AWS re:Start

LAB

Working with AWS CloudTrail

WEEK 11

aws re/start

Overview

AWS CloudTrail is a vital service for monitoring and auditing AWS account activity. By configuring a CloudTrail trail, you can capture and log API calls and actions across your AWS infrastructure, providing comprehensive visibility into all account activities. This is crucial for security and compliance, as it allows you to trace and audit actions performed on your AWS resources.

Importing CloudTrail log data into Amazon Athena enables you to run SQL-like queries for detailed analysis. This helps resolve security concerns by identifying unauthorized actions and unusual activity within your AWS account and on EC2 Linux instances. Using Athena to query CloudTrail logs allows for quick detection and response to potential security issues, ensuring your AWS environment remains secure and compliant.

Topics covered

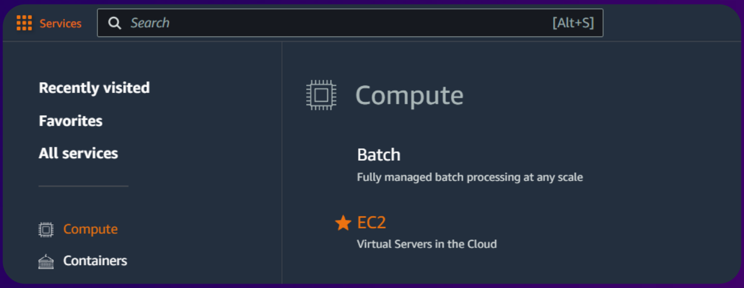
- Configure a CloudTrail trail
- Analyze CloudTrail logs by using various methods to discover relevant information
- Import CloudTrail log data into Athena
- Run queries in Athena to filter CloudTrail log entries
- Resolve security concerns within the AWS account and on an EC2 Linux instance

Task 1

Modifying a security group and observing the website

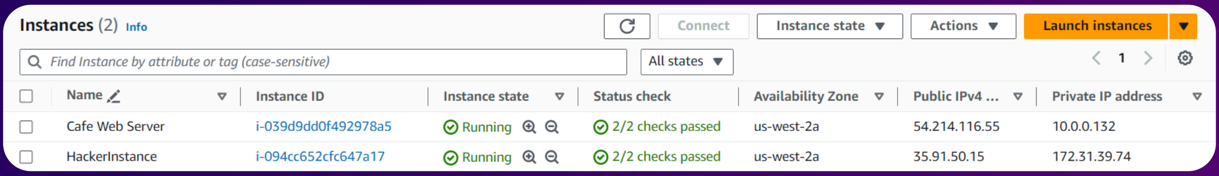
Step 1: Access the EC2 Management Console

Access the AWS Management Console, and select EC2.



Step 2: Review Instances

Navigate to the **Instances** section, and select the **Cafe Web Server** instance.



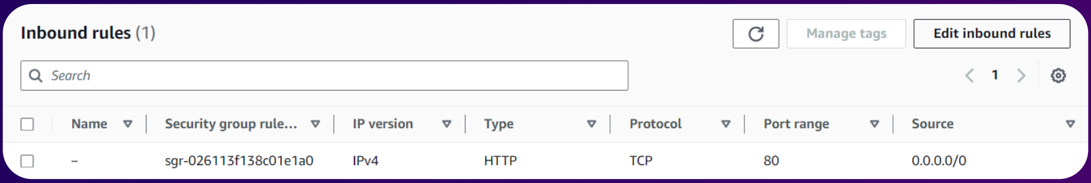
□	Name	Instance ID	Instance state	Status check	Availability Zone	Public IPv4 ...	Private IP address
□	Cafe Web Server	i-039d9dd0f492978a5	Running	2/2 checks passed	us-west-2a	54.214.116.55	10.0.0.132
□	HackerInstance	i-094cc652cfc647a17	Running	2/2 checks passed	us-west-2a	35.91.50.15	172.31.39.74

Task 1

Modifying a security group and observing the website

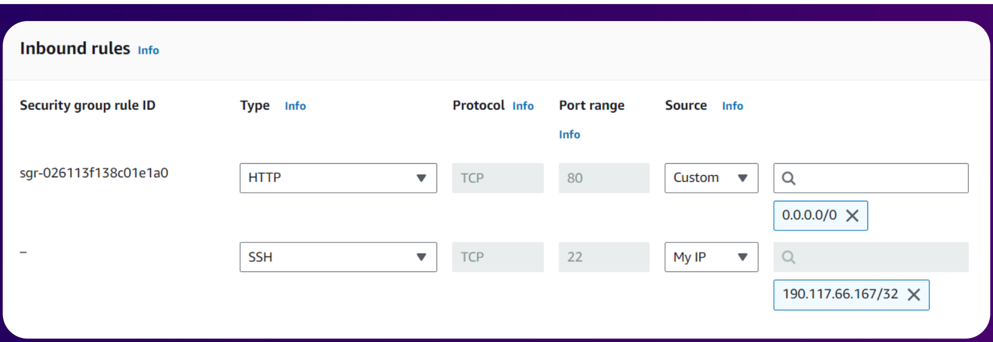
Step 3: Modify a Security Group

Review the **WebSecurityGroup**, and add a new inbound rule to allow SSH traffic only from your IP.



Inbound rules (1)

☐	Name	Security group rule...	IP version	Type	Protocol	Port range	Source
☐	–	sgr-026113f138c01e1a0	IPv4	HTTP	TCP	80	0.0.0.0/0

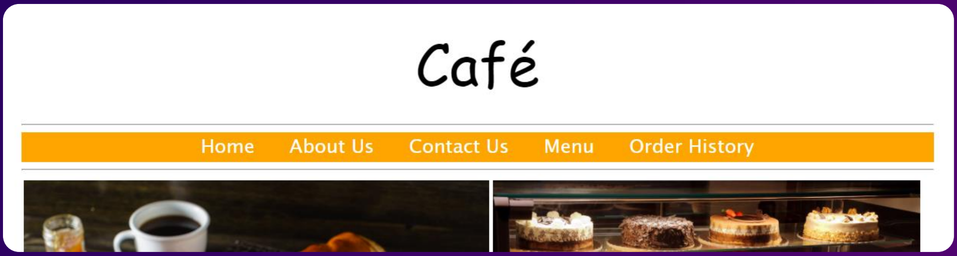


Inbound rules

Security group rule ID	Type	Protocol	Port range	Source	
sgr-026113f138c01e1a0	HTTP	TCP	80	Custom	0.0.0.0/0
–	SSH	TCP	22	My IP	190.117.66.167/32

Step 4: Observe the Café website

Access the Café website using the web server Public IPv4 address. Notice that the photos are all appropriate for a bakery café.

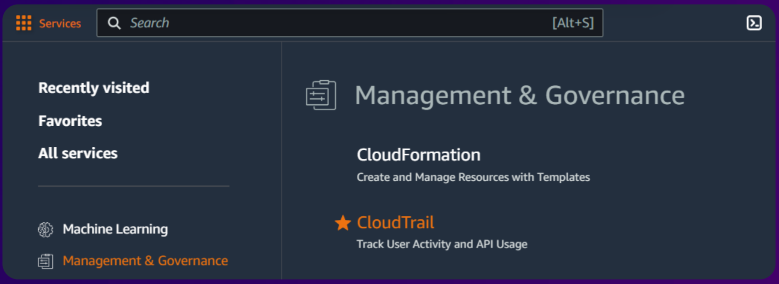


Task 2

Creating a CloudTrail log and observing the hacked website

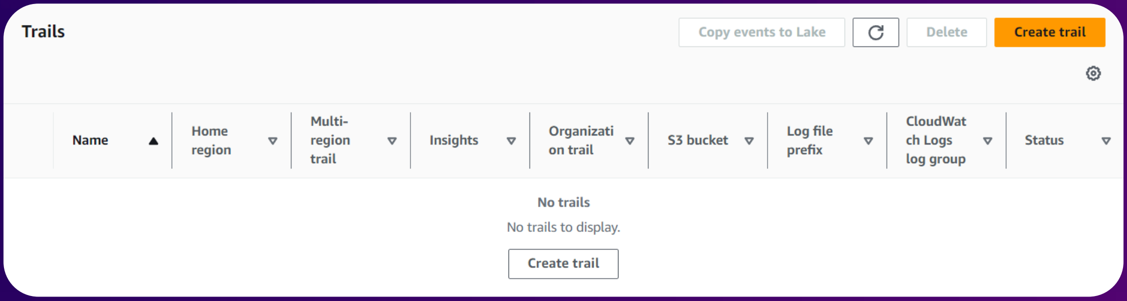
Step 1: Access the CloudTrail Console

In the AWS Management Console, select CloudTrail



Step 2: Create trail

Navigate to the **Trails** section, and select Create trail.



