Flux de travail

ONTAP Automation

NetApp
July 11, 2024

Sommaire

Flux de travail

Préparez l'utilisation des workflows

Vous devez connaître la structure et le format des flux de travail avant de les utiliser avec un déploiement ONTAP en direct.

Vérifiez que votre version d'ONTAP prend en charge tous les appels d'API dans les workflows que vous prévoyez d'utiliser. Voir "Référence API" pour en savoir plus.

Introduction

Un *workflow* est une séquence d'une ou de plusieurs étapes nécessaires à la réalisation d'une tâche ou d'un objectif administratif spécifique. Les workflows ONTAP incluent les étapes clés et les paramètres dont vous avez besoin pour mener à bien chaque tâche. Elles constituent un point de départ pour la personnalisation de votre environnement d'automatisation ONTAP.

Types d'étape

Chaque étape d'un flux de travail ONTAP est l'un des types suivants :

- Appel d'API REST (avec des détails tels que des exemples Curl et JSON)
- Exécuter ou appeler un autre flux de travail ONTAP
- Tâches liées diverses (telles que la prise d'une décision de configuration)

Appels API REST

La plupart des étapes du workflow sont des appels d'API REST. Ces étapes utilisent un format commun qui inclut un exemple de boucle et d'autres informations. Voir la "Référence API" Pour plus de détails sur les appels de l'API REST.

Flux de production en une seule étape

Un flux de travail ne peut contenir qu'une seule étape. Ces flux de travail à une seule étape sont formatés légèrement différemment des flux de travail contenant plusieurs étapes. Par exemple, le nom explicite de l'étape est supprimé. L'action ou l'opération doit être claire en fonction du titre du flux de travail.

Variables d'entrée

Les flux de travail sont conçus pour être aussi généraux que possible et peuvent donc être utilisés dans n'importe quel environnement ONTAP. Dans cet esprit, les appels de l'API REST utilisent des variables dans les exemples de boucles et d'autres entrées. Les appels de l'API REST peuvent ensuite être facilement adaptés à différents environnements ONTAP.

Format d'URL de base

Vous pouvez accéder directement à l'API REST ONTAP via curl ou un langage de programmation. Dans ce cas, l'URL de base est différente de l'URL que vous utilisez lorsque vous accédez à la documentation en ligne de ONTAP ou à System Manager.

Lorsque vous accédez directement à l'API, vous devez ajouter **api** au domaine ou à l'adresse IP. Par exemple :

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

Voir "Comment accéder à l'API REST de ONTAP" pour en savoir plus.

Paramètres d'entrée communs

Il existe plusieurs paramètres d'entrée couramment utilisés avec la plupart des appels API REST. Ces paramètres ne sont généralement pas décrits dans chaque flux de travail. Vous devez connaître les paramètres. Voir "Variables d'entrée contrôlant une requête API" pour en savoir plus.

Si des paramètres supplémentaires sont nécessaires pour un appel d'API REST spécifique, ils sont inclus dans la section **Paramètres d'entrée supplémentaires pour l'exemple de boucle** pour chaque flux de travail.

Format variable

Les valeurs d'ID et les autres variables utilisées avec les exemples de workflow sont opaques et peuvent varier en fonction du cluster ONTAP. Pour améliorer la lisibilité des exemples, les valeurs réelles ne sont pas utilisées. Les variables sont utilisées à la place. Cette approche, basée sur un format et un ensemble cohérents de noms réservés, présente plusieurs avantages, notamment :

- Les échantillons Curl et JSON sont plus lisibles et plus faciles à comprendre.
- Comme tous les mots-clés utilisent le même format, vous pouvez rapidement les identifier.
- Il n'y a pas d'exposition de sécurité car les valeurs ne peuvent pas être copiées et réutilisées.

Les variables sont formatées pour être utilisées dans un environnement shell Bash. Chaque variable commence par un signe dollar et est placée entre guillemets si nécessaire. Cela les rend reconnaissables à Bash. La casse supérieure est toujours utilisée pour les noms.

Voici quelques mots clés de variable communs. Cette liste n'est pas exhaustive et d'autres variables sont utilisées si nécessaire. Leur signification devrait être évidente sur la base du contexte.

Mot-clé	Type	Description
$FQDN_IP	URL	Nom de domaine complet ou adresse IP du LIF de gestion ONTAP.
$CLUSTER_ID	Chemin	Valeur UUIDv4 identifiant le cluster ONTAP sur lequel s'exécutent les opérations de l'API.
$BASIC_AUTH	En-tête	Chaîne d'informations d'identification utilisée pour l'authentification de base HTTP.

Exemples d'entrée JSON

Certains appels de l'API REST, tels que ceux utilisant POST ou PATCH, nécessitent une entrée JSON dans le corps de la requête. Pour plus de clarté, les exemples d'entrée JSON sont présentés séparément des exemples de boucles. Vous pouvez utiliser les exemples d'entrée JSON avec l'une des techniques décrites ci-dessous.

Enregistrer dans le fichier local

Vous pouvez copier l'exemple d'entrée JSON dans un fichier et l'enregistrer localement. La commande curl fait référence au fichier utilisant le `--data` paramètre avec la valeur indiquant le nom du fichier avec un `@` préfixe.

Coller dans la borne après l'exemple de courbure

Vous devez tout d'abord copier et coller l'exemple de boucle dans une coque de terminal. Modifiez ensuite l'exemple pour supprimer complètement le `--data` à la fin du paramètre et remplacez-le par le `--data-raw` paramètre. Enfin, copiez et collez dans l'exemple JSON afin qu'il suive la commande curl avec le paramètre mis à jour. Vous devez utiliser des guillemets simples pour envelopper l'exemple d'entrée JSON.

