# AWS Control Tower & AFT Multi-Account Strategy Implementation

AWS Control Tower Management Structure

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Component | Purpose | Scope / Hierarchy | Billing Role | Access Management Role | Key Notes |
| Root | Top-level entity of AWS Organizations | Root of the Org Tree | Management Account handles consolidated billing | No direct user access; manages policies | Only one per organization; cannot be deleted |
| Organizational Units (OUs) | Logical grouping of accounts for governance | Nested under Root | Helps group accounts for cost analysis | SCPs applied to control account permissions | SCPs inherited by all accounts in OU |
| AWS Account | Isolated unit for running AWS workloads | Leaf nodes under OUs or Root | Individual usage rolled up to Management Account | IAM users/roles and Identity Center access apply | Best practice: isolate workloads by account for security and cost management |
| Service Control Policies (SCPs) | Guardrails for what AWS actions are allowed across accounts | Applied to Root/OUs/Accounts | N/A | Restrict max available permissions | Do not grant permissions—only restrict them |
| AWS Control Tower | Automates setup of secure, governed multi-account AWS environment | Orchestrates entire hierarchy | Prepares account structure for consolidated billing | Integrates with IAM Identity Center | Sets up Landing Zone, Account Factory, SCPs, logging, and auditing |
| IAM Identity Center | Centralized SSO and permissions across AWS accounts | Spans all accounts under Org | N/A | Manages user access and permission sets | Can connect to external IdPs like Azure AD or Okta |

## 1. High-Level Components Overview

• AWS Organizations: Creates a hierarchy of AWS accounts and OUs. Foundation for centralized governance.

• Control Tower: Orchestrates creation of a secure, well-architected Landing Zone using best practices.

• Landing Zone: Pre-configured multi-account architecture with governance: includes log archive, audit, etc.

• IAM Identity Center (AWS SSO): Centralized identity management with permission sets to access multiple accounts securely.

• Account Factory: AWS Control Tower feature to create accounts manually or via Service Catalog.

• Account Factory for Terraform (AFT): IaC solution to create and customize accounts using Git + Terraform.

• Organizational Units (OUs): Logical containers for accounts; used to apply Service Control Policies (SCPs).

• Service Control Policies (SCPs): Permission guardrails for accounts under OUs—enforced via AWS Organizations.

• CloudTrail / Config / GuardDuty: Centralized logging and compliance visibility.

• Shared Services Account: Hosts VPCs, DNS, CI/CD, AD or proxy for Identity Center etc.

• Security Tooling Account: Hosts security-related tools like GuardDuty, Security Hub.

• Log Archive Account: Central sink for CloudTrail, Config logs. Immutable logging.

## 2. How Components Work Together

1. Foundation: AWS Organizations provides the root of the multi-account hierarchy. Control Tower sets up a secure Landing Zone with mandatory accounts and predefined OUs (Security, Sandbox, Workloads).  
2. IAM Identity Center: Enables centralized SSO access with permission sets mapped to users/groups from external identity providers (e.g., Azure AD).  
3. AFT Account Provisioning: Developers submit account requests to a Git repo. AFT automates account creation, OU placement, SCP application, and baseline setup using Terraform.  
4. Governance & Logging: SCPs restrict sensitive actions, and audit logs are sent to the Log Archive account. GuardDuty and Config are centralized for security visibility.

## 3. Logical Architecture Flow

+------------------------------+  
| AWS Control Tower |  
|------------------------------|  
| Landing Zone Setup |  
| ├── Security OU |  
| ├── Workloads OU |  
| ├── Sandbox OU |  
| ├── Suspended OU |  
| └── Root OU |  
+------------------------------+  
 ↓  
+------------------------------+  
| AWS Organizations |  
| └── OU Hierarchy |  
| └── SCP Guardrails |  
+------------------------------+  
 ↓  
+------------------------------+  
| IAM Identity Center (SSO) |  
| └── User Access & Roles |  
| └── Federated Login |  
+------------------------------+  
 ↓  
+------------------------------+  
| AFT (Terraform + CodePipeline) |  
| └── Git-driven Account Creation |  
| └── OU placement + Baseline |  
+------------------------------+  
 ↓  
+------------------------------+  
| Account Outputs |  
| ├── Shared Services |  
| ├── Workloads (prod/dev) |  
| ├── Security Tooling |  
| ├── Log Archive |  
+------------------------------+