Name ▲	Home region ▽	Multi-region trail ▽	Insights ▽	Organization trail ▽	S3 bucket ▽	Log file prefix ▽	CloudWatch Logs log group ▽	Status ▽

No trails

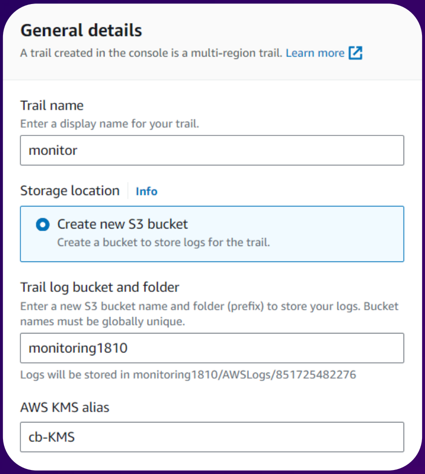
No trails to display.

Task 2

Creating a CloudTrail log and observing the hacked website

Step 3: General details

In the **General details** section, configure the following settings.



General details

A trail created in the console is a multi-region trail. Learn more

Trail name
Enter a display name for your trail.
monitor

Storage location Info

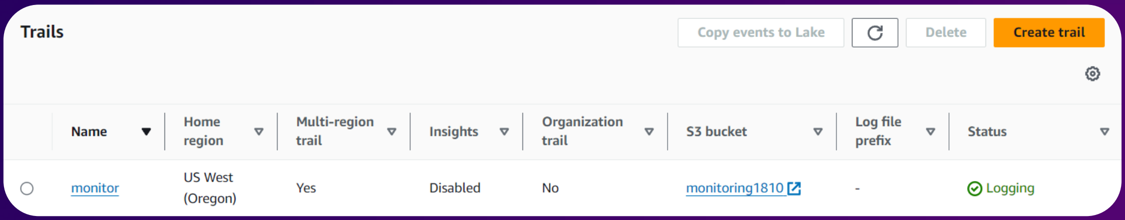
● Create new S3 bucket
Create a bucket to store logs for the trail.

Trail log bucket and folder
Enter a new S3 bucket name and folder (prefix) to store your logs. Bucket names must be globally unique.
monitoring1810
Logs will be stored in monitoring1810/AWSLogs/851725482276

AWS KMS alias
cb-KMS

Step 4: Review Trails

Verify that your see your newly created trail on the **Trails** page.



Trails

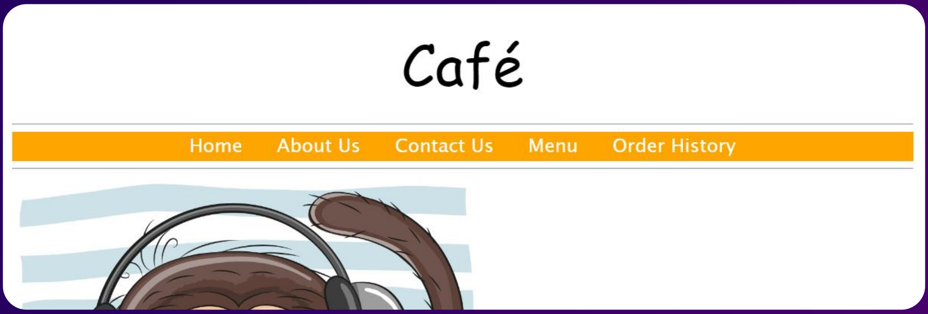
	Name	Home region	Multi-region trail	Insights	Organization trail	S3 bucket	Log file prefix	Status
○	monitor	US West (Oregon)	Yes	Disabled	No	monitoring1810	-	✓ Logging

Task 2

Creating a CloudTrail log and observing the hacked website

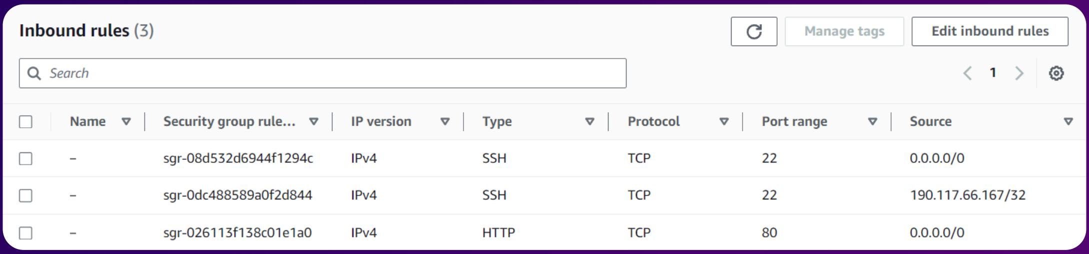
Step 5: Refresh the website

Notice that the website has been hacked. That image certainly does not look correct.



Step 6: Review Security Group

Review the **WebSecurityGroup** inbound rules. Someone else created an additional inbound rule that allows Secure Shell (SSH) access from anywhere (0.0.0.0/0).



Inbound rules (3)

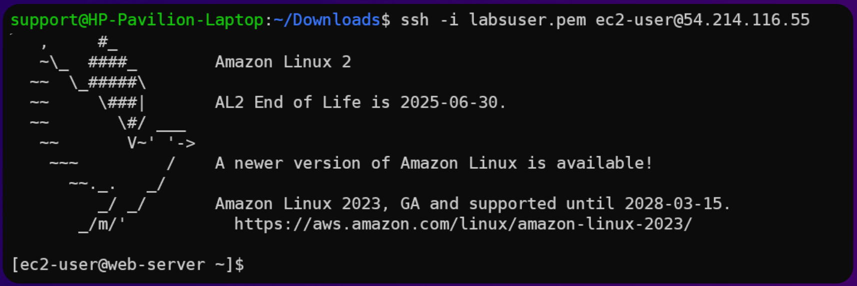
□	Name	Security group rule...	IP version	Type	Protocol	Port range	Source
□	–	sgr-08d532d6944f1294c	IPv4	SSH	TCP	22	0.0.0.0/0
□	–	sgr-0dc488589a0f2d844	IPv4	SSH	TCP	22	190.117.66.167/32
□	–	sgr-026113f138c01e1a0	IPv4	HTTP	TCP	80	0.0.0.0/0

Task 3

Analyzing the CloudTrail logs by using grep

Step 1: Connect to the Web Server

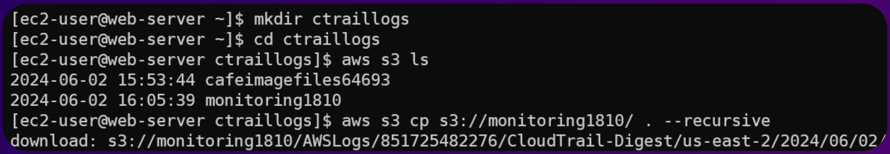
Connect to the **Cafe Web Server** host EC2 instance by using SSH.



