



# AWS IAM





# Table of Contents

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



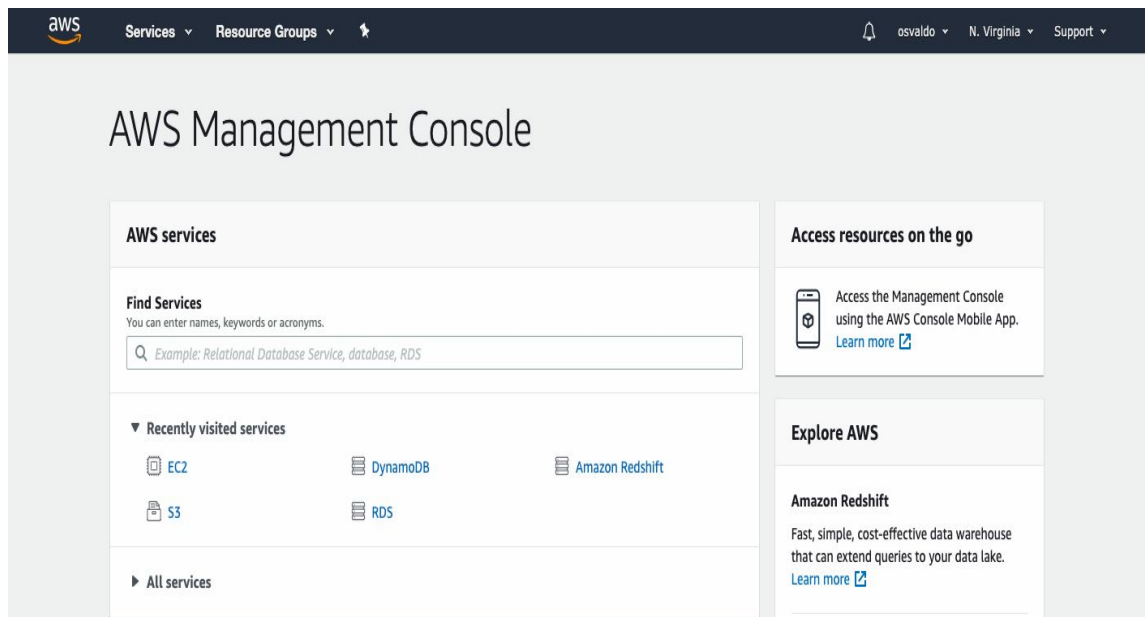
1

# Introduction to IAM

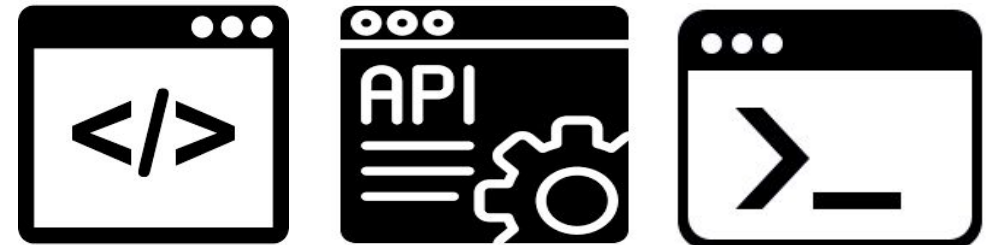
# Accessing AWS



## AWS Management Console



## Programmatic Access



CLI, SDK, API

# Management Console



Console login for **Root User**

The screenshot shows the AWS 'Sign in' page. Under the 'Sign in' heading, there are two options: 'Root user' (selected with a blue dot) and 'IAM user' (unselected with a grey dot). The 'Root user' option is described as 'Account owner that performs tasks requiring unrestricted access.' Below these options, there is a field for 'Root user email address' containing the text 'osvaldo@clarusway.com'. At the bottom, there is a blue 'Next' button.

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



Console login for **IAM User**

The screenshot shows the AWS 'Sign in as IAM user' page. It features three input fields: 'Account ID (12 digits) or account alias', 'IAM user name', and 'Password'. Below these fields is a blue 'Sign In' button. At the bottom, there are links for 'Sign in using root user email' and 'Forgot password?'.

