polaris

Polaris OAuth 2.0 Integration with Microsoft AD FS Guide

7.2

innovative
Part of Clarivate

World Headquarters
1900 Powell Street, Suite 400
Emeryville, CA 94608
+1.510.655.6200

Legal Notices

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

Contents

Introduction

Polaris System Administration (web-based) requires OAuth 2.0 with OpenID and PKCE. As of version 7.2, Leap supports using OAuth 2.0 with OpenID.

When configured, staff authentication for Polaris System Administration (web-based) and Leap is handled by Active Directory and Active Directory Federation Services.

Important:
The mechanism used to connect an Active Directory user to a Polaris user is the user principal name (UPN) in the format of an email address. For example, user@mydomain.com. During the account verification process, we request the UPN claim from Active Directory. This must return a UPN in the name@domain format. The Polaris.AdminServices (API) can then use that information to map the AD user to a Polaris user.

Minimum Requirements

Polaris System Administration (Web-Based)

To use Polaris System Administration (web-based), you must have the following installed:

- Windows Server 2019 Standard
  - Polaris requires OAuth 2.0 with PKCE support
  - AD FS on Windows Server 2019 supports PKCE
- Active Directory Domain Services
- SSL Certificate
  - Publicly trusted CA signed certificate
- Polaris 7.1

Leap with Active Directory and AD FS Authentication

To use Leap with Active Directory and AD FS authentication, you must have the following installed:

- Windows Server 2019 Standard
  - Polaris requires OAuth 2.0 with PKCE support
  - AD FS on Windows Server 2019 supports PKCE
- Active Directory Domain Services
- SSL Certificate
  - Publicly trusted CA signed certificate
- Polaris 7.2

Process Overview

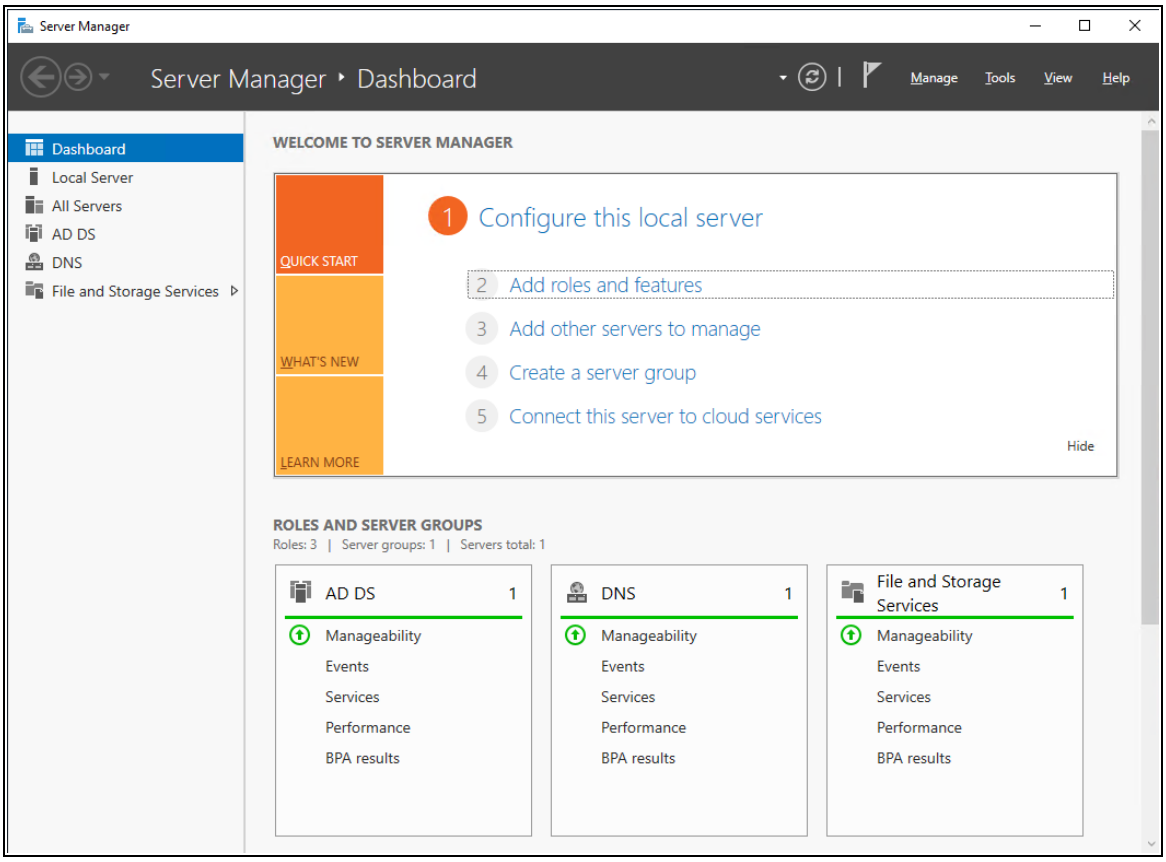
To configure Polaris OAuth Support with AD FS, perform the following tasks:

1. Install Active Directory Federation Services.
2. Configure Active Directory Federation Services.
3. Verify that Active Directory Federation Services is running.
4. Verify that OAuth 2.0 is Enabled.
5. Create an Application Group for Polaris LeapWebApp.
6. Configure the AD FS Web Application: Claims and Permitted Scopes.
7. Enable CORS on AD FS to accept requests from Polaris APIs.
8. Set up web services and applications.
9. Enable session storage for LeapWebApp.
10. Add a URL rewrite rule for LeapWebApp.
11. Customize the AD FS pages.
12. Change the access token lifetime.
13. Bind a new SSL certificate.
14. Troubleshoot.

Install Active Directory Federation Services

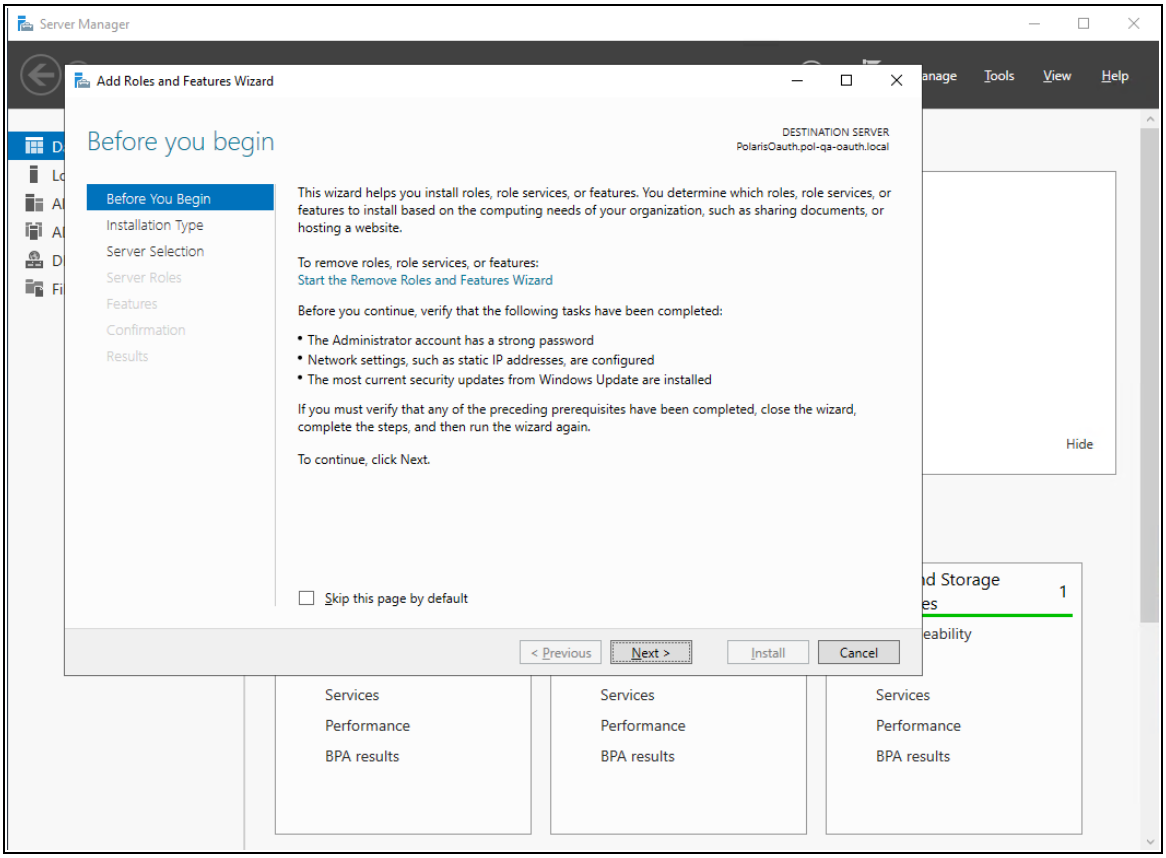
To install AD FS

1. Sign in to Windows Server 2019 with administrative privileges.
2. Start the Server Manager desktop application.

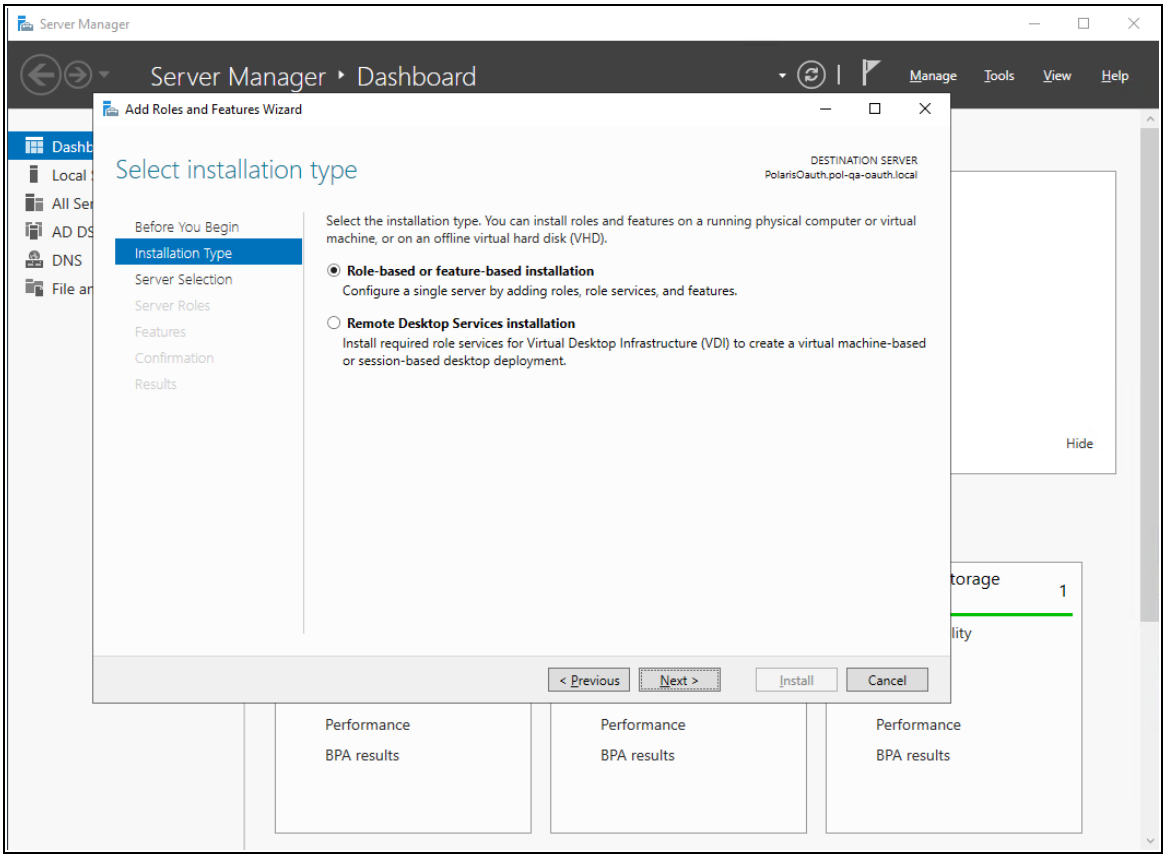


3. On the **Server Manager Dashboard** view, select **Add roles and features**.

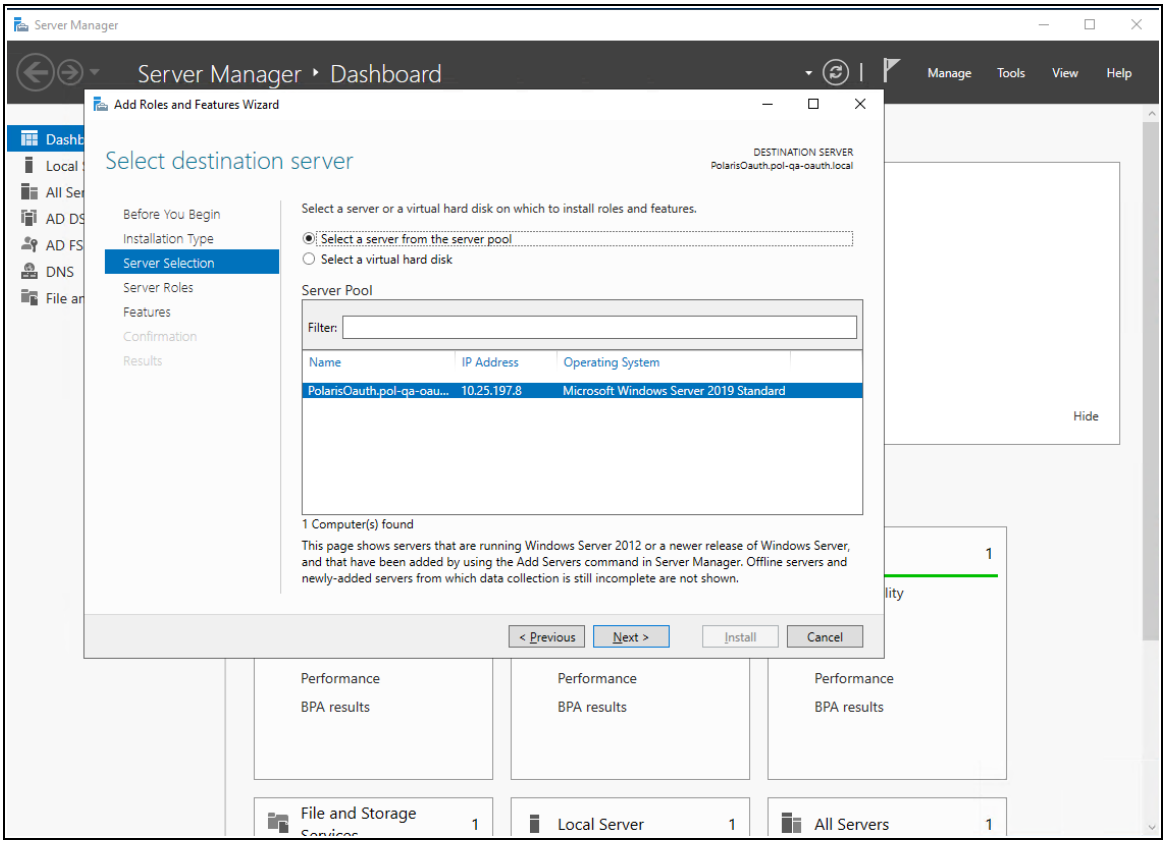
   The Add Roles and Features Wizard opens.



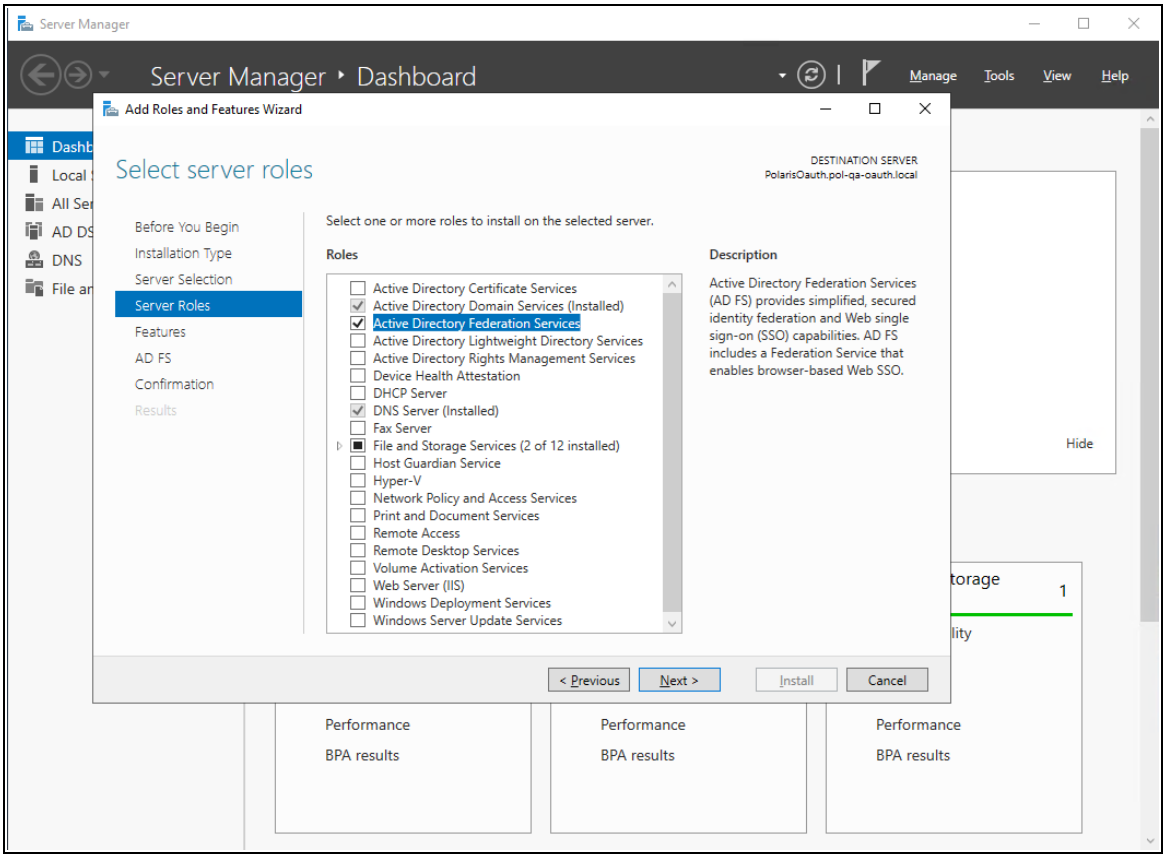
4. On the **Before You Begin** tab, select **Next**.



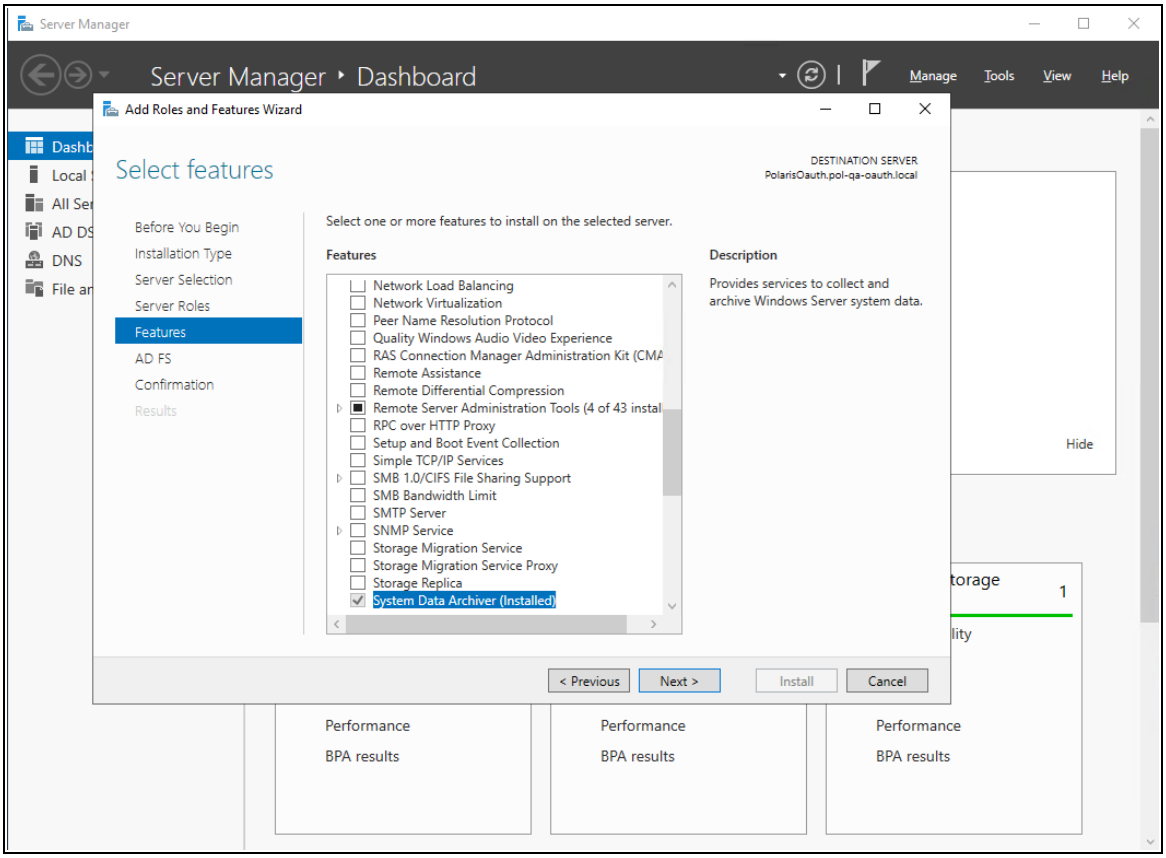
5. On the **Installation Type** tab, select **Role-based or feature-based installation**, and then select **Next**.



