# AWS IAM

CLARUSWAY
WAY TO REINVENT YOURSELF

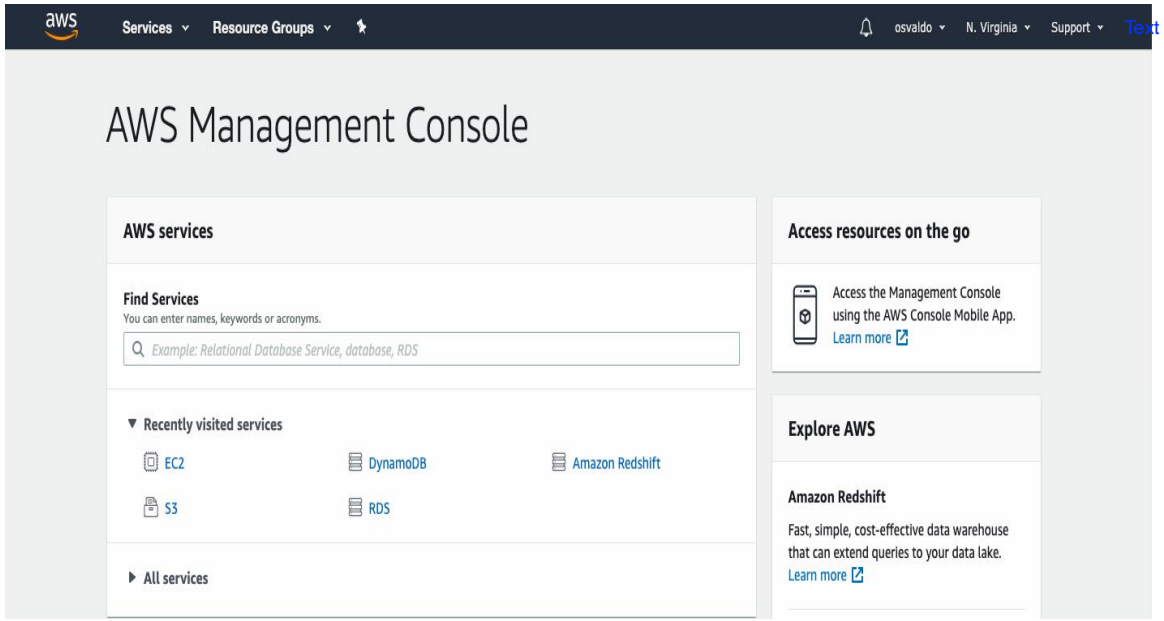
# Table of Contents


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

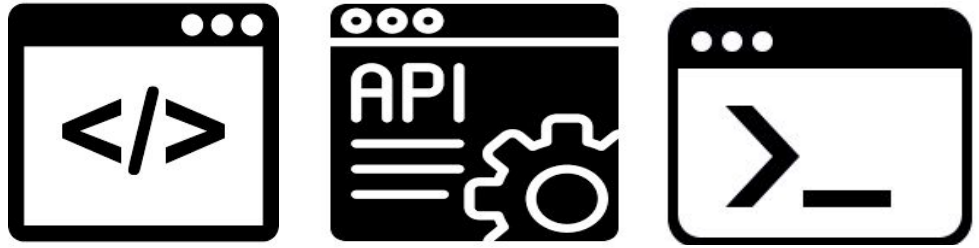
# 1 Introduction to IAM

# Accessing AWS


## AWS Management Console




## Programmatic Access

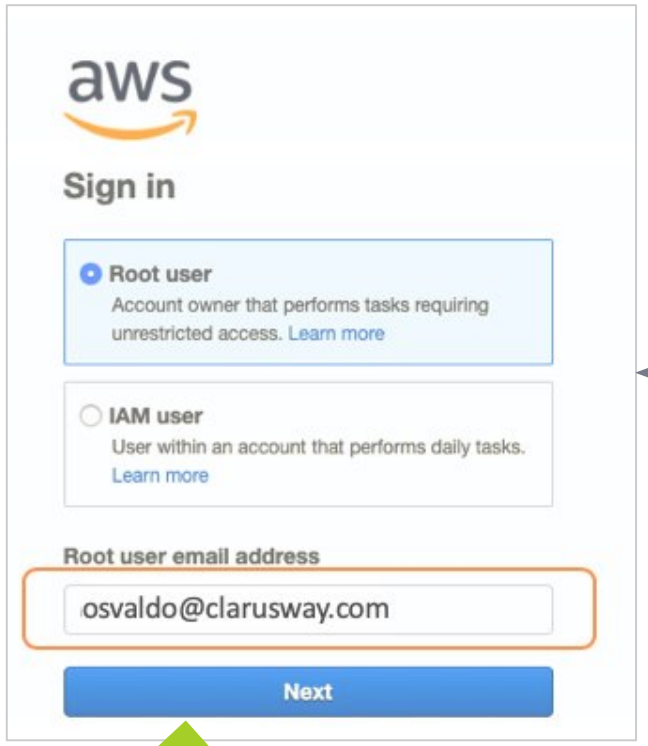



**CLI, SDK, API**

1. Comand Line Interface
2.Software Development Kids
3.Aplication Programing Interface

# Management Console

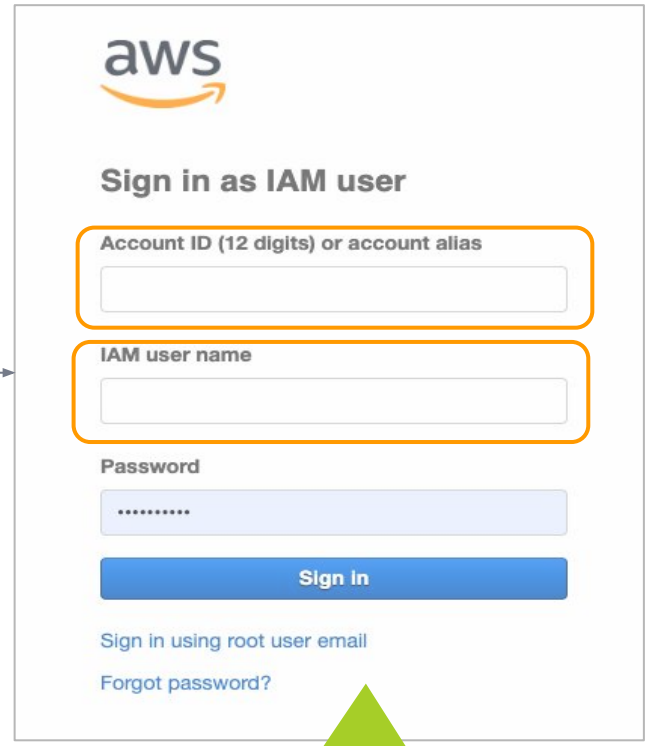
Console login for **Root User**




Log in to the console with **email address** as the username.

