Automated S3 Remediation to Enforce Block Public Access

Scenario

This implementation ensures that AWS automatically enforces S3 Block Public Access settings across all S3 buckets. If Block Public Access is disabled on any bucket due to unintentional modifications or unauthorized actions, the system will detect the non-compliant configuration and restore the secure settings automatically.

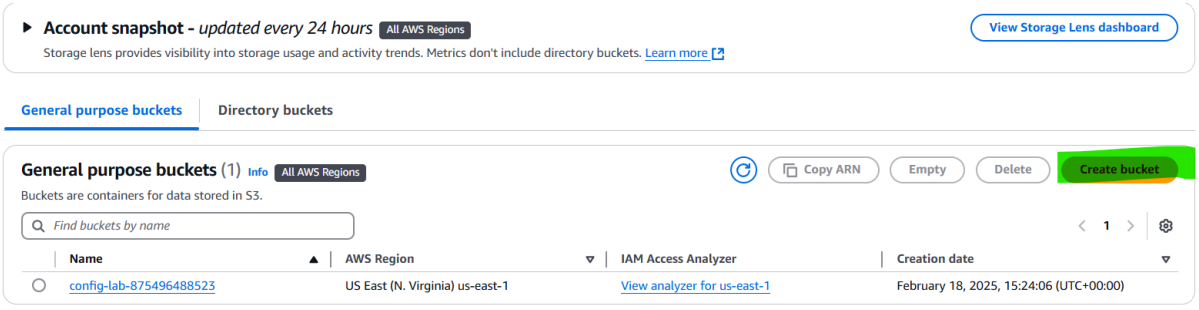
As a security best practice, S3 buckets should always start as private. Public access should only be allowed in specific cases, such as hosting a public website. Ideally, private and public buckets should be kept in separate AWS accounts to enforce settings like Block Public Access at the account level.

For this demonstration, we will implement Block Public Access at the bucket level.

