Overview

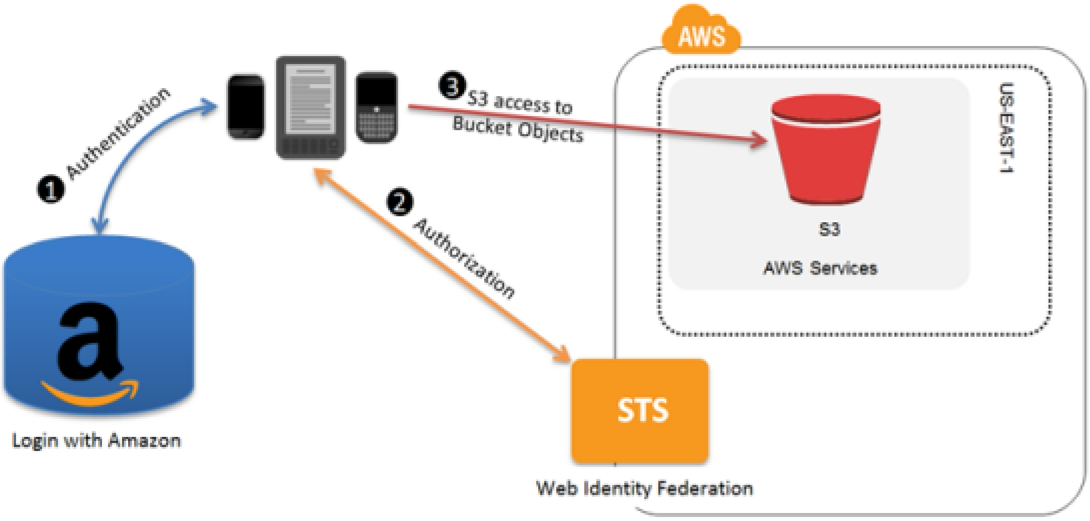
It allows users to authenticate with a Web Identity Provider (Google, Facebook, Amazon)

Application and AWS Environment setups:

a. From Mobile or Web Application site:
    i. This needs to be configured with Identity provider (Google, Amazon, Facebook, or OpenID compatible IdP)

a. From AWS Account Site:
    i. Create an Identity Provider Entity for OIDC compatible IdP in IAM
    ii. Create an IAM role and define the following two policies:
        1. Trust Policy: Specify the IdP (like Amazon) as the Principal (the trusted entity), and include a Condition that matches the IdP assigned app ID
        2. Permission Policy: specify the permissions the application can assume

Main steps of workflow:



Step 1:

a. Application calls the sign-in interface for the IdP to login
b. IdP authenticates the user and returns an **authentication token** (OAuth access token or OIDC ID token) with information about the user to the application

Step 2:

a. Application then makes an unsigned call to the STS service with the **AssumeRoleWithWebIdentity** action to request temporary security credentials.

b. Application passes the IdP’s authentication token along with the Amazon Resource Name (ARN) for the IAM role created for that IdP.

c. AWS verifies that the token is trusted and valid and if so, returns temporary security credentials (***access key, secret access key, session token, expiry time***) to the application that has the permissions for the role that you name in the request.

d. STS response also includes metadata about the user from the IdP, such as the unique user ID that the IdP associates with the user.

Step 3:

a. Using the Temporary credentials, the application makes signed requests to AWS

b. User ID information from the identity provider can distinguish users in the app for e.g., objects can be put into S3 folders that include the user ID as prefixes or suffixes. This lets you create access control policies that lock the folder so only the user with that ID can access it.

Step 4:

a. Application can cache the temporary security credentials and refresh them before their expiry accordingly. Temporary credentials, by default, are good for an hour.