

281-[SF]-Lab - Monitor an EC2 Instance

Umi Nur F | nurfatih@gmail.com

A. Lab overview

Logging and monitoring are techniques implemented to achieve a common goal. They work together to help ensure that a system's performance baselines and security guidelines are always met.

Logging refers to recording and storing data events as log files. Logs contain low-level details that can give you visibility into how your application or system performs under certain circumstances. From a security standpoint, logging helps security administrators identify red flags that are easily overlooked in their system.

Monitoring is the process of analyzing and collecting data to help ensure optimal performance. Monitoring helps detect unauthorized access and helps align your services' usage with organizational security.

In this lab, you create an Amazon CloudWatch alarm that initiates when the Amazon Elastic Compute Cloud (Amazon EC2) instance exceeds a specific central processing unit (CPU) utilization threshold. You create a subscription using Amazon Simple Notification Service (Amazon SNS) that sends an email to you if this alarm is goes off. You log in to the EC2 instance and run a stress test command that causes the CPU utilization of the EC2 instance to reach 100 percent.

This test simulates a malicious actor gaining control of the EC2 instance and spiking the CPU. CPU spiking has various possible causes, one of which is malware.

After completing this lab, you should be able to:

- Create an Amazon SNS notification
- Configure a CloudWatch alarm
- Stress test an EC2 instance
- Confirm that an Amazon SNS email was sent
- Create a CloudWatch dashboard

This lab requires approximately 60 minutes to complete.

B. Lab environment

The lab environment includes one preconfigured EC2 instance named Stress Test with an attached AWS Identity and Access Management (IAM) role that you can use to connect via AWS Systems Manager session manager.

All backend components, such as Amazon EC2, IAM roles, and some AWS services, have been built into the lab already.

C. Accessing the AWS Management Console

1. At the upper-right corner of these instructions, choose Start Lab

Troubleshooting tip: If you get an Access Denied error, close the error box, and choose Start Lab again.

2. The following information indicates the lab status:

- A red circle next to AWS at the upper-left corner of this page indicates that the lab has not been started.
- A yellow circle next to AWS at the upper-left corner of this page indicates that the lab is starting.
- A green circle next to AWS at the upper-left corner of this page indicates that the lab is ready.

Wait for the lab to be ready before proceeding.

3. At the top of these instructions, choose the green circle next to AWS

This option opens the AWS Management Console in a new browser tab. The system automatically sign you in.

Tip: If a new browser tab does not open, a banner or icon at the top of your browser might indicate that your browser is preventing the site from opening pop-up windows. Choose the banner or icon, and choose Allow pop-ups.

4. If you see a dialog prompting you to switch to the new console home, choose Switch to the new Console Home.

5. Arrange the AWS Management Console tab so that it displays alongside these instructions. Ideally, you should be able to see both browser tabs at the same time so that you can follow the lab steps.

Do not change the lab Region unless specifically instructed to do so.

Task 1: Configure Amazon SNS

In this task, you create an SNS topic and then subscribe to it with an email address.

Amazon SNS is a fully managed messaging service for both application-to-application (A2A) and application-to-person (A2P) communication.

6. In the AWS Management Console, enter SNS in the search bar, and then choose Simple Notification Service.
7. On the left, choose the button, choose Topics, and then choose Create topic.
8. On the Create topic page in the Details section, configure the following options:
 - Type: Choose Standard.
 - Name: Enter `MyCwAlarm`
9. Choose Create topic.

The screenshot shows the 'Create topic' page in the AWS Management Console. The breadcrumb navigation at the top reads 'Amazon SNS > Topics > Create topic'. The page title is 'Create topic'. Under the 'Details' section, the 'Type' is set to 'Standard' (selected with a radio button). The 'Name' field contains 'MyCwAlarm'. Below the name field, there is a note: 'Maximum 256 characters. Can include alphanumeric characters, hyphens (-) and underscores (_).' At the bottom of the page, there are two optional sections: 'Tags - optional' and 'Active tracing - optional'. The 'Active tracing' section is expanded, showing a description and a link to 'Learn more'. At the bottom right, there are 'Cancel' and 'Create topic' buttons.

Amazon SNS > Topics > Create topic

Create topic

Details

Type [Info](#)

Topic type cannot be modified after topic is created

☐ FIFO (first-in, first-out)

- Strictly-preserved message ordering
- Exactly-once message delivery
- Subscription protocols: SQS

☒ Standard

- Best-effort message ordering
- At-least once message delivery
- Subscription protocols: SQS, Lambda, Data Firehose, HTTP, SMS, email, mobile application endpoints

Name

MyCwAlarm

Maximum 256 characters. Can include alphanumeric characters, hyphens (-) and underscores (_).

These settings configure the logging of message delivery status to CloudWatch Logs.

