PRJ-DOP-022: Automated Governance & Compliance Framework

Certification: AWS Certified DevOps Engineer – Professional
Domain: Governance and Compliance Automation

---

1. Project Overview

This project demonstrates how to establish a robust framework for automated governance, compliance, and security across a multi-account AWS environment. As an organization scales, managing policies and ensuring compliance in dozens or hundreds of accounts becomes impossible to do manually. This project leverages a suite of AWS governance services to create a centralized, enforceable, and auditable environment.

We will use **AWS Organizations** to manage accounts and apply preventive guardrails with **Service Control Policies (SCPs)**. We will then deploy **AWS Config** rules across all accounts to detect non-compliant resources. Finally, we will build an **automated remediation** workflow using **EventBridge** and **AWS Systems Manager (SSM)** to automatically fix non-compliant resources, moving from a reactive to a proactive compliance posture.

Key Objectives

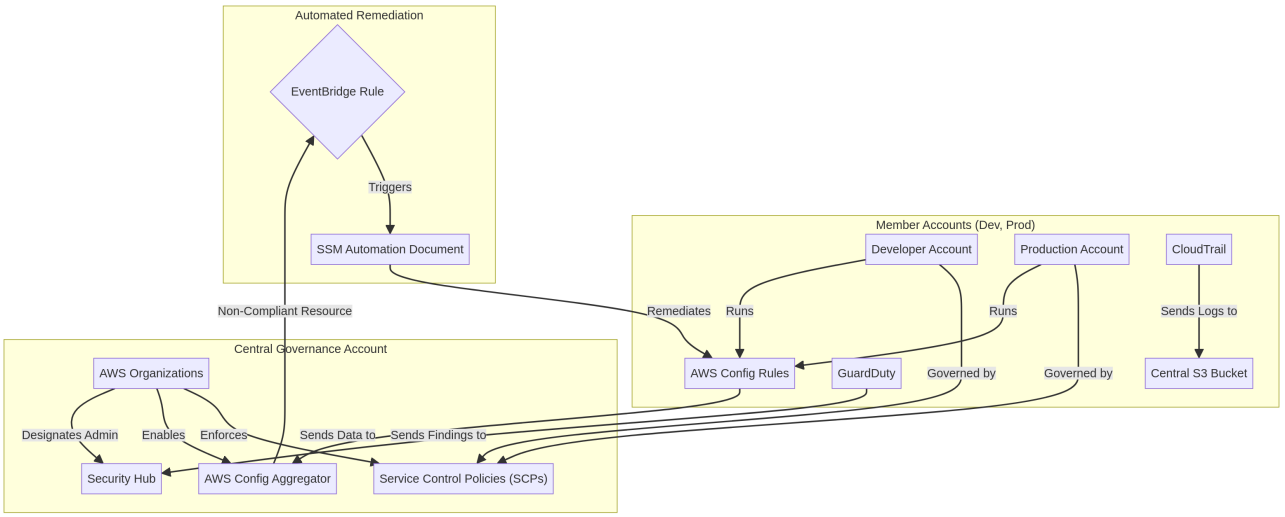
- Set up a multi-account structure using AWS Organizations.
- Implement preventive controls using Service Control Policies (SCPs) to restrict risky actions (e.g., disabling CloudTrail).
- Deploy detective controls using AWS Config rules to continuously monitor resource configurations for compliance.
- Create a central view of compliance status across all accounts using an AWS Config aggregator.

- Implement an automated remediation workflow that triggers an SSM Automation document to fix a non-compliant resource.
- Centralize security findings and logs using AWS Security Hub and a central S3 bucket for CloudTrail.

---

2. Architecture

The architecture is centered around a dedicated “Governance” or “Audit” account that manages and monitors the member accounts.



Framework Components:

1. **Central Governance (AWS Organizations):**
    - **AWS Organizations** is used to create a hierarchy of accounts, grouped into Organizational Units (OUs) (e.g., Prod, Dev, Staging).
    - **Service Control Policies (SCPs)** are attached to OUs or the root of the organization. These are preventive guardrails that define the maximum permissions available to IAM users and roles in the member accounts. For example, an SCP can deny the ability to delete CloudTrail logs or leave the organization.
2. **Detective Controls (AWS Config):**
    - **AWS Config Rules** are deployed in each member account to evaluate the configuration of AWS resources against best practices. Examples include

`s3-bucket-public-read-prohibited` or `encrypted-volumes`.

