# AWS Foundation

**Security – IAM Part I**

# Agenda

1 Pre-IAM

2 Amazon Resource Name (ARN)

3 IAM Users

4 IAM Groups

5 Multi-Factor Authentication (MFA)

6 Demo

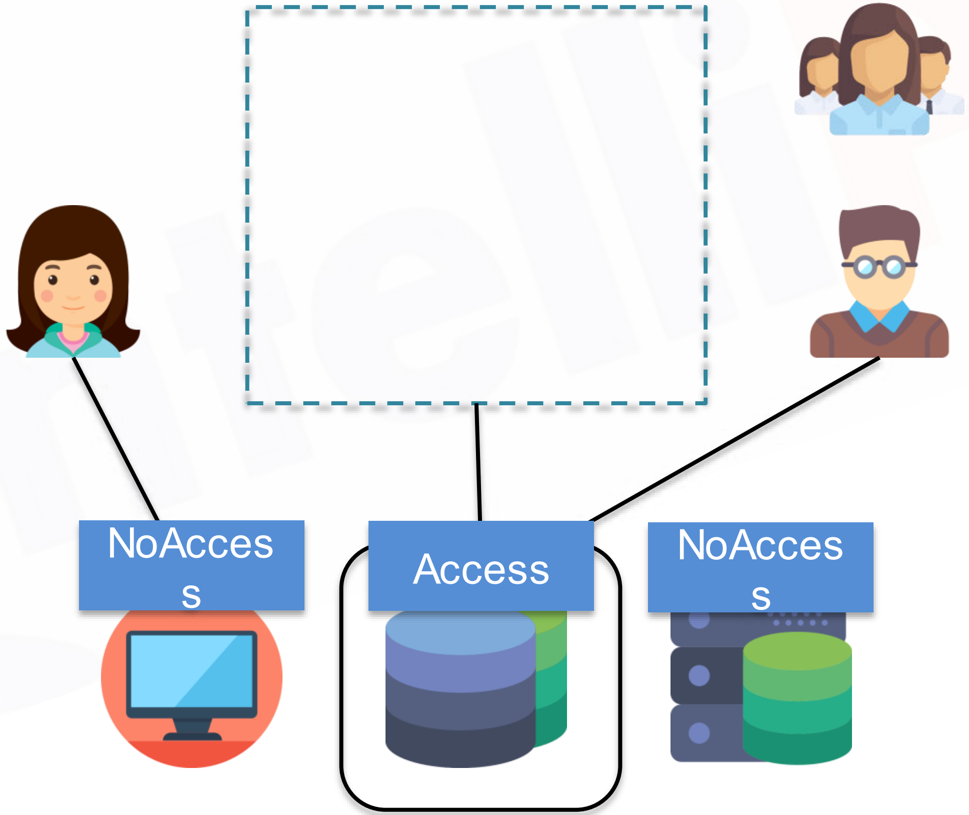
7 JSON

8 Policies

# Pre-IAM

## Users, Groups

- Authentication and Authorization
- Users
- Groups
- Permissions



# IAM Concepts

## Amazon Resource Name

- Amazon Resource Names uniquely identify AWS resources. Every resource in AWS is provided with an ARN.
- ARN Format

arn:partition:service:region:account-id:resource

arn:partition:service:region:account-id:resourcetype/resource

arn:partition:service:region:account-id:resourcetype:resource

Partition is a logical place where AWS resource resides in. For Standard AWS regions its "aws", for other regions its aws-partition. For example for Mumbai its "aws-in", for Beijing "aws-cn" etc.

Service identifies the AWS product. e.g. S3, IAM, RDS, EC2 etc.

Region is where the AWS resource resides. e.g. for N. Virginia its "us-east-1".

Numeric ID of the account which owns the AWS resource.

Varies by service, contains type of resource and the name or ID of the resource.

