Cognito:Cognito User Pools,Cognito Identi...

Amazon Cognito

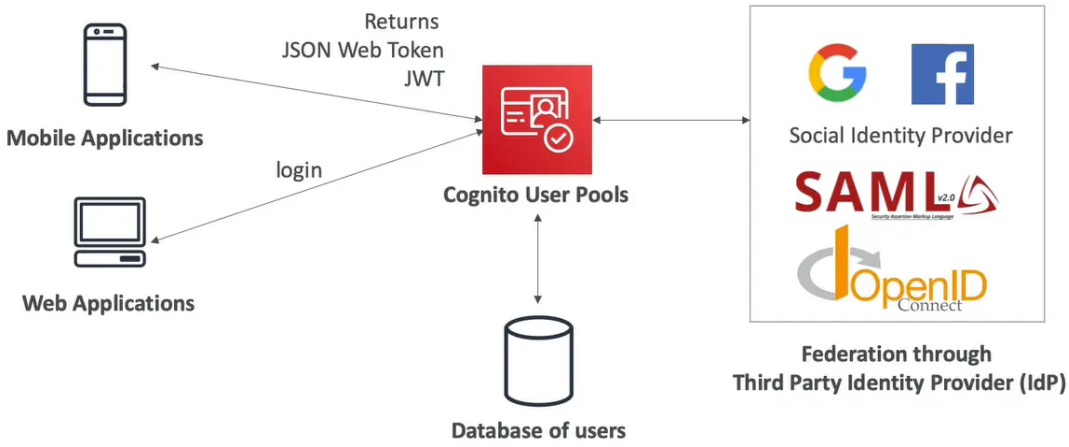
- give users an identity to interact with [交互] our web or mobile application
- Cognito User Pools
  - sign in functionality for app users
  - integrate with API Gateway & Application Load Balancer
- Cognito Identity Pools (Federated Identity)
  - provide AWS credentials to users so they can access AWS resources directly
  - integrate with cognito user pools as an identity provider

- Cognito vs IAM : ”hundreds of users”,”mobile users”,”authenticate with SAML”

Cognito User Pools (CUP) – User Features

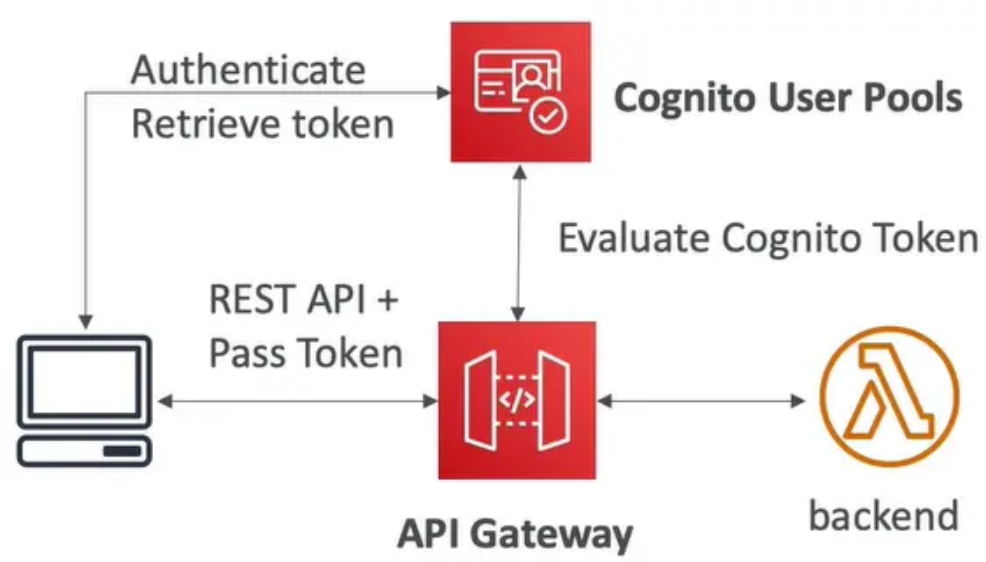
- create a serverless database of user for your web & mobile apps
- simple login: username (or email) / password combination
- password reset
- email & phone number verification
- multi-factor authentication (MFA)
- federated [联合] identities : users from facebook, google ,SAML
- feature : block users if their credentials are compromises elsewhere
- login sends back a JSON Web Token (JWT)

