



## Table of Contents

- ▶ Introduction to IAM
- ▶ IAM - Users
- ▶ IAM - Policies
- ▶ IAM - User Groups
- ▶ IAM - Roles

### ▶ What Is IAM?



IAM = Intity & Access Management

#### Authentication Prove your identity

- Username + Password + [MFA]  
or
- Access Key + Secret Key  
or
- Access Key + Secret Key + Session Token

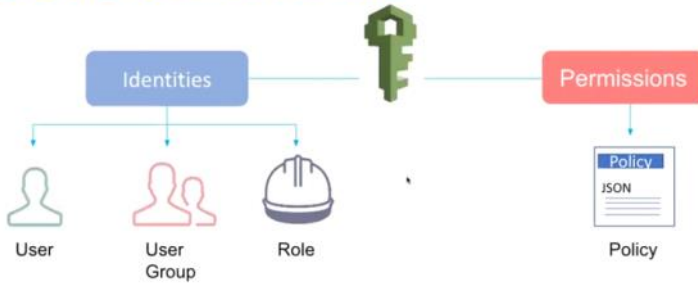
#### Authorization Permission to access resources

- IAM Policies  
and/or
- Resource Policies

KİMLİK VE GİRİŞ YÖNETİMİ diye düşünebiliriz.  
Sıklıkla karşımıza çıkacak iki kavram: Authentication ve Authorization.  
AWS kitabın ortasıdır.  
AWS genel olarak birçok servisi içinde barındırır.  
Authentication senin kim olduğunu sorgular.  
Authorization ise yetkileri belirler.

## Introduction to IAM

### Categorizing IAM Components



- IAM components can be mainly categorized under two terms; **Identities** and **Permissions**.

Kategoriye ayircak olsak 4 e ayirmak mumkun. Bunlarin en tepesinde Principle diye bir kavram var. Principle istekte bulunan demek. Her türlü özneyi ifade eder. 4 componenti 2 baslik altında inceleyecegiz. Identities sahip olarak tek olarak bütün olarak özneyi temsil eder. Permission ise bunlara atanan yetkiyi temsil eder. Biz AWS işlemlerini bir user olarak, userin dahil olduğu bir grup olarak veya role dedigimiz bir kavram ile gerçekleştiriyoruz. Console ya da CLI üzerinden. Ama bunlari yaparken yetkilerimi policyler belirliyor json formatındaki.

## 2 IAM Users

### IAM Users

#### What is IAM User?

(IAM) user is an entity that you create in AWS to represent the person or application that uses it to interact with AWS

Userlar AWS icinde yaratilan userlar olabilir. Ya da softwarelar, applicationlar service accountlari da bir user olabilir. Real kisilik olmak zorunda degil bir tek.



### IAM Users

#### What is Root User and IAM User.

#### AWS Account Owner - Root User (You)



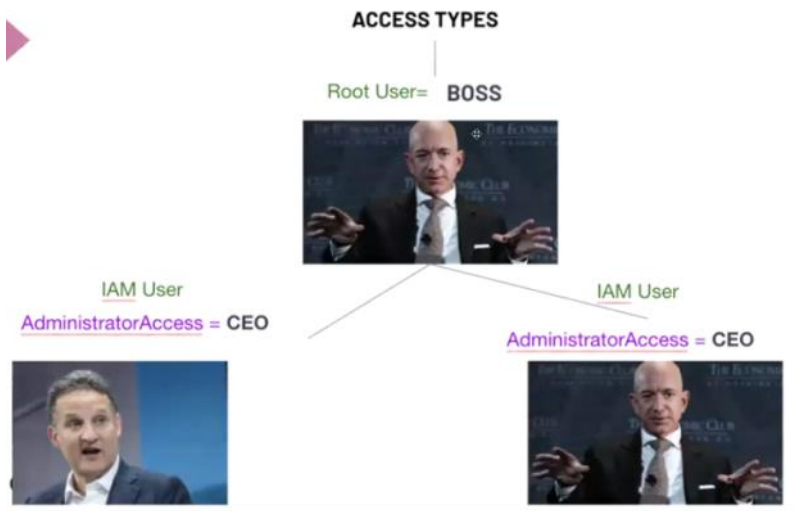
- Root User is a special user
- Username is email used to create account
- Generally, cannot limit permissions of Root User
- Cannot delete Root User
- Best practices:
  - Enable MFA for Root User
  - Don't use Root User for day-to-day work
  - Keep password in a secure location

User konusunda bizi esas ilgilendiren user cesitleri. Root user tum erisimlerin olduğu ve grup calismalarinda özellikle tavsiye edilmeyen user sekli. Root user sudo gibi herseyi yapabiliyor. Bu durumda yeteriligi olmayan birisi root userdan uygulamalar yaparken yanlislikla bir seyleri silebilir ya da kotu niyetli kisiler istediklerini buradan gerçeklestirebilirler. Iste bu noktada da IAM ihtiyaci doguyur. Email ile giris yapiyoruz. Sinirsiz limiti var rootun. Root useri silemiyorsun. Mfa ile dogrulamayi kullaniyoruz Day to day worklerine uygun degil. Passwordu security icin tutmak lazim ozelde.

