

Project Title: AWS CloudWatch Dashboards for Billing, Logs, Network Performance, and Security & Compliance Monitoring

Objective-:

To create a practical, low-cost AWS monitoring system using CloudWatch. The goal was to build four interactive dashboards to track billing, application logs, network activity, and security & compliance using CloudWatch Metrics, Logs Insights, CloudWatch Agent, GuardDuty, and AWS Config.

Services used:

- Amazon CloudWatch (Dashboards, Metrics, Logs Insights)
- AWS Config (for compliance)
- AWS GuardDuty (for security threat detection)
- AWS CloudTrail (for API monitoring)
- IAM (for access control)
- EC2 (host to push logs)
- S3 (used for AWS Config)
- Application Load Balancer (for network monitoring)

Dashboard 1: Billing & Cost Monitoring:

Goal:

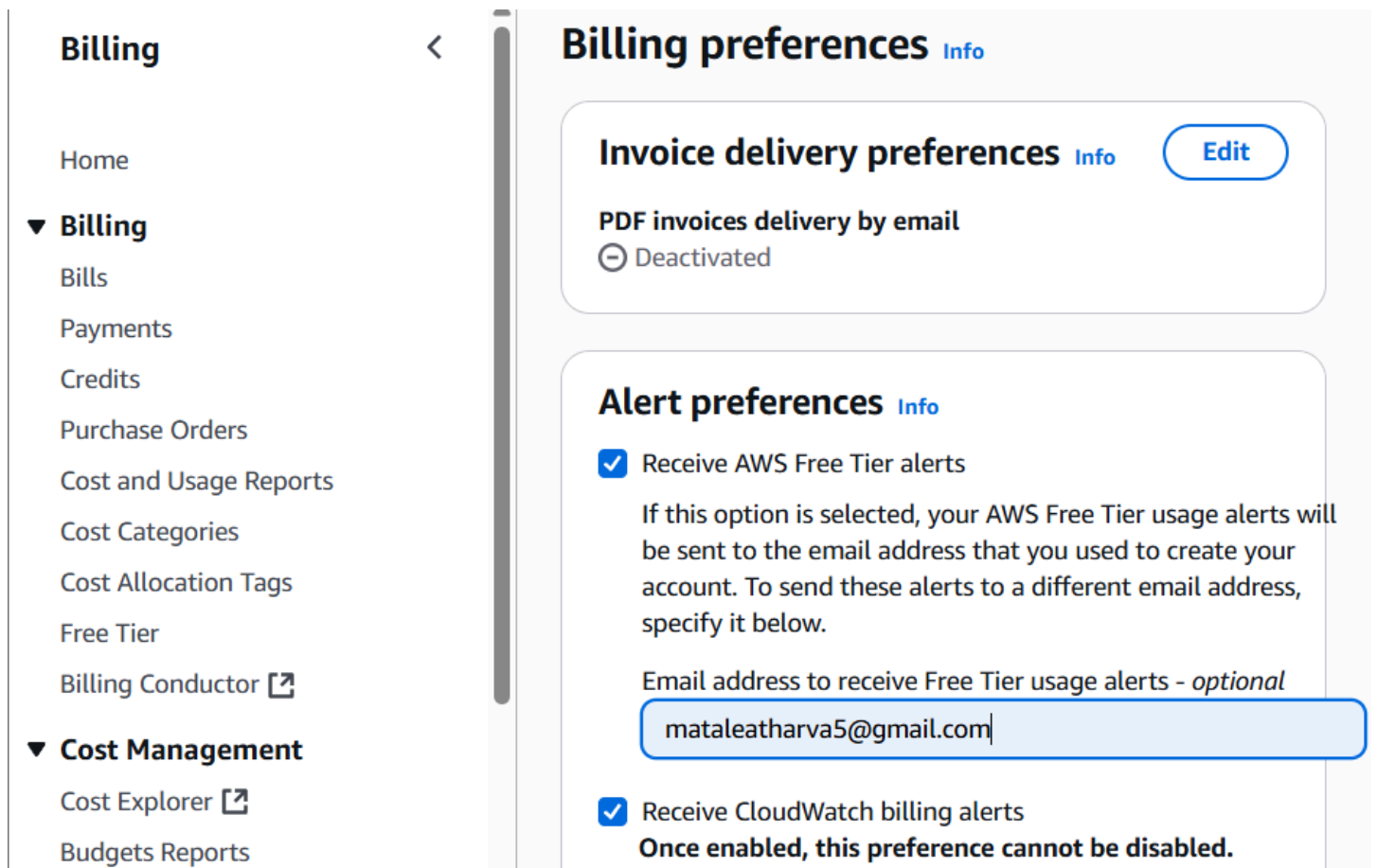
This dashboard helps you keep track of your AWS spending in real time.

It shows how much you're being charged overall and which services are costing the most, so you can manage your budget better.

Steps:

1) Enable Billing Alerts:

- i) Navigate to *Billing* → *Preferences* → *Receive Billing Alerts*.
- ii) This step ensures billing metrics appear in CloudWatch.



The screenshot shows the AWS Billing console interface. On the left is a navigation menu with 'Billing' and 'Cost Management' sections. The main content area is titled 'Billing preferences' and contains two sections: 'Invoice delivery preferences' and 'Alert preferences'. The 'Invoice delivery preferences' section shows 'PDF invoices delivery by email' is 'Deactivated'. The 'Alert preferences' section has two checked options: 'Receive AWS Free Tier alerts' and 'Receive CloudWatch billing alerts'. The 'Receive AWS Free Tier alerts' section includes a text input field for an email address, which contains 'mataleatharva5@gmail.com'. The 'Receive CloudWatch billing alerts' section has a note: 'Once enabled, this preference cannot be disabled.'

Billing

- Home
- ▼ **Billing**
 - Bills
 - Payments
 - Credits
 - Purchase Orders
 - Cost and Usage Reports
 - Cost Categories
 - Cost Allocation Tags
 - Free Tier
 - Billing Conductor [↗](#)
- ▼ **Cost Management**
 - Cost Explorer [↗](#)
 - Budgets Reports

Billing preferences [Info](#)

Invoice delivery preferences [Info](#) [Edit](#)

PDF invoices delivery by email

⊖ Deactivated

Alert preferences [Info](#)

☒ Receive AWS Free Tier alerts

If this option is selected, your AWS Free Tier usage alerts will be sent to the email address that you used to create your account. To send these alerts to a different email address, specify it below.

Email address to receive Free Tier usage alerts - *optional*

☒ Receive CloudWatch billing alerts

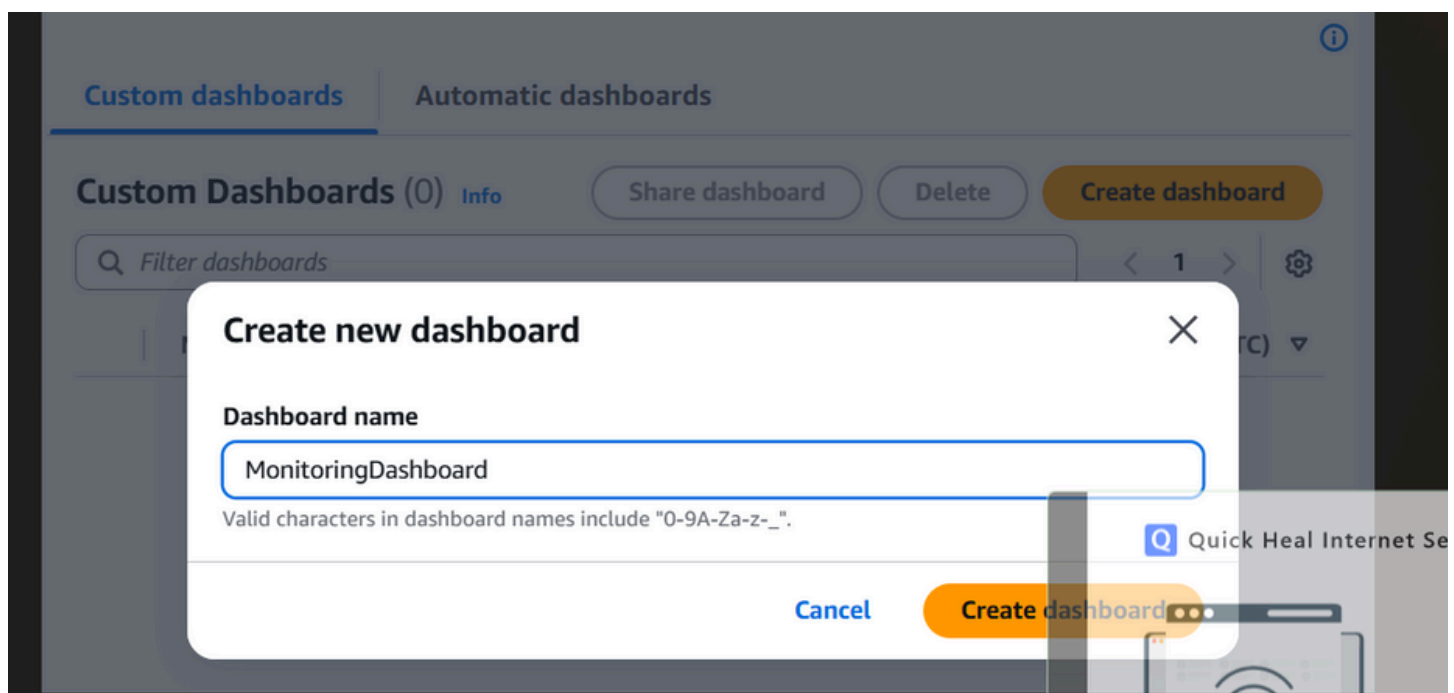
Once enabled, this preference cannot be disabled.

iii)we can see that the metrics are available in the cloud watch metrics.

2)Creating Dashboard:

Go to CloudWatch > Dashboards > Create Dashboard

Name it monitoringDashboard.



The screenshot shows the 'Create new dashboard' dialog box in the AWS CloudWatch console. The dialog has a title bar with a close button. Inside, there is a 'Dashboard name' label and a text input field containing 'MonitoringDashboard'. Below the input field, a note states: 'Valid characters in dashboard names include "0-9A-Za-z-_"'. At the bottom of the dialog are two buttons: 'Cancel' and 'Create dashboard'.

Custom dashboards Automatic dashboards

Custom Dashboards (0) [Info](#)

[Share dashboard](#) [Delete](#) [Create dashboard](#)

Valid characters in dashboard names include "0-9A-Za-z-_".