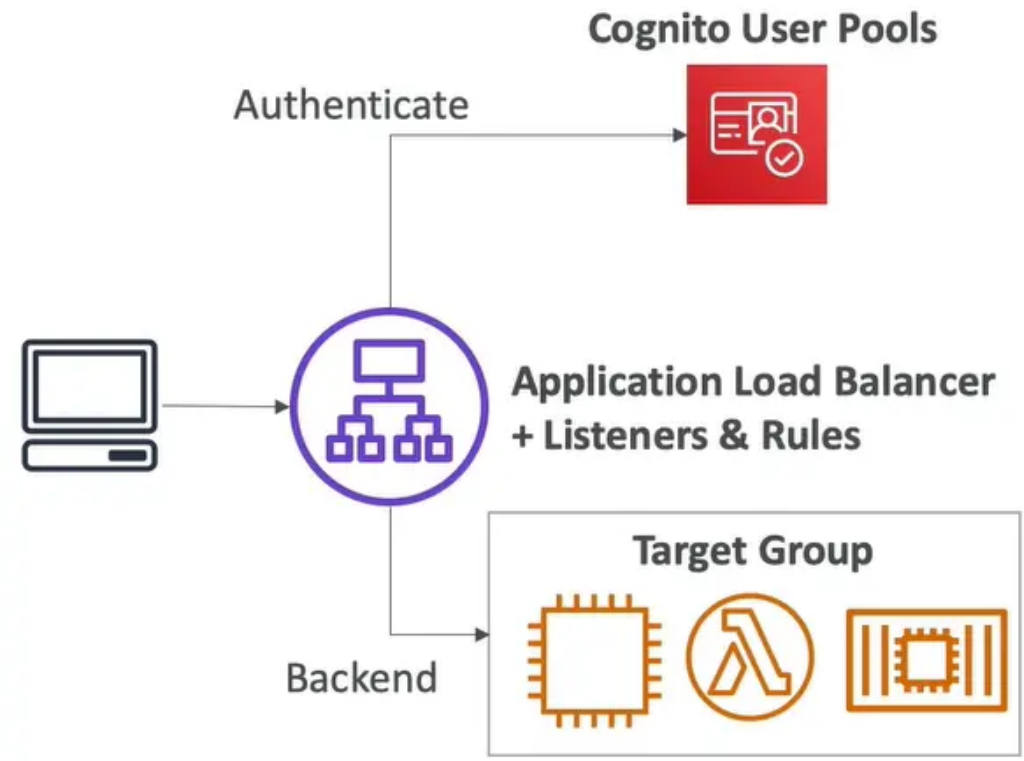
Diagram



integrations

- CUP integrates with API Gateway and Application Load Balancer





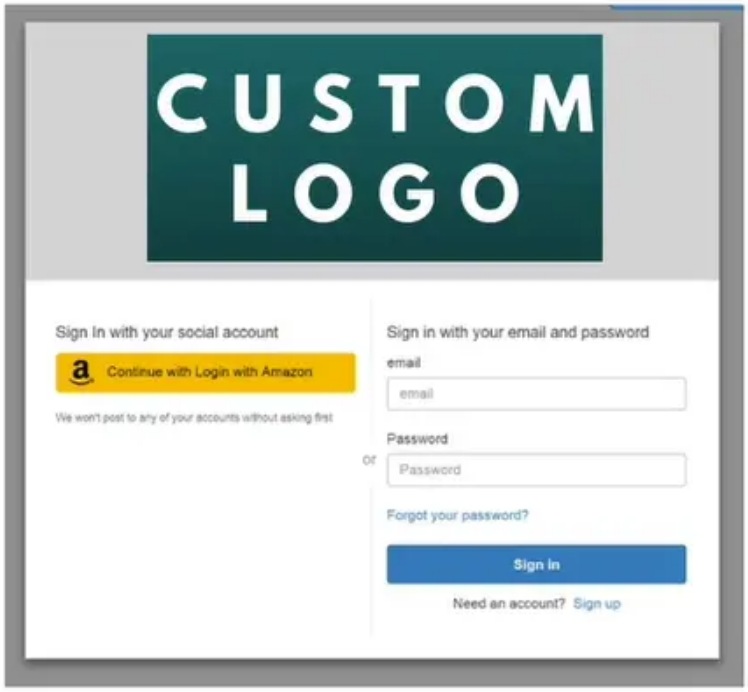
Lambda Triggers

- CUP cna invoke a Lambda function synchronously on these triggers

User Pool Flow	Operation	Description
Authentication Events	Pre Authentication Lambda Trigger	Custom validation to accept or deny the sign-in request
	Post Authentication Lambda Trigger	Event logging for custom analytics
	Pre Token Generation Lambda Trigger	Augment or suppress token claims
Sign-Up	Pre Sign-up Lambda Trigger	Custom validation to accept or deny the sign-up request
	Post Confirmation Lambda Trigger	Custom welcome messages or event logging for custom analytics
	Migrate User Lambda Trigger	Migrate a user from an existing user directory to user pools
Messages	Custom Message Lambda Trigger	Advanced customization and localization of messages
Token Creation	Pre Token Generation Lambda Trigger	Add or remove attributes in Id tokens

https://docs.aws.amazon.com/cognito/latest/developerguide/cognito-user-identity-pools-working-with-aws-lambda-triggers.html

