IBM® Storage

Business Continuity Orchestration for IBM FlashSystem Hybrid Cloud with Red Hat Ansible V1R2

© Copyright International Business Machines Corporation 2021.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

© Copyright IBM Corp. 2021.

About this document

This document is intended to facilitate the deployment of Red Hat Ansible for IBM® FlashSystem. The document showcases the automation and orchestration of storage provisioning and copy services, such as Global Mirror, Global Mirror with change volumes, Metro Mirror for IBM FlashSystem® in hybrid cloud environments. The Ansible-based automation and orchestration for managing high availability environments by using HyperSwap for IBM FlashSystem storage also is described.

To complete the tasks that are described, you must understand IBM FlashSystem and Red Hat Ansible.

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 FlashSystem storage devices are supported and entitled and where the issues are specific to a blueprint implementation.

© Copyright IBM Corp. 2021.

Executive summary

In today's world with the speed, scale, and complexity of hybrid cloud and even traditional on-premises environments, automation is a top priority. IBM FlashSystem family for Hybrid cloud includes Ansible integration. This integration allows IT to create an Ansible playbook that automates the tasks that are repeated across an organization in a consistent way, which improves outcomes and reducing risk. It also standardizes how IT and application owners interact.

Automation is becoming an integral part these days, with IT environments that are complex and must scale up or down quickly for system administrators and developers to meet business requirements. Ansible is becoming popular for orchestration automation because of following reasons:

- It is simple to set up and use with no special skills required to create Ansible playbooks.
- You can customize your environment based on your needs in on-premises or hybrid cloud environments, which gives it agility and flexibility.
- No other software or agents need to be installed on the client system that you want to automate.
- With Ansible and IBM Storage, clients can easily use cutting-edge technology by automating tasks, such as configuration management, provisioning, workflow orchestration, application deployment, and lifecycle management.
- By using Ansible and IBM Storage, clients can reduce system inconsistencies with the automation modules.
- Ansible can also be used to configure end-to-end infrastructure in an orchestrated fashion.
- Ansible provides a single pane of glass visibility to multi-cluster, hybrid cloud environments. Lines of business can use those playbooks to meet their goals without needing to understand the details of how the work is done.

IBM is a Red Hat Certified Support Module Vendor. It provides simple management for IBM FlashSystem and IBM Spectrum® Virtualize.

Scope

This blueprint guide provides:

- A solutions architecture and related solution configuration workflows, with the following essential software and hardware components:
  - IBM FlashSystem
  - IBM Spectrum Virtualize for Public Cloud on IBM Cloud® or AWS
  - Red Hat Ansible 2.9 or higher
- Detailed technical configuration steps for building the Ansible playbooks

This technical report does not:

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

Prerequisites

This technical paper assumes that the user has basic knowledge of the following products:

- IBM FlashSystem
- IBM Spectrum Virtualize for Public Cloud on AWS and IBM Cloud
- Red Hat Ansible

Getting started: Automation using Ansible for IBM FlashSystem

This section describes the essential building material for running Ansible playbooks and automating and orchestrating tasks for IBM FlashSystem by using Ansible.

IBM FlashSystem family

IBM FlashSystem family is an excellent platform to simplify your hybrid cloud storage.

The new IBM FlashSystem family simplifies storage for hybrid cloud environments. With a unified set of software, tools, and APIs, IBM FlashSystem addresses the entire range of storage needs, all from one data platform that extends enterprise functions across over 500 heterogeneous storage network environments.

With IBM Spectrum Virtualize software, the IBM FlashSystem family is an industry-leading storage solution that includes technologies that complement and enhance virtual environments to achieve a simpler, more scalable, and cost-efficient IT infrastructure.

To further drive your IT transformation, IBM Spectrum Virtualize for Public Cloud offers multiple ways to create hybrid cloud solutions between on-premises private clouds and the public cloud. It enables real-time storage-based data replication and disaster recovery, and data migration between local storage and AWS. This replication enables storage administration at a cloud service provider's site in the same way as on-premises, regardless of the type of storage.

For more information about the IBM FlashSystem, see *IBM FlashSystem family: High-performance, highly functional solutions that make hybrid cloud storage simple for every enterprise*.

Consider the following points regarding IBM FlashSystem storage solutions:

- NVMe-accelerated flash arrays with control enclosures that are end-to-end NVMe-enabled, with flexibility to choose and mix between IBM FlashCore® Modules, industry standard NVMe drives and Storage-Class Memory. The systems offer industry-leading performance and scalability with support for bare-metal, virtual, and containerized environments.
- Built with IBM Spectrum Virtualize, with a full range of industry-leading data services such as dynamic tiering, IBM FlashCopy® management, data mobility, and high-performance data encryption, among many other data management features
- Hybrid cloud ready, with support for private, hybrid, or public cloud deployments. The solutions include ready-to-use, proven, validated "cloud blueprints" with support for cloud API automation and secondary data orchestration software.
- Cost-efficient, with innovative data reduction pool (DRP) technology that includes deduplication and hardware-accelerated compression technology, plus SCSI UNMAP support and all the thin provisioning, copy management, and efficiency you'd expect from IBM Spectrum Virtualize based storage.
- Hybrid storage enabled, with multiple expansion enclosure options based on 12 Gbps SAS that supports solid-state drives (SSD) and hard disk drives (HDD).
- IBM FlashSystem® is ready for new generation applications, and supports Red Hat OpenShift, Container Storage Interface (CSI), Ansible automation, and Kubernetes along with traditional VMWare and bare metal environments.

