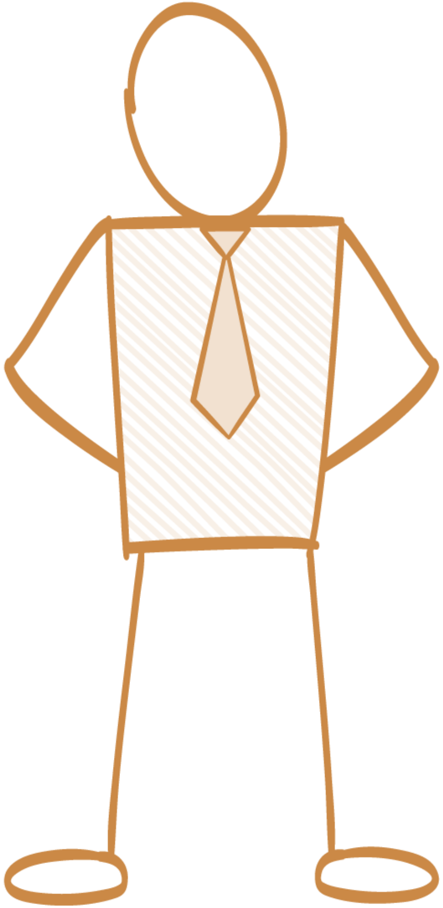
Training and Certification

Architecting on AWS Student Guide

Version 3.1

100-ARC-31-EN-SG



Module 5: Identity, Authentication, and Authorization

Topics

- Authentication, authorization, and where they apply
- AWS Identity and Access Management (IAM)
- Amazon Cognito

Authentication, authorization, and where they apply

- The three major realms where authentication and authorization
- Occur within AWS
- Multi-factor authentication and how to implement it
- Your AWS master account
- Creating users and groups with IAM
- The role of authorization policies

The three realms: WordPress example

- We want to run WordPress on AWS
  1. Login to management console and launch EC2 instance

     -> Management lever authentication and authentication
  2. Login to instance, install WordPress and configure DB connection

     -> component level authentication and authentication
  3. Login to Word press and write a blog post

     -> Application level authentication and authentication

Login to Management console and launch EC2 instance

Authentication and authorization to AWS APIs:

Everything is an API at AWS

You have to make authenticated API requests

Examples of API requests:

```
EC2 -> RunInstance
```

Login to instance, install WordPress and configure DB connection

- Authentication and authorization to **OS**:
  - Local Linux user (for example: `roor@, ubuntu@, ec2-user@`)
  - Local Windows user (Administrator)
- Authentication and authorization to **database**:
  - MySQL username and password
  - SQL Server username and password

Login to WordPress and write a blog post

- Authentication and authorization to the **application**:
  - WordPress authentication to a database
  - Some applications authenticate to Active Directory
  - Others authenticate via Oauth 2.0 and so on

WordPress Example

Task	Can AWS help
Login to Management console and launch EC2 Instance	Yes, a lot
Login to instance, install WordPress and configure DB connection	Yes, some
Login to WordPress and write a blog post	Depends on the application

AWS Solutions Architect Associate

IAM

Identity Access Management Introduction

amazon web services

Identity Access Management (IAM)

Manages **access** of AWS **users** and **resources.**

AWS Solutions Architect Associate

IAM

IAM Core Components

IAM – Core Components

IAM allows **management** of access of **users** and **resources**

IAM Identities

IAM **Users** End users who log into the console or interact with AWS resource programmatically

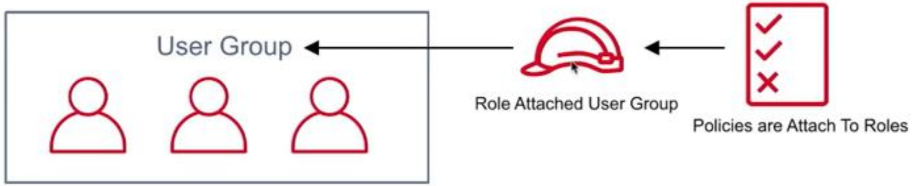
IAM **Groups** Group up your Users so they all share permission levels of the group
eg. Administrators, Developers, Auditors

IAM **Roles** Associate permissions to a Role and then assign this to an Users or Groups

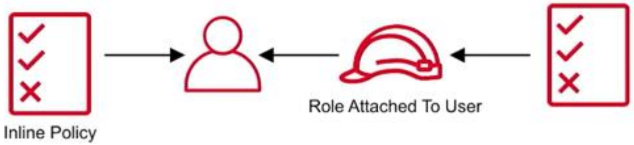
IAM **Policies** JSON documents which grant permissions for a specific user, group, or role to access services. Policies are attached to to **IAM Identities**

IAM – Core Components

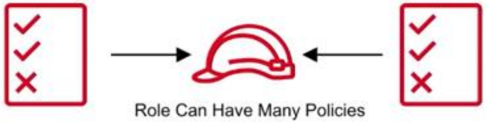
A user can belong to a group.
Roles can be applied to groups to quickly add and remove permissions en-masse to users



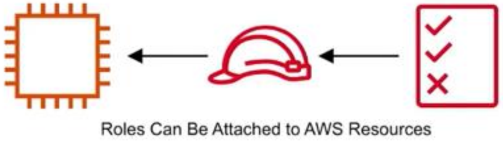
A user can have a role directly attached
An policy can be directly attached to a user (called an **Inline Policy**)



Roles can have many policies attached



Various AWS resources allow you attach roles directly to them.



AWS Solutions Architect Associate

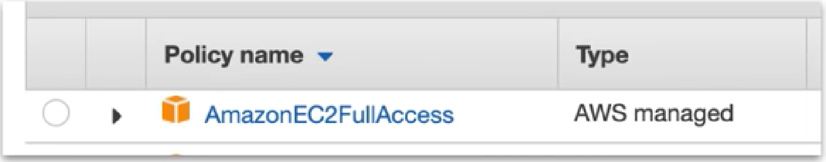
IAM

Types of Policies

IAM – Managed vs Customer vs Inline Policy

Managed Policies

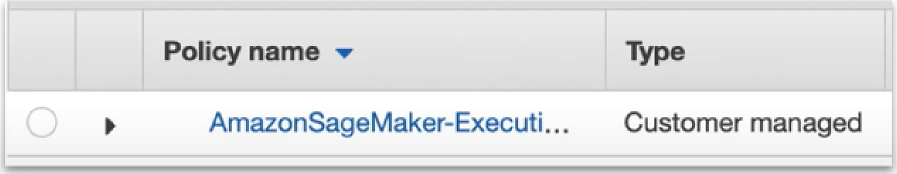
A policy which is managed by AWS, which you cannot edit. Managed policies are labeled with an **orange box**



		Policy name ▾	Type
○	▸	AmazonEC2FullAccess	AWS managed

Customer Managed Policies

A policy created by the customer which is editable. Customer policies have no symbol beside them.



		Policy name ▾	Type
○	▸	AmazonSageMaker-Executi...	Customer managed

Inline Policies

A policy which is directly attached to the user.

AWS Solutions Architect Associate

IAM

Policy Structure

IAM – Policies

Version policy language version. **2012-10-17** is the latest version.

