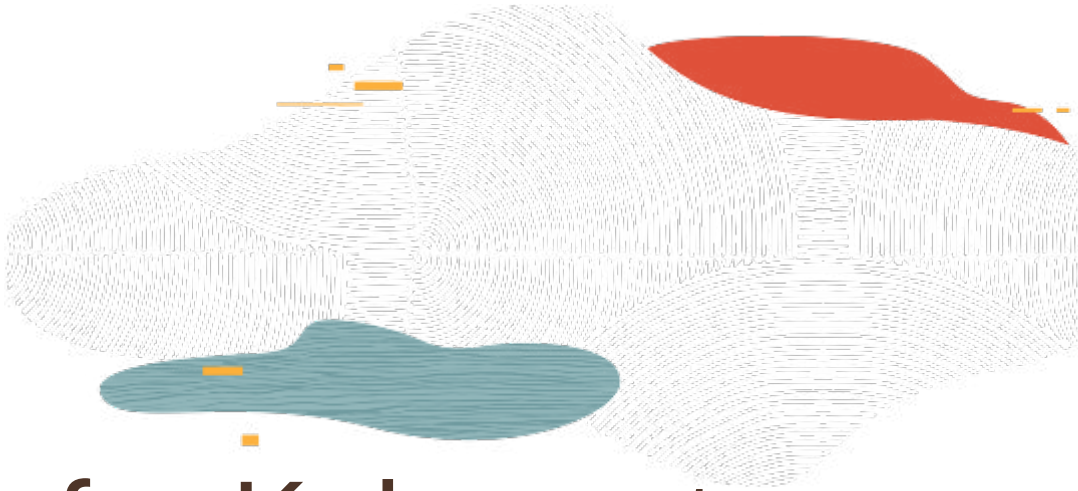


Oracle Container Engine for Kubernetes

Level 100

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:

- Containers, Docker container engine
- Orchestration systems and Kubernetes
- Oracle Container Engine for Kubernetes
- Creating a K8s cluster in OCI using ‘quickstart’

Key Containers / Orchestration Use Cases

	Share	Container Use Cases	Orchestration Use Cases
Development	65%	Developer productivity; Consistent appstacks in Dev, Test & Production	Automated deploys to accelerate application release cadence
CI/CD/DevOps	48%	Containerized dependencies; Container registries;	Rolling updates and reversals
Operations	41%	Standardized environments for dev, testing and operations	Resilient, self-healing systems; High Availability; Elastic Scalability
Refactor Legacy Apps	34%	Refactor from N-tier to portable containerized applications	Run distributed, stateful apps on scale-out infrastructure
Migrate to Cloud	33%	Move entire appstacks and see them run identically in the cloud	Cloud bursting; Reduce infrastructure costs by avoiding over-provisioning
New Microservice Apps	32%	Create small purpose-built services that can be assembled to scalable custom applications	Dynamically manage large-scale microservices infrastructure

SOURCE: THE EVOLUTION OF THE MODERN SOFTWARE SUPPLY CHAIN, DOCKER SURVEY 2016

Docker and Kubernetes

Docker Containers

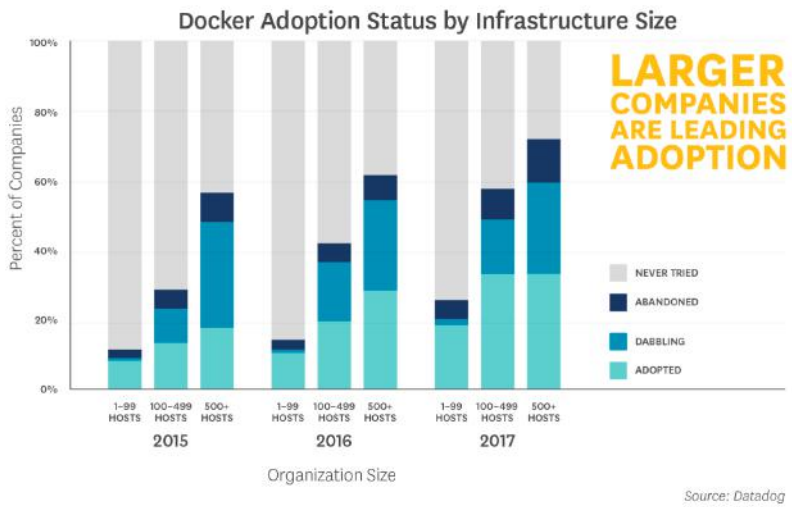
- Popular, easy to use tooling targeting developer productivity
- De facto standard container runtime and image format
- Used for developer on-boarding and 1st generation application management

Kubernetes Orchestration

- Production grade container management targeting DevOps and operations, with widespread adoption
- Complex but powerful toolset supporting cloud scale applications
- Rich operations feature set, autoscaling, rolling upgrades, stateful apps and more.

Docker & Kubernetes Lead the Market

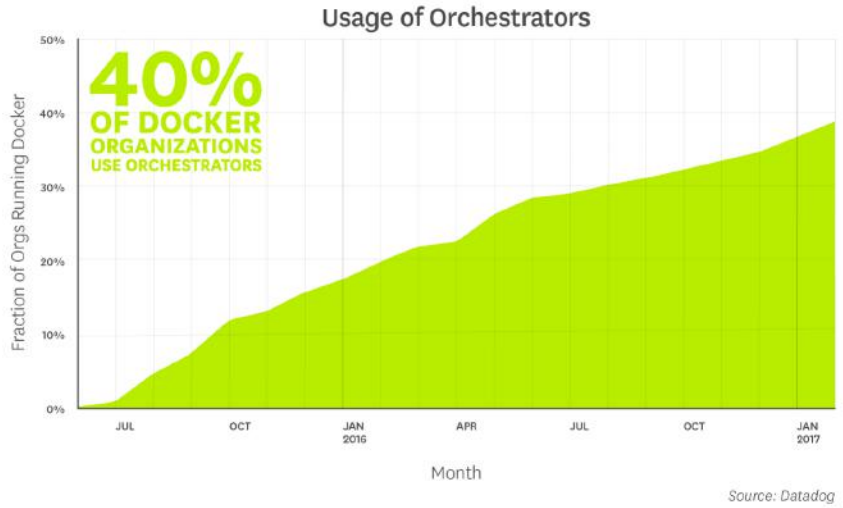
Containers (Docker)



60% of enterprise companies (500+ hosts) use Docker

15% of all the hosts at these companies run Docker

Orchestration (Kubernetes)



