

# シングルサインオンが有効な場合は、 **API** を使用します StorageGRID 11.8

NetApp May 17, 2024

This PDF was generated from https://docs.netapp.com/ja-jp/storagegrid-118/admin/use-api-if-sso-is-enabled-adfs.html on May 17, 2024. Always check docs.netapp.com for the latest.

# 目次

シングルサインオンが有効な場合は、 API を使用します · · · · · · · · · · · · · · · · · · ·	'	1
シングルサインオンが有効な場合( Active Directory )は API を使用 · · · · · · · · · · · · · · · · · ·	′	1
シングルサインオンが有効な場合( Azure )は API を使用・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	8	8
シングルサインオンが有効な場合は API を使用( PingFederate ) · · · · · · · · · · · · · · · · · ·	(	9

## シングルサインオンが有効な場合は、 **API** を使用し ます

### シングルサインオンが有効な場合( **Active Directory** )は **API** を使用

ある場合 "シングルサインオン( SSO )の設定と有効化" また、 Active Directory を SSO プロバイダとして使用する場合は、一連の API 要求を問題 で実行して、グリッド 管理 API またはテナント管理 API で有効な認証トークンを取得する必要があります。

シングルサインオンが有効な場合は、 API にサインインします

ここで説明する手順は、 Active Directory を SSO アイデンティティプロバイダとして使用する場合に該当し ます。

作業を開始する前に

- StorageGRID ユーザグループに属するフェデレーテッドユーザの SSO ユーザ名とパスワードが必要です。
- テナント管理 API にアクセスする場合は、テナントアカウント ID を確認しておきます。

このタスクについて

認証トークンを取得するには、次のいずれかの例を使用します。

- storagegrid-ssoauth.py Pythonスクリプト。StorageGRID インストールファイルのディレクトリ にあります (./rpms Red Hat Enterprise Linuxの場合は、./debs UbuntuまたはDebianの場合は、および ./vsphere VMwareの場合)をクリックします。
- ・ cURL 要求のワークフローの例。

cURL ワークフローは、実行に時間がかかりすぎるとタイムアウトする場合があります。次のエラーが表示される場合があります。 A valid SubjectConfirmation was not found on this Response。



cURL ワークフローの例では、パスワードが他のユーザに表示されないように保護されてい ません。

URLエンコード問題 を使用している場合は、次のエラーが表示されることがあります。 Unsupported SAML version。

手順

1. 認証トークンを取得するには、次のいずれかの方法を選択します。

<sup>。</sup>を使用します storagegrid-ssoauth.py Pythonスクリプト。手順2に進みます。

° curl 要求を使用します。手順3に進みます。

2. を使用する場合は、を参照してください storagegrid-ssoauth.py スクリプトを使用して、Pythonイ ンタープリタにスクリプトを渡し、スクリプトを実行します。 プロンプトが表示されたら、次の引数の値を入力します。

- <sup>。</sup>SSO 方式。ADFS または ADFS と入力します。
- 。SSO ユーザ名
- 。StorageGRID がインストールされているドメイン
- ° StorageGRID のアドレス
- <sup>。</sup>テナント管理 API にアクセスする場合は、テナントアカウント ID 。

StorageGRID 認証トークンが出力に表示されます。SSO を使用していない場合の API の使用方法と同様 に、トークンを他の要求に使用できるようになりました。

- 3. cURL 要求を使用する場合は、次の手順 を使用します。
  - a. サインインに必要な変数を宣言します。

export SAMLUSER='my-sso-username' export SAMLPASSWORD='my-password' export SAMLDOMAIN='my-domain' export TENANTACCOUNTID='12345' export STORAGEGRID ADDRESS='storagegrid.example.com' export AD FS ADDRESS='adfs.example.com'



グリッド管理APIにアクセスするには、として0を使用します TENANTACCOUNTID。

b. 署名付き認証URLを受信するには、へのPOST要求を問題 に送信します `/api/v3/authorize-saml`をクリックし、応答からJSONエンコードを削除します。

次の例は、の署名付き認証URLに対するPOST要求を示しています TENANTACCOUNTID。結果はに渡 されます python -m json.tool をクリックしてJSONエンコーディングを削除します。

```
curl -X POST "https://$STORAGEGRID_ADDRESS/api/v3/authorize-saml" \
    -H "accept: application/json" -H "Content-Type: application/json"
\
    --data "{\"accountId\": \"$TENANTACCOUNTID\"}" | python -m
json.tool
```

