概念
ONTAP 9

NetApp
April 24, 2024

目次

概念 1
認証サーバとアクセストークン 1
ONTAPクライアント許可のオプション 3
OAuth 2.0の導入シナリオ 7
相互TLSを使用したクライアント認証 10

概念

認証サーバとアクセストークン

認可サーバーは、OAuth 2.0 Authorizationフレームワーク内の中心的なコンポーネントとしていくつかの重要な機能を実行します。

OAuth 2.0認可サーバ

認証サーバは、主にアクセストークンの作成と署名を行います。これらのトークンには、クライアントアプリケーションが保護されたリソースに選択的にアクセスできるように、IDおよび承認情報が含まれています。これらのサーバは通常、相互に分離されており、スタンドアロンの専用サーバとして、またはより大きなIDおよびアクセス管理製品の一部として、いくつかの異なる方法で実装できます。

> OAuth 2.0の機能がより大きなIDおよびアクセス管理製品または解決策内にパッケージ化されている場合は特に、認可サーバーに異なる用語が使用されることがあります。たとえば、*アイデンティティプロバイダ（IdP）*という用語は、*認証サーバ*と同じ意味でよく使用されます。

管理

アクセストークンの発行に加えて、認可サーバーは一般的にWebユーザーインターフェイスを介して関連する管理サービスも提供します。たとえば、次の項目を定義および管理できます。

- ユーザおよびユーザ認証
- スコープ
- テナントとレルムによる管理の分離
- ポリシーの適用
- さまざまな外部サービスへの接続
- その他のIDプロトコル（SAMLなど）のサポート

ONTAPは、OAuth 2.0標準に準拠した認可サーバーと互換性があります。

ONTAPニテイキ

1つ以上の認可サーバをONTAPに定義する必要があります。ONTAPは、各サーバとセキュアに通信してトークンを検証し、クライアントアプリケーションをサポートするその他の関連タスクを実行します。

ONTAP構成の主な側面を以下に示します。も参照してください "OAuth 2.0の導入シナリオ" を参照してください。

アクセストークンの検証方法と検証場所

アクセストークンを検証するには、2つのオプションがあります。

- ローカル検証

  ONTAPは、トークンを発行した認可サーバーから提供された情報に基づいて、アクセストークンをローカルで検証できます。認証サーバから取得された情報はONTAPによってキャッシュされ、定期的に更新され

ます。

- リモートイントロスペクション

  リモートイントロスペクションを使用して、認証サーバーでトークンを検証することもできます。イントロスペクションは、許可された当事者がアクセストークンについて認可サーバーに問い合わせることを可能にするプロトコルです。ONTAPは、アクセストークンから特定のメタデータを抽出し、トークンを検証する方法を提供します。ONTAPは、パフォーマンス上の理由から一部のデータをキャッシュします。

ネットワークの場所

ONTAPはファイアウォールの背後にある可能性があります。この場合は、設定の一部としてプロキシを指定する必要があります。

許可サーバの定義方法

ONTAPに対する認証サーバは、CLI、System Manager、REST APIなどの任意の管理インターフェイスを使用して定義できます。たとえば、CLIでは次のコマンドを使用します。 `security oauth2 client create`。

認証サーバの数

1つのONTAPクラスタに対して最大8つの許可サーバを定義できます。発行者または発行者/オーディエンスの要求が一意である限り、同じ認証サーバを同じONTAPクラスタに複数回定義できます。たとえば、Keycloakでは、異なるレルムを使用する場合は常にこのようになります。

OAuth 2.0アクセストークンの使用

認証サーバによって発行されたOAuth 2.0アクセストークンはONTAPによって検証され、REST APIクライアント要求のロールベースアクセスの決定に使用されます。

アクセストークンの取得

REST APIを使用するONTAPクラスタに定義されている認証サーバからアクセストークンを取得する必要があります。トークンを取得するには、認可サーバーに直接問い合わせる必要があります。

> ONTAPは、問題アクセストークンを使用したり、クライアントからの要求を認可サーバにリダイレクトしたりすることはありません。

トークンの要求方法は、次のようないくつかの要因によって異なります。

- 認可サーバとその設定オプション
- OAuth 2.0認可タイプ
- 要求の問題に使用するクライアントまたはソフトウェアツール

付与タイプ

a_grant_は、OAuth 2.0アクセストークンの要求と受信に使用される、ネットワークフローのセットを含む明確に定義されたプロセスです。クライアント、環境、およびセキュリティの要件に応じて、いくつかの異なる権限付与タイプを使用できます。一般的な付与タイプの一覧を以下の表に示します。