- IBM Cloud Satellite™ helps you deploy consistently across all on-premises, edge computing and public cloud environments from any cloud vendor. The result is greater developer productivity and development velocity. IBM FlashSystem family is the perfect storage choice for IBM Cloud Satellite because of its simplicity, high performance, and low latency.

As a Red Hat Certified Support Module Vendor, IBM provides simple management for the storage provisioning commands that are used in the IBM Spectrum® Virtualize Ansible Collection. For more information, see this web page.

These capabilities include:

- Basic Automation and Orchestration for storage provisioning:
  - Collect facts: Gather array information, such as hosts, host groups, snapshots, consistency groups, and volumes
  - Manage hosts: Create, delete, or modify hosts
  - Manage volumes: Create, delete, or extend the capacity of volumes
  - Manage MDisk: Create or delete a managed disk
  - Manage Pool: Create or delete a pool (managed disk group)
  - Manage Volume Map: Create or delete a volume map
- Advance Automation and Orchestration for Snapshot management:
  - Manage Snapshots: Create, delete, start, and stop snapshot mappings
  - Manage Snapshot consistency groups: Create, Delete, start, and stop consistency groups
  - Manage Clones: Create, Delete, start, stop clone mappings
- Advance Automation and Orchestration for Remote copy management for FlashSystem:
  - Manage Metro Mirror, Global Mirror, and GMCV relationships: Create, Delete, Modify relationships
  - Manage Metro Mirror, Global Mirror, and GMCV Consistency Groups: Create, Delete, Modify Consistency Groups
  - Start or Stop relationships or consistency groups
  - Create Change Volumes by using the provided base volume
  - Create and manage HyperSwap® provisioned volumes

The same Ansible modules can be used for automating IBM® Spectrum Virtualize for Public Cloud in AWS because it is one code base for IBM FlashSystem Storage on premises and IBM Spectrum Virtualize for Public Cloud software that is running in public clouds, such as AWS and IBM Cloud®.

IBM Spectrum Virtualize for Public Cloud in AWS

IBM Spectrum Virtualize for Public Cloud is a version of IBM Spectrum Virtualize implemented in a cloud environment.

Designed for public cloud IaaS, IBM Spectrum Virtualize for Public Cloud is a solution for public cloud implementations and includes technologies that complement and enhance public cloud IaaS offering capabilities.

IBM Spectrum Virtualize for Public Cloud enables deployment of IBM Spectrum Virtualize-based software in public clouds: IBM Cloud and AWS. IBM Spectrum Virtualize for Public Cloud on AWS is a BYOL (Bring your Own License) offering that can be purchased as a perpetual or a monthly license.

IBM Spectrum Virtualize for Public Cloud (see Table 1) can be deployed on AWS IaaS by way of the AWS Marketplace to enable hybrid cloud solutions. It offers the ability to transfer data between on-premises data centers by using any IBM Spectrum Virtualize-based system and AWS.

Table 1 IBM Spectrum Virtualize for Public Cloud at a glance

Storage supported	AWS EBS block storage
Licensing approach	Simple, flat cost per managed Terabyte monthly licensing, or perpetual licensing
Platform	IBM Spectrum Virtualize for Public Cloud on AWS installed on supported EC2 instance

Ansible Playbook: Sample configuration architecture

Figure 1 shows the architecture that was deployed in lab to showcase the orchestration and automation of storage replication by using Global Mirror with change volume (GMCV) with a consistency group.

In this sample configuration, two IBM FlashSystem 9100 units are used, which are designated as FlashSystem #1 and #2.