40% of Docker users also use orchestrators

80% of these orchestration users prefer Kubernetes

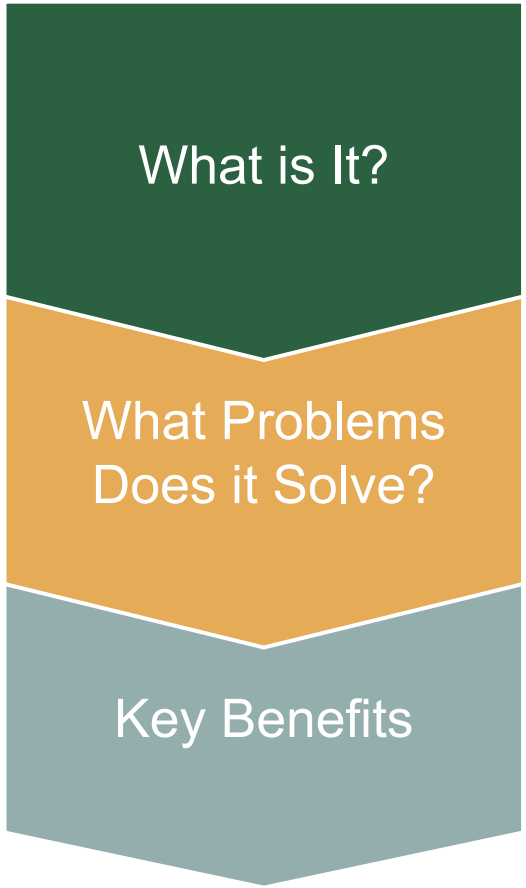
Container Orchestration And Containers as a Service (CaaS)

- Multi-container apps
- Scheduling
- Service Discovery
- Maintaining Desired State

- Orchestration as a service
- Hosted Container Runtime
- Minimize operational overhead

Container Engine for Kubernetes - OKE

Introducing Container Engine for Kubernetes - OKE



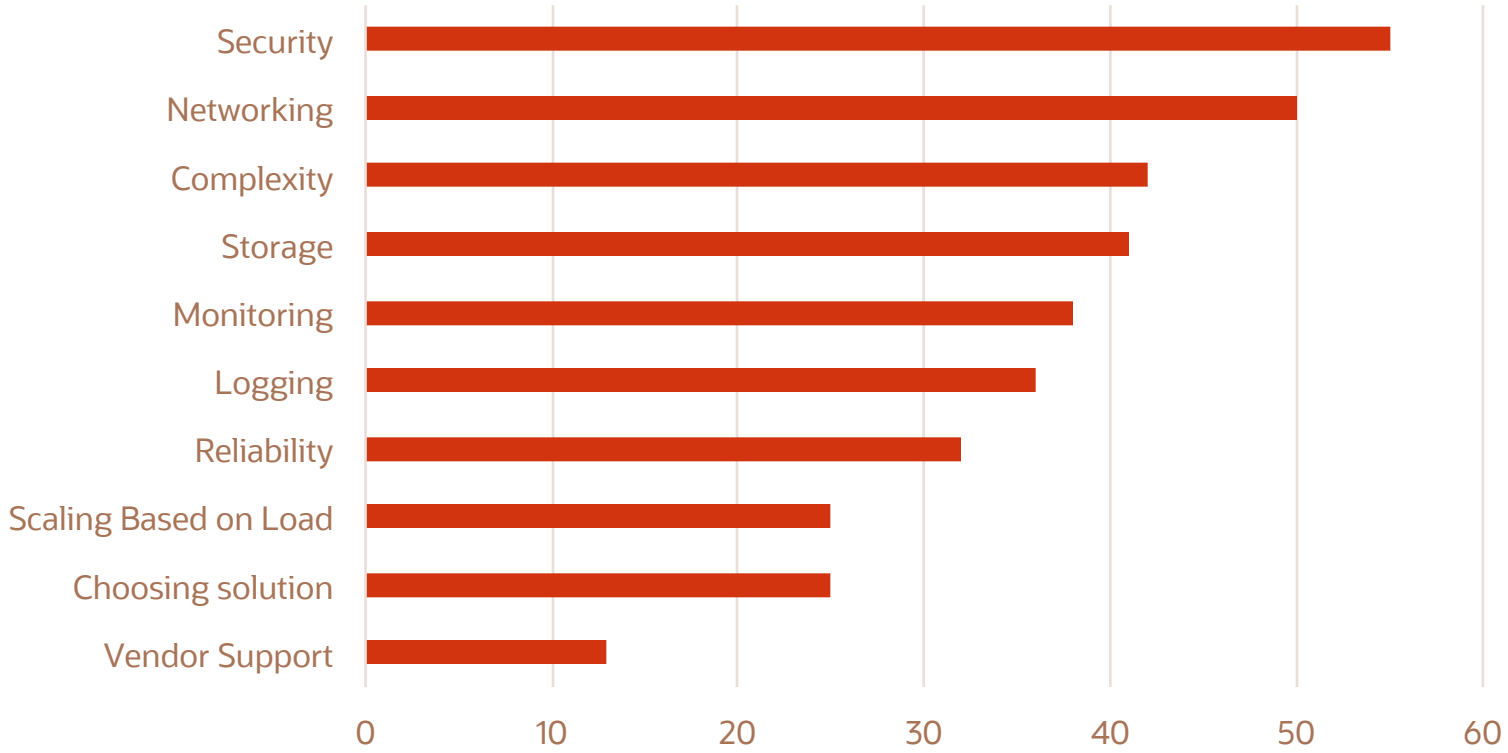
- Managed Kubernetes container service to deploy and run your own container based apps
- Tooling to create, scale, manage & control your own standard Kubernetes clusters instantly

- Too complex, costly and time consuming to build & maintain 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

