OAuth 2.0を使用した認証と許可

ONTAP 9

NetApp
April 24, 2024

目次

OAuth 2.0を使用した認証と許可

ONTAP OAuth 2.0実装の概要

ONTAP 9.14以降では、Open Authorization（OAuth 2.0）フレームワークを使用してONTAPクラスタへのアクセスを制御できます。この機能は、ONTAP CLI、System Manager、REST APIなど、ONTAP管理インターフェイスを使用して設定できます。ただし、OAuth 2.0の承認とアクセス制御の決定は、クライアントがREST APIを使用してONTAPにアクセスする場合にのみ適用できます。

> OAuth 2.0のサポートはONTAP 9.14.0で初めて導入されたため、使用しているONTAPリリースに依存します。を参照してください "ONTAP リリースノート" を参照してください。

機能とメリット

ONTAPでOAuth 2.0を使用する主な機能と利点を以下に説明します。

OAuth 2.0標準のサポート

OAuth 2.0は業界標準の認可フレームワークです。署名付きアクセストークンを使用して、保護されたリソースへのアクセスを制限および制御するために使用されます。OAuth 2.0を使用すると、次のような利点があります。

- 認証設定の多くのオプション
- パスワードを含むクライアントのクレデンシャルは絶対に公開しない
- トークンは構成に基づいて有効期限が切れるように設定できます
- REST APIでの使用に最適

いくつかの一般的な承認サーバーでテスト済み

ONTAPの実装は、OAuth 2.0準拠の認可サーバーと互換性があるように設計されています。次の一般的なサーバまたはサービスでテスト済みです。

- Auth0
- Active Directoryフェデレーションサービス（ADFS）
- キークローク

複数の同時認証サーバのサポート

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。これにより、多様なセキュリティ環境のニーズに柔軟に対応できます。

RESTロールトノトウゴウ

ONTAP認証の決定は、最終的にはユーザまたはグループに割り当てられたRESTロールに基づいて行われます。これらのロールは、自己完結型スコープとしてアクセストークン内で伝送されるか、Active DirectoryまたはLDAPグループとともにローカルONTAP定義に基づいて伝送されます。

送信者に制約されたアクセストークンを使用するオプション

クライアント認証を強化するMutual Transport Layer Security（MTLS）を使用するようにONTAPおよび認可サーバを設定できます。これにより、OAuth 2.0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が保証されます。この機能は、FAPIやMITERによって確立されたものを含む、いくつかの一般的なセキュリティ推奨事項をサポートし、それらと一致しています。

実装と構成

大まかに言えば、OAuth 2.0の実装と構成にはいくつかの側面があり、開始時に考慮する必要があります。

ONTAP内の**OAuth 2.0**エンティティ

OAuth 2.0認証フレームワークは、データセンターまたはネットワーク内の実際の要素または仮想要素にマッピングできる複数のエンティティを定義します。OAuth 2.0エンティティとそのONTAPへの適応を以下の表に示します。

OAuth 2.0エンティティ	説明
リソース	内部ONTAPコマンドを使用してONTAPリソースへのアクセスを提供するREST APIエンドポイント。
リソース所有者	保護されたリソースを作成した、またはデフォルトでそのリソースを所有しているONTAPクラスタユーザ。
リソースサーバ	保護されているリソースのホスト（ONTAPクラスタ）。
クライアント	リソース所有者に代わって、または権限を持ってREST APIエンドポイントへのアクセスを要求するアプリケーション。
許可サーバ	通常、アクセストークンの発行と管理ポリシーの適用を担当する専用サーバです。

コア**ONTAP**構成

OAuth 2.0を有効にして使用するようにONTAPクラスタを設定する必要があります。これには、認可サーバへの接続の確立と、必要なONTAP認可設定の定義が含まれます。この設定は、次のいずれかの管理インターフェイスを使用して実行できます。

- ONTAP コマンドラインインターフェイス
- System Manager の略
- ONTAP REST API

環境およびサポートサービス

ONTAP定義に加えて、認可サーバも設定する必要があります。グループとロールのマッピングを使用している場合は、Active DirectoryグループまたはLDAPに相当するものも設定する必要があります。