Statement container for the policy element you are allowed to have multiples

Sid (optional) a way of labeling your statements.

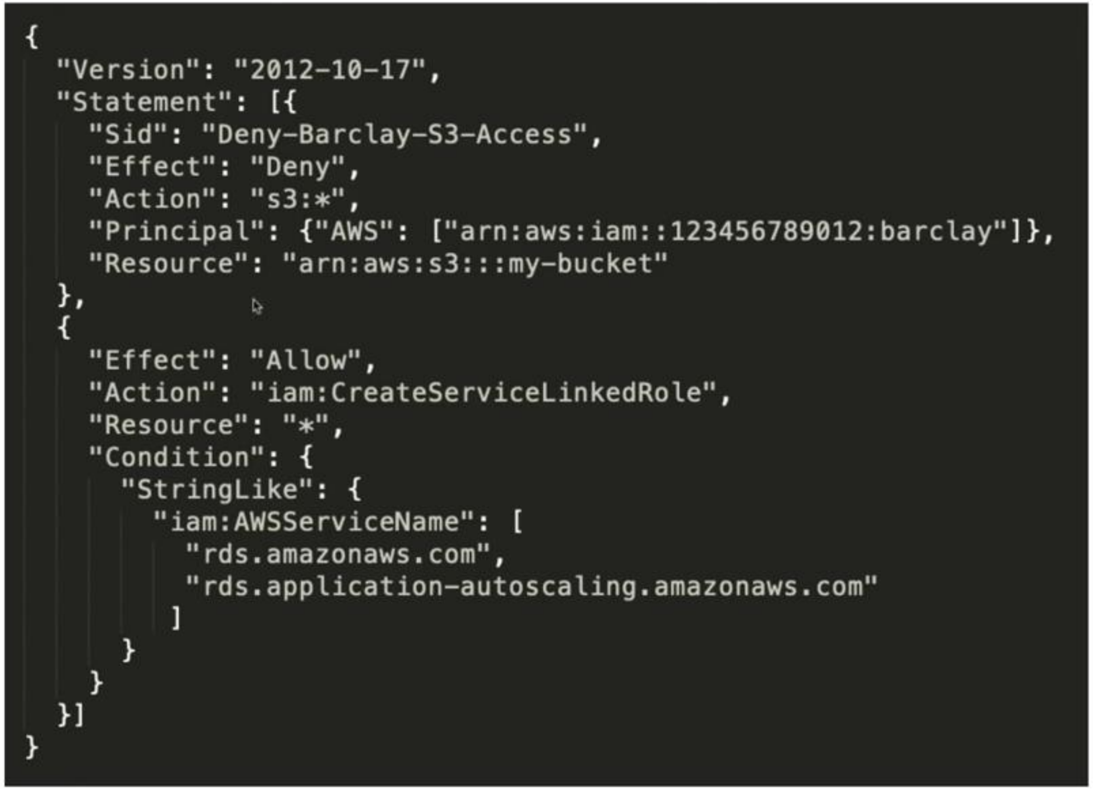
Effect Set whether the policy will Allow or Deny

Principal account, user, role, or federated user to which you would like to allow or deny access

Action list of actions that the policy allows or denies

Resource the resource to which the action(s) applies

Condition (optional) circumstances under which the policy grants permission



```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "Deny-Barclay-S3-Access",
    "Effect": "Deny",
    "Action": "s3:*",
    "Principal": {"AWS": ["arn:aws:iam::123456789012:barclay"]},
    "Resource": "arn:aws:s3:::my-bucket"
  },
  {
    "Effect": "Allow",
    "Action": "iam:CreateServiceLinkedRole",
    "Resource": "*",
    "Condition": {
      "StringLike": {
        "iam:AWSServiceName": [
          "rds.amazonaws.com",
          "rds.application-autoscaling.amazonaws.com"
        ]
      }
    }
  }]
}
```

AWS Solutions Architect Associate

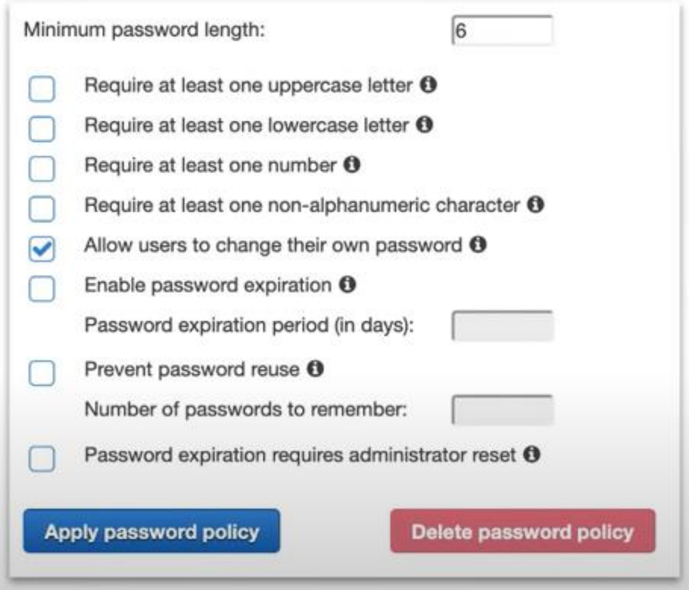
IAM

Password Policy

IAM – Password Policy

In IAM you can set a **Password Policy**

To set the minimum requirements of a password and **rotate** passwords so users have to update their passwords after X days



Minimum password length: 6

- ☐ Require at least one uppercase letter
- ☐ Require at least one lowercase letter
- ☐ Require at least one number
- ☐ Require at least one non-alphanumeric character
- ☑ Allow users to change their own password
- ☐ Enable password expiration

  Password expiration period (in days):
- ☐ Prevent password reuse

  Number of passwords to remember:
- ☐ Password expiration requires administrator reset

