TABLE OF CONTENTS

1. Abstract

The AWS - Intelligent Threat Detection and Real-Time Prevention project aims to enhance the security of cloud-based systems by to detect and prevent cyber threats in real-time. The project utilizes AWS services such as Amazon GuardDuty, and analyze data from various sources, including network traffic, logs, and user behavior, to identify potential threats. The project also employs automated responses and remediation actions to mitigate the impact of detected threats. Overall, the AWS - Intelligent Threat Detection and Real-Time Prevention project seeks to provide a comprehensive and proactive approach to security in the cloud.

2. Introduction and overview of project

POC on AWS – Intelligent Threat Detection and Real-Time Prevention.

GuardDuty is an AWS managed Threat detection service and customers speak a lot about securing their AWS infrastructure and its automated remediation. GuardDuty uses a combination of AWS CloudTrail, Amazon VPC Flow Logs and DNS Logs to detect malicious behavior and generate alerts if a possible compromise has been detected.

A GuardDuty finding represents a potential security issue detected within the network. GuardDuty generates a finding whenever it detects unexpected and potentially malicious activity in your AWS environment.

So using GuardDuty, we will deliberately create findings and can see all those events in the GuardDuty console followed by remediation using AWS CloudWatch events and Lambda functions.

All the findings that are generated here are considered safe in the sense that they don’t require penetration requests and none of these findings should result in AWS abuse content.

3. Introduction to AWS

Amazon Web Services (AWS) is a cloud computing platform offered by Amazon.com that provides a wide range of services to businesses and individuals. AWS offers a vast collection of cloud-based services that enable users to store, manage, process, and analyze data, as well as build and run applications and services in a secure, scalable, and cost-effective manner.

AWS offers various services in multiple categories such as compute, storage, database, analytics, security, machine learning, and more. Some of the most popular AWS services include Amazon EC2 (Elastic Compute Cloud) for virtual machine instances, Amazon S3 (Simple Storage Service) for object storage, Amazon RDS (Relational Database Service) for managed relational databases, and Amazon Lambda for serverless computing.

AWS offers a pay-as-you-go pricing model that allows users to pay only for the resources they use, without any upfront costs or long-term commitments. This makes it easy for businesses and individuals to scale up or down their computing resources as their needs change.

Overall, AWS has become a popular choice for businesses and individuals seeking flexible, scalable, and reliable cloud computing solutions. The platform is widely used by startups, large enterprises, government agencies, and non-profit organizations.

3.1 GuardDuty

Amazon GuardDuty is a threat detection service offered by AWS that provides continuous monitoring and threat detection across an organization's AWS accounts and workloads. GuardDuty uses machine learning and other detection techniques to analyze data from AWS CloudTrail event logs, Amazon VPC Flow Logs, and DNS logs, and then identifies and prioritizes potential security issues such as unauthorized access, compromised instances, and data exhilaration attempts.

GuardDuty continuously monitors AWS accounts and workloads and generates alerts in real-time whenever suspicious activity is detected. These alerts can be viewed and managed through the AWS Management Console or sent to other AWS services such as Amazon SNS or AWS Lambda for further analysis and automated response.

GuardDuty also integrates with other AWS security services, including AWS Security Hub, AWS Identity and Access Management (IAM), and AWS CloudFormation, allowing organizations to manage their security posture and response across multiple accounts and services.

Overall, GuardDuty helps organizations improve their security posture and minimize the risk of cyberattacks in their AWS environments by providing continuous monitoring and threat detection capabilities that are easy to use and cost-effective.

3.2 EC2 (Elastic Compute Cloud)

Amazon Elastic Compute Cloud (EC2) is a web service offered by AWS that provides resizable compute capacity in the cloud. EC2 allows users to create and manage virtual machine instances, known as EC2 instances, in the AWS cloud.

EC2 instances can be launched in various sizes and configurations, depending on the user's computing needs. Users can choose from a range of pre-configured Amazon Machine Images (AMIs) that include different operating systems, applications, and server software, or create their own custom AMIs.

EC2 instances can be launched in different regions and availability zones, providing users with high availability and scalability options. EC2 also supports a variety of storage options, including Amazon Elastic Block Store (EBS) for block-level storage and Amazon S3 for object storage.

Users can access their EC2 instances using a remote desktop client or a secure shell (SSH) client. EC2 also provides security features such as network security groups, access control lists, and encryption options for data in transit and at rest.

EC2 instances can be used for a wide range of use cases, such as running web applications, hosting databases, and running data processing and analysis workloads. EC2 also supports auto-scaling, which allows users to automatically adjust their compute capacity based on demand.

Overall, EC2 provides users with flexible, scalable, and cost-effective compute capacity in the cloud, enabling them to run a wide range of workloads and applications

3.3 SNS (Simple Notification service)

Amazon Simple Notification Service (SNS) is a fully managed pub/sub messaging service provided by AWS that enables users to send and receive messages from various sources and endpoints. SNS allows users to decouple the sending and receiving of messages, enabling multiple subscribers to receive the same message simultaneously.

SNS supports multiple messaging protocols, including HTTP, HTTPS, email, SMS, and mobile push notifications, allowing users to send messages to a wide range of endpoints. SNS also supports message filtering, allowing users to filter and route messages based on their content or attributes.

Users can create topics in SNS to which messages can be published. Subscribers can then subscribe to these topics to receive messages via the protocol of their choice. SNS also provides a fan-out feature that allows users to replicate messages to multiple endpoints, ensuring that all subscribers receive the same message.

SNS integrates with other AWS services such as AWS Lambda, AWS CloudFormation, and AWS CloudTrail, enabling users to automate workflows and trigger actions based on SNS messages.

Overall, SNS provides users with a flexible, scalable, and reliable messaging service that can be used to send and receive messages between applications, services, and endpoints. SNS is widely used for real-time notifications, event-driven architectures, and application integration scenarios

3.4 CloudWatch

Amazon CloudWatch is a monitoring and observability service provided by AWS that enables users to collect, analyze, and act on metrics, logs, and events from various AWS resources and applications. CloudWatch provides a unified view of an organization's infrastructure and applications, allowing users to troubleshoot issues and optimize performance.

CloudWatch offers several features that enable users to monitor and analyze their AWS resources, including EC2 instances, Lambda functions, and Elastic Load Balancers. CloudWatch collects and stores metrics and logs from these resources, providing real-time insights into resource utilization, performance, and errors.

CloudWatch also enables users to set alarms on metrics and logs, allowing them to proactively monitor their resources and applications for potential issues. When an alarm is triggered, CloudWatch can automatically perform actions such as sending notifications or executing AWS Lambda functions.

CloudWatch also provides a centralized log management solution that allows users to collect, store, and analyze logs from various sources, including AWS services, operating systems, and applications. CloudWatch Logs also enables users to search and filter logs, extract meaningful insights, and troubleshoot issues.

In addition, CloudWatch offers features such as dashboards, which allow users to create custom visualizations of metrics and logs, and CloudWatch Events, which allows users to respond to events and changes within their AWS resources and applications.

Overall, CloudWatch provides users with a powerful monitoring and observability solution that enables them to optimize the performance and availability of their AWS resources and applications.

3.5. IAM (Identity and Access Management)

AWS Identity and Access Management (IAM) is a web service provided by AWS that enables users to manage access to AWS resources and services securely. IAM allows users to create and manage users, groups, and roles that have different levels of access to AWS resources.

With IAM, users can control access to AWS resources by creating IAM policies that define permissions for different users, groups, or roles. IAM policies can specify which AWS resources a user can access, what actions they can perform on those resources, and under what conditions.

IAM also provides several features to enhance security and manage access to AWS resources, such as multi-factor authentication (MFA) for users, access keys for programmatic access, and temporary security credentials for accessing AWS resources.

IAM enables users to create roles, which are a set of permissions that can be assumed by AWS services and applications. Roles can be used to grant permissions to AWS services such as EC2 instances and Lambda functions, or to enable cross-account access to AWS resources.

IAM also provides auditing and compliance features, such as AWS CloudTrail, which logs all API calls made to AWS services, enabling users to track changes and monitor activity on their AWS accounts.

Overall, IAM provides users with a centralized and secure way to manage access to AWS resources, enabling them to control access to their resources and ensure compliance with security best practices.

3.6 AWS Lambda

AWS Lambda is a serverless computing service provided by AWS that enables users to run code without having to manage or provision servers. With Lambda, users can upload their code and AWS Lambda takes care of everything required to run and scale the code with high availability.

Lambda supports a variety of programming languages, including Python, Java, Node.js, and C#. Users can also use Lambda to run custom code or scripts, and to process events or data from other AWS services such as S3, SNS, and DynamoDB.

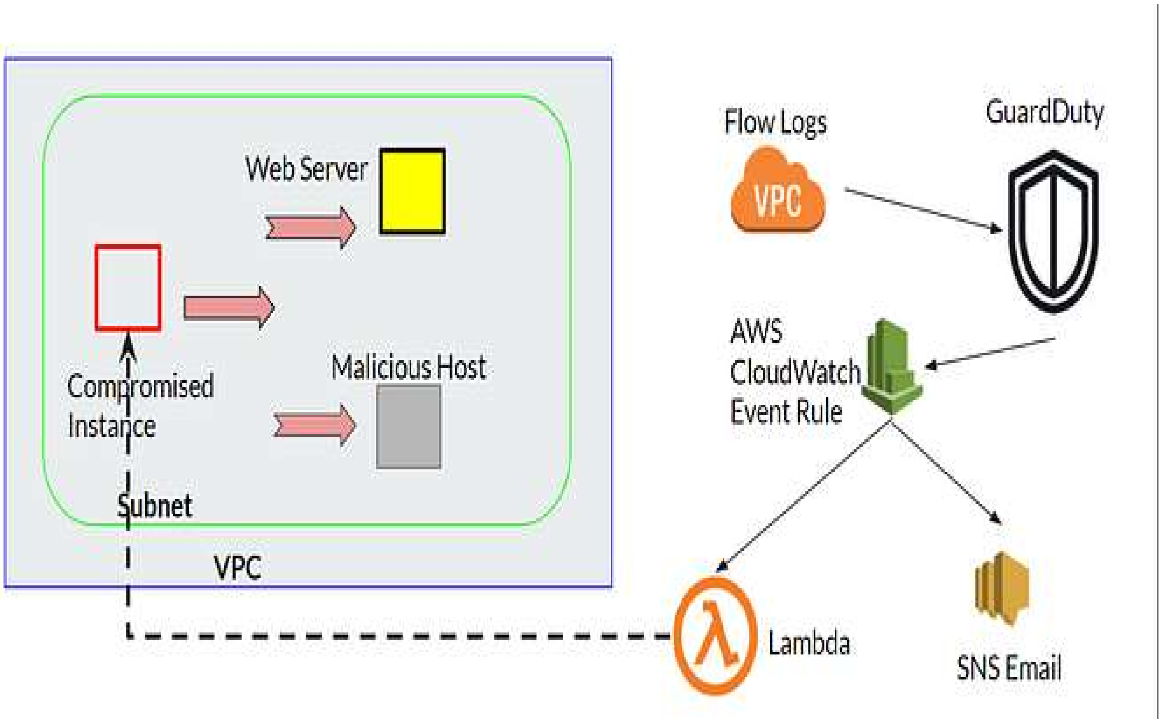
Lambda functions can be triggered by a variety of events, including HTTP requests, S3 object uploads, and CloudWatch events. Lambda can also be integrated with other AWS services such as API Gateway, Step Functions, and SNS, enabling users to build complex serverless applications.

Lambda functions are charged based on the number of requests, the duration of each request, and the amount of memory allocated to the function. Lambda scales automatically to handle incoming requests, and users only pay for the compute time they consume.

Lambda provides users with several benefits, including the ability to build highly scalable and fault-tolerant applications without worrying about server management, reduced time-to-market for new applications, and cost savings by paying only for the compute time used.

Overall, AWS Lambda provides users with a flexible, scalable, and cost-effective way to build and run applications and services, and is a key component of AWS's serverless computing portfolio.

