

# Creating a CPU usage alarm

You can create an CloudWatch alarm that sends a notification using Amazon SNS when the alarm changes state from OK to ALARM.

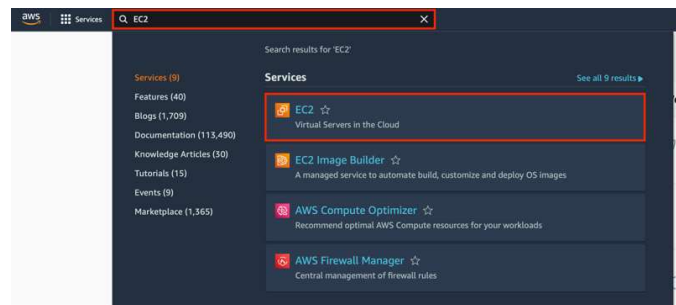
The alarm changes to the ALARM state when the average CPU use of an EC2 instance exceeds a specified threshold for consecutive specified periods.

Steps:

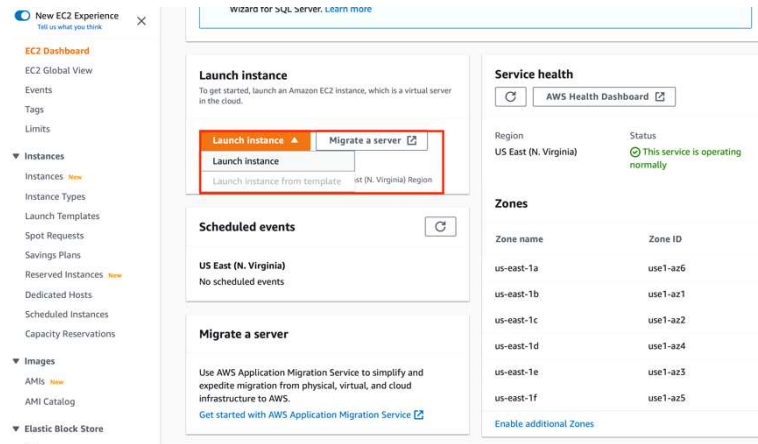
- Launch an EC2 Instance
- Set up a CPU usage alarm using the AWS Management Console
- Initiate an SSH connection to the EC2 instance
- Install the stress package and run the stress command
- Monitor the CPU Utilization of EC2 Instance via CloudWatch
- Terminate your instance

## Step A: Launch an EC2 instance

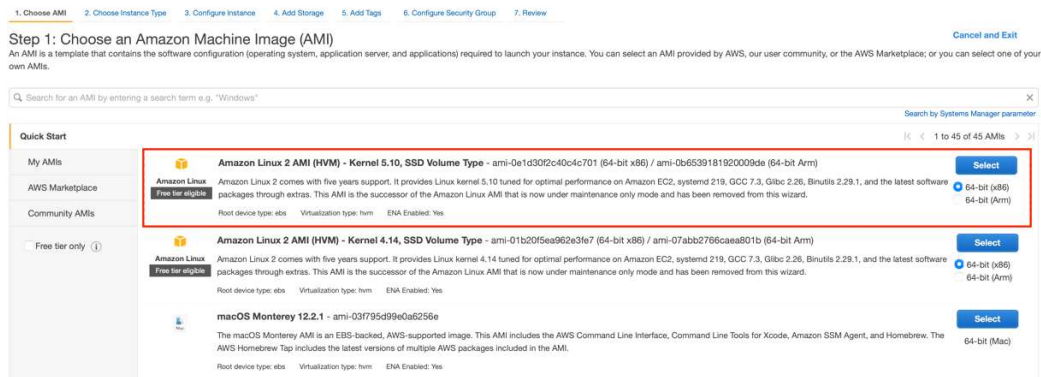
- Login to AWS Management Console and search for EC2 in the search bar. Once EC2 service shows up in the list, click and go to EC2 dashboard.



- Click Launch Instance.



- On Step 1: Choose an Amazon Machine Image (AMI), select the Amazon Linux 2 AMI.



- On Step 2: Choose an Instance Type, choose t2.micro as the instance type and click Review and Launch.