サポートされる**ONTAP**クライアント

ONTAP 9.14以降では、REST APIクライアントからOAuth 2.0を使用してONTAPにアクセスできます。REST API呼び出しを実行する前に、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。次に、クライアントは、HTTP認証要求ヘッダーを使用して、このトークンを_bearer token_としてONTAPクラスタに渡します。必要なセキュリティのレベルに応じて、クライアントで証明書を作成してインストールし、MTLSに基づいて送信者に制約されたトークンを使用することもできます。

選択した用語

ONTAPを使用したOAuth 2.0デプロイメントの検討を開始する際には、いくつかの用語について理解しておくと役立ちます。を参照してください "その他のリソース" OAuth 2.0に関する詳細情報へのリンクについては、を参照してください。

アクセストークン

認証サーバーによって発行され、保護されたリソースへのアクセス要求を行うためにOAuth 2.0クライアントアプリケーションによって使用されるトークン。

JSON Webトークン

アクセストークンのフォーマットに使用される標準。JSONは、OAuth 2.0の要求を3つの主要セクションに配置したコンパクトな形式で表現するために使用されます。

送信者に制約されたアクセストークン

Mutual Transport Layer Security（MTLS）プロトコルに基づくオプションの機能。トークンで追加の確認要求を使用することで、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ようになります。

JSON Webキーセット

JWKSは、ONTAPがクライアントから提示されたJWTトークンを検証するために使用する公開鍵の集まりです。キーセットは、通常、認証サーバで専用のURIを使用して使用できます。

適用範囲

スコープは、ONTAP REST APIなどの保護されたリソースへのアプリケーションのアクセスを制限または制御する手段を提供します。これらは、アクセストークン内の文字列として表されます。

ONTAP RESTロール

RESTロールはONTAP 9.6で導入され、ONTAP RBACフレームワークの中核をなす機能です。これらのロールは、ONTAPで引き続きサポートされている以前の従来のロールとは異なります。ONTAPのOAuth 2.0実装では、RESTロールのみがサポートされています。

HTTP認証ヘッダー

REST API呼び出しの一部としてクライアントと関連する権限を識別するためのHTTP要求に含まれるヘッダー。認証と認可の実行方法に応じて、いくつかの種類または実装があります。OAuth 2.0アクセストークンをONTAPに提示する場合、トークンは_bearer token_として識別されます。

HTTPベーシック認証

初期のHTTP認証技術はまだONTAPでサポートされています。プレーンテキストのクレデンシャル（ユーザ名とパスワード）はコロンで連結され、base64でエンコードされます。文字列は認可要求ヘッダーに配置され、サーバに送信されます。

FAPI

OpenID Foundationのワーキンググループで、金融業界向けにプロトコル、データスキーマ、およびセキュリティに関する推奨事項を提供しています。このAPIは元 々 Financial Grade APIとして知られていた。

マイター

米国空軍と米国政府に技術的および安全保障上のガイダンスを提供する民間の非営利企業。

その他のリソース

いくつかの追加リソースを以下に示します。OAuth 2.0と関連規格の詳細については、これらのサイトを参照してください。

プロトコルと標準

- "RFC 6749: OAuth 2.0認可フレームワーク"
- "RFC 7519：JSON Webトークン（JWT）"
- "RFC 7523: OAuth 2.0クライアントの認証と承認のためのJSON Webトークン（JWT）プロファイル"
- "RFC 7662：『OAuth 2.0 Token Introspection』"
- "RFC 7800：『Proof-of-Possession Key for JWT』"
- "RFC 8705：『OAuth 2.0 Mutual-TLS Client Authentication and Certificate-Bound Access Tokens』"

組織

- "OpenID基盤"
- "FAPIワーキンググループ"
- "マイター"
- "IANA-JWT"

製品とサービス

- "Auth0"
- "ADFSの概要"
- "キークローク"

