IBM® Storage

Hybrid Multicloud Business Continuity for OpenShift Workloads with IBM Spectrum Virtualize in AWS

IBM Storage Team

Contents

About this document

This publication is intended to facilitate the deployment of the hybrid cloud business continuity solution with Red Hat OpenShift Container Platform and IBM® block CSI (Container Storage Interface) driver plug-in for IBM Spectrum® Virtualize on Public Cloud AWS (Amazon Web Services). This solution is designed to protect the data by using IBM Storage-based Global Mirror replication.

For demonstration purposes, MySQL containerized database is installed on the on-premises IBM FlashSystem® that is connected to the Red Hat OpenShift Container Platform (OCP) cluster in the vSphere environment through the IBM block CSI driver. The volume (LUN) on IBM FlashSystem storage system is replicated by using global mirror on IBM Spectrum Virtualize for Public Cloud on AWS. Red Hat OpenShift cluster (OCP cluster) and the IBM block CSI driver plug-in are installed on AWS by using Installer-Provisioned Infrastructure (IPI) methodology.

The information in this document is distributed on an as-is basis without any warranty that is either expressed or implied. Support assistance for the use of this material is limited to situations where IBM Spectrum Virtualize for Public Cloud is supported and entitled, and where the issues are specific to this Blueprint implementation.

Executive summary

In today's environment, many organizations use some form of cloud services, whether private, public, or hybrid multicloud. Storage infrastructure is a part of these services and deployments.

For the Red Hat OpenShift environment that is deployed on AWS Cloud, the quick start includes AWS CloudFormation templates that build the AWS infrastructure by using AWS best practices, and then passes that environment to Ansible playbooks to build the OpenShift environment. The AWS CloudFormation templates use AWS Lambda to generate a dynamic SSH key pair that is loaded into an Auto Scaling group. The Ansible inventory file is auto generated. The combination of AWS CloudFormation and Ansible enables you to deploy and tear down your OpenShift environment by using CloudFormation stacks.

IBM released its open source CSI driver, which allows dynamic provisioning storage for containers on Kubernetes and Red Hat OpenShift container platform on AWS. IBM Spectrum Storage™ family and IBM Spectrum Virtualize for Public Cloud (SVPC AWS) supports clients in their IT architectural transformation and migration towards the cloud service model. This transformation enables hybrid cloud strategies while maintaining the benefits and advanced functions of sophisticated storage systems.

With IBM Spectrum Virtualize and Spectrum Virtualize For Public Cloud on AWS, organizations can have multicloud environments with data replication between the following components:

- On-premises or private cloud to public cloud (AWS Cloud)
- Two public clouds (AWS Cloud)

  IBM Spectrum Virtualize for Public Cloud enables data on heterogeneous storage systems to be replicated or migrated between on-premises and IBM Cloud® or AWS.

  IBM Spectrum Virtualize and IBM Spectrum Virtualize for Public Cloud together support mirroring between on-premises and cloud data centers or between cloud data centers.

  These functions can be used to:

  - Migrate data between on-premises and public cloud data centers or between public cloud data centers. Data management is consistent between on-premises storage and the public cloud.
  - Implement disaster recovery strategies between on-premises and public cloud data centers.
  - Enable cloud-based DevOps with easy replication of data from on-premises sources.

Support for the Blueprint and its configurations

Support for the underlying components that make up this solution are provided by way of the standard procedures and processes that are available for each of those components, as governed by the support entitlement that is available for those components.

For more information about these components, see “Prerequisites” on page 3.

Requesting assistance

All components of the solutions are part of this unified support structure. Support assistance of the solution that is described in this Blueprint is available by requesting assistance for any of the components in the solution and is the preferred method.

Scope of this document

This Blueprint provides a solutions architecture and related configuration tasks. The solution relies on the following software components and related documents:

- Red Hat OpenShift (OCP) 4.x
- IBM Spectrum Virtualize for Public Cloud on AWS (for more information, see *IBM Spectrum Virtualize for Public Cloud on AWS Implementation Guide*, REDP-5534
- Red Hat OpenShift on AWS
- Spectrum Virtualize Family of Products - Global Mirror
- MySQL deployment in container
- Detailed technical configuration steps for building an end-to-end solution
- VPN connectivity, on-premises to public cloud (for more information, see *Solutions for Hybrid Cloud Networking Configuration Version 1 Release1*, REDP-5542

This Blueprint does not include the following information:

- Provide scalability and performance analysis from a user perspective
- Replace any official manuals and documents that are issued by IBM

Prerequisites

This technical report assumes that the person who is implementing this solution has the basic knowledge of or access to the following information:

- IBM Spectrum Virtualize for Public Cloud on AWS installation and configuration
- AWS Cloud login and required user rights and billing approval.
- Red Hat OCP: Red Hat OpenShift container platform 4.x
- Red Hat login credentials to download binaries and tools
- Storage Replication (IBM Global Mirror)
- VPC and VPN connectivity between On-premises Cloud to AWS public cloud
- CSI driver plug-ins
- Containerized MySQL database
- iSCSI basics and connectivity
- Required user names and passwords for AWS, SVPC, OCP and others accounts for installation
- VPN connectivity from On-premises network to AWS public cloud network

Demonstration introduction and architecture

The architecture that is used for this demonstration is shown in Figure 1.

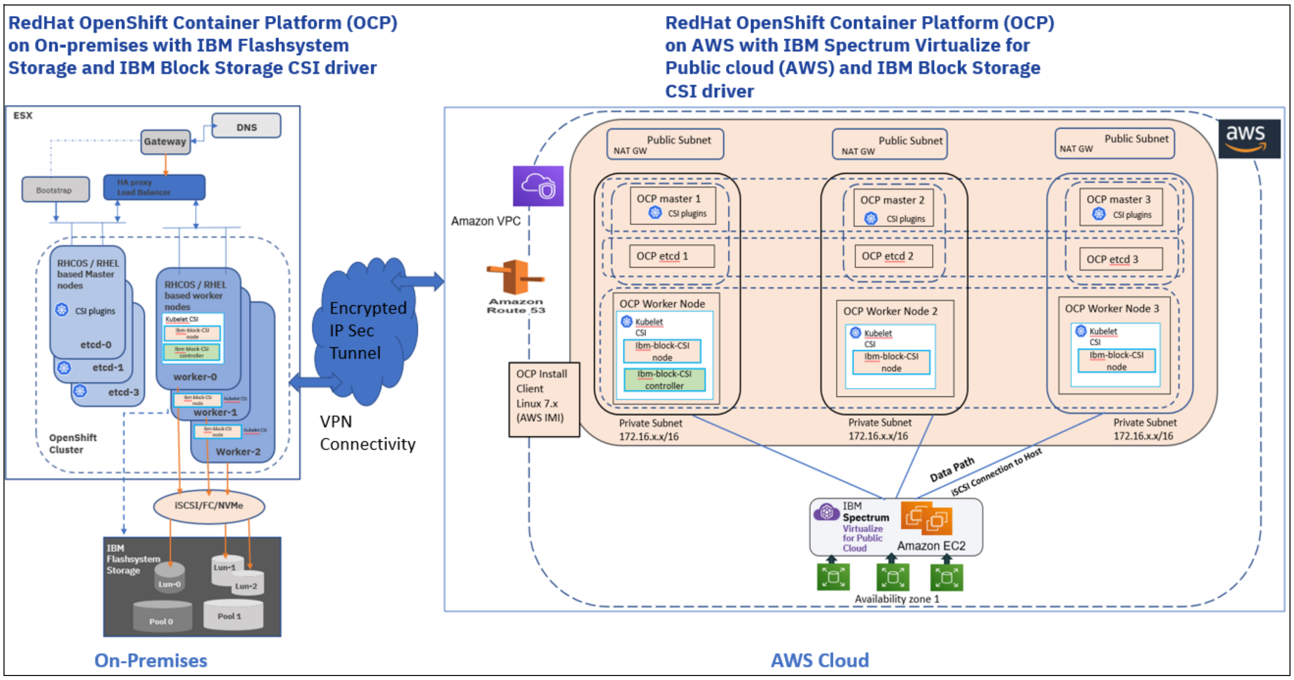


Figure 1 Hybrid multicloud architecture

Figure 2 shows the VPN connections between the On-premises and Public Cloud. For more information, see *IBM Solutions for Hybrid Cloud Networking Configuration Version 1 Release1*, REDP-5542.

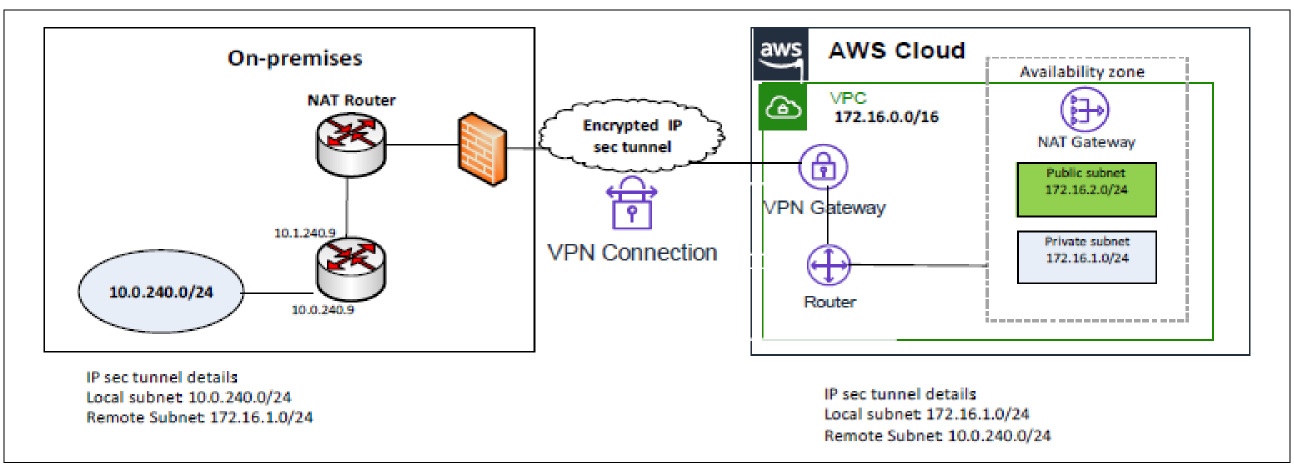


Figure 2 Hybrid cloud network connectivity

Demonstration purpose

The purpose of this document is to showcase the hybrid multicloud scenario for data replication between on-premises to public cloud (AWS).

This document is intended to facilitate the deployment of the hybrid multicloud business continuity solution with Red Hat OpenShift Container Platform and IBM block CSI driver plug-in for IBM Spectrum Virtualize on Public Cloud AWS. This solution is designed to protect the data by using the IBM Spectrum Virtualize Global Mirror function.

For demonstration purposes, MySQL containerized database is installed on the on-premises IBM FlashSystem® Storage that is connected to the OCP cluster (Red Hat OpenShift Container Platform cluster in the vSphere environment) by using the IBM block CSI driver. The volume (LUN) on IBM FlashSystem Storage is replicated by way of global mirror on IBM Spectrum® Virtualize for Public Cloud on AWS. Red Hat OpenShift cluster (OCP cluster) along with IBM block CSI driver plug-in, is installed on AWS by using IPI methodology of Red Hat OpenShift.

About OpenShift

Red Hat OpenShift Container Platform provides developers and IT organizations with a hybrid cloud application platform for deploying new and existing applications on secure, scalable resources with minimal configuration and management overhead. OpenShift Container Platform supports many programming languages and frameworks, such as Java, JavaScript, Python, Ruby, and PHP.

Built on Red Hat Enterprise Linux and Kubernetes, OpenShift Container Platform provides a more secure and scalable multi-tenant operating system for today's enterprise-class applications, while delivering integrated application run times and libraries. OpenShift Container Platform enables organizations to meet security, privacy, compliance, and governance requirements.

For more information, see this web page.

About OpenShift on AWS

The Quick Start includes AWS CloudFormation templates that build the AWS infrastructure that use AWS best practices, and then pass that environment to Ansible playbooks to build out the OpenShift environment. The AWS CloudFormation templates use AWS Lambda to generate a dynamic SSH key pair that is loaded into an Auto Scaling group. The Ansible inventory file is auto-generated. The combination of AWS CloudFormation and Ansible enables you to deploy and tear down your OpenShift environment by using CloudFormation stacks.

About IBM CSI and SVPC on AWS

IBM released its open-source CSI driver, which allows dynamic provisioning storage for containers on Kubernetes and Red Hat OpenShift container platform by using IBM Storage systems.

IBM Spectrum Storage™ family, IBM Spectrum Virtualize for Public Cloud (SVPC) and AWS supports clients in their IT architectural transformation and migration towards the cloud service model. This configuration enables hybrid cloud strategies or, for a cloud-native workload, provides the benefits of familiar and sophisticated storage functions on public cloud data centers which enhances the existing cloud offering.

For more information about SVPC on public cloud, see *IBM Spectrum Virtualize for Public Cloud on AWS Implementation Guide*, REDP-5534.

Demonstration 1: On-premises systems

Complete the following steps:

1. Red Hat OpenShift Installation and configuration (OCP4.x).

   For demonstration purposes, we installed the Red Hat OpenShift Platform on the VMware vSphere environment for the on-premises deployment.

   For more information about installation instructions and the prerequisites for installing OCP on vSphere, see *IBM Storage for Red Hat OpenShift Blueprint Version 1 Release 5*, REDP-5565.

2. Install IBM CSI Driver plug-in.

   For more information about installing IBM CSI driver on OCP, see this web page.

