AWS Organizations

1. Introduction to AWS Organizations

AWS Organizations is a cloud service that helps you centrally manage and govern your environment as you grow and scale your AWS resources. It provides a single point of control for multiple AWS accounts, enabling you to create new accounts programmatically, group them into organizational units (OUs), and apply policies across your accounts.

What is AWS Organizations?

AWS Organizations allows you to consolidate multiple AWS accounts under a single organization, providing centralized management capabilities including:

- Account creation and management
- Hierarchical grouping of accounts
- Policy-based governance
- Consolidated billing
- Centralized logging and monitoring

Why Use AWS Organizations?

Organizations become essential when managing multiple AWS accounts for different environments, teams, or business units. It provides governance, security, and cost management at scale while maintaining account isolation.

---

2. Core Concepts and Terminology

Organization

The top-level container that holds all your AWS accounts. An organization has one master account and zero or more member accounts.

Master Account (Management Account)

The AWS account that creates the organization. This account:

- Pays for all charges incurred by member accounts
- Cannot be restricted by Service Control Policies
- Has full administrative control over the organization

Member Account

AWS accounts that belong to an organization but are not the master account. These accounts:

- Are subject to policies applied by the organization
- Have their charges consolidated to the master account
- Can be moved between organizational units

Organizational Unit (OU)

A container for accounts within your organization. OUs allow you to group accounts and apply policies to multiple accounts simultaneously.

Root

The top-level parent container for all accounts and OUs in your organization. The root is created automatically when you create an organization.

Service Control Policy (SCP)

JSON policies that define the maximum permissions for accounts in your organization. SCPs act as guardrails, limiting what actions can be performed.

---

3. Key Features and Benefits

Centralized Account Management

- **Programmatic Account Creation**: Create new AWS accounts through APIs or console
- **Account Lifecycle Management**: Manage accounts from creation to closure

- **Automated Setup**: Apply configurations and policies to new accounts automatically

Hierarchical Organization Structure

- **Flexible Grouping**: Organize accounts by business unit, environment, or function
- **Nested OUs**: Create multi-level hierarchies for complex organizations
- **Policy Inheritance**: Apply policies at different levels of the hierarchy

Policy-Based Governance

- **Service Control Policies**: Define what services and actions are allowed
- **Preventive Controls**: Stop non-compliant actions before they occur
- **Compliance Enforcement**: Ensure adherence to organizational standards

Consolidated Billing

- **Single Bill**: Receive one bill for all accounts in the organization
- **Volume Discounts**: Benefit from aggregated usage across accounts
- **Cost Allocation**: Track costs by account, OU, or tags
- **Reserved Instance Sharing**: Share Reserved Instance benefits across accounts

Enhanced Security

- **CloudTrail Integration**: Centralized logging of API calls across accounts
- **Config Integration**: Monitor compliance across multiple accounts
- **Cross-Account Roles**: Simplified access management between accounts

4. Architecture and Structure

Typical Organization Structure

```
Root
├── Production OU
│   ├── Prod-WebApp-Account
│   ├── Prod-Database-Account
│   └── Prod-Monitoring-Account
├── Development OU
│   ├── Dev-Team-A-Account
│   ├── Dev-Team-B-Account
│   └── Shared-Dev-Services-Account
├── Testing OU
│   ├── QA-Account
│   └── Staging-Account
└── Security OU
    ├── Logging-Account
    ├── Audit-Account
    └── Identity-Account
```

Design Principles

Account Separation by Environment

- Separate production, staging, and development environments
- Isolate workloads to prevent cross-contamination
- Apply different policies based on environment criticality

Functional Account Separation

- Dedicated accounts for specific functions (logging, security, networking)
- Shared services accounts for common resources
- Application-specific accounts for isolation

Organizational Alignment

- Structure OUs to match business units or teams
- Apply governance policies that reflect organizational requirements
- Enable decentralized management while maintaining control

5. Service Control Policies (SCPs)

Understanding SCPs

Service Control Policies are JSON-based policies that define the maximum permissions for accounts in your organization. They work as filters, allowing or denying access to AWS services and actions.