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

### Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Filter by: All instance families Current generation Show/Hide Columns

Currently selected: t2.micro (- ECUs, 1 vCPUs, 2.5 GHz, ~, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
<input type="checkbox"/>	t2	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	t2	t2.micro <small>Free tier eligible</small>	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.large	2	8	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.xlarge	4	16	EBS only	-	Moderate	Yes

Cancel Previous **Review and Launch** Next: Configure Instance Details

- **On Step 7: Review Instance Launch, click Launch.**

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

### Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

**⚠ Improve your instances' security. Your security group, launch-wizard-13, is open to the world.**

Your instances may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only. You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. [Edit security groups](#)

AMI Details [Edit AMI](#)

**Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type - ami-0c02fb59950c7d316**

Free tier eligible Amazon Linux 2 comes with five years support. It provides Linux kernel 5.10 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras. This AMI is the successor of the Amazon Linux AMI that is n...

Root Device Type: ebs Virtualization type: hvm

Instance Type [Edit instance type](#)

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	-	1	1	EBS only	-	Low to Moderate

Security Groups [Edit security groups](#)

Security group name: launch-wizard-13  
Description: launch-wizard-13 created 2022-03-18T10:51:07.524+05:30

Cancel Previous **Launch**

- **When prompted to select an existing key pair or create a new key pair, choose Proceed without a key pair. For this demo, access to instance via SSH is not required.**

Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance. Amazon EC2 supports ED25519 and RSA key pair types.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

☒ Choose an existing key pair
 ☐ Create a new key pair
 ☐ Proceed without a key pair

☐ I acknowledge that I have access to the corresponding private key file, and that without this file, I won't be able to log into my instance.

Cancel
Launch Instances

- Check the acknowledgment option and click **Launch Instances** to launch the EC2 instance.

Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance. Amazon EC2 supports ED25519 and RSA key pair types.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Proceed without a key pair

☒ I acknowledge that without a key pair, I can connect to this instance only by using EC2 Instance Connect or if I know the password built into the AMI. Note that EC2 Instance Connect is only supported on Amazon Linux 2 and Ubuntu. [Learn more](#).

Cancel
Launch Instances

- Click on the instance ID to view the EC2 instance we just launched.

#### Launch Status

Your instances are now launching

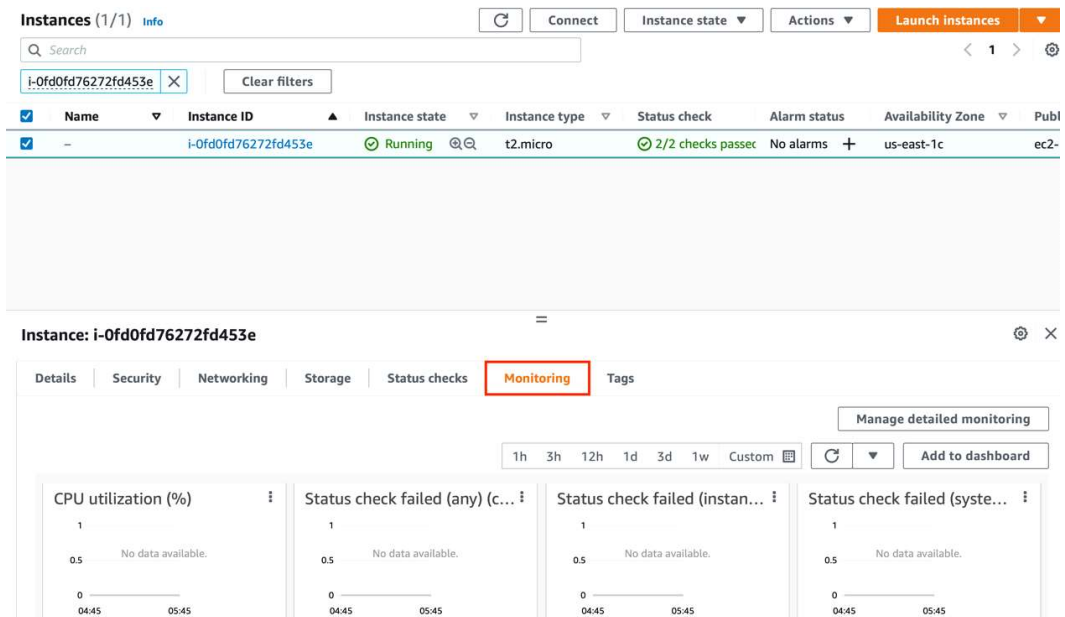
The following instance launches have been initiated: -0fd0fd76272fd453e [View launch log](#)