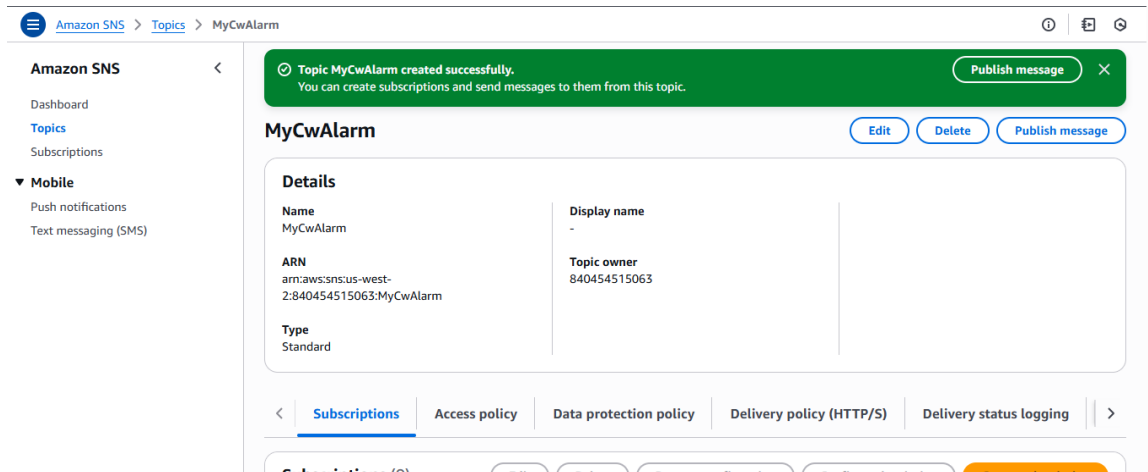
► **Tags - optional**

A tag is a metadata label that you can assign to an Amazon SNS topic. Each tag consists of a key and an optional value. You can use tags to search and filter your topics and track your costs. [Learn more](#)

► **Active tracing - optional** [Info](#)

Use AWS X-Ray active tracing for this topic to view its traces and service map in Amazon CloudWatch. Additional costs apply.

[Cancel](#) [Create topic](#)

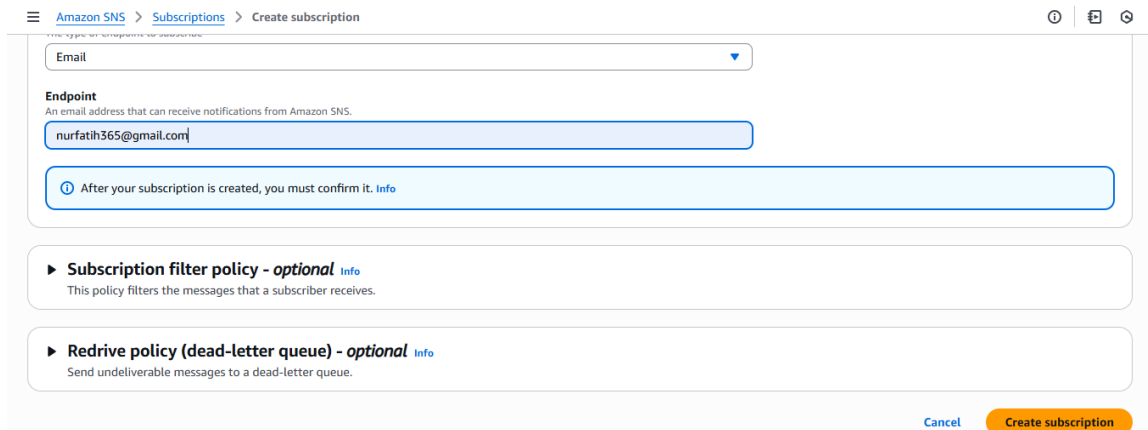


10. On the MyCwAlarm details page, choose the Subscriptions tab, and then choose Create subscription.

11. On the Create subscription page in the Details section, configure the following options:

- Topic ARN: Leave the default option selected.
- Protocol: From the dropdown list, choose Email.
- Endpoint: Enter a valid email address that you can access.

12. Choose Create subscription.



MyCwAlarm > Subscription: 6e32f0b8-0316-4634-92ea-4bee120efdda

Subscription to MyCwAlarm created successfully.
The ARN of the subscription is arn:aws:sns:us-west-2:840454515063:MyCwAlarm:6e32f0b8-0316-4634-92ea-4bee120efdda.

Subscription: 6e32f0b8-0316-4634-92ea-4bee120efdda

Edit Delete

Details	Status
ARN arn:aws:sns:us-west-2:840454515063:MyCwAlarm:6e32f0b8-0316-4634-92ea-4bee120efdda	⌚ Pending confirmation
Endpoint nurfatih365@gmail.com	Protocol EMAIL
Topic MyCwAlarm	
Subscription Principal arn:aws:iam::840454515063:role/voclabs	

In the Details section, the Status should be Pending confirmation. You should have received an AWS Notification - Subscription Confirmation email message at the email address that you provided in the previous step.

13. Open the email that you received with the Amazon SNS subscription notification, and choose Confirm subscription.

Hapus selamanya | Bukan spam | 1 dari 10

AWS Notification - Subscription Confirmation

Spam x

AWS Notifications <no-reply@sns.amazonaws.com> 16:50 (0 menit yang lalu) ☆ 😊 ↩ ⋮
kepada saya ▾

Mengapa pesan ini berada di spam? Pesan ini serupa dengan pesan yang dulu diidentifikasi sebagai spam.

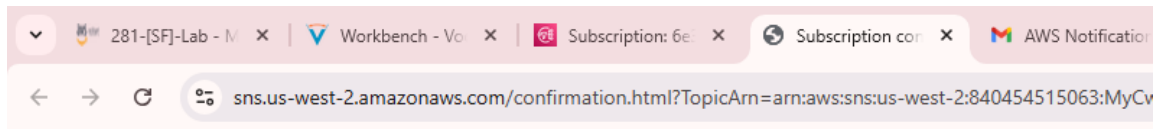
Laporkan bukan spam ⓘ

🗣 Terjemahkan ke Indonesia ✕

You have chosen to subscribe to the topic:
arn:aws:sns:us-west-2:840454515063:MyCwAlarm

To confirm this subscription, click or visit the link below (If this was in error no action is necessary):
[Confirm subscription](#)

Please do not reply directly to this email. If you wish to remove yourself from receiving all future SNS subscription confirmation requests please send an email to [sns-opt-out](#)



Simple Notification Service

Subscription confirmed!