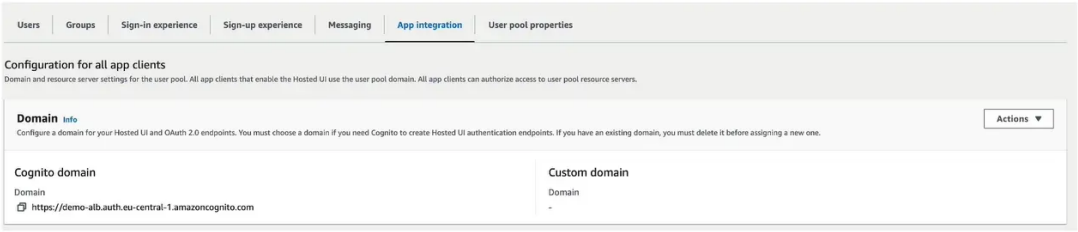
Hosted Authentication UI



- cognito has a **hosted authentication UI** that you can add to your app to handle sign-up and sign-in workflows
- using the hosted UI , you have a foundation for integration with social logins, OIDC or SAML
- can customize with a **custom logo** and **custom CSS**

Hosted UI Custom Domain

- for custom domains ,you must create an ACM certificate in **us-east-1**
- the custom domain must be defined in the "APP integration" section



CUP Adaptive Authetication

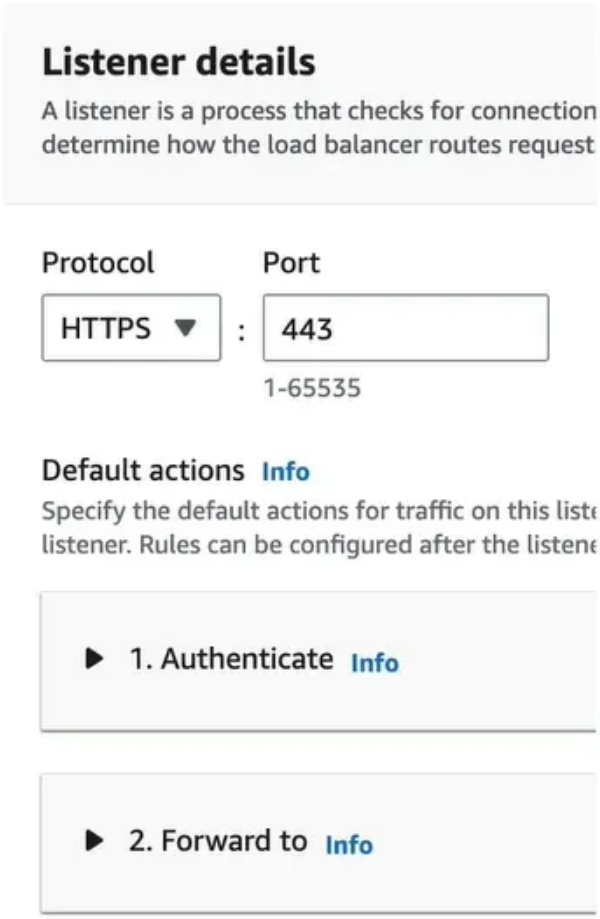
- Block sign-ins or require MFA if the login appears suspicious [可疑的]
- cognito examines each sign-in attempt and generate a risk score (low, medium,high) for how likely the sign-in request is to be from a malicious [恶意的] attacker
- users are prompted for a second MFA only when risk is detected
- risk score is based on different factors such as if the user has used the same device, location, or IP address
- checks for compromised credentials ,account takeover protection, and phone and email verification
- integration with CloudWatch Logs (sign-in attempts, risk score,failed challenges...)