3. Log in to the node (`gw-10`), which is the node that was used to install the OCP cluster.

4. Issue the following commands to install the IBM CSI driver and check the status.

```
[root@gw-10 cli-tools]# curl https://raw.githubusercontent.com/IBM/ibm-block-csi-operator/master/deploy/installer/generated/ibm-block-csi-operato
.yaml > ibm-block-csi-operator.yaml
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 93246  100 93246    0     0   238k      0 --:--:-- --:--:-- --:--:--  239k
[root@gw-10 cli-tools]# ls
add-cluster-config-to-KUBECONFIG        docker-login-oc-registry  ibm-block-csi-operator.yaml  list-clusters  remove-cluster   restart-workers
check-image-push-status-during-install  download-mcm-cli-tools    icp-managed-cluster-login    mcm-login      remove-labels    soft-aliases
config                                  get-cert-files            import-cluster               oc-login       restart-masters  untag-images
[root@gw-10 cli-tools]# kubectl -n kube-system apply -f ibm-block-csi-operator.yaml
customresourcedefinition.apiextensions.k8s.io/ibmblockcsis.csi.ibm.com created
deployment.apps/ibm-block-csi-operator created
clusterrole.rbac.authorization.k8s.io/ibm-block-csi-operator created
clusterrolebinding.rbac.authorization.k8s.io/ibm-block-csi-operator created
serviceaccount/ibm-block-csi-operator created
[root@gw-10 cli-tools]# kubectl get pod -l app.kubernetes.io/name=ibm-block-csi-operator -n kube-system
NAME                                      READY   STATUS              RESTARTS   AGE
ibm-block-csi-operator-66b7d8b6b-8fmcs   0/1     ContainerCreating   0          <invalid>
[root@gw-10 cli-tools]# kubectl get pod -l app.kubernetes.io/name=ibm-block-csi-operator -n kube-system
NAME                                      READY   STATUS              RESTARTS   AGE
ibm-block-csi-operator-66b7d8b6b-8fmcs   0/1     ContainerCreating   0          22s
[root@gw-10 cli-tools]#
[root@gw-10 cli-tools]# kubectl get pod -l app.kubernetes.io/name=ibm-block-csi-operator -n kube-system
NAME                                      READY   STATUS    RESTARTS   AGE
ibm-block-csi-operator-66b7d8b6b-8fmcs   1/1     Running   0          <invalid>
[root@gw-10 cli-tools]#
```

Figure 3   Download the required files

```
[root@gw-10 cli-tools]# curl https://raw.githubusercontent.com/IBM/ibm-block-csi-operator/master/deploy/crds/csi.ibm.com_v1_ibmblockcsi_cr.yaml >
csi.ibm.com_v1_ibmblockcsi_cr.yaml
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  2013  100  2013    0     0   6542      0 --:--:-- --:--:-- --:--:--  6557
[root@gw-10 cli-tools]# kubectl -n kube-system apply -f csi.ibm.com_v1_ibmblockcsi_cr.yaml
ibmblockcsi.csi.ibm.com/ibm-block-csi created
[root@gw-10 cli-tools]#
```

Figure 4   Download the required files and install the driver

```
[root@gw-10 cli-tools]# kubectl get all -n kube-system  -l csi
NAME                                         READY   STATUS    RESTARTS   AGE
pod/ibm-block-csi-controller-0               4/4     Running   0          <invalid>
pod/ibm-block-csi-node-f6jc4                 3/3     Running   0          <invalid>
pod/ibm-block-csi-node-fdd5p                 3/3     Running   0          <invalid>
pod/ibm-block-csi-node-h56t7                 3/3     Running   0          <invalid>
pod/ibm-block-csi-node-hlw67                 3/3     Running   0          <invalid>
pod/ibm-block-csi-operator-66b7d8b6b-8fmcs   1/1     Running   0          <invalid>

NAME                                DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR   AGE
daemonset.apps/ibm-block-csi-node   4         4         4       4            4           <none>          104

NAME                                     READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/ibm-block-csi-operator   1/1     1            1           <invalid>

NAME                                               DESIRED   CURRENT   READY   AGE
replicaset.apps/ibm-block-csi-operator-66b7d8b6b   1         1         1       <invalid>

NAME                                        READY   AGE
statefulset.apps/ibm-block-csi-controller   1/1     105s
[root@gw-10 cli-tools]#
```

Figure 5   Pods status

This procedure is a command-line procedure that is used to install the IBM CSI driver. You also can install the CSI driver from the Operator hub by logging in to the OCP cluster by using GUI. Installing driver from the Operator hub is the recommended procedure to install the IBM CSI driver.

Complete the following steps to configure iSCSI storage on the worker nodes (in our example, it is FlashSystem storage):

1. Log in to the node (`gw-10`) and SSH to worker node. The `gw-10` node is the node that was used to install the cluster.
2. Log in with the core user:

   `[root@gw-10]# ssh core@sha-w1)`

3. For more information about configuring iSCSI storage, see IBM Knowledge Center.
4. Create the `/etc/multipath.conf` file on the worker node.

   A sample `multipath.conf` file is shown in Example 1.

Example 1   Sample /etc/multipath.conf file

```
defaults {
    path_checker tur
    path_selector "round-robin 0"
    rr_weight uniform
    prio const
    rr_min_io_rq 1
    polling_interval 30
    path_grouping_policy multibus
    find_multipaths yes
    no_path_retry fail
    user_friendly_names yes
    failback immediate
    checker_timeout 10
    fast_io_fail_tmo off
}
devices {
    device {
        path_checker tur
        product "FlashSystem"
        vendor "IBM"
        rr_weight uniform
        rr_min_io_rq 4
```

```
        path_grouping_policy multibus
        path_selector "round-robin 0"
        no_path_retry fail
        failback immediate
    }
    device {
        path_checker tur
        product "FlashSystem-9840"
        vendor "IBM"
        fast_io_fail_tmo off
        rr_weight uniform
        rr_min_io_rq 1000
        path_grouping_policy multibus
        path_selector "round-robin 0"
        no_path_retry fail
        failback immediate
    }
    device {
        vendor "IBM"
        product "2145"
        path_checker tur
        features "1 queue_if_no_path"
        path_grouping_policy group_by_prio
        path_selector "service-time 0" # Used by Red Hat 7.x
        prio alua
        rr_min_io_rq 1
        no_path_retry "5"
        dev_loss_tmo 120
        failback immediate
   }
```

5. Run the commands that are shown in Figure 6 on the worker node and follow the similar steps on all the worker nodes.

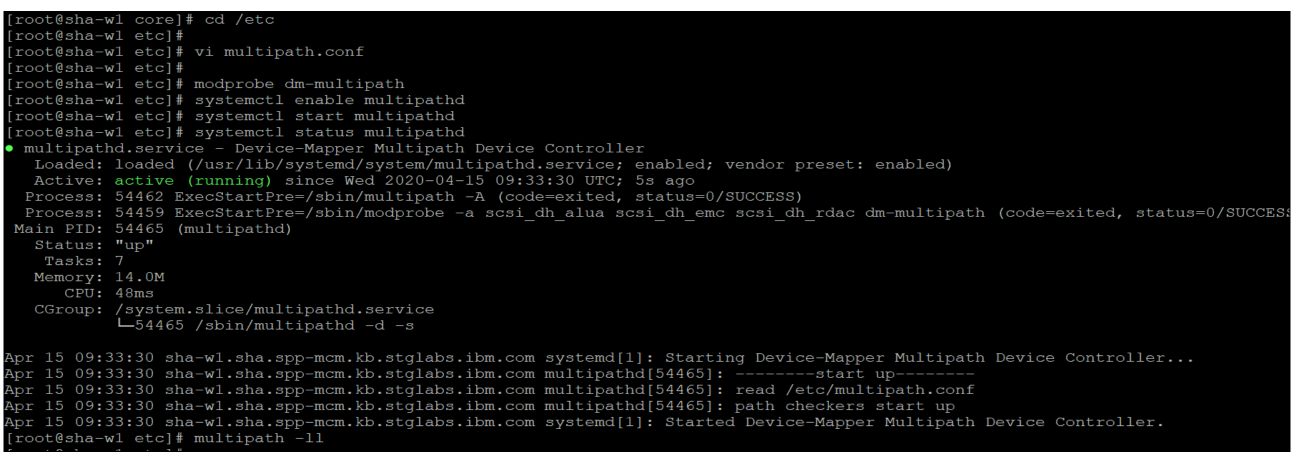


Figure 6   Configure multipath on the worker nodes

6. Identify the IQN number for each worker node and create a host mapping in the FlashSystem storage for all the worker nodes.

7. Log in to the worker node and **cat** this file to get IQN:

```
[sha-w1]# cat /etc/iscsi/initiatorname.iscsi
[sha-w1[# iscsiadm -m discoverydb -t st -p <storage ctrl IP>:3260 --discover
[sha-w1]# iscsiadm -m node  -p  <storage ctrl IP>:3260 –login
```

8. Configure Storage host mapping with the iSCSI IQN for each worker node (see Figure 7).

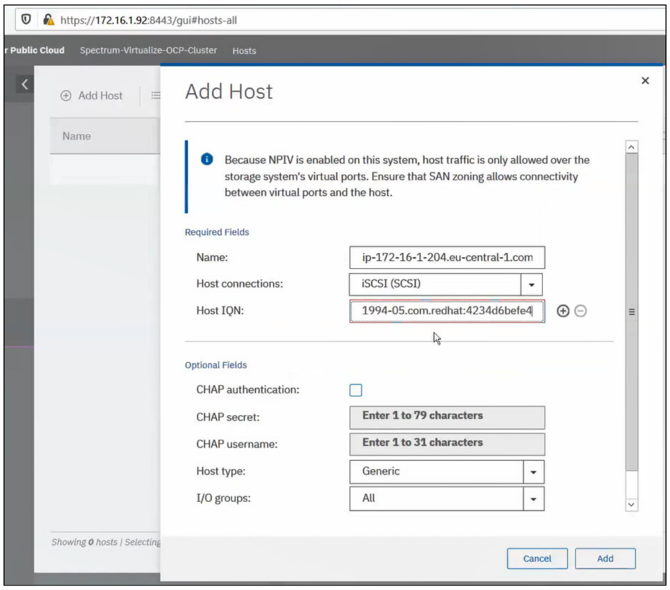


Figure 7 Sample output

Demonstration 2: AWS Cloud

For more information, see the custom installation method at this web page.

Note: With OCP version 4.3, installation of OpenShift can be done on existing VPC on AWS, ensure to complete the VPC prerequisites for installation.

Complete the following steps to install and configure Red Hat OpenShift, OCP4.3 on AWS by using the IPI method:

1. Create a RHEL 7.x Linux node with public IP from AWS Marketplace from the AWS console and wget the required files for installation. Ensure that you create this Linux node in the existing VPC network (see Figure 8 on page 10).

```
[root@ip-172-16-2-185 bin]# wget https://mirror.openshift.com/pub/openshift-v4/clients/ocp/stable-4.3/openshift-client-linux-4.3.9.tar.gz
--2020-04-13 09:51:34--  https://mirror.openshift.com/pub/openshift-v4/clients/ocp/stable-4.3/openshift-client-linux-4.3.9.tar.gz
Resolving mirror.openshift.com (mirror.openshift.com)... 54.172.173.155
Connecting to mirror.openshift.com (mirror.openshift.com)|54.172.173.155|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 27496983 (26M) [application/x-gzip]
Saving to: 'openshift-client-linux-4.3.9.tar.gz'

100%[=============================>] 27,496,983  7.35MB/s   in 3.9s

2020-04-13 09:51:38 (6.75 MB/s) - 'openshift-client-linux-4.3.9.tar.gz' saved [27496983/27496983]

[root@ip-172-16-2-185 bin]#
[root@ip-172-16-2-185 bin]# wget https://mirror.openshift.com/pub/openshift-v4/clients/ocp/stable-4.3/openshift-install-linux-4.3.9.tar.gz
--2020-04-13 09:51:52--  https://mirror.openshift.com/pub/openshift-v4/clients/ocp/stable-4.3/openshift-install-linux-4.3.9.tar.gz
Resolving mirror.openshift.com (mirror.openshift.com)... 54.172.173.155
Connecting to mirror.openshift.com (mirror.openshift.com)|54.172.173.155|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 82425374 (79M) [application/x-gzip]
Saving to: 'openshift-install-linux-4.3.9.tar.gz'

100%[=============================>] 82,425,374  6.94MB/s   in 13s

2020-04-13 09:52:05 (6.23 MB/s) - 'openshift-install-linux-4.3.9.tar.gz' saved [82425374/82425374]

[root@ip-172-16-2-185 bin]# pwd
```

Figure 8 Download required files for installation

For more information about the installation program, see this web page.

```
[root@ip-172-16-2-185 .ssh]# ssh-keygen -t rsa -b 4096 -N ''
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:bxu55hmHJMu9geB3WrWqHChaoWG/xpKCg5nQPRRsVl4 root@ip-172-16-2-18
The key's randomart image is:
+---[RSA 4096]----+
|    . .. E       |
|     =. .        |
|    o ..         |
|     .           |
| .oo. . S . .    |
|...+oo + B + .   |
|+o.o+.o * % o    |
|B oooo o *.@     |
| o.o.    ++B     |
+----[SHA256]-----+
[root@ip-172-16-2-185 .ssh]# pwd
/root/.ssh
```

Figure 9 Configure SSH

For more information about configuring `ssh-keygen`, see this web page.

```
[root@ip-172-16-2-185 .ssh]# eval "$(ssh-agent -s)"
Agent pid 29830
[root@ip-172-16-2-185 .ssh]#
```

Figure 10 Configure ssh-agent

For more information about the configuring agent, see this web page.

```
[root@ip-172-16-2-185 .ssh]# ssh-add /root/.ssh/id_rsa
Identity added: /root/.ssh/id_rsa (/root/.ssh/id_rsa)
[root@ip-172-16-2-185 .ssh]#
```

Figure 11 Configure SSH

2. Create the installation configuration file and customize the file for installation. When customized, the cluster is created on VPC of AWS. A sample `install-config.yaml` file is shown in Example 2 on page 11.

```
[root@ip-172-16-2-185 bin]# ./openshift-install create install-config --dir=/home/ec2-user/ocp43/config
? SSH Public Key /root/.ssh/id_rsa.pub
? Platform aws
? AWS Access Key ID AKIA4SGQ35FY32VV65PG
? AWS Secret Access Key [? for help] ****************************************
INFO Writing AWS credentials to "/root/.aws/credentials" (https://docs.aws.amazon.com/cli/latest/userguide/cli-configure-files.html)
? Region eu-central-1
? Base Domain ocp42svpc.com
? Cluster Name ocp43cluster
? Pull Secret [? for help] **************************************************************************************************************
*************************************************************************************************************************************************
*************************************************************************************************************************************************
*************************************************************************************************************************************************
*************************************************************************************************************************************************
*************************************************************************************************************************************************
*************************************************************************************************************************************************
*************************************************************************************************************************************************
*************************************************************************************************************************************************
*************************************************************************************************************************************************
*************************************************************************************************************************************************
*************************************************************************************************************************************************
*************************************************************************************************************************************************
*************************************************************************************************************************************************
*************************************************************************************************************************************************
*************************************************************************************************************************************************
*************************************************************************************************************************************************
*************************************************************************************************************************************************
*************************************************************************************************************************************************
[root@ip-172-16-2-185 bin]#
[root@ip-172-16-2-185 bin]# pwd
/home/ec2-user/ocp43/bin
```

Figure 12 Custom create install-config.yaml

3. This yaml file is a sample `install-config.yaml` file. Change the file per your environment and ensure that pull secret and ssh-key are added correctly in the `install-config.yaml` file.
4. Use m4.xlarge configuration for worker and master nodes. This sample can be used as the machine type per your requirement.
5. Ensure the correct subnet ID of the VPC for public and private network in your `install-config.yaml` file (see Example 2). For more information, see this web page.