Require **Username & Password** to Authenticate

Console login for **IAM User**

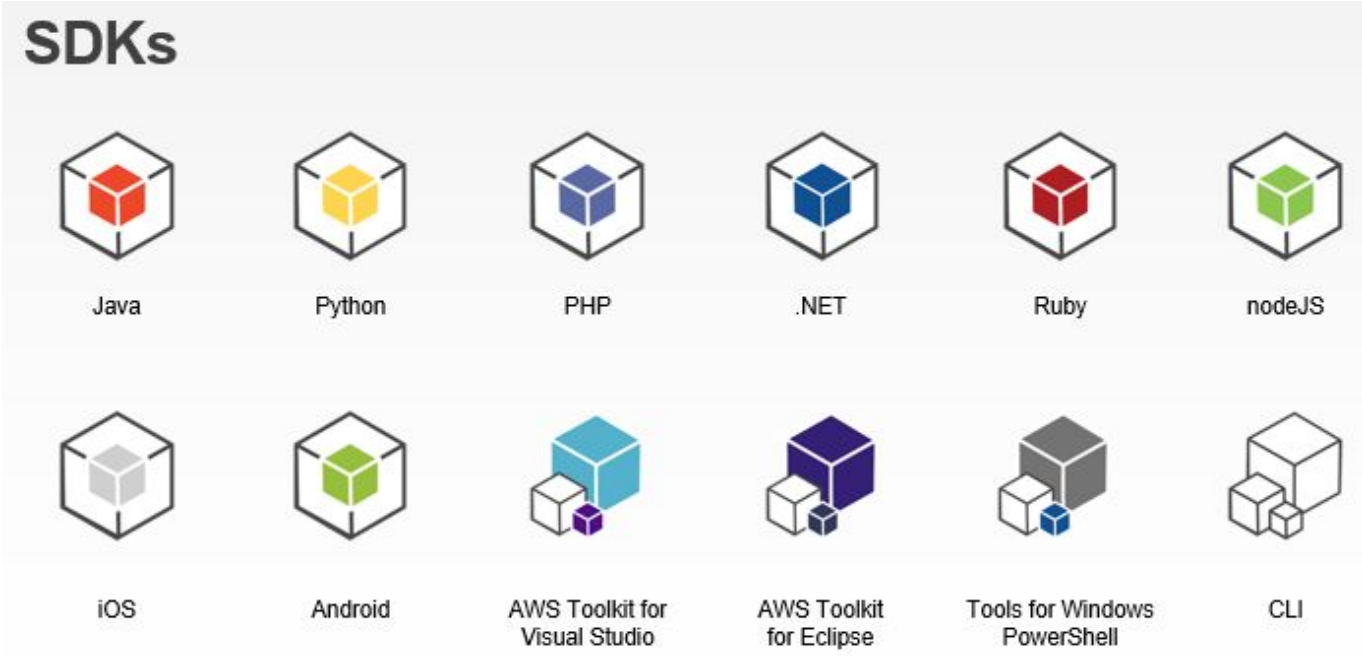



IAM users need to provide **account ID (or alias)** and **username**.

# Programmatic Access

| CLI | SDK | API |
| --- | --- | --- |
| Command Line Interface | Software Development Kit | Application Programming Interface |

```
guile@LAPTOP-O4O3C63R: ~
guile@LAPTOP-O4O3C63R:~$ aws s3 ls
2020-08-25 22:11:59 cf-templates-5mfgdye7649f-eu-central-1
2022-02-18 17:06:58 cf-templates-5mfgdye7649f-eu-west-1
2021-04-03 01:13:30 cf-templates-5mfgdye7649f-sa-east-1
2020-06-20 16:44:01 cf-templates-5mfgdye7649f-us-east-1
2020-06-20 15:52:29 cf-templates-5mfgdye7649f-us-east-2
2020-10-14 12:24:44 cf-templates-5mfgdye7649f-us-west-2
2022-03-11 12:36:54 clarusway-cf-demo
2021-11-30 04:15:33 clarusway.destination.lambda
2020-07-22 02:41:26 clarusway.lambda.images
2020-07-22 02:41:47 clarusway.lambda.images-resized
2021-11-22 12:19:24 clarusway.us
2021-06-26 15:11:23 codepipeline-eu-west-1-423764695465
2021-06-24 00:09:08 codepipeline-us-east-1-531089785775
2022-03-17 11:39:06 davids-test-petclinic-helm-charts
2021-11-24 19:39:55 elasticbeanstalk-eu-central-1-046402772087
```




Require **Access Key + Secret Key** to Authenticate

# What is IAM?

1. Authentication: Kimliklendirme, Sen Kimsin, Who are you
2. authorization : Yetkilendirme: Which permisions do you have?

IAM = **I**dentity & **A**ccess **M**anagement

- IAM is a web service that helps you securely control access to AWS resources.

**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

# IAM Terms

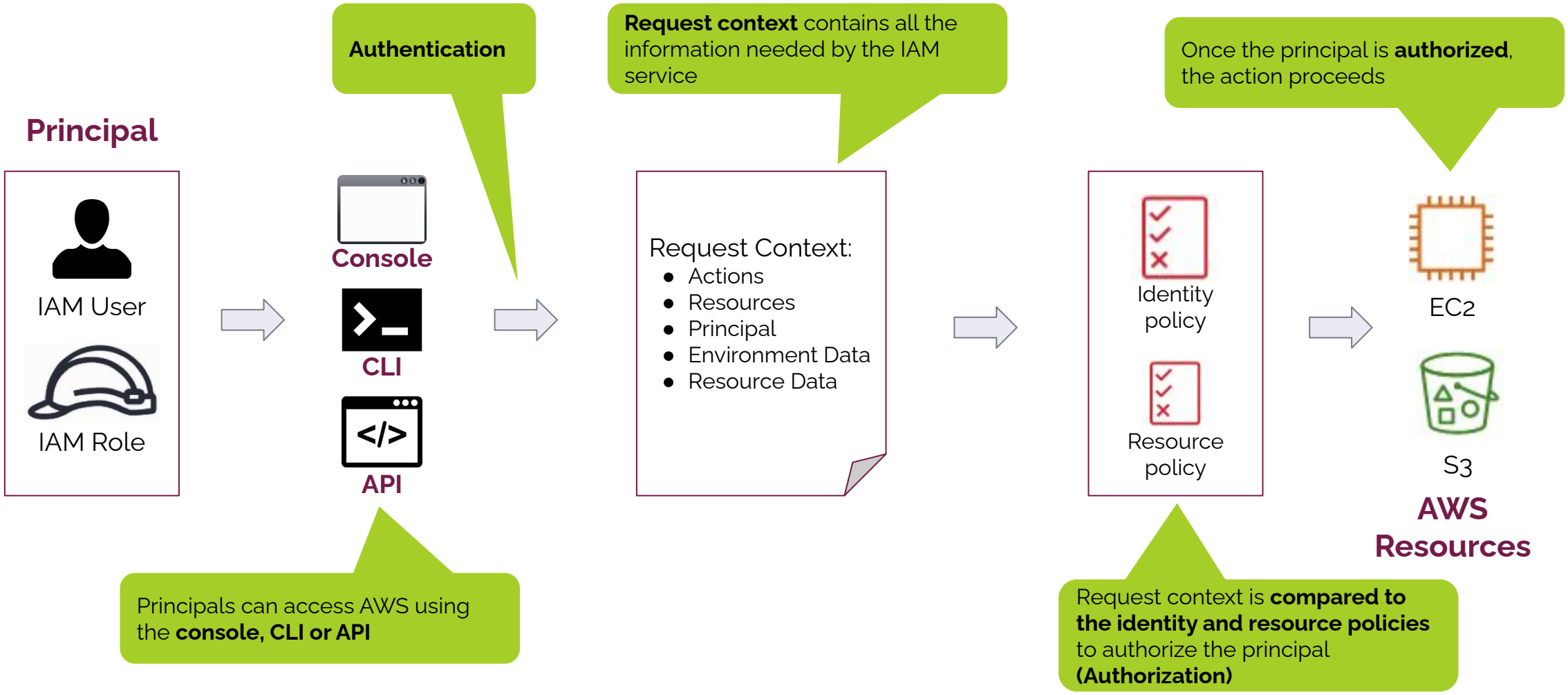
- IAM Resources: The user, group, role, policy, and identity provider objects that are stored in IAM.