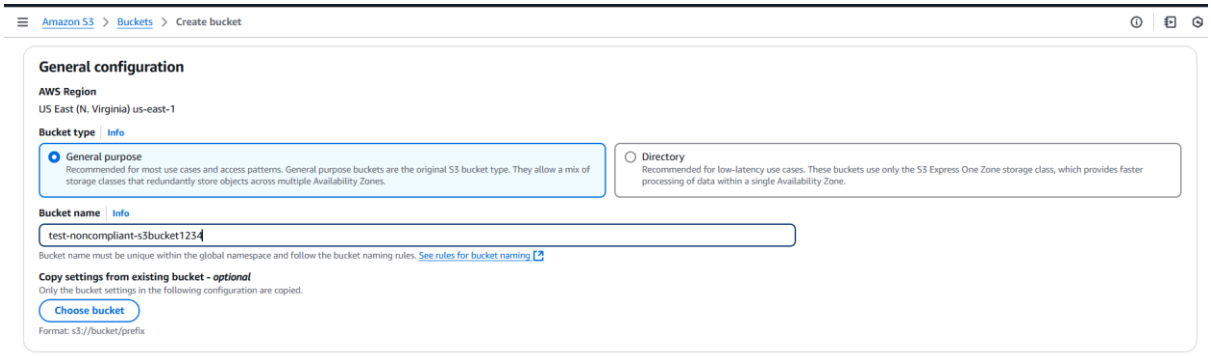
Steps to Implement

1. Log in to AWS Console

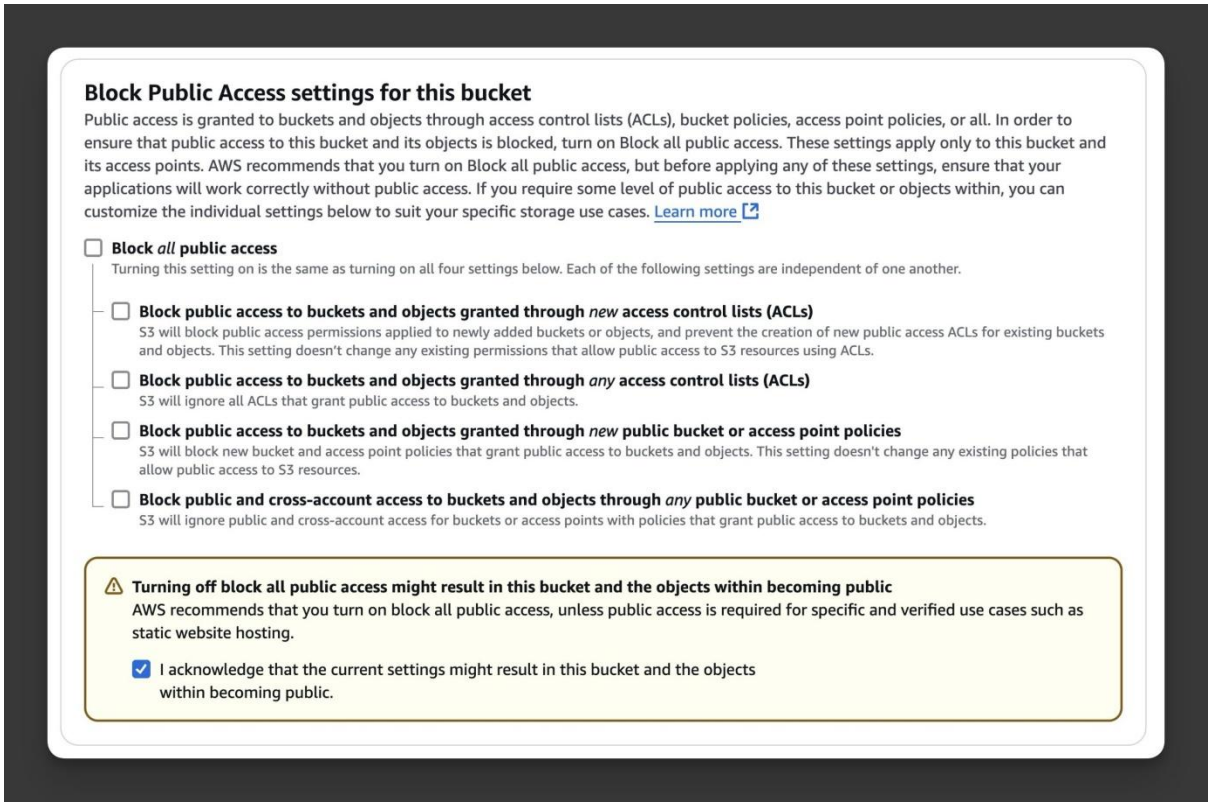
2. Create a Non-Compliant S3 Bucket



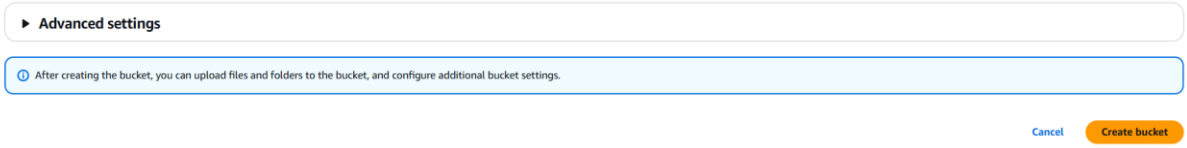
- Create an S3 bucket with a unique name, e.g., test-noncompliant-s3bucket-1234.



- Disable "Block Public Access" by unchecking the checkbox.

