TestCoLab

tanzu-ubuntu-fips-node-rwo-rwx-validation-notes-proc-rvw-01

Updated 1/9/2024

[Procedure review notes: 4](#_Toc155724988)

[Overview 4](#_Toc155724989)

[1/9/2024 dr-apps-01-cluster\_v1.26.10\_ubuntu guest cluster 5](#_Toc155724990)

[k get tkr | grep ubuntu 5](#_Toc155724991)

[dr-apps-01-cluster\_v1.26.10\_ubuntu.yaml" 5](#_Toc155724992)

[k get nodes 6](#_Toc155724993)

[k describe [worker node] 6](#_Toc155724994)

[Kernel Version: 5.4.0-166-generic 6](#_Toc155724995)

[Kubelet Version: v1.26.10+vmware.1-fips.1 6](#_Toc155724996)

[OS-Image: Ubuntu 20.04.6 LTS 6](#_Toc155724997)

[Container Runtime Version: containerd://1.6.24 6](#_Toc155724998)

[Portworx version 3.0.4.0-1396ef3 6](#_Toc155724999)

[kubectl get secret -n dr-01 dr-apps-01-kubeconfig -o jsonpath='{.data.value}' | base64 -d > config\_dr-apps-01 11](#_Toc155725000)

[k create ns portworx; label ns: pod-security privileged audit, enforce, warn; rolebinding to clusterrole psp:vmware-system-privileged 11](#_Toc155725001)

[k create ns central; label ns: pod-security privileged audit, enforce, warn; rolebinding to clusterrole psp:vmware-system-privileged 11](#_Toc155725002)

[kubectl apply -f 'https://install.portworx.com/3.0?comp=pxoperator&kbver=1.26.10&ns=portworx' 11](#_Toc155725003)

[kubectl label [worker nodes] px/metadata-node=true 11](#_Toc155725004)

[kubectl apply -f 'https://install.portworx.com/3.0?operator=true&mc=false&kbver=1.26.10&ns=portworx&b=true&iop=6&csicd=true&mz=8&s=%22sc%3Ddr-tanzu-storage-01%2Csize%3D150%22&j=auto&c=px-cluster-a05d75b0-3ec3-44e5-bf96-09ce929ba2ac&stork=true&csi=true&mon=true&tel=false&st=k8s&e=PRE-EXEC%3Diptables%20-A%20INPUT%20-p%20tcp%20--match%20multiport%20--dports%209001%3A9020%20-j%20ACCEPT%2CPRE-EXEC%3Diptables%20-A%20INPUT%20-p%20tcp%20--match%20multiport%20--dports%202049%20-j%20ACCEPT&promop=true' 11](#_Toc155725005)

[kubectl get pods -n portworx -l name=portworx | grep -v NAME | awk '{print $1}' 12](#_Toc155725006)

[Portworx pvcs, kvdb pods provisioning 14](#_Toc155725007)

[Pxctl status 15](#_Toc155725008)

[1/9/2024 PX provisioning test/cfm - RWO 16](#_Toc155725009)

[Prep default namespace 16](#_Toc155725010)

[pxd-repl1-portworx-sc.yaml 16](#_Toc155725011)

[pvc-repl1-01.yaml 16](#_Toc155725012)

[sc, pvc created, bound 16](#_Toc155725013)

[deploy-rwo-px-nginx-test.yaml 18](#_Toc155725014)

[kubectl get all,pvc -n default 19](#_Toc155725015)

[Pxctl volume list 19](#_Toc155725016)

[Pxctl volume inspect 19](#_Toc155725017)

[1/9/2024 PX provisioning test/cfm - RWX 20](#_Toc155725018)

[Prep default namespace 20](#_Toc155725019)

[pxd-repl1-portworx-sc.yaml 20](#_Toc155725020)

[pvc-repl3-01.yaml 20](#_Toc155725021)

[sc, pvc created, bound 20](#_Toc155725022)

[deploy-rwx-px-nginx-test.yaml 22](#_Toc155725023)

[kubectl get all,pvc -n default 23](#_Toc155725024)

[Pxctl volume list 23](#_Toc155725025)

[Pxctl volume inspect 23](#_Toc155725026)

[Exec into each rwx pod, write data, review other pods’ written data 25](#_Toc155725027)

# Procedure review notes:

## Overview

* These are informal, as-is observations to help illuminate a testing and troubleshooting process.
* The configuration overview is:

dr-tanzu01 TKGS Supervisor cluster running on vSphere v7.0.3

dr-apps-01 TKGS guest cluster

* The guest control and worker nodes are using the 1.26.10+vmware.1-fips.1-tkg.1.ubuntu distribution
* The issues being considered are:

What are the steps to deploy the the 1.26.10+vmware.1-fips.1-tkg.1.ubuntu distribution on a TKGS guest cluster

What are the steps to prepare the cluster and namespaces to support a Portworx deployment with RWO and RWX volumes?

What are the steps to prepare the guest worker nodes to host the Portworx kvdb metadata volumes?

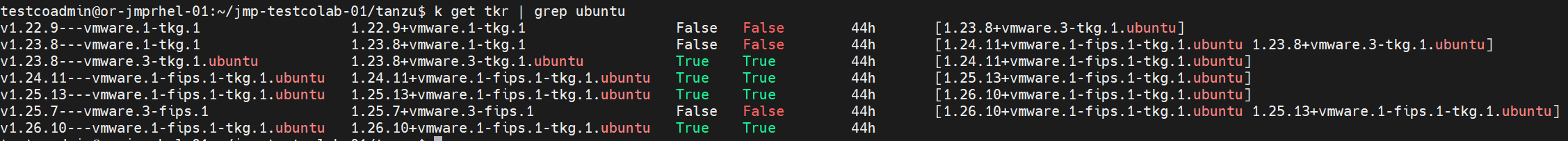
What are the artifacts and procedures to validate a single, RWO deployment on a Portworx provisioned volume?

What are the artifacts and procedures to validate a single, RWX deployment on a Portworx provisioned volume?

What are the steps to confirm the ability of multiple pods to read and write data to a RWX deployment on a Portworx provisioned volume?

