

CLOUD SECURITY & MANAGEMENT
LAB

Name: Harsh Ranjan

SAP ID: 500097019

Roll no: R2142211262

Batch :B7

SUBMISSION TO:

Ms. Avita Katal

Experiment 6 a)-Cloud Monitoring Tool Cloudwatch integrated with EC2

STEP 1: Firstly, go to IAM and create a role for ec2 instance for these following 2 access controls:

- Ec2 Full Access
- CloudWatch Full Access

The image displays two screenshots of the AWS IAM console interface.

The top screenshot shows the 'Roles' page. The left sidebar contains the 'Identity and Access Management (IAM)' menu with options like Dashboard, Access management, User groups, Users, Roles, Policies, Identity providers, Account settings, Access reports, Access Analyzer, External access, Unused access, Analyzer settings, and Credential report. The main content area shows 'Roles (2)' with a search bar and a table listing roles. The table has columns for Role name, Trusted entities, and Last activity. Two roles are listed: 'AWSServiceRoleForSupport' and 'AWSServiceRoleForTrustedAdvisor'. Below the table, there is a 'Roles Anywhere' section with a 'Manage' button and three cards: 'Access AWS from your non AWS workloads', 'X.509 Standard', and 'Temporary credentials'.

The bottom screenshot shows the 'Create role' wizard, Step 1: Select trusted entity. The left sidebar shows the progress: Step 1 (Select trusted entity), Step 2 (Add permissions), and Step 3 (Name, review, and create). The main content area is titled 'Select trusted entity' and shows five options for 'Trusted entity type': 'AWS service' (selected), 'AWS account', 'Web identity', 'SAML 2.0 federation', and 'Custom trust policy'. Below this, the 'Use case' section shows 'Service or use case' set to 'EC2' and a 'Choose a use case for the specified service.' section with 'EC2' selected.

Permissions policies (2) Info

Simulate

Remove

Add permissions

You can attach up to 10 managed policies.

Search

Filter by Type

All types

< 1 >

<input type="checkbox"/>	Policy name	Type	Attached entities
<input type="checkbox"/>	AmazonEC2FullAccess	AWS managed	1
<input type="checkbox"/>	CloudWatchFullAccess	AWS managed	1

STEP 2 : Assign the name of the role.

Create role | IAM | Global

us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/roles/create?selectedUseCase=EC2&trustedEntityType=AWS_SERVIC...

GD8 online Debug... YouTube Pre Post SEO : Onl... Mahabharat all epis... Youtube Multi Dow... Free Online Video... Telegram Web WhatsApp Gemini Healer | Watch with...

aws Services Search [Alt+S]

Global Nidhi Mishra

IAM > Roles > Create role

Step 1

Select trusted entity

Step 2

Add permissions

Step 3

Name, review, and create

Name, review, and create

Role details

Role name

Enter a meaningful name to identify this role.

cloudwatch_ec2_full_access_roles

Maximum 64 characters. Use alphanumeric and '+,=,@,-,_' characters.

Description

Add a short explanation for this role.

Allows EC2 instances to call AWS services on your behalf.

Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following characters: '+,=,@,-,_[\]]#\$%&'*~:;<>'

Step 1: Select trusted entities

Edit

Trust policy

1 = {

2 "Version": "2012-10-17",

3 "Statement": [

STEP 3: Now go to EC2 tab and launch a Amazon AMI 2 Machine and attach the created IAM role to that ec2 in advanced tab.

Instances | EC2 | us-east-1

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#instances:

GD8 online Debug... YouTube Pre Post SEO : Onl... Mahabharat all epis... Youtube Multi Dow... Free Online Video... Telegram Web WhatsApp Gemini Healer | Watch with...

aws Services Search [Alt+S]

N. Virginia Nidhi Mishra

EC2 Dashboard

EC2 Global View

Events

Console-to-Code [Preview](#)

▼ Instances

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity

Reservations [New](#)