# IAM Concepts

## Amazon Resource Name

- EC2

Instance > arn:aws:ec2:region:account-id:instance/instance-id

AMI > arn:aws:ec2:region::image/image-id

Key-pair > arn:aws:ec2:region:account-id:key-pair/key-pair-name

N/W Interface > arn:aws:ec2:region:account-id:network-interface/eni-id

- EBS

Volume > arn:aws:ec2:region:account-id:volume/volume-id

Snapshot > arn:aws:ec2:region:account-id:snapshot/snapshot-id

# IAM Concepts

## Amazon Resource Name

- VPC

VPC > arn:aws:ec2:region:account-id:vpc/vpc-id

Route Table > arn:aws:ec2:region:account-id:route-table/route-table-id

SG > arn:aws:ec2:region:account-id:security-group/security-group-id

NACL > arn:aws:ec2:region:account-id:network-acl/nacl-id

IGW > arn:aws:ec2:region:account-id:internet-gateway/igw-id

Subnet > arn:aws:ec2:region:account-id:subnet/subnet-id

Peering > arn:aws:ec2:region:account-id:vpc-peering-connection/peering-id

# IAM Concepts

## Amazon Resource Name

- VPC

```
arn:aws:elasticloadbalancing:region:account-id:loadbalancer/app/load-balancer-name/load-balancer-id
```

```
arn:aws:elasticloadbalancing:region:account-id:listener/app/load-balancer-name/load-balancer-id/listener-id
```

```
arn:aws:elasticloadbalancing:region:account-id:listener-rule/app/load-balancer-name/load-balancer-id/listener-id/rule-id
```

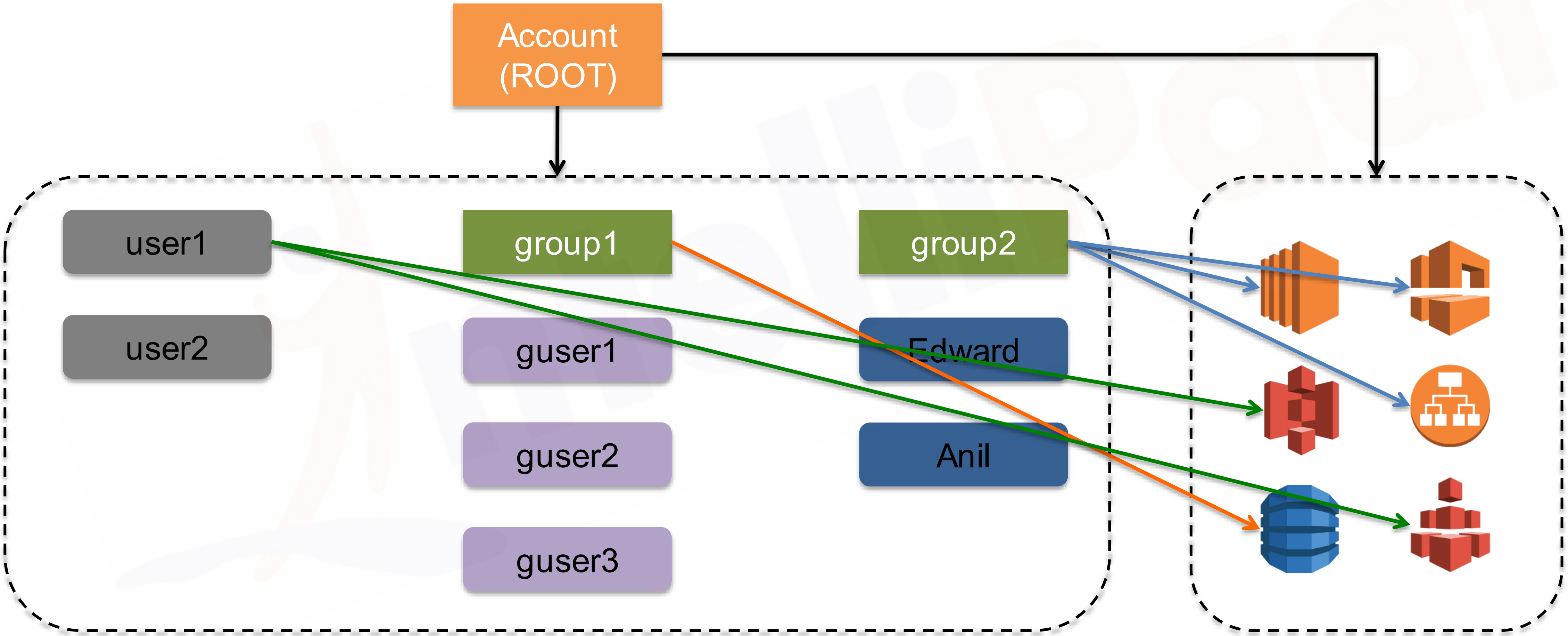
```
arn:aws:elasticloadbalancing:region:account-id:targetgroup/target-group-name/target-group-id
```

```
arn:aws:elasticloadbalancing:region:account-id:loadbalancer/name
```

