SSO Learning Roadmap (Hands-On)

## **Phase 1: Core Concepts (Day 1–2)**

### **What You’ll Learn:**

* What is SSO and why it’s used
* Identity Provider (IdP) vs Service Provider (SP)
* Key protocols: SAML, OAuth2, OpenID Connect (OIDC)
* Tokens: SAML assertions and JWTs

### **Resources:**

* [Microsoft Learn - Intro to Azure AD](https://learn.microsoft.com/en-us/training/modules/introduction-azure-active-directory/)
* Okta - Intro to SSO
* Auth0 - Identity Fundamentals

## **Phase 2: Labs and Real Scenarios (Day 3–7)**

You'll simulate SSO between Azure AD (as your IdP) and various SPs.

### **✅ Lab 1: SAML SSO with Azure AD and SP**

#### **Tools:**

* Azure AD (via [M365 Developer Program](https://developer.microsoft.com/en-us/microsoft-365/dev-program))
* A public Service Provider like sptest.iamshowcase.com

#### **Steps:**

1. **Create a free Azure AD tenant**
   * Join the M365 Developer Program and set up a sandbox tenant.
2. **Create a new Enterprise Application** in Azure AD  
   * Go to **Azure Portal > Azure Active Directory > Enterprise Applications > New Application**
   * Select **“Non-gallery application”**
   * Name it SP-Test
3. **Get SP metadata from sptest.iamshowcase.com**
   * Visit: https://sptest.iamshowcase.com/instructions
   * Note:  
     + **Entity ID**: https://sptest.iamshowcase.com/metadata
     + **ACS URL**: https://sptest.iamshowcase.com/acs
4. **Configure Azure AD SAML SSO**
   * In your app > **Single Sign-On > SAML**
   * Fill in:  
     + **Identifier (Entity ID)**: https://sptest.iamshowcase.com/metadata
     + **Reply URL (ACS)**: https://sptest.iamshowcase.com/acs
   * Save the configuration
5. **Download Azure AD Federation Metadata**
   * Still in the SAML page, download **Federation Metadata XML**
6. **Upload Azure AD Metadata to the SP**
   * Go back to https://sptest.iamshowcase.com/instructions
   * Upload your XML file (from step 5)
7. **Test SSO**
   * Use the login URL generated by the SP
   * Sign in using a user from your Azure tenant

### **⭐POST LAB 1 NOTES Summary**

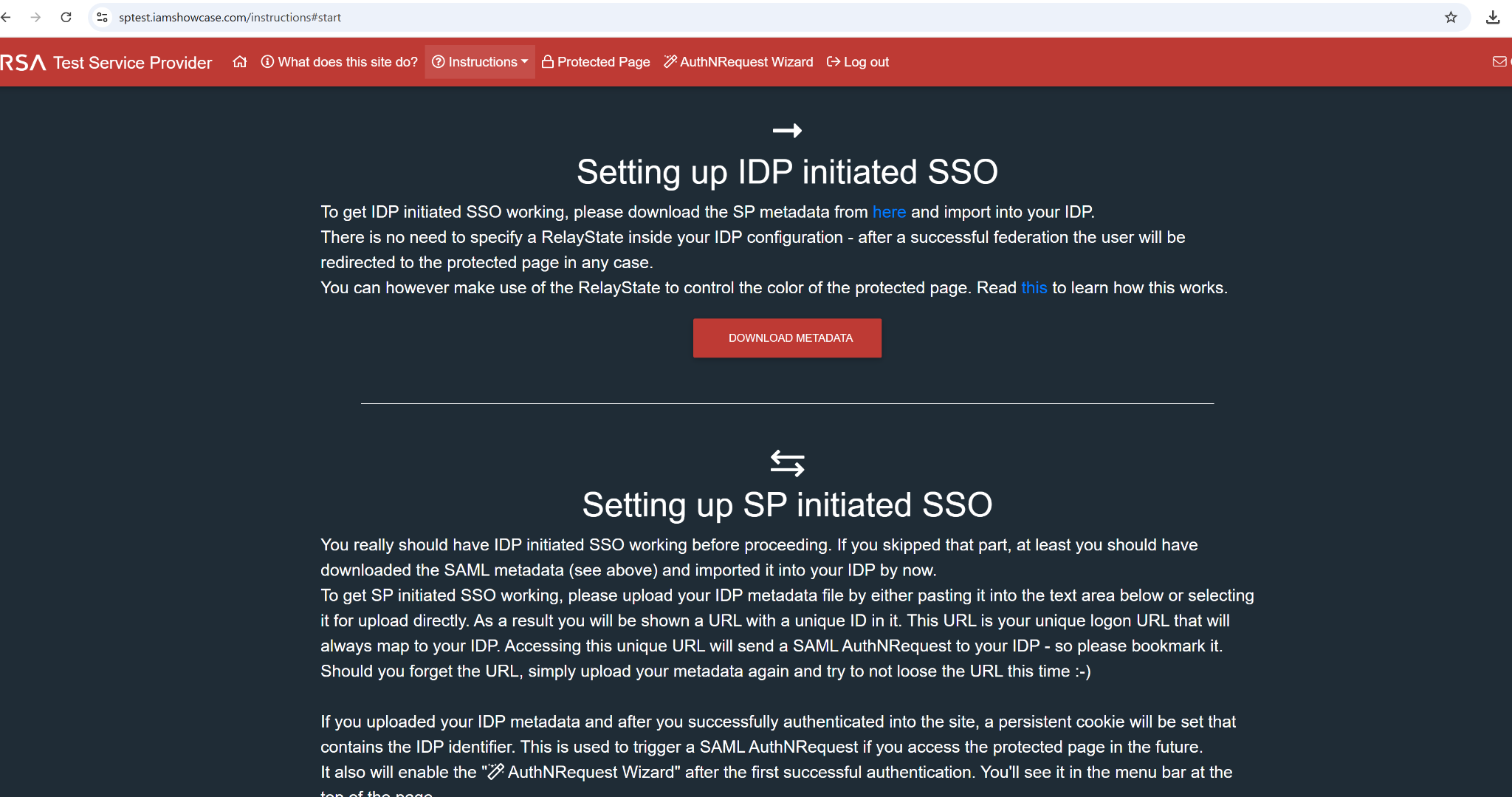
To create SAML SSO in a third party SaaS website I:

1. Created a new Enterprise Application in my AAD tenant
2. Click the Set Up Single Sign On option
3. Filled out mandatory fields in Basic SAML configuration field

