# Concepts

## ONTAP Select

NetApp
February 09, 2024

This PDF was generated from https://docs.netapp.com/fr-fr/ontap-select-9111/concept_api_rest.html on February 09, 2024. Always check docs.netapp.com for the latest.

# Sommaire

# Concepts

## Base de services Web REST

Representational State Transfer (REST) est un style qui permet de créer des applications Web distribuées. Lorsqu'il est appliqué à la conception d'une API de services Web, il établit un ensemble de technologies et de meilleures pratiques pour l'exposition des ressources basées sur serveur et la gestion de leurs États. Il utilise des protocoles et des normes courants pour offrir une base flexible pour le déploiement et la gestion des clusters ONTAP Select.

### Contraintes classiques et architectures

REST a été formellement articulé par Roy Fielding dans son doctorat "thèse" À UC Irvine en 2000. Il définit un style architectural à travers un ensemble de contraintes, qui ont collectivement amélioré les applications basées sur le Web et les protocoles sous-jacents. Ces contraintes créent une application de services web RESTful basée sur une architecture client/serveur utilisant un protocole de communication sans état.

### Ressources et représentation d'état

Les ressources sont les composants de base d'un système basé sur le Web. Lors de la création d'une application de services Web REST, les premières tâches de conception incluent :

- Identification des ressources système ou serveur
  Chaque système utilise et gère les ressources. Une ressource peut être un fichier, une transaction commerciale, un processus ou une entité administrative. L'une des premières tâches de conception d'une application basée sur des services Web REST consiste à identifier les ressources.
- Définition des États de ressource et des opérations d'état associées
  Les ressources se trouvent toujours dans un des États finis. Les États, ainsi que les opérations associées utilisées pour affecter les changements d'état, doivent être clairement définis.

Les messages sont échangés entre le client et le serveur pour accéder aux ressources et les modifier selon le modèle CRUD générique (Créer, lire, mettre à jour et Supprimer).

### Terminaux URI

Chaque ressource REST doit être définie et mise à disposition à l'aide d'un schéma d'adressage bien défini. Les noeuds finaux où les ressources sont situées et identifiées utilisent un URI (Uniform Resource identifier). L'URI fournit un cadre général pour créer un nom unique pour chaque ressource du réseau. L'URL (Uniform Resource Locator) est un type d'URI utilisé avec les services Web pour identifier et accéder aux ressources. Les ressources sont généralement exposées dans une structure hiérarchique similaire à un répertoire de fichiers.

### Messages HTTP

Le protocole HTTP (Hypertext Transfer Protocol) est le protocole utilisé par le client et le serveur de services Web pour échanger des messages de requête et de réponse sur les ressources. Dans le cadre de la conception d'une application de services Web, les verbes HTTP (COMME GET et POST) sont mappées aux ressources et aux actions de gestion d'état correspondantes.

Le HTTP est sans état. Par conséquent, pour associer un ensemble de requêtes et de réponses associées sous une même transaction, des informations supplémentaires doivent être incluses dans les en-têtes HTTP des flux de données requête/réponse.

### Formatage JSON

Bien que les informations puissent être structurées et transférées de plusieurs façons entre un client et un serveur, l'option la plus populaire (et celle utilisée avec l'API REST de déploiement) est JavaScript Object notation (JSON). JSON est une norme de l'industrie qui représente les structures de données simples en texte brut et permet de transférer les informations d'état décrivant les ressources.

# Comment accéder à l'API de déploiement

En raison de la flexibilité inhérente des services web REST, l'API de déploiement ONTAP Select est accessible de plusieurs façons.

### Déployez l'interface utilisateur native de l'utilitaire

La principale façon dont vous accédez à l'API consiste à utiliser l'interface utilisateur Web de ONTAP Select Deploy. Le navigateur fait des appels à l'API et reformate les données en fonction de la conception de l'interface utilisateur. Vous pouvez également accéder à l'API via l'interface de ligne de commande de l'utilitaire de déploiement.

### Page de documentation en ligne sur le déploiement de ONTAP Select

La page de documentation en ligne de ONTAP Select Deploy fournit un point d'accès alternatif lorsque vous utilisez un navigateur. En plus de fournir un moyen d'exécuter directement des appels API individuels, la page comprend également une description détaillée de l'API, y compris les paramètres d'entrée et d'autres options pour chaque appel. Les appels API sont organisés en plusieurs zones fonctionnelles ou catégories différentes.

