



# **Conceitos**

## **ONTAP Select**

NetApp  
January 29, 2026

# Índice

Conceitos .....	1
Base de serviços web REST para implantação e gerenciamento de clusters ONTAP Select .....	1
Arquitetura e restrições clássicas .....	1
Recursos e representação estatal .....	1
Pontos finais de URI .....	1
Mensagens HTTP .....	1
Formatação JSON .....	2
Como acessar a API ONTAP Select Deploy .....	2
Implantar interface de usuário nativa do utilitário .....	2
Página de documentação on-line do ONTAP Select Deploy .....	2
Programa personalizado .....	2
Versão da API de implantação do ONTAP Select .....	2
Características operacionais básicas da API ONTAP Select Deploy .....	3
Host do hipervisor versus ONTAP Select .....	3
Identificadores de objetos .....	3
Identificadores de solicitação .....	3
Chamadas síncronas e assíncronas .....	3
Confirmar a conclusão de um trabalho de longa duração .....	4
Segurança .....	4
Transação de API de solicitação e resposta para ONTAP Select .....	4
Variáveis de entrada que controlam uma solicitação de API .....	4
Interpretar uma resposta de API .....	6
Processamento assíncrono usando o objeto Job para ONTAP Select .....	8
Solicitações assíncronas descritas usando o objeto Job .....	8
Consultar o objeto Job associado a uma solicitação de API .....	8
Procedimento geral para emissão de uma solicitação assíncrona .....	8

# Conceitos

## Base de serviços web REST para implantação e gerenciamento de clusters ONTAP Select

A Transferência de Estado Representacional (REST) é um estilo para a criação de aplicações web distribuídas. Quando aplicada ao design de uma API de serviços web, ela estabelece um conjunto de tecnologias e práticas recomendadas para expor recursos baseados em servidor e gerenciar seus estados. Ela utiliza protocolos e padrões tradicionais para fornecer uma base flexível para a implantação e o gerenciamento de clusters ONTAP Select .

### Arquitetura e restrições clássicas

O REST foi formalmente articulado por Roy Fielding em seu doutorado "[dissertação](#)" na UC Irvine em 2000. Ela define um estilo arquitetônico por meio de um conjunto de restrições que, coletivamente, aprimoram aplicações web e os protocolos subjacentes. As restrições estabelecem uma aplicação de serviços web RESTful baseada em uma arquitetura cliente/servidor, utilizando um protocolo de comunicação sem estado.

### Recursos e representação estatal

Os recursos são os componentes básicos de um sistema web. Ao criar uma aplicação de serviços web REST, as tarefas iniciais de design incluem:

- Identificação de recursos baseados em sistema ou servidor. Todo sistema utiliza e mantém recursos. Um recurso pode ser um arquivo, uma transação comercial, um processo ou uma entidade administrativa. Uma das primeiras tarefas ao projetar uma aplicação baseada em serviços web REST é identificar os recursos.
- Definição de estados de recursos e operações de estado associadas. Os recursos estão sempre em um de um número finito de estados. Os estados, bem como as operações associadas usadas para afetar as mudanças de estado, devem ser claramente definidos.

As mensagens são trocadas entre o cliente e o servidor para acessar e alterar o estado dos recursos de acordo com o modelo genérico CRUD (Criar, Ler, Atualizar e Excluir).

### Pontos finais de URI

Cada recurso REST deve ser definido e disponibilizado usando um esquema de endereçamento bem definido. Os endpoints onde os recursos estão localizados e identificados usam um Identificador Uniforme de Recursos (URI). O URI fornece uma estrutura geral para a criação de um nome exclusivo para cada recurso na rede. O Localizador Uniforme de Recursos (URL) é um tipo de URI usado com serviços web para identificar e acessar recursos. Os recursos são normalmente expostos em uma estrutura hierárquica semelhante a um diretório de arquivos.

### Mensagens HTTP

O Protocolo de Transferência de Hipertexto (HTTP) é o protocolo usado pelo cliente e servidor de serviços web para trocar mensagens de solicitação e resposta sobre os recursos. Como parte do projeto de um aplicativo de serviços web, verbos HTTP (como GET e POST) são mapeados para os recursos e as ações de gerenciamento de estado correspondentes.

O HTTP não possui estado. Portanto, para associar um conjunto de solicitações e respostas relacionadas em uma única transação, informações adicionais devem ser incluídas nos cabeçalhos HTTP transmitidos com os fluxos de dados de solicitação/resposta.

