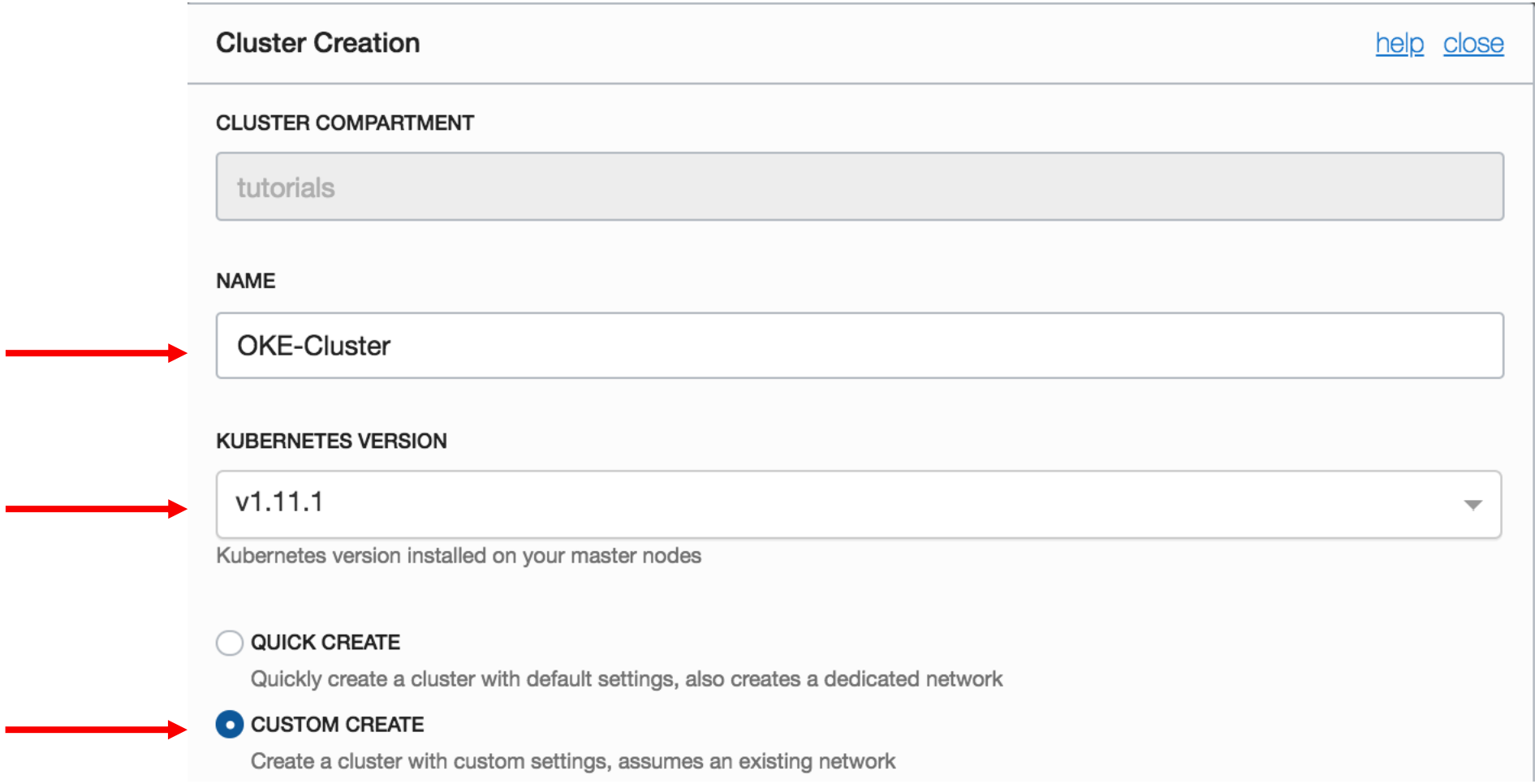
Kubernetes Cluster Creation – Custom Cluster