### Programme personnalisé

Vous pouvez accéder à l'API de déploiement à l'aide de plusieurs langages et outils de programmation différents. Les options les plus populaires sont Python, Java et Curl. Un programme, un script ou un outil qui utilise l'API agit comme un client de services Web REST. L'utilisation d'un langage de programmation permet de mieux comprendre l'API et offre une possibilité d'automatiser les déploiements ONTAP Select.

# Déploiement des versions d'API

Un numéro de version est attribué à l'API REST incluse dans ONTAP Select Deploy. Le numéro de version de l'API est indépendant du numéro de version de déploiement. Il est important de connaître la version d'API incluse dans votre version de déploiement et d'identifier en quoi cela pourrait affecter votre utilisation de l'API.

La version actuelle de l'utilitaire d'administration Deploy inclut la version 3 de l'API REST. Les anciennes versions de l'utilitaire Deploy comprennent les versions d'API suivantes :

**Déploiement 2.8 et versions ultérieures**

ONTAP Select Deploy 2.8, ainsi que que que toutes les versions ultérieures incluent la version 3 de l'API REST.

**Déployer les versions 2.7.2 et antérieures**

ONTAP Select Deploy 2.7.2. Toutes les versions antérieures comprennent la version 2 de l'API REST.

> Les versions 2 et 3 de l'API REST ne sont pas compatibles. Si vous effectuez une mise à niveau vers le déploiement de la version 2.8 ou ultérieure à partir d'une version antérieure incluant la version 2 de l'API, vous devez mettre à jour tout code existant qui accède directement à l'API ainsi que tous les scripts à l'aide de l'interface de ligne de commande.

# Caractéristiques opérationnelles de base

Alors QUE REST établit un ensemble commun de technologies et de meilleures pratiques, les détails de chaque API peuvent varier en fonction des choix de conception. Vous devez connaître les détails et les caractéristiques opérationnelles de l'API de déploiement ONTAP Select avant d'utiliser l'API.

## Hôte de l'hyperviseur ou nœud ONTAP Select

Un *hyperviseur host* est la plate-forme matérielle principale qui héberge une machine virtuelle ONTAP Select. Lorsqu'une machine virtuelle ONTAP Select est déployée et active sur un hôte hyperviseur, la machine virtuelle est considérée comme un *ONTAP Select node*. Avec la version 3 de l'API REST déployée, les objets hôte et nœud sont distincts. Cela permet une relation un-à-plusieurs, dans laquelle un ou plusieurs nœuds ONTAP Select peuvent s'exécuter sur le même hôte hyperviseur.

## Identifiants d'objets

Un identifiant unique est attribué à chaque instance de ressource ou objet lors de sa création. Ces identifiants sont globalement uniques dans une instance spécifique du déploiement ONTAP Select. Après l'émission d'un appel API qui crée une nouvelle instance d'objet, la valeur d'ID associée est renvoyée à l'appelant dans le `location` En-tête de la réponse HTTP. Vous pouvez extraire l'identificateur et l'utiliser sur les appels suivants lorsque vous faites référence à l'instance de ressource.

> Le contenu et la structure interne des identificateurs d'objet peuvent changer à tout moment. Vous ne devez utiliser les identificateurs sur les appels API applicables que si nécessaire lorsque vous faites référence aux objets associés.

## Demander des identifiants

Un identifiant unique est attribué à chaque requête d'API réussie. L'identifiant est renvoyé dans le `request-id` En-tête de la réponse HTTP associée. Vous pouvez utiliser un identificateur de demande pour faire référence collectivement aux activités d'une seule transaction de réponse de requête API spécifique. Par exemple, vous pouvez récupérer tous les messages d'événement pour une transaction basée sur l'ID de demande

## Appels synchrones et asynchrones

Il existe deux méthodes principales pour qu'un serveur exécute une requête HTTP reçue d'un client :

- Synchrone
  Le serveur exécute la demande immédiatement et répond avec un code d'état 200, 201 ou 204.
- Asynchrone

Le serveur accepte la demande et répond avec le code d'état 202. Cela indique que le serveur a accepté la demande du client et a lancé une tâche en arrière-plan pour terminer la demande. Le succès ou l'échec final n'est pas immédiatement disponible et doit être déterminé par d'autres appels API.