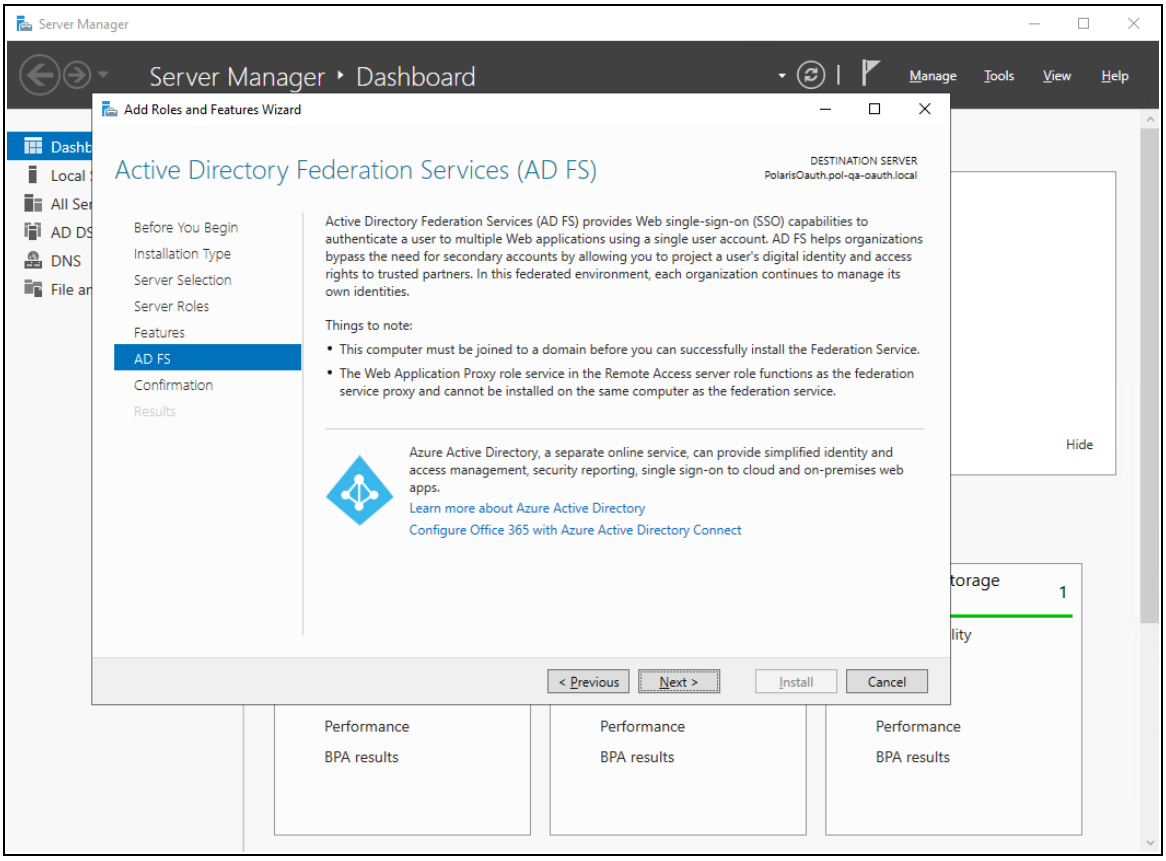
6. On the **Server Selection** tab, select the server, and then select **Next**.



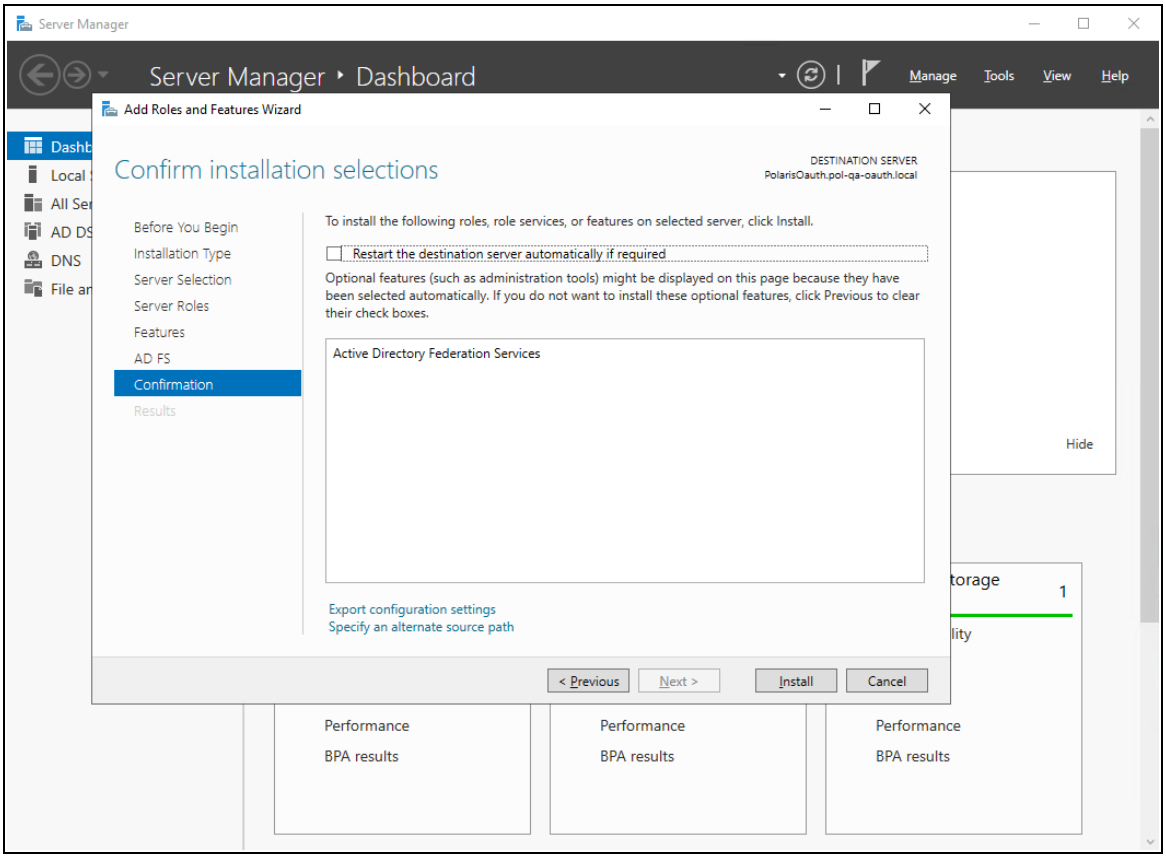
7. On the **Server Roles** tab, do the following:
    a. Verify that **Active Directory Domain Services** are installed.
    b. Select the **Active Directory Federation Services** role.
    c. Select **Next**.



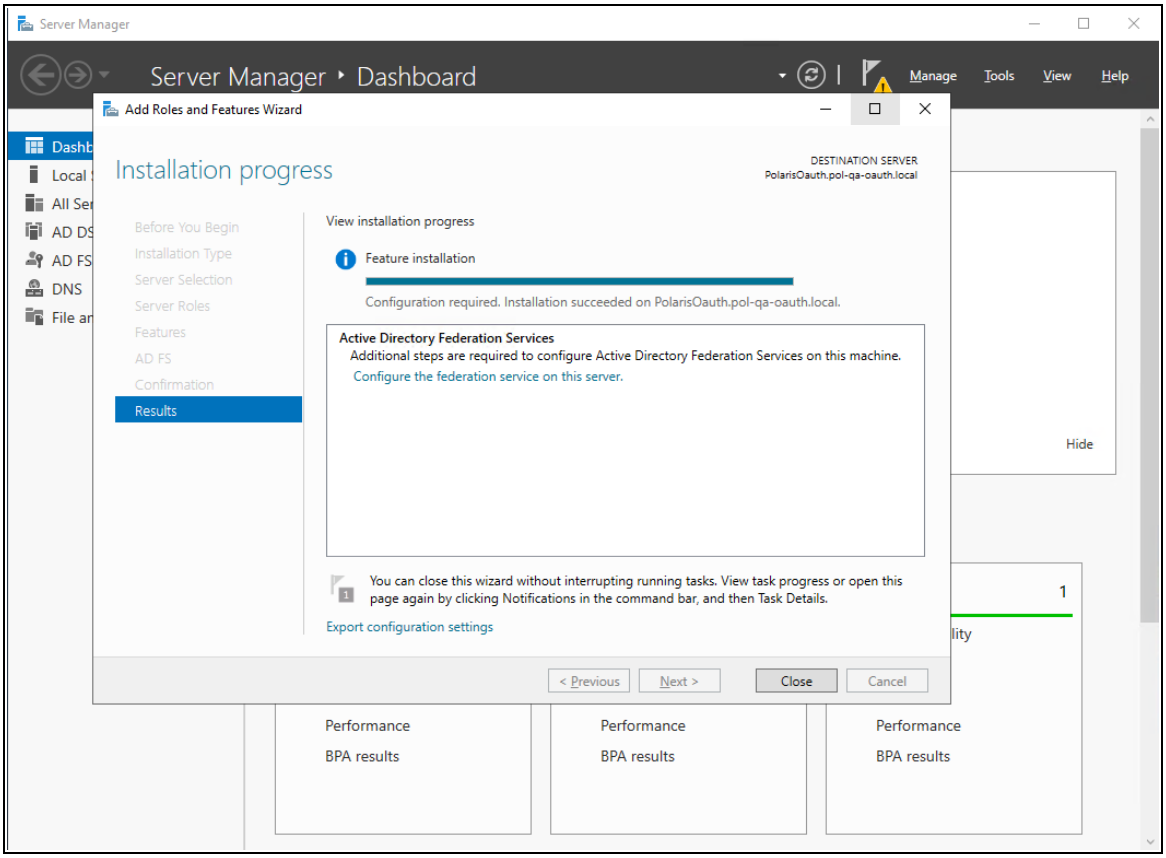
8. On the **Features** tab, select **Next**.



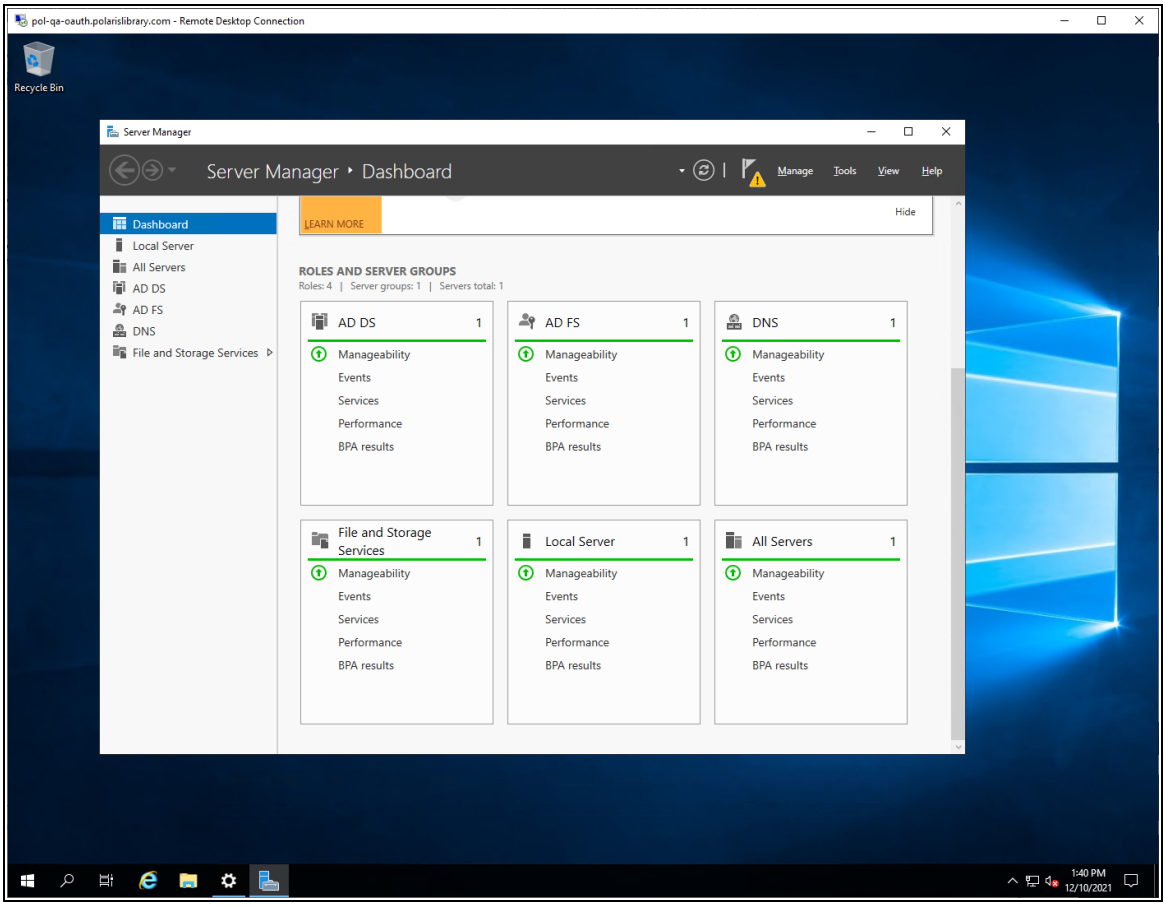
9. On the **AD FS** tab, read the Active Directory Federation Services (AD FS) information, and then select **Next**.



10. On the **Confirmation** tab, confirm your selections, and then select **Install**.



11. On the **Results** tab, select **Close** when the installation is complete.



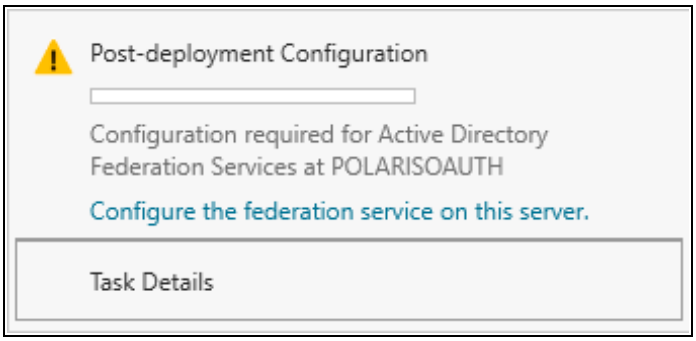
12. On the Server Manager dashboard, verify that AD FS is an installed role.
13. Restart the server.

Configure Active Directory Federation Services

To configure Active Directory Federation Services

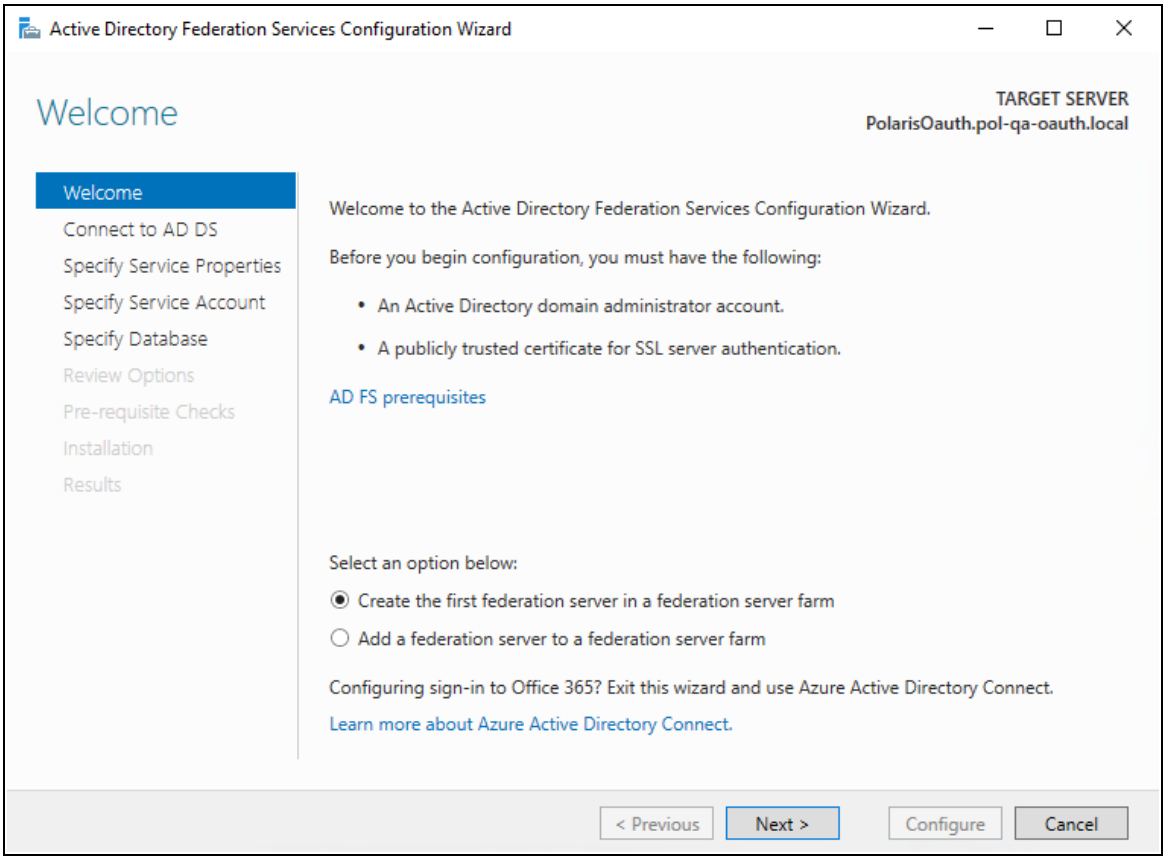
1. Start the Server Manager desktop application.

   The system generates a configuration notification.

