



EMC ViPR Controller

Version 2.4.0.0

ViPR Controller REST API Virtual Data Center Configuration Guide

302-002-417

01

Copyright © 2013-2015 EMC Corporation. All rights reserved. Published in USA.

Published November, 2015

EMC believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

The information in this publication is provided as is. EMC Corporation makes no representations or warranties of any kind with respect to the information in this publication, and specifically disclaims implied warranties of merchantability or fitness for a particular purpose. Use, copying, and distribution of any EMC software described in this publication requires an applicable software license.

EMC², EMC, and the EMC logo are registered trademarks or trademarks of EMC Corporation in the United States and other countries. All other trademarks used herein are the property of their respective owners.

For the most up-to-date regulatory document for your product line, go to EMC Online Support (<https://support.emc.com>).

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

CONTENTS

Chapter 1	Virtual Data Center Configuration Overview	7
Chapter 2	Authenticating with the REST API	9
	Overview.....	10
	Authenticate with cookies.....	10
	Authenticate without cookies.....	11
	How to handle an HTTP 302 authentication redirect.....	12
	Handle authentication redirects using cookies.....	12
	Handle authentication redirects without using cookies.....	13
	Logout.....	14
	A ViPR Controller user can view their own name, tenant associate, and roles using the whoami API call.....	15
Chapter 3	Customizing the names of resources created by ViPR Controller on physical systems	17
	Customizing resource names created on physical systems.....	18
	Naming policy syntax.....	19
	Available functions.....	20
	Add a custom naming convention using the REST API.....	21
	Additional REST API calls to manage custom naming conventions.....	25
Chapter 4	Adding storage systems	27
	Data Domain prerequisites and sample REST API request	28
	Elastic Cloud Storage prerequisites and sample REST API request.....	28
	Hitachi Data Systems prerequisites and sample REST API request.....	29
	IBM XIV prerequisites and sample REST API request.....	29
	EMC Isilon prerequisites and sample REST API request.....	30
	NetApp prerequisites and sample REST API request.....	31
	ScaleIO prerequisites and sample REST API request	31
	Best practices for ScaleIO to work with discoverable hosts.....	32
	How ScaleIO components are integrated into ViPR Controller.....	32
	Third-party block storage prerequisites and sample REST API request.....	33
	VNX for Block and VMAX prerequisites and sample REST API request	33
	VNX for File prerequisites and sample REST API request.....	34
	VNXe prerequisites and sample REST API request.....	35
	VPLEX prerequisites and sample REST API request.....	35
	EMC XtremIO prerequisites and sample REST API request.....	36
	Network configuration for block storage systems.....	36
	IP network configuration requirements for storage.....	37
	Important REST API calls to manage and configure storage systems and storage providers.....	37
	vNAS server REST API calls.....	38
Chapter 5	Adding a data protection system	41
	Adding a RecoverPoint data protection system.....	42

Chapter 6	Adding network systems (fabric managers) and SAN networks	45
	Overview.....	46
	Add a network system (fabric manager)	46
	Manually creating and updating networks in ViPR Controller.....	48
	Important REST API calls for managing network systems (fabric managers)....	50
Chapter 7	Adding hosts and clusters	53
	Overview.....	54
	Add and discover a LINUX host.....	54
	Add and discover a Windows host.....	57
	Add an AIX host or AIX VIO server to ViPR Controller.....	59
	Add a host other than Windows, AIX or Linux.....	62
	Add host ports to an IP network.....	65
	Add a host to a cluster.....	65
	Replace Host Initiators after a storage volume has been exported to a host... Update ViPR Controller after a host initiator is replaced outside of ViPR	67 67
	Replace a host initiator of an undiscovered host after a ViPR Controller export operation.....	71
	Important REST API calls to manage hosts and clusters.....	79
Chapter 8	Adding vCenters and ESX/ESXi clusters	81
	Overview.....	82
	Add and discover a vCenter Server.....	82
	Network configuration requirements for vCenters.....	84
	Important REST API calls to manage vCenters and clusters.....	85
Chapter 9	Configuring and adding Vblock System Components	87
	Configure and add Vblock system components.....	88
	Add a Vblock compute system to ViPR Controller..... Additional REST API calls to manage your Vblock Compute Systems	88 91
	Add Compute Image Servers..... Additional API calls to manage your Compute Image Servers.....	92 94
	Add Compute Images to ViPR Controller..... Additional REST API calls to manage your Compute Images.....	94 97
Chapter 10	Creating and configuring a virtual array	99
	Overview.....	100
	Create a virtual array and assign an ACL.....	100
	Add and configure the networks for block storage in the virtual array..... Add an existing fibre channel network to a virtual array..... Configure a new fibre channel network for a virtual array.....	102 102 106
	Add and configure the networks for file storage in a virtual array..... Add an existing IP network to a virtual array..... Configure a new IP network for a virtual array.....	108 108 111
	Optionally restrict access to storage pools by manually assigning them to a virtual array.....	114
	Virtual Array requirements for Vblock system services.....	121

Chapter 11	Creating virtual pools for block storage	123
	Create ViPR Controller Virtual Pools for Block Storage.....	124
	Important REST API calls to manage and configure block virtual pools.....	132
Chapter 12	Creating virtual pools for file storage	135
	Create ViPR Controller virtual pools for file storage.....	136
Chapter 13	Creating virtual pools for Vblock Compute Systems	143
	Create a Compute Virtual Pool using the REST API.....	144
	UCS blades and service profile templates.....	146
	ViPR requirements for service profile templates.....	146
	Additional REST API calls to manage your Compute Virtual Pools.....	147
Chapter 14	Creating virtual pools for object storage	149
	Create ViPR Controller Virtual Pools for Object Storage.....	150
Chapter 15	Assigning an Access Control List (ACL) to a resource	157
	ACLs.....	158
	Virtual array and virtual pool ACLs.....	158
	Examples: Virtual array and virtual pool ACL APIs.....	159
	Project ACLs.....	160
	Examples: Project ACL APIs.....	161
	Examples: Changing a project's owner.....	162
Chapter 16	Assigning roles to users and groups	165
	Overview of assigning roles using the REST API.....	166
	Assign a Virtual Data Center Role to a User or Group.....	166
	Assign a Tenant Role to a User or Group.....	167
Chapter 17	Using Tasks to track the status of asynchronous operations	169
	Overview.....	170
	Track status of asynchronous operations using task details.....	170
	Search for tasks by resource ID or state.....	172
	Get statistics for tenant and system tasks.....	174
	Delete a task that is stuck in the pending state.....	175
	Change task-related configuration settings.....	175
	Important REST API calls to manage tasks.....	176
Chapter 18	Understanding and setting up port selection using performance metrics	179
	Overview.....	180
	How does ViPR Controller select a port when using performance metrics.....	180
	Global default port selection.....	183
	Set up metering prerequisites in ViPR Controller.....	184
	Change the default port allocation parameters using the REST API.....	185
	Prerequisites for VNX and HDS metrics-based port selection.....	188
	HDS performance metrics.....	188
	VMAX performance metrics.....	189

CONTENTS

VNX for Block performance metrics.....190

CHAPTER 1

Virtual Data Center Configuration Overview

After EMC ViPR Controller installation, ViPR Controller System Administrators, and Tenant Administrators can use the ViPR Controller UI, or ViPR Controller REST API to configure the ViPR Controller Virtual Data Center (VDC).

This document provides the steps to configure the ViPR Controller VDC using the ViPR Controller REST API. Before you begin to configure your VDC:

- Review the information, and requirements in the *ViPR Controller Virtual Data Center Requirements and Information Guide* on the [ViPR Controller Product Documentation Index](#).
- For steps to configure the VDC using the ViPR Controller UI, see the *ViPR Controller User Interface Virtual Data Center Configuration Guide* on the [ViPR Controller Product Documentation Index](#).

You can add any of the physical or virtual assets to ViPR Controller at anytime. However, if you are setting up the VDC for the first time, you should follow these steps:

Procedure

1. Authenticate with the ViPR Controller REST API with credentials that match the roles required for the REST requests used to setup the virtual data center. The root user for your ViPR Controller has all of the role assignments you need. See [Authenticating with the REST API on page 9](#).
2. Add physical assets to ViPR.
 - [Adding storage systems on page 27](#)
 - [Adding a data protection system on page 41](#)
 - [Adding network systems \(fabric managers\) and SAN networks on page 45](#)
 - [Adding hosts and clusters on page 53](#)
 - [Adding vCenters and ESX/ESXi clusters on page 81](#)
 - For VCE Vblock System, [Configuring and adding Vblock System Components on page 87](#) and [Add Compute Images to ViPR Controller on page 94](#).
3. Create ViPR virtual assets
 - a. [Creating and configuring a virtual array on page 99](#)
 - b. Create virtual pools.
 - [Creating virtual pools for block storage on page 123](#)
 - [Creating virtual pools for file storage on page 135](#)
 - [Creating virtual pools for Vblock Compute Systems on page 143](#)
 - [Create ViPR Controller Virtual Pools for Object Storage on page 150](#)

CHAPTER 2

Authenticating with the REST API

This chapter contains the following topics:

- [Overview](#)..... 10
- [Authenticate with cookies](#)..... 10
- [Authenticate without cookies](#)..... 11
- [How to handle an HTTP 302 authentication redirect](#)..... 12
- [Logout](#)..... 14
- [A ViPR Controller user can view their own name, tenant associate, and roles using the whoami API call](#)..... 15

Overview

ViPR Controller uses a token-based authentication system for all its REST API calls. Examples are provided for authentication with the ViPR Controller REST API, with cookies and without cookies.

Once a user is authenticated against ViPR Controller, an authentication token is returned and can be used to authenticate the user in subsequent calls.

Authentication tokens expire after eight hours or after two hours of idle time. Once expired, the token is internally destroyed. If a client makes a request with the expired token:

- An HTTP 401 code is returned if the client is automatically following redirects, indicating that you need to login and authenticate to obtain a new token.
- An HTTP 302 code is returned if the client is not automatically following redirects. The 302 code directs the client to where to get re-authenticated.

Note

If running a REST API script, you can get a proxy token to run the script. A proxy token does not expire. See [Use a proxyuser to run a ViPR Controller REST API script](#).

You can retrieve and use authentication tokens by:

- Saving the X-SDS-AUTH-TOKEN cookie from a successful authentication request and sending that cookie along in subsequent requests.
- Reading the X-SDS-AUTH-TOKEN HTTP header from a successful authentication request and copying that header into any subsequent request.

The *EMC ViPR Controller REST API Reference* on the [ViPR Controller Product Documentation Index](#) provides a description and complete list of parameters for the REST API methods used in the following examples.

Authenticate with cookies

This example shows how to use authentication tokens by saving the cookie from a successful authentication request, then passing the cookie in a subsequent request. The examples here are written in `curl` and formatted for readability.

In this example, you specify the `?using-cookies=true` parameter to indicate that you want to receive cookies in addition to the normal HTTP header. This `curl` command saves the authentication token to a file named `cookiefile` in the current directory.

```
curl -L --location-trusted
-k https://<ViPR_Controller_VIP>:4443/login?using-cookies=true
-u "root:Password"
-c cookiefile
-v
```

The next command passes the cookie with the authentication token through the `-b` switch, and returns the user's tenant information.

```
curl -k https://10.247.100.247:4443/tenant -b cookiefile -v

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<tenant_info><name>Provider Tenant</name>
```

```

    <link href="/tenants/
urn:storageos:TenantOrg:d61d9fa1-9886-40ef-85d3-c40b6de2c72f:"
rel="self"/>
    <id>urn:storageos:TenantOrg:d61d9fa1-9886-40ef-85d3-c40b6de2c72f:</
id>
</tenant_info>

```

Authenticate without cookies

This example shows how to use authentication tokens by reading the X-SDS-AUTH-TOKEN http header from a successful authentication request and copying that header into a subsequent request. This example does not use cookies. The examples here are written in `curl` and formatted for readability.

This command executes a GET on the `/login` resource. The `-u` option indicates the user of basic authentication header. The user designation must be included in the request. Upon successful authentication, a HTTP 200 code is returned as well as the X-SDS-AUTH-TOKEN header containing the encoded token.

```

curl -L --location-trusted -k https://10.247.100.247:4443/login -u
"root:ChangeMe" -v

> GET /login HTTP/1.1
> Authorization: Basic cm9vdDpDaGFuZ2VNZQ==
> User-Agent: curl/7.24.0 (i386-pc-win32) libcurl/7.24.0 OpenSSL/
0.9.8t zlib/1.2.5
> Host: 10.247.100.247:4443
> Accept: */*
>
< HTTP/1.1 200 OK
< Date: Tue, 26 Nov 2013 22:18:25 GMT
< Content-Type: application/xml
< Content-Length: 93
< Connection: keep-alive
< X-SDS-AUTH-TOKEN:
BAAcQ0xOd3g0MjRCUG4zT3NjdnNum1AvQTFyblNrPQMAUAQADTEzODU0OTQ4NzYzNTICAA
EABQA5dXJu

OnN0b3JhZ2VvczpbUb2t1bjo2MjIxOTcyZS01NGUyLTRmNWQtYWZjOC1kMGE3ZDZmZDU3Mm
U6AgAC0A8=
<
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<loggedIn>
  <user>root</user>
</loggedIn>
* Connection #0 to host 10.247.100.247 left intact
* Closing connection #0
* SSLv3, TLS alert, Client hello (1):

```

The token can then be passed back in the next API call. You can copy the X-SDS-AUTH-TOKEN contents and pass it to the next request through `curl`'s `-H` switch.

```

curl https://10.247.100.247:4443/tenant
-k
-H "X-SDS-AUTH-TOKEN:
BAAcOHZLaGF4MTl3eFhpY0czZ0tWUGhJV2xreUE4PQMAUAQADTEzODU0OTQ4NzYzNTICAA
EABQA5dXJu

```

```
OnN0b3JhZ2VvczpwUb2t1bjpkYzc3ODU3Mi04NWRmLTQ2YjMtYjgwZi05YTdlNDFkY2QwZDg6AgAC0A8="
```

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<tenant_info>
  <name>Provider Tenant</name>\
  <link href="/tenants/
urn:storageos:TenantOrg:d61d9fa1-9886-40ef-85d3-c40b6de2c72f:"
rel="self"/>
  <id>urn:storageos:TenantOrg:d61d9fa1-9886-40ef-85d3-c40b6de2c72f:</
id>
</tenant_info>
```

How to handle an HTTP 302 authentication redirect

If you try to access a ViPR Controller REST resource without a valid token, ViPR Controller will issue an HTTP 302 redirect code.

An HTTP 302 redirect code includes the URL of the authentication resource, appended with a `service=` parameter that indicates where to be redirected after successful authentication.

Handle authentication redirects using cookies

Using cookies, you can automatically follow redirects.

The following curl example allows access to the API resource by passing credentials to it, and passing curl's `-L` option which instructs curl to automatically follow the redirects. The final HTTP response code is 200 OK, although in the full verbose output you can see how curl received the HTTP 302 Found and automatically followed the redirect.

Request

```
curl -k "<ViPR_Controller_VIP>:4443/tenant?using-cookies=true" -u
"root:ChangeMe" -c cookie1 -b cookie1 -L -v
```

Response

```
HTTP 200 OK
```

Response Body

```
GET /tenant?using-cookies=true HTTP/1.1
Authorization: Basic cm9vdDpDaGFuZ2VNZQ==
User-Agent: curl/7.24.0 (i386-pc-win32) libcurl/7.24.0 OpenSSL/0.9.8t
zlib/1.2.5
Host: 10.247.100.247:4443
Accept: */*
Cookie: X-SDS-AUTH-
TOKEN=BAAcVlM5TkkwdnRvUFBJbXJkbzVqSzB3azZBQ0VnPQMAUAQADTEzODU0OTQ4NzYz
NTICAAEABQA5dXJuOnN0b3JhZ2VvczpwUb2t1bjpkYzc3ODU3Mi04NWRmLTQ2YjMtYjgwZi05YTdlNDFkY2QwZDg6AgAC0A8=
xNwVmOTU6AgAC0A8=

HTTP/1.1 200 OK
Date: Wed, 27 Nov 2013 18:57:12 GMT
Content-Type: application/xml
Content-Length: 276
```

```

Connection: keep-alive

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<tenant_info>
<name>Provider Tenant</name>
<link href="/tenants/urn:storageos:TenantOrg:d61d9fa1-9886-40ef-85d3-
c40b6de2c72f:" rel="self"/>
<id>urn:storageos:TenantOrg:d61d9fa1-9886-40ef-85d3-c40b6de2c72f:</id>
</tenant_info>

```

Handle authentication redirects without using cookies

This example shows what to do when you authenticate with an invalid security token or no token. This example does not use cookies. The examples in this section are written in curl.

In this example, the initial request for the current user's tenant returns an HTTP 302 error. Note the following:

- The X-SDS-AUTH-TOKEN header has to be copied into each request.
- The X-SDS-AUTH-TOKEN header is a custom HTTP header.
- If you are not using cookies, HTTP clients that use the **automatically follow redirects** option need to disable it. Automatically following redirects would mean the client follows all HTTP 302 responses without copying the custom header. This results in an authentication failure.

Procedure

1. Request the current user's tenant.

Request

```
curl -k "<ViPR_Controller_VIP>:4443/tenant" -v
```

Response

```

GET /tenants HTTP/1.1
User-Agent: curl/7.24.0 (i386-pc-win32) libcurl/7.24.0 OpenSSL/
0.9.8t zlib/1.2.5
Host: 10.247.100.247:4443
Accept: */*

HTTP/1.1 302 Found
Date: Wed, 27 Nov 2013 15:30:13 GMT
Content-Length: 0
Connection: keep-alive
Location: <ViPR_Controller_VIP>/login?service={LocationString}

```

2. Perform a GET against the location in the response body.

Request

```
curl -k "<ViPR_Controller_VIP>:4443/login?service={LocationString}" -v
```

Response

```

HTTP/1.1 401 Unauthorized
WWW-Authenticate: basic realm="ViPR"

```

3. Present basic authentication credentials to the URL returned in step 1.

Request

```
curl -k "<ViPR_Controller_VIP>:4443/login?service={LocationString}"
-v -u "root:ChangeMe"
```

Response

```
GET /login?service={LocationString} HTTP/1.1
> Authorization: Basic cm9vdDpDaGFuZ2VNZQ==
> User-Agent: curl/7.24.0 (i386-pc-win32) libcurl/7.24.0 OpenSSL/
0.9.8t zlib/1.2.5
> Host: 10.247.100.247:4443
> Accept: */*
>
< HTTP/1.1 302 Found
< Date: Wed, 27 Nov 2013 16:53:28 GMT
< Content-Type: application/xml
< Content-Length: 0
< Connection: keep-alive
< Location: https://10.247.100.247:4443/tenant?auth-redirected
< X-SDS-AUTH-TOKEN: {Token_String}
```

On successful authentication, you see another 302 code, this time redirecting you to the original service. The authentication token is also in the HTTP header.

4. Access the location, making sure to supply the X-SDS-AUTH-TOKEN HTTP header.

Request

```
curl -k <ViPR_Controller_VIP>:4443/tenant?auth-redirected -H
"X-SDS-AUTH-TOKEN:{token_text}"
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<tenant_info>
<name>Provider Tenant</name>
<link href="/tenants/
urn:storageos:TenantOrg:d61d9fa1-9886-40ef-85d3-c40b6de2c72f:"
rel="self"/>
<id>urn:storageos:TenantOrg:d61d9fa1-9886-40ef-85d3-c40b6de2c72f:</
id>
</tenant_info>
```

Logout

The logout API ends a session.

A given user is allowed a maximum of 100 concurrent authentication tokens. Past this limit, the system refuses any new connection for this user until tokens free up. They can free up by expiring naturally, or by explicitly calling this URI:

```
https://<ViPR_Controller_VIP>:4443/logout
```

If you have multiple sessions running simultaneously, this URI forces the termination of all tokens related to the current user.

```
GET <ViPR_Controller_VIP>:4443/logout?force=true
```

An example logout request follows.

Request

```
GET: <ViPR_Controller_VIP>:4443/logout
X-SDS-AUTH-TOKEN: {Auth_Token}
```

Pass in the header or cookie with the authentication token to logout.

Response

```
HTTP 200
```

A ViPR Controller user can view their own name, tenant associate, and roles using the `whoami` API call.

Request

```
GET <ViPR_Controller_VIP>:4443/user/whoami
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>
```

Response

```
HTTP 200
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<user>
  <common_name>root</common_name>
  <distinguished_name>root</distinguished_name>
  <home_tenant_roles>
    <home_tenant_role>TENANT_ADMIN</home_tenant_role>
  </home_tenant_roles>
  <subtenant_roles/>
  <tenant>urn:storageos:TenantOrg:6c7dde31-
ec48-4028-8672-5e74f5754656:global</tenant>
  <vdc_roles>
    <vdc_role>SYSTEM_AUDITOR</vdc_role>
    <vdc_role>SECURITY_ADMIN</vdc_role>
    <vdc_role>SYSTEM_ADMIN</vdc_role>
    <vdc_role>SYSTEM_MONITOR</vdc_role>
  </vdc_roles>
</user>
```

This example shows the `whoami` output for the Root user. Root is associated with the tenant indicated in the `<tenant>` field. Root has the tenant roles listed in the `<home_tenant_roles>` and `<subtenant_roles>` fields. It also has the virtual data center roles listed in the `<vdc_role>` field.

CHAPTER 3

Customizing the names of resources created by ViPR Controller on physical systems

This chapter contains the following topics:

- [Customizing resource names created on physical systems](#)..... 18
- [Naming policy syntax](#)..... 19
- [Available functions](#).....20
- [Add a custom naming convention using the REST API](#)..... 21
- [Additional REST API calls to manage custom naming conventions](#)..... 25

Customizing resource names created on physical systems

As you add physical assets, ViPR Controller automatically creates a number of resources on the physical system, such as masking views and zones, using a single global hard-coded naming convention for each type of resource. You can override these default names and provide your own naming convention for several types of resources.

If you define your own naming convention for a resource, ViPR Controller uses your convention. Otherwise, the default naming convention is used to name the resource. Your custom naming convention applies to all new instances of that resource. It does not rename existing instances.

Note

ViPR Controller does not enforce uniqueness for custom names. To avoid any naming conflicts, make sure your naming conventions are unique.

You can configure custom naming conventions globally or per system type scope.

The name that is generated from your custom naming convention must adhere to the following restrictions imposed by the storage or network system to which it applies:

- The name can not exceed the maximum length for the resource.
- The name must only include characters that are part of the valid characters set for the resource.

Note

For a clone set with the BCV attribute, ViPR Controller does not support VMAX Masking for Host Masking View Name, Cluster Masking View Name, Host Storage Group Name, Cluster Storage Group Name, Host Port Group Name and Cluster Port Group Name.

You can modify the default names of these ViPR Controller resources.

- San Zoning
 - Zoning - scope can be set globally or by system type
- VMAX Masking
 - Host Masking View Name
 - Cluster Masking View Name
 - Host Storage Group Name
 - Cluster Storage Group Name
 - Host Cascaded IG Name
 - Cluster Cascaded IG Name
 - Host Cascaded SG Name
 - Cluster Cascaded SG Name
 - Host Initiator Group Name
 - Cluster Initiator Group Name
 - Host Port Group Name
 - Cluster Port Group Name
- VNX Storage Groups

- Host Storage Group Name
- VPLEX
 - Storage View Name
- XtremIO
 - Volume Folder Name
 - Initiator Group Name
 - Host Initiator Group Folder Name
 - Cluster Initiator Group Folder Name
- HDS
 - Host Storage Domain Name
 - Host Storage Domain Nick Name

Naming policy syntax

When you create a custom naming convention for a resource, it must follow a specific syntax.

For each of the resources for which you can provide a custom naming convention, there are a set of variables and [functions on page 20](#) that you can use to create the name. The functions are the same for all of the resources, but the variables will differ by resource.

A custom naming convention can include the following:

- Literal strings.
- Special characters, such as underscores (`_`), that are part of the valid character set for the resource. When using a dot (`.`) you must preface it with a backslash (`\`), such as `emc\.com`.
- Variable name surrounded by curly brackets.
- Function,
 - Applied to an individual variable string to select certain parts of the string value for the name, using the syntax `<variable_name>.<function_name>(<function values>)`
 - Applied to the entire custom naming mask, using the syntax, `(<entire_name_mask>).<function_name>(<function_values>)`

Refer to the documentation for the physical system for the list of valid characters for the literal strings and special characters for each resource.

In this example, `host_name` and `array_serial_number` are variables, and `FIRST` and `LAST` are [functions on page 20](#) that are applied to those variables to select the part of the string that is to become part of the custom name. The name of the resource will be comprised of the first 12 characters of the host name, followed by an underscore (`_`) and the last 3 characters of the array serial number, followed by `_CSG`.

```
{host_name.FIRST(12)}_{array_serial_number.LAST(3)}_CSG
```

Note

Any function being applied to a variable is within the curly brackets for that variable.

If any variables contain invalid characters for the resource name, by default ViPR Controller removes those invalid characters. For example, if the zone name variable,

{host_name}, contains ".", ViPR Controller removes them after all the string functions are applied.

Available functions

For each of the resources for which you can provide a custom naming convention, there are a set of variables and functions that you can use to create the naming policy. The functions are the same for all of the resources, but the variables will differ by resource.

The functions that are available are described in the table. The general format for using a function is: {<variable_name>.<function_name>(<function_values>)}

Note

These examples refer to the variable *host_name* which corresponds to the **Host** field when you added the host to ViPR Controller. The *host_name* variable is the host FQDN or IP address. There is also a *host_assigned_name* variable that you can use in some of your custom names that corresponds to the **Name** field when you added the host. The *host_assigned_name* variable is just a label that you can assign to the host.

Function	Description
FIRST	Use the first n characters of the string value of a variable. For example, <pre>{host_name.FIRST(60)}</pre>
LAST	Use the last n characters of the string value of a variable. For example, <pre>{array_serial_number.LAST(3)}</pre>
REPLACE	Replace a character with another character. In this example, all of the dashes in the string value of the variable, <i>hba_port_wwn</i> , are replaced with a null string, essentially deleting all of the dashes. <pre>{hba_port_wwn.REPLACE("-", "")}</pre>
SUBSTRING	Use part of a string. You specify the beginning and ending character of the string that defines the substring. In this example, only the characters 3-9 are selected from the <i>host_name</i> variable to be part of the custom name. <pre>{host_name.SUBSTRING(3,9)}</pre>
TOLOWER	Change all characters in the specified string to lower case. <pre>{host_name.TOLOWER()}</pre>
TOUPPER	Change all characters in the specified string to upper case. <pre>{host_name.TOUPPER()}</pre>

Function	Description
TRIM	<p>Remove leading and trailing characters from a string. The TRIM function can be used on the final generated name or on the individual variable strings.</p> <p>To apply the function to the entire generated name, the syntax is as shown. Note that the entire custom name mask is contained within parentheses.</p> <pre>{{cluster_name.FIRST(19)}_{host_name.FIRST(2)}}.TRIM("_")</pre> <p>To apply the function to an individual variable, the syntax is as shown.</p> <pre>{{cluster_name.FIRST(19)}}.TRIM(";")</pre>

You can also concatenate functions on an individual variable, with the functions being evaluated from left to right. In this example, the name would use the first 15 characters of the value of the variable *host_name* and change those characters to all lowercase.

```
{host_name.FIRST(15).TOLOWER() }
```

Add a custom naming convention using the REST API

System administrators can add custom naming conventions using the REST API.

Before you begin

- Authenticate with the ViPR Controller REST API as a Security Administrator.
- Review the maximum name length and the list of valid characters for the resource name on the physical system for which you are adding a custom naming convention.
- ViPR Controller does not enforce uniqueness for your custom name so ensure that your naming convention produces unique names to avoid conflicts.
- The *EMC ViPR Controller REST API Reference* provides a complete list of parameters for the REST API methods used in this discussion. You can access this document from the [ViPR Controller Product Documentation Index](#).

In this example, a custom naming convention is being added for the configuration type, *SanZoneName*, which determines how ViPR Controller names SAN zones.

Procedure

1. Determine the name of the controller configuration for which you want to provide a custom naming convention by sending `GET /config/controller/types`. This method returns a list of the names of all of the available controller configurations. If you already know the name, then you can skip to step 2.

Request

```
GET https://<ViPR_Controller_VIP>:4443/config/controller/types
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>
```

Response

Note

For readability, only part of the list of controller configuration names is shown in this response.

```

HTTP 200 OK
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<config_types>
  <config_type>
    <name>SanZoneName</name>
    <link rel="self" href="/config/controller/types/
SanZoneName"/>
  </config_type>
  <config_type>
    <name>VNXHostStorageGroupName</name>
    <link rel="self" href="/config/controller/types/
VNXHostStorageGroupName"/>
  </config_type>
  <config_type>
    <name>VMAXHostMaskingViewName</name>
    <link rel="self" href="/config/controller/types/
VMAXHostMaskingViewName"/>
  </config_type>
  <config_type>
    <name>VMAXClusterMaskingViewName</name>
    <link rel="self" href="/config/controller/types/
VMAXClusterMaskingViewName"/>
  </config_type>
  <config_type>
    <name>VMAXHostStorageGroupName</name>
    <link rel="self" href="/config/controller/types/
VMAXHostStorageGroupName"/>
  </config_type>
</config_types>

```

- Retrieve the variables that define the configuration type by sending `GET /config/controller/types/<configuration_type>`. These variables can be used, along with the [available functions on page 20](#), to create your custom naming convention.

Each configuration type includes one or more scopes that can be defined. The response includes the type of scope and the valid values for the scope. In addition, each variable that is returned includes the name of the variable, sample values, and whether the variable is recommended for inclusion in your custom naming convention to produce a unique resource name.

Request

```

GET https://<ViPR_Controller_VIP>:4443/config/controller/types/
SanZoneName
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

```

Response

```

HTTP 200 OK
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<config_type>
  <name>SanZoneName</name>
  <rules>
    <rule>IllegalCharsConstraint</rule>
    <rule>MaxLengthConstraint</rule>
    <rule>IllegalFirstCharConstraint</rule>
  </rules>

```

```

<scopes>
  <scope>
    <type>global</type>
    <value>default</value>
  </scope>
  <scope>
    <type>systemType</type>
    <value>mds</value>
    <value>brocade</value>
  </scope>
</scopes>
<type>String</type>
<variables>
  <variable>
    <recommended>true</recommended>
    <display_name>hba_port_wwn</display_name>
    <sample_value>10:00:00:90:FA:34:4C:EE</sample_value>
  </variable>
  <variable>
    <recommended>false</recommended>
    <display_name>hba_name</display_name>
    <sample_value>hba01</sample_value>
  </variable>
  <variable>
    <recommended>true</recommended>
    <display_name>array_port_wwn</display_name>
    <sample_value>50:00:09:73:00:18:95:24</sample_value>
  </variable>
  <variable>
    <recommended>false</recommended>
    <display_name>array_port_name</display_name>
    <sample_value>FA-10E:0</sample_value>
  </variable>
  <variable>
    <recommended>false</recommended>
    <display_name>host_name</display_name>
    <sample_value>xyz015.company.com</sample_value>
  </variable>
  <variable>
    <recommended>false</recommended>
    <display_name>host_type</display_name>
    <sample_value>Windows</sample_value>
  </variable>
  <variable>
    <recommended>false</recommended>
    <display_name>vsan/fabric</display_name>
    <sample_value>fid3100</sample_value>
  </variable>
  <variable>
    <recommended>false</recommended>
    <display_name>array_serial_number</display_name>
    <sample_value>000123456789</sample_value>
  </variable>
  <variable>
    <recommended>false</recommended>
    <display_name>array_type</display_name>
    <sample_value>vmax</sample_value>
  </variable>
</variables>
</config_type>

```

- Use POST `/config/controller/preview` to preview how variables will appear in your custom name to ensure that you generate a unique name with your custom naming convention.

The request payload includes your custom naming convention for the configuration type, the scope, and the variable name and value that you want to preview.

Request

```
POST https://<ViPR_Controller_VIP>:4443/config/controller/preview
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<config_preview>
  <config_type>SanZoneName</config_type>

<value>ZN_{host_name.FIRST(8)}_{hba_port_wwn}_{array_port_wwn}</
value>
  <scope>
    <type>systemType</type>
    <value>brocade</value>
  </scope>
  <preview_variable>
    <variable_name>host_name</variable_name>
    <value>abc12570.mycompany.com</value>
  </preview_variable>
</config_preview>
```

Response

```
HTTP 200 OK
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<preview>

<resolved_value>ZN_abc12570_10000090FA344CEE_5000097300189524</
resolved_value>
</preview>
```

4. Add the custom naming convention value for the configuration type using POST / config/controller.

Request

```
POST https://<ViPR_Controller_VIP>:4443/config/controller/
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<config_create>
  <config_type>SanZoneName</config_type>

<value>ZN_{host_name.FIRST(8)}_{hba_port_wwn}_{array_port_wwn}</
value>
  <scope>
    <type>systemType</type>
    <value>brocade</value>
  </scope>
</config_create>
```

Response

```
HTTP 200 OK
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<config>
  <id>urn:storageos:CustomConfig:f1c798ac-8a21-4d1c-
a069-59c34131d3b5:vdcl</id>
  <link rel="self" href="/config/controller/
urn:storageos:CustomConfig:f1c798ac-8a21-4d1c-
a069-59c34131d3b5:vdcl"/>
  <name>systemType.brocade.SanZoneName</name>
  <tags/>
  <config_type>
```



```

        <name>SanZoneName</name>
        <link rel="self" href="/config/controller/types/
SanZoneName"/>
    </config_type>
    <registered>true</registered>
    <scope>
        <type>systemType</type>
        <value>brocade</value>
    </scope>
    <system_default>false</system_default>
<value>ZN_{host_name.FIRST(8)}_{hba_port_wwn}_{array_port_wwn}</
value>
</config>

```

ViPR Controller has a default naming convention for each of the configuration types for which you can add a new naming convention. You cannot change the default naming convention for a configuration type. But once you add a new naming convention for a configuration type, then ViPR Controller uses your new convention, instead of the default convention. The `system_default` setting of `false` in the response indicates that this value is not the default ViPR naming convention.

Additional REST API calls to manage custom naming conventions

The table shows some important APIs that are used to manage custom naming conventions, in addition to the APIs already described in the example.

Table 1 REST API Calls to Manage Custom naming conventions

REST API Call	Description
GET /config/controller	Get all configuration types.
PUT /config/controller/{ID}	Modify a custom naming convention. <pre> <config_update> <value>ZN_{host_name.FIRST(8)}_{hba_port_wwn}_{array_port_wwn} value> </config_update> </pre>
POST /config/controller/{ID}/deactivate	Delete a custom naming convention.
GET /config/controller/search?parameter={search_parameter}	Search for a configuration type. You can search by a single parameter or multiple parameters. The search parameters are: <ul style="list-style-type: none"> • config_type • scope • name • system_default Sample search using config_type: <pre> GET /config/controller/search?config_type=SanZoneName </pre>

Table 1 REST API Calls to Manage Custom naming conventions (continued)

REST API Call	Description
	<p>Sample search using config_type and scope:</p> <pre>GET /config/controller/search? config_type=SanZoneName&scope=systemType.brocade</pre> <p>Sample search using name:</p> <pre>GET /config/controller/search? name=systemType.brocade.SanZoneName</pre> <p>Sample search for the default values of all configuration types:</p> <pre>GET /config/controller/search?system_default=true</pre> <p>Sample search for the default value of the specified config_type</p> <pre>GET /config/controller/search? config_type=SanZoneName&system_default=true</pre>
<p>POST /config/controller/{ID}/register</p>	<p>Make the naming convention active.</p> <hr/> <p>Note</p> <p>When you use POST /config/controller to create a new naming convention, it is automatically registered by ViPR Controller.</p>
<p>POST /config/controller/{ID}/deregister</p>	<p>Deregister the naming convention so it is not used.</p>

CHAPTER 4

Adding storage systems

This chapter contains the following topics:

- [Data Domain prerequisites and sample REST API request](#) 28
- [Elastic Cloud Storage prerequisites and sample REST API request](#)..... 28
- [Hitachi Data Systems prerequisites and sample REST API request](#)..... 29
- [IBM XIV prerequisites and sample REST API request](#)..... 29
- [EMC Isilon prerequisites and sample REST API request](#)..... 30
- [NetApp prerequisites and sample REST API request](#).....31
- [ScaleIO prerequisites and sample REST API request](#) 31
- [Third-party block storage prerequisites and sample REST API request](#).....33
- [VNX for Block and VMAX prerequisites and sample REST API request](#)33
- [VNX for File prerequisites and sample REST API request](#)..... 34
- [VNXe prerequisites and sample REST API request](#)..... 35
- [VPLEX prerequisites and sample REST API request](#).....35
- [EMC XtremIO prerequisites and sample REST API request](#)..... 36
- [Network configuration for block storage systems](#).....36
- [IP network configuration requirements for storage](#).....37
- [Important REST API calls to manage and configure storage systems and storage providers](#).....37
- [vNAS server REST API calls](#)..... 38

Data Domain prerequisites and sample REST API request

Data Domain storage is added to ViPR Controller by adding the Data Domain Management Center as a storage provider.

Prerequisites:

- Data Domain storage is added to ViPR Controller by adding the Data Domain Management Center as a storage provider.
- Interface Type - ddmc.
- IP Address of the Data Domain Management Center (DDMC).
- Port Number - The port used to SSH into the DDMC host. The default is 3009.
- user_name - user name for the DDMC. Storage system administrator privileges are required.
- password - password associated with the user_name.

A sample request:

```
POST https://<ViPR_Controller_VIP>:4443/vdc/storage-providers
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<storage_provider_create>
  <name>DD_File_West2</name>
  <ip_address>192.168.0.0</ip_address>
  <port_number>3009</port_number>
  <user_name>sysadmin</user_name>
  <password>Password1</password>
  <interface_type>ddmc</interface_type>
</storage_provider_create>
```

Elastic Cloud Storage prerequisites and sample REST API request

Elastic Cloud Storage storage is added to ViPR Controller using POST /vdc/storage-systems. The array is automatically discovered and registered.

Prerequisites:

- system_type - ecs
- ip_address - Direct the discovery request to the Elastic Cloud Storage array. There are four nodes and you can provide the IP address of any of the nodes.
- port_number - The default is 4443.
- user_name - Name of a user account on the Elastic Cloud Storage array. The user can be a local or domain user, but they must have system management privileges for the Elastic Cloud Storage system.
- password - Password of the user account on the Elastic Cloud Storage array.

Sample request:

```
POST https://<ViPR_Controller_VIP>:4443/vdc/storage-systems
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<storage_system_create>
  <name>ECS_west</name>
```

```
<system_type>ecs</system_type>
<ip_address>192.168.0.0</ip_address>
<port_number>4443</port_number>
<user_name>root</user_name>
<password>Password1</password>
</storage_system_create>
```

Hitachi Data Systems prerequisites and sample REST API request

Hitachi Data Systems storage is added to ViPR Controller by adding the Hitachi HiCommand Device Manager storage provider to the ViPR Controller Physical Assets, using POST /vdc/storage-providers.

Prerequisites:

- When the Hitachi HiCommand Device Manager storage provider is added, all the storage systems managed by the storage provider will be added to ViPR Controller. If you do not want all the storage systems on an storage provider to be managed by ViPR Controller :
 - Before adding the storage provider to ViPR Controller, configure it to only manage the storage systems that will be added to ViPR Controller. Move the storage systems that will not be managed to ViPR Controller onto a different storage provider.
 - Or, after adding the storage provider to ViPR Controller de-register or delete the storage systems that will not be used as a ViPR Controller resource.
- IP Address of the Hitachi HiCommand Device Manager.
- Port Number - The port number used to communicate with the Hitachi HiCommand Device Manager. The default is 2001 .
- interface_type - hicommand.
- use_ssl - false
- user_name - user name with the HiCommand storage provider. Storage system administrator privileges are required.
- password - password associated with the user name.

A sample REST API request:

```
POST https://<ViPR_Controller_VIP>:4443/vdc/storage-providers
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<storage_provider_create>
  <name>hdsdevicemgr</name>
  <interface_type>hicommand</interface_type>
  <ip_address>192.168.0.0</ip_address>
  <port_number>2001</port_number>
  <user_name>user1</user_name>
  <password>Password1</password>
  <use_ssl>false</use_ssl>
</storage_provider_create>
```

IBM XIV prerequisites and sample REST API request

IBM XIV storage is added to ViPR Controller using POST /vdc/storage-providers.

Prerequisites:

- `interface_type` - `ibmxiv`
- `ip_address` - IP address of the host.
- `port_number` - The port used to SSH into the host. The default is 5989.
- `use_ssl` - `true`.
- `user_name` - Name of the user used to SSH into the host. Storage system administrator privileges are required.
- `password` - Password associated with the user name.

A sample REST API request:

```
POST https://<ViPR_Controller_VIP>:4443/vdc/storage-providers
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<storage_provider_create>
  <name>ibmxiv_block_1</name>
  <interface_type>ibmxiv</interface_type>
  <ip_address>192.168.0.0</ip_address>
  <port_number>5989</port_number>
  <use_ssl>true</use_ssl>
  <user_name>admin</user_name>
  <password>Password1</password>
</storage_provider_create>
```

EMC Isilon prerequisites and sample REST API request

When you add an Isilon to ViPR Controller, the array is automatically added to the ViPR Controller discovery queue. If the credentials are correct, the array is automatically discovered and registered. Storage system discovery adds the storage pools and storage ports to ViPR Controller. Storage systems that are registered are available to be used and managed by ViPR Controller.

Prerequisites:

- `system_type` - `isilon`.
- `ip_address` - The SmartConnect IP address. Directs the discovery request to the Isilon SmartConnect host IP.
- `port_number` - 8080.
- User credentials with Isilon storage system administrator privileges. The Isilon user is independent of the currently authenticated ViPR Controller user. All ViPR Controller operations, which you perform on an Isilon storage system, are executed as this Isilon user.

A sample request:

```
POST https://<ViPR_Controller_VIP>:4443/vdc/storage-systems
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<storage_system_create>
  <name>isilon1</name>
  <system_type>isilon</system_type>
  <ip_address>192.168.0.0</ip_address>
  <port_number>8080</port_number>
  <user_name>rootid</user_name>
  <password>password</password>
</storage_system_create>
```

NetApp prerequisites and sample REST API request

NetApp storage is added to ViPR Controller using `POST /vdc/storage-systems`. The array is automatically discovered and registered.

Prerequisites:

- `system_type` - netapp
- `ip_address` - Direct the discovery request to the NetApp OnTap IP.
- `port_number` - The default is 443.
- `user_name` - Name of the root account on the NetApp array.
- `password` - Password of the root account on the NetApp array.

Sample request:

```
POST https://<ViPR_Controller_VIP>:4443/vdc/storage-systems
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<storage_system_create>
  <name>NetApp4</name>
  <system_type>netapp</system_type>
  <ip_address>192.168.0.0</ip_address>
  <port_number>443</port_number>
  <user_name>root</user_name>
  <password>dangerous1</password>
</storage_system_create>
```

ScaleIO prerequisites and sample REST API request

ScaleIO is added to ViPR Controller as a storage provider, using `POST /vdc/storage-providers`.

ViPR Controller creates storage systems from discovered protection domains in the ScaleIO. ViPR Controller also automatically creates storage ports, hosts, host initiators, and a network for the ScaleIO. These can not be edited. For additional information on how ViPR Controller uses the ScaleIO components to create the required ViPR Controller objects see [How ScaleIO components are integrated into ViPR Controller on page 32](#).

Prerequisites:

- `interface_type` - scaleioapi.
- `ip_address` - IP address or FQDN of the ScaleIO Gateway host.
- `use_ssl` - SSL enabled (`true`) or disabled (`false`).
- `port_number` - The port used to communicate with the ScaleIO REST API service.
 - With SSL enabled, the default is 443.
 - With SSL disabled, the default is 80.
- `user_name` - name of the user that can access the Primary MDM.
- `password` - password of the user that can access the Primary MDM.

Sample request:

```
POST https://<ViPR_Controller_VIP>:4443/vdc/storage-providers
Content-Type: application/xml
```

```
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<storage_provider_create>
  <name>scaleio_block_1</name>
  <interface_type>scaleioapi</interface_type>
  <ip_address>lg45.mycomp.com</ip_address>
  <use_ssl>true</use_ssl>
  <port_number>443</port_number>
  <user_name>admin</user_name>
  <password>Password2</password>
</storage_provider_create>
```

Best practices for ScaleIO to work with discoverable hosts

When you add ScaleIO storage, ViPR Controller automatically creates a host of type `Other` for each of the SDCs, and creates the host initiators. But these hosts are not discoverable. There are two methods that allow you to associate these ViPR Controller-created hosts with their discoverable hosts.

First method

1. Add the discoverable host, such as Windows or Linux, to ViPR Controller.
2. Add the ScaleIO storage provider to ViPR Controller.

The ScaleIO discovery adds the ScaleIO initiators to the correct discoverable host.

Second method

1. Add the ScaleIO storage provider to ViPR Controller. This creates hosts of type `Other` for each of the SDCs.
2. Edit the `Other` host. Change its type to match its operating system, such as Windows, and provide login credentials, and enable **Discoverable**.

When the host is discovered it is now associated with the ScaleIO initiators that were created when the ScaleIO storage provider was added to ViPR Controller.

How ScaleIO components are integrated into ViPR Controller

To ViPR Controller, ScaleIO block storage is a set of storage devices. ScaleIO storage is added as a storage provider of the type, `scaleioapi`.

Table 2 Mapping of ScaleIO components to ViPR Controller components

ScaleIO Component	ViPR Controller Component
Protection Domain	Storage System
Storage Pool	Storage Pool
SDS	Storage Port
SDC	Host

When ViPR Controller discovers the ScaleIO storage:

1. The protection domains are discovered and one storage system is created in ViPR Controller for each ScaleIO protection domain.
2. Storage ports are created in ViPR Controller: one storage port in a ViPR Controller storage system for each discovered SDS that is part of the ScaleIO protection domain that is mapped to the storage system. The name of the storage port maps to the name of the SDS ID.

3. ViPR Controller automatically creates a network for the ScaleIO using the SDCs, and the storage ports that were created from all of the discovered SDSs.
4. ViPR Controller automatically creates hosts and host initiators: one host for each SDC.

Once ViPR Controller has created the storage ports, network, and hosts, then a system administrator can:

1. Create a virtual array for the ScaleIO storage.
2. Create a block virtual pool and associate it with the ScaleIO virtual array.

Once the virtual arrays and block virtual pools are created, you can provision the ScaleIO block storage.

Third-party block storage prerequisites and sample REST API request

Third-party block storage is added to ViPR Controller using `POST /vdc/storage-providers`.

Prerequisites:

- `interface_type` - `cinder`.
- `ip_address` - The IP address of the Third-party block storage provider.
- `user_name` - The username for connecting to the storage provider. System administrator privileges are required.
- `password` - Password of the account provided with `user_name`.
- `port_number` - The port number used to connect to the storage provider. The default port number for a Third-party block storage provider is 22.

Sample request:

```
POST https://<ViPR_Controller_VIP>:4443/vdc/storage-providers
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<storage_provider_create>
  <name>TP_east_1</name>
  <interface_type>cinder</interface_type>
  <ip_address>192.168.0.0</ip_address>
  <port_number>22</port_number>
  <user_name>admin</user_name>
  <password>Password1</password>
  <use_ssl>>false</use_ssl>
</storage_provider_create>
```

VNX for Block and VMAX prerequisites and sample REST API request

For VNX for Block and VMAX, adding an SMI-S provider provides access to the storage systems. Once the SMI-S providers are added to {Conref} ViPR , then the underlying storage systems, storage pools, and storage ports are discovered automatically without any further user input.

Prerequisites:

- `ip_address` - IP address of the SMI-S provider.
- `port_number` - The port number used to connect to the SMI-S provider. The default is 5989.
- `user_name` - The username used for connecting to the SMI-S provider.
- `password` - The password for the username used for connecting to the SMI-S provider.

Sample request:

```
POST https://<ViPR_Controller_VIP>:4443/vdc/storage-providers
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<storage_provider_create>
  <name>vnx_block_west3</name>
  <interface_type>smis</interface_type>
  <ip_address>192.168.0.0</ip_address>
  <port_number>5989</port_number>
  <user_name>admin1</user_name>
  <password>Password1</password>
  <use_ssl>true</use_ssl>
</storage_provider_create>
```

VNX for File prerequisites and sample REST API request

Prerequisites:

- For the VNX File Control Station:
 - The IP address of the Control Station.
 - The port number used to connect to the Control Station. The default is 443.
 - VNX File user credentials with storage system administrator privileges.
- For the Onboard Storage Provider:
 - The storage provider host.
 - The port number used to connect to the VNX File Onboard Storage Provider. The default is 5988.
 - User credentials to access the Onboard Storage Provider.
 - Whether SSL is used to communicate with the Onboard Storage Provider.
- The system type is `vnxfile`.

Sample request:

```
POST https://<ViPR_Controller_VIP>:4443/vdc/storage-systems
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<storage_system_create>
  <name>file_west_1</name>
  <system_type>vnxfile</system_type>
  <ip_address>192.168.0.0</ip_address>
  <port_number>443</port_number>
  <user_name>administrator</user_name>
  <password>password</password>
  <smis_provider_ip>192.160.100.0</smis_provider_ip>
  <smis_port_number>5988</smis_port_number>
  <smis_user_name>sp_admin</smis_user_name>
  <smis_password>Password1</smis_password>
```

```
<smis_use_ssl>>false</smis_use_ssl>
</storage_system_create>
```

VNXe prerequisites and sample REST API request

VNXe storage is added to ViPR Controller as a storage system., using `POST /vdc/storage-systems`. Once you add the storage system to ViPR Controller, then the underlying storage pools and storage ports are discovered automatically without any further user input.

Prerequisites:

- `system_type` - vnx.
- `ip_address` - The IP address of the host.
- `port_number` - The port used to SSH into the host. The default is 443.
- `user_name` - The name of the user used to SSH into the host. Storage system administrator privileges are required.
- `password` - Password associated with the `user_name`.

Sample request:

```
https://<ViPR_Controller_VIP>:4443/vdc/storage-systems
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<storage_system_create>
  <name>vnx_west_1</name>
  <system_type>vnx</system_type>
  <ip_address>192.168.0.0</ip_address>
  <port_number>443</port_number>
  <user_name>admin</user_name>
  <password>Password1</password>
</storage_system_create>
```

VPLEX prerequisites and sample REST API request

EMC VPLEX storage systems are added ViPR Controller as a storage provider, using `POST /vdc/storage-providers`.

Prerequisites:

- The IP address of the VPLEX management server.
- The username and password for connecting to the storage system. System administrator privileges are required.
- The port number used to connect to the storage system. The default port number for a VPLEX management server is 443.

A sample REST API request:

```
POST https://<ViPR_Controller_VIP>:4443/vdc/storage-providers
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<storage_provider_create>
  <name>VPLEX_West2</name>
  <interface_type>vplex</interface_type>
  <ip_address>192.168.0.0</ip_address>
```

```
<port_number>443</port_number>
<user_name>admin</user_name>
<password>Password1</password>
<use_ssl>true</use_ssl>
</storage_provider_create>
```

EMC XtremIO prerequisites and sample REST API request

Add the XtremIO storage by sending a POST `/vdc/storage-providers` request. The clusters managed by the XtremIO Management Server (XMS) are discovered as individual XtremIO storage systems. Storage system discovery adds the storage pools and storage ports to ViPR Controller.

Prerequisites:

- `interface_type` - xtremio
- `ip_address` - Direct the discovery request to the IP address of the XtremIO Management Server.
- `use_ssl` - Specify whether to use SSL as the authentication method.
- `port_number` - The port used to connect to the XtremIO Management Server host. The default is 443.
- `user_name` - Name of the root account on the XtremIO Management Server.
- `password` - Password of the root account on the XtremIO Management Server.

Sample request:

```
POST https://<ViPR_Controller_VIP>:4443/vdc/storage-providers
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<storage_provider_create>
  <name>xtremio_west1</name>
  <interface_type>xtremio</interface_type>
  <ip_address>192.168.0.0</ip_address>
  <port_number>443</port_number>
  <use_ssl>true</use_ssl>
  <user_name>admin1</user_name>
  <password>Password1</password>
</storage_provider_create>
```

Network configuration for block storage systems

After the storage system is added to ViPR Controller, configure the SAN switches, using POST `/vdc/network-systems`, before adding the storage systems to a virtual array.

- When a SAN switch is added to ViPR Controller, the Fibre Channel networks (Brocade Fabrics or Cisco VSANs), are automatically discovered and registered in ViPR Controller. Additionally, through discovery of the SAN switch topology, ViPR Controller discovers, and registers the host initiators for hosts on the network, and identifies which storage systems are associated with the SAN switch.

See [Adding network systems \(fabric managers\) and SAN networks on page 45](#) for more information.

- For Storage Systems that use ViPR Controller services with the iSCSI protocol, the iSCSI host ports must be logged into the correct target array ports before they can be used in the service.

IP network configuration requirements for storage

After the storage system is added to ViPR Controller, create the IP networks to include in the virtual array, and be sure to add the necessary storage and host ports to use to provision the storage to the hosts.

ViPR Controller can discover the ports of IP connected storage systems and hosts, but it cannot discover the paths between them, so it is necessary to create IP networks, and then add the host, and storage system ports, which will be provisioned together, to the same IP network. See [Creating and configuring a virtual array on page 99](#)

Important REST API calls to manage and configure storage systems and storage providers

The table shows some important APIs that are used to manage and configure storage systems and storage providers.

Table 3 REST API Calls to Manage and Configure Storage Systems and Storage Providers

ViPR REST API Call	Description
POST /vdc/storage-pools/{id}/deregister	By default, all discovered storage pools are available to use for provisioning in ViPR Controller. Deregister storage pools to make them unavailable to ViPR Controller for provisioning. Sets the registration_status attribute of the storage pool to UNREGISTERED.
POST /vdc/storage-systems/{id}/storage-pools/{poolId}/register	Manually register the discovered storage pool on the storage system.
PUT /vdc/storage-systems/{id}	By default, storage systems are configured with an unlimited amount of resources that can be used by ViPR. This API call allows you to set resource limits, which defines the amount of storage that is available for use by ViPR Controller. In addition, you can also update the storage system credentials.
POST /vdc/storage-ports/{id}/deregister	By default, all storage ports are available to use for provisioning in ViPR Controller when the storage system is discovered by ViPR Controller. Deregister storage ports to make them unavailable to ViPR Controller for provisioning.
POST /vdc/storage-systems/{id}/storage-ports/{portId}/register	Manually register the discovered storage port on the storage system.
POST /vdc/storage-systems/{id}/deactivate	Remove a storage system. This method removes the storage system from ViPR Controller control and removes all resources associated with the storage system from the database. Resources (pools, ports, volumes, etc.) are not removed from the storage system physically, but become unavailable for use by ViPR Controller.
GET /vdc/storage-systems	List all storage systems.
GET /vdc/storage-systems/{id}/storage-pools	List all storage pools for the storage system.

Table 3 REST API Calls to Manage and Configure Storage Systems and Storage Providers (continued)

ViPR REST API Call	Description
GET /vdc/storage-systems/{id}/storage-ports	List all storage ports for the storage system.
POST /vdc/storage-systems/{id}/discover	Rediscover a storage system on demand.
POST /vdc/storage-providers/scan	Scan and rediscover all storage systems on demand. Note Rediscovery of a standalone ScaleIO storage provider discovers new SDC hosts or removes deleted SDC hosts, but it does not update the ScaleIO initiators. To rediscover the ScaleIO initiators, rediscover the individual storage systems associated with the ScaleIO storage provider. See POST /vdc/storage-systems/{id}/discover.
GET /vdc/storage-providers/{id}	Show the details of a storage provider.
GET /vdc/storage-providers/{id}/storage-systems	Lists the ID, name, and link for all storage systems visible to the storage provider.

vNAS server REST API calls

When you add a VNX File storage system, ViPR Controller discovers and registers its vNAS servers and attributes, such as logical interfaces and base directory. Review this section to understand the REST APIs that retrieve information about the vNAS servers.

For information about the VNX File storage system API, see [VNX for File prerequisites and sample REST API request on page 34](#).

The following table describes the VDC REST API calls specific to vNAS servers. You can also use the POST /vdc/storage-systems/discover and POST /vdc/storage-systems/{storage_system_id}/discover VDC REST API calls to discover existing and new vNAS servers in a storage system as well as update existing vNAS servers with any changes.

Table 4 vNAS server REST API calls

ViPR Controller vNAS server REST API Call	Description
GET /vdc/vnas-servers Response payload: <pre><vnas_servers> <vnas_server> <id>vnas_server_id</id> <name>vnas_server_name</name></pre>	Retrieves a list of vNAS servers.

Table 4 vNAS server REST API calls (continued)

ViPR Controller vNAS server REST API Call	Description
<pre> </vnas_server> <vnas_server> . . . </vnas_server> </vnas_servers> </pre>	
<p>GET /vdc/vnas-servers/{vnas_server_id}</p> <p>Response payload:</p> <pre> <vnas_server> <id>vnas_server_id</id> <name>vnas_server_name</name> <network_interface>network interface</ network_interface> <auth_provider>network interface</ auth_provider> </vnas_server> </pre>	Gets the details of a vNAS server.
<p>GET on /vdc/varrays/{varray_id}/vnas-servers</p> <p>Response payload:</p> <pre> <vnas_servers> <vnas_server> <id>vnas_server_id</id> <name>vnas_server_name</name> </vnas_server> <vnas_server> . . . </vnas_server> </vnas_servers> </pre>	Retrieves a list of vNAS servers on a virtual array.
<p>PUT /projects/{project_id}/assign-vnas-servers</p> <p>Response payload:</p> <pre> <vnas_servers> <vnas_server> { vnas_server_uri1 } </vnas_server> <vnas_server> { vnas_server_uri2 } </vnas_server> . . . </vnas_servers> </pre>	Assigns vNAS servers to projects.
<p>PUT /projects/{project_id}/unassign-vnas-servers</p> <p>Response payload:</p> <pre> <vnas_servers> <vnas_server> { vnas_server_uri1 } </vnas_server> <vnas_server> { vnas_server_uri2 } </vnas_server> . . . </vnas_servers> </pre>	Unassigns vNAS servers from projects.

Table 4 vNAS server REST API calls (continued)

CHAPTER 5

Adding a data protection system

This chapter contains the following topics:

- [Adding a RecoverPoint data protection system](#)..... 42

Adding a RecoverPoint data protection system

You add a RecoverPoint data protection system to ViPR Controller using `POST /vdc/protection-systems`.

Before you begin

- Authenticate with the ViPR REST API as a System Administrator
- For pre-configuration and support requirements, see *ViPR Controller Virtual Data Center Requirements and Information Guide* on the [ViPR Controller Product Documentation Index](#).
- You need the port used to communicate with EMC RecoverPoint. The default port is 7225.
- The *EMC ViPR Controller REST API Reference* on the [ViPR Controller Product Documentation Index](#) provides a description and complete list of parameters for the REST API methods used in this article.

ViPR Controller also supports data protection using Symmetrix Remote Data Facility (SRDF). See *ViPR Controller Integration with VMAX and VNX Storage Systems Guide* on the [ViPR Controller Product Documentation Index](#).

After you add a data protection system to ViPR Controller, you can select data protection when you create your block virtual pools. Any block volumes created from those block virtual pools will have their data protected by the selected data protection system.

Procedure

1. Add the RecoverPoint protection system by sending a `POST <ViPR_Controller_VIP>:4443/vdc/protection-systems` request with a `system_type` of `rp`.

The request returns a task whose URI can be queried to determine the status of the task.

Request

```
POST https://<ViPR_Controller_VIP>:4443/vdc/protection-systems
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<protection_system_create>
  <name>rp_west1</name>
  <system_type>rp</system_type>
  <ip_address>192.168.0.0</ip_address>
  <port_number>7225</port_number>
  <user_name>admin</user_name>
  <password>admin</password>
</protection_system_create>
```

Response

```
HTTP 202 Accepted
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<task>
  <creation_time>1423583101338</creation_time>
  <global>>false</global>
  <id>urn:storageos:Task:
5f7d64c9-28e8-4fef-87f3-15eb173bb5ad:vdc1</id>
  <inactive>>false</inactive>
```

```

    <internal>false</internal>
    <link rel="self" href="/vdc/tasks/urn:storageos:Task:
5f7d64c9-28e8-4fef-87f3-15eb173bb5ad:vdc1"/>
    <name>DISCOVER STORAGESYSTEM</name>
    <remote>false</remote>
    <tags/>
    <vdc>
      <id>urn:storageos:VirtualDataCenter:99c5c185-
ebb0-468a-8fe7-3ea6e65b7b57:vdc1</id>
      <link rel="self" href="/vdc/
urn:storageos:VirtualDataCenter:99c5c185-
ebb0-468a-8fe7-3ea6e65b7b57:vdc1"/>
    </vdc>
    <associated_resources/>
    <description>discover a storage systems</description>
    <op_id>e95b27ef-ec82-4002-aae0-af046820af54</op_id>
    <progress>0</progress>
    <resource>
      <id>urn:storageos:ProtectionSystem:
11865a28-5953-4d7a-84fa-2cceb1038d97:vdc1</id>
      <link rel="self" href="/vdc/protection-systems/
urn:storageos:ProtectionSystem:
11865a28-5953-4d7a-84fa-2cceb1038d97:vdc1"/>
      <name>rp_west1</name>
    </resource>
    <start_time>1423583101336</start_time>
    <state>pending</state>
</task>

```

- Repeat the query of the add protection system task, using the task URI from the response body of the POST request, until the message attribute of the task is Operation completed successfully.

Request

```

GET https://<ViPR_Controller_VIP>:4443/vdc/tasks/
urn:storageos:Task:5f7d64c9-28e8-4fef-87f3-15eb173bb5ad:vdc1
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

```

Response

```

HTTP 200 OK
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<task>
  <creation_time>1423583101338</creation_time>
  <global>false</global>
  <id>urn:storageos:Task:
5f7d64c9-28e8-4fef-87f3-15eb173bb5ad:vdc1</id>
  <inactive>false</inactive>
  <internal>false</internal>
  <link rel="self" href="/vdc/tasks/urn:storageos:Task:
5f7d64c9-28e8-4fef-87f3-15eb173bb5ad:vdc1"/>
  <name>DISCOVER STORAGESYSTEM</name>
  <remote>false</remote>
  <tags/>
  <vdc>
    <id>urn:storageos:VirtualDataCenter:99c5c185-
ebb0-468a-8fe7-3ea6e65b7b57:vdc1</id>
    <link rel="self" href="/vdc/
urn:storageos:VirtualDataCenter:99c5c185-
ebb0-468a-8fe7-3ea6e65b7b57:vdc1"/>
  </vdc>
  <associated_resources/>

```

```
<description>discover a storage systems</description>
<end_time>1423583873389</end_time>
<message>Operation completed successfully</message>
<op_id>e95b27ef-ec82-4002-aae0-af046820af54</op_id>
<progress>100</progress>
<resource>
  <id>urn:storageos:ProtectionSystem:
11865a28-5953-4d7a-84fa-2cceb1038d97:vdc1</id>
  <link rel="self" href="/vdc/protection-systems/
urn:storageos:ProtectionSystem:
11865a28-5953-4d7a-84fa-2cceb1038d97:vdc1"/>
  <name>rp_west1</name>
</resource>
<start_time>1423583101336</start_time>
<state>ready</state>
</task>
```

CHAPTER 6

Adding network systems (fabric managers) and SAN networks

This chapter contains the following topics:

- [Overview](#)..... 46
- [Add a network system \(fabric manager\)](#) 46
- [Manually creating and updating networks in ViPR Controller](#).....48
- [Important REST API calls for managing network systems \(fabric managers\)](#).....50

Overview

This chapter describes the ViPR Controller support requirements, and prerequisite information to prepare SAN (Brocade and Cisco) switches for ViPR Controller integration using the ViPR Controller REST API. It also includes the steps to add and configure the Fibre Channel (FC) networks in ViPR Controller.

You can create a network system for the following SAN switches:

- Cisco = MDS switch
- Brocade = CMCNE

When you add a SAN switch to ViPR Controller, ViPR Controller discovers and registers the Fibre Channel networks (Brocade Fabrics or Cisco VSANs). Through discovery of the SAN topology, ViPR Controller can identify which hosts and storage systems that were added to ViPR Controller, are connected through the same fibre channel network. This allows ViPR Controller to automatically build the connectivity between the hosts and storage systems when you provision a service, such as creating a block volume for a host.

After the network system is added and discovered, it is re-discovered periodically. ViPR Controller continues to update its networks as host and storage system ports are added and removed. Once a network system is discovered, ViPR Controller creates and removes SAN zones when block volumes are exported or un-exported. If that is not the intended behavior, then set the auto-zoning flag false in the virtual array associated with this network.

Add a network system (fabric manager)

Before you begin

Authenticate with the REST API as a system administrator. See [Authenticate with the REST API](#).

Depending on the type of switch you are adding, you will need the following information:

Type	Required Information
Brocade	<ul style="list-style-type: none"> • Pre-configuration and support requirements are described in the <i>ViPR Controller Virtual Data Center Requirements and Information Guide</i> on the ViPR Controller Product Documentation Index . • SMI-S Host address • Enable/disable SSL • SMI-S Port • Credentials for an account that has administrator privileges to the SMI-S host
Cisco	<ul style="list-style-type: none"> • Pre-configuration and support requirements are described in the <i>ViPR Controller Virtual Data Center Requirements and Information Guide</i> on the ViPR Controller Product Documentation Index • Host address for the switch • Port • Credentials for an account that has administrator privileges to the switch

The *EMC ViPR Controller REST API Reference* on the [ViPR Controller Product Documentation Index](#) provides a description and complete list of parameters for the REST API methods used in this example.

Procedure

1. Add a network system (fabric manager) using POST `/vdc/network-systems`.

The request returns a task.

In this example, a Brocade switch is being added to ViPR.

Request

```
POST https://<ViPR_Controller_VIP>:4443/vdc/network-systems

Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<network_system_create>
  <name>brocade_west1</name>
  <system_type>brocade</system_type>
  <smis_provider_ip>192.168.0.0</smis_provider_ip>
  <smis_port_number>5989</smis_port_number>
  <smis_user_name>administrator</smis_user_name>
  <smis_password>password</smis_password>
  <smis_use_ssl>>false</smis_use_ssl>
</network_system_create>
```

Response

```
HTTP 202 Accepted
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<task>
  <creation_time>1423588035002</creation_time>
  <global>>false</global>
  <id>urn:storageos:Task:
56a98fa7-10b9-48f0-8075-94e66a25e4ce:vdc1</id>
  <inactive>>false</inactive>
  <internal>>false</internal>
  <link rel="self" href="/vdc/tasks/urn:storageos:Task:
56a98fa7-10b9-48f0-8075-94e66a25e4ce:vdc1"/>
  <name>DISCOVER NETWORKSYSTEM</name>
  <remote>>false</remote>
  <tags/>
  <vdc>
    <id>urn:storageos:VirtualDataCenter:99c5c185-
ebb0-468a-8fe7-3ea6e65b7b57:vdc1</id>
    <link rel="self" href="/vdc/
urn:storageos:VirtualDataCenter:99c5c185-
ebb0-468a-8fe7-3ea6e65b7b57:vdc1"/>
  </vdc>
  <associated_resources/>
  <description>discover one network system</description>
  <op_id>b1d7e95c-730c-4658-971f-49e1893ddf41</op_id>
  <progress>0</progress>
  <resource>
    <id>urn:storageos:NetworkSystem:
7875ed0e-1d2a-430d-94d0-20fcd7e38bb4:vdc1</id>
    <link rel="self" href="/vdc/network-systems/
urn:storageos:NetworkSystem:
7875ed0e-1d2a-430d-94d0-20fcd7e38bb4:vdc1"/>
    <name>brocade_west1</name>
  </resource>
  <start_time>1423588035000</start_time>
```

```
<state>pending</state>
</task>
```

ViPR Controller discovers, and registers the fabric manager and the associated networks.

2. Query the discover network system task URI until the message attribute of the task is Operation completed successfully.

Request

```
GET https://<ViPR_Controller_VIP>:4443/vdc/tasks/
urn:storageos:Task:56a98fa7-10b9-48f0-8075-94e66a25e4ce:vdc1
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>
```

Response

```
HTTP 200 OK
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<task>
  <creation_time>1423588035002</creation_time>
  <global>false</global>
  <id>urn:storageos:Task:
56a98fa7-10b9-48f0-8075-94e66a25e4ce:vdc1</id>
  <inactive>false</inactive>
  <internal>false</internal>
  <link rel="self" href="/vdc/tasks/urn:storageos:Task:
56a98fa7-10b9-48f0-8075-94e66a25e4ce:vdc1"/>
  <name>DISCOVER NETWORKSYSTEM</name>
  <remote>false</remote>
  <tags/>
  <vdc>
    <id>urn:storageos:VirtualDataCenter:99c5c185-
ebb0-468a-8fe7-3ea6e65b7b57:vdc1</id>
    <link rel="self" href="/vdc/
urn:storageos:VirtualDataCenter:99c5c185-
ebb0-468a-8fe7-3ea6e65b7b57:vdc1"/>
  </vdc>
  <associated_resources/>
  <description>discover one network system</description>
  <end_time>1423588035135</end_time>
  <message>Operation completed successfully</message>
  <op_id>b1d7e95c-730c-4658-971f-49e1893ddf41</op_id>
  <progress>100</progress>
  <resource>
    <id>urn:storageos:NetworkSystem:
7875ed0e-1d2a-430d-94d0-20fcd7e38bb4:vdc1</id>
    <link rel="self" href="/vdc/network-systems/
urn:storageos:NetworkSystem:
7875ed0e-1d2a-430d-94d0-20fcd7e38bb4:vdc1"/>
    <name>brocade_west1</name>
  </resource>
  <start_time>1423588035000</start_time>
  <state>ready</state>
</task>
```

Manually creating and updating networks in ViPR Controller

Manually add storage ports, and host ports to the Fibre Channel network (Brocade Fabrics and Cisco VSANs) to define the connectivity ViPR Controller will use when storage is provisioned to the hosts. When ports are not assigned to the Fibre Channel network, ViPR

Controller will automatically assign the ports when the storage is provisioned to the hosts.

- Only system administrators can assign storage and host ports to Fibre Channel networks.
- Fibre Channel networks are automatically added, discovered, and registered in ViPR Controller when the associated Brocade or Cisco switch is added to ViPR Controller using the `POST /vdc/network-systems` REST API call.
- Storage ports, host ports, or both can be defined for the Fibre Channel network by adding or removing network endpoints, using `PUT /vdc/networks/{Network_URN}`.

The table list some of the important REST API calls for managing the storage and host ports, and the Fibre Channel network.

Table 5 Important REST API calls for managing networks

ViPR REST API Call	Description
<code>POST /vdc/networks/{id}/deregister</code>	Deregister a network to make it unavailable to ViPR Controller for provisioning. Sets the <code>registration_status</code> attribute of the network to UNREGISTERED.
<code>GET /vdc/networks/{id}/storage-ports</code>	List all storage ports associated with the network end points.
<code>PUT /vdc/networks/{id}</code>	Update a network's name, endpoints or varrays. When endpoints are changed, added or removed, and the endpoints match some storage ports, the storage ports associations to the network are updated accordingly. If the endpoints added exist in another network, they are first removed from their current network. Discovered endpoints cannot be removed from their current networks or added to another one. When the storage ports networks are changed, their corresponding storage pools are also update to reflect any change in varray connectivity that may have resulted from the change.
<code>GET /vdc/networks/{id}/initiators</code>	List all initiators associated with the network end points.
<code>GET /vdc/networks/{id}/ip-interfaces</code>	List all IP interfaces associated with the network end points.
<code>POST /vdc/networks</code>	Create a network of type FC, IP or Ethernet. The network can optionally be added to virtual arrays and populated with endpoints. When the network has endpoints and the endpoints are matched to storage ports, the storage ports become assigned to the network. When the network is also added to virtual arrays, the storage ports' array pools are updated to show they are connected to the networks' virtual arrays.

Important REST API calls for managing network systems (fabric managers)

The table lists some of the important REST API calls for managing network systems (fabric managers).

Table 6 Important ViPR Controller REST API calls for managing network systems

ViPR REST API Call	Description
POST /vdc/network-systems/{id}/deregister	Deregister a network system (fabric manager) to make it unavailable to ViPR Controller for provisioning. Sets the registration_status attribute of the network-systems to UNREGISTERED.
PUT /vdc/network-systems/{id}	Change the IP address, port, credentials, or name of a network system. A discovery is asynchronously initiated as a result of this call. <pre><network_system_update> <name>mds-a</name> <ip_address>192.168.0.2</ip_address> <port_number>22</port_number> <user_name>no-user</user_name> <password>nopass</password> </network_system_update></pre>
GET /vdc/network-systems/{id}/fc-endpoints	List the cached fiber channel connectivity information between the network system and external systems, such as host initiators or storage array ports. The connectivity information is periodically updated, or can be refreshed on demand using a POST /vdc/network-systems/{id}/refresh.
GET /vdc/network-systems/{id}/san-fabrics	List all VSAN or fabric names configured on this network system.
GET /vdc/network-systems/{id}/san-fabrics/{fabricId}	List all of the active zones (and their zone members) for the specified fabric or VSAN in a network system.
POST /vdc/network-systems	Create a new network system. This can either represent an SSH connection to a Cisco MDS or Nexus switch, or an SMI-S connection to the Brocade Network Advisor. The call will return before communication has been established, but discovery of the device will be initiated.
GET /vdc/network-systems/{id}/san-aliases	Returns a list of aliases for the specified network device. For Brocade, aliases are retrieved per fabric. For MDS, the full list of aliases for the network system is returned. This is a synchronous call to the device.
POST /vdc/network-systems/{id}/san-aliases	Adds one or more aliases to the specified network system.
POST /vdc/network-systems/{id}/san-aliases/remove	Removes one or more aliases from the specified network system.

Table 6 Important ViPR Controller REST API calls for managing network systems (continued)

ViPR REST API Call	Description
PUT /vdc/network-systems/{id}/san-aliases	Changes the WWN member of or more aliases of the specified network system. This is an asynchronous call.

Adding network systems (fabric managers) and SAN networks

CHAPTER 7

Adding hosts and clusters

This chapter contains the following topics:

- [Overview](#)..... 54
- [Add and discover a LINUX host](#)..... 54
- [Add and discover a Windows host](#)..... 57
- [Add an AIX host or AIX VIO server to ViPR Controller](#).....59
- [Add a host other than Windows, AIX or Linux](#).....62
- [Add host ports to an IP network](#).....65
- [Add a host to a cluster](#).....65
- [Replace Host Initiators after a storage volume has been exported to a host](#)..... 67
- [Important REST API calls to manage hosts and clusters](#)..... 79

Overview

You can add Linux, Windows, AIX, AIX VIO, HP-UX, and other types of hosts using the ViPR Controller REST API. You can also add host initiators and add the host to a ViPR Controller cluster.

Hosts are computers that use software including Windows, Linux, and VMware for their operating system. In ViPR Controller, hosts are tenant resources like volumes and file systems. Unlike those resources, however, hosts are imported and discovered rather than provisioned by ViPR Controller.

Hosts must be imported into ViPR Controller by a Tenant Administrator before storage may be exported and attached to them. By default, hosts are not assigned to a project which means only the Tenant Administrator may see them and export/attach storage to them. If further delegation is required, the Tenant Administrator may assign a host to a project. Anyone who has privileges to manage resources in that project may then see and export/attach storage to that host.

Hosts are not explicitly associated with virtual arrays. The host-to-virtual array association is implied based on network connectivity.

Add and discover a LINUX host

To manage a LINUX host in ViPR Controller, use `POST /compute/hosts` to add the host to ViPR Controller and discover the host, and its initiator ports.

Before you begin

- Preconfiguration and support requirements are described in *ViPR Controller Virtual Data Center Requirements and Information Guide*.
- The LINUX host must be running the correct version of LINUX, and be properly configured.
- Authenticate with the ViPR REST API as a Tenant Administrator. See [Authenticate with the REST API on page 9](#).
- `type - Linux`
- `discoverable`
 - `true` = Allow ViPR Controller to automatically discover and register the host. This is the default setting if you do not include it in your request payload.
 - `false` = Do not allow ViPR Controller to automatically discover the host. You must manually register the host initiators in ViPR Controller using `POST /compute/hosts/{Host_URN}/initiators`.
- `port_number` - The SSH connection port. The default is 22.
- The *EMC ViPR Controller REST API Reference* on the [ViPR Controller Product Documentation Index](#) provides a description and complete list of parameters for the REST API methods used in this section.

Procedure

1. Get the URN of your tenant.