### Confirmez la fin d'une tâche en cours d'exécution

Généralement, toute opération qui peut prendre un certain temps est traitée de manière asynchrone à l'aide d'un
tâche en arrière-plan sur le serveur. Avec l'API REST de Deploy, chaque tâche en arrière-plan est ancrée dans un
Objet de travail qui suit la tâche et fournit des informations, telles que l'état actuel. Un objet travail,
Avec son identifiant unique, est renvoyé dans la réponse HTTP après la création d'une tâche en arrière-plan.

Vous pouvez interroger l'objet Job directement pour déterminer le succès ou l'échec de l'appel API associé. Pour plus d'informations, reportez-vous à la section *traitement asynchrone à l'aide de l'objet travail*.

Outre l'objet travail, vous pouvez déterminer la réussite ou l'échec d'un de plusieurs manières
demande, comprenant :

- Messages d'événement
  Vous pouvez récupérer tous les messages d'événement associés à un appel d'API spécifique en utilisant l'ID de demande renvoyé avec la réponse d'origine. Les messages d'événement contiennent généralement une indication de réussite ou d'échec et peuvent également être utiles lors du débogage d'une condition d'erreur.
- État ou état de la ressource
  Plusieurs des ressources conservent une valeur d'état ou d'état que vous pouvez interroger pour déterminer indirectement le succès ou l'échec d'une demande.

### Sécurité

L'API de déploiement utilise les technologies de sécurité suivantes :

- Sécurité de la couche de transport
  Tout le trafic envoyé sur le réseau entre le serveur de déploiement et le client est chiffré via TLS. L'utilisation du protocole HTTP sur un canal non crypté n'est pas prise en charge. TLS version 1.2 est pris en charge.
- Authentification HTTP
  L'authentification de base est utilisée pour chaque transaction d'API. Un en-tête HTTP, qui inclut le nom d'utilisateur et le mot de passe dans une chaîne base64, est ajouté à chaque requête.

# Transaction d'API de demande et de réponse

Chaque appel API de déploiement est exécuté sous forme de requête HTTP vers la machine virtuelle de déploiement, qui génère une réponse associée au client. Cette paire de requête/réponse est considérée comme une transaction API. Avant d'utiliser l'API de déploiement, vous devez connaître les variables d'entrée disponibles pour contrôler une requête et le contenu de la sortie de réponse.

## Variables d'entrée contrôlant une requête API

Vous pouvez contrôler le traitement d'un appel API à l'aide de paramètres définis dans la requête HTTP.

### En-têtes de demande

Vous devez inclure plusieurs en-têtes dans la requête HTTP, notamment :

- type-contenu
  Si le corps de la demande inclut JSON, cet en-tête doit être défini sur application/json.
- accepter
  Si le corps de réponse inclut JSON, cet en-tête doit être défini sur application/json.
- autorisation
  L'authentification de base doit être définie avec le nom d'utilisateur et le mot de passe codés dans une chaîne base64.

### Corps de la demande

Le contenu du corps de la demande varie en fonction de l'appel spécifique. Le corps de requête HTTP comprend l'un des éléments suivants :

- Objet JSON avec variables d'entrée (par exemple, le nom d'un nouveau cluster)
- Vide

### Filtrer les objets

Lors de l'émission d'un appel API utilisant GET, vous pouvez limiter ou filtrer les objets renvoyés en fonction de n'importe quel attribut. Par exemple, vous pouvez spécifier une valeur exacte à associer :

```
<field>=<query value>
```

En plus d'une correspondance exacte, d'autres opérateurs sont disponibles pour renvoyer un ensemble d'objets sur une plage de valeurs. ONTAP Select prend en charge les opérateurs de filtrage indiqués ci-dessous.

| Opérateur | Description |
|---|---|
| = | Égal à |
| < | Inférieur à |
| > | Supérieur à |
| &lt ;= | Inférieur ou égal à |
| >= | Supérieur ou égal à |
| | Ou |
| ! | Différent de |
| * | Un caractère générique gourmand |

Vous pouvez également renvoyer un ensemble d'objets en fonction de la définition ou non d'un champ spécifique à l'aide du mot clé null ou de sa négation (!null) dans le cadre de la requête.

