IAM Implementation and Enforcement

Starter Template

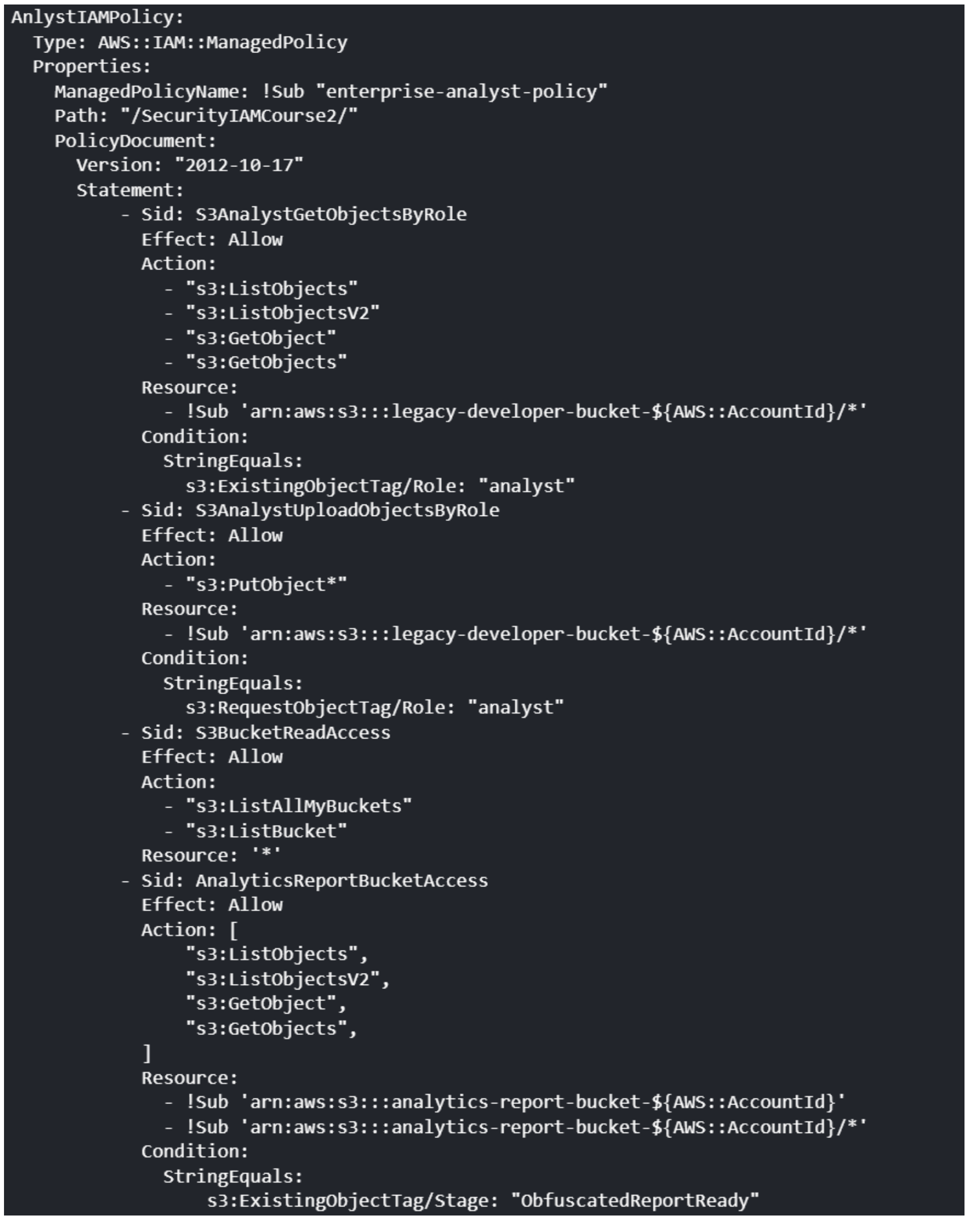
STEP 0: Overview Instructions

Project Overview	In this project, you will apply skills learned through this course to implement and enforce identity access management within AWS. You will implement a role structure with policies that will be evaluated and enforced. You will be provided with an access control matrix that outlines the appropriate roles, resources, and actions to be implemented and specific organizational requirements for policies. Next, you will evaluate the access control matrix and restrictions to ensure that each role has the appropriate policies and permissions implemented with the principle of least privilege. Upon implementation of the policies and permissions, you will create an AWS Config rule that will detach a policy that does not meet the organizational requirements. You'll finish this project with a visualization of organizational roles and policies.
Project Steps	1. Based on the access control matrix, evaluate the provided IAM policies to ensure that they are implemented with the principle of least privilege. 2. Update the policies with the permissions that meet the requirements of the access control matrix and organizational requirements 3. Attach the IAM policies to the proper IAM roles based on the access control matrix 4. Assume the IAM roles and validate the access that has been granted 5. Update Lambda code for the AWS config rule with the organizational resource restrictions 6. Execute AWS Config re-evaluation to validate that the bad policy that was created is marked as non-compliant 7. Create an architecture diagram to visualize the organizational role structure along with the permissions granted and denied.

STEPS 1 & 2: Permissions and Policies

Submission 1 Instructions:	Provide a screenshot of each updated policy with the proper statements removed to meet the organizational requirements, and if needed, copy the json policy into a text file and take a screenshot to ensure that the whole policy can be reviewed.

Submission 1a: *Submit a screenshot of the JSON for your:* enterprise-analyst-policy	The statement to be removed is: - Sid: AnalyticsSQSAccess Effect: Allow Action: ["sqs:PutMessage"] Resource: - 'arn:aws:sqs:us-east-2::analytics-queue' AnalyticsSQSAccess is not mentioned in the Access Control Matrix. Therefore, this statement should be removed.



```
AnlystIAMPolicy:
  Type: AWS::IAM::ManagedPolicy
  Properties:
    ManagedPolicyName: !Sub "enterprise-analyst-policy"
    Path: "/SecurityIAMCourse2/"
    PolicyDocument:
      Version: "2012-10-17"
      Statement:
          - Sid: S3AnalystGetObjectsByRole
            Effect: Allow
            Action:
              - "s3:ListObjects"
              - "s3:ListObjectsV2"
              - "s3:GetObject"
              - "s3:GetObjects"
            Resource:
              - !Sub 'arn:aws:s3:::legacy-developer-bucket-${AWS::AccountId}/*'
            Condition:
              StringEquals:
                s3:ExistingObjectTag/Role: "analyst"
          - Sid: S3AnalystUploadObjectsByRole
            Effect: Allow
            Action:
              - "s3:PutObject*"
            Resource:
              - !Sub 'arn:aws:s3:::legacy-developer-bucket-${AWS::AccountId}/*'
            Condition:
              StringEquals:
                s3:RequestObjectTag/Role: "analyst"
          - Sid: S3BucketReadAccess
            Effect: Allow
            Action:
              - "s3:ListAllMyBuckets"
              - "s3:ListBucket"
            Resource: '*'
          - Sid: AnalyticsReportBucketAccess
            Effect: Allow
            Action: [
                "s3:ListObjects",
                "s3:ListObjectsV2",
                "s3:GetObject",
                "s3:GetObjects",
            ]
            Resource:
              - !Sub 'arn:aws:s3:::analytics-report-bucket-${AWS::AccountId}'
              - !Sub 'arn:aws:s3:::analytics-report-bucket-${AWS::AccountId}/*'
            Condition:
              StringEquals:
                  s3:ExistingObjectTag/Stage: "ObfuscatedReportReady"
```

