Table of Contents

SCP-ACCOUNT-1

Prevent account region enable and disable actions

Rationale

- Restrict enabling or disabling regions for an account to an infrastructure automation framework role and/or administrator role

References

- https://docs.aws.amazon.com/general/latest/gr/rande-manage.html

Test Scenarios

	Test Scenario	Steps	Expected Result

	Test Scenario	Steps	Expected Result
1	Enable new region	1. Log in to the AWS console with a role that is not the INFRASTRUCTURE_AUTOMATION_ROLE in the statement but has account access 2. Enable a new region	Access Denied

Example SCP Statement

```
{
  "Effect": "Deny",
  "Action": [
        "account:EnableRegion",
        "account:DisableRegion"
    ],
  "Resource": [
        "*"
    ],
  "Condition": {
        "ArnNotLike": {
            "aws:PrincipalARN": [
                "arn:aws:iam::*:role/[INFRASTRUCTURE_AUTOMATION_ROLE]"
            ]
        }
    }
}
```

SCP-BILLING-1

Prevent billing modification actions

Rationale

- Restrict billing modification actions to an infrastructure automation framework role and/or administrator role

References

- https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/getting-viewing-bill.html

Test Scenarios

	Test Scenario	Steps	Expected Result

	Test Scenario	Steps	Expected Result
1	Modify billing configuration	1. Log in to the AWS console with a role that is not the INFRASTRUCTURE_AUTOMATION_ROLE in the statement but has aws-portal access 2. Modify billing configurations	Access Denied

Example SCP Statement

```
{
  "Effect": "Deny",
  "Action": [
        "aws-portal:ModifyAccount",
        "aws-portal:ModifyBilling",
        "aws-portal:ModifyPaymentMethods"
    ],
  "Resource": [
        "*"
    ],
  "Condition": {
        "ArnNotLike": {
            "aws:PrincipalARN": [
                "arn:aws:iam::*:role/[INFRASTRUCTURE_AUTOMATION_ROLE]"
            ]
        }
    }
}
```

SCP-CLOUDFORMATION-1

Prevent modifications to specific CloudFormation resources

Rationale

- Restrict CloudFormation actions to specific CloudFormation Stacks and StackSets that were created by an infrastructure automation framework

References

-

Test Scenarios

	Test Scenario	Steps	Expected Result

	Test Scenario	Steps	Expected Result
1	Modify protected CloudFormation Stack	1. Log in to the AWS console with a role that is not the INFRASTRUCTURE_AUTOMATION_ROLE in the statement but has CloudFormation access 2. Modify a parameter on one of the restricted CloudFormation stacks	Access Denied

Example SCP Statement

```
{
  "Effect": "Deny",
  "Action": [
        "cloudformation:CreateChangeSet",
        "cloudformation:CreateStack",
        "cloudformation:CreateStackInstances",
        "cloudformation:CreateStackSet",
        "cloudformation:CreateUploadBucket",
        "cloudformation:DeleteChangeSet",
        "cloudformation:DeleteStack",
        "cloudformation:DeleteStackInstances",
        "cloudformation:DeleteStackSet",
        "cloudformation:DetectStackDrift",
        "cloudformation:DetectStackResourceDrift",
        "cloudformation:DetectStackSetDrift",
        "cloudformation:ExecuteChangeSet",
        "cloudformation:SetStackPolicy",
        "cloudformation:StopStackSetOperation",
        "cloudformation:UpdateStack",
        "cloudformation:UpdateStackInstances",
        "cloudformation:UpdateStackSet",
        "cloudformation:UpdateTerminationProtection"
    ],
  "Resource": [
        "arn:aws:cloudformation:*:*:stackset/[STACKSET_PREFIX]*",
        "arn:aws:cloudformation:*:*:stack/[STACK_PREFIX]*",
        "arn:aws:cloudformation:*:*:stack/[STACK_NAME]"
    ],
  "Condition": {
        "ArnNotLike": {
            "aws:PrincipalARN": [
                "arn:aws:iam::*:role/[INFRASTRUCTURE_AUTOMATION_ROLE]"
            ]
        }
    }
}
```

SCP-CLOUDTRAIL-1

Prevent modifications to specific CloudTrails

Rationale

- Restrict CloudTrail actions to specific CloudTrails that are required by the security or compliance teams

References

- https://docs.aws.amazon.com/awscloudtrail/latest/userguide/best-practices-security.html

