



工作流 ONTAP automation

NetApp
January 19, 2026

This PDF was generated from https://docs.netapp.com/zh-cn/ontap-automation/workflows/prepare_workflows.html on January 19, 2026. Always check docs.netapp.com for the latest.

目录

工作流	1
准备使用ONTAP REST API工作流	1
简介	1
输入变量	1
身份验证选项	2
将示例与Bash结合使用	4
集群	4
使用ONTAP REST API获取集群配置	4
使用ONTAP REST API更新集群联系人	5
使用ONTAP REST API获取作业实例	6
NAS	7
文件安全权限	7
网络	16
列出使用ONTAP REST API的IP接口	16
安全性	24
—帐户	24
证书和密钥	26
RBAC	29
存储	38
列出使用ONTAP REST API的聚合	38
列出使用ONTAP REST API的磁盘	40
支持	42
EMS	43
SVM	49
列出使用ONTAP REST API的SVM	49

工作流

准备使用ONTAP REST API工作流

在实时ONTAP部署中使用工作流之前、您应熟悉这些工作流的结构和格式。



您应确保您的ONTAP版本支持您计划使用的工作流中的所有API调用。请参见 ["API 参考"](#) 有关详细信息 ...

简介

`_Workflow_` 是完成特定管理任务或目标所需的一个或多个步骤的序列。ONTAP工作流包括完成每项任务所需的核心步骤和参数。它们为自定义ONTAP自动化环境提供了一个起点。

步骤类型

ONTAP工作流中的每个步骤都属于以下类型之一：

- REST API 调用（包含 curl 和 JSON 示例等详细信息）
- 执行或调用其他ONTAP工作流
- 其他相关任务(例如制定配置决策)

REST API调用

大多数工作流步骤都是REST API调用。这些步骤使用通用格式、其中包括卷曲示例和其他信息。请参见 ["API 参考"](#) 有关REST API调用的更多详细信息、请参见。

单步工作流

一个工作流只能包含一个步骤。这些"单步工作流"的格式与包含多个步骤的工作流略有不同。例如、删除显式步骤名称。操作或操作应根据工作流标题明确。

输入变量

这些工作流的设计尽可能通用、因此可以在任何ONTAP环境中使用。考虑到这一点、REST API调用会在cURL示例和其他输入中使用变量。然后、可以轻松地根据不同的ONTAP环境调整REST API调用。

基本URL格式

您可以直接通过CURL或编程语言访问ONTAP REST API。在这种情况下、基本URL与访问ONTAP联机文档或System Manager时使用的URL不同。

直接访问API时，需要将*API*附加到域或IP地址。例如：

<https://ontap.demo-example.com/api>

请参见 ["如何访问ONTAP REST API"](#) 有关详细信息 ...

通用输入参数

大多数REST API调用通常使用多个输入参数。这些参数通常不会在各个工作流中进行介绍。您应熟悉这些参数。请参见 ["控制 API 请求的输入变量"](#) 有关详细信息 ...

如果特定REST API调用需要其他参数、请参见每个工作流的* cURL示例的其他输入参数*一节。

可变格式

工作流示例中使用的ID值和其他变量是不透明的、并且可能因每个ONTAP集群而异。为了提高示例的可读性、不使用实际值。而是使用变量。此方法基于一致的格式和一组保留名称、具有以下几个优势：

- CURL和JSON示例更易读、更易于理解。
- 由于所有关键字都使用相同的格式、因此您可以快速识别它们。
- 不存在安全风险、因为这些值无法复制和重复使用。

这些变量的格式设置为在Bash shell环境中使用。每个变量都以美元符号开头、并根据需要用双引号括起来。这使得他们可以巴什所识别。名称始终使用大写。

下面是一些常见的变量关键字。此列表并非详尽无遗、需要时可使用其他变量。根据具体情况，其含义应该是显而易见的。

关键字	Type	Description
\$FQDN_IP	URL	ONTAP管理LIF的完全限定域名或IP地址。
\$cluster ID	路径	UUIDv4值、用于标识运行API操作的ONTAP集群。
\$BASIC AUT1.	标题	用于HTTP基本身份验证的凭据字符串。

JSON输入示例

某些REST API调用(例如使用POST或PATCH)的调用需要在请求正文中输入JSON。为了清楚起见、JSON输入示例与CURL示例分开显示。您可以将JSON输入示例与下面所述的一种技术结合使用。

保存到本地文件

您可以将JSON输入示例复制到文件并将其保存在本地。CURL命令引用使用的文件 `--data` 带有值的参数、用于指示带有的文件名 `@` 前缀。

在卷曲示例后粘贴到终端中

首先、您需要将CURL示例复制并粘贴到终端Shell中。然后编辑此示例以完全删除 `--data` 参数、并将其替换为 `--data-raw` 参数。最后、复制并粘贴JSON示例、使其遵循带有更新参数的URL命令。您应使用单引号将JSON输入示例包装起来。

身份验证选项

REST API可用的主要身份验证技术是HTTP基本身份验证。从ONTAP 9.14开始、您还可以选择使用开放式授权(OAuth2.0)框架以及基于令牌的身份验证和授权。

HTTP基本身份验证

使用基本身份验证时、每个HTTP请求都必须包含用户凭据。有两种发送凭据的方法。

构建HTTP请求标头

您可以手动构建授权标头并将其包含在HTTP请求中。如果在CLI中使用cURL命令或对自动化代码使用编程语言、则可以执行此操作。高级别步骤包括：

1. 将用户和密码值与冒号串联：

```
admin:david123
```

2. 将整个字符串转换为base64：

```
YWRtaW46ZGF2aWQxMjM=
```

3. 构建请求标头：

```
Authorization: Basic YWRtaW46ZGF2aWQxMjM=
```

工作流卷曲示例包含此标题以及变量*\$BASIC _AUT*、使用前需要更新此标题。

使用Curl参数

使用Curl时的另一个选项是删除Authorization(授权)标头，而改用Curl *user*参数。例如：

```
--user username:password
```