## **Formatação JSON**

Embora as informações possam ser estruturadas e transferidas entre um cliente e um servidor de diversas maneiras, a opção mais popular (e a usada com a API REST do Deploy) é a JavaScript Object Notation (JSON). JSON é um padrão do setor para representar estruturas de dados simples em texto simples e é usado para transferir informações de estado que descrevem os recursos.

## **Como acessar a API ONTAP Select Deploy**

Devido à flexibilidade inerente dos serviços web REST, a API ONTAP Select Deploy pode ser acessada de diversas maneiras diferentes.

### **Implantar interface de usuário nativa do utilitário**

A principal maneira de acessar a API é por meio da interface de usuário web do ONTAP Select Deploy. O navegador faz chamadas para a API e reformata os dados de acordo com o design da interface de usuário. Você também acessa a API por meio da interface de linha de comando do utilitário Deploy.

### **Página de documentação on-line do ONTAP Select Deploy**

A página de documentação online do ONTAP Select Deploy oferece um ponto de acesso alternativo ao usar um navegador. Além de fornecer uma maneira de executar chamadas de API individuais diretamente, a página também inclui uma descrição detalhada da API, incluindo parâmetros de entrada e outras opções para cada chamada. As chamadas de API são organizadas em diversas áreas ou categorias funcionais.

### **Programa personalizado**

Você pode acessar a API de Implantação usando qualquer uma das diversas linguagens de programação e ferramentas. As opções mais populares incluem Python, Java e cURL. Um programa, script ou ferramenta que utiliza a API atua como um cliente de serviços web REST. O uso de uma linguagem de programação permite que você entenda melhor a API e oferece a oportunidade de automatizar as implantações do ONTAP Select .

## **Versão da API de implantação do ONTAP Select**

A API REST incluída no ONTAP Select Deploy recebe um número de versão. O número da versão da API é independente do número da versão do Deploy. Você deve estar ciente da versão da API incluída na sua versão do Deploy e de como isso pode afetar seu uso da API.

A versão atual do utilitário de administração Deploy inclui a versão 3 da API REST. Versões anteriores do utilitário Deploy incluem as seguintes versões da API:

### **Implantar 2.8 e posterior**

ONTAP Select Deploy 2.8 e todas as versões posteriores incluem a versão 3 da API REST.

## Implantar 2.7.2 e versões anteriores

ONTAP Select Deploy 2.7.2 e todas as versões anteriores incluem a versão 2 da API REST.



As versões 2 e 3 da API REST não são compatíveis. Se você atualizar para o Deploy 2.8 ou posterior a partir de uma versão anterior que incluía a versão 2 da API, será necessário atualizar todo o código existente que acessa diretamente a API, bem como todos os scripts que usam a interface de linha de comando.

## Características operacionais básicas da API ONTAP Select Deploy

Embora o REST estabeleça um conjunto comum de tecnologias e práticas recomendadas, os detalhes de cada API podem variar de acordo com as escolhas de design. Você deve estar ciente dos detalhes e das características operacionais da API ONTAP Select Deploy antes de usá-la.

### Host do hipervisor versus ONTAP Select

Um *host hipervisor* é a plataforma de hardware central que hospeda uma máquina virtual ONTAP Select. Quando uma máquina virtual ONTAP Select é implantada e está ativa em um host hipervisor, a máquina virtual é considerada um *nó ONTAP Select*. Com a versão 3 da API REST de Implantação, os objetos host e nó são separados e distintos. Isso permite um relacionamento um-para-muitos, em que um ou mais nós ONTAP Select podem ser executados no mesmo host hipervisor.

### Identificadores de objetos

Cada instância de recurso ou objeto recebe um identificador exclusivo quando é criado. Esses identificadores são globalmente exclusivos dentro de uma instância específica do ONTAP Select Deploy. Após emitir uma chamada de API que cria uma nova instância de objeto, o valor do ID associado é retornado ao chamador no `location` Cabeçalho da resposta HTTP. Você pode extrair o identificador e usá-lo em chamadas subsequentes ao se referir à instância do recurso.



O conteúdo e a estrutura interna dos identificadores de objeto podem mudar a qualquer momento. Você deve usar os identificadores somente nas chamadas de API aplicáveis, conforme necessário, ao se referir aos objetos associados.

