Monitor an EC2 Instance

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.

Objectives

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

Duration

This lab requires approximately **60 minutes** to complete.

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.

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**.
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**.

    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**.
14. Go back to the AWS Management Console. In the left navigation pane, choose **Subscriptions**.

The **Status** should now be ✔ **Confirmed**.

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:

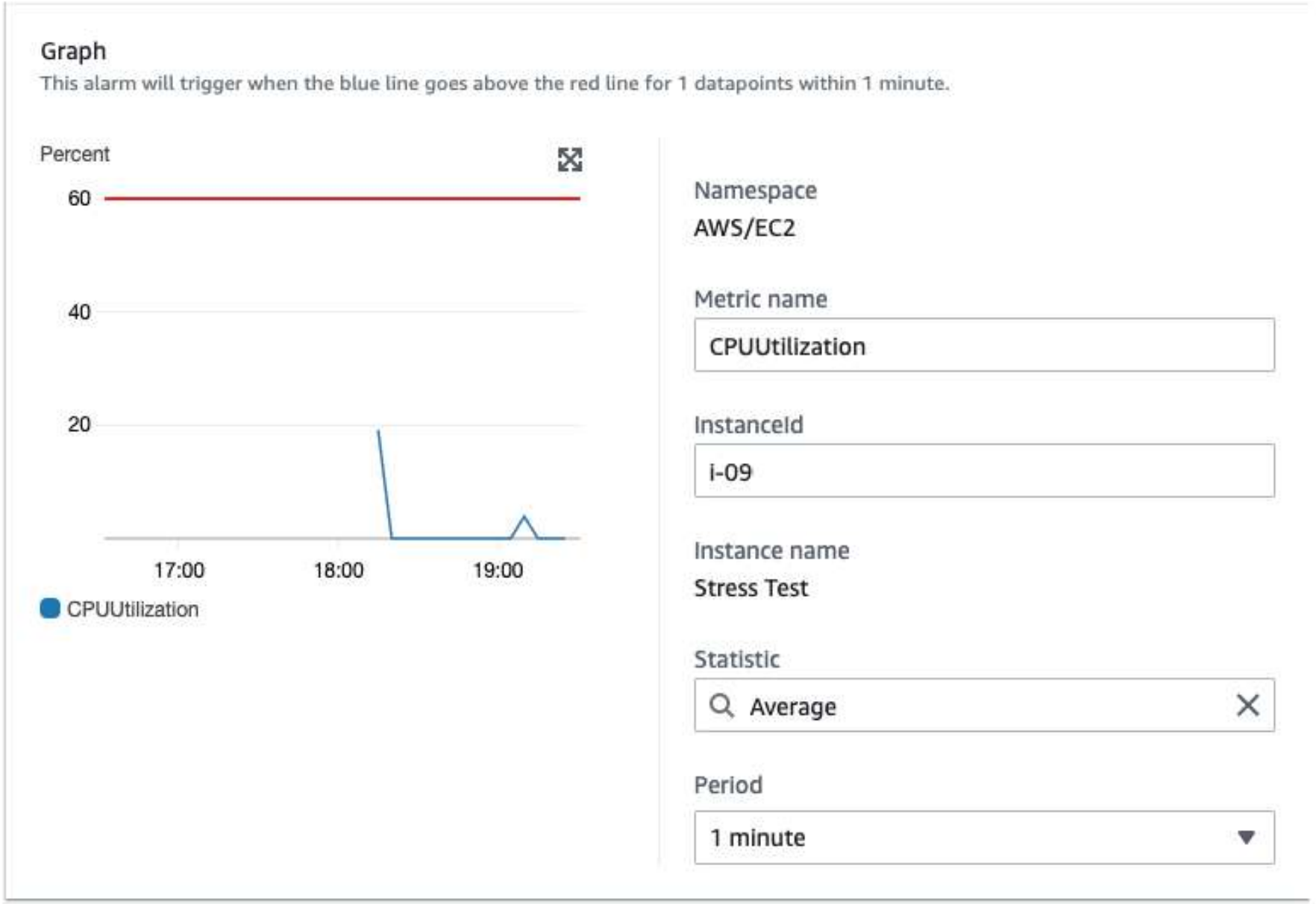
Metric

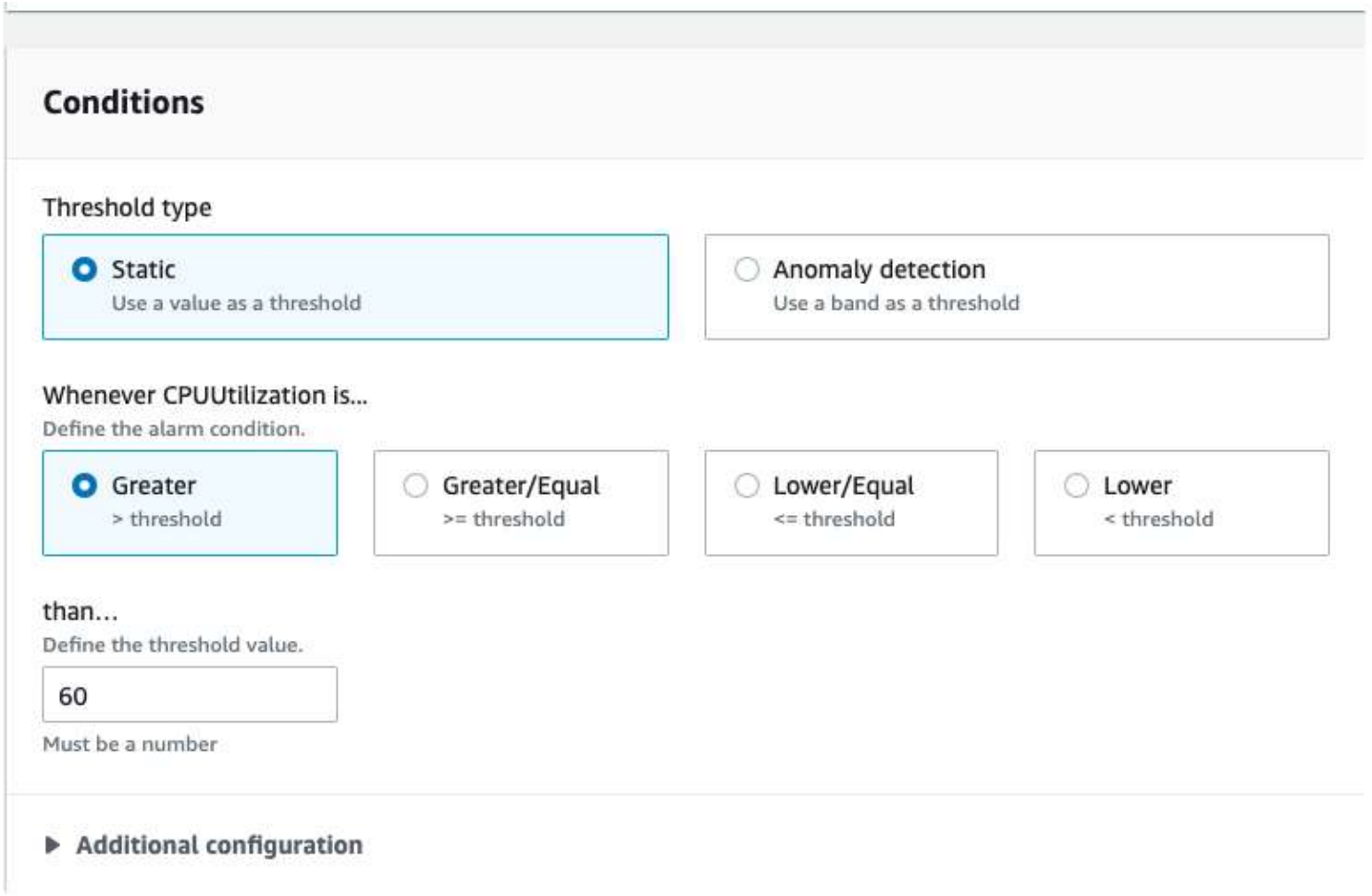
- **Metric name**: Enter `CPUUtilization`
- **InstanceId**: Leave the default option selected.
- **Statistic**: Enter `Average`
- **Period**: From the dropdown list, choose **1 minute**.

Conditions

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

The following image illustrates an example of an alarm configuration.





