Oracle on Kubernetes 101

Nelson Calero
Pythian Principal Consultant

hroug '22
Autumn | Jesen
11. - 14.10.2022., Rovinj

Nelson Calero

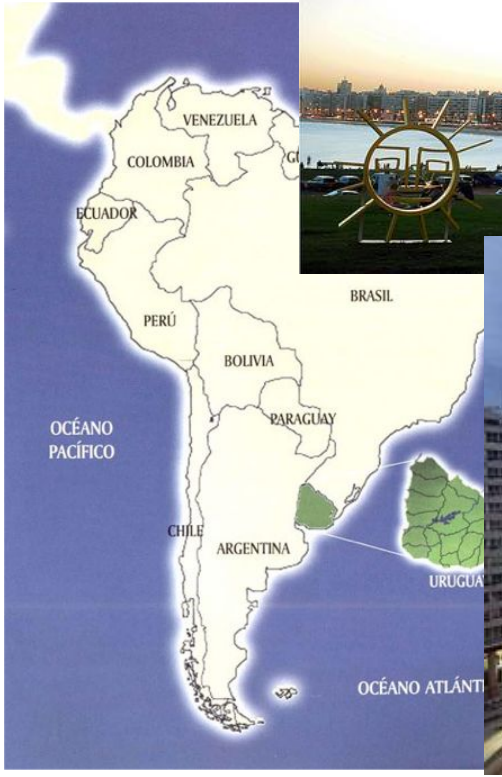
Pythian Principal Consultant

- 20+ years Oracle database experience
- Oracle ACE Director
- Montevideo, Uruguay

LinkedIn: http://www.linkedin.com/in/ncalero

Twitter: @ncalerouy

Uruguay


VENEZUELA
COLOMBIA
ECUADOR
PERÚ
BRASIL
BOLIVIA
PARAGUAY
OCÉANO
PACÍFICO
CHILE
ARGENTINA
URUGUA
OCÉANO ATLÁNT


Monte

Solutions designed to power the value of your data.

Solution that maximize the entire data state

Data & Analytics	CloudOps	Applications
Data & Analytics Consulting	Cloud Migrations	Oracle EBS
Database Services	Cloud Management – InfraOps / FinOps	SAP
Cloud Data Platform Services	Google Cloud	Collaboration & Workspace
Data Insights & Advanced Analytics	DevOps – CI/CD	Application Development, Migration & Refactoring

Wherever your data and applications reside

Public Cloud

Hybrid Cloud

On Premises

No matter what platform you choose

Google Cloud

SAP

amazon web services

ORACLE

snowflake

Microsoft Azure

PostgreSQL

25

Years in Business

420+

Experts across every Data Domain & Technology

400+

Global Customers

Oracle ACE Program

500+ technical experts helping peers globally

The **Oracle ACE Program** recognizes and rewards community members for their technical and community contributions to the Oracle community

3 membership tiers

Oracle ACE Director | Oracle ACE Pro | Oracle ACE Associate

For more details on Oracle ACE Program:
ace.oracle.com

Oracle ACE

Nominate

yourself or someone you know:

ace.oracle.com/nominate

Connect : **aceprogram_ww@oracle.com** Facebook.com/OracleACEs @oracleace

Agenda

- ➔ Concepts review
- ➔ OraOperator & ElCarro projects
- ➔ OCI Single instance Example

Oracle on Docker - old history

Lot of material from the community, until Oracle created their own:

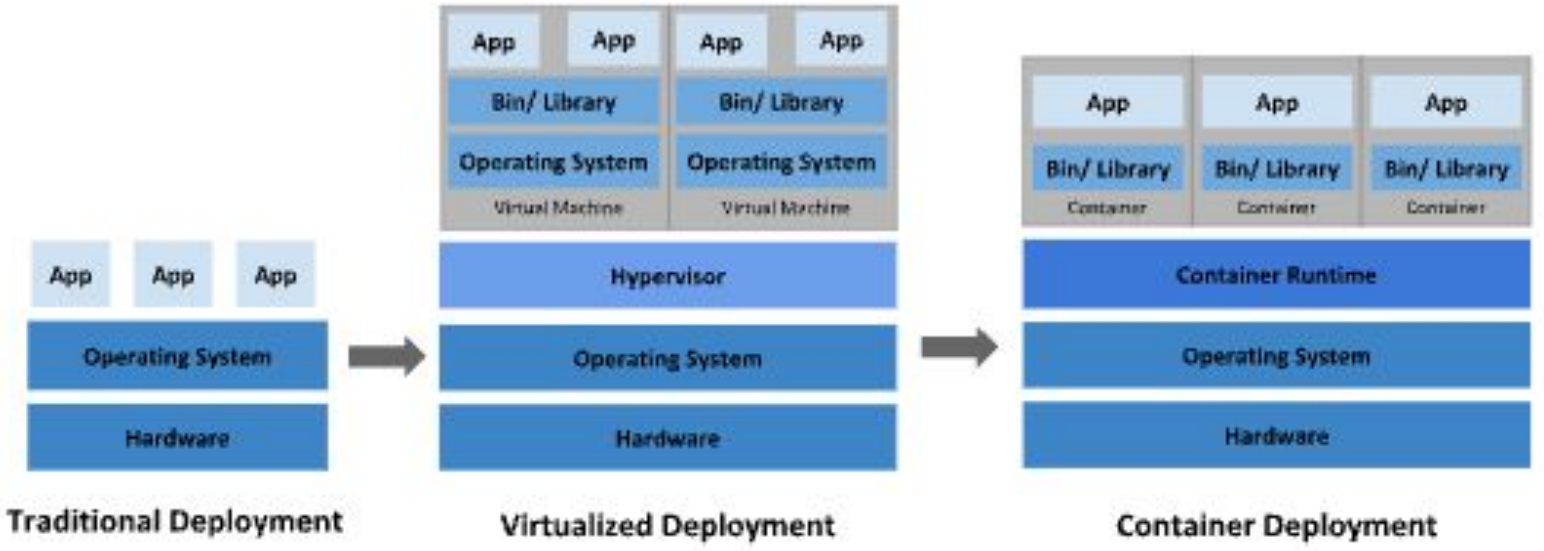
- docker image builds: SI/RAC/Sharding
  https://github.com/oracle/docker-images/tree/main/OracleDatabase

- prebuilt images in Oracle registry: SI/RAC/instantclient/+
  https://container-registry.oracle.com/

Docker is one of many container runtimes, but does not implement the Container Runtime Interface (CRI) used by Kubernetes