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

Require  
**Username & Password**  
to Authenticate

# Programmatic Access



## CLI

Command Line 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
```

## SDK

Software Development Kit

## API

Application Programming Interface

### SDKs



Java



Python



PHP



.NET



Ruby



nodeJS



iOS



Android



AWS Toolkit for  
Visual Studio



AWS Toolkit  
for Eclipse



Tools for Windows  
PowerShell



CLI

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



# What is IAM?



IAM = Intity & Access Management

- **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

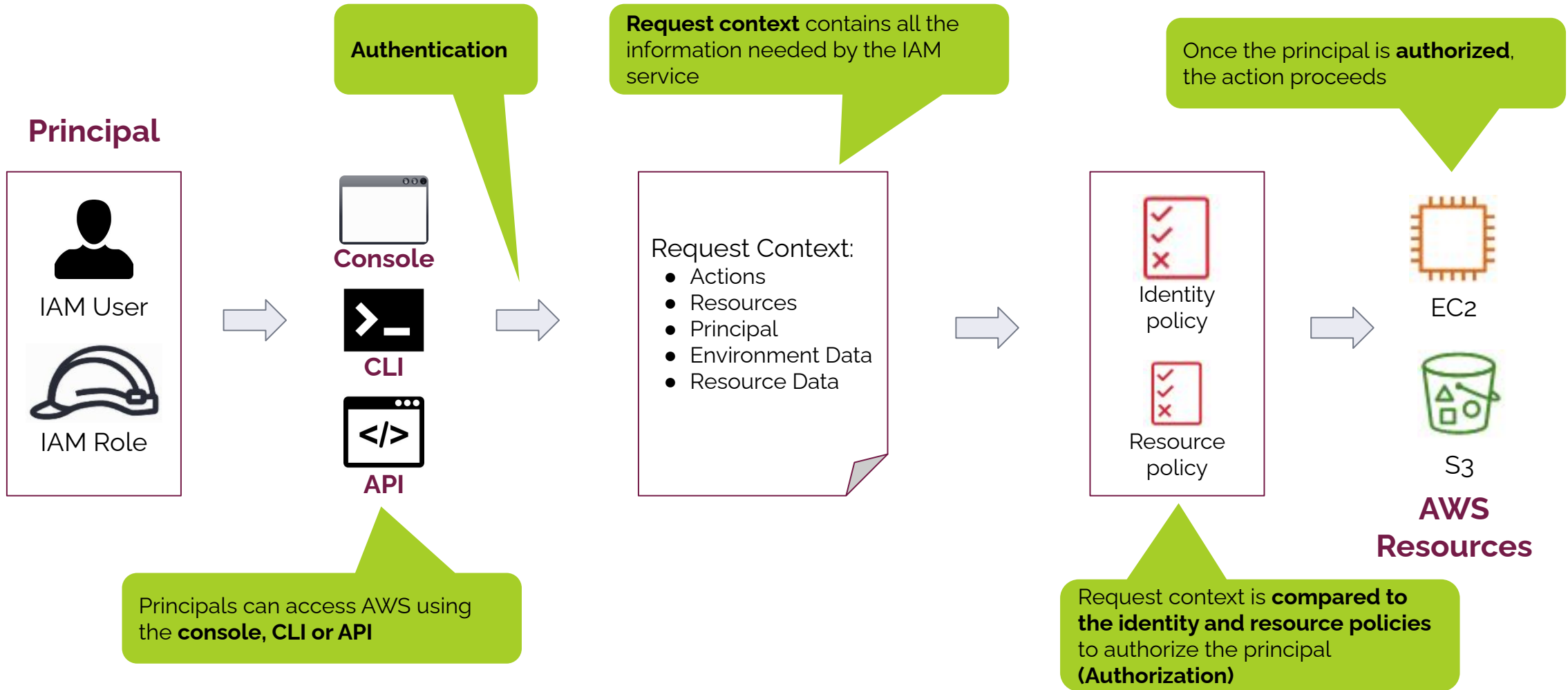


# IAM Terms

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

The screenshot shows the AWS 'Sign in' page. The 'Root user' option is selected with a radio button. Below it, the 'IAM user' option is available. The 'Root user email address' field is highlighted with an orange border and contains the text 'osvaldo@clarusway.com'. A blue 'Next' button is at the bottom of the form.

