# Task Details:

1. Sign into the AWS Management Console.
2. Create an EC2 Instance.
3. SSH into EC2 Instance and install necessary Softwares.
4. Create SNS Topic.
5. Subscribe to an SNS Topic.
6. Check EC2 CPU Utilization Metrics in CloudWatch Metrics.
7. Create CloudWatch Alarm.
8. Testing CloudWatch Alarm by Stressing CPU Utilization.
9. Checking For an Email from the SNS Topic.
10. Checking the CloudWatch Alarm Graph.
11. Create a CloudWatch Dashboard.

# ****Lab Steps****

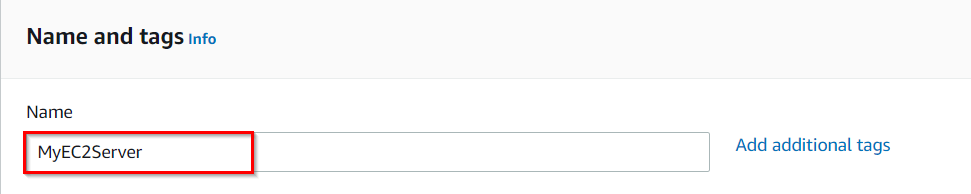
## ****Task 1: Sign in to AWS Management Console****

1. Click on the **Open Console**button, and you will get redirected to AWS Console in a new browser tab.
2. On the AWS sign-in page,
   * Leave the Account ID as default. Never edit/remove the 12 digit Account ID present in the AWS Console. otherwise, you cannot proceed with the lab.
   * Now copy your **User Name** and **Password** in the Lab Console to the **IAM Username and Password** in AWS Console and click on the **Sign in** button.
3. Once Signed In to the AWS Management Console, Make the default AWS Region as **US East (N. Virginia) us-east-1.**

## ****Task 2: Launching an EC2 Instance****

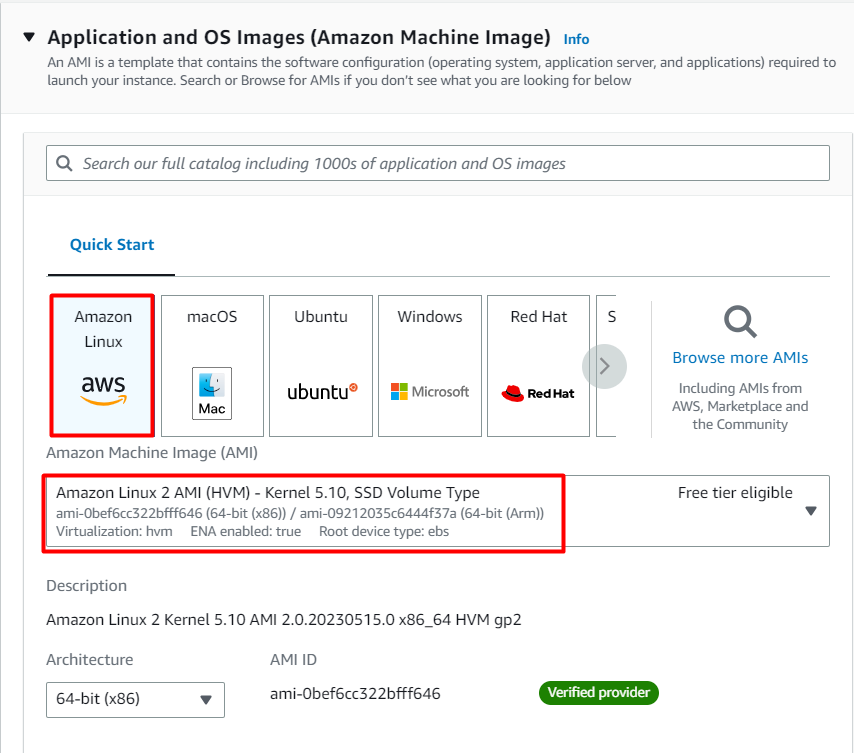
In this task, we are going to launch an EC2 Instance that will be used for checking various features in CloudWatch.

