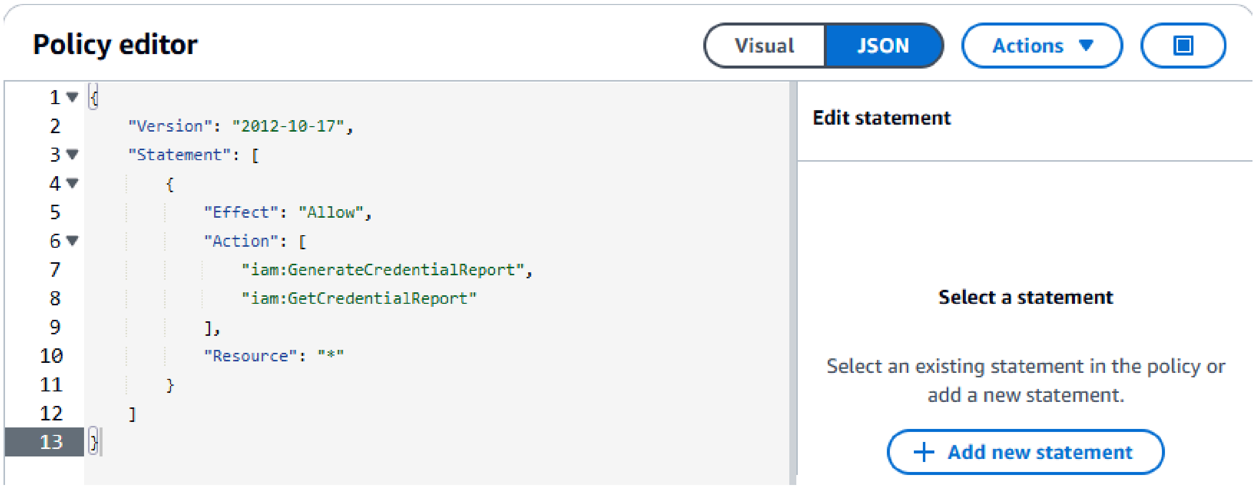
Part 1: IAM Using Console

In this part, first, you will create policies, users, and groups, and then, we will explore the IAM credential report.

Task 1 (5%): Open the IAM console-> choose Policies, and then choose to Create policy then you can use Visual editor or JSON file to create a custom policy. Create a policy with the least privilege strategy which can get the IAM credential report (name it IAM-Auditor-Policy).



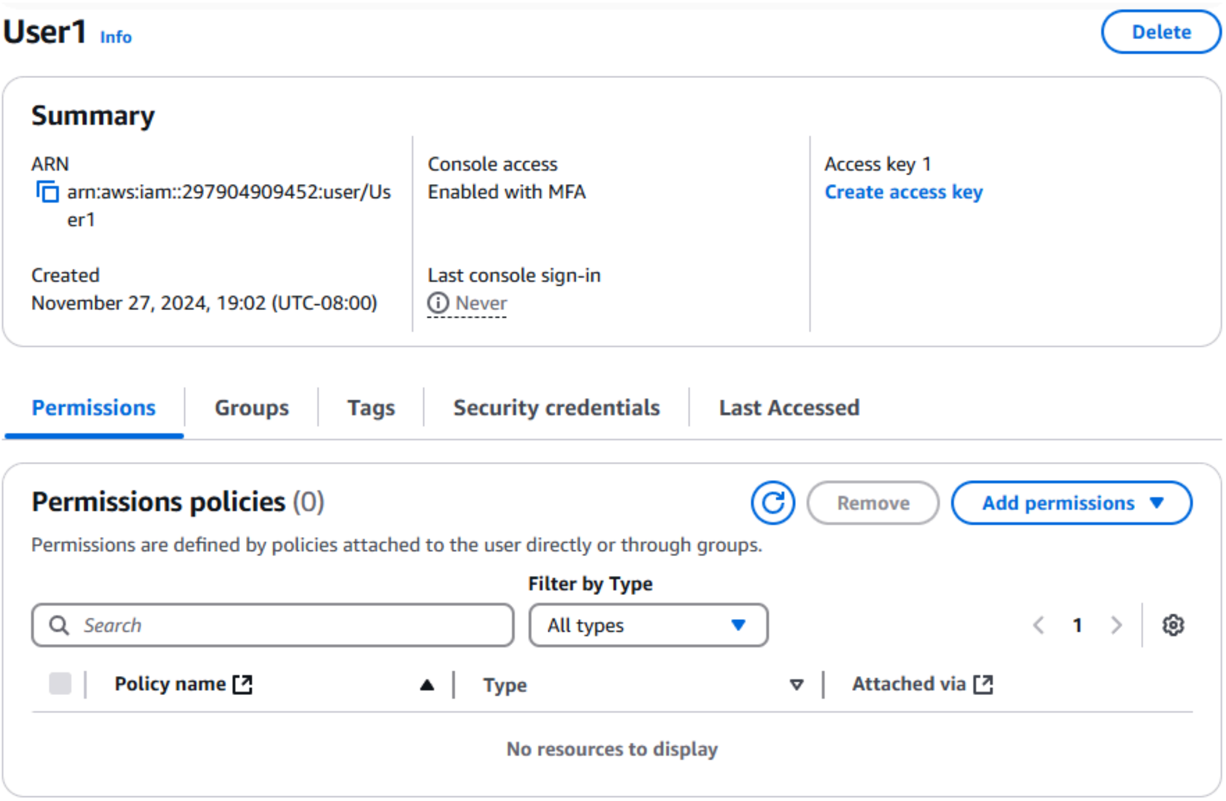
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iam:GenerateCredentialReport",
                "iam:GetCredentialReport"
            ],
            "Resource": "*"
        }
    ]
}
```

Select an existing statement in the policy or add a new statement.

Task 2 (5%): Create the following users:

- User1 without any policy or permission with console access
  - Enable MFA for User1 by your admin user and log in with user1 while MFA is enabled.
- User2 without any policy or permission with console access
  - Generate one access key for User2.
- User3 without any policy or permission with only console access

User 1



User1

Summary

ARN
arn:aws:iam::297904909452:user/User1

Created
November 27, 2024, 19:02 (UTC-08:00)

Console access
Enabled with MFA

Last console sign-in
Never

Access key 1
Create access key

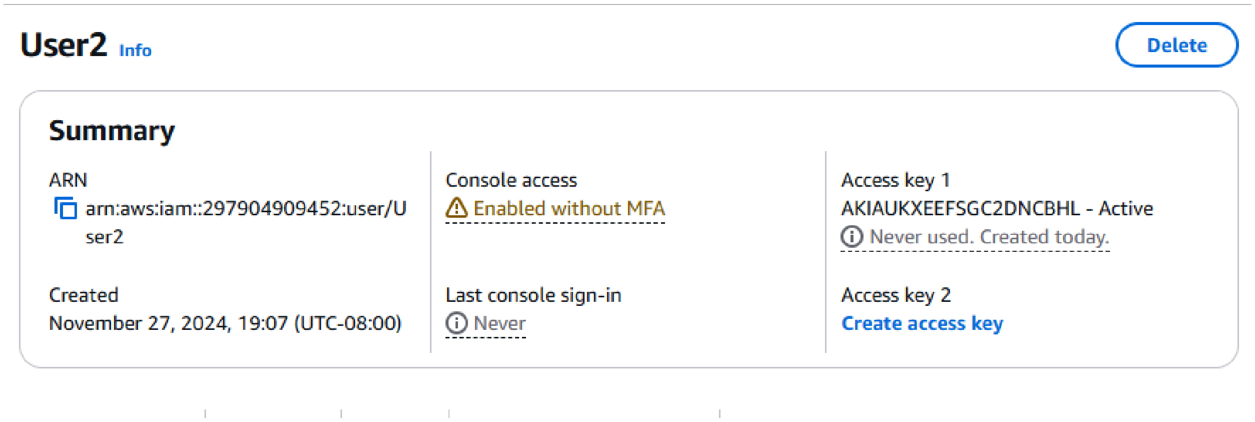
Permissions policies (0)

Permissions are defined by policies attached to the user directly or through groups.