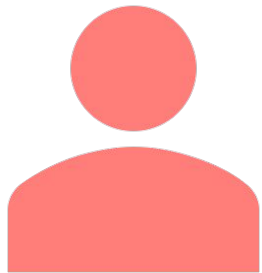
Log in to the console with **email address** rather than user name.

# IAM Users



## What is IAM User?

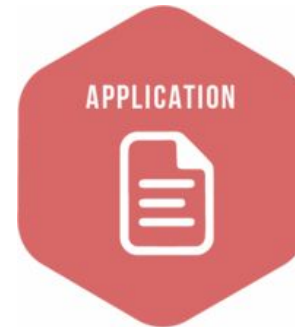
IAM User is an entity that you create in AWS to represent the person or application that uses it to interact with AWS



Real person



Software

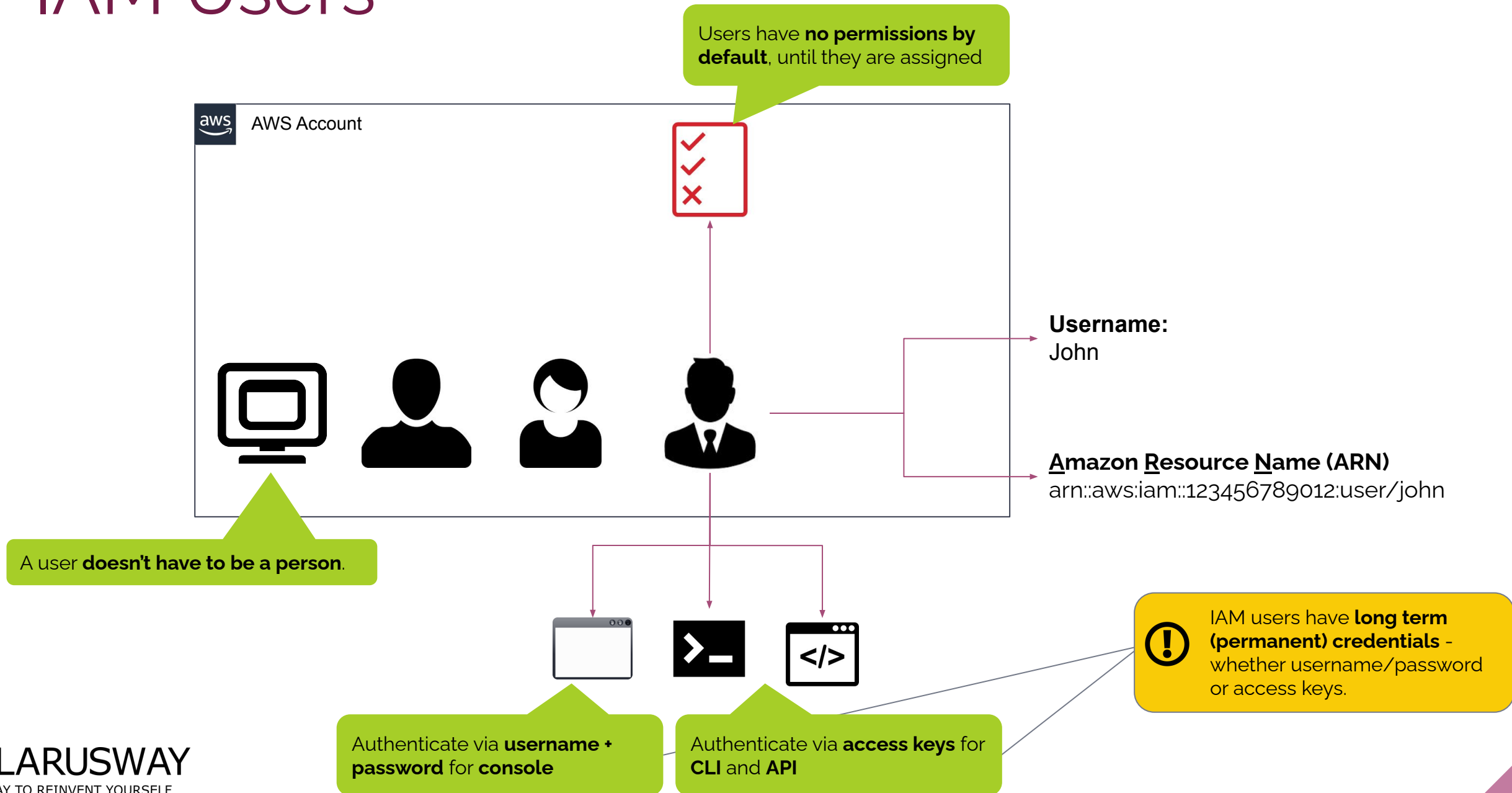


Web Application



Services Account

# IAM Users

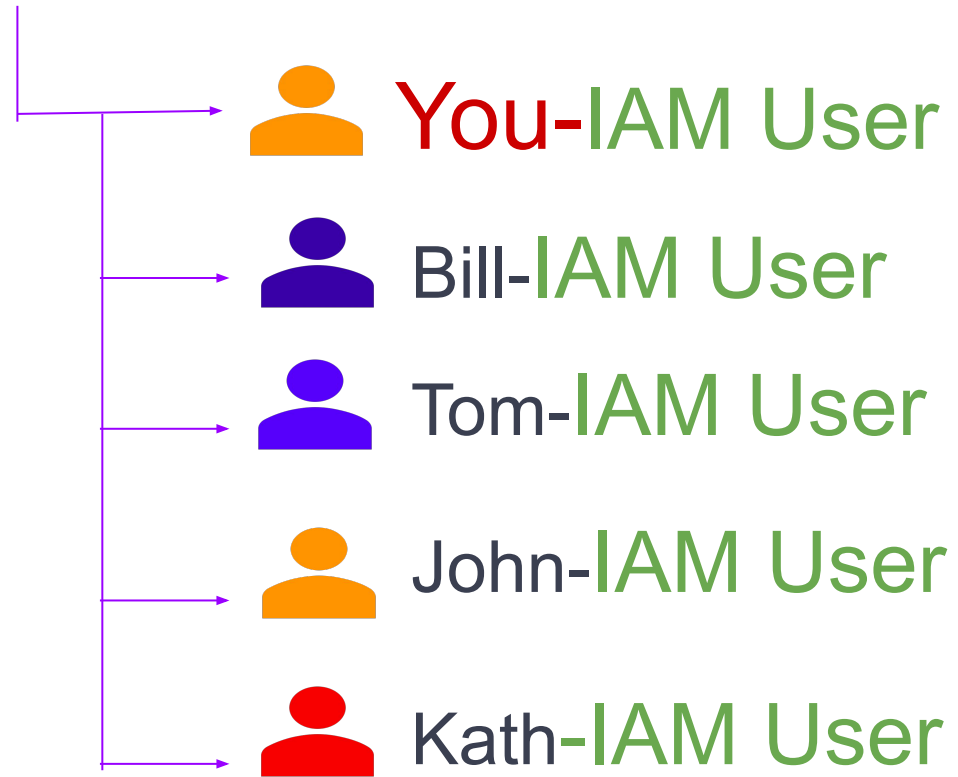


# IAM Users



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 Policies

## What is a Policy?

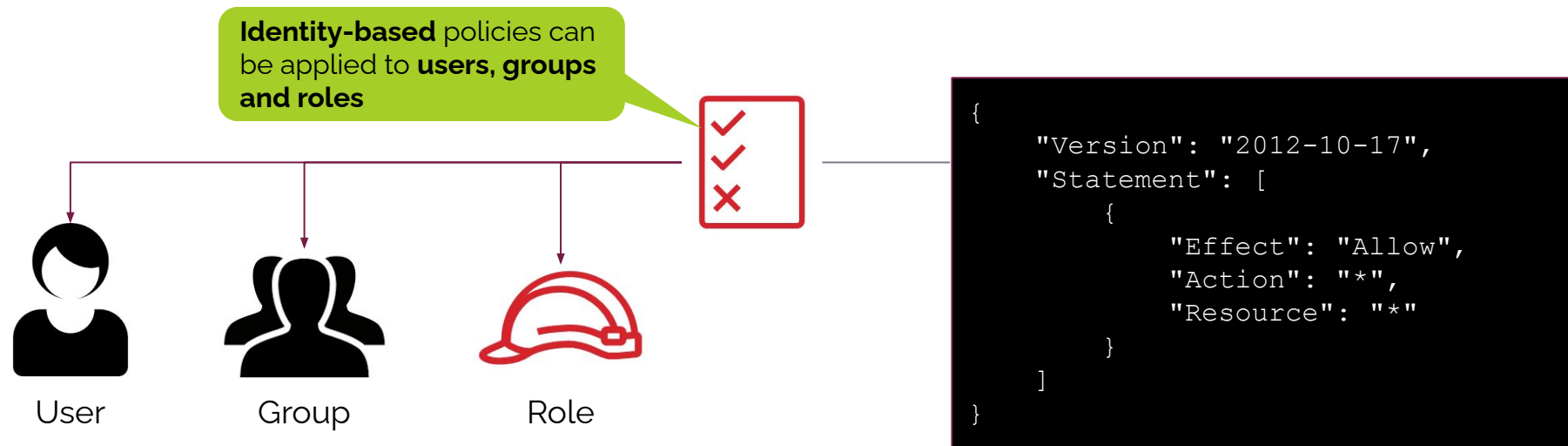


- A policy is an object used to define the **permissions** of an identity or resource in AWS
- Permissions in the policies determine whether the request is **allowed** or **denied**.
- Policies are stored in AWS as **JSON** documents.



# IAM Policies

## What is a Policy?



# IAM Policies

## Policy Structure