[Cancel](#) [Create dashboard](#)

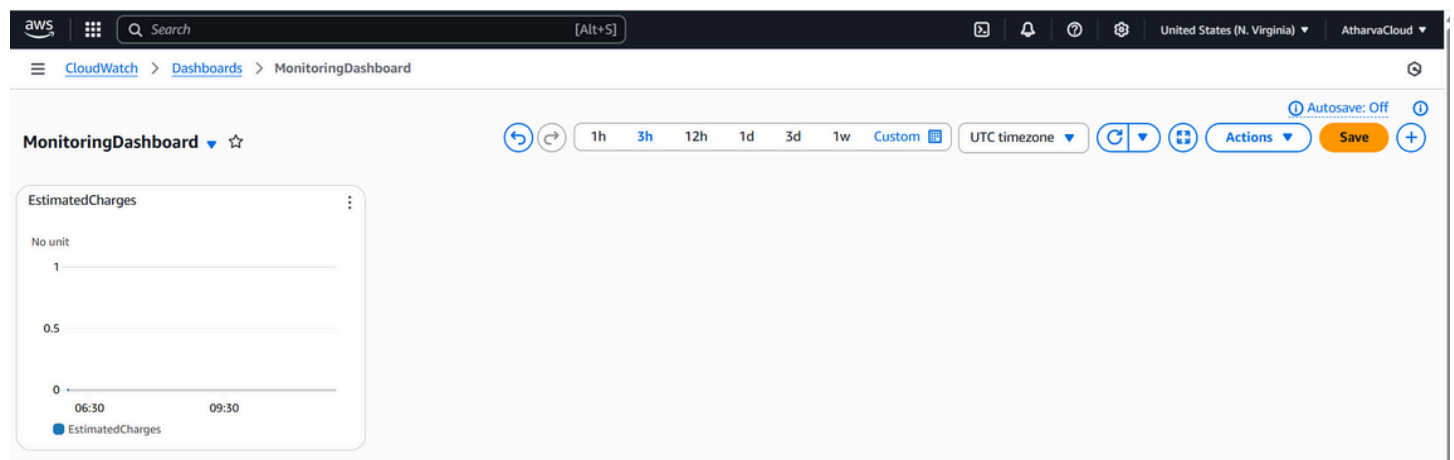
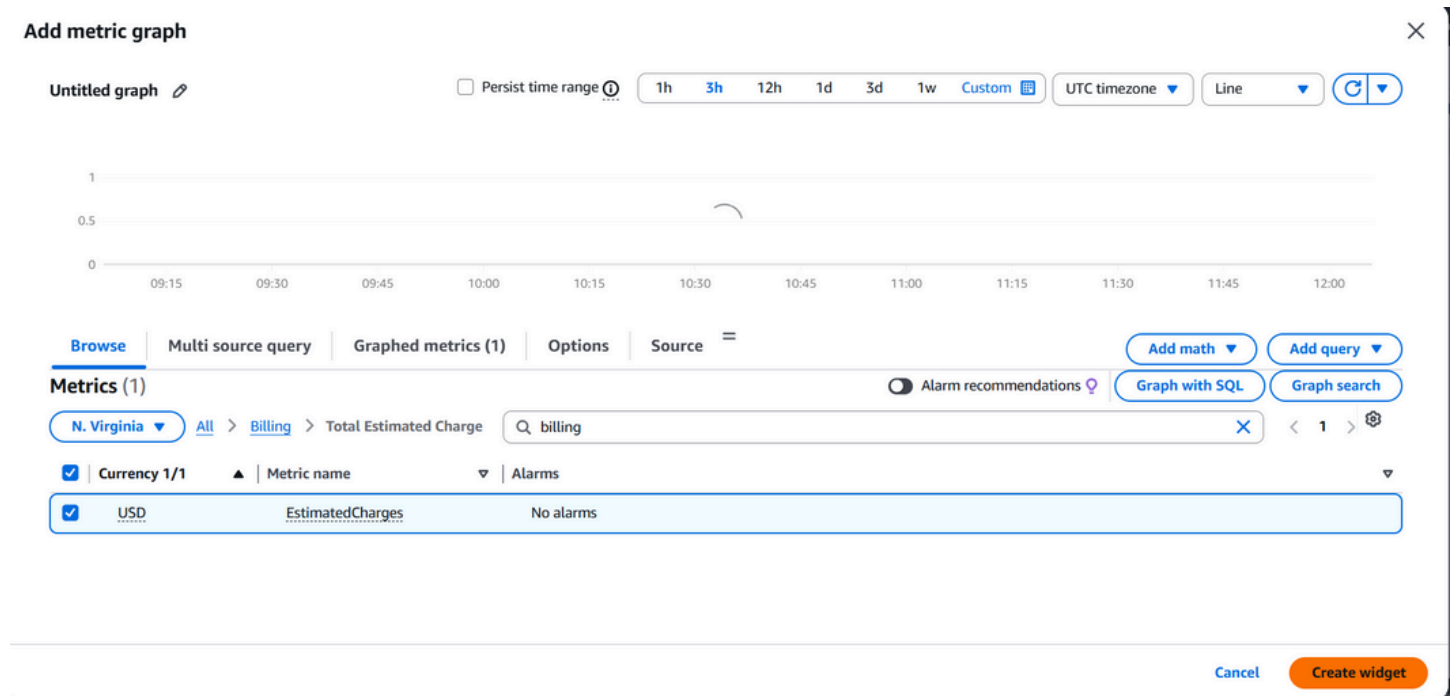
3.Add First Widget – Total Estimated Charges:

Namespace: AWS/Billing

Metric: EstimatedCharges

Visualization: Line Chart

purpose: The "Total Estimated Charges" widget helps you track the overall AWS spending in real time. It's crucial for cost visibility, especially if you're monitoring usage under a free-tier or have a specific monthly budget.



4) Add Second Widget – Charges by Service (Bar Chart)

Grouped by service


Helps identify top-cost contributors


pupose: This widget displays AWS charges grouped by individual services (e.g., EC2, S3, RDS), helping you quickly identify which services are contributing the most to your total bill. It's useful for pinpointing high-cost services and optimizing your AWS spending accordingly.


Add widget


Data sources types

- ☒ Cloudwatch
- ☐ Other content types
- ☐ Create data sources


See the latest value of a metric within a range 

☐ **Stacked area**
Compare the total over time 


☒ **Bar**
Compare categories of data 

☐ **Pie**
Show percentage or proportional data 

Add metric graph



Untitled graph 

☐ Persist time range ⓘ



1h 3h 12h 1d 3d 1w Custom 

UTC timezone ▼

Bar ▼

| Browse | Multi source query | Graphed metrics (5) | Options | Source | |
|-------------------------------------|--------------------|---------------------|------------------|-----------|--|
| <input type="checkbox"/> | AmazonCloudWatch | USD | EstimatedCharges | No alarms | |
| <input checked="" type="checkbox"/> | AmazonEC2 | USD | EstimatedCharges | No alarms | |
| <input checked="" type="checkbox"/> | AmazonECRPublic | USD | EstimatedCharges | No alarms | |
| <input checked="" type="checkbox"/> | AmazonRDS | USD | EstimatedCharges | No alarms | |
| <input checked="" type="checkbox"/> | AmazonS3 | USD | EstimatedCharges | No alarms | |
| <input type="checkbox"/> | AmazonVPC | USD | EstimatedCharges | No alarms | |
| <input type="checkbox"/> | AWSDataTransfer | USD | EstimatedCharges | No alarms | |
| <input checked="" type="checkbox"/> | AWSELB | USD | EstimatedCharges | No alarms | |
| <input type="checkbox"/> | AWSMarketplace | USD | EstimatedCharges | No alarms | |

 Add math ▼
  Add query ▼

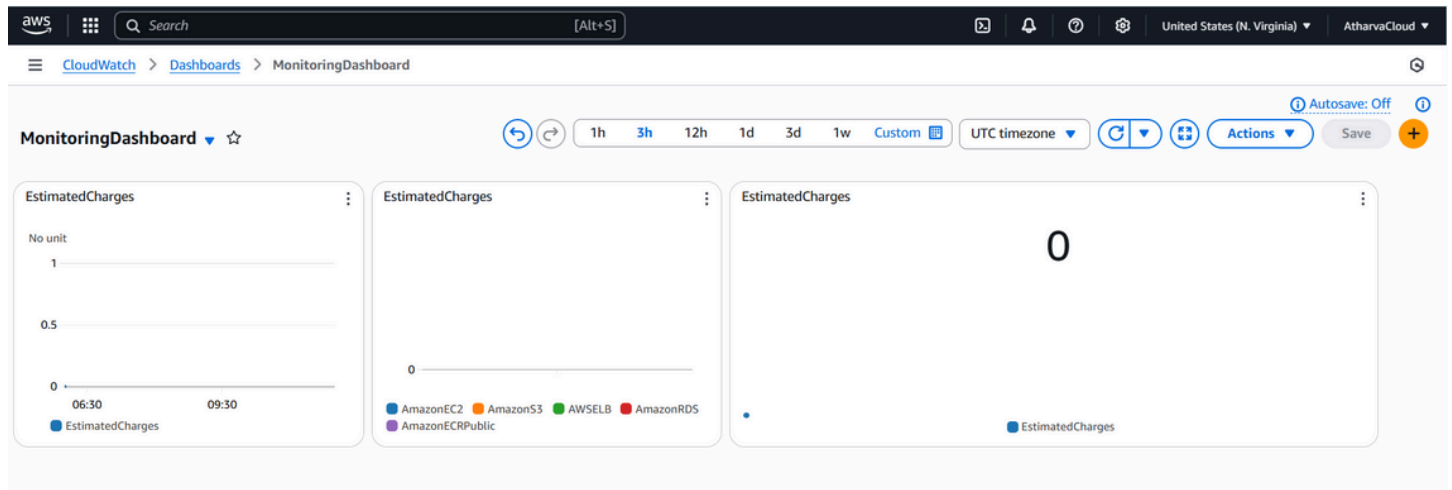
[Cancel](#)
[Create widget](#)

5)Add Third Widget – Number View for Charges

Clean numeric display for quick overview

Purpose: This widget provides a clear, real-time numeric display of your total estimated AWS charges.

It offers a quick-glance overview without needing to analyze charts or graphs.



