



Application Workloads to the Oracle Cloud with Kubernetes



ORACLE
ACE

Michel Schildmeijer
Solutions Architect
Qualogy

MAY 15 & 16, 2019

CLEVELAND PUBLIC AUDITORIUM, CLEVELAND, OHIO

WWW.NEOOUG.ORG/GLOC

Me

- From Amsterdam, the Netherlands
- Solutions Architect at Qualogy
- Oracle ACE since 2012
- Authored 2 books about WebLogic in 2011
- Started with UNIX in 1994
- Started with BEA technology in 2000
- Started in 2017 looking at containerization strategies, DevOps etc.

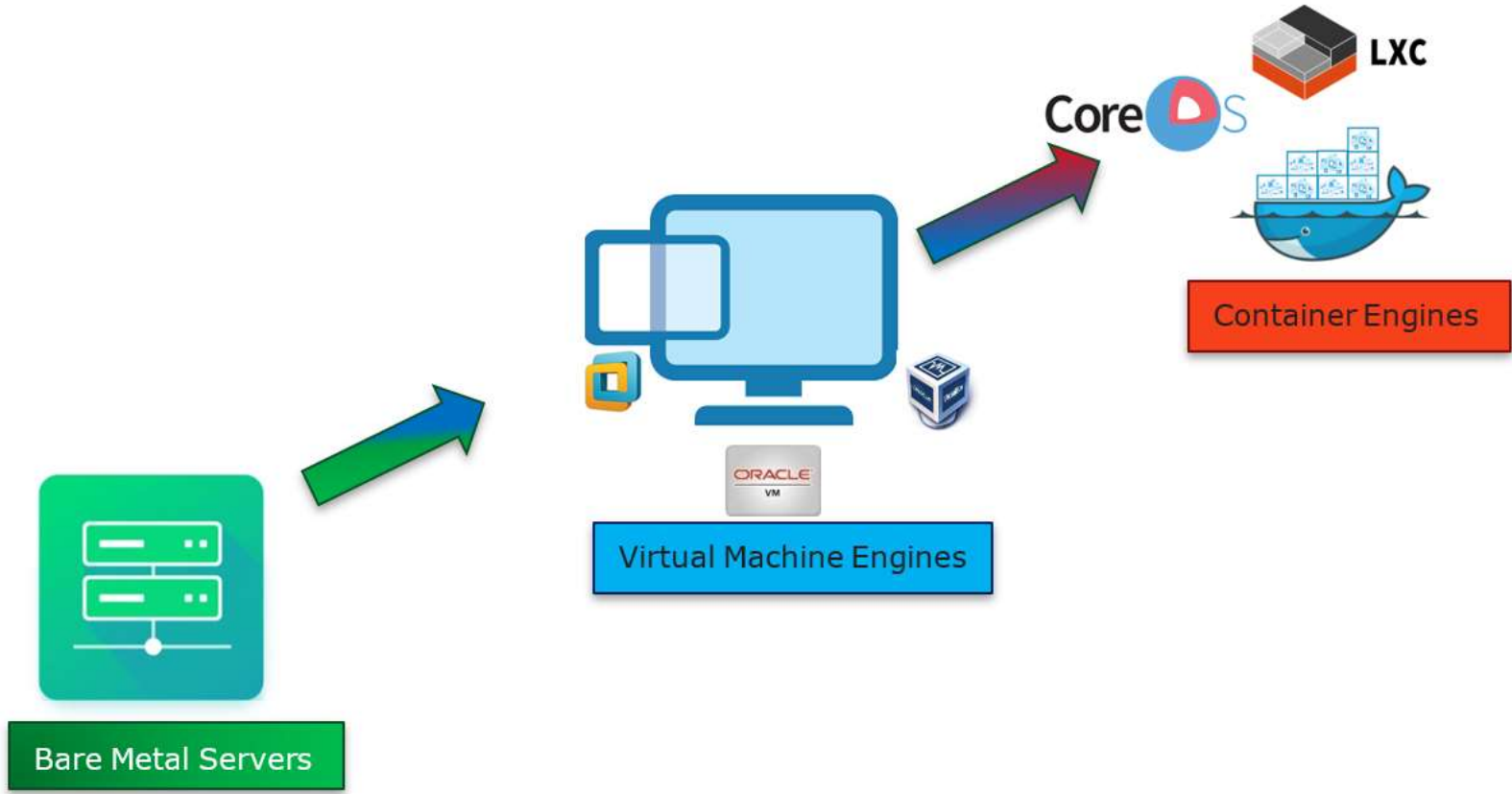


1



Oracle adopts Kubernetes

Evolution



Containers

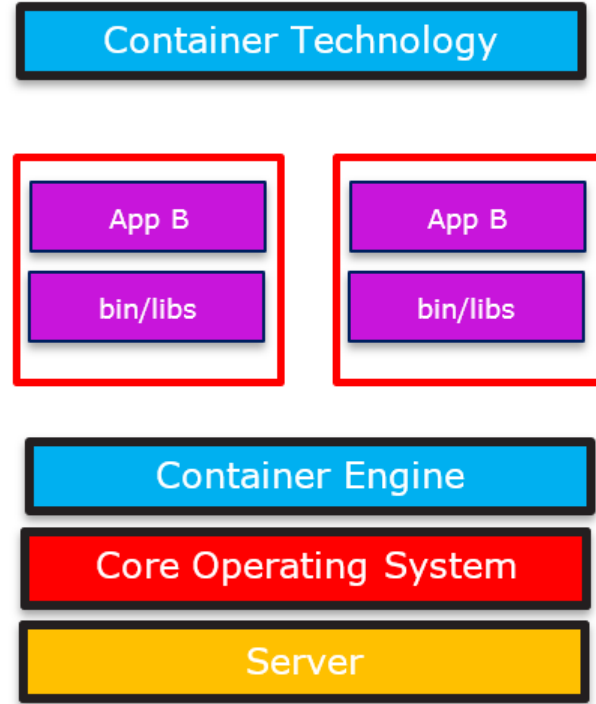
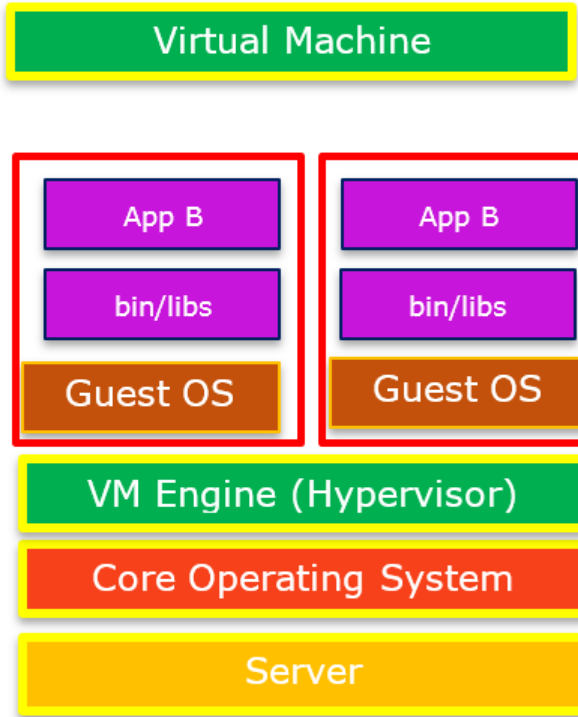
- Application and Infrastructure software more as a whole
 - Bare Metal Server: Hardware, without virtualization
 - Virtual Machine: Virtualized hardware system with Guest OS
 - Virtualize the OS
 - Container: Configurable unit for small set of services/applications
 - Share the OS kernel – No hypervisor
 - Isolate resources
- Container based technology:
 - Docker –
Runs as a process, is based on Linux Containers, doesn't run an independent version of the OS kernel.
 - CoreOS – Rocket
Using an opensource Container Image, now part of RedHat's OpenShift
 - LXC – Linux Containers
Based on chroots

All supported by the Open Container Initiative (OCI)

Use cases to get containerized

Case	Container Use Case
Development	Developer productivity; Consistent; app stacks in Dev, Test & Production
CI/CD	Containerized dependencies; Container registries;
Operations	Standardized environments for dev, testing and operations
Legacy	Refactor from applications tier to portable containerized applications
Cloud Strategy	Move entire application landscape and see them run identically in the cloud
Microservice Strategy	Break monoliths into scalable custom applications

