



Tempered – Azure AD Integration Guide

Setting up your Conductor to use the Authentication Platform Azure AD and optionally MFA for Conductor users and/or Remote Access Users

Table of Contents

Introduction	3
Overview	3
Preparing to Configure OIDC Integration.....	5
Integrate Third-party Authentication with OpenID Connect	6
User Roles	6
Multi-factor Authentication.....	7
Integrate Authentication with the Conductor	7
1. Create and configure an application in your authentication provider	8
2. Configure OIDC on the Airwall Conductor.....	9
3. Set up the Airwall Agents.....	10
Require third-party authentication.....	11
For Azure AD - Register Application and Set Up Group Claims.....	11
Verify third-party authentication is working	19
Troubleshooting Third-party Authentication User Login.....	20

Introduction

When using Tempered's Zero Trust Architecture, Software-Defined Perimeter solution all policy is controlled from a central location in the Airwall Conductor. As a result, securing access and using appropriate role-based access control is important when setting up a production environment. Authentication platforms provide significant value for managing user identities, delivering capabilities like single sign-on and multi-factor authentication (MFA). When companies already have an authentication platform provider or are looking to add MFA support to Tempered's networking security solution integrating these two solutions can often be the best architecture.

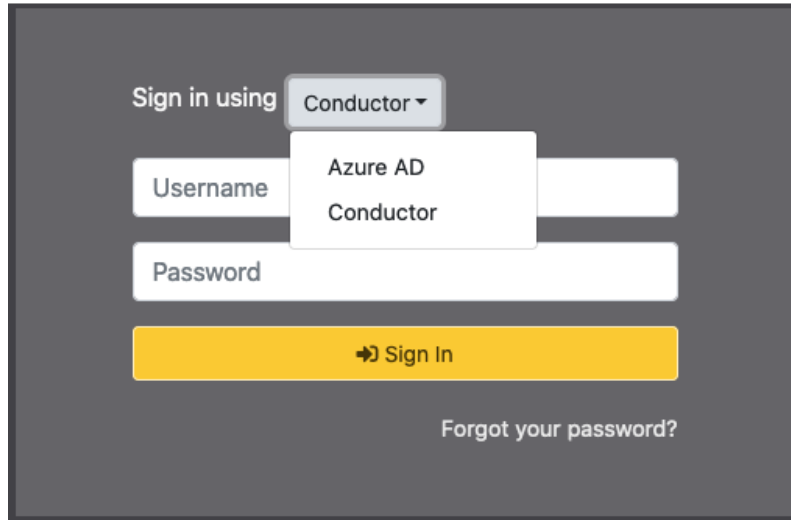
There are two primary use cases for integrating the Conductor with an authentication platform: accessing the Conductor itself and/or providing user authentication for remote users leveraging the Airwall Agent.

The purpose of this document is to provide the necessary steps to integrate Tempered's Conductor with Azure AD's authentication platform using OpenID Connect.

OpenID Connect is a simple identity layer on top of the OAuth 2.0 protocol, which allows computing clients like the Airwall Conductor to verify the identity of an end-user based on the authentication performed by an authorization server – in this case Azure AD, as well as to obtain basic profile information about the end-user in an interoperable and REST-like manner. In technical terms, OpenID Connect specifies a RESTful HTTP API, using JSON as a data format.

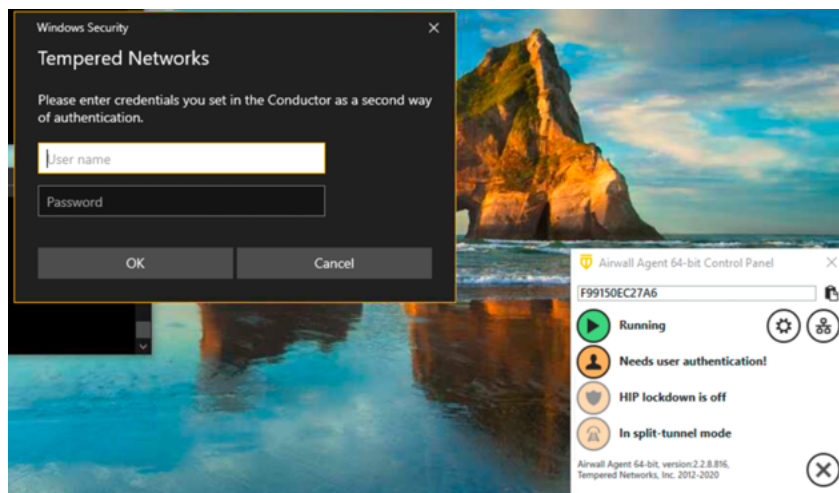
Overview

When a user goes to login to the Airwall Conductor, they can either authenticate using the Conductor's configured users (People) or choose from a pull-down that allows them to authenticate using a third-party authentication platform like Azure AD.



When the user chooses the Azure AD option, they will be redirected to the Azure AD authentication page where they will need to enter their Azure AD credentials and connect using MFA if configured. Once the user is authenticated, they will be redirected back to the Conductor.

For remote access users running Airwall Agents, if user authentication is enabled and linked to an authentication platform, the user will be prompted for credentials by the Agent that is communicating with the Conductor. They will then be redirected to Azure AD and once again they will need to enter their username and password and perform the MFA steps if configured. Once authenticated, the Airwall Agent will be able to connect to the devices configured in their overlay networks.



