EMC ViPR Controller

Version 3.6.1.0

New Features and Changes

302-004-100

REV 01

EMC²

Published August 2017

EMC Corporation
Hopkinton, Massachusetts 01748-9103
1-508-435-1000 In North America 1-866-464-7381
www.EMC.com

CHAPTER 1

New Features

Documentation changes introduced with ViPR Controller version 3.6.1

Future versions of ViPR Controller publications will be updated with the new and changed features introduced with the 3.6 Service Pack 1 release.

Documentation changes affecting the *ViPR Controller CLI Reference Guide*

- ViPR Controller CLI 3.6.1 changes on page 4
- Improved CLI commands for metropoint virtual pools on page 27

Documentation changes affecting *ViPR Controller Ingest Services for Existing Environments* :

- Isilon file enhancements on page 5
- Zone ingestion on page 10

Documentation changes affecting the *ViPR Controller Service Catalog Reference Guide*

- Order failures, rollbacks, and retries on page 25
- Impact of changing virtual pool field values on page 44
- Export service added to all-flash services on page 26
- Create and manage file systems on page 9

Documentation changes affecting the *ViPR Controller User Interface Virtual Data Center Configuration Guide*

- Port allocation enhancements on page 11

Documentation changes affecting the *ViPR Controller Integration with VMAX and VNX Storage Systems Guide*

- Export a VMAX3 resource to multiple compute resources on page 26

Documentation changes affecting the *EMC ViPR Controller Security Configuration Guide*

- Clarification of Disaster Recovery ports 500 and 4500 in the ViPR Controller ports on page 48 topic.

Documentation changes affecting the *ViPR Controller Support for VPLEX and VPLEX with EMC Data Protection User and Administration Guide*

- Block Storage Services > Unexport and Remove Block Volume on page 54

Documentation changes affecting the *ViPR Controller Virtual Data Center Requirements and Information Guide*

- ViPR requirements for service profile templates on page 55
- Inter-VSAN Routing (IVR) support on page 39

ViPR Controller CLI 3.6.1 changes

Table 1 Revision history

Revision Date	Description of change
August 2017	Commands that have been updated:

Table 1 Revision history

Revision Date	Description of change

	• `viprcli filesystem export`
	`viprcli filesystem export -n HomeDirFS -pr HR -pl NFS -sec sys -per rw -endpoint lglw8248.lss.emc.com -bypassdnscheck true`
	• `viprcli filesystem export`
	`viprcli filesystem reduce -n HomeDirFS -pr HR -size 5G`
	• `viprcli vpool create`
	The `metropoint` and `activeProtectionAtHASite` options were added.
	The syntax of the `ha` and `rp_policy` options changed.
	• `viprcli vpool update`
	The syntax of the `rp_policy` option changed.

Isilon file enhancements

This release offers file enhancements for Isilon storage systems.

- Storage systems may be discovered using either the Host IP address or the Fully Qualified Domain Name (FQDN). This allows ViPR Controller discovery of Isilon storage systems when STIG hardening profiles are in use.
  - When using STIG hardening profiles where root SSH use is disabled, ViPR Controller uses the FQDN of the SmartConnect service IP address to discover the cluster. ViPR Controller does not support the 7.x version with hardening profiles. Specify the FQDN when creating the Isilon storage system in the **Physical** > **Storage Systems** UI page.
- You can override client and username conflicts or errors when creating or modifying an export. The **Create File System and NFS Export**, and **Create NFS Export for File System** UI pages provide a flag under the **Advanced** tab to bypass the DNS check.
  - There is also API and CLI support for bypassing the DNS check. Example: `viprcli filesystem export -n HomeDirFS -pr HR -pl NFS -sec sys -per rw -endpoint <FQDN.com> -bypassdnscheck true`
- The **Reduce File System Quota** service provides an operation to reduce the size of an Isilon file system. The size must not be less than the capacity currently in use. Use this service to release a quota to use elsewhere.
  - There is API and CLI support for reducing file system quotas. This example shows a reduction to 5GB: `viprcli filesystem reduce -n HomeDirFS -pr HR -size 5G`

Introduction

System Administrators can incorporate unmanaged file systems under ViPR Controller management by ingesting them. This ingestion includes the subdirectories of the file

systems and their shares and exports. The operation requires two catalog services: a discovery service followed by an ingestion service.

You must first assign these file systems to file virtual pools and ensure these file virtual pools have the same criteria as their corresponding file systems. During discovery, ViPR Controller finds the file systems on a specified array and matches them with the file virtual pools. It discovers the subdirectories, shares, and access control lists associated with the shares and exports. When you run the ingest service, only the discovered file systems belonging to the specified file virtual pool are ingested.

Once under ViPR Controller management, you manage the ingested storage resources by provisioning users in the same way as if you created them in ViPR Controller. You can export them to hosts, expand them, and use snapshot and copy techniques to protect them.

What is supported

- Ingestion of file systems which are exported to hosts.
- Ingestion of filesystems which are not exported.
- Ingestion of file systems with snapshots. (Snapshots are not ingested.)
- Ingestion of file systems with quota directories
- Ingestion of file systems with CIFS shares, CIFS ACLs and NFS ACLs

Limitations

- Isilon only. Ingestion of file systems which have no hard quota assigned is not supported. Isilon allows you to create file systems with no hard quota. In such cases, these file systems can expand to fill up the entire available space on the storage system. ViPR Controller always provisions file systems with a hard quota using the `size` parameter in the ViPR Controller REST API. During ingestion, if a file system is found with no hard quota, it is skipped for ingestion.
- Ingestion of filesystems having multiple exports with the same type of security is not supported.
- Ingestion of filesystems which have a common export. In this case filesystems will be ingested but the export will be ignored.
- Isilon only. Ingestion for system access zone filesystems uses the default path, `/ifs/vipr`. You may change the default path of the system access zone base directory in the **Physical** > **Controller Config** > **Isilon** UI page.

Best practices for ingestion

When allocating one access zone per project, create a new project for each access zone and allocate the access zone to that project before starting ingestion.

When allocating multiple access zones to a single project, ensure that the file systems on multiple access zones do not have matching names. (For example, if there are two file systems with the name, "Demo," on two different access zones, both will show up with the same name inside the project when both are ingested.)

It is possible to ingest file systems from access zones even if they are not allocated to any project. But in such cases, it's possible that the file systems may get ingested to the wrong project. The best practice is to first allocate access zones to specific projects before starting the ingestion process.

The virtual pool must be configured with the `Provisioning Type` set to Thin.

Special care must be taken while ingesting file systems from the system access zone. Ensure the path of the system access zone base directory is set correctly.

Ingest file systems from the ViPR Controller UI

To ingest file systems using the ViPR Controller Service Catalog, perform these tasks.

Procedure

1. Discover all unmanaged file systems on an array.

   See Discover unmanaged file systems on page 7.

2. Ingest unmanaged file systems.

   See Ingest unmanaged file systems on page 7.

Discover unmanaged file systems

The ViPR Controller Service Catalog provides a Discover Unmanaged File Systems service that finds file systems which are not under ViPR Controller management. The operation is also supported from the ViPR Controller API and CLI.

Before you begin

The following prerequisites are applicable:

- This operation requires the System Administrator role in ViPR Controller.
- The virtual array and virtual pool into which you want to ingest the storage pools must exist when the discovery is performed.

  The discovery process finds storage pools on a selected storage system and identifies the virtual array and virtual pool that discovered each file system matches with.
- File systems are only discovered on file storage system that were added to ViPR Controller as physical assets.

Procedure

1. Select **Service Catalog** > **View Catalog** > **File Storage Services** > **Discover Unmanaged File Systems**
2. Select the physical file storage systems from which you want to discover unmanaged file systems. You can select more than one file storage system.
3. Select **Order**.

   The orders page is displayed and shows the progress of the request. If the order is successfully fulfilled, you can use the Ingest Unmanaged File Systems to bring them under management by ViPR Controller.

Ingest unmanaged file systems

The ViPR Controller Service Catalog provides an Ingest Unmanaged File Systems service that brings previously discovered unmanaged file systems under ViPR Controller management. You can also perform this operation using the ViPR Controller API and CLI.

Before you begin

The following prerequisites are applicable:

Procedure

1. Select **Service Catalog** > **View Catalog** > **File Storage Services** > **Ingest Unmanaged File Systems**
2. Select the storage system from which you want to ingest file systems.

3. Select the virtual array whose virtual pools contain the storage system physical pools that host the file systems you want to import. The storage system might contribute physical storage pools to a number of virtual pools.
   - Only System Administrators can ingest unmanaged file systems.
   - Ensure the unmanaged file systems are in physical pools that are already associated with a ViPR Controller virtual storage pool.
   - Configure virtual pools. The storage pool from which you want to ingest file systems must match the virtual pool.
     - The virtual pool must have been configured with the `Provisioning Type` set to **`Thin`**. Isilon file systems are thinly provisioned. Thin resources can only be created in thin virtual pools.
     - By default, the Isilon file system exports will be ingested from System access zone (`/ifs/vipr`) and from all valid user access zones. You can also ingest them using a custom path. In the ViPR Controller UI, go to **Physical** > **Controller Config** > **Isilon**, and select **Unmanaged File System Locations**. The default path is shown in the first row and is grayed out. Add another line and specify a new value to override the default.
   - Configure virtual arrays. The Isilon storage system from which you want to ingest file systems must match the virtual array.
   - Create a project and assign the files systems to be ingested to the project. You must have write-permissions on the project.
   - Configure the controller configuration Isilon directory path. Ensure the name matches the naming convention used by the organization prior to deploying ViPR Controller.
   - Ensure the Discover Unmanaged File Systems service has been run on the file storage system.
   - Rerun the Discover Unmanaged File Systems service if the virtual array or virtual pools were modified after the last time the Discover Unmanaged File Systems service ran.
   - Quotas at a certain level as defined in **Physical** > **Controller Config** are considered to be filesystems and are ingested to ViPR Controller. Quotas at other levels are not ingested. For example, if the directory path is `/{vpool_name}/{tenant_name}/{project_name}`, then quotas at the sixth level are considered as filesystems. (This assumes the prefix would be `/ifs/<accesszone>`.)
   - Isilon only. An ingested Isilon Quota Directory always has a security style of "parent." The option for "parent" is not listed when first creating the Quota Directory. Instead, it is set during the ingestion of the Isilon Quota Directory.

   It is possible that not all of the storage system physical pools are included in the virtual array that forms part of your virtual data center. For this reason, you do not want to ingest all unmanged file systems on the storage system, just those in physical storage pools that meet the criteria of your virtual pools.
4. Select the virtual pool that the unmanaged file systems are in.
5. Select the project that you want the unmanaged file systems to be assigned to.
6. Select the file system type, Exported or Unexported.
7. Select **Order**.

The orders page is displayed showing the progress of the request. If the order is successfully fulfilled, you can look at the **Resources** > **File Systems** page to see the imported file systems.

Create and manage file systems

ViPR Controller provides the ability to create file systems and to make them available as CIFS shares and NFS exports, or both.

ViPR Controller supports concurrent file system provisioning operations in VNX for one or multiple arrays for the following services:

- Create a File System and CIFS Share
- Create a File System and NFS Export
- Remove a File System (with CIFS/NFS)

The following services are provided to enable the creation and management of file systems.

Table 2 Services to create and manage file systems

Service	Description
Create a File System	Enables you to create a new file system from a specified file virtual pool. The file system can be made available as a CIFS share or NFS export, or both.
Expand File System	Enables you to expand an existing file system.
Remove File System	Enables you to remove a file system. You can delete the file system from the ViPR Controller database (Inventory Only) or from both the ViPR Controller database and its backend storage system (Full). A Full delete removes the file system from the ViPR Controller database and its backend storage system. An Inventory Only delete removes the file system and all objects referencing the file system from the ViPR Controller database. **Note** ViPR Controller does not support the Full Deletion Type if the File System (FS) has objects referencing the file system, such as CIFS shares, snapshots, exports, access control lists, quota directories, and so forth. To remove this type of File System, you must delete all references first using other catalog services, for example, Remove NFS exports, Remove CIFS shares, Remove Quota Directory, Delete snapshots. Then delete the File System.
Reduce File System Quota	Isilon only. Use to reduce the file system quota. This does not shrink the file system. Instead, it changes the hard quota as set on Isilon. The soft quota and notification quota are also changed accordingly. The quota cannot be set below the actual usage on Isilon.

If you run the Create a File System service, the file system will create either CIFS shares, or NFS exports, or both depending on the settings in your virtual pool. If the virtual pool is set to both CIFS, and NFS, but the storage system is not enabled for

NFS, then at the time the service is run the CIFS share is created, no NFS export is created, and the following error is returned:

```
com.emc.vipr.client.exceptions.ServiceErrorException: Error 1034
(http: 400): An error occurred while finding a suitable placement to
handle the request. No Storage Port was assigned to virtual array...
```

The **Resources** > **File Systems** area enables you to view information about a file share and how it has been made available as a CIFS share or NFS export.

Expanding File Systems on EMC Isilon

When expanding a file system that resides on an EMC Isilon array, you can specify a file system size that is larger than the available capacity.

All Isilon file systems are thinly provisioned and do not consume any capacity when created. Hence, when extending a file system, a client may violate the limit, but the array will always alert when the array is near full utilization.

Provisioning error for vNAS servers

If a vNAS server is in an invalid state, such as the unloaded state, or was deleted from its storage system, ViPR Controller is unable to detect this until the next array discovery. ViPR Controller still selects these vNAS servers for provisioning, resulting in an error. You can run a provisioning operation again after the storage system has been rediscovered.

Zone ingestion

ViPR Controller will re-ingest zones of previously exported ViPR volumes if specific criteria are met.

ViPR Controller will not delete existing zones (that is, a zone that was on a switch already and was not created by ViPR Controller). This logic needed to be adjusted, however, for this use case:

1. Use ViPR Controller to create and export a volume.
2. Inventory delete the volume.
3. Re-ingest the volume that was originally created by ViPR Controller.