- Principals: A person or application that uses the AWS account root user, an IAM user, or an IAM role to sign in and make requests to AWS. Principals include federated users and assumed roles.

- IAM Identities: The IAM resource objects that are used to identify and group. You can attach a policy to an IAM identity. These include users, groups, and roles.

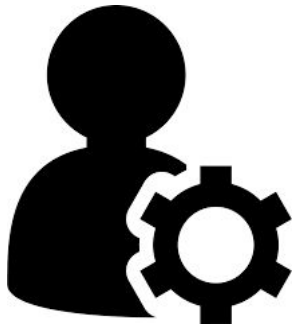
- IAM Entities: The IAM resource objects that AWS uses for authentication. These include IAM users and roles.

# How IAM Works

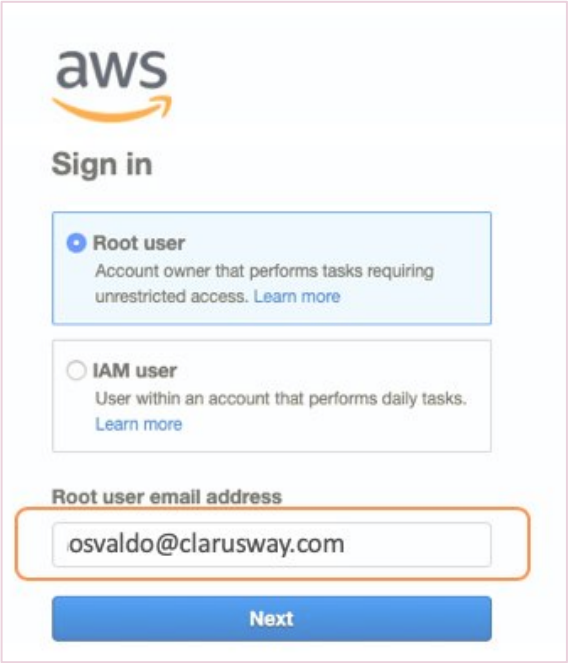


# 2 IAM Users

# Root User



- 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 user Root User for **day-to-day work**
  - Keep **password** in a secure location

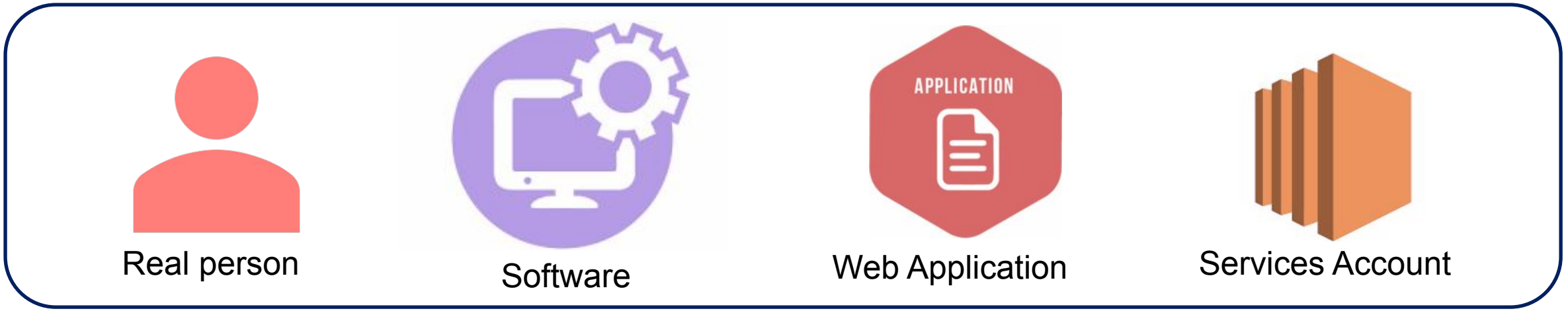



Log in to the console with **email address** rather than user name.

