MAYADATA

Solution Document

# NuoDB and OpenEBS

Certified with OpenEBS version 0.8.1

This solution document describes the design and best practices when using OpenEBS as persistent storage for NuoDB when running in container native deployment environments like Red Hat Openshift or Open source Kubernetes. Topics include: how OpenEBS and NuoDB are designed together for production usage; how to introduce chaos insertion testing to validate and harden production deployments and increase operational resilience; and how to monitor production deployments to optimize and plan the capacity and performance needs for short and long term needs.

# CONTENTS

# INTRODUCTION

NuoDB is an container-native distributed SQL database designed with distributed application deployment challenges in mind. It is a relational SQL database  that provides all the properties of ACID-compliant transactions and standard ANSI SQL language support. It’s also designed from the start as a distributed system that scales the way a cloud service has to scale, providing high availability and resiliency with no single points of failure. Different from traditional shared-disk or shared-nothing architectures, NuoDB’s presents a new kind of peer-to-peer, on-demand independence that yields continuous availability, low-latency, and a deployment model that is easy to manage.

Kubernetes has become the dominant platform for running container workloads and is expanding to run virtual machine based workloads as well.  As of early 2019 every major cloud provider offers at least one version of Kubernetes based managed services and companies including RedHat, Cisco, Nutanix, VMware and others offer on premise open source based distributions. . The ACID compliant Elastic SQL capabilities of NuoDB provides a required middle tier to distributed data architecture models for many use cases that require transaction consistency and cross node and zone resiliency.

NuoDB and OpenEBS on Kubernetes

OpenEBS applies common technologies such as Kubernetes and containers as well as preferred architectures such as microservices to deliver run anywhere storage services including block and file storage and local disk management as well as optional data protection and replication features.  . OpenEBS utilizes Container Attached Storage or CAS architectural principles, where the storage software itself runs inside containers that are themselves orchestrated by Kubernetes, making OpenEBS Kubernetes native and cloud native. Some advantages resulting from this architecture include:

**Immediate provisioning**
As quick as less than one minute thanks in part to integration with Helm Charts.

**Per workload and per team control**
Each workload and team has its own OpenEBS with their own storage policies. This approach is consistent with DevOps governance and culture.

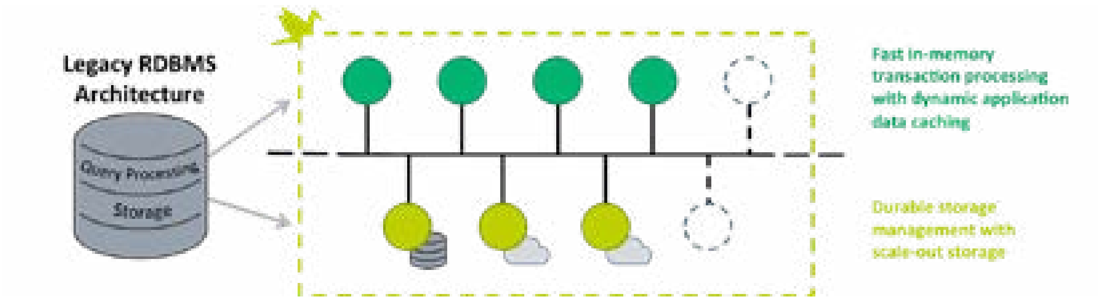
**Kubernetes orchestrated**
Benefits of a containerized and Kubernetes orchestrated architecture. These include simpler upgrades, higher velocity of development, independent scaling, and cloud independence.

**Enterprise storage per workload**
OpenEBS includes enterprise class storage features such as snapshots / cloning, replication for data recovery or high availability such as cross availability zone operations and cross cloud replication and migration.

# NUODB ARCHITECTURE

NewSQL systems are designed to operate in a distributed cluster of shared-nothing nodes, in which each node owns a subset of the data.

Although it appears as a single, logical SQL database to the application, its architecture consists of two independent processing layers that retain strict transactional consistency. It can even be deployed across multiple availability zones (even on different clouds!) and is optimized for in-memory speeds, continuous availability, and adaptive scale-out that adjusts to application needs.


Legacy RDBMS Architecture
Query Processing
Storage
Fast in-memory transaction processing with dynamic application data caching
Durable storage management with scale-out storage

**Transaction Engines (TEs):** The TE layer is used for (ACID) SQL and caching, made up of in-memory process nodes that coordinate with each other and the SM layer.