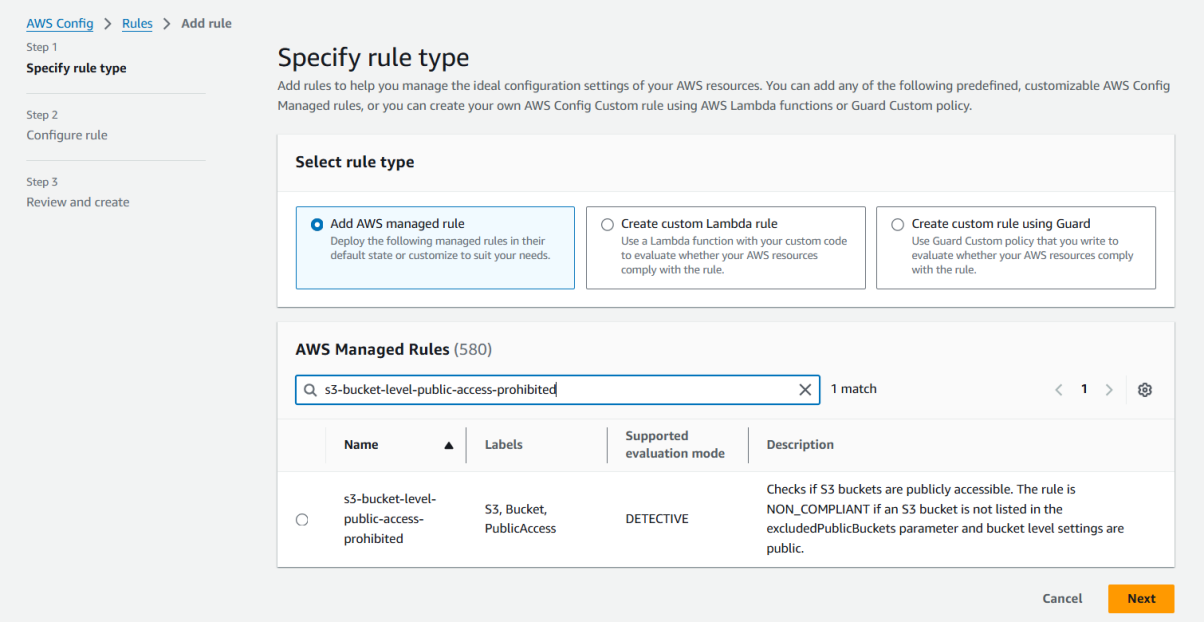


- Confirm the settings and create the bucket.

