



## Ch-2. IAM (Identity and Access Management)



# This chapter will cover following topics...

- Basics of Authentication and authorization
- Introduction of IAM
- Elements of IAM
- Creating and managing Users
- Creating and managing Groups
- Creating and managing Role
- Policy and policy classification
- Managed policy v/s resource based policy
- Few terminologies regarding policy
- Example of customer managed policy
- Example of resource based policy
- Overview of Active Directory Federation Services(ADFS)
- Overview of Web Identity Federation
- Overview of Security Token Service(STS)
- AWS CLI command respect to IAM

# 1. Basics of authentication and authorization

## Authentication:

It is the process of verifying who you are.

When you log on to a PC with a user name and password, you are authenticating.

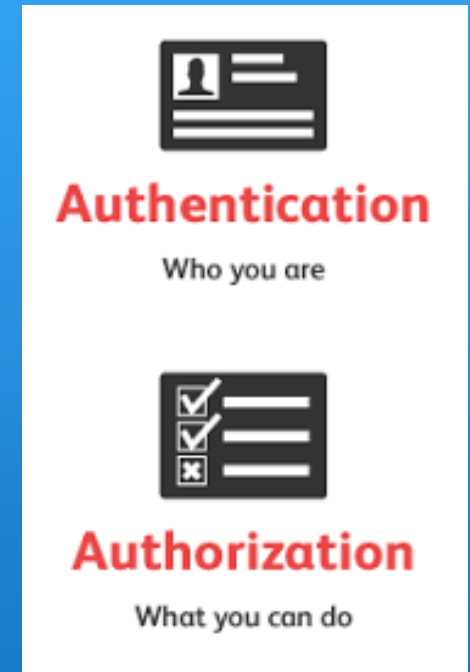
Authentication is about who somebody is.

## Authorization:

It is the process of verifying that you have access to something.

Gaining access to a resource (e.g. directory on a hard disk) because the permissions configured on it allow you access is authorization.

Authorization is about what they're allowed to do.



## 2. Introduction of IAM

IAM is a global service. Not a region specific.

It is a free service. No limitations.

It is specially designed to create and manage users, groups, roles and policies for securely controlling access to various AWS resources.

**I** – Identity – Identity to authorize someone

**A** – Access – Authority to access or to do something

**M** – Management

**It controls :**

who can use your resource – Authentication

what resources that can use in what ways – Authorization

## 2. Introduction of IAM(Conti...)

### AWS Root User:

When you create an account, it will automatically create root user.

Root account credential = email address + password used while creating account

Root user has complete and unrestricted access on AWS services.

Root user's permission cannot be altered by any other user.

On a newly created AWS account, it is recommended that you create individual IAM users based on the organizational need and assign them required permissions.

These non-root user accounts should be used for day-to-day activities.



# 3. Elements of IAM

Real-life organizational users and their permissions to access AWS resources as per their roles and responsibilities.

## User:

Any person / application

A user can access AWS resources with either a username and password or with an access key and secret key.

## Access Key:

An access key is a 20-character alphanumeric key that acts as a **user ID**.

## Secret Key:

A secret key is a 40-character alphanumeric key that acts as a **password** or **secret key**.

The access key and secret key are used together for initiating API, SDK, and CLI authentication.

# 3. Elements of IAM

## **Password policy:**

It specifies the complexity requirement of a password and defines the mandatory rotation period for a password associated with IAM users.

## **Multi-Factor Authentication (MFA):**

It is extra layer of security protection for user authentication that requires users to enter a six-digit token on top of the username and password.

## **Group:**

It is a collection of IAM users.

## **Role:**

It does not have any identity credential with it.

It required one or more IAM policies that define permission.

## **Policy:**

It is a document written in JSON format that formally states one or more permissions as per the IAM policy standards.



## 4. Creating and managing users

Users can be actual user or application.

Usually, an individual user is authenticated by username and password.

