AWS Multi-Account Landing Zone — Case Study / Architecture Guide / Tutorial

by **SATYAM UPADHYAY**

Linkedin GitHub LeetCode

Problem Statement

Modern companies use cloud platforms like AWS to deploy, manage, and scale their applications and data. As a business grows, the number of environments it needs to manage also increases, such as:

- **Development**
- **Testing / QA**
- **Staging**
- **Production**

Problem Description

When all environments and workloads are deployed within a single AWS account, it becomes difficult to enforce security controls, track costs, manage permissions, and maintain compliance. A single configuration mistake or security breach can impact the entire organization. As the number of teams and applications increases, a single-account model becomes unmanageable and risky.

Risk Section

Operating everything within a single AWS account introduces several risks:

1. **Security Risk** – No strong isolation between environments increases the blast radius.
2. **Operational Risk** – Misconfiguration or accidental changes can impact production.
3. **Compliance Risk** – Difficult to meet regulatory requirements such as SOC2, ISO, HIPAA, etc.
4. **Cost Risk** – Cloud spending cannot be clearly tracked per team or project.
5. **Monitoring Risk** – Logs and audit data are scattered and difficult to analyze.

Business Impact

These challenges lead to increased downtime risk, higher security exposure, lack of visibility into cloud spending, operational inefficiencies, audit failures, and poor environment separation. This directly impacts customer trust, delivery velocity, and business continuity.

Solution Overview

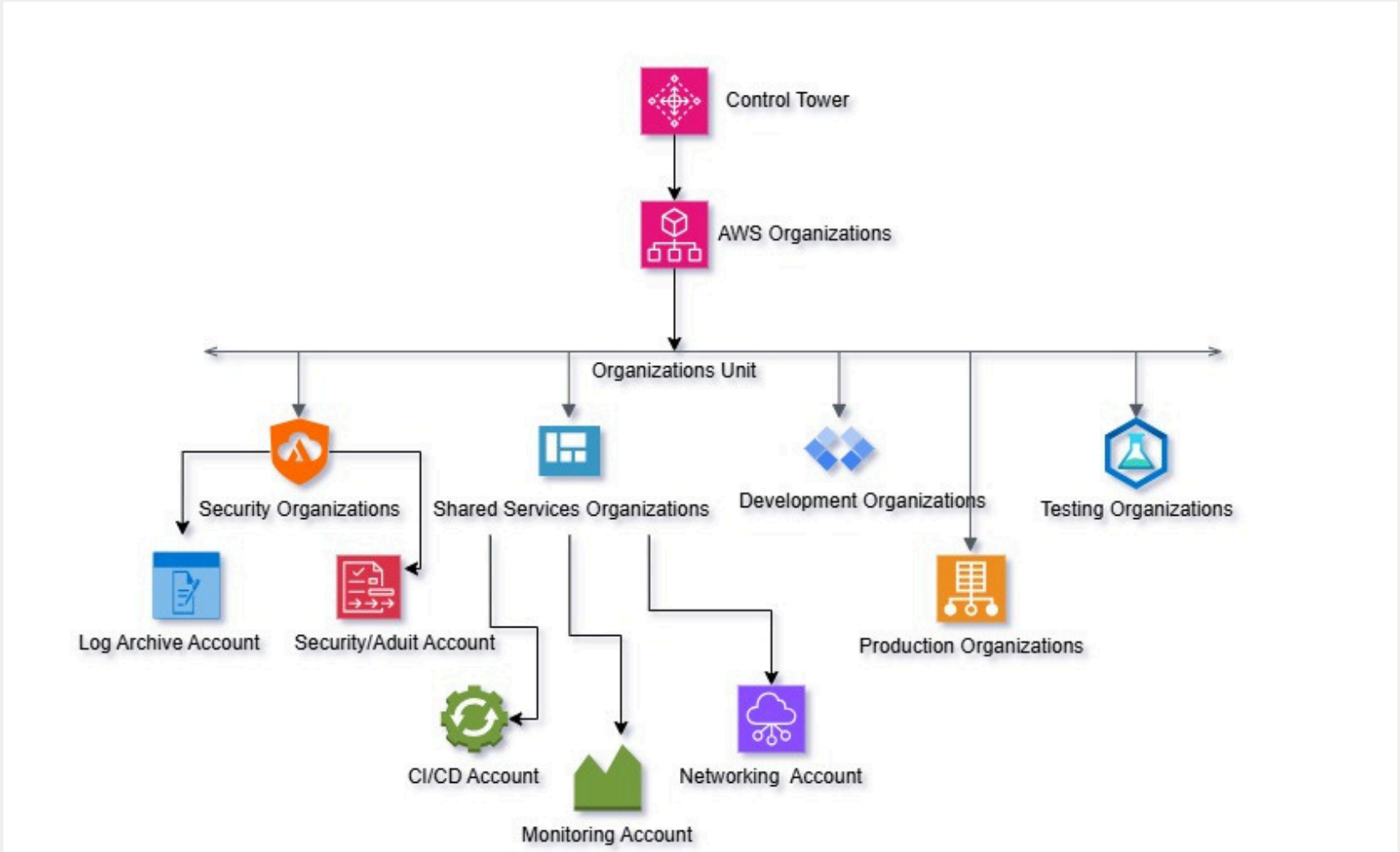
A landing zone is a secure, pre-configured AWS foundation that defines how multiple AWS accounts are created, governed, secured, and monitored. Instead of operating from one large AWS account, different workloads, environments, or teams are separated into dedicated AWS accounts while governance remains centralized. This provides strong isolation, consistent security controls, and better operational management.

Role of AWS Control Tower

AWS Control Tower is a managed service that helps organizations set up and govern a secure AWS landing zone. It automates account creation and applies standardized security baselines across all accounts. Key capabilities include:

- **Account Factory** for provisioning new AWS accounts with predefined security and configuration standards.
- **Guardrails** that enforce preventive and detective controls across accounts.
- **Centralized identity management** using AWS IAM Identity Center (SSO).
- **Centralized logging** for AWS Cloudtrail and AWS Config.
- **Security monitoring integration** through services such as AWS Security Hub and Amazon Guardduty.

“multi-Account architecture improves isolation, policy control, logging & security posture”



AWS Organizations

central services to manage multiple AWS account under one umbrella.

1.central billing

2.Account hierarchy

3.SCPs

Organizations Units

Group Account logically so each group can hv its own policies and controls.

Security Organizations

Stores all logs from all account in one tamper-proof place.

1.cloudtrail

2.AWS Config

3.Access logs

Security/Aduit Account

Dedicated Account for security tooling and monitoring.

1.GuardDuty 2.Security hub 3.IAM access

Networking Account

Central hub for networking.

1.VPC sharing

2.Transit Gateway

3.peering

4.DNS

Monitoring Account

Stores all monitoring and home for pipeline and alerting tools.

1.CloudWatch

2.Prometheus

3.Grafana

Ci/CD Account

automation tools.

1.Jenkins

2.Github Action

3. AWS codepipeline

Development Organizations

for Developers to build and Organizations

1.cost limits

2.lower restrictions

productions Organizations

Runs real customer workloads.

1.Maximum Security

2.Strict policies

3.Limited human access

testing Organizations

pre-production env for QA/UAT.

1.Stricter than dev

2.Mirrors prod

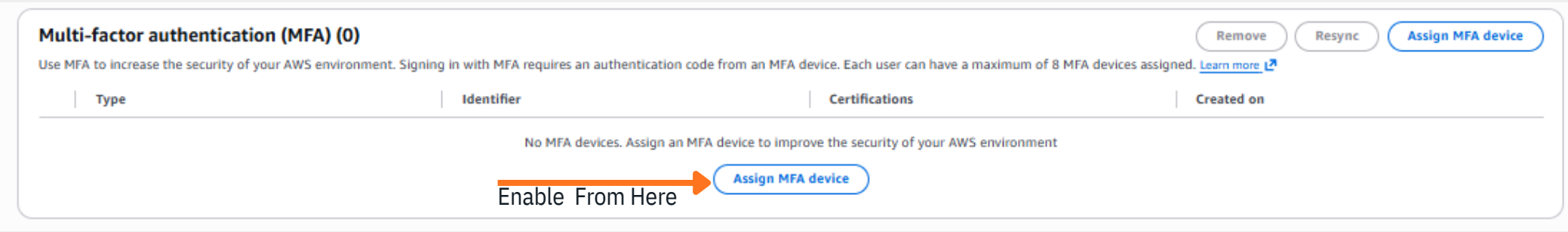
3.controlled Access

Tutorial: Set Up an AWS Multi-Account Landing Zone Using AWS Control Tower

Phase 0 — Pre-Setup & Foundation Controls

Step 1 — Secure the AWS Root Account

1. Sign in as the root user
2. Enable MFA



3. Console Path:

AWS Console → IAM → Security Credentials → Multi-Factor Authentication

4. Set a strong password
5. Store root credentials securely (offline)

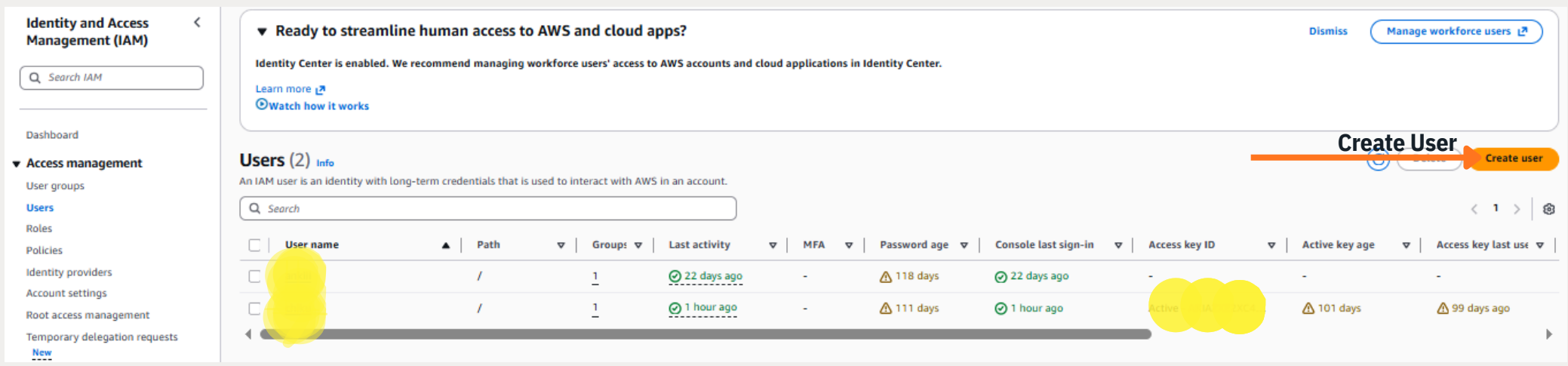
Do not use the root account for daily administration.