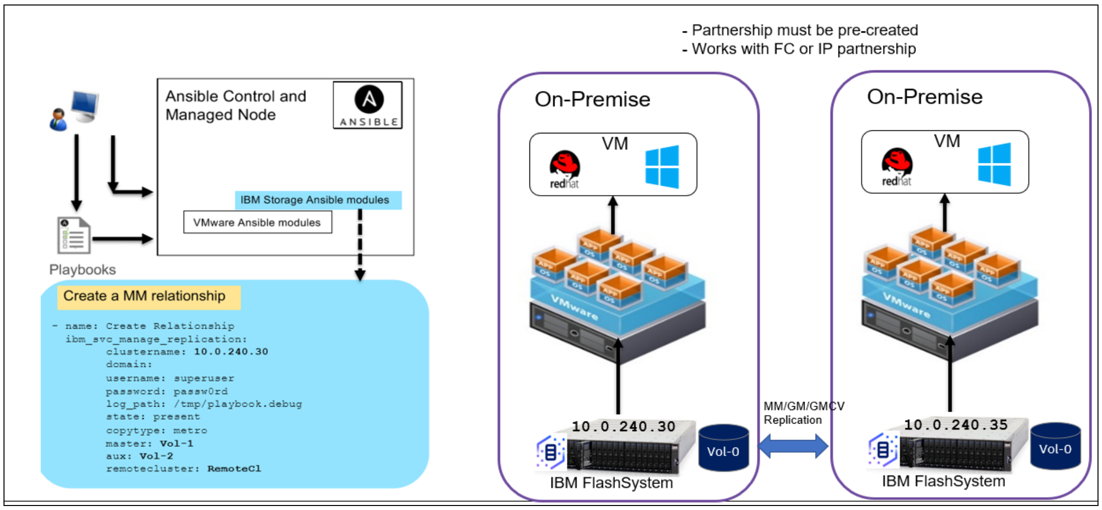

Figure 1 Sample configuration architecture

The sample configuration demonstrates the following automation steps of using IBM Spectrum Virtualize Ansible modules:

1. Create a VDisk on IBM FlashSystem.
2. Create a consistency group.
3. Create the GMCV relationship.
4. Start and stop the GMCV replication.

Figure 2 shows the architecture of the use of the same Ansible modules for replication between on-premises FlashSystem storage and the IBM Spectrum Virtualize instance running in AWS cloud by using IBM Spectrum Virtualize native IP replication capabilities.

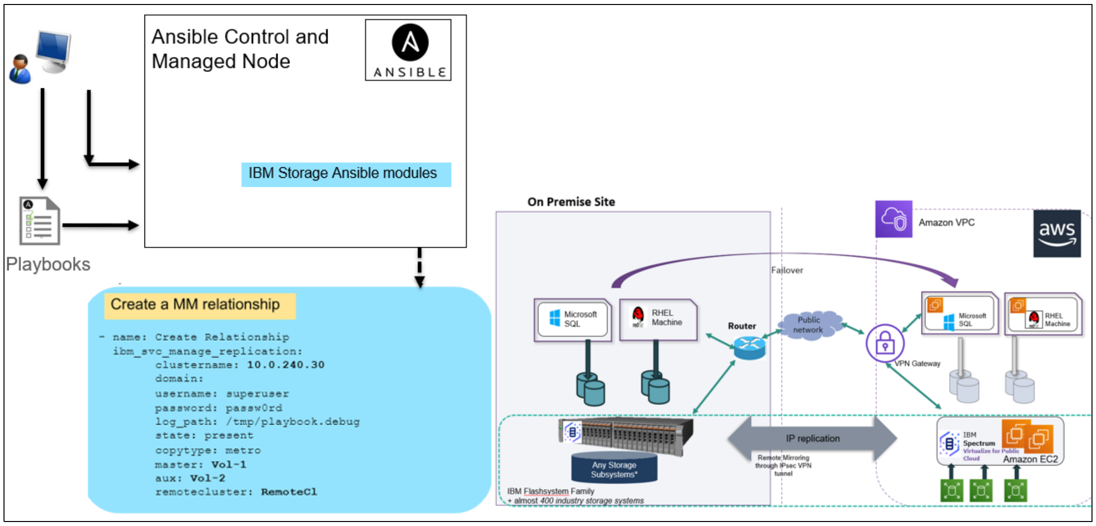

Figure 2   Sample configuration for hybrid cloud Ansible automation for replication

IBM FlashSystem HyperSwap is the high availability (HA) solution for IBM FlashSystem that provides continuous data availability in case of hardware, power, or connectivity failures, or disasters. A sample HyperSwap environment configuration is shown in Figure 3.

