Class 23

IAM: Identity & Access Management

Is used to provide or restrict access to users or distribute access to users

We access “AWS account” management through IAM users

In real time we will have multiple accounts (e.g.: one account for billing management, one for legal team, one for IT team, multiple accounts for multiple projects, multiple accounts for multiple users…)

**Lab: -**

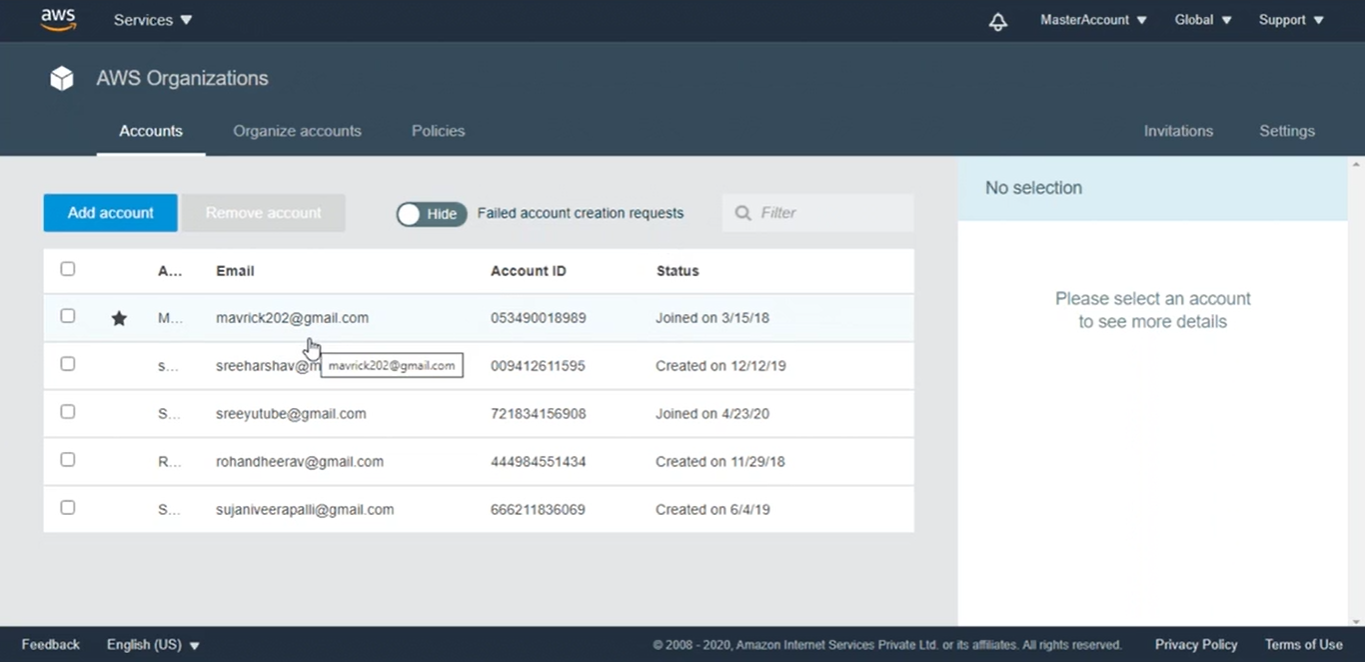
1, First, we go to Organizations, we have all AWS accounts here