Get notified of estimated charges

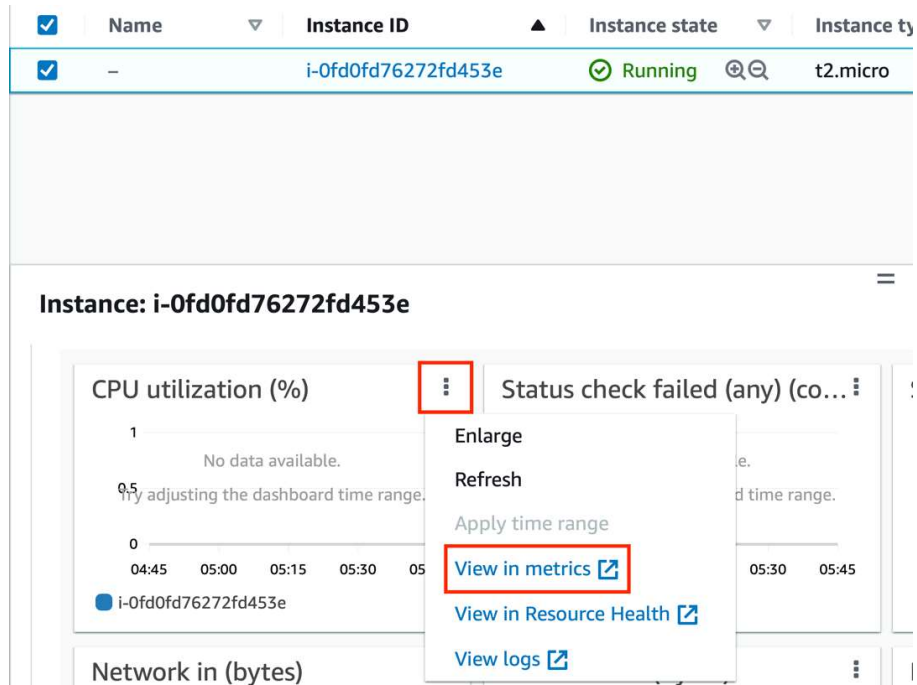
Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).

## Step B: Set up a CPU usage alarm using the AWS Management Console

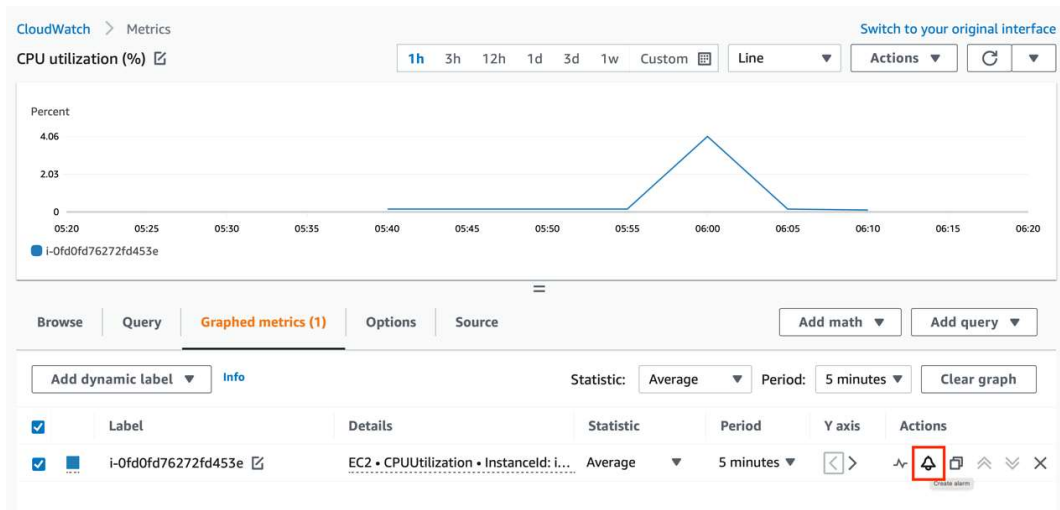
- Go to the Monitoring section to access and view all metrics based on which this instance's performance will be measured.



- Within CPU utilization metric, click three vertical dots and then click View in metrics.