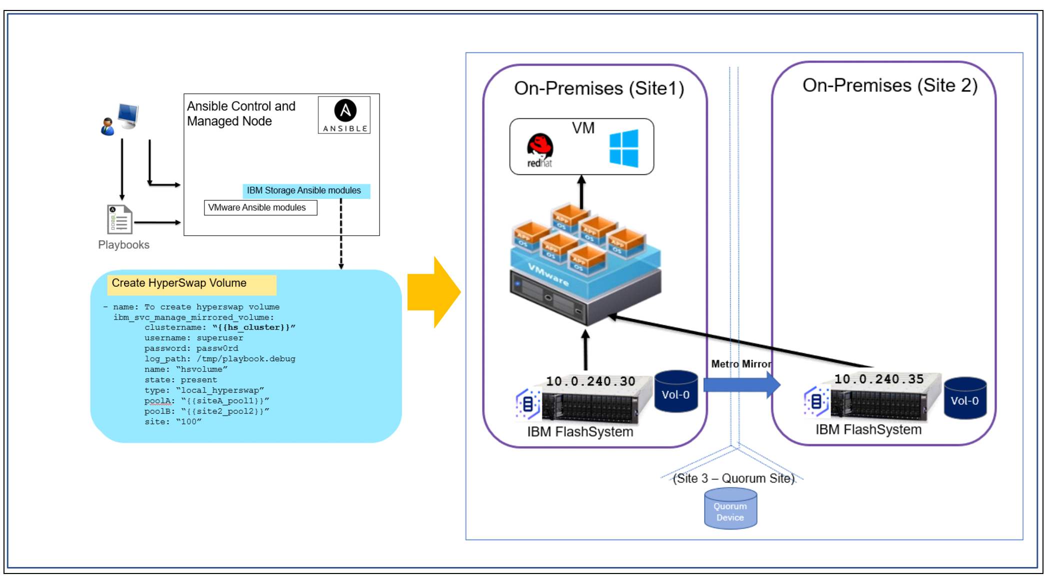

Figure 3   Sample configuration for HyperSwap environment

Before version 7.5, the only available data protection feature on IBM FlashSystem was the Volume Mirroring feature that provides data availability in a failure of internal or external storage. However, the Volume Mirroring feature does not provide any protection in a loss of the FlashSystem cluster or control enclosures.

To protect data against the complete loss of storage systems, host-based data availability solutions can be implemented. These solutions rely on a combination of storage system and application or operating system capabilities. Often, they delegate the management of the storage loss events to the host.

The IBM FlashSystem Version 7.5 introduced the IBM HyperSwap technology that provides an HA solution (which is transparent to the host) between two locations at up 300 km (186.4 miles) apart. A new Metro Mirror capability, the active-active Metro Mirror, is used to maintain a fully independent copy of the data at each site. When data is written by hosts at either site, both copies are synchronously updated before the write operation is completed. The HyperSwap function automatically optimizes to minimize the data that is transmitted between sites and to minimize host read and write latency.

Only HyperSwap volume management can be done using by Ansible modules.

For more information about HyperSwap Implementation and best practices for building a HyperSwap environment, see the following publications:

- *IBM Spectrum Virtualize HyperSwap SAN Implementation and Design Best Practices*, REDP-5597
- *IBM Storwize V7000, Spectrum Virtualize, HyperSwap, and VMware Implementation*, SG24-8317

With Red Hat Ansible modules `ibm_svc_manage_mirrored_volume` for IBM Spectrum Virtualize, the HyperSwap Volume can be created by using Ansible playbooks.

Configuring and installing Ansible and IBM Spectrum Virtualize modules

This section describes the configuration and deployment of Ansible.

Ansible is an agentless automation management tool that uses the SSH protocol. Ansible can be run from any machine with Python 2 (version 2.7) or Python 3 (versions 3.5 and higher) installed, including Red Hat, Debian, CentOS, macOS, and any of the BSDs.

Note: Windows is not supported for the control node.

For more information about installing Ansible for each operating system, see this web page.

For IBM Spectrum Virtualize modules, Ansible version 2.9 or higher is required.

For more information about installing the IBM Spectrum Virtualize modules, see this web page.

After the IBM Spectrum Virtualize modules are installed, the tree structure of the collection looks as shown in Figure 4.

```
[root@vm09-ansible collections]# cd ibm
[root@vm09-ansible ibm]# tree
.
└── spectrum_virtualize
    ├── galaxy.yml
    ├── playbooks
    │   ├── 1
    │   ├── createpool-FOCN.yaml
    │   ├── createpool.yaml
    │   ├── create_vdisk.yaml
    │   ├── deletepool-FOCN.yaml
    │   ├── deletepool.yaml
    │   ├── gather_info_collections-FOCN.yaml
    │   └── gather_info_collections.yaml
    ├── plugins
    │   ├── modules
    │   │   ├── ibm_svc_host.py
    │   │   ├── ibm_svc_info.py
    │   │   ├── ibm_svc_mdiskgrp.py
    │   │   ├── ibm_svc_mdisk.py
    │   │   ├── ibm_svc_vdisk.py
    │   │   ├── ibm_svc_vol_map.py
    │   │   └── __init__.py
    │   ├── module_utils
    │   │   └── ibm_svc_utils.py
    │   └── README.md
    └── README.md
```

Figure 4   Collection tree structure

Creating an Ansible playbook for replication using Global Mirror and Metro Mirror

This section describes how to create the Ansible playbook for an IBM Spectrum Virtualize system for replication by using Global Mirror and Global Mirror with Change Volume and Metro Mirror.

The section also provides a sample playbook for creating a HyperSwap volume.

An Ansible playbook is an organized unit of scripts that defines work or tasks for managing infrastructure (in this case IBM Spectrum Virtualize by the automation tool. Ansible plays are written in yaml.

The playbook is the core component of any Ansible configuration. An Ansible playbook contains one or multiple plays, each of which defines a task to be done for a configuration on a managed storage system.

In the solution lab test environment, the sample playbook for GMCV between two FlashSystem 9100s with a consistency group is shown in Figure 5 on page 13.

This single playbook that is shown consists of the following parts:

- The first part is to create a VDisk on IBM FlashSystem that is in the sample playbook (referred to as `FlashSystem_1` and `FlashSystem_2`) and create a consistency group.
- The second part of the playbook creates a GMCV relationship between the master and auxiliary volumes. It also creates a master change volume and auxiliary change volume.