Decoding a ID Token; JWT- JSON Web Token

ID JWT Token Payload

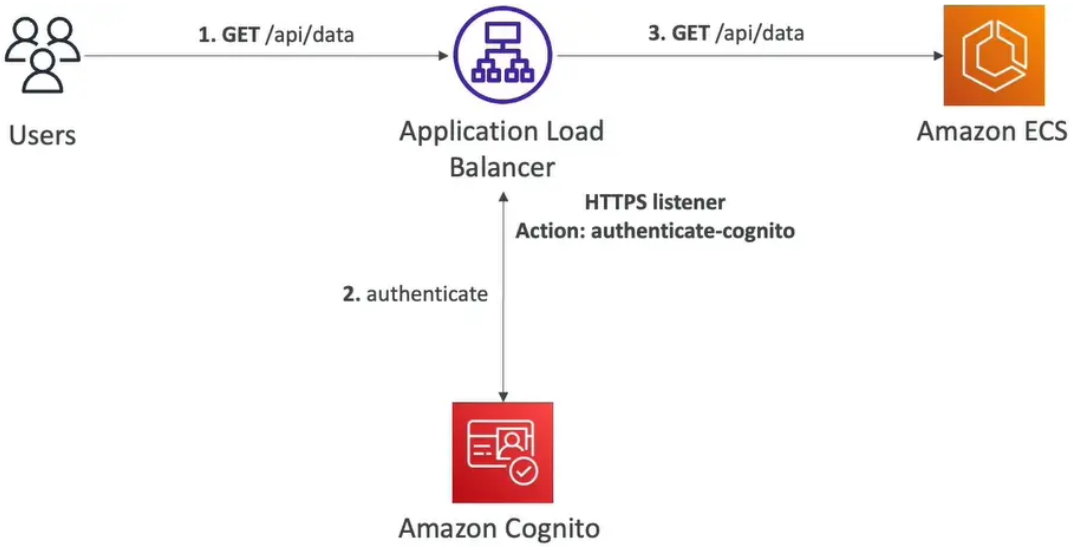
- CUP issues JWT tokens (base64 encoded)
  - Header
  - Payload
  - Signature
- the signature must be verified to ensure the JWT can be trusted
- Libraries can help you verify the validity [合法性] of JWT tokens issued by Cognito User Pools
- the payload [有效负载] will contain the user information (sub UUID,give_name,email,phone_number,atttributes...)
- from the sub UUID, you can retrieve all users details from Cognito / OIDC

