Managing Services - Monitoring

> **Note**
>
> All labs rely on previous courseware and lab information.

Objectives

In this lab, you will:

- Check the status of the service **httpd** to ensure that it is running, and that you can make an http connection to the local host IP address
- You will also learn how to monitor your Amazon Linux 2 EC2 instance
  - Using the Linux **top** command
  - Using **AWS CloudWatch**

Duration

30 minutes

AWS service restrictions

In this lab environment, access to AWS services and service actions might be restricted to the ones that are needed to complete the lab instructions. You might encounter errors if you attempt to access other services or perform actions beyond the ones that are described in this lab.

Accessing the AWS Management Console

1. At the top of these instructions, choose Start Lab to launch your lab.
   A **Start Lab** panel opens, and it displays the lab status.

   > **Tip**: If you need more time to complete the lab, choose the Start Lab button again to restart the timer for the environment.

2. Wait until you see the message *Lab status: ready*, then close the **Start Lab** panel by choosing the **X**.
3. At the top of these instructions, choose AWS.
   This opens the AWS Management Console in a new browser tab. The system will automatically log you in.

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

4. Arrange the AWS Management Console tab so that it displays along side these instructions. Ideally, you will be able to see both browser tabs at the same time so that you can follow the lab steps more easily.

Task 1: Use SSH to connect to an Amazon Linux EC2 instance

In this task, you will connect to a Amazon Linux EC2 instance. You will use an SSH utility to perform all of these operations. The following instructions vary slightly depending on whether you are using Windows or Mac/Linux.

Windows Users: Using SSH to Connect

These instructions are specifically for Windows users. If you are using macOS or Linux, skip to the next section.

5. Select the Details drop-down menu above these instructions you are currently reading, and then select Show. A Credentials window will be presented.
6. Select the **Download PPK** button and save the **labsuser.ppk** file.
   Typically your browser will save it to the Downloads directory.
7. Make a note of the **PublicIP** address.
8. Then exit the Details panel by selecting the **X**.
9. Download **PuTTY** to SSH into the Amazon EC2 instance. If you do not have PuTTY installed on your computer, download it here.
10. Open **putty.exe**
11. Configure PuTTY timeout to keep the PuTTY session open for a longer period of time.:
    - Select **Connection**
    - Set **Seconds between keepalives** to `30`
12. Configure your PuTTY session:
    - Select **Session**
    - **Host Name (or IP address):** Paste the **Public DNS or IPv4 address** of the instance you made a note of earlier.
      Alternatively, return to the EC2 Console and select **Instances**. Check the box next to the instance you want to connect to and in the *Description* tab copy the **IPv4 Public IP** value.
    - Back in PuTTY, in the **Connection** list, expand **SSH**
    - Select **Auth** *(don't expand it)*
    - Select **Browse**
    - Browse to and select the lab#.ppk file that you downloaded
    - Select **Open** to select it
    - Select **Open** again.

13. Select **Yes**, to trust and connect to the host.
14. When prompted **login as**, enter: `ec2-user`
    This will connect you to the EC2 instance.
15. Windows Users: Select here to skip ahead to the next task.

macOS and Linux Users

These instructions are specifically for Mac/Linux users. If you are a Windows user, skip ahead to the next task.

16. Select the Details drop-down menu above these instructions you are currently reading, and then select Show. A Credentials window will be presented.
17. Select the **Download PEM** button and save the **labsuser.pem** file.
18. Make a note of the **PublicIP** address.
19. Then exit the Details panel by selecting the **X**.
20. Open a terminal window, and change directory `cd` to the directory where the *labsuser.pem* file was downloaded. For example, if the *labuser.pem* file was saved to your Downloads directory, run this command:

    ```
    cd ~/Downloads
    ```

21. Change the permissions on the key to be read-only, by running this command:

    ```
    chmod 400 labsuser.pem
    ```

22. Run the below command *(replace **<public-ip>** with the **PublicIP** address you copied earlier)*. Alternatively, return to the EC2 Console and select **Instances**. Check the box next to the instance you want to connect to and in the *Description* tab copy the **IPv4 Public IP** value.:

    ```
    ssh -i labsuser.pem ec2-user@<public-ip>
    ```

23. Type `yes` when prompted to allow the first connection to this remote SSH server.
    Because you are using a key pair for authentication, you will not be prompted for a password.