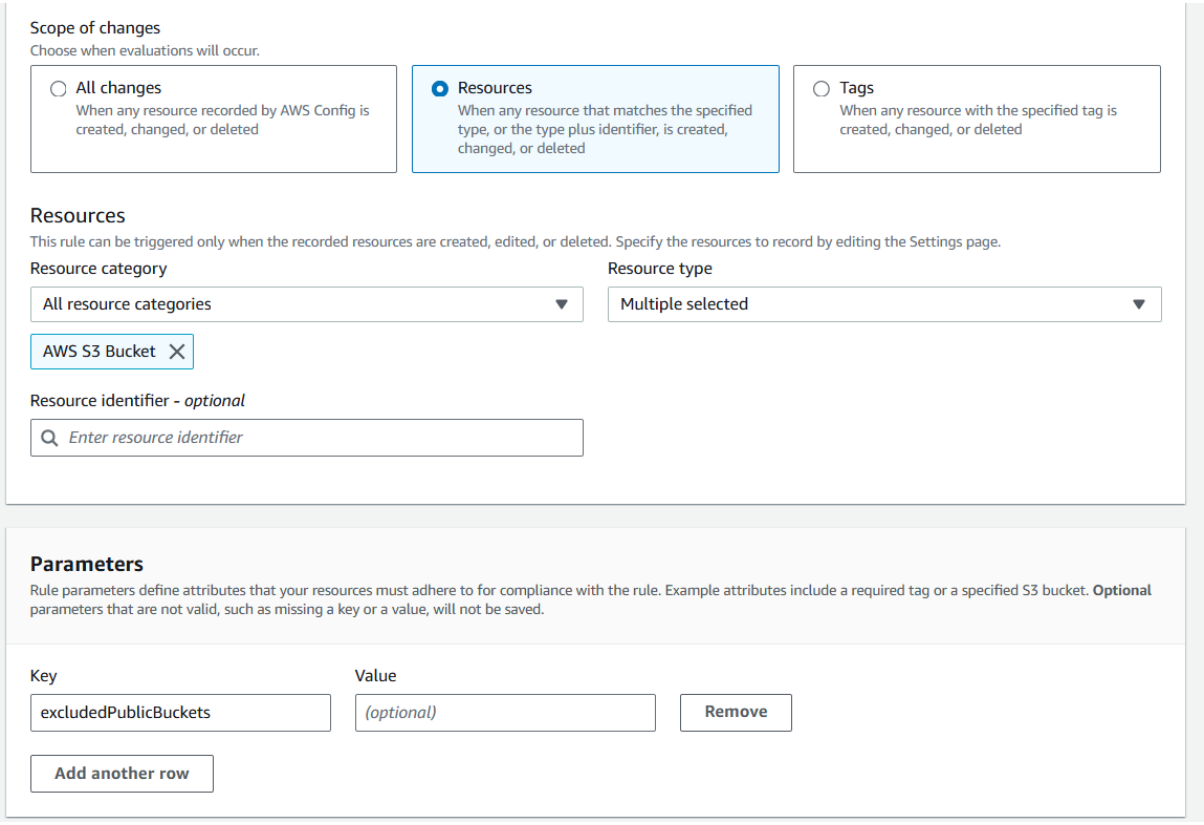


3. Create an AWS Config Rule

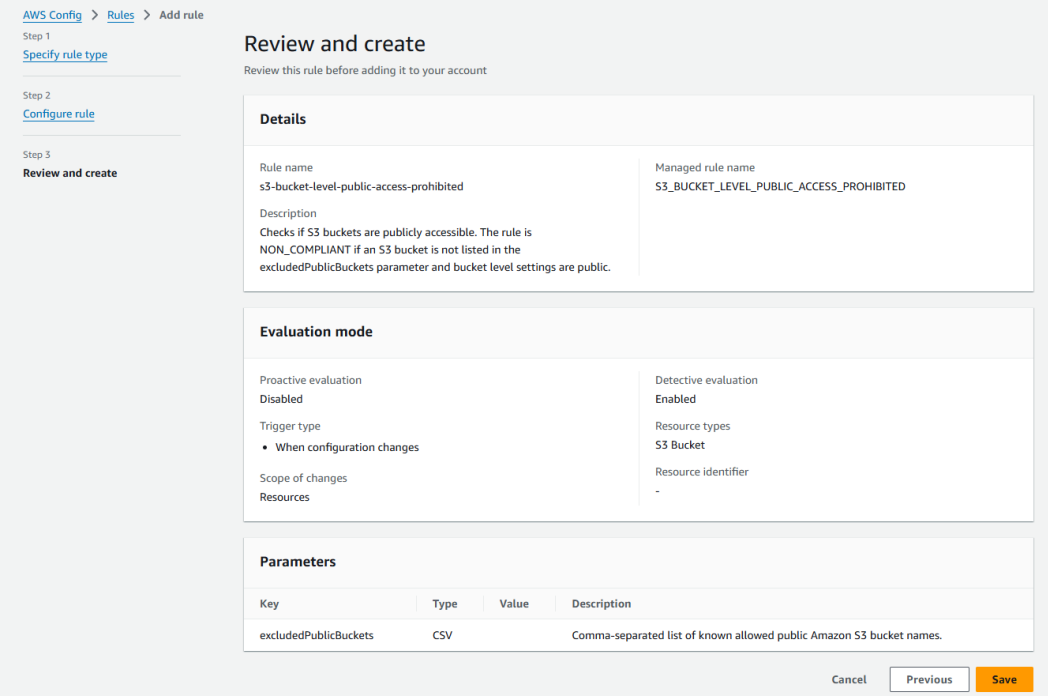
1. Navigate to **AWS Config → Rules → Add Rule**.
2. Select **Add AWS Managed Rule** and search for s3-bucket-level-public-access-prohibited.



3. Under **Evaluation mode**, select Resources to track AWS resource changes.
4. For **Resource category**, choose AWS resources.

