nirmata

Kubernetes: Persistent Storage with Rook

Contents

Revision 1

Rook Introduction 2

- Prerequisites 2
- Install Linux packages 3
- Configure and deploy operator.yaml 3
- Configure and deploy cluster.yaml 6
- Setup storage-class and replica pool - 14
- Create a new environment and run your application 17
- Verify data persistence across replication with host failure scenario - 21
- Disable host running the current pod 24
- Verify data in the new Pod 25

Revision

Version	Date	Author	Changes
0.1.0	08/24/2018	Anubhav Sharma	Initial Draft

Rook Introduction

Rook is a CNCF open-source project built to deliver storage solution for Kubernetes leveraging battle-tested open-source storage technologies including Ceph, which has years of production deployments and runs some of the worlds largest clusters. Rook is available under Apache 2.0 license.

As containers are ephemeral by nature, without a persistent storage solution, you can lose your data as container dies. This problem is solved through persistent storage solutions that can be accessed by Kubernetes applications and deliver scale, performance and availability required for large data stores for cloud native environment.

Rook is a great fit to solve the persistent storage problem for containers and this paper, we demonstrate an easy integration of rook with Nirmata platform and provisioning of highly available replicated storage.

With Rook you can either build dedicated storage clusters or hyper-converged clusters where your apps run alongside storage. Rook integrates Ceph with multiple storage presentations including object storage (compatible with S3 and swift), block storage, and POSIX-compliant shared file system.

Rook efficiently distributes and replicates your data across your cluster to minimize the risk of data loss. With snapshots, cloning and versioning, no more losing sleep over your data.

Prerequisites

1. Nirmata Cluster with Kubernetes 1.7+, minimum 3 nodes in a cluster (We are using Kubernetes version 1.9.4 with 4 nodes).
2. Flex Volume Configuration: Enabled by default with Nirmata (Directory: "/opt/nirmata/volume-plugins")
3. kubectl: 1.9+ (for setting rook cluster)
4. dataDirHostPath Storage: Path on VM node(host) to store config and data for rook services (enough to meet container persistent storage requirements); default is /var/lib/rook
5. Linux packages: rbd-fuse, ceph-fs-common

Install Linux packages

On Ubuntu, run these 2 commands as root (or sudo) on each node VM

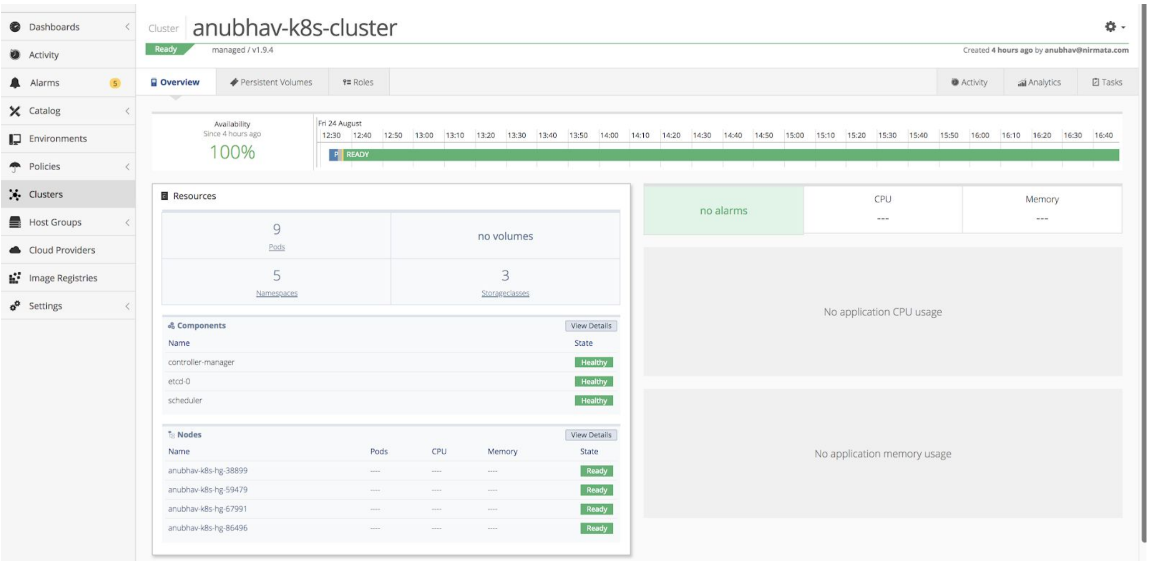
- `apt-get install rbd-fuse`
- `apt-get install ceph-fs-common`

```
anubhav@anubhav-k8s-hg-86496:~$ sudo apt-get install rbd-fuse
anubhav@anubhav-k8s-hg-86496:~$ sudo apt-get install ceph-fs-common
```

Setup your Kubernetes Cluster through Nirmata

1. Setup your Cloud-provider.
2. Setup your container Hostgroup.
3. Setup your Kubernetes Cluster.

Your Kubernetes cluster will look as below -



Cluster anubhav-k8s-cluster

Ready managed / v1.9.4

Created **4 hours ago** by **anubhav@nirmata.com**

Availability
Since 4 hours ago
100%

Resources

9 Pods

no volumes

5 Namespaces

3 Storageclasses

no alarms

CPU

Memory

No application CPU usage

No application memory usage

Components

Name	State
controller-manager	Healthy
etcd-0	Healthy
scheduler	Healthy

Nodes

Name	Pods	CPU	Memory	State
anubhav-k8s-hg-38899	----	----	----	Ready
anubhav-k8s-hg-59479	----	----	----	Ready
anubhav-k8s-hg-67991	----	----	----	Ready
anubhav-k8s-hg-86496	----	----	----	Ready

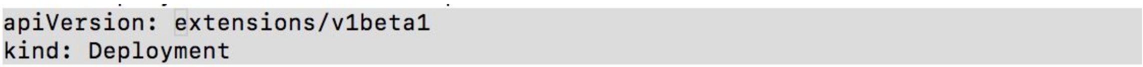
Configure and deploy operator.yaml

To work with Nirmata version 2.1.0, you need make couple of configuration changes to operator.yaml file.

Full yaml file is available here - operator.yaml

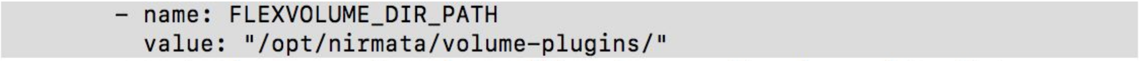
Modify following parameters in operator.yaml -

Using extensions instead of apps -



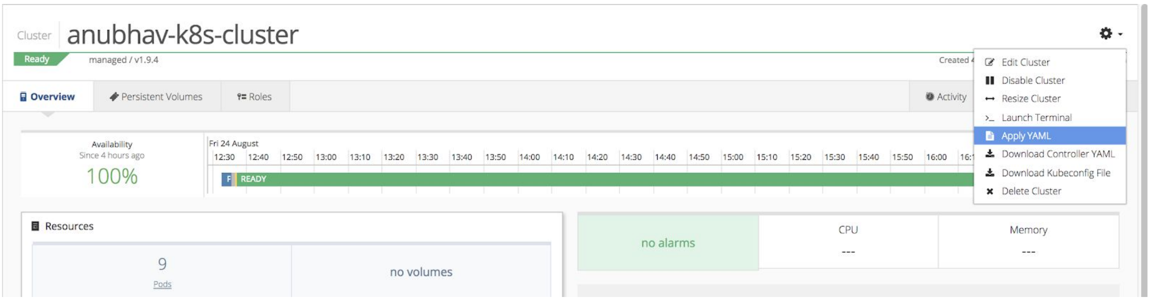
```
apiVersion: extensions/v1beta1
kind: Deployment
```

Configure flex-volume path for ceph volume-plugins.

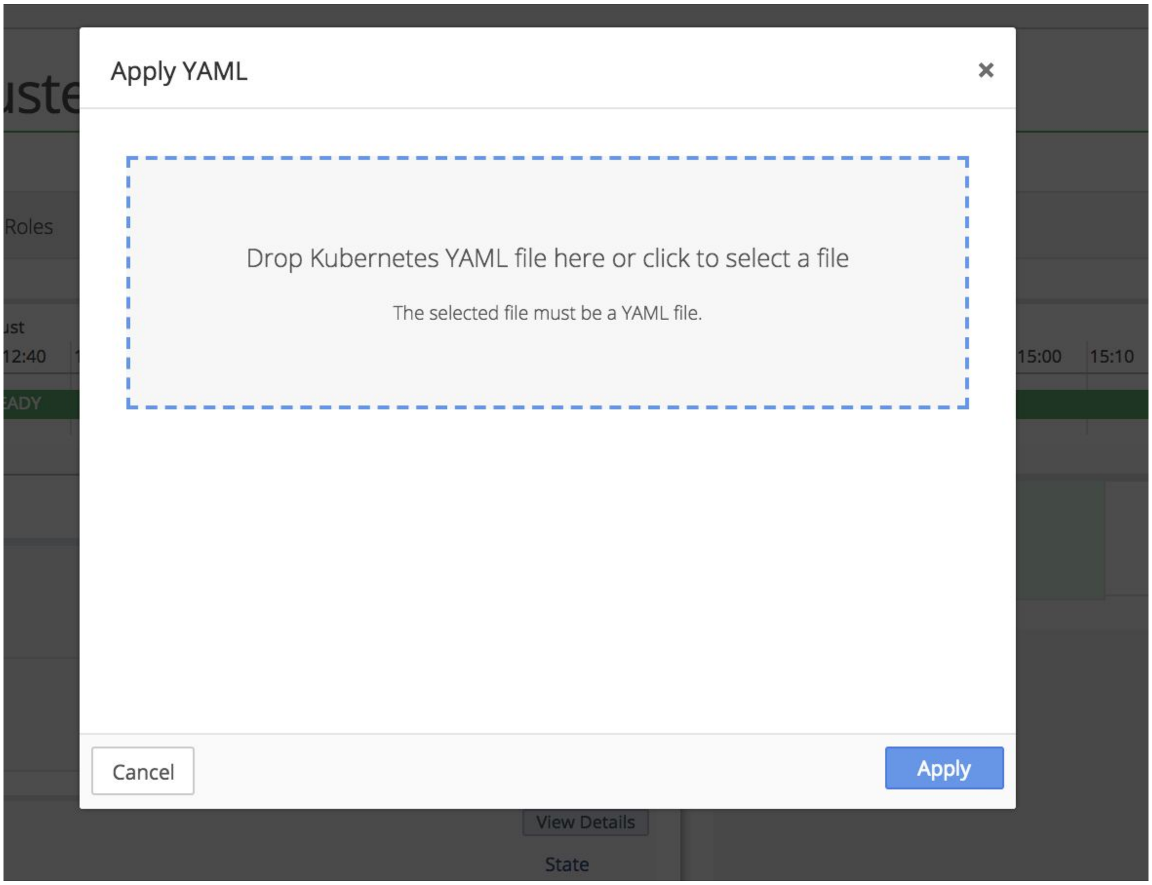


```
        - name: FLEXVOLUME_DIR_PATH
          value: "/opt/nirmata/volume-plugins/"
```

Apply operator.yaml to your cluster through "Apply YAML" option in cluster pulldown menu on top right -



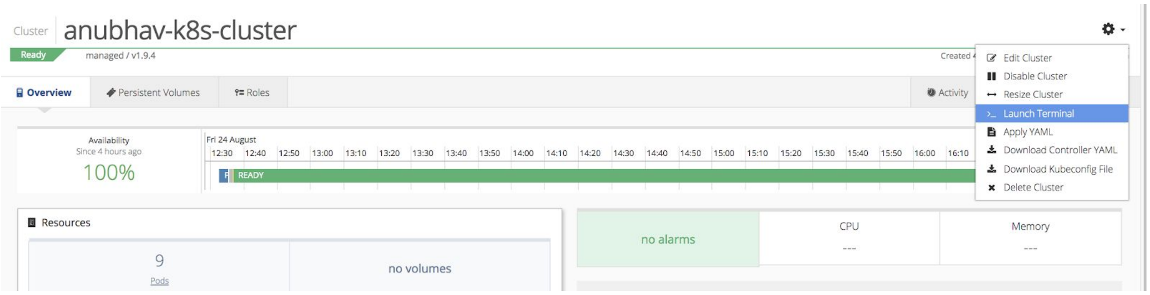
Drop your operator.yaml file here or select the file from directory -

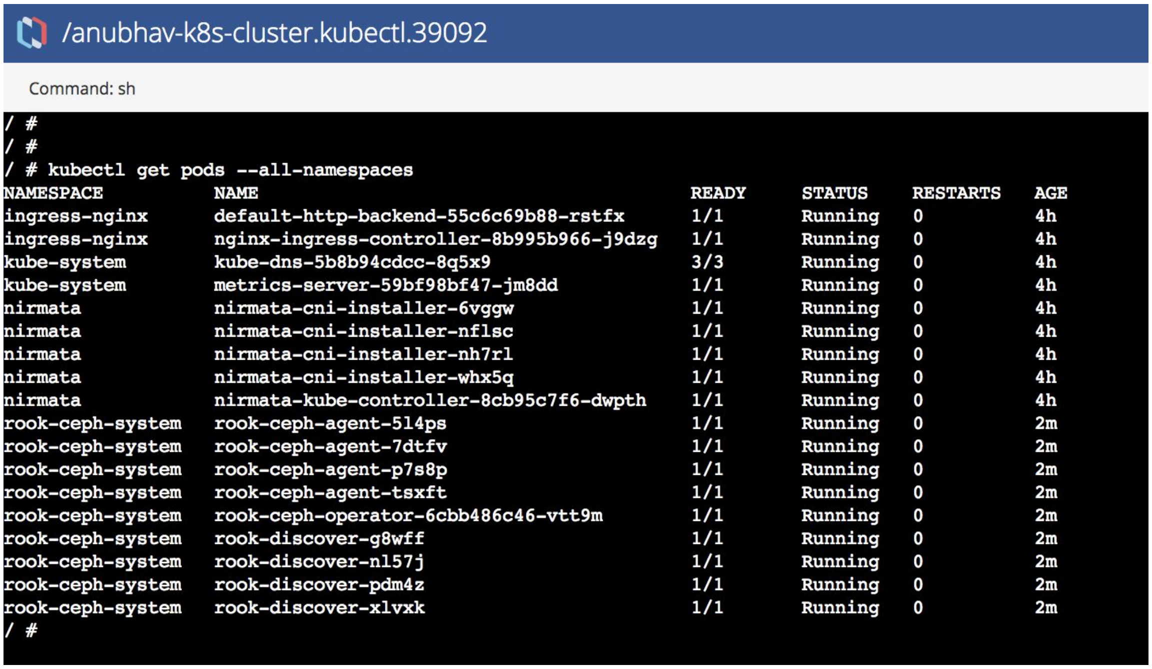


Drop Kubernetes YAML file here or click to select a file

The selected file must be a YAML file.