▼ Images

AMIs

AMI Catalog

▼ Elastic Block Store

Volumes

Instances Info

Connect

Instance state

Actions

Launch instances

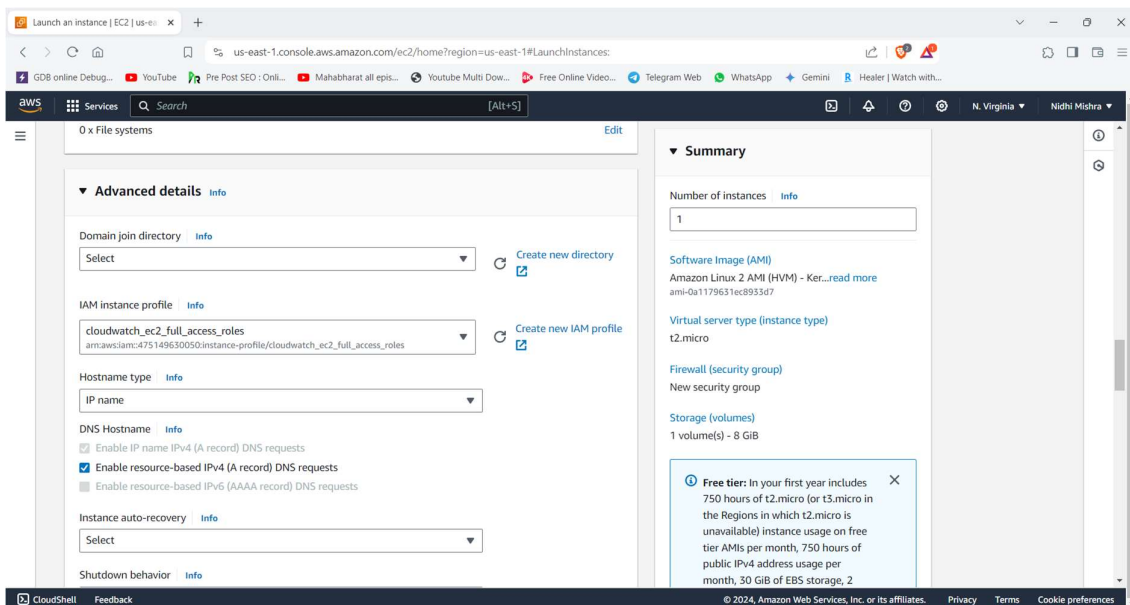
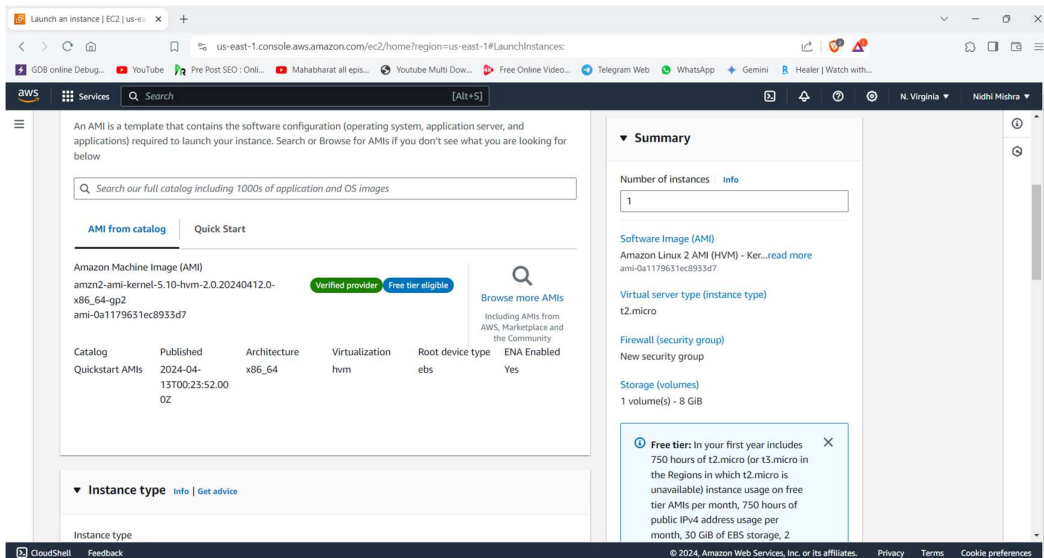
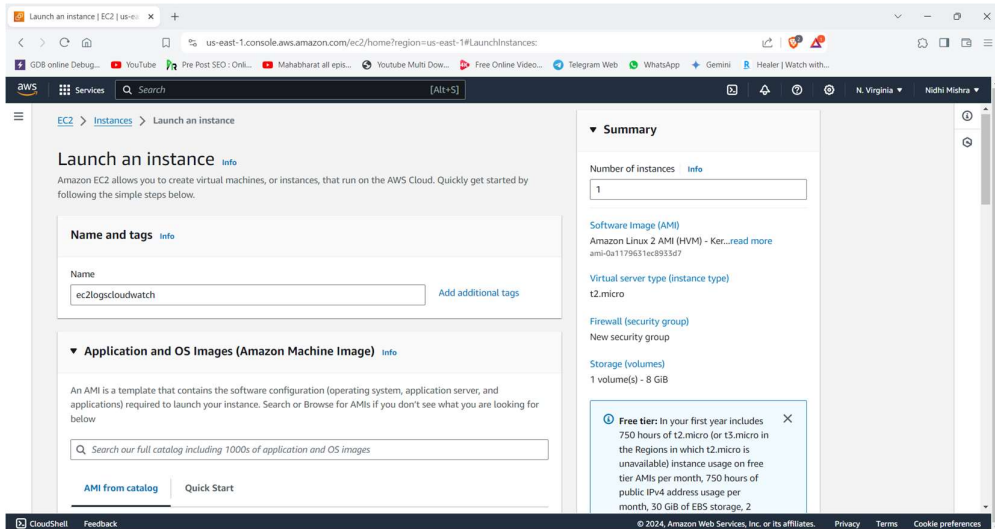
Find Instance by attribute or tag (case-sensitive)