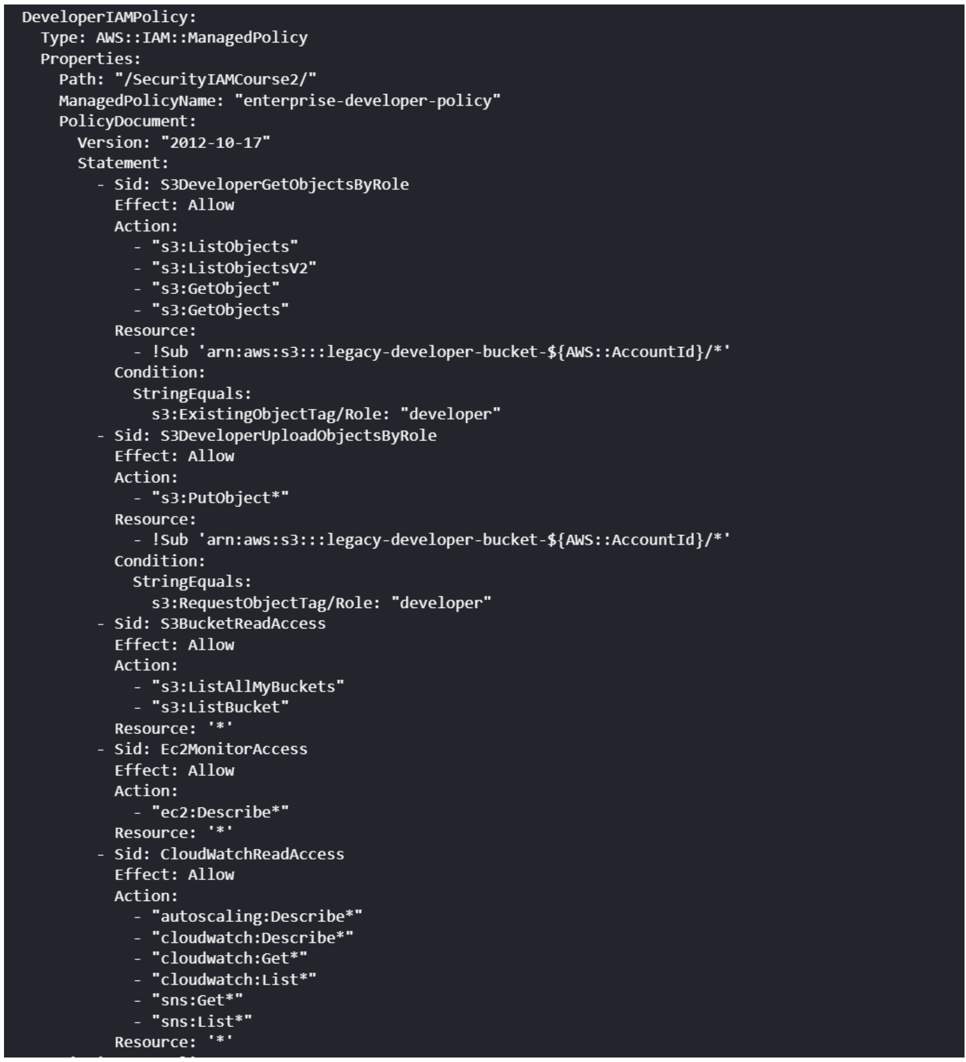
Submission 1b:

Submit a screenshot of the JSON for your:
enterprise-developer-policy

The following statement is removed:

```
- Sid: DeveloperWebhookPermissions

      Effect: Allow

      Action:

        - "sns:Publish"

      Resource: !Sub 'arn:aws:sns:us-east-2:${AWS::AccountId}:developer-webh
```

Similar to other policies, the developer-webhooks resource is not mentioned in the Access Control Matrix. As a result, the edited JSON for enterprise-developer-policy:



```
DeveloperIAMPolicy:
  Type: AWS::IAM::ManagedPolicy
  Properties:
    Path: "/SecurityIAMCourse2/"
    ManagedPolicyName: "enterprise-developer-policy"
    PolicyDocument:
      Version: "2012-10-17"
      Statement:
        - Sid: S3DeveloperGetObjectsByRole
          Effect: Allow
          Action:
            - "s3:ListObjects"
            - "s3:ListObjectsV2"
            - "s3:GetObject"
            - "s3:GetObjects"
          Resource:
            - !Sub 'arn:aws:s3:::legacy-developer-bucket-${AWS::AccountId}/*'
          Condition:
            StringEquals:
              s3:ExistingObjectTag/Role: "developer"
        - Sid: S3DeveloperUploadObjectsByRole
          Effect: Allow
          Action:
            - "s3:PutObject*"
          Resource:
            - !Sub 'arn:aws:s3:::legacy-developer-bucket-${AWS::AccountId}/*'
          Condition:
            StringEquals:
              s3:RequestObjectTag/Role: "developer"
        - Sid: S3BucketReadAccess
          Effect: Allow
          Action:
            - "s3:ListAllMyBuckets"
            - "s3:ListBucket"
          Resource: '*'
        - Sid: Ec2MonitorAccess
          Effect: Allow
          Action:
            - "ec2:Describe*"
          Resource: '*'
        - Sid: CloudWatchReadAccess
          Effect: Allow
          Action:
            - "autoscaling:Describe*"
            - "cloudwatch:Describe*"
            - "cloudwatch:Get*"
            - "cloudwatch:List*"
            - "sns:Get*"
            - "sns:List*"
          Resource: '*'
```

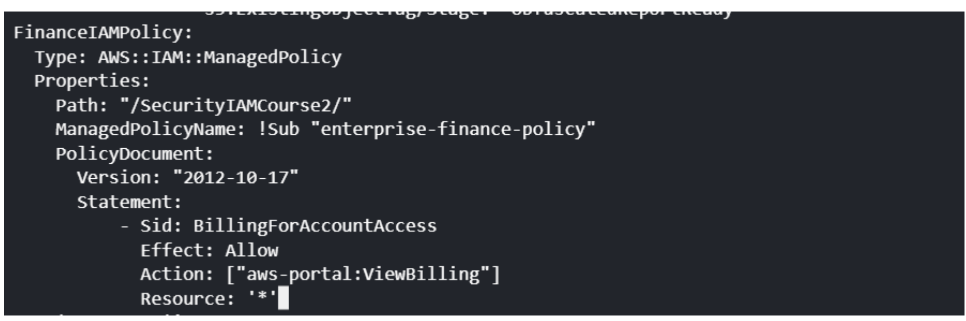
Submission 1c:

Submit a screenshot of the JSON for your:
enterprise-finance-policy

The statement to be removed from enterprise-finance-policy is:

```
- Sid: S3FinanceObjectsByUserAccess
  Effect: Allow
  Action: ["s3:PutObject", "s3:GetObject", "s3:DeleteObject"]
  Resource: 'arn:aws:s3:::legacy-developer-bucket-${AWS::AccountId}/*'
  Condition:
    StringEquals:
      s3:ExistingObjectTag/Owner: "${aws:username}"
```

According to the Access Control Matrix, enterprise-finance-policy does not have access to the legacy-developer-bucket. Therefore, its permission needs to be removed. As a result, we have a new edited policy for enterprise-finance-policy:



```
FinanceIAMPolicy:
  Type: AWS::IAM::ManagedPolicy
  Properties:
    Path: "/SecurityIAMCourse2/"
    ManagedPolicyName: !Sub "enterprise-finance-policy"
    PolicyDocument:
      Version: "2012-10-17"
      Statement:
          - Sid: BillingForAccountAccess
            Effect: Allow
            Action: ["aws-portal:ViewBilling"]
            Resource: '*'
```