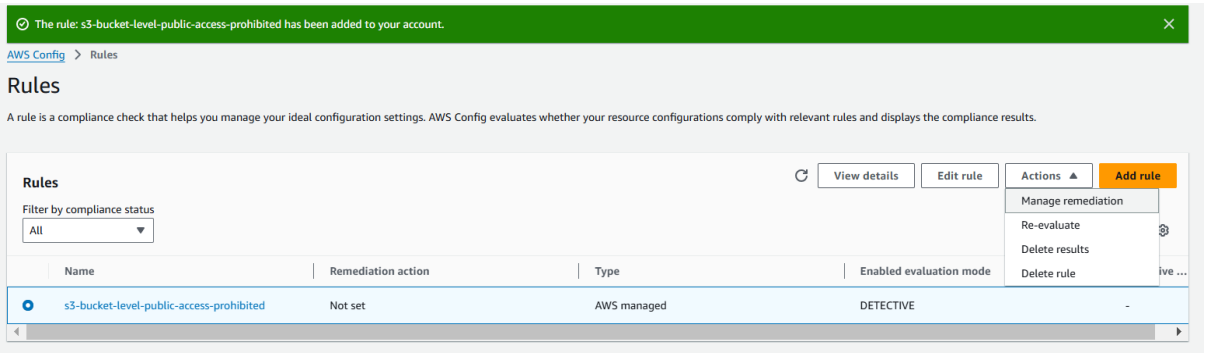


5. Select **AWS S3 Bucket** as the **Resource type**.
6. (Optional) Use **excludedPublicBuckets** to exclude specific public buckets if needed.
7. Click **Next** and then **Save**.

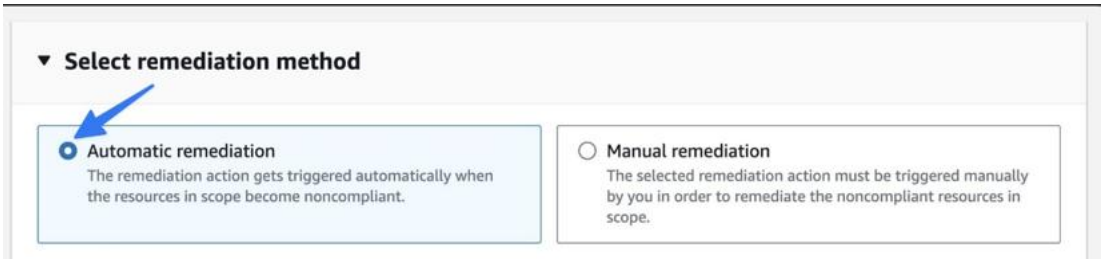


4. Enable Automated Remediation

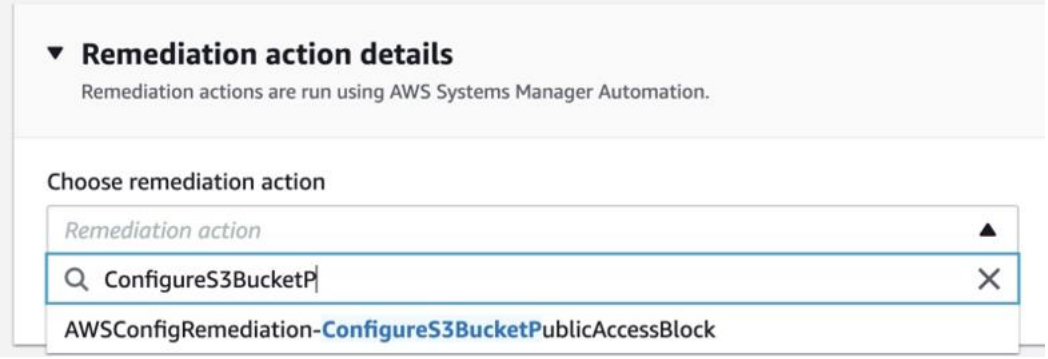
1. Select the created rule and go to **Actions → Manage remediation**.