Use Nirmata shell and run Kubectl command to verify that operator, discover and agent pods are up and running.





/anubhav-k8s-cluster.kubectl.39092

Command: sh

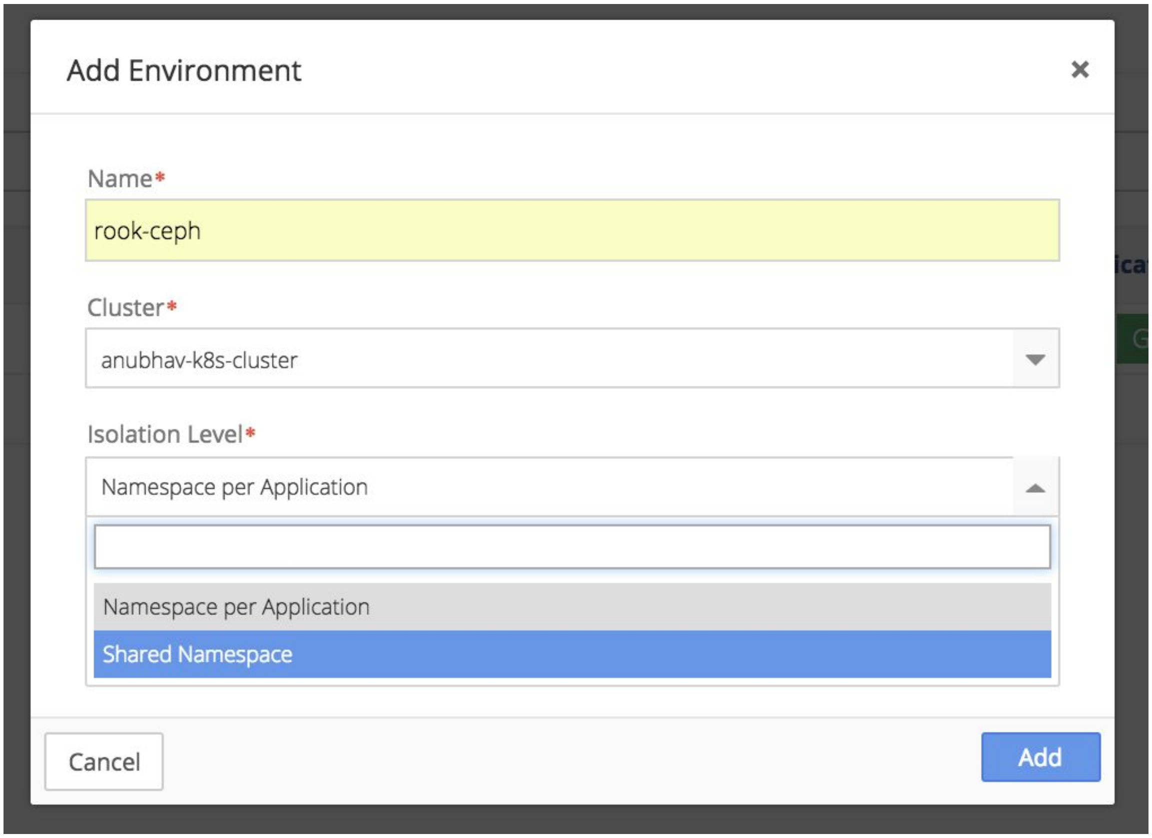
```
/ #
/ #
/ # kubectl get pods --all-namespaces
NAMESPACE          NAME                                       READY     STATUS    RESTARTS   AGE
ingress-nginx      default-http-backend-55c6c69b88-rstfx      1/1       Running   0          4h
ingress-nginx      nginx-ingress-controller-8b995b966-j9dzg   1/1       Running   0          4h
kube-system        kube-dns-5b8b94cdcc-8q5x9                  3/3       Running   0          4h
kube-system        metrics-server-59bf98bf47-jm8dd            1/1       Running   0          4h
nirmata            nirmata-cni-installer-6vggw                1/1       Running   0          4h
nirmata            nirmata-cni-installer-nflsc                1/1       Running   0          4h
nirmata            nirmata-cni-installer-nh7rl                1/1       Running   0          4h
nirmata            nirmata-cni-installer-whx5q                1/1       Running   0          4h
nirmata            nirmata-kube-controller-8cb95c7f6-dwpth    1/1       Running   0          4h
rook-ceph-system   rook-ceph-agent-514ps                      1/1       Running   0          2m
rook-ceph-system   rook-ceph-agent-7dtfv                      1/1       Running   0          2m
rook-ceph-system   rook-ceph-agent-p7s8p                      1/1       Running   0          2m
rook-ceph-system   rook-ceph-agent-tsxft                      1/1       Running   0          2m
rook-ceph-system   rook-ceph-operator-6cbb486c46-vtt9m        1/1       Running   0          2m
rook-ceph-system   rook-discover-g8wff                        1/1       Running   0          2m
rook-ceph-system   rook-discover-nl57j                        1/1       Running   0          2m
rook-ceph-system   rook-discover-pdm4z                        1/1       Running   0          2m
rook-ceph-system   rook-discover-xlvxk                        1/1       Running   0          2m
/ #
```

Configure and deploy cluster.yaml

We modify cluster.yaml for deployment in Nirmata. Cluster.yaml will install in roo-ceph namespace. Nirmata has construct of environments for applications. All applications are deployed in an environment. For application level isolation application namespace is “applicationname-environmentname” and for shared namespace within an environment, it “environmentname”.

For our purpose, we will deploy cluster.yaml as an application in rook-ceph environment. You can do that with following steps -

1. Create rook-ceph environment with shared namespace isolation level.



Add Environment

Name*

rook-ceph

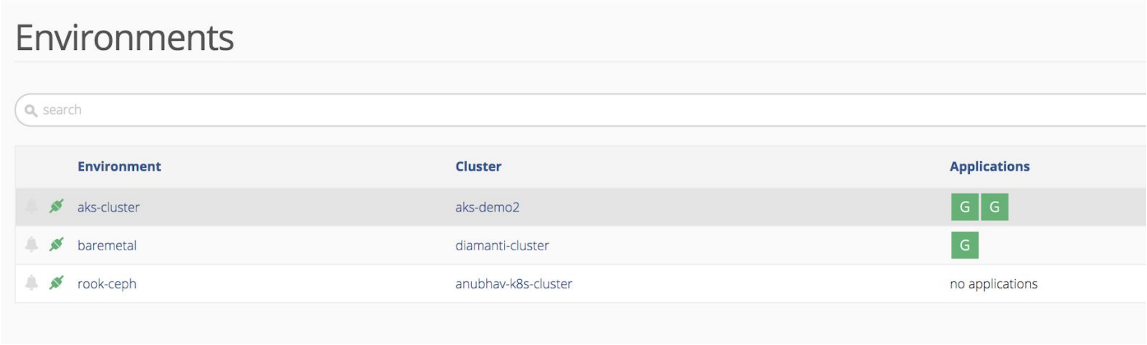
Cluster*

anubhav-k8s-cluster

Isolation Level*

Namespace per Application

Once setup, your environment will look as below -



Environments

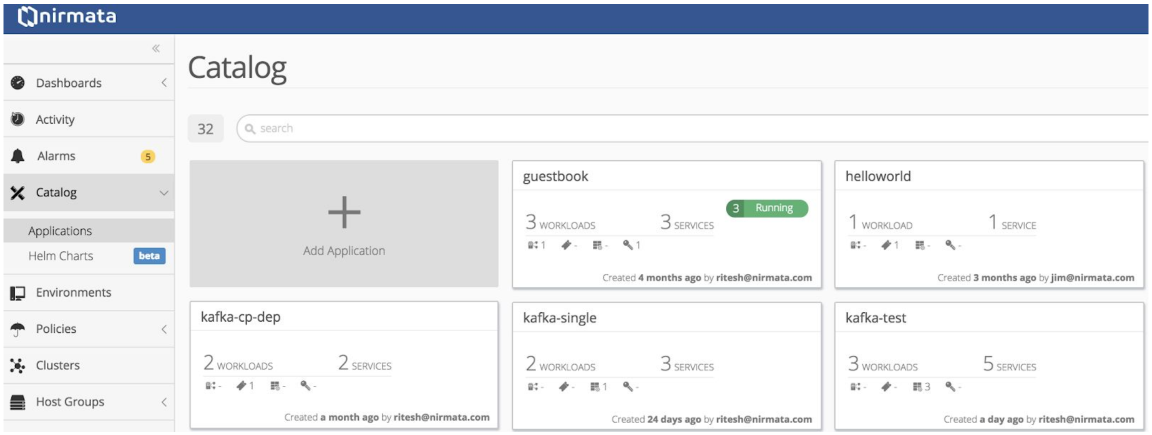
Environment	Cluster	Applications
aks-cluster	aks-demo2	G G
baremetal	diamanti-cluster	G
rook-ceph	anubhav-k8s-cluster	no applications

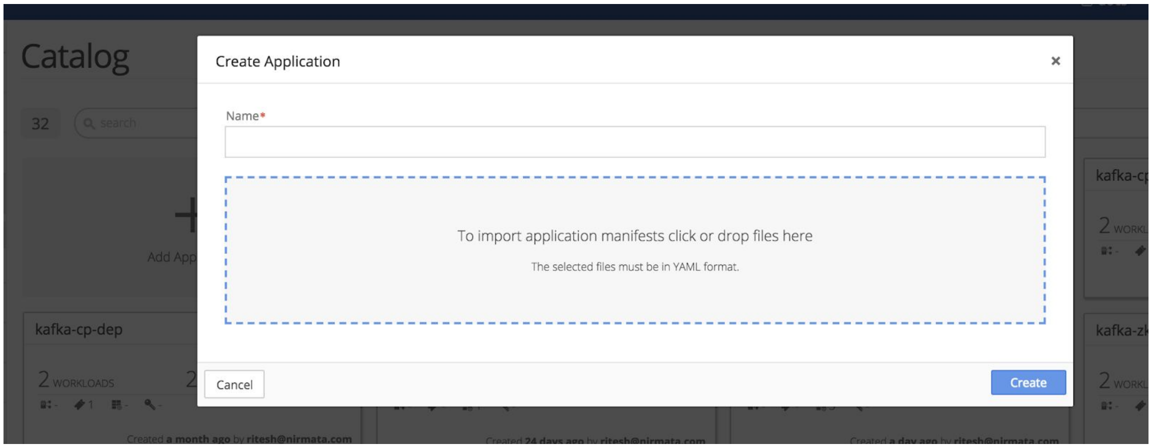
2. Create application “rook-cluster” in application catalog using cluster.yaml as shown below -

Cluster.yaml file spec -

```
apiVersion: ceph.rook.io/v1beta1
kind: Cluster
metadata:
  name: rook-ceph
spec:
  dataDirHostPath: /var/lib/rook
  # The service account under which to run the daemon pods in this cluster if
the default account is not sufficient (OSDs)
  serviceAccount: rook-ceph-cluster
  # set the amount of mons to be started
  mon:
    count: 3
    allowMultiplePerNode: true
  # enable the ceph dashboard for viewing cluster status
  dashboard:
    enabled: true
  network:
    # toggle to use hostNetwork
    hostNetwork: false
# The requests and limits set here, allow the mgr pod to use half of one CPU
core and 1 gigabyte of memory
#    mgr:
#      limits:
#        cpu: "500m"
#        memory: "1024Mi"
#      requests:
#        cpu: "500m"
#        memory: "1024Mi"
# The above example requests/limits can also be added to the mon and osd
components
#    mon:
#    osd:
  storage: # cluster level storage configuration and selection
    useAllNodes: true
    useAllDevices: false
    deviceFilter:
    location:
    config:
      # The default and recommended storeType is dynamically set to bluestore
for devices and filestore for directories.
      # Set the storeType explicitly only if it is required not to use the
default.
      # storeType: bluestore
      databaseSizeMB: "1024" # this value can be removed for environments with
normal sized disks (100 GB or larger)
      journalSizeMB: "1024"  # this value can be removed for environments with
normal sized disks (20 GB or larger)
# Cluster level list of directories to use for storage. These values will be
set for all nodes that have no `directories` set.
#    directories:
#    - path: /rook/storage-dir
# Individual nodes and their config can be specified as well, but 'useAllNodes'
above must be set to false. Then, only the named
# nodes below will be used as storage resources.  Each node's 'name' field
should match their 'kubernetes.io/hostname' label.
#    nodes:
#    - name: "172.17.4.101"
#      directories: # specific directories to use for storage can be specified
for each node
#      - path: "/rook/storage-dir"
#      resources:
#        limits:
#          cpu: "500m"
#          memory: "1024Mi"
#        requests:
#          cpu: "500m"
#          memory: "1024Mi"
#     - name: "172.17.4.201"
#       devices: # specific devices to use for storage can be specified for each
node
#       - name: "sdb"
#       - name: "sdc"
#       config: # configuration can be specified at the node level which
overrides the cluster level config
#         storeType: filestore
#     - name: "172.17.4.301"
#       deviceFilter: "^sd."
```

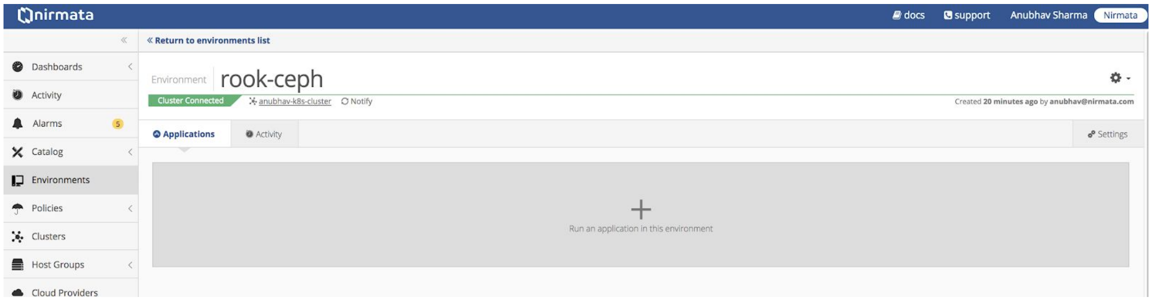
Create a cluster application in the catalog using above cluster.yaml file -




Catalog
32
search
Create Application
×
Name*
To import application manifests click or drop files here
The selected files must be in YAML format.
Cancel
Create
kafka-cp-dep
2 WORKLOADS
kafka-cp
kafka-zk

3. Run cluster application in rook-ceph environment and apply role-binding -

Go to environment rook-ceph and click on run an application tab -

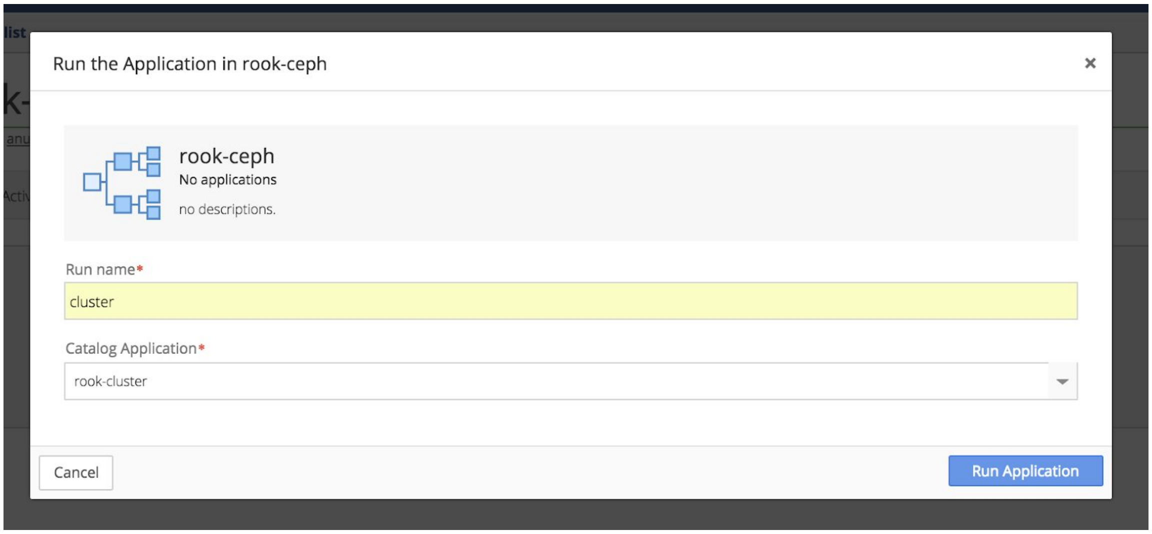


Environment rook-ceph

Cluster Connected anubhav-k8s-cluster Notify

Created **20 minutes ago** by **anubhav@nirmata.com**

Choose the rook-cluster application



Run the Application in rook-ceph

rook-ceph
No applications
no descriptions.

