**Amazon Cognito**

Amazon Cognito is a service that helps you manage user authentication, authorization, and sign-in for your web and mobile applications. It provides several features to achieve this, including:

* **User Pools:** For managing user registration, sign-in, and account recovery.
* **Identity Pools:** For providing temporary AWS credentials to users to access other AWS services.
* **User Migration:** For migrating users from existing user directories to Cognito.
* **Social Sign-in:** For allowing users to sign in with their existing social media accounts (e.g., Facebook, Google).

**Key components of Cognito:**

* **User Pools:** Manage user registration, authentication, and account recovery directly in your application.
* **Identity Pools:** Grant temporary access to AWS resources for authenticated users from various sources, including User Pools, social identity providers, and your own identity system.

**Cognito User Pools**

**Purpose:** User pools manage user registration, sign-in, and account recovery for your applications. They provide features like:

* **Signup and Sign-in:** Users can create accounts and sign in using email, username, or social media credentials.
* **MFA (Multi-Factor Authentication):** Enhances security by requiring additional verification factors during sign-in.
* **Password Management:** Users can reset their passwords and manage their account settings.

**Use Cases:**

* Building secure web and mobile applications with user authentication.
* Implementing social sign-in for user convenience.
* Managing user accounts and providing self-service options.

**Cognito Identity Pools**

**Purpose:** Identity pools are used to authorize users to access other AWS services. They provide features like:

* **Federated Identities:** Allows users to sign in using various identity providers (e.g., User Pools, social media, SAML).
* **Temporary AWS Credentials:** Grants temporary access to resources like S3 or DynamoDB based on user identity.
* **Role-based Access Control (RBAC):** Defines permissions for different user groups within the identity pool.

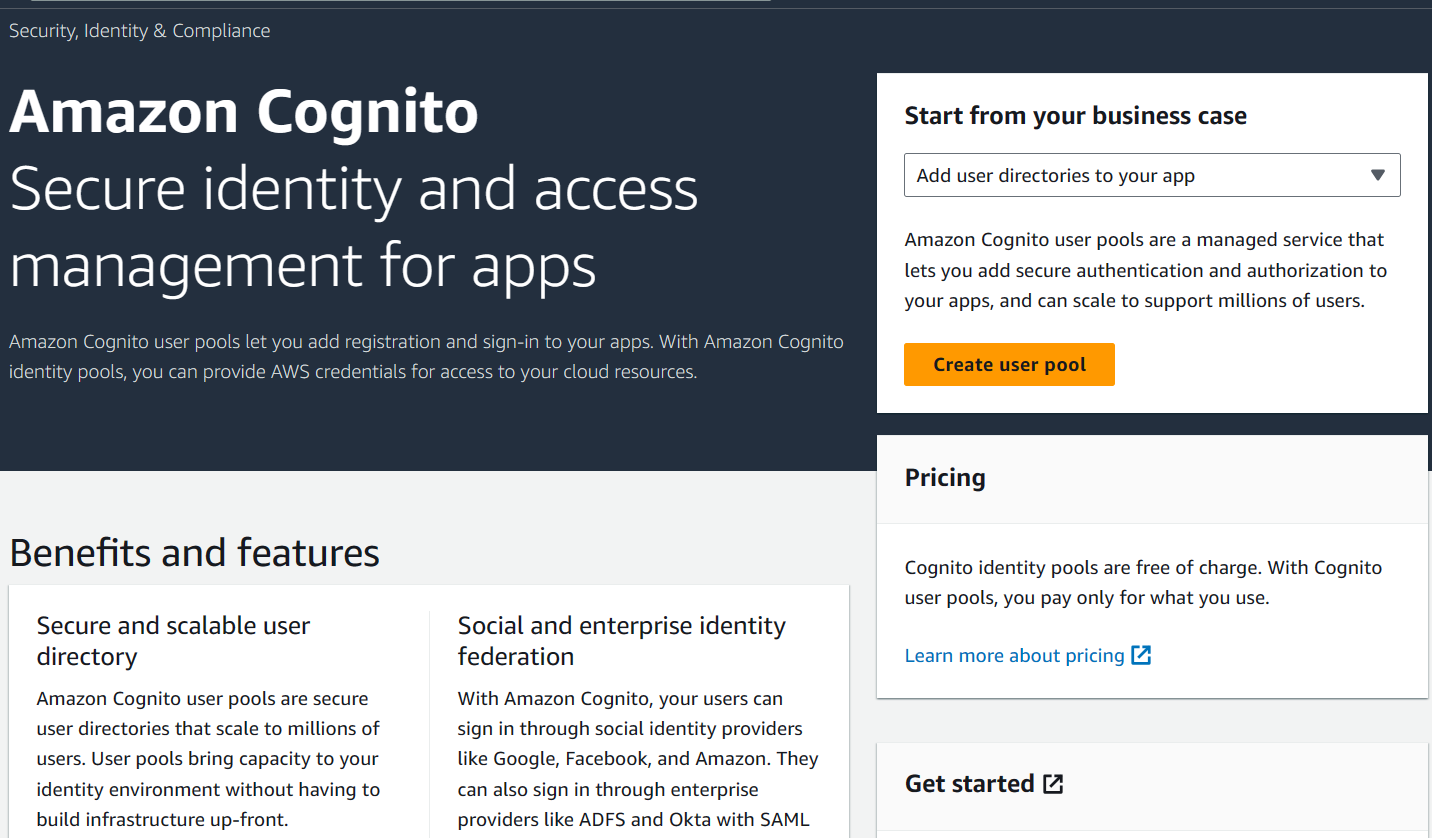
**Use Cases:**

* Granting fine-grained access to AWS resources based on user identity.
* Enabling secure access to your AWS resources from your applications.
* Implementing RBAC to control user permissions for different AWS services.

**Cognito User Pools vs. Identity Pools**

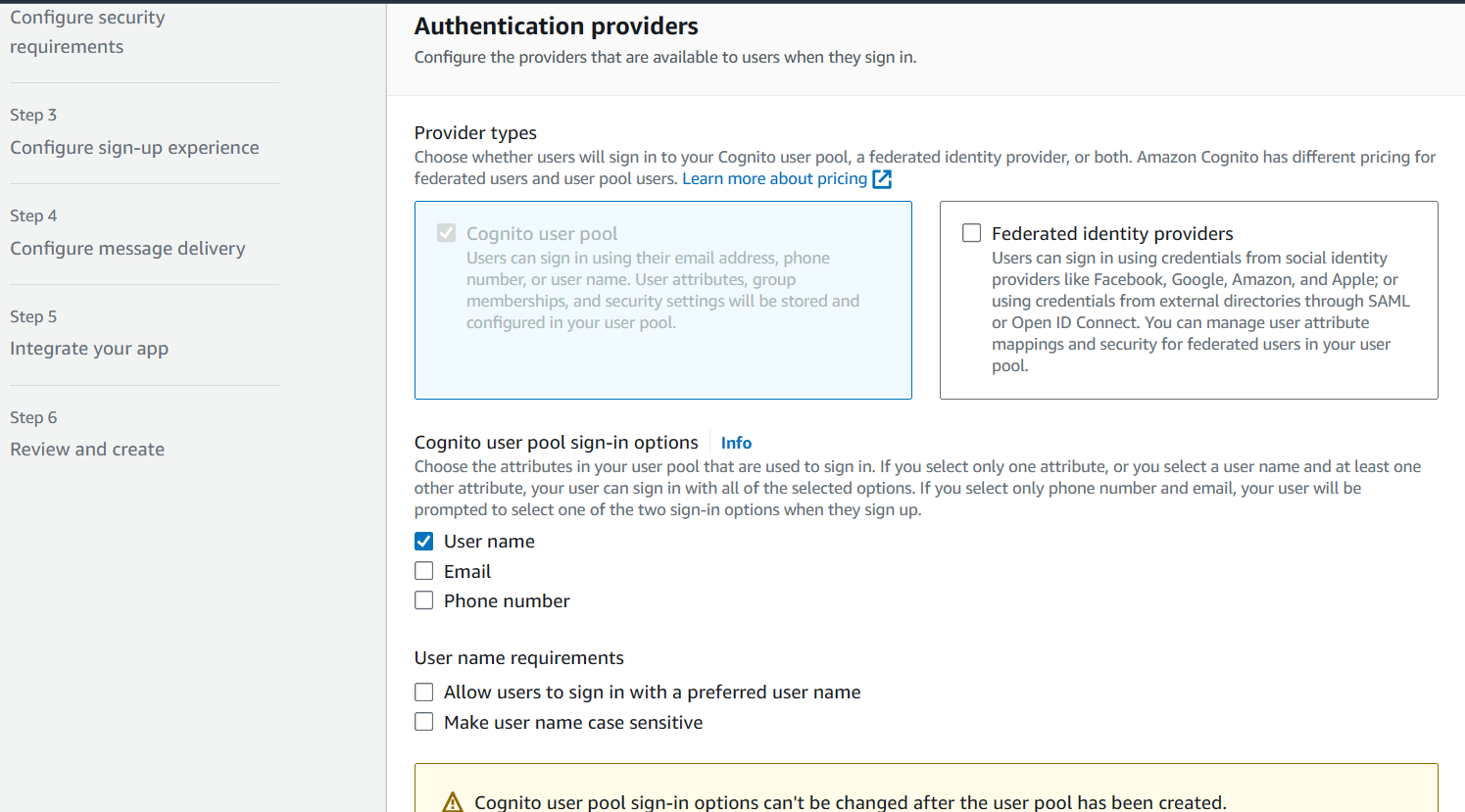
|  |  |  |
| --- | --- | --- |
| Feature | User Pools | Identity Pools |
| **Purpose** | User registration, authentication, and data management | Grant temporary access to AWS resources |
| **Authentication Source** | User Pools, social logins, or custom | User Pools, social logins, SAML providers, custom |
| **Use Cases** | User accounts, sign-in, password management | Accessing other AWS services, RBAC |