Similarly, programmatic access (that is SDKs and CLIs, also known as applications) are authenticated using an access key and secret key.

By default, access key and secret keys are not generated for all users.

It is recommended to generate access key and secret key only for those users who wants to access AWS resources via API,CLI or SDKs.

By default, only root user has billing access.

Newly created user does not have any privileges in AWS, unless and until you attach proper policies.





## 4. Creating and managing users

Services ▾

Resource Groups ▾

EC2

S3

IAM

VPC

🔔

rajan13 ▾

Global ▾

Add user

1234

Set user details

---

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name\*

+

Add another user

Select AWS access type

---

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

Access type\*

☐ Programmatic access

Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

☐ AWS Management Console access

Enables a **password** that allows users to sign-in to the AWS Management Console.

\* Required

Cancel


Next: Permissions


## 4. Creating and managing users(Conti...)


### Add user

1234

▼ Set permissions

 Add user to group

 Copy permissions from existing user

 Attach existing policies directly

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

### Add user to group

Create group

Refresh

Showing 1 result

Group ▼	Attached policies
<input type="checkbox"/> administrator	<a href="#">AdministratorAccess</a>

Cancel

Previous

Next: Review

## 4. Creating and managing users(Conti...)

### Review

Review your choices. After you create the user, you can view and download the autogenerated password and access key.

### User details

<b>User name</b>	RajanT
<b>AWS access type</b>	Programmatic access and AWS Management Console access
<b>Console password type</b>	Autogenerated
<b>Require password reset</b>	Yes
<b>Permissions boundary</b>	Permissions boundary is not set

### Permissions summary

The user shown above will be added to the following groups.

Type	Name
Group	<a href="#">administrator</a>

[Cancel](#)[Previous](#)[Create user](#)

## 4. Creating and managing users(Conti...)

Services ▾

Resource Groups ▾

EC2

S3

IAM

VPC

🔔

rajan13 ▾

Global ▾

Support ▾

### Add user

1

2

3

4

✓

**Success**  
You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.  
  
Users with AWS Management Console access can sign-in at: [https://\[REDACTED\]signin.aws.amazon.com/console](https://[REDACTED]signin.aws.amazon.com/console)

Download .csv

	User	Access key ID	Secret access key	Password	Email login instructions
▶	✓ RajanT	[REDACTED]	***** Show	***** Show	Send email ↗

## 5. Creating and managing groups

Any group have many users and one user can be a member of many groups

Groups can not be nested. It means one group can not be a part of another group

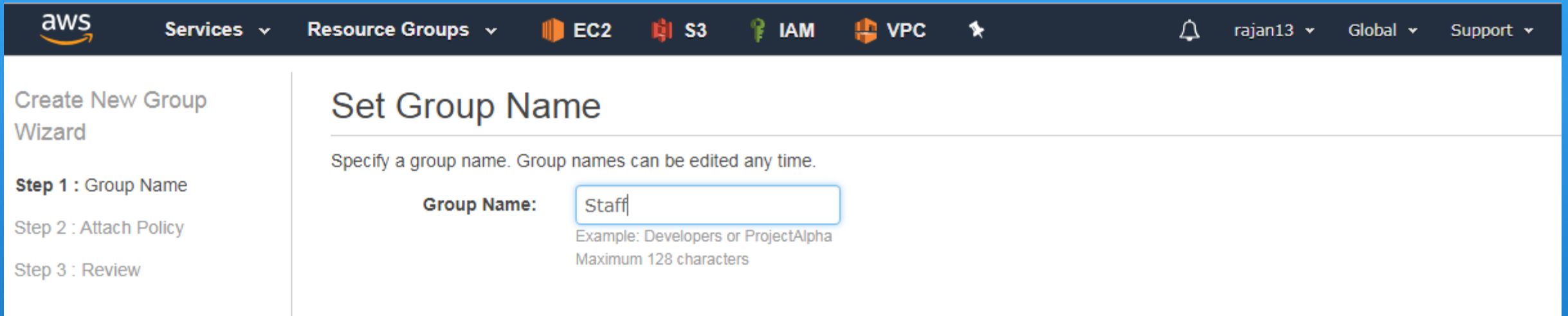
By default, user is not a part of any group