For either type of integration, the user configured in Azure AD will be part of a group and that information is sent back to the Conductor so that the user can be assigned to the proper role or People Group in the Conductor.

Preparing to Configure OIDC Integration

Gather the following information in preparation for configuring the OpenID connection settings in the Conductor

Parameter	Example	Notes
Authentication Provider Name that will be visible to users	MyCompanyAzureAD	
Conductor URL	https://conductor.mycompany.com	
People Group names that will be configured in both the Conductor and the OpenID provider	cond_system_admins, cond_readonly_admins, cond_network_admins, and cond_remote_users	Note that multiple groups can be configured for each role.
OpenID Connect host	Example: https://sts.windows.net	
OpenID Connect Issuer	Example: https://sts.windows.net/123456e7-2b88-9876-87a6-8ecc41caa76c	
OpenID Connect	Example: 0dfa1234-5678-4ee8-9cf2-f89a9f5a3af7	

Client ID (sometimes called Identifier)		
OpenID Connect Secret	Example: abcdefghijklmnopghc60f3e068f42118fd9a51479d966ace97dc6132c1c11db85aa123456	
OpenID Connect Test User Account		

Integrate Third-party Authentication with OpenID Connect

You can integrate a third-party authentication provider with person authentication in the Conductor using OpenID Connect (OIDC). If your users are already configured for single sign-on (SSO) with a third party, or if you have a large number of users, this integration streamlines your user management.

Note: You can only configure one OpenID Connect provider on the Conductor at a time. If you need to support many OIDC authentication providers simultaneously, you can choose providers that support federated login so you can connect to one provider and have that provider connect to other providers to authenticate users.

Important: To use OpenID Connect on macOS or iOS Airwall Agents, you must have a public certificate on your Conductor.

User Roles

In the Airwall Conductor, you configure person roles in OIDC by including them in groups. The OIDC group names are pre-configured in the Conductor, so when you make a person a member of one of the OIDC groups in the OIDC provider, they are automatically given that role in the Conductor. For instance, you can declare that all members of the OIDC provider's `cond_system_admins` group are system administrators in the Conductor, and that members of the OIDC `cond_remote_users` group are remote-access users.

This is configured in the Conductor when you set up the Authentication Provider. Each group is configured to be in one of four groups that directly link them to the equivalent Role in the Conductor.

Edit Authentication Provider ×

Group settings

Use groups to manage roles

Groups can be used to manage user roles on the Airwall Conductor when enabled. Multiple groups can be specified as a comma-separated list.

System admin groups

cond_system_admins

Read-only admin groups

cond_readonly_admins

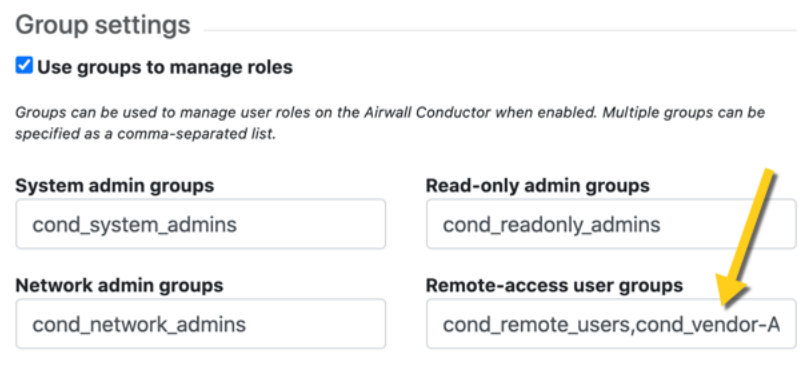
Network admin groups

cond_network_admins

Remote-access user groups

cond_remote_users

For example, if you would like to manage a group of remote access users in a group called vendor-A, you would add that group to the Remote-access user groups box as shown here:



Group settings

Use groups to manage roles

Groups can be used to manage user roles on the Airwall Conductor when enabled. Multiple groups can be specified as a comma-separated list.

System admin groups
cond_system_admins

Read-only admin groups
cond_readonly_admins

Network admin groups
cond_network_admins

Remote-access user groups
cond_remote_users,cond_vendor-A

Note that the groups defined here won't show up in the People Groups tab of the Conductor until a user is added by the Authentication Provider (you can also manually pre-configure it).

Multi-factor Authentication

If your OIDC provider supports multi-factor authentication (MFA), you can use MFA on your provider to require MFA for logging into your Conductor or for Airwall Agent session authentication.

Integrate Authentication with the Conductor

To successfully integrate authentication, you must:

1. Register an application in your authentication provider.
2. Configure OIDC on the Conductor.

3. Set up Airwall Agents.
4. Verify third-party authentication is working

Since each provider is different, refer to the basics required here, and then the Azure AD-specific instructions that follow.