Run name*: cluster

Catalog Application*: rook-cluster

And click “Run Application”.

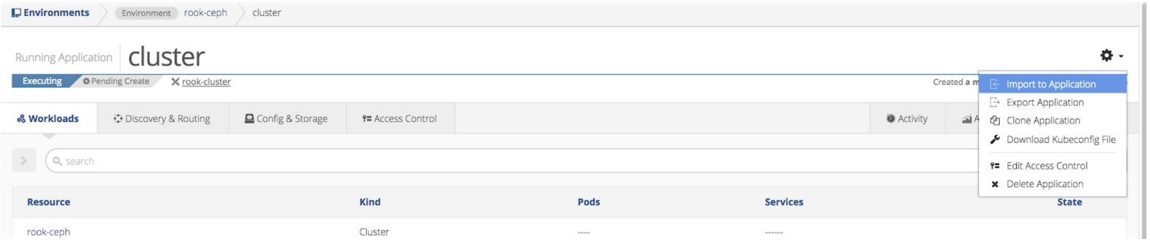
Import role-bindings into the application -

Here is sample yaml for role-binding and Service Account definitions -

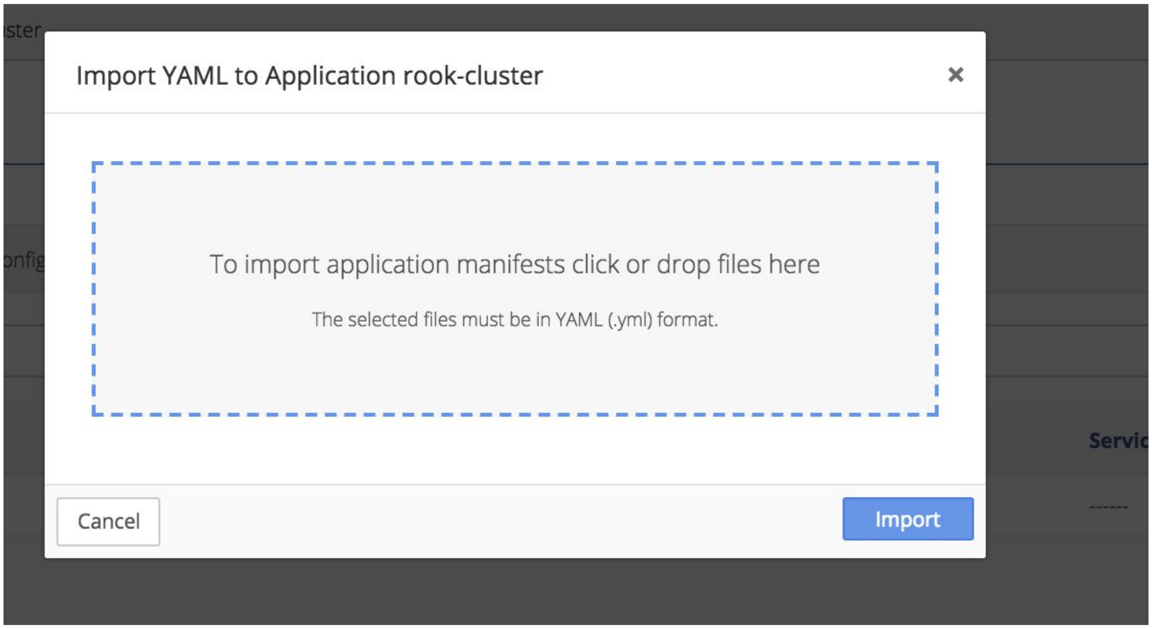
```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: rook-ceph-cluster
  namespace: rook-ceph
---
kind: Role
apiVersion: rbac.authorization.k8s.io/v1beta1
metadata:
  name: rook-ceph-cluster
  namespace: rook-ceph
rules:
- apiGroups: [""]
  resources: ["configmaps"]
  verbs: [ "get", "list", "watch", "create", "update", "delete" ]
---
# Allow the operator to create resources in this cluster's namespace
```

```
kind: RoleBinding
apiVersion: rbac.authorization.k8s.io/v1beta1
metadata:
  name: rook-ceph-cluster-mgmt
  namespace: rook-ceph
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: rook-ceph-cluster-mgmt
subjects:
- kind: ServiceAccount
  name: rook-ceph-system
  namespace: rook-ceph-system
---
# Allow the pods in this namespace to work with configmaps
kind: RoleBinding
apiVersion: rbac.authorization.k8s.io/v1beta1
metadata:
  name: rook-ceph-cluster
  namespace: rook-ceph
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: rook-ceph-cluster
subjects:
- kind: ServiceAccount
  name: rook-ceph-cluster
```

Import the above yaml into the application using menu below -



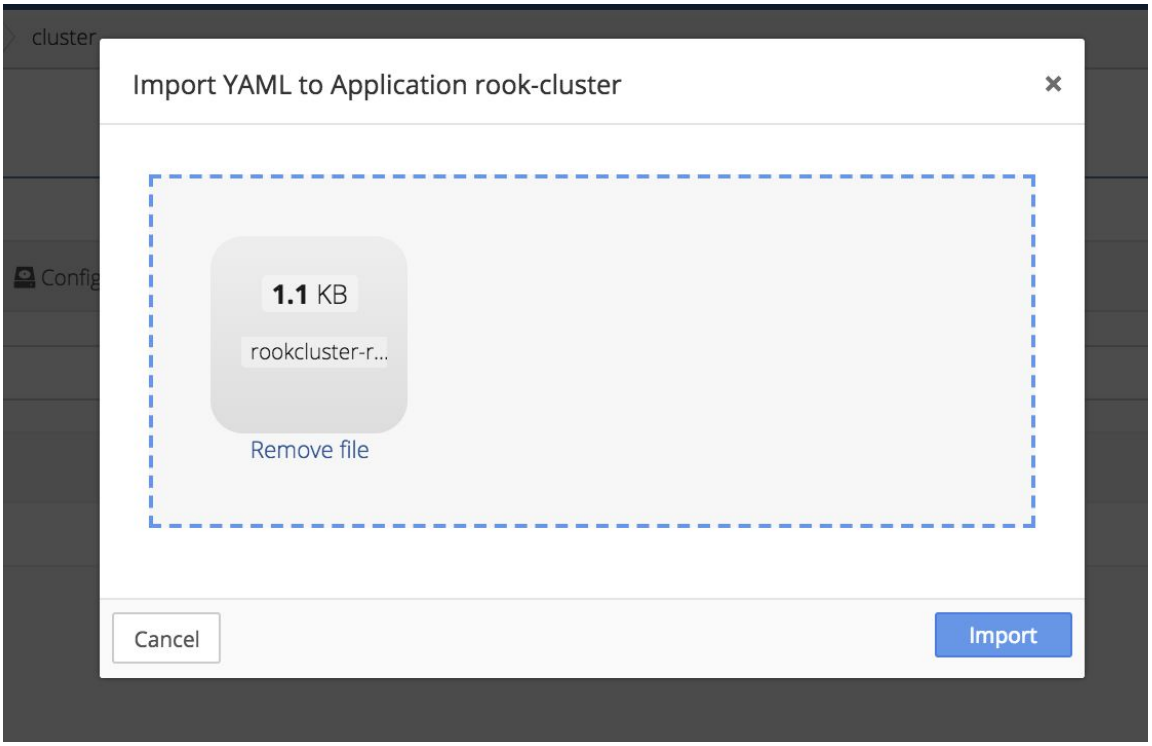
Resource	Kind	Pods	Services	State
rook-ceph	Cluster	----	------	



Import YAML to Application rook-cluster

To import application manifests click or drop files here

The selected files must be in YAML (.yml) format.



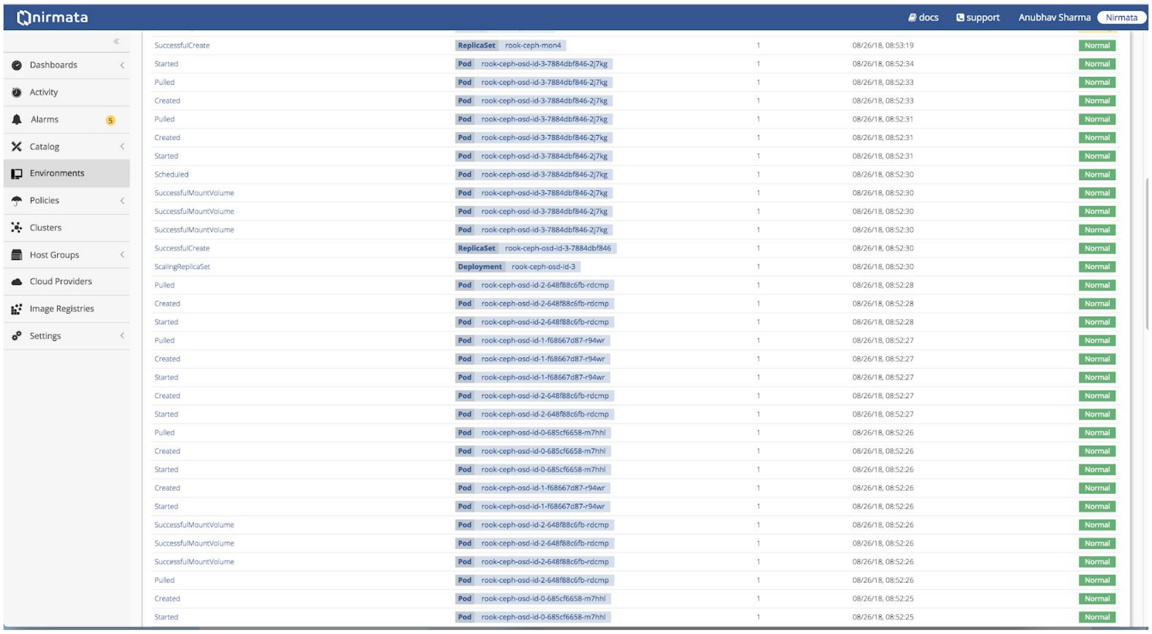
Import YAML to Application rook-cluster

1.1 KB

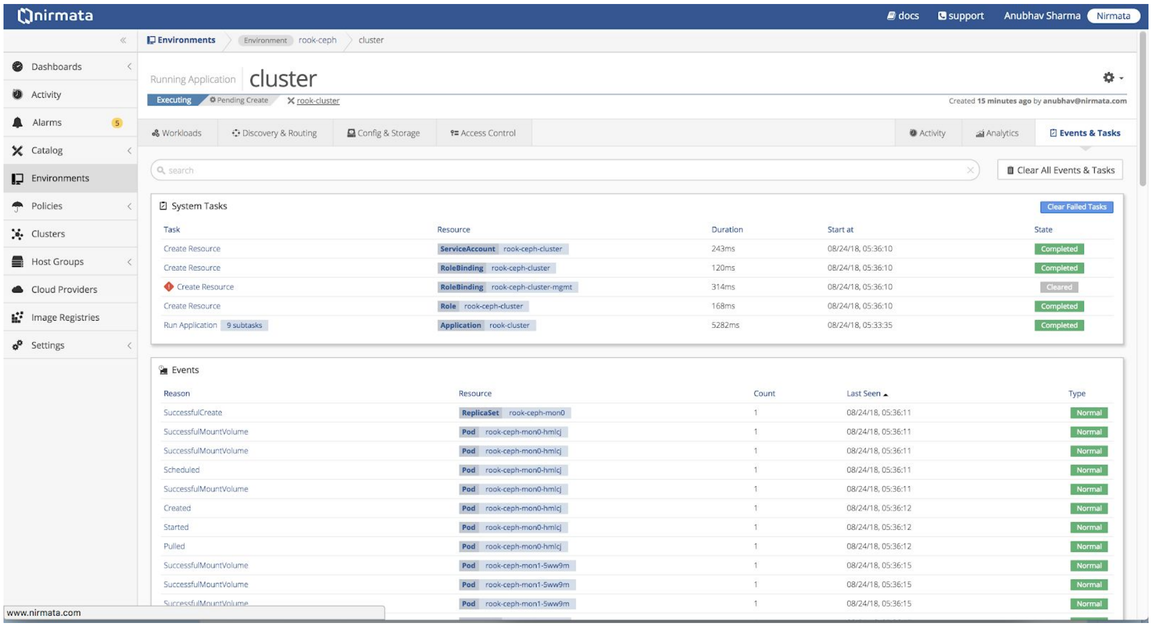
rookcluster-r...

