

IBM Storage Defender Copy Data Management

REST API Guide

2.3.1



Contents

RESTful API Overview	6
RESTful API Conventions	8
Account Management.....	10
Role	10
Resource Base URL.....	10
Related Resources.....	10
Methods.....	10
User	10
Resource Base URL.....	10
Related Resources.....	10
Methods.....	10
Session	11
Resource Base URL.....	11
Methods.....	11
Alert Management.....	12
Notification	12
Resource Base URL.....	12
Related Resources.....	12
Methods.....	12
Alert	12
Resource Base URL.....	12
Related Resources.....	12
Methods.....	13
Firetime	13
Resource Base URL.....	13
Related Resources.....	13
Methods.....	13
Analytics	14
Report.....	14
Resource Base URL.....	14
Related Resources.....	14
Methods.....	14
Report Instance.....	15
Resource Base URL.....	15
Related Resources.....	15
Methods.....	15
Catalog Management.....	16
Catalog	16
Resource Base URL.....	16
Related Resources.....	16
Methods.....	16
VMware Catalog.....	17
Resource Base URL.....	17
Related Resources.....	17
Methods.....	17
NetApp ONTAP Catalog	18
Resource Base URL.....	18
Related Resources.....	18
Methods.....	18
Application Catalog	18
Resource Base URL.....	18
Related Resources.....	19
Methods.....	19
Configuration Management	20
Policy	20
Resource Base URL.....	20
Related Resources.....	20
Methods.....	20
Event Management.....	21
Log	21
Resource Base URL.....	21
Related Resources.....	21
Methods.....	21

Job Management.....	22
Job	22
Resource Base URL.....	22
Related Resources.....	22
Methods.....	22
Jobsession	22
Resource Base URL.....	23
Related Resources.....	23
Methods.....	23
Trigger	23
Resource Base URL.....	23
Related Resources.....	23
Methods.....	24
License Management.....	25
License	25
Resource Base URL.....	25
Methods.....	25
Resource Providers	26
LDAP	26
Resource Base URL.....	26
Related Resources.....	26
Methods.....	26
NetApp ONTAP	26
Resource Base URL.....	27
Related Resources.....	27
Methods.....	27
vCloud	28
Resource Base URL.....	28
Related Resources.....	28
Methods.....	28
vSphere	29
Resource Base URL.....	29
Related Resources.....	29
Methods.....	29
SMTP	31
Resource Base URL.....	31
Related Resources.....	31
Methods.....	31
Search Management.....	33
Search	33
Resource Base URL.....	33
Methods.....	33
Snapshot Management.....	34
Offload Snapshot.....	35
Restore Snapshot	37
List Sessions	38
Get Session Status	40
Examples	42
VMware-NetApp Snapshot Policy	42
Create a Catalog Job	44
Create a Policy	45
Execute a Search	47
Getting a Session ID	49
Monitor a Catalog Job	50
Register a Resource Provider	54
VMware-NetApp Recovery Policy	57
Start a Catalog Job	58
Offload and Restore Snapshot Workflow.....	61
Notices.....	65
Trademarks.....	66
Terms and conditions for product documentation.....	66
Privacy policy considerations	67
Glossary	68
Index	69

Note:

Before you use this information and the product it supports, read the information in “Notices” on page 65.

This edition applies to version 2, release 3, modification 1 of IBM® Storage Defender Copy Data Management (product numbers 5737-B34, 5641-CD4, 5641-CD5, 5641-CD6) and to all subsequent releases and modifications until otherwise indicated in new editions.

RESTful API Overview

The RESTful APIs that make up IBM® Storage Defender Copy Data Management are described below.

Account Management

A set of RESTful APIs used for account management processes. This group of APIs handles authorizing a user to perform certain actions based on the role they belong to and the permissions allowed by that role. It also is responsible for authenticating users. These users can reside in an LDAP system or be native users that are managed internally. The Session API provides a session key that is used by all other APIs.

- [“Role” on page 10](#)
- [“User” on page 10](#)
- [“Session” on page 11](#)

Alert Management

A set of RESTful APIs used to facilitate system alerts such as notifications, alert, and firetime.

- [“Notification” on page 12](#)
- [“Alert” on page 12](#)
- [“Firetime” on page 13](#)

Analytics

A set of RESTful APIs used to manage reports.

- [“Report” on page 14](#)
- [“Report Instance” on page 15](#)

Catalog Management

A set of RESTful APIs used to manage the storage, retrieval, and organization of objects and metadata. Also managed by these RESTful APIs are statistics used within the dashboard and an aggregation framework used by our analytics engine.

- [“Catalog” on page 16](#)
- [“NetApp ONTAP Catalog” on page 18](#)
- [“VMware Catalog” on page 17](#)
- [“Application Catalog” on page 18](#)

Configuration Management

A set of RESTful APIs relative to jobs. Policies dictate what set of resources, managed by resource providers, participate in a cataloging operation. Policies also govern the execution of reports and script based jobs.

- [“Policy” on page 20](#)

Event Management

A set of RESTful APIs used to capture and report on system events. Log management falls into this category. Jobs produce events and those events are persisted and manifested as job log events.

- [“Log” on page 21](#)

Job Management

A set of RESTful APIs used to control and monitor jobs. Jobs are asynchronous, potentially long-running activities that you can start, stop, schedule, monitor, and report on. Jobs entail cataloging or ingesting data and metadata via resource providers, running analytical processes with the Report Management RESTful APIs, or running script-based jobs.

- [“Job” on page 22](#)
- [“Jobsession” on page 22](#)

- [“Trigger” on page 23](#)

License Management

A set of RESTful APIs used for license management processes such as the retrieval of license information.

- [“License” on page 25](#)

Resource Providers

A set of RESTful APIs that performs actions against managed objects. They allow integration with external resources: LDAP, NetApp ONTAP, vCloud, vSphere, SMTP.

- [“LDAP” on page 26](#)
- [“NetApp ONTAP” on page 26](#)
- [“vCloud” on page 28](#)
- [“vSphere” on page 29](#)
- [“SMTP” on page 31](#)

Search

A set of RESTful APIs used for search management processes such as performing search operations.

- [“Search” on page 33](#)

VMware and NetApp Snapshot Policy

A set of RESTful APIs used to programmatically create a Backup snapshot policy.

- [“VMware-NetApp Snapshot Policy” on page 42](#)

VMware and NetApp Recovery Policy

A set of RESTful APIs used to programmatically create a Recovery policy.

- [“VMware-NetApp Recovery Policy” on page 57](#)

Snapshot Management

The snapshot management RESTful APIs allow users to manage asynchronous offload and restore operations for IBM Storage FlashSystem snapshots.

- [“Snapshot Management” on page 34](#)

RESTful API Conventions

The following concepts and formatting details are used in the IBM® Storage Defender Copy Data Management REST API documentation.

JSON

RESTful API requests and responses are in JSON format.

HATEOAS

Responses also support Hypermedia as the Engine of Application State. The main component of HATEOAS is the notion of discoverability. The client application must discover, via Hypermedia or response links, resources that are accessible within the response. This means the client does not require up-front knowledge of methods of resource interaction.

HATEOAS is primarily driven by the response's link format. The examples below offer a sense of how the RESTful API is designed and what responses are expected.

Discover ability Links

All responses include a links field. The link field is a JSON object that allows links to be accessed iteratively as well as by name directly:

Predefined Named Links

- **self**
A link to the current resource.
- **up**
A navigation link; up one level in the hierarchy.
- **create**
A link to create a resource.
- **edit**
A link to edit a resource.
- **delete**

Headers

The following headers are required for every RESTful API operation:

- **X-endeavour-sessionid:** with a valid sessionid value.
- **Content-Type:** with the value set to *application/json*.
- **Accept:** with the value set to *application/json*.

Pagination

Pagination is supported in the REST API using GET.

Sorting

Sorting is supported in the REST API using GET.

Filtering

Filtering of results is supported. The following is an example of filtering schedules, without the line breaks:


```
GET
https://<ip>:<port>/api/endeavour/trigger?filterType=all&filter=[
{
  "property": "name",
  "op": "=",
  "value": "Daily"
},
{
  "property": "creationDate",
  "op": "<",
  "value": "1234567890"
}
]
```

Note: filterType can either be "all" or "any". All must match all criteria, any must match any criteria. Also, Operator is an SQL operator, such as "<", "<=", "=", "!=", ">=", ">". If "operator" is not specified, "=" is the default value.

Embedded Objects

You can embed a related link in the response. To do this, add the embed query parameter. Following is the parameter format:

```
GET
https://<ip>:<port>/api/endeavour/ api/endeavour/job?embed=(policy)
```

Tip: This eliminates round trips to the server when retrieving job resources that are tied to specific policies. An embedded policy object can be returned for each job within the response of a job list request.

Actions

RESTful API leverages HTTP verbs to perform resource-based operations. These verbs are typically mapped to CRUD operations; create, read, update, delete. Create maps to Post, Read maps to GET, Update maps to Put, and Delete maps to Delete. What is missing from the standard RESTful API verbs is one that maps to the semantics of an action. Endeavour leverages the POST verb to perform actions. The POST verb along with a JSON request body manages the execution of various actions. The action is specified as a query parameter called "action" on the URI. The following is an example of performing an action:

```
POST
https://<ip>:<port>/api/vsphere/{vsphereId}/vm/{vmid}?action=createSnapshot

REQUEST BODY
{
  "name": "snapshot.1"
}
```

Related information

[RESTful API Overview](#)

Account Management

A set of RESTful APIs used for account management processes. This group of APIs handles authorizing a user to perform certain actions based on the role they belong to and the permissions allowed by that role. It also is responsible for authenticating users. These users can reside in an LDAP system or be native users that are managed internally. The Session API provides a session key that is used by all other APIs.

Related information

[Role](#)

[User](#)

[Session](#)

Role

You can get information about roles in IBM® Storage Defender Copy Data Management through the REST API.

Resource Base URL

```
http://<ipaddress>:<port>/api/security/role
```

Related Resources

[User](#)

[Session](#)

Methods

The following RESTful API methods and endpoints are used to view the configured roles and the user accounts assigned to them:

Table 1: Role API		
API	Endpoint	Description
Get	/api/security/role	Get a list of roles.
Get	/api/security/role/{roleId}	Get information about a single role.

User

You can get information about the users in IBM® Storage Defender Copy Data Management through the REST API.

Resource Base URL

```
http://<ipaddress>:<port>/api/security/user
```

Related Resources

[Role](#)

[Session](#)

Methods

The following RESTful API methods and endpoints are used to add, view, edit, and delete user accounts:

Table 2: User API		
API	Endpoint	Description
Get	/api/security/user	Get a list of security users including native users as well as provisioned LDAP users.
Get	/api/security/user{userId}	Get information about a specific user with the specified user ID.
Get	/api/security/user{userId}/role	Get information about roles assigned to a user with the specified user ID.
Post	/api/security/user	Add a user.
Post	/api/security/user{userId}	Change the password for a user with the specified user ID.
Post	/api/security/user{userId}/role	Assign a role to a user with the specified user ID.
Put	/api/security/user{userId}	Update a user with the specified user ID.
Delete	/api/security/user{userId}	Delete a user with the specified user ID.
Delete	/api/security/user{userId}/role/{roleId}	Delete the specified role with role ID from the specified user with the user ID.

Session

You can get information about the session in IBM® Storage Defender Copy Data Management through the REST API.

Resource Base URL

```
http://<ipaddress>:<port>/api/endeavour/session
```

Methods

The following RESTful API methods and endpoints are used to login and logout a user session:

Table 3: Session API		
API	Endpoint	Description
Post	/api/endeavour/session	Login to IBM® Storage Defender Copy Data Management.
Delete	/api/endeavour/session	Logout of IBM® Storage Defender Copy Data Management.

Alert Management

A set of RESTful APIs used to facilitate system alerts such as notifications, alerts, and firetime.

Related information

[Notification](#)

[Alert](#)

[Firetime](#)

Notification

You can get information about notifications in IBM® Storage Defender Copy Data Management through the REST API.

Resource Base URL

```
http://<ipaddress>:<port>/api/endeavour/notification
```

Related Resources

[Alert](#)

[Firetime](#)

Methods

The following RESTful API methods and endpoints are used to view, add, update, and delete notifications such as emails:

Table 4: Notification API		
API	Endpoint	Description
Get	/api/endeavour/notification	Get notifications in IBM® Storage Defender Copy Data Management.
Get	/api/endeavour/notification/{notificationId}	Get a specific notification by using the notification ID.
Post	/api/endeavour/notification/	Add a notification.
Post	/api/endeavour/notification/aggregate	Aggregate.
Put	/api/endeavour/notification/{notificationId}	Update a notification by using the notification ID.
Delete	/api/endeavour/notification/{notificationId}	Delete a notification by using the notification ID.

