# RBAC

## ONTAP Automation

NetApp
October 21, 2024

# 目录

# RBAC

## 准备使用RBAC

根据您的环境、您可以通过多种不同的方式使用ONTAP RBAC功能。本节将以工作流的形式介绍一些常见情形。在每种情况下、重点都放在特定的安全和管理目标上。

在创建任何角色并将角色分配给ONTAP用户帐户之前、您应查看下面介绍的主要安全要求和选项、以做好准备。此外、请务必查看中的常规工作流概念 "准备使用这些工作流"。

您使用的是哪个**ONTAP** 版本?

ONTAP 版本可确定可用的REST端点和RBAC功能。

确定受保护的资源和范围

您需要确定要保护的资源或命令以及范围(集群或SVM)。

用户应具有哪些访问权限?

确定资源和范围后、您需要确定要授予的访问级别。

用户将如何访问**ONTAP** ?

用户可以通过REST API或CLI访问ONTAP 、也可以同时使用这两者。

内置角色之一是否足以满足要求、或者是否需要自定义角色?

使用现有内置角色更方便、但您可以根据需要创建新的自定义角色。

需要什么类型的角色?

根据安全要求和ONTAP 访问、您需要选择是创建REST角色还是传统角色。

## 创建角色

### 限制对**SVM**卷操作的访问

### 您可以定义一个角色来限制SVM中的存储卷管理。

关于此工作流

首先创建一个传统角色、以便最初允许访问除克隆以外的所有主要卷管理功能。此角色具有以下特征：

- 能够执行所有CRUD卷操作、包括获取、创建、修改和删除
- 无法创建卷克隆

然后、您可以根据需要更新此角色。在此工作流中、角色在第二步中进行了更改、以允许用户创建卷克隆。

第**1**步：创建角色

您可以通过问题描述调用来创建RBAC角色。

**HTTP方法和端点**

此REST API调用使用以下方法和端点。

| HTTP 方法 | 路径 |
| --- | --- |
| 发布 | /api/安全性/角色 |

**curl 示例**

```
curl --request POST \
--location "https://$FQDN_IP/api/security/roles" \
--include \
--header "Accept: */*" \
--header "Authorization: Basic $BASIC_AUTH" \
--data @JSONinput
```

**JSON 输入示例**

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

第**2**步：更新角色

您可以通过问题描述调用来更新现有角色。

**HTTP方法和端点**

此REST API调用使用以下方法和端点。

| HTTP 方法 | 路径 |
| --- | --- |
| 发布 | /api/安全性/角色 |

**CURL示例的其他输入参数**

除了所有REST API调用通用的参数之外、此步骤中的cURL示例还会使用以下参数。

| 参数 | Type | Required | Description |
| --- | --- | --- | --- |
| $SVM_ID | 路径 | 是的。 | 这是包含角色定义的SVM的UUID。 |

| 参数 | Type | Required | Description |
| --- | --- | --- | --- |
| $Role_name | 路径 | 是的。 | 这是要更新的SVM中的角色名称。 |

**curl** 示例

```
curl --request POST \
--location
"https://$FQDN_IP/api/security/roles/$SVM_ID/$ROLE_NAME/priveleges" \
--include \
--header "Accept: */*" \
--header "Authorization: Basic $BASIC_AUTH" \
--data @JSONinput
```

**JSON** 输入示例

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

## 启用数据保护管理

## 您可以为用户提供有限的数据保护功能。

关于此工作流

创建的传统角色具有以下特征：

- 能够创建和删除快照以及更新SnapMirror关系
- 无法创建或修改更高级别的对象、例如卷或SVM

**HTTP**方法和端点

此REST API调用使用以下方法和端点。

| **HTTP** 方法 | 路径 |
| --- | --- |
| 发布 | /api/安全性/角色 |

**curl** 示例

```
curl --request POST \
--location "https://$FQDN_IP/api/security/roles" \
--include \
--header "Accept: */*" \
--header "Authorization: Basic $BASIC_AUTH" \
--data @JSONinput
```

**JSON 输入示例**

```
{
  "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"}
  ]
}
```

## 允许生成ONTAP报告

## 您可以创建REST角色、使用户能够生成ONTAP 报告。

关于此工作流

创建的角色具有以下特征：

- 能够检索与容量和性能相关的所有存储对象信息(例如卷、qtree、LUN、聚合、节点、 和SnapMirror关系)
- 无法创建或修改更高级别的对象(例如卷或SVM)

**HTTP方法和端点**

此REST API调用使用以下方法和端点。

| HTTP 方法 | 路径 |
|---|---|
| 发布 | /api/安全性/角色 |

**curl 示例**

```
curl --request POST \
--location "https://$FQDN_IP/api/security/roles" \
--include \
--header "Accept: */*" \
--header "Authorization: Basic $BASIC_AUTH" \
--data @JSONinput
```