4. Data Flow Diagram



5. AWS Project Setup

Let’s get started. Enable GuardDuty to capture findings. GuardDuty uses VPC flow logs, CloudTrail logs and DNS logs to detect malicious behavior and generate alerts on the GuardDuty console if a possible compromise has been detected.

Now we will create three EC2 instances in VPC’s public subnet. We are saying the first EC2 as a compromised instance because it's doing two things. One its doing a port scan to an internal server, two it is constantly pinging host which is considered to be malicious.
Now we are calling the second EC2 instance as malicious because the Elastic IP attached to the instance is in the Threat list of the GuardDuty. GuardDuty generates findings for IP addresses that are included in threat lists.
The third instance is an internal server that has few ports exposed as an API endpoint for other application servers. So we need a Security group which has inbound rules for few ports and that will be attached to the internal server.

Now we need to create a CloudWatch event rule that will collect logs from the event source and then it will forward it to the target service which will be used for alert notification and remediation purposes. Target services according to the diagram will be SNS(Simple Notification Service) and AWS Lambda.

To create a CloudWatch event rule, we need to create a target service first. For SNS, create an SNS Topic “guardduty-security-topic”, followed by creating a subscription. Select Protocol as “Email” and enter an endpoint email address, all alert notification will be mail to the added email address in the subscriptions.

Go to CloudWatch->Events->Rules and create a new Rule “guardduty-findings-rule”. Select Service Name as “GuardDuty” and Event Type as “GuardDuty Findings”. Now we have to select targets, so Targets are used to invoke when an event matches or triggers. The first target is SNS which we have already created in the previous step, so select SNS Topic and Topic Name i.e. “guardduty-security-topic” thus click on add target.

So until now, we have completed out one pipeline i.e. from VPC Flow logs to SNS. As of now if any findings are detected by GuardDuty, subscribers of SNS Topic “guardduty-security-topic” will receive an email notification. We are now left with the remediation part, so we will use the boto3 framework in AWS Lambda functions.

Lambda:
The idea behind the remediation is to change the security group of the compromised instance. There are two things which are needed for Lambda:

Security Group “compromised-ec2-sg” which has no Inbound and outbound rules. Simply this means the affected instance will be isolated from the environment.

Create an IAM role “lambda-guardduty-role” for Lambda which has sufficient permission for EC2 Security group changes. For now, let's give AmazonEC2FullAccess, AWSLambdaBasicExecutionRole and AmazonSNSFullAccess.

Create AWS Lambda function “guardduty-pipeline-lambda” with the runtime as Python3.8 and in the permission section select “use an existing role”, specify the above-created role “lambda-guardduty-role”. On Lambda, add trigger as CloudWatch Events and select the rule which we have created “guardduty-findings-rule”. Now the Lambda trigger is successfully configured. You can verify the trigger on CloudWatch Events Rules “guardduty-findings-rule” target lists. It will now have two entries i.e. for Lambda function and SNS topic.

Now we will write a snippet for which we will be using boto3 for accessing AWS resources. The code below, snippet represents if finding Recon:EC2/Portscan is detected(reproduction discussed in Attack section) then the victim machine’s security group will be changed to “compromised-ec2-sg” and in our case, it’s the first EC2 i.e. compromised instance. We are isolating the instance from every other resource. This is just to make sure there are no backdoor connections that exist on the compromised instance.

Code of Lambda Function

```
import boto3
ec2 = boto3.resource ('ec2')
isolated_sg = 'sg-02fa6e457e0965774' # ID of Security group “compromised-ec2-sg”
def lambda_handler(event, context):
#Method to change the security group of the affected EC2 instance
 print(f"PFB event\n{event}")
 finding_type = event['detail']['type']
 instance_id = event['detail']['resource']['instanceDetails']['instanceId']

 # logging the finding and instance details
 print(f"Finding type: {finding_type}")
 print(f"Instance ID: {instance_id}")

 if finding_type == 'Recon:EC2/Portscan':
  victim_ec2 = ec2.Instance(instance_id)
  victim_ec2.modify_attribute(Groups=[isolated_sg])
  print("successfull")
 # If any suspicious activity is detected by GuardDuty, then the affected ec2 will be moved to this security group

 ## SNS code below - sending mail to the stakeholders
 subject = event["detail"]["title"]
 body = event["detail"]["description"]
 body += " on " + event["detail"]["createdAt"]

 sns_arn = "arn:aws:sns:ap-south-1:008306497099:Cdac_Pro"

 client = boto3.client('sns')
 response = client.publish(
    TopicArn=sns_arn,
    Message=body,
    Subject=subject
```

ATTACK AND REMEDIATION IN WORKING:

The scenario here is the compromised instance is doing port scanning to the internal server and pinging to the Malicious host. Login to the compromised instance and use Nmap for port scanning.

```
ec2-compromised$ nmap -Pn <IP-internal-server>
Nmap scan report for ec2-compromised
Host is up (0.00050s latency).
Not shown: 995 filtered ports
PORT     STATE  SERVICE
22/tcp   open   ssh
80/tcp   closed http
445/tcp  closed microsoft-ds
2049/tcp closed nfs
5432/tcp closed postgresqlec2-compromised$ ping <IP-malicious-instance>
```

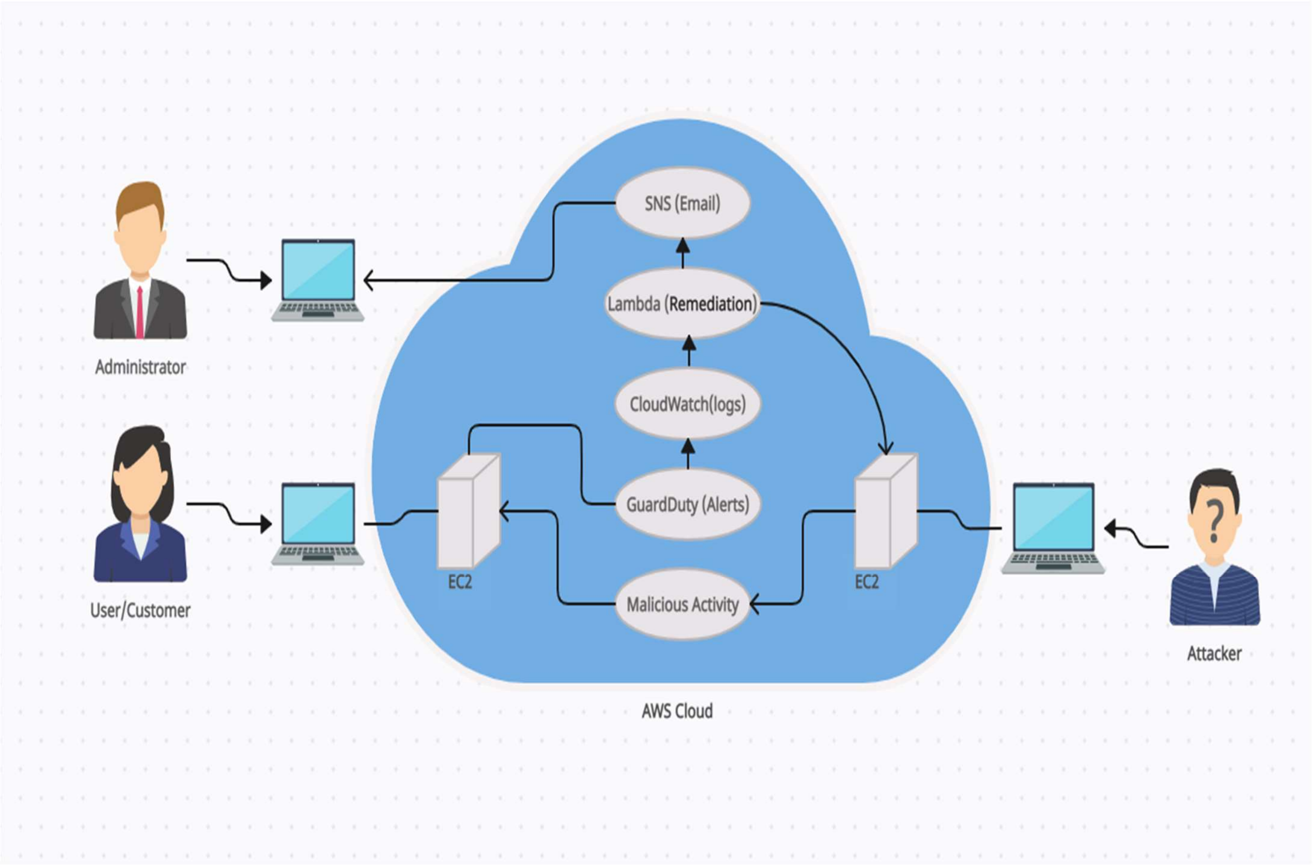
All these network activities are stored in VPC flow logs which GuardDuty takes as input for threat analysis. Now doing port scanning and pinging to malicious host from compromised instance that will give us Recon:EC2/Portscan and unauthorized EC2 access malicious IP caller finding on the GuardDuty console. Now it takes a few minutes to display the finding on the GuardDuty console. Findings are automatically sent to CloudWatch Events and new findings are exported within 5 minutes.

After the required amount of time, the CloudWatch event rule will receive those finding logs and if the finding events are matched it will invoke target service. Here we have SNS Topic and Lambda function as out target service.
SNS Topic subscribers will receive an email notification containing finding details.

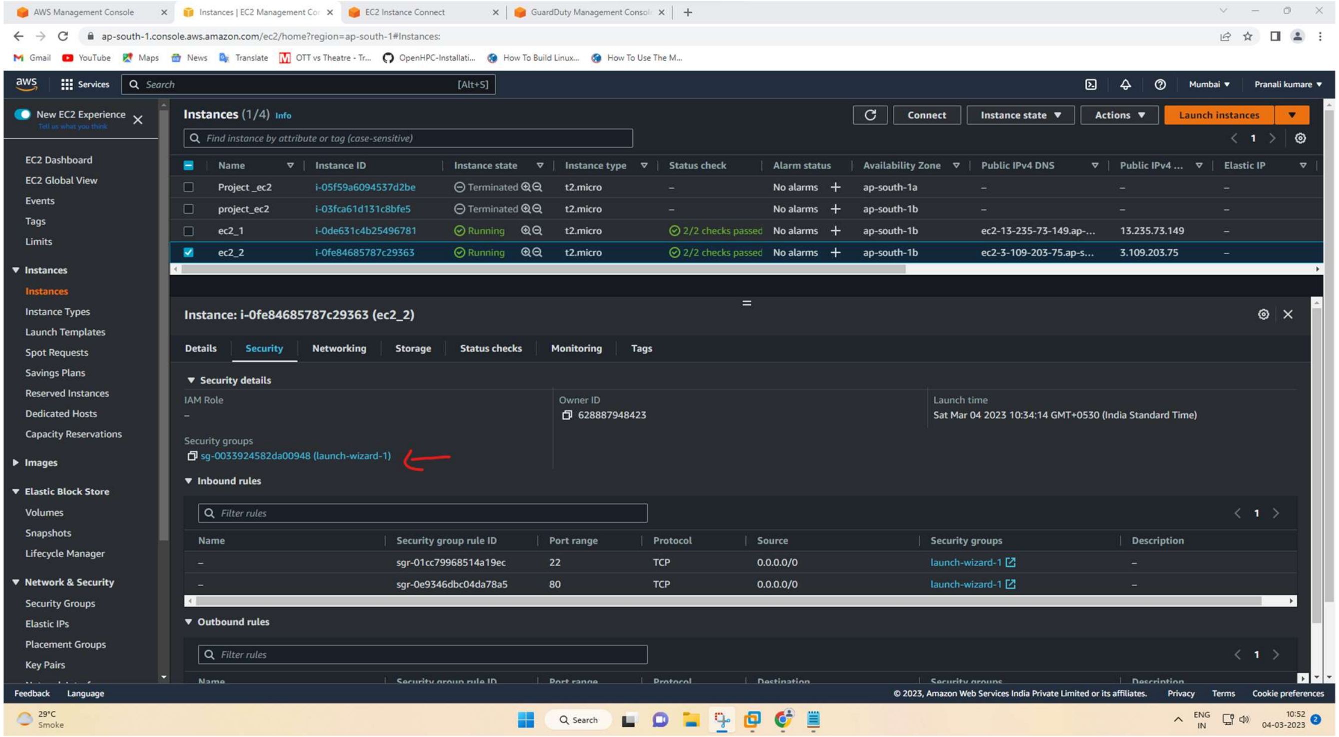
Now when Lambda gets triggered it will change the security group of the compromised instance. Compromised instances details are available in the event logs and the Lambda function is extracting the affected instance details from the event variable. The Security Group will isolate the compromised instance from the whole infrastructure.