```
GET https://<ViPR_Controller_VIP>:4443/tenant
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>
```

- Use the tenant URN in the request payload to create a host. The `POST /compute/hosts` method also initiates discovery of the host and its initiator ports.

Request

```
POST https://<ViPR_Controller_VIP>:4443/compute/hosts
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<host_create>
  <type>Linux</type>
  <host_name>lg1w7151.example.com</host_name>
  <name>lg1w7151</name>
  <port_number>22</port_number>
  <user_name>root</user_name>
  <password>Password1</password>
  <use_ssl>false</use_ssl>
  <tenant>urn:storageos:TenantOrg:c104b49e-4e6b-449c-b92b-
a19b247a613b:global</tenant>
</host_create>
```

Response

```
HTTP 202 Accepted
Content-Type: application/xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<task>
  <creation_time>1423511829227</creation_time>
  <global>false</global>
  <id>urn:storageos:Task:5cbe14c7-5540-4d98-abea-
fcf5ba3896a0:vdcl</id>
  <inactive>false</inactive>
  <internal>false</internal>
  <link rel="self" href="/vdc/tasks/urn:storageos:Task:
5cbe14c7-5540-4d98-abea-fcf5ba3896a0:vdcl"/>
  <name>DISCOVER HOST</name>
  <remote>false</remote>
  <tags/>
  <vdc>
    <id>urn:storageos:VirtualDataCenter:99c5c185-
ebb0-468a-8fe7-3ea6e65b7b57:vdcl</id>
    <link rel="self" href="/vdc/
urn:storageos:VirtualDataCenter:99c5c185-
ebb0-468a-8fe7-3ea6e65b7b57:vdcl"/>
  </vdc>
  <associated_resources/>
  <description>discover a compute host</description>
  <op_id>f9b296b2-ad28-4074-9b23-0e5766530f27</op_id>
  <progress>0</progress>
  <resource>
    <id>urn:storageos:Host:a9a4a18d-df40-443c-9ab0-
e83823d156fa:vdcl</id>
    <link rel="self" href="/compute/hosts/
urn:storageos:Host:a9a4a18d-df40-443c-9ab0-
e83823d156fa:vdcl"/>
    <name>lg1w7151</name>
  </resource>
  <start_time>1423511829226</start_time>
  <state>pending</state>
</task>
```

You will not be able to fully manage the host until you discover it.

- Check the user interface to see if the host has completed discovery, or check the host resource. Use the resource URI included in the task returned from the create a host request. It should contain the information shown in the following example.

Request

```
GET https://<ViPR_Controller_VIP>:4443/compute/hosts/
urn:storageos:Host:ead39b5a-07f4-4cb4-8124-35aa864fe760:
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>
```

Response

```
HTTP 200 OK
Content-Type: application/xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<host>
  <creation_time>1423511829212</creation_time>
  <global>>false</global>
  <id>urn:storageos:Host:a9a4a18d-df40-443c-9ab0-
e83823d156fa:vdcl</id>
  <inactive>>false</inactive>
  <internal>>false</internal>
  <link rel="self" href="/compute/hosts/
urn:storageos:Host:a9a4a18d-df40-443c-9ab0-e83823d156fa:vdcl"/>
  <name>lglw7151</name>
  <remote>>false</remote>
  <tags/>
  <vdc>
    <id>urn:storageos:VirtualDataCenter:99c5c185-
ebb0-468a-8fe7-3ea6e65b7b57:vdcl</id>
    <link rel="self" href="/vdc/
urn:storageos:VirtualDataCenter:99c5c185-
ebb0-468a-8fe7-3ea6e65b7b57:vdcl"/>
    </vdc>
    <native_guid>6C:41:6A:1E:71:68</native_guid>
    <compatibility_status>COMPATIBLE</compatibility_status>
    <job_discovery_status>COMPLETE</job_discovery_status>
    <last_discovery_run_time>1423511839761</
last_discovery_run_time>
    <last_discovery_status_message></
last_discovery_status_message>
    <last_metering_run_time>0</last_metering_run_time>
    <job_metering_status>CREATED</job_metering_status>
    <next_discovery_run_time>1423598229542</
next_discovery_run_time>
    <next_metering_run_time>0</next_metering_run_time>
    <registration_status>REGISTERED</registration_status>
    <success_discovery_time>1423511839761</success_discovery_time>
    <success_metering_time>0</success_metering_time>
    <tenant>
      <id>urn:storageos:TenantOrg:c104b49e-4e6b-449c-b92b-
a19b247a613b:global</id>
      <link rel="self" href="/tenants/
urn:storageos:TenantOrg:c104b49e-4e6b-449c-b92b-
a19b247a613b:global"/>
      </tenant>
      <discoverable>>true</discoverable>
      <host_name>lglw7151.example.com</host_name>
      <os_version>RHEL 6.3.0.3.el6</os_version>
      <port_number>22</port_number>
      <type>Linux</type>
      <use_ssl>>false</use_ssl>
      <user_name>root</user_name>
    </host>
```


Add and discover a Windows host

To manage a Windows host in ViPR Controller, use `POST /compute/hosts` to add the host to ViPR Controller and discover the host, and its initiator ports.

Before you begin

- Your Windows host must be running the correct version of Windows, and be properly configured. In particular, your ViPR Controller installation, the Windows host, and the user name you use to authenticate must all be in the same Windows domain. See the *ViPR Controller Virtual Data Center Requirements and Information Guide* on the [ViPR Controller Product Documentation Index](#).
- Authenticate with the ViPR REST API as a Tenant Administrator. See [Authenticate with the ViPR REST API on page 9](#).
- `type` - Windows
- `discoverable`
 - `true` = Allow ViPR Controller to automatically discover and register the host. This is the default setting if you do not include it in your request payload.
 - `false` = Do not allow ViPR Controller to automatically discover the host. You must manually register the host initiators in ViPR Controller using `POST /compute/hosts/{Host_URN}/initiators`.
- `port_number` - The SSH connection port. The default is 5986.
- `host_name` - When using domain credentials, this must be the hostname (not IP address) and resolvable through DNS.
- The *EMC ViPR Controller REST API Reference* on the [ViPR Controller Product Documentation Index](#) provides a description and complete list of parameters for the REST API methods used in this section.

Procedure

1. Get the URN of your tenant.

```
GET https://<ViPR_Controller_VIP>:4443/tenant
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>
```

2. Use the tenant URN as one of the parameters to create a host. The `POST /compute/hosts` method creates a host and initiates discovery of the host and its initiator ports.

Request

```
POST https://<ViPR_Controller_VIP>:4443/compute/hosts
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<host_create>
  <type>Windows</type>
  <host_name>192.168.0.0</host_name>
  <name>hostlwin</name>
  <port_number>5986</port_number>
  <user_name>root</user_name>
  <password>Password123</password>
  <use_ssl>>false</use_ssl>
```

```

    <tenant>urn:storageos:TenantOrg:167d6c87-209e-4f9a-a688-
    c1c73a2c9de8:global</tenant>
  </host_create>

```

Response

```

HTTP 202 Accepted
Content-Type: application/xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<task>
  <creation_time>1415198092146</creation_time>
  <global>>false</global>
  <id>urn:storageos:Task:b9366da9-31ed-4809-
  ad46-74364f83e637:vdc1</id>
  <inactive>>false</inactive>
  <internal>>false</internal>
  <link rel="self" href="/vdc/tasks/
  urn:storageos:Task:b9366da9-31ed-4809-ad46-74364f83e637:vdc1"/>
  <name>DISCOVER HOST</name>
  <remote>>false</remote>
  <tags/>
  <vdc>
    <id>urn:storageos:VirtualDataCenter:d649923c-3d28-4195-
    bd6e-d7139c0df033:vdc1</id>
    <link rel="self" href="/vdc/
    urn:storageos:VirtualDataCenter:d649923c-3d28-4195-bd6e-
    d7139c0df033:vdc1"/>
  </vdc>
  <associated_resources/>
  <op_id>9e150aa2-d92e-47ee-8fab-d86733d852d0</op_id>
  <progress>0</progress>
  <resource>
    <id>urn:storageos:Host:4582fc0e-ff38-462a-9a18-
    f7488d4dd940:vdc1</id>
    <link rel="self" href="/compute/hosts/urn:storageos:Host:
    4582fc0e-ff38-462a-9a18-f7488d4dd940:vdc1"/>
    <name>host1win</name>
  </resource>
  <start_time>1415198092145</start_time>
  <state>pending</state>
  <tenant>
    <id>urn:storageos:TenantOrg:167d6c87-209e-4f9a-a688-
    c1c73a2c9de8:global</id>
    <link rel="self" href="/tenants/urn:storageos:TenantOrg:
    167d6c87-209e-4f9a-a688-c1c73a2c9de8:global"/>
  </tenant>
</task>

```

You will not be able to fully manage the host until discovery of the host, and its initiator ports has completed.

3. Check the user interface to see if the host has completed discovery, or use the host URN contained in the task returned from the create request to check the host resource. The host should contain the status shown in the following example.

Request

```

GET https://<ViPR_Controller_VIP>:4443/compute/hosts/
urn:storageos:Host:4582fc0e-ff38-462a-9a18-f7488d4dd940:vdc1
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

```

Response

```

HTTP 200 OK
Content-Type: application/xml

```

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<host>
  <creation_time>1415198092133</creation_time>
  <global>>false</global>
  <id>urn:storageos:Host:4582fc0e-ff38-462a-9a18-
f7488d4dd940:vdcl</id>
  <inactive>>false</inactive>
  <internal>>false</internal>
  <link rel="self" href="/compute/hosts/urn:storageos:Host:
4582fc0e-ff38-462a-9a18-f7488d4dd940:vdcl"/>
  <name>host1win</name>
  <remote>>false</remote>
  <tags/>
  <vdc>
    <id>urn:storageos:VirtualDataCenter:d649923c-3d28-4195-
bd6e-d7139c0df033:vdcl</id>
    <link rel="self" href="/vdc/
urn:storageos:VirtualDataCenter:d649923c-3d28-4195-bd6e-
d7139c0df033:vdcl"/>
  </vdc>
  <native_guid/>
  <compatibility_status>COMPATIBLE</compatibility_status>
  <job_discovery_status>COMPLETE</job_discovery_status>
  <last_discovery_run_time>1415198092615</
last_discovery_run_time>
  <last_discovery_status_message/>
  <last_metering_run_time>0</last_metering_run_time>
  <job_metering_status>CREATED</job_metering_status>
  <next_discovery_run_time>1415284492176</
next_discovery_run_time>
  <next_metering_run_time>0</next_metering_run_time>
  <registration_status>REGISTERED</registration_status>
  <success_discovery_time>0</success_discovery_time>
  <success_metering_time>0</success_metering_time>
  <tenant>
    <id>urn:storageos:TenantOrg:167d6c87-209e-4f9a-a688-
c1c73a2c9de8:global</id>
    <link rel="self" href="/tenants/urn:storageos:TenantOrg:
167d6c87-209e-4f9a-a688-c1c73a2c9de8:global"/>
  </tenant>
  <discoverable>>true</discoverable>
  <host_name>192.168.0.0</host_name>
  <port_number>5986</port_number>
  <type>Windows</type>
  <use_ssl>>false</use_ssl>
  <user_name>root</user_name>
</host>

```

Add an AIX host or AIX VIO server to ViPR Controller

To manage an AIX host or AIX VIO host, use POST `/compute/hosts`.

Before you begin

- Pre-configuration and support requirements are described in the *ViPR Controller Virtual Data Center Requirements and Information Guide* on the [ViPR Controller Product Documentation Index](#).
- Authenticate with the ViPR REST API as a Tenant Administrator. See [Authenticating with the REST API on page 9](#).
- type
 - AIX = Adding an AIX host.
 - AIXVIO = Adding an AIX VIO host.

- discoverable
 - true = Allow ViPR Controller to automatically discover and register the host. This is the default setting if you do not include it in your request payload.
 - false = Do not allow ViPR Controller to automatically discover the host. You must manually register the host initiators in ViPR Controller using `POST /compute/hosts/{Host_URN}/initiators`.
- port_number - The SSH connection port. The default is 22.
- The *EMC ViPR Controller REST API Reference* on the [ViPR Controller Product Documentation Index](#) provides a description and complete list of parameters for the REST API methods used in this section.

Procedure

1. Get the URN of your tenant.

```
GET https://<ViPR_Controller_VIP>:4443/tenant
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>
```

2. Use the tenant URN as one of the parameters to create a host. The `POST /compute/hosts` method creates a host and initiates discovery of the host and its initiator ports.

In this example, an AIX is added to ViPR Controller.

Request

```
POST https://<ViPR_Controller_VIP>:4443/compute/hosts
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<host_create>
  <type>AIX</type>
  <host_name>192.168.0.0</host_name>
  <name>host1aix</name>
  <port_number>22</port_number>
  <user_name>root</user_name>
  <password>pancake</password>
  <use_ssl>>false</use_ssl>
  <tenant>urn:storageos:TenantOrg:167d6c87-209e-4f9a-a688-
c1c73a2c9de8:global</tenant>
</host_create>
```

Response

```
HTTP 202 Accepted

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<task>
  <creation_time>1415137101100</creation_time>
  <global>>false</global>
  <id>urn:storageos:Task:743ecb3f-349a-4b95-ab2a-
b7365ae13e29:vdcl</id>
  <inactive>>false</inactive>
  <internal>>false</internal>
  <link rel="self" href="/vdc/tasks/urn:storageos:Task:
743ecb3f-349a-4b95-ab2a-b7365ae13e29:vdcl"/>
  <name>DISCOVER HOST</name>
  <remote>>false</remote>
  <tags/>
</vdc>
```

```

        <id>urn:storageos:VirtualDataCenter:d649923c-3d28-4195-
bd6e-d7139c0df033:vdcl</id>
        <link rel="self" href="/vdc/
urn:storageos:VirtualDataCenter:d649923c-3d28-4195-bd6e-
d7139c0df033:vdcl"/>
    </vdc>
    <associated_resources/>
    <op_id>546f5647-4cf3-4fc9-a551-6f1ba934bc4e</op_id>
    <progress>0</progress>
    <resource>
        <id>urn:storageos:Host:d9d2312c-d172-4099-b3dd-
e17b49b01797:vdcl</id>
        <link rel="self" href="/compute/hosts/
urn:storageos:Host:d9d2312c-d172-4099-b3dd-e17b49b01797:vdcl"/>
        <name>hostlaix</name>
    </resource>
    <start_time>1415137101097</start_time>
    <state>pending</state>
    <tenant>
        <id>urn:storageos:TenantOrg:167d6c87-209e-4f9a-a688-
c1c73a2c9de8:global</id>
        <link rel="self" href="/tenants/urn:storageos:TenantOrg:
167d6c87-209e-4f9a-a688-c1c73a2c9de8:global"/>
    </tenant>
</task>

```

You will not be able to fully manage the host until discovery of the host, and its initiator ports has completed.

3. Check the user interface to see if the host has completed discovery, or use the host URN contained in the task returned from the create request to check the host resource. It should contain the information shown in the following example.

Request

```

POST https://<ViPR_Controller_VIP>:4443/compute/hosts/
urn:storageos:Host:d9d2312c-d172-4099-b3dd-e17b49b01797:vdcl
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

```

Response

```

HTTP 200 OK

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<host>
    <creation_time>1415137101050</creation_time>
    <global>false</global>
    <id>urn:storageos:Host:d9d2312c-d172-4099-b3dd-
e17b49b01797:vdcl</id>
    <inactive>false</inactive>
    <internal>false</internal>
    <link rel="self" href="/compute/hosts/
urn:storageos:Host:d9d2312c-d172-4099-b3dd-e17b49b01797:vdcl"/>
    <name>hostlaix</name>
    <remote>false</remote>
    <tags/>
    <vdc>
        <id>urn:storageos:VirtualDataCenter:d649923c-3d28-4195-
bd6e-d7139c0df033:vdcl</id>
        <link rel="self" href="/vdc/
urn:storageos:VirtualDataCenter:d649923c-3d28-4195-bd6e-
d7139c0df033:vdcl"/>
    </vdc>
    <native_guid>BA:10:95:D0:7C:02</native_guid>
    <compatibility_status>COMPATIBLE</compatibility_status>

```

```

    <job_discovery_status>COMPLETE</job_discovery_status>
    <last_discovery_run_time>1415137121159</
last_discovery_run_time>
    <last_discovery_status_message/>
    <last_metering_run_time>0</last_metering_run_time>
    <job_metering_status>CREATED</job_metering_status>
    <next_discovery_run_time>1415223501712</
next_discovery_run_time>
    <next_metering_run_time>0</next_metering_run_time>
    <registration_status>REGISTERED</registration_status>
    <success_discovery_time>1415137121159</success_discovery_time>
    <success_metering_time>0</success_metering_time>
    <tenant>
      <id>urn:storageos:TenantOrg:167d6c87-209e-4f9a-a688-
c1c73a2c9de8:global</id>
      <link rel="self" href="/tenants/urn:storageos:TenantOrg:
167d6c87-209e-4f9a-a688-c1c73a2c9de8:global"/>
    </tenant>
    <discoverable>true</discoverable>
    <host_name>192.168.0.0</host_name>
    <os_version>7.1.0.0
</os_version>
<port_number>22</port_number>
<type>AIX</type>
<use_ssl>>false</use_ssl>
<user_name>root</user_name>
</host>

```

Add a host other than Windows, AIX or Linux

Adding a host other than Windows, AIX or Linux requires you to manually register the initiators with the host after adding the host. You can also manually assign host initiators and interfaces to any host you are registering with ViPR Controller .

Before you begin

- Authenticate with the ViPR REST API as a Tenant Administrator. See [Authenticating with the REST API on page 9](#).
- HP-UX hosts are not automatically discovered and registered in ViPR Controller. When you set `type` equal to `HPUX` it:
 - Sets the Volume Set Addressing (VSA) flag, which is required for exporting EMC VMAX, and VPLEX volumes to HP-UX hosts.
 - Is required to use the Host Mode Option when provisioning with HDS storage systems.

Note

The Host Mode Option is set by using `PUT /config/controller/{ID}` where `{ID}` is the ID of the `HDSHostModeOption` controller type.

- For hosts other than HP-UX, set `type` to `Other` .
- When you add an HP-UX host to ViPR, you will still need to manually add and register the host initiators in ViPR Controller
- The *EMC ViPR Controller REST API Reference* on the [ViPR Controller Product Documentation Index](#) provides a description and complete list of parameters for the REST API methods used in this section.
- You need the following information:

- The name of the host being registered.
- The fully qualified domain name or IP address of the host.
- The node address of an initiator
- The port address of an initiator.

In this example, a host other than Windows, AIX or Linux is added. Once the host is added, an initiator is manually registered with the host.

Procedure

1. Get the URN of your tenant.

Request

```
GET https://<ViPR_Controller_VIP>:4443/tenant
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>
```

2. Use the tenant URN as one of the parameters to add the host to ViPR Controller.

- If the host will be attached to EMC VMAX volumes, use `<type>HPUX</type>`
- Otherwise, use `<type>Other</type>`

Note

Hosts of type Other, and HPUX are not automatically discovered, therefore, set the `<discoverable>` flag to false.

Request

```
POST https://<ViPR_Controller_VIP>:4443/compute/hosts
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<host_create>
  <type>Other</type>
  <host_name>192.168.0.1</host_name>
  <name>myhost2</name>
  <discoverable>>false</discoverable>
  <tenant>urn:storageos:TenantOrg:c104b49e-4e6b-449c-b92b-
a19b247a613b:global</tenant>
</host_create>
```

Response

```
HTTP 202 Accepted
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<task>
  <creation_time>1423519740612</creation_time>
  <global>>false</global>
  <id>urn:storageos:Task:7692ec89-
c388-40f1-9d5d-70029556fe9b:vdc1</id>
  <inactive>>false</inactive>
  <internal>>false</internal>
  <link rel="self" href="/vdc/tasks/urn:storageos:Task:7692ec89-
c388-40f1-9d5d-70029556fe9b:vdc1"/>
  <name>DISCOVER HOST</name>
  <remote>>false</remote>
```

```

    <tags/>
    <vdc>
      <id>urn:storageos:VirtualDataCenter:99c5c185-
ebb0-468a-8fe7-3ea6e65b7b57:vdc1</id>
      <link rel="self" href="/vdc/
urn:storageos:VirtualDataCenter:99c5c185-
ebb0-468a-8fe7-3ea6e65b7b57:vdc1"/>
    </vdc>
    <associated_resources/>
    <description>discover a compute host</description>
    <end_time>1423519740611</end_time>
    <message>Host is not discoverable</message>
    <op_id>b738e081-e671-44be-959d-cd670ceaf75f</op_id>
    <progress>100</progress>
    <resource>
      <id>urn:storageos:Host:4f96c363-f6dd-4e5f-968e-
a4778671972b:vdc1</id>
      <link rel="self" href="/compute/hosts/urn:storageos:Host:
4f96c363-f6dd-4e5f-968e-a4778671972b:vdc1"/>
      <name>myhost2</name>
    </resource>
    <start_time>1423519740610</start_time>
    <state>ready</state>
    <tenant>
      <id>urn:storageos:TenantOrg:c104b49e-4e6b-449c-b92b-
a19b247a613b:global</id>
      <link rel="self" href="/tenants/
urn:storageos:TenantOrg:c104b49e-4e6b-449c-b92b-
a19b247a613b:global"/>
    </tenant>
  </task>

```

If the state from the returned task is `ready`, the host is now added to ViPR Controller.

3. Register an initiator associated with the host by calling the following POST . The new host initiator is returned.

Request

```

POST https://<ViPR_Controller_VIP>:4443/compute/hosts/{Host_URN}/
initiators
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<initiator_create>
  <protocol>FC</protocol>
  <initiator_port>10:13:27:65:60:38:68:BE</initiator_port>
</initiator_create>

```

Response

```

HTTP 200 OK
Content-Type: application/xml
<initiator>
  <creation_time>1379202603661</creation_time>
  <id>urn:storageos:Initiator:07b2e71d-
cb4c-49c9-94fe-1feab7878d35:vdc1</id>
  <inactive>false</inactive>
  <link
    href="/compute/initiators/urn:storageos:Initiator:
07b2e71d-cb4c-49c9-94fe-1feab7878d35:vdc1"
    rel="self"/>
  <tags/>
  <host>
    <id>urn:storageos:Host:c2fe902f-0de0-4a92-
a2a9-46ba87279bd2:vdc1</id>
    <link

```



```

        href="/compute/hosts/
urn:storageos:Host:c2fe902f-0de0-4a92-a2a9-46ba87279bd2:vdc1"
        rel="self"/>
    </host>
    <protocol>FC</protocol>
    <hostname>myhost</hostname>
    <initiator_port>10:13:27:65:60:38:68:BE</initiator_port>
</initiator>

```

Add host ports to an IP network

If adding hosts to provision over an IP network, the host ports must be added to the IP network.

IP networks are created by system administrators. Tenant administrators cannot configure IP networks or add host ports to the network.

You use `POST /vdc/networks` to create a new IP network or `PUT /vdc/networks/{ID}` to add your host ports to an existing IP network.

Add a host to a cluster

Optionally, hosts can be added to ViPR Controller clusters. Adding hosts to clusters allows service operations to be performed exclusively on a single host, or shared across all the hosts in a cluster.

Before you begin

- Authenticate with the ViPR REST API as a Tenant Administrator. See [Authenticating with the REST API on page 9](#).
- Clusters can only contain the same type of hosts.
- A host can only be used in one cluster.
- You need the following information:
 - The URN of the tenant, if you need to create the cluster before adding the host. To get the URN of the current tenant, use `GET /tenant`.
 - The URN of the host being added to the cluster. To determine the ID of the host, you can use `GET /compute/hosts/bulk` to get the list of all hosts IDs and then `GET /compute/hosts/{Host_URN}` to show the details of a specific host.
 - The URN of the cluster.

The *EMC ViPR Controller REST API Reference* on the [ViPR Controller Product Documentation Index](#) provides a description and complete list of parameters for the REST API methods used in this section.

Procedure

1. If the cluster does not already exist, create one with the following POST.

Request

```

POST https://<ViPR_Controller_VIP>:4443/tenants/{Tenant_URN}/
clusters
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>
<cluster_create>

```

```
<name>Win_cluster_1</name>
</cluster_create>
```

Response

```
HTTP 200 OK
Content-Type: application/xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<cluster>
  <creation_time>1399748901762</creation_time>
  <global>false</global>
  <id>urn:storageos:Cluster:
2f6b27f2-5eb9-4b8f-8519-050a43f3c733:vdc1</id>
  <inactive>false</inactive>
  <internal>false</internal>
  <link rel="self" href="/compute/clusters/urn:storageos:Cluster:
2f6b27f2-5eb9-4b8f-8519-050a43f3c733:vdc1"/>
  <name>Win_cluster_1</name>
  <remote>false</remote>
  <tags/>
  <vdc>
    <id>urn:storageos:VirtualDataCenter:440b70de-fd36-4361-
a455-b6ca65fb8228:vdc1</id>
    <link rel="self" href="/vdc/
urn:storageos:VirtualDataCenter:440b70de-fd36-4361-a455-
b6ca65fb8228:vdc1"/>
  </vdc>
  <tenant>
    <id>urn:storageos:TenantOrg:6c7dde31-
ec48-4028-8672-5e74f5754656:global</id>
    <link rel="self" href="/tenants/urn:storageos:TenantOrg:
6c7dde31-ec48-4028-8672-5e74f5754656:global"/>
  </tenant>
</cluster>
```

2. A host is added to a cluster by updating the `<cluster>` attribute of a host, using `PUT /compute/hosts/{Host_URN}`. This example uses the URN of the new cluster, from the cluster create call. A task is returned.

Request

```
PUT https://<ViPR_Controller_VIP>:4443/compute/hosts/
urn:storageos:Host:24e9c36a-804d-4d05-a646-a64a47accc7e:vdc1
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<host_update>
  <cluster>urn:storageos:Cluster:9a18479e-d36e-44cc-bdae-
f8d44720dbe3:vdc1</cluster>
</host_update>
```

Response

```
HTTP/1.1 202 Accepted
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<task>
  <associated_resources/>
  <description>DISCOVER_HOST</description>
  <op_id>5e7a5c9d-2a37-4f43-9534-727dc396804f</op_id>
  <resource>
    <id>urn:storageos:Host:24e9c36a-804d-4d05-a646-
```

```

a64a47acc7e:vdc1</id>
  <link rel="self" href="/compute/hosts/urn:storageos:Host:
24e9c36a-804d-4d05-a646-a64a47acc7e:vdc1"/>
  <name>host219</name>
</resource>
<link rel="self" href="/compute/hosts/urn:storageos:Host:
24e9c36a-804d-4d05-a646-a64a47acc7e:vdc1/tasks/
5e7a5c9d-2a37-4f43-9534-727dc396804f"/>
  <start_time>1399749237037</start_time>
  <state>pending</state>
</task>

```

Results

If the message from the returned task is `Operation completed successfully`, the host is now added to the cluster.

Replace Host Initiators after a storage volume has been exported to a host

Once you have used ViPR Controller to export a volume to a host, you can add a host initiator to the export, remove a host initiator from the export, or perform both actions to swap out one host initiator for another in the export group.

Update ViPR Controller after a host initiator is replaced outside of ViPR

After you use ViPR Controller to export a volume to an AIX, ESX, Linux, or Windows host, which was added to ViPR Controller as discoverable, you can, add, remove, or replace a host initiator used by ViPR Controller for an export operation directly on the host, using an application other than ViPR Controller.

Before you begin

- The ID of the host containing the initiator that was replaced. You can use `GET /compute/hosts` to get a lists of the IDs, URIs, and names of all hosts in the system.
- Authenticate with the ViPR REST API as a System Administrator for the network-related REST API calls. See [Authenticating with the REST API on page 9](#).
- Authenticate with the ViPR REST API as a Tenant Administrator for the host-related REST API calls.

After you add, remove, or replace a host initiator that was used by ViPR Controller in an export operation, using an application other than ViPR Controller, you must rediscover the network-system (fabric manager) which provides connectivity between the host and the storage.

Procedure

1. Get a list of the available network-systems using `GET /vdc/network-systems`. The list includes the name, ID, and URI of each of network systems (fabric managers).

Request

```

GET https://<ViPR_Controller_VIP>:4443/vdc/network-systems
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

```

Response

```

HTTP 200 OK
Content-Type: application/xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<network_systems>
  <network_system>
    <id>urn:storageos:NetworkSystem:
7875ed0e-1d2a-430d-94d0-20fcd7e38bb4:vdc1</id>
    <link rel="self" href="/vdc/network-systems/
urn:storageos:NetworkSystem:
7875ed0e-1d2a-430d-94d0-20fcd7e38bb4:vdc1"/>
    <name>brocade_west1</name>
  </network_system>
</network_systems>

```

2. Select the network system on which the network between the host and the storage is configured and rediscover the network system using POST /vdc/network-systems/{id}/discover. A task is returned whose URI can be queried to determine the status of the task.

Request

```

POST https://<ViPR_Controller_VIP>:4443/vdc/network-systems/
urn:storageos:NetworkSystem:
7875ed0e-1d2a-430d-94d0-20fcd7e38bb4:vdc1/discover
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

```

Response

```

HTTP 202 Accepted
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<task>
  <creation_time>1423664227579</creation_time>
  <global>false</global>
  <id>urn:storageos:Task:e2fdf765-0086-4a1c-8645-
c58fa66f8cc0:vdc1</id>
  <inactive>false</inactive>
  <internal>false</internal>
  <link rel="self" href="/vdc/tasks/
urn:storageos:Task:e2fdf765-0086-4a1c-8645-
c58fa66f8cc0:vdc1"/>
  <name>DISCOVER NETWORKSYSTEM</name>
  <remote>false</remote>
  <tags/>
  <vdc>
    <id>urn:storageos:VirtualDataCenter:99c5c185-
ebb0-468a-8fe7-3ea6e65b7b57:vdc1</id>
    <link rel="self" href="/vdc/
urn:storageos:VirtualDataCenter:99c5c185-
ebb0-468a-8fe7-3ea6e65b7b57:vdc1"/>
  </vdc>
  <associated_resources/>
  <description>discover one network system</description>
  <op_id>553dba5e-509b-48be-b586-370cbe243438</op_id>
  <progress>0</progress>
  <resource>
    <id>urn:storageos:NetworkSystem:
7875ed0e-1d2a-430d-94d0-20fcd7e38bb4:vdc1</id>
    <link rel="self" href="/vdc/network-systems/
urn:storageos:NetworkSystem:
7875ed0e-1d2a-430d-94d0-20fcd7e38bb4:vdc1"/>

```

```

        <name>brocade_west1</name>
      </resource>
    <start_time>1423664227577</start_time>
    <state>pending</state>
  </task>

```

3. Query the network system discover task, using the task ID from the response body of the POST request. When the message attribute of the task is Operation completed successfully, the operation has completed and the network system has been rediscovered.

Request

```

GET https://<ViPR_Controller_VIP>:4443/vdc/tasks/
urn:storageos:Task:e2fdf765-0086-4a1c-8645-c58fa66f8cc0:vdc1
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

```

Response

```

HTTP 200 OK
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<task>
  <creation_time>1423664227579</creation_time>
  <global>false</global>
  <id>urn:storageos:Task:e2fdf765-0086-4a1c-8645-
c58fa66f8cc0:vdc1</id>
  <inactive>false</inactive>
  <internal>false</internal>
  <link rel="self" href="/vdc/tasks/
urn:storageos:Task:e2fdf765-0086-4a1c-8645-c58fa66f8cc0:vdc1"/>
  <name>DISCOVER NETWORKSYSTEM</name>
  <remote>false</remote>
  <tags/>
  <vdc>
    <id>urn:storageos:VirtualDataCenter:99c5c185-
ebb0-468a-8fe7-3ea6e65b7b57:vdc1</id>
    <link rel="self" href="/vdc/
urn:storageos:VirtualDataCenter:99c5c185-
ebb0-468a-8fe7-3ea6e65b7b57:vdc1"/>
  </vdc>
  <associated_resources/>
  <description>discover one network system</description>
  <end_time>1423664228010</end_time>
  <message>Operation completed successfully</message>
  <op_id>553dba5e-509b-48be-b586-370cbe243438</op_id>
  <progress>100</progress>
  <resource>
    <id>urn:storageos:NetworkSystem:
7875ed0e-1d2a-430d-94d0-20fcd7e38bb4:vdc1</id>
    <link rel="self" href="/vdc/network-systems/
urn:storageos:NetworkSystem:
7875ed0e-1d2a-430d-94d0-20fcd7e38bb4:vdc1"/>
    <name>brocade_west1</name>
  </resource>
  <start_time>1423664227577</start_time>
  <state>ready</state></task>

```

4. Rediscover the host that is associated with the replaced initiator so that ViPR Controller can rediscover all of the initiators on the host. During host discovery, if any new or removed initiators are discovered on the host, existing exports will be automatically updated. Use POST /compute/hosts/{id}/discover. A task is returned whose URI can be queried to determine the status of the task.

Request

```
POST https://<ViPR_Controller_VIP>:4443/compute/hosts/{host_ID}/
discover
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>
```

Response

```
HTTP 202 Accepted
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<task>
  <creation_time>1424181597255</creation_time>
  <global>false</global>
  <id>urn:storageos:Task:
933b8a8e-8223-49f6-8d52-8ddfb03cfed9:vdc1</id>
  <inactive>false</inactive>
  <internal>false</internal>
  <link rel="self" href="/vdc/tasks/urn:storageos:Task:
933b8a8e-8223-49f6-8d52-8ddfb03cfed9:vdc1"/>
  <name>DISCOVER HOST</name>
  <remote>false</remote>
  <tags/>
  <vdc>
    <id>urn:storageos:VirtualDataCenter:99c5c185-
ebb0-468a-8fe7-3ea6e65b7b57:vdc1</id>
    <link rel="self" href="/vdc/
urn:storageos:VirtualDataCenter:99c5c185-
ebb0-468a-8fe7-3ea6e65b7b57:vdc1"/>
  </vdc>
  <associated_resources/>
  <description>discover a compute host</description>
  <op_id>82b9ca2f-a30a-475f-b7cd-f374b1995e32</op_id>
  <progress>0</progress>
  <resource>
    <id>urn:storageos:Host:a9a4a18d-df40-443c-9ab0-
e83823d156fa:vdc1</id>
    <link rel="self" href="/compute/hosts/
urn:storageos:Host:a9a4a18d-df40-443c-9ab0-e83823d156fa:vdc1"/>
    <name>lglw7151</name>
  </resource>
  <start_time>1424181597253</start_time>
  <state>pending</state>
  <tenant>
    <id>urn:storageos:TenantOrg:c104b49e-4e6b-449c-b92b-
a19b247a613b:global</id>
    <link rel="self" href="/tenants/
urn:storageos:TenantOrg:c104b49e-4e6b-449c-b92b-
a19b247a613b:global"/>
  </tenant>
</task>
```

5. Query the host discover task, using the task ID from the response body of the POST request. When the message attribute of the task is Operation completed successfully, the operation has completed and the host has been rediscovered.

Request

```
POST https://<ViPR_Controller_VIP>:4443/vdc/tasks/
urn:storageos:Task:933b8a8e-8223-49f6-8d52-8ddfb03cfed9:vdc1
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>
```

Response

```

HTTP 200 OK
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<task>
  <creation_time>1424181597255</creation_time>
  <global>false</global>
  <id>urn:storageos:Task:
933b8a8e-8223-49f6-8d52-8ddfb03cfed9:vdc1</id>
  <inactive>false</inactive>
  <internal>false</internal>
  <link rel="self" href="/vdc/tasks/urn:storageos:Task:
933b8a8e-8223-49f6-8d52-8ddfb03cfed9:vdc1"/>
  <name>DISCOVER HOST</name>
  <remote>false</remote>
  <tags/>
  <vdc>
    <id>urn:storageos:VirtualDataCenter:99c5c185-
ebb0-468a-8fe7-3ea6e65b7b57:vdc1</id>
    <link rel="self" href="/vdc/
urn:storageos:VirtualDataCenter:99c5c185-
ebb0-468a-8fe7-3ea6e65b7b57:vdc1"/>
    </vdc>
    <associated_resources/>
    <description>discover a compute host</description>
    <end_time>1424181602095</end_time>
    <message>Operation completed successfully</message>
    <op_id>82b9ca2f-a30a-475f-b7cd-f374b1995e32</op_id>
    <progress>100</progress>
    <resource>
      <id>urn:storageos:Host:a9a4a18d-df40-443c-9ab0-
e83823d156fa:vdc1</id>
      <link rel="self" href="/compute/hosts/
urn:storageos:Host:a9a4a18d-df40-443c-9ab0-e83823d156fa:vdc1"/>
      <name>lg1w7151</name>
    </resource>
    <start_time>1424181597253</start_time>
    <state>ready</state>
    <tenant>
      <id>urn:storageos:TenantOrg:c104b49e-4e6b-449c-b92b-
a19b247a613b:global</id>
      <link rel="self" href="/tenants/
urn:storageos:TenantOrg:c104b49e-4e6b-449c-b92b-
a19b247a613b:global"/>
    </tenant>
  </task>

```

Replace a host initiator of an undiscovered host after a ViPR Controller export operation.

If ViPR Controller was used to export a volume to a host, which was added to ViPR Controller as “Other,” or if the host was added as undiscoverable, use the following procedures to replace a host initiator with a different host initiator after the export operation.

Before you begin

- The name of the initiator being added and the name of the initiator being removed.
- The ID of the initiator being removed.
- The ID of the host containing the initiators being added and replaced.
- Authenticate with the ViPR REST API as a System Administrator for the network-related REST API calls. See [Authenticating with the REST API on page 9](#).

- Authenticate with the ViPR REST API as a Tenant Administrator for the host-related and initiator-related REST API calls.
- The *EMC ViPR Controller REST API Reference* on the [ViPR Controller Product Documentation Index](#) provides a description and complete list of parameters for the REST API methods used in this article.

In ViPR Controller, add the new host initiator to the same network from which the host initiator is being replaced, so that ViPR Controller will see the connectivity between the host and the storage after the swap is complete. ViPR Controller automatically updates the host initiators in the export group after the add, and remove operations are detected in ViPR Controller

Procedure

1. Get the list of networks using GET /vdc/networks.

Request

```
GET https://<ViPR_Controller_VIP>:4443/vdc/networks
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>
```

Response

```
HTTP 200 OK
Content-Type: application/xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<networks>
  <network>
    <id>urn:storageos:Network:57351790-d19b-4a5b-
b77b-7cbea6f3b0fe:vdc1</id>
    <link rel="self" href="/vdc/networks/
urn:storageos:Network:57351790-d19b-4a5b-b77b-7cbea6f3b0fe:vdc1"/>
    <name>FABRIC_fake array fabric</name>
  </network>
  <network>
    <id>urn:storageos:Network:3a42b101-d309-4a92-
b29f-16215aa09c0f:vdc1</id>
    <link rel="self" href="/vdc/networks/
urn:storageos:Network:3a42b101-d309-4a92-b29f-16215aa09c0f:vdc1"/>
    <name>FABRIC_Vplex_LGL6220_FID_30-10:00:00:27:f8:58:f6:c1</name>
  </network>
  <network>
    <id>urn:storageos:Network:79596cc5-e8b2-4033-
b576-8282222b7eb0:vdc1</id>
    <link rel="self" href="/vdc/networks/
urn:storageos:Network:79596cc5-e8b2-4033-b576-8282222b7eb0:vdc1"/>
    <name>FABRIC_Vplex_Meta_Fid_20-10:00:00:27:f8:58:f6:c3</
name>
  </network>
  <network>
    <id>urn:storageos:Network:80d3e51d-7181-4b31-
baad-9f7d66bbd55e:vdc1</id>
    <link rel="self" href="/vdc/networks/
urn:storageos:Network:80d3e51d-7181-4b31-baad-9f7d66bbd55e:vdc1"/>
    <name>FABRIC_Vplex_WAN-10:00:00:27:f8:58:f6:bc</name>
  </network>
  <network>
    <id>urn:storageos:Network:1da5b653-dce0-4f1c-8ac6-
c0ad4faa176a:vdc1</id>
    <link rel="self" href="/vdc/networks/
urn:storageos:Network:1da5b653-dce0-4f1c-8ac6-c0ad4faa176a:vdc1"/>
    <name>FABRIC_vplex154nbr2</name>
  </network>
</networks>
```



```

        <id>urn:storageos:Network:454b5f84-403d-4d1b-bce7-
c920b93fc9d4:vdcl</id>
        <link rel="self" href="/vdc/networks/
urn:storageos:Network:454b5f84-403d-4d1b-bce7-c920b93fc9d4:vdcl"/>
        <name>FABRIC_VPlex_LGL6221_FID_40</name>
    </network>
    <network>
        <id>urn:storageos:Network:b53c8e06-0e80-457d-9764-
c2bb82437de9:vdcl</id>
        <link rel="self" href="/vdc/networks/
urn:storageos:Network:b53c8e06-0e80-457d-9764-c2bb82437de9:vdcl"/>
        <name>FABRIC_losam082-fabric</name>
    </network>
    <network>
        <id>urn:storageos:Network:3a5e76fe-2541-48ca-a293-
bdc6ead8fa84:vdcl</id>
        <link rel="self" href="/vdc/networks/
urn:storageos:Network:3a5e76fe-2541-48ca-a293-bdc6ead8fa84:vdcl"/>
        <name>FABRIC_VPLEX_WAN_lglw6208/lglw6209</name>
    </network>
</networks>

```

2. Select the network in which you want to swap the initiator. You can also use GET /vdc/networks/{id} if you want to view more details about a particular network.
3. Add the new initiator to the network using PUT /vdc/networks/{id}. The updated network representation is returned with the new endpoint.

Note

The manually added endpoint is listed in the <endpoints_discovered> section, however the value is false, whereas those endpoints that ViPR Controller discovered have a value of true.

