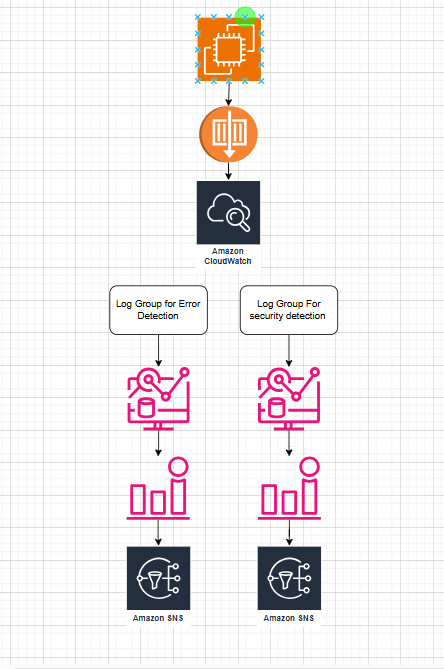
**Application Error Monitoring** and **Security Event Detection**

****

**Part 1: Application Error Monitoring**

Objective: Monitor application logs for errors and generate alerts.

**Step 1: Set Up Log Sources**

1. **Launch EC2 Instance**
   * Use Amazon Linux 2 or Ubuntu.
   * Install a web server (Apache/Nginx).

sudo yum install nginx -y

sudo systemctl start nginx.service

sudo systemctl enable nginx.service

* + Enable error logging:
    - Apache: /var/log/httpd/error\_log
    - Nginx: /var/log/nginx/error.log

**Step 2: Install CloudWatch Agent to Send Logs**

1. **Install the CloudWatch Agent**

# For Amazon Linux

sudo yum install amazon-cloudwatch-agent -y

# For Ubuntu

sudo apt install amazon-cloudwatch-agent -y

1. **Configure the agent**

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

* + Choose logs to monitor (/var/log/nginx/error.log or /var/log/httpd/error\_log).
  + Set log group name (e.g., App-Error-Logs).

1. **Start the CloudWatch Agent**

sudo systemctl start amazon-cloudwatch-agent

**Step 3: Create CloudWatch Log Group & Stream**

1. Go to AWS Console → CloudWatch → Logs.
2. Click **Create log group** (e.g., App-Error-Logs).
3. Click **Create log stream** under the log group.

**Step 4: Create CloudWatch Metric Filter for Errors**

1. Go to **CloudWatch Logs** → Log group (App-Error-Logs).
2. Click **Create metric filter**.
3. Enter a pattern to detect errors: ERROR
4. Click **Assign metric**, give it a name (e.g., AppErrorCount), and click **Create filter**.

**Step 5: Create a CloudWatch Alarm for Errors**

1. Go to **CloudWatch → Alarms → Create Alarm**.
2. Select **Metric** → AppErrorCount.
3. Set a threshold:
   * Example: Trigger an alarm if **error count exceeds 10 in 5 minutes**.
4. Choose an **SNS topic** to send email alerts.
   * If not created, go to SNS → Create a new topic → Subscribe with your email.
5. Click **Create Alarm**.

**🔹 Part 2: Security Event Detection (Failed SSH Login Attempts)**

**Objective: Detect unauthorized SSH access attempts.**

**Step 1: Enable Logging for SSH Authentication**

* On Amazon Linux, failed SSH login attempts are stored /var/log/secure

**Step 2: Create CloudWatch Agent Configuration for SSH Logs**

1. **Modify amazon-cloudwatch-agent.json to include SSH logs**:

{

"logs": {

"logs\_collected": {

"files": {

"collect\_list": [

{

"file\_path": "/var/log/auth.log",

"log\_group\_name": "SSH-Login-Attempts",

"log\_stream\_name": "{instance\_id}"

}

]

}

}

}

}

1. **Apply Configuration**:

sudo amazon-cloudwatch-agent-ctl -a fetch-config -m ec2 -c file:/opt/aws/amazon-cloudwatch-agent.json -s

1. **Restart CloudWatch Agent**:

sudo systemctl restart amazon-cloudwatch-agent

**Step 3: Create CloudWatch Metric Filter for Failed SSH Attempts**

1. Go to **CloudWatch Logs → Log Groups → SSH-Login-Attempts**.
2. Click **Create metric filter**.
3. Enter the following filter pattern:
4. "Failed password for"
5. Assign a metric name (e.g., FailedSSHAttempts).
6. Click **Create filter**.