By default soft limit is : 100 IAM groups and 5000 users

Group name length up to 128 chars. long



## 5. Creating and managing groups(Conti...)



The screenshot shows the AWS IAM console interface for creating a new group. The top navigation bar includes the AWS logo, a 'Services' dropdown, and links to 'Resource Groups', 'EC2', 'S3', 'IAM', and 'VPC'. On the right side of the navigation bar, there is a notification bell, the user 'rajan13', a 'Global' region selector, and a 'Support' link. The left sidebar contains the 'Create New Group Wizard' with three steps: 'Step 1 : Group Name' (selected), 'Step 2 : Attach Policy', and 'Step 3 : Review'. The main content area is titled 'Set Group Name' and includes the instruction 'Specify a group name. Group names can be edited any time.' Below this, there is a 'Group Name:' label followed by a text input field containing the text 'Staff'. Underneath the input field, there is a hint that says 'Example: Developers or ProjectAlpha' and a note 'Maximum 128 characters'.

aws Services ▾ Resource Groups ▾ EC2 S3 IAM VPC

rajan13 ▾ Global ▾ Support ▾

Create New Group Wizard

**Step 1 : Group Name**

Step 2 : Attach Policy

Step 3 : Review








### Set Group Name

Specify a group name. Group names can be edited any time.

**Group Name:**

Example: Developers or ProjectAlpha  
Maximum 128 characters

## 5. Creating and managing groups(Conti...)

 **Services** ▾ **Resource Groups** ▾  **EC2**  **S3**  **IAM**  **VPC**   **rajan13** ▾ **Global** ▾ **Support** ▾






**Create New Group Wizard**  
**Step 1 : Group Name**  
**Step 2 : Attach Policy**  
**Step 3 : Review**

### Attach Policy

Select one or more policies to attach. Each group can have up to 10 policies attached.