Verify that mon and mgr pods are getting deployed. You can verify by checking events and tasks and by going to cluster shell and running kubectl commands -

Output from Application events and tasks tab -



SuccessfulCreate	ReplicaSet rook-ceph-mon4	1	08/26/18, 08:53:19	Normal
Started	Pod rook-ceph-osd-id-3-7884dbf846-2j7kg	1	08/26/18, 08:52:34	Normal
Pulled	Pod rook-ceph-osd-id-3-7884dbf846-2j7kg	1	08/26/18, 08:52:33	Normal
Created	Pod rook-ceph-osd-id-3-7884dbf846-2j7kg	1	08/26/18, 08:52:33	Normal
Pulled	Pod rook-ceph-osd-id-3-7884dbf846-2j7kg	1	08/26/18, 08:52:31	Normal
Created	Pod rook-ceph-osd-id-3-7884dbf846-2j7kg	1	08/26/18, 08:52:31	Normal
Started	Pod rook-ceph-osd-id-3-7884dbf846-2j7kg	1	08/26/18, 08:52:31	Normal
Scheduled	Pod rook-ceph-osd-id-3-7884dbf846-2j7kg	1	08/26/18, 08:52:30	Normal
SuccessfulMountVolume	Pod rook-ceph-osd-id-3-7884dbf846-2j7kg	1	08/26/18, 08:52:30	Normal
SuccessfulMountVolume	Pod rook-ceph-osd-id-3-7884dbf846-2j7kg	1	08/26/18, 08:52:30	Normal
SuccessfulMountVolume	Pod rook-ceph-osd-id-3-7884dbf846-2j7kg	1	08/26/18, 08:52:30	Normal
SuccessfulCreate	ReplicaSet rook-ceph-osd-id-3-7884dbf846	1	08/26/18, 08:52:30	Normal
ScalingReplicaSet	Deployment rook-ceph-osd-id-3	1	08/26/18, 08:52:30	Normal
Pulled	Pod rook-ceph-osd-id-2-648f88c6fb-rdcmp	1	08/26/18, 08:52:28	Normal
Created	Pod rook-ceph-osd-id-2-648f88c6fb-rdcmp	1	08/26/18, 08:52:28	Normal
Started	Pod rook-ceph-osd-id-2-648f88c6fb-rdcmp	1	08/26/18, 08:52:28	Normal
Pulled	Pod rook-ceph-osd-id-1-f68667d87-r94wr	1	08/26/18, 08:52:27	Normal
Created	Pod rook-ceph-osd-id-1-f68667d87-r94wr	1	08/26/18, 08:52:27	Normal
Started	Pod rook-ceph-osd-id-1-f68667d87-r94wr	1	08/26/18, 08:52:27	Normal
Created	Pod rook-ceph-osd-id-2-648f88c6fb-rdcmp	1	08/26/18, 08:52:27	Normal
Started	Pod rook-ceph-osd-id-2-648f88c6fb-rdcmp	1	08/26/18, 08:52:27	Normal
Pulled	Pod rook-ceph-osd-id-0-685cf6658-m7hhl	1	08/26/18, 08:52:26	Normal
Created	Pod rook-ceph-osd-id-0-685cf6658-m7hhl	1	08/26/18, 08:52:26	Normal
Started	Pod rook-ceph-osd-id-0-685cf6658-m7hhl	1	08/26/18, 08:52:26	Normal
Created	Pod rook-ceph-osd-id-1-f68667d87-r94wr	1	08/26/18, 08:52:26	Normal
Started	Pod rook-ceph-osd-id-1-f68667d87-r94wr	1	08/26/18, 08:52:26	Normal
SuccessfulMountVolume	Pod rook-ceph-osd-id-2-648f88c6fb-rdcmp	1	08/26/18, 08:52:26	Normal
SuccessfulMountVolume	Pod rook-ceph-osd-id-2-648f88c6fb-rdcmp	1	08/26/18, 08:52:26	Normal
SuccessfulMountVolume	Pod rook-ceph-osd-id-2-648f88c6fb-rdcmp	1	08/26/18, 08:52:26	Normal
Pulled	Pod rook-ceph-osd-id-2-648f88c6fb-rdcmp	1	08/26/18, 08:52:26	Normal
Created	Pod rook-ceph-osd-id-0-685cf6658-m7hhl	1	08/26/18, 08:52:25	Normal
Started	Pod rook-ceph-osd-id-0-685cf6658-m7hhl	1	08/26/18, 08:52:25	Normal



Running Application cluster

Executing Pending Create rook-cluster

Created **15 minutes ago** by **anubhav@nirmata.com**

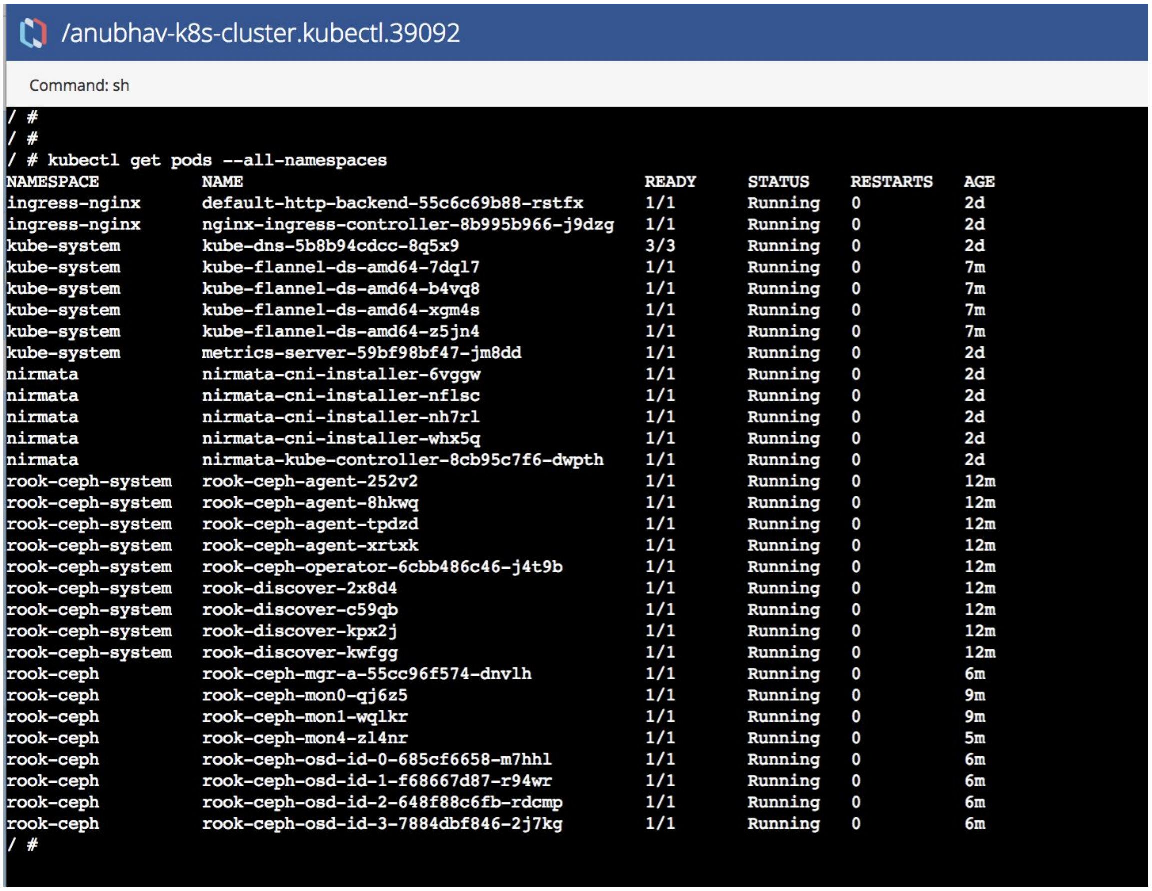
System Tasks

Task	Resource	Duration	Start at	State
Create Resource	ServiceAccount rook-ceph-cluster	243ms	08/24/18, 05:36:10	Completed
Create Resource	RoleBinding rook-ceph-cluster	120ms	08/24/18, 05:36:10	Completed
Create Resource	RoleBinding rook-ceph-cluster-mgmt	314ms	08/24/18, 05:36:10	Cleared
Create Resource	Role rook-ceph-cluster	168ms	08/24/18, 05:36:10	Completed
Run Application 9 subtasks	Application rook-cluster	5282ms	08/24/18, 05:33:35	Completed

Events

Reason	Resource	Count	Last Seen ▲	Type
SuccessfulCreate	ReplicaSet rook-ceph-mon0	1	08/24/18, 05:36:11	Normal
SuccessfulMountVolume	Pod rook-ceph-mon0-hmlcj	1	08/24/18, 05:36:11	Normal
SuccessfulMountVolume	Pod rook-ceph-mon0-hmlcj	1	08/24/18, 05:36:11	Normal
Scheduled	Pod rook-ceph-mon0-hmlcj	1	08/24/18, 05:36:11	Normal
SuccessfulMountVolume	Pod rook-ceph-mon0-hmlcj	1	08/24/18, 05:36:11	Normal
Created	Pod rook-ceph-mon0-hmlcj	1	08/24/18, 05:36:12	Normal
Started	Pod rook-ceph-mon0-hmlcj	1	08/24/18, 05:36:12	Normal
Pulled	Pod rook-ceph-mon0-hmlcj	1	08/24/18, 05:36:12	Normal
SuccessfulMountVolume	Pod rook-ceph-mon1-5ww9m	1	08/24/18, 05:36:15	Normal
SuccessfulMountVolume	Pod rook-ceph-mon1-5ww9m	1	08/24/18, 05:36:15	Normal
SuccessfulMountVolume	Pod rook-ceph-mon1-5ww9m	1	08/24/18, 05:36:15	Normal

Output through Nirmata shell into cluster



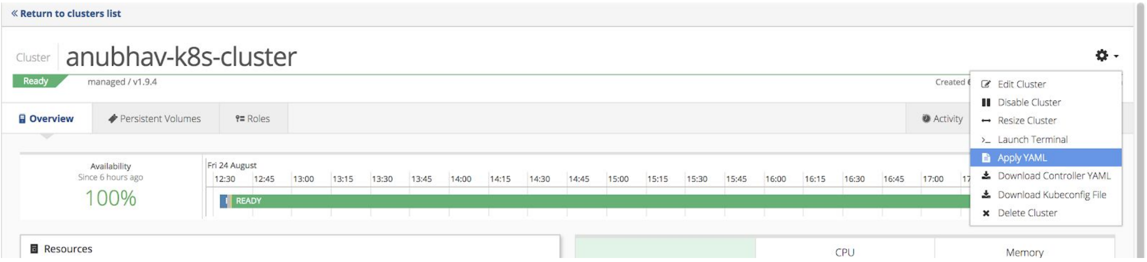
```
/ #
/ #
/ # kubectl get pods --all-namespaces
NAMESPACE          NAME                                      READY     STATUS    RESTARTS    AGE
ingress-nginx      default-http-backend-55c6c69b88-rstfx     1/1       Running   0           2d
ingress-nginx      nginx-ingress-controller-8b995b966-j9dzg  1/1       Running   0           2d
kube-system        kube-dns-5b8b94cdcc-8q5x9                 3/3       Running   0           2d
kube-system        kube-flannel-ds-amd64-7dql7               1/1       Running   0           7m
kube-system        kube-flannel-ds-amd64-b4vq8               1/1       Running   0           7m
kube-system        kube-flannel-ds-amd64-xgm4s               1/1       Running   0           7m
kube-system        kube-flannel-ds-amd64-z5jn4               1/1       Running   0           7m
kube-system        metrics-server-59bf98bf47-jm8dd           1/1       Running   0           2d
nirmata            nirmata-cni-installer-6vggw               1/1       Running   0           2d
nirmata            nirmata-cni-installer-nflsc               1/1       Running   0           2d
nirmata            nirmata-cni-installer-nh7rl               1/1       Running   0           2d
nirmata            nirmata-cni-installer-whx5q               1/1       Running   0           2d
nirmata            nirmata-kube-controller-8cb95c7f6-dwpth   1/1       Running   0           2d
rook-ceph-system   rook-ceph-agent-252v2                     1/1       Running   0           12m
rook-ceph-system   rook-ceph-agent-8hkwq                     1/1       Running   0           12m
rook-ceph-system   rook-ceph-agent-tpdzd                     1/1       Running   0           12m
rook-ceph-system   rook-ceph-agent-xrtxk                     1/1       Running   0           12m
rook-ceph-system   rook-ceph-operator-6cbb486c46-j4t9b       1/1       Running   0           12m
rook-ceph-system   rook-discover-2x8d4                       1/1       Running   0           12m
rook-ceph-system   rook-discover-c59qb                       1/1       Running   0           12m
rook-ceph-system   rook-discover-kpx2j                       1/1       Running   0           12m
rook-ceph-system   rook-discover-kwfgg                       1/1       Running   0           12m
rook-ceph          rook-ceph-mgr-a-55cc96f574-dnvlh          1/1       Running   0           6m
rook-ceph          rook-ceph-mon0-qj6z5                      1/1       Running   0           9m
rook-ceph          rook-ceph-mon1-wqlkr                      1/1       Running   0           9m
rook-ceph          rook-ceph-mon4-zl4nr                      1/1       Running   0           5m
rook-ceph          rook-ceph-osd-id-0-685cf6658-m7hhl        1/1       Running   0           6m
rook-ceph          rook-ceph-osd-id-1-f68667d87-r94wr        1/1       Running   0           6m
rook-ceph          rook-ceph-osd-id-2-648f88c6fb-rdcmp       1/1       Running   0           6m
rook-ceph          rook-ceph-osd-id-3-7884dbf846-2j7kg       1/1       Running   0           6m
/ #
```

Setup storage-class and replica pool -

Use the storage-class yaml for block storage -

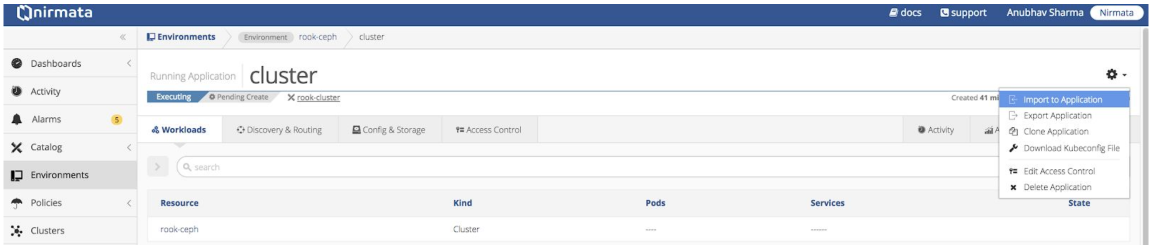
```
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
   name: rook-ceph-block
provisioner: ceph.rook.io/block
parameters:
  pool: replicapool
  # Specify the namespace of the rook cluster from which to create volumes.
  # If not specified, it will use `rook` as the default namespace of the
cluster.
  # This is also the namespace where the cluster will be
  clusterNamespace: rook-ceph
  # Specify the filesystem type of the volume. If not specified, it will use
`ext4`.
#   fstype: xfs
```