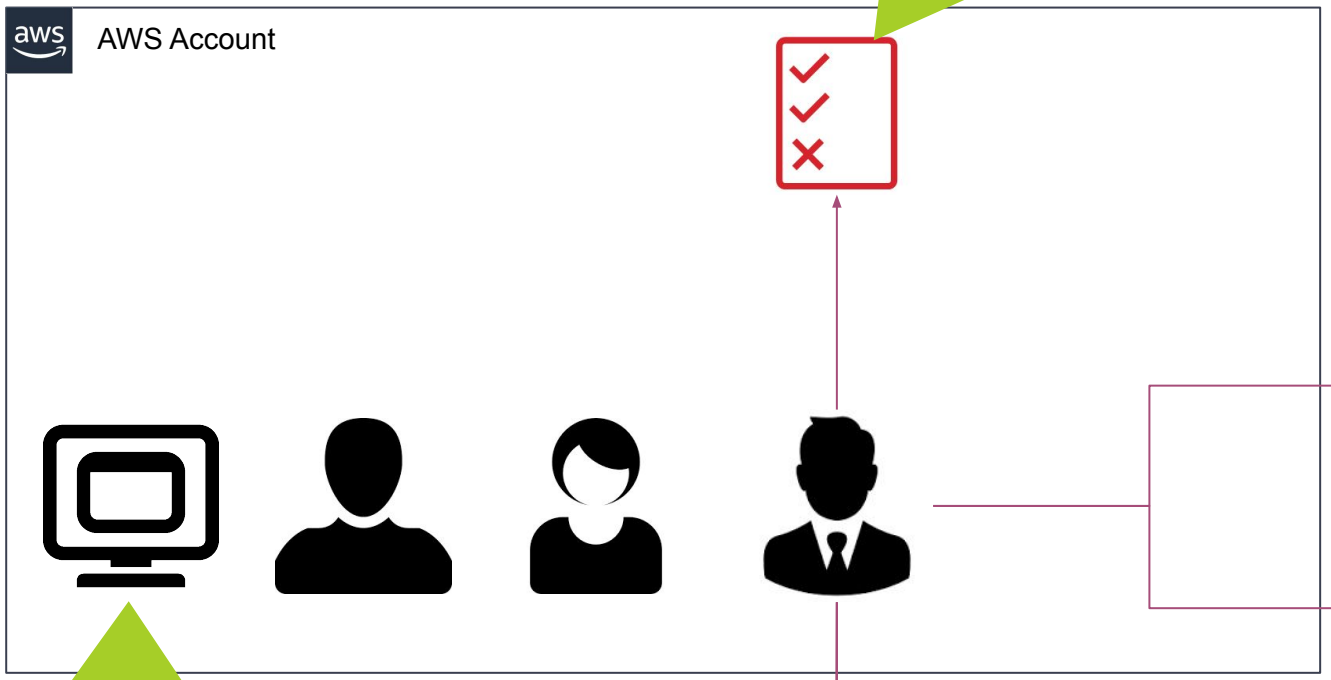
# IAM Users

## 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



# IAM Users




Users have **no permissions by default**, until they are assigned

**Username:**
John

**Amazon Resource Name (ARN)**
arn::aws:iam::123456789012:user/john

A user **doesn't have to be a person**.

Authenticate via **username + password** for **console**

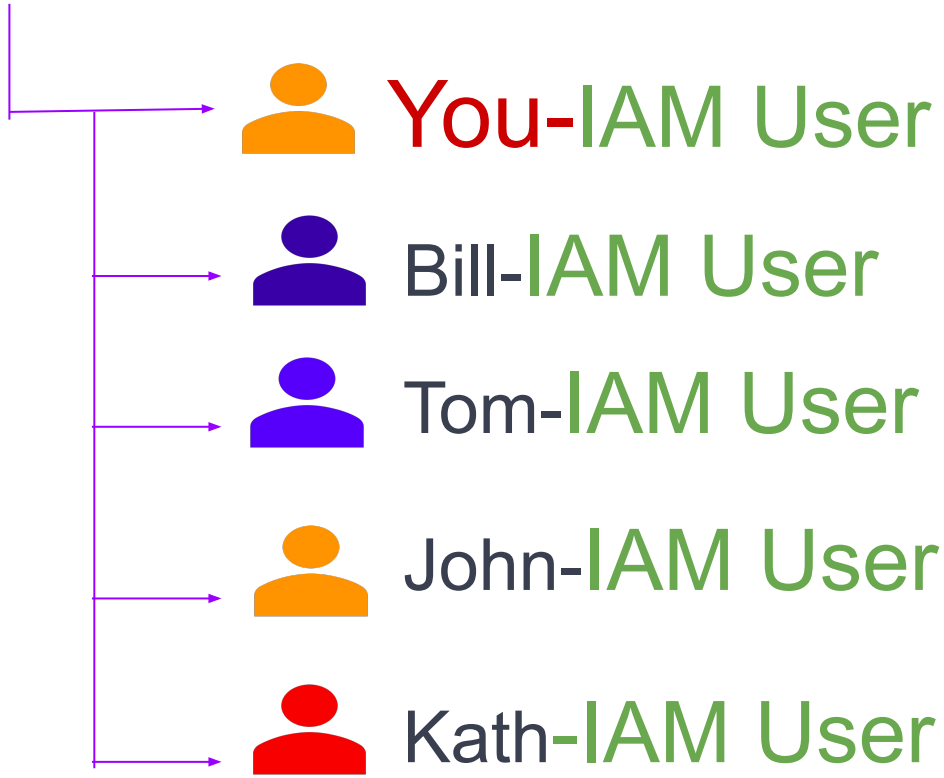
Authenticate via **access keys** for **CLI** and **API**

IAM users have **long term (permanent) credentials** - whether username/password or access keys.

# IAM Users

## What is Root User and IAM User.

AWS Account Owner - Root User (You)

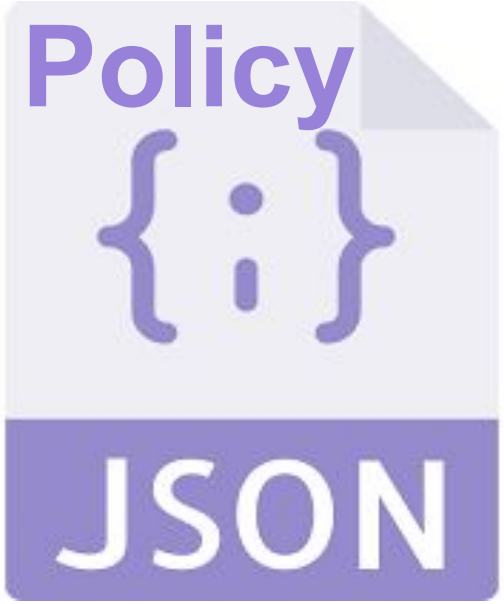



- Root User is a special user
- Username is email used to create account
- Generally, cannot limit permissions of Root User
- Cannot delete Root User
- Best practices:
  - Enable MFA for Root User
  - Don't user Root User for day-to-day work
  - Keep password in a secure location

# 3 IAM Polices

# IAM Polices

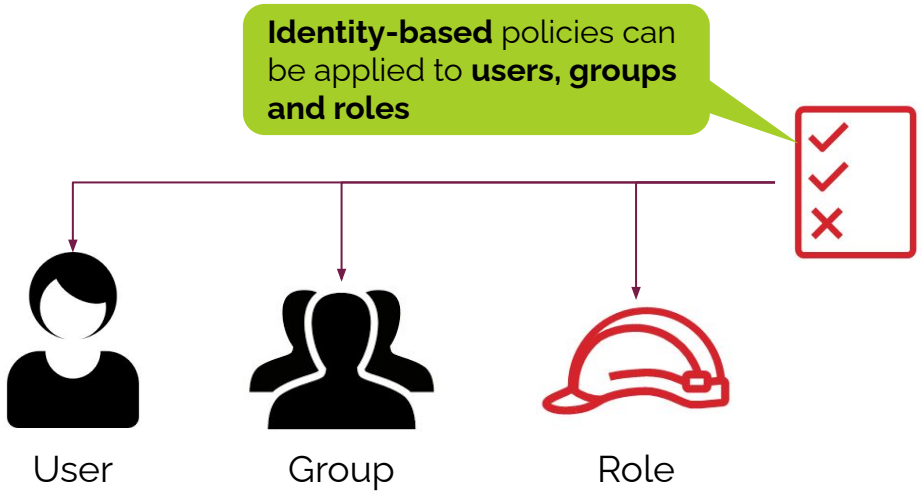
## What is a Policy?