Step 2 — Create an IAM Administrator User

1.Sign in as root

Navigate to

IAM Console → Users/Group



2.Create a new user (example: admin-user)

3.Attach policy: AdministratorAccess

4.Enable console sign-in

5.Enable MFA for this user

6.Assign User to Admin Group

From this point, stop using the root account.

Step 3 — Configure Billing Alerts

This ensures you are notified if AWS charges increase.

Method 1: AWS Budgets

Go to:

Billing and cost Management Console → Budgets

Budgets

Create a budget

Set threshold (example: $10 / ₹800)

Add email alerts

Method 2: CloudWatch Billing Alerts

Optional but recommended.

Step 4 — Prepare Dedicated Emails for Sub-Accounts

security@domain.com

logs@domain.com

dev@domain.com

qa@domain.com

prod@domain.com

Now your AWS environment is safe to expand.

Phase 1 — Enable AWS Organizations

Step 1 — Enable AWS Organizations

AWS Organizations allows us to centrally manage multiple AWS accounts. We enabled it by navigating to AWS Organizations in the console and creating an Organization with “All Features Enabled.” We then created OUs to logically group our accounts.

Step 1: Log in as IAM Admin User

Use the admin-user you created.

Step 2: Open AWS Organizations

Go to

Services → AWS Organizations

Note Point : If you Create This First time Then Follow this Step Otherwise Your AWS Organization is already Create you can Only Add other Account and OU.

On the AWS Organizations page, click Create an organization

Select the feature set:

Choose Enable All Features (Recommended)

Click Create

Once created, AWS will automatically:

- Create a Root Organizational Unit (Root OU)
- Convert your current AWS account into the Management Account

Step 3: Create Organizational Units (Optional but Recommended)

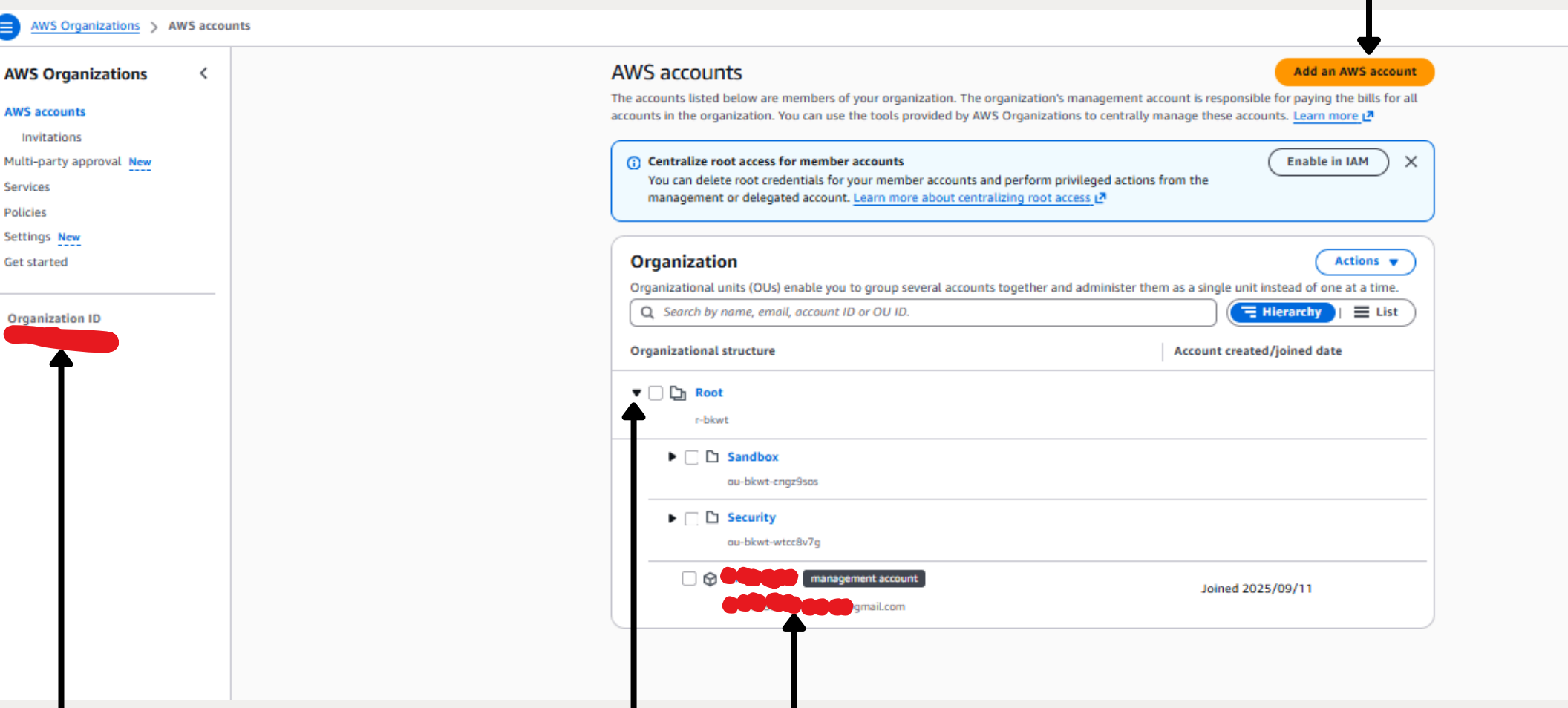
1. Go to AWS Organizations → AWS Accounts
2. Click Actions → Create new organizational unit (OU)
3. Enter an OU Name
4. Example:
    - Sandbox
    - Security
5. Click Create OU

Step 4: Add AWS Accounts (Optional)------->Below Page

You can either create a new AWS account or invite an existing one.

Create a New Member Account

1. Click Add an AWS account
2. Select Create an AWS account
3. Enter:
    - Account name
    - Email address
4. Select the OU where the account should be placed
5. Click Create AWS account



Step 5: Verify Organization

You should now see:

✓ Organization ID

✓ Management Account

✓ Root container

Phase 2 — Deploy AWS Control Tower (Landing Zone)

Step 1 : Enable and Deploy AWS Control Tower

What you will do

In this step, you will enable AWS Organizations and set up AWS Control Tower will automatically create a secure multi-account env with guardrails, loggiing , and ideantity mangement.

Prerequisites

Before starting , ensure:

- You have an **AWS root or admin user account**
- MFA is enabled on the root account (recommended)
- Region: AWS Control Tower supported region (for example, us-east-1)

To enable AWS Control Tower

Steps:1.Sign in to the AWS Management Console using an administrator account.

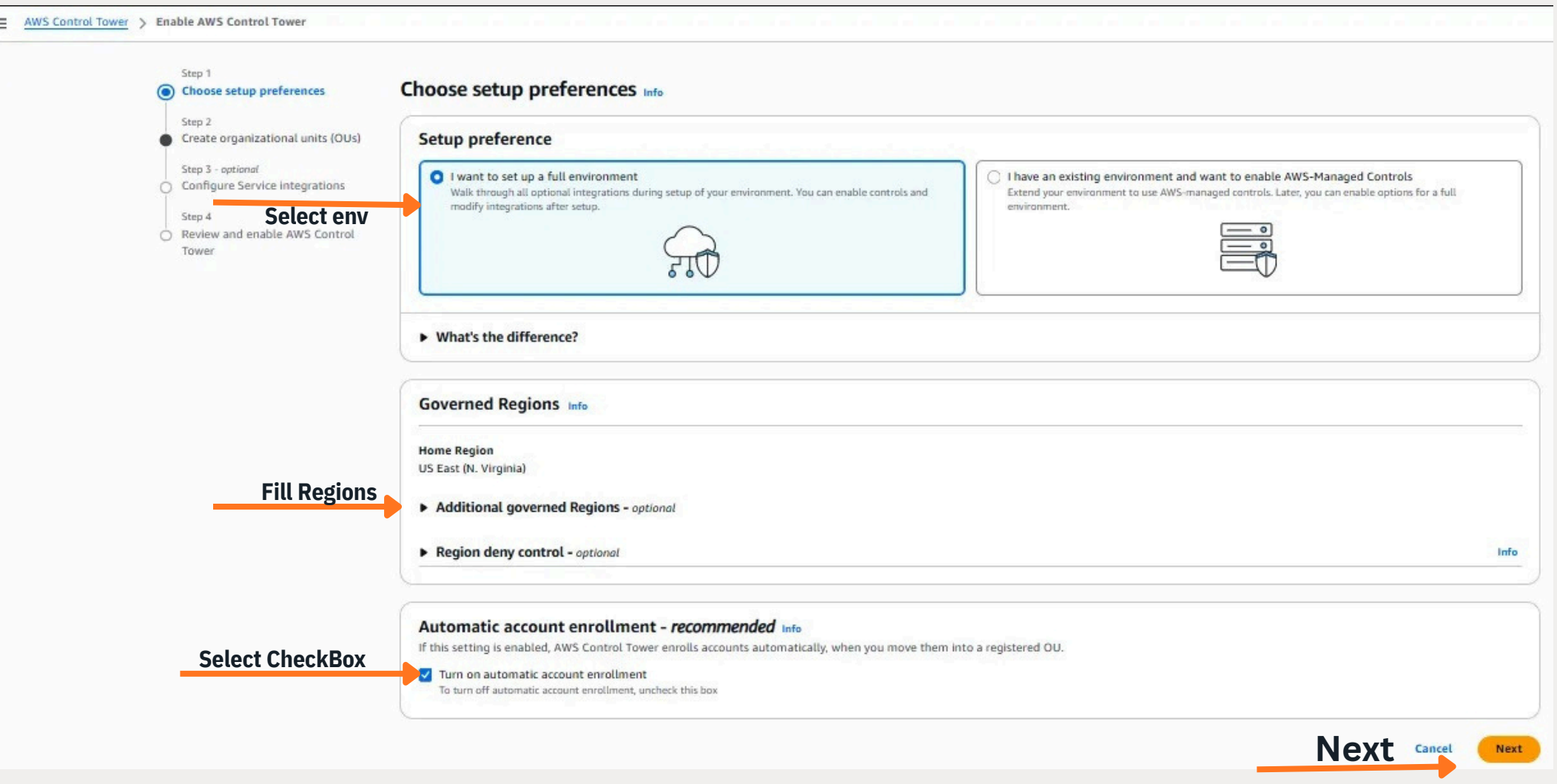
Steps:2.Open the AWS Control Tower console