In this way we can identify threats using GuardDuty on our AWS infrastructure and have an automated prevention mechanism using AWS Lambda functions.

6. Use Case Diagram



7. Result



Instances (1/4) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
Project _ec2	i-05f59a6094537d2be	Terminated	t2.micro	–	No alarms	ap-south-1a	–	–	–
project_ec2	i-03fca61d131c8bfe5	Terminated	t2.micro	–	No alarms	ap-south-1b	–	–	–
ec2_1	i-0de631c4b25496781	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1b	ec2-13-235-73-149.ap-...	13.235.73.149	–
ec2_2	i-0fe84685787c29363	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1b	ec2-3-109-203-75.ap-s...	3.109.203.75	–

Instance: i-0fe84685787c29363 (ec2_2)

Security details

IAM Role: –

Owner ID: 628887948423

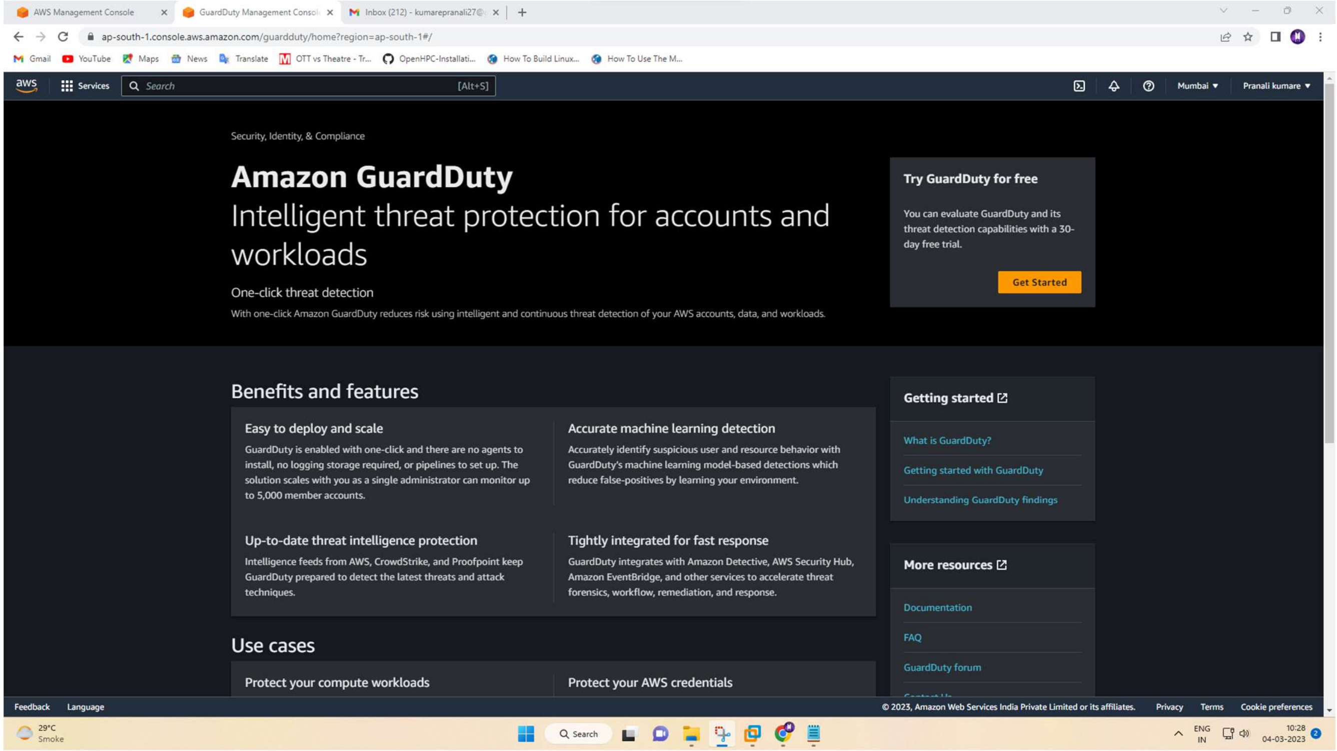
Launch time: Sat Mar 04 2023 10:34:14 GMT+0530 (India Standard Time)

Security groups: sg-0033924582da00948 (launch-wizard-1)

Inbound rules

Name	Security group rule ID	Port range	Protocol	Source	Security groups	Description
–	sgr-01cc79968514a19ec	22	TCP	0.0.0.0/0	launch-wizard-1	–
–	sgr-0e9346dbc04da78a5	80	TCP	0.0.0.0/0	launch-wizard-1	–

Outbound rules



Security, Identity, & Compliance

Amazon GuardDuty

Intelligent threat protection for accounts and workloads

One-click threat detection

With one-click Amazon GuardDuty reduces risk using intelligent and continuous threat detection of your AWS accounts, data, and workloads.

Try GuardDuty for free

You can evaluate GuardDuty and its threat detection capabilities with a 30-day free trial.

Get Started

Benefits and features

Easy to deploy and scale

GuardDuty is enabled with one-click and there are no agents to install, no logging storage required, or pipelines to set up. The solution scales with you as a single administrator can monitor up to 5,000 member accounts.

Accurate machine learning detection

Accurately identify suspicious user and resource behavior with GuardDuty's machine learning model-based detections which reduce false-positives by learning your environment.

Up-to-date threat intelligence protection

Intelligence feeds from AWS, CrowdStrike, and Proofpoint keep GuardDuty prepared to detect the latest threats and attack techniques.

Tightly integrated for fast response

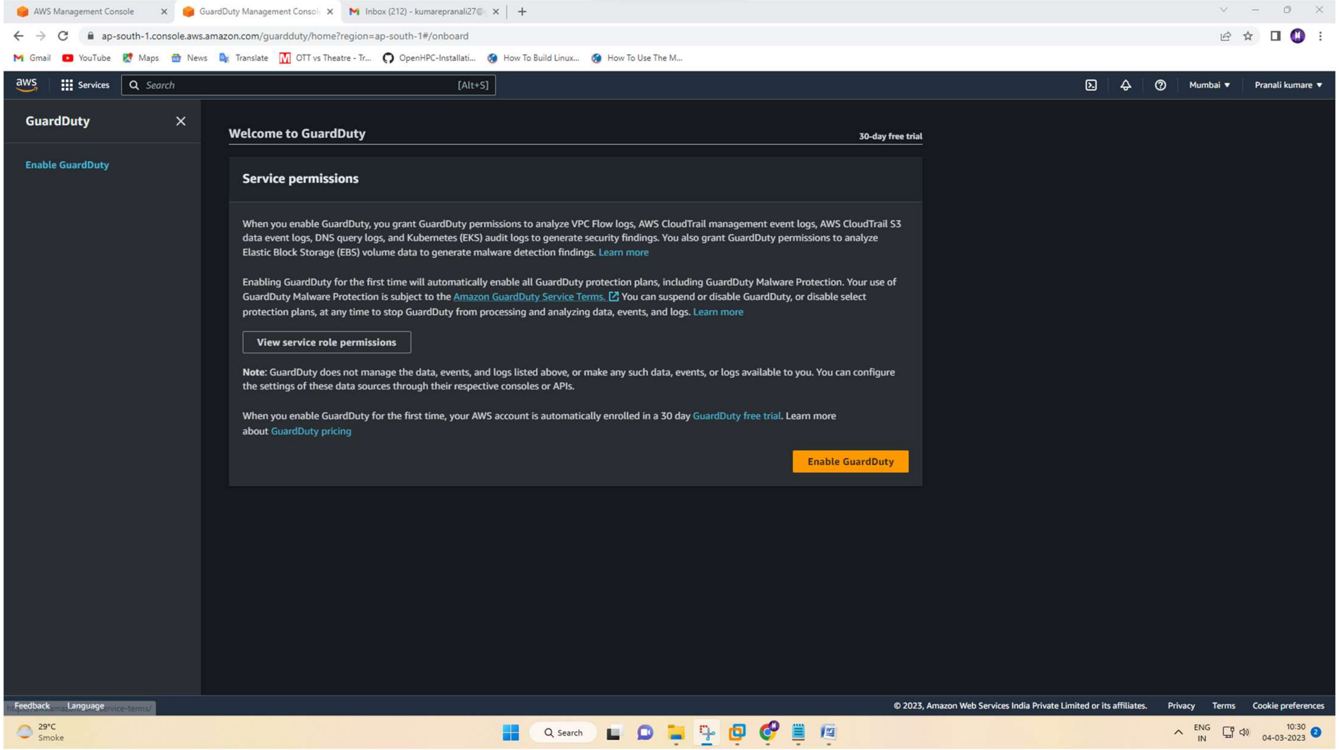
GuardDuty integrates with Amazon Detective, AWS Security Hub, Amazon EventBridge, and other services to accelerate threat forensics, workflow, remediation, and response.

Use cases

Protect your compute workloads

Protect your AWS credentials

AWS Management Console
GuardDuty Management Console
Inbox (212) - kumarepranali27@
ap-south-1.console.aws.amazon.com/guardduty/home?region=ap-south-1#/
Services
Search
[Alt+S]
Mumbai
Pranali kumare
Getting started
What is GuardDuty?
Getting started with GuardDuty
Understanding GuardDuty findings
More resources
Documentation
FAQ
GuardDuty forum
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
29°C
Smoke
10:28
04-03-2023



Welcome to GuardDuty

30-day free trial

Service permissions

When you enable GuardDuty, you grant GuardDuty permissions to analyze VPC Flow logs, AWS CloudTrail management event logs, AWS CloudTrail S3 data event logs, DNS query logs, and Kubernetes (EKS) audit logs to generate security findings. You also grant GuardDuty permissions to analyze Elastic Block Storage (EBS) volume data to generate malware detection findings. Learn more

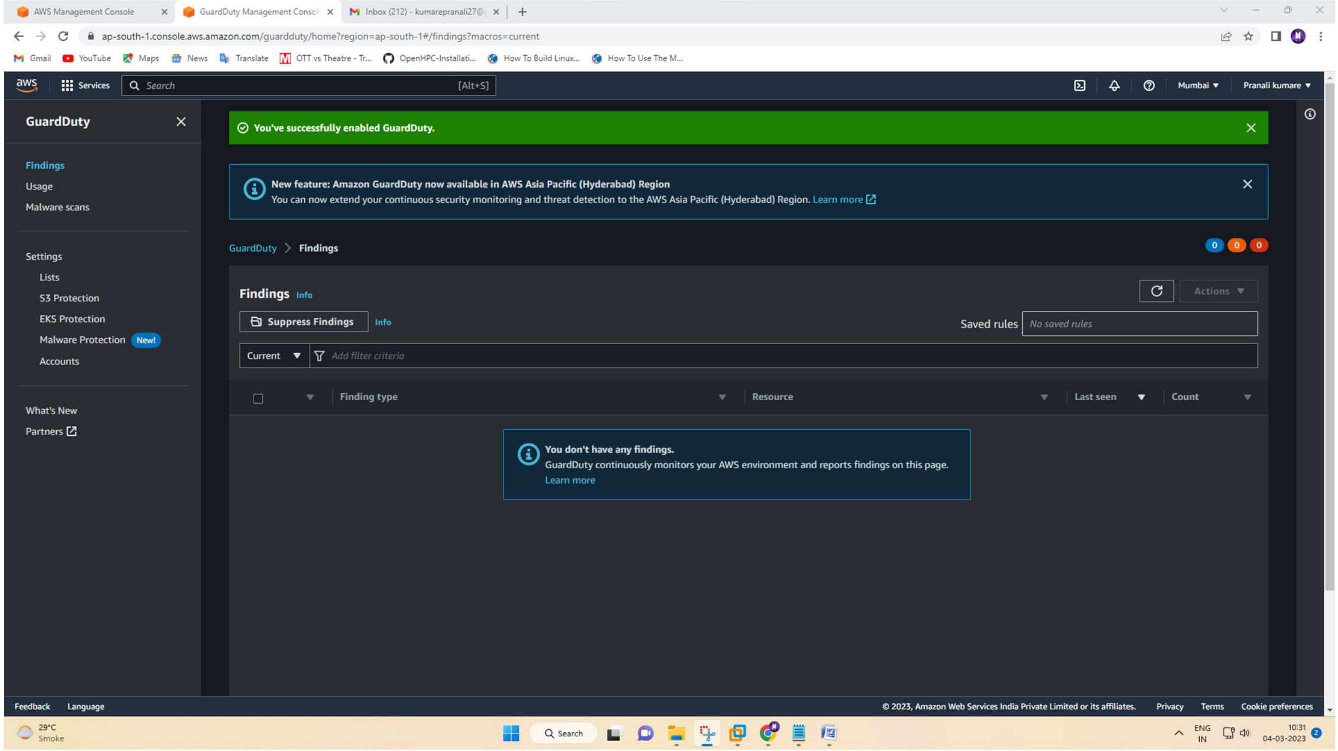
Enabling GuardDuty for the first time will automatically enable all GuardDuty protection plans, including GuardDuty Malware Protection. Your use of GuardDuty Malware Protection is subject to the Amazon GuardDuty Service Terms. You can suspend or disable GuardDuty, or disable select protection plans, at any time to stop GuardDuty from processing and analyzing data, events, and logs. Learn more