この例の応答には、 URL エンコードされた署名済み URL が含まれていますが、 JSON エンコードさ れたレイヤは含まれていません。

```
{
    "apiVersion": "3.0",
    "data":
    "https://adfs.example.com/adfs/ls/?SAMLRequest=fZHLbsIwEEV%2FJTuv7...
sSl%2BfQ33cvfwA%3D&RelayState=12345",
    "responseTime": "2018-11-06T16:30:23.355Z",
    "status": "success"
}
```

C. を保存します SAMLRequest 後続のコマンドで使用する応答から。

export SAMLREQUEST='fZHLbsIwEEV%2FJTuv7...sSl%2BfQ33cvfwA%3D'

d. AD FS からクライアント要求 ID を含む完全な URL を取得します。

1 つは、前の応答の URL を使用してログインフォームを要求する方法です。

```
curl "https://$AD_FS_ADDRESS/adfs/ls/?SAMLRequest=
$SAMLREQUEST&RelayState=$TENANTACCOUNTID" | grep 'form method="post"
id="loginForm"'
```

応答にはクライアント要求 ID が含まれています。

```
<form method="post" id="loginForm" autocomplete="off"
novalidate="novalidate" onKeyPress="if (event && event.keyCode == 13)
Login.submitLoginRequest();" action="/adfs/ls/?
SAMLRequest=fZHRToMwFIZfhb...UJikvo77sXPw%3D%3D&RelayState=12345&clie
nt-request-id=00000000-0000-0000-ee02-008000000de" >
```

e. 応答からクライアント要求 ID を保存します。

export SAMLREQUESTID='00000000-0000-ee02-008000000de'

f. 前の応答のフォームアクションにクレデンシャルを送信します。

```
curl -X POST "https://$AD_FS_ADDRESS
/adfs/ls/?SAMLRequest=$SAMLREQUEST&RelayState=$TENANTACCOUNTID&client
-request-id=$SAMLREQUESTID" \
--data "UserName=$SAMLUSER@$SAMLDOMAIN&Password=
$SAMLPASSWORD&AuthMethod=FormsAuthentication" --include
```

AD FS からヘッダーに追加情報 が含まれた 302 リダイレクトが返されます。

(i)

SSO システムで多要素認証( MFA )が有効になっている場合、フォームポストには 2 つ目のパスワードまたはその他のクレデンシャルも含まれます。

HTTP/1.1 302 Found Content-Length: 0 Content-Type: text/html; charset=utf-8 Location: https://adfs.example.com/adfs/ls/?SAMLRequest=fZHRToMwFIZfhb...UJikvo 77sXPw%3D%3D&RelayState=12345&client-request-id=00000000-0000-0000ee02-0080000000de Set-Cookie: MSISAuth=AAEAADAvsHpXk6ApV...pmP0aEiNtJvWY=; path=/adfs; HttpOnly; Secure Date: Tue, 06 Nov 2018 16:55:05 GMT

9. を保存します MSISAuth 応答からのCookie。

export MSISAuth='AAEAADAvsHpXk6ApV...pmP0aEiNtJvWY='

h. 認証 POST からクッキーを使用して、指定した場所に GET 要求を送信します。

```
curl "https://$AD_FS_ADDRESS/adfs/ls/?SAMLRequest=
$SAMLREQUEST&RelayState=$TENANTACCOUNTID&client-request-
id=$SAMLREQUESTID" \
--cookie "MSISAuth=$MSISAuth" --include
```