- 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

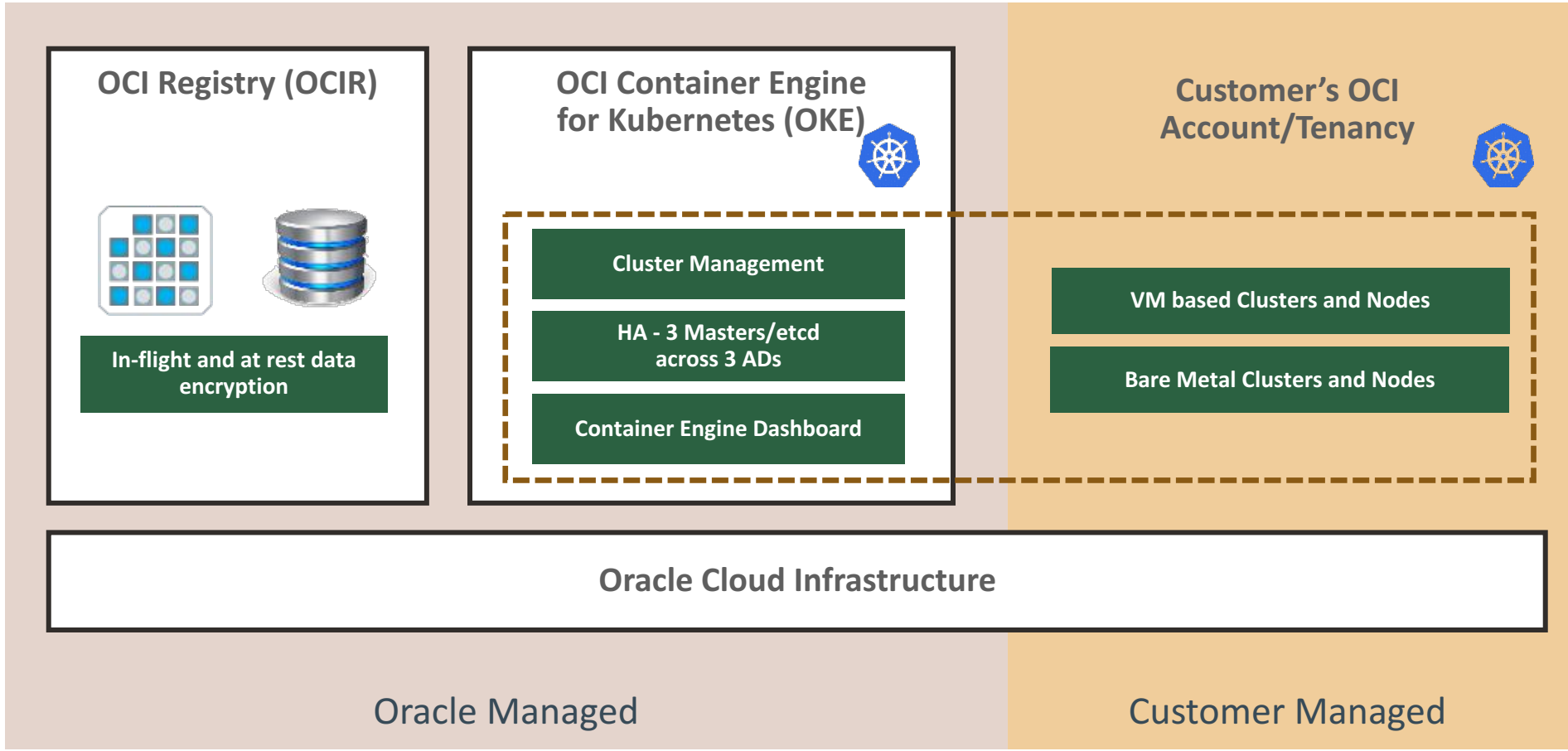
Kubernetes Challenges

- Managing Kubernetes Infrastructure, upgrading, security
- Container networking & persistent storage
- Managing Teams & Access
- CI/CD Integration, automated testing, conditional release

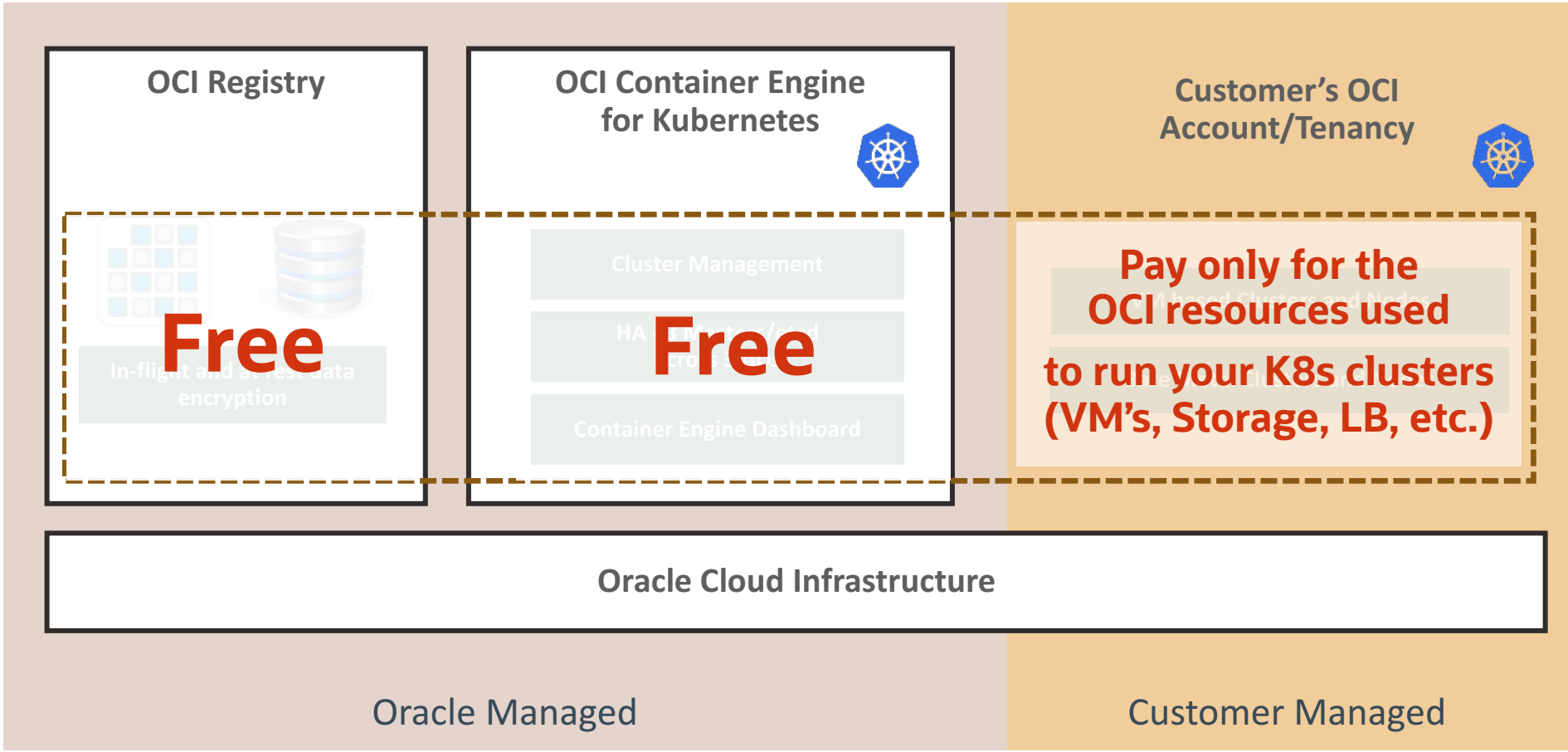


(Source: CNCF Survey, The New Stack, 22 Mar 2018)

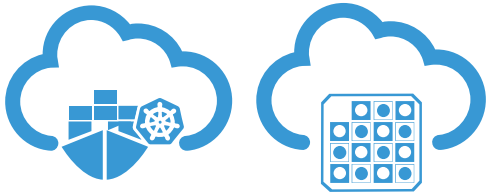
Working with OKE and OCIR on OCI



OKE/OCIR Pricing and Packaging



Oracle Container Engine (OKE) and 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