**Storage Managers (SMs):** The SM layer is used for storage and consists of process nodes that have both in-memory and on-disk storage components. SMs provide on-disk data durability guarantees, and multiple SMs can be used to increase data redundancy.

Both TE and SM node types can be easily added or removed to align with need.

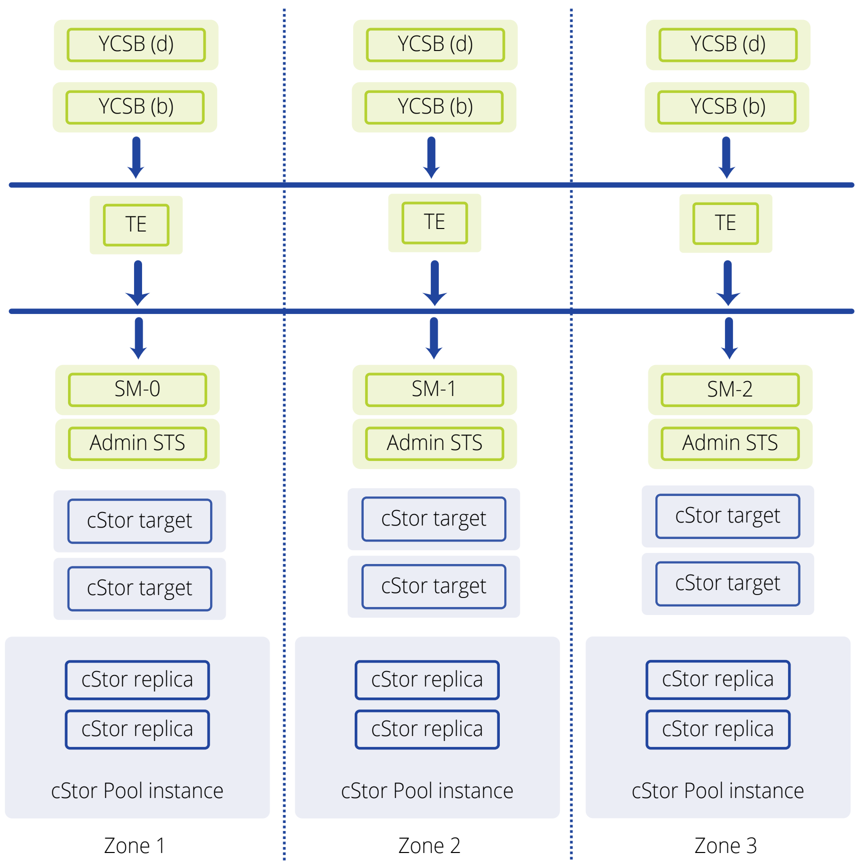
## WHY USE OPENEBS FOR NUODB ?

Running applications on traditional virtualized systems or baremetal is different than running them on Kubernetes. NuoDB is a horizontally scalable database with needs of scale up in performance on the fly from the underlying storage system. OpenEBS is a perfect choice to run NuoDB on Kubernetes both on-premise and cloud because of the following benefits.

- Easy and straightforward installation to deploy NuoDB with OpenEBS
- Local disks or cloud disks are managed seamless and in the same way by OpenEBS
- Persistent volumes (PVs) can be made available to NuoDB from a shared pool of disks. You can run multiple stateful databases out of the same physical disks present on the Kubernetes worker nodes, simplifying operations, increasing density, and decreasing costs
- Large size PVs can be provisioned for NuoDB. OpenEBS supports volumes upto petabytes in size
- Start with small storage and add disks as needed on the fly. Sometimes NuoDB instances are scaled up because of capacity on the nodes. With OpenEBS persistent volumes, capacity can be thin provisioned and disks can be added to OpenEBS on the fly without disruption of service
- NuoDB backups can be taken at the storage level and stored in S3. Using these backup/restore capabilities of OpenEBS, NuoDB instances can be moved across clouds or Kubernetes clusters seamlessly. There is no vendor lock-in problem when OpenEBS is used as underlying storage.