Navigate to:

https://console.aws.amazon.com/controltower

Steps:3.Select the setup preference

In this step, we configure the initial setup preferences for AWS Control Tower. This defines how the landing zone will be created and which Regions and accounts will be governed.



Steps:1.Select the setup preference

Choose the option:

“Iwant to set up a full environment”

This option enables all recommended AWS Control Tower features and integrations during setup. You can modify

Steps:2.Select the governed Region

Under **Home Region**, select the AWS Region you want AWS Control Tower to govern

(for example: **US East (N. Virginia)**).

You may optionally add **additional governed Regions** if your workloads run across multiple Regions.

Steps:3.Configure Region deny control (Optional)

You can enable Region deny control to restrict resource creation in Regions that are not governed. This helps improvesecurity and cost control, but it is

Steps:4.Enable automatic account enrollment

Keep the option enabled:

“Turn on automatic account enrollment”

When enabled, AWS Control Tower will automatically enroll any new AWS account that is moved into a registeredOrganizational Unit (OU).

Steps:5.Proceed to the next step

Click **Next** to continue.

Steps:4. Create Organizational Units (OUs)

AWS Control Tower works with AWS Organizations to create a structured multi-account environment. In this step, Control Tower creates foundational

Organizational Units (OUs) and shared accounts.

Control Tower automatically creates the following recommended OUs:

Security OU

ThisOU contains the two shared security accounts:

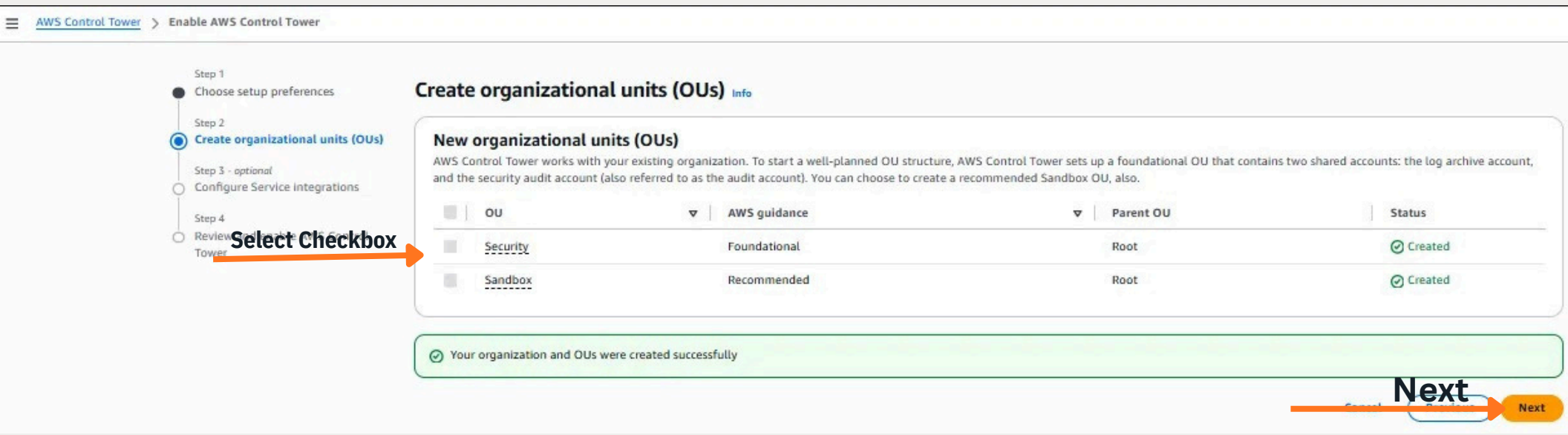
1.Log Archive Account

2.Security Audit Account

Theseaccounts store centralized logs and provide auditing capabilities across the landing zone.

Sandbox OU

This OU is optional and is used for development or experimentation workloads. It allows teams to test without impacting production environments.



After reviewing the OU structure, select **Next** to proceed.

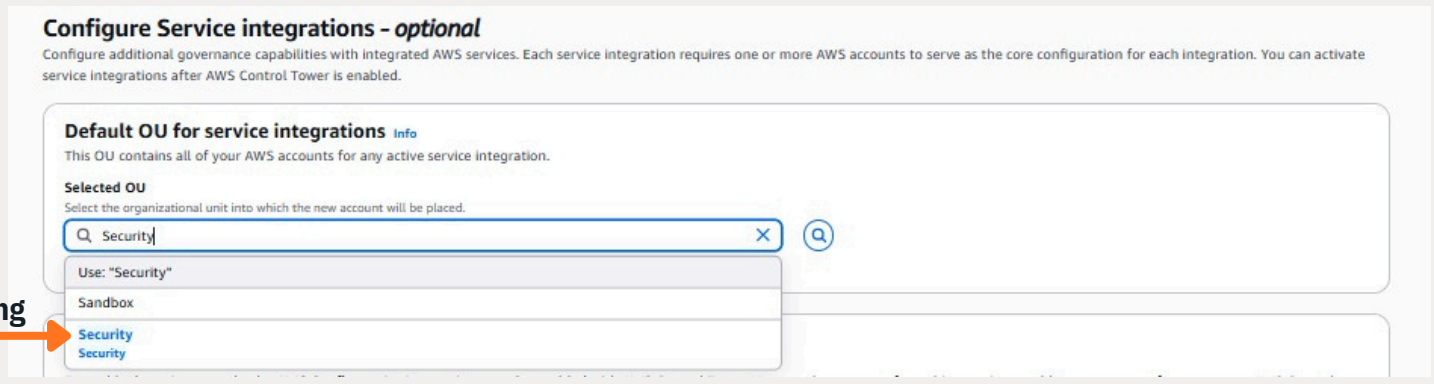
Steps:5. Configure Service Integrations (Optional)

AWS Control Tower can integrate with additional AWS services to extend governance and security capabilities.

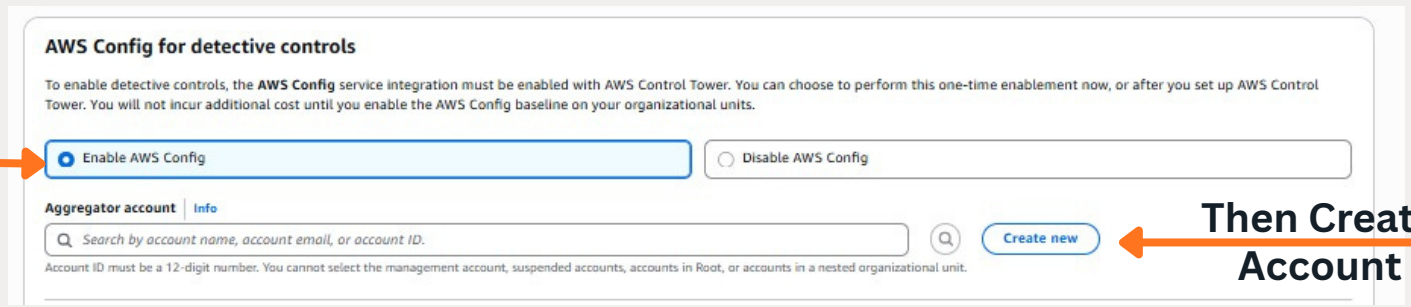
These integrations are optional and can be
enabled during setup or later.

1. Select the default OU for service integrations
   Choose the OU where new accounts will be placed when integrations create supporting resources.

For example, select: **Security OU**



2.Enable AWS Config for detective controls
Select:



AWS Config continuously records and evaluates AWS resource configurations to support security audits and compliance checks.