All states

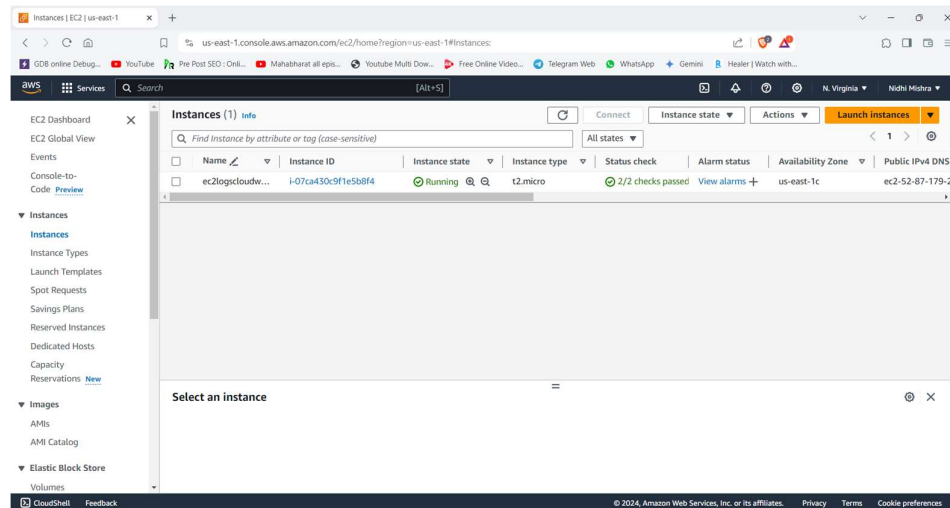
< 1 >

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
No instances								
You do not have any instances in this region								
Launch instances								

Select an instance

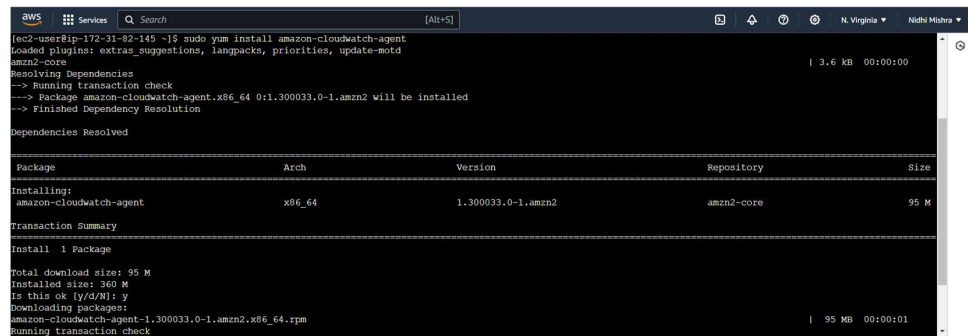


STEP 4: Now wait for the instance to get ready and then connect it.