Options d'authentification

La technique d'authentification principale disponible pour l'API REST est l'authentification de base HTTP. À partir de ONTAP 9.14, vous avez également la possibilité d'utiliser l'infrastructure d'autorisation ouverte (OAuth 2.0) avec authentification et autorisation basées sur des jetons.

Authentification de base HTTP

Lors de l'utilisation de l'authentification de base, les informations d'identification de l'utilisateur doivent être incluses avec chaque requête HTTP. Il existe deux options pour envoyer les informations d'identification.

Construisez l'en-tête de requête HTTP

Vous pouvez construire manuellement l'en-tête autorisation et l'inclure aux requêtes HTTP. Cela peut être fait lors de l'utilisation d'une commande curl dans l'interface de ligne de commande ou d'un langage de programmation avec votre code d'automatisation. Les étapes générales comprennent :

1. Concaténez les valeurs d'utilisateur et de mot de passe avec deux points :

   ```
   admin:david123
   ```

2. Convertissez la chaîne entière en base64 :

   ```
   YWRtaW46ZGF2aWQxMjM=
   ```

3. Construisez l'en-tête de la demande :

   ```
   Authorization: Basic YWRtaW46ZGF2aWQxMjM=
   ```

Les exemples de boucles de flux de travail incluent cet en-tête avec la variable **$BASIC_AUTH** que vous devez mettre à jour avant d'utiliser.

Utilisez un paramètre de courbure

Une autre option lors de l'utilisation de curl consiste à supprimer l'en-tête autorisation et à utiliser le paramètre curl **user** à la place. Par exemple :

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

Vous devez remplacer les informations d'identification appropriées pour votre environnement. Les informations d'identification ne sont pas codées en base64. Lors de l'exécution de la commande curl avec ce paramètre, la chaîne est codée et l'en-tête autorisation est généré pour vous.

OAuth 2.0

Lorsque vous utilisez OAuth 2.0, vous devez demander un jeton d'accès à un serveur d'autorisation externe et l'inclure à chaque requête HTTP. Les étapes générales de base sont décrites ci-dessous. Voir aussi "Présentation de la mise en œuvre de ONTAP OAuth 2.0" Pour plus d'informations sur OAuth 2.0 et sur son utilisation avec ONTAP.

Préparez votre environnement ONTAP

Avant d'utiliser l'API REST pour accéder à ONTAP, vous devez préparer et configurer l'environnement ONTAP. À un niveau élevé, les étapes comprennent :

- Identifier les ressources et les clients protégés par ONTAP
- Vérifiez le rôle REST ONTAP et les définitions d'utilisateur existantes
- Installez et configurez le serveur d'autorisation
- Concevoir et configurer les définitions d'autorisation client
- Configurez ONTAP et activez OAuth 2.0

Demander un jeton d'accès

Avec ONTAP et le serveur d'autorisation défini et actif, vous pouvez effectuer un appel d'API REST à l'aide d'un jeton OAuth 2.0. La première étape consiste à demander un jeton d'accès au serveur d'autorisation. Cette opération est effectuée en dehors de ONTAP en utilisant l'une des différentes techniques basées sur le serveur. ONTAP n'émet pas de tokens d'accès ni n'effectue de redirection.

Construisez l'en-tête de requête HTTP

Après avoir obtenu un jeton d'accès, vous pouvez construire un en-tête autorisation et l'inclure aux requêtes HTTP. Que vous utilisiez curl ou un langage de programmation pour accéder à l'API REST, vous devez inclure l'en-tête à chaque demande client. Vous pouvez construire l'en-tête comme suit :

```
Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSld …
```

En utilisant les exemples avec Bash

Si vous utilisez directement les exemples de boucles de flux de travail, vous devez mettre à jour les variables qu'ils contiennent avec les valeurs appropriées à votre environnement. Vous pouvez modifier manuellement les exemples ou vous appuyer sur le shell de hachage pour effectuer la substitution pour vous, comme décrit ci-dessous.

L'un des avantages de Bash est que vous pouvez définir les valeurs de variable une fois dans une session shell au lieu d'une fois par commande curl.

Étapes

1. Ouvrez le shell Bash fourni avec Linux ou un système d'exploitation similaire.
2. Définissez les valeurs variables incluses dans l'exemple de boucle que vous souhaitez exécuter. Par exemple :

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

3. Copiez l'exemple de boucle depuis la page de flux de travail et collez-le dans le terminal shell.
4. Appuyez sur **ENTER** pour effectuer les opérations suivantes :
   a. Remplacez les valeurs de variable que vous avez définies
   b. Exécutez la commande curl

Cluster

Obtenez la configuration du cluster

Vous pouvez récupérer la configuration d'un cluster ONTAP avec des champs spécifiques. Vous pouvez le faire dans le cadre de l'évaluation de l'état du cluster ou avant la mise à jour de la configuration.

Méthode HTTP et noeud final

Cet appel d'API REST utilise la méthode et le point de terminaison suivants.

Méthode HTTP	Chemin
OBTENEZ	/api/cluster

Paramètres d'entrée supplémentaires pour les exemples de boucles

Outre les paramètres communs à tous les appels API REST, les paramètres suivants sont également utilisés dans l'exemple curl de cette étape.

Paramètre	Type	Obligatoire	Description
champs	Requête	Non	Sélectionnez les valeurs que vous souhaitez renvoyer. Voici quelques exemples `contact` et `version`.