Task 2: Check the Status of the httpd Service

Httpd is the service for the Apache http server that is installed on your host. This is a lightweight web server like the ones that run your favorite websites (think let's .... say amazon.com) . In this exercise you check the status of the httpd service, and start it using the **systemctl** command and verify the service is working.

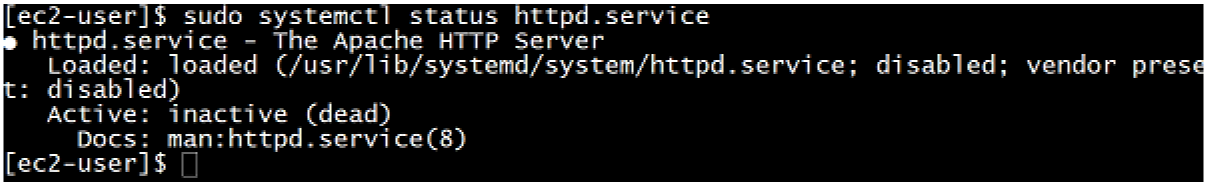
> **Helpful Hint**

> You may have to use **sudo** to complete this exercise if you are not root.

24. Check the status of the httpd service by using the **systemctl** commands as shown below and pressing ENTER

    ```
    sudo systemctl status httpd.service
    ```

    Expected Output:



```
[ec2-user]$ sudo systemctl status httpd.service
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor prese
t: disabled)
   Active: inactive (dead)
     Docs: man:httpd.service(8)
[ec2-user]$ 
```

Figure: The command sudo systemctl status httpd.service and the output says that the httpd service section is inactive (dead).

This indicates that the **httpd** service is **loaded**, which means it is installed and ready to work but is **inactive**.
So the next step is to start it

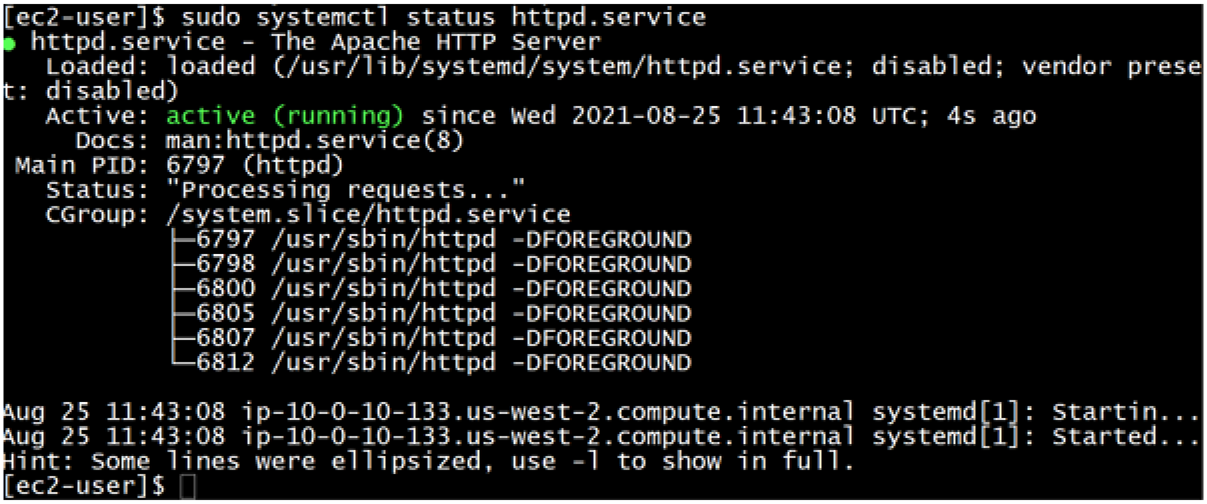
25. Check the status of the httpd service by using the **systemctl** commands as shown below and pressing ENTER