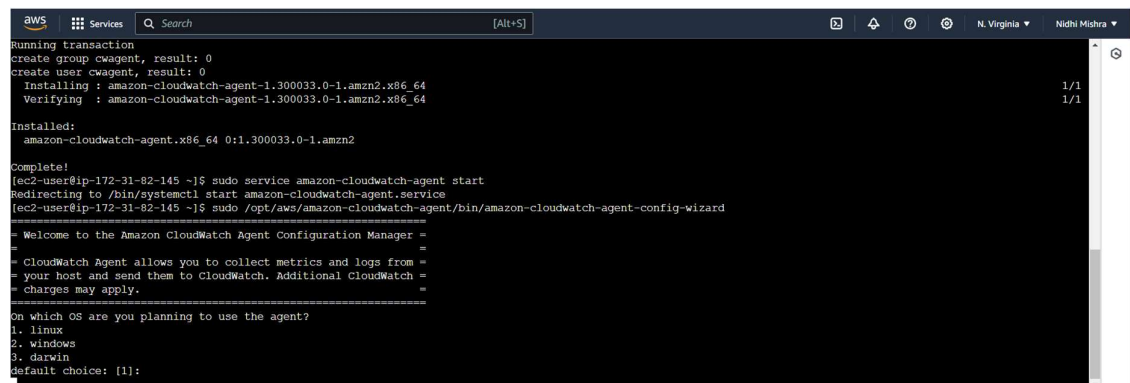
STEP 5: Use the following command to install CloudWatch agent in it.

- `sudo yum install amazon-cloudwatch-agent`



STEP 6: Now open the CloudWatch agent using the following command:

- `sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-config-wizard`



STEP 7: Now Choose the options as per your need or as shown below

```
aws Services Search [Alt+S] N. Virginia Nidhi Mishra
On which OS are you planning to use the agent?
1. linux
2. windows
3. darwin
default choice: [1]:
1
Trying to fetch the default region based on ec2 metadata...
!! imds retry client will retry 1 timesAre you using EC2 or On-Premises hosts?
1. EC2
2. On-Premises
default choice: [1]:
1
Which user are you planning to run the agent?
1. cwagent
2. root
3. others
default choice: [1]:
2
Do you want to turn on StatsD daemon?
1. yes
2. no
default choice: [1]:
1
Which port do you want StatsD daemon to listen to?
default choice: [8125]
```

```
aws Services Search [Alt+S] N. Virginia Nidhi Mishra
Which port do you want StatsD daemon to listen to?
default choice: [8125]

What is the collect interval for StatsD daemon?
1. 10s
2. 30s
3. 60s
default choice: [1]:
1

What is the aggregation interval for metrics collected by StatsD daemon?
1. Do not aggregate
2. 10s
3. 30s
4. 60s
default choice: [4]:
4

Do you want to monitor metrics from CollectD? WARNING: CollectD must be installed or the Agent will fail to start
1. yes
2. no
default choice: [1]:
2

Do you want to monitor any host metrics? e.g. CPU, memory, etc.
1. yes
2. no
default choice: [1]:
```

```
aws Services Search [Alt+S] N. Virginia Nidhi Mishra
1. yes
2. no
default choice: [1]:
1

Do you want to monitor cpu metrics per core?
1. yes
2. no
default choice: [1]:
1

Do you want to add ec2 dimensions (ImageId, InstanceId, InstanceType, AutoScalingGroupName) into all of your metrics if the info is available?
1. yes
2. no
default choice: [1]:
1

Do you want to aggregate ec2 dimensions (InstanceId)?
1. yes
2. no
default choice: [1]:
1

Would you like to collect your metrics at high resolution (sub-minute resolution)? This enables sub-minute resolution for all metrics, but you can customize for specific metrics in the output json file.
1. 1s
2. 10s
3. 30s
4. 60s
default choice: [4]:
```

STEP 8: In path write: /var/log/messages

```
aws Services Search [Alt+S] N. Virginia Nidhi Mishra
Are you satisfied with the above config? Note: it can be manually customized after the wizard completes to add additional items.
1. yes
2. no
default choice: [1]:
1

Do you have any existing CloudWatch Log Agent (http://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/AgentReference.html) configuration file to import for migration?
1. yes
2. no
default choice: [2]:
2

Do you want to monitor any log files?
1. yes
2. no
default choice: [1]:
1

Log file path:
/var/log/messages
Log group name:
default choice: [messages]
ec2logscloudwatchcheckpcam
Log group class:
1. STANDARD
2. INFREQUENT_ACCESS
default choice: [1]:
1
```

```
        "metrics_collection_interval": 60
    },
    "statsd": {
        "metrics_aggregation_interval": 60,
        "metrics_collection_interval": 10,
        "service_address": ":8125"
    },
    "swap": {
        "measurement": [
            "swap_used_percent"
        ],
        "metrics_collection_interval": 60
    }
}
}
}

Please check the above content of the config.
The config file is also located at /opt/aws/amazon-cloudwatch-agent/bin/config.json.
Edit it manually if needed.
Do you want to store the config in the SSM parameter store?
1. yes
2. no
default choice: [1]:
2
Program exits now.
[ec2-user@ip-172-31-82-145 ~]$
```