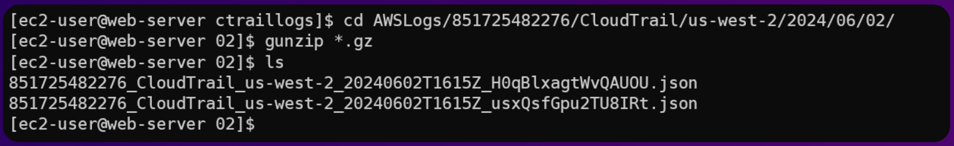
```
support@HP-Pavilion-Laptop:~/Downloads$ ssh -i labsuser.pem ec2-user@54.214.116.55
`
   ,     #_
   ~\_  ####_        Amazon Linux 2
  ~~  \_#####\
  ~~     \###|       AL2 End of Life is 2025-06-30.
  ~~       \#/ ___
   ~~       V~' '->
    ~~~         /    A newer version of Amazon Linux is available!
      ~~._.   _/
         _/ _/       Amazon Linux 2023, GA and supported until 2028-03-15.
       _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@web-server ~]$
```

Step 2: Download the CloudTrail logs

To download and extract the CloudTrail logs, run the following commands.



```
[ec2-user@web-server ~]$ mkdir ctraillogs
[ec2-user@web-server ~]$ cd ctraillogs
[ec2-user@web-server ctraillogs]$ aws s3 ls
2024-06-02 15:53:44 cafeimagefiles64693
2024-06-02 16:05:39 monitoring1810
[ec2-user@web-server ctraillogs]$ aws s3 cp s3://monitoring1810/ . --recursive
download: s3://monitoring1810/AWSLogs/851725482276/CloudTrail-Digest/us-east-2/2024/06/02/
```



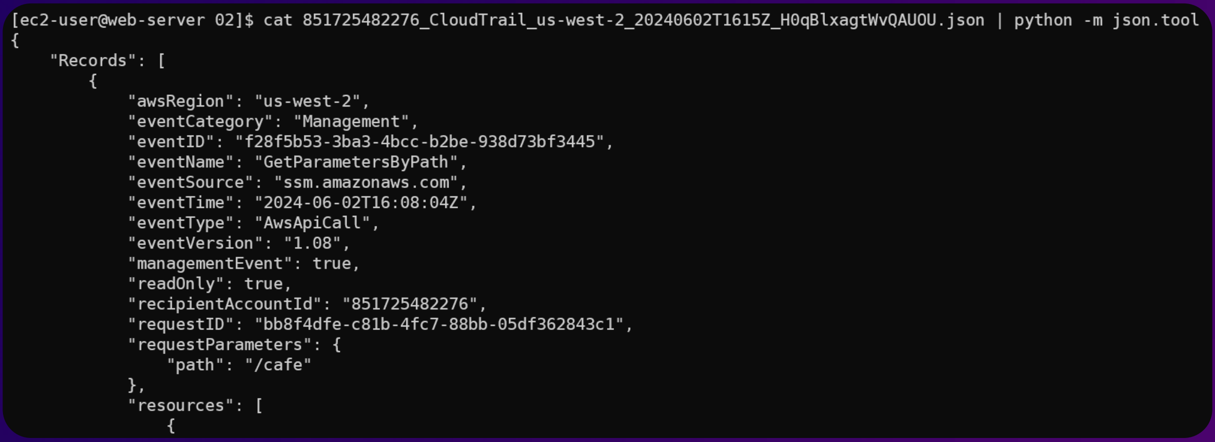
```
[ec2-user@web-server ctraillogs]$ cd AWSLogs/851725482276/CloudTrail/us-west-2/2024/06/02/
[ec2-user@web-server 02]$ gunzip *.gz
[ec2-user@web-server 02]$ ls
851725482276_CloudTrail_us-west-2_20240602T1615Z_H0qBlxagtWvQAUOU.json
851725482276_CloudTrail_us-west-2_20240602T1615Z_usxQsfGpu2TU8IRt.json
[ec2-user@web-server 02]$
```

Task 3

Analyzing the CloudTrail logs by using grep

Step 3: Analyze the structure of the logs

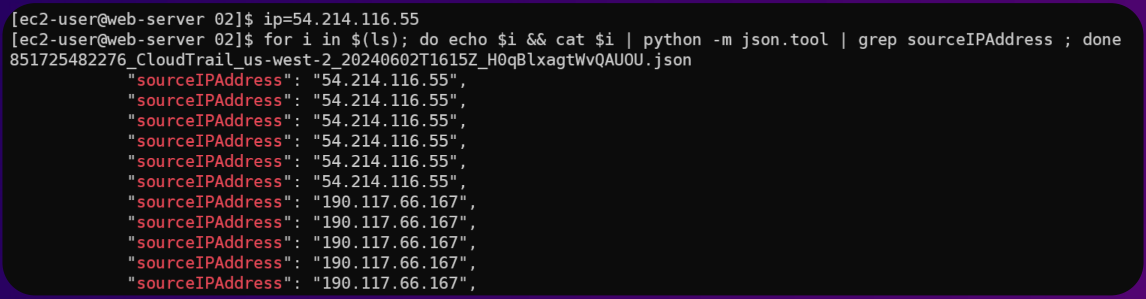
To analyze the structure of the logs, run the following command.



```
[ec2-user@web-server 02]$ cat 851725482276_CloudTrail_us-west-2_20240602T1615Z_H0qBlxagtWvQAUOU.json | python -m json.tool
{
    "Records": [
        {
            "awsRegion": "us-west-2",
            "eventCategory": "Management",
            "eventID": "f28f5b53-3ba3-4bcc-b2be-938d73bf3445",
            "eventName": "GetParametersByPath",
            "eventSource": "ssm.amazonaws.com",
            "eventTime": "2024-06-02T16:08:04Z",
            "eventType": "AwsApiCall",
            "eventVersion": "1.08",
            "managementEvent": true,
            "readOnly": true,
            "recipientAccountId": "851725482276",
            "requestID": "bb8f4dfe-c81b-4fc7-88bb-05df362843c1",
            "requestParameters": {
                "path": "/cafe"
            },
            "resources": [
                {
```

Step 4: Review sourceIPAddress

Filter the log results where the sourceIpAddress matches the IP address of the Cafe Web Server instance.



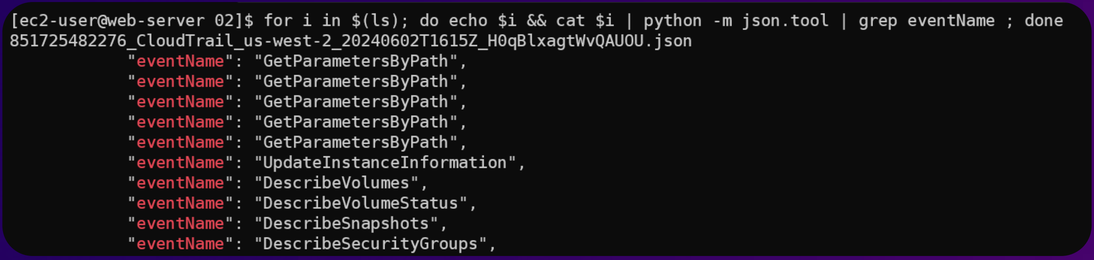
```
[ec2-user@web-server 02]$ ip=54.214.116.55
[ec2-user@web-server 02]$ for i in $(ls); do echo $i && cat $i | python -m json.tool | grep sourceIPAddress ; done
851725482276_CloudTrail_us-west-2_20240602T1615Z_H0qBlxagtWvQAUOU.json
            "sourceIPAddress": "54.214.116.55",
            "sourceIPAddress": "54.214.116.55",
            "sourceIPAddress": "54.214.116.55",
            "sourceIPAddress": "54.214.116.55",
            "sourceIPAddress": "54.214.116.55",
            "sourceIPAddress": "54.214.116.55",
            "sourceIPAddress": "190.117.66.167",
            "sourceIPAddress": "190.117.66.167",
            "sourceIPAddress": "190.117.66.167",
            "sourceIPAddress": "190.117.66.167",
            "sourceIPAddress": "190.117.66.167",
```