```
1  {
2    "Version": "2012-10-17",
3    "Statement": [
4      {
5        "Effect": "Allow",
6        "Action": "*",
7        "Resource": "*"
8      }
9    ]
10 }
```

**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.

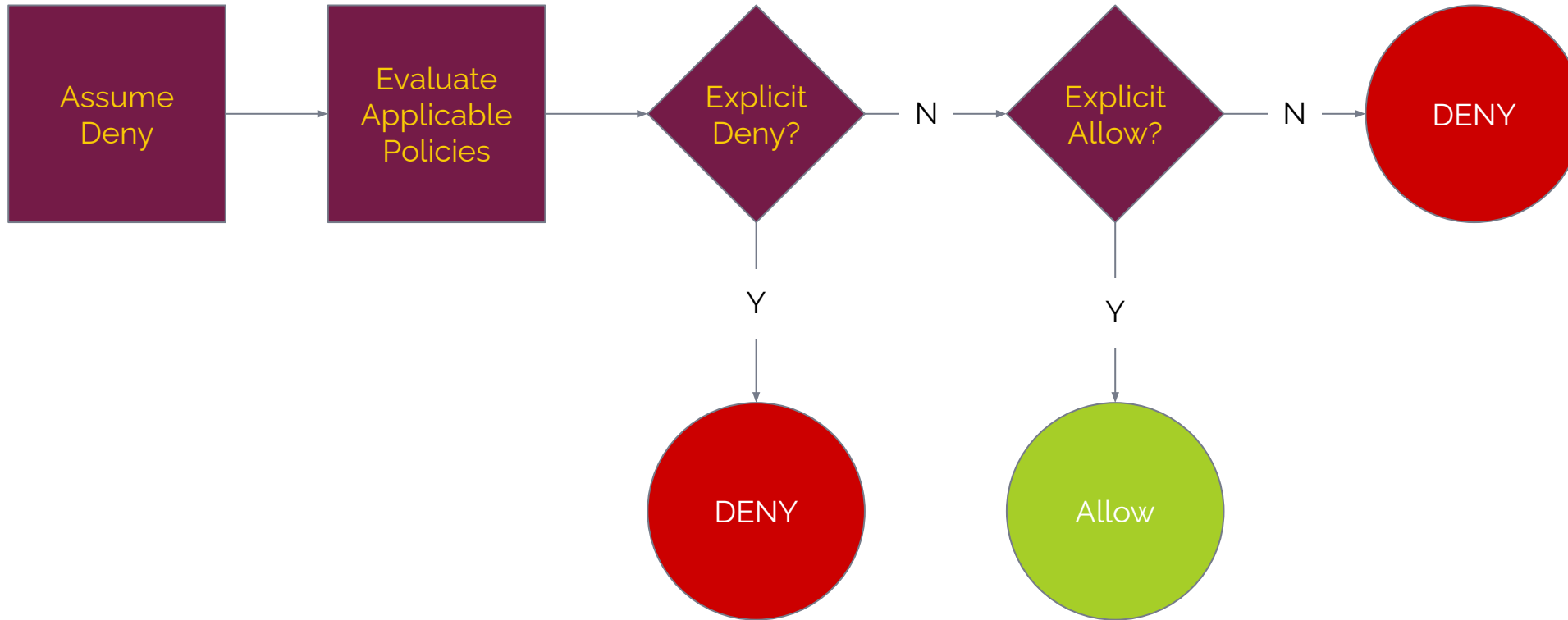
**Actions:** Determines which actions the identity can perform.

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



# IAM Policies

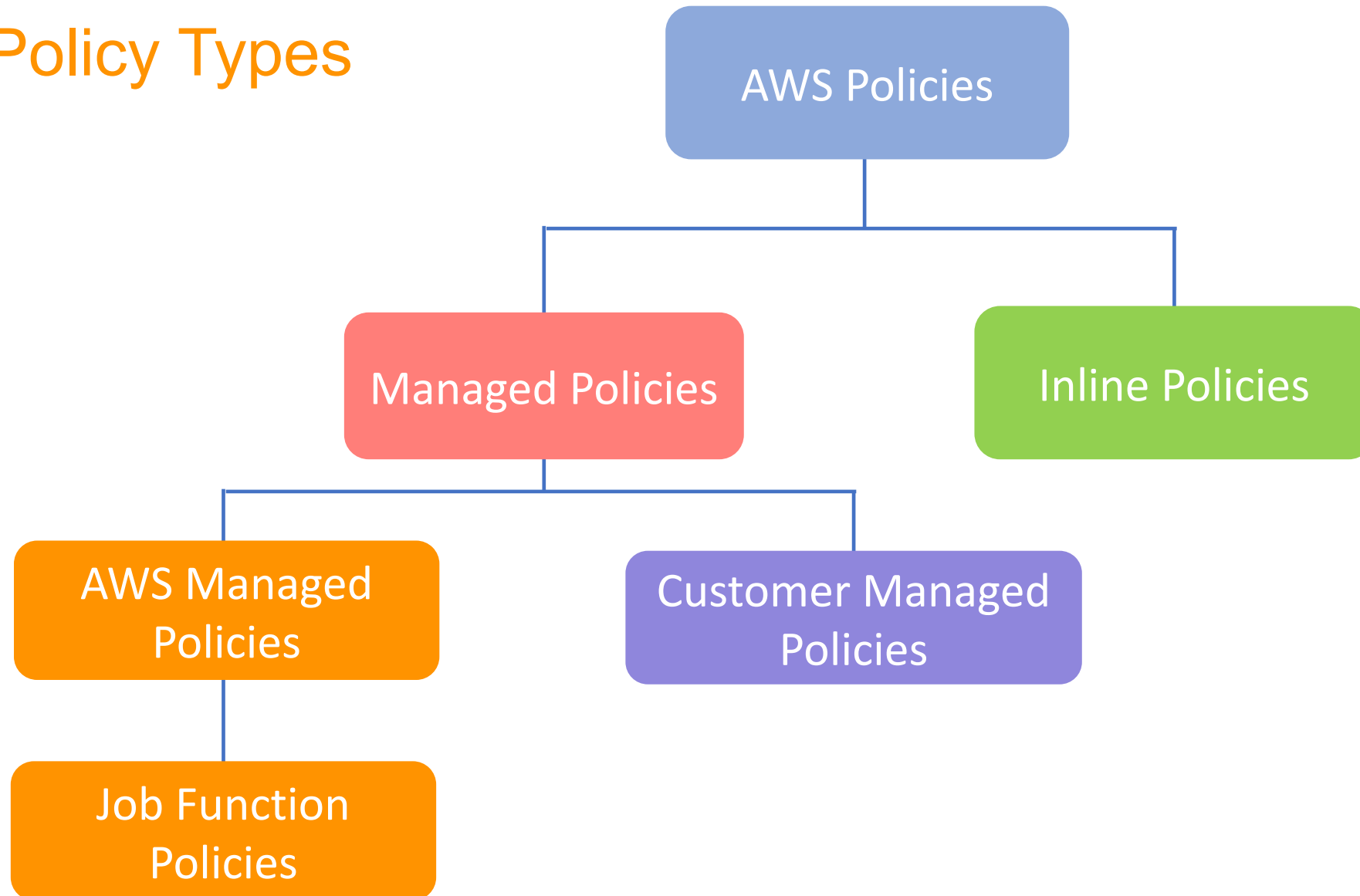
## Policy Evaluation



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

# IAM Policies

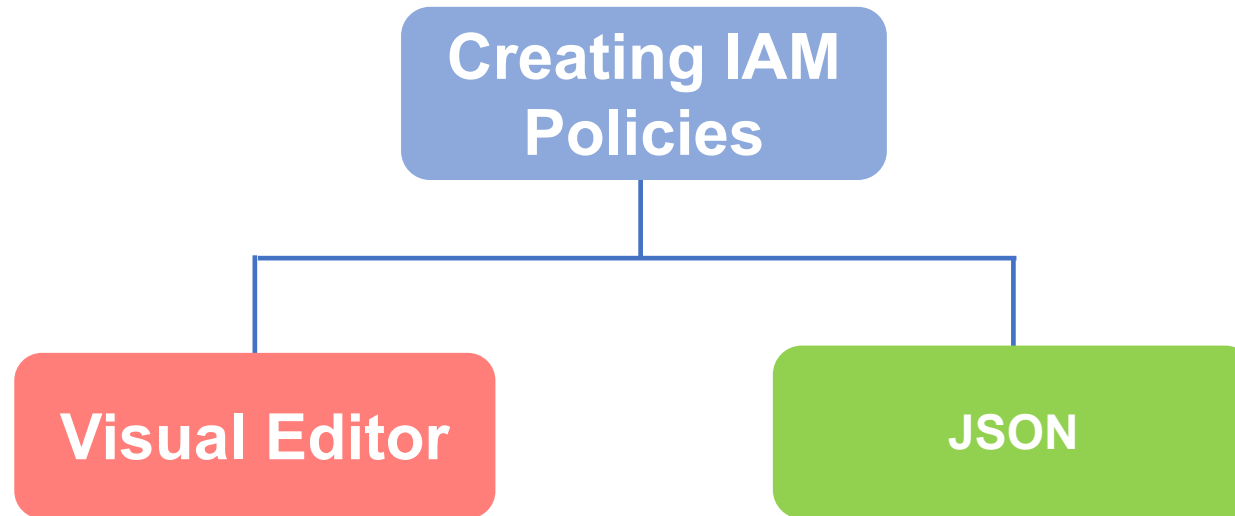