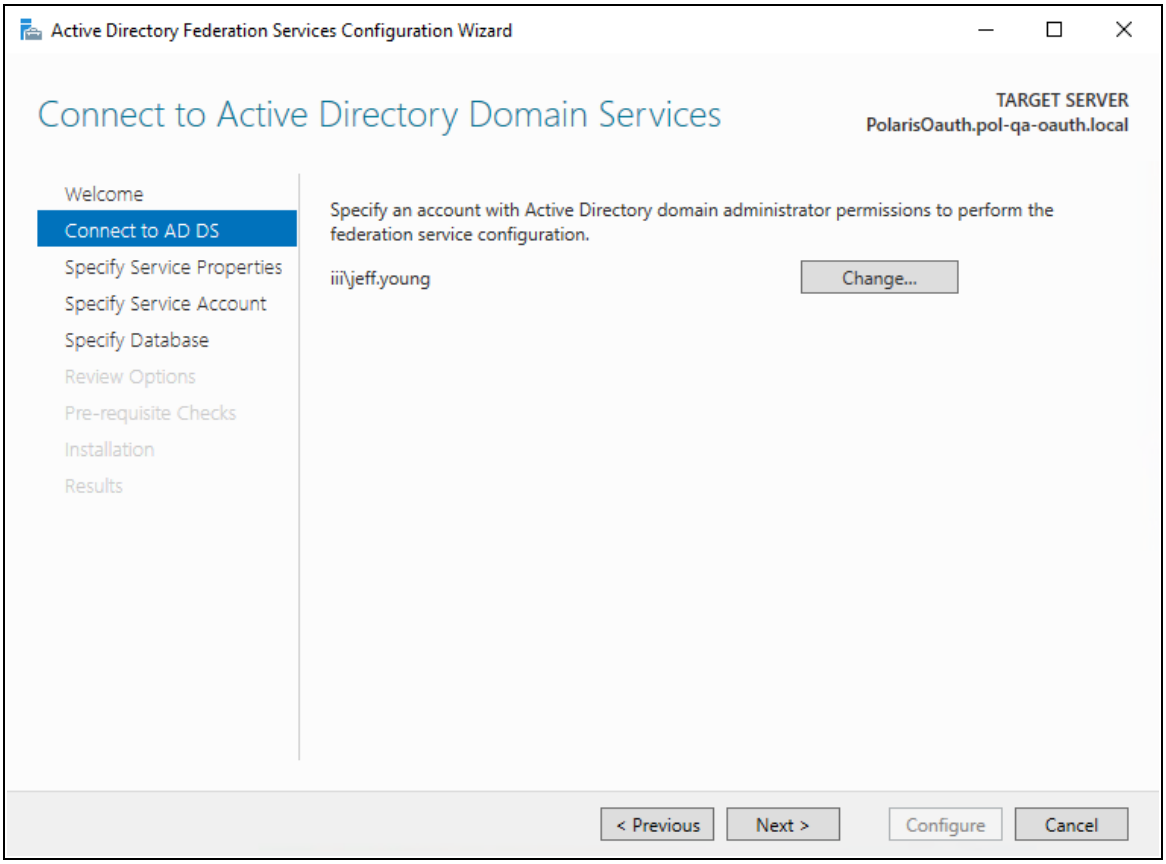


2. Open the notification, and select **Configure the federation service on this server**.

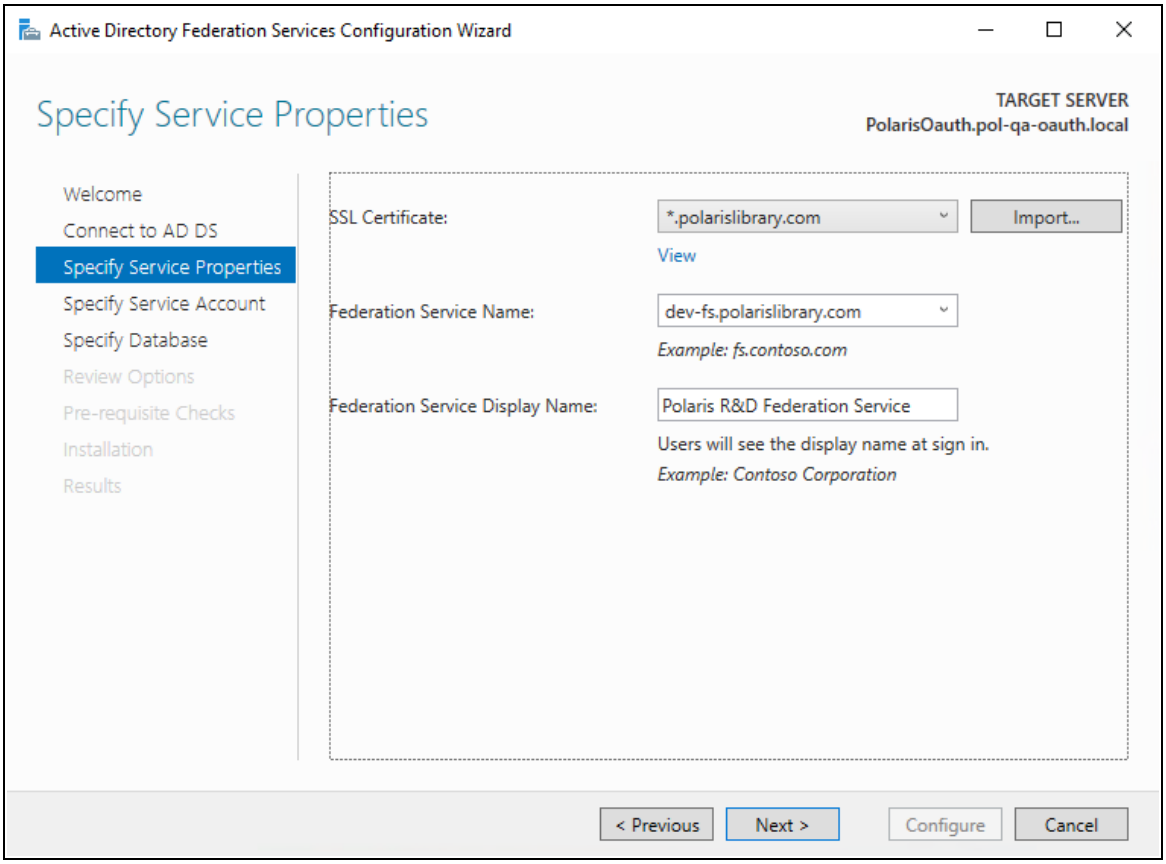
   The Active Directory Federation Services Configuration wizard opens.



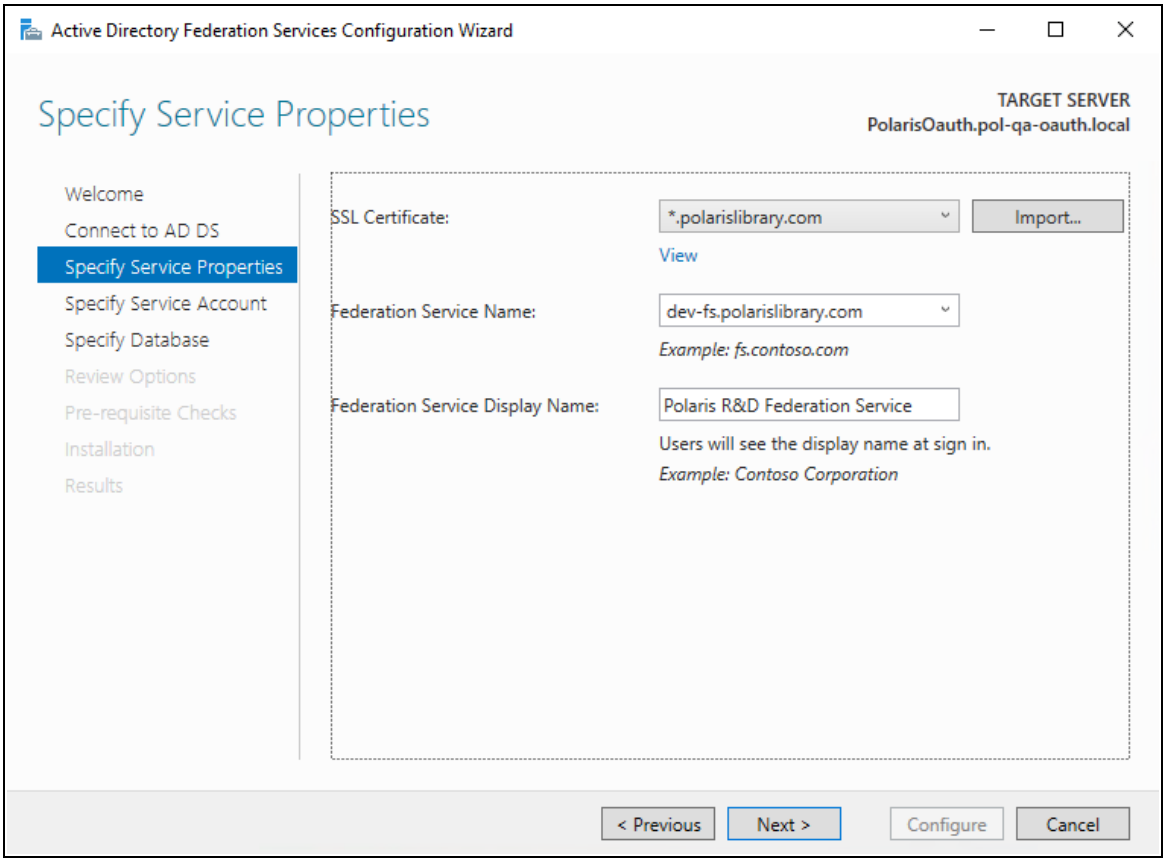
3. On the Welcome tab, select **Next**.



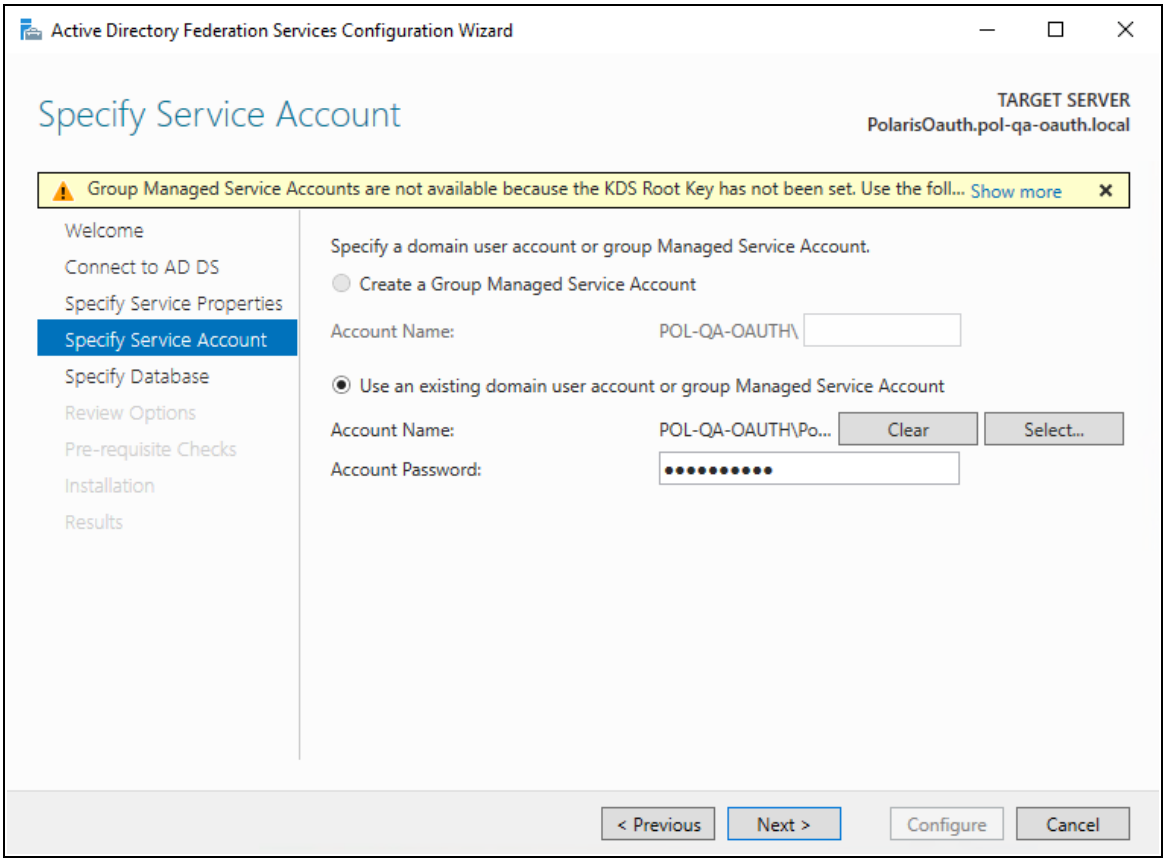
4. Select **Change**, and provide an administrator account. Then select **Next**.



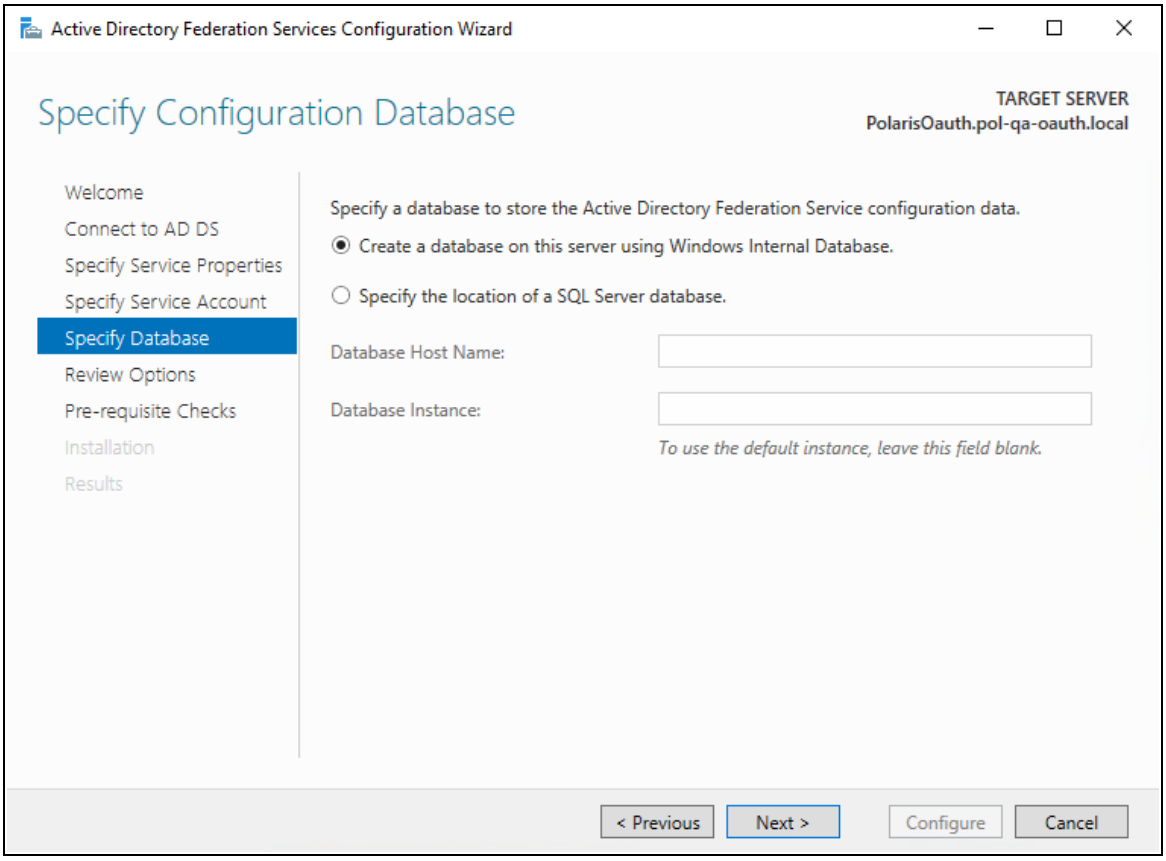
5. If not already installed on the server, select **Import** to install an SSL certificate.