Task 3

Analyzing the CloudTrail logs by using grep

Step 5: Review eventName

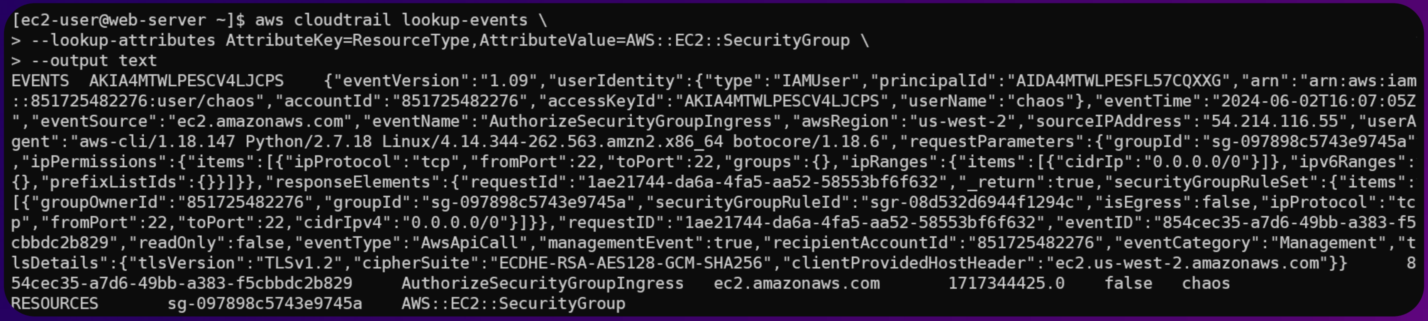
Run a similarly structured command but where the command returns the eventName of every captured event.



```
[ec2-user@web-server 02]$ for i in $(ls); do echo $i && cat $i | python -m json.tool | grep eventName ; done
851725482276_CloudTrail_us-west-2_20240602T1615Z_H0qBlxagtWvQAUOU.json
            "eventName": "GetParametersByPath",
            "eventName": "GetParametersByPath",
            "eventName": "GetParametersByPath",
            "eventName": "GetParametersByPath",
            "eventName": "GetParametersByPath",
            "eventName": "UpdateInstanceInformation",
            "eventName": "DescribeVolumes",
            "eventName": "DescribeVolumeStatus",
            "eventName": "DescribeSnapshots",
            "eventName": "DescribeSecurityGroups",
```

Step 6: Analyze the logs using AWS CLI

Run the following command to find any actions that were taken on security groups in the AWS account.



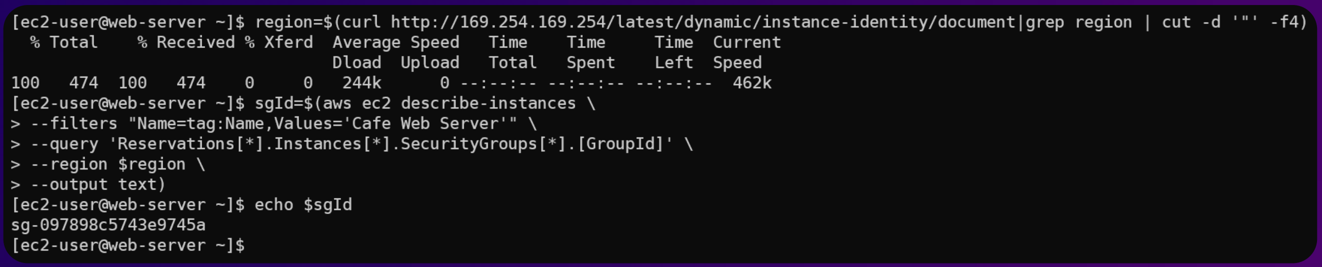
```
[ec2-user@web-server ~]$ aws cloudtrail lookup-events \
> --lookup-attributes AttributeKey=ResourceType,AttributeValue=AWS::EC2::SecurityGroup \
> --output text
EVENTS  AKIA4MTWLPESCV4LJCPS    {"eventVersion":"1.09","userIdentity":{"type":"IAMUser","principalId":"AIDA4MTWLPESFL57CQXXG","arn":"arn:aws:iam
::851725482276:user/chaos","accountId":"851725482276","accessKeyId":"AKIA4MTWLPESCV4LJCPS","userName":"chaos"},"eventTime":"2024-06-02T16:07:05Z
","eventSource":"ec2.amazonaws.com","eventName":"AuthorizeSecurityGroupIngress","awsRegion":"us-west-2","sourceIPAddress":"54.214.116.55","userA
gent":"aws-cli/1.18.147 Python/2.7.18 Linux/4.14.344-262.563.amzn2.x86_64 botocore/1.18.6","requestParameters":{"groupId":"sg-097898c5743e9745a"
,"ipPermissions":{"items":[{"ipProtocol":"tcp","fromPort":22,"toPort":22,"groups":{},"ipRanges":{"items":[{"cidrIp":"0.0.0.0/0"}]},"ipv6Ranges":
{},"prefixListIds":{}}]}},"responseElements":{"requestId":"1ae21744-da6a-4fa5-aa52-58553bf6f632","_return":true,"securityGroupRuleSet":{"items":
[{"groupOwnerId":"851725482276","groupId":"sg-097898c5743e9745a","securityGroupRuleId":"sgr-08d532d6944f1294c","isEgress":false,"ipProtocol":"tc
p","fromPort":22,"toPort":22,"cidrIpv4":"0.0.0.0/0"}]}},"requestID":"1ae21744-da6a-4fa5-aa52-58553bf6f632","eventID":"854cec35-a7d6-49bb-a383-f5
cbbdc2b829","readOnly":false,"eventType":"AwsApiCall","managementEvent":true,"recipientAccountId":"851725482276","eventCategory":"Management","t
lsDetails":{"tlsVersion":"TLSv1.2","cipherSuite":"ECDHE-RSA-AES128-GCM-SHA256","clientProvidedHostHeader":"ec2.us-west-2.amazonaws.com"}}      8
54cec35-a7d6-49bb-a383-f5cbbdc2b829     AuthorizeSecurityGroupIngress   ec2.amazonaws.com       1717344425.0    false   chaos
RESOURCES       sg-097898c5743e9745a    AWS::EC2::SecurityGroup
```

Task 3

Analyzing the CloudTrail logs by using grep

Step 7: Find the Security Group ID

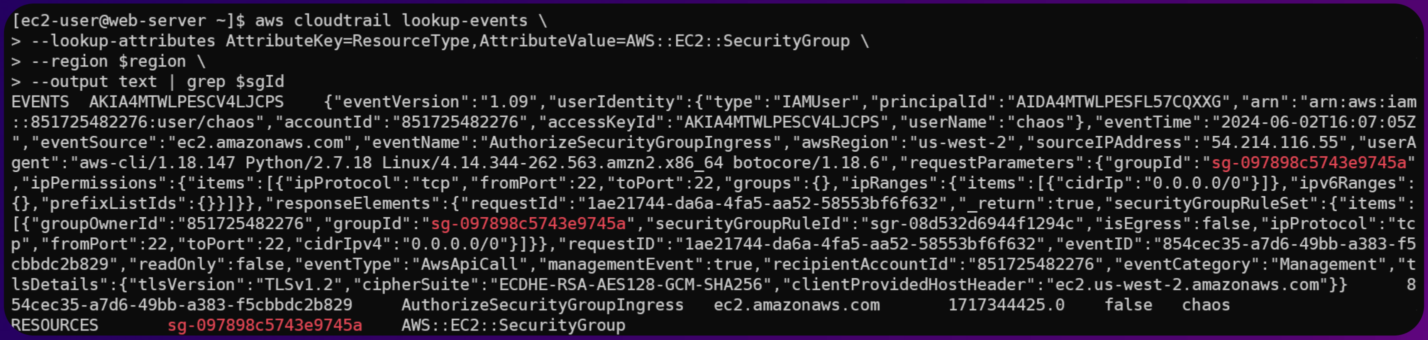
Run the following commands to find the security group ID that is used by the **Cafe Web Server** instance



```
[ec2-user@web-server ~]$ region=$(curl http://169.254.169.254/latest/dynamic/instance-identity/document|grep region | cut -d '"' -f4)
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   474  100   474    0     0   244k      0 --:--:-- --:--:-- --:--:--  462k
[ec2-user@web-server ~]$ sgId=$(aws ec2 describe-instances \
> --filters "Name=tag:Name,Values='Cafe Web Server'" \
> --query 'Reservations[*].Instances[*].SecurityGroups[*].[GroupId]' \
> --region $region \
> --output text)
[ec2-user@web-server ~]$ echo $sgId
sg-097898c5743e9745a
[ec2-user@web-server ~]$
```

Step 8: Filter command results

Use the security group ID that the previous command returned to further filter your AWS CLI CloudTrail command results.