許可タイプ	説明
クライアントクレデンシャル	クレデンシャル（IDや共有シークレットなど）のみを使用する一般的な付与タイプ。クライアントは、リソース所有者と密接な信頼関係を持っていると想定されます。
パスワード	リソース所有者パスワード資格情報付与タイプは、リソース所有者がクライアントとの信頼関係を確立している場合に使用できます。また、レガシーHTTPクライアントをOAuth 2.0に移行する場合にも役立ちます。
認証コード	これは機密クライアントにとって理想的な認可タイプであり、リダイレクトベースのフローに基づいています。アクセストークンとリフレッシュトークンの両方を取得するために使用できます。

JWTの内容

OAuth 2.0アクセストークンはJWT形式です。コンテンツは、設定に基づいて認可サーバによって作成されます。ただし、トークンはクライアントアプリケーションには不透明です。クライアントには、トークンを検査したり、コンテンツを認識したりする理由はありません。

各JWTアクセストークンには、クレームのセットが含まれています。クレームは、発行者の特性と認可サーバーでの管理定義に基づいた認可を記述します。この規格に登録されている請求の一部は、次の表に記載されています。すべての文字列で大文字と小文字が区別されます。

請求	キーワード	説明
発行者	ISS	トークンを発行したプリンシパルを識別します。請求処理はアプリケーション固有です。
件名	サブ	トークンのサブジェクトまたはユーザ。名前のスコープは、グローバルまたはローカルで一意になります。
対象者	豪ドル	トークンの対象となる受信者。文字列の配列として実装されます。
有効期限	有効期限	トークンが期限切れになり、拒否されるまでの時間。

を参照してください "RFC 7519：JSON Webトークン" を参照してください。

ONTAPクライアント許可のオプション

ONTAPクライアント許可をカスタマイズするには、いくつかのオプションを使用できます。承認の決定は、最終的には、アクセストークンに含まれるか、アクセストークンから派生したONTAP RESTロールに基づいて行われます。

> 使用できるのは "ONTAP RESTロール" OAuth 2.0の認可を設定する場合。以前のONTAPの従来のロールはサポートされていません。

はじめに

ONTAP内のOAuth 2.0の実装は、柔軟性と堅牢性を考慮して設計されており、ONTAP環境を保護するために必要なオプションを提供します。大まかには、ONTAPクライアント許可を定義するための3つの主要な設定カテゴリがあります。これらの設定オプションを同時に指定することはできません。

ONTAPでは、構成に応じて最適な1つのオプションが適用されます。を参照してください "ONTAPニヨルアクセスノケツテイホウホウ" を参照して、アクセスを決定するためにONTAPで構成定義をどのように処理するかを確認してください。

OAuth 2.0の自己完結型スコープ

これらのスコープには、1つ以上のカスタムRESTロールが含まれており、それぞれが1つの文字列にカプセル化されています。ONTAPロールの定義には依存しません。認可サーバーでこれらのスコープ文字列を定義する必要があります。

ローカルの**ONTAP**固有の**REST**ロールとユーザ

設定に基づいて、ローカルONTAP ID定義を使用してアクセスを決定できます。オプションは次のとおりです。

- 単一のネームドRESTロール
- ユーザ名とローカルONTAPユーザの照合

指定したロールのscope構文は、* ontap-role-*<URL-encoded-ONTAP-role-name>です。たとえば、ロールが「admin」の場合、スコープ文字列は「ontap-role-admin」になります。

Active Directoryまたは**LDAP**グループ

ローカルONTAPの定義を調べても、アクセスを決定できない場合は、Active Directory（「domain」）またはLDAP（「nsswitch」）グループが使用されます。グループ情報は、次の2つの方法のいずれかで指定できます。

- OAuth 2.0スコープ文字列

  グループメンバーシップを持つユーザがいない場合、クライアントのクレデンシャルフローを使用して機密アプリケーションをサポートします。スコープには* ontap-group-*<URL-encoded-ONTAP-group-name>という名前を付けます。たとえば、グループが「development」の場合、スコープ文字列は「ontap-group-development」になります。

