ワークフロー
ONTAP automation

NetApp
January 18, 2026

目次

ワークフロー

ONTAP REST APIワークフローを使用するための準備

実際のONTAP環境でワークフローを使用する前に、ワークフローの構造と形式を理解しておく必要があります。

使用するワークフローで、ONTAPリリースがすべてのAPI呼び出しをサポートしていることを確認する必要があります。を参照してください "API リファレンス" を参照してください。

はじめに

a_workflow_ は、特定の管理タスクまたは目標を達成するために必要な 1 つ以上のステップのシーケンスです。ONTAPワークフローには、各タスクを実行するために必要な主要な手順とパラメータが含まれています。ONTAP自動化環境をカスタマイズするための出発点となります。

ステップタイプ

ONTAPワークフローの各手順は、次のいずれかのタイプです。

- REST API 呼び出し（ cURL や JSON の例などの詳細を含む）
- 別のONTAPワークフローの実行または起動
- その他の関連タスク（構成の決定など）

REST API呼び出し

ワークフローの手順のほとんどはREST API呼び出しです。これらの手順では、カールの例やその他の情報を含む一般的な形式を使用します。を参照してください "API リファレンス" を参照してください。

ワンステップのワークフロー

ワークフローには1つのステップのみを含めることができます。these_single-step workflows_は、複数のステップを含むワークフローとは少し異なります。たとえば、明示的なステップ名は削除されます。アクションまたは操作は、ワークフローのタイトルに基づいて明確にする必要があります。

入力変数

ワークフローは、任意のONTAP環境で使用できるように、できるだけ一般的なものにするように設計されています。そのため、REST API呼び出しでは、curlサンプルやその他の入力で変数が使用されます。REST API呼び出しは、さまざまなONTAP環境に簡単に適合させることができます。

ベース**URL**形式

ONTAP REST APIには、curlまたはプログラミング言語から直接アクセスできます。この場合、ベースURLは、ONTAPオンラインドキュメントまたはSystem Managerにアクセスするときに使用するURLとは異なります。

APIに直接アクセスする場合は、ドメインまたはIPアドレスに* api *を追加する必要があります。例：

```
https://ontap.demo-example.com/api
```

を参照してください "ONTAP REST APIにアクセスする方法" を参照してください。

共通の入力パラメータ

REST API呼び出しのほとんどでよく使用される入力パラメータがいくつかあります。これらのパラメータは、通常、個 々 のワークフローでは説明されていません。パラメータをよく理解しておく必要があります。を参照してください "API 要求を制御する入力変数" を参照してください。

特定のREST API呼び出しに追加のパラメータが必要な場合は、各ワークフローの* curlサンプルの追加の入力パラメータ*に記載されています。

変数の形式

このワークフローの例で使用されているID値やその他の変数は不透明であり、ONTAPクラスタごとに異なる場合があります。例を読みやすくするために、実際の値は使用しません。代わりに変数が使用されます。この方法では、一貫した形式と予約名のセットに基づいて、次のような利点があります。

- curlとjsonのサンプルは読みやすく、わかりやすくなっています。
- すべてのキーワードが同じ形式を使用しているため、すばやく識別できます。
- 値をコピーして再利用することはできないため、セキュリティ上の影響はありません。

変数はBashシェル環境で使用されるようにフォーマットされています。各変数はドル記号で始まり、必要に応じて二重引用符で囲みます。これにより、Bashに認識されるようになります。名前には常に大文字が使用されます。

ここでは、一般的な変数キーワードの一部を示します。このリストは完全なものではなく、必要に応じて追加の変数が使用されます。その意味はコンテキストに基づいて明確になる必要があります。

キーワード	を入力します	説明
$FQDN_IP	URL	ONTAP管理LIFの完全修飾ドメイン名またはIPアドレス。
$クラスタID	パス	UUIDv4の値。API処理を実行するONTAPクラスタを識別します。
$BASIC_AUTH	ヘッダー	HTTPベーシック認証に使用するクレデンシャル文字列。

JSONの入力例

POSTやPATCHを使用するREST API呼び出しなど、一部のREST API呼び出しでは、要求の本文にJSON入力が必要です。JSON入力の例は、わかりやすくするためにcurlの例とは別に表示されています。JSON入力例は、以下で説明するいずれかの方法で使用できます。

ローカルファイルに保存

JSON入力例をファイルにコピーしてローカルに保存できます。curlコマンドは、 `--data` ファイル名を示す値を持つパラメータ。 `@` プレフィックス。

curlの例の後に端末に貼り付けます

最初に、カールの例をコピーしてターミナルシェルに貼り付ける必要があります。次に、例を編集して、 `--data` 最後のパラメータをに置き換えます。 `--data-raw` パラメータ最後に、JSONの例をコピーして貼り付け、更新されたパラメータを使用してcurlコマンドに従うようにします。JSON入力の例では、一重引用符を使用して折り返します。

