



RBAC

ONTAP Automation

NetApp
July 19, 2024

This PDF was generated from https://docs.netapp.com/zh-cn/ontap-automation/workflows/wf_rbac_prepare.html on July 19, 2024. Always check docs.netapp.com for the latest.

目录

RBAC	1
准备使用RBAC	1
创建角色	1
创建具有角色的用户	5

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。

版权信息

版权所有 © 2024 NetApp, Inc.。保留所有权利。中国印刷。未经版权所有者事先书面许可，本档中受版权保护的任何部分不得以任何形式或通过任何手段（图片、电子或机械方式，包括影印、录音、录像或存储在电子检索系统中）进行复制。

从受版权保护的 NetApp 资料派生的软件受以下许可和免责声明的约束：

本软件由 NetApp 按“原样”提供，不含任何明示或暗示担保，包括但不限于适销性以及针对特定用途的适用性的隐含担保，特此声明不承担任何责任。在任何情况下，对于因使用本软件而以任何方式造成的任何直接性、间接性、偶然性、特殊性、惩罚性或后果性损失（包括但不限于购买替代商品或服务；使用、数据或利润方面的损失；或者业务中断），无论原因如何以及基于何种责任理论，无论出于合同、严格责任或侵权行为（包括疏忽或其他行为），NetApp 均不承担责任，即使已被告知存在上述损失的可能性。

NetApp 保留在不另行通知的情况下随时对本文档所述的任何产品进行更改的权利。除非 NetApp 以书面形式明确同意，否则 NetApp 不承担因使用本文档所述产品而产生的任何责任或义务。使用或购买本产品不表示获得 NetApp 的任何专利权、商标权或任何其他知识产权许可。

本手册中描述的产品可能受一项或多项美国专利、外国专利或正在申请的专利的保护。

有限权利说明：政府使用、复制或公开本文档受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中“技术数据权利 — 非商用”条款第 (b)(3) 条规定的限制条件的约束。

本文档中所含数据与商业产品和/或商业服务（定义见 FAR 2.101）相关，属于 NetApp, Inc. 的专有信息。根据本协议提供的所有 NetApp 技术数据和计算机软件具有商业性质，并完全由私人出资开发。美国政府对这些数据的使用权具有非排他性、全球性、受限且不可撤销的许可，该许可既不可转让，也不可再许可，但仅限在与交付数据所依据的美国政府合同有关且受合同支持的情况下使用。除本文档规定的情形外，未经 NetApp, Inc. 事先书面批准，不得使用、披露、复制、修改、操作或显示这些数据。美国政府对国防部的授权仅限于 DFARS 的第 252.227-7015(b)（2014 年 2 月）条款中明确的权利。

商标信息

NetApp、NetApp 标识和 <http://www.netapp.com/TM> 上所列的商标是 NetApp, Inc. 的商标。其他公司和产品名称可能是其各自所有者的商标。