

AWS CLOUD WATCH

Lets understand the applications using a small example:

1.Create an EC2 Instance

Launch an instance [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags [Info](#)

Name

[Add additional tags](#)

▼ Application and OS Images (Amazon Machine Image) [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Quick Start

 aws	 Mac	 ubuntu®	 Microsoft	 Red Hat	 SUSE	 Browse more AMIs Including AMIs from AWS, Marketplace and the Community
---	--	--	--	--	---	---

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type
ami-075449515af5df0d1 (64-bit (x86)) / ami-07a66cb30628a9eaa (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible ▼

Description

Ubuntu Server 24.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

2. Login to instance through terminal using ssh command.

```
C:\Users\ORANGE\Downloads>ssh -i aws-login.pem ubuntu@13.60.220.8
The authenticity of host '13.60.220.8 (13.60.220.8)' can't be established.
ECDSA key fingerprint is SHA256:301CzeZltxrriZKPyCLufnpzh2/uzaw4fx9HbvTTkY.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '13.60.220.8' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1018-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Thu Dec 26 05:30:09 UTC 2024

System load:  0.0           Temperature:  -273.1 C
Usage of /:   24.7% of 6.71GB Processes:      107
Memory usage: 24%          Users logged in: 0
Swap usage:   0%           IPv4 address for ens5: 172.31.26.50

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

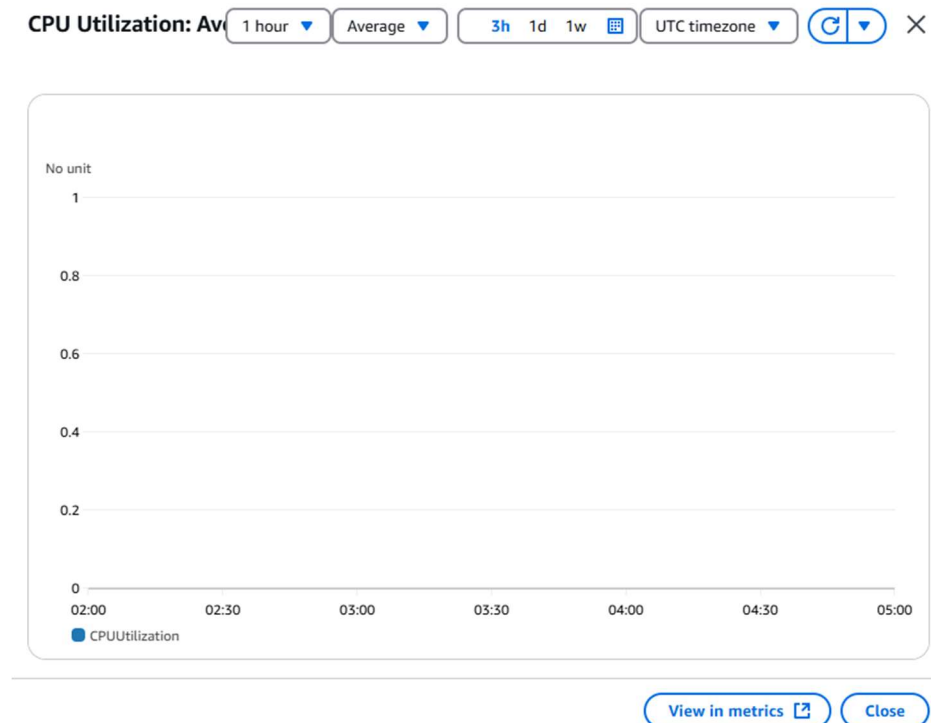
ubuntu@ip-172-31-26-50:~$
```