## 4. Optional Enhancements

**List of AFT Terraform Module Templates**

| **Category** | **Module Name** | **Purpose / Description** |
| --- | --- | --- |
| 🏗️ Core Provisioning | aft-account-request | Entry module for account creation requests via AFT Git repo |
| 🔁 AFT Bootstrap | aft-bootstrap | Sets up AFT framework resources (CodePipeline, S3, DynamoDB, etc.) |
| 📁 Organizational Units | aft-organization-units | Automates creation of OUs within AWS Organizations |
| 👥 Accounts | aft-account-provisioning | Core module to create AWS accounts in OUs using Control Tower |
| ⚙️ Baseline Configuration | aft-account-baseline | Applies baseline configs (logging, SCP attachment, IAM roles) to new accounts |
| 🛡️ SCP Management | aft-scps | Deploys and attaches Service Control Policies to OUs |
| 📊 Budget Control | aft-budgets | Creates AWS Budgets and sets notifications per account |
| 🔐 IAM & SSO | aft-iam-baseline | Defines IAM roles/policies and/or configures SSO for new accounts |
| 📜 CloudTrail & Logging | aft-centralized-logging | Enables GuardDuty, Config, and CloudTrail into centralized Log Archive account |
| 🌐 Networking | aft-shared-vpc | Provisions and shares a central VPC across accounts using Resource Access Manager (RAM) |
| 🔄 Customizations | aft-customizations | For additional per-account resources (e.g., DNS zones, backups, tagging policies) |
| 🏷️ Tag Enforcement | aft-tag-enforcer | Applies tag policies or validates tags using AWS Config rules |

**Folder Structure in Terraform**

terraform/

├── modules/

│ ├── aft-account-request/

│ ├── aft-bootstrap/

│ ├── aft-organization-units/

│ ├── aft-account-provisioning/

│ ├── aft-account-baseline/

│ ├── aft-scps/

│ ├── aft-budgets/

│ ├── aft-iam-baseline/

│ ├── aft-centralized-logging/

│ ├── aft-shared-vpc/

│ └── aft-customizations/

├── environments/

│ ├── prod/

│ ├── non-prod/

│ └── sandbox/

└── terraform.tfvars

• Custom Terraform Modules: Enforce tagging, deploy VPCs, IAM roles per environment

• Notification Integrations: Slack, email, or ChatOps alerts on new account creation

• Cross-account roles: For centralized DevOps tools or Security teams

• Cost Explorer + Budgets: Track spend per OU or project

## 5. Governance: SCPs, CloudTrail, Config, GuardDuty, Account Management

### 5.1 Service Control Policies (SCPs)

SCPs are used to enforce permission guardrails on AWS accounts at the Organizational Unit (OU) level. They do not grant permissions but define the maximum available permissions. Sample SCP use cases:

• Deny deletion of security tools (e.g., GuardDuty, AWS Config)

• Restrict IAM policy changes in Production accounts

• Enforce tagging or prevent specific regions/services usage

• Deny `iam:\*`, `ec2:TerminateInstances`, `s3:DeleteBucket` for non-prod accounts

### 5.2 AWS CloudTrail

CloudTrail is enabled across all accounts, with trails aggregated into a centralized S3 bucket in the Log Archive account. This provides a tamper-proof audit log of all AWS API activity.

### 5.3 AWS Config

AWS Config is used to record configuration changes across resources and evaluate compliance against predefined rules. Its data is sent to the Log Archive account for central visibility. Examples of rules:

• Enforce encryption on S3 buckets

• Disallow public access to security groups

• Ensure IAM policies are attached only to groups or roles

### 5.4 GuardDuty

GuardDuty is enabled across all accounts and centrally managed in the Security Tooling account. It detects threats such as:

• Unusual API calls

• Reconnaissance activities

• Compromised IAM credentials or EC2 instances

### 5.5 AWS Account Management with AFT

AFT manages account provisioning via GitOps workflows. Terraform modules are triggered by changes to account request repositories. Key account lifecycle operations include:

• Provisioning accounts under specific OUs

• Applying baseline configurations (e.g., IAM roles, SCPs)

• Enabling centralized logging and monitoring

• Applying budgets and tag enforcement