**Enforce and Monitor S3 Compliance Across AWS Accounts with AWS Config, Aggregators, and CloudWatch**

**Author**: Gabin Sime  
**Tags**: AWS, DevOps, Security, Automation, AWS Config, S3, Remediation, IAM

**Introduction**

Security in the cloud is a shared responsibility, and one of the most common misconfigurations we still see is publicly exposed Amazon S3 buckets. While AWS provides many tools to help monitor configurations, the real power comes when you can **automatically remediate non-compliant resources**.

In this post, we’ll walk through how to use **AWS Config** in combination with **AWS Systems Manager (SSM)** and **IAM** to automatically block public access to any S3 bucket that violates your compliance policies.

**Architecture Overview**

Here's what we’ll implement:

1. Create a secure IAM role for automation.
2. Provision a purposely non-compliant S3 bucket.
3. Enable AWS Config with a managed rule.
4. Attach an automatic remediation action.
5. Test the setup to ensure it works as expected.

**Step 1: Create an IAM Role for Automation**

We’ll begin by creating an IAM role that AWS Config and SSM can use to remediate S3 bucket permissions.

**Create Role via Console**

1. Go to **IAM > Roles** and click **Create role**.
2. Choose **Trusted entity**: AWS service, then select **Systems Manager** as the use case.
3. In **Permissions**, attach the AmazonSSMAutomationRole policy.
4. Click **Next**, and under **Permissions boundary**, choose:
   * **Use a permissions boundary**
   * Attach the LabSecureAccess policy.
5. Name the role AmazonSSMAutomationRole and click **Create role**.

**Add Inline Policy to Manage S3**

1. Open the newly created AmazonSSMAutomationRole.
2. Under the **Permissions** tab, click **Add permissions** > **Create inline policy**.
3. Select the **S3** service.
4. Under **Access level**:
   * Enable all **List actions**.
   * Enable all **Permissions management actions**.
5. Apply to **All resources**, then click **Next**.
6. Name the policy S3ManagePermissions and click **Create policy**.

**Step 2: Create a Noncompliant S3 Bucket**

To test the rule, we’ll intentionally create an insecure bucket.

1. Navigate to **S3 > Create bucket**.
2. Name it testbucket-52098983 (or any unique name).
3. Region: us-east-2 (Ohio)
4. **Uncheck** “Block all public access.”
5. Acknowledge the warning and click **Create bucket**.

This bucket now violates best practices and will be used to trigger our AWS Config rule.

**Step 3: Enable AWS Config**

To monitor and enforce compliance, we must enable AWS Config.

1. Go to **AWS Config** in the Ohio region (us-east-2).
2. Use **1-click setup** to enable it (which sets up default resources for logging and recording).
3. Once setup is complete, go to **Rules** and click **Add rule**.
4. Search and select the rule: s3-bucket-level-public-access-prohibited.
5. Click through **Next** and **Save**.

At this point, AWS Config will evaluate all buckets against this rule, but it won’t take any actions yet.

**Step 4: Set Up Automatic Remediation**

To enforce the rule, let’s define an **automatic remediation action** using a pre-built SSM automation document.

1. On the **Rules** page, select the s3-bucket-level-public-access-prohibited rule.
2. Click **Actions** > **Manage remediation**.
3. Set the **Remediation method** to: Automatic remediation.
4. In **Remediation action**, search for:
   * AWSConfigRemediation-ConfigureS3BucketPublicAccessBlock
5. Configure parameters:
   * BucketName: testbucket-52098983
   * AutomationAssumeRole:  
     arn:aws:iam::<YOUR\_ACCOUNT\_ID>:role/AmazonSSMAutomationRole
6. Click **Save changes**.

💡 You can find your account ID in the upper-right corner of the AWS console.

**Step 5: Test the Automatic Remediation**

Now, let’s test the setup to see automatic remediation in action.

1. Return to the **Rules** tab in AWS Config.
2. Select s3-bucket-level-public-access-prohibited and click **Actions > Re-evaluate**.
3. After ~2-3 minutes, refresh the page.
4. The bucket testbucket-52098983 should now show as **Compliant**.

**Verify Remediation**

To confirm the S3 bucket has been remediated:

1. Go to **S3** and open the testbucket-52098983.
2. Click the **Permissions** tab.
3. Check that **Block all public access** is turned **On**.

**Step 6: Enforce Across Accounts Using AWS Config Aggregators**

If you're operating in a multi-account AWS Organization, you can centralize rule enforcement and compliance tracking using **AWS Config Aggregators**.

**Create an Aggregator**

1. Navigate to **AWS Config > Aggregators** in the **master (management) account**.
2. Click **Create aggregator**.
3. Name your aggregator, e.g., OrganizationAggregator.
4. Choose **Aggregator source**:
   * Select **Another AWS account or AWS Organization**.
   * For most org-wide setups, choose **AWS Organization**.
   * Ensure you have enabled **trusted access for AWS Config** in AWS Organizations.
5. Choose regions:
   * Select **Include all regions** for global visibility.
6. Click **Create aggregator**.

Once created, you’ll get a centralized dashboard showing all AWS Config rule evaluations from your member accounts.

💡 If the s3-bucket-level-public-access-prohibited rule is not deployed in the other accounts, consider using **StackSets** or **Control Tower** to propagate it consistently.

**Step 7: Set Up CloudWatch Alarms for Remediation Notifications**

To increase observability, let’s configure CloudWatch to alert you whenever automatic remediation occurs.

**Step 7.1: Create a CloudWatch Log Group**

SSM Automation execution logs are published under:

bash

/aws/ssm/AWSConfigRemediation-ConfigureS3BucketPublicAccessBlock

Ensure this log group exists by running a test remediation or manually creating it.

**Step 7.2: Create Metric Filter**

1. Go to **CloudWatch > Log groups**.
2. Select the log group /aws/ssm/AWSConfigRemediation-ConfigureS3BucketPublicAccessBlock.
3. Click **Create metric filter**.
4. Define the pattern, e.g.:

json

CopyEdit

{ $.status = "Success" }

1. Assign a name like RemediationSuccessFilter, and choose a namespace (e.g., RemediationMetrics).
2. Set metric name: S3PublicBlockRemediated, with a value of 1.

**Step 7.3: Create Alarm**

1. Go to **CloudWatch > Alarms**.
2. Click **Create alarm**.
3. Select the custom metric from namespace RemediationMetrics > S3PublicBlockRemediated.
4. Set threshold:
   * When metric is **>= 1** within a 5-minute window.
5. Set up notification:
   * Choose an existing SNS topic or create one (e.g., RemediationAlertsTopic).
   * Subscribe your email or Slack via HTTPS webhook.

**Wrapping Up**

With these two advanced additions, your setup is now:

* Enforcing public S3 bucket security **across your entire AWS Organization**
* Alerting your team in real-time when **remediation actions are triggered**

This makes your infrastructure not just compliant, but **secure, automated, and observable**—hallmarks of mature DevSecOps practices.

**Conclusion**

With AWS Config, SSM Automation, and CloudWatch, you can automatically remediate public S3 buckets and get real-time alerts—all without manual intervention. By adding Config Aggregators, you extend this protection across your organization. This setup ensures your cloud stays secure, compliant, and self-healing at scale.