Containers vs Virtual Machines

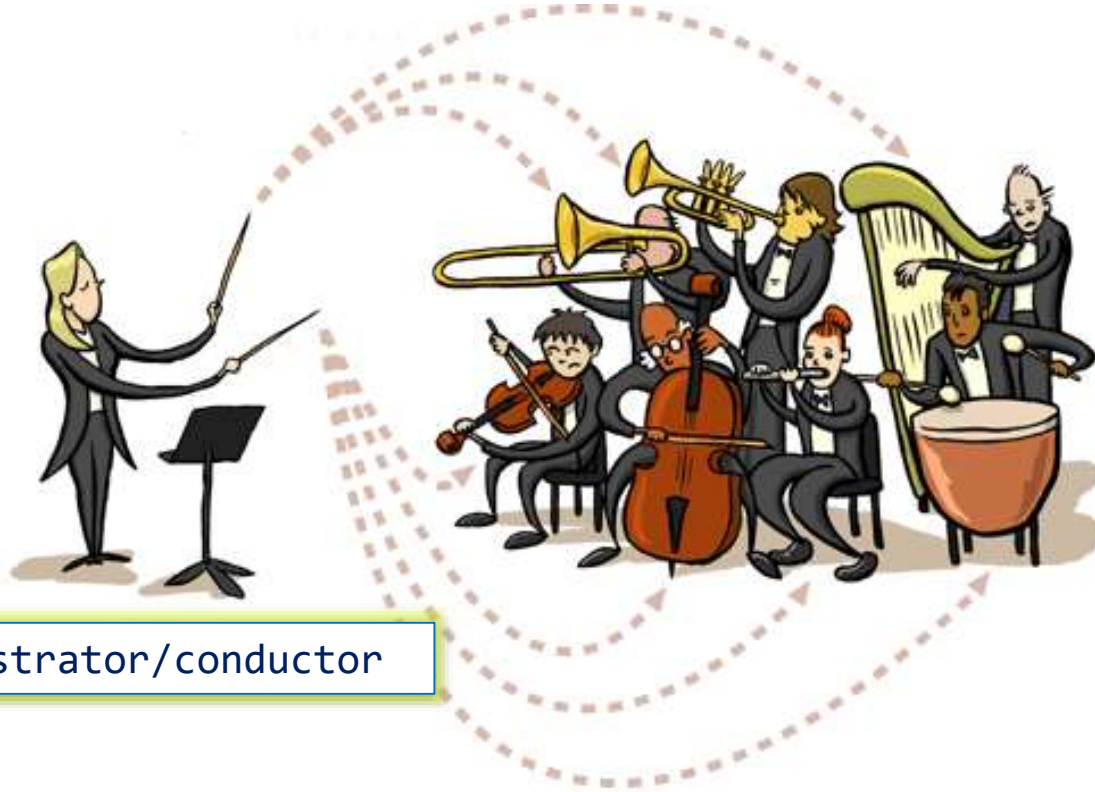


Containermanagement

- There can be a large number of containers
 - who will manage and structure these?

Storage, compute resources, networking

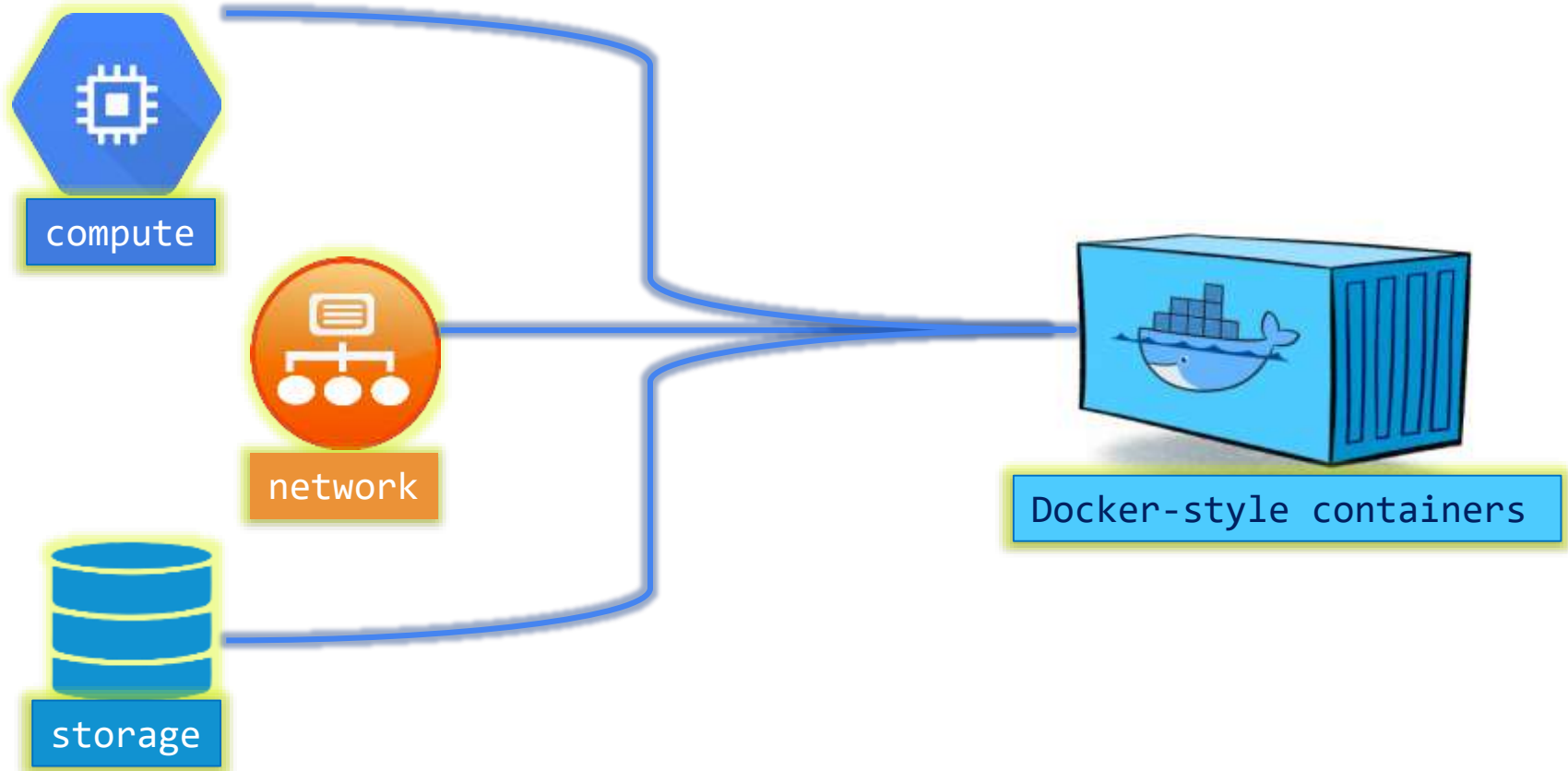
The Orchestrator



Orchestrator/conductor

Kubernetes

- Manages & orchestrates container processes (docker)



Some basics and entities

Master: Controls Kubernetes nodes.

Node: Perform requested, assigned tasks by the master

Pod: A group of one or more containers deployed to a single node.

Replication controller: Control of identical pods across a cluster.

Service: Work definitions from the pods.

Kubelet: Reads container manifests to watch if containers are started and running.

kubectl: command line configuration tool for Kubernetes.

Cloud Native Com Launches Certified Program with 32 c

• Oracle, Oracle Container Engine, Oracle Linux, Oracle Terraform Kubernetes Installer

By [cnf](#) | November 13, 2017 | [Announcement](#)

Oracle certified for CNCF, Kubernetes

- Alibaba Cloud, Alibaba Cloud Container Service
- Apprenda, Kismatic Enterprise Toolkit (KET)
- Appscore, Pharmer
- Caicloud, Compass
- Canonical, Canonical Distribution of Kubernetes
- Cisco Systems, Cisco Container Platform
- Cloud Foundry Foundation, Cloud Foundry Container Runtime
- CoreOS, Tectonic
- DaoCloud, DaoCloud Enterprise
- Docker, Docker Enterprise Edition
- Google, Google Kubernetes Engine
- Heptio, AWS-Quickstart
- Huawei, Huawei Cloud Container Engine
- IBM, IBM Cloud Container Service and IBM Cloud Private
- Loodse, Kubermatic Container Engine
- Mesosphere, Kubernetes on DC/OS

- Netease, Netease Container Service Dedicated
- Oracle, Oracle Container Engine, Oracle Linux, Oracle Terraform Kubernetes Installer
- Pivotal Software, Pivotal Container Service (PKS)
- Poseidon, Typhoon
- Rancher, Inc., Rancher Kubernetes
- Red Hat, OpenShift
- SAP, Cloud Platform – Gardener
- SUSE, SUSE CaaS (Container as a Service) Platform
- Samsung SDS, Kraken
- StackPointCloud, Stackpoint.io
- Tencent Cloud, Tencent Cloud Container Service
- VMware, Pivotal Container Service (PKS)
- Weaveworks, kubeadm
- Wise2C Technology, WiseCloud



2



Oracle & Kubernetes
“do it yourself”

If the image, Virtual Cloud Network, or Subnet is in a different Compartment than the Instance, enable Compartment selection for those resources: [Click here](#).