Filter by Type: All types

□	Policy name ▲	Type ▽	Attached via
	No resources to display		

User 2



User2

Summary

ARN
arn:aws:iam::297904909452:user/User2

Created
November 27, 2024, 19:07 (UTC-08:00)

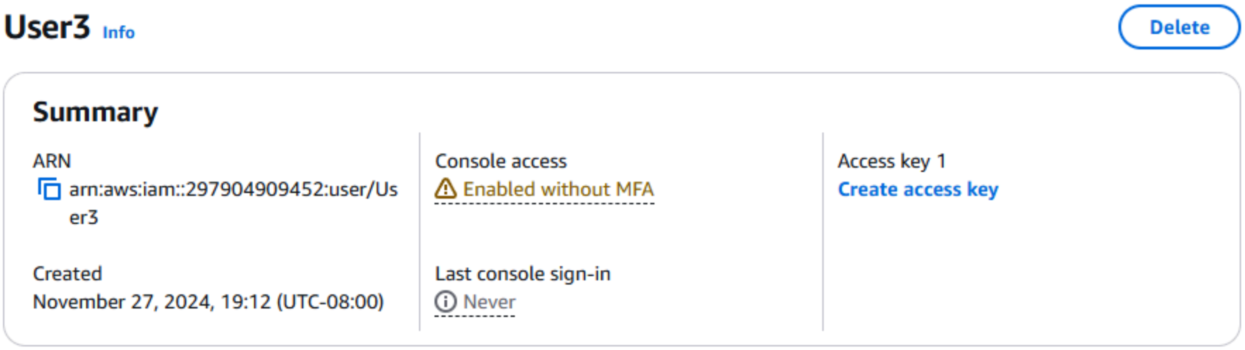
Console access
⚠ Enabled without MFA

Last console sign-in
Never

Access key 1
AKIAUKXEEFSGC2DNCBHL - Active
Never used. Created today.

Access key 2
Create access key

User 3



User3

Summary

ARN
arn:aws:iam::297904909452:user/User3

Console access
⚠ Enabled without MFA

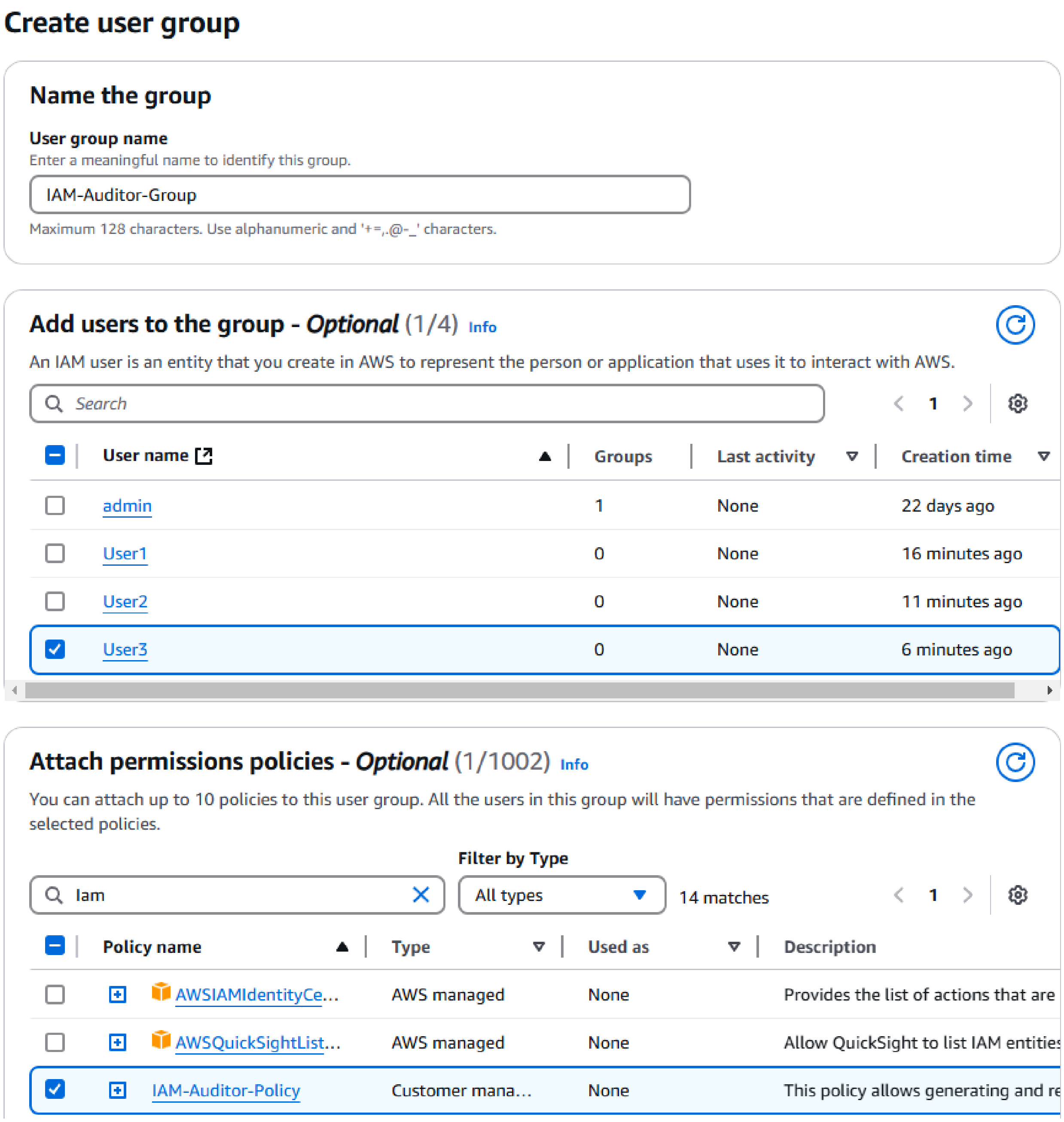
Access key 1

Created
November 27, 2024, 19:12 (UTC-08:00)

Last console sign-in
ⓘ Never

Task 3 (3%): Create a user group named IAM-Auditor-Group and attach the custom policy of task 1 to this user group.

Task 4 (2%): Add User3 to the IAM-Auditor group.



Create user group

Name the group

User group name

Enter a meaningful name to identify this group.

IAM-Auditor-Group

Maximum 128 characters. Use alphanumeric and '+=,.@-_' characters.

Add users to the group - *Optional* (1/4) Info

An IAM user is an entity that you create in AWS to represent the person or application that uses it to interact with AWS.

☐	User name ▲	Groups	Last activity ▽	Creation time ▽
☐	admin	1	None	22 days ago
☐	User1	0	None	16 minutes ago
☐	User2	0	None	11 minutes ago
☑	User3	0	None	6 minutes ago

Attach permissions policies - *Optional* (1/1002) Info

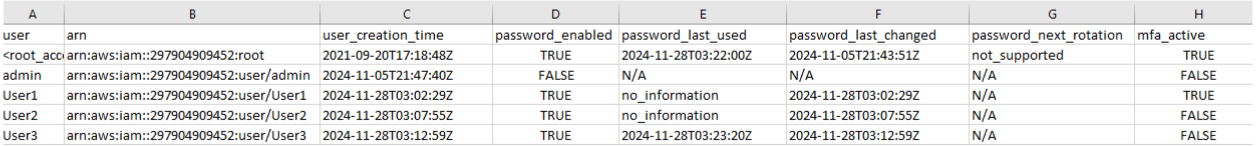
You can attach up to 10 policies to this user group. All the users in this group will have permissions that are defined in the selected policies.

Filter by Type

Iam | All types | 14 matches

☐	Policy name ▲	Type ▽	Used as ▽	Description
☐	AWSIAMIdentityCe...	AWS managed	None	Provides the list of actions that are
☐	AWSQuickSightList...	AWS managed	None	Allow QuickSight to list IAM entities
☑	IAM-Auditor-Policy	Customer mana...	None	This policy allows generating and re

Task 5 (5%): Login with User3. Go to IAM service -> Setting and get the IAM credential report. Report your findings about each user from the generated credential report.



