



Revolutionising Database Deployment with the Oracle Database Operator

Ron Ekins

nIOUG & OBUG Amersfoort

ron@purestorage.com

16th October 2024

NL.OUG

NL ORACLE USER GROUP



Ron Ekins

Director Field Solutions Architecture, EMEA & LATAM

Email: ron@purestorage.com

X: [@RonEkins](#)

Blog: <http://ronekins.com>

GitHub: <http://github.com/raekins>



Who Are Pure Storage

A Leader for the Eleventh Consecutive Year*

Named A Leader again in the
2024 Gartner® Magic
Quadrant™ for Primary
Storage.

Figure 1: Magic Quadrant for Primary Storage Platforms



Gartner.



The Oracle ACE Program

400+ 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 levels: Director, Pro, and Associate
- Nominate yourself or a colleague at ace.oracle.com/nominate
- Learn more at ace.oracle.com



ACE Member Benefits



Key Benefits

Cool **swag***, Digital awards for social media, Oracle **CloudWorld pass***, & more



Direct Access to Product Management

Multiple **direct communication channels** to product management and fellow ACEs



Exclusive Content

Exclusive **monthly virtual meetings** with product development teams + engaging guest speakers



Networking

In-person & virtual networking opportunities for ACEs to **connect with product development** and each other.



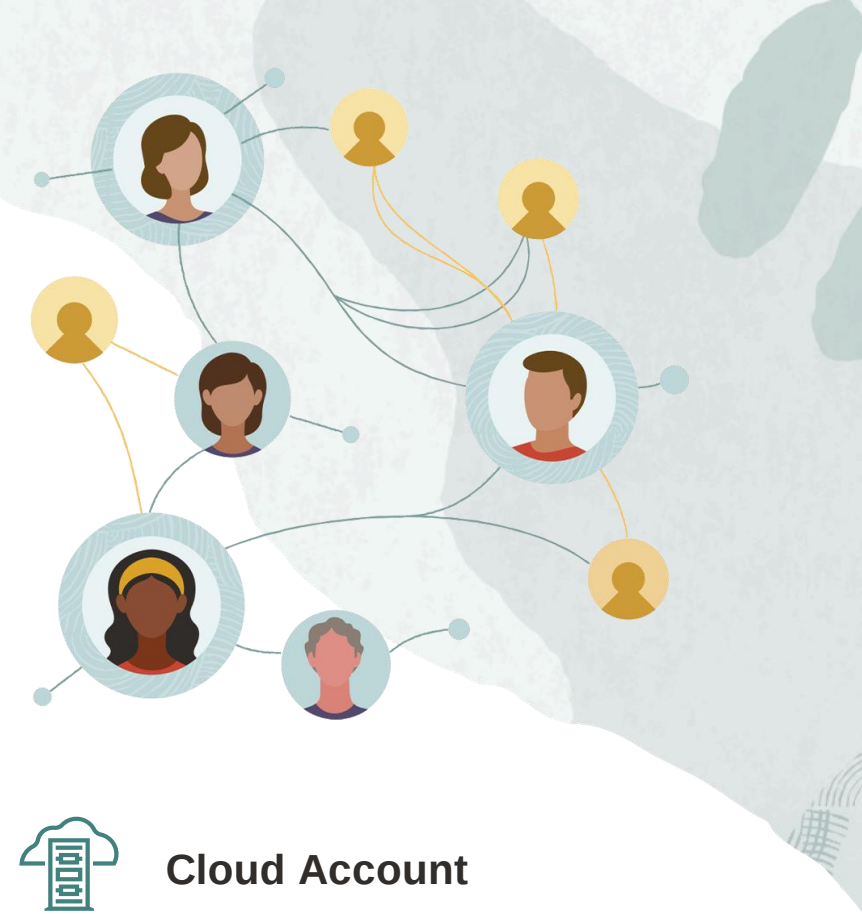
Cloud Account

\$5k USD Cloud account*



Travel Support

ACE Directors are eligible for travel support to give presentations or lead workshops at conferences globally



Agenda

- Introduction to Oracle and Containerization
- The Oracle Database Kubernetes Operator
- Oracle 23ai on OKE





Oracle and Containerization

Oracle Database on GitHub

Oracle provided Docker Build Scripts

- 23ai (23.5.0) FREE
- 21c (21.3.0) EE and SE2
- 19c (19.3.0) EE and SE2
- 18c (18.4.0) XE
- 18c (18.3.0) EE and SE2
- 12c Release 2 (12.2.0.2) EE and SE2
- 12c Release 1 (12.1.0.2) EE and SE2
- 11g Release 2 (11.2.0.2) XE

Oracle Database on Docker

Sample Docker build files to facilitate installation, configuration, and environment setup for DevOps users. For more information about Oracle Database please see the [Oracle Database Online Documentation](#).

SingleInstance

Provides Docker build files to create an Oracle Database Single Instance Docker image. For more details, see [SingleInstance/README.md](#).

Oracle Sharding

Provides terraform scripts to deploy Oracle Sharding in Oracle Cloud with Oracle Database Cloud Service, Docker build files and Sharding on OKE. For more details, see [oracle/db-sharding](#).

RAC

Provides Docker build files to create an Oracle RAC Database docker image. For more details, see [RAC/README.md](#).

<https://github.com/oracle/docker-images/tree/main/OracleDatabase>



Oracle Docker images

Oracle Container Registry



The screenshot shows the Oracle Container Registry interface. At the top, there's a header with the Oracle logo and 'Oracle Container Registry'. Below that, a breadcrumb trail reads 'Database Repositories \'. The main heading is 'Oracle Database Enterprise Edition'. Underneath, there's a 'Description' section. The title of the description is 'Oracle Database Server Release 19c Docker Image Documentation'. The text describes Oracle Database Server Release 19c as an industry leading relational database server, noting that the Docker image contains Oracle Database server Release 19c (19.3.0.0) Enterprise or Standard Edition running on Oracle Linux 7. It also mentions a default database in a multitenant configuration. At the bottom, it provides a link for more information: <http://docs.oracle.com/en/database/>.

ORACLE Oracle Container Registry

Database Repositories \

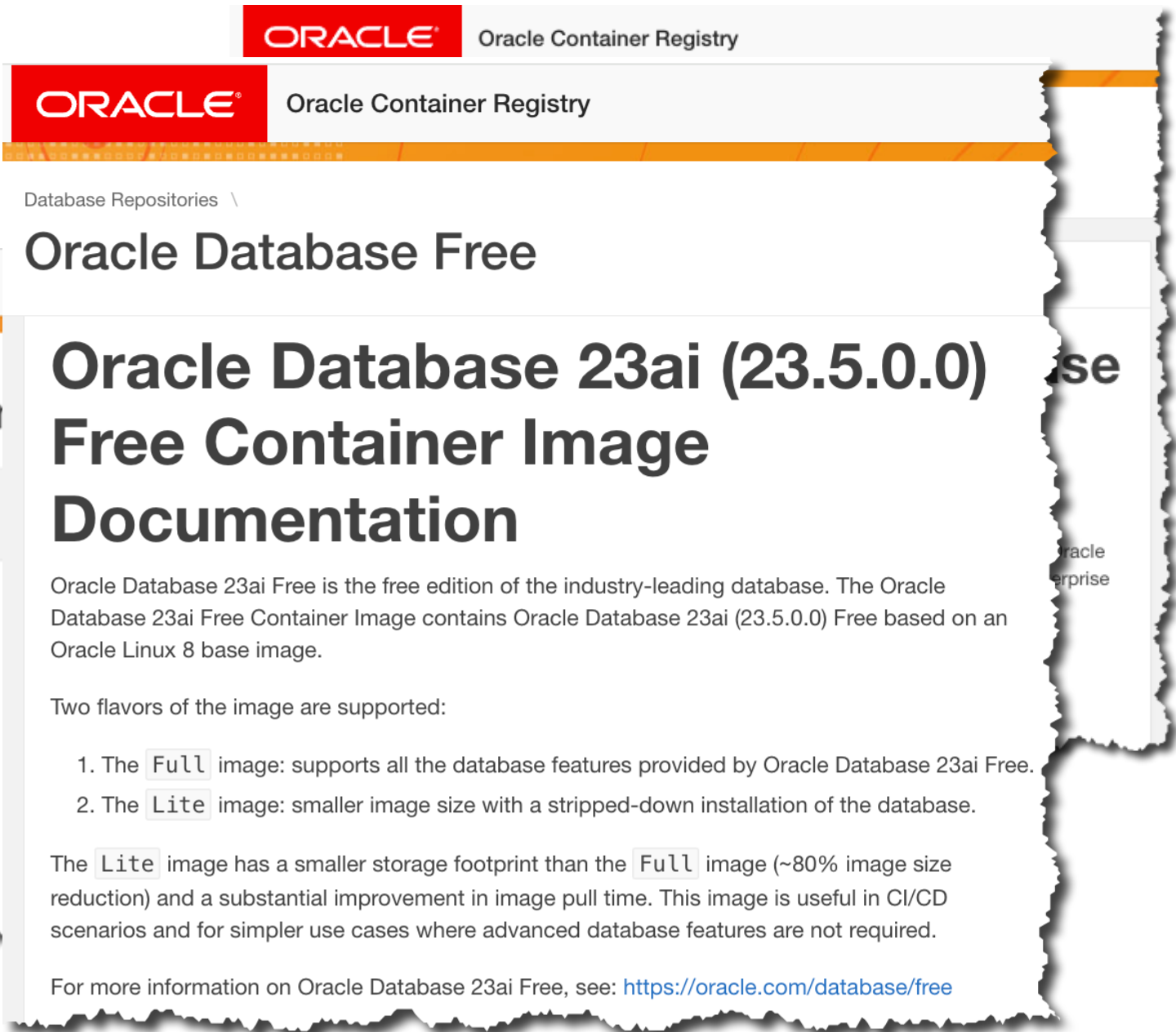
Oracle Database Enterprise Edition

Description

Oracle Database Server Release 19c Docker Image Documentation

Oracle Database Server Release 19c is an industry leading relational database server. The Oracle Database server docker image contains Oracle Database server Release 19c (19.3.0.0) Enterprise or Standard Edition running on Oracle Linux 7. This image contains a default database in a multitenant configuration with one pluggable database.

For more information on Oracle Database server Release 19c, refer to <http://docs.oracle.com/en/database/>.



The screenshot shows the Oracle Container Registry interface for Oracle Database 23ai Free. The header is identical to the previous screenshot. The breadcrumb trail is 'Database Repositories \'. The main heading is 'Oracle Database Free'. Below that, the title is 'Oracle Database 23ai (23.5.0.0) Free Container Image Documentation'. The text describes Oracle Database 23ai Free as the free edition of the industry-leading database, noting that the Docker image contains Oracle Database 23ai (23.5.0.0) Free based on an Oracle Linux 8 base image. It then lists two supported flavors: 'Full' (supports all features) and 'Lite' (smaller size, stripped-down installation). It also compares the 'Lite' image to the 'Full' image, noting a ~80% size reduction and faster pull time. At the bottom, it provides a link for more information: <https://oracle.com/database/free>.