Under “Services” dropdown

- Management & Governance

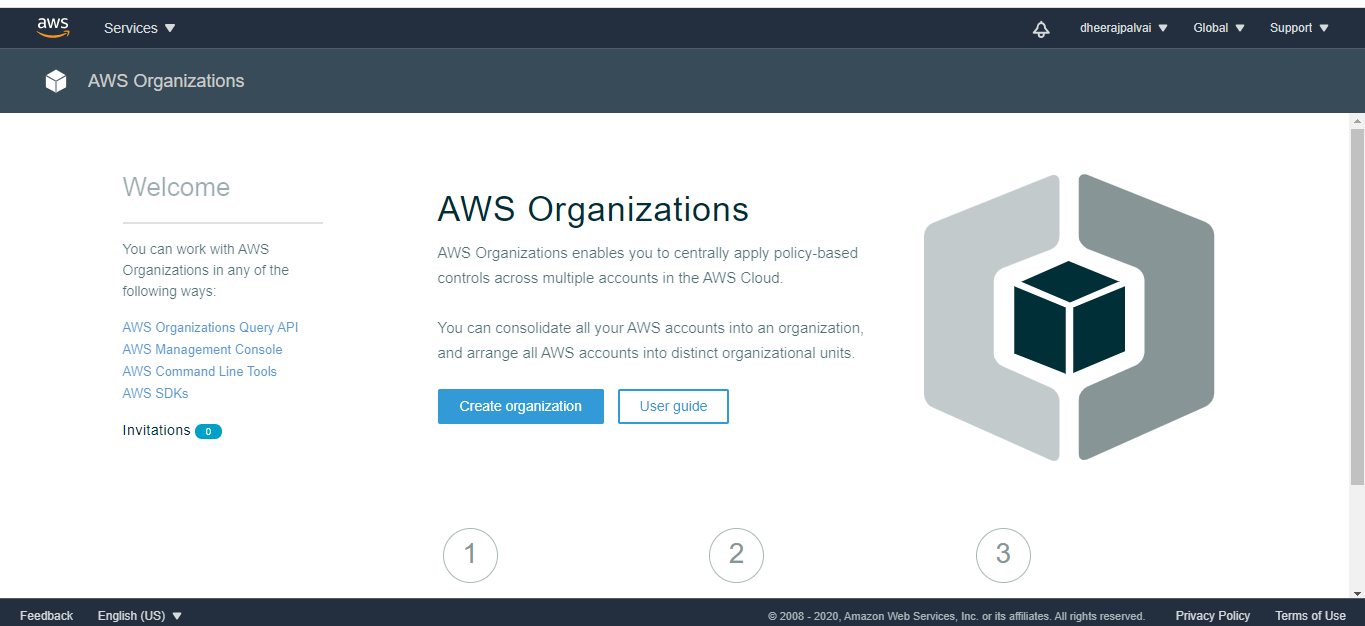
- AWS Organizations

[we have all AWS accounts here, if the account which we opened is main account we can see “Master account id & email-id”

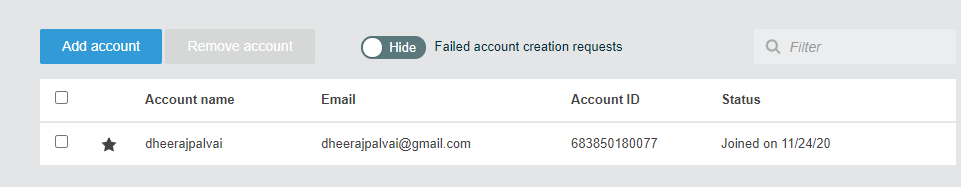


* If we open master or parent account, here we can see \* mark which is main account, and others are sub accounts
* In any organization they use multiple accounts, because if one account compromises also other account will work, they don’t keep all operations in one account.
* Master account or main account or Root account is used only for **bulling** purpose, we do not keep any servers here

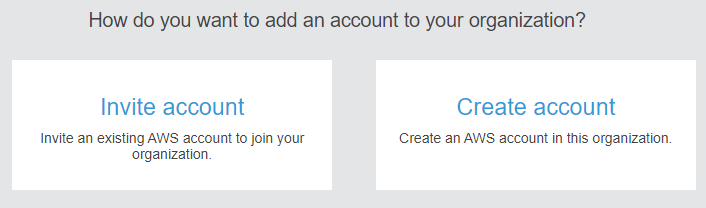
🡺Initially we get “create organization” as follows



Create organization -> Here we get our account with star mark, that is main account



[Add account](https://console.aws.amazon.com/organizations/home?region=us-east-1#/accounts/add) -> click on Add account -> we get Invite account & Create account



Invite account: To Invite account, an AWS account should be existing

Create account: To create account, we should have valid email-id of user, and after giving that created id to user access it by “resetting password”

