BATCH : BATCH 85

LESSON : **AWS**

DATE : 05.08.2022

SUBJECT : **AWS-IAM**

techproeducation

techproeducation

techproeducation

techproeducation

techproedu

techproeducation.com

info@techproeducation.com

+1 (917) 768-7466

# IAM

**What is IAM ?**

- AWS IAM stands for Identity & Access Management and is the primary service that handles authentication and authorization processes within AWS environments.

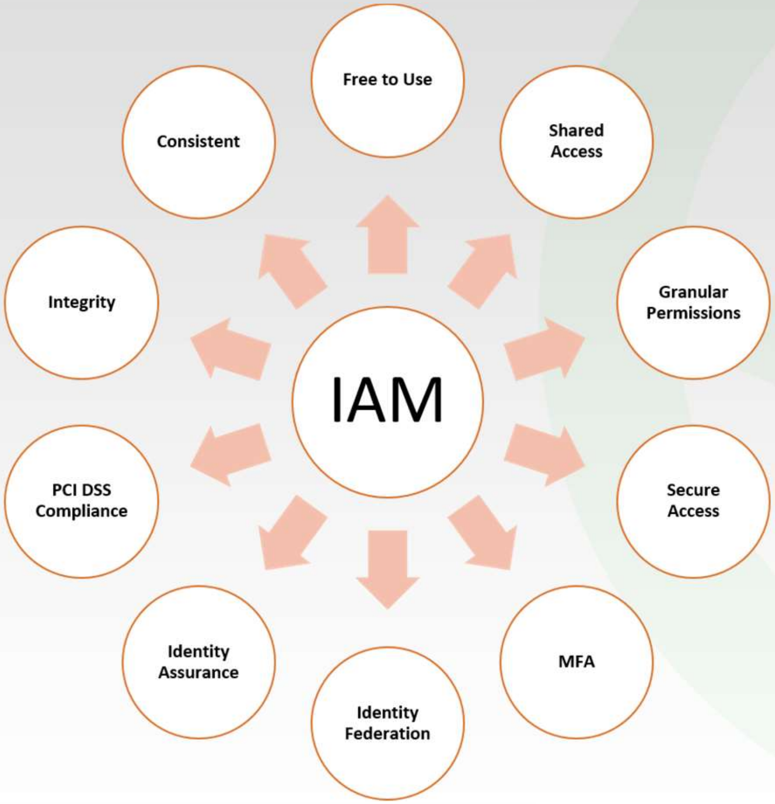
# IAM

**What is IAM?:**

- By using AWS IAM, you can manage users and their access level.
- All account settings are made through this service.
- It allows us to create and manage objects such as User, Group, Role, and Policy.
- Account owner can identify and allow the user to use specified services.
- All kinds of user password restrictions and multifactor authentication settings are also made through IAM.