ORACLE Oracle Container Registry

Database Repositories \

Oracle Database Free

Oracle Database 23ai (23.5.0.0) Free Container Image Documentation

Oracle Database 23ai Free is the free edition of the industry-leading database. The Oracle Database 23ai Free Container Image contains Oracle Database 23ai (23.5.0.0) Free based on an Oracle Linux 8 base image.

Two flavors of the image are supported:

1. The **Full** image: supports all the database features provided by Oracle Database 23ai Free.
2. The **Lite** image: smaller image size with a stripped-down installation of the database.

The **Lite** image has a smaller storage footprint than the **Full** image (~80% image size reduction) and a substantial improvement in image pull time. This image is useful in CI/CD scenarios and for simpler use cases where advanced database features are not required.

For more information on Oracle Database 23ai Free, see: <https://oracle.com/database/free>

Oracle Support for Database Running in Docker

MOS Doc ID 2216342.1

APPLIES TO:

Oracle Database - Enterprise Edition - Version 12.1.0.2 and later
Oracle Database - Standard Edition - Version 12.1.0.2 and later
Linux x86-64

PURPOSE

Clarify Oracle's support of Oracle Database running on Docker

SCOPE

For customers running Oracle Database (single instance configuration) in Docker containerized environments.

DETAILS

Oracle plans to certify the latest versions of Oracle Database to run in Docker containers which are built and supported with Oracle Linux as the host.

Additionally, Oracle does support customers running Oracle Database (single instance) in Docker containers running on Oracle Linux 7 with UEK4 (and later) or Red Hat Enterprise Linux 7. Docker binaries are available in the Addons channel for Oracle Linux. Details on Installation can be found in Chapter 2 of the [Oracle Linux 7 Docker Users Guide](#). Other details on support for Docker on Oracle Linux can be found in [Support for Docker Running on Oracle Linux \(Doc ID 1921163.1\)](#).

For information on Oracle RAC on Docker, see Note [2488326.1](#) Oracle RAC on Docker - Released Versions and Known Issues

If you are planning to deploy Oracle database on Oracle Linux 8, you need to use [Podman](#) with UEK6. Podman binaries are available in the Oracle Linux 8 on ULN and the Oracle Linux yum server. You can install all of these packages by installing the container-tools module using the dnf command. Podman in Oracle Linux 8 is a drop-in replacement for Oracle Container Runtime for Docker. However, Docker-produced images will continue to work with Open Container Initiative compatible container runtimes.

Additional information which might be helpful:

- [Support for Docker Running on Oracle Linux \(Doc ID 1921163.1\)](#)
- Information on Docker for Oracle Linux can be found at [Oracle Linux Cloud Native Environment](#)
- [Certified Virtualization and Partitioning Technologies for Oracle Database and RAC Product Releases](#)
- [Creating an Oracle Database Docker Image](#) (blog post)



Oracle RAC Database Support Now includes 21.3.0/19.16.0

MOS Doc ID 2488326.1

APPLIES TO:					
Oracle Database - Enterprise Edition - Version 12.2.0.1 and later Linux OS - Version Oracle Linux 7.5 with Unbreakable Enterprise Kernel [4.14.35] to Oracle Linux 7.9 with Unbreakable Enterprise Kernel [5.4.17] [Release OL7U5 to OL7U9] Linux x86-64					
PURPOSE	Released Versions	Available on GitHub	Available on OCR	Oracle Documentation Link	Support
This article is a	21.3.0	Yes	Yes	https://docs.oracle.com/en/database/oracle/oracle-database/21/racdk/index.html	Production
SCOPE					
The contents c	19.16.0	Yes	No	https://docs.oracle.com/en/database/oracle/oracle-database/19/racdk/	Production
DETAILS					
Docker lets you virtually anywhere and overcomes the Docker provide	19.3.0	Yes	Yes	None	Dev and Test
Related Repe	18.3.0	Yes	Yes	None	Dev and Test
	12.2.0	Yes	Yes	None	Dev and Test



Deploy Oracle 19c database in Kubernetes

With Helm

- Oracle Database Docker images can be deployed in Kubernetes using Helm charts
 - Charts describe the application structure so that Helm can install and configure the pieces of the application
 - e.g. RAC and sharding
 - Install Oracle database in Kubernetes cluster with **helm install**
- Oracle Database Helm charts are published on GitHub
 - Supports customisation and extension



Oracle Helm Install

- The Oracle 19c database Helm chart creates a multi-tenant, single pdb Oracle Database 19c running on Oracle Linux 7.
- Clone GitHub repo

```
% git clone https://github.com/oracle/docker-images.git
```

```
Cloning into 'docker-images'... remote: Enumerating objects: 15267, done.
```

```
remote: Counting objects: 100% (129/129), done. remote: Compressing objects: 100% (96/96), done.
```

```
remote: Total 15267 (delta 48), reused 73 (delta 23), pack-reused 15138
```

```
Receiving objects: 100% (15267/15267), 10.23 MiB | 4.70 MiB/s, done.
```

```
Resolving deltas: 100% (8973/8973), done.
```

```
% helm package docker-images/OracleDatabase/SingleInstance/helm-charts/oracle-db
```

```
Successfully packaged chart and saved it to: /root/oracle-db-1.0.0.tgz
```

- Helm Dry Run Install

```
% helm install db19c -f values.yaml oracle-db-1.0.0.tgz --dry-run
```

- Helm Install

```
% helm install db19c -f values.yaml oracle-db-1.0.0.tgz
```



Running Oracle 23ai on Mac with Colima

```
rekins@rekins--Mac14 ~ % colima start \
--arch x86_64 \
--vm-type \
--vz-ros \
--mount- \
--memory
INFO[0000] s
INFO[0000] r
INFO[0000] s
INFO[0062] p
INFO[0063] s
INFO[0069] d
rekins@rekins--Mac14 ~ % sql system/oracle@//localhost:1522/FREE

SQLcl: Release 24.1 Production on Tue Jul 02 11:31:33 2024

Copyright (c) 1982, 2024, Oracle. All rights reserved.

Last Successful login time: Tue Jul 02 2024 11:31:34 +01:00

Connected to:
Oracle Database 23ai Free Release 23.0.0.0.0 - Develop, Learn, and Run for Free
Version 23.4.0.24.05

SQL> select INSTANCE_NAME, HOST_NAME, VERSION_FULL from v$instance;

INSTANCE_NAME      HOST_NAME          VERSION_FULL
-----
FREE               d0b7e66bb8fc      23.4.0.24.05

SQL>
```



Why Kubernetes Operators

What is a Kubernetes Operator



Kubernetes Operators provide a way to package, deploy, and manage a Kubernetes application.



Kubernetes Operators help in building cloud-native applications by automating deployment, scaling, backup and location independent restores.



Kubernetes Operators provide a way of delivering consistent deployments across cloud vendors and on-premises.

OperatorHub.IO

38 Databases

 Aerospike Kubernetes Operator provided by Aerospike The Aerospike Kubernetes Operator automates the	 Altinity Operator for ClickHouse provided by Altinity ClickHouse Operator manages full lifecycle of ClickHouse	 ArangoDB provided by ArangoDB GmbH ArangoDB Kubernetes Operator	 Cassandra provided by Instaclustr Manage the full lifecycle of the Cassandra clusters.	 Cloud Native PostgreSQL provided by EnterpriseDB Corporation Operator to manage Postgres high availability clusters with	 MongoDB Operator provided by Opstree Solutions A golang based operator to create and manage MongoDB standalone setup and cluster	 Operator for Apache CouchDB provided by IBM Apache CouchDB is a highly available NOSQL database	 Percona Distribution for MongoDB Operator provided by Percona Percona Distribution for MongoDB Operator	 Percona Distribution for MySQL Operator provided by Percona Percona Distribution for MySQL Operator manages	 Percona Operator for PostgreSQL provided by Percona Percona Operator for PostgreSQL manages the
 CockroachDB Helm Operator provided by Helm Community CockroachDB Operator based on the CockroachDB helm	 Couchbase Operator provided by Couchbase The Couchbase Autonomous Operator allows users to easily deploy, manage, and maintain	 Crunchy Postgres for Kubernetes provided by Crunchy Data Production Postgres Made Easy	 DataStax Kubernetes Operator for Apache Cassandra provided by DataStax Simple provisioning, turn-key	 druid-operator provided by Apache Druid Community druid-operator allows creating Apache Druid cluster and	 PlanetScale Operator for Viteess provided by PlanetScale PlanetScale's operator for Viteess deploys and manages	 Postgres-Operator provided by Zalando SE Postgres operator creates and manages PostgreSQL clusters running in Kubernetes.	 Postgresql Operator provided by Openlabs Deploys postgresql based applications	 PostgreSQL Operator by Dev4Devs.com provided by Dev4Devs.com Operator in Go developed using the Operator Framework	 Qserv operator provided by Vera C. Rubin Observatory Create and maintain highly-available Qserv clusters on
 Elasticsearch (ECK) Operator provided by Elastic Run Elasticsearch, Kibana, APM Server, Beats, Enterprise	 etcd provided by CNCF Create and maintain highly-available etcd clusters on Kubernetes	 Ext Postgres Operator provided by movetokube.com Manage databases and roles in external PostgreSQL server or cluster	 GrdsCloud MySQL for Kubernetes provided by Grds Cloud Enterprise open source MySQL-as-a-Service	 Hazelcast Platform Operator provided by Hazelcast, Inc Install Hazelcast clusters in Kubernetes environments.	 Redis Enterprise provided by Redis Labs, Inc An operator to run Redis Enterprise Clusters	 Redis Operator provided by Opstree Solutions A Golang based redis operator that will make/oversee Redis standalone/cluster mode setup	 Robin Cloud Native Storage provided by Robin.io Robin Cloud Native Storage operator enables advanced data management capabilities	 Steerd Presto Operator provided by Falarica Analytics Private Limited SteerD Presto Operator is a Kubernetes Operator for	 TiDB Operator provided by PingCAP TiDB Operator manages TiDB clusters on Kubernetes and automates tasks related to
 Infinispan Operator provided by Infinispan Create and manage Infinispan clusters.	 Logging Operator provided by Opstree Solutions A golang based CRD operator to set up and manage logging stack	 Mariadb Operator provided by Manoj Dhanorkar Create mariadb server and database on Kubernetes	 MongoDB Atlas Operator provided by MongoDB, Inc The MongoDB Atlas Kubernetes Operator enables easy management of Clusters	 MongoDB Enterprise Operator provided by MongoDB, Inc The MongoDB Enterprise Kubernetes Operator enables	 Upbound Universal Crossplane (UXP) provided by Upbound Inc. Upbound Universal Crossplane (UXP) is Upbound's official	 VerticaDB Operator provided by Vertica Operator that manages a Vertica Eon Mode database.	 Yugabyte Operator provided by YugabyteDB This operator installs and manages a YugabyteDB cluster.		