Request

```

PUT https://<ViPR_Controller_VIP>:4443/vdc/networks/
urn:storageos:Network:454b5f84-403d-4d1b-bce7-c920b93fc9d4:vdcl
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<network_update>
  <endpoint_changes>
    <add>
      <endpoint>50:00:14:42:90:71:23:02</endpoint>
    </add>
  </endpoint_changes>
</network_udpate>

```

Response

```

HTTP 200 OK
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<network>
  <creation_time>1423688165153</creation_time>
  <global>false</global>
  <id>urn:storageos:Network:454b5f84-403d-4d1b-bce7-
c920b93fc9d4:vdcl</id>
  <inactive>false</inactive>
  <internal>false</internal>

```

```

<link rel="self" href="/vdc/networks/urn:storageos:Network:
454b5f84-403d-4dlb-bce7-c920b93fc9d4:vdc1"/>
<name>FABRIC_VPlex_LGL6221_FID_40</name>
<remote>>false</remote>
<tags/>
<vdc>
  <id>urn:storageos:VirtualDataCenter:99c5c185-
ebb0-468a-8fe7-3ea6e65b7b57:vdc1</id>
  <link rel="self" href="/vdc/
urn:storageos:VirtualDataCenter:99c5c185-
ebb0-468a-8fe7-3ea6e65b7b57:vdc1"/>
</vdc>
<native_guid>FC+BROCADE+10:00:00:27:F8:58:F6:C2</native_guid>
<discovered>>true</discovered>
<endpoints>
  <endpoint>50:01:24:81:00:6A:6D:BE</endpoint>
  <endpoint>10:00:00:90:FA:66:22:E3</endpoint>
  <endpoint>50:06:01:62:39:A0:30:98</endpoint>
  <endpoint>50:01:24:81:00:71:ED:28</endpoint>
  <endpoint>50:01:24:81:00:71:ED:29</endpoint>
  <endpoint>10:00:00:90:FA:1C:A3:B8</endpoint>
  <endpoint>10:00:00:90:FA:1C:A3:B9</endpoint>
  <endpoint>50:00:14:42:90:71:23:02</endpoint>
  <endpoint>50:01:24:81:00:6A:6D:BF</endpoint>
  <endpoint>21:00:00:1B:32:8A:37:B3</endpoint>
  <endpoint>50:00:14:42:90:71:23:00</endpoint>
</endpoints>
<endpoints_discovered>
  <endpoint_discovered>
    <name>50:01:24:81:00:6A:6D:BE</name>
    <value>>true</value>
  </endpoint_discovered>
  <endpoint_discovered>
    <name>10:00:00:90:FA:66:22:E3</name>
    <value>>true</value>
  </endpoint_discovered>
  <endpoint_discovered>
    <name>50:06:01:62:39:A0:30:98</name>
    <value>>true</value>
  </endpoint_discovered>
  <endpoint_discovered>
    <name>50:01:24:81:00:71:ED:28</name>
    <value>>true</value>
  </endpoint_discovered>
  <endpoint_discovered>
    <name>50:01:24:81:00:71:ED:29</name>
    <value>>true</value>
  </endpoint_discovered>
  <endpoint_discovered>
    <name>10:00:00:90:FA:1C:A3:B8</name>
    <value>>true</value>
  </endpoint_discovered>
  <endpoint_discovered>
    <name>10:00:00:90:FA:1C:A3:B9</name>
    <value>>true</value>
  </endpoint_discovered>
  <endpoint_discovered>
    <name>50:00:14:42:90:71:23:02</name>
    <value>>false</value>
  </endpoint_discovered>
  <endpoint_discovered>
    <name>50:01:24:81:00:6A:6D:BF</name>
    <value>>true</value>
  </endpoint_discovered>
  <endpoint_discovered>
    <name>21:00:00:1B:32:8A:37:B3</name>
    <value>>false</value>
  </endpoint_discovered>
  <endpoint_discovered>

```

```

        <name>50:00:14:42:90:71:23:00</name>
        <value>>true</value>
    </endpoint_discovered>
</endpoints_discovered>
<fabric_id>Vplex_LGL6221_FID_40</fabric_id>
<network_systems>
    <network_system>urn:storageos:NetworkSystem:
7875ed0e-1d2a-430d-94d0-20fcd7e38bb4:vdc1</network_system>
</network_systems>
    <registration_status>REGISTERED</registration_status>
    <transport_type>FC</transport_type>
</network>

```

4. For the initiator that you just added as an endpoint in the network, create an initiator on the required host in ViPR Controller.

Request

```

POST https://<ViPR_Controller_VIP>:4443/compute/hosts/{host_id}/
initiators
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<initiator_create>
    <protocol>FC</protocol>
    <initiator_port>50:00:14:42:90:71:23:02</initiator_port>
</initiator_create>

```

Response

```

HTTP 200 OK
Content-Type: application/xml

<initiator>
    <creation_time>1379202603661</creation_time>
    <id>urn:storageos:Initiator:07b2e71d-
cb4c-49c9-94fe-1feab7878d35:vdc1</id>
    <inactive>false</inactive>
    <link
        href="/compute/initiators/urn:storageos:Initiator:
07b2e71d-cb4c-49c9-94fe-1feab7878d35:vdc1"
        rel="self"/>
    <tags/>
    <host>
        <id>urn:storageos:Host:c2fe902f-0de0-4a92-
a2a9-46ba87279bd2:vdc1</id>
        <link
            href="/compute/hosts/
urn:storageos:Host:c2fe902f-0de0-4a92-a2a9-46ba87279bd2:vdc1"
            rel="self"/>
        </host>
        <protocol>FC</protocol>
        <hostname>myhost</hostname>
        <initiator_port>50:00:14:42:90:71:23:02</initiator_port>
    </initiator>

```

The initiator is automatically registered, and any exports are updated automatically for the host.

5. Remove the host initiator, which is being replaced by the host initiator added in the previous steps, from the network in ViPR Controller.

Note

You can only remove those initiators which you manually added to ViPR Controller. Any initiators that ViPR Controller discovered cannot be removed.

Request

```
PUT https://<ViPR_Controller_VIP>:4443/vdc/networks/
urn:storageos:Network:454b5f84-403d-4d1b-bce7-c920b93fc9d4:vdc1
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<network_update>
  <endpoint_changes>
    <remove>
      <endpoint>21:00:00:1B:32:8A:37:B3</endpoint>
    </remove>
  </endpoint_changes>
</network_update>
```

Response

```
HTTP 200 OK
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<network>
  <creation_time>1423688165153</creation_time>
  <global>>false</global>
  <id>urn:storageos:Network:454b5f84-403d-4d1b-bce7-
c920b93fc9d4:vdc1</id>
  <inactive>>false</inactive>
  <internal>>false</internal>
  <link rel="self" href="/vdc/networks/urn:storageos:Network:
454b5f84-403d-4d1b-bce7-c920b93fc9d4:vdc1"/>
  <name>FABRIC_VPlex_LGL6221_FID_40</name>
  <remote>>false</remote>
  <tags/>
  <vdc>
    <id>urn:storageos:VirtualDataCenter:99c5c185-
ebb0-468a-8fe7-3ea6e65b7b57:vdc1</id>
    <link rel="self" href="/vdc/
urn:storageos:VirtualDataCenter:99c5c185-
ebb0-468a-8fe7-3ea6e65b7b57:vdc1"/>
  </vdc>
  <native_guid>FC+BROCADE+10:00:00:27:F8:58:F6:C2</native_guid>
  <discovered>>true</discovered>
  <endpoints>
    <endpoint>50:01:24:81:00:6A:6D:BE</endpoint>
    <endpoint>10:00:00:90:FA:66:22:E3</endpoint>
    <endpoint>50:06:01:62:39:A0:30:98</endpoint>
    <endpoint>50:01:24:81:00:71:ED:28</endpoint>
    <endpoint>50:01:24:81:00:71:ED:29</endpoint>
    <endpoint>10:00:00:90:FA:1C:A3:B8</endpoint>
    <endpoint>10:00:00:90:FA:1C:A3:B9</endpoint>
    <endpoint>50:00:14:42:90:71:23:02</endpoint>
    <endpoint>50:01:24:81:00:6A:6D:BF</endpoint>
    <endpoint>50:00:14:42:90:71:23:00</endpoint>
  </endpoints>
  <endpoints_discovered>
    <endpoint_discovered>
      <name>50:01:24:81:00:6A:6D:BE</name>
      <value>>true</value>
    </endpoint_discovered>
    <endpoint_discovered>
      <name>10:00:00:90:FA:66:22:E3</name>
```

```

        <value>true</value>
      </endpoint_discovered>
    <endpoint_discovered>
      <name>50:06:01:62:39:A0:30:98</name>
      <value>true</value>
    </endpoint_discovered>
  <endpoint_discovered>
    <name>50:01:24:81:00:71:ED:28</name>
    <value>true</value>
  </endpoint_discovered>
  <endpoint_discovered>
    <name>50:01:24:81:00:71:ED:29</name>
    <value>true</value>
  </endpoint_discovered>
  <endpoint_discovered>
    <name>10:00:00:90:FA:1C:A3:B8</name>
    <value>true</value>
  </endpoint_discovered>
  <endpoint_discovered>
    <name>10:00:00:90:FA:1C:A3:B9</name>
    <value>true</value>
  </endpoint_discovered>
  <endpoint_discovered>
    <name>50:00:14:42:90:71:23:02</name>
    <value>false</value>
  </endpoint_discovered>
  <endpoint_discovered>
    <name>50:01:24:81:00:6A:6D:BF</name>
    <value>true</value>
  </endpoint_discovered>
  <endpoint_discovered>
    <name>50:00:14:42:90:71:23:00</name>
    <value>true</value>
  </endpoint_discovered>
</endpoints_discovered>
<fabric_id>VPLex_LGL6221_FID_40</fabric_id>
<network_systems>
  <network_system>urn:storageos:NetworkSystem:
7875ed0e-1d2a-430d-94d0-20fcd7e38bb4:vdcl</network_system>
</network_systems>
<registration_status>REGISTERED</registration_status>
<transport_type>FC</transport_type>
</network>

```

6. Delete the initiator from ViPR Controller using POST `/compute/initiators/{id}/deactivate`. This will automatically remove the initiator from any existing exports. A task is returned whose URI can be queried to determine the status of the task.

Request

```

POST https://<ViPR_Controller_VIP>:4443/compute/initiators/
urn:storageos:Initiator:b90c23cf-8152-4dcc-b3e0-a1d882f9a79e:vdcl/
deactivate
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

```

Response

```

HTTP 202 Accepted
Content-Type: application/xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<task>
  <creation_time>1424203404060</creation_time>
  <global>false</global>
  <id>urn:storageos:Task:864cfa1b-8a51-4175-

```

```

b3d3-7d4a5131c33d:vdcl</id>
  <inactive>>false</inactive>
  <internal>>false</internal>
  <link rel="self" href="/vdc/tasks/urn:storageos:Task:
864cfalb-8a51-4175-b3d3-7d4a5131c33d:vdcl"/>
  <name>DELETE INITIATOR</name>
  <remote>>false</remote>
  <tags/>
  <vdc>
    <id>urn:storageos:VirtualDataCenter:fd8f9b83-69c4-4826-
b83b-49944710bd28:vdcl</id>
    <link rel="self" href="/vdc/
urn:storageos:VirtualDataCenter:fd8f9b83-69c4-4826-
b83b-49944710bd28:vdcl"/>
  </vdc>
  <associated_resources/>
  <description>delete initiator</description>
  <op_id>928dd2ed-3db4-4e91-98fc-91b34e78237a</op_id>
  <progress>0</progress>
  <resource>
    <id>urn:storageos:Initiator:b90c23cf-8152-4dcc-b3e0-
ald882f9a79e:vdcl</id>
    <link rel="self" href="/compute/initiators/
urn:storageos:Initiator:b90c23cf-8152-4dcc-b3e0-
ald882f9a79e:vdcl"/>
    <name/>
  </resource>
  <start_time>1424203404059</start_time>
  <state>pending</state>
</task>

```

7. Query the initiator deletion task, using the task ID from the response body of the POST request. When the message attribute of the task is Operation completed successfully, the operation has completed and the initiator has been deleted. In addition, if the initiator is in use by exports, those exports are updated by removing the initiator from them.

Request

```

GET https://<ViPR_Controller_VIP>:4443/vdc/tasks/
urn:storageos:Task:864cfalb-8a51-4175-b3d3-7d4a5131c33d:vdcl
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

```

Response

```

HTTP 200 OK
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<task>
  <creation_time>1424203404060</creation_time>
  <global>>false</global>
  <id>urn:storageos:Task:864cfalb-8a51-4175-
b3d3-7d4a5131c33d:vdcl</id>
  <inactive>>false</inactive>
  <internal>>false</internal>
  <link rel="self" href="/vdc/tasks/urn:storageos:Task:
864cfalb-8a51-4175-b3d3-7d4a5131c33d:vdcl"/>
  <name>DELETE INITIATOR</name>
  <remote>>false</remote>
  <tags/>
  <vdc>
    <id>urn:storageos:VirtualDataCenter:fd8f9b83-69c4-4826-
b83b-49944710bd28:vdcl</id>
    <link rel="self" href="/vdc/

```

```

urn:storageos:VirtualDataCenter:fd8f9b83-69c4-4826-
b83b-49944710bd28:vdcl"/>
  </vdc>
  <associated_resources/>
  <description>delete initiator</description>
  <end_time>1424203467146</end_time>
  <message>Operation completed successfully</message>
  <op_id>928dd2ed-3db4-4e91-98fc-91b34e78237a</op_id>
  <progress>100</progress>
  <resource>
    <id>urn:storageos:Initiator:b90c23cf-8152-4dcc-b3e0-
ald882f9a79e:vdcl</id>
    <link rel="self" href="/compute/initiators/
urn:storageos:Initiator:b90c23cf-8152-4dcc-b3e0-
ald882f9a79e:vdcl"/>
    <name/>
  </resource>
  <start_time>1424203404059</start_time>
  <state>ready</state>
  <workflow>
    <id>urn:storageos:Workflow:cd0a4b0f-2adb-458f-a34e-
a4331790f7ba:vdcl</id>
    <link rel="self" href="/vdc/workflows/
urn:storageos:Workflow:cd0a4b0f-2adb-458f-a34e-a4331790f7ba:vdcl"/>
  </workflow>
</task>

```

Important REST API calls to manage hosts and clusters

The table shows some important APIs that are used to manage hosts, vCenters and Windows clusters.

Table 7 ViPR Controller REST API Calls to Manage Hosts and Clusters

ViPR REST API Call	Description
POST /tenant/ {tenant_urn}/hosts	Create a host resource in ViPR Controller.
PUT /compute/hosts/ {Host_URN}	Modify parameters for a host. In addition, it updates any exports affected by adding or removing the host to or from a cluster. Discovery is then run for the host.
POST /compute/hosts/ {Host_URN}/discover	Add a host to the discovery queue. Host discovery is performed asynchronously.
POST /compute/hosts/ (Host_URN)/detach- storage	Detaches storage from the host. Updates export groups and fileshare exports by removing references to the given host, the host's initiators, and the host's IP interfaces. Volumes and fileshares are left intact.
GET /compute/ clusters/ {Cluster_URN}/hosts	Lists the hosts in a cluster.
GET /tenants/ {Tenant_URN}/clusters	List all clusters that belong to the specified tenant. Clusters, like hosts and vCenters, are tenant-level resources.
GET /compute/ clusters/{Cluster_URN}	Retrieve a cluster resource.
POST /tenants/{id}/ clusters	Build a ViPR Controller cluster.

Table 7 ViPR Controller REST API Calls to Manage Hosts and Clusters (continued)

ViPR REST API Call	Description
PUT /compute/hosts/{Host_URN}	<p>Add a host to a cluster or remove a host from a cluster. The payload for this call allows you to specify the URN of a cluster. When removing a host from a cluster, the cluster would be specified without any value, such as <cluster></cluster>.</p> <pre data-bbox="727 489 1465 678"> <host_update> <type>Windows</type> <host_name>myhost.corp.com</host_name> <name>myHost</name> <user_name>admin</user_name> <password>password</password> <cluster>{Cluster_URN}</cluster> </host_update> </pre>
POST /compute/clusters/{Cluster_URN}/detach-storage	<p>Detaches storage from the cluster. Updates export groups that are referenced by the given cluster by removing the cluster reference and initiators belonging to hosts in this cluster. Volumes are left intact.</p>
POST /compute/hosts/{Host_URN}/initiators	<p>Register an initiator for the host and update the host export groups by adding the initiator to them. The payload for this call allows you to specify the initiator.</p> <pre data-bbox="727 982 1465 1098"> <initiator_create> <protocol>iSCSI</protocol> <initiator_port>iqn.2001-10.com.emc.lss.host1</initiator_port> </initiator_create> </pre>

CHAPTER 8

Adding vCenters and ESX/ESXi clusters

This chapter contains the following topics:

- [Overview](#)..... 82
- [Add and discover a vCenter Server](#)..... 82
- [Network configuration requirements for vCenters](#)..... 84
- [Important REST API calls to manage vCenters and clusters](#)..... 85

Overview

This chapter provides ViPR Controller System and Tenant Administrators the ViPR Controller support requirements, and prerequisite information to prepare vCenters for ViPR Controller integration, the steps to add the host to ViPR Controller, and the important REST API calls used to manage vCenter and ESX/ESXi clusters.

Add and discover a vCenter Server

You add a vCenter Server to ViPR Controller using `POST /tenants/{tenant_id}/vcenters`. ViPR Controller automatically discovers the host, clusters and other resources managed by that server.

Before you begin

To complete this procedure:

- For pre-configuration and support requirements, see *ViPR Controller Virtual Data Center Requirements and Information Guide* on the [ViPR Controller Product Documentation Index](#).
- Authenticate with the ViPR REST API as a Tenant Administrator. See [Authenticating with the REST API on page 9](#).
- The *EMC ViPR Controller REST API Reference* on the [ViPR Controller Product Documentation Index](#) provides a description and complete list of parameters for the REST API methods used in this article.

Procedure

1. Get the URN of your tenant.

```
GET https://<ViPR_Controller_VIP>:4443/tenant
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>
```

2. Use the tenant URN to create a vCenter.

Request

```
POST https://<ViPR_Controller_VIP>:4443/tenants/{Tenant_URN}/
vcenters
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<vcenter_create>
  <ip_address>192.168.0.0</ip_address>
  <name>vcenter_west</name>
  <port_number>443</port_number>
  <user_name>admin_user</user_name>
  <password>Password1</password>
</vcenter_create>
```

Response

```
HTTP 202 Accepted
Content-Type: application/xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<task>
  <creation_time>1423595062584</creation_time>
```

```

    <global>false</global>
    <id>urn:storageos:Task:e2a06ffe-1a96-4f54-9463-
f426fc4083d8:vdcl</id>
    <inactive>false</inactive>
    <internal>false</internal>
    <link rel="self" href="/vdc/tasks/
urn:storageos:Task:e2a06ffe-1a96-4f54-9463-
f426fc4083d8:vdcl"/>
    <name>DISCOVER VCENTER</name>
    <remote>false</remote>
    <tags/>
    <vdc>
        <id>urn:storageos:VirtualDataCenter:99c5c185-
ebb0-468a-8fe7-3ea6e65b7b57:vdcl</id>
        <link rel="self" href="/vdc/
urn:storageos:VirtualDataCenter:99c5c185-
ebb0-468a-8fe7-3ea6e65b7b57:vdcl"/>
    </vdc>
    <associated_resources/>
    <description>discover a vcenter</description>
    <op_id>c3eefcb2-5a13-4446-b31f-76a172dc7701</op_id>
    <progress>0</progress>
    <resource>
        <id>urn:storageos:Vcenter:316465e4-8661-4d73-a60d-
c2132f0ff868:vdcl</id>
        <link rel="self" href="/compute/vcenters/
urn:storageos:Vcenter:316465e4-8661-4d73-a60d-c2132f0ff868:vdcl"/>
        <name>vcenter_west</name>
    </resource>
    <start_time>1423595062583</start_time>
    <state>pending</state>
    <tenant>
        <id>urn:storageos:TenantOrg:c104b49e-4e6b-449c-b92b-
a19b247a613b:global</id>
        <link rel="self" href="/tenants/
urn:storageos:TenantOrg:c104b49e-4e6b-449c-b92b-
a19b247a613b:global"/>
    </tenant>
</task>

```

3. Query the vCenter create task, using the task URL from the response body of the POST request, until the message attribute of the task is `Operation completed successfully` which indicates that the operation has completed and the vCenter has been added to ViPR Controller and successfully discovered.

Request

```

GET https://<ViPR_Controller_VIP>:4443/vdc/tasks/
urn:storageos:Task:e2a06ffe-1a96-4f54-9463-f426fc4083d8:vdcl
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

```

Response

```

HTTP 200 OK
Content-Type: application/xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<task>
    <creation_time>1423595062584</creation_time>
    <global>false</global>
    <id>urn:storageos:Task:e2a06ffe-1a96-4f54-9463-
f426fc4083d8:vdcl</id>
    <inactive>false</inactive>
    <internal>false</internal>
    <link rel="self" href="/vdc/tasks/
urn:storageos:Task:e2a06ffe-1a96-4f54-9463-f426fc4083d8:vdcl"/>

```

```

    <name>DISCOVER VCENTER</name>
    <remote>>false</remote>
    <tags/>
    <vdc>
      <id>urn:storageos:VirtualDataCenter:99c5c185-
ebb0-468a-8fe7-3ea6e65b7b57:vdc1</id>
      <link rel="self" href="/vdc/
urn:storageos:VirtualDataCenter:99c5c185-
ebb0-468a-8fe7-3ea6e65b7b57:vdc1"/>
    </vdc>
    <associated_resources/>
    <description>discover a vcenter</description>
    <end_time>1423595086871</end_time>
    <message>Operation completed successfully</message>
    <op_id>c3eefcb2-5a13-4446-b31f-76a172dc7701</op_id>
    <progress>100</progress>
    <resource>
      <id>urn:storageos:Vcenter:316465e4-8661-4d73-a60d-
c2132f0ff868:vdc1</id>
      <link rel="self" href="/compute/vcenters/
urn:storageos:Vcenter:316465e4-8661-4d73-a60d-c2132f0ff868:vdc1"/>
      <name>vcenter_west</name>
    </resource>
    <start_time>1423595062583</start_time>
    <state>ready</state>
    <tenant>
      <id>urn:storageos:TenantOrg:c104b49e-4e6b-449c-b92b-
a19b247a613b:global</id>
      <link rel="self" href="/tenants/
urn:storageos:TenantOrg:c104b49e-4e6b-449c-b92b-
a19b247a613b:global"/>
    </tenant>
  </task>

```

Network configuration requirements for vCenters

After a host is added to ViPR, configure the networks if required before using the host in a service.

- Fibre Channel
 - No action is required when a vCenter is added on a Fibre Channel network. The host initiators for ESX/ESXi hosts on Fibre Channel networks are automatically discovered and registered in ViPR when vCenter is added to ViPR. At the time the fabric switch is added to the ViPR, using `POST /vdc/network-systems`, ViPR also discovers the storage systems on the same network. During provisioning ViPR selects the storage and host ports that will be used to connect the hosts and storage.
 - Optionally, the resources used for provisioning can be customized by deregistering the host initiators that can be used for provisioning, or specifying which host and storage ports can be used when provisioning storage to the ESX/ESXi hosts.
- IP

Add the ESX/ESXi ports to an IP network.

ViPR can discover the ports of IP connected storage systems and hosts, but it cannot discover the paths between them, so it is necessary to create IP networks, and then add the host, and storage system ports, which will be provisioned together, to the same IP network.

If creating a network for a virtual array that will be used for file system exports to an ESXi cluster, add all ESXi server IP interface addresses (Management IP, vMotion IPs, and any other IP VMNIC visible in vCenter) per cluster.

- iSCSI
Hosts that use ViPR Controller services with the iSCSI protocol must have their iSCSI ports logged into the correct target array ports before they can be used in the service.

Important REST API calls to manage vCenters and clusters

The table shows some important APIs that are used to manage vCenters and clusters.

Table 8 ViPR Controller REST API Calls to Manage vCenters and Clusters

ViPR REST API Call	Description
POST /tenants/{Tenant_URN}/vcenters	Add a vCenter resource to the tenant organization.
GET /compute/vcenters/{Vcenter_URN}/clusters	List the clusters in a vCenter.
GET /compute/vcenters/{Vcenter_URN}/hosts	List the hosts in a vCenter.
PUT /compute/vcenters/{Vcenter_URN}	Update a vCenter. The vCenter attributes to be updated are included in the request payload.
POST /compute/vcenters/{vcenter_URN}/discover	Discover a vCenter.
GET /tenants/{Tenant_URN}/vCenters	List all vCenters that belong to the specified tenant.
POST /tenants/{Tenant_URN}/clusters	Build a ViPR Controller cluster.

CHAPTER 9

Configuring and adding Vblock System Components

This chapter contains the following topics:

- [Configure and add Vblock system components](#) 88
- [Add a Vblock compute system to ViPR Controller](#) 88
- [Add Compute Image Servers](#) 92
- [Add Compute Images to ViPR Controller](#) 94

Configure and add Vblock system components

You must add each Vblock system component to ViPR Controller as an individual physical asset. Once all of the physical assets of the Vblock system have been added to ViPR Controller, you can set up storage visibility using the ViPR Controller virtual arrays. After configuring virtual arrays, you can configure ViPR Controller compute virtual pools.

Before adding the Vblock system components to ViPR Controller, review the Vblock system requirements and information described in the *ViPR Controller Virtual Data Center Requirements and Information Guide* which is available from the [ViPR Controller Product Documentation Index](#).

At a minimum, the following Vblock components must be added to the ViPR Controller physical assets, to use ViPR Controller to perform bare metal provisioning on the Vblock compute systems during a provisioning operation.

- Vblock compute system (UCS)
See: [Add a Vblock compute system to ViPR Controller on page 88](#)
- Vblock storage system
See the section of this guide which provides the steps to configure and add the type of storage system which is configured in your Vblock system.
- Vblock Fabric managers (Cisco MDS)
See: [Adding network systems \(fabric managers\) and SAN networks on page 45](#)

Add and configure components for OS Installation

In addition to the components listed above, you will need to add at least one compute image server, and one compute image to the ViPR Controller physical assets, and associate each Vblock compute system with a compute image server, to use the ViPR Controller to install an operating system on the Vblock compute systems during a Vblock system provisioning service operation.

Refer to the following sections for steps to add the compute image server, and compute image to the ViPR Controller physical assets, and to associate the Vblock compute system with a compute image server.

- See: [Add a Vblock compute system to ViPR Controller on page 88](#)
- [Add Compute Image Servers to ViPR Controller on page 92](#)
- [Add Compute Images to ViPR Controller on page 94](#)

Add a Vblock compute system to ViPR Controller

You add a Vblock compute system to ViPR Controller using `POST /vdc/compute-systems`.

Before you begin

- Prior to performing this operation, review the Vblock system requirements and information in the *ViPR Controller Virtual Data Center Requirements and Information Guide*, which is available from the [ViPR Controller Product Documentation Index](#).
- Authenticate with the ViPR REST API as a System Administrator. See [Authenticating with the REST API on page 9](#).
- You will need to provide the following information:

ip_address

IP address of the compute system.

port_number

The port for ViPR Controller to connect with the compute system. The default is 443.

user_name and password

The user credentials for ViPR Controller to log into UCSM.

system_type

ucs is one example

os_install_network

The OS Install Network is a private VLAN for operating system (OS) installation. The OS Install Network is used by ViPR Controller during provisioning for communication between the hosts and the ViPR Controller compute image server. Since ViPR Controller utilizes a PXE boot process, a DHCP server is used and must be isolated from the customer network. During provisioning, the compute blades communicate with the image server and the operating system installation is performed over the OS Install Network. Once the OS installation is complete for a given host, the OS Install Network is no longer used to communicate to that host.

compute_image_server

Compute image server identifier. You can get the list of compute image server identifiers using GET /compute/imageservers.

Procedure

1. Add a Vblock compute system to ViPR Controller by sending a POST /vdc/compute-systems request. The request returns a task whose URI can be queried to determine the status of the task.

Request

```
POST https://<ViPR_Controller_VIP>:4443/vdc/compute-systems
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<compute_system_create>
  <name>compute_west_1</name>
  <ip_address>192.168.0.0</ip_address>
  <port_number>443</port_number>
  <use_ssl>true</use_ssl>
  <user_name>admin</user_name>
  <password>Password1</password>
  <system_type>ucs</system_type>
  <os_install_network>630</os_install_network>
  <compute_image_server>
    urn:storageos:ComputeImageServer:45882d87-1a85-4bde-b658-
    c36f6377bc1d:vdcl</compute_image_server>
</compute_system_create>
```

Response

```
HTTP 202 Accepted
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<task>
  <creation_time>1421861349322</creation_time>
  <global>false</global>
  <id>urn:storageos:Task:
4434fb76-4f6f-4525-996c-2659ffb2624b:vdcl</id>
  <inactive>false</inactive>
  <internal>false</internal>
```

```

<link rel="self" href="/vdc/tasks/urn:storageos:Task:
4434fb76-4f6f-4525-996c-2659ffb2624b:vdc1"/>
<name>DISCOVER COMPUTE SYSTEM</name>
<remote>>false</remote>
<tags/>
<vdc>
  <id>urn:storageos:VirtualDataCenter:99c5c185-
ebb0-468a-8fe7-3ea6e65b7b57:vdc1</id>
  <link rel="self" href="/vdc/
urn:storageos:VirtualDataCenter:99c5c185-
ebb0-468a-8fe7-3ea6e65b7b57:vdc1"/>
</vdc>
<associated_resources/>
<description>discover a compute system.</description>
<op_id>4b661145-702d-417e-951d-cec23e062d26</op_id>
<progress>0</progress>
<resource>
  <id>urn:storageos:ComputeSystem:a9e3a23e-1020-40ca-b3d6-
e5dc9b80c906:vdc1</id>
  <link rel="self" href="/vdc/compute-systems/
urn:storageos:ComputeSystem:a9e3a23e-1020-40ca-b3d6-
e5dc9b80c906:vdc1"/>
  <name>compute_west_1</name>
</resource>
<start_time>1421861349321</start_time>
<state>pending</state>
</task>

```

2. Query the Vblock compute system create task, using the task URL from the response body of the POST request. When the message attribute of the task is `Operation completed successfully`, the operation has completed and the Vblock compute system has been successfully discovered.

Request

```

GET https://<ViPR_Controller_VIP>:4443/vdc/tasks/
urn:storageos:Task:4434fb76-4f6f-4525-996c-2659ffb2624b:vdc1
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

```

Response

```

HTTP 200 OK
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<task>
  <creation_time>1421861349322</creation_time>
  <global>>false</global>
  <id>urn:storageos:Task:
4434fb76-4f6f-4525-996c-2659ffb2624b:vdc1</id>
  <inactive>>false</inactive>
  <internal>>false</internal>
  <link rel="self" href="/vdc/tasks/urn:storageos:Task:
4434fb76-4f6f-4525-996c-2659ffb2624b:vdc1"/>
  <name>DISCOVER COMPUTE SYSTEM</name>
  <remote>>false</remote>
  <tags/>
  <vdc>
    <id>urn:storageos:VirtualDataCenter:99c5c185-
ebb0-468a-8fe7-3ea6e65b7b57:vdc1</id>
    <link rel="self" href="/vdc/
urn:storageos:VirtualDataCenter:99c5c185-
ebb0-468a-8fe7-3ea6e65b7b57:vdc1"/>
  </vdc>
  <associated_resources/>

```

```

<description>discover a compute system.</description>
<end_time>1421861354723</end_time>
<message>Operation completed successfully</message>
<op_id>4b661145-702d-417e-951d-cec23e062d26</op_id>
<progress>100</progress>
<resource>
  <id>urn:storageos:ComputeSystem:a9e3a23e-1020-40ca-b3d6-
e5dc9b80c906:vdcl</id>
  <link rel="self" href="/vdc/compute-systems/
urn:storageos:ComputeSystem:a9e3a23e-1020-40ca-b3d6-
e5dc9b80c906:vdcl"/>
  <name>compute_west_1</name>
</resource>
<start_time>1421861349321</start_time>
<state>ready</state>
</task>

```

Additional REST API calls to manage your Vblock Compute Systems

The table shows some additional APIs that are used to manage your Vblock Compute Systems

Table 9 REST API calls to manage your Vblock Compute Systems

REST API Call	Description
GET /vdc/compute-systems/{ID}	Get the details of a Vblock compute system.
GET /vdc/compute-systems	Get a list of the names, IDs, and links of all Vblock Compute Systems.
PUT /vdc/compute-systems/{ID}	Update a Vblock compute system and rediscover it.
POST /vdc/compute-systems/{ID}/deactivate	Delete a Vblock compute system.
GET /vdc/compute-systems/{ID}/compute-elements	Gets all of the compute elements belonging to a Vblock compute system in ViPR Controller.
GET /vdc/compute-systems/search?parameter={search_parameter}	Search for a Vblock compute system. For example: GET /vdc/compute-systems/search?name=vblock_east
POST /vdc/compute-systems/{ID}/register	Registers a previously de-registered Vblock compute system. Note When you use POST /vdc/compute-systems to create a Vblock compute system, it is automatically registered by ViPR Controller.
POST /vdc/compute-systems/{ID}/deregister	De-register a Vblock compute system so it is not used.
POST /vdc/compute-elements/{ID}/deregister	De-register an individual compute element in the Vblock compute system.

Add Compute Image Servers

You can add a compute image server to ViPR Controller using `Post /vdc/compute/imageservers`

Before you begin

- Prior to performing this operation, review the Vblock system requirements and information in the *ViPR Controller Virtual Data Center Requirements and Information Guide* on the [ViPR Controller Product Documentation Index](#).
- Authenticate with the ViPR REST API as a System Administrator. See [Authenticating with the REST API on page 9](#).
- You will need to provide the following information:

imageserver_ip

FQDN or IP address of the compute image server.

imageserver_secondip

The address of the OS installation network. The OS Install Network is a private VLAN for operating system (OS) installation. The OS Install Network is used by ViPR Controller during provisioning for communication between the hosts and the ViPR Controller compute image server. Since ViPR Controller utilizes a PXE boot process, a DHCP server is used and must be isolated from the customer network. During provisioning, the compute blades communicate with the image server and the operating system installation is performed over the OS Install Network. Once the OS installation is complete for a given host, the OS Install Network is no longer used to communicate to that host.

imageserver_user

Log in name of the user to access the compute image server from the ViPR Controller.

imageserver_password

User password to log in to the compute image server.

tftpBootDir

Path to TFTPBOOT directory on the compute image server. Default is `/opt/tftpboot/`.

osinstall_timeoutms

Timeout value for OS installation (in seconds). Default value is 3600.

Procedure

1. Add a compute image server to ViPR Controller by sending a `POST /vdc/compute/imageservers` request. The request returns a task whose URI can be queried to determine the status of the task.

Request

```
<compute_imageserver_create>
  <imageserver_ip>hostname.companyname.com</imageserver_ip>
  <imageserver_secondip>10.20.30.40</imageserver_secondip>
  <imageserver_user>SampleUser</imageserver_user>
  <imageserver_password>SamplePassword</imageserver_password>
  <tftpBootDir>/opt/tftpboot/</tftpBootDir>
  <osinstall_timeoutms>3600</osinstall_timeoutms>
</compute_imageserver_create>
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<task>
  <creation_time>1441795316852</creation_time>
  <global>>false</global>

  <id>urn:storageos:Task:e676e9d5-9bce-4f97-9dec-882d33163cb9:vdc1</id>
  <inactive>>false</inactive>
  <internal>>false</internal>
  <link rel="self" href="/vdc/tasks/urn:storageos:Task:e676e9d5-9bce-4f97-9dec-882d33163cb9:vdc1" />
  <name>CREATE AND VERIFY COMPUTE IMAGE SERVER</name>
  <remote>>false</remote>
  <tags />
  <vdc>
    <id>urn:storageos:VirtualDataCenter:e5cb40fd-b531-4e19-b7f2-9f8080405d3c:vdc1</id>
    <link rel="self" href="/vdc/urn:storageos:VirtualDataCenter:e5cb40fd-b531-4e19-b7f2-9f8080405d3c:vdc1" />
  </vdc>
  <associated_resources />
  <description>Create and verify a compute image server</description>
  <op_id>00fd9827-fff3-472e-8984-bf20106797af</op_id>
  <progress>0</progress>
  <resource>
    <id>urn:storageos:ComputeImageServer:2e554899-b51f-472d-9a79-9ad079171767:vdc1</id>
    <link rel="self" href="/compute/imageservers/urn:storageos:ComputeImageServer:2e554899-b51f-472d-9a79-9ad079171767:vdc1" />
    <name>lss.emc.com</name>
  </resource>
  <start_time>1441795316849</start_time>
  <state>pending</state>
</task>
```

2. Query the compute image server create and verify task, using the task URL from the response body of the POST request. When the message attribute of the task is Operation completed successfully, the operation has completed and the Vblock compute system has been successfully discovered.

Request

```
GET https://<ViPR_Controller_VIP>:4443//vdc/tasks/urn:storageos:Task:e676e9d5-9bce-4f97-9dec-882d33163cb9:vdc1
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<task>
  <creation_time>1441795316852</creation_time>
  <global>>false</global>

  <id>urn:storageos:Task:e676e9d5-9bce-4f97-9dec-882d33163cb9:vdc1</id>
  <inactive>>false</inactive>
  <internal>>false</internal>
  <link rel="self" href="/vdc/tasks/urn:storageos:Task:e676e9d5-9bce-4f97-9dec-882d33163cb9:vdc1" />
```

```

<name>CREATE AND VERIFY COMPUTE IMAGE SERVER</name>
<remote>>false</remote>
<tags />
<vdc>
  <id>urn:storageos:VirtualDataCenter:e5cb40fd-b531-4e19-
b7f2-9f8080405d3c:vdc1</id>
  <link rel="self" href="/vdc/
urn:storageos:VirtualDataCenter:e5cb40fd-b531-4e19-
b7f2-9f8080405d3c:vdc1" />
</vdc>
  <associated_resources />
  <description>Create and verify a compute image server</
description>
  <end_time>1450795316852</end_time>
  <message>Operation completed successfully</message>
  <op_id>00fd9827-fff3-472e-8984-bf20106797af</op_id>
  <progress>100</progress>
  <resource>
    <id>urn:storageos:ComputeImageServer:2e554899-
b51f-472d-9a79-9ad079171767:vdc1</id>
    <link rel="self" href="/compute/imageservers/
urn:storageos:ComputeImageServer:2e554899-
b51f-472d-9a79-9ad079171767:vdc1" />
    <name>lss.emc.com</name>
  </resource>
  <start_time>1441795316849</start_time>
  <state>Ready</state>
</task>

```

Additional API calls to manage your Compute Image Servers

The table shows some additional APIs that are used to manage your compute image servers.

Table 10 REST API calls to manage your compute image servers

REST API Call	Description
GET /vdc/ imageservers/ {ID}	Get the details of a compute image server, and the status of whether the compute image successfully uploaded to the compute image server or not. The compute images that were successfully uploaded are listed under <imageserverstatus>AVAILABLE<imageservicestatus>. Compute images, which were not successfully uploaded to the compute image server is listed under failed_compute_image.
GET /vdc/ imageservers	Get a list of the names, IDs, and links of all compute image servers.
PUT /vdc/ imageservers/ {ID}	Edit the compute image server properties.
POST /vdc/ imageservers/ {ID}/deactivate	Delete a compute image server Vblock compute system.

Add Compute Images to ViPR Controller

Compute Images are operating system (OS) installation files (ISO images) that ViPR Controller uses to deploy operating systems on Vblock compute elements that have been

registered to ViPR Controller. If ViPR Controller is used to provision ESX clusters, ViPR Controller can also be used to add the cluster to a vCenter datacenter that has been registered to ViPR Controller. Add a compute image to ViPR Controller using `POST /compute/images`.