Kubernetes

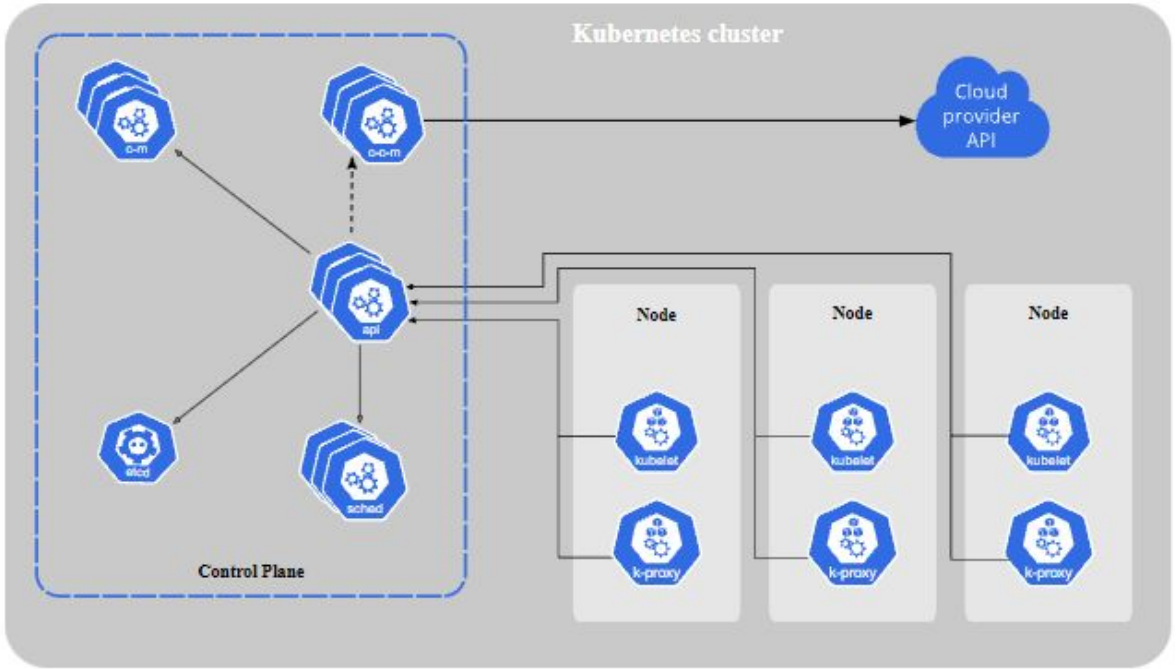
- Managed container cluster: HA, DR, scalability, orchestration
- Before it existed:



https://kubernetes.io/docs/concepts/overview/components/

Kubernetes

- Managed container cluster: HA, DR, scalability, orchestration
- Kubernetes components:



https://kubernetes.io/docs/concepts/overview/components/

Kubernetes operators

- Software extension providing lifecycle management of custom resources https://kubernetes.io/docs/concepts/extend-kubernetes/operator/
- Operators framework: https://operatorframework.io/
- What operators provide:
  - Custom resources (CRDs)
  - Lifecycle management in controllers: start/stop/terminate/others
  - monitoring and logging
- Key management features:
  - pull image from public container registry (GCP, Oracle, etc.)
  - using k8s persistent storage - data (volume) stays after cluster is destroyed
- Extensive adoption: servers, database engines, applications, cloud resources

Why databases on Kubernetes?

https://cloud.google.com/blog/products/databases/to-run-or-not-to-run-a-database-on-kubernetes-what-to-consider

Oracle database Kubernetes operators

- Oracle database operators:
  - May 2021: https://github.com/GoogleCloudPlatform/elcarro-oracle-operator
  - Sep 2021: v0.1 https://github.com/oracle/oracle-database-operator
  - Jun 2022: v0.2 OraOperator
- Persistent storage
  - Volume attached to container, not deleted with cluster
  - Using Container Storage Interfase (CSI) plugins:
    https://kubernetes-csi.github.io/docs/drivers.html
  - Makes operators portable between clouds, as each cloud has its own provisioner:
    - OCI: blockvolume.csi.oraclecloud.com
    - AWS: ebs.csi.aws.com
    - GCP: pd.csi.storage.gke.io

Where to use Oracle on kubernetes

- Google Kubernetes Engine (GKE) https://cloud.google.com/kubernetes-engine
- Amazon Elastic Kubernetes Engine (EKS) https://aws.amazon.com/eks/
- Oracle Container Engine for Kubernetes (OKE)
  https://www.oracle.com/cloud-native/container-engine-kubernetes/
- Local
  - Kind https://github.com/kubernetes-sigs/kind
  - Minikube https://minikube.sigs.k8s.io/docs/start/
  - Oracle Cloud Native Environment (OLCNE)
    https://docs.oracle.com/en/operating-systems/olcne/concepts/cncf.html

Oracle® Cloud Native Environment Concepts for Release 1.4

- 2 Components
  - 2.1 Container Runtimes
  - 2.2 Container Orchestration
  - 2.3 Cloud Native Networking
  - 2.4 Cloud Native Storage
- 3 Architecture
  - 3.1 Platform API Server
  - 3.2 Platform Agent
  - 3.3 Platform CLI
  - 3.4 Authentication
- 4 Environments and Modules
  - 4.1 Environments
  - 4.2 Modules
    - 4.2.1 Kubernetes Module
    - 4.2.2 Istio Module
    - 4.2.3 Helm Module
    - 4.2.4 Prometheus Module
    - 4.2.5 Grafana Module
    - 4.2.6 Operator Lifecycle Manager Module
    - 4.2.7 Oracle Cloud Infrastructure Container Storage Interface Module
    - 4.2.8 Gluster Container Storage Interface Module

Chapter 4 Environments and Modules

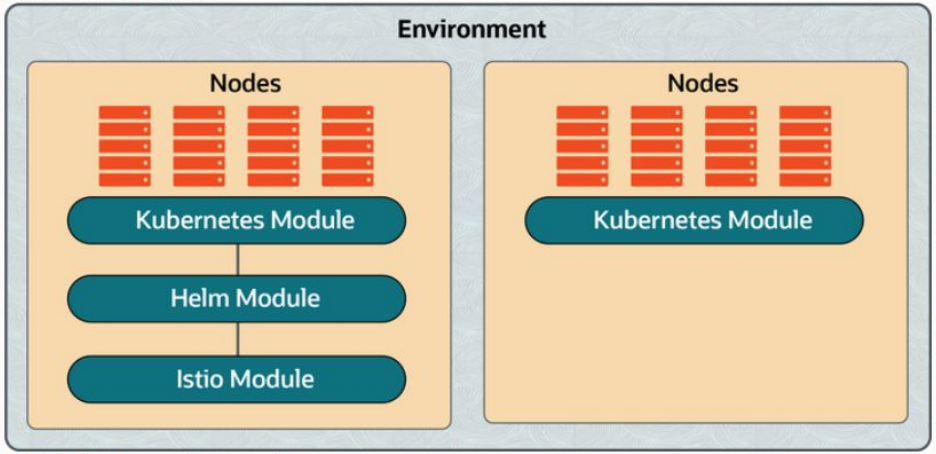
This chapter introduces the concepts of environments and modules in Oracle Cloud Native Environment.

4.1 Environments

An *environment* is a namespace that encapsulates the software installed and managed by Oracle Cloud Native Environment. Each environment contains at least the Kubernetes module.

The Platform CLI allows you to create and manage multiple deployments. Each deployment contains an environment, and each environment may potentially contain multiple modules. This allows you to create multiple Kubernetes clusters using the same Oracle Cloud Native Environment installation. Each Kubernetes cluster must have dedicated nodes, that is, a server cannot be used in two clusters, or environments.