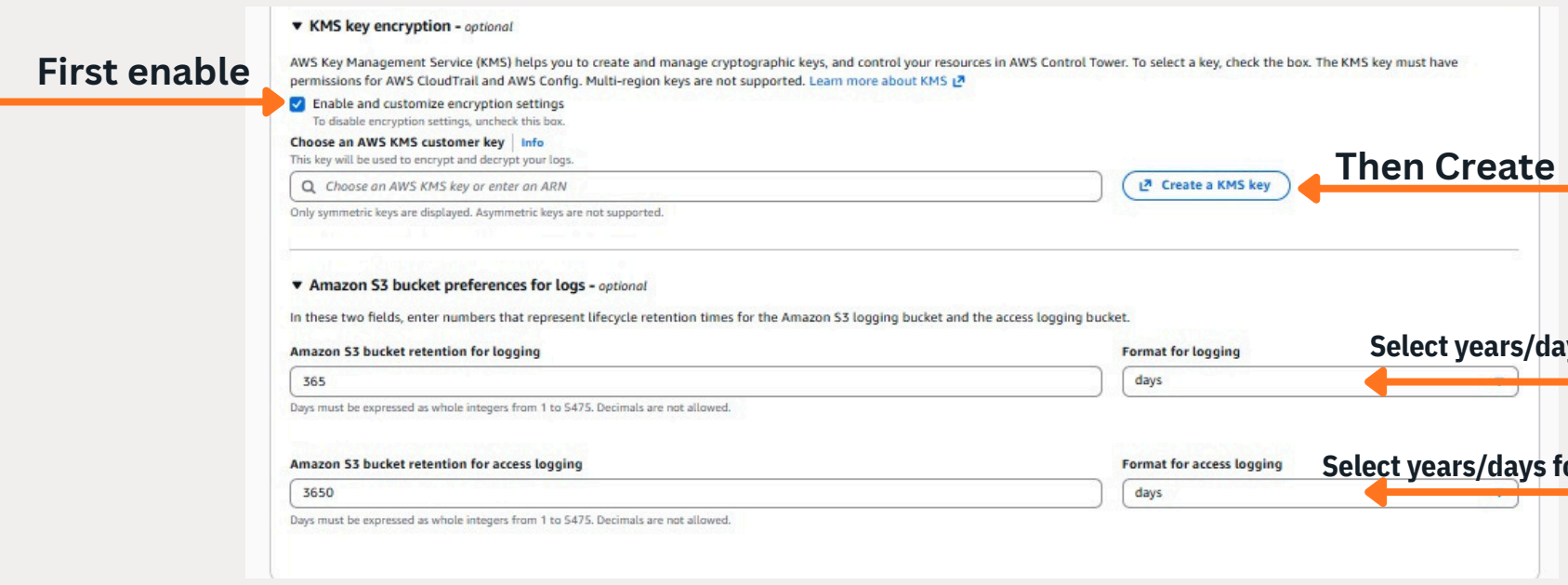
Specify an **aggregator account** (commonly the Log Archive or Security account).

3.(Optional) Enable KMS encryption for logs

If required, enable encryption and select or create a KMS key.

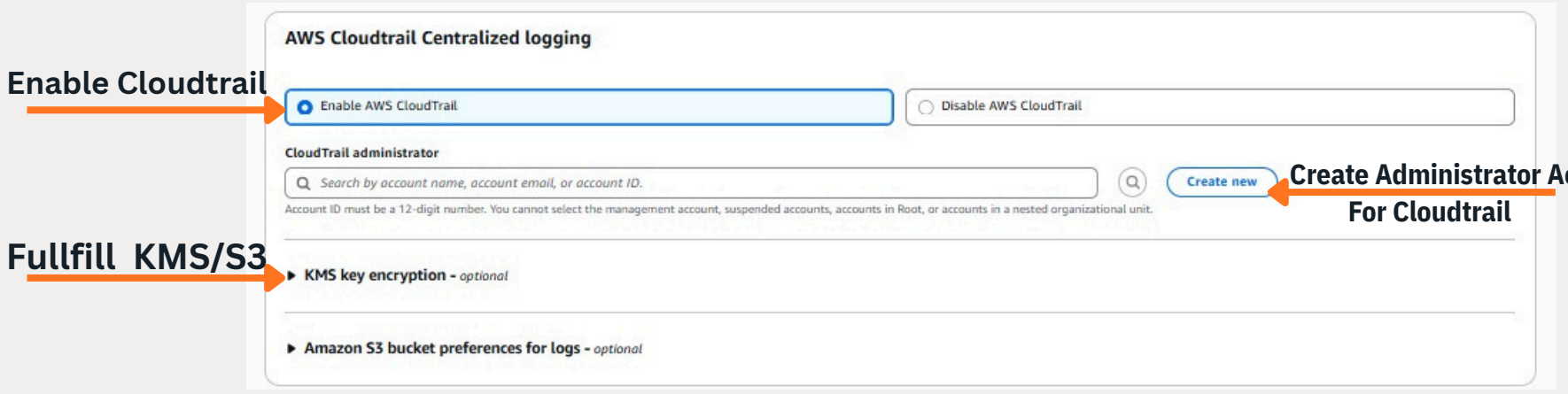
4.Configure S3 log retention (Optional)

Specify the number of days logs should be retained in the logging bucket.



5.Enable AWS CloudTrail centralized logging

Select: **Enable AWS CloudTrail**



This enables organization-wide logging for API activity.

Choose the CloudTrail administrator account. Also create the KMS key and Amazon S3 bucket for Cloudtrail encryption and store logs Click **Next** to continue.

Steps:6. Configure IAM Identity Center and AWS Backup

In this step, we configure authentication and optional backup services.

1.IAMIdentity Center Access

Select:**AWS Control Tower sets up AWS account access with IAM Identity Center**

2.AWSBackup (Optional)

Youmaychoose one of the following:

EnableAWS Backup

Enables centralized, automated backup management across AWS accounts

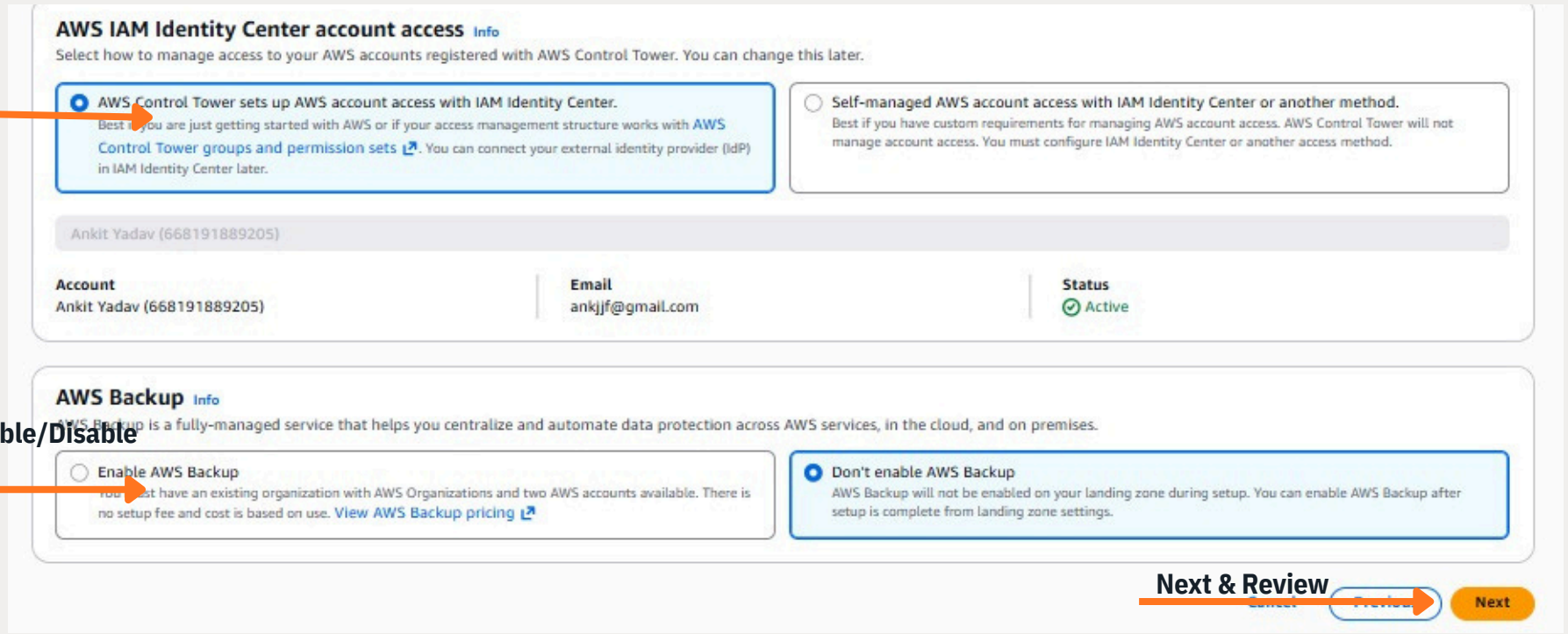
or

Don’tenable AWS Backup

Backup can be enabled later from Control Tower settings

After reviewing the configuration, click **Next**.

Select Access Account Setting



Backup Enable/Disable

PHASE 3 — Organizational Unit (OU) Structure Design

Note Point:

An Organizational Unit (OU) is like a folder inside AWS Organizations where you place AWS accounts. Each OU can have different security controls and policies. This helps apply governance, security, and compliance rules consistently.

In this phase, we design and create the OU hierarchy required for a secure AWS Landing Zone.

