



Technical Report

# Storage Agility and Integration Simplicity

## NetApp OnCommand API Services Integration with HP Operations Orchestration

Akash Shukla, NetApp  
August 2016 | TR-4533

### Abstract

The goal of this document is to give IT administrators and operational staff information about how NetApp® shared storage integrates with HP Operations Orchestration (HP OO) by using NetApp OnCommand® API Services (API-S). It provides users with a perspective on how to use the product and describes the simplicity with which the integration can be implemented. The document covers the integration of storage provisioning flow and its consumption by using HP OO software. The architecture described in this document provides clean separation between different administrative roles in a data center.

## TABLE OF CONTENTS

<b>1</b>	<b>Executive Summary.....</b>	<b>3</b>
<b>2</b>	<b>Introduction.....</b>	<b>3</b>
2.1	Scenario and Use Case .....	3
<b>3</b>	<b>Software Components and Integration .....</b>	<b>5</b>
3.1	NetApp OnCommand API Services .....	5
3.2	HP Operations Orchestration .....	6
3.3	NetApp OnCommand Performance Manager .....	6
3.4	NetApp OnCommand Unified Manager .....	7
<b>4</b>	<b>Integrating OnCommand API-S with HP Operations Orchestration.....</b>	<b>7</b>
4.1	Creating Your First Flow (Provisioning a Volume) .....	7
<b>5</b>	<b>Conclusion .....</b>	<b>15</b>
	<b>References.....</b>	<b>16</b>

## LIST OF FIGURES

Figure 1)	Use cases for integrating HP OO and OnCommand API Services. ....	4
Figure 2)	Sample list of APIs that are available with OnCommand API Services. ....	5
Figure 3)	HP Operations Orchestration functional architecture. ....	6
Figure 4)	HP Operations Orchestration author environment. ....	8
Figure 5)	HPOO designer pane for creating a new workflow.....	9
Figure 6)	HP OO Command Inspector pane for editing command attributes. ....	10
Figure 7)	XPath query filter for extracting SVM names. ....	10
Figure 8)	Screenshot of the subflow for selecting an SVM. ....	11
Figure 9)	Screenshot of the subflow for selecting an aggregate.....	12
Figure 10)	HP OO Command Inspector pane for a <code>POST</code> request.....	13
Figure 11)	Subflow for creating a new volume and for checking the job status .....	14
Figure 12)	End result of the flow.....	14
Figure 13)	HP OO Debug pane. ....	15

# 1 Executive Summary

NetApp OnCommand API Services is the single REST (representational state transfer) endpoint to monitor and manage all underlying NetApp ONTAP® resources. OnCommand API Services provides API offerings for storage management and control operations, which include monitoring, storage provisioning, and data management in your NetApp environment. It provides a programmatic way through which third-party applications can issue requests to retrieve information about the storage environment and provision storage resources.

This document describes how to leverage the comprehensive set of REST APIs that OnCommand API Services offers with HP Operations Orchestration (HP OO) through the REST interface. You can use these APIs to execute and manage key storage operations in your data center. This document contains:

- An overview of the integration of HP OO and OnCommand API Services
- A demonstration of how to simplify storage automation in complex and growing environments by using a simple, service-oriented interface
- A description of how HP OO interacts with OnCommand API Services and demonstrates the consumption of RESTful APIs
- Sample use cases that demonstrate the consumption of OnCommand API Services in HP OO flows

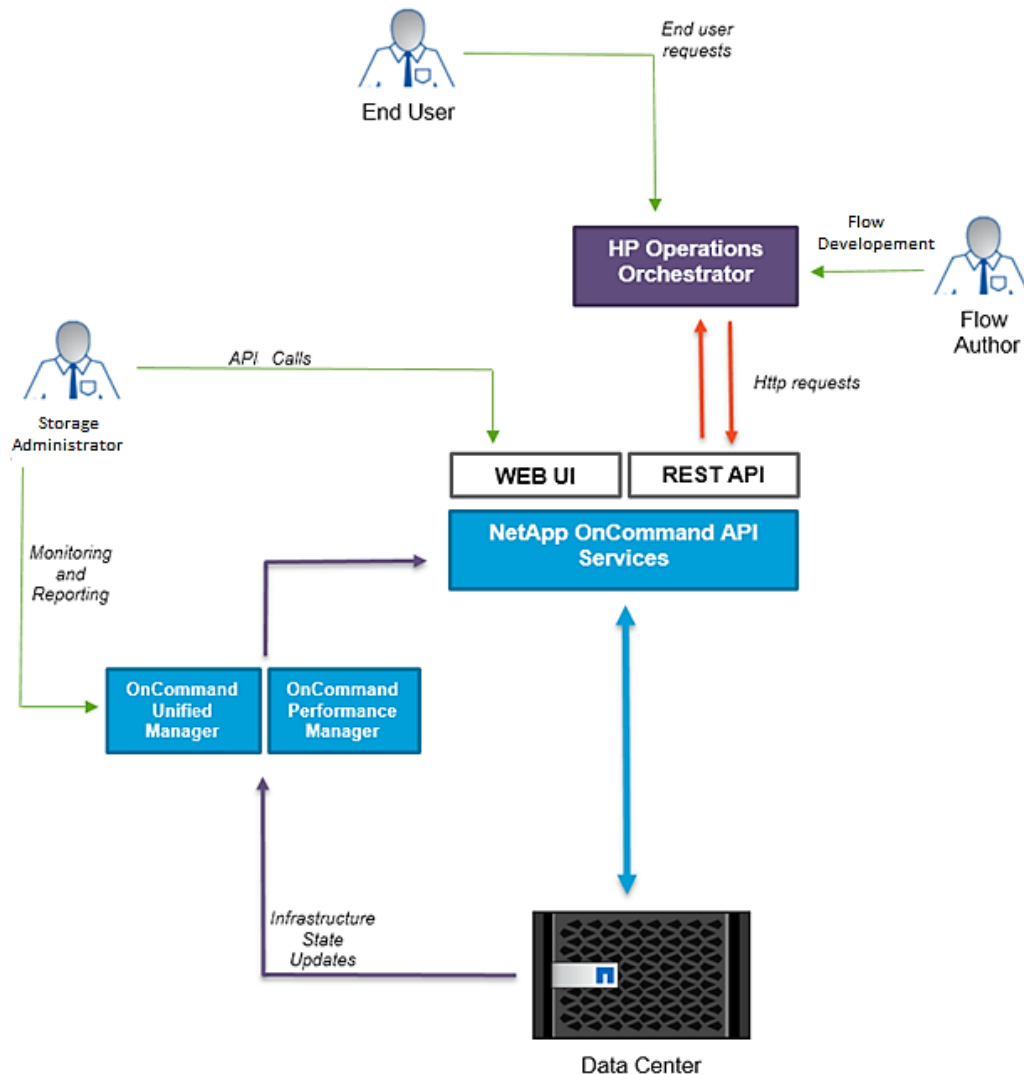
## 2 Introduction

This document illustrates the much-applied approach of integrating HP OO with NetApp OnCommand API Services. The primary focus is to demonstrate monitoring and managing your storage in an efficient NetApp shared storage infrastructure. This approach is well established for using NetApp storage resources in an enterprise organization.

### 2.1 Scenario and Use Case

The integration of HP OO and OnCommand API Services is described from the perspective of a large enterprise. We assume the presence of the HP OO framework. In this context, we describe the use cases for creating HP OO flows and for executing some of them. Figure 1 shows the scenario of and use cases for integrating HP OO and OnCommand API Services.

Figure 1) Use cases for integrating HP OO and OnCommand API Services.



In this scenario, there are four roles:

- **Flow author.** The flow author creates and debugs flows in HP OO Studio. The flow author leverages HP OO out-of-the-box content, content that was developed by action developers, and the utilities that are provided.
- **End user.** The end user triggers and monitors the flows. The end user can access entitled HP OO flows directly through HP OO Central or indirectly through an embedded web UI in another application.
- **Storage administrator.** The storage administrator is responsible for the complete management and monitoring of the data center that contains shared storage systems.
- **Storage operator.** The storage operator is responsible for proactive performance and monitoring tasks of the data center that contains shared storage systems.

## 3 Software Components and Integration

This section describes the integration of the individual software components to implement storage monitoring and provisioning. The focus of HP OO is to automate IT processes and reduce manual activity. It orchestrates across the functions of multiple management technology frameworks for the execution of the end-to-end workflow. Adapters interface between HP OO flows, IT components, and management frameworks.

### 3.1 NetApp OnCommand API Services

NetApp OnCommand API Services (API-S) provides an API through which partner applications can issue programmatic requests to execute specified data storage monitoring and managing operations on NetApp ONTAP storage systems.

To support data center monitoring for systems that run ONTAP, API-S contains the following set of REST APIs:

- **Provisioning APIs**, which enable you to provision and allocate storage resources in your data center.
- **Monitoring APIs**, which enable you to retrieve information about logical and physical storage resources in the data center.
- **Event APIs**, which enable you to retrieve information about any events that occurred in the data center.
- **Performance monitoring APIs**, which enable you to retrieve performance-related information about the data center storage resources.

Figure 2 lists a sample of APIs that are available with OnCommand API Services.

Figure 2) Sample list of APIs that are available with OnCommand API Services.

#### OnCommand API Services REST APIs

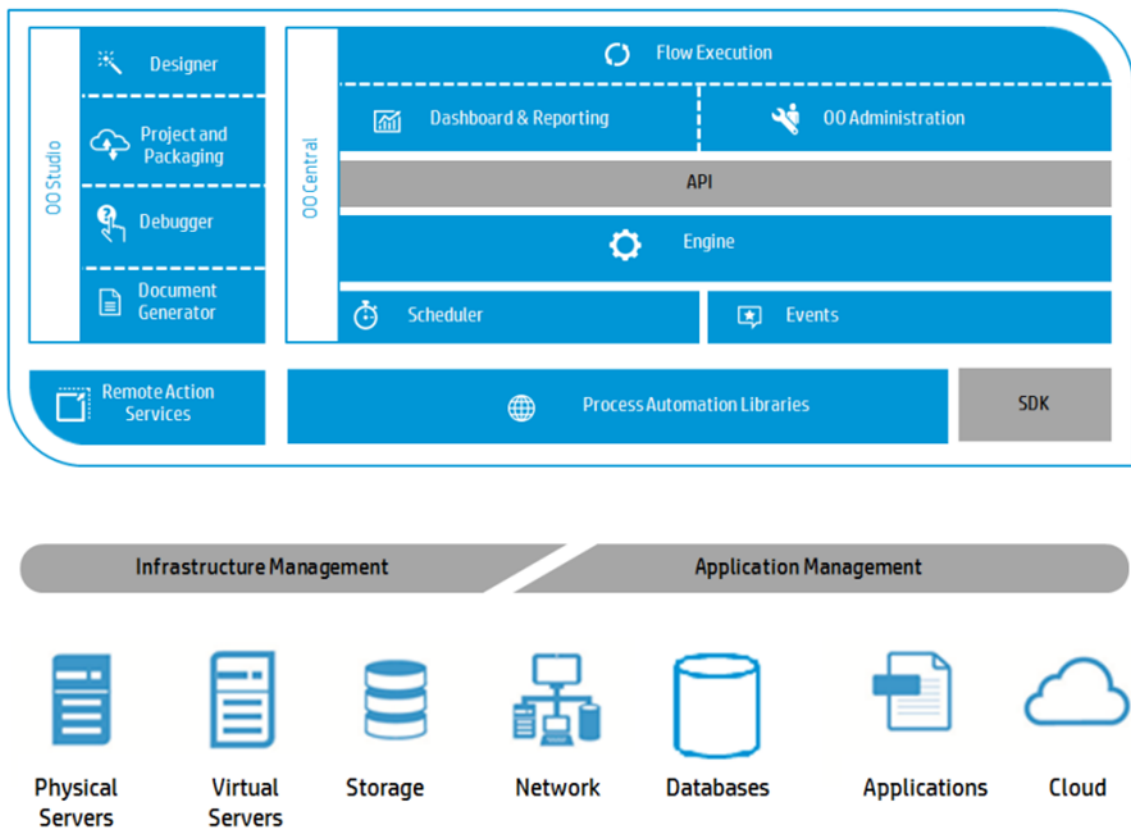
admin : Admin operations [Role: Administrator]		Show/Hide	List Operations	Expand Operations	Raw
jobs : Job operations [Roles: Administrator, Operator]		Show/Hide	List Operations	Expand Operations	Raw
ontap : Clustered Data ONTAP operations [Roles: Administrator, Monitor, Operator]		Show/Hide	List Operations	Expand Operations	Raw
GET	/ontap/aggregates	Retrieve all the aggregates			
GET	/ontap/aggregates/{key}	Retrieve the specified aggregate			
GET	/ontap/aggregates/{key}/volumes	Retrieve all the volumes of specified aggregate			
GET	/ontap/aggregates/{key}/events	Retrieve all the events of specified aggregate			
GET	/ontap/aggregates/metrics	Retrieve all the metrics of all aggregates			
GET	/ontap/aggregates/{resource_key}/metrics	Retrieve all the metrics of specified aggregate			
GET	/ontap/aggregate-disk-relationships	Retrieve all the aggregate-disk-relationships			
GET	/ontap/aggregate-disk-relationships/{key}	Retrieve the specified aggregate-disk-relationship			
GET	/ontap/cifs-shares	Retrieve all the cifs-shares			
POST	/ontap/cifs-shares	Creates cifs-shares			
GET	/ontap/cifs-shares/{key}	Retrieve the specified cifs-share			
DELETE	/ontap/cifs-shares/{key}	Deletes cifs-shares			
PUT	/ontap/cifs-shares/{key}	Modifies cifs-shares			
GET	/ontap/cifs-shares/{key}/cifs-share-acls	Retrieve all the cifs-share-acls of specified cifs-share			
GET	/ontap/cifs-share-acls	Retrieve all the cifs-share-acls			
POST	/ontap/cifs-share-acls	Creates cifs-share-acls			

### 3.2 HP Operations Orchestration

HP Operations Orchestration (HP OO) software integrates enterprise management systems and automates standard operations as well as critical IT infrastructure library processes through the REST programming interface. HP OO provides out-of-the-box workflows for data centers and can also be extended and customized to suit particular deployment requirements. This next-generation IT process automation solution is designed to increase automation both in traditional data centers and in hybrid cloud environments. HP OO is a system for creating and using actions in structured sequences (called *flows*), which maintain your IT resources by performing repetitive tasks and executing the flows with the provided specifications. Figure 3 shows the functional architecture of HP OO.

Figure 3) HP Operations Orchestration functional architecture.

#### Functional Architecture



### 3.3 NetApp OnCommand Performance Manager

NetApp OnCommand Performance Manager provides performance monitoring and event root-cause analysis of systems that run ONTAP software. It is the performance management part of OnCommand Unified Manager. Performance Manager helps you identify workloads that are overusing cluster components and are decreasing the performance of other workloads on the cluster. It alerts you to these performance events so that you can take corrective action and return performance back to normal operation. You can view and analyze events in the Performance Manager web UI or view them in the OnCommand Unified Manager UI.

### 3.4 NetApp OnCommand Unified Manager

NetApp OnCommand Unified Manager (OCUM) is the primary monitoring and reporting tool for NetApp storage systems. It provides both a human interface and an API for integration with other management software. The API uses well-defined XML objects for communication, providing transparency and interoperability.

Data collected by OCUM is queried by API-S to provide a subset of relevant information for use with the flow's decision. Multiple OCUM instances can feed into a single API-S instance. The storage administrator can directly access OCUM for reporting and monitoring of the underlying NetApp storage.

## 4 Integrating OnCommand API-S with HP Operations Orchestration

This section provides a technical overview of the integration of HP OO and NetApp OnCommand API Services. Following the REST industry standard, both OnCommand API-S and HP OO can communicate with each other by issuing `GET`, `POST`, `PUT`, and `DELETE` HTTP requests.

HP OO allows the flow architect to translate requirement specifications onto structured sequences, which are called a *flow*. The flow is an operation sequence with various connected and predefined library commands, such as the `HTTP Client Get` request, `List Item Grabber`, and so on. Each of these commands has different command-specific attributes. Tweaking these attributes helps in realizing the required behavior from the command. For example, the body attribute of the `Http Client Post` request specifies that the content be posted on the URL that is defined in the URL attribute.

Each command has a success and a failure condition, which is also visualized as an annotation of a check mark or a cross mark, respectively, on the command. The condition is activated according to the result from the execution of the command. You can adjust the flow direction by dragging and connecting these conditional arrows to the next command, or you can also connect them to endpoints, such as `Resolved: success`.

Section 4.1 describes one of the use cases in detail.

### 4.1 Creating Your First Flow (Provisioning a Volume)

#### Assumption

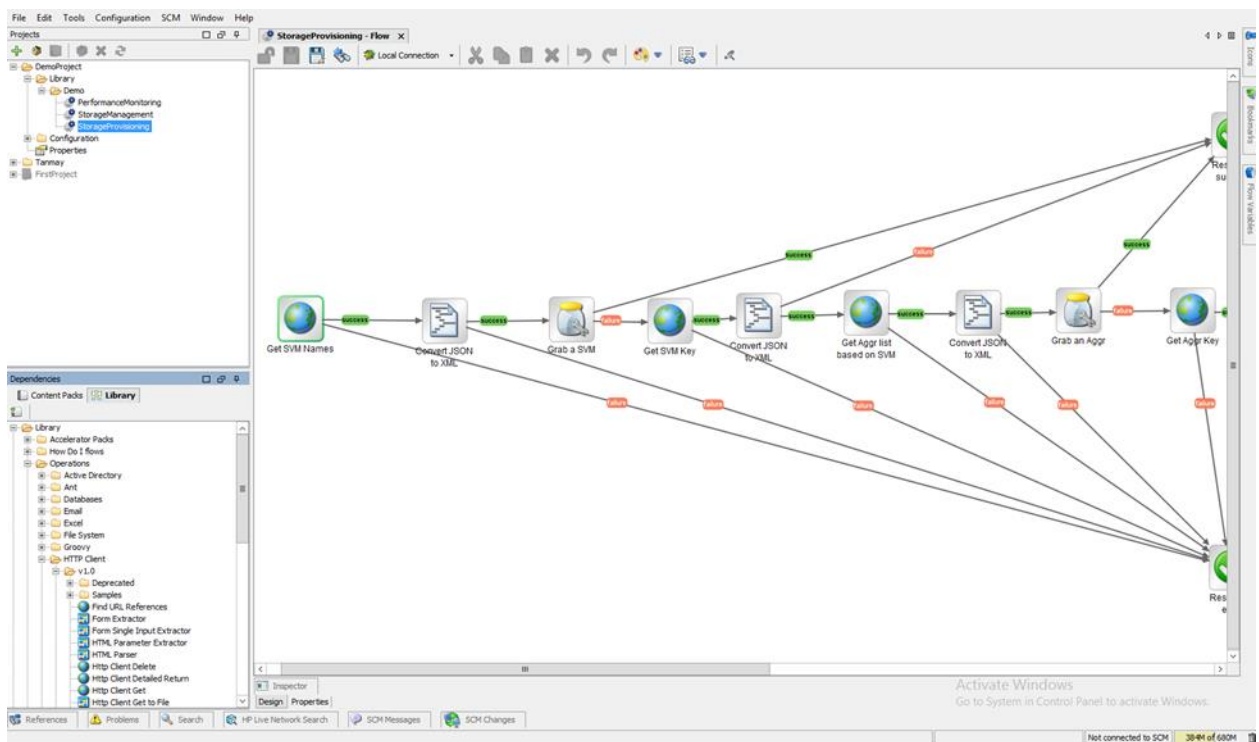
Let's assume that you have completed the installation of OnCommand API-S and HP OO on separate machines. This section guides you on how to create your first flow in the HP OO Studio authoring environment and how to execute the flow. The designer studio allows you to fabricate the flow, and HP OO Central Studio helps you to execute the flow remotely, or you can execute the flow directly from the local designer instance.

If the setup is not complete, follow the API-S installation and best practices guide to finish the installation of OnCommand API Services. Also, follow the best practices in the introduction and installation guide for HP Operations Orchestration.

#### Step 1: Create the Flow

Launch your HP OO designer by using the shortcut icon. On the top left pane, you see the project explorer. On the bottom left pane is the object explorer, where you can find all the required commands. On the center right is the designer pane, where you create your flow. Figure 4 shows the HP OO author environment.

Figure 4) HP Operations Orchestration author environment.



The purpose of this flow is to provision a volume on the NetApp storage system by using API-S. The abstract for the flow would be as follows:

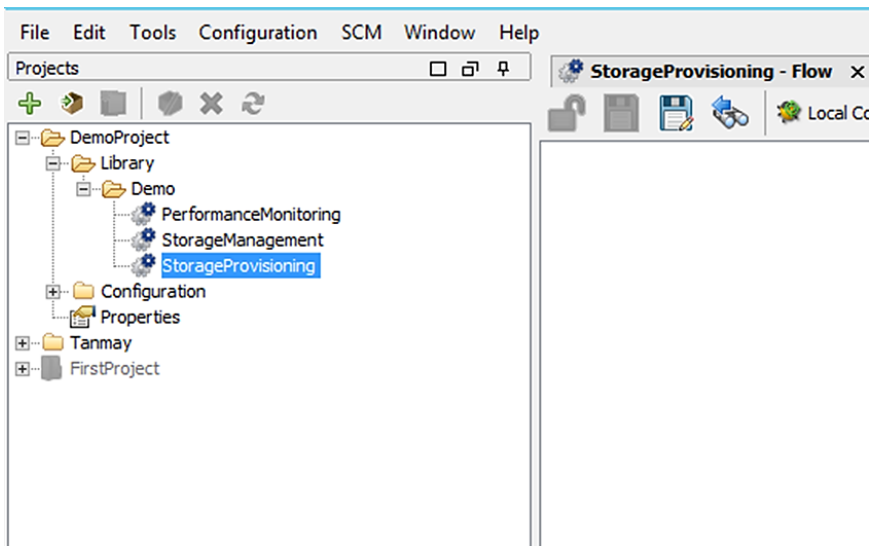
1. Search for all the storage virtual machines (SVMs) on the storage controller and let the user select one of the SVMs from drop-down list.
2. Depending on the SVM selected, search for all the aggregates on the storage controller and let the user select one of the aggregates from the drop-down list.
3. Prompt the user to input the volume name and volume size.
4. Run the command to provision a new volume on the storage controller by using the result of the previous steps. The output is the job initiation IP for the process.
5. Poll in the job IP and wait for the process of creating a new volume on the storage server to complete.
6. If the process is successful, prompt the volume created; otherwise, define the reason for failure.

## Design the Flow

The first subflow would be to create a new project called `DemoProject` in the project explorer. Right-click to create a new folder demonstration and then create a new flow called `StorageProvisioning`, which creates a blank flow. Next, author the complete `StorageProvisioning` flow. Figure 5 represents the HP OO designer pane for creating a new workflow.



Figure 5) HPOO designer pane for creating a new workflow.



After a blank flow has been created, drag the `Http Client Get` command under the `Operations > HttpClient` category from the bottom left Library pane to the designer pane. Right-click this command and select it as the start step. Double-click the command, and the Command Inspector appears at the bottom. Perform the following steps:

7. Select the URL attribute and in the right pane, select the Use Constant radio button and specify the URL as [https://api-server/api/1.0/ontap/storage\\_vms](https://api-server/api/1.0/ontap/storage_vms).
8. Select the Authentication mode as Basic and specify the user name and password credential attributes.
9. Edit `x509HostNameVerifier` as `allow_all`.
10. Rename the step as Get SVM Names.

The command attributes are complete. The output of this result would be a JSON output with the records of all the SVMs that are available. Figure 6 shows the HP OO Command Inspector pane for editing the attributes of the command.

Figure 6) HP OO Command Inspector pane for editing command attributes.

Input	Sensitiv...	Required	Type	Assign From	Otherwise
url	<input type="checkbox"/>	<input checked="" type="checkbox"/>	url	url	Use the constant: https://10.235...
authType	<input type="checkbox"/>	<input type="checkbox"/>	authType	authType	Use the constant: Basic
username	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<not assigned>	<not assigned>	Use the constant: admin
password	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<not assigned>	<not assigned>	Use the constant: *****
preemptiveAuth	<input type="checkbox"/>	<input type="checkbox"/>	<not assigned>	<not assigned>	Use the constant: false
proxyHost	<input type="checkbox"/>	<input type="checkbox"/>	<not assigned>	<not assigned>	Use the constant:
proxyPort	<input type="checkbox"/>	<input type="checkbox"/>	<not assigned>	<not assigned>	Use the constant: 8080
proxyUsername	<input type="checkbox"/>	<input type="checkbox"/>	<not assigned>	<not assigned>	Use the constant:
proxyPassword	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<not assigned>	<not assigned>	Use the constant:
trustAllRoots	<input type="checkbox"/>	<input type="checkbox"/>	<not assigned>	<not assigned>	Use the constant: true
x509HostNameVerifier	<input type="checkbox"/>	<input type="checkbox"/>	<not assigned>	<not assigned>	Use the constant: allow_all
trustKeystore	<input type="checkbox"/>	<input type="checkbox"/>	trustKeystore	trustKeystore	Use the constant: true
trustPassword	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<not assigned>	<not assigned>	Use the constant: *****
keystore	<input type="checkbox"/>	<input type="checkbox"/>	<not assigned>	<not assigned>	Use the constant: true
keystorePassword	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<not assigned>	<not assigned>	Use the constant: *****
connectTimeout	<input type="checkbox"/>	<input type="checkbox"/>	<not assigned>	<not assigned>	Use the constant: 0
socketTimeout	<input type="checkbox"/>	<input type="checkbox"/>	<not assigned>	<not assigned>	Use the constant: 0
useCookies	<input type="checkbox"/>	<input type="checkbox"/>	<not assigned>	<not assigned>	Use the constant: true
keepAlive	<input type="checkbox"/>	<input type="checkbox"/>	<not assigned>	<not assigned>	Use the constant: true
connectionsMaxPerR...	<input type="checkbox"/>	<input type="checkbox"/>	<not assigned>	<not assigned>	Use the constant: 2
connectionsMaxTotal	<input type="checkbox"/>	<input type="checkbox"/>	<not assigned>	<not assigned>	Use the constant: 20

Name: url Input Type: Sing

Assign from Variable: url

Otherwise: Constant Value: https://10.235.224.206:8443/api/1.0/ontap/storage-vm

☒ Use Constant

☐ Prompt User

☐ Use Previous Step Result

☐ System Account

☐ Logged-in User Credentials

The response that is shown in Figure 6 contains sample information; the SVM names must be filtered out. Take the result from the last command and convert the JSON output to XML format. Use the Convert JSON to XML step and select the input for it as the result from the preceding step. In the Convert JSON to XML step, create a new result field, which is an XPath query that filters the names of SVMs (XPath query: rootTag/result/records/item/name) from the total result. Figure 7 shows the XPath query filter for extracting SVM names.

Figure 7) XPath query filter for extracting SVM names.

Inspector

Step Name: Convert JSON to XML

Inputs Results Display Description Advanced Scriptlet

Step Results > SVMNameList

Details for: XPath Query

Filters an XML document based on an xpath query and returns the results of the query.

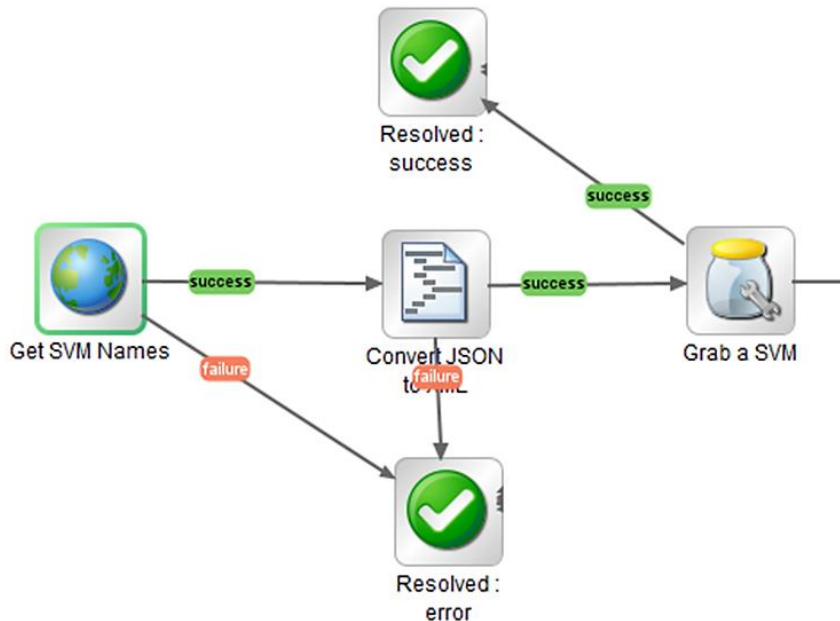
XPath Query: rootTag/result/records/item/name

XPath: rootTag/result/records/item/

Replace All '<name>' with "

Next, use this SVM name list to pass through the List Item Grabber command and to let the user get the prompt to select the SVM name from a drop-down list and pass it to the next step. Figure 8 shows a screenshot of the subflow for selecting an SVM.

Figure 8) Screenshot of the subflow for selecting an SVM.

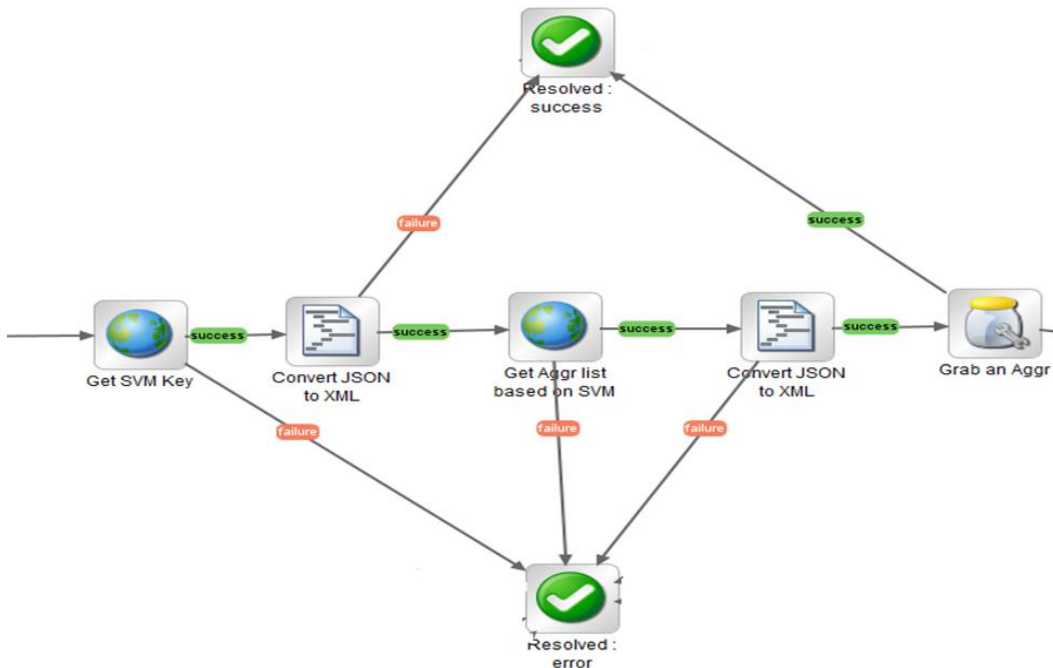


Now we have the SVM name. Next, to get the SVM unique ID, put an HTTP Client Get request with the URL as `https://api-server/api/1.0/ontap/storage_vms?name={selectedSVMname}`. Filter the SVM key from the output.

We have completed the first subflow of selecting an SVM.

Extend the flow to make use of the SVM key to find the list of aggregates and to again let the user select one aggregate from the drop-down list. The design is similar to selecting an SVM, with the only difference being that the SVM is replaced by an aggregate. Figure 9 shows a screenshot of the subflow for selecting an aggregate.

Figure 9) Screenshot of the subflow for selecting an aggregate.



The second subflow for selecting an aggregate is now complete.

For user input of the volume name and volume size, use the `Evaluate Expression` command because it lets the user input a value. Use this entered value as a flow variable and then you can use it for the next step.

The third subflow for getting the input of the volume name and size is now complete.

Because we have the outputs from previous steps stored in the flow variables, now we drag an `Http Client Post` command into the designer pane. Because we want to create a new volume, the URL of this command should be `https://api-server/api/1.0/ontap/volumes`. Provide the required user name and password credentials as you did earlier in the `GET` commands. However, here you have to provide the body of the `Post` request as the list of fields that are required to create a new volume. The compulsory four parameters for creating a new volume in NetApp ONTAP are the SVM key, aggregate key, volume name, and volume size.

The body of the post request is as follows:

```
"storage_vm_key": "{SelectedSVMKey}", "aggregate_key": "{SelectedAggrKey}", "name": "{VolName}",  
"volume_size": "{VolSize}" ]
```

Figure 10 shows the HP OO Command Inspector pane for the preceding command.

Figure 10) HP OO Command Inspector pane for a POST request.

Inspector

Step Name: POST Create a Volume

Inputs Results Display Description Advanced Scriptlet

Input	Sensitiv...	Required	Type	Assign From	Otherwise
url	<input type="checkbox"/>	<input checked="" type="checkbox"/>		<not assigned>	Use the constant: https://10.235...
authType	<input type="checkbox"/>	<input type="checkbox"/>		<not assigned>	Use the constant: Basic
username	<input type="checkbox"/>	<input checked="" type="checkbox"/>		<not assigned>	Use the constant: admin
password	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<not assigned>	Use the constant: *****
preemptiveAuth	<input type="checkbox"/>	<input type="checkbox"/>		<not assigned>	Use the constant: true
proxyHost	<input type="checkbox"/>	<input type="checkbox"/>		<not assigned>	Use the constant: true
proxyPort	<input type="checkbox"/>	<input type="checkbox"/>		<not assigned>	Use the constant: 8080
proxyUsername	<input type="checkbox"/>	<input type="checkbox"/>		<not assigned>	Use the constant: true
proxyPassword	<input checked="" type="checkbox"/>	<input type="checkbox"/>		<not assigned>	Use the constant: true
trustAllRoots	<input type="checkbox"/>	<input type="checkbox"/>		<not assigned>	Use the constant: true
x509HostnameVerifier	<input type="checkbox"/>	<input type="checkbox"/>		<not assigned>	Use the constant: allow_all
trustKeystore	<input type="checkbox"/>	<input type="checkbox"/>		<not assigned>	Use the constant: true
trustPassword	<input checked="" type="checkbox"/>	<input type="checkbox"/>		<not assigned>	Use the constant: *****
keystore	<input type="checkbox"/>	<input type="checkbox"/>		<not assigned>	Use the constant: true
keystorePassword	<input checked="" type="checkbox"/>	<input type="checkbox"/>		<not assigned>	Use the constant: *****
connectTimeout	<input type="checkbox"/>	<input type="checkbox"/>		<not assigned>	Use the constant: 0
socketTimeout	<input type="checkbox"/>	<input type="checkbox"/>		<not assigned>	Use the constant: 0
useCookies	<input type="checkbox"/>	<input type="checkbox"/>		<not assigned>	Use the constant: true
keepAlive	<input type="checkbox"/>	<input type="checkbox"/>		<not assigned>	Use the constant: true
connectionsMaxPerR...	<input type="checkbox"/>	<input type="checkbox"/>		<not assigned>	Use the constant: 2
connectionsMaxTotal	<input type="checkbox"/>	<input type="checkbox"/>		<not assigned>	Use the constant: 20
headers	<input type="checkbox"/>	<input type="checkbox"/>		<not assigned>	Use the constant: true
sourceFile	<input type="checkbox"/>	<input type="checkbox"/>		<not assigned>	Use the constant: true
body	<input type="checkbox"/>	<input checked="" type="checkbox"/>		body	Use the constant: { "aggregate_key": "\${AggrKey}", "size": "\${VolSize}", "storage_vm_key": "\${SVMKey}", "name": "\${VolName}" }
contentType	<input type="checkbox"/>	<input checked="" type="checkbox"/>		<not assigned>	Use the constant: application/json

Name: body Input Type: List of Values

Assign from Variable: body

Otherwise:

☒ Use Constant

☐ Prompt User

☐ Use Previous Step Result

Constant Value:

```
{ "aggregate_key": "${AggrKey}", "size": "${VolSize}", "storage_vm_key": "${SVMKey}", "name": "${VolName}" }
```

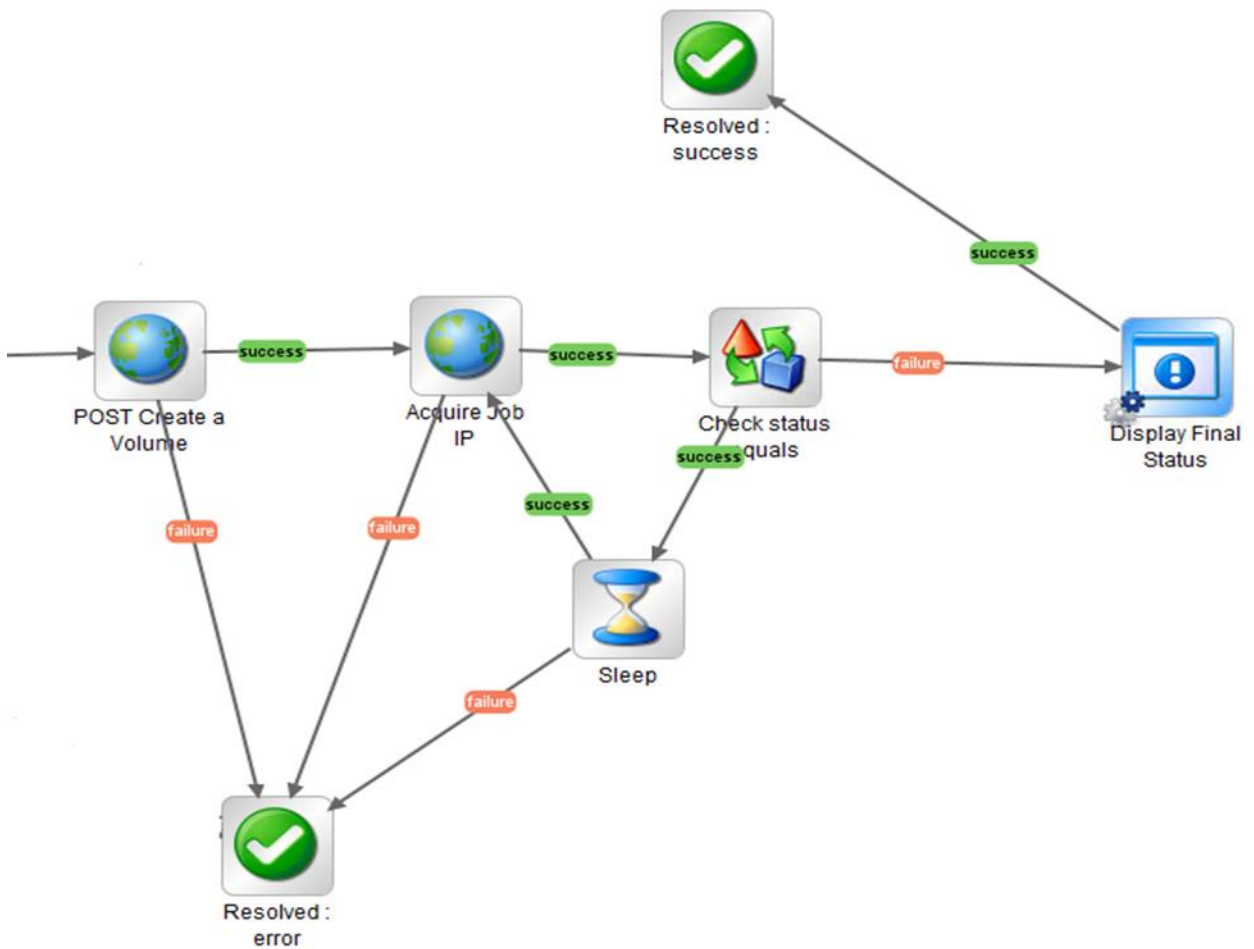
**Note:** The content inside the curly brackets ({} ) is exchanged with the correct values at run time.

At the end of the POST request, we obtain the output that provides a process job IP that was generated for the creation of that volume.

From the result, filter out that job IP and put a GET request that pools the result of that step; check the Status tag content. As soon as that tag changes from Pending to Completed, prompt the user through the Status Message command that the process has completed and specify the new volume details.

Check for any warning while saving the flow. Figure 11 shows the subflow for creating a new volume and for checking the job status.

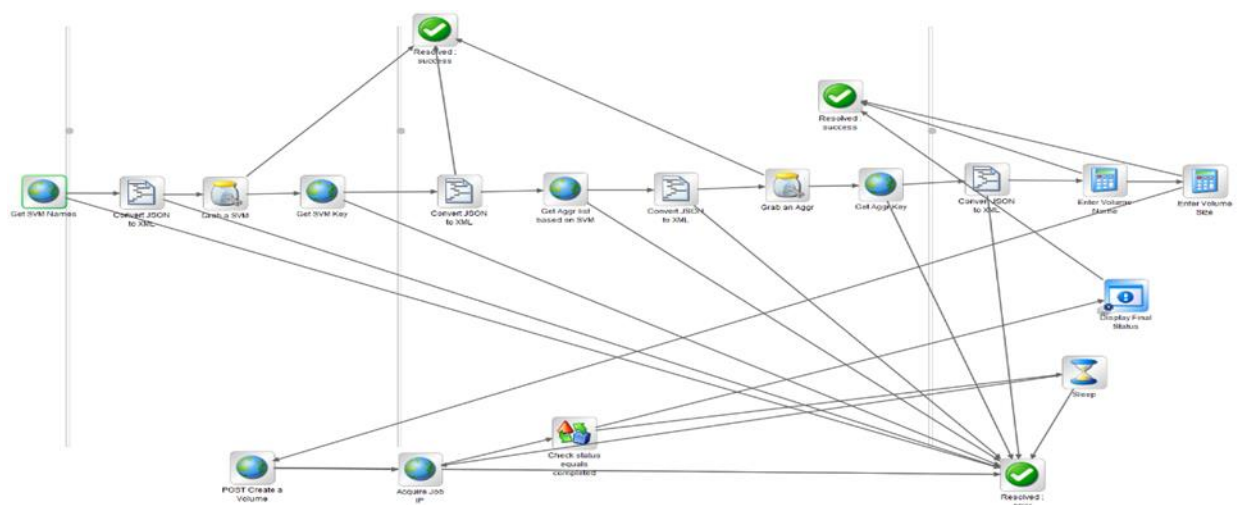
Figure 11) Subflow for creating a new volume and for checking the job status



Use the `Resolved: success` and `Resolved: error` commands for the flow end.

Figure 12 shows what the flow ultimately looks like.

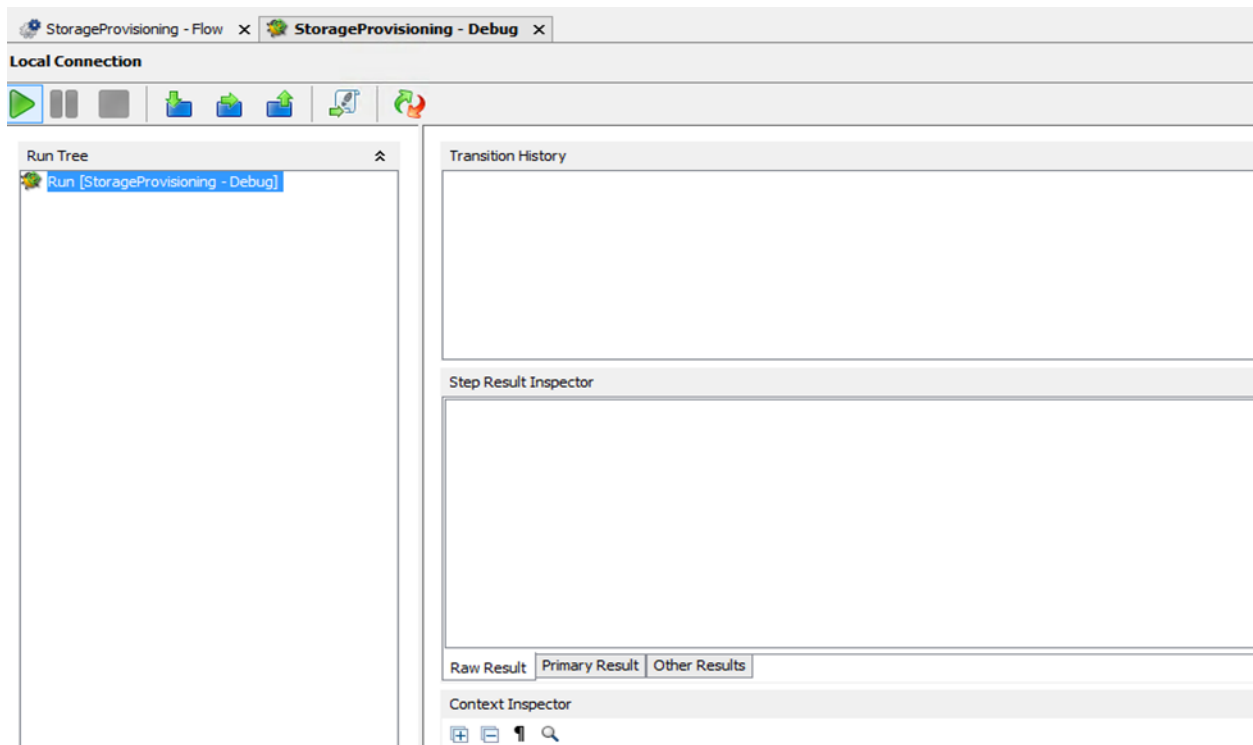
Figure 12) End result of the flow.



## Step 2: Debug the Flow

After your flow design has been completed, you might want to debug the flow for any errors. HP OO provides a local debugging feature that you can start by clicking the Local Debug icon on the toolbar, which is on top of the designer pane. Figure 13 shows the HP OO Debug pane.

Figure 13) HP OO Debug pane.



When you click the Local Debug icon, you are directed to a new debug page. The top left toolbar shows the icon for running the flow. When you click the button, your flow begins from the starting step and continues through the end step. The center console shows the Run Tree and Transition History and helps you to identify bugs. The user prompt appears on the screen whenever the flow passes a user prompt state or requires user attention.

## Step 3: Execute the Flow

After testing and debugging your flow, you are ready to publish the flow. This published flow can either be run in the local machine through the local debugging tool or be run remotely through the HP OO Studio Central tool. The Studio Central tool is available only with the enterprise-licensed edition of HP OO.

To run your flow, go to the HP OO Studio Central page and search for the project and then your flow. Right-click the flow and select the Run the Flow option. The flow runs similar to the debugging stage without exposing the run-tree details.

## 5 Conclusion

This document:

- Describes how to make use of NetApp OnCommand API Services in HP Operations Orchestration flows to manage and monitor the underlying NetApp storage

- Describes a walk-through of the individual steps that are required to create an HP Operations Orchestration flow for provisioning a volume on the storage
- Provides a foundation for an end-to-end storage automation scenario that provides storage agility and integration simplicity of any third-party tool with OnCommand API Services

For an experienced HP OO consultant with a basic knowledge of OnCommand API Services, performing the steps described in this document should take only a few minutes. That estimate assumes that both HP OO and OnCommand API Services are preinstalled.

## References

The following references were used in this TR:

- NetApp OnCommand API Services  
<http://www.netapp.com/us/products/management-software/oncommand/api.aspx>
- NetApp OnCommand API Services Installation Guide  
[https://library.netapp.com/ecm/ecm\\_download\\_file/ECMLP2310708](https://library.netapp.com/ecm/ecm_download_file/ECMLP2310708)
- HP Operations Orchestration Installation Guide  
[https://nast01pccache.saas.hpe.com/asset/resources/pd/oo/1um1459950215/OO10.20\\_Installation\\_Guide.pdf](https://nast01pccache.saas.hpe.com/asset/resources/pd/oo/1um1459950215/OO10.20_Installation_Guide.pdf)
- Guide to Authoring HP Operations Orchestration Flows  
[https://nast01pccache.saas.hpe.com/asset/resources/pd/oo/wf61459946196/Studio\\_AuthorsGuide\\_4.pdf](https://nast01pccache.saas.hpe.com/asset/resources/pd/oo/wf61459946196/Studio_AuthorsGuide_4.pdf)



Refer to the [Interoperability Matrix Tool \(IMT\)](#) on the NetApp Support site to validate that the exact product and feature versions described in this document are supported for your specific environment. The NetApp IMT defines the product components and versions that can be used to construct configurations that are supported by NetApp. Specific results depend on each customer's installation in accordance with published specifications.

## Copyright Information

Copyright © 1994–2016 NetApp, Inc. All rights reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.277-7103 (October 1988) and FAR 52-227-19 (June 1987).

## Trademark Information

NetApp, the NetApp logo, Go Further, Faster, AltaVault, ASUP, AutoSupport, Campaign Express, Cloud ONTAP, Clustered Data ONTAP, Customer Fitness, Data ONTAP, DataMotion, Flash Accel, Flash Cache, Flash Pool, FlashRay, FlexArray, FlexCache, FlexClone, FlexPod, FlexScale, FlexShare, FlexVol, FPolicy, GetSuccessful, LockVault, Manage ONTAP, Mars, MetroCluster, MultiStore, NetApp Fitness, NetApp Insight, OnCommand, ONTAP, ONTAPI, RAID DP, RAID-TEC, SANshare, SANtricity, SecureShare, Simplicity, Simulate ONTAP, SnapCenter, SnapCopy, Snap Creator, SnapDrive, SnapIntegrator, SnapLock, SnapManager, SnapMirror, SnapMover, SnapProtect, SnapRestore, Snapshot, SnapValidator, SnapVault, SolidFire, StorageGRID, Tech OnTap, Unbound Cloud, WAFL, and other names are trademarks or registered trademarks of NetApp Inc., in the United States and/or other countries. All other brands or products are trademarks or registered trademarks of their respective holders and should be treated as such. A current list of NetApp trademarks is available on the web at <http://www.netapp.com/us/legal/netapptmlist.aspx>. TR-4533-0816