Key Characteristics

- **Preventive Controls**: SCPs don't grant permissions; they only limit them
- **Inheritance**: Policies are inherited down the organizational hierarchy
- **Combination Logic**: Multiple SCPs are combined using logical AND operations
- **Master Account Exception**: SCPs don't apply to the master account

SCP Examples

Restrict EC2 Instance Types

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "ec2:RunInstances"
      ],
      "Resource": "arn:aws:ec2:*:*:instance/*",
      "Condition": {
        "ForAllValues:StringNotEquals": {
          "ec2:InstanceType": [
            "t3.micro",
            "t3.small",
            "t3.medium"
          ]
        }
      }
    }
  ]
}
```

Prevent Root User Actions

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "*",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:PrincipalType": "Root"
        }
      }
    }
  ]
}
```

Region Restriction Policy

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "*",
      "Resource": "*",
      "Condition": {
        "StringNotEquals": {
          "aws:RequestedRegion": [
            "us-east-1",
            "us-west-2",
            "eu-west-1"
          ]
        }
      }
    }
  ]
}
```

Best Practices for SCPs

- Start with permissive policies and gradually restrict
- Test policies in non-production environments first
- Use condition keys for fine-grained control
- Document policy purposes and exceptions
- Regular review and updates of policies

---

6. Account Management

Account Creation Strategies

Automated Account Creation

- Use AWS Organizations APIs for programmatic creation
- Implement account vending machines for standardized setup

- Apply tags and configurations automatically

Account Naming Conventions

Format: [Environment]-[BusinessUnit]-[Application]-[Region]
Examples:
- prod-finance-erp-us-east-1
- dev-marketing-website-eu-west-1
- shared-security-logging-global

Account Lifecycle Management

Account Setup Process

1. Create account through Organizations
2. Apply appropriate OUs and policies
3. Configure basic security settings
4. Set up monitoring and logging
5. Establish cross-account access roles
6. Apply resource tagging strategy

Account Governance

- Regular access reviews
- Compliance monitoring
- Cost optimization reviews
- Security assessments
- Policy compliance checks

Cross-Account Access Management

Assume Role Strategy

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::MASTER-ACCOUNT-ID:root"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "sts:ExternalId": "unique-external-id"
        }
      }
    }
  ]
}
```

---

7. Billing and Cost Management

Consolidated Billing Benefits

Single Payment Method

- One bill for all accounts in the organization
- Simplified financial management
- Centralized payment processing

Volume Discounts

- Aggregated usage across all accounts
- Qualification for volume pricing tiers
- Shared Reserved Instance benefits
- Savings Plans optimization

Cost Allocation and Tracking

Cost Categories

- Group costs by business unit, project, or environment
- Create custom cost groupings for reporting
- Track spending against budgets

Tagging Strategy

Required Tags:
- Environment: prod/dev/test
- Owner: team-name
- Project: project-identifier
- CostCenter: department-code
- Application: app-name

Budget Management

- Set budgets at organization, OU, or account level
- Configure alerts for spending thresholds
- Monitor usage patterns and trends
- Implement cost controls through SCPs

Cost Optimization Techniques

Reserved Instance Management

- Centralized RI purchasing in master account
- Automatic sharing across member accounts
- Regular utilization reviews

Savings Plans

- Organization-wide Savings Plans coverage
- Optimized commitment strategies
- Cross-account benefit sharing

---

8. Security and Compliance

Security Best Practices

Multi-Account Security Strategy

- Dedicated security account for centralized logging
- Separate audit account for compliance monitoring
- Centralized identity management account

CloudTrail Configuration

```
{
  "TrailName": "OrganizationTrail",
  "S3BucketName": "org-cloudtrail-logs",
  "IsOrganizationTrail": true,
  "EnableLogFileValidation": true,
  "EventSelectors": [
    {
      "ReadWriteType": "All",
      "IncludeManagementEvents": true,
      "DataResources": []
    }
  ]
}
```

Config Rules for Compliance

- Organization-wide Config rules
- Automatic remediation actions
- Compliance dashboards and reporting

Identity and Access Management

Federated Access Strategy

- Single Sign-On (SSO) integration
- Centralized identity provider
- Role-based access control

Permission Boundaries

- Additional guardrails for IAM entities

- Complement SCPs with granular controls
- Prevent privilege escalation

Compliance Framework

Regulatory Requirements

- GDPR compliance across accounts
- HIPAA requirements for healthcare workloads
- SOC 2 compliance monitoring
- PCI DSS for payment processing

Audit Preparation

- Centralized logging and monitoring
- Automated compliance reporting
- Evidence collection and retention
- Regular security assessments

---

9. Implementation Best Practices

Planning Phase

Assessment and Design

1. Inventory existing AWS accounts
2. Define organizational structure
3. Design OU hierarchy
4. Plan policy framework
5. Establish naming conventions

