TestCoLab

postgresql-ha-px-enterprise-aks-deployment-proc-rvw-01.docx

Updated: 1/8/2024

[Procedures Overview 4](#_Toc155599269)

[Create pxd-repl3-sc StorageClass 5](#_Toc155599270)

[pxd-repl3-sc.yaml 5](#_Toc155599271)

[Create pvc-postgres-pxd PersistentVolumeClaim 6](#_Toc155599272)

[pvc-postgres-pxd.yaml 6](#_Toc155599273)

[Create postgres-pass Secret 7](#_Toc155599274)

[Create postgres-01 Deployment 8](#_Toc155599275)

[deploy-postgres-01.yaml 8](#_Toc155599276)

[kubectl get all,pvc -n default – confirm pods running 10](#_Toc155599277)

[pxctl volume inspect 838655065192126723- – confirm (3) replicas attached on (3) nodes 11](#_Toc155599278)

[HA : 3 11](#_Toc155599279)

[Node : 10.224.1.133 (Pool 034b1b29-5bd2-4d73-9a8b-251daf80e48e ) 11](#_Toc155599280)

[Node : 10.224.1.242 (Pool 89b4574a-00f7-4792-96cf-c03532707566 ) 11](#_Toc155599281)

[Node : 10.224.1.24 (Pool ce76bf4a-5dcc-4f34-8db6-ae902d0bd277 ) 11](#_Toc155599282)

[- Name : postgres-01-785b698799-7bcp8 (1662363c-6ccf-4f7e-a00f-7be6686eb901) (Pod) 11](#_Toc155599283)

[Connect to PostgreSQL pod, create database of large size 12](#_Toc155599284)

[k get po 12](#_Toc155599285)

[k exec -it postgres-01-785b698799-7gx8m bash 12](#_Toc155599286)

[root@postgres-01-785b698799-7gx8m:/# psql 12](#_Toc155599287)

[pgbench=# create database pxdemo; 12](#_Toc155599288)

[pgbench=# \l 12](#_Toc155599289)

[List of databases 12](#_Toc155599290)

[pgbench=# \q 12](#_Toc155599291)

[root@postgres-01-785b698799-7bcp8:/# pgbench -i -s 50 pxdemo; 12](#_Toc155599292)

[root@postgres-01-785b698799-7bcp8:/# psql pxdemo 13](#_Toc155599293)

[pxdemo=# \dt 13](#_Toc155599294)

[pxdemo=# select count(\*) from pgbench\_accounts; 13](#_Toc155599295)

[5000000 13](#_Toc155599296)

[pxctl volume inspect 838655065192126723- – confirm increased volume size, note which node single pod is running on 14](#_Toc155599297)

[Bytes used : 1.4 GiB 14](#_Toc155599298)

[Running on : aks-pxstore01-17862373-vmss000000 15](#_Toc155599299)

[Cordon single node running PostgreSQL pod 16](#_Toc155599300)

[kubectl cordon aks-pxstore01-17862373-vmss000000 16](#_Toc155599301)

[Confirm node cordoned 16](#_Toc155599302)

[Delete single PostgreSQL pod from cordoned node, confirm relocated to uncordoned node with volume mounted 17](#_Toc155599303)

[k delete po -n default postgres-01-785b698799-7bcp8 17](#_Toc155599304)

[k describe po postgres-01-785b698799-xg9t7 - Confirm single PostgreSQL is now running on uncordoned storage node and volume is mounted 17](#_Toc155599305)

[Node: aks-pxstore01-17862373-vmss000001/10.224.1.133 17](#_Toc155599306)

[Volumes: 18](#_Toc155599307)

[postgredb: 18](#_Toc155599308)

[Type: PersistentVolumeClaim (a reference to a PersistentVolumeClaim in the same namespace) 18](#_Toc155599309)

[ClaimName: pvc-postgres-pxd 18](#_Toc155599310)

[pxctl volume inspect 838655065192126723 – confirm relocated single PostgreSQL pod is shown as a volume consumer from the uncordoned storage node 19](#_Toc155599311)

[- Name : postgres-01-785b698799-xg9t7 (3dd8f7a6-7a54-4618-b668-9458197ce861) (Pod) 19](#_Toc155599312)

[Namespace : default 19](#_Toc155599313)

[Running on : aks-pxstore01-17862373-vmss000001 19](#_Toc155599314)