Instance

NAME

AVAILABILITY DOMAIN

BOOT VOLUME

ORACLE-PROVIDED OS IMAGE CUSTOM IMAGE BOOT VOLUME IMAGE OCID

IMAGE OPERATING SYSTEM

The image will be booted using native mode.

SHAPE TYPE

VIRTUAL MACHINE BARE METAL MACHINE

SHAPE

Shape compatibility based on selected operating system.

IMAGE VERSION

≡ MENU

Compute

Instances

Kubernetes “do it yourself”

- On premises
- Addon yum repo's
- Enable docker and login to Oracle container registry

```
[root@qkube01-occs-wkr-1 ~]# docker login container-registry-ash.oracle.com
Username: mschildmeijer@qualogy.com
Password:
Login Succeeded
[root@qkube01-occs-wkr-1 ~]# █
```

Add-On yum repo's

```
cat <<EOF > /etc/yum.repos.d/kubernetes.repo > [kubernetes]
>name=Kubernetes
>baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86_64
> enabled=1 > gpgch
> gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg
>          https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg
> EOF
```

If not using Oracle Linux

```
[ol7_addons]
name=Oracle Linux $releasever Add ons ($basearch)
baseurl=http://yum.oracle.com/repo/OracleLinux/OL7/addons/$basearch/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
gpgcheck=1
enabled=1
```

Add Kubernetes repo to yum

Kubernetes “do it yourself”

- Set to Kubernetes system variables

```
# export KUBE_REPO_PREFIX=container-registry-ash.oracle.com/kubernetes  
# echo 'export KUBE_REPO_PREFIX=container-registry-ash.oracle.com/kubernetes' > ~/.bashrc
```

- yum install

```
yum install kubeadm
```

- kubeadm init sets the basics

```
# kubeadm init
```

- Or run with script kubeadm-setup.sh
 - Provided by Oracle with the kubeadm package
- With terraform (see 5)

kubeadm_setup.s

```
Checking iptables default rule ...
Checking br_netfilter module ...
Checking sysctl variables ...
Check successful, ready to run 'up' command ...
Waiting for kubeadm to setup master cluster...
Please wait ...
- - 75% completed
Waiting for the control plane to become ready ...
Starting to initialize .....
Checking if env is ready 100% completed
Checking whether docker clusterrole "flannel" created
Checking access to configmap "kube-flannel-cfg" created
v1.9.1-1: Pulling from kube-flannel-cfg: kube-flannel-cfg
Digest: sha256:f525d6... daemonset "kube-flannel-ds" created
Status: Image is up to date
Checking whether docker Installing kubernetes-dashboard ...
Checking iptables default rule ...
Checking br_netfilter module ...
Checking sysctl variables ...
Check successful, ready to run 'up' command ...
Waiting for kubeadm to setup master cluster...
Please wait ...
- - 75% completed
writing new private key to 'dashboard.key'
-----
No value provided for Subject Attribute C, skipped
No value provided for Subject Attribute ST, skipped
No value provided for Subject Attribute L, skipped
No value provided for Subject Attribute O, skipped
No value provided for Subject Attribute OU, skipped
Signature ok
subject=/CN=kubernetes-dashboard
Getting Private key
```

```
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
export KUBECONFIG=$HOME/.kube/config
echo 'export KUBECONFIG=$HOME/.kube/config' >> $HOME/.bashrc
```

```
kubectl get pods -n kube-system
```

NAME	READY	STATUS	RESTARTS	AGE
etcd-docker.oracleworld.com	1/1	Running	0	13m
kube-apiserver-docker.oracleworld.com	1/1	Running	0	13m
kube-controller-manager-docker.oracleworld.com	1/1	Running	0	13m
kube-dns-855949bbf-9zshx	3/3	Running	0	14m
kube-flannel-ds-7tgng	1/1	Running	0	13m
kube-proxy-qwnll	1/1	Running	0	14m
kube-scheduler-docker.oracleworld.com	1/1	Running	0	13m
kubernetes-dashboard-7c966ddf6d-tj5tc	1/1	Running	0	13m

```
kubectl -n kube-system describe $(kubectl -n kube-system \
> get secret -n kube-system -o name | grep namespace) | grep
token
```

Kubernetes Dashboard

Authentication method:

Kubeconfig

Token

Token*

|

SIGN IN

SKIP



3



Kubernetes
for Oracle Container
Engine (OKE)

- Other than “Classic”, initiated for OCCS
- OCCS is Docker with a Oracle Cloud frontend
- Not Kubernetes managed

Dashboard

Dashboard

Search

Tasks & Events

Services

Stacks

Deployments

Containers

Images

Hosts

Resource Pools

Registries

Tags

Service Discovery

Overview

2

Nodes

0
Deployments

OK

There are no running deployments

1
Hosts

OK

All hosts are active and reachable.

1 Active

3
Resource Pools

3 Healthy

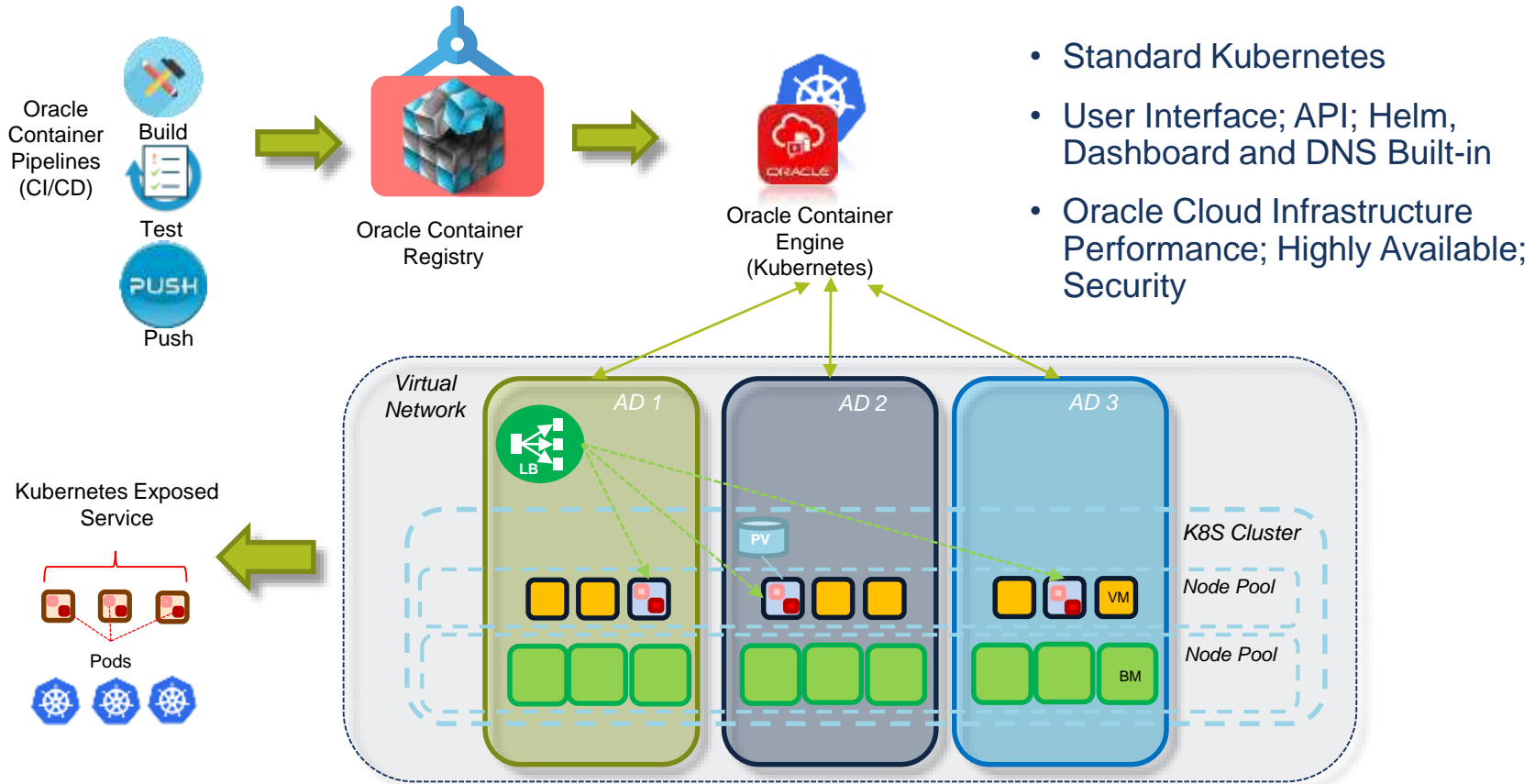
Name	Hosts	Memory	CPU
<u>default</u>	1	3%	0%
<u>Development</u>	0	0%	0%
<u>Production</u>	0	0%	0%

Services	25
Stacks	3
Deployments	0
Resource Pools	3

Containers	0
Images	0
Hosts	1

Oracle Kubernetes Engine

- Configured Kubernetes Engine in the Cloud
- Oracle Cloud Infrastructure (OCI)
- Setup with Cloud console
- Setup with oci interface
- Setup with Terraform
- The “easy way”



Oracle Kubernetes Engine

The image shows the Oracle Cloud Infrastructure (OCI) console interface. On the left, there is a navigation menu with a 'MENU' button and a 'COMPARTMENT' dropdown set to 'Select a Compartment'. The main content area is titled 'Compute' and contains a 'Compartment Filter' dropdown and a text block explaining that OCI uses Compartments for resource management, with a link to 'Learn more about Compartments'. A dark overlay menu is open, listing various services under three categories: CORE INFRASTRUCTURE, DATABASE, and SOLUTIONS, PLATFORM AND EDGE. The 'Container Clusters (OKE)' and 'Registry (OCIR)' items are highlighted with a red box. The 'Compute' page content is partially visible behind the menu.