- A **Config Aggregator**, set up in the central governance account, collects the compliance data from all member accounts, providing a single dashboard to view the compliance status of the entire organization.

3. **Automated Remediation:**
    - When AWS Config finds a non-compliant resource, it generates an event.
    - An **EventBridge rule** in the member account is configured to listen for these non-compliance events.
    - The rule triggers an **SSM Automation document**. This document is a runbook that contains the steps to fix the specific issue (e.g., a script to enable encryption on an unencrypted S3 bucket).
4. **Centralized Logging & Security:**
    - **AWS CloudTrail** is enabled in all accounts, and the logs are configured to be delivered to a single, highly restricted S3 bucket in a dedicated "Log Archive" account.
    - **Amazon GuardDuty** and **AWS Security Hub** are enabled organization-wide, with the central governance account designated as the delegated administrator. This allows security teams to view all threats and security findings from a single console.

---

3. Prerequisites

- An AWS account to act as the management account for AWS Organizations.
- At least one other AWS account to act as a member account.
- Administrative permissions in all accounts.

---

4. Step-by-Step Implementation Guide

Step 4.1: Set Up AWS Organizations and SCPs

1. **Create the Organization:**
    - In your chosen management account, go to the **AWS Organizations console** and **Create an organization**.
    - Invite your other AWS account(s) to join the organization.
    - Accept the invitation from the member account(s).
    - Create an Organizational Unit (OU) named `Workloads` and move the member account into it.
2. **Enable Service Control Policies:**
    - In the Organizations console, go to **Policies -> Service Control Policies** and **Enable SCPs**.
3. **Create and Attach an SCP:**
    - Create a new SCP named `Deny-Leaving-Org`.
    - Use the following JSON policy, which denies the `organizations:LeaveOrganization` action:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "organizations:LeaveOrganization",
      "Resource": "*"
    }
  ]
}
```

    - Attach this SCP to the `Workloads` OU. Now, no user or role in the member account can remove that account from the organization, even if they have

full admin permissions.

Step 4.2: Deploy Detective Controls with AWS Config

1. **Enable AWS Config:** In **each** member account, go to the AWS Config console and enable it. Choose to record all resources.
2. **Deploy a Managed Config Rule:**
    - In a member account, go to **AWS Config -> Rules -> Add rule**.
    - Search for and select the managed rule `s3-bucket-public-read-prohibited`.
    - Save the rule. Config will now evaluate all your S3 buckets against this rule.
3. **Create a Config Aggregator (in Management Account):**
    - Go to the **AWS Config console** in your **management account**.
    - Go to **Aggregators -> Create aggregator**.
    - **Name:** `Organization-Wide-Compliance`
    - **Source accounts:** Choose **Add my organization**.
    - Create the aggregator. It will take some time to pull the initial data. Once complete, you will be able to see the compliance status of all rules in all member accounts from this single view.

Step 4.3: Implement Automated Remediation

We will create a workflow to automatically remediate the `s3-bucket-public-read-prohibited` rule.

1. **Find the Remediation Action:**
    - In the **AWS Config console** (in the member account), select the `s3-bucket-public-read-prohibited` rule.
    - Under **Remediation action**, you will see a recommended action: `AWS-DisableS3BucketPublicReadWrite`. This is a pre-built SSM Automation document.
2. **Configure the Remediation:**

- Click **Manage remediation**.
- **Remediation method:** Automatic remediation.
- **Remediation action:** `AWS-DisableS3BucketPublicReadWrite`.
- **Parameters:** The `S3BucketName` parameter will be automatically filled by Config with the name of the non-compliant bucket.
- **Resource ID parameter:** `S3BucketName`.
- You will need to create an IAM role that grants SSM permissions to perform the remediation (in this case, `s3:PutBucketPublicAccessBlock`).
- Save the remediation.

Step 4.4: Test the Entire Workflow

1. **Create a Non-Compliant Resource:**
   - In the **member account**, go to the S3 console and create a new bucket.