# IAM

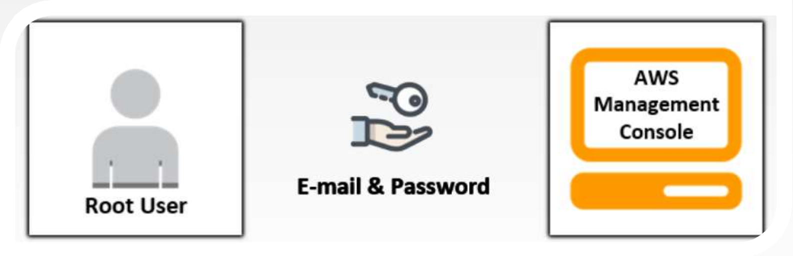
**IAM Features:**

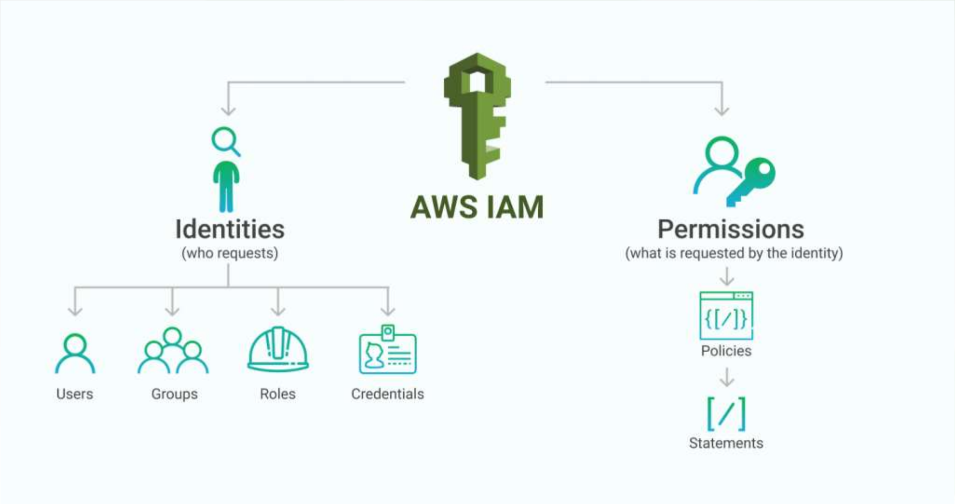


# IAM

### Categorizing IAM Components

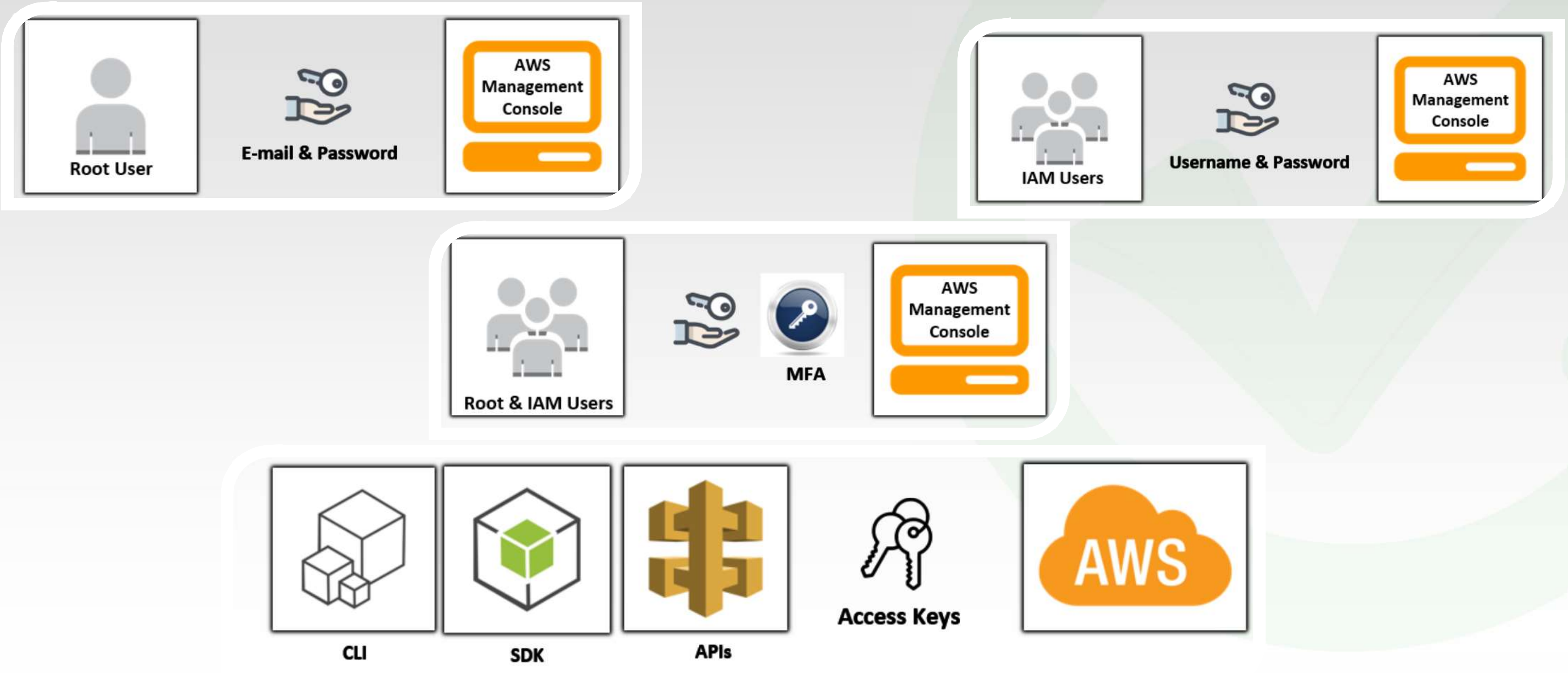
- IAM components can be mainly categorized under two term; identities and permissions.





# IAM

- IAM components can be mainly categorized under two term; identities and permissions.