1. Make sure you are in the **N.Virginia** Region.
2. Navigate to **EC2** by clicking on the **Services** menu in the top, then click on **EC2** in the **Compute** section.
3. Navigate to **Instances** from the left side menu and click on **Launch instances**button.
4. Name : Enter **MyEC2Server**

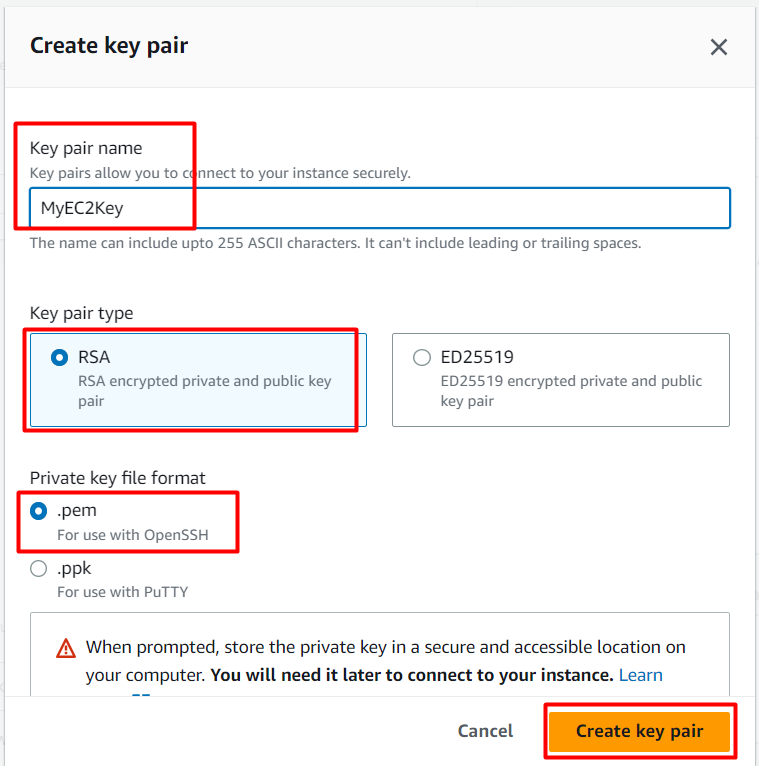
****

1. **For Amazon Machine Image (AMI):** Select **Amazon Linux** and the select **Amazon Linux 2 AMI** from the drop-down.

**Note: if there are two AMI's present for Amazon Linux 2 AMI, choose any of them.**

****

1. For Instance Type: Select **t2.micro**
2. **For Key pair:** Select **Create a new key pair** Button
   * Key pair name: **MyEC2Key**
   * Key pair type: **RSA**
   * Private key file format: **.pem**
3. Select **Create key pair** Button.