Prior to the 3.6.1 release, ViPR Controller considered these zones to be "existing" and, when the last volume was decommisssioned, the zone was left on the switch. The logic to determine whether a previously exported ViPR Controller volume is ViPR-created or not has been adjusted.

Zone ingestion considerations for exported volumes

As of version 3.6.1, ViPR Controller attempts to discover any existing zones for unmanaged exports before ingesting them.

Ingestion criteria

If the following criteria are met, ViPR Controller tries to assume control over the zones so they can be deleted when they are no longer in use.

- All volumes using the zone in the UnManagedExportMask must be ingested or be in the process of being ingested. At this point all the FC zone references are marked `existingZone=false`, but only after all known volumes are ingested. When ViPR Controller attempts to delete a zone, the export mask is refreshed. If the export mask has any "existingVolumes," that is, volumes present on the array that were not created or ingested by ViPR Controller, the zone will not be deleted, even though the zone references no longer have the `existingZone` flag set.

- For ViPR Controller to manage a zone, the zone must consist of a single initiator and a single storage port. This rule disqualifies "smart zones," which ViPR Controller can't update.
- The generated zone name (using the Zone Naming Configuration in ViPR Controller) must match the zone name that was discovered from the switch. Matching names indicate that ViPR Controller probably created the zone in the past, but the volumes were inventory deleted or created by an older ViPR Controller instance. The check for matching names can be disabled by EMC service personnel if needed.

If all the criteria above are met and after the volumes using the zone have been ingested, the zones are marked `existingZone=false`. This flag indicates that when the last reference to the zone is deleted in ViPR Controller, ViPR Controller should schedule zone deletion for the zone.

When the last volume using the zone is being deleted from ViPR Controller, if `existingZone=false`, the zone will be scheduled for deletion. Otherwise, this Warning Message will be added to the Task and Order doing the volume deletion indicating the zone(s) may still be in use: `"Zones which will not be deleted because they may be used externally: <zone_names>"`

Port allocation enhancements

The method of determining overall port usage has been enhanced.

Port usage now uses the Physical > Controller Config > Port allocation information to combine port metrics with a term based on the number of volumes. For example, `PortUsage = PortMetric + (VolumeCoefficient * NumberOfVolumes * 100.0 / 2048)`.

- The 3.6.1 release introduces “floor values” for Port Percent Busy and Cpu Percent Busy. These are referred to as **Port Utilization Floor** and **Cpu Utilization Floor** respectively. Metric values below their respective floor are ignored. The floors represent the maximum percent busy found on an idle port, that is, a port not receiving any I/O.
- The port with the lowest PortUsage is chosen among ports that provide equal hardware redundancy.
- After identifying a set of ports that can provide adequate hardware redundancy, the Storage Port Allocator uses port metrics to determine the least heavily loaded port for allocation.
- The port metric is determined from averaging the Port Percent Busy and Cpu Percent Busy for VMAX, VPLEX, VNX, and HDS arrays. (HDS does not provide Cpu Percent Busy).
- Other arrays that do not provide port metrics use the lowest volume count as a metric. There is a problem in that the port metrics do not work well on new arrays, because there is no significant I/O load (which results in idle values of Port Percent Busy and Cpu Percent Busy). This could cause unbalanced allocation of volumes to ports that had no significant I/O.

Overview of metrics-based port selection

Learn how to define the maximum performance-based limits for ports and how those limits are used by ViPR Controller for allocating new ports. Allocating new ports based

on performance metrics, computed metrics, and user-defined maximum limits is supported on VMAX, VPLEX , VNX for Block, and Hitachi Data Systems (HDS).

Several performance-based metrics are collected from VMAX, VPLEX, VNX for Block, and HDS and are used to determine:

- Port percent busy.
- CPU percent busy.

Two additional metrics are also computed from the ViPR Controller database:

- Number of initiators using a storage port.
- Number of volumes using a storage port.

These metrics are then used to allocate new ports to avoid:

- Ports that are overloaded with too many volumes or too high of an I/O load.
- Ports that reside on CPUs where the CPU percent busy is too high or the CPU is servicing too many volumes.
- Allocating more storage on arrays that are overloaded.

For information on how ViPR Controller allocates new ports, based on these metrics, see How does ViPR Controller select a port when using performance metrics on page 12.

Before ViPR Controller can allocate new ports based on performance metrics, there are configuration requirements you must set up on HDS, and VNX for Block storage systems. For configuration requirements refer to the: *ViPR Controller Virtual Data Center Requirements and Information Guide* on the ViPR Controller Product Documentation Index.

You must also enable ViPR Controller for the collection of metrics from the storage arrays, as described in Set up metering prerequisites in ViPR Controller on page 18.

How does ViPR Controller select a port when using performance metrics

ViPR Controller takes averages of the performance-based metrics collected from the storage arrays and the number of initiators and volumes that it has already allocated to ports, and then compares these metrics to maximum limits (ceilings) that you configure fo determine which ports to select.

Metric calculations and averages

Several performance-based metrics are collected from VMAX, VPLEX , VNX for Block, and HDS and are used to determine:

- Port percent busy
- CPU percent busy

In addition, two additional metrics are computed byViPR Controller :

- Number of initiators ViPR Controller has already assigned to a storage port
- Number of volumes ViPR Controller has already assigned to a storage port

Note

On VMAX2, the number of volumes is computed across both ports on a director.

These numbers may not reflect all exports done outside of ViPR Controller.

If you do an Unmanaged Volume Discovery, the unmanaged volumes and initiators discovered for the storage array will be added to the ViPR Controller-provisioned volumes and initiators.

Averaging the metrics values

The metrics collected for CPU Percent Busy and Port Percent Busy are averaged over time so that they reflect a relatively long term view of whether the port is overloaded. The system administrator can control this averaging process. There are three important time periods:

- The `Metering Interval` controls how often metering records will be read from the storage arrays. The default time period for this is one hour. This can be reduced to 30 minutes, or increased to multiple hours. To get accurate metrics on heavily loaded ports, it may be necessary to decrease the metering interval to 30 minutes, although this will cause increased ViPR Controller load for systems with many arrays. Increasing the metering interval will reduce the load. It is not recommended to have a metering interval greater than four hours. For information on how to set `Metering Interval`, see Set up metering prerequisites in ViPR Controller on page 18.
- The `Days to Average Utilization`, one of the ViPR Controller Port Allocation parameters, controls how long various samples are averaged together using a modified moving average. The default averaging period is 1 day, but you can configure the period from 1 to 30 days. The longer the averaging period, the less an instantaneous change in load is reflected in the average, and the less affect a current sample will have on the average. After the averaging period has been completed, a new average starts and will be computed. For information on how to set `Days to Average Utilization`, see *EMC ViPR Controller REST API Reference*.
- At the end of each averaging period, the modified moving average is added into a longer term Exponential Moving Average (EMA) that is calculated for each metric. The purpose of the EMA is to retain history about the port's utilization over time. An EMA is used because it weights recent values higher, and past values with exponentially decreasing weights as the sample's age increases. In that way recent port utilization is more important than past utilization. The `Weight for Exponential Moving Average` controls the weight of the current modified moving average versus past averages. For information on how to set `Weight for Exponential Moving Average`, see *EMC ViPR Controller REST API Reference*.

The default weight of the EMA is set at 0.6, but you can configure the weight from greater than 0 to less than or equal to 1. The higher the EMA weighting factor the more weight that the current modified moving average has on the EMA. A value of 1.0 uses only the current averaging period. For example, if the EMA weight is 0.6, the current modified moving average is multiplied by 0.6 and added to the previous EMA multiplied by 0.4 (1 - 0.6).

User-configurable parameters

There are several maximum limits (ceilings) that you can set, in addition to sampling times and the weight to use for the exponential moving average (EMA). As of ViPR Controller version 3.6.1, there are three new parameters related to port metrics: `Port Utilization Floor`, `Cpu Utilization Floor`, and `Metric Volume Coefficient`.

When a port reaches or exceeds one of the ceiling values, it is no longer available for new allocations, even if that causes provisioning to fail. You can change the settings using the user-configurable parameters. See the *EMC ViPR Controller REST API Reference*.

Table 3 User-configurable parameters affecting port allocation

Parameter	Type	Default value	Minimum value	Maximum value	Description
Initiator Ceiling	Ceiling	unlimited	1	N/A	If the number of initiators using a port is equal to or greater than the ceiling, the port is disqualified from allocation. Use this value to control the absolute maximum limit of the number of initiators that will use a port.
Volume Ceiling	Ceiling	unlimited	1	N/A	If the number of volumes using a port is equal to or greater than the ceiling, the port is disqualified from allocation. Use this value to control the absolute maximum limit of the number of volumes that will use a port.
Port Utilization Ceiling	Ceiling	100%	0%	100%	If the port percent busy of a port is equal to or greater than the ceiling, the port is disqualified from allocation. Use this value to set a maximum limit on the port utilization expressed in percent.
Cpu Utilization Ceiling	Ceiling	100%	0%	100%	If the cpu percent busy of a port is equal to or greater than the ceiling, the port is disqualified from allocation. Use this value to set a maximum limit on the cpu utilization expressed in percent.
Days to Average Utilization	Time	1	1	30	The number of days that samples are averaged before being considered a valid "metric."
Weight for Exponential Moving Average	Time	0.6	0	1.0	The weight, `k`, given the current metric versus previous metrics (`k * metric[T] + (1-k) * metric [T-1]`)
Port Utilization Floor	Floor	3%	0%	100%	If the port metric is below the floor, it is arbitrarily eliminated from consideration by setting it to `0.0`. This should be set just slightly higher than the idle `Port Percent Busy` on a port on which essentially no I/O is being performed.
Cpu Utilization Floor	Floor	8%	0%	100%	If the cpu metric is below the floor, it is arbitrarily eliminated from consideration by setting it to `0.0`. This should be set just slightly higher than the idle `Cpu Percent Busy` on an idle array where essentially no I/O is being performed.
Metric Volume Coefficient	Coefficient	1.0	0.0	5.0	A coefficient controlling the importance of the number of volumes in the metric calculation. A port is assumed to service a maximum of 2048 volumes (arbitrarily.)

Table 3 User-configurable parameters affecting port allocation (continued)

Parameter	Type	Default value	Minimum value	Maximum value	Description
					Therefore about 20.5 volumes is assumed to be equivalent to a 1% volume load. The volume load (in percent) is added to the metric containing the port and cpu utilizations. Setting the `Metric Volume Coefficient` to a higher value will make the volume counts using a port more predominate in the selection of the least used port.
Metrics Enabled	N/A	True	N/A	N/A	• When set to **`True`**, use collected metrics and calculate Port percent busy and CPU percent busy. • When set to **`False`**, only use the number of initiators and the number of volumes to allocate ports. Ignore the collected metrics and do not calculate `Port percent busy` and `CPU percent busy`. CPU percent busy is not calculated for HDS.
Zoned Ports Used For Host Exports	Miscellaneous	False	N/A	N/A	If set to true, before allocating ports ViPR Controller checks for pre-existing zones between the initiators that are being provisioned and checks for array storage ports assigned to the applicable virtual array. If present, these ports will be used first before allocating any new ports. This applies when ViPR Controller is provisioning exports to Hosts.
Zoned Ports Used For Backend Ports	Miscellaneous	False	N/A	N/A	If set to true, ViPR Controller checks for existing zones when provisioning ports for use by the VPLEX back-end ports that are acting as initiators to a storage system. If there are existing zones, ViPR Controller generates an export from the backend to the array using the ports in the existing zones if possible.
Switch Affinity Enabled	Miscellaneous	True	N/A	N/A	If set to true, ViPR Controller will consider switch affinity as part of the storage port allocation rules. This helps avoid use of ISL links when initiator to port connections through a single switch can be accomplished.

Note

- Volumes may be added to existing exports, such as masking views, storage groups, and storage views, without allocating new ports. These will put additional port load on the ports in that existing export. Therefore, you should set your ceilings lower than the maximum limit you require.
- Take care in setting ceilings. These are absolute limits. Ports which have one or more metrics over a ceiling will not be used for any allocations until such time as all metrics return to a value under the ceilings (or the ceiling limits are increased).
- Take care in setting the Port Utilization Floor and Cpu Utilization Floor. They are intended to represent the maximum metric observed on an idle port (that is, a port that has no I/O.) If the metrics are below the floor settings, then the metrics are ignored and only the volume counts are used to determine the least used port.

Allocating a port

ViPR Controller uses the exponential moving averages of `Port percent busy` and `Cpu percent busy` and compares these against the corresponding ceilings that you have configured. If either value is above the configured ceiling, the port is disqualified from being considered for allocation. Additionally, if the current volume count or initiator count handled by a port exceeds the corresponding ceilings that you have configured, the port is also disqualified.

A port metric value ranging from 0 - 100% is computed for the remaining ports that are not disqualified. This value represents an average of the `Port percent busy` and the `Cpu percent busy` (if applicable). This represents the I/O load on the port.

Volume calculations

The volume usage factor is also calculated for each port. It is expressed in percent as the number of volumes using the port times 100 divided by 2048. That sets a 100% volume usage factor as 2048 volumes using the port.

Ports will be selected based on these factors:

- Ports must not exceed any of the ceiling values (initiator ceiling, volume ceiling, port percent busy ceiling, or cpu percent busy ceiling). Any port that is exceeding one or more of the ceiling parameters is eliminated from further consideration
- The port metric value (averaged by the Exponential Moving Average) is added to the volume usage factor times the volume metric volume coefficient. Each of these numbers is in percent, so the possible sum is now 0 - 200% (with the default metric volume coefficient of 1.0). This represents an overall port utilization. The port with the lowest port utilization will be selected (after the hardware redundancy rules have been applied.)
- The importance given to the port metric versus the number of volumes using the port in determining which port to allocate is controlled by the Metric Volume Coefficient. Increase the Metric Volume Coefficient to give more significance to the number of volumes. Reduce the value of the coefficient to give more significance to the Port Metric.