Apply the storageclass.yaml to your Kubernetes cluster -

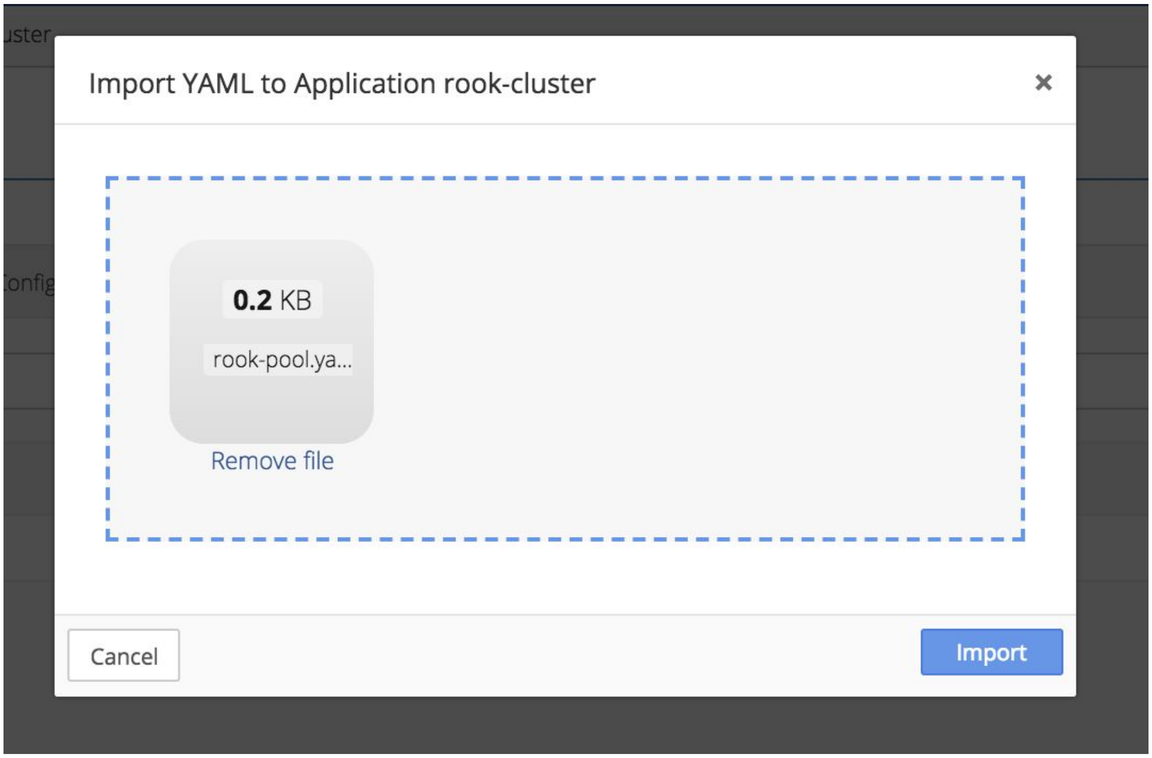


Import replica-pool setting into your cluster application in rook-ceph environment using YAML manifest below -

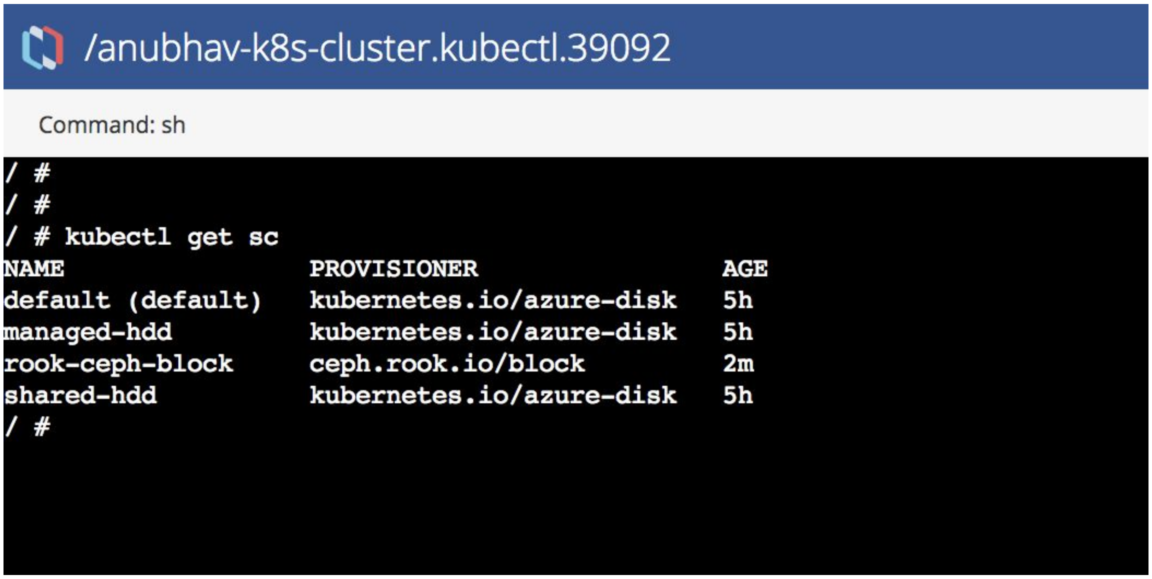
```
apiVersion: ceph.rook.io/v1beta1
kind: Pool
metadata:
  name: replicapool
  namespace: rook-ceph
spec:
  failureDomain: host
  replicated:
    size: 2
```



Resource	Kind	Pods	Services	State
rook-ceph	Cluster	----	------	

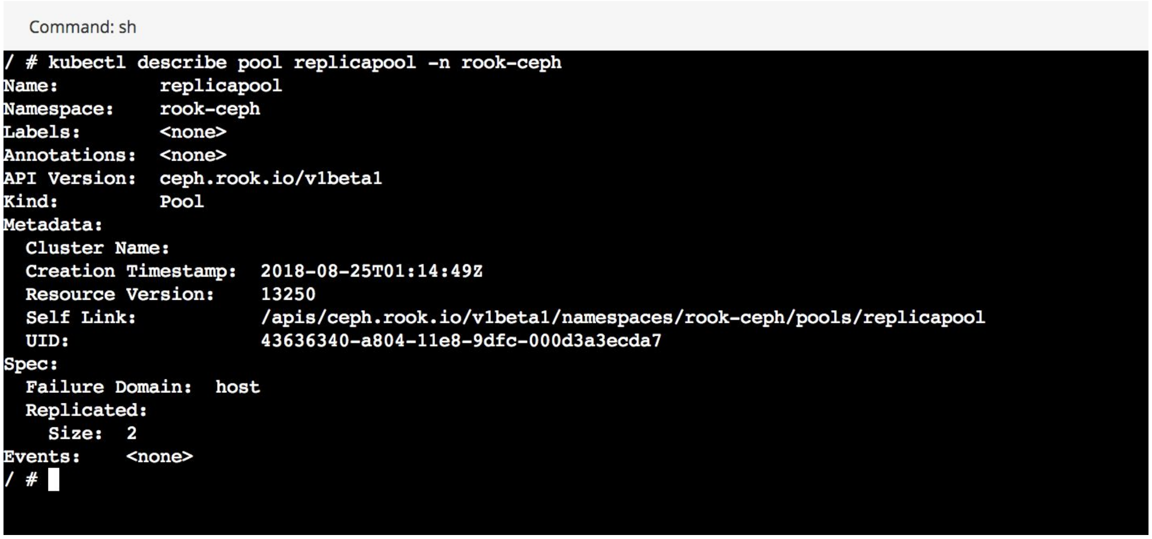


Verify that storage-class is configured on the cluster through NIrmata shell -



```
/ #
/ #
/ # kubectl get sc
NAME                PROVISIONER                 AGE
default (default)   kubernetes.io/azure-disk    5h
managed-hdd         kubernetes.io/azure-disk    5h
rook-ceph-block     ceph.rook.io/block          2m
shared-hdd          kubernetes.io/azure-disk    5h
/ #
```

Verify the pool is setup with replication size of 2 -

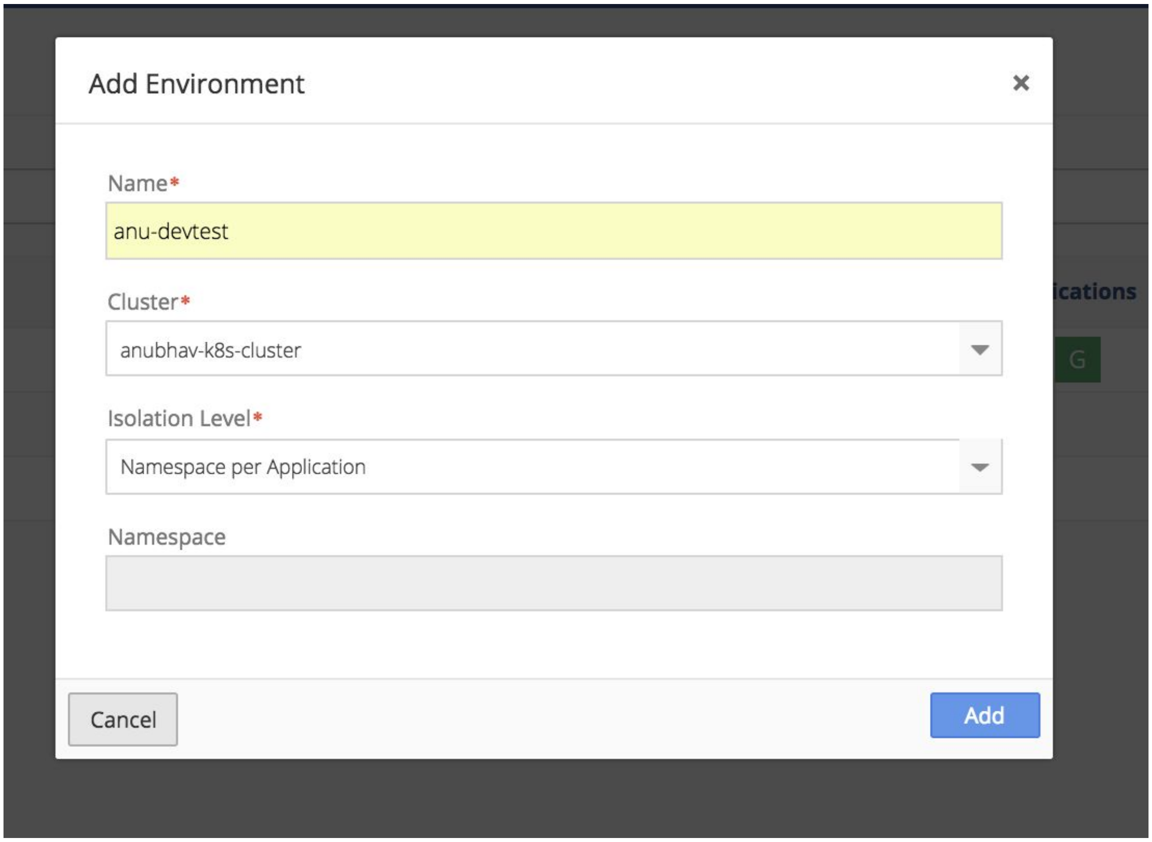


Command: sh

```
/ # kubectl describe pool replicapool -n rook-ceph
Name:         replicapool
Namespace:    rook-ceph
Labels:       <none>
Annotations:  <none>
API Version:  ceph.rook.io/v1beta1
Kind:         Pool
Metadata:
  Cluster Name:
  Creation Timestamp:  2018-08-25T01:14:49Z
  Resource Version:    13250
  Self Link:           /apis/ceph.rook.io/v1beta1/namespaces/rook-ceph/pools/replicapool
  UID:                 43636340-a804-11e8-9dfc-000d3a3ecda7
Spec:
  Failure Domain:  host
  Replicated:
    Size:  2
Events:     <none>
/ #
```

Create a new environment and run your application

Choose Environment menu and create a new environment to run your application -



Name*: anu-devtest

Cluster*: anubhav-k8s-cluster

Isolation Level*: Namespace per Application

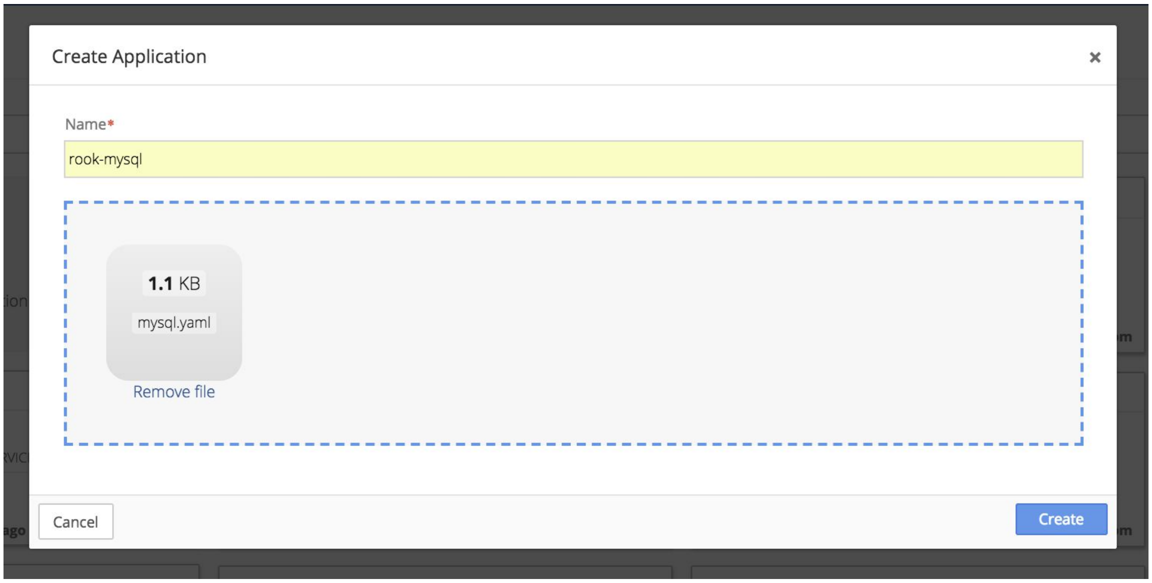
Namespace:

In this example, we will use mysql application with following yaml -

```
apiVersion: v1
kind: Service
metadata:
  name: wordpress-mysql
  labels:
    app: wordpress
spec:
  ports:
    - port: 3306
  selector:
    app: wordpress
    tier: mysql
  clusterIP: None
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: mysql-pv-claim
  labels:
    app: wordpress
spec:
  storageClassName: rook-ceph-block
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 20Gi
---
apiVersion: apps/v1beta1
kind: Deployment
metadata:
  name: wordpress-mysql
  labels:
    app: wordpress
spec:
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        app: wordpress
        tier: mysql
    spec:
      containers:
      - image: mysql:5.6
        name: mysql
        env:
        - name: MYSQL_ROOT_PASSWORD
          value: changeme
        ports:
        - containerPort: 3306
          name: mysql
        volumeMounts:
        - name: mysql-persistent-storage
          mountPath: /var/lib/rook
      volumes:
```

```
      - name: mysql-persistent-storage
        persistentVolumeClaim:
          claimName: mysql-pv-claim
```

Create an application in the catalog using the above yaml -



Create Application

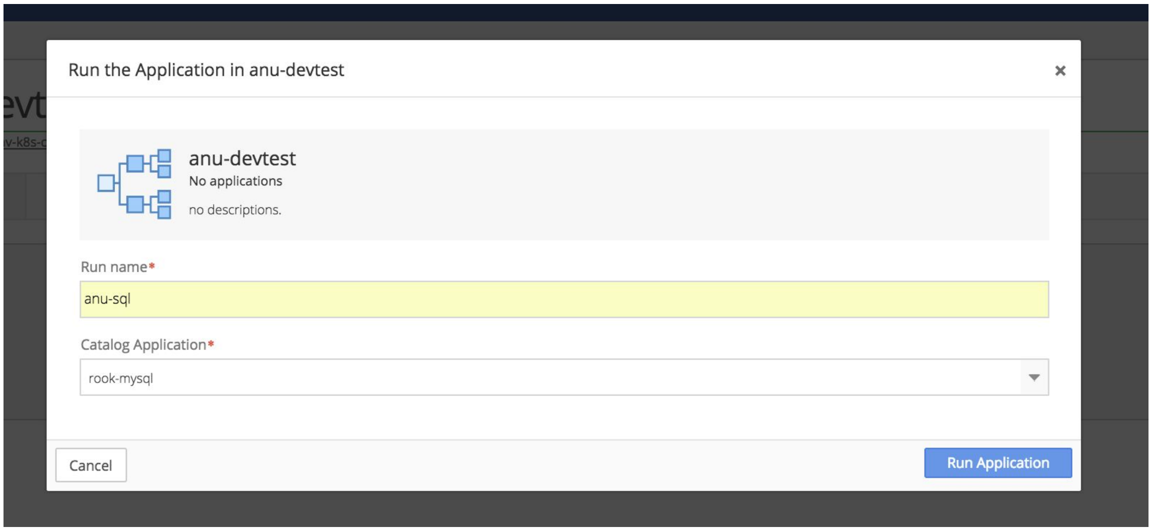
Name*

rook-mysql

1.1 KB

mysql.yaml

Run it in your environment -



Run the Application in anu-devtest

anu-devtest

No applications

no descriptions.