## IAM Users Access Types



Interview için mesela root kullanmak büyük bir sikinti.  
IAM user için iki türlü giriş söz konusu: Konsol üzerinden ROOT- IAM user olarak ya da CLI üzerinden.



Bossun kendisi de dahil olmak üzere root kullanmıyor. Herkese bir IAM user hesabı açıyor.

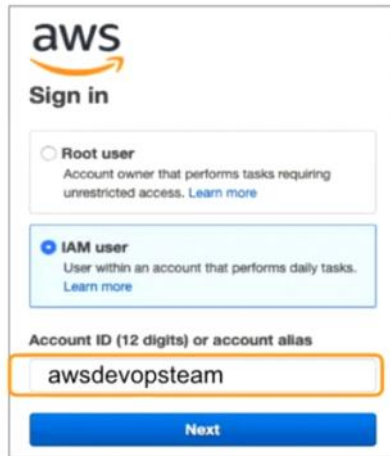
## IAM Users

### Sign in with Root User- AWS Management Console Access

The diagram shows the two-step process for signing in as the Root User. The first screenshot shows the 'Sign in' page with options for 'Root user' (selected) and 'IAM user'. The 'Root user email address' field is populated with 'osvaldo@clarusway.com'. A large blue arrow points to the second screenshot, which is the 'Root user sign in' page. This page shows the 'Email' field with 'osvaldo@clarusway.com' and a 'Password' field with masked characters. Below the password field is a 'Sign in' button. At the bottom of the second screenshot, there are links for 'Sign in to a different account' and 'Create a new AWS account'.

## ► IAM Users

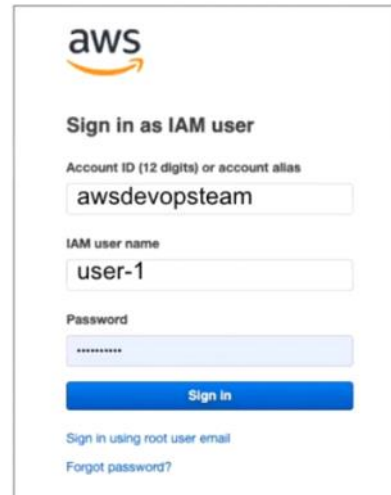
### Sign in with IAM User- AWS Management Console Access



The screenshot shows the AWS Management Console sign-in page. It has the AWS logo and 'Sign in' text. There are two radio buttons: 'Root user' (unselected) and 'IAM user' (selected). Below the radio buttons is a text input field containing 'awsdevopsteam'. At the bottom is a blue 'Next' button.



Account ID/Alias  
User name  
Password



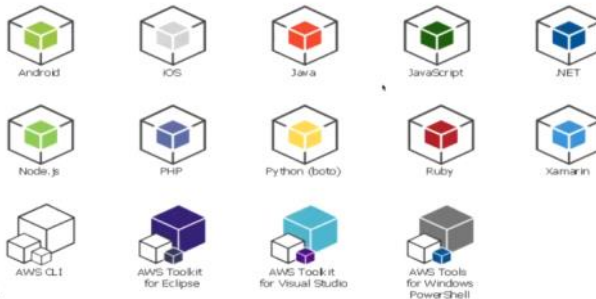
The screenshot shows the 'Sign in as IAM user' page. It has the AWS logo and 'Sign in as IAM user' text. There are three text input fields: 'Account ID (12 digits) or account alias' with 'awsdevopsteam', 'IAM user name' with 'user-1', and 'Password' with masked characters. Below the password field is a blue 'Sign in' button. At the bottom are links for 'Sign in using root user email' and 'Forgot password?'.

ACCOUNT ID : TAKIM ISMI  
IAM USER NAME: O TAKIMIN ICINDEKI SENIN ADIN NE  
PASSWORD: KENDI SIFREN

## ► IAM Users

### Sign in with IAM User- Programmatic Access

SDKs

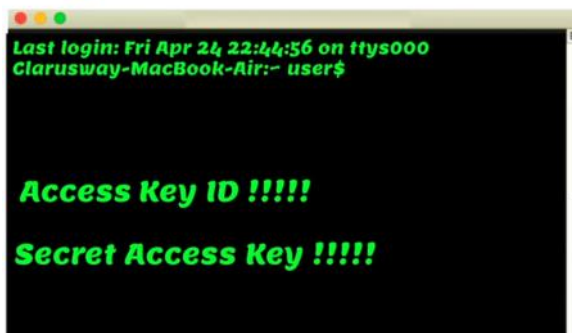


CLARUSWA

Programmatic access üzerinden giriş bizim çoğunlukla kullanacağımız şey siyah ekran olacak. Siyah ekranlarda SDK lar olacak. AWS ile yazılımlar arası iletişimi sağlamak için software development kitlelerini kullanıyor olacağız. AWS CLI bizim mesela linux üzerinden aws kullanmamıza yardımcı olacak kanal.