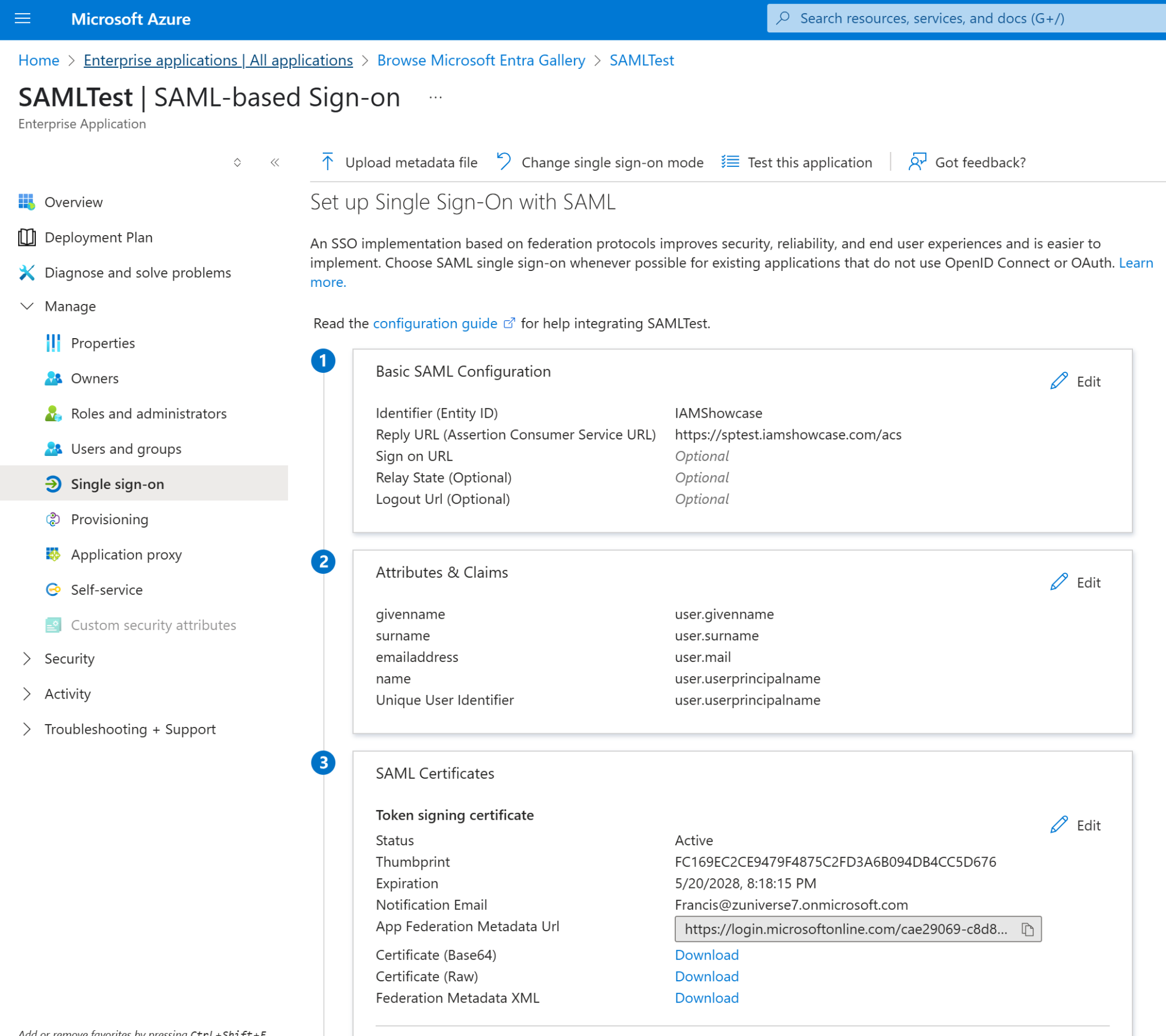
-Identifier (Entity ID)

-Reply URL (Assertion Consumer Service URL)

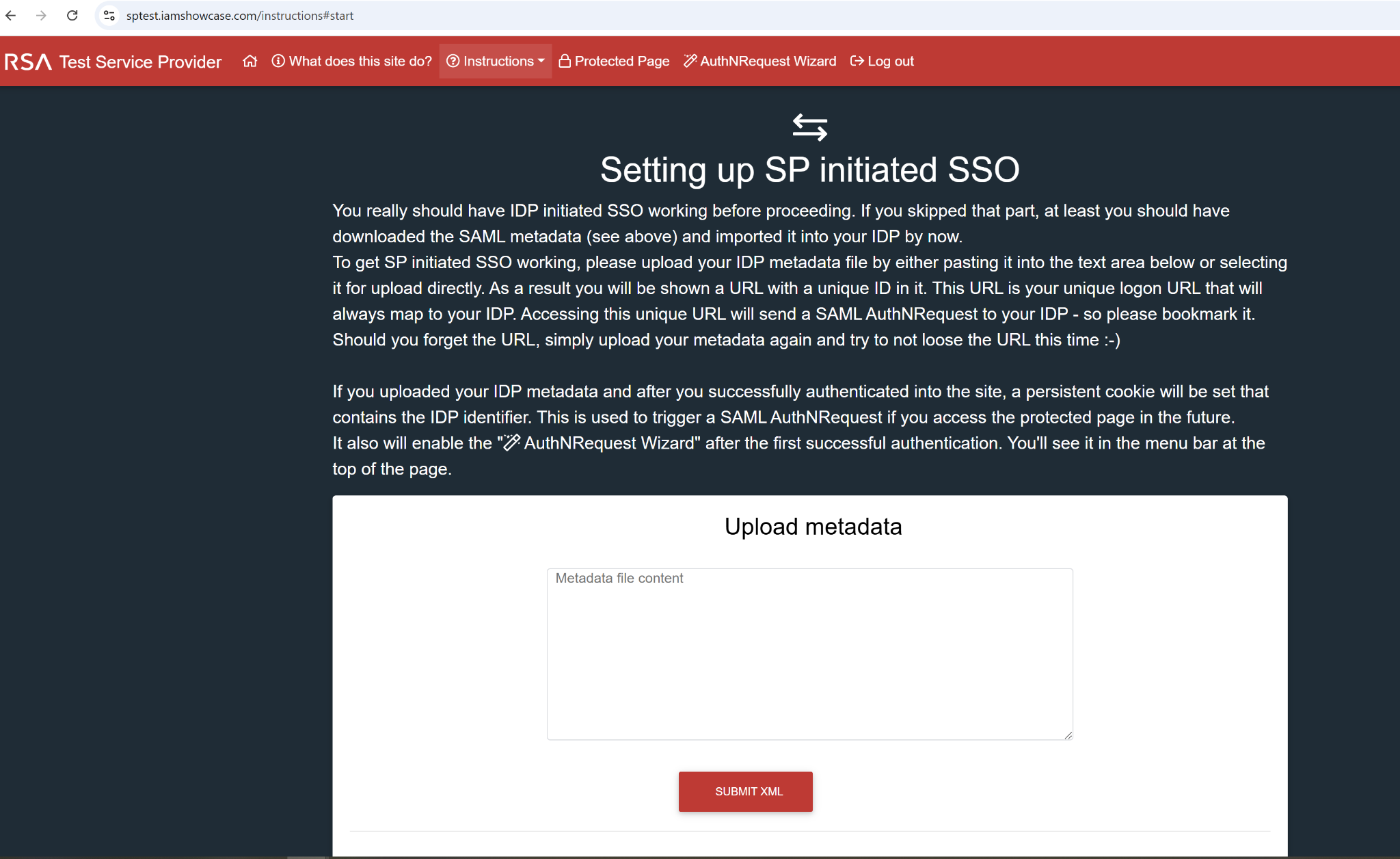
1. To get this information I went to my 3rd party site(https://sptest.iamshowcase.com/), and went to a section that said ‘download metadata’. I found the values in that file



1. Once my AAD Enterprise Application has the correct metadata values for the 3rd party site, I need to provide the 3rd party site with a SAML Certification. This will let AAD know that this site is trusted/federated and will allow the redirect to Azure for SSO