- 「グループ」の主張

  これは、リソース所有者(パスワード付与)フローを使用してADFSによって発行されるアクセストークンを対象としています。

自己完結型**OAuth 2.0**スコープ

自己完結型スコープは、アクセストークンで伝送される文字列です。各ロールは完全なカスタムロール定義であり、アクセスを決定するためにONTAPが必要とするすべての機能が含まれています。スコープは、ONTAP内で定義されているRESTロールとは別のものです。

スコープ文字列の形式

基本レベルでは、スコープは連続した文字列として表され、コロンで区切られた6つの値で構成されます。スコープ文字列で使用されるパラメータについては、以下で説明します。

ONTAPリテラル

スコープはリテラル値で始まる必要があります `ontap` 小文字で入力します。これにより、範囲がONTAPに固有であることが識別されます。

クラスタ

スコープ環境となるONTAPクラスタを定義します。次の値を指定できます。

- クラスタUUID

  単一のクラスタを識別します。

- アスタリスク(*)

  スコープ環境のすべてのクラスタを示します。

ONTAP CLIコマンドを使用できます。 `cluster identity show` をクリックしてクラスタのUUIDを表示します。指定しない場合は、スコープ環境all clustersになります。

ロール

自己完結型スコープに含まれるRESTロールの名前。この値は、ONTAPで検証されたり、ONTAPに定義されている既存のRESTロールと照合されたりすることはありません。この名前はロギングに使用されます。

アクセスレベル

この値は、スコープ内でAPIエンドポイントを使用するときにクライアントアプリケーションに適用されるアクセスレベルを示します。次の表に示す6つの値があります。

アクセスレベル	説明
なし	指定したエンドポイントへのすべてのアクセスを拒否します。
- 読み取り専用	GETを使用した読み取りアクセスのみを許可します。
READ_CREATE	POSTを使用して、読み取りアクセスと新しいリソースインスタンスの作成を許可します。
READ_MODIFY	読み取りアクセスを許可し、PATCHを使用して既存のリソースを更新する機能を許可します。
READ_CREATE _MODIFY	削除以外のすべてのアクセスを許可します。許可される処理は、GET（読み取り）、POST（作成）、およびPATCH（更新）です。
すべて	フルアクセスを許可します。

SVM

クラスタ内のスコープ環境内のSVMの名前。すべてのSVMを示すために、*（アスタリスク）を使用します。

> この機能は、ONTAP 9.14.1では完全にはサポートされていません。SVMのパラメータは無視して、プレースホルダにアスタリスクを使用できます。を確認します "ONTAP リリースノート" をクリックしてSVMの今後のサポートを確認してください。

REST API URI

リソースまたは関連リソースのセットへの完全パスまたは部分パス。文字列は次で始まる必要があります：`/api`。値を指定しない場合は、スコープ環境All APIエンドポイントがONTAPクラスタで指定されます。

範囲の例

自己完結型スコープの例を以下に示します。

ONTAP：：joes-role：read_create_modify：：/api/cluster

このロールを割り当てられたユーザに、 `/cluster` エンドポイント。

CLI管理ツール

自己完結型スコープの管理を容易にし、エラーが発生しにくくするために、ONTAPにはCLIコマンドが用意されています。 `security oauth2 scope` 入力パラメータに基づいてスコープ文字列を生成します。

コマンド `security oauth2 scope` 入力内容に基づいて、次の2つのユースケースがあります。

- 文字列をスコープするCLIパラメータ

  このバージョンのコマンドを使用すると、入力パラメータに基づいてスコープ文字列を生成できます。

- scope string to CLIパラメータ

  このバージョンのコマンドを使用すると、入力スコープ文字列に基づいてコマンドパラメータを生成できます。

例

次の例では、次のコマンド例のあとに出力が含まれたスコープ文字列を生成します。定義は、すべてのクラスタを環境します。

```
security oauth2 scope cli-to-scope -role joes-role -access readonly -api
/api/cluster
```

`ontap:*:joes-role:readonly:*:/api/cluster`

ONTAPニヨルアクセスノケツテイホウホウ

OAuth 2.0を適切に設計および実装するには、ONTAPが許可設定を使用してクライアントのアクセスを決定する方法を理解する必要があります。

ステップ**1**：自己完結型スコープ

アクセストークンに自己完結型のスコープが含まれている場合、ONTAPは最初にそれらのスコープを調べます。自己完結型スコープがない場合は、ステップ2に進みます。

1つ以上の自己完結型スコープが存在する場合、ONTAPは明示的な*allow*または*deny*決定が行われるまで、各スコープを適用します。明示的な決定が行われた場合、処理は終了します。