認証オプション

REST APIで使用できる主な認証手法はHTTPベーシック認証です。ONTAP 9.14以降では、トークンベースの認証と承認でOpen Authorization (OAuth 2.0)フレームワークを使用するオプションもあります。

HTTPベーシック認証

ベーシック認証を使用する場合は、各HTTP要求にユーザクレデンシャルを含める必要があります。クレデンシャルを送信する方法は2つあります。

HTTP要求ヘッダーを作成する

Authorizationヘッダーは手動で作成し、HTTP要求に含めることができます。これは、CLIでcurlコマンドを使用する場合、またはオートメーションコードでプログラミング言語を使用する場合に実行できます。手順の概要は次のとおりです。

1. ユーザとパスワードの値をコロンで連結します。

   ```
   admin:david123
   ```

2. 文字列全体をbase64に変換します。

   ```
   YWRtaW46ZGF2aWQxMjM=
   ```

3. 要求ヘッダーを作成します。

   ```
   Authorization: Basic YWRtaW46ZGF2aWQxMjM=
   ```

ワークフローカールの例には、このヘッダーと変数*$BASIC_AUTH *が含まれています。このヘッダーは、を使用する前に更新する必要があります。

curlパラメータを使用する

curlを使用する場合のもう1つのオプションは、Authorizationヘッダーを削除し、代わりにcurl * user *パラメーターを使用することです。例：

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

使用する環境に応じた適切なクレデンシャルに置き換える必要があります。クレデンシャルはbase64でエンコードされていません。このパラメータを指定してcurlコマンドを実行すると、文字列がエンコードされ、Authorizationヘッダーが生成されます。

OAuth 2.0

OAuth 2.0を使用する場合は、外部認可サーバーからアクセストークンを要求し、各HTTPリクエストに含める必要があります。基本的な手順の概要を次に示します。も参照してください "ONTAP OAuth 2.0実装の概要" OAuth 2.0の詳細とONTAPでの使用方法については、を参照してください。

ONTAP環境の準備

REST APIを使用してONTAPにアクセスする前に、ONTAP環境を準備して設定する必要があります。手順の概要は次のとおりです。

- ONTAPで保護されるリソースとクライアントを特定する

- 既存のONTAP RESTロールとユーザ定義の確認
- 認証サーバのインストールと設定
- クライアント許可定義の設計と設定
- ONTAPの設定とOAuth 2.0の有効化

アクセストークンのリクエスト

ONTAPと認可サーバーが定義されてアクティブになっている場合、OAuth 2.0トークンを使用してREST API呼び出しを行うことができます。最初のステップは、認可サーバーにアクセストークンを要求することです。これは、サーバに基づくいくつかの異なる技術のいずれかを使用して、ONTAPの外部で行われます。ONTAPでは、問題アクセストークンやリダイレクションは実行されません。

HTTP要求ヘッダーを作成する

アクセストークンを取得したら、Authorizationヘッダーを作成してHTTP要求に含めることができます。REST APIにアクセスするためにcurlとプログラミング言語のどちらを使用するかに関係なく、すべてのクライアント要求にヘッダーを含める必要があります。ヘッダーは次のように構成できます。

```
Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSld …
```

Bashでの例の使用

ワークフローカールの例を直接使用する場合は、変数に含まれる変数を環境に適した値に更新する必要があります。以下で説明するように、サンプルを手動で編集するか、Bashシェルに依存して置換を行うことができます。

> Bashを使用する利点の1つは、curlコマンドごとに1回ではなく、シェルセッションで変数値を一度だけ設定できることです。

手順

1. Linuxまたは同様のオペレーティングシステムで提供されているBashシェルを開きます。
2. 実行するcurlサンプルに含まれる変数値を設定します。例：

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

3. ワークフローページからcurlの例をコピーし、シェルターミナルに貼り付けます。
4. ENTER*を押すと、次の処理が実行されます。
   a. 設定した変数値を置き換えます。
   b. curlコマンドを実行します。