Dashboard 2: Application & System Logs Monitoring:

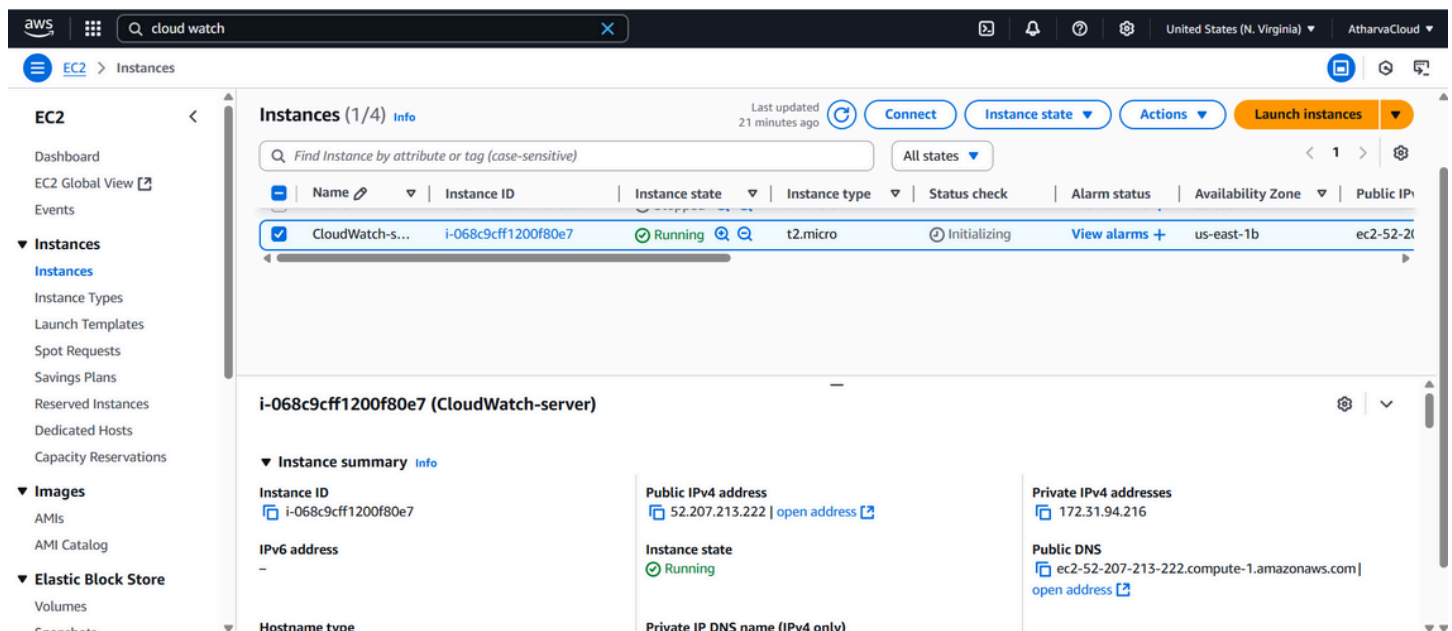
purpose:

This dashboard helps you view system logs like syslog and auth.log from your EC2 instance in real time.

It lets you easily monitor system activity and detect issues like failed login attempts using CloudWatch Logs and Insights.

Steps:

1)Launch Ubuntu EC2 instance:



2)Install CloudWatch Agent:

Run:

“sudo apt update

sudo apt install -y unzip

wget https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb

sudo dpkg -i amazon-cloudwatch-agent.deb”

Why?

- i) Installing the CloudWatch Agent is necessary to collect and send system-level metrics and log files (like syslog and auth.log) from your EC2 instance to Amazon CloudWatch.
- ii) Without the agent, CloudWatch cannot access logs stored locally on your EC2 instance — so it's essential for log monitoring and insights dashboards.

```
ubuntu@ip-172-31-94-216:~$ sudo dpkg -i -E ./amazon-cloudwatch-agent.deb
Selecting previously unselected package amazon-cloudwatch-agent.
(Reading database ... 70681 files and directories currently installed.)
Preparing to unpack ./amazon-cloudwatch-agent.deb ...
create group cwagent, result: 0
create user cwagent, result: 0
Unpacking amazon-cloudwatch-agent (1.300057.0b1161-1) ...
Setting up amazon-cloudwatch-agent (1.300057.0b1161-1) ...
```

```
ubuntu@ip-172-31-94-216:~$ sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-
cloudwatch-agent-config-wizard
=====
= Welcome to the Amazon CloudWatch Agent Configuration Manager =
=
= CloudWatch Agent allows you to collect metrics and logs from =
= your host and send them to CloudWatch. Additional CloudWatch =
= charges may apply.
=====
On which OS are you planning to use the agent?
1. linux
2. windows
3. darwin
default choice: [1]:

Trying to fetch the default region based on ec2 metadata...
I! imds retry client will retry 1 timesAre you using EC2 or On-Premises host
s?
1. EC2
2. On-Premises
default choice: [1]:

Which user are you planning to run the agent?
1. cwagent
2. root
3. others
default choice: [1]:

Do you want to turn on StatsD daemon?
1. yes
2. no
default choice: [1]:
```

```

lled or the Agent will fail to start
. yes
. no
default choice: [1]:

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

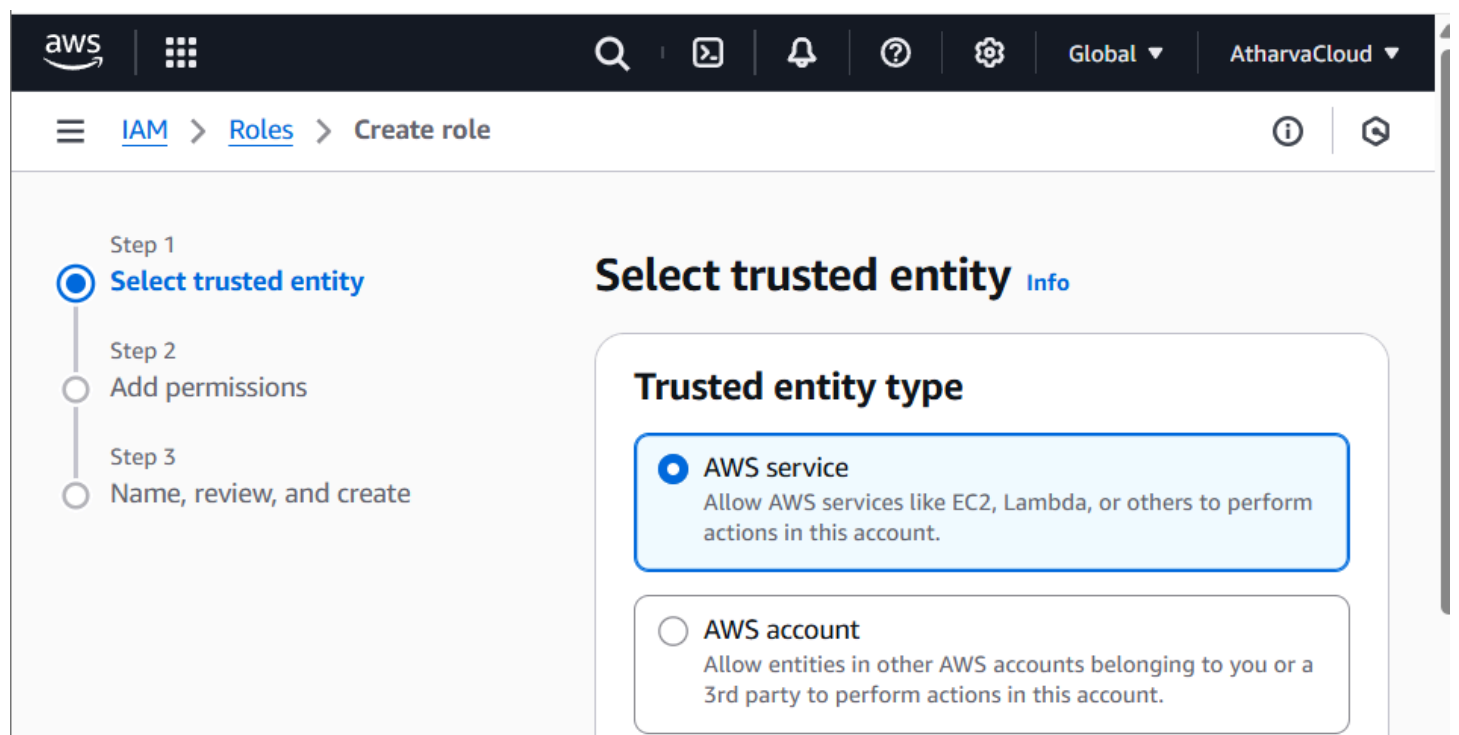
o you have any existing CloudWatch Log Agent (http://docs.aws.amazon.com/Am
zonCloudWatch/latest/logs/AgentReference.html) configuration file to import
for migration?
. yes
. no
default choice: [2]:

o you want to monitor any log files?
. yes

```

3) Create IAM Role for EC2 with CloudWatchAgentServerPolicy:

i) The CloudWatch Agent needs permission to push log data (e.g., syslog, auth.log) and metrics from the EC2 to CloudWatch:



ii) IAM Role for EC2 Instance (Logs Monitoring Dashboard)

Role Type: EC2 instance role

Policy Attached: CloudWatchAgentServerPolicy

Purpose: Allows the CloudWatch Agent on EC2 to push logs (e.g., syslog, auth.log) to CloudWatch Logs.

aws | IAM > Roles > Create role

Permissions policies (2/1067) Info

Choose one or more policies to attach to your new role.

Filter by Type

Search: AgentServerPolicy X All t... 1 match

< 1 > Settings

| <input checked="" type="checkbox"/> | Policy name | Type |
|-------------------------------------|--------------------|-------------|
| <input checked="" type="checkbox"/> | CloudWatchAgent... | AWS managed |

Permissions

You manage access in AWS by creating policies and attaching them to IAM identities (users, groups of users, or roles) or AWS resources. A policy is an object in AWS that, when associated with an identity or resource, defines their permissions.

Was this content helpful?

Yes No

iii) IAM Role for EC2 Management via SSM (Used Alongside CloudWatch Agent)

Role Type: EC2 instance role

Policy Attached: AmazonSSMManagedInstanceCore

Purpose: Enables AWS Systems Manager (SSM) to manage and interact with the EC2 instance (e.g., for Session Manager, patching, and sending/receiving parameters).

aws | IAM > Roles > Create role

Permissions policies (2/1067) Info

Choose one or more policies to attach to your new role.

Filter by Type

Search: agedInstanceCore X All t... 1 match

< 1 > Settings

| <input checked="" type="checkbox"/> | Policy name | Type |
|-------------------------------------|------------------|-------------|
| <input checked="" type="checkbox"/> | AmazonSSMMana... | AWS managed |

Permissions

You manage access in AWS by creating policies and attaching them to IAM identities (users, groups of users, or roles) or AWS resources. A policy is an object in AWS that, when associated with an identity or resource, defines their permissions.