Navigation Menu:

- CORE INFRASTRUCTURE
 - Compute
 - Block Storage
 - Object Storage
 - File Storage
 - Networking
- DATABASE
 - Bare Metal, VM, and Exadata
 - Autonomous Data Warehouse
 - Autonomous Transaction Processing
- SOLUTIONS, PLATFORM AND EDGE
 - Email Delivery
 - Edge Services
 - Developer Services

Highlighted Items:

- Container Clusters (OKE)
- Registry (OCIR)

Main Content Area:

Compute

Compartment

The Oracle Cloud Infrastructure uses Compartments to organize resources. To view and manage your resources, select a Compartment. [Learn more about Compartments](#)

resources.

type using the filters on the left.

Oracle Kubernetes Engine

- Redirect to your local client
- Download & configure oci for management

```
bash -c "$(curl -L https://raw.githubusercontent.com/oracle/oci-cli/master/scripts/install/install.sh)"
```

```
mkdir -p $HOME/.kube  
oci ce cluster create-kubeconfig --cluster-id  
ocid1.cluster.oc1.eu-frankfurt-  
1.aaaaaaaaaftggnbqgq2dam3fmi4tamrvmu3tkyrugmytgyzqgcrtinzzmmzw  
--file $HOME/.kube/config
```



ACTIVE

QOKE1

[Access Kubeconfig](#)
[Delete Cluster](#)

Cluster Details

Cluster Status: ✔ Active

Node Pools: 2

Cluster Id: ...[Show](#) [Copy](#)
Launched: Fri, 28 Sep 2018 14:07:22 GMT

Services CIDR: ...

Kubernetes Version: v1.11.1

Kubernetes Address: ...com:6443 [Show](#) [Copy](#)
VCN Name: [kubernetes](#)
VCN Id: ...[Show](#) [Copy](#)
Pods CIDR: ...

Resources

[Node Pools](#)
[Work Requests](#)
[Getting Started](#)

Node Pools

[Add Node Pool](#)

 qoke-worker1 ⓘ
[Details](#)
[Labels](#)
Kubernetes Ver: v1.11.1

Shape: VM.Standard2.1

Node Pool Id: ...[Show](#) [Copy](#)
Image Name: Oracle-Linux-7.5

Total Worker Nodes: 2

Nodes Per Subnet: 1

Number of Subnets: 2

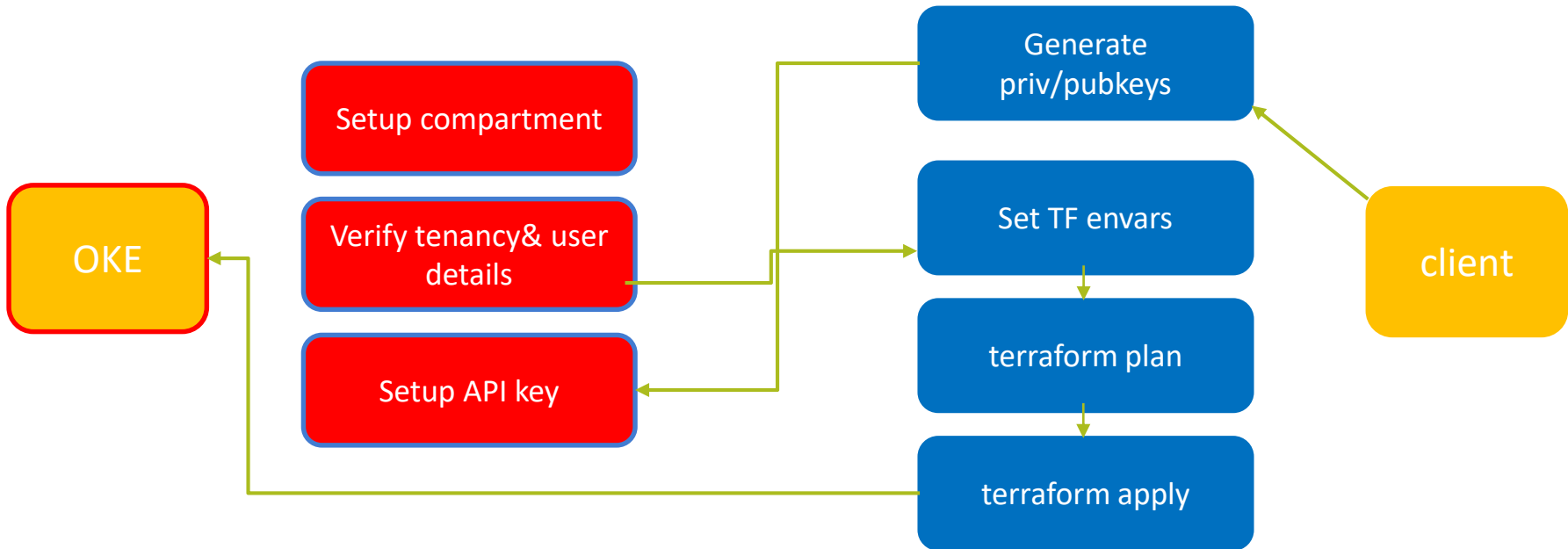


5



Kubernetes, OCI &
Terraform

- Configured Kubernetes Engine
- Oracle Cloud Infrastructure (OCI)
- Setup with terraform



Identity

Users

Users

Groups

Dynamic Groups

Policies

Compartments

Federation

MENU ORACLE
Cloud Infrastructure

Identity » Users » User Details



ACTIVE



ACTIVE

mschildmeijer@qualogy.com

Description: Admin

Created: Wed, 29 Aug 2018 19:43:55 GM

```
[terraform-kubernetes-installer]$ openssl genrsa -out ~/.oci/oci_api_key.pem 2048
Generating RSA private key, 2048 bit long modulus
.....+++
.....+++
e is 65537 (0x10001)

terraform-kubernetes-installer]$ chmod go-rwx ~/.oci/oci_api_key.pem
terraform-kubernetes-installer]$ openssl rsa -pubout -in ~/.oci/oci_api_key.pem -out ~/.oci/oci_api_key_public.pem
```

- Generate keypair and fingerprint for OCI



ACTIVE

Description: Admin

Create/Reset Password

Unblock

Delete

Apply Tag(s)

User Information

Tags

OCID: ...cfdeoq [Show Copy](#)

Status: Active

Created: Wed, 29 Aug 2018 19:43:55 GMT

Resources

API Keys (1)

Auth Tokens (1)

SMTP Credentials (0)

Amazon S3 Compatibility API Keys (0)

Groups (1)

API Keys

Displaying 1 API Keys

Add Public Key



Fingerprint: f6:d3:45:b3:85:fc:d1:b9:41:5b:a4:6f:b2:01:4a:d3

Time Created: Sun, 23 Sep 2018 07:26:42 GMT



TF Envars

OCI authentication

```
tenancy_ocid = "ocid1.tenancy.oc1..aaaaaaaagozl47dkv6gpbkffed5imrm3f6rjiwamxhz5ns7m2wqwfoplhhzq"  
compartment_ocid = "ocid1.compartment.oc1..aaaaaaaamluifb5mxw2tiexvy5m4dga2phffeshb2efk7w7xbf5mngp1sceq"  
fingerprint = "f6:d3:45:b3:85:fc:d1:b9:41:5b:a4:6f:b2:01:4a:d3"  
private_key_path = "/home/oracle/.oci/oci_api_key.pem"  
user_ocid = "ocid1.user.oc1..aaaaaaaawqmr61c2jj3s6gv2kxebmr6vndbz3ykfp653cq2zkz3zocfdeoq"  
Region = "eu-frankfurt-1"
```