# 1/9/2024 dr-apps-01-cluster\_v1.26.10\_ubuntu guest cluster

## k get tkr | grep ubuntu



## dr-apps-01-cluster\_v1.26.10\_ubuntu.yaml"

apiVersion: run.tanzu.vmware.com/v1alpha1

kind: TanzuKubernetesCluster

metadata:

name: dr-apps-01

namespace: dr-01

spec:

distribution:

**fullVersion: 1.26.10+vmware.1-fips.1-tkg.1.ubuntu**

topology:

controlPlane:

count: 1

class: best-effort-small

storageClass: dr-tanzu-storage-01

workers:

count: 3

class: best-effort-medium

storageClass: dr-tanzu-storage-01

volumes:

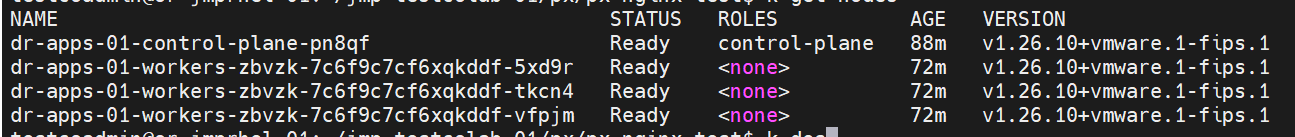
* name: containerd

mountPath: /var/lib/containerd

capacity:

storage: 32Gi

## k get nodes



## k describe [worker node]

## Kernel Version: 5.4.0-166-generic

## Kubelet Version: v1.26.10+vmware.1-fips.1

## OS-Image: Ubuntu 20.04.6 LTS

## Container Runtime Version: containerd://1.6.24

## Portworx version 3.0.4.0-1396ef3

Defaulted container "portworx" out of: portworx, csi-node-driver-registrar

Volume : 494233727207238724

Name : pvc-6679d209-30d1-4134-87e1-552c15056cfa

Size : 2.0 GiB

Format : ext4

HA : 1

IO Priority : LOW

Creation time : Jan 10 03:43:53 UTC 2024

Shared : v4 (service)

Status : up

State : Attached: c3aa1842-ec5e-4234-933d-003585bb5099 (10.2.6.108)

Last Attached : Jan 10 03:47:55 UTC 2024

Device Path : /dev/pxd/pxd494233727207238724

Labels : namespace=default,pvc=pvc-repl3-01,repl=1

Mount Options : discard

Sharedv4 Client Mount Options : soft,timeo=600,vers=4.0,actimeo=60,port=2049,proto=tcp,retrans=4

Reads : 11

Reads MS : 3

Bytes Read : 45056

Writes : 46

Writes MS : 107

Bytes Written : 33636352

IOs in progress : 0

Bytes used : 1.1 MiB

Replica sets on nodes:

Set 0

Node : 10.2.6.108 (Pool 1ee254c9-76a8-4ade-a51b-3b9f99bd0041 )

Replication Status : Up

Volume consumers :

- Name : deploy-rwx-px-nginx-test-5f5867cd44-7gpn6 (11a424bd-7a64-4a98-b525-3c2fa8ca83f6) (Pod)

Namespace : default

Running on : dr-apps-01-workers-zbvzk-7c6f9c7cf6xqkddf-5xd9r

Controlled by : deploy-rwx-px-nginx-test-5f5867cd44 (ReplicaSet)

- Name : deploy-rwx-px-nginx-test-5f5867cd44-gl5d2 (4de5549f-55f5-47d6-9d73-ec70c48a82f8) (Pod)

Namespace : default

Running on : dr-apps-01-workers-zbvzk-7c6f9c7cf6xqkddf-5xd9r

Controlled by : deploy-rwx-px-nginx-test-5f5867cd44 (ReplicaSet)

- Name : deploy-rwx-px-nginx-test-5f5867cd44-pms7q (6e622f3f-fbbd-47aa-8bda-9906b02efb23) (Pod)

Namespace : default

Running on : dr-apps-01-workers-zbvzk-7c6f9c7cf6xqkddf-vfpjm

Controlled by : deploy-rwx-px-nginx-test-5f5867cd44 (ReplicaSet)

testcoadmin@or-jmprhel-01:~/jmp-testcolab-01/px/px-nginx-test$

testcoadmin@or-jmprhel-01:~/jmp-testcolab-01/px/px-nginx-test$ k get nodes

NAME STATUS ROLES AGE VERSION

dr-apps-01-control-plane-pn8qf Ready control-plane 88m v1.26.10+vmware.1-fips.1

dr-apps-01-workers-zbvzk-7c6f9c7cf6xqkddf-5xd9r Ready <none> 72m v1.26.10+vmware.1-fips.1

dr-apps-01-workers-zbvzk-7c6f9c7cf6xqkddf-tkcn4 Ready <none> 72m v1.26.10+vmware.1-fips.1

dr-apps-01-workers-zbvzk-7c6f9c7cf6xqkddf-vfpjm Ready <none> 72m v1.26.10+vmware.1-fips.1

testcoadmin@or-jmprhel-01:~/jmp-testcolab-01/px/px-nginx-test$ k describe node dr-apps-01-workers-zbvzk-7c6f9c7cf6xqkddf-5xd9r | more

Name: dr-apps-01-workers-zbvzk-7c6f9c7cf6xqkddf-5xd9r

Roles: <none>

Labels: beta.kubernetes.io/arch=amd64

beta.kubernetes.io/os=linux

kubernetes.io/arch=amd64

kubernetes.io/hostname=dr-apps-01-workers-zbvzk-7c6f9c7cf6xqkddf-5xd9r

kubernetes.io/os=linux

px/metadata-node=true

run.tanzu.vmware.com/kubernetesDistributionVersion=v1.26.10\_vmware.1-fips.1-tkg.1.ubuntu

topology.portworx.io/node=dr-apps-01-workers-zbvzk-7c6f9c7cf6xqkddf-5xd9r

Annotations: cluster.x-k8s.io/cluster-name: dr-apps-01

cluster.x-k8s.io/cluster-namespace: dr-01

cluster.x-k8s.io/labels-from-machine:

cluster.x-k8s.io/machine: dr-apps-01-workers-zbvzk-7c6f9c7cf6xqkddf-5xd9r

cluster.x-k8s.io/owner-kind: MachineSet

cluster.x-k8s.io/owner-name: dr-apps-01-workers-zbvzk-7c6f9c7cf6xqkddf

csi.volume.kubernetes.io/nodeid:

{"csi.vsphere.vmware.com":"dr-apps-01-workers-zbvzk-7c6f9c7cf6xqkddf-5xd9r","pxd.portworx.com":"40d7e5d4-e995-40a3-a2a9-d9536a6ab58b"}

kubeadm.alpha.kubernetes.io/cri-socket: unix:///var/run/containerd/containerd.sock

node.alpha.kubernetes.io/ttl: 0

volumes.kubernetes.io/controller-managed-attach-detach: true

CreationTimestamp: Tue, 09 Jan 2024 18:42:01 -0800

Taints: <none>

Unschedulable: false

Lease:

HolderIdentity: dr-apps-01-workers-zbvzk-7c6f9c7cf6xqkddf-5xd9r

AcquireTime: <unset>

RenewTime: Tue, 09 Jan 2024 19:56:26 -0800

Conditions:

Type Status LastHeartbeatTime LastTransitionTime Reason Message

---- ------ ----------------- ------------------ ------ -------

MemoryPressure False Tue, 09 Jan 2024 19:53:34 -0800 Tue, 09 Jan 2024 18:42:01 -0800 KubeletHasSufficientMemory kubelet has sufficient memory available

DiskPressure False Tue, 09 Jan 2024 19:53:34 -0800 Tue, 09 Jan 2024 18:42:01 -0800 KubeletHasNoDiskPressure kubelet has no disk pressure

PIDPressure False Tue, 09 Jan 2024 19:53:34 -0800 Tue, 09 Jan 2024 18:42:01 -0800 KubeletHasSufficientPID kubelet has sufficient PID available

Ready True Tue, 09 Jan 2024 19:53:34 -0800 Tue, 09 Jan 2024 18:43:11 -0800 KubeletReady kubelet is posting ready status. AppArmor enabled

Addresses:

InternalIP: 10.2.6.107

Hostname:

Capacity:

cpu: 2

ephemeral-storage: 19780832Ki

hugepages-1Gi: 0

hugepages-2Mi: 0

memory: 8136084Ki

pods: 110

Allocatable:

cpu: 2

ephemeral-storage: 18230014742

hugepages-1Gi: 0

hugepages-2Mi: 0

memory: 8033684Ki

pods: 110

System Info:

Machine ID: 8be71f9dd053402a8a71df04e6baf4a4

System UUID: f5621542-aee7-280d-39d0-83e4dcba3095

Boot ID: cbf42e82-a284-464d-93d2-16b3a721f1f1

Kernel Version: 5.4.0-166-generic

OS Image: Ubuntu 20.04.6 LTS

Operating System: linux

Architecture: amd64

Container Runtime Version: containerd://1.6.24

Kubelet Version: v1.26.10+vmware.1-fips.1

Kube-Proxy Version: v1.26.10+vmware.1-fips.1

PodCIDR: 192.168.2.0/24

PodCIDRs: 192.168.2.0/24

ProviderID: vsphere://421562f5-e7ae-0d28-39d0-83e4dcba3095

Non-terminated Pods: (14 in total)

Namespace Name CPU Requests CPU Limits Memory Requests Memory Limits Age

--------- ---- ------------ ---------- --------------- ------------- ---

default deploy-rwx-px-nginx-test-5f5867cd44-7gpn6 0 (0%) 0 (0%) 0 (0%) 0 (0%) 12m

default deploy-rwx-px-nginx-test-5f5867cd44-gl5d2 0 (0%) 0 (0%) 0 (0%) 0 (0%) 12m

kube-system antrea-agent-flhmj 400m (20%) 0 (0%) 0 (0%) 0 (0%) 78m

kube-system docker-registry-dr-apps-01-workers-zbvzk-7c6f9c7cf6xqkddf-5xd9r 0 (0%) 0 (0%) 0 (0%) 0 (0%) 77m

kube-system kube-proxy-vq7nd 0 (0%) 0 (0%) 0 (0%) 0 (0%) 78m

portworx portworx-api-t9nfk 0 (0%) 0 (0%) 0 (0%) 0 (0%) 67m

portworx portworx-kvdb-4bfbn 0 (0%) 0 (0%) 0 (0%) 0 (0%) 40m

portworx portworx-operator-579774cc76-tbrqk 0 (0%) 0 (0%) 0 (0%) 0 (0%) 68m

portworx portworx-pvc-controller-754695c57f-g44kq 200m (10%) 0 (0%) 0 (0%) 0 (0%) 67m

portworx px-cluster-a05d75b0-3ec3-44e5-bf96-09ce929ba2ac-2h59j 0 (0%) 0 (0%) 0 (0%) 0 (0%) 67m

portworx px-csi-ext-75d7b9f76d-brc2k 0 (0%) 0 (0%) 0 (0%) 0 (0%) 67m

portworx stork-6d6959f767-4fmd2 100m (5%) 0 (0%) 0 (0%) 0 (0%) 67m

portworx stork-scheduler-5ffb6cdd9d-hf8bj 100m (5%) 0 (0%) 0 (0%) 0 (0%) 67m

vmware-system-csi vsphere-csi-node-hvjj5 0 (0%) 0 (0%) 0 (0%) 0 (0%) 78m

Allocated resources:

(Total limits may be over 100 percent, i.e., overcommitted.)

Resource Requests Limits

-------- -------- ------

cpu 800m (40%) 0 (0%)

memory 0 (0%) 0 (0%)

ephemeral-storage 0 (0%) 0 (0%)

hugepages-1Gi 0 (0%) 0 (0%)

hugepages-2Mi 0 (0%) 0 (0%)

Events:

Type Reason Age From Message

---- ------ ---- ---- -------

Normal RegisteredNode 63m node-controller Node dr-apps-01-workers-zbvzk-7c6f9c7cf6xqkddf-5xd9r event: Registered Node dr-apps-01-workers-zbvzk-7c6f9c7cf6xqkddf-5xd9r in Controller

Normal RegisteredNode 60m node-controller Node dr-apps-01-workers-zbvzk-7c6f9c7cf6xqkddf-5xd9r event: Registered Node dr-apps-01-workers-zbvzk-7c6f9c7cf6xqkddf-5xd9r in Controller

Normal RegisteredNode 58m node-controller Node dr-apps-01-workers-zbvzk-7c6f9c7cf6xqkddf-5xd9r event: Registered Node dr-apps-01-workers-zbvzk-7c6f9c7cf6xqkddf-5xd9r in Controller

Normal RegisteredNode 50m node-controller Node dr-apps-01-workers-zbvzk-7c6f9c7cf6xqkddf-5xd9r event: Registered Node dr-apps-01-workers-zbvzk-7c6f9c7cf6xqkddf-5xd9r in Controller

