Client authorization

ONTAP 9

NetApp
January 23, 2026

Table of Contents

Client authorization

Overview and options for ONTAP client authorization

The ONTAP OAuth 2.0 implementation is designed to be flexible and robust, providing the features you need to secure your ONTAP environment. There are several mutually exclusive configuration options available. The authorization decisions are ultimately based on the ONTAP REST roles either contained in or derived from the OAuth 2.0 access tokens.

You can only use ONTAP REST roles when configuring authorization for OAuth 2.0. The earlier ONTAP traditional roles are not supported.

ONTAP applies the single most appropriate authorization option based on your configuration. See How ONTAP determines access for more about how ONTAP makes client access decisions.

OAuth 2.0 self-contained scopes

These scopes contain one or more custom REST roles, each encapsulated within a single string in the access token. They are independent of the ONTAP role definitions. You need to configure the scope strings at your authorization server. See Self-contained OAuth 2.0 scopes for more information.

Local ONTAP REST roles

A single named REST role, either builtin or custom, can be used. The scope syntax for a named role is **ontap-role-**<URL-encoded-ONTAP-role-name>. For example, if the ONTAP role is `admin` the scope string will be `ontap-role-admin`.

Users

The username in the access token defined with access to the application "http" can be used. A user is tested in the following order based on the defined authentication method: password, domain (Active Directory), nsswitch (LDAP).

Groups

The authorization servers can be configured to use ONTAP groups for authorization. If the local ONTAP definitions are examined but no access decision can be made, the Active Directory ("domain") or LDAP ("nsswitch") groups are used. Group information can be specified in one of two ways:

- OAuth 2.0 scope string

  Supports confidential applications using the client credentials flow where there is no user with a group membership. The scope should be named **ontap-group-**<URL-encoded-ONTAP-group-name>. For example, if the group is "development" the scope string will be "ontap-group-development".

- In the "group" claim

  This is intended for access tokens issued by ADFS using the resource owner (password grant) flow.

See Working with OAuth 2.0 or SAML IdP groups in ONTAP for more information.

Self-contained OAuth 2.0 scopes in ONTAP

Self-contained scopes are strings carried in the access token. Each is a complete custom role definition and includes everything ONTAP needs to make an access decision. The scope is separate and distinct from any of the REST roles defined within ONTAP itself.

Format of the scope string

At a base level, the scope is represented as a contiguous string and composed of six colon-separated values. The parameters used in the scope string are described below.

ONTAP literal

The scope must begin with the literal value `ontap` in lowercase. This identifies the scope as specific to ONTAP.

Cluster

This defines which ONTAP cluster the scope applies to. The values can include:

- Cluster UUID

  Identifies a single cluster.

- Asterisk (*)

  Indicates the scope applies to all clusters.

You can use the ONTAP CLI command `cluster identity show` to display the UUID of your cluster. If not specified, the scope applies to all clusters. Learn more about `cluster identity show` in the ONTAP command reference.

Role

The name of the REST role contained in the self-contained scope. This value is not examined by ONTAP or matched to any existing REST roles defined to ONTAP. The name is used for logging.

Access level

This value indicates the access level applied to the client application when using the API endpoint in the scope. There are six possible values as described in the table below.

Access level	Description
none	Denies all access to the specified endpoint.
readonly	Allows only read access using GET.
read_create	Allows read access as well as the creation of new resource instances using POST.
read_modify	Allows read access as well as the ability to update existing resources using PATCH.

Access level	Description
read_create_modify	Allows all access except delete. The allowed operations include GET (read), POST (create), and PATCH (update).
all	Allows full access.

SVM

The name of the SVM within the cluster the scope applies to. Use the * value (asterisk) to indicate all SVMs.

This feature is not fully supported with ONTAP 9.14.1. You can ignore the SVM parameter and use an asterisk as a placeholder. Review the ONTAP release notes to check for future SVM support.

REST API URI

The complete or partial path to a resource or set of related resources. The string must begin with `/api`. If you don't specify a value, the scope applies to all API endpoints at the ONTAP cluster.

Scope examples

A few examples of self-contained scopes are presented below.

ontap:*:joes-role:read_create_modify:*:/api/cluster

Provides the user assigned this role read, create, and modify access to the `/cluster` endpoint.

CLI administrative tool

To make the administration of the self-contained scopes easier and less error-prone, ONTAP provides the CLI command `security oauth2 scope` to generate scope strings based on your input parameters.

The command `security oauth2 scope` has two use cases based on your input:

- CLI parameters to scope string

  You can use this version of the command to generate a scope string based on the input parameters.

- Scope string to CLI parameters

  You can use this version of the command to generate the command parameters based on the input scope string.

Example

The following example generates a scope string with the output included after the command example below. The definition applies to all clusters.

```
security oauth2 scope cli-to-scope -role joes-role -access readonly -api
/api/cluster
```

```
ontap:*:joes-role:readonly:*:/api/cluster
```

Learn more about `security oauth2 scope` in the ONTAP command reference.

OAuth 2.0 external role mapping in ONTAP

An external role is defined at an identify provider configured for use by ONTAP. You can create and administer mapping relationships between these external roles and the ONTAP roles using the ONTAP CLI.

You can also configure the external role mapping feature using the ONTAP REST API. Learn more in the ONTAP automation documentation.

External roles in an access token

Here's a fragment of a JSON access token containing two external roles.

```
...
"appidacr": "1",
"family_name": "User",
"name": "Test User 1",
"oid": "4c2215c7-6d52-40a7-ce71-096fa41379ba",
"roles": [
  "Global Administrator",
  "Application Administrator"
],
"ver": "1.0",
...
```

Configuration

You can use the ONTAP command line interface to administer the external role mapping feature.