**Hands-on: Creating an Amazon Cognito User Pool**



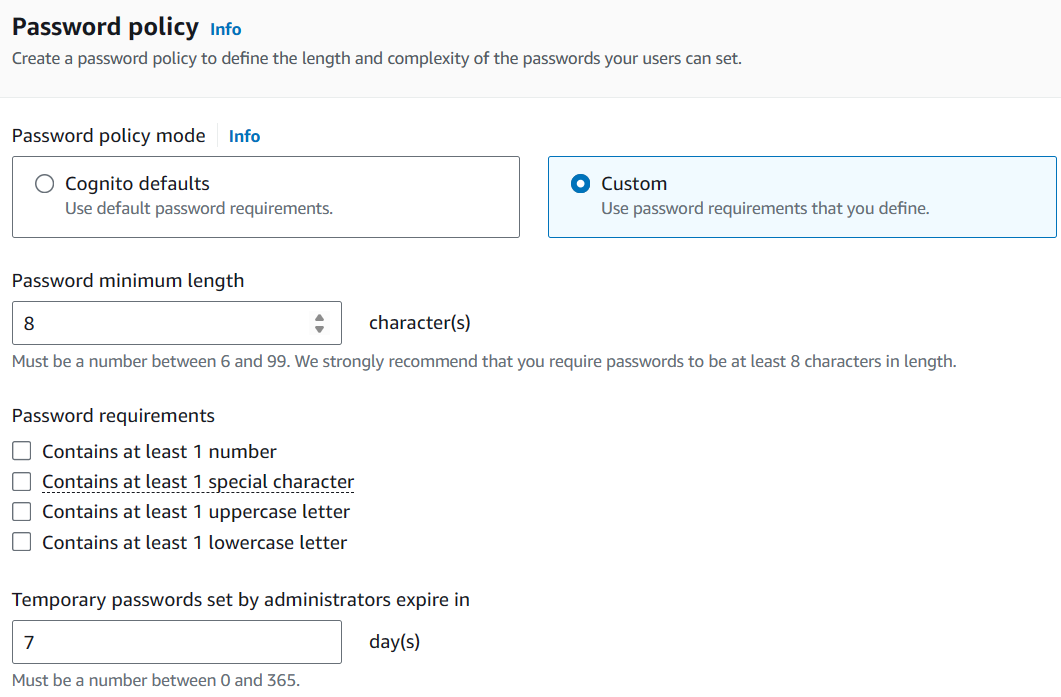
**1. Sign-in Experience:**

* Choose **Cognito User Sign-in** option.
* Select **Username** as the username format.

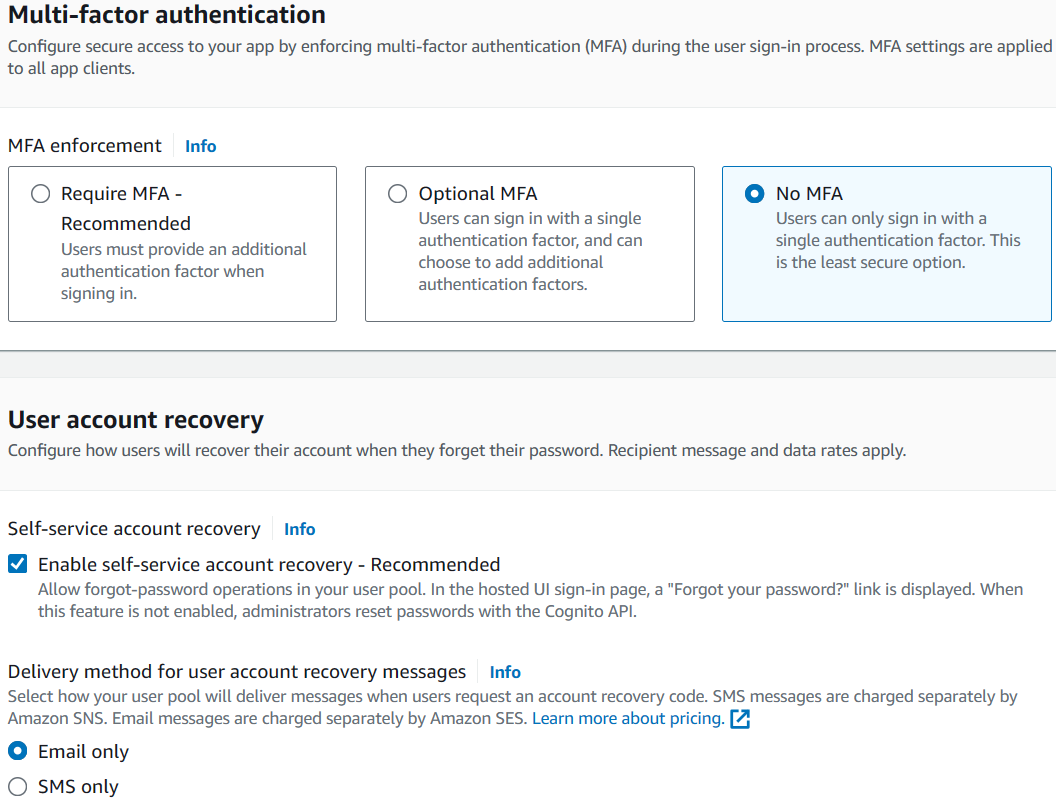


**2. Security Requirements:**

* Set **Custom** password policy and define desired length and complexity requirements.

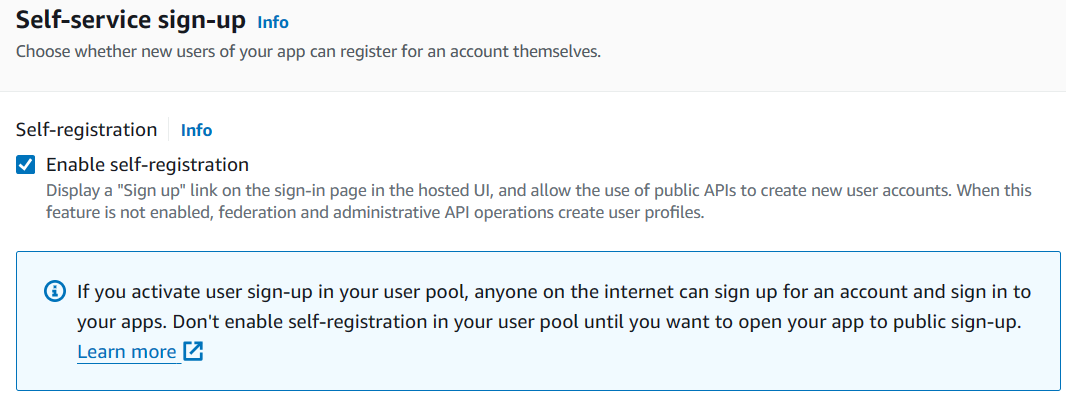


* Disable **MFA** for user sign-in and enable self-service account recovery.