AWS Solutions Architect Associate

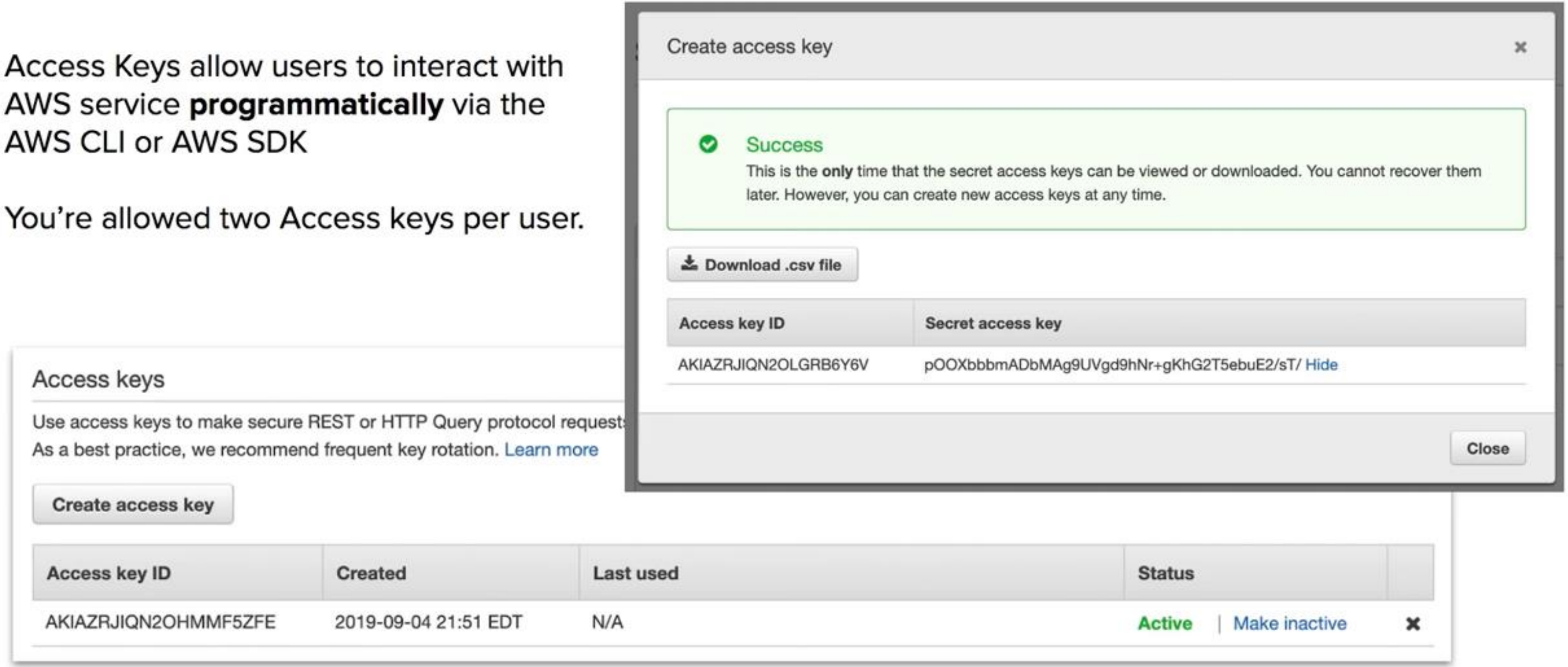
IAM

Programmatic Access Keys

IAM – Access Keys

Access Keys allow users to interact with AWS service **programmatically** via the AWS CLI or AWS SDK

You’re allowed two Access keys per user.



Create access key

Success

This is the **only** time that the secret access keys can be viewed or downloaded. You cannot recover them later. However, you can create new access keys at any time.

Access key ID	Secret access key
AKIAZRJIQN2OLGRB6Y6V	pOOXbbbmADbMAg9UVgd9hNr+gKhG2T5ebuE2/sT/ Hide

Access keys

Use access keys to make secure REST or HTTP Query protocol requests
As a best practice, we recommend frequent key rotation. Learn more

Access key ID	Created	Last used	Status	
AKIAZRJIQN2OHMMF5ZFE	2019-09-04 21:51 EDT	N/A	Active | Make inactive	✖

AWS Solutions Architect Associate

IAM

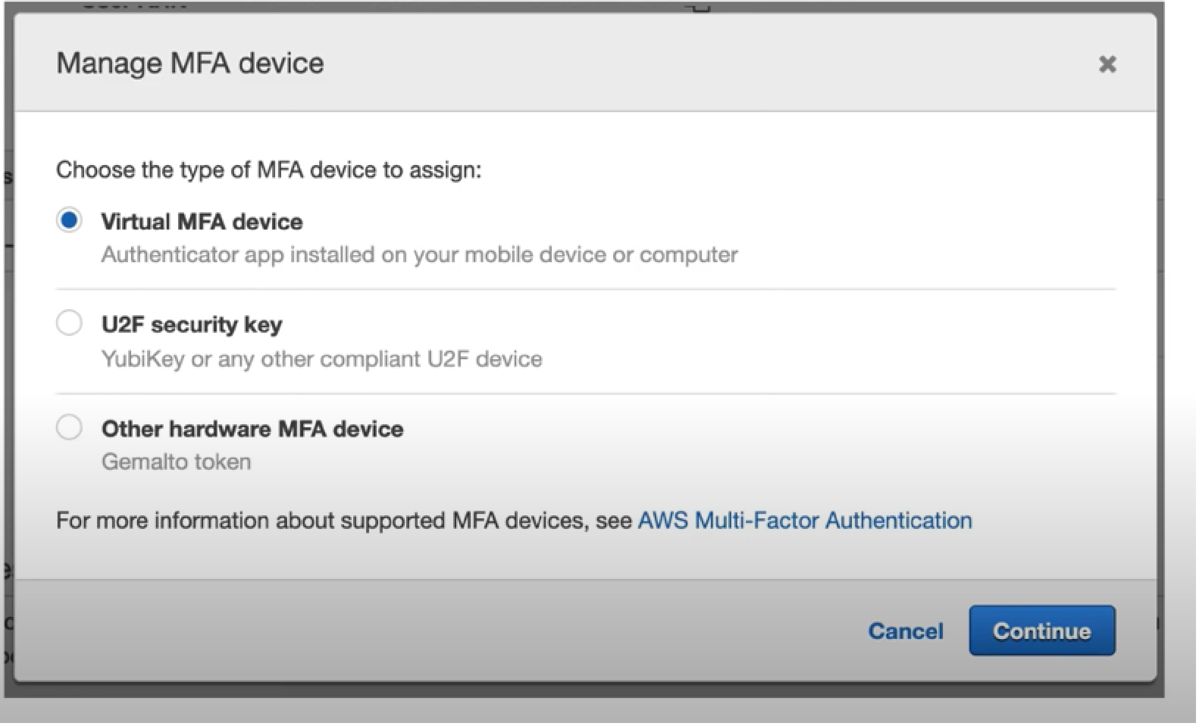
Multi-Factor Authenication

IAM – MFA

Multi-factor authentication (MFA) can be turned on per user.

The user has to turn on MFA themselves, Administrator cannot directly enforce users to have MFA.

They Administrator account could create a policy requiring MFA to access certain resources.



Manage MFA device ×

Choose the type of MFA device to assign:

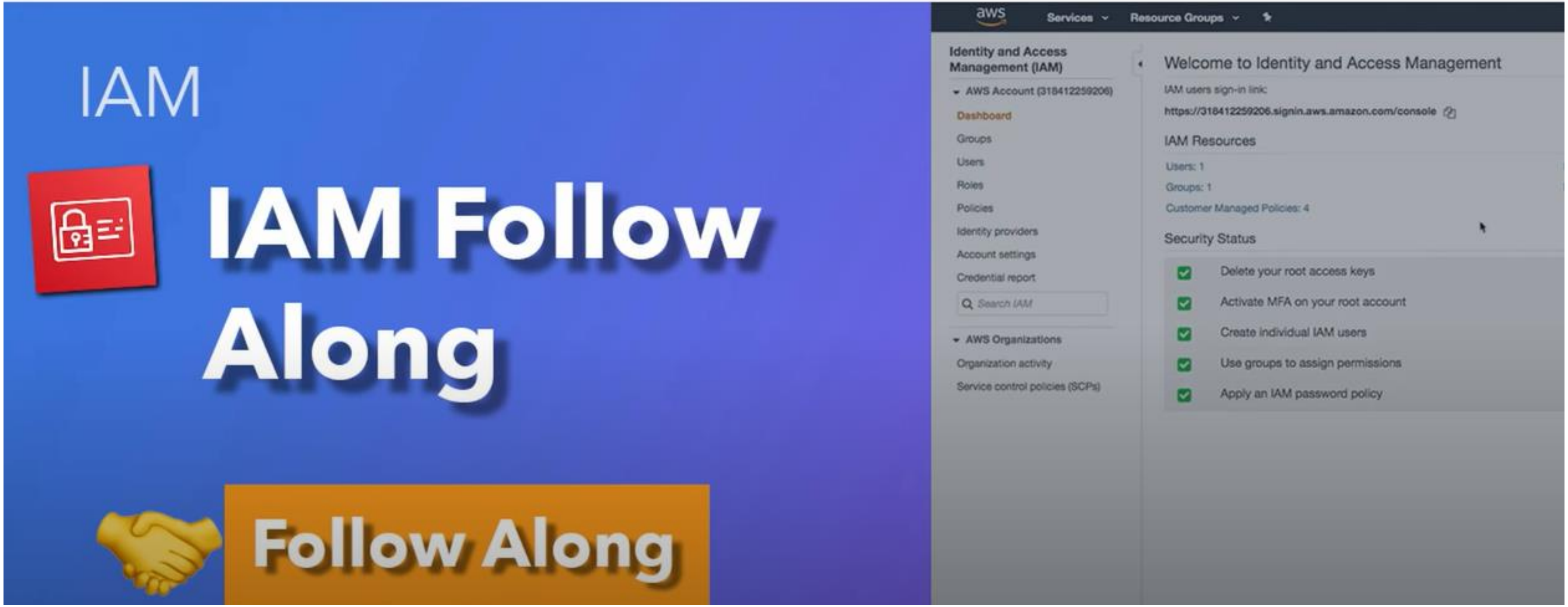
● **Virtual MFA device**
Authenticator app installed on your mobile device or computer

○ **U2F security key**
YubiKey or any other compliant U2F device

○ **Other hardware MFA device**
Gemalto token

For more information about supported MFA devices, see AWS Multi-Factor Authentication

AWS Solutions Architect Associate



Identity and Access Management (IAM)

- AWS Account (318412259206)
- Dashboard
- Groups
- Users
- Roles
- Policies
- Identity providers
- Account settings
- Credential report

Search IAM

- AWS Organizations
- Organization activity
- Service control policies (SCPs)

Welcome to Identity and Access Management

IAM users sign-in link:

https://318412259206.signin.aws.amazon.com/console

IAM Resources

Users: 1

Groups: 1

Customer Managed Policies: 4

Security Status

- ☑ Delete your root access keys
- ☑ Activate MFA on your root account
- ☑ Create individual IAM users
- ☑ Use groups to assign permissions
- ☑ Apply an IAM password policy

AWS Solutions Architect Associate

IAM

IAM Cheat Sheet

IAM CheatSheet

- **Identity Access Management** is used to manage **access** to users and resources
- IAM is a universal system. (applied to all regions at the same time). IAM is a free service
- A root account is the account initially created when AWS is set up (full administrator)
- New IAM accounts have no permissions by default until granted
- New users get assigned an Access Key Id and Secret when first created when you give them programmatic access
- Access Keys are only used for CLI and SDK (cannot access console)
- Access keys are only shown once when created. If lost they must be deleted/recreated again.
- Always setup MFA for Root Accounts
- Users must enable MFA on their own, Administrator cannot turn it on for each user
- IAM allows your set password policies to set minimum password requirements or rotate passwords
- **IAM Identities** as Users, Groups, and Roles
- **IAM Users** End users who log into the console or interact with AWS resources programmatically
- **IAM Groups** Group up your Users so they all share permission levels of the group
- eg. Administrators, Developers, Auditors
- **IAM Roles** Associate permissions to a Role and then assign this to an Users or Groups
- **IAM Policies** JSON documents which grant permissions for a specific user, group, or role to access services. Policies are attached to to IAM Identities
- **Managed Policies** are policies provided by AWS and cannot be edited
- **Customer Managed Policies** are policies created by use the customer, which you can edit
- **Inline Policies** are policies which are directly attached to a user