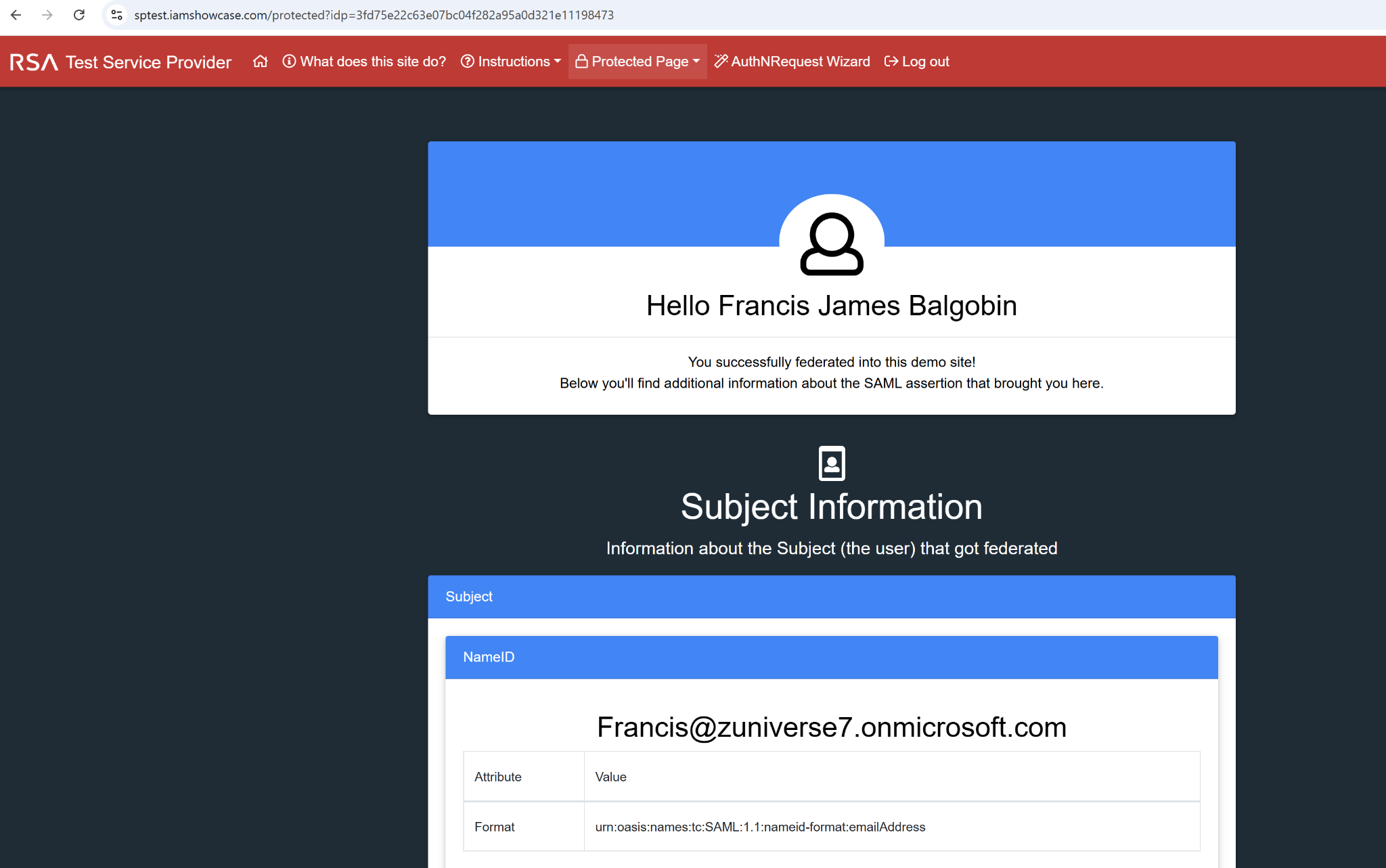


1. To get the SAML certification from AAD, in the Enterprise Application GO to Single Sign On menu, then under SAML Certificates click Download on Federation MetadataXML
2. Once you have that file, copy the XML data, go to the 3rd party site and Paste it n the Upload Metadata field



1. This will generate a Unique URL. This URL is your unique logon URL that will always map to your IDP.

<https://sptest.iamshowcase.com/ixs?idp=3fd75e22c63e07bc04f282a95a0d321e11198473>

Use the URL and it should re-direct to AAD for Authentication 

### 

### **✅** Lab 2: OIDC SSO - Use Azure AD to Sign Into Grafana Cloud via OIDC

#### What You Need:

* Azure AD access to register apps
* A free Grafana Cloud account →<https://grafana.com>
* Any Azure AD user to test login

Step 1: Sign up for Grafana Cloud

1. Go to<https://grafana.com>
2. Create a **Free account**
3. Set up a new **Grafana Cloud stack** (hosted instance)
4. Once it’s deployed, open your **Grafana URL** (e.g., your-stack.grafana.net)