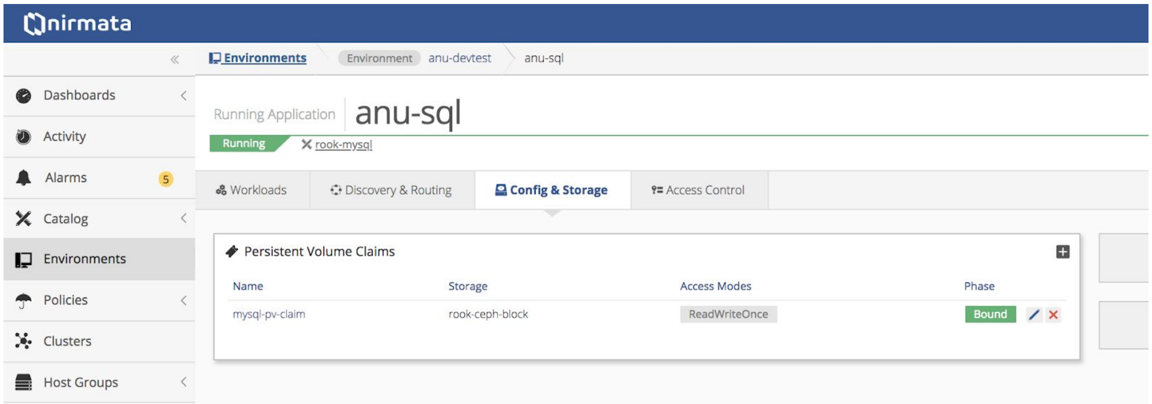
Run name*

anu-sql

Catalog Application*

rook-mysql

Verify that application is running using persistent volumes -



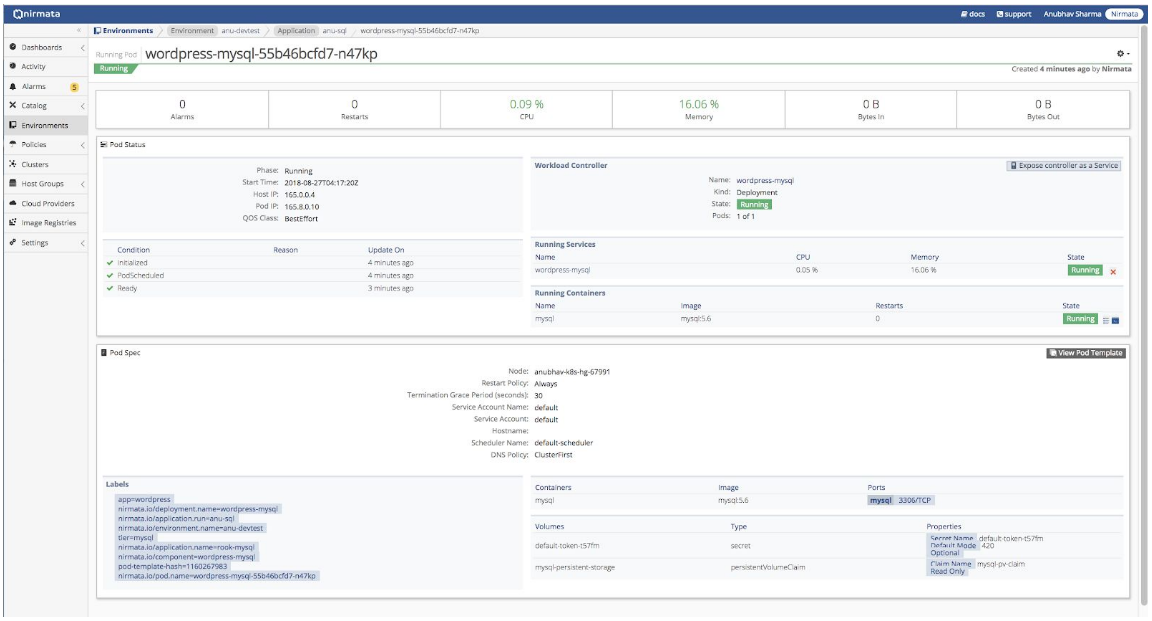
Running Application anu-sql

Running ✕ rook-mysql

Persistent Volume Claims

Name	Storage	Access Modes	Phase
mysql-pv-claim	rook-ceph-block	ReadWriteOnce	Bound

Verify workload status -



Running Pod wordpress-mysql-55b46bcfd7-n47kp

Running

Created **4 minutes ago** by **Nirmata**

Alarms	Restarts	CPU	Memory	Bytes In	Bytes Out
0	0	0.09 %	16.06 %	0 B	0 B

Pod Status

Phase: Running
Start Time: 2018-08-27T04:17:20Z
Host IP: 165.0.0.4
Pod IP: 165.8.0.10
QOS Class: BestEffort

Condition	Reason	Update On
✓ Initialized		4 minutes ago
✓ PodScheduled		4 minutes ago
✓ Ready		3 minutes ago

Workload Controller

Name: wordpress-mysql
Kind: Deployment
State: Running
Pods: 1 of 1

Running Services

Name	CPU	Memory	State
wordpress-mysql	0.05 %	16.06 %	Running

Running Containers

Name	Image	Restarts	State
mysql	mysql:5.6	0	Running

Pod Spec

Node: anubhav-k8s-hg-67991
Restart Policy: Always
Termination Grace Period (seconds): 30
Service Account Name: default
Service Account: default
Hostname:
Scheduler Name: default-scheduler
DNS Policy: ClusterFirst

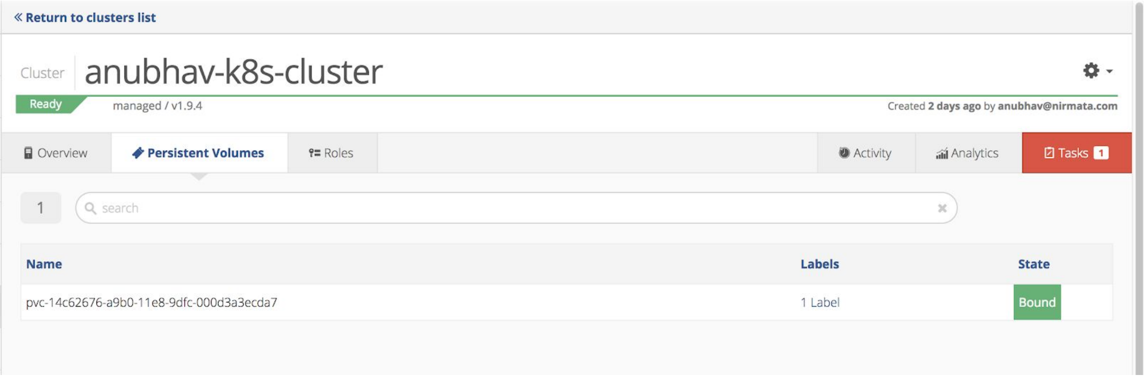
Labels

app=wordpress
nirmata.io/deployment.name=wordpress-mysql
nirmata.io/application.run=anu-sql
nirmata.io/environment.name=anu-devtest
tier=mysql
nirmata.io/application.name=rook-mysql
nirmata.io/component=wordpress-mysql
pod-template-hash=1160267983
nirmata.io/pod.name=wordpress-mysql-55b46bcfd7-n47kp

Containers	Image	Ports
mysql	mysql:5.6	mysql 3306/TCP

Volumes	Type	Properties
default-token-t57fm	secret	Secret Name default-token-t57fm Default Mode 420 Optional
mysql-persistent-storage	persistentVolumeClaim	Claim Name mysql-pv-claim Read Only

Verify persistent volume info



Cluster anubhav-k8s-cluster

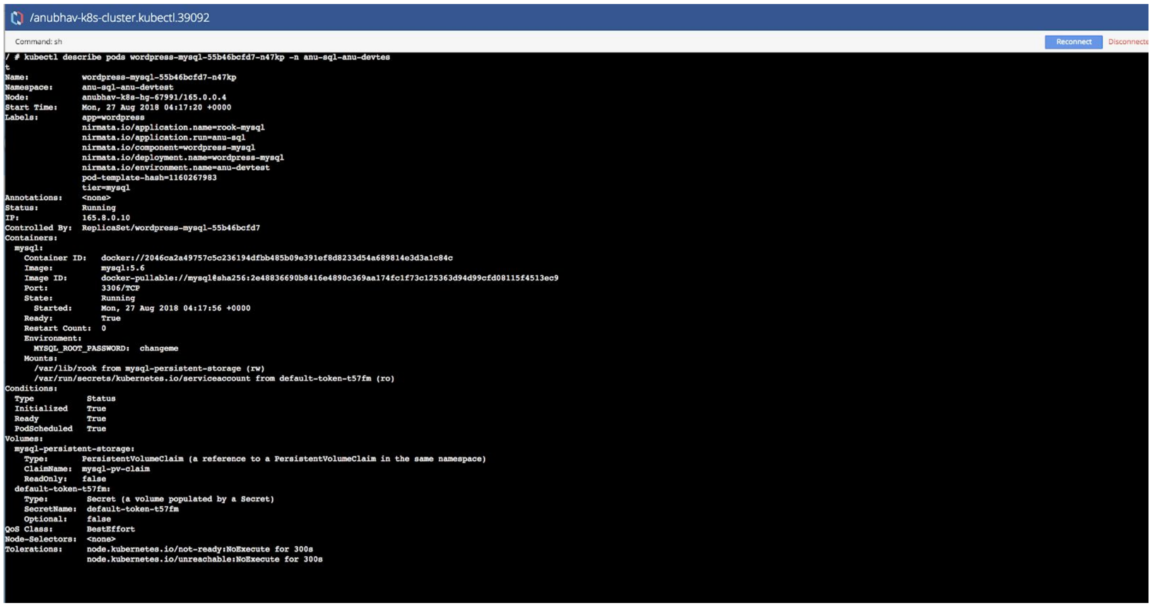
Ready managed / v1.9.4

Created **2 days ago** by **anubhav@nirmata.com**

1

Name	Labels	State
pvc-14c62676-a9b0-11e8-9dfc-000d3a3ecda7	1 Label	Bound

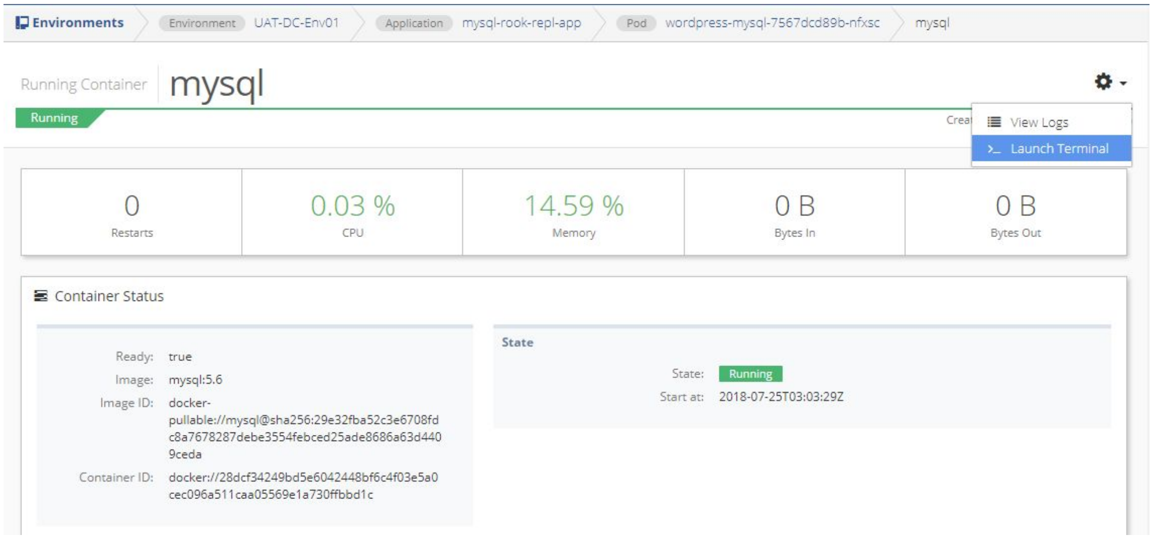
Verify pod status from Nirmata shell -



```
/ # kubectl describe pods wordpress-mysql-55b46bcfd7-n47kp -n anu-sql-anu-devtes
t
Name:           wordpress-mysql-55b46bcfd7-n47kp
Namespace:      anu-sql-anu-devtest
Node:           anubhav-k8s-hg-67991/165.0.0.4
Start Time:     Mon, 27 Aug 2018 04:17:20 +0000
Labels:         app=wordpress
                nirmata.io/application.name=rook-mysql
                nirmata.io/application.run=anu-sql
                nirmata.io/component=wordpress-mysql
                nirmata.io/deployment.name=wordpress-mysql
                nirmata.io/environment.name=anu-devtest
                pod-template-hash=1160267983
                tier=mysql
Annotations:    <none>
Status:         Running
IP:             165.8.0.10
Controlled By:  ReplicaSet/wordpress-mysql-55b46bcfd7
Containers:
  mysql:
    Container ID:   docker://2046ca2a49757c5c236194dfbb485b09e391ef8d8233d54a689814e3d3a1c84c
    Image:          mysql:5.6
    Image ID:       docker-pullable://mysql@sha256:2e48836690b8416e4890c369aa174fc1f73c125363d94d99cfd08115f4513ec9
    Port:           3306/TCP
    State:          Running
      Started:      Mon, 27 Aug 2018 04:17:56 +0000
    Ready:          True
    Restart Count:  0
    Environment:
      MYSQL_ROOT_PASSWORD:  changeme
    Mounts:
      /var/lib/rook from mysql-persistent-storage (rw)
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-t57fm (ro)
Conditions:
  Type           Status
  Initialized    True
  Ready          True
  PodScheduled   True
Volumes:
  mysql-persistent-storage:
    Type:       PersistentVolumeClaim (a reference to a PersistentVolumeClaim in the same namespace)
    ClaimName:  mysql-pv-claim
    ReadOnly:   false
  default-token-t57fm:
    Type:        Secret (a volume populated by a Secret)
    SecretName:  default-token-t57fm
    Optional:    false
QoS Class:       BestEffort
Node-Selectors:  <none>
Tolerations:     node.kubernetes.io/not-ready:NoExecute for 300s
                 node.kubernetes.io/unreachable:NoExecute for 300s
```

