AWS Organizations

# AWS Organizations

* AWS Organizations is an account management service that enables consolidating multiple AWS accounts into an organization that can be created and centrally managed.
* AWS Organizations includes [consolidated billing](https://jayendrapatil.com/aws-consolidated-billing/) and account management capabilities that enable one to better meet the budgetary, security, and compliance needs of your business.
* As an administrator of an organization, new accounts can be created in an organization and invite existing accounts to join the organization.
* AWS Organizations enables you to
  + Centrally manage policies across multiple AWS accounts
  + Control access to AWS services
  + Automate AWS account creation and management
  + Consolidate billing across multiple AWS accounts

## AWS Organizations

## AWS Organization Features

### Centralized management of all of your AWS accounts

* Combine existing accounts into or create new ones within an organization that enables them to be managed centrally
* Policies can be attached to accounts that affect some or all of the accounts

### Consolidated billing for all member accounts

* [Consolidated billing](https://jayendrapatil.com/aws-consolidated-billing/) is a feature of AWS Organizations.
* Master account of the organization can be used to consolidate and pay for all member accounts.

### Hierarchical grouping of accounts to meet budgetary, security, or compliance needs

* Accounts can be grouped into organizational units (OUs) and each OU can be attached different access policies.
* OUs can also be nested to a depth of five levels, providing flexibility in how you structure your account groups.

### Control over AWS services and API actions that each account can access

* As an administrator of the master account of an organization, access to users and roles in each member account can be restricted to which AWS services and individual API actions
* **Organization permissions overrule account permissions.**
* This restriction even overrides the administrators of member accounts in the organization.
* When AWS Organizations blocks access to a service or API action for a member account, a user or role in that account can’t access any prohibited service or API action, even if an administrator of a member account explicitly grants such permissions in an IAM policy.

### Integration and support for AWS IAM

* IAM provides granular control over users and roles in individual accounts.
* AWS Organizations expands that control to account level by giving control over what users and roles in an account or a group of accounts can do
* User can access only what is allowed by both the AWS Organizations policies and IAM policies.
* Resulting permissions are the logical intersection of what is allowed by AWS Organizations at the account level, and what permissions are explicitly granted by IAM at the user or role level within that account.
* If either blocks an operation, the user can’t access that operation.