## kubectl get secret -n dr-01 dr-apps-01-kubeconfig -o jsonpath='{.data.value}' | base64 -d > config\_dr-apps-01

Modify cluster name, user name as needed etc. as described in Initial procedures above

## k create ns portworx; label ns: pod-security privileged audit, enforce, warn; rolebinding to clusterrole psp:vmware-system-privileged

namespace/portworx created

kubectl label --overwrite ns portworx pod-security.kubernetes.io/audit=privileged pod-security.kubernetes.io/enforce=privileged pod-security.kubernetes.io/warn=privileged

kubectl create rolebinding rolebinding-default-privileged-sa-ns\_default --namespace=portworx --clusterrole=psp:vmware-system-privileged --group=system:serviceaccounts

## k create ns central; label ns: pod-security privileged audit, enforce, warn; rolebinding to clusterrole psp:vmware-system-privileged

namespace/central created

kubectl label --overwrite ns central pod-security.kubernetes.io/audit=privileged pod-security.kubernetes.io/enforce=privileged pod-security.kubernetes.io/warn=privileged

kubectl create rolebinding rolebinding-default-privileged-sa-ns\_default --namespace=central --clusterrole=psp:vmware-system-privileged --group=system:serviceaccounts

## kubectl apply -f 'https://install.portworx.com/3.0?comp=pxoperator&kbver=1.26.10&ns=portworx'

## kubectl label [worker nodes] px/metadata-node=true

## kubectl apply -f 'https://install.portworx.com/3.0?operator=true&mc=false&kbver=1.26.10&ns=portworx&b=true&iop=6&csicd=true&mz=8&s=%22sc%3Ddr-tanzu-storage-01%2Csize%3D150%22&j=auto&c=px-cluster-a05d75b0-3ec3-44e5-bf96-09ce929ba2ac&stork=true&csi=true&mon=true&tel=false&st=k8s&e=PRE-EXEC%3Diptables%20-A%20INPUT%20-p%20tcp%20--match%20multiport%20--dports%209001%3A9020%20-j%20ACCEPT%2CPRE-EXEC%3Diptables%20-A%20INPUT%20-p%20tcp%20--match%20multiport%20--dports%202049%20-j%20ACCEPT&promop=true'

## kubectl get pods -n portworx -l name=portworx | grep -v NAME | awk '{print $1}'

px-cluster-a05d75b0-3ec3-44e5-bf96-09ce929ba2ac-2h59j

px-cluster-a05d75b0-3ec3-44e5-bf96-09ce929ba2ac-2j8lm

px-cluster-a05d75b0-3ec3-44e5-bf96-09ce929ba2ac-574k4

kubectl exec -t -n portworx px-cluster-a05d75b0-3ec3-44e5-bf96-09ce929ba2ac-2h59j -- nsenter --mount=/host\_proc/1/ns/mnt bash -c "iptables -A INPUT -p tcp --match multiport --dports 9001:9020 -j ACCEPT"

Defaulted container "portworx" out of: portworx, csi-node-driver-registrar

kubectl exec -t -n portworx px-cluster-a05d75b0-3ec3-44e5-bf96-09ce929ba2ac-2h59j -- nsenter --mount=/host\_proc/1/ns/mnt bash -c "iptables -A INPUT -p tcp --match multiport --dports 2049 -j ACCEPT"

Defaulted container "portworx" out of: portworx, csi-node-driver-registrar

kubectl exec -t -n portworx px-cluster-a05d75b0-3ec3-44e5-bf96-09ce929ba2ac-2j8lm -- nsenter --mount=/host\_proc/1/ns/mnt bash -c "iptables -A INPUT -p tcp --match multiport --dports 9001:9020 -j ACCEPT"

Defaulted container "portworx" out of: portworx, csi-node-driver-registrar

kubectl exec -t -n portworx px-cluster-a05d75b0-3ec3-44e5-bf96-09ce929ba2ac-2j8lm -- nsenter --mount=/host\_proc/1/ns/mnt bash -c "iptables -A INPUT -p tcp --match multiport --dports 2049 -j ACCEPT"

Defaulted container "portworx" out of: portworx, csi-node-driver-registrar

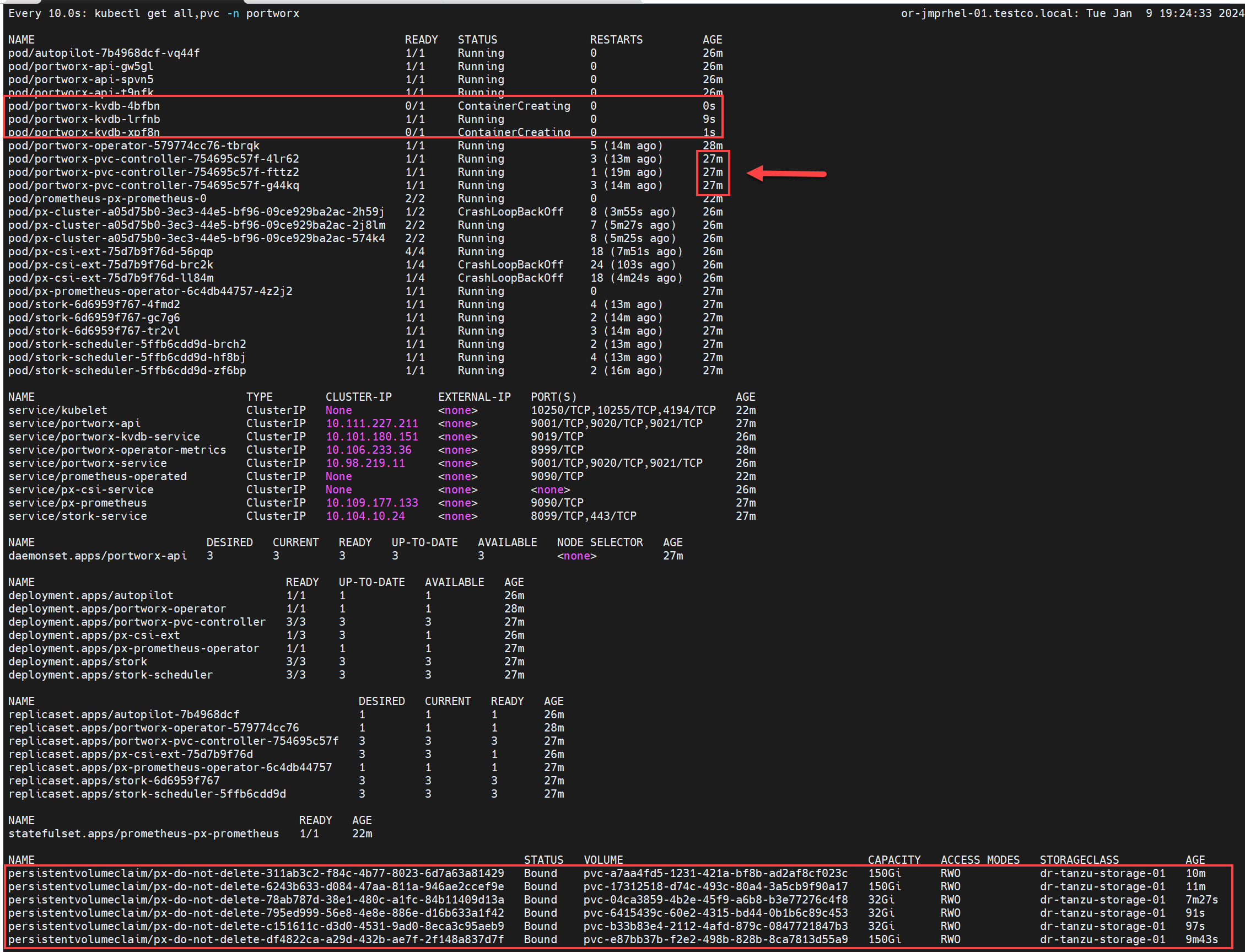
kubectl exec -t -n portworx px-cluster-a05d75b0-3ec3-44e5-bf96-09ce929ba2ac-574k4 -- nsenter --mount=/host\_proc/1/ns/mnt bash -c "iptables -A INPUT -p tcp --match multiport --dports 9001:9020 -j ACCEPT"

