😊 Amazon Cognito

Amazon Cognito is a service offered by AWS (Amazon Web Services) that helps you manage user authentication, authorization, and access control for your web and mobile applications. It provides secure user sign-up, sign-in, and access to your apps by enabling you to easily authenticate users through social identity providers like Facebook, Google, and Apple, as well as through your own custom identity provider.

Key Features of Amazon Cognito:

Amazon Cognito is a comprehensive identity management solution within the AWS ecosystem, designed to simplify the process of adding authentication and authorization to web and mobile applications. Here's a more detailed overview:

1. User Pools

- User Sign-Up and Sign-In: Amazon Cognito User Pools provide user directories that allow users to sign up and sign in to your apps. The sign-up process can include user verification through email or phone number. You can customize the sign-up process with required attributes, such as name or address, and configure it to support multi-factor authentication (MFA).
- OAuth 2.0 and SAML 2.0 Support: User Pools support OAuth 2.0 authorization flows, enabling easy integration with various identity providers. You can also integrate with SAML-based identity providers for enterprise use cases.
- Customizable UI: You can customize the UI for the hosted sign-up and sign-in pages, or you can create your own UI while using the Cognito APIs in the backend.

2. Identity Pools

- Authentication and Authorization: With Identity Pools, you can create unique identities for users and authenticate them through Cognito User Pools or other identity providers. Identity Pools enable users to get temporary AWS credentials to directly access AWS services like Amazon S3, DynamoDB, and others.
- Unauthenticated Users: Identity Pools allow you to define permissions for unauthenticated users (guest users) who have not signed in. This is useful for allowing limited access to resources before users sign up or sign in.
- Role-Based Access Control (RBAC): You can define IAM roles with specific permissions and map users to these roles based on their identity pool or attributes, allowing for fine-grained access control.

3. Federated Identities

- Social Identity Providers: Amazon Cognito supports authentication through popular social identity providers like Facebook, Google, Amazon, and Apple. This allows users to sign in using their existing social media credentials.

- Enterprise Identity Providers: You can also integrate with enterprise identity providers using SAML 2.0 or OpenID Connect (OIDC), enabling single sign-on (SSO) for enterprise applications.
- Custom Identity Providers: If your application uses a custom authentication mechanism, you can integrate it with Cognito by defining a custom identity provider.

4. Security and Compliance

- Multi-Factor Authentication (MFA): Cognito supports both SMS-based and TOTP (Time-based One-Time Password) MFA, enhancing the security of user accounts.
- Encryption: All data managed by Amazon Cognito is encrypted at rest and in transit, ensuring the security and privacy of user data.
- Compliance: Amazon Cognito is compliant with several industry standards, including GDPR, HIPAA, and SOC 2, making it suitable for applications with strict regulatory requirements.

5. Integration with Other AWS Services

- AWS Lambda: You can use Lambda triggers to customize authentication workflows. For example, you can use a pre-sign-up trigger to validate user attributes or a post-confirmation trigger to send a welcome email.
- Amazon API Gateway: Cognito can be integrated with API Gateway to provide authentication and authorization for your APIs, leveraging Cognito's user pools for token-based authentication.
- AWS Amplify: If you're building mobile or web applications using AWS Amplify, Amazon Cognito is the default solution for handling authentication, providing a seamless integration experience.

6. Analytics and Monitoring

- Amazon CloudWatch: You can monitor Amazon Cognito metrics using Amazon CloudWatch to track the performance and usage of your user pools and identity pools.
- AWS CloudTrail: Track all API calls made to Cognito using AWS CloudTrail, providing you with an audit trail of all activities within your Cognito environment.

7. Customization and Extensibility

- Custom Attributes: You can define custom attributes in your user pools to store additional information specific to your application.
- Custom Authentication Flows: Cognito allows you to implement custom authentication flows using AWS Lambda triggers. This gives you control over how authentication is handled, enabling scenarios like custom passwordless login mechanisms.

8. Pricing

- User Pools: Pricing is based on the number of Monthly Active Users (MAUs). You are charged for each MAU that authenticates during the month.

- Identity Pools: Charges are based on the number of operations, such as token exchanges or user synchronizations. Additional charges apply if your users access other AWS services using temporary credentials provided by Cognito.
- Free Tier: Amazon Cognito offers a free tier, which includes a limited number of MAUs and operations, making it a cost-effective option for small applications.