Before you begin

Prior to performing this operation, review the Vblock system requirements and information in the *ViPR Controller Virtual Data Center Requirements and Information Guide* on the [ViPR Controller Product Documentation Index](#).

name

The installation file name that will be seen by ViPR Controller users when selecting the OS installation file to use for a service operation.

image_url

The URL location where the image file was added. If a user name and password are required to access the site, specify them in the URL.

- The ViPR Controller compute image server must have been added to ViPR Controller prior to adding the compute images to the ViPR Controller physical assets. For details see [Add Compute Image Servers to ViPR Controller on page 92](#).
- Authenticate with the ViPR REST API as a System Administrator. See [Authenticating with the REST API on page 9](#)
- For a description and complete list of parameters for the REST API methods used in this article, refer to [ViPR Controller REST API Reference](#).

Procedure

1. Add the compute image to ViPR Controller using `POST /compute/images`.

Request

```
POST https://<ViPR_Controller_VIP>:4443/compute/images
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<compute_image_create>
  <name>MyImage</name>
  <image_url>ftp://MyServer/myimage</image_url>
</compute_image_create>
```

Response

```
HTTP 202 Accepted
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<task>
  <creation_time>1421941720408</creation_time>
  <global>false</global>
  <id>urn:storageos:Task:2e97d85b-
e160-45d7-99cf-0b57386a7974:vdc1</id>
  <inactive>false</inactive>
  <internal>false</internal>
  <link rel="self" href="/vdc/tasks/urn:storageos:Task:2e97d85b-
e160-45d7-99cf-0b57386a7974:vdc1"/>
  <name>IMPORT IMAGE</name>
  <remote>false</remote>
  <tags/>
  <vdc>
    <id>urn:storageos:VirtualDataCenter:a2c45423-20a4-4d50-
a8cc-f81175a4ee75:vdc1</id>
```

```

        <link rel="self" href="/vdc/
urn:storageos:VirtualDataCenter:a2c45423-20a4-4d50-a8cc-
f81175a4ee75:vdcl"/></vdc>
        <associated_resources/>
        <description>import an image.</description>
        <op_id>1e9e4280-d40f-44cd-94c6-b6114169fd59</op_id>
        <progress>0</progress>
        <resource>
            <id>urn:storageos:ComputeImage:c3c45b49-20ad-4b19-
a4a0-5f72694cbcf:vdcl</id>
            <link rel="self" href="/compute/images/
urn:storageos:ComputeImage:c3c45b49-20ad-4b19-
a4a0-5f72694cbcf:vdcl"/>
            <name>MyImage</name>
        </resource>
        <start_time>1421941720407</start_time>
        <state>pending</state>
</task>

```

A task is returned whose URI can be queried to determine the status of the task.

2. Query the add compute image task, using the task ID from the response body of the POST request. When the message attribute of the task is Operation completed successfully, the operation has completed and the compute image has been added to ViPR Controller.

Request

```

https://<ViPR_Controller_VIP>:4443/vdc/tasks/urn:storageos:Task:
2e97d85b-e160-45d7-99cf-0b57386a7974:vdcl
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

```

Response

```

HTTP 200 OK
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<task>
    <creation_time>1421941720408</creation_time>
    <global>false</global>
    <id>urn:storageos:Task:2e97d85b-
e160-45d7-99cf-0b57386a7974:vdcl</id>
    <inactive>false</inactive>
    <internal>false</internal>
    <link rel="self" href="/vdc/tasks/urn:storageos:Task:2e97d85b-
e160-45d7-99cf-0b57386a7974:vdcl"/>
    <name>IMPORT IMAGE</name>
    <remote>false</remote>
    <tags/>
    <vdc>
        <id>urn:storageos:VirtualDataCenter:a2c45423-20a4-4d50-
a8cc-f81175a4ee75:vdcl</id>
        <link rel="self" href="/vdc/
urn:storageos:VirtualDataCenter:a2c45423-20a4-4d50-a8cc-
f81175a4ee75:vdcl"/></vdc>
        <associated_resources/>
        <description>import an image.</description>
        <end_time>1421945593699</end_time>
        <message>Operation completed successfully</message>
        <op_id>1e9e4280-d40f-44cd-94c6-b6114169fd59</op_id>
        <progress>100</progress>
        <resource>
            <id>urn:storageos:ComputeImage:c3c45b49-20ad-4b19-
a4a0-5f72694cbcf:vdcl</id>

```



```

        <link rel="self" href="/compute/images/
urn:storageos:ComputeImage:c3c45b49-20ad-4b19-
a4a0-5f72694cbcfef:vdcl"/>
        <name>MyImage</name>
    </resource>
    <start_time>1421941720407</start_time>
    <state>ready</state>
</task>

```

Additional REST API calls to manage your Compute Images

The table shows some additional APIs that are used to manage your Compute Images

Table 11 REST API calls to manage your Compute Images

REST API call	Description
GET /compute/images/{ID}	Get the details of a compute image, and the status of whether the compute image successfully uploaded to the compute image server or not. The compute images were successfully uploaded to the compute image servers listed under <code>available_image_servers</code> . Compute images were not successfully uploaded to the compute image servers listed under <code>failed_image_servers</code> .
GET /compute/images	Get a list of the names, IDs, and links of all compute images.
PUT /compute/images/{ID}	Update a compute image.
POST /compute/images/{ID}/deactivate	Delete a compute image.
GET /compute/images/search?parameter={search_parameter}	Search for a compute image . For example: GET /compute/images/search?name=myimage

CHAPTER 10

Creating and configuring a virtual array

This chapter contains the following topics:

- [Overview](#)..... 100
- [Create a virtual array and assign an ACL](#)..... 100
- [Add and configure the networks for block storage in the virtual array](#)..... 102
- [Add and configure the networks for file storage in a virtual array](#)..... 108
- [Optionally restrict access to storage pools by manually assigning them to a virtual array](#)..... 114
- [Virtual Array requirements for Vblock system services](#)..... 121

Overview

ViPR Controller system administrators can use the ViPR Controller REST API to create a virtual array, and assign an ACL to the virtual array. The REST API is also used to configure the networks, storage ports, and storage pools assigned to the virtual array.

For information on virtual arrays, see *ViPR Controller Concepts* on the [ViPR Controller Product Documentation Index](#).

Create a virtual array and assign an ACL

Before you begin

- Authenticate with the ViPR REST API as a System Administrator to create or edit virtual arrays. See [Authenticating with the REST API on page 9](#).
- For virtual array configuration requirements and recommendations, see *Configuration Considerations While Virtualizing Your Storage in ViPR Controller*.
- At a minimum, a virtual array must include at least one network, and one storage system connected to the network.
 - Storage systems can be added directly to the virtual array, or can be brought into the virtual array with the networks.
 - Networks can be added directly to the virtual array, or can be brought in with the storage systems, if the networks were previously added to the ViPR Controller physical assets.
- Optionally, physical storage ports, and storage pools associated with the storage systems in the virtual array, can be assigned for use in the virtual array. When assigned, only those storage ports and storage pools will be used as resources when storage is provisioned from the virtual array.
- A description and complete list of parameters for the REST API methods used in this article are described in the *EMC ViPR Controller REST API Reference* on the [ViPR Controller Product Documentation Index](#).

Procedure

1. Create a virtual array by sending a POST `/vdc/varrays`.

The request returns the representation of the new virtual array.

Request

```
POST https://<ViPR_Controller_VIP>:4443/vdc/varrays
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<varray_create>
  <name>Phoenix</name>
  <auto_san_zoning>true</auto_san_zoning>
</varray_create>
```

Response

```
HTTP 200 OK
Content-Type: application/xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
```

```

<varray>
  <creation_time>1399841344524</creation_time>
  <global>false</global>
  <id>urn:storageos:VirtualArray:44232490-75d6-4bec-
b637-091e14cff7ec:vdcl</id>
  <inactive>false</inactive>
  <internal>false</internal>
  <link_rel="self" href="/vdc/varrays/urn:storageos:VirtualArray:
44232490-75d6-4bec-b637-091e14cff7ec:vdcl"/>
  <name>Phoenix</name>
  <remote>false</remote>
  <tags/>
  <vdc>
    <id>urn:storageos:VirtualDataCenter:440b70de-
fd36-4361-a455-b6ca65fb8228:vdcl</id>
    <link_rel="self" href="/vdc/
urn:storageos:VirtualDataCenter:440b70de-fd36-4361-a455-
b6ca65fb8228:vdcl"/>
    </vdc>
    <auto_san_zoning>true</auto_san_zoning>
    <block_settings>
      <auto_san_zoning>true</auto_san_zoning>
    </block_settings>
    <object_settings>
      <device_registered>false</device_registered>
      <protection_type></protection_type>
    </object_settings>
  </vdc>
</varray>

```

2. Create an ACL for the virtual array and assign the tenant user to the ACL by sending a PUT /vdc/varrays/{Varray_URN}/acl. You can use GET /tenant to get the ID of your own tenant or GET /tenants/bulk to return a list of all tenants in ViPR.

Request

```

PUT https://<ViPR_Controller_VIP>:4443/vdc/varrays/
urn:storageos:VirtualArray:44232490-75d6-4bec-
b637-091e14cff7ec:vdcl/acl
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<acl_assignment_changes>
  <add>
    <privilege>USE</privilege>
    <tenant>urn:storageos:TenantOrg:6c7dde31-
ec48-4028-8672-5e74f5754656:global</tenant>
  </add>
</acl_assignment_changes>

```

Response

```

HTTP 200 OK
Content-Type: application/xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <acl_assignments>
    <acl_assignment>
      <privilege>USE</privilege>

<tenant>urn:storageos:TenantOrg:c7ba7600-1a02-4558-954d-3de642eeacc
e:global</tenant>
    </acl_assignment>
    <acl_assignment>
      <privilege>USE</privilege>
      <tenant>urn:storageos:TenantOrg:6c7dde31-
ec48-4028-8672-5e74f5754656:global</tenant>

```

```

    </acl_assignment>
  </acl_assignments>

```

Add and configure the networks for block storage in the virtual array

Fibre Channel networks are automatically added, discovered, and registered in ViPR when the fabric switch is added to the ViPR physical assets.

- When selecting the Fibre Channel networks to add to the virtual array the storage systems and hosts to which the storage will be provisioned must be configured on the same network.
- If the hosts were added to ViPR, and not discovered, the host ports must be manually added to the networks.
- Optionally, storage ports can be added to the network to control which ports will be used when the storage is provisioned on the host.

Add an existing fibre channel network to a virtual array

Fibre Channel networks are automatically added, discovered, and registered in ViPR when the fabric switch is added to the ViPR Controller physical assets.

Before you begin

Procedure

1. You can determine which networks are available using GET /vdc/networks.

Request

```

GET https://<ViPR_Controller_VIP>:4443/vdc/networks
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

```

Response

```

HTTP 200 OK
Content-Type: application/xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<networks>
  <network>
    <id>urn:storageos:Network:
079b9be1-2342-41a0-84e3-26b4d0fd108a:vdc1</id>
    <link rel="self" href="/vdc/networks/urn:storageos:Network:
079b9be1-2342-41a0-84e3-26b4d0fd108a:vdc1"/>
    <name>FABRIC_vplex154nbr2-10:00:00:27:f8:58:f6:c2</name>
  </network>
  <network>
    <id>urn:storageos:Network:922f1519-e8ee-438a-a11a-
b7f3a139e0b3:vdc1</id>
    <link rel="self" href="/vdc/networks/urn:storageos:Network:
922f1519-e8ee-438a-a11a-b7f3a139e0b3:vdc1"/>
    <name>VSAN_56</name>
  </network>
  <network>
    <id>urn:storageos:Network:c67c9728-4d89-4e57-a345-
dc88bf1213ec:vdc1</id>

```

```

        <link rel="self" href="/vdc/networks/
urn:storageos:Network:c67c9728-4d89-4e57-a345-dc88bf1213ec:vdc1"/>
        <name>FABRIC_Brocade B FE</name>
    </network>
    <network>
        <id>urn:storageos:Network:a0fe52d9-9683-46d8-
b52d-12c6e73b7b98:vdc1</id>
        <link rel="self" href="/vdc/networks/
urn:storageos:Network:a0fe52d9-9683-46d8-b52d-12c6e73b7b98:vdc1"/>
        <name>VSAN_57</name>
    </network>
</networks>

```

2. Use GET /vdc/networks/{Network_URN} to obtain further details about specific networks.

Request

```

GET https://<ViPR_Controller_VIP>:4443/vdc/networks/
urn:storageos:Network:a0fe52d9-9683-46d8-b52d-12c6e73b7b98:vdc1
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

```

Response

```

HTTP 200 OK
Content-Type: application/xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<network>
    <creation_time>1398806587983</creation_time>
    <global>false</global>
    <id>urn:storageos:Network:a0fe52d9-9683-46d8-
b52d-12c6e73b7b98:vdc1</id>
    <inactive>false</inactive>
    <internal>false</internal>
    <link rel="self" href="/vdc/networks/
urn:storageos:Network:a0fe52d9-9683-46d8-b52d-12c6e73b7b98:vdc1"/>
    <name>VSAN_57</name>
    <remote>false</remote>
    <tags/>
    <vdc>
        <id>urn:storageos:VirtualDataCenter:440b70de-fd36-4361-
a455-b6ca65fb8228:vdc1</id>
        <link rel="self" href="/vdc/
urn:storageos:VirtualDataCenter:440b70de-fd36-4361-a455-
b6ca65fb8228:vdc1"/>
    </vdc>
    <native_guid>FC+MDS+20:39:00:2A:6A:3E:3D:41</native_guid>
    <assigned_varrays>
        <assigned_varray>urn:storageos:VirtualArray:27a6a3a8-
ead4-4e11-92af-7d4bc2e5f5af:vdc1</assigned_varray>
    </assigned_varrays>
    <connected_varrays>
        <connected_varray>urn:storageos:VirtualArray:27a6a3a8-
ead4-4e11-92af-7d4bc2e5f5af:vdc1</connected_varray>
    </connected_varrays>
    <discovered>true</discovered>
    <endpoints>
        <endpoint>50:00:14:42:80:60:0B:13</endpoint>
        <endpoint>50:00:14:42:90:60:0B:13</endpoint>
        <endpoint>50:06:01:6B:46:E0:66:37</endpoint>
        <endpoint>50:00:09:73:F0:06:71:40</endpoint>
        <endpoint>50:06:01:63:46:E0:66:37</endpoint>
        <endpoint>50:00:09:73:F0:06:71:44</endpoint>
    </endpoints>

```

```

<endpoints_discovered>
  <endpoint_discovered>
    <name>50:00:14:42:80:60:0B:13</name>
    <value>>true</value>
  </endpoint_discovered>
  <endpoint_discovered>
    <name>50:06:01:6B:46:E0:66:37</name>
    <value>>true</value>
  </endpoint_discovered>
  <endpoint_discovered>
    <name>50:00:14:42:90:60:0B:13</name>
    <value>>true</value>
  </endpoint_discovered>
  <endpoint_discovered>
    <name>50:00:09:73:F0:06:71:40</name>
    <value>>true</value>
  </endpoint_discovered>
  <endpoint_discovered>
    <name>50:06:01:63:46:E0:66:37</name>
    <value>>true</value>
  </endpoint_discovered>
  <endpoint_discovered>
    <name>50:00:09:73:F0:06:71:44</name>
    <value>>true</value>
  </endpoint_discovered>
</endpoints_discovered>
<fabric_id>57</fabric_id>
<network_systems>
  <network_system>urn:storageos:NetworkSystem:74a20311-
cd89-4ce7-90b3-54a3a8bc814a:vdc1</network_system>
</network_systems>
<registration_status>REGISTERED</registration_status>
<transport_type>FC</transport_type>
<varray>
  <id>urn:storageos:VirtualArray:27a6a3a8-
ead4-4e11-92af-7d4bc2e5f5af:vdc1</id>
  <link rel="self" href="/vdc/varrays/
urn:storageos:VirtualArray:27a6a3a8-
ead4-4e11-92af-7d4bc2e5f5af:vdc1"/>
</varray>
</network>

```

3. Select the appropriate fiber channel network (<transport_type>FC</transport_type>), and assign the network to the virtual array by sending a PUT /vdc/networks/{Network_URN} request, including the URN of the virtual array to which the network is to be assigned in the request payload.

Request

```

PUT https://<ViPR_Controller_VIP>:4443/vdc/networks/
urn:storageos:Network:a0fe52d9-9683-46d8-b52d-12c6e73b7b98:vdc1
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<network_update>
  <varrays>
    <varray>urn:storageos:VirtualArray:27a6a3a8-
ead4-4e11-92af-7d4bc2e5f5af:vdc1</varray>
  </varrays>
</network_update>

```

Response

```
HTTP 200 OK
```



```

Content-Type: application/xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<network>
  <creation_time>1398806587983</creation_time>
  <global>>false</global>
  <id>urn:storageos:Network:a0fe52d9-9683-46d8-
b52d-12c6e73b7b98:vdcl</id>
  <inactive>>false</inactive>
  <internal>>false</internal>
  <link rel="self" href="/vdc/networks/
urn:storageos:Network:a0fe52d9-9683-46d8-b52d-12c6e73b7b98:vdcl"/>
  <name>VSAN_57</name>
  <remote>>false</remote>
  <tags/>
  <vdc>
    <id>urn:storageos:VirtualDataCenter:440b70de-fd36-4361-
a455-b6ca65fb8228:vdcl</id>
    <link rel="self" href="/vdc/
urn:storageos:VirtualDataCenter:440b70de-fd36-4361-a455-
b6ca65fb8228:vdcl"/>
  </vdc>
  <native_guid>FC+MDS+20:39:00:2A:6A:3E:3D:41</native_guid>
  <assigned_varrays>
    <assigned_varray>urn:storageos:VirtualArray:27a6a3a8-
ead4-4e11-92af-7d4bc2e5f5af:vdcl</assigned_varray>
  </assigned_varrays>
  <connected_varrays>
    <connected_varray>urn:storageos:VirtualArray:27a6a3a8-
ead4-4e11-92af-7d4bc2e5f5af:vdcl</connected_varray>
  </connected_varrays>
  <discovered>>true</discovered>
  <endpoints>
    <endpoint>50:00:14:42:80:60:0B:13</endpoint>
    <endpoint>50:00:14:42:90:60:0B:13</endpoint>
    <endpoint>50:06:01:6B:46:E0:66:37</endpoint>
    <endpoint>50:00:09:73:F0:06:71:40</endpoint>
    <endpoint>50:06:01:63:46:E0:66:37</endpoint>
    <endpoint>50:00:09:73:F0:06:71:44</endpoint>
  </endpoints>
  <endpoints_discovered>
    <endpoint_discovered>
      <name>50:00:14:42:80:60:0B:13</name>
      <value>>true</value>
    </endpoint_discovered>
    <endpoint_discovered>
      <name>50:06:01:6B:46:E0:66:37</name>
      <value>>true</value>
    </endpoint_discovered>
    <endpoint_discovered>
      <name>50:00:14:42:90:60:0B:13</name>
      <value>>true</value>
    </endpoint_discovered>
    <endpoint_discovered>
      <name>50:00:09:73:F0:06:71:40</name>
      <value>>true</value>
    </endpoint_discovered>
    <endpoint_discovered>
      <name>50:06:01:63:46:E0:66:37</name>
      <value>>true</value>
    </endpoint_discovered>
    <endpoint_discovered>
      <name>50:00:09:73:F0:06:71:44</name>
      <value>>true</value>
    </endpoint_discovered>
  </endpoints_discovered>
  <fabric_id>57</fabric_id>
  <network_systems>
    <network_system>urn:storageos:NetworkSystem:74a20311-
cd89-4ce7-90b3-54a3a8bc814a:vdcl</network_system>

```

```

</network_systems>
<registration_status>REGISTERED</registration_status>
<transport_type>FC</transport_type>
<varray>
  <id>urn:storageos:VirtualArray:27a6a3a8-
ead4-4e11-92af-7d4bc2e5f5af:vdc1</id>
  <link rel="self" href="/vdc/varrays/
urn:storageos:VirtualArray:27a6a3a8-
ead4-4e11-92af-7d4bc2e5f5af:vdc1"/>
</varray>
</network>

```

Configure a new fibre channel network for a virtual array

Procedure

1. Get the list of virtual arrays.

Request

```

GET https://<ViPR_Controller_VIP>:4443/vdc/varrays
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

```

Response

```

HTTP 200 OK
Content-Type: application/xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<varrays>
  <varray>
    <id>urn:storageos:VirtualArray:69b363c5-
f412-4fac-9f5d-0c5367628c17:vdc1</id>
    <link rel="self" href="/vdc/varrays/
urn:storageos:VirtualArray:69b363c5-
f412-4fac-9f5d-0c5367628c17:vdc1"/>
    <name>snapshot test</name>
  </varray>
  <varray>
    <id>urn:storageos:VirtualArray:eed33299-3fa4-4de1-
be44-73949a7a42ea:vdc1</id>
    <link rel="self" href="/vdc/varrays/
urn:storageos:VirtualArray:eed33299-3fa4-4de1-
be44-73949a7a42ea:vdc1"/>
    <name>varray_west_1</name>
  </varray>
  <varray>
    <id>urn:storageos:VirtualArray:e6f1d340-9a32-4c4a-990d-
c346125b4141:vdc1</id>
    <link rel="self" href="/vdc/varrays/
urn:storageos:VirtualArray:e6f1d340-9a32-4c4a-990d-
c346125b4141:vdc1"/>
    <name>varray_east_1</name>
  </varray>
  <varray>
    <id>urn:storageos:VirtualArray:bd15ab0c-
fc25-4256-9ccb-73dea570d65c:vdc1</id>
    <link rel="self" href="/vdc/varrays/
urn:storageos:VirtualArray:bd15ab0c-
fc25-4256-9ccb-73dea570d65c:vdc1"/>
    <name>varray2</name>
  </varray>
</varrays>

```

- Using the URN of the selected virtual array, configure a new fibre channel network for the selected virtual array by sending POST `/vdc/varrays/{id}/networks`.

Request

```
POST https://<ViPR_Controller_VIP>:4443/vdc/varrays/
urn:storageos:VirtualArray:eed33299-3fa4-4de1-
be44-73949a7a42ea:vdc1/networks
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<network_create>
  <name>FC_network_west_1</name>
  <transport_type>FC</transport_type>
  <endpoints>
    <endpoint>10:00:20:00:c9:79:f1:46 </endpoint>
    <endpoint>10:00:20:00:c9:79:f1:47 </endpoint>
  </endpoints>
</network_create>
```

Reponse

```
HTTP 200 OK
Content-Type: application/xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<network>
  <creation_time>1400177060451</creation_time>
  <global>false</global>
  <id>urn:storageos:Network:e508d8b6-6738-4311-
ad4d-4c582cf7b0d6:vdc1</id>
  <inactive>false</inactive>
  <internal>false</internal>
  <link rel="self" href="/vdc/networks/
urn:storageos:Network:e508d8b6-6738-4311-ad4d-4c582cf7b0d6:vdc1"/>
  <name>FC_network_west_1</name>
  <remote>false</remote>
  <tags/>
  <vdc>
    <id>urn:storageos:VirtualDataCenter:030618c2-c6b2-40b0-
a105-6b669983f58f:vdc1</id>
    <link rel="self" href="/vdc/
urn:storageos:VirtualDataCenter:030618c2-c6b2-40b0-
a105-6b669983f58f:vdc1"/>
  </vdc>
  <native_guid></native_guid>
  <assigned_varrays>

<assigned_varray>urn:storageos:VirtualArray:eed33299-3fa4-4de1-
be44-73949a7a42ea:vdc1</assigned_varray>
</assigned_varrays>
<connected_varrays>

<connected_varray>urn:storageos:VirtualArray:eed33299-3fa4-4de1-
be44-73949a7a42ea:vdc1</connected_varray>
</connected_varrays>
<discovered>false</discovered>
<endpoints>
  <endpoint>10:00:20:00:C9:79:F1:47</endpoint>
  <endpoint>10:00:20:00:C9:79:F1:46</endpoint>
</endpoints>
<endpoints_discovered>
  <endpoint_discovered>
    <name>10:00:20:00:C9:79:F1:47</name>
    <value>false</value>
  </endpoint_discovered>
  <endpoint_discovered>
    <name>10:00:20:00:C9:79:F1:46</name>
    <value>false</value>
```

```

        </endpoint_discovered>
    </endpoints_discovered>
    <registration_status>REGISTERED</registration_status>
    <transport_type>FC</transport_type>
    <varray>
        <id>urn:storageos:VirtualArray:eed33299-3fa4-4de1-
be44-73949a7a42ea:vdcl</id>
        <link rel="self" href="/vdc/varrays/
urn:storageos:VirtualArray:eed33299-3fa4-4de1-
be44-73949a7a42ea:vdcl"/>
    </varray>
</network>

```

Add and configure the networks for file storage in a virtual array

ViPR Controller can discover the ports of IP connected storage systems and hosts, but it cannot discover the paths between them, so it is necessary to create IP networks, and then add the host, and storage system ports, which will be provisioned together, to the same IP network.

Add an existing IP network to a virtual array

If the IP network has already been created, and configured using POST `/vdc/networks`, you will just need to add the network to the virtual array.

Before you begin

URN of the virtual array to which the IP network is being added.

Procedure

1. You can get which networks are available using GET `/vdc/networks`.

Request

```

GET https://<ViPR_Controller_VIP>:4443/vdc/networks
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

```

Response

```

HTTP 200 OK
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<networks>
  <network>
    <id>urn:storageos:Network:b10c046f-0968-4c30-91cd-
b2b34f1d7f5e:vdcl</id>
    <link rel="self" href="/vdc/networks/
urn:storageos:Network:b10c046f-0968-4c30-91cd-b2b34f1d7f5e:vdcl"/>
    <name>FABRIC_vplex154nbr2</name>
  </network>
  <network>
    <id>urn:storageos:Network:fc2824bd-0935-43e4-
a062-748b48f7f99e:vdcl</id>
    <link rel="self" href="/vdc/networks/
urn:storageos:Network:fc2824bd-0935-43e4-a062-748b48f7f99e:vdcl"/>
    <name>FABRIC_VPlex_Meta_Fid_20</name>
  </network>
  <network>
    <id>urn:storageos:Network:abaf3cec-cdd1-4751-b9bf-

```

```

ba02dafa7603:vdcl</id>
  <link rel="self" href="/vdc/networks/
urn:storageos:Network:abaf3cec-cdd1-4751-b9bf-ba02dafa7603:vdcl"/>
  <name>FABRIC_VPLEX_WAN_lglw6208/lglw6209</name>
</network>
<network>
  <id>urn:storageos:Network:6c5cbf7f-d0eb-40f8-9e35-
e35e2c8d5f77:vdcl</id>
  <link rel="self" href="/vdc/networks/urn:storageos:Network:
6c5cbf7f-d0eb-40f8-9e35-e35e2c8d5f77:vdcl"/>
  <name>FABRIC_VPlex_LGL6221_FID_40</name>
</network>
<network>
  <id>urn:storageos:Network:beb93fc4-d696-4be3-
a4c9-80fab40d34b4:vdcl</id>
  <link rel="self" href="/vdc/networks/
urn:storageos:Network:beb93fc4-d696-4be3-a4c9-80fab40d34b4:vdcl"/>
  <name>IP_network_west1</name>
</network>
<network>
  <id>urn:storageos:Network:602444e4-
b338-4620-9e07-84247de5406d:vdcl</id>
  <link rel="self" href="/vdc/networks/urn:storageos:Network:
602444e4-b338-4620-9e07-84247de5406d:vdcl"/>
  <name>FABRIC_Vplex_WAN-10:00:00:27:f8:58:f6:bc</name>
</network>
<network>
  <id>urn:storageos:Network:eff96c90-
d67a-400e-9b07-2c308ba5ee51:vdcl</id>
  <link rel="self" href="/vdc/networks/
urn:storageos:Network:eff96c90-d67a-400e-9b07-2c308ba5ee51:vdcl"/>
  <name>FABRIC_fake array fabric</name>
</network>
<network>
  <id>urn:storageos:Network:0776alc7-97d6-498d-ad01-
d03c9cc10925:vdcl</id>
  <link rel="self" href="/vdc/networks/urn:storageos:Network:
0776alc7-97d6-498d-ad01-d03c9cc10925:vdcl"/>
  <name>FABRIC_VPlex_LGL6220_FID_30-10:00:00:27:f8:58:f6:c1</
name>
</network>
<network>
  <id>urn:storageos:Network:cb3a82dd-0202-4d6a-8f3f-3966fd817e82:vdcl
</id>
  <link rel="self" href="/vdc/networks/
urn:storageos:Network:cb3a82dd-0202-4d6a-8f3f-3966fd817e82:vdcl"/>
  <name>FABRIC_losam082-fabric</name>
</network>
</networks>

```

2. Use GET /vdc/networks/{Network_URN} to get further details about a specific network.

Request

```

GET https://<ViPR_Controller_VIP>:4443/vdc/networks/
urn:storageos:Network:beb93fc4-d696-4be3-a4c9-80fab40d34b4:vdcl
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

```

Response

```

HTTP 200 OK
Content-Type: application/xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

```

```

<network>
  <creation_time>1400172861659</creation_time>
  <global>false</global>
  <id>urn:storageos:Network:beb93fc4-d696-4be3-a4c9-80fab40d34b4:vdc1</id>
  <inactive>false</inactive>
  <internal>false</internal>
  <link_rel="self" href="/vdc/networks/urn:storageos:Network:beb93fc4-d696-4be3-a4c9-80fab40d34b4:vdc1"/>
  <name>IP_network_west1</name>
  <remote>false</remote>
  <tags/>
  <vdc>
    <id>urn:storageos:VirtualDataCenter:030618c2-c6b2-40b0-a105-6b669983f58f:vdc1</id>
    <link_rel="self" href="/vdc/urn:storageos:VirtualDataCenter:030618c2-c6b2-40b0-a105-6b669983f58f:vdc1"/>
  </vdc>
  <native_guid></native_guid>
  <discovered>false</discovered>
  <endpoints>
    <endpoint>192.168.0.10</endpoint>
    <endpoint>192.168.0.20</endpoint>
  </endpoints>
  <endpoints_discovered>
    <endpoint_discovered>
      <name>192.168.0.10</name>
      <value>false</value>
    </endpoint_discovered>
    <endpoint_discovered>
      <name>192.168.0.20</name>
      <value>false</value>
    </endpoint_discovered>
  </endpoints_discovered>
  <registration_status>REGISTERED</registration_status>
  <transport_type>IP</transport_type>
</network>

```

3. Select the appropriate network, and assign the IP network to the virtual array by sending a PUT /vdc/networks/{Network_URN} request, including the URN of the virtual array to which the network is to be assigned in the request payload. The updated network representation, with the varray added, is returned.

Request

```

PUT https://<ViPR_Controller_VIP>:4443/vdc/networks/urn:storageos:Network:beb93fc4-d696-4be3-a4c9-80fab40d34b4:vdc1
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<network_update>
  <varrays>
    <varray>urn:storageos:VirtualArray:bd15ab0c-fc25-4256-9ccb-73dea570d65c:vdc1</varray>
  </varrays>
</network_update>

```

Response

```

HTTP 200 OK
Content-Type: application/xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<network>
  <creation_time>1400172861659</creation_time>

```

```

    <global>false</global>
    <id>urn:storageos:Network:beb93fc4-d696-4be3-
a4c9-80fab40d34b4:vdc1</id>
    <inactive>false</inactive>
    <internal>false</internal>
    <link rel="self" href="/vdc/networks/
urn:storageos:Network:beb93fc4-d696-4be3-a4c9-80fab40d34b4:vdc1"/>
    <name>IP_network_west1</name>
    <remote>false</remote>
    <tags/>
    <vdc>
      <id>urn:storageos:VirtualDataCenter:030618c2-c6b2-40b0-
a105-6b669983f58f:vdc1</id>
      <link rel="self" href="/vdc/
urn:storageos:VirtualDataCenter:030618c2-c6b2-40b0-
a105-6b669983f58f:vdc1"/>
    </vdc>
    <native_guid></native_guid>
    <assigned_varrays>
      <assigned_varray>urn:storageos:VirtualArray:bd15ab0c-
fc25-4256-9ccb-73dea570d65c:vdc1</assigned_varray>
    </assigned_varrays>
    <connected_varrays>
      <connected_varray>urn:storageos:VirtualArray:bd15ab0c-
fc25-4256-9ccb-73dea570d65c:vdc1</connected_varray>
    </connected_varrays>
    <discovered>false</discovered>
    <endpoints>
      <endpoint>192.168.0.10</endpoint>
      <endpoint>192.168.0.20</endpoint>
    </endpoints>
    <endpoints_discovered>
      <endpoint_discovered>
        <name>192.168.0.10</name>
        <value>false</value>
      </endpoint_discovered>
      <endpoint_discovered>
        <name>192.168.0.20</name>
        <value>false</value>
      </endpoint_discovered>
    </endpoints_discovered>
    <registration_status>REGISTERED</registration_status>
    <transport_type>IP</transport_type>
    <varray>
      <id>urn:storageos:VirtualArray:bd15ab0c-
fc25-4256-9ccb-73dea570d65c:vdc1</id>
      <link rel="self" href="/vdc/varrays/
urn:storageos:VirtualArray:bd15ab0c-
fc25-4256-9ccb-73dea570d65c:vdc1"/>
    </varray>
  </network>

```

Configure a new IP network for a virtual array

Procedure

1. Get the list of virtual arrays.

Request

```

GET https://<ViPR_Controller_VIP>:4443/vdc/varrays
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

```

Response

```

HTTP 200 OK
Content-Type: application/xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<varrays>
  <varray>
    <id>urn:storageos:VirtualArray:69b363c5-
f412-4fac-9f5d-0c5367628c17:vdcl</id>
    <link rel="self" href="/vdc/varrays/
urn:storageos:VirtualArray:69b363c5-
f412-4fac-9f5d-0c5367628c17:vdcl"/>
    <name>snapshot test</name>
  </varray>
  <varray>
    <id>urn:storageos:VirtualArray:eed33299-3fa4-4de1-
be44-73949a7a42ea:vdcl</id>
    <link rel="self" href="/vdc/varrays/
urn:storageos:VirtualArray:eed33299-3fa4-4de1-
be44-73949a7a42ea:vdcl"/>
    <name>varray_west_1</name>
  </varray>
  <varray>
    <id>urn:storageos:VirtualArray:e6f1d340-9a32-4c4a-990d-
c346125b4141:vdcl</id>
    <link rel="self" href="/vdc/varrays/
urn:storageos:VirtualArray:e6f1d340-9a32-4c4a-990d-
c346125b4141:vdcl"/>
    <name>varray_east_1</name>
  </varray>
  <varray>
    <id>urn:storageos:VirtualArray:bd15ab0c-
fc25-4256-9ccb-73dea570d65c:vdcl</id>
    <link rel="self" href="/vdc/varrays/
urn:storageos:VirtualArray:bd15ab0c-
fc25-4256-9ccb-73dea570d65c:vdcl"/>
    <name>varray2</name>
  </varray>
</varrays>

```

- Using the URN of the selected virtual array, configure a new IP network for the selected virtual array by sending POST /vdc/varrays/{id}/networks.

Request

```

POST https://<ViPR_Controller_VIP>:4443/vdc/varrays/
urn:storageos:VirtualArray:e6f1d340-9a32-4c4a-990d-
c346125b4141:vdcl/networks
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<network_create>
  <name>IP_network_east_1</name>
  <transport_type>IP</transport_type>
  <endpoints>
    <endpoint>192.169.100.10</endpoint>
    <endpoint>192.168.100.40</endpoint>
  </endpoints>
</network_create>

```

Response

```

HTTP 200 OK
Content-Type: application/xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

```



```

<network>
  <creation_time>1400181561583</creation_time>
  <global>false</global>
  <id>urn:storageos:Network:f6e2d237-
a770-4085-924a-0d11afdb8e6a:vdc1</id>
  <inactive>false</inactive>
  <internal>false</internal>
  <link rel="self" href="/vdc/networks/
urn:storageos:Network:f6e2d237-a770-4085-924a-0d11afdb8e6a:vdc1"/>
  <name>IP_network_east_1</name>
  <remote>false</remote>
  <tags/>
  <vdc>
    <id>urn:storageos:VirtualDataCenter:030618c2-c6b2-40b0-
a105-6b669983f58f:vdc1</id>
    <link rel="self" href="/vdc/
urn:storageos:VirtualDataCenter:030618c2-c6b2-40b0-
a105-6b669983f58f:vdc1"/>
    </vdc>
    <native_guid></native_guid>
    <assigned_varrays>

<assigned_varray>urn:storageos:VirtualArray:e6f1d340-9a32-4c4a-990d-
c346125b4141:vdc1</assigned_varray>
  </assigned_varrays>
  <connected_varrays>

<connected_varray>urn:storageos:VirtualArray:e6f1d340-9a32-4c4a-990
d-c346125b4141:vdc1</connected_varray>
  </connected_varrays>
  <discovered>false</discovered>
  <endpoints>
    <endpoint>192.168.100.40</endpoint>
    <endpoint>192.169.100.10</endpoint>
  </endpoints>
  <endpoints_discovered>
    <endpoint_discovered>
      <name>192.168.100.40</name>
      <value>false</value>
    </endpoint_discovered>
    <endpoint_discovered>
      <name>192.169.100.10</name>
      <value>false</value>
    </endpoint_discovered>
  </endpoints_discovered>
  <registration_status>REGISTERED</registration_status>
  <transport_type>IP</transport_type>
  <varray>
    <id>urn:storageos:VirtualArray:e6f1d340-9a32-4c4a-990d-
c346125b4141:vdc1</id>
    <link rel="self" href="/vdc/varrays/
urn:storageos:VirtualArray:e6f1d340-9a32-4c4a-990d-
c346125b4141:vdc1"/>
    </varray>
  </network>

```

Optionally restrict access to storage pools by manually assigning them to a virtual array

When no specific storage pools are manually assigned to a virtual array, then all available pools from the storage arrays are available to the virtual array. Optionally, you can restrict access to certain pools by manually assigning them to a specific virtual array.

Before you begin

Authenticate with the ViPR REST API as a System Administrator. See [Authenticating with the REST API on page 9](#).

Procedure

1. Get the URN of the storage system, on which the storage pools were created, by sending GET /vdc/storage-systems

This request returns a list of all of the physical storage arrays.

Request

```
GET https://<ViPR_Controller_VIP>:4443/vdc/storage-systems
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>
```

Response

```
HTTP 200 OK
Content-Type: application/xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<storage_systems>
  <storage_system>
    <id>urn:storageos:StorageSystem:
2370c474-4d8d-44a9-8ebf-58aa87b5b3ea:vdcl</id>
    <link rel="self" href="/vdc/storage-systems/
urn:storageos:StorageSystem:
2370c474-4d8d-44a9-8ebf-58aa87b5b3ea:vdcl"/>
    <name>VNXE+FNM00135000061</name>
  </storage_system>
  <storage_system>
    <id>urn:storageos:StorageSystem:acc7c01a-0d1d-48b8-
afce-93636e0cb748:vdcl</id>
    <link rel="self" href="/vdc/storage-systems/
urn:storageos:StorageSystem:acc7c01a-0d1d-48b8-
afce-93636e0cb748:vdcl"/>
    <name>CLARIION+APM00112800557</name>
  </storage_system>
  <storage_system>
    <id>urn:storageos:StorageSystem:a27276b9-
b2ae-4bba-8741-08be646efc92:vdcl</id>
    <link rel="self" href="/vdc/storage-systems/
urn:storageos:StorageSystem:a27276b9-
b2ae-4bba-8741-08be646efc92:vdcl"/>
    <name>losat018.lss.emc.com</name>
  </storage_system>
  <storage_system>
    <id>urn:storageos:StorageSystem:848eedd3-f5a8-4c14-bde8-
c11a82b93dce:vdcl</id>
    <link rel="self" href="/vdc/storage-systems/
urn:storageos:StorageSystem:848eedd3-f5a8-4c14-bde8-
c11a82b93dce:vdcl"/>
    <name>IBMXIV+IBM.2810-7825363</name>
  </storage_system>
```

```

    <storage_system>
      <id>urn:storageos:StorageSystem:d8012fac-4237-4953-8857-
adc217a7c5dd:vdcl</id>
      <link rel="self" href="/vdc/storage-systems/
urn:storageos:StorageSystem:d8012fac-4237-4953-8857-
adc217a7c5dd:vdcl"/>
      <name>isilon1</name>
    </storage_system>
    <storage_system>
      <id>urn:storageos:StorageSystem:2b576fd1-c40a-4bbf-
bdb6-73c529b13487:vdcl</id>
      <link rel="self" href="/vdc/storage-systems/
urn:storageos:StorageSystem:2b576fd1-c40a-4bbf-
bdb6-73c529b13487:vdcl"/>
      <name>SYMMETRIX+000198700406</name>
    </storage_system>
  </storage_systems>

```

2. Use the storage system URN to get detailed information about the storage array by sending a GET /vdc/storage-systems/{Storage_URN} request.