Submission 1d:

Submit a screenshot of the JSON for your:
enterprise-restrictions-policy

The following statement is removed:
- Sid: "DenyS3Resources"

    Effect: Deny

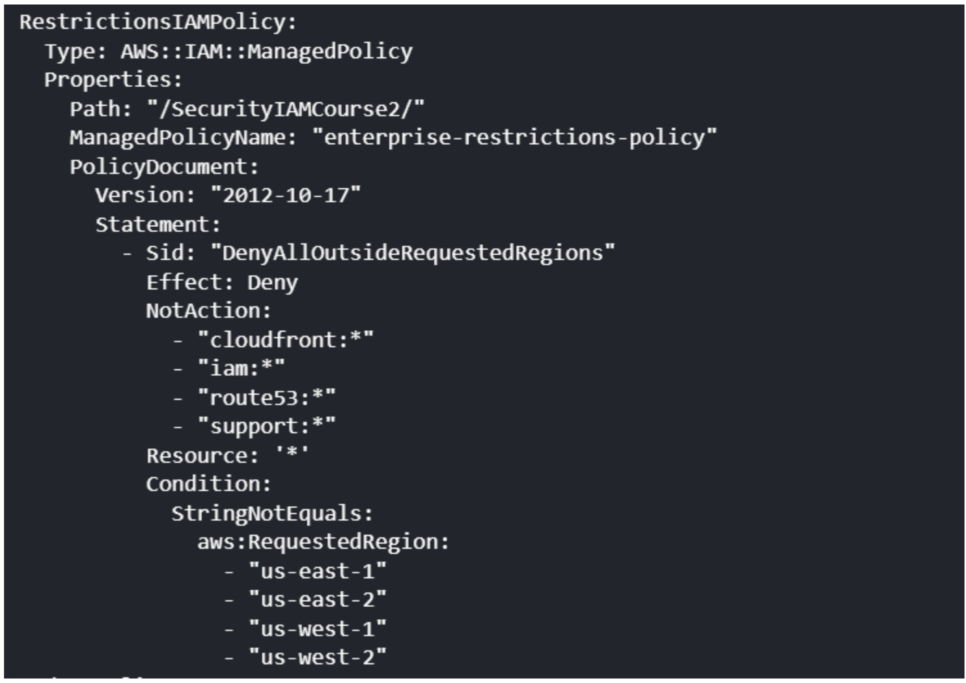
    Resource:

     - "arn:aws:s3:::*"

    Action:

     - "s3:*"

Read access to S3 is granted to enterprise-analyst-role and enterprise-developer-role.



```
RestrictionsIAMPolicy:
  Type: AWS::IAM::ManagedPolicy
  Properties:
    Path: "/SecurityIAMCourse2/"
    ManagedPolicyName: "enterprise-restrictions-policy"
    PolicyDocument:
      Version: "2012-10-17"
      Statement:
        - Sid: "DenyAllOutsideRequestedRegions"
          Effect: Deny
          NotAction:
            - "cloudfront:*"
            - "iam:*"
            - "route53:*"
            - "support:*"
          Resource: '*'
          Condition:
            StringNotEquals:
              aws:RequestedRegion:
                - "us-east-1"
                - "us-east-2"
                - "us-west-1"
                - "us-west-2"
```

STEPS 3 & 4: IAM Roles

Submission 2 Instructions:

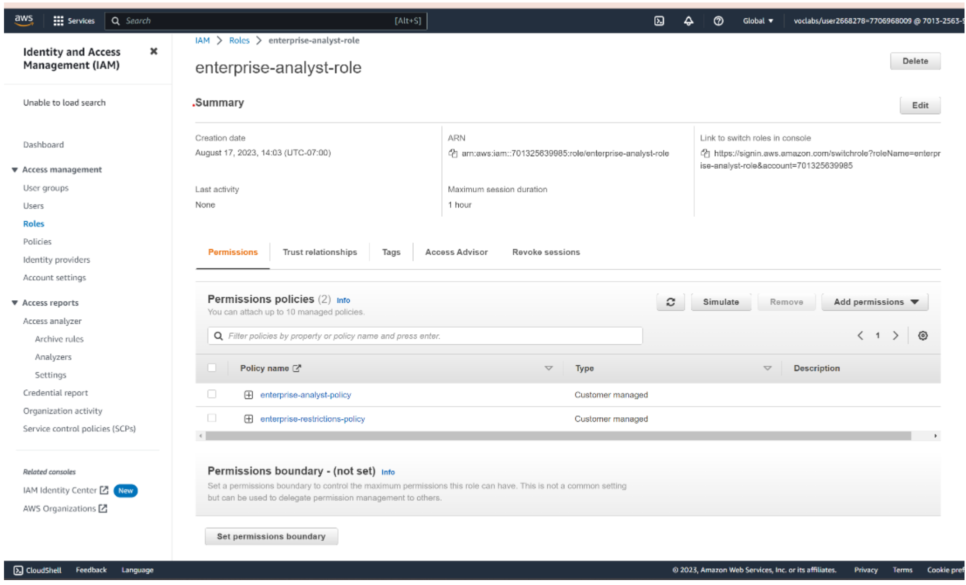
Attach the appropriate IAM policies to the appropriate IAM roles.

- The following roles should have the below policies attached.
  - enterprise-analyst-role
    - enterprise-analyst-policy
    - enterprise-restrictions-policy
  - enterprise-developer-role
    - enterprise-developer-policy
    - enterprise-restrictions-policy
  - enterprise-finance-role
    - enterprise-finance-policy
    - enterprise-restrictions-policy

Submission 2a:

Submit a screenshot of your IAM Policies for:

enterprise-analyst-role.



IAM > Roles > enterprise-analyst-role

enterprise-analyst-role

Summary

Creation date
August 17, 2023, 14:03 (UTC-07:00)

ARN
arn:aws:iam::701325639985:role/enterprise-analyst-role

Link to switch roles in console
https://signin.aws.amazon.com/switchrole?roleName=enterprise-analyst-role&account=701325639985

Last activity
None

Maximum session duration
1 hour

Permissions policies (2) Info
You can attach up to 10 managed policies.

Filter policies by property or policy name and press enter.

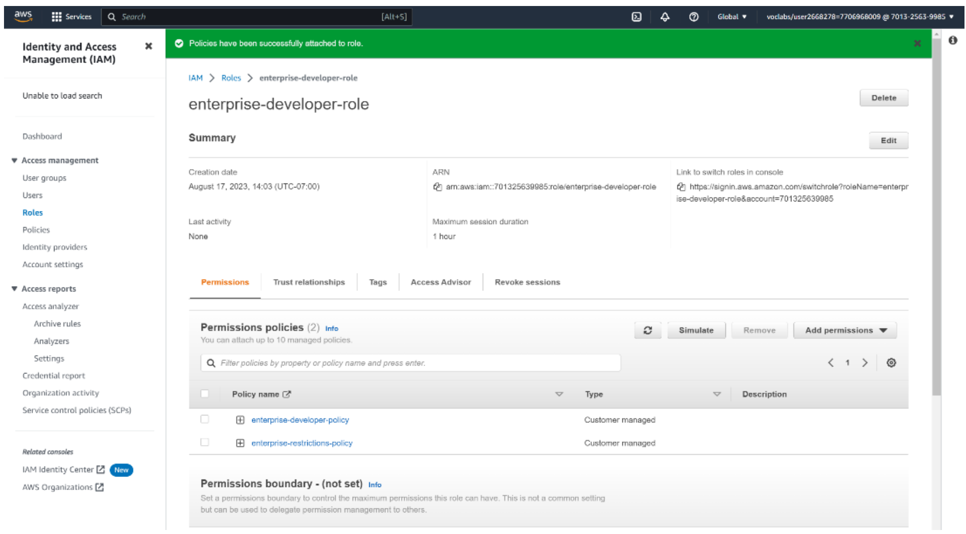
Policy name	Type	Description
enterprise-analyst-policy	Customer managed	
enterprise-restrictions-policy	Customer managed	

Permissions boundary - (not set) Info
Set a permissions boundary to control the maximum permissions this role can have. This is not a common setting but can be used to delegate permission management to others.