応答ヘッダーには、あとでログアウトに使用する AD FS セッション情報が含まれます。応答の本文に は、非表示のフォームフィールドに SAMLResponse が含まれています。

```
HTTP/1.1 200 OK
Cache-Control: no-cache, no-store
Pragma: no-cache
Content-Length: 5665
Content-Type: text/html; charset=utf-8
Expires: -1
Server: Microsoft-HTTPAPI/2.0
P3P: ADFS doesn't have P3P policy, please contact your site's admin
for more details
Set-Cookie:
SamlSession=a3dpbnRlcnMtUHJpbWFyeS1BZG1pbi0xNzgmRmFsc2Umcng4NnJDZmFKV
XFxVWx3bkl1MnFuUSUzZCUzZCYmJiYmXzE3MjAyZTA5LThmMDqtNDRkZC04Yzq5LTQ3ND
UxYzA3ZjkzYw==; path=/adfs; HttpOnly; Secure
Set-Cookie: MSISAuthenticated=MTEvNy8yMDE4IDQ6MzI6NTkgUE0=;
path=/adfs; HttpOnly; Secure
Set-Cookie: MSISLoopDetectionCookie=MjAxOC0xMS0wNzoxNjozMjo10VpcMQ==;
path=/adfs; HttpOnly; Secure
Date: Wed, 07 Nov 2018 16:32:59 GMT
<form method="POST" name="hiddenform"
action="https://storagegrid.example.com:443/api/saml-response">
  <input type="hidden" name="SAMLResponse"</pre>
value="PHNhbWxwOlJlc3BvbnN...1scDpSZXNwb25zZT4=" /><input</pre>
type="hidden" name="RelayState" value="12345" />
```

i. を保存します SAMLResponse 非表示フィールドから:

export SAMLResponse='PHNhbWxwOlJlc3BvbnN...1scDpSZXNwb25zZT4='

j. を使用して保存します `SAMLResponse`をクリックして、StorageGRID を作成します/api/samlresponse StorageGRID 認証トークンの生成要求

の場合 `RelayState`をクリックします。グリッド管理APIにサインインする場合は、テナントアカウントIDを使用します。

```
curl -X POST "https://$STORAGEGRID_ADDRESS:443/api/saml-response" \
   -H "accept: application/json" \
   --data-urlencode "SAMLResponse=$SAMLResponse" \
   --data-urlencode "RelayState=$TENANTACCOUNTID" \
   | python -m json.tool
```

応答には認証トークンが含まれています。

```
{
    "apiVersion": "3.0",
    "data": "56eb07bf-21f6-40b7-af0b-5c6cacfb25e7",
    "responseTime": "2018-11-07T21:32:53.486Z",
    "status": "success"
}
```

a. 認証トークンを応答にという名前で保存します MYTOKEN。

export MYTOKEN="56eb07bf-21f6-40b7-af0b-5c6cacfb25e7"

これで、を使用できます MYTOKEN その他の要求の場合は、SSOを使用していない場合のAPIの使用方法と同様です。

シングルサインオンが有効な場合は、 API からサインアウトします

シングルサインオン( SSO )が有効になっている場合は、グリッド管理 API またはテナント管理 API からサ インアウトするための一連の API 要求を問題 で処理する必要があります。 ここで説明する手順は、 Active Directory を SSO アイデンティティプロバイダとして使用する場合に該当します

このタスクについて

必要に応じて、組織のシングルログアウトページからログアウトすることで、StorageGRID APIからサインア ウトできます。または、 StorageGRID からシングルログアウト( SLO )を実行することもできます。この 場合、有効な StorageGRID ベアラトークンが必要です。

手順

1. 署名されたログアウト要求を生成するには、「cookie "sso=true」をSLO APIに渡します。

```
curl -k -X DELETE "https://$STORAGEGRID_ADDRESS/api/v3/authorize" \
-H "accept: application/json" \
-H "Authorization: Bearer $MYTOKEN" \
--cookie "sso=true" \
| python -m json.tool
```

ログアウト URL が返されます。

```
{
    "apiVersion": "3.0",
    "data":
    "https://adfs.example.com/adfs/ls/?SAMLRequest=fZDNboMwEIRfhZ...HcQ%3D%3
D",
    "responseTime": "2018-11-20T22:20:30.839Z",
    "status": "success"
}
```

2. ログアウト URL を保存します。

```
export LOGOUT_REQUEST
='https://adfs.example.com/adfs/ls/?SAMLRequest=fZDNboMwEIRfhZ...HcQ%3D%
3D'
```

3. 要求をログアウト URL に送信し、 SLO を実行して StorageGRID にリダイレクトします。

```
curl --include "$LOGOUT REQUEST"
```