3. Check CPU utilization using top command

```
ubuntu@ip-172-31-26-50:~$ top
top - 05:31:31 up 16 min, 1 user, load average: 0.02, 0.01, 0.00
tasks: 107 total, 1 running, 106 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.2 us, 0.0 sy, 0.0 ni, 99.8 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 914.2 total, 419.3 free, 360.5 used, 287.1 buff/cache
MiB Swap: 0.0 total, 0.0 free, 0.0 used, 553.6 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM    TIME+  COMMAND
 1214 ubuntu    20   0   12396    5760   3584 R   0.3   0.6   0:00.01 top
   1 root      20   0   22480   13416   9448 S   0.0   1.4   0:01.78 systemd
   2 root      20   0         0         0     0 S   0.0   0.0   0:00.00 kthreadd
   3 root      20   0         0         0     0 S   0.0   0.0   0:00.00 pool_workqueue_release
   4 root      0 -20         0         0     0 I   0.0   0.0   0:00.00 kworker/R-rcu_p
   5 root      0 -20         0         0     0 I   0.0   0.0   0:00.00 kworker/R-rcu_p
   6 root      0 -20         0         0     0 I   0.0   0.0   0:00.00 kworker/R-slub_
   7 root      0 -20         0         0     0 I   0.0   0.0   0:00.00 kworker/R-netns
   9 root      0 -20         0         0     0 I   0.0   0.0   0:00.00 kworker/0:0H-events_highpri
  11 root     20   0         0         0     0 I   0.0   0.0   0:00.07 kworker/u4:0-events_power_efficient
  12 root      0 -20         0         0     0 I   0.0   0.0   0:00.00 kworker/R-mm_pg
  13 root     20   0         0         0     0 I   0.0   0.0   0:00.00 rcu_tasks_rude_kthread
  14 root     20   0         0         0     0 I   0.0   0.0   0:00.00 rcu_tasks_trace_kthread
  15 root     20   0         0         0     0 S   0.0   0.0   0:00.01 ksoftirqd/0
  16 root     20   0         0         0     0 I   0.0   0.0   0:00.05 rcu_sched
  17 root      rt   0         0         0     0 S   0.0   0.0   0:00.00 migration/0
  18 root    -51   0         0         0     0 S   0.0   0.0   0:00.00 idle_inject/0
  19 root     20   0         0         0     0 S   0.0   0.0   0:00.00 cpuhp/0
  20 root     20   0         0         0     0 S   0.0   0.0   0:00.00 cpuhp/1
  21 root    -51   0         0         0     0 S   0.0   0.0   0:00.00 idle_inject/1
  22 root      rt   0         0         0     0 S   0.0   0.0   0:00.06 migration/1
  23 root     20   0         0         0     0 S   0.0   0.0   0:00.02 ksoftirqd/1
  25 root     0 -20         0         0     0 I   0.0   0.0   0:00.00 kworker/1:0H-events_highpri
  26 root     20   0         0         0     0 S   0.0   0.0   0:00.00 kdevtmpfs
  27 root     0 -20         0         0     0 I   0.0   0.0   0:00.00 kworker/R-inet_
  28 root     20   0         0         0     0 I   0.0   0.0   0:00.13 kworker/u4:1-events_power_efficient
  29 root     20   0         0         0     0 S   0.0   0.0   0:00.00 kauditd
  30 root     20   0         0         0     0 I   0.0   0.0   0:00.19 kworker/1:1-cgroup_destroy
  31 root     20   0         0         0     0 S   0.0   0.0   0:00.00 khungtaskd
  32 root     20   0         0         0     0 S   0.0   0.0   0:00.00 oom_reaper
  34 root     0 -20         0         0     0 I   0.0   0.0   0:00.00 kworker/R-write
  35 root     20   0         0         0     0 S   0.0   0.0   0:00.03 kcompactd0
  36 root     25   5         0         0     0 S   0.0   0.0   0:00.00 ksmd
  37 root     39  19         0         0     0 S   0.0   0.0   0:00.00 khugepaged
  38 root     0 -20         0         0     0 I   0.0   0.0   0:00.00 kworker/R-kinte
  39 root     0 -20         0         0     0 I   0.0   0.0   0:00.00 kworker/R-kbloc
  40 root     0 -20         0         0     0 I   0.0   0.0   0:00.00 kworker/R-blkcq
  41 root    -51   0         0         0     0 S   0.0   0.0   0:00.00 irq/9-acpi
  42 root     0 -20         0         0     0 I   0.0   0.0   0:00.00 kworker/R-tpm_d
  43 root     0 -20         0         0     0 I   0.0   0.0   0:00.00 kworker/R-ata_s
  44 root     0 -20         0         0     0 I   0.0   0.0   0:00.00 kworker/R-md
  45 root     0 -20         0         0     0 I   0.0   0.0   0:00.00 kworker/R-md_b1
  46 root     0 -20         0         0     0 I   0.0   0.0   0:00.00 kworker/R-edac-
  47 root     0 -20         0         0     0 I   0.0   0.0   0:00.00 kworker/R-devfr
  48 root    -51   0         0         0     0 S   0.0   0.0   0:00.00 watchdogd
  49 root     0 -20         0         0     0 I   0.0   0.0   0:00.00 kworker/1:1H-kblockd
  50 root     20   0         0         0     0 S   0.0   0.0   0:00.00 kswapd0
  51 root     20   0         0         0     0 S   0.0   0.0   0:00.00 ecryptfs-kthread
```