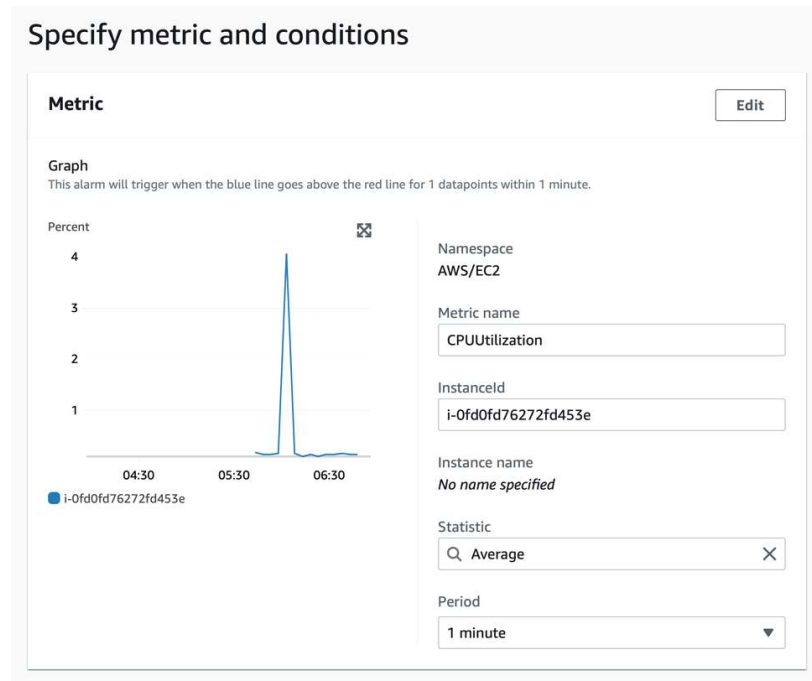


- This will take you the CloudWatch dashboard. Under **Graphed metrics**, you can access and view the instance ID along with other relevant details. On this row, click on the bell icon under **Actions**.



- Under **Specify metric and conditions**, ensure that following values are assigned:

- Metric Name = CPUUtilization
- InstanceID = Actual instance ID of the launched instance
- Statistic = Average
- Period = 1 minute



- Under **Conditions**, specify the following:
- For **Threshold type**, choose **Static**.
- For **Whenever CPUUtilization** is, specify **Greater**. Under **than...**, specify the **threshold that is to trigger the alarm to go to ALARM state if the CPU utilization exceeds this percentage. For example, 70.**
- Choose **Next**.

**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.

70

Must be a number

► Additional configuration

Cancel Next

- Under **Notification**, choose **In alarm** and create an SNS topic to notify when the alarm is in *ALARM* state.
- For **Select an SNS topic**, choose **Create a new topic**.
- Within **Create a new topic...**, specify the SNS topic name or keep it to default.
- Within **Email endpoints that will receive the notification...**, add a comma-separated list of email addresses. Each address will be added as a subscription to the topic above.
- Click **Create Topic**.

**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

Select an 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

Create a new topic...

The topic name must be unique.

Default\_CloudWatch\_Alarms\_Topic

SNS topic names can contain only alphanumeric characters, hyphens (-) and underscores (\_).

Email endpoints that will receive the notification...

Add a comma-separated list of email addresses. Each address will be added as a subscription to the topic above.

rohan@example.com, steve@example.com

user1@example.com, user2@example.com

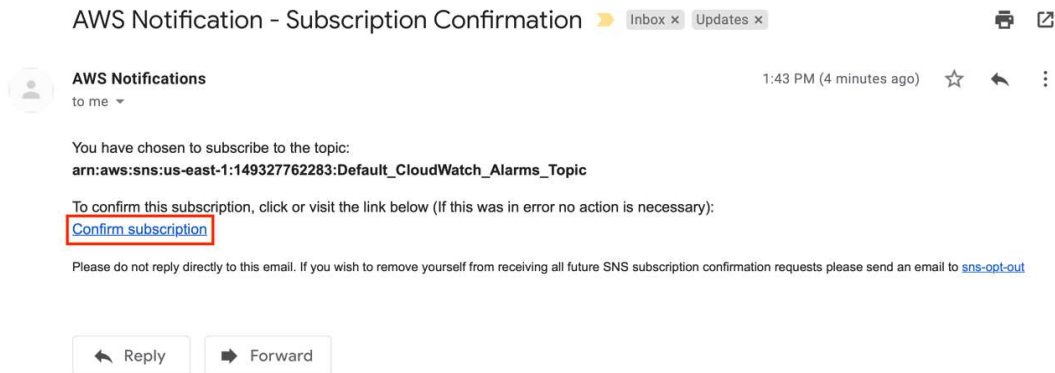
Create topic

Add notification

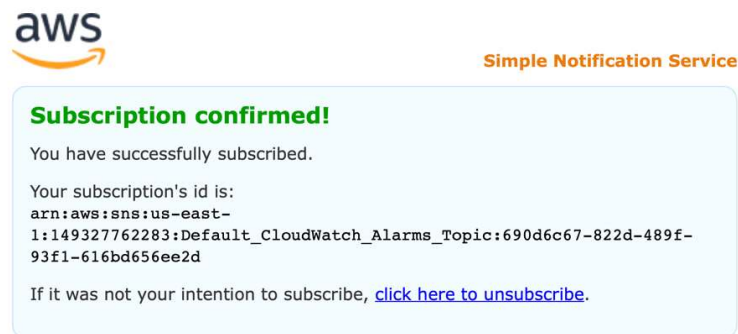
- You must confirm the subscription before notifications can be sent to an email address.
- Check your inbox and look for subscription e-mail received.