- AS
- Route53
- S3
- DynamoDB
- RDS

# IAM Concepts

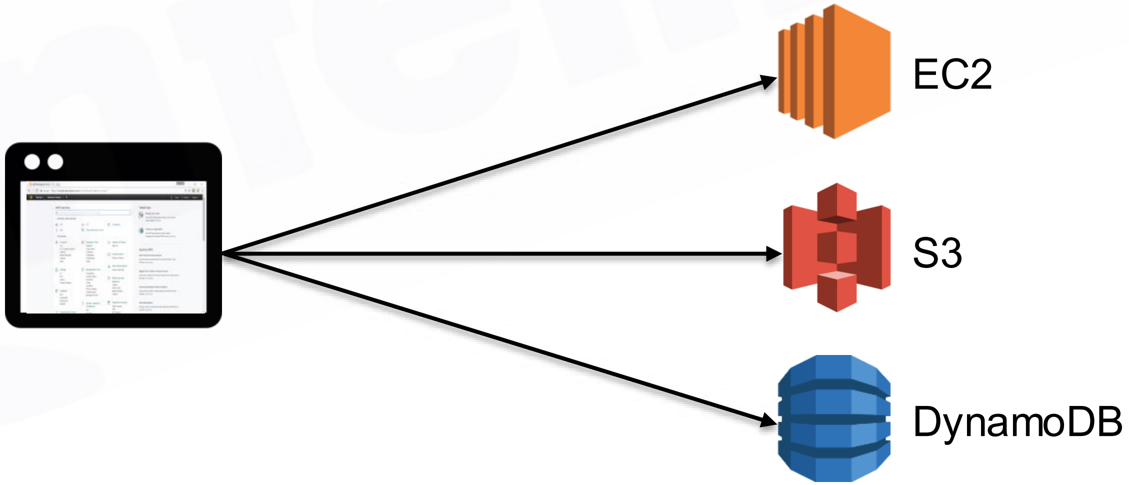
## IAM Hierarchy



# IAM Concepts

## IAM Users

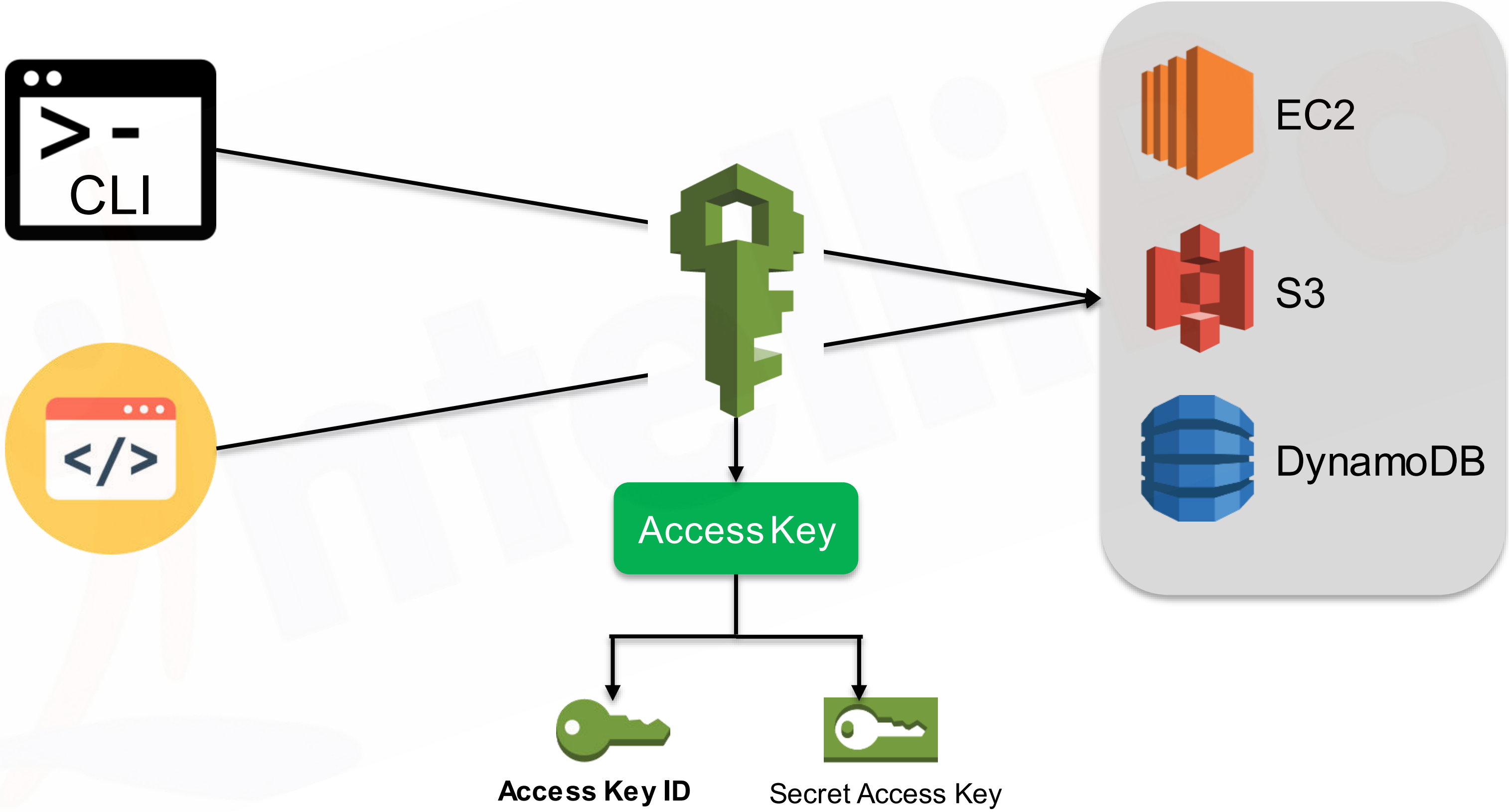
- Represents an entity that is created in AWS, can be a person or service.
- No permissions by default. Nothing is allowed.
- Access requirement
  - ➢ Programmatic Access: User needs to make API calls from programs or uses CLI to access AWS resources.
  - ➢ Management Console Access: User needs to access AWS resources from management console.