View service role permissions

Note: GuardDuty does not manage the data, events, and logs listed above, or make any such data, events, or logs available to you. You can configure the settings of these data sources through their respective consoles or APIs.

When you enable GuardDuty for the first time, your AWS account is automatically enrolled in a 30 day GuardDuty free trial. Learn more about GuardDuty pricing

Enable GuardDuty

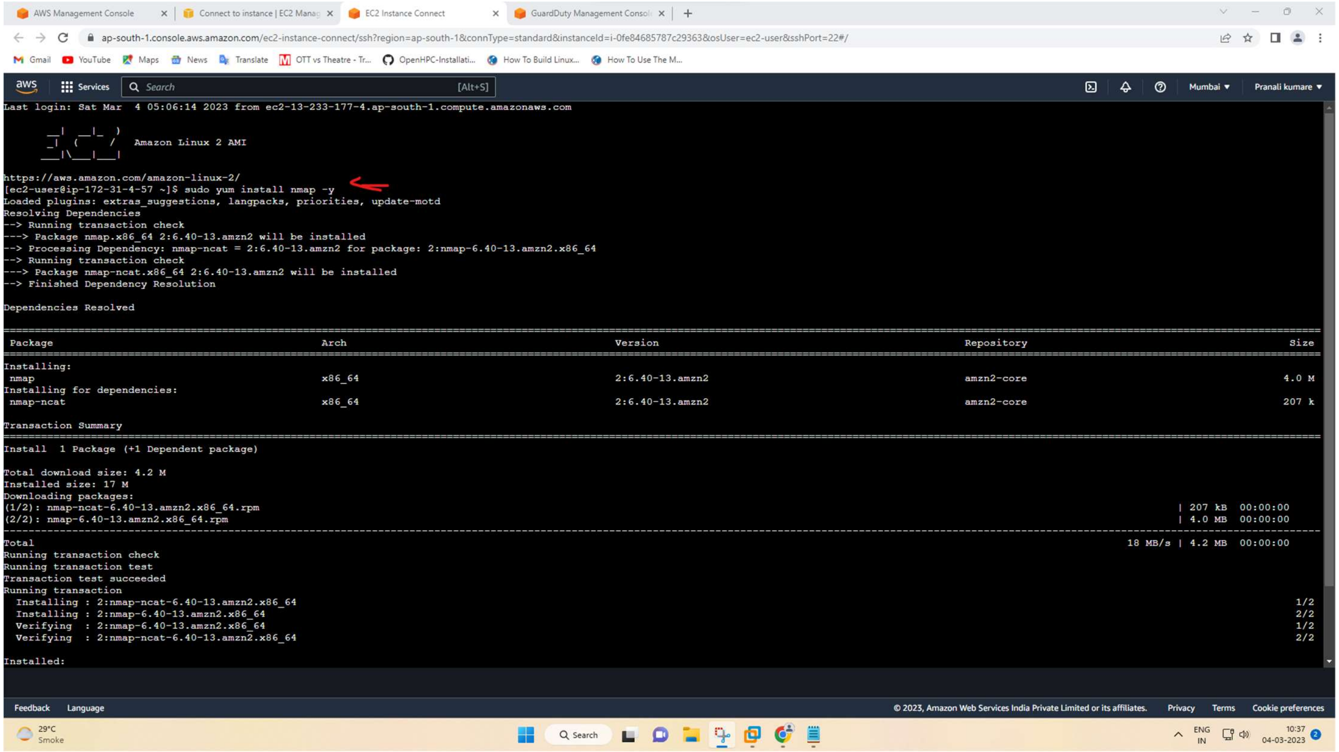


You've successfully enabled GuardDuty.

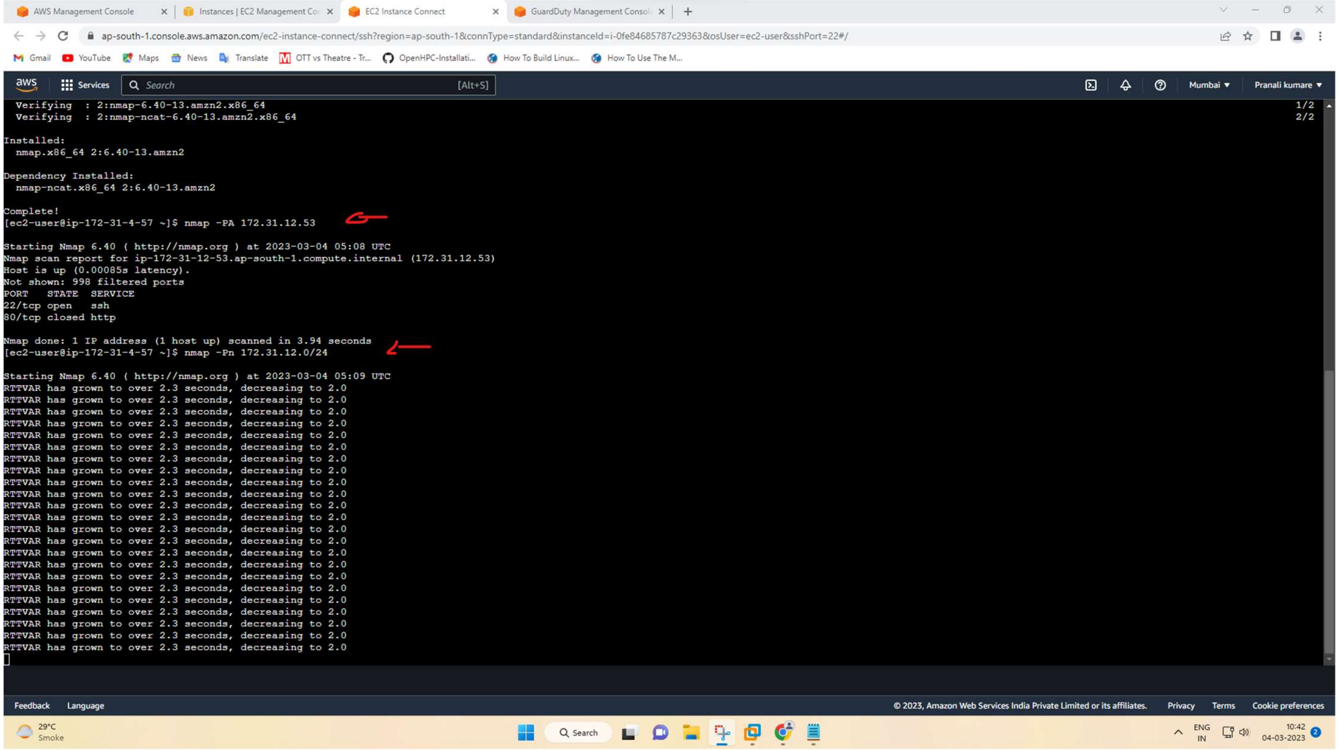
New feature: Amazon GuardDuty now available in AWS Asia Pacific (Hyderabad) Region
You can now extend your continuous security monitoring and threat detection to the AWS Asia Pacific (Hyderabad) Region. Learn more

Findings Info

You don't have any findings.
GuardDuty continuously monitors your AWS environment and reports findings on this page.
Learn more