# IAM

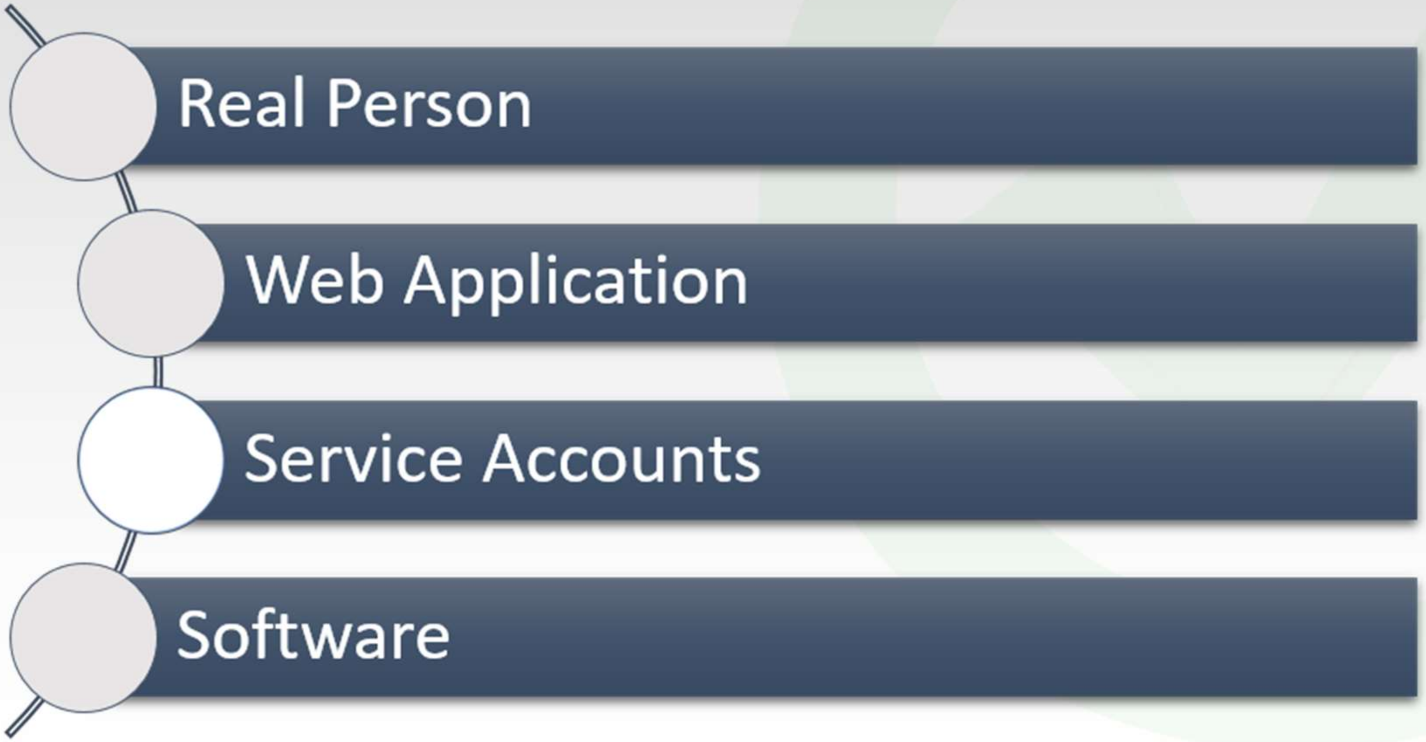
**What is an IAM User?**

An IAM user is an entity that you create in AWS.

- The IAM user represents the person or service who uses AWS services.
- A primary use for IAM users is to give people the ability to sign in to the AWS Management Console for interactive tasks and to make programmatic requests to AWS services using the API or CLI.
- A user in AWS consists of a name, a password to sign in to the AWS Management Console, and up to two access keys that can be used with the API or CLI.
- When you create an IAM user, you grant it permissions by making it a member of a group that has appropriate permission policies attached (recommended), or by directly attaching policies to the user.
- You can also clone the permissions of an existing IAM user, which automatically makes the new user a member of the same groups and attaches all the same policies.

# IAM

## IAM User Types



- Real Person
- Web Application
- Service Accounts
- Software

# IAM

**IAM – Users – Account Root User**

- By first creating an AWS account, you create a root user identity account that is used to log in to the AWS. This identity is called the AWS Account Root User.

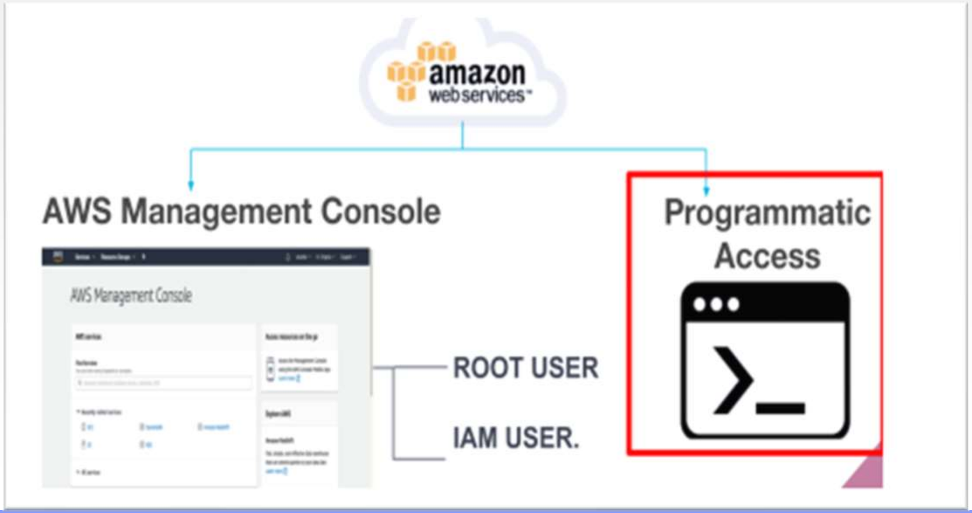
- A root user can create new IAM users and give them authorization for using AWS services within the account. The limit of creating new IAM users is restricted to 5000 users per account.

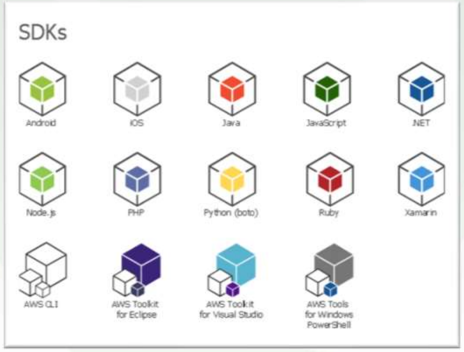
# IAM

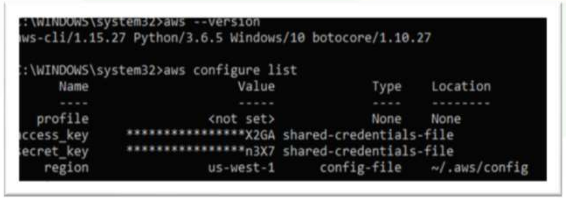
### What is an IAM user & Credentials

- An IAM user represents a person or service that interacts with AWS. You define the user within your AWS account.