25. Choose **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**.

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

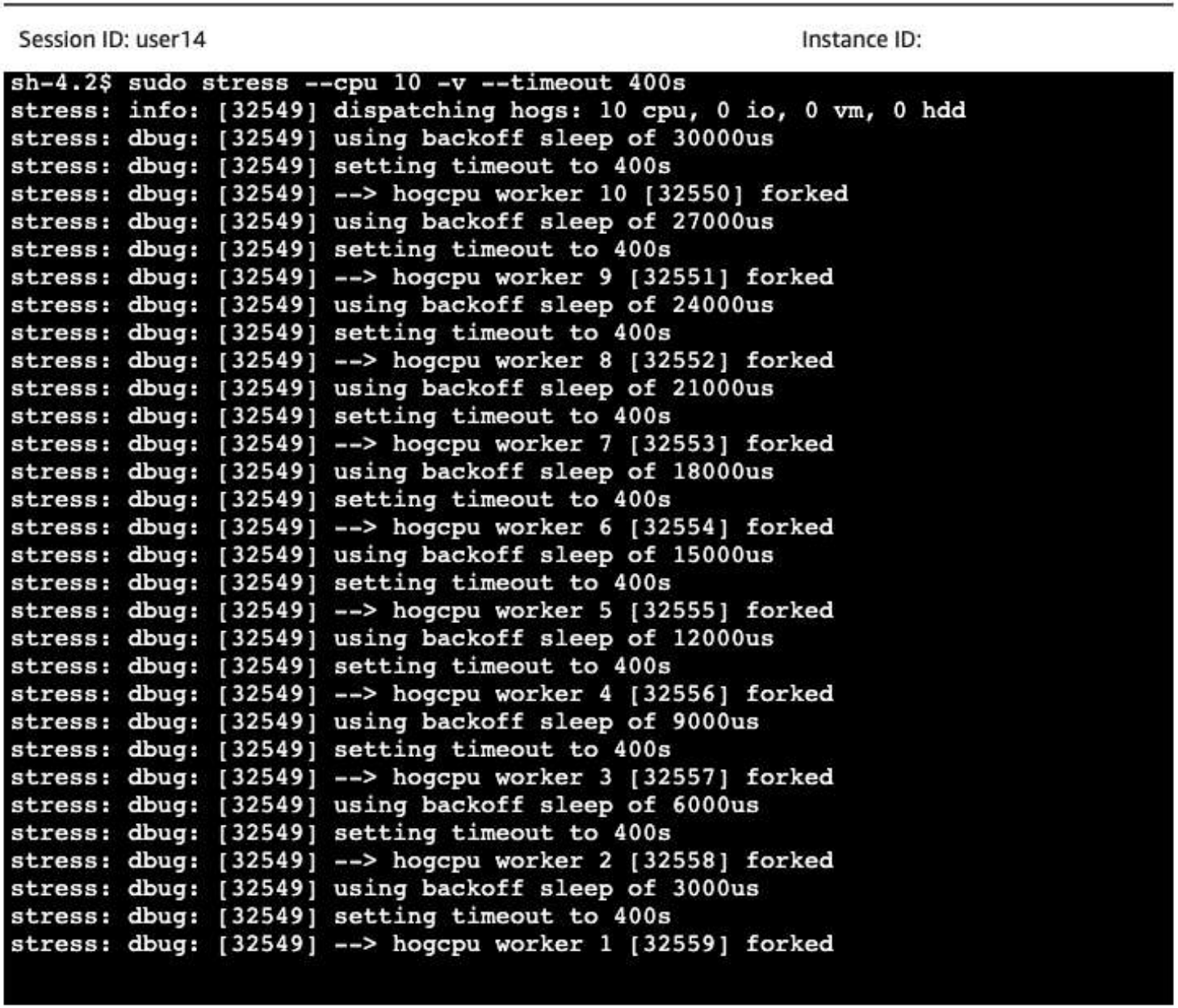
In this task, you log in to the **Stress Test** EC2 instance and run a command that stresses the CPU load to 100 percent. This increase in CPU utilization activates the CloudWatch alarm, which causes Amazon SNS to send an email notification to the email address associated with the SNS topic.

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

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

The output from the command should look similar to the following image.



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

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.
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.

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.