ONTAPが明示的にアクセスを決定できない場合は、手順2に進みます。

手順**2**：ローカルロールフラグを確認する

ONTAPがフラグの値を調べる `use-local-roles-if-present`。このフラグの値は、ONTAPに定義された認可サーバーごとに個別に設定されます。

- の場合 `true` 手順3に進みます。
- の場合 `false` 処理が終了し、アクセスが拒否されます。

手順**3**：名前付き**ONTAP REST**ロール

アクセストークンに名前付きRESTロールが含まれている場合、ONTAPはそのロールを使用してアクセスを決定します。これにより、常に* allow または deny *の決定が行われ、処理が終了します。

名前付きRESTロールがない場合、またはロールが見つからない場合は、手順4に進みます。

手順**4**：ローカル**ONTAP**ユーザ

アクセストークンからユーザ名を抽出し、ローカルONTAPユーザと照合してみます。

ローカルONTAPユーザが一致した場合、ONTAPはそのユーザ用に定義されたロールを使用してアクセスを決定します。これにより、常に* allow または deny *の決定が行われ、処理が終了します。

ローカルONTAPユーザが一致しない場合、またはアクセストークンにユーザ名がない場合は、手順5に進みます。

手順**5**：グループとロールのマッピング

アクセストークンからグループを抽出し、グループと照合してみます。グループは、Active Directoryまたは同等のLDAPサーバを使用して定義します。

一致するグループがある場合、ONTAPはそのグループに定義されたロールを使用してアクセスを決定します。これにより、常に* allow または deny *の決定が行われ、処理が終了します。

一致するグループがない場合、またはアクセストークンにグループがない場合、アクセスは拒否され、処理は終了します。

OAuth 2.0の導入シナリオ

ONTAPに認可サーバーを定義するときに使用できる設定オプションはいくつかあります。これらのオプションに基づいて、展開環境に適した承認サーバーを作成できます。

設定パラメータの概要

ONTAPに認可サーバーを定義する際には、いくつかの設定パラメータを使用できます。これらのパラメータは、一般にすべての管理インターフェイスでサポートされています。

パラメータ名は、ONTAP管理インターフェイスによって多少異なります。たとえば、リモートイントロスペクションを設定する場合、エンドポイントはCLIコマンドパラメータを使用して識別されます。`-introspection-endpoint`。ただし、System Managerでは、同等のフィールドは_Authorizationサーバトークンイントロスペクション URI_です。すべてのONTAP管理インターフェイスに対応するために、パラメータの一般的な概要が用意されています。正確なパラメータまたはフィールドは、コンテキストに基づいて明確にする必要があります。

パラメータ	説明
名前	ONTAPで認識されている認可サーバの名前。
アプリケーション	ONTAP内部アプリケーション定義環境。これは* http *である必要があります。

パラメータ	説明
発行者URI	トークンを発行するサイトまたは組織を識別するパスを持つFQDN。
プロバイダJWKS URI	ONTAPがアクセストークンの検証に使用するJSON Webキーセットを取得するパスとファイル名を含むFQDN。
JWKS更新間隔	ONTAPがプロバイダーJWKS URIから証明書情報を更新する頻度を決定する時間間隔。値はISO-8601形式で指定します。
イントロスペクションエンドポイント	ONTAPがイントロスペクションを通じてリモートトークン検証を実行するために使用するパスを持つFQDN。
クライアント ID	認可サーバで定義されているクライアントの名前。この値が含まれている場合は、インターフェイスに基づいて関連付けられたクライアントシークレットも指定する必要があります。
発信プロキシ	これは、ONTAPがファイアウォールの背後にある場合に、認可サーバへのアクセスを提供するためです。URIはcurl形式で指定する必要があります。
ローカルロールがある場合は使用	ローカルONTAP定義が使用されているかどうかを判断するブーリアンフラグ（名前付きRESTロールとローカルユーザを含む）。
ユーザ要求の削除	ONTAPがローカルユーザとの照合に使用する別名。を使用します `sub` ローカルユーザ名と一致するアクセストークンのフィールド。