[Connect to relocated single PostgreSQL pod, confirm pxdemo is still present on replicated volume with 5000000 data rows 20](#_Toc155599315)

[k get po 20](#_Toc155599316)

[k exec -it postgres-01-785b698799-xg9t7 bash 20](#_Toc155599317)

[root@ postgres-01-785b698799-xg9t7:/# psql pxdemo 20](#_Toc155599318)

[pxdemo=# select count(\*) from pgbench\_accounts; 20](#_Toc155599319)

[5000000 20](#_Toc155599320)

[PostgreSQL Cheat Sheet 21](#_Toc155599321)

[\l – list databases 21](#_Toc155599322)

[CREATE DATABASE [dB name]; 21](#_Toc155599323)

[\q quit 21](#_Toc155599324)

[Logs 22](#_Toc155599325)

[Troubleshooting 23](#_Toc155599326)

[Resources 24](#_Toc155599327)

[How to Run HA PostgreSQL on AKS (Azure Kubernetes Service) 24](#_Toc155599328)

[PostgreSQL Cheat Sheet 24](#_Toc155599329)

# Procedures Overview

* Create pxd-repl3-sc StorageClass
* Create pvc-postgres-pxd PersistentVolumeClaim
* Create postgres-pass Secret
* Create postgres-01 Deployment
* kubectl get all,pvc -n default – confirm pods running
* pxctl volume inspect [id] – confirm (3) replicas attached on (3) nodes
* Connect to PostgreSQL pod, create database of large size
* pxctl volume inspect 838655065192126723- – confirm increased volume size, note which node single pod is running on
* Cordon single node running PostgreSQL pod
* Delete single PostgreSQL pod from cordoned node, confirm relocated to uncordoned node with volume mounted
* Connect to relocated single PostgreSQL pod, confirm pxdemo is still present on replicated volume with 5000000 data rows

# Create pxd-repl3-sc StorageClass

## pxd-repl3-sc.yaml

kind: StorageClass

apiVersion: storage.k8s.io/v1

metadata:

name: pxd-repl3-sc

provisioner: pxd.portworx.com

parameters:

repl: "3"

# Create pvc-postgres-pxd PersistentVolumeClaim

## pvc-postgres-pxd.yaml

kind: PersistentVolumeClaim

apiVersion: v1

metadata:

name: pvc-postgres-pxd

namespace: default

spec:

storageClassName: pxd-repl3-sc

accessModes:

- ReadWriteOnce

resources:

requests:

storage: 3Gi

# Create postgres-pass Secret

$ echo postgres123 > password.txt

$ tr --delete '\n' <password.txt >.strippedpassword.txt && mv .strippedpassword.txt password.txt

$ kubectl create secret generic postgres-pass --from-file=password.txt

# Create postgres-01 Deployment

## deploy-postgres-01.yaml

apiVersion: apps/v1

kind: Deployment

metadata:

name: postgres-01

spec:

strategy:

rollingUpdate:

maxSurge: 1

maxUnavailable: 1

type: RollingUpdate

replicas: 1

selector:

matchLabels:

app: postgres

template:

metadata:

labels:

app: postgres

spec:

containers:

- name: postgres

image: postgres:14.10

imagePullPolicy: "Always"

ports:

- containerPort: 5432

env:

- name: POSTGRES\_USER

value: pgbench

- name: PGUSER

value: pgbench

- name: POSTGRES\_PASSWORD

valueFrom:

secretKeyRef:

name: postgres-pass

key: password.txt

- name: PGBENCH\_PASSWORD

value: superpostgres

- name: PGDATA

value: /var/lib/postgresql/data/pgdata

volumeMounts:

- mountPath: /var/lib/postgresql/data

name: postgredb

volumes:

- name: postgredb

persistentVolumeClaim:

claimName: pvc-postgres-pxd

# kubectl get all,pvc -n default – confirm pods running

NAME READY STATUS RESTARTS AGE

pod/postgres-01-785b698799-7gx8m 1/1 Running 0 62s

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

service/kubernetes ClusterIP 10.0.0.1 <none> 443/TCP 24h

NAME READY UP-TO-DATE AVAILABLE AGE

deployment.apps/postgres-01 1/1 1 1 62s

NAME DESIRED CURRENT READY AGE

replicaset.apps/postgres-01-785b698799 1 1 1 62s

NAME STATUS VOLUME CAPACITY ACCESS MODES STORAGECLASS AGE

persistentvolumeclaim/pvc-postgres-pxd Bound pvc-7f749b57-bb64-4e03-8a65-4ee2a0bee403 1Gi RWO pxd-repl3-sc 5m15s