# IAM Concepts

## IAM Users

- Access Keys

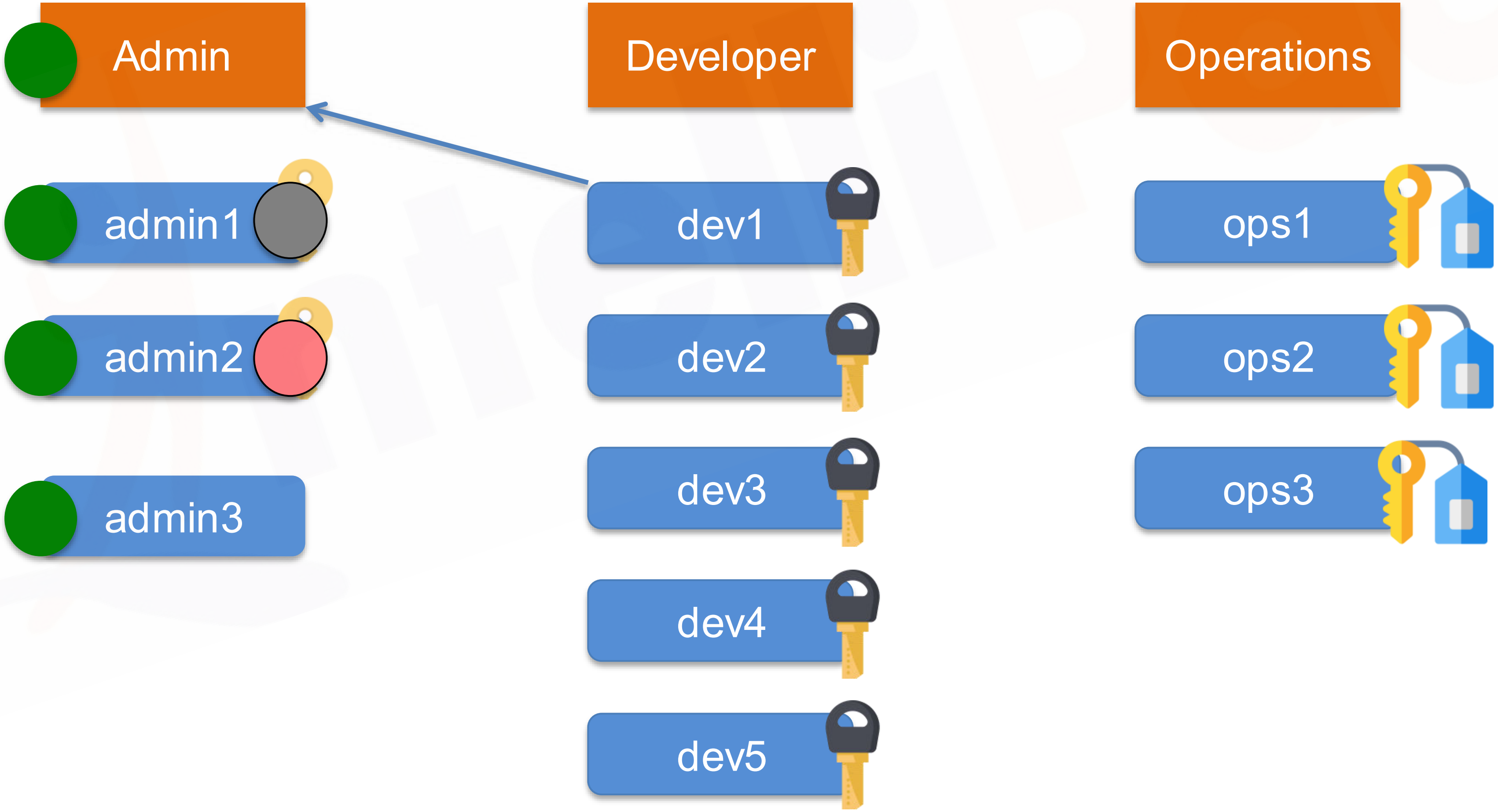


- Max 2 ACTIVE access keys at a time.
- When disabled access keys cannot be used to make CLI or API calls.

# IAM Concepts

IAM Groups

- Groups are collection of IAM users.



# IAM Concepts

## Multi-Factor Authentication

- Security Token Based

- SMS Based

# Demo 1: IAM Users & Groups

- Create 2 users using IAM console – admin1, user1.
- Use “admin1” and “user1” to sign in to the console.
- Login to the management console using both the users.
- Create 2 groups – awsfoundation, consolegroup.
- Add “admin1” to group awsfoundation and “user1” to consolegroup.
- Create access keys for both the users.
- Deactivate the access keys.
- Rotate access keys (only using CLI).
- Find unused passwords and access keys.
- Check credential report.
- Delete all the users and groups.
- Enable MFA for admin1 user.