Exemple curl : permet de récupérer les informations de contact du cluster

Cet exemple illustre comment récupérer un seul champ. Pour obtenir l'ensemble de l'objet et de la configuration du cluster, vous devez supprimer le `fields` paramètre de requête.

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

Exemple de sortie JSON

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

Mettre à jour le contact du cluster

Vous pouvez mettre à jour les coordonnées d'un cluster. Étant donné que la demande est traitée de manière asynchrone, vous devez également déterminer si la tâche d'arrière-plan associée s'est terminée avec succès.

Étape 1 : mettez à jour les coordonnées du cluster

Vous pouvez émettre un appel d'API pour mettre à jour les informations de contact du cluster.

Méthode HTTP et noeud final

Cet appel d'API REST utilise la méthode et le point de terminaison suivants.

Méthode HTTP	Chemin
CORRECTIF	/api/cluster

Type de traitement

Asynchrone

Exemple de boucle

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

Exemple d'entrée JSON

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

Exemple de sortie JSON

Un objet de travail est renvoyé. Vous devez enregistrer l'identifiant du travail pour l'utiliser à l'étape suivante.

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

Étape 2 : récupération de l'état du travail

Exécutez le flux de travail "Obtenir l'instance de travail" et confirmez le `state` la valeur est `success`.

Étape 3 : confirmez les coordonnées du cluster

Exécutez le flux de travail "Obtenez la configuration du cluster". Vous devez définir le `fields` interroger le paramètre sur `contact`.

Obtenir l'instance de travail

Vous pouvez récupérer l'instance d'un travail ONTAP spécifique. Vous devez généralement effectuer cette opération pour déterminer si le travail et l'opération associée ont réussi.

Vous avez besoin de l'UUID de l'objet de travail, généralement fourni après l'émission d'une requête asynchrone. Consultez également "Traitement asynchrone à l'aide de l'objet travail" Avant de travailler avec des travaux internes ONTAP.

Méthode HTTP et noeud final

Cet appel d'API REST utilise la méthode et le point de terminaison suivants.

Méthode HTTP	Chemin
OBTENEZ	/api/cluster/jobs/{uuid}

Type de traitement

Synchrone

Paramètres d'entrée supplémentaires pour les exemples Curl

Outre les paramètres communs à tous les appels API REST, les paramètres suivants sont également utilisés dans les exemples de boucles pour cette étape.

Paramètre	Type	Obligatoire	Description
$JOB_ID	Chemin	Oui.	Nécessaire pour identifier le travail demandé.

Exemple de boucle

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

Exemple de sortie JSON

La valeur d'état et d'autres champs sont inclus dans l'objet de travail renvoyé. Dans cet exemple, la tâche a été exécutée dans le cadre de la mise à jour d'un cluster ONTAP.

```
{
  "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

Autorisations de sécurité des fichiers

Préparez-vous à gérer la sécurité des fichiers et les stratégies d'audit

Vous pouvez gérer les autorisations et les règles d'audit pour les fichiers disponibles via les SVM au sein d'un cluster ONTAP.

Présentation

ONTAP utilise les listes de contrôle d'accès système (CLS) et les listes de contrôle d'accès discrétionnaire (listes ACL) pour attribuer des autorisations aux objets de fichier. Depuis ONTAP 9.9.1, l'API REST prend en charge la gestion des autorisations SACL et DACL. Vous pouvez utiliser l'API pour automatiser l'administration des autorisations de sécurité des fichiers. Dans la plupart des cas, vous pouvez utiliser un seul appel d'API REST au lieu de plusieurs commandes CLI ou appels ONTAPI (ZAPI).

> Pour les versions ONTAP antérieures à la version 9.9.1, vous pouvez automatiser l'administration des autorisations SACL et DACL à l'aide de la fonction de passerelle CLI. Voir "Considérations relatives à la migration" et "Utilisation de la passerelle CLI privée avec l'API REST de ONTAP" pour en savoir plus.

Plusieurs exemples de workflows sont disponibles pour illustrer la manière de gérer les services de sécurité des fichiers ONTAP à l'aide de l'API REST. Avant d'utiliser les flux de travail et d'émettre l'un des appels de l'API REST, assurez-vous de passer en revue "Préparez l'utilisation des workflows".

Si vous utilisez Python, consultez également le script "file_security_permissions.py" pour des exemples d'automatisation de certaines activités de sécurité des fichiers.

Comparaison des commandes de l'API REST ONTAP et de l'interface CLI ONTAP

Pour de nombreuses tâches, l'utilisation de l'API REST ONTAP requiert moins d'appels que les commandes CLI ONTAP ou les appels ONTAPI (ZAPI) équivalents. Le tableau ci-dessous présente une liste d'appels API et l'équivalent des commandes CLI nécessaires à chaque tâche.

L'API REST DE ONTAP	INTERFACE DE LIGNE DE COMMANDES DE ONTAP
`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`

Informations associées

- "Script Python illustrant les autorisations de fichier"
- "Gestion simplifiée des autorisations de sécurité de fichiers avec les API REST ONTAP"
- "Utilisation de la passerelle CLI privée avec l'API REST de ONTAP"

Obtenez les autorisations efficaces pour un fichier

Vous pouvez récupérer les autorisations effectives actuelles pour un fichier ou un dossier spécifique.

Méthode HTTP et noeud final

Cet appel d'API REST utilise la méthode et le point de terminaison suivants.

Méthode HTTP	Chemin
OBTENEZ	/api/protocoles/sécurité-fichier/autorisations-effectives/{svm.uuid}/{path}

Type de traitement

Synchrone

Paramètres d'entrée supplémentaires pour les exemples de boucles

Outre les paramètres communs à tous les appels API REST, les paramètres suivants sont également utilisés dans l'exemple curl de cette étape.

Paramètre	Type	Obligatoire	Description
$SVM_ID	Chemin	Oui.	Il s’agit de l’UUID du SVM contenant le fichier.
$FILE_PATH	Chemin	Oui.	Il s’agit du chemin d’accès au fichier ou au dossier.

Exemple de boucle