1. Create and configure an application in your authentication provider

Create and configure the application in your provider before connecting it to the Airwall Conductor. Each provider’s workflow is different, but here are the general steps:

1. Create an OpenID Connect application.
2. Configure it with the following information:

Field	Enter
Name	Whatever you want. For example, “Airwall Conductor”
Login Redirect URI	Your Conductor URI followed by <code>/user/auth/openid_connect/callback</code> . For example: https://conductor.mycompany.com/user/auth/openid_connect/callback . Note – If your Conductor is HA paired, add a second login redirect URI, with the same path added.
Logout Redirect URI	Your Conductor URI: https://conductor.mycompany.com (Optional – not typically configured in OneLogin)

3. Depending on your provider, set the authentication method to **basic**, or indicate you are using an **authorization code** for authentication (not a refresh token).
4. Allow the **groups** claim for grant. The **groups** claim is what allows the Conductor to match a user’s group with what role they are given. Because **groups** is not a default OIDC claim, it must be turned on in the provider. For more details, see the specific instructions for Azure AD below.
5. Create four groups: `cond_system_admins`, `cond_readonly_admins`, `cond_network_admins`, and `cond_remote_users` to indicate the four different Conductor roles. Other group names can be added but must be configured in both the Conductor and OIDC provider.
6. Add users to each group so they are assigned the correct role when logging into Conductor.
7. Give your users access to the application you created in your provider.
8. If you want to require MFA to log in, set it up in the OIDC provider. Generally MFA is associated with the app. Please consult your provider documentation for detailed instructions on setting up MFA.

2. Configure OIDC on the Airwall Conductor

1. Go to Conductor **Settings**.
2. Next to Authentication, select Add provider.
3. Select **OpenID Connect** and then select **Next**.
4. On the **Add Authentication Provider** page, under **General settings**, configure the Provider settings as follows (see the Azure AD-specific instructions below for help in finding this information):

For this Setting	Enter
Provider name	Give your provider a descriptive name. This name appears as an option when logging into the Conductor. Example: MyCompany-AzureAD
Conductor host	Host of your Conductor. Must be in the format <code>https://conductor.mycompany.com</code> (no trailing slash)
OpenID Connect host	Must be in the format <code>https://hostname.com:{optional port}</code> Example: <code>https://sts.windows.net</code>
Issuer	Issuer provided by your OIDC provider. Sometimes this value is the same as the OpenID Connect host depending on the provider. Example: <code>https://sts.windows.net/{long customer string}</code>
Client ID (sometimes called Identifier)	Token provided by your OIDC provider associated with the provider application
Secret	Secret token that goes with the Client ID

Note: OpenID Connect logout is not supported with Azure AD.

5. For **HA-paired Conductor host**, enter the Host of your HA Conductor (if applicable).
6. Configure the **Group** settings as follows, and then click **Next**:

For this Setting	Enter
Use groups to manage roles	Checked
System admin groups	Comma-separated list of groups from your provider that will give your user this role. Example: <code>cond_system_admins</code>

Read-only admin groups	Comma-separated list of groups from your provider that will give your user this role. Example: cond_readonly_admins
Network admin groups	Comma-separated list of groups from your provider that will give your user this role. Example: cond_network_admins
Remote-access user groups	Comma-separated list of groups from your provider that will give your user this role. Example: cond_remote_users

Note: If users are in groups that match more than one of the roles, they are given the highest level of access possible (system admin, read-only admin, network admin, then remote-access user).

7. Configure any Group filters you want and click **Finish**.

Group filters

When a user logs in, the Airwall Conductor receives a list of the user's group membership from the authentication provider. This filter limits which of those groups are applied to user role selection and people group membership.

People groups filter



Filter value

8. If you have non-public DNS servers configured in the Conductor under **Global Airwall Agent/client settings**, your users won't be able to reach the public addresses on their devices that include the OpenID Connect providers. You may need to configure DNS servers on the Conductor to add your OpenID Connect provider's DNS server.
9. After changing OIDC configuration, you need to log out and log back in to the Conductor to restart it. When you log back in, you can now choose your third-party authentication provider.

3. Set up the Airwall Agents

Any Airwall Agents authenticating using your third-party provider also need to be set up:

1. Provision and manage Airwall Agents in the Conductor.
2. (Optional starting with 2.2.8) Go to the **Overlays** page, scroll down to **People**, and click **Update**, and add the Airwall Agent as a member.
3. Also check that:
 - a. Airwall Agents are included in your Airwall Relay rules.
 - b. Airwall Agent devices have been added to the appropriate Overlays, and you've set device trust on the Overlays as needed.

Your users should now be able to log in using the third-party authentication provider.

Require third-party authentication

You can also require users to authenticate using the third-party provider either individually (per agent) or globally (all agents). On the agent's **Airwall Agent** tab, or on a **People Group Properties** tab:

- Check **Require Authenticated Airwall Session**.

Airwall agent authentication

Require authenticated Airwall session

Provider **Session timeout** ?

Azure AD Global default (24)

hours

Retain session on service restart ?

Global default (Off)

- For Global configuration, go to Conductor, Settings:

Global Airwall agent authentication settings ?

Require Airwall agent authentication for all agents ⌵

Retain session on service restart * ?

