RBAC security

ONTAP Automation

NetApp
November 14, 2022

Table of Contents

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.

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.

Sample role definitions

The ONTAP RBAC capability can be used in different ways depending on your environment. A few common scenarios are presented below. In each case the focus is on a specific security and administrative goal with an example of the corresponding role definition.

All the examples create and modify roles using `/api/security/roles` and the derived REST endpoints. For clarity, each of the curl commands refers to a separate JSON input file.

Limit access to SVM volume operations

You might want to restrict storage volume administration within an SVM. The example below illustrates this with a role that is first created and then optionally updated.

Create the initial role

A traditional role is created to initially allow access to all the major volume administration functions except cloning. The role presented below is defined with the following specific characteristics:

- Able to perform all CRUD volume operations including get, create, modify, and delete
- Cannot create a volume clone

curl example

```
curl --location -i --request POST
'https://10.63.56.136/api/security/roles' -u admin:password -k --header
'Accept: */*' --data @JSONinput
```

JSON input example

```
{
  "name": "role1",
  "owner": {
    "name": "cluster-1",
    "uuid": "852d96be-f17c-11ec-9d19-005056bbad91"
  },
  "privileges": [
      { "path": "volume create", "access": "all" },
      { "path": "volume delete", "access": "all" }
    ]
}
```

Update the role

The same role can be modified to allow the user to also create a volume clone.

curl example

```
curl --location -i --request POST
'https://10.63.56.136/api/security/roles/852d96be-f17c-11ec-9d19-
005056bbad91/role1/privileges' -u admin:password -k --header 'Accept: */*'
--data @JSONinput
```

JSON input example

```
{
  "path": "volume clone",
  "access": "all"
}
```

Data protection administration

In certain situations you may want to provide a user with limited data protection capabilities. The traditional role presented below is defined with the following characteristics:

- Able to create and delete snapshots as well as update SnapMirror relationships
- Cannot create or modify higher level objects such as volumes or SVMs

curl example

```
curl --location -i --request POST
'https://10.63.56.136/api/security/roles' -u admin:password -k --header
'Accept: */*' --data @JSONinput
```

JSON input example

```
{
  "name": "role1",
  "owner": {
    "name": "cluster-1",
    "uuid": "852d96be-f17c-11ec-9d19-005056bbad91"
  },
  "privileges": [
      {"path": "volume snapshot create", "access": "all"},
      {"path": "volume snapshot delete", "access": "all"},
      {"path": "volume show", "access": "readonly"},
      {"path": "vserver show", "access": "readonly"},
      {"path": "snapmirror show", "access": "readonly"},
      {"path": "snapmirror update", "access": "all"}
  ]
}
```

Generating ONTAP reports

You can create a REST role to provide users with the ability to generate ONTAP reports. The role presented below is defined with the following characteristics:

- Able to retrieve all storage object information related to capacity and performance (such as volume, qtree, LUN, aggregates, node, and SnapMirror relationships)
- Cannot create or modify higher level objects (such as volumes or SVMs)

curl example

```
curl --location -i --request POST
'https://10.63.56.136/api/security/roles' -u admin:password -k --header
'Accept: */*' --data @JSONinput
```

JSON input example

```
{
  "name": "rest_role1",
  "owner": {
    "name": "cluster-1",
    "uuid": "852d96be-f17c-11ec-9d19-005056bbad91"
  },
  "privileges": [
      {"path": "/api/storage/volumes", "access": "readonly"},
      {"path": "/api/storage/qtrees", "access": "readonly"},
      {"path": "/api/storage/luns", "access": "readonly"},
      {"path": "/api/storage/aggregates", "access": "readonly"},
      {"path": "/api/cluster/nodes", "access": "readonly"},
      {"path": "/api/snapmirror/relationships", "access": "readonly"},
      {"path": "/api/svm/svms", "access": "readonly"}
  ]
}
```

Create a user with a role

You can use the workflow described below to create a user with an associated REST role. Before reviewing the workflow, you should be familiar with the general preparation steps.

Prepare to create an ONTAP user with an assigned role

Before creating a role and assigning it to an ONTAP user account, you should first prepare by reviewing the major security requirements and options.

What ONTAP release are you using?

The ONTAP release determines what REST endpoints and RBAC features are available.