Submission 2b:

Submit a screenshot of your IAM Policies for:

enterprise-developer-role.



Policies have been successfully attached to role.

IAM > Roles > enterprise-developer-role

enterprise-developer-role

Summary

Creation date
August 17, 2023, 14:03 (UTC-07:00)

ARN
arn:aws:iam::701325639985:role/enterprise-developer-role

Link to switch roles in console
https://signin.aws.amazon.com/switchrole?roleName=enterprise-developer-role&account=701325639985

Last activity
None

Maximum session duration
1 hour

Permissions policies (2) Info
You can attach up to 10 managed policies.

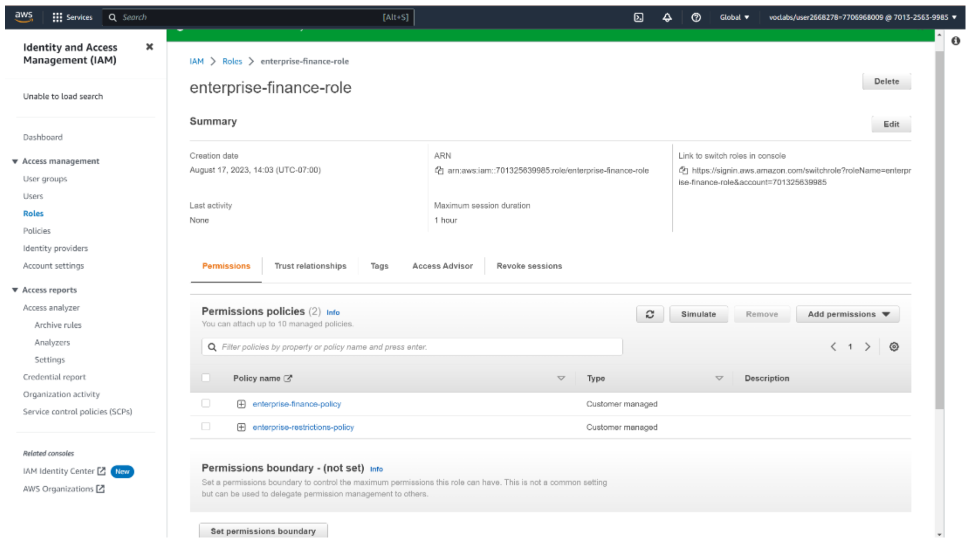
Filter policies by property or policy name and press enter.

Policy name	Type	Description
enterprise-developer-policy	Customer managed	
enterprise-restrictions-policy	Customer managed	

Permissions boundary - (not set) Info
Set a permissions boundary to control the maximum permissions this role can have. This is not a common setting but can be used to delegate permission management to others.

Submission 2c:

Submit a screenshot of your IAM Policies for:
enterprise-finance-role.



Identity and Access Management (IAM)

IAM > Roles > enterprise-finance-role

enterprise-finance-role

Summary

Creation date: August 17, 2023, 14:03 (UTC-07:00)

ARN: arn:aws:iam::701325639985:role/enterprise-finance-role

Link to switch roles in console: https://signin.aws.amazon.com/switchrole?roleName=enterprise-finance-role&account=701325639985

Last activity: None

Maximum session duration: 1 hour

Permissions policies (2)
You can attach up to 10 managed policies.

Policy name	Type	Description
enterprise-finance-policy	Customer managed	
enterprise-restrictions-policy	Customer managed	

Permissions boundary - (not set)
Set a permissions boundary to control the maximum permissions this role can have. This is not a common setting but can be used to delegate permission management to others.

Submission 3 Instructions:

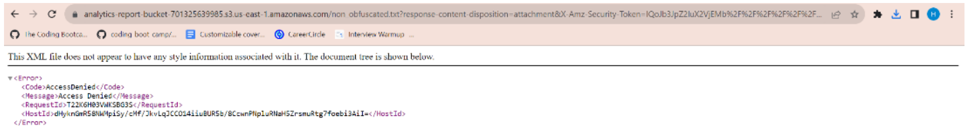
While assuming each role in the AWS console, take a screenshot of the results of accessing each service or taking action. Ensure that the role name is present in the screenshot so that the role and permissions can be validated.

The following roles should have screenshots with the proofs that the allowed/denied permissions work as expected:

- enterprise-analyst-role
  - Screenshot with access denied to non_obfuscated.txt object in s3
  - Screenshot of downloaded obfuscated.txt object
  - Screenshot of successfully uploaded analyst.txt object with supported tags to developer bucket
- enterprise-developer-role
  - Screenshot of a CloudWatch Metric
  - Screenshot of security group under EC2
  - Screenshot of successfully uploaded developer.txt to developer bucket
  - Screenshot of successfully downloaded developer.txt from developer bucket
- enterprise-finance-role
  - Screenshot of a current usage in billing console

Submission 3a:

Submit a screenshot for your enterprise-analyst-role:
Access denied to non_obfuscated.txt object in s3.

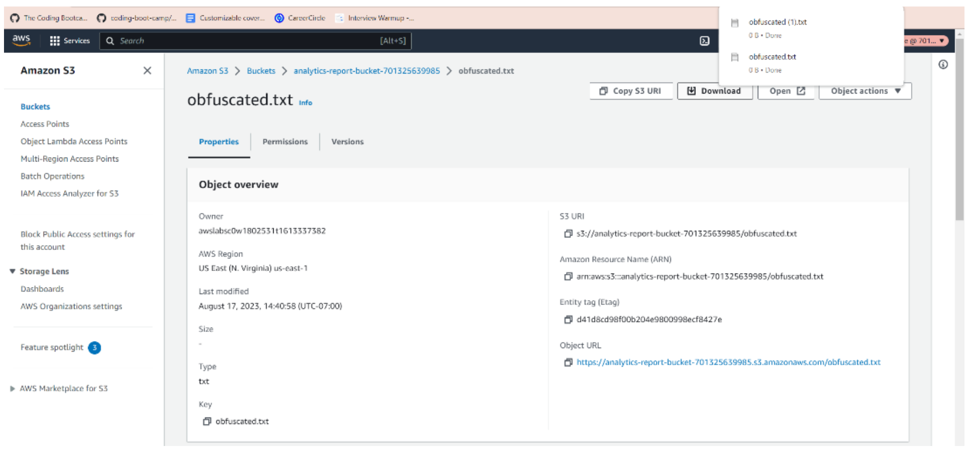

analytics-report-bucket-701325639985.s3.us-east-1.amazonaws.com/non_obfuscated.txt?response-content-disposition=attachment&X-Amz-Security-Token=IQoJb3JpZ2luX2VjEMb%2F%2F%2F%2F%2F%2F...