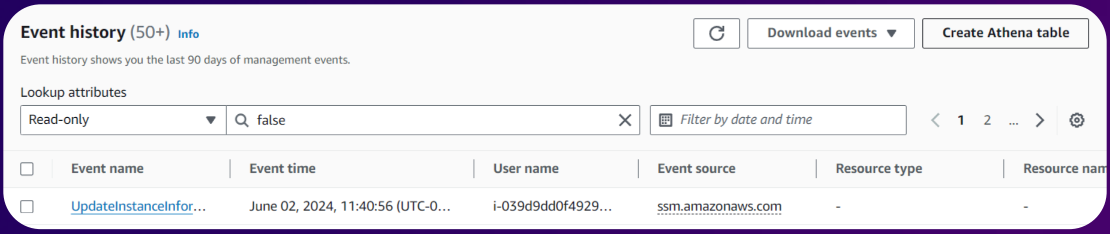
```
[ec2-user@web-server ~]$ aws cloudtrail lookup-events \
> --lookup-attributes AttributeKey=ResourceType,AttributeValue=AWS::EC2::SecurityGroup \
> --region $region \
> --output text | grep $sgId
EVENTS  AKIA4MTWLPESCV4LJCPS    {"eventVersion":"1.09","userIdentity":{"type":"IAMUser","principalId":"AIDA4MTWLPESFL57CQXXG","arn":"arn:aws:iam
::851725482276:user/chaos","accountId":"851725482276","accessKeyId":"AKIA4MTWLPESCV4LJCPS","userName":"chaos"},"eventTime":"2024-06-02T16:07:05Z
","eventSource":"ec2.amazonaws.com","eventName":"AuthorizeSecurityGroupIngress","awsRegion":"us-west-2","sourceIPAddress":"54.214.116.55","userA
gent":"aws-cli/1.18.147 Python/2.7.18 Linux/4.14.344-262.563.amzn2.x86_64 botocore/1.18.6","requestParameters":{"groupId":"sg-097898c5743e9745a"
,"ipPermissions":{"items":[{"ipProtocol":"tcp","fromPort":22,"toPort":22,"groups":{},"ipRanges":{"items":[{"cidrIp":"0.0.0.0/0"}]},"ipv6Ranges":
{},"prefixListIds":{}}]}},"responseElements":{"requestId":"1ae21744-da6a-4fa5-aa52-58553bf6f632","_return":true,"securityGroupRuleSet":{"items":
[{"groupOwnerId":"851725482276","groupId":"sg-097898c5743e9745a","securityGroupRuleId":"sgr-08d532d6944f1294c","isEgress":false,"ipProtocol":"tc
p","fromPort":22,"toPort":22,"cidrIpv4":"0.0.0.0/0"}]}},"requestID":"1ae21744-da6a-4fa5-aa52-58553bf6f632","eventID":"854cec35-a7d6-49bb-a383-f5
cbbdc2b829","readOnly":false,"eventType":"AwsApiCall","managementEvent":true,"recipientAccountId":"851725482276","eventCategory":"Management","t
lsDetails":{"tlsVersion":"TLSv1.2","cipherSuite":"ECDHE-RSA-AES128-GCM-SHA256","clientProvidedHostHeader":"ec2.us-west-2.amazonaws.com"}}      8
54cec35-a7d6-49bb-a383-f5cbbdc2b829     AuthorizeSecurityGroupIngress   ec2.amazonaws.com       1717344425.0    false   chaos
RESOURCES       sg-097898c5743e9745a    AWS::EC2::SecurityGroup
```

Task 4

Analyzing the CloudTrail logs by using Athena

Step 1: Create Athena table

In the CloudTrail console, navigate to the **Event history** section, and select Create Athena table.



Event history (50+) Info

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

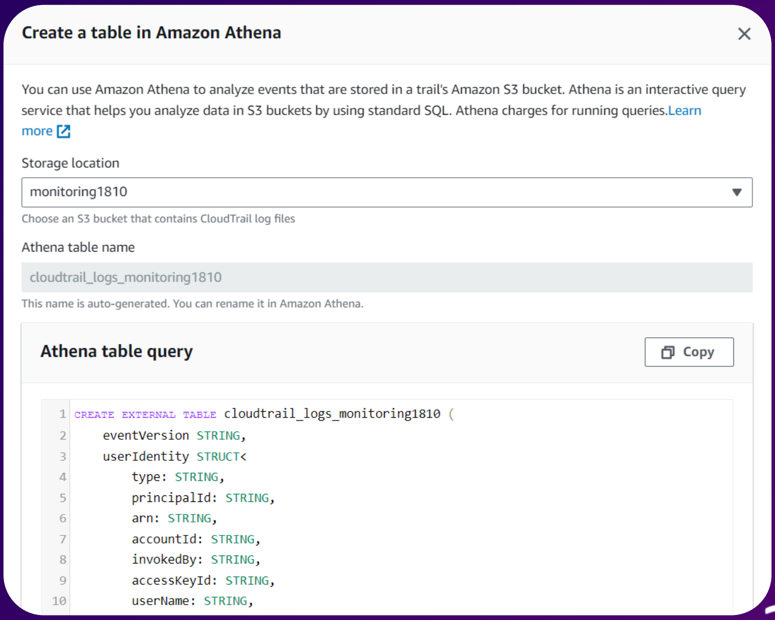
Lookup attributes

Read-only ▼ | false

☐	Event name	Event time	User name	Event source	Resource type	Resource nam
☐	UpdateInstanceInfor…	June 02, 2024, 11:40:56 (UTC-0…	i-039d9dd0f4929…	ssm.amazonaws.com	-	-

Step 2: Create a table in Amazon Athena

In the **Create a table in Amazon Athena** section, configure the following settings, and review the Athena table query.



Create a table in Amazon Athena

You can use Amazon Athena to analyze events that are stored in a trail's Amazon S3 bucket. Athena is an interactive query service that helps you analyze data in S3 buckets by using standard SQL. Athena charges for running queries.Learn more

Storage location: monitoring1810

Choose an S3 bucket that contains CloudTrail log files

Athena table name: cloudtrail_logs_monitoring1810

This name is auto-generated. You can rename it in Amazon Athena.

Athena table query

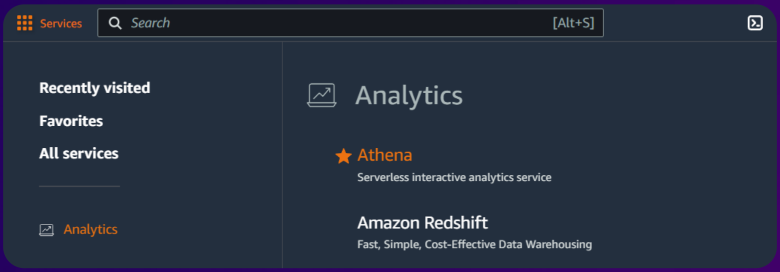
```
CREATE EXTERNAL TABLE cloudtrail_logs_monitoring1810 (
    eventVersion STRING,
    userIdentity STRUCT<
        type: STRING,
        principalId: STRING,
        arn: STRING,
        accountId: STRING,
        invokedBy: STRING,
        accessKeyId: STRING,
        userName: STRING,
```

Task 4

Analyzing the CloudTrail logs by using Athena

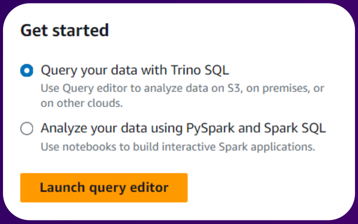
Step 3: Access the Athena service

In the AWS Management Console, select Athena.



Step 4: Launch query editor

In the Get started section, select Launch query editor.



Get started

● Query your data with Trino SQL
Use Query editor to analyze data on S3, on premises, or on other clouds.