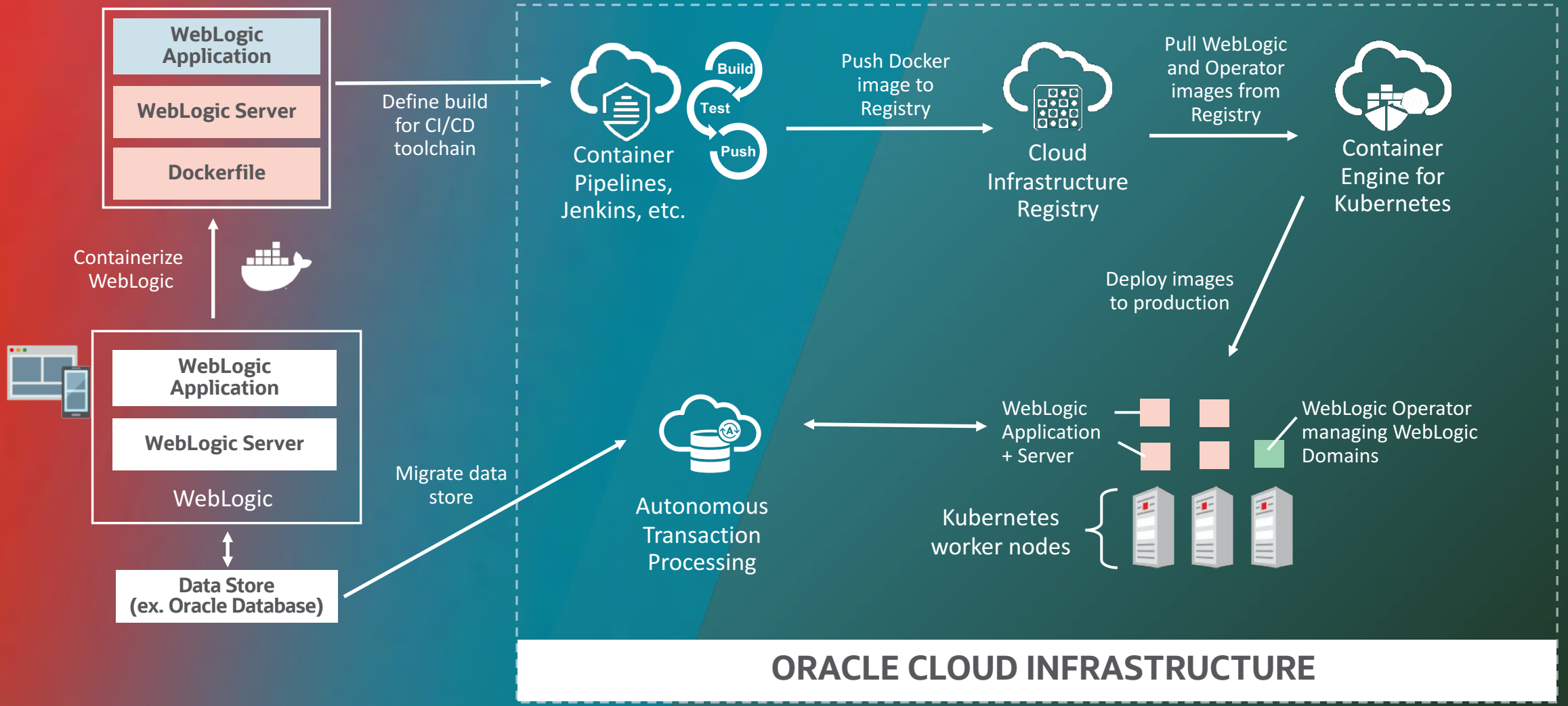
Developer Friendly

- **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

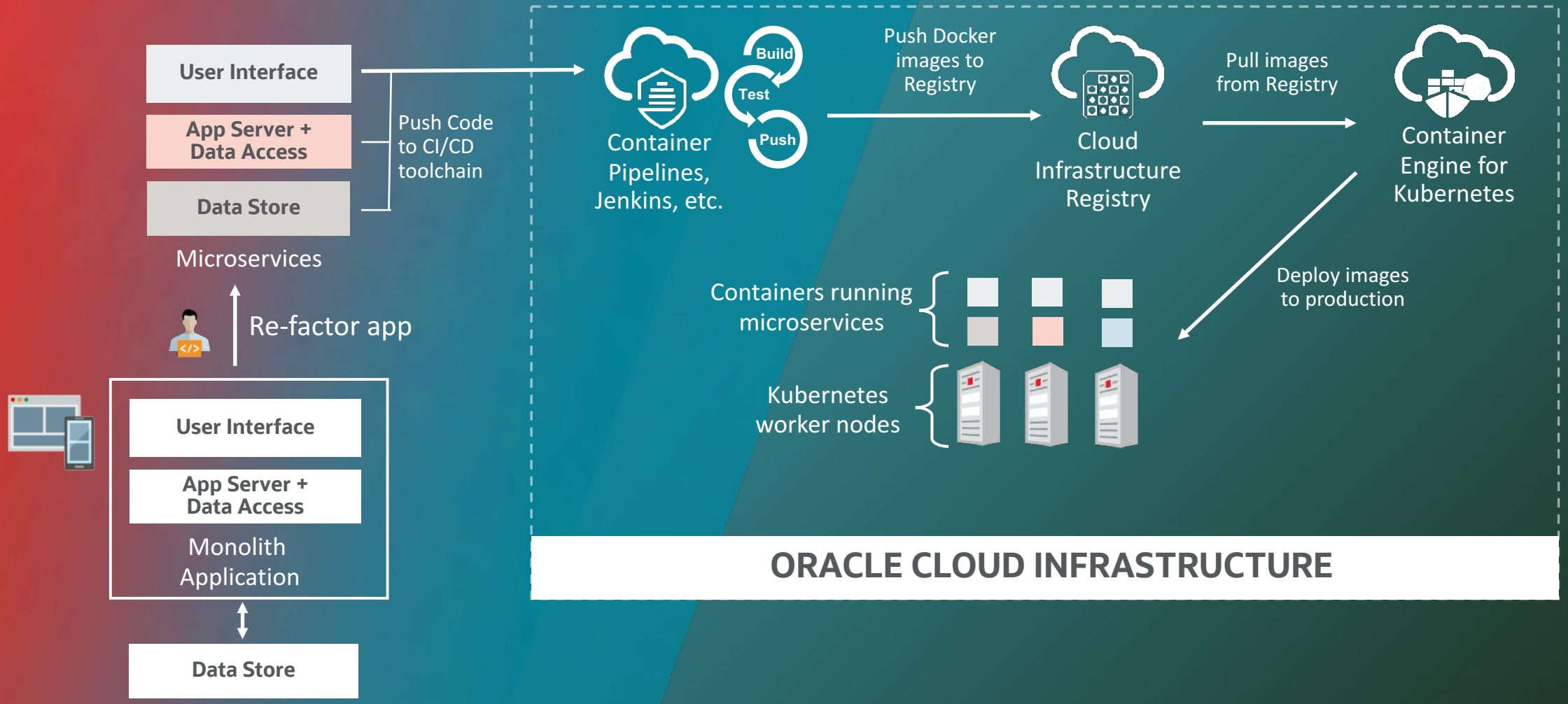
Enterprise Ready

- **Simplified Cluster Operations**
  - Fully managed, highly available registry, master nodes and control plane
  - One-click Quick Create for secure Private Worker Nodes/Subnets
- **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