クラスタ

ONTAP REST APIを使用したクラスタ設定の取得

特定のフィールドを含むONTAPクラスタの設定を取得できます。この処理は、クラスタの状態の評価の一環として、または構成の更新前に行うことができます。

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
取得	/api/クラスタ

curlの例の追加入力パラメータ

この手順のcurlの例では、すべてのREST API呼び出しに共通のパラメータに加えて、次のパラメータも使用しています。

パラメータ	を入力します	必須	説明
フィールド	クエリ	いいえ	返される値を選択します。例： `contact` および `version`。

curlの例：クラスタの連絡先情報を取得する

この例では、単一のフィールドを取得する方法を示します。クラスタオブジェクトと設定全体を取得するには、 `fields` クエリパラメータ。

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

JSON 出力例

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

ONTAP REST APIを使用したクラスタ連絡先の更新

クラスタの連絡先情報を更新できます。要求は非同期で処理されるため、関連するバックグラウンドジョブが正常に完了したかどうかも確認する必要があります。

手順**1**：クラスタの連絡先情報を更新する

API呼び出しを問題して、クラスタの連絡先情報を更新できます。

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
パッチ	/api/クラスタ

処理のタイプ

非同期

カールの例

```
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 の入力例

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

JSON 出力例

ジョブオブジェクトが返されます。次の手順で使用するには、ジョブIDを保存する必要があります。

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

手順**2**：ジョブのステータスを取得する

ワークフローを実行 "ジョブインスタンスの取得" をクリックし、 `state` 値は `success`。

手順**3**：クラスタの連絡先情報を確認する

ワークフローを実行 "クラスタ構成を取得"。を設定する必要があります `fields` クエリパラメータ `contact`。

ONTAP REST APIを使用したジョブインスタンスの取得

特定のONTAPジョブのインスタンスを取得できます。通常、この操作は、ジョブおよび関連する処理が正常に完了したかどうかを確認するために行います。

ジョブオブジェクトのUUIDが必要です。このUUIDは通常、非同期要求の実行後に指定されます。また、 "ジョブオブジェクトを使用した非同期処理" ONTAP内部ジョブを操作する前に。

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
取得	/api/cluster/jobs/ {uuid}

処理のタイプ

同期

Curlの例の追加入力パラメータ

すべての REST API 呼び出しに共通するパラメータに加えて、この手順の curl の例では次のパラメータも使用されます。

パラメータ	を入力します	必須	説明
$JOB_ID	パス	はい。	要求されているジョブを識別するために必要です。

カールの例

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

JSON 出力例

stateの値とその他のフィールドは、返されるジョブオブジェクトに含まれます。この例のジョブは、ONTAPクラスタ更新の一環として実行されました。

```
{
  "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

ファイルセキュリティ権限

ONTAP REST APIを使用してファイルセキュリティと監査ポリシーを管理するための準備

ONTAPクラスタ内のSVMから使用可能なファイルの権限と監査ポリシーを管理できます。

概要

ONTAP では、システムアクセス制御リスト（ SACL ）と随意アクセス制御リスト（ DACL ）を使用してファイルオブジェクトに権限を割り当てます。ONTAP 9.9.1以降では、REST APIでSACLとDACLの権限の管理がサポートされます。APIを使用すると、ファイルセキュリティ権限の管理を自動化できます。多くの場合、複数のCLIコマンドやONTAPI（ZAPI）呼び出しではなく、1つのREST API呼び出しを使用できます。

> 9.9.1より前のONTAPリリースでは、CLIパススルー機能を使用して、SACLおよびDACL権限の管理を自動化できます。を参照してください "移行に関する考慮事項" および "ONTAP REST API でプライベート CLI パススルーを使用する" を参照してください。

REST APIを使用してONTAPファイルセキュリティサービスを管理する方法を示すワークフローの例をいくつか紹介します。ワークフローを使用してREST API呼び出しを実行する前に、 "ワークフローを使用する準備をします"。

Pythonを使用する場合は、スクリプトも参照してください。 "file_security_permissions.py" ファイルセキュリティアクティビティの一部を自動化する方法の例を参照してください。

ONTAP REST API コマンドと ONTAP CLI コマンドの比較

多くのタスクで、ONTAP REST APIを使用する場合、同等のONTAP CLIコマンドやONTAPI（ZAPI）呼び出しよりも少ない呼び出しで済みます。次の表に、API呼び出しと、各タスクに必要なCLIコマンドを示します。

ONTAP REST API	ONTAP CLI
「 get/protocols/file-security/effected-permissions/ 」	vserver security file-directory show-effected-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 」

関連情報

- "ファイル権限を示すPythonスクリプト"
- "ONTAP REST API を使用してファイルセキュリティ権限を簡単に管理できます"
- "ONTAP REST API でプライベート CLI パススルーを使用する"

ONTAP REST APIを使用して、ファイルに対して有効な権限を取得する

特定のファイルまたはフォルダに対して現在有効な権限を取得できます。

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
取得	/api/protocols/file-security/effective-permissions/｛svm.uuid｝/｛path｝

処理のタイプ

同期

curlの例の追加入力パラメータ

この手順のcurlの例では、すべてのREST API呼び出しに共通のパラメータに加えて、次のパラメータも使用しています。

パラメータ	を入力します	必須	説明
$SVM_ID	パス	はい。	これは、ファイルが含まれているSVMのUUIDです。
$file_path	パス	はい。	ファイルまたはフォルダへのパスです。