# DESIGN CONSIDERATIONS

A typical production deployment of NuoDB using OpenEBS is shown below. In this section we discuss the resources required initially and also for expansion. YCSB is the sample OLTP SQL application with (b) update workload and (d) insert workload with 90% read to 10% write workload mix



When it comes to the planning of required Kubernetes resources, both OpenEBS and NuoDB have to be considered together. There are two stateful components in NuoDB that consume OpenEBS storage. Storage Manager and admin controller. It is recommended that capacity and performance requirements from Storage point of view is designed first and then consider the design for NuoDB application. OpenEBS persistent volumes of many applications can reside in common cStorPools, and for this reason consideration must be given to allocate enough CPU and memory and disk capacity to OpenEBS cStorPools.

## RESOURCE SIZING

Below table gives details of resources estimation for starting phase.

| | Node 1 | | Node 2 | | Node 3 | |
|---|---|---|---|---|---|---|
| | vCPU | Memory (GB) | CPU | Memory (GB) | CPU | Memory (GB) |
| OpenEBS cStorPool | 4 | 8-12 | 4 | 8-12 | 4 | 8-12 |
| OpenEBS cStor target | 1 | 1 | 1 | 1 | 1 | 1 |
| NuoDB SM | 4 | 4 | 4 | 4 | 4 | 4 |
| NuoDB TE | 4 | 4 | 4 | 4 | 4 | 4 |
| NuoDB ADM | 1 | 1 | 1 | 1 | 1 | 1 |
| YCSB or Other apps | 2 | 4 | 2 | 4 | 2 | 4 |
| Total | 16 | 22-26 | 16 | 22-26 | 16 | 22-26 |

Based on the above table, it is recommended to choose nodes with minimum 16 vCPUs and 32GB RAM. Application consuming NuoDB will have its own resource requirements that need to be added on top this recommendation.

It is advised to run a minimum of **3 storage managers** for NuoDB to obtain high availability of NuoDB services even when one of the Kubernetes nodes is under maintenance.

## SCALING OF NUODB INSTANCES

NuoDB storage managers or admin components can be scaled by increasing the replica count and setting the NodeSelectors appropriately. Before scaling up the storage manager, OpenEBS cStorPool has to be made available on the target node such that PVC scheduling becomes successful.

## SCALING OF OPENEBS CAPACITY

OpenEBS supports thin provisioning. Capacity can be scaled up on demand within a cStorPool on a given node. It is common to host data of multiple applications in a single cStorPool where the initial and growth capacity requirements of all the applications are not known at the beginning. For NuoDB capacity requirements, start the OpenEBS volume sizes in the range of 100GB-500GB and increase the volume size on the fly when needed.

## NODE SELECTORS

For OpenEBS volumes and pools, NodeSelectors are automatically configured by OpenEBS provisioner. It is recommended to configure the **TargetNodeSelector** policy such that the OpenEBS targets are on the same node as that of Storage Managers or admin components.

## HIGH AVAILABILITY

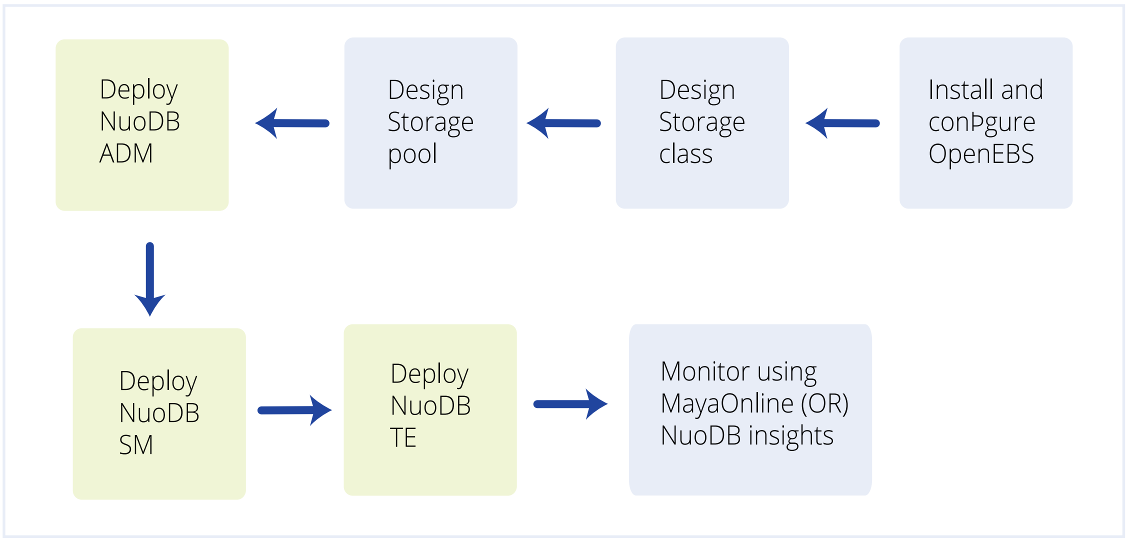
In this solution, NuoDB and OpenEBS are designed for redundancy of all components or modules involved. Following table illustrates the redundancy decision that have to be considered in the design phase.