Figure 4.1 Environments



Click to view larger image environment.png

Operator portability

Minor changes required to run on different clouds:

- Grant k8s cluster privileges to access cloud native services (k8s engine, volumes)
- Store credentials using local services (vault) syntax
- Use k8s storage class from the cloud provider
    - Example: OCI already includes "oci-bv" CSI volume plugin into OKE clusters, but AWS requires manual installation of their EBS-CSI driver into EKS
- Use of native build service if creating your own container image (instead of pulling one from public repositories):
    - Example: AWS CodeBuild instead of GCP Cloud Build

El Carro operator (GCP)

All about it:

- https://opensource.googleblog.com/2021/05/modernizing-oracle-operations-with-kubernetes-el-carro.html
- https://opensource.googleblog.com/2021/08/el-carro-extends-flexibility-and-choices-for-Oracle-databases-on-Kubernetes.html

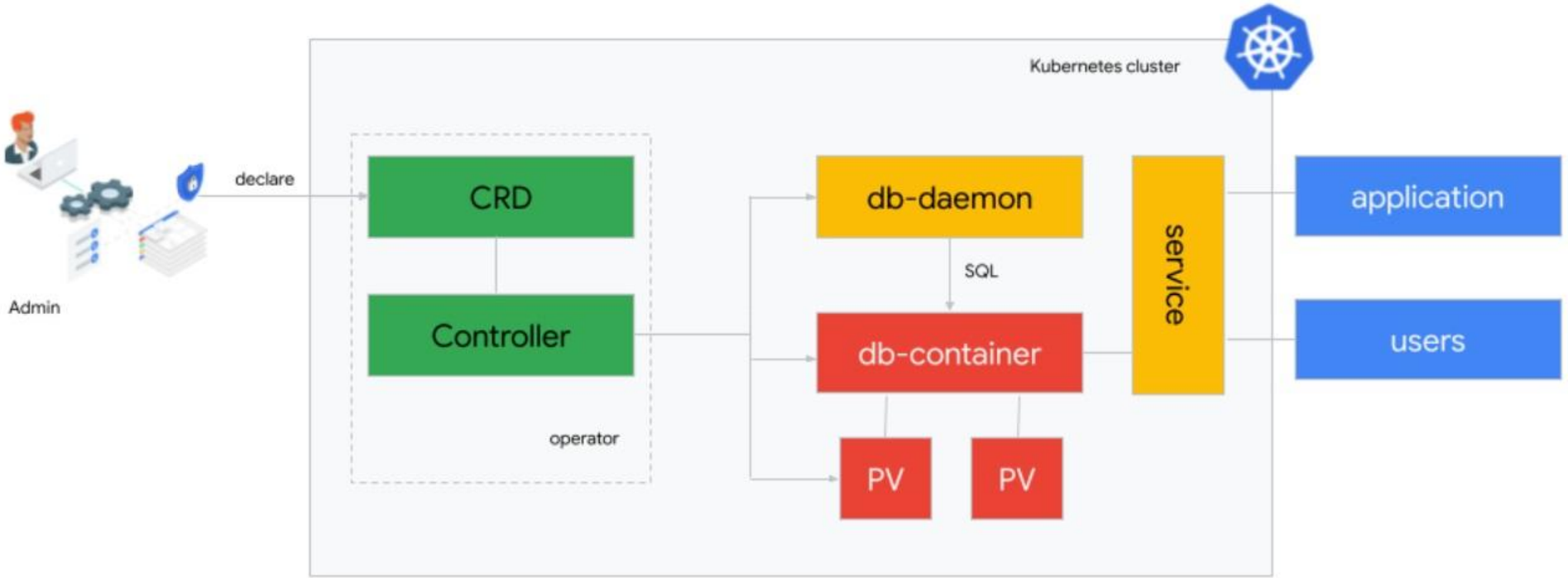
Key concepts:

- SI XE/12c/18c/19c
- Backup/restore operations
- Well documented - 4 ways to deploy the container image

Contributed to testing ElCarro in AWS EKS and published the guide:

- https://github.com/GoogleCloudPlatform/elcarro-oracle-operator/blob/main/docs/content/aws.md
- https://blog.pythian.com/using-el-carro-operator-on-aws/

ElCarro components



OraOperator: Oracle Database Operator for Kubernetes

Where to start

- https://medium.com/oracledevs/making-oracle-database-kubernetes-native-401233c02780
- https://medium.com/oracledevs/oracle-database-operator-for-kubernetes-take-2-e0227ca7b7c3
- https://github.com/oracle/oracle-database-operator

First version (v0.1.0) supports:

- Autonomous Databases (all flavors:ATP, ADW, AJD, Apex) on Shared Infra (ADB-S)
- Single Instance DB (SE, EE) in OKE (OCI) and OLCNE (on-prem)
- Sharding DB in OKE and OLCNE

v0.2.0 adds:

- ADB-D, Autonomous Container Database (ACD), Database Cloud Service (DBCS)
- Single Instance DB of pre-built XE provisioning

OraOperator v0.1 operations (as of Nov/2021)

Autonomous DB (ADB, off k8s):

- provision, bind, start, stop, terminate, scale (up/down)

Single instance

- provision
- Update init params (ex: SGA size)
- Update config (ex: flashback, archiving)
- Clone existing DB
- Patch existing Database (in place/out of place)

Sharded DB:

- provision
- add/remove shards

OraOperator - prerequisites

1) A kubernetes cluster running. Using OCI in examples below

If using OCI OKE, the Cloud Shell can be used (as it comes with kubectl and helm preinstalled) - no need for a compute instance

OraOperator - prerequisites

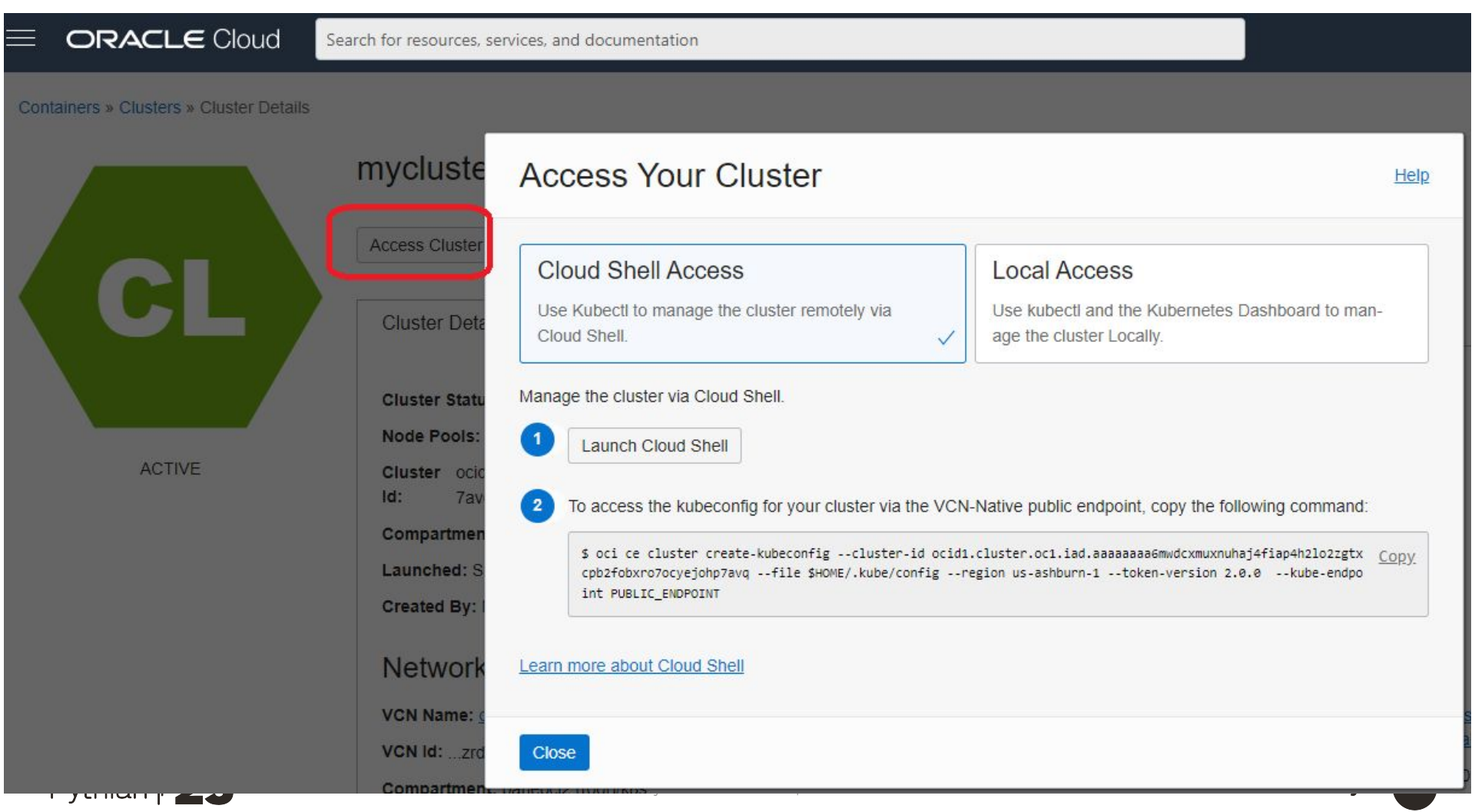
1) A kubernetes cluster running. Using OCI in examples below