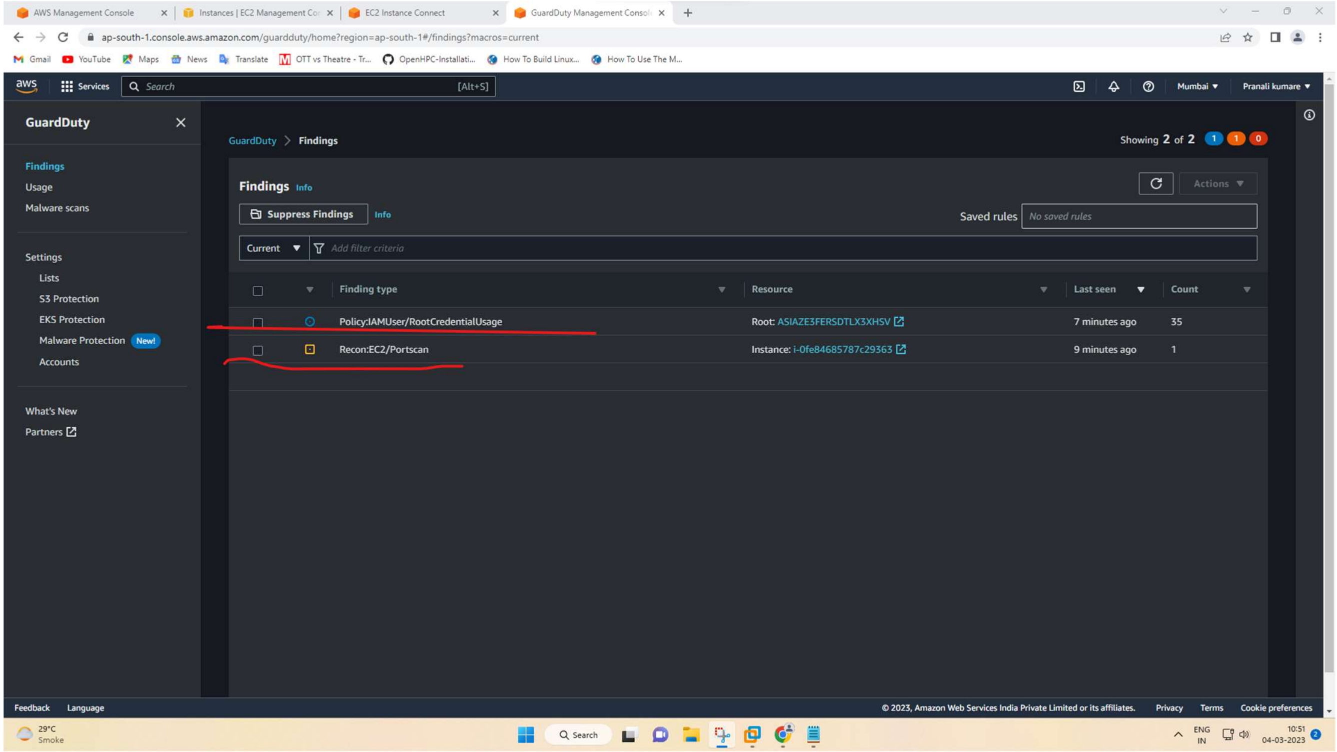

AWS Management Console
Connect to instance | EC2 Manag
EC2 Instance Connect
GuardDuty Management Consol
ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=ap-south-1&connType=standard&instanceId=i-0fe84685787c29363&osUser=ec2-user&sshPort=22#/
Gmail
YouTube
Maps
News
Translate
OTT vs Theatre - Tr...
OpenHPC-Installati...
How To Build Linux...
How To Use The M...
aws
Services
Search
[Alt+S]
Mumbai
Pranali kumare
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
29°C
Smoke
Search
ENG
IN
10:37
04-03-2023

```
Last login: Sat Mar  4 05:06:14 2023 from ec2-13-233-177-4.ap-south-1.compute.amazonaws.com

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-4-57 ~]$ sudo yum install nmap -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
---> Package nmap.x86_64 2:6.40-13.amzn2 will be installed
--> Processing Dependency: nmap-ncat = 2:6.40-13.amzn2 for package: 2:nmap-6.40-13.amzn2.x86_64
--> Running transaction check
---> Package nmap-ncat.x86_64 2:6.40-13.amzn2 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

==============================================================================================================================================
 Package                          Arch                        Version                            Repository                             Size
==============================================================================================================================================
Installing:
 nmap                             x86_64                      2:6.40-13.amzn2                    amzn2-core                            4.0 M
Installing for dependencies:
 nmap-ncat                        x86_64                      2:6.40-13.amzn2                    amzn2-core                            207 k

Transaction Summary
==============================================================================================================================================
Install  1 Package (+1 Dependent package)

Total download size: 4.2 M
Installed size: 17 M
Downloading packages:
(1/2): nmap-ncat-6.40-13.amzn2.x86_64.rpm                                                                           | 207 kB  00:00:00
(2/2): nmap-6.40-13.amzn2.x86_64.rpm                                                                                | 4.0 MB  00:00:00
----------------------------------------------------------------------------------------------------------------------------------------------
Total                                                                                                       18 MB/s | 4.2 MB  00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : 2:nmap-ncat-6.40-13.amzn2.x86_64                                                                                          1/2
  Installing : 2:nmap-6.40-13.amzn2.x86_64                                                                                               2/2
  Verifying  : 2:nmap-6.40-13.amzn2.x86_64                                                                                               1/2
  Verifying  : 2:nmap-ncat-6.40-13.amzn2.x86_64                                                                                          2/2

Installed:
```



```
  Verifying  : 2:nmap-6.40-13.amzn2.x86_64                                                                                               1/2
  Verifying  : 2:nmap-ncat-6.40-13.amzn2.x86_64                                                                                          2/2

Installed:
  nmap.x86_64 2:6.40-13.amzn2

Dependency Installed:
  nmap-ncat.x86_64 2:6.40-13.amzn2

Complete!
[ec2-user@ip-172-31-4-57 ~]$ nmap -PA 172.31.12.53

Starting Nmap 6.40 ( http://nmap.org ) at 2023-03-04 05:08 UTC
Nmap scan report for ip-172-31-12-53.ap-south-1.compute.internal (172.31.12.53)
Host is up (0.00085s latency).
Not shown: 998 filtered ports
PORT   STATE  SERVICE
22/tcp open   ssh
80/tcp closed http

Nmap done: 1 IP address (1 host up) scanned in 3.94 seconds
[ec2-user@ip-172-31-4-57 ~]$ nmap -Pn 172.31.12.0/24

Starting Nmap 6.40 ( http://nmap.org ) at 2023-03-04 05:09 UTC
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
```

Threat Detection-



ap-south-1.console.aws.amazon.com/guardduty/home?region=ap-south-1#/findings?macros=current

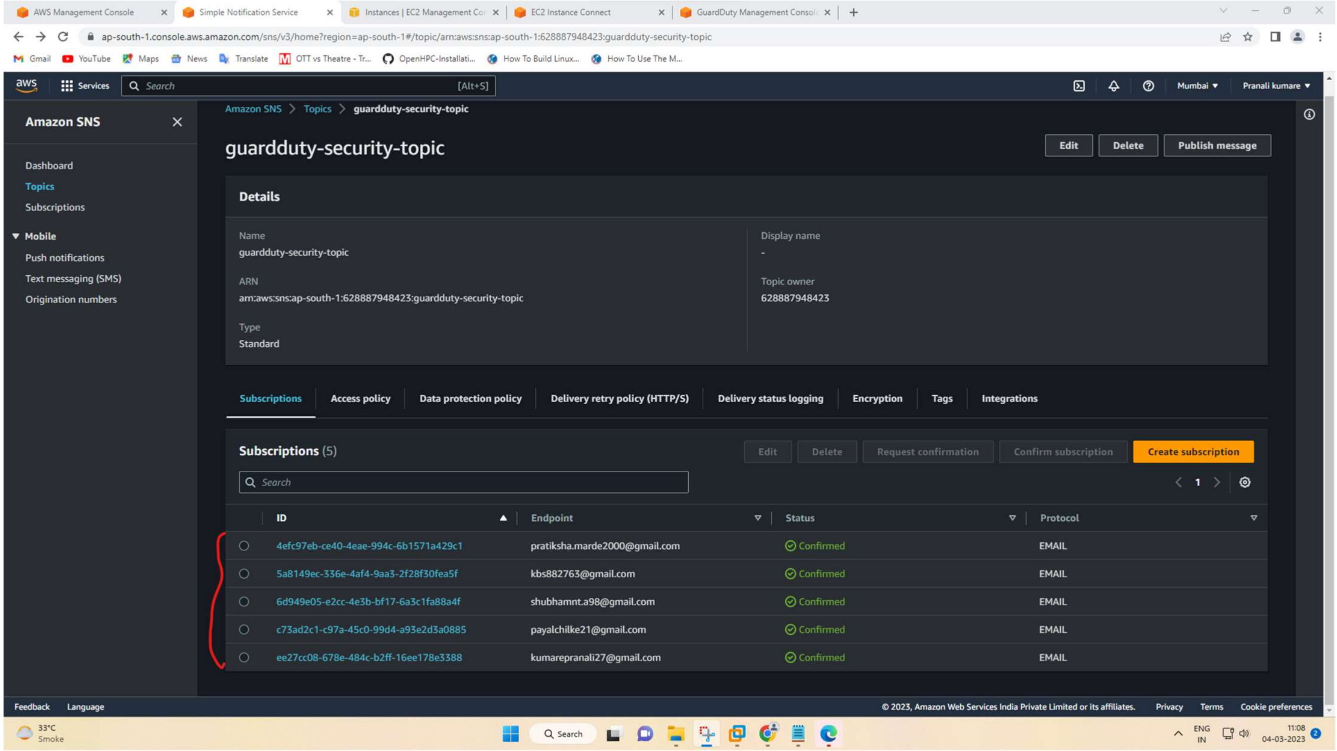
Findings Info

Showing 2 of 2 1 1 0

Saved rules: No saved rules

	Finding type	Resource	Last seen	Count
□	Policy:IAMUser/RootCredentialUsage	Root: ASIAZE3FERSDTLX3XHSV	7 minutes ago	35
□	Recon:EC2/Portscan	Instance: i-0fe84685787c29363	9 minutes ago	1

© 2023, Amazon Web Services India Private Limited or its affiliates.



ap-south-1.console.aws.amazon.com/sns/v3/home?region=ap-south-1#/topic/arn:aws:sns:ap-south-1:628887948423:guardduty-security-topic

Amazon SNS > Topics > guardduty-security-topic

guardduty-security-topic

Details

Name
guardduty-security-topic

Display name
-

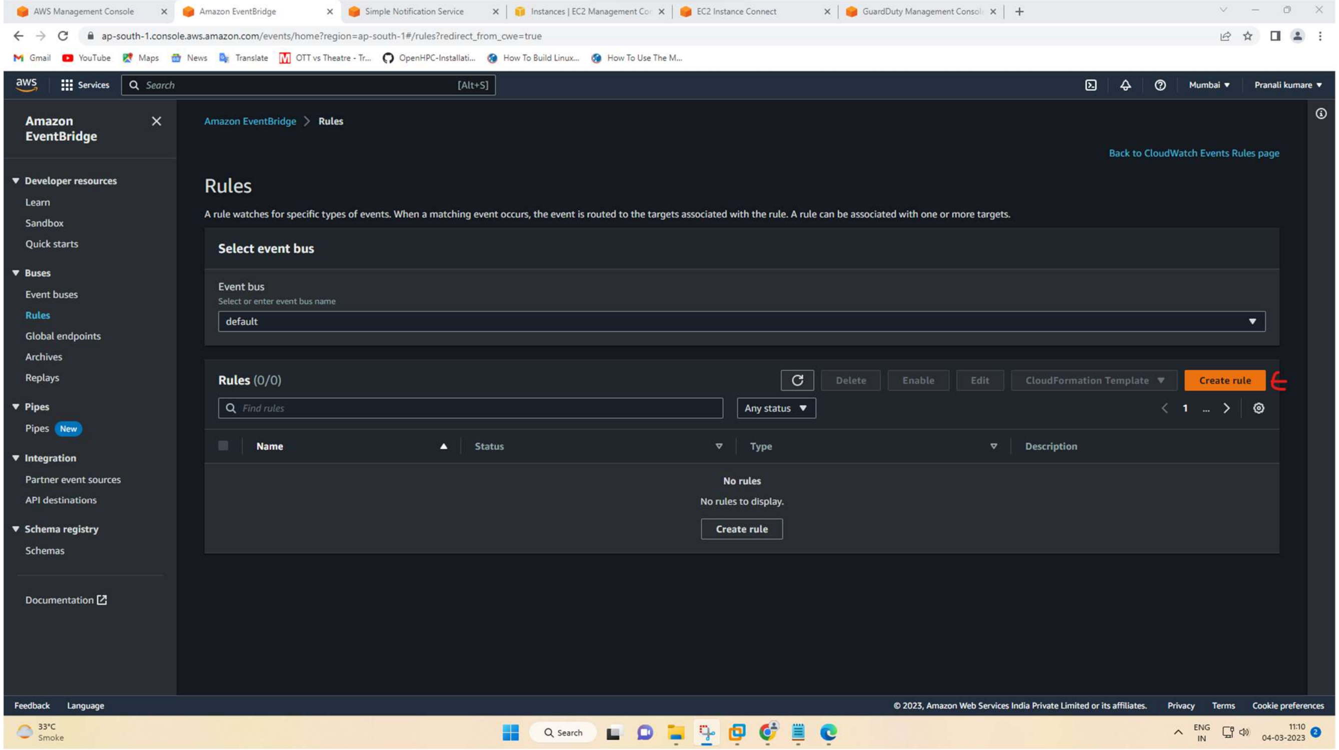
ARN
arn:aws:sns:ap-south-1:628887948423:guardduty-security-topic

Topic owner
628887948423

Type
Standard

Subscriptions (5)

ID	Endpoint	Status	Protocol
4efc97eb-ce40-4eae-994c-6b1571a429c1	pratiksha.marde2000@gmail.com	Confirmed	EMAIL
5a8149ec-336e-4af4-9aa3-2f28f30fea5f	kbs882763@gmail.com	Confirmed	EMAIL
6d949e05-e2cc-4e3b-bf17-6a3c1fa88a4f	shubhamnt.a98@gmail.com	Confirmed	EMAIL
c73ad2c1-c97a-45c0-99d4-a93e2d3a0885	payalchilke21@gmail.com	Confirmed	EMAIL
ee27cc08-678e-484c-b2ff-16ee178e3388	kumarepranali27@gmail.com	Confirmed	EMAIL



ap-south-1.console.aws.amazon.com/events/home?region=ap-south-1#/rules?redirect_from_cwe=true

Amazon EventBridge > Rules

Rules

A rule watches for specific types of events. When a matching event occurs, the event is routed to the targets associated with the rule. A rule can be associated with one or more targets.