Example 2 Sample install-config.yaml file for the custom installation

```
apiVersion: v1
baseDomain: ocp42svpc.com
controlPlane:
  hyperthreading: Enabled
  name: master
  platform:
    aws:
      zones:
     - eu-central-1a
      rootVolume:
        iops: 2000
        size: 500
        type: io1
      type: m4.xlarge
  replicas: 3
compute:
 hyperthreading: Enabled
  name: worker
```

```
  platform:
    aws:
      rootVolume:
        iops: 2000
        size: 500
        type: io1
      type: m4.large
      zones:
     - eu-central-1a
  replicas: 3
metadata:
  name: ocp43cluster
networking:
  machineCIDR: 172.16.0.0/16
platform:
  aws:
    region: eu-central-1
    subnets:
   - subnet-0aa84476708f32710
   - subnet-0f173ab757c352b11
pullSecret:
sshKey: |
```

6. With the modified `install-config.yaml` file, create the cluster (see Figure 13). This cluster is created in your existing VPC. For more information about VPC requirements, see this web page.

```
[root@ip-172-16-2-185 bin]# ./openshift-install create cluster --dir=/ho
me/ec2-user/ocp43/config --log-level=info
INFO Consuming Install Config from target directory
INFO Creating infrastructure resources...
INFO Waiting up to 30m0s for the Kubernetes API at https://api.ocp43clus
ter.ocp42svpc.com:6443...
INFO API v1.16.2 up
INFO Waiting up to 30m0s for bootstrapping to complete...
INFO Destroying the bootstrap resources...
INFO Waiting up to 30m0s for the cluster at https://api.ocp43cluster.ocp
42svpc.com:6443 to initialize...
INFO Waiting up to 10m0s for the openshift-console route to be created..
.
INFO Install complete!
INFO To access the cluster as the system:admin user when using 'oc', run
 'export KUBECONFIG=/home/ec2-user/ocp43/config/auth/kubeconfig'
INFO Access the OpenShift web-console here: https://console-openshift-co
nsole.apps.ocp43cluster.ocp42svpc.com
INFO Login to the console with user: kubeadmin, password: 64KRa-S8hkD-y6
rtk-FIm6k
[root@ip-172-16-2-185 bin]# export KUBECONFIG=/home/ec2-user/ocp43/confi
g/auth/kubeconfig
[root@ip-172-16-2-185 bin]#
```

Figure 13 Creation of OCP cluster on AWS

7. Check the status of the nodes and cluster by using the login and password information that is shown in Figure 13:

```
Export KUBECONFIG=/home/ec2-user/ocp43/config/
auth/kubeconfig
```

Figure 13 shows the output of the **openshift-install create cluster** command. Successful completion of this command provides the login, password, and export variable information. Use this information to check the status of installation.

```
[root@ip-172-16-2-185 bin]# export KUBECONFIG=/home/ec2-user/ocp43/conf
g/auth/kubeconfig
[root@ip-172-16-2-185 bin]#
```

Figure 14 Export kubeconfig

8. Check the status of nodes and cluster (see Figure 15 and Figure 16).

```
[root@ip-172-16-2-185 bin]# ./oc get nodes
NAME                                           STATUS   ROLES    AGE
VERSION
ip-172-16-1-153.eu-central-1.compute.internal   Ready    master   18m
v1.16.2
ip-172-16-1-204.eu-central-1.compute.internal   Ready    worker   11m
v1.16.2
ip-172-16-1-21.eu-central-1.compute.internal    Ready    worker   11m
v1.16.2
ip-172-16-1-244.eu-central-1.compute.internal   Ready    master   18m
v1.16.2
ip-172-16-1-53.eu-central-1.compute.internal    Ready    master   18m
v1.16.2
ip-172-16-1-94.eu-central-1.compute.internal    Ready    worker   11m
v1.16.2
```

Figure 15 Nodes status

```
[root@ip-172-16-2-185 bin]# ./oc get nodes -o wide
NAME                                            STATUS   ROLES    AGE
VERSION   INTERNAL-IP    EXTERNAL-IP   OS-IMAGE
                               KERNEL-VERSION                CONTAINER-R
NTIME
ip-172-16-1-153.eu-central-1.compute.internal   Ready    master   18m
v1.16.2   172.16.1.153   <none>        Red Hat Enterprise Linux CoreOS
3.81.202003230848.0 (Ootpa)   4.18.0-147.5.1.el8_1.x86_64   cri-o://1.1
.3-28.dev.rhaos4.3.git9aad8e4.el8
ip-172-16-1-204.eu-central-1.compute.internal   Ready    worker   11m
v1.16.2   172.16.1.204   <none>        Red Hat Enterprise Linux CoreOS
3.81.202003230848.0 (Ootpa)   4.18.0-147.5.1.el8_1.x86_64   cri-o://1.1
.3-28.dev.rhaos4.3.git9aad8e4.el8
ip-172-16-1-21.eu-central-1.compute.internal    Ready    worker   11m
v1.16.2   172.16.1.21    <none>        Red Hat Enterprise Linux CoreOS
3.81.202003230848.0 (Ootpa)   4.18.0-147.5.1.el8_1.x86_64   cri-o://1.1
.3-28.dev.rhaos4.3.git9aad8e4.el8
ip-172-16-1-244.eu-central-1.compute.internal   Ready    master   18m
v1.16.2   172.16.1.244   <none>        Red Hat Enterprise Linux CoreOS
3.81.202003230848.0 (Ootpa)   4.18.0-147.5.1.el8_1.x86_64   cri-o://1.1
.3-28.dev.rhaos4.3.git9aad8e4.el8
ip-172-16-1-53.eu-central-1.compute.internal    Ready    master   18m
v1.16.2   172.16.1.53    <none>        Red Hat Enterprise Linux CoreOS
3.81.202003230848.0 (Ootpa)   4.18.0-147.5.1.el8_1.x86_64   cri-o://1.1
.3-28.dev.rhaos4.3.git9aad8e4.el8
ip-172-16-1-94.eu-central-1.compute.internal    Ready    worker   11m
v1.16.2   172.16.1.94    <none>        Red Hat Enterprise Linux CoreOS
3.81.202003230848.0 (Ootpa)   4.18.0-147.5.1.el8_1.x86_64   cri-o://1.1
.3-28.dev.rhaos4.3.git9aad8e4.el8
[root@ip-172-16-2-185 bin]#
```

Figure 16 Nodes status

Installing IBM CSI driver on the OCP cluster that is installed on AWS

Complete the following steps:

1. Log in to the cluster GUI with the link and the user name and password that is provided in the output of the **`openshift-install create cluster`** command.
2. Follow the driver installation instructions that are available at this web page.
3. Select **Operators** from the left side of the window. A list of operators is shown on the right side (see Figure 17).

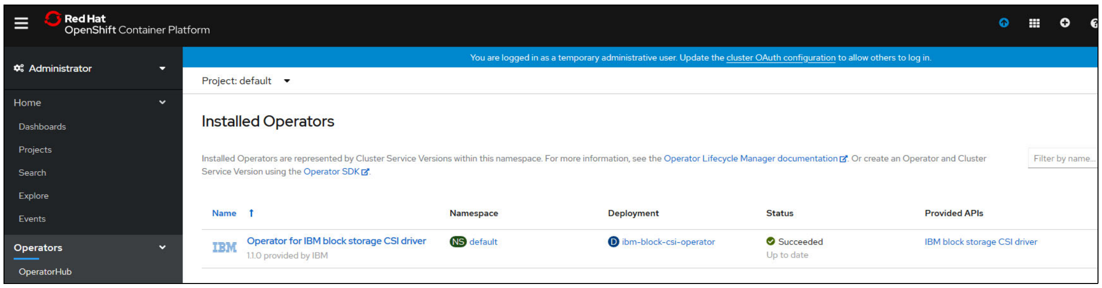


Figure 17 GUI log in operators option

4. Select the **IBM Block Storage CSI Driver**. Follow the procedure to create the operator and check the status of the Operator pod (see Figure 18).

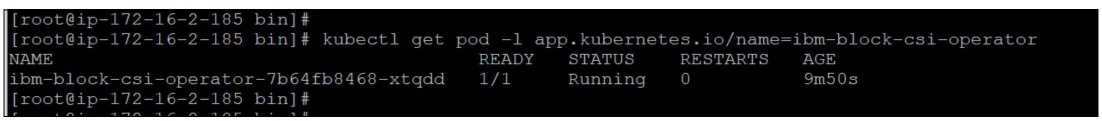


Figure 18 Status of CSI Operator pod

5. Click the installed operators in the left window. You see the IBM block storage CSI driver operator (see Figure 19). Click the operator and then, **Create instance**.

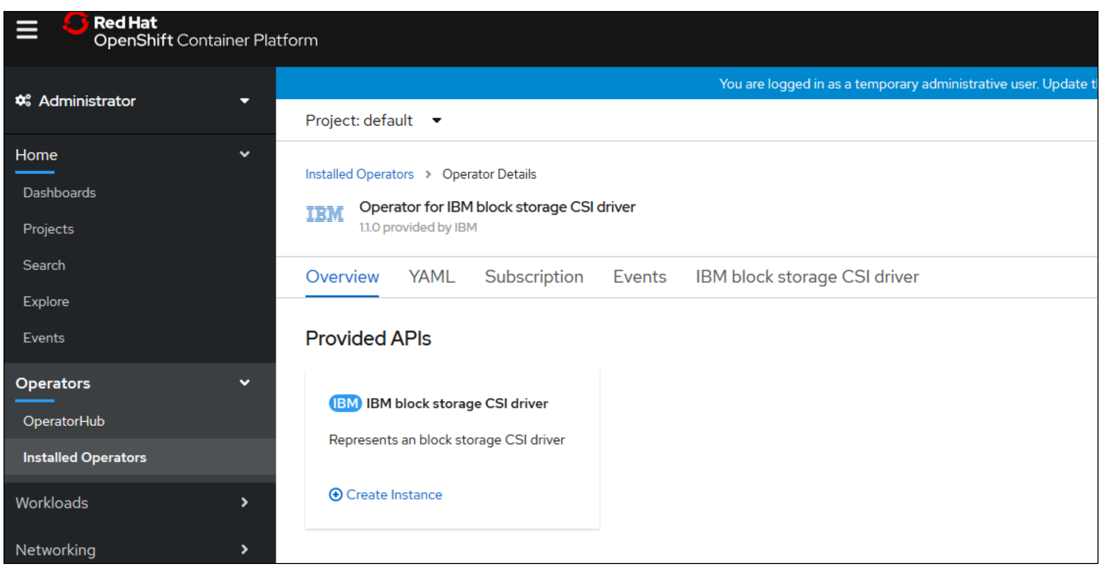


Figure 19 Creating an instance

The instance is created (see Figure 20).