Was this content helpful?

Yes No

iv) role is created

Identity and Access Management (IAM)

Search IAM

Dashboard

▼ **Access management**

- User groups
- Users
- Roles**
- Policies
- Identity providers

Role EC2CloudWatchAgentRole created. [View role](#)

Roles (28) [Info](#) [Refresh](#) [Delete](#) [Create role](#)

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

Search

< 1 2 > [Settings](#)

| <input type="checkbox"/> | Role name |
|--------------------------|--|
| <input type="checkbox"/> | EC2CloudWatchAgentRole |

v) Attached this role to EC2

EC2 > **Instances** > **i-068c9cff1200f80e7** > **Modify IAM role**

Modify IAM role [Info](#)

Attach an IAM role to your instance.

Instance ID
[i-068c9cff1200f80e7](#) (CloudWatch-server)

IAM role
Select an IAM role to attach to your instance or create a new role if you haven't created any. The role you select replaces any roles that are currently attached to your instance.

[EC2CloudWatchAgentRole](#) ▼

4) Generate Config File using wizard:

The config wizard helps you create a custom CloudWatch Agent configuration file to define what logs and metrics to collect.

In this case, it's used to specify the log file paths (/var/log/syslog, /var/log/auth.log) that should be sent to CloudWatch for monitoring.

```
{
  "agent": {
    "metrics_collection_interval": 60,
    "run_as_user": "root"
  },
  "logs": {
    "logs_collected": {
      "files": {
        "collect_list": [
          {
            "file_path": "/var/log/auth.log",
            "log_group_class": "STANDARD",
            "log_group_name": "auth-log-group",
            "log_stream_name": "{instance_id}",
            "retention_in_days": 30
          },
          {
            "file_path": "/var/log/syslog",
            "log_group_class": "STANDARD",
            "log_group_name": "syslog-group",
            "log_stream_name": "{instance_id}",
            "retention_in_days": 30
          }
        ]
      }
    }
  }
}
```

5)Start the Agent using below command:

```
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl \
-a fetch-config -m ec2 \
-c file:/opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.d/file_config.json \
-s
```

iii)log groups are created :

▼ Metrics

All metrics

Explorer

| | |
|--------------------------|--------------------------------|
| <input type="checkbox"/> | auth-log-group |
| <input type="checkbox"/> | syslog-group |

6)Use CloudWatch Logs Insights Queries:

i)Query1(syslog):

[CloudWatch](#) > [Logs Insights](#)

CloudWatch

Favorites and recents

Dashboards

▶ AI Operations [New](#)

▶ Alarms ⚠ 0 ✅ 0 ⋮ 0

▼ Logs

Log groups

Log Anomalies

Live Tail

Logs Insights

Contributor Insights

UTC timezone

Select log groups by

Log group name

Selection criteria

Select up to 50 log groups

☐ /ecs/mynewtaskdef
Standard

☐ syslog-group
Standard

[Query generator](#)

Run query

Cancel

Save

Discovered fields

Saved and sample queries

Query commands

ii) This CloudWatch Logs Insights query filters log messages containing the word "error", sorts them by timestamp in descending order (latest first), and shows the top 20 results. It displays two fields: the log time (@timestamp) and the message content (@message) for quick error analysis.

```

1 fields @timestamp, @message
2 | filter @message like /error/
3 | sort @timestamp desc
4 | limit 20

```

[Query generator](#)

Run query

Cancel

Save

History

Logs Insights QL query can run for maximum of 60 minutes.

✅ Completed. Query executed for 1 log group.

Query commands

iii) Query2(auth log):

Add to dashboard

Select a dashboard

Select an existing dashboard or create a new one.

"LogsDashboard"

MonitoringDashboard

Widget type

Select a widget type to add to the dashboard.

Logs table

Customize widget title

Widgets get an automatic title. You can optionally customize the title here.

Preview

This is how your chart will appear in your dashboard.

Log group: syslog-group

No data found.

Try adjusting the dashboard time range or log query.

Cancel

Add to dashboard

iv) This below query looks for log messages in the syslog-group that contain the phrase "Failed password" (usually failed login attempts). It then counts how many such events happened every 5 minutes, helping you spot unusual login activity over time.

aws

United States (N. Virginia)

AtharvaCloud

CloudWatch

Logs Insights

CloudWatch

Favorites and recents

Dashboards

LogsDashboard

AI Operations

Alarms

Logs

Log groups

Log Anomalies

Live Tail

Select log groups by

Log group name

Selection criteria

Select up to 50 log groups

syslog-group

Clear all

Browse log groups

```

1 fields @timestamp, @message
2 | filter @message like /Failed pas
3 | stats count() by bin(5m)
4

```

Query generator

Discovered fields

Saved and sample queries

Query commands

v) we can see the two widgets:

| <div> <div>aws</div> <div> <div></div> <div>Search</div> <div>[Alt+S]</div> </div> <div> <div></div> <div></div> <div></div> <div></div> <div></div> </div> <div>United States (N. Virginia) AtharvaCloud</div> </div> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|--------------------------|---|---|------------|----------|---|--------------------------|---|---|--------------------------|---|---|--------------------------|---|---|--------------------------|---|---|--------------------------|--|---|--------------------------|---|---|--------------------------|---|---|--------------------------|--|---|--------------------------|---|
| <div> <div>CloudWatch</div> <div>Dashboards</div> <div>LogsDashboard</div> </div> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <div> <div>LogsDashboard</div> <div> <div>1h</div> <div>3h</div> <div>12h</div> <div>1d</div> <div>3d</div> <div>1w</div> <div>Custom</div> </div> <div>UTC timezone</div> <div>15 minutes</div> <div>Autosave: Off</div> <div>Actions</div> <div>Save</div> </div> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <div> <div>Log group: auth-log-group</div> <table> <tr> <th>#</th><th>@timestamp</th><th>@message</th></tr> <tr> <td>1</td><td>2025-07-15T06:27:19.111Z</td><td>2025-07-15T06:27:14.319073+00:00 ip-172-31-94-216 sshd[10602]: Connection closed by 221.159.21.170 port 43010 [preauth]</td></tr> <tr> <td>2</td><td>2025-07-15T06:26:56.111Z</td><td>2025-07-15T06:26:51.738174+00:00 ip-172-31-94-216 sshd[10600]: Connection closed by 221.159.21.170 port 34260 [preauth]</td></tr> <tr> <td>3</td><td>2025-07-15T06:26:26.111Z</td><td>2025-07-15T06:26:21.302833+00:00 ip-172-31-94-216 sshd[10597]: Connection closed by 221.159.21.170 port 42944 [preauth]</td></tr> <tr> <td>4</td><td>2025-07-15T06:26:07.111Z</td><td>2025-07-15T06:26:02.330459+00:00 ip-172-31-94-216 sshd[10591]: Connection closed by 221.159.21.170 port 37948 [preauth]</td></tr> <tr> <td>5</td><td>2025-07-15T06:25:56.111Z</td><td>2025-07-15T06:25:50.889585+00:00 ip-172-31-94-216 sshd[10590]: Connection reset by 221.159.21.170 port 34416</td></tr> <tr> <td>6</td><td>2025-07-15T06:25:51.122Z</td><td>2025-07-15T06:25:48.718606+00:00 ip-172-31-94-216 sshd[10579]: Connection closed by 221.159.21.170 port 44958 [preauth]</td></tr> <tr> <td>7</td><td>2025-07-15T06:25:51.122Z</td><td>2025-07-15T06:25:50.889275+00:00 ip-172-31-94-216 sshd[10590]: error: kex_exchange_identification: read: Connection reset by peer</td></tr> <tr> <td>8</td><td>2025-07-15T06:25:48.111Z</td><td>2025-07-15T06:25:43.391567+00:00 ip-172-31-94-216 sshd[10589]: Connection reset by 221.159.21.170 port 58824</td></tr> <tr> <td>9</td><td>2025-07-15T06:25:43.605Z</td><td>2025-07-15T06:25:43.391411+00:00 ip-172-31-94-216 sshd[10589]: error: kex_exchange_identification: read: Connection reset by peer</td></tr> </table> </div> | | | # | @timestamp | @message | 1 | 2025-07-15T06:27:19.111Z | 2025-07-15T06:27:14.319073+00:00 ip-172-31-94-216 sshd[10602]: Connection closed by 221.159.21.170 port 43010 [preauth] | 2 | 2025-07-15T06:26:56.111Z | 2025-07-15T06:26:51.738174+00:00 ip-172-31-94-216 sshd[10600]: Connection closed by 221.159.21.170 port 34260 [preauth] | 3 | 2025-07-15T06:26:26.111Z | 2025-07-15T06:26:21.302833+00:00 ip-172-31-94-216 sshd[10597]: Connection closed by 221.159.21.170 port 42944 [preauth] | 4 | 2025-07-15T06:26:07.111Z | 2025-07-15T06:26:02.330459+00:00 ip-172-31-94-216 sshd[10591]: Connection closed by 221.159.21.170 port 37948 [preauth] | 5 | 2025-07-15T06:25:56.111Z | 2025-07-15T06:25:50.889585+00:00 ip-172-31-94-216 sshd[10590]: Connection reset by 221.159.21.170 port 34416 | 6 | 2025-07-15T06:25:51.122Z | 2025-07-15T06:25:48.718606+00:00 ip-172-31-94-216 sshd[10579]: Connection closed by 221.159.21.170 port 44958 [preauth] | 7 | 2025-07-15T06:25:51.122Z | 2025-07-15T06:25:50.889275+00:00 ip-172-31-94-216 sshd[10590]: error: kex_exchange_identification: read: Connection reset by peer | 8 | 2025-07-15T06:25:48.111Z | 2025-07-15T06:25:43.391567+00:00 ip-172-31-94-216 sshd[10589]: Connection reset by 221.159.21.170 port 58824 | 9 | 2025-07-15T06:25:43.605Z | 2025-07-15T06:25:43.391411+00:00 ip-172-31-94-216 sshd[10589]: error: kex_exchange_identification: read: Connection reset by peer |
| # | @timestamp | @message | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 2025-07-15T06:27:19.111Z | 2025-07-15T06:27:14.319073+00:00 ip-172-31-94-216 sshd[10602]: Connection closed by 221.159.21.170 port 43010 [preauth] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | 2025-07-15T06:26:56.111Z | 2025-07-15T06:26:51.738174+00:00 ip-172-31-94-216 sshd[10600]: Connection closed by 221.159.21.170 port 34260 [preauth] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | 2025-07-15T06:26:26.111Z | 2025-07-15T06:26:21.302833+00:00 ip-172-31-94-216 sshd[10597]: Connection closed by 221.159.21.170 port 42944 [preauth] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | 2025-07-15T06:26:07.111Z | 2025-07-15T06:26:02.330459+00:00 ip-172-31-94-216 sshd[10591]: Connection closed by 221.159.21.170 port 37948 [preauth] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | 2025-07-15T06:25:56.111Z | 2025-07-15T06:25:50.889585+00:00 ip-172-31-94-216 sshd[10590]: Connection reset by 221.159.21.170 port 34416 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6 | 2025-07-15T06:25:51.122Z | 2025-07-15T06:25:48.718606+00:00 ip-172-31-94-216 sshd[10579]: Connection closed by 221.159.21.170 port 44958 [preauth] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7 | 2025-07-15T06:25:51.122Z | 2025-07-15T06:25:50.889275+00:00 ip-172-31-94-216 sshd[10590]: error: kex_exchange_identification: read: Connection reset by peer | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 8 | 2025-07-15T06:25:48.111Z | 2025-07-15T06:25:43.391567+00:00 ip-172-31-94-216 sshd[10589]: Connection reset by 221.159.21.170 port 58824 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 9 | 2025-07-15T06:25:43.605Z | 2025-07-15T06:25:43.391411+00:00 ip-172-31-94-216 sshd[10589]: error: kex_exchange_identification: read: Connection reset by peer | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <div> <div>Log group: syslog-group</div> <table> <tr> <th>#</th><th>bin(1m)</th><th>logCount</th></tr> <tr> <td>1</td><td>2025-07-15T06:38:00.000Z</td><td>1</td></tr> <tr> <td>2</td><td>2025-07-15T06:37:00.000Z</td><td>1</td></tr> <tr> <td>3</td><td>2025-07-15T06:36:00.000Z</td><td>1</td></tr> <tr> <td>4</td><td>2025-07-15T06:35:00.000Z</td><td>2</td></tr> <tr> <td>5</td><td>2025-07-15T06:34:00.000Z</td><td>2</td></tr> <tr> <td>6</td><td>2025-07-15T06:33:00.000Z</td><td>2048</td></tr> </table> </div> | | | # | bin(1m) | logCount | 1 | 2025-07-15T06:38:00.000Z | 1 | 2 | 2025-07-15T06:37:00.000Z | 1 | 3 | 2025-07-15T06:36:00.000Z | 1 | 4 | 2025-07-15T06:35:00.000Z | 2 | 5 | 2025-07-15T06:34:00.000Z | 2 | 6 | 2025-07-15T06:33:00.000Z | 2048 | | | | | | | | | |
| # | bin(1m) | logCount | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 2025-07-15T06:38:00.000Z | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | 2025-07-15T06:37:00.000Z | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | 2025-07-15T06:36:00.000Z | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | 2025-07-15T06:35:00.000Z | 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | 2025-07-15T06:34:00.000Z | 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6 | 2025-07-15T06:33:00.000Z | 2048 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Dashboard 3: Network Performance Monitoring:

Purpose:

This dashboard helps you monitor network traffic and performance of your EC2 and ALB resources in real time.

It shows how much data is flowing, how fast targets are responding, and highlights any 4xx/5xx errors for quick troubleshooting.

Steps:

1)Create Dashboard → Add EC2 Network Metrics

aws

Search

United States (N. Virginia) AtharvaCloud

CloudWatch

Dashboards

NetworkPerformanceDashboard

NetworkPerform...

3h

1d

1w

3h

UTC timezone

15 minutes

Autosave: Off

Actions

Save

+

No widget on this dashboard.

+ Add a first widget

Documentation

2) Create Application Load Balancer (ALB)

- Create Target Group (add EC2 instance)
- Create ALB (enable port 80)
- Install NGINX on EC2 for real HTTP response

Ports for the selected instances
Ports for routing traffic to the selected instances.
80
1-65535 (separate multiple ports with commas)
Include as pending below
1 selection is now pending below. Include more or register targets when ready.

Review targets

Targets (1) Remove all pending

Filter targets Show only pending

| Instance ID | Name | Port | State | Security groups | Zone | Private IPv4 address | Subnet ID |
|---------------------|-------------------|------|---------|------------------|------------|----------------------|--------------------|
| i-068c9cff1200f80e7 | CloudWatch-server | 80 | Running | launch-wizard-28 | us-east-1b | 172.31.94.216 | subnet-0b47a059b8c |

1 pending

Cancel Previous Create target group

Load balancers (1) Actions Create load balancer

Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

Filter load balancers

| Name | DNS name | State | VPC ID | Availability Zones | Type | Date create |
|------------------|---------------------------|--------|-----------------------|----------------------|-------------|--------------|
| myALB-CloudWatch | myALB-CloudWatch-19803... | Active | vpc-0a9a3d9fa840b298a | 4 Availability Zones | application | July 15, 202 |

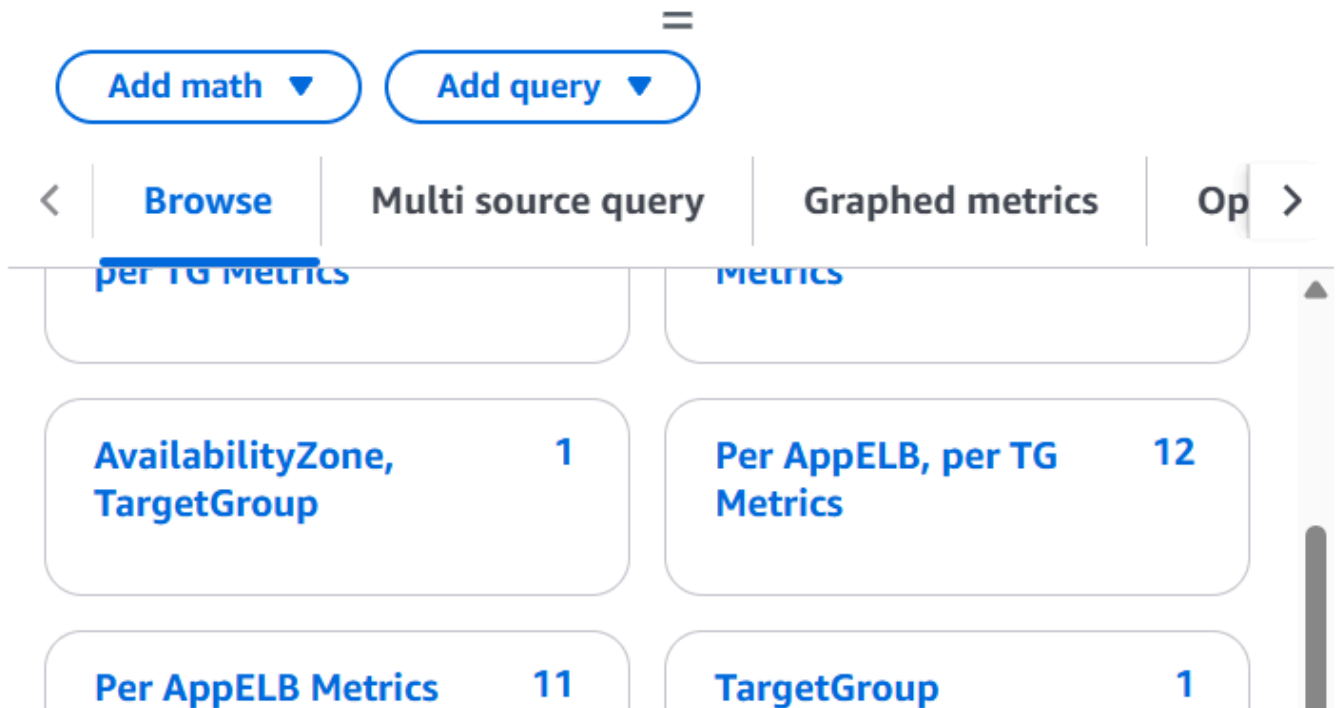
3) Why Metrics Weren't Initially Visible

ALB metrics appear only after valid HTTP traffic is processed.

Before NGINX, targets were unhealthy → no metrics

4) Add ALB Widgets

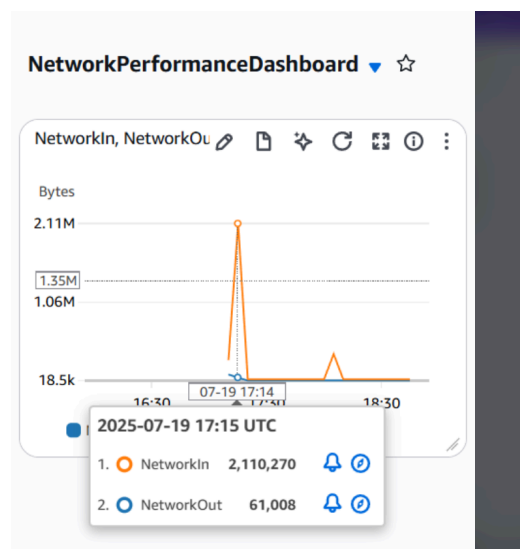
- RequestCount (Line Chart)
- TargetResponseTime (Line Chart)
- HTTPCode_Target_4XX_Count (Gauge)
- HTTPCode_Target_5XX_Count (Gauge)



i) Added 1st widget for NetworkIn and NetworkOut:

These metrics show the amount of incoming (NetworkIn) and outgoing (NetworkOut) data from your EC2 instance.