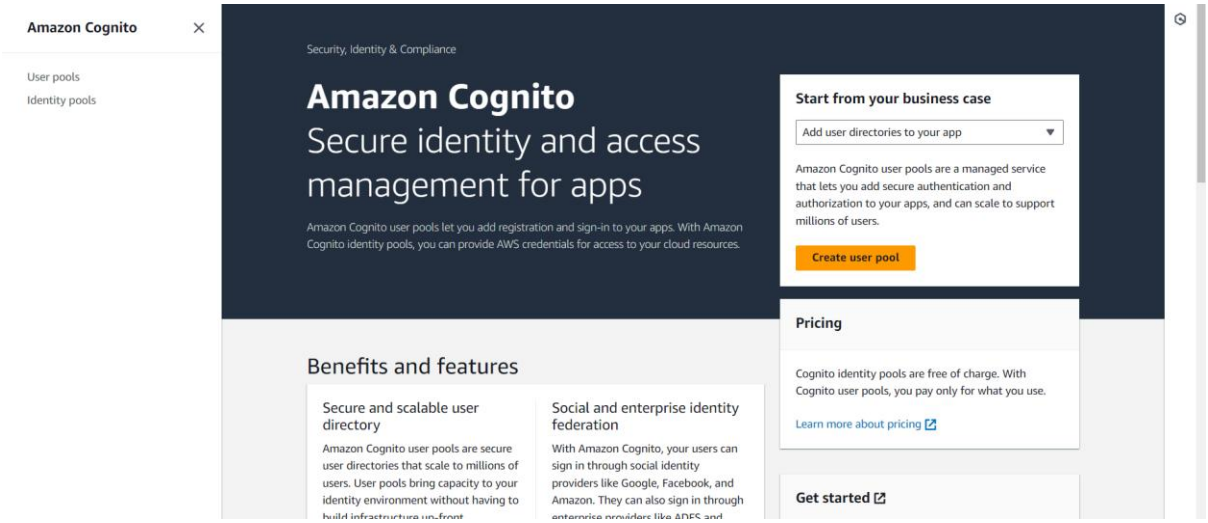
9. Use Cases

- Mobile and Web Apps: Cognito is widely used in mobile and web apps that require secure user management, particularly in e-commerce, social networking, and content delivery platforms.
- Enterprise Applications: Its ability to integrate with SAML and OIDC makes it a strong choice for enterprise applications that need to support single sign-on (SSO) with existing corporate identities.
- IoT Applications: Cognito's support for unauthenticated access and fine-grained access control makes it suitable for IoT applications where devices or sensors need access to AWS services.

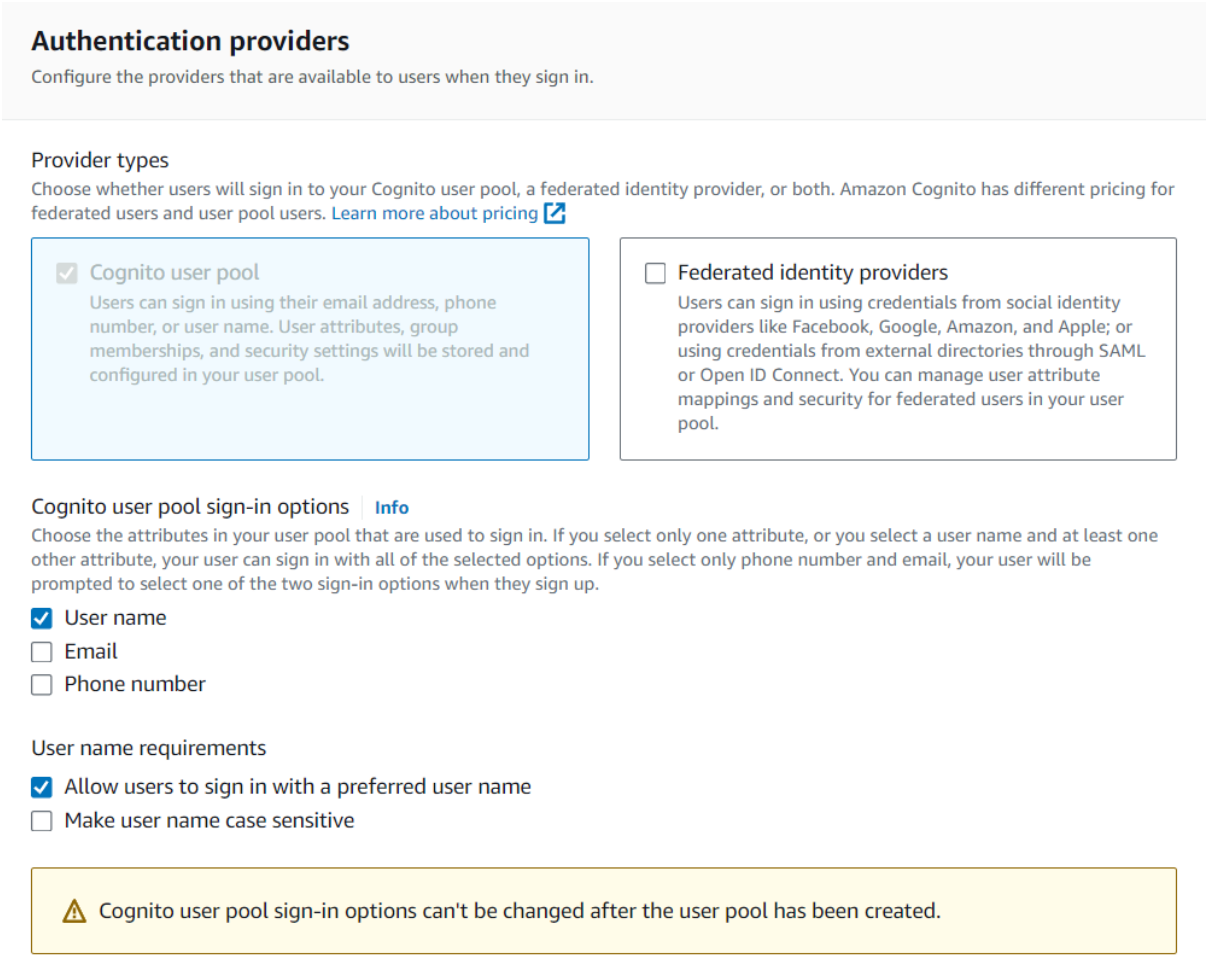
Amazon Cognito provides a powerful and flexible way to manage user identities and permissions, making it a popular choice for developers building scalable and secure applications on AWS.