- **Name** – Name of the K8s Cluster
- **Version** - The version of Kubernetes to run on the master node of the cluster
- **VCN** - The name of an existing virtual cloud network that has been configured for cluster creation and deployment
- **Kubernetes Service LB Subnets:** The two subnets configured to host load balancers.
- **Kubernetes Service CIDR Block (Optional):** The available group of network addresses that can be exposed as Kubernetes services (ClusterIPs), expressed as a single, contiguous IPv4 CIDR block. For example, 10.96.0.0/16. Must not overlap with VCN CIDR
- **Pods CIDR Block (Optional):** The available group of network addresses that can be allocated to pods running in the cluster, expressed as a single, contiguous IPv4 CIDR block. For example, 10.244.0.0/16. Must not overlap with VCN CIDR
- **Kubernetes Dashboard and Helm** are enabled by default

Kubernetes Worker Nodes – Node pools

- **Name** – Name of the node pool
- **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 node is selected. The Kubernetes version on worker nodes must be either the same version as that on the master node, or an earlier version that is still compatible
- **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 number of CPUs and the amount of memory allocated to each node in the node pool.
- **Subnet:** One or more subnets configured to host worker nodes. The worker node subnets must be different to the load balancer subnets.
- **Quantity per Subnet:** The number of worker nodes to create for the node pool in each subnet.
- **Public SSH Key:** (Optional) The public key portion of the key pair you want to use for SSH access to each node in the node pool.
- **Kubernetes Labels (key, value) -** 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

Kubernetes Cluster Creation – Custom Cluster



Kubernetes Cluster Creation – Custom Cluster

Network Selection

NETWORK COMPARTMENT

tutorials

jamalarif (root)/tutorials

VCN

oke-vcn-quick-cluster1-20181204205349

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

× oke-svclbsubnet-quick-cluster1-20181204205349-fyhg:PHX-AD-2

× oke-svclbsubnet-quick-cluster1-20181204205349-fyhg:PHX-AD-1

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).

Additional Add Ons

- [x] KUBERNETES DASHBOARD ENABLED
- [x] TILLER (HELM) ENABLED

Kubernetes Cluster Creation – Custom Cluster

Node Pool

NAME

NodePoolCustom

VERSION

v1.11.1

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

IMAGE

Oracle-Linux-7.5

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

SHAPE

VM.Standard2.1

The shape for all nodes in the node pool.

SUBNETS

- × oke-subnet-quick-cluster1-20181204205349-fyhg:PHX-AD-3
- × oke-subnet-quick-cluster1-20181204205349-fyhg:PHX-AD-1
- × oke-subnet-quick-cluster1-20181204205349-fyhg:PHX-AD-2

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.

PUBLIC SSH KEY *OPTIONAL*

Input SSH public key

The SSH public key to access your nodes.

Kubernetes Labels

prod | AppA

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

Kubernetes Cluster Details and Node Pools

OKE-Cluster-Demo

Access Kubeconfig | Delete Cluster

Cluster Details

Cluster Status: Active

Node Pools: 1

Cluster ID: ...izdemy4d Show Copy

Launched: Thu, 17 May 2018 20:07:18 GMT

Services Cidr: 10.96.0.0/16

Kubernetes Version: v1.9.7

Kubernetes Address: ...com:6443 Show Copy

VCN Name: OKE-VCN-Ashburn

VCN Id: ...wbqnt7bq Show Copy

Pods Cidr: 10.244.0.0/16

Node Pools

Add Node Pool

NodePool1

Actions

Kubernetes Ver: v1.9.7

Shape: VM.Standard1.2

Image Name: Oracle-Linux-7.4

Total Worker Nodes: 3

Nodes Per Subnet: 1

Number of Subnets: 3

Hide Node Details

Instance Name	Node State	Subnet	Public IP
oke-czgizdemy4d-n4dgyjwge2g-sbpj5bkk37a-0	ACTIVE	Sub-AD1	129.213.84.121
oke-czgizdemy4d-n4dgyjwge2g-sirhjog44iq-0	ACTIVE	Sub-AD2	129.213.57.201
oke-czgizdemy4d-n4dgyjwge2g-sw56r34ywkq-0	ACTIVE	Sub-AD3	129.213.35.90

Showing 3 Item(s)

Scaling Node Pools

Node Pools

Add Node Pool

NodePool1

Actions

- Edit
- Scale
- Delete Node Pool

Kubernetes Ver: v1.9.7
Shape: VM.Standard1.2
Image Name: Oracle-Linux-7.4
Total Worker Nodes: 3
Nodes Per Subnet: 1
Number of Subnets: 3

Hide Node Details

Instance Name	Node State	Subnet	Public IP
oke-czgizdemy4d-n4dgyjwge2g-sbpj5bkk37a-0	ACTIVE	Sub-AD1	129.213.84.121
oke-czgizdemy4d-n4dgyjwge2g-sirhjog44iq-0	ACTIVE	Sub-AD2	129.213.57.201
oke-czgizdemy4d-n4dgyjwge2g-sw56r34ywkq-0	ACTIVE	Sub-AD3	129.213.35.90

Showing 3 Item(s)

Scale Node Pool

help close

You may scale certain characteristics of a node pool.