- An IAM user consists of a name and a set of credentials. When creating a user, you can choose to provide the user:







```
:\WINDOWS\system32>aws --version
ws-cli/1.15.27 Python/3.6.5 Windows/10 botocore/1.10.27

:\WINDOWS\system32>aws configure list
    Name                    Value             Type    Location
    ----                    -----             ----    --------
 profile                <not set>             None    None
ccess_key     ****************X2GA shared-credentials-file
ecret_key     ****************n3X7 shared-credentials-file
  region                us-west-1      config-file    ~/.aws/config
```

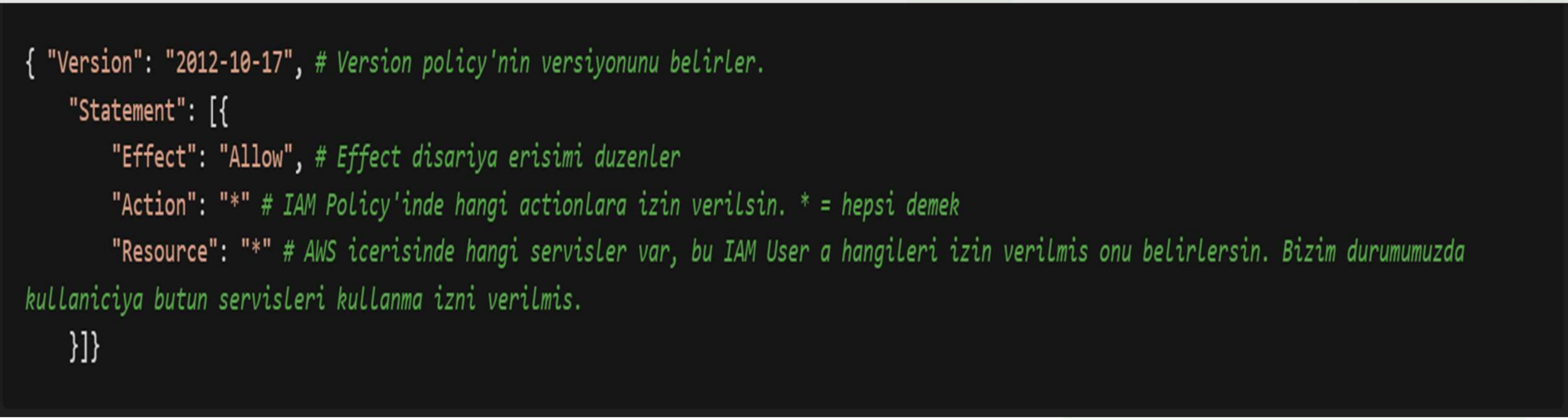
# IAM

**What is an IAM Policy?**

To manage access and provide permissions to AWS services and resources, you create IAM policies and attach them to IAM users, groups, and roles.

Most policies are stored in AWS as JSON documents with several policy elements.

# IAM



```
{ "Version": "2012-10-17", # Version policy'nin versiyonunu belirler.
    "Statement": [{
        "Effect": "Allow", # Effect disariya erisimi duzenler
        "Action": "*" # IAM Policy'inde hangi actionlara izin verilsin. * = hepsi demek
        "Resource": "*" # AWS icerisinde hangi servisler var, bu IAM User a hangileri izin verilmis onu belirlersin. Bizim durumumuzda
kullaniciya butun servisleri kullanma izni verilmis.
    }]}
```

# IAM

In this policy, there are four major JSON elements: Version, Effect, Action, and Resource.

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

The **Version** element defines the version of the policy language.

The Effect element specifies whether the statement will allow or deny access. In this policy, the Effect is "Allow", which means you're providing access to a particular resource.

The Action element describes the type of action that should be allowed or denied. In the above policy, the action is "*". This is called a wildcard, and it is used to symbolize every action inside your AWS account.

The **Resource** element specifies the object or objects that the policy statement covers. In the policy example above, the resource is also the wildcard "*". This represents all resources inside your AWS console.

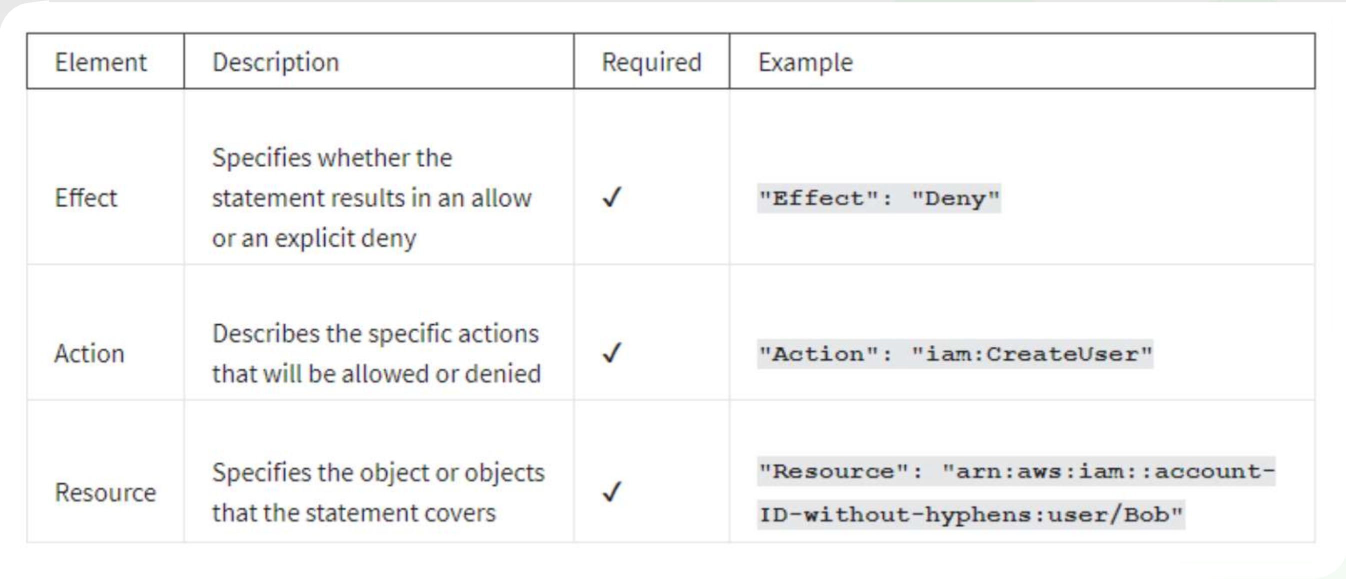
# IAM

- In this policy, there are four major JSON elements: Version, Effect, Action, and Resource.

```
{ "Version": "2012-10-17",
        "Statement": [{
           "Effect": "Allow",
           "Action": [
               "iam: ChangePassword",
               "iam: GetUser"
               ]
           "Resource":"arn:aws:iam::123456789012:user/${aws:username}"
        }]
```