We will create the following OUs:

Foundation Layer (already created by Control Tower)

- Security OU
- Sandbox OU (Optional)

Business Layer

- Shared Services OU
- Development OU
- Testing OU
- Production OU

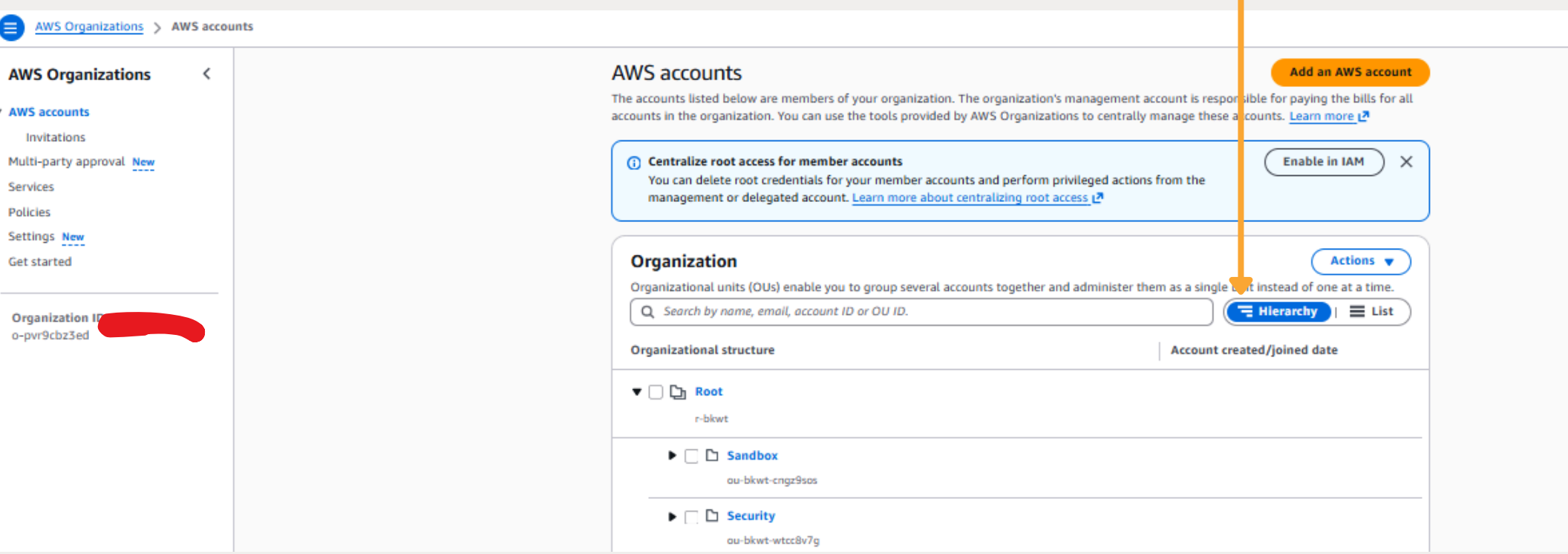
This structure separates environments and reduces blast-radius risk.

Step-1: Open AWS Organizations

1. Sign in to the AWS Console using your Management Account
2. In the search bar at the top, type “Organizations”
3. Click AWS Organizations

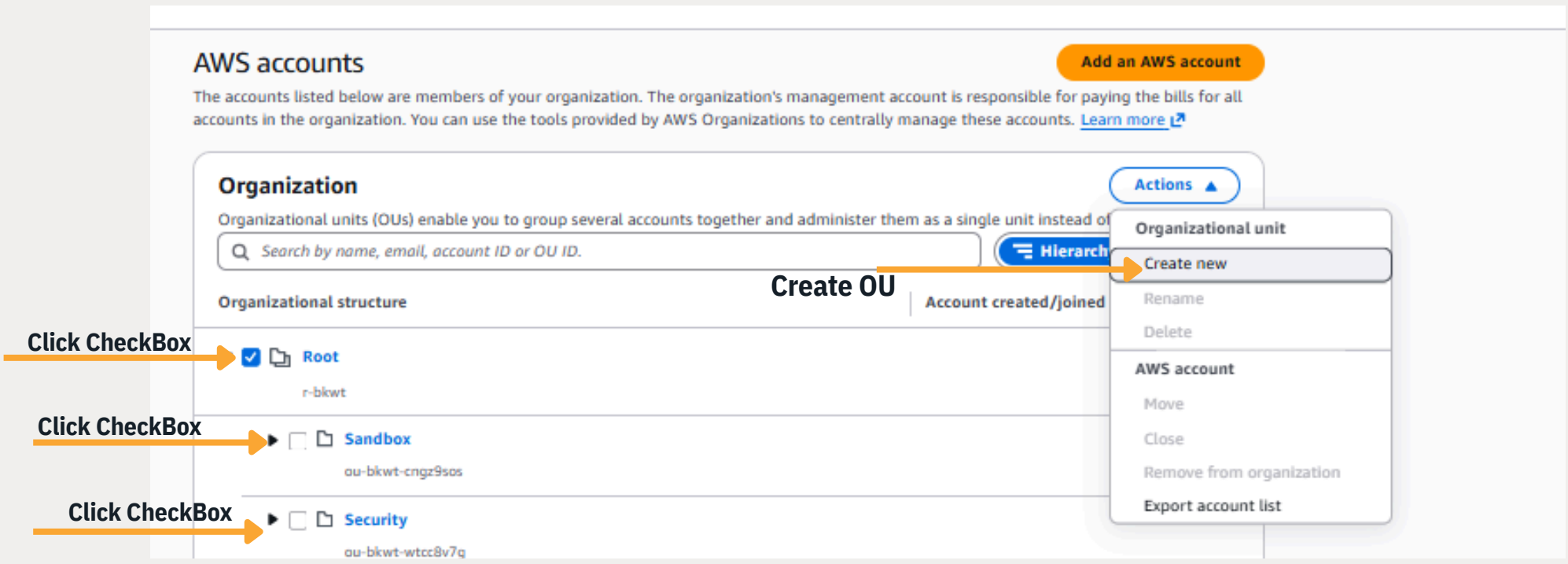
Step-2: Switch to Hierarchy View

1. At the top-right of the accounts list, click
2. Hierarchy View (if not already selected)



Step-3: Create a New Organizational Unit (OU)

1. Move your mouse over Root
2. Click the Actions button
3. Select
4. Create organizational unit

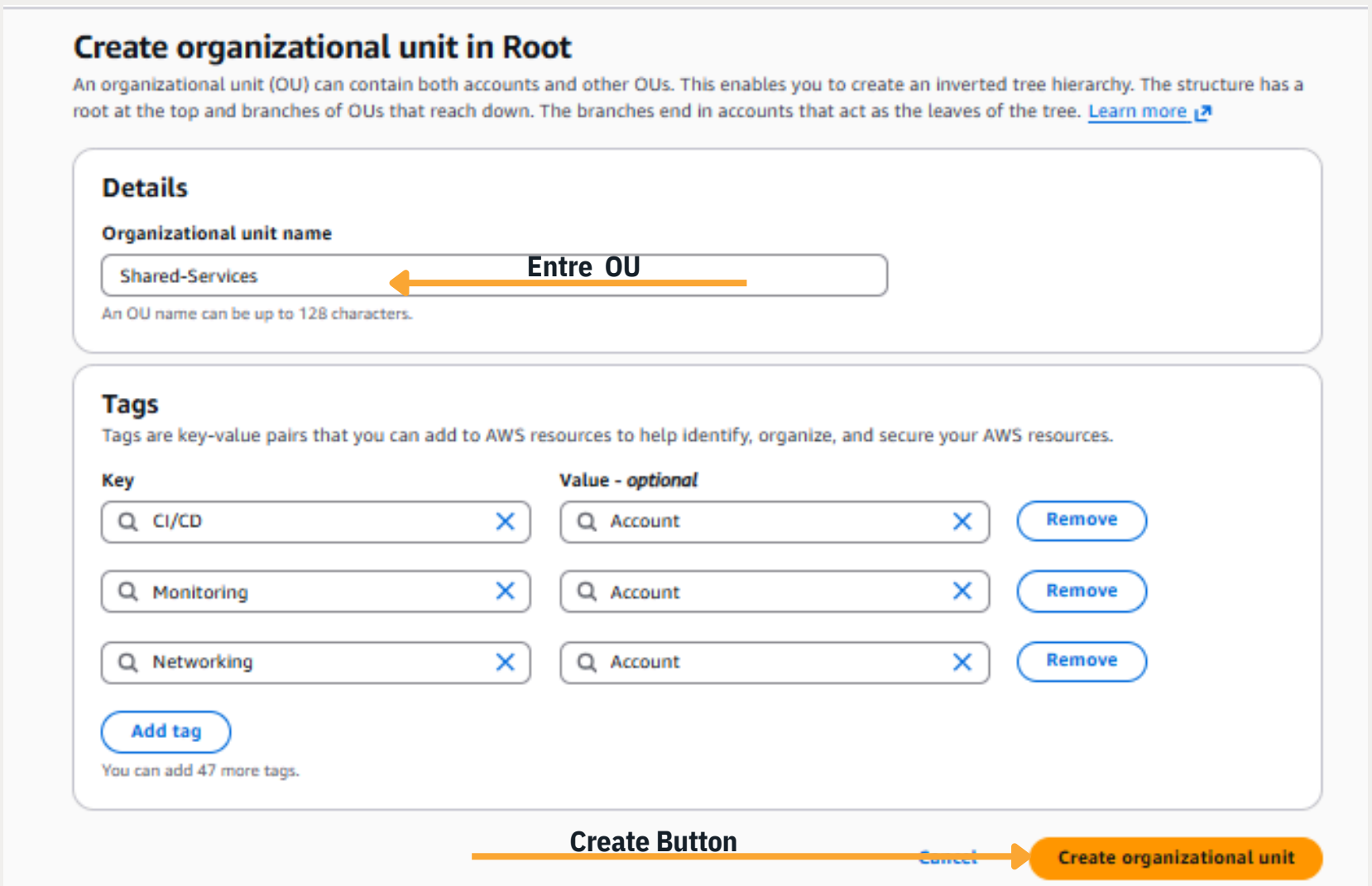


Step-4: Enter OU Details

1. Enter a name such as:

   Shared-Services

2. Click Create organizational unit



Your OU is now added under Root.