導入シナリオ

いくつかの一般的な導入シナリオを次に示します。これらは、トークン検証がONTAPによってローカルで実行されるか、認証サーバによってリモートで実行されるかに基づいて編成されます。各シナリオには、必要な設定オプションのリストが含まれています。を参照してください "ONTAPでのOAuth 2.0の導入" コンフィギュレーションコマンドの例については、を参照してください。

> 認可サーバを定義したら、ONTAP管理インターフェイスを使用してその設定を表示できます。たとえば、次のコマンドを使用します。 `security oauth2 client show` ONTAP CLIを使用します。

ローカル検証

次の導入シナリオは、ローカルでトークン検証を実行するONTAPに基づいています。

プロキシなしで自己完結型スコープを使用する

これは、OAuth 2.0の自己完結型スコープのみを使用する最も単純な展開です。ローカルONTAP ID定義は使用されません。次のパラメータを指定する必要があります。

- 名前
- アプリケーション（http）
- プロバイダJWKS URI
- 発行者URI

また、認可サーバーでスコープを追加する必要があります。

プロキシで自己完結型スコープを使用する

この展開シナリオでは、OAuth 2.0の自己完結型スコープを使用します。ローカルONTAP ID定義は使用されません。ただし、認可サーバはファイアウォールの内側にあるため、プロキシを設定する必要があります。次のパラメータを指定する必要があります。

- 名前
- アプリケーション（http）
- プロバイダJWKS URI
- 発信プロキシ
- 発行者URI
- 対象者

また、認可サーバーでスコープを追加する必要があります。

ローカルユーザロールとデフォルトユーザ名のマッピングをプロキシで使用する

この導入シナリオでは、ローカルユーザロールとデフォルトのネームマッピングを使用します。リモートユーザ要求では、のデフォルト値が使用されます。 `sub` アクセストークンのこのフィールドはローカルユーザー名と一致するために使用されます。ユーザ名は40文字以下にする必要があります。認証サーバはファイアウォールの内側にあるため、プロキシを設定する必要もあります。次のパラメータを指定する必要があります。

- 名前
- アプリケーション（http）
- プロバイダJWKS URI
- ローカルロールがある場合は使用 (`true`)
- 発信プロキシ
- 発行者

ローカルユーザがONTAPに定義されていることを確認する必要があります。

ローカルユーザロールと代替ユーザ名マッピングをプロキシで使用する

この導入シナリオでは、ローカルユーザロールと代替ユーザ名を使用して、ローカルONTAPユーザを照合します。認証サーバはファイアウォールの背後にあるため、プロキシを設定する必要があります。次のパラメータを指定する必要があります。

- 名前
- アプリケーション（http）
- プロバイダJWKS URI
- ローカルロールがある場合は使用 (`true`)
- リモートユーザの要求
- 発信プロキシ
- 発行者URI
- 対象者

ローカルユーザがONTAPに定義されていることを確認する必要があります。

リモートイントロスペクション

次の展開構成は、イントロスペクションを介してリモートでトークン検証を実行するONTAPに基づいています。

プロキシなしで自己完結型スコープを使用する

これは、OAuth 2.0の自己完結型スコープを使用したシンプルな展開です。ONTAP ID定義は使用されません。次のパラメータを指定する必要があります。

- 名前
- アプリケーション（http）
- イントロスペクションエンドポイント
- クライアント ID
- 発行者URI

認可サーバーでは、スコープ、およびクライアントシークレットを定義する必要があります。

相互TLSを使用したクライアント認証

セキュリティのニーズに応じて、オプションでMutual TLS（MTLS）を設定して強力なクライアント認証を実装できます。OAuth 2.0展開の一部としてONTAPで使用される場合、MTLSはアクセストークンが最初に発行されたクライアントによってのみ使用されることを保証します。

OAuth 2.0を使用した相互TLS

Transport Layer Security（TLS）は、2つのアプリケーション（通常はクライアントブラウザとWebサーバ）間にセキュアな通信チャネルを確立するために使用されます。相互TLSは、クライアント証明書を介してクライアントを強力に識別できるようにすることで、これを拡張します。OAuth 2.0を使用したONTAPクラスタで使用する場合、送信者に制約されたアクセストークンを作成して使用することで、基本的なMTLS機能が拡張されます。

送信者に制約されたアクセストークンは、最初に発行されたクライアントのみが使用できます。この機能をサポートするために、新しい確認請求 (`cnf`）がトークンに挿入されます。フィールドにプロパティが含まれています `x5t#S256` アクセストークンを要求するときに使用されるクライアント証明書のダイジェストを保持します。この値は、トークンの検証の一環としてONTAPによって検証されます。送信者に制約されていない許可サーバーによって発行されたアクセストークンには、追加の確認要求は含まれません。