If using OCI OKE, the Cloud Shell can be used (as it comes with kubectl and helm preinstalled) - no need for a compute instance

- Needs a one-time config like this, using your k8s OCID:

```
$ oci ce cluster create-kubeconfig --cluster-id ocid1.cluster.oc1.phx.aaaaaaaaae... --file
$HOME/.kube/config --region us-phoenix-1 --token-version 2.0.0
```

- “access your cluster“ button in web console


ORACLE Cloud
Search for resources, services, and documentation
Containers » Clusters » Cluster Details
CL
ACTIVE
mycluste
Access Cluster
Cluster Deta
Cluster Statu
Node Pools:
Cluster Id: ocid 7av
Compartmen
Launched: S
Created By:
Network
VCN Name:
VCN Id: ...zrd
Compartmen
Access Your Cluster
Help
Cloud Shell Access
Use Kubectl to manage the cluster remotely via Cloud Shell.
Local Access
Use kubectl and the Kubernetes Dashboard to manage the cluster Locally.
Manage the cluster via Cloud Shell.
1
Launch Cloud Shell
2
To access the kubeconfig for your cluster via the VCN-Native public endpoint, copy the following command:
$ oci ce cluster create-kubeconfig --cluster-id ocid1.cluster.oc1.iad.aaaaaaaa6mwdcxmuxnuhaj4fiap4h2lo2zgtx cpb2fobxro7ocyejohp7avq --file $HOME/.kube/config --region us-ashburn-1 --token-version 2.0.0 --kube-endpoint PUBLIC_ENDPOINT
Copy
Learn more about Cloud Shell
Close

OraOperator - prerequisites

1) A kubernetes cluster running. Using OCI in examples below

If using OCI OKE, the Cloud Shell can be used (as it comes with kubectl and helm preinstalled) - no need for a compute instance

- Needs a one-time config like this, using your k8s OCID:

```
$ oci ce cluster create-kubeconfig --cluster-id ocid1.cluster.oc1.phx.aaaaaaaaae... --file
$HOME/.kube/config --region us-phoenix-1 --token-version 2.0.0
```

- “access your cluster“ button in web console

Step-by-step instructions (useful outside of cloud shell):
https://www.oracle.com/webfolder/technetwork/tutorials/obe/oci/oke-cloudshell/index.html

OraOperator - prerequisites

2) Install cert-manager into your k8s cluster:

```
$ kubectl apply -f https://github.com/jetstack/cert-manager/releases/latest/download/cert-manager.yaml
```

```
Nelson_Cal@cloudshell:~ (us-ashburn-1)$kubectl apply -f
https://github.com/jetstack/cert-manager/releases/latest/download/cert-manager.yaml

customresourcedefinition.apiextensions.k8s.io/certificaterequests.cert-manager.io created
customresourcedefinition.apiextensions.k8s.io/certificates.cert-manager.io created
customresourcedefinition.apiextensions.k8s.io/challenges.acme.cert-manager.io created
customresourcedefinition.apiextensions.k8s.io/clusterissuers.cert-manager.io created
customresourcedefinition.apiextensions.k8s.io/issuers.cert-manager.io created
customresourcedefinition.apiextensions.k8s.io/orders.acme.cert-manager.io created
namespace/cert-manager created
serviceaccount/cert-manager-cainjector created
serviceaccount/cert-manager created
serviceaccount/cert-manager-webhook created
clusterrole.rbac.authorization.k8s.io/cert-manager-cainjector created
clusterrole.rbac.authorization.k8s.io/cert-manager-controller-issuers created
clusterrole.rbac.authorization.k8s.io/cert-manager-controller-clusterissuers created
clusterrole.rbac.authorization.k8s.io/cert-manager-controller-certificates created
clusterrole.rbac.authorization.k8s.io/cert-manager-controller-orders created
clusterrole.rbac.authorization.k8s.io/cert-manager-controller-challenges created
clusterrole.rbac.authorization.k8s.io/cert-manager-controller-ingress-shim created
clusterrole.rbac.authorization.k8s.io/cert-manager-view created
clusterrole.rbac.authorization.k8s.io/cert-manager-edit created
clusterrole.rbac.authorization.k8s.io/cert-manager-controller-approve:cert-manager-io created
...
service/cert-manager created
service/cert-manager-webhook created
deployment.apps/cert-manager-cainjector created
deployment.apps/cert-manager created
deployment.apps/cert-manager-webhook created
mutatingwebhookconfiguration.admissionregistration.k8s.io/cert-manager-webhook created
validatingwebhookconfiguration.admissionregistration.k8s.io/cert-manager-webhook created
Nelson_Cal@cloudshell:~ (us-ashburn-1)$
```

OraOperator - prerequisites

2) Install cert-manager into your k8s cluster:

```
$ kubectl apply -f https://github.com/jetstack/cert-manager/releases/latest/download/cert-manager.yaml
```

3) Deploy the operator:

```
$ kubectl apply -f
https://raw.githubusercontent.com/oracle/oracle-database-operator/main/oracle-database-operator.yaml
```