     9. In Network Settings Click on **Edit** button:

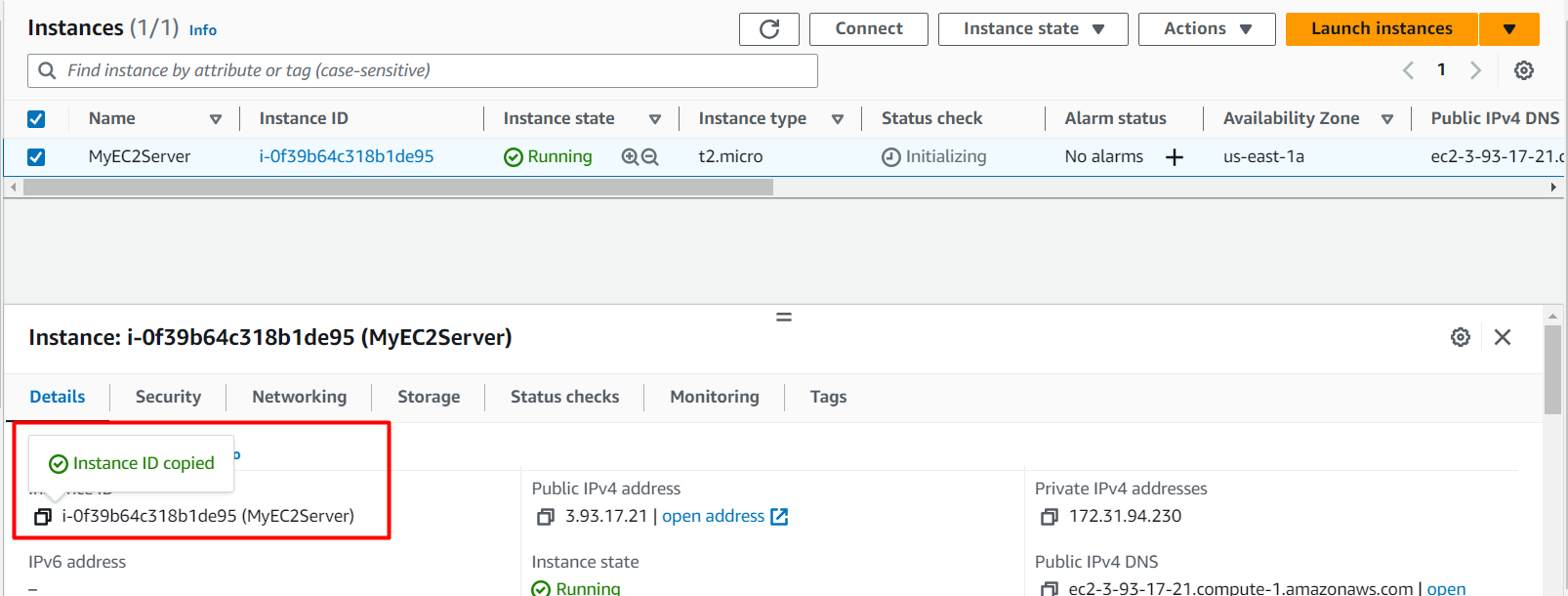
* Auto-assign public IP: **Enable**
* Select **Create new Security group**
* Security group name : Enter **MyEC2Server\_SG**
* Description : Enter **Security Group to allow traffic to EC2**
* To add **SSH** :
  + Choose Type: Select **SSH**
  + Source: Select **Anywhere**

    10. Keep Rest the things as Default and Click on **Launch Instance** Button.

    11. Select **View all Instances** to View the Instance you created.

12. **Launch Status:**Your instance is now launching. Click on the instance ID and wait for complete initialization of the instance (until the status changes to running).

**Note:** Select the instance andCopy the Instance-ID and save it for later, we need to search the metrics in CloudWatch based on this.



## ****Task 3 : SSH into EC2 Instance and install necessary Softwares****

1. Folow the instructions provided in [/labs/support-document/ssh-into-ec-instance](https://www.whizlabs.com/labs/support-document/ssh-into-ec-instance) to SSH into the EC2 instance you created.
2. Once you are logged into the EC2 instance, switch to root user.

sudo su

1. Update :

yum update -y

1. Stress Tool : Amazon Linux 2 AMI does not have the stress tool installed by default, we will need to install some packages