Application Load Balancer – Authenticate Users

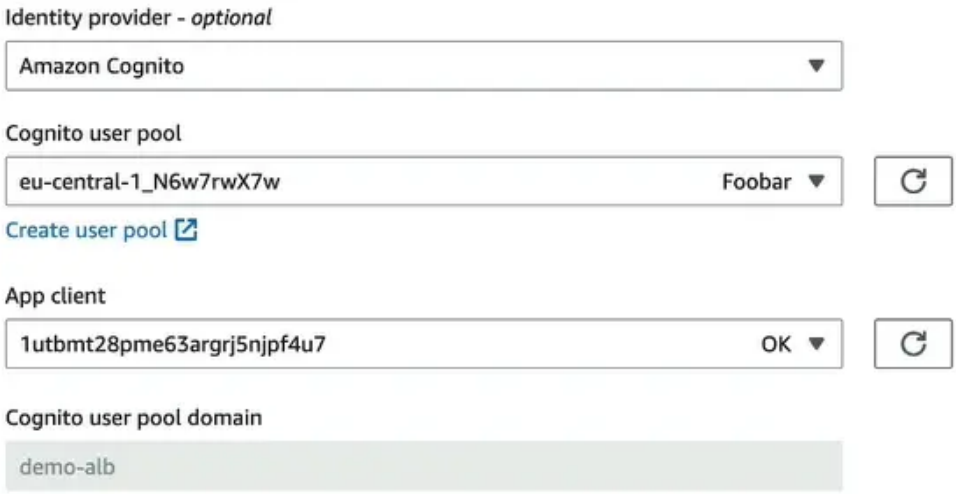


- your application load balancer can securely authenticate users
  - offload [卸载] the work of authenticating users to your load balancer
  - your applications can focus on their bussiness logic
- authenticate users through
  - Identitiy Provider(IdP):OpenID Connect (OIDC) compliant [合规的]
  - Cognito User Pools
    - social IdPs, such as Amazon, Facebook,or Google
    - Corporate identities using SAML, LDAP, or Microsoft AD
- Must use an HTTPS listener to set authenticate-oidc & authenticate-cognito rules
- OnUnauthticatedRequest – authenticate (default),deny,allow

