2. OAuth Provider Setup

Roost supports various authentication mechanisms as mentioned below

1. Okta
2. Google
3. Microsoft Azure ADFS

OKTA Auth Client Setup

- Sign in to your OKTA account with admin privileges (*If you do not have an existing Okta account, then sign-up at* Home | Okta Developer)
- From the left navigation menu, go to Applications -> Applications.
- Select Create App Integration → OIDC - OpenID Connect → Web Application, then click Next
- Fill in the suitable **App integration name,** upload the logo.
- Add **Sign-in redirect URIs**
  - https://<DNS_NAME>/login
- Allow Access to users thru Assignments → Controlled Access
  - Select the groups of users or Allow access to everyone
- Save and Make a note of the Okta Client ID and the Client Secret (It is needed later in the config below)
- From the left navigation menu, go to Security -> API
- Make a note of **Issuer URI** for default Authorisation Server
  - something like https://{your_domain}.okta.com/oauth2/default

Google Auth Client Setup

- Integrating Google Sign-In into your web app | Google Sign-In for Websites | Google Developers
- Login to https://console.cloud.google.com/apis/credentials
- Create Credentials, Select OAuth Client and Application Type as Web Application
- Add Authorised JavaScript Origin as
  - https://<DNS_NAME>

- Add Authorised redirect URIs

    - https://<DNS_NAME>/login
    - https://<DNS_NAME>/api/auth/redirect/google
- Download the JSON
- Make a note of the Google Client ID and the Client Secret (It is needed later in the config below)

Azure ADFS Auth Client Setup

`Roost 0Auth2 Setup - Windows Server 2016/2019 - ADFS 4.0`

1. Open the **Server Manager** from **Start Menu,** Select **Tools** > **AD FS Management**
2. From the **AD FS Management** screen, go to **AD FS → Application Groups**
3. Click **Add Application Group** on right panel

1. Fill in a **name** (**Roost**) for the application group
2. Select **Server Application Web browser accessing a web API** and click **Next**
3. Make note of the **Client Identifier** value. This will be the value for the `AZURE_ADFS_CLIENT_ID` variable
4. Fill the **Redirect URI** (https://<DNS_NAME>/login) and click Add, then Next
5. Check the **Genrate a shared secret** box
6. Use the **Copy to clipboard** button to retrieve the secret. This will be the value for the `AZURE_ADFS_CLIENT_SECRET` variable. Click **Next**
7. Enter the Web API identifier (Same as RedirectUri - https://<DNS_NAME>/login) and click A**dd,** then **Next**
8. On the **Access Control Policy** screen, select a policy, usually **Permit everyone** and click **Next**
9. On the **Configure Application Permissions** screen, select the scope **openid** and click **Next**
10. **Review the settings and click Next**
11. Close the wizard by clicking **Close**. Our application is now registered in ADFS.

1. Now, we need to **Configure the Claims** for Application

1. Open the P**roperties** for the application group we just created.
2. Select the **Web application** entry (**Roost - Web API**) and click **Edit**
3. On the **Issuance Transform Rules** tab, click the **Add Rule** button
4. Select **Send LDAP Attributes as Claims** and click **Next**
5. Give the rule a name (**Roost Claims**) and select **Active Directory** as the attribute store.
6. Now configure the below claims (**LDAP Attribute => Outgoing Claim Type**):

1. E-Mail-Addresses => E-Mail Address
2. Given-Name => Given Name
3. Surname => Surname

4. SAM-Account-Name => Windows Account Name
5. User-Principal-Name => UPN

1. Click **Finish** to save the claims
2. You should now see the rule added. Click **OK** a couple of times to save the settings.

1. Now the setup is complete. We set these 3 values as environment variables:

1. `AZURE_ADFS_CLIENT_ISSUER` - Domain of ADFS Server (https://adfs.contoso.com)
2. `AZURE_ADFS_CLIENT_ID` - Client Identifier of server application
3. `AZURE_ADFS_CLIENT_SECRET` - Client Secret we copied to clipboard

If don’t want to use Client Secret, then Add an Native Application and pass `AZURE_ADFS_CLIENT_SECRET` variable as empty

---

Revision #3
Created 15 March 2023 19:21:15 by Rakesh
Updated 5 September 2025 06:52:08 by Harish