STEP 9: Now go to /opt/aws/amazon-cloudwatch-agent/bin folder using command:

- `cd /opt/aws/amazon-cloudwatch-agent/bin`

and list using: “ls”

```

Please check the above content of the config.
The config file is also located at /opt/aws/amazon-cloudwatch-agent/bin/config.json.
Edit it manually if needed.
Do you want to store the config in the SSM parameter store?
1. yes
2. no
default choice: [1]:
2
Program exits now.
[ec2-user@ip-172-31-82-145 ~]$ cd /opt/aws/amazon-cloudwatch-agent/bin
[ec2-user@ip-172-31-82-145 bin]$ ls
amazon-cloudwatch-agent  amazon-cloudwatch-agent-ctl  config.json  CWAGENT_VERSION
amazon-cloudwatch-agent-config-wizard  config-downloader  config-translator  start-amazon-cloudwatch-agent
[ec2-user@ip-172-31-82-145 bin]$
```

STEP 10: Validate the config.json file using following command:

- `sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m ec2 -s -c file:configuration-file-path`

For Example, here path is /opt/aws/amazon-cloudwatch-agent/bin/config.json

```
aws  Services  Search  [Alt+S]  N. Virginia  Nidhi Mishra
[ec2-user@ip-172-31-82-145 ~]$ cd /opt/aws/amazon-cloudwatch-agent/bin
[ec2-user@ip-172-31-82-145 bin]$ ls
amazon-cloudwatch-agent  amazon-cloudwatch-agent-ctl  config.json  CWAGENT_VERSION
amazon-cloudwatch-agent-config-wizard  config-downloader  config-translator  start-amazon-cloudwatch-agent
[ec2-user@ip-172-31-82-145 bin]$ sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m ec2 -s -c file:/opt/aws/amazon-cloudwatch-agent/bin/config.json
***** processing amazon-cloudwatch-agent *****
I! Trying to detect region from ec2 D! [EC2] Found active network interface I! imds retry client will retry 1 timesSuccessfully fetched the config and saved in /opt/a
ws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.d/file_config.json.tmp
Start configuration validation...
2024/04/22 06:54:27 Reading json config file path: /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.d/file_config.json.tmp ...
2024/04/22 06:54:27 I! Valid json input schema.
2024/04/22 06:54:27 D! ec2tagger processor required because append_dimensions is set
2024/04/22 06:54:27 D! delta processor required because metrics with diskio or net are set
2024/04/22 06:54:27 D! ec2tagger processor required because append_dimensions is set
2024/04/22 06:54:27 Configuration validation first phase succeeded
I! Detecting run as user...
I! Trying to detect region from ec2
D! [EC2] Found active network interface
I! imds retry client will retry 1 times
/opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent -schematest -config /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.toml
Configuration validation second phase succeeded
Configuration validation succeeded
amazon-cloudwatch-agent has already been stopped
Created symlink from /etc/systemd/system/multi-user.target.wants/amazon-cloudwatch-agent.service to /etc/systemd/system/amazon-cloudwatch-agent.service.
[ec2-user@ip-172-31-82-145 bin]$
```

STEP 11: Now install the stress package using following command and stress it to test the cpu utilization:

- `sudo amazon-linux-extras install epel -y` # Enable the EPEL repository
- `sudo yum install stress -y` # Install the stress package


```
aws [Services] Search [Alt+S] N. Virginia Nidhi Mishra
[ec2-user@ip-172-31-82-145 bin]$ sudo amazon-linux-extras install epel -y # Enable the EPEL repository
Installing epel-release
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Cleaning repos: amzn2-core amzn2extra-docker amzn2extra-epel amzn2extra-kernel-5.10
17 metadata files removed
6 sqlite files removed
0 metadata files removed
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core | 3.6 kB 00:00:00
amzn2extra-docker | 2.9 kB 00:00:00
amzn2extra-epel | 3.0 kB 00:00:00
amzn2extra-kernel-5.10 | 3.0 kB 00:00:00
(1/9): amzn2-core/2/x86_64/group.gz | 2.7 kB 00:00:00
(2/9): amzn2-core/2/x86_64/updateinfo | 839 kB 00:00:00
(3/9): amzn2extra-epel/2/x86_64/primary.db | 1.8 kB 00:00:00
(4/9): amzn2extra-kernel-5.10/2/x86_64/updateinfo | 56 kB 00:00:00
(5/9): amzn2extra-docker/2/x86_64/updateinfo | 15 kB 00:00:00
(6/9): amzn2extra-epel/2/x86_64/updateinfo | 76 B 00:00:00
(7/9): amzn2extra-docker/2/x86_64/primary.db | 106 kB 00:00:00
(8/9): amzn2extra-kernel-5.10/2/x86_64/primary.db | 25 MB 00:00:00
(9/9): amzn2-core/2/x86_64/primary.db | 73 MB 00:00:01
Resolving Dependencies
--> Running transaction check
--> Package epel-release.noarch 0:7-11 will be installed
--> Finished Dependency Resolution
```

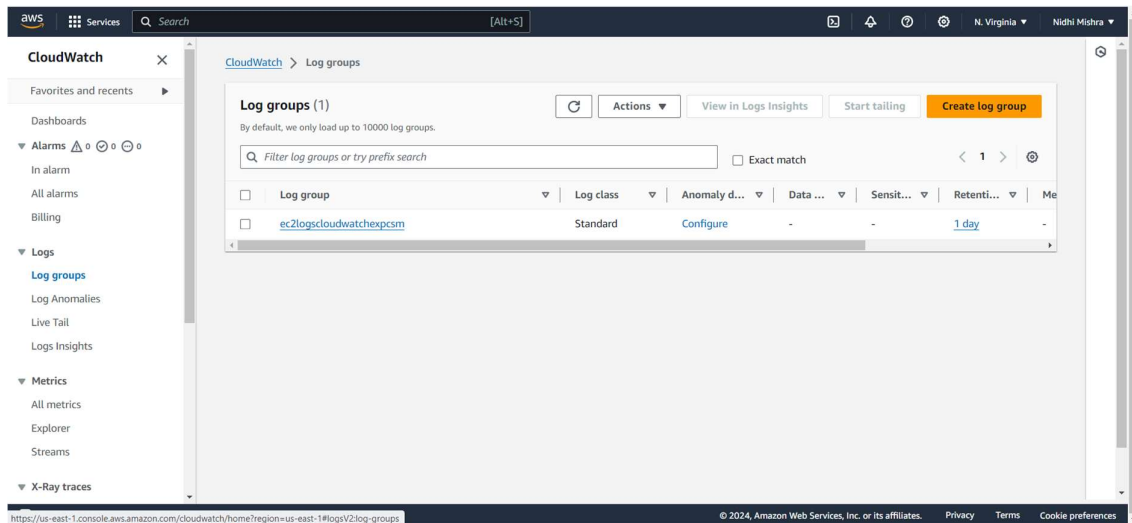
STEP 12: Use stress command:

- stress -cpu 1

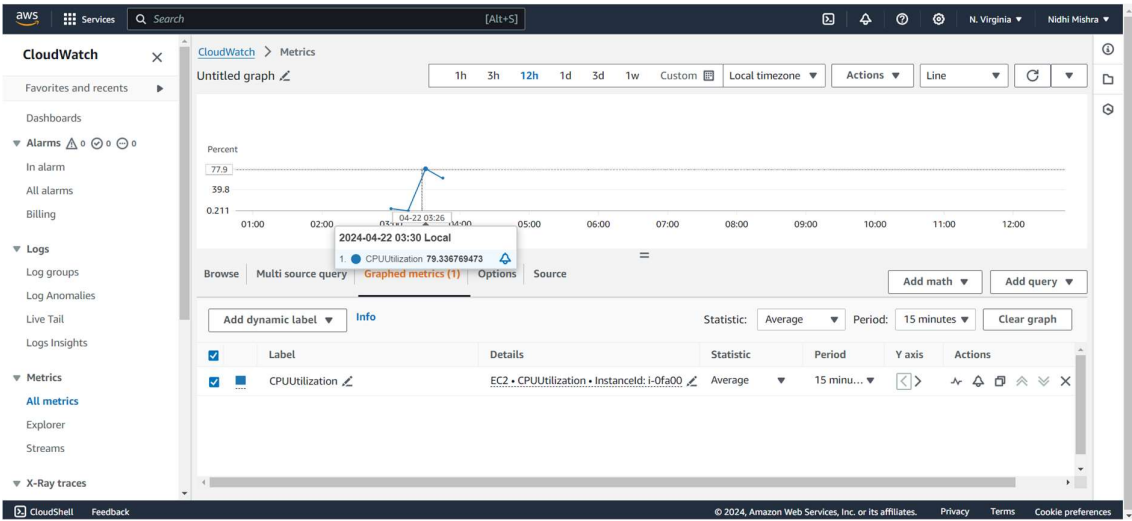
```
Installed:
stress.x86_64 0:1.0.4-16.el7

Complete!
[ec2-user@ip-172-31-82-145 bin]$ stress --cpu 1
stress: info: [862] dispatching hogs: 1 cpu, 0 io, 0 vm, 0 hdd
```