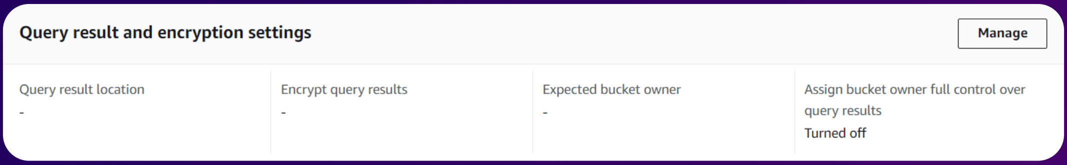
○ Analyze your data using PySpark and Spark SQL
Use notebooks to build interactive Spark applications.

Task 4

Analyzing the CloudTrail logs by using Athena

Step 5: Manage settings

Choose the **Settings** tab, and in the **Query result and encryption settings** section, choose Manage.

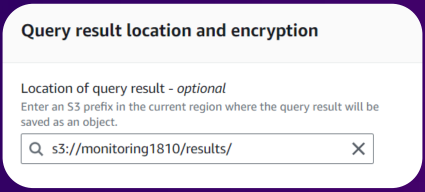


Query result and encryption settings

Query result location	Encrypt query results	Expected bucket owner	Assign bucket owner full control over query results
-	-	-	Turned off

Step 6: Query result location

In the **Query result location and encryption** section, configure the following settings.



Query result location and encryption

Location of query result - *optional*

Enter an S3 prefix in the current region where the query result will be saved as an object.

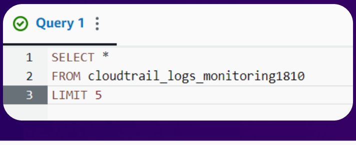
s3://monitoring1810/results/

Task 4

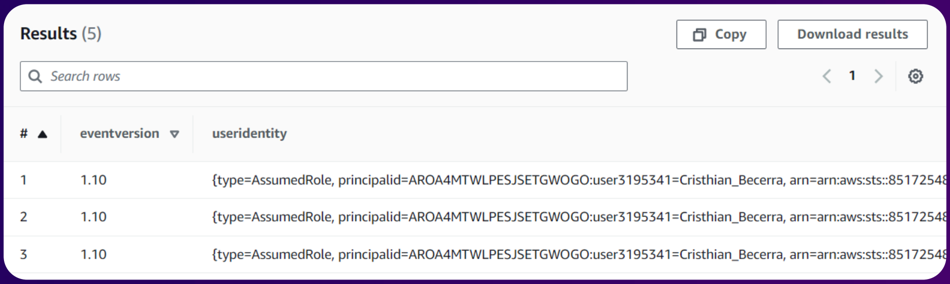
Analyzing the CloudTrail logs by using Athena

Step 7: Run a first SQL query

Run a first SQL query in the Athena Query Editor.



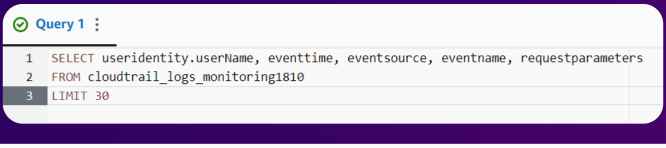
```
SELECT *
FROM cloudtrail_logs_monitoring1810
LIMIT 5
```



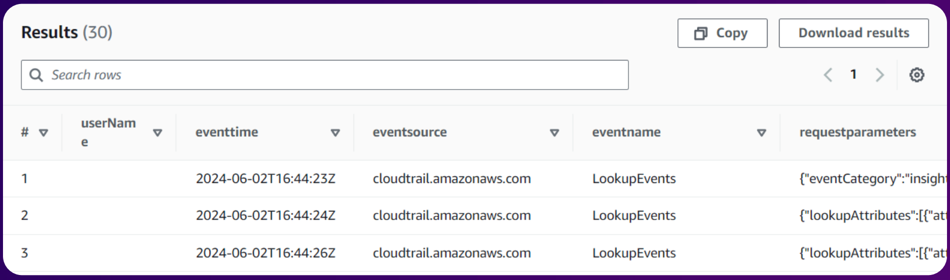
#	eventversion	useridentity
1	1.10	{type=AssumedRole, principalid=AROA4MTWLPESJSETGWOGO:user3195341=Cristhian_Becerra, arn=arn:aws:sts::85172548
2	1.10	{type=AssumedRole, principalid=AROA4MTWLPESJSETGWOGO:user3195341=Cristhian_Becerra, arn=arn:aws:sts::85172548
3	1.10	{type=AssumedRole, principalid=AROA4MTWLPESJSETGWOGO:user3195341=Cristhian_Becerra, arn=arn:aws:sts::85172548

Step 8: Run a second SQL query

Run a second SQL query in the Athena Query Editor.



```
SELECT useridentity.userName, eventtime, eventsource, eventname, requestparameters
FROM cloudtrail_logs_monitoring1810
LIMIT 30
```



#	userName	eventtime	eventsource	eventname	requestparameters
1		2024-06-02T16:44:23Z	cloudtrail.amazonaws.com	LookupEvents	{"eventCategory":"insight
2		2024-06-02T16:44:24Z	cloudtrail.amazonaws.com	LookupEvents	{"lookupAttributes":[{"at
3		2024-06-02T16:44:26Z	cloudtrail.amazonaws.com	LookupEvents	{"lookupAttributes":[{"at

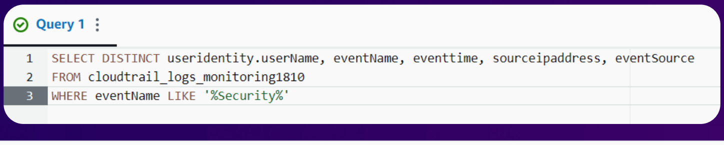
aws re/start

Task 4

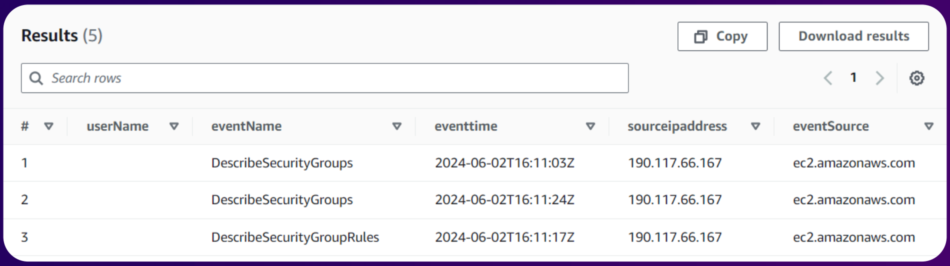
Analyzing the CloudTrail logs by using Athena

Step 9: Run a third SQL query

Run a third SQL query in the Athena Query Editor.



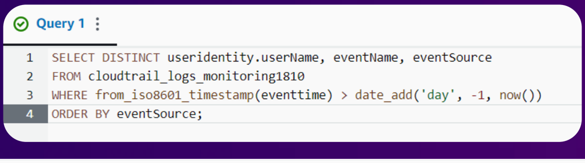
```sql
SELECT DISTINCT useridentity.userName, eventName, eventtime, sourceipaddress, eventSource
FROM cloudtrail_logs_monitoring1810
WHERE eventName LIKE '%Security%'
```



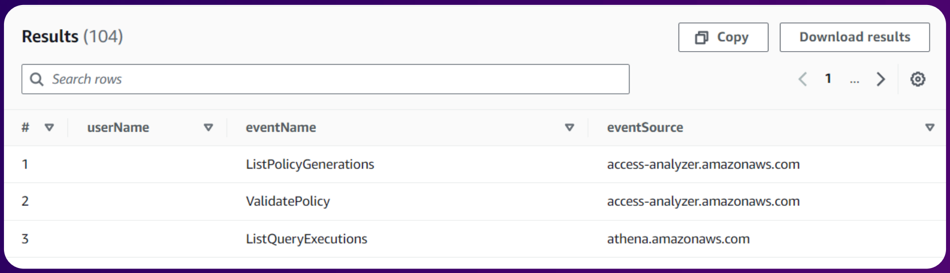
#	userName	eventName	eventtime	sourceipaddress	eventSource
1		DescribeSecurityGroups	2024-06-02T16:11:03Z	190.117.66.167	ec2.amazonaws.com
2		DescribeSecurityGroups	2024-06-02T16:11:24Z	190.117.66.167	ec2.amazonaws.com
3		DescribeSecurityGroupRules	2024-06-02T16:11:17Z	190.117.66.167	ec2.amazonaws.com