<https://operatorhub.io/?category=Database>



Multiple Operators Offering Different Features



Crunchy Postgres for Kubernetes

Production Postgres Made Easy

[Home](#) > Crunchy Postgres for Kubernetes

Crunchy Postgres for Kubernetes

PGO, the [Postgres Operator](#) from [Crunchy Data](#), gives you a declarative Postgres solution. Designed for your GitOps workflows, it is [easy to get started](#) with Postgres on Kubernetes cluster complete with high availability, disaster recovery, and monitoring, all over seconds cluster to tailor it to your workload!

With conveniences like cloning Postgres clusters to using rolling updates to roll out data at every stage of your release pipeline. Built for resiliency and uptime, PGO will keep your data always available.

PGO is developed with many years of production experience in automating Postgres to keep your data always available.

- **PostgreSQL Cluster Provisioning:** [Create, Scale, & Delete PostgreSQL clusters](#)
- **High-Availability:** Safe, automated failover backed by a [distributed consensus](#) to configure how aggressive this can be! Failed primaries automatically heal, allow and set your backup retention policy
- **Disaster Recovery:** [Backups and restores](#) leverage the open source [pgBackRest](#) [efficient delta restores](#). Set how long you want your backups retained for. Works across multiple Kubernetes clusters.
- **Monitoring:** [Track the health of your PostgreSQL clusters](#) using the open source [Percona Monitoring and Management \(PMM\)](#)
- **Clone:** [Create new clusters from your existing clusters or backups](#) with efficient [pgBouncer](#)
- **TLS:** All connections are over [TLS](#). You can also [bring your own TLS infrastructure](#)
- **Connection Pooling:** Advanced [connection pooling](#) support using [pgBouncer](#)
- **Affinity and Tolerations:** Have your PostgreSQL clusters deployed to [Kubernetes](#) and more rules to customize your deployment topology!
- **Full Customizability:** Crunchy PostgreSQL for Kubernetes makes it easy to get deployments, including:
 - Choose the resources for your Postgres cluster: [container resources](#) and [imagePullSecrets](#)
 - Use your own container image repository, including support [imagePullSecrets](#)
 - [Customize your PostgreSQL configuration](#)

and much more!



Percona Operator for PostgreSQL

Percona Operator for PostgreSQL manages the lifecycle of Percona PostgreSQL clusters

[Home](#) > Percona Operator for PostgreSQL

Percona Operator for PostgreSQL

Percona is Cloud Native

Percona Operator for PostgreSQL automates and simplifies deploying and managing open source PostgreSQL based on Postgres Operator developed by Crunchy Data. Whether you need to get a simple PostgreSQL cluster tolerant cluster in production, or are running your own database-as-a-service, the Operator provides the Consult the [documentation](#) on the Percona Operator for PostgreSQL for complete details on capabilities

Supported Features

- **PostgreSQL Cluster Provisioning** Create, Scale, & Delete PostgreSQL clusters with ease, while full
- **High Availability**
Safe, automated failover backed by a distributed consensus based high-availability solution. Uses P enforced this can be. Failed primaries automatically heal, allowing for faster recovery time. Support across multiple Kubernetes clusters.
- **Disaster Recovery**
Backups and restores leverage the open source [pgBackRest](#) utility and includes support for full, incremental restores. Set how long you want your backups retained for. Works great with very large databases!
- **Communication Security**
Secure communication between your applications and data servers by enabling TLS for your PostgreSQL connections to use TLS.
- **Monitoring**
Track the health of your PostgreSQL clusters with Percona Monitoring and Management (PMM).
- **PostgreSQL User Management**
Quickly add and remove users from your PostgreSQL clusters with powerful commands. Manage p



Cloud Native PostgreSQL

Operator to manage Postgres high availability clusters with a primary/standby architecture.

[Home](#) > Cloud Native PostgreSQL

Cloud Native PostgreSQL

Cloud Native PostgreSQL is an operator designed by EnterpriseDB to manage PostgreSQL workloads on any supported Kubernetes cluster running in private, public, or hybrid cloud environments, including OpenShift. Cloud Native PostgreSQL adheres to DevOps principles and concepts such as declarative configuration and immutable infrastructure.

It defines a new Kubernetes resource called "Cluster" representing a PostgreSQL cluster made up of a single primary and an optional number of replicas that co-exist in a chosen Kubernetes namespace for High Availability and offloading of read-only queries.

Applications that reside in the same Kubernetes cluster can access the PostgreSQL database using a service which is solely managed by the operator, without having to worry about changes of the primary role following a failover or a switchover. Applications that reside outside the Kubernetes cluster need to configure an Ingress object to expose the service via TCP.

Cloud Native PostgreSQL works with PostgreSQL (14, 13, 12, 11 and 10) and EDB Postgres Advanced (14, 13, 12, 11 and 10), and it is available under the EnterpriseDB Limited Use License. You can evaluate Cloud Native PostgreSQL for free. You need a valid license key to use Cloud Native PostgreSQL in production.

Main features

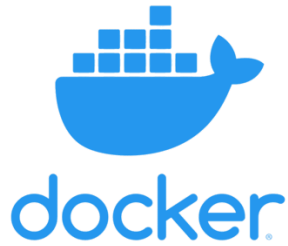
- Direct integration with Kubernetes API server for High Availability, without requiring an external tool
- Self-Healing capability, through:
 - failover of the primary instance by promoting the most aligned replica
 - automated recreation of a replica
- Planned switchover of the primary instance by promoting a selected replica
- Scale up/down capabilities
- Definition of an arbitrary number of instances (minimum 1 - one primary server)
- Definition of the *read-write* service, to connect your applications to the only primary server of the cluster
- Definition of the *read-only* service, to connect your applications to any of the instances for reading workloads
- Declarative management of PostgreSQL configuration, including certain popular Postgres extensions through the cluster spec: `pg_audit`, `auto_explain`, and `pg_stat_statements`
- Support for Local Persistent Volumes with PVC templates
- Reuse of Persistent Volumes storage in Pods
- Rolling updates for PostgreSQL minor versions
- In-place or rolling updates for operator upgrades
- TLS connections and client certificate authentication
- Support for custom TLS certificates (including integration with cert-manager)
- Continuous backup to an object store (AWS S3 and S3-compatible, Azure Blob Storage, and Google Cloud Storage)
- Backup retention policy (the backup retention window)



The Oracle Database Operator for Kubernetes

Aka 'OraOperator'

The Evolution of Containerisation for the Oracle Database



Docker

- Running Containers on Docker Engine
- Oracle Images (Single Instance, Sharding, Oracle RAC & CMAN)
- Standalone Environments
- OL7 Support
- Docker Compose



Podman

- Alternative to Docker
- Secure
- Oracle Images (Single Instance, Sharding, Oracle RAC & CMAN)
- Standalone Environments
- OL8 Support
- Podman Compose



Kubernetes

- Container Orchestration
- Software Defined Networking
- Observability and Scaling
- Single Instance & Sharding Support
- Helm



DB Operator

- Database-specific Controller
- Software extension of the Kubernetes APIs
- ADB, BaseDB, Single Instance database, Sharding & Multitenant
- Go



Containerized Oracle Real Application Clusters (RAC) on Kubernetes



Oracle extends its commitment to making Oracle RAC easily accessible via:

- Oracle RAC in Docker Containers

- Available for Oracle 19c, 21c and 23ai
 - <https://blogs.oracle.com/maa/post/oracle-rac-on-docker-now-with-full-production-support>

- Oracle RAC on Podman

- Podman images allow for a fast, lightweight deployment of Oracle RAC 19c, 21c, 23ai
 - First container environment to support Oracle Linux 8
- A base Oracle RAC Podman image is available on GitHub/the Oracle Container Registry
 - Customers can manually apply RUs to the base image and create custom images as needed



- Oracle RAC on Kubernetes

- Kubernetes orchestrates the deployment of Oracle RAC container images across pods
- Initial support on Oracle Linux Cloud Native Environment (OLCNE) with Oracle RAC 23ai
- **Oracle Operator for Oracle RAC on Kubernetes coming soon**



Oracle Database Operator for Kubernetes v1.1.0

Make Oracle Database Kubernetes Native

As part of Oracle's resolution to make Oracle Database Kubernetes native (that is, observable and operable by Kubernetes), Oracle released *Oracle Database Operator for Kubernetes* (`OraOperator` or the operator). OraOperator extends the Kubernetes API with custom resources and controllers for automating Oracle Database lifecycle management.

In this v1.1.0 production release, `OraOperator` supports the following database configurations and infrastructure:

- Oracle Autonomous Database:
 - Oracle Autonomous Database shared Oracle Cloud Infrastructure (OCI) (ADB-S)
 - Oracle Autonomous Database on dedicated Cloud infrastructure (ADB-D)
 - Oracle Autonomous Container Database (ACD) (infrastructure) is the infrastructure for provisioning Autonomous Databases.
- Containerized Single Instance databases (SIDB) deployed in the Oracle Kubernetes Engine (OKE) and any k8s where OraOperator is deployed
- Containerized Sharded databases (SHARDED) deployed in OKE and any k8s where OraOperator is deployed
- Oracle Multitenant Databases (CDB/PDBs)
- Oracle Base Database Cloud Service (BDBCS)
- Oracle Data Guard (Preview status)
- Oracle Database Observability (Preview status)

Oracle will continue to extend `OraOperator` to support additional Oracle Database configurations.