# IAM Concepts

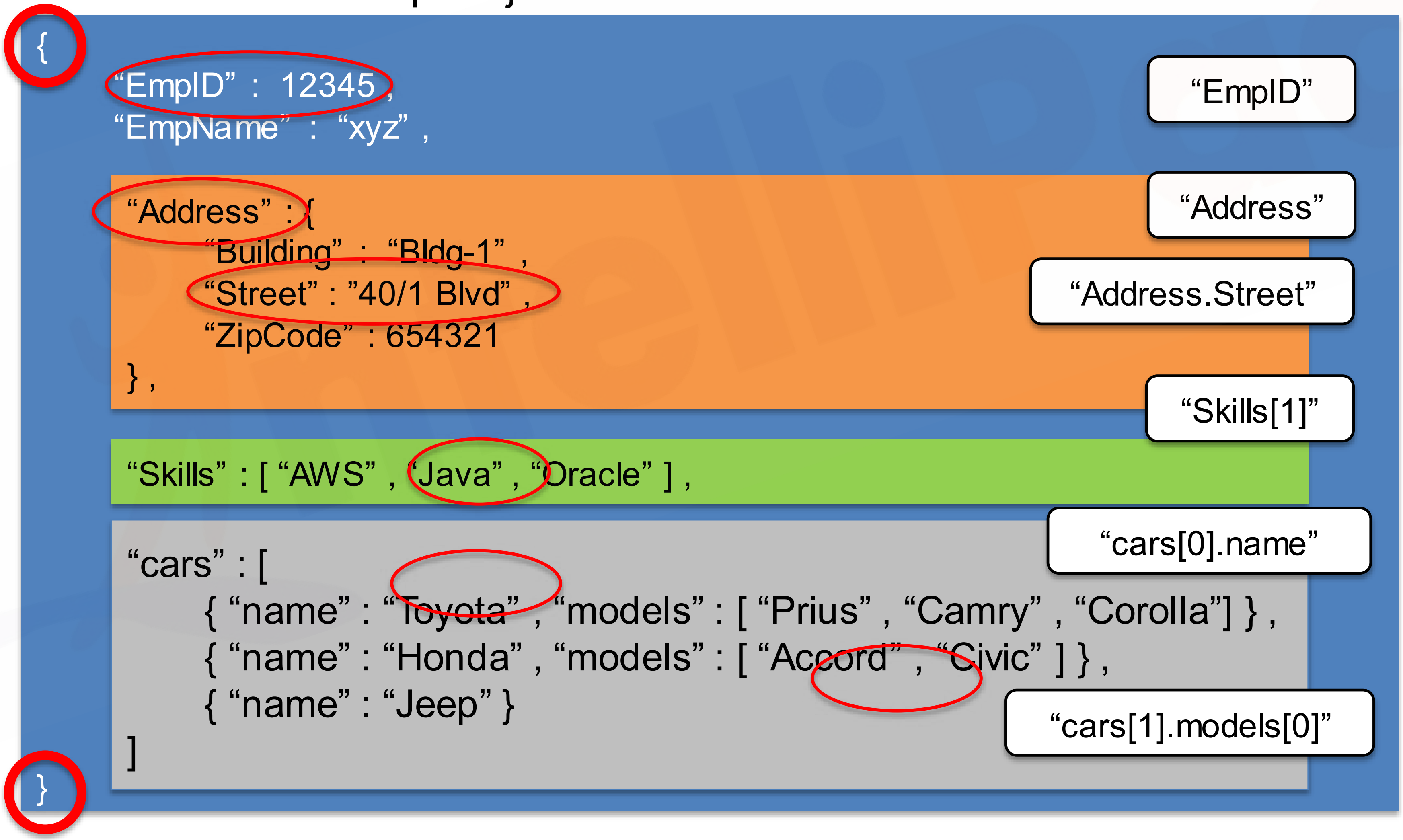
## The “ROOT” User

- Root user should not be used at all.
- MFA should be enabled for ROOT user as well.
- ROOT user can also be used for programmatic access.
- Access ID and Secret Access key can be created for ROOT user as well.

# IAM Concepts

## JSON

- Introduction to JSON – Java Script Object Notation.



```
{
    "EmpID" : 12345 ,
    "EmpName" : "xyz" ,

    "Address" : {
          "Building" : "Bldg-1" ,
          "Street" : "40/1 Blvd" ,
          "ZipCode" : 654321
    } ,

    "Skills" : [ "AWS" , "Java" , "Oracle" ] ,

    "cars" : [
          { "name" : "Toyota" , "models" : [ "Prius" , "Camry" , "Corolla"] } ,