They help you monitor real-time traffic flow, detect abnormal spikes or drops, and ensure your instance is handling network load properly.



ii Added 2nd widgets for RequestCount And TargetResponseTime:

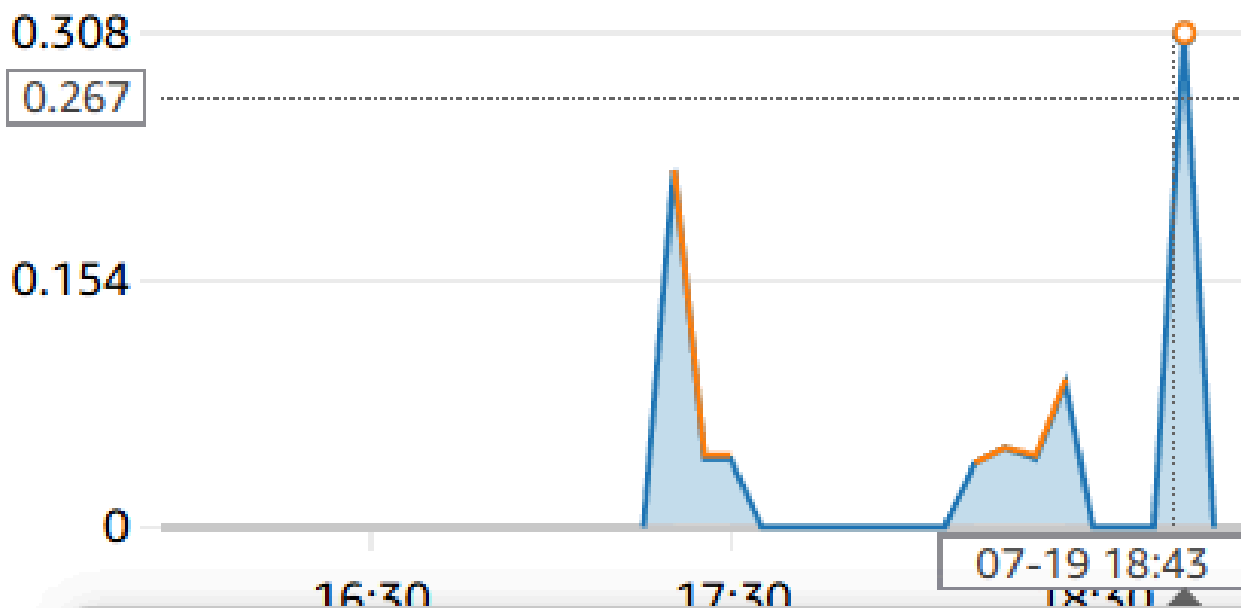
RequestCount shows how many requests are hitting your Load Balancer, helping you track traffic volume.

TargetResponseTime shows how long it takes for your backend (EC2) to respond, helping you monitor performance and detect slowdowns.

RequestCount, TargetR 



Various units



2025-07-19 18:45 UTC

1.  TargetResponseTime 0.00073575  
2.  RequestCount 0.30769230769  

Browse | Multi source query | Graphed metrics (2) | Options | Source = Add math ▼ Add query ▼

Metrics (11) Alarm recommendations Graph with SQL Graph search

N. Virginia ▼ **Per AppELB Metrics**

| <input type="checkbox"/> | LoadBalancer 11/11 | Metric name | Alarms |
|-------------------------------------|----------------------------------|---------------------------|-----------|
| <input type="checkbox"/> | app/myserviceLb/8c8f79c44e4dbc19 | HTTPCode_Target_2XX_Count | No alarms |
| <input checked="" type="checkbox"/> | app/myserviceLb/8c8f79c44e4dbc19 | TargetResponseTime | No alarms |

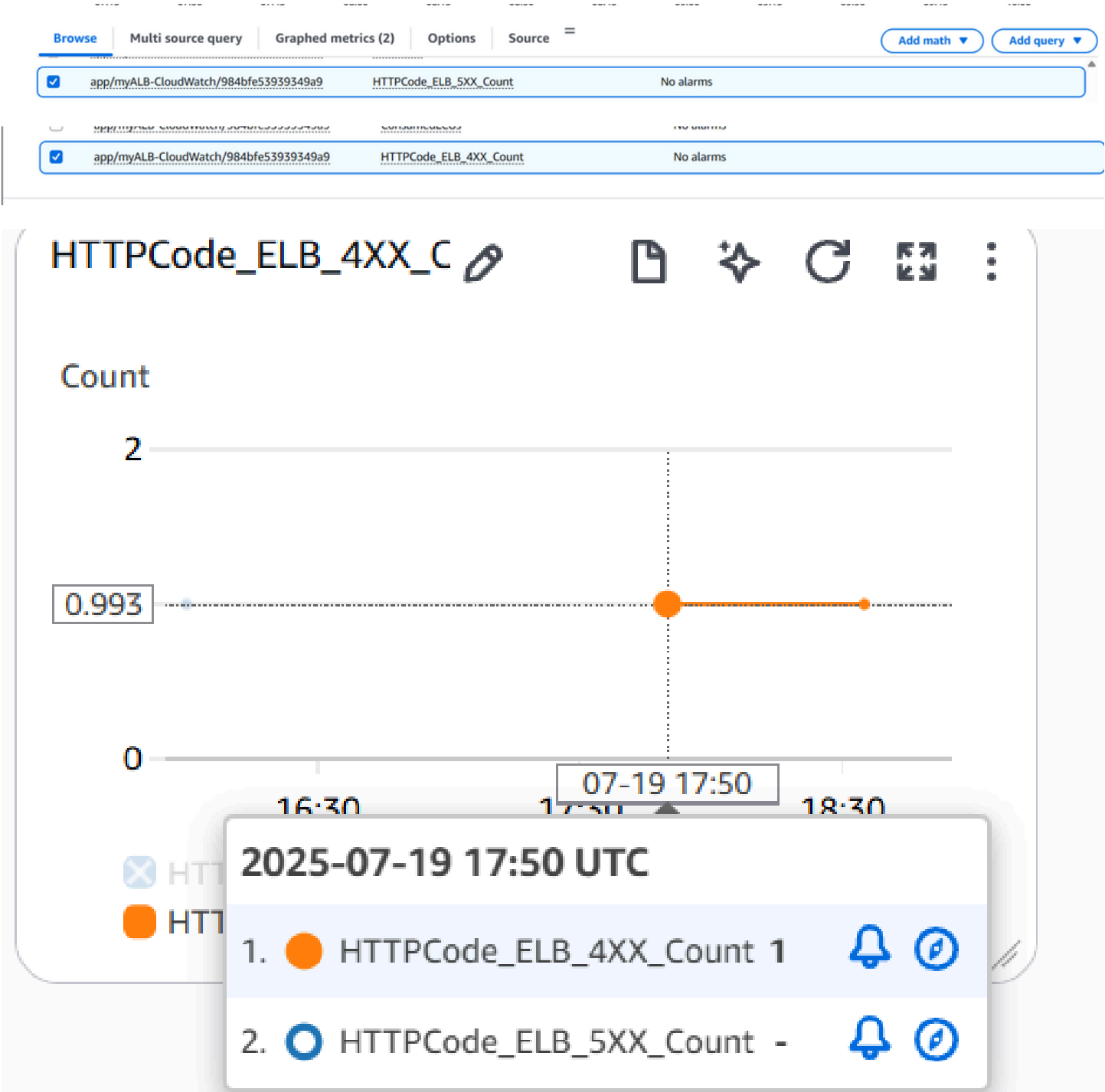
Browse | Multi source query | Graphed metrics (2) | Options | Source = Add math ▼ Add query ▼

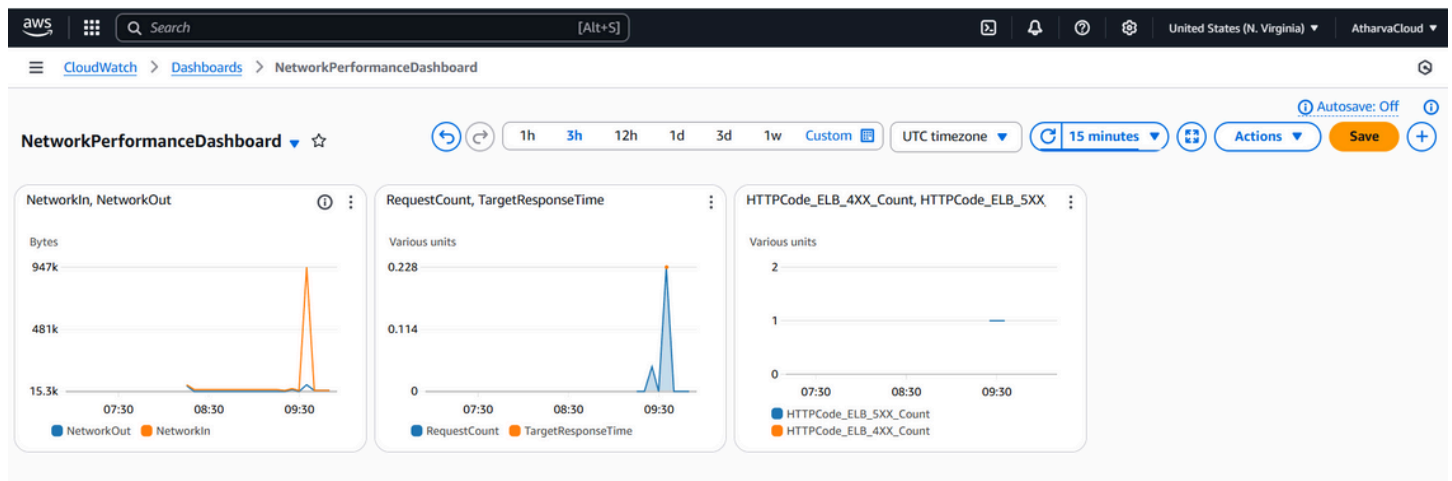
| | | | |
|-------------------------------------|----------------------------------|------------------------|-----------|
| <input type="checkbox"/> | app/myserviceLb/8c8f79c44e4dbc19 | HTTPCode_ELB_503_Count | No alarms |
| <input type="checkbox"/> | app/myserviceLb/8c8f79c44e4dbc19 | HTTPCode_ELB_5XX_Count | No alarms |
| <input type="checkbox"/> | app/myserviceLb/8c8f79c44e4dbc19 | HTTPCode_ELB_4XX_Count | No alarms |
| <input type="checkbox"/> | app/myserviceLb/8c8f79c44e4dbc19 | ConsumedLCUs | No alarms |
| <input type="checkbox"/> | app/myserviceLb/8c8f79c44e4dbc19 | PeakLCUs | No alarms |
| <input checked="" type="checkbox"/> | app/myserviceLb/8c8f79c44e4dbc19 | RequestCount | No alarms |

iii) Added 3rd Widgets for HTTPCode_Target_4XX_Count And HTTPCode_Target_5XX_Count:

These widgets show how many client-side errors (4XX) and server-side errors (5XX) are happening on your Application Load Balancer.

They help you quickly spot issues like bad requests from users or problems with your backend server (e.g., NGINX or app failures).





Dashboard 4: Security & Compliance Monitoring:

Purpose:

This dashboard helps you detect security threats and check compliance issues using logs from GuardDuty, AWS Config, IAM, and CloudTrail.

It shows alerts like unauthorized access, misconfigurations, failed logins, and API misuse — all visualized using custom log groups and queries, even in a free-tier setup.

Steps:

1)Enable GaurdDuty:

Service → Get Started → Enable

GuardDuty is a security service that automatically detects threats in your AWS account. Enabling it helps you monitor for suspicious activities like unauthorized access, malware, or unusual API calls — keeping your environment safe.