New in V1.1.0 Release

- Enhanced security with namespace scope deployment option
- Support for Oracle Database 23ai Free (with SIDB)
- Automatic Storage Expansion for SIDB and Sharded DB
- User-Defined Sharding
- TCPS support customer provided certs
- Execute custom scripts during DB setup/startup
- Patching for SIDB Primary/Standby in Data Guard
- Long-term backup for Autonomous Databases (ADB): Support for [long-term retention backup](#) and removed support for the deprecated mandatory backup
- Wallet expiry date for ADB: A user-friendly enhancement to display the wallet expiry date in the status of the associated ADB
- Wait-for-Completion option for ADB: Supports `kubectl wait` command that allows the user to wait for a specific condition on ADB
- OKE workload Identify: Supports OKE workload identity authentication method (i.e., uses OKE credentials). For more details, refer to [Oracle Autonomous Database \(ADB\) Prerequisites](#)
- Database Observability (Preview - Metrics)



Oracle Database 23ai Free

Requires v1.0.1

Install cert-manager v1.14.2 (OKE default v1.14.2)

- `kubectl apply -f https://github.com/cert-manager/cert-manager/releases/download/v1.14.2/cert-manager.yaml`

Install Database Operator v1.0.1 (OKE default v1.0.0)

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

Install CertManager and Database Operator

Available in GitHub

- Install Certificate Manager
 - The Database Operator uses webhooks for validating user input before persisting it in etcd. Webhooks require TLS certificates that are generated and managed by a certificate manager.

```
kubectl apply -f https://github.com/cert-manager/cert-manager/releases/download/v1.14.4/cert-manager.yaml
```

- Pull Oracle Database Operator for Kubernetes
 - Pull the latest Oracle Database Operator for Kubernetes from the Oracle GitHub Repo

```
git clone https://github.com/oracle/oracle-database-operator.git
```

OKE OraOperator Installation

OCI Developer Services

Kubernetes Clusters (OKE)

The screenshot displays the Oracle Cloud Developer Services console. At the top, the Oracle Cloud logo is on the left, followed by a search bar containing the text "Search resources, services, documentation, and Marketplace". On the right of the header, it shows the region "UK South (London)" and several utility icons. The left sidebar contains a navigation menu with items: Home, Compute, Storage, Networking, Oracle Database, Databases, Analytics & AI, **Developer Services** (highlighted with a dashed border), Identity & Security, Observability & Management, Hybrid, and Migration & Disaster Recovery. The main content area is titled "<> Developer Services" and is divided into four columns. The first column, "Containers & Artifacts", includes "Kubernetes Clusters (OKE)" (marked with a star), "Container Instances", "Container Registry", "Artifact Registry", "Service Mesh", "Functions" (with sub-items "Applications" and "Pre-Built Functions"), "APEX Application Development" (with "APEX Instances"), and "Database Tools" (with "Connections" and "SQL Worksheet"). The second column, "DevOps", includes "Overview", "Projects", "Resource Manager" (with "Overview", "Stacks", "Jobs", "Private Templates", "Configuration Source Providers", and "Private Endpoints"), and "App Dependency Management" (with "Overview", "Knowledge Bases", and "Vulnerability Audits"). The third column, "Related services", lists "Compute", "VMware Solution", "Autonomous Database", "MySQL HeatWave", "Oracle NoSQL Database", "Logging", "Application Performance Monitoring", "Streaming", and "Connector Hub". The fourth column, "Help", lists "OCI Developer Guide", "OCI Developer Tutorials", "Set up CLI", "REST APIs", "Developer Community", and "Developer Live". At the bottom of the console, there are three featured cards: "Predict the result of the next race" (with a "RACING" icon), "Deploy a WordPress website" (with a "6-8 mins" timer), and "Deploy a low-code app on Autonomous Database using APEX" (with a "New Marketplace" badge). The footer contains links for "Terms of Use and Privacy" and "Cookie Preferences", a copyright notice "Copyright © 2023, Oracle and/or its affiliates. All rights reserved.", and the slogan "Uncomplicate Data Storage, Forever".

Oracle Database Operator v1.0.0 available as OKE Add-On

Edit Database Operator [Help](#)

☒ **Enable Database Operator** ⓘ
OraOperator extends the Kubernetes API with custom resources and controllers for automating Oracle Database lifecycle management.

☐ **Automatic Updates**
Select to automatically update your addon

☒ **Choose a version**
Select a specific addon version

v1.1.0 ▼

Add configuration

Native pod networking
OCI VCN-Native Pod Networking CNI plugin. Enabled

Database Operator
OraOperator extends the Kubernetes API with custom resources and controllers for automating Oracle Database lifecycle management. Enabled

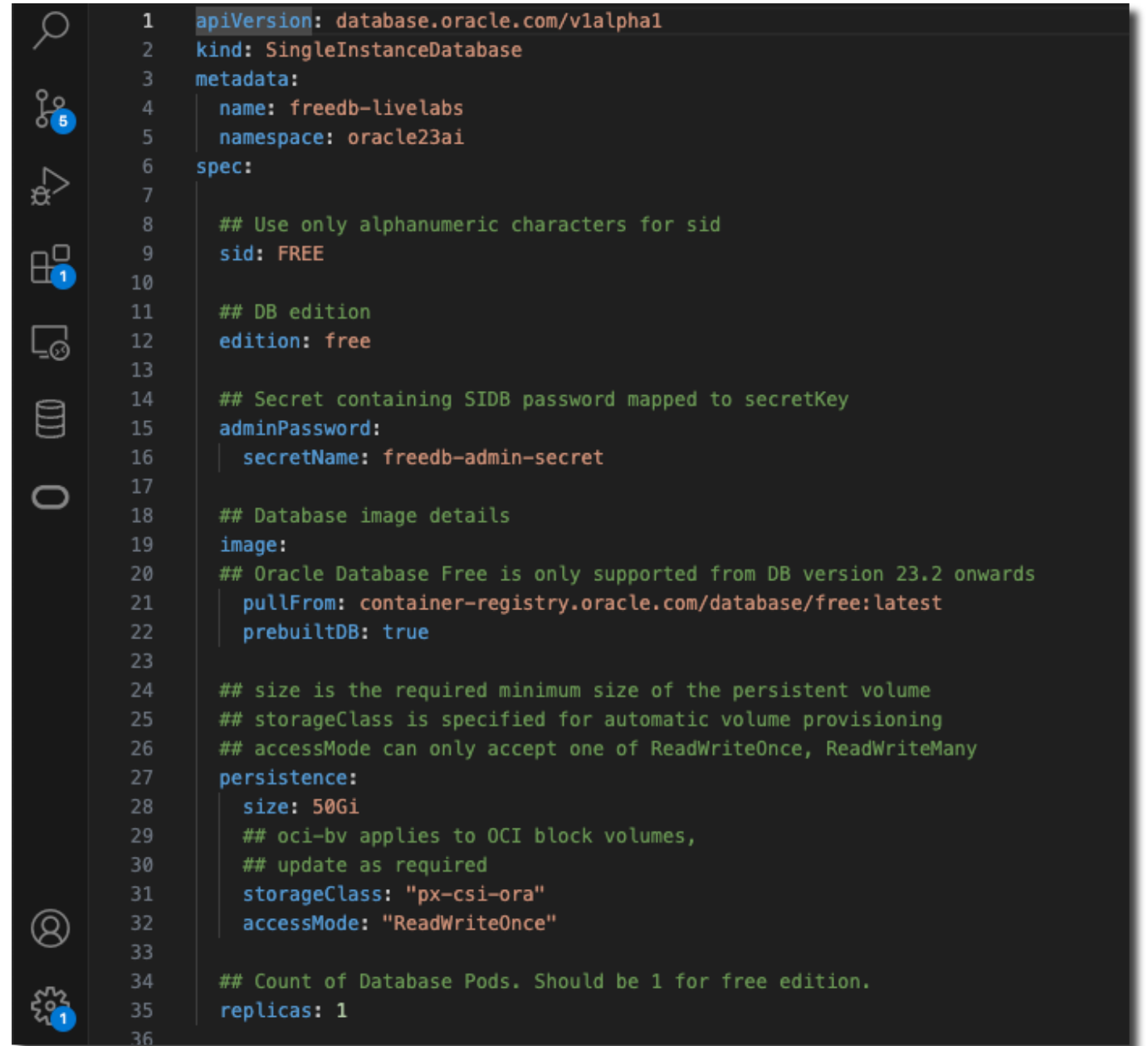
Oracle Database 23ai on OKE using OraOperator

Using OraOperator v1.1.0

Configure database

23ai Free

- Localise configuration:
 - *name*
 - *namespace*
 - *secretName*
 - *storageClass*
 - *loadBalancer*



```
1  apiVersion: database.oracle.com/v1alpha1
2  kind: SingleInstanceDatabase
3  metadata:
4    name: freedb-livelabs
5    namespace: oracle23ai
6  spec:
7
8    ## Use only alphanumeric characters for sid
9    sid: FREE
10
11    ## DB edition
12    edition: free
13
14    ## Secret containing SIDB password mapped to secretKey
15    adminPassword:
16      secretName: freedb-admin-secret
17
18    ## Database image details
19    image:
20      ## Oracle Database Free is only supported from DB version 23.2 onwards
21      pullFrom: container-registry.oracle.com/database/free:latest
22      prebuiltDB: true
23
24    ## size is the required minimum size of the persistent volume
25    ## storageClass is specified for automatic volume provisioning
26    ## accessMode can only accept one of ReadWriteOnce, ReadWriteMany
27    persistence:
28      size: 50Gi
29      ## oci-bv applies to OCI block volumes,
30      ## update as required
31      storageClass: "px-csi-ora"
32      accessMode: "ReadWriteOnce"
33
34    ## Count of Database Pods. Should be 1 for free edition.
35    replicas: 1
36
```

Provision containerized Oracle Database 23ai Free

- Provision database

`kubectl create -f singleinstancedatabase_free.yaml`

```
rekins@rekins--Mac14 23ai % kubectl create -f singleinstancedatabase_free.yaml
singleinstancedatabase.database.oracle.com/freedb-livelabs created
rekins@rekins--Mac14 23ai %
rekins@rekins--Mac14 23ai %
```

- Check pod status

`kubectl get pods -n <namespace>`

```
rekins@rekins--Mac14 23ai % kubectl get pods -n oracle23ai
NAME                                READY STATUS RESTARTS AGE
freedb-livelabs-bmu14              1/1   Running 0      4m2s
rekins@rekins--Mac14 23ai %
```

Logs

Oracle Database 23ai Free

- Use kubectl logs to view the database alert.log file
Kubectl logs <podname> -n <namespace>
- Use *–follow* to view database installation progress.
- Check
 - Database name
 - Oracle version
 - Status
 - ...

```
-zsh
rekins@rekins--Mac14 23ai % kubectl logs freedb-livelabs-bmu14 freedb-livelabs -n oracle23ai
Starting Oracle Net Listener.
Oracle Net Listener started.
Starting Oracle Database instance FREE.
Oracle Database instance FREE started.

The Oracle base remains unchanged with value /opt/oracle

SQL*Plus: Release 23.0.0.0.0 - Production on Fri Jun 7 10:00:17 2024
Version 23.4.0.24.05

Copyright (c) 1982, 2024, Oracle. All rights reserved.

Connected to:
Oracle Database 23ai Free Release 23.0.0.0.0 - Develop, Learn, and Run for Free
Version 23.4.0.24.05

SQL>
User altered.

SQL>
User altered.

SQL>
Session altered.

SQL>
User altered.

SQL> Disconnected from Oracle Database 23ai Free Release 23.0.0.0.0 - Develop, Learn, and Run for Free
Version 23.4.0.24.05
The Oracle base remains unchanged with value /opt/oracle
#####
DATABASE IS READY TO USE!
#####
The following output is now a tail of the alert.log:
2024-06-07T10:00:17.780564+00:00
```

Check Oracle Database 23ai Free

Using kubectl

- Check Kubernetes services

```
rekins@rekins--Mac14 23ai % kubectl get services -n oracle23ai
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
freedb-livelabs	ClusterIP	10.96.152.96	<none>	1521/TCP	5m54s
freedb-livelabs-ext	NodePort	10.96.189.1	<none>	5500:30918/TCP,1521:30127/TCP	5m54s

```
rekins@rekins--Mac14 23ai %
```

- Check Kubernetes Persistent Volume Claim (PVC)

`kubectl get pvc -n <namespace>`