### Sélection des champs d'objet

Par défaut, l'émission d'un appel API à l'aide DE GET renvoie uniquement les attributs qui identifient de manière unique l'objet ou les objets. Cet ensemble minimal de champs sert de clé pour chaque objet et varie en fonction du type d'objet. Vous pouvez sélectionner des propriétés d'objet supplémentaires à l'aide du paramètre de requête de champs de la manière suivante :

- Champs peu coûteux
  Spécifiez `fields=*` pour récupérer les champs d'objet qui sont conservés dans la mémoire du serveur local ou nécessitant peu de traitement pour accéder à.
- Champs coûteux
  Spécifiez `fields=**` pour récupérer tous les champs d'objet, y compris ceux nécessitant un traitement serveur supplémentaire pour accéder.
- Sélection de champ personnalisée
  Utiliser `fields=FIELDNAME` pour spécifier le champ exact souhaité. Lorsque vous demandez plusieurs champs, les valeurs doivent être séparées par des virgules sans espaces.

Vous devez toujours identifier les champs spécifiques que vous souhaitez. Vous ne devriez récupérer que l'ensemble de champs bon marché ou coûteux si nécessaire. Cette classification économique et onéreuse est déterminée par NetApp sur la base de l'analyse interne des performances. La classification d'un champ donné peut changer à tout moment.

### Trier les objets dans le jeu de sortie

Les enregistrements d'une collection de ressources sont renvoyés dans l'ordre par défaut défini par l'objet. Vous pouvez modifier l'ordre à l'aide du paramètre de requête order_by avec le nom de champ et la direction de tri comme suit :
`order_by=<field name> asc|desc`

Par exemple, vous pouvez trier le champ de type par ordre décroissant, suivi d'un ID par ordre croissant :
`order_by=type desc, id asc`

Lorsque vous ajoutez plusieurs paramètres, vous devez séparer les champs par une virgule.

### Pagination

Lors de l'émission d'un appel API à l'aide DE GET pour accéder à une collection d'objets du même type, tous les objets correspondants sont renvoyés par défaut. Si nécessaire, vous pouvez limiter le nombre d'enregistrements renvoyés à l'aide du paramètre de requête max_records avec la demande. Par exemple :
`max_records=20`

Si nécessaire, vous pouvez combiner ce paramètre avec d'autres paramètres de requête pour affiner le jeu de résultats. Par exemple, ce qui suit renvoie jusqu'à 10 événements système générés après le temps spécifié :
`time⇒ 2019-04-04T15:41:29.140265Z&max_records=10`

Vous pouvez émettre plusieurs requêtes à la page via les événements (ou tout type d'objet). Chaque appel d'API suivant doit utiliser une nouvelle valeur de temps basée sur le dernier événement du dernier jeu de résultats.

## Interpréter une réponse API

Chaque requête d'API génère une réponse au client. Vous pouvez examiner la réponse pour déterminer

s'il a réussi et qu'il récupère des données supplémentaires si nécessaire.

### Code d'état HTTP

Les codes d'état HTTP utilisés par l'API REST de déploiement sont décrits ci-dessous.

| Code | Signification | Description |
|---|---|---|
| 200 | OK | Indique que les appels qui ne créent pas d'objet ont réussi. |
| 201 | Créé | Un objet est créé avec succès ; l'en-tête de réponse d'emplacement inclut l'identifiant unique de l'objet. |
| 202 | Accepté | Une tâche en arrière-plan longue durée a été démarrée pour exécuter la demande, mais l'opération n'a pas encore été terminée. |
| 400 | Demande incorrecte | L'entrée de la demande n'est pas reconnue ou est inappropriée. |
| 403 | Interdit | L'accès est refusé en raison d'une erreur d'autorisation. |
| 404 | Introuvable | La ressource mentionnée dans la demande n'existe pas. |
| 405 | Méthode non autorisée | Le verbe HTTP de la demande n'est pas pris en charge pour la ressource. |
| 409 | Conflit | La tentative de création d'un objet a échoué car celui-ci existe déjà. |
| 500 | Erreur interne | Une erreur interne générale s'est produite sur le serveur. |
| 501 | Non mis en œuvre | L'URI est connu mais ne peut pas exécuter la demande. |

### En-têtes de réponse

Plusieurs en-têtes sont inclus dans la réponse HTTP générée par le serveur de déploiement, notamment :

- id-demande
  Un identifiant de demande unique est attribué à chaque demande d'API réussie.
- emplacement
  Lorsqu'un objet est créé, l'en-tête d'emplacement inclut l'URL complète vers le nouvel objet, y compris l'identifiant unique de l'objet.