AWS Solutions Architect Associate

Cognito

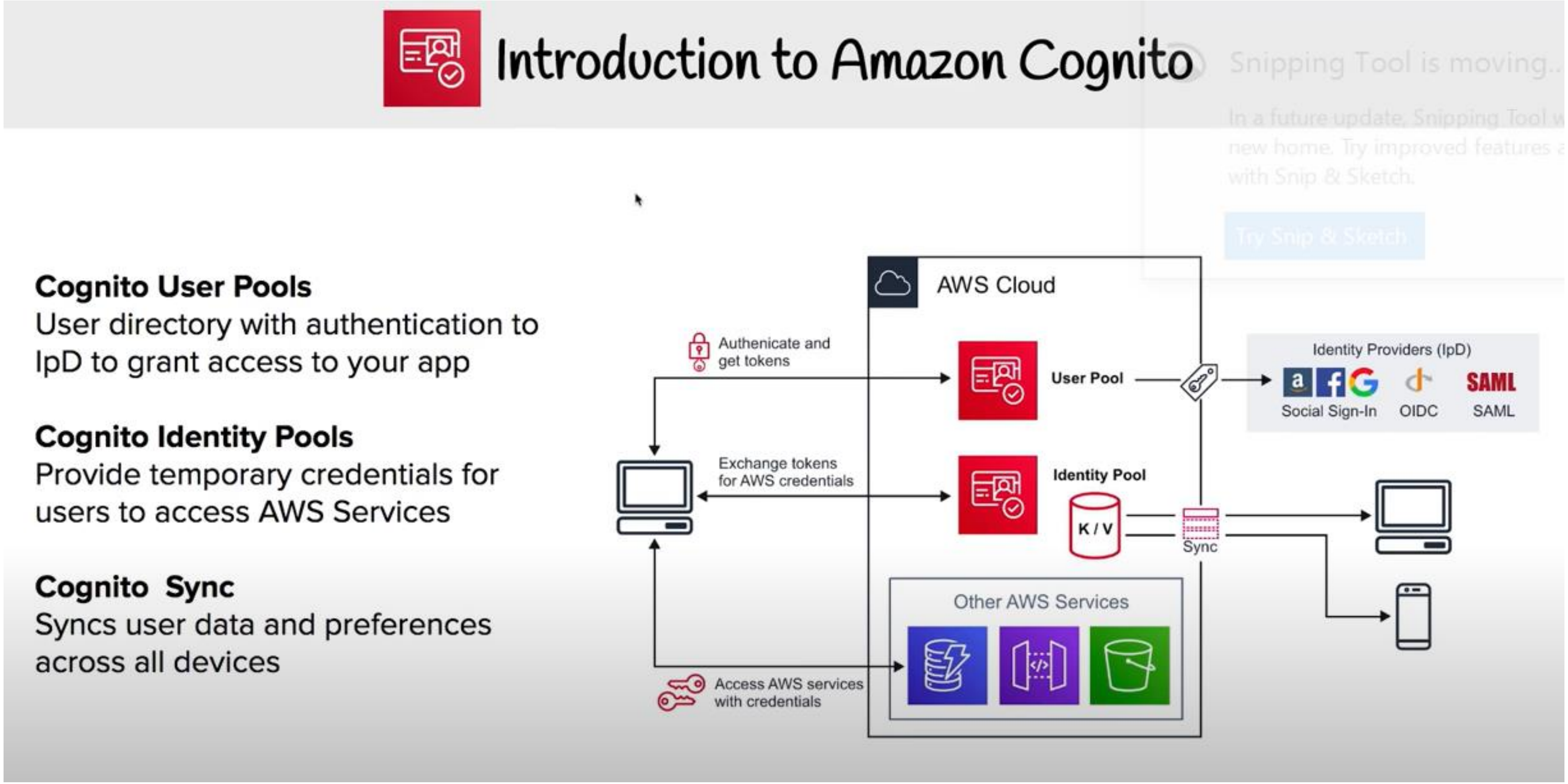
Amazon Cognito Introduction

Amazon Cognito

Decentralized Managed **Authentication.**

Sign-up, sign-in integration for your apps.

Social identity provider eg. Facebook, Google.



Introduction to Amazon Cognito

Snipping Tool is moving...

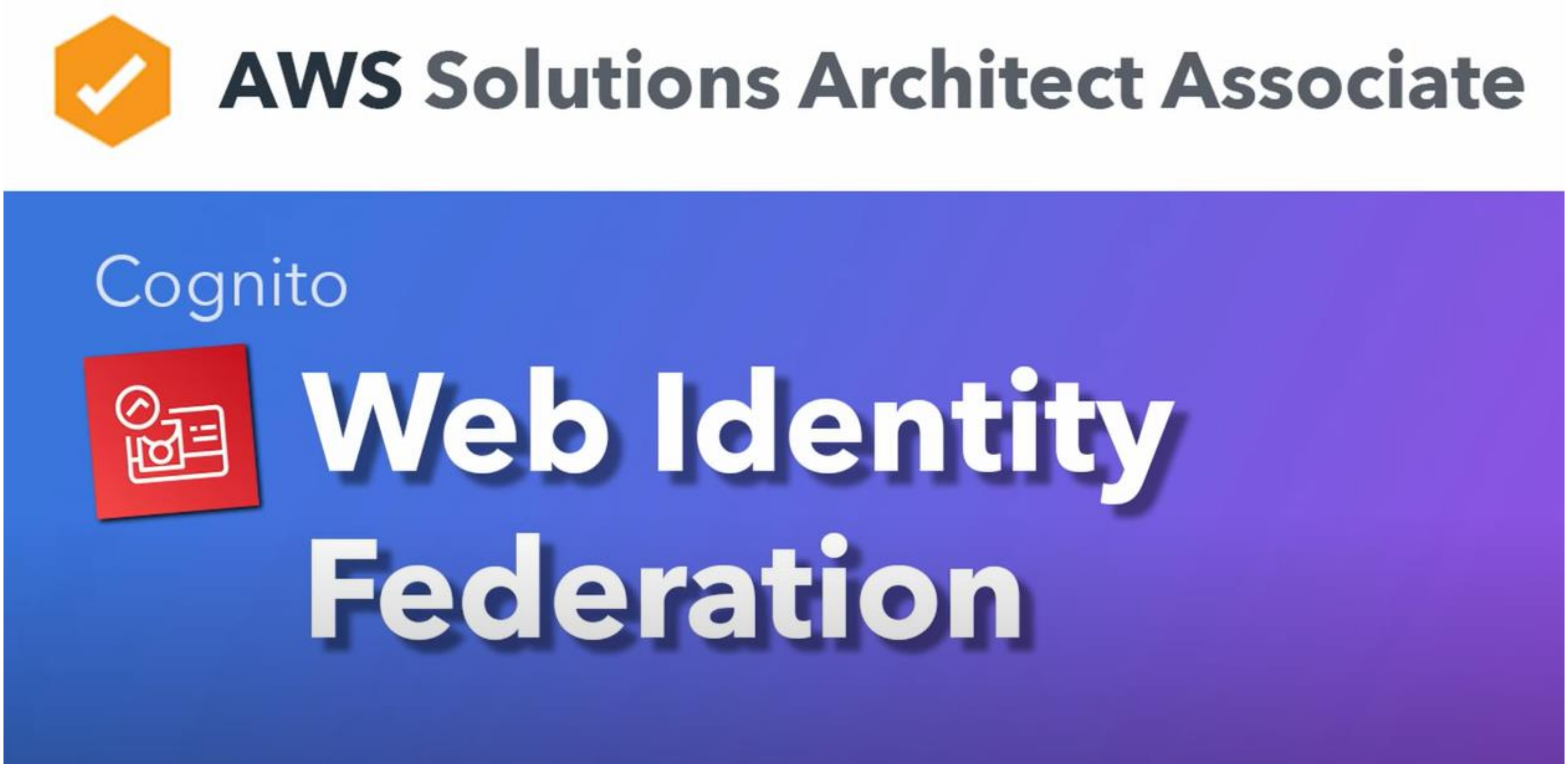
In a future update, Snipping Tool w
new home. Try improved features a
with Snip & Sketch.

Try Snip & Sketch

Cognito User Pools
User directory with authentication to IpD to grant access to your app

Cognito Identity Pools
Provide temporary credentials for users to access AWS Services

Cognito Sync
Syncs user data and preferences across all devices



AWS Solutions Architect Associate

Cognito

Web Identity Federation

Web Identity Federation and IpD

Web Identity Federation
To exchange identity and security information between an identity provider (IdP) and an application

Identity Provider (IdP)
a trusted provider of your user identity that lets you use authenticate to access other services.
Identity Providers could be: **Facebook, Amazon, Google, Twitter, Github, LinkedIn**

Types of Identity Providers

The technology that behind the Identity Providers

Security Assertion Markup Language (SAML)
Single Sign On (SSO)

SAML

OpenID Connect (OIDC)
OAuth

OpenID

This is for **Web** Identity Federation

AWS Solutions Architect Associate

Cognito

User Pools

Cognito User Pools

User Pools are user directories used to manage the actions for web and mobile apps such as:

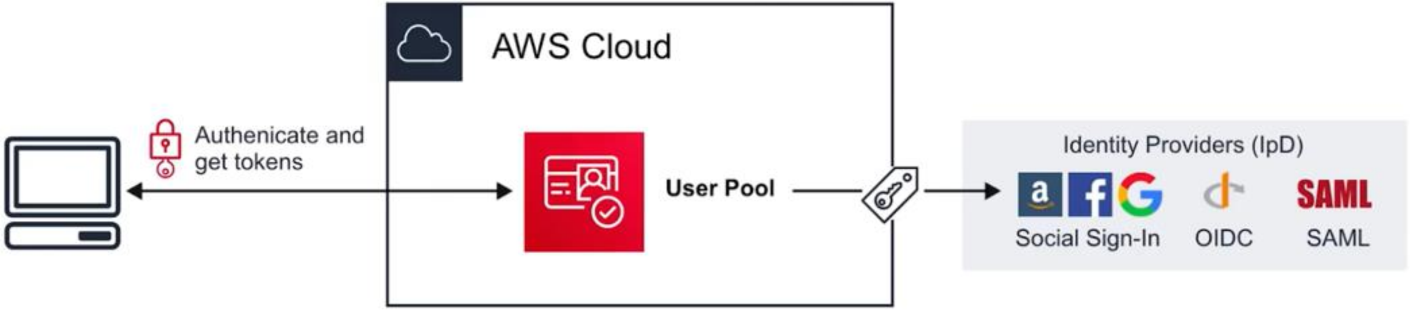
- **Sign-up**
- **Sign-in**
- **Account recovery**
- **Account confirmation**

Allows users to sign-in directly to the User Pool, or using Web Identity Federation.