You have successfully subscribed.

Your subscription's id is:

arn:aws:sns:us-west-2:840454515063:MyCwAlarm:6e32f0b8-0316-4634-92ea-4bee120efdda

If it was not your intention to subscribe, [click here to unsubscribe](#).

14. Go back to the AWS Management Console. In the left navigation pane, choose Subscriptions. The Status should now be Confirmed.

alarm

MyCwAlarm	
ARN arn:aws:sns:us-west-2:840454515063:MyCwAlarm	Topic owner 840454515063
Type Standard	

< Subscriptions Access policy Data protection policy Delivery policy (HTTP/S) Delivery status logging >

Subscriptions (1) Edit Delete Request confirmation Confirm subscription Create subscription

Search

ID	Endpoint	Status	Protocol
6e32f0b8-0316-4634-92ea-...	nurfatih365@gmail.com	Confirmed	EMAIL

Summary of task 1

In this task, you created an SNS topic and then created a subscription for the topic by using an email address. This topic is now able to send alerts to the email address that you associated with the Amazon SNS subscription.

Task 2: Create a CloudWatch alarm

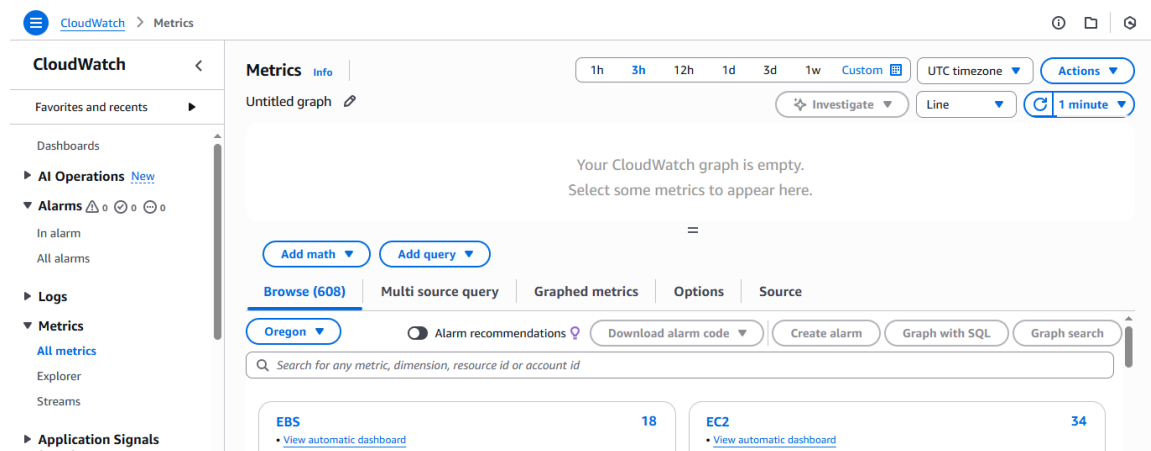
In this task, you view some metrics and logs stored within CloudWatch. You then create a CloudWatch alarm to initiate and send an email to your SNS topic if the Stress Test EC2 instance increases to more than 60 percent CPU utilization.

CloudWatch is a monitoring and observability service built for DevOps engineers, developers, site reliability engineers (SREs), IT managers, and product owners. CloudWatch provides you with data and actionable insights to monitor your applications, respond to system-wide performance changes, and optimize resource utilization. CloudWatch collects monitoring and operational data in the form of logs, metrics, and events. You get a unified view of operational health and gain visibility of your AWS resources, applications, and services running on AWS and on premises.

15. In the AWS Management Console, enter Cloudwatch in the search bar, and then choose it.

16. In the left navigation pane, choose the Metrics dropdown list, and then choose All metrics.

CloudWatch usually takes 5-10 minutes after the creation of an EC2 instance to start fetching metric details.