    ```
    sudo systemctl start httpd.service
    ```

26. Check again the status of the httpd service by using the **systemctl** commands as shown below and pressing ENTER

    ```
    sudo systemctl status httpd.service
    ```

    Expected Output



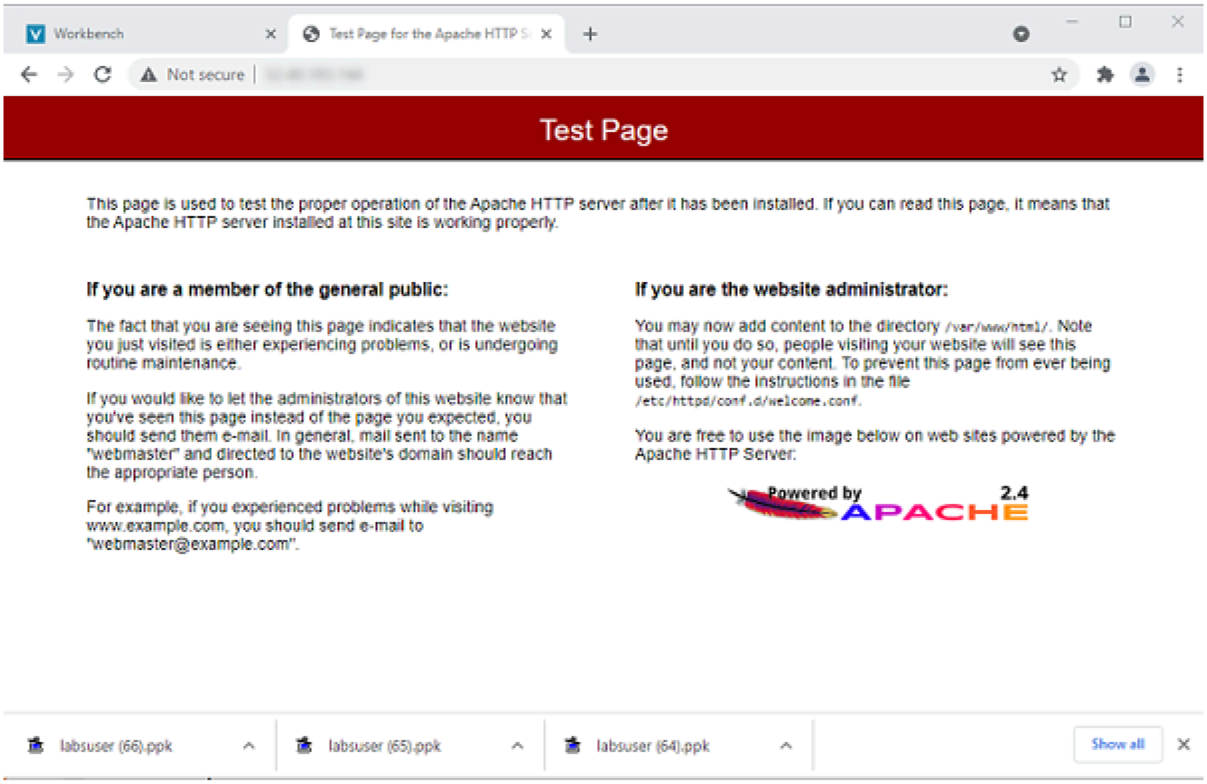
```
[ec2-user]$ sudo systemctl status httpd.service
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor prese
t: disabled)
   Active: active (running) since Wed 2021-08-25 11:43:08 UTC; 4s ago
     Docs: man:httpd.service(8)
 Main PID: 6797 (httpd)
   Status: "Processing requests..."
   CGroup: /system.slice/httpd.service
           ├─6797 /usr/sbin/httpd -DFOREGROUND
           ├─6798 /usr/sbin/httpd -DFOREGROUND
           ├─6800 /usr/sbin/httpd -DFOREGROUND
           ├─6805 /usr/sbin/httpd -DFOREGROUND
           ├─6807 /usr/sbin/httpd -DFOREGROUND
           └─6812 /usr/sbin/httpd -DFOREGROUND

Aug 25 11:43:08 ip-10-0-10-133.us-west-2.compute.internal systemd[1]: Startin...
Aug 25 11:43:08 ip-10-0-10-133.us-west-2.compute.internal systemd[1]: Started...
Hint: Some lines were ellipsized, use -l to show in full.
[ec2-user]$ 
```

Figure: The command sudo systemctl status httpd.service and the output says that the httpd service section is active (running).