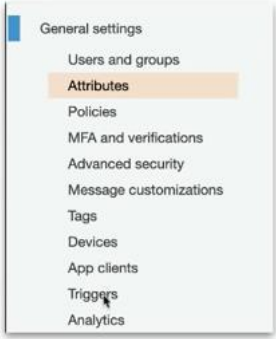
Uses AWS Cognito as the identity broker between AWS and the identity provider.

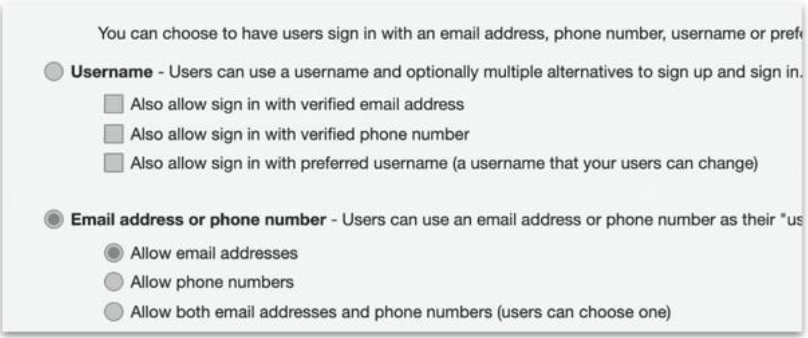
Successful user authentication generates a JSON Web Token (JWTs).

User Pools can be thought of as the account used to access the system (ie email address and password)



Cognito User Pools





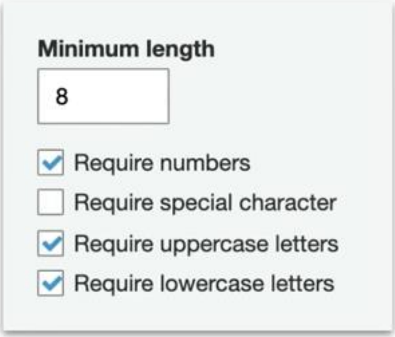
You can choose to have users sign in with an email address, phone number, username or prefe

○ **Username** - Users can use a username and optionally multiple alternatives to sign up and sign in.

- □ Also allow sign in with verified email address
- □ Also allow sign in with verified phone number
- □ Also allow sign in with preferred username (a username that your users can change)

● **Email address or phone number** - Users can use an email address or phone number as their "us

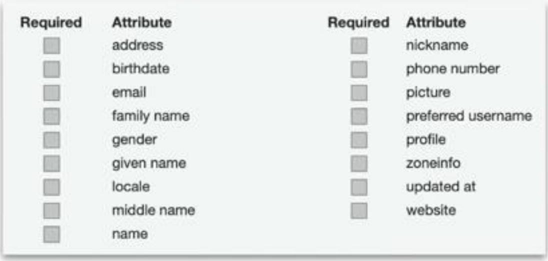
- ● Allow email addresses
- ○ Allow phone numbers
- ○ Allow both email addresses and phone numbers (users can choose one)



Minimum length

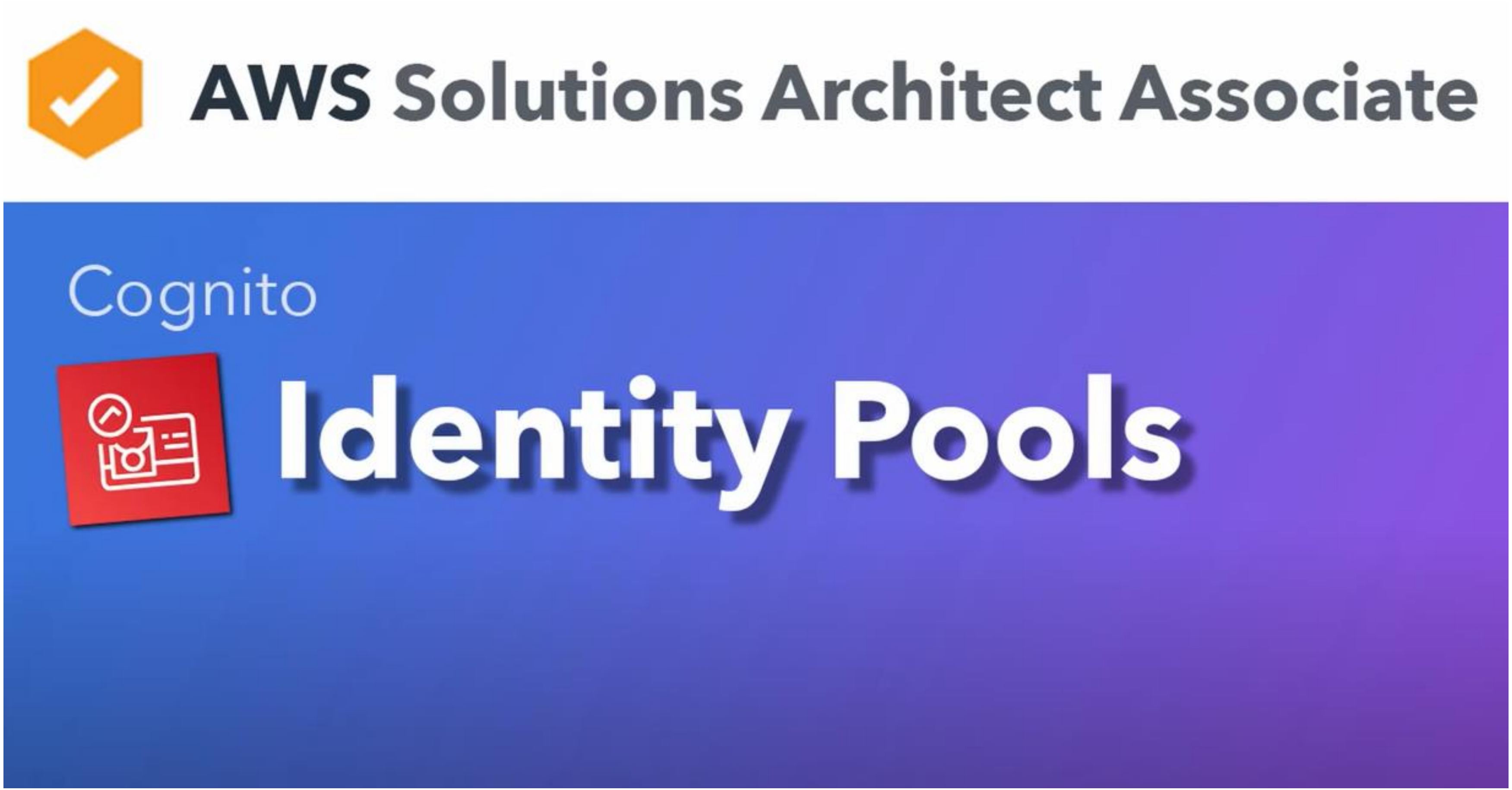
8

- ☑ Require numbers
- □ Require special character
- ☑ Require uppercase letters
- ☑ Require lowercase letters