302 応答が返されます。リダイレクト先は API のみのログアウトには適用されません。

```
HTTP/1.1 302 Found
Location: https://$STORAGEGRID_ADDRESS:443/api/saml-
logout?SAMLResponse=fVLLasMwEPwVo7ss%...%23rsa-sha256
Set-Cookie: MSISSignoutProtocol=U2FtbA==; expires=Tue, 20 Nov 2018
22:35:03 GMT; path=/adfs; HttpOnly; Secure
```

4. StorageGRID Bearer トークンを削除します。

StorageGRID Bearer トークンを削除すると、 SSO を使用しない場合と同じように動作します。「cookie "sso=true"」が指定されていない場合、ユーザはSSO状態に影響を与えずにStorageGRIDからログアウト されます。

```
curl -X DELETE "https://$STORAGEGRID_ADDRESS/api/v3/authorize" \
-H "accept: application/json" \
-H "Authorization: Bearer $MYTOKEN" \
--include
```

A 204 No Content 応答として、ユーザがサインアウトしたことが示されます。

### シングルサインオンが有効な場合(Azure)はAPIを使用

ある場合 "シングルサインオン( SSO )の設定と有効化" また、 Azure を SSO プロバ イダとして使用している場合は、 2 つのサンプルスクリプトを使用して、グリッド管理 API またはテナント管理 API で有効な認証トークンを取得できます。

Azure シングルサインオンが有効な場合は、 API にサインインします

以下の手順は、 Azure を SSO アイデンティティプロバイダとして使用する場合に該当します

作業を開始する前に

- StorageGRID ユーザグループに属するフェデレーテッドユーザの SSO E メールアドレスとパスワードが 必要です。
- テナント管理 API にアクセスする場合は、テナントアカウント ID を確認しておきます。

このタスクについて

認証トークンを取得するには、次のサンプルスクリプトを使用します。

- ・。 storagegrid-ssoauth-azure.py Python スクリプト
- ・。 storagegrid-ssoauth-azure.js Node.jsスクリプト

どちらのスクリプトも、StorageGRIDインストールファイルディレクトリにあります。 (./rpms Red Hat Enterprise Linuxの場合は、./debs UbuntuまたはDebianの場合は、および./vsphere VMwareの場合)を クリックします。

Azureと独自のAPI統合を作成するには、を参照してください storagegrid-ssoauth-azure.py スクリプト: Python スクリプトは、 StorageGRID に対して 2 つの要求を直接実行し(まず SAMLRequest を取得し、あとで認証トークンを取得するため)、さらに Node.js スクリプトを呼び出して、 SSO 処理を実行します。

SSO 処理は一連の API 要求を使用して実行できますが、実行するのは簡単ではありません。puppeteer Node.js モジュールは、 Azure SSO インターフェイスを破棄するために使用します。

URLエンコード問題を使用している場合は、次のエラーが表示されることがあります。 Unsupported SAML version。

#### 手順

- 1. 必要な依存関係を次のようにインストールします。
  - a. Node.js をインストールします(を参照) "https://nodejs.org/en/download/")。
  - b. 必要な Node.js モジュール( puppeteer および jsdom )を取り付けます。

npm install -g <module>

2. Python スクリプトを Python インタープリタに渡して、スクリプトを実行します。

Python スクリプトは、対応する Node.js スクリプトを呼び出して、 Azure SSO のインタラクションを実行します。

- 3. プロンプトが表示されたら、次の引数の値を入力します(または、パラメータを使用して渡します)。
  - <sup>。</sup>Azure へのサインインに使用する SSO E メールアドレス
  - ° StorageGRID のアドレス
  - <sup>。</sup>テナント管理 API にアクセスする場合は、テナントアカウント ID
- プロンプトが表示されたら、パスワードを入力し、要求された場合に Azure に対する MFA 認証を提供で きるように準備します。



このスクリプトでは、 MFA が Microsoft Authenticator を使用して実行されていることを前 提として他の形式のMFAをサポートするようにスクリプトを変更する必要がある場合があ ります(テキストメッセージで受信したコードの入力など)。

StorageGRID 認証トークンが出力に表示されます。SSO を使用していない場合の API の使用方法と同様 に、トークンを他の要求に使用できるようになりました。