- 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.

# IAM Polices

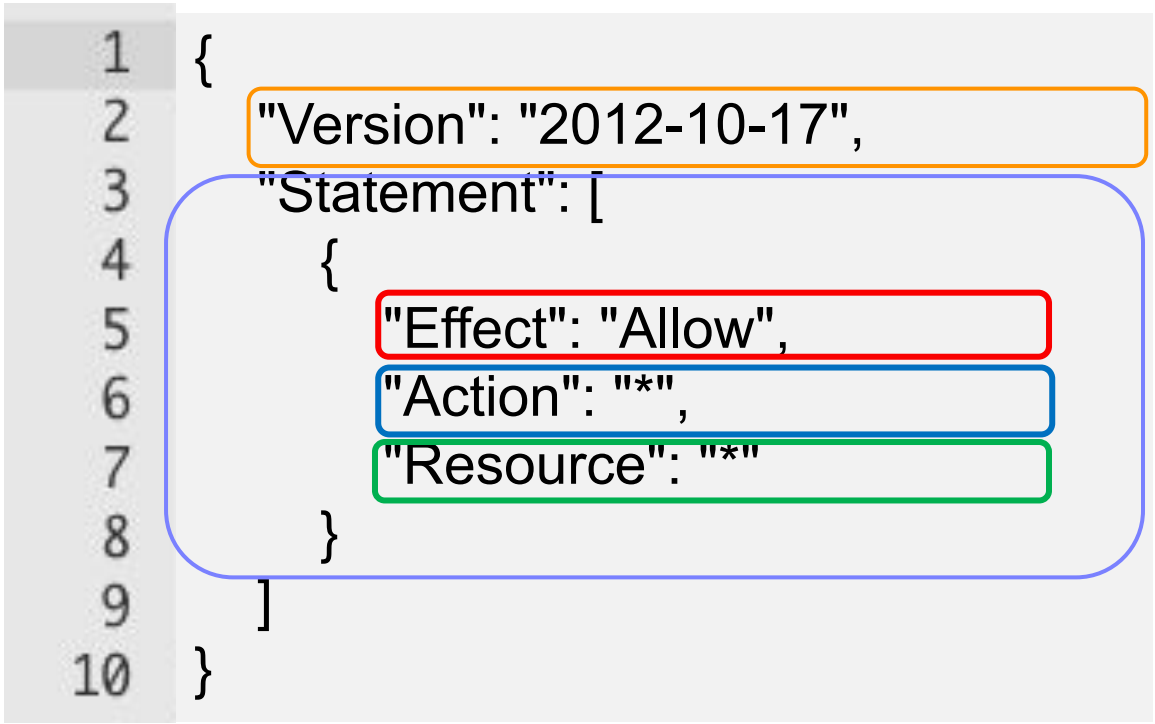
## What is a Policy?




```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "*",
            "Resource": "*"
        }
    ]
}
```

# IAM Polices

## Policy Structure




**Version:** Specifies the version of the policy document.

**Statement:** The basic part of a policy where you define permissions

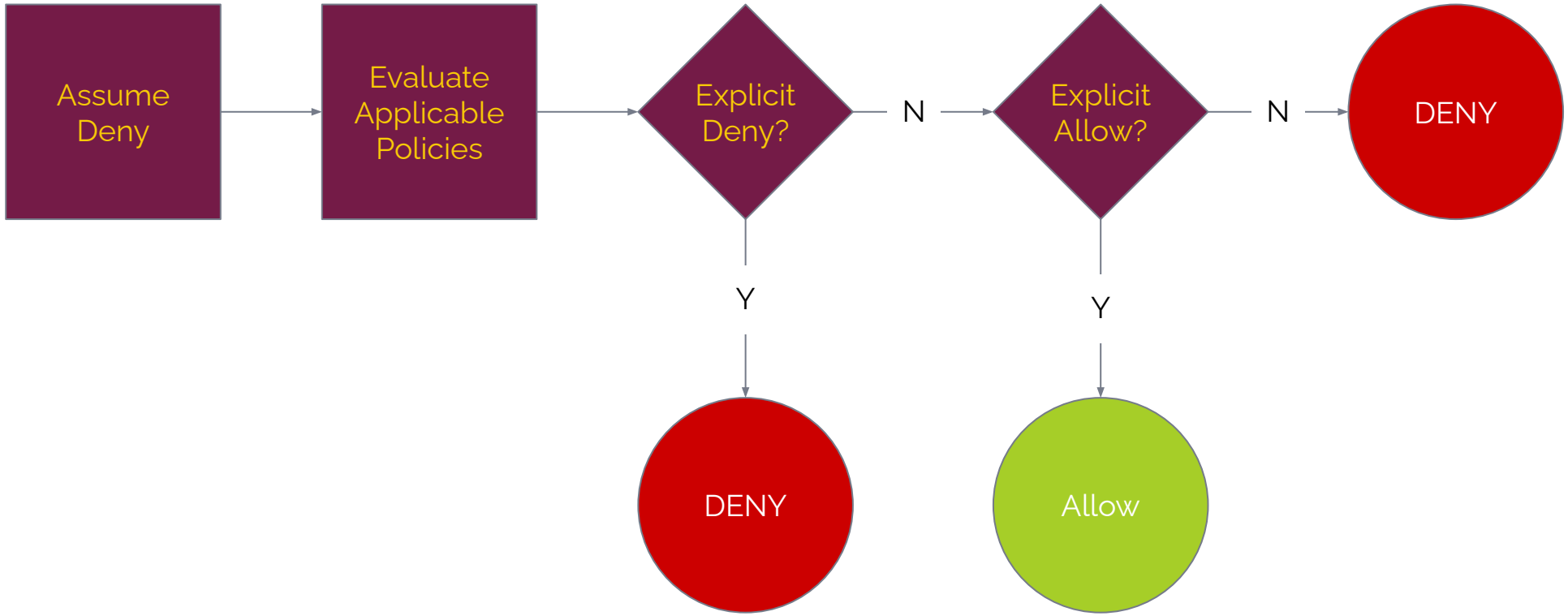
**Effect:** It determines what the statement actually does. Can contain only the **Allow** or **Deny** values.