Verify data persistence across replication with host failure scenario -

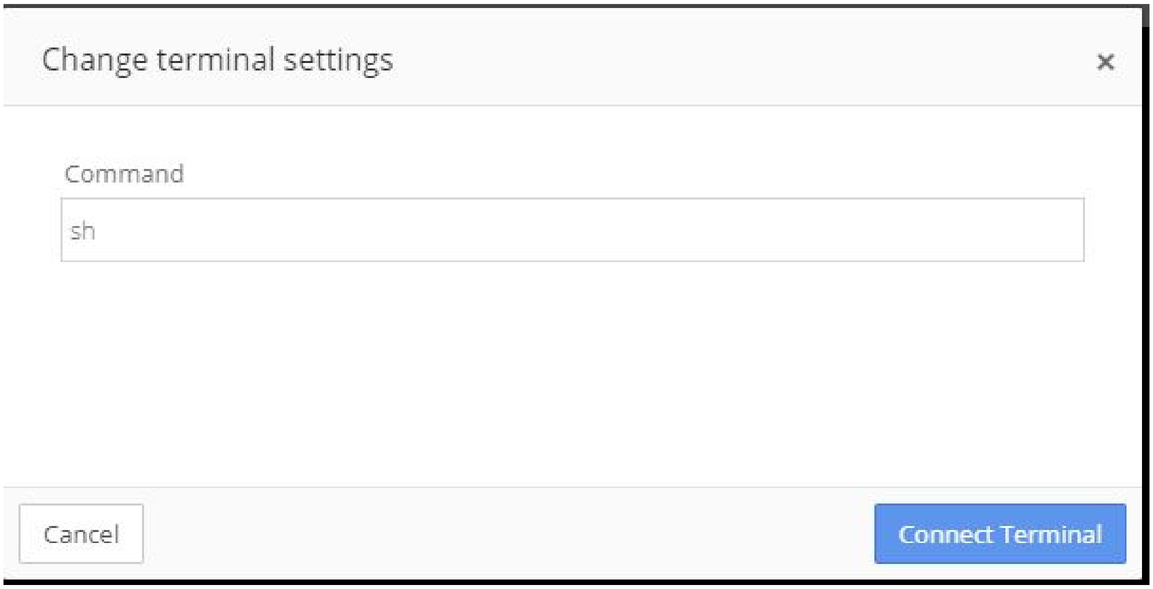
1. Navigate to Environments→*Environment Name*→*Application*→*Pod Name*
2. Click the container name (mysql) under Running Containers
3. Click the Gear Icon on top right→Launch Terminal



Ready: true
Image: mysql:5.6
Image ID: docker-pullable://mysql@sha256:29e32fba52c3e6708fdc8a7678287debe3554febced25ade8686a63d4409ceda
Container ID: docker://28dcf34249bd5e6042448bf6c4f03e5a0cec096a511caa05569e1a730ffbbd1c

State: Running
Start at: 2018-07-25T03:03:29Z

4. Leave sh as Command, click Connect Terminal

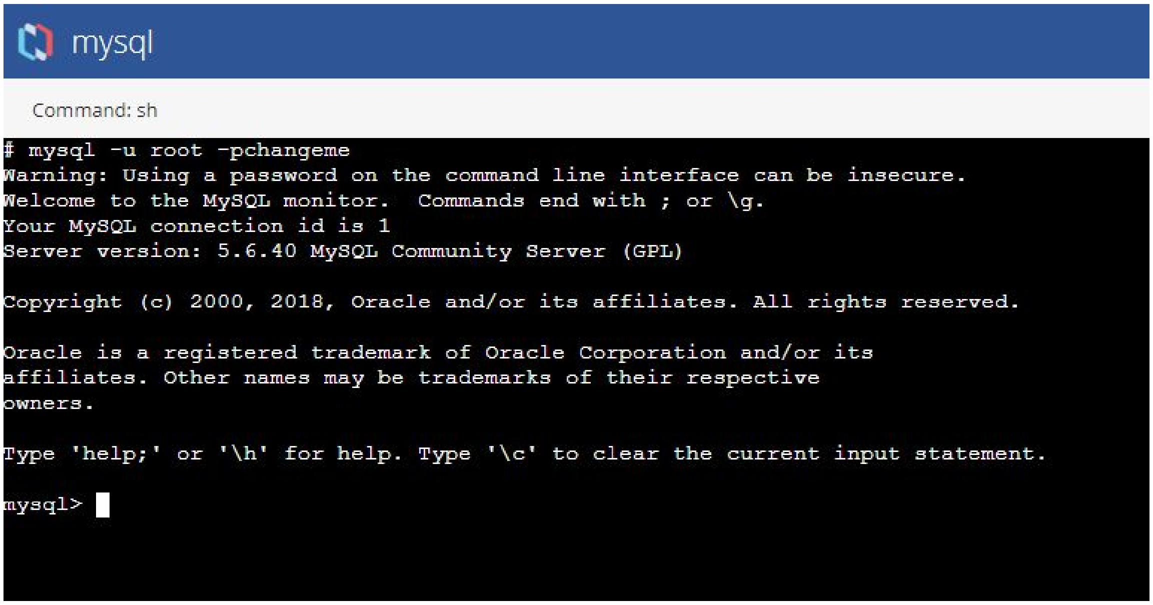


Change terminal settings

Command

sh

5. In the terminal window, enter mysql command to connect to mysql database as root user
mysql -u root -pchangeme



Command: sh

```
# mysql -u root -pchangeme
Warning: Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.6.40 MySQL Community Server (GPL)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

6. Create new database

mysql> *create database testrook;*

```
Query OK, 1 row affected (0.00 sec)
```

7. Verify

mysql> *show databases;*

```
+---------------------+
| Database            |
+---------------------+
| information_schema  |
| #mysql50#lost+found |
| mysql               |
| performance_schema  |
| testrook            |
+---------------------+
5 rows in set (0.01 sec)
```

8. Connect to new database

```
mysql> use testrook;
Database changed
```

9. Create new table

```
mysql> create table employee (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name varchar(20),
    dept varchar(10),
    salary int(10));

Query OK, 0 rows affected (0.12 sec)
```

10. Insert few records in the new table

```
mysql> insert into employee values(100,'Thomas','Sales',5000);
Query OK, 1 row affected (0.08 sec)

mysql> insert into employee values(200,'Jason','Technology',5500);
Query OK, 1 row affected (0.11 sec)

mysql> insert into employee values(300,'Mayla','Technology',7000);
Query OK, 1 row affected (0.06 sec)

mysql> insert into employee values(400,'Nisha','Marketing',9500);
Query OK, 1 row affected (0.05 sec)

mysql> ;
Query OK, 1 row affected (0.05 sec)
```

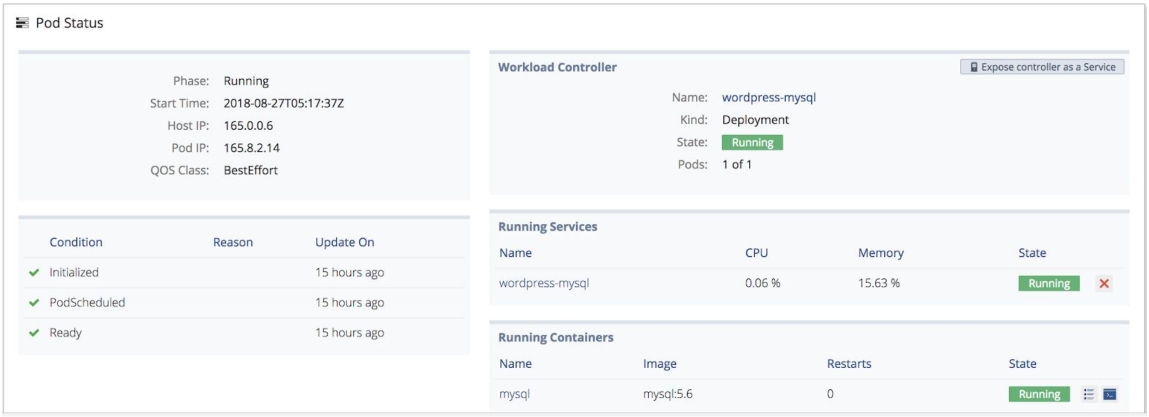
11. Verify