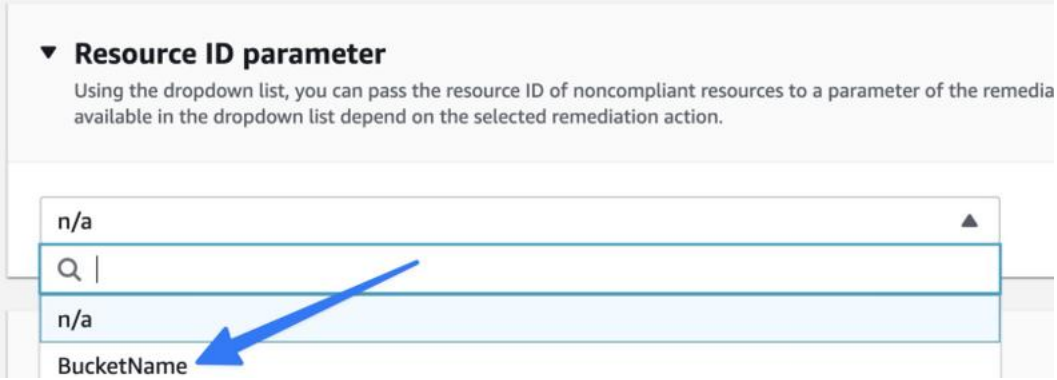
2. Choose **Automatic remediation**.



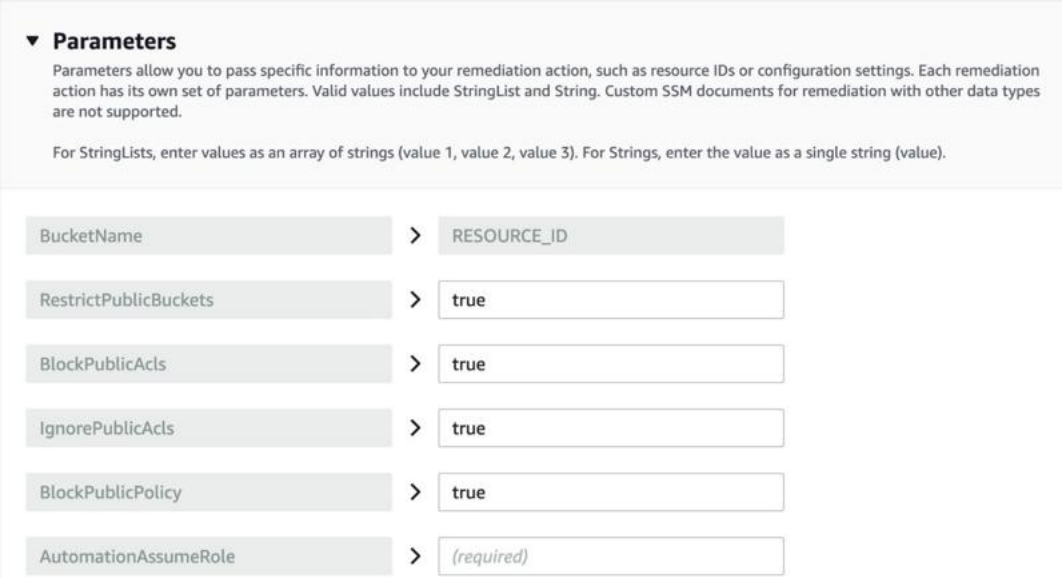
3. Under **Choose remediation action**, search and select **AWSConfigRemediation-ConfigureS3BucketPublicAccessBlock**.



4. For **Resource ID parameter**, select BucketName.



5. Configure parameters:
    - Set all values to true.
    - Provide an **AutomationAssumeRole ARN**.



5. IAM Role for Remediation

This role must have the following permissions:

Search for “AWSConfigRemediation-ConfigureS3BucketPublicAccessBlock” with your search engine and you’ll find this page.