### シングルサインオンが有効な場合は API を使用( PingFederate)

ある場合 "シングルサインオン(SSO)の設定と有効化" また、 SSO プロバイダとして PingFederate を使用するには、グリッド管理 API またはテナント管理 API で有効な認証 トークンを取得するための一連の API 要求を問題 で処理する必要があります。

シングルサインオンが有効な場合は、API にサインインします

これらの手順は、 SSO アイデンティティプロバイダとして PingFederate を使用している場合に適用されま す

作業を開始する前に

- StorageGRID ユーザグループに属するフェデレーテッドユーザの SSO ユーザ名とパスワードが必要です。
- ・テナント管理 API にアクセスする場合は、テナントアカウント ID を確認しておきます。

このタスクについて

認証トークンを取得するには、次のいずれかの例を使用します。

 storagegrid-ssoauth.py Pythonスクリプト。StorageGRID インストールファイルのディレクトリ にあります (./rpms Red Hat Enterprise Linuxの場合は、./debs UbuntuまたはDebianの場合は、および ./vsphere VMwareの場合)をクリックします。

・ cURL 要求のワークフローの例。

cURL ワークフローは、実行に時間がかかりすぎるとタイムアウトする場合があります。次のエラーが表示される場合があります。 A valid SubjectConfirmation was not found on this Response。



URLエンコード問題 を使用している場合は、次のエラーが表示されることがあります。 Unsupported SAML version。

#### 手順

- 1. 認証トークンを取得するには、次のいずれかの方法を選択します。
  - <sup>。</sup>を使用します storagegrid-ssoauth.py Pythonスクリプト。手順2に進みます。
  - <sup>。</sup> curl 要求を使用します。手順 3 に進みます。
- を使用する場合は、を参照してください storagegrid-ssoauth.py スクリプトを使用して、Pythonインタープリタにスクリプトを渡し、スクリプトを実行します。

プロンプトが表示されたら、次の引数の値を入力します。

- SSO 方式。「PingFederate」の任意のバリエーション(PingFederate、PingFederateなど)を入力で きます。
- <sup>。</sup>SSO ユーザ名
- <sup>。</sup>StorageGRID がインストールされているドメイン。このフィールドは PingFederate には使用されま せん。空白のままにするか、任意の値を入力できます。
- ° StorageGRID のアドレス
- 。テナント管理 API にアクセスする場合は、テナントアカウント ID 。

python3 storagegrid-ssoauth.py
sso_method: pingfederate
saml_user: my-sso-username
saml_domain:
sg_address: storagegrid.example.com
tenant_account_id: 12345
Enter the user's SAML password:
***************************************
***************************************
StorageGRID Auth Token: 56eb07bf-21f6-40b7-afob-5c6cacfb25e7

StorageGRID 認証トークンが出力に表示されます。SSO を使用していない場合の API の使用方法と同様 に、トークンを他の要求に使用できるようになりました。

3. cURL 要求を使用する場合は、次の手順 を使用します。

a. サインインに必要な変数を宣言します。

```
export SAMLUSER='my-sso-username'
export SAMLPASSWORD='my-password'
export TENANTACCOUNTID='12345'
export STORAGEGRID ADDRESS='storagegrid.example.com'
```

( i

グリッド管理APIにアクセスするには、として0を使用します TENANTACCOUNTID。

b. 署名付き認証URLを受信するには、へのPOST要求を問題 に送信します `/api/v3/authorize-saml`をク リックし、応答からJSONエンコードを削除します。

次の例は、 TENANTACCOUNTID の署名済み認証 URL を取得するための POST 要求です。結果は python-m json ツールに渡され、 JSON エンコードが削除されます。

```
curl -X POST "https://$STORAGEGRID_ADDRESS/api/v3/authorize-saml" \
    -H "accept: application/json" -H "Content-Type: application/json"
\
    --data "{\"accountId\": \"$TENANTACCOUNTID\"}" | python -m
json.tool
```

この例の応答には、 URL エンコードされた署名済み URL が含まれていますが、 JSON エンコードさ れたレイヤは含まれていません。

```
"apiVersion": "3.0",
"data": "https://my-pf-baseurl/idp/SSO.saml2?...",
"responseTime": "2018-11-06T16:30:23.355Z",
"status": "success"
}
```