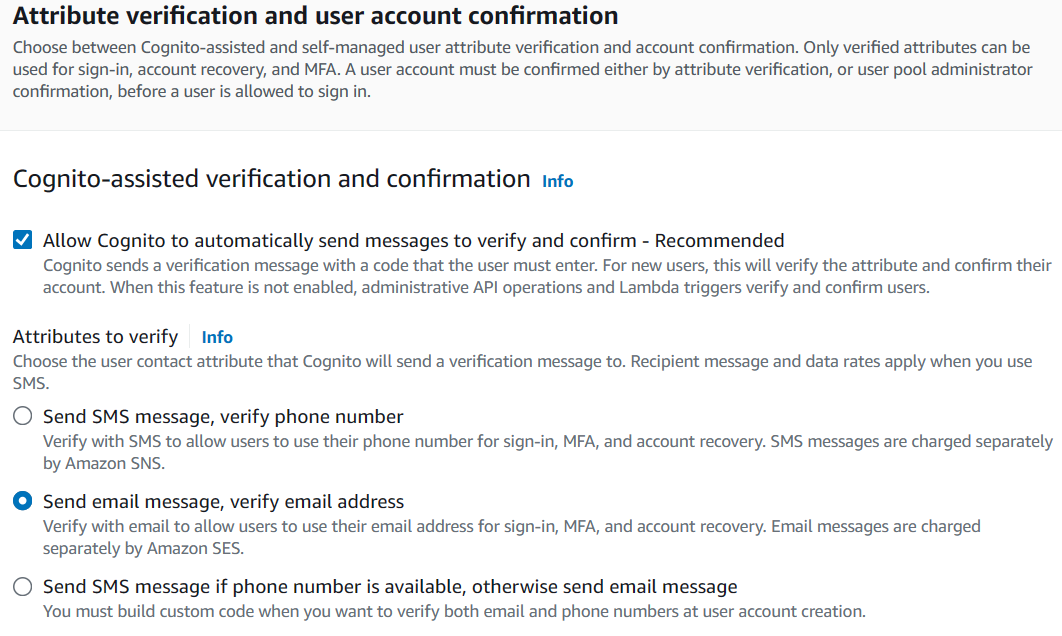


**3. Sign-up Experience:**

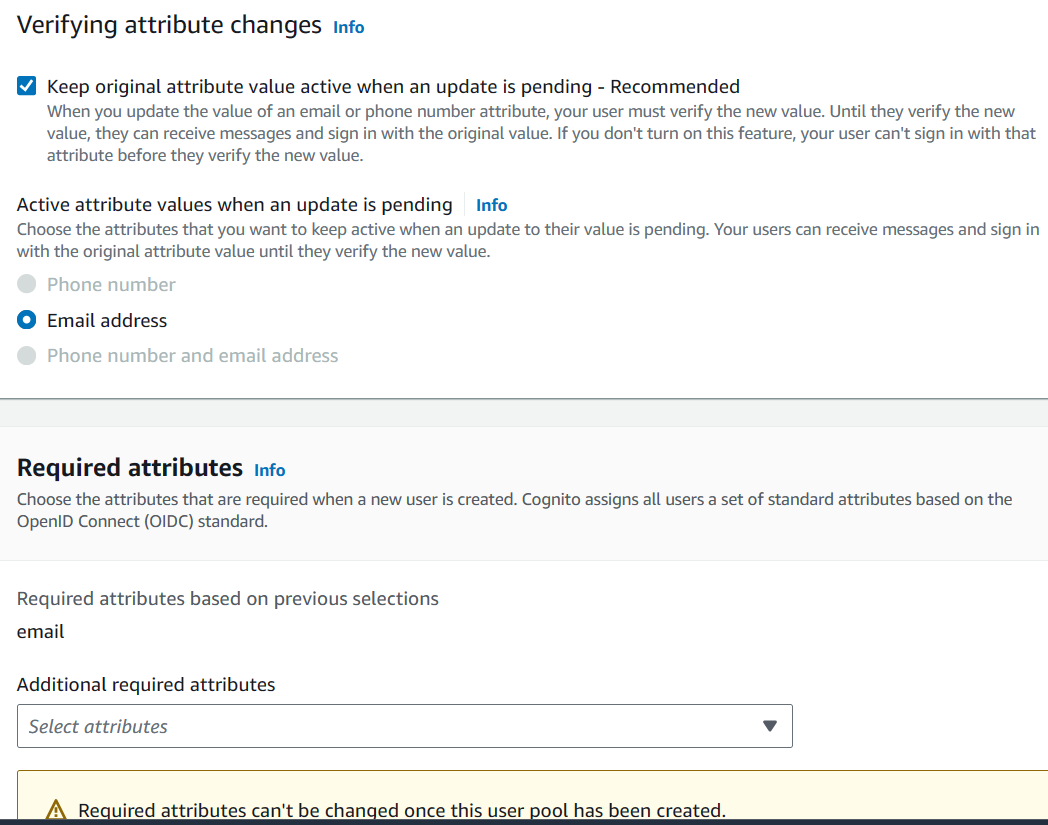
* Enable **Self-service sign-up**.



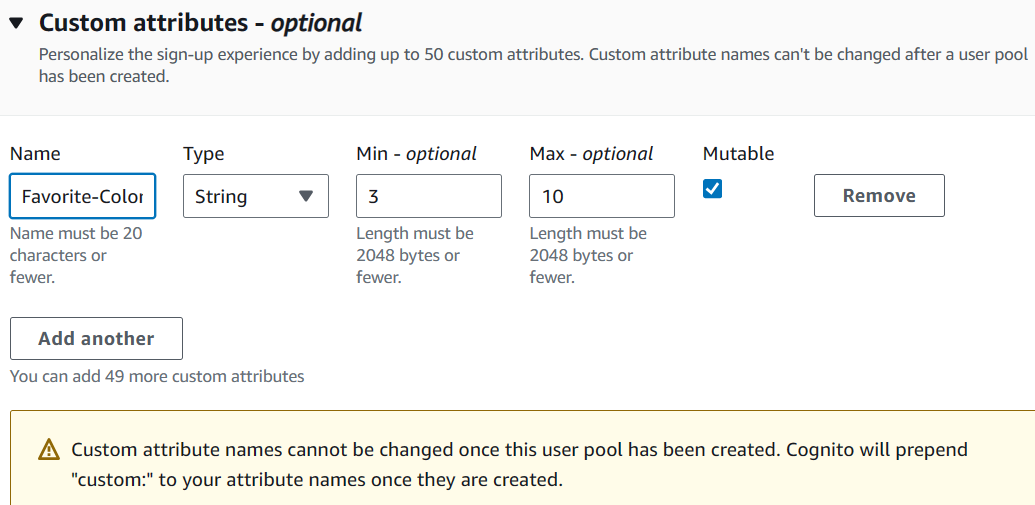
* Keep default settings for **Attribute verification** and **User account confirmation**.



* Set **Required attributes**:
  + **Email** (default)

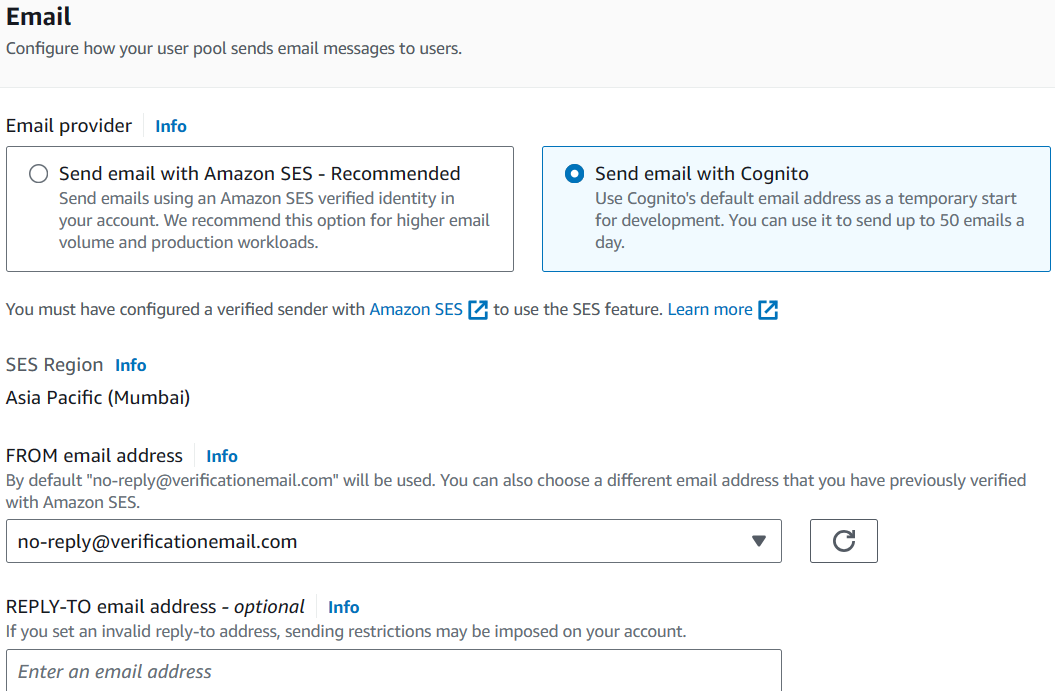


* Add a **Custom attribute**:
  + Name: **Favorite Color**
  + Type: **String**
  + Min/Max length: Define desired range