sudo amazon-linux-extras install epel -y

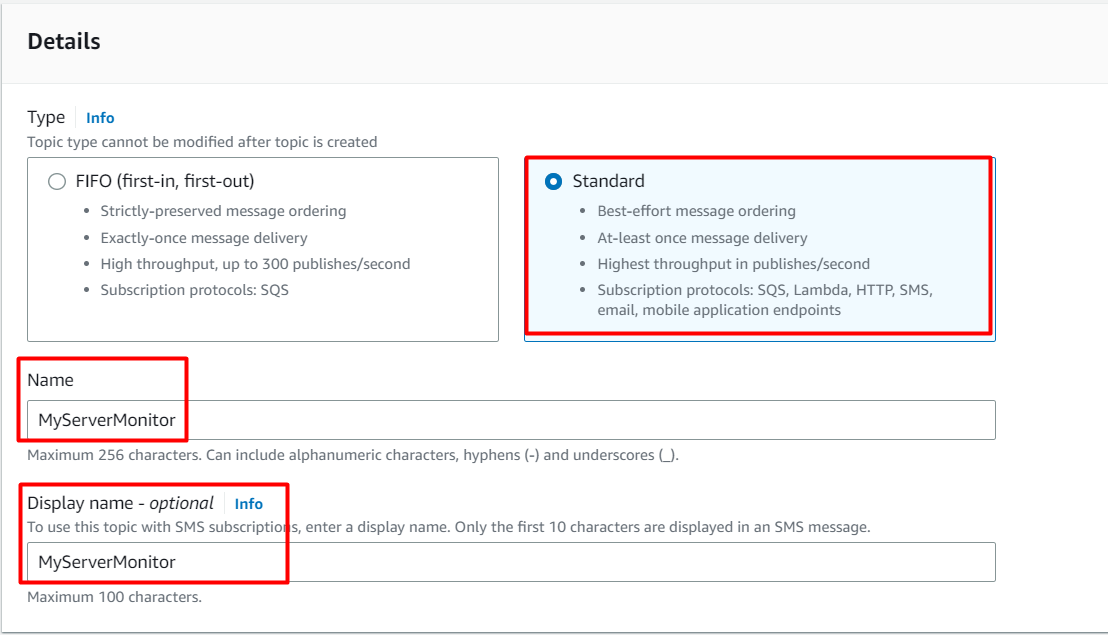
yum install stress -y

1. Stress tool will be used for simulating EC2 metrics. Once we create the CloudWatch Alarm, we shall come back to SSH and trigger **CPUUtilization** using it.

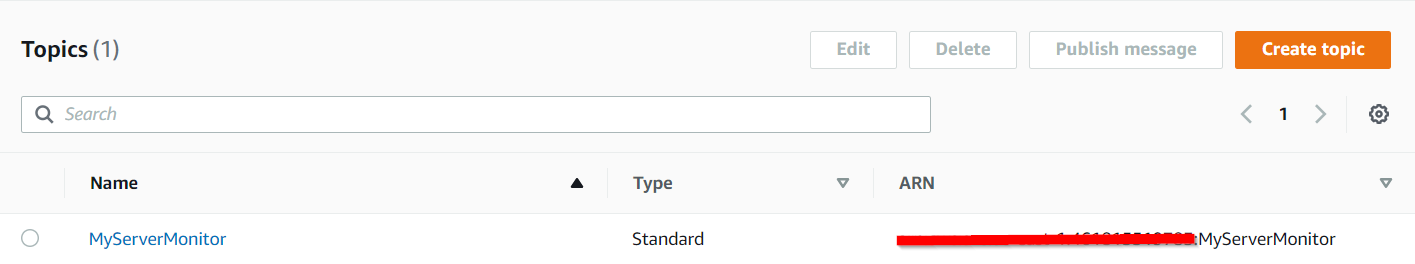
## ****Task 4: Create SNS Topic****

In this task, we are going to create a SNS Topic.

1. Make sure you are in the **N.Virginia** Region.
2. Navigate to **Simple Notification Service** by clicking on the **Services** menu available under the **Application Integration** section.
3. Click on **Topics** in the left panel and then click on **Create topic**button.
4. Under **Details**:
   * Type: Select **Standard**
   * Name: Enter **MyServerMonitor**
   * Display name: Enter **MyServerMonitor**

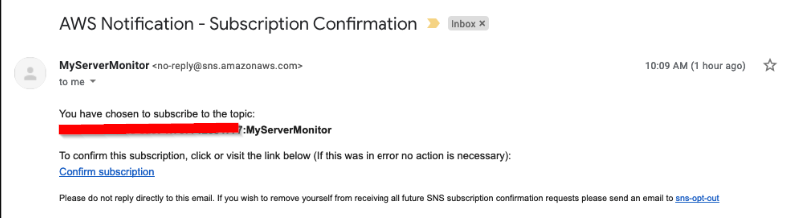
****

     5. Leave other options as default and click on **Create topic**button. A SNS topic will be created.

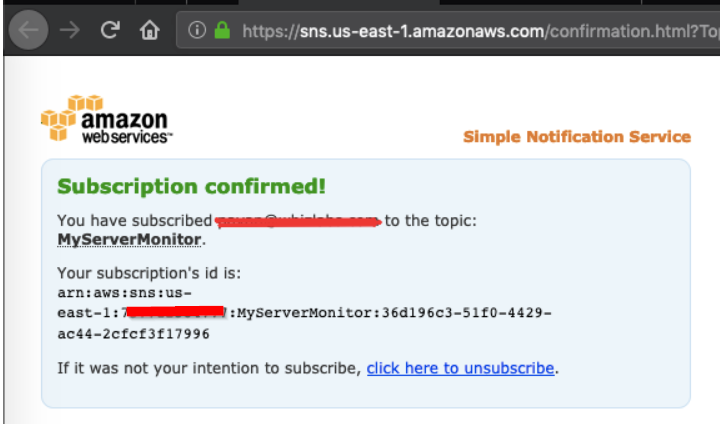


## ****Task 5: Subscribe to an SNS Topic****

1. Once SNS topic is created, click on SNS topic **MyServerMonitor.**
2. Click on **Create subscription**button.
3. Under Details:
   * Protocol : Select **Email**
   * Endpoint : Enter your email address
   * **Note:**Make sure you give proper email address as this is where your notification will be delivered.
4. You will receive a subscription confirmation to your email address