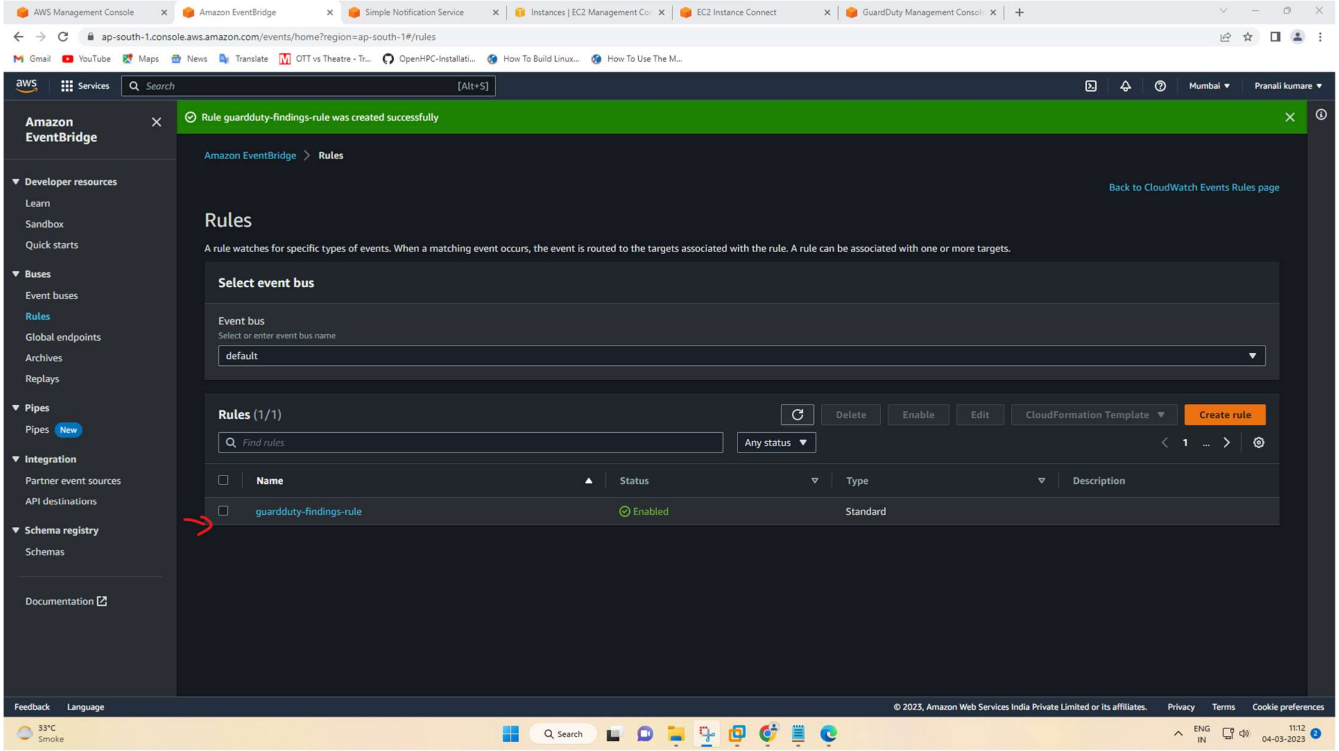
Select event bus

Event bus
Select or enter event bus name
default

Rules (0/0)

Name	Status	Type	Description

No rules
No rules to display.



Rule guardduty-findings-rule was created successfully

Amazon EventBridge > Rules

Back to CloudWatch Events Rules page

Rules

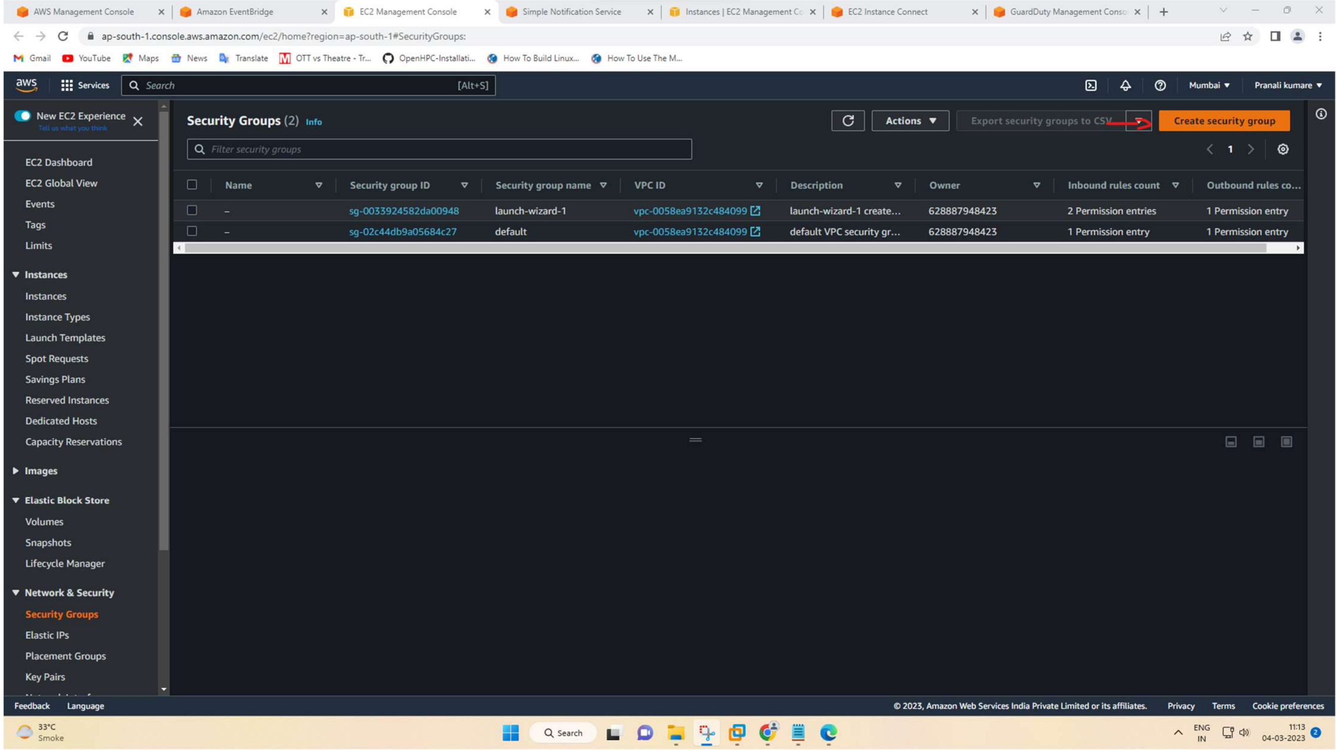
A rule watches for specific types of events. When a matching event occurs, the event is routed to the targets associated with the rule. A rule can be associated with one or more targets.

Select event bus

Event bus
Select or enter event bus name
default

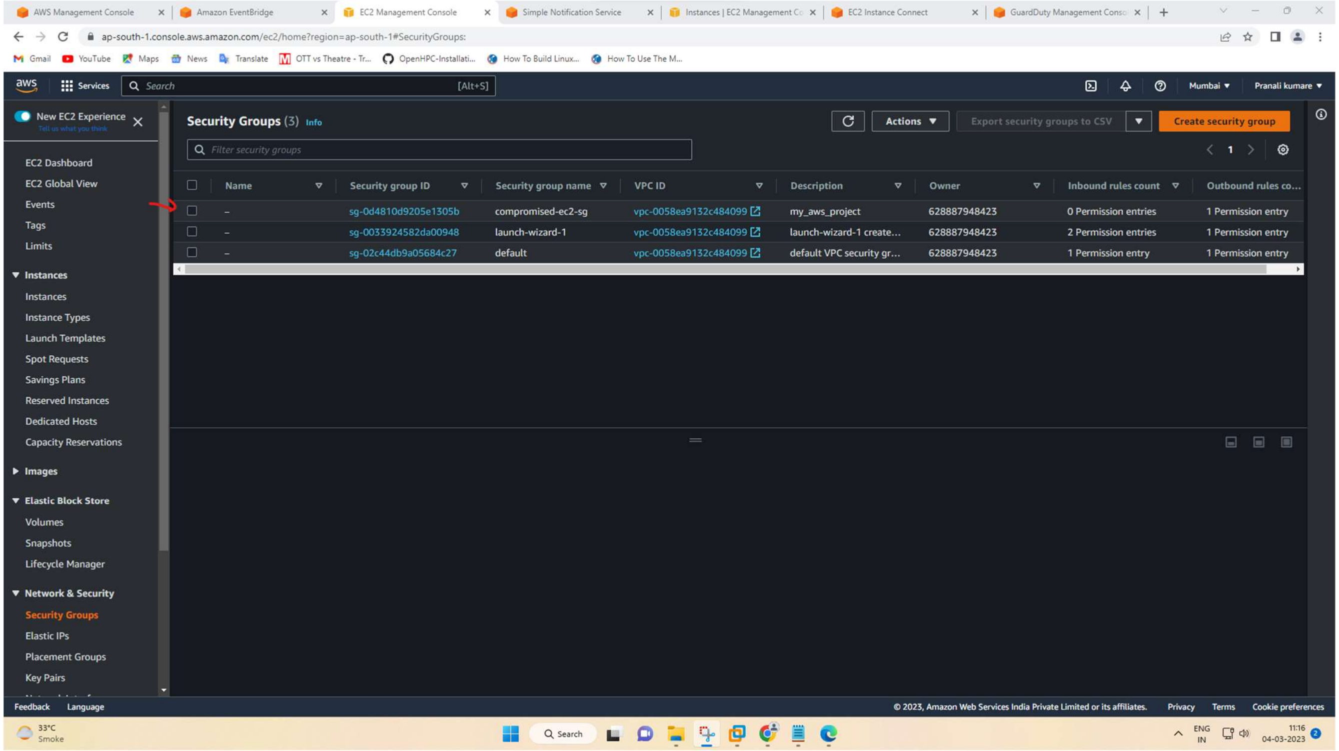
Rules (1/1)

Name	Status	Type	Description
guardduty-findings-rule	Enabled	Standard	



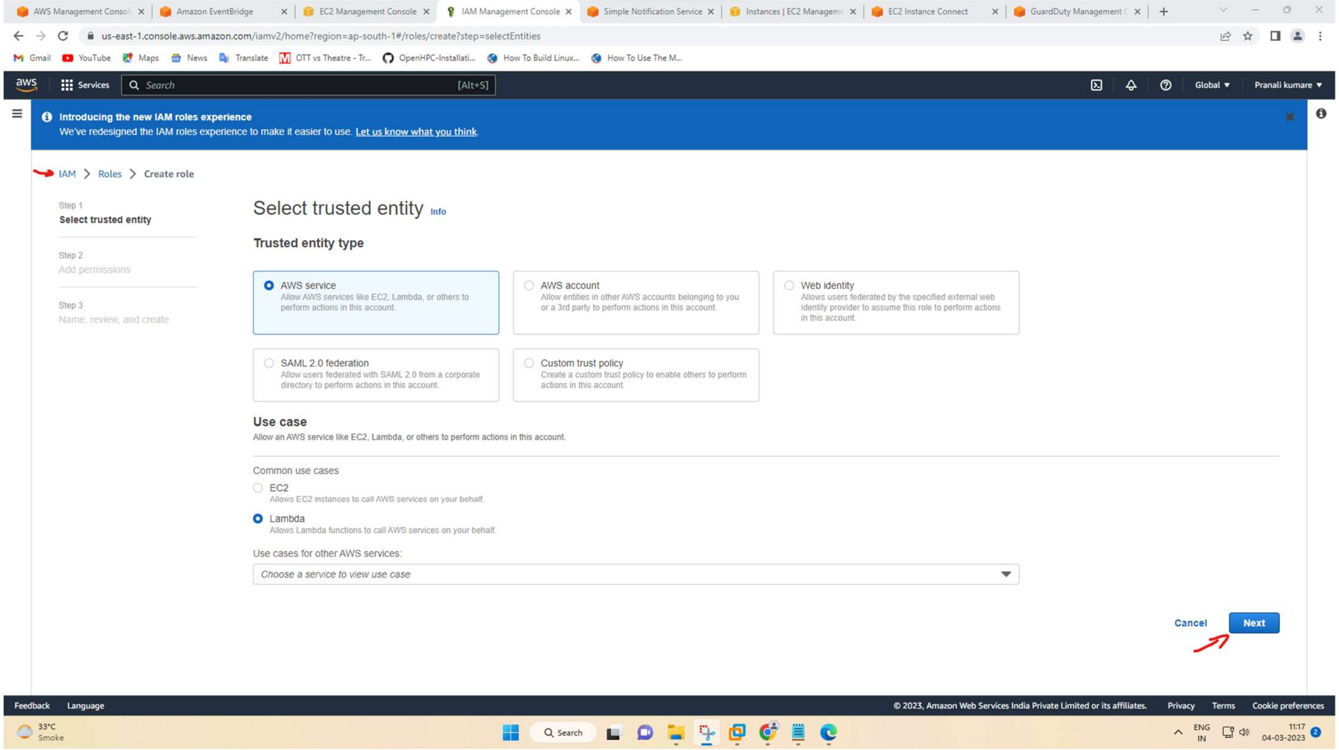
Security Groups (2) Info

Name	Security group ID	Security group name	VPC ID	Description	Owner	Inbound rules count	Outbound rules co...
–	sg-0033924582da00948	launch-wizard-1	vpc-0058ea9132c484099	launch-wizard-1 create...	628887948423	2 Permission entries	1 Permission entry
–	sg-02c44db9a05684c27	default	vpc-0058ea9132c484099	default VPC security gr...	628887948423	1 Permission entry	1 Permission entry