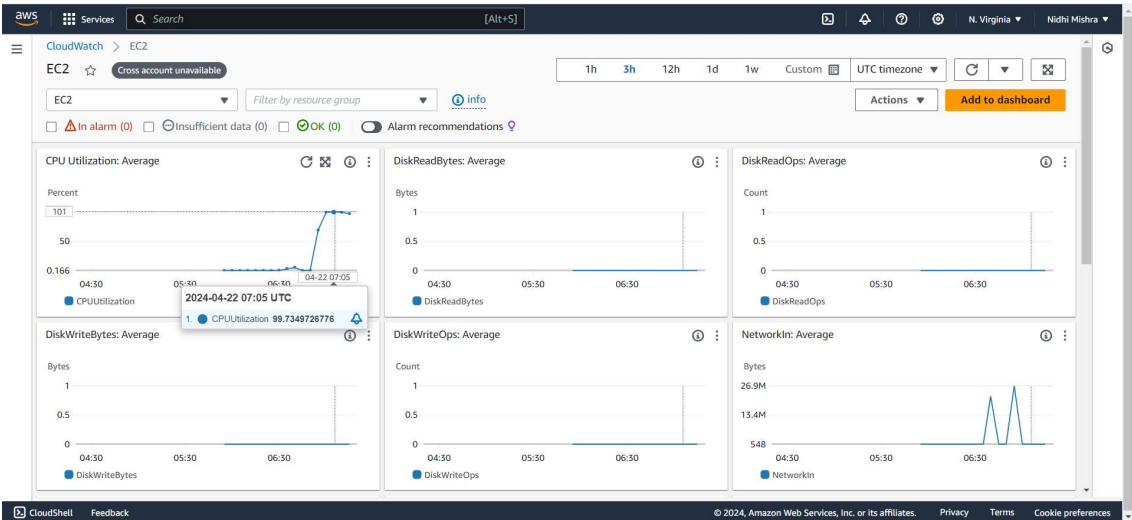
STEP 13: Now go to CloudWatch and go to Log Groups tab and see whether log files are created or not.



STEP 14: Go to CloudWatch, Matrices Tab and in EC2 tab select to get matrices:



STEP 15: You can check the graph of all EC2 resources.



QUESTIONS:

1. What is CloudWatch and how does it integrate with EC2 instances?

CloudWatch is a monitoring service provided by AWS that allows you to collect and track metrics, logs, and events from various AWS resources in real-time. It provides insights into the operational health of your AWS infrastructure and applications. CloudWatch integrates seamlessly with EC2 instances by automatically collecting and monitoring key performance metrics such as CPU utilization, disk I/O, and network traffic. This integration enables you to monitor the performance and health of your EC2 instances and take proactive measures to optimize their usage and performance.

2. Explain the key metrics monitored by CloudWatch for EC2 instances:

CloudWatch monitors a range of metrics for EC2 instances, including:

- CPU utilization: Percentage of CPU capacity in use.
 - Disk metrics: Read/write operations and throughput on instance storage volumes.
 - Network metrics: Incoming and outgoing network traffic.
 - Status checks: Checks on the instance's system status and instance status.
- These metrics provide valuable insights into the performance and health of EC2 instances, helping you to optimize resource utilization, identify performance bottlenecks, and troubleshoot issues.