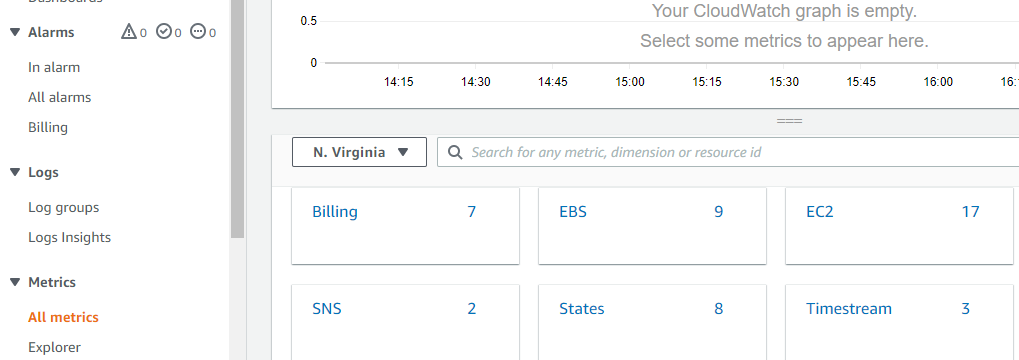
1. Click on **Confirm subscription**.



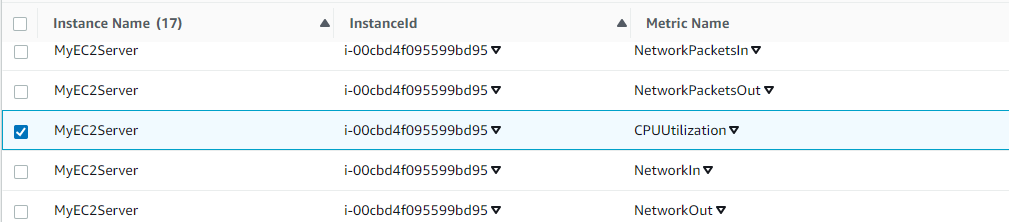
1. Your email address is now subscribed to SNS Topic **MyServerMonitor**.

## ****Task 6: Using CloudWatch to Check EC2 CPU Utilization Metrics in CloudWatch Metrics****

1. Navigate to **CloudWatch** by clicking on the **Services** menu available under the **Management & Governance** section.
2. Click on **All metrics** under **Metrics** in the Left Panel.
3. You should be able to see **EC2** under **All Metrics.** If EC2 is not visible, please wait for 5-10 minutes, CloudWatch usually takes around 5-10 minutes after the creation of EC2 to start fetching metric details.



1. Click on **EC2**. Select **Per-Instance Metrics.**
2. Here you can see various metrics. Select the CPUUtilization metric to see the graph.