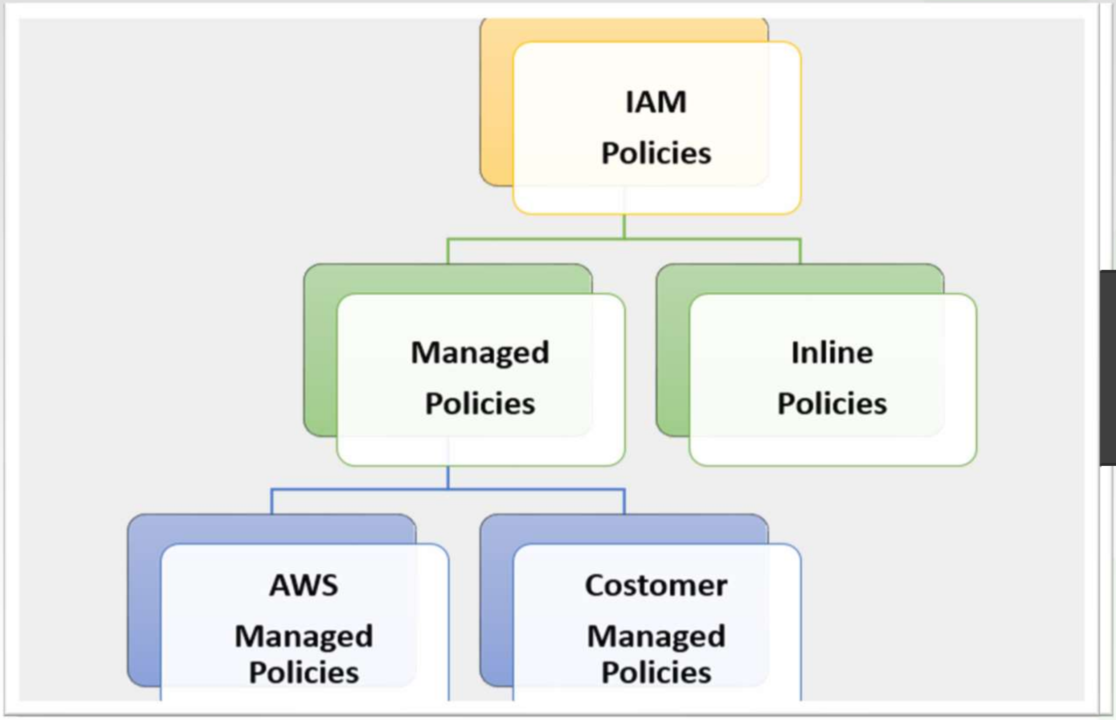
# IAM

- When creating a policy, it is required to have each of the following elements inside a policy statement.



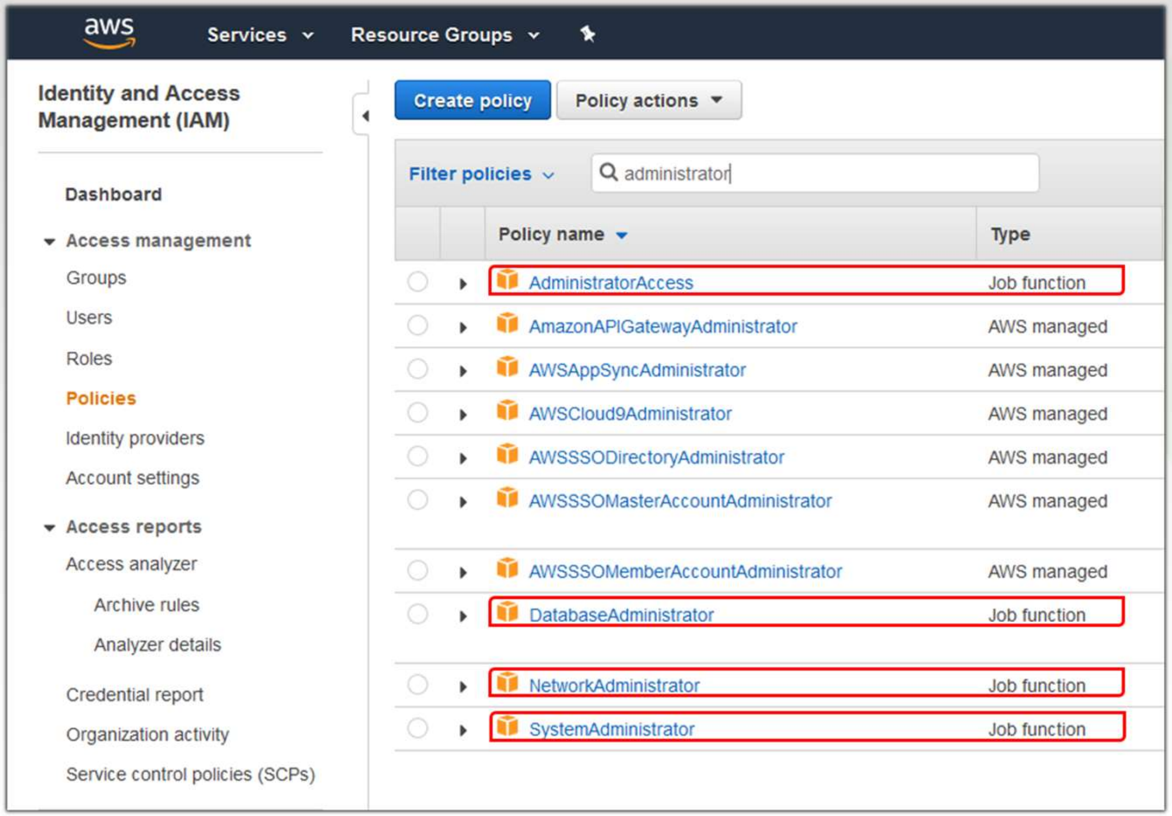
| Element | Description | Required | Example |
|---|---|---|---|
| Effect | Specifies whether the statement results in an allow or an explicit deny | ✓ | `"Effect": "Deny"` |
| Action | Describes the specific actions that will be allowed or denied | ✓ | `"Action": "iam:CreateUser"` |
| Resource | Specifies the object or objects that the statement covers | ✓ | `"Resource": "arn:aws:iam::account-ID-without-hyphens:user/Bob"` |

