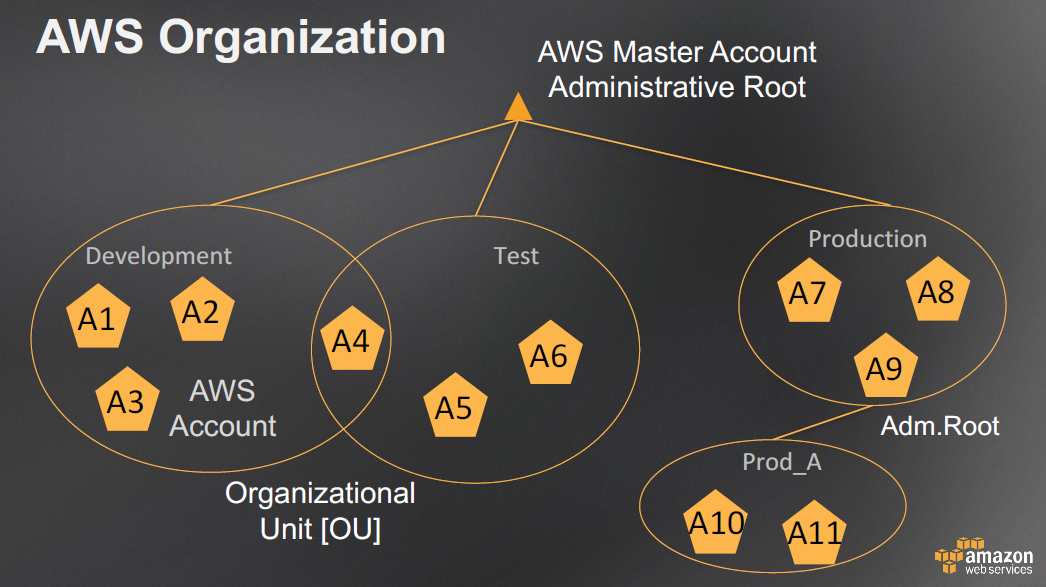
AWS Organization Implementation Steps

**AWS Organization Structure:**



As part of standard AWS Best Practices for managing multiple accounts and for product deployments, the team at AWS has recommended to use AWS Organization Units along with multiple accounts which are possible with AWS Organizations.

**Why we need AWS Organizational Unit (OU) along with Organization?**

* AWS OU is an integral part of the Organization
* As per the guidelines it is best practice to use AWS OU while implementing Organization
* When you want to enable AWS multi-cloud environment OU is essential as it is providing mechanisms to do so in a secure, scalable, and resilient manner
* It also reduces the operational overhead of managing the structure.
* Service Control Policies (SCPs) are primarily applied at the OU level to make it simple for management
* OUs are to be created based on the Business Units and even service policy roles are being created by keeping the Business Units in mind.
* OUs will simplify policy management and any potential troubleshooting.

**What is AWS Organization unit (OU)?**

* You can use organizational units (OUs) to group accounts together to administer as a single unit. In general, applying policies at the OU-level is a better practice.
* You can use organizational units (OUs) to group accounts together to administer as a single unit.
* You can attach a policy-based control to an OU, and all accounts within the OU automatically inherit the policy.
* You can create multiple OUs within a single organization, and you can create OUs within other OUs. Each OU can contain multiple accounts,
* You can move accounts from one OU to another. However, OU names must be unique within a parent OU or root.

