### **AWS Organizations and Service Control Policies (SCPs)**

### **1. Overview of AWS Organizations**

* AWS Organizations is a service that allows you to centrally manage and govern multiple AWS accounts within your organization.
* It provides features for managing accounts, setting policies, and automating workflows across a large-scale AWS environment.
* AWS Organizations helps you simplify billing, security, and compliance management for multiple accounts.

#### **2. Key Components of AWS Organizations**

* **Organization:**
  + A root entity that contains all AWS accounts under your management.
  + The organization is managed by a master account (now called the **management account**), which has control over the entire organization and can create and manage accounts.
* **Management Account:**
  + The account that creates and manages the organization.
  + It has the highest level of control, including the ability to create accounts, manage policies, and configure billing.
* **Member Accounts:**
  + These are the individual AWS accounts within the organization that are managed by the management account.
  + Member accounts inherit policies from the organization, but they operate independently regarding resources and services.
* **Organizational Units (OUs):**
  + OUs are groupings of AWS accounts within an organization.
  + They allow you to apply policies to specific sets of accounts and organize accounts based on function, department, or project.
  + You can create hierarchical structures with OUs (e.g., parent and child OUs) to apply policies at different levels.
* **Consolidated Billing:**
  + AWS Organizations allows you to consolidate billing across multiple AWS accounts.
  + This provides a single bill for all accounts in the organization and can lead to cost savings through volume discounts and cost allocation tracking.

#### **3. Service Control Policies (SCPs)**

* SCPs are policies that manage permissions for AWS accounts in an organization.
* They act as a guardrail, setting the maximum permissions that can be granted to any entity (users, groups, or roles) within the accounts in your organization.
* SCPs do not grant permissions themselves but define the outer boundaries of permissions that can be assigned.
* **Structure of SCPs:**
  + SCPs are JSON policy documents similar to IAM policies.
  + They include elements like Effect, Action, Resource, and Condition.
  + Example: An SCP can be written to deny all access to the EC2 service ("Action": "ec2:\*"), regardless of any permissions granted within individual accounts.

#### **4. How SCPs Work**

* **Allowlisting and Denylisting:**
  + SCPs can be used to create allowlists, specifying only the actions that are permitted (everything else is denied by default).
  + Alternatively, they can be used to deny specific actions while allowing others.
  + Deny-based SCPs are commonly used to block access to sensitive services or actions across an entire organization (e.g., preventing the deletion of S3 buckets).
* **Inheritance:**
  + SCPs are inherited hierarchically. If an SCP is attached to an OU, all accounts within that OU inherit the policy.
  + Multiple SCPs can be applied at different levels (root, OU, or account), and the most restrictive settings apply.
* **Explicit Deny:**
  + SCPs can explicitly deny actions, overriding any permissions granted within individual IAM policies in an account.
  + An explicit deny in an SCP will prevent the action, even if a user or role has permission to perform that action in their IAM policy.

#### **5. Types of SCPs**

* **Full Access SCP:**
  + This SCP allows all actions in all services by default.
  + Typically, it’s attached to the root of the organization to grant full access, and then additional SCPs are used to restrict permissions as needed.
* **Deny All Except SCP:**
  + This SCP denies access to all services and actions except those explicitly allowed.
  + Useful for restricting accounts or OUs to specific services (e.g., only allowing access to S3 and Lambda).
* **Deny Specific Services SCP:**
  + This SCP denies access to specific services or actions (e.g., preventing users from creating EC2 instances).
  + Ideal for ensuring compliance with organizational security policies by restricting access to high-risk services.