#### 

#### 

#### 

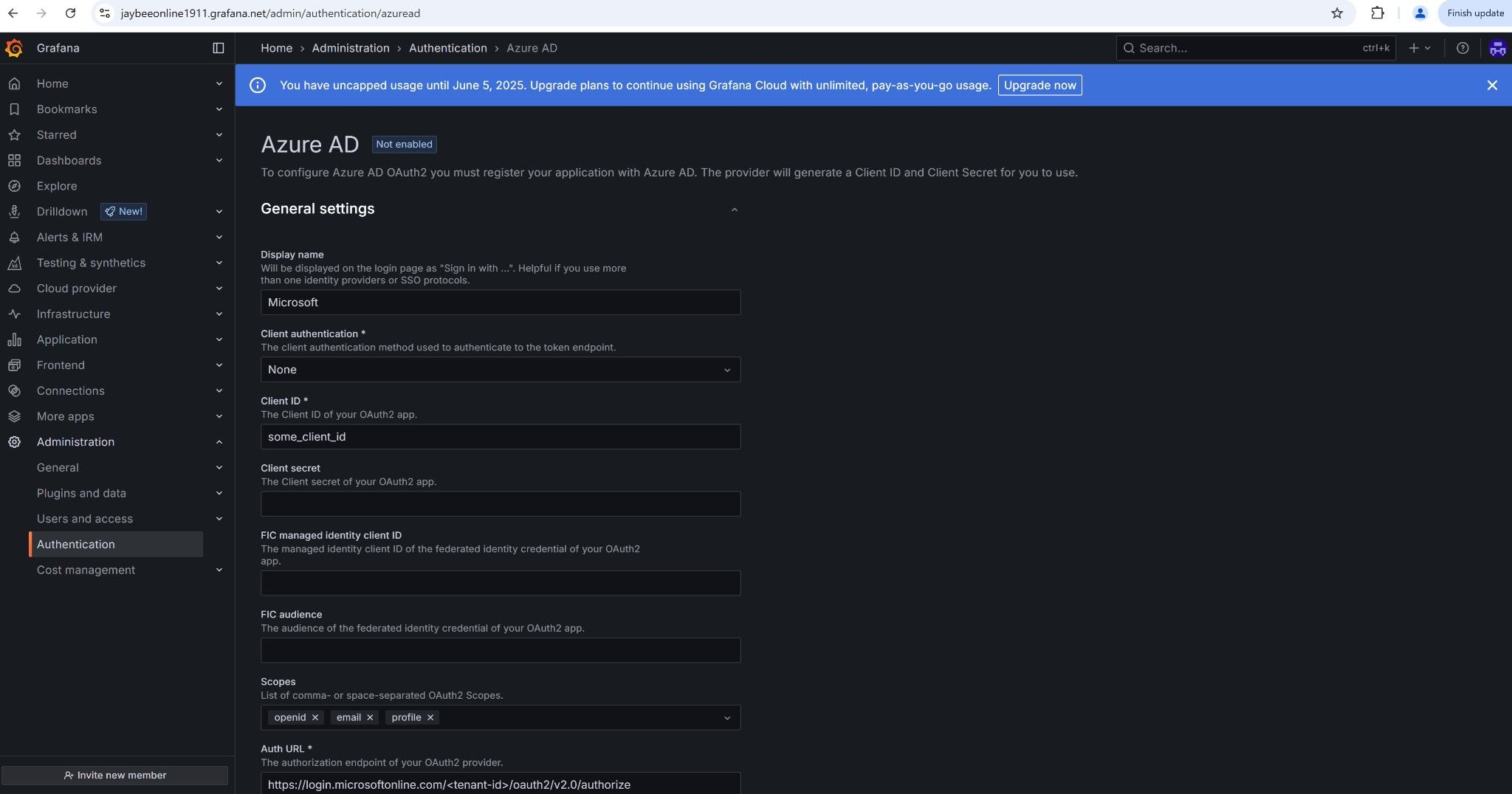
#### 

#### **Step 2: Enable Azure AD Login in Grafana**

In the Grafana dashboard:

1. Go to **Administration > Authentication**
2. Choose **OpenID Connect**
3. You’ll be asked for:  
   * **Client ID**
   * **Client Secret**
   * **Issuer URL**
   * **Scopes**
   * **Redirect URI**

Leave this page open — you’ll fill it out after registering the app.

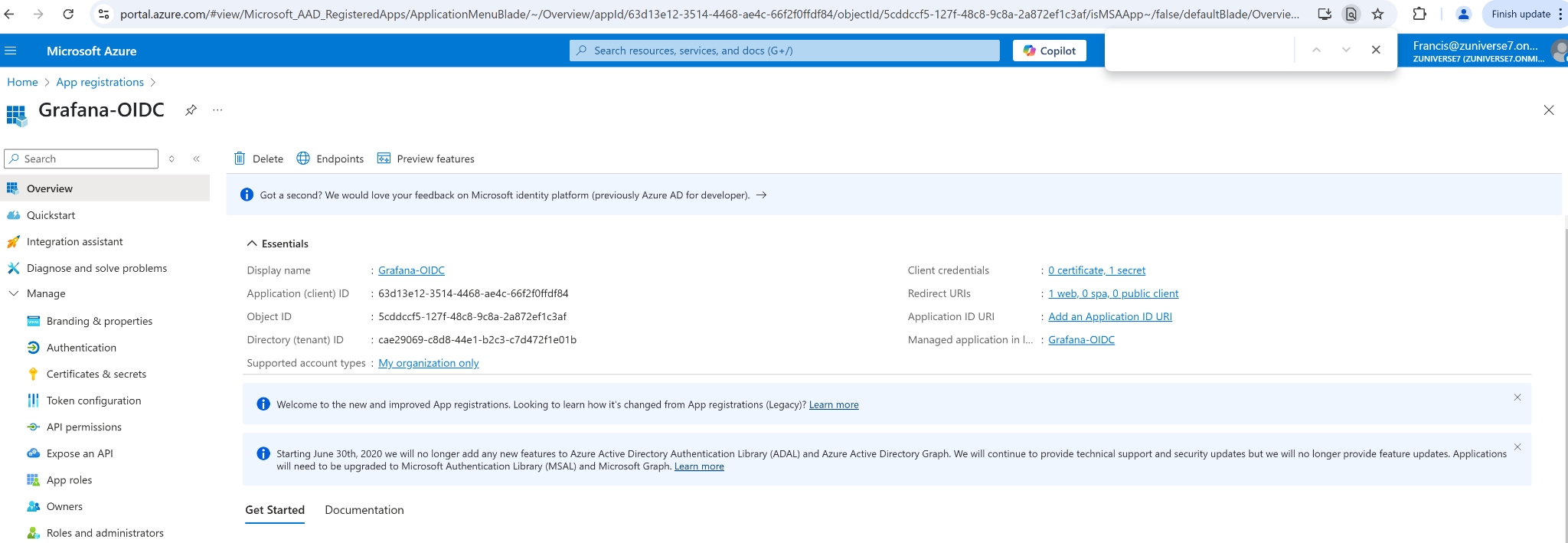


#### Step 3: Register App in Azure AD

1. Go to **Azure > App registrations > New registration**
2. Name: Grafana-OIDC

Redirect URI:  
  
 arduino  
CopyEdit  
https://<your-grafana-stack>.[grafana.net/login/](http://grafana.net/login/generic_oauth)azuread

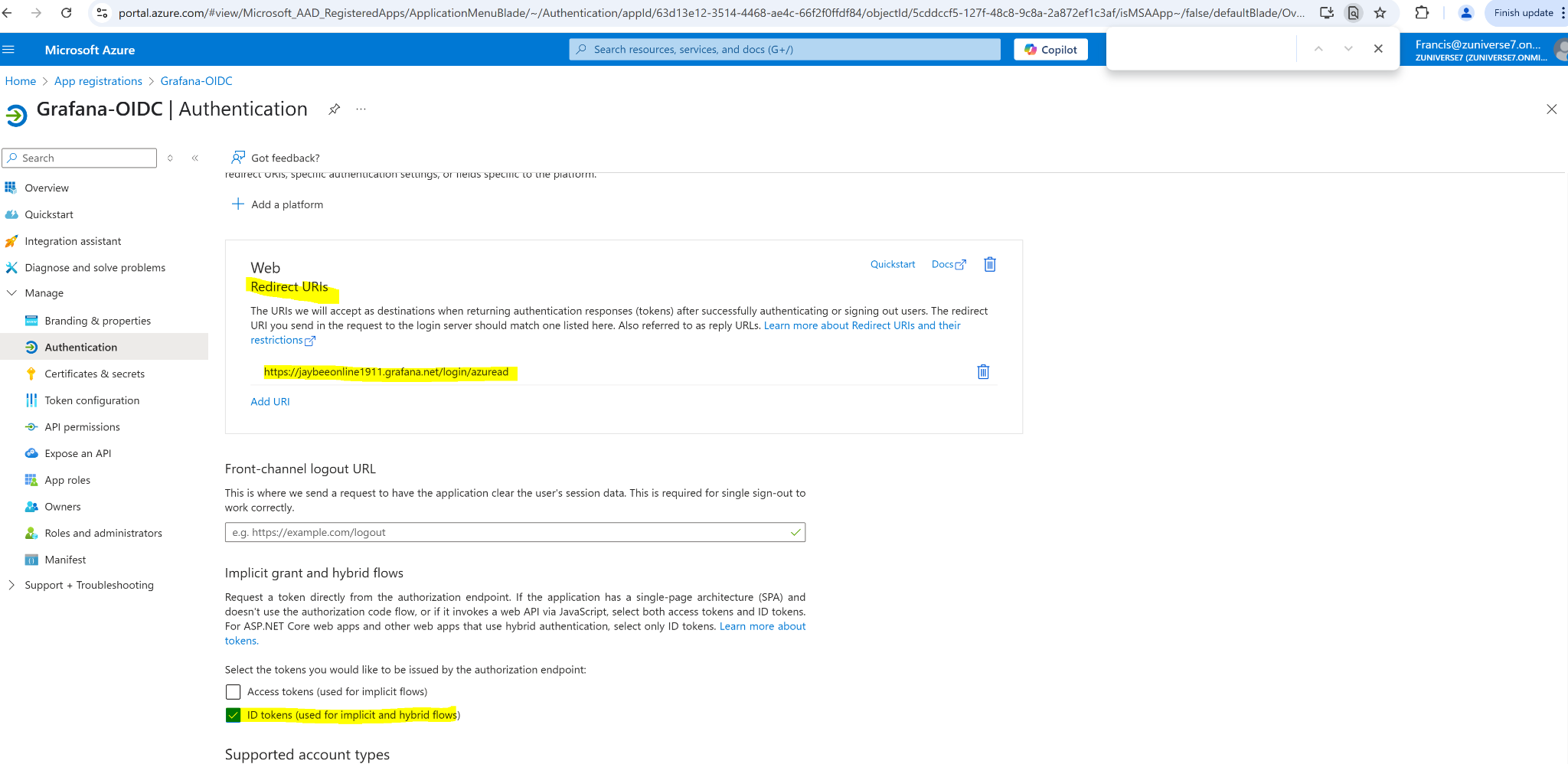
1. Register it.