4. CPU utilization in AWS Cloudwatch



5. Use a custom python script for CPU Simulation

```
import time

def simulate_cpu_spike(duration=30, cpu_percent=80):
    print(f'Simulating CPU spike at {cpu_percent}%...')
    start_time = time.time()

    # Calculate the number of iterations needed to achieve the desired CPU utilization
    target_percent = cpu_percent / 100
    total_iterations = int(target_percent * 5_000_000) # Adjust the number as needed

    # Perform simple arithmetic operations to spike CPU utilization
    for _ in range(total_iterations):
        result = 0
        for i in range(1, 1001):
            result += i

    # Wait for the rest of the time interval
    elapsed_time = time.time() - start_time
    remaining_time = max(0, duration - elapsed_time)
    time.sleep(remaining_time)

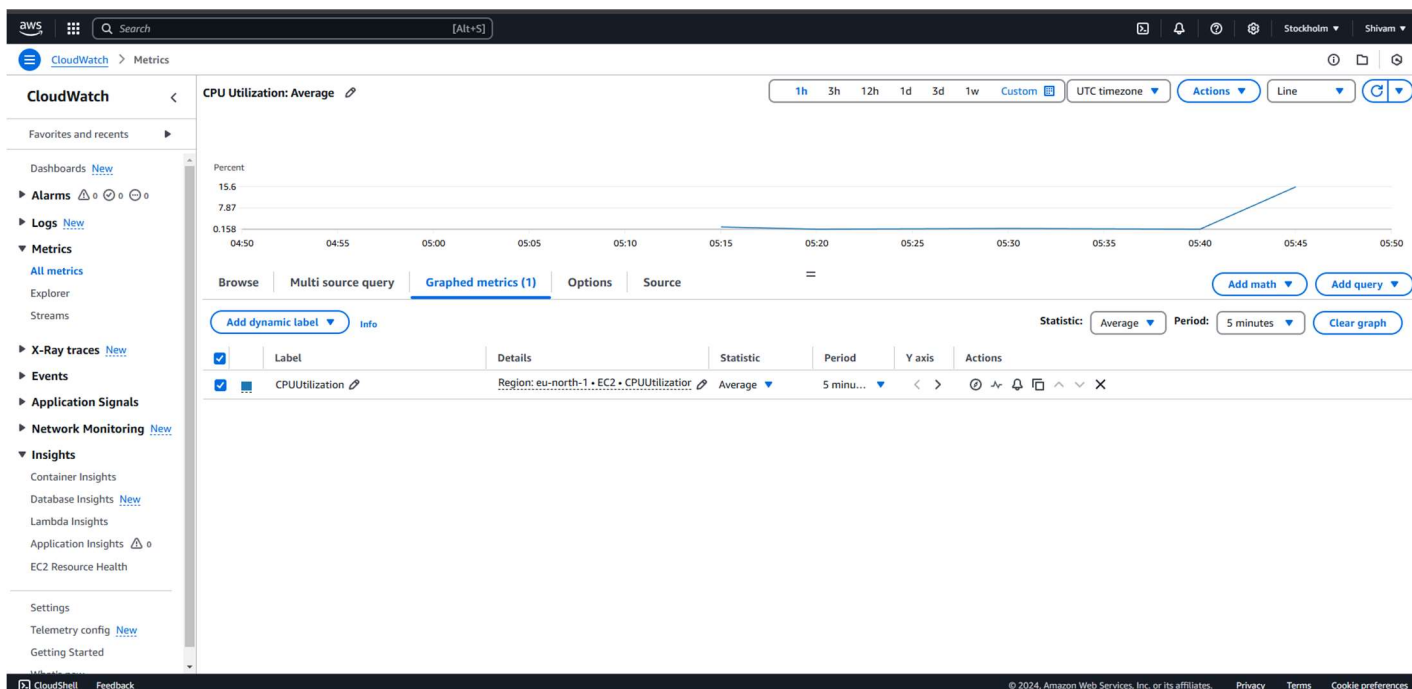
    print("CPU spike simulation completed.")

if __name__ == '__main__':
    # Simulate a CPU spike for 30 seconds with 80% CPU utilization
    simulate_cpu_spike(duration=30, cpu_percent=80)
```