# IAM Policy Types



Job Function Policies

# Job Function Policies



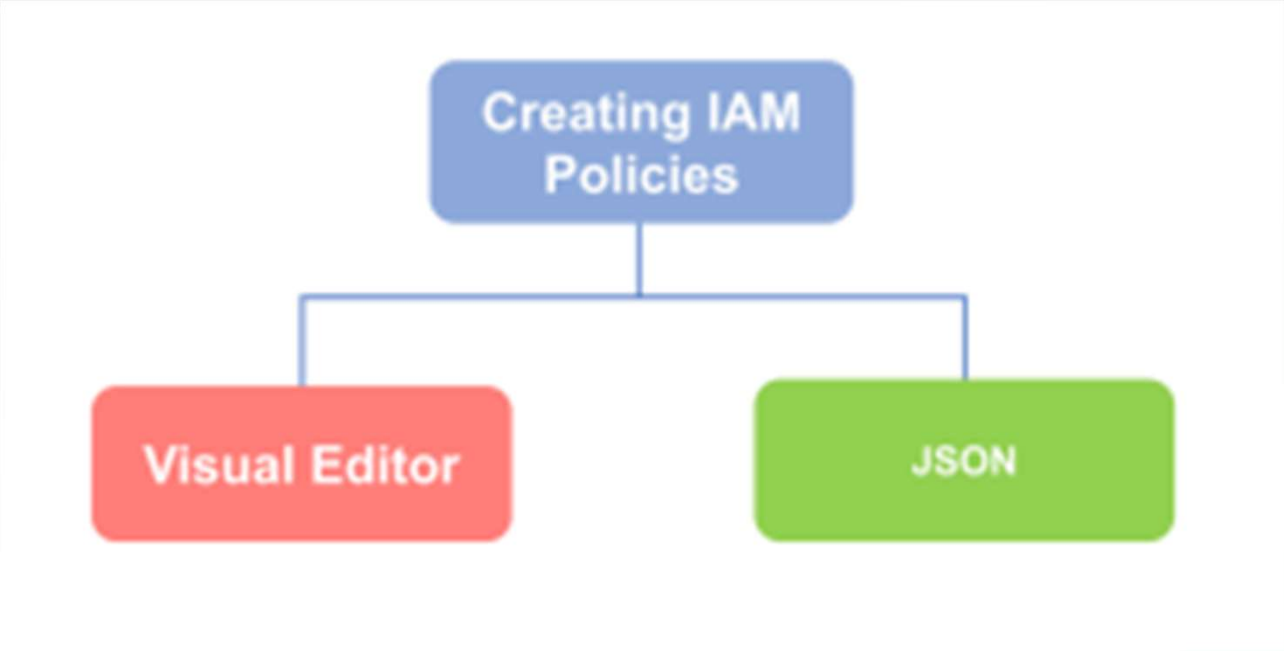
administrator

| Policy name | Type |
|---|---|
| AdministratorAccess | Job function |
| AmazonAPIGatewayAdministrator | AWS managed |
| AWSAppSyncAdministrator | AWS managed |
| AWSCloud9Administrator | AWS managed |
| AWSSSODirectoryAdministrator | AWS managed |
| AWSSSOMasterAccountAdministrator | AWS managed |
| AWSSSOMemberAccountAdministrator | AWS managed |
| DatabaseAdministrator | Job function |
| NetworkAdministrator | Job function |
| SystemAdministrator | Job function |

**Managed policies in job function status are listed below:**

- Administrator
- Billing
- Database Administrator
- Data Scientist
- Developer Power User
- Network Administrator
- Security Auditor
- Support User
- System Administrator
- View-Only User