The `.vars` file features all of the information about the username, password, and IP address of the storage system that is used, which are called *variables* in the playbook.

```
---
- name: Using IBM Spectrum Virtualize collection to create vdisk and map to host
  hosts: localhost
  vars_files:
    /Hemanand/vars.yaml
  collections:
    - ibm.spectrum_virtualize
  gather_facts: no
  connection: local
  tasks:
    - name: make vdisk Flashsystem_1
      ibm_svc_vdisk:
        clustername: "{{Flashsystem_1IP}}"
        domain:
        username: "{{Flashsystem_1Username}}"
        password: "{{Flashsystem_1password}}"
        log_path: /root/.ansible/playbook.debug
        name: "vol_{{inventory_hostname}}_Master"
        state: present
        mdiskgrp: SVPC Pool
        easytier: 'off'
        size: "5"
        unit: gb
    - name: make vdisk Flashsystem_2
      ibm_svc_vdisk:
        clustername: "{{Flashsystem_2IP}}"
        domain:
        username: "{{Flashsystem_2Username}}"
        password: "{{Flashsystem_2password}}"
        log_path: /root/.ansible/playbook.debug
        name: "vol_{{inventory_2hostname}}_AUX"
        state: present
        mdiskgrp: DRP_POOL
        easytier: 'off'
        size: "5"
        unit: gb
    - name: create remote copy cg
      ibm_svc_manage_replicationgroup:
        name: "cg1"
        clustername: "{{Flashsystem_1IP}}"
        username: "{{Flashsystem_1Username}}"
        password: "{{Flashsystem_1password}}"
        log_path: /root/.ansible/playbook.debug
        state: present
        remotecluster: "{{remote_flashsystem_2}}"

    - name: make Remote Replication Relationship
      ibm_svc_manage_replication:
        name: gmcv
        clustername: "{{Flashsystem_1IP}}"
        username: "{{Flashsystem_1Username}}"
        password: "{{Flashsystem_1password}}"
        log_path: /root/.ansible/playbook.debug
        state: present
        remotecluster: "{{remote_flashsystem_2}}"
        master: "vol_{{inventory_hostname}}_Master"
        aux: "vol_{{inventory_2hostname}}_AUX"
        copytype: GMCV
        consistgrp: cg1
        sync: true
    - name: create master change volume
      ibm_svc_manage_cv:
        state: present
        basevolume: "vol_{{inventory_hostname}}_Master"
        cvname: masterchange_cv
        rname: gmcv
        ismaster: true
        clustername: "{{Flashsystem_1IP}}"
        username: "{{Flashsystem_1Username}}"
        password: "{{Flashsystem_1password}}"
        log_path: /tmp/ansible.log
    - name: create auxiliary change volume
      ibm_svc_manage_cv:
        state: present
        basevolume: "vol_{{inventory_2hostname}}_AUX"
        cvname: auxchange_cv
        rname: gmcv
        ismaster: false
        clustername: "{{Flashsystem_2IP}}"
        username: "{{Flashsystem_2Username}}"
        password: "{{Flashsystem_2password}}"
        log_path: /tmp/ansible.log
```

Figure 5 Sample Ansible playbook for GMCV replication between two IBM FlashSystem storages

After the playbook is run successfully, the volumes are created with the specified names and a GMCV relationship is established with the associated change volumes, as shown in Figure 6. The volume that is created on IBM FlashSystem 9100 is referred as `FlashSystem_1` in the Ansible playbook.

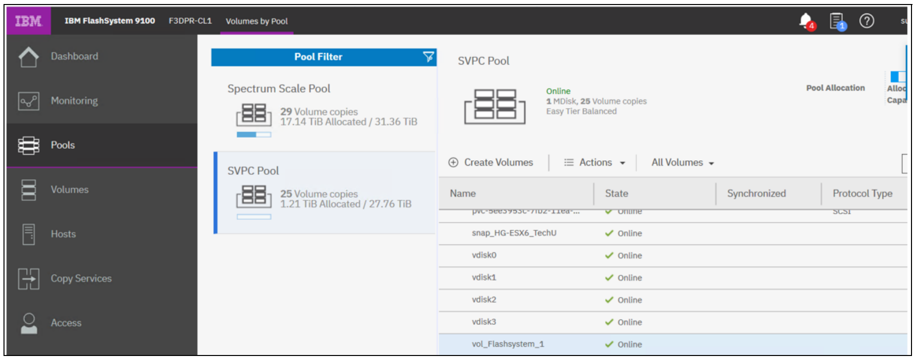

Figure 6 Volume creation on primary IBM FlashSystem 9100 by using Ansible playbook

Figure 7 shows the volume creation activity that is completed in the DRP_POOL on IBM FlashSystem 9100, which is referred as `FlashSystem_2` volume creation.