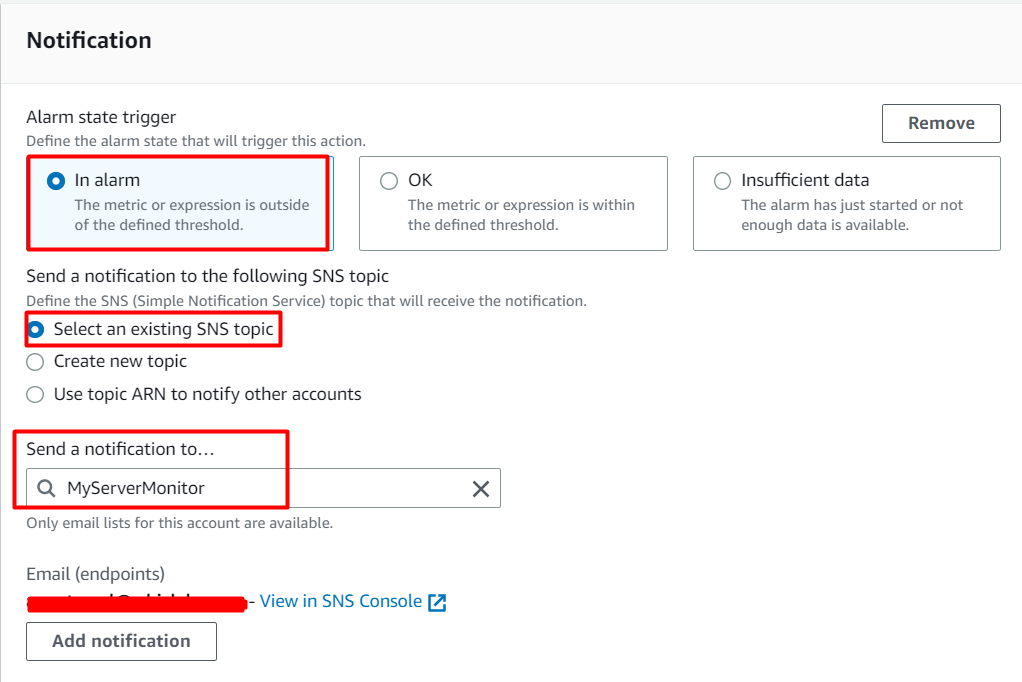


1. Now at the top of the screen, you can see the CPU Utilization graph (which is at zero since we have not stressed the CPU yet).

## ****Task 7: Create CloudWatch Alarm****

CloudWatch alarms are used to watch a single CloudWatch metric or the result of a math expression based on CloudWatch metrics.

1. Click on **In alarms** under **Alarms** in the left panel of the CloudWatch dashboard.
2. Click on **Create alarm** available on the top right corner.
3. In the **Specify metric and conditions** page:
   * Click on **Select metric**. It will open the **Select Metrics** page.
   * Scroll down and Select **EC2**.
   * Select **Per-Instance Metrics**
   * Enter your EC2 **Instance-ID** in the search bar to get metrics for **MyEC2Server**
   * Choose the **CPU Utilization** metric.
   * Click on **Select metric** button.
4. Now, configure the alarm with the following details:
   * Under **Metrics**
     + Period: Select **1 Minute**
   * Under **Conditions**
     + Threshold type: Choose **Static**
     + Whenever CPUUtilization is…: Choose **Greater**
     + than:Enter **30**
   * Leave other values as **default** and click on **Next**button.
5. In **Configure actions** page:
   * Under **Notification**
     + Alarm state trigger: Choose **In Alarm**
     + Select an SNS topic: Choose **Select an existing SNS topic**
     + Send a notification to… : Choose **MyServerMonitor** SNS topic which was created earlier.



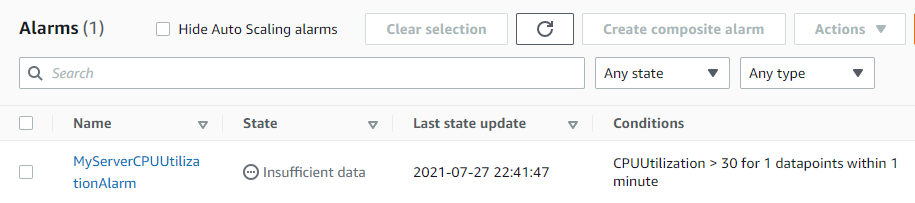
* Leave other fields as default. Click on **Next**button.

6. **In the Add a description** page, (under Name and Description):

* Name: Enter the Name **MyServerCPUUtilizationAlarm**
* Click on **Next**button.

     7. A preview of the Alarm will be shown. Scroll down and click on **Create alarm**button.

     8. A new CloudWatch Alarm is now created.



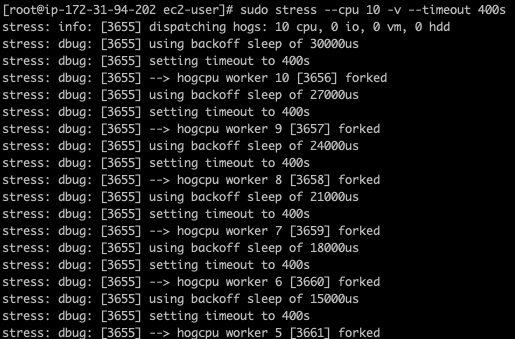
* Whenever the CPU Utilization goes above **30** for **more than 1 minute**, an SNS Notification will be triggered and you will receive an email.