### Identificadores de solicitação

Cada solicitação de API bem-sucedida recebe um identificador exclusivo. O identificador é retornado no `request-id` Cabeçalho da resposta HTTP associada. Você pode usar um identificador de solicitação para se referir coletivamente às atividades de uma única transação de solicitação-resposta de API específica. Por exemplo, você pode recuperar todas as mensagens de eventos de uma transação com base no ID da solicitação.

### Chamadas síncronas e assíncronas

Há duas maneiras principais pelas quais um servidor executa uma solicitação HTTP recebida de um cliente:

- **Síncrono** O servidor executa a solicitação imediatamente e responde com um código de status 200, 201 ou 204.

- Assíncrono: O servidor aceita a solicitação e responde com o código de status 202. Isso indica que o servidor aceitou a solicitação do cliente e iniciou uma tarefa em segundo plano para concluí-la. O sucesso ou a falha final não são imediatamente conhecidos e devem ser determinados por meio de chamadas de API adicionais.

## Confirmar a conclusão de um trabalho de longa duração

Geralmente, qualquer operação que possa levar muito tempo para ser concluída é processada de forma assíncrona usando uma tarefa em segundo plano no servidor. Com a API REST de Implantação, cada tarefa em segundo plano é ancorada por um objeto Job, que rastreia a tarefa e fornece informações, como o estado atual. Um objeto Job, incluindo seu identificador exclusivo, é retornado na resposta HTTP após a criação de uma tarefa em segundo plano.

Você pode consultar o objeto Job diretamente para determinar o sucesso ou a falha da chamada de API associada. Consulte *processamento assíncrono usando o objeto Job* para obter mais informações.

Além de usar o objeto Job, há outras maneiras de determinar o sucesso ou a falha de uma solicitação, incluindo:

- Mensagens de evento: Você pode recuperar todas as mensagens de evento associadas a uma chamada de API específica usando o ID da solicitação retornado com a resposta original. As mensagens de evento geralmente contêm uma indicação de sucesso ou falha e também podem ser úteis na depuração de uma condição de erro.
- Estado ou status do recurso Vários recursos mantêm um valor de estado ou status que você pode consultar para determinar indiretamente o sucesso ou a falha de uma solicitação.

## Segurança

A API de implantação usa as seguintes tecnologias de segurança:

- Segurança da Camada de Transporte: Todo o tráfego enviado pela rede entre o servidor de implantação e o cliente é criptografado por TLS. O uso do protocolo HTTP em um canal não criptografado não é suportado. A versão 1.2 do TLS é suportada.
- Autenticação HTTP: A autenticação básica é usada para todas as transações da API. Um cabeçalho HTTP, que inclui o nome de usuário e a senha em uma string base64, é adicionado a cada solicitação.

## Transação de API de solicitação e resposta para ONTAP Select

Cada chamada à API de implantação é realizada como uma solicitação HTTP para a máquina virtual de implantação, que gera uma resposta associada ao cliente. Esse par solicitação/resposta é considerado uma transação de API. Antes de usar a API de implantação, você deve estar familiarizado com as variáveis de entrada disponíveis para controlar uma solicitação e o conteúdo da saída da resposta.

## Variáveis de entrada que controlam uma solicitação de API

Você pode controlar como uma chamada de API é processada por meio de parâmetros definidos na solicitação HTTP.

## Cabeçalhos de solicitação

Você deve incluir vários cabeçalhos na solicitação HTTP, incluindo:

- **content-type** Se o corpo da solicitação incluir JSON, este cabeçalho deverá ser definido como `application/json`.
- **aceitar** Se o corpo da resposta incluir JSON, este cabeçalho deverá ser definido como `application/json`.
- **autorização** A autenticação básica deve ser definida com o nome de usuário e a senha codificados em uma string base64.

## Corpo da solicitação

O conteúdo do corpo da solicitação varia dependendo da chamada específica. O corpo da solicitação HTTP consiste em um dos seguintes:

- Objeto JSON com variáveis de entrada (como o nome de um novo cluster)
- Vazio

## Filtrar objetos

Ao emitir uma chamada de API que usa GET, você pode limitar ou filtrar os objetos retornados com base em qualquer atributo. Por exemplo, você pode especificar um valor exato para corresponder a:

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

Além da correspondência exata, existem outros operadores disponíveis para retornar um conjunto de objetos em um intervalo de valores. O ONTAP Select suporta os operadores de filtragem mostrados abaixo.