| Components or module | Redundancy |
|---|---|
| NuoDB storage manager | Three replicas of Storage Manager are deployed. Even if one is lost, the system is still in protected state. |
| NuoDB Admin component | Three replicas of Admin component are deployed. Even if one is lost, the system is still in protected state. |
| NuoDB transaction engine | Multiple replicas of transaction en-ginge are configured to adaptively meet the requierments of the SQL application. |
| OpenEBS presistent volume | When OpenEBS volumes are de-ployed in a single replica mode, the application (SM or admin component) will manage the redundancy. In such cases losing storage volume tempo-rarily will not cause application down-time. |
| Physical disk where data is stored | Within a cStorPool, the disks can be configured in mirror or RAIDZ mode for redundancy in case of a disk fail-ure. Replacement of a failed disk with a hotspare and associated resilvering of data will ensure that redundancy of data within a pool is maintained even when disks fail. |

# DEPLOYING NUODB

Following instructions are provided for installing NuoDB 3.4 enterprise edition on OpenShift 3.9 or above with OpenEBS 0.8.1 version.

For NuoDB on OpenShift, you need to disable Disable Transparent Huge Pages on the OpenShit nodes and also use NuoDB enterprise edition. For easy convenience, OpenEBS litmus books can be used to deploy various components of NuoDB. The flowchart below shows various tasks you need to consider in the deployment process.



Below are the step by step instructions, for more detailed and latest documentation on OpenEBS, refer to https://docs.openebs.io/docs/next/nuodb.html

If you are using OpenEBS for some other stateful application, and want to use the same storagepool for NuoDB, you can skip the first two steps and directly go to step 3 (StorageClass creation)

Step1: Install OpenEBS
Choose disks on each node and create storage-pool-config.yaml and create cStor

Step2: Pool called cstor-pool
https://docs.openebs.io/docs/next/configurepools.html#manual-mode

Step3: Create “nuodb-cStor” StorageClass from the above cStorPool
https://docs.openebs.io/docs/next/configuresc.html#creating-a-new-class

Step4: Deploy NuoDB

Deploy NuoDB admin and storage manager as statefulsets with 3 replicas and transaction engine as deployment with 3 replicas. Note that only NuoDB admin and storage manager require OpenEBS storage and transaction engine is a stateless application.

For more details, refer to https://docs.openebs.io/docs/next/nuodb.html

**Note:** The litmus books for NuoDB at https://github.com/openebs/litmus/tree/master/apps/nuodb/deployers have a reference to NuoDB community edition. For deploying three Storage Manager as three replicas, you need enterprise edition. Hence consider replacing the image “image: nuodb/nuodb-ce:latest” with “image: nuodb/nuodb-ee:latest”

# RUNNING TEST LOAD

YCSB is a tool that you can use to easily generate SQL workload to load test and verify if the deployment is working as designed. There is an easy-to-use OpenEBS litmus book that you can use to run YCSB.

Download the below file from openebs litmus github repo and change the number of YCSB replicas to 3 and run the YAML file using kubectl.

### Download the YCSB litmus deployer

```
wget https://raw.githubusercontent.com/openebs/litmus/master/apps/nuodb/
deployers/nuodb-ycsb.yaml
```

### Change the replicas to 3

```
spec:
  replicas: 3
  selector:
    app: ycsb-load
```

### Change the load type to either “b” (95%R 5%W) or “d” (100% W) and NO_OF_PROCESSES to 10 or a lower number

```
    - { name: YCSB_WORKLOAD,       value: "b" }
     - { name: LB_POLICY,           value: "" }
     - { name: NO_OF_PROCESSES,     value: "2" }
```

### Run YCSB

```
Kubectl apply -f <nuodb-ycsb.yaml>
```

# MONITORING NUODB

## Monitor NuoDB metrics through Insights dashboard

To view the Nuodb load metrics, use nuodb-insights

If you are using open source Kubernetes, Insights application needs to be started manually. Follow the below instructions to start nuodb-insights and get access to NuoDB insights dashboard.