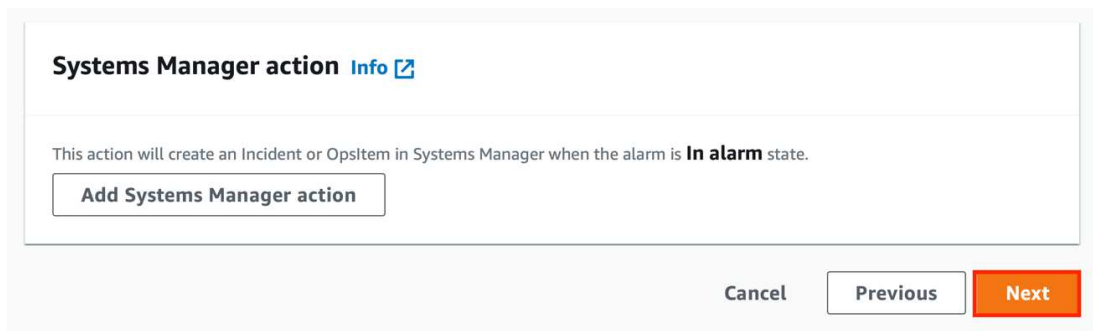
- Open message and click **Confirm Subscription**.



- Once successfully subscribed, you will be sent to subscription confirmation page.



- Go back to CloudWatch Management Console and choose **Next**.



- Enter a name and description for the alarm. The name must contain only ASCII

characters. Then choose **Next**.

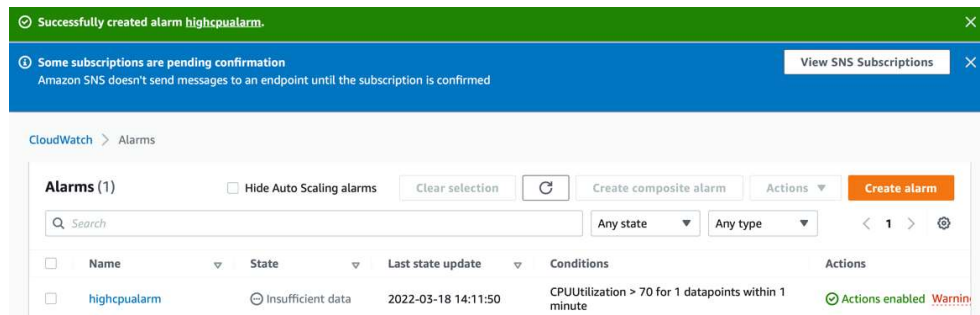
The screenshot shows the 'Add name and description' step of the AWS CloudWatch 'Create alarm' wizard. It features a 'Name and description' section with an 'Alarm name' field containing 'highcpualarm' and an 'Alarm description - optional' text area with the text: 'This alarm changes to the ALARM state when the average CPU use of an EC2 instance exceeds a specified threshold for consecutive specified periods.' At the bottom right, there are 'Cancel', 'Previous', and 'Next' buttons.

- Under **Preview and create**, confirm that the information and conditions are what you want, then choose **Create alarm**.

The screenshot shows the 'Preview and create' step of the AWS CloudWatch 'Create alarm' wizard. It includes a sidebar with steps: Step 1 (Specify metric and conditions), Step 2 (Configure actions), Step 3 (Add name and description), and Step 4 (Preview and create). The main area is titled 'Step 1: Specify metric and conditions' and contains a 'Metric' section. This section includes a 'Graph' showing a line chart with a red threshold line at 60% and a blue data line. To the right of the graph, the following details are listed: Namespace (AWS/EC2), Metric name (CPUUtilization), InstanceId (i-0fd0fd76272fd453e), Instance name (No name specified), Statistic (Average), and Period (1 minute). An 'Edit' button is located in the top right corner.