6. Enter the following, and then select **Next**:
    - Federation Service Name
    - Federation Service Display Name

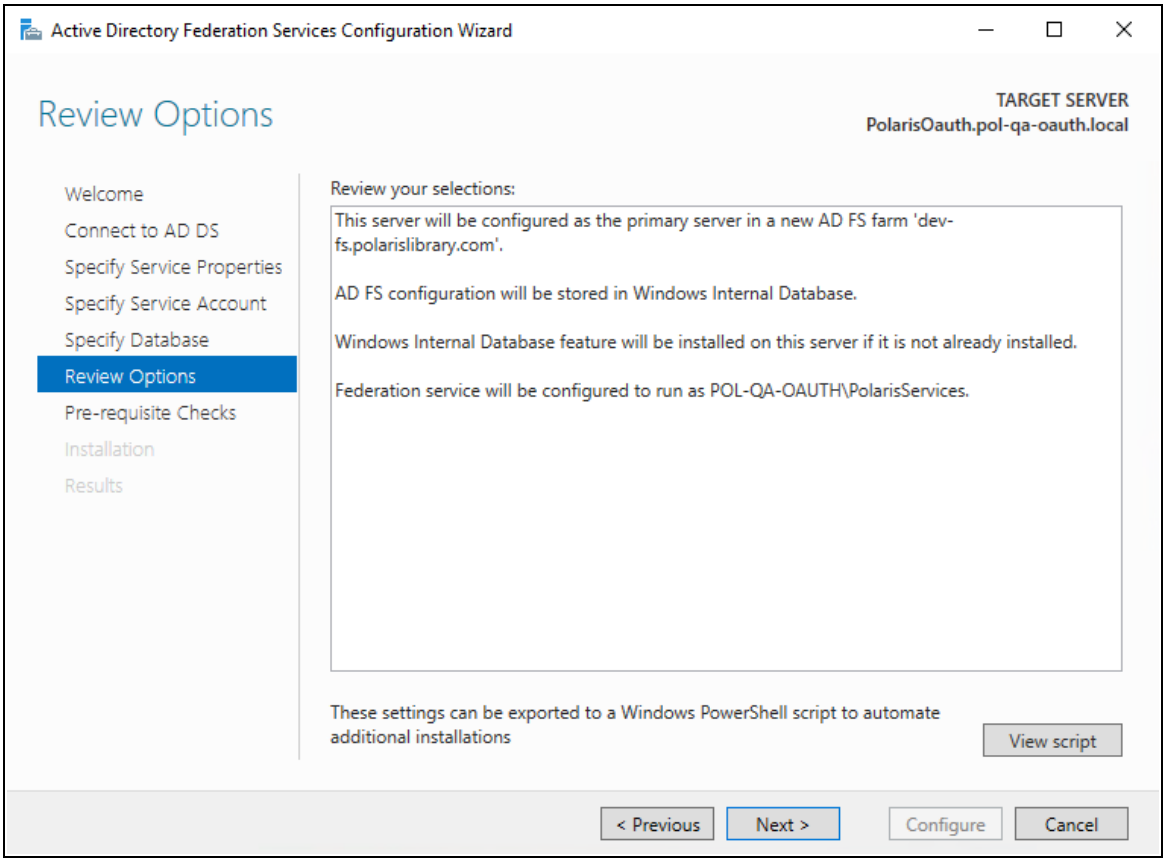


7. Specify a service account, and then select **Next**.

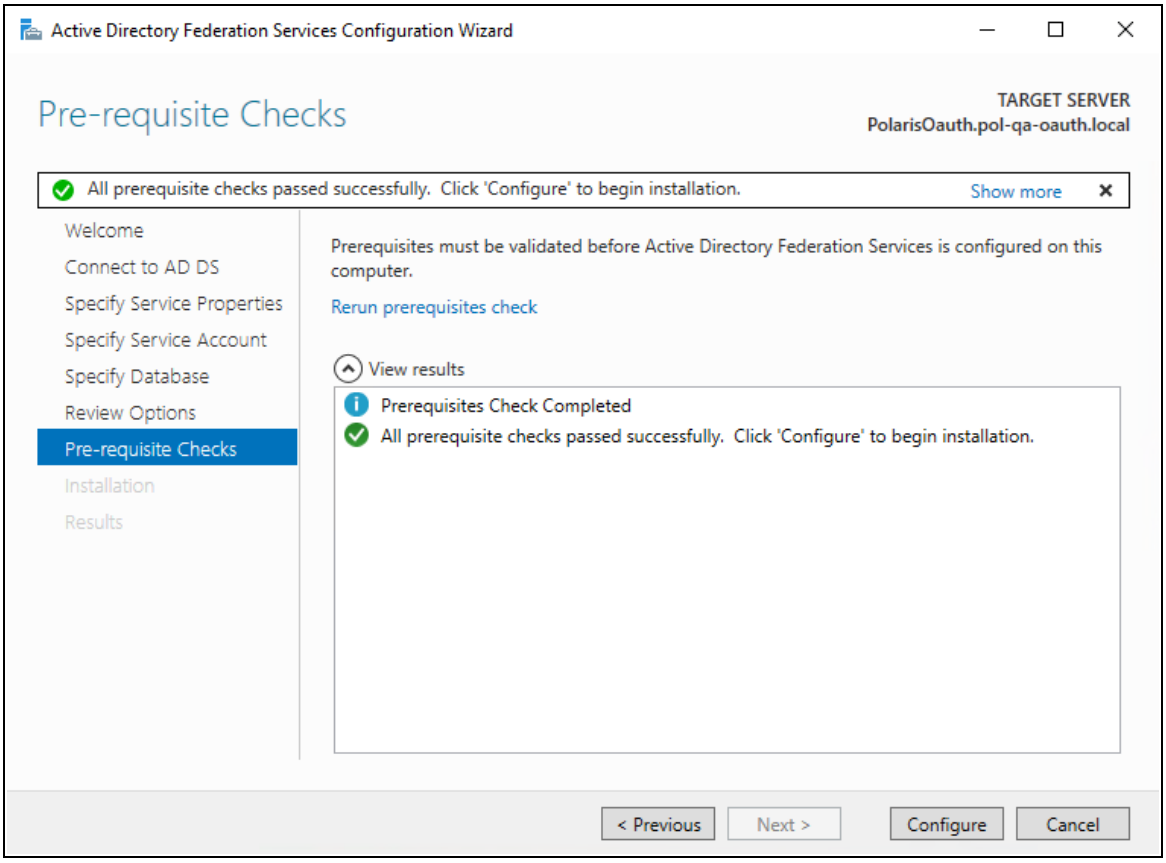


8. Specify the location of the AD FS configuration database, and then select **Next**.

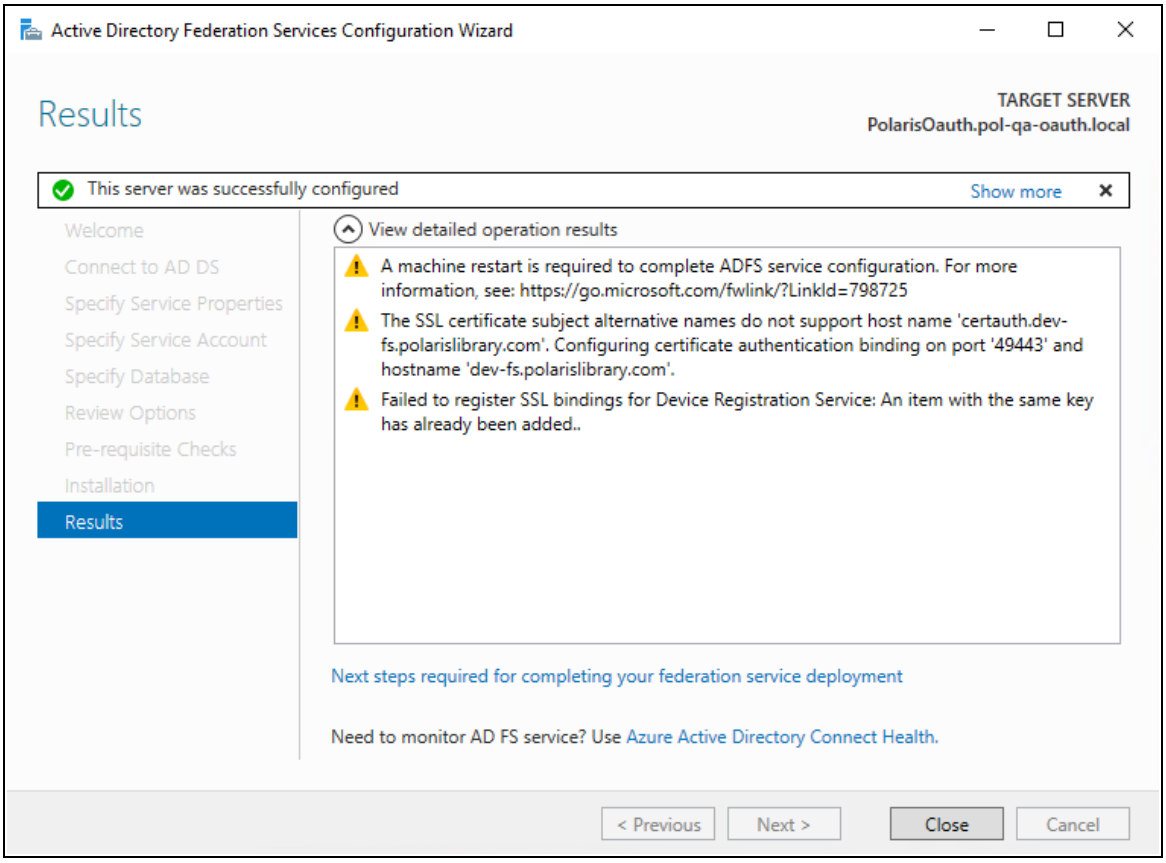
   For simple scenarios, creating the local database is acceptable.



9. Review your selections, and then select **Next**.



10. After you complete all pre-requisite checks, select **Configure**.



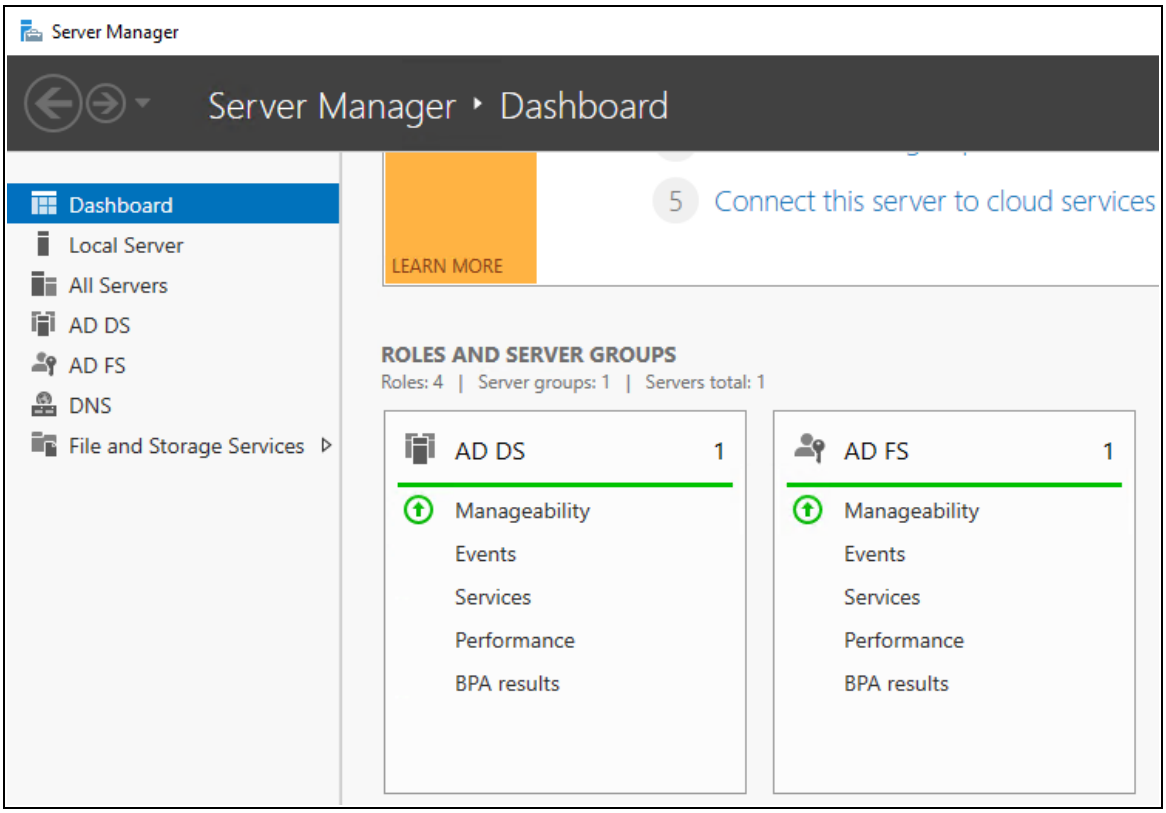
11. When the configuration wizard has completed successfully, select **Close**, and then restart the server.

Verify Active Directory Federation Services Is Running

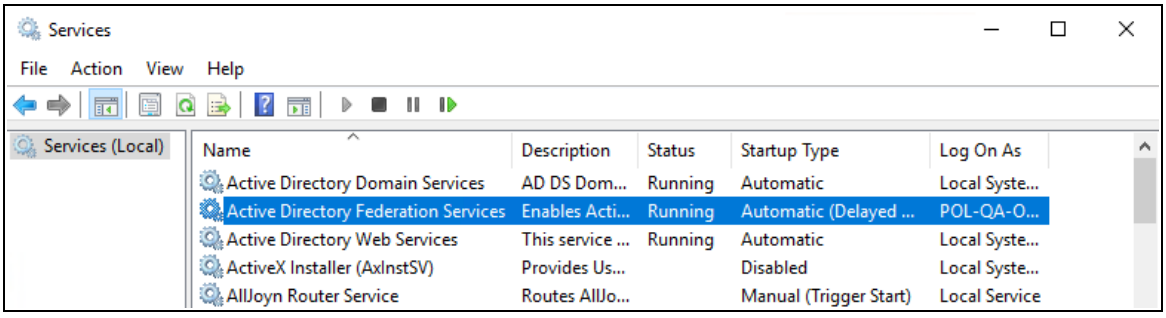
To verify that Active Directory Federation Services is running

1. Start the Server Manager desktop application.

   AD FS should be green.

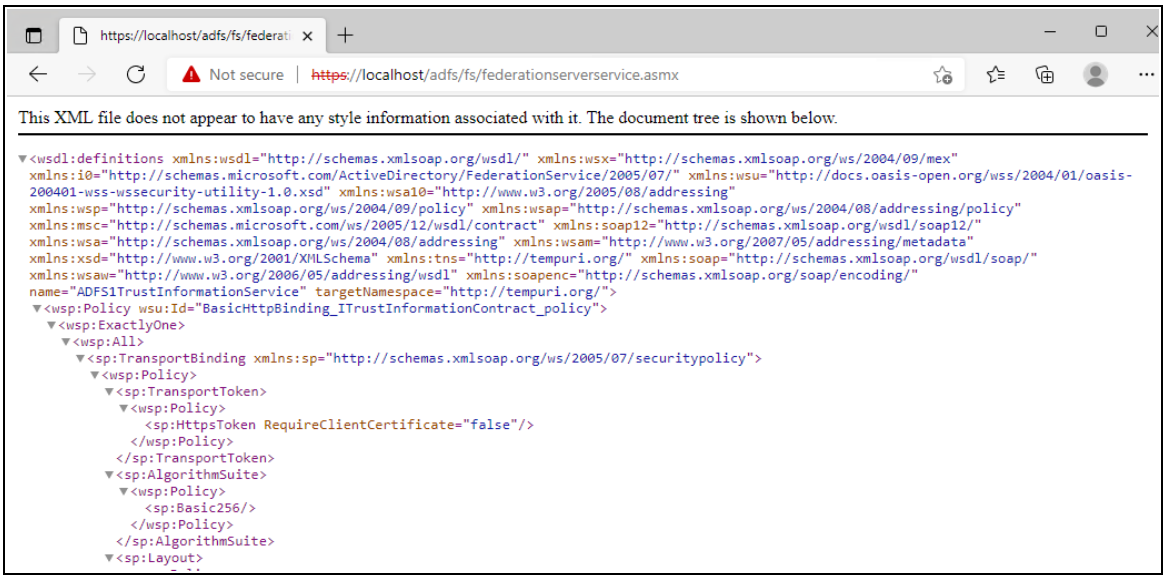


2. Start the Services application and check the status.



3. Open the Edge (or Chrome) web browser and go to https://localhost/adfs/fs/federationserverservice.asmx
    - If you want to ignore certificate errors, select **Advanced**.

    A page similar to the following image opens:

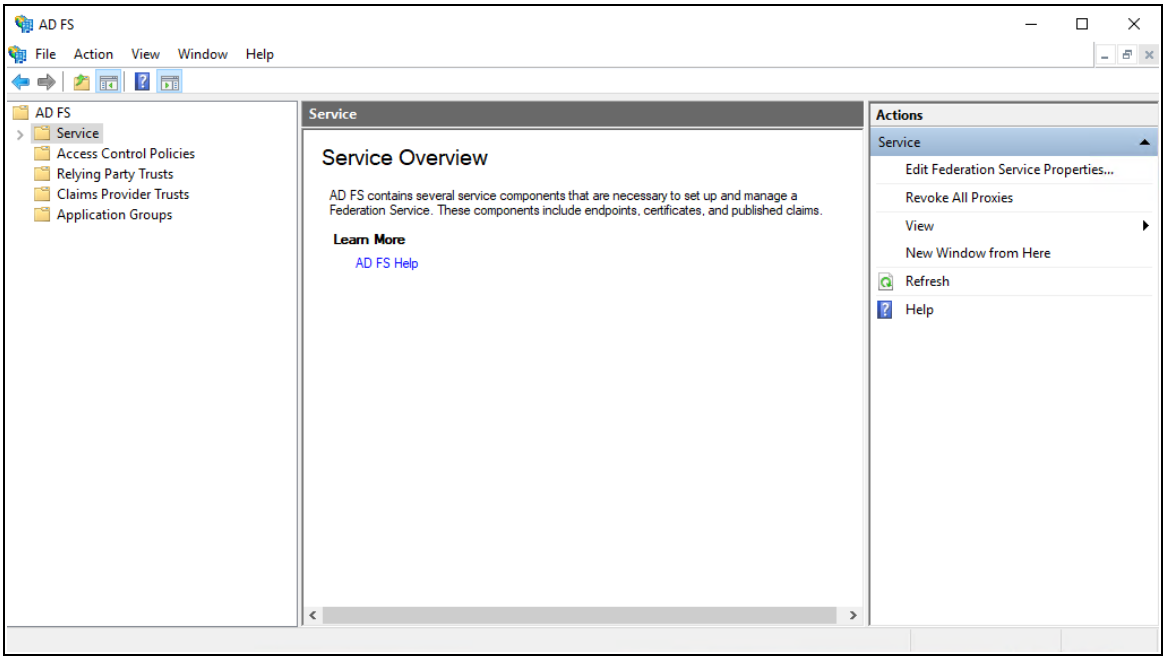


    This indicates that Active Directory Federation Services is running.

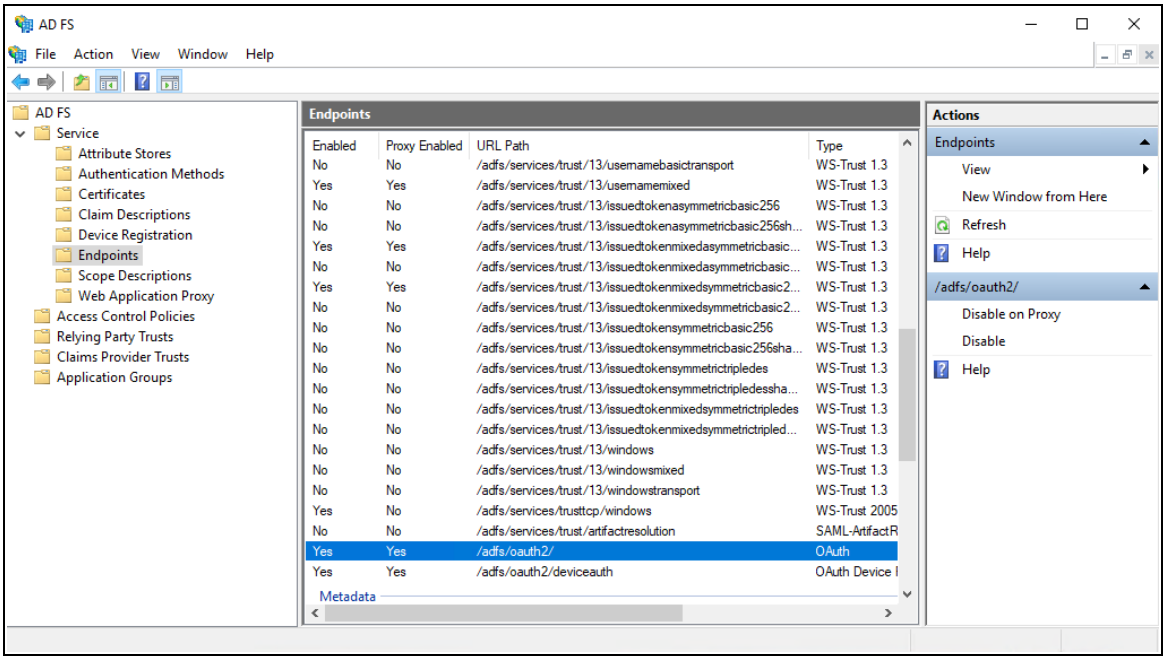
Verify that OAuth 2.0 is Enabled

To verify that OAuth 2.0 is enabled

1. Open the AD FS Management desktop application.



2. Open the **Service** folder, and then select the **Endpoint** folder.



3. Search for the oauth2 path.
4. In either the Edge or Chrome web browser, go to https://localhost/adfs/.well-known/openid-configuration
    - If you want to ignore certificate errors, select **Advanced**.

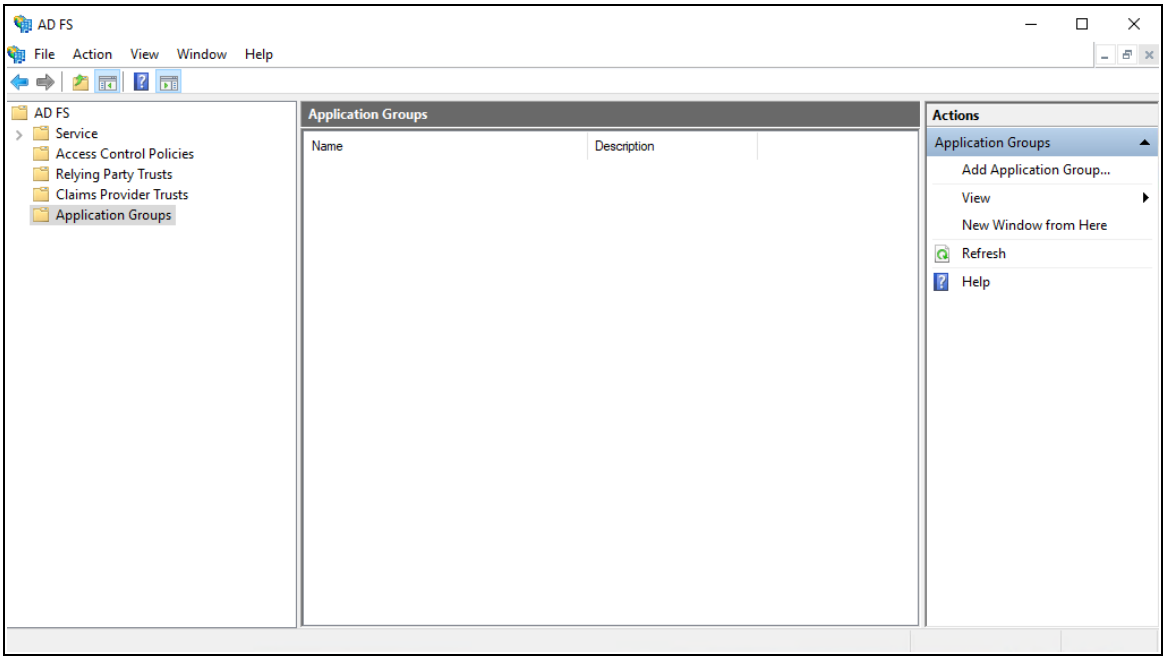
A page similar to the following image opens:

This indicates that OAuth 2.0 is available.