# pxctl volume inspect 838655065192126723- – confirm (3) replicas attached on (3) nodes

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

Volume : 838655065192126723

Name : pvc-6fd4b13c-f40f-437c-bc17-2bf733811e4b

Size : 3.0 GiB

Format : ext4

## HA : 3

IO Priority : LOW

Creation time : Jan 4 04:06:14 UTC 2024

Shared : no

Status : up

State : Attached: 497f95d0-ef9d-4077-8424-5ed0cc06ccf5 (10.224.1.242)

Last Attached : Jan 4 04:06:32 UTC 2024

Device Path : /dev/pxd/pxd838655065192126723

Labels : namespace=default,pvc=pvc-postgres-pxd,repl=3

Mount Options : discard

Reads : 48

Reads MS : 44

Bytes Read : 401408

Writes : 2488

Writes MS : 20292

Bytes Written : 104087552

IOs in progress : 0

Bytes used : 43 MiB

Replica sets on nodes:

Set 0

## Node : 10.224.1.133 (Pool 034b1b29-5bd2-4d73-9a8b-251daf80e48e )

## Node : 10.224.1.242 (Pool 89b4574a-00f7-4792-96cf-c03532707566 )

## Node : 10.224.1.24 (Pool ce76bf4a-5dcc-4f34-8db6-ae902d0bd277 )

Replication Status : Up

Volume consumers :

## - Name : postgres-01-785b698799-7bcp8 (1662363c-6ccf-4f7e-a00f-7be6686eb901) (Pod)

Namespace : default

Running on : aks-pxstore01-17862373-vmss000000

Controlled by : postgres-01-785b698799 (ReplicaSet)

# Connect to PostgreSQL pod, create database of large size

## k get po

NAME READY STATUS RESTARTS AGE

postgres-01-785b698799-7gx8m 1/1 Running 0 13m

## k exec -it postgres-01-785b698799-7gx8m bash

kubectl exec [POD] [COMMAND] is DEPRECATED and will be removed in a future version. Use kubectl exec [POD] -- [COMMAND] instead.

## root@postgres-01-785b698799-7gx8m:/# psql

## pgbench=# create database pxdemo;

CREATE DATABASE

## pgbench=# \l

## List of databases

Name | Owner | Encoding | Collate | Ctype | Access privileges

-----------+---------+----------+------------+------------+---------------------

pgbench | pgbench | UTF8 | en\_US.utf8 | en\_US.utf8 |

postgres | pgbench | UTF8 | en\_US.utf8 | en\_US.utf8 |

pxdemo | pgbench | UTF8 | en\_US.utf8 | en\_US.utf8 |

template0 | pgbench | UTF8 | en\_US.utf8 | en\_US.utf8 | =c/pgbench +

| | | | | pgbench=CTc/pgbench

template1 | pgbench | UTF8 | en\_US.utf8 | en\_US.utf8 | =c/pgbench +

| | | | | pgbench=CTc/pgbench

(5 rows)

## pgbench=# \q

## root@postgres-01-785b698799-7bcp8:/# pgbench -i -s 50 pxdemo;

**Note**: By default, pgbench will create 4 tables (pgbench\_branches, pgbench\_tellers, pgbench\_accounts, and pgbench\_history) with 100,000 rows in the main pgbench\_accounts table. This creates a simple 16MB database.

The -s option is used for multiplying the number of rows entered into each table. In the command below, we enter a “scaling” option of 50. This tells pgbench to create a database with 50 times the default size.

What this means is our pgbench\_accounts table now has 5,000,000 records. It also means our database size is now 800MB (50 x 16MB).

dropping old tables...

NOTICE: table "pgbench\_accounts" does not exist, skipping

NOTICE: table "pgbench\_branches" does not exist, skipping

NOTICE: table "pgbench\_history" does not exist, skipping

NOTICE: table "pgbench\_tellers" does not exist, skipping

creating tables...

generating data (client-side)...