😄 To begin with the Lab:

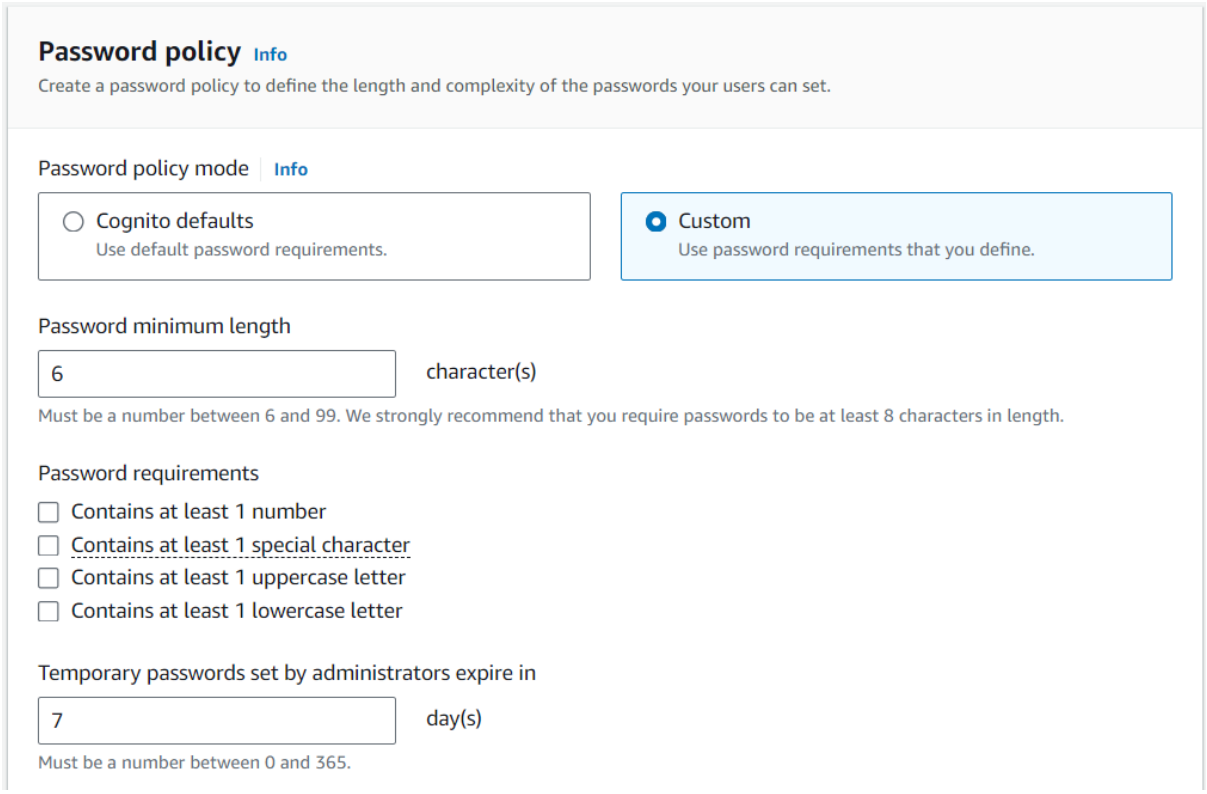
1. In your AWS Console, search for Cognito and go toward its dashboard, then from here you have to click on Create user pools.