```
rekins@rekins--Mac14 23ai % kubectl get pvc -n oracle23ai
```

NAME	STATUS	VOLUME	CAPACITY	ACCESS MODES	STORAGECLASS	VOLUMEATTRIBUTESCLASS	AGE
freedb-livelabs	Bound	pvc-e9497978-e0d7-4470-a389-c3a3d5f907cb	50Gi	RWO	px-csi-ora	<unset>	5m4s

```
rekins@rekins--Mac14 23ai %
```

Oracle Database 23ai Free status

Using OraOperator

- Watch State change from Pending -> Creating -> Patching -> Healthy
`watch kubectl get singleinstancedatabase -n <namespace>`

```
watch
Every 2.0s: kubectl get singleinstancedatabase -n oracle23ai
NAME          EDITION  STATUS  ROLE    VERSION    CONNECT STR          TCPS CONNECT STR  OEM EXPRESS URL
freedb-livelabs  Free     Healthy PRIMARY  23.4.0.24.05  10.0.10.139:30127/FREE  Unavailable       Unavailable
```

- Get database status
`kubectl get singleinstancedatabase freedb-livelabs -n <namespace> -o "jsonpath={.status.status}"`

```
-zsh
rekins@rekins--Mac14 23ai % kubectl get singleinstancedatabase freedb-livelabs -n oracle23ai -o "jsonpath={.status.status}"
Healthy
rekins@rekins--Mac14 23ai %
```



Oracle Database 23ai Free details

- Get database name

`export ORACLE_SID=$(kubectl get singleinstancedatabase -o 'jsonpath={.items[0].metadata.name}' -n <namespace>)`

```
rekins@rekins--Mac14 23ai % export ORACLE_SID=$(kubectl get singleinstancedatabase -o 'jsonpath={.items[0].metadata.name}' -n oracle23ai)
rekins@rekins--Mac14 23ai % echo $ORACLE_SID
freedb-livelabs
rekins@rekins--Mac14 23ai %
```

- Get database connection string

`kubectl get singleinstancedatabase freedb-livelabs -n oracle23ai -o "jsonpath={.status.connectString}"`

```
rekins@rekins--Mac14 23ai % kubectl get singleinstancedatabase freedb-livelabs -n oracle23ai -o "jsonpath={.status.connectString}"
10.0.10.139:30127/FREEX
rekins@rekins--Mac14 23ai %
```

Connect to Oracle Database 23ai free

Connecting via Kubernetes Database Container

```
kubectl exec -it pods/${ORA_POD} -n oracle23ai -- sqlplus sys/${ORA_PASS}@${ORACLE_SID} as sysdba
```

```
kubectl
rekins@rekins--Mac14 23ai % kubectl exec -it pods/${ORA_POD} -n oracle23ai -- sqlplus sys/${ORA_PASS}@${ORACLE_SID} as sysdba
Defaulted container "freedb-livelabs" out of: freedb-livelabs, init-prebuiltdb (init)

SQL*Plus: Release 23.0.0.0.0 - Production on Fri Jun 7 11:29:34 2024
Version 23.4.0.24.05

Copyright (c) 1982, 2024, Oracle. All rights reserved.

Connected to:
Oracle Database 23ai Free Release 23.0.0.0.0 - Develop, Learn, and Run for Free
Version 23.4.0.24.05

SQL> set linesize 200
SQL> select INSTANCE_NAME, HOST_NAME, VERSION_FULL, EDITION from v$instance;

INSTANCE_NAME   HOST_NAME                               VERSION_FULL   EDITION
-----
FREE            freedb-livelabs-bmu14                   23.4.0.24.05   FREE

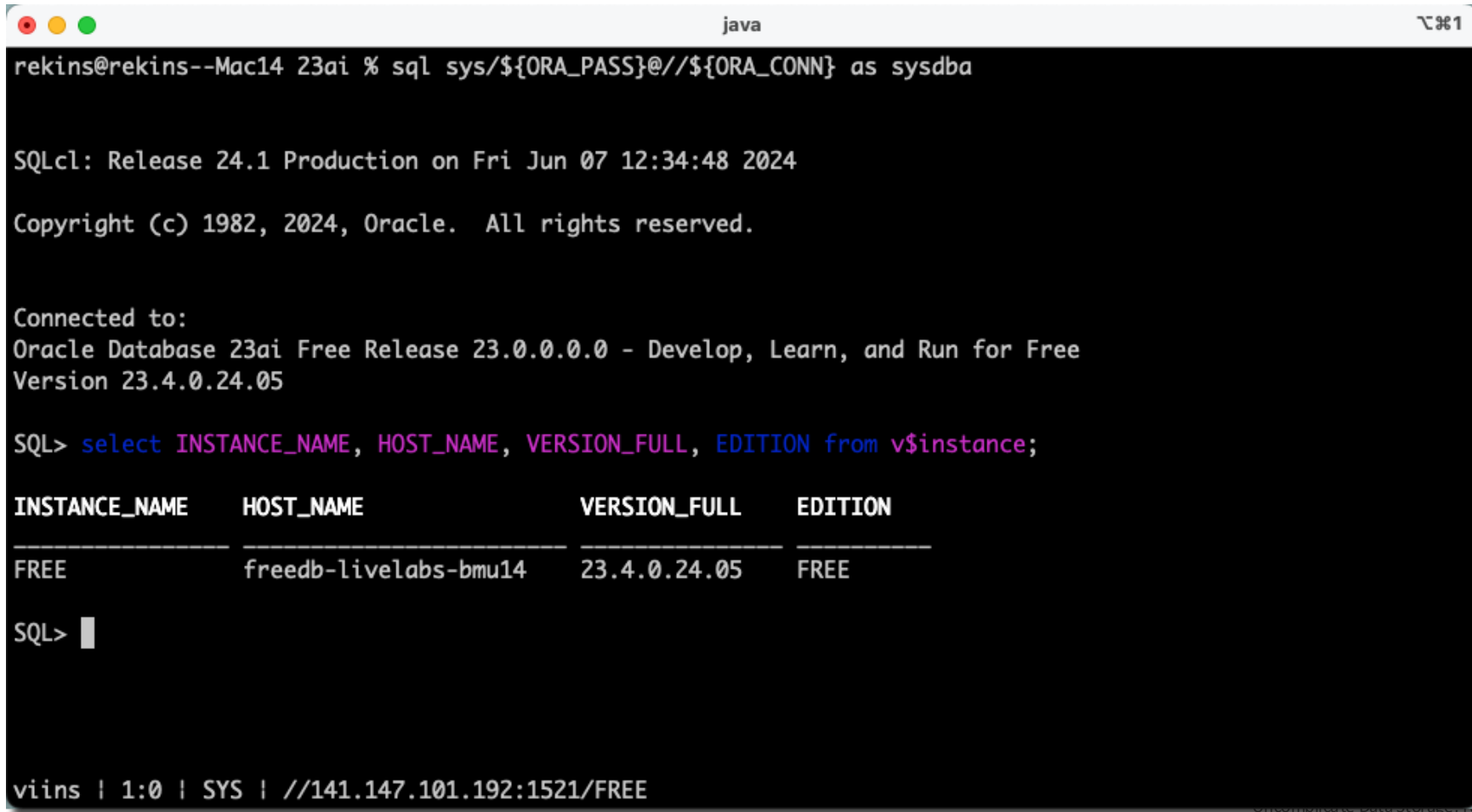
SQL> █
```



Connect to Oracle Database 23ai free

Connecting via OCI LoadBalancer