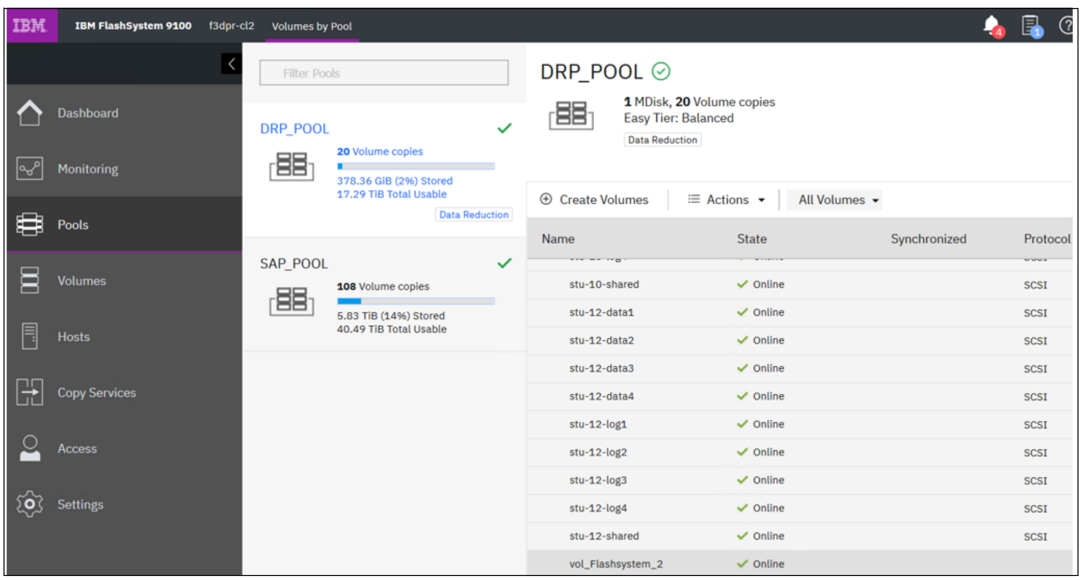

Figure 7 Volume creation on secondary IBM FlashSystem 9100 by using Ansible playbook

The GMCV relationship between the master and auxiliary volumes are created between two IBM FlashSystem 9100 storages with the consistency group name `gmcv`, as referenced in the Ansible playbook.

The GMCV relationship is stopped, as shown in Figure 8. To start the replication, use the Ansible module for starting and stopping the consistency group replication process.

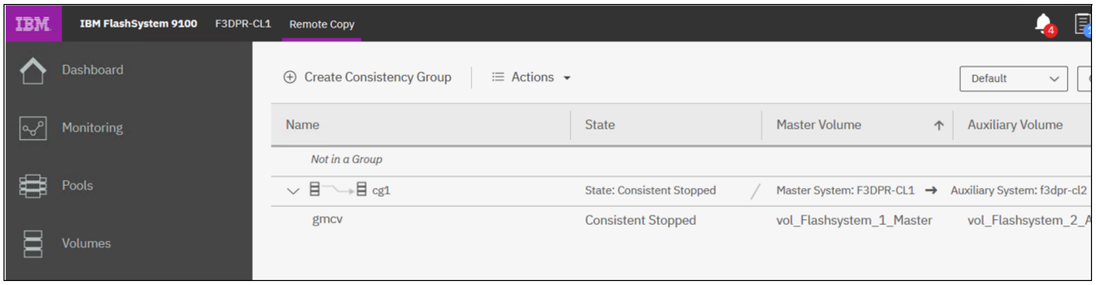

Figure 8 Consistency group creation

After the consistency group is created, the change volume that was created for the master volume and auxiliary volume by using the Ansible module `ibm_svc_manage_cv` creates change volumes for the master and auxiliary base volumes, as shown in Figure 9.