**Step 4: Create CloudWatch Alarm for Unauthorized Access**

1. Go to **CloudWatch → Alarms → Create Alarm**.
2. Select **Metric** → FailedSSHAttempts.
3. Set a threshold:
   * Example: Trigger an alarm if **more than 5 failed attempts occur within 10 minutes**.
4. Choose an **SNS topic** for notifications.
5. Click **Create Alarm**.

**🔹 Summary**

✅ **Application Error Monitoring**

* Collects logs from /var/log/nginx/error.log or /var/log/httpd/error\_log.
* Creates a CloudWatch metric filter and alarm for errors.
* Sends alerts when errors exceed a threshold.

✅ **Security Event Detection (SSH Monitoring)**

* Collects logs from /var/log/auth.log (Ubuntu).
* Filters failed SSH login attempts.
* Triggers an alarm if **more than 5 failed login attempts occur in 10 minutes**.

Architecture 🡪

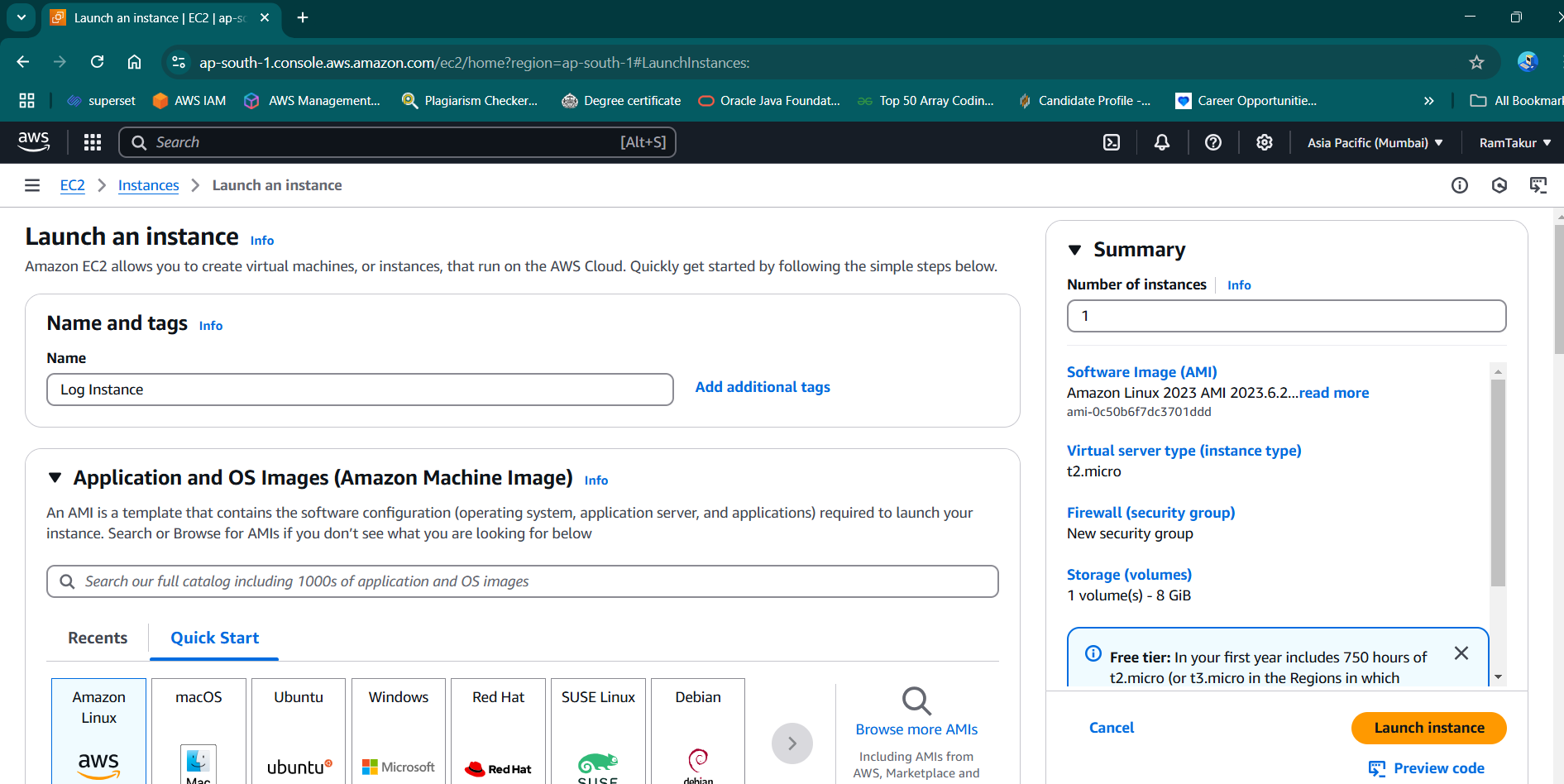
EC2 instance sending logs to CloudWatch

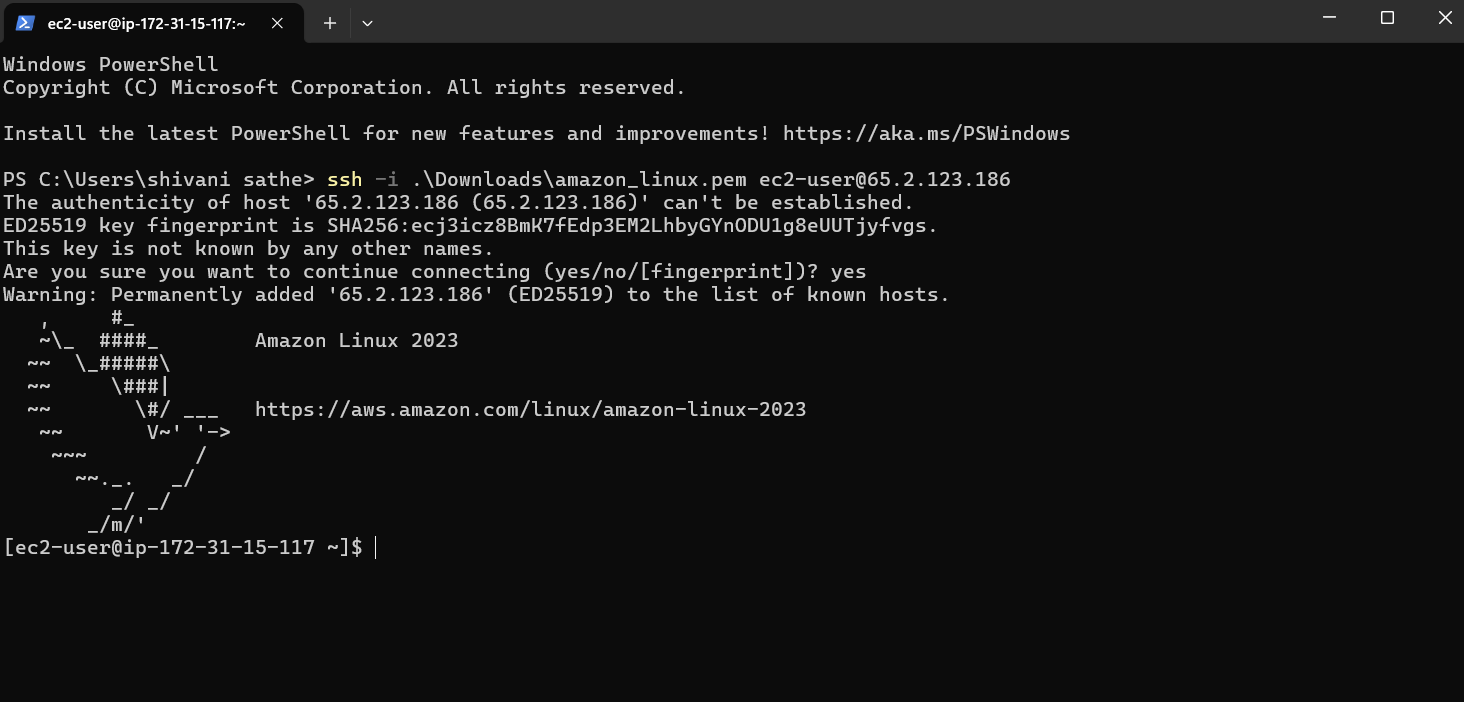
Separate log groups for **application errors** and **SSH login attempts**