OraOperator - prerequisites

```
Nelson_Cal@cloudshell:~ (us-ashburn-1)$kubectl apply -f
https://raw.githubusercontent.com/oracle/oracle-database-operator/main/oracle-database-operator.yaml
namespace/oracle-database-operator-system created
customresourcedefinition.apiextensions.k8s.io/autonomousdatabases.database.oracle.com created
customresourcedefinition.apiextensions.k8s.io/shardingdatabases.database.oracle.com created
customresourcedefinition.apiextensions.k8s.io/singleinstancedatabases.database.oracle.com created
role.rbac.authorization.k8s.io/oracle-database-operator-leader-election-role created
clusterrole.rbac.authorization.k8s.io/oracle-database-operator-manager-role created
clusterrole.rbac.authorization.k8s.io/oracle-database-operator-metrics-reader created
clusterrole.rbac.authorization.k8s.io/oracle-database-operator-oracle-database-operator-proxy-role created
rolebinding.rbac.authorization.k8s.io/oracle-database-operator-oracle-database-operator-leader-election-rolebinding created
clusterrolebinding.rbac.authorization.k8s.io/oracle-database-operator-oracle-database-operator-manager-rolebinding created
clusterrolebinding.rbac.authorization.k8s.io/oracle-database-operator-oracle-database-operator-proxy-rolebinding created
service/oracle-database-operator-controller-manager-metrics-service created
service/oracle-database-operator-webhook-service created
deployment.apps/oracle-database-operator-controller-manager created
certificate.cert-manager.io/oracle-database-operator-serving-cert created
issuer.cert-manager.io/oracle-database-operator-selfsigned-issuer created
mutatingwebhookconfiguration.admissionregistration.k8s.io/oracle-database-operator-mutating-webhook-configuration created
validatingwebhookconfiguration.admissionregistration.k8s.io/oracle-database-operator-validating-webhook-configuration created
Nelson_Cal@cloudshell:~ (us-ashburn-1)$
```

```
Nelson_Cal@cloudshell:~ (us-ashburn-1)$kubectl get pods -n oracle-database-operator-system
NAME                                                        READY   STATUS    RESTARTS   AGE
oracle-database-operator-controller-manager-58447bcbf8-8n7x6   1/1     Running   0          85s
oracle-database-operator-controller-manager-58447bcbf8-h75nj   1/1     Running   0          86s
oracle-database-operator-controller-manager-58447bcbf8-q6vkb   1/1     Running   0          86s

Nelson_Cal@cloudshell:~ (us-ashburn-1)$kubectl get nodes -o wide
NAME          STATUS   ROLES   AGE   VERSION    INTERNAL-IP   EXTERNAL-IP   OS-IMAGE                   KERNEL-VERSION
CONTAINER-RUNTIME
10.0.10.209   Ready    node    80m   v1.20.11   10.0.10.209   <none>        Oracle Linux Server 7.9
5.4.17-2102.206.1.el7uek.x86_64   cri-o://1.20.2
10.0.10.216   Ready    node    80m   v1.20.11   10.0.10.216   <none>        Oracle Linux Server 7.9
5.4.17-2102.206.1.el7uek.x86_64   cri-o://1.20.2
10.0.10.87    Ready    node    80m   v1.20.11   10.0.10.87    <none>        Oracle Linux Server 7.9
5.4.17-2102.206.1.el7uek.x86_64   cri-o://1.20.2

Nelson_Cal@cloudshell:~ (us-ashburn-1)$kubectl describe deployments/oracle-database-operator-controller-manager -n
oracle-database-operator-system
Name:                   oracle-database-operator-controller-manager
Namespace:              oracle-database-operator-system
CreationTimestamp:      Sun, 05 Dec 2021 19:55:49 +0000
Labels:                 control-plane=controller-manager
Annotations:            deployment.kubernetes.io/revision: 1
Selector:               control-plane=controller-manager
Replicas:               3 desired | 3 updated | 3 total | 3 available | 0 unavailable
StrategyType:           RollingUpdate
...
```

OraOperator - prerequisites

2) Install cert-manager into your k8s cluster:

```
$ kubectl apply -f https://github.com/jetstack/cert-manager/releases/latest/download/cert-manager.yaml
```

3) Deploy the operator:

```
$ kubectl apply -f
https://raw.githubusercontent.com/oracle/oracle-database-operator/main/oracle-database-operator.yaml
```

4) If using Oracle Cluster Registry, need credentials available to k8s.
One way to do this:

```
$ docker login container-registry.oracle.com

$ kubectl create secret generic oracrs --from-file=.dockerconfigjson=$HOME/.docker/config.json
--type=kubernetes.io/dockerconfigjson
```

https://kubernetes.io/docs/tasks/configure-pod-container/pull-image-private-registry/

OraOperator - prerequisites

2) Install cert-manager into your k8s cluster

```
$ kubec
```

3) Deploy th

```
$ kubec
https:/
```

4) If using O
One way t

```
$ docke

$ kubec
--type=
```

http

```
Nelson_Cal@cloudshell:~ (us-ashburn-1)$docker login container-registry.oracle.com
Username: nelson.calero@gmail.com
Password: ****
WARNING! Your password will be stored unencrypted in /home/Nelson_Cal/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
Nelson_Cal@cloudshell:~ (us-ashburn-1)$

Nelson_Cal@cloudshell:~ (us-ashburn-1)$kubectl create secret generic oracrs
--from-file=.dockerconfigjson=$HOME/.docker/config.json  --type=kubernetes.io/dockerconfigjson

secret/oracrs created
Nelson_Cal@cloudshell:~ (us-ashburn-1)$
Nelson_Cal@cloudshell:~ (us-ashburn-1)$kubectl get secret
NAME                  TYPE                                  DATA   AGE
default-token-xpqrq   kubernetes.io/service-account-token   3      118m
oracrs                kubernetes.io/dockerconfigjson        1      57s
Nelson_Cal@cloudshell:~ (us-ashburn-1)$
```

OraOperator Example - Single Instance

Provisioning

- Customize sample .yaml

  https://github.com/oracle/oracle-database-operator/blob/main/config/samples/sidb/singleinstancedatabase.yaml

```
$ vi singleinstancedatabase.yaml
```

- Parameters requiring additional work:
  - Image to use: custom or prebuilt from a registry?
  - adminPassword: secret should be created
  - Storage class: pre-created or new?

OraOperator Example - Single Instance

Provisioning - customization

- image: from Oracle container registry, using secret *oracrs* created before

```
  image:
    pullFrom: container-registry.oracle.com/database/enterprise:19.3.0.0
    pullSecrets: oracrs
```

- adminPassword: credential accessible from k8s to use for the DB creation.
  One way to do it (just for testing, use vault for serious deployments):

```
Nelson_Cal@cloudshell:~ (us-ashburn-1)$kubectl create secret generic admin-pwd
--from-literal=mydb-admin-pwd='my0Racl3'
secret/admin-pwd created
Nelson_Cal@cloudshell:~ (us-ashburn-1)$kubectl get secrets
NAME                  TYPE                                  DATA   AGE
admin-pwd             Opaque                                1      5m40s
default-token-xpqrq   kubernetes.io/service-account-token   3      100m
Nelson_Cal@cloudshell:~ (us-ashburn-1)$
```

OraOperator Example - Single Instance

Provisioning - customization

- persistence: storage to use

```
  persistence:
    size: 20Gi
    storageClass: "oci"
    accessMode: "ReadWriteOnce"
```

- Above creates a new OCI block volume.
- To reuse an existing OCI volume, need extra setup, as described here; https://rohitchaware.medium.com/static-volume-provisioning-in-oracle-container-engine-for-kubernetes-oke-with-container-storage-a63846a18aeb

```
Nelson_Cal@cloudshell:~ (us-ashburn-1)$kubectl apply -f oci_static_pv.yaml
persistentvolume/static-pv created
Nelson_Cal@cloudshell:~ (us-ashburn-1)$kubectl apply -f oci_static_pvc.yaml
persistentvolumeclaim/static-pvc created
Nelson_Cal@cloudshell:~ (us-ashburn-1)$kubectl get pvc
NAME         STATUS   VOLUME      CAPACITY   ACCESS MODES   STORAGECLASS   AGE
static-pvc   Bound    static-pv   50Gi       RWX            oci-bv         8s
```