Identify the protected resources and scope

You need to identify the resources or commands to be protected and the scope (cluster or SVM).

What access should the user have?

After identifying the resources and scope, you need to determine the access level to be granted.

How will the users access ONTAP?

The user can access ONTAP through the REST API or CLI or both.

Is one of the built-in roles sufficient or is a custom role needed?

It is more convenient to use an existing built-in role but you can create a new custom role if needed.

What type of role is needed?

Based on the security requirements and the ONTAP access, you need to choose whether to create a REST or traditional role.

Create a user with a custom role

The workflow described below includes the typical steps needed to create a custom REST role and associate it with a new user account. Both the user and role have an SVM scope and are associated with a specific data SVM.

The workflow is meant to illustrate the complete process. Some of the steps may be optional or need to change based on your environment.

1. List the data SVMs in the cluster

Perform the following REST API call to list the SVMs in the cluster. The UUID and name of each SVM is provided in the output.

HTTP method	Path
GET	/api/svm/svms

curl example

```
curl --location -i --request GET
'https://10.222.81.101/api/svm/svms?order_by=name' -u admin:password -k
--header 'Accept: */*'
```

After you finish

Select the desired SVM from the list where you will create the new user and role.

2. List the users defined to the SVM

Perform the following REST API call to list the users defined in the SVM you selected. You can identify the SVM through the owner parameter.

HTTP method	Path
GET	/api/security/accounts

curl example

```
curl --location -i --request GET
'https://10.222.81.101/api/security/accounts/?owner.name=dmp' -u
admin:password -k --header 'Accept: */*'
```

After you finish

Based on the users already defined in the SVM, choose a unique name for the new user.

3. List the REST roles defined to the SVM

Perform the following REST API call to list the roles defined in the SVM you selected. You can identify the SVM through the owner parameter.

HTTP method	Path
GET	/api/security/roles

curl example

```
curl --location -i --request GET
'https://10.222.81.101/api/security/roles/?owner.name=dmp' -u
admin:password -k --header 'Accept: */*'
```

After you finish

Based on the roles already defined in the SVM, choose a unique name for the new role.

4. Create a custom REST role

Perform the following REST API call to a create a custom REST role in the SVM. The role initially has only one privilege which establishes a default access of **none** so that all access is denied.

HTTP method	Path
POST	/api/security/roles

JSON input example

```
{
  "name": "dprole1",
  "owner": {
    "name": "dmp",
    "uuid": "752d96be-f17c-11ec-9d19-005056bbad91"
  },
  "privileges": [
      {"path": "/api", "access": "none"},
  ]
}
```

curl example

```
curl --location -i --request POST
'https://10.222.81.101/api/security/roles' --data @JSONinput -u
admin:password -k --header 'Accept: */*'
```

After you finish

Optionally perform step 3 again to display the new role. You can also display the roles at the ONTAP CLI.

5. Update the role by adding more privileges

Perform the following REST API call to modify the role by adding privileges as needed.

HTTP method	Path
POST	/api/security/roles/{owner.uuid}/{name}/privileges

JSON input example

```
{
  "path": "/api/storage/volumes", "access": "readonly"}
}
```

curl example

```
curl --location -i --request POST
'https://10.222.81.101/api/security/roles/752d96be-f17c-11ec-9d19-
005056bbad91/dprole1/privileges' --data @JSONinput -u admin:password -k
--header 'Accept: */*'
```

After you finish

Optionally perform step 3 again to display the new role. You can also display the roles at the ONTAP CLI.

6. Create a user

Perform the following REST API call to a create a user account. The role `dprole1` created above is associated with the new user.

You can include the user without a role. In this case you'll need to modify the user to assign a role.

HTTP method	Path
POST	/api/security/accounts

JSON input example

```
{
  "owner": {"uuid":"daf84055-248f-11ed-a23d-005056ac4fe6"},
  "name": "david",
  "applications": [
      {"application":"ssh",
       "authentication_methods":["password"],
       "second_authentication_method":"none"}
  ],
  "role":"dprole1",
  "password":"netapp123"
}
```

curl example

```
curl --location -i --request POST
'https://10.222.81.101/api/security/accounts' --data @JSONinput -u
admin:password -k --header 'Accept: */*'
```

After you finish

You can sign in to the SVM management interface using the credentials for the new user.

Copyright information

Trademark information