## IAM Policy Types





# IAM Policies

## Creating IAM Policies





# IAM Policies

## 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?#>

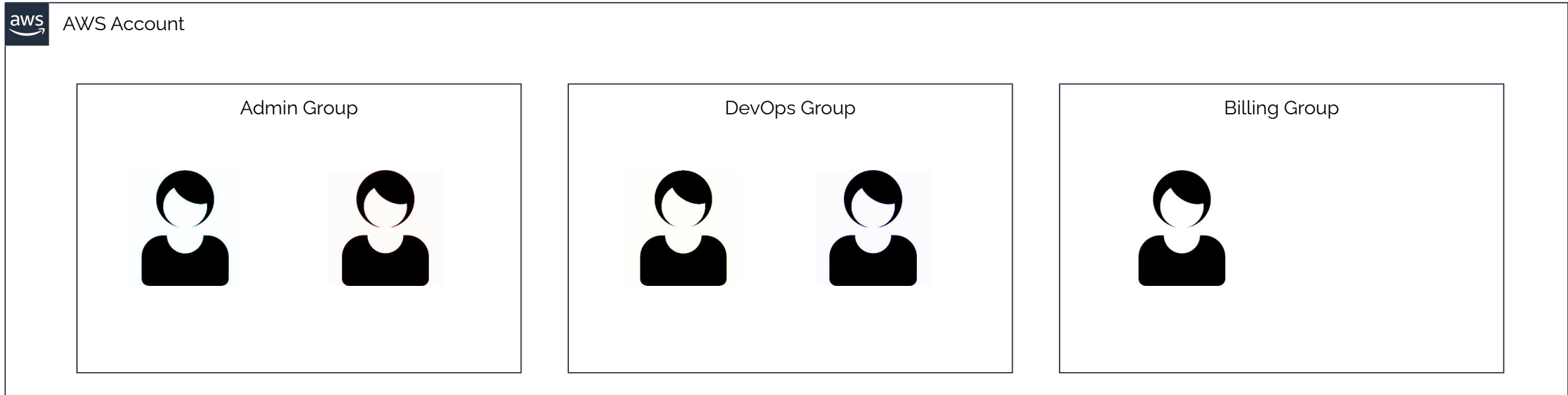


4

# IAM User Groups

# IAM User Groups

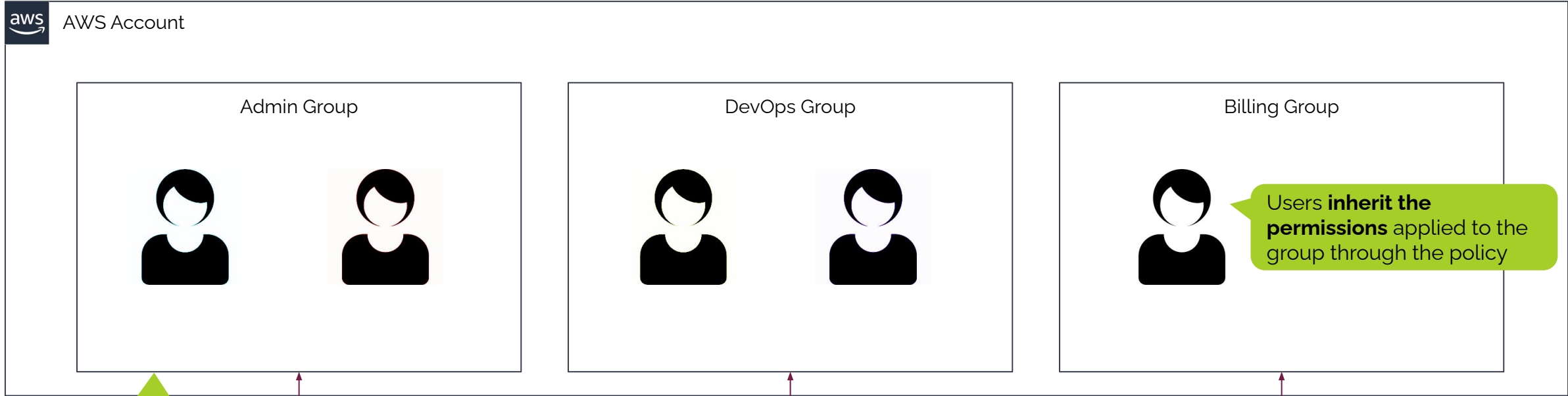
## What is User Group in AWS?



- 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



Groups are **collections of users**. Users can be members of upto 10 groups.



Purpose of using groups is to **apply permissions to users**.



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?

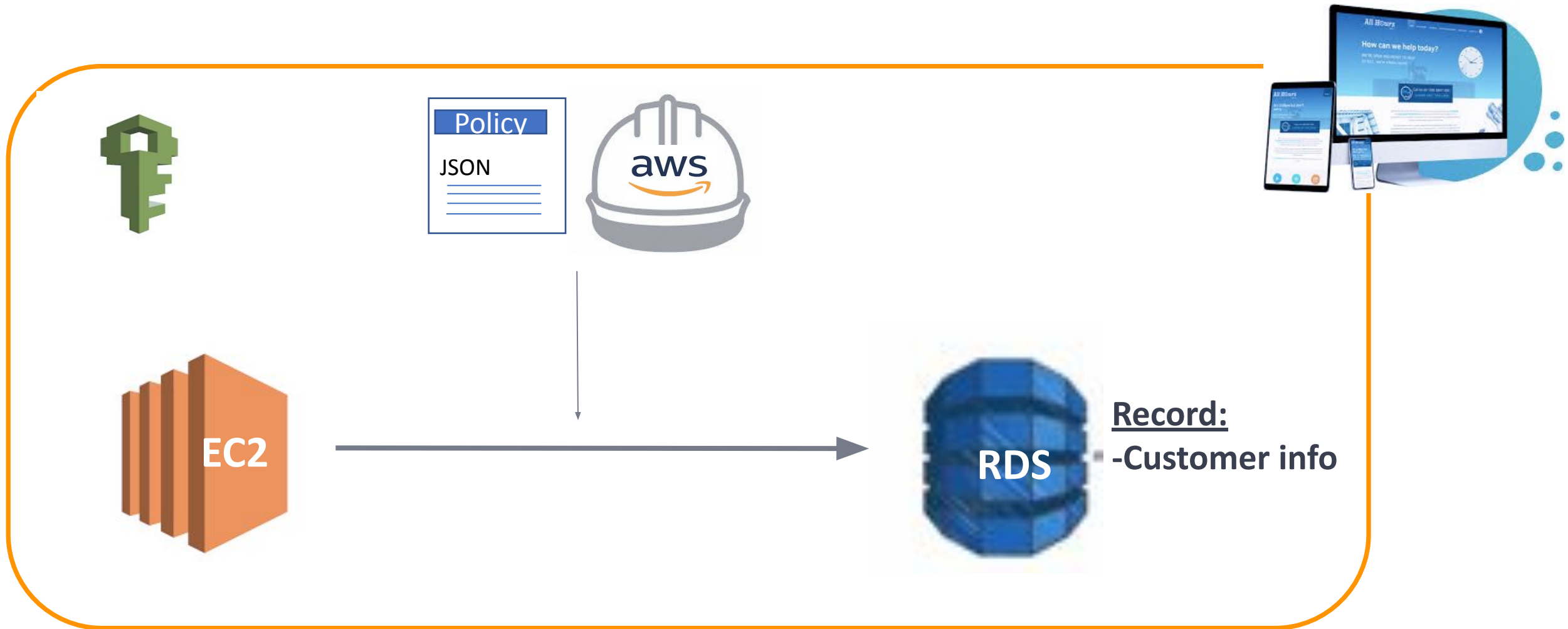


- An IAM role is an IAM identity that you can create in your account that has specific permissions. It's an authorization system where we determine how an identity can **access the AWS resources**.
