AWS Identity and Access Management

Objectives
Key objectives of this chapter ▪ Overview of the AWS Identity and Access Management Service

1.1 AWS Identity and Access Management (IAM)

- IAM is the AWS user management, authentication and authorization service
- It manages users and their permissions within your AWS account through **users**, **groups**, and **roles** by applying security **policies**
- IAM uses AWS Key Management Service (KMS) to create and control the user encryption keys
- IAM is natively integrated into all the AWS Services
- AWS offers IAM for no charge

1.2 Working with IAM

- You interface with IAM using:
  - AWS Management Console
  - AWS CLI (the *aws* tool)
  - AWS SDKs

1.3 Need a Directory Service?

- Use AWS Directory Service, which is Microsoft's Active Directory-compliant
  - You can use it to integrate with your on-prem AD

1.4 Need Identity Management for Mobile Apps?

- To enable your clients to sign-up and sign-in to your mobile and web app, use Amazon Cognito
- With Cognito, you have the options to authenticate users through SAML identity solutions, social identity providers such as Facebook and Twitter
- With Cognito you can synchronize data across your devices

Notes:

Source: https://aws.amazon.com/cognito/

1.5 Root Account Access vs. IAM User Access

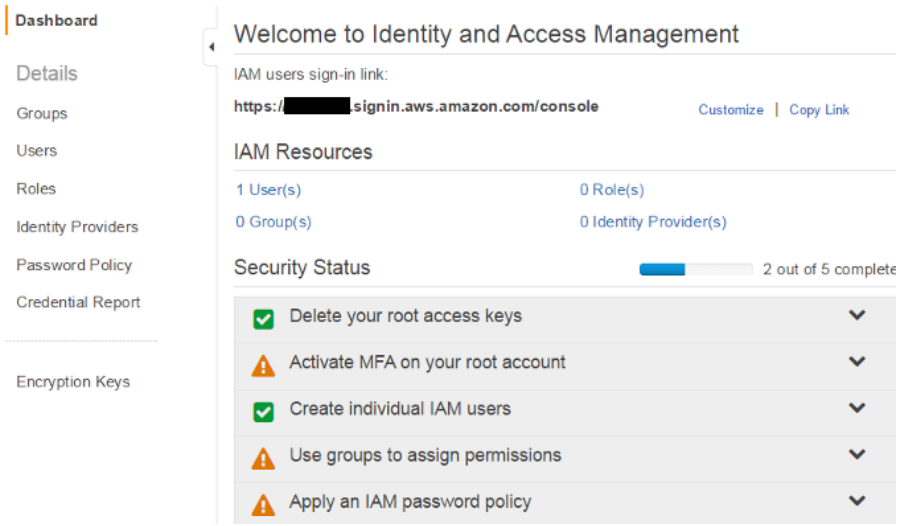
- AWS root account is created when you sign up with AWS
- You cannot control this account privileges which have full access to AWS resources
- IAM users are created by the root account in line with the least privilege principle

- ◇ All permissions are implicitly denied by default unless explicitly allowed
- ◇ There is no default permissions

- The root controls IAM users permissions by revoking or modifying them; an IAM user can also be removed altogether

1.6 The IAM Dashboard

- Access to the IAM Dashboard is available from the AWS management console
- The IAM Dashboard provides a single point of control of all the aspects of identity and access management



1.7 AWS Key Management Service (KMS)

- KMS is the focal point for managing user encryption keys, which include such operations as
  - ◇ Create, enable, disable and audit the use of keys

- You need to define an IAM user and role within an account to administer keys
- Keys are created within an AWS Region
  - The key has the region id prefixed to it (this concept is referred to as the ARN (Amazon Resource Name))
    - e.g.: **arn:aws:kms:us-east-1:xxxxxxxxx**
- EBS, S3 and other services use KMS for data encryption services
- KMS Software Development Kit (SDK) enables you to incorporate encryption in your applications as well

Notes:

IMS audit trails are captured in log files with such recorded information as which key was used to access which data along with the resource access timestamp.

The audit trails enable you to meet compliance and regulatory requirements.

1.8 User Management

- IAM performs the following standard user management operations:
  - Manage user password
    - IAM can assign an auto-generated password
    - The password can be setup to expire at next sign-in
  - Manages access keys
    - Access keys enable you to make secure REST, command line interface (CLI), the AWS SDKs calls to any AWS service API
  - Manage signing certificates
  - Delete user

- Add user to groups

Notes:

AWS administers access keys for IAM users as follows:

"When you create the access key, IAM returns the access key ID and a secret access key. You should save these in a secure location and give them to the user. To ensure the security of your AWS account, the secret access key is accessible only when you create the access key. If a secret key is lost, you can delete the access key for the associated user and then create a new key.

By default, when you create an access key, its status is Active, which means the user can use the access key for API calls. Each user can have two sets of active keys, which is useful when you need to rotate the user's access keys. You can disable a user's access key, which means it can't be used for API calls. You might do this while you're rotating keys (for more information, see Rotating Credentials) or to revoke API access for a user."
[http://docs.aws.amazon.com/IAM/latest/UserGuide/ManagingCredentials.html]

1.9 Password Policies

- IAM supports user password policies (rules) to enhance password strength
- The following password rules are available (which can be used in combination):
  - 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