認可サーバごとにMTLSを個別に使用するようにONTAPを設定する必要があります。たとえば、CLIコマンド `security oauth2 client` パラメータを含む `use-mutual-tls` 次の表に示す3つの値に基づいてMTLS処理を制御します。

> 各構成で、ONTAPによって実行される結果とアクションは、構成パラメータの値、およびアクセストークンとクライアント証明書の内容によって異なります。テーブル内のパラメータは、最小から最も制限の厳しいものに分類されています。

パラメータ	説明
なし	OAuth 2.0相互TLS認証は、認可サーバーでは完全に無効になっています。ONTAPは、確認要求がトークンに含まれている場合やクライアント証明書がTLS接続で提供されている場合でも、MTLSクライアント証明書認証を実行しません。
リクエスト	OAuth 2.0相互TLS認証は、送信者に制約されたアクセストークンがクライアントによって提示された場合に適用されます。つまり、MTLSは、確認請求（財産を含む）の場合にのみ適用されます。 `x5t#S256`）がアクセストークンに含まれています。これがデフォルト設定です。
必須	OAuth 2.0相互TLS認証は、認可サーバーによって発行されたすべてのアクセストークンに適用されます。したがって、すべてのアクセストークンは送信者に制約される必要があります。アクセストークンに確認要求がない場合、または無効なクライアント証明書がある場合、認証およびREST API要求は失敗します。

導入フローの概要

ONTAP環境でOAuth 2.0でMTLSを使用する場合の一般的な手順を以下に示します。を参照してください "RFC 8705：『OAuth 2.0 Mutual-TLS Client Authentication and Certificate-Bound Access Tokens』" 詳細：

手順**1**：クライアント証明書を作成してインストールする

クライアントIDの確立は、クライアントの秘密鍵に関する知識の証明に基づいています。対応する公開鍵は、クライアントから提示された署名付きX.509証明書に配置されます。クライアント証明書の作成手順の概要は次のとおりです。

1. 公開鍵と秘密鍵のペアを生成する
2. 証明書署名要求を作成する
3. CSRファイルを既知のCAに送信する
4. CAが要求を検証し、署名済み証明書を発行

通常、クライアント証明書はローカルのオペレーティングシステムにインストールするか、curlなどの一般的なユーティリティを使用して直接使用できます。

ステップ**2**：**MTLS**を使用するように**ONTAP**を設定する

MTLSを使用するようにONTAPを設定する必要があります。この設定は、認可サーバごとに個別に行われます。たとえば、CLIでは次のコマンドを使用します。 `security oauth2 client` は、オプションのパラメータとともに使用されます。 `use-mutual-tls`。を参照してください "ONTAPでのOAuth 2.0の導入" を参照してください。

手順**3**：クライアントがアクセストークンを要求する

クライアントは、ONTAPに設定された認証サーバからアクセストークンを要求する必要があります。クライアントアプリケーションは、手順1で作成およびインストールした証明書でMTLSを使用する必要があります。

ステップ**4:**認証サーバーがアクセストークンを生成する

認可サーバはクライアント要求を検証し、アクセストークンを生成します。この一部として、クライアント証明書のメッセージダイジェストが作成されます。このダイジェストは、トークンに確認要求として含まれます（フィールド `cnf`）。

手順**5**：クライアントアプリケーションが**ONTAP**にアクセストークンを提示する

クライアントアプリケーションは、ONTAPクラスタへのREST API呼び出しを実行し、アクセストークンを* bearerトークン*として承認要求ヘッダーに含めます。クライアントは、アクセストークンの要求に使用したのと同じ証明書を持つMTLSを使用する必要があります。

ステップ**6: ONTAP**はクライアントとトークンを検証します。

ONTAPは、HTTP要求でアクセストークンと、MTLS処理の一部として使用されるクライアント証明書を受信します。ONTAPは最初にアクセストークンの署名を検証します。設定に基づいて、ONTAPはクライアント証明書のメッセージダイジェストを生成し、トークン内の確認要求* cnf*と比較します。2つの値が一致する場合、ONTAPは、API要求を行うクライアントがアクセストークンが最初に発行されたクライアントと同じであることを確認しました。