Example: Five different ports have a Port Metric and Volume Usage Factor as shown below. Which port would be selected? (assuming default volume coefficient of 1.0)

Port Metric	Number of Volumes Using Port	Volume Usage Factor	Port Utilization %
0%	1000	48.8%	48.8%

Port Metric	Number of Volumes Using Port	Volume Usage Factor	Port Utilization %
28%	500	24.4%	52.4%
30%	100	4.8%	34.8%
32%	2	0.1%	32.1% (Lowest Port Utilization chosen)
35%	0	0.0%	35.0%
70%	2500	122.0%	192.0%

Note

If you have already allocated ports to a host or cluster, and you are just adding volumes to the same host, then ViPR Controller does not reallocate ports, it just adds the volumes to the export structure.

Global default port selection

ViPR Controller has a default port selection algorithm that can be used globally across all arrays.

The global default port selection algorithm is used:

- When performance-metrics collection is disabled for VMAX, VPLEX , VNX for Block, or Hitachi Data Systems (HDS).
- For storage arrays other than VMAX, VPLEX , VNX for Block, and HDS.

Calculated values

ViPR Controller automatically calculates two values from its database:

- Number of initiators ViPR Controller has already assigned to a storage port.
- Number of volumes ViPR Controller has already assigned to a storage port.

  Note

  On VMAX2, the number of volumes is computed across both ports on a director.

These numbers may not reflect all exports done outside of ViPR Controller.

User-configurable parameters

You can set a maximum limit for the number of initiators and volumes that use the port before new allocations are not allowed.

Volumes may be added to existing exports, such as masking views, storage groups, and storage views, with allocating new ports. These will put additional port load on the ports in that existing export. Therefore, you should set your ceilings lower than the maximum limit you require.

When a port exceeds one of the ceiling values, it is no longer available for new allocations, even if that causes provisioning to fail. You can change the settings in the ViPR Controller UI, as explained Change the port allocation parameters using the UI.

Note

You should take care in setting ceilings. These are absolute limits. Ports which have one or more of the number of initiators or volumes over their ceiling will not be used for any allocations until such time as both the number of initiators and the number of volumes return to a value under the ceilings (or the ceiling limits are increased).

Allocating a port

The port is determined as follows:

1. Ports are checked against the ceilings for the number of initiators and volumes.
2. The ports below their ceilings are checked for redundancy. When you require more than one port allocated, ViPR Controller tries to choose two ports that are on different hardware units.
3. From the set of ports with the most redundancy, the ports with the fewest number of volumes are selected.

Note

If you have already allocated ports to a host or cluster, and you are just adding volumes to the same host, then ViPR Controller does not reallocate ports, it just adds the volumes to the export structure.

Set up metering prerequisites in ViPR Controller

There are two configuration properties that you must ensure are set in ViPR Controller to enable the collection of metrics from VMAX, VPLEX , VNX for Block, and HDS.

You can set these configuration properties using both the ViPR Controller UI and the ViPR Controller UI REST API.

Use the ViPR Controller UI

After logging into the ViPR Controller UI as a system administrator, check whether metering is enabled, and to what value the metering interval is set.

Selecting **System** > **General Configuration** > **Controller** displays:

- The value of **Enable Metering** that must be set to **true** to collect metrics from the arrays.
- The value of **Metering Interval** that defines how often ViPR Controller collects data from the arrays. The metering interval can be set from 1800 seconds (30 minutes) up to 4 hours. The lower the number of seconds, the more accurate are the results. However, the higher the metering interval, the less overhead there is on ViPR Controller and the array.

Prerequisites for VNX and HDS metrics-based port selection

There are configuration settings on the VNX and HDS that are required for metrics-based port selection.

For prerequisite configuration settings for both VNX and HDS, see the *ViPR Controller Virtual Data Center Requirements and Information Guide* on the ViPR Controller Product Documentation Index.

Change the default port allocation parameters

System administrators can change the default values of the port allocation parameters.

You can use ViPR Controller UI or the REST API to change the port allocation parameters.

Change the port allocation parameters using the UI

You change the default values of the port allocation parameters by adding a new parameter setting. When you add a new parameter setting, ViPR Controller uses your setting value instead of the default value.

Before you begin

- You must set how often ViPR Controller will collect data from the array.
- Only system administrators can change port allocation parameters.

You can change these parameters:

Initiator Ceiling

If the number of initiators using a port is equal to or greater than the ceiling, the port is disqualified from allocation. Use this value to control the absolute maximum limit of the number of initiators that will use a port.

Volume Ceiling

If the number of volumes using a port is equal to or greater than the ceiling, the port is disqualified from allocation. Use this value to control the absolute maximum limit of the number of volumes that will use a port.

Port Utilization Ceiling

If the port percent busy of a port is equal to or greater than the ceiling, the port is disqualified from allocation. Use this value to set a maximum limit on the port utilization expressed in percent.

Cpu Utilization Ceiling

If the cpu percent busy of a port is equal to or greater than the ceiling, the port is disqualified from allocation. Use this value to set a maximum limit on the cpu utilization expressed in percent.

Days to average Utilization

The number of days that samples are averaged before being considered a valid "metric." The sample averaging time in days (1 -30 days). Default is one day.

Weight for Exponential Moving Average

The weight, `k`, given the current metric versus previous metrics (`k * metric[T] + (1-k) * metric [T-1]`)

Port Utilization Floor

If the port metric is below the floor, it is arbitrarily eliminated from consideration by setting it to 0.0. This should be set just slightly higher than the idle "Port Percent Busy" on a port on which essentially no I/O is being performed.

Cpu Utilization Floor

If the cpu metric is below the floor, it is arbitrarily eliminated from consideration by setting it to 0.0. This should be set just slightly higher than the idle "Cpu Percent Busy" on an idle array where essentially no I/O is being performed.

Metric Volume Coefficient

A coefficient controlling the importance of the number of volumes in the metric calculation. A port is assumed to service a maximum of 2048 volumes (arbitrarily.) Therefore about 20.5 volumes is assumed to be equivalent to a 1% volume load. The volume load (in percent) is added to the metric containing the port and cpu utilizations. Setting the Volume Coefficient to a higher value will make the volume counts using a port more predominate in the selection of the least used port.

Metrics Enabled

Note

CPU percent busy is not calculated for HDS

- true = use collected metrics and calculate Port percent busy and CPU percent busy. This is the default.
- false = only use the number of initiators and the number of volumes to allocate ports; ignore the collected metrics and do not calculate Port percent busy and CPU percent busy.

Zoned Ports Used For Host Exports

If set to true, before allocating ports ViPR Controller checks for pre-existing zones between the initiators that are being provisioned and checks for array storage ports assigned to the applicable virtual array. If present, these ports will be used first before allocating any new ports. This applies when ViPR Controller is provisioning exports to Hosts.

Zoned Ports Used For Backend Ports

f set to true, ViPR Controller checks for existing zones when provisioning ports for use by the VPLEX back-end ports that are acting as initiators to a storage system. If there are existing zones, ViPR Controller generates an export from the backend to the array using the ports in the existing zones if possible.

Switch Affinity Enabled

If set to true, ViPR Controller will consider switch affinity as part of the storage port allocation rules. This helps avoid use of ISL links when initiator to port connections through a single switch can be accomplished.

Procedure

1. Log into the ViPR Controller UI with System Administrator privileges.
2. Select **Physical** > **Controller Config**
3. Select **Port Allocation**.
4. Select the port allocation parameter that you want to change.
5. Click **Add.**
6. Select the **Scope Type**.
7. Select the **Scope Value**.
8. Type the value of the parameter.
9. Click **Save.**

VMAX performance metrics

The VMAX metrics collection is contingent on having metering turned on and configured.

The table describes the metrics collected from VMAX that ViPR Controller uses to allocate ports.

Table 4 Performance metrics collected on VMAX

Metric	Variable	Description
FEPort, FEAdapt: StatisticTime	sampleTime	A string representing the current time with the format, *yyyyMMddHHmmss.SSSSSS sutc*, where: • yyyy - is a 4 digit year • MM - is the month • dd - is the day of the month • HH - is the hour (24 hour clock) • mm - is the minut;e ss - is the second • mmmmmm - is the number of microseconds • sutc gives the sign and offset from GMT
FEAdapt: TotalIOs	iops	The cumulative number of I/O operations for the CPU (read and write).
FEAdapt: EMCIdleTimeDir	idle	The cumulative number of idle ticks.
FEPort: TotalIOs	iops	The cumulative number of IO requests for a port (read and write).
FEPort: KbytesTransferred	kbytesTransferred	The cumulative number of kilobytes transferred for read or write.
FEAdapt: EMCCollectionTimeDir	ticks	The cumulative number of ticks.

These metrics are used to calculate two values:

- Percent busy for the port (FEPort) which is computed from kbytesTransferred over the time period since the last valid sample.
- Percent busy for the CPU (FEAdapt) which is computed from the non IdleTime over the time period since the last valid sample.

VPLEX performance metrics

The VPLEX metrics collection is contingent on having metering turned on and configured. Set **Enable Metering** to `true` in **System** > **General Configuration** > **Controller** .

Each management server in a VPLEX MetroPoint configuration is a storage provider for VPLEX . Add the provider details for each of the VPLEX management servers using the **Physical** > **Storage Providers** > **Add** page in the ViPR Controller UI. This adds both cluster manager IP addresses to ViPR Controller and enables VPLEX port performance on both front-end ports.

The table describes the metrics collected from VPLEX that ViPR Controller uses to allocate ports.

Table 5 Performance metrics collected on VPLEX

Metric	Variable	Description
Timestamp	Time	The sample time in format: yyyy-mm-dd hh:mm:ss in UTC.
Director percent busy	director.busy	The percent time the director is busy performing I/O operations.
Director IOPs/sec	director.fe-ops	The number of I/O operations executed per second by the director.
Port IOPs/sec	fe-prt.ops	The number ofI/O operations executed per second by the port.
Port KB read/sec	fe-prt.read	The number of Kilobytes read per second by the port
Port KB write/sec	fe-prt.write	The number of Kilobytes written per second by the port

These metrics are used to calculate:

- Percent busy for the port (FEPort) which is computed from kbytesTransferred over the time period since the last valid sample.

VNX for Block performance metrics

The table describes the metrics that are collected onVNX for Block which ViPR Controller uses to allocate ports.

Note

VNX for Block metrics collection is contingent on having metering turned on and configured. See Prerequisites for VNX and HDS metrics-based port selection for more information.

Table 6 Performance metrics collected on VNX for Block

Metric	Variable	Description
FEPort: Total IOPs	iops	The cumulative number of IO requests for a port (read and write).
FEPort: KbytesTransferred	kbytesTransferred	The cumulative number of kilobytes transferred for read or write.
FEAdapt: IdleTimeCounter	idle	The cumulative ticks of idle time (idleTicksValue)
FEAdapt: IOTimeCounter	ioTime	The cumulative ticks of I/O busy time.
FEAdapt: TotalIOs	iops	he cumulative number of I/O operations for the CPU (read and write).
FEPort, FEAdapt: StatisticTime	sampleTime	A string representing the current time, of the format *yyyyMMddHHmmss.SSSSSS sutc* where: • yyyy - is a 4 digit year • MM - is the month • dd - is the day of the month • HH - is the hour (24 hour clock) • mm - is the minutes • ss - is the seconds • mmmmmm - is the number of microseconds • sutc gives the sign and offset from GMT

These metrics are used to calculate two values:

- Percent busy for the port (FEPort) which is computed from kbytesTransferred over the time period since the last valid sample.
- Percent busy for the CPU (FEAdapt) which is computed from the non idle time over the time period since the last valid sample.

HDS performance metrics

The table describes the metrics collected from HDS that ViPR Controller uses to allocate ports.

Table 7 Performance metrics collected on HDS

Metric	Variable	Description
FEPort: Total IOPs	iops	The cumulative number of IO requests for a port (read and write).
FEPort: KbytesTransferred	kbytesTransferred	The cumulative number of kilobytes transferred for read or write.
FEPort, FEAdapt: StatisticTime	sampleTime	A string representing the current time, of the format *yyyyMMddHHmmss.SSSSSS sutc* where: • yyyy - is a 4 digit year • MM - is the month • dd - is the day of the month • HH - is the hour (24 hour clock) • mm - is the minutes • ss - is the seconds • mmmmmm - is the number of microseconds • sutc gives the sign and offset from GMT

These metrics are used to calculate:

- Percent busy for the port (FEPort) which is computed from kbytesTransferred over the time period since the last valid sample.

Rollback improvements

This release continued improvements in the handling of failed operations and rollbacks.

The goals of the improvements are:

- Avoid inconsistent ViPR-C database information caused by failed operations.
- Prevent situations where the customer cannot rerun a job that failed.
- Add checks to make sure the host has access to storage before attempting certain operations.
- Continue the rollback automation framework used in the 3.6 release, including improved error messages and ability to inject failures in the workflows.
- Support for rollback automation in the service catalog.

Services updated with rollback improvements

In this release, rollback improvements are incorporated into the workflows associated with volume and datastore extensions and expansions, and SRDF volume creation.

The following services were improved with better prechecks and rollback features for host-related access problems:

- Extend Datastore with New Volume
- Extend Datastore with Existing Volume
- Expand Volume and Datastore
- Expand Linux Mount

The following SRDF services were improved with better prechecks and rollback features:

- SRDF non-CG Volume Creation
- SRDF Consistency Group Volume Creation
- Add SRDF Protection via Vpool Change

Order failures, rollbacks, and retries

Orders can fail for a variety of reasons. ViPR Controller attempts to reduce the negative impact of failed operations.

ViPR Controller services include checks to ensure the validity of the parameters supplied for an order and that conditions will permit a successful outcome. If an order fails, ViPR Controller does the following:

- Produces error messages that clearly state the root cause of the failure, enabling the customer to fix the conditions that caused the failure.
- Ensures that you can rerun a failed order.
- Ensures that the ViPR Controller databases are left in a consistent state.
- Rolls back completed operations in a workflow that ends in failure If appropriate and if possible. The rollbacks ensure database consistency and the ability to retry an order without encountering additional errors.