CCM user

```
#cloud_controller_user_ocid = "ocid1.tenancy.oc1..aaaaaaa763cu5f3m7qpzwnvr2shs3o26ftrn7fkgz55cpzgxmgltui3v7q"  
#cloud_controller_user_fingerprint = "ed:51:83:3b:d2:04:f4:af:9d:7b:17:96:dd:8a:99:bc"  
#cloud_controller_user_private_key_path = "/tmp/oci_api_key.pem"
```

```
#etcdShape = "VM.Standard1.2"  
k8sMasterShape = "VM.Standard1.8"  
k8sWorkerShape = "VM.Standard1.8"
```

```
#etcdAd1Count = "1"  
#etcdAd2Count = "1"  
#etcdAd3Count = "1"
```

```
#k8sMasterAd1Count = "1"  
#k8sMasterAd2Count = "1"  
#k8sMasterAd3Count = "1"
```

```
#k8sWorkerAd1Count = "1"  
#k8sWorkerAd2Count = "1"  
#k8sWorkerAd3Count = "1"
```

```
etcdLBShape = "400Mbps"  
k8sMasterLBShape = "400Mbps"
```

TF init, plan and apply

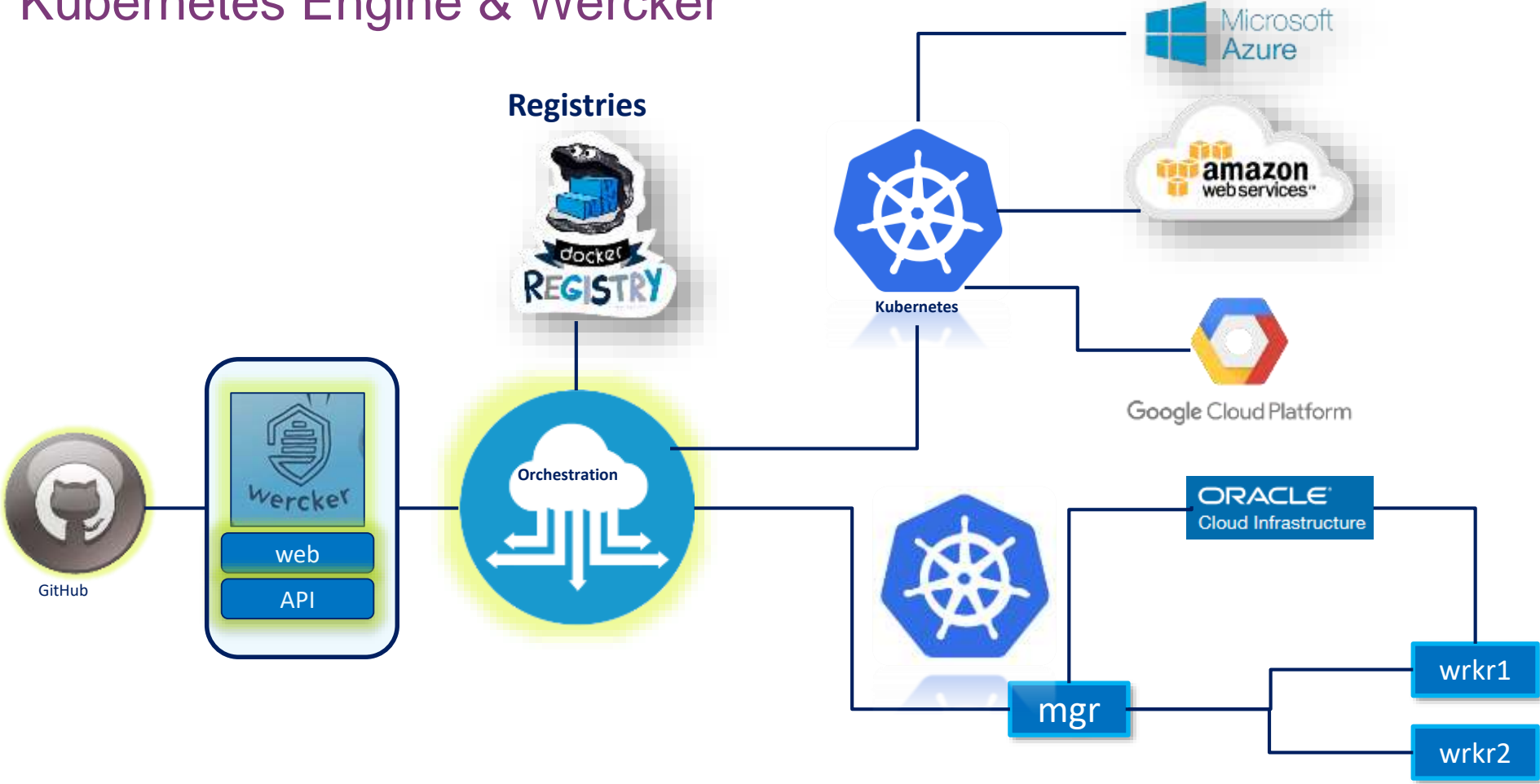
```
+ module.vcn.oci_core_subnet.k8sWorkerSubnetAD3
  id: <computed>
  availability_domain: "tvm0:EU-FRANKFURT-1-AD-3"
  cidr_block: "10.0.42.0/24"
  compartment_id: "ocid1.compartment.oc1..aaaaaaaamluifb5mxw2tiexvy5m4dga2phffeshb2efk7w7xbf5mnqplsceq"
  dhcp_options_id: "${oci_core_virtual_network.CompleteVCN.default_dhcp_options_id}"
  display_name: "publicK8SWorkerSubnetAD3"
  dns_label: "k8sworkerad3"
  prohibit_public_ip_on_vnic: "false"
  route_table_id: "${var.control_plane_subnet_access == \"private\" ? coalesce(join(\" \", oci_core_route_table.NATInstanceAD3RouteTable.NATInstanceAD1RouteTable.*.id), join(\" \", oci_core_route_table.NATInstanceAD2RouteTable.*.id), oci_core_route_table.PublicRouteTable.id) : oci_core_route_table.PublicRouteTable.id}"
  security_list_ids.#: <computed>
  state: <computed>
  subnet_domain_name: <computed>
  time_created: <computed>
  vcn_id: "${oci_core_virtual_network.CompleteVCN.id}"
  virtual_router_ip: <computed>
  virtual_router_mac: <computed>

+ module.vcn.oci_core_virtual_network.CompleteVCN
  id: <computed>
  cidr_block: "10.0.0.0/16"
  compartment_id: "ocid1.compartment.oc1..aaaaaaaamluifb5mxw2tiexvy5m4dga2phffeshb2efk7w7xbf5mnqplsceq"
  default_dhcp_options_id: <computed>
  default_route_table_id: <computed>
  default_security_list_id: <computed>
  display_name: "kubernetes"
  dns_label: "kubernetes"
  state: <computed>
  time_created: <computed>
  vcn_domain_name: <computed>
```

Check....

```
[root@k8s-master-ad1-0 ~]# while true; do kubectl get nodes; sleep 10; done  
NAME                                STATUS    ROLES    AGE     VERSION  
k8s-master-ad1-0.k8smasterad1.kubernetes.oraclevcn.com Ready    master   23d    v1.9.6  
k8s-worker-ad1-0.k8sworkerad1.kubernetes.oraclevcn.com Ready    node     23d    v1.9.6
```

Kubernetes Engine & Wercker



Oracle Kubernetes Engine & Wercker: CI CD for cloud

- Wercker
 - CI on Docker and Kubernetes
 - Special kubernetes yml on g
 - Add Pipeline: deploy-to-c
 - Add K8S specifics



KUBERNETES_TOKEN

KUBERNETES_MASTER

```
#Deploy our container from the Docker Hub to Kubernetes
deploy-to-cluster:
  box:
    id: alpine
    cmd: /bin/sh
  steps:
    - bash-template

    - script:
      name: "Visualise Kubernetes config"
      code: cat kubernetes.yml

    - kubectl:
      name: deploy to kubernetes
      server: $KUBERNETES_MASTER
      #username: $KUBERNETES_USERNAME
      token: $KUBERNETES_TOKEN
      insecure-skip-tls-verify: true
      command: apply -f kubernetes.yml
```



6



Helm & Tiller

Helm & Tiller

- Release and Package Management for Kubernetes
- Helm is client, Tiller runs as Server in Kubernetes

Helm & Tiller

- Usually:
- Helm as client on CI CD or any other
- Tiller runs as Server in Kubernetes
- Comparable with YUM for Linux

Helm & Tiller

Easy does it

```
curl https://raw.githubusercontent.com/helm/helm/master/scripts/get >  
get_helm.sh
```


Helm & Tiller

```
helm init
```

```
$HELM_HOME has been configured at /home/oracle/.helm.
```

```
Tiller (the Helm server-side component) has been installed into your  
Kubernetes Cluster.
```

Deployments

Name	Namespace	Labels
 tiller-deploy	kube-system	app: helm name: tiller

Happy Helming!