The request returns the representation of the storage system.

Request

```

GET https://<ViPR_Controller_VIP>:4443/vdc/storage-systems/
urn:storageos:StorageSystem:
2370c474-4d8d-44a9-8ebf-58aa87b5b3ea:vdcl
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

```

Response

```

HTTP 200 OK
Content-Type: application/xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<storage_system>
  <creation_time>1421364124360</creation_time>
  <global>false</global>
  <id>urn:storageos:StorageSystem:
2370c474-4d8d-44a9-8ebf-58aa87b5b3ea:vdcl</id>
  <inactive>false</inactive>
  <internal>false</internal>
  <link rel="self" href="/vdc/storage-systems/
urn:storageos:StorageSystem:
2370c474-4d8d-44a9-8ebf-58aa87b5b3ea:vdcl"/>
  <name>VNXE+FNM00135000061</name>
  <remote>false</remote>
  <tags/>
  <vdc>
    <id>urn:storageos:VirtualDataCenter:99c5c185-
ebb0-468a-8fe7-3ea6e65b7b57:vdcl</id>
    <link rel="self" href="/vdc/
urn:storageos:VirtualDataCenter:99c5c185-
ebb0-468a-8fe7-3ea6e65b7b57:vdcl"/>
  </vdc>
  <native_guid>VNXE+FNM00135000061</native_guid>
  <compatibility_status>COMPATIBLE</compatibility_status>
  <job_discovery_status>COMPLETE</job_discovery_status>
  <last_discovery_run_time>1423607690987</
last_discovery_run_time>
  <last_discovery_status_message>Discovery completed
successfully for Storage System: urn:storageos:StorageSystem:
2370c474-4d8d-44a9-8ebf-58aa87b5b3ea:vdcl</
last_discovery_status_message>

```

```

        <last_metering_run_time>1423607644158</last_metering_run_time>
        <job_metering_status>COMPLETE</job_metering_status>
        <next_discovery_run_time>1423611274176</
next_discovery_run_time>
        <next_metering_run_time>1423611244106</next_metering_run_time>
        <registration_status>REGISTERED</registration_status>
        <success_discovery_time>1423607690987</success_discovery_time>
        <success_metering_time>1423607644158</success_metering_time>
        <system_type>vnx</system_type>
        <associated_systems/>
        <export_masks/>
        <firmware_version>3.0.1</firmware_version>
        <ip_address>10.247.40.221</ip_address>
        <max_resources>-1</max_resources>
        <num_resources>0</num_resources>
        <port_number>443</port_number>
        <protocols/>
        <smis_providers/>
        <reachable>true</reachable>
        <connected_systems/>
        <secondary_ips/>
        <serial_number>FNM00135000061</serial_number>
        <smis_use_ssl>>false</smis_use_ssl>
        <async_actions>
            <async_action>CreateGroupReplica</async_action>
            <async_action>CreateElementReplica</async_action>
        </async_actions>
        <supported_replication_types>
            <supported_replication_type>LOCAL</
supported_replication_type>
supported_replication_type>
        </supported_replication_types>
        <user_name>bourne</user_name>
    </storage_system>

```

3. Get the list of virtual arrays by sending GET /vdc/varrays.

Request

```

GET https://<ViPR_Controller_VIP>:4443/vdc/varrays
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

```

Response

```

HTTP 200 OK
Content-Type: application/xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<varrays>
    <varray>
        <id>urn:storageos:VirtualArray:ac294bc7-0518-48dd-9f54-4980f0012f86:vdcl</id>
        <link rel="self" href="/vdc/varrays/urn:storageos:VirtualArray:ac294bc7-0518-48dd-9f54-4980f0012f86:vdcl"/>
        <name>vblock_varray1</name>
    </varray>
    <varray>
        <id>urn:storageos:VirtualArray:0d3276fc-a65b-45c5-bc41-527251476738:vdcl</id>
        <link rel="self" href="/vdc/varrays/urn:storageos:VirtualArray:0d3276fc-a65b-45c5-bc41-527251476738:vdcl"/>
        <name>Phoenix</name>
    </varray>
</varrays>

```

4. Use the virtual array URN to get detailed information about the virtual array.

Request

```
GET https://<ViPR_Controller_VIP>:4443/vdc/varrays/
urn:storageos:VirtualArray:ac294bc7-0518-48dd-9f54-4980f0012f86:vdc
1
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>
```

Response

```
HTTP 200 OK
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<varray>
  <creation_time>1422289959135</creation_time>
  <global>>false</global>

  <id>urn:storageos:VirtualArray:ac294bc7-0518-48dd-9f54-4980f0012f86
:vdc1</id>
  <inactive>>false</inactive>
  <internal>>false</internal>
  <link rel="self" href="/vdc/varrays/
urn:storageos:VirtualArray:ac294bc7-0518-48dd-9f54-4980f0012f86:vdc
1"/>
  <name>vblock_varray1</name>
  <remote>>false</remote>
  <tags/>
  <vdc>
    <id>urn:storageos:VirtualDataCenter:99c5c185-
ebb0-468a-8fe7-3ea6e65b7b57:vdc1</id>
    <link rel="self" href="/vdc/
urn:storageos:VirtualDataCenter:99c5c185-
ebb0-468a-8fe7-3ea6e65b7b57:vdc1"/>
    </vdc>
    <auto_san_zoning>>true</auto_san_zoning>
    <block_settings>
      <auto_san_zoning>>true</auto_san_zoning>
    </block_settings>
    <object_settings>
      <device_registered>>false</device_registered>
      <protection_type/>
    </object_settings>
  </vdc>
</varray>
```

5. Get the list of storage pools on the storage system by sending GET /vdc/storage-systems/{Storage_URN}/storage-pools.

Request

```
GET https://<ViPR_Controller_VIP>:4443/vdc/storage-systems/
urn:storageos:StorageSystem:
2370c474-4d8d-44a9-8ebf-58aa87b5b3ea:vdc1/storage-pools
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>
```

Response

```
HTTP 200 OK
Content-Type: application/xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<storage_pools>
```

```

    <storage_pool>
      <id>urn:storageos:StoragePool:33e228e5-9547-4f69-a7c3-
c3bc7d36bc78:vdcl</id>
      <link rel="self" href="/vdc/storage-systems/
urn:storageos:StorageSystem:
2370c474-4d8d-44a9-8ebf-58aa87b5b3ea:vdcl/storage-pools/
urn:storageos:StoragePool:33e228e5-9547-4f69-a7c3-
c3bc7d36bc78:vdcl"/>
      <name>VNXE+FNM00135000061+POOL+pool_1</name>
    </storage_pool>
    <storage_pool>
      <id>urn:storageos:StoragePool:4ba86c2b-19c1-4714-aff4-
d8ebc5fc808a:vdcl</id>
      <link rel="self" href="/vdc/storage-systems/
urn:storageos:StorageSystem:
2370c474-4d8d-44a9-8ebf-58aa87b5b3ea:vdcl/storage-pools/
urn:storageos:StoragePool:4ba86c2b-19c1-4714-aff4-
d8ebc5fc808a:vdcl"/>
      <name>VNXE+FNM00135000061+POOL+pool_8</name>
    </storage_pool>
    <storage_pool>
      <id>urn:storageos:StoragePool:b401f7ce-
fe40-4be0-8eea-02bd332329fd:vdcl</id>
      <link rel="self" href="/vdc/storage-systems/
urn:storageos:StorageSystem:
2370c474-4d8d-44a9-8ebf-58aa87b5b3ea:vdcl/storage-pools/
urn:storageos:StoragePool:b401f7ce-
fe40-4be0-8eea-02bd332329fd:vdcl"/>
      <name>VNXE+FNM00135000061+POOL+pool_2</name>
    </storage_pool>
    <storage_pool>
      <id>urn:storageos:StoragePool:f42a22a4-398c-4d4d-
a9f2-49a38010272b:vdcl</id>
      <link rel="self" href="/vdc/storage-systems/
urn:storageos:StorageSystem:
2370c474-4d8d-44a9-8ebf-58aa87b5b3ea:vdcl/storage-pools/
urn:storageos:StoragePool:f42a22a4-398c-4d4d-
a9f2-49a38010272b:vdcl"/>
      <name>VNXE+FNM00135000061+POOL+pool_4</name>
    </storage_pool>
  </storage_pools>

```

6. Get the detailed information of the storage pool to add to the virtual array by sending GET /vdc/storage-systems/{Storage_URN}/storage-pools/{Pool_URN}.

Request

```

GET https://<ViPR_Controller_VIP>:4443/vdc/storage-systems/
urn:storageos:StorageSystem:
2370c474-4d8d-44a9-8ebf-58aa87b5b3ea:vdcl/storage-pools/
urn:storageos:StoragePool:33e228e5-9547-4f69-a7c3-c3bc7d36bc78:vdcl
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

```

Response

```

HTTP 200 OK
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<storage_pool>
  <creation_time>1421364142991</creation_time>
  <global>false</global>
  <id>urn:storageos:StoragePool:33e228e5-9547-4f69-a7c3-
c3bc7d36bc78:vdcl</id>

```

```

    <inactive>>false</inactive>
    <internal>>false</internal>
    <link rel="self" href="/vdc/storage-systems/
urn:storageos:StorageSystem:
2370c474-4d8d-44a9-8ebf-58aa87b5b3ea:vdc1/storage-pools/
urn:storageos:StoragePool:33e228e5-9547-4f69-a7c3-
c3bc7d36bc78:vdc1"/>
    <name>VNXE+FNM00135000061+POOL+pool_1</name>
    <remote>>false</remote>
    <tags/>
    <vdc>
      <id>urn:storageos:VirtualDataCenter:99c5c185-
ebb0-468a-8fe7-3ea6e65b7b57:vdc1</id>
      <link rel="self" href="/vdc/
urn:storageos:VirtualDataCenter:99c5c185-
ebb0-468a-8fe7-3ea6e65b7b57:vdc1"/>
    </vdc>
    <native_guid>VNXE+FNM00135000061+POOL+pool_1</native_guid>
    <auto_tiering_supported>>false</auto_tiering_supported>
    <compatibility_status>COMPATIBLE</compatibility_status>
    <controller_params/>
    <copy_types7/>
    <discovery_status>VISIBLE</discovery_status>
    <drive_types>
      <drive_type>SAS</drive_type>
    </drive_types>
    <free_gb>590</free_gb>
    <long_term_retention>>false</long_term_retention>
    <max_pool_utilization_percentage>75</
max_pool_utilization_percentage>
    <max_resources>-1</max_resources>
    <max_thin_pool_subscription_percentage>300</
max_thin_pool_subscription_percentage>
    <maximum_thick_volume_size_gb>590</
maximum_thick_volume_size_gb>
    <maximum_thin_volume_size_gb>590</maximum_thin_volume_size_gb>
    <minimum_thick_volume_size_gb>0</minimum_thick_volume_size_gb>
    <minimum_thin_volume_size_gb>0</minimum_thin_volume_size_gb>
    <operational_status>READY</operational_status>
    <percent_subscribed>89</percent_subscribed>
    <percent_used>45</percent_used>
    <pool_name>StoragePool100</pool_name>
    <pool_service_type>block_file</pool_service_type>
    <protocols>
      <protocol>iSCSI</protocol>
      <protocol>CIFS</protocol>
      <protocol>FC</protocol>
      <protocol>NFS</protocol>
    </protocols>
    <raid_levels>
      <raid_level>RAID5</raid_level>
    </raid_levels>
    <registration_status>REGISTERED</registration_status>
    <storage_system>
      <id>urn:storageos:StorageSystem:
2370c474-4d8d-44a9-8ebf-58aa87b5b3ea:vdc1</id>
      <link rel="self" href="/vdc/storage-systems/
urn:storageos:StorageSystem:
2370c474-4d8d-44a9-8ebf-58aa87b5b3ea:vdc1"/>
    </storage_system>
    <subscribed_gb>945</subscribed_gb>
    <supported_resource_types>THIN_AND_THICK</
supported_resource_types>
    <thin_volume_preallocation_supported>>false</
thin_volume_preallocation_supported>
    <tier_utilization_percentages/>
    <usable_gb>1072</usable_gb>
    <used_gb>482</used_gb>
  </storage_pool>

```

7. Assign a storage pool to the virtual array by sending PUT /vdc/storage-pools/{Pool_URN}. The URN of the virtual array to which the storage pool is being assigned is included in the request payload.

The updated representation of the virtual array is returned.

Request

```
PUT https://<ViPR_Controller_VIP>:4443/vdc/storage-pools/
urn:storageos:StoragePool:33e228e5-9547-4f69-a7c3-c3bc7d36bc78:vdc1
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<storage_pool_update>
  <varray_assignment_changes>
    <add>

<varray>urn:storageos:VirtualArray:ac294bc7-0518-48dd-9f54-4980f001
2f86:vdc1</varray>
    </add>
  </varray_assignment_changes>
</storage_pool_update>
```

Response

```
HTTP 200 OK
Content-Type: application/xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<storage_pool>
  <creation_time>1421364142991</creation_time>
  <global>>false</global>
  <id>urn:storageos:StoragePool:33e228e5-9547-4f69-a7c3-
c3bc7d36bc78:vdc1</id>
  <inactive>>false</inactive>
  <internal>>false</internal>
  <link rel="self" href="/vdc/storage-systems/
urn:storageos:StorageSystem:
2370c474-4d8d-44a9-8ebf-58aa87b5b3ea:vdc1/storage-pools/
urn:storageos:StoragePool:33e228e5-9547-4f69-a7c3-
c3bc7d36bc78:vdc1"/>
  <name>VNXE+FNM00135000061+POOL+pool_1</name>
  <remote>>false</remote>
  <tags/>
  <vdc>
    <id>urn:storageos:VirtualDataCenter:99c5c185-
ebb0-468a-8fe7-3ea6e65b7b57:vdc1</id>
    <link rel="self" href="/vdc/
urn:storageos:VirtualDataCenter:99c5c185-
ebb0-468a-8fe7-3ea6e65b7b57:vdc1"/>
  </vdc>
  <native_guid>VNXE+FNM00135000061+POOL+pool_1</native_guid>

<assigned_varrays>urn:storageos:VirtualArray:ac294bc7-0518-48dd-9f5
4-4980f0012f86:vdc1</assigned_varrays>

<tagged_varrays>urn:storageos:VirtualArray:ac294bc7-0518-48dd-9f54-
4980f0012f86:vdc1</tagged_varrays>
  <auto_tiering_supported>>false</auto_tiering_supported>
  <compatibility_status>COMPATIBLE</compatibility_status>
  <controller_params/>
  <copy_types/>
  <discovery_status>VISIBLE</discovery_status>
  <drive_types>
    <drive_type>SAS</drive_type>
  </drive_types>
  <free_gb>590</free_gb>
  <long_term_retention>>false</long_term_retention>
```



```

    <max_pool_utilization_percentage>75</
max_pool_utilization_percentage>
    <max_resources>-1</max_resources>
    <max_thin_pool_subscription_percentage>300</
max_thin_pool_subscription_percentage>
    <maximum_thick_volume_size_gb>590</
maximum_thick_volume_size_gb>
    <maximum_thin_volume_size_gb>590</maximum_thin_volume_size_gb>
    <minimum_thick_volume_size_gb>0</minimum_thick_volume_size_gb>
    <minimum_thin_volume_size_gb>0</minimum_thin_volume_size_gb>
    <operational_status>READY</operational_status>
    <percent_subscribed>89</percent_subscribed>
    <percent_used>45</percent_used>
    <pool_name>StoragePool100</pool_name>
    <pool_service_type>block_file</pool_service_type>
    <protocols>
      <protocol>iSCSI</protocol>
      <protocol>CIFS</protocol>
      <protocol>FC</protocol>
      <protocol>NFS</protocol>
    </protocols>
    <raid_levels>
      <raid_level>RAID5</raid_level>
    </raid_levels>
    <registration_status>REGISTERED</registration_status>
    <storage_system>
      <id>urn:storageos:StorageSystem:
2370c474-4d8d-44a9-8ebf-58aa87b5b3ea:vdc1</id>
      <link_rel="self" href="/vdc/storage-systems/
urn:storageos:StorageSystem:
2370c474-4d8d-44a9-8ebf-58aa87b5b3ea:vdc1"/>
    </storage_system>
    <subscribed_gb>945</subscribed_gb>
    <supported_resource_types>THIN_AND_THICK</
supported_resource_types>
    <thin_volume_preallocation_supported>false</
thin_volume_preallocation_supported>
    <tier_utilization_percentages/>
    <usable_gb>1072</usable_gb>
    <used_gb>482</used_gb>
</storage_pool>

```

Virtual Array requirements for Vblock system services

For Vblock systems, storage must be accessible to compute systems through the virtual array. Vblock systems configured using the VCE logical build guide will have networks configured that connect the Cisco Unified Computing System™ (UCS) compute system to the storage via the SAN switches.

In ViPR Controller, virtual arrays should be created just as you would for any non-Vblock system. The networks that are defined in the virtual arrays will then determine whether the UCS systems have visibility to ViPR Controller storage.

The most effective thing is to do is discover all the Vblock system physical assets before defining virtual arrays. After discovering all components, consult with the UCS administrator to determine which networks (VSANs) will be used on a given Vblock system. Use those networks to define the ViPR Controller virtual arrays. On less complicated Vblock system configurations, for example, a single Vblock system, simply adding the storage system to the virtual array may be enough. Once the virtual arrays are defined, they will be used by ViPR Controller for the following:

- ViPR Controller will automatically determine which UCS compute systems are available to compute virtual pools based on the selection of virtual arrays.

- ViPR Controller will automatically determine which blades to use to provision hosts based on the virtual arrays and compute virtual pools.

ViPR Controller makes these determinations by calculating which UCS compute systems have visibility to storage through the networks in the virtual arrays.

If working with updating service profile templates

When using updating service profile templates, it is recommended to create a dedicated virtual array that:

- Includes only the specific storage arrays that are intended to be used with the updating service profile template.
- Includes only the specific storage ports that are intended to be used with the updating service profile template.

CHAPTER 11

Creating virtual pools for block storage

This chapter contains the following topics:

- [Create ViPR Controller Virtual Pools for Block Storage](#)..... 124
- [Important REST API calls to manage and configure block virtual pools](#)..... 132

Create ViPR Controller Virtual Pools for Block Storage

Create a virtual pool for block storage by specifying the criteria for provisioned block storage volumes. These criteria are also used to match physical storage pools which can be used for virtual pool volumes.

Before you begin

For information on virtual pools, see *ViPR Controller Concepts* on the [ViPR Controller Product Documentation Index](#).

- Authenticate with the ViPR REST API as a System Administrator. See [Authenticating with the REST API on page 9](#).
- For block virtual pool configuration requirements and recommendations, see *ViPR Controller Virtual Data Center Requirements and Information Guide* on the [ViPR Controller Product Documentation Index](#).
- You need the URN of the tenant to be assigned as the ACL to the virtual pool. You can determine the current tenant using `GET /tenant` or use `GET /tenants/bulk` to get a list of all of the tenants.

Table 12 Required Parameters in Request Payload

Required Parameter	Description
provisioning_type	<p>The provisioning type for the virtual pool.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • Thin • Thick <hr/> <p>Note</p> <p>For VMAX journal volumes, set <code>provisioning_type</code> to <code>Thick</code> as ViPR Controller does not pre-allocate the volumes.</p>
protocol	<p>The block protocols supported by the physical storage pools that will comprise the virtual pool.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • FC • iSCSI

There are additional optional parameters that can be set for RAID levels, high availability, SAN multi-path, and so on. A complete list of parameters for `POST /block/vpools` is described in the *EMC ViPR Controller REST API Reference* on the [ViPR Controller Product Documentation Index](#).

Table 13 Optional Request Payload Parameters Used in this Example

Optional Parameters used in this example	Description
name	Since the virtual pool will be the target for all provisioning operations, select a name that conveys some information about the type of storage

Table 13 Optional Request Payload Parameters Used in this Example (continued)

Optional Parameters used in this example	Description
	that it provides (its performance and protection levels) or how it should be used. For example, "gold", "tier1", or "backup", etc.
varray	The URNs of the virtual arrays to which the virtual pool is assigned.
description	Description for the virtual pool.
use_matched_pools	Determines if matched or valid assigned pools are returned from the command to retrieve the list of storage pools. Valid values: <ul style="list-style-type: none"> • true • false
protection	Parameters to support data protection: <ul style="list-style-type: none"> • continuous_copies = Parameter settings if you are using VMAX SRDF/A • recoverpoint = Parameter settings if you are using RecoverPoint • remote_copies = Parameter settings if you are using VMAX SRDF/S • snapshots <hr/> Note If you are setting continuous copies for VMAX SRDF/A, you must also set: <ul style="list-style-type: none"> • multi-volume_consistency = true • remote_copy_mode = ASYNCHRONOUS If you are setting remote copies for VMAX SRDF/S, you must also set: <ul style="list-style-type: none"> • multi-volume_consistency = false • remote_copy_mode = SYNCHRONOUS

Procedure

1. To create a block virtual pool, use `POST /block/vpools`. The representation of the new virtual pool is returned.

Request

```
POST https://<ViPR_Controller_VIP>:4443/block/vpools
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<block_vpool_create>
  <description>Basic Virtual Pool for Block Provisioning</
description>
  <name>basic_vpool</name>
  <max_paths>4</max_paths>
  <min_paths>1</min_paths>
  <paths_per_initiator>1</paths_per_initiator>
  <protection>
```

```

        <snapshots>
            <max_native_snapshots>10</max_native_snapshots>
        </snapshots>
    </protection>
    <protocols>
        <protocol>FC</protocol>
    </protocols>
    <provisioning_type>Thin</provisioning_type>
    <use_matched_pools>true</use_matched_pools>
    <varrays>
<varray>urn:storageos:VirtualArray:ac294bc7-0518-48dd-9f54-4980f001
2f86:vdcl</varray>
    </varrays>
</block_vpool_create>

```

Response

```

HTTP 200 OK
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<block_vpool>
    <creation_time>1423598650598</creation_time>
    <global>false</global>
    <id>urn:storageos:VirtualPool:ba745027-2385-437c-bf44-
a49fb32bb45c:vdcl</id>
    <inactive>false</inactive>
    <internal>false</internal>
    <link rel="self" href="/block/vpools/
urn:storageos:VirtualPool:ba745027-2385-437c-bf44-
a49fb32bb45c:vdcl"/>
    <name>basic_vpool</name>
    <remote>false</remote>
    <tags/>
    <vdc>
        <id>urn:storageos:VirtualDataCenter:99c5c185-
ebb0-468a-8fe7-3ea6e65b7b57:vdcl</id>
        <link rel="self" href="/vdc/
urn:storageos:VirtualDataCenter:99c5c185-
ebb0-468a-8fe7-3ea6e65b7b57:vdcl"/>
    </vdc>
    <assigned_storage_pools/>
    <description>Basic Virtual Pool for Block Provisioning</
description>
    <invalid_matched_pools/>
    <matched_storage_pools>
        <storage_pool>
            <id>urn:storageos:StoragePool:78caaf4a-673e-4580-ae41-
a545a45e5e28:vdcl</id>
            <link rel="self" href="/vdc/storage-systems/
urn:storageos:StorageSystem:56d8aa7e-45fe-4383-b49a-
fec72e9927d3:vdcl/storage-pools/urn:storageos:StoragePool:
78caaf4a-673e-4580-ae41-a545a45e5e28:vdcl"/>
        </storage_pool>
        <storage_pool>
            <id>urn:storageos:StoragePool:ad0dae57-d318-409c-
a859-6c9e59a28251:vdcl</id>
            <link rel="self" href="/vdc/storage-systems/
urn:storageos:StorageSystem:56d8aa7e-45fe-4383-b49a-
fec72e9927d3:vdcl/storage-pools/urn:storageos:StoragePool:ad0dae57-
d318-409c-a859-6c9e59a28251:vdcl"/>
        </storage_pool>
        <storage_pool>
            <id>urn:storageos:StoragePool:62d2e40d-
ced0-4114-8bb7-8d333e7ef878:vdcl</id>
            <link rel="self" href="/vdc/storage-systems/
urn:storageos:StorageSystem:56d8aa7e-45fe-4383-b49a-

```

```

fec72e9927d3:vdcl/storage-pools/urn:storageos:StoragePool:62d2e40d-
ced0-4114-8bb7-8d333e7ef878:vdcl"/>
  </storage_pool>
</matched_storage_pools>
<num_paths>4</num_paths>
<protocols>
  <protocol>FC</protocol>
</protocols>
<provisioning_type>Thin</provisioning_type>
<type>block</type>
<use_matched_pools>true</use_matched_pools>
<varrays>
  <varray>

<id>urn:storageos:VirtualArray:ac294bc7-0518-48dd-9f54-4980f0012f86
:vdcl</id>
  <link rel="self" href="/vdc/varrays/
urn:storageos:VirtualArray:ac294bc7-0518-48dd-9f54-4980f0012f86:vdcl
1"/>
  </varray>
</varrays>
<expandable>>false</expandable>
<fast_expansion>>false</fast_expansion>
<max_paths>4</max_paths>
<min_paths>1</min_paths>
<multi_volume_consistency>>false</multi_volume_consistency>
<paths_per_initiator>1</paths_per_initiator>
<protection>
  <snapshots>
    <max_native_snapshots>10</max_native_snapshots>
  </snapshots>
  <continuous_copies>
    <max_native_continuous_copies>0</
max_native_continuous_copies>
  </continuous_copies>
</protection>
<raid_levels/>
<unique_auto_tier_policy_names>>false</
unique_auto_tier_policy_names>
</block_vpool>

```

2. Tenant access to each virtual pool is controlled by an Access Control List (ACL). Only tenants that are included in the virtual pool's ACL are permitted access to that virtual pool. If no ACL is set, the virtual pool can be accessed by all tenants. Apply an ACL to the virtual pool using `PUT /block/vpools/{vpool_urn}/acl`. The updated ACL is returned.

Request

```

PUT https://<ViPR_Controller_VIP>:4443/block/vpools/
urn:storageos:VirtualPool:ba745027-2385-437c-bf44-
a49fb32bb45c:vdcl/acl
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<acl_assignment_changes>
  <add>
    <privilege>USE</privilege>
    <tenant>urn:storageos:TenantOrg:c104b49e-4e6b-449c-b92b-
a19b247a613b:global</tenant>
  </add>
</acl_assignment_changes>

```

Response

```
HTTP 200 OK
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<acl_assignments>
  <acl_assignment>
    <privilege>USE</privilege>
    <tenant>urn:storageos:TenantOrg:c104b49e-4e6b-449c-b92b-
a19b247a613b:global</tenant>
  </acl_assignment>
</acl_assignments>
```

3. ViPR Controller automatically matches existing physical pools on the ViPR Controller-managed storage systems to the virtual pool characteristics. You have the option of allowing ViPR Controller to automatically associate the matching physical pools to the virtual pool you created, or you can manually select a subset of the matching physical pools to associate to the virtual pool. To manually select a subset of the matching physical pools:

- a. POST /block/vpools/matching-pools is used to get a list of storage pools that match a given set of criteria. In this example, the criteria that are sent in the request payload match the properties of the virtual pool that was just created.

Request

```
POST https://<ViPR_Controller_VIP>:4443/block/vpools/matching-
pools
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<block_vpool_create>
  <description>Basic Virtual Pool for Block Provisioning</
description>
  <name>basic_vpool</name>
  <num_paths>1</num_paths>
  <protection>
    <snapshots>
      <max_native_snapshots>10</max_native_snapshots>
    </snapshots>
  </protection>
  <protocols>
    <element>FC</element>
  </protocols>
  <provisioning_type>Thin</provisioning_type>
  <use_matched_pools>true</use_matched_pools>
  <varrays>
    <varray>urn:storageos:VirtualArray:ac294bc7-0518-48dd-9f54-4980f
0012f86:vdcl</varray>
  </varrays>
</block_vpool_create>
```

Response

```
HTTP 200 OK
Content-Type: application/xml
<?xml version="1.0" encoding="UTF-8"?>
<storage_pools>
  <storage_pool>
    <id>urn:storageos:StoragePool:62d2e40d-
ced0-4114-8bb7-8d333e7ef878:vdcl</id>
    <link rel="self" href="/vdc/storage-systems/
urn:storageos:StorageSystem:56d8aa7e-45fe-4383-b49a-
```



```

fec72e9927d3:vdc1/storage-pools/urn:storageos:StoragePool:
62d2e40d-ced0-4114-8bb7-8d333e7ef878:vdc1"/>
  <name>SYMMETRIX+000198700420+POOL+TP+3R5-A</name>
</storage_pool>
  <storage_pool>
    <id>urn:storageos:StoragePool:9e551963-d596-48ae-
b208-1b870126d195:vdc1</id>
    <link rel="self" href="/vdc/storage-systems/
urn:storageos:StorageSystem:56d8aa7e-45fe-4383-b49a-
fec72e9927d3:vdc1/storage-pools/urn:storageos:StoragePool:
9e551963-d596-48ae-b208-1b870126d195:vdc1"/>
    <name>SYMMETRIX+000198700420+POOL+TP+M3</name>
  </storage_pool>
  <storage_pool>
    <id>urn:storageos:StoragePool:ad0dae57-d318-409c-
a859-6c9e59a28251:vdc1</id>
    <link rel="self" href="/vdc/storage-systems/
urn:storageos:StorageSystem:56d8aa7e-45fe-4383-b49a-
fec72e9927d3:vdc1/storage-pools/
urn:storageos:StoragePool:ad0dae57-d318-409c-
a859-6c9e59a28251:vdc1"/>
    <name>SYMMETRIX+000198700420+POOL+TP+M2</name>
  </storage_pool>
  <storage_pool>
    <id>urn:storageos:StoragePool:78caaf4a-673e-4580-ae41-
a545a45e5e28:vdc1</id>
    <link rel="self" href="/vdc/storage-systems/
urn:storageos:StorageSystem:56d8aa7e-45fe-4383-b49a-
fec72e9927d3:vdc1/storage-pools/urn:storageos:StoragePool:
78caaf4a-673e-4580-ae41-a545a45e5e28:vdc1"/>
    <name>SYMMETRIX+000198700420+POOL+TP+data</name>
  </storage_pool>
  <storage_pool>
    <id>urn:storageos:StoragePool:38e2a0c2-89c9-4ec7-
ad9b-29420e464012:vdc1</id>
    <link rel="self" href="/vdc/storage-systems/
urn:storageos:StorageSystem:56d8aa7e-45fe-4383-b49a-
fec72e9927d3:vdc1/storage-pools/urn:storageos:StoragePool:
38e2a0c2-89c9-4ec7-ad9b-29420e464012:vdc1"/>
    <name>SYMMETRIX+000198700420+POOL+TP+t3</name>
  </storage_pool>
  <storage_pool>
    <id>urn:storageos:StoragePool:dbdcd42a-
b0a4-49b4-8a6e-204a09b8b1ef:vdc1</id>
    <link rel="self" href="/vdc/storage-systems/
urn:storageos:StorageSystem:56d8aa7e-45fe-4383-b49a-
fec72e9927d3:vdc1/storage-pools/
urn:storageos:StoragePool:dbdcd42a-
b0a4-49b4-8a6e-204a09b8b1ef:vdc1"/>
    <name>SYMMETRIX+000198700420+POOL+TP+3R5-B</name>
  </storage_pool>
</storage_pools>

```

- b. Using the URN returned for the new virtual pool returned from the POST `/block/vpools` request, assign one or more of the storage pools that match the properties of the virtual pool by sending a PUT `/block/vpools/{identifier}/assign-matched-pools` request. The representation of the updated virtual pool is returned.

Request

```

PUT https://<ViPR_Controller_VIP>:4443/block/vpools/
urn:storageos:VirtualPool:ba745027-2385-437c-bf44-
a49fb32bb45c:vdc1/assign-matched-pools
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

```

```
<vpool_pool_update>
  <assigned_pool_changes>
    <add>
      <storage_pool>
        <storage_pool>urn:storageos:StoragePool:62d2e40d-ced0-4114-8bb7-8d333e7ef878:vdc1</storage_pool>
        <storage_pool>urn:storageos:StoragePool:78caaf4a-673e-4580-ae41-a545a45e5e28:vdc1</storage_pool>
      </storage_pool>
      <storage_pool>urn:storageos:StoragePool:ad0dae57-d318-409c-a859-6c9e59a28251:vdc1</storage_pool>
    </storage_pool>
  </add>
</assigned_pool_changes>
</vpool_pool_update>
```

Response

```
HTTP 200 OK
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8"?>
<block_vpool>
  <creation_time>1379202604198</creation_time>
  <global>>false</global>
  <id>urn:storageos:VirtualPool:6cd9f843-1b41-4b9f-8eb1-a26797d0a268:vdc1</id>
  <inactive>>false</inactive>
  <internal>>false</internal>
  <link rel="self" href="/block/vpools/urn:storageos:VirtualPool:6cd9f843-1b41-4b9f-8eb1-a26797d0a268:vdc1"/>
  <name>basic_vpool</name>
  <remote>>false</remote>
  <tags />
  <vdc>
    <id>urn:storageos:VirtualDataCenter:030618c2-c6b2-40b0-a105-6b669983f58f:vdc1</id>
    <link rel="self" href="/vdc/urn:storageos:VirtualDataCenter:030618c2-c6b2-40b0-a105-6b669983f58f:vdc1"/>
  </vdc>
  <assigned_storage_pools>
    <storage_pool>
      <id>urn:storageos:StoragePool:78caaf4a-673e-4580-ae41-a545a45e5e28:vdc1</id>
      <link rel="self" href="/vdc/storage-systems/urn:storageos:StorageSystem:56d8aa7e-45fe-4383-b49a-fec72e9927d3:vdc1/storage-pools/urn:storageos:StoragePool:78caaf4a-673e-4580-ae41-a545a45e5e28:vdc1"/>
    </storage_pool>
    <storage_pool>
      <id>urn:storageos:StoragePool:ad0dae57-d318-409c-a859-6c9e59a28251:vdc1</id>
      <link rel="self" href="/vdc/storage-systems/urn:storageos:StorageSystem:56d8aa7e-45fe-4383-b49a-fec72e9927d3:vdc1/storage-pools/urn:storageos:StoragePool:ad0dae57-d318-409c-a859-6c9e59a28251:vdc1"/>
    </storage_pool>
    <storage_pool>
      <id>urn:storageos:StoragePool:62d2e40d-ced0-4114-8bb7-8d333e7ef878:vdc1</id>
      <link rel="self" href="/vdc/storage-systems/urn:storageos:StorageSystem:56d8aa7e-45fe-4383-b49a-fec72e9927d3:vdc1/storage-pools/urn:storageos:StoragePool:62d2e40d-ced0-4114-8bb7-8d333e7ef878:vdc1"/>
    </storage_pool>
  </assigned_storage_pools>
</block_vpool>
```

```

    </storage_pool>
  </assigned_storage_pools>
  <description>Basic Virtual Pool for Block Provisioning</
description>
  <invalid_matched_pools />
  <matched_storage_pools>
    <storage_pool>
      <id>urn:storageos:StoragePool:78caaf4a-673e-4580-
ae41-a545a45e5e28:vdcl</id>
      <link rel="self" href="/vdc/storage-systems/
urn:storageos:StorageSystem:56d8aa7e-45fe-4383-b49a-
fec72e9927d3:vdcl/storage-pools/urn:storageos:StoragePool:
78caaf4a-673e-4580-ae41-a545a45e5e28:vdcl"/>
    </storage_pool>
    <storage_pool>
      <id>urn:storageos:StoragePool:ad0dae57-d318-409c-
a859-6c9e59a28251:vdcl</id>
      <link rel="self" href="/vdc/storage-systems/
urn:storageos:StorageSystem:56d8aa7e-45fe-4383-b49a-
fec72e9927d3:vdcl/storage-pools/
urn:storageos:StoragePool:ad0dae57-d318-409c-
a859-6c9e59a28251:vdcl"/>
    </storage_pool>
    <storage_pool>
      <id>urn:storageos:StoragePool:62d2e40d-
ced0-4114-8bb7-8d333e7ef878:vdcl</id>
      <link rel="self" href="/vdc/storage-systems/
urn:storageos:StorageSystem:56d8aa7e-45fe-4383-b49a-
fec72e9927d3:vdcl/storage-pools/urn:storageos:StoragePool:
62d2e40d-ced0-4114-8bb7-8d333e7ef878:vdcl"/>
    </storage_pool>
  </matched_storage_pools>
  <num_paths>1</num_paths>
  <num_resources>4</num_resources>
  <protocols>
    <protocol>FC</protocol>
  </protocols>
  <provisioning_type>Thin</provisioning_type>
  <system_type>NONE</system_type>
  <type>block</type>
  <use_matched_pools>true</use_matched_pools>
  <varrays>
    <varray>
      <id>urn:storageos:VirtualArray:dc09417d-9028-40de-8bff-74574996f
2d5:vdcl</id>
      <link rel="self" href="vdc/varrays/
urn:storageos:VirtualArray:dc09417d-9028-40de-8bff-74574996f2d5:
vdcl"/>
    </varray>
  </varrays>
  <drive_type>NONE</drive_type>
  <expandable>true</expandable>
  <fast_expansion>false</fast_expansion>
  <high_availability/>
  <max_paths>2</max_paths>
  <min_paths>1</min_paths>
  <multi_volume_consistency>true</multi_volume_consistency>
  <paths_per_initiator>1</paths_per_initiator>
  <protection>
    <continuous_copies>
      <max_native_continuous_copies>0</
max_native_continuous_copies>
    </continuous_copies>
    <snapshots>
      <max_native_snapshots>10</max_native_snapshots>
    </snapshots>
  </protection>
  <protocols>

```

```

        <element>FC</element>
    </protocols>
    <provisioning_type>Thin</provisioning_type>
    <raid_levels />
    <thin_volume_preallocation_percentage>0</
thin_volume_preallocation_percentage>
    <unique_auto_tier_policy_names>>false</
unique_auto_tier_policy_names>
</block_vpool>

```

Important REST API calls to manage and configure block virtual pools

The table shows some important APIs that are used to manage and configure block virtual pools.