A	B	C	D	E	F	G	H
user	arn	user_creation_time	password_enabled	password_last_used	password_last_changed	password_next_rotation	mfa_active
<root_acc	arn:aws:iam::297904909452:root	2021-09-20T17:18:48Z	TRUE	2024-11-28T03:22:00Z	2024-11-05T21:43:51Z	not_supported	TRUE
admin	arn:aws:iam::297904909452:user/admin	2024-11-05T21:47:40Z	FALSE	N/A	N/A	N/A	FALSE
User1	arn:aws:iam::297904909452:user/User1	2024-11-28T03:02:29Z	TRUE	no_information	2024-11-28T03:02:29Z	N/A	TRUE
User2	arn:aws:iam::297904909452:user/User2	2024-11-28T03:07:55Z	TRUE	no_information	2024-11-28T03:07:55Z	N/A	FALSE
User3	arn:aws:iam::297904909452:user/User3	2024-11-28T03:12:59Z	TRUE	2024-11-28T03:23:20Z	2024-11-28T03:12:59Z	N/A	FALSE

1. **Root Account**: Secure with MFA enabled and no active access keys.
2. **User1**: MFA is enabled, but the user hasn't used the account for any activity yet.
3. **User2**: Access key is active and used, but MFA is not enabled. This poses a security risk.
4. **User3**: Successfully logged in but has no permissions or active access keys, as expected.

Task 6 (5%): When you are still logged in through User3, add User2 to the IAM-Auditor group. Explain your observation. In case of failure, describe the steps needed to be taken in order for User3 to add User2 to the IAM Auditor.

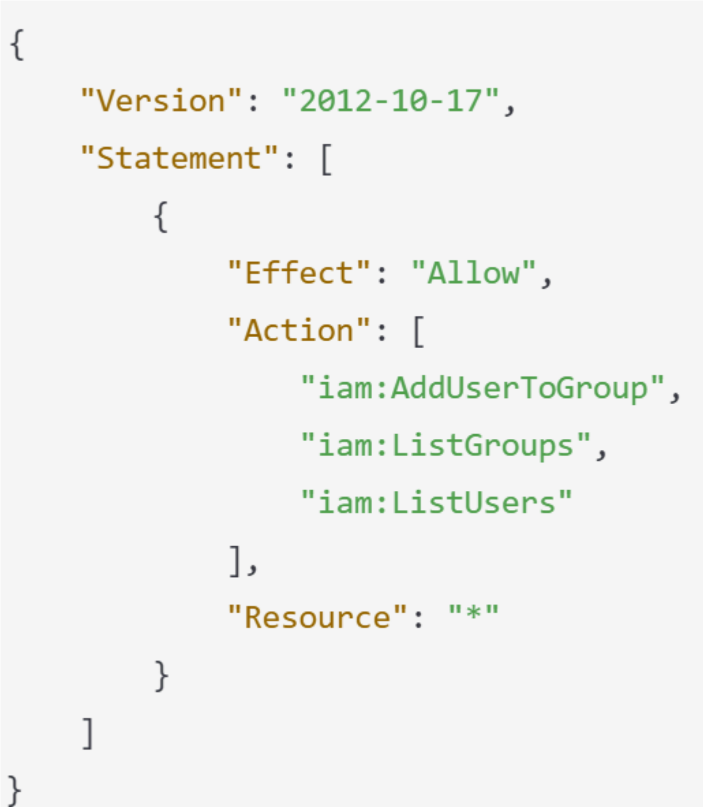
Adding User2 to the IAM-Auditor-Group failed. Because following reasons:

- **User3** has no permissions or policies assigned.
- To perform IAM operations such as modifying group memberships, specific permissions are required

To allow **User3** to add **User2** to the group, we need to grant **User3** the necessary IAM permissions.

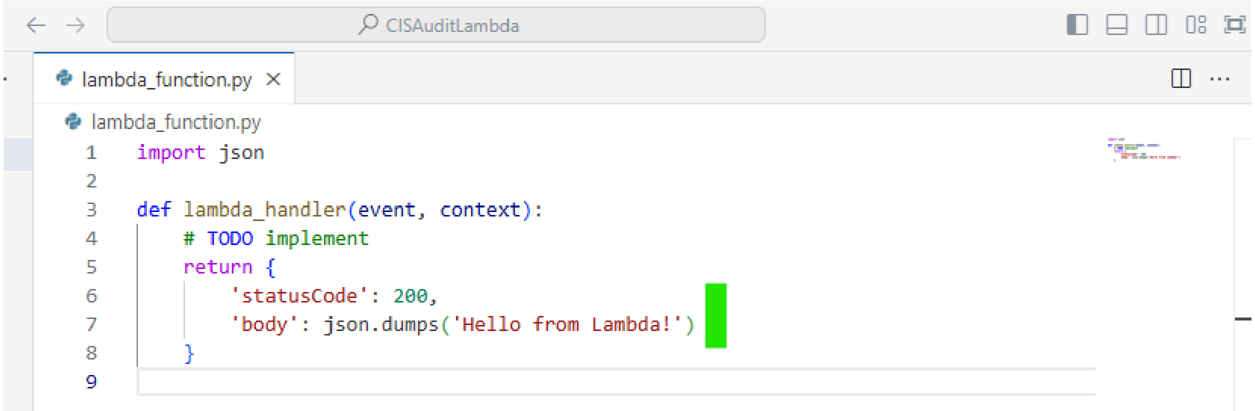
We can create a policy with the least privilege strategy which is similar to task 1.

In the Action, we can add "iam:AddUserToGroup", "iam:ListGroups", "iam:ListUsers"



```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iam:AddUserToGroup",
                "iam:ListGroups",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Task 7 (5%): Create a lambda function using Python 3.9 as a runtime, and x86_64 as an Architecture.

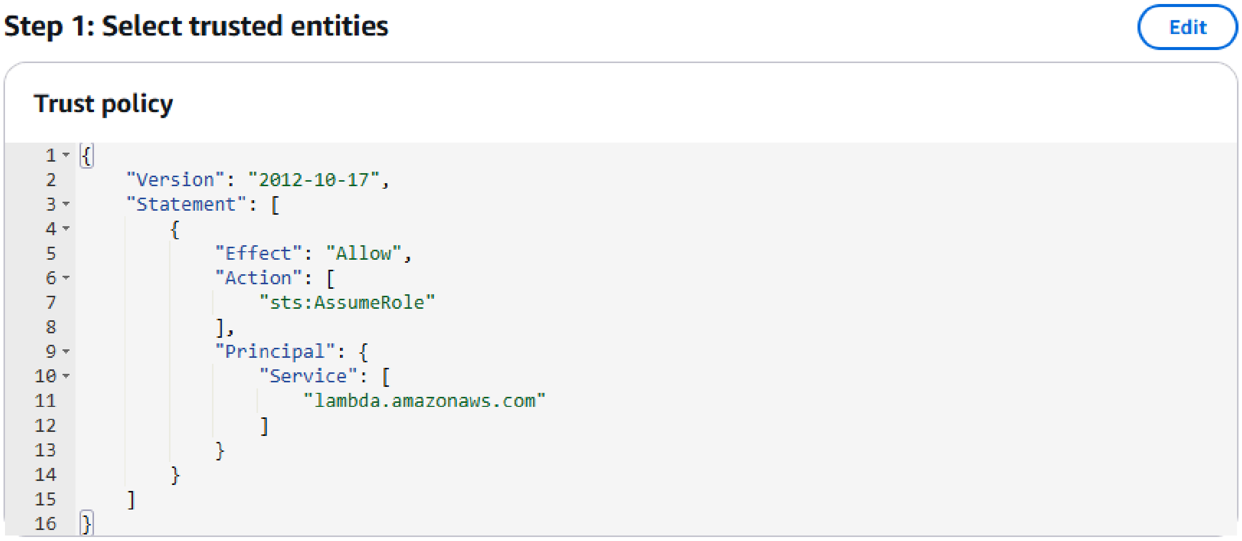


```python
import json

def lambda_handler(event, context):
    # TODO implement
    return {
        'statusCode': 200,
        'body': json.dumps('Hello from Lambda!')