Helm & Tiller

Installed in OKE



The screenshot shows the Kubernetes dashboard interface. The top navigation bar includes the 'kubernetes' logo, a search bar, and a '+ CREATE' button. The main content area is titled 'Deployments' and contains a table with the following data:

Name	Namespace	Labels	Pods	Age	Images
tiller-deploy	kube-system	app:helm, name:tiller	1 / 1	2 minutes	gcr.io/kubernetes-helm/tiller:v2.11.0

As an Add On

Additional Add Ons

- KUBERNETES DASHBOARD ENABLED
- TILLER (HELM) ENABLED

Add Helm repo - KubeApps

- Add bitnami repo

```
helm repo add bitnami https://charts.bitnami.com/bitnami
helm install --namespace kubeapps -n kubeapps bitnami/kubeapps
```

- RBAC rolebinding

```
kubectl create serviceaccount kubeapps-operator
kubectl create clusterrolebinding kubeapps-operator --clusterrole=cluster-admin --
serviceaccount=default:kubeapps-operator
```

- Accessing KubeApps dashboard

```
echo "Kubeapps URL: http://127.0.0.1:8080"
export POD_NAME=$(kubectl get pods --namespace kubeapps -l "app=kubeapps" -o
jsonpath="{.items[0].metadata.name}")
kubectl port-forward --namespace kubeapps $POD_NAME 8080:8080
```

KubeApps dashboard

The screenshot displays the KubeApps dashboard interface. At the top, there is a blue navigation bar with the Kubeapps logo on the left, followed by navigation links for 'Applications', 'Charts', and 'Service Instances (alpha)'. On the right side of the bar, there is a dropdown menu currently set to 'default', and links for 'Configuration' and 'Logout'.

Below the navigation bar, the main content area is titled 'Charts' and includes a search input field with the placeholder text 'Q search charts...'. The charts are presented in a grid of 16 items, each with a unique icon, a name, a version number, and a 'bitnami' logo in the bottom right corner. The visible charts are:

- apache** (version 2.4.25)
- bitnami-common** (version 0.0.1)
- consul** (version 1.0.6)
- crypto**
- dokuwiki** (version 0.2090422201805030040)
- drupal** (version 8.6.1)
- elasticsearch** (version 6.4.1)
- etcd** (version 3.3.9)
- external-dns** (version 0.5.4)
- ghost** (version 2.14)
- jasperreports** (version 7.10)
- jenkins** (version 2.190.1)

Install a Jenkins chart

```
kubectl create serviceaccount --namespace kube-system tiller
serviceaccount "tiller" created
kubectl create clusterrolebinding tiller-cluster-rule --clusterrole=cluster-admin --
serviceaccount=kube-system:tiller
clusterrolebinding "tiller-cluster-rule" created
kubectl patch deploy --namespace kube-system tiller-deploy -p
'{"spec":{"template":{"spec":{"serviceAccount":"tiller"}}}}'
deployment "tiller-deploy" patched
```

Example:Set the proper accounts and install a Jenkins chart

```
helm install bitnami/jenkins
NAME:      messy-bobcat
LAST DEPLOYED: Wed Oct  3 05:56:18 2018
NAMESPACE: default
STATUS:    DEPLOYED
```

Get the Jenkins URL, un/pwd, exposed by the ingress controller

```
export SERVICE_IP=$(kubectl get svc --namespace default messy-bobcat-jenkins --template
"{{ range (index .status.loadBalancer.ingress 0) }}{{.}}{{ end }}")
echo "Jenkins URL: http://\$SERVICE\_IP/"
Jenkins URL: http://130.61.15.94/
echo Username: user
echo Password: $(kubectl get secret --namespace default messy-bobcat-jenkins -o
jsonpath="{.data.jenkins-password}" | base64 --decode)
```




7

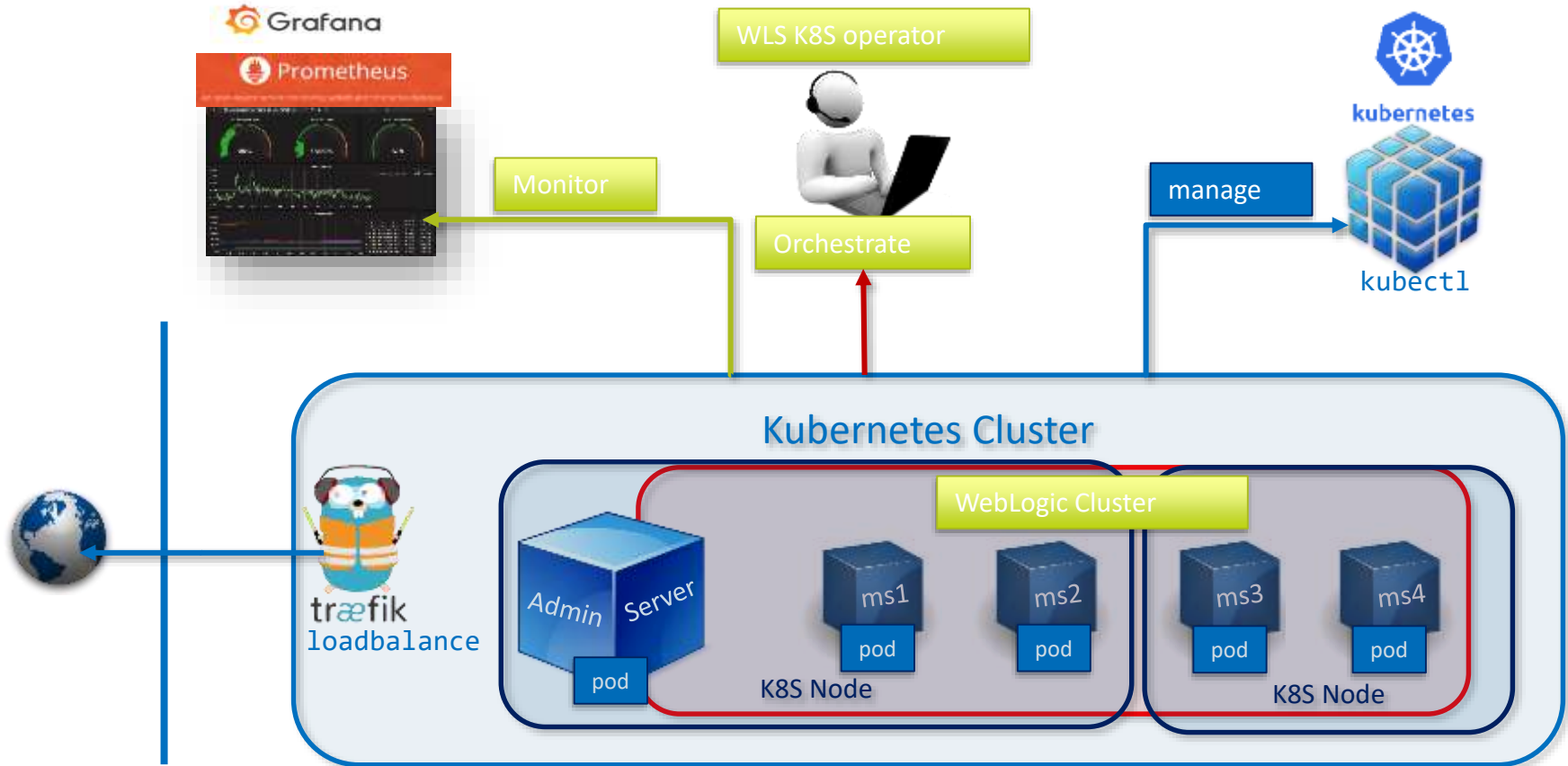


WebLogic Kubernetes

WebLogic Kubernetes Support

- WebLogic and Docker certification
 - Docker images, Dockerfiles
- WebLogic 12.2.1.3 Kubernetes
- Manage, migrate and deploy
 - Management: WLS K8s Operator
 - Monitoring: Exporter for Prometheus
 - Migration: Deploy tooling
- Certification on Oracle Cloud
 - OCI K8S cluster created with Terraform
 - Oracle Kubernetes Engine

WebLogic Kubernetes Support



WebLogic

- WebLogic running in Docker PODs in Kubernetes
 - WebLogic 12.2.1.3+
 - Oracle Linux 7
 - Kubernetes 1.7.5, 1.8.0, 1.9.0, and 1.10.0
 - Docker 17.03-ce
 - Network Fabric - Flannel v0.9.1-amd64
 - Or others like Traefik, Apache
 - Config resources (JDBC,JMS,JTA)
 - Auto scaling (WLDF and Prometheus)