The following sections describe some specific rollback and retry situations.

In the case of failures caused by host access problems:

- Error messages include the complete ViPR CLI command string, complete with argument values, that can fix issues caused by lack of access to the storage resource.
- On a retry, the Expand Datastore and Linux Mount catalog services skip expanding the volume if those resources are already expanded. If the datastore or filesystem expansion failed in the previous order, then the retry will attempt the expansion again.
- On VMware hosts, an order does not fail if one of the ESXi hosts in the cluster stops responding during a VMware extend operation.
- On a Linux host, if an error occurs during an expand operation, the rollback operation mounts the volume, ensuring continued data availability.

In the case of failures during SRDF expansion:

- Rollback ensures the integrity of the ViPR Controller databases regarding the involved resources.
- Rollback attempts to clean up the physical resources (volumes, CGs, RDF groups) on the array/switch/compute.
- Error messages describe:
  - The root cause of the failure.
  - Any problems encountered during the rollback operations,
  - Specific information about resources that could not be cleaned up during the rollback. Enough detail is provided so the user can perform the cleanup manually before attempting a retry.

Export service added to all-flash services

You can now export an all-flash volume from the ViPR Controller catalog.

In previous releases, the all-flash export service was missing, requiring the user to perform the export on the command line.

To use this service, select **Catalog** > **View Catalog** > **All Flash Services** > **Export Volume to a Host**.

For help completing the input fields, click the **?** icon.

Export a VMAX3 resource to multiple compute resources

You can export a VMAX3 volume or replica to multiple storage groups.

Previously, ViPR Controller was unable to export the same VMAX3 resource to multiple compute resources whenever the original storage group was FAST-managed. Starting with ViPR Controller 3.6.1, VMAX3 resources (volume, linked snapshot, full copy, continuous copy) can be successfully exported to multiple compute resources (hosts or clusters). In addition, performance for exporting and deleting multiple VMAX3 volumes is greatly enhanced.

Background information about FAST-managed resources

The performance of an array partially depends on the placement of frequently accessed data on high-speed disks such as Flash, and infrequently-accessed data on slower storage. The Fully Automated Storage Tiering (FAST) moves data among drive types to optimize array performance.

FAST-managed resources in VMAX3 have the following characteristics:

- A storage group is under FAST control if it has any or all of the following: SLO or workload assigned, or SRP set.
- Workload cannot be set on a storage group without SLO.
- Compression can be enabled only on FAST-managed storage groups.

Multiple exports with a FAST-managed resource

Devices can be included in more than one storage group, but only one of those storage groups can be FAST-managed. This ensures that a single device cannot be managed by more than one service level objective or have data allocated in more than one storage resource pool.

To accommodate the FAST limitations while also allowing for multiple exports, the export feature is implemented as follows:

- The same resources (devices) can be part of multiple storage groups through different masking views, but only one of those storage groups will be FAST-managed. To accomplish this goal, when a resource is exported for the second and subsequent times, if the device is already FAST-managed, it is automatically assigned into a non-FAST group.
- During un-export, if a device is removed from the FAST-managed storage group but still belongs to other non-FAST storage groups, it is added to a parking SLO storage group that is FAST-managed. The parking storage group uses the virtual pool's FAST settings.
- The parking storage group also enhances efficiency during multiple volume exports and multiple volume deletes.

Host I/O limits and multiple export support limitation
Multiple exports are not supported if Host I/O Limits is set on a device. You must unset Host I/O Limits before you can export the resource to multiple compute resources.

To unset:

1. Use the **Change Virtual Pool** catalog service.
2. Select the devices in the source virtual pool whose Host I/O Limits need to be turned off.
3. Choose the operation **Change Auto-tiering Policy, Host IO Limits or Compression Setting**.
4. Choose the target virtual pool that has no Host I/O Limits set.
5. Proceed with multiple exports of those devices.

Improved CLI commands for metropoint virtual pools

The `vpool create` and `vpool update` commands are improved to allow for easier creation of virtual pools for metropoint RecoverPoint.

Changes to `viprcli vpool create` are:

- The syntax for the `rp_source_policy` option changed. The new syntax is:

```
remotecopymode:rpovalue:rpotype:journalsize:journal_varray:journ
al_vpool:standby_journal_varray:standby_journal_vpool
```

- The syntax for the `-ha` option changed. The two Boolean parameters for metropoint and active protection at HA site were deleted from this option and added as separate options.
- `metropoint` option added. This option enables (or disables) metropoint. The `-ha` option is required when using this option. Syntax is:

```
-metropoint {true | false}
```

- `activeProtectionAtHASite` option added. This option specifies if the HA site is the active site for RecoverPoint with VPLEX, or for metropoint. The `-ha` option is required when using this option. Syntax is: .

```
-activeProtectionAtHASite {true | false}
```

Changes to `viprcli vpool update` are:

- The syntax for the `rp_source_policy` option changed. The new syntax is:

```
remotecopymode:rpovalue:rpotype:journalsize:journal_varray:journ
al_vpool:standby_journal_varray:standby_journal_vpool
```

Example: Creating a virtual pool for metropoint

```
# viprcli vpool create -n V1_rp_metropoint -t block -pl FC -pt Thin -msnp 100 -mxp 1 -
expandable true -mv true -acc true -desc V1_rp_metropoint -va varray1 -ha
vplex_distributed:varray3:VP3_vplex -rp varray1:VP1_vplex:10GB varray3:VP3_vplex:10GB -
rp_source_policy ASYNCHRONOUS:25:SECONDS:10GB:varray1:VP1_vplex:varray3:VP3_vplex -
metropoint true -activeProtectionAtHASite false
```

CLI documentation

The revised pages for these commands from the *ViPR Command Line Reference Guide* follow.

viprcli vpool create

Create a virtual storage pool in ViPR Controller. A virtual storage pool is a group of storage pools according to their attributes and properties.

Syntax

```
            viprcli vpool create
[-h]
[-hostname <hostname>]
[-port|po <port_number>]
[-portui|pu    <ui_port_number>]
[-cookiefile|cf    <cookie_file>]
[-name|n <vpoolname>
[-protocol|pl <protocol>    [<protocol. ...]
[-varrays|va <varray>    [<varrays> ...]
[-provisiontype|pt <provisiontype>    { Thick | Thin }]
[-maxsnapshots|msnp    <max_snapshots>]
[-maxretention|mxrtn        <max_retention>]
[-mindatacenters|mndcs        <min_datacenters>]
[-longtermretention|ltrtn    <long_term_retention>]
[-maxcontinuouscopies|mcc        <max_continuos_copies>]
[-continuouscopiesvpool|ccv    <continuoscopies_vpool>]
[-highavailability|ha    <highavailability>]
[-rp <rp> [<rp>    ...]]
[-rp_source_policy|rp_policy    <rp_source_policy>]
[-metropoint {true | false}]
[-activeProtectionAtHASite {true | false}]
[-file_replication_policy|frpol    <file_replication_policy>]
[-file_replication_copies    <file_replication_copies>
[<file_replication_copies> ...]]
[-systemtype|st <systemtype>    {isilon | vnxblock | vnxfile | vmax
| netapp | netappc |vplex | hds | openstack
        | scaleio | xtremio | ddmc | ibmxiv | vnxe | datadomain}]
[-raidlevel|rl <raidlevel>    [<raidlevel> ...]{RAID1 | RAID2 |
RAID3 | RAID4 | RAID5 | RAID6 |
        RAID10}]
[-fastpolicy|fp    <fastpolicy>]
[-drivetype|dt <drivetype> {SSD |        FC | SAS | NL_SAS | SATA |
NONE}]
[-type|t <vpooltype> { file |    block }]
[-description|desc]        <description>
[-usematchedpools|ump        <useMatchedPools>]
[-multivolconsistency|mvc        {true,false}]
```

```
[-expandable|ex <expandable>        ]
[-autoCrossConnectExport|acc        <autoCrossConnectExport> {true
| false}]
[-fastexpansion|fe        <fastexpansion>
[-autotierpolicynames|apn        <unique_auto_tier_policy_names>> ]
[-enablecompression <enable_compression>]
[-maxpaths|mxp        <max_paths>]
[-minpaths|mnp        <min_paths>]
[thinpreallocper|tpap        <thinpreallocper>]
[frontendbandwidth|febw        <frontendbandwidth>]
[iopersec|iops        <iopersec>]
[-pathsperinitiator|ppi        <PathsPerInitiator>]
[-snapshotschedule|snapsched        {true,false}]
[-srdf <SRDF> [<srdf>        ...]]
[-placementpolicy|pp <placementpolicy>]
```

Options

name|n

The name of the virtual storage pool. This is a mandatory parameter.

protocol|pl

For file storage, Protocol can be `NFS`, `NFSv4`,`CIFS` or `ScaleIO`. For block storage, Protocol is `FC`, `iSCSI`, or `ScaleIO`. This is a mandatory parameter.

varrays|va

The list of virtual storage arrays where the primary storage is available (empty set means available in all virtual storage arrays). This is a mandatory parameter.

provisiontype|pt

Provision type. Type of `Thin` or `Thick`. This is an optional parameter.

Note

This is a mandatory parameter for block storage type. If you are creating this virtual pool for VMAX journal volumes, `provisiontype` must be set to `Thick`, as ViPR Controller does not pre-allocate the volumes.

---

maxsnapshots|msnp

The maximum number of native snapshots. This must be greater than '0'. This is a mandatory parameter to create a snapshot. This is an optional parameter.

maxretention|mxrtn

Maximum retention period. This is an optional parameter.

mindatacenters|mndcs

Minimum number of data centers. This is an optional parameter.

longtermretention|ltrtn

Long term retention. This is an optional parameter.

maxcontinuouscopies|mcc

The maximum number of native continuous copies. You can set this only for a vpool of type Block. This is an optional parameter.

Note

This is set to zero '0' by default. You must set it to a value greater than zero and set `expandable` to **`false`**, if you want to create continuous copy of a volume using `viprcli volume continuous_copies create` command with the protection type as native.

continouscopiesvpool|ccv

The vpool name for continuous copies. You can set this only for a vpool of type Block. This is an optional parameter.

Note

Optionally, you can set this parameter, if you want to create continuous copy of a volume using `viprcli volume continuous_copies create` command with the protection type as native.

highavailability|ha

High-availability is available only for VPLEX arrays. The possible values are vplex_local or vplex_distributed. This is an optional parameter.

Note

For vplex_distributed you must also specify varray:vpool with high-availability type.

rp

RecoverPoint target. Specify in the form set of varray:vpool:journalsize. This is an optional parameter.

rp_source_policy|rp_policy

Sets parameters for RecoverPoint copies. The syntax is:

```
remotecopymode:rpovalue:rpotype:journalsize:journal_varray:journ
al_vpool:standby_journal_varray:standby_journal_vpool
```

- `remotecopymode` is the remote copy mode for the RecoverPoint CG. Values are: `ASYNCHRONOUS` (the default) or `SYNCHRONOUS`. When using `SYNCHRONOUS`, the next two parameters must be `None`, as follows:

  ```
  SYNCHRONOUS:None:None
  ```

- `rpovalue` is the value sent to RP. Use the value `None` if mode is `SYNCHRONOUS`.
- `rpotype` is the RPO unit. Default is SECONDS. Value values are: `SECONDS MINUTES HOURS WRITES BYTES KB MB GB TB`. Use the value `None` if mode is `SYNCHRONOUS`.
- `journalsize` is journal size.
- `journal_varray` is an optional separate virtual array for the RecoverPoint copy journal volume. The default is the same virtual array as the RecoverPoint copy.

- `journal_vpool` is an optional separate virtual pool for the RecoverPoint copy journal volume. The default is the same virtual pool as the RecoverPoint copy.
- `standby_journal_varray` is the target virtual array.
- `standby_journal_vpool` is the target virtual pool.

This is an optional parameter.

metropoint

Flag to indicate whether or not MetroPoint configuration will be used. Specify `true` or `false` as the parameter value. You must also specify `ha`. This is an optional parameter.

activeProtectionAtHASite

Indicates whether or not to use the HA side of the VPLEX as the RecoverPoint protected site in an RP+VPLEX setup. In a MetroPoint context, if true, this field indicates that the HA VPLEX site will be the active site. Specify `true` or `false` as the parameter value. You must also specify `ha`. This is an optional parameter.

file_replication_policy|frpol

File replication policy, for example, `replicationtype:copymode:rpovalue:rpotype`. This is an optional parameter.

file_replication_copies

File replication remote copies, for example, `varray1:vpool1 varray2:vpool2`. This is an optional parameter.

systemtype|st

Type of the storage system used to create the vpool. It could be isilon, vnxblock, vnxfile, vmax, netapp, netappc, vplex, hds, openstack, scaleio, xtremio, ddmc, ibmxiv, vnxe, or datadomain. This is an optional parameter.

Note

Only when this parameter is set to `vmax`, can you define the `srdf` parameter.