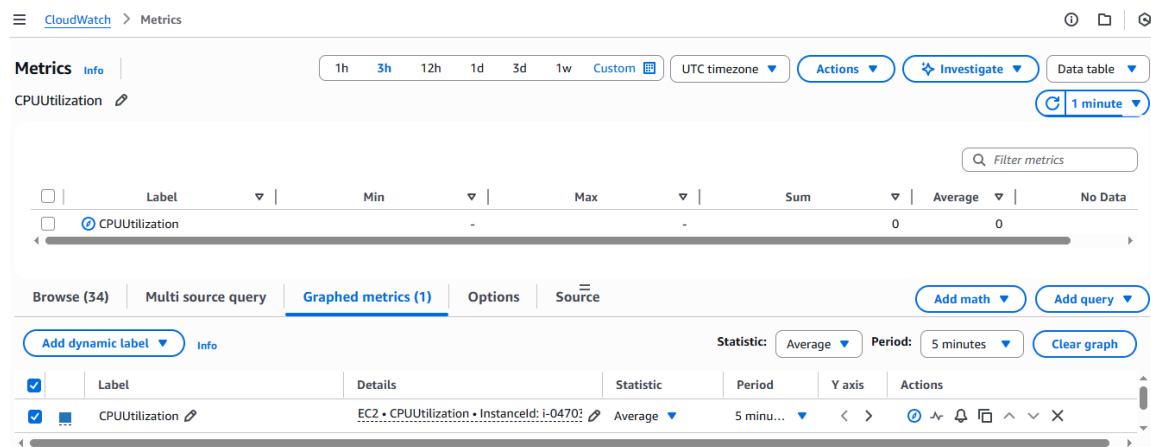
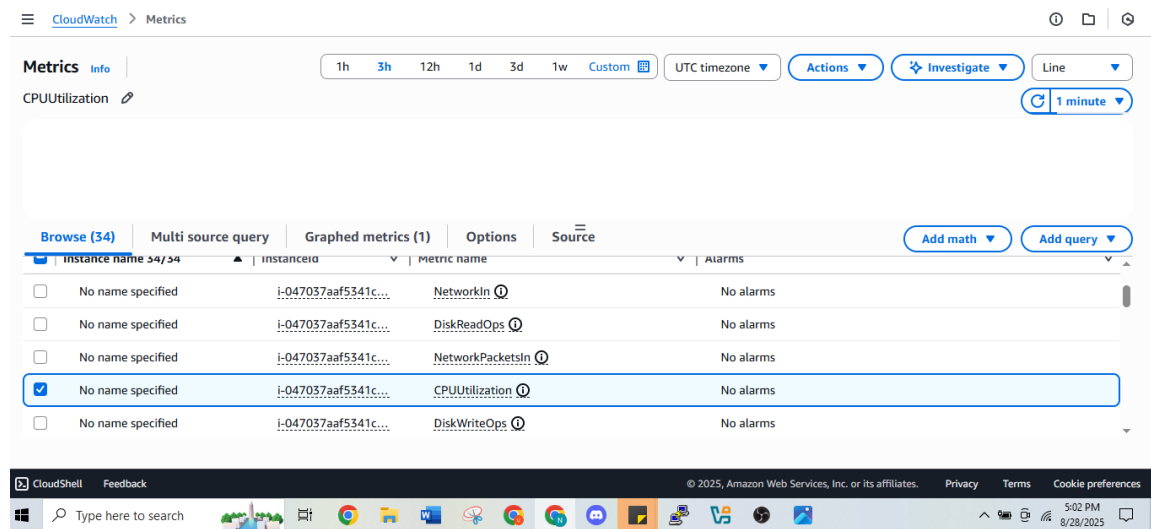
17. On the Metrics page, choose EC2, and choose Per-Instance Metrics.

From this page, you can view all the metrics being logged and the specific EC2 instance for the metrics.

18. Select the check box with CPUUtilization as the Metric name for the Stress Test EC2 instance.

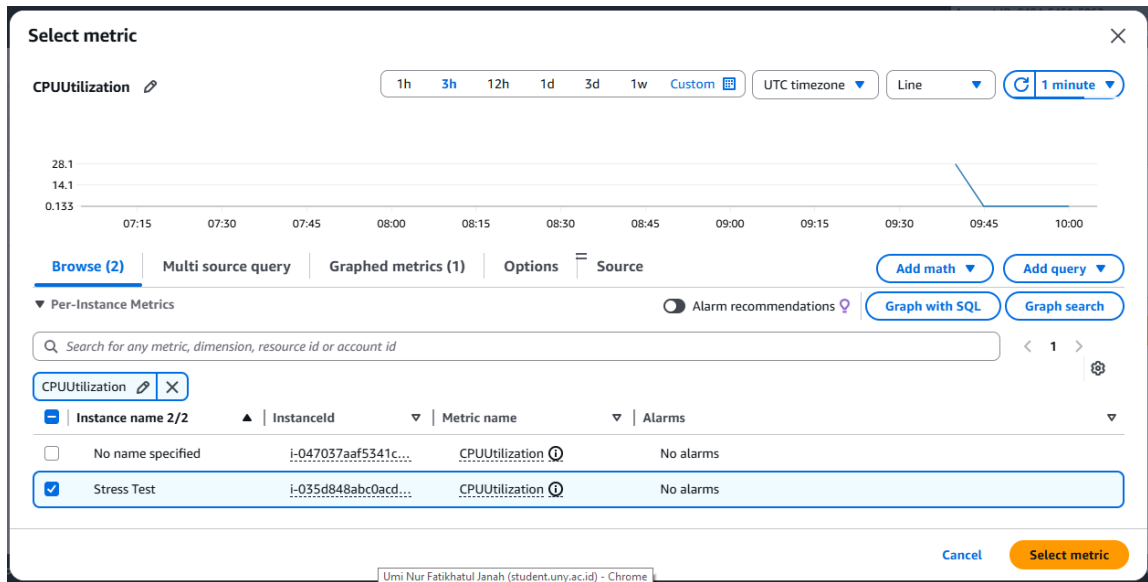
The following image shows the metrics and instance that you should select.

This option displays the graph for the CPU utilization metric, which should be approximately 0 because nothing has been done yet.



19. In the left navigation pane, choose the Alarms dropdown list, and then choose All alarms. You now create a metric alarm. A metric alarm watches a single CloudWatch metric or the result of a math expression based on CloudWatch metrics. The alarm performs one or more actions based on the value of the metric or expression relative to a threshold over a number of time periods. The action then sends a notification to the SNS topic that you created earlier.
20. Choose Create alarm.
21. Choose Select metric, choose EC2, and then choose Per-Instance Metrics.

22. Select the check box with CPUUtilization as the Metric name for the Stress Test instance name.



23. Choose Select metric.

24. On the Specify metric and conditions page, configure the following options:

Metric

- Metric name: Enter CPUUtilization
- Instanceld: Leave the default option selected.
- Statistic: Enter Average
- Period: From the dropdown list, choose 1 minute.