Create an Application Group

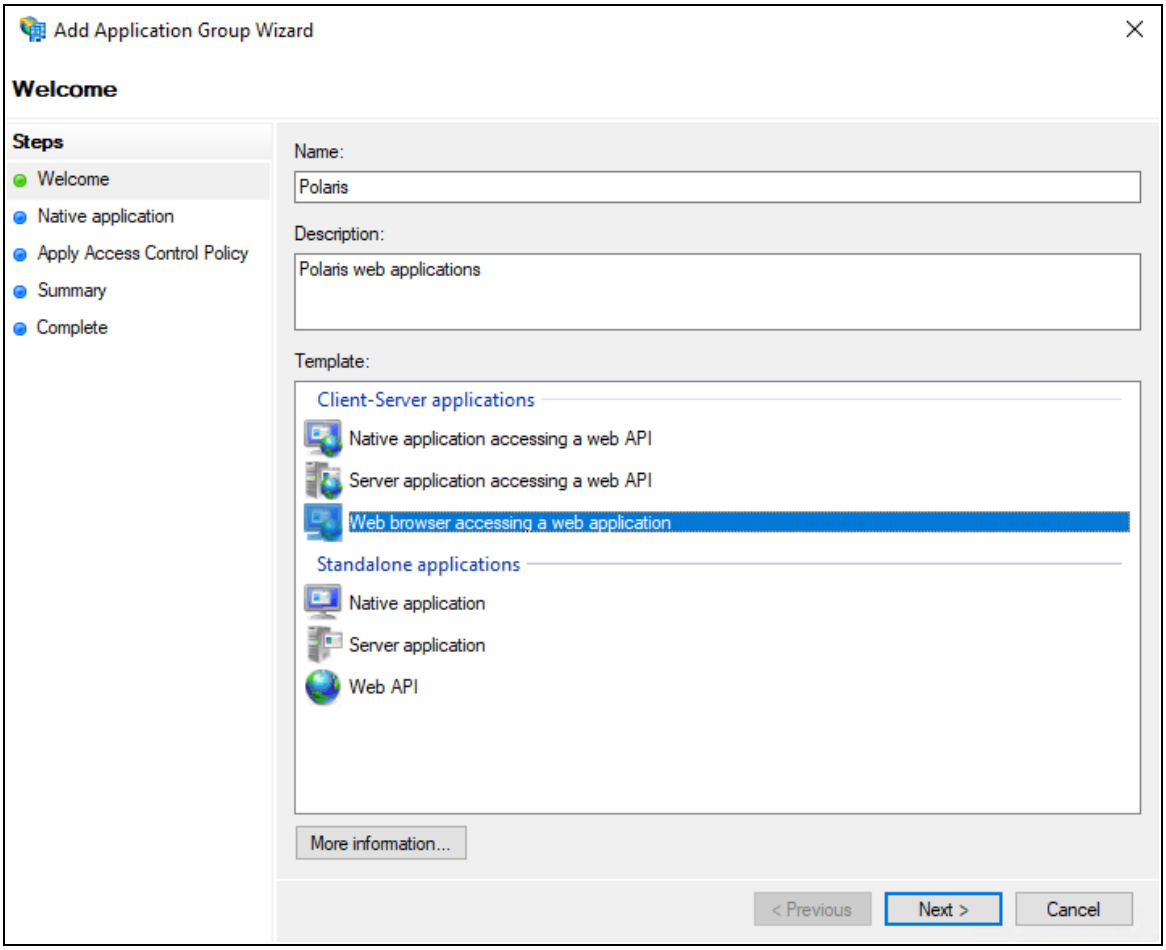
To create an application group for use with Polaris Admin and LeapWebApp

1. Open the AD FS Management desktop application.

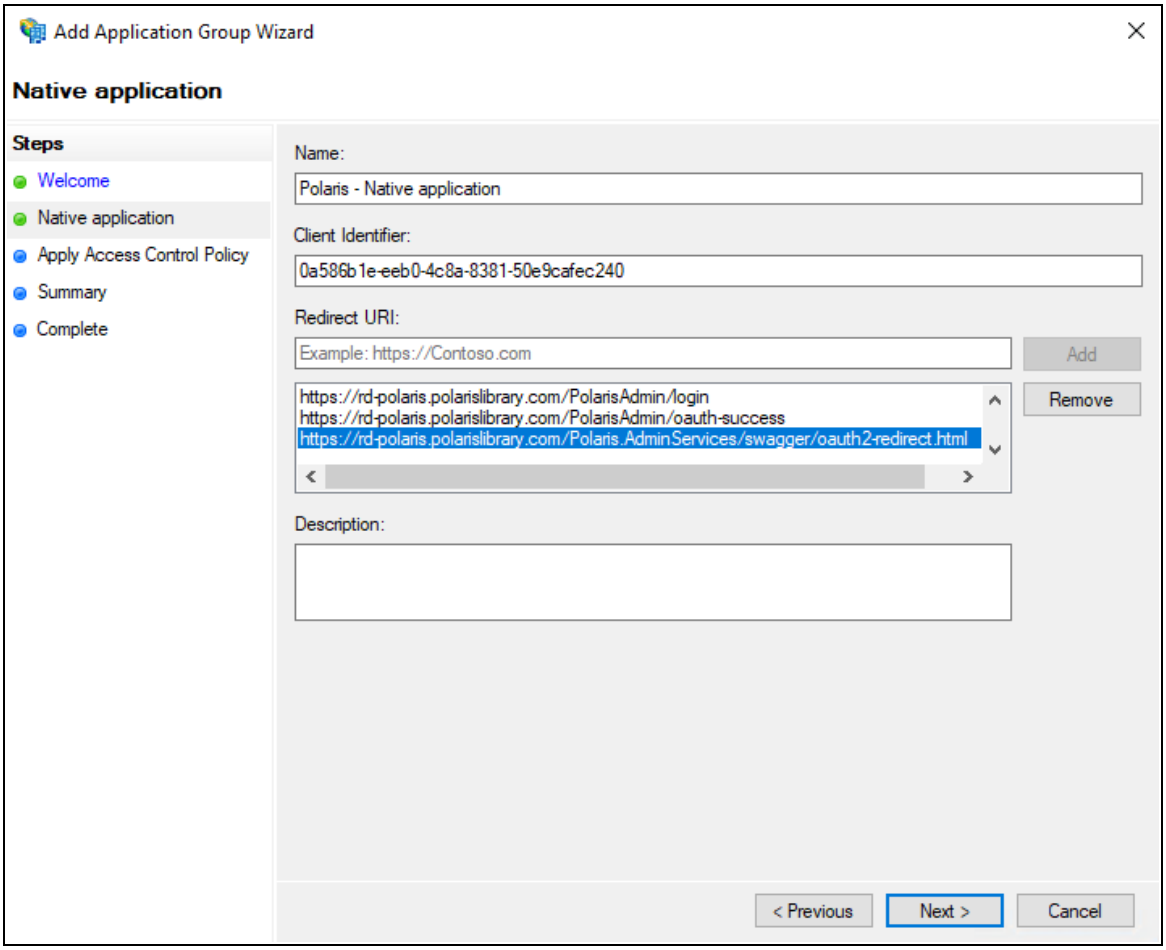


2. Select the **Application Groups** folder.
3. Under **Actions**, select **Add Application Group**.

   The Add Application Group wizard opens.



4. On the **Welcome** tab, do the following:
    a. In the **Name** box, enter **Polaris**.
    b. In the **Description** box, enter **Polaris web applications**.
    c. From the Template section, select **Web browser accessing a web application**.
5. Select **Next**.



6. On the **Native application** tab, in the **Redirect URI** box, enter the following URIs:
    - https://*server address*/PolarisAdmin/
    - https://*server address*/PolarisAdmin/login
    - https://*server address*/PolarisAdmin/oauth-success
    - https://*server address*/Polaris.AdminServices/swagger/oauth2-redirect.html
    - https://*server address*/LeapWebApp/signin-oidc
    - https://*server address*/LeapWebApp/signin-override-oidc
    - https://*server address*/LeapWebApp/signout-callback-oidc
    - https://*server address*/Polaris.ApplicationServices/swagger/oauth2-redirect.html

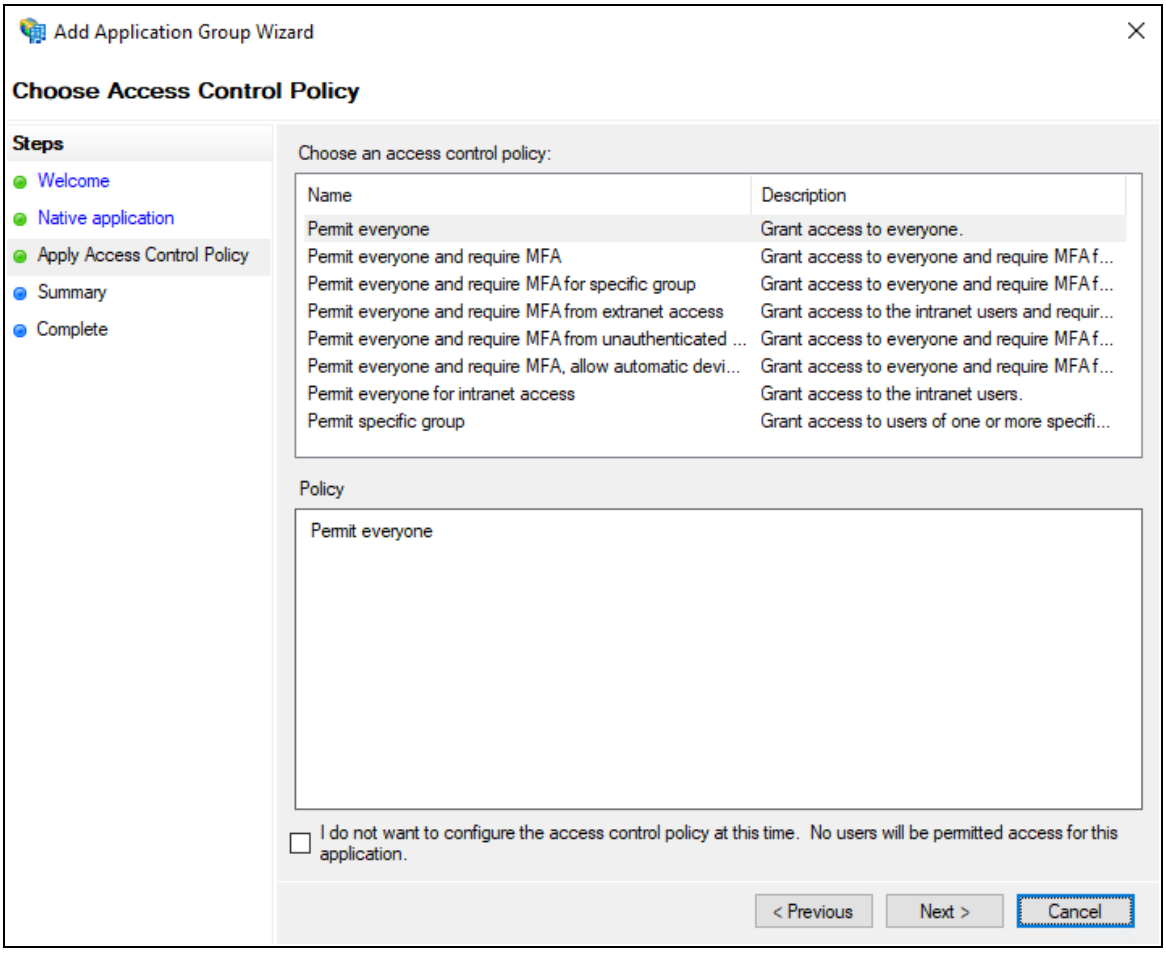
    Note:
    Replace *server address* with the FQDN that matches your Polaris

System Administration (web-based) or Leap server name and certificate.

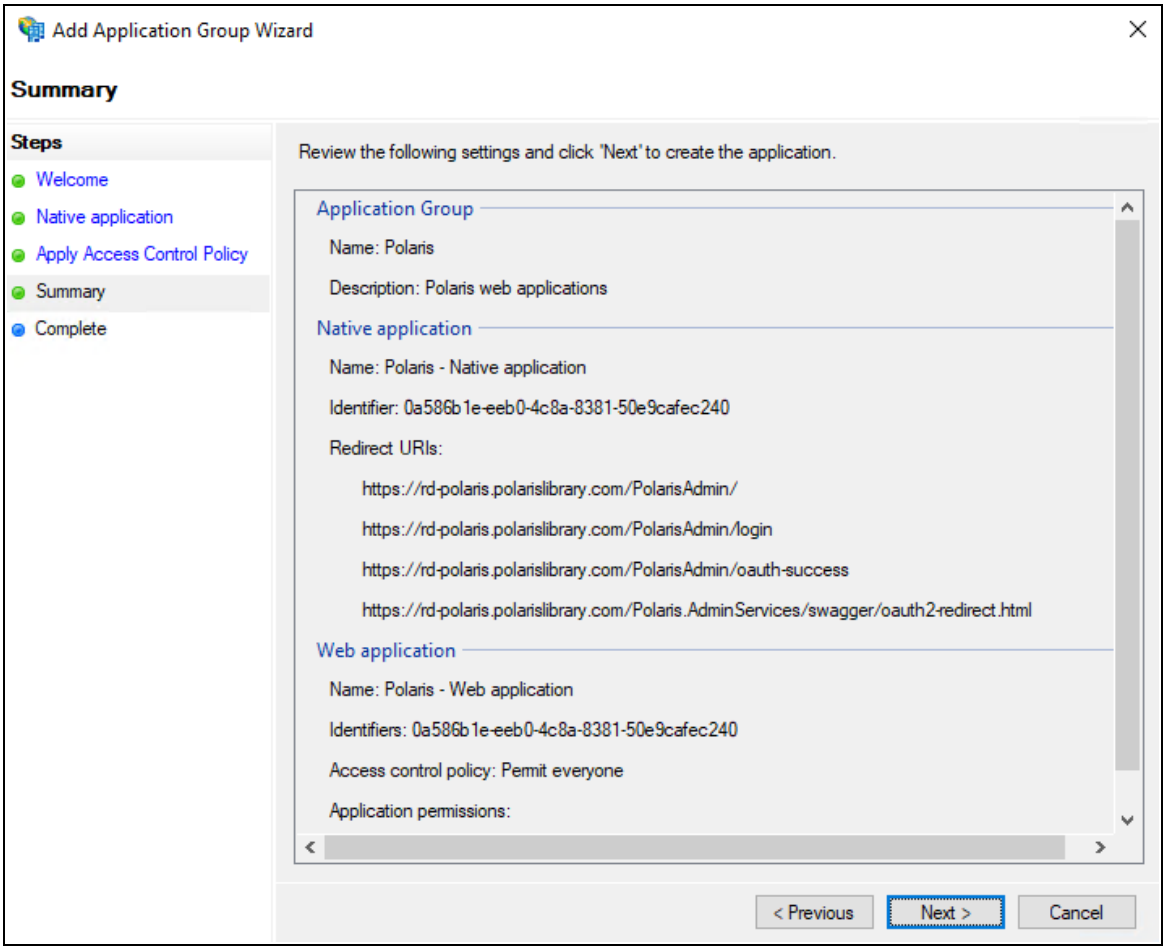
7. Copy the value in the **Client Identifier** box to Notepad.

   You'll need this when you set up PolarisAdmin's appsettings.user.json.

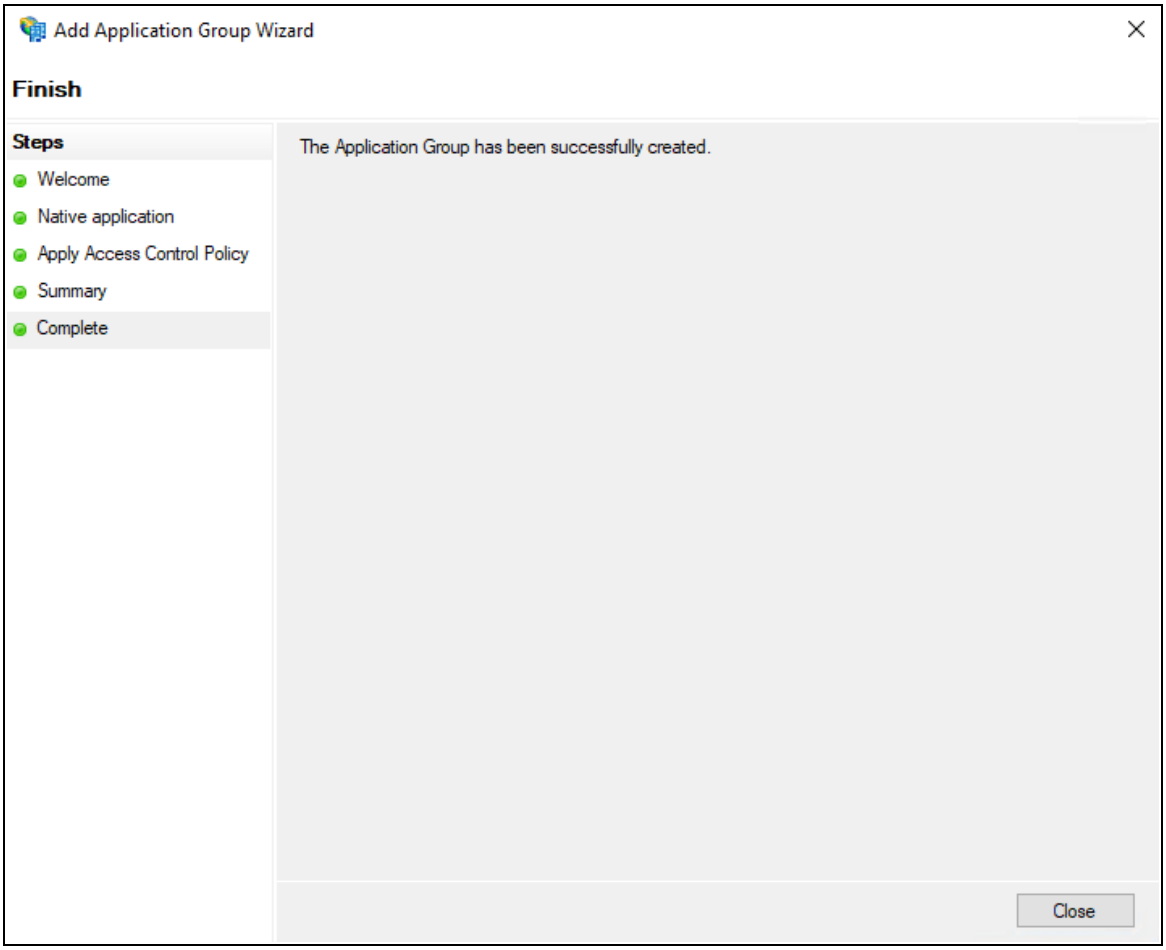
8. Select **Next**.