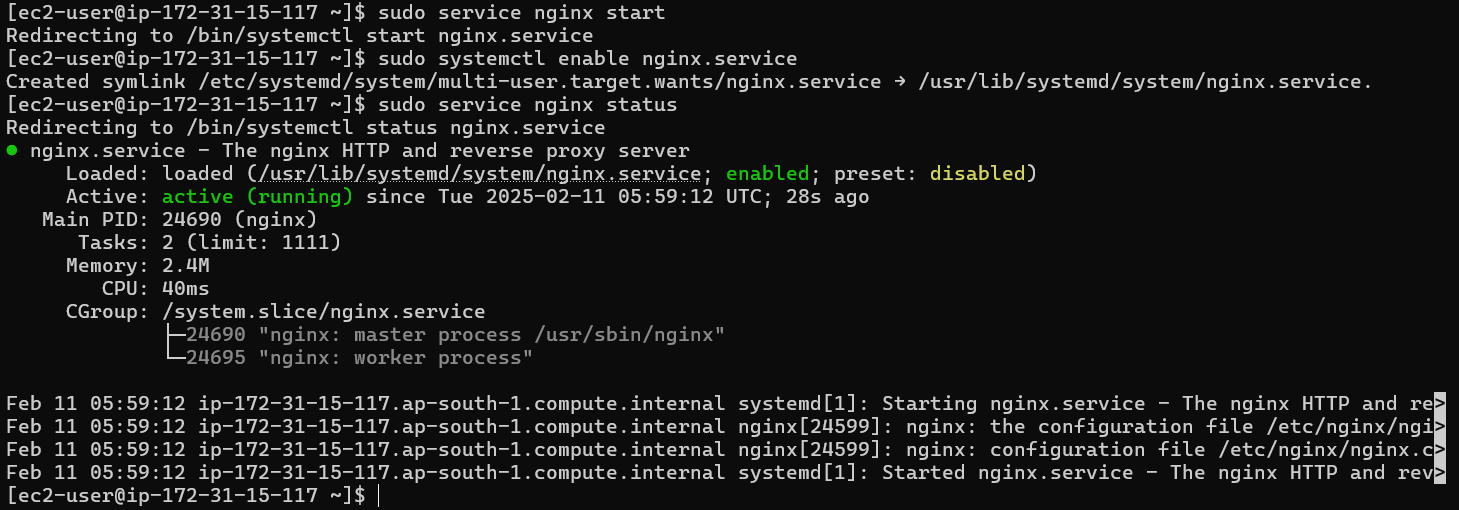
CloudWatch metric filters analyzing logs

CloudWatch alarms triggering based on thresholds

SNS notifications for alerts







[ec2-user@ip-172-31-15-117 ~]$ 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]:

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 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]:

2

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]:

1

Do you want to monitor cpu metrics per core?

1. yes

2. no

default choice: [1]:

2

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]:

4

Which default metrics config do you want?