C. を保存します SAMLRequest 後続のコマンドで使用する応答から。

export SAMLREQUEST="https://my-pf-baseurl/idp/SSO.saml2?..."

d. 応答とクッキーをエクスポートし、応答をエコーします。

```
RESPONSE=$(curl -c - "$SAMLREQUEST")
```

```
echo "$RESPONSE" | grep 'input type="hidden" name="pf.adapterId"
id="pf.adapterId"'
```

e. 'pf.adapterID' 値をエクスポートし、応答をエコーします。

export ADAPTER='myAdapter'

```
echo "$RESPONSE" | grep 'base'
```

f. 「 href 」値をエクスポートし(末尾のスラッシュ / を削除)、応答をエコーします。

export BASEURL='https://my-pf-baseurl'

echo "\$RESPONSE" | grep 'form method="POST"'

g. 「 action 」の値をエクスポートします。

export SSOPING='/idp/.../resumeSAML20/idp/SSO.ping'

h. クレデンシャルとともに Cookie を送信する:

```
curl -b <(echo "$RESPONSE") -X POST "$BASEURL$SSOPING" \
--data "pf.username=$SAMLUSER&pf.pass=
$SAMLPASSWORD&pf.ok=clicked&pf.cancel=&pf.adapterId=$ADAPTER"
--include</pre>
```

i. を保存します SAMLResponse 非表示フィールドから:

export SAMLResponse='PHNhbWxwOlJlc3BvbnN...1scDpSZXNwb25zZT4='

j. を使用して保存します `SAMLResponse`をクリックして、StorageGRID を作成します/api/samlresponse StorageGRID 認証トークンの生成要求

の場合 `RelayState`をクリックします。グリッド管理APIにサインインする場合は、テナントアカウントIDを使用します。

```
curl -X POST "https://$STORAGEGRID_ADDRESS:443/api/saml-response" \
    -H "accept: application/json" \
    --data-urlencode "SAMLResponse=$SAMLResponse" \
    --data-urlencode "RelayState=$TENANTACCOUNTID" \
    | python -m json.tool
```

応答には認証トークンが含まれています。

```
{
    "apiVersion": "3.0",
    "data": "56eb07bf-21f6-40b7-af0b-5c6cacfb25e7",
    "responseTime": "2018-11-07T21:32:53.486Z",
    "status": "success"
}
```

a. 認証トークンを応答にという名前で保存します MYTOKEN。

export MYTOKEN="56eb07bf-21f6-40b7-af0b-5c6cacfb25e7"

これで、を使用できます MYTOKEN その他の要求の場合は、SSOを使用していない場合のAPIの使用方法と同様です。

シングルサインオンが有効な場合は、 API からサインアウトします

シングルサインオン( SSO )が有効になっている場合は、グリッド管理 API またはテナント管理 API からサ インアウトするための一連の API 要求を問題 で処理する必要があります。 これらの手順は、 SSO アイデン ティティプロバイダとして PingFederate を使用している場合に適用されます

このタスクについて

必要に応じて、組織のシングルログアウトページからログアウトすることで、StorageGRID APIからサインア ウトできます。または、 StorageGRID からシングルログアウト( SLO )を実行することもできます。この 場合、有効な StorageGRID ベアラトークンが必要です。

手順

1. 署名されたログアウト要求を生成するには、「cookie "sso=true」をSLO APIに渡します。

```
curl -k -X DELETE "https://$STORAGEGRID_ADDRESS/api/v3/authorize" \
-H "accept: application/json" \
-H "Authorization: Bearer $MYTOKEN" \
-cookie "sso=true" \
| python -m json.tool
```

ログアウト URL が返されます。

```
{
    "apiVersion": "3.0",
    "data": "https://my-ping-
url/idp/SLO.saml2?SAMLRequest=fZDNboMwEIRfhZ...HcQ%3D%3D",
    "responseTime": "2021-10-12T22:20:30.839Z",
    "status": "success"
}
```

2. ログアウト URL を保存します。

```
export LOGOUT_REQUEST='https://my-ping-
url/idp/SLO.saml2?SAMLRequest=fZDNboMwEIRfhZ...HcQ%3D%3D'
```

3. 要求をログアウト URL に送信し、 SLO を実行して StorageGRID にリダイレクトします。

```
curl --include "$LOGOUT REQUEST"
```