Timeline

Description automatically generated

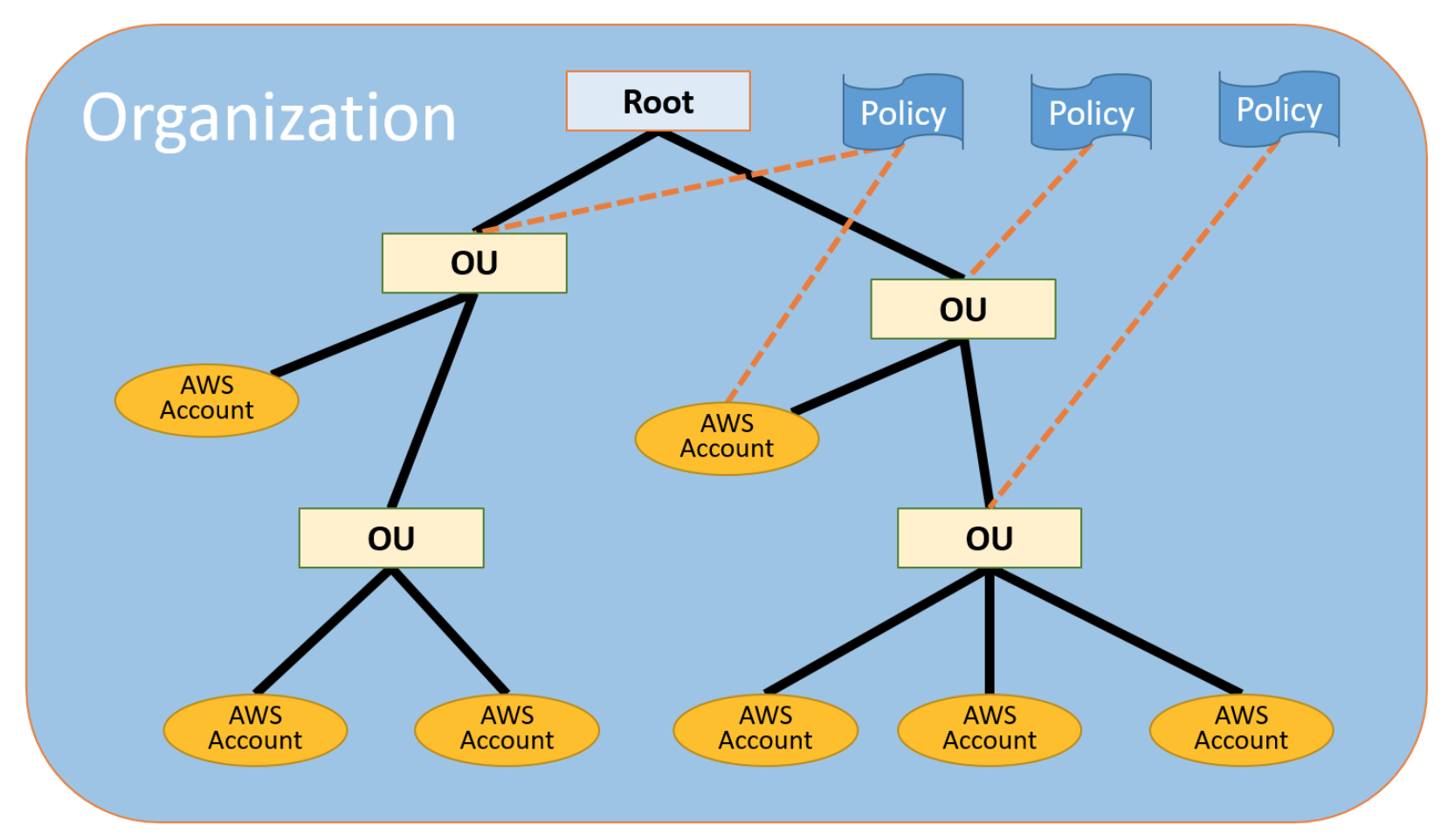
### Integration with other AWS services

* Select AWS services can be enabled to access accounts in the organization and perform actions on the resources in the accounts.
* When another service is configured and authorized to access with the organization, AWS Organizations creates an **IAM service-linked role** for that service in each member account.
* Service-linked role has predefined IAM permissions that allow the other AWS service to perform specific tasks in the organization and its accounts
* All accounts in an organization automatically have a service-linked role created, which enables the AWS Organizations service to create the service-linked roles required by AWS services for which you enable trusted access
* These additional service-linked roles come with policies that enable the specified service to perform only those required tasks

### Data replication that is eventually consistent

* AWS Organizations is **eventually consistent**.
* AWS Organizations achieves high availability by replicating data across multiple servers in AWS data centers within its region.
* If a request to change some data is successful, the change is committed and safely stored.
* However, the change must then be replicated across the multiple servers.

## AWS Organizations Terminology and Concepts



### Organization

* An entity created to consolidate AWS accounts.
* An organization has one master account along with zero or more member accounts.
* An organization has the functionality that is determined by the feature set that you enable i.e. **All features** or **Consolidated Billing only**

### Root

* Parent container for all the accounts for the organization.
* Policy applied to the root is applied to all the organizational units (OUs) and accounts in the organization.
* Root account itself is excluded from SCP permissions.
* There can be only one root currently and AWS Organization automatically creates it when an organization is created

### Organization unit (OU)

* A container for accounts within a root.
* An OU also can contain other OUs, enabling hierarchy creation that resembles an upside-down tree, with a root at the top and branches of OUs that reach down, ending in accounts that are the leaves of the tree.
* A policy attached to one of the nodes in the hierarchy, flows down and affects all branches (OUs) and leaves (accounts) beneath it
* **An OU can have exactly one parent, and currently each account can be a member of exactly one OU.**

### Account

* A standard AWS account that contains AWS resources.
* Each account can be directly in the root, or placed in one of the OUs in the hierarchy.
* **Policy can be attached to an account to apply controls to only that one account.**
* Accounts can be organized in a hierarchical, tree-like structure with a root at the top and organizational units nested under the root.
* Master account
  + Primary account which creates the organization
  + can create new accounts in the organization, invite existing accounts, remove accounts, manage invitations, apply policies to entities within the organization.
  + has the responsibilities of a payer account and is responsible for paying all charges that are accrued by the member accounts.
* Member account
  + Rest of the accounts within the organization are member accounts.
  + An account can be a member of only one organization at a time.