#### Step 4: Configure Authentication

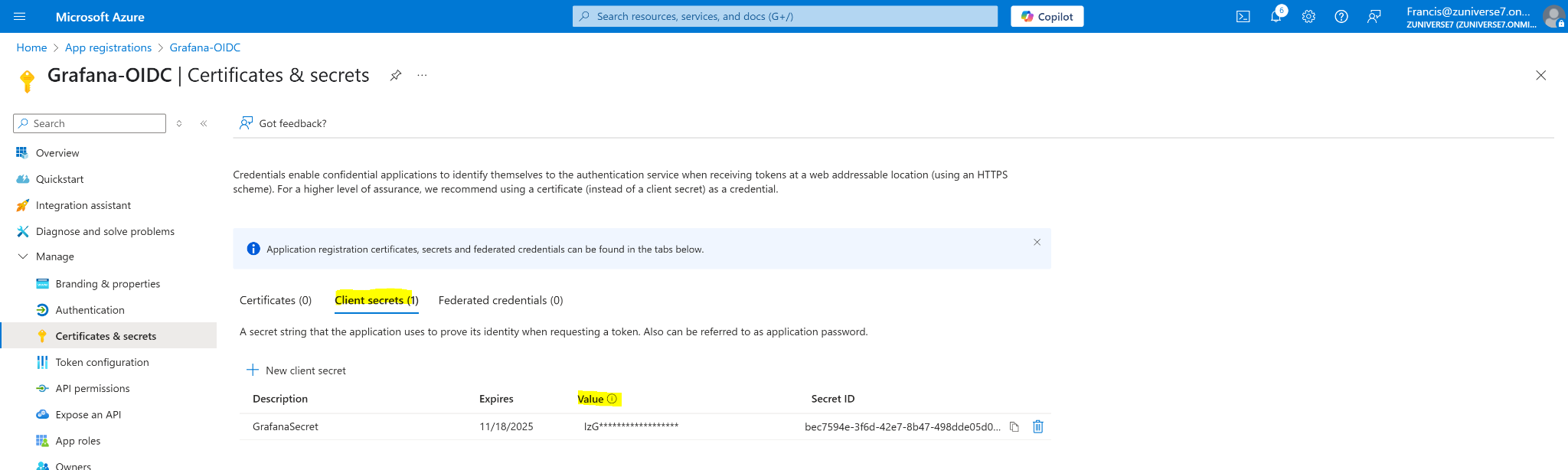
In the App Registration:

1. Go to **Authentication**
2. Check **ID Tokens**
3. Add the same **Redirect URI** again to ensure it's saved



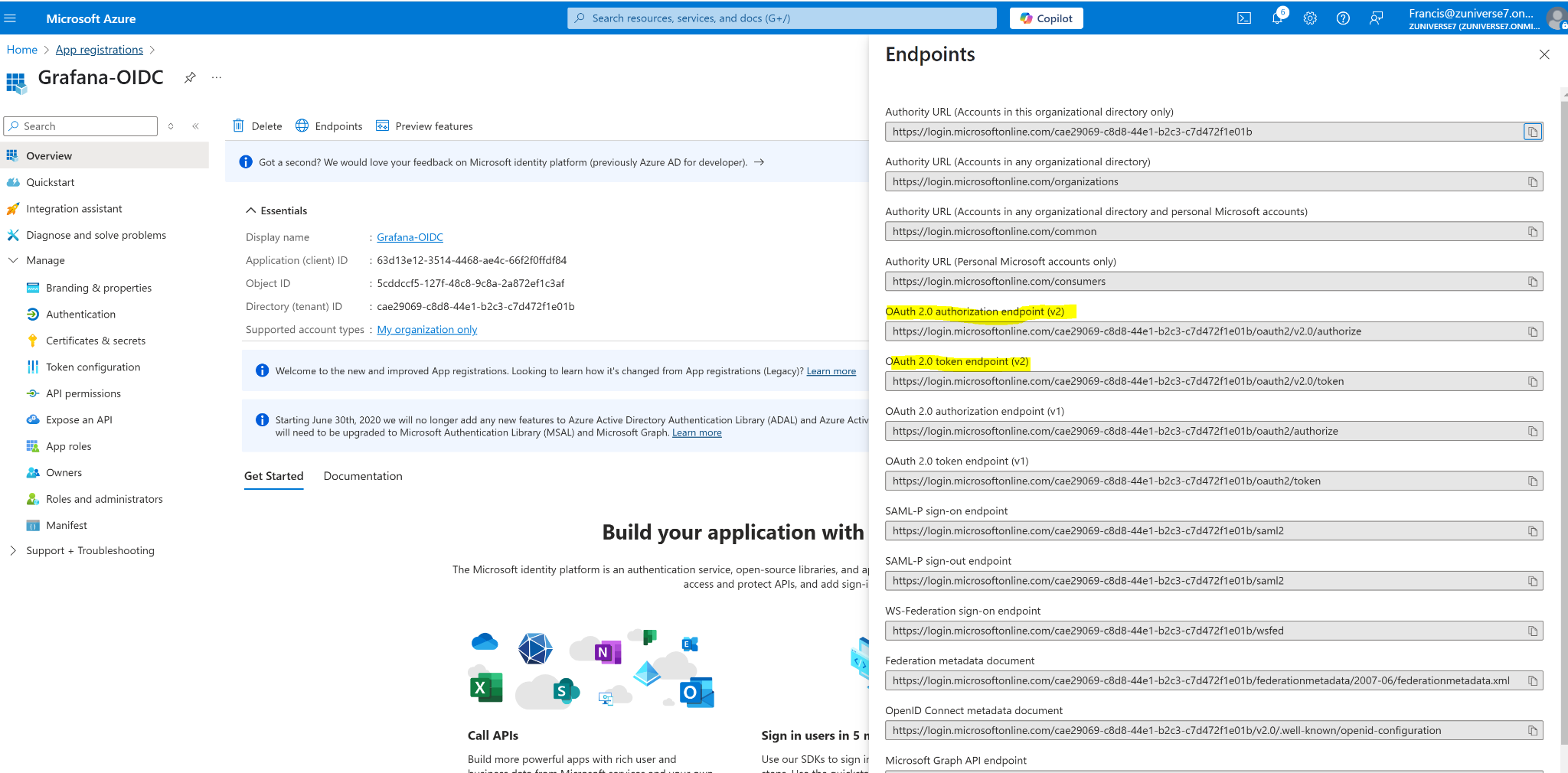
#### Step 5: Create Client Secret

1. Go to **Certificates & Secrets**
2. Click **New Client Secret** → copy the secret value



#### Step 6: Get Azure Endpoints

Go to the App → **Overview → Endpoints** Copy these:

* **Issuer (v2)** → used for “Issuer URL” in Grafana
* **Authorization endpoint**
* **Token endpoint**

#### Step 7: Paste into Grafana

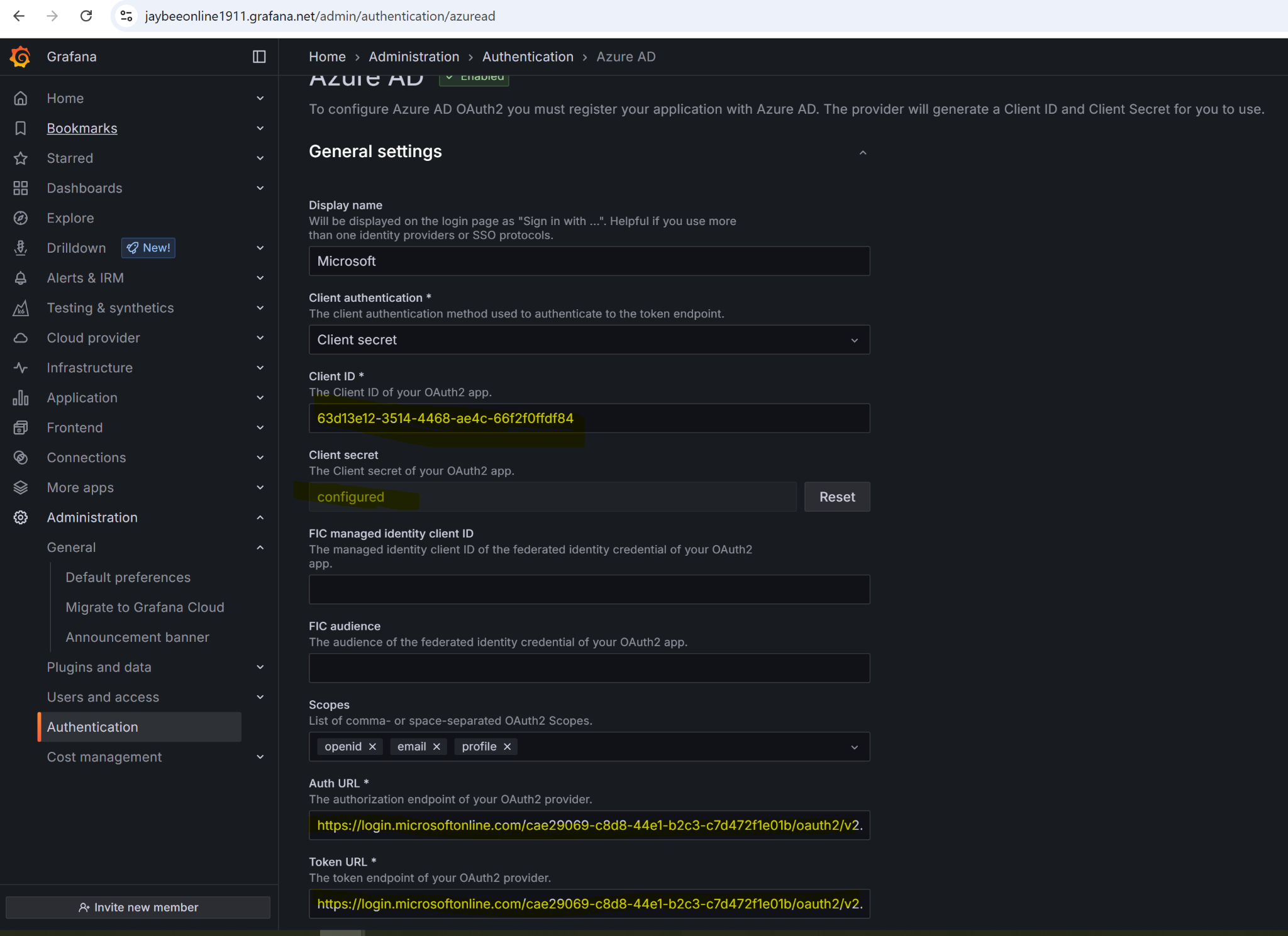
Back in your Grafana dashboard:

* Client ID → from Azure
* Client Secret → from Azure
* Issuer URL → from Azure (e.g., https://login.microsoftonline.com/<tenant>/v2.0)

Scopes:  
  
 nginx  
CopyEdit  
openid profile email

* Redirect URI → should already be auto-filled

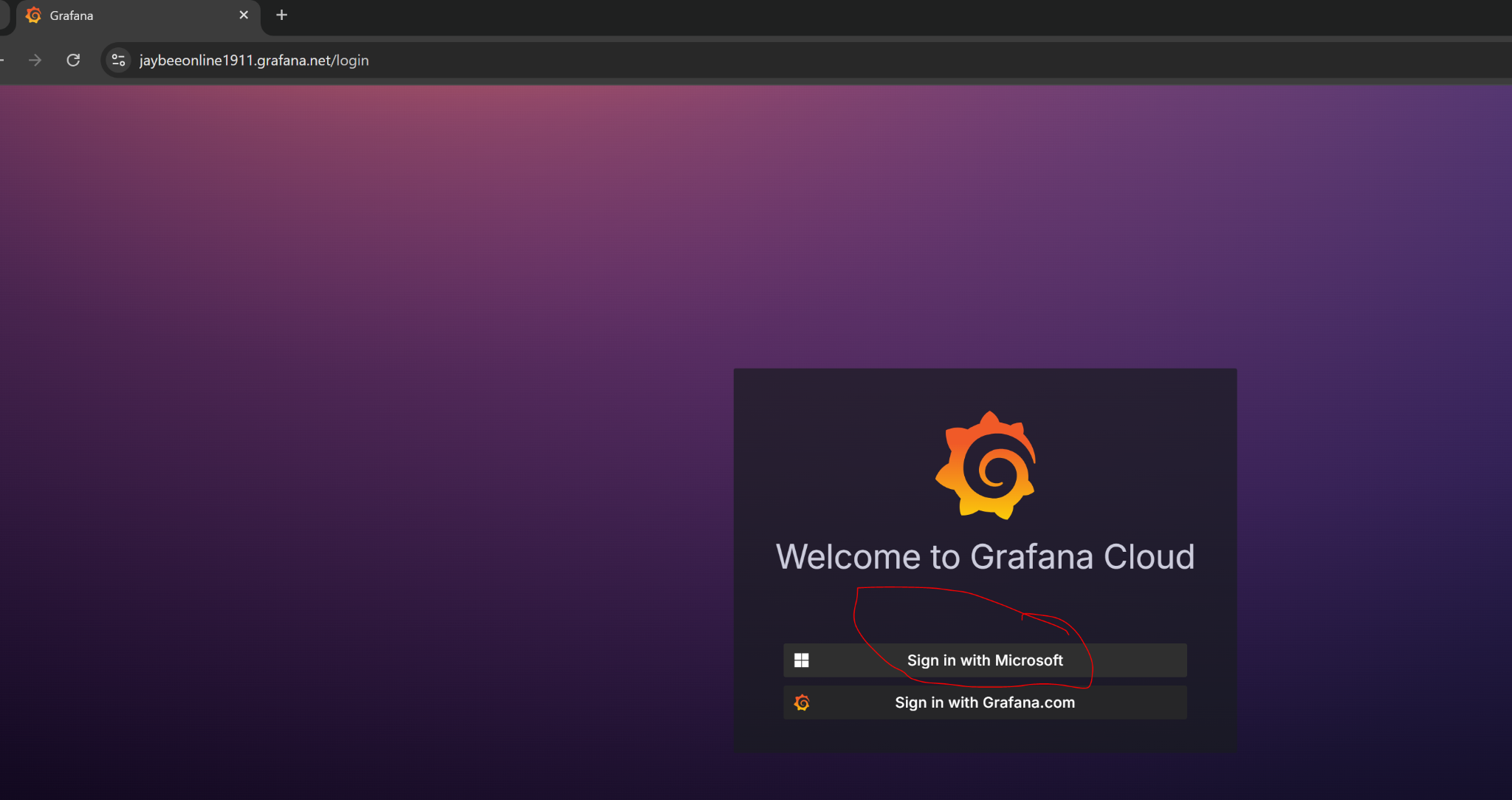
Save.