```
sql sys/${ORA_PASS}@//${ORA_CONN} as sysdba
```

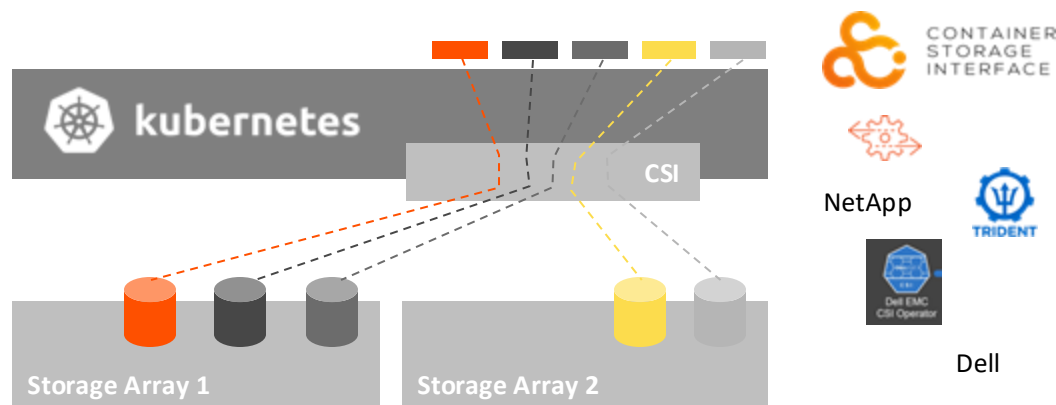
A terminal window titled 'java' with standard macOS window controls. The prompt is 'rekins@rekins--Mac14 23ai'. The user enters the command 'sql sys/\${ORA_PASS}@//\${ORA_CONN} as sysdba'. The terminal shows the SQL*Plus release information (Release 24.1 Production on Fri Jun 07 12:34:48 2024) and copyright notice. It then shows the connection status: 'Connected to: Oracle Database 23ai Free Release 23.0.0.0.0 - Develop, Learn, and Run for Free Version 23.4.0.24.05'. The user enters the query 'select INSTANCE_NAME, HOST_NAME, VERSION_FULL, EDITION from v\$instance;'. The results are displayed in a table with four columns: INSTANCE_NAME, HOST_NAME, VERSION_FULL, and EDITION. The single row of data shows 'FREE', 'freedb-livelabs-bmu14', '23.4.0.24.05', and 'FREE'. The prompt 'SQL>' is shown with a cursor. At the bottom, a status bar reads 'viins | 1:0 | SYS | //141.147.101.192:1521/FREE'.

Oracle 23ai Failover

Using Portworx Enterprise Cloud Native Software Defined Storage

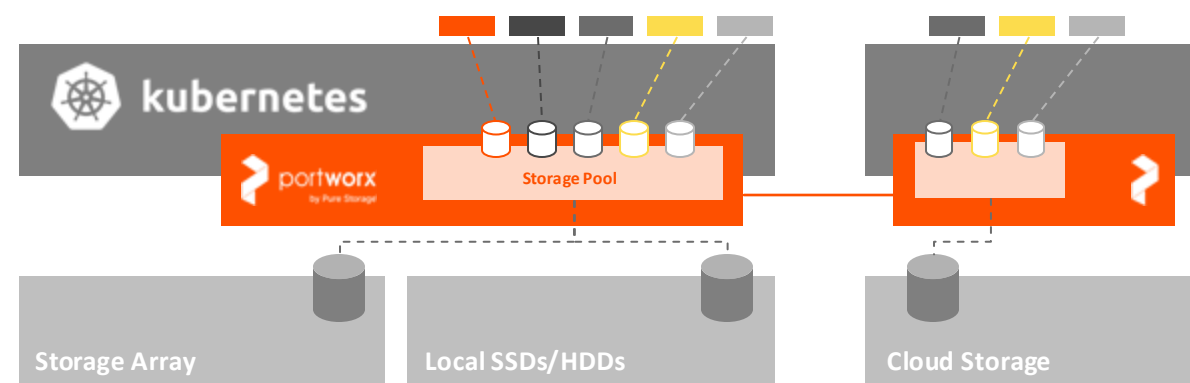
Container-Native vs. Connector Approach

"Connector" Approach



- 1:1 container <> volume mapping
- Container capabilities vary depending on the capabilities of the underlying array
- Storage arrays often are challenged by the object count (# volumes, connections) and change rate (10,000s changes/day)
- Innovation limited to the features in the open CSI specification
- Storage arrays managed independently from K8s

Container-Native Storage Approach

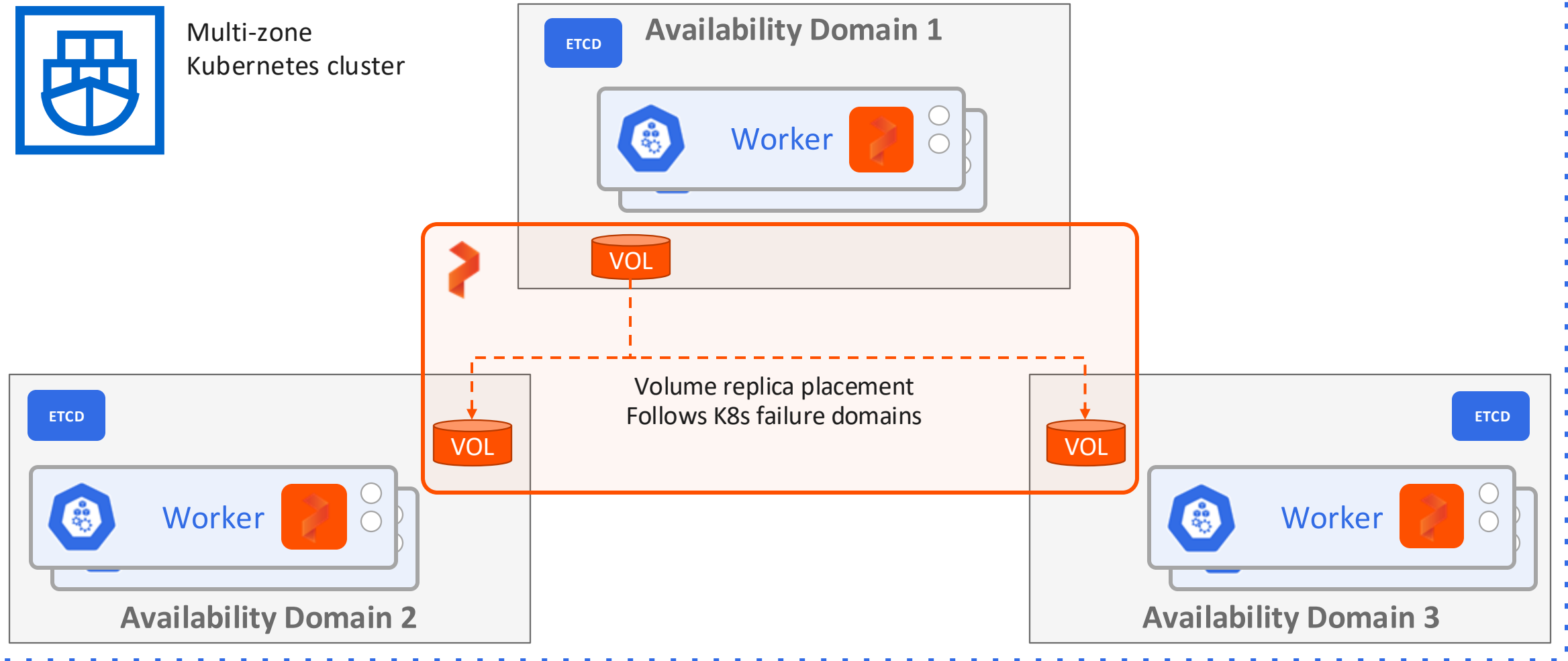


- Many (1,000s):1 container <> volume mapping
- Designed to scale to 10,000s operations/day
- Ensures consistent storage services across ANY infrastructure
- Storage services container-granular and application consistent
- Storage "encapsulated" in a container-native virtual volume that is free to move with the container
- Managed as a native part of K8s

Multi-zone high availability (HA)



Multi-zone
Kubernetes cluster



Portworx for Oracle Container Engine for Kubernetes (OKE)

Product Line [← Back for Product Catalog](#)

Generate Spec *required

Step 1: Select Your Platform

Portworx Version* **3.1**

Platform* **Oracle**

Portworx dynamically provisions and manages storage using underlying Oracle block volumes. To modify this, click the "Customize" button in Step 3.

Step 2: Select Kubernetes Distribution

Distribution Name* **Oracle Kubernetes Engine (OKE)**

Namespace* **portworx**

OCI Block Volume Performance Units (V



Basic

Kubernetes Version: 1.29.1

BuiltIn etcd

Storage



Select your environment *

On Premises

On Cloud

Select Cloud Platform *

Oracle

Configure storage devices

If you are running on Oracle Kubernetes Engine (OKE), it is recommended to use the Portworx ASG feature, where Portworx manages the entire lifecycle of Oracle block storage.

For ASG, Portworx will create disks using a spec.

Select type of disk *

Create Using a Spec

Consume Unused

Select number of VPU's *

10

Size (GB) *

150

Encryption

None

Encryption Key

Not Applicable.

Max storage nodes per availability zone (Optional)

Default IO Profile

Auto

Journal Device

None

Elastic Performance Level	Volume Performance Units (VPUs)	IOPS per GB	Max IOPS per Volume	Size for Max IOPS (GB)	KBPS per GB	Max MBPS per Volume
Lower Cost	0	2	3,000	1,500	240	480
Balanced	10	60	25,000	417	480	480
Higher Performance	20	75	50,000	667	600	680
Ultra High Performance	30	90	75,000	833	720	880
Ultra High Performance	40	105	100,000	952	840	1,080
Ultra High Performance	50	120	125,000	1,042	960	1,280
Ultra High Performance	60	135	150,000	1,111	1,080	1,480
Ultra High Performance	70	150	175,000	1,167	1,200	1,680
Ultra High Performance	80	165	200,000	1,212	1,320	1,880
Ultra High Performance	90	180	225,000	1,250	1,440	2,080
Ultra High Performance	100	195	250,000	1,282	1,560	2,280
Ultra High Performance	110	210	275,000	1,310	1,680	2,480
Ultra High Performance	120	225	300,000	1,333	1,800	2,680



Kubernetes Portworx Storage Class

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: px-csi-ora
provisioner: pxd.portworx.com
parameters:
  repl: "3"
  io_profile: "auto"
  priority_io: "high"
allowVolumeExpansion: true
```