### Invitation

* Process of asking another account to join an organization.
* An invitation can be issued only by the organization’s master account and is extended to either the account ID or the email address that is associated with the invited account.
* Invited account becomes a member account in the organization, after it accepts the invitation.
* Invitations can be sent to existing member accounts as well, to approve the change from supporting only consolidated billing feature to supporting all features
* Invitations work by accounts exchanging handshakes.

### Handshake

* A multi-step process of exchanging information between two parties
* Primary use in AWS Organizations is to serve as the underlying implementation for invitations.
* Handshake messages are passed between and responded to by the handshake initiator (master account) and the recipient (member account) in such a way that it ensures that both parties always know what the current status is.

### Available feature sets

#### **Consolidated billing**

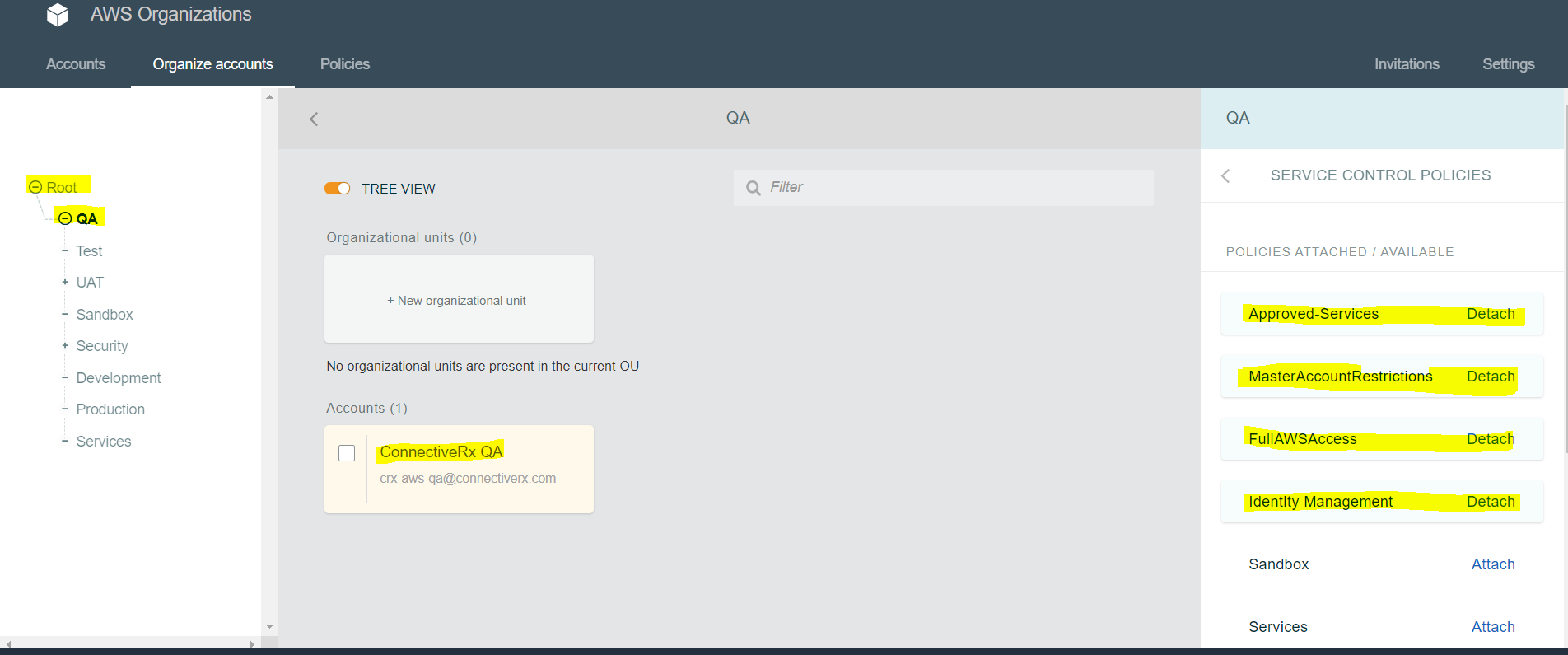
* provides shared billing functionality

#### **All features**

* includes all the functionality of consolidated billing,
* includes advanced features that gives more control over accounts in the organization.
* allows master account to have full control over what member accounts can do
* master account can apply SCPs to restrict the services and actions that users (including the root user) and roles in an account can access, and it can prevent member accounts from leaving the organization.

### Service control policy (SCP)

* Service control policy specifies the services and actions that users and roles can use in the accounts that the SCP affects.
* SCPs are similar to IAM permission policies except that they don’t grant any permissions. Rather they are used to restrict permission given by FullAWSAccess policy.
* SCPs are filters that allow only the specified services and actions to be used in affected accounts.
* **SCPs override IAM permission policy. So even if a user is granted full administrator permissions with an IAM permission policy, any access that is not explicitly allowed or that is explicitly denied by the SCPs affecting that account is blocked.** For e.g., if you assign an SCP that allows only database service access to your “database” account, then any user, group, or role in that account is denied access to any other service’s operations.
* **SCP can be attached to**
  + **A root, which affects all accounts in the organization**
  + **An OU, which affects all accounts in that OU and all accounts in any OUs in that OU subtree**
  + **An individual account**
* **Master account of the organization is not affected by any SCPs that are attached either to it or to any root or OU the master account might be in.**



### Whitelisting vs. blacklisting

Whitelisting and blacklisting are complementary techniques used to apply SCPs to filter the permissions available to accounts.