OraOperator Example - Single Instance

Provisioning

- Create the database:

```
Nelson_Cal@cloudshell:~ (us-ashburn-1)$kubectl apply -f singleinstancedatabase.yaml
singleinstancedatabase.database.oracle.com/mydb created
Nelson_Cal@cloudshell:~ (us-ashburn-1)$kubectl get singleinstancedatabase
NAME    EDITION      STATUS    VERSION    CONNECT STR              OEM EXPRESS URL
mydb    Enterprise   Pending   Unknown    10.0.10.87:31171/ORCL1   https://10.0.10.87:31452/em
```

- Creation status: Pending → Unhealthy → Patching → Healthy

- Once ready:

```
Nelson_Cal@cloudshell:~ (us-ashburn-1)$kubectl get singleinstancedatabase
NAME    EDITION      STATUS    VERSION                  CONNECT STR              OEM EXPRESS URL
mydb    Enterprise   Healthy   19.3.0.0.0 (29517242)    10.0.10.87:31171/ORCL1   https://10.0.10.87:31452/em
```

```
Nelson_Cal@cloudshell:~ (us-ashburn-1)$  cat singleinstancedatabase.yaml
#
# Copyright (c) 2021, Oracle and/or its affiliates.
# Licensed under the Universal Permissive License v 1.0 as shown at http://oss.oracle.com/licenses/upl.
#
apiVersion: database.oracle.com/v1alpha1
kind: SingleInstanceDatabase
metadata:
  name: mydb
  namespace: default
spec:

  ## Use only alphanumeric characters for sid
  sid: ORCL1

  ## A source database ref to clone from, leave empty to create a fresh database
  cloneFrom: ""

  ## NA if cloning from a SourceDB (cloneFrom is set)
  edition: enterprise

  ## Should refer to SourceDB secret if cloning from a SourceDB (cloneFrom is set)
  ## Secret containing SIDB password mapped to secretKey
  ## This secret will be deleted after creation of the database unless keepSecret is set to true
  adminPassword:
    secretName: admin-pwd
    secretKey: mydb-admin-pwd
    keepSecret: false
 ...
```

```
Nelson_Cal@cloudshell:~ (us-ashburn-1)$  cat singleinstancedatabase.yaml
#
# Copyright (c) 2021, Oracle and/or its a
# Licensed under the Universal Permissive
#
apiVersion: database.oracle.com/v1alpha1
kind: SingleInstanceDatabase
metadata:
  name: mydb
  namespace: default
spec:

  ## Use only alphanumeric characters fo
  sid: ORCL1

  ## A source database ref to clone from
  cloneFrom: ""

  ## NA if cloning from a SourceDB (clone
  edition: enterprise

  ## Should refer to SourceDB secret if
  ## Secret containing SIDB password mapp
  ## This secret will be deleted after c
  adminPassword:
    secretName: admin-pwd
    secretKey: mydb-admin-pwd
    keepSecret: false
 ...
```

```
  ## NA if cloning from a SourceDB (cloneFrom is set)
  charset: AL32UTF8

  ## NA if cloning from a SourceDB (cloneFrom is set)
  pdbName: orclpdb1

  ## Enable/Disable Flashback
  flashBack: false

  ## Enable/Disable ArchiveLog
  archiveLog: false

  ## Enable/Disable ForceLogging
  forceLog: false

  ## NA if cloning from a SourceDB (cloneFrom is set)
  ## Specify both sgaSize and pgaSize (in MB) or dont specify both
  ## Specify Non-Zero value to use
  initParams:
    cpuCount: 1
    processes: 100
    sgaTarget: 0
    pgaAggregateTarget: 0

  ## Database image details
  ## Database can be patched by updating the RU version/image
  ## Major version changes are not supported
  image:
```

```
Nelson_Cal@cloudshell:~ (us-ashburn-1)$  cat singleinstancedatabase.yaml
#
# Copyright (c) 2021, Oracle and/or its
# Licensed under the Universal Permissive
#
apiVersion: database.oracle.com/v1alpha1
kind: SingleInstanceDatabase
metadata:
  name: mydb
  namespace: default
spec:

  ## Use only alphanumeric characters fo
  sid: ORCL1

  ## A source database ref to clone from
  cloneFrom: ""

  ## NA if cloning from a SourceDB (clone
  edition: enterprise

  ## Should refer to SourceDB secret if
  ## Secret containing SIDB password mapp
  ## This secret will be deleted after c
  adminPassword:
    secretName: admin-pwd
    secretKey: mydb-admin-pwd
    keepSecret: false
 ...
```

```
  ## NA if cloning from a SourceDB (cloneFrom is set)
  charset: AL32UTF8

  ## NA if cloning from a S
  pdbName: orclpdb1

  ## Enable/Disable Flashba
  flashBack: false

  ## Enable/Disable Archive
  archiveLog: false

  ## Enable/Disable ForceLo
  forceLog: false

  ## NA if cloning from a S
  ## Specify both sgaSize a
  ## Specify Non-Zero value
  initParams:
    cpuCount: 1
    processes: 100
    sgaTarget: 0
    pgaAggregateTarget: 0

  ## Database image details
  ## Database can be patche
  ## Major version changes
  image:
```

```
    image:
      pullFrom:
container-registry.oracle.com/database/enterprise:19.3.0.0
      pullSecrets: oracrs

    ## size : Minimum size of pvc | class : PVC storage Class
    ## AccessMode can only accept one of ReadWriteOnce, ReadWriteMany
    persistence:
      size: 20Gi
      storageClass: "oci"
      accessMode: "ReadWriteOnce"

    ## Type of service . Applicable on cloud enviroments only
    ## if loadBalService : false, service type = "NodePort". else
"LoadBalancer"
    loadBalancer: false

    ## Deploy only on nodes having required labels. Format label_name :
label_value
    ## Leave empty if there is no such requirement.
    ## Uncomment to use
    # nodeSelector:
    #   failure-domain.beta.kubernetes.io/zone: bVCG:PHX-AD-1
    #   pool: sidb

    ## Count of Database Pods. Applicable only for "ReadWriteMany"
AccessMode
    replicas: 1
```

```
Nelson_Cal@cloudshell:~ (us-ashburn-1)$  kubectl get events --sort-by=.metadata.creationTimestamp
LAST SEEN   TYPE      REASON                   OBJECT                          MESSAGE
6m20s       Normal    ExternalProvisioning     persistentvolumeclaim/mydb      waiting for a volume to be created, either by external
provisioner "oracle.com/oci" or manually created by system administrator
6m16s       Warning   FailedScheduling         pod/mydb-w970s                  0/3 nodes are available: 3 pod has unbound immediate
PersistentVolumeClaims.
6m28s       Warning   FailedScheduling         pod/mydb-w970s                  0/3 nodes are available: 3 pod has unbound immediate
PersistentVolumeClaims.
6m17s       Normal    Provisioning             persistentvolumeclaim/mydb      External provisioner is provisioning volume for claim
"default/mydb"
6m16s       Normal    ProvisioningSucceeded    persistentvolumeclaim/mydb      Successfully provisioned volume
ocid1.volume.oc1.iad.abuwcljtqeis5roxlmgi44kvoawxo4dpxomhku37qsekfav745kg7jrtva7a
6m12s       Normal    Scheduled                pod/mydb-w970s                  Successfully assigned default/mydb-w970s to 10.0.10.87
5m51s       Normal    SuccessfulAttachVolume   pod/mydb-w970s                  AttachVolume.Attach succeeded for volume
"ocid1.volume.oc1.iad.abuwcljtqeis5roxlmgi44kvoawxo4dpxomhku37qsekfav745kg7jrtva7a"
5m30s       Normal    Pulling                  pod/mydb-w970s                  Pulling image
"container-registry.oracle.com/database/enterprise:19.3.0.0"
50s         Normal    Started                  pod/mydb-w970s                  Started container init-permissions
50s         Normal    Pulled                   pod/mydb-w970s                  Successfully pulled image
"container-registry.oracle.com/database/enterprise:19.3.0.0" in 4m40.330492154s
50s         Normal    Created                  pod/mydb-w970s                  Created container init-permissions
49s         Normal    Pulled                   pod/mydb-w970s                  Container image
"container-registry.oracle.com/database/enterprise:19.3.0.0" already present on machine
49s         Normal    Created                  pod/mydb-w970s                  Created container init-wallet
49s         Normal    Started                  pod/mydb-w970s                  Started container init-wallet
39s         Normal    Pulling                  pod/mydb-w970s                  Pulling image
"container-registry.oracle.com/database/enterprise:19.3.0.0"
39s         Normal    Database Pending         singleinstancedatabase/mydb     waiting for database pod to be ready
38s         Normal    Started                  pod/mydb-w970s                  Started container mydb
38s         Normal    Created                  pod/mydb-w970s                  Created container mydb
38s         Normal    Pulled                   pod/mydb-w970s                  Successfully pulled image
"container-registry.oracle.com/database/enterprise:19.3.0.0" in 607.044715ms
14s         Warning   Unhealthy                pod/mydb-w970s                  Readiness probe failed: [2021:12:05 21:00:07]: Connecting to
the lock process /tmp/.ORCL1.create_lck
```

```
Nelson_Cal@cloudshell:~ (us-ashburn-1)$  kubectl logs pod/mydb-w970s
[2021:12:05 20:59:43]: Acquiring lock .ORCL1.create_lck with heartbeat 30 secs
[2021:12:05 20:59:43]: Lock acquired
[2021:12:05 20:59:43]: Starting heartbeat
[2021:12:05 20:59:43]: Lock held .ORCL1.create_lck
ORACLE EDITION: ENTERPRISE

LSNRCTL for Linux: Version 19.0.0.0.0 - Production on 05-DEC-2021 20:59:43

Copyright (c) 1991, 2019, Oracle.  All rights reserved.

Starting /opt/oracle/product/19c/dbhome_1/bin/tnslsnr: please wait...

TNSLSNR for Linux: Version 19.0.0.0.0 - Production
System parameter file is /opt/oracle/product/19c/dbhome_1/network/admin/listener.ora
Log messages written to /opt/oracle/diag/tnslsnr/mydb-w970s/listener/alert/log.xml
Listening on: (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1)))
Listening on: (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=0.0.0.0)(PORT=1521)))

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC1)))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 19.0.0.0.0 - Production
Start Date                05-DEC-2021 20:59:43
Uptime                    0 days 0 hr. 0 min. 0 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File   /opt/oracle/product/19c/dbhome_1/network/admin/listener.ora
Listener Log File         /opt/oracle/diag/tnslsnr/mydb-w970s/listener/alert/log.xml
Listening Endpoints Summary...
```

```
Nelson_Cal@cloudshell:~ (us-ashburn-1)$  kubectl logs pod/mydb-w970s
[2021:12:05 20:59:43]: Acquiring lock .ORCL1.create_lck with heartbeat 30 secs
[2021:12:05 20:59:43]: Lock acquired
[2021:12:05 20:59:43]: Starting heartbeat
[2021:12:05 20:59:43]: Lock held .ORCL1.create_lck
ORACLE EDITION: ENTERPRISE

LSNRCTL for Linux: Version 19.0.0.0.0 - Production on 05-DEC-2021 20:59:43

Copyright (c) 1991, 2019, Oracle.  All rights

Starting /opt/oracle/product/19c/dbhome_1/bin

TNSLSNR for Linux: Version 19.0.0.0.0 - Produ
System parameter file is /opt/oracle/product/
Log messages written to /opt/oracle/diag/tnsl
Listening on: (DESCRIPTION=(ADDRESS=(PROTOCOL
Listening on: (DESCRIPTION=(ADDRESS=(PROTOCOL

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for Linux:
Start Date                05-DEC-2021 20:59:4
Uptime                    0 days 0 hr. 0 min.
Trace Level               off
Security                  ON: Local OS Authen
SNMP                      OFF
Listener Parameter File   /opt/oracle/product/19c/dbhome_1/network/admin/listener.ora
Listener Log File         /opt/oracle/diag/tnslsnr/mydb-w970s/listener/alert/log.xml
Listening Endpoints Summary...
```

```
Nelson_Cal@cloudshell:~ (us-ashburn-1)$  kubectl logs pod/mydb-w970s | tail
Copying database files
31% complete
Creating and starting Oracle instance
32% complete
36% complete
40% complete
43% complete
46% complete
Completing Database Creation
51% complete
...
Pluggable database PDB$SEED opened read write
Completed: alter pluggable database PDB$SEED open READ WRITE services=NONE
2021-12-05T21:16:28.384568+00:00
QPI: opatch file present, opatch
QPI: qopiprep.bat file present

Nelson_Cal@cloudshell:~ (us-ashburn-1)$
```