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<Error>
   <Code>AccessDenied</Code>
   <Message>Access Denied</Message>
   <RequestId>T22K6H03VWKSBG3S</RequestId>
   <HostId>dHyknGmR5BNWMpiSy/cMf/JkvLqJCCO14iiuBUR5b/8CcwnPNpluRNaH5ZrsmuRtg7foebi3AiI=</HostId>
 </Error>
```

Submission 3b:

Submit a screenshot for your enterprise-analyst-role:
Downloaded obfuscated.txt object


obfuscated (1).txt
0 B • Done
obfuscated.txt
0 B • Done

Amazon S3 > Buckets > analytics-report-bucket-701325639985 > obfuscated.txt

obfuscated.txt

Object overview

Owner
awslabsc0w1802531t1613337382

AWS Region
US East (N. Virginia) us-east-1

Last modified
August 17, 2023, 14:40:58 (UTC-07:00)

Size
-

Type
txt

Key
obfuscated.txt

S3 URI
s3://analytics-report-bucket-701325639985/obfuscated.txt

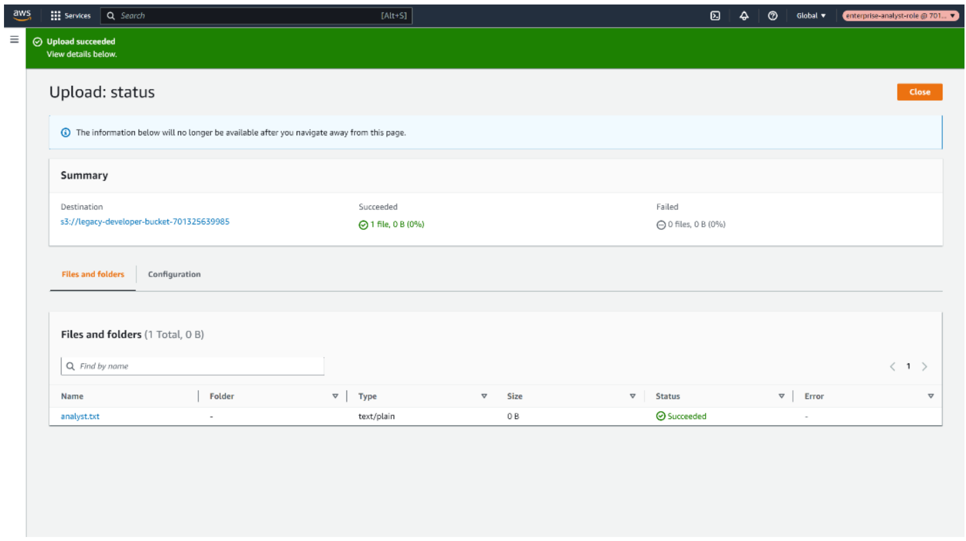
Amazon Resource Name (ARN)
arn:aws:s3:::analytics-report-bucket-701325639985/obfuscated.txt

Entity tag (Etag)
d41d8cd98f00b204e9800998ecf8427e

Object URL
https://analytics-report-bucket-701325639985.s3.amazonaws.com/obfuscated.txt

Submission 3c:

Submit a screenshot for your enterprise-analyst-role:
Successfully uploaded analyst.txt object with supported tags to developer bucket.


Upload succeeded
View details below.

Upload: status

The information below will no longer be available after you navigate away from this page.

Summary

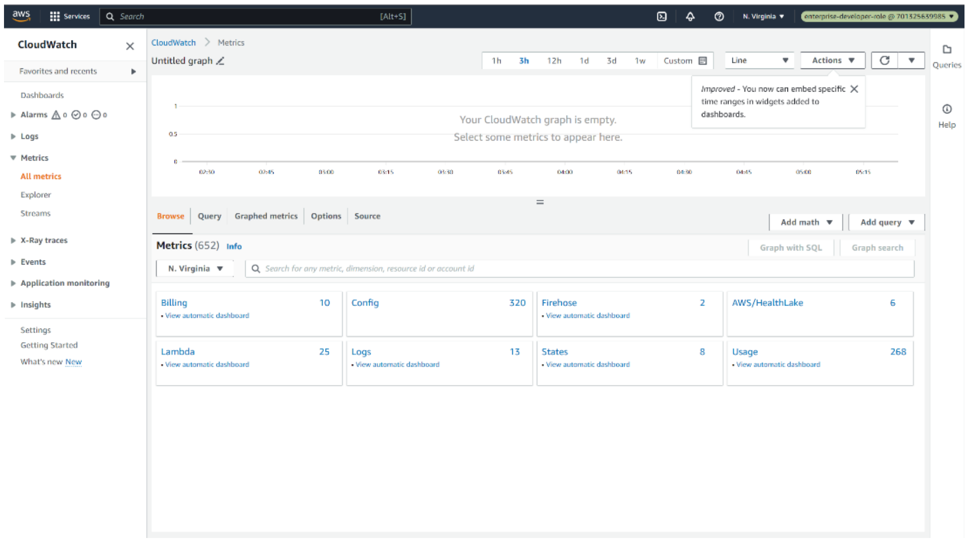
Destination	Succeeded	Failed
s3://legacy-developer-bucket-701325639985	1 file, 0 B (0%)	0 files, 0 B (0%)

Files and folders (1 Total, 0 B)

Name	Folder	Type	Size	Status	Error
analyst.txt	-	text/plain	0 B	Succeeded	-

Submission 3d:

Submit a screenshot for your enterprise-developer-role:
Accessing a CloudWatch Metric.



enterprise-developer-role @ 701325639985

CloudWatch > Metrics

Untitled graph

Your CloudWatch graph is empty.
Select some metrics to appear here.

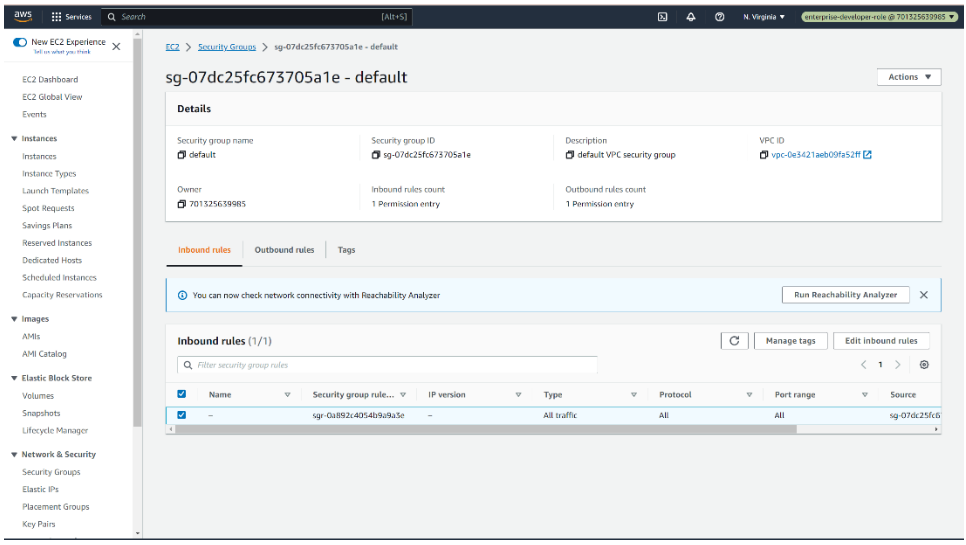
Improved - You now can embed specific time ranges in widgets added to dashboards.

Metrics (652)

Billing	10	Config	320	Firehose	2	AWS/HealthLake	6
• View automatic dashboard				• View automatic dashboard			
Lambda	25	Logs	13	States	8	Usage	268
• View automatic dashboard		• View automatic dashboard		• View automatic dashboard		• View automatic dashboard	

Submission 3e:

Submit a screenshot for your enterprise-developer-role:
Accessing security group under EC2



enterprise-developer-role @ 701325639985

EC2 > Security Groups > sg-07dc25fc673705a1e - default

sg-07dc25fc673705a1e - default

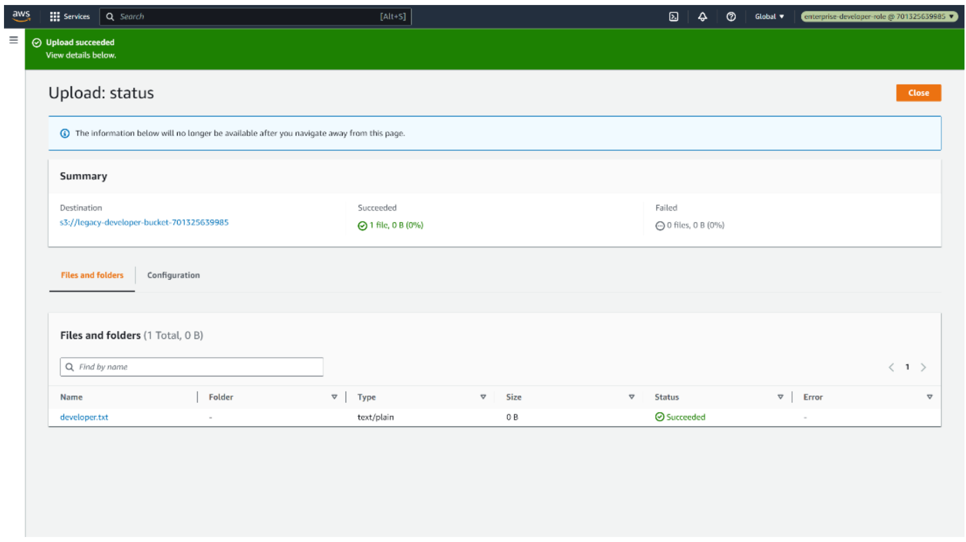
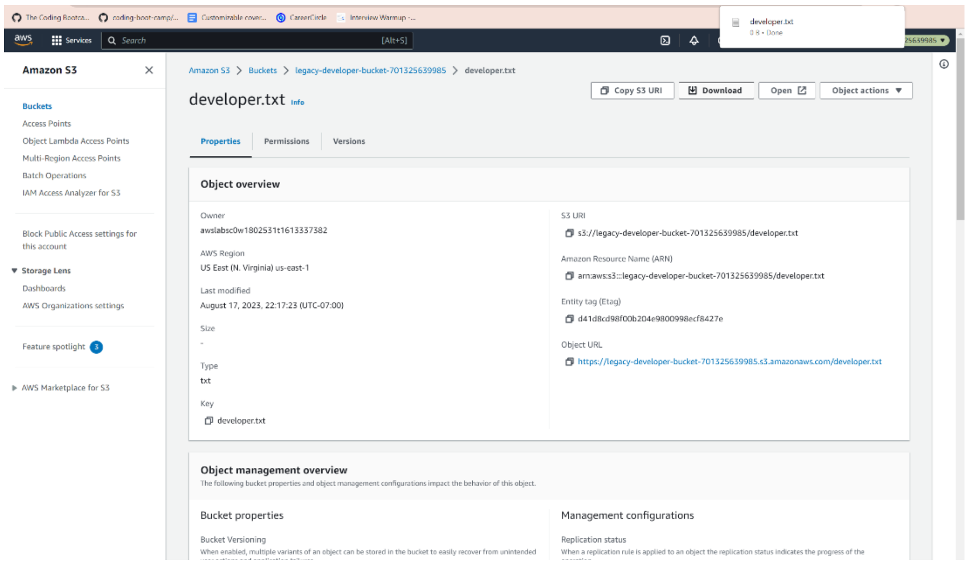
Details

Security group name	Security group ID	Description	VPC ID
default	sg-07dc25fc673705a1e	default VPC security group	vpc-0e3421aeb09fa52ff
Owner	Inbound rules count	Outbound rules count	
701325639985	1 Permission entry	1 Permission entry	