您需要使用适用于您的环境的凭据进行替换。这些凭据不会在base64中进行编码。使用此参数执行URL命令时、系统会对字符串进行编码、并为您生成Authorization.标题。

OAuth2.0

使用OAuth2.0时、您需要从外部授权服务器请求访问令牌、并将其包含在每个HTTP请求中。下面将介绍基本的高级步骤。另请参见 ["ONTAP OAuth2.0实施概述"](#) 有关OAuth2.0以及如何将其与ONTAP结合使用的详细信息。

准备ONTAP环境

在使用REST API访问ONTAP之前、您需要准备和配置ONTAP环境。概括地说、这些步骤包括：

- 确定受ONTAP保护的资源和客户端
- 查看现有ONTAP REST角色和用户定义
- 安装和配置授权服务器
- 设计和配置客户端授权定义
- 配置ONTAP并启用OAuth2.0

请求访问令牌

在ONTAP以及授权服务器已定义且处于活动状态的情况下、您可以使用OAuth2.0令牌进行REST API调用。第一步是从授权服务器请求访问令牌。这是在ONTAP之外使用基于服务器的几种不同技术之一完成的。ONTAP不会通过问题描述访问令牌或执行重定向。

构建HTTP请求标头

获取访问令牌后、您可以构建授权标头并将其包含在HTTP请求中。无论使用CURL还是编程语言访问REST API、都必须在每个客户端请求中包含标头。您可以按如下所示构建报头：

```
Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSld ...
```

将示例与Bash结合使用

如果您直接使用工作流卷曲示例、则必须使用适合您环境的值更新其包含的变量。您可以手动编辑这些示例、也可以按如下所述使用bash shell进行替换。



使用"Bash"的一个优点是、您可以在shell会话中设置一次变量值、而不是在每个CURL命令中设置一次。

步骤

- 1. 打开随Linux或类似操作系统提供的bash shell。
- 2. 设置要运行的CURL示例中包含的变量值。例如：

```
CLUSTER_ID=ce559b75-4145-11ee-b51a-005056aee9fb
```

- 3. 从工作流页面复制卷曲示例并将其粘贴到shell终端中。
- 4. 按*enter*，将执行以下操作：
 - a. 替换您设置的变量值
 - b. 执行Curl命令

集群

使用ONTAP REST API获取集群配置

您可以检索ONTAP集群的配置、包括特定字段。您可以在评估集群状态期间或更新配置之前执行此操作。

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/cluster

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
fields	查询	否	选择要返回的值。示例包括 <code>contact</code> 和 <code>version</code> 。

CURL示例：检索集群联系信息

此示例说明了如何检索单个字段。要获取整个集群对象和配置、您需要删除 `fields` 查询参数。

```
curl --request GET \  
--location "https://$FQDN_IP/api/cluster?fields=contact" \  
--include \  
--header "Accept: */*" \  
--header "Authorization: Basic $BASIC_AUTH"
```

JSON 输出示例

```
{  
  "contact": "support@company-demo.com"  
}
```

使用ONTAP REST API更新集群联系人

您可以更新集群的联系信息。由于此请求是异步处理的、因此您还需要确定关联的后台作业是否已成功完成。

第1步：更新集群联系信息

您可以通过问题描述调用来更新集群联系信息。

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
patch	/API/cluster

处理类型

异步

curl 示例

```
curl --request PATCH \  
--location "https://$FQDN_IP/api/cluster" \  
--include \  
--header "Content-Type: application/json" \  
--header "Accept: */*" \  
--header "Authorization: Basic $BASIC_AUTH" \  
--data @JSONinput
```

JSON 输入示例

```
{
  "contact": "support@company-demo.com"
}
```

JSON 输出示例

返回作业对象。您应保存作业标识符、以便在下一步中使用。

```
{ "job": {
  "uuid": "d877f5bb-3aa7-11e9-b6c6-005056a78c89",
  "_links": {
    "self": {
      "href": "/api/cluster/jobs/d877f5bb-3aa7-11e9-b6c6-005056a78c89"
    }
  }
}
```

第2步：检索作业状态

执行工作流 ["获取作业实例"](#) 并确认 state 值为 success。

第3步：确认集群联系信息

执行工作流 ["获取集群配置"](#)。您应设置 fields 将参数查询到 contact。

使用ONTAP REST API获取作业实例

您可以检索特定ONTAP作业的实例。通常、您可以执行此操作来确定作业和关联操作是否成功完成。



您需要作业对象的UUID、此UUID通常在发出异步请求后提供。另请查看 ["使用作业对象进行异步处理"](#) 在处理ONTAP内部作业之前。

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/cluster-Jobs/ {UUID}

处理类型

同步

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，此步骤的 curl 示例还使用以下参数。

参数	Type	Required	Description
\$job_ID	路径	是的。	用于标识所请求的作业。

curl 示例

```
curl --request GET \  
--location "https://$FQDN_IP/api/cluster/jobs/$JOB_ID" \  
--include \  
--header "Accept: */*" \  
--header "Authorization: Basic $BASIC_AUTH"
```

JSON 输出示例

状态值和其他字段包括在返回的作业对象中。此示例中的作业是在更新ONTAP集群过程中运行的。

```
{  
  "uuid": "d877f5bb-3aa7-11e9-b6c6-005056a78c89",  
  "description": "PATCH /api/cluster",  
  "state": "success",  
  "message": "success",  
  "code": 0,  
  "_links": {  
    "self": {  
      "href": "/api/cluster/jobs/d877f5bb-3aa7-11e9-b6c6-005056a78c89"  
    }  
  }  
}
```

NAS

文件安全权限

准备使用**ONTAP REST API**管理文件安全和审核策略

您可以管理ONTAP集群中通过SVM提供的文件的权限和审核策略。

概述

ONTAP 使用系统访问控制列表（SACL）和随机访问控制列表（DACL）为文件对象分配权限。从ONTAP 9.9.1开始，REST API支持管理SACL和DACL权限。您可以使用API自动管理文件安全权限。在许多情况下，您可以使用单个REST API调用，而不是多个命令行界面命令或ONTAP PI (ZAPI)调用。



对于9.9.1之前的ONTAP版本、您可以使用命令行界面直通功能自动管理SACL和DACL权限。请参见 ["迁移注意事项"](#) 和 ["将专用命令行界面直通与 ONTAP REST API 结合使用"](#) 有关详细信息 ...