**Standard Details of having Organizational Units (OUs):**

* Multiple members account will be further linked to one or more Organization Units as mentioned below.
* Primarily we are planning to have 3 Organization Units (OU)
* Development / Staging, UAT and Production and in the below level the members account will be distributed based on the nature of the account.
* AWS recommends that you start with security and infrastructure in mind which is covered in **Foundational OUs**
* There are several types of OU segregation based on the need of Organization as below:

**Foundational OUs**

OU: Infrastructure (Used for shared infrastructure services, such as networking and IT services)

OU: Security (hosting security-related access and services)

**Additional OUs**

OU: Sandbox (individual technologists to learn AWS services)

OU: Workloads (for software lifecycles to be created)

OU: PolicyStaging (verify results of applying a policy)

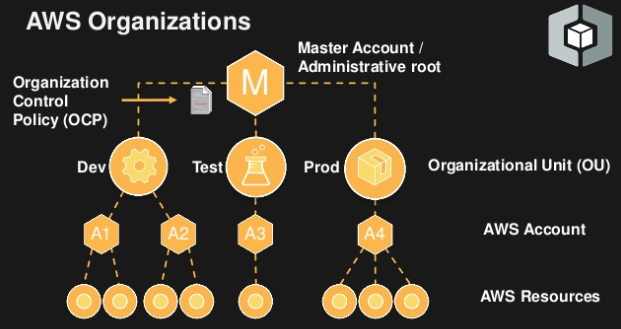
OU: Suspended (closed account and are waiting to be deleted from the organization)

OU: Exceptions (the accounts with given a customized security stance)

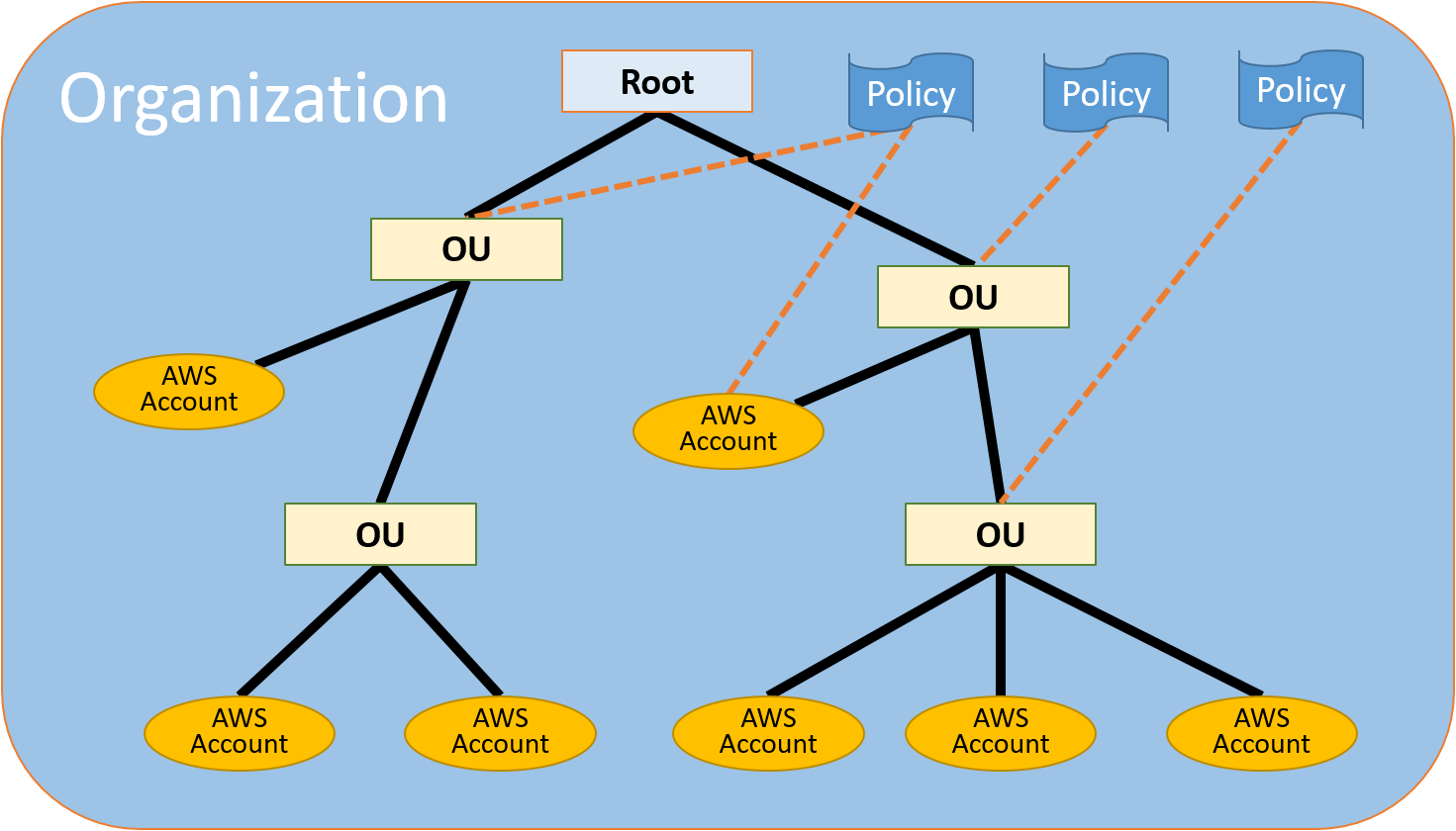
OU: Deployments (for CI/CD pipeline deployments)