          { "name" : "Honda" , "models" : [ "Accord" , "Civic" ] } ,
          { "name" : "Jeep" }
    ]
}
```

"EmpID"

"Address"

"Address.Street"

"Skills[1]"

"cars[0].name"

"cars[1].models[0]"

# IAM Concepts

JSON

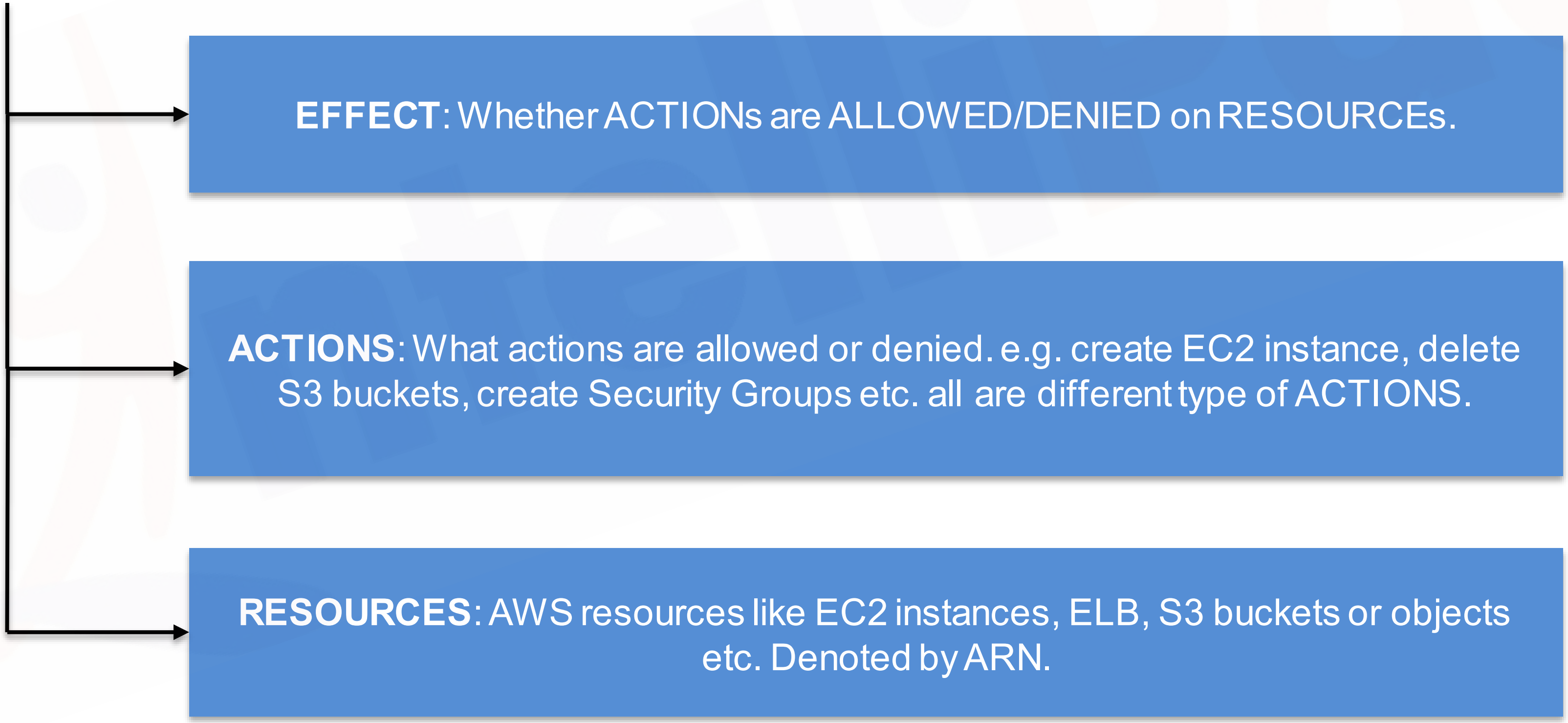
- Previous Record

```
{
    “EmpID” : 12345 ,
    “EmpName” : “xyz” ,
    “Address” : {
        “Building” : “Bldg-1” ,
        “Street” : “40/1 Blvd” ,
        “ZipCode” : 654321 ,
    } ,
    “Skills” : [ “AWS” , “Java” , “Oracle” ] ,
    “cars” : [
        { “name” : “Toyota” , “models” : [ “Prius” , “Camry” , “Corolla”] } ,
        { “name” : “Honda” , “models” : [ “Accord” , “Civic” ] } ,
        { “name” : “Jeep” }
    ]
}
```