    }

```

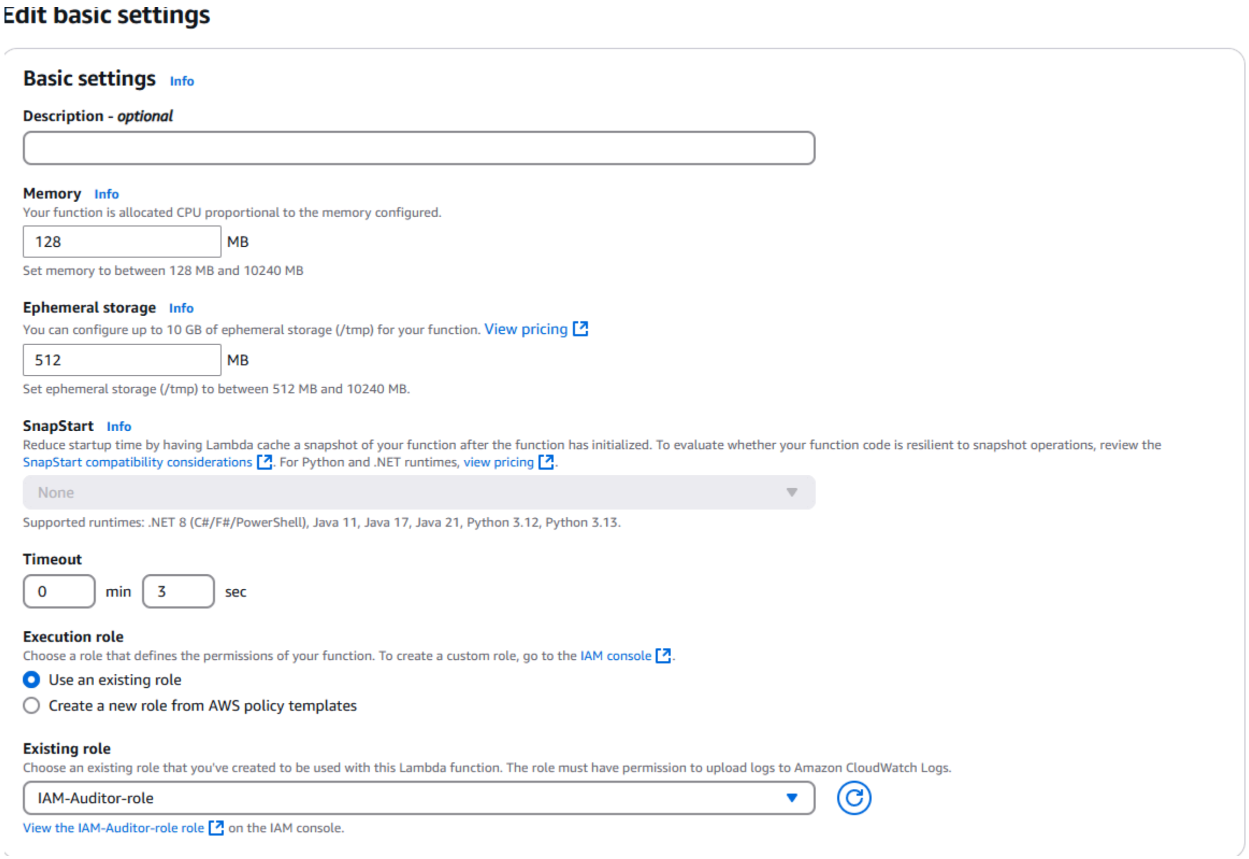
Task 8 (10%): Create a role “IAM-Auditor-role” that can be assumed by a role of Lambda function, and attach the IAM-Auditor policy to this role.



Step 1: Select trusted entities

Trust policy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sts:AssumeRole"
            ],
            "Principal": {
                "Service": [
                    "lambda.amazonaws.com"
                ]
            }
        }
    ]
}
```



Edit basic settings

Basic settings Info

Description - *optional*

Memory Info

Your function is allocated CPU proportional to the memory configured.

128 MB

Set memory to between 128 MB and 10240 MB

Ephemeral storage Info

You can configure up to 10 GB of ephemeral storage (/tmp) for your function. View pricing

512 MB

Set ephemeral storage (/tmp) to between 512 MB and 10240 MB.

SnapStart Info

Reduce startup time by having Lambda cache a snapshot of your function after the function has initialized. To evaluate whether your function code is resilient to snapshot operations, review the SnapStart compatibility considerations. For Python and .NET runtimes, view pricing.

None ▼

Supported runtimes: .NET 8 (C#/F#/PowerShell), Java 11, Java 17, Java 21, Python 3.12, Python 3.13.

Timeout

0 min 3 sec

Execution role

Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console.

● Use an existing role

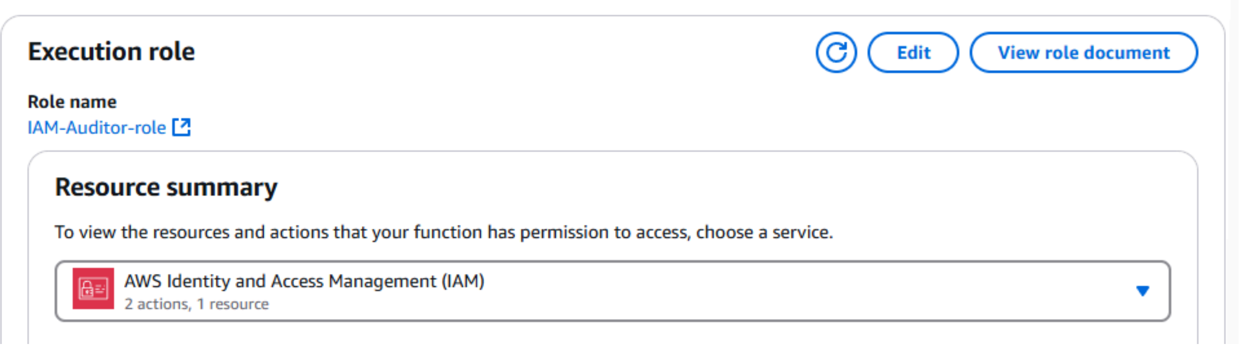
○ Create a new role from AWS policy templates

Existing role

Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.

IAM-Auditor-role ▼

View the IAM-Auditor-role role on the IAM console.



Execution role

Role name
IAM-Auditor-role

Resource summary

To view the resources and actions that your function has permission to access, choose a service.

AWS Identity and Access Management (IAM)
2 actions, 1 resource

Task 9 (20%): In the Lambda function you created in Task 7:

- Define a function to assume the role that you created in task 8 (“IAM-Auditor-role”).
- Define a function to generate and get the IAM Credential Report.
- Provide a script that uses the first function to assume the IAM-Auditor role and the second function to generate and get the IAM credential report.

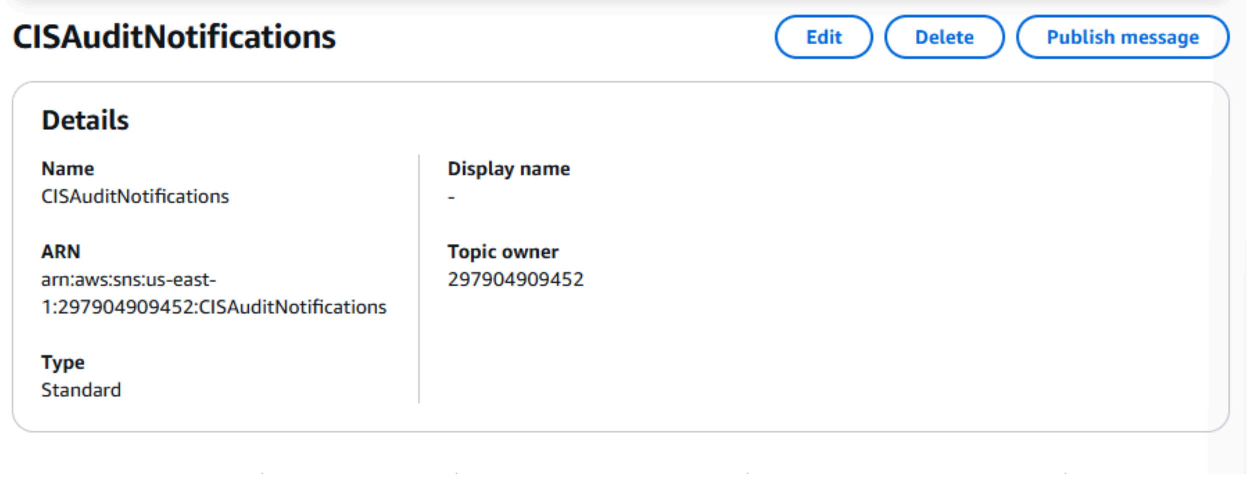
Task 10 (15%): In this task, you need to use the IAM credential report generated in the previous task to do a security audit. We are going to audit the AWS CIS controls 1.1 and 1.12.

