Concepts

ONTAP Select

NetApp
November 09, 2026

This PDF was generated from https://docs.netapp.com/fr-fr/ontap-select-9191/concept_api_rest.html on November 09, 2026. Always check docs.netapp.com for the latest.

Sommaire

Concepts . . . 1
- Découvrez les services Web REST pour le déploiement et la gestion des clusters ONTAP Select . . . 1
  - Architecture et contraintes classiques . . . 1
  - Ressources et représentation de l'état . . . 1
  - Points de terminaison URI . . . 1
  - Messages HTTP . . . 1
  - formatage JSON . . . 2
- Découvrez comment accéder à l'API ONTAP Select Deploy . . . 2
  - Déployer l'interface utilisateur native de l'utilitaire . . . 2
  - Page de documentation en ligne ONTAP Select Deploy . . . 2
  - Programme personnalisé . . . 2
- Caractéristiques opérationnelles de base de l'API ONTAP Select Deploy . . . 2
  - Hôte hyperviseur versus nœud ONTAP Select . . . 3
  - identificateurs d'objets . . . 3
  - Identifiants de requête . . . 3
  - Appels synchrones et asynchrones . . . 3
  - Confirmer l'achèvement d'un travail de longue durée . . . 3
  - Sécurité . . . 4
- Transaction API de requête et de réponse pour ONTAP Select . . . 4
  - Variables d'entrée contrôlant une requête API . . . 4
  - Interpréter une réponse d'API . . . 6
- Traitement asynchrone utilisant l'objet Job pour ONTAP Select . . . 7
  - Requêtes asynchrones décrites à l'aide de l'objet Job . . . 7
  - Interroger l'objet Job associé à une requête API . . . 7
  - Procédure générale pour l'émission d'une requête asynchrone . . . 8

Concepts

Découvrez les services Web REST pour le déploiement et la gestion des clusters ONTAP Select

REST (Representational State Transfer) est un style de conception pour les applications web distribuées. Appliqué à la conception d'une API de services web, il définit un ensemble de technologies et de bonnes pratiques pour exposer les ressources basées sur serveur et gérer leur état. Il utilise des protocoles et des normes courants afin de fournir une base flexible pour le déploiement et la gestion des clusters ONTAP Select.

Architecture et contraintes classiques

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

Ressources et représentation de l'état

Les ressources sont les composants de base d'un système web. Lors de la création d'une application de services web REST, les premières tâches de conception comprennent :

- Identification des ressources système ou serveur : Tout système utilise et gère des ressources. Une ressource peut être un fichier, une transaction métier, un processus ou une entité administrative. L'une des premières étapes de la conception d'une application basée sur les services web REST consiste à identifier les ressources.
- Définition des états des ressources et des opérations associées : les ressources se trouvent toujours dans l'un des états finis. Ces états, ainsi que les opérations associées permettant de les modifier, doivent être clairement définis.

Des messages sont échangés entre le client et le serveur pour accéder aux ressources et modifier leur état selon le modèle CRUD générique (Créer, Lire, Mettre à jour et Supprimer).

Points de terminaison URI

Chaque ressource REST doit être définie et accessible via un schéma d'adressage précis. Les points d'accès où les ressources sont localisées 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 semblable à un répertoire de fichiers.

Messages HTTP

Le protocole de transfert hypertexte (HTTP) est utilisé par le client et le serveur de services web pour échanger des requêtes et des réponses concernant les ressources. Lors de la conception d'une application de services web, les verbes HTTP (tels que GET et POST) sont associés aux ressources et aux actions de gestion d'état correspondantes.

Le protocole HTTP est sans état. Par conséquent, pour associer un ensemble de requêtes et de réponses

liées à une même transaction, des informations supplémentaires doivent être incluses dans les en-têtes HTTP accompagnant les flux de données de requête/réponse.

formatage JSON

Bien que les informations puissent être structurées et transférées entre un client et un serveur de plusieurs manières, l'option la plus courante (et celle utilisée avec l'API REST Deploy) est la JavaScript Object Notation (JSON). JSON est une norme industrielle permettant de représenter des structures de données simples en texte brut et sert à transférer les informations d'état décrivant les ressources.

Découvrez comment accéder à l'API ONTAP Select Deploy

Grâce à la flexibilité inhérente des services Web REST, l'API ONTAP Select Deploy peut être accessible de plusieurs manières différentes.

L'API REST incluse avec ONTAP Select Deploy se voit attribuer un numéro de version. Le numéro de version de l'API est indépendant du numéro de version de Deploy. L'utilitaire d'administration ONTAP Select Deploy 9.19.1 inclut la version 3 de l'API REST.