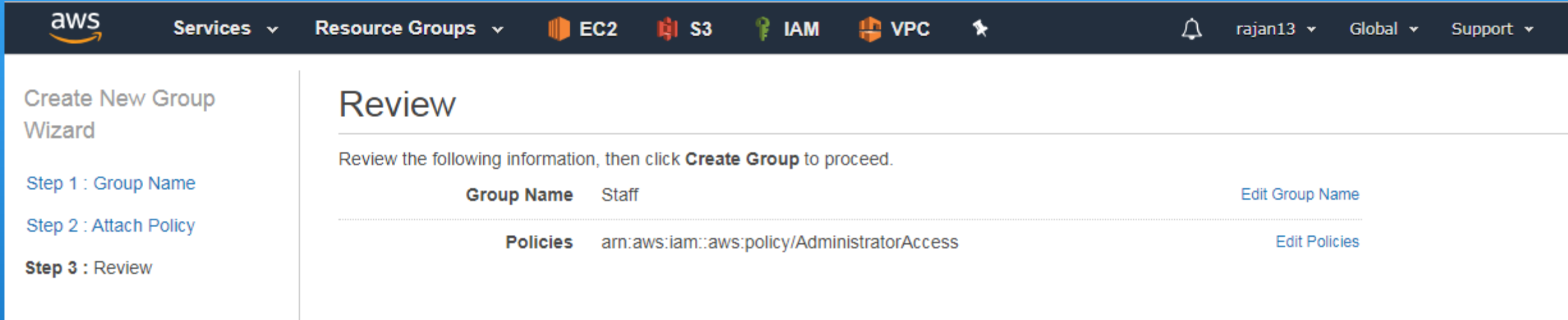
**Filter:** Policy Type ▾

Showing 358 results

		Policy Name ▴ ▾	Attached Entities ▴ ▾	Creation Time ▴ ▾	Edited Time ▴ ▾
<input type="checkbox"/>		IAMUserChangePassword	3	2016-11-15 05:55 UTC+0530	2016-11-16 04:48 UTC+...
<input type="checkbox"/>		AdministratorAccess	1	2015-02-07 00:09 UTC+0530	2015-02-07 00:09 UTC+...
<input type="checkbox"/>		AmazonDynamoDBReadOn...	1	2015-02-07 00:10 UTC+0530	2018-01-10 01:16 UTC+...
<input type="checkbox"/>		AmazonS3FullAccess	1	2015-02-07 00:10 UTC+0530	2015-02-07 00:10 UTC+...
<input type="checkbox"/>		s3crr_for_rajan13bk_to_raja...	1	2017-11-26 13:53 UTC+0530	2017-11-26 13:53 UTC+...
<input type="checkbox"/>		s3crr_for_rajan13bkuseast_t...	1	2017-11-26 14:03 UTC+0530	2017-11-26 14:03 UTC+...
<input type="checkbox"/>		s3crr_for_rajanbkoregon1_t...	1	2018-01-13 10:36 UTC+0530	2018-01-13 10:36 UTC+...
<input type="checkbox"/>		AlexaForBusinessDeviceSet...	0	2017-11-30 22:17 UTC+0530	2017-11-30 22:17 UTC+...

**Cancel** **Previous** **Next Step**

## 5. Creating and managing groups(Conti...)



The screenshot shows the AWS IAM console interface. The top navigation bar includes the AWS logo, a 'Services' dropdown, and links to 'Resource Groups', 'EC2', 'S3', 'IAM', and 'VPC'. On the right, there is a notification bell, the user 'rajan13', a 'Global' region selector, and a 'Support' link. The left sidebar contains the 'Create New Group Wizard' with three steps: 'Step 1 : Group Name', 'Step 2 : Attach Policy', and 'Step 3 : Review' (which is the active step). The main content area is titled 'Review' and contains the instruction: 'Review the following information, then click **Create Group** to proceed.' Below this, there is a table with two rows of configuration details.

<b>Group Name</b>	Staff	<a href="#">Edit Group Name</a>
<b>Policies</b>	arn:aws:iam::aws:policy/AdministratorAccess	<a href="#">Edit Policies</a>



## 6. Creating and managing role

Role is AWS identity

IAM policies can be associated with IAM user or IAM group

IAM role cannot be associated with user or group

Role name – up to 64 chars

It is recommended that AWS resource permission in the form of IAM policies are attached to an IAM role rather than being attached to IAM users or groups.



## 6. Creating and managing role(Conti...)

Services

Resource Groups

EC2

S3

IAM

VPC

rajan13

Global


Create role


1


2


3

Select type of trusted entity

**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

Allows AWS services to perform actions on your behalf. [Learn more](#)

Choose the service that will use this role

**EC2**  
Allows EC2 instances to call AWS services on your behalf.

**Lambda**  
Allows Lambda functions to call AWS services on your behalf.

API Gateway	Config	EMR	IoT	Rekognition
AWS Support	DMS	ElastiCache	Kinesis	S3
AppSync	Data Lifecycle Manager	Elastic Beanstalk	Lambda	SMS

\* Required

Cancel

Next: Permissions

## 6. Creating and managing role(Conti...)

Services ▾

Resource Groups ▾

EC2

S3

IAM

VPC

🔔

rajan13 ▾

Global ▾

Create role

1

2

3

▾ Attach permissions policies

Choose one or more policies to attach to your new role.