9. On the **Apply Access Control Policy** tab, select an access control policy, and then select **Next**.



10. On the **Summary** tab, review the settings and then select **Next**.

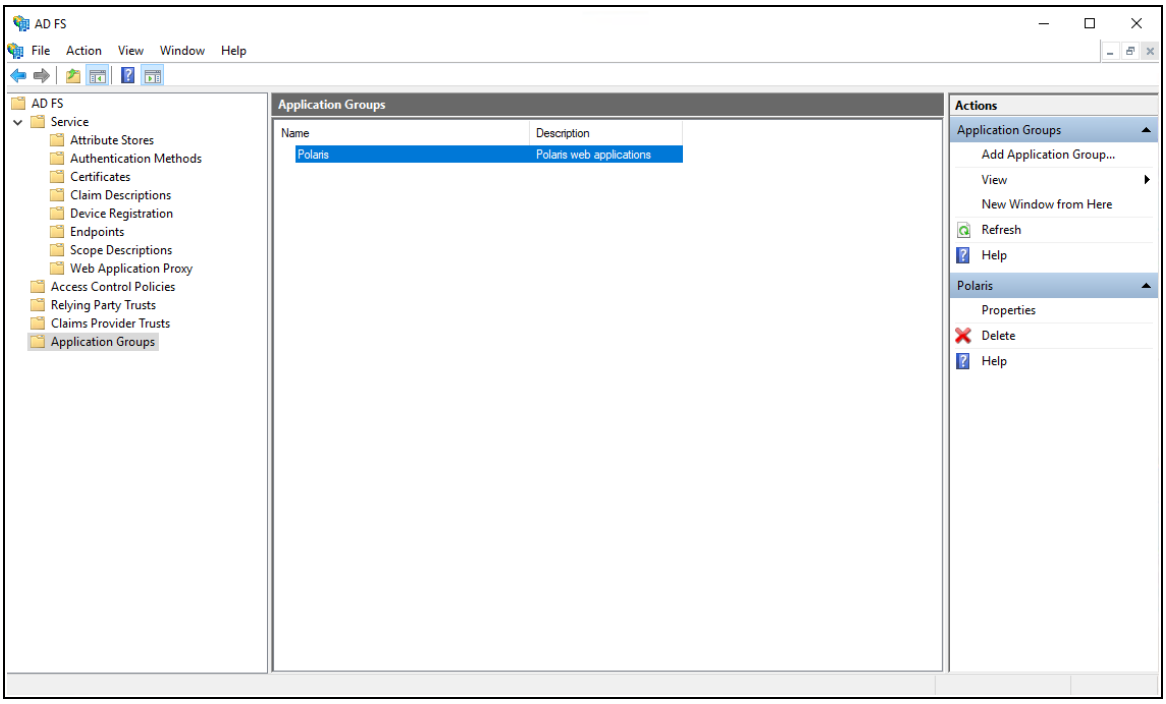


11. On the **Complete** tab, select **Close**.

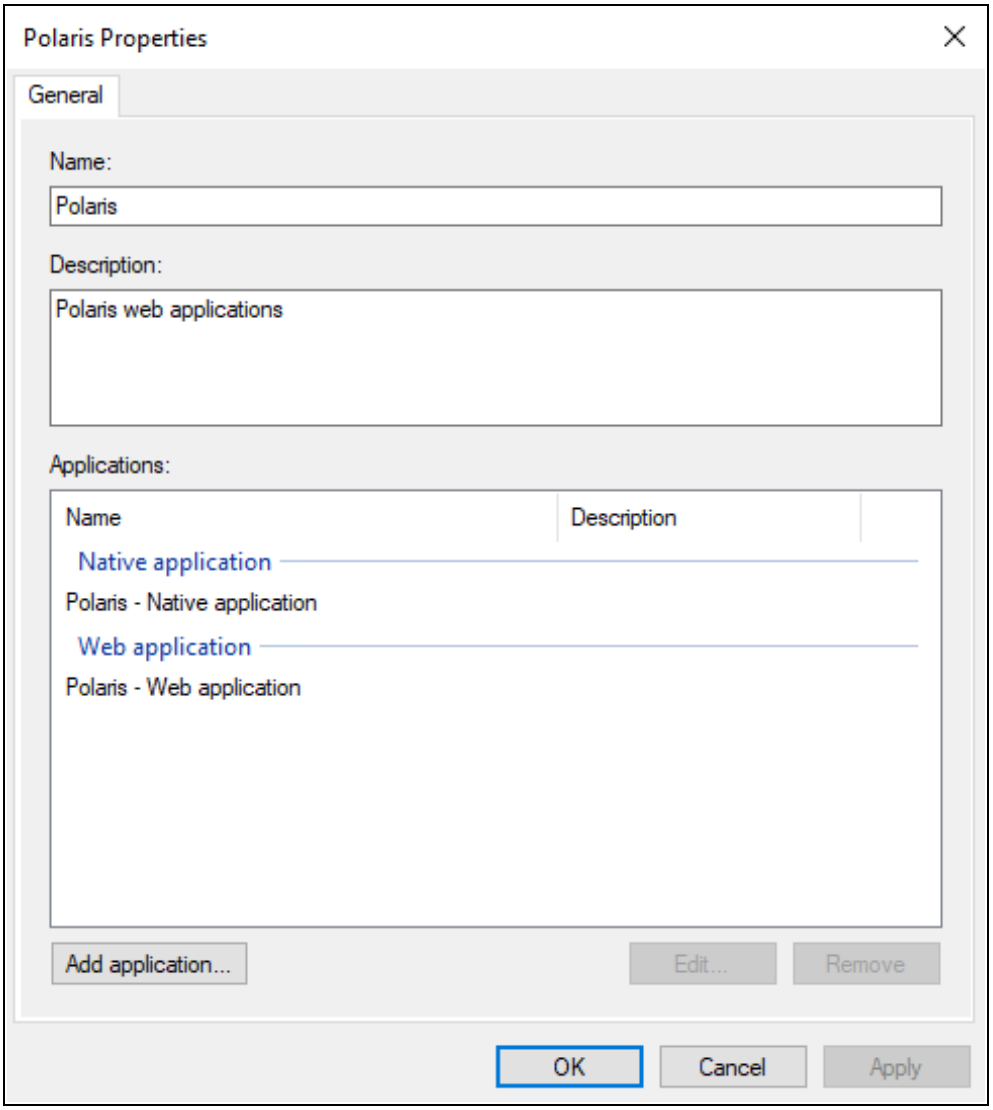
Configure the AD FS Web Application: Claims and Permitted Scopes

To configure the AD FS web application

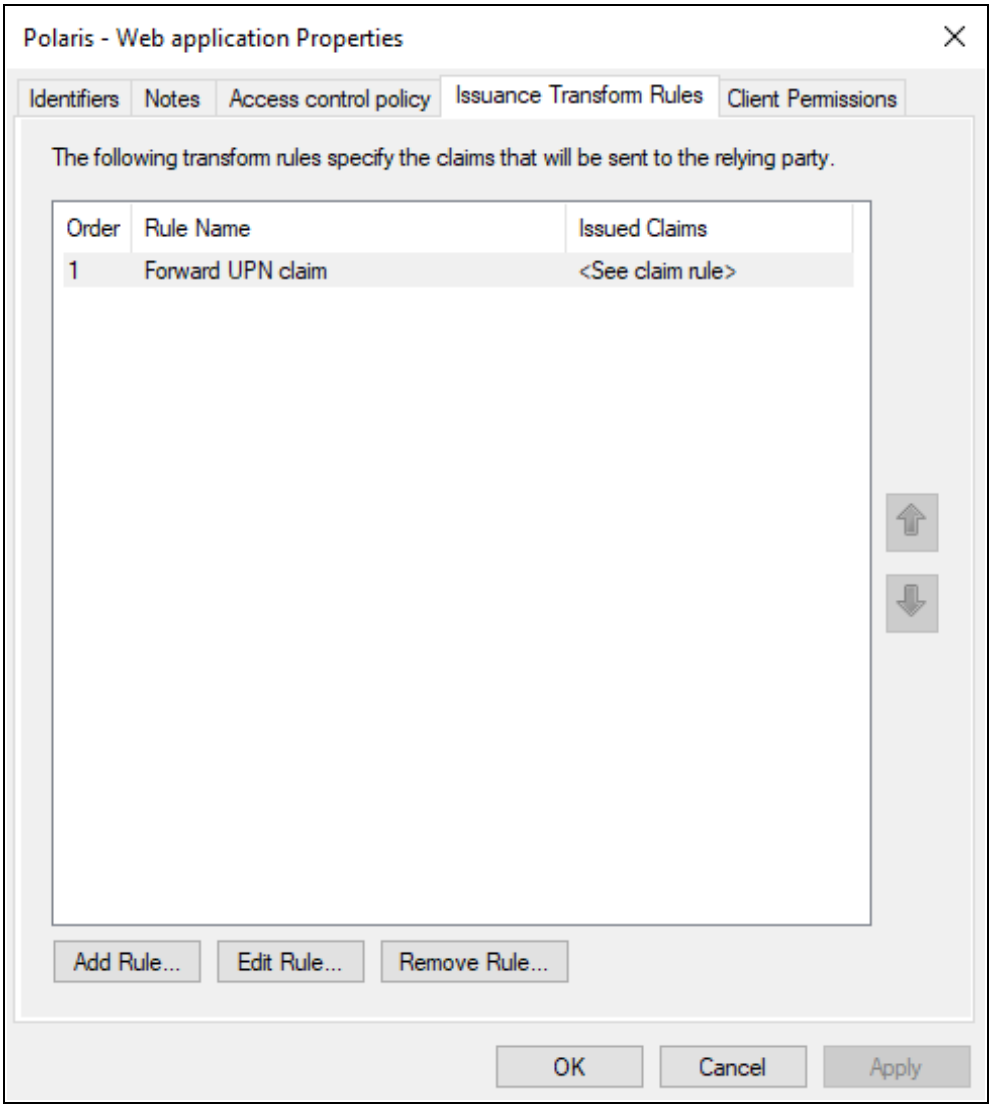
1. Open the AD FS Management desktop application.



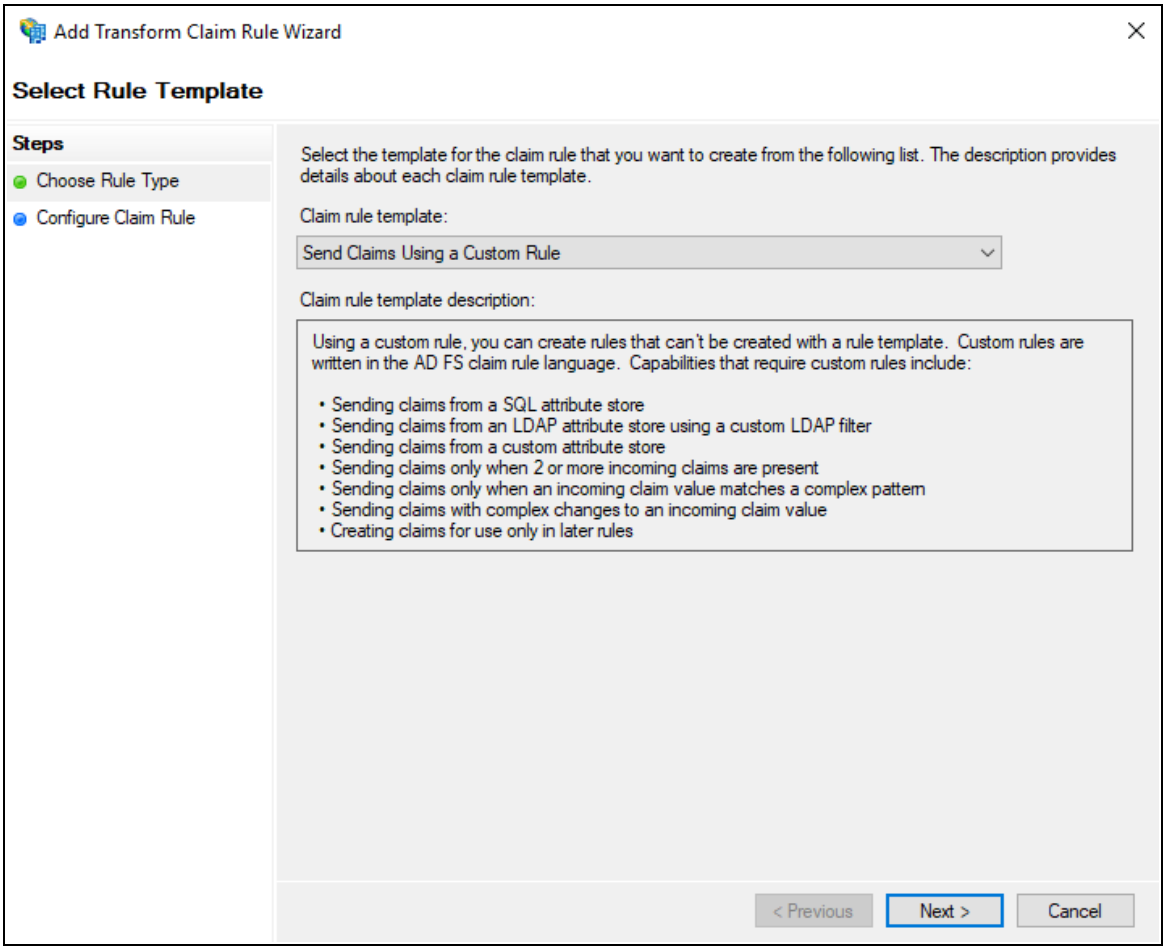
2. Select the **Application Groups** folder.
3. Select the **Polaris** application group, and then select **Properties**.



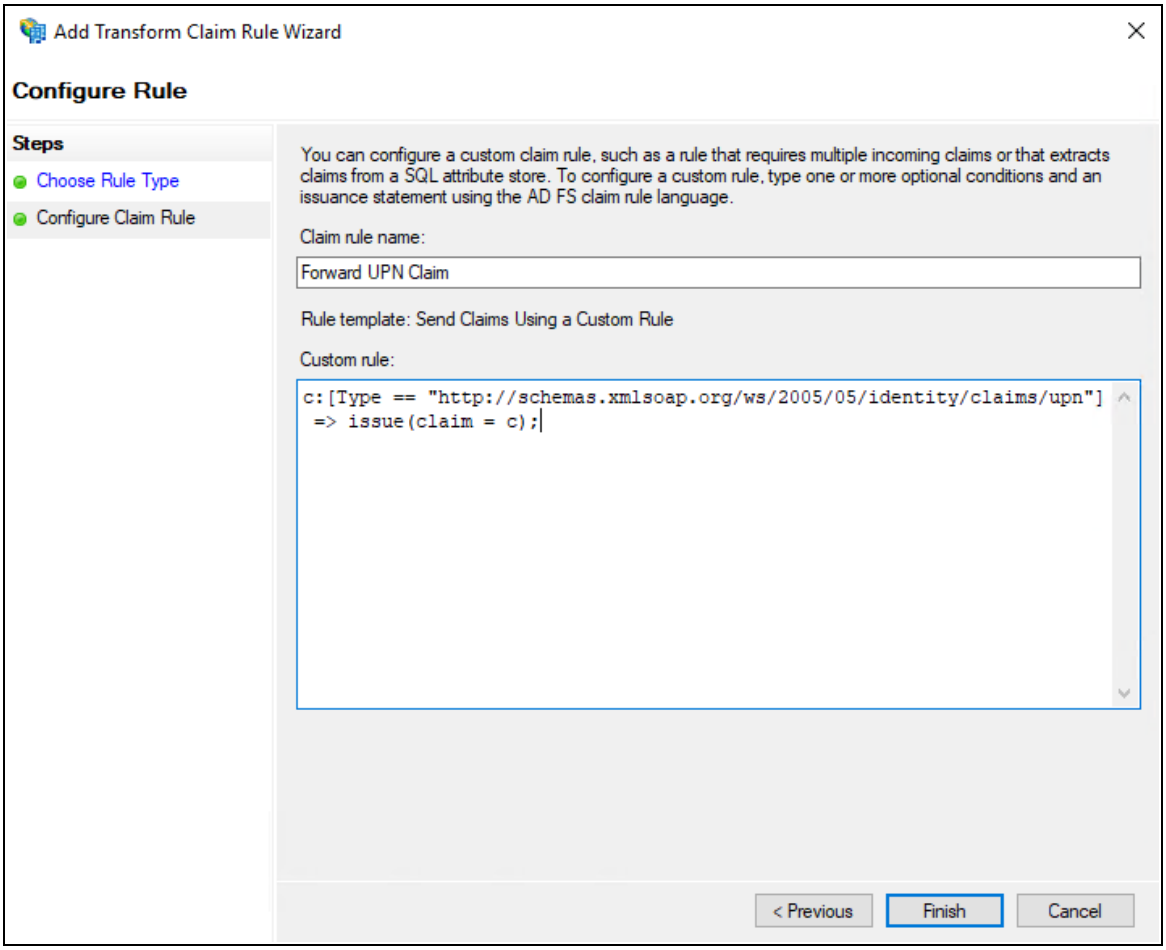
4. Select **Polaris** - **Web application**, and then select **Edit**.



5. Select the **Issuance Transform Rules** tab, and then select **Add Rule**.



6. On the Add Transform Claim Rule Wizard, select **Send Claims Using a Custom Rule** from the **Claim rule template** list, and then select **Next**.



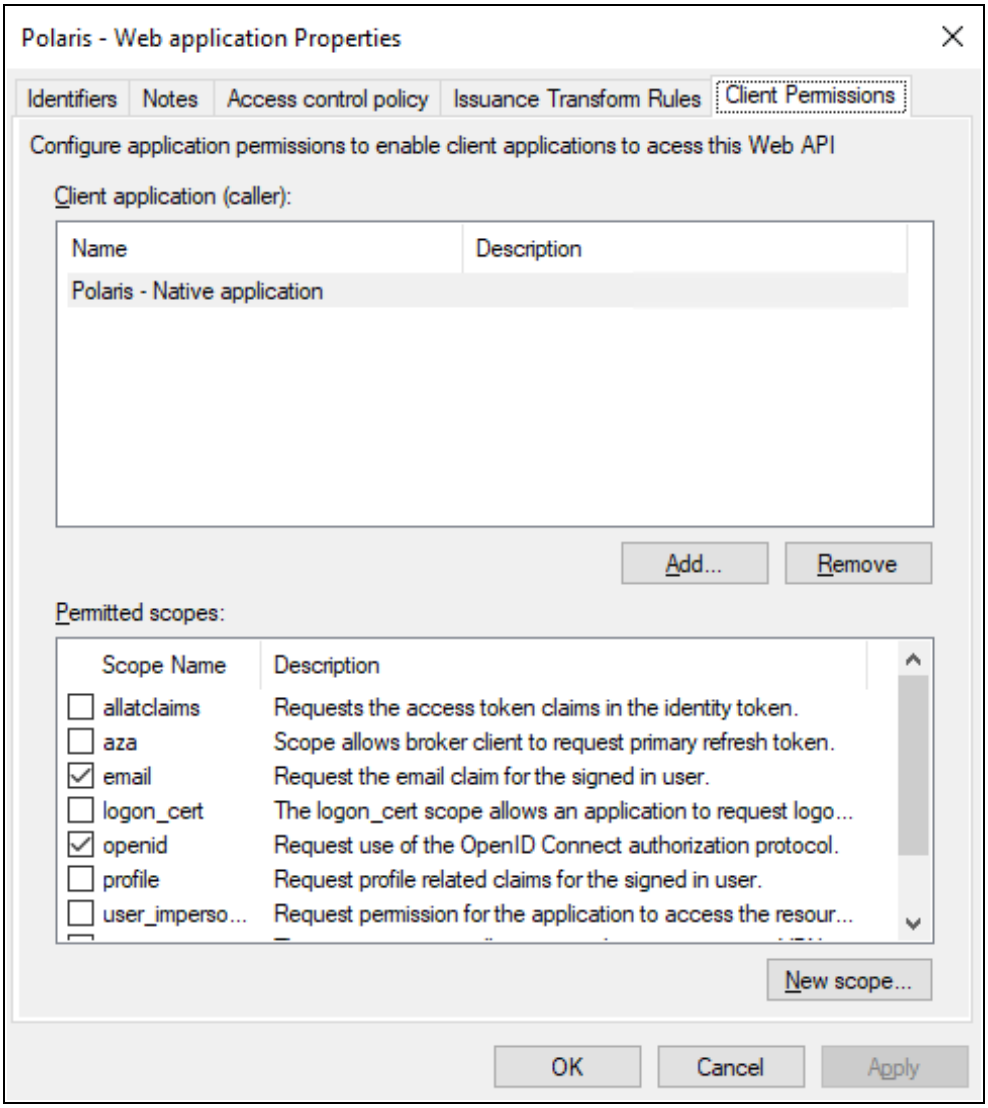
7. In the **Claim rule name** box, enter **Forward UPN Claim**.
8. In the **Custom rule** box, enter the following rule:

```
c:[Type ==
"http://schemas.xmlsoap.org/ws/2005/05/identity/claims/upn"]
=> issue(claim = c);
```

9. Select **Finish**.
10. On the **Issuance Transform Rules** tab, select **Add Rule**.
11. On the Add Transform Claim Rule Wizard, select **Send Claims Using a Custom Rule** from the **Claim rule template** list, and then select **Next**.
12. In the **Claim rule name** box, enter **Add TenantId**.
13. In the **Custom rule** box, enter the following rule:

```
=> issue(Type =
"http://schemas.microsoft.com/identity/claims/tenantid",
Value = "polaris");
```

14. Select **Finish**.



15. On the **Client Permissions** tab, verify that **email** and **openid** are selected.
16. Select **OK** to close the **Web application Properties** dialog.
17. Select **OK** to close the **Polaris properties** dialog.
18. Using the services applet, restart the Active Directory Federation Services service.

Enable CORS on AD FS To Accept Requests from Polaris APIs

To enable CORS on AD FS to accept requests from Polaris APIs

1. Refer to the information on the following page:
    - https://docs.microsoft.com/en-us/windows-server/identity/ad-fs/operations/customize-http-security-headers-ad-fs#cross-origin-resource-sharing-cors-headers
2. Use the following commands:
    - `Set-AdfsResponseHeaders -EnableCORS $true`
    - `Set-AdfsResponseHeaders -CORSTrustedOrigins` *`https://rd-polaris.polarislibrary.com,https://example2.com`*

> **Note:**
> Replace *https://rd-polaris.polarislibrary.com* and *https://example2.com* with your own URL or list of URLs.