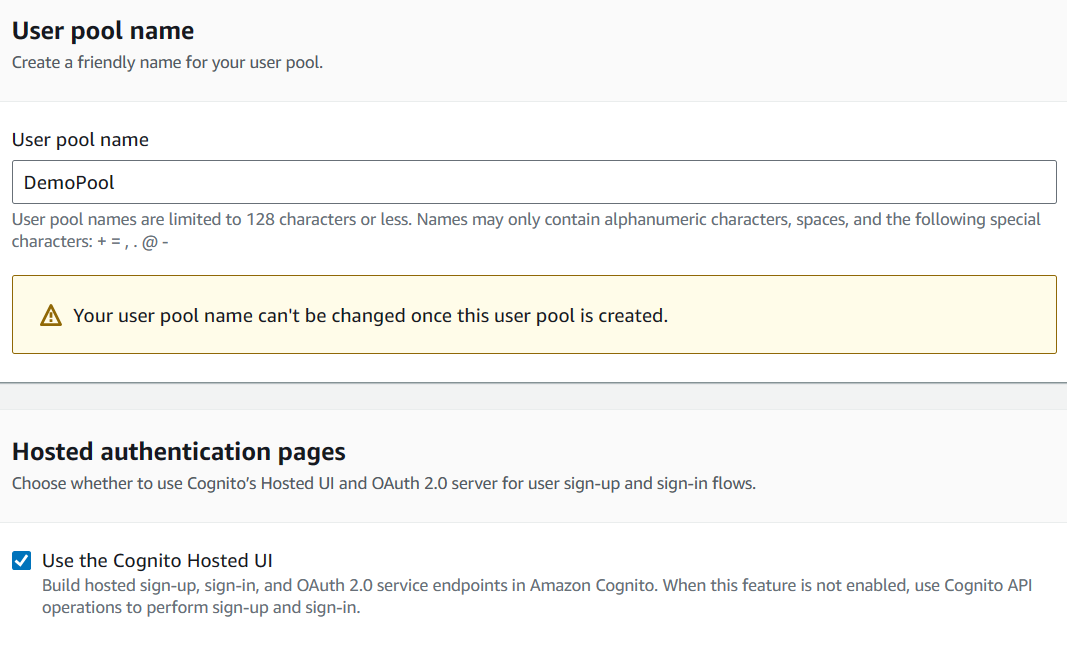
**4. Message Delivery:**

* Choose **Send email with Cognito**.