---

raidlevel|rl

RAID level for the storage system used. Possible values are: `RAID1`, `RAID2`, `RAID3`, `RAID4`, `RAID5`, `RAID6`, or `RAID10`. This is an optional parameter.

fastpolicy|fp

Autotiering policy. Specify a fast policy only when you specify a system type. This is an optional parameter.

drivetype|dt

Supported drive type. This is an optional parameter. Supported types are SSD, FC, SAS, NL_SAS, SATA, or NONE.

type|t

Type of the virtual storage pool. Type of `file` or `block`. Default is `file` type. This is an optional parameter.

description|desc

A string describing the virtual storage pool. This is a mandatory parameter.

usematchedpools|ump

The virtual storage pool uses the matched pools. This is an optional parameter.

multivolconsistency|mvc

When included on the command line, resources provisioned from the pool support the use of consistency groups. If not included on the command line, a resource cannot be assigned to a consistency group when running ViPR Controller block provisioning services. This is an optional parameter.

expandable|ex

Indicates whether non-disruptive volume expansion should be supported. Native continuous copies are not supported. When not included, storage is selected based on performance over the ability to expand. This is an optional parameter.

autoCrossConnectExport|acc

Cross connect is enabled and export to both sides (true) or export to only one side (false). This is available only when there is High Availability VPLEX configuration, such as a shared VSAN between two VPLEX clusters in a VPLEX Metro configuraiton. This is an optional parameter.

fastexpansion|fe

Indicates that vpool volumes should use concatenated meta volumes, not striped. This is an optional parameter.

autotierpolicynames|apn

The Fully Automated Storage Tiering (FAST) policy for the virtual pool.

enablecompression

To only allow VMAX3 compression enabled storage pools to be included in the virtual pool.

maxpaths|mxp

The maximum number of paths that can be used between a host and storage volume. This is an optional parameter.

minpaths|mnp

The minimum number of paths that can be used between a host and storage volume. This is an optional parameter.

thinpreallocper|tpap

If you have `provisiontype` as `thin`, then this specifies the percentage of the physical storage to initially allocate to a volume. This is an optional percentage

frontendbandwidth|febw

Controls VMAX resource consumption at the storage group level by limiting the amount of front-end bandwidth that are consumed by the VMAX devices provisioned from this virtual pool. This value is measured in MB/s. To allow unlimited front-end bandwidth consumption, set this value to zero. This is an optional parameter.

Note

This parameter does not get set on a virtual pool that includes RecoverPoint protection.

---

iopersec|iops

Controls VMAX resource consumption at the storage group level by limiting the amount of I/Os per second (IOPS) that are consumed by the VMAX devices provisioned from this virtual pool. This value is measured in IOPS. To allow unlimited front-end I/O consumption, set this value to `0`. This is an optional parameter.

Note

This parameter does not get set on a virtual pool that includes RecoverPoint protection.

---

pathsperinitiator|ppi

The number of paths per host initiator. This is an optional parameter.

snapshotschedule|snapsched

Specifies whether to support a snapshot schedule. The allowable values are true and false. This is an optional parameter.

srdf

The VMAX SRDF protection parameters. There are two modes of protection: synchronous and asynchronous. This is available only between two VMAX arrays deployed in the datacenter. Each virtual storage array must have a protection vpool and policy. For example, myvarray:basepool:SYNCHRONOUS. This is an optional parameter.

Note

For using this parameter, you must have defined the `systemtype|st` parameter as **`VMAX`**.

-placementpolicy|pp

Resource placement policy (default_policy, or array_affinity) when set to array_affinity, provisioning will be performed based on the host/array affinity settings.

Common Arguments

This operation also takes the Common Parameters on page 39.

Example 1

This command creates a virtual storage pool with the NFS protocol and type file.

```
# viprcli vpool create -name SILVER -protocol NFS -t file -desc
"Protocol NFS"
```

Example 2 Using SRDF

This command creates a virtual storage pool with the SRDF options. This example shows a normal vpool creation first.

Example 2 Using SRDF (continued)

```
# viprcli vpool create -n basepool -t block -pl FC -pt Thick -va
myvarray -msnp 4

# viprcli vpool create -n srdfpool1 -t block -pl FC -pt Thick -va
myvarray -msnp 4 -st vmax -srdf myvarray:basepool:SYNCHRONOUS
```

Example 3 Using a High Availability VPLEX Configuration

This command creates a virtual storage pool with the `autoCrossConnectExport` option. This example shows auto cross connect enabled.

```
viprcli vpool create -name vplex_1_target1 -t block -pt Thin -pl FC
-desc "target pool" -ha vplex_local -autoCrossConnectExport true -
va Site1
```

Example 4 Creating a virtual pool for a journal volume

```
# viprcli vpool create -name myfull -protocol FC  -va  vmax573vplex154cl1varray -
provisiontype Thin -description test -type block  -rp vnx1303varray:my_base_new:
10GB:vmax573vplex154cl1varray:my_base_vplex -rp_source_policy
10GB:vmax573vplex154cl1varray:my_base_vplex:vmax573vplex154cl1varray:my_base_vplex:ASYNCHRO
NOUS:30:SECONDS -mvc true -ha vplex_distributed:vmax573vplex154cl1varray:my_base_vplex
```

Example 5 Creating a virtual pool for metropoint

```
# viprcli vpool create -n V1_rp_metropoint -t block -pl FC -pt Thin
-msnp 100 -mxp 1 -expandable true -mv true -acc true -desc
V1_rp_metropoint -va varray1 -ha
vplex_distributed:varray3:VP3_vplex -rp varray1:VP1_vplex:10GB
varray3:VP3_vplex:10GB -rp_source_policy ASYNCHRONOUS:25:SECONDS:
10GB:varray1:VP1_vplex:varray3:VP3_vplex -metropoint true -
activeProtectionAtHASite false
```

viprcli vpool update

Update the virtual pool.

Syntax

```
            viprcli vpool update
-name|n <vpoolname>
[-protocol_add|pa <protocol_add> [<protocol_add>...] {NFS | CIFS |
FC | iSCSI}]
[-protocol_remove|prm        <protocol_remove>
```

```
[<protocol_remove>...]]
[-varray_add|va_add <varray_add>        [<varray_add> ...] ]
[-varray_remove|va_rm        <varray_remove> [<varray_remove> ...] ]
[-usematchedpools|ump        <useMatchedPools>]
[-label|l <label>]
[-highavailability|ha        <highavailability>]
[-maxsnapshots|msnp        <max_snapshots>]
[-maxcontinuouscopies|mcc        <max_continuos_copies>]
[-type|t <vpooltype> { file |        block }]
[-description|desc        <description>]
[-mutlivolconsistency|mvc        <multivolconsistency>]
[-expandable|ex        <expandable>]
[-fastpolicy|fp        <fastpolicy>]
[-autotierpolicynames|apn        <unique_auto_tier_policy_names>]
[-enablecompression <enable_compression>]
[-maxpaths|mxp        <max_paths>]
[-minpaths|mnp        <min_paths>]
[-pathsperinitiator|ppi        <PathsPerInitiator>]
[-autoCrossConnectExport|acc        <autoCrossConnectExport> {true
| false}]
[-srdf_add <srdfadd> ]
[-srdf_remove <srdfremove>        ]
[-rp_add <rp_add> [<rp_add>        ...]]
[-rp_remove <rp_remove>        [<rp_remove> ...]]
[-rp_source_policy|rp_policy        <rp_source_policy>]
[-quota_enable|qe <quota_enable>        ]
[-quota_capacity|qc        <quota_capacity> ]
[-hostname|hn        <hostname>]
[-port|po <port_number>]
[-portui|pu        <ui_port_number>]
[-cookiefile|cf        <cookie_file>]
[-h]
[-placementpolicy|pp <placementpolicy>]
```

Description

`viprcli vpool update` command updates a virtual pool in ViPR Controller. This command is silent on success.

Options

name|n

The name of the virtual pool. This is a mandatory parameter.

protocol_add|pa

The protocol to be added to the virtual pool. For file storage, Protocol can be NFS or CIFS. For block storage, Protocol is FC or iSCSI. This is an optional parameter.

protocol_remove|prm

The protocol to be removed from the virtual pool. For file storage, Protocol can be NFS or CIFS. For block storage, Protocol is FC or iSCSI. This is an optional parameter.

varray_add|va_add

The virtual storage arrays to be added to the virtual pool. This is an optional parameter.

varray_remove|va_rm

The virtual storage arrays to be removed from the virtual pool. This is an optional parameter.

usematchedpools|ump

The virtual pool uses the matched pools.

label|l

New name for the virtual pool. This is an optional parameter.

maxsnapshots|msnp

The maximum number of native snapshots. This must be greater than '0'. This is an optional parameter.

maxcontinuouscopies|mcc

The maximum number of native continuous copies. You can set this only for a vpool of type Block.

Note

This is set to zero '0' by default. You must set it to a value greater than zero and set `expandable` to **`false`**, if you want to create continuous copy of a volume using `viprcli volume continuous_copies create` command with the protection type as native.

---

expandable|ex

Expandable file system or volume. This is set to true by default. This is an optional parameter.

Note

You must set `maxcontinuouscopies` to a value greater than zero and set `expandable` to **`false`**, if you want to create continuous copy of a volume using `viprcli volume continuous_copies create` command with the protection type as native.

---

fastpolicy|fp

Name of the FAST policy. This is an optional parameter.

highavailability|ha

Highavailability is available only for VPLEX arrays. The possible values are vplex_local or vplex_distributed. This is an optional parameter.

type|t

Type of the virtual storage pool. Type of file or block. Default is file type. This is an optional parameter.

description|desc

A string describing the virtual storage pool. This is an optional parameter.

multivolconsistency|mvc

Mutliple volume consistency. This is an optional parameter.

autotierpolicynames|apn

Enable or disable unique auto-tiering policies for FAST policy. This is an optional parameter.

enablecompression

To only allow VMAX3 compression enabled storage pools to be included in the virtual pool.

maxpaths|mxp

The maximum number of paths that can be used between a host and storage volume. This is an optional parameter.

minpaths|mnp

The minimum number of paths that can be used between a host and storage volume. This is an optional parameter.

pathsperinitiator|ppi

The number of paths per host initiator. This is an optional parameter.

autoCrossConnectExport|acc

Cross connect is enabled and export to both sides (true) or export to only one side (false). This is available only when there is High Availability VPLEX configuration, such as a shared VSAN between two VPLEX clusters in a VPLEX Metro configuraiton. This is an optional parameter.

srdf_add

Add VMAX SRDF protection parameters. This is available only between two VMAX arrays deployed in the datacenter. For example, myvarray:basepool:SYNCHRONOUS. This is an optional parameter.

srdf_remove

Delete VMAX SRDF protection parameters. For example, myvarray:basepool:SYNCHRONOUS. This is an optional parameter.

rp_add

Add RP protection parameters.

rp_remove

Remove RP protection parameters.

rp_source_policy|rp_policy

Sets parameters for RecoverPoint copies. The syntax is:

```
remotecopymode:rpovalue:rpotype:journalsize:journal_varray:journ
al_vpool:standby_journal_varray:standby_journal_vpool
```

- `remotecopymode` is the remote copy mode for the RecoverPoint CG. Values are: `ASYNCHRONOUS` (the default) or `SYNCHRONOUS`. When using `SYNCHRONOUS`, the next two parameters must be `None`, as follows:

  ```
  SYNCHRONOUS:None:None
  ```

- `rpovalue` is the value sent to RP. Use the value `None` if mode is `SYNCHRONOUS`.
- `rpotype` is the RPO unit. Default is SECONDS. Value values are: `SECONDS MINUTES HOURS WRITES BYTES KB MB GB TB`. Use the value `None` if mode is `SYNCHRONOUS`.
- `journalsize` is journal size.
- `journal_varray` is an optional separate virtual array for the RecoverPoint copy journal volume. The default is the same virtual array as the RecoverPoint copy.

- `journal_vpool` is an optional separate virtual pool for the RecoverPoint copy journal volume. The default is the same virtual pool as the RecoverPoint copy.
- `standby_journal_varray` is the target virtual array.
- `standby_journal_vpool` is the target virtual pool.

This is an optional parameter.

quota_enable|qe

Enable or disable quota. Possible values are: true or false. This is an optional parameter.

quota_capacity|qc

Quota capacity in GB. Quota reserves the maximum limit for allocated storage space for the specified vpool user. This is an optional parameter and used only when you set the `quota_enable` to true.

-placementpolicy|pp

Resource placement policy (default_policy, or array_affinity) when set to array_affinity, provisioning will be performed based on the array_affinity settings.

Common Arguments

This operation also takes the Common Parameters on page 39.

Example 6

This command updates the virtual storage pool 'silver' with CIFS protocol.

```
# viprcli vpool update -n silver -pa CIFS
```

The command is silent on execution. Use the list command as shown here to view the details.

```
# viprcli vpool list
NAME   TYPE PROTOCOLS TENANTS_ALLOWED
silver file CIFS,NFS
```

Example 7 Using a High Availability VPLEX Configuration

This command updates the virtual storage pool with the autoCrossConnectExport option. This example shows auto cross connect disabled so that the export is only to one side.

```
viprcli vpool update -name vplex_1_check -t block -desc "target
pool" -ha vplex_local -autoCrossConnectExport false -va_add Site1
```

Common Parameters

All viprcli commands take these optional parameters in the following table.

Table 8 List of Common Parameters

Parameters	Description
hostname|hn	The Fully Qualified Domain Name (FQDN), IPv4 or IPv6 address of a ViPR Controller host. This setting overrides the *ViPR_HOSTNAME* environment variable. This is an optional parameter. For IPv6 appliances, the IPv6 address must be entered using square brackets. For example [2520:0:140:280f::106]. **Note** 'hn' is not the short form for 'hostname' in the `viprcli host` commands.
port|po	The port of the ViPR Controller specified in the hostname argument. The default ViPR Controller port is 443. This is an optional parameter.
portui|pu	The https port of the ViPR Controller Portal UI. The default ViPR Controller port is 443. This is an optional parameter.
cookiefile|cf	A valid cookie file as returned by the viprcli authenticate operation. This is an optional parameter.
version|v	Version number of the ViPR Controller CLI program.
help|h	Show a help page for the command operation.

Inter-VSAN Routing (IVR) support

Improvements were made in the support of Cisco configurations that contain a mixture of IVR and non-IVR switches.

Specifically, you can now provision and export volumes in environments that include intermixed IVR and non-IVR switches.

The following documentation describes ViPR Controller support for IVR configurations. The scenarios that are presented have been tested and verified as fully supported in ViPR Controller 3.6.1.

ViPR Controller support for Inter-VSAN Routing with Cisco switches

ViPR Controller includes support for Inter-VSAN Routing (IVR) configurations using Cisco switches. IVR is designed to allow hosts, storage arrays, and other devices residing in separate VSANs to communicate without merging the VSANs.