Table 14 REST API calls to manage and configure block virtual pools

REST API call	Description
POST /block/vpools/{id}/vpool-change/vpool	Returns all potential virtual pools, which supported the given virtual pool change operation for a virtual pool change of the volumes specified in the request.
POST /block/vpools/matching-pools	You provide a set of block virtual pool attributes, and the method returns block virtual pools that match the given set of attributes. You can use this API to find matching pools before creating a block virtual pool. Sample request payload: <pre> <block_vpool_create> <use_matched_pools>>true</use_matched_pools> <system_type>vnxblock</system_type> <provisioning_type>thin</provisioning_type> </block_vpool_create> </pre>
GET /block/vpools	List all block virtual pools.
GET /block/vpools/{id}	Show the details of the specified block virtual pool.
PUT /block/vpools/{id}	Update the specified block virtual pool. <hr/> Note <hr/> No associated resources such as volumes or snapshots should exist. <hr/> Sample request payload: <pre> <block_vpool_update> <protection> <continuous_copies> <max_native_continuous_copies>1</ max_native_continuous_copies> </continuous_copies> </protection> </block_vpool_update> </pre>

Table 14 REST API calls to manage and configure block virtual pools (continued)

REST API call	Description
GET /block/vpools/{id}/varrays/{varrayid}/capacity	Gets storage capacity information for the specified virtual pool and neighborhood instances. The method returns set of metrics for capacity available for block storage provisioning: <p>free_gb free storage capacity</p> <p>used_gb used storage capacity</p> <p>provisioned_gb subscribed storage capacity (may be larger than usable capacity)</p> <p>percent_used percent of usable capacity which is used - percent_subscribed</p> <p>percent_subscribed percent of usable capacity which is subscribed (may be more than 100)</p>
GET /block/vpools/{id}/quota	Show the quota and available capacity before the quota is exhausted.
PUT /block/vpools/{id}/quota	Updates quota and available capacity before quota is exhausted. <pre><quota_update> <quota_enabled>true</quota_enabled> <quota_gb>2</quota_gb> </quota_update></pre>
PUT /block/vpools/{id}/acl	Add or remove individual block store virtual pool ACL entry(s). Request body must include at least one add or remove operation. <p>Sample request payload:</p> <pre><acl_assignment_changes> <add> <privilege>USE</privilege> <tenant>urn:storageos:TenantOrg:dbeb4135-e297-40d9- a5d4-9b40c73bdb4b:</tenant> </add> </acl_assignment_changes></pre>

CHAPTER 12

Creating virtual pools for file storage

This chapter contains the following topics:

- [Create ViPR Controller virtual pools for file storage](#)..... 136

Create ViPR Controller virtual pools for file storage

Create a virtual pool for file storage by specifying the criteria for provisioned file storage volumes. These criteria are also used to match physical storage pools which can be used for virtual pool volumes.

Before you begin

- Authenticate with the ViPR REST API as a System Administrator. See [Authenticating with the REST API on page 9](#).
- For information on virtual pools, see *ViPR Controller Concepts* on the [ViPR Controller Product Documentation Index](#).
- For file virtual pool configuration requirements and recommendations, see *ViPR Controller Virtual Data Center Requirements and Information Guide* on the [ViPR Controller Product Documentation Index](#).
- You need the URN of the tenant to be assigned as the ACL to the virtual pool. You can determine the current tenant using `GET /tenant` or use `GET /tenants/bulk` to get a list of all of the tenants.

Table 15 Additional Required Parameters for the Request Payload

Required Parameter	Description
provisioning_type	The provisioning type for the virtual pool. Valid values: <ul style="list-style-type: none"> • Thin • Thick
protocol	The file protocols supported by the physical storage pools that will comprise the virtual pool. Valid values: <ul style="list-style-type: none"> • CIFS • NFS
long_term_retention	Valid values: <ul style="list-style-type: none"> • True • False
name	Since the virtual pool will be the target for all provisioning operations, select a name that conveys some information about the type of storage that it provides (its performance and protection levels) or how it should be used. For example, "gold", "tier1", or "backup", etc.
varray	The URNs of the virtual arrays to which the virtual pool is assigned.
description	Description for the virtual pool.
use_matched_pools	Determines if matched or valid assigned pools are returned from the command to retrieve the list of storage pools. Valid values: <ul style="list-style-type: none"> • true • false

Table 15 Additional Required Parameters for the Request Payload (continued)

Required Parameter	Description
protection	Parameters if you are using a data protection system: <ul style="list-style-type: none"> • snapshots

A complete list of parameters for POST /file/vpools is described in the *EMC ViPR Controller REST API Reference* on the [ViPR Controller Product Documentation Index](#).

Procedure

1. Create a virtual pool for file storage by sending a POST /file/vpools. The representation of the new file virtual pool is returned.

Request

```
POST https://<ViPR_Controller_VIP>:4443/file/vpools
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<file_vpool_create>
  <name>Isilon VPool</name>
  <description>Isilon VPool</description>
  <varrays>
    <varray>urn:storageos:VirtualArray:f66ee94d-12d2-4a99-8514-
f74beb4844e6:vdcl</varray>
  </varrays>
  <provisioning_type>Thick</provisioning_type>
  <protocols>
    <protocol>NFS</protocol>
  </protocols>
  <protection>
    <snapshots>
      <max_native_snapshots>0</max_native_snapshots>
    </snapshots>
  </protection>
  <use_matched_pools>true</use_matched_pools>
  <long_term_retention>>false</long_term_retention>
</file_vpool_create>
```

Response

```
HTTP 200 OK
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<file_vpool>
  <creation_time>1399901093414</creation_time>
  <global>>false</global>
  <id>urn:storageos:VirtualPool:07f9b2f4-baa2-4136-b0f7-
cfb6572c853c:vdcl</id>
  <inactive>>false</inactive>
  <internal>>false</internal>
  <link rel="self" href="/file/vpools/urn:storageos:VirtualPool:
07f9b2f4-baa2-4136-b0f7-cfb6572c853c:vdcl"/>
  <name>Isilon VPool</name>
  <remote>>false</remote>
  <tags/>
  <vdc>
    <id>urn:storageos:VirtualDataCenter:ed581473-3453-4c05-
bd61-2068ad66942a:vdcl</id>
    <link rel="self" href="/vdc/
```

```
urn:storageos:VirtualDataCenter:ed581473-3453-4c05-
bd61-2068ad66942a:vdcl"/>
</vdc>
<assigned_storage_pools/>
<description>Isilon VPool</description>
<invalid_matched_pools/>
<matched_storage_pools>
  <storage_pool>
    <id>urn:storageos:StoragePool:6a3eb06f-4670-4ee3-b812-
dc04771c5118:vdcl</id>
    <link_rel="self" href="/vdc/storage-systems/
urn:storageos:StorageSystem:de38e08a-8cf4-4767-8291-5765bba63091:vd
c1/storage-pools/urn:storageos:StoragePool:6a3eb06f-4670-4ee3-b812-
dc04771c5118:vdcl"/>
  </storage_pool>
</matched_storage_pools>
<num_resources>1</num_resources>
<protocols>
  <protocol>NFS</protocol>
</protocols>
<provisioning_type>Thick</provisioning_type>
<system_type>NONE</system_type>
<type>file</type>
<use_matched_pools>true</use_matched_pools>
<varrays>
  <varray>
    <id>urn:storageos:VirtualArray:f66ee94d-12d2-4a99-8514-
f74beb4844e6:vdcl</id>
    <link_rel="self" href="/vdc/varrays/
urn:storageos:VirtualArray:f66ee94d-12d2-4a99-8514-
f74beb4844e6:vdcl"/>
  </varray>
</varrays>
<long_term_retention>>false</long_term_retention>
<protection>
  <snapshots>
    <max_native_snapshots>0</max_native_snapshots>
  </snapshots>
</protection>
</file_vpool>
```

2. Tenant access to each virtual pool is controlled by an Access Control List (ACL). Only tenants that are included in the virtual pool's ACL are permitted access to that virtual pool. If no ACL is set, the virtual pool can be accessed by all tenants. Apply an ACL to the virtual pool by sending a PUT request to `/file/vpools/{vpool_urn}/acl`. The updated ACL is returned.

Request

```
PUT https://<ViPR_Controller_VIP>:4443/file/vpools/
urn:storageos:VirtualPool:07f9b2f4-baa2-4136-b0f7-
cfb6572c853c:vdcl/acl
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<acl_assignment_changes>
  <add>
    <privilege>USE</privilege>
    <tenant>urn:storageos:TenantOrg:757053b7-
b952-41b9-83e1-44204b67a368:vdcl</tenant>
  </add>
</acl_assignment_changes>
```

Response

```
HTTP 200 OK
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8"?>
<acl_assignments>
  <acl_assignment>
    <privilege>USE</privilege>
    <tenant>urn:storageos:TenantOrg:757053b7-
b952-41b9-83e1-44204b67a368:vdc1</tenant>
  </acl_assignment>
</acl_assignments>
```

3. ViPR Controller automatically matches existing physical pools on the ViPR Controller-managed storage systems to the virtual pool characteristics. You have the option of allowing ViPR Controller to automatically associate the matching physical pools to the virtual pool you created, or you can manually select a subset of the matching physical pools to associate to the virtual pool. To manually select a subset of matching physical pools:

- a. POST /file/vpools/matching-pools is used to get a list of storage pools that match a given set of criteria. In this example, the criteria that are sent in the request payload match the properties of the virtual pool that was just created.

Request

```
POST https://<ViPR_Controller_VIP>:4443/file/vpools/matching-
pools
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<file_vpool_create>
  <name>Isilon VPool</name>
  <description>Isilon VPool</description>
  <varrays>

  <varray>urn:storageos:VirtualArray:f66ee94d-12d2-4a99-8514-
f74beb4844e6:vdc1</varray>
</varrays>
  <provisioning_type>Thick</provisioning_type>
  <protocols>
    <protocol>NFS</protocol>
  </protocols>
  <protection>
    <snapshots>
      <max_native_snapshots>0</max_native_snapshots>
    </snapshots>
  </protection>
  <use_matched_pools>true</use_matched_pools>
</file_vpool_create>
```

Response

```
HTTP 200 OK
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<storage_pools>
  <storage_pool>
    <id>urn:storageos:StoragePool:6a3eb06f-4670-4ee3-b812-
dc04771c5118:vdc1</id>
    <link rel="self" href="/vdc/storage-systems/
urn:storageos:StorageSystem:de38e08a-8cf4-4767-8291-5765bba63091
:vdc1/storage-pools/urn:storageos:StoragePool:
```

```
6a3eb06f-4670-4ee3-b812-dc04771c5118:vdcl"/>
  <name>ISILON+6805ca00ad44cad54252f514e871a3c03333+POOL
+x200_5.5tb_200gb-ssd_6gb</name>
</storage_pool>
</storage_pools>
```

- b. Using the URN returned for the new virtual pool returned from the POST /file/vpools request, assign one or more of the storage pools that match the properties of the virtual pool by sending a PUT/file/vpools/{identifier}/assign-matched-pools request. The representation of the updated virtual pool is returned.

Request

```
PUT https://<ViPR_Controller_VIP>:4443/file/vpools/
urn:storageos:VirtualPool:07f9b2f4-baa2-4136-b0f7-
cfb6572c853c:vdcl/assign-matched-pools
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<vpool_pool_update>
  <assigned_pool_changes>
    <add>
      <storage_pool>
        <storage_pool>urn:storageos:StoragePool:
6a3eb06f-4670-4ee3-b812-dc04771c5118:vdcl</storage_pool>
      </storage_pool>
    </add>
  </assigned_pool_changes>
</vpool_pool_update>
```

Response

```
HTTP 200 OK
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<file_vpool>
  <creation_time>1399901093414</creation_time>
  <global>false</global>
  <id>urn:storageos:VirtualPool:07f9b2f4-baa2-4136-b0f7-
cfb6572c853c:vdcl</id>
  <inactive>false</inactive>
  <internal>false</internal>
  <link rel="self" href="/file/vpools/
urn:storageos:VirtualPool:07f9b2f4-baa2-4136-b0f7-
cfb6572c853c:vdcl"/>
  <name>Isilon VPool</name>
  <remote>false</remote>
  <tags/>
  <vdc>
    <id>urn:storageos:VirtualDataCenter:ed581473-3453-4c05-
bd61-2068ad66942a:vdcl</id>
    <link rel="self" href="/vdc/
urn:storageos:VirtualDataCenter:ed581473-3453-4c05-
bd61-2068ad66942a:vdcl"/>
  </vdc>
  <assigned_storage_pools>
    <storage_pool>
      <id>urn:storageos:StoragePool:6a3eb06f-4670-4ee3-
b812-dc04771c5118:vdcl</id>
      <link rel="self" href="/vdc/storage-systems/
urn:storageos:StorageSystem:de38e08a-8cf4-4767-8291-5765bba63091
:vdcl/storage-pools/urn:storageos:StoragePool:
6a3eb06f-4670-4ee3-b812-dc04771c5118:vdcl"/>
```

```

    </storage_pool>
  </assigned_storage_pools>
  <description>Isilon VPool</description>
  <invalid_matched_pools/>
  <matched_storage_pools>
    <storage_pool>
      <id>urn:storageos:StoragePool:6a3eb06f-4670-4ee3-
b812-dc04771c5118:vdcl</id>
      <link rel="self" href="/vdc/storage-systems/
urn:storageos:StorageSystem:de38e08a-8cf4-4767-8291-5765bba63091
:vdcl/storage-pools/urn:storageos:StoragePool:
6a3eb06f-4670-4ee3-b812-dc04771c5118:vdcl"/>
    </storage_pool>
  </matched_storage_pools>
  <protocols>
    <protocol>NFS</protocol>
  </protocols>
  <provisioning_type>Thick</provisioning_type>
  <system_type>NONE</system_type>
  <type>file</type>
  <use_matched_pools>true</use_matched_pools>
  <varrays>
    <varray>

<id>urn:storageos:VirtualArray:f66ee94d-12d2-4a99-8514-
f74beb4844e6:vdcl</id>
    <link rel="self" href="/vdc/varrays/
urn:storageos:VirtualArray:f66ee94d-12d2-4a99-8514-
f74beb4844e6:vdcl"/>
    </varray>
  </varrays>
  <long_term_retention>>false</long_term_retention>
  <protection>
    <snapshots>
      <max_native_snapshots>0</max_native_snapshots>
    </snapshots>
  </protection>
</file_vpool>

```


CHAPTER 13

Creating virtual pools for Vblock Compute Systems

This chapter contains the following topics:

- [Create a Compute Virtual Pool using the REST API](#)..... 144
- [UCS blades and service profile templates](#).....146
- [Additional REST API calls to manage your Compute Virtual Pools](#)..... 147

Create a Compute Virtual Pool using the REST API

Compute virtual pools are a pool of compute system blades. When a Vblock System Service is run, ViPR Controller pulls required compute resources from the selected compute virtual pool.

Before you begin

- Authenticate with the ViPR REST API as a System Administrator. See [Authenticating with the REST API on page 9](#)
- For a compute virtual pool based on UCS compute systems to be used by ViPR Controller for provisioning, at least one service profile template must be selected in the compute virtual pool. Review the ViPR Controller requirements for service profile templates, and discuss with your UCS administrator which service profile template should be used by ViPR Controller to provision the Vblock system. See [ViPR Controller requirements for service profile templates on page 146](#).
- `system_type` - `Cisco_UCSM` is one example.
- `varrays` - URN of the varrays. You can find the ID of the varray using `GET /vdc/varrays`
- `service_profile_template` - the URNs of the service profile templates which you want associated with this virtual pool. You can find the URNs of the service profile templates using `GET /vdc/compute-systems/{ID}`. You can include more than one service profile template but you can only use one service profile template per compute system.
- Optional: Minimum and maximum values to eliminate blades, which do not match the criteria. The range of values is 1 - 65535. When no minimum is set, ViPR Controller defaults to zero. There is no maximum value. The maximum is unlimited when it is not set. For example:
 - If processors are set with a minimum of 6 and no maximum, then only blades with 6 or more processors will be available to use in the compute virtual pool.
 - If processors are set with no minimum, and a maximum of 16, then any blade with 16 or less processors will be available to use in the compute virtual pool.
 - If no minimum or maximum value is set for processors ViPR Controller will include available blades with any number of processors in the virtual pool.
 - `min_processors` and `max_processors`
 - `min_total_cores` and `max_total_cores`
 - `min_total_threads` and `max_total_threads`
 - `min_cpu_speed` and `max_cpu_speed`
 - `min_memory` and `max_memory`
 - `min_nics` and `max_nics`
 - `min_hbas` and `max_hbas`
- The *EMC ViPR Controller REST API Reference* on the [ViPR Controller Product Documentation Index](#) provides a description and complete list of parameters for the REST API methods used in this article.

Procedure

1. Create a compute virtual pool using `POST /compute/vpools`. The new compute vpool is returned.

Request

```

POST https://<ViPR_Controller_VIP>:4443/compute/vpools
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<compute_vpool_create>
  <name>compute_vpool_west1</name>
  <description>vpool for west</description>
  <system_type>Cisco_UCSM</system_type>
  <min_processors>3</min_processors>
  <varrays>
    <varray>urn:storageos:VirtualArray:ac294bc7-0518-48dd-9f54-4980f001
    2f86:vdcl</varray>
  </varrays>
  <use_matched_elements>true</use_matched_elements>
  <service_profile_templates>
    <service_profile_template>urn:storageos:UCSServiceProfileTemplate:
    6c6236dd-e98d-47da-b2e6-749c09554ccd:vdcl</
    service_profile_template>
  </service_profile_templates>
</compute_vpool_create>

```

Response

```

HTTP 200 OK
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<compute_vpool>
  <creation_time>1422389775019</creation_time>
  <global>>false</global>
  <id>urn:storageos:ComputeVirtualPool:75ec968b-8001-4889-
  abd7-80090c4010a9:vdcl</id>
  <inactive>>false</inactive>
  <internal>>false</internal>
  <name>compute_vpool_west1</name>
  <remote>>false</remote>
  <tags/>
  <vdc>
    <id>urn:storageos:VirtualDataCenter:99c5c185-
    ebb0-468a-8fe7-3ea6e65b7b57:vdcl</id>
    <link rel="self" href="/vdc/urn:storageos:VirtualDataCenter:
    99c5c185-ebb0-468a-8fe7-3ea6e65b7b57:vdcl"/>
  </vdc>
  <available_matched_compute_elements/>
  <description>vpool for west</description>
  <in_use>>false</in_use>
  <matched_compute_elements/>
  <min_processors>3</min_processors>
  <service_profile_templates>
    <service_profile_template>
      <id>urn:storageos:UCSServiceProfileTemplate:6c6236dd-
      e98d-47da-b2e6-749c09554ccd:vdcl</id>
      <name>ST-ServerTemplate</name>
    </service_profile_template>
  </service_profile_templates>
  <system_type>Cisco_UCSM</system_type>
  <use_matched_elements>true</use_matched_elements>
  <varrays>
    <varray>
      <id>urn:storageos:VirtualArray:ac294bc7-0518-48dd-9f54-4980f0012f86
      :vdcl</id>
      <link rel="self" href="/vdc/varrays/

```

```
urn:storageos:VirtualArray:ac294bc7-0518-48dd-9f54-4980f0012f86:vdc
1"/>
  </varray>
</varrays>
</compute_vpool>
```

UCS blades and service profile templates

Some compute systems are blade servers with hardware virtualization. This means that many properties of the servers that were traditionally burned into the hardware are actually soft settable. The UCS uses the concept of "service profiles" as a container for all the required settings to stand up a blade.

Service profiles are associated to blades to assign the required settings. Additionally, the UCS has the concept of, "service profile templates, (SPTs)," and "updating Service Profile Templates (uSPTs)," that must be set up by UCS administrators. These service profile templates can be used by non-admin users to create the service profiles that turn a blade into a host.

ViPR Controller does not perform the functions of the UCS administrator, rather ViPR Controller utilizes service profile templates to assign the required properties to blades. A UCS administrator will need to create service profile templates that ViPR Controller can use to provision servers and hosts.

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.

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.

Additional REST API calls to manage your Compute Virtual Pools

The table shows some additional APIs that are used to manage your Compute Virtual Pools.

Table 16 REST API calls to manage your Compute Virtual Pools

ViPR Controller REST API Call	Description
GET /compute/vpools/{ID}	Get the details of a compute virtual pools .
GET /compute/vpools	Get a list of the names, IDs, and links of all compute virtual pools.
PUT /compute/vpools/{ID}	Update a compute virtual pool.

Table 16 REST API calls to manage your Compute Virtual Pools (continued)

ViPR Controller REST API Call	Description
	<p>Note</p> <p>Once a compute virtual pool has been used by ViPR Controller, only some of the attributes can be changed. You will receive an error message in the response if you try to update a field that cannot be changed.</p>
POST /compute/vpools/{ID}/deactivate	Delete a compute virtual pools.
POST /compute/vpools/matching-compute-elements	Get compute elements that match the compute virtual pool criteria.
GET /compute/vpools/search?parameter={search_parameter}	<p>Search for a compute virtual pool . For example:</p> <pre>GET /compute/vpools/search?name=cvpool1</pre>

CHAPTER 14

Creating virtual pools for object storage

This chapter contains the following topics:

- [Create ViPR Controller Virtual Pools for Object Storage](#)..... 150

Create ViPR Controller Virtual Pools for Object Storage

Create a virtual pool for object storage by specifying the criteria for provisioned object storage buckets. These criteria are also used to match physical storage pools which can be used for the buckets.

Before you begin

- Authenticate with the ViPR REST API as a System Administrator. See [Authenticating with the REST API on page 9](#).
- For object virtual pool configuration requirements and recommendations, see *ViPR Controller Virtual Data Center Requirements and Information Guide* on the [ViPR Controller Product Documentation Index](#).
- You need the URN of the tenant to be assigned as the ACL to the virtual pool. You can determine the current tenant using `GET /tenant` or use `GET /tenants/bulk` to get a list of all of the tenants.

Table 17 Required Parameters in Request Payload

Required Parameter	Description
system_type	The system type for the virtual pool. Valid value: <ul style="list-style-type: none"> • ecs
protocol	The object protocols supported by the physical storage pools that will comprise the virtual pool. Valid values: <ul style="list-style-type: none"> • Swift • Atmos • S3
name	Since the virtual pool will be the target for all provisioning operations, select a name that conveys some information about the type of storage that it provides (its performance and protection levels) or how it should be used. For example, "gold", "tier1", or "backup", etc.
varray	The URNs of the virtual arrays to which the virtual pool is assigned.
description	Description for the virtual pool.
use_matched_pools	Determines if matched or valid assigned pools are returned from the command to retrieve the list of storage pools. Valid values: <ul style="list-style-type: none"> • true • false

A complete list of parameters for `POST /object/vpools` is described in the *EMC ViPR Controller REST API Reference* on the [ViPR Controller Product Documentation Index](#).

Procedure

1. To create an object virtual pool, use `POST /object/vpools`. The representation of the new virtual pool is returned.

Request

```

POST https://<ViPR_Controller_VIP>:4443/object/vpools
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<object_vpool_create>
  <name>vpool1234</name>
  <description>vpool1234 description</description>
  <protocols>
    <protocol>S3</protocol>
  </protocols>
  <max_retention>100</max_retention>
  <use_matched_pools>true</use_matched_pools>
  <varrays>
    <varray>urn:storageos:VirtualArray:
588568c8-5330-4e62-8fa4-bcb7f1ebff27:vdcl</varray>
  </varrays>
</object_vpool_create>

```

Response

```

HTTP 200 OK
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<object_vpool>
  <creation_time>1445956741889</creation_time>
  <global>false</global>
  <id>urn:storageos:VirtualPool:dc96b4d8-d60e-4520-8029-
c903aac06974:vdcl</id>
  <inactive>false</inactive>
  <internal>false</internal>
  <link rel="self" href="/object/vpools/
urn:storageos:VirtualPool:dc96b4d8-d60e-4520-8029-
c903aac06974:vdcl"/>
  <name>vpool1234</name>
  <remote>false</remote>
  <tags/>
  <vdc>
    <id>urn:storageos:VirtualDataCenter:8d832457-b944-4140-
bec6-7ab8649c1d1c:vdcl</id>
    <link rel="self" href="/vdc/
urn:storageos:VirtualDataCenter:8d832457-b944-4140-
bec6-7ab8649c1d1c:vdcl"/>
  </vdc>
  <assigned_storage_pools/>
  <description>vpool1234 description</description>
  <invalid_matched_pools/>
  <matched_storage_pools/>
  <protocols>
    <protocol>S3</protocol>
  </protocols>
  <provisioning_type>Thin</provisioning_type>
  <type>object</type>
  <use_matched_pools>true</use_matched_pools>
  <varrays>
    <varray>
<id>urn:storageos:VirtualArray:ff6f40c2-0612-416d-95f3-33a68fadb010:
vdcl</id>
    <link rel="self" href="/vdc/varrays/
urn:storageos:VirtualArray:ff6f40c2-0612-416d-95f3-33a68fadb010:vdcl
1"/>
  </varray>
</varrays>

```

```
</object_vpool>
```

2. Tenant access to each virtual pool is controlled by an Access Control List (ACL). Only tenants that are included in the virtual pool's ACL are permitted access to that virtual pool. If no ACL is set, the virtual pool can be accessed by all tenants. Apply an ACL to the virtual pool using `PUT /block/vpools/{vpool_urn}/acl`. The updated ACL is returned.

Request

```
PUT https://<ViPR_Controller_VIP>:4443/object/vpools/
urn:storageos:VirtualPool:dc96b4d8-d60e-4520-8029-
c903aac06974:vdcl/acl
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<acl_assignment_changes>
  <add>
    <privilege>USE</privilege>
    <tenant>urn:storageos:TenantOrg:db3e2c77-cacc-4d60-ae49-
a2ebfd3379a2:global</tenant>
  </add>
</acl_assignment_changes>
```

Response

```
HTTP 200 OK
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<acl_assignments>
  <acl_assignment>
    <privilege>USE</privilege>
    <tenant>urn:storageos:TenantOrg:db3e2c77-cacc-4d60-ae49-
a2ebfd3379a2:global</tenant>
  </acl_assignment>
</acl_assignments>
```

3. ViPR Controller automatically matches existing physical pools on the ViPR Controller-managed storage systems to the virtual pool characteristics. You have the option of allowing ViPR Controller to automatically associate the matching physical pools to the virtual pool you created, or you can manually select a subset of the matching physical pools to associate to the virtual pool. To manually select a subset of the matching physical pools:

- a. `POST /object/vpools/matching-pools` is used to get a list of storage pools that match a given set of criteria. In this example, the criteria that are sent in the request payload match the properties of the virtual pool that was just created.

Request

```
POST https://<ViPR_Controller_VIP>:4443/object/vpools/matching-
pools
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<block_vpool_create>
  <description>Basic Virtual Pool for Block Provisioning</
description>
  <name>basic_vpool</name>
```



```

<num_paths>1</num_paths>
<protection>
  <snapshots>
    <max_native_snapshots>10</max_native_snapshots>
  </snapshots>
</protection>
<protocols>
  <element>FC</element>
</protocols>
<provisioning_type>Thin</provisioning_type>
<use_matched_pools>>true</use_matched_pools>
<varrays>

<varray>urn:storageos:VirtualArray:ac294bc7-0518-48dd-9f54-4980f
0012f86:vdcl</varray>
  </varrays>
</block_vpool_create>

```

Response

```

HTTP 200 OK
Content-Type: application/xml
<?xml version="1.0" encoding="UTF-8"?>
<storage_pools>
  <storage_pool>>
    <id>urn:storageos:StoragePool:62d2e40d-
ced0-4114-8bb7-8d333e7ef878:vdcl</id>
    <link rel="self" href="/vdc/storage-systems/
urn:storageos:StorageSystem:56d8aa7e-45fe-4383-b49a-
fec72e9927d3:vdcl/storage-pools/urn:storageos:StoragePool:
62d2e40d-ced0-4114-8bb7-8d333e7ef878:vdcl"/>
    <name>SYMMETRIX+000198700420+POOL+TP+3R5-A</name>
  </storage_pool>
  <storage_pool>
    <id>urn:storageos:StoragePool:9e551963-d596-48ae-
b208-1b870126d195:vdcl</id>
    <link rel="self" href="/vdc/storage-systems/
urn:storageos:StorageSystem:56d8aa7e-45fe-4383-b49a-
fec72e9927d3:vdcl/storage-pools/urn:storageos:StoragePool:
9e551963-d596-48ae-b208-1b870126d195:vdcl"/>
    <name>SYMMETRIX+000198700420+POOL+TP+M3</name>
  </storage_pool>
  <storage_pool>
    <id>urn:storageos:StoragePool:ad0dae57-d318-409c-
a859-6c9e59a28251:vdcl</id>
    <link rel="self" href="/vdc/storage-systems/
urn:storageos:StorageSystem:56d8aa7e-45fe-4383-b49a-
fec72e9927d3:vdcl/storage-pools/
urn:storageos:StoragePool:ad0dae57-d318-409c-
a859-6c9e59a28251:vdcl"/>
    <name>SYMMETRIX+000198700420+POOL+TP+M2</name>
  </storage_pool>
  <storage_pool>
    <id>urn:storageos:StoragePool:78caaf4a-673e-4580-ae41-
a545a45e5e28:vdcl</id>
    <link rel="self" href="/vdc/storage-systems/
urn:storageos:StorageSystem:56d8aa7e-45fe-4383-b49a-
fec72e9927d3:vdcl/storage-pools/urn:storageos:StoragePool:
78caaf4a-673e-4580-ae41-a545a45e5e28:vdcl"/>
    <name>SYMMETRIX+000198700420+POOL+TP+data</name>
  </storage_pool>
  <storage_pool>
    <id>urn:storageos:StoragePool:38e2a0c2-89c9-4ec7-
ad9b-29420e464012:vdcl</id>
    <link rel="self" href="/vdc/storage-systems/
urn:storageos:StorageSystem:56d8aa7e-45fe-4383-b49a-
fec72e9927d3:vdcl/storage-pools/urn:storageos:StoragePool:
38e2a0c2-89c9-4ec7-ad9b-29420e464012:vdcl"/>

```

```

        <name>SYMMETRIX+000198700420+POOL+TP+t3</name>
    </storage_pool>
    <storage_pool>
        <id>urn:storageos:StoragePool:dbdcd42a-
b0a4-49b4-8a6e-204a09b8b1ef:vdcl</id>
        <link rel="self" href="/vdc/storage-systems/
urn:storageos:StorageSystem:56d8aa7e-45fe-4383-b49a-
fec72e9927d3:vdcl/storage-pools/
urn:storageos:StoragePool:dbdcd42a-
b0a4-49b4-8a6e-204a09b8b1ef:vdcl"/>
        <name>SYMMETRIX+000198700420+POOL+TP+3R5-B</name>
    </storage_pool>
</storage_pools>

```

- b. Using the URN returned for the new virtual pool returned from the POST / object/vpools request, assign one or more of the storage pools that match the properties of the virtual pool by sending a PUT/object/vpools/{identifier}/assign-matched-pools request. The representation of the updated virtual pool is returned.

Request

```

PUT https://<ViPR_Controller_VIP>:4443/block/vpools/
urn:storageos:VirtualPool:ba745027-2385-437c-bf44-
a49fb32bb45c:vdcl/assign-matched-pools
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<vpool_pool_update>
  <assigned_pool_changes>
    <add>
      <storage_pool>
        <storage_pool>urn:storageos:StoragePool:
62d2e40d-ced0-4114-8bb7-8d333e7ef878:vdcl</storage_pool>
        <storage_pool>urn:storageos:StoragePool:
78caaf4a-673e-4580-ae41-a545a45e5e28:vdcl</storage_pool>

      <storage_pool>urn:storageos:StoragePool:ad0dae57-d318-409c-
a859-6c9e59a28251:vdcl</storage_pool>
    </storage_pool>
    </add>
  </assigned_pool_changes>
</vpool_pool_update>

```

Response

```

HTTP 200 OK
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8"?>
<block_vpool>
  <creation_time>1379202604198</creation_time>
  <global>false</global>
  <id>urn:storageos:VirtualPool:6cd9f843-1b41-4b9f-8eb1-
a26797d0a268:vdcl</id>
  <inactive>false</inactive>
  <internal>false</internal>
  <link rel="self" href="/block/vpools/
urn:storageos:VirtualPool:6cd9f843-1b41-4b9f-8eb1-
a26797d0a268:vdcl"/>
  <name>basic_vpool</name>
  <remote>false</remote>
  <tags />
</vdc>

```

```

        <id>urn:storageos:VirtualDataCenter:030618c2-c6b2-40b0-
a105-6b669983f58f:vdc1</id>
        <link rel="self" href="/vdc/
urn:storageos:VirtualDataCenter:030618c2-c6b2-40b0-
a105-6b669983f58f:vdc1"/>
    </vdc>
    <assigned_storage_pools>
        <storage_pool>
            <id>urn:storageos:StoragePool:78caaf4a-673e-4580-
ae41-a545a45e5e28:vdc1</id>
            <link rel="self" href="/vdc/storage-systems/
urn:storageos:StorageSystem:56d8aa7e-45fe-4383-b49a-
fec72e9927d3:vdc1/storage-pools/urn:storageos:StoragePool:
78caaf4a-673e-4580-ae41-a545a45e5e28:vdc1"/>
        </storage_pool>
        <storage_pool>
            <id>urn:storageos:StoragePool:ad0dae57-d318-409c-
a859-6c9e59a28251:vdc1</id>
            <link rel="self" href="/vdc/storage-systems/
urn:storageos:StorageSystem:56d8aa7e-45fe-4383-b49a-
fec72e9927d3:vdc1/storage-pools/
urn:storageos:StoragePool:ad0dae57-d318-409c-
a859-6c9e59a28251:vdc1"/>
        </storage_pool>
        <storage_pool>
            <id>urn:storageos:StoragePool:62d2e40d-
ced0-4114-8bb7-8d333e7ef878:vdc1</id>
            <link rel="self" href="/vdc/storage-systems/
urn:storageos:StorageSystem:56d8aa7e-45fe-4383-b49a-
fec72e9927d3:vdc1/storage-pools/urn:storageos:StoragePool:
62d2e40d-ced0-4114-8bb7-8d333e7ef878:vdc1"/>
        </storage_pool>
    </assigned_storage_pools>
    <description>Basic Virtual Pool for Block Provisioning</
description>
    <invalid_matched_pools />
    <matched_storage_pools>
        <storage_pool>
            <id>urn:storageos:StoragePool:78caaf4a-673e-4580-
ae41-a545a45e5e28:vdc1</id>
            <link rel="self" href="/vdc/storage-systems/
urn:storageos:StorageSystem:56d8aa7e-45fe-4383-b49a-
fec72e9927d3:vdc1/storage-pools/urn:storageos:StoragePool:
78caaf4a-673e-4580-ae41-a545a45e5e28:vdc1"/>
        </storage_pool>
        <storage_pool>
            <id>urn:storageos:StoragePool:ad0dae57-d318-409c-
a859-6c9e59a28251:vdc1</id>
            <link rel="self" href="/vdc/storage-systems/
urn:storageos:StorageSystem:56d8aa7e-45fe-4383-b49a-
fec72e9927d3:vdc1/storage-pools/
urn:storageos:StoragePool:ad0dae57-d318-409c-
a859-6c9e59a28251:vdc1"/>
        </storage_pool>
        <storage_pool>
            <id>urn:storageos:StoragePool:62d2e40d-
ced0-4114-8bb7-8d333e7ef878:vdc1</id>
            <link rel="self" href="/vdc/storage-systems/
urn:storageos:StorageSystem:56d8aa7e-45fe-4383-b49a-
fec72e9927d3:vdc1/storage-pools/urn:storageos:StoragePool:
62d2e40d-ced0-4114-8bb7-8d333e7ef878:vdc1"/>
        </storage_pool>
    </matched_storage_pools>
    <num_paths>1</num_paths>
    <num_resources>4</num_resources>
    <protocols>
        <protocol>FC</protocol>
    </protocols>
    <provisioning_type>Thin</provisioning_type>

```

```

<system_type>NONE</system_type>
<type>block</type>
<use_matched_pools>true</use_matched_pools>
  <varrays>
    <varray>
<id>urn:storageos:VirtualArray:dc09417d-9028-40de-8bff-74574996f
2d5:vdc1</id>
      <link rel="self" href="vdc/varrays/
urn:storageos:VirtualArray:dc09417d-9028-40de-8bff-74574996f2d5:
vdc1/>
    </varray>
  </varrays>
  <drive_type>NONE</drive_type>
  <expandable>true</expandable>
  <fast_expansion>>false</fast_expansion>
  <high_availability/>
  <max_paths>2</max_paths>
  <min_paths>1</min_paths>
  <multi_volume_consistency>true</multi_volume_consistency>
  <paths_per_initiator>1</paths_per_initiator>
  <protection>
    <continuous_copies>
      <max_native_continuous_copies>0</
max_native_continuous_copies>
    </continuous_copies>
    <snapshots>
      <max_native_snapshots>10</max_native_snapshots>
    </snapshots>
  </protection>
  <protocols>
    <element>FC</element>
  </protocols>
  <provisioning_type>Thin</provisioning_type>
  <raid_levels />
  <thin_volume_preallocation_percentage>0</
thin_volume_preallocation_percentage>
  <unique_auto_tier_policy_names>>false</
unique_auto_tier_policy_names>
</block_vpool>

```

CHAPTER 15

Assigning an Access Control List (ACL) to a resource

This chapter contains the following topics:

- [ACLs](#).....158
- [Virtual array and virtual pool ACLs](#)..... 158
- [Project ACLs](#)..... 160

ACLs

An Access Control List (ACL) is a list of permissions attached to a ViPR Controller resource that specifies which tenants are authorized to access VDC-level resources and which users or groups are authorized to access tenant-level resources. ACLs also indicate which operations are allowed on the resource.

Assigning an ACL to a resource is one means of setting up which users and groups are authorized to access ViPR Controller functionality . Assigning roles to users and groups is another method. See [Assigning roles to users and groups on page 165](#).

The *EMC ViPR Controller REST API Reference* on the [ViPR Controller Product Documentation Index](#) provides a description and complete list of parameters for the REST API methods used in this chapter.

Virtual array and virtual pool ACLs

At creation time, virtual arrays and virtual pools are public. They are accessible to all tenants.

A System or Security Administrator can assign an ACL to a virtual pool or virtual array to restrict them to be used by specified tenants.

The ACL permission associated with virtual arrays and pools is of the type USE. If a specific tenant has a USE ACL on a virtual pool, this means that all the users who are mapped to that tenant will be allowed to use that virtual pool in their provisioning operations.

All newly created virtual arrays and pools will have an empty ACL. The System or Security Administrator is responsible for managing the ACL. If no ACLs are set, the virtual arrays and pools remain accessible to the provider tenant and all other tenants in the ViPR system.

For virtual pools and virtual arrays, you cannot assign an ACL to a user (subject ID) or group. Only tenants can be assigned ACLs for these resources.

The following table shows the APIs that allow a system or security administrator to modify ACLs for virtual arrays and virtual pools. Note that there are separate API calls for file and block virtual pools.

Table 18 REST APIs for Modifying ACLs for Virtual Arrays and Virtual Pools

API	Description
GET /block/vpools/{id}/acl	Show ACL assignment for block store virtual pool .
PUT /block/vpools/{id}/acl	Add or remove block store virtual pool ACL entries.
GET /file/vpools/{id}/acl	Show ACL entries for file store VirtualPool.
PUT /file/vpools/{id}/acl	Add or remove ACL entries from file store VirtualPool.
GET /compute/vpools/{id}acl	Show the ACL assignment for a compute virtual pool.
PUT /compute/vpools/{id}/acl	Add or remove compute virtual pool ACL entries.
GET /vdc/varrays/{id}/acl	Show VirtualArray ACL Virtual Array.
PUT /vdc/varrays/{id}/acl	Add or remove ACL for VirtualArray Virtual Array.

Table 18 REST APIs for Modifying ACLs for Virtual Arrays and Virtual Pools (continued)

Examples: Virtual array and virtual pool ACL APIs

The examples in this section show some commonly-used APIs for managing virtual array and virtual pool ACLs.

Virtual array: Assigning the USE ACL to a tenant

The following example shows how to give a tenant privileges to use a virtual array. If no ACL exists on the virtual array, all tenants have access to it.