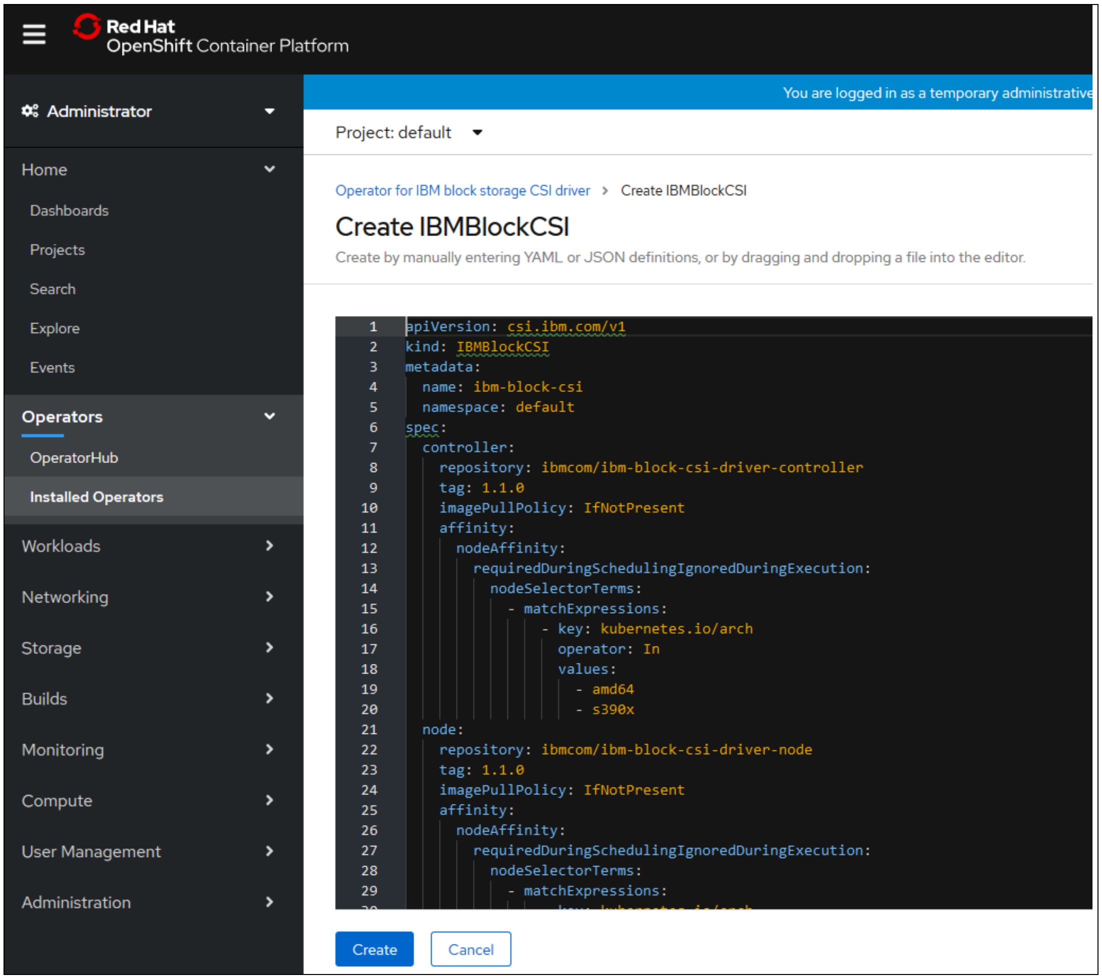


Figure 20 Creating instance option

6. Check the status of the pods:

```
# kubectl get all -n kube-system -l csi
```

You can see the similar output for the pods (see Figure 21).

```
NAME                                          READY   STATUS    RESTARTS   AGE
pod/ibm-block-csi-controller-0                4/4     Running   0          <invalid>
pod/ibm-block-csi-node-f6jc4                  3/3     Running   0          <invalid>
pod/ibm-block-csi-node-fdd5p                  3/3     Running   0          <invalid>
pod/ibm-block-csi-node-h56t7                  3/3     Running   0          <invalid>
pod/ibm-block-csi-node-hlw67                  3/3     Running   0          <invalid>
pod/ibm-block-csi-operator-66b7d8b6b-8fmcs    1/1     Running   0          <invalid>

NAME                                 DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR   AGE
daemonset.apps/ibm-block-csi-node    4         4         4       4            4           <none>          104s

NAME                                     READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/ibm-block-csi-operator   1/1     1            1           <invalid>

NAME                                                DESIRED   CURRENT   READY   AGE
replicaset.apps/ibm-block-csi-operator-66b7d8b6b    1         1         1       <invalid>

NAME                                         READY   AGE
statefulset.apps/ibm-block-csi-controller    1/1     105s
```

Figure 21 Pod status

Configuring iSCSI on SVPC Storage in AWS

Complete the following steps:

1. Log in to the node `[root@ip-172.16.2.185]#` and SSH to the worker node. The `[root@ip-172.16.2.185]#` is the node that was used for cluster installation.
2. Log in to the worker node with the core user:

   ```
   [root@ip-172.16.2.185]# ssh core@ ip-172-16-1-94.eu-central-1.compute.internal
   ```

3. Configure iSCSI storage by using the process that is described at this web page.

   Or

   Log in to the worker node and create the `/etc/multipath.conf` file. A sample `multipath.conf` file is shown in Example 3.

Example 3 Sample multipath.conf file

```
defaults {
    path_checker tur
    path_selector "round-robin 0"
    rr_weight uniform
    prio const
    rr_min_io_rq 1
    polling_interval 30
    path_grouping_policy multibus
    find_multipaths yes
    no_path_retry fail
    user_friendly_names yes
    failback immediate
    checker_timeout 10
    fast_io_fail_tmo off
}
devices {
    device {
        path_checker tur
        product "FlashSystem"
        vendor "IBM"
        rr_weight uniform
        rr_min_io_rq 4
        path_grouping_policy multibus
        path_selector "round-robin 0"
        no_path_retry fail
        failback immediate
    }
    device {
        path_checker tur
        product "FlashSystem-9840"
        vendor "IBM"
        fast_io_fail_tmo off
        rr_weight uniform
        rr_min_io_rq 1000
        path_grouping_policy multibus
        path_selector "round-robin 0"
        no_path_retry fail
        failback immediate
    }
```

```
        device {
            vendor "IBM"
            product "2145"
            path_checker tur
            features "1 queue_if_no_path"
            path_grouping_policy group_by_prio
            path_selector "service-time 0" # Used by Red Hat 7.x
            prio alua
            rr_min_io_rq 1
            no_path_retry "5"
            dev_loss_tmo 120
            failback immediate
       }
}
```

4. Identify the IQN number for each worker node and create a host mapping in the SVPC storage on all the worker nodes.
5. Log in to the worker node and cat the following file to get IQN (see Figure 22):

```
[root@ip-172-16-1-204]#  cat /etc/iscsi/initiatorname.iscsi
```

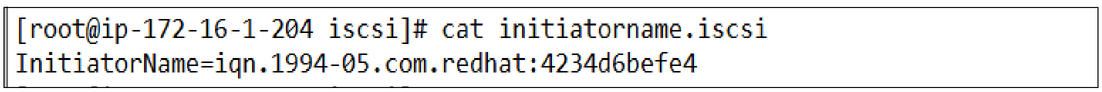


Figure 22 Check iSCSI initiator

6. Create host-mapping in the SVPC storage and check the status (see Figure 23 and Figure 24).

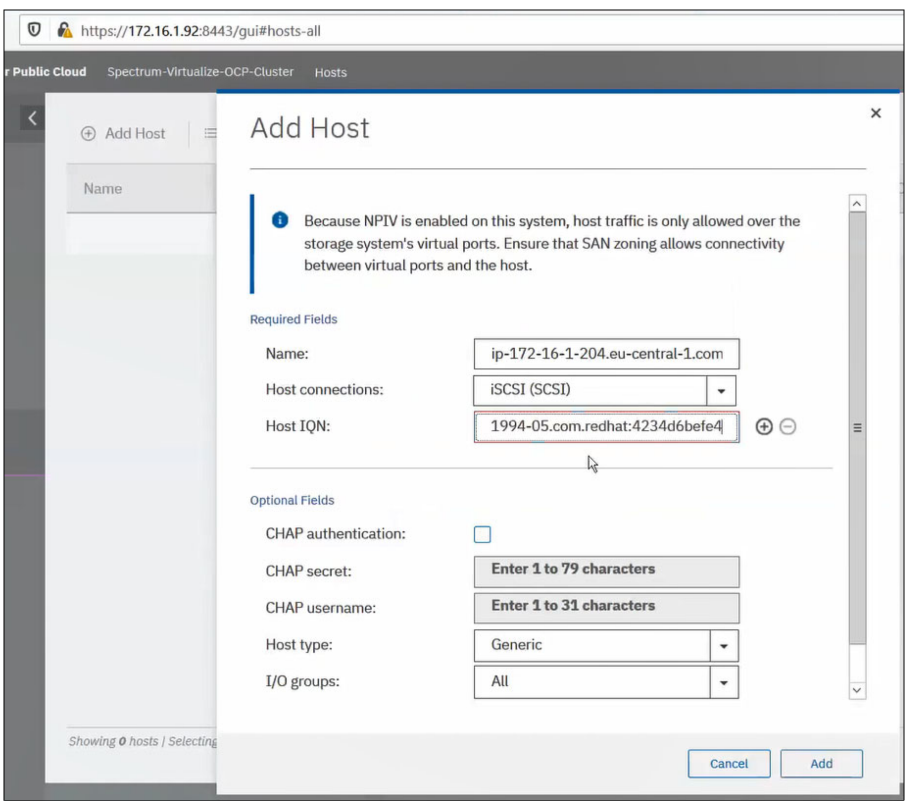


Figure 23 Configure host mapping

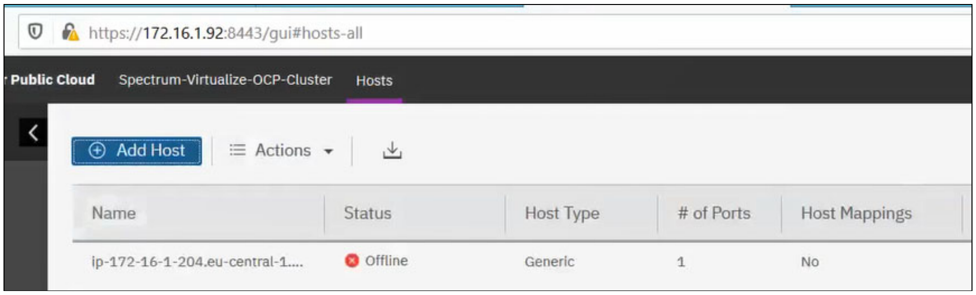


Figure 24 Add host

7. Check the IP address of the storage iSCSI network (see Figure 25).

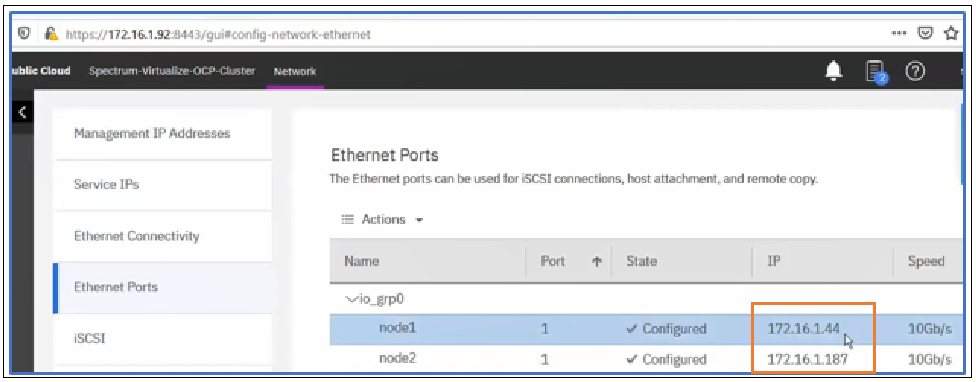


Figure 25 IP address for iSCI storage network

8. Run the following commands on each worker nodes (see Figure 26).

```
[root@ip-172-16-1-204]# modprobe dm-multipath
[root@ip-172-16-1-204]# systemctl enable multipathd
[root@ip-172-16-1-204]# systemctl start multipathd
[root@ip-172-16-1-204]# systemctl status multipathd
[root@ip-172-16-1-204]# multipath -ll
[root@ip-172-16-1-204]# iscsiadm -m discoverydb -t st -p <SVPC storage ctrl
IP>:3260 –discover
[root@ip-172-16-1-204]#iscsiadm -m node  -p  <SVPC storage ctrl IP>:3260 –login
```

```
[root@ip-172-16-1-204 iscsi]# iscsiadm -m discoverydb -t st -p 172.16.1.44:3260 --discover
           172.16.1.44:3260,1 iqn.1986-03.com.ibm:2145.spectrum-virtualize-ocp-cluster.node1
172.16.1.247:3260,1 iqn.1986-03.com.ibm:2145.spectrum-virtualize-ocp-cluster.node1

[root@ip-172-16-1-204 iscsi]# iscsiadm -m node  -p 172.16.1.44:3260 --login
Logging in to [iface: default, target: iqn.1986-03.com.ibm:2145.spectrum-virtualize-ocp-cluster.node1, portal: 172.16.1.44,3260]
Login to [iface: default, target: iqn.1986-03.com.ibm:2145.spectrum-virtualize-ocp-cluster.node1, portal: 172.16.1.44,3260]
successful.

[root@ip-172-16-1-204 iscsi]#
```

Figure 26 Configuring iSCI

9. Check the status of host in the storage (see Figure 27 on page 19).

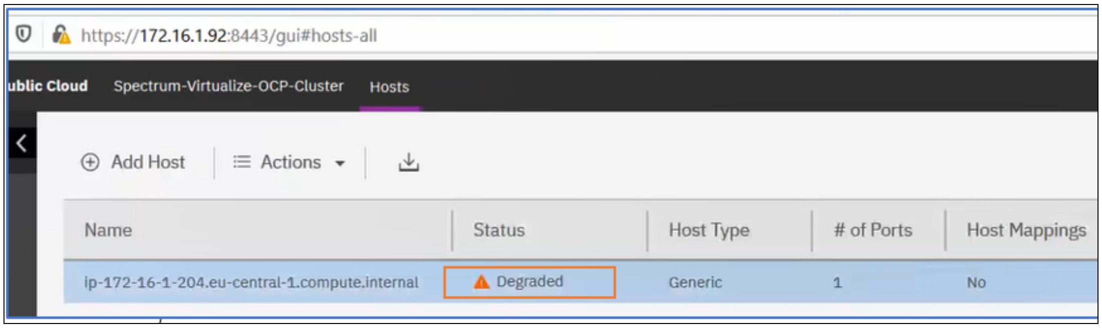


Figure 27 Host status

Post iSCSI configuration, the IBM CSI Block Storage driver is ready to use.