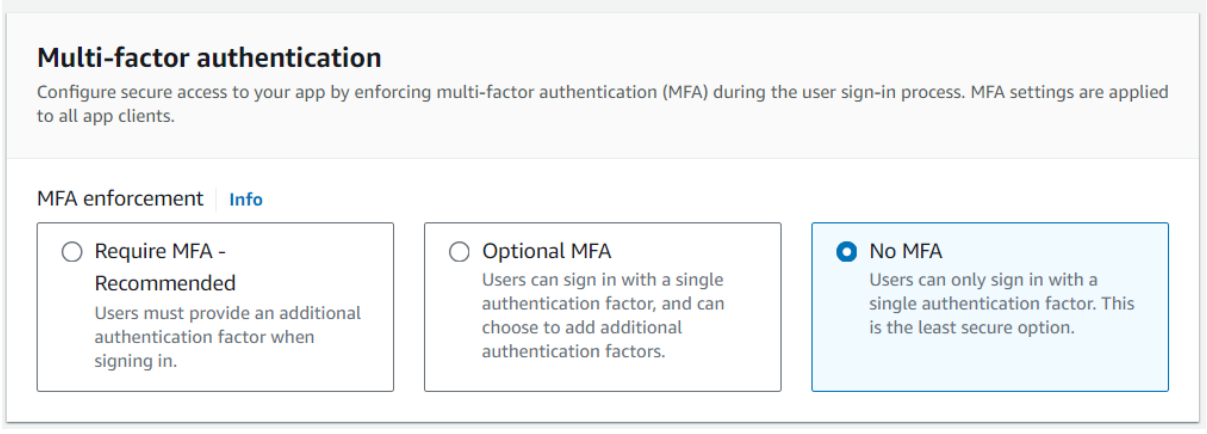
2. Now on step 1 you have to choose the same specifications as you can see below in the snapshot and click on next.