Exec into admin pod and run

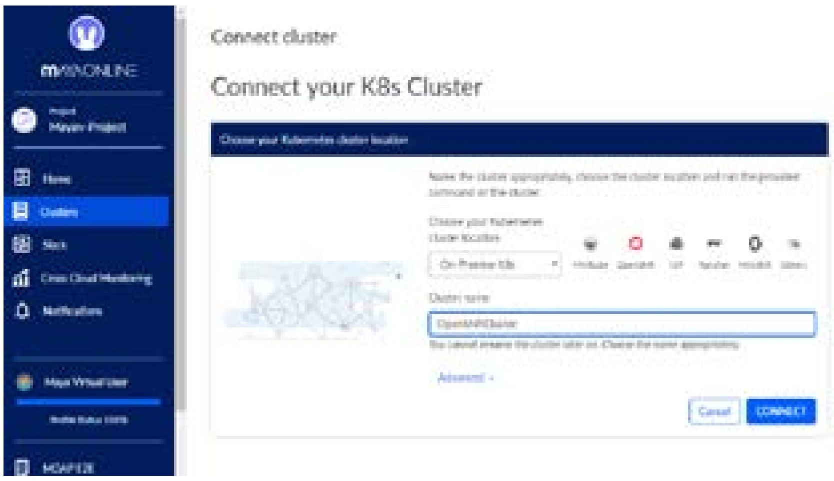
-
  ```
  > nuoca register insights --insights-url https://insights.nuodb.com/api/1 --enable-insights
  ```

-
  ```
  > nuoca start nuoca --insights &
  ```

# MONITOR OPENEBS VOLUME METRICS THROUGH MAYAONLINE

Connecting the Kubernetes cluster to MayaOnline provides good visibility of storage resources. MayaOnline has various support options for enterprise customers.

1. Login to mayaonline.io.
2. Go to the "Clusters" and click on "Connect a new cluster".
3. Select "On-Premises K8s" and click on the OpenShift icon.
4. Give your cluster a name.
5. Click on "Connect" button.
6. Copy the command and run it on your OpenShift cluster.



An example screenshot of OpenEBS volumes traffic corresponding to NuoDB storage manager statefulset is shown below

# CHAOS INSERTION TESTING

The Chaos Engineering is performed on a test rig with NuoDB 3.4 EE involving components 3 Admin (STS), 3 StorageManager (STS) and 3 Transaction Engine (Deployment) configuration on OpenShift 3.10 EE with active 3 YCSB(Replica Controller) B workload running.

OpenEBS Litmus books are provided not only for NuoDB deployment but for inserting various types of chaos as well.

**Prerequisites for running Litmus books:**

https://github.com/openebs/litmus#pre-requisites-for-running-a-specific-test

**Litmus books for NuoDB application components(Admin,SM and TE) chaos tests using application specific labels:**

https://github.com/openebs/litmus/blob/master/apps/nuodb/chaos/run_litmus_test.yml

Litmus books for performing OpenEBS storage Chaos:

**Litmus books for OpenEBS target chaos tests:**

https://github.com/openebs/litmus/blob/master/apps/percona/chaos/openebs_target_failure/run_litmus_test.yml

**Litmus books for OpenEBS replica chaos tests:**

https://github.com/openebs/litmus/blob/master/apps/percona/chaos/openebs_pool_failure/run_litmus_test.yml

## SUMMARY

NuoDB is an ACID compliant adaptive scaleout distributed SQL database that is well supported on Kubernetes where it needs an easy-to-deploy and easy-to-manage native persistent storage.

OpenEBS is used successfully to manage the persistent storage needs of NuoDB on Kubernetes on any cloud or hybrid cloud platform. On a moderately scaled setup like the one mentioned in this solution document, users can achieve upwards of 50,000 SQL NuoDB transactions per second and terabytes of highly available storage. OpenEBS is managed in a native Kubernetes way with no special skills needed from a Kubernetes administrator. The scheduling, capacity and performance needs of OpenEBS volumes for NuoDB are highly tunable through StorageClass policies.

MAYADATA