Demonstration 3: VPN connectivity and SVPC on AWS

The required components are now installed on AWS Public cloud and On-Premises Systems.

This demonstration describes the VPN connection between On-premises Systems and AWS Public Cloud. For more information, see Figure 1 on page 4 and Figure 2 on page 4.

VPN connectivity from on-premises network to AWS Cloud network

More information about VPN connectivity between on-premises-private cloud to public cloud on AWS, see *IBM Solutions for Hybrid Cloud Networking Configuration Version 1 Release1*, REDP-5542.

Configuring IBM Spectrum Virtualize on Public Cloud running on AWS

For more information about SVPC on public cloud, see *IBM Spectrum Virtualize for Public Cloud on AWS Implementation Guide*, REDP-5534.

Demonstration 4: Hybrid multicloud business continuity

The required components are now installed and configured.

This demonstration is the final part of the demonstration of the Hybrid cloud business continuity use case.

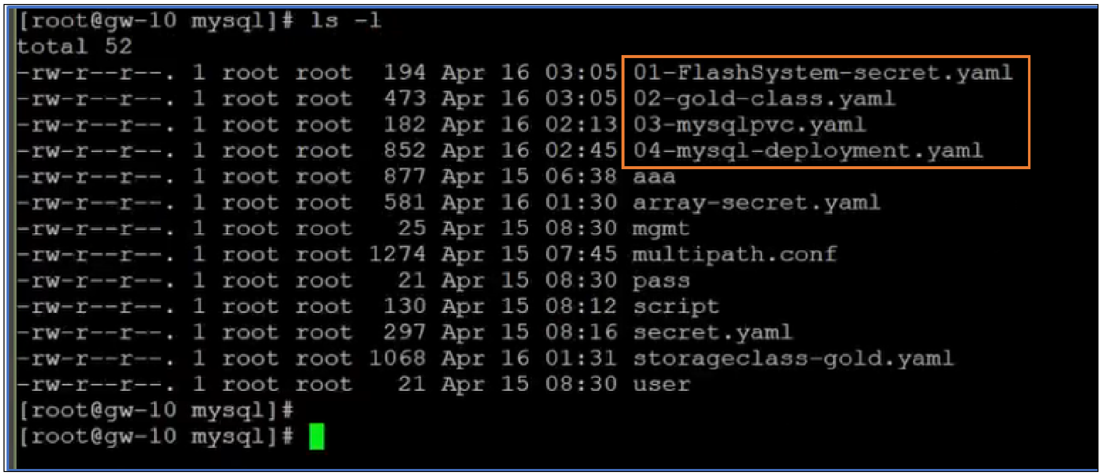
Installing MySQL on the OCP Clusters on-premises

We use MySQL as a sample database to be deployed on the OpenShift Clusters. MySQL is deployed on the On-premises OpenShift cluster and Public cloud (AWS) OpenShift cluster by using the sample `yaml` file.

Post deployment of MySQL on the On-premise OpenShift cluster uses the sample `yaml` files. MySQL database uses On-premises IBM FlashSystem Storage to create the persistent volume. This Persistent Volume is replicated to the volume on SVPC (AWS Cloud).

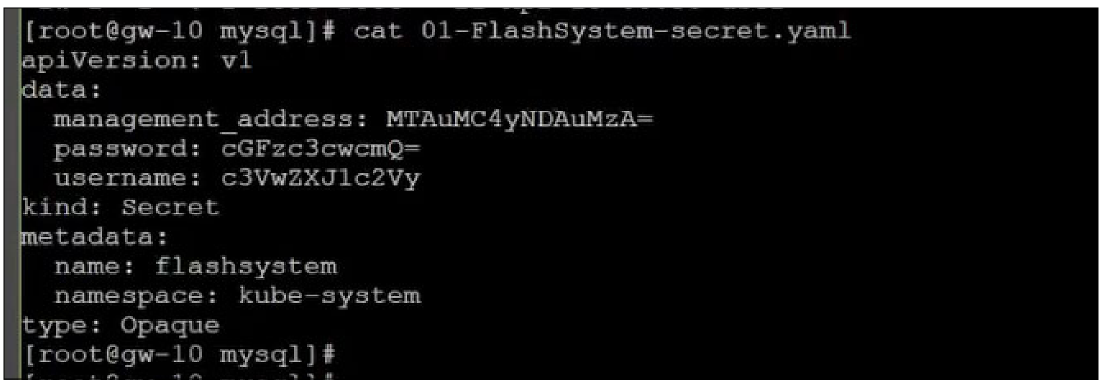
Follow this section for more information about how to deploy the sample My SQL database and replicate the data between On-Premises and AWS Cloud.

Figure 28 - Figure 33 on page 22 display the MySQL deployment steps and configuration files.



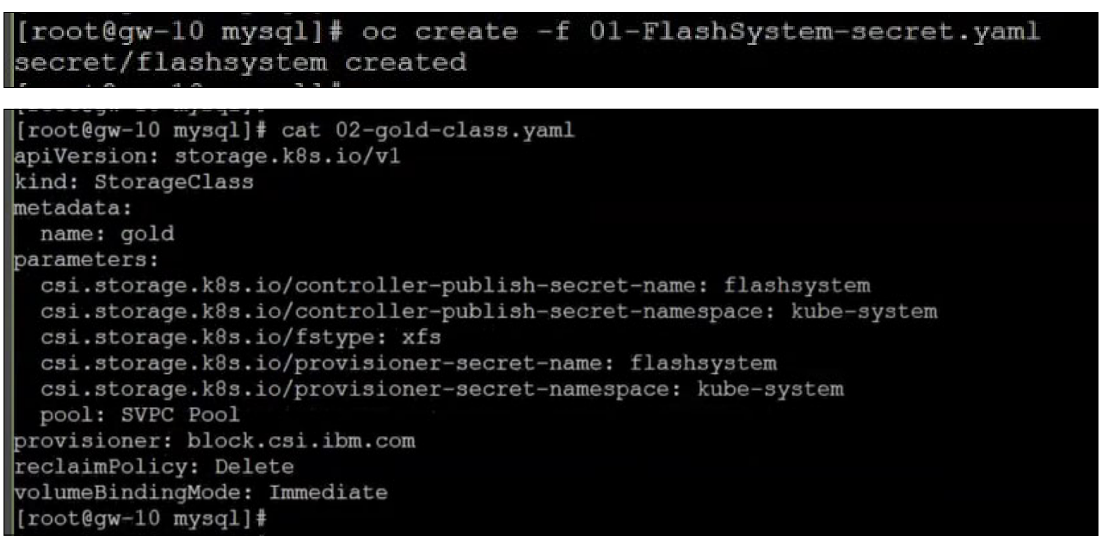
```
[root@gw-10 mysql]# ls -l
total 52
-rw-r--r--. 1 root root  194 Apr 16 03:05 01-FlashSystem-secret.yaml
-rw-r--r--. 1 root root  473 Apr 16 03:05 02-gold-class.yaml
-rw-r--r--. 1 root root  182 Apr 16 02:13 03-mysqlpvc.yaml
-rw-r--r--. 1 root root  852 Apr 16 02:45 04-mysql-deployment.yaml
-rw-r--r--. 1 root root  877 Apr 15 06:38 aaa
-rw-r--r--. 1 root root  581 Apr 16 01:30 array-secret.yaml
-rw-r--r--. 1 root root   25 Apr 15 08:30 mgmt
-rw-r--r--. 1 root root 1274 Apr 15 07:45 multipath.conf
-rw-r--r--. 1 root root   21 Apr 15 08:30 pass
-rw-r--r--. 1 root root  130 Apr 15 08:12 script
-rw-r--r--. 1 root root  297 Apr 15 08:16 secret.yaml
-rw-r--r--. 1 root root 1068 Apr 16 01:31 storageclass-gold.yaml
-rw-r--r--. 1 root root   21 Apr 15 08:30 user
[root@gw-10 mysql]#
[root@gw-10 mysql]#
```

Figure 28 List of yaml files



```
[root@gw-10 mysql]# cat 01-FlashSystem-secret.yaml
apiVersion: v1
data:
  management_address: MTAuMC4yNDAuMzA=
  password: cGFzc3cwcmQ=
  username: c3VwZXJ1c2Vy
kind: Secret
metadata:
  name: flashsystem
  namespace: kube-system
type: Opaque
[root@gw-10 mysql]#
```

Figure 29 Storage secret creation



```
[root@gw-10 mysql]# oc create -f 01-FlashSystem-secret.yaml
secret/flashsystem created
```

```
[root@gw-10 mysql]# cat 02-gold-class.yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: gold
parameters:
  csi.storage.k8s.io/controller-publish-secret-name: flashsystem
  csi.storage.k8s.io/controller-publish-secret-namespace: kube-system
  csi.storage.k8s.io/fstype: xfs
  csi.storage.k8s.io/provisioner-secret-name: flashsystem
  csi.storage.k8s.io/provisioner-secret-namespace: kube-system
  pool: SVPC Pool
provisioner: block.csi.ibm.com
reclaimPolicy: Delete
volumeBindingMode: Immediate
[root@gw-10 mysql]#
```

Figure 30 Storage class creation

```
[root@gw-10 mysql]# oc create -f 02-gold-class.yaml
storageclass.storage.k8s.io/gold created
```

```
[root@gw-10 mysql]# cat 03-mysqlpvc.yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: mysql-pvc
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 20Gi
  storageClassName: gold
[root@gw-10 mysql]#
```

Figure 31 Storage PVC creation

```
[root@gw-10 mysql]# oc create -f 03-mysqlpvc.yaml
persistentvolumeclaim/mysql-pvc created
```

```
[root@gw-10 mysql]# cat 04-mysql-deployment.yaml
```

Figure 32 Deployment of MySQL

```
kind: Service
metadata:
  name: mysql
spec:
  ports:
  - port: 3306
  selector:
    app: mysql
  clusterIP: None
---
apiVersion: apps/v1 # for versions before 1.9.0 use apps/v1beta2
kind: Deployment
metadata:
  name: mysql
spec:
  selector:
    matchLabels:
      app: mysql
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        app: mysql
    spec:
      containers:
      - image: mysql:5.6
        name: mysql
        env:
          # Use secret in real usage
        - name: MYSQL_ROOT_PASSWORD
          value: password
        ports:
        - containerPort: 3306
          name: mysql
        volumeMounts:
        - name: mysql-persistent-storage
          mountPath: /var/lib/mysql
      volumes:
      - name: mysql-persistent-storage
        persistentVolumeClaim:
          claimName: mysql-pvc
```

Figure 33 Deployment of mysql

```
[root@gw-10 mysql]# oc create -f 04-mysql-deployment.yaml
service/mysql created
deployment.apps/mysql created
[root@gw-10 mysql]#
```

MySQL deployment is completed, check the status of the pods and the deployment.

Figure 34 on page 23 - Figure 39 on page 25 display the configuration files and steps to create a sample MySQL database and data in the database for On-premises Systems.

The On-premises MySQL database is installed on the Storage volumes that are created on the On-Premises IBM FlashSystem storage. This storage volume is replicated to SVPC Volume on AWS Cloud by using the IBM spectrum Virtualize family global mirror function.

Follow the steps to complete the replication and deployment of MySQL on AWS Cloud and On-premises OpenShift Cluster.

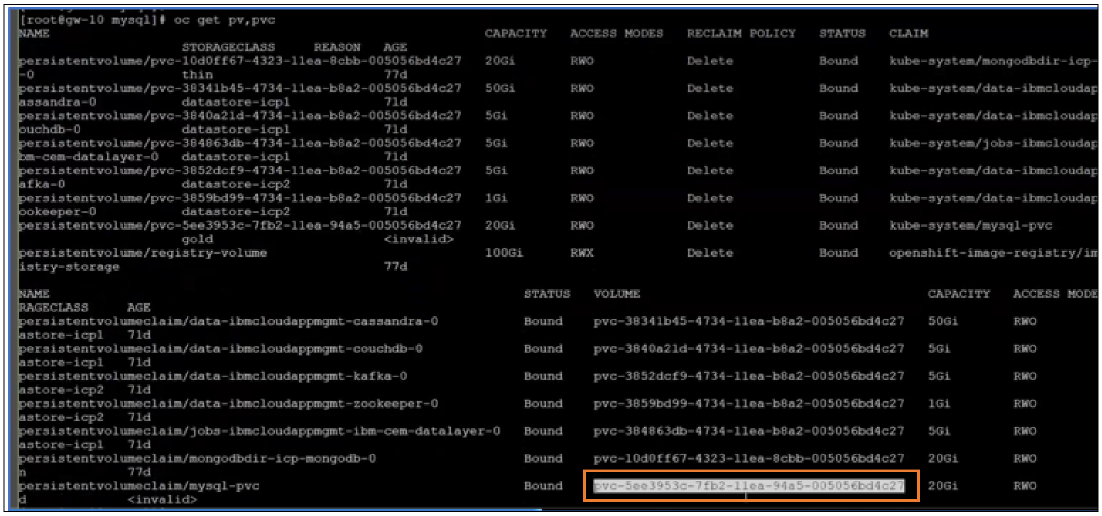


Figure 34 Status of PVC and volume ID

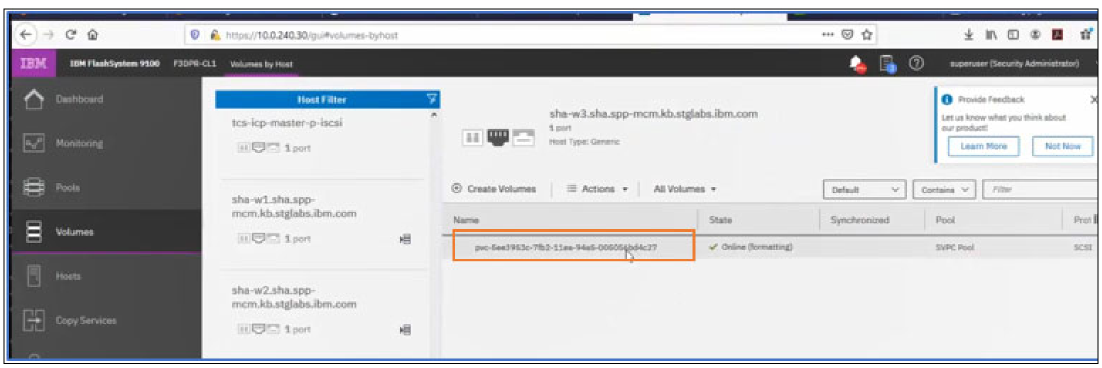


Figure 35 Name of PVC and ID

```
[root@gw-10 mysql]# oc get deployment | grep mysql
mysql                                    1/1     1            1           2m18s
[root@gw-10 mysql]#
[root@gw-10 mysql]# oc get pods | grep mysql
mysql-5bb6fc74b4-lj9fs                                  1/1     Running              0          2m43s
[root@gw-10 mysql]#
[root@gw-10 mysql]#
[root@gw-10 mysql]# oc rsh mysql-5bb6fc74b4-lj9fs
#
#
#
# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.6.47 MySQL Community Server (GPL)

Copyright (c) 2000, 2020, 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> exit
```