Step 10: Run a fourth SQL query

Run a fourth SQL query in the Athena Query Editor.



```sql
SELECT DISTINCT useridentity.userName, eventName, eventSource
FROM cloudtrail_logs_monitoring1810
WHERE from_iso8601_timestamp(eventtime) > date_add('day', -1, now())
ORDER BY eventSource;
```



#	userName	eventName	eventSource
1		ListPolicyGenerations	access-analyzer.amazonaws.com
2		ValidatePolicy	access-analyzer.amazonaws.com
3		ListQueryExecutions	athena.amazonaws.com

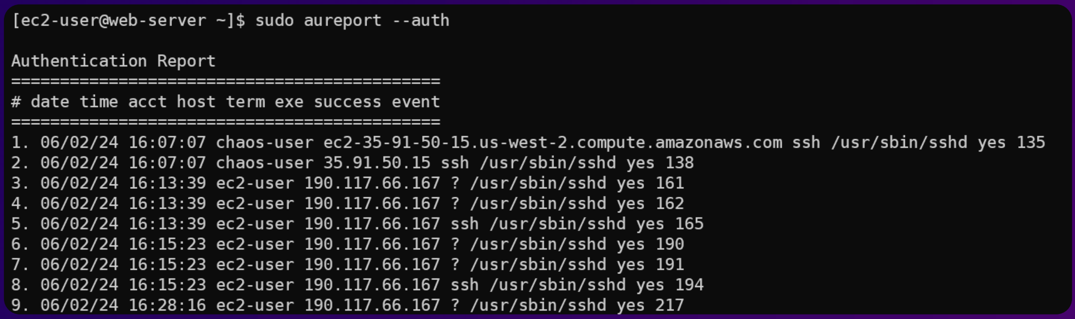
aws re/start

Task 5

Analyzing the hack further and improving security

Step 1: Review Authentication Report

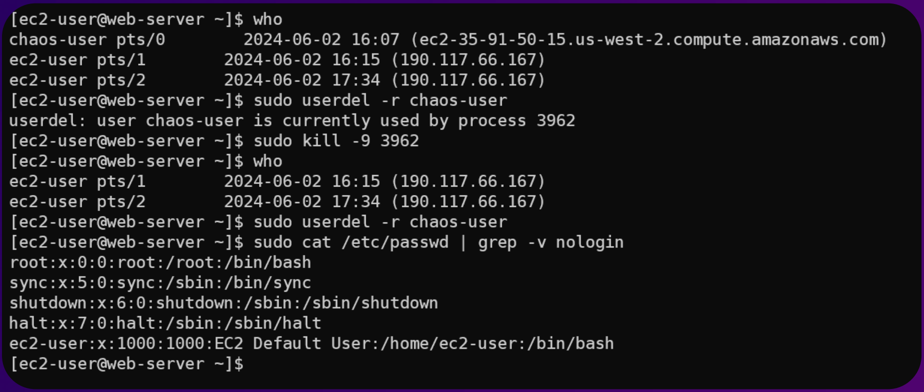
Run the following command to find out who has recently logged into this operating system (OS).



```
[ec2-user@web-server ~]$ sudo aureport --auth

Authentication Report
============================================
# date time acct host term exe success event
============================================
1. 06/02/24 16:07:07 chaos-user ec2-35-91-50-15.us-west-2.compute.amazonaws.com ssh /usr/sbin/sshd yes 135
2. 06/02/24 16:07:07 chaos-user 35.91.50.15 ssh /usr/sbin/sshd yes 138
3. 06/02/24 16:13:39 ec2-user 190.117.66.167 ? /usr/sbin/sshd yes 161
4. 06/02/24 16:13:39 ec2-user 190.117.66.167 ? /usr/sbin/sshd yes 162
5. 06/02/24 16:13:39 ec2-user 190.117.66.167 ssh /usr/sbin/sshd yes 165
6. 06/02/24 16:15:23 ec2-user 190.117.66.167 ? /usr/sbin/sshd yes 190
7. 06/02/24 16:15:23 ec2-user 190.117.66.167 ? /usr/sbin/sshd yes 191
8. 06/02/24 16:15:23 ec2-user 190.117.66.167 ssh /usr/sbin/sshd yes 194
9. 06/02/24 16:28:16 ec2-user 190.117.66.167 ? /usr/sbin/sshd yes 217
```

Step 2: Delete the chaos-user

Run the following commands to stop the process that has the active chaos-user login session, delete the chaos-user, and to verify no other suspicious OS users who can login.



```
[ec2-user@web-server ~]$ who
chaos-user pts/0        2024-06-02 16:07 (ec2-35-91-50-15.us-west-2.compute.amazonaws.com)
ec2-user pts/1        2024-06-02 16:15 (190.117.66.167)
ec2-user pts/2        2024-06-02 17:34 (190.117.66.167)
[ec2-user@web-server ~]$ sudo userdel -r chaos-user
userdel: user chaos-user is currently used by process 3962
[ec2-user@web-server ~]$ sudo kill -9 3962
[ec2-user@web-server ~]$ who
ec2-user pts/1        2024-06-02 16:15 (190.117.66.167)
ec2-user pts/2        2024-06-02 17:34 (190.117.66.167)
[ec2-user@web-server ~]$ sudo userdel -r chaos-user
[ec2-user@web-server ~]$ sudo cat /etc/passwd | grep -v nologin
root:x:0:0:root:/root:/bin/bash
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
ec2-user:x:1000:1000:EC2 Default User:/home/ec2-user:/bin/bash
[ec2-user@web-server ~]$
```

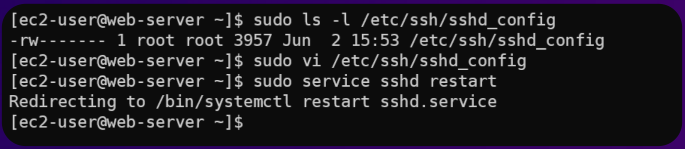
aws re/start

Task 5

Analyzing the hack further and improving security

Step 3: Edit SSH settings

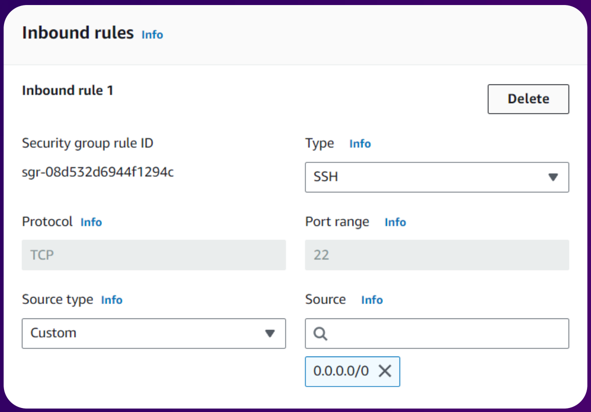
Check the SSH settings on this instance. Notice the last modified timestamp for the file. This file was modified today. Edit the SSH configuration file, disable password authentication, and restart the SSH service so that the changes go into effect.



```
[ec2-user@web-server ~]$ sudo ls -l /etc/ssh/sshd_config
-rw------- 1 root root 3957 Jun  2 15:53 /etc/ssh/sshd_config
[ec2-user@web-server ~]$ sudo vi /etc/ssh/sshd_config
[ec2-user@web-server ~]$ sudo service sshd restart
Redirecting to /bin/systemctl restart sshd.service
[ec2-user@web-server ~]$
```

Step 4: Delete the inbound rule

Delete the **WebSecurityGroup** inbound rule that allows port 22 access from 0.0.0.0/0 (the one the hacker created).