Containers Use Case: Lift & Shift WebLogic Application



Containers Use Case: Refactor an Existing Application



Creating an OKE Cluster in OCI

Pre-requisites for creating a K8s Cluster via Quickstart

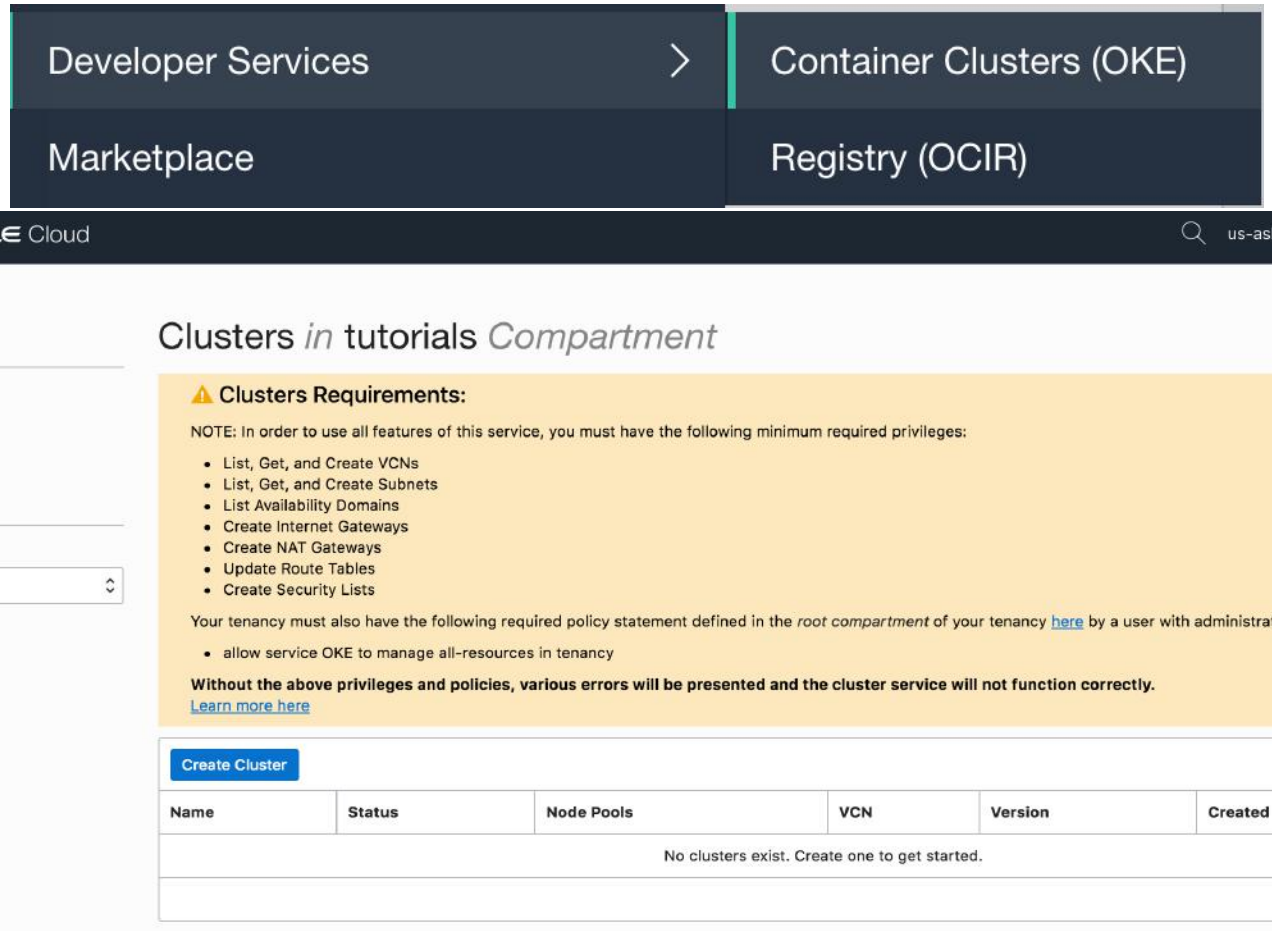
- Monthly universal Credits have limit of 3 clusters per OCI region with 1000 nodes in a cluster and Pay-as-you-go or Promo accounts have a limit for One Cluster (by default)
- Must also have compute Instance Quota (Required) – to launch k8s worker nodes in an AD or across ADs for HA
- Required Policy 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: Polices 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

OKE Quickstart

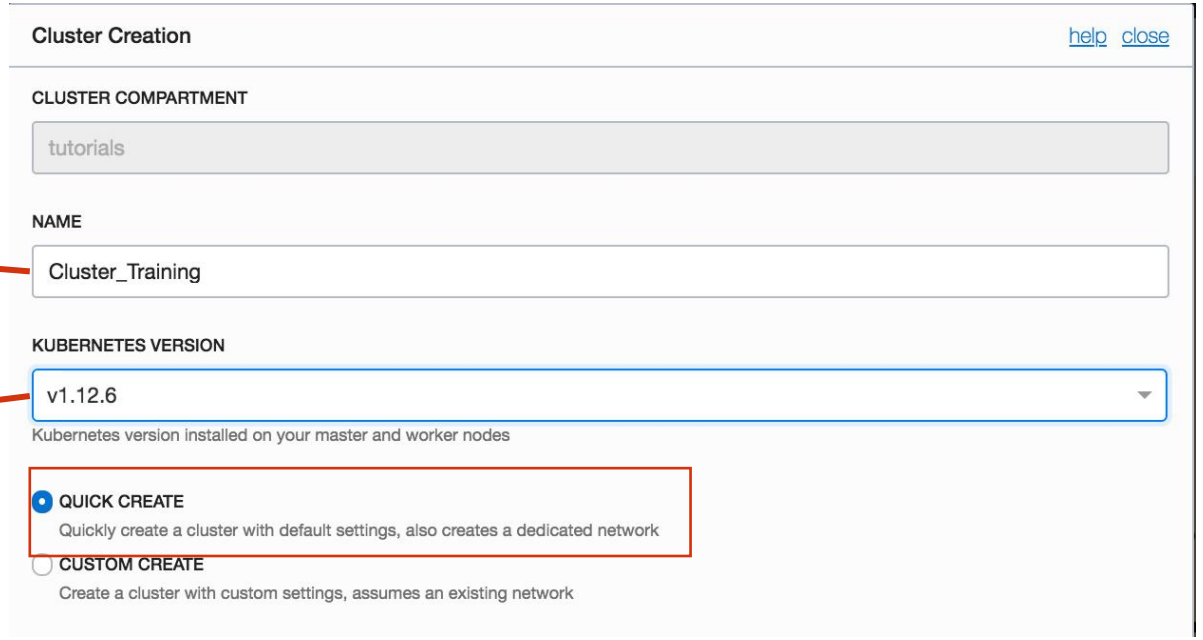
Step 1: Navigate to Menu → Developer Services → Container Clusters (OKE) → Create Cluster