1. Basic

2. Standard

3. Advanced

4. None

default choice: [1]:

1

Current config as follows:

{

"agent": {

"metrics\_collection\_interval": 60,

"run\_as\_user": "root"

},

"metrics": {

"aggregation\_dimensions": [

[

"InstanceId"

]

],

"append\_dimensions": {

"AutoScalingGroupName": "${aws:AutoScalingGroupName}",

"ImageId": "${aws:ImageId}",

"InstanceId": "${aws:InstanceId}",

"InstanceType": "${aws:InstanceType}"

},

"metrics\_collected": {

"disk": {

"measurement": [

"used\_percent"

],

"metrics\_collection\_interval": 60,

"resources": [

"\*"

]

},

"mem": {

"measurement": [

"mem\_used\_percent"

],

"metrics\_collection\_interval": 60

}

}

}

}

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]:

yes

The value yes is not valid to this question.

Please retry to answer:

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/nginx/error.log

Log group name:

default choice: [error.log]

App-Error-Logs

Log group class:

1. STANDARD

2. INFREQUENT\_ACCESS

default choice: [1]:

1

Log stream name:

default choice: [{instance\_id}]

Log Group Retention in days

1. -1

2. 1

3. 3

4. 5

5. 7

6. 14

7. 30

8. 60

9. 90

10. 120

11. 150

12. 180

13. 365

14. 400

15. 545

16. 731

17. 1096

18. 1827

19. 2192

20. 2557

21. 2922

22. 3288

23. 3653

default choice: [1]:

2

Do you want to specify any additional log files to monitor?

1. yes

2. no

default choice: [1]:

1

Log file path:

/var/log/secure

Log group name:

default choice: [secure]

SSH-Login-Attempts

Log group class:

1. STANDARD

2. INFREQUENT\_ACCESS

default choice: [1]:

1

Log stream name:

default choice: [{instance\_id}]

Log Group Retention in days

1. -1

2. 1

3. 3

4. 5

5. 7

6. 14

7. 30

8. 60

9. 90

10. 120

11. 150

12. 180

13. 365

14. 400

15. 545

16. 731

17. 1096

18. 1827

19. 2192

20. 2557

21. 2922

22. 3288

23. 3653

default choice: [1]:

2

Do you want to specify any additional log files to monitor?

1. yes

2. no

default choice: [1]:

2

Do you want the CloudWatch agent to also retrieve X-ray traces?

1. yes

2. no

default choice: [1]:

2

Existing config JSON identified and copied to: /opt/aws/amazon-cloudwatch-agent/etc/backup-configs

Saved config file to /opt/aws/amazon-cloudwatch-agent/bin/config.json successfully.

Current config as follows:

{

"agent": {

"metrics\_collection\_interval": 60,

"run\_as\_user": "root"

},

"logs": {

"logs\_collected": {

"files": {

"collect\_list": [

{

"file\_path": "/var/log/nginx/error.log",

"log\_group\_class": "STANDARD",

"log\_group\_name": "App-Error-Logs",

"log\_stream\_name": "{instance\_id}",

"retention\_in\_days": 1

},

{

"file\_path": "/var/log/secure",

"log\_group\_class": "STANDARD",

"log\_group\_name": "SSH-Login-Attempts",

"log\_stream\_name": "{instance\_id}",

"retention\_in\_days": 1

}

]

}

}

},

"metrics": {

"aggregation\_dimensions": [

[

"InstanceId"

]

],

"append\_dimensions": {

"AutoScalingGroupName": "${aws:AutoScalingGroupName}",

"ImageId": "${aws:ImageId}",

"InstanceId": "${aws:InstanceId}",

"InstanceType": "${aws:InstanceType}"

},

"metrics\_collected": {

"disk": {

"measurement": [

"used\_percent"

],

"metrics\_collection\_interval": 60,

"resources": [

"\*"

]

},

"mem": {

"measurement": [

"mem\_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-172-31-15-117 ~]$