Traditionally, VSANs are an abstraction to logically group storage ports and host initiators to allow communication between those ports and initiators. VSANs can span multiple switches when inter-connected and multiple VSANs can be configured on the same switch to logically separate out these entities based on business needs and other factors. In the traditional sense, no inter-VSAN communication was possible. However, with the introduction of Inter-VSAN Routing (IVR), storage ports and initiators can communicate even when they are in different VSANs.

IVR allows separate autonomous VSANs to maintain their own VSAN-wide services and provide the following benefits:

- Accesses resources across VSANs without compromising other VSAN benefits.
- Transports data traffic between specific initiators and targets on different VSANs without merging VSANs into a single logical fabric.
- Establishes proper interconnected routes that traverse one or more VSANs across multiple switches. IVR is not limited to VSANs present on a common switch.
- Shares valuable resources across VSANs without compromise. Fibre Channel traffic does not flow between VSANs, nor can initiators access resources across VSANs other than the designated VSAN.
- Provides efficient business continuity or disaster recovery solutions when used in conjunction with FCIP.
- IVR Is in compliance with Fibre Channel standards.

For more information on IVR, see http://www.cisco.com/c/en/us/td/docs/switches/datacenter/mds9000/sw/nx-os/configuration/guides/ivr/fm_ivr/fm_ivr/ivrbasic.html.

Prerequisites for IVR

The prerequisites for IVR communication are:

- The switches must support IVR and have the Cisco `ivr` feature enabled on them. See the Cisco documentation reference above for the switch and fabric OS versions that support IVR.
- Transit VSANs must be configured between IVR switches. See the Cisco documentation reference above for transit VSAN information and best practices.

Topologies supported in ViPR Controller

Three topologies are used to illustrate ViPR Controller support regarding IVR.

- Scenario 1: No IVR switches in the configuration
- Scenario 2: Only IVR switches in the configuration
- Scenario 3: Mix of non-IVR and IVR switches in the configuration.

These topologies are tested and verified in ViPR Controller 3.6.1 and later.

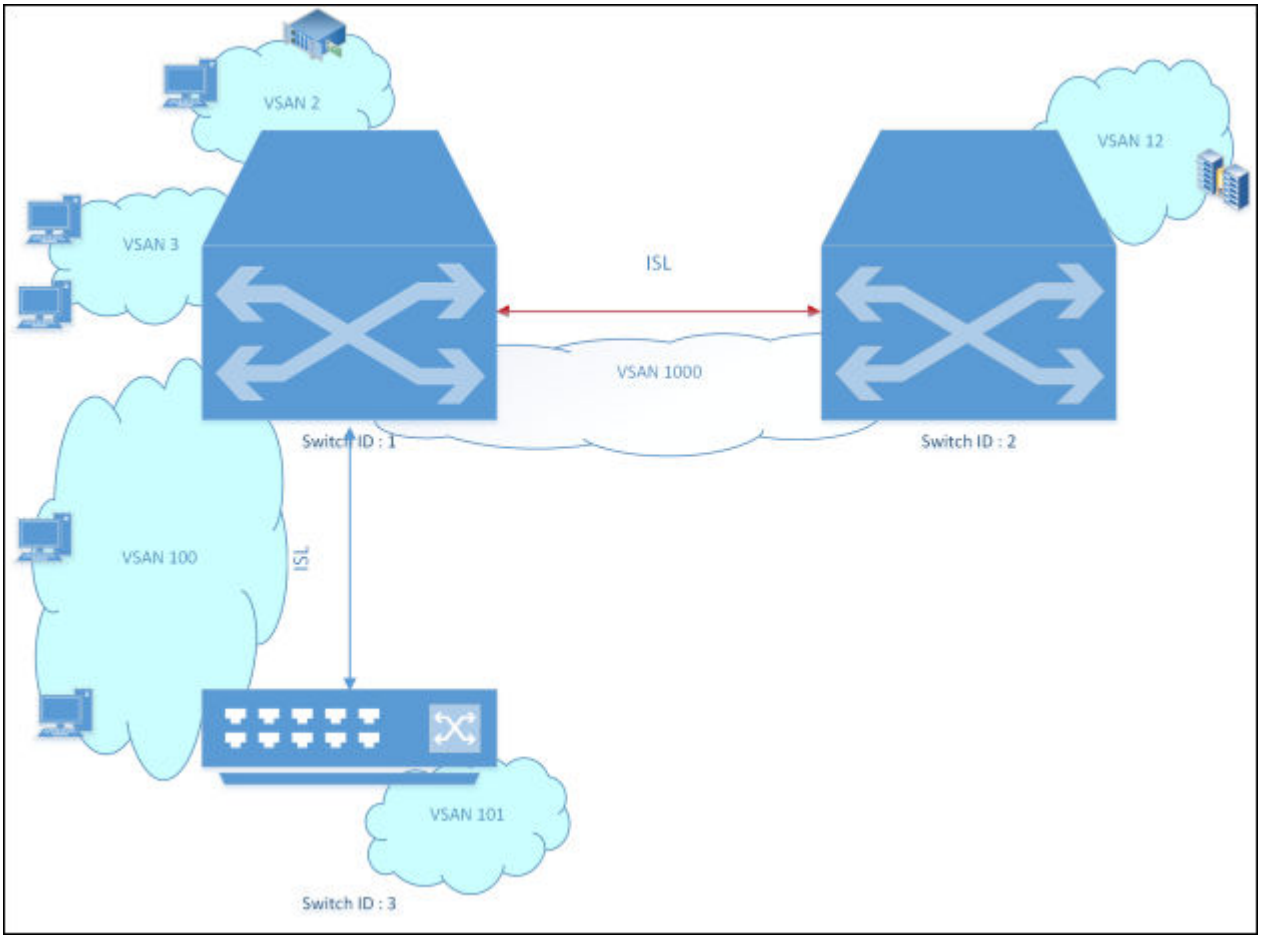
Note

Ingestion of IVR zones is not supported.

Scenario 1: No IVR switches in the configuration

In this case, there are multiple switches connected to each other with inter-switch links (ISLs), but none of the switches are capable of performing inter-VSAN routing.

Figure 1 Scenario 1—No IVR switches



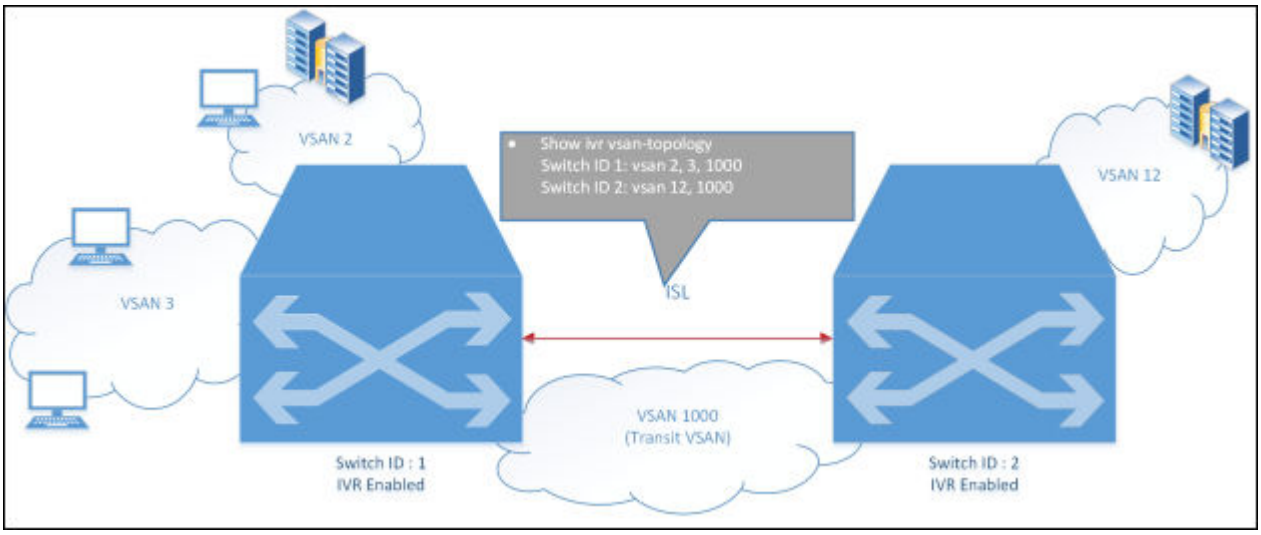
Notes:

- Because the switches are not IVR-capable, no routing is possible between, for example, VSAN3 on switch 1 to VSAN12 on switch 2.
- No routing is possible between VSAN2 and VSAN3 on switch 1.

Scenario 2: Only IVR switches in the configuration

All of the participating switches in this configuration are IVR-capable. In this configuration, technically, any VSAN can route to any other VSAN, provided that all of the switches are correctly configured and there are valid transit-VSANs between the switches.

Figure 2 Scenario 2—Only IVR switches



Notes:

- Switch 1 and Switch 2 are IVR-capable Cisco switches.
- Switch 1 and Switch 2 are interconnected and VSAN 1000 is the transit VSAN between the two switches. In an IVR configuration, the transit VSAN carries the traffic from a VSAN in one switch to a VSAN in another switch.
- In Switch 1, routing is possible between VSAN2 and VSAN3. This communication does not use the transit VSAN because all of the participating switches are local to the switch.
- Routing is also possible between VSAN2 on switch1 and VSAN12 on switch2. Internally, VSAN1000 is used to carry that traffic.
- The Cisco router `show ivr-vsan topology` command lists all of the switches in the IVR configuration and all the participating VSANs on those switches. In this case, the output would be:

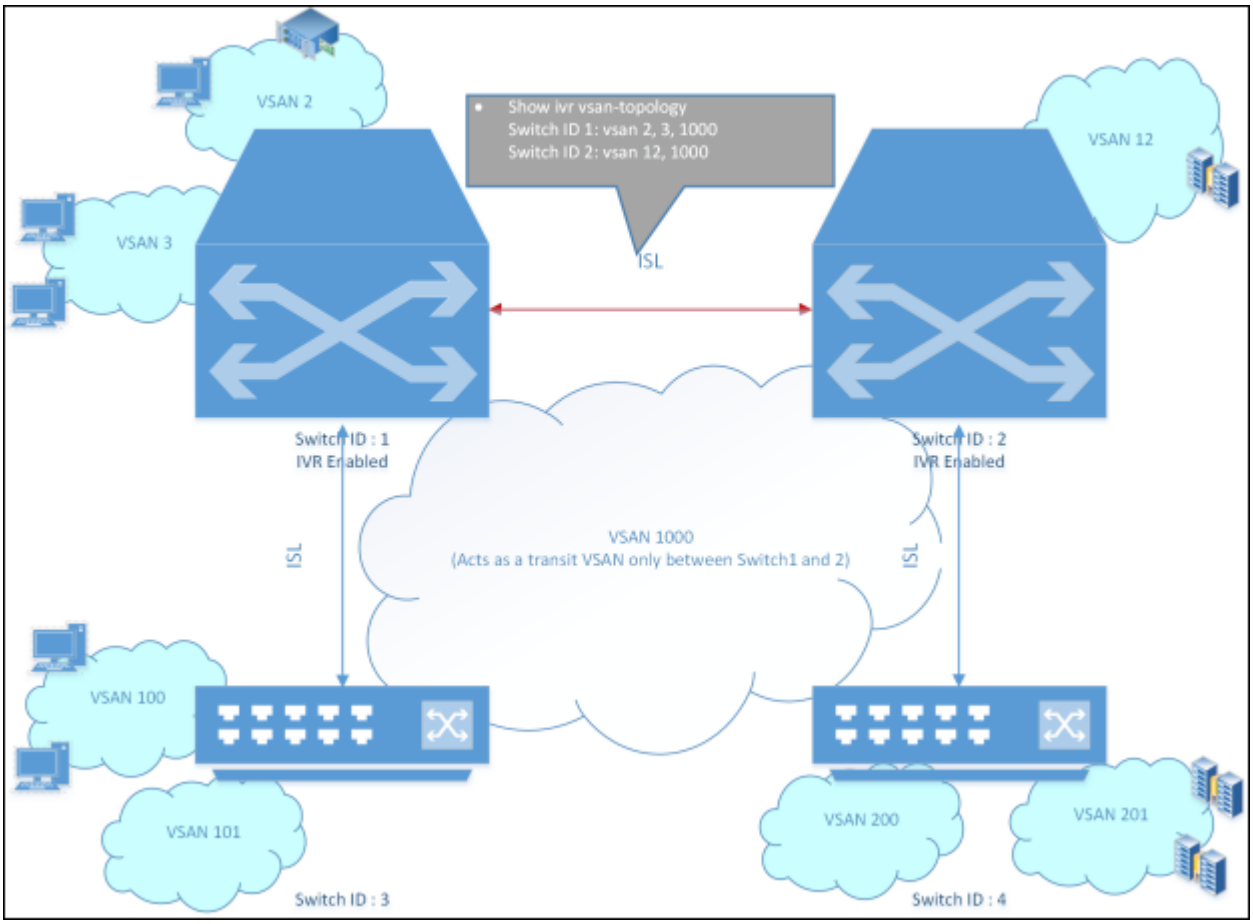
```
show ivr vsan-topology
Switch ID 1: vsan 2,3, 1000
Switch ID 2: vsan 12, 1000
```

- Hosts in VSAN3 can talk to storage in VSAN2 via Switch 1. VSAN2 and VSAN3 are adjacent VSANs.
- Hosts in VSANs 2 and 3 can talk to storage in VSAN 12 using the transit VSAN 1000 because they are on the IVR path.

Scenario 3: Mix of non-IVR and IVR switches in the configuration

This configuration includes both IVR-capable and non-IVR-capable switches connected to each other.