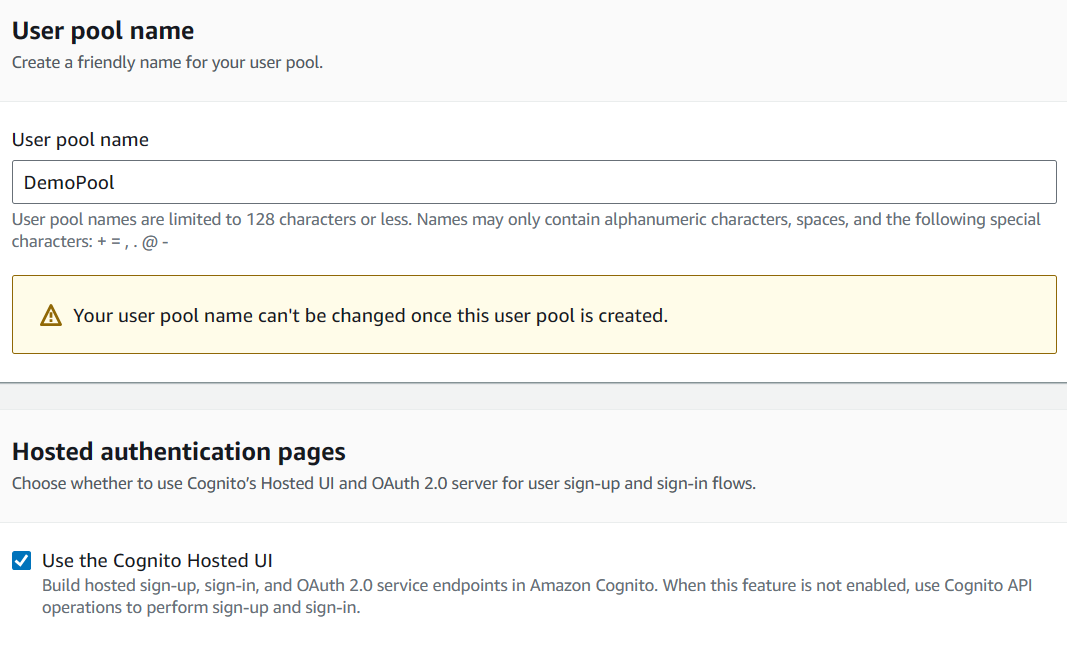


**5. Integrate your App:**

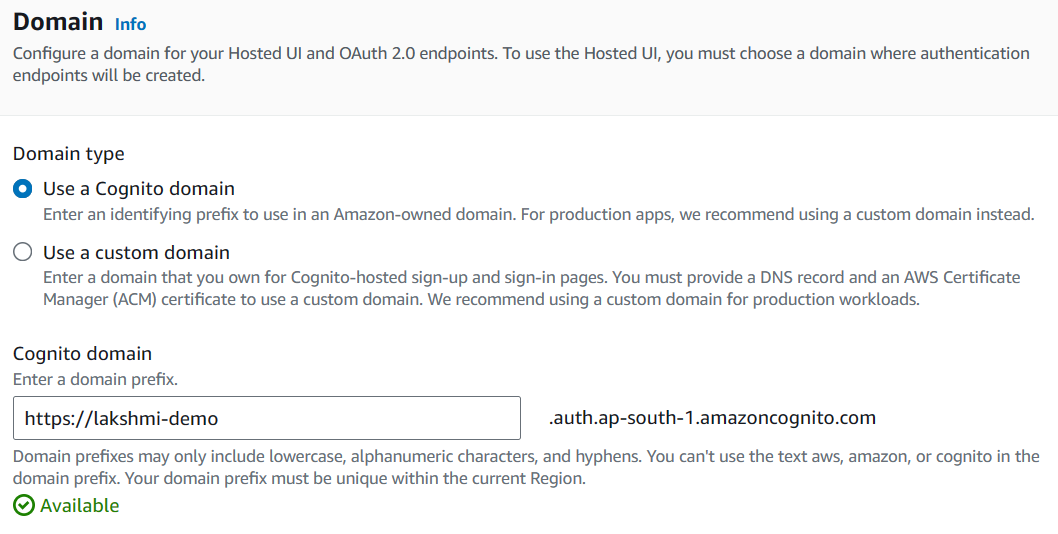
* **User pool name:** DemoPool



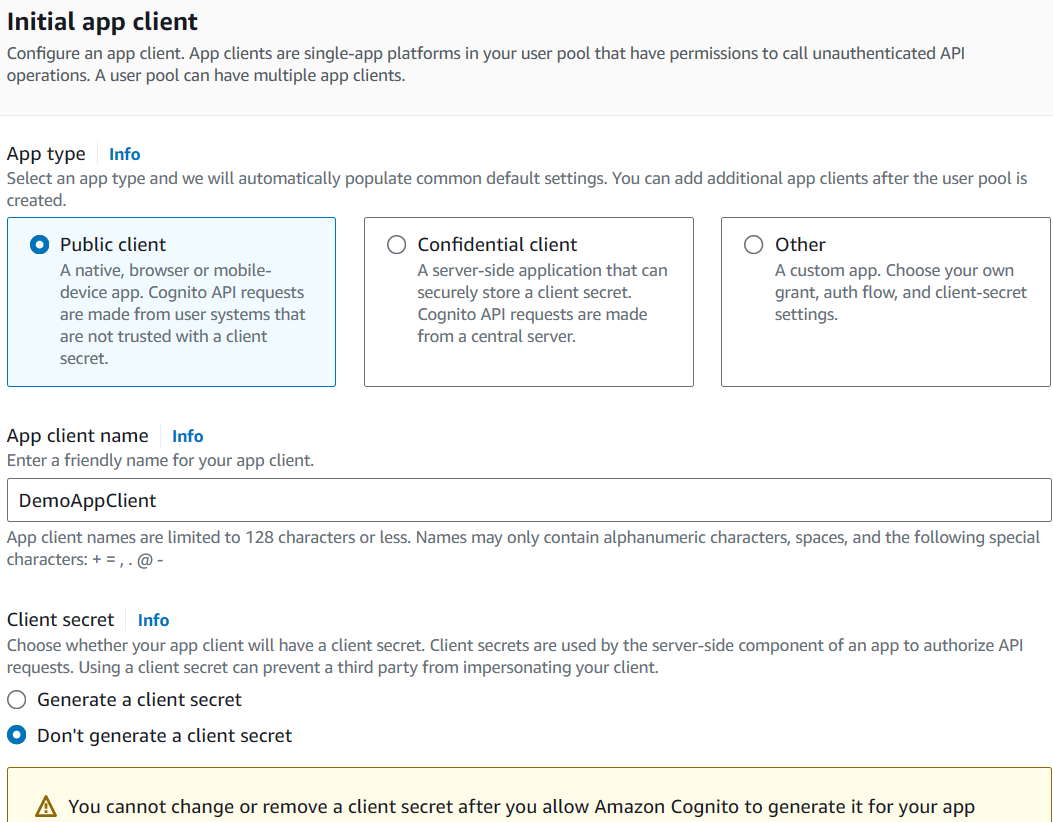
* **Use Cognito Hosted UI:** Checked

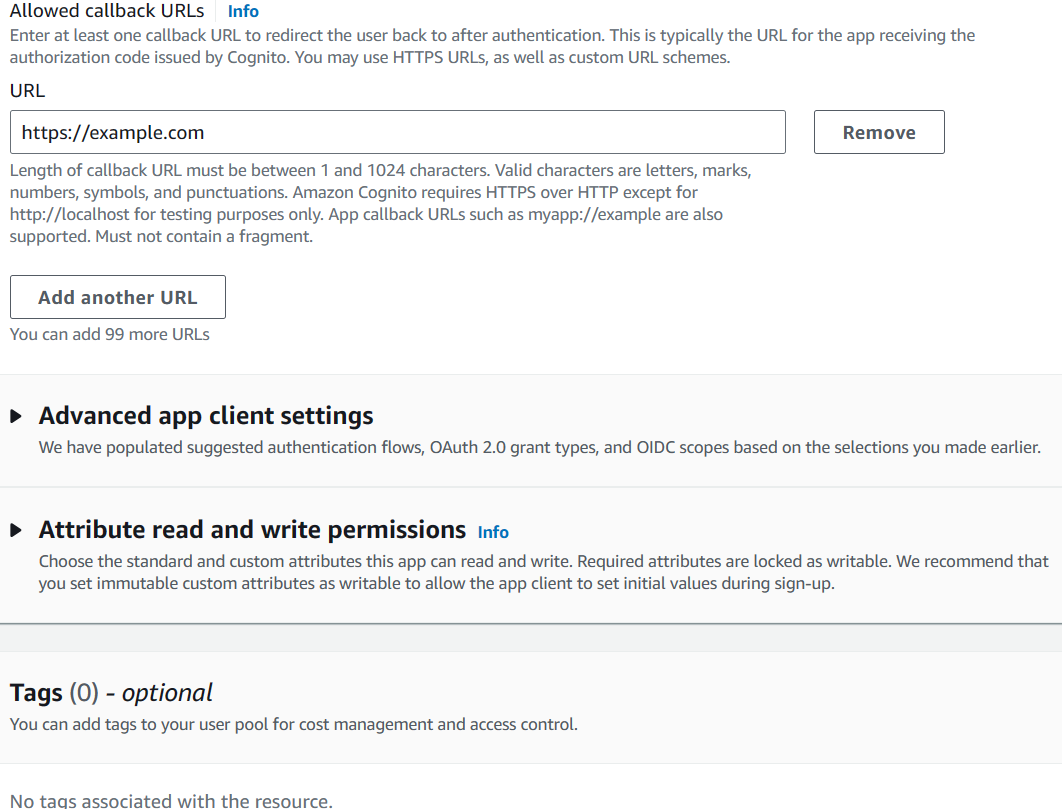


* **Domain type:** Use Cognito domain (<https://lakshmi-demo>)



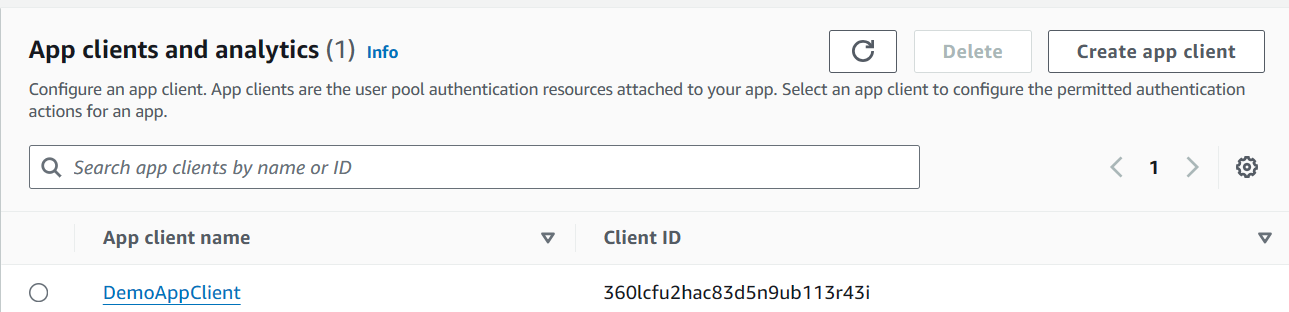
* **App client:**
  + **Name:** DemoAppClient
  + **Type:** Public
  + **Client secret:** Not generated
  + **Callback URL:** example.com

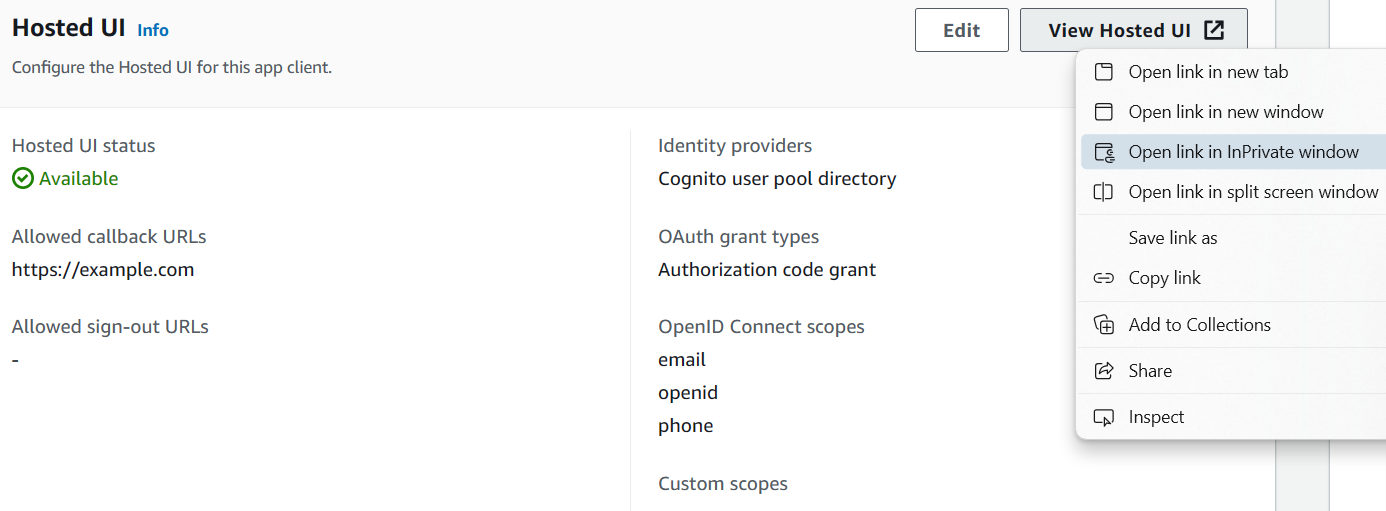




**6. Review and Create:**

* Review your configuration and create the user pool

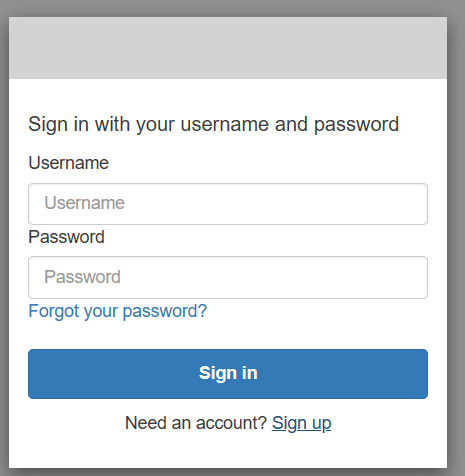




**Exploring User Management in Amazon Cognito DemoPool**

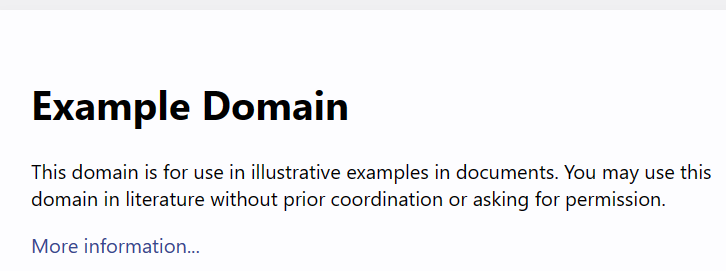
**1. Accessing the Hosted UI:**

* Navigate to **App integration** and locate **DemoAppClients**.
* Click on **DemoAppClients** and then **View hosted UI**.



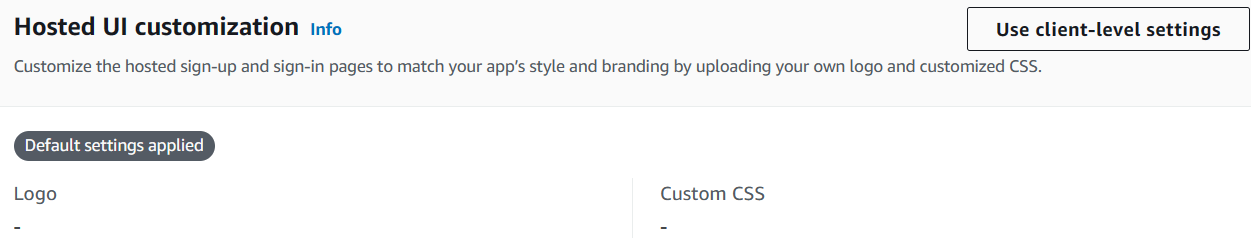
**2. User Creation and Confirmation:**

* Create a new account using the sign-up form (username, email, password).
* Confirm your account by entering the verification code received via email.
* Sign in using the confirmed account. You will be redirected to the callback URL (example.com).