Application Load Balancer – Cognito Auth

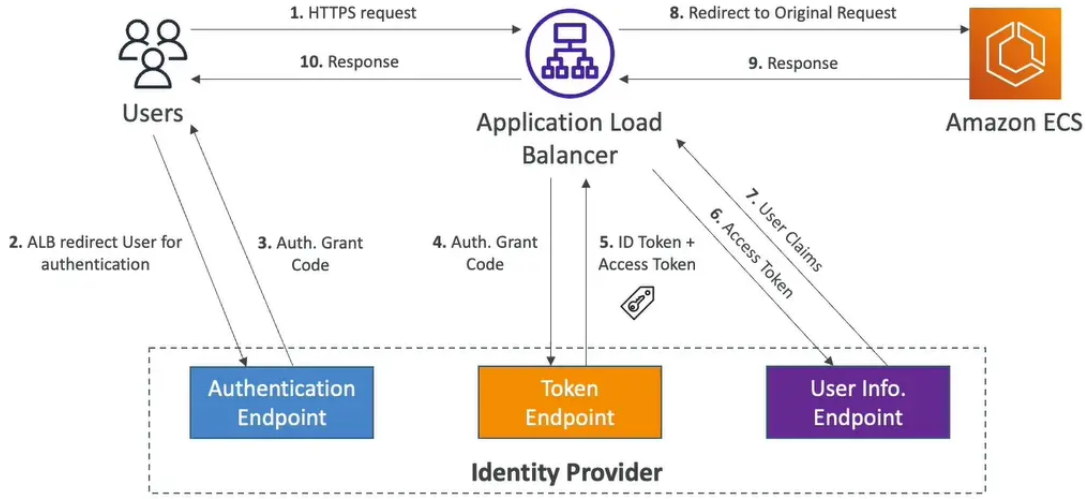


ALB – Auth through Cognito User Pools

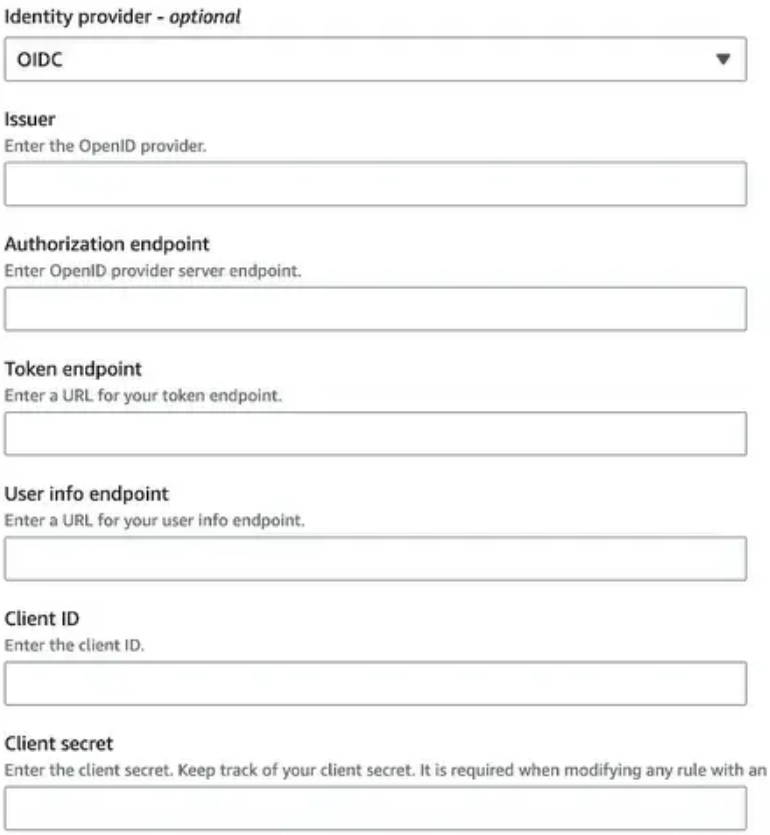


- create Cognito user pool, Client and Domain
- make sure an ID token is returned
- add the social or corporate IdP if needed
- several URL redirections are necessary
- allow your Cognito User Pool Domain on your IdP app's callback URL, For example:
  - http://domain-prefix.auth.region.amazoncognito.com/sam/idpresponse

Application Load Balancer – OIDC Auth



ALB – Auth. Through an Identity Provider (IdP) that is OpenID Connect (OIDC) compliant [合规]



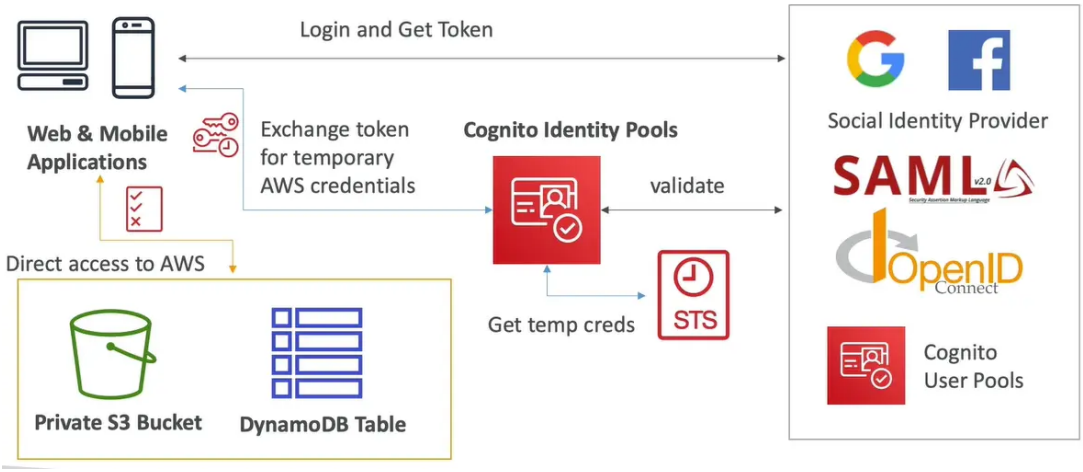
- configure a Client ID & Client Secret
- allow redirect from OIDC to you application Load balancer DNS name (AWS provided) and CNAME (DNS Alias of your app)
  - https://DNS/oauth2/idpresponse
  - https://CNAME/oauth2/idpreponse

Cognito Identity Pools (Federated [联邦] Identities)