302 応答が返されます。リダイレクト先は API のみのログアウトには適用されません。

```
HTTP/1.1 302 Found
Location: https://$STORAGEGRID_ADDRESS:443/api/saml-
logout?SAMLResponse=fVLLasMwEPwVo7ss%...%23rsa-sha256
Set-Cookie: PF=QoKs...SgCC; Path=/; Secure; HttpOnly; SameSite=None
```

4. StorageGRID Bearer トークンを削除します。

StorageGRID Bearer トークンを削除すると、 SSO を使用しない場合と同じように動作します。「cookie "sso=true"」が指定されていない場合、ユーザはSSO状態に影響を与えずにStorageGRIDからログアウト されます。

```
curl -X DELETE "https://$STORAGEGRID_ADDRESS/api/v3/authorize" \
-H "accept: application/json" \
-H "Authorization: Bearer $MYTOKEN" \
--include
```

A 204 No Content 応答として、ユーザがサインアウトしたことが示されます。

HTTP/1.1 204 No Content

Copyright © 2024 NetApp, Inc. All Rights Reserved. Printed in the U.S.このドキュメントは著作権によって保護されています。著作権所有者の書面による事前承諾がある場合を除き、画像媒体、電子媒体、および写真複写、記録媒体、テープ媒体、電子検索システムへの組み込みを含む機械媒体など、いかなる形式および方法による複製も禁止します。

ネットアップの著作物から派生したソフトウェアは、次に示す使用許諾条項および免責条項の対象となりま す。

このソフトウェアは、ネットアップによって「現状のまま」提供されています。ネットアップは明示的な保 証、または商品性および特定目的に対する適合性の暗示的保証を含み、かつこれに限定されないいかなる暗示 的な保証も行いません。ネットアップは、代替品または代替サービスの調達、使用不能、データ損失、利益損 失、業務中断を含み、かつこれに限定されない、このソフトウェアの使用により生じたすべての直接的損害、 間接的損害、偶発的損害、特別損害、懲罰的損害、必然的損害の発生に対して、損失の発生の可能性が通知さ れていたとしても、その発生理由、根拠とする責任論、契約の有無、厳格責任、不法行為(過失またはそうで ない場合を含む)にかかわらず、一切の責任を負いません。

ネットアップは、ここに記載されているすべての製品に対する変更を随時、予告なく行う権利を保有します。 ネットアップによる明示的な書面による合意がある場合を除き、ここに記載されている製品の使用により生じ る責任および義務に対して、ネットアップは責任を負いません。この製品の使用または購入は、ネットアップ の特許権、商標権、または他の知的所有権に基づくライセンスの供与とはみなされません。

このマニュアルに記載されている製品は、1つ以上の米国特許、その他の国の特許、および出願中の特許によって保護されている場合があります。

権利の制限について:政府による使用、複製、開示は、DFARS 252.227-7013(2014年2月)およびFAR 5252.227-19(2007年12月)のRights in Technical Data -Noncommercial Items(技術データ - 非商用品目に関 する諸権利)条項の(b)(3)項、に規定された制限が適用されます。

本書に含まれるデータは商用製品および / または商用サービス(FAR 2.101の定義に基づく)に関係し、デー タの所有権はNetApp, Inc.にあります。本契約に基づき提供されるすべてのネットアップの技術データおよび コンピュータ ソフトウェアは、商用目的であり、私費のみで開発されたものです。米国政府は本データに対 し、非独占的かつ移転およびサブライセンス不可で、全世界を対象とする取り消し不能の制限付き使用権を有 し、本データの提供の根拠となった米国政府契約に関連し、当該契約の裏付けとする場合にのみ本データを使 用できます。前述の場合を除き、NetApp, Inc.の書面による許可を事前に得ることなく、本データを使用、開 示、転載、改変するほか、上演または展示することはできません。国防総省にかかる米国政府のデータ使用権 については、DFARS 252.227-7015(b)項(2014年2月)で定められた権利のみが認められます。

#### 商標に関する情報

NetApp、NetAppのロゴ、http://www.netapp.com/TMに記載されているマークは、NetApp, Inc.の商標です。その他の会社名と製品名は、それを所有する各社の商標である場合があります。