```
mysql> select * from employee;
+-----+--------+------------+--------+
| id  | name   | dept       | salary |
+-----+--------+------------+--------+
| 100 | Thomas | Sales      |   5000 |
| 200 | Jason  | Technology |   5500 |
| 300 | Mayla  | Technology |   7000 |
| 400 | Nisha  | Marketing  |   9500 |
| 500 | Randy  | Technology |   6000 |
+-----+--------+-----------+--------+
5 rows in set (0.00 sec)
```

12. Exit from mysql and terminal
mysql> exit

Disable host running the current pod

1. Find the Host where the Pod is running by navigating to Environments→*Environment Name*→*Application*→*Pod Name* (Host IP shown under Pod Status)



Pod Status

Phase: Running
Start Time: 2018-08-27T05:17:37Z
Host IP: 165.0.0.6
Pod IP: 165.8.2.14
QOS Class: BestEffort

Condition	Reason	Update On
✓ Initialized		15 hours ago
✓ PodScheduled		15 hours ago
✓ Ready		15 hours ago

Workload Controller

Expose controller as a Service

Name: wordpress-mysql
Kind: Deployment
State: Running
Pods: 1 of 1

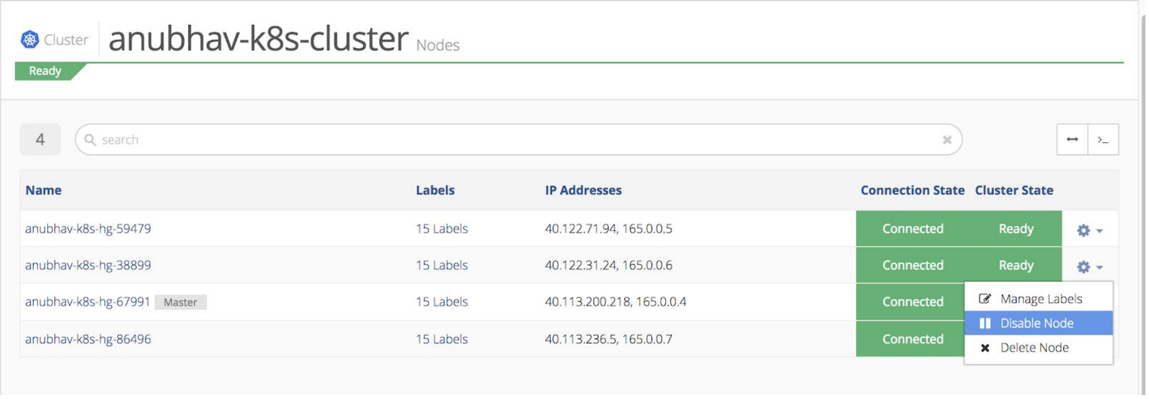
Running Services

Name	CPU	Memory	State
wordpress-mysql	0.06 %	15.63 %	Running ✕

Running Containers

Name	Image	Restarts	State
mysql	mysql:5.6	0	Running

2. As an admin user, go to Clusters→*Cluster Name (running this environment)*. Click View details in Node box, click the gear icon next to the Host running the Pod, click Disable Node



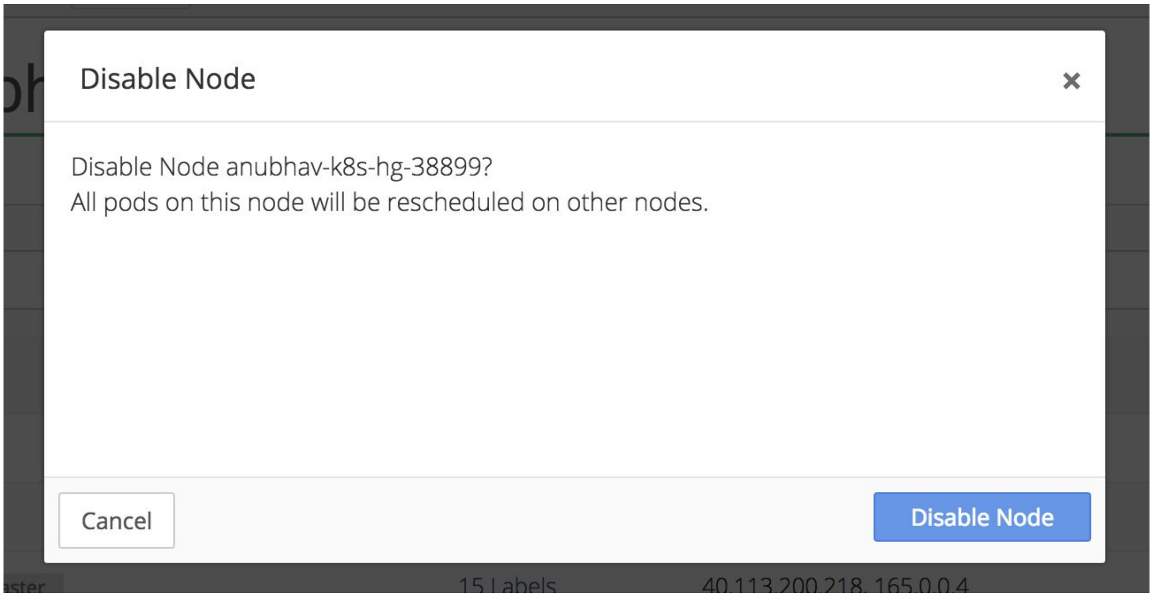
Cluster anubhav-k8s-cluster Nodes

Ready

4 search

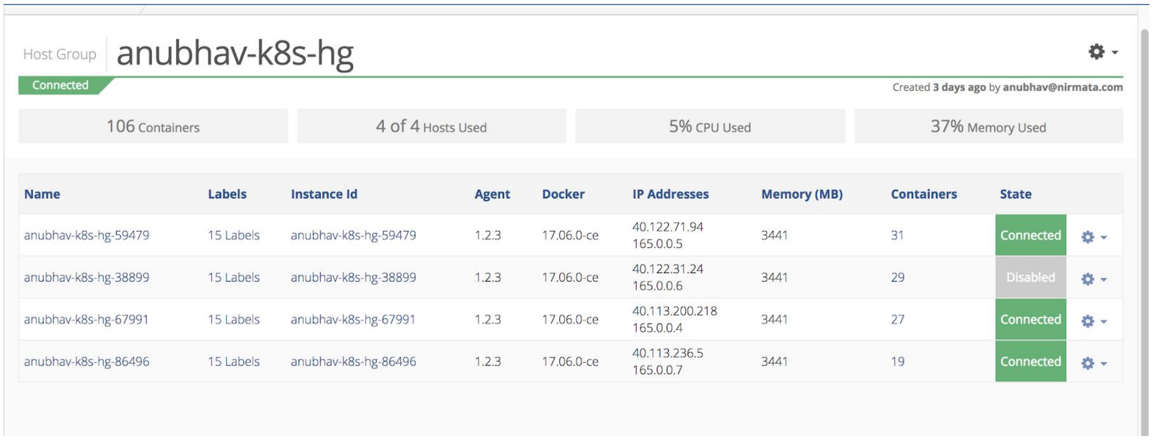
Name	Labels	IP Addresses	Connection State	Cluster State
anubhav-k8s-hg-59479	15 Labels	40.122.71.94, 165.0.0.5	Connected	Ready
anubhav-k8s-hg-38899	15 Labels	40.122.31.24, 165.0.0.6	Connected	Ready
anubhav-k8s-hg-67991 Master	15 Labels	40.113.200.218, 165.0.0.4	Connected	
anubhav-k8s-hg-86496	15 Labels	40.113.236.5, 165.0.0.7	Connected	

4.Click Disable Node to confirm



Disable Node

Disable Node anubhav-k8s-hg-38899?
All pods on this node will be rescheduled on other nodes.



Host Group anubhav-k8s-hg

Connected

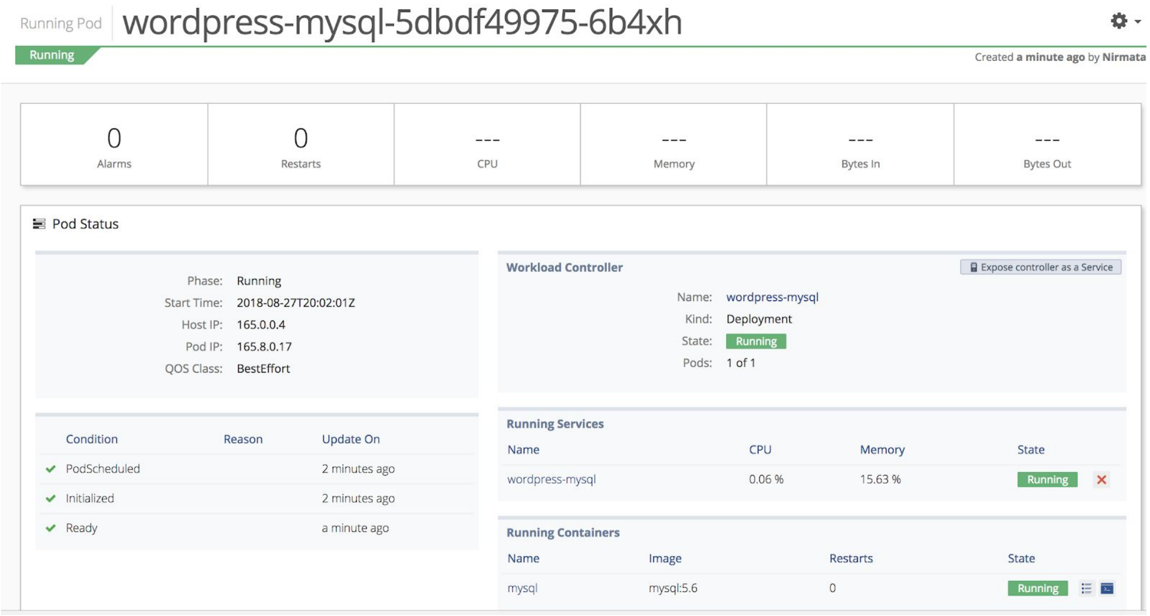
Created **3 days ago** by **anubhav@nirmata.com**

106 Containers | 4 of 4 Hosts Used | 5% CPU Used | 37% Memory Used

Name	Labels	Instance Id	Agent	Docker	IP Addresses	Memory (MB)	Containers	State
anubhav-k8s-hg-59479	15 Labels	anubhav-k8s-hg-59479	1.2.3	17.06.0-ce	40.122.71.94 165.0.0.5	3441	31	Connected
anubhav-k8s-hg-38899	15 Labels	anubhav-k8s-hg-38899	1.2.3	17.06.0-ce	40.122.31.24 165.0.0.6	3441	29	Disabled
anubhav-k8s-hg-67991	15 Labels	anubhav-k8s-hg-67991	1.2.3	17.06.0-ce	40.113.200.218 165.0.0.4	3441	27	Connected
anubhav-k8s-hg-86496	15 Labels	anubhav-k8s-hg-86496	1.2.3	17.06.0-ce	40.113.236.5 165.0.0.7	3441	19	Connected

Verify data in the new Pod

On disabling the node where the original Pod was running, Kubernetes will reschedule the Pod on another available host in the cluster. Pod name will be different from the original one. Verify by navigating to Environments→*Environment Name→Application→Pod Name*



Running Pod wordpress-mysql-5dbdf49975-6b4xh

Running

Created **a minute ago** by **Nirmata**

0	0	---	---	---	---
Alarms	Restarts	CPU	Memory	Bytes In	Bytes Out

Pod Status

Phase: Running
Start Time: 2018-08-27T20:02:01Z
Host IP: 165.0.0.4
Pod IP: 165.8.0.17
QOS Class: BestEffort

Condition	Reason	Update On
✓ PodScheduled		2 minutes ago
✓ Initialized		2 minutes ago
✓ Ready		a minute ago

Workload Controller

Name: wordpress-mysql
Kind: Deployment
State: Running
Pods: 1 of 1

Running Services

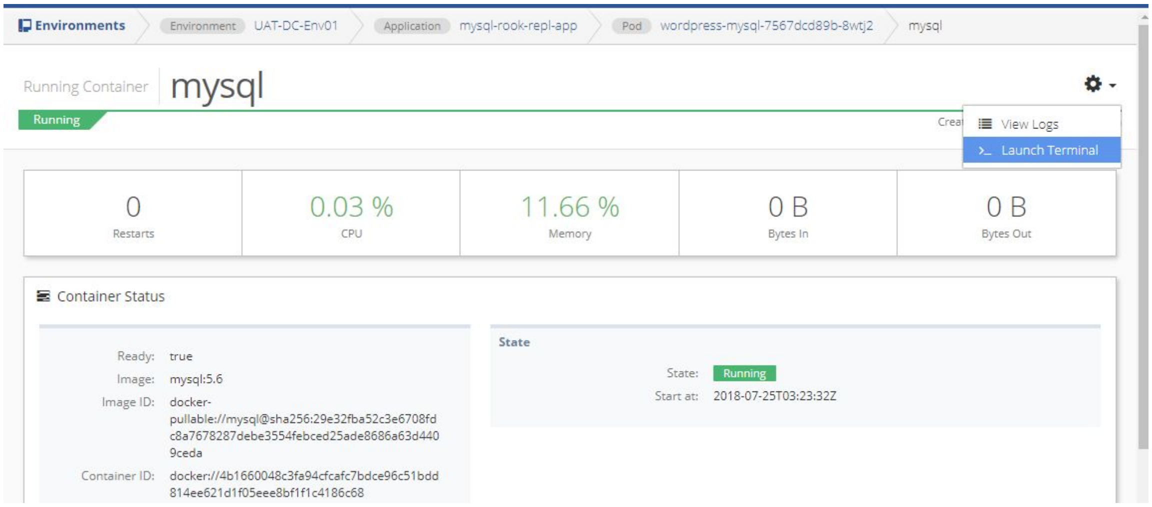
Name	CPU	Memory	State
wordpress-mysql	0.06 %	15.63 %	Running

Running Containers

Name	Image	Restarts	State
mysql	mysql:5.6	0	Running

Login to mysql database

1. Navigate to Environments→*Environment Name*→*Application*→*Pod Name*→mysql (under Running containers). Click the gear icon on top right→Launch Terminal



Running Container mysql

Running

0	0.03 %	11.66 %	0 B	0 B
Restarts	CPU	Memory	Bytes In	Bytes Out

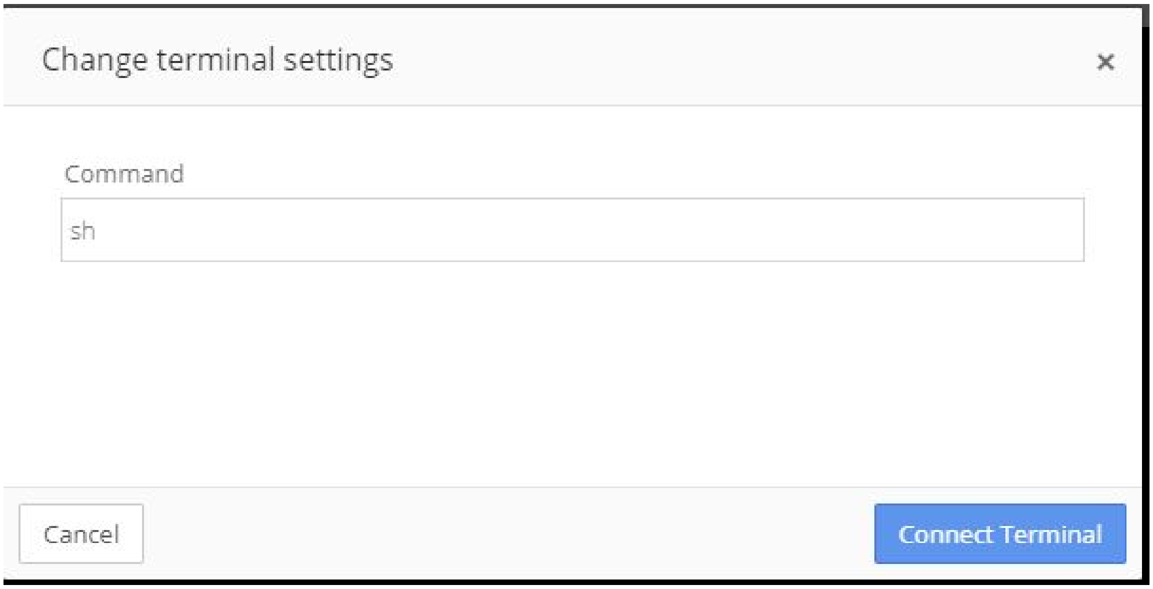
Container Status

Ready: true
Image: mysql:5.6
Image ID: docker-pullable://mysql@sha256:29e32fba52c3e6708fdc8a7678287debe3554febced25ade8686a63d4409ceda
Container ID: docker://4b1660048c3fa94cfcafc7bdce96c51bdd814ee621d1f05eee8bf1f1c4186c68

State

State: Running
Start at: 2018-07-25T03:23:32Z

1. Leave sh as Command, click Connect Terminal

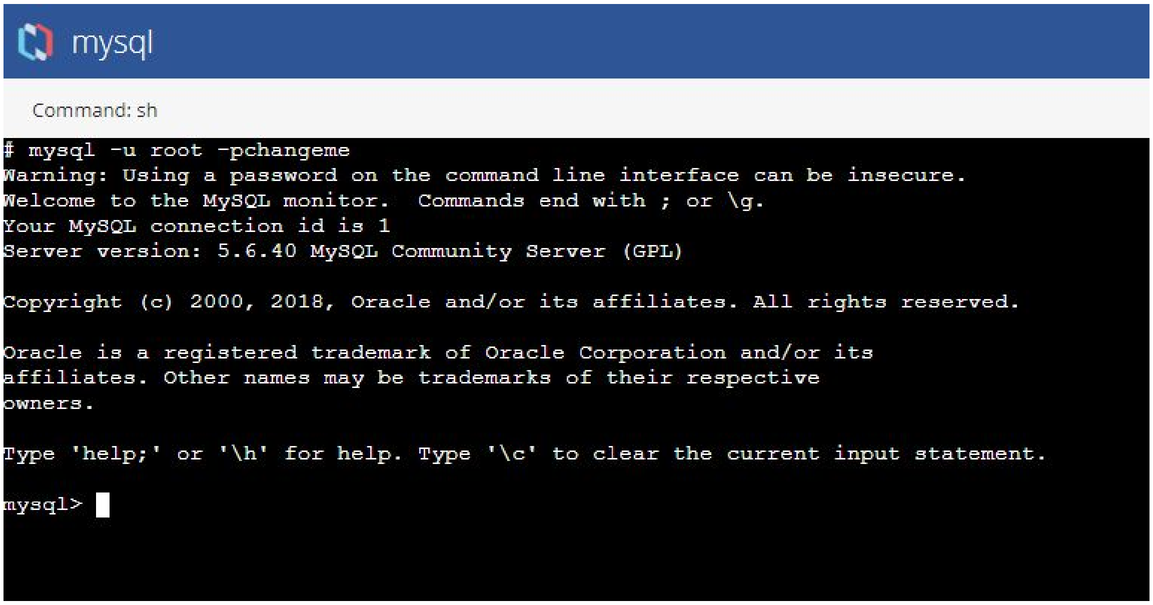


Change terminal settings

Command

sh

2. In the terminal window, enter mysql command to connect to mysql database as root user
mysql -u root -pchangeme



mysql

Command: sh

```
# mysql -u root -pchangeme
Warning: Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.6.40 MySQL Community Server (GPL)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

3. Verify "testrook" database

mysql> *show databases;*

```
+--------------------+
| Database           |
+--------------------+
| information_schema |
| #mysql50#lost+found |
| mysql              |
| performance_schema  |
| testrook            |
+---------------------+
5 rows in set (0.01 sec)
```

4. Connect to testrook database

```
mysql> use testrook;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
```

5. Verify records in employee table

```
mysql> select * from employee;
+-----+--------+------------+--------+
| id  | name   | dept       | salary |
+-----+--------+------------+--------+
| 100 | Thomas | Sales      |   5000 |
| 200 | Jason  | Technology |   5500 |
| 300 | Mayla  | Technology |   7000 |
| 400 | Nisha  | Marketing  |   9500 |
| 500 | Randy  | Technology |   6000 |
+-----+--------+------------+--------+
5 rows in set (0.00 sec)
```

6. Exit from mysql and terminal

```
mysql>exit;
```