Déployer l'interface utilisateur native de l'utilitaire

La principale façon d'accéder à l'API est via l'interface web ONTAP Select Deploy. Le navigateur effectue des appels à l'API et reformate les données selon la conception de l'interface utilisateur. Vous pouvez également accéder à l'API via l'interface en ligne de commandes de l'utilitaire Deploy.

Page de documentation en ligne ONTAP Select Deploy

La page de documentation en ligne d'ONTAP Select Deploy offre un point d'accès alternatif lors de l'utilisation d'un navigateur. Outre la possibilité d'exécuter directement des appels d'API individuels, cette page fournit également une description détaillée de l'API, incluant les paramètres d'entrée et les autres options pour chaque appel. Les appels d'API sont organisés en plusieurs domaines fonctionnels ou catégories.

Programme personnalisé

Vous pouvez accéder à l'API Deploy à l'aide de plusieurs langages et outils de programmation. Parmi les plus courants figurent Python, Java et cURL. Un programme, un script ou un outil utilisant 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 la possibilité d'automatiser les déploiements ONTAP Select.

Caractéristiques opérationnelles de base de l'API ONTAP Select Deploy

Bien que REST définisse un ensemble commun de technologies et de bonnes pratiques, les détails de chaque API peuvent varier selon les choix de conception. Vous devez connaître les détails et les caractéristiques opérationnelles de l'API ONTAP Select Deploy avant d'utiliser l'API.

Hôte hyperviseur versus nœud ONTAP Select

Un hôte hyperviseur est la plateforme 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 nœud ONTAP Select. Avec la version 3 de l'API REST Deploy, les objets hôte et nœud sont séparés et distincts. Cela permet une relation un-à-plusieurs, où un ou plusieurs nœuds ONTAP Select peuvent s'exécuter sur le même hôte hyperviseur.

identificateurs d'objets

Chaque instance de ressource ou objet se voit attribuer un identifiant unique lors de sa création. Ces identifiants sont globalement uniques au sein d'une instance spécifique d'ONTAP Select Deploy. Après l'émission d'un appel API créant une nouvelle instance d'objet, la valeur de l'identifiant associé est renvoyée à l'appelant dans l' `location`en-tête de la réponse HTTP. Vous pouvez extraire l'identifiant et l'utiliser lors des appels ultérieurs faisant référence à l'instance de ressource.



Le contenu et la structure interne des identifiants d'objet peuvent changer à tout moment. Vous ne devez utiliser les identifiants que dans les appels d'API concernés, lorsque cela est nécessaire pour faire référence aux objets associés.

Identifiants de requête

Chaque requête API réussie se voit attribuer un identifiant unique. L'identifiant est renvoyé dans l `request-id` en-tête de la réponse HTTP associée. Vous pouvez utiliser un identifiant de requête pour faire référence collectivement aux activités d'une transaction requête-réponse API spécifique. Par exemple, vous pouvez récupérer tous les messages d'événement d'une transaction à partir de l'identifiant de la requête.

Appels synchrones et asynchrones

Il existe deux manières principales pour un serveur de traiter une requête HTTP reçue d'un client :

- Synchrone Le serveur exécute la requête immédiatement et répond avec un code d'état 200, 201 ou 204.
- En mode asynchrone, le serveur accepte la requête et renvoie un code d'état 202. Cela indique que le serveur a accepté la requête du client et a lancé une tâche en arrière-plan pour la traiter. Le résultat final (succès ou échec) n'est pas immédiatement disponible et doit être déterminé par des appels API supplémentaires.

Confirmer l'achèvement d'un travail de longue durée

En général, toute opération longue est traitée de manière asynchrone via une tâche en arrière-plan sur le serveur. Avec l'API REST Deploy, chaque tâche en arrière-plan est associée à un objet Job qui assure le suivi de la tâche et fournit des informations, comme son état actuel. Un objet Job, y compris son identifiant unique, est renvoyé dans la réponse HTTP après la création d'une tâche en arrière-plan.

Vous pouvez interroger directement l'objet Job pour déterminer la réussite ou l'échec de l'appel API associé. Consultez la section *traitement asynchrone à l'aide de l'objet Job* pour plus d'informations.

Outre l'utilisation de l'objet Job, il existe d'autres moyens de déterminer la réussite ou l'échec d'une requête, notamment :

- Messages d'événement Vous pouvez récupérer tous les messages d'événement associés à un appel d'API spécifique à l'aide de l'identifiant de requête renvoyé avec la réponse initiale. 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 statut des ressources Plusieurs ressources conservent une valeur d'état ou de statut que vous pouvez interroger pour déterminer indirectement le succès ou l'échec d'une requête.