#### **6. Best Practices for Using SCPs**

* **Start with Full Access:**
  + Begin by applying a Full Access SCP at the root of the organization, then add restrictive SCPs at the OU or account level as needed.
  + This approach helps avoid accidentally locking yourself out of critical services.
* **Use SCPs for Guardrails:**
  + SCPs are best used as guardrails to enforce mandatory security controls, such as preventing the use of unapproved regions or blocking access to certain services.
  + SCPs should complement IAM policies by setting boundaries rather than micromanaging permissions.
* **Test SCPs in a Sandbox Environment:**
  + Before applying SCPs broadly, test them in a sandbox or non-production environment to ensure they don’t inadvertently block necessary actions.
  + Use AWS’s Policy Simulator to validate the impact of SCPs.
* **Apply SCPs at the Appropriate Level:**
  + Use hierarchical OUs to group accounts by function or risk level, applying SCPs at the appropriate level.
  + For example, apply restrictive SCPs to OUs containing production accounts and less restrictive SCPs to development environments.
* **Monitor and Audit SCPs:**
  + Regularly review SCPs to ensure they align with evolving security and compliance requirements.
  + Use AWS CloudTrail and AWS Config to monitor changes to SCPs and their effectiveness in enforcing policies.

#### **7. SCPs vs. IAM Policies**

* **Scope of Control:**
  + SCPs control the maximum permissions available across an entire AWS account, whereas IAM policies are applied to specific users, groups, or roles within an account.
  + SCPs take precedence over IAM policies, meaning that if an action is denied by an SCP, it cannot be allowed by an IAM policy.
* **Global Application:**
  + SCPs apply globally across all AWS accounts in an organization, while IAM policies are confined to a specific account.
  + This makes SCPs ideal for enforcing organization-wide security and governance standards.

#### **8. Organizational Units (OUs) and SCPs**

* **OU Structure:**
  + OUs allow you to create a logical hierarchy of accounts. You can organize accounts by department, project, or security level.
  + SCPs can be applied at the OU level, making it easy to manage permissions for all accounts within that OU.
* **Parent and Child OUs:**
  + A parent OU can have child OUs, each with its own SCPs.
  + Child OUs inherit SCPs from their parent OU, but you can also apply additional SCPs to child OUs to further restrict permissions.
* **Use Cases for OUs:**
  + **Security Segmentation:** Group accounts with similar security requirements together (e.g., production vs. development).
  + **Cost Management:** Group accounts by business unit or project for more granular cost tracking and billing management.

#### **9. Cross-Account Management and SCPs**

* **Cross-Account Roles:**
  + SCPs can control the use of cross-account roles, ensuring that only authorized accounts can access specific resources.
  + This is crucial for maintaining security when sharing resources between accounts in different OUs.
* **Delegated Administrators:**
  + AWS Organizations allows you to delegate administrative permissions for certain services (e.g., AWS Config, CloudFormation) to specific accounts within your organization.
  + SCPs can be used to limit the actions of these delegated administrators to ensure compliance with organizational policies.

#### **10. Cost Management and Billing in AWS Organizations**

* **Consolidated Billing:**
  + Consolidated billing aggregates usage across all accounts in the organization, allowing you to maximize volume discounts and manage costs more effectively.
  + Each account still retains its own resources, but all charges are combined into a single invoice.
* **Cost Allocation Tags:**
  + Cost allocation tags can be used to track and allocate costs to different business units, departments, or projects within your organization.
  + AWS Organizations allows you to apply tags at the account level to streamline cost management and reporting.
* **Budgets and Alerts:**
  + You can set up budgets and alerts at the organization level to monitor spending and ensure that no accounts exceed their allocated budget.
  + This is particularly useful for preventing cost overruns in non-production environments.

#### **11. Security and Compliance in AWS Organizations**

* **Service Control Policies for Compliance:**
  + SCPs can enforce compliance by ensuring that all accounts adhere to security standards, such as restricting access to specific AWS regions or services.
  + SCPs can also help enforce data residency requirements by preventing resources from being created outside of approved regions.
* **AWS Security Hub Integration:**
  + AWS Security Hub integrates with AWS Organizations to provide a unified view of security findings across all accounts.
  + You can use Security Hub to monitor compliance with security best practices and ensure that SCPs are effective in enforcing organizational policies.
* **Audit and Monitoring:**
  + AWS CloudTrail can be used to audit all API activity across the organization, providing a comprehensive log of actions taken within member accounts.
  + AWS Config rules can be applied organization-wide to monitor compliance with resource configurations and ensure that SCPs are being enforced.

