Cognito

Web Identity Federation

lets you give your users access to AWS resources after they have successfully authenticated with a web-based identity provider like Amazon, FB, Google.

Following successful authentication, the user receives an auth code from the web ID provider which they can trade for temporary AWS security credentials.

- sign up and sign in for your apps
- access for guest users
- acts as an identity brokeer between your application and web id providers - you don't need to write any additional code
    - temp credentials that Cognito generates from the FB login maps to an IAM role allowing access to required resources.
    - no need for the app to embed or store AWS credentials locally on the device and it gives users a seamless experience across all mobiel devices
- sync user data for multiple devices
- recommended for Web ID Federation and espseically all mobile applications for AWS services

User Pools

User Pools are user directories used to managed sign-up ann sign-in functionality for mobile and web applications.

Identity Pools enable you to create unique identities for your users and authenticate them with IDPs. With an identity, you can obtain temporary limited-priviliege AWS credentials to access other AWS services.

> After successfully authenticating a user, Amazon Cognito issues JSON web tokens (JWT) that you can use to secure and authorize access to your own APIs, or exchange for AWS credentials.

Cognito tracks association between user identity and the various different devices they sign in from.

- in order to provide a seamless user experience. Cognito uses **Push Synchronization** to push updates and sync user data across multiple devices.
- Amazon SNS is used to send a silent push notification to all the devices associated with a given user identity whenever data stored in the cloud changes.

Inline Policies vs Managed Policies vs Custom Policies

Managed Policy - AWS-managed default policies (***AWS recommended***)

- for common use cases based on job function e,g, DynamoDBFullAccess, EC2ReadOnlyAccess
- these policies allow you to assign appropriate permission to your users, groups, and roles without having to write the policy yourself.
- a single Managed Policy can be attached to multiple users, groups, or roles within the same AWS account and across different accounts.
- Can't change the permissions defined in an AWS managed Policy

Customer Managed Policies - standalone IAM policy that you create and admin inside your own AWS account. Can attach to multiple users, groups, and roles **only in your own account**

- Can create Customer Managed Policy by copying existing AWS Managed Policy and customizing it.

Inline Policies - embedded within the user, group, or role to which it applies. There is a strict 1:1 relationship between the entity and the policy.

When you delete the user, group, or role in which the policy is embedded, the policy is also deleted.

- useful when you want to be sure that the permiossions in a policy are not inadvertently assigned to any other user, group, or role than the one intended.