27. Now that the **httpd** is running, let's check it works correctly. Open a new tab on your browser and enter : **http://<publicip>** . Replace <publicip> with the public ip that you retrieved at the beginning of the course.

    Expected Output



Test Page

This page is used to test the proper operation of the Apache HTTP server after it has been installed. If you can read this page, it means that the Apache HTTP server installed at this site is working properly.

If you are a member of the general public:

The fact that you are seeing this page indicates that the website you just visited is either experiencing problems, or is undergoing routine maintenance.

If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.

For example, if you experienced problems while visiting www.example.com, you should send e-mail to "webmaster@example.com".

If you are the website administrator:

You may now add content to the directory `/var/www/html/`. Note that until you do so, people visiting your website will see this page, and not your content. To prevent this page from ever being used, follow the instructions in the file `/etc/httpd/conf.d/welcome.conf`.

You are free to use the image below on web sites powered by the Apache HTTP Server:

Powered by APACHE 2.4

Figure: When httpd is successfully running, an Apache Test page will apear with general information about the proper operation of the Apache HTTP server.

28. You can now stop the service by entering the command below and pressing ENTER

```
sudo systemctl stop httpd.service
```

Task 3: Monitoring a Linux EC2 instance

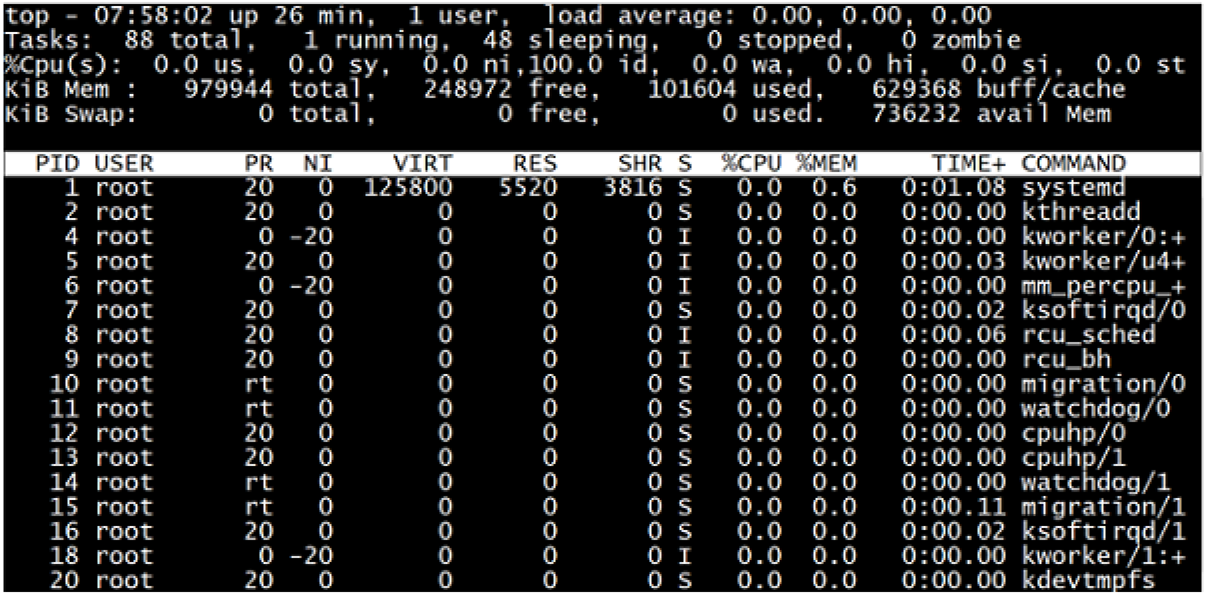
In this exercise you will use Linux commands to monitor the Amazon Linux2 EC2 instance. You will also open the AWS Console and log into CloudWatch to see how this service can provide you with data to monitor your instance.

> **Helpful Hint**
>
> You may have to use **sudo** to complete this exercise if you are not root.

