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

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

**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

Not subject to restriction.
Container Engine for Kubernetes - OKE
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 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
Kubernetes Challenges

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

Percentages reported by companies with >1,000 containers
Working with OKE and OCIR on OCI

Oracle Registry (OCIR)
- In-flight and at rest data encryption

OCI Container Engine for Kubernetes (OKE)
- Cluster Management
- HA - 3 Masters/etcd across 3 ADs
- Container Engine Dashboard

Customer’s OCI Account/Tenancy
- VM based Clusters and Nodes
- Bare Metal Clusters and Nodes

Oracle Cloud Infrastructure

Oracle Managed

Customer Managed
OKE/OCIR Pricing and Packaging

- **OCI Registry**: Free
- **OCI Container Engine for Kubernetes**: Free
- **Oracle Cloud Infrastructure**: Free
- **Cluster Management**: Pay only for the OCI resources used
to run your K8s clusters (VM's, Storage, LB, etc.)

**Oracle Managed**

**Customer Managed**
# 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

**WebLogic Application**

**WebLogic Server**

**Dockerfile**

Containerize WebLogic

**WebLogic Application**

**WebLogic Server**

**WebLogic**

Data Store (ex. Oracle Database)

**Containerize**

Define build for CI/CD toolchain

**Container Pipelines, Jenkins, etc.**

Build

Test

Push

Push Docker image to Registry

Cloud Infrastructure Registry

Pull WebLogic and Operator images from Registry

Container Engine for Kubernetes

**Migrate data store**

**Autonomous Transaction Processing**

**WebLogic Application + Server**

**WebLogic Operator managing WebLogic Domains**

**ORACLE CLOUD INFRASTRUCTURE**
Containers Use Case: Refactor an Existing Application

- **User Interface**
- **App Server + Data Access**
- **Data Store**

Microservices:

- **User Interface**
- **App Server + Data Access**
- **Monolith Application**
- **Data Store**

Re-factor app

Push Code to CI/CD toolchain

Container Pipelines, Jenkins, etc.

Push Docker images to Registry

Cloud Infrastructure Registry

Pull images from Registry

Container Engine for Kubernetes

Deploy images to production

Containers running microservices

Kubernetes worker nodes

ORACLE CLOUD INFRASTRUCTURE
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
Step 2: Cluster Creation

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

TestCluster

Access Kubeconfig  Delete Cluster

Cluster Information

Cluster Status: Active
Node Pools: 1
Cluster Id: -cdgobwmm4t  Show  Copy
Compartment: jamalarif (root)/Kubernetes

Kubernetes Version: v1.12.6
Kubernetes Address: ...com:6443  Show  Copy
Created By: sardar.jamal.arif@oracle.com

Network Information

VCN Name: oke-vcn-quick-TestCluster-20190313172836
VCN Id: -m2oll7ea  Show  Copy
Compartment: jamalarif (root)/Kubernetes

Pods CIDR: 10.244.0.0/16
Services CIDR: 10.96.0.0/16
Service LB Subnet 1: ...US-ASHBURN-AD-1  Show  Copy
Service LB Subnet 2: ...US-ASHBURN-AD-2  Show  Copy
# 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](#).

<table>
<thead>
<tr>
<th>Pool</th>
<th>Details</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Node Pools**

<table>
<thead>
<tr>
<th>Kubernetes Ver</th>
<th>Image Name</th>
<th>Nodes Per Subnet</th>
<th>Number of Subnets</th>
</tr>
</thead>
<tbody>
<tr>
<td>v1.12.6</td>
<td>Oracle-Linux-7.5</td>
<td>1</td>
<td>3</td>
</tr>
</tbody>
</table>

**Node Pool Id:** ...ayzsg43g

**Shape:** VM.Standard1.2

<table>
<thead>
<tr>
<th>Instance Name</th>
<th>Compute Node State</th>
<th>Subnet</th>
<th>Public IP</th>
<th>Kubernetes Ver</th>
</tr>
</thead>
<tbody>
<tr>
<td>oke-cvlgobwmm4t-nzdavzsg43g-shs27wiyf6-0</td>
<td>ACTIVE</td>
<td>oke-subnet-quick-TestCluster-20190313172636-fyhg-US-ASHBURN-AD-1 (Private)</td>
<td>Unavailable</td>
<td>v1.12.6</td>
</tr>
<tr>
<td>oke-cvlgobwmm4t-nzdavzsg43g-suzpischaf6-0</td>
<td>ACTIVE</td>
<td>oke-subnet-quick-TestCluster-20190313172636-fyhg-US-ASHBURN-AD-3 (Private)</td>
<td>Unavailable</td>
<td>v1.12.6</td>
</tr>
</tbody>
</table>

Showing 3 Item(s)
Accessing the K8s Cluster - Dashboard

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 occurred.

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
Accessing the K8s Cluster - Dashboard

- Workloads
  - Deployments
    - jenkins-master
      - app: jenkins
        - Pods: 1/1
        - Age: 3 days
        - Images: jenkins
      - nginx
        - pods: 3/3
        - Age: 3 days
        - Images: nginx
  - Replica Sets
    - jenkins-master-6bbd4f5c76
      - Node: 10.0.10.2
      - Status: Running
      - Restarts: 0
      - Age: 3 days
    - nginx-cdb6b1d5b6-sxbfv
      - Node: 10.0.12.2
      - Status: Running
      - Restarts: 0
      - Age: 3 days
    - nginx-cdb6b1d5b6-sh035c
      - Node: 10.0.10.2
      - Status: Running
      - Restarts: 0
      - Age: 3 days
    - nginx-cdb6b1d5b6-m45x
      - Node: 10.0.11.2
      - Status: Running
      - Restarts: 0
      - Age: 3 days
  - Replica Sets
    - jenkins-master-6bbd4f5c76
      - app: jenkins-master
        - pods: 1/1
        - Age: 3 days
        - Images: jenkins
      - pod-template-hash: cdbbb5b95
        - pods: 3/3
        - Age: 3 days
        - Images: nginx

- Discovery and Load Balancing
  - Services
    - jenkins-master
      - Labels: app: jenkins
        - Cluster IP: 10.96.159.152
        - Internal endpoints: jenkins-master:80 TCP
          - jenkins-master:300076 TCP
        - External endpoints: 129.213.193.160 80
        - Age: 3 days
      - Kubernetes
        - Component: api-server
          - Provider: kubelets
          - Cluster IP: 10.96.0.1
          - Services: kubelets:443 TCP
        - Age: 5 days
Accessing the K8s Cluster with kubectl

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 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