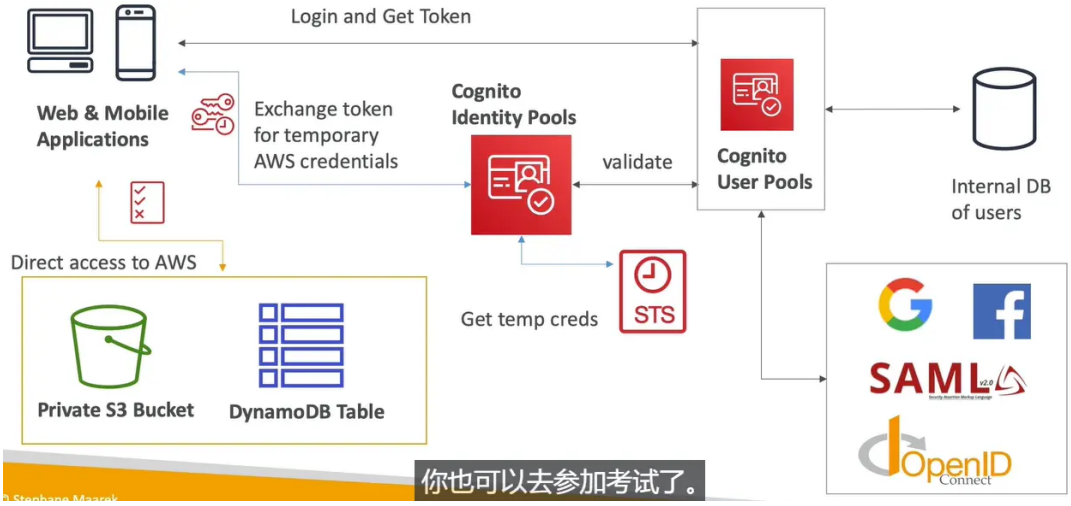
- get identities for "users" so they obtain temporary AWS credentials
- your identity pool (eg,identity source) can include
  - public providers (login with Amazon, Facebook,Google,Apple)
  - Users in an Amazon Cognito user pool
  - openID connect provider & SAML identity providers
  - developer Authenticated Identities (custom login server)
  - cognito identity pools allow for unauthenticated (guest) access

- users can then access AWS services directly or through API Gateway
  - the IAM policies applied to the credentials are defined in cognito
  - they can be customized based on the user_id for fine grained control

Diagram



Cognito Identity Pools – Diagram with CUP



Cognito Identity Pools – IAM Roles

- default IAM Roles for authenticated and guest users
- define rules to choose the role for each user based on the user's ID
- you can partition [划分] your users' access using policy variables

- IAM credentials are obtained by Cognito Identity Pools through STS
- the roles must have a "trust" policy of Cognito Identity Pools

Guest User Example

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "s3:GetObject"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:s3:::mybucket/assets/my_picture.jpg"
            ]
        }
    ]
}
```

Policy variable on S3

DynamoDB

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "dynamodb:GetItem", "dynamodb:BatchGetItem", "dynamodb:Query",
                "dynamodb:PutItem", "dynamodb:UpdateItem", "dynamodb:DeleteItem",
                "dynamodb:BatchWriteItem"
            ],
            "Resource": [
                "arn:aws:dynamodb:us-west-2:123456789012:table/MyTable"
            ],
            "Condition": {
                "ForAllValues:StringEquals": {
                    "dynamodb:LeadingKeys": [
                        "${cognito-identity.amazonaws.com:sub}"
                    ]
                }
            }
        }
    ]
}
```

Cognito User Pools vs Identity Pools

- cognito user Pools (for authentication [验证] = identity verification)
  - databas of users for your web and mobile application
  - allows to federate [联合] logins through public social, OIDC, SAML..
  - can customized the hosted UI for authentication (including the logo)
  - has triggers with AWS Lambda during the authentication flow
  - adapt the sign-in experience to different risk levels (MFA,adaptive authentication, etc...)
- Cognito Identitiy Pools (for authorization [授权] = access control)
  - obtain AWS credentials for your users
  - users can login through public social,OIDC, SAML & Cognito User pools
  - users can be unauthenticated (guest)
  - users are mapped to IAM roles & policies, can leverage policy variables
- CUP + CIP = authentication + authorization

Cognito Identity Pools – Diagram with CUP