Set Up Web Services and Applications

To set up each of the following web services and applications, you must configure a .json file for each of the following:

- Polaris.AdminServices (the API service)
- PolarisAdmin (the web-based Polaris System Administration application)
- Polaris.ApplicationServices (Leap's API service)
- LeapWebApp (Leap)

The four .json files are all named appsettings.user.json, but they reside in different directories:

- C:\Program Files\Polaris\7.2\Polaris.AdminServices
- C:\Program Files\Polaris\7.2\PolarisAdmin\assets
- C:\Program Files\Polaris\7.2\Polaris.ApplicationServices
- C:\Program Files\Polaris\7.2\Polaris\LeapWebApp

This section contains the following topics:

Set Up Polaris.AdminServices

To set up Polaris.AdminServices

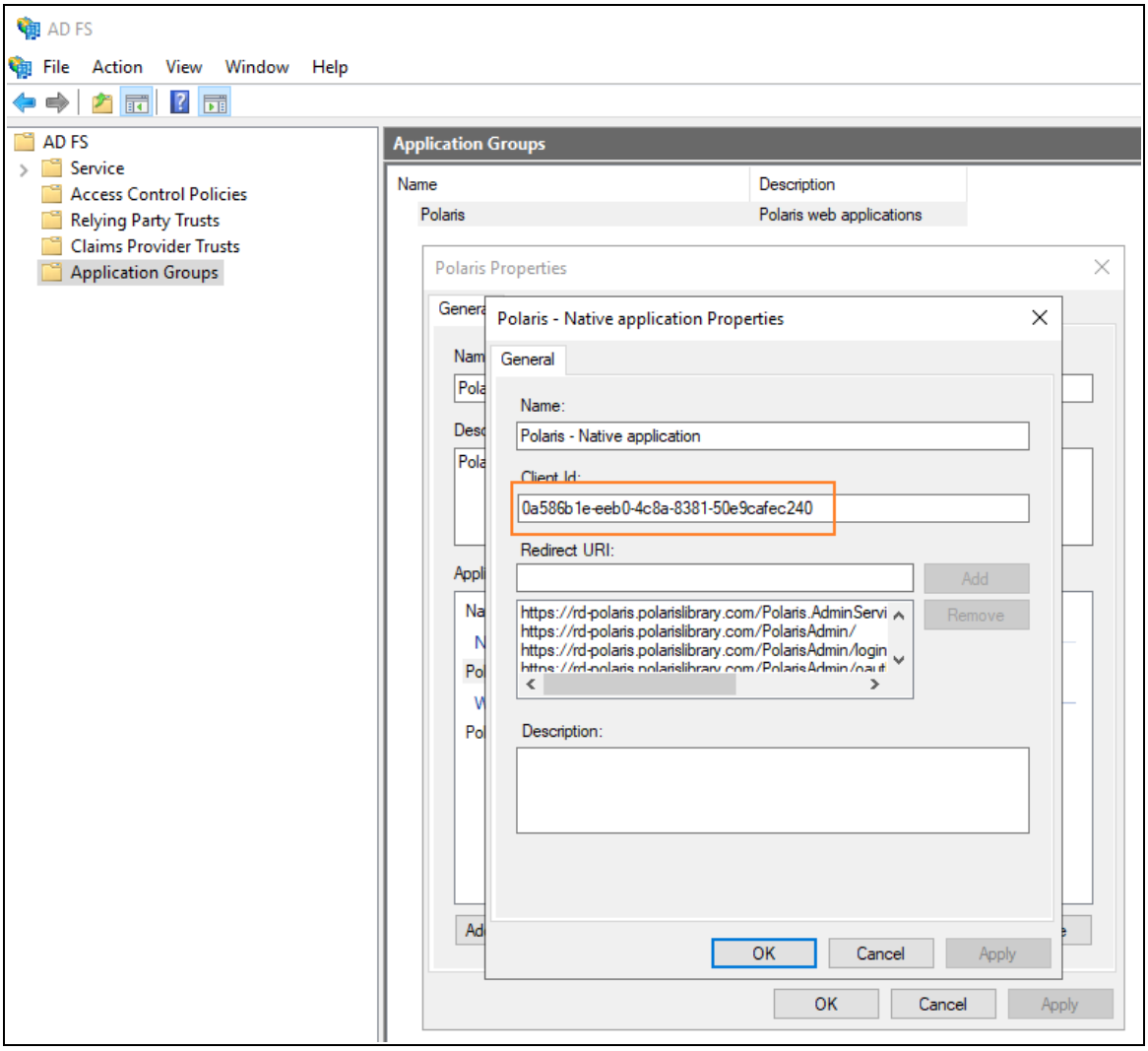
Verify that OAuth is Enabled

- Open C:\Program Files\Polaris\7.2\Polaris.AdminServices\appsettings.user.json and verify Polaris.OAuth.Enabled is set to true.

```
"Polaris": {
  "CachePermissions": true,
  "CORS": {
    "AllowedHosts": "https://rd-polaris.polarislibrary.com"
  },
  "BasicAuth": {
    "Enabled": false
  },
  "OAuth": {
    "Enabled": true,
    "ClientID": "0a586b1e-eeb0-4c8a-8381-50e9cafec240",
    "Authority": "https://dev-fs.polarislibrary.com/adfs/",
    "Audience": "microsoft:identityserver:0a586b1e-eeb0-4c8a-8381-50e9cafec240",
    "ValidIssuer": "http://dev-fs.polarislibrary.com/adfs/services/trust",
    "ValidAudience": "microsoft:identityserver:0a586b1e-eeb0-4c8a-8381-50e9cafec240",
    "AuthorizationUrl": "https://dev-fs.polarislibrary.com/adfs/oauth2/authorize",
    "TokenUrl": "https://dev-fs.polarislibrary.com/adfs/oauth2/token"
  },
```

Update the Client Id

1. On the AD FS server, open AD FS Management desktop application.



2. Copy the Client Id from the Polaris - Native application properties dialog.
3. Paste the copied Client Id into the appsettings.user.json file.
4. If you started from the template, replace [client-id-that-might-look-like-a-guid] with the copied Client Id.

   It should look like the following image when complete (your Client Id will be different):

```
"Polaris": {
  "CachePermissions": true,
  "CORS": {
    "AllowedHosts": "https://rd-polaris.polarislibrary.com"
  },
  "BasicAuth": {
    "Enabled": false
  },
  "OAuth": {
    "Enabled": true,
    "ClientID": "0a586b1e-eeb0-4c8a-8381-50e9cafec240",
    "Authority": "https://dev-fs.polarislibrary.com/adfs/",
    "Audience": "microsoft:identityserver:0a586b1e-eeb0-4c8a-8381-50e9cafec240",
    "ValidIssuer": "http://dev-fs.polarislibrary.com/adfs/services/trust",
    "ValidAudience": "microsoft:identityserver:0a586b1e-eeb0-4c8a-8381-50e9cafec240",
    "AuthorizationUrl": "https://dev-fs.polarislibrary.com/adfs/oauth2/authorize",
    "TokenUrl": "https://dev-fs.polarislibrary.com/adfs/oauth2/token"
  },
```

Update the AD FS Server Location

1. If you started from the template, replace [my-adfs-server-domain-name] with the AD FS server address.
2. It should look like the following when complete (your AD FS server address will be different):

```
"Polaris": {
  "CachePermissions": true,
  "CORS": {
    "AllowedHosts": "https://rd-polaris.polarislibrary.com"
  },
  "BasicAuth": {
    "Enabled": false
  },
  "OAuth": {
    "Enabled": true,
    "ClientID": "0a586b1e-eeb0-4c8a-8381-50e9cafec240",
    "Authority": "https://dev-fs.polarislibrary.com/adfs/",
    "Audience": "microsoft:identityserver:0a586b1e-eeb0-4c8a-8381-50e9cafec240",
    "ValidIssuer": "http://dev-fs.polarislibrary.com/adfs/services/trust",
    "ValidAudience": "microsoft:identityserver:0a586b1e-eeb0-4c8a-8381-50e9cafec240",
    "AuthorizationUrl": "https://dev-fs.polarislibrary.com/adfs/oauth2/authorize",
    "TokenUrl": "https://dev-fs.polarislibrary.com/adfs/oauth2/token"
  },
```

Set Up PolarisAdmin

To set up PolarisAdmin

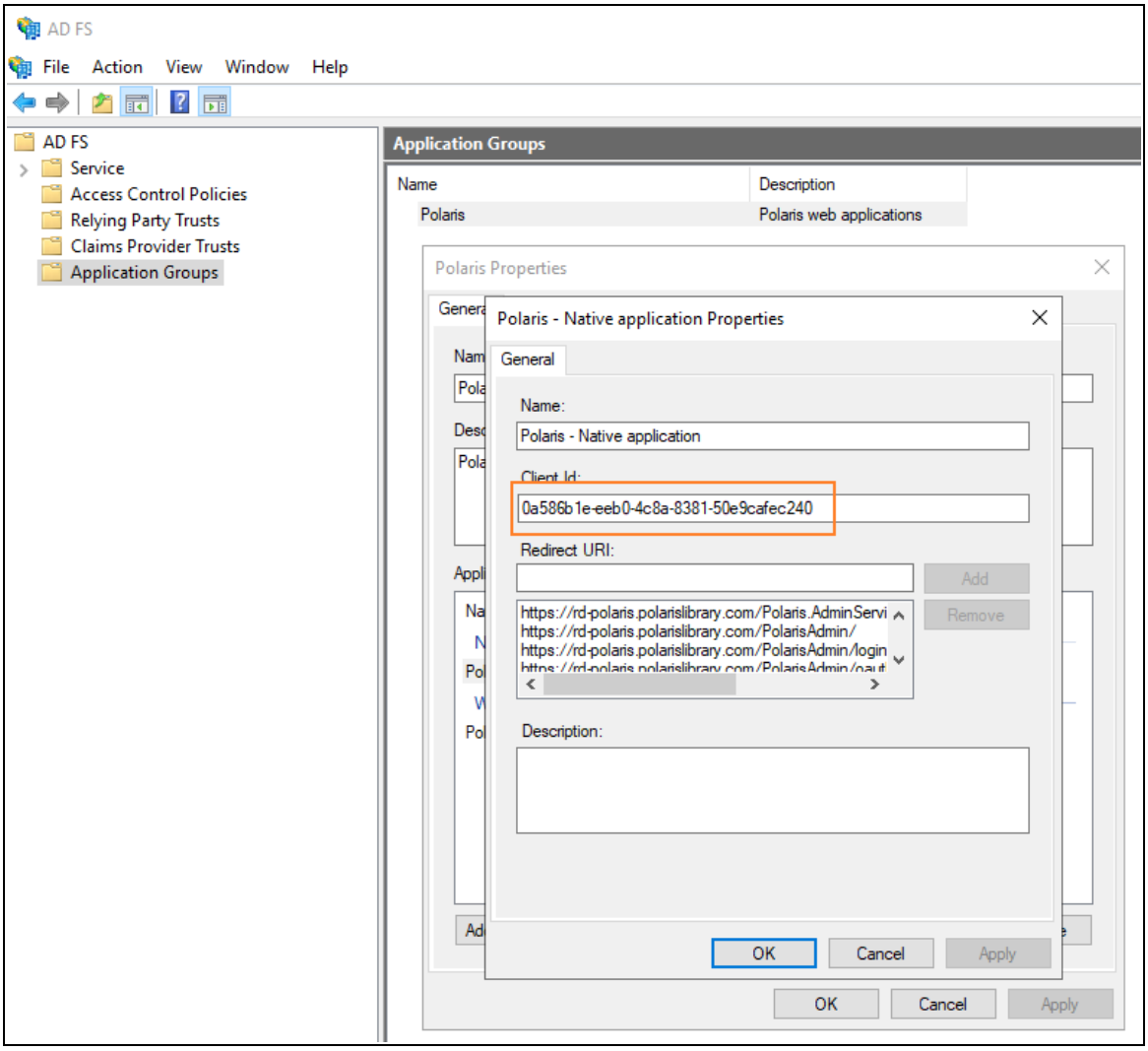
Verify that OAuth is Enabled

- Open C:\Program Files\Polaris\7.2\PolarisAdmin\assets\appsettings.user.json and verify that oauthEnabled is set to true.

```
{
    "apiUrlRoot": "https://rd-polaris.polarislibrary.com/polaris.adminservices/api/",
    "oauthEnabled": true,
    "msal": {
        "auth": {
          "clientId": "0a586b1e-eeb0-4c8a-8381-50e9cafec240",
          "authority": "https://dev-fs.polarislibrary.com/adfs/",
          "knownAuthorities": ["dev-fs.polarislibrary.com"],
          "redirectUri": "https://rd-polaris.polarislibrary.com/PolarisAdmin/oauth-success",
          "postLogoutRedirectUri": "https://rd-polaris.polarislibrary.com/PolarisAdmin",
          "protocolMode": "OIDC",
          "navigateToLoginRequestUrl": false
        },
        "cache": {
            "cacheLocation" : "localStorage",
            "storeAuthStateInCookie": false,
            "secureCookies": true
        },
        "guard": {
            "interactionType": "redirect",
            "authRequest": {
                "scopes": ["openid", "profile", "email", "urn:microsoft:userinfo"]
            },
            "loginFailedRoute": "/login-failed"
        },
        "interceptor": {
            "interactionType": "redirect",
            "protectedResourceMap": [
                ["https://rd-polaris.polarislibrary.com/Polaris.AdminServices/api/protected/", ["email"]]
            ]
        }
    }
}
```

Update the Client Id

1. On the AD FS server, open AD FS Management desktop application.



2. Copy the Client Id from the Polaris - Native application Properties dialog.
3. Paste the copied Client Id into the appsettings.user.json file.
4. If you started from the template, replace [CLIENTID-ASSIGNED-IN-ADFS] with the copied Client Id.

   It should look like the following when complete (your Client Id will be different):

```
{
    "apiUrlRoot": "https://rd-polaris.polarislibrary.com/polaris.adminservices/api/",
    "oauthEnabled": true,
    "msal": {
        "auth": {
          "clientId": "0a586b1e-eeb0-4c8a-8381-50e9cafec240",
          "authority": "https://dev-fs.polarislibrary.com/adfs/",
          "knownAuthorities": ["dev-fs.polarislibrary.com"],
          "redirectUri": "https://rd-polaris.polarislibrary.com/PolarisAdmin/oauth-success",
          "postLogoutRedirectUri": "https://rd-polaris.polarislibrary.com/PolarisAdmin",
          "protocolMode": "OIDC",
          "navigateToLoginRequestUrl": false
        },
        "cache": {
            "cacheLocation" : "localStorage",
            "storeAuthStateInCookie": false,
            "secureCookies": true
        },
        "guard": {
            "interactionType": "redirect",
            "authRequest": {
                "scopes": ["openid", "profile", "email", "urn:microsoft:userinfo"]
            },
            "loginFailedRoute": "/login-failed"
        },
        "interceptor": {
            "interactionType": "redirect",
            "protectedResourceMap": [
                ["https://rd-polaris.polarislibrary.com/Polaris.AdminServices/api/protected/", ["email"]]
            ]
        }
    }
}
```

Update AD FS Server Location

- If you started from the template, replace [ADFS-SERVER-ADDR] with the AD FS server address.

  It should look like the following when complete (your AD FS server address will be different):

```
{
    "apiUrlRoot": "https://rd-polaris.polarislibrary.com/polaris.adminservices/api/",
    "oauthEnabled": true,
    "msal": {
        "auth": {
          "clientId": "0a586b1e-eeb0-4c8a-8381-50e9cafec240",
          "authority": "https://dev-fs.polarislibrary.com/adfs/",
          "knownAuthorities": ["dev-fs.polarislibrary.com"],
          "redirectUri": "https://rd-polaris.polarislibrary.com/PolarisAdmin/oauth-success",
          "postLogoutRedirectUri": "https://rd-polaris.polarislibrary.com/PolarisAdmin",
          "protocolMode": "OIDC",
          "navigateToLoginRequestUrl": false
        },
        "cache": {
            "cacheLocation" : "localStorage",
            "storeAuthStateInCookie": false,
            "secureCookies": true
        },
        "guard": {
            "interactionType": "redirect",
            "authRequest": {
                "scopes": ["openid", "profile", "email", "urn:microsoft:userinfo"]
            },
            "loginFailedRoute": "/login-failed"
        },
        "interceptor": {
            "interactionType": "redirect",
            "protectedResourceMap": [
                ["https://rd-polaris.polarislibrary.com/Polaris.AdminServices/api/protected/", ["email"]]
            ]
        }
    }
}
```

Update Polaris Admin Server Location

- If you started from the template, replace [POLADMIN-SERVER-ADDR] with the AD FS server address.

  It should look like the following image when complete (your AD FS server address will be different):

```
{
    "apiUrlRoot": "https://rd-polaris.polarislibrary.com/polaris.adminservices/api/",
    "oauthEnabled": true,
    "msal": {
        "auth": {
          "clientId": "0a586b1e-eeb0-4c8a-8381-50e9cafec240",
          "authority": "https://dev-fs.polarislibrary.com/adfs/",
          "knownAuthorities": ["dev-fs.polarislibrary.com"],
          "redirectUri": "https://rd-polaris.polarislibrary.com/PolarisAdmin/oauth-success",
          "postLogoutRedirectUri": "https://rd-polaris.polarislibrary.com/PolarisAdmin",
          "protocolMode": "OIDC",
          "navigateToLoginRequestUrl": false
        },
        "cache": {
            "cacheLocation" : "localStorage",
            "storeAuthStateInCookie": false,
            "secureCookies": true
        },
        "guard": {
            "interactionType": "redirect",
            "authRequest": {
                "scopes": ["openid", "profile", "email", "urn:microsoft:userinfo"]
            },
            "loginFailedRoute": "/login-failed"
        },
        "interceptor": {
            "interactionType": "redirect",
            "protectedResourceMap": [
                ["https://rd-polaris.polarislibrary.com/Polaris.AdminServices/api/protected/", ["email"]]
            ]
        }
    }
}
```

Update Polaris Admin Services (API) Server Location

- If you started from the template, replace [POLADMINSVC-SERVER-ADDR] with the AD FS server address.

  It should look like the following image when complete (your AD FS server address will be different):

```
{
    "apiUrlRoot": "https://rd-polaris.polarislibrary.com/polaris.adminservices/api/",
    "oauthEnabled": true,
    "msal": {
        "auth": {
          "clientId": "0a586b1e-eeb0-4c8a-8381-50e9cafec240",
          "authority": "https://dev-fs.polarislibrary.com/adfs/",
          "knownAuthorities": ["dev-fs.polarislibrary.com"],
          "redirectUri": "https://rd-polaris.polarislibrary.com/PolarisAdmin/oauth-success",
          "postLogoutRedirectUri": "https://rd-polaris.polarislibrary.com/PolarisAdmin",
          "protocolMode": "OIDC",
          "navigateToLoginRequestUrl": false
        },
        "cache": {
            "cacheLocation" : "localStorage",
            "storeAuthStateInCookie": false,
            "secureCookies": true
        },
        "guard": {
            "interactionType": "redirect",
            "authRequest": {
                "scopes": ["openid", "profile", "email", "urn:microsoft:userinfo"]
            },
            "loginFailedRoute": "/login-failed"
        },
        "interceptor": {
            "interactionType": "redirect",
            "protectedResourceMap": [
                ["https://rd-polaris.polarislibrary.com/Polaris.AdminServices/api/protected/", ["email"]]
            ]
        }
    }
}
```

Set Up Polaris.ApplicationServices

To set up Polaris.ApplicationServices

Verify that OAuth Is Enabled

- Open C:\Program Files\Polaris\7.2\Polaris.ApplicationServices\appsettings.user.json and verify that OAuth.Enabled is set to true.

```
"OAuth": {
  "Enabled": true,
  "Authorities": [
    {
      "Name": "ADFS",
      "Authority": "https://dev-fs.polarislibrary.com/adfs/",
      "Audience": "microsoft:identityserver:3eb2a79f-db5a-4ba0-b22f-e7d16a616d4a",
      "MetaAddress": "https://dev-fs.polarislibrary.com/adfs/.well-known/openid-configuration",
      "RequireHttpsMetadata": true,
      "RequireSignedTokens": true,
      "ValidateIssuer": true,
      "ValidIssuers": [
        "https://dev-fs.polarislibrary.com/adfs",
        "http://dev-fs.polarislibrary.com/adfs/services/trust"
      ],
      "ValidateAudience": true,
      "ValidAudiences": [
        "3eb2a79f-db5a-4ba0-b22f-e7d16a616d4a",
        "microsoft:identityserver:3eb2a79f-db5a-4ba0-b22f-e7d16a616d4a"
      ],
      "ClaimTypeUPN": "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/upn"
    }
  ],
  "Swagger": {
    "ClientID": "3eb2a79f-db5a-4ba0-b22f-e7d16a616d4a",
    "AppName": "Polaris.ApplicationServices",
    "AuthorizationUrl": "https://dev-fs.polarislibrary.com/adfs/oauth2/authorize",
    "TokenUrl": "https://dev-fs.polarislibrary.com/adfs/oauth2/token",
    "RefreshTokenUrl": "https://dev-fs.polarislibrary.com/adfs/oauth2/token",
    "LogoutUrl": "https://dev-fs.polarislibrary.com/adfs/oauth2/logout"
  }
},
```