Test Scenarios

	Test Scenario	Steps	Expected Result
1	Disable CloudTrail	1. Log in to the AWS console with a role that is not the INFRASTRUCTURE_AUTOMATION_ROLE in the statement but has CloudTrail access 2. Stop logging on the specified CloudTrail	Access Denied

Example SCP Statement

```
{
  "Effect": "Deny",
  "Action": [
        "cloudtrail:DeleteTrail",
        "cloudtrail:PutEventSelectors",
        "cloudtrail:StopLogging",
        "cloudtrail:UpdateTrail"
    ],
  "Resource": [
        "arn:aws:cloudtrail:${Region}:${Account}:trail/[CLOUDTRAIL_NAME]"
    ],
  "Condition": {
        "ArnNotLike": {
            "aws:PrincipalARN": [
                "arn:aws:iam::*:role/[INFRASTRUCTURE_AUTOMATION_ROLE]"
            ]
        }
    }
}
```

SCP-CLOUDWATCH-1

Prevent deleting specific CloudWatch Log groups and streams

Rationale

- Security policies require that CloudWatch logs are retained for forensic investigations

References

- https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/security.html

Test Scenarios

	Test Scenario	Steps	Expected Result
1	Delete log stream in protected log group	1. Log in to the AWS console with a role that is not the INFRASTRUCTURE_AUTOMATION_ROLE in the statement but has access to CloudWatch Logs 2. Delete an old log stream in one of the protected log groups	Access Denied

Example SCP Statement

```
{
  "Effect": "Deny",
  "Action": [
        "logs:DeleteLogGroup",
        "logs:DeleteLogStream"
    ],
  "Resource": [
        "arn:aws:logs:*:*:log-group:[LOG_GROUP_PREFIX]*"
    ],
  "Condition": {
        "ArnNotLike": {
            "aws:PrincipalARN": [
                "arn:aws:iam::*:role/[INFRASTRUCTURE_AUTOMATION_ROLE]"
            ]
        }
    }
}
```

SCP-CONFIG-1

Prevent enabling and disabling AWS Config

Rationale

- Restrict enabling/disabling AWS Config to an infrastructure automation framework

References

- https://aws.amazon.com/controltower/
- https://aws.amazon.com/solutions/aws-landing-zone/

Test Scenarios

	Test Scenario	Steps	Expected Result
1	Delete configuration recorder	1. Log in to the AWS console with a role that is not the INFRASTRUCTURE_AUTOMATION_ROLE in the statement but has AWS Config access 2. Delete the configuration recorder	Access Denied

Example SCP Statement

```
{
  "Effect": "Deny",
  "Action": [
        "config:DeleteConfigurationAggregator",
        "config:DeleteConfigurationRecorder",
        "config:DeleteDeliveryChannel",
        "config:DeleteRetentionConfiguration",
        "config:PutConfigurationAggregator",
        "config:PutConfigurationRecorder",
        "config:PutDeliveryChannel",
        "config:PutRetentionConfiguration",
        "config:StopConfigurationRecorder"
    ],
  "Resource": [
        "*"
    ],
  "Condition": {
        "ArnNotLike": {
            "aws:PrincipalARN": [
                "arn:aws:iam::*:role/[INFRASTRUCTURE_AUTOMATION_ROLE]"
            ]
        }
    }
}
```

SCP-CONFIG-2

Prevent modifications to tagged AWS Config rules

Rationale

- Restrict enabling/disabling AWS Config except for an infrastructure automation framework role

References

- https://aws.amazon.com/controltower/
- https://aws.amazon.com/solutions/aws-landing-zone/

Test Scenarios

	Test Scenario	Steps	Expected Result
1	Update protected AWS Config rule	1. Log in to the AWS console with a role that is not the INFRASTRUCTURE_AUTOMATION_ROLE in the statement but has AWS Config access 2. Update a config rule that is tagged with the system tag	Access Denied

Example SCP Statement

```
{
  "Effect": "Deny",
  "Action": [
        "config:DeleteConfigRule",
        "config:PutConfigRule",
        "config:TagResource",
        "config:UntagResource"
    ],
  "Resource": [
        "*"
    ],
  "Condition": {
        "ArnNotLike": {
            "aws:PrincipalARN": [
                "arn:aws:iam::*:role/[INFRASTRUCTURE_AUTOMATION_ROLE]"
            ]
        },
        "StringEquals": {
            "aws:ResourceTag/system": "[SYSTEM_NAME]"
        }
    }
}
```