Inbound rules Info

Inbound rule 1

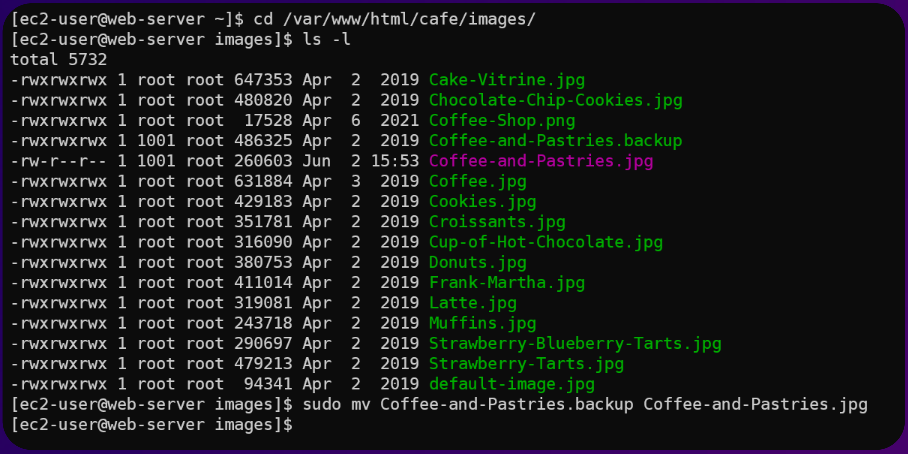
Security group rule ID sgr-08d532d6944f1294c	Type Info SSH ▼
Protocol Info TCP	Port range Info 22
Source type Info Custom ▼	Source Info 🔍 0.0.0.0/0 ✕

Task 5

Analyzing the hack further and improving security

Step 5: Fix the website

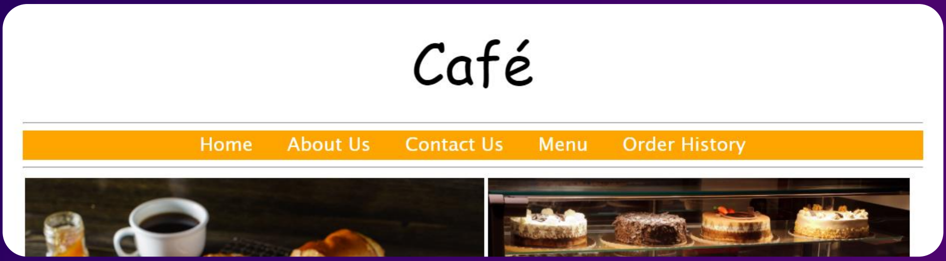
Run the following commands to restore the original graphic on the website.



```
[ec2-user@web-server ~]$ cd /var/www/html/cafe/images/
[ec2-user@web-server images]$ ls -l
total 5732
-rwxrwxrwx 1 root root 647353 Apr  2  2019 Cake-Vitrine.jpg
-rwxrwxrwx 1 root root 480820 Apr  2  2019 Chocolate-Chip-Cookies.jpg
-rwxrwxrwx 1 root root  17528 Apr  6  2021 Coffee-Shop.png
-rwxrwxrwx 1 1001 root 486325 Apr  2  2019 Coffee-and-Pastries.backup
-rw-r--r-- 1 1001 root 260603 Jun  2 15:53 Coffee-and-Pastries.jpg
-rwxrwxrwx 1 root root 631884 Apr  3  2019 Coffee.jpg
-rwxrwxrwx 1 root root 429183 Apr  2  2019 Cookies.jpg
-rwxrwxrwx 1 root root 351781 Apr  2  2019 Croissants.jpg
-rwxrwxrwx 1 root root 316090 Apr  2  2019 Cup-of-Hot-Chocolate.jpg
-rwxrwxrwx 1 root root 380753 Apr  2  2019 Donuts.jpg
-rwxrwxrwx 1 root root 411014 Apr  2  2019 Frank-Martha.jpg
-rwxrwxrwx 1 root root 319081 Apr  2  2019 Latte.jpg
-rwxrwxrwx 1 root root 243718 Apr  2  2019 Muffins.jpg
-rwxrwxrwx 1 root root 290697 Apr  2  2019 Strawberry-Blueberry-Tarts.jpg
-rwxrwxrwx 1 root root 479213 Apr  2  2019 Strawberry-Tarts.jpg
-rwxrwxrwx 1 root root  94341 Apr  2  2019 default-image.jpg
[ec2-user@web-server images]$ sudo mv Coffee-and-Pastries.backup Coffee-and-Pastries.jpg
[ec2-user@web-server images]$
```

Step 6: Test the fix

Reload the café website in the browser. That looks better.

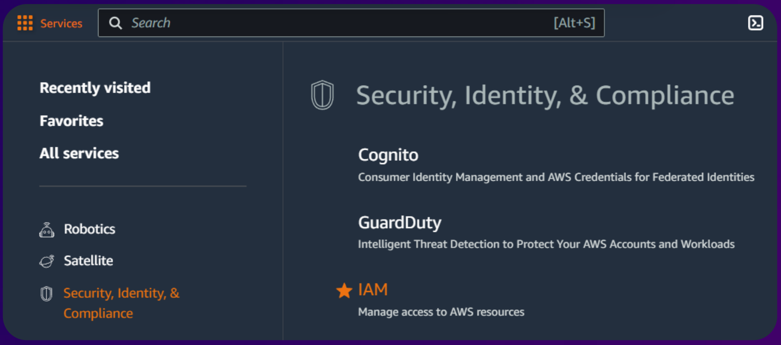


Task 5

Analyzing the hack further and improving security

Step 7: Access the IAM Management Console

In the AWS Management Console, select IAM.



Security, Identity, & Compliance

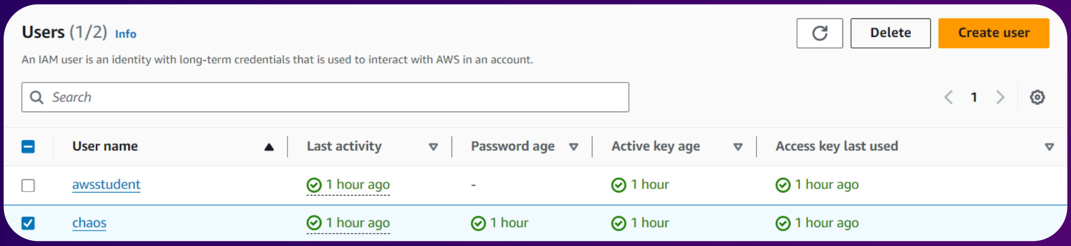
Cognito
Consumer Identity Management and AWS Credentials for Federated Identities

GuardDuty
Intelligent Threat Detection to Protect Your AWS Accounts and Workloads

★ IAM
Manage access to AWS resources

Step 8: Delete the chaos IAM user

Navigate to the **Users** section, select the **chaos** IAM user, and choose Delete.



Users (1/2)

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

☐	User name	Last activity	Password age	Active key age	Access key last used
☐	awsstudent	✓ 1 hour ago	-	✓ 1 hour	✓ 1 hour ago
☑	chaos	✓ 1 hour ago	✓ 1 hour	✓ 1 hour	✓ 1 hour ago

Conclusions

AWS CloudTrail

AWS CloudTrail provides comprehensive logging of AWS account activity, ensuring visibility and auditability for security and compliance purposes

CloudTrail Trails

Configuring CloudTrail trails enables continuous logging of AWS API calls, capturing detailed records of all actions taken on your AWS resources.

CloudTrail Logs

CloudTrail Logs capture detailed information about every action taken in your AWS environment, aiding in audit and forensic analysis.

The aws cloudtrail commands

The aws cloudtrail commands enable you to manage and interact with CloudTrail trails and logs programmatically, enhancing automation and operational efficiency.

Analyzing logs using Amazon Athena

Analyzing CloudTrail logs using Amazon Athena allows for efficient, SQL-based querying and detailed insights into AWS account activity and security events.

aws re/start

Cristhian Becerra

cristhian-becerra-espinoza

+51 951 634 354

cristhianbecerra99@gmail.com

Lima, Peru