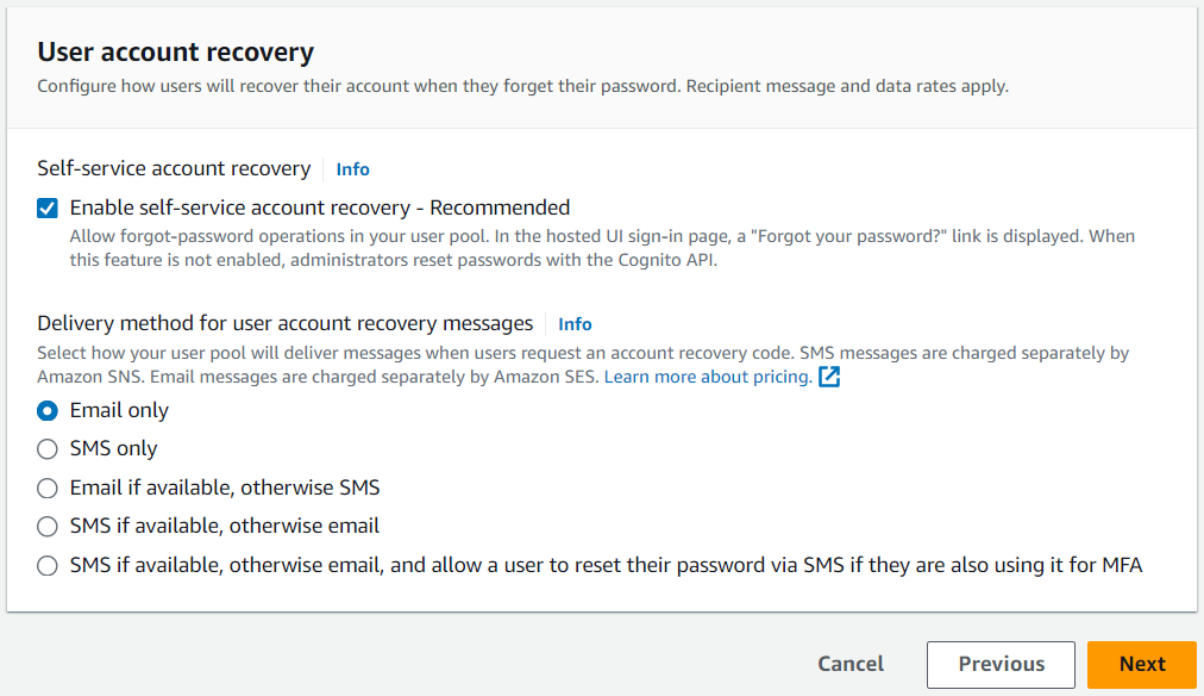


3. Then in step 2 you have to choose custom for password policy mode and set the password minimum length to 6 characters. After that disable all password requirements and scroll down.