Diagram

Description automatically generated

Graphical user interface, text, application, chat or text message

Description automatically generated

#### **Whitelisting(Allow Listing)**

* Explicitly specify the access that is allowed.
* All other access is implicitly blocked.
* **By default, all permissions are whitelisted.**
* For restricting permissions, replace the FullAWSAccess policy with one that allows only the more limited, desired set of permissions.
* Users and roles in the affected accounts can then exercise only that level of access, even if their IAM policies allow all actions.
* If you replace the default policy on the root, all accounts in the organization are affected by the restrictions.
* You can’t add them back at a lower level in the hierarchy because an SCP never grants permissions; it only filters them.

#### **Blacklisting (Deny Listing) (Default)**

* Default behavior of AWS Organizations.
* Explicitly specify the access that is not allowed.
* Explicit deny of a service action overrides any allow of that action.
* All other permissions are allowed unless explicitly blocked
* **By default, AWS Organizations attaches an AWS managed policy called FullAWSAccess to all roots, OUs, and accounts. This allows any account to access any service or operation with no AWS Organizations–imposed restrictions.**
* **With blacklisting, additional policies are attached that explicitly deny access to the unwanted services and actions**

Graphical user interface, application

Description automatically generated

## ****AWS Certification Exam Practice Questions****

1. An organization that is currently using consolidated billing has recently acquired another company that already has a number of AWS accounts. How could an Administrator ensure that all AWS accounts, from both the existing company and the acquired company, are billed to a single account?
   1. Merge the two companies, AWS accounts by going to the AWS console and selecting the “Merge accounts” option.
   2. **Invite the acquired company’s AWS account to join the existing company’s organization using AWS Organizations.**
   3. Migrate all AWS resources from the acquired company’s AWS account to the master payer account of the existing company.
   4. Create a new AWS account and set it up as the master payer. Move the AWS resources from both the existing and acquired companies’ AWS accounts to the new account.
2. Which of the following are the benefits of AWS Organizations? Choose the 2 correct answers:
   1. **Centrally manage access polices across multiple AWS accounts.**
   2. **Automate AWS account creation and management.**
   3. Analyze cost across all multiple AWS accounts.
   4. Provide technical help (by AWS) for issues in your AWS account.
3. A company has several departments with separate AWS accounts. Which feature would allow the company to enable consolidate billing?
   1. AWS Inspector
   2. AWS Shield
   3. **AWS Organizations**
   4. AWS Lightsail