カールの例

```
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 出力例


```
{
  "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"
  ]
}
```


ONTAP REST APIを使用したファイルの監査情報の取得

特定のファイルまたはフォルダの監査情報を取得できます。

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
取得	/api/protocols/file-security/permissions/｛svm.uuid｝/｛path｝

処理のタイプ

同期

curlの例の追加入力パラメータ

この手順のcurlの例では、すべてのREST API呼び出しに共通のパラメータに加えて、次のパラメータも使用しています。

パラメータ	を入力します	必須	説明
$SVM_ID	パス	はい。	これは、ファイルが含まれているSVMのUUIDです。
$file_path	パス	はい。	ファイルまたはフォルダへのパスです。

カールの例

```
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 出力例

```
{
  "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"
}
```

ONTAP REST APIを使用してファイルに新しい権限を適用する

新しいセキュリティ記述子を特定のファイルまたはフォルダに適用できます。

手順**1**：新しい権限を適用する

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
投稿（ Post ）	/api/protocols/file-security/permissions/ ｛svm.uuid｝ / ｛path｝

処理のタイプ

非同期

curlの例の追加入力パラメータ

この手順のcurlの例では、すべてのREST API呼び出しに共通のパラメータに加えて、次のパラメータも使用しています。

パラメータ	を入力します	必須	説明
$SVM_ID	パス	はい。	これは、ファイルが含まれているSVMのUUIDです。
$file_path	パス	はい。	ファイルまたはフォルダへのパスです。

カールの例

```
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
'{ \"acls\": [ { \"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 出力例

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

手順**2**：ジョブのステータスを取得する

ワークフローを実行 "ジョブインスタンスの取得" をクリックし、 `state` 値は `success`。

ONTAP REST APIを使用したセキュリティ記述子情報の更新

特定のセキュリティ記述子を、プライマリ所有者、グループ、制御フラグなど、特定のファイルまたはフォルダに対して更新できます。

手順**1**：セキュリティ記述子を更新する

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
パッチ	/api/protocols/file-security/permissions/｛svm.uuid｝/｛path｝

処理のタイプ

非同期

curlの例の追加入力パラメータ

この手順のcurlの例では、すべてのREST API呼び出しに共通のパラメータに加えて、次のパラメータも使用しています。

パラメータ	を入力します	必須	説明
$SVM_ID	パス	はい。	これは、ファイルが含まれているSVMのUUIDです。
$file_path	パス	はい。	ファイルまたはフォルダへのパスです。

カールの例

```
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 出力例

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

手順2：ジョブのステータスを取得する

ワークフローを実行 "ジョブインスタンスの取得" をクリックし、 `state` 値は `success`。

ONTAP REST APIを使用してアクセス制御エントリを削除する

特定のファイルまたはフォルダから既存のAccess Control Entry（ACE；アクセス制御エントリ）を削除できます。変更はすべての子オブジェクトに伝播されます。

手順1：ACEの削除

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
削除	/api/protocols/file-security/permissions/｛svm.uuid｝/｛path｝

処理のタイプ

非同期

curlの例の追加入力パラメータ

この手順のcurlの例では、すべてのREST API呼び出しに共通のパラメータに加えて、次のパラメータも使用しています。

パラメータ	を入力します	必須	説明
$SVM_ID	パス	はい。	これは、ファイルが含まれているSVMのUUIDです。
$file_path	パス	はい。	ファイルまたはフォルダへのパスです。

カールの例

```
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 出力例

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

手順2：ジョブのステータスを取得する

ワークフローを実行 "ジョブインスタンスの取得" をクリックし、 `state` 値は `success`。

ネットワーキング

ONTAP REST APIを使用したIPインターフェイスの一覧表示

クラスタおよびSVMに割り当てられているIP LIFを取得できます。この操作は、ネットワーク設定を確認する場合や、別のLIFを追加する場合に実行します。

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
取得	/api/network/ip/interfaces

処理のタイプ

同期

Curlの例の追加入力パラメータ

すべての REST API 呼び出しに共通するパラメータに加えて、この手順の curl の例では次のパラメータも使用されます。

パラメータ	を入力します	必須	説明
フィールド	クエリ	いいえ	関連する設定値の限定されたリストを返します。

curlの例：すべてのLIFをデフォルトの設定値で返す

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

curlの例：特定の4つの設定値を持つすべてのLIFを返す