Figure 3 Scenario 3—Mix of non-IVR and IVR switches



Notes

- Switch 1 and Switch2 are IVR-capable switches. Switch 3 and Switch 4 are not IVR-capable.
- VSAN1000 is a transit VSAN between the IVR-capable switches; in this case, only between Switch 1 and Switch 2.
- VSANs on IVR switches are routable to each other.
- Storage ports/initiators from Switch 3 and Switch 4 and only on VSAN1000 can route to other VSANs on the IVR switches. Even though Switches 3 and 4 are not IVR-capable, VSAN 1000 is on the IVR path. Hence routing is possible between any port/initiators in VSAN 1000, even if they are physically on a non-IVR switch.
- VSANs such as VSAN100 and VSAN101 on Switch 3 are not routable to any other VSAN because they reside entirely on a non-IVR switch. The same restriction applies for VSAN200 and VSAN201 on Switch 4.
- The Cisco router `show ivr-vsan topology` command does not list any non-IVR switches. In this case, the output is:

```
show ivr vsan-topology
Switch ID 1: vsan 2, 3, 1000
Switch ID 2: vsan 12, 1000
```

About the show ivr-vsan topology command

The Cisco router `show ivr-vsan topology` command is only valid when executed against an IVR-capable switch. ViPR Controller relies on this command to fetch information about all of the VSANs on IVR-capable switches. ViPR Controller uses the information to determine which VSANs are routable to other VSANs.

Impact of changing virtual pool field values

Changing the field values populated during the change virtual pool operation has an impact when you compare the current or source virtual pool to the new or target virtual pool.

The change virtual pool operation has an impact on:

1. The current or source volume for which the change is requested.
2. The current or source virtual pool for the volume in #1 that it will be moved FROM.
3. The new or target virtual pool for the volume in #1 that it will be moved TO.

Note

Generally, the same virtual pool will always be excluded.

Use this key to determine which fields should or should not change when comparing the current or source virtual pool to the new or target virtual pool:

Category	Description
Special	Special change virtual pool conditions
Must change	Fields that must be different
Must NOT change	Fields that cannot be different
Ignore change	Fields indicated are ignored. It does not matter if they are the same or changed
Must be present	Fields that have to be on the new or target virtual pool

Operation: VPLEX common checks...(applies to most VPLEX operations)

Special: (RecoverPoint plus VPLEX) Must not change on HA virtual pool: `TYPE, VARRAYS, PROTECTION_VARRAY_SETTINGS, REF_VPOOL, MIRROR_VPOOL, FAST_EXPANSION, ACLS, INACTIVE, NUM_PATHS`

Special: (Plain VPLEX) Must not change on HA virtual pool: `TYPE, VARRAYS, REF_VPOOL, MIRROR_VPOOL, FAST_EXPANSION, ACLS, INACTIVE, NUM_PATHS`

Must NOT change: `TYPE, VARRAYS, REF_VPOOL, FAST_EXPANSION, ACLS, INACTIVE, NUM_PATHS, MIRROR_VPOOL` (only if specified on source virtual pool), `MULTI_VOLUME_CONSISTENCY` (only if specified on source virtual pool), `HIGH_AVAILABILITY` unless `VPLEX_LOCAL_TO_DISTRIBUTED`, otherwise, cannot change.

Operation: `virtualPoolChange.operation.VPLEX_LOCAL_TO_DISTRIBUTED=Change From VPLEX Local To VPLEX Distributed`

Special: Volume must not be in a consistency group.

Must change: `HIGH_AVAILABILITY` (Only VPLEX local to VPLEX distributed.)

Must NOT change: Everything else.

Ignore change: `PROTOCOLS, PROVISIONING_TYPE, USE_MATCHED_POOLS, ARRAY_INFO, DRIVE_TYPE, AUTO_TIER_POLICY_NAME, HOST_IO_LIMIT_IOPS, HOST_IO_LIMIT_BANDWIDTH, VMAX_COMPRESSION_ENABLED, IS_THIN_VOLUME_PRE_ALLOCATION_ENABLED, ASSIGNED_STORAGE_POOL`

Operation: **`virtualPoolChange.operation.VPLEX_DATA_MIGRATION=VPLEX Data Migration (Plain VPLEX)`**

Must change (At least one): `PROTOCOLS, PROVISIONING_TYPE, USE_MATCHED_POOLS, ARRAY_INFO, DRIVE_TYPE, AUTO_TIER_POLICY_NAME, HOST_IO_LIMIT_IOPS, HOST_IO_LIMIT_BANDWIDTH, VMAX_COMPRESSION_ENABLED, IS_THIN_VOLUME_PRE_ALLOCATION_ENABLED, ASSIGNED_STORAGE_POOLS`

Must NOT change: Everything else.

Operation: **`virtualPoolChange.operation.VPLEX_DATA_MIGRATION=VPLEX Data Migration (RP+VPLEX/MetroPoint)`**

Special: Volume must be RP (RecoverPoint)-protected.

Special: Source must specify the same RP Protection as Target (except virtual pools can change to indicate migrations at Source, Target, and Journal levels).

Special: Number of RP Targets must remain the same

Special: Cannot change virtual array for RP Targets and RP Journals

Special: ALL RP+VPLEX Migrations (Source, Target, Journal) are subject to regular VPLEX migration

Must NOT Change on RP Source: `TYPE, VARRAYS, REF_VPOOL, MIRROR_VPOOL, FAST_EXPANSION, ACLS, INACTIVE, NUM_PATHS, METROPOINT, HIGH_AVAILABILITY, RP_RPO_TYPE, RP_RPO_VALUE, RP_COPY_MODE`

Must NOT Change on RP Target:`TYPE, REF_VPOOL, MIRROR_VPOOL, FAST_EXPANSION, ACLS, INACTIVE, NUM_PATHS`

Must NOT Change on RP Journal: `YPE, REF_VPOOL, MIRROR_VPOOL, FAST_EXPANSION, ACLS, INACTIVE, NUM_PATHS`

Operation: **`virtualPoolChange.operation.NON_VPLEX_TO_VPLEX=Move Into VPLEX`**

Must change: `HIGH_AVAILABILITY`

Must NOT change: Everything else

Ignore change: `ACLS, ASSIGNED_STORAGE_POOLS, DESCRIPTION, HA_VARRAY_VPOOL_MAP, LABEL, MATCHED_POOLS, INVALID_MATCHED_POOLS, NUM_PATHS, STATUS, TAGS, CREATION_TIME, THIN_VOLUME_PRE_ALLOCATION_PERCENTAGE, NON_DISRUPTIVE_EXPANSION, AUTO_CROSS_CONNECT_EXPORT, MIRROR_VPOOL, REMOTECOPY_VARRAY_SETTINGS`

Operation: `virtualPoolChange.operation.NON_VPLEX_TO_VPLEX=Move Into VPLEX` and `virtualPoolChange.operation.RP_PROTECTED=Add RecoverPoint Protection`

Special: RP protection cannot already exist on the volume.

Special: Adding RP+VPLEX/MetroPoint protection to a non-VPLEX volume is not supported. Import to VPLEX first.

Special: Adding MetroPoint protection to a VPLEX Local volume is not supported. Upgrade to Distributed first.

Special: Apply VPLEX common checks if adding protection to a VPLEX volume.

Must change: `PROTECTION_VARRAY_SETTINGS`

Must NOT change: `TYPE, VARRAYS, REF_VPOOL, MIRROR_VPOOL, FAST_EXPANSION, ACLS, INACTIVE, PROTOCOLS, PROVISIONING_TYPE, USE_MATCHED_POOLS, ARRAY_INFO, DRIVE_TYPE, AUTO_TIER_POLICY_NAME, HOST_IO_LIMIT_IOPS, HOST_IO_LIMIT_BANDWIDTH, VMAX_COMPRESSION_ENABLED, IS_THIN_VOLUME_PRE_ALLOCATION_ENABLED, ASSIGNED_STORAGE_POOLS`

Operation: `virtualPoolChange.operation.RP_REMOVE_PROTECTION=Remove RecoverPoint Protection`

Special: Volume must be RP protected.

Special: Source must specify RP protection.

Special: Target must NOT specify RP protection.

Must NOT change: Everything else.

Ignore changes: `PROTECTION_VARRAY_SETTINGS, RP_RPO_VALUE, RP_RPO_TYPE, RP_COPY_MODE, ARRAY_INFO, DRIVE_TYPE, JOURNAL_SIZE, JOURNAL_VARRAY, JOURNAL_VPOOL, MULTI_VOLUME_CONSISTENCY, METROPOINT, STANDBY_JOURNAL_VARRAY, STANDBY_JOURNAL_VPOOL, ACLS, DESCRIPTION, LABEL, STATUS, TAGS, CREATION_TIME, INVALID_MATCHED_POOLS, MATCHED_POOLS, NON_DISRUPTIVE_EXPANSION`

Operation: `virtualPoolChange.operation.RP_UPGRADE_TO_METROPOINT=Change RecoverPoint Protection to MetroPoint`

Special: Source must be RP+VPLEX Distributed

Special: Target must be MetroPoint CRR

Special: Source must only specify one distributed RP target

Special: Target must only specify one distributed RP target

Special: RP Target virtual array and virtual pool (along with Journal varray/vpool) cannot have any changes

Must NOT change: `TYPE, VARRAYS, REF_VPOOL, MIRROR_VPOOL, HIGH_AVAILABILITY, FAST_EXPANSION, ACLS, INACTIVE, HA_CONNECTED_TO_RP, JOURNAL_SIZE`

Operation: `virtualPoolChange.operation.SRDF_PROTECED=Add SRDF Protection`

Special: Volume must have Storage Type=VMAX.

Must change: `REMOTECOPY_VARRAY_SETTINGS`

Must NOT change: `TYPE, VARRAYS, REF_VPOOL, MIRROR_VPOOL, HIGH_AVAILABILITY, PROTECTION_VARRAY_SETTINGS, FAST_EXPANSION, ACLS, INACTIVE, NUM_PATHS, PATHS_PER_INITIATOR, MIN_PATHS, AUTO_TIER_POLICY_NAME, HOST_IO_LIMIT_BANDWIDTH, HOST_IO_LIMIT_IOPS, MAX_COMPRESSION_ENABLED`

Operation: `virtualPoolChange.operation.ADD_MIRRORS=Add Continuous Copy Protection (VLEX Local)`

Must NOT change: `TYPE, VARRAYS, REF_VPOOL, HIGH_AVAILABILITY, PROTECTION_VARRAY_SETTINGS, FAST_EXPANSION, ACLS, INACTIVE, DRIVE_TYPE, ARRAY_INFO, PROVISIONING_TYPE, PROTOCOLS`

Must be present: `MATCHED_POOLS, ASSIGNED_STORAGE_POOLS`

Operation: `virtualPoolChange.operation.ADD_MIRRORS=Add Continuous Copy Protection (VPLEX Distributed)`

Special (VPLEX Distributed Source side): New vpool must have max native continuous copies() > 0

Special (VPLEX Distributed Source side): New vpool must have value present for mirror virtual pool

Must NOT change (VPLEX Distributed Source side): `TYPE, VARRAYS, REF_VPOOL, HIGH_AVAILABILITY, PROTECTION_VARRAY_SETTINGS, FAST_EXPANSION, ACLS, INACTIVE, DRIVE_TYPE, ARRAY_INFO, PROVISIONING_TYPE, PROTOCOLS`

Special (VPLEX Distributed HA side): New vpool must have max native continuous copies() > 0

Special (VPLEX Distributed HA side): New vpool must have value present for mirror virtual pool

Must NOT change (VPLEX Distributed HA side): `TYPE, VARRAYS, REF_VPOOL, PROTECTION_VARRAY_SETTINGS, FAST_EXPANSION, ACLS, INACTIVE, DRIVE_TYPE, ARRAY_INFO, PROVISIONING_TYPE, PROTOCOLS`

Operation: `virtualPoolChange.operation.EXPORT_PATH_PARAMS=Change Export Path Parameters`

Must change: `NUM_PATHS, PATHS_PER_INITIATOR`

Must NOT change: Everything else

Ignore changes: `NUM_PATHS, PATHS_PER_INITIATOR, MIN_PATHS, THIN_VOLUME_PRE_ALLOCATION_PERCENTAGE, ACLS, DESCRIPTION, LABEL, STATUS, TAGS, CREATION_TIME, INVALID_MATCHED_POOLS, MATCHED_POOLS, NON_DISRUPTIVE_EXPANSION`

Operation: `virtualPoolChange.operation.AUTO_TIERING_POLICY=Change Auto-tiering Policy, Host IO Limits or Compression Setting`

Special: The system_type between source vPool/Volume and target vPool must be the same.

Special: Target vPool must have Volume's Storage Pool in its matched/assigned pools list. Also applies to HA vpool for VPLEX.

Must change: `AUTO_TIER_POLICY_NAME, HOST_IO_LIMIT_BANDWIDTH, HOST_IO_LIMIT_IOPS, VMAX_COMPRESSION_ENABLED`

Must NOT change: Everything else.

Ignore changes: `AUTO_TIER_POLICY_NAME, HOST_IO_LIMIT_BANDWIDTH, HOST_IO_LIMIT_IOPS,VMAX_COMPRESSION_ENABLED, ARRAY_INFO, UNIQUE_AUTO_TIERING_POLICY_NAMES, ASSIGNED_STORAGE_POOLS, USE_MATCHED_POOLS, THIN_VOLUME_PRE_ALLOCATION_PERCENTAGE, ACLS, DESCRIPTION, LABEL, STATUS, TAGS, CREATION_TIME, INVALID_MATCHED_POOLS, MATCHED_POOLS, NON_DISRUPTIVE_EXPANSION, AUTO_TIER_POLICY_NAME, HOST_IO_LIMIT_BANDWIDTH, HOST_IO_LIMIT_IOPS,VMAX_COMPRESSION_ENABLED, ARRAY_INFO, UNIQUE_AUTO_TIERING_POLICY_NAMES, ASSIGNED_STORAGE_POOLS, USE_MATCHED_POOLS, THIN_VOLUME_PRE_ALLOCATION_PERCENTAGE, RP_RPO_VALUE, PROTECTION_VARRAY_SETTINGS, HA_VARRAY_VPOOL_MAP`, except for HA virtual pool of VPLEX.