**Actions:** Determines which actions the identity can perform.

**Resource:** Explains **in which AWS resources** the statement will perform the operations.

# IAM Polices

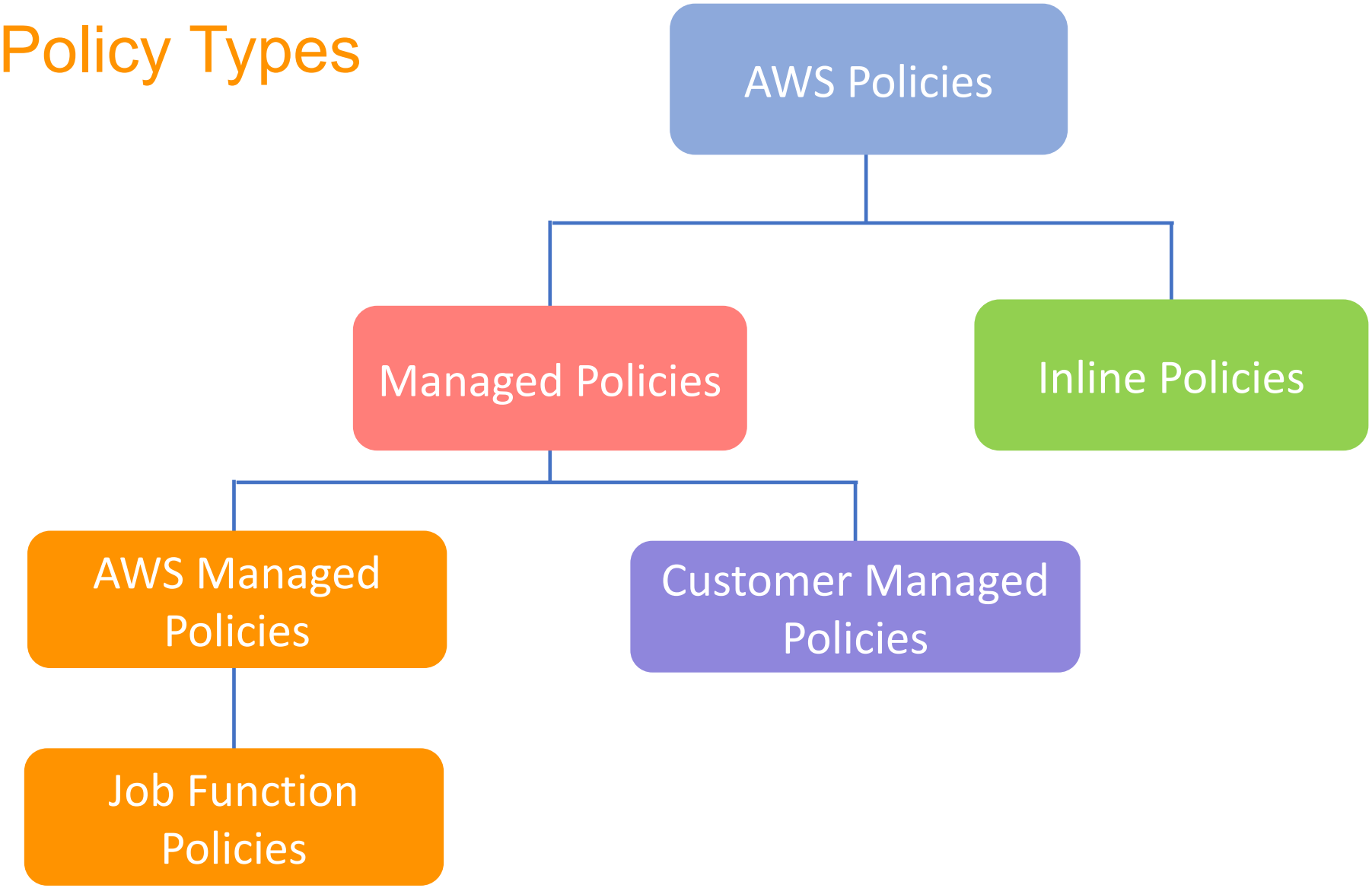
## Policy Evaluation




- Deny by **default**
- Deny takes **precedence** over allow

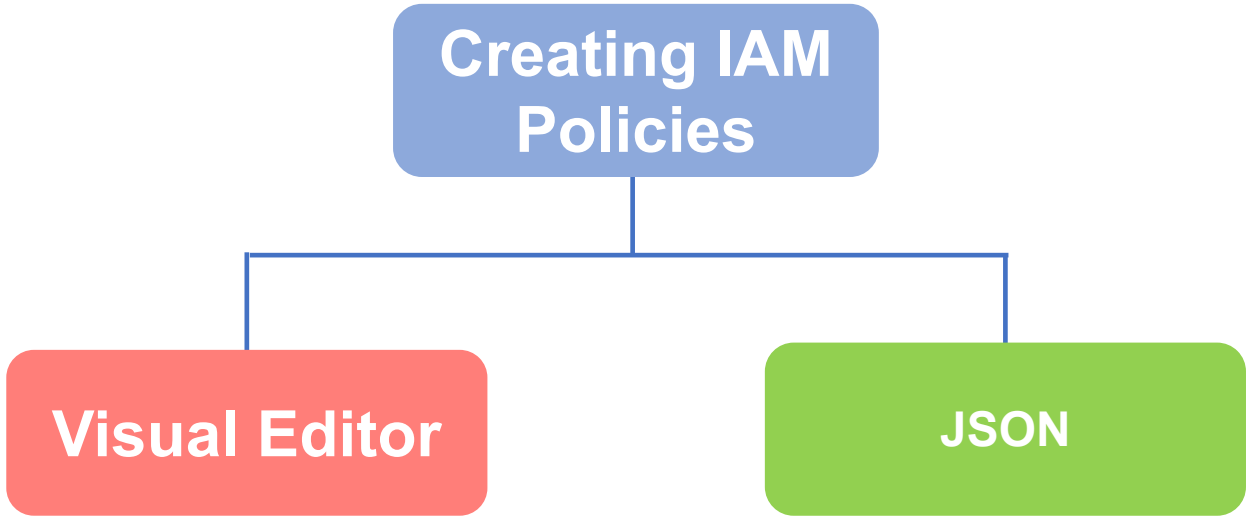
# IAM Polices

## IAM Policy Types



# IAM Polices

## Creating IAM Policies



# IAM Polices

## Helpful Tools

### Policy Generator

- Tool that enables you to **create JSON policy** documents using a **GUI**
- https://awspolicygen.s3.amazonaws.com/policygen.html