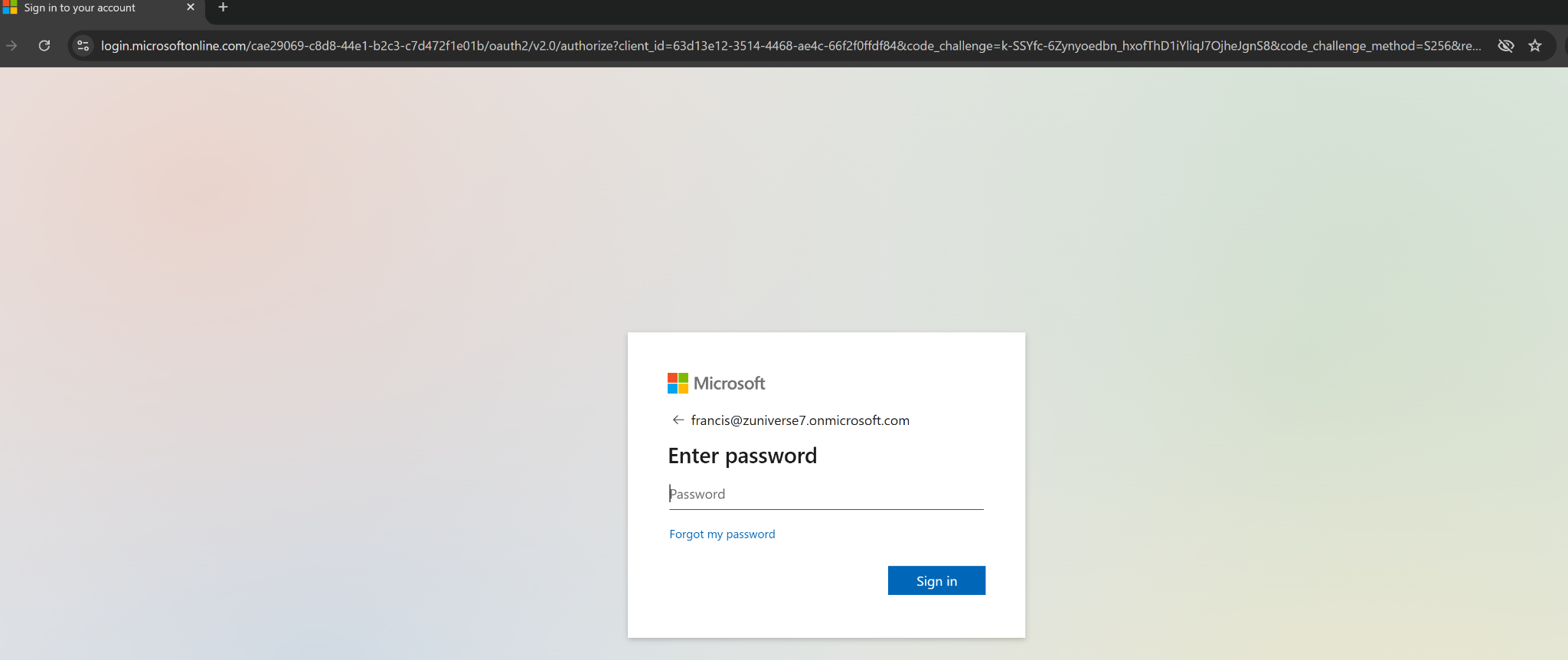
#### Step 8: Test Login

Go to your Grafana login page and choose **Azure AD login**.

🎉 If everything's configured correctly, you'll be redirected to Microsoft, and once logged in, sent back to Grafana as an authenticated user.