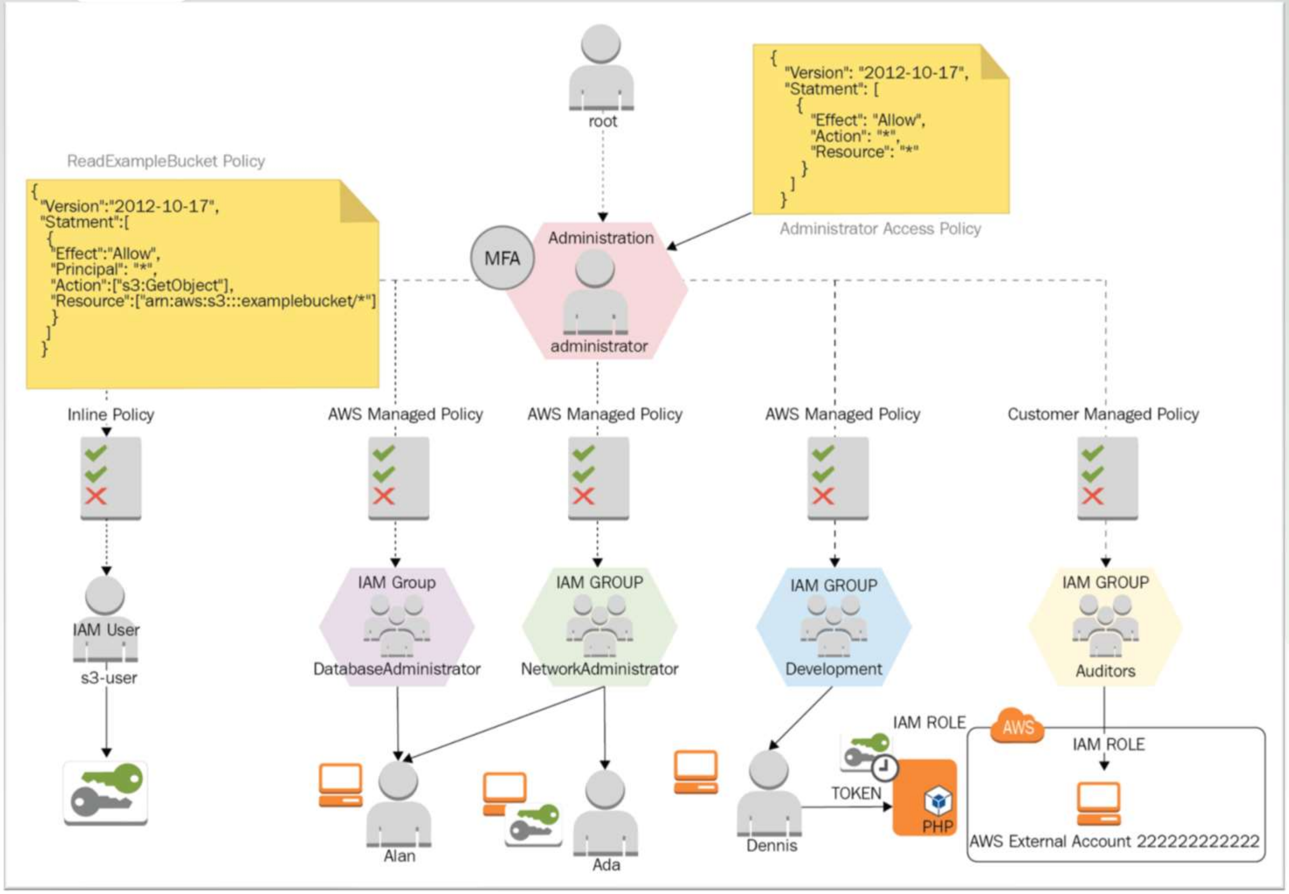
# Creating IAM Policies


Creating IAM Policies

Visual Editor

JSON

# Designing IAM Groups



ReadExampleBucket Policy

```
{
 "Version":"2012-10-17",
 "Statment":[
  {
   "Effect":"Allow",
   "Principal": "*",
   "Action":["s3:GetObject"],
   "Resource":["arn:aws:s3:::examplebucket/*"]
  }
 ]
}
```

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

Administrator Access Policy

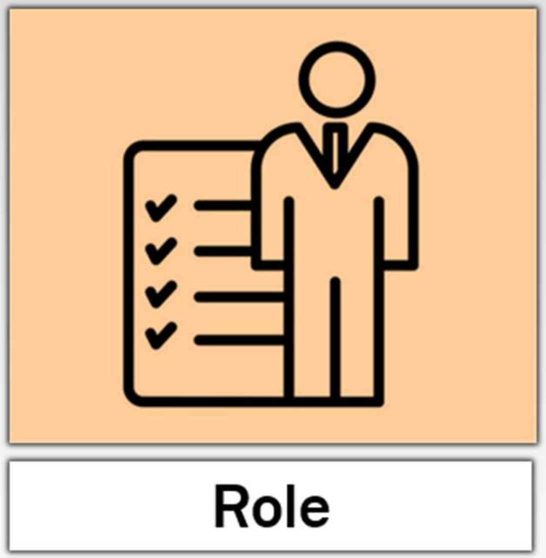
1
- Create IAM Groups as many as you need (max=300).

2
- Attach policies to the groups. (One or more managed/inline policies)

3
- If not, create IAM users for groups.

4
- Assign users to the groups.

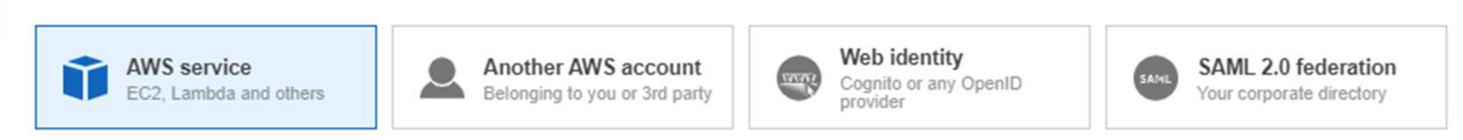
# IAM Roles



- It is the authorization system that we determine how and with which authorizations 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.

- Who can assume an IAM Role ?