我们提供了几个示例工作流来说明如何使用REST API管理ONTAP文件安全服务。在使用工作流并发出任何REST API调用之前、请务必查看 ["准备使用这些工作流"](#)。

如果您使用Python、另请参见脚本 ["file_security_permissions.py"](#) 有关如何自动执行某些文件安全活动的示例。

ONTAP REST API 与 ONTAP 命令行界面命令

对于许多任务、使用ONTAP REST API所需的调用比等效的ONTAP命令行界面命令或ONTAP API (ZAPI)调用更少。下表列出了API调用以及每项任务所需的等效命令行界面命令。

ONTAP REST API	ONTAP 命令行界面
get /protocols/file-security/effective-permissions/	vserver security file-directory show-effective-permissions
POST /protocols/file-security/permissions/	1. vserver security file-directory ntfs create 2. vserver security file-directory ntfs dacl add 3. vserver security file-directory ntfs sacl add 4. vserver security file-directory policy create 5. vserver security file-directory policy task add 6. Vserver security file-directory apply
patch /protocols/file-security/permissions/	vserver security file-directory ntfs modify
delete /protocols/file-security/permissions/	1. vserver security file-directory ntfs dacl remove 2. vserver security file-directory ntfs sacl remove

相关信息

- ["展示文件权限的Python脚本"](#)
- ["使用 ONTAP REST API 简化文件安全权限的管理"](#)
- ["将专用命令行界面直通与 ONTAP REST API 结合使用"](#)

使用ONTAP REST API获取文件的有效权限

您可以检索特定文件或文件夹的当前有效权限。

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/protocols文件安全性/有效权限/ {svm.unid} / {path}

处理类型

同步

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。	这是包含该文件的SVM的UUID。
\$file_path	路径	是的。	这是文件或文件夹的路径。

curl 示例

```
curl --request GET \
--location "https://$FQDN_IP/api/protocols/file-security/effective-
permissions/$SVM_ID/$FILE_PATH" \
--include \
--header "Accept: */*" \
--header "Authorization: Basic $BASIC_AUTH"
```

JSON 输出示例

```
{
  "svm": {
    "uuid": "cf5f271a-1beb-11ea-8fad-005056bb645e",
    "name": "vs1"
  },
  "user": "administrator",
  "type": "windows",
  "path": "/",
  "share": {
    "path": "/"
  },
  "file_permission": [
    "read",
    "write",
    "append",
    "read_ea",
    "write_ea",
    "execute",
    "delete_child",
    "read_attributes",
    "write_attributes",
    "delete",
    "read_control",
    "write_dac",
    "write_owner",
    "synchronize",
    "system_security"
  ],
  "share_permission": [
    "read",
    "read_ea",
    "execute",
    "read_attributes",
    "read_control",
    "synchronize"
  ]
}
```

使用**ONTAP REST API**获取文件的审核信息

您可以检索特定文件或文件夹的审核信息。

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/protocols /文件安全性/权限/ {svm.unid} / {path}

处理类型

同步

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。	这是包含该文件的SVM的UUID。
\$file_path	路径	是的。	这是文件或文件夹的路径。

curl 示例

```
curl --request GET \
--location "https://$FQDN_IP/api/protocols/file-
security/permissions/$SVM_ID/$FILE_PATH" \
--include \
--header "Accept: */*" \
--header "Authorization: Basic $BASIC_AUTH"
```

JSON 输出示例

```
{
  "svm": {
    "uuid": "9479099d-5b9f-11eb-9c4e-0050568e8682",
    "name": "vs1"
  },
  "path": "/parent",
  "owner": "BUILTIN\\Administrators",
  "group": "BUILTIN\\Administrators",
  "control_flags": "0x8014",
  "acls": [
    {
      "user": "BUILTIN\\Administrators",
      "access": "access_allow",
      "apply_to": {
        "files": true,
        "sub_folders": true,
        "this_folder": true
      },
      "advanced_rights": {
        "append_data": true,
        "delete": true,
```

```

        "delete_child": true,
        "execute_file": true,
        "full_control": true,
        "read_attr": true,
        "read_data": true,
        "read_ea": true,
        "read_perm": true,
        "write_attr": true,
        "write_data": true,
        "write_ea": true,
        "write_owner": true,
        "synchronize": true,
        "write_perm": true
    },
    "access_control": "file_directory"
},
{
    "user": "BUILTIN\\Users",
    "access": "access_allow",
    "apply_to": {
        "files": true,
        "sub_folders": true,
        "this_folder": true
    },
    "advanced_rights": {
        "append_data": true,
        "delete": true,
        "delete_child": true,
        "execute_file": true,
        "full_control": true,
        "read_attr": true,
        "read_data": true,
        "read_ea": true,
        "read_perm": true,
        "write_attr": true,
        "write_data": true,
        "write_ea": true,
        "write_owner": true,
        "synchronize": true,
        "write_perm": true
    },
    "access_control": "file_directory"
}
],
"inode": 64,
"security_style": "mixed",

```

```

"effective_style": "ntfs",
"dos_attributes": "10",
"text_dos_attr": "----D---",
"user_id": "0",
"group_id": "0",
"mode_bits": 777,
"text_mode_bits": "rwxrwxrwx"
}

```

使用**ONTAP REST API**将新权限应用于文件

您可以将新的安全描述符应用于特定文件或文件夹。

第1步：应用新权限

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/protocols /文件安全性/权限/ {svm.unid} / {path}