6. Run the Script to trigger CPU Simulation

```
ubuntu@ip-172-31-26-50: ~  
ubuntu@ip-172-31-26-50:~$ python3 cpu_spike.py  
Simulating CPU spike at 80%...
```

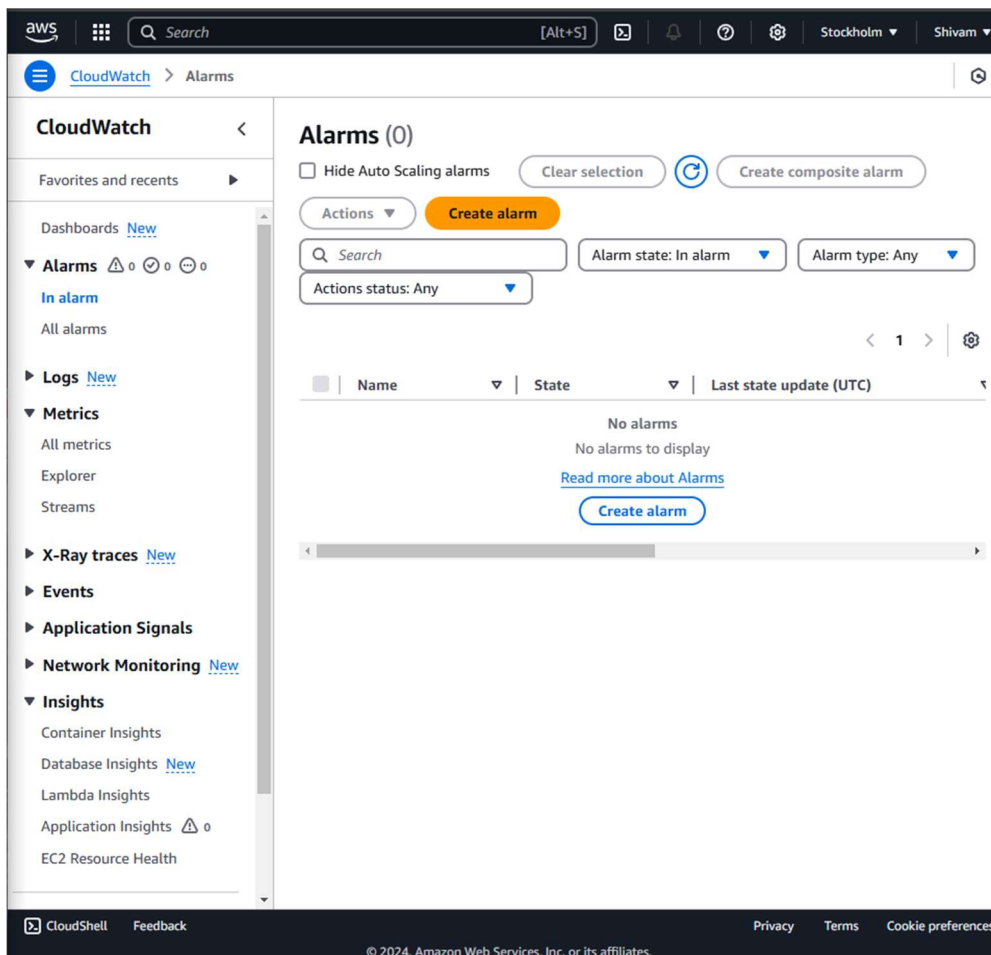
7. EC2 overview on Cloudwatch after running script



8. CPU spike simulation completion

```
ubuntu@ip-172-31-26-50: ~  
ubuntu@ip-172-31-26-50:~$ python3 cpu_spike.py  
Simulating CPU spike at 80%...  
CPU spike simulation completed.  
ubuntu@ip-172-31-26-50:~$
```

9. Create an alarm that triggers when CPU utilization reaches at 50 percent of its limit.



10. Select Metrics

Select metric

×

Untitled graph

3h1d1w

UTC timezone

Line

Percent

17.9

9.01

0.158

03:00

03:30

04:00

04:30

05:00

05:30

06:00

CPUUtilization

+

Add math

Add query

Browse

Multi source query

Graphed metrics (1)

Options

Source

Metrics (1)

☐ Alarm recommendations

Graph with SQL

Graph search

Stockholm

All > ... > Per-Instance Metrics

Search for any metric, dimension, resource id or account id

< 1 >

cpu

utilization

☒ Instance name 1/1

Instanceld

Metric name

Alarms

☒ cloudwatch-demo

i-09b5f0a96eb32...