Figure 36 Status of mysql and pod login

```
[root@gw-10 mysql]# oc rsh mysql-5bb6fc74b4-lj9fs
#
#
# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.6.47 MySQL Community Server (GPL)

Copyright (c) 2000, 2020, 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>
mysql>
mysql>  show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
+--------------------+
3 rows in set (0.00 sec)

mysql>
```

Figure 37 MySQL database list

```
mysql> create database ocp_svpc
    -> ;
Query OK, 1 row affected (0.00 sec)

mysql> use database ocp_svpc;
ERROR 1049 (42000): Unknown database 'database'
mysql> use ocp_svpc;
Database changed
mysql>
mysql>
mysql> create table ocp_svpc_table ( col1 int );
Query OK, 0 rows affected (0.03 sec)

mysql> insert into ocp_svpc_table values (1);
Query OK, 1 row affected (0.00 sec)

mysql> insert into ocp_svpc_table values (2);
Query OK, 1 row affected (0.01 sec)

mysql> insert into ocp_svpc_table values (3);
Query OK, 1 row affected (0.00 sec)

mysql> insert into ocp_svpc_table values (4);
Query OK, 1 row affected (0.00 sec)

mysql> insert into ocp_svpc_table values (5);
Query OK, 1 row affected (0.00 sec)

mysql>
```

Figure 38 mysql database creation and data insertion

```
mysql>
mysql> select * from ocp_svpc_table;
+-------+
| col1  |
+-------+
|     1 |
|     2 |
|     3 |
|     4 |
|     5 |
+-------+
5 rows in set (0.00 sec)

mysql> [root@gw-10 mysql]#
```

Figure 39 Data in mysql database

MySQL is successfully deployed and the `ocp_svpc` database is created with the table name `ocp_svpc_table`. This table includes five rows that were inserted.

When we complete the Global mirror business continuity case, these five rows and the same database are available in public cloud MySQL.

Complete the following steps to configure storage-based replications (IBM Global mirror):

1. Log in to the on-premises FlashSystem storage and create a partnership with the Public cloud SVPC storage.
2. Create an IP-based partnership with the public cloud AWS SVPC (see Figure 40 on page 26).

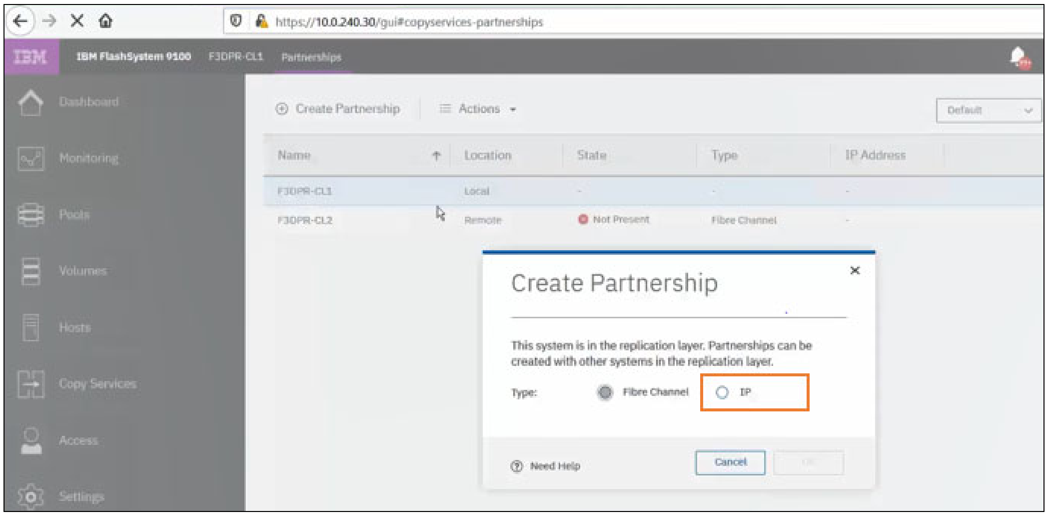


Figure 40 Creating the storage partnership

The details of the partnership are shown in Figure 41.

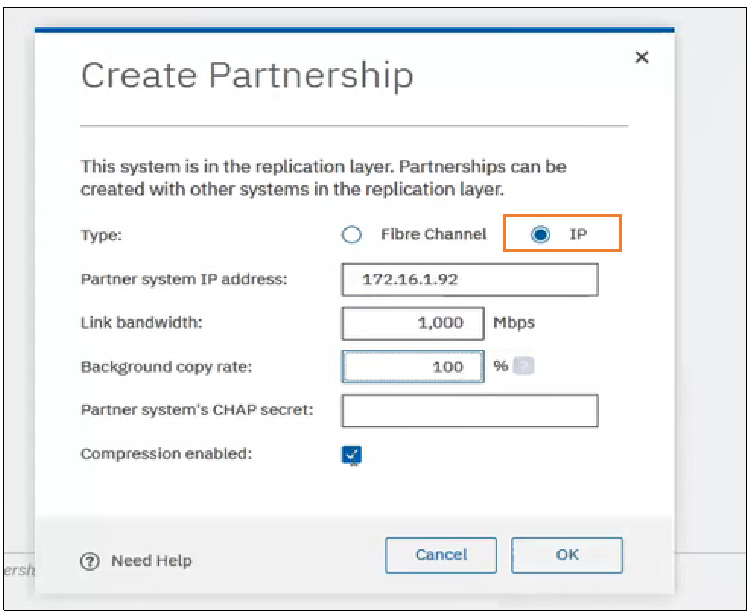


Figure 41 Partnership details

The partnership is created, as shown in Figure 42.

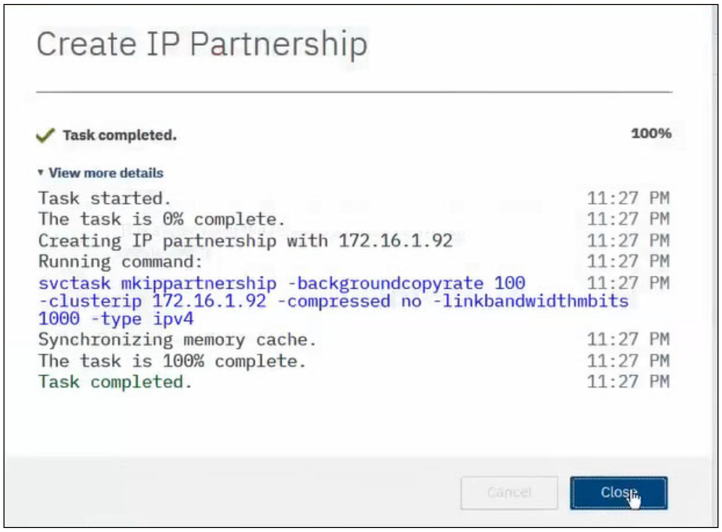


Figure 42 Partnership is created

The status of the created partnership is displayed, as shown in Figure 43.

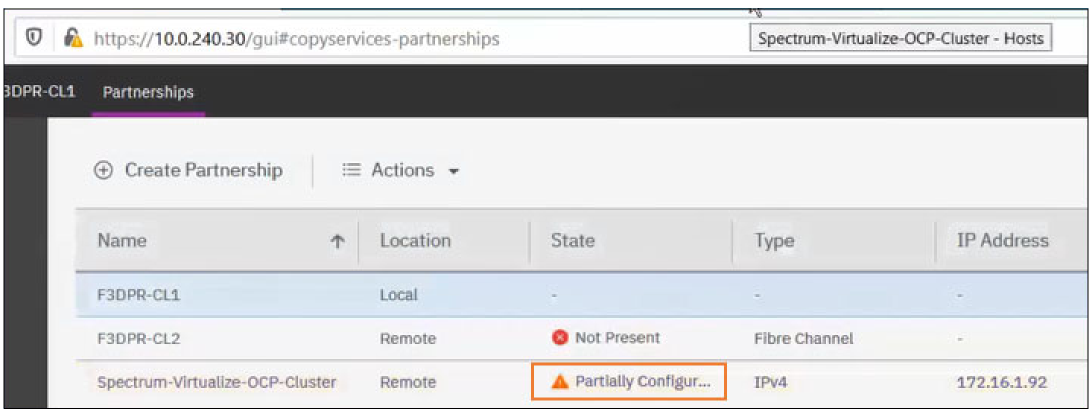


Figure 43 Status of creating partnership

3. Log in to the AWS Cloud, SVPC storage and create the partnership.

4. Modify the IPv4 remote copy to enabled (see Figure 44).

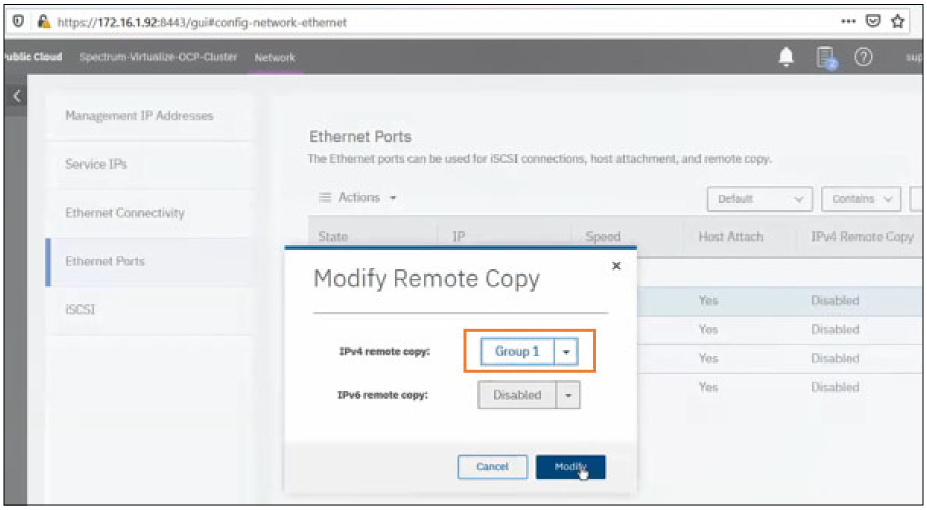


Figure 44 Modifying the parameter

5. Select the **IP** option for the Fibre Channel (see Figure 45).

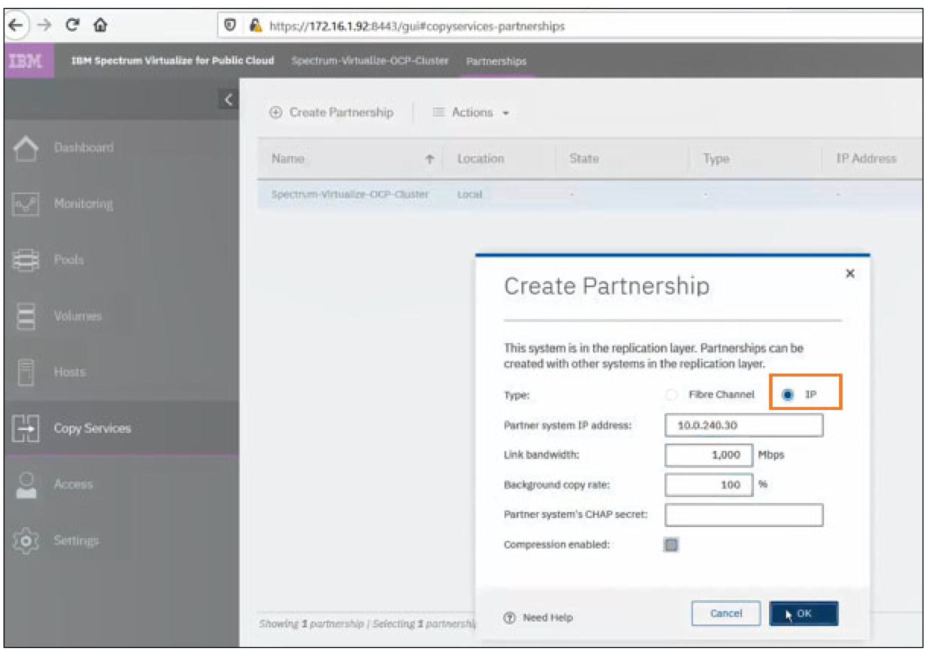


Figure 45 Selecting IP option

The progress of the synchronization process is displayed (see Figure 46).