```
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"
```

JSON 出力例


```
{
  "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": "d9fce1a3-999e-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/d9fce1a3-999e-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-vsim-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-vsim-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"
    }
  }
}
```

セキュリティ

アカウント

ONTAP REST APIを使用してアカウントを表示する

アカウントのリストを取得できます。これは、セキュリティ環境を評価するため、または新しいアカウントを作成する前に行うことができます。

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
取得	/api/security/accounts（/api/security/アカウント）

処理のタイプ

同期

カールの例

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

JSON 出力例


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

証明書とキー

ONTAP REST APIを使用してインストールされている証明書を表示する

ONTAPクラスタにインストールされている証明書を表示できます。これは、特定の証明書が使用可能かどうかを確認したり、特定の証明書のIDを取得したりするために実行します。

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
取得	/api/security/certificates

curlの例の追加入力パラメータ

この手順のcurlの例では、すべてのREST API呼び出しに共通のパラメータに加えて、次のパラメータも使用しています。

パラメータ	を入力します	必須	説明
max_records	クエリ	いいえ	返されるレコードの数を指定します。

curlの例：3つの証明書を返す

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

JSON 出力例


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

ONTAP REST APIを使用した証明書のインストール

署名済みX.509証明書をONTAPクラスタにインストールできます。これは、強力な認証を必要とするONTAP機能またはプロトコルの設定の一環として行うことができます。

作業を開始する前に

インストールする証明書が必要です。また、必要に応じて中間証明書がインストールされていることを確認してください。

以下に示すJSON入力例を使用する前に、 `public_certificate` 環境の証明書を使用して値を設定します。

手順1：証明書をインストールする

API呼び出しを問題して証明書をインストールできます。

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
投稿（ Post ）	/api/security/certificates

cURLの例：ルート**CA**証明書をクラスタレベルでインストールする

```
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 の入力例

```
{
    "type": "server_ca",
    "public_certificate":
    "-----BEGIN CERTIFICATE-----
MIID0TCCArkCFGYdznvTVvaY1VZPNfy4yCCyPph6MA0GCSqGSIb3DQEBCwUAMIGk
MQswCQYDVQQGEwJVUzELMAkGA1UECAwCTkMxDDAKBgNVBAcMA1JUUDEWMBQGA1UE
CgwNT05UQVAgRXhhbXBsZTETMBEGA1UECwwKT05UQVAgOS4xNDEcMBoGA1UEAwwT
Ki5vbnRhcC1leGFtcGxlLmNvbTEvMC0GCSqGSIb3DQEJARYgZGF2aWQucGV0ZXJz
b25Ab250YXAtZXhhbXBsZS5jb20wHhcNMjMxMDA1MTUyOTE4WhcNMjQxMDA0MTUy
OTE4WjCBpDELMAkGA1UEBhMCVVMxCzAJBgNVBAgMAk5DMQwwCgYDVQQHDANSVFAx
FjAUBgNVBAoMDU9OVEFQIEV4YW1wbGUxEzARBgNVBAsMCk9OVEFQIDkuMTQxHDAa
BgNVBAMMEyoub250YXAtZXhhbXBsZS5jb20xLzAtBgkqhkiG9w0BCQEWIGRhdmlk
LnBldGVyc29uQG9udGFwLWV4YW1wbGUuY29tMIIBIjANBgkqhkiG9w0BAQEFAAOC
AQ8AMIIBCgKCAQEAxQgy8mhb1Jhkf0D/MBodpzgW0aSp2jGbWJ+Zv2G8BXkp1762
dPHRkv1hnx9JvwkK4Dba05GiCiD5t3gjH/jUQMSFb+VwDbVmubVFnXjkm/4Q7sea
tMtA/ZpQdZbQFZ5RKtdWz7dzzPYEl2x8Q1Jc8Kh7NxERNMtgupGWZZn7mfXKYr4O
N/+vgahIhDibS8YK5rf1w6bfmrik9E2D+PEab9DX/1DL5RX4tZ1H2OkyN2UxoBR6
Fq7l6n1Hi/5yR0Oi1xStN6sO7EPoGak+KSlK41q+EcIKRo0bP4mEQp8WMjJuiTkb
5MmeYoIpWEUgJK7S0M6Tp/3bTh2CST3AWxiNxQIDAQABMA0GCSqGSIb3DQEBCwUA
A4IBAQABfBqOuROmYxdfrj93OyIiRoDcoMzvo8cHGNUsuhnlBDnL2O3qhWEs97s0
mIy6zFMGnyNYa0t4i1cFsGDKP/JuljmYHjvv+2lHWnxHjTo7AOQCnXmQH5swoDbf
o1Vjqz8Oxz+PRJ+PA3dF5/8zqaAR6QreAN/iFR++6nUq1sbbM7w03tthBVMgo/h1
E9I2jVOZsqMFujm2CYfMs4XkZtrYmN6nZA8JcUpDjIWcAVbQYurMnna9r42oS3GB
WB/FE9n+P+FfJyHJ93KGcCXbH5RF2pi3wLlHilbvVuCjLRrhJ8U20I5mZoiXvABc
IpYuBcuKXLwAarhDEacXttVjC+Bq
-----END CERTIFICATE-----"
}
```