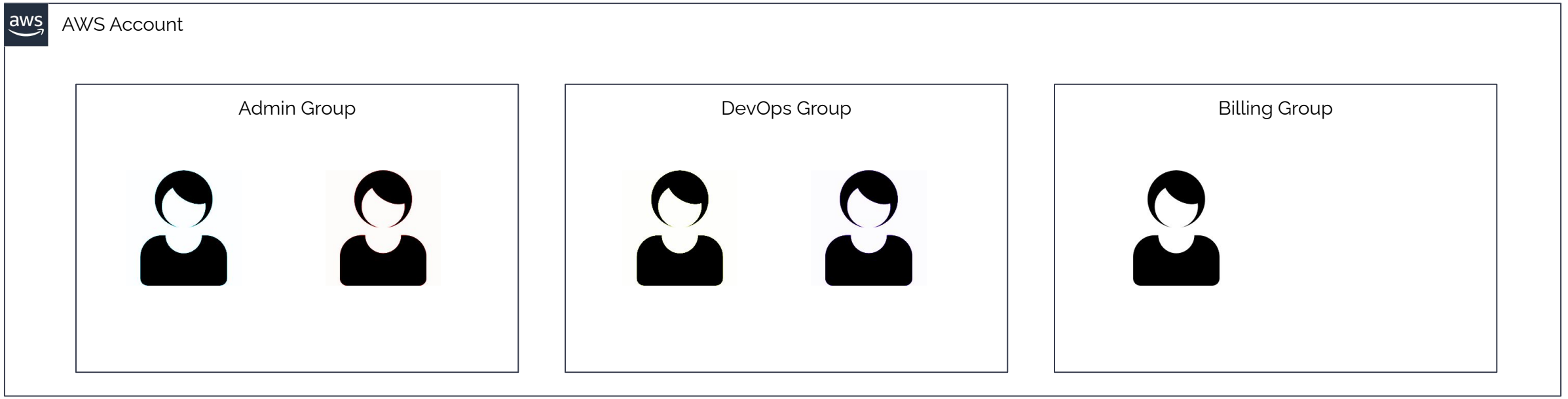
### Policy Simulator

- Tool that enables you **test** IAM-based policies
- Excellent for troubleshooting
- Slightly complex to use
- https://policysim.aws.amazon.com/home/index.jsp?#

# 4 IAM User Groups

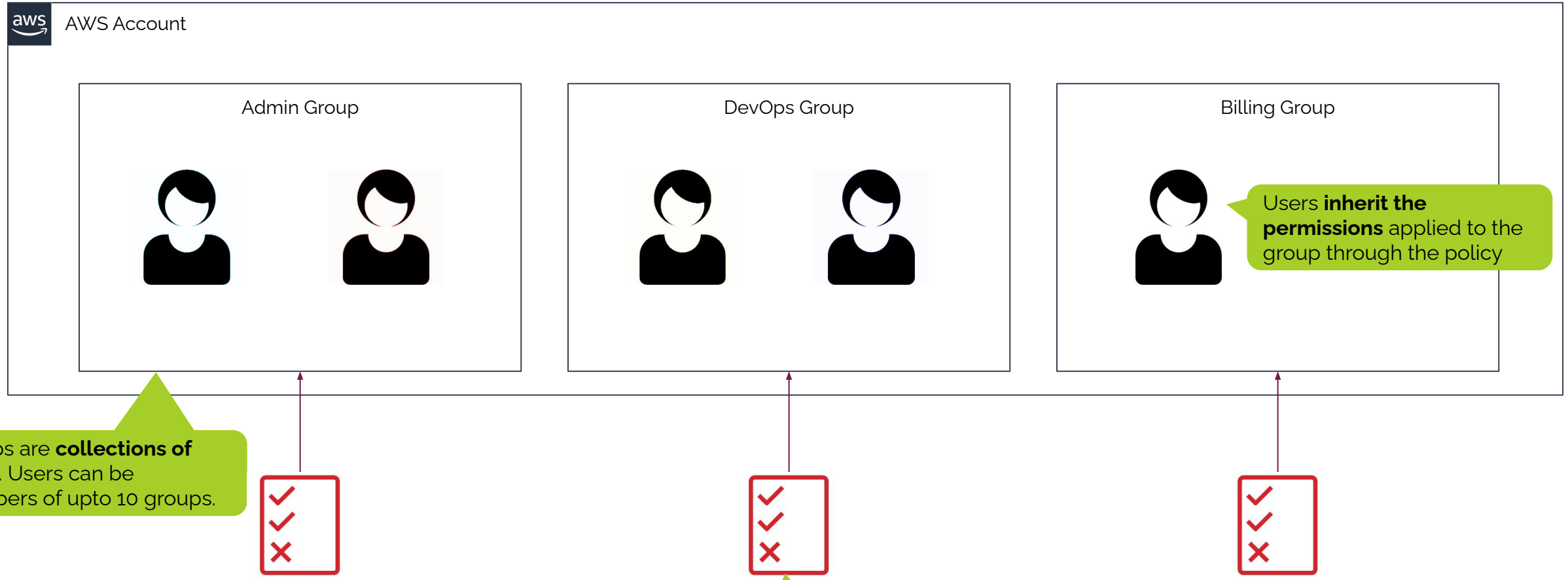
# IAM User Groups

## What is User Group in AWS?




- An IAM user group is a collection of IAM users. User groups let you specify permissions for multiple users, which can make it easier to manage the permissions for those users.

# IAM User Groups




Note that a **User Group** is **not** an **IAM principal**. It is used only as a convenience to manage users.

# IAM User Groups

## IAM User Group Features

- Managed IAM policies can be attached to user groups
- Inline IAM policies can be added to user groups
- The limit of IAM users in a user group is equal to 5000
- User can be a member of 10 different IAM user groups

# 5 IAM Roles

# IAM Roles

## What is a Role in AWS?


- An IAM role is an IAM identity that you can create in your account that has specific permissions. It's an authorization system where we determine how an identity can **access the AWS resources.**

- An IAM role, similar to an IAM user, is an IAM identity that **has specific permissions** that you can create in your account.