その他のツールとユーティリティ

- "Auth0によるJWT"
- "OpenSSL"

NetAppのドキュメントとリソース

- "ONTAPの自動化" ドキュメント

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。
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。

構成と導入

ONTAPを使用した**OAuth 2.0**の導入準備

ONTAP環境でOAuth 2.0を構成する前に、展開の準備をする必要があります。主なタスクと決定事項の概要を以下に示します。セクションの配置は、通常、従うべき順序に沿って配置されます。ただし、ほとんどの環境に適用できますが、必要に応じて環境に適応する必要があります。また、正式な導入計画の作成も検討する必要があります。

環境に応じて、ONTAPに定義されている認証サーバの設定を選択できます。これには、導入のタイプごとに指定する必要があるパラメータ値も含まれます。を参照してください "OAuth 2.0の導入シナリオ" を参照してください。

リソースとクライアントアプリケーションを保護

OAuth 2.0は、保護されたリソースへのアクセスを制御するための承認フレームワークです。このため、導入の最初の重要なステップは、使用可能なリソースと、それらにアクセスする必要があるクライアントを特定することです。

クライアントアプリケーションを特定する

REST API呼び出しを発行するときにOAuth 2.0を使用するクライアントと、アクセスが必要なAPIエンドポイントを決定する必要があります。

既存の**ONTAP REST**ロールとローカルユーザの確認

RESTロールやローカルユーザなど、既存のONTAP IDの定義を確認する必要があります。OAuth 2.0の設定方法によっては、これらの定義を使用してアクセスを決定できます。

OAuth 2.0へのグローバルな移行

OAuth 2.0認証を段階的に実装することもできますが、各認証サーバーにグローバルフラグを設定することで、すべてのREST APIクライアントをOAuth 2.0にすぐに移動することもできます。これにより、自己完結型スコープを作成することなく、既存のONTAP構成に基づいてアクセスを決定できます。

認証サーバ

認証サーバーは、アクセストークンを発行し、管理ポリシーを適用することで、OAuth 2.0の展開において重要な役割を果たします。

認可サーバーを選択してインストールします。

1つ以上の認可サーバーを選択してインストールする必要があります。スコープの定義方法など、アイデンティティプロバイダの設定オプションと手順を理解することが重要です。

認証ルート**CA**証明書をインストールする必要があるかどうかを判断する

ONTAPでは、認証サーバの証明書を使用して、クライアントから提示された署名済みアクセストークンを検証します。これを行うには、ONTAPにルートCA証明書と中間証明書が必要です。ONTAPがプリインストールされている場合があります。そうでない場合は、インストールする必要があります。

ネットワークの場所と構成の評価

認証サーバがファイアウォールの背後にある場合は、プロキシサーバを使用するようにONTAPを設定する必要があります。

クライアントの認証と許可

クライアントの認証と許可には、いくつかの側面を考慮する必要があります。

自己完結型スコープまたはローカル**ONTAP ID**定義

大まかに言えば、認可サーバーで定義された自己完結型スコープを定義することも、役割やユーザーを含む既存のローカルONTAP ID定義に依存することもできます。

ローカル**ONTAP**処理を使用するオプション

ONTAP ID定義を使用する場合は、適用するものを次のように決定する必要があります。

- ネームドRESTロール
- ローカルユーザの一致
- Active DirectoryまたはLDAPグループ

ローカル検証またはリモートイントロスペクション

アクセストークンがONTAPによってローカルで検証されるか、イントロスペクションによって認可サーバーで検証されるかを決定する必要があります。また、更新間隔など、いくつかの関連する値も考慮する必要があります。