```
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"
```

Exemple de sortie JSON


```
{
  "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"
  ]
}
```


Obtenez les informations d'audit d'un fichier

Vous pouvez récupérer les informations d'audit d'un fichier ou d'un dossier spécifique.

Méthode HTTP et noeud final

Cet appel d'API REST utilise la méthode et le point de terminaison suivants.

Méthode HTTP	Chemin
OBTENEZ	/api/protocoles/sécurité-fichier/permissions/{svm.uuid}/{path}

Type de traitement

Synchrone

Paramètres d'entrée supplémentaires pour les exemples de boucles

Outre les paramètres communs à tous les appels API REST, les paramètres suivants sont également utilisés dans l'exemple curl de cette étape.

Paramètre	Type	Obligatoire	Description
$SVM_ID	Chemin	Oui.	Il s'agit de l'UUID du SVM contenant le fichier.
$FILE_PATH	Chemin	Oui.	Il s'agit du chemin d'accès au fichier ou au dossier.

Exemple de boucle

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

Exemple de sortie JSON

```
{
  "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"
}
```

Appliquer de nouvelles autorisations à un fichier

Vous pouvez appliquer un nouveau descripteur de sécurité à un fichier ou dossier spécifique.

Étape 1 : appliquez les nouvelles autorisations

Méthode HTTP et noeud final

Cet appel d'API REST utilise la méthode et le point de terminaison suivants.

Méthode HTTP	Chemin
POST	/api/protocoles/sécurité-fichier/permissions/{svm.uuid}/{path}

Type de traitement

Asynchrone

Paramètres d'entrée supplémentaires pour les exemples de boucles

Outre les paramètres communs à tous les appels API REST, les paramètres suivants sont également utilisés dans l'exemple curl de cette étape.

Paramètre	Type	Obligatoire	Description
$SVM_ID	Chemin	Oui.	Il s'agit de l'UUID du SVM contenant le fichier.
$FILE_PATH	Chemin	Oui.	Il s'agit du chemin d'accès au fichier ou au dossier.

Exemple de boucle

```
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\"}'
```

Exemple de sortie JSON

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

Étape 2 : récupération de l'état du travail

Exécutez le flux de travail "Obtenir l'instance de travail" et confirmez le `state` la valeur est `success`.

Mettez à jour les informations du descripteur de sécurité

Vous pouvez mettre à jour un descripteur de sécurité spécifique dans un fichier ou un dossier spécifique, y compris les indicateurs de propriétaire, de groupe ou de contrôle principal.

Étape 1 : mettez à jour le descripteur de sécurité

Méthode HTTP et noeud final

Cet appel d'API REST utilise la méthode et le point de terminaison suivants.

Méthode HTTP	Chemin
CORRECTIF	/api/protocoles/sécurité-fichier/permissions/{svm.uuid}/{path}

Type de traitement

Asynchrone

Paramètres d'entrée supplémentaires pour les exemples de boucles

Outre les paramètres communs à tous les appels API REST, les paramètres suivants sont également utilisés dans l'exemple curl de cette étape.

Paramètre	Type	Obligatoire	Description
$SVM_ID	Chemin	Oui.	Il s'agit de l'UUID du SVM contenant le fichier.
$FILE_PATH	Chemin	Oui.	Il s'agit du chemin d'accès au fichier ou au dossier.

Exemple de boucle

```
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\"}'
```

Exemple de sortie JSON

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

Étape 2 : récupération de l'état du travail

Exécutez le flux de travail "Obtenir l'instance de travail" et confirmez le `state` la valeur est `success`.

Supprimer une entrée de contrôle d'accès

Vous pouvez supprimer une entrée de contrôle d'accès (ACE) existante d'un fichier ou d'un dossier spécifique. La modification se propage à tous les objets enfants.

Étape 1 : supprimez l'ACE

Méthode HTTP et noeud final

Cet appel d'API REST utilise la méthode et le point de terminaison suivants.

Méthode HTTP	Chemin
SUPPRIMER	/api/protocoles/sécurité-fichier/permissions/{svm.uuid}/{path}

Type de traitement

Asynchrone

Paramètres d'entrée supplémentaires pour les exemples de boucles

Outre les paramètres communs à tous les appels API REST, les paramètres suivants sont également utilisés dans l'exemple curl de cette étape.

Paramètre	Type	Obligatoire	Description
$SVM_ID	Chemin	Oui.	Il s'agit de l'UUID du SVM contenant le fichier.
$FILE_PATH	Chemin	Oui.	Il s'agit du chemin d'accès au fichier ou au dossier.

Exemple de boucle