29. Display the list of running processes by entering the command below and pressing ENTER

```
top
```

Expected Output



```
top - 07:58:02 up 26 min,  1 user,  load average: 0.00, 0.00, 0.00
Tasks:  88 total,   1 running,  48 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem :   979944 total,   248972 free,   101604 used,   629368 buff/cache
KiB Swap:        0 total,        0 free,        0 used.   736232 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
    1 root      20   0  125800   5520   3816 S   0.0  0.6   0:01.08 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:+
    5 root      20   0       0      0      0 I   0.0  0.0   0:00.03 kworker/u4+
    6 root       0 -20       0      0      0 I   0.0  0.0   0:00.00 mm_percpu_+
    7 root      20   0       0      0      0 S   0.0  0.0   0:00.02 ksoftirqd/0
    8 root      20   0       0      0      0 I   0.0  0.0   0:00.06 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
   13 root      20   0       0      0      0 S   0.0  0.0   0:00.00 cpuhp/1
   14 root      rt   0       0      0      0 S   0.0  0.0   0:00.00 watchdog/1
   15 root      rt   0       0      0      0 S   0.0  0.0   0:00.11 migration/1
   16 root      20   0       0      0      0 S   0.0  0.0   0:00.02 ksoftirqd/1
   18 root       0 -20       0      0      0 I   0.0  0.0   0:00.00 kworker/1:+
   20 root      20   0       0      0      0 S   0.0  0.0   0:00.00 kdevtmpfs
```

Figure: When running the command top, the output will show the current running processes and resource usages.

Top displays the processes currently running as well as the resource usage like CPU usage and memory usage. Hit **Q** to exit and return to the shell.

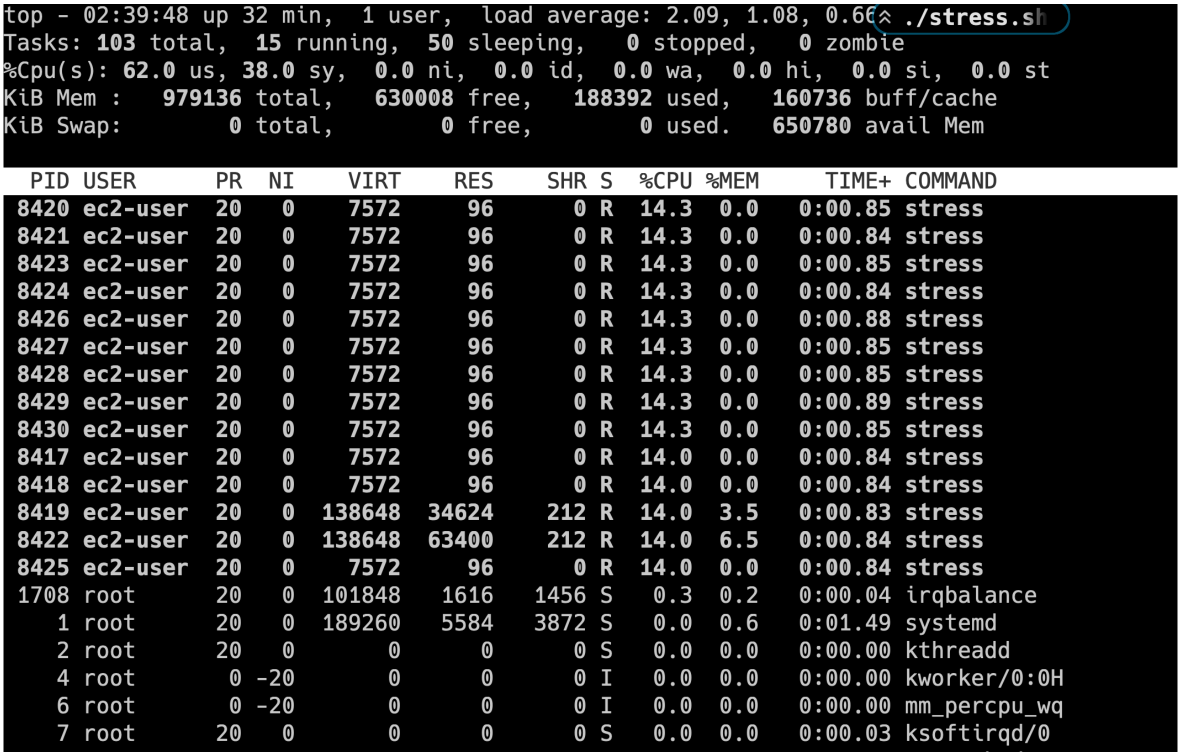
In the next step you are going to run a script that simulates a workload on the CPU.