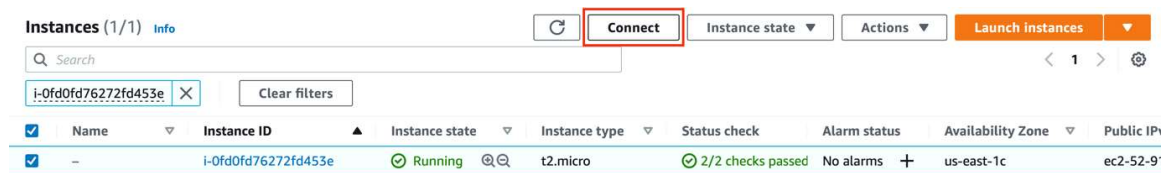
The screenshot shows the 'Step 3: Add name and description' step of the AWS CloudWatch 'Create alarm' wizard. It features a 'Name and description' section with a 'Name' field containing 'highcpualarm' and a 'Description' text area with the text: 'This alarm changes to the ALARM state when the average CPU use of an EC2 instance exceeds a specified threshold for consecutive specified periods.' At the bottom right, there are 'Cancel', 'Previous', and 'Create alarm' buttons. The 'Create alarm' button is highlighted with a red border.

- This will complete the process of alarm creation on CloudWatch.



## Step C: Initiate an SSH connection to EC2 instance

- Go back to instance dashboard, choose the instance, and click **Connect**.



- Under EC2 Instance Connect, click **Connect**.

### Connect to instance [Info](#)

Connect to your instance i-0fd0fd76272fd453e using any of these options


EC2 Instance Connect

Session Manager


SSH client

EC2 Serial Console

Instance ID

 i-0fd0fd76272fd453e


Public IP address

 52.91.200.32

User name

ec2-user

Connect using a custom user name, or use the default user name ec2-user for the AMI used to launch the instance.

 **Note:** In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

Cancel

Connect

## Step D: Install the stress utility and run the stress command

- Run following commands to install stress utility:
- sudo amazon-linux-extras install epel -y
- sudo yum install stress -y

```
Connect to instance | EC2 Management Console | i-0fd0fd76272f453e | EC2 Instance Connect
Last login: Fri Mar 18 12:12:05 2022 from ec2-18-206-107-25.compute-1.amazonaws.com

  _|  _|  _|
 _|  _|  _|  Amazon Linux 2 AMI
 _|  _|  _|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-80-36 ~]$ sudo amazon-linux-extras install epel -y
Installing epel-release
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Cleaning repos: amzn2-core amzn2extra-docker amzn2extra-epel amzn2extra-kernel-5.10
17 metadata files removed
6 sqlite files removed
0 metadata files removed
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core | 3.7 kB 00:00:00
amzn2extra-docker | 3.0 kB 00:00:00
amzn2extra-epel | 3.0 kB 00:00:00
amzn2extra-kernel-5.10 | 3.0 kB 00:00:00
(1/9): amzn2-core/2/x86_64/group_gz | 2.5 kB 00:00:00
(2/9): amzn2-core/2/x86_64/updateinfo | 452 kB 00:00:00
(3/9): amzn2extra-epel/2/x86_64/primary_db | 1.8 kB 00:00:00
(4/9): amzn2extra-kernel-5.10/2/x86_64/updateinfo | 12 kB 00:00:00
(5/9): amzn2extra-docker/2/x86_64/updateinfo | 5.9 kB 00:00:00
```

i-0fd0fd76272f453e

Public IPs: 52.91.200.32 Private IPs: 172.31.80.36

```
Connect to instance | EC2 Management Console | i-0fd0fd76272f453e | EC2 Instance Connect
59 postgresql13 available [ =stable ]
60 mock2 available [ =stable ]
61 dnsmasq2.85 available [ =stable ]
[ec2-user@ip-172-31-80-36 ~]$ sudo yum install stress -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
209 packages excluded due to repository priority protections
Resolving Dependencies
--> Running transaction check
--> Package stress.x86_64 0:1.0.4-16.el7 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package Arch Version Repository Size
=====
Installing:
stress x86_64 1.0.4-16.el7 epel 39 k
Transaction Summary
=====
Install 1 Package

Total download size: 39 k
```

i-0fd0fd76272f453e

Public IPs: 52.91.200.32 Private IPs: 172.31.80.36

- Now, run the following command to impose stress to spike CPU utilization of the instance.