```
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\"}'
```

Exemple de sortie JSON

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

Étape 2 : récupération de l'état du travail

Exécutez le flux de travail "Obtenir l'instance de travail" et confirmez le `state` la valeur est `success`.

Mise en réseau

Répertorie les interfaces IP

Vous pouvez récupérer les LIFs IP attribuées au cluster et aux SVM. Vous pouvez le faire pour confirmer votre configuration réseau ou lorsque vous prévoyez d'ajouter une autre LIF.

Méthode HTTP et noeud final

Cet appel d'API REST utilise la méthode et le point de terminaison suivants.

Méthode HTTP	Chemin
OBTENEZ	/api/network/ip/interfaces

Type de traitement

Synchrone

Paramètres d'entrée supplémentaires pour les exemples Curl

Outre les paramètres communs à tous les appels API REST, les paramètres suivants sont également utilisés dans les exemples de boucles pour cette étape.

Paramètre	Type	Obligatoire	Description
champs	Requête	Non	Renvoie une liste limitée des valeurs de configuration pertinentes.

Exemple curl : renvoie toutes les LIFs avec les valeurs de configuration par défaut

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

Exemple curl : renvoie toutes les LIFs avec quatre valeurs de configuration spécifiques

```
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"
```

Exemple de sortie JSON


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

Sécurité

Comptes

Répertoriez les comptes

Vous pouvez récupérer une liste des comptes. Vous pouvez procéder ainsi pour évaluer votre environnement de sécurité ou avant de créer un nouveau compte.

Méthode HTTP et noeud final

Cet appel d'API REST utilise la méthode et le point de terminaison suivants.

Méthode HTTP	Chemin
OBTENEZ	/api/sécurité/comptes

Type de traitement

Synchrone

Exemple de boucle

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

Exemple de sortie JSON


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

Certificats et clés

Répertorie les certificats installés

Vous pouvez afficher la liste des certificats installés dans votre cluster ONTAP. Vous pouvez le faire pour voir si un certificat particulier est disponible ou pour obtenir l'ID d'un certificat spécifique.

Méthode HTTP et noeud final

Cet appel d'API REST utilise la méthode et le point de terminaison suivants.

Méthode HTTP	Chemin
OBTENEZ	/api/sécurité/certificats

Paramètres d'entrée supplémentaires pour les exemples de boucles

Outre les paramètres communs à tous les appels API REST, les paramètres suivants sont également utilisés dans l'exemple curl de cette étape.

Paramètre	Type	Obligatoire	Description
max_records	Requête	Non	Spécifiez le nombre d'enregistrements que vous souhaitez renvoyer.

Exemple curl : renvoie trois certificats

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

Exemple de sortie JSON


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

Installez un certificat

Vous pouvez installer un certificat X.509 signé dans votre cluster ONTAP. Vous pouvez le faire dans le cadre de la configuration d'une fonctionnalité ou d'un protocole ONTAP nécessitant une authentification forte.

Avant de commencer

Vous devez disposer du certificat que vous souhaitez installer. Vous devez également vous assurer que tous les certificats intermédiaires sont installés selon les besoins.

> Avant d'utiliser les exemples d'entrée JSON inclus ci-dessous, assurez-vous de mettre à jour le `public_certificate` valeur avec le certificat pour votre environnement.

Étape 1 : installez le certificat

Vous pouvez émettre un appel d'API pour installer le certificat.

Méthode HTTP et noeud final

Cet appel d'API REST utilise la méthode et le point de terminaison suivants.

Méthode HTTP	Chemin
POST	/api/sécurité/certificats

Exemple curl : installez un certificat CA racine au niveau du cluster

```
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
```

Exemple d'entrée JSON


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


Étape 2 : confirmez que le certificat a été installé

Exécutez le flux de travail "Répertorie les certificats installés" et vérifiez que le certificat est disponible.

RBAC

Préparez-vous à utiliser le RBAC

Selon votre environnement, vous pouvez utiliser la fonctionnalité RBAC de ONTAP de plusieurs manières. Cette section présente quelques scénarios courants sous forme de flux de travail. Dans chaque cas, l'accent est mis sur un objectif spécifique de sécurité et d'administration.

Avant de créer des rôles et d'attribuer un rôle à un compte utilisateur ONTAP, vous devez vous préparer en examinant les principales exigences et options de sécurité présentées ci-dessous. Assurez-vous également de passer en revue les concepts généraux du workflow à l'adresse "Préparez l'utilisation des workflows".

Quelle version de ONTAP utilisez-vous ?

La version d'ONTAP détermine les terminaux REST et les fonctionnalités RBAC disponibles.

Identifier les ressources protégées et la portée

Vous devez identifier les ressources ou les commandes à protéger et le périmètre (cluster ou SVM).

Quel accès l'utilisateur doit-il disposer ?

Après avoir identifié les ressources et la portée, vous devez déterminer le niveau d'accès à accorder.

Comment les utilisateurs pourront-ils accéder à ONTAP ?

Celui-ci peut accéder au ONTAP via l'API REST, l'interface de ligne de commandes ou les deux.

L'un des rôles intégrés est-il suffisant ou le rôle personnalisé requis ?

Il est plus pratique d'utiliser un rôle intégré existant, mais vous pouvez en créer un nouveau si nécessaire.

Quel type de rôle est nécessaire ?

En fonction des exigences de sécurité et de l'accès ONTAP, vous devez choisir de créer un rôle REST ou traditionnel.

Créer des rôles

Limiter l'accès aux opérations de volume du SVM

Vous pouvez définir un rôle de restriction de l'administration des volumes de stockage au sein d'une SVM.

A propos de ce flux de travail

Un rôle traditionnel est d'abord créé pour autoriser initialement l'accès à toutes les principales fonctions d'administration des volumes, à l'exception du clonage. Le rôle est défini avec les caractéristiques suivantes :

- Possibilité d'effectuer toutes les opérations de volume CRUD, y compris obtenir, créer, modifier et supprimer
- Impossible de créer un clone de volume

Vous pouvez ensuite, si nécessaire, mettre à jour le rôle. Dans ce workflow, le rôle est modifié dans la deuxième étape pour permettre à l'utilisateur de créer un clone de volume.

Étape 1 : créer le rôle

Vous pouvez émettre un appel d'API pour créer le rôle RBAC.

Méthode HTTP et noeud final

Cet appel d'API REST utilise la méthode et le point de terminaison suivants.

Méthode HTTP	Chemin
POST	/api/sécurité/rôles

Exemple de boucle

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

Exemple d'entrée JSON

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

Étape 2 : mettez à jour le rôle

Vous pouvez émettre un appel API pour mettre à jour le rôle existant.

Méthode HTTP et noeud final

Cet appel d'API REST utilise la méthode et le point de terminaison suivants.

Méthode HTTP	Chemin
POST	/api/sécurité/rôles

Paramètres d'entrée supplémentaires pour les exemples de boucles

Outre les paramètres communs à tous les appels API REST, les paramètres suivants sont également utilisés dans l'exemple curl de cette étape.

Paramètre	Type	Obligatoire	Description
$SVM_ID	Chemin	Oui.	Il s'agit de l'UUID du SVM qui contient la définition du rôle.
$NOM_RÔLE	Chemin	Oui.	Voici le nom du rôle au sein de la SVM à mettre à jour.

Exemple de boucle