   - Go to the **Permissions** tab and **Edit** the **Block public access** settings.
   - **Uncheck** "Block all public access" and save the changes. This makes the bucket non-compliant.
2. **Observe Detection and Remediation:**
   - **Detection:** Within a few minutes, **AWS Config** will re-evaluate the rule and mark the new bucket as **Non-compliant**.
   - **Remediation:** The non-compliant finding will trigger the automatic remediation action you configured.
   - The **SSM Automation document** `AWS-DisableS3BucketPublicReadWrite` will be executed.
   - **Verification:** Go back to the S3 bucket's permissions. You will see that the **Block public access** settings have been automatically re-enabled. The resource is now compliant.
   - The compliance status in AWS Config will update to **Compliant**.

---

5. Best Practices

- **Start with Detective, then move to Preventive:** It's often best to start by deploying Config rules to detect issues (detective controls) before implementing restrictive SCPs (preventive controls) that might break existing workflows.
- **Test Remediation Carefully:** Always test automated remediation actions in a non-production account first. A misconfigured remediation script can cause significant disruption.
- **Use Conformance Packs:** AWS Config Conformance Packs provide a way to deploy a collection of Config rules and remediation actions as a single entity, often mapped to a specific compliance standard like PCI-DSS or HIPAA.
- **Least Privilege for Remediation:** The IAM roles used for remediation should have the absolute minimum permissions required to fix the specific issue.

---

6. Cleanup

1. **Delete the non-compliant S3 bucket** you created for testing.
2. **Remove the automatic remediation** configuration from the AWS Config rule.
3. **Delete the AWS Config rule.**
4. **Delete the Config aggregator** in the management account.
5. **Remove the SCP** from the OU in AWS Organizations.
6. If desired, you can remove the member account from the organization and then delete the organization itself.

Business Context

The Problem

Development teams face slow deployment cycles, manual testing bottlenecks, and security vulnerabilities introduced during the CI/CD process. Traditional deployment methods lack security scanning, leading to vulnerabilities reaching production. Manual processes create inconsistency and human error.

The Solution

Secure CI/CD pipeline with integrated security scanning, automated testing, and infrastructure as code. Implements shift-left security with SAST, DAST, and dependency scanning. Automates deployments while enforcing security gates and compliance checks at every stage.

Business Value

- **Faster Time to Market:** Automated pipelines reduce deployment time from weeks to hours
- **Improved Security Posture:** Catches vulnerabilities before production deployment
- **Reduced Manual Effort:** Eliminates 80%+ of manual deployment tasks
- **Audit Trail:** Complete deployment history and security scan results for compliance

Risk Mitigation

Prevents deployment of vulnerable code, hardcoded secrets, misconfigured infrastructure, and non-compliant resources to production environments.

GRC Mapping

Compliance Frameworks

- **NIST CSF:** PR.IP-1 (Baseline configuration), PR.DS-6 (Integrity checking), DE.CM-4 (Code analysis)
- **ISO 27001:** A.12.1 (Operational procedures), A.14.2 (Security in development), A.12.4 (Logging)
- **CIS Controls:** Control 16 (Application Software Security), Control 11 (Secure Configuration)
- **OWASP DevSecOps:** Security testing integration, Secret management, Dependency checking

Security Controls Implemented

- Static Application Security Testing (SAST)
- Dynamic Application Security Testing (DAST)
- Software Composition Analysis (SCA)
- Secrets scanning and prevention
- Infrastructure as Code security validation

Audit Evidence

- Pipeline execution logs with security scan results
- Code commit and approval records
- Security test reports (SAST/DAST/SCA)
- Deployment approval and rollback records

Regulatory Alignment

- **SOX:** Section 404 (Change management controls)
- **PCI DSS:** Requirement 6.3 (Secure development), Requirement 6.5 (Common vulnerabilities)
- **GDPR:** Article 25 (Data protection by design), Article 32 (Security of processing)
- **SOC 2:** CC8.1 (Change management), CC7.2 (System monitoring)