Airwall agent authentication provider * **Session timeout** *

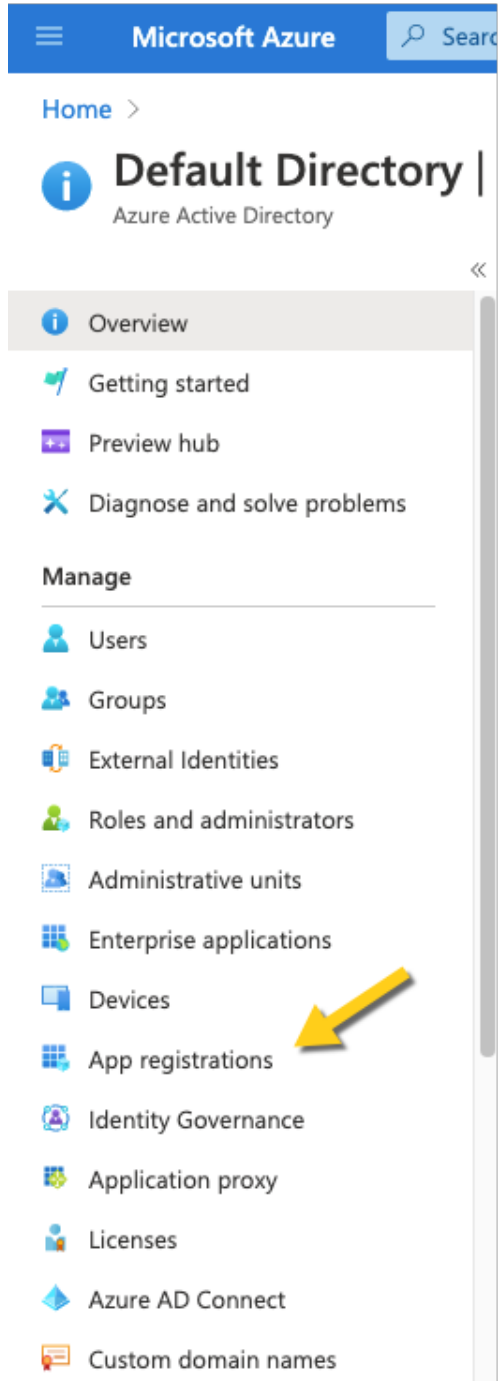
Username and password 24 hours

⚠ Updating the session timeout will not impact any existing sessions

For Azure AD - Register Application and Set Up Group Claims

Here are specific instructions for Azure AD. Note that the Azure AD documentation may be more up-to-date and the settings in your Azure AD account may vary.

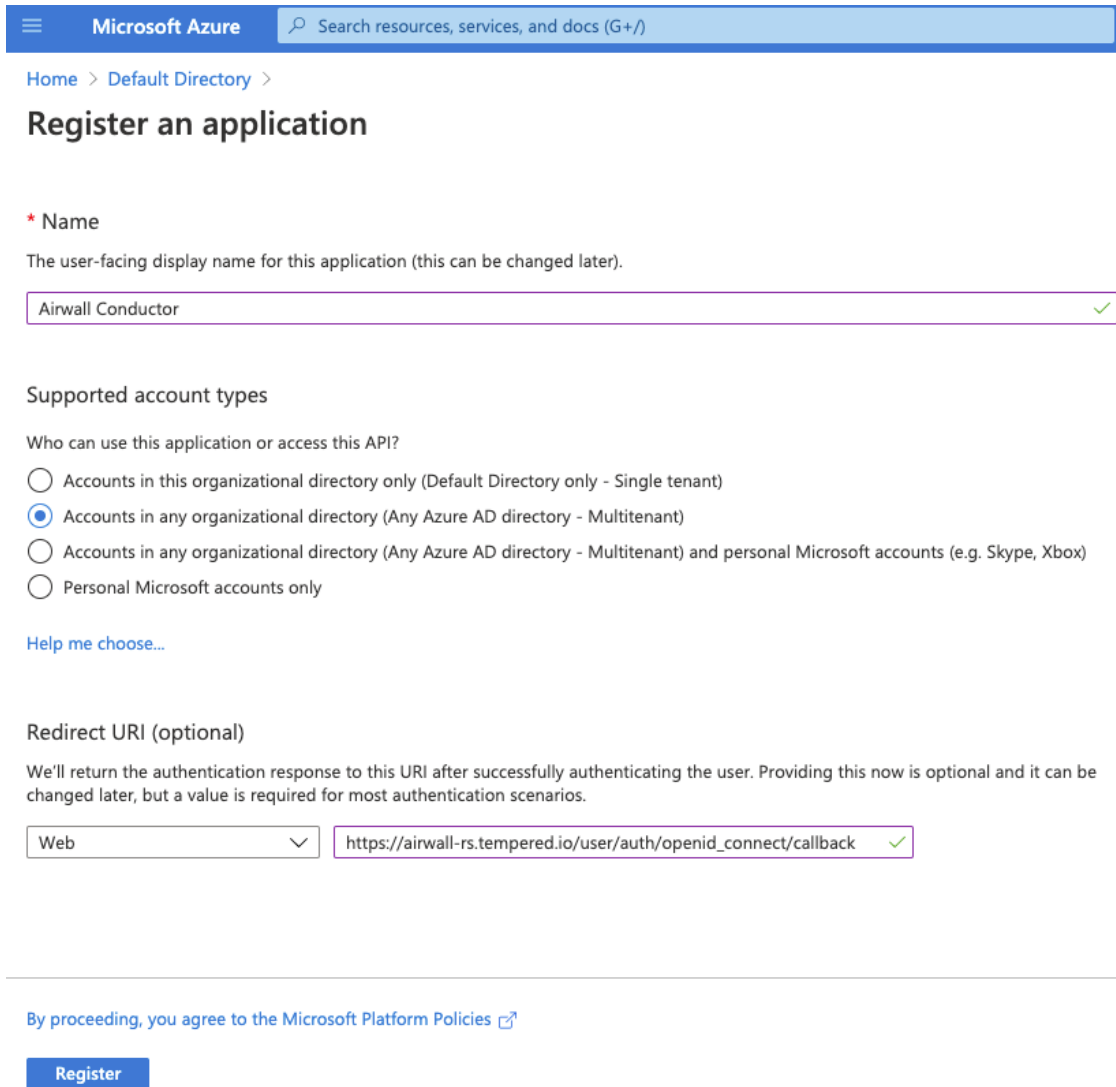
1. In Azure AD, select **App registrations**:



The screenshot shows the Microsoft Azure portal interface. At the top, there is a blue header with the text "Microsoft Azure" and a search bar. Below the header, the breadcrumb "Home >" is visible. The main heading is "Default Directory |" with "Azure Active Directory" underneath. A navigation menu is displayed on the left side, containing the following items:

- Overview
- Getting started
- Preview hub
- Diagnose and solve problems
- Manage**
- Users
- Groups
- External Identities
- Roles and administrators
- Administrative units
- Enterprise applications
- Devices
- App registrations (indicated by a yellow arrow)
- Identity Governance
- Application proxy
- Licenses
- Azure AD Connect
- Custom domain names

2. + New Registration. Enter a Name for the Application (i.e., Airwall Conductor). Select Support account types: Accounts in any organizational directory (Any Azure AD directory – multitenant). Set the Redirect URI to Web, and the URL of your Conductor following by /user/auth/openid_connect/callback. Click Register.



Microsoft Azure Search resources, services, and docs (G+)

Home > Default Directory >

Register an application

*** Name**
The user-facing display name for this application (this can be changed later).

Airwall Conductor ✓

Supported account types

Who can use this application or access this API?

- Accounts in this organizational directory only (Default Directory only - Single tenant)
- Accounts in any organizational directory (Any Azure AD directory - Multitenant)
- Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)
- Personal Microsoft accounts only

[Help me choose...](#)

Redirect URI (optional)

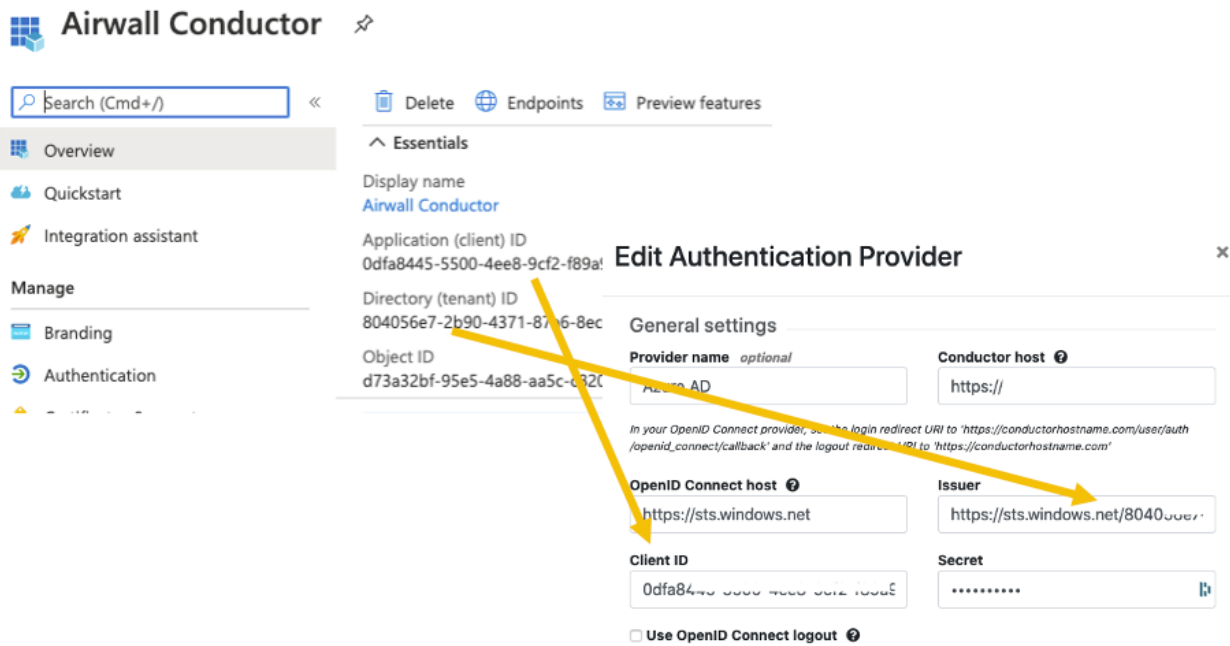
We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.

Web ✓ ✓

[By proceeding, you agree to the Microsoft Platform Policies](#) ↗

[Register](#)

- Once you've registered this Application, Azure AD will provide a set of IDs that will be configured in the Conductor when you set up the OIDC provider. Note that the names will vary in authentication providers, but here are the mappings for Azure AD.