OraOperator Example - Single Instance

```
Nelson_Cal@cloudshell:~ (us-ashburn-1)$kubectl exec -it pods/mydb-w970s -- bash -i
[oracle@mydb-w970s ~]$ps -eaf | grep lsnr
oracle      51     1  0 20:59 ?        00:00:00 /opt/oracle/product/19c/dbhome_1/bin/tnslsnr LISTENER -inherit
oracle    5180  5161  0 21:23 pts/0    00:00:00 grep --color=auto lsnr
[oracle@mydb-w970s ~]$ps -eaf | grep pmon
oracle    3783     1  0 21:15 ?        00:00:00 ora_pmon_ORCL1
oracle    5182  5161  0 21:23 pts/0    00:00:00 grep --color=auto pmon
[oracle@mydb-w970s ~]$df -h
Filesystem      Size  Used Avail Use% Mounted on
overlay          39G   16G   23G  41% /
tmpfs            64M     0   64M   0% /dev
tmpfs           7.6G     0  7.6G   0% /sys/fs/cgroup
shm              64M     0   64M   0% /dev/shm
tmpfs           7.6G   12M  7.6G   1% /etc/hostname
/dev/sda3        39G   16G   23G  41% /etc/hosts
/dev/sdb         49G  3.8G   43G   9% /opt/oracle/oradata
tmpfs           7.6G   12K  7.6G   1% /run/secrets/kubernetes.io/serviceaccount
tmpfs           7.6G     0  7.6G   0% /proc/acpi
tmpfs           7.6G     0  7.6G   0% /proc/scsi
tmpfs           7.6G     0  7.6G   0% /sys/firmware
[oracle@mydb-w970s ~]$
```