Operador	Descrição
=	Igual a
<	Menor que
>	Maior que
≤	Menor ou igual a
≥	Maior ou igual a
	Ou
!	Não é igual a
*	Curinga ganancioso

Você também pode retornar um conjunto de objetos com base em se um campo específico está definido ou não, usando a palavra-chave `null` ou sua negação (`!null`) como parte da consulta.

## Selecionando campos de objeto

Por padrão, emitir uma chamada de API usando GET retorna apenas os atributos que identificam exclusivamente o(s) objeto(s). Esse conjunto mínimo de campos atua como uma chave para cada objeto e varia de acordo com o tipo de objeto. Você pode selecionar propriedades adicionais do objeto usando o parâmetro de consulta `fields` das seguintes maneiras:

- Campos baratos Especificar `fields=*` para recuperar os campos de objeto que são mantidos na memória do servidor local ou que exigem pouco processamento para acesso.
- Campos caros Especificar `fields=**` para recuperar todos os campos do objeto, incluindo aqueles que exigem processamento adicional do servidor para acesso.
- Seleção de campo personalizado Usar `fields=FIELDNAME` para especificar o campo exato desejado. Ao solicitar vários campos, os valores devem ser separados por vírgulas, sem espaços.



Como prática recomendada, você deve sempre identificar os campos específicos que deseja. Você só deve recuperar o conjunto de campos de baixo custo ou alto custo quando necessário. A classificação de baixo custo e alto custo é determinada pela NetApp com base em análises internas de desempenho. A classificação de um determinado campo pode mudar a qualquer momento.

## Classificar objetos no conjunto de saída

Os registros em uma coleção de recursos são retornados na ordem padrão definida pelo objeto. Você pode alterar a ordem usando o parâmetro de consulta `order_by` com o nome do campo e a direção de classificação da seguinte maneira:

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

Por exemplo, você pode classificar o campo `tipo` em ordem decrescente seguido pelo `id` em ordem crescente:

```
order_by=tipo desc, id asc
```

Ao incluir vários parâmetros, você deve separar os campos com uma vírgula.

## Paginação

Ao emitir uma chamada de API usando GET para acessar uma coleção de objetos do mesmo tipo, todos os objetos correspondentes são retornados por padrão. Se necessário, você pode limitar o número de registros retornados usando o parâmetro de consulta `max_records` na solicitação. Por exemplo:

```
max_records=20
```

Se necessário, você pode combinar este parâmetro com outros parâmetros de consulta para restringir o conjunto de resultados. Por exemplo, o seguinte retorna até 10 eventos do sistema gerados após o tempo especificado:

```
time⇒ 2019-04-04T15:41:29.140265Z&max_records=10
```

Você pode emitir várias solicitações para navegar pelos eventos (ou qualquer tipo de objeto). Cada chamada de API subsequente deve usar um novo valor de tempo com base no evento mais recente no último conjunto de resultados.

## Interpretar uma resposta de API

Cada solicitação de API gera uma resposta ao cliente. Você pode examinar a resposta para determinar se foi bem-sucedida e recuperar dados adicionais, se necessário.

## Código de status HTTP

Os códigos de status HTTP usados pela API REST de implantação são descritos abaixo.



Código	Significado	Descrição
200	OK	Indica sucesso para chamadas que não criam um novo objeto.
201	Criado	Um objeto foi criado com sucesso; o cabeçalho de resposta de localização inclui o identificador exclusivo do objeto.
202	Aceito	Um trabalho em segundo plano de longa execução foi iniciado para executar a solicitação, mas a operação ainda não foi concluída.
400	Pedido ruim	A entrada solicitada não é reconhecida ou é inadequada.
403	Proibido	Acesso negado devido a um erro de autorização.
404	Não encontrado	O recurso mencionado na solicitação não existe.
405	Método não permitido	O verbo HTTP na solicitação não é suportado para o recurso.
409	Conflito	Uma tentativa de criar um objeto falhou porque o objeto já existe.
500	Erro interno	Ocorreu um erro interno geral no servidor.
501	Não implementado	O URI é conhecido, mas não é capaz de executar a solicitação.