- An IAM role, similar to an IAM user, is an IAM identity that **has specific permissions** that you can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

# IAM Roles

What does IAM ROLE do ?

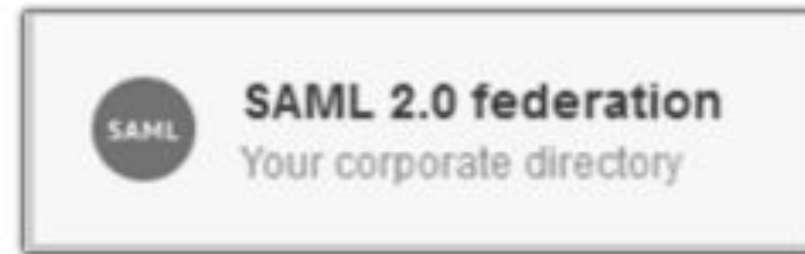
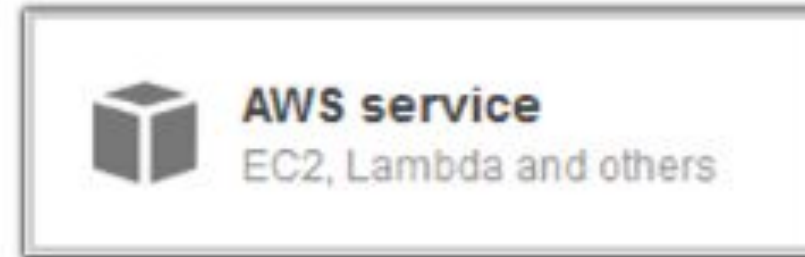
www.e-commerce...





# IAM Roles

Who can assume an IAM Role?



# IAM Roles

## Anatomy of a Role



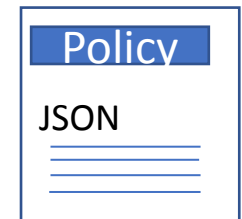
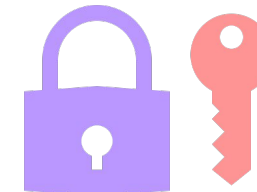
**Trusted Entity**



**AWS service**  
EC2, Lambda and others



**Permission Policy**



# Role Credentials