You can now check network connectivity with Reachability Analyzer

Inbound rules (1/1)

☑	Name	Security group rule...	IP version	Type	Protocol	Port range	Source
☑	–	sgr-0a892c4054b9a9a3e	–	All traffic	All	All	sg-07dc25fc6

Submission 3f: *Submit a screenshot for your enterprise-developer-role:* Successfully uploaded developer.txt to developer bucket	
Submission 3g: *Submit a screenshot for your enterprise-developer-role:* Successfully downloaded developer.txt from developer bucket	

Upload succeeded
View details below.

Upload: status

The information below will no longer be available after you navigate away from this page.

Summary

Destination: s3://legacy-developer-bucket-701325639985
Succeeded: 1 file, 0 B (0%)
Failed: 0 files, 0 B (0%)

Files and folders (1 Total, 0 B)

Name	Folder	Type	Size	Status	Error
developer.txt	-	text/plain	0 B	Succeeded	-

Amazon S3 > Buckets > legacy-developer-bucket-701325639985 > developer.txt

developer.txt

Object overview

Owner: awslabsc0w1802531t1613337382
AWS Region: US East (N. Virginia) us-east-1
Last modified: August 17, 2023, 22:17:23 (UTC-07:00)
Size: -
Type: txt
Key: developer.txt
S3 URI: s3://legacy-developer-bucket-701325639985/developer.txt
Amazon Resource Name (ARN): arn:aws:s3:::legacy-developer-bucket-701325639985/developer.txt
Entity tag (Etag): d41d8cd98f00b204e9800998ecf8427e
Object URL: https://legacy-developer-bucket-701325639985.s3.amazonaws.com/developer.txt

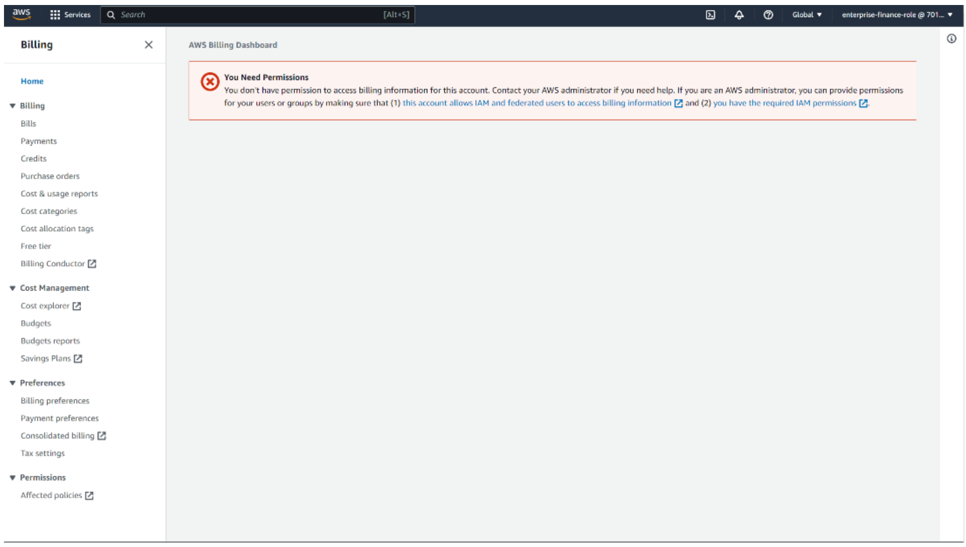
Object management overview
The following bucket properties and object management configurations impact the behavior of this object.

Bucket properties
Bucket Versioning
When enabled, multiple variants of an object can be stored in the bucket to easily recover from unintended

Management configurations
Replication status
When a replication rule is applied to an object the replication status indicates the progress of the

Submission 3h:

Submit a screenshot for your enterprise-finance-role:
Current usage in billing console



AWS Billing Dashboard

You Need Permissions
You don't have permission to access billing information for this account. Contact your AWS administrator if you need help. If you are an AWS administrator, you can provide permissions for your users or groups by making sure that (1) this account allows IAM and federated users to access billing information and (2) you have the required IAM permissions.

For some reason, this enterprise-finance-role does not have access to the Biling Console.

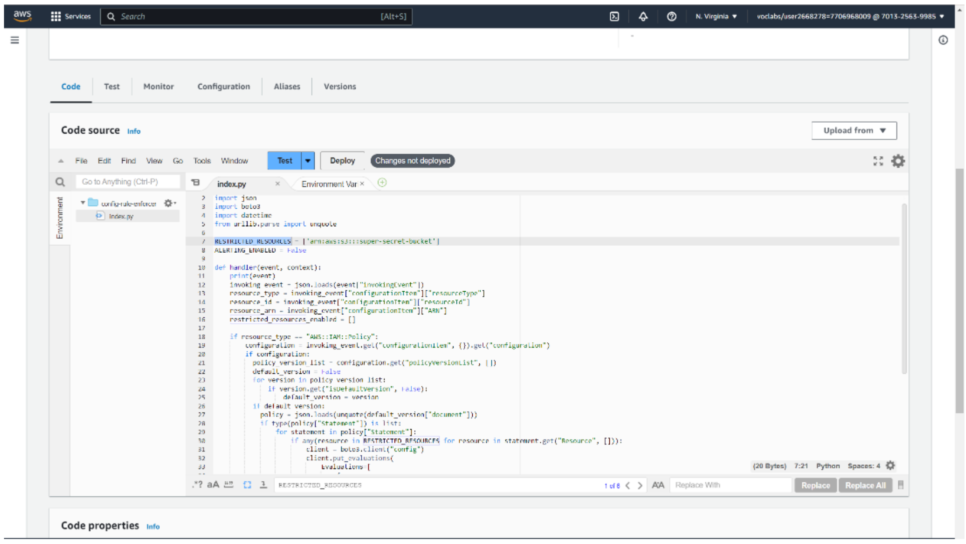
STEPS 5 & 6: AWS Config

Submission 4 Instructions:

Provide a screenshot of the Lambda console with the code updated to ensure that the RESTRICTED_RESOURCE variable includes the “arn:aws:s3:::super-secret-bucket” string in the list.

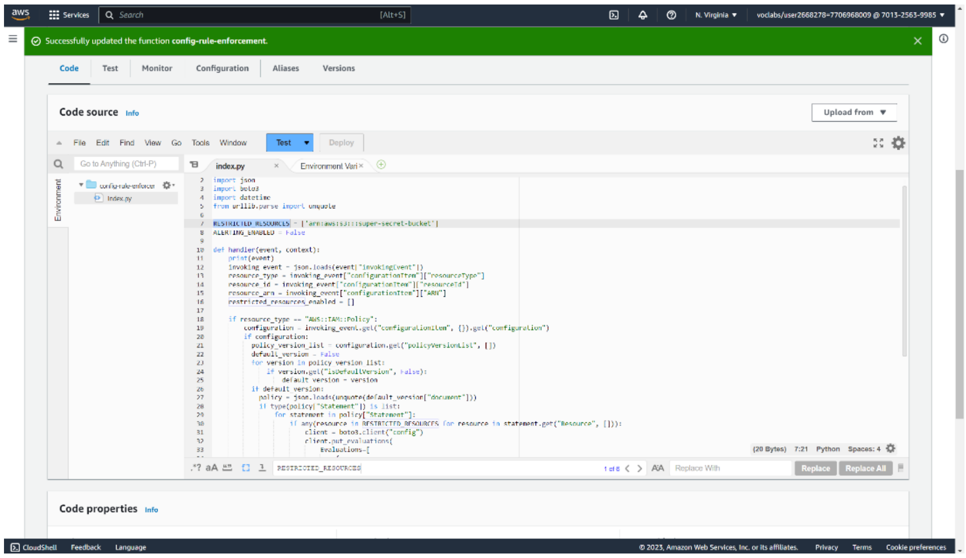
Submission 4:

Submit a screenshot (or multiple screenshots, if necessary) of your Lambda code.