## Cabeçalhos de resposta

Vários cabeçalhos são incluídos na resposta HTTP gerada pelo servidor de implantação, incluindo:

- request-id Cada solicitação de API bem-sucedida recebe um identificador de solicitação exclusivo.
- localização Quando um objeto é criado, o cabeçalho de localização inclui o URL completo para o novo objeto, incluindo o identificador exclusivo do objeto.

## Corpo de resposta

O conteúdo da resposta associada a uma solicitação de API varia de acordo com o objeto, o tipo de processamento e o sucesso ou falha da solicitação. O corpo da resposta é renderizado em JSON.

- Objeto único: Um único objeto pode ser retornado com um conjunto de campos com base na solicitação. Por exemplo, você pode usar GET para recuperar propriedades selecionadas de um cluster usando o identificador exclusivo.
- Vários objetos Vários objetos de uma coleção de recursos podem ser retornados. Em todos os casos, há um formato consistente usado, com `num_records` indicando o número de registros e registros contendo uma matriz de instâncias do objeto. Por exemplo, você pode recuperar todos os nós definidos em um cluster específico.
- Objeto Job: Se uma chamada de API for processada de forma assíncrona, um objeto Job será retornado, ancorando a tarefa em segundo plano. Por exemplo, a solicitação POST usada para implantar um cluster é processada de forma assíncrona e retorna um objeto Job.
- Objeto de erro: Se ocorrer um erro, um objeto de erro sempre será retornado. Por exemplo, você receberá um erro ao tentar criar um cluster com um nome que já existe.
- Vazio Em certos casos, nenhum dado é retornado e o corpo da resposta fica vazio. Por exemplo, o corpo da resposta fica vazio após usar DELETE para excluir um host existente.

# Processamento assíncrono usando o objeto Job para ONTAP Select

Algumas chamadas da API de implantação, especialmente aquelas que criam ou modificam um recurso, podem levar mais tempo para serem concluídas do que outras chamadas. O ONTAP Select Deploy processa essas solicitações de longa duração de forma assíncrona.

## Solicitações assíncronas descritas usando o objeto Job

Após realizar uma chamada de API executada de forma assíncrona, o código de resposta HTTP 202 indica que a solicitação foi validada e aceita com sucesso, mas ainda não foi concluída. A solicitação é processada como uma tarefa em segundo plano que continua em execução após a resposta HTTP inicial ao cliente. A resposta inclui o objeto Job que ancora a solicitação, incluindo seu identificador exclusivo.



Você deve consultar a página de documentação on-line do ONTAP Select Deploy para determinar quais chamadas de API operam de forma assíncrona.

## Consultar o objeto Job associado a uma solicitação de API

O objeto Job retornado na resposta HTTP contém diversas propriedades. Você pode consultar a propriedade `state` para determinar se a solicitação foi concluída com sucesso. Um objeto Job pode estar em um dos seguintes estados:

- Na fila
- Correndo
- Sucesso
- Falha

Há duas técnicas que você pode usar ao pesquisar um objeto Job para detectar um estado terminal para a tarefa, seja sucesso ou falha:

- Solicitação de pesquisa padrão O estado do trabalho atual é retornado imediatamente
- Solicitação de pesquisa longa O estado do trabalho é retornado somente quando ocorre uma das seguintes situações:
  - O estado mudou mais recentemente do que o valor de data e hora fornecido na solicitação de pesquisa
  - O valor de tempo limite expirou (1 a 120 segundos)

A pesquisa padrão e a pesquisa longa usam a mesma chamada de API para consultar um objeto Job. No entanto, uma solicitação de pesquisa longa inclui dois parâmetros de consulta: `poll_timeout` e `last_modified`.



Você deve sempre usar long polling para reduzir a carga de trabalho na máquina virtual Deploy.

## Procedimento geral para emissão de uma solicitação assíncrona

Você pode usar o seguinte procedimento de alto nível para concluir uma chamada de API assíncrona:

1. Emita a chamada de API assíncrona.
2. Receba uma resposta HTTP 202 indicando aceitação bem-sucedida da solicitação.
3. Extraia o identificador do objeto Job do corpo da resposta.
4. Dentro de um loop, execute o seguinte em cada ciclo:
  - a. Obtenha o estado atual do trabalho com uma solicitação de pesquisa longa
  - b. Se o trabalho estiver em um estado não terminal (na fila, em execução), execute o loop novamente.
5. Pare quando o trabalho atingir um estado terminal (sucesso, falha).