Conditions

- Threshold type: Choose Static.
- Whenever CPUUtilization is...: Choose Greater > threshold.
- than... Define the threshold value: Enter 60

25. Choose Next.

CloudWatch > Alarms > Create alarm

Step 3
☐ Add alarm details
☐ Step 4
☐ Preview and create

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

Percent

60 60

30.1

0.133

08:00 09:00 10:00

■ CPUUtilization

Namespace
 AWS/EC2

Metric name
 CPUUtilization

InstanceId
 i-035d848abc0acd4dc

Instance name
 Stress Test

Statistic
 Average

Period
 1 minute

CloudWatch > Alarms > Create alarm

Conditions

Threshold type

☒ Static
 Use a value as a threshold

☐ Anomaly detection
 Use a band as a threshold

Whenever CPUUtilization is...
 Define the alarm condition.

☒ Greater
 > threshold

☐ Greater/Equal
 >= threshold

☐ Lower/Equal
 <= threshold

☐ Lower
 < threshold

than...
 Define the threshold value.

60

Must be a number.

► **Additional configuration**

Cancel Next

26. On the Configure actions page, configure the following options:

Notification

- Alarm state trigger: Choose In alarm.
- Select an SNS topic: Choose Select an existing SNS topic.
- Send a notification to...: Choose the text box, and then choose MyCwAlarm.

27. Choose Next, and then configure the following options:

Name and description

- Alarm name: Enter LabCPUUtilizationAlarm
- Alarm description - *optional*: Enter CloudWatch alarm for Stress Test EC2 instance CPUUtilization

28. Choose Next

29. Review the Preview and create page, and then choose Create alarm.

CloudWatch > Alarms > Create alarm

Step 2

Configure actions

Step 3

Add alarm details

Step 4

Preview and create

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...

Q MyCwAlarm

MyCwAlarm

MyCwAlarm

consent:sperson@amazon.com

nurfatih365@gmail.com - View in SNS Console

Add notification

CloudWatch > Alarms > Create alarm

Step 2

Configure actions

Step 3

Add alarm details

Step 4

Preview and create

Name and description

Alarm name

LabCPUUtilizationAlarm

Alarm description - optional View formatting guidelines

Edit

Preview

CloudWatch alarm for Stress Test EC2 instance CPUUtilization

Lin to 1024 characters (60/1024)

CloudWatch > Alarms > Create alarm

Step 3: Add alarm details

Alarm details

Name

LabCPUUtilizationAlarm

Description

CloudWatch alarm for Stress Test EC2 instance CPUUtilization

Tags (0)

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

Create alarm

CloudWatch > Alarms

CloudWatch

Favorites and recents

Dashboards

AI Operations New

Alarms 0 0 1

In alarm

All alarms

Successfully created alarm LabCPUUtilizationAlarm.

View alarm

Alarms (1)

Hide Auto Scaling alarms

Clear selection

Create composite alarm

Actions

Create alarm

Search

Alarm state: Any

Alarm type: Any

Actions status: Any

< 1 >

	Name	State	Last state update (UTC)	Conditions
<input type="checkbox"/>	LabCPUUtilizationAlarm	Insufficient data	2025-08-28 10:15:38	CPUUtilization > 60 for 1 datapoints within 1 minute

Summary of task 2

In this task, you viewed some Amazon EC2 metrics within CloudWatch. You then created a CloudWatch alarm that initiates an In alarm state when the CPU utilization threshold exceeds 60 percent.

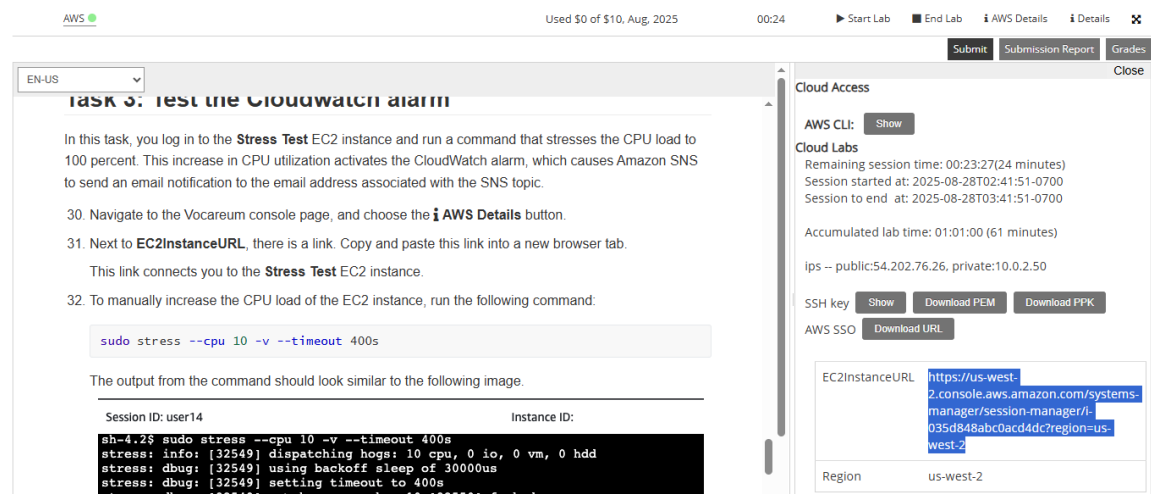
Task 3: Test the Cloudwatch alarm

In this task, you log in to the Stress Test EC2 instance and run a command that stresses the CPU load to 100 percent. This increase in CPU utilization activates the CloudWatch alarm, which causes Amazon SNS to send an email notification to the email address associated with the SNS topic.

30. Navigate to the Vocareum console page, and choose the AWS Details button.

31. Next to EC2InstanceURL, there is a link. Copy and paste this link into a new browser tab.

This link connects you to the Stress Test EC2 instance.