## ****Task 8: Testing CloudWatch Alarm by Stressing CPU Utilization****

1. SSH back into the EC2 instance - **MyEC2Server**.
2. The stress tool has already been installed. Lets run a command to increase the CPU Utilization manually.

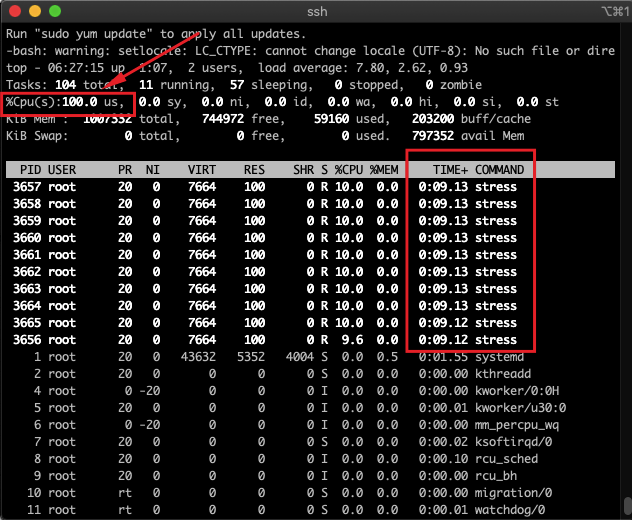
sudo stress --cpu 10 -v --timeout 400s

1. This command shall monitor the process created by the stresstool(which we triggered manually). It will run for **6 minutes and 40 seconds**. It will monitor CPU utilization, which should remain very near 100% for that amount of time.



1. Open another Terminal on your local machine and SSH back in EC2 instance - **MyEC2Server**.
2. Run this command to see the CPU utilization if you are a MAC or Linux User. For Windows User, you can navigate to Task manager.

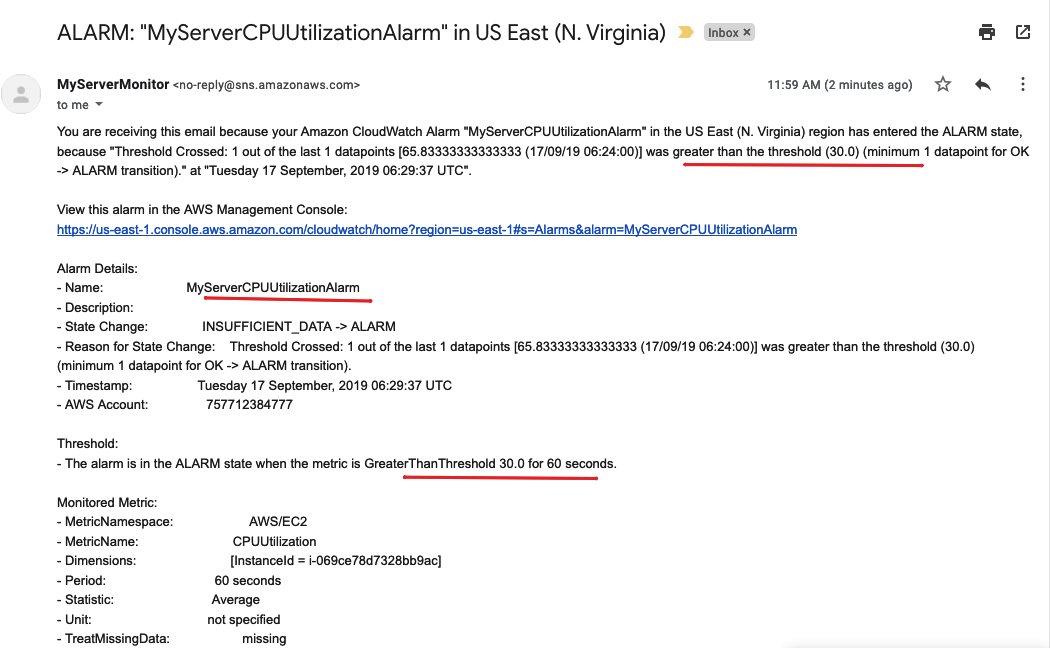
top



1. You can now see that **%Cpu(s)**is **100**. By running this stress command, we have manually increased the CPU utilization of the EC2 Instance.
2. After 400 Seconds, the %Cpu will reduce back to **0**.