SCP-EC2-1

Prevent disabling default EBS encryption

Rationale

- Security policies require that all EBS volumes are encrypted by default

References

- https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSEncryption.html

Test Scenarios

	Test Scenario	Steps	Expected Result

	Test Scenario	Steps	Expected Result
1	Modify default EBS encryption setting	1. Log in to the AWS console with a role that is not the ALLOWED_ROLE_NAME in the statement but has access to EC2 2. Go to EC2 settings and uncheck the 'Always encrypt new EBS volumes' 3. Save	Access Denied

Example SCP Statement

```
{
  "Effect": "Deny",
  "Action": [
        "ec2:DisableEbsEncryptionByDefault"
    ],
  "Resource": [
        "*"
    ],
  "Condition": {
        "ArnNotLike": {
            "aws:PrincipalARN": [
                "arn:aws:iam::*:role/[ALLOWED_ROLE_NAME]"
            ]
        }
    }
}
```

SCP-EC2-2

Prevent Creating Default VPC and Subnet

Rationale

- All VPCs and Subnets are created by the Network team following specific configurations

References

- https://docs.aws.amazon.com/vpc/latest/userguide/default-vpc.html

Test Scenarios

	Test Scenario	Steps	Expected Result
1	Create default VPC	1. Log in to the AWS console with a role that has access to create VPCs 2. Create Default VPC	Access Denied

Example SCP Statement

```
{
  "Effect": "Deny",
  "Action": [
        "ec2:CreateDefaultSubnet",
        "ec2:CreateDefaultVpc"
    ],
  "Resource": [
        "*"
    ]
}
```

SCP-GLACIER-1

Prevent Glacier Deletion

Rationale

- Security policies require that all S3 Glacier Vaults and Archives cannot be deleted

References

- https://docs.aws.amazon.com/amazonglacier/latest/dev/security.html

Test Scenarios

	Test Scenario	Steps	Expected Result
1	Delete Glacier Vault	1. Log in to the AWS console with a role that has Glacier access 2. Go to S3 Glacier 3. Create Vault 4. Delete Vault	Access Denied

Example SCP Statement

```
{
  "Effect": "Deny",
  "Action": [
        "glacier:DeleteArchive",
        "glacier:DeleteVault"
    ],
  "Resource": [
        "arn:aws:glacier:*:*:vaults/*"
    ]
}
```

SCP-GUARDDUTY-1

Prevent disabling and modifying GuardDuty

Rationale

- Restrict disabling and modifying GuardDuty to an infrastructure automation framework role

References

- https://docs.aws.amazon.com/guardduty/latest/ug/guardduty_suspend-disable.html

Test Scenarios

	Test Scenario	Steps	Expected Result
1	Disable GuardDuty	1. Log in to the AWS console with a role that is not the INFRASTRUCTURE_AUTOMATION_ROLE in the statement but has GuardDuty access 2. Disassociate the account in the Accounts screen 3. Suspend GuardDuty	Access Denied

Example SCP Statement

```
{
  "Effect": "Deny",
  "Action": [
        "guardduty:DeclineInvitations",
        "guardduty:Disassociate*",
        "guardduty:DeleteDetector",
        "guardduty:DeleteInvitations",
        "guardduty:DeleteIPSet",
        "guardduty:DeleteMembers",
        "guardduty:DeleteThreatIntelSet",
        "guardduty:StopMonitoringMembers",
        "guardduty:UpdateDetector"
    ],
  "Resource": [
        "*"
    ],
  "Condition": {
        "ArnNotLike": {
            "aws:PrincipalARN": [
                "arn:aws:iam::*:role/[INFRASTRUCTURE_AUTOMATION_ROLE]"
            ]
        }
    }
}
```

SCP-IAM-1

Prevent the root user from performing any actions.

Rationale

- The root user should not have access keys per AWS security best practices.

References

- https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html
- https://docs.aws.amazon.com/kms/latest/developerguide/key-policies.html#key-policy-default-allow-root-enable-iam

Test Scenarios

	Test Scenario	Steps	Expected Result
1	Create S3 bucket with root user	1. Log in to the AWS console as root user 2. Go to S3 and create a bucket	Access Denied

Example SCP Statement

```
{
  "Effect": "Deny",
  "Action": [
        "*"
    ],
  "Resource": [
        "*"
    ],
  "Condition": {
        "ArnLike": {
            "aws:PrincipalArn": "arn:aws:iam::*:root"
        }
    }
}
```

SCP-IAM-2

Prevent creating access keys for the root user.