```
import json
import boto3
import datetime
from urllib.parse import unquote

RESTRICTED_RESOURCES = ['arn:aws:s3:::super-secret-bucket']
ALERTING_ENABLED = False

def handler(event, context):
    print(event)
    invoking_event = json.loads(event["invokingEvent"])
    resource_type = invoking_event["configurationItem"]["resourceType"]
    resource_id = invoking_event["configurationItem"]["resourceId"]
    resource_arn = invoking_event["configurationItem"]["ARN"]
    restricted_resources_enabled = []

    if resource_type == "AWS::IAM::Policy":
        configuration = invoking_event.get("configurationItem", {}).get("configuration")
        if configuration:
          policy_version_list = configuration.get("policyVersionList", [])
          default_version = False
          for version in policy_version_list:
              if version.get("isDefaultVersion", False):
                  default_version = version
          if default_version:
            policy = json.loads(unquote(default_version["document"]))
            if type(policy["Statement"]) is list:
                for statement in policy["Statement"]:
                    if any(resource in RESTRICTED_RESOURCES for resource in statement.get("Resource", [])):
                        client = boto3.client("config")
                        client.put_evaluations(
                            Evaluations=[
```

```
RESTRICTED_RESOURCES
```

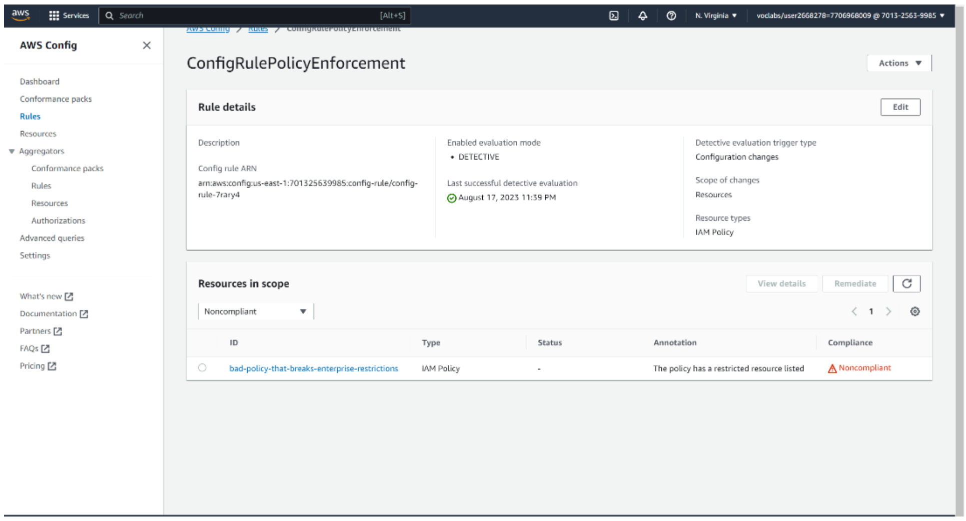


Successfully updated the function config-rule-enforcement.