处理类型

异步

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。	这是包含该文件的SVM的UUID。
\$file_path	路径	是的。	这是文件或文件夹的路径。

curl 示例

```
curl --request POST --location "https://$FQDN_IP/api/protocols/file-security/permissions/$SVM_ID/$FILE_PATH?return_timeout=0" --include --header "Accept */*" --header "Authorization: Basic $BASIC_AUTH" --data '{ \acl\": [ { \access\": \access_allow\", \advanced_rights\": { \append_data\": true, \delete\": true, \delete_child\": true, \execute_file\": true, \full_control\": true, \read_attr\": true, \read_data\": true, \read_ea\": true, \read_perm\": true, \write_attr\": true, \write_data\": true, \write_ea\": true, \write_owner\": true, \write_perm\": true }, \apply_to\": { \files\": true, \sub_folders\": true, \this_folder\": true }, \user\": \administrator\" } ], \control_flags\": \32788\", \group\": \S-1-5-21-2233347455-2266964949-1780268902-69700\", \ignore_paths\": [ \parent/child2\" ], \owner\": \S-1-5-21-2233347455-2266964949-1780268902-69304\", \propagation_mode\": \propagate\''
```

JSON 输出示例

```
{
  "job": {
    "uuid": "3015c294-5bbc-11eb-9c4e-0050568e8682",
    "_links": {
      "self": {
        "href": "/api/cluster/jobs/3015c294-5bbc-11eb-9c4e-0050568e8682"
      }
    }
  }
}
```

第2步：检索作业状态

执行工作流 ["获取作业实例"](#) 并确认 state 值为 success。

使用ONTAP REST API更新安全描述符信息

您可以将特定安全描述符更新到特定文件或文件夹、包括主所有者、组或控制标志。

第1步：更新安全描述符

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
patch	/api/protocols /文件安全性/权限/ {svm.unid} / {path}

处理类型
异步

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。	这是包含该文件的SVM的UUID。
\$file_path	路径	是的。	这是文件或文件夹的路径。

curl 示例

```
curl --request POST --location "https://$FQDN_IP/api/protocols/file-security/permissions/$SVM_ID/$FILE_PATH?return_timeout=0" --include --header "Accept */*" --header "Authorization: Basic $BASIC_AUTH" --data '{ \"control_flags\": \"32788\", \"group\": \"everyone\", \"owner\": \"user1\"}'
```

JSON 输出示例

```
{
  "job": {
    "uuid": "6f89e612-5bbd-11eb-9c4e-0050568e8682",
    "_links": {
      "self": {
        "href": "/api/cluster/jobs/6f89e612-5bbd-11eb-9c4e-0050568e8682"
      }
    }
  }
}
```

第2步：检索作业状态
执行工作流 ["获取作业实例"](#) 并确认 state 值为 success。

使用ONTAP REST API删除访问控制条目

您可以从特定文件或文件夹中删除现有访问控制条目(ACE)。更改会传播到任何子对象。

第1步：删除ACE

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
删除	/api/protocols /文件安全性/权限/ {svm.unid} / {path}

处理类型

异步

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。	这是包含该文件的SVM的UUID。
\$file_path	路径	是的。	这是文件或文件夹的路径。

curl 示例

```
curl --request DELETE --location "https://$FQDN_IP/api/protocols/file-security/permissions/$SVM_ID/$FILE_PATH?return_timeout=0" --include --header "Accept */*" --header "Authorization: Basic $BASIC_AUTH" --data '{ \"access\": \"access_allow\", \"apply_to\": { \"files\": true, \"sub_folders\": true, \"this_folder\": true }, \"ignore_paths\": [ \"/parent/child2\" ], \"propagation_mode\": \"propagate\"}'
```

JSON 输出示例

```
{
  "job": {
    "uuid": "3015c294-5bbc-11eb-9c4e-0050568e8682",
    "_links": {
      "self": {
        "href": "/api/cluster/jobs/3015c294-5bbc-11eb-9c4e-0050568e8682"
      }
    }
  }
}
```

第2步：检索作业状态

执行工作流 ["获取作业实例"](#) 并确认 state 值为 success。

网络

列出使用ONTAP REST API的IP接口

您可以检索分配给集群和SVM的IP SVM。您可以执行此操作来确认网络配置或计划添加另

一个LIF。

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/network/IP/接口

处理类型

同步

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，此步骤的 curl 示例还使用以下参数。

参数	Type	Required	Description
fields	查询	否	返回相关配置值的有限列表。

CURL示例：返回具有默认配置值的所有LI

```
curl --request GET \  
--location "https://$FQDN_IP/api/network/ip/interfaces" \  
--include \  
--header "Accept: */*" \  
--header "Authorization: Basic $BASIC_AUTH"
```

CURL示例：返回具有四个特定配置值的所有LI

```
curl --request GET \  
--location \  
"https://$FQDN_IP/api/network/ip/interfaces?fields=name,scope,svm.name,ip.  
address" \  
--include \  
--header "Accept: */*" \  
--header "Authorization: Basic $BASIC_AUTH"
```

```
{
  "records": [
    {
      "uuid": "5ded9e38-999e-11ee-acad-005056ae6bd8",
      "name": "sti214-vsim-sr027o_mgmt1",
      "ip": {
        "address": "172.29.151.116"
      },
      "scope": "cluster",
      "_links": {
        "self": {
          "href": "/api/network/ip/interfaces/5ded9e38-999e-11ee-acad-005056ae6bd8"
        }
      }
    },
    {
      "uuid": "bb03c162-999e-11ee-acad-005056ae6bd8",
      "name": "cluster_mgmt",
      "ip": {
        "address": "172.29.186.156"
      },
      "scope": "cluster",
      "_links": {
        "self": {
          "href": "/api/network/ip/interfaces/bb03c162-999e-11ee-acad-005056ae6bd8"
        }
      }
    },
    {
      "uuid": "c5ffbd03-999e-11ee-acad-005056ae6bd8",
      "name": "sti214-vsim-sr027o_data1",
      "ip": {
        "address": "172.29.186.150"
      },
      "scope": "svm",
      "svm": {
        "name": "vs0"
      },
      "_links": {
        "self": {
          "href": "/api/network/ip/interfaces/c5ffbd03-999e-11ee-acad-"