OKE Quickstart

Step 2: Cluster Creation

Name of the Cluster

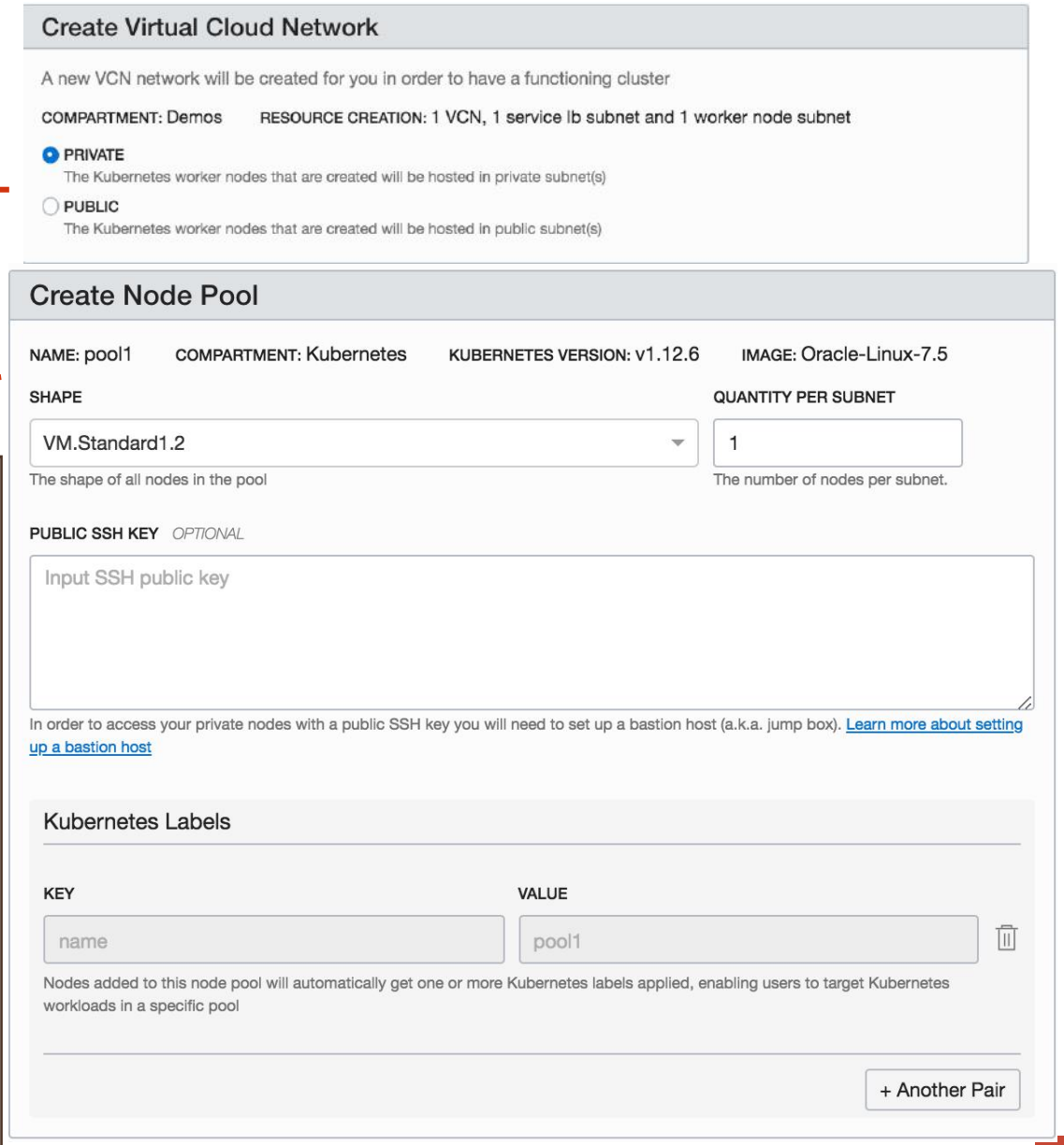


The version of Kubernetes to run on the master nodes and worker nodes of the cluster. Either accept the default version or select a version of your choice. Amongst other things, the Kubernetes version you select determines the default set of admission controllers that are turned on in the created cluster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

OKE Quickstart (contd…)

Step 2: Cluster Creation

New network resources for the cluster are created automatically, the worker nodes in a 'quick cluster' can be created in private subnets or public. A NAT gateway is created in case of private subnets.



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.

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)

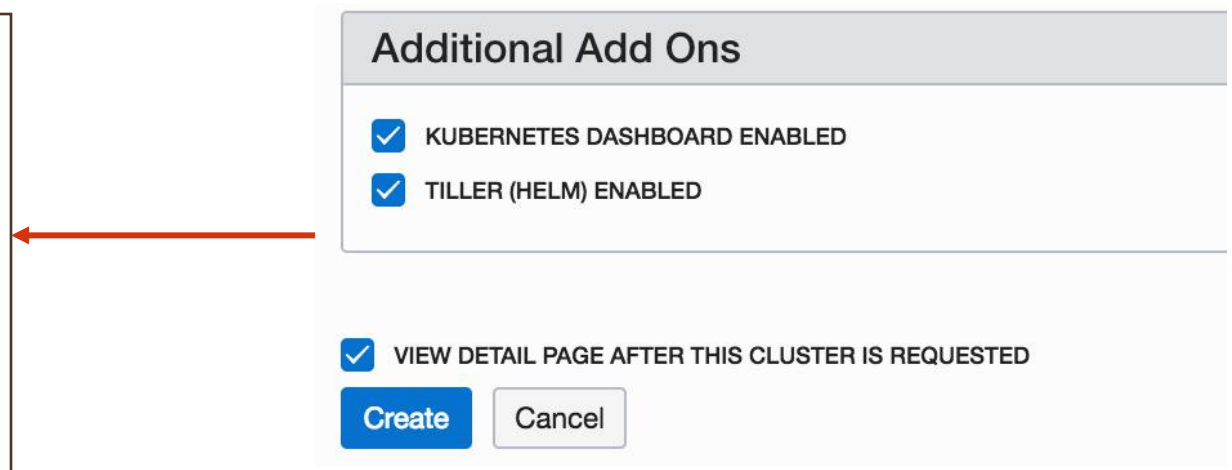
Kubernetes Labels: One or more labels (in addition to a default label) to add to worker nodes in the node pool to enable the targeting of workloads at specific node pools.

OKE Quickstart (contd…)