Operation: `virtualPoolChange.operation.REPLICATION_MODE=Change Replication Mode for RecoverPoint Volumes`

Special: Both the source and target vpools must specify RP protection.

Must change: `RP_COPY_MODE`

Must NOT change: Everything else.

Ignore changes: `RP_COPY_MODE, RP_RPO_VALUE, RP_RPO_TYPE, PROTECTION_VARRAY_SETTINGS, ACLS, DESCRIPTION, LABEL, STATUS, TAGS, CREATION_TIME, INVALID_MATCHED_POOLS, MATCHED_POOLS, NON_DISRUPTIVE_EXPANSION`

ViPR Controller ports

Correct operation of ViPR Controller and its services requires certain ports to be open in the firewall. When installed, these ports are automatically configured.

ViPR Controller Authentication Provider Ports

These ports are the default listening (incoming) ports in the external AD or LDAP through which ViPR Controller tries to establish the connection. When the AD or LDAP is not listening through these default ports, the server_url (ldap(s)://<ip:port>) in the ViPR Controller authorization provider configuration can be modified to specify ports other than the defaults.

Table 9 ViPR Controller Authentication Provider Ports

Port	Protocol	Direction	Description
88	TCP and UDP	Outbound	Domain Controller to which ViPR Controller connects during Windows host discovery for Kerberos authentication
389	TCP	Outbound	For non-secure communication with external authentication providers like AD or LDAP
636	TCP	Outbound	For secure communication (SSL) with external authentication providers like AD or LDAP
35357	TCP	Outbound	Keystone (OpenStack Authentication Provider)

ViPR Controller VM ports

ViPR Controller VM ports are deployed with the firewall enabled by default, with these ports open:

Note

Ports exposed to outside ViPR nodes are: 7, 22, 25, 123, 162, 443, 500, 990, 4443, 4500, 7012, 7100, 8776, 9083, and 9998. All other ports are ViPR inter-nodes connections.

Table 10 ViPR Controller VM ports

Port	Protocol	Direction	Description
7	UDP	Inbound	echo protocol
22	TCP	Inbound	SSH port
25	TCP	Outbound	SMTP port
123	UDP	Bi-directional	NTP
162	UDP	Outbound	SNMP
443	TCP	Bi-directional	Standard HTTPS port to be redirected to 4443
500	UDP, IPsec	Bi-directional	IPsec
990	FTPS	Outbound	ConnectEMC - Outbound Only
2181 and 2889	TCP	Bi-directional	Coordinator service
2888	TCP	Bi-directional	Zookeeper peers connect to each other
1098	TCP		Internal communication port. Open manually only for JMX in development
1443	TCP		Nginx uses for accessing the REST apis
6080	TCP		REST interface for S3 API
4443	TCP	Bi-directional	Reverse Proxy/Load balancer for ViPR REST APIs ; GUI port
4500	UDP, IPsec	Bi-directional	IPsec

Table 10 ViPR Controller VM ports (continued)

Port	Protocol	Direction	Description
5000	UDP	Inbound	CIM adapter for internal nodes
6443	TCP	Inbound	ViPR Controller user interface
7000	TCP	Bi-directional	DB service
7001	TCP	Bi-directional	DB SSL
7012	TCP	Inbound	CIM adapter
7100	TCP	Bi-directional	GEO (across VDCs) database connections; protected by IPsec
7199	TCP	Bi-directional	DB service
7200	TCP	Bi-directional	DB service
7299	TCP	Bi-directional	JMX server and register ports
7300	TCP	Bi-directional	JMX server and register ports
7399	TCP	Bi-directional	Coordinator service
7400	TCP	Bi-directional	Coordinator service
7443	TCP	Inbound	Authentication service
8080	TCP	Inbound	API service
8443	TCP	Inbound	API service
8444	TCP	Outbound	SA service
8543	TCP	Bi-directional	Nginx
8776	TCP	Bi-directional	Cinder-compatible REST API
9083	TCP	Inbound	VASA service
9093	TCP	Inbound	VASA service
9160	TCP	Bi-directional	DB service
9260	TCP	Bi-directional	Geo DB service
9993	TCP	Bi-directional	sys service
9998	TCP	Bi-directional	syssvc CLI download (unauthenticated)
10099	TCP	Bi-directional	Controller service
10099	TCP		Controller service
10100	TCP		Open manually only for JMX in development on extranode
10101	TCP		Open manually only for JMX in development on extranode
40201	TCP	Bi-directional	Controller service

Firewall ports required to be open for implementing ViPR Controller Disaster Recovery

ViPR Controller uses these ports in support of disaster recovery.

Certain ports are required to be opened in bi-directional fashion for ViPR Controller to be deployed in the presence of firewalls. (For example, when replicating data to another physical location.)

- All active or standby nodes should have unique IP addresses and be reachable by others. This requirement supports "hot" standby required by Cassandra/Zookeeper replication.
- Ensure that you have quality speed network infrastructure between datacenters. The maximum supported latency between Disaster Recovery (DR) sites is less than or equal to 150ms. This supports the synchronous replication of Cassandra/Zookeeper and storage management to remote sites.
- NAT across data centers is not supported. No ViPR Controller nodes can be behind NAT proxy. This is a requirement for Cassandra/Zookeeper replication.
- Ports 2888(ZK), 2889(ZK), 7100(dbsvc), 7000(geodbsvc), 500(udp, ipsec), 4500(udp, ipsec) should be allowed for all nodes in remote data center on firewall for ViPR Controller data replication. Allow port 443 (HTTPS) access on the cross-datacenter firewall to issue inter-site control commands.
- For ports 500 and 4500, the firewall must allow protocols 17 (UDP) and 50 (IPSec) through between all nodes of ViPR Controller Primary and ViPR Controller Disaster Recovery sites.
- The Disaster Recovery instance of ViPR Controller is subject to the same firewall port open requirements as the main instance. Otherwise the failover may work but management of infrastructure from the Disaster Recovery side may fail due to ports being closed by firewalls. Test the ViPR Controller failover periodically to confirm the management of infrastructure from the remote ViPR Controller instance works as expected.

Ensure that you have quality speed network infrastructure between datacenters. NAT across data centers is not supported. The maximum supported latency between System Disaster Recovery (DR) sites is <= 150ms.

Table 11 ViPR Controller ports required to be open between disaster recovery sites

Port	Protocol	Direction	Description
443	TCP	Bi-directional	Standard HTTPS port to be redirected to 4443
500	UDP, IPsec	Bi-directional	IPsec **Note** For ports 500 and 4500, the settings for UDP, IPSEC are 17, 50 respectively in the firewall rules, not just 17. This is important in firewalls that are based on port and protocol, and not port only.
2888	TCP	Bi-directional	Zookeeper (Zookeeper peers connect to each other)
2181 and 2889	TCP	Bi-directional	Coordinator service

Table 11 ViPR Controller ports required to be open between disaster recovery sites (continued)

Port	Protocol	Direction	Description
4500	UDP, IPsec	Bi-directional	IPsec **Note** For ports 500 and 4500, the settings for UDP, IPSEC are 17, 50 respectively in the firewall rules, not just 17. This is important in firewalls that are based on port and protocol, and not port only.
7000	TCP	Bi-directional	DB Service
7100	TCP	Bi-directional	Outbound Virtual data center to virtual data center communication port used for GEO and System Disaster Recovery sites communication. **Note** This port must be open for inbound and outbound traffic.

Storage-related ports

ViPR Controller uses these storage ports.

Table 12 ViPR Controller storage-related ports

Port	Protocol	Direction	Description
22	TCP	Outbound	ScaleIO, non-SSL
22	TCP	Outbound	Cinder; third-party block discovery
22	TCP	Outbound	Cisco switches
443	TCP	Outbound	ScaleIO, SSL
443	TCP	Outbound	VNX File
443	TCP	Outbound	VPLEX
443	TCP	Outbound	XtremIO
443	TCP	Outbound	NetApp
443	TCP	Outbound	EMC Unity
2001	TCP	Outbound	Hitachi
3033	TCP	Outbound	The ViPR Controller Dell SC driver uses the REST API available with Dell Storage Manager 2015 R3 or above. All API communication uses HTTPS over port 3033.
5000	TCP	Outbound	Keystone; third-party block authentication
5988	TCP	Bi-directional	VNX File
5988	TCP	Bi-directional	VNX Block, non-SSL

Table 12 ViPR Controller storage-related ports (continued)

Port	Protocol	Direction	Description
5988	TCP	Bi-directional	VMAX, non-SSL
5989	TCP	Bi-directional	VNX File
5989	TCP	Outbound	SMI-S for XIV, SSL
5989	TCP	Bi-directional	VNX Block, SSL
5989	TCP	Bi-directional	VMAX, SSL
7100	TCP	Outbound	Virtual data center to virtual data center communication port used for GEO and System Disaster Recovery sites communication **Note** This port must be open for inbound and outbound traffic.
7225	TCP	Outbound	RecoverPoint
8080	TCP	Outbound	Isilon
8443	TCP	Outbound	Hyper-Scale Manager for XIV, REST API
9998	TCP	Outbound	CLI download (unauthenticated)

Host access ports

ViPR Controller accesses hosts over the following ports.

Table 13 ViPR Controller host access ports

Port	Protocol	Direction	Description
22	TCP	Bi-directional	Linux, AIX, AIX VIO, HP-UX, SSH port
22	TCP	Bi-directional	AIX
22	TCP	Bi-directional	AIX VIO
22	TCP	Bi-directional	HP-UX
443	TCP	Bi-directional	vCenter HTTP port
5985	TCP	Bi-directional	Windows WinRM HTTP port
5986	TCP	Bi-directional	Windows WinRM HTTPS port

Fabric provider ports

ViPR Controller accesses fabric providers over the following ports.

Table 14 ViPR Controller fabric provider ports

Port	Protocol	Direction	Description
22	TCP	Bi-directional	Cisco MDS

Table 14 ViPR Controller fabric provider ports (continued)

Port	Protocol	Direction	Description
5988	TCP	Bi-directional	Brocade SMI-S provider, for discovering Brocade switches (non-SSL)
5989	TCP	Bi-directional	Brocade SMI-S provider, for discovering Brocade switches (SSL)

Compute Image server ports

ViPR Controller accesses compute image servers over the following ports.

Table 15 ViPR Controller compute image server ports

Port	Protocol	Direction	Description
22	TCP	Outbound	SSH

Block Storage Services > Unexport and Remove Block Volume

You can run the **Unexport and Remove Block Volume** service to remove a VPLEX virtual volume built with the Create Block Volume or Create Block Volume for Host services.

Note

VPLEX virtual volumes that are deleted outside of ViPR Controller or virtual volumes that were partially removed during a failed ViPR Controller delete volume workflow, may take several hours to clean up.

To run the **Unexport and Remove Block Volume** service:

1. Log in as a ViPR Controller tenant administrator.
2. From the User menu, choose **Block Storage Services** > **Unexport and Remove Block Volume**.
3. Choose the project that contains the volume to remove.
4. Click the check box next to the volume to remove.
5. Choose one of the following Deletion Types:

Full	• Deletes the Volume from ViPR Controller. • Deletes the Virtual Volume from the VPLEX . • Deletes all exports, snapshots, clones, and mirrors associated with the volume. • Deletes the physical back end storage from the physical arrays, including removing the volume from consistency groups.

Inventory Only	Deletes the volume from ViPR Controller. All VPLEX and physical storage objects are left intact.

6. Click **Order.**

ViPR requirements for service profile templates

The following sections explain the requirements to configure a service profile template for ViPR Controller provisioning operations.

Note

If existing service profile templates do not match the following requirements, clone one of the service profile template to create a new service profile template and alter the settings as required by ViPR Controller.

General properties

- The service profile template must not be associated to a server pool. Blade selection is performed by the ViPR Controller Compute Virtual Pools.
- UUID assignment must be from a valid UUID Suffix Pool set up in the UCS with available addresses.
- To avoid issues with ESX host discovery when using multiple domains, ensure the UUID pools are unique across domains. When provisioning, ensure there are unique UCS pools (UUID, MAC, WWNN, WWPN, and so forth).

Storage

ViPR Controller currently supports Fibre Channel boot for UCS servers. The following lists the Fibre Channel requirements:

- World Wide Node Name (WWNN) assignment must be from a valid UUID Suffix Pool set up in the UCS with available addresses.
- The Local Disk Configuration Policy must be set to a local disk configuration policy where the **Mode** is set to **No Local Storage**.
- There must be at least one vHBA interface.
- For each vHBA, the World Wide Port Name (WWPN) assignment must be from a valid WWPN pool set up in the UCS with available addresses.
- The VSAN set on each vHBA must be a valid network discovered by ViPR Controller. The VSAN must match one of the networks in a ViPR Controller virtual array.
- Policy settings on the vHBAs are not set by ViPR Controller provisioning and are at the administrator's discretion.

Network

- Policy settings on the vNICs are not set by ViPR Controller provisioning and are at the administrator's discretion.
- There must be at least one vNIC interface.
- For each vNIC, the MAC Address Assignment must be from a valid MAC pool that was set up in the UCS with available addresses.
- Each vNIC must have at least one VLAN.

Boot Policy and Boot Order

There are no Boot Policy requirements. ViPR Controller ignores all Boot Policy settings in the service profile template and overwrites any existing parameters when it creates service profiles.

Policies

ViPR Controller does not set any policies. The UCS administrator is responsible for setting the policies.

Updating service profile templates

If provisioning with updating service profile templates,

- The boot policy of the updating service profile template must specify SAN as the first boot device.
- If the boot policy of the updating service profile template enforces vNIC and vHBA names, the names of the vNICs and vHBAs in the service profile template must match those in its boot policy.
- The compute virtual pool with which the updating service profile template is being associated, must be associated to a virtual array that has storage ports on the VSANs that the vHBAs of the template use.
- If the boot policy of the updating service profile template specifies SAN boot target WWPNs, then compute virtual pool that the template is associated with must be associated with a virtual array that includes those storage ports on the appropriate VSANs.