```
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
```

Exemple d'entrée JSON

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

Administration de la protection des données

Vous pouvez offrir à un utilisateur des fonctionnalités de protection des données limitées.

A propos de ce flux de travail

Le rôle traditionnel créé est défini avec les caractéristiques suivantes :

- Création et suppression de snapshots et mise à jour des relations SnapMirror
- Ne peut créer ou modifier des objets de niveau supérieur tels que des volumes ou des SVM

Méthode HTTP et noeud final

Cet appel d'API REST utilise la méthode et le point de terminaison suivants.

Méthode HTTP	Chemin
POST	/api/sécurité/rôles

Exemple de boucle

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

Exemple d'entrée JSON

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

Autoriser la génération de rapports ONTAP

Vous pouvez créer un rôle REST pour permettre aux utilisateurs de générer des rapports ONTAP.

A propos de ce flux de travail

Le rôle créé est défini avec les caractéristiques suivantes :

- Possibilité de récupérer toutes les informations relatives à la capacité et aux performances de l'objet de stockage (par exemple, volume, qtree, LUN, agrégats, nœud, Et relations SnapMirror)
- Ne peut créer ni modifier des objets de niveau supérieur (tels que des volumes ou des SVM)

Méthode HTTP et noeud final

Cet appel d'API REST utilise la méthode et le point de terminaison suivants.

Méthode HTTP	Chemin
POST	/api/sécurité/rôles

Exemple de boucle

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

Exemple d'entrée JSON

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

Créer un utilisateur avec un rôle

Vous pouvez utiliser ce flux de travail pour créer un utilisateur avec un rôle REST associé.

A propos de ce flux de travail

Ce flux de travail comprend les étapes types nécessaires pour créer un rôle REST personnalisé et l'associer à un nouveau compte utilisateur. L'utilisateur et le rôle ont une étendue SVM et sont associés à un SVM de données spécifique. Certaines étapes peuvent être facultatives ou doivent être modifiées en fonction de votre environnement.

Étape 1 : liste des SVM de données dans le cluster

Effectuer l'appel d'API REST suivant pour lister les SVM dans le cluster. L'UUID et le nom de chaque SVM sont fournis dans le résultat.

Méthode HTTP et noeud final

Cet appel d'API REST utilise la méthode et le point de terminaison suivants.

Méthode HTTP	Chemin
OBTENEZ	/api/svm/svm

Exemple de boucle

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

Une fois que vous avez terminé

Sélectionner le SVM souhaité dans la liste dans laquelle vous allez créer le nouvel utilisateur et le nouveau rôle.

Étape 2 : liste des utilisateurs définis pour la SVM

Effectuer l'appel de l'API REST suivant pour répertorier les utilisateurs définis dans la SVM que vous avez sélectionnée. Vous pouvez identifier le SVM via le paramètre propriétaire.

Méthode HTTP et noeud final

Cet appel d'API REST utilise la méthode et le point de terminaison suivants.

Méthode HTTP	Chemin
OBTENEZ	/api/sécurité/comptes

Exemple de boucle

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

Une fois que vous avez terminé

En fonction des utilisateurs déjà définis au sein du SVM, choisissez un nom unique pour le nouvel utilisateur.

Étape 3 : liste des rôles REST définis pour la SVM

Effectuer l'appel de l'API REST suivant pour répertorier les rôles définis dans la SVM que vous avez sélectionnée. Vous pouvez identifier le SVM via le paramètre propriétaire.

Méthode HTTP et noeud final

Cet appel d'API REST utilise la méthode et le point de terminaison suivants.

Méthode HTTP	Chemin
OBTENEZ	/api/sécurité/rôles

Exemple de boucle

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

Une fois que vous avez terminé

En fonction des rôles déjà définis dans le SVM, choisissez un nom unique pour le nouveau rôle.

Étape 4 : créez un rôle REST personnalisé

Effectuer l'appel d'API REST suivant pour créer un rôle REST personnalisé dans la SVM. Au départ, le rôle n'a qu'un privilège qui établit un accès par défaut de **none** de sorte que tout accès soit refusé.

Méthode HTTP et noeud final

Cet appel d'API REST utilise la méthode et le point de terminaison suivants.

Méthode HTTP	Chemin
POST	/api/sécurité/rôles

Exemple de boucle

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

Exemple d'entrée JSON

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

Une fois que vous avez terminé

Vous pouvez également effectuer de nouveau l'étape 3 pour afficher le nouveau rôle. Vous pouvez également afficher les rôles au niveau de l'interface de ligne de commandes ONTAP.

Étape 5 : mettez à jour le rôle en ajoutant des privilèges supplémentaires

Effectuez l'appel d'API REST suivant pour modifier le rôle en ajoutant des privilèges si nécessaire.

Méthode HTTP et noeud final

Cet appel d'API REST utilise la méthode et le point de terminaison suivants.

Méthode HTTP	Chemin
POST	/api/sécurité/rôles/{owner.uuid}/{name}/privilèges

Paramètres d'entrée supplémentaires pour les exemples de boucles

Outre les paramètres communs à tous les appels API REST, les paramètres suivants sont également utilisés dans l'exemple curl de cette étape.

Paramètre	Type	Obligatoire	Description
$SVM_ID	Chemin	Oui.	UUID du SVM qui contient la définition de rôle.
$NOM_RÔLE	Chemin	Oui.	Nom du rôle au sein de la SVM à mettre à jour.

Exemple de boucle