CPUUtilization

No alarms

Cancel

Select metric

11. Configure metrics

aws CloudWatch > Alarms > Create alarm

Step 1: Specify metric and conditions

Specify metric and conditions

☐ Alarm recommendations [View details](#)

Metric

Graph
This alarm will trigger when the blue line goes above the red line for 1 datapoints within 5 minutes.

Percent
47.7
24
0.175
03:00 06:00
CPUUtilization

Namespace
AWS/EC2

Metric name
CPUUtilization

Instanceld
i-09b5f0a96eb327854

Instance name
cloudwatch-demo

Statistic
Maximum

Period
5 minutes

[Edit](#)

12. Create notification actions

aws CloudWatch > Alarms > Create alarm

Step 1: Specify metric and conditions

Step 2: Configure actions

Configure actions

Notification

Alarm state trigger
Define the alarm state that will trigger this action.

☒ **In alarm**
The metric or expression is outside of the defined threshold.

☐ **OK**
The metric or expression is within the defined threshold.

☐ **Insufficient data**
The alarm has just started or not enough data is available.

[Remove](#)

Send a notification to the following SNS topic
Define the SNS (Simple Notification Service) topic that will receive the notification.

☒ **Select an existing SNS topic**

☐ Create new topic

☐ Use topic ARN to notify other accounts

Send a notification to...
Cloudwatch-topic

Only topics belonging to this account are listed here. All persons and applications subscribed to the selected topic will receive notifications.

Email (endpoints)
effinfriday2@gmail.com - [View in SNS Console](#)

[Add notification](#)

13. Add specifications of alarm

aws

Search

[Alt+S]

Stockholm

Shivam

CloudWatch > Alarms > Create alarm

Alarm recommendations available

Turn on Recommendations to pre-populate the wizard with the recommended alarms.

Step 1

Specify metric and conditions

Step 2

Configure actions

Step 3

Add name and description

Step 4

Preview and create

Add name and description

Name and description

Alarm name

EC2 INSTANCE CPU REACHED 50 PERCENT

Alarm description - optional [View formatting guidelines](#)

EditPreview

HEY TEAM, this an automated notification from cloud watch that your instance CPU has spiked 50 percent or above. Please take necessary actions required.

Up to 1024 characters (153/1024)

Markdown formatting is only applied when viewing your alarm in the console. The description will remain in plain text in the alarm notifications.