5000000 of 5000000 tuples (100%) done (elapsed 20.59 s, remaining 0.00 s)

vacuuming...

creating primary keys...

done in 36.79 s (drop tables 0.00 s, create tables 0.01 s, client-side generate 20.67 s, vacuum 1.56 s, primary keys 14.56 s).

## root@postgres-01-785b698799-7bcp8:/# psql pxdemo

psql (14.10 (Debian 14.10-1.pgdg120+1))

Type "help" for help.

## pxdemo=# \dt

List of relations

Schema | Name | Type | Owner

--------+------------------+-------+---------

public | pgbench\_accounts | table | pgbench

public | pgbench\_branches | table | pgbench

public | pgbench\_history | table | pgbench

public | pgbench\_tellers | table | pgbench

(4 rows)

## pxdemo=# select count(\*) from pgbench\_accounts;

count

---------

## 5000000

(1 row)

# pxctl volume inspect 838655065192126723- – confirm increased volume size, note which node single pod is running on

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

Volume : 838655065192126723

Name : pvc-6fd4b13c-f40f-437c-bc17-2bf733811e4b

Size : 3.0 GiB

Format : ext4

HA : 3

IO Priority : LOW

Creation time : Jan 4 04:06:14 UTC 2024

Shared : no

Status : up

State : Attached: 497f95d0-ef9d-4077-8424-5ed0cc06ccf5 (10.224.1.242)

Last Attached : Jan 4 04:06:32 UTC 2024

Device Path : /dev/pxd/pxd838655065192126723

Labels : namespace=default,pvc=pvc-postgres-pxd,repl=3

Mount Options : discard

Reads : 101

Reads MS : 461

Bytes Read : 1355776

Writes : 6352

Writes MS : 434141

Bytes Written : 2200809472

IOs in progress : 0

## Bytes used : 1.4 GiB

Replica sets on nodes:

Set 0

Node : 10.224.1.133 (Pool 034b1b29-5bd2-4d73-9a8b-251daf80e48e )

Node : 10.224.1.242 (Pool 89b4574a-00f7-4792-96cf-c03532707566 )

Node : 10.224.1.24 (Pool ce76bf4a-5dcc-4f34-8db6-ae902d0bd277 )

Replication Status : Up

Volume consumers :

- Name : postgres-01-785b698799-7bcp8 (1662363c-6ccf-4f7e-a00f-7be6686eb901) (Pod)

Namespace : default

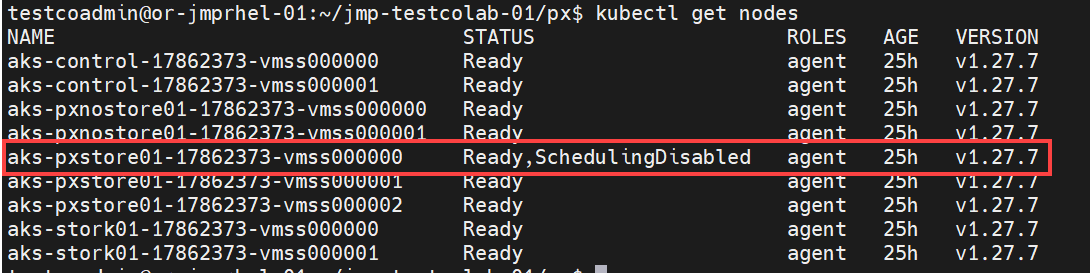
## Running on : aks-pxstore01-17862373-vmss000000

Controlled by : postgres-01-785b698799 (ReplicaSet)

# Cordon single node running PostgreSQL pod

## kubectl cordon aks-pxstore01-17862373-vmss000000

## Confirm node cordoned



# Delete single PostgreSQL pod from cordoned node, confirm relocated to uncordoned node with volume mounted

## k delete po -n default postgres-01-785b698799-7bcp8

pod "postgres-01-785b698799-7bcp8" deleted

## k describe po postgres-01-785b698799-xg9t7 - Confirm single PostgreSQL is now running on uncordoned storage node and volume is mounted

Name: postgres-01-785b698799-xg9t7

Namespace: default

Priority: 0

Service Account: default

## Node: aks-pxstore01-17862373-vmss000001/10.224.1.133

Start Time: Wed, 03 Jan 2024 20:24:45 -0800

Labels: app=postgres

pod-template-hash=785b698799

Annotations: <none>

Status: Running

IP: 10.224.1.164

IPs:

IP: 10.224.1.164

Controlled By: ReplicaSet/postgres-01-785b698799

Containers:

postgres:

Container ID: containerd://72ab2f0b23220dfe81139f7950c54b06d218d4e49b8d5242bb54411fca03e2ba

Image: postgres:14.10

Image ID: docker.io/library/postgres@sha256:972139559e71e48e8b0c0d237933c59a52fecad5491eaa11aa7f692321fabcc8

Port: 5432/TCP

Host Port: 0/TCP

State: Running

Started: Wed, 03 Jan 2024 20:24:55 -0800

Ready: True

Restart Count: 0

Environment:

POSTGRES\_USER: pgbench

PGUSER: pgbench

POSTGRES\_PASSWORD: <set to the key 'password.txt' in secret 'postgres-pass'> Optional: false

PGBENCH\_PASSWORD: superpostgres

PGDATA: /var/lib/postgresql/data/pgdata

Mounts:

/var/lib/postgresql/data from postgredb (rw)

/var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-t57b4 (ro)

Conditions:

Type Status

Initialized True

Ready True

ContainersReady True

PodScheduled True

## Volumes:

## postgredb:

## Type: PersistentVolumeClaim (a reference to a PersistentVolumeClaim in the same namespace)

## ClaimName: pvc-postgres-pxd

ReadOnly: false

kube-api-access-t57b4:

Type: Projected (a volume that contains injected data from multiple sources)

TokenExpirationSeconds: 3607

ConfigMapName: kube-root-ca.crt

ConfigMapOptional: <nil>

DownwardAPI: true

## pxctl volume inspect 838655065192126723 – confirm relocated single PostgreSQL pod is shown as a volume consumer from the uncordoned storage node

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

Volume : 838655065192126723

Name : pvc-6fd4b13c-f40f-437c-bc17-2bf733811e4b

Size : 3.0 GiB

Format : ext4

HA : 3

IO Priority : LOW

Creation time : Jan 4 04:06:14 UTC 2024

Shared : no

Status : up

State : Attached: ae2beade-c471-4dd7-9cac-3a90412984ae (10.224.1.133)

Last Attached : Jan 4 04:24:46 UTC 2024

Device Path : /dev/pxd/pxd838655065192126723

Labels : repl=3,namespace=default,pvc=pvc-postgres-pxd

Mount Options : discard

Reads : 200

Reads MS : 60

Bytes Read : 3497984

Writes : 244

Writes MS : 375

Bytes Written : 2461696

IOs in progress : 0

Bytes used : 1.4 GiB

Replica sets on nodes:

Set 0

Node : 10.224.1.133 (Pool 034b1b29-5bd2-4d73-9a8b-251daf80e48e )

Node : 10.224.1.242 (Pool 89b4574a-00f7-4792-96cf-c03532707566 )

Node : 10.224.1.24 (Pool ce76bf4a-5dcc-4f34-8db6-ae902d0bd277 )

Replication Status : Up

Volume consumers :

## - Name : postgres-01-785b698799-xg9t7 (3dd8f7a6-7a54-4618-b668-9458197ce861) (Pod)

## Namespace : default

## Running on : aks-pxstore01-17862373-vmss000001

Controlled by : postgres-01-785b698799 (ReplicaSet)

# Connect to relocated single PostgreSQL pod, confirm pxdemo is still present on replicated volume with 5000000 data rows

## k get po

NAME READY STATUS RESTARTS AGE

postgres-01-785b698799-xg9t7 1/1 Running 0 13m

## k exec -it postgres-01-785b698799-xg9t7 bash

kubectl exec [POD] [COMMAND] is DEPRECATED and will be removed in a future version. Use kubectl exec [POD] -- [COMMAND] instead.

## root@ postgres-01-785b698799-xg9t7:/# psql pxdemo

psql (14.10 (Debian 14.10-1.pgdg120+1))

Type "help" for help.

## pxdemo=# select count(\*) from pgbench\_accounts;

count

---------

## 5000000

(1 row)

# PostgreSQL Cheat Sheet

## \l – list databases

## CREATE DATABASE [dB name];

## \q quit

# Logs

# Troubleshooting

# Resources

## How to Run HA PostgreSQL on AKS (Azure Kubernetes Service)

https://portworx.com/blog/ha-postgresql-azure-aks/

## PostgreSQL Cheat Sheet

https://timescale.com/learn/postgres-cheat-sheet/databases