Security Groups (3) Info

	Name	Security group ID	Security group name	VPC ID	Description	Owner	Inbound rules count	Outbound rules co...
□	–	sg-0d4810d9205e1305b	compromised-ec2-sg	vpc-0058ea9132c484099	my_aws_project	628887948423	0 Permission entries	1 Permission entry
□	–	sg-0033924582da00948	launch-wizard-1	vpc-0058ea9132c484099	launch-wizard-1 create...	628887948423	2 Permission entries	1 Permission entry
□	–	sg-02c44db9a05684c27	default	vpc-0058ea9132c484099	default VPC security gr...	628887948423	1 Permission entry	1 Permission entry



Introducing the new IAM roles experience
We've redesigned the IAM roles experience to make it easier to use. Let us know what you think.

Step 1
Select trusted entity

Step 2
Add permissions

Step 3
Name, review, and create

Select trusted entity Info

Trusted entity type

● AWS service
Allow AWS services like EC2, Lambda, or others to perform actions in this account.

○ AWS account
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

○ Web identity
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.

○ SAML 2.0 federation
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

○ Custom trust policy
Create a custom trust policy to enable others to perform actions in this account.

Use case

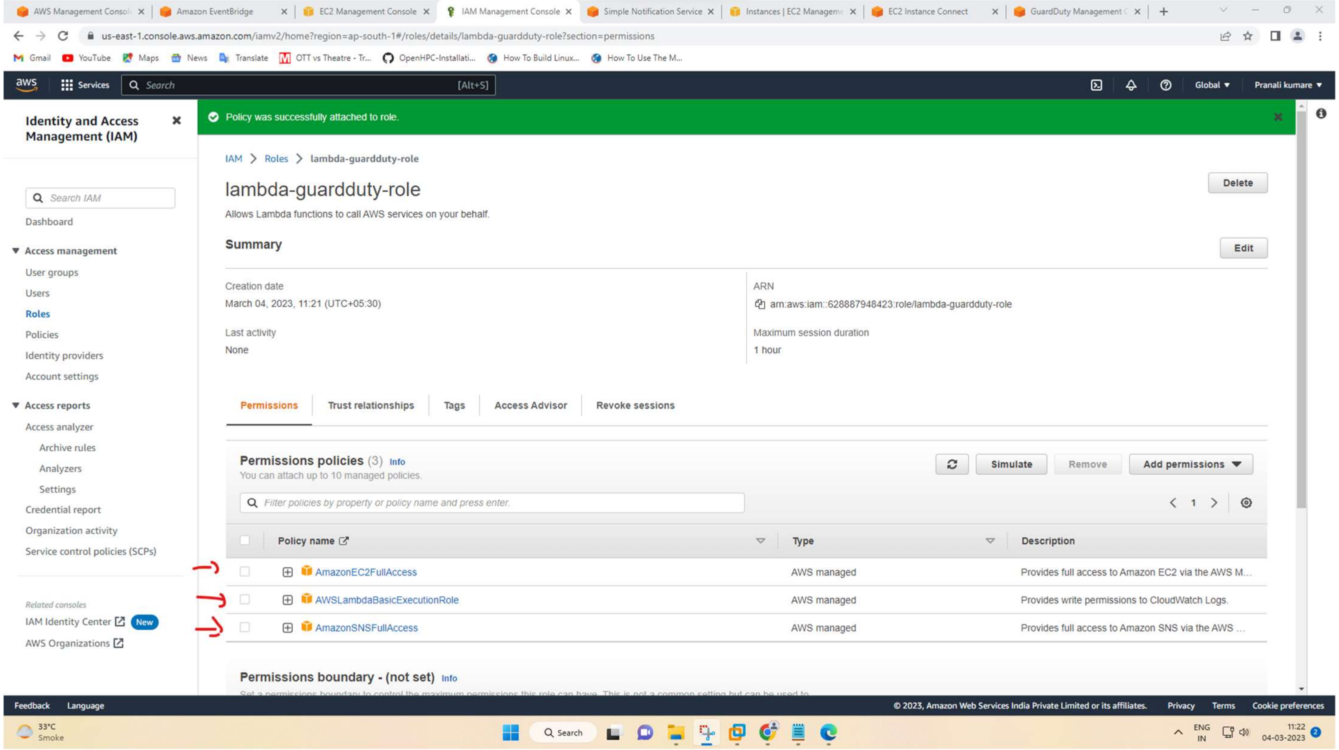
Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Common use cases

○ EC2
Allows EC2 instances to call AWS services on your behalf.

● Lambda
Allows Lambda functions to call AWS services on your behalf.

Use cases for other AWS services:
Choose a service to view use case



Policy was successfully attached to role.

IAM > Roles > lambda-guardduty-role

lambda-guardduty-role

Allows Lambda functions to call AWS services on your behalf.

Summary

Creation date: March 04, 2023, 11:21 (UTC+05:30)

ARN: arn:aws:iam::628887948423:role/lambda-guardduty-role

Last activity: None

Maximum session duration: 1 hour

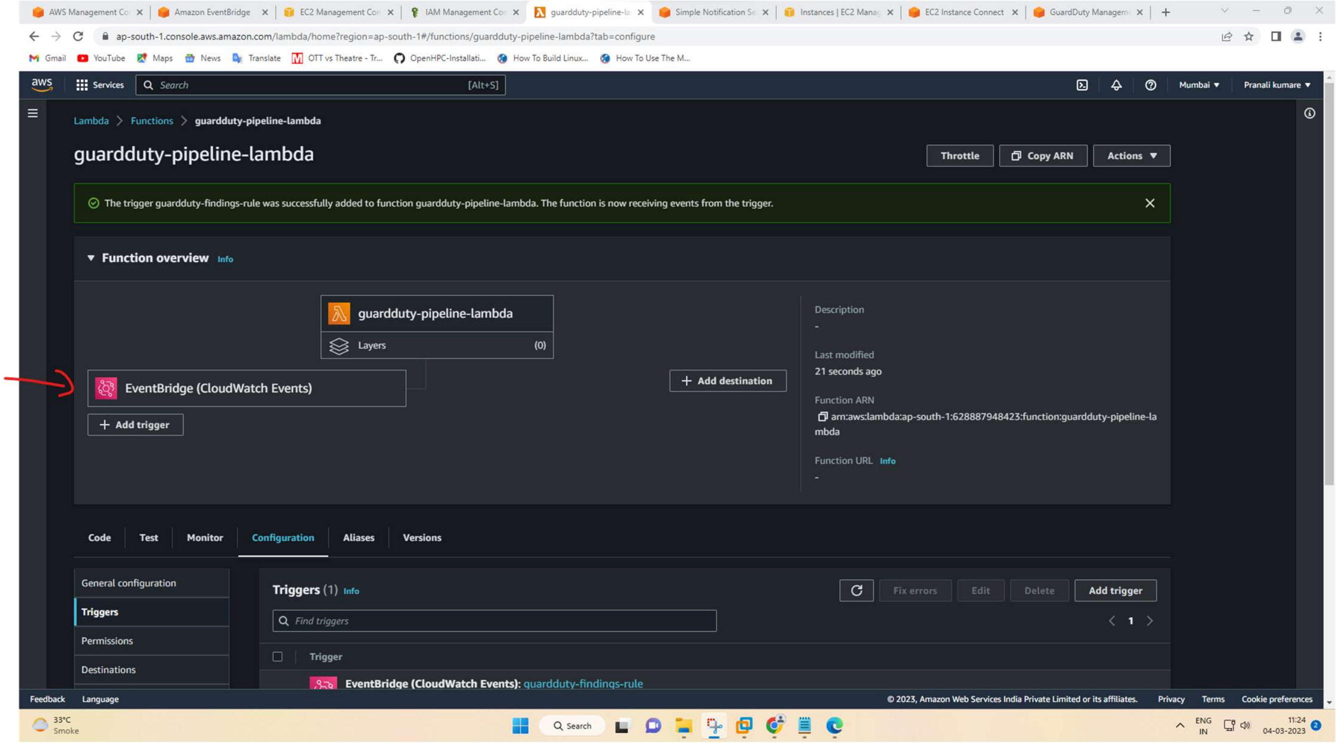
Permissions | Trust relationships | Tags | Access Advisor | Revoke sessions

Permissions policies (3) Info

You can attach up to 10 managed policies.

Policy name	Type	Description
AmazonEC2FullAccess	AWS managed	Provides full access to Amazon EC2 via the AWS M...
AWSLambdaBasicExecutionRole	AWS managed	Provides write permissions to CloudWatch Logs.
AmazonSNSFullAccess	AWS managed	Provides full access to Amazon SNS via the AWS ...

Permissions boundary - (not set) Info



Lambda > Functions > guardduty-pipeline-lambda

guardduty-pipeline-lambda

The trigger guardduty-findings-rule was successfully added to function guardduty-pipeline-lambda. The function is now receiving events from the trigger.

Function overview Info

guardduty-pipeline-lambda

Layers (0)

EventBridge (CloudWatch Events)

Description: -

Last modified: 21 seconds ago

Function ARN: arn:aws:lambda:ap-south-1:628887948423:function:guardduty-pipeline-lambda