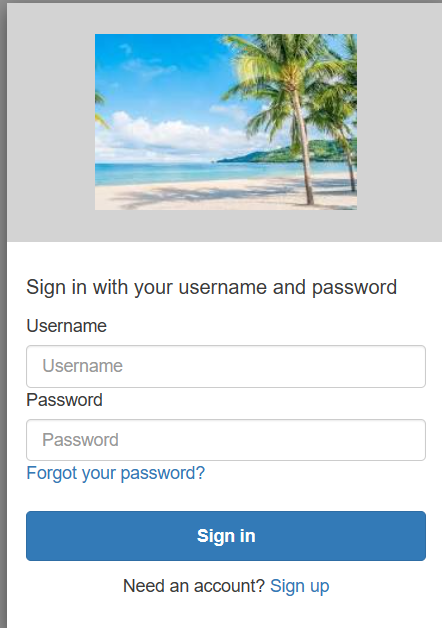


**3. Customizing the Hosted UI:**

* Access **Client-level settings** and upload a logo.

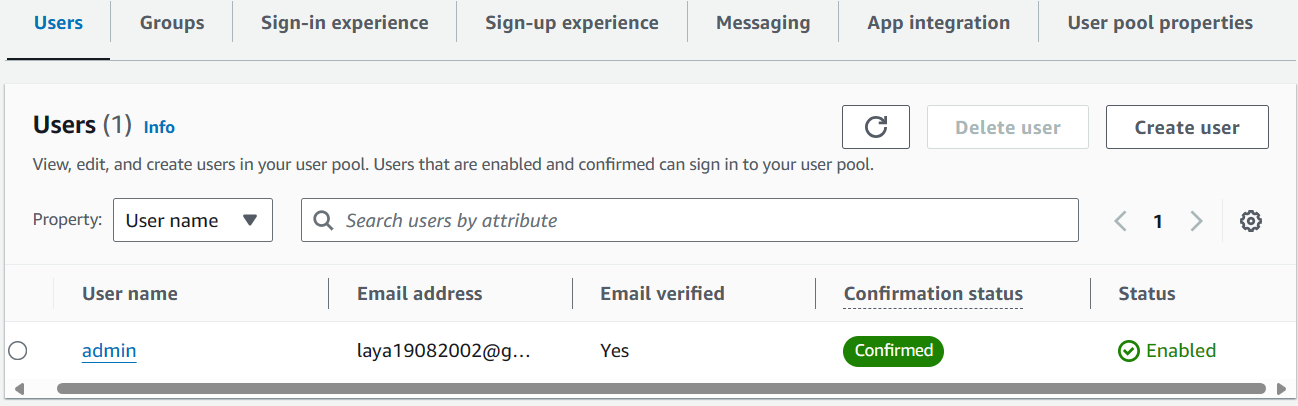


* Use a private window to view the updated login UI with your logo.

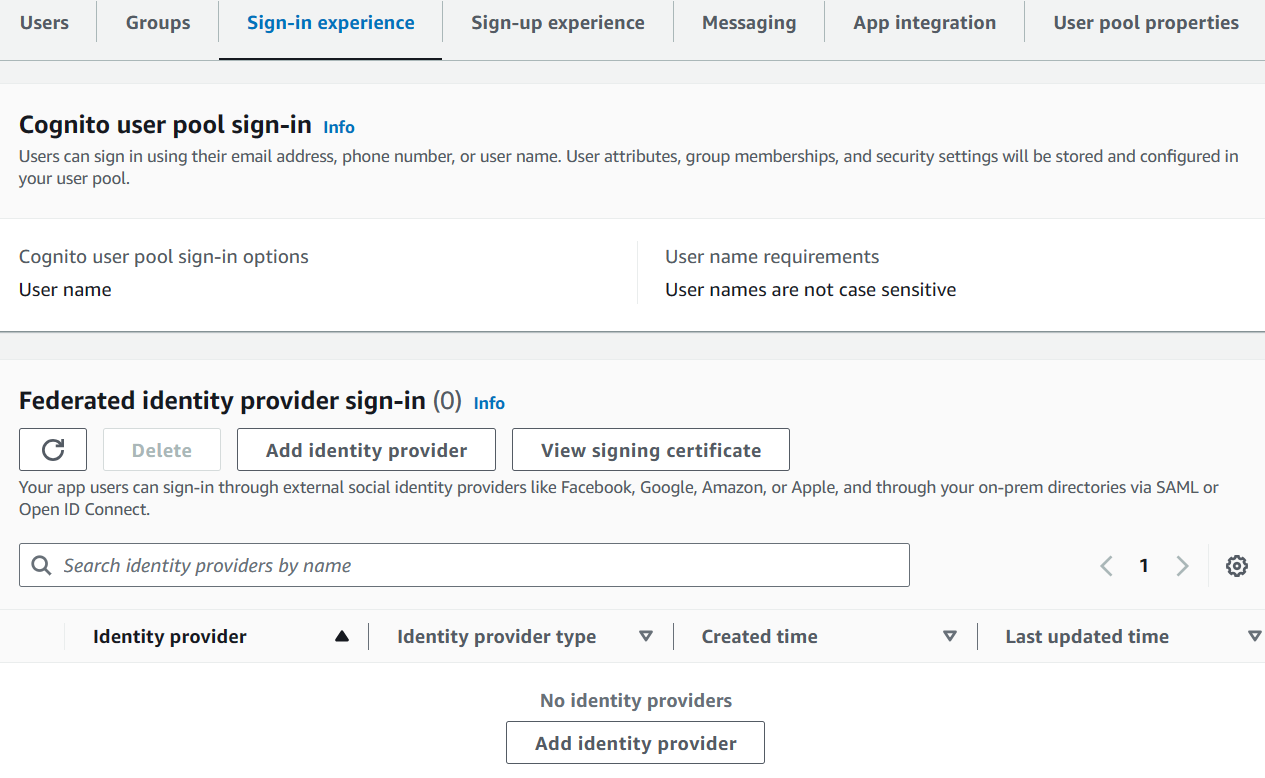


**4. Additional Features:**

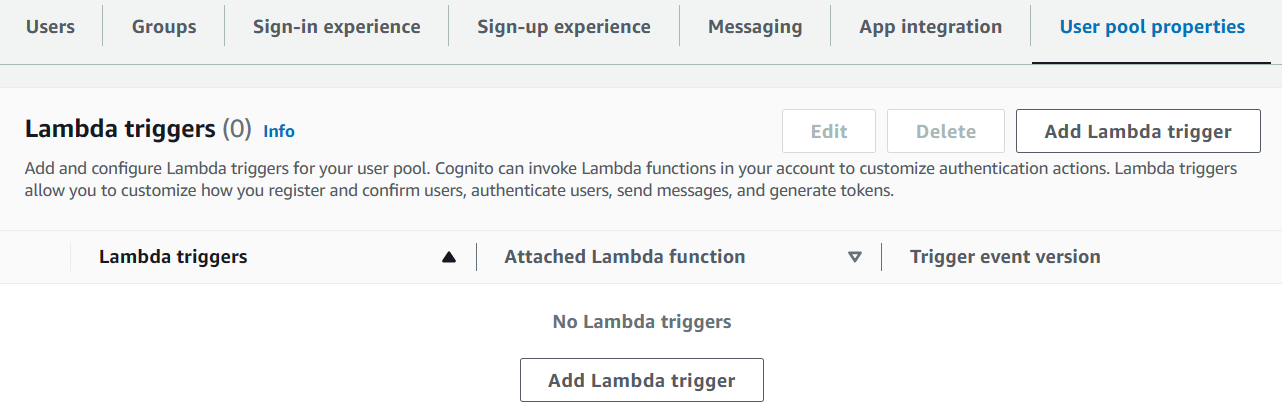
* **User Management:**
  + View the created user under **Users** with verified email and confirmed status.
  + Manually create and invite users by entering username, email, etc.
  + Group users for enhanced security.



* **Federated Identity Providers:**
  + Integrate login options like Google, Facebook, Amazon, Apple, SAML, or OIDC.