送信者に制約されたアクセストークン

高度なセキュリティが必要な環境では、MTLSに基づいて送信制限付きアクセストークンを使用できます。これには、クライアントごとに証明書が必要です。

管理インターフェイス

OAuth 2.0の管理は、次のいずれかのONTAPインターフェイスを使用して実行できます。

- コマンドラインインターフェイス
- System Manager の略
- REST API

クライアントニヨルアクセストークンノヨウキュウホウホウ

クライアントアプリケーションは、許可サーバからアクセストークンを直接要求する必要があります。許可の種類を含め、これをどのように行うかを決定する必要があります。

ONTAPの設定

ONTAPのいくつかの設定タスクを実行する必要があります。

RESTロールとローカルユーザを定義する

認証設定に基づいて、ローカルのONTAP識別処理を使用できます。この場合は、RESTロールとユーザ定義を確認して定義する必要があります。

コア構成

コアONTAP構成の実行には、主に次の3つの手順が必要です。

- 必要に応じて、認証サーバの証明書に署名したCAのルート証明書（および中間証明書）をインストールします。
- 認可サーバを定義します。
- クラスタに対してOAuth 2.0の処理を有効にします。

ONTAPでの**OAuth 2.0**の導入

OAuth 2.0のコア機能の展開には、主に3つのステップがあります。

作業を開始する前に

ONTAPを設定する前に、OAuth 2.0の展開を準備する必要があります。たとえば、証明書がどのように署名されたか、ファイアウォールの内側にあるかなど、承認サーバーを評価する必要があります。を参照してください "ONTAPを使用したOAuth 2.0の導入準備" を参照してください。

手順**1**：認証サーバ証明書をインストールする

ONTAPには、多数のルートCA証明書が事前にインストールされています。そのため、多くの場合、認証サーバの証明書は追加の設定なしでONTAPによってすぐに認識されます。ただし、許可サーバ証明書の署名方法によっては、ルートCA証明書と中間証明書のインストールが必要になる場合があります。

必要に応じて、次の手順に従って証明書をインストールします。必要な証明書はすべてクラスタレベルでインストールする必要があります。

ONTAPへのアクセス方法に基づいて、正しい手順を選択します。

例 **1.** 手順

System Manager の略

1. System Managerで、**[**クラスタ**]>***[設定]*を選択します。
2. [セキュリティ]*セクションまで下にスクロールします。
3. の横にある→*をクリックします。
4. タブで[追加]*をクリックします。
5. [インポート]*をクリックし、証明書ファイルを選択します。
6. 環境に合わせて設定パラメータを設定します。
7. [追加（ Add ）] をクリックします。

CLI の使用

1. インストールを開始します。

   ```
   security certificate install -type server-ca
   ```

2. 次のコンソールメッセージを確認します。

   ```
   Please enter Certificate: Press <Enter> when done
   ```

3. 証明書ファイルをテキストエディタで開きます。
4. 次の行を含む証明書全体をコピーします。

   ```
   -----BEGIN CERTIFICATE-----

   -----END CERTIFICATE-----
   ```

5. コマンドプロンプトの後に証明書を端末に貼り付けます。
6. Enter*キーを押してインストールを完了します。
7. 次のいずれかを使用して証明書がインストールされていることを確認します。

   ```
   security certificate show-user-installed

   security certificate show
   ```

手順**2**：認証サーバを設定する

ONTAPに対する認可サーバーを少なくとも1つ定義する必要があります。設定と導入計画に基づいてパラメータ値を選択する必要があります。レビュー "OAuth2導入シナリオ" をクリックして、構成に必要な正確なパラメータを決定します。

> 認可サーバー定義を変更するには、既存の定義を削除して新しい定義を作成します。

次の例は、最初のシンプルな導入シナリオに基づいています。 "ローカル検証"。自己完結型スコープはプロキシなしで使用されます。

ONTAPへのアクセス方法に基づいて、正しい手順を選択します。CLI手順では、コマンドを実行する前に置き換える必要があるシンボリック変数を使用します。

例 **2.** 手順

System Manager の略

1. System Managerで、**[**クラスタ**]>***[設定]*を選択します。
2. [セキュリティ]*セクションまで下にスクロールします。
3. * OAuth 2.0 authorization の横にある+*をクリックします。
4. [その他のオプション]*を選択します。
5. 導入に必要な値を次のように指定します。
   - 名前
   - アプリケーション（http）
   - プロバイダJWKS URI
   - 発行者URI
6. [追加（ Add ）] をクリックします。