SingleInstanceDatabase

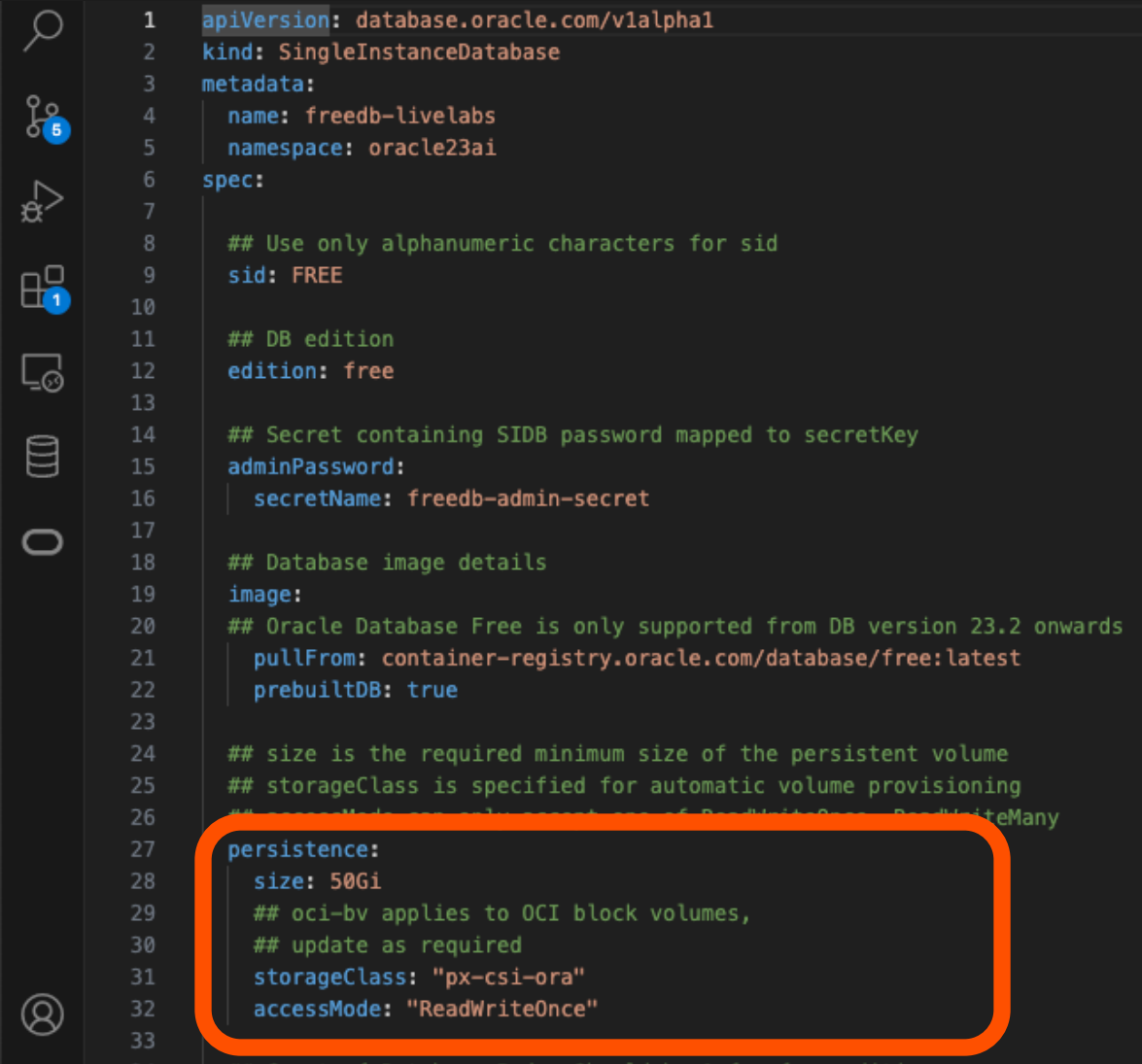
name

namespace

secretName

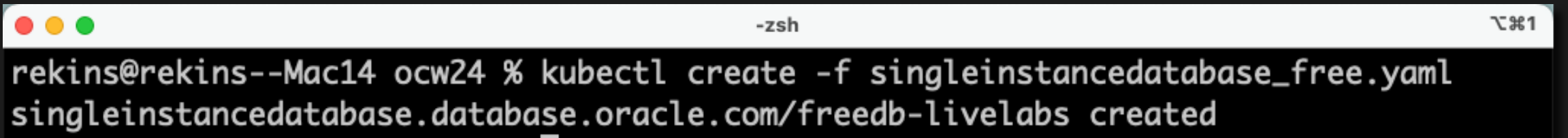
storageClass

loadBalancer



The image shows a code editor with a Kubernetes manifest file. The manifest defines a SingleInstanceDatabase resource. The 'persistence' section is highlighted with an orange box. The manifest includes fields for apiVersion, kind, metadata (name, namespace), and spec (sid, edition, adminPassword, image, and persistence). The persistence section specifies a size of 50Gi, storageClass 'px-csi-ora', and accessMode 'ReadWriteOnce'.

```
1 apiVersion: database.oracle.com/v1alpha1
2 kind: SingleInstanceDatabase
3 metadata:
4   name: freedb-livelabs
5   namespace: oracle23ai
6 spec:
7
8   ## Use only alphanumeric characters for sid
9   sid: FREE
10
11   ## DB edition
12   edition: free
13
14   ## Secret containing SIDB password mapped to secretKey
15   adminPassword:
16     secretName: freedb-admin-secret
17
18   ## Database image details
19   image:
20     ## Oracle Database Free is only supported from DB version 23.2 onwards
21     pullFrom: container-registry.oracle.com/database/free:latest
22     prebuiltDB: true
23
24   ## size is the required minimum size of the persistent volume
25   ## storageClass is specified for automatic volume provisioning
26   ## accessMode can only be one of ReadWriteOnce, ReadWriteMany
27   persistence:
28     size: 50Gi
29     ## oci-bv applies to OCI block volumes,
30     ## update as required
31     storageClass: "px-csi-ora"
32     accessMode: "ReadWriteOnce"
33
```

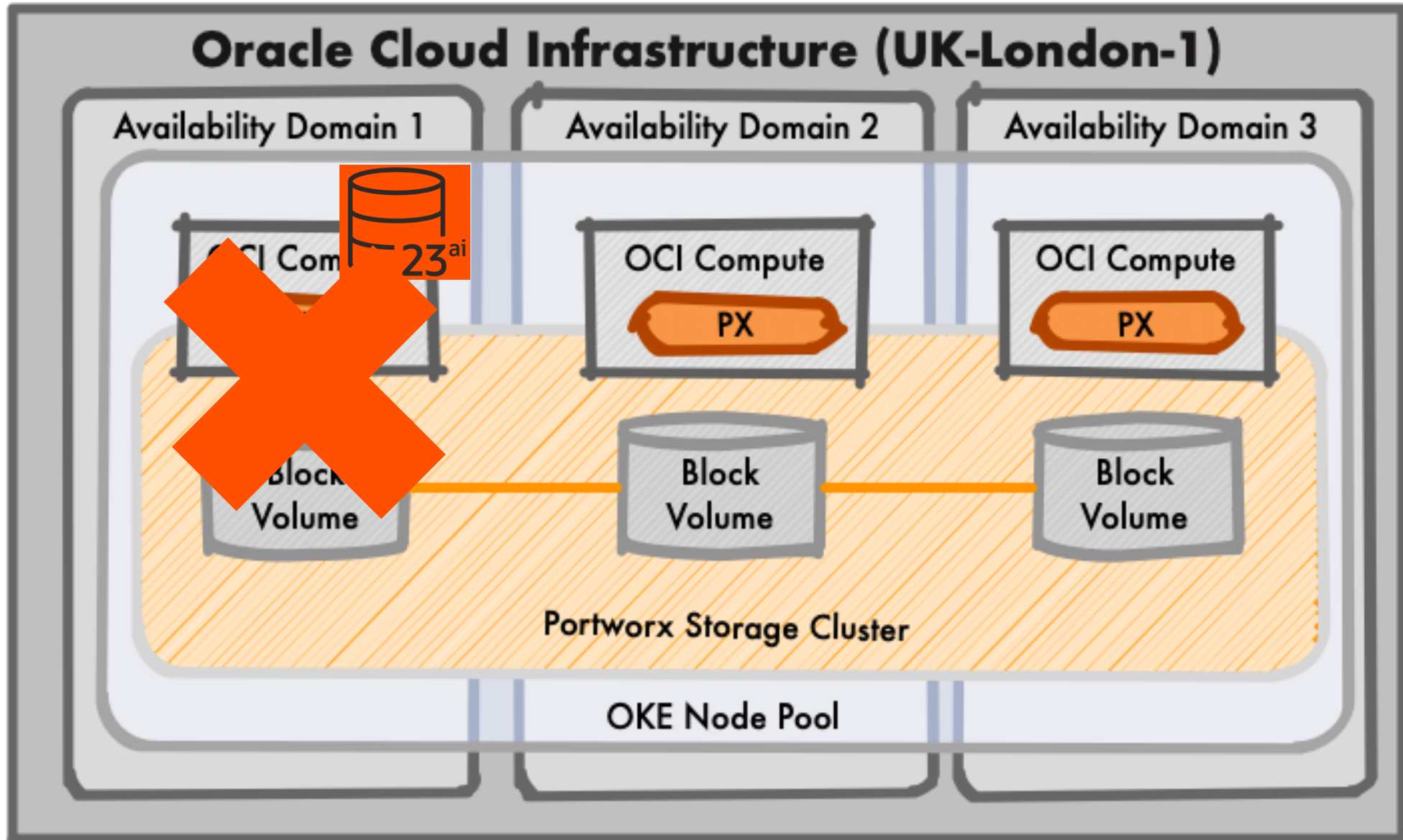


The image shows a terminal window with the command to create the SingleInstanceDatabase resource. The command is: `kubectl create -f singleinstancedatabase_free.yaml`. The output is: `singleinstancedatabase.database.oracle.com/freedb-livelabs created`.

```
-zsh
rekins@rekins--Mac14 ocw24 % kubectl create -f singleinstancedatabase_free.yaml
singleinstancedatabase.database.oracle.com/freedb-livelabs created
```



Oracle Database 23ai Planned / Managed Failover



Cordorn AD-1 & AD-2

```
rekins@rekins--Mac14 23ai % kubectl get nodes
NAME                STATUS    ROLES    AGE   VERSION
10.0.10.133         Ready    node     28d   v1.29.1
10.0.10.139         Ready    node     28d   v1.29.1
10.0.10.80          Ready    node     28d   v1.29.1
rekins@rekins--Mac14 23ai % kubectl cordon 10.0.10.133
node/10.0.10.133 cordoned
rekins@rekins--Mac14 23ai % kubectl cordon 10.0.10.139
node/10.0.10.139 cordoned
rekins@rekins--Mac14 23ai % kubectl get nodes
NAME                STATUS              ROLES    AGE   VERSION
10.0.10.133         Ready,SchedulingDisabled node     28d   v1.29.1
10.0.10.139         Ready,SchedulingDisabled node     28d   v1.29.1
10.0.10.80          Ready               node     28d   v1.29.1
rekins@rekins--Mac14 23ai %
```

Kill Oracle23ai Pod

```
rekins@rekins--Mac14 23ai % kubectl get pods -n oracle23ai
NAME                                READY   STATUS    RESTARTS   AGE
freedb-livelabs-tszq7              1/1     Running   0           15m
rekins@rekins--Mac14 23ai % kubectl get pods -n oracle23ai -o wide
NAME                                READY   STATUS    RESTARTS   AGE   IP              NODE             NOMINATED NODE   READINESS GATES
freedb-livelabs-tszq7              1/1     Running   0           15m   10.0.10.3       10.0.10.139     <none>           <none>
rekins@rekins--Mac14 23ai % kubectl delete pod/freedb-livelabs-tszq7 -n oracle23ai
pod "freedb-livelabs-tszq7" deleted
```

kubectl get pods -n oracle23ai -o custom-columns=NAME:metadata.name,IP:status.hostIP

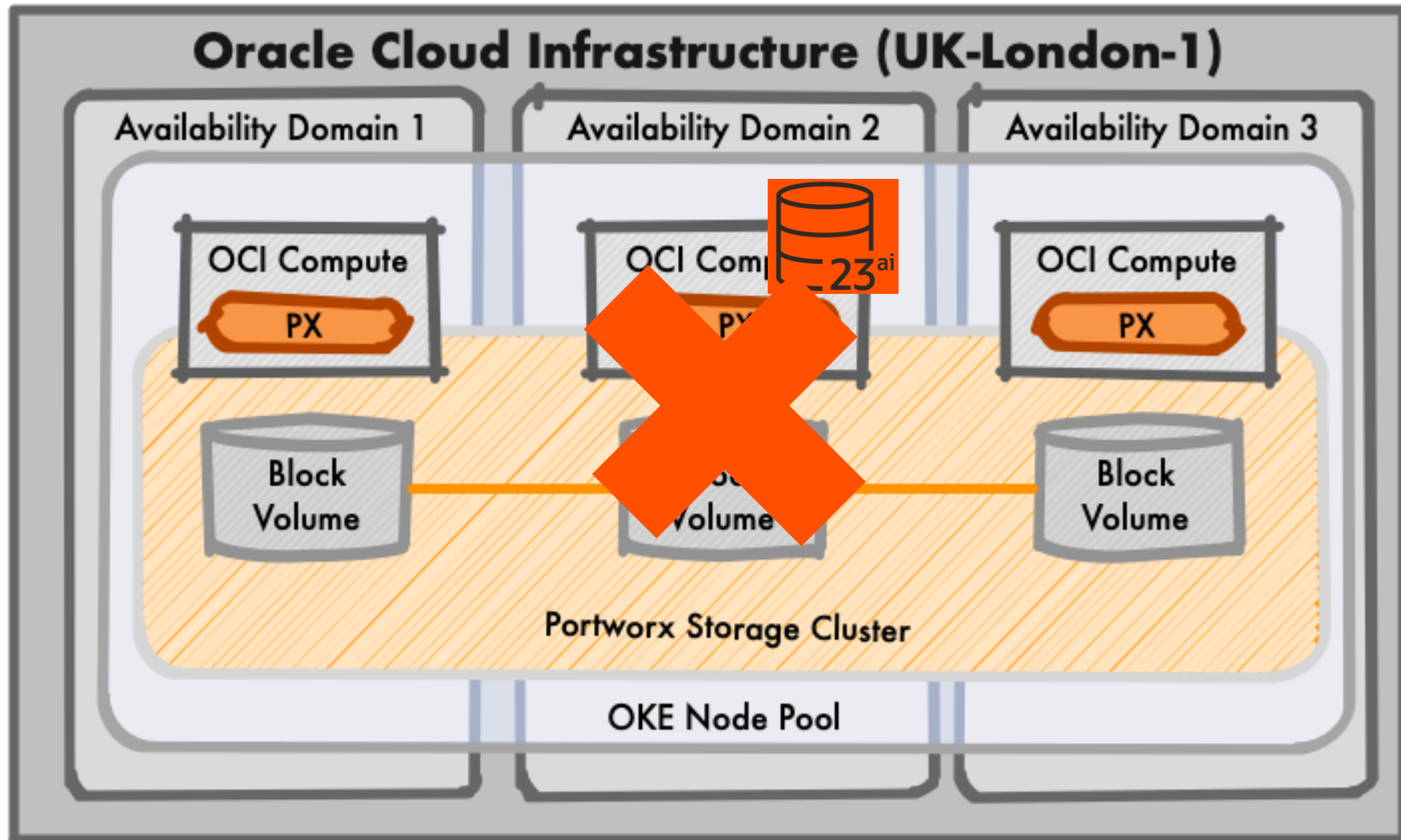
kubectl get singleinstancedatabase -n oracle23ai

```
rekins@rekins--Mac14 23ai % kubectl get pods -n oracle23ai -o custom-columns=NAME:metadata.name,IP:status.hostIP
NAME                                IP
freedb-livelabs-7tf9p              10.0.10.80
rekins@rekins--Mac14 23ai % kubectl get singleinstancedatabase -n oracle23ai
NAME                                EDITION   STATUS   ROLE     VERSION      CONNECT STR          TCPS CONNECT STR   OEM EXPRESS URL
freedb-livelabs                    Free      Healthy  PRIMARY  23.4.0.24.05  79.72.78.119:1521/FREE  Unavailable         Unavailable
rekins@rekins--Mac14 23ai %
```