**Interview Question**

**Q, how do you manage AWS environment?**

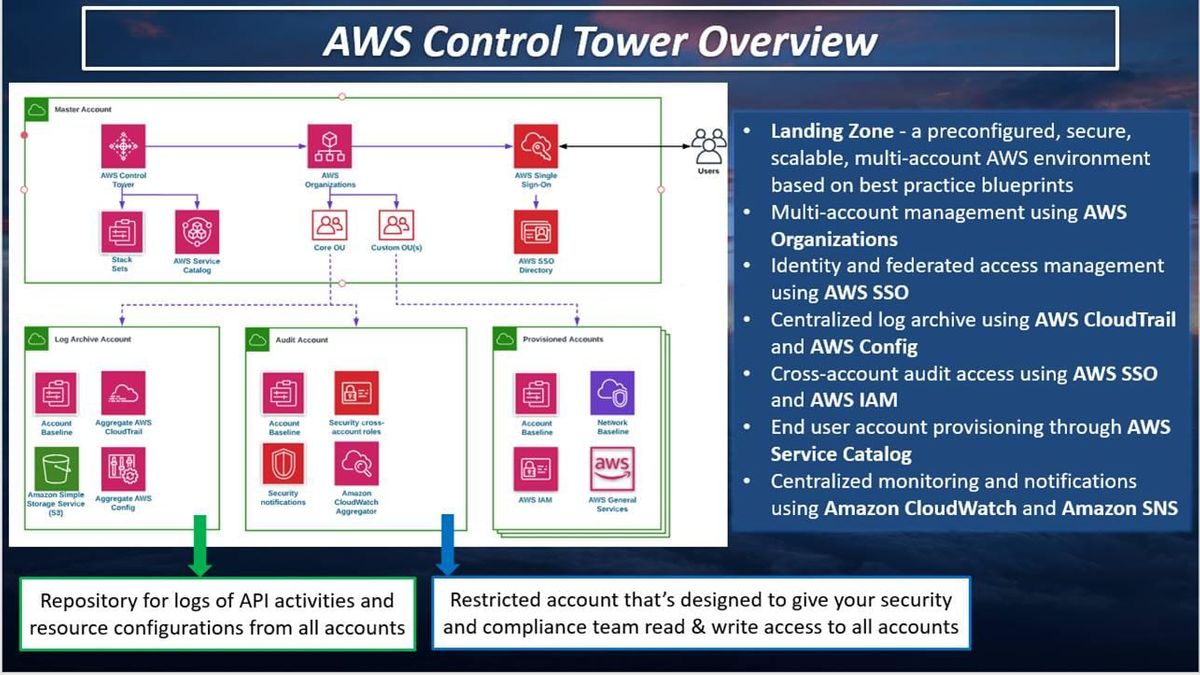
**A, we are running almost like 30 “AWS accounts” and it keep on increasing,**

**And we are using “AWS Organizations” to centrically manage whole infrastructure**

**and nobody has access to “Root or main account” (**[**dheerajpalvai@gmail.com**](mailto:dheerajpalvai@gmail.com)**)**

2, Instead of creating accounts and manage manually, we make automatic using “Control Tower”

Architecture:



* Under “Services”
* Management & Governance
* Control Tower

Logs:

Log Archive Account: logs contain

Who or which account opened and made the changes, who entered last time?

Audit Account: contains

Finance reports like data in banks, pharmacy company

**🡺 IAM**

Identity & access management, is used to restrict or provide access to users or accounts

* Users
* Groups
* Roles
* Policies

1, Users:

They might be humans **or** applications **or** AWS accounts

If we go to:

IAM -> Users -> Add user (Here we can see TWO Access types)

i, Programmatic access: Enables “access key” and “secret key” for the AWS API, CLI, SDK, and other development tools.

Is used for API (Application Programming Interface like python, php, java), CLI (command line interface like putty, command prompt), SDK (Software Development Kit like softwires package) by enabling keys and accessing them

ii, AWS Management Console access: Enables “password” that allows users to sign-in to the AWS Management Console

Is used for “AWS console” by enabling password and accessing it.