Defaulted container "portworx" out of: portworx, csi-node-driver-registrar

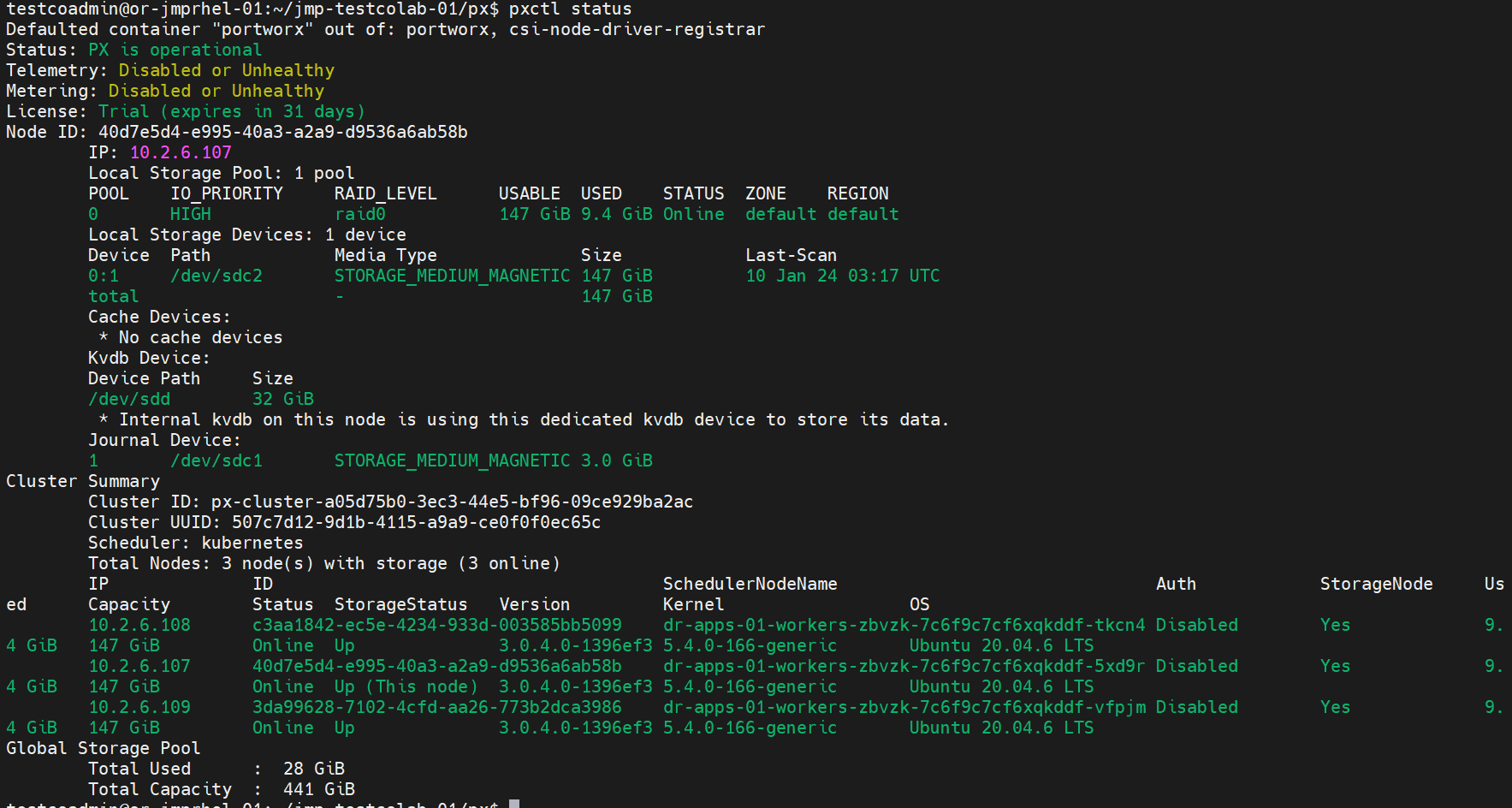
kubectl exec -t -n portworx px-cluster-a05d75b0-3ec3-44e5-bf96-09ce929ba2ac-574k4 -- nsenter --mount=/host\_proc/1/ns/mnt bash -c "iptables -A INPUT -p tcp --match multiport --dports 2049 -j ACCEPT"

Defaulted container "portworx" out of: portworx, csi-node-driver-registrar

## Portworx pvcs, kvdb pods provisioning



# Pxctl status



# 1/9/2024 PX provisioning test/cfm - RWO

## Prep default namespace

kubectl label --overwrite ns default pod-security.kubernetes.io/audit=privileged pod-security.kubernetes.io/enforce=privileged pod-security.kubernetes.io/warn=privileged

kubectl create rolebinding rolebinding-default-privileged-sa-ns\_default --namespace=default --clusterrole=psp:vmware-system-privileged --group=system:serviceaccounts