**JSON 输入示例**

```
{
  "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"}
  ]
}
```

# 创建具有角色的用户

## 您可以使用此工作流创建具有关联REST角色的用户。

关于此工作流

此工作流包括创建自定义REST角色并将其与新用户帐户关联所需的典型步骤。用户和角色都具有SVM范围、并与特定数据SVM关联。某些步骤可能是可选的、也可能需要根据您的环境进行更改。

### 第1步：列出集群中的数据SVM

执行以下REST API调用以列出集群中的SVM。输出中提供了每个SVM的UUID和名称。

**HTTP方法和端点**

此REST API调用使用以下方法和端点。

| HTTP 方法 | 路径 |
| --- | --- |
| 获取 | /apI/SVM/SVM |

**curl 示例**

```
curl --request GET \
--location "https://$FQDN_IP/api/svm/svms?order_by=name" \
--include \
--header "Accept: */*" \
--header "Authorization: Basic $BASIC_AUTH"
```

完成后

从要创建新用户和角色的列表中选择所需的SVM。

## 第2步：列出为SVM定义的用户

执行以下REST API调用以列出在选定SVM中定义的用户。您可以通过owner参数来标识SVM。

**HTTP方法和端点**

此REST API调用使用以下方法和端点。

| HTTP 方法 | 路径 |
| --- | --- |
| 获取 | /api/安全性/帐户 |

**curl 示例**

```
curl --request GET \
--location "https://$FQDN_IP/api/security/accounts?owner.name=dmp" \
--include \
--header "Accept: */*" \
--header "Authorization: Basic $BASIC_AUTH"
```

完成后

根据SVM中已定义的用户、为新用户选择一个唯一名称。

## 第3步：列出为SVM定义的REST角色

执行以下REST API调用以列出在选定SVM中定义的角色。您可以通过owner参数来标识SVM。

**HTTP方法和端点**

此REST API调用使用以下方法和端点。

| HTTP 方法 | 路径 |
| --- | --- |
| 获取 | /api/安全性/角色 |

**curl 示例**

```
curl --request GET \
--location "https://$FQDN_IP/api/security/roles?owner.name=dmp" \
--include \
--header "Accept: */*" \
--header "Authorization: Basic $BASIC_AUTH" \
--data @JSONinput
```

完成后

根据SVM中已定义的角色、为新角色选择一个唯一名称。

## 第4步：创建自定义REST角色

对SVM中的创建自定义REST角色执行以下REST API调用。此角色最初只有一个权限、用于建立默认访问权限*无*、以拒绝所有访问。

**HTTP**方法和端点

此REST API调用使用以下方法和端点。

| **HTTP** 方法 | 路径 |
| --- | --- |
| 发布 | /api/安全性/角色 |

**curl** 示例

```
curl --request POST \
--location "https://$FQDN_IP/api/security/roles" \
--include \
--header "Accept: */*" \
--header "Authorization: Basic $BASIC_AUTH" \
--data @JSONinput
```

**JSON** 输入示例

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

完成后

(可选)再次执行步骤3以显示新角色。您还可以在ONTAP 命令行界面中显示角色。

## 第5步：通过添加更多权限来更新角色

执行以下REST API调用、以便根据需要添加权限来修改角色。

**HTTP**方法和端点

此REST API调用使用以下方法和端点。

| **HTTP** 方法 | 路径 |
| --- | --- |
| 发布 | /api/安全性/角色/｛owner.uuid｝/｛name｝/权限 |

**CURL示例的其他输入参数**

除了所有REST API调用通用的参数之外、此步骤中的cURL示例还会使用以下参数。

| 参数 | Type | Required | Description |
|---|---|---|---|
| $SVM_ID | 路径 | 是的。 | 包含角色定义的SVM的UUID。 |
| $Role_name | 路径 | 是的。 | 要更新的SVM中的角色名称。 |

**curl** 示例

```
curl --request POST \
--location
"https://$FQDN_IP/api/security/roles/$SVM_ID/$ROLE_NAME/privileges" \
--include \
--header "Accept: */*" \
--header "Authorization: Basic $BASIC_AUTH" \
--data @JSONinput
```

**JSON** 输入示例

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

完成后

(可选)再次执行步骤3以显示新角色。您还可以在ONTAP 命令行界面中显示角色。

## 第6步：创建用户

对创建用户帐户执行以下REST API调用。上面创建的角色*dprole1*与新用户关联。

> 您可以创建没有角色的用户。在这种情况下、系统会为用户分配一个默认角色( `admin` 或 `vsadmin`)、具体取决于用户是使用集群还是SVM范围定义的。您需要修改用户以分配其他角色。

**HTTP方法和端点**

此REST API调用使用以下方法和端点。

| **HTTP** 方法 | 路径 |
|---|---|
| 发布 | /api/安全性/帐户 |

**curl 示例**

```
curl --request POST \
--location "https://$FQDN_IP/api/security/accounts" \
--include \
--header "Accept: */*" \
--header "Authorization: Basic $BASIC_AUTH" \
--data @JSONinput
```

**JSON 输入示例**

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

完成后

您可以使用新用户的凭据登录到SVM管理界面。