Migrate tooling

- Still in an early phase, but this tool provides some strong capabilities for getting your WebLogic domain migrated to Kubernetes:
- Domain introspection to:
 - Create a model, a yaml file of your entire domain
 - Migrate existing domains and upgrade them to 12.2.1.X
 - Tested on 10.3.6 to 12.2.1.X
- Customize to get “Kubernetes ready”
- Create WebLogic domain docker images

Container Domain Model – on persistent volume

- In each domain the same standard Image
- To preserve state outside the image
- Logfiles and domain artifacts
- For replication purposes in the cloud between av zones

Container Domain Model – on docker image

- Per domain a different Image
- No runtime state, though configuration
- Logfiles and domain artifacts
- Each pod handles its own replica so no need to worry for replication

WebLogic Helm and Tiller

```
[***** ~]$ kubectl create namespace helm-wls-cluster
namespace "helm-wls-cluster" created
[*****~]$ kubectl create secret generic wls-secret --
from-literal=username=ADMIN-USERNAME --from-
literal=password=ADMIN-PASSWORD --namespace helm-wls-
cluster
secret "wls-secret" created
```


WebLogic Helm and Tiller - components

```
total 20
-rw-rw-r-- 1 281 Sep 21 02:10 Chart.yaml
-rw-rw-r-- 1 6556 Sep 21 02:10 README.md
drwxrwxr-x 1 786 Sep 21 02:10 templates
-rw-rw-r-- 1 4442 Sep 21 02:10 values.yaml
```



WebLogic Kubernetes
Operator

Kubernetes Operators

- Extends the K8S api
- Configure & manage more complex instances
- Why Operators ?
 - Stateless (web) apps are easy to manage, scale
 - But tougher when it is with stateful
 - To leverage more experience based knowledge to k8s

The operators from the field in a software driven solution!

Kubernetes Operators

- Some operators:
 - By CoreOS
 - Etcd –Storing data across cluster
 - Prometheus
- Oracle Operators:
 - MySQL Operator
 - WebLogic Operator

WebLogic Kubernetes Operator

WLS K8S operator



Orchestrate



kubernetes



kubect1

manage



Kubernetes Cluster

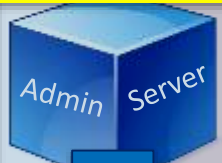
Expose T3 + Admin Console to Ingress

WebLogic Cluster

Manage WLS configured or dynamic clusters



loadbalance



pod



pod



pod



pod



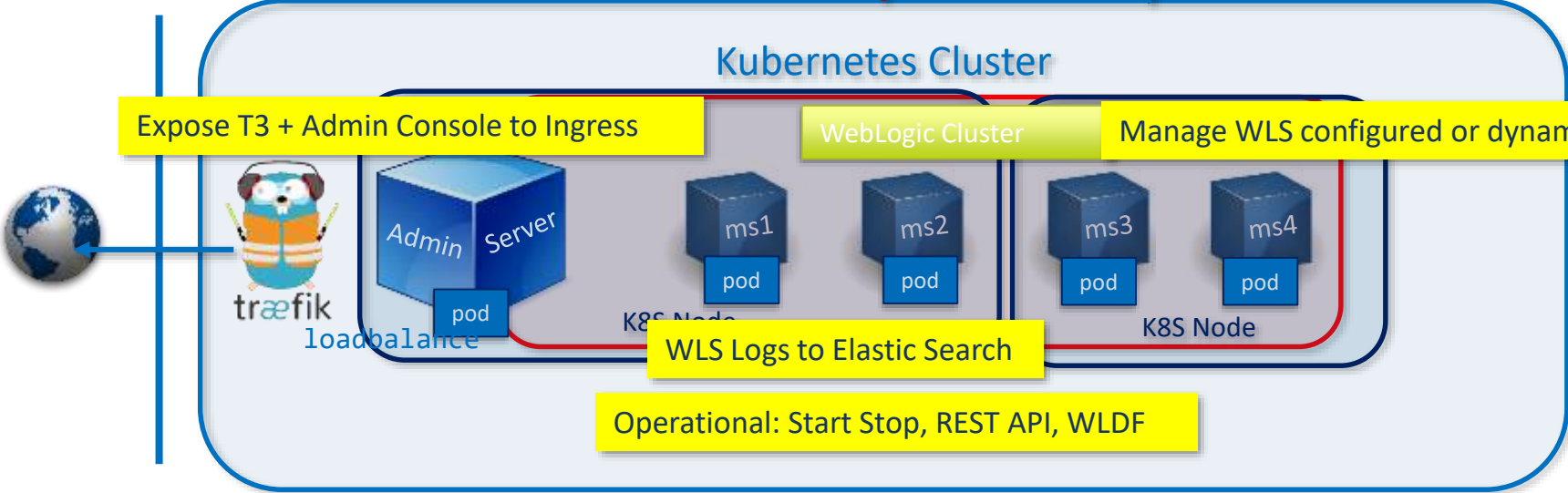
pod

K8S Node

K8S Node

WLS Logs to Elastic Search

Operational: Start Stop, REST API, WLDF



WebLogic Kubernetes Operator

- Simpler WebLogic management in Kubernetes
- Kubernetes resources are allocated for WebLogic domain(s)
- Manages overall WebLogic environment through Kubernetes APIs
 - Load Balancer, Network,
 - Ingress Controllers,
 - Security,
 - HA restart, upgrade, scaling
 - Persistent storage
- Ensures WebLogic best practices are followed
-

WebLogic Kubernetes Operator

- Helm charts for operator install
- Persistent volumes for logfiles



9



WebLogic Kubernetes
Prometheus & Grafana

Prometheus & Grafana

- OpenSource tools to monitor your stack
- Prometheus feeds Grafana
- Grafana visualizes
- With k8s yaml files and json configs

```
kubectl create -f prometheus/prometheus-kubernetes.yml  
kubectl create -f prometheus/grafana-kubernetes.yml
```


- Or helm install

```
helm install --name prometheus stable/prometheus
```

Do not forget to install a timer service account and role

WLS Prometheus & Grafana

- Specially use to monitor WLS in k8s
- Fedded with WebLogic metrics
- Use with the “WebLogic Monitor Exporter”
 - Tailored to export specific WebLogic JMX metrics
- Deploy it on your WebLogic domain

Name 	Test Point
<input type="checkbox"/> wls-exporter	
/wls-cat/index.jsp	ClassLoader Analysis Tool
default	http://10.244.0.64:8001/wls-exporter

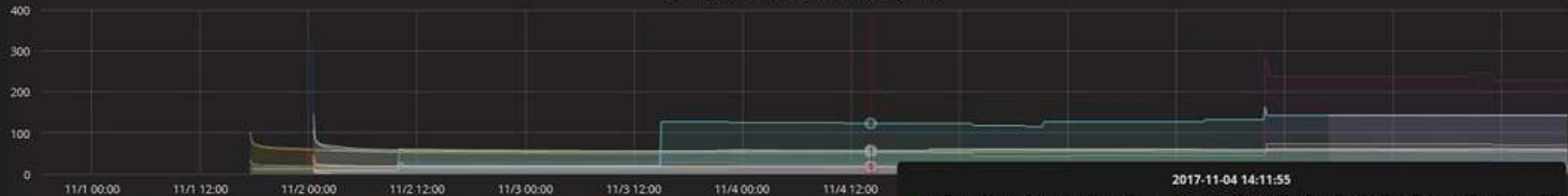
WLS Prometheus & Grafana

- Gather mbean metrics using a config file
 - WLDF query

```
query_sync:
  url: http://coordinator:8999/
  interval: 5
metricsNameSnakeCase: true
queries:
- applicationRuntimes:
  key: name
  keyName: app
  componentRuntimes:
    type: WebAppComponentRuntime
    prefix: webapp_config_
    key: name
    values: [deploymentState, contextRoot, sourceInfo, openSessionsHighCount]
  servlets:
    prefix: weblogic_servlet_
    key: servletName
    values: invocationTotalCount
```