Sécurité

L'API Deploy utilise les technologies de sécurité suivantes :

- Sécurité de la couche transport : tout le trafic échangé sur le réseau entre le serveur Deploy et le client est chiffré via TLS. L'utilisation du protocole HTTP sur un canal non chiffré n'est pas prise en charge. La version 1.2 de TLS est prise en charge.
- L'authentification HTTP de base est utilisée pour chaque transaction API. Un en-tête HTTP, contenant le nom d'utilisateur et le mot de passe sous forme de chaîne base64, est ajouté à chaque requête.

Transaction API de requête et de réponse pour ONTAP Select

Chaque appel à l'API Deploy est effectué sous forme de requête HTTP adressée à la machine virtuelle Deploy, qui génère une réponse associée au client. Cette paire requête/réponse constitue une transaction API. Avant d'utiliser l'API Deploy, il est recommandé de se familiariser avec les variables d'entrée permettant de contrôler une requête et avec le contenu de la réponse.

Variables d'entrée contrôlant une requête API

Vous pouvez contrôler le traitement d'un appel API grâce aux paramètres définis dans la requête HTTP.

En-têtes de requête

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

- content-type Si le corps de la requête inclut du JSON, cet en-tête doit être défini sur application/json.
- accept Si le corps de la réponse inclut du JSON, cet en-tête doit être défini sur application/json.
- L'authentification Basic doit être configurée avec le nom d'utilisateur et le mot de passe encodés dans une chaîne base64.

Corps de la requête

Le contenu du corps de la requête varie selon l'appel spécifique. Le corps d'une requête HTTP se compose de l'un des éléments suivants :

- Objet JSON contenant des variables d'entrée (telles que le nom d'un nouveau cluster)
- Vide

Filtrer les objets

Lors d'un appel API utilisant la méthode 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 faire correspondre :

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

Outre la correspondance exacte, d'autres opérateurs permettent de renvoyer un ensemble d'objets sur une plage de valeurs. ONTAP Select prend en charge les opérateurs de filtrage présentés ci-dessous.

Opérateur	Description
=	Égal à
<	Moins que
>	Supérieur à
⇐	Inférieur ou égal à
>=	Supérieur ou égal à
	Ou
!	Non égal à
*	joker gourmand

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

Sélection des champs d'objet

Par défaut, une requête API GET ne renvoie que les attributs permettant d'identifier de manière unique le ou les objets. Cet ensemble minimal de champs sert de clé pour chaque objet et varie selon le type d'objet. Vous pouvez sélectionner des propriétés supplémentaires d'objet à l'aide du paramètre de requête fields de la manière suivante :

- Champs peu coûteux : Spécifiez `fields=*`pour récupérer les champs d'objet qui sont stockés dans la mémoire du serveur local ou qui nécessitent peu de traitement pour y accéder.
- Champs coûteux : Spécifiez `fields=**`pour récupérer tous les champs de l'objet, y compris ceux nécessitant un traitement serveur supplémentaire pour y accéder.
- Sélection personnalisée des champs Utilisez `fields=FIELDNAME` pour spécifier le champ exact que vous souhaitez. Lors de la demande de plusieurs champs, les valeurs doivent être séparées par des virgules sans espaces.



En tant que bonne pratique, vous devez toujours identifier précisément les champs que vous souhaitez. Vous ne devez récupérer l'ensemble des champs peu coûteux ou coûteux qu'en cas de besoin. La classification peu coûteux ou coûteux est déterminée par NetApp sur la base d'une analyse interne des performances. La classification d'un champ donné peut changer à tout moment.

Trier les objets de l'ensemble de sortie

Les enregistrements d'une collection de ressources sont renvoyés dans l'ordre par défaut défini par l'objet. Vous pouvez modifier cet ordre à l'aide du paramètre de requête order_by, en spécifiant le nom du champ et le sens de tri comme suit :

```
order_by=<field name> asc|desc
```

Par exemple, vous pouvez trier le champ « type » par ordre décroissant, puis le champ « id » par ordre croissant :

```
order_by=type desc, id asc
```

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

Pagination

Lors d'un appel API 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 requête. Par exemple :
`max_records=20`

Au besoin, vous pouvez combiner ce paramètre avec d'autres paramètres de requête pour affiner les résultats. Par exemple, la requête suivante renvoie jusqu'à 10 événements système générés après l'heure spécifiée :
`time⇒ 2019-04-04T15:41:29.140265Z&max_records=10`

Vous pouvez effectuer plusieurs requêtes pour parcourir les événements (ou tout type d'objet). Chaque appel API suivant doit utiliser une nouvelle valeur temporelle basée sur le dernier événement du dernier ensemble de résultats.