### Corps de réponse

Le contenu de la réponse associée à une requête API diffère selon l'objet, le type de traitement et le succès ou l'échec de la requête. Le corps de réponse est rendu au format JSON.

- Objet unique
  Un objet peut être renvoyé avec un ensemble de champs en fonction de la requête. Par exemple, vous pouvez utiliser OBTENIR pour extraire les propriétés sélectionnées d'un cluster à l'aide de l'identifiant unique.
- Objets multiples
  Plusieurs objets d'une collection de ressources peuvent être renvoyés. Dans tous les cas, un format cohérent est utilisé avec `num_records` indique le nombre d'enregistrements et d'enregistrements contenant un tableau des instances d'objet. Par exemple, vous pouvez extraire tous les nœuds définis dans un cluster spécifique.
- Objet travail

Si un appel API est traité de manière asynchrone, un objet travail est renvoyé, qui ancres la tâche d'arrière-plan. Par exemple, la demande POST utilisée pour déployer un cluster est traitée de manière asynchrone et renvoie un objet Job.

- Objet erreur
  Si une erreur se produit, un objet erreur est toujours renvoyé. Par exemple, vous recevrez une erreur lors de la tentative de création d'un cluster dont le nom existe déjà.
- Vide
  Dans certains cas, aucune donnée n'est renvoyée et le corps de réponse est vide. Par exemple, le corps de réponse est vide après avoir utilisé SUPPRIMER pour supprimer un hôte existant.

# Traitement asynchrone à l'aide de l'objet de travail

Certains appels API de déploiement, en particulier ceux qui créent ou modifient une ressource, peuvent prendre plus de temps que d'autres appels. Le déploiement de ONTAP Select traite ces requêtes à long terme de manière asynchrone.

## Demandes asynchrones décrites à l'aide de l'objet travail

Après avoir effectué un appel API qui s'exécute de manière asynchrone, le code de réponse HTTP 202 indique que la demande a été validée et acceptée avec succès, mais pas encore terminée. La requête est traitée comme une tâche d'arrière-plan qui continue à s'exécuter après la réponse HTTP initiale au client. La réponse inclut l'objet Job qui fixe la requête, y compris son identifiant unique.

Reportez-vous à la page de documentation en ligne de ONTAP Select Deploy pour déterminer quels appels API fonctionnent de manière asynchrone.

## Interroger l'objet travail associé à une demande API

L'objet travail renvoyé dans la réponse HTTP contient plusieurs propriétés. Vous pouvez interroger la propriété d'état pour déterminer si la demande a bien été effectuée. Un objet travail peut être dans l'un des États suivants :

- En file d'attente
- Exécution
- Réussite
- Panne

Il existe deux techniques que vous pouvez utiliser lors de l'interrogation d'un objet travail pour détecter un état de terminal pour la tâche, succès ou échec :

- Demande d'interrogation standard
  L'état actuel du travail est immédiatement renvoyé
- Demande d'interrogation longue
  L'état du travail est renvoyé uniquement lorsque l'une des situations suivantes se produit :
  - L'état a changé plus récemment que la valeur date-heure fournie sur la demande d'interrogation
  - La valeur de temporisation a expiré (1 à 120 secondes)

Interrogation standard et interrogation longue utilisent le même appel API pour interroger un objet travail.

Cependant, une demande d'interrogation longue inclut deux paramètres de requête : `poll_timeout` et `last_modified`.

Vous devez toujours utiliser une interrogation longue pour réduire la charge de travail sur la machine virtuelle de déploiement.

## Procédure générale d'émission d'une demande asynchrone

Vous pouvez utiliser la procédure de haut niveau suivante pour effectuer un appel d'API asynchrone :

1. Lancez l'appel d'API asynchrone.
2. Recevoir une réponse HTTP 202 indiquant que la demande a été acceptée avec succès.
3. Extraire l'identifiant de l'objet travail du corps de réponse.
4. Dans une boucle, effectuez les opérations suivantes dans chaque cycle :
    a. Obtenir l'état actuel du travail avec une demande de sondage long
    b. Si le travail est dans un état autre que terminal (en file d'attente, en cours d'exécution), exécutez de nouveau la boucle.
5. Arrêter lorsque le travail atteint un état terminal (réussite, échec).