Cancel

Previous

Next

CloudShell

Feedback

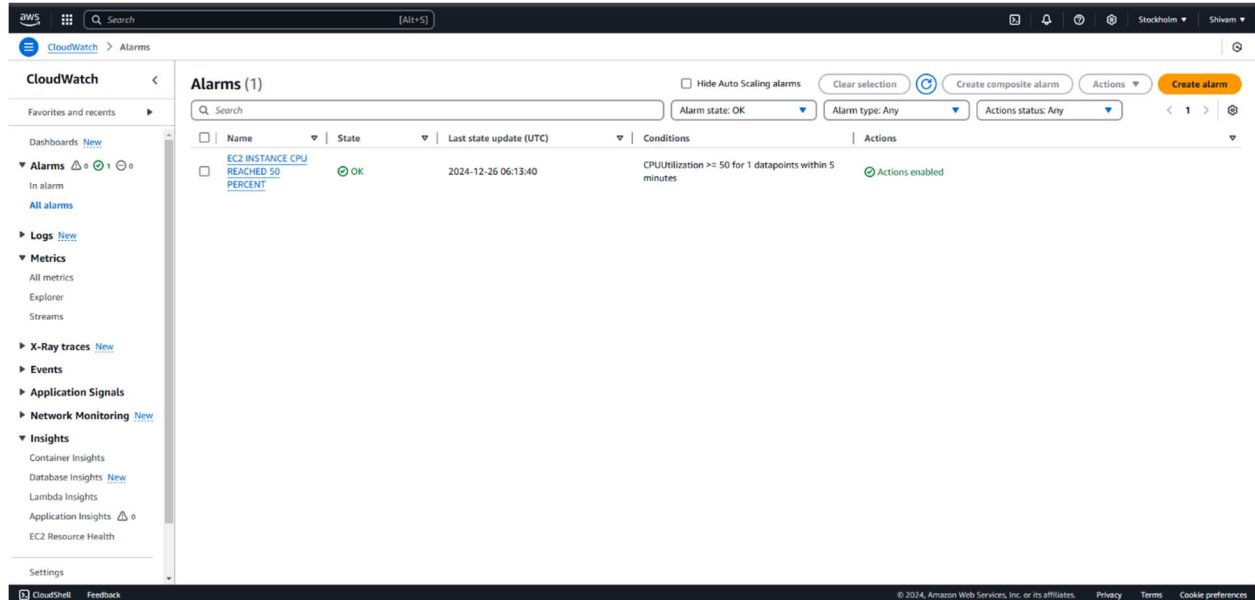
Privacy

Terms

Cookie preferences

© 2024, Amazon Web Services, Inc. or its affiliates.

14. Alarm will be created after confirming the subscription from email address



The screenshot displays the AWS CloudWatch Alarms console. The left sidebar shows the navigation menu with options like Dashboards, Alarms, Logs, Metrics, X-Ray traces, Events, Application Signals, Network Monitoring, and Insights. The main panel is titled 'Alarms (1)' and contains a table with one alarm entry. The alarm is named 'EC2 INSTANCE CPU REACHED 50 PERCENT', is in an 'OK' state, and was last updated on '2024-12-26 06:13:40'. The condition is 'CPUUtilization >= 50 for 1 datapoints within 5 minutes', and the actions are 'Actions enabled'.

Name	State	Last state update (UTC)	Conditions	Actions
EC2 INSTANCE CPU REACHED 50 PERCENT	OK	2024-12-26 06:13:40	CPUUtilization >= 50 for 1 datapoints within 5 minutes	Actions enabled

15. Run the script for testing

```
ubuntu@ip-172-31-26-50: ~  
ubuntu@ip-172-31-26-50:~$ ls  
cpu_spike.py  
ubuntu@ip-172-31-26-50:~$ python3 cpu_spike.py  
Simulating CPU spike at 80%...
```

16. Mail received regarding alarm

ALARM: "EC2 INSTANCE CPU REACHED 50 PERCENT" in EU (Stockholm)

AWS Notifications 11:54

to me

You are receiving this email because your Amazon CloudWatch Alarm "EC2 INSTANCE CPU REACHED 50 PERCENT" in the EU (Stockholm) region has entered the ALARM state, because "Threshold Crossed: 1 out of the last 1 datapoints [47.0383275261324 (26/12/24 06:19:00)] was greater than or equal to the threshold (40.0) (minimum 1 datapoint for OK -> ALARM transition)." at "Thursday 26 December, 2024 06:24:08 UTC".

View this alarm in the AWS Management Console:
<https://eu-north-1.console.aws.amazon.com/cloudwatch/deeplink.js?region=eu-north-1#alarmsV2:alarm/EC2%20INSTANCE%20CPU%20REACHED%2050%20PERCENT>

Alarm Details:

- Name: EC2 INSTANCE CPU REACHED 50 PERCENT
- Description: HEY TEAM, this an automated notification from cloud watch that your instance CPU has spiked 40 percent or above. Please take necessary actions required.
- State Change: OK -> ALARM
- Reason for State Change: Threshold Crossed: 1 out of the last 1 datapoints [47.0383275261324 (26/12/24 06:19:00)] was greater than or equal to the threshold (40.0) (minimum 1 datapoint for OK -> ALARM transition).
- Timestamp: Thursday 26 December, 2024 06:24:08 UTC
- AWS Account: 619071342081
- Alarm Arn: arn:aws:cloudwatch:eu-north-1:619071342081:alarm:EC2 INSTANCE CPU REACHED 50 PERCENT

17. Test log in alarm history

History (6)

Search

Date (UTC)	Type	Description
2024-12-26 06:26:03	Configuration update	Alarm "EC2 INSTANCE CPU REACHED 50 PERCENT" updated
2024-12-26 06:24:08	Action	Successfully executed action arn:aws:sns:eu-north-1:619071342081:Cloudwatch-topic
2024-12-26 06:24:08	State update	Alarm updated from OK to In alarm.
2024-12-26 06:23:17	Configuration update	Alarm "EC2 INSTANCE CPU REACHED 50 PERCENT" updated
2024-12-26 06:13:40	State update	Alarm updated from Insufficient data to OK.
2024-12-26 06:12:27	Configuration update	Alarm "EC2 INSTANCE CPU REACHED 50 PERCENT" created