手順**2**：証明書がインストールされたことを確認する

ワークフローを実行 "インストールされている証明書を表示" 証明書が利用可能であることを確認します。

RBAC

ONTAP REST APIを使用して**RBAC**を使用するための準備

ONTAP RBAC機能は、環境に応じていくつかの方法で使用できます。このセクションでは、いくつかの一般的なシナリオをワークフローとして紹介します。いずれの場合も、特定のセキュリティおよび管理上の目標に焦点を当てています。

ロールを作成してONTAPユーザアカウントにロールを割り当てる前に、次に示す主なセキュリティ要件とオプションを確認して準備しておく必要があります。また、次のWebサイトで一般的なワークフローの概念を確認してください。 "ワークフローを使用する準備をします"。

どの**ONTAP** リリースを使用していますか？

ONTAP リリースによって、使用可能なRESTエンドポイントとRBAC機能が決まります。

保護対象のリソースと範囲を特定

保護対象のリソースまたはコマンドとその範囲（クラスタまたはSVM）を特定する必要があります。

ユーザにはどのようなアクセス権が必要ですか。

リソースと範囲を特定したら、許可するアクセスレベルを決定する必要があります。

ユーザはどのように**ONTAP** にアクセスしますか。

ユーザはREST API、CLI、またはその両方を使用してONTAP にアクセスできます。

組み込みの役割の**1**つで十分か、またはカスタムの役割が必要か。

既存の組み込みロールを使用する方が便利ですが、必要に応じて新しいカスタムロールを作成することもできます。

どのような種類の役割が必要ですか？

セキュリティ要件とONTAP アクセスに基づいて、RESTと従来のどちらのロールを作成するかを選択する必要があります。

ロールの作成

ONTAP REST APIを使用して**SVM**ボリューム処理へのアクセスを制限する

SVM内でのストレージボリュームの管理を制限するロールを定義できます。

このワークフローについて

最初に、クローニングを除くすべての主要なボリューム管理機能へのアクセスを許可するために、トラディショナルロールが作成されます。ロールは次の特性で定義されます。

- GET、CREATE、MODIFY、DELETEなどのCRUDボリューム操作をすべて実行できる
- ボリュームクローンを作成できません

その後、必要に応じてロールを更新できます。このワークフローでは、2番目の手順でロールが変更され、ユーザがボリュームクローンを作成できるようになります。

手順**1**：ロールを作成する

API呼び出しを問題してRBACロールを作成できます。

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
投稿（ Post ）	/api/security/rolesのように入力します

カールの例

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

JSON の入力例

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

手順**2**：ロールを更新する

API呼び出しを問題して既存のロールを更新できます。

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
投稿（ Post ）	/api/security/rolesのように入力します

curlの例の追加入力パラメータ

この手順のcurlの例では、すべてのREST API呼び出しに共通のパラメータに加えて、次のパラメータも使用しています。

パラメータ	を入力します	必須	説明
$SVM_ID	パス	はい。	これは、ロールの定義が含まれているSVMのUUIDです。
$ロール名	パス	はい。	更新するSVM内のロールの名前を指定します。