```

```

005056ae6bd8"
    }
  }
},
{
  "uuid": "c6612abe-999e-11ee-acad-005056ae6bd8",
  "name": "sti214-vsim-sr027o_data2",
  "ip": {
    "address": "172.29.186.151"
  },
  "scope": "svm",
  "svm": {
    "name": "vs0"
  },
  "_links": {
    "self": {
      "href": "/api/network/ip/interfaces/c6612abe-999e-11ee-acad-005056ae6bd8"
    }
  }
},
{
  "uuid": "c6b21b94-999e-11ee-acad-005056ae6bd8",
  "name": "sti214-vsim-sr027o_data3",
  "ip": {
    "address": "172.29.186.152"
  },
  "scope": "svm",
  "svm": {
    "name": "vs0"
  },
  "_links": {
    "self": {
      "href": "/api/network/ip/interfaces/c6b21b94-999e-11ee-acad-005056ae6bd8"
    }
  }
},
{
  "uuid": "c7025322-999e-11ee-acad-005056ae6bd8",
  "name": "sti214-vsim-sr027o_data4",
  "ip": {
    "address": "172.29.186.153"
  },
  "scope": "svm",
  "svm": {

```

```

      "name": "vs0"
    },
    "_links": {
      "self": {
        "href": "/api/network/ip/interfaces/c7025322-999e-11ee-acad-005056ae6bd8"
      }
    }
  },
  {
    "uuid": "c752cc66-999e-11ee-acad-005056ae6bd8",
    "name": "sti214-vsim-sr027o_data5",
    "ip": {
      "address": "172.29.186.154"
    },
    "scope": "svm",
    "svm": {
      "name": "vs0"
    },
    "_links": {
      "self": {
        "href": "/api/network/ip/interfaces/c752cc66-999e-11ee-acad-005056ae6bd8"
      }
    }
  },
  {
    "uuid": "c7a03719-999e-11ee-acad-005056ae6bd8",
    "name": "sti214-vsim-sr027o_data6",
    "ip": {
      "address": "172.29.186.155"
    },
    "scope": "svm",
    "svm": {
      "name": "vs0"
    },
    "_links": {
      "self": {
        "href": "/api/network/ip/interfaces/c7a03719-999e-11ee-acad-005056ae6bd8"
      }
    }
  },
  {
    "uuid": "ccd4c59c-999e-11ee-acad-005056ae6bd8",
    "name": "sti214-vsim-sr027o_data4_inet6",

```

```

    "ip": {
      "address": "fd20:8b1e:b255:300f::ac5"
    },
    "scope": "svm",
    "svm": {
      "name": "vs0"
    },
    "_links": {
      "self": {
        "href": "/api/network/ip/interfaces/ccd4c59c-999e-11ee-acad-005056ae6bd8"
      }
    }
  },
  {
    "uuid": "d9144c30-999e-11ee-acad-005056ae6bd8",
    "name": "sti214-vsim-sr027o_data6_inet6",
    "ip": {
      "address": "fd20:8b1e:b255:300f::ac7"
    },
    "scope": "svm",
    "svm": {
      "name": "vs0"
    },
    "_links": {
      "self": {
        "href": "/api/network/ip/interfaces/d9144c30-999e-11ee-acad-005056ae6bd8"
      }
    }
  },
  {
    "uuid": "d961c13b-999e-11ee-acad-005056ae6bd8",
    "name": "sti214-vsim-sr027o_data1_inet6",
    "ip": {
      "address": "fd20:8b1e:b255:300f::ac2"
    },
    "scope": "svm",
    "svm": {
      "name": "vs0"
    },
    "_links": {
      "self": {
        "href": "/api/network/ip/interfaces/d961c13b-999e-11ee-acad-005056ae6bd8"
      }
    }
  }
}

```

```

    }
  },
  {
    "uuid": "d9ac8d6a-999e-11ee-acad-005056ae6bd8",
    "name": "sti214-vsim-sr027o_data5_inet6",
    "ip": {
      "address": "fd20:8b1e:b255:300f::ac6"
    },
    "scope": "svm",
    "svm": {
      "name": "vs0"
    },
    "_links": {
      "self": {
        "href": "/api/network/ip/interfaces/d9ac8d6a-999e-11ee-acad-005056ae6bd8"
      }
    }
  },
  {
    "uuid": "d9fc1a3-999e-11ee-acad-005056ae6bd8",
    "name": "sti214-vsim-sr027o_data2_inet6",
    "ip": {
      "address": "fd20:8b1e:b255:300f::ac3"
    },
    "scope": "svm",
    "svm": {
      "name": "vs0"
    },
    "_links": {
      "self": {
        "href": "/api/network/ip/interfaces/d9fc1a3-999e-11ee-acad-005056ae6bd8"
      }
    }
  },
  {
    "uuid": "da4995a0-999e-11ee-acad-005056ae6bd8",
    "name": "sti214-vsim-sr027o_data3_inet6",
    "ip": {
      "address": "fd20:8b1e:b255:300f::ac4"
    },
    "scope": "svm",
    "svm": {
      "name": "vs0"
    },
  },

```

```

    "_links": {
      "self": {
        "href": "/api/network/ip/interfaces/da4995a0-999e-11ee-acad-005056ae6bd8"
      }
    },
    {
      "uuid": "da9e7afd-999e-11ee-acad-005056ae6bd8",
      "name": "sti214-vsrm-sr027o_cluster_mgmt_inet6",
      "ip": {
        "address": "fd20:8b1e:b255:300f::ac8"
      },
      "scope": "cluster",
      "_links": {
        "self": {
          "href": "/api/network/ip/interfaces/da9e7afd-999e-11ee-acad-005056ae6bd8"
        }
      }
    },
    {
      "uuid": "e6db58b4-999e-11ee-acad-005056ae6bd8",
      "name": "sti214-vsrm-sr027o_mgmt1_inet6",
      "ip": {
        "address": "fd20:8b1e:b255:3008::1a0"
      },
      "scope": "cluster",
      "_links": {
        "self": {
          "href": "/api/network/ip/interfaces/e6db58b4-999e-11ee-acad-005056ae6bd8"
        }
      }
    }
  ],
  "num_records": 16,
  "_links": {
    "self": {
      "href":
"/api/network/ip/interfaces?fields=name,scope,svm.name,ip.address"
    }
  }
}