The screenshot displays the Vocareum console interface for Task 3. The main content area contains the task instructions and a terminal window. The terminal shows the command `sudo stress --cpu 10 -v --timeout 400s` being executed, with output indicating that 10 CPU hogs are dispatched. On the right side, the 'Cloud Access' panel is visible, showing the 'EC2InstanceURL' as `https://us-west-2.console.aws.amazon.com/systems-manager/session-manager/i-035d848abc0acd4dc7region=us-west-2` and the 'Region' as 'us-west-2'.

32. To manually increase the CPU load of the EC2 instance, run the following command:

```
sudo stress --cpu 10 -v --timeout 400s
```

```
sh-4.2$ sudo stress --cpu 10 -v --timeout 400s
stress: info: [3444] dispatching hogs: 10 cpu, 0 io, 0 vm, 0 hdd
stress: debug: [3444] using backoff sleep of 30000us
stress: debug: [3444] setting timeout to 400s
stress: debug: [3444] --> hogcpu worker 10 [3445] forked
stress: debug: [3444] using backoff sleep of 27000us
stress: debug: [3444] setting timeout to 400s
stress: debug: [3444] --> hogcpu worker 9 [3446] forked
stress: debug: [3444] using backoff sleep of 24000us
stress: debug: [3444] setting timeout to 400s
stress: debug: [3444] --> hogcpu worker 8 [3447] forked
stress: debug: [3444] using backoff sleep of 21000us
stress: debug: [3444] setting timeout to 400s
stress: debug: [3444] --> hogcpu worker 7 [3448] forked
stress: debug: [3444] using backoff sleep of 18000us
stress: debug: [3444] setting timeout to 400s
stress: debug: [3444] --> hogcpu worker 6 [3449] forked
stress: debug: [3444] using backoff sleep of 15000us
stress: debug: [3444] setting timeout to 400s
stress: debug: [3444] --> hogcpu worker 5 [3450] forked
stress: debug: [3444] using backoff sleep of 12000us
stress: debug: [3444] setting timeout to 400s
stress: debug: [3444] --> hogcpu worker 4 [3451] forked
stress: debug: [3444] using backoff sleep of 9000us
stress: debug: [3444] setting timeout to 400s
stress: debug: [3444] --> hogcpu worker 3 [3452] forked
stress: debug: [3444] using backoff sleep of 6000us
stress: debug: [3444] setting timeout to 400s
stress: debug: [3444] --> hogcpu worker 2 [3453] forked
stress: debug: [3444] using backoff sleep of 3000us
stress: debug: [3444] setting timeout to 400s
stress: debug: [3444] --> hogcpu worker 1 [3454] forked
```

This command runs for 400 seconds, loads the CPU to 100 percent, and then decreases the CPU to 0 percent after the allotted time.

33. Navigate to the Vocareum console page, and choose the AWS Details button.
34. Copy and paste the URL text next to EC2InstanceURL into another new browser tab to open a second terminal for the Stress Test instance.
35. In the new terminal, run the following command:

top

This command shows the live CPU usage.

```
top - 10:22:05 up 37 min, 0 users, load average: 8.13, 2.86, 1.03
tasks: 100 total, 11 running, 52 sleeping, 0 stopped, 0 zombie
%Cpu(s):100.0 us, 0.0 sy, 0.0 ni, 0.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 993492 total, 465028 free, 102904 used, 425560 buff/cache
MiB Swap: 0 total, 0 free, 0 used. 749144 avail Mem
```