**Conclusion for AWS OUs:**

A well-architected multi-account strategy along with Organizational Unit helps you innovate faster in AWS, while helping you meet your security and scalability needs. The points described in this presentation represent AWS best practices that you should use as a starting point for your AWS journey.



* Above is the hierarchical representation of AWS Organization along with OUs
* In this , we are considering overall 3-4 Organizational Units
* Here in this representation we can avoid creating instances for only front-end applications & directly using AWS Cloud front with CDN features.



* There are two types of accounts in an organization: a single account that is designated as the master account, and member accounts.

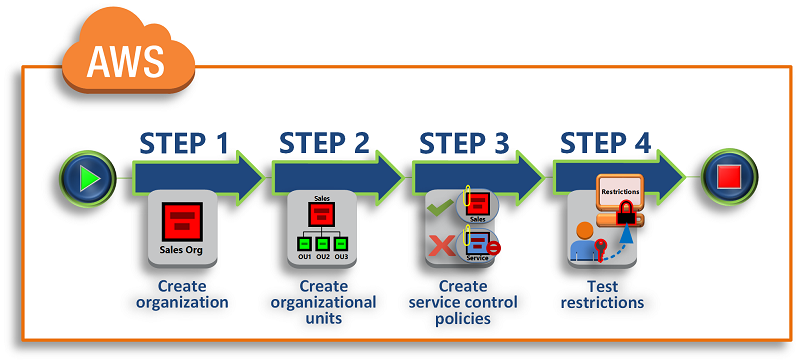
The master account is the account that creates the organization. From the organization's master account, you can follow below steps:

* Create accounts in the organization
* Invite other existing accounts to the organization
* Remove accounts from the organization
* Manage invitations
* Apply policies to entities (roots, OUs, or accounts) within the organization

**Invitation for member accounts:**

* An invitation can be issued only by the organization's master account. The invitation is extended to either the account ID or the email address that is associated with the member account. After the invited account accepts an invitation, it becomes a member account in the organization.

**Start to end illustration of AWS Organizations:**



* **Pre-requisites to implement the AWS Organizations:**

- There is already one AWS member accounts right now for emxcel solutions with ID = 1234567890

- We need to create **8** more accounts similar to start with like we created above mentioned AWS account.

- The break-up details of those 8 accounts are as follow:

1) Root account used for **consolidated billing** from which AWS Org will be enabled.