```

安全性

一帐户

使用**ONTAP REST API**列出帐户

您可以检索帐户列表。您可以在评估安全环境或创建新帐户之前执行此操作。

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

处理类型

同步

curl 示例

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

```
{
  "records": [
    {
      "owner": {
        "uuid": "642573a8-9d14-11ee-9330-005056aed3de",
        "name": "vs0",
        "_links": {
          "self": {
            "href": "/api/svm/svms/642573a8-9d14-11ee-9330-005056aed3de"
          }
        }
      },
      "name": "vsadmin",
      "_links": {
        "self": {
          "href": "/api/security/accounts/642573a8-9d14-11ee-9330-005056aed3de/vsadmin"
        }
      }
    },
    {
      "owner": {
        "uuid": "fdb6fe29-9d13-11ee-9330-005056aed3de",
        "name": "sti214nscluster-1"
      },
      "name": "admin",
      "_links": {
        "self": {
          "href": "/api/security/accounts/fdb6fe29-9d13-11ee-9330-005056aed3de/admin"
        }
      }
    },
    {
      "owner": {
        "uuid": "fdb6fe29-9d13-11ee-9330-005056aed3de",
        "name": "sti214nscluster-1"
      },
      "name": "autosupport",
      "_links": {
        "self": {
          "href": "/api/security/accounts/fdb6fe29-9d13-11ee-9330-"

```

```

005056aed3de/autosupport"
    }
  }
},
"num_records": 3,
"_links": {
  "self": {
    "href": "/api/security/accounts"
  }
}
}
}

```

证书和密钥

使用**ONTAP REST API**列出已安装的证书

您可以列出ONTAP集群中安装的证书。您可以执行此操作来查看特定证书是否可用或获取特定证书的ID。

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/secury/certificates

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
max_记录	查询	否	指定要返回的记录数。

CURL示例：返回三个证书

```

curl --request GET \
--location "https://$FQDN_IP/api/security/certificates?max_records=3" \
--include \
--header "Accept: */*" \
--header "Authorization: Basic $BASIC_AUTH"

```

JSON 输出示例

```
{
  "records": [
    {
      "uuid": "dad822c2-573c-11ee-a310-005056aecc29",
      "name": "vs0_17866DB5C933E2EA",
      "_links": {
        "self": {
          "href": "/api/security/certificates/dad822c2-573c-11ee-a310-005056aecc29"
        }
      }
    },
    {
      "uuid": "7d8e5570-573c-11ee-a310-005056aecc29",
      "name": "BuypassClass3RootCA",
      "_links": {
        "self": {
          "href": "/api/security/certificates/7d8e5570-573c-11ee-a310-005056aecc29"
        }
      }
    },
    {
      "uuid": "7dbb2191-573c-11ee-a310-005056aecc29",
      "name": "EntrustRootCertificationAuthority",
      "_links": {
        "self": {
          "href": "/api/security/certificates/7dbb2191-573c-11ee-a310-005056aecc29"
        }
      }
    }
  ],
  "num_records": 3,
  "_links": {
    "self": {
      "href": "/api/security/certificates?max_records=3"
    },
    "next": {
      "href": "/api/security/certificates?start.svm_id=sti214nscluster-1&start.uuid=7dbb2191-573c-11ee-a310-005056aecc29&max_records=3"
    }
  }
}
```

使用ONTAP REST API安装证书

您可以在ONTAP集群中安装签名的X.509证书。您可以在配置需要强身份验证的ONTAP功能或协议时执行此操作。

开始之前

您必须具有要安装的证书。您还应确保根据需要安装任何中间证书。



在使用下面提供的JSON输入示例之前、请确保更新 `public_certificate` 价值。

第1步：安装证书

您可以通过问题描述调用来安装证书。

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/security/certificates

cURL示例：在集群级别安装根CA证书

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

JSON 输入示例

```
{
  "type": "server_ca",
  "public_certificate":
    "-----BEGIN CERTIFICATE-----
MIID0TCCArkCFGYdznvTVvaY1VZPNfy4yCCyPph6MA0GCSqGSIb3DQEBCwUAMIGk
MQswCQYDVQQGEwJVUzELMAkGA1UECAwCTkMxDDAKBgNVBACMA1JUUDEWMBQGA1UE
CgwNT05UQVAgRXhhbXBsZTETMBEGA1UECwwKT05UQVAgOS4xNDEcMBoGA1UEAwWT
Ki5vbnRhcC1leGFtcGxlLmNvbTEvMC0GCSqGSIb3DQEJARYgZGF2aWQucGV0ZXJz
b25Ab250YXAtZXhhbXBsZS5jb20wHhcNMjMxMDA1MTUyOTE4WhcNMjMxMDA0MTUy
OTE4WjCBpDELMAkGA1UEBhMCVVMxMzA1BgNVBAGMAk5DMQwwCgYDVQQHDANSVFAX
FjAUBgNVBAoMDU90VEFQIEV4YW1wbGUxEzARBgNVBASMcK90VEFQIDkuMTQxHDAa
BgNVBAMMEyoub250YXAtZXhhbXBsZS5jb20xLzAtBgkqhkiG9w0BCQEWIGRhdm1k
LnBlbGVyc29uQG9udGFwLWV4YW1wbGUuY29tMIIBIjANBgkqhkiG9w0BAQEFAAOc
AQ8AMIIBCgKCAQEAXQgy8mhb1Jhkf0D/MBodpZgW0aSp2jGbWJ+Zv2G8BXkp1762
dPHRkv1hnX9JvwkK4DBa05GiCiD5t3gjH/jUQMSFb+VwDbVmubVFnxjkm/4Q7sea
tMtA/ZpQdZbQFZ5RKtdWz7dzzPYEl2x8Q1Jc8Kh7NxERNMtgupGWZzn7mfXKYr4O
N/+vgahIhDibS8YK5rflw6bfmrik9E2D+PEab9DX/1DL5RX4tZ1H2OkYN2UxoBR6
Fq7l6n1Hi/5yR0OilxStN6s07EPoGak+KS1K41q+EcIKRo0bP4mEQp8WMjJuiTkb
5MmeYoIpWEUgJK7S0M6Tp/3bTh2CST3AWxiNxQIDAQABMA0GCSqGSIb3DQEBCwUA
A4IBAQAQABfBqOuR0mYxdfjrj930yIiRoDcoMzvo8cHGNUsuhnlBDnL2O3qhWEs97s0
mIy6zFMGnyNYa0t4ilcFsGDKP/JuljmYHjvv+2lHWnxHjTo7AOQCnXmQH5swoDbf
o1Vjqz8Oxz+PRJ+PA3dF5/8zqaAR6QreAN/iFR++6nUqlsbbM7w03tthBVMgo/h1
E9I2jVOZsqMFujm2CYfMs4XkZtrYmN6nZA8JcUpDjIWcAVbQYurMnna9r42oS3GB
WB/FE9n+P+FfJyHJ93KGcCXbH5RF2pi3wLlHilbvVuCjLRrhJ8U20I5mZoiXvAbc
IpYuBcuKXLwAarhDEacXttVjC+Bq
-----END CERTIFICATE-----"
}
```