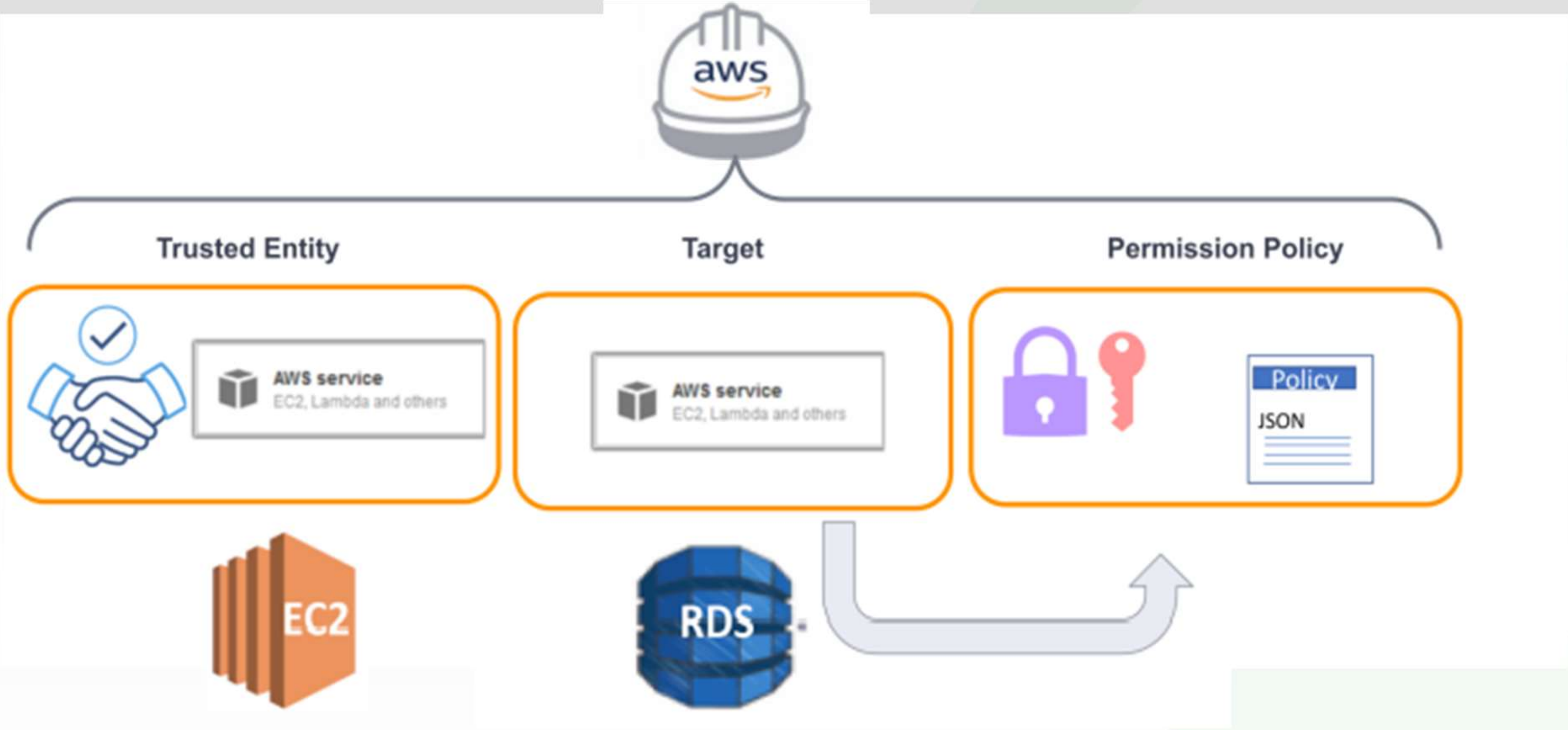
AWS service
EC2, Lambda and others

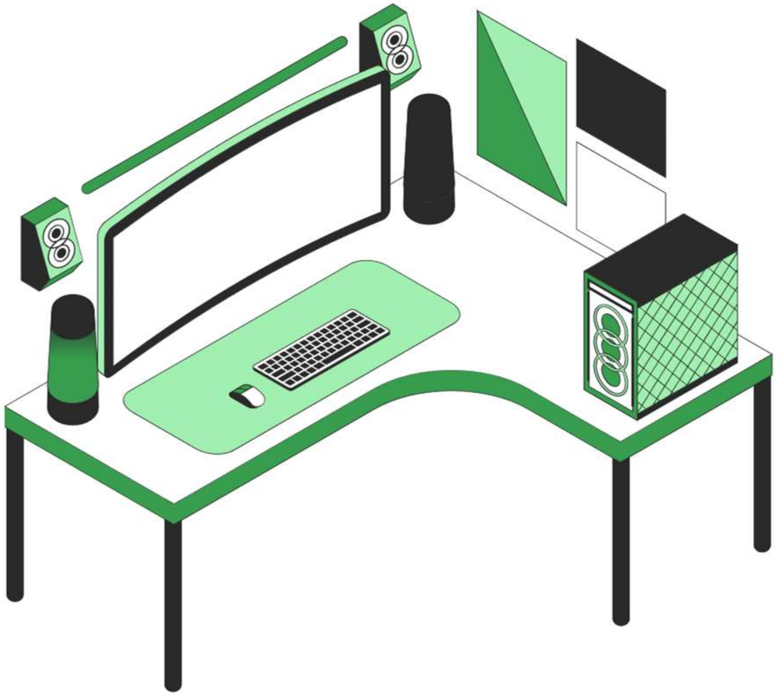
Another AWS account
Belonging to you or 3rd party

Web identity
Cognito or any OpenID provider

SAML 2.0 federation
Your corporate directory

# Anatomy of  Role





# Do you have any questions?

Send it to us! We hope you learned something new.