Create policy

↺

Filter policies ▾

🔍 Search

Showing 403 results

	Policy name ▾	Used as	Description
<input type="checkbox"/>	▶  AdministratorAccess	Permissions policy (1)	Provides full access to AWS services ...
<input type="checkbox"/>	▶  AlexaForBusinessDeviceSetup	None	Provide device setup access to Alexa...
<input type="checkbox"/>	▶  AlexaForBusinessFullAccess	None	Grants full access to AlexaForBusines...
<input type="checkbox"/>	▶  AlexaForBusinessGatewayExecution	None	Provide gateway execution access to ...
<input type="checkbox"/>	▶  AlexaForBusinessReadOnlyAccess	None	Provide read only access to AlexaFor...
<input type="checkbox"/>	▶  AmazonAPIGatewayAdministrator	None	Provides full access to create/edit/dele...

\* Required

Cancel

Previous

Next: Review

## 6. Creating and managing role(Conti...)

### Create role

1

2

3

#### Review

Provide the required information below and review this role before you create it.

**Role name\***

Use alphanumeric and '+=, @-\_' characters. Maximum 64 characters.



**Role description**

Maximum 1000 characters. Use alphanumeric and '+=, @-\_' characters.

**Trusted entities**

AWS service: s3.amazonaws.com

**Policies**

 [AmazonS3FullAccess](#) 

\* Required

[Cancel](#) [Previous](#) [Create role](#)

## 7. Policy and Policy Classification

### Policy:

Policy is a JSON document that states one or more permissions to access AWS resources

A policy can be associated with one or more IAM users, groups, roles and resources.



## 7. Policy and Policy Classification(Conti...)

### Policy Classification:

#### Managed policy:

- Customer managed policy
- AWS managed policy

#### Inline policy:

- Customer managed policy with 1 to 1 mapping between policy and principal.
- Directly can be attached with single user, group and role.

#### Resource based policy:

- It is an inline policy but specific to any AWS service / resource.

## 8. Managed policy v/s resource based policy

The major difference between a managed policy and a resource-based policy is -

A resource based policy specifies **who has access to the resource (principal) and list of permitted actions**, whereas in a managed policies **only list of actions is specified, not the principal** entity.

An IAM resource-based policy can also be generated with the help of the AWS policy generator.

The URL for the AWS policy generator is  
***<https://awspolicygen.s3.amazonaws.com/policygen.html>***

## 9. Few terminologies regarding Policy

### **Version:**

Specifies IAM policy language version.

Latest and current version is 2012-10-17. It should be used for all the policies (that is, managed or resource-based policy).

### **Effect:**

It defines whether the specified list of actions in Action elements on specified resources specified in Resource elements are allowed or denied.

By default, every service and resource is denied the access. Usually, policies are written to allow resource access.

### **Actions:**

This defines a list of actions.

Each AWS service has got its own set of actions.

### **Resources:**

This section specifies the list of resources on which the preceding specified list of actions are allowed.



## 10. Example of customer managed policy

```
{  
  "Version": "2012-10-17",  
  "Statement": {  
    "Effect": "Allow",  
    "Action": "s3:*",  
    "Resource": "arn:aws:s3:::mybucket"  
  }  
}
```

Amazon Resource Name (ARN) is a unique identifier for each of the AWS resources.

It is used in IAM policies, API calls, and wherever it's required to identify AWS resources unambiguously.

An basic example of ARN is as follows:

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

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

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

# 11. Example of resource based policy

## Account A:

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "AccountBAccess1",
    "Effect": "Allow",
    "Principal": {"AWS": "111122223333"},
    "Action": "s3:*",
    "Resource": [
      "arn:aws:s3:::mybucket",
      "arn:aws:s3:::mybucket/*"
    ]
  }
}
```

## Account B:

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "s3:List*",
    "Resource": [
      "arn:aws:s3:::mybucket",
      "arn:aws:s3:::mybucket/*"
    ]
  }
}
```

### **In first policy,**

Here, account A's S3 bucket is named mybucket, and account B's account number is 111122223333. It does not specify any individual users or groups in account B, only the account itself.

### **In second policy,**

To implement this policy, account B uses IAM to attach it to the appropriate user (or group) in account B. Means user B only mention the bucket name of user A.