Towards the bottom, you’ll see “Required IAM permissions” which tells us we need:

- ssm:StartAutomationExecution
- ssm:GetAutomationExecution
- s3:GetAccountPublicAccessBlock
- s3:PutAccountPublicAccessBlock
- s3:GetBucketPublicAccessBlock
- s3:PutBucketPublicAccessBlock

In terms of the JSON IAM policy, this is what that would translate to:

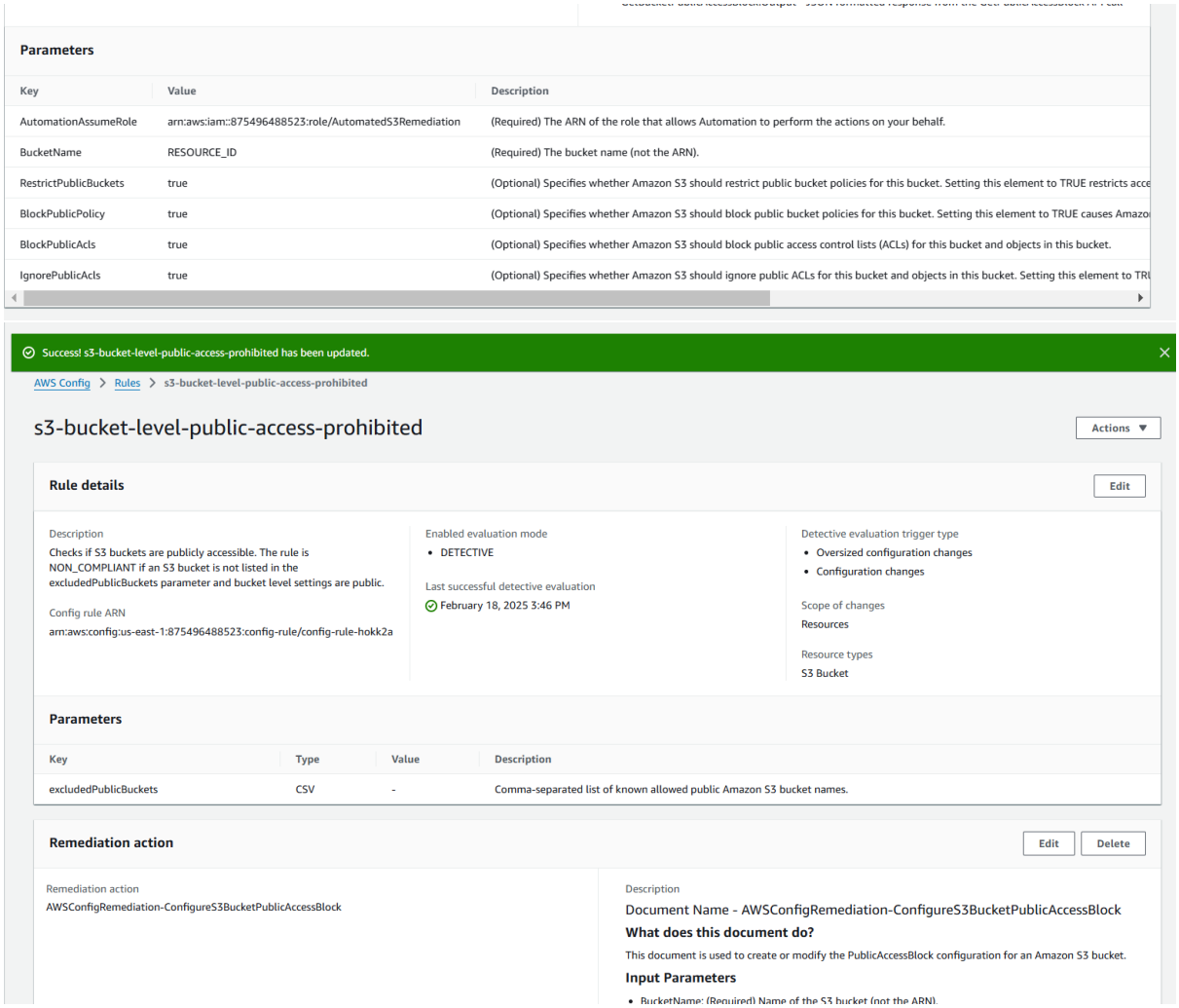
C: > Users > shaik > {} imrole.json > [ ] Statement