Amazon GuardDuty

Intelligent threat protection for accounts and workloads

Single-step threat detection

Designed to reduce security risk by using continuous intelligent threat detection capabilities for your AWS accounts, containers, workloads, and data.

Try threat detection with GuardDuty

☒ Amazon GuardDuty - all features

Experience threat detection capabilities in your AWS environment.

☐ GuardDuty Malware Protection for S3 only

Detect malicious files that are newly uploaded to your Amazon S3 buckets. You don't need to enable Amazon GuardDuty.

Get started

GuardDuty

[Enable GuardDuty](#) [Learn more](#)

- 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. Suspending or disabling GuardDuty doesn't impact Malware Protection for S3. To stop GuardDuty from scanning your S3 bucket for malware, you must delete the Malware Protection plan for each protected S3 bucket separately.

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 in a supported Region for the first time, your account gets automatically enrolled in a 30-day free trial. By default, some protection plans may also get included in a 30-day free trial. [Learn more](#)

Enable GuardDuty

2)Enable AWS Config:

i)Create S3 bucket

Add rules (e.g., cloudtrail-enabled, restricted-ssh, etc.)

Enabling AWS Config allows you to track changes and evaluate the security and compliance of your AWS resources over time.

The S3 bucket stores detailed configuration snapshots and rule evaluations.

The rules (like cloudtrail-enabled, restricted-ssh) help automatically detect non-compliant or insecure setups, so you can fix issues early.

AWS Config

Record and evaluate configurations of your AWS resources

AWS Config provides a detailed view of the resources associated with your AWS account, including how they are configured, how they are related to one another, and how the configurations and their relationships have changed over time.

Set up AWS Config

A summarized view of AWS and non-AWS resources and the compliance status of the rules and the resources in each AWS Region.

[Get started](#)[1-click setup](#)

Delivery channel

Amazon S3 bucket

☒ Create a bucket☐ Choose a bucket from your account☐ Choose a bucket from another account

Ensure appropriate permissions are available in this S3 bucket's policy. [Learn more](#).


S3 Bucket name (required)

Rules

AWS Managed Rules (154)

 Find Rules

< 1 2 3 4 5 6 7 ... 16 > 

|  | Name ▲ | Resource types | Trigger type | Description |
|---|----------------------------------|-----------------------|--------------|--|
| <input checked="" type="checkbox"/> | account-part-of-organization | | PERIODIC | Rule checks whether AWS rule is NON_COMPLIANT Organizations or AWS Or match rule parameter Ma |
| <input checked="" type="checkbox"/> | acm-certificate-expiration-check | AWS::ACM::Certificate | HYBRID | Checks whether ACM Cer expiration within the spe by ACM are automatically renew certificates that yo |
| | alb https | | | |

ii)Enabled recording in Aws config:

Enabling recording in AWS Config means it starts tracking and saving the configuration details of your AWS resources (like EC2, S3, IAM, etc.).

It records every time a resource is created, changed, or deleted, so you can see who changed what and when, which is useful for audits, troubleshooting, and compliance checks.

Review

Review your AWS Config setup details. You can go back to edit changes for each section. Choose **Confirm** to finish setting up AWS Config.

Recording method

Recording strategy

Record all resource types with customizable overrides

Default recording frequency

Continuous

► **Resource types with override settings (4)**

► **Resource types with default settings (435)**

3)Enable CloudTrail for API Monitoring:

i)Enabling CloudTrail lets you record all API calls and actions made in your AWS account (like starting EC2, changing IAM roles, etc.).

It helps you monitor user activity, detect suspicious behavior, and maintain security and compliance by keeping a full history of events.

Step 1

Choose trail attributes

Step 2

Choose log events

Step 3

Review and create

Choose trail attributes

General details

A trail created in the console is a multi-region trail. [Learn more](#)

Trail name

Enter a display name for your trail.

3-128 characters. Only letters, numbers, periods, underscores, and dashes are allowed.

☐ Enable for all accounts in my organization

To review accounts in your organization, open AWS Organizations. [See all accounts](#)

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.

Logs will be stored in config-bucket-2004atharva/AWSLogs/440744244333

Log file SSE-KMS encryption | [Info](#)

☒ Enabled

Customer managed AWS KMS key

☒ New

☐ Existing

AWS KMS alias

KMS key and S3 bucket must be in the same region.


Create a new SNS topic

- ☒ New
☐ Existing

SNS topic

aws-cloudtrail-logs-440744244333-9628471c

CloudWatch Logs - *optional*

Configure CloudWatch Logs to monitor your trail logs and notify you when specific activity occurs. Standard CloudWatch and CloudWatch Logs charges apply. [Learn more](#) 

CloudWatch Logs | [Info](#)

- ☒ Enabled

Log group [Info](#)

- ☒ New
☐ Existing

Log group name

IAM Role [Info](#)

AWS CloudTrail assumes this role to send CloudTrail events to your CloudWatch Logs log group.

- ☒ New
☐ Existing

Role name

CloudTrailRoleForCloudWatchLogs_{trail-name}

► Policy document

Tags - *optional* [Info](#)

You can add one or more tags to help you manage and organize your resources, including trails.

Key

project

Value - *optional*

SecurityDashboard

- Step 1
● Choose trail attributes
- Step 2
● **Choose log events**
- Step 3
○ Review and create

Choose log events

Events [Info](#)

Record API activity for individual resources, or for all current and future resources in AWS account. [Additional charges apply](#) [↗](#)

Event type

Choose the type of events that you want to log.

☒ **Management events**

Capture management operations performed on your AWS resources.

☒ **Data events**

Log the resource operations performed on or within a resource.

☐ **Insights events**

Identify unusual activity, errors, or user behavior in your account.

☐ **Network activity events**

Network activity events provide information about resource operations performed on a resource within a virtual private cloud endpoint.

Management events [Info](#)

Management events show information about management operations performed on resources in your AWS account.

ⓘ No additional charges apply to log management events on this trail because this is your first copy of management events.

API activity

Choose the activities you want to log.

☒ **Read**

☒ **Write**

☐ **Exclude AWS KMS events**

☐ **Exclude Amazon RDS Data API events**

over the data events captured by your trail.

Switch to basic event selectors

▼ Data event: S3

Remove

Resource type

Choose the resource type for which you want to log data events.

S3

Log selector template

Log all events

Selector name - optional

s3-DataEvents

1,000 character limit

► JSON view

● Choose trail attributes

Step 2

● Choose log events

Step 3

● Review and create

Review and create

Step 1: Choose trail attributes

Edit

General details

Trail name

SecurityAuditTrail

Multi-region trail

Yes

Apply trail to my organization

Not enabled

Trail log location

config-bucket-2004atharva/AWSLogs/440744244333

Log file SSE-KMS encryption

Not enabled

Log file validation

Enabled

SNS notification delivery

aws-cloudtrail-logs-440744244333-9628471c

ii)Manually Push Logs(simulated):

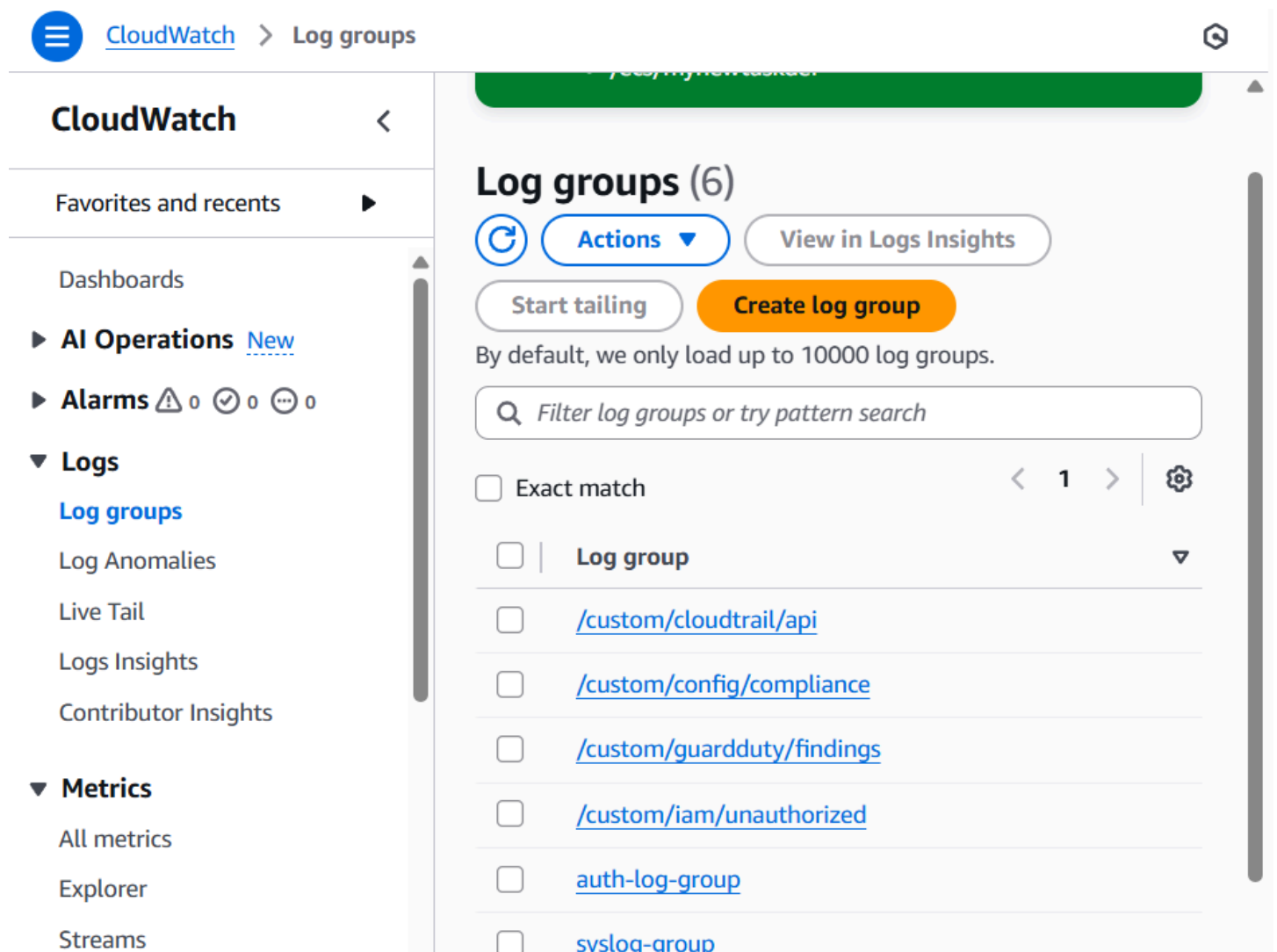
Example:

```
echo '{ "timestamp": "...", "type": "Trojan", "resource": "i-abc123" }' >> /var/log/custom-logs/guardduty.log
```