カールの例

```
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 の入力例

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

ONTAP REST APIを使用したデータ保護の管理

ユーザに提供できるデータ保護機能は限られています。

このワークフローについて

従来のロールは、次の特性で定義されます。

- Snapshotの作成と削除、およびSnapMirror関係の更新が可能です
- ボリュームやSVMなどの上位のオブジェクトを作成または変更することはできません

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
投稿（ Post ）	/api/security/rolesのように入力します

カールの例

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

JSON の入力例

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

ONTAP REST APIを使用した**ONTAP**レポートの生成を許可する

ONTAP レポートを生成する機能をユーザに提供するRESTロールを作成できます。

このワークフローについて

作成されるロールは、次の特性で定義されます。

- 容量とパフォーマンス（ボリューム、qtree、LUN、アグリゲート、ノード、 SnapMirror関係の場合）
- 上位のオブジェクト（ボリュームやSVMなど）を作成または変更できない

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
投稿（ Post ）	/api/security/rolesのように入力します

カールの例

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

JSON の入力例

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

ONTAP REST APIを使用してロールを持つユーザを作成する

このワークフローを使用して、RESTロールを関連付けたユーザを作成できます。

このワークフローについて

このワークフローには、カスタムRESTロールを作成して新しいユーザアカウントに関連付けるために必要な一般的な手順が含まれています。ユーザとロールの両方にSVMスコープがあり、特定のデータSVMに関連付けられています。一部の手順はオプションである場合もあれば、環境に応じて変更する必要がある場合もあります。

手順**1**：クラスタ内のデータ**SVM**をリストする

次のREST API呼び出しを実行して、クラスタ内のSVMを表示します。各SVMのUUIDと名前が出力に表示されます。

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
取得	/api/SVM/SVMs

カールの例

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

完了後

リストから必要なSVMを選択し、新しいユーザとロールを作成します。

手順2：SVMに定義されているユーザを表示する

選択したSVMで定義されているユーザを表示するために、次のREST API呼び出しを実行します。SVMは、ownerパラメータを使用して識別できます。

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
取得	/api/security/accounts（/api/security/アカウント）

カールの例

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

完了後

SVMですでに定義されているユーザに基づいて、新しいユーザの一意の名前を選択します。

手順3：SVMに定義されているRESTロールを表示する

次のREST API呼び出しを実行し、選択したSVMで定義されているロールをリストします。SVMは、ownerパラメータを使用して識別できます。

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
取得	/api/security/rolesのように入力します

カールの例

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

完了後

SVMですでに定義されているロールに基づいて、新しいロールに一意の名前を選択します。

手順4：カスタムRESTロールを作成する

次のREST API呼び出しを実行して、SVMでカスタムのRESTロールを作成します。最初は権限を1つしか持たず、すべてのアクセスが拒否されるようにするために、このロールにはデフォルトのアクセスである* none *が設定されます。

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
投稿（ Post ）	/api/security/rolesのように入力します

カールの例

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

JSON の入力例

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

完了後

必要に応じて、手順3をもう一度実行して新しいロールを表示します。ONTAP CLIでもロールを表示できます。

手順**5**：権限を追加してロールを更新する

必要に応じて権限を追加してロールを変更するには、次のREST API呼び出しを実行します。

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
投稿（ Post ）	/api/security/roles/｛owner.uuid｝/｛name｝/privileges

curlの例の追加入力パラメータ

この手順のcurlの例では、すべてのREST API呼び出しに共通のパラメータに加えて、次のパラメータも使用しています。

パラメータ	を入力します	必須	説明
$SVM_ID	パス	はい。	ロールの定義が含まれているSVMのUUID。
$ロール名	パス	はい。	更新するSVM内のロールの名前を指定します。

カールの例

```
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 の入力例

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

完了後

必要に応じて、手順3をもう一度実行して新しいロールを表示します。ONTAP CLIでもロールを表示できます。

手順**6**：ユーザを作成する

ユーザアカウントを作成するには、次のREST API呼び出しを実行します。上で作成したロール*dprole1*は、新しいユーザに関連付けられています。

ロールを指定せずにユーザを作成できます。この場合、ユーザにはデフォルトのロール（ `admin` または `vsadmin`）ユーザがクラスタスコープとSVMスコープのどちらで定義されているかに応じて変わります。別のロールを割り当てるには、ユーザを変更する必要があります。

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
投稿（ Post ）	/api/security/accounts（/api/security/アカウント）

カールの例

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

JSON の入力例

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

完了後

SVM管理インターフェイスにサインインするには、新しいユーザのクレデンシャルを使用します。

ストレージ

ONTAP REST APIを使用してアグリゲートを表示する

クラスタ内のアグリゲートのリストを取得できます。これは、利用率とパフォーマンスを評価するために行うことができます。

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
取得	/API/ストレージ/ディスク

処理のタイプ

同期

Curlの例の追加入力パラメータ

すべての REST API 呼び出しに共通するパラメータに加えて、この手順の curl の例では次のパラメータも使用されます。

パラメータ	を入力します	必須	説明
node.name	クエリ	いいえ	を使用して、各アグリゲートが接続されているノードを特定できます。

curlの例：デフォルトの設定値を使用してすべてのアグリゲートを返す

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

curlの例：特定の設定値を持つすべてのアグリゲートを返す

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

JSON 出力例


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


ONTAP REST APIを使用してディスクを表示する

クラスタ内のディスクのリストを取得できます。この操作は、アグリゲートの作成時に使用する1つ以上のスペアを特定する場合に行います。

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
取得	/API/ストレージ/ディスク

処理のタイプ

同期

Curlの例の追加入力パラメータ

すべての REST API 呼び出しに共通するパラメータに加えて、この手順の curl の例では次のパラメータも使用されます。

パラメータ	を入力します	必須	説明
状態	クエリ	いいえ	を使用すると、新しいアグリゲートに使用できるスペアディスクを特定できます。

curlの例：すべてのディスクを返す

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

curlの例：スペアディスクを返す

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

JSON 出力例


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


サポート

EMS

ONTAP REST APIを使用してEMSサポートサービスを管理する準備

ONTAPクラスタのイベント管理システム（EMS）処理を設定し、必要に応じてEMSメッセージを取得できます。

概要

ここでは、ONTAP EMSサービスの使用方法を示すワークフローの例をいくつか紹介します。ワークフローを使用してREST API呼び出しを実行する前に、 "ワークフローを使用する準備をします"。