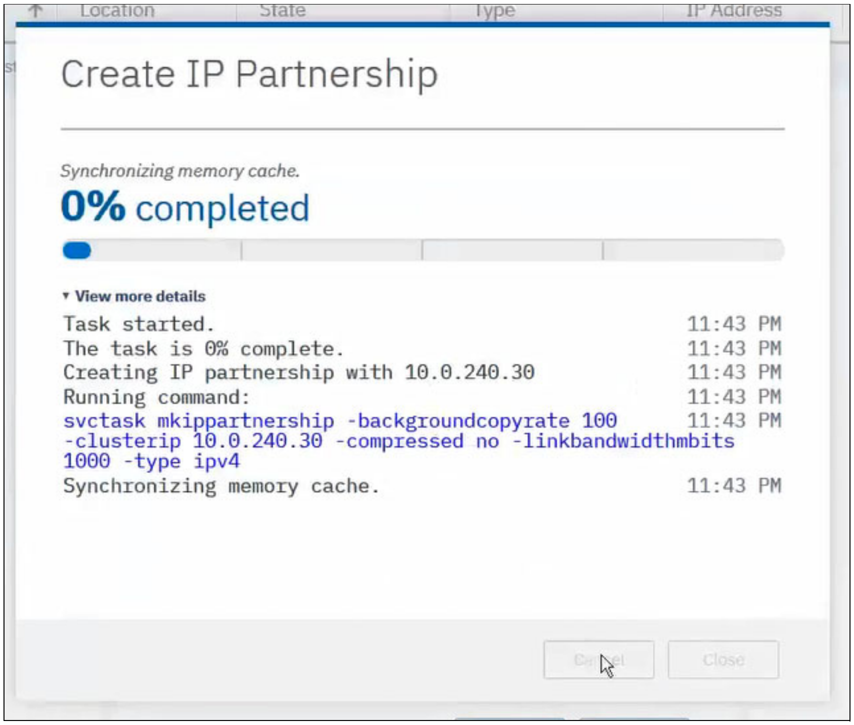


Figure 46 Synchronization progress

6. Check the state of partnership; it should be fully configured (see Figure 47).

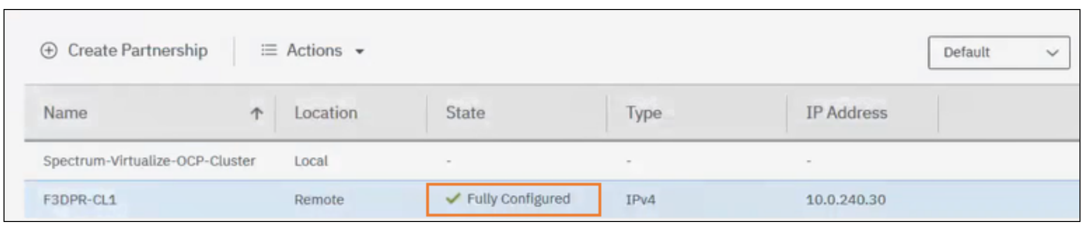


Figure 47 Partnership status

Creating an equivalent storage capacity LUN on Public cloud AWS SVPC

Complete the following steps:

1. Log in to the Public cloud AWS SVPC storage and create a LUN for replication (see Figure 48).

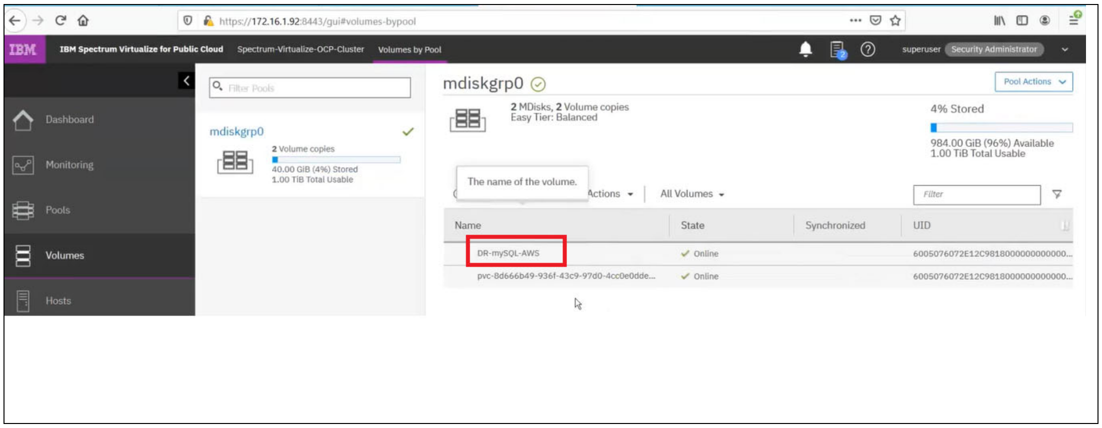


Figure 48 Replicated volume name of SVPC on AWS

2. Identify the storage LUN where MySQL is installed on the on-premises FlashSystem Storage and replicate the storage LUN from on-premises FlashSystem storage (volume `pvc-5ee3953c-7fb2-11ea-94a5-005056bd4c27`) to public cloud AWS SVPC LUN (DR-mySQL-AWS is the name of the LUN).

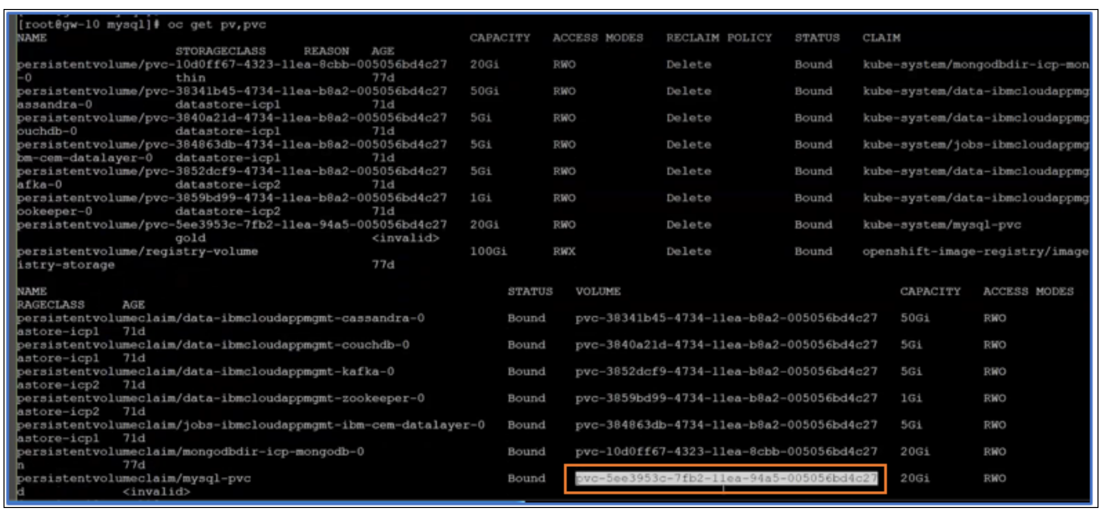


Figure 49 Volume ID for MySQL PVC

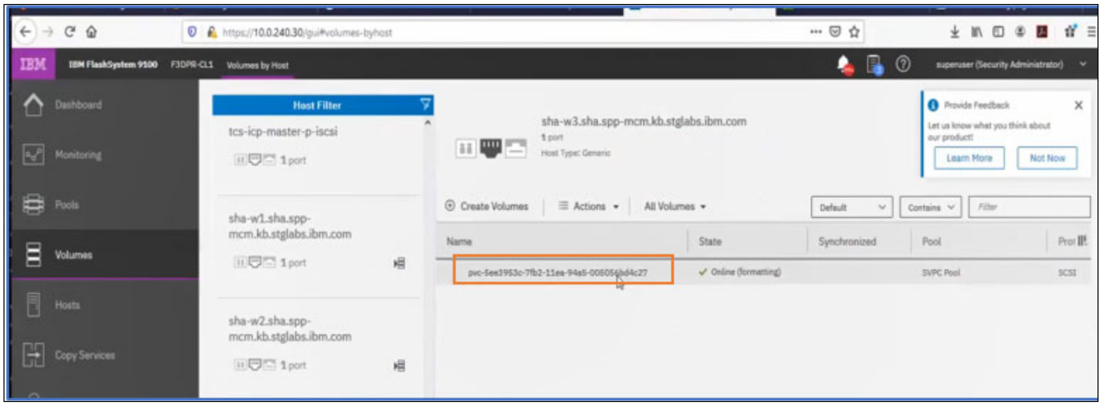


Figure 50 MySQL volume name on the on-premises FlashSystem storage

3. Check the status and progress of the LUN copy.
4. Break the global mirror session and open the LUN in read/write mode.

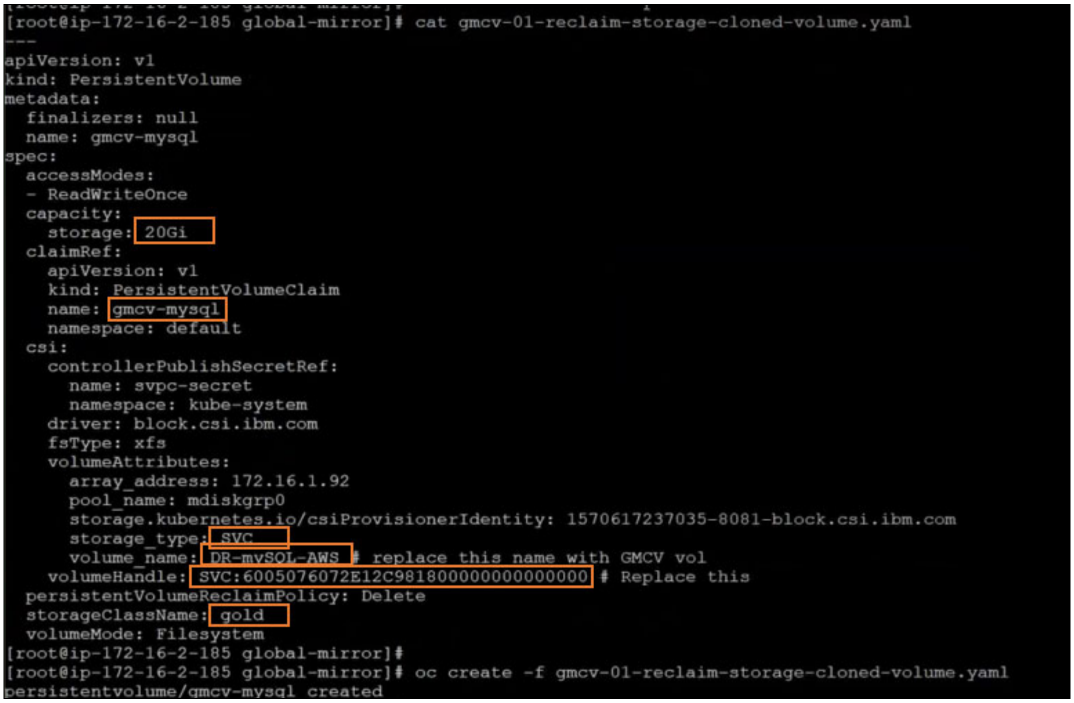
Complete the following steps to use the replicated MySQL storage LUN and check the status MySQL database and tables. The replicated LUN should include the data (that is, five rows):

1. Log in to the installer node (`172.16.2.185`) of the OCP that is installed on AWS and run the commands that are shown in Figure 51 - Figure 61 on page 35. The `yaml` file for deploying MySQL is shown in Figure 51.

```
[root@ip-172-16-2-185 global-mirror]# ls -l
total 16
-rw-r--r--. 1 root root 1243 Apr 20 11:33 commands
-rw-r--r--. 1 root root  827 Apr 20 11:00 gmcv-01-reclaim-storage-cloned-volume.yaml
-rw-r--r--. 1 root root  206 Apr 20 07:02 gmcv-02-pvc-defination-for-reclaimPV.yaml
-rw-r--r--. 1 root root  882 Apr 20 07:04 gmcv-03-deployemnt-on-cloned-volume.yaml
[root@ip-172-16-2-185 global-mirror]# 
```

Figure 51 The yaml file for deploying MySQL

The replicated PVC volume on SVPC is shown in Figure 52.