“access key” and “secret key” = “password”

**Lab: -**

1, IAM -> Users -> Add user -> Username: testuser1 **&** enable “Programmatic access” ” AWS Management Console access” **&** Console password: Custom password(06[61A0542) ->Finish

* **We login aws management console using “ID and Password” – this is for users**
* **We also create and use “Access key ID & Secret access key” – this is for users/apps**
* Now we get “Access key ID & Secret access key”

Access key ID: AKIAZ6OFFUXWZF3TYVGD

Secret access key: 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

i, Programmatic access:

now if I want to use by using “**Programmatic access**” Download “**AWS CLI**” and install

Open windows command prompt: -

**CMD: aws configure**

**AWS Access Key ID [None]: AKIAZ6OFFUXWZF3TYVGD** (copy generated Access key ID)

**AWS Secret Access Key [None]: 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** (copy generated Secret access key)

**Default region name [None]: us-east-1** (give region US East (N. Virginia) us-east-1) **or** press **Enter**

**Default output format [None]: json** (give json (JavaScript Object Notation)) **or** press **Enter**

**CMD: aws ec2 describe-vpcs** (though we gave “Access key ID & Secret access key” we get error, we must give permissions)

* So, we have to give access to **users** (or) keep all users in **group** and give access to group
* **For giving permission to User**: IAM -> Users -> testuser1 -> permissions -> Add permissions -> Attach existing policies directly -> “AmazonEC2ReadOnlyAccess” (for example)
* **For giving permission to Groups**: IAM -> Groups -> Create New Group -> Group Name: "mygroup" -> "AmazonEC2ReadOnlyAccess"(for example)

**Add Users to group**: IAM -> Groups -> “mygroup"(select) and go to "Group Actions" and select "Add Users to Group" and add testuser1 & testuser2(create testuser2 like 1)

**CMD: aws ec2 describe-vpcs** (now it shows all vpc information, because we added permissions)

* Now if we delete group also, we still get access because permissions are added also to users

If we add permission of “AWS s3Readonly Access” to group and

**CMD: aws s3 ls** (we get access)

ii, AWS Management Console access:

now if I want to use by using “**AWS Management Console**” go to

**IAM -> Dashboard -> Sign-in URL for IAM users in this account**

**https://683850180077.signin.aws.amazon.com/console (copy this URL and use in browser) and give created username and password**

IAM user name: testuser1

Password: 06[61A0542

Note: this user has no permissions to access initially

Same as above go to main account and add permissions to testuser1 or group, then it gets access

2, Groups:

Create groups and add users into that group, and give access permissions

**Lab: -**

IAM -> Groups -> Create New Group -> Group Name: "mygroup" -> "AmazonEC2ReadOnlyAccess"(for example)

**Add Users to group**: IAM -> Groups -> “mygroup"(select) and go to "Group Actions" and select "Add Users to Group" and add testuser1 & testuser2(create testuser2 like 1)

3, Roles:

Roles are not for users, those are only for aws services

* Aws Role allows to access, ‘one aws service’ to ‘another aws service’ (s3 access ec2)

**Lab: -**

1, IAM -> Roles -> Create role -> AWS service -> EC2 (AWS services) -> AmazonS3ReadOnlyAccess -> Role name: s3\_role

* By default ec2 instance have “Aws CLI”

2, ec2 instance -> putty

**Aws --version** (to know version of aws)

**Aws s3 ls** (error, because we didn’t give any iam role)

To give Iam role to ec2 instance, goto

2, Ec2 instance -> RC -> Security -> Modify IAM role -> choose IAM role ‘s3\_role’

**Aws s3 ls** (now s3 list comes)

* If I want to get ec2 Vpcs list

1, IAM -> Roles -> s3\_role -> Permissions -> Attach policies -> "AmazonEC2ReadOnlyAccess"

Putty -> **aws ec2 describe-vpcs** (since we added ec2read only policy to role and role is attached to ec2 we get, we have to give region. For that)

Putty -> **Aws configure** (we don’t give“Access key ID & Secret access key” just press enter but only gives Region to get ec2 VPCs information)

* For accessing **Users**, we use “Access key ID & Secret access key” and get VPCs information
* In **Roles** also we get VPCs information, but we don’t give any “Access key ID & Secret access key” here,

roles also accessed by “Access key ID & Secret access key” in backend, when we create IAM Role, aws creates “Access key ID & Secret access key” and gives to role,

* we can see that aws created “Access key ID & Secret access key”

Putty -> **curl** [**http://169.254.169.254/latest/meta-data/iam/security-credentials/s3\_role**](http://169.254.169.254/latest/meta-data/iam/security-credentials/s3_role)(we can see that aws created “Access key ID & Secret access key”)

Workflow: **[ec2**(IAM role) **– IAM role** (s3readonly, ec2readonly) **-]**