Rationale

- The root user should not have access keys per AWS security best practices.

References

- https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html

Test Scenarios

	Test Scenario	Steps	Expected Result
1	Create access key for root user	1. Log in to the AWS console as root user 2. Create an access key following these instructions: https://docs.aws.amazon.com/general/latest/gr/managing-aws-access-keys.html	

Example SCP Statement

```
{
  "Effect": "Deny",
  "Action": [
        "iam:CreateAccessKey"
    ],
  "Resource": [
        "arn:aws:iam::*:root"
    ]
}
```

SCP-IAM-3

Prevent modifications to specific IAM roles.

Rationale

- Infrastructure automation frameworks use specific IAM roles that should only be modified by the automation framework.
- Prevent IAM administrators from modifying infrastructure automation created roles.

References

- https://aws.amazon.com/controltower/
- https://aws.amazon.com/solutions/aws-landing-zone/

Test Scenarios

	Test Scenario	Steps	Expected Result
1	Modify protected role	1. Log in to the AWS console with a role that is not the INFRASTRUCTURE_AUTOMATION_ROLE in the statement but has IAM full access 2. Modify one of the protected roles by attaching a new policy	

Example SCP Statement

```
{
  "Effect": "Deny",
  "Action": [
        "iam:AttachRolePolicy",
        "iam:CreateRole",
        "iam:DeleteRole",
        "iam:DeleteRolePermissionsBoundary",
        "iam:DeleteRolePolicy",
        "iam:DetachRolePolicy",
        "iam:PutRolePermissionsBoundary",
        "iam:PutRolePolicy",
        "iam:UpdateRole",
        "iam:UpdateRoleDescription"
    ],
  "Resource": [
        "arn:aws:iam::*:role/[PROTECTED_ROLE_PREFIX]*",
        "arn:aws:iam::*:role/*[PARTIAL_PROTECTED_ROLE_NAME]*",
        "arn:aws:iam::*:role/[PROTECTED_ROLE_NAME]"
    ],
  "Condition": {
        "ArnNotLike": {
            "aws:PrincipalARN":
"arn:aws:iam::*:role/[INFRASTRUCTURE_AUTOMATION_ROLE]"
        }
    }
}
```

SCP-IAM-4

Prevent iam:UpdateAssumeRolePolicy on specific IAM roles.

Rationale

- Infrastructure automation frameworks use highly privileged roles and should only be assumed from specific roles
- Infrastructure automation frameworks use specific IAM roles that should only be modified by the automation framework
- Prevent IAM administrators from modifying infrastructure automation created roles

References

- https://aws.amazon.com/controltower/
- https://aws.amazon.com/solutions/aws-landing-zone/

Test Scenarios

Test Scenario	Steps	Expected Result

	Test Scenario	Steps	Expected Result
1	Modify protected role's assume role policy	1. Log in to the AWS console with a role that is not the ALLOWED_LAMBDA_ROLE_NAME in the statement but has IAM full access 2. Modify one of the protected roles by modifying the assume role policy to add another role	Access Denied

Example SCP Statement

```
{
  "Effect": "Deny",
  "Action": [
        "iam:UpdateAssumeRolePolicy"
    ],
  "Resource": [
        "arn:aws:iam::*:role/[PROTECTED_ROLE_PREFIX]*",
        "arn:aws:iam::*:role/*[PARTIAL_PROTECTED_ROLE_NAME]*",
        "arn:aws:iam::*:role/[PROTECTED_ROLE_NAME]"
    ],
  "Condition": {
        "ArnNotLike": {
            "aws:PrincipalARN":
"arn:aws:iam::*:role/[ALLOWED_LAMBDA_ROLE_NAME]"
        }
    }
}
```

SCP-IAM-5

Prevent specific IAM actions

Rationale

- Restrict specific IAM actions to approved roles

References

- https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html

Test Scenarios

	Test Scenario	Steps	Expected Result

	Test Scenario	Steps	Expected Result
1	Create new user	1. Log in to the AWS console with a role that is not the ALLOWED_ROLE_NAME in the statement but has IAM access 2. Create a new user 3. Attach a policy to an existing user	Access Denied