第2步：确认证书已安装

执行工作流 ["列出已安装的证书"](#) 并确认证书可用。

RBAC

准备使用 **ONTAP REST API** 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

使用**ONTAP REST API**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。
\$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/privileges" \
--include \
--header "Accept: */*" \
--header "Authorization: Basic $BASIC_AUTH" \
--data @JSONinput
```

JSON 输入示例

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

使用ONTAP REST API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"}
  ]
}
```

允许使用**ONTAP REST API**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"}
  ]
}
```

使用ONTAP REST API创建具有某个角色的用户

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": "<password>"
}
```

完成后

您可以使用新用户的凭据登录到SVM管理界面。

存储

列出使用ONTAP REST API的聚合

您可以检索集群中的聚合列表。您可以执行此操作来评估利用率和性能。

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/storage/disks

处理类型

同步

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，此步骤的 curl 示例还使用以下参数。

参数	Type	Required	Description
node.name	查询	否	可用于标识每个聚合连接到的节点。

CURL示例：返回具有默认配置值的所有聚合

```
curl --request GET \
--location "https://$FQDN_IP/api/storage/aggregates" \
--include \
--header "Accept: */*" \
--header "Authorization: Basic $BASIC_AUTH"
```

CURL示例：返回具有特定配置值的所有聚合

```
curl --request GET \
--location "https://$FQDN_IP/api/storage/aggregates?fields=node.name" \
--include \
--header "Accept: */*" \
--header "Authorization: Basic $BASIC_AUTH"
```

JSON 输出示例

```
{
  "records": [
    {
      "uuid": "760d8137-fc59-47da-906a-cc28db0a1c1b",
      "name": "sti214_vsim_sr027o_aggr1",
      "node": {
        "name": "sti214-vsim-sr027o"
      },
      "_links": {
        "self": {
          "href": "/api/storage/aggregates/760d8137-fc59-47da-906a-cc28db0a1c1b"
        }
      }
    }
  ],
  "num_records": 1,
  "_links": {
    "self": {
      "href": "/api/storage/aggregates?fields=node.name"
    }
  }
}
```

列出使用ONTAP REST API的磁盘

您可以检索集群中的磁盘列表。您可以执行此操作来查找要在创建聚合过程中使用的一个或多个备用磁盘。

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/storage/disks

处理类型

同步

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，此步骤的 curl 示例还使用以下参数。

参数	Type	Required	Description
state	查询	否	可用于标识可用于新聚合的备用磁盘。

CURL示例：返回所有磁盘

```
curl --request GET \
--location "https://$FQDN_IP/api/storage/disks" \
--include \
--header "Accept: */*" \
--header "Authorization: Basic $BASIC_AUTH"
```

CURL示例：返回备用磁盘

```
curl --request GET \
--location "https://$FQDN_IP/api/storage/disks?state=spare" \
--include \
--header "Accept: */*" \
--header "Authorization: Basic $BASIC_AUTH"
```

```
{
  "records": [
    {
      "name": "NET-1.20",
      "state": "spare",
      "_links": {
        "self": {
          "href": "/api/storage/disks/NET-1.20"
        }
      }
    },
    {
      "name": "NET-1.12",
      "state": "spare",
      "_links": {
        "self": {
          "href": "/api/storage/disks/NET-1.12"
        }
      }
    },
    {
      "name": "NET-1.7",
      "state": "spare",
      "_links": {
        "self": {
          "href": "/api/storage/disks/NET-1.7"
        }
      }
    }
  ],
  "num_records": 3,
  "_links": {
    "self": {
      "href": "/api/storage/disks?state=spare"
    }
  }
}
```

支持

EMS

准备使用**ONTAP REST API**管理**EMS**支持服务

您可以为ONTAP集群配置事件管理系统(EMS)处理、并根据需要检索EMS消息。

概述

您可以通过多个示例工作流来说明如何使用ONTAP EMS服务。在使用工作流并发出任何REST API调用之前、请务必查看 ["准备使用这些工作流"](#)。

如果您使用Python、还可以参见"py" ["events.py"](#) 有关如何自动执行某些EMS相关活动的示例。

ONTAP REST API 与 ONTAP 命令行界面命令

对于许多任务、使用ONTAP REST API所需的调用比等效的ONTAP命令行界面命令少。下表列出了API调用以及每项任务所需的等效命令行界面命令。

ONTAP REST API	ONTAP 命令行界面
获取/support/ems	event config show
POST /support/EMS/目标	1. 事件通知目标创建 2. 事件通知创建
GET /support/EMS/Events	事件日志显示
POST /support/EMS/filters	1. event filter create -filter-name <FilterName> 2. event filter rule add -filter-name <FilterName>