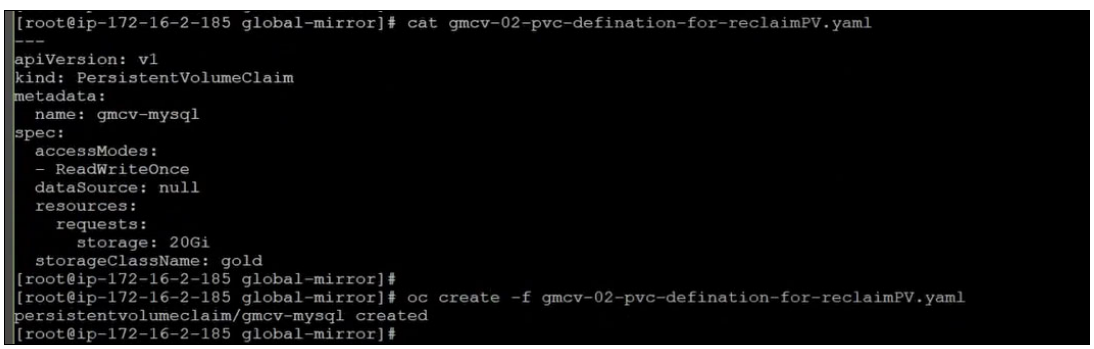


```
[root@ip-172-16-2-185 global-mirror]# cat gmcv-01-reclaim-storage-cloned-volume.yaml
---
apiVersion: v1
kind: PersistentVolume
metadata:
  finalizers: null
  name: gmcv-mysql
spec:
  accessModes:
  - ReadWriteOnce
  capacity:
    storage: 20Gi
  claimRef:
    apiVersion: v1
    kind: PersistentVolumeClaim
    name: gmcv-mysql
    namespace: default
  csi:
    controllerPublishSecretRef:
      name: svpc-secret
      namespace: kube-system
    driver: block.csi.ibm.com
    fsType: xfs
    volumeAttributes:
      array_address: 172.16.1.92
      pool_name: mdiskgrp0
      storage.kubernetes.io/csiProvisionerIdentity: 1570617237035-8081-block.csi.ibm.com
      storage_type: SVC
      volume_name: DR-mySQL-AWS # replace this name with GMCV vol
    volumeHandle: SVC:6005076072E12C9818000000000000000 # Replace this
  persistentVolumeReclaimPolicy: Delete
  storageClassName: gold
  volumeMode: Filesystem
[root@ip-172-16-2-185 global-mirror]#
[root@ip-172-16-2-185 global-mirror]# oc create -f gmcv-01-reclaim-storage-cloned-volume.yaml
persistentvolume/gmcv-mysql created
```

Figure 52 Replicated PVC volume on SVPC

2. Modify the `Volume_name` and `volumeHandle` as needed. Post storage volume replication is completed on the SVPC storage volume (LUN). Identify the volume name and volume handle for the LUN that you created (see Figure 48 on page 30). Modify the sample `yaml` file:

   `gmcv-01-reclaim-storage-cloned-volume.yaml`



```
[root@ip-172-16-2-185 global-mirror]# cat gmcv-02-pvc-defination-for-reclaimPV.yaml
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: gmcv-mysql
spec:
  accessModes:
  - ReadWriteOnce
  dataSource: null
  resources:
    requests:
      storage: 20Gi
  storageClassName: gold
[root@ip-172-16-2-185 global-mirror]#
[root@ip-172-16-2-185 global-mirror]# oc create -f gmcv-02-pvc-defination-for-reclaimPV.yaml
persistentvolumeclaim/gmcv-mysql created
[root@ip-172-16-2-185 global-mirror]#
```

Figure 53 PVC creation

The MySQL deployment with the replicated volume is shown in Figure 54.

```
[root@ip-172-16-2-185 global-mirror]# cat gmcv-03-deployemnt-on-cloned-volume.yaml
---
apiVersion: v1
kind: Service
metadata:
  name: gmcv-mysql
spec:
  ports:
  - port: 3306
  selector:
    app: gmcv-mysql
  clusterIP: None
---
apiVersion: apps/v1 # for versions before 1.9.0 use apps/v1beta2
kind: Deployment
metadata:
  name: gmcv-mysql
spec:
  selector:
    matchLabels:
      app: gmcv-mysql
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        app: gmcv-mysql
    spec:
      containers:
      - image: mysql:5.6
        name: gmcv-mysql
        env:
          # Use secret in real usage
        - name: MYSQL_ROOT_PASSWORD
          value: password
        ports:
        - containerPort: 3306
          name: mysql
        volumeMounts:
        - name: mysql-persistent-storage
          mountPath: /var/lib/mysql
      volumes:
      - name: mysql-persistent-storage
        persistentVolumeClaim:
          claimName: gmcv-mysql
[root@ip-172-16-2-185 global-mirror]#
```

Figure 54 MySQL deployment with the replicated volume

The status of the pod creation process is shown in Figure 55.

```
[root@ip-172-16-2-185 global-mirror]# oc create -f gmcv-03-deployemnt-on-cloned-volume.yaml
service/gmcv-mysql created
deployment.apps/gmcv-mysql created
[root@ip-172-16-2-185 global-mirror]#
```

```
[root@ip-172-16-2-185 global-mirror]# oc get pod
NAME                                        READY   STATUS              RESTARTS   AGE
gmcv-mysql-6d75cf5b9c-v2sn6                 0/1     ContainerCreating   0          30s
ibm-block-csi-controller-0                  4/4     Running             0          3d16h
ibm-block-csi-node-44cr8                    3/3     Running             0          3d16h
ibm-block-csi-node-mbqqm                    3/3     Running             0          3d16h
ibm-block-csi-node-pdgm6                    3/3     Running             0          3d16h
ibm-block-csi-operator-6f5cdc875c-xmq2c     1/1     Running             0          3d16h
mysql-dbb9f67f9-ltb6z                       1/1     Running             0          3d5h
[root@ip-172-16-2-185 global-mirror]#
```

Figure 55 Status of pod creation

The successful creation of the MySQL pod and that it is in a running state is shown in Figure 56.

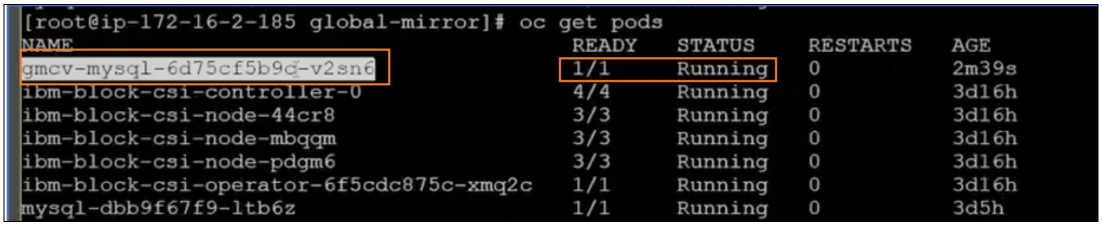


Figure 56 Successful creation of MySQL pod

3. Now that the execution of the `yaml` files is complete, check the status of MySQL and the data on the replicated LUN. Figure 57 - Figure 61 on page 35 show the steps to check the status of MySQL and the data availability on the replicated LUN.

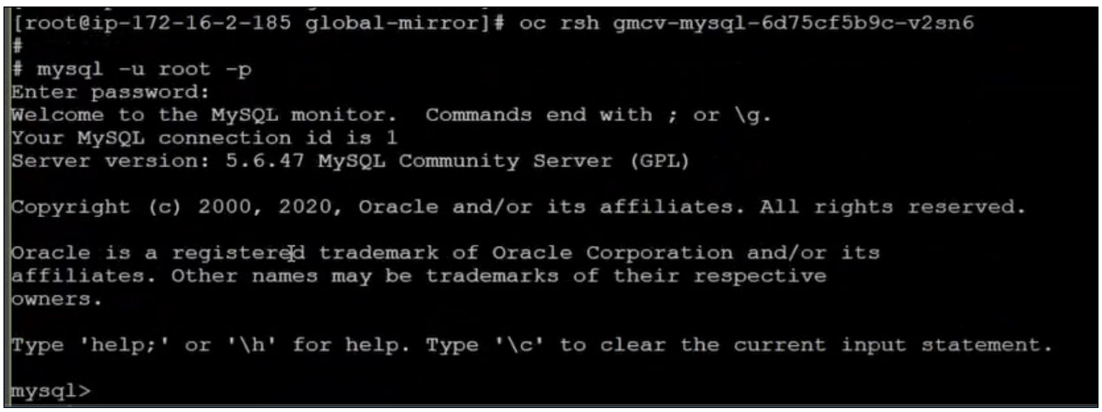


Figure 57 MySQL pod login

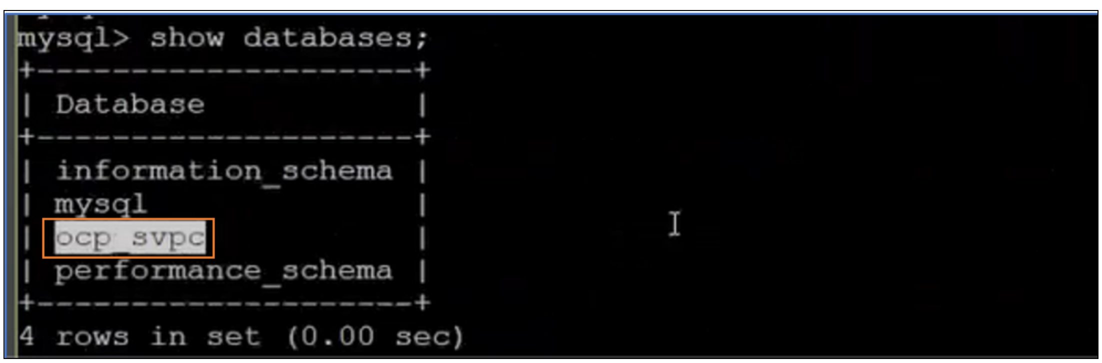


Figure 58 MySQL database

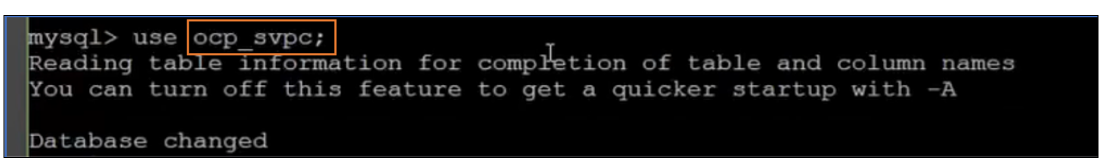


Figure 59 MySQL database

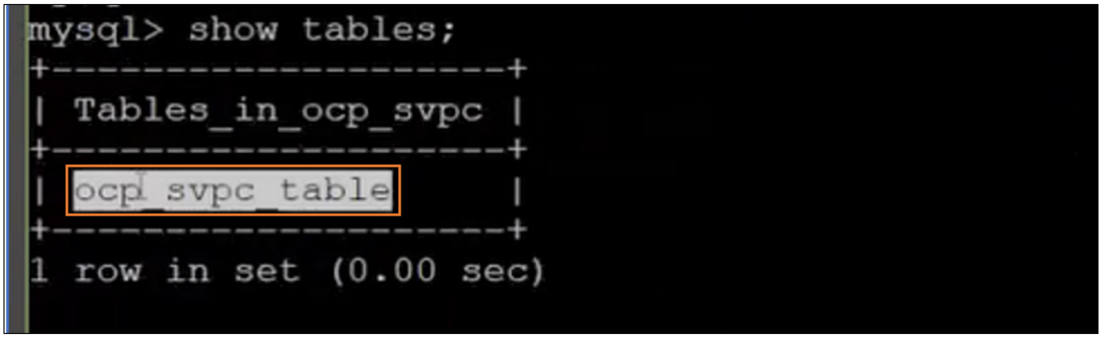


Figure 60 MySQL table space

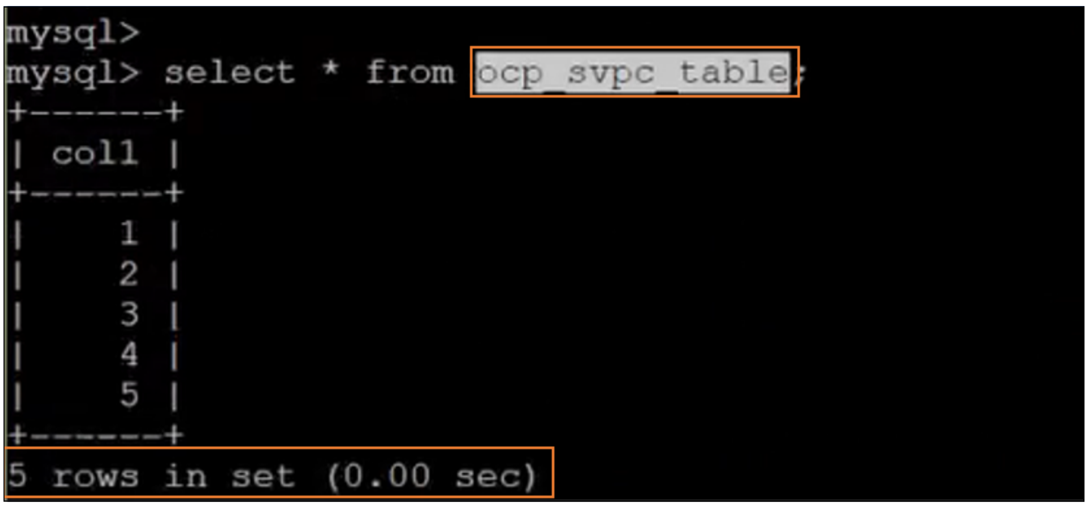


Figure 61 Availability of replicated volume

As described in "Demonstration 4: Hybrid multicloud business continuity" on page 19, the team replicated the data from On-premise IBM FlashSystem storage to SVPC storage that is hosted in AWS Cloud. A sample MySQL deployment was done on OpenShift cluster by using IBM Block Storage CSI driver to cater for database storage needs. For the replication, the test team used Global Mirror functionality that is provided by IBM FlashSystems.

The steps in this demonstration show how the on-premise data can be made available to remote sites and public clouds by using the components that are described in this Blueprint.

For more information about steps that can be taken to ensure database data consistency, see the specific product documentation.

Summary

This Blueprint shows the data availability and Disaster Recovery (DR) demonstrations for the hybrid multicloud environment that is created in an on-premises OCP 4.x environment. This environment includes an IBM Block Storage CSI driver that communicates with the on-premises FlashSystem storage.

The DR site is considered on AWS and installed with OCP 4.x environment. The IBM Block Storage CSI driver communicates with IBM Spectrum Virtualize for Public loud on AWS.

With IBM Spectrum Virtualize for Public Cloud, customers can optimize their heterogeneous storage infrastructure and plan for hybrid-cloud DR between on-premises and Amazon Elastic Block Storage for the containerized environments.

IBM Storage Global mirror feature is used to replicate the data between on-premises FlashSystem storage and IBM Spectrum Virtualize for Public Cloud on AWS.

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, MD-NC119, Armonk, NY 10504-1785, US

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Trademarks

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Privacy policy considerations

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at http://www.ibm.com/privacy and IBM's Online Privacy Statement at http://www.ibm.com/privacy/details in the section entitled "Cookies, Web Beacons and Other Technologies," and the "IBM Software Products and Software-as-a-Service Privacy Statement" at http://www.ibm.com/software/info/product-privacy.

ISBN 0738459038

REDP-5607-00