E.g. account ID = 1111 & mail ID = aws1111@emxcelsolutions.com

2) **Common platform services** account for Jenkins, Sonar etc (members account)

E.g. account ID = 2222 & mail ID = aws2222@emxcelsolutions.com

3) Account for all product **development** workload (members account)

E.g. account ID = 3333 & mail ID = aws3333@emxcelsolutions.com

4) Account for all product **staging** workload (members account)

E.g. account ID = 4444 & mail ID = aws4444@emxcelsolutions.com

5) Account for all product **UAT** workload (members account)

E.g. account ID = 5555 & mail ID = aws5555@emxcelsolutions.com

6) Account for separate **first product** Production workload (members account)

E.g. account ID = 6666 & mail ID = aws6666@emxcelsolutions.com

7) Account for separate **second product** Production workload (members account)

E.g. account ID = 7777 & mail ID = aws7777@emxcelsolutions.com

8) Account for separate **third product** Production workload (members account)

E.g. account ID = 8888 & mail ID = aws8888@emxcelsolutions.com

- From master account will send invite to all the member accounts mentioned above to join the AWS Organization having account ID = 1111

- Once the member accounts created and the invite sent to the registered email ID, can receive the invite from email ID and join the Organization hierarchy.

- After this activity, one master account having Organization and all the other member accounts associated with it.

* **Traditional way of Creating & managing AWS Organization:**

**Step01: Create your organization**

Sign in to account 1111 as an administrator, create an organization with that account as the master account, and invite an existing account 2222, to join as a member account.

- Sign in to AWS as an administrator of account 1111 and open the AWS Organizations console at https://console.aws.amazon.com/organizations/.

- On the introduction page, choose Create organization.

- In the Create organization confirmation dialog box, choose Create organization. On the Accounts tab, the star next to the account email indicates that it's the master account.

- A verification email is automatically sent to the address that is associated with your master account. Verify your email address within 24 hours.

You now have an organization with your account as its only member. This is the master account of the organization.

**Step02: Invite an existing account to join your organization:-**

- Open the Organizations console at https://console.aws.amazon.com/organizations/.

- Choose the Accounts tab. Now you can invite other accounts to join as member accounts

- On the Accounts tab, choose Add account and then choose Invite account.

- In the Account ID or email box, enter the email address of the owner of the account that you want to invite, like: aws2222@emxcelsolutions.com.

- Choose Invite. AWS Organizations sends the invitation to the account owner via mail

- Open the email that AWS sent from the master account and choose the link to accept the invitation. When it prompted to sign in, do so as an administrator in the invited member account.

- Open the AWS Organizations console (https://console.aws.amazon.com/organizations/) and sign in as an administrator of the member account. Choose Invitations. The number beside the link indicates how many invitations this account has.

- On the Invitations page, choose Accept and then choose Confirm.

**Step03: Create a member account to for your organization:-**

- On the AWS Organizations console, on the Accounts tab, choose Add account.

- For Full name, enter a name for the account, such as QA/UAT Account.

- For Email, enter the email address of the individual who is to receive communications on behalf of the account. This value must be globally unique. No two accounts can have the same email address.

- Choose Create. You might need to wait a short while and refresh the page to see the new account appear on the Accounts tab.

**Step 4: Create the organizational units**

To create and populate the OUs:

- On the AWS Organizations console, choose the Organize Accounts tab and then choose + New organizational unit.

- For the name of the OU, enter name you want to select and then choose Create organizational unit.

- Choose your newly created OU to navigate into it and then choose + new organizational unit.

- For the name of the second OU, enter other name and then choose Create organizational unit.

**Step 5: Create the service control policies**

- select the policies for Organization & accounts in such a way that it will have granular level of access for the list of users in a simple manner.

- Create policy based on the access roles given to the users

- OUs are to be created based on the Business Units and even service policy roles are being created by considering the Business Units (development team, QA Team, DBA team, Devops Team, Technical Architect / Leads)

- Primarily there will be 4 levels of access roles to be considered in Organization across all accounts

- Details of the types of access roles managed using Organization mentioned below

**Developers** - very basic and limited access to be provided with only required execution rights

**Architects** / leads - limited read-only & execution access on the required resources

**DBA** - limited execution and read/write access only for database related resources

**DevOps** - administrator level access having full access of Read/Write on almost all resources

We also need to enable 2FA for each and every user to login to the console of AWS resources to make sure of secure authorization while doing access control.

Service policies will be designed to strengthen the security and managing operations easily.