相关信息

- ["用于展示EMS的Python脚本"](#)
- ["ONTAP REST API：自动通知高严重性事件"](#)

使用ONTAP REST API列出EMS日志事件

您可以检索所有事件通知消息、也可以仅检索具有特定特征的消息。

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/support/EMS/Events

处理类型

同步

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，此步骤的 curl 示例还使用以下参数。

参数	Type	Required	Description
fields	查询	否	用于请求将特定字段包括在响应中。
max_记录	查询	否	可用于限制单个请求中返回的记录数。
log_message	查询	否	用于搜索特定文本值、并仅返回匹配的消息。
message.severity	查询	否	将返回的消息限制为具有特定严重性的消息、例如 alert。

CURL示例：返回最新消息和名称值

```
curl --request GET \  
--location \  
"https://$FQDN_IP/api/support/ems/events?fields=message.name&max_records=1" \  
--include \  
--header "Accept: */*" \  
--header "Authorization: Basic $BASIC_AUTH"
```

CURL示例：返回包含特定文本和严重性的消息

```
curl --request GET \  
--location \  
"https://$FQDN_IP/api/support/ems/events?log_message=*disk*&message.severity=alert" \  
--include \  
--header "Accept: */*" \  
--header "Authorization: Basic $BASIC_AUTH"
```

JSON 输出示例

```
{
  "records": [
    {
      "node": {
        "name": "malha-vsim1",
        "uuid": "da4f9e62-9de3-11ec-976a-005056b369de",
        "_links": {
          "self": {
            "href": "/api/cluster/nodes/da4f9e62-9de3-11ec-976a-005056b369de"
          }
        }
      },
      "index": 4602,
      "time": "2022-03-18T06:37:46-04:00",
      "message": {
        "severity": "alert",
        "name": "raid.autoPart.disabled"
      },
      "log_message": "raid.autoPart.disabled: Disk auto-partitioning is disabled on this system: the system needs a minimum of 4 usable internal hard disks.",
      "_links": {
        "self": {
          "href": "/api/support/ems/events/malha-vsim1/4602"
        }
      }
    }
  ],
  "num_records": 1,
  "_links": {
    "self": {
      "href": "/api/support/ems/events?log_message=*disk*&message.severity=alert&max_records=1"
    },
    "next": {
      "href": "/api/support/ems/events?start.keytime=2022-03-18T06%3A37%3A46-04%3A00&start.node.name=malha-vsim1&start.index=4602&log_message=*disk*&message.severity=alert"
    }
  }
}
```

使用ONTAP REST API获取EMS配置

您可以检索ONTAP集群的当前EMS配置。您可以在更新配置或创建新EMS通知之前执行此操作。

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/support/ems

处理类型

同步

curl 示例

```
curl --request GET \  
--location "https://$FQDN_IP/api/support/ems" \  
--include \  
--header "Accept: */*" \  
--header "Authorization: Basic $BASIC_AUTH"
```

JSON 输出示例

```
{  
  "proxy_url": "https://proxyserver.mycompany.com",  
  "proxy_user": "proxy_user",  
  "mail_server": "mail@mycompany.com",  
  "_links": {  
    "self": {  
      "href": "/api/resourcelink"  
    }  
  },  
  "pubsub_enabled": "1",  
  "mail_from": "administrator@mycompany.com"  
}
```

使用ONTAP REST API创建EMS通知

您可以使用以下工作流创建新的EMS通知目标以接收选定事件消息。

第1步：配置系统范围的电子邮件设置

您可以通过问题描述执行以下API调用来配置系统范围的电子邮件设置。

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
patch	/API/support/ems

处理类型

同步

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，此步骤的 curl 示例还使用以下参数。

参数	Type	Required	Description
mail_from	查询	是的。	设置 from 字段。
mail_server	查询	是的。	配置目标SMTP邮件服务器。

curl 示例

```
curl --request PATCH \  
--location \  
"https://$FQDN_IP/api/support/ems?mail_from=administrator@mycompany.com&mail_server=mail@mycompany.com" \  
--include \  
--header "Accept: */*" \  
--header "Authorization: Basic $BASIC_AUTH"
```

第2步：定义邮件过滤器

您可以对API调用进行问题描述、以定义与消息匹配的筛选器规则。

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/support/EMS/筛选器

处理类型

同步

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，此步骤的 curl 示例还使用以下参数。

参数	Type	Required	Description
筛选器	body	是的。	包括筛选器配置的值。

curl 示例

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

JSON 输入示例

```
{
  "name": "test-filter",
  "rules.type": ["include"],
  "rules.message_criteria.severities": ["emergency"]
}
```

第3步：创建消息目标

您可以通过问题描述调用来创建消息目标。

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/support/EMS/目标

处理类型

同步

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，此步骤的 curl 示例还使用以下参数。

参数	Type	Required	Description
目标配置	body	是的。	包括事件目标的值。

curl 示例

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

JSON 输入示例

```
{
  "name": "test-destination",
  "type": "email",
  "destination": "administrator@mycompany.com",
  "filters.name": ["important-events"]
}
```

SVM

列出使用ONTAP REST API的SVM

您可以列出在ONTAP集群中定义的Storage Virtual Machine (SVM)。您可以在查找特定SVM的标识符时执行此操作、也可以在创建新SVM之前确保名称的唯一性。

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" \
--include \
--header "Accept: */*" \
--header "Authorization: Basic $BASIC_AUTH"
```

JSON 输出示例

```
{
  "records": [
    {
      "uuid": "71bd74f8-40dc-11ee-b51a-005056aee9fa",
      "name": "vs0",
      "_links": {
        "self": {
          "href": "/api/svm/svms/71bd74f8-40dc-11ee-b51a-005056aee9fa"
        }
      }
    }
  ],
  "num_records": 1,
  "_links": {
    "self": {
      "href": "/api/svm/svms"
    }
  }
}
```

版权信息

版权所有 © 2026 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，本文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。