```
aws_access_key_id=ASIA5RBXKVCZWCMV4AFJ
```

```
aws_secret_access_key=23uUyY07I0PKG1URM6iQPV+A8wSsvLEbmHEA37wF
```

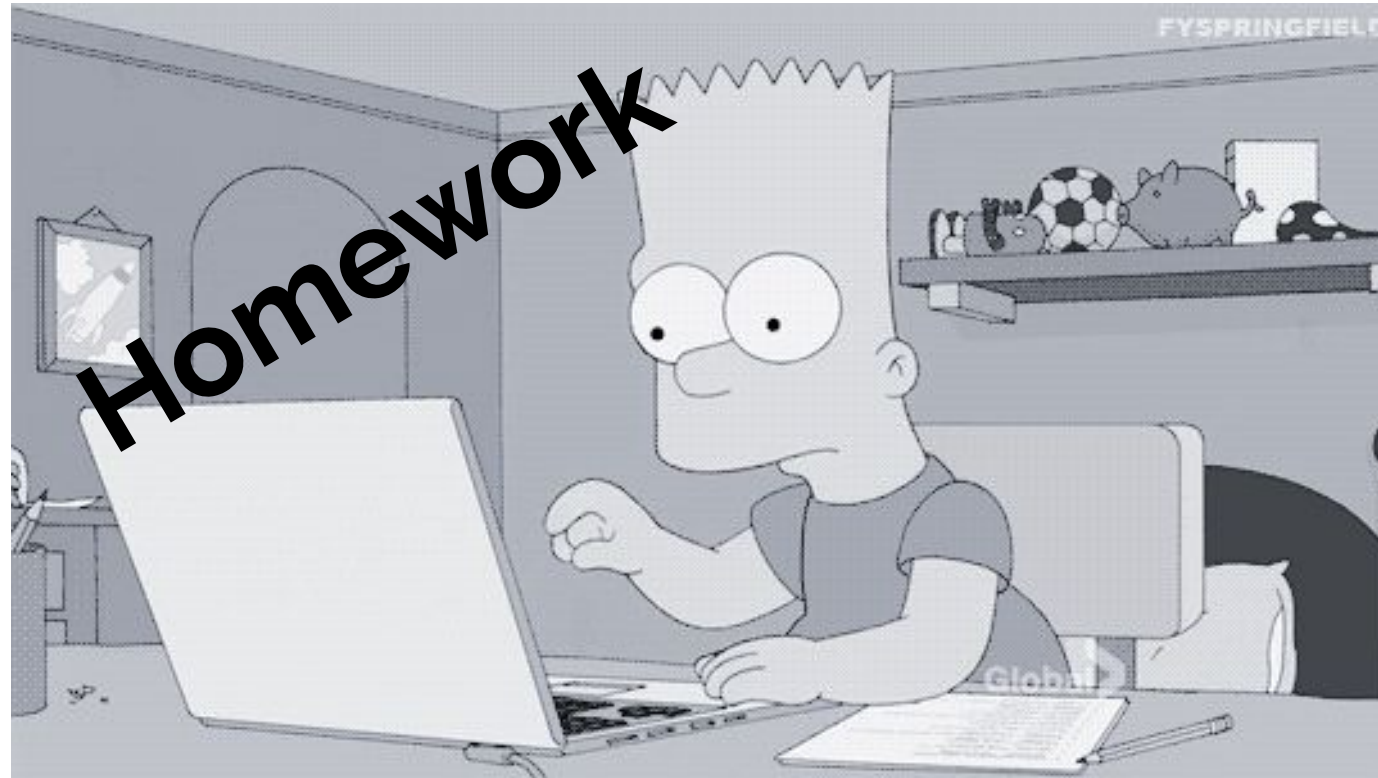
```
aws_session_token=IQoJb3JpZ2luX2VjEK//////////wEaCXVzLWVhc3QtMSJHMEUCIGrn7HEV38ejafaba56pEv1UxDIPjFdYLjgLSv0UvpmiA  
iEA4b9Z2Noc0Ah3ru6bogoW+iBRtUrdg05zk7LkM4HQaNsqqgMIFxACGgw5Mjk5NzY0NjE0OTEiDAwgg62YKfWxIzb1TSrvArdvoRgYW4EvWtPAkM9R  
IPk6EpWeHVMbDgVtyk7TGXCRTF6uZpyWSX33QS3Pwvb6d0pwiqomeOFDgG28U82eXrXGoKZnbTmnC+7X0QWggqAUI0Ku2kU/KLLwbLhjpv1Ai/oFpAvG  
0FmZMtVZH+w6/uuyHgZFmPjwgrLTOj0AlnRfA1rjYJm6b2QD6ou5ZMK1JrV/jdW2z0Os7sPVkSA4lH6VPZ2D6vjAnRWDC+0uBV6QUfK1LLeJ1F51bTI  
F3tI2Yu9VnXEV6usAblStCt3NnTpZRnGQTiyUcICLzAiGhJUdZpGQofdLrLEL/MatyglwVA45RpT2MhgH+HPuoIGGT0uISBSt6YQV4/1wf9w2KSIT4U  
dZgaQt8L+TDXiz1/ywn4f11dU0K9vwIINIwp+8s9le7hn1vQPm7HAetLi5mRE30vzXJ6Eoai9RbfgFW7HpxffZLImdOgealQ51w+0Zu7Rx4jGWhWLMo  
WyrJQQw+ZXhgwY6pgESvD6LuI39m2hhJMC3781E8Q4OL+Jn17CysdjNpBH9AjNwGuI9Ad3y3q1u8z1849KzCZCx9GbG/n9YYy3fGnBrrvNY3nrwiA4c  
XKP4KfZU8OIQ3G1LJkK1d24lhhe9UBL3I1ySfMbvDbRoMOXESF6tCpMVLNMa4QaoVY7aThxDvAA6p51pftyPhCK3MJe4qBL4zTC3pXFJe+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)