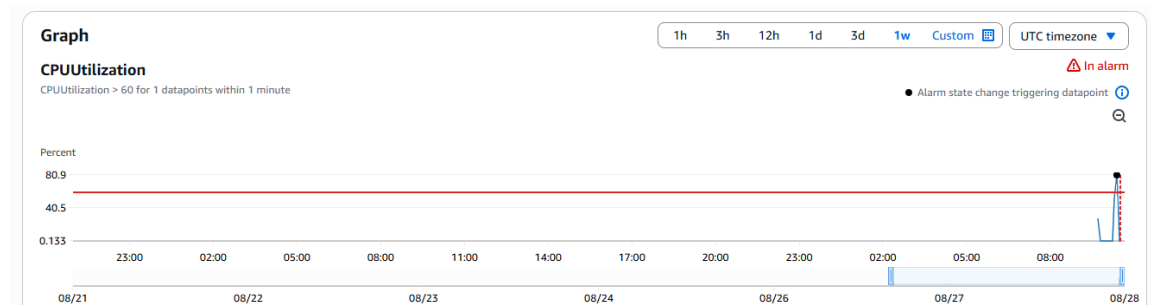
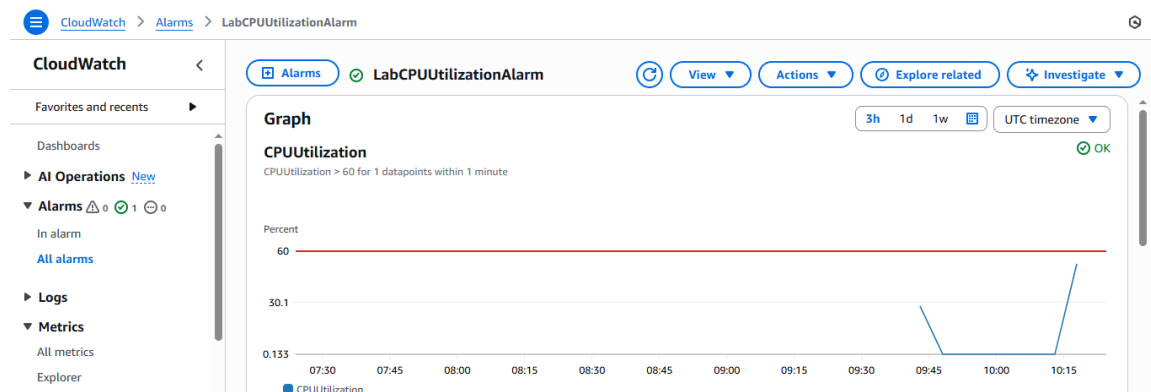
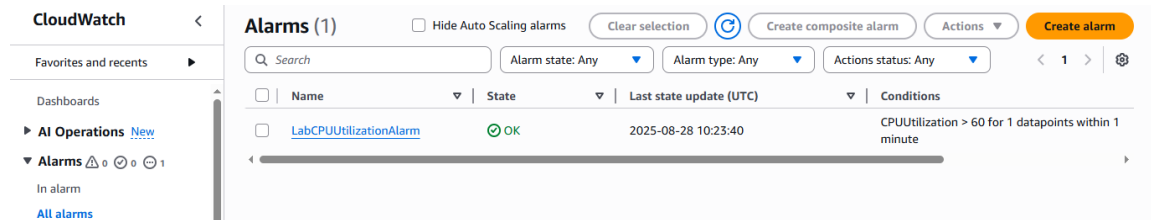
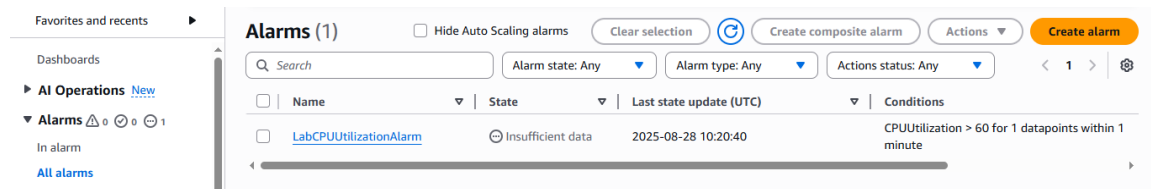
PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
3449	root	20	0	7580	92	0	R	10.3	0.0	0:10.06	stress
3451	root	20	0	7580	92	0	R	10.3	0.0	0:10.06	stress
3452	root	20	0	7580	92	0	R	10.3	0.0	0:10.06	stress
3445	root	20	0	7580	92	0	R	10.0	0.0	0:10.05	stress
3446	root	20	0	7580	92	0	R	10.0	0.0	0:10.05	stress
3450	root	20	0	7580	92	0	R	10.0	0.0	0:10.05	stress
3447	root	20	0	7580	92	0	R	9.7	0.0	0:10.05	stress
3448	root	20	0	7580	92	0	R	9.7	0.0	0:10.05	stress
3453	root	20	0	7580	92	0	R	9.7	0.0	0:10.05	stress
3454	root	20	0	7580	92	0	R	9.7	0.0	0:10.05	stress
3456	root	20	0	1244700	24296	12992	S	0.3	2.4	0:00.07	ssm-session-wor
1	root	20	0	123492	5320	3844	S	0.0	0.5	0:02.13	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/0:0H
5	root	20	0	0	0	0	I	0.0	0.0	0:00.11	kworker/u30:0
6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	mm_percpu_wq
7	root	20	0	0	0	0	S	0.0	0.0	0:00.04	ksoftirqd/0
8	root	20	0	0	0	0	I	0.0	0.0	0:00.24	rcu_sched
9	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_bh
10	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	migration/0
11	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	watchdog/0
12	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/0
14	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kdevtmpfs
15	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	netns
16	root	20	0	0	0	0	I	0.0	0.0	0:00.01	kworker/u30:1
192	root	20	0	0	0	0	S	0.0	0.0	0:00.00	khungtaskd
193	root	20	0	0	0	0	S	0.0	0.0	0:00.00	oom_reaper