30. Run the **stress.sh** script that simulates a heavy workload on the EC2 instance by entering the following command and pressing ENTER

    ```
    ./stress.sh & top
    ```

31. As in step 1, display the list of running processes by entering the **top** command and pressing ENTER

    Expected Output



```
top - 02:39:48 up 32 min,  1 user,  load average: 2.09, 1.08, 0.6 ./stress.sh
Tasks: 103 total,  15 running,  50 sleeping,   0 stopped,   0 zombie
%Cpu(s): 62.0 us, 38.0 sy,  0.0 ni,  0.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem :   979136 total,   630008 free,   188392 used,   160736 buff/cache
KiB Swap:        0 total,        0 free,        0 used.   650780 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
 8420 ec2-user  20   0    7572     96      0 R  14.3  0.0   0:00.85 stress
 8421 ec2-user  20   0    7572     96      0 R  14.3  0.0   0:00.84 stress
 8423 ec2-user  20   0    7572     96      0 R  14.3  0.0   0:00.85 stress
 8424 ec2-user  20   0    7572     96      0 R  14.3  0.0   0:00.84 stress
 8426 ec2-user  20   0    7572     96      0 R  14.3  0.0   0:00.88 stress
 8427 ec2-user  20   0    7572     96      0 R  14.3  0.0   0:00.85 stress
 8428 ec2-user  20   0    7572     96      0 R  14.3  0.0   0:00.85 stress
 8429 ec2-user  20   0    7572     96      0 R  14.3  0.0   0:00.89 stress
 8430 ec2-user  20   0    7572     96      0 R  14.3  0.0   0:00.85 stress
 8417 ec2-user  20   0    7572     96      0 R  14.0  0.0   0:00.84 stress
 8418 ec2-user  20   0    7572     96      0 R  14.0  0.0   0:00.84 stress
 8419 ec2-user  20   0  138648  34624    212 R  14.0  3.5   0:00.83 stress
 8422 ec2-user  20   0  138648  63400    212 R  14.0  6.5   0:00.84 stress
 8425 ec2-user  20   0    7572     96      0 R  14.0  0.0   0:00.84 stress
 1708 root      20   0  101848   1616   1456 S   0.3  0.2   0:00.04 irqbalance
    1 root      20   0  189260   5584   3872 S   0.0  0.6   0:01.49 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
    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   0:00.03 ksoftirqd/0
```

Figure: The command prompt shows a high CPU usage after running a script. It shows the user as ec2-user the percentage of CPU at 14-14.3 and the command used was stress.

You can see that the process you just ran has a high CPU usage. The script is designed to run for 6 minutes, before stopping.

In the next steps you open the **AWS Management Console** and start the **AWS CloudWatch** application that will give you better insight into you EC2 instance.

32. On the top right of you screen select the `AWS` button. This displays the **AWS Management Console** in a new tab.
33. In the Search bar on the top of the screen, enter **CloudWatch** and press ENTER.