Oracle Database 23ai Unplanned Failover



Oracle 23ai running on AD-2

kubectl get pods -n oracle23ai -o custom-columns=NAME:metadata.name,IP:status.hostIP

```
rekins@rekins--Mac14 ~ % kubectl get pods -n oracle23ai -o custom-columns=NAME:metadata.name,IP:status.hostIP
NAME                                IP
freedb-livelabs-jquet              10.0.10.133
rekins@rekins--Mac14 ~ %
```

kubectl get nodes -L topology.kubernetes.io/region,topology.kubernetes.io/zone

```
rekins@rekins--Mac14 ~ % kubectl get nodes -L topology.kubernetes.io/region,topology.kubernetes.io/zone
NAME                                STATUS    ROLES    AGE   VERSION   REGION           ZONE
10.0.10.133                        Ready     node     27d   v1.29.1   uk-london-1      UK-LONDON-1-AD-2
10.0.10.139                        Ready     node     27d   v1.29.1   uk-london-1      UK-LONDON-1-AD-1
10.0.10.80                         Ready     node     27d   v1.29.1   uk-london-1      UK-LONDON-1-AD-3
rekins@rekins--Mac14 ~ %
```

Stop AD-2 Compute Instance

Oracle Cloud

Search resources, services, documentation, and Marketplace

UK South (London)

Compute

Overview

Instances

Dedicated Virtual Machine Hosts

Instance Configurations

Instance Pools

Cluster Networks

Compute Clusters

Autoscaling Configurations

Capacity Reservations

Custom Images

Instances in PX-compartment compartment

An [instance](#) is a compute host. Choose between virtual machines (VMs) and bare metal instances. The image that you use to launch an instance determines its operating system and other software.

Create instance Actions

☐	Name	State	Public
☐	oke-cbpmdmfgbea-nwackrpenug-ssha4m7gkxa-3	Running	-
☐	oke-cbpmdmfgbea-nwackrpenug-ssha4m7gkxa-4	Running	-
☐	oke-cbpmdmfgbea-nwackrpenug-ssha4m7gkxa-5	Running	-

0 selected

Create instance Actions

☐ Name

☐ [oke-cbpmdmfgbea-nwackrpenug-ssha4m7gkxa-3](#)

☒ [oke-cbpmdmfgbea-nwackrpenug-ssha4m7gkxa-4](#)

☐ [oke-cbpmdmfgbea-nwackrpenug-ssha4m7gkxa-5](#)

1 selected

Reboot

Start

Stop

Terminate

Table settings

OKE Compute Instance Stopped

ORACLE Cloud

Search resources, services, documentation, and Marketplace

UK South (London)

Compute

Overview

Instances

Dedicated Virtual Machine Hosts

Instance Configurations

Instance Pools

Cluster Networks

Compute Clusters

Autoscaling Configurations

Capacity Reservations

Custom Images

Instances in PX-compartment *compartment*

An [instance](#) is a compute host. Choose between virtual machines (VMs) and bare metal instances. The image that you use to launch an instance determines its operating system and other software.

Create instance

Actions

☐	Name	State	Public IP	Private IP	Shape	OCPU count	Memory (GB)	Availability domain
☐	oke-cbpmdmfgbea-nwackrpenug-ssha4m7gkxa-3	Running	-	10.0.10.139	VM.Standard.E3.Flex	16	64	AD-1
☐	oke-cbpmdmfgbea-nwackrpenug-ssha4m7gkxa-4	Stopped	-	10.0.10.133	VM.Standard.E3.Flex	16	64	AD-2
☐	oke-cbpmdmfgbea-nwackrpenug-ssha4m7gkxa-5	Running	-	10.0.10.80	VM.Standard.E3.Flex	16	64	AD-3

0 selected

Showing 3 items < 1 of 1 >

-zsh

```
rekins@rekins--Mac14 ~ % kubectl get nodes -L topology.kubernetes.io/region,topology.kubernetes.io/zone
```

NAME	STATUS	ROLES	AGE	VERSION	REGION	ZONE
10.0.10.133	NotReady	node	28d	v1.29.1	uk-london-1	UK-LONDON-1-AD-2
10.0.10.139	Ready	node	28d	v1.29.1	uk-london-1	UK-LONDON-1-AD-1
10.0.10.80	Ready	node	28d	v1.29.1	uk-london-1	UK-LONDON-1-AD-3

```
rekins@rekins--Mac14 ~ %
```



Oracle 23ai started on Availability Domain 1

kubectl get pods -n oracle23ai -o custom-columns=NAME:metadata.name,IP:status.hostIP

```
rekins@rekins--Mac14 23ai % kubectl get pods -n oracle23ai -o custom-columns=NAME:metadata.name,IP:status.hostIP
```

NAME	IP
freedb-livelabs-tszo7	10.0.10.139

```
rekins@rekins--Mac14 23ai %
```

kubectl get nodes -L topology.kubernetes.io/region,topology.kubernetes.io/zone

```
rekins@rekins--Mac14 ~ % kubectl get nodes -L topology.kubernetes.io/region,topology.kubernetes.io/zone
```

NAME	STATUS	ROLES	AGE	VERSION	REGION	ZONE
10.0.10.133	Ready	node	27d	v1.29.1	uk-london-1	UK-LONDON-1-AD-2
10.0.10.139	Ready	node	27d	v1.29.1	uk-london-1	UK-LONDON-1-AD-1
10.0.10.80	Ready	node	27d	v1.29.1	uk-london-1	UK-LONDON-1-AD-3

```
rekins@rekins--Mac14 ~ %
```

Kubernetes has started a new Oracle 23ai Pod

Pod relocated from AD-2 -> AD-1

`sql sys/${ORA_PASS}@//${ORA_CONN} as sysdba`

```
rekins@rekins--Mac14 23ai % sql sys/${ORA_PASS}@//${ORA_CONN} as sysdba

SQLcl: Release 24.1 Production on Tue Jul 02 16:51:52 2024

Copyright (c) 1982, 2024, Oracle. All rights reserved.

Connected to:
Oracle Database 23ai Free Release 23.0.0.0.0 - Develop, Learn, and Run for Free
Version 23.4.0.24.05

SQL> select INSTANCE_NAME, HOST_NAME, VERSION_FULL, EDITION from v$instance;

INSTANCE_NAME  HOST_NAME                VERSION_FULL  EDITION
-----
FREE           freedb-livelabs-tszq7    23.4.0.24.05  FREE

SQL>
```



```
rekins@rekins--Mac14 23ai % kubectl get singleinstancedatabase -n oracle23ai
```

Deploy containerized Oracle Database 23ai Free on OKE using Portworx Enterprise storage

[Share](#)[Start](#)[Copy link](#) 40 minutes

Outline

- Create an Oracle Container Engine for Kubernetes (OKE) Cluster using wizard
- Deploy Portworx Enterprise (PX) Cloud Native Storage
- Enable the Oracle Database Operator for Kubernetes (OraOperator)
- Provision containerized Oracle Database 23ai Free

Prerequisites

- OCI account

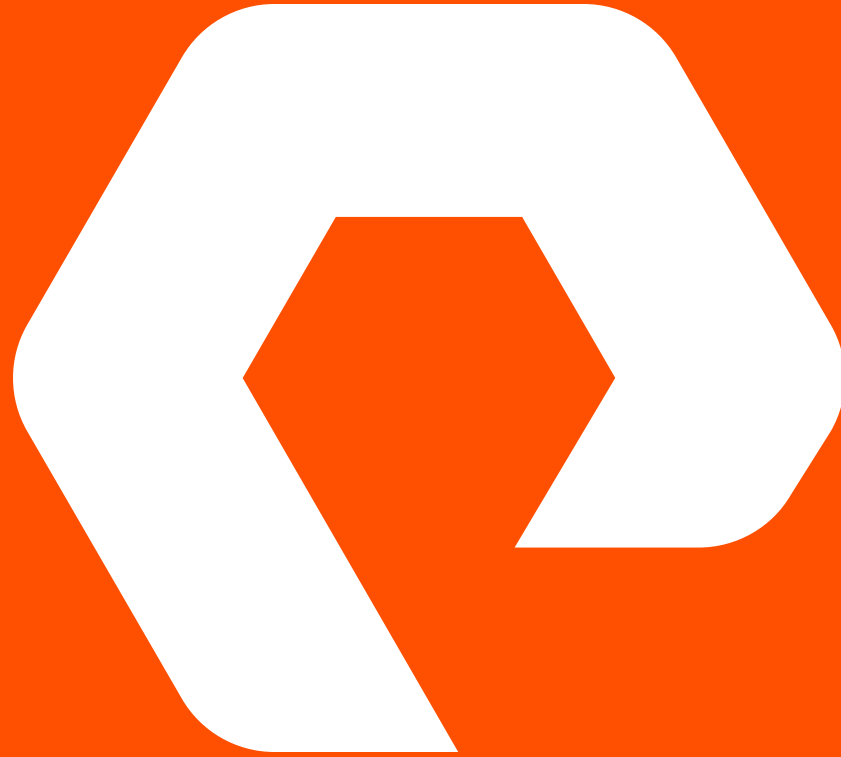


About This Workshop

Learn how to create an Oracle Container Engine for Kubernetes (OKE) cluster, install Portworx Enterprise (PX) cloud native storage, and provision a containerized Oracle Database 23ai Free using the Oracle Database Operator for Kubernetes.



Any Questions...



Uncomplicate Data Storage, Forever