# IAM Concepts

## IAM Policies

- Policies are JSON documents which mention what an user or group can do on AWS resources. It defines the Authorization paradigm for AWS resources.
- Contains 3 components at the least (EAR):



**EFFECT**: Whether ACTIONs are ALLOWED/DENIED on RESOURCEs.

**ACTIONS**: What actions are allowed or denied. e.g. create EC2 instance, delete S3 buckets, create Security Groups etc. all are different type of ACTIONS.

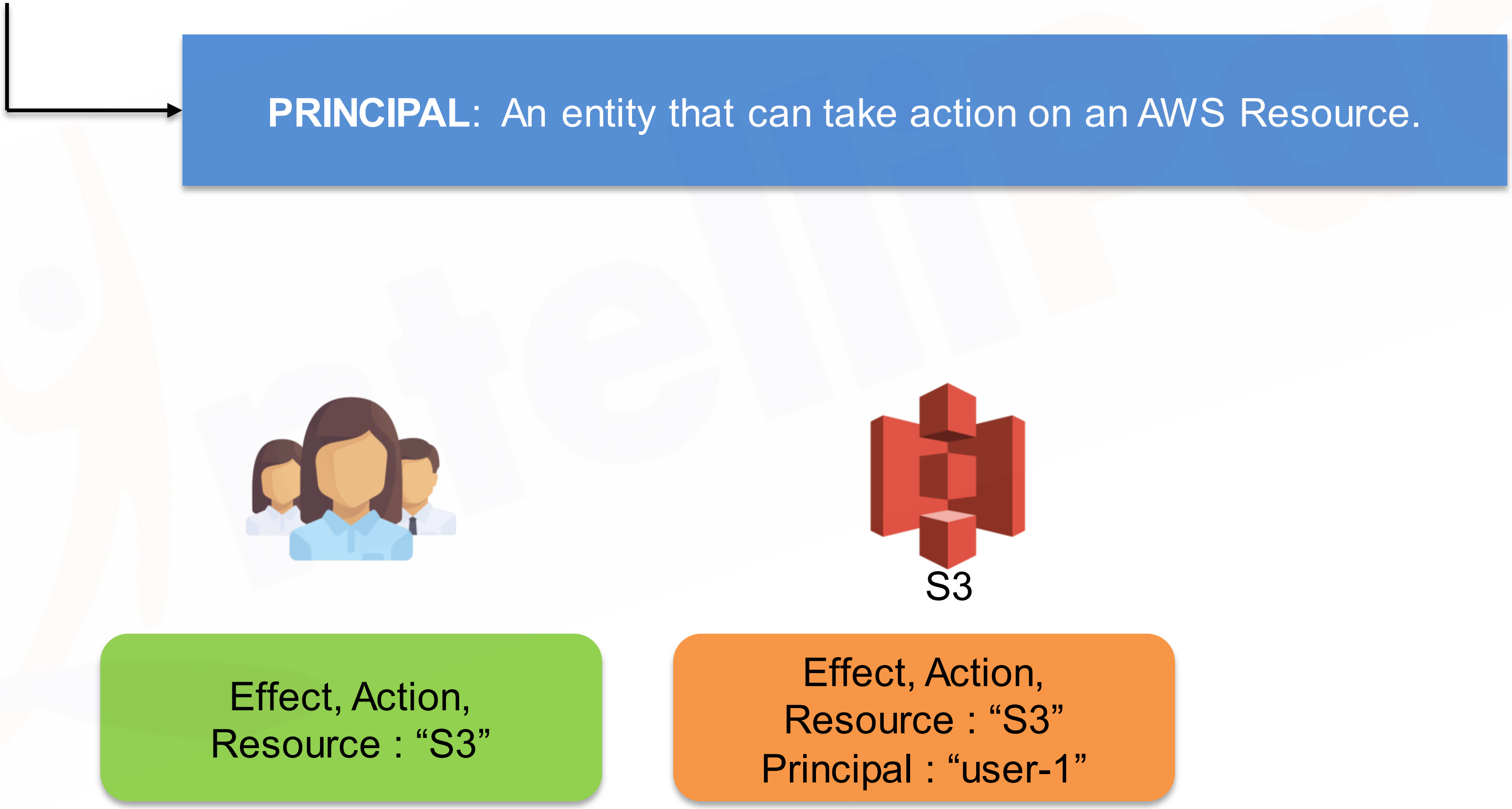
**RESOURCES**: AWS resources like EC2 instances, ELB, S3 buckets or objects etc. Denoted by ARN.