CLI の使用

1. 定義を再作成します。

```
security oauth2 client create -config-name <NAME> -provider-jwks-uri
<URI_JWKS> -application http -issuer <URI_ISSUER>
```

例：

```
security oauth2 client create \
-config-name auth0 \
-provider-jwks-uri https://superzap.dev.netapp.com:8443/realms/my-
realm/protocol/openid-connect/certs \
-application http \
-issuer https://superzap.dev.netapp.com:8443/realms/my-realm
```

手順**3**：**OAuth 2.0**を有効にする

最後のステップは、OAuth 2.0を有効にすることです。これはONTAPクラスタのグローバル設定です。

> ONTAP、認可サーバー、およびサポートサービスがすべて正しく設定されていることを確認するまで、OAuth 2.0の処理を有効にしないでください。

ONTAPへのアクセス方法に基づいて、正しい手順を選択します。

例 **3.** 手順

System Manager の略

1. System Managerで、**[**クラスタ**]>***[設定]*を選択します。
2. [セキュリティ]セクション*まで下にスクロールします。
3. * OAuth 2.0 authorization の横にある→*をクリックします。
4. * OAuth 2.0認証*を有効にします。

CLI の使用

1. OAuth 2.0を有効にします。

   ```
   security oauth2 modify -enabled true
   ```

2. OAuth 2.0が有効になっていることを確認します。

   ```
   security oauth2 show
   Is OAuth 2.0 Enabled: true
   ```

OAuth 2.0を使用した**REST API**呼び出しの問題

ONTAPのOAuth 2.0実装では、REST APIクライアントアプリケーションがサポートされています。curlを使用して簡単なREST API呼び出しを問題し、OAuth 2.0の使用を開始できます。次の例は、ONTAPクラスタのバージョンを取得します。

作業を開始する前に

ONTAPクラスタに対してOAuth 2.0機能を設定して有効にする必要があります。これには、認可サーバーの定義が含まれます。

ステップ**1**：アクセストークンを取得する

REST API呼び出しで使用するアクセストークンを取得する必要があります。トークン要求はONTAPの外部で実行され、正確な手順は認可サーバとその設定によって異なります。Webブラウザ、curlコマンド、またはプログラミング言語を使用してトークンを要求できます。

説明のために、curlを使用してKeycloakからアクセストークンを要求する方法の例を以下に示します。

キークロークの例

```
curl --request POST \
--location
'https://superzap.dev.netapp.com:8443/realms/peterson/protocol/openid-
connect/token' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'client_id=dp-client-1' \
--data-urlencode 'grant_type=client_credentials' \
--data-urlencode 'client_secret=5iTUf9QKLGxAoYa1iR33vlD5A2xq09V7'
```

返されたトークンをコピーして保存する必要があります。

手順**2**：**REST API**呼び出しを問題する

有効なアクセストークンを取得したら、curlコマンドとアクセストークンを使用してREST API呼び出しを問題できます。

パラメータと変数

curlの例の2つの変数について、次の表で説明します。

変数（ **Variable** ）	説明
$FQDN_IP	ONTAP管理LIFの完全修飾ドメイン名またはIPアドレス。
$access_token	認可サーバーによって発行されたOAuth 2.0アクセストークン。

curlの例を発行する前に、まずBashシェル環境でこれらの変数を設定する必要があります。たとえば、Linux CLIで次のコマンドを入力して、FQDN変数を設定および表示します。

```
FQDN_IP=172.14.31.224
echo $FQDN_IP
172.14.31.224
```

両方の変数をローカルのBashシェルで定義したら、curlコマンドをコピーしてCLIに貼り付けることができます。Enter *を押して変数を置き換え、コマンドを問題します。

カールの例

```
curl --request GET \
--location "https://$FQDN_IP/api/cluster?fields=version" \
--include \
--header "Accept: */*" \
--header "Authorization: Bearer $ACCESS_TOKEN"
```