OraOperator Example - Single Instance

```
Nelson_Cal@cloudshell:~ (us-ashburn-1)$kubectl exec -it pods/mydb-w970s -- bash -i
[oracle@mydb-w970s ~]$ps -eaf | grep lsnr
oracle      51     1  0 20:59 ?        00:00:00 /opt/oracle/product/19c/dbhome_1/bin/tnslsnr LISTENER -inherit
oracle    5180  5161  0 21:23 pts/0    00:00:00 grep --color=auto lsnr
[oracle@mydb-w970s ~]$ps -eaf | grep pmon
orac
orac
[ora
File
over
tmpf
tmpf
shm
tmpf
/dev
/dev
tmpf
tmpf
tmpf
tmpfs           7.6G     0  7.6G   0% /sys/firmware
[oracle@mydb-w970s ~]$
```

```
Nelson_Cal@cloudshell:~ (us-ashburn-1)$kubectl exec -it pods/mydb-w970s -- sqlplus system/my0Racl3@ORCL1

SQL*Plus: Release 19.0.0.0.0 - Production on Sun Dec 5 21:21:44 2021
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle.  All rights reserved.


Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0

SQL>
```

```
$ kubectl describe singleinstancedatabase

Name:         mydb
Namespace:    default
Labels:       <none>
Annotations:  <none>
API Version:  database.oracle.com/v1alpha1
Kind:         SingleInstanceDatabase
Metadata: ....
Spec:
  Admin Password:
    Secret Key:   mydb-admin-pwd
    Secret Name:  admin-pwd
  Charset:        AL32UTF8
  Edition:        enterprise
  Image:
    Pull From:     container-registry.oracle.com/database/enterprise:19.3.0.0
    Pull Secrets:  oracrs
  Init Params:
    Cpu Count:  1
    Processes:  100
  Pdb Name:     orclpdb1
  Persistence:
    Access Mode:    ReadWriteOnce
    Size:           20Gi
    Storage Class:  oci
  Replicas:         1
  Sid:              ORCL1
Status:
  Archive Log:             false
  Charset:                 AL32UTF8
  Clone From:              Unavailable
  Cluster Connect String:  mydb.default:1521/ORCL1
  Conditions:
    Last Transition Time:  2021-12-05T21:14:56Z
    Message:               could not execute: command terminated with exit code
1
    Observed Generation:   1
    Reason:                LastReconcileCycleQueued
    Status:                True
```

```
$ kubectl describe singleinstancedatabase

Name:         mydb
Namespace:    default
Labels:       <none>
Annotations:  <none>
API Version:  database.oracle.com/v1alpha1
Kind:         SingleInstanceDatabase
Metadata: ....
Spec:
  Admin Password:
    Secret Key:   mydb-admin-pwd
    Secret Name:  admin-pwd
  Charset:        AL32UTF8
  Edition:        enterprise
  Image:
    Pull From:     container-registry.oracle.com/database/enterprise:19.3.0.0
    Pull Secrets:  oracrs
  Init Params:
    Cpu Count:  1
    Processes:  100
  Pdb Name:     orclpdb1
  Persistence:
    Access Mode:    ReadWriteOnce
    Size:           20Gi
    Storage Class:  oci
  Replicas:         1
  Sid:              ORCL1
Status:
  Archive Log:             false
  Charset:                 AL32UTF8
  Clone From:              Unavailable
  Cluster Connect String:  mydb.default:1521/ORCL1
  Conditions:
    Last Transition Time:  2021-12-05T21:14:56Z
    Message:               could not execute: command terminated with exit cod
1
    Observed Generation:   1
    Reason:                LastReconcileCycleQueued
    Status:                True
```

```
...
    Type:                   ReconcileQueued
    Last Transition Time:   2021-12-05T21:16:04Z
    Message:                processing datapatch execution
    Observed Generation:    1
    Reason:                 LastReconcileCycleBlocked
    Status:                 True
    Type:                   ReconcileBlocked
    Last Transition Time:   2021-12-06T04:23:57Z
    Message:                no reconcile errors
    Observed Generation:    1
    Reason:                 LastReconcileCycleCompleted
    Status:                 True
    Type:                   ReconcileComplete
  Connect String:           10.0.10.87:31171/ORCL1
  Datafiles Created:        true
  Datafiles Patched:        true
  Edition:                  Enterprise
  Flash Back:               false
  Force Log:                false
  Init Params:
    Cpu Count:          1
    Processes:          100
  Oem Express URL:      https://10.0.10.87:31452/em
  Pdb Connect String:   10.0.10.87:31171/ORCLPDB1
  Pdb Name:             orclpdb1
  Persistence:
    Access Mode:    ReadWriteOnce
    Size:           20Gi
    Storage Class:  oci
  Release Update:   19.3.0.0.0 (29517242)
  Replicas:         1
  Role:             PRIMARY
  Sid:              ORCL1
  Status:           Healthy
Events:             <none>
Nelson_Cal@cloudshell:~ (us-ashburn-1)$
```

OraOperator - Single Instance

Patching

- create a patched image using patching extension
  https://github.com/oracle/docker-images/tree/main/OracleDatabase/SingleInstance/extensions/k8s

```
$ buildExtensions.sh -b ora19c
```

- specify a new release update for the image attribute. The database pods will be restarted with the new release update image:

```
$ kubectl --type=merge -p '{"spec":{"image": {"pullFrom":
"dbcs.dev-docker.dockerhub-phx.oci.oraclecorp.com/oracle/database:1911"}}}' patch
singleinstancedatabase ora19c


# after some time:

$ kubectl get singleinstancedatabase sidb-sample -o "jsonpath={.status.datafilesPatched}"

  true
```

OraOperator - ADB provisioning

Provisioning a new ADB:

- get compartment OCID from console
- create a K8s secret to store the password

```
$ kubctl create secret generic admin-password --from-literal=adb-admin-password=$YOURPWD
```

- prepare create_adb.yaml file including above details

```
apiVersion: database.oracle.com/v1alpha1
kind: AutonomousDatabase
metadata:
  name: ADB-test
spec:
  details:
    compartmentOCID: ocid1.compartment...
    dbName: myADB
    displayName: myADB
    cpuCoreCount: 1
    adminPassword:
      k8sSecretName: admin-password              # name of the secret defined above
    dataStorageSizeInTBs: 1
  ociConfig:
    configMapName: oci-cred
    secretName: oci-privatekey
```

OraOperator - ADB provisioning

Apply .yaml to create it:

```
$ kubectl apply -f create_adb.yaml
```

Check ADB status on OCI

```
$ oci db autonomous-database list \
   --compartment-id $COMP_ID \
   --output table \
   --query "data [*].{dbname:\"display-name\",state:\"lifecycle-state\"}”
+--------------+-----------+
| dbname       | state     |
+--------------+-----------+
| myADB        | AVAILABLE |
+--------------+-----------+
```

Check k8s status

```
$ kubectl get adb
NAME            STATE
myADB           AVAILABLE
```

OraOperator - ADB operations

To bind to an existing ADB instead of creating a new one: use ADB OCID instead of compartmenet OCID in .yaml file (in previous example):

```
details:
  autonomousDatabaseOCID: ocid1.autonomousdatabase...
```

To scale-up existing ADB:

```
...
spec:
  details:
     autonomousDatabaseOCID: ocid1.autonomousdatabase...
     cpuCoreCount: 2
...
```

To stop an ADB:

```
...
spec:
  details:
     autonomousDatabaseOCID: ocid1.autonomousdatabase...
     lifecycleState: STOPPED
...
```

OraOperator - Sharded DB

Few scenarios listed, several components in each:

- Primary GSM Pods gsm1 and standby GSM Pod gsm2
- Two sharding Pods: shard1 and shard2
- One Catalog Pod: catalog

Lot of details:
https://github.com/oracle/oracle-database-operator/tree/main/docs/sharding

ElCarro operator - example

If time available, can describe the work done for this:
https://github.com/GoogleCloudPlatform/elcarro-oracle-operator/blob/main/docs/content/aws.md

Thank You / Questions?

calero@pythian.com
@ncaleroy
http://www.linkedin.com/in/ncalero