Step-5: Repeat for Other OUs

Follow the same steps to create the remaining OUs:

1.Development

2.Testing

3.Production

Notes (Important but short)

• Only the Management Account can create OUs

• OUs are logical folders, not AWS resources

• Policies can be applied later (SCPs, Guardrails etc.)

• AWS Control Tower works on top of OUs

Result should look like:

Root

├── Sandbox

├── Security

├── Shared-Services

├── Development

├── Testing

└── Production

PHASE-4 — Create AWS Accounts Inside Organizational Units (OUs)

In this phase, we will create multiple AWS accounts under the required Organizational Units (OUs), using AWS Control Tower. These will become separate, fully-managed accounts inside the AWS Organization.

You will create dedicated accounts such as:

✓ Networking Account

✓ CI/CD Account

✓ Monitoring Account

✓ Log-Archive Account

✓ Aduit Account

✓ Development Account

✓ Testing / QA Account

✓ Production Account

Each account will be mapped to the correct OU.

Step 1. Open AWS Organizations

1. Login using your Management Account
2. In the AWS Console search bar, type:
3. Click AWS Organizations

   You will see the Accounts & OU hierarchy page.

Step 2. Switch to Hierarchy View (if not already)

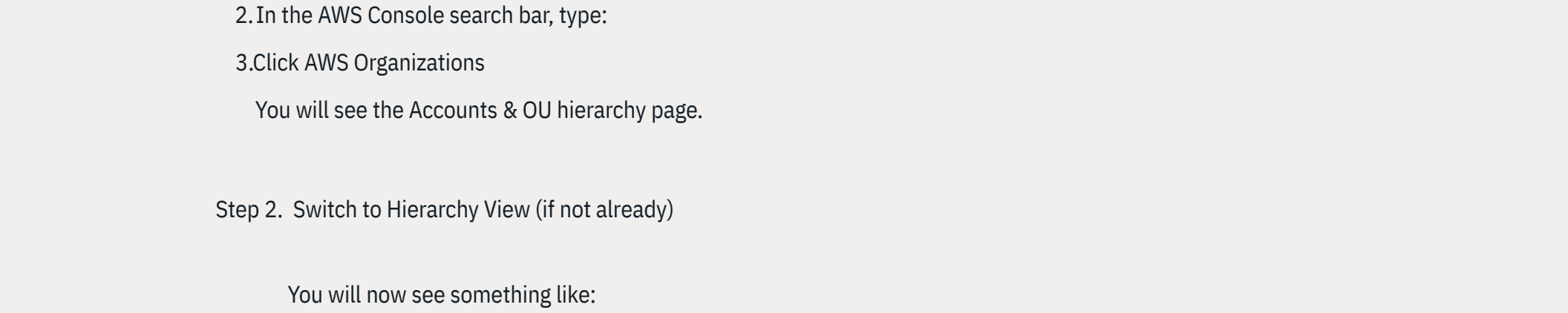
You will now see something like:

Root

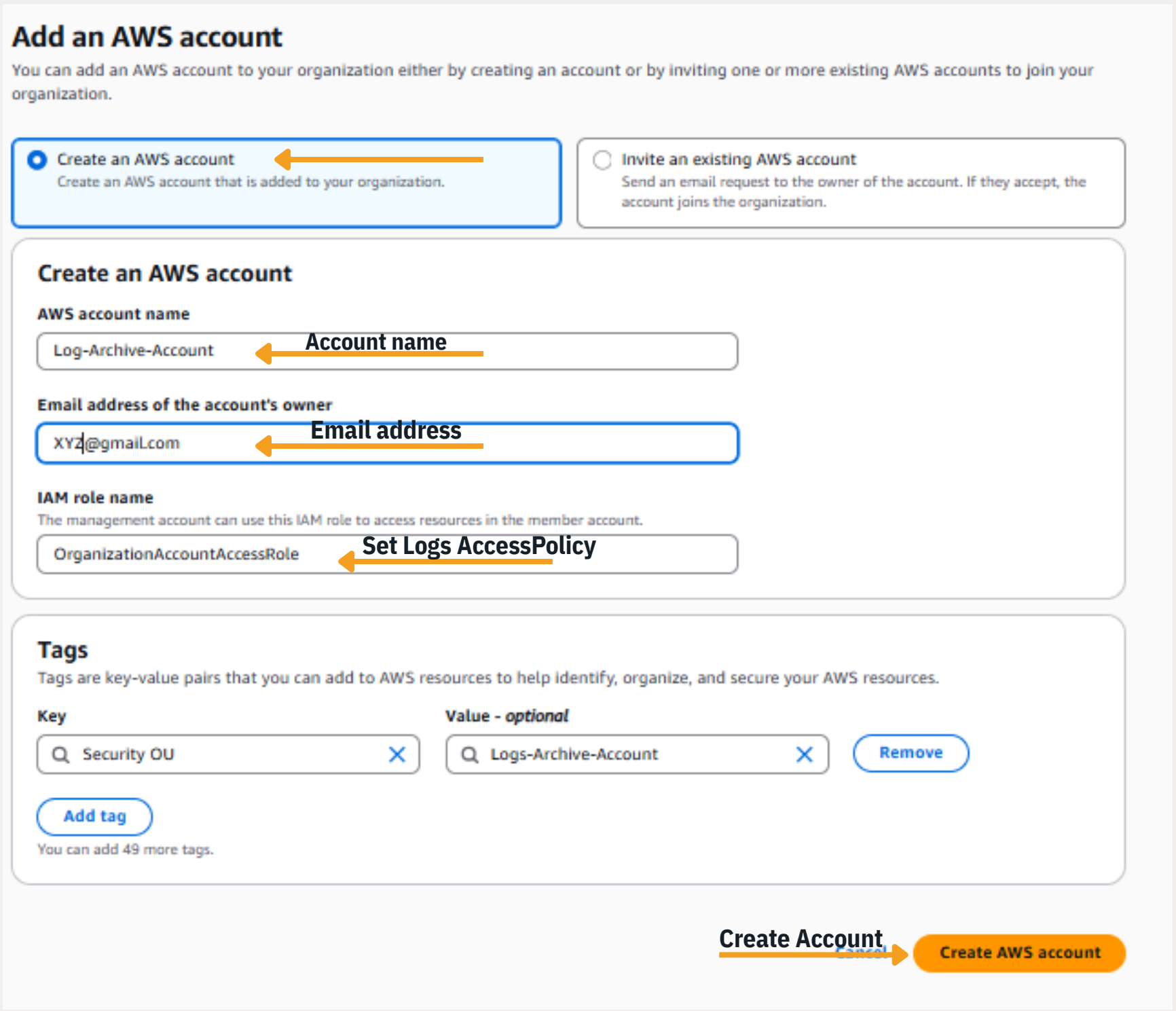
- ├── Security
- ├── Shared-Services
- ├── Development
- ├── Testing
- └── Production

Step 3. Start Creating a New AWS Account

Click: **Add an AWS Account**



Step 4. Fill Account Details



Step 5. Repeat for remaining accounts

Create:

1. Development

Account Name: Development-Account
Email: dev@domain.com
OU: Development

2. Testing

Account Name: QA-Account
Email: qa@domain.com
OU: Testing

3. Production

Account Name: Production-Account
Email: prod@domain.com
OU: Production

Step 6. Move an AWS Account into an Organizational Unit (OU)

1. Select the AWS Account You Want To Move

   Click the checkbox next to the account name

   Example: Log-Archive-Account

2. Move the Account into the OU

   Click the Actions button (top-right)

   Select: Move

Note Point : A popup appears listing all OUs

Select the target OU

Example:

Security OU

3.Click:

Move an AWS Account

Result

The account now appears under:

```
Root
└──── Security OU
    └──── Log-Archive-Account
```

FINAL AWS ORGANIZATION HIERARCHY (Enterprise-Grade)

Root
 |
 ├── Security OU
 | |
 | ├── Log Archive Account
 | | - Centralized S3 Log Storage
 | | - Stores CloudTrail / Config / Access Logs
 | |
 | └── Audit Account
 | - Read-only security investigation account
 | - Used by Security team for audit & compliance
 |
 ├── Shared Services OU
 | |
 | ├── Networking Account
 | | - Centralized VPC / Transit Gateway / Peering
 | |
 | ├── CI/CD Account
 | | - CodePipeline / Jenkins / GitHub runners
 | |
 | └── Monitoring Account
 | - CloudWatch dashboards
 | - Alerts & SNS
 | - Observability stack
 |
 ├── Development OU
 | |
 | └── Dev Account
 | - Developer sandbox
 | - Lower-risk workloads
 |
 ├── Testing OU
 | |
 | └── QA Account
 | - QA / UAT workloads
 |
 └── Production OU
  |
  └── Production Account
  - Mission-critical applications
  - Restricted access

Phase-5: Identity & Access Management using AWS IAM Identity Center (AWS SSO)

The goal of this phase is to centralize authentication and authorization across all AWS accounts inside the Organization. Instead of maintaining separate users and passwords in every AWS account, IAM Identity Center (formerly AWS SSO) allows users to log in once and securely access multiple AWS accounts with assigned roles and permissions.