Step 2: Cluster Creation

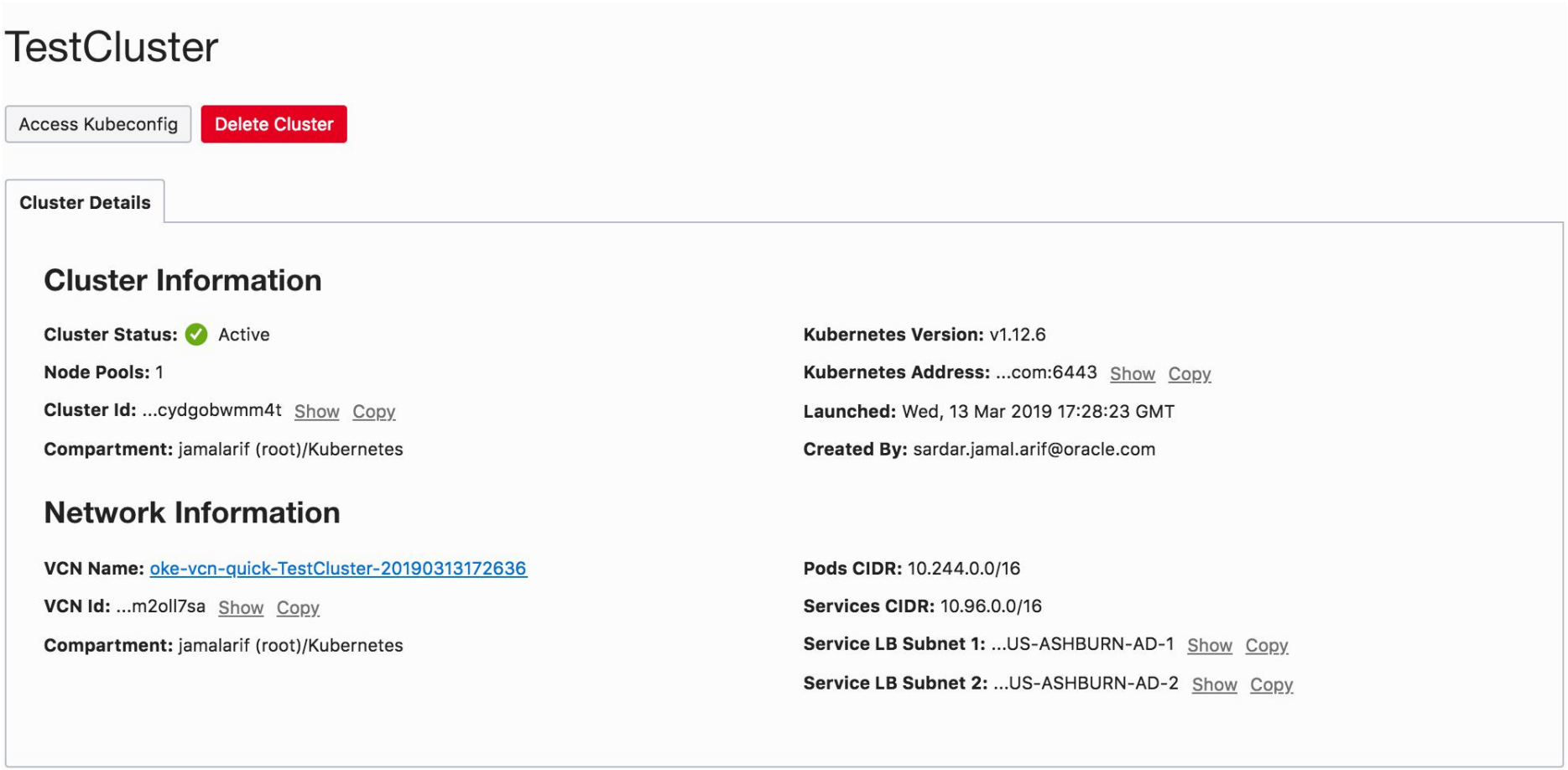
Kubernetes Dashboard Enabled: Select if you want to use the Kubernetes Dashboard to deploy and troubleshoot containerized applications, and to manage Kubernetes resources. See Starting the Kubernetes Dashboard.

Tiller (Helm) Enabled: Select if you want Tiller (the server portion of Helm) to run in the Kubernetes cluster. With Tiller running in the cluster, you can use Helm to manage Kubernetes resources.



K8s Cluster in minutes ..

Cluster details



K8s Cluster in minutes ..

Node Pool details

Node Pools

Add Node Pool

To access private nodes with a public SSH key, set up a bastion host. Dismiss

pool1 Actions

Details Labels

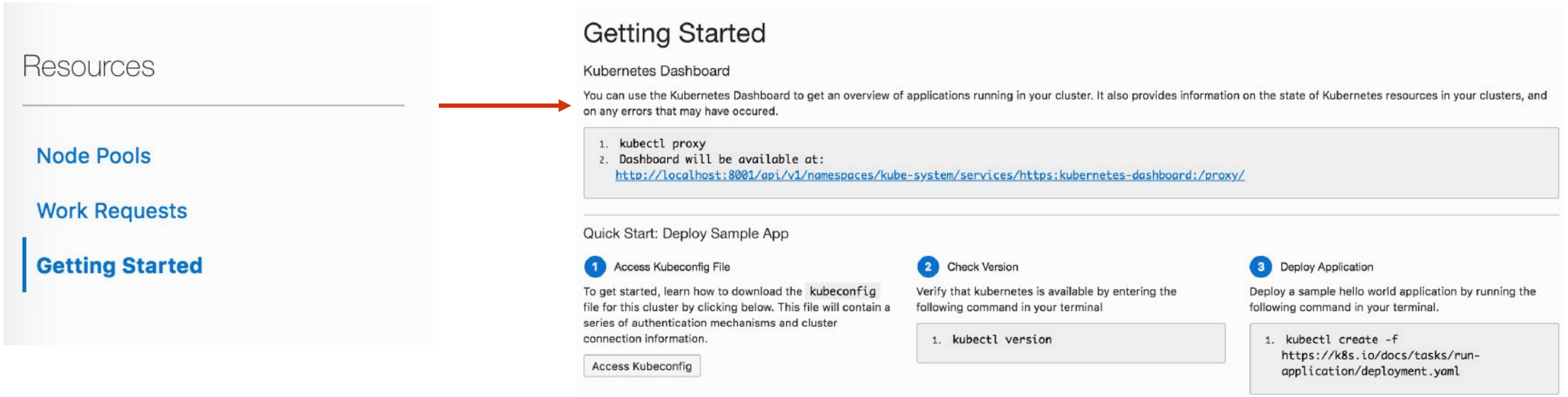
Kubernetes Ver: v1.12.6
Image Name: Oracle-Linux-7.5
Nodes Per Subnet: 1
Shape: VM.Standard1.2
Total Worker Nodes: 3
Number of Subnets: 3
Node Pool Id: ...ayzsg43g Show Copy