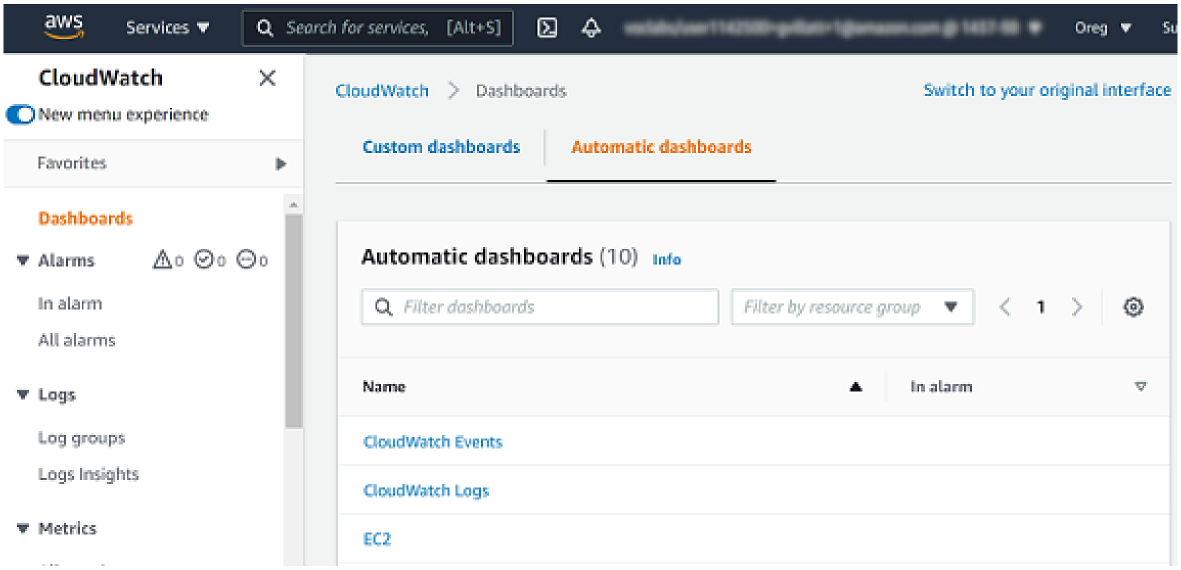
    Expected Output

    The search bar on the AWS console with the word CloudWatch entered.

    The AWS console includes a search bar that you can use to search for services.

34. On the left section of the navigation pane, select **Dashboard**, then select **Automatic dashboards**. In the **Automatic dashboards** list, select **EC2**

    Expected Output



Automatic dashboards (10) Info

Name	In alarm
CloudWatch Events	
CloudWatch Logs	
EC2	

Figure: The CloudWatch dashboards shows CloudWatch Events, CloudWatch Logs, and EC2 as the top three dashboards.

This opens up the **EC2 dashboard** created for you by **AWS**.