# IAM Roles

## What does IAM ROLE do ?

**www.e-commerce...**


Policy

JSON

aws

EC2

RDS

**Record:**

**-Customer info**

# IAM Roles

## Who can assume an IAM Role?

**Another AWS account**
Belonging to you or 3rd party

**AWS service**
EC2, Lambda and others

**Web identity**
Cognito or any OpenID provider

**SAML 2.0 federation**
Your corporate directory

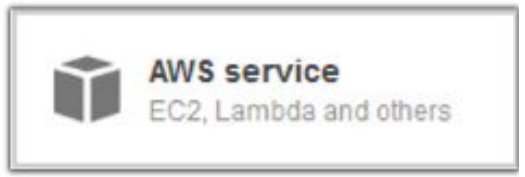


okta

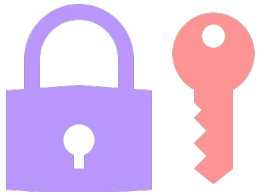
# IAM Roles

## Anatomy of a Role


**Trusted Entity**




**Permission Policy**






CLARUSWAY
WAY TO REINVENT YOURSELF

# Role Credentials

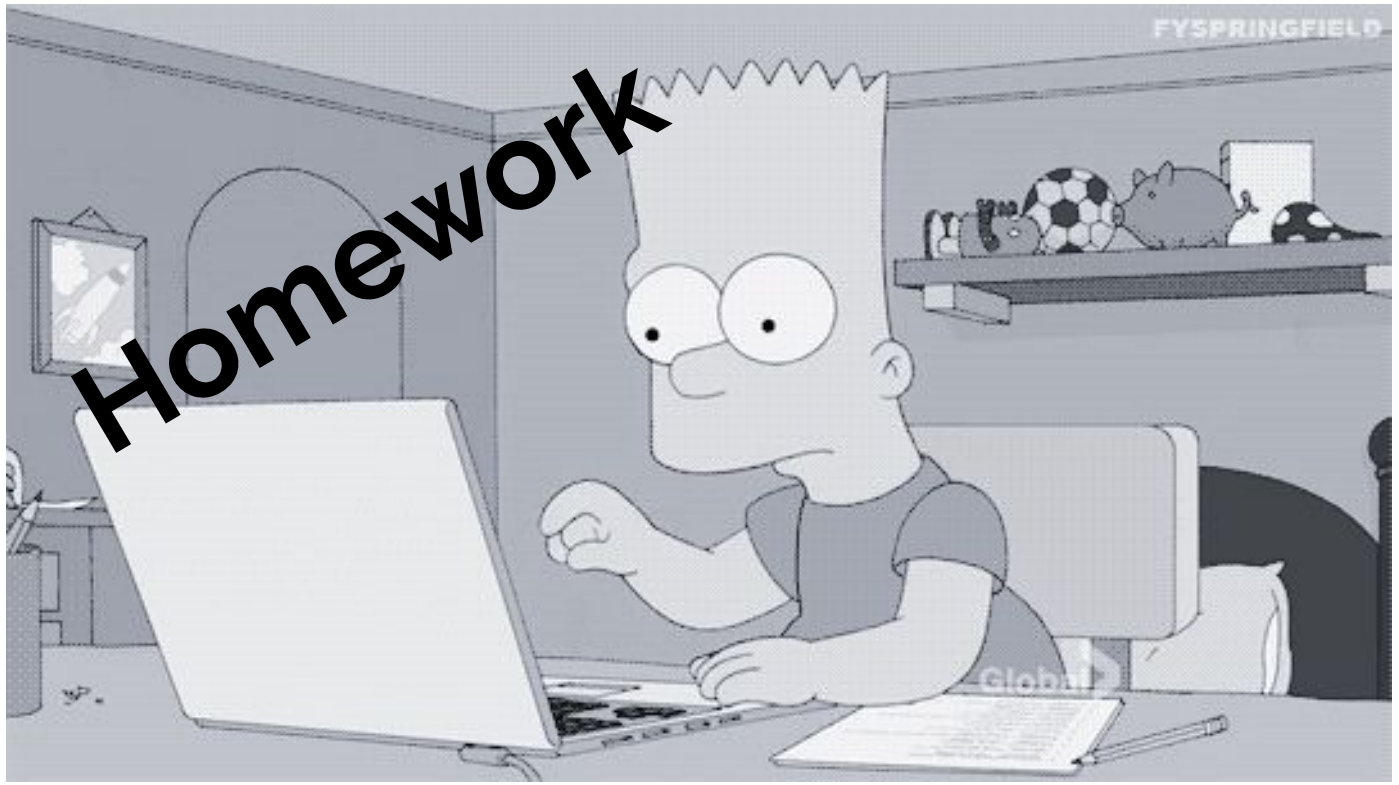
```
aws_access_key_id=ASIA5RBXKVCZWCMV4AFJ

aws_secret_access_key=23uUyY07I0PKG1URM6iQPV+A8wSsvLEbmHEA37wF

aws_session_token=IQoJb3JpZ2luX2VjEK/////////wEaCXVzLWVhc3QtMSJHMEUCIGrn7HEV38ejafaba56pEv1UxDIPjFdYLjgLSv0UvpmiA
iEA4b9Z2Noc0Ah3ru6bogoW+iBRtUrdg05zk7LkM4HQaNsqkgMIFxACGgw5Mjk5NzY0NjE0OTEiDAwgg62YKfWxIzb1TSrvArdvoRgYW4EvWtPAkM9R
IPk6EpWeHVMbDgVtyk7TGXCRTF6uZpyWSX33QS3Pwvb6d0pwiqomeOFDgG28U82eXrXGoKZnbTmnC+7X0QWgqAUI0Ku2kU/KLLwbLhjpv1Ai/oFpAvG
0FmZMtVZH+w6/uuyHgzFmPjwgrLTOj0AlnRfA1rjYJm6b2QD6ou5ZMK1JrV/jdW2z0Os7sPVkSA4lH6VPZ2D6vjAnRWDC+0uBV6QUfKlLLeJ1F51bTI
F3tI2Yu9VnXEV6usAblStCt3NnTpZRnGQTIyUcICLzAiGhJUdZpGQofdLrLEL/Matyg1wVA45RpT2MhgH+HPuoIGGT0uISBSt6YQV4/1wf9w2KSIT4U
dZgaQt8L+TDXIz1/ywn4f11dU0K9vwIINIwp+8s9le7hn1vQPm7HAetLi5mRE30vzXJ6Eoai9RbfgFW7HpxffZLImdOgealQ51w+0Zu7Rx4jGWhWLMd
WyrJQQw+ZXhgwY6pgESvD6LuI39m2hhJMC378lE8Q4OL+Jnl7CysdjNpBH9AjNwGuI9Ad3y3q1u8z1849KzCZCx9GbG/n9YYy3fGnBrrvNY3nrwiA4d
XKP4KfZU8OIQ3GlLJkK1d24lhhe9UBL3I1ySfMbvDbRoMOXESF6tCpMVLNMa4QaoVY7aThxDvAA6p51pftyPhCK3MJe4qBL4zTC3pXFJe+LPc6uwZ1F
sL/OTBH
```

Once an entity assumes a role, it receives **temporary credentials** in the form of an **access key**, **secret key** and **session token**.

Note that with an **IAM user**, there is **no session token**, since the credentials are **permanent**




# Video on Multi-Factor Authentication (MFA)

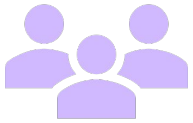
https://lms.clarusway.com/mod/lesson/view.php?id=7626&pageid=7570

# Let's get our hands dirty!

## AWS Account owner - Root User (you)

Administrator (you)-Newly Created IAM user

Database
(RDS FullAccess)



Bill

Tom

Developer
(S3 and EC2 FullAccess)

Mike

Tom*

# THANKS!

## Any questions?

You can find me at:

- @Guile - Instructor
- guile@clarusway.com