Alert

You can get information about alerts in IBM® Storage Defender Copy Data Management through the REST API.

Resource Base URL

```
http://<ipaddress>:<port>/api/endeavour/alert
```

Related Resources

[Notification](#)

Firetime

Methods

The following RESTful API methods and endpoints are used to get, add, and delete alerts:

Table 5: Alert API		
API	Endpoint	Description
Get	/api/endeavour/alert	Returns a list of alerts.
Get	/api/endeavour/alert/{alertId}	Returns information about a specific alert.
Post	/api/endeavour/alert	Add an alert.
Post	/api/endeavour/alert/aggregate	Aggregate.
Put	/api/endeavour/alert/{alertId}	Update a specific alert.
Delete	/api/endeavour/alert/{alertId}	Delete a specific alert.

Firetime

You can get information about firetimes in IBM® Storage Defender Copy Data Management through the REST API.

Resource Base URL

```
http://<ipaddress>:<port>/api/endeavour/firetime
```

Related Resources

[“Notification” on page 12](#)

[Alert](#)

Methods

The following RESTful API methods and endpoints are used to get firetimes:

Table 6: Firetime API		
API	Endpoint	Description
Get	/api/endeavour/firetime	Get firetimes in IBM® Storage Defender Copy Data Management.
Get	/api/endeavour/{firetimeId}	Get a specific firetime by using the firetime ID.

Analytics

A set of RESTful APIs used to manage reports.

Related information

[Report](#)

[Report Instance](#)

Report

You can get information about reports in IBM® Storage Defender Copy Data Management through the REST API.

Resource Base URL

```
http://<ipaddress>:<port>/api/endeavour/report
```

Related Resources

[Report Instance](#)

Methods

The following RESTful API methods and endpoints are used to add, view, edit, and delete reports:

Table 7: Report API		
API	Endpoint	Description
Get	/api/endeavour/report	Get all reports in IBM® Storage Defender Copy Data Management.
Get	/api/endeavour/report/categories	Get all report categories.
Get	/api/endeavour/report/categories{categoryStr}	Get a report category by using the category string.
Get	/api/endeavour/report/download/sample{parentName}	Download sample report.
Get	/api/endeavour/report/engine	Get the report engine.
Get	/api/endeavour/report/{reportIdStr}	Get a report by using the report ID string.
Get	/api/endeavour/report/{reportIdStr}/parameter	Get report parameters.
Get	/api/endeavour/report/{reportIdStr}/parameter/{parameterStr}	Get report parameters by using the specified parameter string.
Get	/api/endeavour/report/{reportIdStr}/parameter/{parameterStr}/values	Get the parameter values of a report by using the report ID string and parameter string.
Get	/api/endeavour/report/{reportIdStr}/schema	Get the report schema.
Post	/api/endeavour/report	Create a report.
Post	/api/endeavour/report/{reportIdStr}	Create a report action by using the report ID string.
Put	/api/endeavour/report/{reportIdStr}	Update a report by using the report ID string.

API	Endpoint	Description
Delete	/api/endeavour/report/ {reportIdStr}	Delete a report by using the report ID string.

Report Instance

You can get information about report instances in IBM® Storage Defender Copy Data Management through the REST API.

Resource Base URL

`http://<ipaddress>:<port>/api/endeavour/reportInstance`

Related Resources

[“Report” on page 14](#)

Methods

The following RESTful API methods and endpoints are used to add, view, edit, and delete report instances:

Table 8: Report Instance API		
API	Endpoint	Description
Get	/api/endeavour/reportInstance	Get all report instances in IBM® Storage Defender Copy Data Management.
Get	/api/endeavour/ reportInstance/schema	Get the report instance schema.
Get	/api/endeavour/ reportInstance/ {reportInstanceIdStr}	Get the report instance by using the report instance ID string.
Get	/api/endeavour/ reportInstance/ {reportInstanceIdStr}/ download/{formatType}	Download the report by using the report instance ID string and format type.
Get	/api/endeavour/ reportInstance/ {reportInstanceIdStr}/view/ {formatType}	View the report in HTML type by using the report instance ID string and format type.
Get	/api/endeavour/ reportInstance/download/ image/{directory}/{fileName}	Download the report by specifying the directory and filename.
Post	/api/endeavour/reportInstance	Create a report instance.
Post	/api/endeavour/ reportInstance/ {reportInstanceIdStr}/ download	Download the report by using the report instance ID string.
Delete	/api/endeavour/ reportInstance/ {reportInstanceIdStr}	Delete the report instance by using the report instance ID string.

Catalog Management

A set of RESTful APIs used to manage the storage, retrieval, and organization of objects and metadata. Also managed by these RESTful APIs are statistics used within the dashboard and an aggregation framework used by our analytics engine.

Related information

[Catalog](#)
[VMware Catalog](#)
[NetApp ONTAP Catalog](#)
[Application Catalog](#)

Catalog

You can get information about the catalog in IBM® Storage Defender Copy Data Management through the REST API.

Resource Base URL

```
http://<ipaddress>:<port>/api/endeavour/catalog
```

Related Resources

[“VMware Catalog” on page 17](#)

[“NetApp ONTAP Catalog” on page 18](#)

[“Application Catalog” on page 18](#)

Methods

The following RESTful API methods and endpoints are used to view, update, and delete supported catalog objects:

Table 9: Catalog API		
API	Endpoint	Description
Get	/api/endeavour/catalog	Get all supported catalogs in IBM® Storage Defender Copy Data Management.
Get	/api/endeavour/catalog{supportedCatalogName}	Get the supported catalog by using the catalog name.
Get	/api/endeavour/catalog{supportedCatalogName}/{supportedObjectName}	Get the supported catalog object by using the catalog name.
Get	/api/endeavour/catalog{supportedCatalogName}/{supportedObjectName}/{objectId}	Get the supported catalog object by using the object ID.
Post	/api/endeavour/catalog	Add the supported catalog.
Post	/api/endeavour/catalog{supportedCatalogName}	Add the supported catalog object by using the catalog name.
Post	/api/endeavour/catalog{supportedCatalogName}/{supportedObjectName}	Add a supported object by name using the supported catalog name.

API	Endpoint	Description
Put	/api/endeavour/catalog{supportedCatalogName}/{supportedObjectName}/{objectId}	Update the object by using the object ID of the supported object name by using the supported catalog name.
Delete	/api/endeavour/catalog{supportedCatalogName}	Delete the catalog by using the catalog name.
Delete	/api/endeavour/catalog{supportedCatalogName}/{supportedObjectName}	Delete the supported object name by using the catalog name.
Delete	/api/endeavour/catalog{supportedCatalogName}/{supportedObjectName}/{objectId}	Delete the object using the object ID of the supported object name by using the supported catalog name.

VMware Catalog

You can get information about the VMware catalog in IBM® Storage Defender Copy Data Management through the REST API.

Resource Base URL

```
http://<ipaddress>:<port>/api/endeavour/catalog/vmware
```

Related Resources

[“Catalog” on page 16](#)

[“NetApp ONTAP Catalog” on page 18](#)

[“Application Catalog” on page 18](#)

Methods

The following RESTful API methods and endpoints are used to add, view, edit, and delete VMware objects:

Table 10: VMware Catalog API		
API	Endpoint	Description
Get	/api/endeavour/catalog/vmware	Get a list of supported objects in the VMware catalog in IBM® Storage Defender Copy Data Management.
Get	/api/endeavour/catalog/vmware{supportedObjectName}	Get a list of objects of type by using the supported object name.
Get	/api/endeavour/catalog/vmware/{supportedObjectName}/{objectId}	Get the properties of a supported object by using the object ID.
Get	/api/endeavour/catalog/vmware/{supportedObjectName}/pk.{objectPk}	Get the object by using the pk object.
Get	/api/endeavour/catalog/vmware/{supportedObjectName}/pk.{objectPk}/recoverypoint	Get the object by the pk object recovery point.
Post	/api/endeavour/catalog/vmware/{supportedObjectName}	Create an action on the catalog by using the supported object name.
Put	/api/endeavour/catalog/vmware/{supportedObjectName}/{objectId}	Update the properties of a supported object by using the object ID.

API	Endpoint	Description
Delete	/api/endeavour/catalog/vmware/{supportedObjectName}/{objectId}	Delete a supported object by using the object ID.

NetApp ONTAP Catalog

You can get information about the NetApp ONTAP catalog in IBM® Storage Defender Copy Data Management through the REST API.

Resource Base URL

```
http://<ipaddress>:<port>/api/endeavour/catalog/netapp
```

Related Resources

[“Catalog” on page 16](#)

[“VMware Catalog” on page 17](#)

[“Application Catalog” on page 18](#)

Methods

The following RESTful API methods and endpoints are used to view, update, and delete supported NetApp ONTAP Catalog objects:

Table 11: NetApp Catalog API		
API	Endpoint	Description
Get	/api/endeavour/catalog/netapp	Get a list of supported objects in the NetApp ONTAP catalog in IBM® Storage Defender Copy Data Management.
Get	/api/endeavour/catalog/netapp{supportedObjectName}	Get a list of objects of type by using the supported object name.
Get	/api/endeavour/catalog/netapp{supportedObjectName}/{objectId}	Get the properties of a supported object by using the object ID.
Post	/api/endeavour/catalog/netapp/{supportedObjectName}	Create an action on the catalog by using the supported object name.
Put	/api/endeavour/catalog/netapp/{supportedObjectName}/{objectId}	Update the properties of a supported object by using the object ID.
Delete	/api/endeavour/catalog/netapp/{supportedObjectName}/{objectId}	Delete a supported object by using the object ID..

Application Catalog

You can get information about the application catalog in IBM® Storage Defender Copy Data Management through the REST API.

Resource Base URL

```
http://<ipaddress>:<port>/api/endeavour/catalog/application
```

Related Resources

Catalog

VMware Catalog

NetApp ONTAP Catalog

Methods

The following RESTful API methods and endpoints are used to add, view, edit, and delete supported Application Catalog objects:

Table 12: Application Catalog API		
API	Endpoint	Description
Get	/api/endeavour/catalog/application	Get a list of application objects.
Get	/api/endeavour/catalog/application/{supportedObjectName}	Get an application object by name.
Get	/api/endeavour/catalog/application/{supportedObjectName}/{objectId}	Get an application object using an object ID.
Post	endeavour/catalog/application/{supportedObjectName}	Perform action on an application object by name.
Put	/api/endeavour/catalog/application/{supportedObjectName}/{objectId}	Update an application object using an object ID.
Delete	/api/endeavour/catalog/application/{supportedObjectName}/{objectId}	Delete an application object using an object ID.

Configuration Management

A set of RESTful APIs relative to configuration. Policies specify what storage vendor and type of protection is used.

Related information

[Policy](#)

Policy

You can get information about policies in IBM® Storage Defender Copy Data Management through the REST API.

Resource Base URL

```
http://<ipaddress>:<port>/api/endeavour/policy
```

Related Resources

Methods

The following RESTful API methods and endpoints are used to view, add, edit, and delete catalog policies:

Table 13: NetApp Catalog API		
API	Endpoint	Description
Get	/api/endeavour/policy	Get a list of all policies in IBM® Storage Defender Copy Data Management.
Get	/api/endeavour/policy/{policyId}	Get a specific policy by using the policy ID.
Post	/api/endeavour/policy	Add a policy.
Put	/api/endeavour/policy/{policyId}	Edit a specific policy by using the specific policy ID.
Delete	/api/endeavour/policy/{policyId}	Delete a specific policy by using the specific policy ID.

Event Management

A set of RESTful APIs used to capture and report on system events. Log management falls into this category. Jobs produce events and those events are persisted and manifested as job log events.

Related information

[Log](#)

Log

You can get information about logs in IBM® Storage Defender Copy Data Management through the REST API.

Resource Base URL

```
http://<ipaddress>:<port>/api/endeavour/log
```

Related Resources

Methods

The following RESTful API methods and endpoints are used to retrieve job logs:

Table 14: Log API		
API	Endpoint	Description
Get	/api/endeavour/log/job/{logId}	Get a specific job log by using the log ID.
Get	/api/endeavour/log/job/	Get a list of all job logs.
Get	/api/endeavour/log/job/download/json	Download job logs in JSON format.
Put	/api/endeavour/log/job	Add a job log.
Delete	/api/endeavour/log/job/jobsession/{jobsessionId}	Delete job logs of a particular job session by using the specific job session ID.

Job Management

A set of RESTful APIs used to control and monitor jobs. Jobs are asynchronous, potentially long-running activities that you can start, stop, schedule, monitor, and report on. Jobs entail cataloging or ingesting data and metadata via resource providers, running analytical processes with the Report Management RESTful APIs, or running script-based jobs.