Create

You can define a role mapping configuration with the `security login external-role-mapping create` command. You need to be at the ONTAP **admin** privilege level to issue this command as well as the related options.

Parameters

The parameters used to create a group mapping are described below.

Parameter	Description
`external-role`	The name of the role defined at the external identity provider.
`provider`	The name of the identity provider. This should be the identifier for the system.
`ontap-role`	Indicates the existing ONTAP role the external role is mapped to.

Example

```
security login external-role-mapping create -external-role "Global
Administrator" -provider entra -ontap-role admin
```

Learn more about `security login external-role-mapping create` in the ONTAP command reference.

Additional CLI operations

The command supports several additional operations, including:

- Show
- Modify
- Delete

Related information

- ONTAP command reference

How ONTAP determines client access

To properly design and implement OAuth 2.0, you need to understand how your authorization configuration is used by ONTAP to make access decisions for the clients. The major steps used to determine access are presented below based on the ONTAP release.

There were no significant OAuth 2.0 updates with ONTAP 9.15.1. If you are using the 9.15.1 release, refer to the description for ONTAP 9.14.1.

Related information

- OAuth 2.0 features supported in ONTAP

ONTAP 9.16.1

ONTAP 9.16.1 expands the standard OAuth 2.0 support to include Microsoft Entra ID specific extensions for native Entra ID groups as well as external role mapping.

Determine client access for ONTAP 9.16.1

Step 1: Self-contained scopes

If the access token contains any self-contained scopes, ONTAP examines these scopes first. If there are no self-contained scopes, go to step 2.

With one or more self-contained scopes present, ONTAP applies each scope until an explicit **ALLOW** or **DENY** decision can be made. If an explicit decision is made, processing ends.

If ONTAP can't make an explicit access decision, continue to step 2.

Step 2: Check the local roles flag

ONTAP examines the boolean parameter `use-local-roles-if-present`. The value of this flag is set separately for each authorization server defined to ONTAP.

- If the value is `true` continue to step 3.
- If the value is `false` processing ends and access is denied.

Step 3: Named ONTAP REST role

If the access token contains a named REST role in the `scope` or `scp` field, or as a claim, ONTAP uses the role to make the access decision. This always results in an **ALLOW** or **DENY** decision and processing ends.

If there is no named REST role or the role is not found, continue to step 4.

Step 4: Users

Extract the username from the access token and attempt to match it to users that have access to the application "http". The users are examined based on the authentication method in the following order:

- password
- domain (Active Directory)
- nsswitch (LDAP)

If a matching user is found, ONTAP uses the role defined for the user to make an access decision. This always result in an **ALLOW** or **DENY** decision and processing ends.

If a user is not matched or if there's no username in the access token, continue to step 5.

Step 5: Groups

If one or more groups are included, the format is examined. If the groups are represented as UUIDs, an internal group mapping table is searched. If there's a group match and an associated role, ONTAP uses the role defined for the group to make an access decision. This always result in an **ALLOW** or **DENY** decision and processing ends. For more information see Working with OAuth 2.0 or SAML IdP groups in ONTAP.

If groups are represented as names and configured with domain or nsswitch authorization, ONTAP attempts to match them to an Active Directory or LDAP group, respectively. If there's a group match, ONTAP uses the role defined for the group to make an access decision. This always result in an **ALLOW** or **DENY** decision and processing ends.

If there's no group match or if there's no group in the access token, access is denied and processing ends.

ONTAP 9.14.1

Initial OAuth 2.0 supported is introduced with ONTAP 9.14.1 based on the standard OAuth 2.0 features.

Determine client access for ONTAP 9.14.1

Step 1: Self-contained scopes

If the access token contains any self-contained scopes, ONTAP examines these scopes first. If there are no self-contained scopes, go to step 2.

With one or more self-contained scopes present, ONTAP applies each scope until an explicit **ALLOW** or **DENY** decision can be made. If an explicit decision is made, processing ends.

If ONTAP can’t make an explicit access decision, continue to step 2.

Step 2: Check the local roles flag

ONTAP examines the boolean parameter `use-local-roles-if-present`. The value of this flag is set separately for each authorization server defined to ONTAP.

- If the value is `true` continue to step 3.
- If the value is `false` processing ends and access is denied.

Step 3: Named ONTAP REST role

If the access token contains a named REST role in the `scope` or `scp` field, ONTAP uses the role to make the access decision. This always results in an **ALLOW** or **DENY** decision and processing ends.

If there is no named REST role or the role is not found, continue to step 4.

Step 4: Users

Extract the username from the access token and attempt to match it to users that have access to the application "http". The users are examined based on the authentication method in the following order:

- password
- domain (Active Directory)
- nsswitch (LDAP)

If a matching user is found, ONTAP uses the role defined for the user to make an access decision. This always result in an **ALLOW** or **DENY** decision and processing ends.

If a user is not matched or if there’s no username in the access token, continue to step 5.

Step 5: Groups

If one or more groups are included and configured with domain or nsswitch authorization, ONTAP attempts to match them to an Active Directory or LDAP group, respectively.

If there’s a group match, ONTAP uses the role defined for the group to make an access decision. This always result in an **ALLOW** or **DENY** decision and processing ends.

If there’s no group match or if there’s no group in the access token, access is denied and processing ends.

Copyright information

Trademark information