## ► IAM Users

### Sign in with IAM User- Programmatic Access



\*\*ROOT USER  
IAM USER.

CLI konsolda yaptığımız herşeyi buradan da yapabiliyoruz. CLI da aynı konsol gibi access key id ve secret access key istiyor. CLI konsoldan daha güçlüdür.

# 3

## IAM Policies

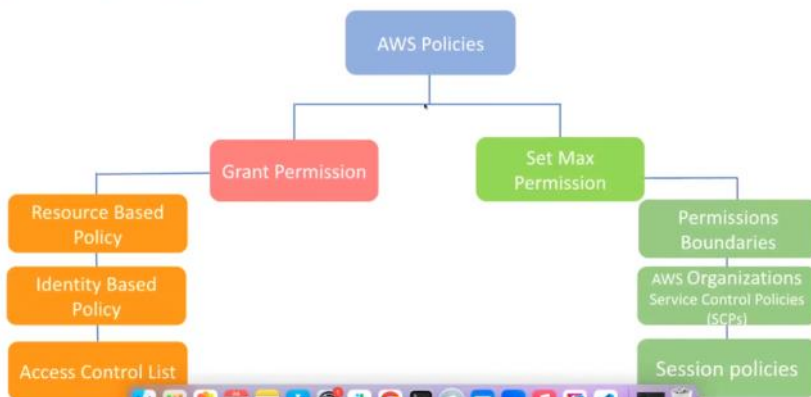
### IAM Policies What is a Policy?



Json formatinda yazilmis.  
Json policylerini identitye atariz. Boylece yetkilendirme yapmis oluruz.

- A policy is an object used to define the **permissions** of an identity or resource in AWS
- Permissions in the policies determine whether the request is **allowed** or **denied**.
- Policies are stored in AWS as **JSON** documents.

### Policy Types



Grant permissionlar verme uzerine kurallar.  
Set max permissionlar ise tirpanlamak uzerine yani kisitlamak uzerine kurulmus kurallar. Var olan haklardan kisitlayarak belirlenir.

### 3-Account Security: Security Policies



## Identity-based vs. Resource-based Policies



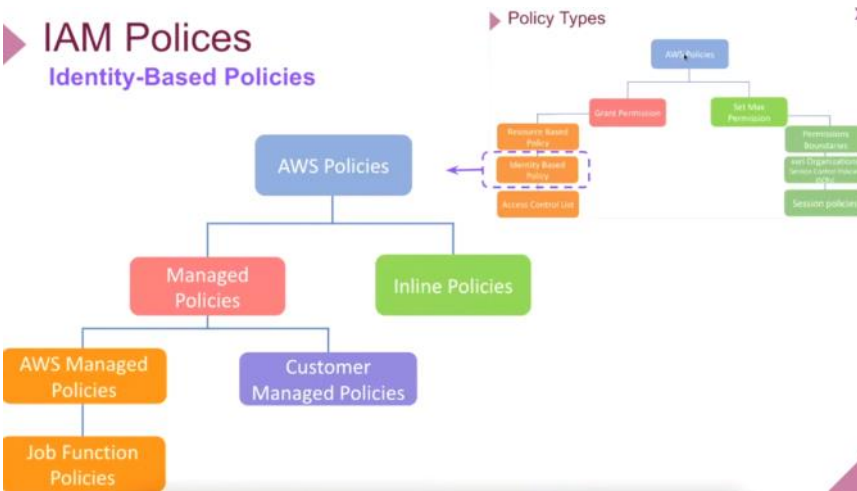
Identity role, user, gruplara atanıyor. atadığım yetkilere göre kişiler, gruplar ya da roller kullanım gösterebiliyorlar.

Resource ise kaynaga göre erişim sağlamak esas. Mesela s3 e su su özellikteki kullanıcılar erişim sağlayabilir gibi.

Bu ikisi birlikte kullanılıyor. Çift katmanlı.

## IAM Policies

### Identity-Based Policies



Identity based policle iki ayrılıyor: Managed ve inline policies. Managed policle ise ikiye ayrılıyor: AWS Managed policies (aws kendisi güncelliyor kuralları) ve customer managed policies (biz kendimiz customerlara göre düzenleme yapıyoruz). AWS managed altında ise job function policies yer alıyor. Inline policy kime atandığıyla ilgili kuralların.