```
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
```

Exemple d'entrée JSON

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

Une fois que vous avez terminé

Vous pouvez également effectuer de nouveau l'étape 3 pour afficher le nouveau rôle. Vous pouvez également afficher les rôles au niveau de l'interface de ligne de commandes ONTAP.

Étape 6 : créer un utilisateur

Effectuez l'appel d'API REST suivant pour créer un compte utilisateur. Le rôle **dprole1** créé ci-dessus est associé au nouvel utilisateur.

Vous pouvez créer l'utilisateur sans rôle. Dans ce cas, un rôle par défaut est attribué à l'utilisateur (soit `admin` ou `vsadmin`) Selon que l'utilisateur est défini ou non avec le périmètre du cluster ou du SVM. Vous devrez modifier l'utilisateur pour attribuer un rôle différent.

Méthode HTTP et noeud final

Cet appel d'API REST utilise la méthode et le point de terminaison suivants.

Méthode HTTP	Chemin
POST	/api/sécurité/comptes

Exemple de boucle

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

Exemple d'entrée JSON

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

Une fois que vous avez terminé

Vous pouvez vous connecter à l'interface de gestion du SVM en utilisant les identifiants du nouvel utilisateur.

Stockage

Lister les agrégats

Vous pouvez récupérer la liste des agrégats dans le cluster. Vous pourriez le faire pour évaluer l'utilisation et les performances.

Méthode HTTP et noeud final

Cet appel d'API REST utilise la méthode et le point de terminaison suivants.

Méthode HTTP	Chemin
OBTENEZ	/api/stockage/disques

Type de traitement

Synchrone

Paramètres d'entrée supplémentaires pour les exemples Curl

Outre les paramètres communs à tous les appels API REST, les paramètres suivants sont également utilisés dans les exemples de boucles pour cette étape.

Paramètre	Type	Obligatoire	Description
node.name	Requête	Non	Peut être utilisé pour identifier le nœud auquel chaque agrégat est rattaché.

Exemple de bouclage : renvoie tous les agrégats avec les valeurs de configuration par défaut

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

Exemple de bouclage : renvoie tous les agrégats avec une valeur de configuration spécifique

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

Exemple de sortie JSON

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

Répertoriez les disques

Vous pouvez récupérer la liste des disques du cluster. Vous pouvez ainsi localiser une ou plusieurs réserves à utiliser dans le cadre de la création d'un agrégat.

Méthode HTTP et noeud final

Cet appel d'API REST utilise la méthode et le point de terminaison suivants.

Méthode HTTP	Chemin
OBTENEZ	/api/stockage/disques

Type de traitement

Synchrone

Paramètres d'entrée supplémentaires pour les exemples Curl

Outre les paramètres communs à tous les appels API REST, les paramètres suivants sont également utilisés dans les exemples de boucles pour cette étape.

Paramètre	Type	Obligatoire	Description
état	Requête	Non	Peut être utilisé pour identifier les disques de spare disponibles pour les nouveaux agrégats.

Exemple curl : renvoie tous les disques

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

Exemple curl : renvoyez les disques de rechange

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

Exemple de sortie JSON


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


Assistance

EMS

Préparez-vous à gérer les services de support EMS

Vous pouvez configurer le traitement EMS (Event Management System) pour un cluster ONTAP et récupérer les messages EMS si nécessaire.

Présentation

Plusieurs exemples de flux de travail sont disponibles pour illustrer l'utilisation des services EMS de ONTAP. Avant d'utiliser les flux de travail et d'émettre l'un des appels de l'API REST, assurez-vous de passer en revue "Préparez l'utilisation des workflows".

Si vous utilisez Python, voyez aussi le scripy "events.py" Pour des exemples de la façon d'automatiser certaines des activités liées au SGE.

Comparaison des commandes de l'API REST ONTAP et de l'interface CLI ONTAP

Pour de nombreuses tâches, l'utilisation de l'API REST ONTAP requiert moins d'appels que les commandes CLI ONTAP équivalentes. Le tableau ci-dessous présente une liste d'appels API et l'équivalent des commandes CLI nécessaires à chaque tâche.

L'API REST DE ONTAP	INTERFACE DE LIGNE DE COMMANDES DE ONTAP
OBTENIR /support/ems	`event config show`
POST /support/ems/destinations	1. `event notification destination create` 2. `event notification create`
`GET /support/ems/events`	`event log show`
`POST /support/ems/filters`	1. `event filter create -filter-name <filtername>` 2. `event filter rule add -filter-name <filtername>`

Informations associées

- "Script Python illustrant EMS"
- "API REST ONTAP : automatisation des notifications d'événements de forte gravité"

Répertorie les événements du journal EMS

Vous pouvez récupérer tous les messages de notification d'événements ou uniquement ceux ayant des caractéristiques spécifiques.

Méthode HTTP et noeud final

Cet appel d'API REST utilise la méthode et le point de terminaison suivants.

Méthode HTTP	Chemin
OBTENEZ	/api/support/ems/events

Type de traitement

Synchrone

Paramètres d'entrée supplémentaires pour les exemples Curl

Outre les paramètres communs à tous les appels API REST, les paramètres suivants sont également utilisés dans les exemples de boucles pour cette étape.

Paramètre	Type	Obligatoire	Description
champs	Requête	Non	Permet de demander l'inclusion de champs spécifiques dans la réponse.
max_records	Requête	Non	Peut être utilisé pour limiter le nombre d'enregistrements renvoyés dans une seule demande.
message_journal	Requête	Non	Utilisé pour rechercher une valeur de texte spécifique et renvoyer uniquement les messages correspondants.
message.severity	Requête	Non	Limitez les messages renvoyés à ceux dont le niveau de gravité est spécifique, par exemple `alert`.

Exemple de boucle : renvoie le dernier message et la valeur du nom