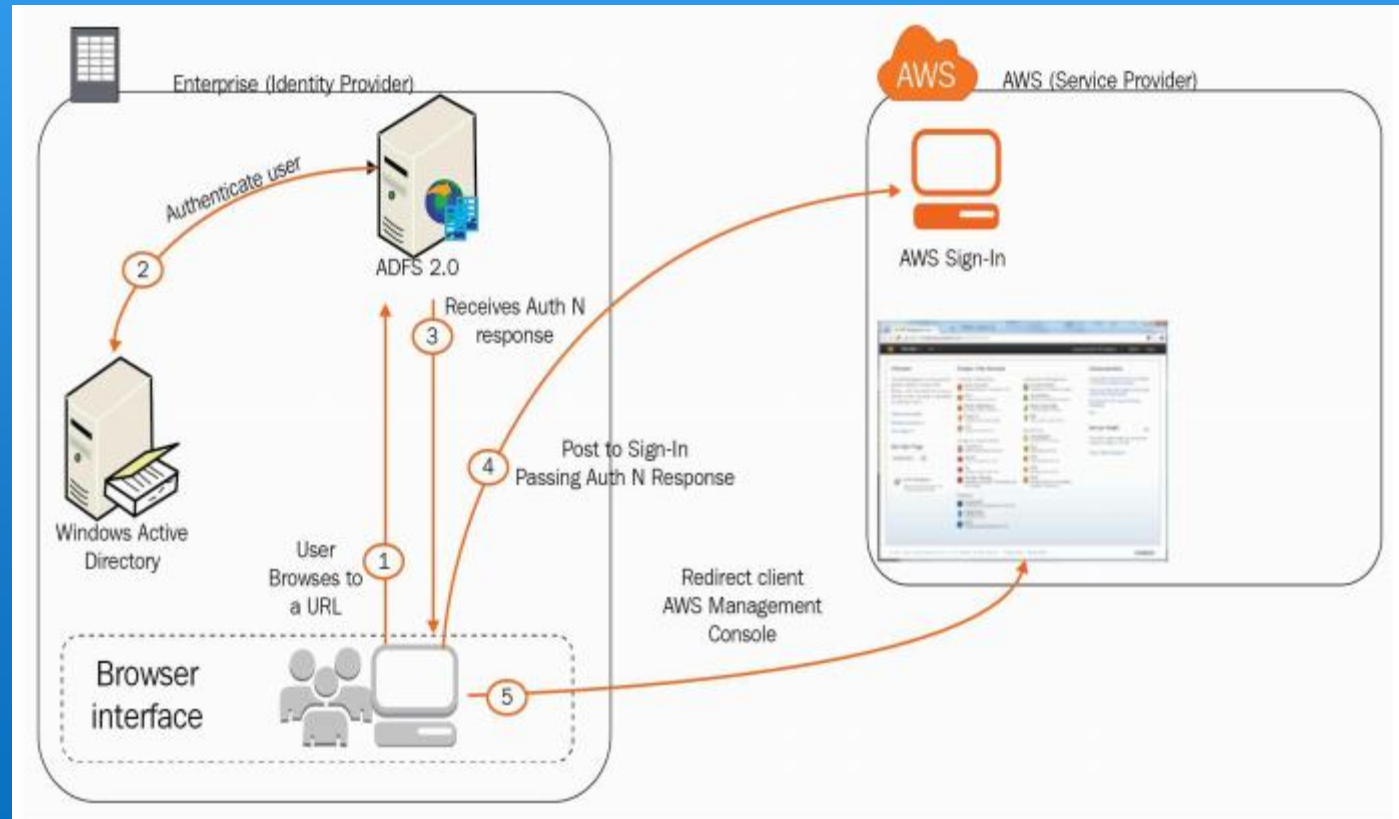
# 11. Overview of Active Directory Federation Services(ADFS)