Example SCP Statement

```
{
  "Effect": "Deny",
  "Action": [
        "iam:AttachUserPolicy",
        "iam:CreateAccessKey",
        "iam:CreateUser",
        "iam:PutUserPolicy",
        "iam:DeleteSAMLProvider"
    ],
  "Resource": [
        "*"
    ],
  "Condition": {
        "ArnNotLike": {
            "aws:PrincipalARN": [
                "arn:aws:iam::*:role/[ALLOWED_ROLE_NAME]"
            ]
        }
    }
}
```

SCP-KMS-1

Prevent KMS Key Deletion

Rationale

- Prevent the accidental or intentional deletion of a KMS key
- Only allow specific roles to delete KMS keys

References

-

Test Scenarios

	Test Scenario	Steps	Expected Result

	Test Scenario	Steps	Expected Result
1	Schedule KMS Key Deletion	1. Log in to the AWS console with a role that is not the ENCRYPTION_DELETE_KEY_ROLE in the statement but has KMS access 2. Go to KMS 3. Schedule a key for deletion	Access Denied

Example SCP Statement

```
{
  "Effect": "Deny",
  "Action": [
        "kms:ScheduleKeyDeletion"
    ],
  "Resource": [
        "*"
    ],
  "Condition": {
        "ArnNotLike": {
            "aws:PrincipalArn": [
                "arn:aws:iam::*:role/[ENCRYPTION_DELETE_KEY_ROLE]"
            ]
        }
    }
}
```

SCP-LAMBDA-1

Prevent Modifications to Specific Lambda Functions

Rationale

- Infrastructure automation solutions deploy Lambda functions that need protection

References

- https://docs.aws.amazon.com/lambda/latest/dg/lambda-security.html

Test Scenarios

	Test Scenario	Steps	Expected Result
1	Modify protected Lambda function	1. Log in to the AWS console with a role that has access to Lambda 2. Modify a protected Lambda function	Access Denied

Example SCP Statement

```
{
  "Effect": "Deny",
  "Action": [
        "lambda:AddPermission",
        "lambda:CreateEventSourceMapping",
        "lambda:CreateFunction",
        "lambda:DeleteEventSourceMapping",
        "lambda:DeleteFunction",
        "lambda:DeleteFunctionConcurrency",
        "lambda:PutFunctionConcurrency",
        "lambda:RemovePermission",
        "lambda:UpdateEventSourceMapping",
        "lambda:UpdateFunctionCode",
        "lambda:UpdateFunctionConfiguration"
    ],
  "Resource": [
        "arn:aws:lambda:*:*:function:[INFRASTRUCTURE_AUTOMATION_PREFIX]*"
    ],
  "Condition": {
        "ArnNotLike": {
            "aws:PrincipalArn": [
                "arn:aws:iam::*:role/[INFRASTRUCTURE_AUTOMATION_ROLE]"
            ]
        }
    }
}
```

SCP-ORGANIZATIONS-1

Prevent organization leave, delete, or remove actions

Rationale

- Restrict organization leave, delete, and remove actions to an infrastructure automation framework role and/or administrator role

References

- https://docs.aws.amazon.com/organizations/latest/userguide/orgs_manage_accounts_remove.html

Test Scenarios

	Test Scenario	Steps	Expected Result
1	Leave the Organization	1. Log in to the AWS console with a role that is not the INFRASTRUCTURE_AUTOMATION_ROLE in the statement but has organizations access 2. Leave the organization	Access Denied

Example SCP Statement

```
{
  "Effect": "Deny",
  "Action": [
        "organizations:LeaveOrganization",
        "organizations:DeleteOrganization"
    ],
  "Resource": [
        "*"
    ],
  "Condition": {
        "ArnNotLike": {
            "aws:PrincipalARN": [
                "arn:aws:iam::*:role/[INFRASTRUCTURE_AUTOMATION_ROLE]"
            ]
        }
    }
}
```

SCP-RAM-1

Prevent sharing resources to accounts outside your organization

Rationale

- Prevent sharing resources to external accounts outside your organization

References

- https://docs.aws.amazon.com/ram/latest/userguide/getting-started-sharing.html#getting-started-sharing-create