SUBNETS

× Sub-AD3 × Sub-AD2 × Sub-AD1

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.

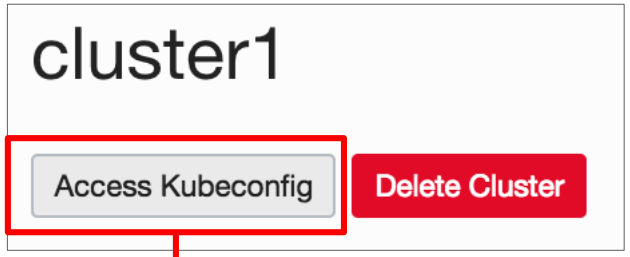
QUANTITY PER SUBNET

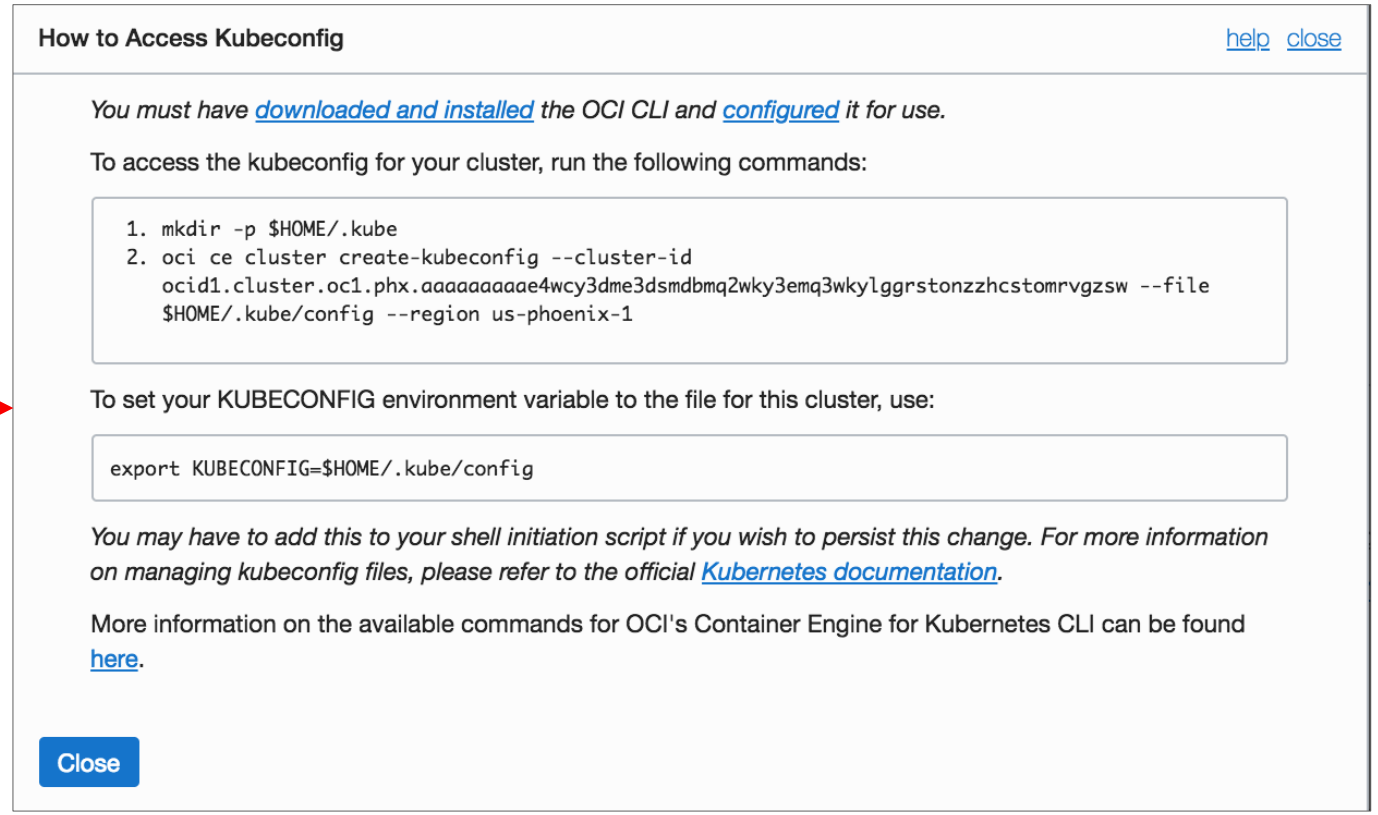
1

The number of nodes per subnet.

Scale

Accessing Kubernetes Cluster using kubectl





Accessing Kubernetes Cluster using kubectl

Resources

Node Pools

Work Requests

Getting Started

Getting Started

Kubernetes Dashboard

You can use the Kubernetes Dashboard to get an overview of applications running in your cluster. It also provides information on the state of Kubernetes resources in your clusters, and on any errors that may have occured.

```
1. kubectl proxy
2. Dashboard will be available at:
   http://localhost:8001/api/v1/namespaces/kube-system/services/https:kubernetes-dashboard:/proxy/
```