Step-1: Enable IAM Identity Center

- Log in to the AWS console using your IAM Administrator user in the Management Account.
- In the AWS console search bar, type:

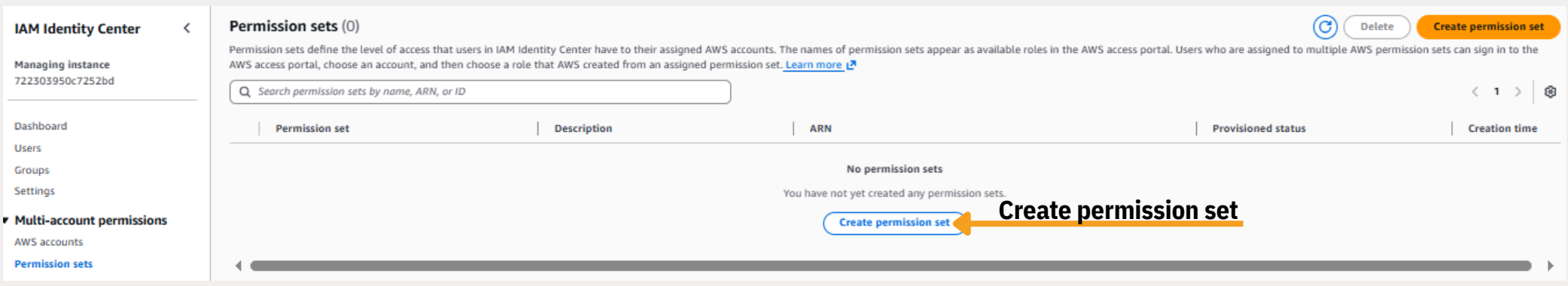
IAM Identity Center

- Open the service.
- Click: Enable IAM Identity Center

Step-2: Create Permission Sets (Role Templates)

Permission sets define the level of access granted to users in AWS accounts.

These are reusable and can be applied across accounts.



1. Cloud Administrator

Used by central platform/admin engineers.

Steps:

1. Open

   IAM Identity Center → Permission Sets

2. Click

   Create permission set

3. Select

   Predefined Permission Set

4. Choose

   AdministratorAccess

5. Name : Cloud-Admin

6. save

2. Developer Access

Used by application and DevOps engineers.

Steps:

1. Click

   Create permission set

2. Select

   Predefined Permission Set

3. Choose

   PowerUserAccess

4. Name it

   Developer

5. Save

3. Read-Only Access

Used by audit and compliance users.

Steps:

1. Click

   Create permission set

2. Select

   Predefined Permission Set

3. Choose

   ReadOnlyAccess

4. Name it

   ReadOnly

5. Save.

Step-3: Create Users in IAM Identity Center

Users will authenticate through IAM Identity Center instead of IAM Users.

Steps:

1. Open
2. IAM Identity Center → Users
3. Click
4. Add user
5. Enter:
    - First Name
    - Last Name
    - Email Address
6. Enable:

    Require MFA
7. Click
8. Create user

The user will receive an email invitation with a login link.

(You may also create Groups here if required, such as Cloud-Admins, Developers, Security-Team, etc.)

Step-4: Assign Users to AWS Accounts and Roles

Now associate:

✓ a user

✓ a permission set

✓ an AWS account

Steps:

Open

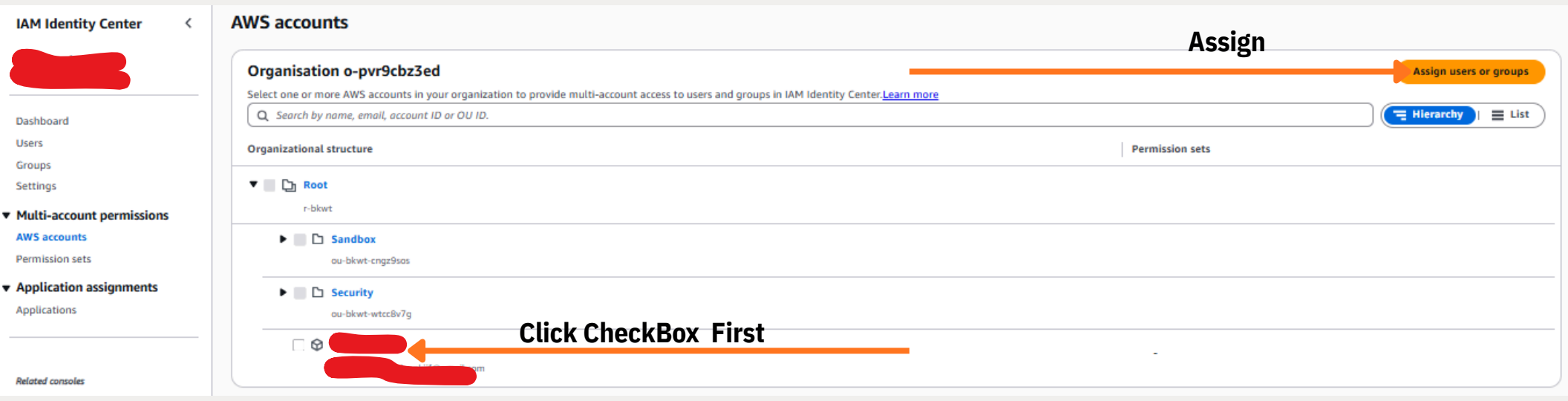
IAM Identity Center → AWS Accounts

1. Select the AWS account where you want to assign access.
2. Click

    Assign users or groups
3. Select the user (or group)
4. Select the permission set

    (Cloud-Admin, Developer, ReadOnly)
5. Click
6. Assign

Repeat for each user and account.



Recommended Access Mapping

Team	Accounts	Role
Cloud Admins	All Accounts	Cloud-Admin
Developers	Dev + QA	Developer
Security Team	Security OU	ReadOnly + Cloud-Admin
Auditors	Log Archive + Audit	ReadOnly

User Login Experience

Once configured:

1. The user receives a login email.
2. They sign in to the IAM Identity Center Portal.
3. They see all AWS accounts they have access to.
4. They select:
    - The AWS account
    - The role assigned to them
5. They are redirected into that AWS account console.

Easy

PHASE-6 — Centralized Networking Setup (Only For Shared Services Account + Transit Gateway)

In a multi-account AWS environment, every account should not create its own random VPCs and networking.

Instead, we create one central networking hub inside the Shared-Services (Networking) Account.

✔ A single secure network backbone

✔ Central routing and connectivity

✔ Easier security monitoring

✔ Cleaner architecture

✔ Lower cost

This is done using:

🟢 A Central VPC

🟢 A Transit Gateway (TGW)

🟢 VPC Attachments for other accounts

🟢 AWS Resource Access Manager (RAM) for sharing

Note Point

Step-1 — Login to the Networking Account

1.Login using IAM Identity Center

2.Select the Networking Account

Step-2 — Create the Central VPC

1. Go to:

AWS Console → VPC

2. Click:

Create VPC

3. Select:

VPC and more

VPC Settings----------

VPC name	Central-VPC
IPv4 CIDR block	10.0.0.0/16
AZs	2 or 3
Public Subnets	Yes
Private Subnets	Yes
NAT Gateways	1 or more
DNS Hostnames	Enabled

4. Click : Create VPC

Step-3 — Create a Transit Gateway (TGW)

1. Go to:

VPC Console → Transit Gateways

2. Click:

Create Transit Gateway

Field	Value
Name	Central-TGW
ASN	Default is fine
Amazon side ASN	leave default
DNS Support	Enabled
Multicast	Disable

3.Click:

Create Transit Gateway

Step-4 — Attach the Central VPC to TGW

1.Go to:

Transit Gateway Attachments

2.Click:

Create attachment

3.Select:

Resource type → VPC

4.Then choose:

VPC → Central-VPC

Subnets → select private subnets

5.Click:

Create attachment

Step-5 — Share the Transit Gateway with Other Accounts

We share the TGW using AWS RAM (Resource Access Manager).

1. Go to:

AWS Console → Resource Access Manager (RAM)

2. Click:

Create resource share

3.Enter:

Name → TGW-Share

Under Resources

4.Select:

Transit Gateway

Under Principals

5.Select:

AWS Organization

6.Click:

Create resource share

🎉 Now Dev, QA, and Prod accounts can use this Transit Gateway.

Step-6 — Attach Other Account VPCs to TGW

Login to each account one-by-one:

• Development

• Testing

• Production

Then in each account:

1.Go to:

VPC Console → Transit Gateway Attachments

2.Click:

Create attachment

3.Select:

Resource → VPC

Transit Gateway → Central-TGW

VPC → That account's VPC

Subnets → Private Subnets

4.Click:

Create attachment

Step-7 — Configure Routes (So Traffic Can Flow)

In each attached VPC

1.Go to:

Route Tables

For Private Route Table

2.Add route:

Destination: 10.0.0.0/8 (or your org CIDRs)

Target: Transit Gateway

Notes Point :

Step-8 — Security Controls
Ensure:
✔ Only Private Subnets use TGW
✔ Public Subnets route to Internet Gateway
✔ Security Groups restrict access
✔ NACLs are not blocking

Easy/Important

PHASE-7 — Centralized Logging & Monitoring (CloudTrail + S3 Log Archive + CloudWatch Monitoring Account)

The goal of this phase is to enable centralized and tamper-proof logging across all AWS accounts in the organization. Every API activity and user action will be logged centrally in a dedicated Log-Archive Account, and monitoring will be performed from a separate Monitoring Account.

✓ Organization-Level CloudTrail
✓ Logs stored in Log-Archive Account (S3)
✓ S3 bucket versioning + encryption enabled
✓ Access logging enabled
✓ Monitoring Account se alerts & dashboards

Part-1 — Organization-Level CloudTrail Enable Karna

Notes Point: Login Scope
Log in using IAM Identity Center into the Management Account (do not use the root user).