Airwall Conductor

Search (Cmd+/) << Delete Endpoints Preview features

Essentials

Display name
Airwall Conductor

Application (client) ID
0dfa8445-5500-4ee8-9cf2-f89a1d73a32bf

Directory (tenant) ID
804056e7-2b90-4371-8716-8ec0d73a32bf

Object ID
d73a32bf-95e5-4a88-aa5c-0320d73a32bf

Edit Authentication Provider

General settings

Provider name *optional*
Azure AD

Conductor host
https://

In your OpenID Connect provider, set the login redirect URI to 'https://conductorhostname.com/user/auth/openid_connect/callback' and the logout redirect URI to 'https://conductorhostname.com'

OpenID Connect host
https://sts.windows.net

Issuer
https://sts.windows.net/804056e7-2b90-4371-8716-8ec0d73a32bf

Client ID
0dfa8445-5500-4ee8-9cf2-f89a1d73a32bf

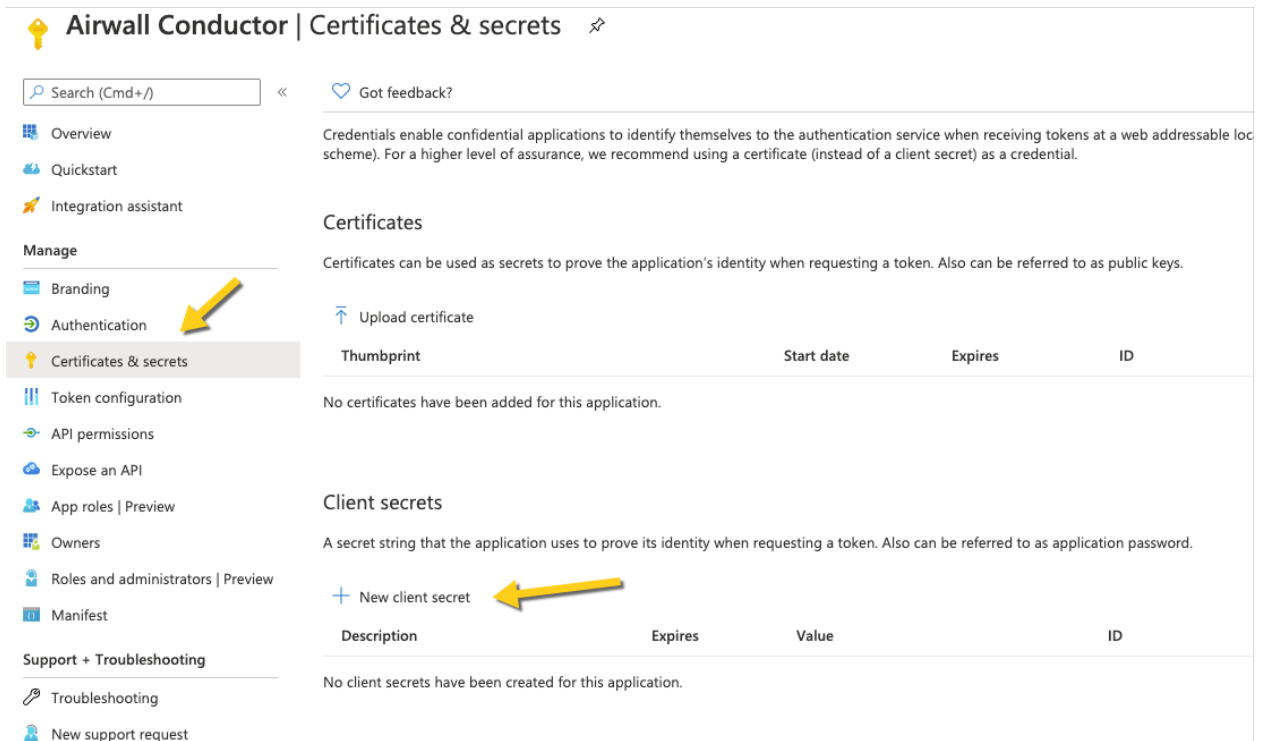
Secret
.....

Use OpenID Connect logout

The Application (client ID) in Azure becomes the Client ID in the Conductor.

The Directory (tenant) ID in Azure gets appended to https://sts.windows.net/ to create the Issuer in the Conductor.

- To create a Client Secret, click on Certificates & secrets and click on + New client secret.



Airwall Conductor | Certificates & secrets

Search (Cmd+/) << Got feedback?

Overview
Quickstart
Integration assistant

Manage

- Branding
- Authentication
- Certificates & secrets**
- Token configuration
- API permissions
- Expose an API
- App roles | Preview
- Owners
- Roles and administrators | Preview
- Manifest

Support + Troubleshooting

- Troubleshooting
- New support request

Credentials enable confidential applications to identify themselves to the authentication service when receiving tokens at a web addressable location. For a higher level of assurance, we recommend using a certificate (instead of a client secret) as a credential.

Certificates

Certificates can be used as secrets to prove the application's identity when requesting a token. Also can be referred to as public keys.

Upload certificate

Thumbprint	Start date	Expires	ID
No certificates have been added for this application.			

Client secrets

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

+ New client secret

Description	Expires	Value	ID
No client secrets have been created for this application.			

- Configure the Description, Expiration timeframe and click Add.