Hide Node Details

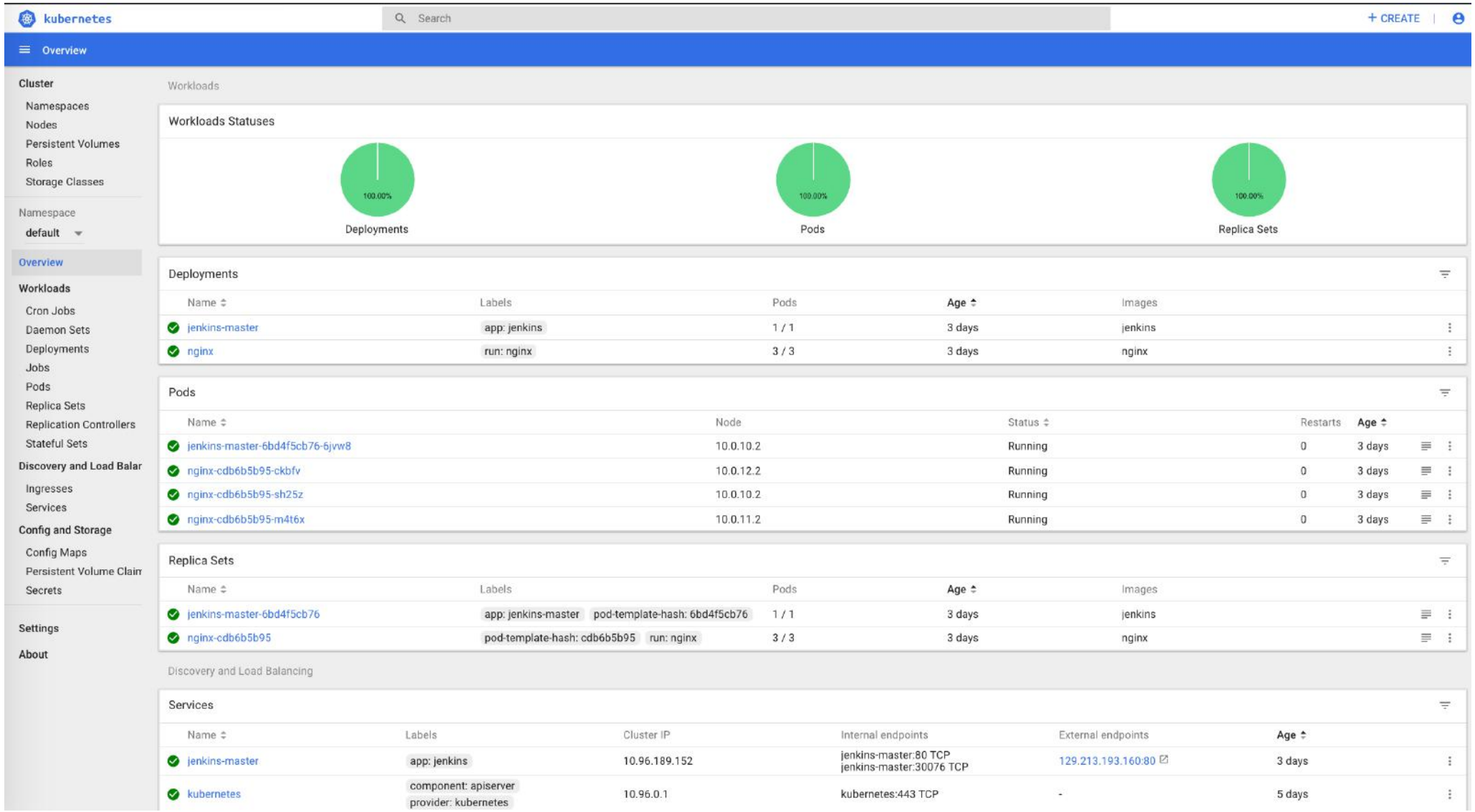
Instance Name ▲	Compute Node State	Subnet	Public IP	Kubernetes Ver
oke-cydgobwmm4t-nzdayzsg43g-shkj2fwiyfq-0	ACTIVE	oke-subnet-quick-TestCluster-20190313172636-fyhg:US-ASHBURN-AD-1 (Private)	*Unavailable*	v1.12.6
oke-cydgobwmm4t-nzdayzsg43g-suzpischafa-0	ACTIVE	oke-subnet-quick-TestCluster-20190313172636-fyhg:US-ASHBURN-AD-3 (Private)	*Unavailable*	v1.12.6
oke-cydgobwmm4t-nzdayzsg43g-szhfpwjjuqa-0	ACTIVE	oke-subnet-quick-TestCluster-20190313172636-fyhg:US-ASHBURN-AD-2 (Private)	*Unavailable*	v1.12.6

Showing 3 Item(s)

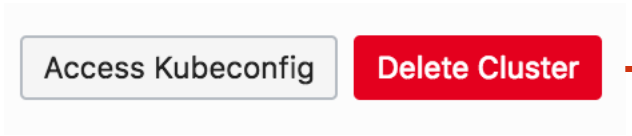
Accessing the K8s Cluster - Dashboard

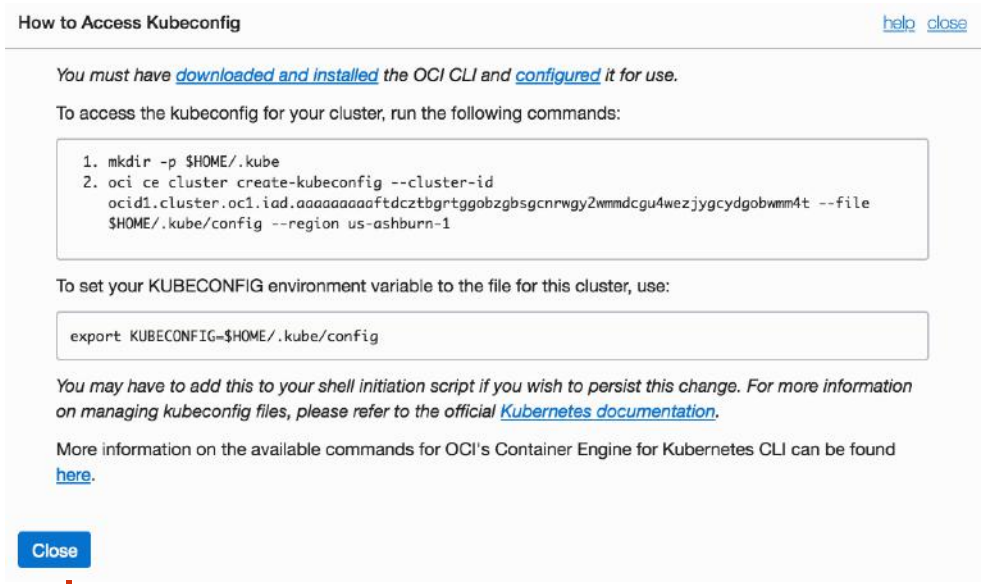


Accessing the K8s Cluster - Dashboard



Accessing the K8s Cluster with kubectl






```
sararif-mac:~ sararif$ kubectl cluster-info
Kubernetes master is running at https://cydgobwmm4t.us-ashburn-1.clusters.oci.oraclecloud.com:6443
KubeDNS is running at https://cydgobwmm4t.us-ashburn-1.clusters.oci.oraclecloud.com:6443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
sararif-mac:~ sararif$ kubectl get nodes -o wide
NAME        STATUS    ROLES     AGE       VERSION   EXTERNAL-IP   OS-IMAGE                 KERNEL-VERSION                  CONTAINER-RUNTIME
10.0.10.2   Ready     node      5d        v1.12.6   <none>        Oracle Linux Server 7.5   4.14.35-1818.2.1.el7uek.x86_64   docker://18.9.1
10.0.11.2   Ready     node      5d        v1.12.6   <none>        Oracle Linux Server 7.5   4.14.35-1818.2.1.el7uek.x86_64   docker://18.9.1
10.0.12.2   Ready     node      5d        v1.12.6   <none>        Oracle Linux Server 7.5   4.14.35-1818.2.1.el7uek.x86_64   docker://18.9.1
sararif-mac:~ sararif$
```

https://kubernetes.io/docs/reference/kubectl/kubectl/

DEMO

http://bit.ly/30cIn3l

Summary

- OCI Container engine for Kubernetes is a managed Kubernetes service
- K8s service is itself free, you only for the resources you use for your worker nodes
- Create a highly available Kubernetes cluster using quickstart in minutes on OCI

ORACLE

Oracle Cloud always free tier:
oracle.com/cloud/free/

OCI training and certification:
https://www.oracle.com/cloud/iaas/training/
https://www.oracle.com/cloud/iaas/training/certification.html
education.oracle.com/oracle-certification-path/pFamily_647

OCI hands-on labs and Terraform Modules:
ocitraining.qloudable.com/provider/oracle

Oracle learning library videos on YouTube:
youtube.com/user/OracleLearning