- **CIS 1.1:** Based on the best practices the root user should not be used for daily activity. Create a function that reports the last time the root account has been used.
- **CIS 1.2:** Ensure multi-factor authentication (MFA) is enabled for all IAM users who have a console password. Now, create a function that reports all users with MFA disabled.
- **CIS 1.12:** We should ensure no access key is attached to the root account. Report if there is any key attached to the root account.

Task 9, 10 see the python file.

Task 11 (10%): In this task, you need to create a simple text report from task 10 and use the SNS service to send the report to your own email address. To do so, go to the Amazon Simple Notification Service (SNS) console -> create an SNS topic, and subscribe your email address to the topic. Then add SNS publish privilege to the Lambda function role. Finally, use Boto3 SNS publish function to send notifications.

In SNS



CISAuditNotifications

Details

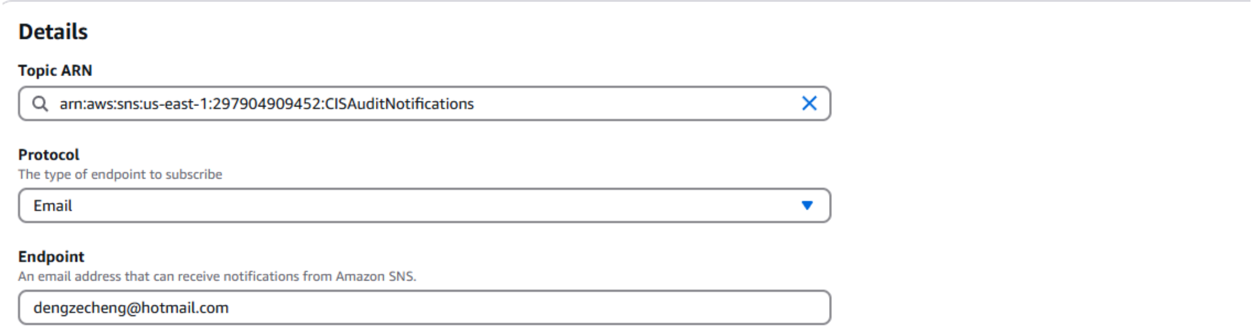
Name
CISAuditNotifications

ARN
arn:aws:sns:us-east-1:297904909452:CISAuditNotifications

Type
Standard

Display name
-

Topic owner
297904909452



Details

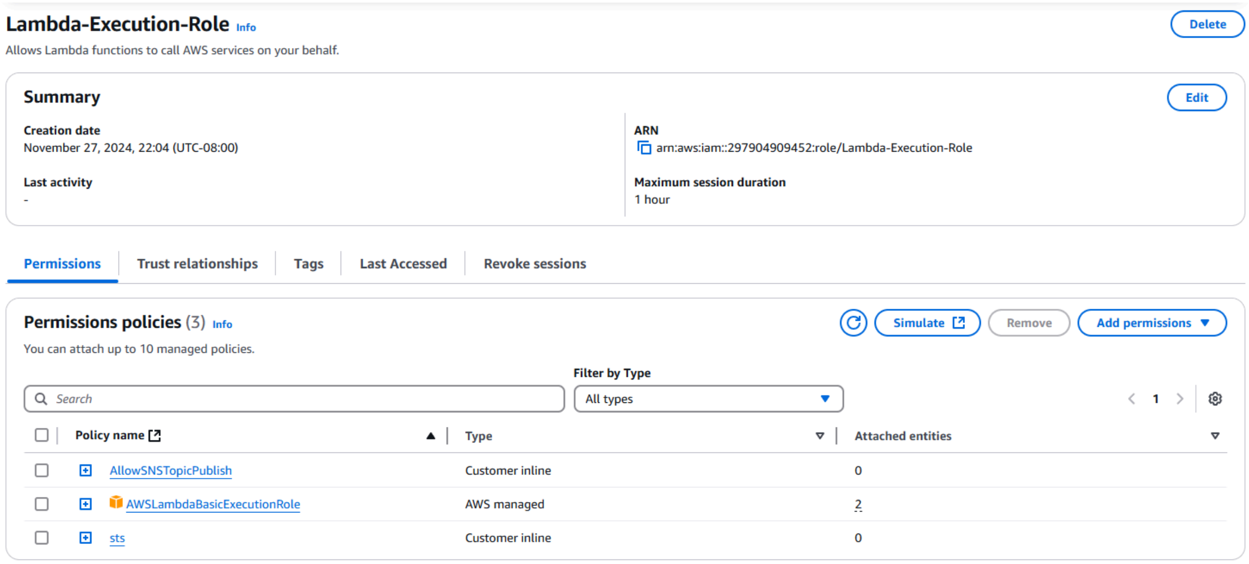
Topic ARN
arn:aws:sns:us-east-1:297904909452:CISAuditNotifications

Protocol
The type of endpoint to subscribe
Email

Endpoint
An email address that can receive notifications from Amazon SNS.
dengzecheng@hotmail.com

In IAM

Create a Lambda Execution Role



Lambda-Execution-Role Info

Allows Lambda functions to call AWS services on your behalf.

Summary

Creation date
November 27, 2024, 22:04 (UTC-08:00)

Last activity
-

ARN
arn:aws:iam::297904909452:role/Lambda-Execution-Role

Maximum session duration
1 hour

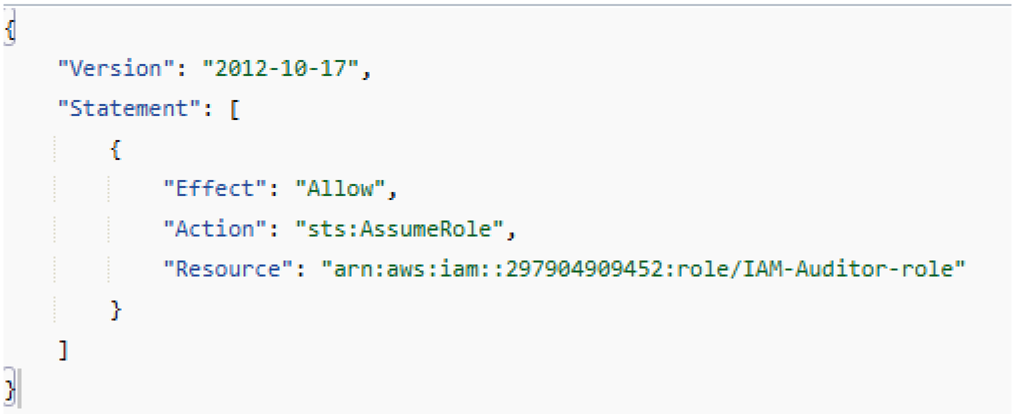
Permissions policies (3) Info

You can attach up to 10 managed policies.