Add a client secret

Description

Expires

In 1 year
 In 2 years
 Never

6. Copy the Value, not the ID to be used as the Secret in the Conductor

Client secrets

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

+ New client secret

Description	Expires	Value	ID
Conductor Client Secret	12/31/2299	1RXRRb--lvZvKowS2ZI43J9_p_Thn~k~3l	eb780cff-a4ed-44d1-956a-97bae7f51bf7

Edit Authentication Provider

General settings

Provider name optional Conductor host
In your OpenID Connect provider, set the login redirect URI to 'https://conductorhostname.com/user/auth/openid_connect/callback' and the logout redirect URI to 'https://conductorhostname.com'

OpenID Connect host Issuer
Client ID Secret
 Use OpenID Connect logout

7. From your new registered application in Azure AD, click on Authentication. Verify that they Implicit grant is set up to use Access tokens.

Microsoft Azure | Search resources, services, and docs (G+)

Home > Default Directory > Airwall Conductor

Airwall Conductor | Authentication

Search (Cmd+) Save Discard Got feedback?

- Overview
- Quickstart
- Integration assistant
- Manage
 - Branding
 - Authentication
 - Certificates & secrets
 - Token configuration
 - API permissions
 - Expose an API
 - App roles | Preview
 - Owners
 - Roles and administrators | Preview
 - Manifest
- Support + Troubleshooting
 - Troubleshooting
 - New support request

Web

Redirect URIs

The URIs we will accept as destinations when returning authentication responses (tokens) after successfully authenticating user.

[Learn more about Redirect URIs and their restrictions](#)

[Add URI](#)

Logout URL

This is where we send a request to have the application clear the user's session data. This is required for single sign-out to work correctly.

Implicit grant

Allows an application to request a token directly from the authorization endpoint. Checking Access tokens and ID tokens is recommended only if the application has a single-page architecture (SPA), has no back-end components, does not use the latest version of MSAL.js with auth code flow, or it invokes a web API via JavaScript. ID Token is needed for ASP.NET Core Web Apps. [Learn more about the implicit grant flow](#)

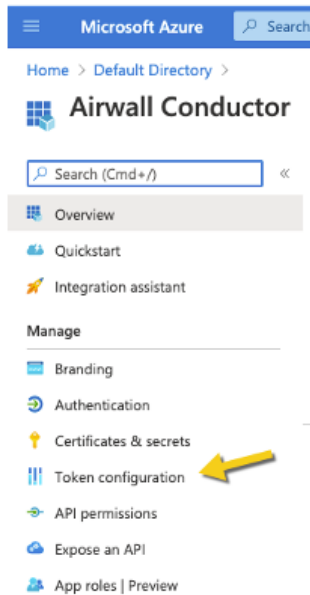
To enable the implicit grant flow, select the tokens you would like to be issued by the authorization endpoint:

Access tokens

ID tokens

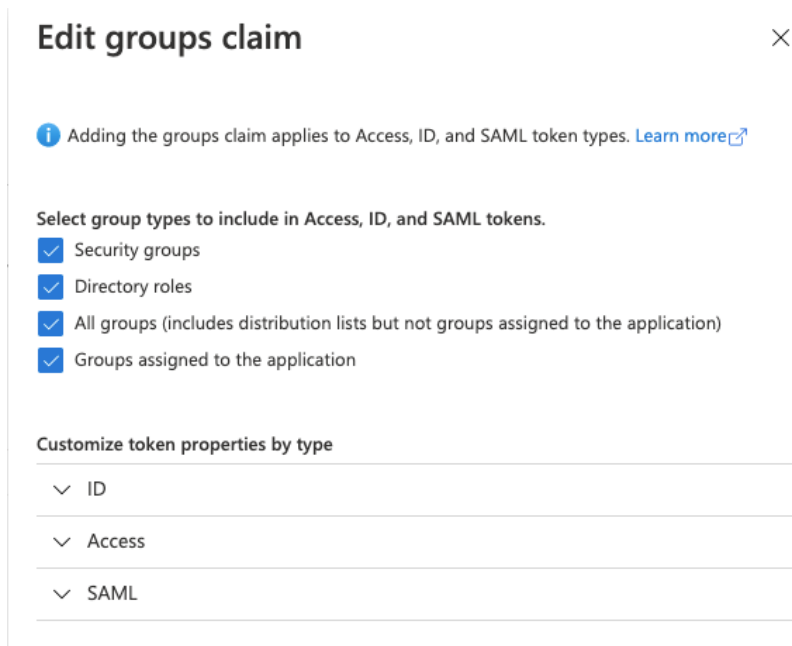
Supported account types

- The key next step is to set up the groups claim in Azure AD. From your new registered application, click on Token configuration:



Click on + Add groups claim.