Repeat for config, IAM, and cloudtrail logs:

iii)Also Added the log groups path of these 4 new created log groups in the same config file:

iii)Final Log Agent Config Contains 6 Log Groups:



The screenshot shows the AWS CloudWatch console interface. On the left is a navigation sidebar with 'CloudWatch' at the top, followed by 'Favorites and recents', 'Dashboards', 'Al Operations' (with a 'New' link), 'Alarms' (with 0 warnings, 0 checks, and 0 errors), 'Logs' (expanded), 'Log groups' (selected), 'Log Anomalies', 'Live Tail', 'Logs Insights', 'Contributor Insights', 'Metrics' (expanded), 'All metrics', 'Explorer', and 'Streams'. The main content area is titled 'Log groups (6)' and includes buttons for 'Actions', 'View in Logs Insights', 'Start tailing', and 'Create log group'. A note states: 'By default, we only load up to 10000 log groups.' Below this is a search bar with the placeholder 'Filter log groups or try pattern search'. A filter section shows 'Exact match' is selected, with pagination controls showing '1' of 1 items. A list of log groups is displayed, each with a checkbox and a link:

| <input type="checkbox"/> | Log group |
|--------------------------|--|
| <input type="checkbox"/> | /custom/cloudtrail/api |
| <input type="checkbox"/> | /custom/config/compliance |
| <input type="checkbox"/> | /custom/guardduty/findings |
| <input type="checkbox"/> | /custom/iam/unauthorized |
| <input type="checkbox"/> | auth-log-group |
| <input type="checkbox"/> | syslog-group |

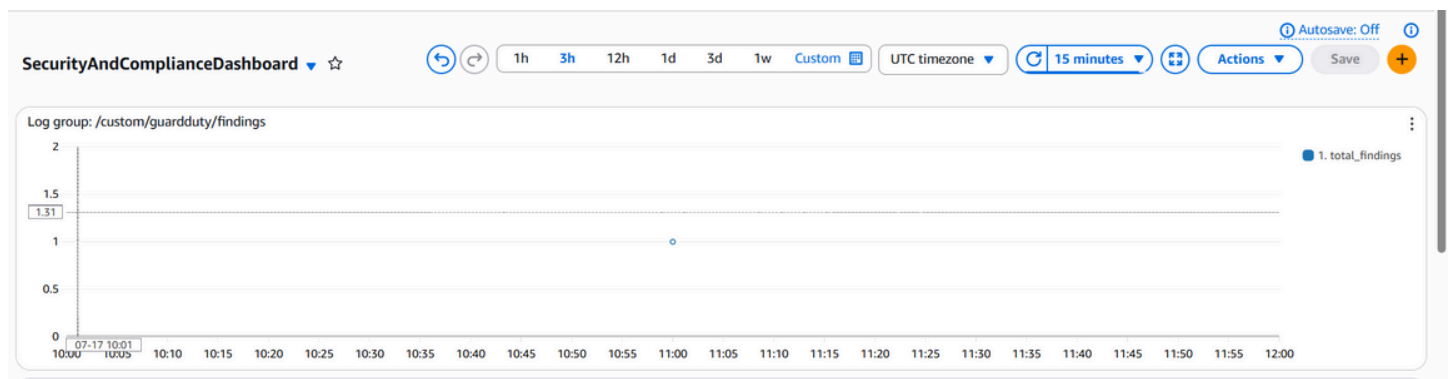
iv)fields @timestamp, type, resource, severity, status

| sort @timestamp desc

| limit 10

This query shows the latest 10 GuardDuty findings with details like time, type of threat, affected resource, severity, and status.

It helps you quickly identify and review recent security alerts in your AWS environment.

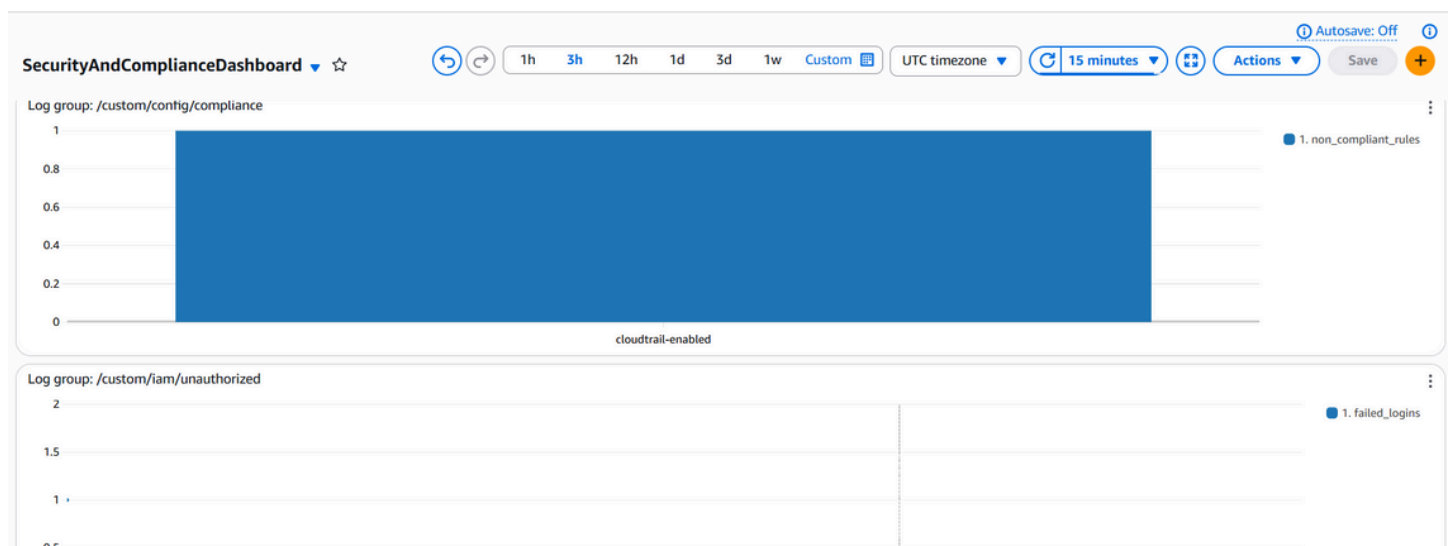


v)fields @timestamp, rule, compliance | stats count(*) by compliance

This query displays how many AWS Config rule evaluations are compliant or non-compliant. It shows the @timestamp, the rule name, and whether it passed or failed, then counts how many times each compliance status occurred. Useful for quickly checking the overall compliance status of your environment.

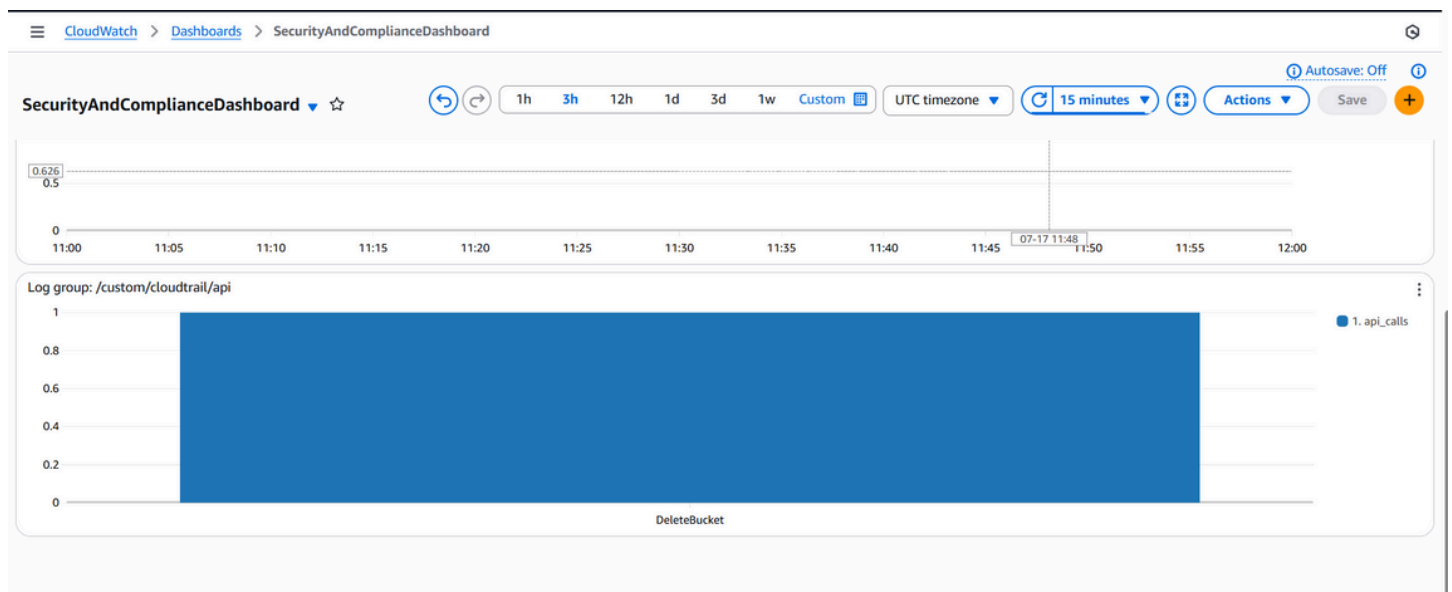
vi)fields @timestamp, user, action, status | filter status="Failed" | stats count(*) by user

This query filters IAM logs to show only failed actions (e.g., failed login attempts). It displays the user, action, and status, then counts how many failures happened per user. Helpful for identifying which users are experiencing or causing failed access attempts.



vii)fields @timestamp, api, user, result | stats count(*) by api, result

This query analyzes CloudTrail logs to count how many times each API call was made and whether it succeeded or failed. It groups the results by API name and result (like "Success" or "AccessDenied"). Useful for tracking frequently used APIs and spotting failed or suspicious operations.



Summary:

This project shows how to create four AWS CloudWatch dashboards to keep an eye on billing, logs, network traffic, and security. I used an EC2 instance with CloudWatch Agent to send system logs. I also created log groups and used simple queries to check failed logins, system messages, and API actions. For network monitoring, I set up a load balancer with NGINX to generate real traffic. I gave the EC2 instance the right IAM roles so it could send logs and be managed using SSM. The whole setup is low-cost and uses only free AWS tools — no Lambda or extra services were needed.