Related information

[Job](#)

[Jobsession](#)

[Trigger](#)

Job

You can get information about jobs in IBM® Storage Defender Copy Data Management through the REST API.

Resource Base URL

```
http://<ipaddress>:<port>/api/endeavour/job
```

Related Resources

[“Jobsession” on page 22](#)

[“Trigger” on page 23](#)

Methods

The following RESTful API methods and endpoints are used to list, add, edit, and delete jobs:

Table 15: Job API		
API	Endpoint	Description
Get	/api/endeavour/job	Get a job list.
Get	/api/endeavour/job/{jobId}	Get a specific job.
Get	/api/endeavour/job/{jobId}/trigger	Get a list of triggers.
Post	/api/endeavour/job	Create a job.
Post	/api/endeavour/job/{jobId}?action=start	Start a job.
Put	/api/endeavour/job/{jobId}	Edit the job.
Put	/api/endeavour/job/{jobId}/{triggerId}	Associate the trigger to a job.
Delete	/api/endeavour/job/{jobId}	Delete a specific job.
Delete	/api/endeavour/job/{jobId}/trigger/{triggerId}	Delete a trigger associated with a job.

Jobsession

You can get information about job sessions in IBM® Storage Defender Copy Data Management through the REST API.

Resource Base URL

`http://<ipaddress>:<port>/api/endeavour/jobsession`

Related Resources

[“Jobsession” on page 22](#)

[“Trigger” on page 23](#)

Methods

The following RESTful API methods and endpoints are used to retrieve information on job sessions and stop a running job:

Table 16: Jobsession API		
API	Endpoint	Description
Get	<code>/api/endeavour/jobsession</code>	Get a list of all jobsessionId (instances) that have run or are running.
Get	<code>/api/endeavour/jobsession/stats</code>	Get statistics of job sessions run during the default past number of days.
Get	<code>/api/endeavour/jobsession/stats/{days}</code>	Get statistics of a jobsession run as specified in the past number of days.
Get	<code>/api/endeavour/jobsession/stats/jobid/{jobId}</code>	Get statistics of a jobsessionId by using the job ID.
Get	<code>/api/endeavour/jobsession/trends</code>	Get completion trends of all jobs in the past default days.
Get	<code>/api/endeavour/jobsession/trends/{days}</code>	Get completion trends of all jobs as specified in the past number of days.
Get	<code>/api/endeavour/jobsession/{sessionId}</code>	Get a specific job session by using the session ID.
Get	<code>/api/endeavour/jobsession/{sessionId}/task</code>	Get a list of tasks of a specific job by using the session ID.
Get	<code>/api/endeavour/jobsession/{sessionId}/task/{taskId}</code>	Get a task of a specific job session using session ID and the associated task using the task ID.
Post	<code>/api/endeavour/jobsession/aggregate</code>	Aggregate.
Post	<code>/api/endeavour/jobsession/{sessionId}</code>	Start a job session by using the session ID.

Trigger

You can get information about triggers in IBM® Storage Defender Copy Data Management through the REST API.

Resource Base URL

`http://<ipaddress>:<port>/api/endeavour/trigger`

Related Resources

[“Job” on page 22](#)

[“Jobsession” on page 22](#)

Methods

The following RESTful API methods and endpoints are used to add, view, edit, and delete triggers:

Table 17: Trigger API		
API	Endpoint	Description
Get	/api/endeavour/trigger	Get a list of all triggers in IBM® Storage Defender Copy Data Management.
Get	/api/endeavour/trigger/{triggerId}	Get the trigger as specified by using the trigger ID.
Post	/api/endeavour/trigger	Create a trigger.
Put	/api/endeavour/trigger/{triggerId}	Edit the specified trigger by using the trigger ID.
Delete	/api/endeavour/trigger/{triggerId}	Delete the specified trigger by using the trigger ID.

License Management

A set of RESTful APIs used for license management processes such as the retrieval of license information.

Related information

[License](#)

License

You can get information about licenses in IBM® Storage Defender Copy Data Management through the REST API.

Resource Base URL

```
http://<ipaddress>:<port>/api/endeavour/license
```

Methods

The following RESTful API methods and endpoints are used to retrieve license information:

Table 18: License API		
API	Endpoint	Description
Get	/api/endeavour/license	Get a list of all licenses in IBM® Storage Defender Copy Data Management.
Get	/api/endeavour/license/{licenseName}	Get the license by using the specified license name.
Get	/api/endeavour/license/{licenseName}/schema	Get the license schema by using the specified license name.

Resource Providers

A set of RESTful APIs that performs actions against managed objects. They allow integration with external resources: LDAP, NetApp ONTAP, vCloud, vSphere, SMTP.

IBM® Storage Defender Copy Data Management wraps these SDKs and exposes their capabilities through what are called resource providers.

Related information

[LDAP](#)

[vCloud](#)

[vSphere](#)

[SMTP](#)

LDAP

You can get information about LDAP resource providers in IBM® Storage Defender Copy Data Management through the REST API.

Resource Base URL

```
http://<ipaddress>:<port>/api/ldap
```

Related Resources

[“NetApp ONTAP” on page 26](#)

[“vCloud” on page 28](#)

[“vSphere” on page 29](#)

[“SMTP” on page 31](#)

Methods

The following RESTful API methods and endpoints are used to add, view, edit, and delete LDAP providers:

Table 19: LDAP API		
API	Endpoint	Description
Get	/api/ldap	Returns a list of configured LDAP providers.
Get	/api/ldap/{ldapId}	Returns information about a single LDAP provider.
Get	/api/ldap/{ldapId}/user	Get a list of LDAP users.
Get	/api/ldap/{ldapId}/user/{userId}	Create an LDAP user.
Post	/api/ldap	Register an LDAP provider.
Put	/api/ldap/{ldapId}	Update an LDAP provider.
Delete	/api/ldap/{ldapId}	Deregister an LDAP provider.

NetApp ONTAP

You can get information about NetApp ONTAP storage in IBM® Storage Defender Copy Data Management through the REST API.

Resource Base URL

`http://<ipaddress>:<port>/api/netapp`

Related Resources

[“LDAP” on page 26](#)

[“vCloud” on page 28](#)

[“vSphere” on page 29](#)

[“SMTP” on page 31](#)

Methods

The following RESTful API methods and endpoints are used to query, register, update, and delete NetApp ONTAP providers:

Table 20: NetApp API		
API	Endpoint	Description
Get	/api/netapp	Get a list of all registered NetApp ONTAP in IBM® Storage Defender Copy Data Management.
Get	/api/netapp/{id}	Get a registered NetApp ONTAP by using the specified ID.
Get	/api/netapp/{netappId}/volume	Get a list of volumes on the NetApp ONTAP as specified by the NetApp ID.
Get	/api/netapp/{netappId}/volume/{volName}	Get a node volume as specified by the volume name on the NetApp ONTAP as specified by the NetApp ID.
Get	/api/netapp/{netappId}/volume/{volName}/qtree	Get a list of qtrees on the volume as specified for volume name on the NetApp ONTAP as specified by the NetApp ID.
Get	/api/netapp/{netappId}/volume/{volName}/qtree/{qtreeName}	Get a qtree as specified by the qtree name on the volume as specified by the volume name on the NetApp ONTAP as specified by the NetApp ID.
Get	/api/netapp/{netappId}/volume/{volName}/snapshot	Get a list of snapshots on the volume as specified by the volume name on the NetApp ONTAP as specified by the NetApp ID.
Get	/api/netapp/{netappId}/volume/{volName}/snapshot/{snapName}	Get a snapshot by using a snapshot name on the volume specified by the volume name on the NetApp ONTAP as specified by the NetApp ID.
Post	/api/netapp	Register a NetApp ONTAP as a storage provider.
Post	/api/netapp/{netappId}/volume/{volName}	
Post	/api/netapp/{netappId}/volume/{volName}/snapshot	Create a snapshot on the specified volume by using the volume name on the specified NetApp ONTAP as specified by the NetApp ID.

API	Endpoint	Description
Put	/api/netapp/{id}	Edit a registered NetApp ONTAP as specified by the ID.
Delete	/api/netapp/{id}	Unregister the NetApp ONTAP as specified by the ID.
Delete	/api/netapp/{netappId}/volume/{volName}/snapshot/{snapName}	Delete a snapshot by using the snapshot name on the volume using volume name on the NetApp ONTAP by using the NetApp ID.

vCloud

You can get information about vCloud providers in IBM® Storage Defender Copy Data Management through the REST API.

Resource Base URL

```
http://<ipaddress>:<port>/api/vcloud
```

Related Resources

[“LDAP” on page 26](#)

[“NetApp ONTAP” on page 26](#)

[“vSphere” on page 29](#)

[“SMTP” on page 31](#)

Methods

The following RESTful API methods and endpoints are used to add, view, edit, and delete vCloud providers:

Table 21: vCloud API		
API	Endpoint	Description
Get	/api/vcloud	Get a list of all registered vCloud providers in IBM® Storage Defender Copy Data Management.
Get	/api/vcloud/{vcloudid}	Get information about a registered vCloud using the vCloud ID.
Get	/api/vcloud/{vcloudid}/org	Get a list of organizations of a vCloud using the vCloud ID.
Get	/api/vcloud/{vcloudid}/org/{orgId}	Get information about a specific organization of a vCloud using the vCloud ID and org ID.
Get	/api/vcloud/{vcloudid}/org/{orgId}/vdc	Get a list of vDCs in an organization of a vCloud using the vCloud ID and the org ID.
Get	/api/vcloud/{vcloudid}/vapp/{vappId}	Get information about a specific vApp by using the vCloud ID and the vApp ID.
Get	/api/vcloud/{vcloudid}/vapp/{vappId}/vm	Get a list of VMs of a vApp by using the vCloud ID and the vApp ID.
Get	/api/vcloud/{vcloudid}/vdc/{vdcId}	Get information about a vDC of a vCloud by using the vCloud ID and vDC ID.

API	Endpoint	Description
Get	/api/vcloud/{vcloudid}/vdc/{vdcId}/vapp	Get a list of vApps in a vDC of a vCloud by using the vCloud ID and the vDC ID.
Get	/api/vcloud/{vcloudid}/vm/{vmId}	Get information about a VM of a vCloud using the vCloud ID and the VM ID.
Get	/api/vcloud/{vcloudid}/vm/{vmId}/disk	Get a list of disks for a VM of a vCloud by using the vCloud ID and the VM ID.
Get	/api/vcloud/{vcloudid}/vm/{vmId}/disk/{diskId}	Get information about a disk for a VM of a vCloud by using the vCloud ID, the VM ID, and the disk ID.
Post	/api/vcloud	Register a vCloud.
Put	/api/vcloud/{vcloudId}	Update a vCloud.
Delete	/api/vcloud/{vcloudId}	Unregister a vCloud.
Delete	/api/vcloud/{vcloudId}/vm/{vmId}	Remove a specific VM from the vCloud by using the vCloud ID and the VM ID.

vSphere

You can get information about VMware vSphere providers in IBM® Storage Defender Copy Data Management through the REST API.

Resource Base URL

```
http://<ipaddress>:<port>/api/vsphere
```

Related Resources

[“LDAP” on page 26](#)

[“NetApp ONTAP” on page 26](#)

[“vCloud” on page 28](#)

[“SMTP” on page 31](#)

Methods

The following RESTful API methods and endpoints are used to query, register, update, and delete VMware vSphere providers:

Table 22: VMware vSphere API		
API	Endpoint	Description
Get	/api/vsphere	Get a list of registered VMware vSphere providers registered in IBM® Storage Defender Copy Data Management.
Get	/api/vsphere/{vsphereId}	Get a registered VMware vSphere provider by using the vSphere ID.
Get	/api/vsphere/{vsphereId}/datacenter	Get datacenter by using the VMware vSphere ID.
Get	/api/vsphere/{vsphereId}/datacenter/{datacenterId}	Get the datacenter by using the VMware vSphere ID and the datacenter ID.