External identities are called Federated users(Not permanent users)

Federated users are granted secure access to your resources without creating IAM users

We cannot attach policies to unknown users

So, login with organization's AD that provides token based SSO service for accessing systems and applications



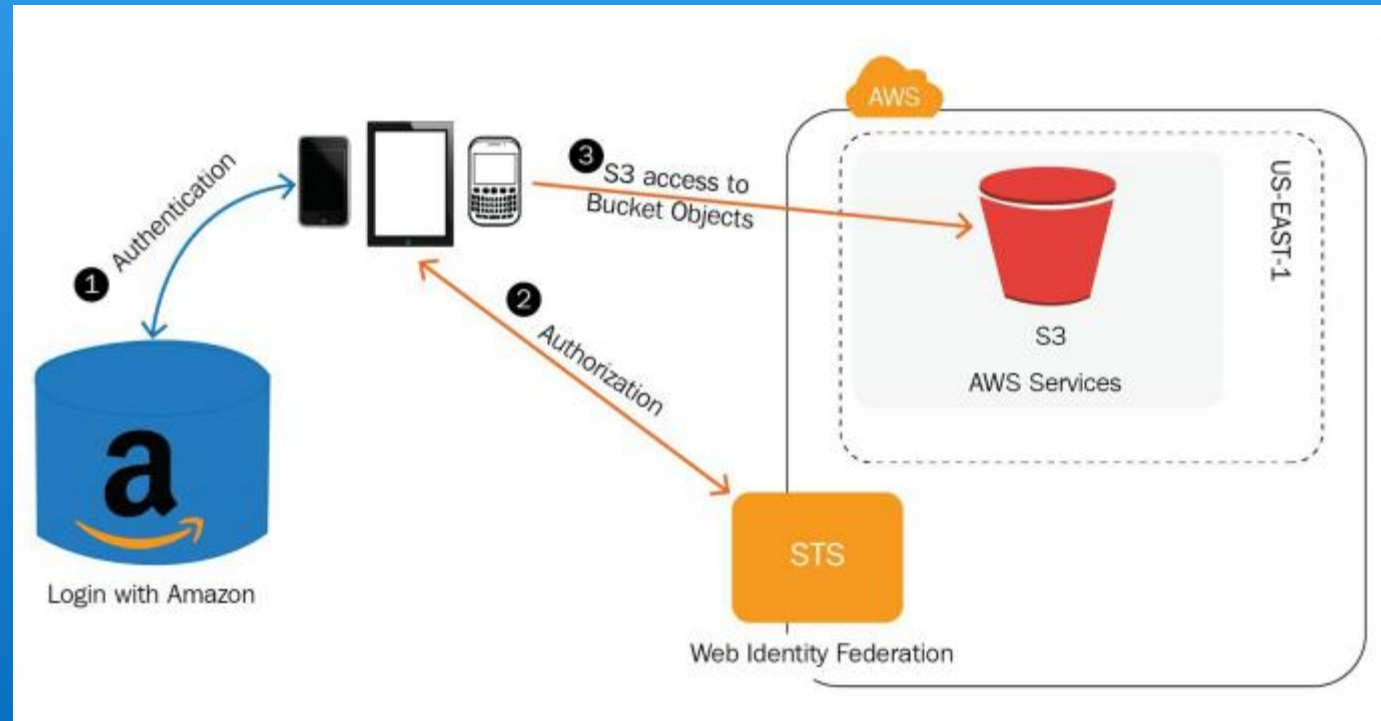
## 12. Overview of Web Identity Federation

While mobile app wants to access AWS resource then, authentication using access key and secret key is not recommended due to security aspects.