- Policies can be attached to Users or Groups.

# IAM Concepts

## IAM Policies

- Resource based policies: when policies are attached to resources.



**PRINCIPAL**: An entity that can take action on an AWS Resource.

S3

Effect, Action,
Resource : “S3”

Effect, Action,
Resource : “S3”
Principal : “user-1”

- S3, SNS, SQS.

# IAM Concepts

## IAM Policies

- Policy with a single statement

```
{
 "Version" : "2012-10-17" ,
 "Statement " : [
 {    "Effect" : "Allow" ,
      "Action " : "s3:ListBucket" ,
      "Resource" : "arn:aws:s3:::aws-foundation-bucket"
 }
 ]
}
```

Version →
2012-10-17, current version.
2008-10-17, previous version.

# IAM Concepts

## IAM Policies

- “Statement” : [ { } , { } , { } ]
  - Sid : Statement ID.
  - Effect : Allow/Deny.
  - Principal : ARN of AWS user, account or service which is allowed or denied access to a AWS resource.
  - Action : Specific action that is allowed or denied on an AWS resource.
  - Resource : ARN of the AWS resource.
  - Condition : Condition when a policy is in effect.

- AWS Managed Policies.
- Customer Managed Policies.
- Inline Policies

# IAM Concepts

## IAM Policies - Examples

- Allow users to access a specific S3 bucket (aws-foundation)

```
{
 "Version": "2012-10-17",
 "Statement": [  // Statement STARTs here
       {
             "Effect": "Allow",
             "Action": "s3:ListAllMyBuckets",
             "Resource": "arn:aws:s3:::*"
       } ,
```

```
    {
             "Effect": "Allow",
             "Action": [
                   "s3:ListBucket",
                   "s3:GetBucketLocation"
             ],
             "Resource": "arn:aws:s3:::aws-foundation"
  },
```

```
       {
             "Effect": "Allow",
             "Action": [
                   "s3:PutObject",
             "s3:GetObject",
             "s3:DeleteObject"
             ] ,
             "Resource": "arn:aws:s3:::aws-foundation/*"
       }
 ]  // Statement ENDs here
}
```

# Demo 2: IAM Policies

- Create a policy with the following
  - Allow to create EC2 instances.
  - Allow to list all EC2 instances.
  - Deny access to terminate EC2 instances.
  - Allow access to create Classic Load Balancer and launch instances under it.

- Create policy with the following
  - Allow access to create VPC, Security Groups, Subnets and Network ACLs.
  - Allow access to list all objects in a specific S3 bucket.

- Resource based policy using S3

# Demo 2: IAM Policies

- Select AMI – Need to see the AMIs
- Select VPC – Need to see all the available VPCs
- Select SG – Need to see all the available SGs
- Select Key-Pair
- Launch the instance

☎ India : +91-7847955955

US : 1-800-216-8930 (TOLL FREE)

✉ sales@intellipaat.com

24X7 Chat with our Course Advisor