- stress -c 5

```

Last login: Sat Mar 19 05:37:28 2022 from ec2-18-206-107-24.compute-1.amazonaws.com

  _|_  _|_  )
 _|_ ( _|_ /  Amazon Linux 2 AMI
 _|_ \ _|_ |

https://aws.amazon.com/amazon-linux-2/
No packages needed for security; 2 packages available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-80-36 ~]$ stress -c 5
stress: info: [8960] dispatching hogs: 5 cpu, 0 io, 0 vm, 0 hdd

```

i-0fd0fd76272fd453e

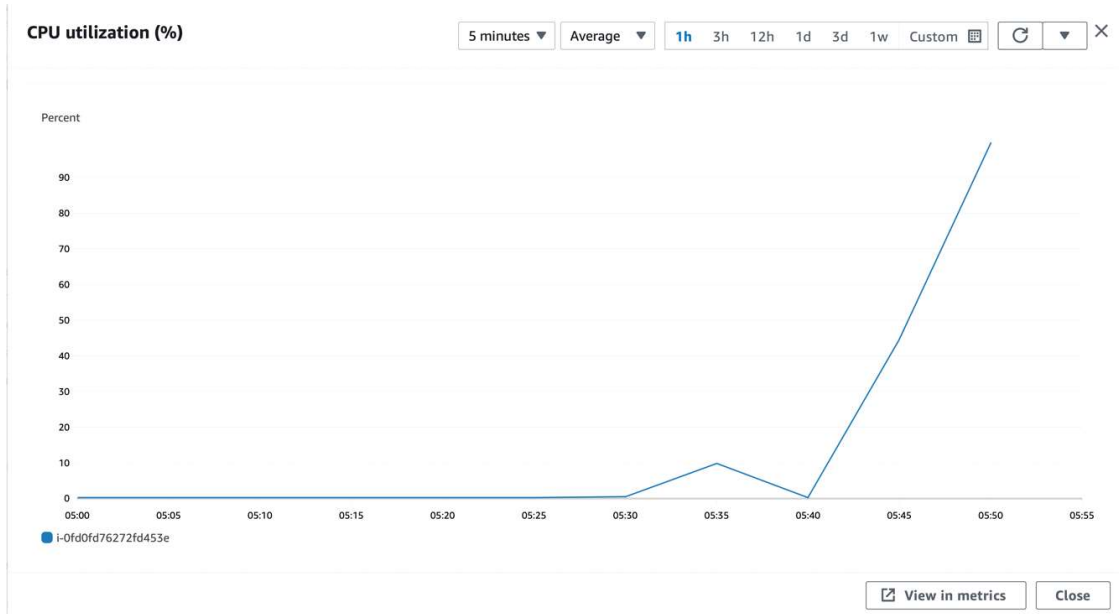
Public IPs: 52.91.200.32 Private IPs: 172.31.80.36

## Step E: Monitor the CPU Utilization of EC2 Instance via CloudWatch

- Go back to EC2 Management Console, choose the EC2 instance, go to **Monitoring**, click three vertical dots on **CPU Utilization** and click **Enlarge**.

The screenshot shows the AWS EC2 Management Console interface. On the left is a navigation sidebar with options like EC2 Dashboard, Events, Tags, Limits, and various instance types. The main panel displays the 'Instances (1/1)' list with one instance, 'i-0fd0fd76272fd453e', in a 'Running' state. Below this, the 'Monitoring' tab is selected for the specific instance. It shows several graphs: 'CPU utilization (%)', 'Status check failed (any...)', 'Status check failed (inst...)', 'Status check failed (s...', 'Network in (bytes)', and 'Network packets in (cou...)' and 'Network packets out'. The 'CPU utilization (%)' graph shows a peak around 05:50. A context menu is open over this graph, with the 'Enlarge' option highlighted in a red box. Other options in the menu include 'Refresh', 'Apply time range', 'View in metrics', 'View in Resource Health', and 'View logs'.

- In next few minutes you will see a spike in CPU utilization.



- Additionally, you can check your mailbox as you will receive an e-mail for this alarm generated via CloudWatch.

ALARM: "highcpualarm" in US East (N. Virginia)

Inbox x Promotions x



**AWS Notifications**

11:29 AM (4 minutes ago)



to me ▼

You are receiving this email because your Amazon CloudWatch Alarm "highcpualarm" in the US East (N. Virginia) region has entered the ALARM state, because "Threshold Crossed: 1 out of the last 1 datapoints [99.73978494623655 (19/03/22 05:53:00)] was greater than the threshold (70.0) (minimum 1 datapoint for OK -> ALARM transition)." at "Saturday 19 March, 2022 05:59:23 UTC".