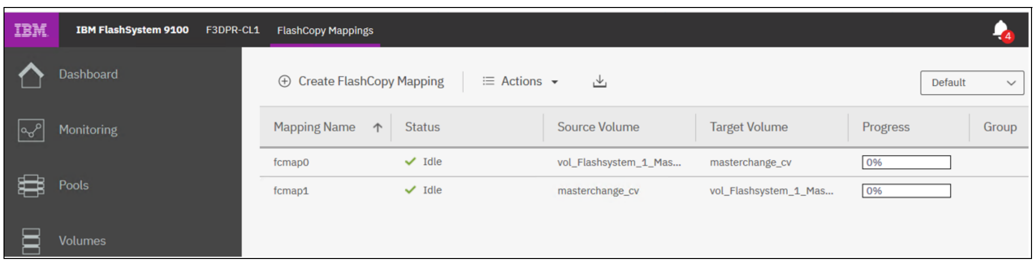

Figure 9 Figure 8: Change volume creation

Similar to the sample playbook that is shown in Figure 5 on page 13, the sample playbook for creating the Global Mirror relationship between IBM FlashSystem 9100 on-premises and IBM Spectrum Virtualize for Public Cloud on an AWS instance is shown in Figure 10.

```
---
- name: Using IBM Spectrum Virtualize collection to create vdisk and
  hosts: localhost
  vars_files:
    /root/.ansible/HG_playbooks/vars.yaml
  collections:
    - ibm.spectrum_virtualize
  gather_facts: no
  connection: local
  tasks:
    - name: make vdisk cloud
      ibm_svc_vdisk:
        clustername: "{{SVC_Cloud_IP}}"
        domain:
        username: "{{SVC_Cloud_Username}}"
        password: "{{SVC_Cloud_password}}"
        log_path: /root/.ansible/playbook.debug
        name: "vol_Cloud_{{inventory_cloud_hostname}}_TechU"
        state: present
        mdiskgrp: mdiskgrp0
        easytier: 'off'
        size: "10"
        unit: gb
      environment: "{{proxy_env}}"
    - name: map volume to host cloud
      ibm_svc_vol_map:
        clustername: "{{SVC_Cloud_IP}}"
        domain:
        username: "{{SVC_Cloud_Username}}"
        password: "{{SVC_Cloud_password}}"
        log_path: /root/.ansible/playbook.debug
        volname: "vol_Cloud_{{inventory_cloud_hostname}}_TechU"
        host: "{{inventory_cloud_hostname}}"
        state: present
      environment: "{{proxy_env}}"

    - name: make vdisk onprem
      ibm_svc_vdisk:
        clustername: "{{SVC_IP}}"
        domain:
        username: "{{SVC_Username}}"
        password: "{{SVC_password}}"
        log_path: /root/.ansible/playbook.debug
        name: "vol_{{inventory_hostname}}_TechU"
        state: present
        mdiskgrp: SVPC Pool
        easytier: 'off'
        size: "10"
        unit: gb
    - name: map volume to host onprem
      ibm_svc_vol_map:
        clustername: "{{SVC_IP}}"
        domain:
        username: "{{SVC_Username}}"
        password: "{{SVC_password}}"
        log_path: /root/.ansible/playbook.debug
        volname: "vol_{{inventory_hostname}}_TechU"
        host: "{{inventory_hostname}}"
        state: present
    - name: Create Relationship 1
      ibm_svc_manage_replication:
        name: rcopy1
        clustername: "{{onprem_cluster_name}}"
        username: "{{SVC_Username}}"
        password: "{{SVC_password}}"
        log_path: /root/.ansible/playbook.debug
        state: present
        remotecluster: "{{cloud_cluster_name}}"
        master: "vol_{{inventory_hostname}}_TechU"
        aux: "vol_Cloud_{{inventory_cloud_hostname}}_TechU}}"
        copytype: global
        sync: true
      environment: "{{proxy_env}}"
```

Figure 10 Playbook for creating and mapping VDisk and setting up replication

After this yaml file is created, the Ansible playbook is run, as shown in Figure 11.

```
[root@vm09-ansible playbooks]# ansible-playbook create_map_cloud.yaml
[WARNING]: running playbook inside collection ibm.spectrum_virtualize

[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not match 'all'


PLAY [Using IBM Spectrum Virtualize collection to create vdisk and map to host]
*************************************************************************************************************

TASK [make vdisk onprem]
****************************************************************************************************************************
changed: [localhost]

TASK [map volume to host onprem]
****************************************************************************************************************************
changed: [localhost]

TASK [make vdisk cloud]
****************************************************************************************************************************
changed: [localhost]

TASK [map volume to host cloud]
****************************************************************************************************************************
changed: [localhost]

TASK [Create Relationship 1]
****************************************************************************************************************************
changed: [localhost]

PLAY RECAP
****************************************************************************************************************************
localhost                  : ok=5    changed=5    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Figure 11 Execution output of playbook

The Ansible playbook uses the following sequence:

1. Uses the `ibm_svc_vdisk` module and creates a 10 GB volume with the VDisk name that is dictated in the playbook.
2. Runs the second task by using the `ibm_svc_vol_map` module to map the volume to that host that is specified in the playbook.
3. Creates a VDisk with the specified name on IBM Spectrum Virtualize for Public Cloud instance running in AWS.
4. Maps the volume to the EC2 instance running in AWS.
5. After the volume creation and mapping is complete, the playbook sets up the Global Mirror replication relationship and starts the replicating the VDisk from the on-premises FlashSystem 9100 to IBM Spectrum Virtualize for Public Cloud instance running in AWS.

The sample playbook for creating a HyperSwap volume that uses the `ibm_svc-_manage_mirrored volume` Ansible module is shown in Figure 12.