Request

```
PUT https://<ViPR_Controller_VIP>:4443/vdc/varrays/
urn:storageos:VirtualArray:f49f6e36-0fe5-4181-9622-49d116204d86:vdc1/
acl
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<acl_assignment_changes>
  <add>
    <privilege>USE</privilege>
    <tenant>urn:storageos:TenantOrg:7985d438-9980-41df-
bba1-29d6a873f811:global</tenant>
  </add>
</acl_assignment_changes>
```

Response:

```
HTTP 200 OK
Content-Type: application/xml
<acl_assignments>
  <acl_assignment>
    <privilege>USE</privilege>
    <tenant>urn:storageos:TenantOrg:7985d438-9980-41df-
bba1-29d6a873f811:global</tenant>
  </acl_assignment>
</acl_assignments>
```

Virtual array: Removing the USE ACL from a tenant

Request

```
PUT https://<ViPR_Controller_VIP>:4443/vdc/varrays/
urn:storageos:VirtualArray:f49f6e36-0fe5-4181-9622-49d116204d86:vdc1/
acl
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<acl_assignment_changes>
  <remove>
    <privilege>USE</privilege>
    <tenant>urn:storageos:TenantOrg:7985d438-9980-41df-
bba1-29d6a873f811:global</tenant>
  </remove>
</acl_assignment_changes>
```

Response:

```
HTTP 200 OK
Content-Type: application/xml
<acl_assignments/>
```

Virtual pool: Assigning the USE ACL to a tenant

The following example shows how to give a tenant privileges to use a virtual pool. If no ACL exists on the virtual pool, all tenants have access to it.

Request

```
PUT https://<ViPR_Controller_VIP>:4443/file/vpools/
urn:storageos:VirtualPool:4394653f-cf2e-4301-8f11-9e8d3e7e7fa9:vdcl/
acl
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<acl_assignment_changes>
  <add>
    <privilege>USE</privilege>
    <tenant>urn:storageos:TenantOrg:d61d9fa1-9886-40ef-85d3-
c40b6de2c72f:global</tenant>
  </add>
</acl_assignment_changes>
```

Response:

```
HTTP 200 OK
Content-Type: application/xml
<acl_assignments>
  <acl_assignment>
    <privilege>USE</privilege>
    <tenant>urn:storageos:TenantOrg:d61d9fa1-9886-40ef-85d3-
c40b6de2c72f:global</tenant>
  </acl_assignment>
</acl_assignments>
```

Project ACLs

A Tenant Administrator can access all projects for their tenant. Project Administrators can only access projects that they own.

Newly created projects will have an empty ACL.

The project ACLs can be created or modified by a Tenant Administrator, a Security Administrator, or a project owner. Project owners are assigned the OWN ACL. The user that creates the project is the owner of that project unless they, or a tenant administrator, transfers ownership of that project to another user.

The default ACL behavior of a project is different from the default ACL behavior of a Virtual Array or Virtual Pool. Whereas, the default ACL for a Virtual Array or Virtual Pool enables anyone to use them, the default ACL for the Project prevents all but the Tenant Admin or Project owner from using it (for example, to create a volume in the project). For other users or groups to use a project, that user or group must be explicitly added to the ACL for that project.

The ACL permissions associated with projects are listed in the following table.

Table 19 Project ACLs

Project ACL	Description
ALL	The user can manage the resources in the project (that is, perform create, read, update, and delete (CRUD) operations on file systems, volumes, snapshots, exports, and buckets).
BACKUP	The user has read-only access to the first-level resources under the project (that is, volumes, file systems, and buckets) and full access to snapshot operations (can create/delete/export snapshots).
OWN	OWN is an internal ACL permission that you use to identify the user as a project owner. The internal OWN ACL on a project is modified by editing a project's properties, not the project's ACL. (The API uses the <code>project update</code> API call is used, not the <code>update project ACL</code> call.) A user with an OWN ACL can: <ul style="list-style-type: none"> • Perform create, read, update, and delete operations on project resources. • Set ACLs on the project. This includes the use of the <code>project update</code> API to set another user as the new owner.) • Delete the project. • Set project properties such as the project name and owner.
ANY	ANY is an internal ACL that identifies users with any of the above ACLs on a project. You cannot use the project ACL assignment API to modify the internal ANY ACL.

Examples: Project ACL APIs

The examples in this section show some commonly-used APIs for managing project ACLs.

Get the ACLs for a project

Request

```
GET https://<ViPR_Controller_VIP>:4443/projects/{Project_URN}/acl
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>
```

Response

```
HTTP 200 OK
Content-Type: application/xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<acl_assignments>
  <acl_assignment>
    <privilege>ALL</privilege>
    <subject_id>jordab@sanity.local</subject_id>
  </acl_assignment>
  <acl_assignment>
    <privilege>BACKUP</privilege>
    <subject_id>jordab2@sanity.local</subject_id>
  </acl_assignment>
</acl_assignments>
```

Assigning the USE ACL to a user

The following example sets the project ACL using a user's `subject_id`. A `subject_id` or group can be assigned the ALL or BACKUP permission.

Request

```
PUT https://<ViPR_Controller_VIP>:4443/projects/<project_urn>/acl
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<acl_assignment_changes>
  <add>
    <privilege>ALL</privilege>
    <subject_id>jordab2@sanity.local</subject_id>
  </add>
</acl_assignment_changes>
```

Response

```
HTTP 200 OK
Content-Type: application/xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<acl_assignments>
  <acl_assignment>
    <privilege>ALL</privilege>
    <subject_id>jordab@sanity.local</subject_id>
  </acl_assignment>
  <acl_assignment>
    <privilege>ALL</privilege>
    <privilege>BACKUP</privilege>
    <subject_id>jordab2@sanity.local</subject_id>
  </acl_assignment>
</acl_assignments>
```

Examples: Changing a project's owner

The example in this section shows how to change the owner of a project.

The OWN ACL is assigned to a project's creator, giving that user ownership rights to that project. A tenant admin, a project admin or the project's owner can reassign the ownership of the project to another user.

Checking the owner of a project

The user that owns the project is displayed in the `<owner>` field of the project resource. Here, the user `jordab@sanity.local` is displayed as the project owner.

```
GET https://<ViPR_Controller_VIP>:4443/projects/{Project_urn}
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>
```

```
HTTP 200 OK
Content-Type: application/xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<project>
  <creation_time>1400267698359</creation_time>
  <global>true</global>
  <id>urn:storageos:Project:7581d618-
e124-4c7f-9a04-624cad271ff2:global</id>
  <inactive>>false</inactive>
  <internal>>false</internal>
  <link rel="self" href="/projects/urn:storageos:Project:7581d618-
e124-4c7f-9a04-624cad271ff2:global"/>
  <name>Snapshot Project</name>
```

```

    <tags/>
    <vdc>
      <id>urn:storageos:VirtualDataCenter:030618c2-c6b2-40b0-
a105-6b669983f58f:vdcl</id>
      <link rel="self" href="/vdc/urn:storageos:VirtualDataCenter:
030618c2-c6b2-40b0-a105-6b669983f58f:vdcl"/>
    </vdc>
    <owner>jordab@sanity.local</owner>
    <tenant>
      <id>urn:storageos:TenantOrg:2b5f6d7c-e670-4aee-9fc1-
ddb0fc8de22:global</id>
      <link rel="self" href="/tenants/urn:storageos:TenantOrg:
2b5f6d7c-e670-4aee-9fc1-ddb0fc8de22:global"/>
    </tenant>
  </project>

```

Changing the owner of a project

This example changes the ownership of the project shown in the previous example to jordab2@sanity.local. Note that this is done by changing the owner attribute of the project, rather than through an ACL call.

Request

```

PUT https://<ViPR_Controller_VIP>:4443/projects/{Project_URN}
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<project_update>
  <owner>jordab2@sanity.local</owner>
</project_update>

```

Response

```

HTTP 200 OK
Content-Type: application/xml

```

Assigning an Access Control List (ACL) to a resource

CHAPTER 16

Assigning roles to users and groups

This chapter contains the following topics:

- [Overview of assigning roles using the REST API](#) 166
- [Assign a Virtual Data Center Role to a User or Group](#) 166
- [Assign a Tenant Role to a User or Group](#) 167

Overview of assigning roles using the REST API

Learn how to assign roles to users and groups in ViPR Controller using the ViPR Controller REST API . ViPR Controller has two types of roles: virtual data center roles and tenant-level roles. When you assign a role to a user or group, you are using one of the methods to control authorization in ViPR Controller.

See *ViPR Controller Concepts* on the [ViPR Controller Product Documentation Index](#) .

The other method to control authorization is to assign an Access Control List (ACL) to a resource. See [Assigning an Access Control List \(ACL\) to a resource on page 157](#).

Assign a Virtual Data Center Role to a User or Group

A Security Administrator can assign roles to users or groups that belong to the top-level provider tenant. These roles identify what functions the user or group can perform, such as adding physical assets or creating users, at the level of the entire virtual data center.

Before you begin

- Authenticate with the ViPR REST API as a Security Administrator. See [Authenticating with the REST API on page 9](#).
- If you plan to assign a role to a user or group that is in LDAP or Active Directory, ensure that you meet the following conditions:
 - You have already added an authentication provider to ViPR Controller.
 - You have already mapped those users, for whom you are assigning a virtual data center role, to the provider tenant.
- The virtual data center level roles include:
 - Security Administrator (SECURITY_ADMIN)
 - System Administrator (SYSTEM_ADMIN)
 - System Monitor (SYSTEM_MONITOR)
 - System Auditor (SYSTEM_AUDITOR)

For information on ViPR Controller users, roles and ACLs, see *ViPR Controller Concepts* on the [ViPR Controller Product Documentation Index](#) .

The *EMC ViPR Controller REST API Reference* on the [ViPR Controller Product Documentation Index](#) provides a description and complete list of parameters for the REST API methods used in this section.

In the following example, a Security Administrator role is being assigned to the user, `username@mycompany.com`, using `PUT https://<ViPR_Controller_VIP>:4443/vdc/role-assignments` . The response is an updated list of role assignments for the virtual data center, including the SECURITY_ADMIN role assigned to the user, `username@mycompany.com`.

Procedure

1. To apply the SECURITY_ADMIN role assignment to `username@mycompany.com`, send the following request.

Request

```
PUT https://<ViPR_Controller_VIP>:4443/vdc/role-assignments HTTP/1.1
```

```

Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<role_assignment_change>
  <add>
    <role>SECURITY_ADMIN</role>
    <subject_id>username@mycompany.com</subject_id>
  </add>
</role_assignment_change>

```

Response

```

HTTP/1.1 200 OK
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<role_assignments_create>
  <role_assignment>
    <role>SYSTEM_ADMIN</role>
    <subject_id>username@mycompany.com</subject_id>
  </role_assignment>
  <role_assignment>
    <role>SYSTEM_AUDITOR</role>
    <role>SECURITY_ADMIN</role>
    <role>SYSTEM_MONITOR</role>
    <role>SYSTEM_ADMIN</role>
    <subject_id>username@mycompany.com</subject_id>
  </role_assignment>
  <link href="/vdc/role-assignments" rel="self"/>
</role_assignments_create>

```

Assign a Tenant Role to a User or Group

A Security Administrator or Tenant Administrator with access to a tenant can assign roles to users or groups in that tenant by sending a PUT `https://<ViPR_Controller_VIP> : 4443/tenants/<tenant_URN>/role-assignments`. These roles identify what functions the user can do at the tenant level.

Before you begin

- Authenticate with the ViPR Controller REST API with the Security Administrator or Tenant Administrator role with access to the tenant.
- You need the URN of the tenant .
- If you plan to assign a role to a user or group that is in LDAP or Active Directory, ensure that you meet the following conditions:
 - You have already added an authentication provider to ViPR Controller.
 - You have already configured users and groups for the provider tenant.
- The tenant level roles include:
 - Tenant Administrator (TENANT_ADMIN)
 - Tenant Approver (TENANT_APPROVER)
 - Project Administrator (PROJECT_ADMIN)

See *Understanding ViPR Users, Roles, and ACLs* on the [ViPR Controller Product Documentation Index](#) .

The *EMC ViPR Controller REST API Reference* on the [ViPR Controller Product Documentation Index](#) provides a description and complete list of parameters for the REST API methods used in this section.

In the following example, a Tenant Administrator role is being assigned to the user, `user@mycompany.com`, using PUT `https://<ViPR_Controller_VIP>:4443/tenants/<tenant_URN>/role-assignments`. The response is an updated list of role assignments for the tenant.

Procedure

1. To apply the TENANT_ADMIN role assignment to `user@mycompany.com`, send the following request:

Request

```
PUT https://<ViPR_Controller_VIP>:4443/tenants/
urn:storageos:TenantOrg:dbeb4135-e297-40d9-
a5d4-9b40c73bdb4b:global/role-assignments HTTP/1.1

Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<role_assignment_change>
  <add>
    <role>TENANT_ADMIN</role>
    <subject_id>user@mycompany.com</subject_id>
  </add>
</role_assignment_change>
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <role_assignments_create>
    <link href="/tenants/urn:storageos:TenantOrg:dbeb4135-
e297-40d9-a5d4-9b40c73bdb4b:global/role-assignments" rel="self"/>
    <role_assignment>
      <role>TENANT_ADMIN</role>
      <subject_id>user@mycompany.com</subject_id>
    </role_assignment>
    <role_assignment>
      <role>TENANT_ADMIN</role>
      <subject_id>root</subject_id>
    </role_assignment>
  </role_assignments_create>
```


CHAPTER 17

Using Tasks to track the status of asynchronous operations

This chapter contains the following topics:

- [Overview](#)..... 170
- [Track status of asynchronous operations using task details](#)..... 170
- [Search for tasks by resource ID or state](#)..... 172
- [Get statistics for tenant and system tasks](#)..... 174
- [Delete a task that is stuck in the pending state](#)..... 175
- [Change task-related configuration settings](#)..... 175
- [Important REST API calls to manage tasks](#)..... 176

Overview

A number of ViPR Controller operations and services are processed asynchronously. Asynchronous operations return a task or list of tasks.

Each task represents a block of work performed by the controller engine. These tasks can be followed to determine if the operation has succeeded, failed or is in progress, and can be followed using both the [UI](#) and the REST API.

There are two types of tasks:

- Tenant tasks, such as adding a host.
 - Any user that is a member of the tenant can view the tasks that are related to that tenant.
 - Any user that is a member of the tenant can view the details of the tasks related to that tenant.
- System tasks that are not associated with any tenant, such as adding a storage array.
 - Only a system administrator can view system tasks.
 - Only system administrators and security administrators can view the details of a system task.

By default, tasks last for seven days from the date of completion. In addition, the time interval between task cleaning operations is 60 minutes. But these values can be changed, as described in [Change task-related configuration settings on page 175](#).

In addition, when you delete a resource the tasks that are associated with the resource are still available for viewing.

The *EMC ViPR Controller REST API Reference* on the [ViPR Controller Product Documentation Index](#) provides additional information concerning the methods in the Task API.

Track status of asynchronous operations using task details

Asynchronous operations return a task. You can periodically check the details of the task, using `GET /vdc/tasks/{id}`, to determine if the operation has succeeded, failed, or is in progress.

When a task is first created, it is in the pending state which indicates that the operation is in progress. In this example, a EMC ScaleIO storage provider is added to ViPR Controller and a task in the pending state is returned.

Request

```
POST https://<ViPR_Controller_VIP>:4443/vdc/storage-providers
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<storage_provider_create>
  <name>scaleio_block_1</name>
  <interface_type>scaleioapi</interface_type>
  <ip_address>192.168.0.0</ip_address>
  <port_number>22</port_number>
  <user_name>admin1</user_name>
  <password>Password1</password>
  <user_name>admin2</secondary_username>
  <password>Password2</secondary_password>
</storage_provider_create>
```

Response

```

HTTP 202 Accepted
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<task>
  <creation_time>1415211256218</creation_time>
  <global>false</global>
  <id>urn:storageos:Task:9d520090-6518-49a2-8b7b-af2c7066416d:vdc1</
id>
  <inactive>false</inactive>
  <internal>false</internal>
  <link rel="self" href="/vdc/tasks/urn:storageos:Task:
9d520090-6518-49a2-8b7b-af2c7066416d:vdc1"/>
  <name>SCAN STORAGE PROVIDER</name>
  <remote>false</remote>
  <tags/>
  <vdc>
    <id>urn:storageos:VirtualDataCenter:d649923c-3d28-4195-bd6e-
d7139c0df033:vdc1</id>
    <link rel="self" href="/vdc/
urn:storageos:VirtualDataCenter:d649923c-3d28-4195-bd6e-
d7139c0df033:vdc1"/>
    </vdc>
    <associated_resources/>
    <op_id>18315650-b792-4ec9-86e9-5198dd28588b</op_id>
    <progress>0</progress>
    <resource>
      <id>urn:storageos:StorageProvider:7c347281-416f-453d-
ade1-31ab03ddf1b9:vdc1</id>
      <link rel="self" href="/vdc/storage-providers/
urn:storageos:StorageProvider:7c347281-416f-453d-
ade1-31ab03ddf1b9:vdc1"/>
      <name>scaleio_block_1</name>
    </resource>
    <start_time>1415211256216</start_time>
    <state>pending</state>
  </task>

```

The task that is returned from an asynchronous operation includes:

- A description of the operation, which in this example is scanning a storage provider.
- The ID of the task.
- The ID, name, and a link to the resource associated with the operation. In this example, it is the storage provider that has been created in ViPR and is being discovered.
- The state of the task, which is pending.

You can use the ID of the task returned from the operation to check the details of the task to determine when it has completed. A task can complete successfully or with an error. To check the details of a task, you use GET /vdc/tasks/{id}.

Request

```

GET https://{Conref}<ViPR_VIP>:4443/vdc/tasks/urn:storageos:Task:
9d520090-6518-49a2-8b7b-af2c7066416d:vdc1
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

```

Response

```

HTTP 200 OK
Content-Type: application/xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

```

```

<task>
  <creation_time>1415211256218</creation_time>
  <global>false</global>
  <id>urn:storageos:Task:9d520090-6518-49a2-8b7b-af2c7066416d:vdc1</id>
  <inactive>false</inactive>
  <internal>false</internal>
  <link rel="self" href="/vdc/tasks/urn:storageos:Task:9d520090-6518-49a2-8b7b-af2c7066416d:vdc1"/>
  <name>SCAN STORAGE PROVIDER</name>
  <remote>false</remote>
  <tags/>
  <vdc>
    <id>urn:storageos:VirtualDataCenter:d649923c-3d28-4195-bd6e-d7139c0df033:vdc1</id>
    <link rel="self" href="/vdc/urn:storageos:VirtualDataCenter:d649923c-3d28-4195-bd6e-d7139c0df033:vdc1"/>
  </vdc>
  <associated_resources/>
  <end_time>1415211260337</end_time>
  <message>Operation completed successfully</message>
  <op_id>18315650-b792-4ec9-86e9-5198dd28588b</op_id>
  <progress>100</progress>
  <resource>
    <id>urn:storageos:StorageProvider:7c347281-416f-453d-ade1-31ab03ddf1b9:vdc1</id>
    <link rel="self" href="/vdc/storage-providers/urn:storageos:StorageProvider:7c347281-416f-453d-ade1-31ab03ddf1b9:vdc1"/>
    <name>scaleio_block_1</name>
  </resource>
  <start_time>1415211256216</start_time>
  <state>ready</state>
</task>

```

The status of the task is determined as follows:

- **In progress:** state is `pending` and `progress` indicates the percentage complete
- **Completed successfully:** message is `Operation completed successfully` and state is `ready`.
- **Completed with an error:** state is `error` and there is an error message and error code included in the task details.

Search for tasks by resource ID or state

GET `/vdc/tasks/search` is used to search for tasks.

The search returns a list of task IDs that match the specified resource ID or state. You can refine the search by specifying a tenant ID. You can also set the tenant ID to `system` to only retrieve system tasks.

The tasks that are returned are not in the order in which they were created.

Search by resource ID

Searching by resource ID returns a list of the tasks that were created for operations involving the specified resource.

```

GET https://<ViPR_Controller_VIP>:4443/vdc/tasks/search?
resource=urn:storageos:StorageProvider:870df5c8-5cae-4d87-9d53-
e080e477edf4:vdc1
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

```

```

HTTP 200 OK
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<results>
  <resource>
    <id>urn:storageos:Task:001b3608-5e50-4dd4-
a62a-3e66b5e76656:vdc1</id>
    <link rel="self" href="/vdc/tasks/urn:storageos:Task:
001b3608-5e50-4dd4-a62a-3e66b5e76656:vdc1"/>
  </resource>
  <resource>
    <id>urn:storageos:Task:
001daa35-567d-4db2-8fd1-91baf87c36b4:vdc1</id>
    <link rel="self" href="/vdc/tasks/urn:storageos:Task:
001daa35-567d-4db2-8fd1-91baf87c36b4:vdc1"/>
  </resource>
  <resource>
    <id>urn:storageos:Task:00213867-7c6f-40e3-
ac20-096d2f26a36b:vdc1</id>
    <link rel="self" href="/vdc/tasks/urn:storageos:Task:
00213867-7c6f-40e3-ac20-096d2f26a36b:vdc1"/>
  </resource>
  <resource>
    <id>urn:storageos:Task:00244adb-4da1-4e8c-8dd0-
dbaff16248fb:vdc1</id>
    <link rel="self" href="/vdc/tasks/urn:storageos:Task:
00244adb-4da1-4e8c-8dd0-dbaff16248fb:vdc1"/>
  </resource>
</results>

```

Search by state

Searching by state returns a list of IDs for all tasks that are in the state that you specify. The state can be:

- pending
- ready
- error

You can also include a tenant ID to specify from which tenant the tasks should be returned. Setting the tenant ID equal to system returns the system tasks that match the requested state. If you do not include a tenant ID, then tasks will be returned from all tenants to which you have access.

Once you have the task IDs, you can view the details of a single task or multiple tasks.

In this example, all system tasks that have completed successfully and are in the ready state are returned.

```

GET https://<ViPR_Controller_VIP>:4443/vdc/tasks/search?
state=ready&tenant=system
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

HTTP 200 OK
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<results>
  <resource>
    <id>urn:storageos:Task:001b3608-5e50-4dd4-
a62a-3e66b5e76656:vdc1</id>
    <link rel="self" href="/vdc/tasks/urn:storageos:Task:
001b3608-5e50-4dd4-a62a-3e66b5e76656:vdc1"/>
  </resource>
</results>

```

```

        <id>urn:storageos:Task:
001daa35-567d-4db2-8fd1-91baf87c36b4:vdc1</id>
        <link rel="self" href="/vdc/tasks/urn:storageos:Task:
001daa35-567d-4db2-8fd1-91baf87c36b4:vdc1"/>
    </resource>
</resource>
        <id>urn:storageos:Task:00213867-7c6f-40e3-
ac20-096d2f26a36b:vdc1</id>
        <link rel="self" href="/vdc/tasks/urn:storageos:Task:
00213867-7c6f-40e3-ac20-096d2f26a36b:vdc1"/>
    </resource>
</resource>
        <id>urn:storageos:Task:00244adb-4da1-4e8c-8dd0-
dbaff16248fb:vdc1</id>
        <link rel="self" href="/vdc/tasks/urn:storageos:Task:
00244adb-4da1-4e8c-8dd0-dbaff16248fb:vdc1"/>
    </resource>
</resource>
        <id>urn:storageos:Task:
0031bcaa-9cea-4954-8b57-99062058a700:vdc1</id>
        <link rel="self" href="/vdc/tasks/urn:storageos:Task:
0031bcaa-9cea-4954-8b57-99062058a700:vdc1"/>
    </resource>
</resource>
        <id>urn:storageos:Task:00a3dded-c348-474b-
a806-239519c5c4e1:vdc1</id>
        <link rel="self" href="/vdc/tasks/urn:storageos:Task:00a3dded-
c348-474b-a806-239519c5c4e1:vdc1"/>
    </resource>
</resource>
        <id>urn:storageos:Task:00daeba5-3574-4cfd-bd51-
b8d828b8c6f6:vdc1</id>
        <link rel="self" href="/vdc/tasks/urn:storageos:Task:
00daeba5-3574-4cfd-bd51-b8d828b8c6f6:vdc1"/>
    </resource>
</resource>
</results>

```

Get statistics for tenant and system tasks

GET `vdc/tasks/stats?tenant=<tenant_URI>` is used to get a count of the number of tenant and system tasks in the error, pending, and ready (completed) state.

When you send a GET `/vdc/tasks/stats` request without any tenant specified, then a task status count is returned for your tenant. In this example, there are no tasks for the tenant of the logged in user.

```

GET https://<ViPR_Controller_VIP>:4443/vdc/tasks/stats
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

HTTP 200 OK
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<task_stats>
  <error>0</error>
  <pending>0</pending>
  <ready>0</ready>
</task_stats>

```

You can also get the task status count for system tasks, by specifying `system` as the tenant. In this example, there are no system tasks in the error or pending state, but there are 1686 tasks that have successfully completed.

```
GET https://<ViPR_Controller_VIP>:4443/vdc/tasks/stats?tenant=system
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

HTTP 200 OK
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<task_stats>
  <error>0</error>
  <pending>0</pending>
  <ready>1686</ready>
</task_stats>
```

Delete a task that is stuck in the pending state

Tasks can become permanently pending due to disruption between a ViPR Controller REST API client and the ViPR Controller nodes.

A system administrator can use the ViPR Controller REST API to send a `POST /vdc/tasks/<task_id>/delete` request to remove any pending tasks that have become stuck due to a ViPR Controller node disruption.

```
POST https://<ViPR_Controller_VIP>:4443/vdc/tasks/<task_id>/delete
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

HTTP 200 OK
```

Change task-related configuration settings

You can configure time settings for tasks using `PUT /config/properties`.

Before you begin

- Authenticate with the ViPR REST API as a Security Administrator. For information on authentication with the REST API, see the *ViPR Controller REST API Virtual Data Center Configuration Guide* on the [ViPR Controller Product Documentation Index](#)
- Cluster state should be STABLE.

<code>task_ttl</code>	Number of minutes to keep tasks once they have completed. The default is 10080 (7 days).
<code>task_task_clean_interval</code>	Time interval in minutes between task cleaning operations. The default is 60 minutes.
	<p>Note</p> <p>After you change this property, you must initiate a reboot of the ViPR Controller nodes. Rebooting ViPR Controller nodes may disrupt ViPR Controller processes that are currently running.</p>

In this example, `task_ttl` is changed to 8 days (11520 minutes). Since only the `task_ttl` is being updated, you do not need to reboot the ViPR Controller nodes.

Procedure

1. Change a task configuration property by sending a PUT `/config/properties` request. The request returns

Request

```
PUT https://<ViPR_Controller_VIP>:4443/config/properties
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<property_update>
  <properties>
    <entry>
      <key>task_ttl</key>
      <value>11520</value>
    </entry>
  </properties>
</property_update>
```

Response

```
HTTP 200 OK

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<cluster_info>
  <cluster_state>UPDATING</cluster_state>
  <link rel="self" href="/upgrade/cluster-state"/>
</cluster_info>
```

Important REST API calls to manage tasks

The table shows some important APIs that are used to manage tasks.

Table 20 Important REST API Calls for Tasks

ViPR REST API Call	Description
GET <code>/vdc/tasks/{id}</code>	Shows the details of a task.
GET <code>/vdc/tasks/stats</code>	Returns the number of tasks in a pending state, a ready state, and an error state for a tenant or the system.
GET <code>/vdc/tasks</code>	Return a list of tasks for a tenant. If a tenant URI is not supplied as a query parameter, then the tenant id of the current user is used. If <code>system</code> is supplied as a query parameter, then a list of system tasks is returned. You can also filter the list of tasks returned by including the start time, the end time, or the maximum count as query parameters.
POST <code>/vdc/tasks/{id}/delete</code>	Deletes a task.
GET <code>/vdc/tasks/search</code>	Searches for tasks by resource ID or state. You can refine the search using a tenant ID.
GET <code>/vdc/tasks/bulk</code>	Returns a list containing IDs of all of the tasks.
POST <code>/vdc/tasks/bulk</code>	Returns the details for all of the tasks whose IDs are specified in the request payload.

Table 20 Important REST API Calls for Tasks (continued)

ViPR REST API Call	Description
GET /vdc/tasks/{id}/tags	List the tags assigned to a task.
PUT /vdc/tasks/{id}/tags	Assign tags to a task.

Using Tasks to track the status of asynchronous operations

CHAPTER 18

Understanding and setting up port selection using performance metrics

This chapter contains the following topics:

- [Overview](#) 180
- [How does ViPR Controller select a port when using performance metrics](#) 180
- [Global default port selection](#) 183
- [Set up metering prerequisites in ViPR Controller](#) 184
- [Change the default port allocation parameters using the REST API](#) 185
- [Prerequisites for VNX and HDS metrics-based port selection](#) 188
- [HDS performance metrics](#) 188
- [VMAX performance metrics](#) 189
- [VNX for Block performance metrics](#) 190

Overview

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, VNX for Block, and Hitachi Data Systems (HDS).

Several performance-based metrics are collected from [VMAX on page 189](#), [VNX for Block on page 190](#), and [HDS on page 188](#) and are used to determine:

- Port percent busy.
- CPU percent busy.

Two additional metrics are also computed:

- 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 180](#).

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 184](#).

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 to determine which ports to select.

Metric calculations and averages

Several performance-based metrics are collected from [VMAX on page 189](#), [VNX for Block on page 190](#), and [HDS on page 188](#) and are used to determine:

- Port percent busy
- CPU percent busy

In addition, two additional metrics are computed by ViPR 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.

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 184](#).
- 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 [Change the port allocation parameters using the REST API on page 185](#).
- 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 [Change the port allocation parameters using the REST API on page 185](#).

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 - 0.4).

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).

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. See [Change the port allocation parameters using the REST API on page 185](#).

You can change the following settings:

- Maximum number of initiators that can use the port before new allocations are not allowed.
- Maximum number of volumes that can use the port before new allocations are not allowed.

Note

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.

-
- Maximum average port percent busy value (from 0 - 100%) before new allocations are not allowed.
 - Maximum average CPU percent busy value (from 0 - 100%) before new allocations are not allowed.
 - The sample averaging time in days (1 -30 days)
 - The weight for the EMA (the EMA factor).
 - Metrics enabled
 - true = use collected metrics and calculate the port percent busy and the CPU percent busy.
 - 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.

Note

You should 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).

Allocating a port

The EMA Factor and (1- EMA Factor) values that you configured are used when ViPR Controller allocates a port. ViPR Controller takes the (modified moving average x EMAfactor) and the (EMA x 1 - EMAfactor) and does an instantaneous check of these values against the ceilings that you configured. For example, if you have the EMA factor set at 0.6, then ViPR Controller takes the (modified moving average x 0.6) and the (EMA x 0.4) for the instantaneous check against your configured ceiling values.

The port with the lowest metric, which has not reached or exceeded a ceiling is selected. When you require more than one port allocated, ViPR Controller tries to choose two ports that are on different hardware units. For example, you need two ports on a VMAX and there are 3 ports available:

- 7E0 has a port metric of 10
- 7F0 has a port metric of 20
- 8E0 has a port metric of 30

ViPR Controller chooses 7E0 as the first port since it has the lowest port metric, but 8E0 is chosen as the second port. Port 8E0 has a higher port metric than 7F0, but 8E0 is on a different director and, therefore, on different hardware units. This provides redundancy against hardware failures.

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, VNX for Block, or Hitachi Data Systems (HDS).
- For storage arrays other than VMAX, 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 using the REST API. See [Change the default port allocation parameters using the REST API on page 185](#).

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.

- 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 to enable the collection of metrics from VMAX, VNX for Block, and HDS. These properties are `controller_enable_metering` and `controller_metering_interval`.

Before you begin

- Authenticate with the ViPR REST API as a System Administrator. See the *ViPR Controller REST API Virtual Data Center Configuration Guide* on the [ViPR Controller Product Documentation Index](#).
- Configuration properties to verify:
 - `controller_enable_metering` - should be set to `true` to collect metrics from the arrays.
 - `controller_metering_interval` - sets how often ViPR Controller will collect 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 will be the results. However, the higher the metering interval is, the less overhead there is on ViPR Controller and the array.
- The *EMC ViPR Controller REST API Reference* on the [ViPR Controller Product Documentation Index](#) provides a description and complete list of parameters for the REST API methods used in this article.

Procedure

- Get the list of configuration properties and their values by sending a `GET /config/properties` request.

Request

```
GET https://<ViPR_Controller_VIP>:4443/config/properties
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>
```

Response

Note

For readability, only a small number of the configuration properties are shown in the response.

```
HTTP 200 OK
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<property_info>
  <properties>
```



```

<entry>
  <key>backup_external_location_password</key>
  <value/>
</entry>
<entry>
  <key>backup_external_location_url</key>
  <value/>
</entry>
<entry>
  <key>backup_external_location_username</key>
  <value/>
</entry>
<entry>
  <key>backup_scheduler_copies_to_keep</key>
  <value>5</value>
</entry>
<entry>
  <key>controller_enable_autoscan</key>
  <value>>true</value>
</entry>
<entry>
  <key>controller_enable_metering</key>
  <value>>true</value>
</entry>
<entry>
  <key>controller_enable_monitoring</key>
  <value>>true</value>
</entry>
<entry>
  <key>controller_metering_interval</key>
  <value>3600</value>
</entry>
<entry>
  <key>controller_netapp_firmware_version</key>
  <value>8.1.1</value>
</entry>
</properties>
</property_info>

```

2. Verify that `controller_enable_metering` is set equal to `true`, and that `controller_metering_interval` is set to the required value.

Change the default port allocation parameters using the REST API

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

Before you begin

- Authenticate with the ViPR Controller REST API as a System Administrator. See the *ViPR Controller REST API Virtual Data Center Configuration Guide* on the [ViPR Controller Product Documentation Index](#).
- Verify [how often ViPR Controller will collect data from the array on page 184](#).
- Verify that [metrics collection from the arrays is enabled on page 184](#).
- The following configuration types can be set for port allocation:
 - `PortAllocationInitiatorCeiling` - Maximum number of initiators that can use the port before new allocations will not be allowed.
 - `PortAllocationVolumeCeiling` - Maximum number of volumes that can use the port before new allocations will not be allowed.

- `PortAllocationPortUtilizationCeiling` - Maximum average port percent busy value (from 0 to 100%) before new allocations are not allowed.
- `PortAllocationCpuUtilizationCeiling` - Maximum average CPU percent busy value (from 0 to 100%) before new allocations are not allowed.
- `PortAllocationDaysToAverageUtilization` - The sample averaging time in days (1 -30 days) .
- `PortAllocationEmaFactor` - The EMA weight for the current sample. The EMA weight is greater than zero and less than or equal to 1.0. A value of 1.0 uses only the current averaging period.
- `PortAllocationMetricsEnabled`

Note

CPU percent busy is not calculated for HDS

- true = use collected metrics and calculate Port percent busy and CPU percent busy.
- 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.

The *EMC ViPR Controller REST API Reference* provides a complete list of parameters of the REST API methods used in this discussion. You can access this document on the [ViPR Controller Product Documentation Index](#) .

In this example, the maximum number of volumes that can use the port before new allocations will not be allowed is changed to 2000000000 for VMAX.

Procedure

1. Retrieve the variables that define the configuration type by sending `GET /config/controller/types/<configuration_type>`.

Each configuration type includes one or more scopes that can be defined. The response includes the type of scope and the valid values for the scope. In addition, each variable that is returned includes the name of the variable and sample values.

Request

```
GET https://<ViPR_Controller_VIP>:4443/config/controller/types/PortAllocationVolumeCeiling
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>
```

Response

```
HTTP 200 OK
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<config_type>
  <name>PortAllocationVolumeCeiling</name>
  <rules>
    <rule>MinConstraintOne</rule>
  </rules>
  <scopes>
    <scope>
      <type>global</type>
      <value>default</value>
    </scope>
  </scopes>
</config_type>
```

```

        <type>systemType</type>
        <value>vnxblock</value>
        <value>vmax</value>
        <value>hds</value>
    </scope>
</scopes>
<type>Integer</type>
</config_type>

```

2. Add your value for PortAllocationVolumeCeiling using POST /config/controller. To change the value for VMAX, the type of scope should be systemType and the value of the scope should be vmax. The new configuration for PortAllocationVolumeCeiling is returned.

Request

```

POST https://<ViPR_Controller_VIP>:4443/config/controller/
Content-Type: application/xml
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<config_create>
  <config_type>PortAllocationVolumeCeiling</config_type>
  <value>2000000000</value>
  <scope>
    <type>systemType</type>
    <value>vmax</value>
  </scope>
</config_create>

```

Response

```

HTTP 200 OK
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<config>
  <id>urn:storageos:CustomConfig:45649fa3-274c-4dbd-
b50c-9e3509ab1b6f:vdcl</id>
  <link rel="self" href="/config/controller/
urn:storageos:CustomConfig:45649fa3-274c-4dbd-
b50c-9e3509ab1b6f:vdcl"/>
  <name>systemType.vmax.PortAllocationVolumeCeiling</name>
  <tags/>
  <config_type>
    <name>PortAllocationVolumeCeiling</name>
    <link rel="self" href="/config/controller/types/
PortAllocationVolumeCeiling"/>
  </config_type>
  <registered>true</registered>
  <scope>
    <type>systemType</type>
    <value>vmax</value>
  </scope>
  <system_default>false</system_default>
  <value>2000000000</value>
</config>

```

The response also includes the URN for your new configuration. If you want to modify the value of your configuration, you can use the URN in the request PUT /config/controller/{ID} to change the value. For example:

```

PUT https://<ViPR_Controller_VIP>:4443/config/controller/
urn:storageos:CustomConfig:45649fa3-274c-4dbd-
b50c-9e3509ab1b6f:vdcl
Content-Type: application/xml

```

```
X-SDS-AUTH-TOKEN: <AUTH_TOKEN>

<config_update>
  <value>1900000000</value>
</config_update>

HTTP 200 OK
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<config>
  <id>urn:storageos:CustomConfig:45649fa3-274c-4dbd-
b50c-9e3509ab1b6f:vdc1</id>
  <link rel="self" href="/config/controller/
urn:storageos:CustomConfig:45649fa3-274c-4dbd-
b50c-9e3509ab1b6f:vdc1"/>
  <name>systemType.vmax.PortAllocationVolumeCeiling</name>
  <tags/>
  <config_type>
    <name>PortAllocationVolumeCeiling</name>
    <link rel="self" href="/config/controller/types/
PortAllocationVolumeCeiling"/>
  </config_type>
  <registered>true</registered>
  <scope>
    <type>systemType</type>
    <value>vmax</value>
  </scope>
  <system_default>false</system_default>
  <value>1900000000</value>
</config>
```

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](#).

HDS performance metrics

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

Table 21 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 <i>yyyyMMddHHmmss.SSSSSSsutc</i> where: <ul style="list-style-type: none"> • yyyy - is a 4 digit year • MM - is the month • dd - is the day of the month • HH - is the hour (24 hour clock)

Table 21 Performance metrics collected on HDS (continued)

Metric	Variable	Description
		<ul style="list-style-type: none"> mm - is the minutes ss - is the seconds mmmmm - 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.

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 22 Performance metrics collected on VMAX

Metric	Variable	Description
FEPort, FEAdapt: StatisticTime	sampleTime	<p>A string representing the current time with the format, <i>yyyyMMddHHmmss.SSSSSSsutc</i>, where:</p> <ul style="list-style-type: none"> 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 minute; ss - is the second mmmmm - 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.

VNX for Block performance metrics

The table describes the metrics that are collected on VNX 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 on page 188](#) for more information.

Table 23 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	The 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 <i>yyyyMMddHHmmss.SSSSSSsutc</i> where: <ul style="list-style-type: none"> • 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.