#Creates S3 bucket and retains it if stack is destroyed. Also, if the bucket exist, allows user to create new stack to update the policy.

Description: Stack to create and manage s3 buckets

Parameters:

MemberAccountNumber:

Description: AWS Account Number for Member Account

Type: String

MaxLength: 12

MinLength: 12

AllowedPattern: "[0-9]+"

Default: '902577807644'

pCreateS3Bucket:

Description: Set to false if the bucket already exist

Default: false

AllowedValues:

- false

- true

Type: String

pS3BucketName:

Description: S3Bucket Name

Type: String

Default: 'expn-cis-ml-discover-s3-output'

pRestrictByVPC:

Default: false

Description: VPC to restrict S3 Access to

AllowedValues:

- false

- true

Type: String

pVPCId:

Description: VPC to restrict S3 Access to

Type: String

Conditions:

cIsAnewBucket: !Equals [!Ref pCreateS3Bucket, true] modified

cRestrictByVPCTrue: !Equals [!Ref pRestrictByVPC, true]

cRestrictByVPCFalse: !Equals [!Ref pRestrictByVPC, false]

Resources:

rS3Bucket:

Type: AWS::S3::Bucket

DeletionPolicy: Retain

Condition: cIsAnewBucket

Properties:

BucketName: !Ref pS3BucketName

BucketEncryption:

ServerSideEncryptionConfiguration:

- ServerSideEncryptionByDefault:

SSEAlgorithm: AES256

VersioningConfiguration:

Status: Enabled

rS3BucketPolicyWithVPCPROD:

Type: AWS::S3::BucketPolicy

DeletionPolicy: Retain

Condition: cRestrictByVPCTrue

Properties:

Bucket: !Ref pS3BucketName

PolicyDocument:

Version: '2012-10-17'

Id: !Sub S3BucketPolicy${pS3BucketName}

Statement:

#Restrict by VPC

- Sid: RestrictToVPC

Effect: Deny

Principal: "\*"

Action: s3:\*

Resource: !Sub arn:aws:s3:::${pS3BucketName}/\*

Condition:

StringNotEquals:

aws:sourceVpc: !Ref pVPCId

- Sid: LocalEMRAccess

Effect: Allow

Principal:

AWS:

- !Sub arn:aws:iam::${MemberAccountNumber}:role/EMR\_EC2\_DefaultRole

Action:

- s3:List\*

- s3:GetObject

- s3:PutObject

- s3:DeleteObject

Resource:

- !Sub arn:aws:s3:::${pS3BucketName}

- !Sub arn:aws:s3:::${pS3BucketName}/\*

- Sid: MapProdRedshiftRole

Effect: Allow

Principal:

AWS:

- arn:aws:iam::227011717743:role/expn-cis-redshift-all-copy-unload

Action:

- s3:List\*

- s3:GetObject

- s3:PutObject

Resource:

- !Sub arn:aws:s3:::${pS3BucketName}

- !Sub arn:aws:s3:::${pS3BucketName}/\*

- Sid: STSToS3ProdInput

Effect: Allow

Principal:

AWS:

- arn:aws:iam::399127776771:user/GFT\_S3\_Access

Action:

- s3:List\*

- s3:GetObject

- s3:PutObject

Resource:

- !Sub arn:aws:s3:::${pS3BucketName}

- !Sub arn:aws:s3:::${pS3BucketName}/\*

Condition:

IpAddress:

aws:SourceIp:

- 205.174.34.7/32

- 167.107.191.217/32

- 167.107.96.46/32

- Sid: DenyIncorrectEncryptionHeader

Effect: Deny

Principal: "\*"

Action: s3:PutObject

Resource: !Sub arn:aws:s3:::${pS3BucketName}/\*

Condition:

StringNotEquals:

s3:x-amz-server-side-encryption: AES256

- Sid: DenyUnEncryptedObjectUploads

Effect: Deny

Principal: "\*"