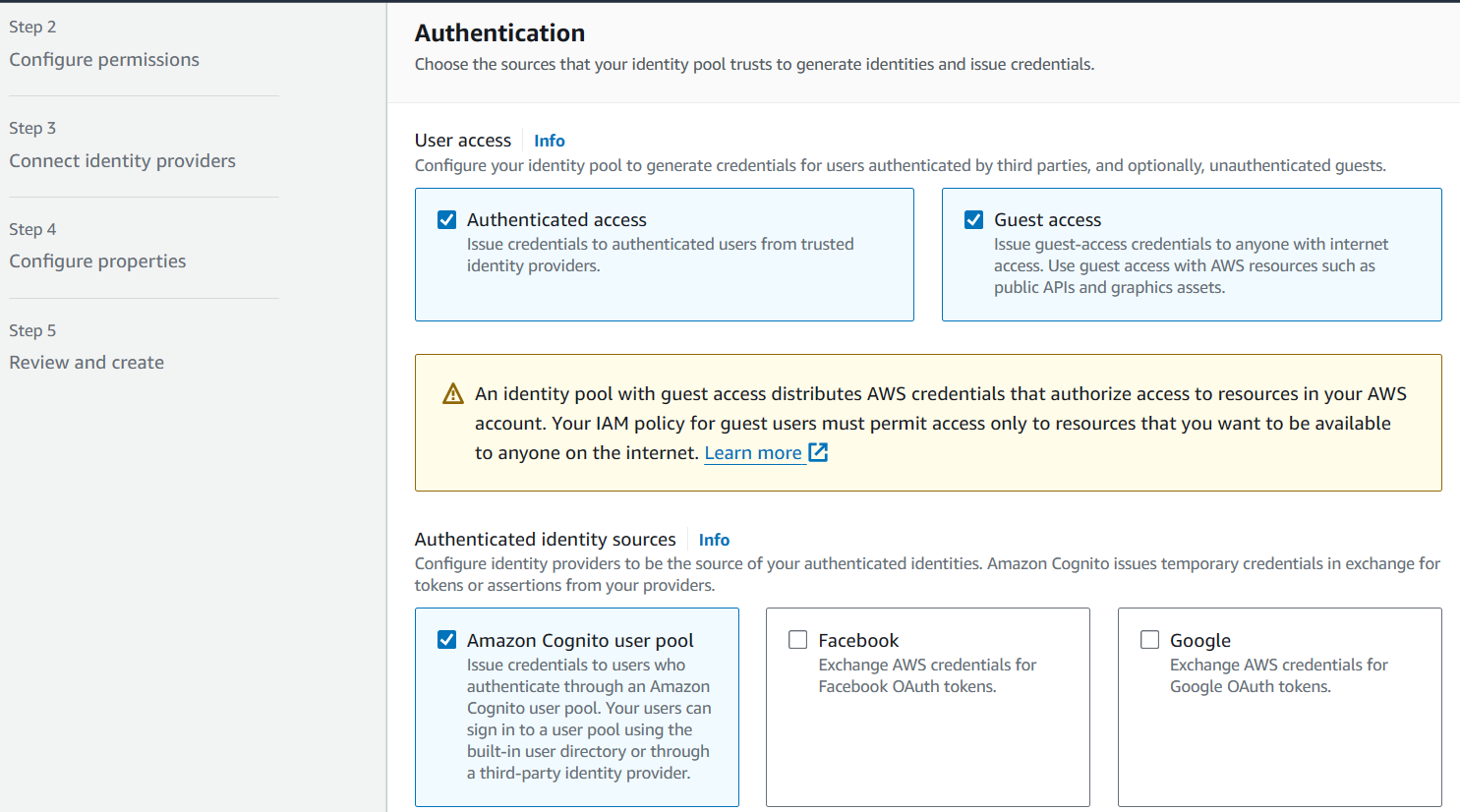
* **Lambda Triggers:**
  + Utilize Lambda functions to react to user pool events (e.g., sign-up, confirmation, custom authentication) and build custom integrations.
  + Explore various trigger types like pre-signup, post-confirmation, and migrate user.



## Creating an Amazon Cognito Identity Pool

**1. Choose Access Type:**

* **Authenticated access:** Users sign in using an identity provider before accessing resources.
* **Guest access (optional):** Allows anyone to access resources without signing in.

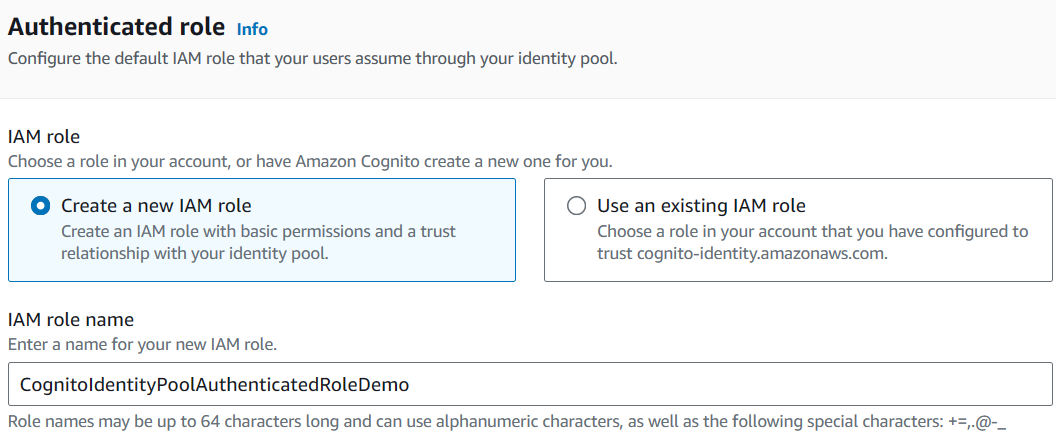


**2. Authentication Source:**

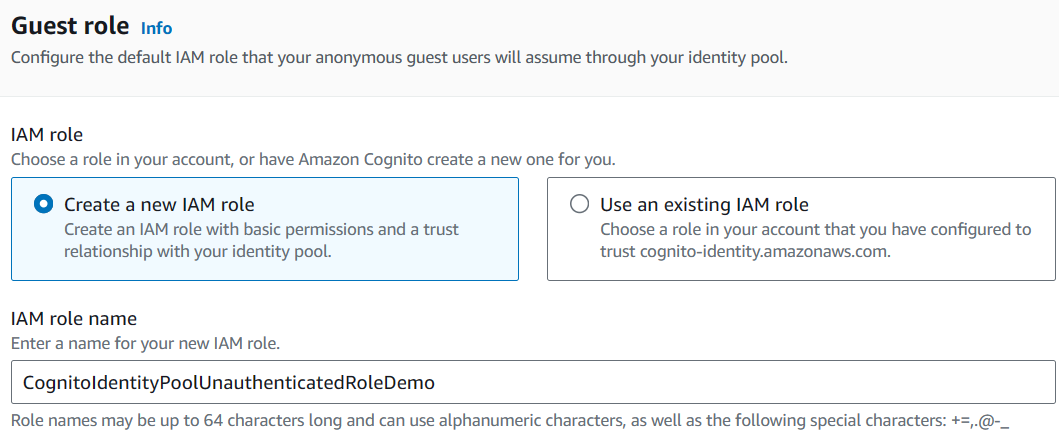
* Select **Amazon Cognito User Pool** as the source (as you've already created one).

**3. Configure Permissions:**

* Create IAM roles for:
  + **Authenticated users:** CognitoIdentityPoolAuthenticatedRoleDemo (define its permissions later).

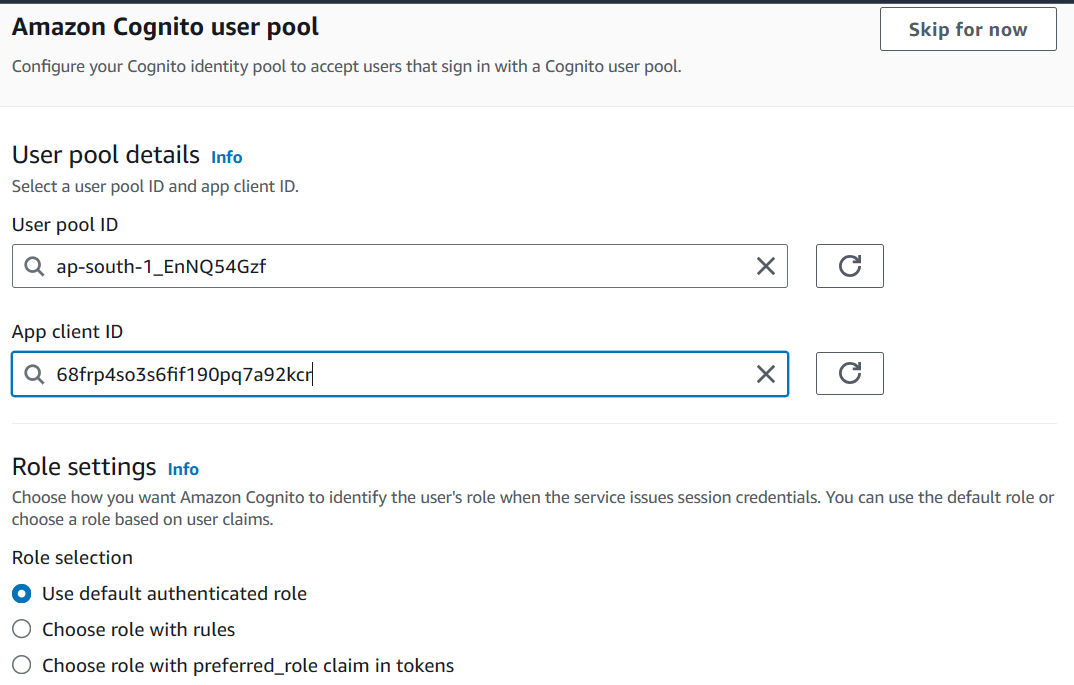


* + **Guest users (optional):** CognitoIdentityPoolUnauthenticatedRoleDemo (define its permissions later).