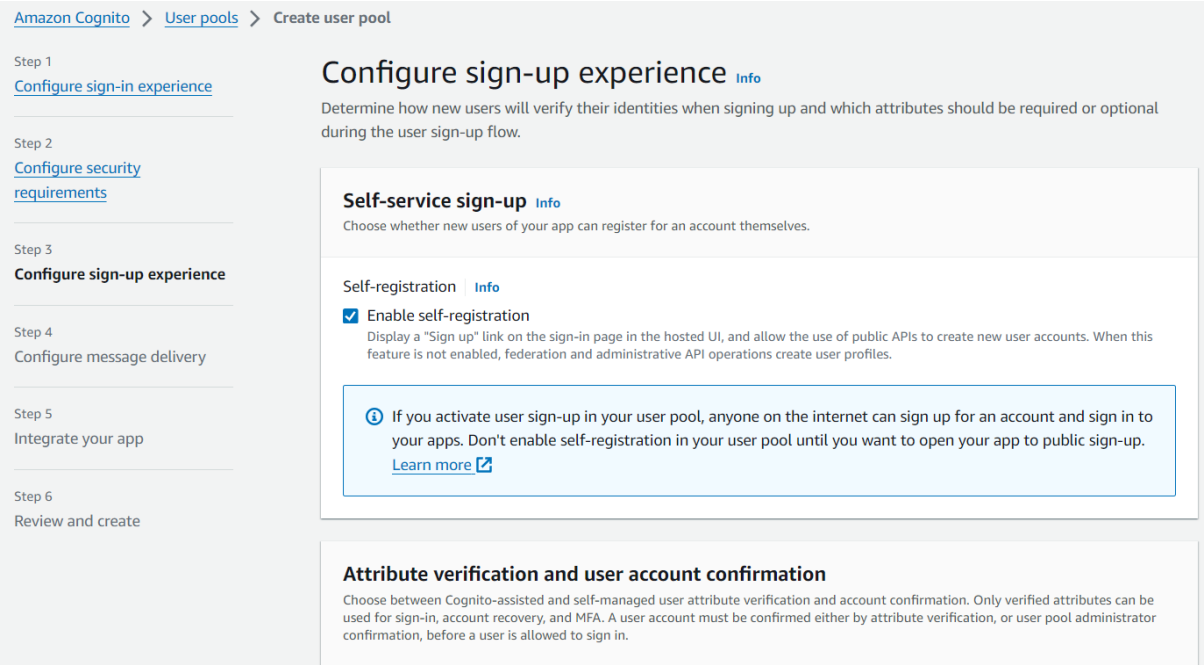


4. Now say no to multi-factor authentication but enable user account recovery as shown below and click on next.

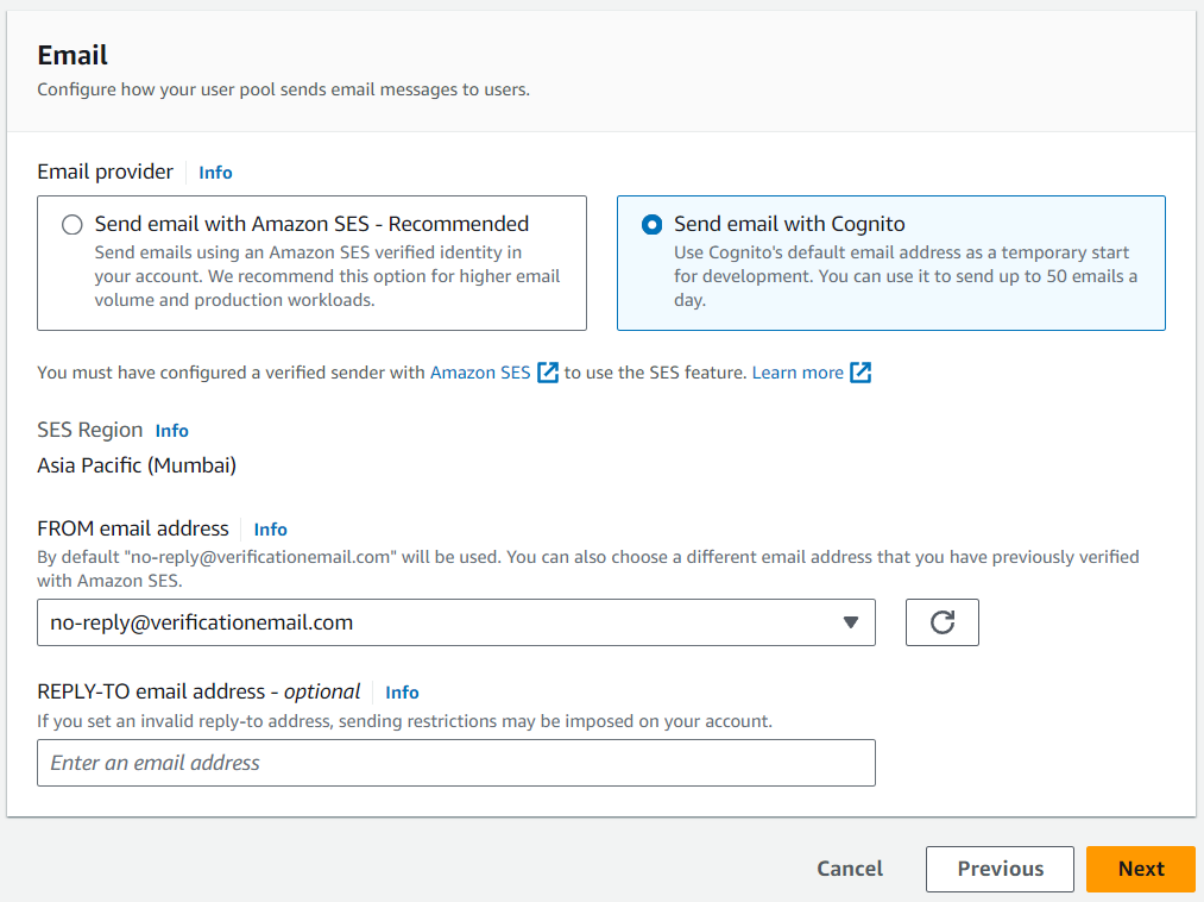




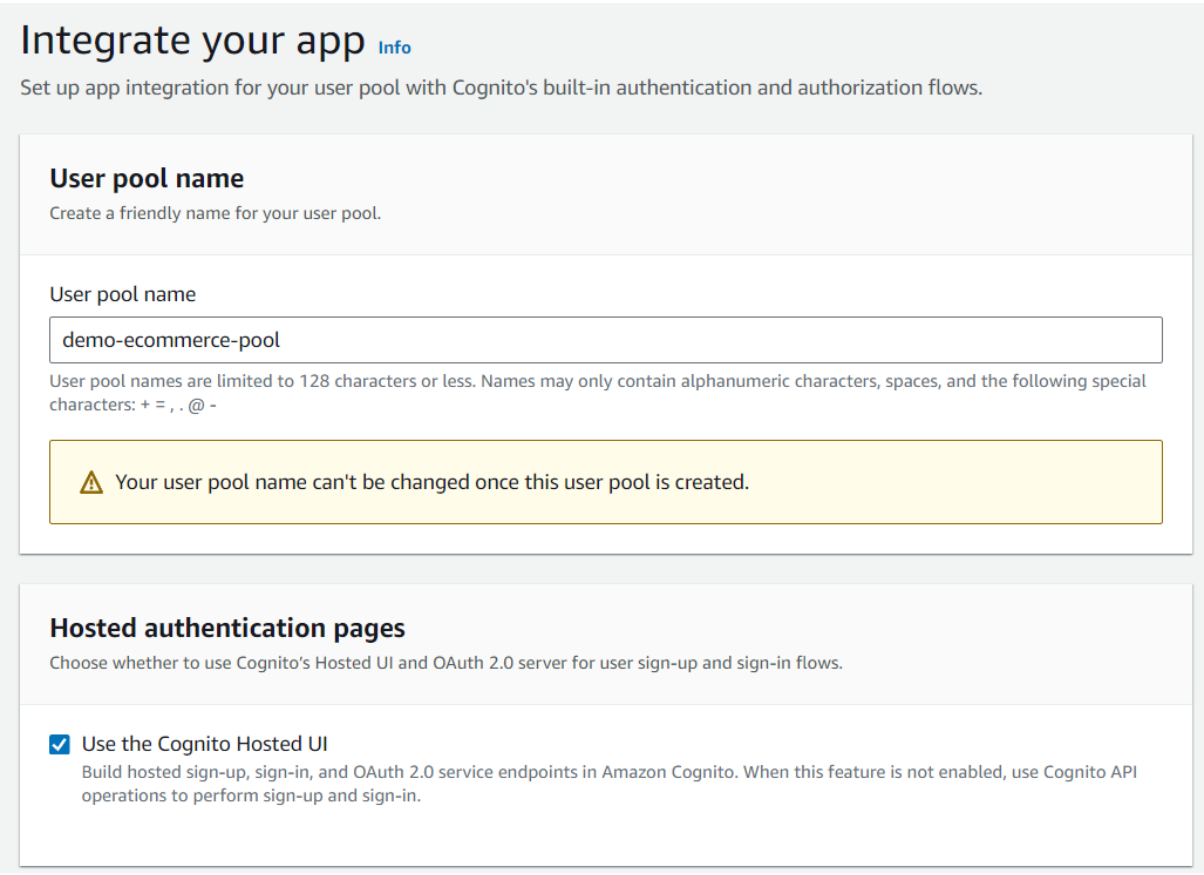
5. So, for step 3 you have to keep this step to default settings and move to the next step.