## **Informações sobre direitos autorais**

Copyright © 2026 NetApp, Inc. Todos os direitos reservados. Impresso nos EUA. Nenhuma parte deste documento protegida por direitos autorais pode ser reproduzida de qualquer forma ou por qualquer meio — gráfico, eletrônico ou mecânico, incluindo fotocópia, gravação, gravação em fita ou storage em um sistema de recuperação eletrônica — sem permissão prévia, por escrito, do proprietário dos direitos autorais.

O software derivado do material da NetApp protegido por direitos autorais está sujeito à seguinte licença e isenção de responsabilidade:

ESTE SOFTWARE É FORNECIDO PELA NETAPP "NO PRESENTE ESTADO" E SEM QUAISQUER GARANTIAS EXPRESSAS OU IMPLÍCITAS, INCLUINDO, SEM LIMITAÇÕES, GARANTIAS IMPLÍCITAS DE COMERCIALIZAÇÃO E ADEQUAÇÃO A UM DETERMINADO PROPÓSITO, CONFORME A ISENÇÃO DE RESPONSABILIDADE DESTES DOCUMENTOS. EM HIPÓTESE ALGUMA A NETAPP SERÁ RESPONSÁVEL POR QUALQUER DANO DIRETO, INDIRETO, INCIDENTAL, ESPECIAL, EXEMPLAR OU CONSEQUENCIAL (INCLUINDO, SEM LIMITAÇÕES, AQUISIÇÃO DE PRODUTOS OU SERVIÇOS SOBRESSALIENTES; PERDA DE USO, DADOS OU LUCROS; OU INTERRUPÇÃO DOS NEGÓCIOS), INDEPENDENTEMENTE DA CAUSA E DO PRINCÍPIO DE RESPONSABILIDADE, SEJA EM CONTRATO, POR RESPONSABILIDADE OBJETIVA OU PREJUÍZO (INCLUINDO NEGLIGÊNCIA OU DE OUTRO MODO), RESULTANTE DO USO DESTES DOCUMENTOS, MESMO SE ADVERTIDA DA RESPONSABILIDADE DE TAL DANO.

A NetApp reserva-se o direito de alterar quaisquer produtos descritos neste documento, a qualquer momento e sem aviso. A NetApp não assume nenhuma responsabilidade nem obrigação decorrentes do uso dos produtos descritos neste documento, exceto conforme expressamente acordado por escrito pela NetApp. O uso ou a compra deste produto não representam uma licença sob quaisquer direitos de patente, direitos de marca comercial ou quaisquer outros direitos de propriedade intelectual da NetApp.

O produto descrito neste manual pode estar protegido por uma ou mais patentes dos EUA, patentes estrangeiras ou pedidos pendentes.

LEGENDA DE DIREITOS LIMITADOS: o uso, a duplicação ou a divulgação pelo governo estão sujeitos a restrições conforme estabelecido no subparágrafo (b)(3) dos Direitos em Dados Técnicos - Itens Não Comerciais no DFARS 252.227-7013 (fevereiro de 2014) e no FAR 52.227- 19 (dezembro de 2007).

Os dados aqui contidos pertencem a um produto comercial e/ou serviço comercial (conforme definido no FAR 2.101) e são de propriedade da NetApp, Inc. Todos os dados técnicos e software de computador da NetApp fornecidos sob este Contrato são de natureza comercial e desenvolvidos exclusivamente com despesas privadas. O Governo dos EUA tem uma licença mundial limitada, irrevogável, não exclusiva, intransferível e não sublicenciável para usar os Dados que estão relacionados apenas com o suporte e para cumprir os contratos governamentais desse país que determinam o fornecimento de tais Dados. Salvo disposição em contrário no presente documento, não é permitido usar, divulgar, reproduzir, modificar, executar ou exibir os dados sem a aprovação prévia por escrito da NetApp, Inc. Os direitos de licença pertencentes ao governo dos Estados Unidos para o Departamento de Defesa estão limitados aos direitos identificados na cláusula 252.227-7015(b) (fevereiro de 2014) do DFARS.

## **Informações sobre marcas comerciais**

NETAPP, o logotipo NETAPP e as marcas listadas em <http://www.netapp.com/TM> são marcas comerciais da NetApp, Inc. Outros nomes de produtos e empresas podem ser marcas comerciais de seus respectivos proprietários.