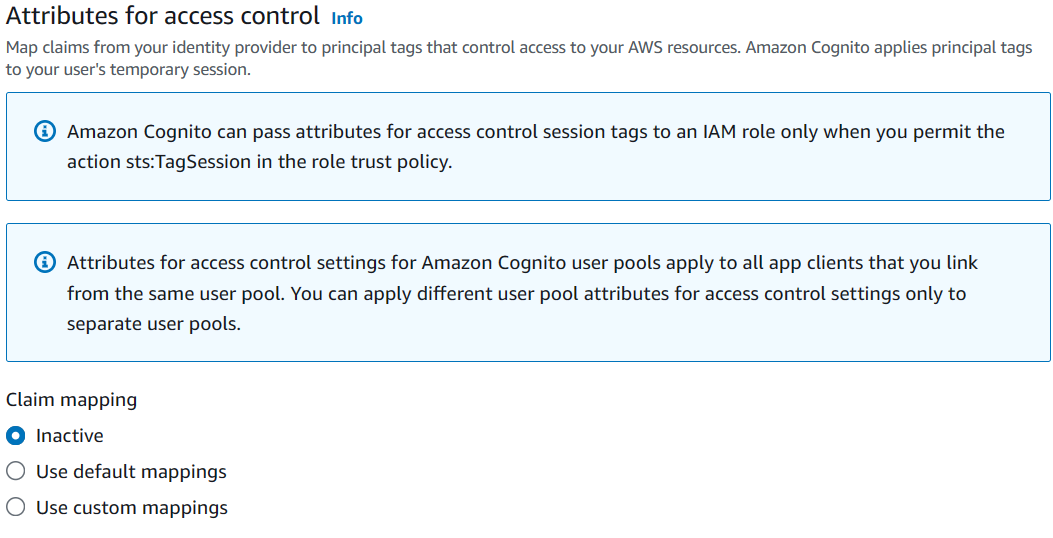
**4. User Pool Configuration:**

* Specify the **User Pool ID** and **App Client ID** from your previously created user pool.
* Choose **use default authenticated role** for simplicity.



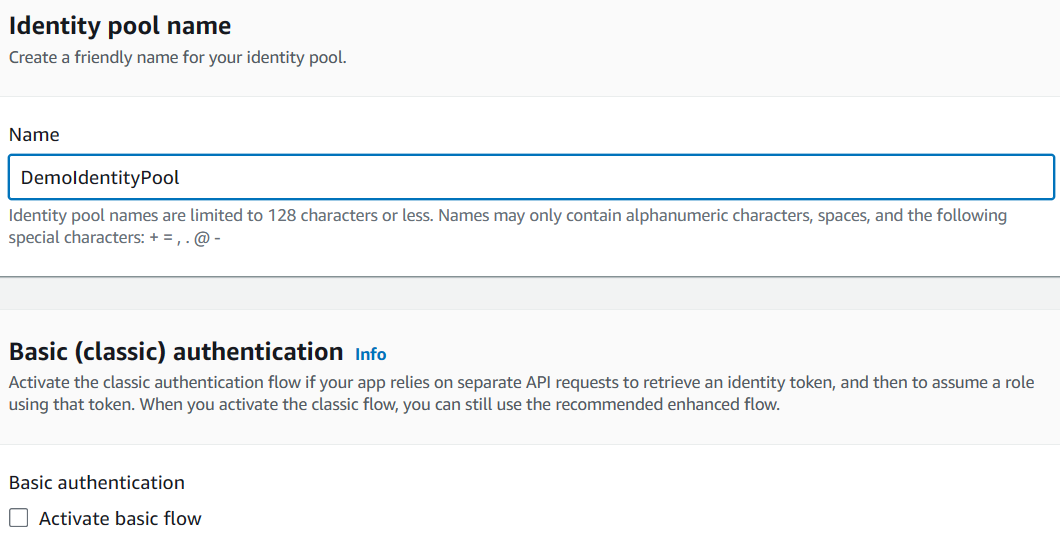
**5. Optional: Attribute Mapping (Advanced):**

* Map user attributes to IAM policies for granular access control.



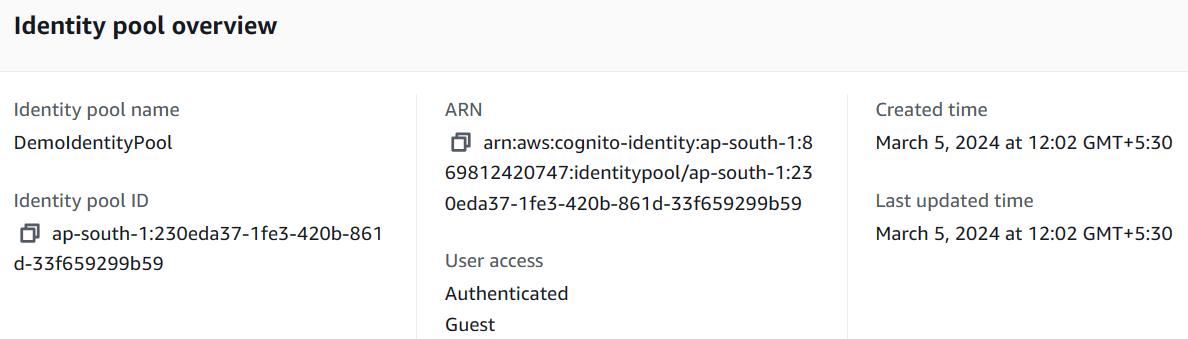
**6. Review and Create:**

* Name your pool (e.g., DemoIdentityPool).
* Choose **Classic authentication** (optional).
* Review configuration and create the identity pool.



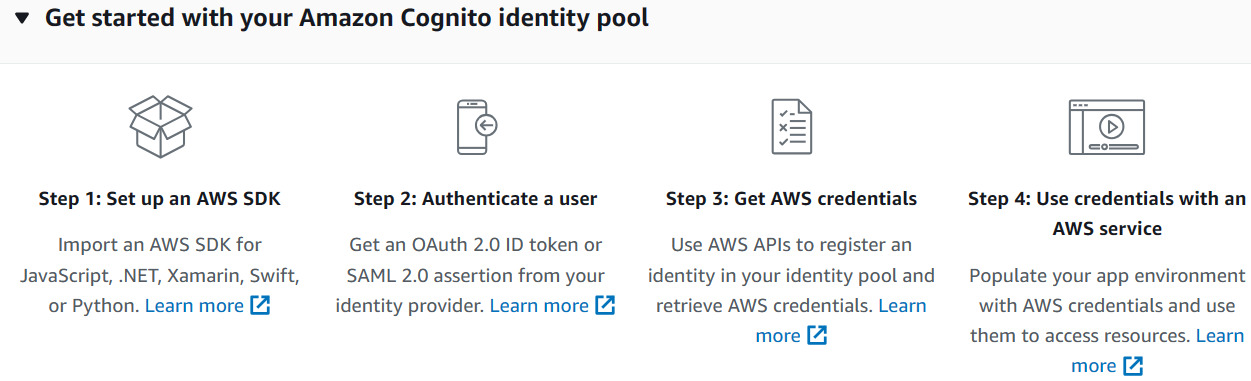
## Integrating Cognito Identity Pool with your Application

1. **Access the Identity Pool:** Navigate to the newly created identity pool in the Cognito console.
2. **Review Roles:** Observe the presence of both **authenticated** and **guest** roles within the pool.



1. **Integrate with your code:**

* **Set up an SDK:** Integrate the appropriate AWS SDK for your programming language (e.g., AWS SDK for JavaScript, Python, etc.) into your application code.
* **Authenticate a user:** Implement user authentication logic in your code using the chosen SDK.
* **Obtain AWS credentials:** Use the SDK to retrieve temporary AWS credentials (access key and secret key) for the authenticated user.



1. **Manage IAM roles:**

* **Modify IAM roles:** In the AWS IAM console, adjust permissions for the CognitoIdentityPoolAuthenticatedRoleDemo and CognitoIdentityPoolUnauthenticatedRoleDemo roles to grant desired access to AWS services for authenticated and guest users, respectively.
* **Set service permissions:** Use IAM policies to define specific permissions for each role, such as read access to S3 buckets or specific actions within other AWS services.