```
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"
```

Exemple de boucle : renvoie un message contenant un texte et une gravité spécifiques

```
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"
```

Exemple de sortie JSON


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

Obtenir la configuration EMS

Vous pouvez récupérer la configuration EMS actuelle pour un cluster ONTAP. Vous pouvez le faire avant de mettre à jour la configuration ou de créer une nouvelle notification EMS.

Méthode HTTP et noeud final

Cet appel d'API REST utilise la méthode et le point de terminaison suivants.

Méthode HTTP	Chemin
OBTENEZ	/api/support/ems

Type de traitement

Synchrone

Exemple de boucle

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

Exemple de sortie JSON

```
{
  "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"
}
```

Créez une notification EMS

Vous pouvez utiliser le flux de travail suivant pour créer une nouvelle destination de notification EMS afin de recevoir les messages d'événement sélectionnés.

Étape 1 : configurer les paramètres de messagerie de l'ensemble du système

Vous pouvez émettre l'appel d'API suivant pour configurer les paramètres de messagerie du système.

Méthode HTTP et noeud final

Cet appel d'API REST utilise la méthode et le point de terminaison suivants.

Méthode HTTP	Chemin
CORRECTIF	/api/support/ems

Type de traitement

Synchrone

Paramètres d'entrée supplémentaires pour les exemples Curl

Outre les paramètres communs à tous les appels API REST, les paramètres suivants sont également utilisés dans les exemples de boucles pour cette étape.

Paramètre	Type	Obligatoire	Description
mail_from	Requête	Oui.	Définit le `from` dans les e-mails de notification.
serveur_de_messagerie	Requête	Oui.	Configure le serveur de messagerie SMTP cible.

Exemple de boucle

```
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"
```

Étape 2 : définir un filtre de message

Vous pouvez émettre un appel d'API pour définir une règle de filtre correspondant aux messages.

Méthode HTTP et noeud final

Cet appel d'API REST utilise la méthode et le point de terminaison suivants.

Méthode HTTP	Chemin
POST	/api/support/ems/filtres

Type de traitement

Synchrone

Paramètres d'entrée supplémentaires pour les exemples Curl

Outre les paramètres communs à tous les appels API REST, les paramètres suivants sont également utilisés dans les exemples de boucles pour cette étape.

Paramètre	Type	Obligatoire	Description
Filtre	Corps	Oui.	Inclut les valeurs de la configuration du filtre.

Exemple de boucle

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

Exemple d'entrée JSON

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

Étape 3 : création d'une destination de message

Vous pouvez émettre un appel API pour créer une destination de message.

Méthode HTTP et noeud final

Cet appel d'API REST utilise la méthode et le point de terminaison suivants.

Méthode HTTP	Chemin
POST	/api/support/ems/destinations

Type de traitement

Synchrone

Paramètres d'entrée supplémentaires pour les exemples Curl

Outre les paramètres communs à tous les appels API REST, les paramètres suivants sont également utilisés dans les exemples de boucles pour cette étape.

Paramètre	Type	Obligatoire	Description
Configuration de la destination	Corps	Oui.	Inclut les valeurs de la destination de l'événement.

Exemple de boucle

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

Exemple d'entrée JSON

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

SVM

Lister les SVM

Vous pouvez afficher la liste des SVM (Storage Virtual machines) définis au sein d'un cluster ONTAP. Pour cela, vous pouvez rechercher l'identifiant d'un SVM spécifique ou assurer son unicité avant de créer un SVM.

Méthode HTTP et noeud final

Cet appel d'API REST utilise la méthode et le point de terminaison suivants.

Méthode HTTP	Chemin
OBTENEZ	/api/svm/svm

Exemple de boucle

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

Exemple de sortie JSON


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

Informations sur le copyright

Informations sur les marques commerciales