Required	Attribute	Required	Attribute
□	address	□	nickname
□	birthdate	□	phone number
□	email	□	picture
□	family name	□	preferred username
□	gender	□	profile
□	given name	□	zoneinfo
□	locale	□	updated at
□	middle name	□	website
□	name		

- Choose what attributes
- Choose password requirements
- Apply MFA
- Restrict whether users are allow to sign up on their own or need admin verification
- Analytics with PinPoint for user campaigns
- Trigger custom log via Lambdas after actions such as after signup



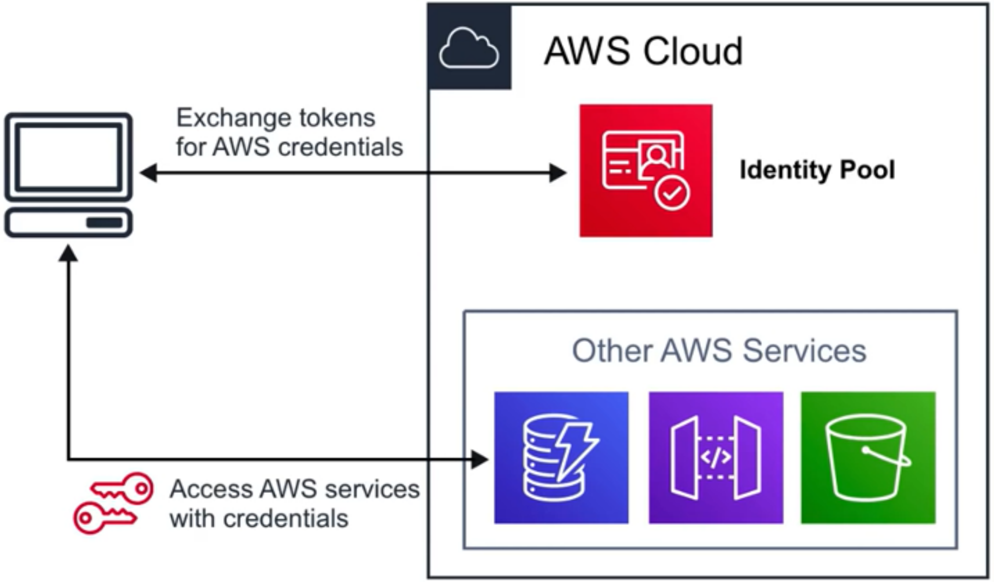
AWS Solutions Architect Associate

Cognito

Identity Pools

Cognito Identity Pools

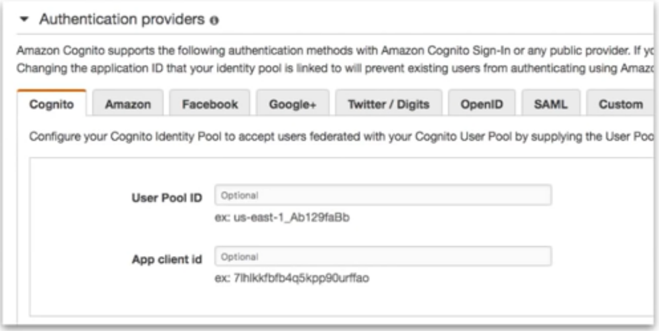
Identity Pools provide **temporary AWS credentials** to access services eg. S3, DynamoDB
Identity Pools can be thought of as the actual mechanism authorizing access to the AWS resources.



AWS Cloud
Exchange tokens for AWS credentials
Identity Pool
Other AWS Services
Access AWS services with credentials

Cognito Identity Pools

Choose who to provide access to:



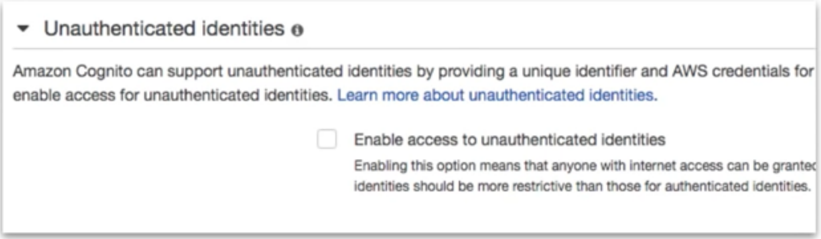
Authentication providers

Amazon Cognito supports the following authentication methods with Amazon Cognito Sign-In or any public provider. If yc
Changing the application ID that your identity pool is linked to will prevent existing users from authenticating using Amazc

Configure your Cognito Identity Pool to accept users federated with your Cognito User Pool by supplying the User Poo

User Pool ID: Optional
ex: us-east-1_Ab129faBb

App client id: Optional
ex: 7lhlkkfbfb4q5kpp90urffao



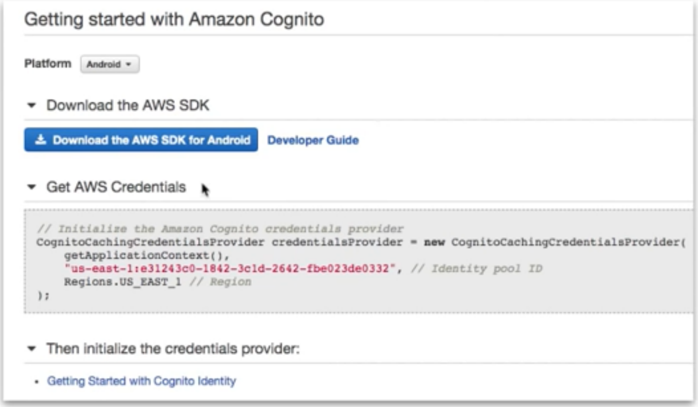
Unauthenticated identities

Amazon Cognito can support unauthenticated identities by providing a unique identifier and AWS credentials for
enable access for unauthenticated identities. Learn more about unauthenticated identities.

□ Enable access to unauthenticated identities

Enabling this option means that anyone with internet access can be grantec
identities should be more restrictive than those for authenticated identities.