Filter by Type: All types

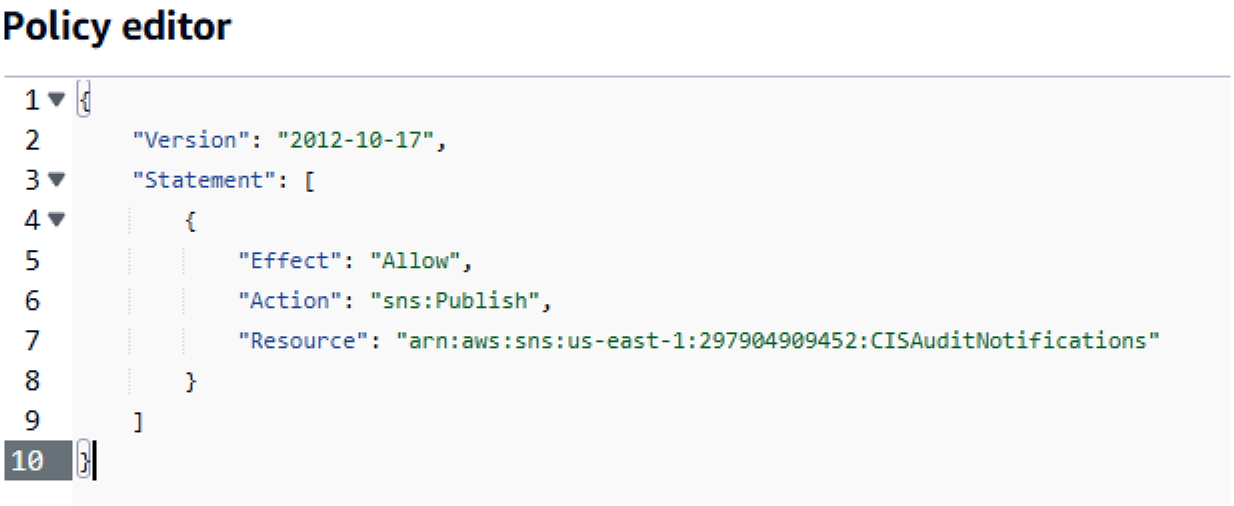
□	Policy name	Type	Attached entities
□	AllowSNSTopicPublish	Customer inline	0
□	AWSLambdaBasicExecutionRole	AWS managed	2
□	sts	Customer inline	0

Sts policy:



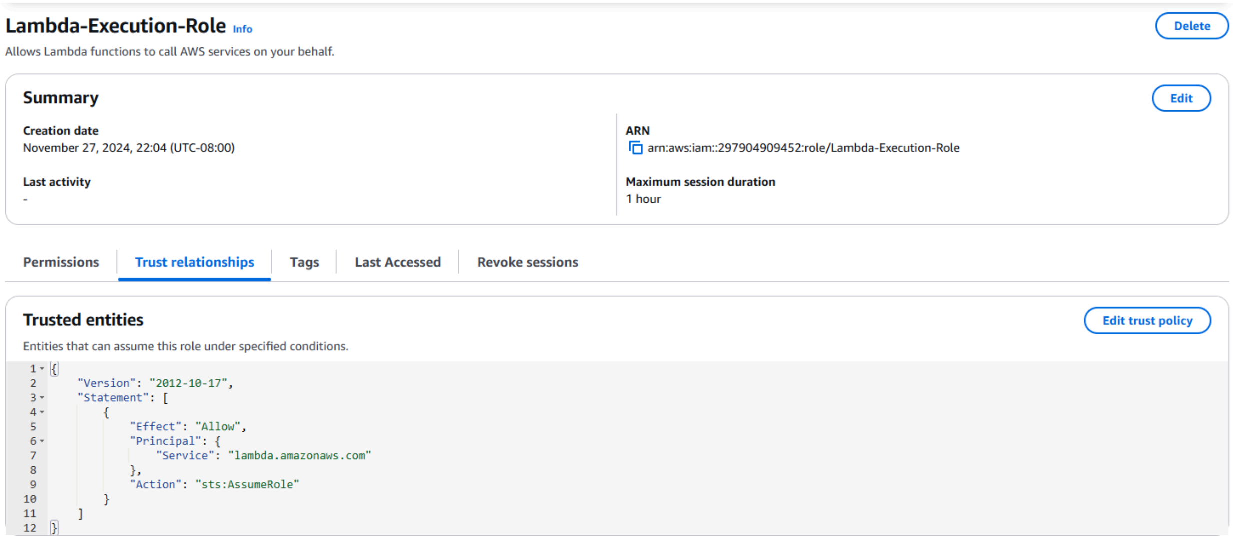
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "sts:AssumeRole",
            "Resource": "arn:aws:iam::297904909452:role/IAM-Auditor-role"
        }
    ]
}
```

AllowsSNSTopicPulish:



Policy editor

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "sns:Publish",
            "Resource": "arn:aws:sns:us-east-1:297904909452:CISAuditNotifications"
        }
    ]
}
```



Lambda-Execution-Role

Allows Lambda functions to call AWS services on your behalf.

Summary

Creation date
November 27, 2024, 22:04 (UTC-08:00)

Last activity
-

ARN
arn:aws:iam::297904909452:role/Lambda-Execution-Role

Maximum session duration
1 hour

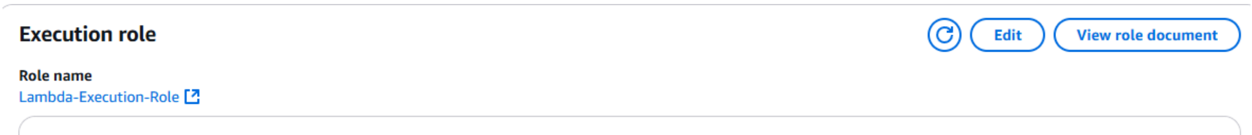
Trusted entities

Entities that can assume this role under specified conditions.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "lambda.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

In Lambda

Set the Execution role to Lambda-Execution Role, let **IAM-Auditor-role** assumed by the Lambda function via the **Lamba-Execution-Role**

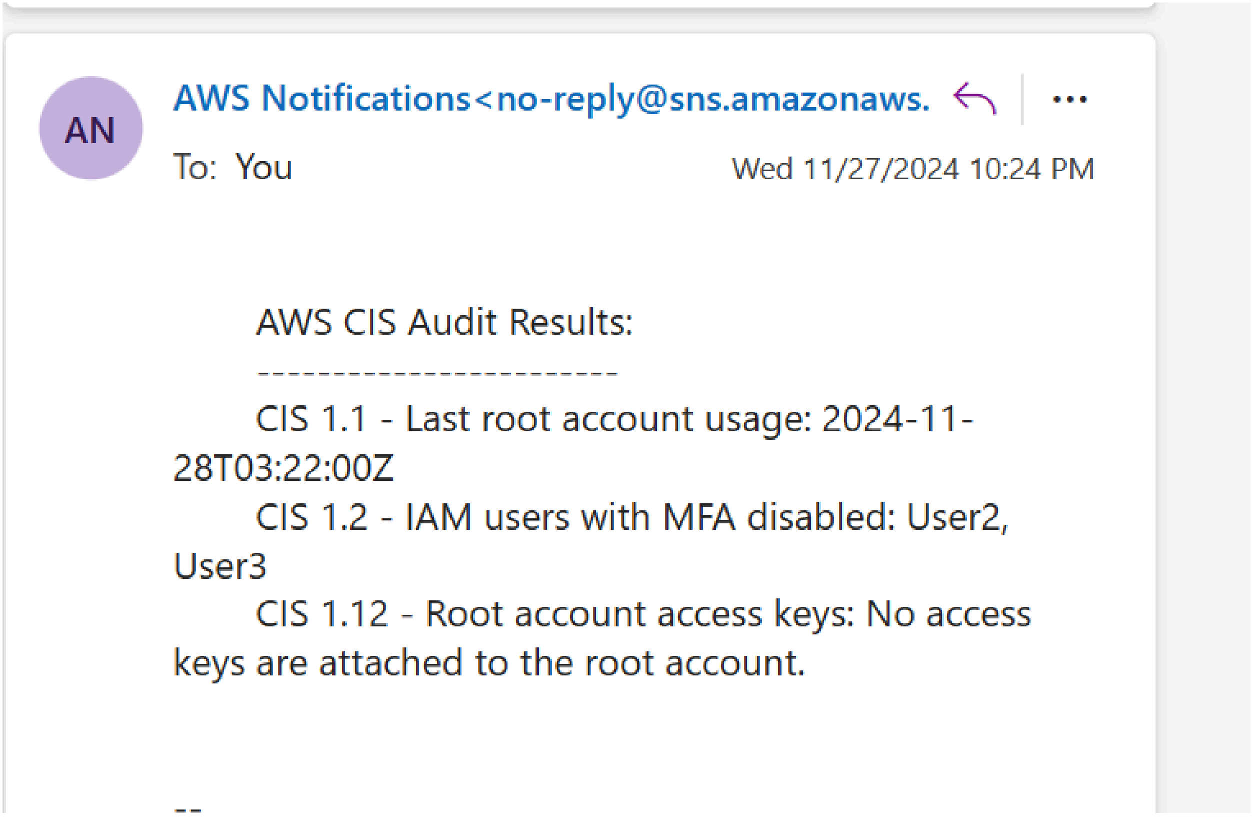


Execution role

Role name
Lambda-Execution-Role

Run the test with event JSON
{ "action": "audit"}

Got the email successfully.