Stakeholder Engagement

- Involve security, finance, and operations teams
- Define roles and responsibilities
- Establish governance processes
- Create training and documentation

Migration Strategies

Greenfield Deployment

- Start with Organizations from the beginning
- Create accounts within organizational structure
- Apply policies and governance from day one

Brownfield Migration

- Invite existing accounts to join organization
- Gradually apply policies and governance
- Minimize disruption to existing workloads

Phased Implementation

Phase 1: Foundation

- Create organization and basic OU structure
- Implement consolidated billing
- Establish basic SCPs

Phase 2: Governance

- Expand policy framework
- Implement security controls
- Add monitoring and compliance

Phase 3: Optimization

- Refine policies based on experience

- Optimize costs and resource usage
- Enhance automation and tooling

Monitoring and Maintenance

Regular Reviews

- Policy effectiveness assessment
- Account usage and compliance audits
- Cost optimization opportunities
- Security posture evaluation

Automation Opportunities

- Account creation workflows
- Policy deployment and updates
- Compliance monitoring and reporting
- Cost allocation and chargeback

---

10. Common Use Cases

Enterprise Multi-Account Strategy

Scenario: Large enterprise with multiple business units, environments, and geographical regions.

Structure:

```
Root
├── Production OU
│   ├── Finance Production
│   ├── HR Production
│   └── Marketing Production
├── Development OU
│   ├── Finance Development
│   ├── HR Development
│   └── Marketing Development
└── Shared Services OU
    ├── Logging Account
    ├── Security Account
    └── Networking Account
```

Benefits:

- Clear separation of concerns
- Environment-specific policies
- Centralized shared services
- Cost allocation by business unit

DevOps and CI/CD Pipeline

Scenario: Software development organization with multiple teams and applications.

Structure:

```
Root
├── Production OU
├── Staging OU
├── Development OU
│   ├── Team Alpha
│   ├── Team Beta
│   └── Team Gamma
└── DevOps OU
    ├── CI/CD Account
    ├── Artifact Repository
    └── Monitoring Account
```

Implementation:

- Automated account provisioning for new projects
- Pipeline-based deployments across environments

- Standardized security and compliance policies

Managed Service Provider (MSP)

Scenario: MSP managing AWS infrastructure for multiple clients.

Structure:

```
Root
├── Client A OU
│   ├── Client A Production
│   └── Client A Development
├── Client B OU
│   ├── Client B Production
│   └── Client B Development
└── MSP Operations OU
    ├── Billing Account
    ├── Monitoring Account
    └── Security Account
```

Benefits:

- Client isolation and security
- Centralized billing and cost management
- Standardized service delivery
- Compliance and audit capabilities

---

11. Troubleshooting and FAQs

Common Issues and Solutions

Account Creation Failures

- **Issue**: Account creation fails with service limit errors
- **Solution**: Request service limit increases for Organizations
- **Prevention**: Monitor account creation limits and plan ahead

Policy Conflicts

- **Issue**: SCPs preventing legitimate actions
- **Solution**: Review policy hierarchy and inheritance
- **Prevention**: Test policies in non-production environments

Billing Discrepancies

- **Issue**: Unexpected charges or allocation errors
- **Solution**: Review cost allocation tags and settings
- **Prevention**: Implement comprehensive tagging strategy

Frequently Asked Questions

Q: Can I move accounts between organizations? A: Yes, but accounts must first leave their current organization before joining a new one. This process affects billing and policy application.

Q: What happens to existing resources when joining an organization? A: Existing resources remain unchanged, but new policies (SCPs) may restrict future actions on those resources.

Q: Can I apply different SCPs to different accounts in the same OU? A: SCPs apply to all accounts within an OU. For account-specific policies, consider creating dedicated OUs or using IAM policies within accounts.

Q: How do I handle emergency access when SCPs block critical actions? A: Plan for emergency procedures using the master account (which isn't restricted by SCPs) or implement break-glass procedures with appropriate approval workflows.

Q: What's the difference between SCPs and IAM policies? A: SCPs set the maximum permissions boundary (what's allowed), while IAM policies grant specific permissions (what's actually accessible).

Monitoring and Alerting

Key Metrics to Monitor:

- Account creation and deletion events
- Policy violations and denied actions
- Cost anomalies and budget breaches
- Security compliance violations

- Resource usage across accounts

Recommended Alerts:

- New account creation
- Policy modification events
- Budget threshold breaches
- Security finding notifications
- Compliance status changes