API	Endpoint	Description
Get	/api/vsphere/{vsphereId}/datacenter/{datacenterId}/content	Get folders on a datacenter by using the VMware vSphere ID and using the datacenter ID.
Get	/api/vsphere/{vsphereId}/datacenter/{datacenterId}/datastore	Get the datastores on a datacenter by using the VMware vSphere ID and using the datacenter ID.
Get	/api/vsphere/{vsphereId}/datacenter/{datacenterId}/vm	Get VMs on a datacenter by using the VMware vSphere ID and using the datacenter ID
Get	/api/vsphere/{vsphereId}/datastore	Get the datastores on a VMware vSphere by using the vSphere ID.
Get	/api/vsphere/{vsphereId}/datastore/{datastoreId}	Get a specific datastore on a VMware vSphere by using the vSphere ID and the datastore ID.
Get	/api/vsphere/{vsphereId}/datastore/{datastoreId}/host	Get datastore hosts on a VMware vSphere by using the vSphere ID and datastore ID.
Get	/api/vsphere/{vsphereId}/datacenter/{datacenterId}/host	Get datacenter hosts on a VMware vSphere by using the vSphere ID and the datacenter ID.
Get	/api/vsphere/{vsphereId}/datastore/{datastoreId}/vm	Get datastore VMs on a VMware vSphere by using the vSphere ID and the datastore ID.
Get	/api/vsphere/{vsphereId}/folder/{folderId}	Get a specific folder on a VMware vSphere by using the vSphere ID and the folder ID.
Get	/api/vsphere/{vsphereId}/folder/{folderId}/content	Get the contents of a specific folder on a VMware vSphere by using the vSphere ID and the folder ID.
Get	/api/vsphere/{vsphereId}/folder/{folderId}/datastore	Get the folder datastores on a VMware vSphere by using the vSphere ID and the folder ID.
Get	/api/vsphere/{vsphereId}/folder/{folderId}/vm	Get the folder VMs on a VMware vSphere by using the vSphere ID and the folder ID.
Get	/api/vsphere/{vsphereId}/host	Get a list of hosts on a VMware vSphere by using the vSphere ID.
Get	/api/vsphere/{vsphereId}/host/{hostId}	Get information on a specific host on a VMware vSphere by using the vSphere ID and the host ID.
Get	/api/vsphere/{vsphereId}/host/{hostId}/vm	Get a list of VMs for a specific host on a VMware vSphere by using the vSphere ID and the host ID.
Get	/api/vsphere/{vsphereId}/resourcepool/{resourcepoolId}/content	Get a specific resource pool on a VMware vSphere by using the vSphere ID and the resource pool ID.
Get	/api/vsphere/{vsphereId}/snapshot	Get a list of snapshots on a VMware vSphere by using the vSphere ID.

API	Endpoint	Description
Get	/api/vsphere/{vsphereId}/snapshot/{snapshotId}	Get information on a specific snapshot on a VMware vSphere by using the vSphere ID and the snapshot ID.
Get	/api/vsphere/{vsphereId}/vapp	Get a list of vApps on a VMware vSphere by using the vSphere ID.
Get	/api/vsphere/{vsphereId}/vapp/{vappId}	Get information on a specific vApp on a VMware vSphere by using the vSphere ID and the vApp ID.
Get	/api/vsphere/{vsphereId}/vapp/{vappId}/datastore	Get a list of VMware datastores by using the vSphere ID and the vApp ID.
Get	/api/vsphere/{vsphereId}/vapp/{vappId}/vm	Get a list of VMware VMs by using the vSphere ID and the vApp ID.
Get	/api/vsphere/{vsphereId}/vapp/{vappId}/vapp	Get a list of VMware vApps by using the vSphere ID and the vApp ID.
Get	/api/vsphere/{vsphereId}/vm	Get a list of VMs by using the vSphere ID.
Get	/api/vsphere/{vsphereId}/vm/{vmId}	Get information about a specific VMware VM by using the vSphere ID and VM ID.
Get	/api/vsphere/{vsphereId}/vm/{vmId}/snapshot	Get a list of VM snapshots by using the vSphere ID and VM ID.
Get	/api/vsphere/{vsphereId}/vm/{vmId}/datastore	Get a list of VM datastores by using the vSphere ID and VM ID.
Post	/api/vsphere	Register a VMware vSphere provider.
Post	/api/vsphere/{vsphereId}/vm/{vmId}	Create a snapshot of a VM using the VM ID on a specific vSphere using the vSphere ID.
Put	/api/vsphere/{vsphereId}	Edit a specific registered VMware vSphere by using a vSphere ID.
Delete	/api/vsphere/{vsphereId}	Unregister a VMware vSphere provider.

SMTP

You can get information about SMTP providers in IBM® Storage Defender Copy Data Management through the REST API.

Resource Base URL

```
http://<ipaddress>:<port>/api/smtp
```

Related Resources

[“LDAP” on page 26](#)

[“vCloud” on page 28](#)

[“vSphere” on page 29](#)

Methods

The following RESTful API methods and endpoints are used to add, view, edit, and delete SMTP providers:

Table 23: SMTP API		
API	Endpoint	Description
Get	/api/smtp	Get a list of all registered SMTP providers.
Get	/api/smtp/{smtpId}	Get a specific SMTP provider by using the SMTP ID.
Post	/api/smtp	Register an SMTP provider.
Put	/api/smtp/{smtpId}	Update a specific SMTP provider by using the SMTP ID.
Delete	/api/smtp/{smtpId}	Unregister an SMTP provider by using the SMTP ID.

Search Management

A set of RESTful APIs used for search management processes such as performing search operations.

Related information

[Search](#)

Search

You can get information about run searches by using the IBM® Storage Defender Copy Data Management REST API.

Resource Base URL

```
http://<ipaddress>:<port>/api/endeavour/search
```

Methods

The following RESTful API methods and endpoints are used to perform searches, retrieve document schema, and delete indexes:

Table 24: Search API		
API	Endpoint	Description
Get	/api/endeavour/search	Search catalogs in IBM® Storage Defender Copy Data Management.
Get	/api/endeavour/search/download/json	Download in JSON format.
Get	/api/endeavour/search/schema/all	Get schema for all searchable document types.
Get	/api/endeavour/search/schema/all/documentType/values	Get schema for the specific document type.
Post	/api/endeavour/search	Index the catalog.
Delete	/api/endeavour/search	Delete the catalog.

Snapshot Management

The snapshot management RESTful APIs allow users to manage asynchronous offload and restore operations for IBM Storage FlashSystem snapshots.

The offload and restore APIs enable you to offload snapshots from IBM Storage FlashSystem to secondary storage and restore them when needed. All operations are asynchronous and tracked through session management.

Functional Areas

The APIs provide the following capabilities:

- **Offload Operations:** Start asynchronous offload operations for snapshots from IBM Storage FlashSystem
- **Restore Operations:** Restore previously offloaded snapshots to a specified destination
- **Session Tracking:** Monitor the status and progress of offload and restore operations

Note: IBM® Storage Defender Copy Data Management supports only one offload or restore operation at a time. Only one offload or restore session can be in the IN_PROGRESS or PENDING state.

Resource Base URL

The service uses the following base context path:

```
https://host:8443/api
```

Authentication

All API requests must include the authentication header:

- `x-endeavour-sessionid`: Session identifier obtained from the Copy Data Management session API

This header is used by the Copy Data Management session filter for authentication and authorization context.

Methods

The following RESTful APIs methods and endpoints are available for snapshot offload and restore operations:

Table 25: Snapshot Management APIs		
Method	Endpoint	Description
POST	/api/offload	Start an offload operation
POST	/api/offload/restore	Start a restore operation
GET	/api/offload/sessions/{sessionId}	Get status of a specific session
GET	/api/offload/sessions	List sessions with optional filters

Asynchronous Processing

Both offload and restore operations are asynchronous. When you initiate an operation, the API immediately returns a session identifier. You can use this identifier to poll the session status and monitor progress. The actual offload or restore workflow runs in the background.

Session Lifecycle

Sessions progress through the following states:

- **PENDING:** Session created, operation queued
- **IN_PROGRESS:** Operation actively running
- **SUCCESSFUL:** Operation completed successfully
- **FAILED:** Operation failed with error

Offload Snapshot

Start an asynchronous offload operation for a snapshot from IBM Storage FlashSystem.

Purpose

This API starts an asynchronous offload operation for a snapshot. The API immediately creates a session and returns session details. The actual offload workflow runs asynchronously in the background.

Headers

The following headers are required:

Table 26: Required Headers	
Header	Value
Content-Type	application/json
x-endeavour-sessionid	Valid session identifier from Copy Data Management

Request Body

The request body must be in JSON format with the following fields:

Table 27: Request Parameters			
Field	Type	Required	Description
storageId	integer	Yes	Storage configuration identifier (IBM Storage FlashSystem ID registered in Copy Data Management)
snapshotId	integer	Yes	Snapshot identifier to offload
name	string	Yes	User-provided unique name for the offload session. Duplicate names are rejected.

Example Request Body:

```
{
  "storageId": 1001,
  "snapshotId": 120,
  "name": "nightly-vol1001-snap120"
}
```

Response

Success Response (HTTP 200):

```
{
  "sessionId": "SESS-1",
  "type": "offload",
  "status": "PENDING",
  "name": "nightly-vol1001-snap120",
  "startTime": "2024-06-01T10:00:00Z",
  "endTime": null,
  "error": null
}
```

Response Parameters:

Table 28: Response Parameters		
Field	Type	Description
sessionId	string	Unique session identifier for tracking the operation
type	string	Session type (offload)
status	string	Current status: PENDING, IN_PROGRESS, SUCCESSFUL, or FAILED
name	string	Name of the offload session

Field	Type	Description
startTime	string	ISO-8601 timestamp when the session was created
endTime	string/null	ISO-8601 timestamp when the session completed or failed
error	string/null	Error message if the session failed

Behavior

The API performs the following operations:

1. Validates storageId, snapshotId, and name
2. Fetches storage configuration using storageId
3. Creates a session in PENDING state
4. Starts asynchronous offload processing
5. Returns session details immediately

The session later transitions to IN_PROGRESS, then to either SUCCESSFUL or FAILED.

Error Responses

HTTP 400 - Validation Error:

```
{
  "code": "VALIDATION_ERROR",
  "message": "Missing required field 'storageId' in request body"
}
```

Possible validation messages:

- Missing required field 'storageId' in request body
- Missing required field 'snapshotId' in request body
- Missing or empty 'name' in request body
- storageId and snapshotId must be valid integers

HTTP 409 - Duplicate Name:

```
{
  "code": "DUPLICATE_NAME",
  "message": "An offload with the given name already exists"
}
```

HTTP 500 - Internal Error:

```
{
  "code": "INTERNAL_ERROR",
  "message": "An unexpected error occurred. Please retry or contact support."
}
```

Example

The following example demonstrates how to start an offload operation using curl:

```
curl --request POST 'https://ip_address:8443/api/offload' \
  --header 'Content-Type: application/json' \
  --header 'x-endeavour-sessionid: session-id-from-cdm' \
  --data '{
    "storageId": 1001,
    "snapshotId": 120,
    "name": "nightly-vol1001-snap120"
  }'
```

Restore Snapshot

Start an asynchronous restore operation using a previously offloaded snapshot.

Purpose

This API starts an asynchronous restore operation using a previously offloaded snapshot name. The service resolves the corresponding offload metadata internally and starts the restore workflow in the background.

Headers

The following headers are required:

Table 29: Required Headers	
Header	Value
Content-Type	application/json
x-endeavour-sessionid	Valid session identifier from Copy Data Management

Request Body

The request body must be in JSON format with the following fields:

Table 30: Request Parameters			
Field	Type	Required	Description
name	string	Yes	Name of a previously offloaded snapshot/session
destinationPath	string	Yes	Destination path where files will be restored

Example Request Body:

```
{
  "name": "nightly-vol1001-snap120",
  "destinationPath": "/restore_target/"
}
```

Response

Success Response (HTTP 200):

```
{
  "sessionId": "7c94af21-3310-4812-a1bd-9f772d55bca3",
  "type": "restore",
  "status": "PENDING",
  "name": "nightly-vol1001-snap120",
  "startTime": "2024-06-01T14:00:00Z",
  "endTime": null,
  "error": null
}
```

Response Parameters:

Table 31: Response Parameters		
Field	Type	Description
sessionId	string	Unique session identifier for tracking the operation
type	string	Session type (restore)
status	string	Current status: PENDING, IN_PROGRESS, SUCCESSFUL, or FAILED
name	string	Name of the restore session
startTime	string	ISO-8601 timestamp when the session was created
endTime	string/null	ISO-8601 timestamp when the session completed or failed

Field	Type	Description
error	string/null	Error message if the session failed

Behavior

The API performs the following operations:

1. Validates name and destinationPath
2. Looks up the previous offload by name
3. Creates a restore session
4. Starts asynchronous restore processing
5. Returns session details immediately

The session later transitions to IN_PROGRESS, then to either SUCCESSFUL or FAILED.

Error Responses

HTTP 400 - Validation Error:

```
{
  "code": "VALIDATION_ERROR",
  "message": "Missing or empty 'destinationPath' in request body"
}
```

Possible validation messages:

- Missing or empty 'name' in request body
- Missing or empty 'destinationPath' in request body

HTTP 404 - Not Found:

```
{
  "code": "NOT_FOUND",
  "message": "No matching offload found for the provided name"
}
```

HTTP 500 - Internal Error:

```
{
  "code": "INTERNAL_ERROR",
  "message": "An unexpected error occurred. Please retry or contact support."
}
```

Example

The following example demonstrates how to start a restore operation:

```
curl --request POST 'https://host:8443/api/offload/restore' \
  --header 'Content-Type: application/json' \
  --header 'x-endeavour-sessionid: session-id-from-cdm' \
  --data '{
    "name": "nightly-vol1001-snap120",
    "destinationPath": "/restore_target/"
  }'
```

List Sessions

Retrieve paginated session data with optional filters for offload and restore operations.

Purpose

This API returns paginated session data with optional filters. Use this API to list and search for offload and restore sessions based on various criteria.

Headers

The following header is required:

Table 32: Required Headers	
Header	Value
x-endeavour-sessionid	Valid session identifier from Copy Data Management

Request Parameters

Table 33: Request Parameters				
Parameter	Type	Required	Default	Description
limit	integer	No	20	Number of records to return. Allowed range: 1 to 100
offset	integer	No	0	Number of records to skip. Must be 0 or greater
name	string	No	-	Exact-match filter on session name
status	string	No	-	Filter by status: PENDING, IN_PROGRESS, SUCCESSFUL, or FAILED
startTimeAfter	string	No	-	ISO-8601 timestamp filter for sessions starting after this time
startTimeBefore	string	No	-	ISO-8601 timestamp filter for sessions starting before this time

Validation Rules

- limit must be between 1 and 100
- offset must be non-negative
- status must be one of: PENDING, IN_PROGRESS, SUCCESSFUL, FAILED
- startTimeAfter and startTimeBefore must be valid ISO-8601 timestamps
- startTimeAfter must not be later than startTimeBefore

Response

Success Response (HTTP 200):

```
{
  "items": [
    {
      "sessionId": "SESS-1001",
      "type": "offload",
      "status": "SUCCESSFUL",
      "name": "nightly-vol1001-snap120",
      "startTime": "2024-06-01T10:00:00Z",
      "endTime": "2024-06-01T10:12:45Z",
      "error": null
    },
    {
      "sessionId": "SESS-1002",
      "type": "restore",
      "status": "IN_PROGRESS",
      "name": "nightly-vol1001-snap120",
      "startTime": "2024-06-01T14:00:00Z",
      "endTime": null,
      "error": null
    }
  ],
  "total": 2,
  "limit": 20,
  "offset": 0
}
```

Response Parameters:

Table 34: Response Parameters		
Field	Type	Description
items	array	Array of session objects matching the query criteria
total	integer	Total number of sessions matching the filter criteria
limit	integer	Number of records returned in this response
offset	integer	Number of records skipped

Error Responses

HTTP 400 - Bad Request:

```
{
  "code": "BAD_REQUEST",
  "message": "Limit must be between 1 and 100"
}
```

Other possible error messages:

- Offset must be non-negative
- Invalid status. Valid values are: PENDING, IN_PROGRESS, SUCCESSFUL, FAILED
- Invalid date format. Use ISO 8601 format (e.g., 2024-06-01T10:00:00Z)
- startTimeAfter must be before startTimeBefore

Example

```
curl --request GET 'http://<host>:8080/api/offload/sessions?
limit=20&offset=0&status=SUCCESSFUL' \
--header 'x-endeavour-sessionid: <session-id-from-cdm>'
```

Get Session Status

Retrieve the current status and details of a specific offload or restore session.

Purpose

This API returns the current status and details of a specific offload or restore session. Use this API to monitor the progress of asynchronous operations.

Headers

The following header is required:

Table 35: Required Headers	
Header	Value
x-endeavour-sessionid	Valid session identifier from Copy Data Management

Parameters

Table 36: Request Parameters			
Parameter	Type	Required	Description
sessionId	string	Yes	Session identifier returned by the offload or restore API

Response

Success Response (HTTP 200):

```
{
  "sessionId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "type": "offload",
  "status": "IN_PROGRESS",
}
```



```

"name": "nightly-vol1001-snap120",
"startTime": "2024-06-01T10:00:00Z",
"endTime": null,
"error": null
}

```

Response Parameters:

Table 37: Response Parameters		
Field	Type	Description
sessionId	string	Unique session identifier
type	string	Session type: offload or restore
status	string	Current status: PENDING, IN_PROGRESS, SUCCESSFUL, or FAILED
name	string	Offload or restore name associated with the session
startTime	string	ISO-8601 timestamp when the session was created
endTime	string/null	ISO-8601 timestamp when the session completed or failed
error	string/null	Error message if the session failed

Error Responses

HTTP 404 - Session Not Found:

```

{
  "code": "NOT_FOUND",
  "message": "Session 3fa85f64-5717-4562-b3fc-2c963f66afa6 not found"
}

```

Example

The following example demonstrates how to get session status using cURL:

```

curl --request GET 'https://ip_address:8443/api/offload/sessions/3fa85f64-5717-4562-b3fc-2c963f66afa6' \
  --header 'x-endeavour-sessionid: session-id-from-cdm'

```

Examples

The examples provide instructions to implement various operations including getting a session ID; executing a search; registering a resource provider; creating a policy; creating, starting, and monitoring a catalog job; and creating Snapshot and recovery policies.

Related information

[Getting a Session ID](#)

[Execute a Search](#)

[Register a Resource Provider](#)

[Create a Policy](#)

[Create a Catalog Job](#)

[Start a Catalog Job](#)

[Monitor a Catalog Job](#)

[VMware-NetApp Snapshot Policy](#)

[Start a Catalog Job](#)

[Offload and Restore Snapshot Workflow](#)

VMware-NetApp Snapshot Policy

This example demonstrates how to programmatically create a VMware-NetApp Snapshot policy.

The RESTful APIs in this scenario are:

Snapshot Endpoint

Table 38: Backup Endpoint		
API	Endpoint	Description
Post	/api/endeavour/policy	Create a policy.

Snapshot Parameters

Table 39: Backup Parameters		
Name	Description	Value
name	Name of the policy.	string, max 128, non-empty
type	SNAPSHOT	string, max 256
subType	VMWARE-NETAPP	string, max 256
description	Description of the policy.	string, max 64
spec	This is an object.	string, max 32
href	URL for the source selection.	string
resourceType	Source type. For example: vm, folder, app, datacenter, and vsphere.	string
name	Name of the source.	string

Snapshot Request

```

{
  "name": "SnapshotPolicy1",
  "type": "SNAPSHOT",
  "subType": "VMWARE-NETAPP",
  "description": "",
  "spec": {
    "source": [
      {
        "href": "http://localhost:8082/api/vsphere/1001/vm/42172c8a-6404-14d6-d092-ec511ffb3553",
        "resourceType": "vm",
        "name": "11 SnapTestVM-GXC",
        "metadata": {
          "path": "vsphere:1001/datacenter:datacenter-31960",
          "registration": "http://localhost:8082/api/vsphere/1001",
          "id": "42172c8a-6404-14d6-d092-ec511ffb3553"
        }
      },
      {
        "href": "http://localhost:8082/api/vsphere/1001/vm/4217b62c-fb76-c08d-7a70-4a90c467ab24",
        "resourceType": "vm",
        "name": "12 SnapTestVM2-GXC",
        "metadata": {
          "path": "vsphere:1001/datacenter:datacenter-31960",
          "registration": "http://localhost:8082/api/vsphere/1001",
          "id": "4217b62c-fb76-c08d-7a70-4a90c467ab24"
        }
      }
    ],
    "option": {
      "proxyName": "",
      "proxyPort": "",
      "retention": "1",
      "excludeControllers": [ ]
    },
    "notification": [ ]
  }
}

```

Snapshot Response

```

{
  "links": {
    "self": {
      "rel": "self",
      "href": "http://71.100.15.107:8082/api/endeavour/policy/1003"
    },
    "up": {
      "rel": "up",
      "href": "http://71.100.15.107:8082/api/endeavour/policy"
    },
    "edit": {
      "rel": "update",
      "href": "http://71.100.15.107:8082/api/endeavour/policy/1003"
    },
    "delete": {
      "rel": "delete",
      "href": "http://71.100.15.107:8082/api/endeavour/policy/1003"
    },
    "jobs": {
      "rel": "related",
      "href": "http://71.100.15.107:8082/api/endeavour/job?filter=[{"property": "policyId", "value": "1003"}]"
    }
  },
  "name": "SnapshotPolicy1",
  "type": "SNAPSHOT",
  "subType": "VMWARE-NETAPP",
  "description": "",
  "version": null,
  "spec": {
    "source": [
      {
        "href": "http://localhost:8082/api/vsphere/1001/vm/42172c8a-6404-14d6-d092-ec511ffb3553",
        "resourceType": "vm",
        "name": "11 SnapTestVM-GXC",
        "metadata": {
          "path": "vsphere:1001/datacenter:datacenter-31960",
          "registration": "http://localhost:8082/api/vsphere/1001",
          "id": "42172c8a-6404-14d6-d092-ec511ffb3553"
        }
      },
      {
        "href": "http://localhost:8082/api/vsphere/1001/vm/4217b62c-fb76-c08d-7a70-4a90c467ab24",
        "resourceType": "vm",
        "name": "12 SnapTestVM2-GXC",
        "metadata": {
          "path": "vsphere:1001/datacenter:datacenter-31960",
          "registration": "http://localhost:8082/api/vsphere/1001",
          "id": "4217b62c-fb76-c08d-7a70-4a90c467ab24"
        }
      }
    ],
    "option": {
      "proxyName": "",
      "proxyPort": "",
      "retention": "1",
      "excludeControllers": [ ]
    },
    "notification": [ ]
  },
  "id": "1003"
}

```

The following is an example of the RESTful API parameters and the steps you can take to traverse the vSphere folder structure to determine the link to a VM resource to backup. The vSphere folder structures look like
vCenter> Datacenter > folder1 > folder2 > vm1:

```

# list of registered vCenters
"/api/vsphere"

# get datacenters in vCenter

```

```
"/api/vsphere/1003/folder/group-d1/content"

# get folders in particular datacenter
"/api/vsphere/1003/folder/group-v3/content"

# get folders within folder1
"/api/vsphere/1003/folder/group-v66556/content"

# get the list of VMs in folder2
"/api/vsphere/1003/folder/group-v73435/content"
```

Create a Catalog Job

This example demonstrates how to programmatically create jobs.

The RESTful APIs in this scenario are:

Job Create Endpoint

Table 40: Job Create Endpoint		
API	Endpoint	Description
Post	/api/endeavour/job	Create a job.

Job Create Parameters

Table 41: Job Create Parameters		
Name	Description	Value
name	Name of the job.	string, max 128, non-empty
triggerIds	Array of existing valid trigger ID strings.	string array
policyId	Existing, valid policy ID string.	string, max 256, non-empty
description	Description of the job.	string, max 64

Snapshot Request

```
POST
URL: https://<ipaddress>:port/api/endeavour/job
Request Body:
{
  "name": "test policy",
  "description": "test description",
  "policyId": "1009",
  "triggerIds": [
  ]
}
```

Snapshot Response

```

{
  "links":{
    "self":{
      "rel":"self",
      "href":"https://<ip>:<port>/api/endeavour/job/1009"
    },
    "up":{
      "rel":"up",
      "href":"https://<ip>:<port>/api/endeavour/job"
    },
    "edit":{
      "rel":"update",
      "href":"https://<ip>:<port>/api/endeavour/job/1009"
    },
    "delete":{
      "rel":"delete",
      "href":"https://<ip>:<port>/api/endeavour/job/1009"
    },
    "policy":{
      "rel":"related",
      "href":"https://<ip>:<port>/api/endeavour/policy/1009"
    },
    "triggers":{
      "rel":"related",
      "href":"https://<ip>:<port>/api/endeavour/job/1009/trigger"
    },
    "jobsessions":{
      "rel":"related",
      "href":"https://<ip>:<port>/api/endeavour/jobsession?filter=[{"property":"jobId","value":"1009"}]"
    },
    "stats":{
      "rel":"related",
      "href":"https://<ip>:<port>/api/endeavour/jobsession/stats/jobid/1009"
    },
    "firetimes":{
      "rel":"related",
      "href":"https://<ip>:<port>/api/endeavour/firetime?filter=[{"property":"jobId","value":"1009"}]"
    },
    "start":{
      "rel":"action",
      "href":"https://<ip>:<port>/api/endeavour/job/1009?action=start"
    }
  },
  "name":"JayTestPolicy2",
  "description":"test description",
  "policyId":"1009",
  "policyName":"test policy",
  "type":"CATALOG",
  "subType":"NETAPP",
  "triggerIds":[
  ],
  "lastRunTime":null,
  "nextFireTime":null,
  "id":"1009"
}

```

Notice that the job ID that is created is the same value as the policy ID. Now that this job is created, you can perform actions on it such as start; as the start named link in the response suggest.

Create a Policy

This example demonstrates how to programmatically configure in preparation for performing cataloging of metadata that is managed by the resource provider.

The RESTful APIs in this scenario are:

Policy Endpoint

Table 42: Policy Endpoint		
API	Endpoint	Description
Post	/api/endeavour/policy	Create a policy.

Policy Parameters

Table 43: Policy Parameters		
Name	Description	Value
name	Name of the policy.	string, max 128, non-empty
type	Catalog, report, etc.	string, max 256
subType	NetApp ONTAP, etc.	string, max 256
description	Description of the policy.	string, max 64
spec	This is an object.	string, max 32

Policy Creation Request

This example request reveals a policy is being created for a NetApp ONTAP type cataloging operation. The spec section indicates the particular resource provider and its associated resources that are expected to be cataloged. Other objects that can be defined within the spec section include option and notification. Once the Post is executed, the response includes an ID to the new policy resource.

```
POST
URL: https://<ipaddress>:port/api/endeavour/policy

Request Body:
{
  "name": "test policy",
  "type": "CATALOG",
  "subType": "NETAPP",
  "description": "test description",
  "spec": {
    "source": [
      {
        "href": "https://<ip>:<port>/api/netapp/1002",
        "resourceType": "netapp",
        "name": "<ip>",
        "id": "1002",
        "tags": {

        }
      }
    ]
  },
  "option": {
    "timeout": "300",
    "retention": "3"
  },
  "notification": [

  ]
}
```

Policy Creation Response

```

{
  "links":{
    "self":{
      "rel":"self",
      "href":"https://<ip>:<port>/api/endeavour/policy/1009"
    },
    "up":{
      "rel":"up",
      "href":"https://<ip>:<port>/api/endeavour/policy"
    },
    "edit":{
      "rel":"update",
      "href":"https://<ip>:<port>/api/endeavour/policy/1009"
    },
    "delete":{
      "rel":"delete",
      "href":"https://<ip>:<port>/api/endeavour/policy/1009"
    },
    "jobs":{
      "rel":"related",
      "href":"https://<ip>:<port>/api/endeavour/job?filter=[{\"property\":\"policyId\",\"value\":\"1009\"}]"
    }
  },
  "name":"test policy",
  "type":"CATALOG",
  "subType":"NETAPP",
  "description":"test description",
  "version":null,
  "spec":{
    "source":[
      {
        "href":"https://<ip>:<port>/api/netapp/1002",
        "resourceType":"netapp",
        "name":"<ip>",
        "id":"1002",
        "tags":{
          }
        }
      ]
    },
    "option":{
      "timeout":"300",
      "retention":"3"
    },
    "notification":[
      ]
  },
  "id":"1009"
}

```

The policy ID within this response can then be used by the job API to launch a catalog operation.

Execute a Search

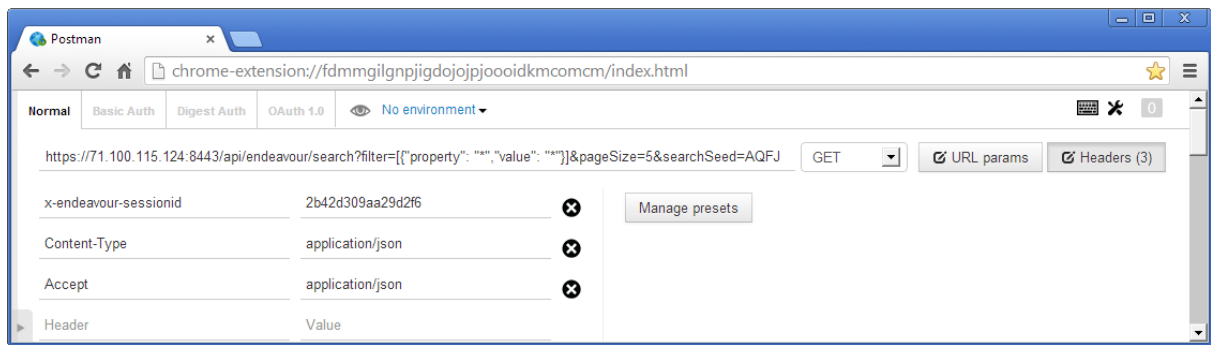
This example demonstrates how to programmatically execute a search.

API Headers

Prior to calling search, set the following headers:

- X-endeavour-sessionid with a valid jobsessionId value.
- Content-Type with the value set to application/json.
- Accept with the value set to application/json.

The following example shows how this value is set within Google's Postman REST API client.



Search Endpoint

Table 44: Search Endpoint		
API	Endpoint	Description
Get	/api/endeavour/search?filter=[{"property": "*", "op": "=", "value": "*abc*"}, {...}]	Execute a search using a filter.

Search Filter Parameters

Search relies on a filter parameter against its endpoint to carry out search operations. Following are the parameters and values to submit to the search filter:

Table 45: Search Filter Parameters		
Name	Description	Value
property	The property parameter is used to indicate which indexed field of the collection of fields to apply the criteria to. An asterisk (*) for the value of the property parameter enables all fields to be searched.	catalogTime, location, documentType, providerName, *
value	This constitutes the search pattern.	Wildcard is supported, such as "?abc", "abc?", "*abc", "abc*". Wildcard characters such as the question mark (?) is for one character, and the asterisk (*) is for zero or more characters. Multiple filters can be combined with the AND operator.

With the session id set as the value of the X-endeavour-sessionid header the following request can be submitted as a GET operation against the search endpoint as follows:

Search Request

```
GET
URL and Request:
https://<ipaddress>:port/api/endeavour/search?filter=[{
  "property": "*",
  "value": "*JWFndmp"
}]
&pageSize=5&searchSeed=AQFJ
```

Search Response


```
{
  "links":{
    "self":{
      "rel":"self",
      "href":"https://<ip>:<port>/api/endeavour/search"
    },
    "firstPage":{
      "rel":"first",
      "href":"https://<ip>:<port>/api/endeavour/search?pageSize=5&filter=[{"property\\": \"*\\",\"value\\": \"*\\"}]"
    },
    "nextPage":{
      "rel":"next",
      "href":"https://<ip>:<port>/api/endeavour/search?pageSize=5&searchSeed=AgFJAU4&filter=[{"property\\": \"*\\",\"value\\": \"*\\"}]"
    },
    "prevPage":{
      "rel":"prev",
      "href":"https://<ip>:<port>/api/endeavour/search?pageSize=5&searchSeed=AA&filter=[{"property\\": \"*\\",\"value\\": \"*\\"}]"
    },
    "csv":{
      "rel":"download",
      "href":"https://<ip>:<port>/api/endeavour/search/download/csv?pageSize=5&filter=[{"property\\": \"*\\",\"value\\": \"*\\"}]&sessionId=2b42d309aa29d2f6"
    }
  },
  "indexSize":9380247,
  "hits":9380247,
  "searchTime":3254,
  "page":2,
  "results":[
    {
      "links":{
        "self":{
          "rel":"self",
          "href":"https://<ip>:<port>/api/endeavour/catalog/netapp/File/52654066e4b0d1d42fc36ce1_1382367131084"
        }
      },
      "score":1,
      "summary":{
        "createTime":1377891916,
        "catalogTime":1382367131084,
        "documentType":"File",
        "accessTime":1382367168,
        "location":"f2240clst/vs_data_01/",
        "name":"/",
        "catalogProvider":"netapp",
        "modifyTime":1351876116,
        "size":11744
      }
    }
  ]
}
```

Getting a Session ID

This example demonstrates how to obtain a session ID. This example assumes that users are configured within the system. By default, there is a predefined user. Consult the documentation for user credential specifics.

The RESTful APIs in this scenario are:

Session Endpoint

Table 46: Session Endpoint		
API	Endpoint	Description
Post	/api/endeavour/session	Create a session.

Endeavour uses HTTP's basic authentication mechanism where the user name and password are sent in the Authorization header. The user name and password must be Base64 encoded. When a POST operation to the endpoint above is performed, which includes a valid Authorization header, the corresponding response contains a session id, which can then be used for subsequent RESTful API calls.

Session Request

The following example provides the steps for establishing the session id using Google Chrome Postman REST client.

1. In the Basic Auth tab, enter the username and password.

2. Click the Refresh headers button. A base64-encoded string is produced.
3. With the Authorization header set, you can now send, which executes a Post operation against the `https://{hostname|ipv4}/endeavour/session` endpoint. The following response, which includes the session ID value, is returned.

Session Response

The response contains the session ID as well as the information for the logged in user. This information is the same as that returned by a GET call against the following URI `/api/endeavour/user/{userID}`.

All further requests must include the session ID in the custom header `X-endeavour-sessionid`.

Monitor a Catalog Job

This example demonstrates how to programmatically monitor jobs.

The RESTful APIs in this scenario are:

Job Monitor Endpoints

Table 47: Job Monitor Endpoints		
API	Endpoint	Description
Get	<code>/api/endeavour/job</code>	Get the job session. Every job embeds the last run time, next firetime, and a snapshot of the last run session.
Get	<code>/api/endeavour/jobsession/{jobsessionId}?embed=(tasks)</code>	Get job sessions. Lists a specific job session with options to list all tasks.

Job Monitor Parameters

Table 48: Job Monitor Parameters		
Name	Description	Value
sessionId	String ID of the job session (instance).	string

Job Monitor Request via Job Endpoint

`https://{hostname|IPv4}/api/endeavour/job`

Job Monitor Response

```

{
  "links":{
    "self":{
      "rel":"self",
      "href":"http://<ip>:<port>/api/endeavour/job"
    },
    "create":{
      "rel":"create",
      "href":"http://<ip>:<port>/api/endeavour/job"
    }
  },
  "jobs":[
    {
      "links":{
        "self":{
          "rel":"self",
          "href":"http://<ip>:<port>/api/endeavour/job/1002"
        },
        "up":{
          "rel":"up",
          "href":"http://<ip>:<port>/api/endeavour/job"
        },
        "edit":{
          "rel":"update",
          "href":"http://<ip>:<port>/api/endeavour/job/1002"
        },
        "delete":{
          "rel":"delete",
          "href":"http://<ip>:<port>/api/endeavour/job/1002"
        },
        "policy":{
          "rel":"related",
          "href":"http://<ip>:<port>/api/endeavour/policy/1002"
        },
        "triggers":{
          "rel":"related",
          "href":"http://<ip>:<port>/api/endeavour/job/1002/trigger"
        },
        "jobsessions":{
          "rel":"related",
          "href":"http://<ip>:<port>/api/endeavour/jobsession?filter=[{"property":"jobId","value":"1002"}]"
        },
        "stats":{
          "rel":"related",
          "href":"http://<ip>:<port>/api/endeavour/jobsession/stats/jobid/1002"
        },
        "firetimes":{
          "rel":"related",
          "href":"http://<ip>:<port>/api/endeavour/firetime?filter=[{"property":"jobId","value":"1002"}]"
        },
        "lastrun":{
          "rel":"related",
          "href":"http://<ip>:<port>/api/endeavour/jobsession/1381264793238"
        },
        "start":{
          "rel":"action",
          "href":"http://<ip>:<port>/api/endeavour/job/1002?action=start"
        }
      },
      "name":"vmjob",
      "description":"Auto-generated job for Policy vmjob",
      "policyId":"1002",
      "policyName":"vmjob",
      "type":"CATALOG",
      "subType":"VMWARE",
      "triggerIds":[
        ],
      "lastRunTime":1381264843226,
      "nextFireTime":null,
      "lastrun":{
        "sessionId":"1381264793238",
        "jobName":"vmjob",
        "type":"CATALOG",
        "subType":"VMWARE",
        "start":1381264843226,
        "end":1381264853767,
        "status":"STOPPED",
        "results":"Completed",
        "properties":null,
        "numTasks":2,
        "lastUpdate":1381264867422,
        "percent":null,
        "policySnapshot":{

```

```

    "id": "1002",
    "name": "vmjob",
    "type": "CATALOG",
    "subType": "JSON",
    "version": null,
    "spec": {
      "source": [
        {
          "href": "http://<ip>:<port>/api/vsphere/1001",
          "resourceType": "vsphere",
          "name": "vmcenter",
          "id": "1001",
          "tags": {

          }
        }
      ]
    },
    "option": {
      "timeout": "300",
      "retention": "3"
    },
    "notification": [

    ]
  }
}

},
{id": "1002"
}
}

```

Job Monitor Request via Jobsession Endpoint

[https://{hostname|IPv4}/api/endeavour/jobsession/{sessionId}?embed=\(tasks\)](https://{hostname|IPv4}/api/endeavour/jobsession/{sessionId}?embed=(tasks))

Job Monitor Response via Jobsession Endpoint

```
{
  "links":{
    "self":{
      "rel":"self",
      "href":"http://<ip>:<port>/api/endeavour/jobsession/1381264793238"
    },
    "up":{
      "rel":"up",
      "href":"http://<ip>:<port>/api/endeavour/jobsession"
    },
    "job":{
      "rel":"related",
      "href":"http://<ip>:<port>/api/endeavour/job/1002"
    },
    "tasks":{
      "rel":"related",
      "href":"http://<ip>:<port>/api/endeavour/jobsession/1381264793238/task"
    },
    "log":{
      "rel":"related",
      "href":"http://<ip>:<port>/api/endeavour/log/job?filter=[{\"property\":\"jobsessionId\",\"value\":\"1381264793238\"}]"
    },
    "csv":{
      "rel":"related",
      "href":"http://<ip>:<port>/api/endeavour/log/job/download/csv?filter=[{\"property\":\"jobsessionId\",\"value\":\"1381264793238\"}]&sessionId=a695b7854783d5eb"
    }
  },
  "jobName":"vmjob",
  "type":"CATALOG",
  "subType":"VMWARE",
  "start":1381264843226,
  "end":1381264853767,
  "status":"STOPPED",
  "indexStatus":"COMPLETED",
  "results":"Completed",
  "properties":null,
  "numTasks":2,
  "lastUpdate":1381264867422,
  "percent":null,
  "policySnapshot":{
    "id":"1002",
    "name":"vmjob",
    "type":"CATALOG",
    "subType":"VMWARE",
    "version":null,
    "spec":{
      "source":[
        {
          "href":"http://<ip>:<port>/api/vsphere/1001",
          "resourceType":"vsphere",
          "name":"vmcenter",
          "id":"1001",
          "tags":{
            "links":{
              "self":{
                "rel":"self",
                "href":"http://<ip>:<port>/api/endeavour/jobsession/1381264793238/task/1?embed=(tasks)"
              },
              "up":{
                "rel":"up",
                "href":"http://<ip>:<port>/api/endeavour/jobsession/1381264793238"
              }
            }
          }
        }
      ]
    }
  },
  "option":{
    "timeout":"300",
    "retention":"3"
  },
  "notification":[]
},
  "tasks":[
    {
      "links":{
        "self":{
          "rel":"self",
          "href":"http://<ip>:<port>/api/endeavour/jobsession/1381264793238/task/1?embed=(tasks)"
        },
        "up":{
          "rel":"up",
          "href":"http://<ip>:<port>/api/endeavour/jobsession/1381264793238"
        }
      }
    }
  ]
}
```

```

    "sessionId": "1381264793238",
    "type": "Resolve",
    "status": "STOPPED",
    "start": 1381264843319,
    "end": 1381264844055,
    "message": "",
    "properties": null,
    "results": "Stopped",
    "lastUpdate": 1381264844055,
    "percent": 0,
    "id": "1"
  },
  {
    "links": {
      "self": {
        "rel": "self",
        "href": "http://<ip>:<port>/api/endeavour/jobsession/1381264793238/task/2?embed=(tasks)"
      },
      "up": {
        "rel": "up",
        "href": "http://<ip>:<port>/api/endeavour/jobsession/1381264793238"
      }
    },
    "sessionId": "1381264793238",
    "type": "Catalog (VCenter)",
    "status": "STOPPED",
    "start": 1381264845318,
    "end": 1381264853632,
    "message": "",
    "properties": {
      "server": "vmcenter"
    },
    "results": "Stopped",
    "lastUpdate": 1381264853632,
    "percent": 0,
    "id": "2"
  }
],
"id": "1381264793238"
}

```

Register a Resource Provider

This example demonstrates how to programmatically configure in preparation for performing various operations such as search, reporting, cataloging, etc. Configuration starts with the registration of resource providers.

The following describes the RESTful API usage for registration of NetApp ONTAP, SMTP, and vSphere resource providers:

Resource Provider Registration Endpoint

Table 49: Resource Provider Registration Endpoint		
API	Endpoint	Description
Post	/api/netapp	Register NetApp ONTAP on IBM® Storage Defender Copy Data Management.
Post	/api/smtp	Register an SMTP in IBM® Storage Defender Copy Data Management.
Post	/api/vsphere	Register a vSphere in IBM® Storage Defender Copy Data Management.

Resource Provider Registration Parameters

Request parameters for registering NetApp ONTAP:

Table 50: Resource Provider Registration Parameters for NetApp ONTAP		
Name	Description	Value
name	Name of the registration.	string, max 128, non-empty
hostAddress	IP address or hostname.	string, max 256

Name	Description	Value
comment	Comment, optional.	string, max 256
username	Username of NetApp ONTAP account.	string, max 64
password	Password of NetApp ONTAP account.	string, max 32
sslConnection	Use secure connection or do not use secure connection.	true or false
portNumber	Port number.	integer

Request parameters for registering SMTP:

<i>Table 51: Resource Provider Registration Parameters for SMTP</i>		
Name	Description	Value
name	Name of the registration.	string, max 128, non-empty
hostAddress	IP address or hostname.	string, max 256
comment	Comment, optional.	string, max 256
username	Username of SMTP account.	string, max 64
password	Password of SMTP account.	string, max 32
portNumber	Port number.	integer (default set to 26)
fromAddress	E-mail from address.	string, max 254
subjectPrefix	E-mail subject prefix.	string, max 78
Subject	E-mail subject.	string, max 256
timeout	Timeout in connecting to SMTP server.	integer in milliseconds

Note: Request parameters for registering SMTP. SMTP is used only for e-mail notifications.

Request parameters for registering vSphere:

<i>Table 52: Resource Provider Registration Parameters for vSphere</i>		
Name	Description	Value
name	Name of the registration.	string, max 128, non-empty
hostAddress	IP address or hostname.	string, max 256
comment	Comment, optional.	string, max 256
username	Username of vSphere account.	string, max 64
password	Password of vSphere account.	string, max 32
sslConnection	Use secure connection or do not use secure connection.	true or false
portNumber	Port number.	integer

Resource Provider Registration Request

POST
URL: <https://<ipaddress>:port/api/netapp>

Request Body:

```
{
  "name": "r200",
  "hostAddress": "<ip>",
  "comment": "",
  "username": "root",
  "password": "password",
  "sslConnection": true,
  "portNumber": 443
}
```

Resource Provider Registration Response

```
{
  "links": {
    "self": {
      "rel": "self",
      "href": "http://<ip>:<port>/api/netapp/1001"
    },
    "up": {
      "rel": "up",
      "href": "http://<ip>:<port>/api/netapp"
    },
    "edit": {
      "rel": "update",
      "href": "http://<ip>:<port>/api/netapp/1001"
    },
    "delete": {
      "rel": "delete",
      "href": "http://<ip>:<port>/api/netapp/1001"
    },
    "volumes": {
      "rel": "related",
      "href": "http://<ip>:<port>/api/netapp/1001/volume"
    }
  },
  "name": "r200",
  "hostAddress": "<ip>",
  "type": "Storage Controller",
  "comment": "",
  "username": "root",
  "password": null,
  "sslConnection": false,
  "portNumber": 80,
  "properties": {
    "sysinfoSerialno": {
      "key": "Serial No",
      "value": "1041727"
    },
    "sysinfoNumberOfprocessors": {
      "key": "Number of Processors",
      "value": "2"
    },
    "sysinfoModel": {
      "key": "Model",
      "value": "R200"
    },
    "sysinfoOsversion": {
      "key": "OS Version",
      "value": "NetApp Release 7.3.7: Thu May 3 03:56:11 PDT 2012"
    },
    "sysinfoApiVersion": {
      "key": "API Version",
      "value": "1.14"
    },
    "sysinfoSystemid": {
      "key": "System ID",
      "value": "0050405016"
    },
    "sysinfoMemory": {
      "key": "Memory",
      "value": "6442450944"
    }
  },
  "resourceType": "netapp",
  "id": "1001"
}
```

After a resource provider is registered, it can be added to a policy. For more information, see [“Create a Policy” on page 45](#).

VMware-NetApp Recovery Policy

This example demonstrates how to programmatically create a VMware-NetApp Recovery policy.

The RESTful APIs in this scenario are:

Recovery Endpoint

Table 53: Restore Endpoint		
API	Endpoint	Description
Post	/api/endeavour/policy	Create a policy.

Recovery Parameters

Table 54: Restore Parameters		
Name	Description	Value
name	Name of the policy.	string, max 128, non-empty
type	RECOVERY	string, max 256
subType	VMWARE-NETAPP	string, max 256
description	Description of the policy.	string, max 64
spec	This is an object.	string, max 32
href	URL for the source selection.	string
recoveryPoint	URL for the recovery point.	string

Recovery Request

```
{
  "name": "RecoveryPolicy1",
  "type": "RECOVERY",
  "subType": "VMWARE-NETAPP",
  "description": "",
  "spec": {
    "source": [
      {
        "href": "http://localhost:8082/api/endeavour/catalog/recoveryvmware/vm/pk.864d814d-870f-47a0-acaa-26998bbaae22-vm-71244",
        "recoveryPoint": "http://localhost:8082/api/endeavour/catalog/recoveryvmware/recoverypoint/pk.4c4b651b-ee77-48da-93f5-790308e03c99",
        "metadata": {
          "path": "864d814d-870f-47a0-acaa-26998bbaae22|vsphere|syncvcenter2/864d814d-870f-47a0-acaa-26998bbaae22-datacenter-31960|datacenter|BEXTech",
          "documentType": "vm",
          "type": "document"
        }
      },
      {
        "href": "http://localhost:8082/api/endeavour/catalog/recoveryvmware/vm/pk.864d814d-870f-47a0-acaa-26998bbaae22-vm-71260",
        "recoveryPoint": "http://localhost:8082/api/endeavour/catalog/recoveryvmware/recoverypoint/pk.4c4b651b-ee77-48da-93f5-790308e03c99",
        "metadata": {
          "path": "864d814d-870f-47a0-acaa-26998bbaae22|vsphere|syncvcenter2/864d814d-870f-47a0-acaa-26998bbaae22-datacenter-31960|datacenter|BEXTech",
          "documentType": "vm",
          "type": "document"
        }
      }
    ],
    "option": {
      "proxyName": "",
      "proxyPort": ""
    },
    "notification": [ ]
  }
}
```

Snapshot Response

```
{
  "links": {
    "self": {
      "rel": "self",
      "href": "http://71.100.15.107:8082/api/endeavour/policy/1004"
    },
    "up": {
      "rel": "up",
      "href": "http://71.100.15.107:8082/api/endeavour/policy"
    },
    "edit": {
      "rel": "update",
      "href": "http://71.100.15.107:8082/api/endeavour/policy/1004"
    },
    "delete": {
      "rel": "delete",
      "href": "http://71.100.15.107:8082/api/endeavour/policy/1004"
    },
    "jobs": {
      "rel": "related",
      "href": "http://71.100.15.107:8082/api/endeavour/job?filter=[{"property":"policyId","value":"1004"}]"
    }
  },
  "name": "RecoveryPolicy1",
  "type": "RECOVERY",
  "subtype": "VMWARE-NETAPP",
  "description": "",
  "version": null,
  "spec": {
    "source": [
      {
        "href": "http://localhost:8082/api/endeavour/catalog/recoveryvmware/vm/pk.864D814D-870F-47A0-ACAA-2699BBAAAE22-vm-71244",
        "recoveryPoint": "http://localhost:8082/api/endeavour/catalog/recoveryvmware/recoverypoint/pk.4c4b651b-ee77-48da-93f5-790308e03c99",
        "metadata": {
          "path": "864D814D-870F-47A0-ACAA-2699BBAAAE22|vsphere|syncvcenter2/864D814D-870F-47A0-ACAA-2699BBAAAE22-datacenter-31960|datacenter|BEXTech",
          "documentType": "vm",
          "type": "document"
        }
      },
      {
        "href": "http://localhost:8082/api/endeavour/catalog/recoveryvmware/vm/pk.864D814D-870F-47A0-ACAA-2699BBAAAE22-vm-71260",
        "recoveryPoint": "http://localhost:8082/api/endeavour/catalog/recoveryvmware/recoverypoint/pk.4c4b651b-ee77-48da-93f5-790308e03c99",
        "metadata": {
          "path": "864D814D-870F-47A0-ACAA-2699BBAAAE22|vsphere|syncvcenter2/864D814D-870F-47A0-ACAA-2699BBAAAE22-datacenter-31960|datacenter|BEXTech",
          "documentType": "vm",
          "type": "document"
        }
      }
    ]
  },
  "option": {
    "proxyName": "",
    "proxyPort": ""
  },
  "notification": []
},
{id": "1004"
}
```

The following is an example of the RESTful API parameters to obtain the recovery point to create the Snapshot recovery policy:

```
# Get a list of VMs that can be recovered.
"/api/endeavour/catalog/recoveryvmware/vm?pagestartIndex=0&pageSize=100&embed=(children) "

# Get the URL for recovery points for a particular VM.
"/api/endeavour/catalog/recoveryvmware/vm/52e1c68dbab809d58cea0213"

# After you get the URL location of recovery points, run the query. Take the output of this
as input into your recovery policy
"/api/endeavour/catalog/recoveryvmware/vm/pk.864D814D-870F-47A0-ACAA-2699BBAAAE22-vm-73436/
recoverypoint"
```

Start a Catalog Job

This example demonstrates how to programmatically start jobs.

The RESTful APIs in this scenario are:

Job Start Endpoint

Table 55: Job Start Endpoint		
API	Endpoint	Description
Post	/api/endeavour/job	Start a catalog job.

Job Start Parameters

Table 56: Job Start Parameters		
Name	Description	Value
action	Query parameter used for performing various actions.	start

Job Start Request

```
POST
URL: https://<ipaddress>:port/api/endeavour/job?action=start"
Request Body:
{}
```

Job Start Response

```

{
  "links":{
    "self":{
      "rel":"self",
      "href":"https://<ip>:<port>/api/endeavour/job/1009"
    },
    "up":{
      "rel":"up",
      "href":"https://<ip>:<port>/api/endeavour/job"
    },
    "edit":{
      "rel":"update",
      "href":"https://<ip>:<port>/api/endeavour/job/1009"
    },
    "delete":{
      "rel":"delete",
      "href":"https://<ip>:<port>/api/endeavour/job/1009"
    },
    "policy":{
      "rel":"related",
      "href":"https://<ip>:<port>/api/endeavour/policy/1009"
    },
    "triggers":{
      "rel":"related",
      "href":"https://<ip>:<port>/api/endeavour/job/1009/trigger"
    },
    "jobsessions":{
      "rel":"related",
      "href":"https://<ip>:<port>/api/endeavour/jobsession?filter=[{"property":"jobId","value":"1009"}]"
    },
    "stats":{
      "rel":"related",
      "href":"https://<ip>:<port>/api/endeavour/jobsession/stats/jobid/1009"
    },
    "firetimes":{
      "rel":"related",
      "href":"https://<ip>:<port>/api/endeavour/firetime?filter=[{"property":"jobId","value":"1009"}]"
    },
    "lastrun":{
      "rel":"related",
      "href":"https://<ip>:<port>/api/endeavour/jobsession/1382994103022"
    },
    "start":{
      "rel":"action",
      "href":"https://<ip>:<port>/api/endeavour/job/1009?action=start"
    }
  }
},

```

```

"name": "test policy",
"description": "test policy",
"policyId": "1009",
"policyName": "test policy",
"type": "CATALOG",
"subType": "NETAPP",
"triggerIds": [
],
"lastRunTime": 1382994103900,
"nextFireTime": null,
"policy": {
  "links": {
    "self": {
      "rel": "self",
      "href": "http://<ip>:<port>/api/endeavour/policy/1009"
    },
    "up": {
      "rel": "up",
      "href": "http://<ip>:<port>/api/endeavour/policy"
    },
    "edit": {
      "rel": "update",
      "href": "http://<ip>:<port>/api/endeavour/policy/1009"
    },
    "delete": {
      "rel": "delete",
      "href": "http://<ip>:<port>/api/endeavour/policy/1009"
    },
    "jobs": {
      "rel": "related",
      "href": "http://<ip>:<port>/api/endeavour/job?filter=[{"property": "policyId", "value": "1009"}]"
    }
  },
  "name": "test policy",
  "type": "CATALOG",
  "subType": "NETAPP",
  "description": "",
  "version": null,
  "spec": {
    "source": [
      {
        "href": "https://<ip>:<port>/api/netapp/1002",
        "resourceType": "netapp",
        "name": "<ip>",
        "id": "1002",
        "tags": {
        }
      }
    ],
    "option": {
      "timeout": "300",
      "retention": "3"
    },
    "notification": [
    ]
  },
  "id": "1009"
},
"triggers": [
],
"lastrun": {
  "sessionId": "1382994103022",
  "jobName": "test policy",
  "type": "CATALOG",
  "subType": "NETAPP",
  "start": null,
  "end": null,
  "status": "WAITING",
  "results": null,
  "properties": null,

```

```

    "numTasks":1,
    "lastUpdate":1382994103900,
    "percent":null,
    "policySnapshot":{
      "id":"1009",
      "name":"test_policy",
      "type":"CATALOG",
      "subType":"JSON",
      "version":null,
      "spec":{
        "source":[
          {
            "href":"https://<ip>:<port>/api/netapp/1002",
            "resourceType":"netapp",
            "name":"<ip>",
            "id":"1002",
            "tags":{
            }
          }
        ]
      },
      "option":{
        "timeout":"300",
        "retention":"3"
      },
      "notification":[
      ]
    }
  },
  "id":"1009"
}

```

The following is an example of the RESTful API parameters and the steps you can take to traverse the vSphere folder structure to determine the link to a VM resource to backup. The vSphere folder structures look like
vCenter> Datacenter > folder1 > folder2 > vm1:

```

# list of registered vCenters
"/api/vsphere"

# get datacenters in vCenter
"/api/vsphere/1003/folder/group-d1/content"

# get folders in particular datacenter
"/api/vsphere/1003/folder/group-v3/content"

# get folders within folder1
"/api/vsphere/1003/folder/group-v66556/content"

# get the list of VMs in folder2
"/api/vsphere/1003/folder/group-v73435/content"

```

Offload and Restore Snapshot Workflow

This example demonstrates a complete workflow for offloading a snapshot and restoring it to a destination path.

Overview

This example shows how to perform the following tasks:

1. Start an offload operation for a snapshot
2. Monitor the offload session status
3. Start a restore operation using the offloaded snapshot
4. Monitor the restore session status

Prerequisites

- You have a valid session ID from the Copy Data Management system
- You have registered an IBM Flash Storage configuration with a known storageId
- You have a snapshot with a known snapshotId that you want to offload

Step 1: Start an Offload Operation

Start the offload operation by sending a POST request to the offload endpoint:

```
curl --request POST 'https://example.com:8443/api/offload' \
--header 'Content-Type: application/json' \
--header 'x-endeavour-sessionid: abc123xyz789' \
--data '{
  "storageId": 1001,
  "snapshotId": 120,
  "name": "nightly-vol1001-snap120"
}'
```

Response:

```
{
  "sessionId": "SESS-1001",
  "type": "offload",
  "status": "PENDING",
  "name": "nightly-vol1001-snap120",
  "startTime": "2024-06-01T10:00:00Z",
  "endTime": null,
  "error": null
}
```

Save the sessionId value (SESS-1001) to monitor the operation.

Step 2: Monitor the Offload Session

Poll the session status endpoint to monitor progress:

```
curl --request GET 'https://example.com:8443/api/offload/sessions/SESS-1001' \
--header 'x-endeavour-sessionid: abc123xyz789'
```

Response (In Progress):

```
{
  "sessionId": "SESS-1001",
  "type": "offload",
  "status": "IN_PROGRESS",
  "name": "nightly-vol1001-snap120",
  "startTime": "2024-06-01T10:00:00Z",
  "endTime": null,
  "error": null
}
```

Continue polling until the status changes to SUCCESSFUL:

```
{
  "sessionId": "SESS-1001",
  "type": "offload",
  "status": "SUCCESSFUL",
  "name": "nightly-vol1001-snap120",
  "startTime": "2024-06-01T10:00:00Z",
  "endTime": "2024-06-01T10:12:45Z",
  "error": null
}
```

Step 3: Start a Restore Operation

After the offload completes successfully, start a restore operation using the same name:

```
curl --request POST 'https://example.com:8443/api/offload/restore' \
--header 'Content-Type: application/json' \
--header 'x-endeavour-sessionid: abc123xyz789' \
--data '{
  "name": "nightly-vol1001-snap120",
  "destinationPath": "/restore_target/"
}'
```

Response:

```
{
  "sessionId": "SESS-1002",
  "type": "restore",
  "status": "PENDING",
  "name": "nightly-vol1001-snap120",
  "startTime": "2024-06-01T14:00:00Z",
  "endTime": null,
  "error": null
}
```

Save the new sessionId value (SESS-1002) to monitor the restore operation.

Step 4: Monitor the Restore Session

Poll the session status endpoint to monitor the restore progress:

```
curl --request GET 'https://example.com:8443/api/offload/sessions/SESS-1002' \
  --header 'x-endeavour-sessionid: abc123xyz789'
```

Response (In Progress):

```
{
  "sessionId": "SESS-1002",
  "type": "restore",
  "status": "IN_PROGRESS",
  "name": "nightly-vol1001-snap120",
  "startTime": "2024-06-01T14:00:00Z",
  "endTime": null,
  "error": null
}
```

Continue polling until the status changes to SUCCESSFUL:

```
{
  "sessionId": "SESS-1002",
  "type": "restore",
  "status": "SUCCESSFUL",
  "name": "nightly-vol1001-snap120",
  "startTime": "2024-06-01T14:00:00Z",
  "endTime": "2024-06-01T14:18:30Z",
  "error": null
}
```

The restore operation is complete. The files have been restored to /restore_target/.

Step 5: List All Sessions (Optional)

You can list all sessions to view both the offload and restore operations:

```
curl --request GET 'https://example.com:8443/api/offload/sessions?name=nightly-vol1001-snap120' \
  --header 'x-endeavour-sessionid: abc123xyz789'
```

Response:

```
{
  "items": [
    {
      "sessionId": "SESS-1001",
      "type": "offload",
      "status": "SUCCESSFUL",
      "name": "nightly-vol1001-snap120",
      "startTime": "2024-06-01T10:00:00Z",
      "endTime": "2024-06-01T10:12:45Z",
      "error": null
    },
    {
      "sessionId": "SESS-1002",
      "type": "restore",
      "status": "SUCCESSFUL",
      "name": "nightly-vol1001-snap120",
      "startTime": "2024-06-01T14:00:00Z",
      "endTime": "2024-06-01T14:18:30Z",

```

```
    "error": null
  }
],
"total": 2,
"limit": 20,
"offset": 0
}
```


Notices

This information was developed for products and services offered in the US. This material might be available from IBM® in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM® may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM® representative for information on the products and services currently available in your area. Any reference to an IBM® product, program, or service is not intended to state or imply that only that IBM® product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM® intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM® product, program, or service.

IBM® may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM® Director of Licensing
IBM® Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM® Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM® Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM® may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM® websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM® product and use of those websites is at your own risk.

IBM® may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM® Director of Licensing
IBM® Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM® under terms of the IBM® Customer Agreement, IBM® International Program License Agreement or any equivalent agreement between us.

The performance data discussed herein is presented as derived under specific operating conditions. Actual results may vary.

Information concerning non-IBM® products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM® has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM® products. Questions on the capabilities of non-IBM® products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM®, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM®, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM® shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows: © (your company name) (year). Portions of this code are derived from IBM® Corp. Sample Programs. © Copyright IBM® Corp. _enter the year or years_.

Trademarks

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

Adobe™ is a registered trademark of Adobe™ Systems Incorporated in the United States, and/or other countries.

Linear Tape-Open™, LTO™, and Ultrium™ are trademarks of HP, IBM® Corp. and Quantum in the U.S. and other countries.

Intel™ and Itanium™ are trademarks or registered trademarks of Intel™ Corporation or its subsidiaries in the United States and other countries.

The registered trademark Linux® is used pursuant to a sublicense from the Linux® Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

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

Java™ and all Java™-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

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

UNIX® is a registered trademark of The Open Group in the United States and other countries.

VMware, VMware vCenter Server™, and VMware vSphere™ are registered trademarks or trademarks of VMware, Inc. or its subsidiaries in the United States and/or other jurisdictions.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM® website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM®.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM®.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM® reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM®, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM® MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Privacy policy considerations

IBM® Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM®'s Privacy Policy at <http://www.ibm.com/privacy> and IBM®'s Online Privacy Statement at <http://www.ibm.com/privacy/details> in the section entitled "Cookies, Web Beacons and Other Technologies," and the "IBM® Software Products and Software-as-a-Service Privacy Statement" at <http://www.ibm.com/software/info/product-privacy>.

Glossary

A glossary is available with terms and definitions for the IBM® Storage Defender Copy Data Management family of products.

See the IBM® Storage Defender Copy Data Management [glossary](#).

© Copyright International Business Machines Corporation 2017, 2025

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