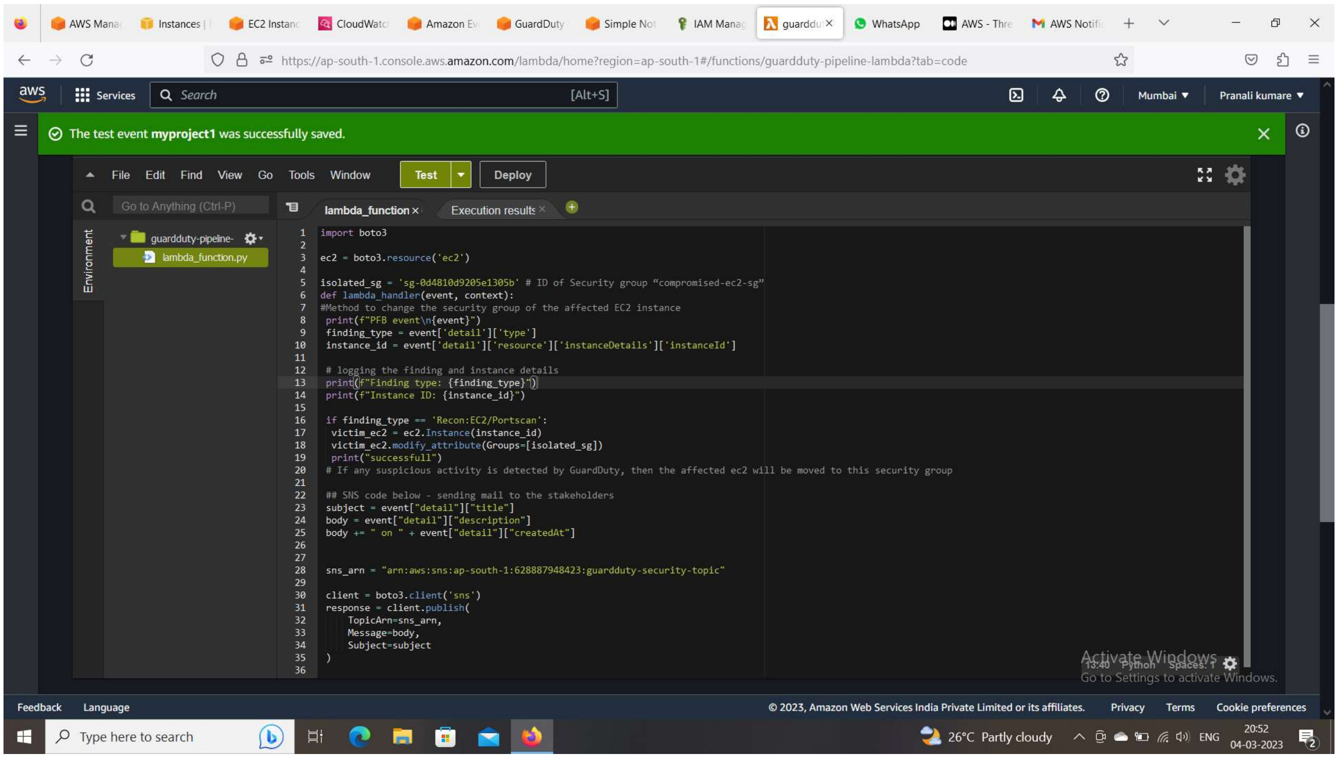
Function URL Info: -

Code | Test | Monitor | Configuration | Aliases | Versions

Triggers (1) Info

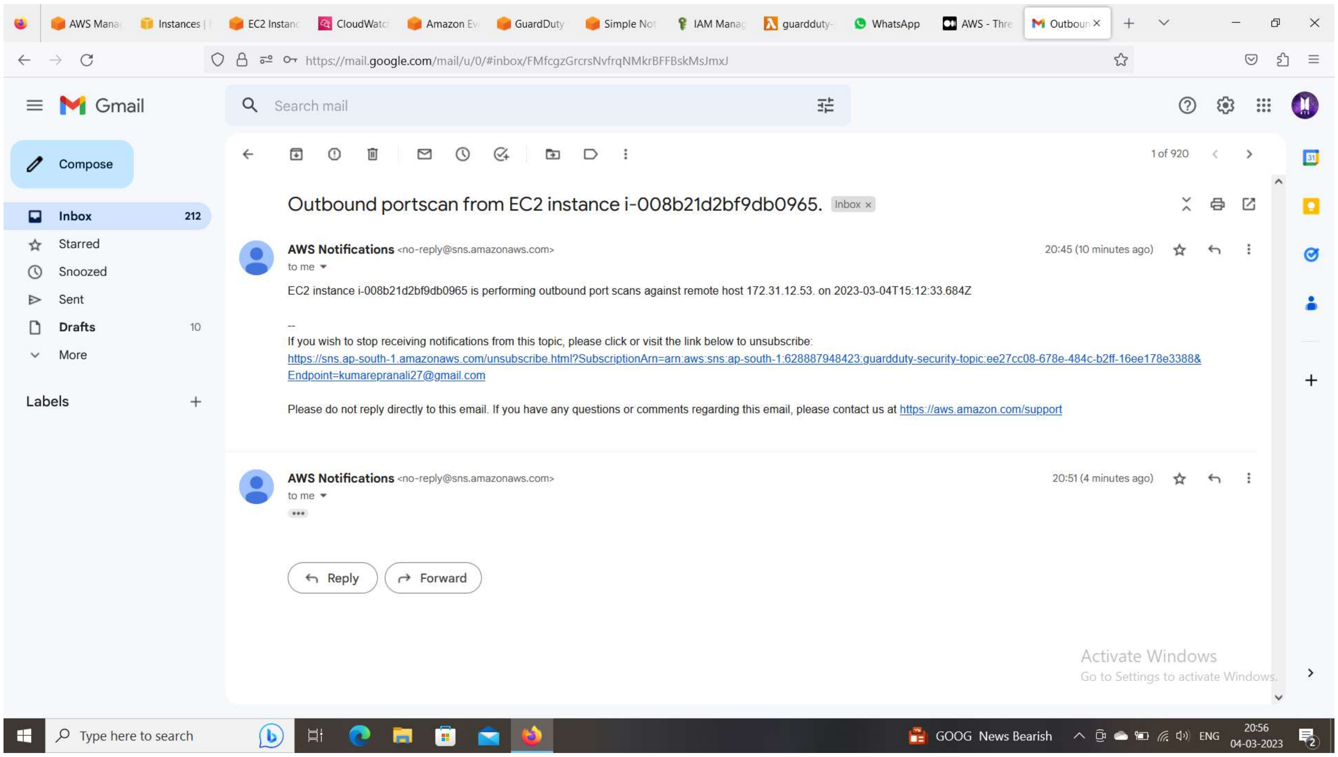
Trigger
EventBridge (CloudWatch Events): guardduty-findings-rule

Threat Prevention-



The test event **myproject1** was successfully saved.

```
import boto3

ec2 = boto3.resource('ec2')

isolated_sg = 'sg-0d4810d9205e1305b' # ID of Security group “compromised-ec2-sg”
def lambda_handler(event, context):
#Method to change the security group of the affected EC2 instance
 print(f"PFB event\n{event}")
 finding_type = event['detail']['type']
 instance_id = event['detail']['resource']['instanceDetails']['instanceId']

 # logging the finding and instance details
 print(f"Finding type: {finding_type}")
 print(f"Instance ID: {instance_id}")

 if finding_type == 'Recon:EC2/Portscan':
  victim_ec2 = ec2.Instance(instance_id)
  victim_ec2.modify_attribute(Groups=[isolated_sg])
  print("successfull")
 # If any suspicious activity is detected by GuardDuty, then the affected ec2 will be moved to this security group

 ## SNS code below - sending mail to the stakeholders
 subject = event["detail"]["title"]
 body = event["detail"]["description"]
 body += " on " + event["detail"]["createdAt"]


 sns_arn = "arn:aws:sns:ap-south-1:628887948423:guardduty-security-topic"

 client = boto3.client('sns')
 response = client.publish(
     TopicArn=sns_arn,
     Message=body,
     Subject=subject
 )

```



Outbound portscan from EC2 instance i-008b21d2bf9db0965.

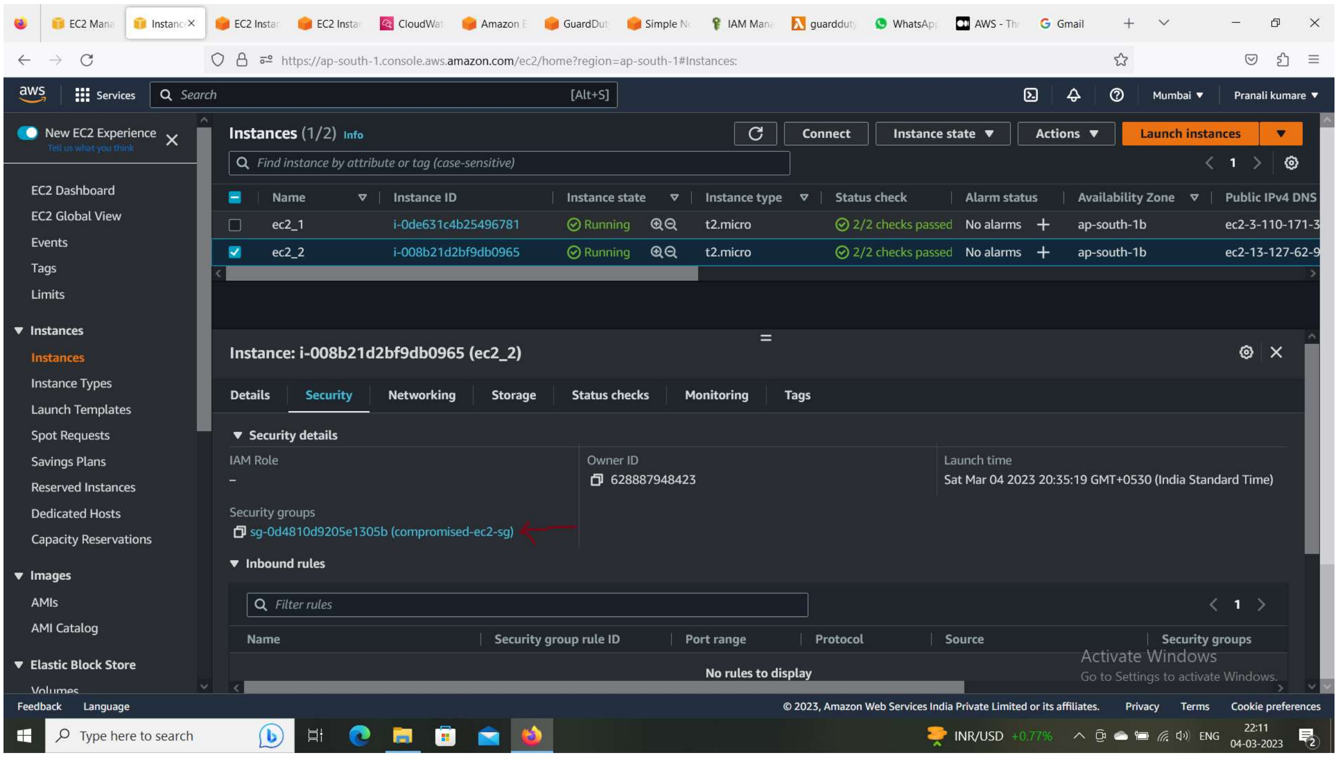
AWS Notifications <no-reply@sns.amazonaws.com>
to me
20:45 (10 minutes ago)

EC2 instance i-008b21d2bf9db0965 is performing outbound port scans against remote host 172.31.12.53. on 2023-03-04T15:12:33.684Z

--
If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:
https://sns.ap-south-1.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:ap-south-1:628887948423:guardduty-security-topic:ee27cc08-678e-484c-b2ff-16ee178e3388&Endpoint=kumarepranali27@gmail.com

Please do not reply directly to this email. If you have any questions or comments regarding this email, please contact us at https://aws.amazon.com/support

AWS Notifications <no-reply@sns.amazonaws.com>
to me
20:51 (4 minutes ago)



https://ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#Instances:

Instances (1/2) Info

☐	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
☐	ec2_1	i-0de631c4b25496781	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1b	ec2-3-110-171-3
☑	ec2_2	i-008b21d2bf9db0965	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1b	ec2-13-127-62-9

Instance: i-008b21d2bf9db0965 (ec2_2)

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

Security details

IAM Role: –

Owner ID: 628887948423

Launch time: Sat Mar 04 2023 20:35:19 GMT+0530 (India Standard Time)

Security groups: sg-0d4810d9205e1305b (compromised-ec2-sg)

Inbound rules

Name	Security group rule ID	Port range	Protocol	Source	Security groups
No rules to display					

8. Conclusion

In the case of AWS, services such as AWS GuardDuty, AWS Security Hub, and AWS Shield provide customers with real-time threat detection and response capabilities, enabling them to quickly identify and mitigate potential security incidents.

Additionally, AWS provides a range of security best practices, tools, and services to help customers improve their security posture and protect against cyber threats. These include services such as AWS Identity and Access Management (IAM), Security Groups , SNS (Simple Notification Service) among others.

Overall, AWS's commitment to security and its range of security services and solutions make it a trusted partner for customers looking to enhance their security posture and protect against cyber threats. However, the effectiveness of any specific project would depend on the particular use case and implementation.

9. Bibliography

- Cloud Computing Black Paperback by Kailash Jayaswal, Jagnnath Kallakurchi, Donald J. Houde, Dr. Deven Shah
- https://boto3.amazonaws.com/v1/documentation/api/latest/reference/services/sns.html
- https://docs.aws.amazon.com/
- Cloud Computing: Concepts, Technology and Architecture by Erl
- https://www.javatpoint.com/aws-lambda