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

2 Check Version

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

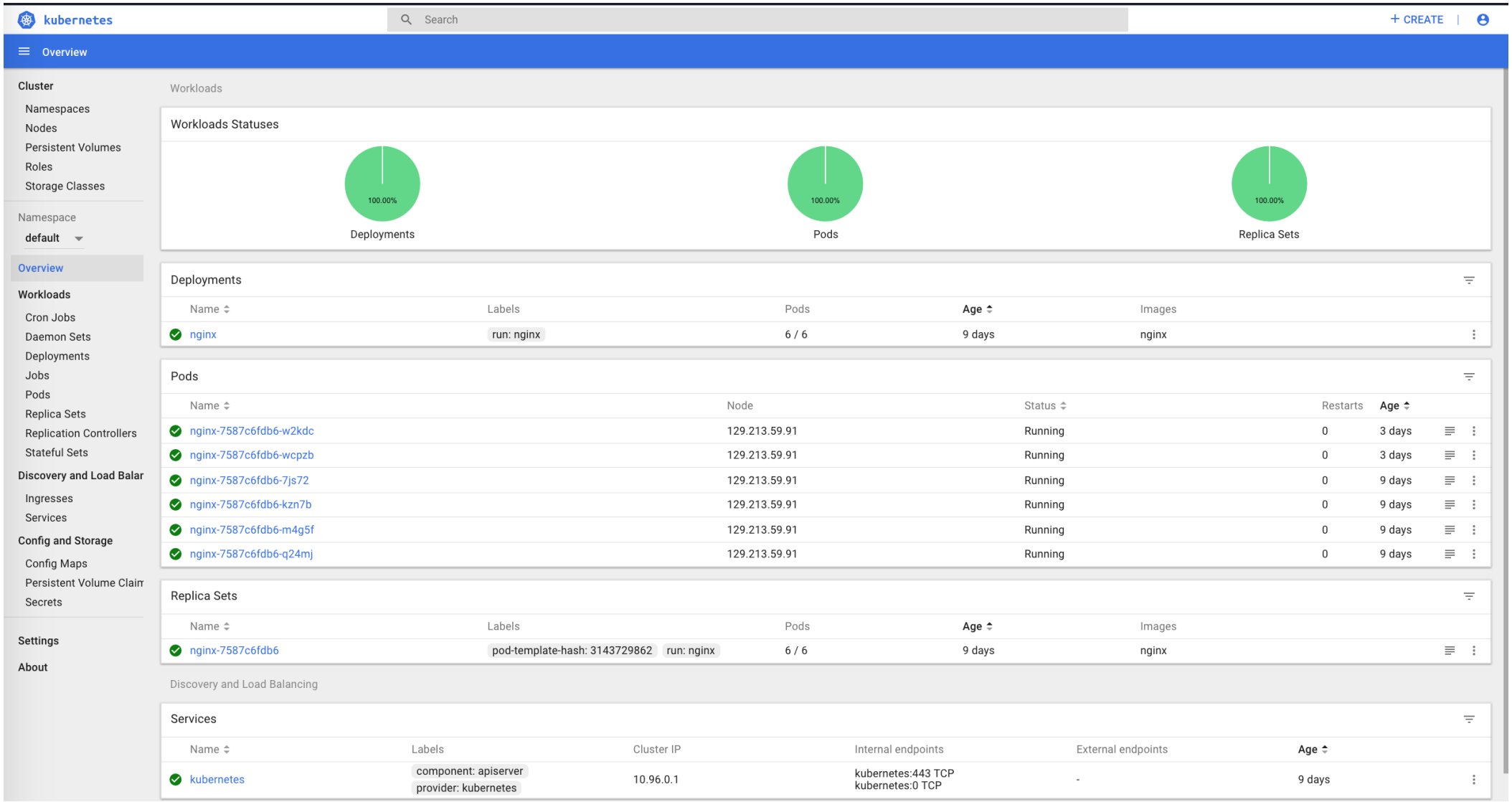
```
1. kubectl version
```

3 Deploy Application

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

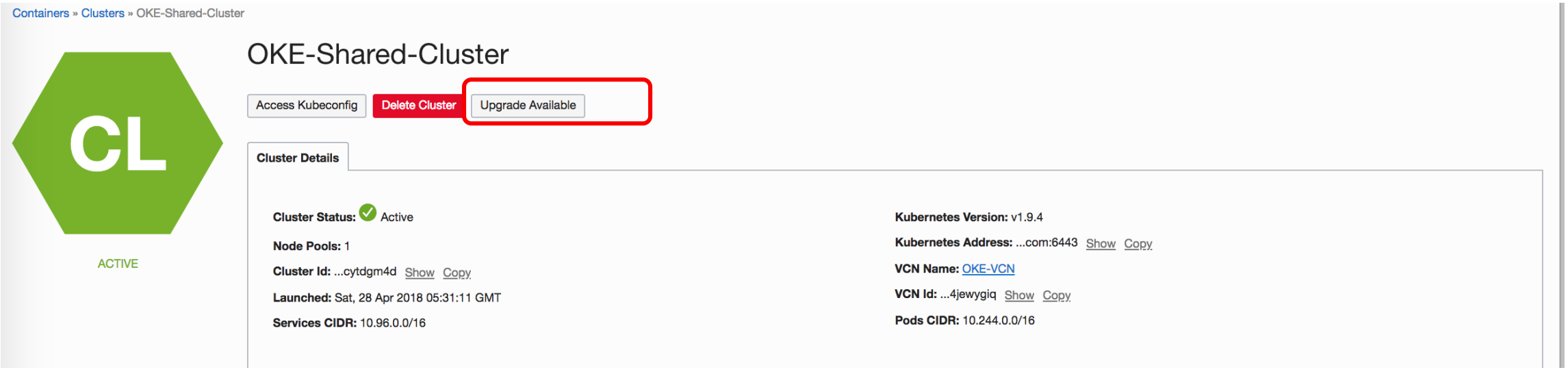
```
1. kubectl create -f
   https://k8s.io/docs/tasks/run-
   application/deployment.yaml
```

Kubernetes Dashboard



Upgrading Kubernetes Master

- OKE Service supports **in-place** upgrade of Kubernetes Master nodes (via Console or API)
  - After upgrading a master node to a newer version of Kubernetes, you cannot downgrade the master node to an earlier Kubernetes version.
  - The versions of Kubernetes running on the master node and the worker nodes must be compatible (that is, the Kubernetes version on the master node must be no more than two minor versions ahead of the Kubernetes version on the worker nodes)



Upgrading Kubernetes Worker Nodes

- Kubernetes Worker nodes are upgraded by performing an ‘out-of-place’ upgrade.
  - To upgrade the version of Kubernetes running on worker nodes in a node pool, you replace the original node pool with a new node pool that has new worker nodes running the appropriate Kubernetes version.
  - ‘drain' existing worker nodes in the original node pool to prevent new pods starting and to delete existing pods. Once no pods exist, the old node pool can be deleted

Summary

- Describe the OCI Container Engine for Kubernetes
- Launch a Kubernetes Cluster on OCI

ORACLE®
Cloud Infrastructure

cloud.oracle.com/iaas

cloud.oracle.com/tryit