Interpréter une réponse d'API

Chaque requête API génère une réponse renvoyée au client. Vous pouvez examiner la réponse pour déterminer si elle a réussi et récupérer des données supplémentaires si nécessaire.

code d'état HTTP

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

Code	Signification	Description
200	OK	Indique la réussite des appels qui ne créent pas de nouvel objet.
201	Créé	Un objet a été créé avec succès ; l'en-tête de réponse de localisation inclut l'identifiant unique de l'objet.
202	Accepté	Une tâche de fond de longue durée a été lancée pour exécuter la requête, mais l'opération n'est pas encore terminée.
400	Mauvaise demande	La saisie de la requête 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 requête n'existe pas.
405	Méthode non autorisée	Le verbe HTTP utilisé dans la requête n'est pas pris en charge pour la ressource.
409	Conflit	La tentative de création d'un objet a échoué car l'objet existe déjà.
500	Erreur interne	Une erreur interne générale s'est produite sur le serveur.
501	Non mis en œuvre	L'URI est connue mais ne permet pas d'effectuer la requête.

En-têtes de réponse

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

- request-id Chaque requête API réussie se voit attribuer un identifiant de requête unique.
- Lors de la création d'un objet, l'en-tête location inclut l'URL complète du nouvel objet, y compris l'identifiant unique de l'objet.

Corps de la réponse

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

- Un seul objet peut être renvoyé avec un ensemble de champs en fonction de la requête. Par exemple, vous pouvez utiliser GET pour récupérer les propriétés sélectionnées d'un cluster à l'aide de l'identifiant unique.
- Plusieurs objets d'une collection de ressources peuvent être retournés. Dans tous les cas, un format cohérent est utilisé, avec `num_records` indiquant le nombre d'enregistrements et les enregistrements contenant un tableau des instances d'objet. Par exemple, vous pouvez récupérer tous les nœuds définis dans un cluster spécifique.
- Objet Job : Si un appel API est traité de manière asynchrone, un objet Job est renvoyé, servant de point d'ancrage à la tâche en arrière-plan. Par exemple, la requête 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 si vous tentez de créer un cluster avec un nom déjà existant.
- Dans certains cas, aucune donnée n'est renvoyée et le corps de la réponse est vide. Par exemple, le corps de la réponse est vide après utilisation de DELETE pour supprimer un hôte existant.

Traitement asynchrone utilisant l'objet Job pour ONTAP Select

Certains appels à l'API Deploy, notamment ceux qui créent ou modifient une ressource, peuvent prendre plus de temps que d'autres appels. ONTAP Select Deploy traite ces requêtes de longue durée de manière asynchrone.

Requêtes asynchrones décrites à l'aide de l'objet Job

Après un appel API asynchrone, le code de réponse HTTP 202 indique que la requête a été validée et acceptée, mais pas encore terminée. La requête est traitée comme une tâche en arrière-plan qui continue de s'exécuter après la réponse HTTP initiale au client. La réponse inclut l'objet Job associé à la requête, ainsi que son identifiant unique.



Vous devriez consulter la page de documentation en ligne ONTAP Select Deploy pour déterminer quels appels d'API fonctionnent de manière asynchrone.

Interroger l'objet Job associé à une requête API

L'objet Job renvoyé dans la réponse HTTP contient plusieurs propriétés. Vous pouvez interroger la propriété state pour déterminer si la requête a été complétée avec succès. Un objet Job peut se trouver dans l'un des états suivants :

- En file d'attente

- Exécution
- Succès
- Échec

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

- La requête d'interrogation standard renvoie immédiatement l'état actuel de la tâche
- L'état de la tâche lors d'une requête d'interrogation longue n'est renvoyé que lorsque l'un des événements suivants se produit :
  - L'état a changé plus récemment que la valeur de date-heure fournie dans la requête de sondage
  - Le délai d'attente a expiré (1 à 120 secondes)

L'interrogation standard et l'interrogation longue utilisent le même appel API pour interroger un objet Job. Cependant, une requête d'interrogation longue inclut deux paramètres de requête : `poll_timeout` et `last_modified`.



Il est toujours conseillé d'utiliser l'interrogation longue pour réduire la charge de travail sur la machine virtuelle Deploy.

Procédure générale pour l'émission d'une requête asynchrone

Vous pouvez utiliser la procédure générale suivante pour effectuer un appel d'API asynchrone :

1. Effectuez l'appel API asynchrone.
2. Recevez une réponse HTTP 202 indiquant l'acceptation réussie de la requête.
3. Extrayez l'identifiant de l'objet Job du corps de la réponse.
4. Dans une boucle, effectuez les opérations suivantes à chaque cycle :
   a. Obtenez l'état actuel du Job grâce à une requête de sondage longue.
   b. Si la tâche est dans un état non terminal (en file d'attente, en cours d'exécution), réexécutez la boucle.
5. Arrêter lorsque la tâche atteint un état terminal (succès, échec).

Informations sur le copyright

Copyright © 2026 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.