Update the AD FS Server Location

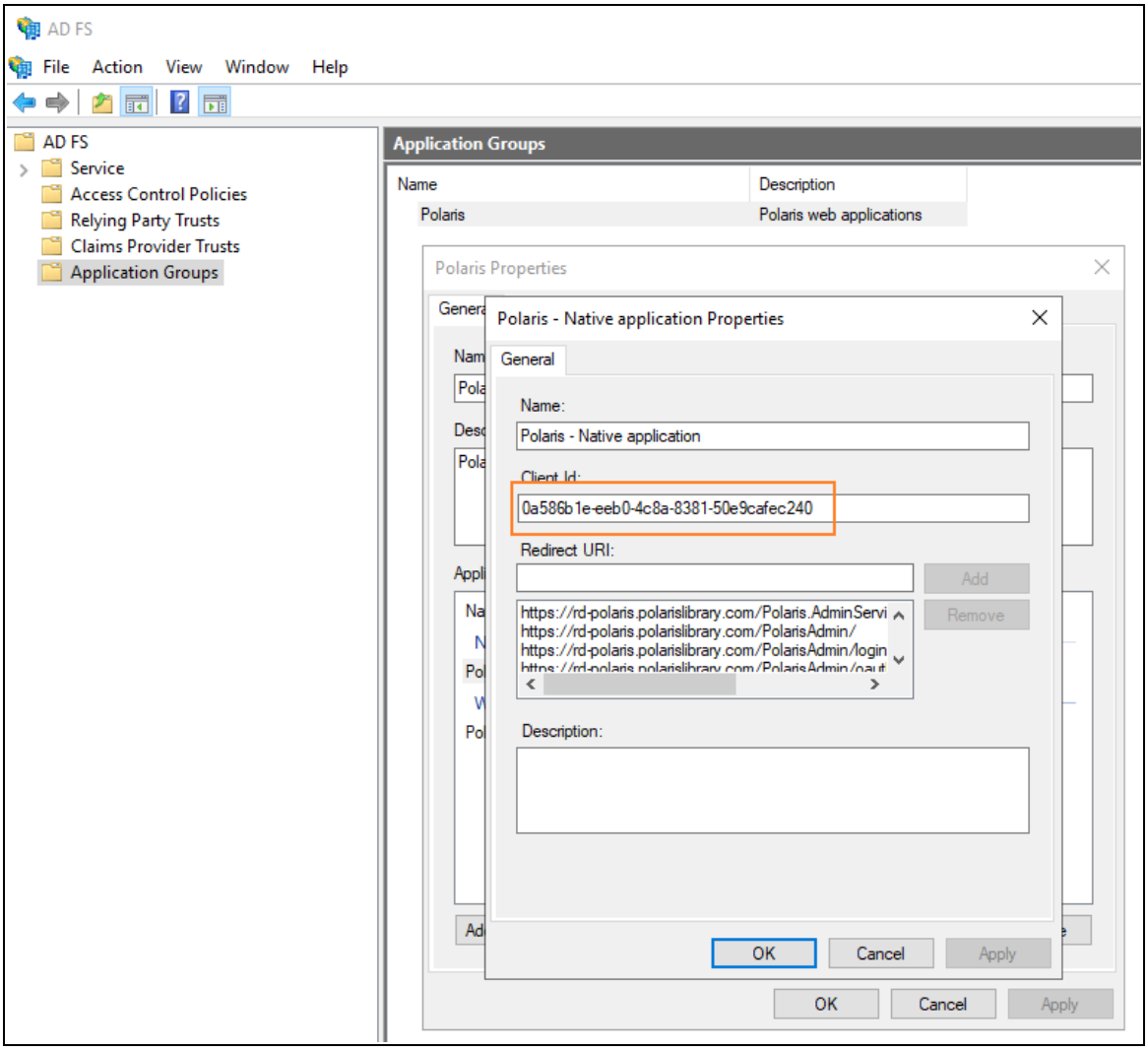
- If you started from the template, replace *adfs-server-address* with the AD FS server address.

  It should look like the following when complete (your AD FS server address will be different):

```
"OAuth": {
  "Enabled": true,
  "Authorities": [
    {
      "Name": "ADFS",
      "Authority": "https://dev-fs.polarislibrary.com/adfs/",
      "Audience": "microsoft:identityserver:3eb2a79f-db5a-4ba0-b22f-e7d16a616d4a",
      "MetaAddress": "https://dev-fs.polarislibrary.com/adfs/.well-known/openid-configuration",
      "RequireHttpsMetadata": true,
      "RequireSignedTokens": true,
      "ValidateIssuer": true,
      "ValidIssuers": [
        "https://dev-fs.polarislibrary.com/adfs",
        "http://dev-fs.polarislibrary.com/adfs/services/trust"
      ],
      "ValidateAudience": true,
      "ValidAudiences": [
        "3eb2a79f-db5a-4ba0-b22f-e7d16a616d4a",
        "microsoft:identityserver:3eb2a79f-db5a-4ba0-b22f-e7d16a616d4a"
      ],
      "ClaimTypeUPN": "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/upn"
    }
  ],
  "Swagger": {
    "ClientID": "3eb2a79f-db5a-4ba0-b22f-e7d16a616d4a",
    "AppName": "Polaris.ApplicationServices",
    "AuthorizationUrl": "https://dev-fs.polarislibrary.com/adfs/oauth2/authorize",
    "TokenUrl": "https://dev-fs.polarislibrary.com/adfs/oauth2/token",
    "RefreshTokenUrl": "https://dev-fs.polarislibrary.com/adfs/oauth2/token",
    "LogoutUrl": "https://dev-fs.polarislibrary.com/adfs/oauth2/logout"
  }
},
```

Update the Client ID

1. On the AD FS server, open the AD FS Management desktop application.



2. Copy the Client ID from the **Polaris - Native application Properties** dialog.
3. Paste the copied Client ID into the appsettings.user.json file.
4. If you started from the template, replace *client-id-configured-in-adfs* with the copied Client ID.

   It should look like the following when complete (your client ID will be different):

```
"OAuth": {
  "Enabled": true,
  "Authorities": [
    {
      "Name": "ADFS",
      "Authority": "https://dev-fs.polarislibrary.com/adfs/",
      "Audience": "microsoft:identityserver:3eb2a79f-db5a-4ba0-b22f-e7d16a616d4a",
      "MetaAddress": "https://dev-fs.polarislibrary.com/adfs/.well-known/openid-configuration",
      "RequireHttpsMetadata": true,
      "RequireSignedTokens": true,
      "ValidateIssuer": true,
      "ValidIssuers": [
        "https://dev-fs.polarislibrary.com/adfs",
        "http://dev-fs.polarislibrary.com/adfs/services/trust"
      ],
      "ValidateAudience": true,
      "ValidAudiences": [
        "3eb2a79f-db5a-4ba0-b22f-e7d16a616d4a",
        "microsoft:identityserver:3eb2a79f-db5a-4ba0-b22f-e7d16a616d4a"
      ],
      "ClaimTypeUPN": "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/upn"
    }
  ],
  "Swagger": {
    "ClientID": "3eb2a79f-db5a-4ba0-b22f-e7d16a616d4a",
    "AppName": "Polaris.ApplicationServices",
    "AuthorizationUrl": "https://dev-fs.polarislibrary.com/adfs/oauth2/authorize",
    "TokenUrl": "https://dev-fs.polarislibrary.com/adfs/oauth2/token",
    "RefreshTokenUrl": "https://dev-fs.polarislibrary.com/adfs/oauth2/token",
    "LogoutUrl": "https://dev-fs.polarislibrary.com/adfs/oauth2/logout"
  }
},
```

Set Up LeapWebApp

To set up LeapWebApp

Verify that OAuth Is Enabled

- Open C:\Program Files\Polaris\7.2\LeapWebApp\appsettings.user.json and verify that OAuthEnabled is set to true.

```
"OAuthEnabled": true,
"OAuth": {
  "Authority": "https://dev-fs.polarislibrary.com/adfs/",
  "ClientSecret": null,
  "MetadataAddress": "https://dev-fs.polarislibrary.com/adfs/.well-known/openid-configuration",
  "KnownAuthorities": [ "dev-fs.polarislibrary.com" ],
  "CallbackPath": "/signin-oidc",
  "SignedOutCallbackPath": "/signout-callback-oidc",
  "SignedOutRedirectUri": "/login",
  "RemoteAuthenticationTimeout": 15,
  "RemoteFailureRedirectUri": "/leapwebapp/logout",
  "ResponseMode": "form_post",
  "ResponseType": "code id_token token",
  "UsePkce": false
  "UsePkce": false
},
```

Update the AD FS Server Location

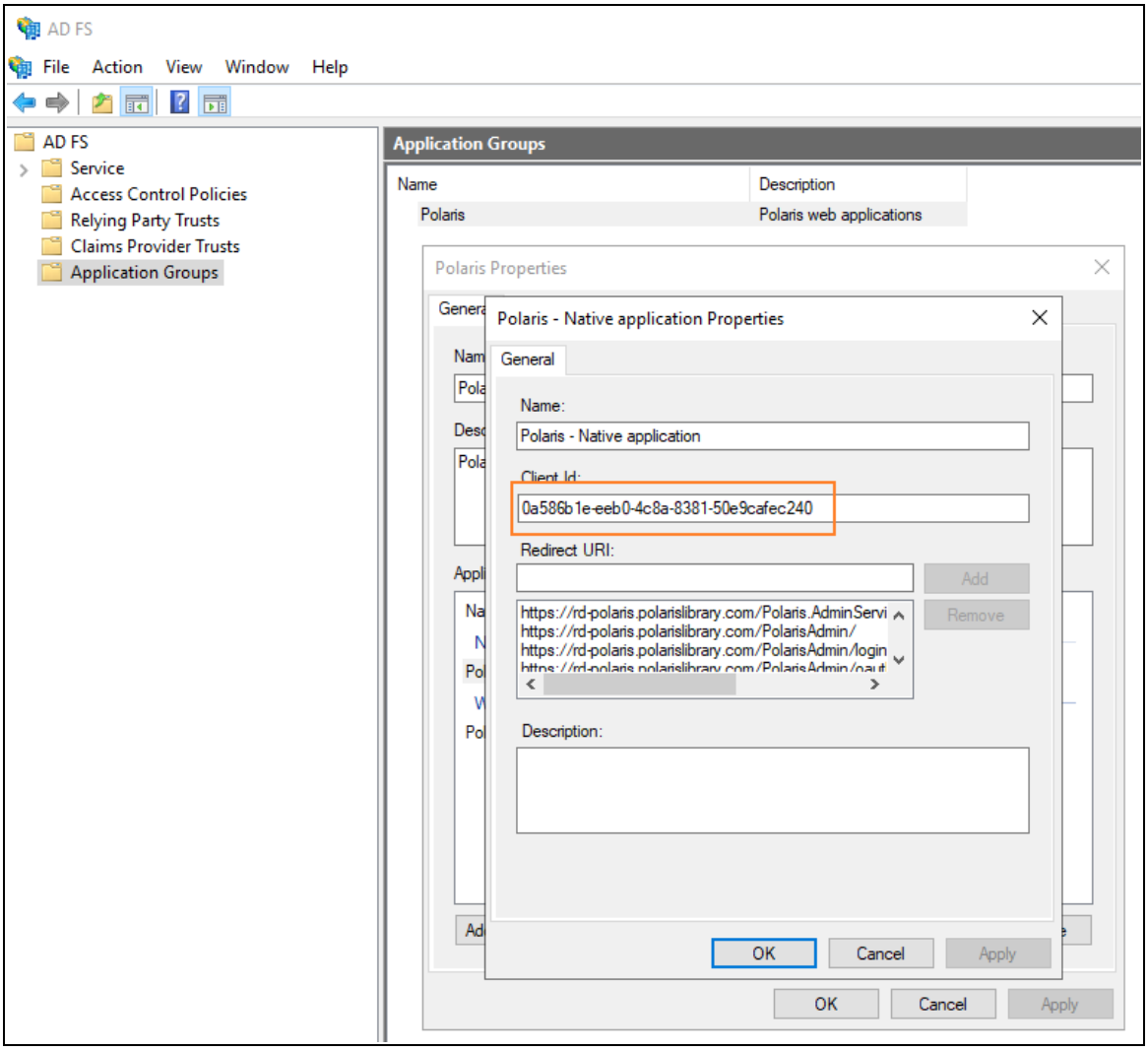
- If you started from the template, replace [adfs-server-address] with the AD FS server address.

  It should look like the following when complete (your AD FS server address will be different):

```
"OAuthEnabled": true,
"OAuth": {
  "Authority": "https://dev-fs.polarislibrary.com/adfs/",
  "ClientId": "0a586b1e-eeb0-4c8a-8381-50e9cafec240",
  "ClientSecret": null,
  "MetadataAddress": "https://dev-fs.polarislibrary.com/adfs/.well-known/openid-configuration",
  "KnownAuthorities": [ "dev-fs.polarislibrary.com" ],
  "CallbackPath": "/signin-oidc",
  "SignedOutCallbackPath": "/signout-callback-oidc",
  "SignedOutRedirectUri": "/login",
  "RemoteAuthenticationTimeout": 15,
  "RemoteFailureRedirectUri": "/leapwebapp/logout",
  "ResponseMode": "form_post",
  "ResponseType": "code id_token token",
  "UsePkce": false
},
```

Update the Client ID

1. On the AD FS server, open the AD FS Management desktop application.



2. Copy the Client ID from the **Polaris - Native application Properties** dialog.
3. Paste the copied Client ID into the appsettings.user.json file.
4. If you started from the template, replace the *client-id-configured-in-adfs* with the copied Client ID.

   It should look like the following when complete (your client ID will be different):

```
"OAuthEnabled": true,
"OAuth": {
  "Authority": "https://dev-fs.polarislibrary.com/adfs/",
  "ClientId": "0a586b1e-eeb0-4c8a-8381-50e9cafec240",
  "ClientSecret": null,
  "MetadataAddress": "https://dev-fs.polarislibrary.com/adfs/.well-known/openid-configuration",
  "KnownAuthorities": [ "dev-fs.polarislibrary.com" ],
  "CallbackPath": "/signin-oidc",
  "SignedOutCallbackPath": "/signout-callback-oidc",
  "SignedOutRedirectUri": "/login",
  "RemoteAuthenticationTimeout": 15,
  "RemoteFailureRedirectUri": "/leapwebapp/logout",
  "ResponseMode": "form_post",
  "ResponseType": "code id_token token",
  "UsePkce": false
},
```

Enable Session Storage for LeapWebApp

Enable session storage for the best user experience when using OAuth and OIDC.

Microsoft SQL Server Express 2019 (or a newer version) must be installed to use session storage. You install SQL Server Express separately. It is not part of the Leap installation.

To enable session storage

- Open C:\Program Files\Polaris\7.2\LeapWebApp\appsettings.user.json and set SessionStore.Enabled to true.

```
"SessionStore": {
  "Enabled": true,
  "ConnectionString": "Data Source=.\\Polaris; Initial Catalog=PolarisCache; Integrated Security=True;",
  "SessionTimeoutMinutes": "1440",
  "SchemaName": "dbo",
  "TableName": "Sessions"
},
```

Add a URL Rewrite Rule for LeapWebApp

Adding a URL rewrite rule redirects incoming URLs to the correct address for the LeapWebApp. This must be done manually, since the library may already use other URL rewrite rules.

The Microsoft IIS URL Rewrite 2.1 extension is required to add a URL rewrite rule. For more information, see https://www.iis.net/downloads/microsoft/url-rewrite.

To add a URL rewrite rule

1. Open the root IIS web.config file, found in the following location:

   C:\inetpub\wwwroot\web.config

2. Add a rewrite rule to the **system.webServer** node.

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
    <system.webServer>
        <rewrite>
            <rules>
                <rule name="UrlToLowercase" stopProcessing="true">
                    <match url="(.*)" ignoreCase="true" />
                        <action type="Redirect" url="https://{HTTP_HOST}{ToLower:{PATH_INFO}}" redirectType="Found" appendQueryString="true" />
                        <conditions>
                            <add input="{PATH_INFO}" pattern="^/LeapWebApp(.*)|^/Leapwebapp(.*)|^/LEAPWEBAPP(.*)" ignoreCase="false" />
                        </conditions>
                    </rule>
            </rules>
        </rewrite>
    </system.webServer>
</configuration>
```

> **Note:**
> For sample rewrite rule text that you can copy and paste, see Sample Rewrite Rule Text.

   In the example above, if the incoming URL includes a path that contains any of the following, the rewrite rule redirects to /leapwebapp:

   - /LeapWebApp
   - /Leapwebapp
   - /LEAPWEBAPP

3. Save the web.config file.

> **Note:**
> When registering redirect URIs for LeapWebApp in AD FS, the URIs should

be lowercase. For example:

- https://rd-polaris.polarislibrary.com/leapwebapp/signin-oidc
- https://rd-polaris.polarislibrary.com/leapwebapp/signin-override-oidc
- https://rd-polaris.polarislibrary.com/leapwebapp/signout-callback-oidc

Sample Rewrite Rule Text

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
   <system.webServer>
      <rewrite>
         <rules>
            <rule name="UrlToLowercase" stopProcessing="true">
               <match url="(.*)" ignoreCase="true" />
                  <action type="Redirect" url="https://{HTTP_HOST}
                  {ToLower:{PATH_INFO}}" redirectType="Found"
                  appendQueryString="true" />
                  <conditions>
                     <add input="{PATH_INFO}" pattern="^/LeapWebApp
                     (.*)|^/Leapwebapp(.*)|^/LEAPWEBAPP(.*)"
                     ignoreCase="false" />
                  </conditions>
            </rule>
         </rules>
      </rewrite>
   </system.webServer>
</configuration>
```

Additional URL Rewrite Resources

See Microsoft's URL Rewrite Module Configuration Reference for additional information:

- https://docs.microsoft.com/en-us/iis/extensions/url-rewrite-module/url-rewrite-module-configuration-reference

Customize the AD FS Pages

Use the following resources to customize AD FS pages:

- https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/dn280950(v=ws.11)
    - Get-AdfsGlobalWebContent
    - Set-AdfsGlobalWebContent

      Examples:
      Customize the examples below to suit your library's needs.

      ```
      PS C:\Windows\system32> Set-AdfsGlobalWebContent -SignOutPageDescriptionText "You have successfully signed out.<br>If you have been directed here immediately after signing in, your session may have timed out."

      PS C:\Windows\system32> Set-AdfsWebTheme -TargetName default -Logo @{path="c:\ADFS Custom\leap_logo.png"}

      PS C:\Windows\system32> Set-AdfsGlobalWebContent -CompanyName "Polaris R&D"
      ```

- Advanced customization:
    - https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/dn636121(v=ws.11)

Change the Access Token Lifetime

The default token lifetime for both access and ID tokens is 60 minutes. Execute the following command to increase the expiration time to 24 hours:

```
Set-AdfsWebApiApplication -TokenLifetime 1440 -TargetIdentifier
"0a586b1e-eeb0-4c8a-8381-50e9cafec240"
```

> **Note:**
> Replace *TargetIdentifier* with the Polaris Application Group native application client ID.

Bind a New SSL Certificate

If your web server certificate expires, use the instructions below to bind a new SSL certificate.

To bind a new SSL certificate

1. Install the certificate using Certificates Management.
2. Set the service communications certificate using the AD FS Management Console:
    a. Expand the Services folder.
    b. Select a new certificate.
    c. Restart the AD FS service.
3. Attach the certificate to AD FS using PowerShell:
    a. Get the certificate's thumbprint by viewing the certificate.

    ```
    c:\> Set-AdfsSslCertificate –Thumbprint
    e8fd5016542796214e94f72d76095f9fc587c731
    ```

    b. Restart the AD FS service.

Troubleshoot

Force a logout

- https://*AD FS server address*/adfs/oauth2/logout

  Note:
  Replace *AD FS server address* with your library's AD FS server address.

AD FS in one-way trust

Problem: Only local accounts are authenticating

Solution: Make sure the account running the AD FS service is a parent domain account and not a local account.

Receiving "User is not a valid Polaris user." error

- Check the setting Polaris.OAuth.ValidIssuer in the Polaris.AdminServices appsettings.user.json file.

  Example value: http://*AD FS server address*/adfs/services/trust

  Note:
  Replace *AD FS server address* with your library's AD FS server address.

- Verify a domain is attached to AD user accounts so the UPN claim can be added to the ID token's claims.

  The UPN claim should look like user@mydomain.com.

Troubleshoot Redirect URIs

Redirect URIs are case-sensitive.