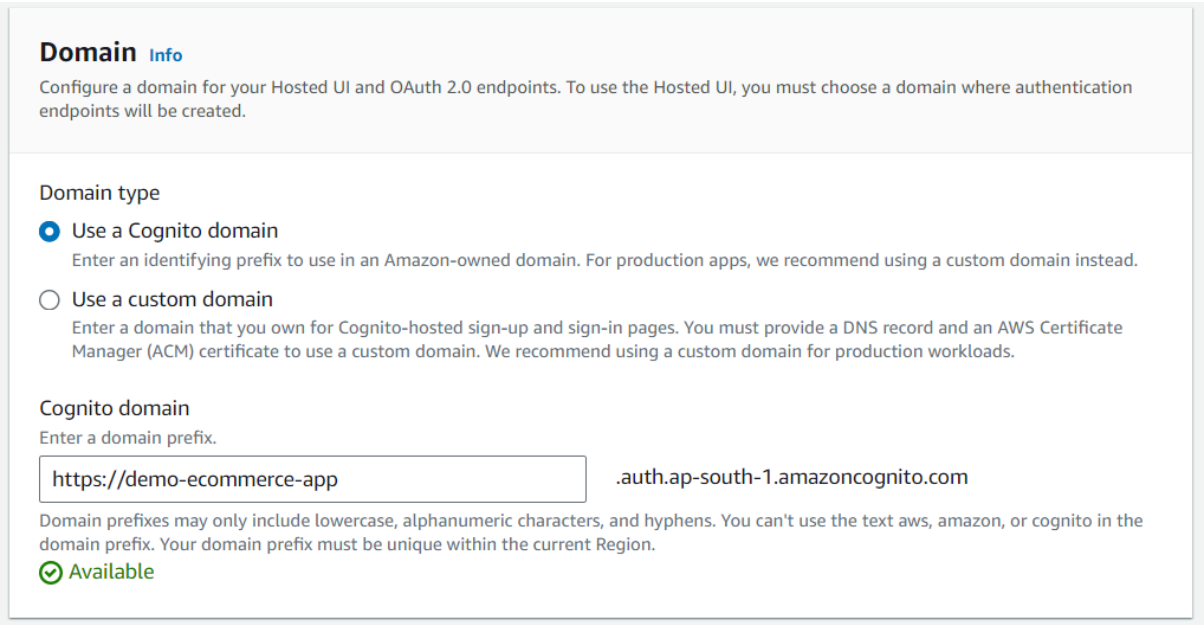
6. Now for step 4 choose Send email with Cognito and click on next.