## Informations sur le copyright

Copyright © 2024 NetApp, Inc. Tous droits réservés. Imprimé aux États-Unis. Aucune partie de ce document protégé par copyright ne peut être reproduite sous quelque forme que ce soit ou selon quelque méthode que ce soit (graphique, électronique ou mécanique, notamment par photocopie, enregistrement ou stockage dans un système de récupération électronique) sans l'autorisation écrite préalable du détenteur du droit de copyright.

Les logiciels dérivés des éléments NetApp protégés par copyright sont soumis à la licence et à l'avis de non-responsabilité suivants :

CE LOGICIEL EST FOURNI PAR NETAPP « EN L'ÉTAT » ET SANS GARANTIES EXPRESSES OU TACITES, Y COMPRIS LES GARANTIES TACITES DE QUALITÉ MARCHANDE ET D'ADÉQUATION À UN USAGE PARTICULIER, QUI SONT EXCLUES PAR LES PRÉSENTES. EN AUCUN CAS NETAPP NE SERA TENU POUR RESPONSABLE DE DOMMAGES DIRECTS, INDIRECTS, ACCESSOIRES, PARTICULIERS OU EXEMPLAIRES (Y COMPRIS L'ACHAT DE BIENS ET DE SERVICES DE SUBSTITUTION, LA PERTE DE JOUISSANCE, DE DONNÉES OU DE PROFITS, OU L'INTERRUPTION D'ACTIVITÉ), QUELLES QU'EN SOIENT LA CAUSE ET LA DOCTRINE DE RESPONSABILITÉ, QU'IL S'AGISSE DE RESPONSABILITÉ CONTRACTUELLE, STRICTE OU DÉLICTUELLE (Y COMPRIS LA NÉGLIGENCE OU AUTRE) DÉCOULANT DE L'UTILISATION DE CE LOGICIEL, MÊME SI LA SOCIÉTÉ A ÉTÉ INFORMÉE DE LA POSSIBILITÉ DE TELS DOMMAGES.

NetApp se réserve le droit de modifier les produits décrits dans le présent document à tout moment et sans préavis. NetApp décline toute responsabilité découlant de l'utilisation des produits décrits dans le présent document, sauf accord explicite écrit de NetApp. L'utilisation ou l'achat de ce produit ne concède pas de licence dans le cadre de droits de brevet, de droits de marque commerciale ou de tout autre droit de propriété intellectuelle de NetApp.

Le produit décrit dans ce manuel peut être protégé par un ou plusieurs brevets américains, étrangers ou par une demande en attente.

LÉGENDE DE RESTRICTION DES DROITS : L'utilisation, la duplication ou la divulgation par le gouvernement sont sujettes aux restrictions énoncées dans le sous-paragraphe (b)(3) de la clause Rights in Technical Data-Noncommercial Items du DFARS 252.227-7013 (février 2014) et du FAR 52.227-19 (décembre 2007).

Les données contenues dans les présentes se rapportent à un produit et/ou service commercial (tel que défini par la clause FAR 2.101). Il s'agit de données propriétaires de NetApp, Inc. Toutes les données techniques et tous les logiciels fournis par NetApp en vertu du présent Accord sont à caractère commercial et ont été exclusivement développés à l'aide de fonds privés. Le gouvernement des États-Unis dispose d'une licence limitée irrévocable, non exclusive, non cessible, non transférable et mondiale. Cette licence lui permet d'utiliser uniquement les données relatives au contrat du gouvernement des États-Unis d'après lequel les données lui ont été fournies ou celles qui sont nécessaires à son exécution. Sauf dispositions contraires énoncées dans les présentes, l'utilisation, la divulgation, la reproduction, la modification, l'exécution, l'affichage des données sont interdits sans avoir obtenu le consentement écrit préalable de NetApp, Inc. Les droits de licences du Département de la Défense du gouvernement des États-Unis se limitent aux droits identifiés par la clause 252.227-7015(b) du DFARS (février 2014).

## Informations sur les marques commerciales

NETAPP, le logo NETAPP et les marques citées sur le site http://www.netapp.com/TM sont des marques déposées ou des marques commerciales de NetApp, Inc. Les autres noms de marques et de produits sont des marques commerciales de leurs propriétaires respectifs.