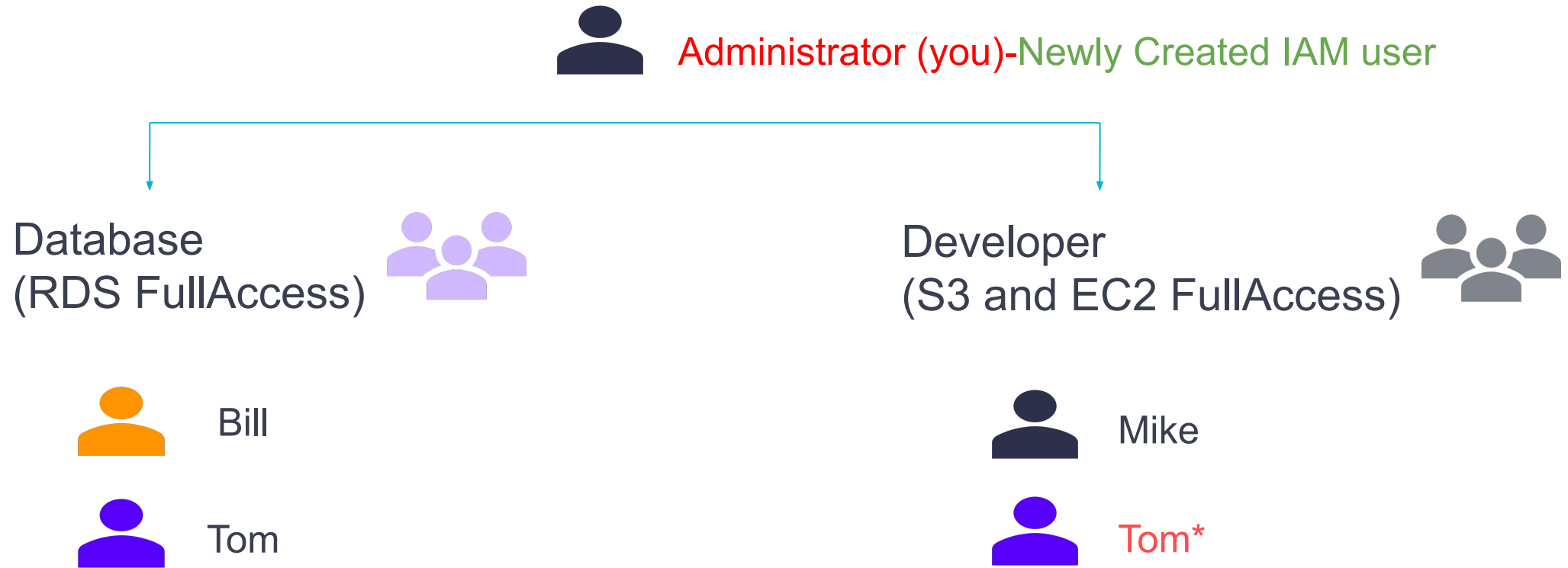
<https://lms.clarusway.com/mod/lesson/view.php?id=7626&pageid=7570>



# Let's get our hands dirty!



AWS Account owner - Root User (you)





# THANKS!

## Any questions?

You can find me at:

- ▶ @Guile - Instructor
- ▶ guile@clarusway.com