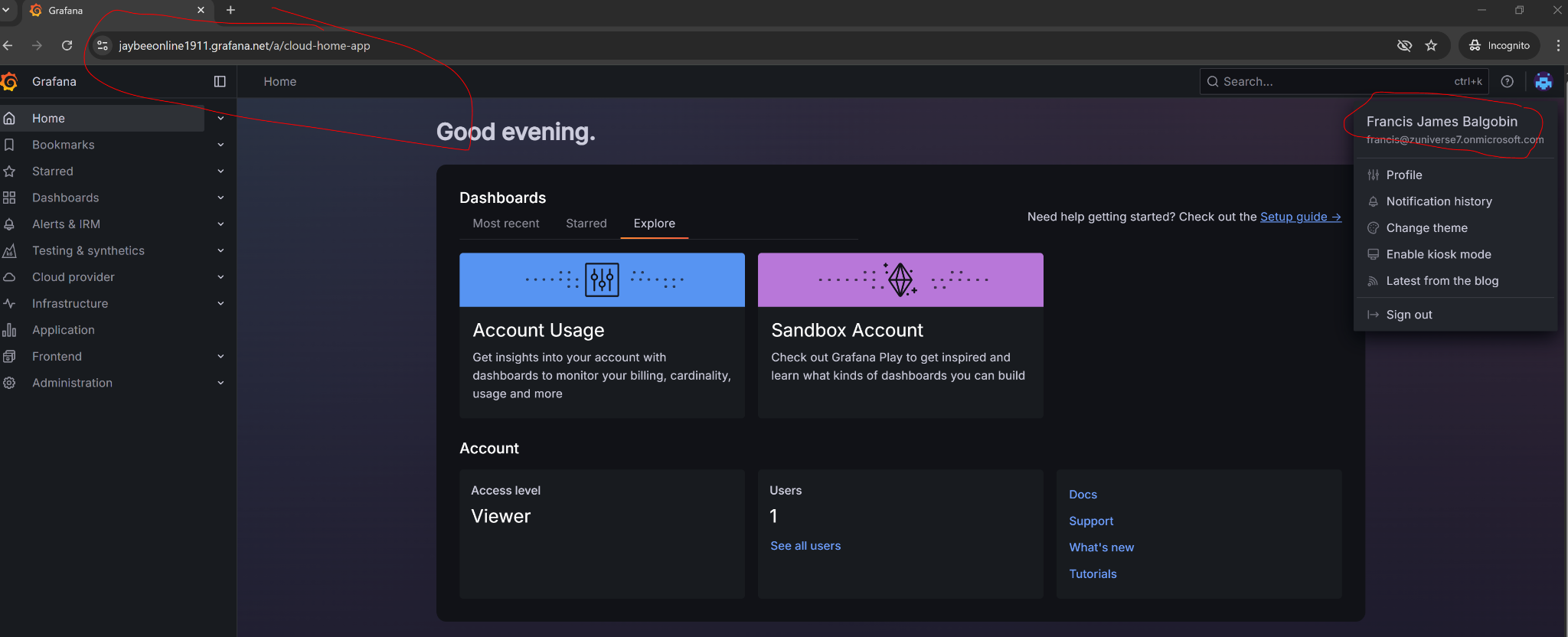


Once Microsoft login is selected, redirect to MS login page



Enter credentials +MFA of an AAD user in tenant

Successful login using an AAD user below



### High-Level OIDC SSO Flow (Once Configured)

1. User visits the website

You go to a protected app like:  
  
https://your-stack.grafana.net

* The app sees you're not authenticated and redirects you to the Azure AD authorization URL.

2. User is redirected to Azure AD

The app builds a redirect to Azure’s authorization endpoint:  
  
https://login.microsoftonline.com/<tenant-id>/oauth2/v2.0/authorize

* It includes query parameters like:  
  + client\_id → the app registration ID
  + redirect\_uri → where Azure should send the user after login
  + scope → openid profile email
  + response\_type → code (in authorization code flow)

3. User logs in to Azure AD

* Azure AD presents the login screen.
* After successful authentication, Azure generates an authorization code.

4. User is redirected back to the app

Redirect URI:  
  
 arduino  
CopyEdit  
https://your-stack.grafana.net/login/generic\_oauth?code=xyz...

* The app receives the code on this route.

5. App exchanges the code for tokens

Grafana (or your app) sends the code + client\_id + client\_secret to Azure’s token endpoint:  
  
 bash  
CopyEdit  
https://login.microsoftonline.com/<tenant-id>/oauth2/v2.0/token

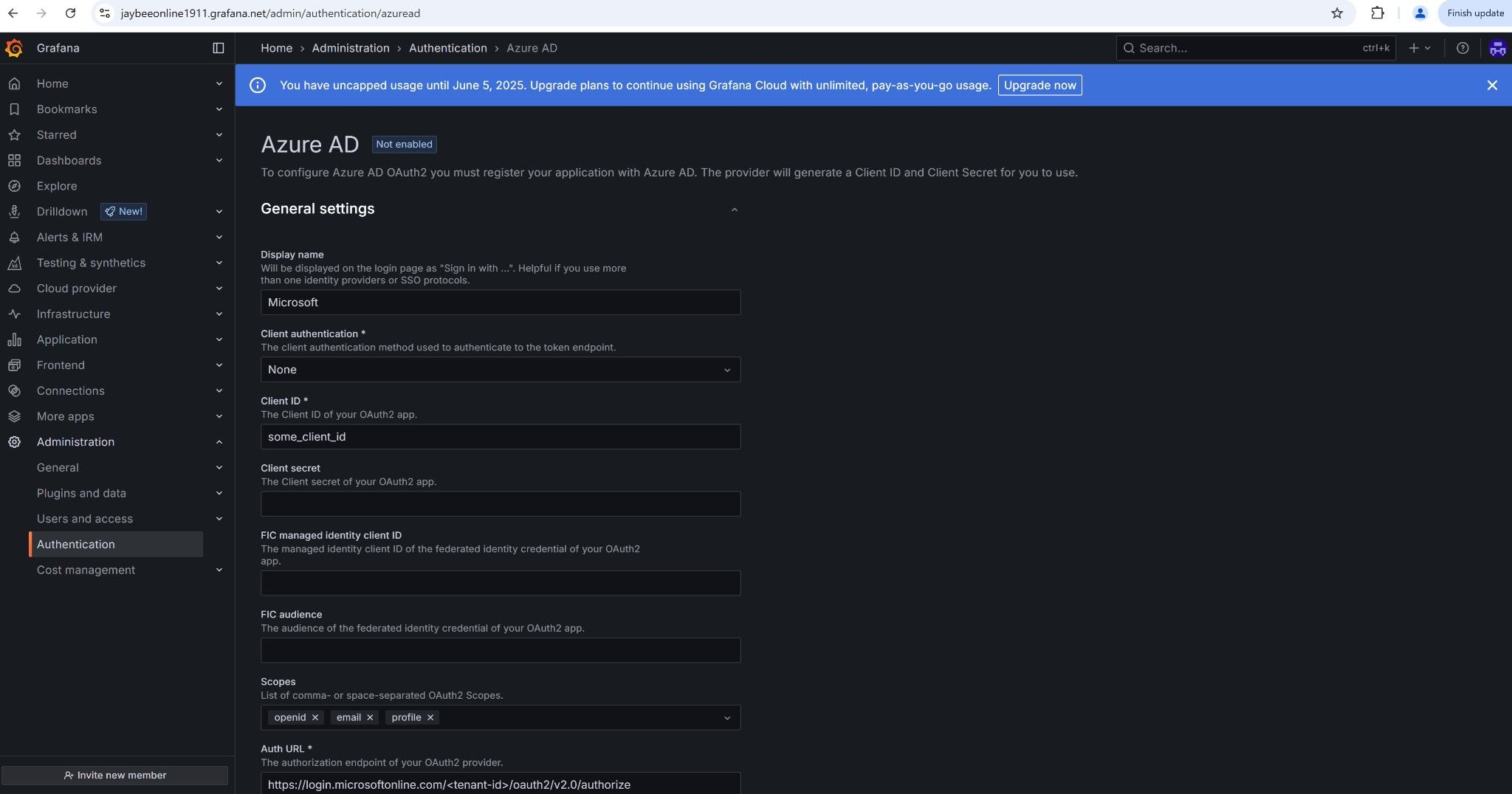
6. Azure returns an ID token (JWT)

* The ID token contains the user’s identity info.
* Grafana parses it, checks the signature, and logs the user in.

### 

### **⭐POST LAB 2 NOTES Summary**

1. Created SSO into a 3rd party site using OAuth/OIDC.
2. Registered for a 3rd paty site instance
3. Went to Authentication options in 3rd party site, located Azure AD option



1. Went to my Azure AD tenant and registered a new App Registration
2. Only thing defined when creating was the Name and Redirect URI
3. We create a client secret, and get the other required information the 3rd party site needs for the Azure AD auth to be enabled

### **Lab 3: Real App Integration**

Choose an open-source or self-hosted app like:

* Jenkins (with SAML plugin)
* WordPress (with SAML plugin)

Use Azure AD SSO to authenticate users into the app.

### **Bonus: SCIM Provisioning**

Once you get SSO working, add user provisioning:

* Enable **automatic user provisioning** from Azure AD to SP
* Use SCIM (if supported by app)
* Watch Azure auto-create and deactivate accounts