Configure all of the group types:



9. Token properties should be configured using the sAMAccountName:

Edit groups claim

Customize token properties by type

^ ID

- Group ID
 - sAMAccountName
 - NetBIOSDomain\sAMAccountName
 - DNSDomain\sAMAccountName
 - On Premises Group Security Identifier
 - Emit groups as role claims
-

^ Access

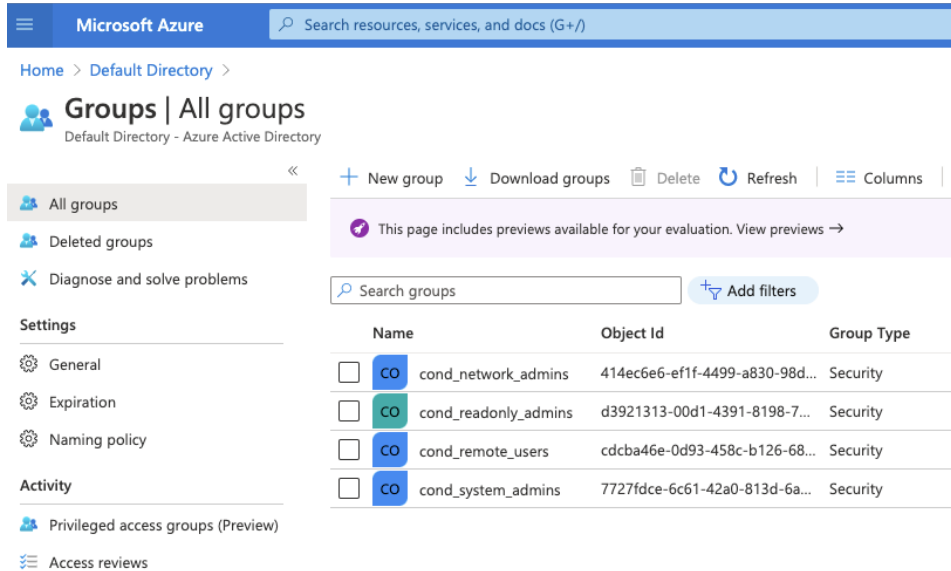
- Group ID
 - sAMAccountName
 - NetBIOSDomain\sAMAccountName
 - DNSDomain\sAMAccountName
 - On Premises Group Security Identifier
 - Emit groups as role claims
-

^ SAML

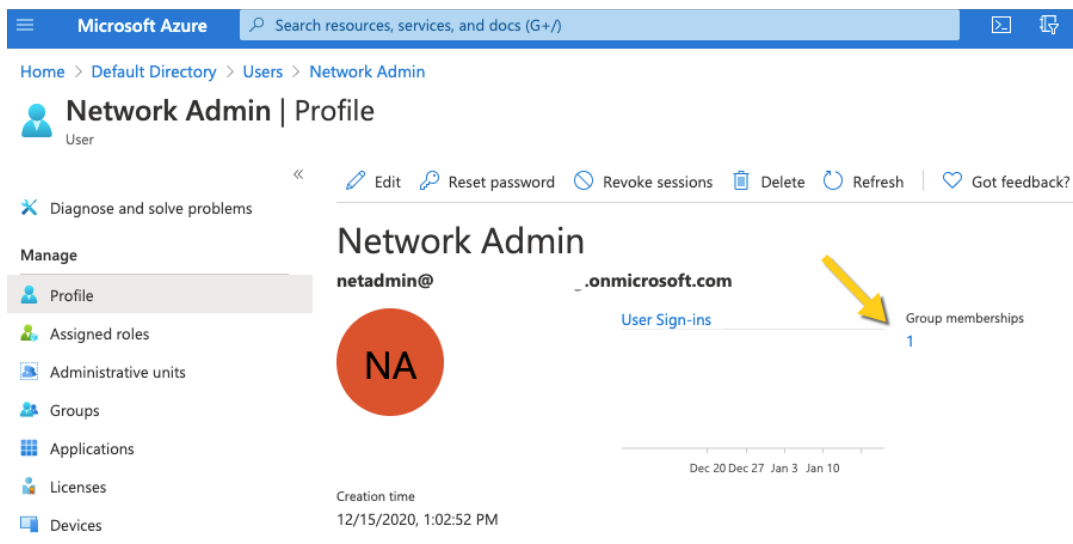
- Group ID
 - sAMAccountName
 - NetBIOSDomain\sAMAccountName
 - DNSDomain\sAMAccountName
 - On Premises Group Security Identifier
 - Emit groups as role claims
-

The SAML setting is not applicable.

10. Create Groups in Azure AD:



11. Add users and assign group membership:



Verify third-party authentication is working

To verify your configuration:

1. Log out of Conductor.
2. Open an incognito window and log in, choosing the provider name you chose in the Conductor.
3. Log in as a user you've set up with third-party provider. You should be able to log in to the Conductor using your third-party provider credentials.

To verify a Airwall Agent can connect:

- After the Airwall Agent logs in using the third-party provider, verify connectivity.

Working example:

People - Rick S
✕

User directory
openid_connect

Role
System Administrator

Status
Active

API access
Disabled

Email

Phone

Alert email trigger level
None

Info

🔖 No tags in use

People groups

People group	Activation code
7727fdce-6c61-42a0-...	None

Non-local users' people group membership must be managed on their authentication provider

Overlay networks

Not a member of any overlay networks

Troubleshooting Third-party Authentication User Login

If user login is failing with “Could not find that username/password combination,” verify:

- The user has been given access to your OIDC application in the third-party provider
- The user is a member of a group in your provider that is mapped to a user role in the Conductor
- The “groups” claim is allowed in your application in the provider
- The user typed in their username and password correctly

Check the Conductor log for additional clues for why the login failed. For instance, you may see a log message that a person does not match any groups to get a role.