```
---
- name: Using IBM Spectrum Virtualize collection to create vdisk and map to host
  hosts: localhost
  vars_files:
    /Hemanand/vars.yaml
  collections:
    - ibm.spectrum_virtualize
  gather_facts: no
  connection: local
  tasks:
    - name: simple use case where you create a hs volume
      ibm_svc_manage_mirrored_volume:
        clustername: "{{Hyperswap_Cluster}}"
        username: "{{Hyperswap_Username}}"
        password: "{{Hyperswap_password}}"
        log_path: /tmp/playbook.debug
        name: "HG_vol_{{Hyperswap_host}}_DB"
        state: present
        type: "local hyperswap"
        poolA: "SAP"
        poolB: "SAP_SITE2"
        size: "51200"
    - name: map volume to host
      ibm_svc_vol_map:
        clustername: "{{Hyperswap_Cluster}}"
        domain:
        username: "{{Hyperswap_Username}}"
        password: "{{Hyperswap_password}}"
        log_path: /root/.ansible/playbook.debug
        volname: "HG_vol_{{Hyperswap_host}}_DB"
        host: "HG-ESXi-10.0.240.21"
        state: present
```

Figure 12 Sample playbook for HyperSwap Volume creation

In this Ansible playbook, the HyperSwap volume is created with the name `HG_vol_{{Hyperswap_host}}_DB` on the pool name `SAP` on the Site 1 storage system and in `SAP_SITE2` on the Site 2 storage system.

In the same playbook, the volume is mapped to the ESXi host by using `ibm_svc_vol_map` module after this HyperSwap volume is created.

The output that results after this playbook is run is as shown in Figure 13.

```
[root@vm09-ansible Hemanand]# ansible-playbook create_hyperswap.yaml
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not match 'all'


PLAY [Using IBM Spectrum Virtualize collection to create vdisk and map to host] ************************************************


TASK [simple use case where you create a hs volume] ****************************************************************************
**********

ok: [localhost]


TASK [map volume to host] ******************************************************************************************************
************************************

ok: [localhost]


PLAY RECAP *********************************************************************************************************************
****************************************************

localhost                  : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Figure 13 Output after running Ansible playbook for HyperSwap

The playbook creates the HyperSwap volume (as shown in Figure 14) and maps the host as dictated in the playbook.

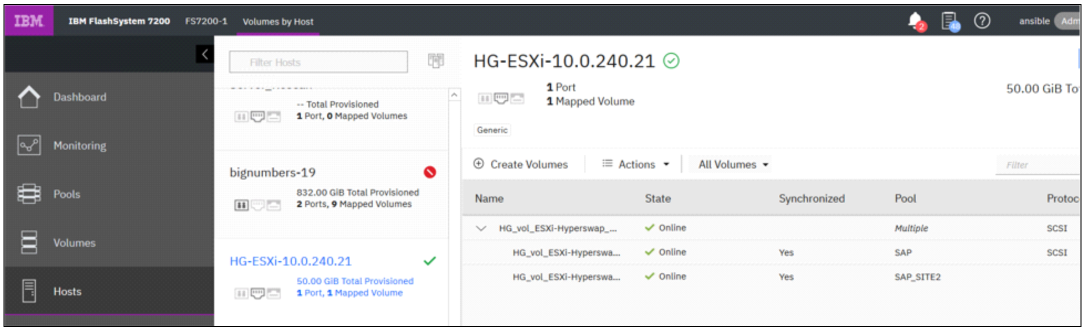

Figure 14 HyperSwap Volume creation and host mapping

The same `ibm_svc_manage_mirrored_volume` Ansible module can be also be used to delete the mirror volume by changing the `state: absent` in the playbook and to create standard mirror volume by changing the `type: "standard"`.

Summary

The deployment of hybrid cloud environments is gaining much attention from organizations. With that attention brings the complexity of managing the on-premises and different cloud environments.

With simplicity in the storage platforms managing complex IT infrastructure, challenges can be addressed by using Orchestration and automation tools, such as Red Hat Ansible. With Red Hat Ansible, support for IBM FlashSystem clients can orchestrate and automate their storage infrastructure management on-premises or in a cloud environment seamlessly with a common tool set.

This solution paper discussed two sample playbooks that use IBM FlashSystem Ansible playbook collections for replicating and orchestrating the management of storage replication between two IBM FlashSystem storages between on-premises data centers.

The paper also showcased the sample playbook that can be used to orchestrate and manage a Global Mirror replication relationship in hybrid cloud environments between IBM FlashSystem 9100 on-premises and IBM Spectrum Virtualize for Public Cloud on AWS.

The paper also provided the sample playbook to create HyperSwap volume on IBM FlashSystem storage systems.

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.

© Copyright IBM Corp. 2021.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at http://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks or registered trademarks of International Business Machines Corporation, and might also be trademarks or registered trademarks in other countries.

FlashCopy®
HyperSwap®
IBM®
IBM Cloud®
IBM Cloud Satellite™
IBM FlashCore®
IBM FlashSystem®
IBM Spectrum®
Redbooks (logo) ®
Satellite™
Storwize®

The following terms are trademarks of other companies:

Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Ansible, OpenShift, Red Hat, are trademarks or registered trademarks of Red Hat, Inc. or its subsidiaries in the United States and other countries.

VMware, and the VMware logo are registered trademarks or trademarks of VMware, Inc. or its subsidiaries in the United States and/or other jurisdictions.

Other company, product, or service names may be trademarks or service marks of others.

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.

© Copyright IBM Corporation

June 2021

US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Please recycle

ISBN 0738459755

REDP-5637-01