We have option called – web identity federation

Application can request temporary security credentials dynamically when required from STS

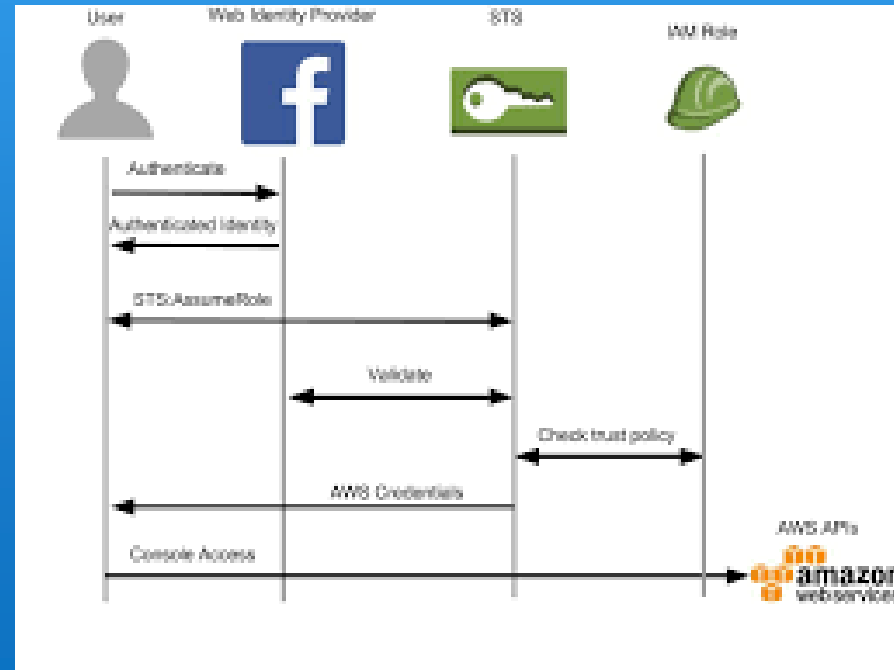
Here, no need to create own custom sign-in code, you can login through Facebook, Amazon, Google etc.



# 13. Overview of Security Token Services(STS)

It is global web service which enables an application to dynamically generate temporary security credentials either for federated users or IAM users.

URL for global service STS – <https://sts.amazonaws.com/>



# 14. AWS CLI commands respect to IAM

CLI is a unified tool to manager AWS resources.

First you need to download CLI in your system from :

<https://docs.aws.amazon.com/cli/latest/userguide/awscli-install-windows.html>

Install CLI by double clicking on that package.

Once installed, open command prompt and perform following steps to configure CLI with your AWS account:

```
C:\Users\Rajan>aws configure
```

```
Root user credentials:<Enter root user password>
```

```
AWS Access Key ID [None]: <Enter your access key id>
```

```
AWS Secret Access Key [None]: <Enter your secret access key>
```

```
Default region name [None]: us-east-1
```

```
Default output format [None]: json
```

Now, you have successfully configured CLI in your systems, you can perform CLI commands now.

## 14. AWS CLI commands respect to IAM(Conti...)

### CLI commands general syntax:

*aws <aws service name> <service specific command> <parameters>*

aws help

aws iam help

aws iam create-user --user-name testadmin

aws iam create-group --group-name julybatch

aws iam delete-user --user-name testadmin

aws iam delete-login-profile --user-name testadmin

aws iam detach-user-policy --user-name Bob --policy-arn arn:aws:iam::123456789012:policy/TesterPolicy

CLI commands for all AWS services can be found at:

<https://docs.aws.amazon.com/cli/latest/reference/>

# Summary

In this chapter, we have gone through following topics

- Basics of Authentication and authorization
- Introduction of IAM
- Elements of IAM
- Creating and managing Users
- Creating and managing Groups
- Creating and managing Role
- Policy and policy classification
- Managed policy v/s resource based policy
- Few terminologies regarding policy
- Example of customer managed policy
- Example of resource based policy
- Overview of Active Directory Federation Services(ADFS)
- Overview of Web Identity Federation
- Overview of Security Token Service(STS)
- AWS CLI command respect to IAM



See you soon...

*Thank You!*