## pxd-repl1-portworx-sc.yaml

kind: StorageClass

apiVersion: storage.k8s.io/v1

metadata:

name: pxd-repl1-portworx-sc

provisioner: pxd.portworx.com

parameters:

repl: "1"

## pvc-repl1-01.yaml

kind: PersistentVolumeClaim

apiVersion: v1

metadata:

name: pvc-repl1-01

namespace: default

spec:

storageClassName: pxd-repl1-portworx-sc

accessModes:

- ReadWriteOnce

resources:

requests:

storage: 2Gi

## sc, pvc created, bound

NAME STATUS VOLUME CAPACITY ACCESS MODES STORAGECLASS AGE

pvc-repl1-01 Bound pvc-75f65ce3-e626-456c-9e8d-39f905558e01 2Gi RWO pxd-repl1-portworx-sc 2s

## deploy-rwo-px-nginx-test.yaml

apiVersion: apps/v1

kind: Deployment

metadata:

name: deploy-rwo-px-nginx-test

namespace: default

labels:

app: rwo-px-nginx-test

spec:

replicas: 1

selector:

matchLabels:

app: rwo-px-nginx-test

template:

metadata:

labels:

app: rwo-px-nginx-test

spec:

containers:

- name: px-nginx

image: nginx

ports:

- containerPort: 80

name: "http-server"

volumeMounts:

- mountPath: "/usr/share/nginx"

name: vol-px-01

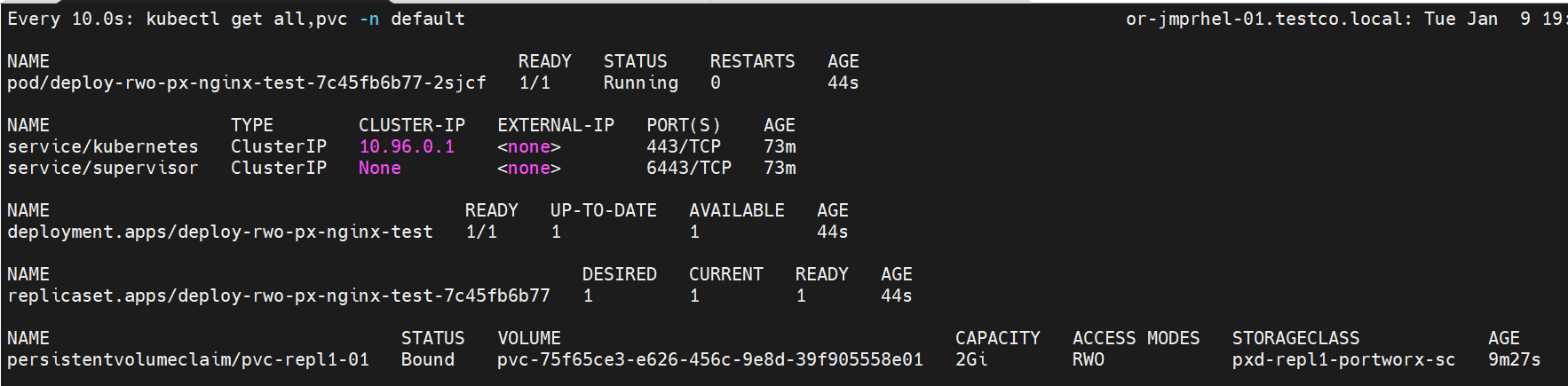
volumes:

- name: vol-px-01

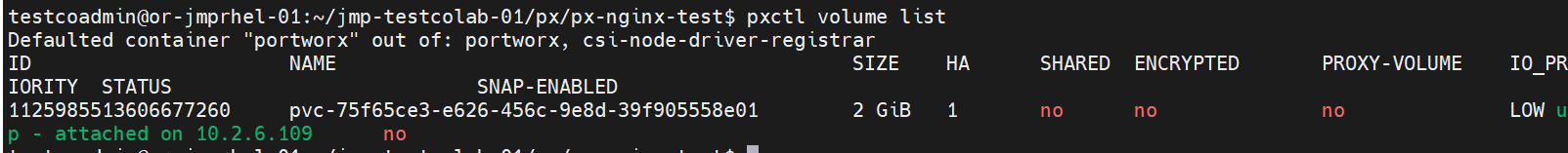
persistentVolumeClaim:

claimName: pvc-repl1-01

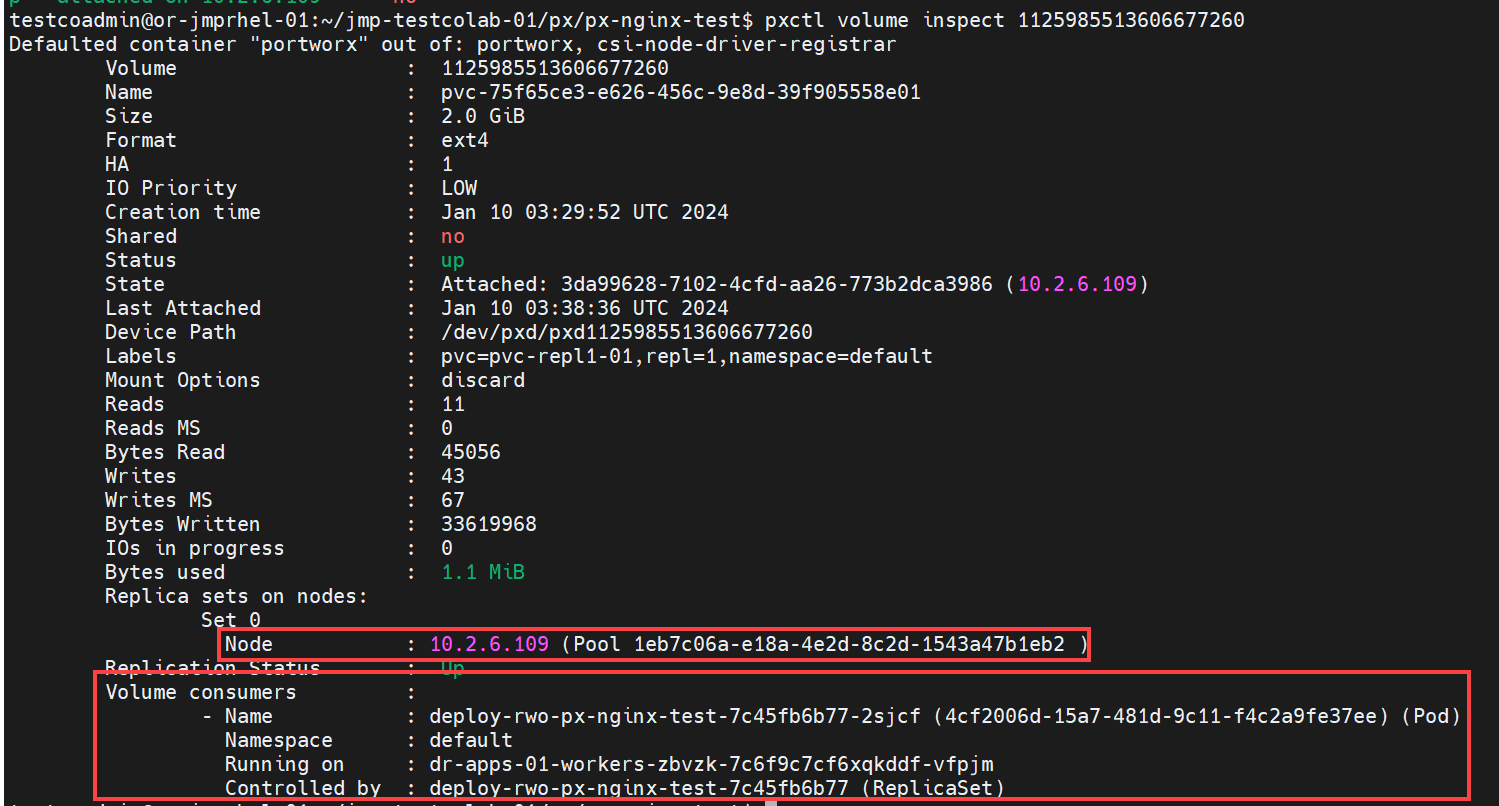
## kubectl get all,pvc -n default



## Pxctl volume list



## Pxctl volume inspect



# 1/9/2024 PX provisioning test/cfm - RWX

## Prep default namespace

kubectl label --overwrite ns default pod-security.kubernetes.io/audit=privileged pod-security.kubernetes.io/enforce=privileged pod-security.kubernetes.io/warn=privileged

kubectl create rolebinding rolebinding-default-privileged-sa-ns\_default --namespace=default --clusterrole=psp:vmware-system-privileged --group=system:serviceaccounts

## pxd-repl1-portworx-sc.yaml

kind: StorageClass

apiVersion: storage.k8s.io/v1

metadata:

name: pxd-repl1-portworx-sc

provisioner: pxd.portworx.com

parameters:

repl: "1"

## pvc-repl3-01.yaml

kind: PersistentVolumeClaim

apiVersion: v1

metadata:

name: pvc-repl1-03

namespace: default

spec:

storageClassName: pxd-repl1-portworx-sc

accessModes:

- ReadWriteMany

resources:

requests:

storage: 2Gi

## sc, pvc created, bound

NAME STATUS VOLUME CAPACITY ACCESS MODES STORAGECLASS AGE

pvc-repl1-01 Bound pvc-75f65ce3-e626-456c-9e8d-39f905558e01 2Gi RWO pxd-repl1-portworx-sc 14m

pvc-repl3-01 Bound pvc-6679d209-30d1-4134-87e1-552c15056cfa 2Gi RWX pxd-repl1-portworx-sc 2s

## deploy-rwx-px-nginx-test.yaml

apiVersion: apps/v1

kind: Deployment

metadata:

name: deploy-rwx-px-nginx-test

namespace: default

labels:i

app: rwx-px-nginx-test

spec:

replicas: 3

selector:

matchLabels:

app: rwx-px-nginx-test

template:

metadata:

labels:

app: rwx-px-nginx-test

spec:

containers:

- name: px-nginx

image: nginx

ports:

- containerPort: 80

name: "http-server"

volumeMounts:

- mountPath: "/usr/share/nginx"

name: vol-px-01

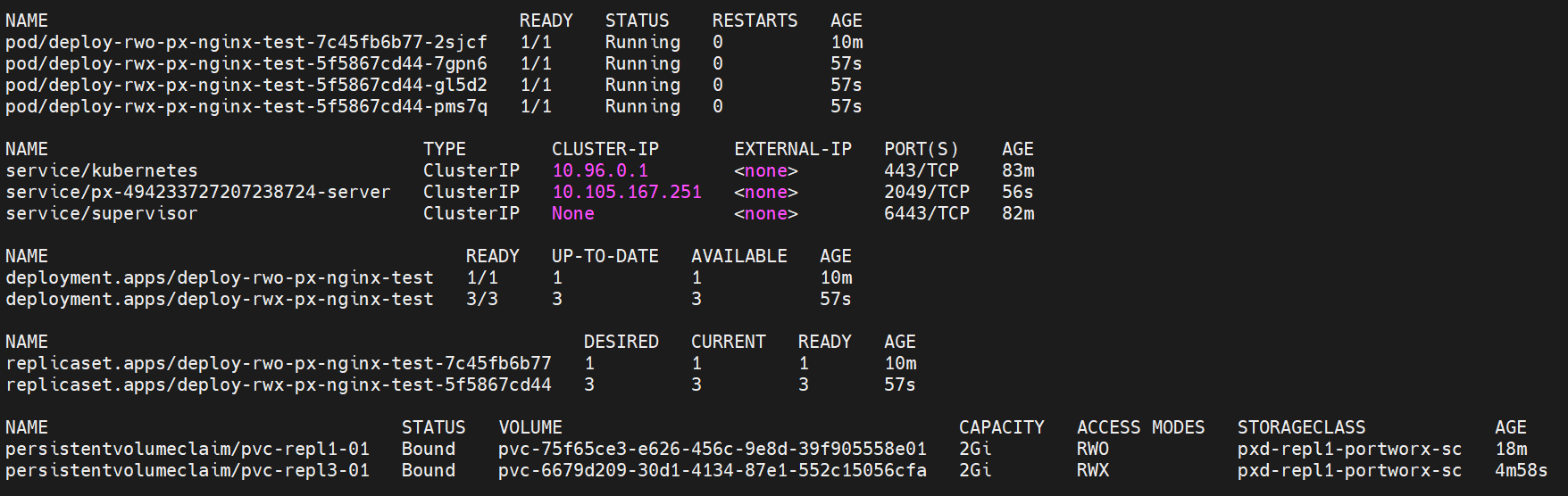
volumes:

- name: vol-px-01

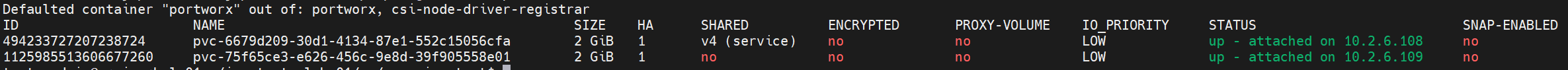
persistentVolumeClaim:

claimName: pvc-repl1-01

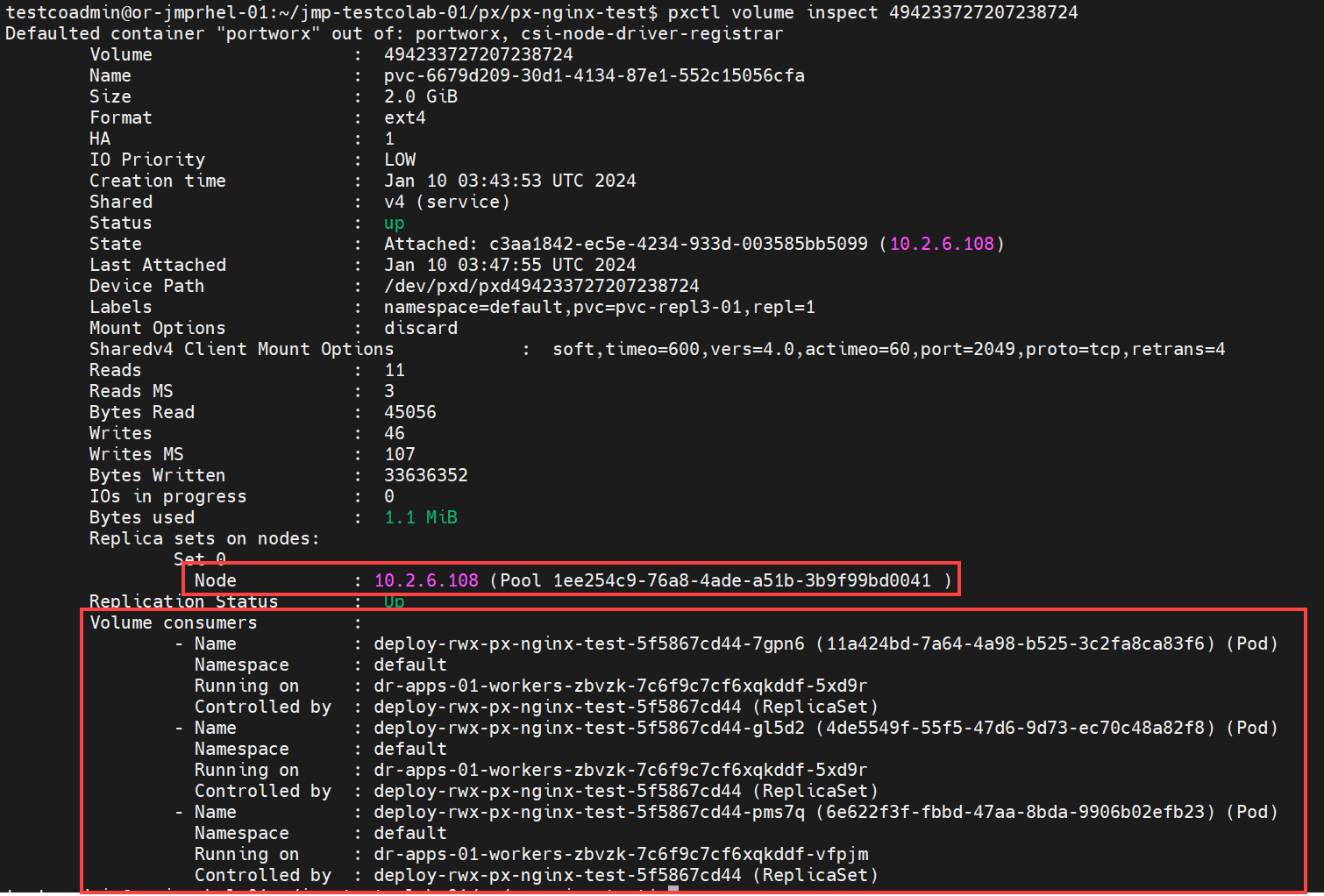
## kubectl get all,pvc -n default



## Pxctl volume list



## Pxctl volume inspect



## Exec into each rwx pod, write data, review other pods’ written data

k exec -it deploy-rwx-px-nginx-test-5f5867cd44-7gpn6 -- /bin/bash

root@deploy-rwx-px-nginx-test-5f5867cd44-7gpn6:/# cd /usr/share/nginx

root@deploy-rwx-px-nginx-test-5f5867cd44-7gpn6:/usr/share/nginx# echo "test from 7gpn6" >> rwxtest.txt

root@deploy-rwx-px-nginx-test-5f5867cd44-7gpn6:/usr/share/nginx# cat rwxtest.txt

test from 7gpn6

testcoadmin@or-jmprhel-01:~/jmp-testcolab-01/px$ k exec -it deploy-rwx-px-nginx-test-5f5867cd44-gl5d2 -- /bin/bash

root@deploy-rwx-px-nginx-test-5f5867cd44-gl5d2:/# cd /usr/share/nginx

root@deploy-rwx-px-nginx-test-5f5867cd44-gl5d2:/usr/share/nginx# echo "test from gl5d2" >> rwxtest.txt

root@deploy-rwx-px-nginx-test-5f5867cd44-gl5d2:/usr/share/nginx# cat rwxtest.txt

test from 7gpn6

test from gl5d2

testcoadmin@or-jmprhel-01:~/jmp-testcolab-01/px$ k exec -it deploy-rwx-px-nginx-test-5f5867cd44-pms7q -- /bin/bash

root@deploy-rwx-px-nginx-test-5f5867cd44-pms7q:/# cd /usr/share/nginx

root@deploy-rwx-px-nginx-test-5f5867cd44-pms7q:/usr/share/nginx# echo "test from pms7q" >> rwxtest.txt

root@deploy-rwx-px-nginx-test-5f5867cd44-pms7q:/usr/share/nginx# cat rwxtest.txt

test from 7gpn6

test from gl5d2

test from pms7q