WebLogic Servlet Execution Average Time



- ms-0 ;appName=bea_wls_cluster_internal ; servletName=MulticastSessionDataRecoveryServlet
- ms-0 ;appName=wls-management-services ; servletName=weblogic.management.rest.Application
- ms-0 ;appName=wls-exporter ; servletName=io.prometheus.wls.rest.MainServlet
- ms-1 ;appName=wls-management-services ; servletName=weblogic.management.rest.Application
- ms-1 ;appName=wls-exporter ; servletName=io.prometheus.wls.rest.MainServlet
- ms-2 ;appName=wls-management-services ; servletName=weblogic.management.rest.Application
- ms-3 ;appName=bea_wls_cluster_internal ; servletName=MulticastSessionDataRecoveryServlet
- ms-3 ;appName=wls-management-services ; servletName=weblogic.management.rest.Application
- ms-4 ;appName=bea_wls_cluster_internal ; servletName=MulticastSessionDataRecoveryServlet
- ms-4 ;appName=wls-management-services ; servletName=weblogic.management.rest.Application
- ms-4 ;appName=wls-exporter ; servletName=io.prometheus.wls.rest.MainServlet
- ms-0 ;appName=bea_wls_deployment_internal ; st
- ms-0 ;appName=wls-exporter ; servletName=io
- ms-1 ;appName=bea_wls_cluster_internal ; servletName=Multica
- ms-2 ;appName=wls-exporter ; servletName=io
- ms-3 ;appName=bea_wls_deployment_internal ; st
- ms-3 ;appName=wls-exporter ; servletName=io
- ms-4 ;appName=bea_wls_deployment_internal ; st
- ms-4 ;appName=wls-exporter ; servletName=io

+ ADD ROW

2017-11-04 14:11:55

- ms-0 ;appName=bea_wls_cluster_internal ; servletName=MulticastSessionDataRecoveryServlet: 51
- ms-0 ;appName=bea_wls_deployment_internal ; servletName=DeploymentService: 19
- ms-0 ;appName=wls-management-services ; servletName=weblogic.management.rest.Application: 19
- ms-0 ;appName=wls-exporter ; servletName=io.prometheus.wls.rest.ExporterServlet: 57
- ms-0 ;appName=wls-exporter ; servletName=io.prometheus.wls.rest.MainServlet: 25
- ms-1 ;appName=bea_wls_cluster_internal ; servletName=MulticastSessionDataRecoveryServlet: 17
- ms-1 ;appName=bea_wls_deployment_internal ; servletName=DeploymentService: 17
- ms-1 ;appName=wls-management-services ; servletName=weblogic.management.rest.Application: 17
- ms-1 ;appName=wls-exporter ; servletName=io.prometheus.wls.rest.ExporterServlet: 54
- ms-1 ;appName=wls-exporter ; servletName=io.prometheus.wls.rest.MainServlet: 12
- ms-2 ;appName=bea_wls_cluster_internal ; servletName=MulticastSessionDataRecoveryServlet: 13
- ms-2 ;appName=bea_wls_deployment_internal ; servletName=DeploymentService: 18
- ms-2 ;appName=wls-management-services ; servletName=weblogic.management.rest.Application: 18
- ms-2 ;appName=wls-exporter ; servletName=io.prometheus.wls.rest.ExporterServlet: 54
- ms-3 ;appName=bea_wls_cluster_internal ; servletName=MulticastSessionDataRecoveryServlet: 13
- ms-3 ;appName=bea_wls_deployment_internal ; servletName=DeploymentService: 18
- ms-3 ;appName=wls-management-services ; servletName=weblogic.management.rest.Application: 18
- ms-3 ;appName=wls-exporter ; servletName=io.prometheus.wls.rest.ExporterServlet: 56
- ms-4 ;appName=bea_wls_cluster_internal ; servletName=MulticastSessionDataRecoveryServlet: 123
- ms-4 ;appName=bea_wls_deployment_internal ; servletName=DeploymentService: 17
- ms-4 ;appName=wls-management-services ; servletName=weblogic.management.rest.Application: 17
- ms-4 ;appName=wls-exporter ; servletName=io.prometheus.wls.rest.ExporterServlet: 55
- ms-4 ;appName=wls-exporter ; servletName=io.prometheus.wls.rest.MainServlet: 55

```

@MetricsServlet: invocation total, count [app="bea_wls_cluster_internal", servlet="MulticastSessionDataRecoveryServlet"]
@MetricsServlet: invocation total, count [app="bea_wls_deployment_internal", servlet="DeploymentService"]
@MetricsServlet: invocation total, count [app="wls-management-services", servlet="weblogic.management.rest.Application"]
@MetricsServlet: invocation total, count [app="wls-exporter", servlet="io.prometheus.wls.rest.ExporterServlet"]
@MetricsServlet: invocation total, count [app="wls-exporter", servlet="io.prometheus.wls.rest.MainServlet"]
@MetricsServlet: invocation total, count [app="bea_wls_cluster_internal", servlet="MulticastSessionDataRecoveryServlet"]
@MetricsServlet: invocation total, count [app="bea_wls_deployment_internal", servlet="DeploymentService"]
@MetricsServlet: invocation total, count [app="wls-management-services", servlet="weblogic.management.rest.Application"]
@MetricsServlet: invocation total, count [app="wls-exporter", servlet="io.prometheus.wls.rest.ExporterServlet"]
@MetricsServlet: invocation total, count [app="wls-exporter", servlet="io.prometheus.wls.rest.MainServlet"]
@MetricsServlet: invocation total, count [app="bea_wls_cluster_internal", servlet="MulticastSessionDataRecoveryServlet"]
@MetricsServlet: invocation total, count [app="bea_wls_deployment_internal", servlet="DeploymentService"]
@MetricsServlet: invocation total, count [app="wls-management-services", servlet="weblogic.management.rest.Application"]
@MetricsServlet: invocation total, count [app="wls-exporter", servlet="io.prometheus.wls.rest.ExporterServlet"]
@MetricsServlet: invocation total, count [app="wls-exporter", servlet="io.prometheus.wls.rest.MainServlet"]

```



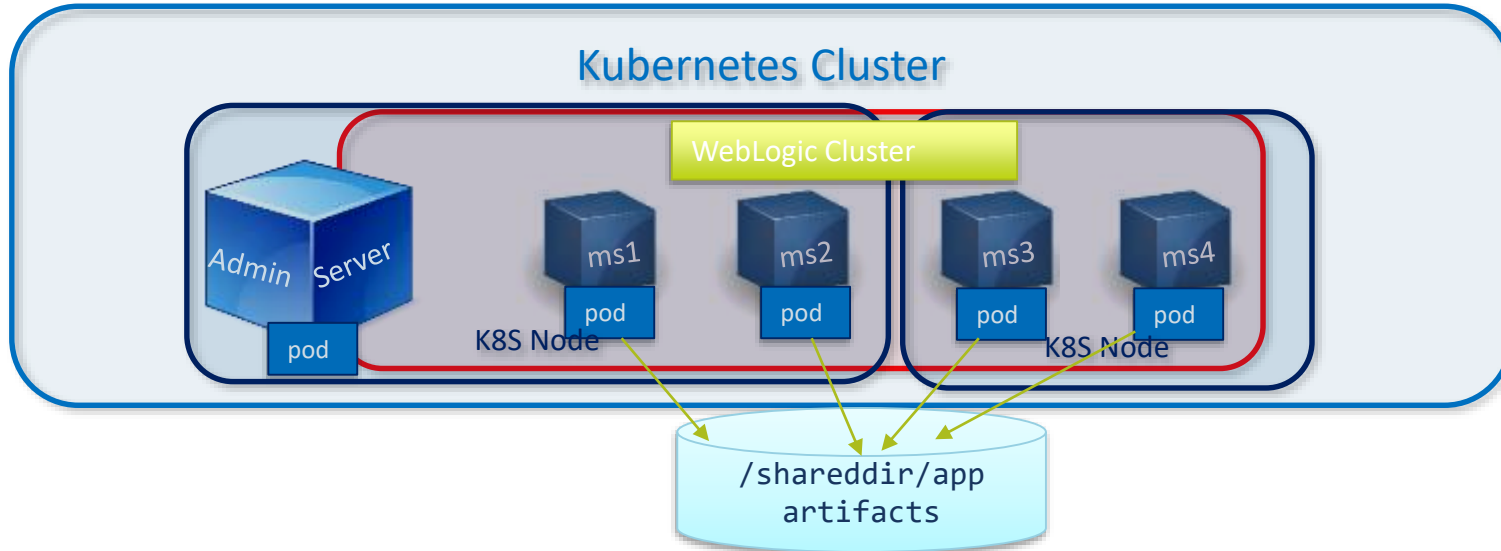
1.



Final.....

Some tips and considerations

- The more services, the more data exchange / network traffic
 - Scale enough resources
- Use HTTP(s)(HTTP/2) for external
- Use TCP/UDP for internal
- Serialization of applications
- Maybe coherence, JSON
- Map application files shareable on a host-volume



Some tips and considerations

- Better use small as possible images
- Use base image for as much as possible purposes
- Ensure that settings such as JVM heap has been synchronized to pods

500+ Technical Experts Helping Peers Globally

ORACLE®
ACE Program



3 Membership Tiers

- Oracle ACE Director
- Oracle ACE
- Oracle ACE Associate

bit.ly/OracleACEProgram

Connect:

✉ oracle-ace_ww@oracle.com

f Facebook.com/oracleaces

t @oracleace



Nominate yourself or someone you know: acenomination.oracle.com