Test Scenarios

	Test Scenario	Steps	Expected Result
1	Create external resource share	1. Log in to the AWS console with a role that has access to Resource Access Manager 2. Create a resource share leaving 'Allow external accounts' checked	Access Denied

Example SCP Statement

```
{
  "Effect": "Deny",
  "Action": [
        "*"
```

```
    ],
  "Resource": [
        "*"
    ],
  "Condition": {
        "Bool": {
            "ram:AllowsExternalPrincipals": "true"
        }
    }
}
```

SCP-S3-1

Prevent disabling S3 account public access block

Rationale

- Security policies require that all S3 buckets are not public within a specific set of accounts

References

- https://docs.aws.amazon.com/AmazonS3/latest/dev/security-best-practices.html

Test Scenarios

	Test Scenario	Steps	Expected Result
1	Modify S3 account public access block	1. Log in to the AWS console with a role that is not the ALLOWED_ROLE_NAME in the statement but has access to S3 2. Go to S3 3. Select Block public access (account settings) in the side menu 4. Edit and uncheck all settings 5. Save changes	Access Denied

Example SCP Statement

```
{
  "Effect": "Deny",
  "Action": [
        "s3:PutAccountPublicAccessBlock"
    ],
  "Resource": [
        "*"
    ],
  "Condition": {
        "ArnNotLike": {
            "aws:PrincipalARN": [
                "arn:aws:iam::*:role/[ALLOWED_ROLE_NAME]"
            ]
```

```
        }
    }
}
```

SCP-S3-2

Prevent S3 unencrypted object uploads

Rationale

- Security policies require that all S3 objects are encrypted when uploaded to buckets

References

- https://docs.aws.amazon.com/AmazonS3/latest/dev/security-best-practices.html

Test Scenarios

	Test Scenario	Steps	Expected Result
1	Upload unencrypted object	1. Log in to the AWS console with a role that has access to S3 2. Go to S3 3. Create an S3 bucket 4. Upload an object with server-side encryption set to false	Access Denied

Example SCP Statement

```
{
  "Effect": "Deny",
  "Action": [
        "s3:PutObject"
    ],
  "Resource": [
        "arn:aws:s3:::*/*"
    ],
  "Condition": {
        "Null": {
            "s3:x-amz-server-side-encryption": "true"
        },
        "StringNotEquals": {
            "s3:x-amz-server-side-encryption": [
                "aws:kms"
            ]
        }
    }
}
```

SCP-S3-3

Prevent S3 public object access

Rationale

- Security policies require that all S3 objects are not public

References

- https://docs.aws.amazon.com/AmazonS3/latest/dev/security-best-practices.html

Test Scenarios

	Test Scenario	Steps	Expected Result
1	Create public S3 object	1. Log in to the AWS console with a role that has access to S3 2. Go to S3 3. Create an S3 bucket 4. Upload an object 5. Modify the object ACL to be public	Access Denied

Example SCP Statement

```
{
  "Effect": "Deny",
  "Action": [
        "s3:PutObjectVersionAcl",
        "s3:PutObjectAcl"
    ],
  "Resource": [
        "arn:aws:s3:::*/*"
    ],
  "Condition": {
        "StringNotEquals": {
            "s3:x-amz-acl": "private"
        }
    }
}
```

SCP-S3-4

Prevent Specific S3 Buckets from Deletion

Rationale

- Security policies require the protection of specific S3 buckets

References

- https://docs.aws.amazon.com/AmazonS3/latest/dev/security.html

Test Scenarios

	Test Scenario	Steps	Expected Result
1	Delete protected S3 bucket	1. Log in to the AWS console with a role that has S3 access 2. Go to S3 3. Create S3 Bucket with a name in the resource of the SCP policy 4. Delete the bucket	Access Denied

Example SCP Statement

```
{
  "Effect": "Deny",
  "Action": [
        "s3:DeleteBucket",
        "s3:DeleteBucketPolicy",
        "s3:DeleteObject",
        "s3:DeleteObjectVersion",
        "s3:DeleteObjectTagging",
        "s3:DeleteObjectVersionTagging"
    ],
  "Resource": [
        "arn:aws:s3:::[BUCKET_TO_PROTECT]",
        "arn:aws:s3:::[BUCKET_TO_PROTECT]/*"
    ]
}
```