Use the SDK to get temporary credentials



Getting started with Amazon Cognito

Platform: Android

Download the AWS SDK

Get AWS Credentials

```
// Initialize the Amazon Cognito credentials provider
CognitoCachingCredentialsProvider credentialsProvider = new CognitoCachingCredentialsProvider(
    getApplicationContext(),
    "us-east-1:e31243c0-1842-3c1d-2642-fbe023de0332", // Identity pool ID
    Regions.US_EAST_1 // Region
);
```

Then initialize the credentials provider:

- Getting Started with Cognito Identity

AWS Solutions Architect Associate
Cognito
Cognito Sync

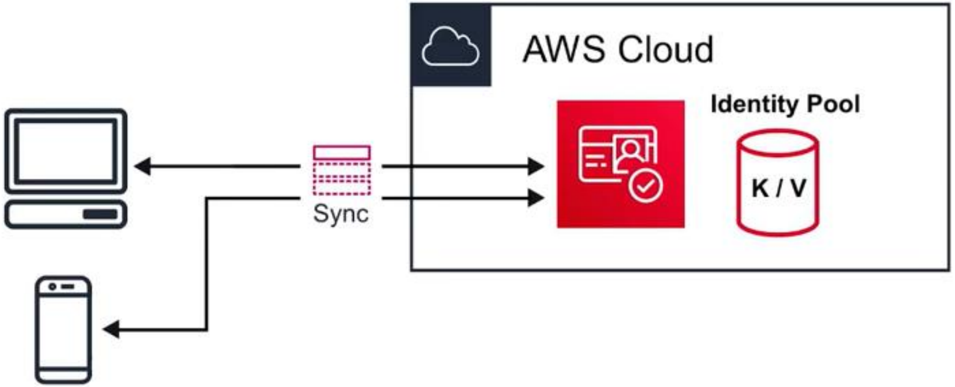
Cognito – Sync

Sync **user data** and **preferences** across devices with one line of code

Cognito uses **push synchronization** to push updates and synchronize data

Uses Simple Notification Service (SNS) to send notifications

to all user devices when data in the cloud changes.



AWS Solutions Architect Associate

Cognito

Cognito Cheat Sheet

Cognito *CheatSheet*

- Cognito is decentralized managed authentication system. When you need to easily add authentication to your mobile and desktop app *think* Cognito
- **User Pools** user directory, allows users to authenticate using OAuth to IpD such as Facebook, Google, Amazon to connect to web-applications. Cognito User Pool is in itself a IpD
- User Pools use **JWTs** for to persist authentication
- **Identity Pools** provide **temporary AWS credentials** to access services eg. S3, DynamoDB
- **Cognito Sync** can sync **user data** and **preferences** across devices with one line of code (powered by SNS)
- **Web Identity Federation** exchange identity and security information between an identity provider (IdP) and an application
- **Identity Provider (IdP)** a trusted provider of your user identity that lets you use authenticate to access other services. eg. Facebook, Twitter, Google, Amazon
- **OIDC** is a type of Identity Provider which uses Oauth
- **SAML** is a type of Identity Provider which is used for Single Sign-on

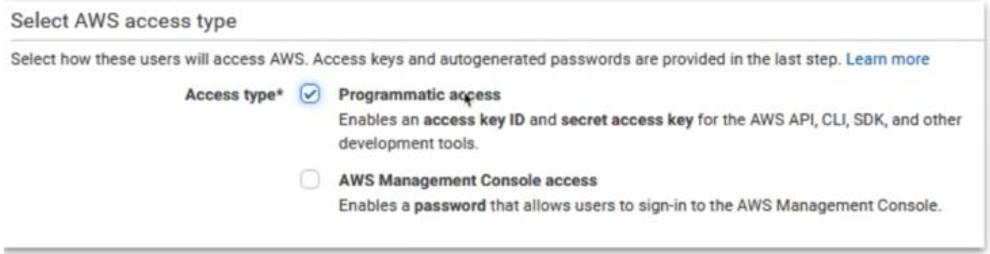
AWS Solutions Architect Associate

AWS CLI & SDK

Programmatic Access
Access Key and Secret

Programmatic Access – Access Key and Secret

When you enable **Programmatic Access** for AWS users



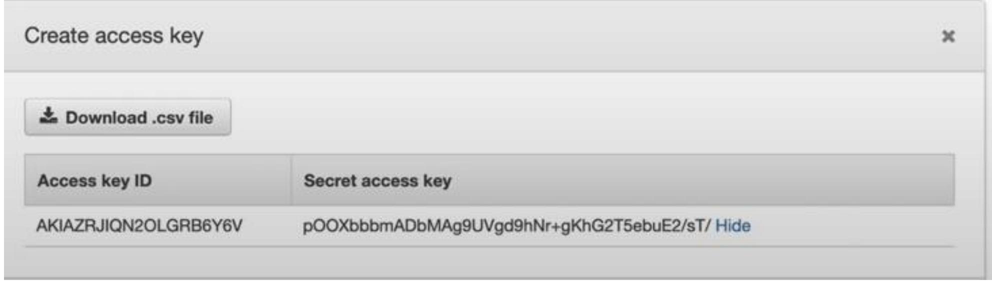
Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. Learn more

Access type* ☑ **Programmatic access**
Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

☐ **AWS Management Console access**
Enables a **password** that allows users to sign-in to the AWS Management Console.

You'll have the ability create **Access Key ID** and **Secret Access Key**
These are collectively known as **AWS Credentials**

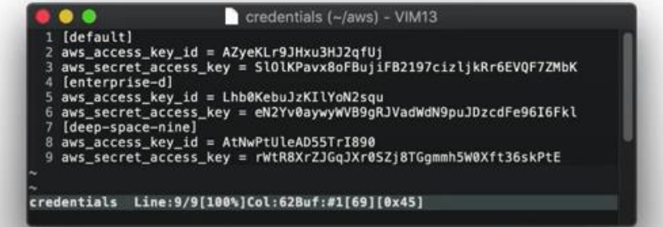


Create access key

Access key ID	Secret access key
AKIAZRJIQN2OLGRB6Y6V	pOOXbbbmADbMAg9UVgd9hNr+gKhG2T5ebuE2/sT/ Hide

You will want to stored you credentials in your user's home eg. ~/.aws/credentials

The credentials files allow you to manage multiple credentials (called profiles)



credentials (~/aws) - VIM13

```
[default]
aws_access_key_id = AZyeKLr9JHxu3HJ2qfUj
aws_secret_access_key = SlOlKPavx8oFBujiFB2197cizljkRr6EVQF7ZMbK
[enterprise-d]
aws_access_key_id = Lhb0KebuJzKIlYoN2squ
aws_secret_access_key = eN2Yv0aywyWVB9gRJVadWdN9puJDzcdFe96I6Fkl
[deep-space-nine]
aws_access_key_id = AtNwPtUleAD55TrI890
aws_secret_access_key = rWtR8XrZJGqJXr0SZj8TGgmmh5W0Xft36skPtE
~
~
credentials  Line:9/9[100%]Col:62Buf:#1[69][0x45]
```

AWS CLI & SDK CheatSheet

- **CLI** stands for Command Line Interface
- **SDK** stands for Software Development Kit
- The **AWS CLI** lets you interact with AWS from anywhere by simply using a command line
- The **AWS SDK** is a set of API libraries that let you integrate AWS services into your applications.
- **Programmatic Access** must be enabled per user via the IAM console to use CLI or SDK
- **aws configure** command used to setup your AWS credentials for the CLI
- The CLI is installed via a Python script
- Credentials get stored in a plain text file (whenever possible use roles instead of AWS credentials)
- The SDK is avaliable for the following programming languages
    - C++
    - Go
    - Java
    - Javascript
    - .NET
    - NodeJs
    - PHP
    - Python
    - Ruby