## Task 9 : Checking For an Email from the SNS Topic

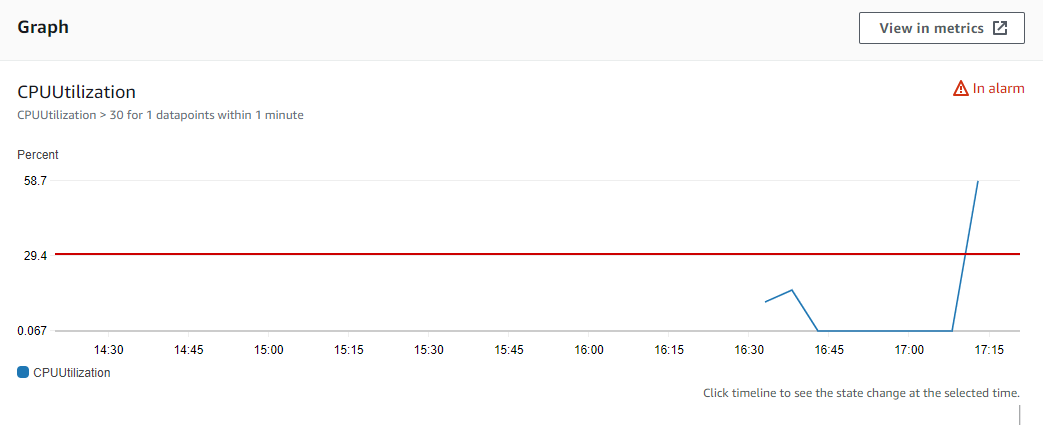
1. Navigate to your mailbox and refresh it. You should see a new email notification for **MyServerCPUUtilizationAlarm**.



1. We can see that mail we received contains details about our CloudWatch Alarm,(name of the alarm, when it was triggered, etc.).

## ****Task 10: Checking the CloudWatch Alarm Graph****

1. Navigate back to CloudWatch page, Click on Alarms.
2. Click on **MyServerCPUUtilizationAlarm**.
3. On the Graph, you can see places where CPUUtilization has gone above the 30% threshold.



1. We can trigger CPUUtilization multiple times to see the spike on the graph.
2. You have successfully triggered a CloudWatch Alarm for CPUUtilzation.

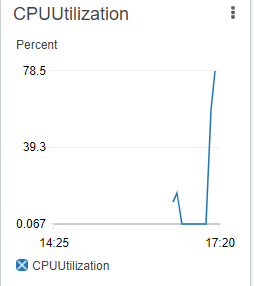
## ****Task 11: Create a CloudWatch Dashboard****

We can create a simple Cloudwatch dashboard to see the CPUUtilization and various other metric widgets.

1. Click on Dashboard in the left panel of the CloudWatch page.
2. Click on **Create dashboard**button.
   * Dashboard name: Enter **MyEC2ServerDashboard**
   * Click on **Create dashboard**
   * Add widget: Select **Line** Graph.
   * Click on **Next**button.
   * Select **Metrics**. Click on **Next**button.
   * On the next page, Choose **EC2** under the **Metrics** tab. Choose **Per-Instance Metrics**.
   * In the search bar, **enter your EC2 Instance ID**. Select **CPUUtilization**.

* Click on **Create Widget**button.

1. Depending on how many times you triggered the stress command, you will see different spikes in the timeline.



1. Now click on the **Save** button.
2. You can also add multiple Widgets to the same Dashboard by clicking on **Add widget**button.

### ****Do you know?****

CloudWatch offers advanced features such as anomaly detection, which uses machine learning algorithms to automatically identify abnormal behavior in your metrics. This helps you to detect and investigate unusual patterns or potential performance bottlenecks in your resources.

# ****Completion and Conclusion****

1. You have created an EC2 Instance for which CloudWatch Monitoring will be carried out.
2. You have successfully created an Amazon SNS Topic used by CloudWatch.
3. You have successfully subscribed to SNS topic using your email address.
4. You have used CloudWatch to see CPUUtilization Metrics using CloudWatch Metrics.
5. You have successfully created and triggered a CloudWatch Alarm based on the CPUUtilzation Metric.