SCP-S3-5

Prevent Access to Specific S3 Buckets

Rationale

- Security policies require limited access to specific S3 buckets

References

- https://docs.aws.amazon.com/AmazonS3/latest/dev/security-best-practices.html

Test Scenarios

	Test Scenario	Steps	Expected Result
1	List objects in protected bucket	1. Log in to the AWS console with a role that has S3 access 2. Go to S3 3. Attempt to view objects within a protected S3 bucket	Access Denied

Example SCP Statement

```
{
  "Effect": "Deny",
  "Action": [
        "s3:GetBucketAcl",
        "s3:GetBucketCORS",
        "s3:GetBucketLocation",
        "s3:GetBucketLogging",
        "s3:GetBucketNotification",
        "s3:GetBucketObjectLockConfiguration",
        "s3:GetBucketPolicy",
        "s3:GetBucketPolicyStatus",
        "s3:GetBucketPublicAccessBlock",
        "s3:GetBucketRequestPayment",
        "s3:GetBucketTagging",
        "s3:GetBucketVersioning",
        "s3:GetBucketWebsite",
        "s3:GetObject",
        "s3:GetObjectAcl",
        "s3:GetObjectLegalHold",
        "s3:GetObjectRetention",
        "s3:GetObjectTagging",
        "s3:GetObjectTorrent",
        "s3:GetObjectVersion",
        "s3:GetObjectVersionAcl",
        "s3:GetObjectVersionForReplication",
        "s3:GetObjectVersionTagging",
        "s3:GetObjectVersionTorrent",
        "s3:GetReplicationConfiguration",
        "s3:ListAllMyBuckets",
        "s3:ListBucket",
        "s3:ListBucketMultipartUploads",
        "s3:ListBucketVersions"
    ],
  "Resource": [
        "arn:aws:s3:::[BUCKET_TO_PROTECT]",
        "arn:aws:s3:::[BUCKET_TO_PROTECT]/*"
    ],
  "Condition": {
        "ArnNotLike": {
            "aws:PrincipalArn": [
```

```
                "arn:aws:iam::*:role/[SECURITY_ROLE]",
                "arn:aws:iam::*:role/[CONFIG_RECORDER_ROLE]",
                "arn:aws:config:::*",
                "arn:aws:iam::*:role/service-role/s3crr_role_for_*"
            ]
        }
    }
}
```

SCP-SNS-1

Prevent Modifications to Specific SNS Topics

Rationale

- Protect infrastructure automation solution SNS Topics

References

- https://docs.aws.amazon.com/sns/latest/dg/sns-security-best-practices.html

Test Scenarios

	Test Scenario	Steps	Expected Result
1	Create subscription for protected SNS Topic	1. Log in to the AWS console with a role that has SNS access 2. Go to SNS 3. Attempt to create a new subscription for the protected SNS topic	Access Denied

Example SCP Statement

```
{
  "Effect": "Deny",
  "Action": [
        "sns:AddPermission",
        "sns:CreateTopic",
        "sns:DeleteTopic",
        "sns:RemovePermission",
        "sns:SetTopicAttributes"
    ],
  "Resource": [
        "arn:aws:sns:*:*:[SNS_TOPIC_TO_PROTECT]"
    ],
  "Condition": {
        "ArnNotLike": {
            "aws:PrincipalArn": [
                "arn:aws:iam::*:role/[SECURITY_ROLE]",
```

```
                "arn:aws:iam::*:role/[INFRASTRUCTURE_AUTOMATION_ROLE]"
            ]
        }
    }
}
```

SCP-EMR-1

Prevent disabling EMR public access block

Rationale

- Security policies require that EMR not be exposed to public Internet

References

- https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-block-public-access.html

Test Scenarios

	Test Scenario	Steps	Expected Result
1	Modify EMR public access block	1. Log in to the AWS console with a role that is not the ALLOWED_ROLE_NAME in the statement but has access to EMR 2. Go to EMR 3. Select Block public access (account settings) in the side menu 4. Click Change (BPA should be on by default) 5. Modify Setting 6. Save changes	Access Denied

Example SCP Statement

```
{
    "Effect": "Deny",
    "Action": [
        "emr:PutBlockPublicAccessConfiguration"
    ],
    "Resource": [
        "*"
    ],
    "Condition": {
        "ArnNotLike": {
            "aws:PrincipalARN": [
                "arn:aws:iam::*:role/[ALLOWED_ROLE_NAME]"
            ]
        }
    }
}
```