View this alarm in the AWS Management Console:

<https://us-east-1.console.aws.amazon.com/cloudwatch/deeplink.js?region=us-east-1#alarmsV2:alarm/highcpualarm>

Alarm Details:

- Name: highcpualarm  
 - Description: This alarm changes to the ALARM state when the average CPU use of an EC2 instance exceeds a specified threshold for consecutive specified periods.  
 - State Change: INSUFFICIENT\_DATA -> ALARM  
 - Reason for State Change: Threshold Crossed: 1 out of the last 1 datapoints [99.73978494623655 (19/03/22 05:53:00)] was greater than the threshold (70.0) (minimum 1 datapoint for OK -> ALARM transition).  
 - Timestamp: Saturday 19 March, 2022 05:59:23 UTC  
 - AWS Account: 149327762283  
 - Alarm Arn: arn:aws:cloudwatch:us-east-1:149327762283:alarm:highcpualarm

Threshold:

- The alarm is in the ALARM state when the metric is GreaterThanThreshold 70.0 for at least 1 of the last 1 period(s) of 60 seconds.

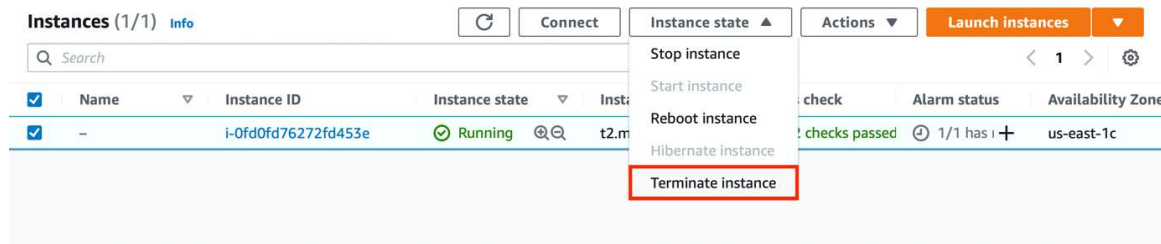
Monitored Metric:

- MetricNamespace: AWS/EC2  
 - MetricName: CPUUtilization  
 - Dimensions: [InstanceId = i-0fd0fd76272fd453e]  
 - Period: 60 seconds  
 - Statistic: Average  
 - Unit: not specified  
 - TreatMissingData: missing

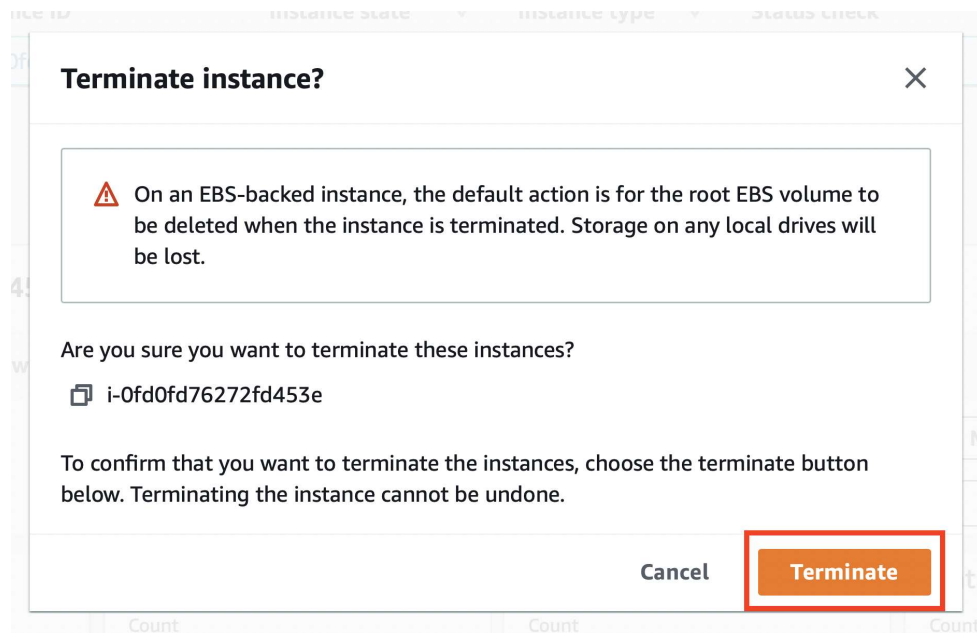
## Step F: Terminate your instance

If you launched an instance that is not within the [AWS Free Tier](#), you'll stop incurring charges for that instance as soon as the instance status changes to shutting down or terminated. To keep your instance for later, but not incur charges, you can stop the instance now and then start it again later.

- In the navigation pane, choose Instances. In the list of instances, select the instance. Choose **Instance state**, **Terminate instance**.



- Choose **Terminate** when prompted for confirmation.





Amazon EC2 shuts down and terminates your instance. After your instance is terminated, it remains visible on the console for a short while, and then the entry is automatically deleted.

You cannot remove the terminated instance from the console display yourself.