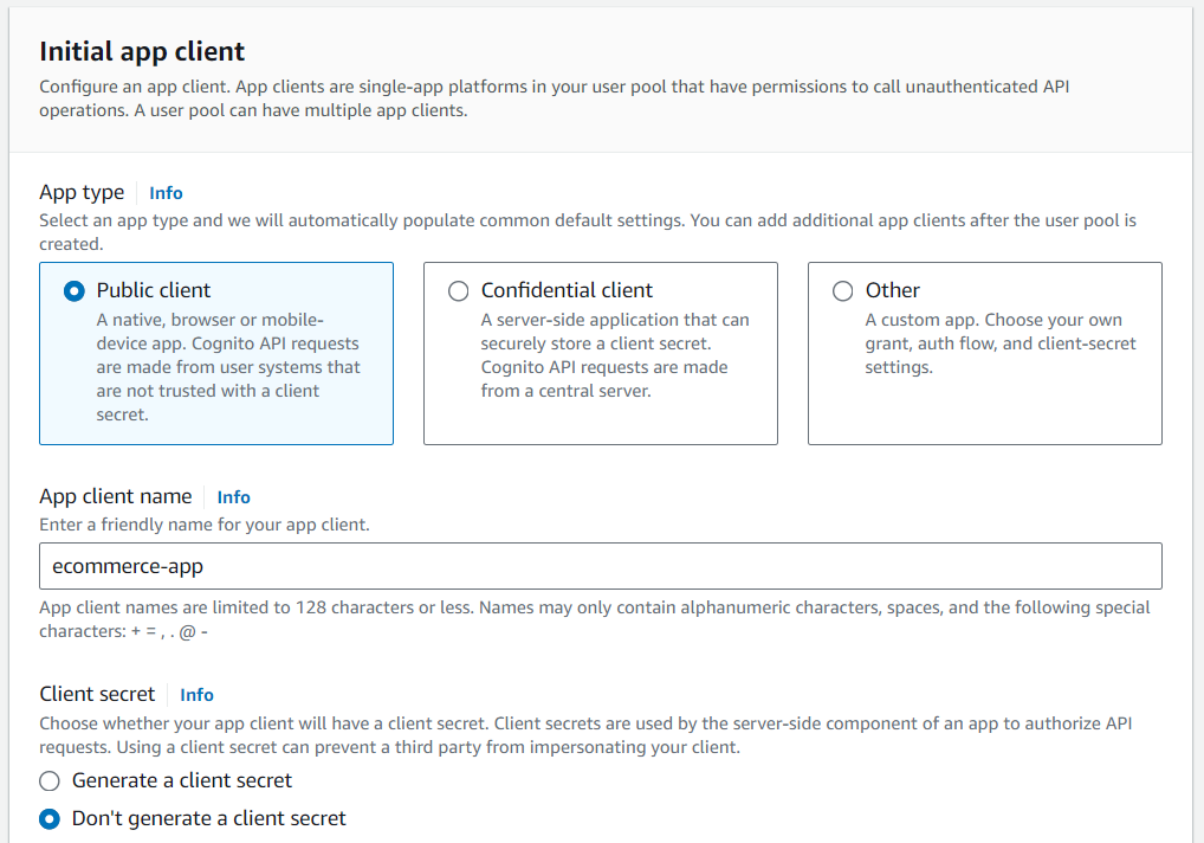
7. For step 5, first give it a user pool name and enable use the Cognito hosted UI.



8. After that in domain choose use a Cognito domain and give a unique domain name.



9. Then in the initial app client you should choose a public client and give an App client name. For client secret choose Don't generate.



10. Now for the allowed callback URL give any localhost and expand Advanced app client settings. Here you just need to check the application flows. These options should be selected which you can see below.
11. Then leave everything as it is and create your user pool.

Allowed callback URLs | Info

Enter at least one callback URL to redirect the user back to after authentication. This is typically the URL for the app receiving the authorization code issued by Cognito. You may use HTTPS URLs, as well as custom URL schemes.

URL

https://localhost:8000 Remove

Length of callback URL must be between 1 and 1024 characters. Valid characters are letters, marks, numbers, symbols, and punctuations. Amazon Cognito requires HTTPS over HTTP except for http://localhost for testing purposes only. App callback URLs such as myapp://example are also supported. Must not contain a fragment.

Add another URL

You can add 99 more URLs

Advanced app client settings

We have populated suggested authentication flows, OAuth 2.0 grant types, and OIDC scopes based on the selections you made earlier.

Authentication flows | Info

Choose authentication flows that your app will support. Refresh token authentication is always enabled. We have populated options based on your app type.

Select authentication flows

ALLOW_REFRESH_TOKEN_AUTH
Refresh token based authentication

ALLOW_USER_SRP_AUTH
SRP (secure remote password) protocol based authentication

ALLOW_CUSTOM_AUTH
Lambda trigger based custom authentication

12. Below you can see that your user pool has been created successfully. You can go inside it and check the configurations.