3. How do you enable detailed monitoring for EC2 instances in CloudWatch and what benefits does it provide?

Detailed monitoring for EC2 instances provides metrics at a granularity of one-minute intervals, compared to the default five-minute intervals of basic monitoring. You can enable detailed monitoring during instance launch or modify it later through the AWS Management Console, CLI, or API. Detailed monitoring offers more granular visibility into the performance of your EC2 instances, especially for workloads with rapid scaling or short-lived bursts of activity. This enables you to capture and analyze performance trends more accurately, making it easier to optimize resource allocation and respond to dynamic workload demands.

4. Explain the process of setting up CloudWatch alarms for EC2 instances, and what actions can be triggered by these alarms:

To set up CloudWatch alarms for EC2 instances, you first define the metric you want to monitor (e.g., CPU utilization), set the threshold value, and configure the alarm action. Alarm actions can include sending notifications via Amazon SNS (Simple Notification Service), triggering Auto Scaling actions to automatically scale your EC2 fleet based on demand, executing an AWS Lambda function, or stopping/terminating instances. CloudWatch alarms help you proactively monitor the health and performance of your EC2 instances and take automated actions based on predefined thresholds.

5. Outline the steps to configure CloudWatch logs for EC2 instances:

Configuring CloudWatch logs for EC2 instances involves several steps:

- Install the CloudWatch Logs agent on the EC2 instance.
- Configure the agent to monitor log files and send log data to CloudWatch Logs.
- Create log groups and log streams in CloudWatch Logs to organize and store the log data.
- Optionally, configure log retention settings and access controls for the log data. By configuring CloudWatch logs, you can centralize and analyze log data from multiple EC2 instances in a single location, enabling you to monitor application and system-level logs, troubleshoot issues, and comply with regulatory requirements.

6. Difference between basic monitoring and detailed monitoring in CloudWatch:

Basic monitoring provides metrics at five-minute intervals and is enabled by default for all AWS resources, including EC2 instances. It provides a cost-effective way to monitor resource utilization and performance. Detailed monitoring, on the other hand, provides metrics at one-minute intervals and offers more granular visibility into resource utilization, especially for workloads with rapid scaling or short-lived bursts of activity. Detailed monitoring incurs additional charges compared to basic monitoring but provides more accurate insights for real-time monitoring and analysis.

7. What is the CloudWatch agent, and how does it enhance monitoring capabilities for EC2 instances?

The CloudWatch agent is a lightweight software component that runs on EC2 instances and enables you to collect system-level metrics, logs, and custom metrics. It enhances monitoring capabilities by providing additional insights into the performance and health of EC2 instances beyond what's available by default in CloudWatch. The agent can collect metrics such as memory usage, disk space, and custom application metrics, as well as monitor log files and send log data to CloudWatch Logs. This enables you to gain deeper visibility into your EC2 instances and applications, troubleshoot issues more effectively, and optimize resource utilization.

8. How do you use CloudWatch dashboards to visualize metrics?

CloudWatch dashboards allow you to create custom dashboards to visualize metrics from various AWS resources, including EC2 instances. You can add widgets to the dashboard, such as line charts, bar charts, or numerical displays, to visualize specific metrics and performance trends. You can customize the layout and design of the dashboard to suit your monitoring needs, such as grouping related metrics, configuring refresh intervals, and defining dashboard permissions. CloudWatch dashboards provide a centralized and customizable way to monitor the health and performance of your AWS infrastructure and applications in real-time.

9. What is AWS Lambda function logs from CloudWatch to S3?

AWS Lambda function logs from CloudWatch to S3 involves configuring CloudWatch Logs to export log data from Lambda functions to an Amazon S3 bucket. This allows you to archive and retain Lambda function logs for long-term storage, analysis, and compliance purposes. You can configure log exports using the AWS Management Console, CLI, or SDKs, specifying the log group, S3 bucket, and optional prefix for storing the log data. Once configured, CloudWatch Logs automatically exports log data to the specified S3 bucket, making it available for further analysis and archival outside of CloudWatch.

10. What is the role of IAM in configuring log exports?

IAM (Identity and Access Management) plays a crucial role in configuring log exports by defining the permissions required for CloudWatch to access and export log data to S3 buckets. You need to create IAM policies that grant permissions for CloudWatch to read log data from CloudWatch Logs and write log data to S3 buckets. These policies specify the actions allowed (e.g., **logs:CreateExportTask**, **s3:PutObject**) and the resources (e.g., log groups, S3 buckets) that CloudWatch can access. You then attach these policies to IAM roles used by CloudWatch, ensuring that the service has the necessary permissions to export log data securely and reliably.