AWS Notifications<no-reply@sns.amazonaws.

To: You

Wed 11/27/2024 10:24 PM

AWS CIS Audit Results:

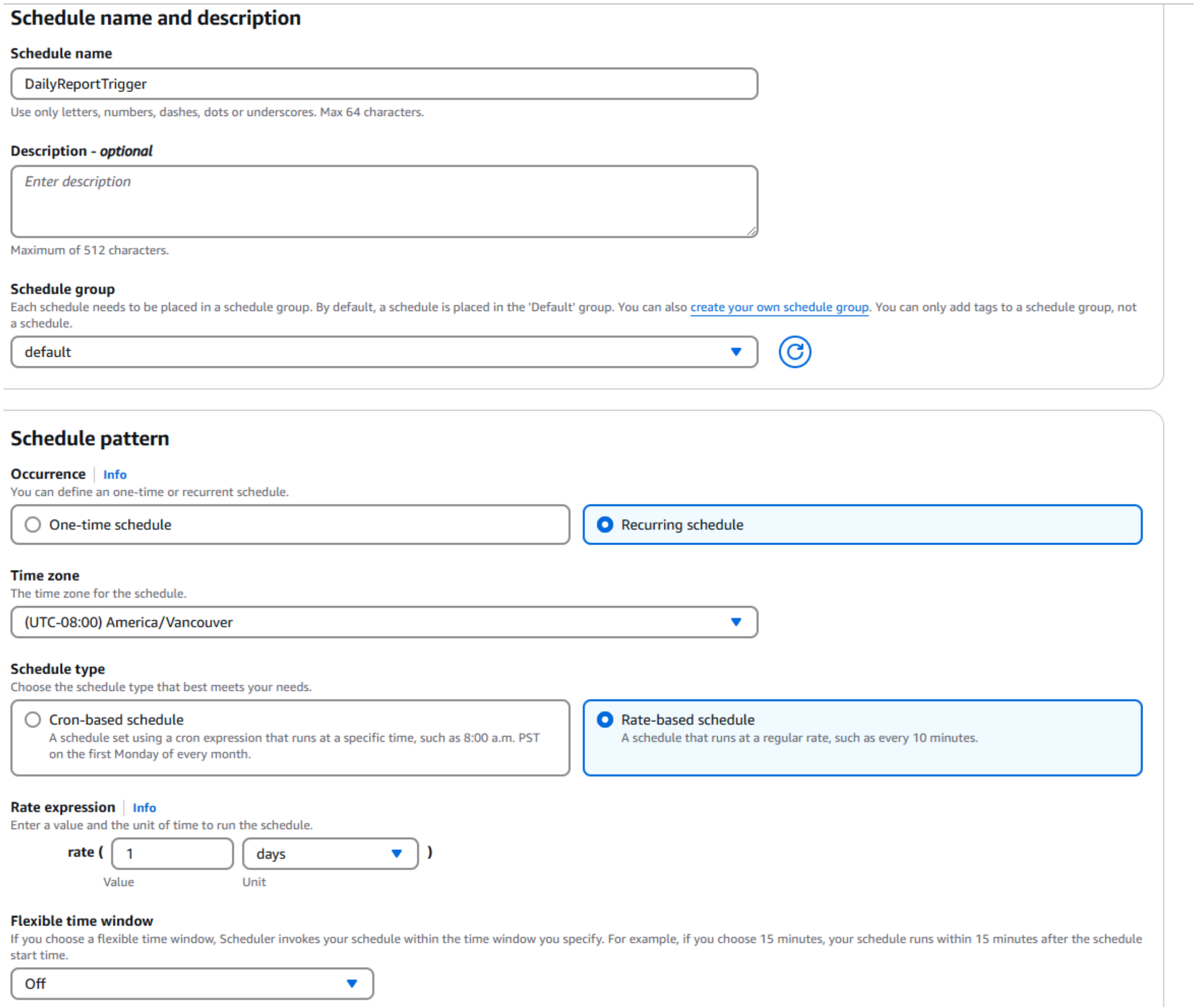
CIS 1.1 - Last root account usage: 2024-11-28T03:22:00Z
CIS 1.2 - IAM users with MFA disabled: User2, User3
CIS 1.12 - Root account access keys: No access keys are attached to the root account.

--

See the change on the lamba_function in the python script

Task 12 (5%): AWS Event bridge enables scheduling events to trigger AWS services such as the Lambda function. Explain the steps that need to be taken to receive a daily report from a script that you created in Task 9.report from a script that you created in Task 9.

Create a new schedule in EventBridge



Schedule name and description

Schedule name

DailyReportTrigger

Use only letters, numbers, dashes, dots or underscores. Max 64 characters.

Description - *optional*

Enter description

Maximum of 512 characters.

Schedule group

Each schedule needs to be placed in a schedule group. By default, a schedule is placed in the 'Default' group. You can also create your own schedule group. You can only add tags to a schedule group, not a schedule.

default

Schedule pattern

Occurrence | Info

You can define an one-time or recurrent schedule.

○ One-time schedule ● Recurring schedule

Time zone

The time zone for the schedule.

(UTC-08:00) America/Vancouver

Schedule type

Choose the schedule type that best meets your needs.

○ Cron-based schedule
A schedule set using a cron expression that runs at a specific time, such as 8:00 a.m. PST on the first Monday of every month.

● Rate-based schedule
A schedule that runs at a regular rate, such as every 10 minutes.

Rate expression | Info

Enter a value and the unit of time to run the schedule.

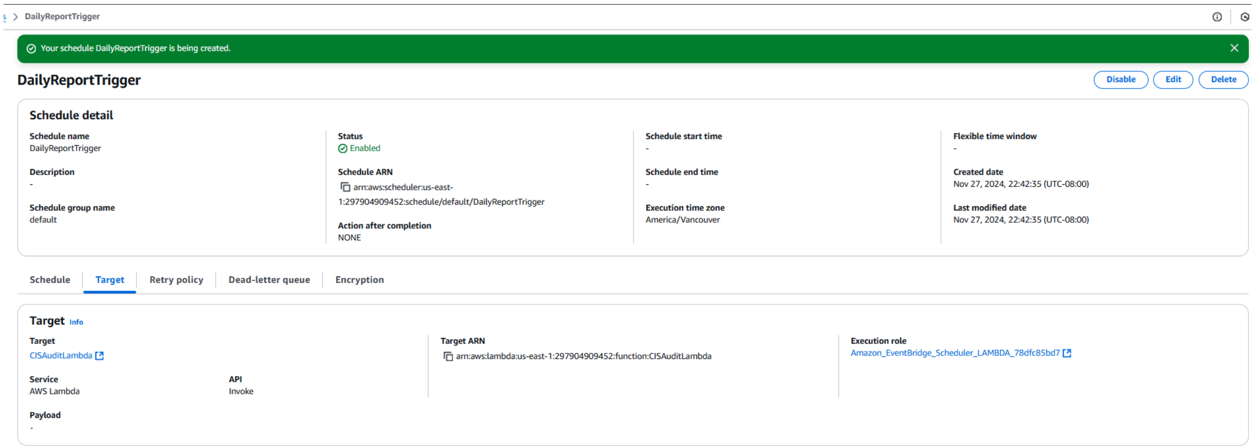
rate (1 days)

Value Unit

Flexible time window

If you choose a flexible time window, Scheduler invokes your schedule within the time window you specify. For example, if you choose 15 minutes, your schedule runs within 15 minutes after the schedule start time.

Off



> DailyReportTrigger

Your schedule DailyReportTrigger is being created.