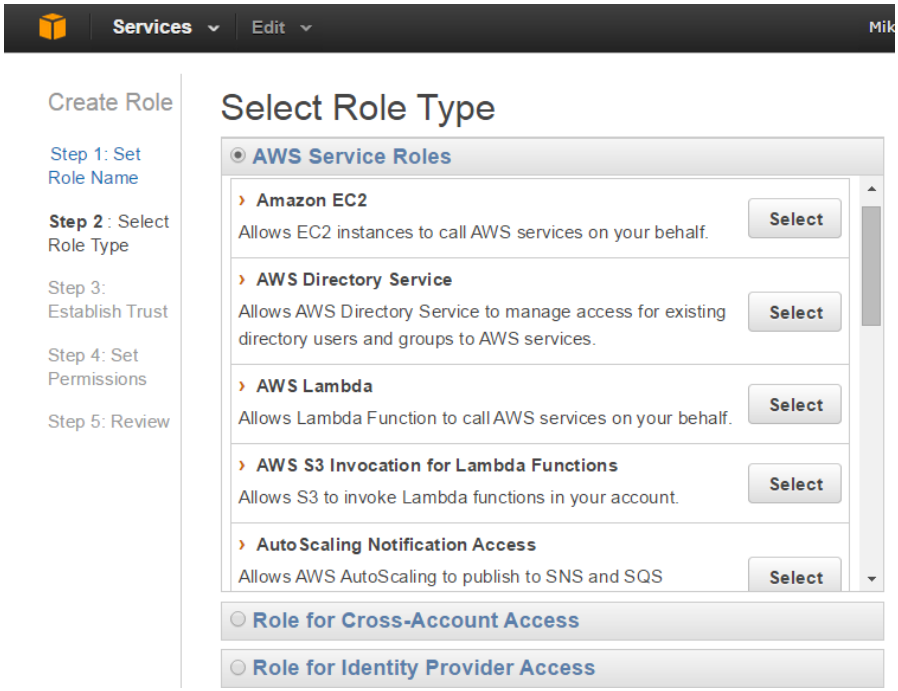
- ✔ Number of passwords to remember
- ◇ Password expiration requires administrator reset

1.10 Groups

- Groups help organize users into units managed together
- Group operations include:
  - ◇ Delete / Rename group
  - ◇ Add / Remove users to / from a group
- Group permissions are defined in IAM policies
- There is no default groups
- Groups cannot be nested
- A single IAM user can belong to multiple groups

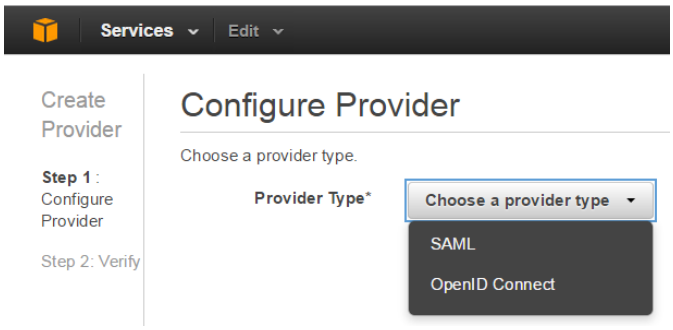
1.11 Roles

- Roles help with role-based access to and trust relationships with AWS resources



1.12 Identify Providers

- IAM supports integration with other Identify Providers
- Single Sign-On to the AWS Management Console using the Security Assertion Markup Language (SAML) 2.0 was introduced at the end of 2013
  - The SAML option enables you to use your organization's existing identity provider without the need to provide AWS credentials when signing in to AWS



Notes:

In October 2014, AWS announced support for OpenID Connect Support for Amazon Cognito[1].

The idea behind this integration is to enable app developers to make use of identities from any identity provider that supports OpenID Connect (OIDC)[2] within the AWS Cloud.

[1] Cognito Amazon Cognito service allows users to save data in the AWS Cloud without using the AWS SDK or managing any infrastructure calls.

[2] OIDC is an open standard for using external identity providers for user authentication.

1.13 Utilizing SAML Providers

- As more businesses move to the Cloud, it is critical for enterprises to connect user identities to cloud applications with both security and convenience. PingFederate, Microsoft ADFS, Centrify, and many others can be deployed on a virtual server, known as an Amazon EC2 instance, to provide identity and access management in a cloud environment.
- Setting such an environment up can be fairly simple or very complex. An example of the steps to setup a Java-based SAML solution are noted below
  - Create an SSH key pair.
  - Set up a Security Group.
  - Provision and launch an Amazon EC2 instance.

- Configure the operating system.
- Install the Oracle Java SE Development Kit (JDK).
- Download the SAML provider installation directly to server and unzip or install the provider.
- Start the SAML Provider solution.
- Verify through all use cases.
- Implement in production for cloud authentication and authorization connecting your on-premise or private cloud with your PaaS/IaaS/Saas/XaaS solution provider

1.14 Using Multi-Factor Authentication Devices

- Multi-Factor Authentication (MFA) greatly enhances security by requiring additional piece of credentials in the form of a unique authentication code generated in an authentication device when accessing the AWS Cloud
- When MFA is enabled, users are required to provide the (usual) username and password (treated as the first factor - something that the user knows), plus an authentication code from their AWS (the second factor - something that the user has)
- IAM provides steps to help you setup and assign an MFA device to the IAM user
- Users should not re-use MFA devices (devices should not be shared)

1.15 Hardware-based and Virtual MFA Devices

- The device can be hardware-based or virtual
  - A hardware device can be a keyfob or a display card (in a convenient shape, like a credit card)

- Hardware devices are "tamper-evident"

- A virtual device is any device on which you can install a "time-based one-time password" application; basically, it can be any smart phone (Android, Blackberry, iPhone, and Windows Phone)
  - Security of virtual devices is rated lower than that provided by physical devices

1.16 Summary

- IAM is the user management, authentication and authorization service
- IAM uses the AWS Key Management Service (KMS) for managing user encryption keys
- You can enhance security of your AWS cloud-based solutions by applying password policies and using the multi-factor authentication option supported by IAM