Action: s3:PutObject

Resource: !Sub arn:aws:s3:::${pS3BucketName}/\*

Condition:

'Null':

s3:x-amz-server-side-encryption: 'true'

- Sid: DenyNonSSLAccess

Effect: Deny

Principal: "\*"

Action: s3:\*

Resource: !Sub arn:aws:s3:::${pS3BucketName}/\*

Condition:

Bool:

aws:SecureTransport: 'false'

- Sid: RestrictToUsers

Effect: Deny

Principal: "\*"

Action: s3:\*

Resource: !Sub arn:aws:s3:::${pS3BucketName}/\*

Condition:

StringNotLike:

aws:userId:

- !Sub ${MemberAccountNumber}

#EMR\_EC2\_DefaultRole

- AROAJJF6N45DN77T24HX4:\*

#AWSAcctReviewDev

- AROAI7VBWBZNR4XVIEW5G

#AWS-Ascend-AccountReview-IAM-Admin

- AROA5EJOUMEOFYCLS5YKL:\*

#STS provided user GFT\_S3\_Access

- AIDAIJ5WAGO3FNYEJQEES

rS3BucketPRODPolicy:

Type: AWS::S3::BucketPolicy

DeletionPolicy: Retain

Condition: cRestrictByVPCFalse

Properties:

Bucket: !Ref pS3BucketName

PolicyDocument:

Version: '2012-10-17'

Id: !Sub S3BucketPolicy${pS3BucketName}

Statement:

- Sid: LocalEMRAccess

Effect: Allow

Principal:

AWS:

- !Sub arn:aws:iam::${MemberAccountNumber}:role/EMR\_EC2\_DefaultRole

Action:

- s3:List\*

- s3:GetObject

- s3:PutObject

- s3:DeleteObject

Resource:

- !Sub arn:aws:s3:::${pS3BucketName}

- !Sub arn:aws:s3:::${pS3BucketName}/\*

- Sid: STSToS3ProdInput

Effect: Allow

Principal:

AWS:

- arn:aws:iam::399127776771:user/GFT\_S3\_Access

Action:

- s3:List\*

- s3:GetObject

- s3:PutObject

Resource:

- !Sub arn:aws:s3:::${pS3BucketName}

- !Sub arn:aws:s3:::${pS3BucketName}/\*

Condition:

IpAddress:

aws:SourceIp:

- 205.174.34.7/32

- 167.107.191.217/32

- 167.107.96.46/32

- Sid: DenyIncorrectEncryptionHeader

Effect: Deny

Principal: "\*"

Action: s3:PutObject

Resource: !Sub arn:aws:s3:::${pS3BucketName}/\*

Condition:

StringNotEquals:

s3:x-amz-server-side-encryption: AES256

- Sid: DenyUnEncryptedObjectUploads

Effect: Deny

Principal: "\*"

Action: s3:PutObject

Resource: !Sub arn:aws:s3:::${pS3BucketName}/\*

Condition:

'Null':

s3:x-amz-server-side-encryption: 'true'

- Sid: DenyNonSSLAccess

Effect: Deny

Principal: "\*"

Action: s3:\*

Resource: !Sub arn:aws:s3:::${pS3BucketName}/\*

Condition:

Bool:

aws:SecureTransport: 'false'

- Sid: RestrictToUsers

Effect: Deny

Principal: "\*"

Action: s3:\*

Resource: !Sub arn:aws:s3:::${pS3BucketName}/\*

Condition:

StringNotLike:

aws:userId:

- !Sub ${MemberAccountNumber}

#EMR\_EC2\_DefaultRole

- AROAJJF6N45DN77T24HX4:\*

#AWSAcctReviewDev

- AROAI7VBWBZNR4XVIEW5G

#AWS-Ascend-AccountReview-IAM-Admin

- AROA5EJOUMEOFYCLS5YKL:\*

#STS provided user GFT\_S3\_Access

- AIDAIJ5WAGO3FNYEJQEES