Expected Output

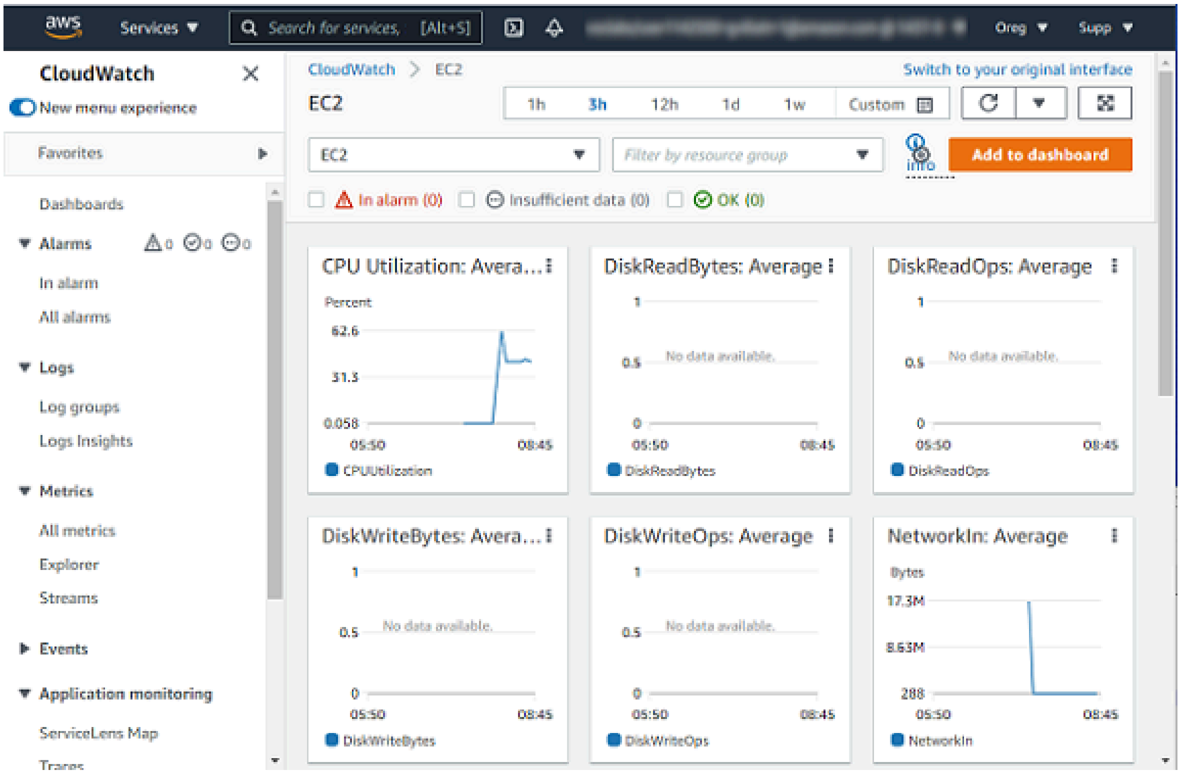


Figure: The graphs shown are CPU Utilization, DiskReadBytes, DiskReadOps, DiskWriteBytes, DiskWriteOps, and NetworkIn for the account's EC2 instances.

You can see that by default the **EC2 CloudWatch dashboard** displays several metrics such as the CPU utilization, Disk reads and writes....

You can see a spike in the CPU utilization that matches the time when you started the stress script earlier.

> **Note**
>
> Dashboards are customizable so you can add or remove widgets,
> reorganize them, customize colors... **AWS CloudWatch** offers many
> more features such as alarms or events triggers that you will discover
> later that makes it a key **AWS Service** to monitor your applications
> in real time. Update the 5 minutes average to 1 second to second review
> updates more quickly.

35. Wait 5 minutes and go back to the AWS CloudWatch dashboard. You see that the CPU utilization dropped

    Expected Output

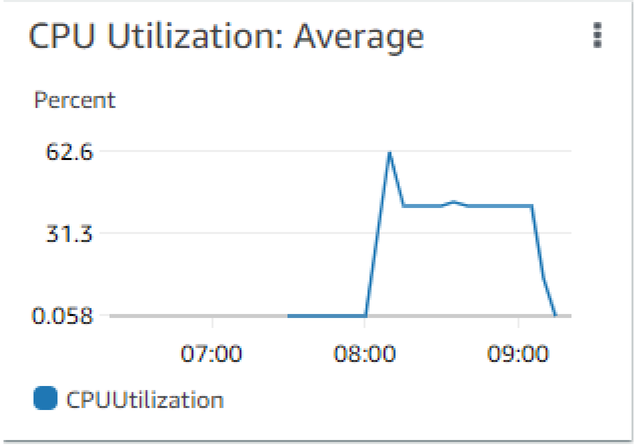


Figure: The graph shown is the CPU Utilization Average. In this average, the highest percent it reaches is 62.6 percent at around 8 minutes and hovers around 33 percent then drops back down right aftger the 9 minute mark.

> **Note**
>
> By default AWS CloudWatch aggregates data for 5 minutes before processing them. This is setup that can be changed

Lab Complete

Congratulations! You have completed the lab.

36. Select End Lab at the top of this page and then select Yes to confirm that you want to end the lab.
    A panel will appear, indicating that "DELETE has been initiated... You may close this message box now."
37. Select the **X** in the top right corner to close the panel.

About the AWS component:

Amazon EC2 provides a wide selection of *instance types* optimized to fit different use cases. Instance types comprise varying combinations of CPU, memory, storage, and networking capacity and give you the flexibility to choose the appropriate mix of resources for your applications. Each instance type includes one or more *instance sizes*, allowing you to scale your resources to the requirements of your target workload.

This lab uses a **t3.micro** instance which should be selected by default. This instance type has 1 virtual CPU and 1 GiB of memory.

Additional Resources

Amazon EC2 Instance Types
Amazon Machine Images (AMI)
Status Checks for Your Instances Amazon EC2 Service Limits
Terminate Your Instance

For more information about AWS Training and Certification, see https://aws.amazon.com/training/.

Your feedback is welcome and appreciated.
If you would like to share any suggestions or corrections, please provide the details in our AWS Training and Certification Contact Form.

© 2022 Amazon Web Services, Inc. and its affiliates. All rights reserved. This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited.