ONTAP REST API

ONTAP Automation

NetApp
April 21, 2024

Table of Contents

ONTAP REST API

REST implementation details

REST web services foundation

Representational State Transfer (REST) is a style for creating distributed web applications. When applied to the design of a web services API, it establishes a set of technologies for exposing server-based resources and managing their states. It uses mainstream protocols and standards to provide a flexible foundation for administering ONTAP clusters.

While REST establishes a common set of technologies and best practices, the details of each API can vary based on the choices made during development. You should be aware of the design characteristics of the ONTAP REST API before using it with a live deployment.

Resources and state representation

Resources are the basic components of a web-based system. When creating a REST web services application, early design tasks include:

- Identification of system or server-based resources

  Every system uses and maintains resources. A resource can be a file, business transaction, process, or administrative entity. One of the first tasks in designing an application based on REST web services is to identify the resources.

- Definition of resource states and associated state operations

  Resources are always in one of a finite number of states. The states, as well as the associated operations used to affect the state changes, must be clearly defined.

URI endpoints

Every REST resource must be defined and made available using a well-defined addressing scheme. The endpoints where the resources are located and identified use a Uniform Resource Identifier (URI). The URI provides a general framework for creating a unique name for each resource in the network. The Uniform Resource Locator (URL) is a type of URI used with web services to identify and access resources. Resources are typically exposed in a hierarchical structure similar to a file directory.

HTTP messages

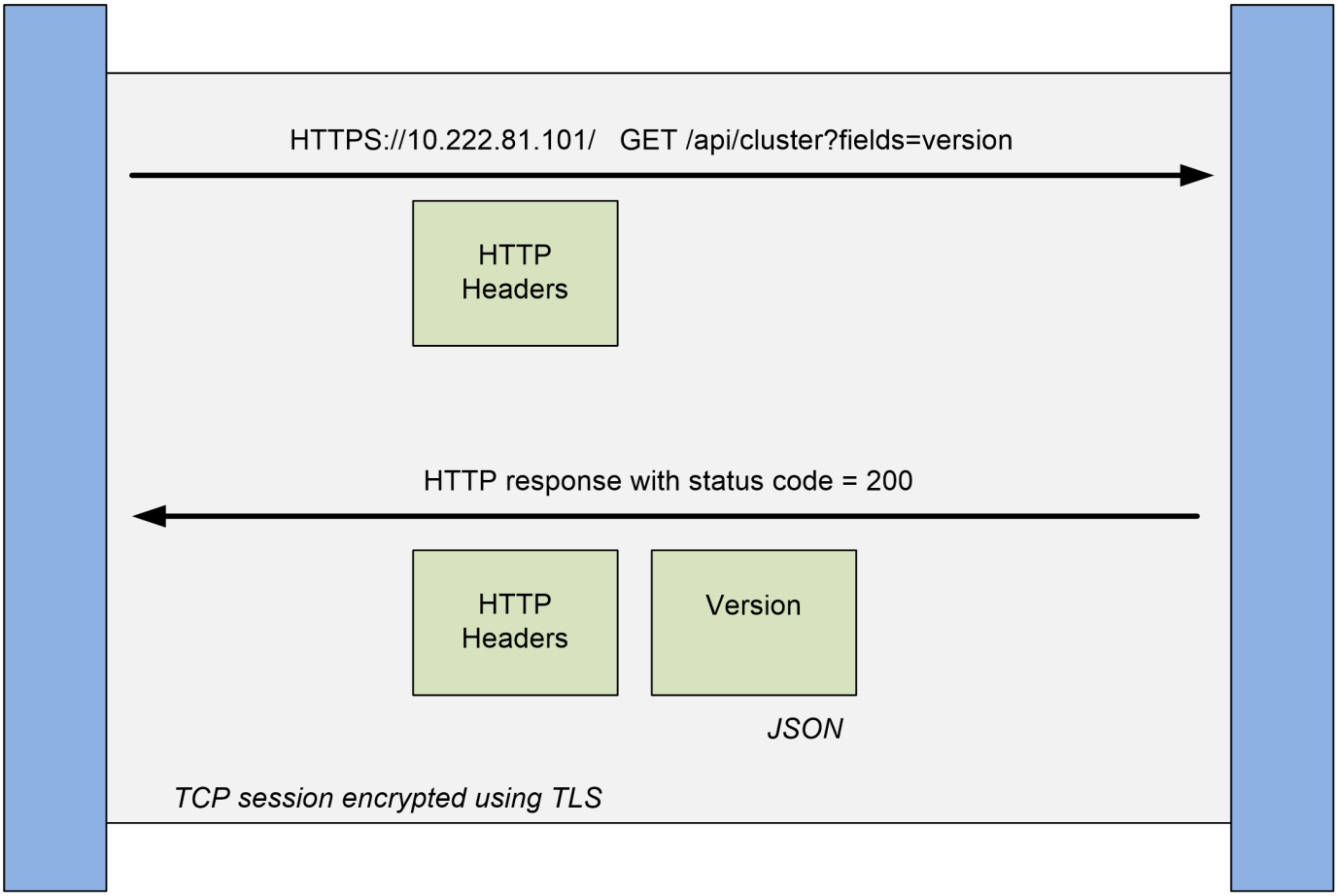
Hypertext Transfer Protocol (HTTP) is the protocol used by the web services client and server to exchange request and response messages about the resources. As part of designing a web services application, HTTP methods are mapped to the resources and corresponding state management actions. HTTP is stateless. Therefore, to associate a set of related requests and responses as part of one transaction, additional information must be included in the HTTP headers carried with the request and response data flows.

JSON formatting

Although information can be structured and transferred between a web services client and server in several ways, the most popular option is JavaScript Object Notation (JSON). JSON is an industry standard for representing simple data structures in plain text and is used to transfer state information describing the resources. The ONTAP REST API uses JSON to format the data carried in the body of each HTTP request and response.

Typical REST API transaction

Every API transaction consists of an HTTP request and the associated response. This illustration shows how to retrieve the version of the ONTAP software used by the cluster.



HTTP request

The request sent from the client to the server consists of the following:

- GET verb
- URL path for the cluster
- Query parameter (fields)
- Request headers, including authorization

HTTP response

The response sent from the server to the client consists of the following:

- Status code 200
- Response headers
- Response body containing the cluster software version

Basic operational characteristics

While REST establishes a common set of technologies and best practices, the details of each API can vary based on the design choices.

Request and response API transaction

Every REST API call is performed as an HTTP request to the ONTAP system which generates an associated response to the client. This request/response pair is considered an API transaction. Before using the API, you should be familiar with the input variables available to control a request and the contents of the response output.

Support for CRUD operations

Each of the resources available through the ONTAP REST API is accessed based on the CRUD model:

- Create
- Read
- Update
- Delete

For some of the resources, only a subset of the operations is supported. You should review the ONTAP API documentation page at your ONTAP cluster for more information about each resource.

Object identifiers

Each resource instance or object is assigned a unique identifier when it is created. In most cases, the identifier is a 128-bit UUID. These identifiers are globally unique within a specific ONTAP cluster. After issuing an API call that creates a new object instance, a URL with the associated id value is returned to the caller in the location header of the HTTP response. You can extract the identifier and use it on subsequent calls when referring to the resource instance.

The content and internal structure of the object identifiers can change at any time. You should only use the identifiers on the applicable API calls as needed when referring to the associated objects.

Object instances and collections

Depending on the resource path and HTTP method, an API call can apply to a specific object instance or a collection of objects.

Synchronous and asynchronous operations

There are two ways that ONTAP performs an HTTP request received from a client.

Synchronous processing

ONTAP performs the request immediately and responds with an HTTP status code of 200 or 201 if it is successful.

Every request using the methods GET, HEAD, and OPTIONS is always performed synchronously. In addition, requests that use POST, PATCH, and DELETE are designed to run synchronously if they are expected to complete in less than two seconds.

Asynchronous processing

If an asynchronous request is valid, ONTAP creates a background task to process the request and a job object to anchor the task. The 202 HTTP status is returned to the caller along with the job object. To determine final success or failure, you must retrieve the state of the job.

Requests that use the methods POST, PATCH, and DELETE are designed to run asynchronously if they are expected to take more than two seconds to complete.

The `return_timeout` query parameter is available with asynchronous API calls and can convert an asynchronous call to complete synchronously. Refer to Asynchronous processing using the Job object for more information.

Security

The security provided with the REST API is based primarily on the existing security features available with ONTAP. The following security is used by the API:

Transport Layer Security

All traffic sent over the network between the client and ONTAP LIF is typically encrypted using TLS, based on the ONTAP configuration settings.

Client authentication

The same authentication options available with ONTAP System Manager and the Network Manageability SDK can also be used with the ONTAP REST API.

HTTP authentication

At an HTTP level, for example when accessing the ONTAP REST API directly, there are two authentication options as described below. In each case, you need to create an HTTP authorization header and include it with each request.

Option	Description
HTTP basic authentication	The ONTAP username and password are concatenated with a colon. The string is converted to base64 and included in the request header.
OAuth 2.0	Beginning with ONTAP 9.14, you can request an access token from an external authorization server and include it as a bearer token in the request header.

For more details about OAuth 2.0 and how it is implemented in ONTAP, see Overview of the ONTAP OAuth 2.0 implementation. Also see Prepare to use the workflows below at this site.

ONTAP authorization

ONTAP implements a role-based authorization model. The account you use when accessing the ONTAP REST API or API documentation page should have the proper authority.

Input variables controlling an API request

You can control how an API call is processed through parameters and variables set in the HTTP request.

HTTP methods

The HTTP methods supported by the ONTAP REST API are shown in the following table.

Not all the HTTP methods are available at each of the REST endpoints. Also, both PATCH and DELETE can be used on a collection. See *Object references and access* for more information.

HTTP method	Description
GET	Retrieves object properties on a resource instance or collection.
POST	Creates a new resource instance based on the supplied input.
PATCH	Updates an existing resource instance based on the supplied input.
DELETE	Deletes an existing resource instance.
HEAD	Effectively issues a GET request but only returns the HTTP headers.
OPTIONS	Determine what HTTP methods are supported at a specific endpoint.

Path variables

The endpoint path used with each REST API call can include various identifiers. Each ID corresponds to a specific resource instance. Examples include cluster ID and SVM ID.

Request headers

You must include several headers in the HTTP request.

Content-type

If the request body includes JSON, this header must be set to `application/json`.

Accept

This header should be set to `application/hal+json`. If it is instead set to `application/json` none of the HAL links will be returned except a link needed to retrieve the next batch of records. If the header is anything else aside from these two values, the default value of the `content-type` header in the response will be `application/hal+json`.

Authorization

Basic authentication must be set with the user name and password encoded as a base64 string. For example:

```
Authorization: Basic YWRtaW46cGV0ZXJzb24=.
```

Request body

The content of the request body varies depending on the specific call. The HTTP request body consists of one of the following:

- JSON object with input variables
- Empty JSON object

Filtering objects

When issuing an API call with the GET method, you can limit or filter the returned objects based on any attribute using a query parameter.

Parsing and interpreting query parameters

A set of one or more parameters can be appended to the URL string beginning after the `?` character. If more than one parameter is provided, the query parameters are split based on the `&` character. Each key and value in the parameter are split at the = character.

For example, you can specify an exact value to match using the equal sign:

```
<field>=<value>
```

For a more complex query, the additional operator is placed after the equal sign. For example, to select the set of objects based on a specific field that is greater than or equal to some value, the query would be:

```
<field>=>=<value>
```

Filtering operators

In addition to the examples provided above, additional operators are available to return objects over a range of values. A summary of the filtering operators supported by the ONTAP REST API is shown in the table below.

Any fields that are not set are generally excluded from matching queries.

Operator	Description
=	Equal to
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to
!	Not equal to
*	Greedy wildcard

You can also return a collection of objects based on whether a specific field is set or not set by using the `null` keyword or its negation `!null` as part of the query.

Workflow examples

Some examples are included below from the REST API workflows at this site.

- List disks

  Filter based on the `state` variable to select the spare disks.

Requesting specific object fields

By default, issuing an API call using GET returns only the attributes that uniquely identify the object or objects, along with a HAL self link. This minimum set of fields acts as a key for each object and varies based on the object type. You can select additional object properties using the `fields` query parameter in the following ways:

- Common or standard fields

  Specify `fields=*` to retrieve the most commonly used object fields. These fields are typically maintained in local server memory or require little processing to access. These are the same properties returned for an object after using GET with a URL path key (UUID).

- All fields

  Specify `fields=**` to retrieve all the object fields, including those requiring additional server processing to access.

- Custom field selection

  Use `fields=<field_name>` to specify the exact field you want. When requesting multiple fields, the values must be separated using commas without spaces.

As a best practice, you should always identify the specific fields you want. You should only retrieve the set of common fields or all fields when needed. Which fields are classified as common, and returned using `fields=*`, is determined by NetApp based on internal performance analysis. The classification of a field might change in future releases.

Sorting objects in the output set

The records in a resource collection are returned in the default order defined by the object. You can change the order using the `order_by` query parameter with the field name and sort direction as follows:

```
order_by=<field name> asc|desc
```

For example, you can sort the type field in descending order followed by id in ascending order:

```
order_by=type desc, id asc
```

Note the following:

- If you specify a sort field but don’t provide a direction, the values are sorted in ascending order.
- When including multiple parameters, you must separate the fields with a comma.

Pagination when retrieving objects in a collection

When issuing an API call using GET to access a collection of objects of the same type, ONTAP attempts to return as many objects as possible based on two constraints. You can control each of these constraints using additional query parameters on the request. The first constraint reached for a specific GET request terminates the request and therefore limits the number of records returned.

If a request ends before iterating over all the objects, the response contains the link needed to retrieve the next batch of records.

Limiting the number of objects

By default, ONTAP returns a maximum of 10,000 objects for a GET request. You can change this limit using the `max_records` query parameter. For example:

```
max_records=20
```

The number of objects actually returned can be less than the maximum in effect, based on the related time constraint as well as the total number of objects in the system.

Limiting the time used to retrieve the objects

By default, ONTAP returns as many objects as possible within the time allowed for the GET request. The default timeout is 15 seconds. You can change this limit using the `return_timeout` query parameter. For example:

```
return_timeout=5
```

The number of objects actually returned can be less than the maximum in effect, based on the related constraint on the number of objects as well as the total number of objects in the system.

Narrowing the result set

If needed, you can combine these two parameters with additional query parameters to narrow the result set. For example, the following returns up to 10 ems events generated after the specified time:

```
time=> 2018-04-04T15:41:29.140265Z&max_records=10
```

You can issue multiple requests to page through the objects. Each subsequent API call should use a new time value based on the latest event in the last result set.

Size properties

The input values used with some API calls as well as certain query parameters are numeric. Rather than provide an integer in bytes, you can optionally use a suffix as shown in the following table.

Suffix	Description
KB	KB Kilobytes (1024 bytes) or kibibytes
MB	MB Megabytes (KB x 1024 bytes) or mebibytes
GB	GB Gigabytes (MB x 1024 bytes) or gibibytes
TB	TB Terabytes (GB x 1024 bytes) or tebibytes
PB	PB Petabytes (TB x 1024 bytes) or pebibytes

Related information

- Object references and access

Interpreting an API response

Each API request generates a response back to the client. You should examine the response to determine whether it was successful and retrieve additional data as needed.

HTTP status code

The HTTP status codes used by the ONTAP REST API are described below.

Code	Reason phrase	Description
200	OK	Indicates success for calls that do not create a new object.
201	Created	An object is successfully created. The location header in the response includes the unique identifier for the object.
202	Accepted	A background job has been started to perform the request, but has not completed yet.
400	Bad request	The request input is not recognized or is inappropriate.
401	Unauthorized	User authentication has failed.
403	Forbidden	Access is denied due to an authorization error.
404	Not found	The resource referred to in the request does not exist.
405	Method not allowed	The HTTP method in the request is not supported for the resource.
409	Conflict	An attempt to create an object failed because a different object must be created first or the requested object already exists.
500	Internal error	A general internal error occurred at the server.

Response headers

Several headers are included in the HTTP response generated by the ONTAP.

Location

When an object is created, the location header includes the complete URL to the new object including the unique identifier assigned to the object.

Content-type

This will normally be `application/hal+json`.

Response body

The content of the response body resulting from an API request differs based on the object, processing type, and the success or failure of the request. The response is always rendered in JSON.

- Single object

  A single object can be returned with a set of fields based on the request. For example, you can use GET to retrieve selected properties of a cluster using the unique identifier.

- Multiple objects

  Multiple objects from a resource collection can be returned. In all cases, there is a consistent format used, with `num_records` indicating the number of records and records containing an array of the object instances. For example, you can retrieve the nodes defined in a specific cluster.

- Job object

If an API call is processed asynchronously, a Job object is returned which anchors the background task. For example, the PATCH request used to update the cluster configuration is processed asynchronously and returns a Job object.

- Error object

  If an error occurs, an Error object is always returned. For example, you will receive an error when attempting to change a field not defined for a cluster.

- Empty JSON object

  In certain cases, no data is returned and the response body includes an empty JSON object.

HAL linking

The ONTAP REST API uses HAL as the mechanism to support Hypermedia as the Engine of Application State (HATEOAS). When an object or attribute is returned that identifies a specific resource, a HAL-encoded link is also included allowing you to easily locate and determine additional details about the resource.

Errors

If an error occurs, an error object is returned in the response body.

Format

An error object has the following format:

```
"error": {
"message": "<string>",
"code": <integer>[,
"target": "<string>"]
}
```

You can use the code value to determine the general error type or category, and the message to determine the specific error. When available, the target field includes the specific user input associated with the error.

Common error codes

The common error codes are described in the following table. Specific API calls can include additional error codes.

Code		Description
1	409	An object with the same identifier already exists.
2	400	The value for a field has an invalid value or is missing, or an extra field was provided.
3	400	The operation is not supported.
4	405	An object with the specified identifier cannot be found.

Code		Description
6	403	Permission to perform the request is denied.
8	409	The resource is in use.

Asynchronous processing using the Job object

After issuing an API request that is designed to run asynchronously, a job object is always created and returned to the caller. The job describes and anchors a background task that processes the request. Depending on the HTTP status code, you must retrieve the state of the job to determine if the request was successful.

Refer to API reference to determine which API calls are designed to be performed asynchronously.

Controlling how a request is processed

You can use the `return_timeout` query parameter to control how an asynchronous API call is processed. There are two possible outcomes when using this parameter.

Timer expires before the request completes

For valid requests, ONTAP returns a 202 HTTP status code along with the job object. You must retrieve the state of the job to determine if the request completed successfully.

Request is completed before the timer expires

If the request is valid and completes successfully before the time expires, ONTAP returns a 200 HTTP status code along with the job object. Because the request is completed synchronously, as indicated by the 200, you do not need to retrieve the job state.

The default value for the `return_timeout` parameter is zero seconds. Therefore, if you don't include the parameter, the 202 HTTP status code is always returned for a valid request.

Querying the Job object associated with an API request

The Job object returned in the HTTP response contains several properties. You can query the state property in a subsequent API call to determine if the request completed successfully. A Job object is always in one of the following states:

Non-terminal states

- Queued
- Running
- Paused

Terminal states

- Success
- Failure

General procedure for issuing an asynchronous request

You can use the following high-level procedure to complete an asynchronous API call. This example assumes the `return_timeout` parameter is not used, or that the time expires before the background job completes.

1. Issue an API call that is designed to be performed asynchronously.
2. Receive an HTTP response 202 indicating acceptance of a valid request.
3. Extract the identifier for the Job object from the response body.
4. Within a timed loop, perform the following in each cycle:
   a. Get the current state of the Job.
   b. If the Job is in a non-terminal state, perform loop again.
5. Stop when the Job reaches a terminal state (success, failure).

Related information

- Update cluster contact
- Get job instance

Object references and access

The resource instances or objects exposed through the ONTAP REST API can be referenced and accessed in several different ways.

Object access paths

At a high level, there are two path types when accessing an object:

- Primary

  The object is the primary or direct target of the API call.

- Foreign

  The object is not the primary reference of the API call, but rather is linked to from the primary object. It is therefore a foreign or downstream object and referenced through a field in the primary object.

Accessing an object using the UUID

Every object is assigned a unique identifier when it is created, which in most cases is a 128-bit UUID. The assigned UUID values are immutable and are used internally within ONTAP to access and manage the resources. Because of this, the UUID generally provides the fastest and most stable way to access objects.

For many of the resource types, a UUID value can be provided as part of the path key in the URL to access a specific object. For example, you can use the following to access a node instance: `` `/cluster/nodes/{uuid} ``

Accessing an object using an object property

In addition to a UUID, you can also access an object using an object property. In most cases, it is convenient to use the name property. For example, you can use the following query parameter in the URL string to access a node instance by its name: `/cluster/nodes?name=node_one`. In addition to a query parameter, a foreign

object can be accessed through a property in the primary object.

While you can use the name or other property to access an object instead of the UUID, there are several possible disadvantages:

- The name field is not immutable and can be changed. If the name of an object is changed before accessing an object, the wrong object will be returned or an object access error will fail.

  This issue can occur with a POST or PATCH method on a foreign object or with a GET method on a primary object.

- ONTAP must translate the name field into the corresponding UUID. This is a type of indirect access which can become a performance issue.

In particular, a performance degradation is possible when one or more of the following is true:

- GET method is used
- A large collection of objects is accessed
- A complex or elaborate query is used

Cluster versus SVM context

There are several REST endpoints that support both a cluster and SVM. When using one of these endpoints, you can indicate the context of the API call through the `scope=[svm|cluster]` value. Examples of endpoints supporting a dual context include IP interfaces and security roles.

The scope value has a default value base on the properties provided for each API call.

Using PATCH and DELETE on a collection of objects

Every REST endpoint supporting PATCH or DELETE on a resource instance also supports the same method on a collection of objects. The only requirement is that at least one field must be provided through a query parameter in the URL string. When issuing a PATCH or DELETE over a collection, this is equivalent to doing the following internally:

- Query-based GET to retrieve the collection
- Serial sequence of PATCH or DELETE calls on each object in the collection

The time out for the operation can be set by `return_timeout` with a default of 15 seconds. If not completed before the timeout, the response includes a link to the next object. You must reissue the same HTTP method using the next link to continue the operation.

Performance metrics for storage resources

ONTAP collects performance metrics about selected SVM storage objects and protocols, and reports this information through the REST API. You can use this data to monitor the performance of an ONTAP system.

For a given storage object or protocol, the performance data falls into three categories:

- IOPS

- Latency
- Throughput

Within each category, one or more of the following types of data is available:

- Read ®
- Write (W)
- Other (O)
- Total (T)

The following table summarizes the performance data available through the ONTAP REST API, including the release when it was added. Refer to the REST API online documentation page at your ONTAP system for more information.

Storage object or protocol	IOPS	Latency	Throughput	ONTAP release
Ethernet port	Not applicable	Not applicable	RWT	9.8
FC port	RWOT	RWOT	RWT	9.8
IP interface	Not applicable	Not applicable	RWT	9.8
FC interface	RWOT	RWOT	RWT	9.8
NVMe namespace	RWOT	RWOT	RWOT	9.8
Qtree statistics	Raw RWOT	Not applicable	Raw RWOT	9.8
Volume Flexcache	RWOT	RWOT	RWT	9.8
Node – process utilization	Process utilization as a numerical value	Process utilization as a numerical value	Process utilization as a numerical value	9.8
Cloud volume	RWOT	RWOT	Not appliable	9.7
LUN	RWOT	RWOT	RWOT	9.7
Aggregate	RWOT	RWOT	RWOT	9.7
SVM NFS protocol	RWOT	RWOT	RWT	9.7
SVM CIFS protocol	RWOT	RWOT	RWT	9.7
SVM FCP protocol	RWOT	RWOT	RWT	9.7
SVM iSCSI protocol	RWOT	RWOT	RWT	9.7
SVM NVMe protocol	RWOT	RWOT	RWT	9.7
Cluster	RWOT	RWOT	RWOT	9.6
Volumes	RWOT	RWOT	RWOT	9.6

RBAC security

Overview of RBAC security

ONTAP includes a robust and extensible role-based access control (RBAC) capability. You can assign each account a different role to control the user's access to the resources exposed through the REST API and CLI. The roles define different levels of administrative access for the various ONTAP users.

The ONTAP RBAC capability has continued to expand and was significantly enhanced with ONTAP 9.11.1 (and subsequent releases). See Summary of RBAC evolution and What's new with the ONTAP REST API and automation for more information.

ONTAP roles

A role is a set of privileges that collectively define what actions the user can take. Each privilege identifies a specific access path and the associated access level. Roles are assigned to user accounts and applied by ONTAP when making access control decisions.

Types of roles

There are two types of roles. They were introduced and tailored to different environments as ONTAP has evolved.

There are advantages and disadvantages when using each type of role. See Comparing the role types for more information.

Type	Description
REST	The REST roles were introduced with ONTAP 9.6 and are generally applied to users accessing ONTAP through the REST API. Creating a REST role automatically creates a traditional *mapping* role.
Traditional	These are the legacy roles included prior to ONTAP 9.6. They were introduced for the ONTAP CLI environment and continue to be fundamental to RBAC security.

Scope

Every role has a scope or context within which it is defined and applied. The scope determines where and how a specific role is used.

ONTAP user accounts also have a similar scope that determines how a user is defined and used.

Scope	Description
Cluster	Roles with a cluster scope are defined at the ONTAP cluster level. They are associated with cluster-level user accounts.
SVM	Roles with an SVM scope are defined for a specific data SVM. They are assigned to user accounts in the same SVM.

Source of the role definitions

There are two ways an ONTAP role can be defined.

Role source	Description
Custom	The ONTAP administrator can create custom roles. These roles can be tailored to a specific environment and security requirements.
Built-in	While custom roles provide more flexibility, there is also a set of built-in roles available at both the cluster and SVM level. These roles are pre-defined and can be used for many common administrative tasks.

Role mapping and ONTAP processing

Depending on the ONTAP release you are using, all or nearly all the REST API calls map to one or more CLI commands. When you create a REST role, a traditional or legacy role is also created. This **mapped** traditional role is based on the corresponding CLI commands and cannot be manipulated or changed.

Reverse role mapping is not supported. That is, creating a traditional role does not create a corresponding REST role.

Summary of RBAC evolution

The traditional roles are included with all ONTAP 9 releases. The REST roles were introduced later and have evolved as described below.

ONTAP 9.6

The REST API was introduced with ONTAP 9.6. The REST roles were included with this release as well. Also, when you create a REST role, a corresponding traditional role is also created.

ONTAP 9.7 through 9.10.1

Each ONTAP release from 9.7 through 9.10.1 includes enhancements to the REST API. For example, additional REST endpoints have been added with each release. However, the creation and management of the two roles types remained separate. Also, ONTAP 9.10.1 added REST RBAC support for the snapshots REST endpoint `/api/storage/volumes/{vol.uuid}/snapshots` which is a resource-qualified endpoint.

ONTAP 9.11.1

The ability to configure and manage traditional roles using the REST API was added with this release. Additional access levels for the REST roles were also added.

Work with roles and users

After understanding the basic RBAC capabilities, you can get started working with the ONTAP roles and users.

See RBAC workflows for examples of how to create and use roles with the ONTAP REST API.

Administrative access

You can create and manage the ONTAP roles through the REST API or command line interface. The access details are described below.

REST API

There are several endpoints that can be used when working with RBAC roles and user accounts. The first four in the table are used to create and manage the roles. The last two are used to create and manage user accounts.

You can access the ONTAP online API reference documentation for more information including examples of how to use the API.

Endpoint	Description
`/security/roles`	This endpoint allows you to create a new REST role. And beginning with ONTAP 9.11.1 you can also create a traditional role. In this case, ONTAP determines the role type based on the input parameters. You can also retrieve a list of the defined roles.
`/security/roles/{owner.UUID}/{name}`	You can retrieve or delete a specific cluster or SVM scoped role. The UUID value identifies the SVM where the role is defined (cluster or data SVM). The name value is the name of the role.
`/security/roles/{owner.UUID}/{name}/privileges`	This endpoint allows you to configure the privileges for a specific role. The built-in roles can be retrieved but not updated. See the API reference documentation for your ONTAP release for more information.
`/security/roles/{owner.UUID}/{name}/privileges/[path]`	You can retrieve, modify, and delete the access level and optional query value for a specific privilege. See the API reference documentation for your ONTAP release for more information.
`/security/accounts`	This endpoint allows you to create a new cluster or SVM scoped user account. Several types of information must be included or subsequently added before the account is operational. You can also retrieve a list of the defined user accounts.
`/security/accounts/{owner.UUID}/{name}`	You can retrieve, modify, and delete a specific cluster or SVM scoped user account. The UUID value identifies the SVM where the user is defined (cluster or data SVM). The name value is the name of the account.

Command line interface

The relevant ONTAP CLI commands are described below. All commands are accessed at the cluster level through an administrator account.

Command	Description
`security login`	This is the directory containing the commands needed to create and manage a user login.
`security login rest-role`	This is the directory containing the commands needed to create and manage a REST role associated with a user login.
`security login role`	This is the directory containing the commands needed to create and manage a traditional role associated with a user login.

Role definitions

The REST and traditional roles are defined through a set of attributes.

Owner and scope

A role can be owned by the ONTAP cluster or a specific data SVM within the cluster. The owner also implicitly determines the scope of the role.

Unique name

Every role must have a unique name within its scope. The name of a cluster role must be unique at the ONTAP cluster level while SVM roles must be unique within the specific SVM.

The name of a new REST role must be unique among the REST roles as well as the traditional roles. This is because creating a REST role also results in a new traditional *mapping* role with the same name.

Set of privileges

Every role contains a set of one or more privileges. Each privilege identifies a specific resource or command and the associated access level.

Privileges

A role can contain one or more privileges. Each privilege definition is a tuple and establishes the level of access to a specific resource or operation.

Resource path

The resource path is identified as either a REST endpoint or CLI command/command directory path.

REST endpoint

An API endpoint identified the target resource for a REST role.

CLI command

A CLI command identifies the target for a traditional role. A command directory can also be specified, which will then include all the downstream commands in the ONTAP CLI hierarchy.

Access level

The access level defines the type of access the role has to the specific resource path or command. The access levels are identified through a set of pre-defined keywords. Three access levels were introduced with ONTAP 9.6. They can be used for both traditional and REST roles. In addition, three new access levels were added with ONTAP 9.11.1. These new access levels can only be used with REST roles.

The access levels follow the CRUD model. With REST, this is based on the primary HTTP methods (POST, GET, PATCH, DELETE). The corresponding CLI operations generally map to the REST operations (create, show, modify, delete).

Access level	REST primitives	Added	REST role only
none	n/a	9.6	No
readonly	GET	9.6	No

Access level	REST primitives	Added	REST role only
all	GET, POST, PATCH, DELETE	9.6	No
read_create	GET, POST	9.11.1	Yes
read_modify	GET, PATCH	9.11.1	Yes
read_create_modify	GET, POST, PATCH	9.11.1	Yes

Optional query

When creating a traditional role, you can optionally include a **query** value to identify the subset of applicable objects for the command or command directory.

Summary of the built-in roles

There are several pre-defined roles included with ONTAP that you can use at either the cluster or SVM level.

Cluster scoped roles

There are several built-in roles available at the cluster scope.

See Predefined roles for cluster administrators for more information.

Role	Description
admin	Administrators with this role have unrestricted rights and can do anything in the ONTAP system. They can configure all cluster-level and SVM-level resources.
autosupport	This is a special role tailored for the AutoSupport account.
backup	This Special role for backup software that needs to back up the system.
snaplock	This is a special role tailored for the SnapLock account.
readonly	Administrators with this role can view everything at the cluster level but can’t make any changes.
none	No administrative capabilities are provided.

SVM scoped roles

There are several built-in roles available at the SVM scope. The **vsadmin** provides access to the most general and powerful capabilities. There are several additional roles tailored to specific administrative tasks, including:

- vsadmin-volume
- vsadmin-protocol
- vsadmin-backup
- vsadmin-snaplock
- vsadmin-readonly

See Predefined roles for SVM administrators for more information.

Comparing the role types

Before selecting a **REST** role or **traditional** role, you should be aware of the differences. Some of the ways the two role types can be compared are described below.

For more advanced or complex RBAC use cases, you should normally use a traditional role.

How the user accesses ONTAP

Before creating a role, it is important to know how the user will access the ONTAP system. Based on this a role type can be determined.

Access	Suggested type
REST API only	The REST role is designed to be used with the REST API.
REST API and CLI	You can define a REST role which also creates a corresponding traditional role.
CLI only	You can create a traditional role.

Precision of the access path

The access path defined for a REST role is based on a REST endpoint. The access path for a traditional role is based on a CLI command or command directory. In addition, you can include an optional query parameter with a traditional role to further restrict access based on the command parameter values.

Summary of the REST resources

Overview of the resource categories

The resources available through the ONTAP REST API are organized in categories. Each of the resource categories includes a brief description along with additional usage considerations where appropriate.

The REST resources described in the summary are based on the latest version of the product. If you need a more detailed understanding of the changes made in previous releases, see What's new with the ONTAP REST API as well as the ONTAP Release Notes.

For many of the REST endpoints, you can include a UUID key as part of the path string to access a specific object instance. However, in many cases you can also access objects using a property value on a query parameter.

Related information

- API reference

Application

You can use these API calls to manage the ONTAP application resources.

Application snapshots

Applications support snapshot copies, which can be created or restored at any time. This resource type was introduced with ONTAP 9.6.

Applications

The ONTAP applications are arranged based on type, including: templates, applications, components, and Snapshot copies. This resource type was introduced with ONTAP 9.6.

Consistency groups

A consistency group is a set of volumes that are grouped together when performing certain operations such as a snapshot. This feature extends the same crash consistency and data integrity implicit with single-volume operations across a set of volumes. This resource type was introduced with ONTAP 9.10 and updated with 9.12. An endpoint to retrieve metric performance and capacity data was added with ONTAP 9.13.

Consistency groups snapshots

You can use these endpoints to copy, create, inventory, and restore SnapShots for a consistency group. This resource type was introduced with ONTAP 9.10.

Cloud

You can use these API calls to manage connections to object storage resources in the cloud.

Targets

A target represents an object storage resource in the cloud. Each target includes the configuration information needed to connect to the storage resource. This resource type was introduced with ONTAP 9.6.

Cluster

You can use these API calls to manage ONTAP clusters and the related resources.

Capacity pools

The capacity pools licensing model allows you to license storage capacity for each cluster node from a shared pool. This resource type is new with ONTAP 9.8.

Chassis

The chassis is the hardware framework supporting a cluster. This resource type was introduced with ONTAP 9.6.

Clusters

An ONTAP cluster contains one or more nodes and the related configuration settings which define the storage system. This resource type was introduced with ONTAP 9.6.

Counter tables

Various statistical information about ONTAP is captured by the Counter Manager subsystem. You can access this information to assess system performance. This resource type was introduced with ONTAP 9.11.

Firmware

You can retrieve a history of the firmware update requests. This resource type is new with ONTAP 9.8.

Jobs

Asynchronous REST API requests are performed using a background task anchored by a job. This resource type was introduced with ONTAP 9.6.

License instance

Each license can be managed as a separate package. This resource type was introduced with ONTAP 9.6.

License managers

You can manage configuration and other information related to each license manager instance associated with an ONTAP cluster. This resource type is new with ONTAP 9.8.

Licenses

The licenses allow you to implement specific ONTAP features and functionality. This resource type was introduced with ONTAP 9.6.

Mediators

You can manage the mediator associated with MetroCluster, including adding or removing the mediator instance. This resource type is new with ONTAP 9.8.

MetroCluster

You can create and manage a MetroCluster deployment, including executing switchover or switchback operations. This resource type is new with ONTAP 9.8 and updated with 9.11.

MetroCluster diagnostics

You can perform a diagnostic operation on a MetroCluster deployment and retrieve the results. This resource type is new with ONTAP 9.8.

MetroCluster DR groups

You can perform operations related to the MetroCluster DR groups. This resource type is new with ONTAP 9.8.

MetroCluster interconnects

You can retrieve the MetroCluster interconnect status. This resource type is new with ONTAP 9.8.

MetroCluster nodes

You can retrieve the status of the individual nodes in a MetroCluster deployment. This resource type is new with ONTAP 9.8.

MetroCluster operations

You can retrieve a list of the recently executed operations for a MetroCluster configuration. This resource type is new with ONTAP 9.8.

MetroCluster SVMs

You can retrieve information about all the SVM pairs in a MetroCluster configuration. This resource type was introduced with ONTAP 9.11.1.

Nodes

ONTAP clusters are composed of one or more nodes. This resource type was introduced with ONTAP 9.6 and updated with ONTAP 9.8.

NTP keys

The Network Time Protocol (NTP) can be configured to use shared private keys between ONTAP and trusted external NTP time servers. This resource type was introduced with ONTAP 9.7.

NTP server

You can use these API calls to configure the ONTAP Network Time Protocol settings, including the external NTP servers and keys. This resource type was introduced with ONTAP 9.7.

Peers

The peer objects represent endpoints and support the cluster peering relationships. This resource type was introduced with ONTAP 9.6.

Performance counters

Previous ONTAP releases have maintained statistical information about the operational characteristics of the system. With the 9.11.1 release, the information has been enhanced and is now available through the REST API. This feature brings the ONTAP REST API closer to parity with the Data ONTAP API (ONTAPI or ZAPI). This resource type was introduced with ONTAP 9.11.

Resource tags

You can use tags to group REST API resources. You might do this to associate related resources within a specific project or organizational group. Using tags can help to organize and track resources more effectively. This resource type was introduced with ONTAP 9.13.

Schedules

Schedules can be used to automate the perform of tasks. This resource type was introduced with ONTAP 9.6.

Sensors

You can use these endpoints to retrieve details about all the platform environment sensors. This resource type was introduced with ONTAP 9.11.

Software

An ONTAP cluster includes the cluster software profile, software packages collection, and software history collection. This resource type was introduced with ONTAP 9.6 and updated with ONTAP 9.8.

Web

You can use these endpoints to update the web services configurations and to retrieve the current configuration. This resource type was introduced with ONTAP 9.10.

Name services

You can use these API calls to manage the name services supported by ONTAP.

Cache

ONTAP name services supports caching which improves performance and resiliency. Configuration of the name services cache can now be access through the REST API. Settings can be applied at multiple levels including: hosts, unix-users, unix-groups, and netgroups. This resource type was introduced with ONTAP 9.11.

DDNS

You can display the Dynamic DNS (DDNS) information and manage the DDNS subsystem. This resource type is new with ONTAP 9.8.

DNS

DNS supports the integration of the ONTAP cluster in your network. This resource type was introduced with ONTAP 9.6 and enhanced with ONTAP 9.13.

Host record

These endpoints allow you to displays the IP address of a specified hostname as well as the hostname for an IP address. This resource type was introduced with ONTAP 9.10.

LDAP

LDAP servers can be used to maintain user information. This resource type was introduced with ONTAP 9.6.

LDAP schemas

You can create, modify, and list the LDAP schemas used by ONTAP. There are four default schemas included. This resource type was introduced with ONTAP 9.11.

Local hosts

You can use these endpoints to display and manage the local mappings for hostnames. This resource type was introduced with ONTAP 9.10.

Name mappings

Name mappings allow you to map identities from one name domain to another. For example, you can map identities from CIFS to UNIX, Kerberos to UNIX, and UNIX to CIFS. This resource type was introduced with ONTAP 9.6.

Netgroup files

You can retrieve netgroup file details and delete a file for an SVM. This resource type was introduced with ONTAP 9.11.

NIS

NIS servers can be used to authenticate users and client workstations. This resource type was introduced with ONTAP 9.6.

UNIX users and groups

Local UNIX users and groups have been a part of previous ONTAP releases. However, support has now been added to the REST API allowing you to display and manage the users and groups. These REST resource types were introduced with ONTAP 9.9 and significantly enhanced with ONTAP 9.10.

NAS

You can use these API calls to manage the CIFS and NFS settings for the cluster and SVMs.

Active Directory

You can manage the Active Directory accounts defined for an ONTAP cluster. This includes creating new accounts as well as displaying, updating, and deleting accounts. This support was added with ONTAP 9.12.

Audit

Certain CIFS and NFS events can be logged for the SVMs, which can help to improve security. This resource type was introduced with ONTAP 9.6.

Audit log redirect

You can redirect NAS auditing events to a specific SVM. This resource type is new with ONTAP 9.8.

CIFS connections

You can retrieve a list of the established CIFS connections. This resource type was introduced with ONTAP 9.11.1.

CIFS domains

Support for CIFS domains has been added at the cluster and SVM level with several categories of endpoints. You can retrieve the domain configuration as well as create and remove preferred domain controllers. This resource type was introduced with ONTAP 9.10 and enhanced with ONTAP 9.13.

CIFS group policies

Endpoints has been added to support the creation and management of CIFS group policies. The configuration information is available and administered through group policy objects that are applied to all or specific SVMs. This support was added with ONTAP 9.12.

CIFS home directory search paths

Home directories for SMB users on a CIFS server can be created without creating an individual SMB share for each user. The home directory search path is a set of absolute paths from the root of an SVM. This resource type was introduced with ONTAP 9.6.

CIFS local groups

The CIFS server can use local groups for authorization when determining share, file, and directory access rights. This resource type was introduced with ONTAP 9.9 and significantly expanded with ONTAP 9.10.

CIFS NetBIOS

You can display information about the NetBIOS connections for the cluster. Details include the IP addresses and registered NetBIOS names. This information can help you to troubleshoot name resolution issues. This resource type was introduced with ONTAP 9.11.1.

CIFS services

The core configuration of the CIFS server. This resource type was introduced with ONTAP 9.6 and updated with ONTAP 9.7.

CIFS session files

You can retrieve a list of open files for the CIFS sessions based on several filtering options. This resource type was introduced with ONTAP 9.11.1.

CIFS sessions

You can use this API to retrieve detailed information about a CIFS session. This resource type was introduced with the ONTAP 9.8 REST API and enhanced with ONTAP 9.9.

CIFS shadow copies

Microsoft Remote Volume Shadow Copy Services is an extension of the existing Microsoft VSS functionality. It extends the VSS capability to support shadow copying of SMB shares. This feature is now available through the ONTAP REST API. This resource type was introduced with ONTAP 9.11.1.

CIFS shares

The SMB shares defined at a CIFS server. This resource type was introduced with ONTAP 9.6.

CIFS shares ACLs

The access control lists (ACLs) controlling access to folders and files on the CIFS shares. This resource type was introduced with ONTAP 9.6.

CIFS UNIX symlink mapping

Both CIFS and UNIX clients can access the same datastore. When UNIX clients create symbolic links, these mappings provide a reference to another file or folder to support the CIFS clients. This resource type was introduced with ONTAP 9.6.

CIFS user and group bulk import

You can use the new REST API endpoints to perform a bulk import of the CIFS local users, groups, and group membership information as well as monitor the status of the request. This resource type was introduced with ONTAP 9.11.1.

File access tracing

You can use these API calls to trace access to specific files. This resource type is new with ONTAP 9.8.

File security permissions

You can use these API calls displays the effective permission granted to Windows or Unix user for a specific file or folder. You can also manage NTFS file security and audit policies. This resource type was introduced with the ONTAP 9.8 REST API and significantly enhanced with ONTAP 9.9.

FPolicy

FPolicy is a file access notification framework used to monitor and manage file access events on the SVMs. This resource type was introduced with ONTAP 9.6.

FPolicy connections

These endpoints allow you to display and update connection status information for external FPolicy servers. This resource type was introduced with ONTAP 9.10.

FPolicy engines

The FPolicy engines allow you to identify the external servers that receive the file access notifications. This resource type was introduced with ONTAP 9.6.

FPolicy events

The configuration identifying how file access is monitored and what events are generated. This resource type was introduced with ONTAP 9.6.

FPolicy persistent store

You can configure and administer a persistent store for the ONTAP FPolicy configuration and events. Each SVM can have one persistent store which is shared for the multiple policies within the SVM. This resource type was introduced with ONTAP 9.14.

FPolicy policies

A container for elements of the FPolicy framework, including FPolicy engines and events. This resource type was introduced with ONTAP 9.6.

Locks

A lock is a synchronization mechanism for enforcing limits on concurrent access to files where many clients are accessing the same file simultaneously. You can use these endpoints to retrieve and delete locks. This resource type was introduced with ONTAP 9.10.

NFS connected client maps

The NFS map information for the connected clients is available through the new endpoint. You can retrieve details about the node, SVM, and IP addresses. This resource type was introduced with ONTAP 9.11.1.

NFS connected clients

You can display a list of connected clients with the details of their connection. This resource type was introduced with ONTAP 9.7.

NFS export policies

The policies including rules that describe the NFS exports. This resource type was introduced with ONTAP 9.6.

NFS Kerberos interfaces

The configuration settings for an interface to Kerberos. This resource type was introduced with ONTAP 9.6.

NFS Kerberos realms

The configuration settings for Kerberos realms. This resource type was introduced with ONTAP 9.6.

NFS services

The core configuration of the NFS server. This resource type was introduced with ONTAP 9.6 and updated with ONTAP 9.7.

Object store

Auditing of the S3 events is a security improvement allowing you to track and log certain S3 events. An S3 audit event selector can be set on a per SVM per bucket basis. This resource type was introduced with ONTAP 9.10.

Vscan

A security feature to protect your data from viruses and other malicious code. This resource type was introduced with ONTAP 9.6.

Vscan on-access policies

The Vscan policies allowing files objects to be actively scanned when accessed by a client. This resource type was introduced with ONTAP 9.6.

Vscan on-demand policies

The Vscan policies allowing files objects to be immediately scanned on demand or according to a set schedule. This resource type was introduced with ONTAP 9.6.

Vscan scanner pools

A set of attributes used to manage the connection between ONTAP and an external virus-scanning server. This resource type was introduced with ONTAP 9.6.

Vscan server status

The status of the external virus-scanning server. This resource type was introduced with ONTAP 9.6.

NDMP

You can use these API calls to manage the NDMP services.

NDMP mode

The NDMP operational mode can be SVM scope or node scope. This resource type was introduced with ONTAP 9.7.

NDMP nodes

You can manage the NDMP configuration of the nodes. This resource type was introduced with ONTAP 9.7.

NDMP sessions

You can retrieve and delete NDMP session details for a specific SVM or node. This resource type was introduced with ONTAP 9.7.

NDMP SVMs

You can manage the NDMP configuration of the SVMs. This resource type was introduced with ONTAP 9.7.

NDMP SVM user passwords

You can generate and retrieve passwords for a specific NDMP user within the SVM content. This resource type was introduced with the ONTAP 9.8 REST API and enhanced with ONTAP 9.9.

Networking

You can use these API calls to manage the physical and logical networking resources used with the cluster.

BGP peer groups

You can create and administer Border Gateway Protocol peer groups. This resource type was introduced with ONTAP 9.7.

Ethernet broadcast domains

An Ethernet broadcast domain is a set of physical ports that appear to be part of the same physical network. All the ports receive a packet when broadcast from one of the ports in the domain. Each broadcast domain is part of an IPspace. This resource type was introduced with ONTAP 9.6.

Ethernet ports

An Ethernet port is a physical or virtual networking endpoint. The ports can be combined into a Link Aggregate Group (LAG) or separated using a Virtual LAN (VLAN). This resource type was introduced with ONTAP 9.6 and updated with ONTAP 9.8.

Ethernet switch ports

You can retrieve the port information for an Ethernet switch. This resource type is new with ONTAP 9.8.

Ethernet switches

You can retrieve or modify the configuration for Ethernet switches used for the ONTAP cluster or storage network. This resource type is new with ONTAP 9.8 and updated with 9.11.

Fibre Channel fabrics

You can use the Fibre Channel (FC) fabric REST API endpoints to retrieve information about the FC network. This includes the connections between the ONTAP cluster and the FC fabric, the switches comprising the fabric, and the zones of the active zoneset. This resource type was introduced with ONTAP 9.11.

Fibre Channel interfaces

A Fibre Channel interface is a logical endpoint associated with an SVM. This resource type was introduced with ONTAP 9.6 and updated with ONTAP 9.8. Support for retrieving performance metrics data was added with ONTAP 9.14.

Fibre Channel ports

A Fibre Channel port is a physical adapter on an ONTAP node used to connect to the Fibre Channel network. This resource type was introduced with ONTAP 9.6 and updated with ONTAP 9.8. Support for retrieving performance metrics data was added with ONTAP 9.14.

HTTP proxy

You can configure an HTTP proxy for either an SVM or a cluster IPSpace. This resource type was introduced with ONTAP 9.7.

IP interfaces

A logical interface (LIF) is an IP address with additional configuration attributes. This resource type was introduced with ONTAP 9.6 and updated with ONTAP 9.8.

IP routes

A routing table is a collection of IP routes used to forward traffic to its destination. This resource type was introduced with ONTAP 9.6.

IP service policies

The IP service polices define the services available at a specific LIF. Service policies can be configured within the context of an SVM or IPspace. This resource type was introduced with ONTAP 9.6 and updated with ONTAP 9.8.

IP subnets

The ONTAP networking capability has been expanded to support IP subnets. The REST API provides access to the configuration and management of the IP subnets within an ONTAP cluster. This resource type was introduced with ONTAP 9.11.

IPspaces

An IPspace creates a networking space to support one or more SVMs. The IPspaces can be isolated from each other, providing security and privacy. This resource type was introduced with ONTAP 9.6.

NVMe

You can use these API calls to manage resources supporting non-volatile memory express (NVMe).

Fibre Channel logins

Fibre Channel logins represent connections formed by Fibre Channel initiators logged in to ONTAP. This resource type was introduced with ONTAP 9.6.

Namespaces

An NVMe namespace is a collection of addressable logical blocks presented to hosts connected to the SVM using the NVMe over Fabrics protocol. This resource type was introduced with ONTAP 9.6 and updated with ONTAP 9.8. Support for retrieving performance metrics data was added with ONTAP 9.14.

NVMe interfaces

NVMe interfaces are the network interfaces configured to support the NVMe over Fabrics (NVMe-oF) protocol. This resource type was introduced with ONTAP 9.6.

NVMe services

An NVMe service defines the properties of the NVMe controller target for an SVM. This resource type was introduced with ONTAP 9.6 and updated with ONTAP 9.7. Support for retrieving performance metrics data was added with ONTAP 9.14.

NVMe subsystem controllers

The NVMe subsystem controllers represent dynamic connections between hosts and a storage solution. This resource type was introduced with ONTAP 9.6.

NVMe subsystem maps

An NVMe subsystem map is an association of an NVMe namespace with an NVMe subsystem. This resource type was introduced with ONTAP 9.6.

NVMe subsystems

An NVMe subsystem maintains configuration state and namespace access control for a set of NVMe-connected hosts. This resource type was introduced with ONTAP 9.6.

Object store

You can use these API calls to access S3-based object storage.

Buckets

A bucket is a container of objects and is structured using an object name space. Each S3 object server can have multiple buckets. This resource type was introduced with ONTAP 9.7 and updated with ONTAP 9.8.

Services

You can create and manage the ONTAP S3 configuration, including servers and bucket configurations. This resource type was introduced with ONTAP 9.7.

Service buckets

A bucket is a container of objects and is structured using an object name space. You can manage the buckets for a specific S3 server. This resource type was introduced with ONTAP 9.7.

S3 bucket rules

The S3 buckets can include a rule definition. Each rule is a list objects and defines the set of actions to be performed on an object within the bucket. This resource type was introduced with ONTAP 9.13.

S3 groups

You can create groups of S3 users and manage access control at the group level. This resource type is new with ONTAP 9.8.

S3 policies

You can create an S3 policy and associate it with a resource to define various permissions. This resource type is new with ONTAP 9.8.

Users

The S3 user accounts are maintained at the S3 server. User accounts are based on a pair of keys and associated with the buckets they control. This resource type was introduced with ONTAP 9.7.

SAN

You can use these API calls to manage storage area networking (SAN) resources.

Fibre Channel logins

Fibre Channel logins represent connections formed by Fibre Channel initiators that have logged in to ONTAP. This resource type was introduced with ONTAP 9.6.

Fibre Channel Protocol services

A Fibre Channel Protocol (FCP) service defines the properties of a Fibre Channel target for an SVM. This resource type was introduced with ONTAP 9.6 and updated with ONTAP 9.7. Support for retrieving performance metrics data was added with ONTAP 9.14.

Fibre Channel WWPN aliases

A world wide port name (WWPN) is a 64-bit value uniquely identifying a Fibre Channel port. This resource type was introduced with ONTAP 9.6.

igroups

An initiator group (igroup) is a collection of Fibre Channel WWPNs (world wide port names), and iSCSI IQNs (qualified names), and iSCSI EUIs (extended unique identifiers) that identify host initiators. This resource type

was originally introduced with ONTAP 9.6.

Nested igroups is a new feature with ONTAP 9.9 and support has also been added to the REST API. This REST resource type was introduced with ONTAP 9.9.

Initiators

An initiator is a Fibre Channel (FC) world wide port name (WWPN), an iSCSI Qualified Name (IQN), or an iSCSI EUI (Extended Unique Identifier) that identifies a host endpoint. You can retrieve initiators for the cluster or a specific SVM. This resource type was introduced with ONTAP 9.14.

iSCSI credentials

The iSCSI credentials object contains authentication credentials which are used by an initiator and ONTAP. This resource type was introduced with ONTAP 9.6.

iSCSI services

An iSCSI service defines the properties of the iSCSI target for an SVM. This resource type was introduced with ONTAP 9.6 and updated with ONTAP 9.7. Support for retrieving performance metrics data was added with ONTAP 9.14.

iSCSI sessions

An iSCSI session is one or more TCP connections that link an iSCSI initiator with an iSCSI target. This resource type was introduced with ONTAP 9.6.

LUN attributes

LUN attributes are caller-defined name/value pairs that can be optionally stored with a LUN. Attributes are available to save small amounts of application-specific metadata and are not interpreted by ONTAP. The endpoints allow you to create, update, delete, and discover attributes for a LUN. This resource type was introduced with ONTAP 9.10.

LUN maps

A LUN map is an association between a LUN and an initiator group. This resource type was introduced with ONTAP 9.6.

LUN maps reporting nodes

The reporting nodes are the cluster nodes from which network paths to a mapped LUN are advertised using the SAN protocols as part of the selective LUN map (SLM) feature of ONTAP. The new endpoints allow you to add, remove, and discover the reporting nodes of a LUN map. This resource type was introduced with ONTAP 9.10.

LUNs

A LUN is the logical representation of storage in a storage area network (SAN). This resource type was introduced with ONTAP 9.6 and updated with ONTAP 9.7. Support for retrieving performance metrics data was added with ONTAP 9.14.

Port sets

A port set is a collection of Fibre Channel or iSCSI network interfaces associated with the *portset* Storage VM. While this feature has existed with previous releases of ONTAP, support has now been added to the REST API. This REST resource type was introduced with ONTAP 9.9.

vVol bindings

A VMware virtual volume (vVol) binding is an association between a LUN of class `protocol_endpoint` and a LUN of class `vvol`. The vVol binding REST API allows you to create, delete, and discover vVol bindings. This resource type was introduced with ONTAP 9.10.

Security

You can use these API calls to manage the cluster and SVM security settings.

Accounts

There is a collection of user accounts for the cluster and SVMs. This resource type was introduced with ONTAP 9.6.

Accounts name

The configuration for a scoped user account. This resource type was introduced with ONTAP 9.6.

Active Directory proxy

You can administer the SVM account information at the Active Directory server. This resource type was introduced with ONTAP 9.7.

Anti-ransomware

ONTAP detects files potentially containing a ransomware threat. There are several categories of endpoints. You can retrieve a list of these suspect files as well as remove them from a volume. This resource type was introduced with ONTAP 9.10.1.

Audit

The settings which determine what is logged to the audit log files. This resource type was introduced with ONTAP 9.6.

Audit destinations

These settings control how audit log information is forwarded to remote systems or splunk servers. This resource type was introduced with ONTAP 9.6.

Audit messages

You can retrieve the audit log messages. This resource type was introduced with ONTAP 9.6.

AWS KMS

Amazon Web Services includes a key management service that provides secure storage for keys and other secrets. You can access this service through the REST API to allow ONTAP to securely store its encryption keys in the cloud. In addition, you can create and list the authentication keys used with NetApp Storage Encryption. This support is new with ONTAP 9.12.

Azure Key Vault

This set of API calls allows you to use the Azure Key Vault to store the ONTAP encryption keys. This resource type is new with ONTAP 9.8.

Certificates

The APIs calls can be used to install, display, and delete certificates used by ONTAP. This resource type was introduced with ONTAP 9.7.

Cisco Duo

Duo provides two-factor authentication for SSH logins. You can configure Duo to operate at the ONTAP cluster or SVM level. This resource type was introduced with ONTAP 9.14.

Cluster security

You can retrieve details of the cluster-wide security and update certain parameters. This resource type was introduced with ONTAP 9.7 and updated with ONTAP 9.8.

GCP KMS

This set of API calls allows you to use the Google Cloud Platform Key Management Service to store and manage the ONTAP encryption keys. This resource type was initially introduced with the ONTAP 9.8 REST API. However, this feature has been redesigned and so is considered to be new, with new resources types, in ONTAP 9.9.

IPSec

Internet Protocol Security (IPSec) is a suite of protocols providing security between two endpoints over an underlying IP network. This resource type is new with ONTAP 9.8.

IPSec CA certificates

You can add, remove, and retrieve IPSec CA certificates. This resource type is new with ONTAP 9.10.

IPSec policies

You can use this set of API calls to manage the policies in effect for an IPSec deployment. This resource type is new with ONTAP 9.8.

IPSec security associations

You can use this set of API calls to manage the security associations in effect for an IPSec deployment. This resource type is new with ONTAP 9.8.

Key manager configurations

These endpoints allow you to retrieve and update the configurations for key managers. This resource type is new with ONTAP 9.10.

Key managers

A key manager allows client modules within ONTAP to securely stored keys. This resource type was introduced with ONTAP 9.6 and updated for ONTAP 9.7. There was another update with ONTAP 9.12 to support authentication keys. A restore capability was added with ONTAP 9.13.

Key stores

A key store describes the type of a key manager. This resource type is new with ONTAP 9.10. Additional endpoints supporting enhanced control were added with ONTAP 9.14.

LDAP authentication

These API calls are used to retrieve and manage the cluster LDAP server configuration. This resource type was introduced with ONTAP 9.6.

Login messages

Used to display and manage the login messages used by ONTAP. This resource type was introduced with ONTAP 9.6.

Multiple administrator verification

The multiple administrator verification feature provides a flexible authorization framework for protecting access to ONTAP commands or operations. There are seventeen new endpoints that support defining, requesting, and approving access in the following areas:

- Rules
- Requests
- Approval groups

Providing the option for multiple administrators to approve access improves the security of your ONTAP and IT environments. These resource types were introduced with ONTAP 9.11.

NIS authentication

These settings are used to retrieve and manage the cluster NIS server configuration. This resource type was introduced with ONTAP 9.6.

OAuth 2.0

Open Authorization (OAuth 2.0) is a token-based framework that can be used to restrict access to your ONTAP storage resources. You can use it with clients that access ONTAP through the REST API. Configuration can be performed with any of the ONTAP administrative interfaces including the REST API. This resource type was introduced with ONTAP 9.14.

Password authentication

This includes the API call used to change the password for a user account. This resource type was introduced with ONTAP 9.6.

Privileges for a role instance

Manage the privileges for a specific role. This resource type was introduced with ONTAP 9.6.

Public key authentication

You can use these API calls to configure the public keys for user accounts. This resource type was introduced with ONTAP 9.7.

Roles

The roles provide a way to assign privileges to user accounts. This resource type was introduced with ONTAP 9.6.

Roles instance

Specific instance of a role. This resource type was introduced with ONTAP 9.6.

SAML service provider

You can display and manage the configuration for the SAML service provider. This resource type was introduced with ONTAP 9.6.

SSH

These calls allow you to set the SSH configuration. This resource type was introduced with ONTAP 9.7.

SSH SVMs

These endpoints allow you to retrieve the SSH security configuration for all SVMs. This resource type was introduced with ONTAP 9.10.

TOTPS

You can use the REST API to configure time-based one-time password (TOTP) profiles for accounts that sign in and access ONTAP using SSH. This resource type was introduced with ONTAP 9.13.

SnapLock

You can use these API calls to administer the ONTAP SnapLock feature.

Log

The SnapLock log structure is based on directories and files on a specific volume which contain the log

records. Log files are filled and archived based on the maximum log size. This resource type was introduced with ONTAP 9.7.

Compliance clock

The compliance clock determines the expiration time of the SnapLock objects. The clock must be initialized outside of the REST API and cannot be changed. This resource type was introduced with ONTAP 9.7.

Event retention

You can use the SnapLock Event Based Retention (EBR) feature to define how long a file is retained after the occurrence of an event. This resource type was introduced with ONTAP 9.7.

File retention and privileged delete

You can manage the retention time of a file created by SnapLock. If needed, you can also delete unexpired WORM files on a SnapLock enterprise volume. This resource type was introduced with ONTAP 9.7.

The only built-in role with authority to execute the delete operation is vsadmin-snaplock.

File fingerprint

You can view and manage the core information describing files and volumes, such as type and expiration date. This resource type was introduced with ONTAP 9.7.

Legal hold

You can use these API calls to manage files that are part of a litigation process. This resource type was introduced with ONTAP 9.7.

SnapMirror

You can use these API calls to manage the SnapMirror data protection technology.

Policies

The SnapMirror policies are applied to relationships, and control the configuration attributes and behavior of each relationship. This resource type was introduced with ONTAP 9.6.

Relationships

Both asynchronous and synchronous relationships establish the connectivity needed transfer data. This resource type was introduced with ONTAP 9.6.

Relationships transfers

You can manage the SnapMirror transfers over existing SnapMirror relationships. This resource type was introduced with ONTAP 9.6.

Storage

You can use these API calls to manage the physical and logical storage.

Aggregate metrics

You can retrieve historical metrics data for a specific aggregate. This resource type was introduced with ONTAP 9.6 and updated with ONTAP 9.7.

Aggregate plexes

A physical copy of the WAFL storage within an aggregate. This resource type was introduced with ONTAP 9.6.

Aggregates

An aggregate consists of one or more RAID groups. This resource type was introduced with ONTAP 9.6.

Bridges

You can retrieve the bridges in a cluster. This resource type was introduced with ONTAP 9.9.

Disks

The physical disks in the cluster. This resource type was introduced with ONTAP 9.6 and updated with ONTAP 9.7 and 9.8.

File clone

You can use these endpoints to create file clones, retrieve split status, and manage split loads. The file cloning endpoint resources were first introduced with ONTAP 9.6 and expanded with ONTAP 9.8. They were significantly expanded again with ONTAP 9.10.

File moves

You can use these REST API endpoints to move a file between two FlexVol volumes or within a FlexGroup volume. After the request is accepted you can monitor the progress and status. This resource type was introduced with ONTAP 9.11.1.

FlexCache

This resource type was introduced with ONTAP 9.6 and updated with ONTAP 9.8.

FlexCache origins

FlexCache is a persistent cache of an origin volume. This resource type was originally introduced with ONTAP 9.6. Support has been enhanced with the ONTAP 9.9 REST API to support modification through the HTTP PATCH method.

Monitored files

You can designate specific files for additional monitoring. This resource type is new with ONTAP 9.8.

Pools

You can create a shared storage pool as well as retrieve the storage pools in a cluster. This resource type was introduced with ONTAP 9.11.1.

Ports

Storage ports of the cluster. This resource type was introduced with ONTAP 9.6 and enhanced with ONTAP 9.11.1.

QOS policies

Quality of service policy configuration. This resource type was introduced with ONTAP 9.6.

QOS options

Endpoints have been introduced to allow you to retrieve and set QOS options for the cluster. For example, you can reserve a percentage of available system processing resources for background tasks. This resource type was introduced with ONTAP 9.14.

QOS workloads

A QOS workload represents a storage object tracked by QOS. You can retrieve the QOS workflows. This resource type was introduced with ONTAP 9.10.

Qtrees

You can use these API calls to management Qtrees, a type of logically divided file system. This resource type was introduced with ONTAP 9.6.

Quota reports

Report on quotas, which is a technique for restricting or tracking files or space usage. This resource type was introduced with ONTAP 9.6.

Quota rules

The rules used to enforce the quotas. This resource type was introduced with ONTAP 9.6 and updated with ONTAP 9.7.

Shelves

Shelves in the cluster. This resource type was introduced with ONTAP 9.6.

Snapshot policies

Snapshots are created based on policies. This resource type was introduced with ONTAP 9.6.

Snapshot schedules

You can control the snapshot schedules. This resource type is newly redesigned with ONTAP 9.8.

Switches

You can retrieve the switches in a cluster. This resource type was introduced with ONTAP 9.9.

Tape devices

You can retrieve the tape devices in a cluster. This resource type was introduced with ONTAP 9.9.

Top metrics

The top metrics endpoints allow you to determine activity for a volume filtered by a specific metric. Filtering can be done based on clients, directories, files, and users. This resource type was introduced with ONTAP 9.10.

Volume efficiency policies

You can use these API calls to configure the efficiencies applied to an entire volume. This resource type is new with ONTAP 9.8.

Volumes

Logical containers are used to serve data to clients. This resource type was originally introduced with ONTAP 9.6 REST API. Many of the parameter values used with the API were significantly expanded with ONTAP 9.9 including those used with space management.

Volume files

You can retrieve a list of files and directories for a specific directory on a volume. This resource type was introduced with ONTAP 9.7 and updated with ONTAP 9.8.

Volumes Snapshots

Snapshots for a volume. This resource type was introduced with ONTAP 9.6.

Support

You can use these API calls to manage the ONTAP features used to support a cluster.

Application log

A standalone application can record EMS events and optionally generated AutoSupport packages at an

ONTAP system by issuing a POST request. This resource type was introduced with ONTAP 9.11.1

Automatic update

The automatic update feature keeps your ONTAP systems current by downloading and applying the latest software updates. There are several endpoint categories to support the feature, including status, configurations, and updates. These resource types were introduced with ONTAP 9.10.

AutoSupport

AutoSupport collects configuration and status details as well as errors, and reports the information to NetApp. This resource type was introduced with ONTAP 9.6.

AutoSupport messages

Each node maintains AutoSupport messages that can be generated and retrieved. This resource type was introduced with ONTAP 9.6.

Configuration backup

You can use these APIs to retrieve and update the current backup settings. This resource type was introduced with ONTAP 9.6.

Configuration backup operations

You can create, retrieve, and delete configuration backup files. This resource type was introduced with ONTAP 9.7.

Core dump

You can use these endpoints to retrieve and manage the memory core dumps generated by a cluster or node. This resource type was introduced with ONTAP 9.10.

EMS

The event management system (EMS) collects events and sends notifications to one or more destinations. This resource type was introduced with ONTAP 9.6.

EMS destinations

The EMS destinations determine how and where notifications are sent. This resource type was introduced with ONTAP 9.6.

EMS destinations instance

An EMS destination instance is defined by type and location. This resource type was introduced with ONTAP 9.6.

EMS events

This is a live collection of system events for the cluster. This resource type was introduced with ONTAP 9.6.

EMS filters

The EMS filters collectively identify the events that require additional processing. This resource type was introduced with ONTAP 9.6.

EMS filters instance

An EMS filter instance is a collection of rules that are applied to the events. This resource type was introduced with ONTAP 9.6.

EMS messages

Provides access to the EMS event catalog. This resource type was introduced with ONTAP 9.6.

EMS role configuration

The EMS support feature allows for the management of roles and the access control configuration assigned to the roles. This provides the ability to limit or filter the events and messages based on the role configuration. This resource type was introduced with ONTAP 9.13.

EMS rules for filter instance

A list of rules can be managed for a specific instance of an EMS filter. This resource type was introduced with ONTAP 9.6.

EMS rules instance for filter instance

An individual rule for a specific instance of an EMS filter. This resource type was introduced with ONTAP 9.6.

SNMP

You can enable and disable SNMP and trap operations for the cluster. This resource type was introduced with ONTAP 9.7.

SNMP trap host

An SNMP trap host is a system that is configured to receive SNMP traps from ONTAP. You can retrieve and define the hosts. This resource type was introduced with ONTAP 9.7.

SNMP trap host instance

You can manage specific SNMP trap hosts. This resource type was introduced with ONTAP 9.7.

SNMP users

You can define and administer SNMP users. This resource type was introduced with ONTAP 9.7.

SNMP users instance

You can administer a specific SNMP user where the engine ID is associated with the administrative SVM or a data SVM. This resource type was introduced with ONTAP 9.7.

SVM

You can use these API calls to manage storage virtual machines (SVMs).

Migrations

You can migrate an SVM from a source cluster to a destination cluster. The new endpoints provide complete control, including the ability to pause, resume, retrieve status, and abort a migration operation. This resource type was introduced with ONTAP 9.10.

Peer permissions

Peer permissions can be assigned which enable the SVM peering relationships. This resource type was introduced with ONTAP 9.6.

Peers

The peering relationships establish connectivity among the SVMs. This resource type was introduced with ONTAP 9.6.

SVMs

You can manage the SVMs that are bound to a cluster. This resource type was introduced with ONTAP 9.6.

Top metrics

You can access additional performance metrics data for a specific SVM instance. There are four lists available and each provides the top I/O activity for ONTAP FlexVol and FlexGroup volumes. The lists include:

- Clients
- Directories
- Files
- Users

These resource types were introduced with ONTAP 9.11.

Web

You can use these endpoints to update and retrieve the web services security configuration for each data SVM. This resource type was introduced with ONTAP 9.10.

Copyright information

Trademark information