Step-1 — Open CloudTrail Console As Management Account Important

Navigate to:

AWS Console → CloudTrail

Step-2 — Create a New Trail

Select:

Create Trail

Step-3 — Configure the Trail

Set the following values:

Field	Value
Trail Name	Org-CloudTrail
Trail Type	Organization Trail
Apply to All Regions	Enabled
Management Events	Read & Write
Data Events	Optional (enable for S3/Lambda when required)
Insight Events	Optional (Recommended = Enabled)

Also enable:

Apply trail to all member accounts

This ensures every AWS account in the organization is covered

Part-2 — Store Logs in the Log-Archive Account

CloudTrail will now require an S3 bucket.
This bucket must exist in the Log-Archive Account.

Step-4 — Create an S3 Log Bucket

Note Point: Login to the Log-Archive Account via IAM Identity Center and navigate to:

AWS Console → S3 → Create Bucket

Field	Value
Bucket-Name	org-cloudtrail-logs-company
Region	Same as Control Tower Home Region
Block Public Access	Enabled
Encryption	AES-256 or KMS
Versioning	Enabled

This ensures logs cannot be silently deleted or overwritten.

Step-5 — Allow CloudTrail to Write Logs

CloudTrail will automatically generate the required bucket policy.
Approve and apply the policy.

Now:
✔ Every AWS account
✔ In every region
✔ Logs to this single bucket

Step-6 — Enable Access Logging on the Log Bucket
This creates an audit trail of who accessed the logs.

S3 → Bucket → Properties → Server Access Logging → Enable

Part-3 — Monitoring & Alerting (Monitoring Account)

Login Scope
Log in to the Monitoring Account

Step-8 — Open CloudWatch Console

AWS Console → CloudWatch

Step-9 — Create Dashboards
Create dashboards for:

✔ Resource Health
✔ Security Events
✔ Log Metrics
✔ Cost Visibility

Step-10 — Configure CloudWatch Alarms

Recommended alerts include:

🚨 Root user login detected
🚨 IAM policy change
🚨 CloudTrail stopped
🚨 Public Security Group exposure
🚨 Billing exceeds threshold

These alarms send notifications to the security team.

Step-11 — Create an SNS Topic for Alerts

AmazonSNS → SNS → Create Topic

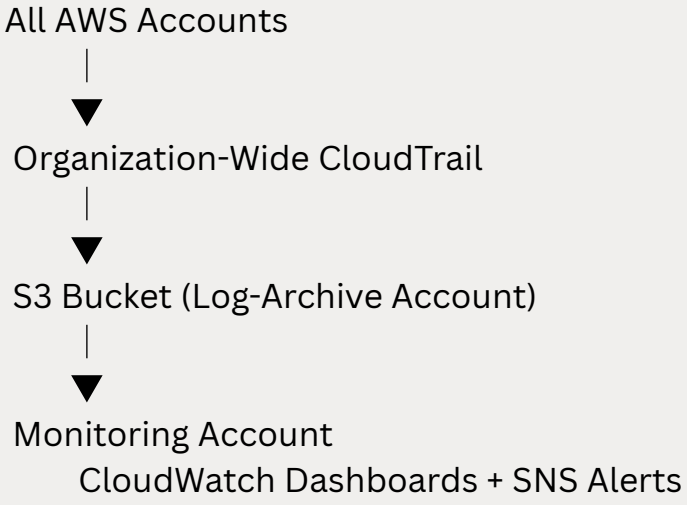
Example: Security-alert

Add subscribers such as:

security@company.com
cloudops@company.com

Confirm email subscriptions.

High-Level Architecture



Phase-8 — Security Foundations (Centralized Security Services Setup)

In this phase, we configure organization-wide security monitoring using the Security Account.
This ensures that all AWS member accounts are monitored for threats, misconfigurations, and suspicious activity.

We will enable:

✔ Amazon GuardDuty
✔ AWS Security Hub
✔ IAM Access Analyzer
✔ Amazon Detective

All security findings will be collected centrally in the Security Account.

Step-1 — Login to the Security Account

1. Open IAM Identity Center (SSO)
2. Select the Security Account
3. Login using your Security/Admin role

Step-2 — Enable Amazon GuardDuty (Organization-wide)

- Go to AWS Console
- Search → GuardDuty
- Open the service

If not enabled yet:

• Click Enable GuardDuty

1. Enable as Organization Admin

To manage all AWS accounts from the Security account:

1. In left menu → Settings
2. Click → Accounts
3. Select → Enable organization

AWS will now detect all existing and future accounts.

2. Auto-Enable GuardDuty for New Accounts

1. Go to Settings
2. Turn ON:

✔ Automatically enable GuardDuty for new accounts

3. Verify Findings Delivery

GuardDuty findings will now flow into:

✔ GuardDuty console
✔ Security Hub (later when enabled)

Step-3 — Enable AWS Security Hub (Central Aggregation)

1. Open Security Hub
   1. Open AWS Console
   2. Search → Security Hub
   3. Click Enable Security Hub

2. Enable Organization Admin
   1. Go to Settings → Accounts
   2. Click → Designate as Security Hub administrator
   3. Select Security Account

3. Auto-Enable for All Member Accounts

Turn ON:

✓ Auto-Enable Security Hub

4. Enable Integrations

Security Hub integrates with:

✓ GuardDuty
✓ IAM Access Analyzer
✓ Detective
✓ Inspector (optional)

Ensure they are ON.

Step-4 — Enable IAM Access Analyzer (Organization-wide)

IAM Access Analyzer detects:

✓ Public S3 Buckets
✓ Cross-Account Access
✓ Unintended Sharing

1 Enable Analyzer

1. Open AWS Console

Search → IAM

2. Go to Access Analyzer

Click → Create Analyzer

3. Select:

Analyzer type → Organization

4. Click Create

Now IAM Access Analyzer will:

✓ Monitor all accounts
✓ Send findings to Security Hub

Step-5 — Enable Amazon Detective

Detective helps analyze suspicious activity & security findings.

1 Open Detective

1.Open AWS Console

Search → Detective

2.Click → Enable Detective

2 Enable Organization Admin Mode

1.Go to Settings

Select → Enable organization

2.Make Security Account as admin

Step-6 — Centralize All Alerts (Highly Recommended)

Alerts should reach your team. Typical channels:

✔ Email
✔ Slack
✔ Microsoft Teams
✔ PagerDuty

1 Create SNS Topic (Alert Hub)

1.Open AWS Console

Search → SNS

2.Click → Create Topic

3.Type = Standard

Name Example: security-alerts-topic

4.Create topic

2 Subscribe Notification Channels

---------->Example — Email

1 Click → Create Subscription

1.Protocol = Email

Enter email → security-team@company.com

2.Confirm email link

3 Send Findings to SNS

From:

✔ Security Hub → Insights → Create Automation
✔ Or CloudWatch Rules for security events

(Create rule for GuardDuty findings too)

Step-7 — Validate Security Setup

Confirm:

- ✓ GuardDuty enabled across org
- ✓ Security Hub receiving findings
- ✓ IAM Access Analyzer detecting sharing
- ✓ Detective active
- ✓ Alerts reaching email/Slack
- ✓ Auto-enable ON for new accounts

Final Architecture Summary — AWS Multi-Account Landing Zone

This solution implements a secure, scalable, and enterprise-grade AWS Landing Zone using AWS Control Tower, AWS Organizations, and multiple dedicated accounts.

The objective is to separate environments, enforce governance, centralize security, and enable safe cloud adoption across the organization.

High-Level Design

The AWS environment is structured using multiple AWS accounts grouped under Organizational Units (OUs):

- Security OU
  - Log Archive Account
  - Audit Account
- Shared Services OU
  - Networking Account
  - CI/CD Account
  - Monitoring Account
- Development OU
  - Development Account
- Testing OU
  - QA / Testing Account
- Production OU
  - Production Account

This structure ensures strong workload isolation, reduced blast-radius, better governance, and easier cost & access control.

Landing Zone & Governance

AWS Control Tower is used to deploy the Landing Zone, which automatically configures:

- Centralized identity and access
- Baseline security guardrails
- Central logging and auditing
- Governance across all accounts
- Automated account provisioning

Identity & Access Management

Authentication and access are centralized using IAM Identity Center (AWS SSO).
Users are assigned to permission groups such as:

- Cloud Administrators
- Developers
- Security Team
- Read-Only Users

Networking Architecture

A dedicated Networking Account hosts the central VPC footprint and shared networking resources. Key elements include:

- Standardized VPC design
- Public & private subnets across Availability Zones
- Centralized routing using AWS Transit Gateway
- Cross-account resource sharing via AWS Resource Access Manager

Centralized Logging & Monitoring

All AWS accounts send audit and activity logs to the Log Archive Account, including:

- AWS CloudTrail organization trails
- S3 access logs
- Other security & audit logs

.

Security Foundation

Core security services are enabled at the organization level:

- Amazon GuardDuty — continuous threat detection
- AWS Security Hub — unified security findings dashboard
- IAM Access Analyzer — public/mis-shared resource detection
- Amazon Detective — investigation and root-cause analysis

Benefits of This Architecture

This Landing Zone delivers the following outcomes:

- Strong Account Isolation — limits blast-radius and security exposure
- Centralized Governance & Identity — consistent access controls across the organization
- Security-by-Default — GuardDuty, logging, and auditing enabled from Day-1
- Compliance Readiness — structured logs and security scanning
- Operational Scalability — accounts can be added and governed easily
- Cost & Ownership Clarity — workloads billed and managed independently
- Production Safety — Development / QA / Production environments fully isolated

Conclusion

By implementing an AWS multi-account Landing Zone, the organization achieves a secure, scalable, and well-governed cloud foundation aligned with enterprise best practices.
This architecture not only improves security, monitoring, and compliance, but also supports future growth, DevOps maturity, and multi-team collaboration — without compromising control or visibility.