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:StartAutomationExecution",
                "ssm:GetAutomationExecution"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetAccountPublicAccessBlock",
                "s3:PutAccountPublicAccessBlock"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketPublicAccessBlock",
                "s3:PutBucketPublicAccessBlock"
            ],
            "Resource": "arn:aws:s3:::*"
        }
    ]
}
```

Use the following ARN format (replace <Account ID> with your AWS account ID):

arn:aws:iam::<Account ID>:role/AutomatedS3Remediation

ARN for the role created by my account id
“arn:aws:iam::875496488523:role/AutomatedS3Remediation”



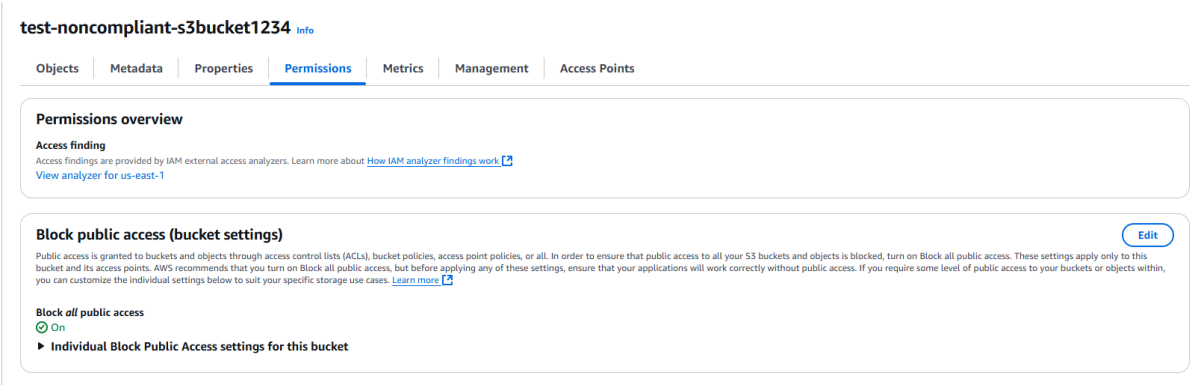
6. Testing Automated Remediation

1. Go to the **AWS Config Rule** page and check **Resources in scope**.

2. If the bucket is not listed as non-compliant, wait a few minutes or click **Re-evaluate**.
3. AWS Config will automatically remediate the issue within 10-30 minutes.
4. Once remediation is successful, the bucket should be marked as compliant.

5. Check the **Resource Timeline** in AWS Config to verify compliance changes.

6. Confirm in **S3 settings** that Block Public Access is enabled.



Conclusion & Next Steps

This setup ensures that any unauthorized attempt to disable Block Public Access on an S3 bucket will automatically be reverted, providing a strong security mechanism.

Next Steps:

1. Implement this setup using **Infrastructure as Code (IaC)** with Terraform or AWS CloudFormation.
2. Extend the remediation to enforce **account-level Block Public Access** instead of just at the bucket level for enhanced security.

References:

Amazon Web Services (AWS), n.d. *AWS Config Developer Guide*. Available at: https://docs.aws.amazon.com/config/ [Accessed 18 Feb. 2025].

Amazon Web Services (AWS), n.d. *Automated S3 Remediation to Enforce Block Public Access*. Available at: https://docs.aws.amazon.com/systems-manager-automation-runbooks/latest/userguide/automation-aws-block-public-s3.html [Accessed 18 Feb. 2025].

Amazon Web Services (AWS), n.d. *Amazon S3 User Guide*. Available at: https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html [Accessed 18 Feb. 2025].

Cybr, n.d. *Introduction to AWS Security - Automated S3 Remediation to Enforce Block Public Access*. Available at: https://cybr.com/courses/introduction-to-aws-security/lessons/lab-automated-s3-remediation-to-enforce-block-public-access/ [Accessed 18 Feb. 2025].