36. Navigate back to the AWS console where you have the CloudWatch Alarms page open.
37. Choose LabCPUUtilizationAlarm.

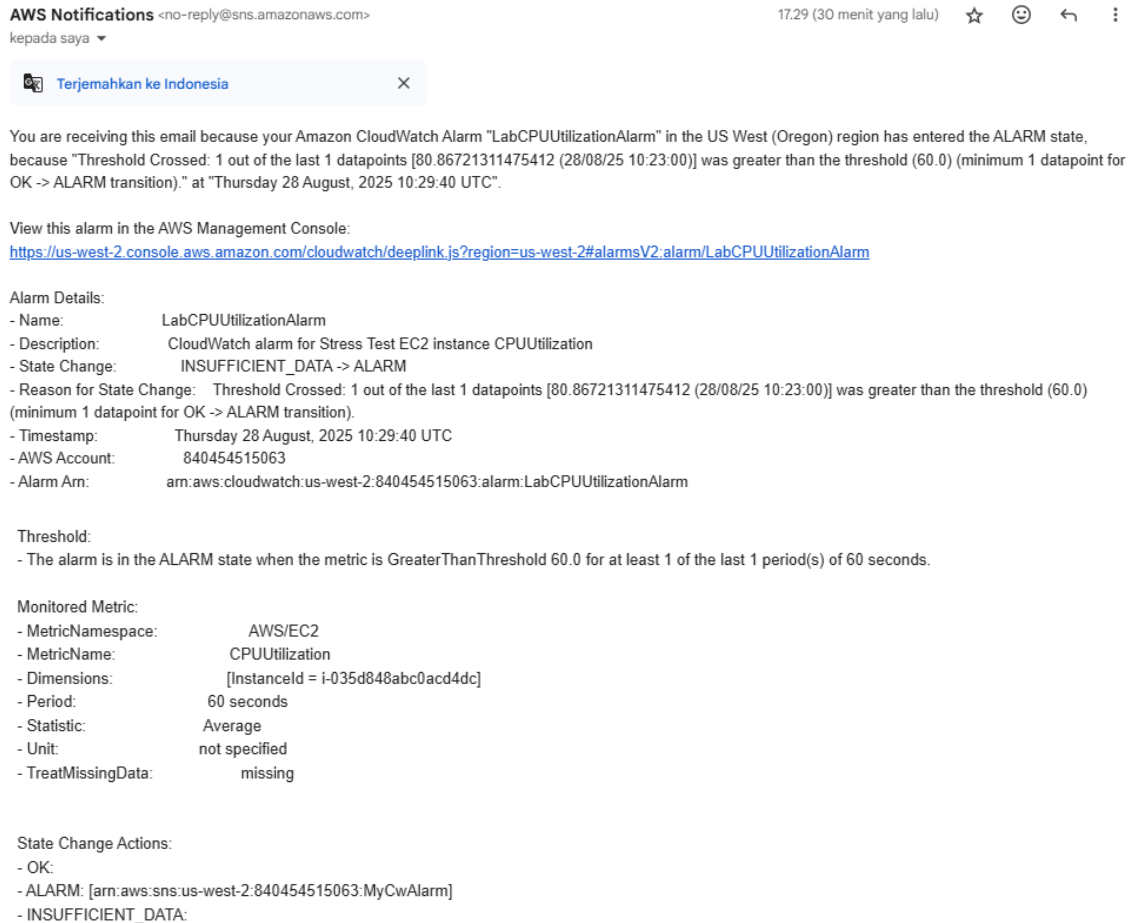
38. Monitor the graph while selecting the refresh button every 1 minute until the alarm status is In alarm.

It takes a few minutes for the alarm status to change to In alarm and for an email to send.

On the graph, you can see where CPUUtilization has increased above the 60 percent threshold.



39. Navigate to your email inbox for the email address that you used to configure the Amazon SNS subscription. You should see a new email notification from AWS Notifications.



Summary of task 3

In this task, you ran a command to load the EC2 instance to 100 percent for 400 seconds. This increase in CPU utilization activated the alarm to go into the In alarm state, and you confirmed the spike in the CPU utilization by viewing the CloudWatch graph. You also received a email notification alerting you of the In alarm state.

Task 4: Create a CloudWatch dashboard

In this task, you create a CloudWatch dashboard using the same CPUUtilization metrics that you have used throughout this lab.

CloudWatch dashboards are customizable home pages in the CloudWatch console that you can use to monitor your resources in a single view. With CloudWatch dashboards, you can even monitor resources that are spread across different Regions. You can use CloudWatch dashboards to create customized views of the metrics and alarms for your AWS resources.

40. Go to the CloudWatch section in the AWS console. In the left navigation pane, choose Dashboards.

41. Choose Create dashboard.

42. For Dashboard name, enter `LabEC2Dashboard` and then choose Create dashboard.

43. Choose Line.

44. Choose Metrics.

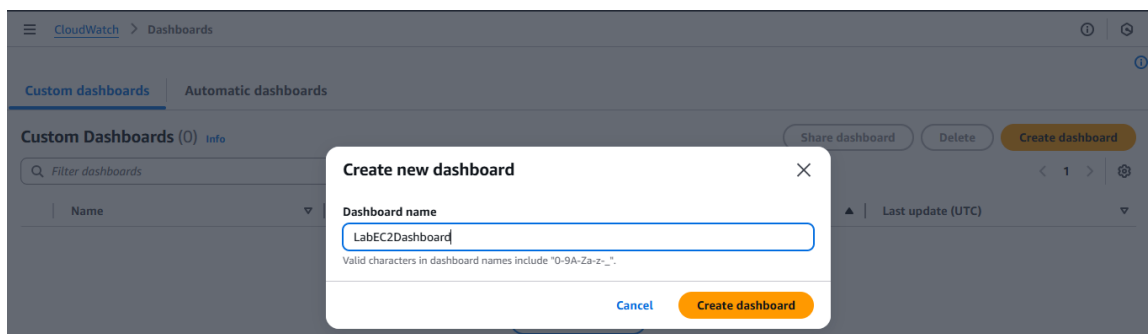
45. Choose EC2, and then choose Per-Instance Metrics.

46. Select the check box with Stress Test for the Instance name and CPUUtilization for the Metric name.

47. Choose Create widget.

48. Choose Save dashboard.

Now you have created a quick access shortcut to view the CPUUtilization metric for the Stress Test instance.



Lab summary

In this lab, you created a CloudWatch alarm that activated when the Stress Test instance exceeded a specific CPU utilization threshold. You created a subscription using Amazon SNS that sent an email to you if this alarm goes off. You logged in to the EC2 instance and ran a stress test command that spiked the EC2 instance to 100 percent CPU utilization.

This test simulated what could happen if a malicious actor were to gain control of an EC2 instance and spike CPU utilization. CPU spiking has various possible causes, one of which is malware.

Conclusion

Congratulations! You now have successfully:

- Created an Amazon SNS notification
- Configured a Cloudwatch alarm
- Stress tested an EC2 instance
- Confirmed that an Amazon SNS email was sent
- Created a CloudWatch dashboard

Lab complete

49. Choose End Lab at the top of this page, and then choose **Yes** to confirm that you want to end the lab.

50. An Ended AWS Lab Successfully message is briefly displayed indicating that the lab has ended.