

# Dell EMC NetWorker

Version 19.1.x

## REST API Getting Started Guide

302-005-704

REV 02

August 2019

Copyright © 2016-2019 Dell Inc. or its subsidiaries. All rights reserved.

Dell 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.” DELL 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 DELL SOFTWARE DESCRIBED IN THIS PUBLICATION REQUIRES AN APPLICABLE SOFTWARE LICENSE.

Dell Technologies, Dell, EMC, Dell EMC and other trademarks are trademarks of Dell Inc. or its subsidiaries. Other trademarks may be the property of their respective owners. Published in the USA.

Dell EMC  
Hopkinton, Massachusetts 01748-9103  
1-508-435-1000 In North America 1-866-464-7381  
[www.DellEMC.com](http://www.DellEMC.com)

# CONTENTS

<b>Tables</b>		<b>5</b>
	<b>Preface</b>	<b>7</b>
<b>Chapter 1</b>	<b>Introduction</b>	<b>13</b>
	Overview.....	14
	API Versioning.....	14
	Authentication and Authorization.....	16
	Limitations.....	17
<b>Chapter 2</b>	<b>API Requests and Response</b>	<b>19</b>
	URI Format.....	20
	NetWorker API Requests.....	21
	Handling the API Response.....	22
<b>Chapter 3</b>	<b>Consuming the NetWorker REST API</b>	<b>25</b>
	Generating the API client library using Swagger.....	26
	Consuming the API client library.....	26
<b>Chapter 4</b>	<b>Supporting Query Filter</b>	<b>29</b>
	Overview of NetWorker REST API Query Filter Support.....	30
	Query Filter.....	30
	Field List Filter.....	34
	Using Query Filter and Field List Filter simultaneously.....	35
	Time Range Based Filtering.....	35
<b>Chapter 5</b>	<b>NetWorker REST API Use Cases</b>	<b>37</b>
	Creating a Client.....	38
	Viewing all clients.....	38
	Creating an Advanced File Type Device.....	40
	Labeling a Device.....	41
	Creating a Protection Group.....	41
	Creating a Protection Policy.....	42
	Creating a Workflow.....	42
	Creating Actions for a Traditional Backup.....	43
	Creating a schedule and associating it to an action in a policy.....	43
	Viewing all schedules.....	44
	Performing an On-Demand Backup.....	45
	Initiating a Backup.....	46
	Initiating a Recovery.....	49
	Viewing all backups for a client.....	49
	Viewing Client File Index.....	51
	Recovering Filesystem backup.....	52
	Working with VMware-Based Clients.....	57
	Performing vCenter Operations.....	57
	Performing vProxy Operations.....	59

Creating a protection group for VMware-based clients.....	62
Creating a protection policy for VMware-based clients.....	63
Adding a New Item to a VMware-Based Protection Group.....	65
Performing a VMware-based on-demand backup with an override....	65
Performing an on-demand backup for a specific VM.....	66
Recovering a VM to a previous state.....	67
Recovering a VM as a new VM.....	67
Browsing files and folders in a backup.....	69
Browsing the Destination VM.....	75
Creating VMware file level recovery to same virtual machine.....	77
Creating VMware file level recovery to other virtual machine.....	80
Supporting Licenses View.....	80
Supporting Plugin Installer.....	82
Viewing CloudBoost Appliances.....	83
Configuring Email Notification.....	84
<b>Viewing data domain systems</b> .....	<b>85</b>

<b>Appendix A</b>	<b>Troubleshooting</b>	<b>87</b>
	NetWorker REST API log files.....	88
	Changing the TLS level.....	88
	NetWorker REST API log file management.....	89
	NSR to REST API Command Map.....	90
	Status Codes and Messages.....	93
	Advance Client Attributes.....	94

# TABLES

1	Revision history.....	7
2	Style conventions.....	9
3	NetWorker REST API versions.....	15
4	Required privileges.....	17
5	Supported HTTP methods.....	23
6	HTTP method for client endpoints.....	23
7	Time range filtering.....	36
8	NetWorker REST API log files.....	88
9	HTTP Status Code.....	93
10	Advanced attributes in 18.2 Release.....	94
11	Advanced attributes in 19.1 Release.....	94



# Preface

As part of an effort to improve its product lines, Dell EMC periodically releases revisions of its software and hardware. Therefore, some functions that are described in this document might not be supported by all versions of the software or hardware currently in use. The product release notes provide the most up-to-date information on product features.

Contact your Dell EMC technical support professional if a product does not function correctly or does not function as described in this document.

**Note:** This document was accurate at publication time. Go to Dell EMC Online Support (<https://support.emc.com>) to ensure that you are using the latest version of this document.

## Purpose

This document describes how to configure and use NetWorker REST API.

## Audience

This guide is part of the NetWorker documentation set, and is intended for use by developers who are creating programmatic interfaces to NetWorker systems.

## Revision history

The following table presents the revision history of this document.

**Table 1** Revision history

Revision	Date	Description
02	August 14, 2019	Updated for NetWorker 19.1.1 release.
01	May 20, 2019	First release of the document for NetWorker 19.1.

## Related documentation

The NetWorker documentation set includes the following publications, available on the Support website:

- *NetWorker E-LAB Navigator*  
Provides compatibility information, including specific software and hardware configurations that NetWorker supports. To access E-LAB Navigator, go to <https://elabnavigator.emc.com/eln/elnhome>.
- *NetWorker Administration Guide*  
Describes how to configure and maintain the NetWorker software.
- *NetWorker Network Data Management Protocol (NDMP) User Guide*  
Describes how to use the NetWorker software to provide data protection for NDMP filers.
- *NetWorker Cluster Integration Guide*  
Contains information related to configuring NetWorker software on cluster servers and clients.
- *NetWorker Installation Guide*

Provides information on how to install, uninstall, and update the NetWorker software for clients, storage nodes, and servers on all supported operating systems.


- *NetWorker Updating from a Previous Release Guide*  
Describes how to update the NetWorker software from a previously installed release.
- *NetWorker Release Notes*  
Contains information on new features and changes, fixed problems, known limitations, environment and system requirements for the latest NetWorker software release.
- *NetWorker Command Reference Guide*  
Provides reference information for NetWorker commands and options.
- *NetWorker Data Domain Boost Integration Guide*  
Provides planning and configuration information on the use of Data Domain devices for data deduplication backup and storage in a NetWorker environment.
- *NetWorker Performance Optimization Planning Guide*  
Contains basic performance tuning information for NetWorker.
- *NetWorker Server Disaster Recovery and Availability Best Practices Guide*  
Describes how to design, plan for, and perform a step-by-step NetWorker disaster recovery.
- *NetWorker Snapshot Management Integration Guide*  
Describes the ability to catalog and manage snapshot copies of production data that are created by using mirror technologies on storage arrays.
- *NetWorker Snapshot Management for NAS Devices Integration Guide*  
Describes how to catalog and manage snapshot copies of production data that are created by using replication technologies on NAS devices.
- *NetWorker Security Configuration Guide*  
Provides an overview of security configuration settings available in NetWorker, secure deployment, and physical security controls needed to ensure the secure operation of the product.
- *NetWorker VMware Integration Guide*  
Provides planning and configuration information on the use of VMware in a NetWorker environment.
- *NetWorker Error Message Guide*  
Provides information on common NetWorker error messages.
- *NetWorker Licensing Guide*  
Provides information about licensing NetWorker products and features.
- *NetWorker REST API Getting Started Guide*  
Describes how to configure and use the NetWorker REST API to create programmatic interfaces to the NetWorker server.
- *NetWorker REST API Reference Guide*  
Provides the NetWorker REST API specification used to create programmatic interfaces to the NetWorker server.
- *NetWorker 19.1 with CloudBoost 19.1 Integration Guide*  
Describes the integration of NetWorker with CloudBoost.
- *NetWorker 19.1 with CloudBoost 19.1 Security Configuration Guide*  
Provides an overview of security configuration settings available in NetWorker and Cloud Boost, secure deployment, and physical security controls needed to ensure the secure operation of the product.




- **NetWorker Management Console Online Help**  
Describes the day-to-day administration tasks performed in the NetWorker Management Console and the NetWorker Administration window. To view the online help, click **Help** in the main menu.
- **NetWorker User Online Help**  
Describes how to use the NetWorker User program, which is the Windows client interface, to connect to a NetWorker server to back up, recover, archive, and retrieve files over a network.

### Special notice conventions that are used in this document

The following conventions are used for special notices:

 **NOTICE** Identifies content that warns of potential business or data loss.

 **Note:** Contains information that is incidental, but not essential, to the topic.

### Typographical conventions

The following type style conventions are used in this document:

**Table 2** Style conventions

<b>Bold</b>	Used for interface elements that a user specifically selects or clicks, for example, names of buttons, fields, tab names, and menu paths. Also used for the name of a dialog box, page, pane, screen area with title, table label, and window.
<i>Italic</i>	Used for full titles of publications that are referenced in text.
Monospace	Used for: <ul style="list-style-type: none"> <li>• System code</li> <li>• System output, such as an error message or script</li> <li>• Pathnames, file names, file name extensions, prompts, and syntax</li> <li>• Commands and options</li> </ul>
<i>Monospace italic</i>	Used for variables.
<b>Monospace bold</b>	Used for user input.
[ ]	Square brackets enclose optional values.
	Vertical line indicates alternate selections. The vertical line means or for the alternate selections.
{ }	Braces enclose content that the user must specify, such as x, y, or z.
...	Ellipses indicate non-essential information that is omitted from the example.

You can use the following resources to find more information about this product, obtain support, and provide feedback.

### Where to find product documentation

- <https://www.dell.com/support>
- <https://community.emc.com>

## Where to get support

The Support website <https://www.dell.com/support> provides access to product licensing, documentation, advisories, downloads, and how-to and troubleshooting information. The information can enable you to resolve a product issue before you contact Support.

To access a product-specific page:

1. Go to <https://www.dell.com/support>.
2. In the search box, type a product name, and then from the list that appears, select the product.

## Knowledgebase

The Knowledgebase contains applicable solutions that you can search for either by solution number (for example, KB000xxxxxx) or by keyword.

To search the Knowledgebase:

1. Go to <https://www.dell.com/support>.
2. On the **Support** tab, click **Knowledge Base**.
3. In the search box, type either the solution number or keywords. Optionally, you can limit the search to specific products by typing a product name in the search box, and then selecting the product from the list that appears.

## Live chat

To participate in a live interactive chat with a support agent:

1. Go to <https://www.dell.com/support>.
2. On the **Support** tab, click **Contact Support**.
3. On the **Contact Information** page, click the relevant support, and then proceed.

## Service requests

To obtain in-depth help from Licensing, submit a service request. To submit a service request:

1. Go to <https://www.dell.com/support>.
2. On the **Support** tab, click **Service Requests**.

**Note:** To create a service request, you must have a valid support agreement. For details about either an account or obtaining a valid support agreement, contact a sales representative. To get the details of a service request, in the `Service Request Number` field, type the service request number, and then click the right arrow.

To review an open service request:

1. Go to <https://www.dell.com/support>.
2. On the **Support** tab, click **Service Requests**.
3. On the **Service Requests** page, under **Manage Your Service Requests**, click **View All Dell Service Requests**.

## Online communities

For peer contacts, conversations, and content on product support and solutions, go to the Community Network <https://community.emc.com>. Interactively engage with customers, partners, and certified professionals online.

**How to provide feedback**

Feedback helps to improve the accuracy, organization, and overall quality of publications. You can send feedback to [DPAD.Doc.Feedback@emc.com](mailto:DPAD.Doc.Feedback@emc.com).



# CHAPTER 1

## Introduction

This chapter includes the following topics:

- [Overview](#) ..... 14
- [API Versioning](#) ..... 14
- [Authentication and Authorization](#) ..... 16
- [Limitations](#) ..... 17

## Overview

The NetWorker REST API provides a mechanism to access the functions provided by NetWorker software. The API interface helps you to build client applications and automate NetWorker operations by using any programming language that supports JSON over HTTP.

The NetWorker APIs are used to build applications that can perform client management, device management, data protection policy management, running backups and recovery.

You can use the NetWorker REST API to perform the following functions:

- Create data protection policies, with specific workflows and action to fulfill the schedule and backup level requirements for performing backups.
- Create notifications for a given event type to trigger an action like email delivery.
- Manage NetWorker clients to backup data that you want to protect.
- Manage probes and directives to perform customization for the NetWorker clients.
- Manage storage nodes and include them as the target for a backup action in the workflow of a specific data protection policy.
- Manage policies for VMware-based clients and perform VM level and granular level recovery.
- Configure devices, media pools, and volumes.
- Perform device operations such as label and mount.
- Define the protection groups to create logical grouping of clients having common data protection requirements and associate it to a specific workflow under a data protection policy.
- Perform immediate backups at client and workflow levels and monitor its progress from the job database.
- Protect file systems, virtual machines from VMware ecosystem.
- Save set recovery and granular level recovery for file system, block based backup and NDMP file system.
- Build restricted data zones in a NetWorker instance including a set of backup clients and storage nodes that access the backup media.
- Monitor NetWorker server configuration, alerts, and server messages.
- Customize the audit log configurations and manage user groups.

For more information on managing NetWorker software, see the *NetWorker Administration Guide*.

## API Versioning

NetWorker REST API enables you to manage and monitor NetWorker servers and clients. They are backward compatible. The latest version of API supports additional

functions and has stable end points. The versions can be viewed at the root URI of `https://{nw-server-hostname}:9090/nwrestapi`.

**Table 3** NetWorker REST API versions

NetWorker API version	NetWorker release	Supported features
v1	NetWorker 9.0 and minor releases	CRUD operations are supported for the following: <ul style="list-style-type: none"> <li>• Alerts</li> <li>• Audit log configuration</li> <li>• Backups</li> <li>• Clients</li> <li>• Devices</li> <li>• Directives</li> <li>• Job groups</li> <li>• Job indications</li> <li>• Jobs</li> <li>• Labels</li> <li>• Notifications</li> <li>• Pools</li> <li>• Probes</li> <li>• Protection groups</li> <li>• Protection policies</li> <li>• Server configurations</li> <li>• Server messages</li> <li>• Server statistics</li> <li>• Sessions</li> <li>• Storage Nodes</li> <li>• Volumes</li> <li>• Tenants</li> </ul>
v2	NetWorker 9.1 and minor releases	<ul style="list-style-type: none"> <li>• CRUD operations are supported for a few VMware resources.</li> <li>• Swagger API definitions for supported URIs.</li> </ul>
v3	NetWorker 9.2, and later	CRUD operations are supported for the following: <ul style="list-style-type: none"> <li>• CloudBoost appliances</li> <li>• Data Domain systems</li> <li>• Licenses</li> <li>• NAS Devices</li> </ul>

**Table 3** NetWorker REST API versions (continued)

NetWorker API version	NetWorker release	Supported features
		<ul style="list-style-type: none"> <li>• Recovers</li> <li>• Rules</li> <li>• Tenants</li> <li>• Browsing items under the Client File Index</li> <li>• File System recovery in File System, Block-Based Backup (BBB), and NDMP-based File System backup.</li> <li>• VMware File level recovery.</li> <li>• VMware browse file functionality</li> <li>• Log Viewer</li> <li>• Plugin Installer</li> <li>• vCenter</li> <li>• vProxy</li> </ul>

Code that is written using a specific version of the API continues to function when subsequent versions of the API are released. However, sometime, older versions of certain resources might be deprecated in the future. NetWorker only deprecates resources when necessary. If an older version of a resource can continue to function, it is accessible at its original API version number URI.

When an updated NetWorker REST API is available, you must regenerate the client library to access new capabilities. For more information on generating client library, see [Generating the API client library using Swagger](#)

The swagger file is updated for every major release. However, the API version number might not change for every NetWorker release.

## Authentication and Authorization

Every request to NetWorker REST API requires authentication.

NetWorker REST APIs use HTTP Basic authentication mechanism to authenticate the REST requests. You need to send the user credentials to the REST API server in every API request. It is supported by most common HTTP tools such curl, Postman, web browsers and programming frameworks.

 **Note:** NetWorker uses basic HTTP authentication with SSL only.

The users who consume the API must have access rights. You can provide privileges to users using the NMC. For more information, see the NetWorker Administration Guide. The following privileges are required to access the APIs:



**Table 4** Required privileges

Action	Role
View data	NetWorker Monitor
Back up and recover	Operate NetWorker

## Limitations

There are a few limitations in the NetWorker REST APIs.

- Configuring and managing of the NetWorker Authentication Service is not supported.
- User management is not supported.
- Token based authentication is not supported.
- Pagination for API response is limited to Client File Index (CFI) and VM browse.
- Creating, updating, and deleting multiple resources using single API call is not supported. For example, you cannot create multiple clients using single API call.
- Only exact matches for the resource attribute are displayed in the query filter. However, time range filtering for certain endpoints is supported.



# CHAPTER 2

## API Requests and Response

This chapter includes the following topics:

- [URI Format](#)..... 20
- [NetWorker API Requests](#)..... 21
- [Handling the API Response](#)..... 22

## URI Format

Each resource in NetWorker API has its own URI. The URIs follow a predictable, hierarchical structure to enhance understandability and usability.

The URI constitutes of API service name, port number, API version and resource name and an identifier. The URI is structured in the following format: `http://<hostname>:<port>/nwrestapi/<version>/global/<API resource>/<identifier>`

**Example 1** Example for API URI format

```
https://mars:9090/nwrestapi/v3/global/backups/e5f3cb7d-00000006-cc73cdb0-5c73cdb0-00355000-24133256
```

A few of the resources are organized hierarchical structure to form a parent-child relationship. The workflow will be listed under the associated protection policy.

**Example 2** Example for URI using hierarchically structure

In the following example, the workflow is listed under policy.

- `https://mars:9090/nwrestapi/v3/global/protectionpolicies`
- `https://mars:9090/nwrestapi/v3/global/protectionpolicies/policy1/workflows/workflow2`

A few of the resources can be accessed with top level end point and also from its associated resource.

In the following example, you can access the backup information from the top level end point. However, if you want to access the backup information related to specific client, then you must use client level URI.

- Top level end point: `https://mars:9090/nwrestapi/v3/global/backups/e5f3cb7d-00000006-cc73cdb0-5c73cdb0-00355000-24133256`
- Client level end point: `https://mars:9090/nwrestapi/v3/global/clients/162.0.184.21.0.0.0.239.1.35.92.10.128.244.163/backups/e5f3cb7d-00000006-cc73cdb0-5c73cdb0-00355000-24133256`

The JSON responses from the NetWorker API server may include links to navigate to a related resource.

If the URL ends with a `/op/<action>`, it indicates that an action that can be initiated by using a POST request.

```
https://mars:9090/nwrestapi/v3/global/protectionpolicies/Gold/workflows/VMware/op/backup
```

## NetWorker API Requests

Depending on the operation that you are performing, you need to select the end points accordingly.

### Requests

When you send a request using HTTP GET the collection URI `/clients` endpoint, you will receive a list of clients. Each client resource instance has a unique URI. HTTP GET request to the specific client URI will retrieve the representation for that client.

When the API calls using POST and PUT, the request body carries the resource definition to create or update the resource. The request body should be in JSON format.

The following POST request creates a new client.

```
POST https://mars:9090/nwrestapi/v3/global/clients
```

### Body

```
{
  "backupCommand": "nsrndmp_save -T dump",
  "backupType": "Filesystem",
  "hostname": "vnx1",
  "ndmp": true,
  "password": "password",
  "remoteUser": "ndmpVNX7500",
  "saveSets": [
    "/tmp/restapi"
  ],
  "protectionGroups": [
    "NDMPAFTD_PG"
  ]
}
```

Certain operations can be requested using POST with an empty body. It is used when you want to perform the API call with the default setting. For example, to start the backup operation on a workflow.

```
POST https://mars:9090/nwrestapi/v3/global/protectionpolicies/
Bronze/workflows/Filesystem/op/backup
```

### Body

```
{}
```

### Request parameters

NetWorker API requests include the following parameters:

- **Header:** The authorization header carries the user credentials in every API request. For POST and PUT calls, the API request carries the `contentType` header. Apart

from these headers, there are a few endpoint specific headers which are used to carry a specific information.

- **Path:** Path parameters are the variable part of the URI and serves as identifier of the resource instance.
- **Query:** Query parameters are used in the endpoint to perform filtering. They are displayed after a question mark (?) in the endpoint and are listed with an ampersand (&) separating them. For more information on query filters, see [Supporting Query Filter](#) on page 29

## Handling the API Response

The NetWorker API service sends the HTTP status code to indicate whether the HTTP request succeeded or failed.

### GET request on specific client

A GET request on a specific client URI, for example `/clients/161.0.120.52.0.0.0.0.210.51.200.91.10.207.81.176` returns information about that particular client resource.

```
{
  aliases: [
    "IP_Address"
  ],
  applicationInformation: [ ],
  backupType: "vProxy",
  .....
  storageNodes: [ ],
  tags: [ ]
}
```

### GET request on client URI

A GET request on `/clients` URI returns the response that includes information about all the client resources.

```
{
  clients: [
    { ... },
    { ..... },
  ],
  count: 2
}
```

### POST request

The POST API request returns a payload along with a location header carrying the URI of the newly created resource. If the resource is immediately created, then the URI for that resource is returned in the API response. In the following example, the API request returns the URI to the new client resource in the response:

```
POST https://mars:9090/nwrestapi/v3/global/clients
Body:
{
  "hostname": "mars"
}
```

For the operation that takes a longer time, a job is spawned and the job URI is returned to the API caller. For example, back up or recovery. Performing `/op/backup` on a workflow endpoint returns the job URI to track the status of an asynchronous request, so that API client can monitor the backup job status.

The following example returns 201 on success along with a URI pointing to the status resource in the location:

```
POST https://mars:9090/nwrestapi/v3/global/protectionpolicies/Bronze/workflows/Filesystem/op/backup
Body:
{ }
```

The following example returns 202 on success along with a URI pointing to the status resource in the location:

```
POST https://NetWorker:9090/nwrestapi/v3/global/devices/aftd/op/verifylabel
```

### Supported HTTP methods

**Table 5** Supported HTTP methods

HTTP method	Change in resource state	Mandatory headers	HTTP success code	Response header
GET	No	None	200	None
POST	Yes	Content-type, set to "application/json"	201/202	"location" header carries the URL of the new resource or a tracking resource.
PUT	Yes	Content-type, set to "application/json"	204	None
DELETE	Yes	None	204	None
OPTIONS	No	None	200	"allow" header shows supported HTTP methods.

**Note:** The table has generic HTTP success codes. For success codes specific to the NetWorker REST API, see the NetWorker Reference Guide.

The table below lists the operations that are supported for clients URI and a specific client URI.

**Table 6** HTTP method for client endpoints

Resource	POST	GET	PUT	DELETE
/clients	Creates a client	Retrieves the information about all the clients	Not available	Not available
/clients/ 161.0.120.52.0.0.0.0. 210.51.200.91.10.207. 81.176	Not available	Retrieves the details for specific client	Updates the specific attributes of the client	Deletes the specific client





# CHAPTER 3

## Consuming the NetWorker REST API

This chapter includes the following topics:

- [Generating the API client library using Swagger](#) ..... 26
- [Consuming the API client library](#) ..... 26

## Generating the API client library using Swagger

NetWorker REST API specification is available in JSON format and follows swagger 2.0 specification.

### Before you begin

- JRE or JDK must be installed.
- You must have Swagger Codegen utility that is installed in your system.

### About this task

Swagger Codegen is a command line utility that is used to generate REST client libraries for the NetWorker API. It can generate libraries in multiple programming languages. When an updated version of the swagger file is available and then; you should regenerate the client libraries.

### Procedure

1. Generate the library by running the command: `java -jar swagger-codegen-cli.jar -i <file_path> -l python -o <client_output_directory>`

#### Where

- `-i` : Specifies the path to the input file. It can be a URL or file path
- `-l` : Specifies the programming language for the client.
- `-o` : Specifies the output directory where the code has to be generated.

```
java -jar swagger-codegen-cli.jar generate -i
<swagger_file_path> -l python -o <client_output_directory>
```

For sample code, see the API Reference Guide.

### Results

The NetWorker REST API client consisting of the following is generated:

- Root package: Contains generic swagger utility files such as `api_client.py`, `configuration.py`, and `rest.py`
- API package: Contains autogenerated classes that are related to the APIs defined in the swagger specification.
- Model package: Contains autogenerated classes that are related to the resource models defined in the swagger specification.

## Consuming the API client library

You can generate the API client library using swagger.

### Before you begin

- You must have python 3 installed in the system.
- You must have the code ready for compilation.

### Procedure

1. Place the API file in the same location as that of the swagger client.
2. Update the configuration file.

You can update the `host`, `username` and `password` in the configuration file.

**Note:** By default, SSL is enabled. To disable SSL, you must change the value of `self.verify_ssl` to *false*.

3. Run the API client by running the command : `python3 <APIClient>.py`

The following code will list all the NetWorker clients that are associated with the NetWorker Server as MyNetworkHost.

```
#!/usr/bin/python3

import time
import swagger_client
from swagger_client.rest import ApiException
from swagger_client import Configuration
from pprint import pprint

# create an instance of the API client and Configure basic
authentication

api_instance = swagger_client.ClientsApi()
configuration = api_instance.api_client.configuration
configuration.username = 'Administrator'
configuration.password = 'Password'

# set the API root endpoint
configuration.host = 'https://<nw-host-name>:9090/nwrestapi/v3/
global'

# call the method to gather all the NetWorker clients
try:
    api_response = api_instance.get_clients()
    pprint(api_response)
except ApiException as e:
    pprint("Exception when connecting to NetWorker API: %s
\n" % e)
```



# CHAPTER 4

## Supporting Query Filter

This chapter includes the following topics:

- [Overview of NetWorker REST API Query Filter Support](#).....30
- [Query Filter](#).....30
- [Field List Filter](#)..... 34
- [Using Query Filter and Field List Filter simultaneously](#)..... 35
- [Time Range Based Filtering](#).....35

## Overview of NetWorker REST API Query Filter Support

This section describes the types of query filters supported by NetWorker REST API.

NetWorker REST API URI supports two types of filters:

- Query filter (q)
- Field list filter(fl)

## Query Filter

This section demonstrates how to use query filter (q) by using NetWorker REST API. Query filter searches for specified key:value in resource list. The [value] should be URL encoded and double quoted.

To query based on single attribute, use the following URI structure:

```
/endpoint?q=AttributeName:Value
```

### Example 3 API request example for query filter with single attribute

```
GET /nwrestapi/v3/global/clients?q=hostname:"pfrxbnw40"
200 OK
{
  "clients": [
    {
      "aliases": [
        "pfrxbnw40"
      ],
      "applicationInformation": [],
      "backupCommand": "savepsm.sh",
      "backupType": "Filesystem",
      "blockBasedBackup": false,
      "checkpointEnabled": false,
      "clientId":
"3bb6e7ab-00000004-5cb5fa5e-5cb5fa5d-00015000-e01b3256",
      "hostname": "pfrxbnw40",
      "indexBackupContent": false,
      "links": [
        {
          "href": "https://mars:9090/nwrestapi/v3/global/
clients/47.0.51.124.0.0.0.24.252.181.92.10.118.244.68",
          "rel": "item"
        }
      ],
      "nasDevice": false,
      "ndmp": false,
      "ndmpMultiStreamsEnabled": false,
      "ndmpVendorInformation": [],
      "parallelSaveStreamsPerSaveSet": false,
      "parallelism": 12,
      "protectionGroups": [
        "NMC server"
      ],
      "remoteAccessUsers": [],
      "resourceId": {
        "id":
```

**Example 3** API request example for query filter with single attribute (continued)

```

"47.0.51.124.0.0.0.0.24.252.181.92.10.118.244.68",
  "sequence": 3
},
"saveSets": [
  "/nsr/nmc/nmcdb_stage"
],
"scheduledBackup": true,
"storageNodes": [
  "nsrserverhost"
],
"tags": []
}
]
"count": 1
}

```

To specify multiple attributes, use the “and” delimiter with the following URI structure:

```
/endpoint?q=AttributeName1:Value1 and AttributeName2:Value2
```

**Example 4** API request example for query filter with multiple attributes

```

GET /nwrestapi/v3/global/clients?q=hostname:"sql" and
backupType:"SQL Server"

200 OK
{
  "clients": [
    {
      "aliases": [
        "sql"
      ],
      "applicationInformation": [
        "NSR_PS_DEBUG_LEVEL=0",
        "NSR_SKIP_SIMPLE_DB=FALSE",
        "NSR_CONSISTENCY_CHECKS=FALSE"
      ],
      "backupCommand": "nsrsqlsv",
      "backupType": "SQL Server",
      "blockBasedBackup": false,
      "checkpointEnabled": false,
      "clientId":
"7814dd58-0000004-5cca7e73-5cca7e72-000b5000-180a3256",
      "hostname": "sql",
      "indexBackupContent": false,
      "links": [
        {
          "href": "https://mars:9090/nwrestapi/v3/global/
clients/46.0.12.48.0.0.0.0.18.7.200.92.10.118.245.14",
          "rel": "item"
        }
      ],
      "nasDevice": false,
      "ndmp": false,
      "ndmpMultiStreamsEnabled": false,
      "ndmpVendorInformation": [],
      "parallelSaveStreamsPerSaveSet": false,

```

**Example 4** API request example for query filter with multiple attributes (continued)

```

        "parallelism": 4,
        "password": "*****",
        "protectionGroups": [
            "SQL_PG"
        ],
        "remoteAccessUsers": [],
        "remoteUser": "administrator",
        "resourceId": {
            "id":
"46.0.12.48.0.0.0.0.18.7.200.92.10.118.245.14",
            "sequence": 5
        },
        "saveSets": [
            "MSSQL:"
        ],
        "scheduledBackup": true,
        "storageNodes": [],
        "tags": []
    }
],
"count": 1
}

```

To specify multiple values for same attribute, use the “,” delimiter with the following URI structure:

```
/endpoint?q=AttributeName:Value1,Value2
```

**Example 5** API request example for query filter with multiple values for same attribute

```

GET /nwrestapi/v3/global/clients?q=hostname:"rhel72-base","sql"
200 OK
{
  "clients": [
    {
      "aliases": [
        "rhel72-base"
      ],
      "applicationInformation": [],
      "backupType": "vProxy",
      "blockBasedBackup": false,
      "checkpointEnabled": false,
      "clientId":
"345394a6-00000004-5cb48658-5cb48c3f-00025000-180a3256",
      "hostname": "rhel72-base",
      "indexBackupContent": false,
      "links": [
        {
          "href": "https://mars:9090/nwrestapi/v3/global/
clients/162.0.75.14.0.0.0.0.244.133.180.92.10.118.245.14",
          "rel": "item"
        }
      ],
      "nasDevice": false,
      "ndmp": false,
      "ndmpMultiStreamsEnabled": false,
      "ndmpVendorInformation": [],

```



**Example 5** API request example for query filter with multiple values for same attribute (continued)

```

        "parallelSaveStreamsPerSaveSet": false,
        "parallelism": 4,
        "protectionGroups": [],
        "remoteAccessUsers": [],
        "resourceId": {
            "id":
"162.0.75.14.0.0.0.244.133.180.92.10.118.245.14",
            "sequence": 4
        },
        "saveSets": [
            "All"
        ],
        "scheduledBackup": true,
        "storageNodes": [],
        "tags": []
    },
    {
        "aliases": [
            "sql"
        ],
        "applicationInformation": [
            "NSR_PS_DEBUG_LEVEL=0",
            "NSR_SKIP_SIMPLE_DB=FALSE",
            "NSR_CONSISTENCY_CHECKS=FALSE"
        ],
        "backupCommand": "nsrsqlsv",
        "backupType": "SQL Server",
        "blockBasedBackup": false,
        "checkpointEnabled": false,
        "clientId":
"7814dd58-0000004-5cca7e73-5cca7e72-000b5000-180a3256",
        "hostname": "sql",
        "indexBackupContent": false,
        "links": [
            {
                "href": "https://MARS:9090/nwrestapi/v3/global/
clients/46.0.12.48.0.0.0.18.7.200.92.10.118.245.14",
                "rel": "item"
            }
        ],
        "nasDevice": false,
        "ndmp": false,
        "ndmpMultiStreamsEnabled": false,
        "ndmpVendorInformation": [],
        "parallelSaveStreamsPerSaveSet": false,
        "parallelism": 4,
        "password": "*****",
        "protectionGroups": [
            "SQL_PG"
        ],
        "remoteAccessUsers": [],
        "remoteUser": "administrator",
        "resourceId": {
            "id":
"46.0.12.48.0.0.0.18.7.200.92.10.118.245.14",
            "sequence": 5
        },
        "saveSets": [
            "MSSQL:"
        ],
        "scheduledBackup": true,
        "storageNodes": [],
        "tags": []
    }
}

```

**Example 5** API request example for query filter with multiple values for same attribute (continued)

```
    ],
    "count": 2
  }
```

## Field List Filter

You can use field list filter (fl) parameter to specify the fields or attributes that are to be listed in the response. This parameter helps to limit the information that is listed in the response. If the filter is not specified, then all the fields are listed.

### Using field list filter

To filter based on single attributes, use the following URI format:

```
/endpoint?fl=AttributeName
```

To filter based on multiple attributes, use the “,” delimiter with the following URI format:

```
/endpoint?fl=AttributeName1,AttributeName2
```

**Example 6** API request example for field list filter

```
GET /nwrestapi/v3/global/clients?fl=hostname,backupType
200 OK
{
  "clients": [
    {
      "backupType": "vProxy",
      "hostname": "MARS"
    },
    {
      "backupType": "Filesystem",
      "hostname": "pfrxbnw40"
    },
    {
      "backupType": "Filesystem",
      "hostname": "pfrxbnw40"
    }
  ],
  "count": 3
}
```

## Using Query Filter and Field List Filter simultaneously

This section demonstrates how to use query and field list filter simultaneously using NetWorker REST API.

To use query filter (q) and field list filter (fl) simultaneously, use the following structure for the request URI:

```
/endpoint?q=[AttributeName]:[Value]&fl=field1,field2
```

### Example 7 API request using q and fl simultaneously

```
GET /nwrestapi/v3/global/backups?q=clientHostname:ISILON and
level:Incr&fl=name,shortId,level

200 OK
{
  "backups": [
    {
      "level": "Incr",
      "name": "/ifs/demo/test",
      "shortId": "4016477993"
    },
    {
      "level": "Incr",
      "name": "/ifs/demo/test",
      "shortId": "3966232744"
    }
  ],
  "count": 2
}
```

## Time Range Based Filtering

You can filter the data by using date as a time range. However, this is supported for /backups, /jobgroups and /jobs endpoints only.

### Example 8 API request example for query filter with time range

```
GET /nwrestapi/v3/global/backups?
fl=saveTime,clientHostName,type&q=saveTime:['2019-04-30T21:00:05'
TO '2019-04-30T22:00:05']

200 OK
{
  "backups": [
    {
      "clientHostname": "mars",
      "saveTime": "2019-05-09T21:00:15-04:00",
      "type": "File"
    },
    {
      "clientHostname": "10.118.245.95",
      "saveTime": "2019-05-09T21:00:14-04:00",

```

**Example 8** API request example for query filter with time range (continued)

```

        "type": "File"
      }
    ],
    "count": 2
  }

```

Time range filtering can be done with or without the time zone in the saveTime specified in the query parameter.

**Table 7** Time range filtering

With time zone	/nwrestapi/v3/global/backups?q=saveTime: ['2019-04-26T10:00:00-04:00' TO '2019-04-26T10:15:00-04:00']
Without time zone	/nwrestapi/v3/global/backups?q=saveTime: ['2019-04-26T10:00:00' TO '2019-04-26T10:15:00']

# CHAPTER 5

## NetWorker REST API Use Cases

This chapter contains the following topics:

• <a href="#">Creating a Client</a> .....	38
• <a href="#">Viewing all clients</a> .....	38
• <a href="#">Creating an Advanced File Type Device</a> .....	40
• <a href="#">Labeling a Device</a> .....	41
• <a href="#">Creating a Protection Group</a> .....	41
• <a href="#">Creating a Protection Policy</a> .....	42
• <a href="#">Creating a Workflow</a> .....	42
• <a href="#">Creating Actions for a Traditional Backup</a> .....	43
• <a href="#">Creating a schedule and associating it to an action in a policy</a> .....	43
• <a href="#">Viewing all schedules</a> .....	44
• <a href="#">Performing an On-Demand Backup</a> .....	45
• <a href="#">Initiating a Backup</a> .....	46
• <a href="#">Initiating a Recovery</a> .....	49
• <a href="#">Viewing all backups for a client</a> .....	49
• <a href="#">Viewing Client File Index</a> .....	51
• <a href="#">Recovering Filesystem backup</a> .....	52
• <a href="#">Working with VMware-Based Clients</a> .....	57
• <a href="#">Supporting Licenses View</a> .....	80
• <a href="#">Supporting Plugin Installer</a> .....	82
• <a href="#">Viewing CloudBoost Appliances</a> .....	83
• <a href="#">Configuring Email Notification</a> .....	84
• <a href="#">Viewing data domain systems</a> .....	85

## Creating a Client

This section demonstrates how to create a client by using the NetWorker REST API.

**Note:** When creating module clients, use the Management Console Client Backup Configuration Wizard rather than the REST API.

### API request

A client is created in the following example by using the NetWorker REST API:

```
POST /nwrestapi/v3/global/clients
{
  "hostname" : "saturn",
  "saveSets" : ["/etc/hosts"]
}
```

### API response

The following response is received from the API:

```
201 Created
```

## Viewing all clients

This section demonstrates how to view all clients configured on a NetWorker server using the NetWorker REST API.

### API request

The following API uses the NetWorker REST API to list all configured clients:

```
GET /nwrestapi/v3/global/clients
```

### API response

The following response is received from the API:

```
200 OK
{
  "clients": [
    {
      "aliases": [
        "IP_address"
      ],
      "applicationInformation": [],
      "backupType": "vProxy",
      "blockBasedBackup": false,
      "checkpointEnabled": false,
      "clientId": "ae014526-00000004-5bea643b-5bea6fb5-00035000-
b2123256",
      "hostname": "mars",
      "indexBackupContent": false,
      "links": [
        {
          "href": "https://mars:9090/nwrestapi/v3/global/clients/
50.0.252.9.0.0.0.0.68.99.234.91.mars",
          "rel": "item"
        }
      ]
    }
  ]
}
```

```

    }
  ],
  "nasDevice": false,
  "ndmp": false,
  "ndmpMultiStreamsEnabled": false,
  "ndmpVendorInformation": [],
  "parallelSaveStreamsPerSaveSet": false,
  "parallelism": 4,
  "protectionGroups": [],
  "remoteAccessUsers": [
    "user=cn=username,cn=Users,dc=Windows2012r2,host=windows2012r2"
  ],
  "resourceId": {
    "id": "50.0.252.9.0.0.0.0.68.99.234.91.mars",
    "sequence": 1
  },
  "saveSets": [
    "All"
  ],
  "scheduledBackup": true,
  "storageNodes": [],
  "tags": []
}
],
"count": 1
}

```

### API request for advanced attributes

The following API uses the NetWorker REST API to list all configured clients with advanced attributes:

```
GET /nwrestapi/v3/global/clients?type=advanced
```

### API response

The following response is received from the API:

```

200 OK
{
  "clients": [
    {
      "aliases": [
        "mars"
      ],
      "applicationInformation": [],
      "backupType": "vProxy",
      "blockBasedBackup": false,
      "checkpointEnabled": false,
      "clientDirectEnabled": true,
      "clientId": "ae014526-00000004-5bea643b-5bea6fb5-00035000-
b2123256",
      "dataDomainBackup": false,
      "dataDomainInterface": "IP",
      "hostname": "mars",
      "indexBackupContent": false,
      "jobControl": [],
      "links": [
        {
          "href": "https://mars:9090/nwrestapi/v3/global/clients/
50.0.252.9.0.0.0.0.68.99.234.91.10.118.252.15",
          "rel": "item"
        }
      ]
    }
  ]
}

```

```

    ],
    "nasDevice": false,
    "ndmp": false,
    "ndmpMultiStreamsEnabled": false,
    "ndmpVendorInformation": [],
    "networkerVersion": "99.0.99.9680.Build.9680",
    "parallelSaveStreamsPerSaveSet": false,
    "parallelism": 4,
    "pool": "Default",
    "protectionGroups": [],
    "remoteAccessUsers": [
      "user=cn=kiran,cn=Users,dc=Windows2012r2,host=windows2012r2"
    ],
    "resourceId": {
      "id": "50.0.252.9.0.0.0.0.68.99.234.91.10.118.252.15",
      "sequence": 1
    },
    "saveOperations": "I18N:mode=nativepath",
    "saveSets": [
      "All"
    ],
    "scheduledBackup": true,
    "serverNetworkInterface": "IP_Address",
    "storageNodes": [],
    "tags": []
  }
],
"count": 1
}

```

## Creating an Advanced File Type Device

This section demonstrates how to create an Advanced File Type Device (AFTD) using the NetWorker REST API.

### API request

An Advanced File Type device is created in the following example by using the NetWorker REST API:

```

POST /nwrestapi/v3/global/devices
{
  "name": "/space/storage",
  "deviceAccessInfo": "/space/storage",
  "mediaType": "adv_file"
}

```

### API response

```

201 Created

```



## Labeling a Device

This section demonstrates how to label a device using the NetWorker REST API.

### API request

In the following example the device is labeled by using the NetWorker REST API:

```
POST /nwrestapi/v3/global/devices/{deviceName}/op/label
{
  "labelWithoutMount": "false",
  "pool": "Default"
}
```

### API response

```
202 Accepted

Location →https://mars:9090/nwrestapi/v3/global/devices/
{deviceName}/op/status
Date →Sat, 11 May 2019 04:43:50 GMT
Content-Type →application/json
Transfer-Encoding →chunked
Server →NSR SERVICES for Authentication
```

### API response body

```
{
  "text": "The request is accepted. To retrieve its status, issue
an HTTP GET request to URL specified in the 'Location' response
header."
}
```

## Creating a Protection Group

This section demonstrates how to create a protection group by using the NetWorker REST API.

### API request

A protection group is created in the following example by using the NetWorker REST API:

```
POST /nwrestapi/v3/global/protectiongroups
{
  "workItemType" : "Client",
  "name" : "EngineeringWorkstations",
  "workItems" :
  ["48.0.254.77.0.0.0.0.149.148.51.87.128.222.109.201"]
}
```

**Note:** When using the REST API, the resourceId of the client must be used for the workItems element. This can be retrieved from the resourceId element in the client resource, as shown in the API response section of [Viewing all clients](#) on page 38.

### API response

```
201 Created
```

## Creating a Protection Policy

This section demonstrates how to create a protection policy by using the NetWorker REST API.

### API request

In the following example a protection policy is created by using the NetWorker REST API:

```
POST /nwrestapi/v3/global/protectionpolicies
{
  "name": "Engineering"
}
```

### API response

```
201 Created
```

## Creating a Workflow

This section demonstrates how to create a workflow by using the NetWorker REST API.

### API request

In the following example a workflow is created by using the NetWorker REST API:

```
POST /nwrestapi/v3/global/protectionpolicies/Engineering/workflows
{
  "name" : "ClientProtection",
  "protectionGroups" : ["EngineeringWorkstations"],
  "startTime" : "21:00"
}
```

### API response

```
201 Created
```

## Creating Actions for a Traditional Backup

This section demonstrates how to create actions for a traditional backup by using the NetWorker REST API.

### API request

In the following example actions are created by using the NetWorker REST API:

```
PUT /nwrestapi/v3/global/protectionpolicies/Engineering/workflows/
ClientProtection
{
  "actions": [
    {
      "name": "Backup",
      "actionSpecificData": {
        "backup": {
          "backupSpecificData": {
            "traditional": {}
          }
        }
      }
    }
  ]
}
```

### API response

```
204 No Content
```

## Creating a schedule and associating it to an action in a policy

NetWorker provides you with pre-configured schedules that you can associate to an action in a policy. You can also create custom schedules and manage them by using the NetWorker REST API.

### API request

In the following example a policy with the attribute "actionReferSchedule" in action is created by using the NetWorker REST API:

```
POST https://mars:9090/nwrestapi/v3/global/protectionpolicies
{
  "name": "Policy_Name",
  "workflows": [
    {
      "actions": [
        {
          "actionSpecificData": {
            "backup": {
              "backupSpecificData": {
                "traditional": {}
              }
            }
          }
        },
        {
          "name": "Action_Name",
```

```

        "actionReferSchedule": "Custom"
      }
    ],
    "name": "Workflow_Name"
  }
]
}

```

### API response

```
201 Created
```

## Viewing all schedules

This section demonstrates how to view all schedules configured on a NetWorker server using the NetWorker REST API.

### API request

The following command uses the NetWorker REST API to list all pre-configured and custom schedules:

```
GET nwrestapi/v3/global/schedules
```

### API response

```

200 OK
{
  "count": 11,
  "schedules": [
    {
      "activityType": "Backup",
      "levels": [
        "full",
        "incr",
        "incr",
        "incr",
        "incr",
        "incr",
        "incr"
      ],
      "links": [
        {
          "href": "https://mars:9090/nwrestapi/v3/global/
schedules/Custom",
          "rel": "item"
        }
      ],
      "name": "Custom",
      "overrides": [],
      "period": "Week",
      "resourceId": {
        "id":
"52.0.244.14.0.0.0.0.126.2.26.93.10.118.139.43",
        "sequence": 1
      }
    },
    .
    .
  ]
}

```

```
.
.
] }
```

## Performing an On-Demand Backup

This section demonstrates how to perform an on-demand backup of a client by using the NetWorker REST API.

### API request

In the following example on-demand backup is performed by using the NetWorker REST API:

```
POST /nwrestapi/v3/global/protectionpolicies/Engineering/workflows/
ClientProtection/op/backup
{
  "clients": ["saturn:/etc/hosts"]
}
```

### API response

The following response is received from the API:

```
201 Created
```

### Retrieving information about the backup job

The location header in the response from the above API contains a URL to the job that was created. Use the following GET request to that URL to retrieve information about the job:

### API request

```
GET /nwrestapi/v3/global/jobs/{jobID}
```

### API response

```
200 OK
{
  "adhocJob": false,
  "command": "/usr/sbin/nsrworkflow -p Engineering -w
ClientProtection -L -c saturn:/etc/hosts",
  "dependentJobIds": [
    0
  ],
  "id": 64081,
  "itemIdLong": 64081,
  "links": [
    {
      "href": "https://mars:9090/nwrestapi/v3/global/jobs/64081/op/
cancel",
      "title": "Cancel job"
    }
  ],
  "logFile": "/nsr/logs/policy/Engineering/
workflow_ClientProtection_064081.raw",
```

```

    "name": "Engineering",
    "ndmp": false,
    "parentJobId": 0,
    "previousJobId": 0,
    "rootParentJobId": 0,
    "runOnHost": "mars",
    "siblingJobIds": [],
    "startTime": "2019-05-17T12:34:07-04:00",
    "state": "Active",
    "stopped": false,
    "tenant": "",
    "type": "workflow job"
  }

```

## Initiating a Backup

You can initiate a backup using NetWorker REST APIs.

### Before you begin

- You must have a backup device.
- You must have a list of mandatory attributes.

### About this task

The following procedure provides you a guideline on initiating a backup using NetWorker REST API. You can modify the sequence as per your requirements.

### Procedure

1. Create a protection group.

**POST /nwrestapi/v3/global/protectiongroups**  
**Payload**

```

{
  "name": "<Name_of_the_protection_group>",
  "workItemType": "Client"
}

```

where,

- name: Name of the protection group

A protection group is created with a “201 created” http status code.

2. Create a client.

**POST /nwrestapi/v3/global/clients**  
**Payload**

```

{
  "aliases": ["networkerserver"],
  "hostname": "<host_name>",
  "saveSets": ["<File Path>"],
  "backupType": "Filesystem",
  "storageNodes": [],
  "protectionGroups":
  ["<Name_of_the_protection_group>"]
}

```

where:

- **hostname:** Host name of the client from which backup is taken.
- **saveSets :** List of savesets that are to be backed-up.
- **protectionGroups:** Name of the protection group that was created in the previous step.

A client is created with a “201 created” http status code.

### 3. Create a Label.

**POST /nwrestapi/v3/global/labels**

**Payload**

```
{
  "fields": ["test", "001-999"],
  "name": "<name of the_label>",
  "next": "test.001",
  "separator": "."
}
```

**where:**

- **fields:** List of label components.
- **name :** Name of the label template.

A label is created with a “201 created” http status code.

### 4. Create a Pool.

**POST /nwrestapi/v3/global/pools**

**Payload**

```
{
  "labelTemplate": "<name of the_label>",
  "poolType": "Backup",
  "name": "<name of the pool>"
}
```

**where:**

- **labelTemplate:** Name of the label template created in the previous step.
- **name :** Name of the pool that is to be created.

A pool is created with “201 created” http status code.

### 5. Create a Device.

**POST /nwrestapi/v3/global/devices**

**Payload**

```
{
  "name": "<name of the device>",
  "mediaType": "adv_file",
  "deviceAccessInfo": "<file_path>"
}
```

**where:**

- **name :** Name of the device that is to be created.
- **deviceAccessInfo:** Specifies the access path for an advance file or a Data Domain device.
- **mediaType:** The media type describes the actual storage media from the predefined list

## 6. Label the device.

**POST /nwrestapi/v3/global/devices/api%20device/op/label**  
**Payload**

```
{
  "labelWithoutMount": "false",
  "pool": "api_pool "
}
```

**where:**

- **pool** : Name of the pool created in earlier step 4.

## 7. Create a protection policy.

**POST /nwrestapi/v3/global/protectionpolicies**  
**Payload**

```
{
  "name": "<name_of_the_Policy>",
  "summaryNotification": {
    "command": "smtpmail -s \"subject\" -h \"mailserver\"
    emailid@domain_name.com,
    "executeOn": "Completion"
  },
  "workflows": [
    {
      "actions": [
        {
          "actionSpecificData": {
            "backup": {
              "destinationStorageNodes":
                ["<name_of_the_storage_node>"],
              "backupSpecificData": {
                "traditional":
                  {"destinationPool": "<name_of_the_pool>"}
              }
            }
          },
          "name": "<<Name_of_the_Action>"
        }
      ],
      "name": "<Name_of_the_workflow>",
      "protectionGroups": ["<Name_of_the_protection_group>"]
    }
  ]
}
```

**where:**

- **pool** : Name of the pool created in earlier step 4.

## 8. Initiate a Backup

**POST /nwrestapi/v3/global/protectionpolicies/**  
**<name\_of\_the\_Policy>/workflows/<Name\_of\_the\_workflow>/op/backup**  
**Payload**

```
{
}
```

## Results

On successful completion of a backup an email notification is sent.



## Initiating a Recovery

### Procedure

1. Get the save time of the specific back up

```
GET https://mars::9090/nwrestapi/v3/global/backups/?fl=name,saveTime
```

2. Get the client resource id.

```
GET /nwrestapi/v3/global/clients/?
fl=hostname,resourceId,saveSets
```

3. Recover from the backup.

```
POST /nwrestapi/v3/global/recovers
Payload
```

```
{
  "recoveryType": "Filesystem",
  "recoveryDestination": "C:\\destination",
  "itemsToRecover": ["C:\\script"],
  "actionForDuplicateItems": "Rename",
  "destinationClientResID":
  "200.0.156.9.0.0.0.0.127.83.106.92.mars",
  "timeStampBasedGranularRecover": {
    "sourceClientResID":
    "200.0.156.9.0.0.0.0.127.83.106.92.mars",
    "timeStamp": "2019-02-20T06:00:41-05:00"
  }
}
```

To perform a file system saveset and granular recovery, see [Recovering Filesystem backup](#) on page 52

## Viewing all backups for a client

This section demonstrates how to view all backups for a client by using the NetWorker REST API.

### API request

A request to view all the backups for a client is created in the following example by using the NetWorker REST API:

```
GET /nwrestapi/v3/global/clients/
82.0.31.22.0.0.0.0.33.12.41.87.128.222.109.201/backups
```

**Note:** When using the REST API, the resourceid of the client must be used for the `workItems` element. This can be retrieved from the `resourceid` element in the client resource, as shown in the API response section of [Viewing all clients](#) on page 38.

### API response

```
200 OK
{
  "backups": [
```

```

{
  "attributes": [
    {
      "key": "*backup start time",
      "values": [
        "1463499698"
      ]
    },
    {
      "key": "*ss clone retention",
      "values": [
        "          1463499698:          1463499698:          -2"
      ]
    },
    {
      "key": "save set features",
      "values": [
        "CLIENT_SAVE TIME"
      ]
    }
  ],
  "browseTime": "2038-01-18T22:14:06-05:00",
  "clientHostname": "mars",
  "clientId":
  "b7d73be0-00000004-57290c2f-57290c2e-00010c00-80c64a29",
  "completionTime": "2016-05-17T11:41:38-04:00",
  "creationTime": "2016-05-17T11:41:38-04:00",
  "fileCount": 3,
  "id": "67f3852c-00000006-f03b3bb2-573b3bb2-00120c00-80c64a29",
  "instances": [
    {
      "clone": false,
      "id": "1463499698",
      "status": "Browsable",
      "volumeIds": [
        "4264957582"
      ]
    }
  ],
  "level": "Manual",
  "links": [
    {
      "href": "https://mars:9090/nwrestapi/v2/global/clients/82.0.31.22.0.0.0.0.33.12.41.87.128.222.109.201/backups/67f3852c-00000006-f03b3bb2-573b3bb2-00120c00-80c64a29",
      "rel": "item"
    }
  ],
  "name": "\/etc\/hosts",
  "retentionTime": "2038-01-18T22:14:06-05:00",
  "saveTime": "2016-05-17T11:41:38-04:00",
  "shortId": "4030413746",
  "size": {
    "unit": "Byte",
    "value": 6736
  },
  "type": "File"
}
...
]
}

```

## Viewing Client File Index

This section describes the process to support indexing with query and pagination, by using NetWorker REST API.

### Viewing Client Indexes and Pagination

The json response can be large, as it depends on client's index database. It provides paginated output. You can set the desired number of entries in `pagesize` query filter. The NetWorker REST API uses custom header `X-NW-CONTINUATION-TOKEN` to paginate the output. The NetWorker server returns `X-NW-CONTINUATION-TOKEN` header in response if client file indexes are more than `pagesize` (default maximum limit is 100). The requester has to provide `X-NW-CONTINUATION-TOKEN` with value received in last response header. The NetWorker REST server reads `X-NW-CONTINUATION-TOKEN` header value and returns next set of index records.


#### Note:

- The user gets a maximum value of 100 for query parameter `pagesize`
- The token has a Time-To-Live (TTL) of 120 sec.
- The token has `Read-Once` property, which means once server gets a valid token and serves the request, the token becomes invalid.
- The query parameters set at the beginning of the request cannot be modified for subsequent calls.

### API request

Following API request helps you to view client indexes:

```
GET /nwrestapi/v3/global/clients/{clientResourceID}/indexes
```

 **Note:** You must use the *ResourceID* of the client as the `clientResourceID`. For example,

```
GET nwrestapi/v3/global/clients
```

```
200 OK
.
.
.
.
"resourceId": {
  "id": "50.0.252.9.0.0.0.0.68.99.234.91.10.118.252.15",
  .
  .
  .
```

#### Example 9 API requestAPI response

```
GET /nwrestapi/v3/global/clients/
159.0.52.3.0.0.0.0.226.242.11.90.10.31.227.39/indexes
```

**Example 9** API requestAPI response (continued)**Response header**

```
Content-Type ?application/json
Date ?Thu, 21 Dec 2017 08:59:41 GMT
Server ?Apache-Coyote/1.1
Transfer-Encoding ?chunked
X-NW-CONTINUATION-TOKEN ?
MjAyZDUwMmI1ZDU2MzMwZDZlYzZiMDcwZDg5MjEwMjU6MTcxMTY4ODA=
```

**Response body**

```
[
  {
    "fileName": "C:\\Program Files\\EMC NetWorker\\Management\\GST\\
    \\cst\\acm_lb.clb",
    "fileSizeInBytes": 4528,
    "namespace": "backup",
    "offset": 0,
    "saveTime": "2018-01-17T14:00:37+05:30"
  },
  {
    "fileName": "C:\\Program Files\\EMC NetWorker\\Management\\GST\\
    \\cst\\acm_lb.clb.bak",
    "fileSizeInBytes": 3920,
    "namespace": "backup",
    "offset": 4528,
    "saveTime": "2018-01-17T14:00:37+05:30"
  },
  ...
]
```

## Recovering Filesystem backup

This section describes the process to perform a file system saveset and granular recovery by using the NetWorker REST API.

### Recovering a Saveset

This section describes the process to perform a saveset based recovery by using the NetWorker REST API.

#### By using NMC

To recover a saveset using NMC:

1. Navigate to the **Recover** tab.
2. Right click and select **New Recover**.
3. In the Recover wizard, select **Traditional NetWorker Client Recovery**.
4. Select a source client, destination client, and available recovery types for the File System, Block Based, and NDMP based File System.
5. Select the **Saveset Recover** tab and the saveset to be recovered.
6. Select **File Path for Recovery** and **Duplicate File Options**.

## By using REST API

To perform file system saveset recovery, perform the steps that are mentioned in the following sections in the defined sequence:

1. Refer [Prerequisites for recovery request](#) on page 53
2. Refer [Creating recover resource by using REST API](#) on page 53
3. Refer [Viewing details of recover resource](#) on page 54
4. Refer [Monitoring the status of recover job](#) on page 54

### Prerequisites for recovery request

Use GET `/nwrestapi/v3/global/backups` to get backupInstance information.

The backup ID can be retrieved from the value of attribute `id` by using the API GET `/backups` or GET `/backups/{backupId}/instances` for a specific backup instance.

### Creating recover resource by using REST API

You can use the NetWorker REST API for a saveset recovery as described in the following example:

Filesystem (FS):

```
POST /nwrestapi/v3/global/recovers
{
  "recoveryType": "Filesystem",
  "recoveryDestination": "e:\\nsrRecover",
  "backupInstance":
  {
    "backupID": "45b6b47c-00000006-faac2017-59ac2017-00075000-
bde5b856",
    "instanceID": "1504883884"
  },
  "actionForDuplicateItems": "Rename"
}
```

Block based backup:

```
POST /nwrestapi/v3/global/recovers
{
  "recoveryType": "BBB",
  "targetVolume": "E:\\",
  "backupInstance":
  {
    "backupID": "59ef04c5-00000006-e3aed96a-59aed96a-001f5000-bde5b856"
    "instanceID": "1504889037"
  },
  "actionForDuplicateItems": "Overwrite"
}
```

NDMP based file system backup:

```
POST /nwrestapi/v3/global/recovers
{
  "recoveryType": "NDMP",
  "recoveryDestination": "/ifs/data/ajay/MohanRec",
  "backupInstance":
  {
    "backupID": "855e9b15-00000006-f9b7fe7b-59b7fe7b-00085000-bde5b856"
  }
}
```

```

    "instanceID":"1506346646"
  }
}

```

When making an API request, start with a backup ID for the client saveset. The following response is received from the REST API:

```
201 Created
```

### Viewing details of recover resource

The API response contains a resource URI in the location header. Use the resource URI to get the recovery resource by using the `GET recovers/{recoverId}` API, which contains a link that you can use to monitor the job. The following is the sample output of `GET recovers/{recoverId}`

```

200 OK
{
  "destinationClientResID":
  "c9797b4d-00000004-59e73721-59e8490b-000c5000-a6138c56",
  "links": [
    {
      "href": "https://mars:9090/nwrestapi/v3/global/jobs/800142",
      "rel": "item"
    }
  ],
  "name": "NDMP_8817_189862517965772",
  "recoveryDestination": "/ifs/data/ajay/MohanRec",
  "recoveryStartTime": "2017-11-23T01:50:51-08:00",
  "recoveryType": "NDMP",
  "sourceClient": "ISILON",
  "backupInstance":
  {
    "backupID": "855e9b15-00000006-f9b7fe7b-59b7fe7b-00085000-
bde5b856"
    "instanceID": "1506346646"
  },
  "resourceId": {
    "id": "49.0.96.18.0.0.0.0.47.81.22.90.10.207.86.34",
    "sequence": 1
  }
}

```

**Note:** In highly loaded systems, job Id might not be created immediately. In such scenarios, consumer of the API has to periodically check the recover resource URI until the job is created. If job is not created for some reason, consumer of the API might need to time out with appropriate message.

### Monitoring the status of recover job

To monitor the recovery job, use the following API:

```
GET jobs/{jobId}
```

### Performing granular recovery of selected files or folders in a backup

This section demonstrates how to recover specific files and folders by using the NetWorker REST API.

### By using NMC

To perform granular recovery using NMC:

1. Navigate to the **Recover** tab.
2. Right click and select **New Recover**.
3. In the Recover wizard, select **Traditional NetWorker Client Recovery**.
4. Select a source client, destination client, and available recovery types for the File System, Block Based, and NDMP based File System.
5. Select the **Browse** tab and select the file and folder to be recovered.
6. Select **File Path** for **Recovery** and **Duplicate File Options**.

### By using REST API

To perform file system granular recovery, perform the steps that are mentioned in the following sections in the defined sequence:

1. Refer [Prerequisites for recovery request](#) on page 55
2. Refer [Creating recover resource by using REST API](#) on page 55
3. Refer [Viewing details of recover resource](#) on page 56
4. Refer [Monitoring the status of recover job](#) on page 57

### Prerequisites for recovery request

For the `recoveryDestination` attribute, user should be aware of the file system.

- Retrieve `sourceClientResID` from the `resourceId` attribute of the API GET `/nwrestapi/v3/global/clients/{client resource id}`.
- Retrieve `timeStamp` from the `saveTime` attribute of the API GET `/nwrestapi/v3/global/backups`.

**Note:** User should know the file system to specify the value for `recoveryDestination` attribute.

### Creating recover resource by using REST API

You can use the NetWorker REST API for a granular recovery as described in the following example:

**Note:** The supported formats for timestamp are epoch time in seconds and human readable format in `<<YYYY-MM-DDTHH:MM:SS+GMT>>`. For example:  
`"2017-10-05T23:50:14+05:30"`

Filesystem (FS):

```
POST /nwrestapi/v3/global/recovers
{
  "recoveryType": "Filesystem",
  "recoveryDestination": "e:\\nsrRecover",
  "timeStampBasedGranularRecover":
  {
    "sourceClientResID": "159.0.80.10.0.0.0.0.176.177.169.89.32.2.190.239.190.239.0.0.149.137.80.106",
    "timeStamp": "1506593756"
  },
  "itemsToRecover": ["C:\\vbm_tools\\vbm_version.txt", "C:\\vbm_tools\\trusted.reg"],
  "actionForDuplicateItems": "Rename"
}
```

**Block based backup:**

```
POST /nwrestapi/v3/global/recovers
{
  "recoveryType": "BBB",
  "recoveryDestination": "E:\\BBB_recover",
  "timeStampBasedGranularRecover":
  {
    "sourceClientResID": "160.0.184.15.0.0.0.0.80.152.83.90.10.63.30.91",
    "timeStamp": "2018-01-09T09:38:33+05:30"
  },
  "itemsToRecover": ["F:\\BBB\\a.pdf", "F:\\BBB\\b.pdf"]
}
```

**NDMP based file system backup:**

```
POST /nwrestapi/v3/global/recovers
{
  "recoveryType": "NDMP",
  "recoveryDestination": "/ifs/data/ajay/MohanRec",
  "timeStampBasedGranularRecover":
  {
    "sourceClientResId": "43.0.168.31.0.0.0.0.252.65.185.89.32.2.190.239.190.239.0.0.149.137.80.106",
    "timeStamp": "1506346646"
  },
  "itemsToRecover": ["/ifs/data/ajay/images/firmware.xml", "/ifs/data/ajay/images/devices.xml"]
}
```

The following response is received from the REST API:

```
201 Created
```

**Viewing details of recover resource**

The API response contains a resource URI in the location header. Use the resource URI to get the recovery resource by using the `GET recovers/{recoverId}` API, which contains a link that you can use to monitor the job. Following is the sample output of `GET recovers/{recoverId}`

```
{
  "destinationClientResID":
  "c9797b4d-00000004-59e73721-59e8490b-000c5000-a6138c56",
  "itemsToRecover": [
    "/ifs/data/ajay/images/firmware.xml",
    "/ifs/data/ajay/images/devices.xml"
  ],
  "links": [
    {
      "href": "https://10.207.86.34:9090/nwrestapi/v3/global/jobs/800142",
      "rel": "item"
    }
  ],
  "name": "NDMP_8817_189862517965772",
  "recoveryDestination": "/ifs/data/ajay/MohanRec",
  "recoveryStartTime": "2017-11-23T01:50:51-08:00",
  "recoveryType": "NDMP",
}
```



```

    "resourceId": {
      "id": "49.0.96.18.0.0.0.0.47.81.22.90.10.207.86.34",
      "sequence": 1
    },
    "sourceClient": "ISILON",
    "timeStampBasedGranularRecover": {
      "sourceClientResID":
        "c9797b4d-00000004-59e73721-59e8490b-000c5000-a6138c56",
      "timeStamp": "2017-11-23T00:38:11-08:00"
    }
  }
}

```

**Note:** In highly loaded systems, job Id might not be created immediately. In such scenarios, consumer of the API has to periodically check the recover resource URI until the job is created. If job is not created for some reason, consumer of the API might need to time out with appropriate message.

### Monitoring the status of recover job

To monitor the recovery job, use the following API:

```
GET jobs/{jobId}
```

## Working with VMware-Based Clients

Because of limited command line support for VMware client operations, the examples in this section use NetWorker Management Console (NMC) examples, rather than command line examples.

### Performing vCenter Operations

This section describes the process to create, view, modify and delete vCenter, by using the NetWorker REST API.

#### Creating vCenter resource

You can create vCenter resource using `POST` call of following URI `/vmware/vcenters`

#### API request

```

POST /nwrestapi/v3/global/vmware/vcenters
{
  "hostname": "mars",
  "userName": "administrator@vsphere.local",
  "userPassword": "password!"
}

```

#### Viewing the list of vCenter resources

You can view the list of all the vCenter resources using NetWorker REST API.

#### API request

```
GET /nwrestapi/v3/global/vmware/vcenters
```

## API response

```

200 OK
{
  "count": 2,
  "vCenters":
  [
    {
      "cloudDeployment": false,
      "hostname": "mars",
      "links":
      [
        {
          "href": "https://mars:9090/nwrestapi/v3/global/vmware/
vcenters/mars",
          "rel": "item"
        }
      ],
      "resourceId":
      {
        "id": "50.0.68.13.0.0.0.0.18.13.116.90.10.63.30.91",
        "sequence": 1
      },
      "userName": "administrator@vsphere.local",
      "userPassword": "*****"
    },
    {
      "cloudDeployment": false,
      "hostname": "mars",
      "links":
      [
        {
          "href": "https://mars:9090/nwrestapi/v3/global/
vmware/vcenters/mars",
          "rel": "item"
        }
      ],
      "resourceId":
      {
        "id": "164.0.184.15.0.0.0.0.80.152.83.90.10.63.30.91",
        "sequence": 5833
      },
      "userName": "administrator@vsphere.local",
      "userPassword": "*****"
    }
  ]
}

```

## Viewing vCenter resource

You can view vCenter resource using GET call of following URI /vmware/vcenters/{vcenter-id}

## API request

```
GET /nwrestapi/v3/global/vmware/vcenters/jupiter
```

## API response

```

200 OK
{
  "cloudDeployment": false,
  "hostname": "jupiter",

```

```

"links":
  [
    {
      "href": "https://mars:9090/nwrestapi/v3/global/vmware/vcenters/
jupiter/op/refresh",
      "title": "Refresh vCenter view"
    },
    {
      "href": "https://mars:9090/nwrestapi/v3/global/vmware/vcenters/
jupiter/protectedvms",
      "title": "List of protected virtual machines"
    },
    {
      "href": "https://mars:9090/nwrestapi/v3/global/vmware/vcenters/
jupiter/vms",
      "title": "List of virtual machines"
    },
    {
      "href": "https://mars:9090/nwrestapi/v3/global/vmware/vcenters/
jupiter/plugins",
      "title": "Intall vCenter plugins"
    }
  ],
"resourceId":
  {
    "id": "44.0.84.7.0.0.0.179.242.94.90.mars",
    "sequence": 1
  },
"userName": "administrator@vsphere.local",
"userPassword": "*****"
}

```

### Modifying vCenter resource

You can modify vCenter resource using PUT call of following URI `/vmware/vcenters/{vcenter-id}`

#### API request

```

PUT /nwrestapi/v3/global/vmware/vcenters/jupiter
{
  "cloudDeployment": true
}

```

### Deleting vCenter resource

You can delete vCenter resource using DELETE call of following URI `/vmware/vcenters/{vcenter-id}`

#### API request

```

DELETE /nwrestapi/v3/global/vmware/vcenters/jupiter

```

## Performing vProxy Operations

This section describes the process to create, register, view, modify and delete vProxy, by using the NetWorker REST API.

### Creating vProxy resource

You can create vProxy resource using POST call of following URI `/vmware/vproxies`

### API request

```
POST /nwrestapi/v3/global/vmware/vproxies
{
  "enabled": true,
  "hostname": "Jupiter",
  "maxHotaddSessions": 13,
  "maxNbdSessions": 8,
  "userName": "admin",
  "vCenterHostname": "IP_Address",
  "password": "mars",
  "vProxyPort": 9090
}
```

### Registering vProxy resource

You can register vProxy resource using POST call of following URI `/vmware/vproxies/op/register`

### API request

```
POST /nwrestapi/v3/global/vmware/vproxies/op/register
{
  "enabled": true,
  "hostname": "Jupiter",
  "maxHotaddSessions": 13,
  "maxNbdSessions": 8,
  "userName": "admin",
  "vCenterHostname": "IP_Address",
  "password": "password",
  "vProxyPort": 9090
}
```

### Viewing the list of vProxy resources

You can view the list of all the vProxy resources using NetWorker REST API.

### API request

```
GET /nwrestapi/v3/global/vmware/vproxies
```

### API response

```
200 OK
{
  "count": 2,
  "vProxies":
  [
    {
      "datastores": [],
      "enabled": true,
      "encryptNbdSessions": false,
      "hostname": "jupiter",
      "links":
      [
        {
          "href": " /nwrestapi/v3/global/vmware/vproxies/jupiter",
          "rel": "item"
        }
      ],
      "maxHotaddSessions": 13,
```

```

"maxNbdSessions": 8,
"password": "*****",
"resourceId":
{
  "id": "53.0.100.15.0.0.0.143.124.105.90.mars",
  "sequence": 7
},
"userName": "admin",
"vCenterHostname": "IP_Address",
"vProxyPort": 9090,
"version": "Release: '3.0.0-1_SNAPSHOT20180126185157', Build
number: '1', Build date: '2018-01-26T18:51:57Z'"
},
{
  "datastores": [],
  "enabled": true,
  "encryptNbdSessions": false,
  "hostname": "Jupiter",
  "links":
[
  {
    "href": " /nwrestapi/v3/global/vmware/vproxies/Jupiter",
    "rel": "item"
  }
],
"maxHotaddSessions": 13,
"maxNbdSessions": 8,
"password": "*****",
"resourceId":
{
  "id": "51.0.68.13.0.0.0.18.13.116.90.mars",
  "sequence": 1
},
"userName": "admin",
"vCenterHostname": "IP_Address",
"vProxyPort": 9090
}
]
}

```

### Viewing vProxy resource

You can view vProxy resource using GET call of following URI /global/vmware/vproxies/{vproxy-id}

### API request

```
GET /nwrestapi/v3/global/vmware/vproxies/Jupiter
```

### API response

```

{
  "datastores": [],
  "enabled": false,
  "encryptNbdSessions": false,
  "hostname": "Jupiter",
  "maxHotaddSessions": 13,
  "maxNbdSessions": 8,
  "password": "*****",
  "resourceId":
  {
    "id": "46.0.84.7.0.0.0.179.242.94.90.mars",
    "sequence": 2
  }
}

```

```

    },
    "userName": "admin",
    "vCenterHostname": "IP_Address",
    "vProxyPort": 9090
  }

```

### Modifying vProxy resource

You can modify vProxy resource using PUT call of following URI `/vmware/vproxies/{vproxy-id}`

#### API request

```

PUT /nwrestapi/v3/global/vmware/vproxies/Jupiter
{
  "maxHotaddSessions": 11,
  "maxNbdSessions": 7
}

```

### Deleting vProxy resource

You can delete vProxy resource using DELETE call of following URI `/vmware/vproxies/{vproxy-id}`

#### API request

```

DELETE /nwrestapi/v3/global/vmware/vproxies/Jupiter

```

#### API response

```

204 No Content

```

## Creating a protection group for VMware-based clients

This section demonstrates how to create a protection group for VMware-based clients by using the NetWorker REST API.

### NMC example

To create a protection group in NMC:

1. Navigate to the **Protection** tab.
2. Select **Groups**.
3. Right-click and select **New** and enter a name for the group.
4. Set the group type to **VMware** and sub-type to **All**.
5. Select the vCenter to be used.
6. Browse the vCenter to select individual VMs and containers such as resource pools.

#### API request

The same protection group is created in the following example by using the NetWorker REST API:

```

POST /nwrestapi/v3/global/protectiongroups
{

```

```

"name": "ApiGroup",
"vmwareWorkItemSelection": {
  "containerMorefs": [
    "resgroup-7675"
  ],
  "vCenterHostname": "hostname.company.com",
  "vmUuids": [
    "50085479-6e0f-2383-7cb8-15883df49fde"
  ],
  "vmdks": []
},
"workItemQueries": [],
"workItemSource": "Static",
"workItemSubType": "All",
"workItemType": "VMware"
}

```

This protection group contains two selection items:

- A VM with a specific UUID—50085479-6e0f-2383-7cb8-15883df49fde. Resources such as `/vmware/vcenters/{vcenter-hostname}/vms`, which returns a list of VMs on the specified vCenter, can be used to retrieve the UUID for a VM.

**Note:** You can use POST at the `/vmware/vcenters/{vcenter-hostname}/op/refresh` endpoint gather the latest information from the specific vCenter.

- A container with a specified Managed Object Reference (MoRef)—`resgroup-7675`  
When a container is included, then all VMs included in this container will be backed up, even if they are added after the protection group was created. The NetWorker REST API does not expose all the containers present in the vCenter. These should be obtained through another mechanism such as directly from the vCenter.

### API response

The following response is received from the API:

```
201 Created
```

## Creating a protection policy for VMware-based clients

This section demonstrates how to create a protection policy for VMware-based clients by using the NetWorker REST API.

### NMC example

To create a protection group in NMC:

1. Navigate to the **Protection** tab.
2. Select **Policies**.
3. Right-click and select **New** and enter a name for the policy.
4. Select **Create a new workflow** and enter a name for the workflow.
5. Associate the new workflow with a protection group. In this example, we will associate it with the ApiGroup created in [Creating a protection group for VMware-based clients](#) on page 62.
6. Create an action, enter a name for the action, and set the action type to **Backup** and the backup subtype to **VMware (vProxy)**.

## API request

The same protection policy is created in the following example by using the NetWorker REST API:

```
POST /nwrestapi/v2/global/protectionpolicies
{
  "name": "ApiPolicy",
  "workflows": [
    {
      "actions": [
        {
          "actionSpecificData": {
            "backup": {
              "backupSpecificData": {
                "vmwareVProxy": {
                  "quiesceApp": false,
                  "destinationPool": "Data Domain Default"
                }
              }
            },
            "destinationStorageNodes": [
              "nsrserverhost"
            ]
          }
        },
        {
          "name": "ApiAction",
          "scheduleActivities": [
            "incr",
            "incr",
            "incr",
            "incr",
            "incr",
            "incr",
            "incr"
          ],
          "schedulePeriod": "Week"
        }
      ],
      "name": "ApiWorkflow",
      "protectionGroups": [
        "ApiGroup"
      ]
    }
  ]
}
```

Some properties, such as `scheduleActivities`, are optional and will default to the same values as those shown in NMC.

After the policy resource is created, you can perform a GET for the resource URL returned in the location header to see the full resource with all populated default values.

## API response

The following response is received from the API:

```
201 Created
```



## Adding a New Item to a VMware-Based Protection Group

This section demonstrates how to add a new item, such as a new VM, to a VMware-based protection group by using the NetWorker REST API.

### NMC example

To add a new item to a VMware-based protection group in NMC:

1. Navigate to the **Protection** tab.
2. Select **Groups**.
3. Select the group.
4. Right-click and select **Properties** and add the item.

### API request

To add an item to a VMware-based protection group by using the NetWorker REST API:

```
POST /nwrestapi/v3/global/protectiongroups/ApiGroup/op/
updatevmwareworkitems
{
  "addWorkItems": {
    "vCenterHostname": "hostname.company.com",
    "vmUuids": [
      "5008edba-edfe-127c-1d00-68830b14b0fc"
    ]
  }
}
```

Resources such as `/vmware/vcenters/{vcenter-hostname}/vms`, which returns a list of VMs on the specified vCenter, can be used to retrieve the UUID for a VM.

### API response

```
204 No Content
```

## Performing a VMware-based on-demand backup with an override

This section demonstrates how to perform a VMware-based on-demand backup with an override by using the NetWorker REST API.

### NMC example

With NMC, you can run a backup only for a specific VM, container, or subset of VMs in a container:

1. Navigate to the **Protection** tab.
2. Double-click **VMware View**.
3. Select a VM or container associated with a protection group.
4. Right-click and select **Start Individual Client...**
5. The **Start Workflow** dialog will appear, where you can verify or adjust the selection.

### API request

An on-demand backup is created in the following example by using the NetWorker REST API:

```
POST /nwrestapi/v3/global/protectionpolicies/ApiPolicy/workflows/
ApiWorkflow/op/backup
{
  "vmwareWorkItemSelection" : {
    "vCenterHostname" : "hostname.company.com",
    "vmUuids" : ["50085479-6e0f-2383-7cb8-15883df49fde"]
  }
}
```

### API response

The following response is received from the API:

```
201 Created
```

The API response will contain a Job URI in the location header. This URI can be used to monitor the job by using GET `jobs/{jobId}`, and to cancel it if necessary by using POST `/jobs/{jobId}/op/cancel`.

## Performing an on-demand backup for a specific VM

This section demonstrates how to perform an on-demand backup for a specific VM by using the NetWorker REST API.

### NMC example

To perform an on-demand backup for a specific VM in NMC:

1. Navigate to the **Protection** tab.
2. Double-click **VMware View**.
3. Select a VM or container associated with a protection group.
4. Right-click and select **Start Individual Client...**
5. The **Start Workflow** dialog will appear, where you can select a policy or workflow if the VM is protected by more than one.

### API request

An on-demand backup is created in the following example by using the NetWorker REST API:

```
POST /nwrestapi/v3/global/vmware/vcenters/hostname.company.com/vms/
5008edba-edfe-127c-1d00-68830b14b0fc/op/backup
{
  "policy" : "ApiPolicy",
  "workflow" : "ApiWorkflow"
}
```

### API response

The following response is received from the API:

```
201 Created
```

The API response will contain a Job URI in the location header. This URI can be used to monitor the job by using `GET jobs/{jobId}`, and to cancel it if necessary by using `POST /jobs/{jobId}/op/cancel`.

## Recovering a VM to a previous state

This section demonstrates how to recover a VM to a previous state (revert) by using the NetWorker REST API.

### NMC example

To recover a VM to a previous state (revert) in NMC:

1. Navigate to the **Recover** tab.
2. Right-click and select **New recover** and enter a name for the group.
3. In the **Recover** wizard, select **Virtual Machine Recovery**.
4. Select a vCenter.
5. Select the VM and backup to be recovered.
6. For **Virtual Machine Recovery Type**, select **Revert a Virtual Machine**.  
This example assumes that all disks will be recovered.

### API request

A recovery is performed in the following example by using the NetWorker REST API:

```
POST /nwrestapi/v3/global/vmware/vcenters/hostname.company.com/
protectedvms/50085479-6e0f-2383-7cb8-15883df49fde/backups/
9524aeb0-00000006-f824e4c1-5824e4c1-00090c00-80c64a29/instances/
1478812865/op/recover
{
  "recoverMode" : "Revert",
  "powerOn": true,
  "reconnectNic": false
}
```

When making this API request, start with a backup (saveset) of the VM. This can be retrieved using `GET /backups/{backupId}/instances`.

### API response

The following response is received from the API:

```
201 Created
```

The API response will contain a Job URI in the location header. This URI can be used to monitor the job by using `GET jobs/{jobId}`, and to cancel it if necessary by using `POST /jobs/{jobId}/op/cancel`.

## Recovering a VM as a new VM

This section demonstrates how to recover a VM as a new VM by using the NetWorker REST API.

### NMC example

To recover a VM as a new VM in NMC:

1. Navigate to the **Recover** tab.

2. Right-click and select **New recover** and enter a name for the group.
3. In the **Recover** wizard, select **Virtual Machine Recovery**.
4. Select a vCenter.
5. Select the VM and backup to be recovered.
6. For **Virtual Machine Recovery Type**, select **Virtual Machine Recovery**.
7. Select the location where the VM should be recovered.
8. Select the disks to be recovered.

### API request

A recovery is performed in the following example by using the NetWorker REST API:

```
POST /nwrestapi/v3/global/vmware/vcenters/hostname.company.com/
protectedvms/50085479-6e0f-2383-7cb8-15883df49fde/backups/
9524aeb0-00000006-f824e4c1-5824e4c1-00090c00-80c64a29/op/recover
{
  "computeResourceMoref" : "domain-s7",
  "datacenterMoref": "datacenter-2",
  "datastoreMoref" : "datastore-11",
  "disks" : [
    {
      "datastoreMoref": "datastore-11",
      "key": "2000",
      "name": "Hard disk 1"
    },
    {
      "datastoreMoref": "datastore-11",
      "key": "2001",
      "name": "Hard disk 2"
    }
  ],
  "hostMoref": "host-9",
  "jobName": "jobName",
  "powerOn": false,
  "reconnectNic" : false,
  "recoverMode" : "New",
  "vCenterHostname" : "hostname.company.com",
  "vmName" : "newVmName"
}
```

This example requires information about the VMware recovery location and disk configuration, which can only be obtained from the vCenter outside of the API. If only recovery to the original VM location is required, most of the parameters in this example, including the disks, can be retrieved from the VM state recorded during the backup. This information is available in the `vmInformation` property of the backup object.

### API response

The following response is received from the API:

```
201 Created
```

The API response will contain a Job URI in the location header. This URI can be used to monitor the job by using `GET jobs/{jobId}`, and to cancel it if necessary by using `POST /jobs/{jobId}/op/cancel`.

## Browsing files and folders in a backup

This section describes the process of browsing files and folders in a backup.

To browse files and folders in a VMware backup, you need to perform the following steps in the defined sequence:

1. Refer [Identifying VMware backup for browsing files and folders](#) on page 69
2. Refer [Mounting the backup](#) on page 70
3. Refer [Viewing the Job ID of a successful mounted VM](#) on page 71
4. Refer [Creating browsing session](#) on page 72
5. Refer [Viewing the browse session details](#) on page 73
6. Refer [Retrieving the content of current working directory of a backup](#) on page 74
7. Refer [Changing the current working directory of a browsing session](#) on page 75

### Identifying VMware backup for browsing files and folders

You can identify the VMware backup, by using API request, for browsing files and folders.

#### API request

A GET request on either of these URI results responds with the list of available VMware backups and details of the backups. One of the backups can be identified for the files and folders recovery.

```
GET /vmware/vcenters/{vcenter-hostname}/protectedvms/{vm-uuid}/backups
```

or

```
GET /vmware/vcenters/{vcenter-hostname}/protectedvms/{vm-uuid}/backups/{backup-id}
```

**Example 10** URI example to identify VMware backup for browsing files and folders

```
https://nwrestapi/v3/global/vmware/vcenters/10.7.6.28/protectedvms/500c2ae2-eb16-373e-ddc1-abdc61dd2625/backups?fl=vmInformation
```

**Example 11** API response

```
200 OK
{
  "backups": [
    {
      "vmInformation": {
        "datastoreMoref": "datastore-107",
        "hostMoref": "host-94",
```

**Example 11** API response (continued)

```

    "disks": [
      {
        "datastoreMoref": "datastore-107",
        "datastoreName": "remote",
        "key": "2000",
        "name": "Hard disk 1",
        "sizeInKb": 16777216,
        "thinProvisioned": false
      }
    ],
    "morefPath": "/datacenter-87/domain-s92/vm-122",
    "vCenterHostname": "mars",
    "vmMoref": "vm-122",
    "vmName": "REHL7"
  }
],
"count": 1
}

```

**Note:** You need to note the values of the attributes `vmMoref` and `vmName` from the identified backups. These values are used in the VM mount request.

**Mounting the backup**

You can mount the identified backup to browse files or folders.

**API request**

You can mount the VMware backup by using the following NetWorker REST API request:

```

POST /nwrestapi/v3/global/vmware/vcenters/{vcenter-hostname}/
protectedvms/{vm-uuid}/backups/{backup-id}/op/vmmount

```

**Example 12** URI example to mount the backup

```

https://nwrestapi/v3/global/vmware/vcenters/10.7.6.28/protectedvms/
/500c8452-2be3-19dd-ce36-854a1d4d4cde/backups/3356bee8-00000006-
f00e8b12-5a0e8b12-00115000-a6138c56/op/vmmount

```

**Example 13** API request example

```

{
  "installFlrAgent": true,
  "targetVCenterHostname": "hostname",
  "targetVmAdminUserId": "administrator",
  "targetVmAdminUserPassword": "Password123!",
  "targetVmMoref": "vm-108",
  "targetVmName": "Win-host2",
  "targetVmUserId": "administrator",
  "targetVmUserPassword": "Password123!",
}

```

**Example 13** API request example (continued)

```

        "uninstallFlrAgent": false,
        "vProxy": "IP
Address"
"
}

```

**Note:** You should note URI from the response header. The URI is in the following format:

```
/jobs/{jobId}
```

**Viewing the Job ID of a successful mounted VM**

Response of a successful mount operation contains Job reference link in the response header. Use a **GET** call on this Job reference URI to view the mounted session ID and the status of the mounting operation. To perform a VMware file level browse operation, value of attribute `vProxyMountState` should be mounted.

**Note:** You need to note the mount Job ID, which is needed for populating the attribute `mountJobId` in VMware FLR recovery.

**API request**

To view the status of mounted session, use the following URI:

```
GET /jobs/{jobId}
```

**Example 14** URI example to view the job ID of a successful mounted VM

```
https://nwrestapi/v3/global/jobs/704689
```

**Example 15** API response

```

{
  "adhocJob": true,
  "command": "nsrvproxy_flr -c '10.7.6.28' -m 'vm-42' -v
'Win2k12_host' -u 'Administrator'
          -s '3356bee8-00000006-
f00e8b12-5a0e8b12-00115000-a6138c56' -i -U 'administrator'",
  "dependentJobIds": [
    0
  ],
  "id": 704689,
  "itemIdLong": 704689,
  "links": [
    {
      "href": "https://mars:9090/nwrestapi/v3/global/jobs/

```

**Example 15** API response (continued)

```

704689/op/cancel",
  "title": "Cancel job"
}
],
"logFile": "C:\\Program Files\\EMC NetWorker\\nsr\\logs\\adhoc\\
\\adhoc-job-000002",
"message": "End of mount request processing; mountpoint is 'C:\\
\\Program Files (x86)\\EMC\\
Vproxy FLR Agent\\flr\\mountpoints\\FLR116562623'
(no error)",
"name": "nsrvproxy_flr",
"ndmp": false,
"parentJobId": 0,
"previousJobId": 0,
"rootParentJobId": 0,
"runOnHost": "win2k12-ser",
"saveSetId": "3356bee8-00000006-f00e8b12-5a0e8b12-00115000-
a6138c56",
"siblingJobIds": [],
"startTime": "2017-11-20T02:46:33-08:00",
"state": "SessionActive",
"stopped": false,
"tenant": "",
"type": "vproxyflr mount job",
"vProxyErrorMessages": [],
"vProxyFlrAgentInstallRequired": false,
"vProxyHostname": "10.207.86.35",
"vProxyMountSessionId": "4d8e3404-bb5e-44cc-806b-b69e1b0d4544",
"vProxyMountState": "Mounted",
"vProxyMountTargetVmInvalidAdminUser": false,
"vProxyMountTargetVmInvalidUser": false,
"vProxyMountTargetVmMoref": "vm-42",
"vProxyMountTargetVmName": "Win2k12_host",
"vProxyMountTargetVmUserId": "Administrator",
"vProxyMountVCenterHostname": "10.7.6.28"
}

```

**Creating browsing session**

You can create a VMware browsing session by using API request.

**By using API request**

To browse specific files or folders, use the following URI:

```

POST /vmware/vcenters/{vcenter-hostname}/protectedvms/{vm-uuid}/
backups/{backup-id}/op/
vmmount/{vproxy-mount-session-id}/vmbrowse

```

**Example 16** URI example to create browsing session

```

https://mars:9090/nwrestapi/v3/global/vmware/vcenters/10.7.6.28/
protectedvms/
500c8452-2be3-19dd-ce36-854a1d4d4cde/backups/3356bee8-00000006-
f00e8b12-5a0e8b12-00115000-a6138c56/
op/vmmount/4d8e3404-bb5e-44cc-806b-b69e1b0d4544/vmbrowse

```



**Example 17** API request example

```
{
  "currentWorkingDirectory": "01-E$"
}
```

**Viewing the browse session details**

You can view the details of browse session by using API request.

**API request**

To view the status of browse operation, use the following NetWorker REST API request:

```
GET /vmware/vcenters/{vcenter-hostname}/protectedvms/{vm-uuid}/
backups/{backup-id}/op/
vmmount/{vproxy-mount-session-id}/vmbrowse/{vproxy-browse-session-
id}
```

Response of GET has the link to view the contents of the current working directory of the backup

**Example 18** URI example to view the browse session details

```
https://mars:9090/nwrestapi/v3/global/vmware/vcenters/10.7.6.28/
protectedvms
/500c8452-2be3-19dd-ce36-854a1d4d4cde/backups/3356bee8-00000006-
f00e8b12-5a0e8b12-00115000-a6138c56
/op/vmmount/4d8e3404-bb5e-44cc-806b-b69e1b0d4544/vmbrowse/
cbe0c2be-5979-4cba-8beb-d7a0af6f61d8
```

**Example 19** API response

```
200 OK
{
  "config": {
    "sessionId": "cbe0c2be-5979-4cba-8beb-d7a0af6f61d8",
    "browseDestination": "false",
    "cacheRetentionSeconds": "60",
    "currentWorkingDirectory": "01-E$",
    "idleTimeout": "300"
  },
  "links": [
    {
      "href": "https://mars:9090/nwrestapi/v3/global/vmware/
vcenters/10.7.6.28
/protectedvms/500c8452-2be3-19dd-ce36-854a1d4d4cde/backups/
3356bee8-00000006-f00e8b12
-5a0e8b12-00115000-a6138c56/op/vmmount/4d8e3404-bb5e-44cc-806b-
b69e1b0d4544/vmbrowse/cbe0c2be-5979
-4cba-8beb-d7a0af6f61d8/contents",
      "title": "VM browse contents"
    }
  ]
}
```

**Example 19** API response (continued)

```

    }
  ],
  "status": {
    "state": "Success",
    "description": "Fetched directory contents of 'C:\\Program
Files (x86)\\EMC\\Vproxy FLR Agent
\\flr\\mountpoints\\FLR116562623\\01-E$'"
  }
}

```

**Retrieving the content of current working directory of a backup**

You can retrieve the content of the current working directory of browsing session by using API request.

**API request**

To retrieve the content of the current working directory of a backup, use the following NetWorker REST API request:

```

GET /vmware/vcenters/{vcenter-hostname}/protectedvms/{vm-uuid}/
backups/{backup-id}/op/
vmmount/{vproxy-mount-session-id}/vmbrowse/{vproxy-browse-session-
id}/contents

```

**Example 20** URI example to retrieve the content of current working directory of a backup

```

https://mars:9090/nwrestapi/v3/global/vmware/vcenters/10.7.6.28/
protectedvms/
500c8452-2be3-19dd-ce36-854a1d4d4cde/backups/3356bee8-00000006-
f00e8b12-5a0e8b12-00115000-a6138c56
/op/vmmount/4d8e3404-bb5e-44cc-806b-b69e1b0d4544/vmbrowse/
cbe0c2be-5979-4cba-8beb-d7a0af6f61d8/contents

```

**Example 21** API response

```

200 OK
{
  "error": {
    "text": "",
    "code": "0"
  },
  "noMoreData": "true",
  "totalObjects": "26",
  "freeSpace": "1915236352",
  "fileList": [
    {
      "fileName": "etc",
      "fileType": "directory",
      "fileSize": "0",
      "fileModificationTime": "2018-03-20T05:25:46Z"
    },
  ],
}

```

**Example 21** API response (continued)

```
{
  "fileName": "recover",
  "fileType": "directory",
  "fileSize": "0",
  "fileModificationTime": "2018-02-07T10:33:27Z"
}
```

**Changing the current working directory of a browsing session**

You can change the current working directory of browsing session by using API request.

**API request**

To change the current working directory of a browsing session, use the following NetWorker REST API request:

```
PUT /vmware/vcenters/{vcenter-hostname}/protectedvms/{vm-uuid}/
backups/{backup-id}/op/
vmmount/{vproxy-mount-session-id}/vmbrowse/{vproxy-browse-session-
id}
```

**Example 22** URI example to change the current working directory of a browsing session

```
https://mars:9090/nwrestapi/v3/global/vmware/vcenters/10.7.6.28/
protectedvms
/500c8452-2be3-19dd-ce36-854a1d4d4cde/backups/3356bee8-00000006-
f00e8b12-5a0e8b12-00115000-a6138c56
/op/vmmount/4d8e3404-bb5e-44cc-806b-b69e1b0d4544/vmbrowse/
cbe0c2be-5979-4cba-8beb-d7a0af6f61d8
```

**Example 23** API request example

```
{
  "currentWorkingDirectory": "01-E$\\demo"
}
```

## Browsing the Destination VM

This section demonstrates how to browse a destination VM by using NetWorker REST API.

**Creating a new resource**

You can create a new resource by using NetWorker REST API.

## API request

```
POST https://mars:9090/nwrestapi/v3/global/vmware/vcenters/
10.7.8.28/protectedvms/
500c8452-2be3-19dd-ce36-854a1d4d4cde/backups/3356bee8-00000006-
f00e8b12-5a0e8b12-00115000-
a6138c56/op/vmmount/4d8e3404-bb5e-44cc-806b-b69e1b0d4544/vmbrowse
```

### Example 24 API request payload

```
{
  "browseDestination": true,
  "osType": "Windows"
}
```

## Updating an existing resource

You can update an existing resource by using NetWorker REST API.

### API request

```
PUT https://mars:9090/nwrestapi/v3/global/vmware/vcenters/
10.7.8.28/protectedvms/
500c8452-2be3-19dd-ce36-854a1d4d4cde/backups/3356bee8-00000006-
f00e8b12-5a0e8b12-00115000-
a6138c56/op/vmmount/4d8e3404-bb5e-44cc-806b-b69e1b0d4544/vmbrowse/
cbe0c2be-5979-4cba-
8beb-d7a0af6f61d8
```

### Example 25 API request payload

```
{
  "currentWorkingDirectory:" "C:\\\"
  "browseDestination": true,
  "osType": "Windows"
}
```

## Browsing the destination VM

You can browse the destination VM by using NetWorker REST API.

### By using API request

```
GET https://mars:9090/nwrestapi/v3/global/vmware/vcenters/10.7.8.28/
protectedvms/
500c8452-2be3-19dd-ce36-854a1d4d4cde/backups/3356bee8-00000006-
f00e8b12-5a0e8b12-00115000-a6138c56/op/
vmmount/4d8e3404-bb5e-44cc-806b-b69e1b0d4544/vmbrowse/
cbe0c2be-5979-4cba-8beb-d7a0af6f61d8/contents
```

### Example 26 API response payload

**Example 26** API response payload (continued)

```

200 OK
{
  "error": {
    "text": "",
    "code": "0"
  },
  "noMoreData": "true",
  "totalObjects": "15",
  "freeSpace": "10831069184",
  "fileList": [
    {
      "fileName": "demo",
      "fileType": "directory",
      "fileSize": "0",
      "fileModificationTime": "2018-03-06T06:32:27Z"
    }
  ]
}

```

## Creating VMware file level recovery to same virtual machine

This section describes the process to perform VMware File level recovery to same virtual machine (VM) that contains the backup, by using the NetWorker REST API.

To create VMware file level recovery to same VM, you need to perform the following in the defined sequence:

1. Refer [Browsing files and folders in a backup](#) on page 69
2. Refer [Recovering specific files or folders from the identified backup to the same VM](#) on page 77
3. Refer [Viewing the status of recovery operation](#) on page 78

### Recovering specific files or folders from the identified backup to the same VM

You can initiate the VMware file level recovery by using API request.

#### API request

Response of the VMware file level recover `POST` operation contains a job reference link.

```
POST /vmware/vcenters/{vcenter-hostname}/protectedvms/{vm-uuid}/
backups/{backup-id}/op/recover
```

A `GET` request on this URI result responds with the status of the VMware recovery operation.

```
/jobs/{jobId}
```

**Note:** In highly loaded system, there is a remote possibility of job creation failure, if the job is not created within 10 seconds. In such cases, the `POST` response

header returns the URI reference link of its own resource. Consumer has to monitor the URI reference link for recover Job ID is created in the response body

```
GET /recovers/{RecoverResourceID}
```

If URI reference link for recover Job ID is not created for some reason, consumer of the API might need to time out with appropriate message.

**Example 27** URI example to recover specific files or folders from the identified backup to same VM

```
https://mars:9090/nwrestapi/v3/global/vmware/vcenters/10.7.6.28/protectedvms/500c8452-2be3-19dd-ce36-854a1d4d4cde/backups/3356bee8-00000006-f00e8b12-5a0e8b12-00115000-a6138c56/op/recover
```

**Example 28** API request

```
{
  "recoverMode": "FLR",
  "vCenterHostname": "IP Address",
  "mountJobId": "704689",
  "vmwareVmFlrOptions": {
    "terminateMountSession": "false",
    "overwrite": "true",
    "itemsToRecover": ["01-E$\demo\\A.txt", "01-E$\demo\\B.txt"],
    "recoveryDestination": "C:\\target"
  }
}
```

**Example 29** API response

```
https://mars:9090/nwrestapi/v3/global/jobs/704997
```

### Viewing the status of recovery operation

You can view the status of VMware file level recovery by using API request.

#### API request

Use GET request on the Job URI with recover Job ID to view the status of VMware file level recovery.

```
GET /jobs/{jobId}
```

**Example 30** URI example to view the status of recovery operation

**Example 30** URI example to view the status of recovery operation (continued)

```
https://mars:9090/nwrestapi/v3/global/jobs/704997
```

**Example 31** API response

```
200 OK
{
  "adhocJob": true,
  "command": "nsrvproxy_flr_recover -m 704689 -s mars -d \"C:\\\\target
  \ -I - -f",
  "completionStatus": "Succeeded",
  "dependentJobIds": [
    0
  ],
  "endTime": "2017-11-20T03:07:05-08:00",
  "exitCode": 0,
  "id": 704997,
  "itemIdLong": 704997,
  "links": [],
  "logFile": "C:\\Program Files\\EMC NetWorker\\nsr\\logs\\adhoc\\
  nsrvproxy_flr_recover\\704997.log",
  "message": "suppressed 18275 bytes of
  output.\\n159373:nsrvproxy_flr_recover:
  vProxy Log: 2017/11/20 03:06:46 INFO:
  [9101]
  -----\\r\\r\\
  -----\\r\\r\\
  n159373:nsrvproxy_flr_recover: vProxy Log: 2017/11/20 03:06:46
  INFO: [9101]
  Total Copied Skipped Mismatch FAILED Extras\\r\\r
  \\n159373:nsrvproxy_flr_recover:
  vProxy Log: 2017/11/20 03:06:46 INFO: [9101] Dirs :
  1 0 0 0
  0 \\r\\r\\n159373:nsrvproxy_flr_recover: vProxy Log:
  2017/11/20 03:06:46 INFO: [9101]
  Files : 1 1 0 0 0
  0\\r\\r\\
  n159373:nsrvproxy_flr_recover: vProxy Log: 2017/11/20 03:06:46
  INFO: [9101] Bytes :
  23 23 0 0 0 0\\r\\r
  \\n159373:nsrvproxy_flr_recover:
  vProxy Log: 2017/11/20 03:06:46 INFO: [9101] Times :
  0:00:00 0:00:00
  0:00:00 0:00:00\\r\\r\\n159373:nsrvproxy_flr_recover: vProxy Log:
  2017/11/20 03:06:46 INFO: [9101] Speed : 23000
  Bytes/sec.\\r\\r\\n159373:nsrvproxy_flr_recover: vProxy Log:
  2017/11/20 03:06:46 INFO: [9101] Speed : 1.316 MegaBytes/min.\\r
  \\r\\n159373:
  nsrvproxy_flr_recover: vProxy Log: 2017/11/20 03:06:46 INFO:
  [9101]
  Ended : Monday, November 20, 2017 3:06:46 AM\\r\\r
  \\n159373:nsrvproxy_flr_recover:
  vProxy Log: 2017/11/20 03:06:46 INFO: [9101] \\r\\r
  \\n159373:nsrvproxy_flr_recover:
  vProxy Log: 2017/11/20 03:06:46 INFO: [9101] \\r\\r
  \\n159373:nsrvproxy_flr_recover: vProxy Log:
  2017/11/20 03:06:46 INFO: [9101] File copying completed.\\r\\r
  \\n159373:nsrvproxy_flr_recover:
  vProxy Log: 2017/11/20 03:06:46 TRACE: [9101] C:\\Program Files
  (x86)\\EMC\\vProxy FLR Agent
  \\bin\\vflrcopy.exe exiting with status code 0\\r\\r
```

**Example 31** API response (continued)

```

\n152793:nsvproxy_flr_recover:
vProxy Log Ends =====\r
\r\n162217:
nsvproxy_flr_recover: FLR recover job completed successfully.\r\r
\n",
  "name": "VMWARE_FLR_4584_373032369185415",
  "ndmp": false,
  "parentJobId": 704996,
  "previousJobId": 0,
  "rootParentJobId": 704996,
  "runOnHost": "10.7.6.34",
  "siblingJobIds": [],
  "startTime": "2017-11-20T03:06:39-08:00",
  "state": "Completed",
  "stopped": true,
  "tenant": "",
  "terminationReason": "vProxy FLR recover job completed
successfully.",
  "type": "recover job"

```

## Creating VMware file level recovery to other virtual machine

This section describes the process to perform VMware file level recovery to other virtual machine. The process is similar to the process of creating VMware file level recovery to same virtual machine (VM). The only difference is that you need to mount the backup to a different target VM by selecting correct values for the attributes `targetVmMoref`, `targetVmName`, `targetVmUserId`, `targetVmUserPassword`, `targetVmAdminUserId` and `targetVmAdminUserPassword`

## Supporting Licenses View

This section describes the process to view the configured NetWorker licenses, by using the NetWorker REST API.

### Viewing the configured NetWorker licenses

The NetWorker licenses can be viewed using the URI `/licenses`.

**Example 32** API request

```
GET https://mars:9090/nwrestapi/v3/global/licenses
```

**Example 33** API response

```

200 OK
{
  "count": 2,
  "licenses": [
    {

```



**Example 33** API response (continued)

```

    "enablerCode": "none",
    "expirationDate": "2018-03-15T00:00:00-07:00",
    "hostId": "a99310e9",
    "licenseType": "E10",
    "links": [
      {
        "href": "https://mars:9090/nwrestapi/v3/global/licenses/33.0.216.26.0.0.0.0.38.106.51.90.10.1.7.174",
        "rel": "item"
      }
    ],
    "name": "NetWorker/10 Eval",
    "resourceId": {
      "id": "33.0.216.26.0.0.0.0.38.106.51.90.10.1.7.174",
      "sequence": 1
    }
  },
  {
    "comment": "testlicense",
    "enablerCode": "79fefb-3385a4-050bc3",
    "expirationDate": "2018-04-12T00:00:00-07:00",
    "hostId": "a99310e9",
    "licenseType": "D15",
    "links": [
      {
        "href": "https://mars:9090/nwrestapi/v3/global/licenses/43.0.108.33.0.0.0.0.227.85.67.90.10.1.7.174",
        "rel": "item"
      }
    ],
    "name": "NetWorker Module for SAP, UNIX Client/1",
    "resourceId": {
      "id": "43.0.108.33.0.0.0.0.227.85.67.90.10.1.7.174",
      "sequence": 1
    }
  }
]
}

```

**Viewing the configured NetWorker license for the given Resource ID**

A configured NetWorker license whose resource id is known can be viewed using the URI `/licenses/{licenseId}`

**Example 34** API request

```

GET https://mars:9090/nwrestapi/v3/global/licenses/33.0.133.27.0.0.0.0.165.61.212.89.10.31.227.66

```

**Example 35** API response

```

200 OK
{
  "comment": "testlicense",
  "enablerCode": "79fefb-3385a4-050bc3",

```

**Example 35** API response (continued)

```

"expirationDate": "2018-04-12T00:00:00-07:00",
"hostId": "a99310e9",
"licenseType": "D15",
"name": "NetWorker Module for SAP, UNIX Client/1",
"resourceId": {
  "id": "43.0.108.33.0.0.0.0.227.85.67.90.10.1.7.174",
  "sequence": 1
}
}

```

## Supporting Plugin Installer

This section describes the process to install EBR or VC plugins, by using the NetWorker REST API.

### Installing the EBR or VC plugins in the vCenter

The NetWorker REST API supports the installation of the EBR or VC plugin in the vCenter using the URI `vmware/vcenters/{vcenter-id}/plugins`.

The mandatory attributes are `pluginType`, `nwUserId`, and `nwPassword`. The optional attributes are `httpPort` (80) and `httpsPort` (443).

### API Request

Following URI helps you to install EBR or VC plugin:

```
POST /vmware/vcenters/{vcenter-id}/plugins
```

**Example 36** API request example for VC plugin type

```

POST https://mars:9090/nwrestapi/v3/global/vmware/vcenters/jupiter/
plugins
{
  "pluginType": "VC",
  "nwUserId": "administrator",
  "nwPassword": "Changeme@123",
  "httpPort": 80,
  "httpsPort": 443
}

```

**Example 37** API response example for VC plugin type

```

200 OK
{
  "resultCode": "Success",
  "data": "Registration Success",
  "debug": "",
  "responseString": null
}

```

**Example 37** API response example for VC plugin type (continued)

**Example 38** API request example for EBR plugin type

```
POST https://mars:9090/nwrestapi/v3/global/vmware/vcenters/jupiter/
plugins
{
  "pluginType": "EBR",
  "nwUserId": "administrator",
  "nwPassword": "Changeme@123",
  "httpPort": 80,
  "httpsPort": 443
}
```

**Example 39** API response example for EBR plugin type

```
200 OK
{
  "@class": "com.emc.nw.model.parameters.base.NwOutput",
  "status": "SUCCESS",
  "messages": {},
  "output":
  {
    "errorMessage": null,
    "registered": true
  }
}
```

## Viewing CloudBoost Appliances

This section describes the process to view the CloudBoost appliances by using the NetWorker REST API.

### Viewing the list of CloudBoost appliances

#### API request

The following API uses the NetWorker REST API to list CloudBoost appliances:

```
GET /nwrestapi/v3/global/cloudboostappliances
```

#### API response

```
200 OK
{
  "cloudBoostAppliances":
  [
    {
      "host": "mars",
      "links":
      [
        {
```

```

    "href": "https://mars:9090/nwrestapi/v3/global/
    cloudboostappliances/
    45.0.248.7.0.0.0.156.112.46.90.10.234.165.139",
    "rel": "item"
  }
],
"name": "marss",
"password": "*****",
"resourceId":{"id":
"45.0.248.7.0.0.0.156.112.46.90.10.234.165.139", "sequence": 7},
"username": "user_name"
}
],
"count": 1
}

```

### Viewing the configured NetWorker CloudBoost appliance for a given Resource ID

You can view the configured NetWorker CloudBoost appliance with a given resource ID (cloudboostapplianceid).

#### API request

```

GET /nwrestapi/v3/global/cloudboostappliances/
{cloudboostapplianceid}

```

#### API response

```

{
  "host": "jupiter",
  "name": "jupiter",
  "password": "*****",
  "resourceId":
  {
    "id": "45.0.248.7.0.0.0.156.112.46.90.10.234.165.139",
    "sequence": 7
  },
  "username": "user_name"
}

```

## Configuring Email Notification

You can receive an email notification reporting you the status of the action performed.

#### API request

The following API sets an email notification reporting the backup status:

```

POST /nwrestapi/v3/global/notifications
{
  "action": "/bin/mail -s \"host <pfrxbnw40>: Bus/Device reset
  detected\" root",
  "events": [
    "WriteCompletion"
  ],
  "name": "Bus_Tesat/Device Reset",
  "priorities": [
    "Info",
    "Notice",
    "Warning",

```

```

    "Waiting",
    "Critical",
    "Alert",
    "Emergency"
  ]
}

```

### API response

```
201 Created
```

## Viewing data domain systems

This section demonstrates how to view all data domain systems configured on a NetWorker server using the NetWorker REST API.

### API request

The following URI of NetWorker REST API is used to list all the data domain systems configured through NMC or nsradmin.

```
GET /nwrestapi/v3/global/datadomainsystems
```

### API response

The following response is received from the API:

```

200 OK
{
  "count": 1,
  "dataDomainSystems": [
    {
      "hosts": "jupiter",
      "links": [
        {
          "href": "https://mars:9090/nwrestapi/v3/global/
datadomainsystems/jupiter",
          "rel": "item"
        }
      ],
      "managementPort": "3009",
      "name": "jupiter",
      "storageNode": "nsrserverhost",
      "userName": "ost"
    }
  ]
}

```



# Troubleshooting

This appendix includes the following topics:

- [NetWorker REST API log files](#)..... 88
- [Changing the TLS level](#)..... 88
- [NetWorker REST API log file management](#)..... 89
- [NSR to REST API Command Map](#)..... 90
- [Status Codes and Messages](#)..... 93
- [Advance Client Attributes](#)..... 94

## NetWorker REST API log files

The following table provides a summary of the log files available for the NetWorker REST API.

**Table 8** NetWorker REST API log files

Component	File name and default location	Description
Installation log Tomcat Access log Apache Catalina log	Refer to the NetWorker Authentication Service logs for these log files.	The NetWorker REST API service is deployed in the same Apache Tomcat container as the NetWorker authentication service. The NetWorker REST API uses the same installation log, Tomcat Access log and Apache Catalina log with NetWorker Authentication service.
NetWorker REST API log	<ul style="list-style-type: none"> <li>• Linux: /nsr/logs/restapi/restapi.log</li> <li>• Windows: C:\Program Files\EMC NetWorker\nsr\logs\restapi\restapi.log</li> </ul>	Main NetWorker REST API log file.

## Changing the TLS level

From NetWorker 19.1 and later, TLS 1.1 and TLS 1.0 is no longer supported by default.

### Procedure

1. On the NetWorker, stop the NetWorker services.
2. Using a text editor, open the `server.xml` file.

The file is located in the following location:

- Windows: C:\Program Files\EMC NetWorker\nsr\authc-server\tomcat\conf
- Linux: /nsr/authc/conf

3. Locate the string `Connector port = 9090`, and then `sslEnabledProtocols="TLSv1.2, TLSv1.1, TLSv1"`.
4. Remove the TLS versions.
5. Save the `server.xml` file.
6. Start the NetWorker services.



## NetWorker REST API log file management

The NetWorker REST API uses the `Logback` API to manage log files.

To modify how NetWorker REST API manages the `restapi.log` log file, edit the `logback.xml` file, which is found in the following locations:

- **Linux:** `/nsr/authc/webapps/nwrestapi/WEB-INF/classes`
- **Windows:** `C:\Program Files\EMC NetWorker\nsr\authc-server\tomcat\webapps\nwrestapi\WEB-INF\classes`

This section describes how to modify the commonly used log attributes in the `logback.xml` file. Logback project documentation provides more detailed information about each attribute in the `logback.xml` file.

### Modifying the logging level

In the `logger` configuration element, the value of the `level` attribute defines the level of logging that the NetWorker REST API writes to the log files. By default, the NetWorker REST API sets the logging level to `info` and messages appear in the log files. There are five standard log levels: `trace`, `debug`, `info`, `warn`, `error`.

To change the logging level to `error`, modify the `level` attribute to appear as follows:

```
<logger name="com.emc.nw.webapi" level="error"/>
```

### Modifying the rollover period

In the `rollingPolicy` configuration element, the `fileNamePattern` element can be used to define the rollover period of the `restapi.log` file. When the log file reaches the end of a rolling period, the NetWorker REST API renames the log file for archival purposes and creates new log file. The rollover period is inferred from the value of `fileNamePattern`. By default, the NetWorker REST API sets the rollover period to `monthly`.

To change the rollover period to `daily`, modify the `fileNamePattern` element to appear as follows:

```
<fileNamePattern>${logdir}/restapi_%d.log</fileNamePattern>
```

Please refer to Logback project document on how to modify the rollover period.

### Modifying the number of rollover log files

In the `rollingPolicy` configuration element, the `maxHistory` element defines the number of `restapi.log` rollover log files that the NetWorker REST API maintains. It controls the maximum number of archive files to keep, deleting older files. By default, the NetWorker REST API maintains six rollover log files.

## NSR to REST API Command Map

Action	CLI command	REST API
Creating a client	<pre>nsradmin&gt; create type: NSR client; name: saturn; save set: /etc/ hosts type: NSR client; name: saturn; save set: /etc/hosts; Create? y created resource id 46.0.236.47.0.0.0.0.71.53. 50.87.10.5.167.140(1)</pre>	<pre>POST /nwrestapi/v3/global/ clients  {   "hostname": "saturn",   "saveSets": ["/etc/hosts"] }</pre>
Creating an Advanced File Type Device	<pre>nsradmin&gt; create type: NSR device; name: /space/ storage; device access information: /space/ storage; media type: adv_file  type: NSR device;  name: /space/storage; device access information: /space/ storage; media type: adv_file; Create? y created resource id 47.0.236.47.0.0.0.0.71.53. 50.87.10.5.167.140(1) nsradmin&gt; q</pre>	<pre>POST /nwrestapi/v3/global/ devices  {   "name": "/space/storage",   "deviceAccessInfo": "/ space/storage",   "mediaType": "adv_file" }</pre>
Labeling a device	<pre># nsrmm -l -m -y -b Default -f /space/storage Using volume name `mars.001' for pool `Default'</pre>	<pre>POST /nwrestapi/v3/global/ devices/{deviceName}/op/ label  {   "labelWithoutMount": "false",   "pool": "Default" }</pre>
Creating a protection group	<pre># nsrpolicy group create client -g EngineeringWorkstations - C saturn nsrpolicy: Group 'EngineeringWorkstations ' was successfully created</pre>	<pre>POST /nwrestapi/v3/global/ protectiongroups  {   "workItemType": "Client",   "name": "EngineeringWorkstations",   "workItems": [ "48.0.254.77.0.0.0.0.149.148 .51.87.128.222.109.201" ] }</pre>

Action	CLI command	REST API
<p><b>Creating a protection policy</b></p>	<pre># nsrpolicy policy create -p Engineering nsrpolicy: Policy 'Engineering' was successfully created</pre>	<pre>POST /nwrestapi/v3/global/ protectionpolicies {   "name": "Engineering" }</pre>
<p><b>Creating a workflow</b></p>	<pre># nsrpolicy workflow create -p Engineering -w ClientProtection -g EngineeringWorkstations - S 21:00 nsrpolicy: workflow 'ClientProtection' was successfully created</pre>	<pre>POST /nwrestapi/v3/global/ protectionpolicies/ Engineering/workflows {   "name": "ClientProtection",   "protectionGroups": [ "EngineeringWorkstations" ],   "startTime": "21:00" }</pre>
<p><b>Creating actions for a traditional backup</b></p>	<pre># nsrpolicy action create backup traditional -p Engineering -w ClientProtection -A Backup nsrpolicy: Assigned default schedule period of 'week' and schedule activity of 'full, incr, incr, incr, incr, incr, incr'. nsrpolicy: Action 'Backup' was given '1 Months' retention period nsrpolicy: Action 'Backup' was given 'nsrserverhost' storage node nsrpolicy: Action 'Backup' was given 'Default' destination pool nsrpolicy: Action 'Backup' was successfully created</pre>	<pre>PUT /nwrestapi/v3/global/ protectionpolicies/ Engineering/workflows/ ClientProtection {   "actions": [     {       "name": "Backup",       "actionSpecificData": {       "backup": { "backupSpecificData": { "traditional": {} } } } ] }</pre>
<p><b>Performing an on-demand backup</b></p>	<pre># nsrpolicy start -p Engineering -w ClientProtection -c saturn:/etc/ hosts 144091:nsrpolicy: Workflow 'Engineering/ ClientProtection' started and has job id 32023</pre>	<pre>POST /nwrestapi/v3/global/ protectionpolicies/ Engineering/workflows/ ClientProtection/op/backup {   "clients": [     "saturn:/etc/hosts"   ] }</pre>

Action	CLI command	REST API
	<pre> or  # nsrworkflow -p Engineering -w ClientProtection -c saturn:/etc/ hosts 133550:nsrworkflow: Starting Protection Policy 'Engineering' workflow 'ClientProtection'. 123316:nsrworkflow: Starting action 'Engineering/ ClientProtection/ Backup' with command: 'savegrp -Z backup:traditional -v'. 123321:nsrworkflow: Action 'Engineering/ ClientProtection/Backup's log will be in '/nsr/logs/policy/ Engineering/ ClientProtection/ Backup_032028.raw'. 123325:nsrworkflow: Action 'Engineering/ ClientProtection/Backup' succeeded. 133553:nsrworkflow: Workflow 'Engineering/ ClientProtection' succeeded.                     </pre>	
<p><b>Viewing all backups for a client</b></p>	<pre> # mminfo -avot -q "client=saturn" volume type client date time size ssid fl lvl name mars.001 adv_file mars 05/17/16 11:41:38 6 KB 4030413746 cb manual /etc/hosts                     </pre>	<pre> GET /nwrestapi/v3/global/ clients/ 82.0.31.22.0.0.0.33.12.41. 87.128.222.109.201/backups? q=clienthostname:mars                     </pre>
<p><b>Listing Client Index</b></p>	<pre> [root@rhel72-base ~]# nsrinfo -n mssql sql scanning client `sql' for all savetimes from the mssql namespace MSSQL:/, date=1556774598 Thu 02 May 2019 01:23:18 AM EDT MSSQL:\$/, date=1556774598 Thu 02 May 2019 01:23:18 AM EDT  02 objects found [root@rhel72-base ~]#                     </pre>	<pre> GET /nwrestapi/v3/global/ clients/ 46.0.12.48.0.0.0.18.7.200. 92.10.118.245.14/indexes? q=namespace:"mssql"                     </pre>

# Status Codes and Messages

NetWorker API status code is displayed to inform you the status of the request.

**Table 9** HTTP Status Code

Status	Description
200 - OK	The operations resulting into this HTTP state will be carrying the resource representation in the payload. All the GET and OPTIONS calls come under this category and the response payload will be different for each operation.
201 - Created	A certain POST operations in NetWorker REST API returns this status. This status indicates that a new resource or an intended job has been created and its associated URL is accessible from location header in the response.
202- Accepted	A few POST operations in NetWorker REST API take time to process and return the status. The 202 status code indicates that the API request has been accepted. The payload indicates the URL to the tracking resource instance which accessible from the location header of the response. T
204 - No Content	The status is displayed when the PUT and DELETE operations are successful in NetWorker REST API. The state indicates that the response payload is empty for this operation and there is no information. To know the status of the resource instance. use a GET request.
400	<p>The server could not process the request due to something that is perceived to be a client error. Request is either not well-formed, syntactically incorrect, or found to be invalid in the specific context. The most common causes include:</p> <ul style="list-style-type: none"> <li>• The payload is not specified in the request or a mandatory field is missing.</li> <li>• An unrecognized field is referenced in the payload or Query, possibly due to violation of case convention.</li> <li>• The payload is not as per data format defined in the NetWorker API specification.</li> <li>• Validation check failed, although request is found to be syntactically correct. For example, the DNS resolution check fails, if specified hostname cannot be resolved.</li> </ul>
401	Authentication failed due to invalid authentication credentials. Authentication information is missing or invalid. Authorization header must be set with a value that starts with “Basic” followed by a space and a base64-encoded “username: password” string.
403	Authorization failed due to inadequate privileges for the NetWorker API user to perform the specific operation. Ensure that the user has the appropriate User Group privileges to perform this operation.
404	The server did not find the resource that matches the requested URI. Either the URI is incorrect or the resource is not available anymore.
500	An internal application error prevented the server from fulfilling the request, possibly due to an unexpected condition during the task execution or communication failure with an internal component.

## Advance Client Attributes

Advanced client attributes can be used to perform special functions in NetWorker Rest API.

The following table lists the advanced attributes available in the NetWorker Rest API in 18.2 release.

**Table 10** Advanced attributes in 18.2 Release

Attribute	Description
serverNetworkInterface	This attribute specifies the name of the network interface on the server to be used for saves and recoveries.
dataDomainInterface	This attribute specifies the client interface over which a Client Direct backup to a Data Domain device should occur.
dataDomainBackup	This attribute specifies whether the backup destination for this client is a Data Domain device.
saveOperations	This attribute specifies the save operation instructions in the format: KEYWORD:TOKEN=STATE[:KEYWORD:TOKEN=STATE;...].
clientDirectEnabled	This attribute indicates if this client is enabled to bypass the storage node and write directly to the target disk device during a scheduled backup of the client.
jobControl	This attribute is intended to be used with custom backup scripts. It controls how savegrp and nsrjobd interpret the end of a job and its status.
Pool	This attribute specifies the media pool used for data target selection during a scheduled backup of the save sets specified in this client.
netWorkerVersion	This attribute contains the NetWorker version that is running on the client machine

The following table lists the advanced attributes available in the NetWorker Rest API in 19.1 release.

**Table 11** Advanced attributes in 19.1 Release

Attribute	Description
ndmpLogSuccessfulFileRecovery	This attribute specifies whether the logging shows the recovered files in the log file.
backupRenamedDirectoriesEnabled	This attribute specifies whether to include renamed directories in an incremental backup.

**Table 11** Advanced attributes in 19.1 Release (continued)

<b>Attribute</b>	<b>Description</b>
autoSelectStorageNodeEnabled	This attribute specifies whether the client can override the attributes of the backup target disks, storage nodes, and recover storage nodes.
checkPointGranularity	This attribute specifies whether to restart the backup from the point of failure at the directory or file level.
recoverStorageNodes	This attribute is an ordered list of storage nodes for the client to use when recovering its data.
backupTargetDisks	This is an ordered list of disk devices for the client to use when saving the data, without referring to the storage node lists.
prepolicyGroup	This attribute specifies the list of groups of which the client was a member before the policy was upgraded.
physicalHost	This attribute specifies the physical hostname, if this resource is for a virtual client.
saveSetMult	This attribute specifies the multi-byte transform of the save set names defined for the client.
indexPath	This attribute is used to allow the NetWorker administrator to balance NetWorker online file index disk utilization across multiple disk partitions.
priority	This attribute specifies the priority for the client.
virtualClient	This attribute indicates whether the client is a virtual machine.
fileInactivityAlertThreshold	This attribute specifies the percentage of space occupied by inactive files.
physicalClientParallelismEnabled	This attribute specifies whether to override the client parallelism and use the physical client's parallelism.
centralizedLogsCollection	This attribute enables or disables local logs collections into the centralized logs storage for this client.
fileInactivityThreshold	This attribute specifies the number of days a file has not been accessed before it is counted as inactive.
archiveServicesEnabled	This attribute determines whether archive services are available for the client.
shortFileNamesEnabled	This attribute determines the status of the short filename processing on the Windows client.
saveSessionDistribution	This attribute specifies the threshold for the save session distribution for all clients.
vbaHostType	This attribute specifies whether the client is for VBA server or proxy host.

**Table 11** Advanced attributes in 19.1 Release (continued)

<b>Attribute</b>	<b>Description</b>
ownerNotification	This attribute specifies a notification action to send the contents of status messages to the owner/primary user of a client (e.g. savegrp completion messages).