DailyReportTrigger

Schedule detail

Schedule name
DailyReportTrigger

Description
-

Schedule group name
default

Status
Enabled

Schedule ARN
arn:aws:scheduler:us-east-1:297904909452:schedule/default/DailyReportTrigger

Action after completion
NONE

Schedule start time
-

Schedule end time
-

Execution time zone
America/Vancouver

Flexible time window
-

Created date
Nov 27, 2024, 22:42:35 (UTC-08:00)

Last modified date
Nov 27, 2024, 22:42:35 (UTC-08:00)

Target Info

Target
CISAuditLambda

Service
AWS Lambda

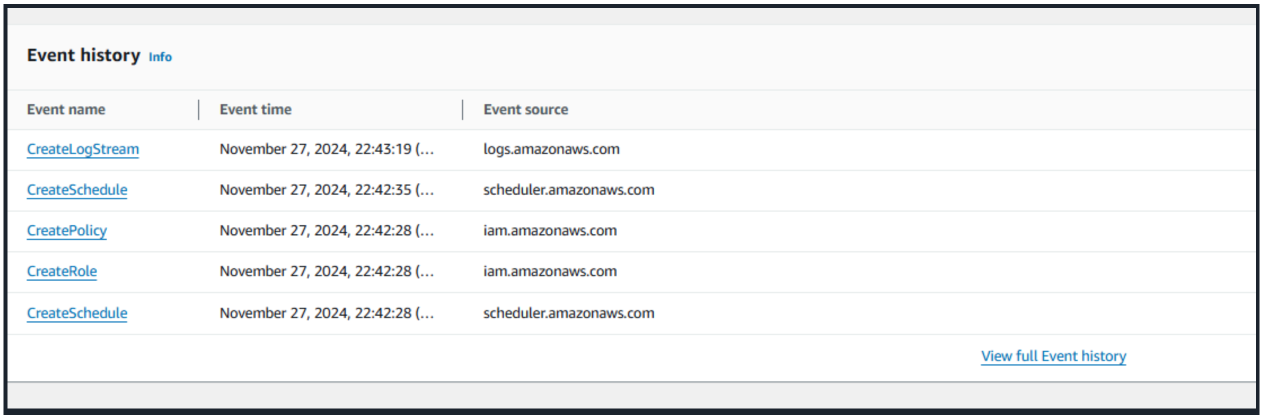
API
Invoke

Payload
-

Target ARN
arn:aws:lambda:us-east-1:297904909452:function:CISAuditLambda

Execution role
Amazon_EventBridge_Scheduler_LAMBDA_78dfc85bd7

Task 13 (10%): Use AWS CloudTrail-> event history to query the last time the root user logged in.



Event history Info

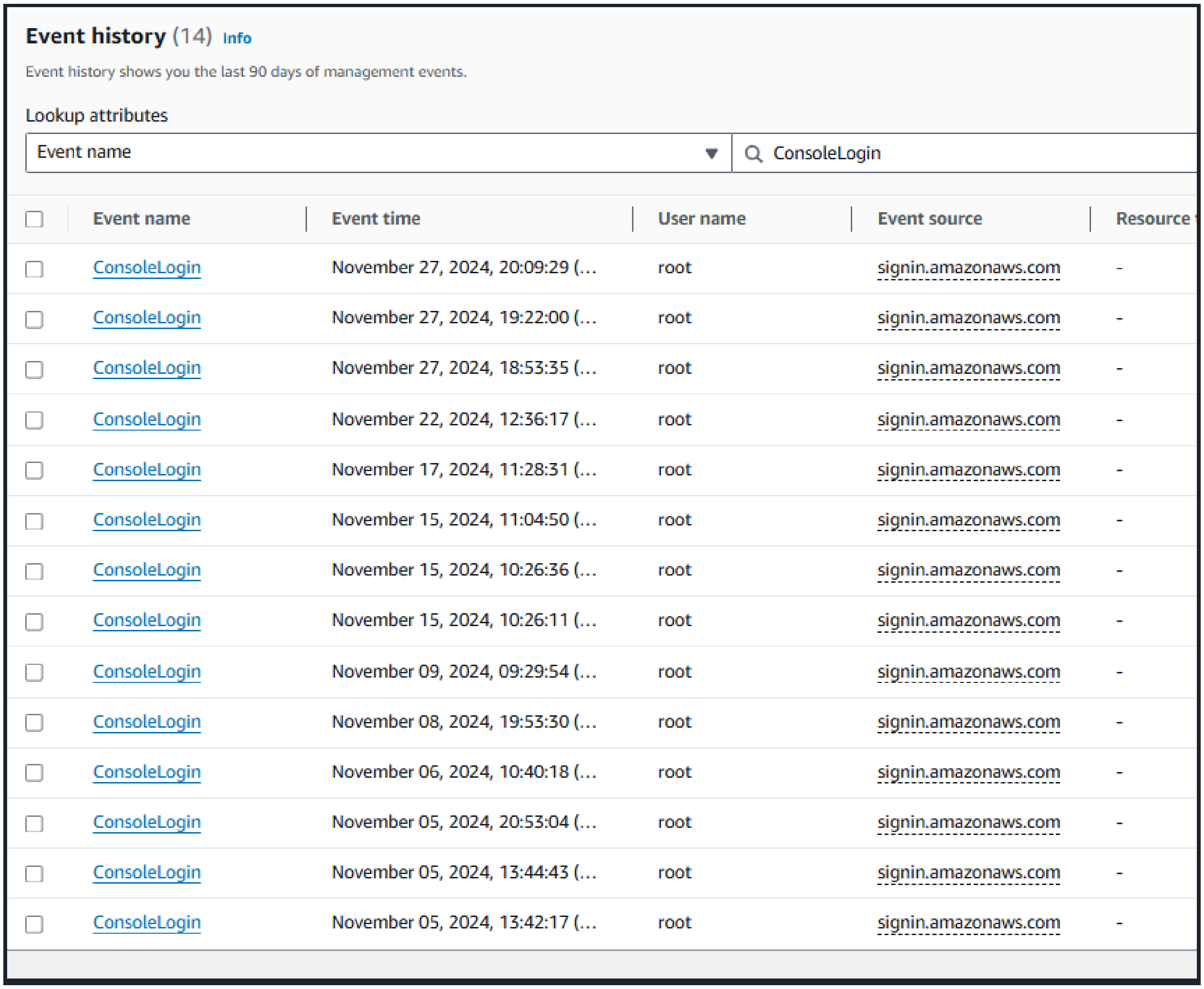
Event name	Event time	Event source
CreateLogStream	November 27, 2024, 22:43:19 (...	logs.amazonaws.com
CreateSchedule	November 27, 2024, 22:42:35 (...	scheduler.amazonaws.com
CreatePolicy	November 27, 2024, 22:42:28 (...	iam.amazonaws.com
CreateRole	November 27, 2024, 22:42:28 (...	iam.amazonaws.com
CreateSchedule	November 27, 2024, 22:42:28 (...	scheduler.amazonaws.com

View full Event history

Use the Search or filter events bar to filter by specific criteria:

- Event name: ConsoleLogin

We can see the root user activities.



Event history (14) Info

Event history shows you the last 90 days of management events.

Lookup attributes

Event name ▼ | ConsoleLogin

□	Event name	Event time	User name	Event source	Resource
□	ConsoleLogin	November 27, 2024, 20:09:29 (...	root	signin.amazonaws.com	-
□	ConsoleLogin	November 27, 2024, 19:22:00 (...	root	signin.amazonaws.com	-
□	ConsoleLogin	November 27, 2024, 18:53:35 (...	root	signin.amazonaws.com	-
□	ConsoleLogin	November 22, 2024, 12:36:17 (...	root	signin.amazonaws.com	-
□	ConsoleLogin	November 17, 2024, 11:28:31 (...	root	signin.amazonaws.com	-
□	ConsoleLogin	November 15, 2024, 11:04:50 (...	root	signin.amazonaws.com	-
□	ConsoleLogin	November 15, 2024, 10:26:36 (...	root	signin.amazonaws.com	-
□	ConsoleLogin	November 15, 2024, 10:26:11 (...	root	signin.amazonaws.com	-
□	ConsoleLogin	November 09, 2024, 09:29:54 (...	root	signin.amazonaws.com	-
□	ConsoleLogin	November 08, 2024, 19:53:30 (...	root	signin.amazonaws.com	-
□	ConsoleLogin	November 06, 2024, 10:40:18 (...	root	signin.amazonaws.com	-
□	ConsoleLogin	November 05, 2024, 20:53:04 (...	root	signin.amazonaws.com	-
□	ConsoleLogin	November 05, 2024, 13:44:43 (...	root	signin.amazonaws.com	-
□	ConsoleLogin	November 05, 2024, 13:42:17 (...	root	signin.amazonaws.com	-