Pythonを使用する場合は、スクリプトも参照してください。 "events.py" EMS関連の一部のアクティビティを自動化する方法の例を参照してください。

ONTAP REST API コマンドと ONTAP CLI コマンドの比較

多くのタスクでは、ONTAP REST APIを使用すると、同等のONTAP CLIコマンドよりも少ない呼び出しで済みます。次の表に、API呼び出しと、各タスクに必要なCLIコマンドを示します。

ONTAP REST API	ONTAP CLI
GET /support/ems	event config show
ポスト/サポート/ EMS /デスティネーション	1. イベント通知の送信先を作成します 2. イベント通知は "' を作成します
「 get/support/ems/events 」	「 event log show 」を参照してください
「 POST/support/ems/filters 」と入力します	1. 'event filter create -filter-name <filtername>` 2. 'event filter rule add-filter-name <filtername>`

関連情報

- "EMSを示すPythonスクリプト"
- "ONTAP REST API ：重大度の高いイベントの通知を自動化します"

ONTAP REST APIを使用したEMSログイベントの一覧表示

すべてのイベント通知メッセージを取得することも、特定の特性を持つメッセージのみを取得することもできます。

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
取得	/api/support/ems/events

処理のタイプ

同期

Curlの例の追加入力パラメータ

すべての REST API 呼び出しに共通するパラメータに加えて、この手順の curl の例では次のパラメータも使用されます。

パラメータ	を入力します	必須	説明
フィールド	クエリ	いいえ	応答に含める特定のフィールドを要求するために使用されます。
max_records	クエリ	いいえ	1回の要求で返されるレコード数を制限するために使用できます。
LOG_MESSAGE	クエリ	いいえ	特定のテキスト値を検索し、一致するメッセージのみを返します。
message.severity	クエリ	いいえ	返されるメッセージは、次のような特定の重大度のメッセージだけに制限します。 `alert`。

curlの例:最新のメッセージと名前の値を返す

```
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の例：特定のテキストと重大度を含むメッセージを返す

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

JSON 出力例


```
{
  "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_rec
ords=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"
    }
  }
}
```

ONTAP REST APIを使用したEMS設定の取得

ONTAPクラスタの現在のEMS設定を取得できます。この処理は、設定を更新する前や新しいEMS通知を作成する前に実行します。

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
取得	/api/support/ems

処理のタイプ

同期

カールの例

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

JSON 出力例

```
{
  "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"
}
```

ONTAP REST APIを使用したEMS通知の作成

次のワークフローを使用して、選択したイベントメッセージを受信する新しいEMS通知の送信先を作成できます。

手順1：システム全体のEメール設定を構成する

次のAPI呼び出しを問題して、システム全体のEメール設定を行うことができます。

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
パッチ	/api/support/ems

処理のタイプ

同期

Curlの例の追加入力パラメータ

すべての REST API 呼び出しに共通するパラメータに加えて、この手順の curl の例では次のパラメータも使用されます。

パラメータ	を入力します	必須	説明
メールの差出人	クエリ	はい。	を設定します。 `from` フィールドをクリックします。
メールサーバ	クエリ	はい。	ターゲットのSMTPメールサーバを設定します。

カールの例

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

手順**2**：メッセージフィルタを定義する

API呼び出しを問題して、メッセージに一致するフィルタルールを定義できます。

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
投稿（ Post ）	/api/support/ems/filters

処理のタイプ

同期

Curlの例の追加入力パラメータ

すべての REST API 呼び出しに共通するパラメータに加えて、この手順の curl の例では次のパラメータも使用されます。

パラメータ	を入力します	必須	説明
フィルタ	ボディ（Body）	はい。	フィルタ設定の値が含まれます。

カールの例

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

JSON の入力例

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

手順**3**：メッセージの送信先を作成する

API呼び出しを問題して、メッセージの送信先を作成できます。

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
投稿（Post）	/api/support/ems/destinations

処理のタイプ

同期

Curlの例の追加入力パラメータ

すべての REST API 呼び出しに共通するパラメータに加えて、この手順の curl の例では次のパラメータも使用されます。

パラメータ	を入力します	必須	説明
デスティネーションの設定	ボディ（Body）	はい。	イベントの送信先の値が含まれます。

カールの例

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

JSON の入力例

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

SVM

ONTAP REST APIを使用して**SVM**を表示する

ONTAPクラスタ内で定義されているStorage Virtual Machine（SVM）をリストできます。この処理は、特定のSVMの識別子を検索する場合や、新しいSVMを作成する前に名前を一意にする場合に実行します。

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
取得	/api/SVM/SVMs

カールの例

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

JSON 出力例


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

著作権に関する情報

商標に関する情報