#### **13. Common Use Cases for AWS Organizations and SCPs**

* **Centralized Security Management:**
  + AWS Organizations and SCPs allow you to enforce security policies across multiple accounts, ensuring that all accounts adhere to your organization’s security standards.
  + Example: Use SCPs to enforce the use of encryption across all accounts or prevent the disabling of security services like GuardDuty or CloudTrail.
* **Account Isolation:**
  + Use OUs and SCPs to create isolated environments for different teams, departments, or projects.
  + Example: Create separate OUs for production, development, and testing environments, each with its own SCPs to control access and enforce security.
* **Cost Management and Budgeting:**
  + Consolidated billing allows you to manage costs across all accounts in your organization, while SCPs can be used to control the use of expensive resources.
  + Example: Use SCPs to restrict the creation of high-cost EC2 instances in non-production environments.
* **Compliance and Regulatory Requirements:**
  + AWS Organizations and SCPs can help ensure compliance with regulatory requirements by enforcing policies at the organization level.
  + Example: Use SCPs to restrict the use of AWS regions that do not meet your data residency requirements.
* **Cross-Account Access Management:**
  + SCPs can control cross-account access by limiting which roles and resources can be accessed by other accounts.
  + Example: Use SCPs to restrict cross-account role assumption to specific accounts or services.
* **Delegated Administration:**
  + AWS Organizations allows you to delegate administrative responsibilities to specific accounts or OUs, reducing the need for centralized management of every task.
  + Example: Delegate administrative control of AWS Config or CloudFormation to specific accounts while maintaining overall governance with SCPs.
* **Operational Efficiency:**
  + Automate account provisioning, policy enforcement, and resource management across multiple accounts using AWS Organizations and SCPs.
  + Example: Automate the creation of new accounts with pre-configured SCPs and resource limits using AWS Control Tower or Lambda.

#### 

#### **14. Security and Compliance Best Practices with AWS Organizations and SCPs**

* **Enable AWS CloudTrail Across All Accounts:**
  + Ensure that CloudTrail is enabled in all accounts to capture and log all API activity. SCPs can enforce this requirement by denying actions that disable CloudTrail.
* **Use AWS Organizations Service Control Policies for Data Protection:**
  + Use SCPs to enforce encryption for data at rest and in transit. You can deny actions that create resources (like S3 buckets) without encryption enabled.
* **Implement Least Privilege Access with SCPs:**
  + SCPs should be designed to enforce the principle of least privilege by allowing only the actions necessary for each account’s function.
* **Monitor Compliance with AWS Security Hub and Config:**
  + Use AWS Security Hub and AWS Config to continuously monitor and evaluate the security and compliance posture of your accounts. SCPs can help enforce these configurations.
* **Enforce Strong Authentication and MFA:**
  + SCPs can be used to enforce multi-factor authentication (MFA) for all accounts, particularly for privileged access to sensitive resources.
* **Limit Root Account Access:**
  + SCPs can restrict actions taken by the root account to prevent accidental or malicious changes to the organization’s configuration.
* **Centralize Logs and Security Data:**
  + Use AWS Organizations to centralize logging and security data in a dedicated account. SCPs can ensure that all accounts send logs to a central location.
* **Regularly Review and Update SCPs:**
  + As your organization grows and evolves, regularly review SCPs to ensure they remain aligned with current security policies, compliance requirements, and business needs.
* **Audit and Track Changes to SCPs:**
  + Use AWS CloudTrail to audit changes to SCPs and ensure that any modifications are approved and documented.

**AWS Control Tower:**

* AWS Control Tower automates the setup and governance of a multi-account environment using AWS Organizations.
* It provides a prescriptive framework for setting up a secure, compliant, and scalable multi-account architecture.
* Control Tower sets up **Landing Zones**, which are pre-configured environments with best-practice blueprints for security, governance, and compliance.
* **Guardrails** are pre-configured policies in Control Tower that implement SCPs, AWS Config rules, and other governance mechanisms to enforce security and compliance across the organization.
* Control Tower helps manage AWS Organizations by simplifying account creation, applying SCPs, and automating account management.