```
import json
import boto3
import datetime
from urllib.parse import unquote

RESTRICTED_RESOURCES = ['arn:aws:s3:::super-secret-bucket']
ALERTING_ENABLED = False

def handler(event, context):
    print(event)
    invoking_event = json.loads(event["invokingEvent"])
    resource_type = invoking_event["configurationItem"]["resourceType"]
    resource_id = invoking_event["configurationItem"]["resourceId"]
    resource_arn = invoking_event["configurationItem"]["ARN"]
    restricted_resources_enabled = []

    if resource_type == "AWS::IAM::Policy":
        configuration = invoking_event.get("configurationItem", {}).get("configuration")
        if configuration:
          policy_version_list = configuration.get("policyVersionList", [])
          default_version = False
          for version in policy_version_list:
              if version.get("isDefaultVersion", False):
                  default_version = version
          if default_version:
            policy = json.loads(unquote(default_version["document"]))
            if type(policy["Statement"]) is list:
                for statement in policy["Statement"]:
                    if any(resource in RESTRICTED_RESOURCES for resource in statement.get("Resource", [])):
                        client = boto3.client("config")
                        client.put_evaluations(
                            Evaluations=[
```

```
RESTRICTED_RESOURCES
```

© 2023, Amazon Web Services, Inc. or its affiliates.

Submission 5 Instructions:	Manually remediate the AWS Config rule to indicate that policy that breaks the enterprise restrictions is marked as non-compliant within AWS Config. Provide a screenshot of the ConfigRulePolicyEnforcement rule with the bad-policy-that-breaks-enterprise-restrictions policy marked as non-compliant in the AWS Config service.
Submission 5: *Submit screenshot for AWS Config Rule marking bad-policy-that-breaks-enterprise-restrictions as non-compliant*	

AWS Config > Rules > ConfigRulePolicyEnforcement

N. Virginia · voclabs/user2668278=7706968009 @ 7013-2563-9985

ConfigRulePolicyEnforcement

Rule details

Description

Config rule ARN
arn:aws:config:us-east-1:701325639985:config-rule/config-rule-7rary4

Enabled evaluation mode
- DETECTIVE

Last successful detective evaluation
August 17, 2023 11:39 PM

Detective evaluation trigger type
Configuration changes

Scope of changes
Resources

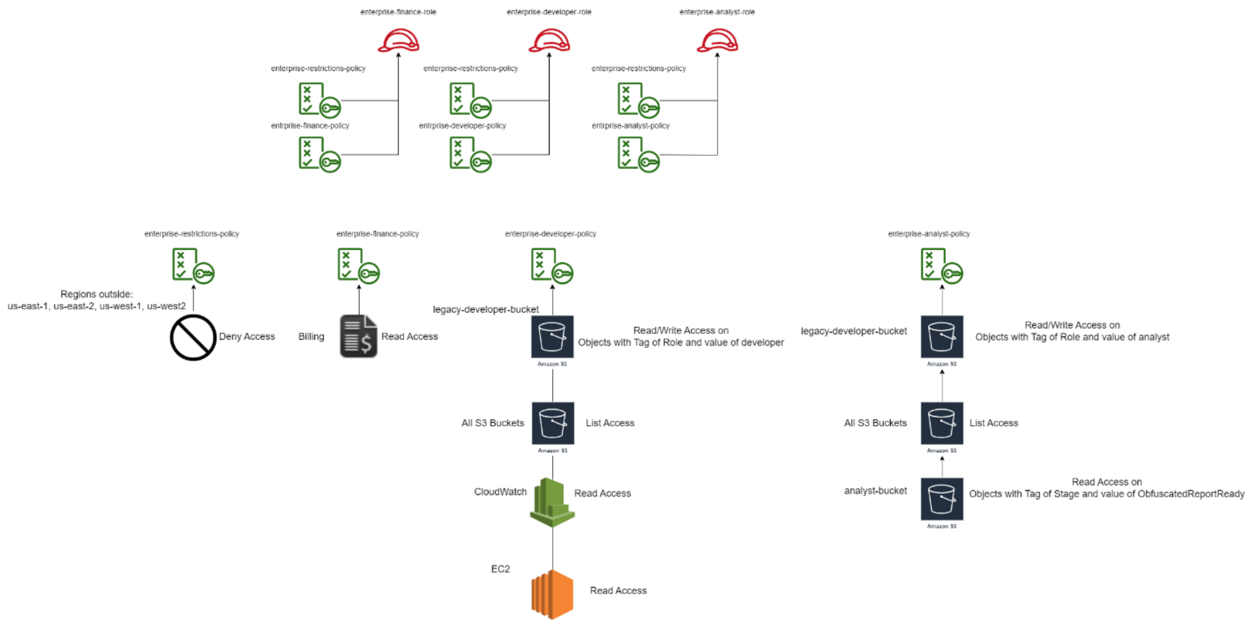
Resource types
IAM Policy

Resources in scope

Noncompliant

ID	Type	Status	Annotation	Compliance
bad-policy-that-breaks-enterprise-restrictions	IAM Policy	-	The policy has a restricted resource listed	Noncompliant

STEP 7: Organizational Role and Policy Visualization


enterprise-finance-role
enterprise-developer-role
enterprise-analyst-role
enterprise-restrictions-policy
entrprise-finance-policy
enterprise-restrictions-policy
entrprise-developer-policy
enterprise-restrictions-policy
entrprise-analyst-policy
enterprise-restrictions-policy
Regions outside:
us-east-1, us-east-2, us-west-1, us-west2
Deny Access
enterprise-finance-policy
Billing
Read Access
enterprise-developer-policy
legacy-developer-bucket
Read/Write Access on
Objects with Tag of Role and value of developer
All S3 Buckets
List Access
CloudWatch
Read Access
EC2
Read Access
enterprise-analyst-policy
legacy-developer-bucket
Read/Write Access on
Objects with Tag of Role and value of analyst
All S3 Buckets
List Access
analyst-bucket
Read Access on
Objects with Tag of Stage and value of ObfuscatedReportReady
Amazon S3

Submission 6 Instructions:	Using the draw.io organizational role diagram provided in the resources, provide a screenshot of the updated diagram to ensure that each resource defined in each policy has the appropriate permission associated with the resource. Using the permissions defined in the "Permissions To Match to Resources in Policy" table, drag each permission to the right of the proper resource in the policy. These permissions should reflect those defined in the Access Control Matrix.
Submission 6: *Submit a screenshot of your* organizational role and policy visualization.	

Optional: Standout Suggestions

Standout Suggestion #1 Instructions	Using the optional AWS Config diagram, make the connections between the resources to provide an overview for how AWS config is used to monitor and alert on changes.
Standout Submission #1 *Submit a screenshot (or multiple screenshots) of your* updated condition statements.	
Standout Suggestion #2 Instructions	Update lambda code to deliver a notification via SNS to alert on the non-compliant policy, subscribe email to provided SNS topic, and display email being sent from SNS.
Standout Submission #2 *Submit a screenshot (or multiple screenshots) of your* updated Lambda code.	