

# AWS Cloud Watch Custom Metrics:

CloudWatch is designed to help users collect and track various metrics, collect and monitor log files, and set alarms. CloudWatch provides a comprehensive set of tools for monitoring AWS resources, applications, and services in real-time.

1. **Metrics:** CloudWatch collects data in the form of metrics, which represent various aspects of your AWS resources and applications. Metrics could include CPU utilization, network traffic, disk I/O, and more
2. **Alarms:** Users can set alarms on metrics to receive notifications or take automated actions when certain thresholds are breached. This allows for proactive monitoring and response to potential issues.
3. **Logs:** CloudWatch Logs allow you to collect, store, and monitor log files from AWS resources or applications. This can be useful for troubleshooting and analysing system behaviour.
4. **Events:** CloudWatch Events enables you to respond to state changes in your AWS resources. You can create rules that match events and take automated actions.
5. **Dashboards:** CloudWatch Dashboards provide a customizable view of your metrics and logs, allowing you to create visualizations that are meaningful for your specific use case.
6. **Insights:** CloudWatch Logs Insights allows you to search and analyse log data using queries to gain insights into your applications and systems.
7. **Custom Metrics:** In addition to AWS service metrics, CloudWatch allows you to publish your own custom metrics.

**In Amazon CloudWatch, custom metrics allow you to monitor and collect data from your own applications, services, and resources. While CloudWatch provides a variety of pre-defined metrics for AWS services, you may have specific metrics relevant to your application or infrastructure that you want to monitor. Custom metrics fill this gap by allowing you to send your own data to CloudWatch for monitoring and analysis**

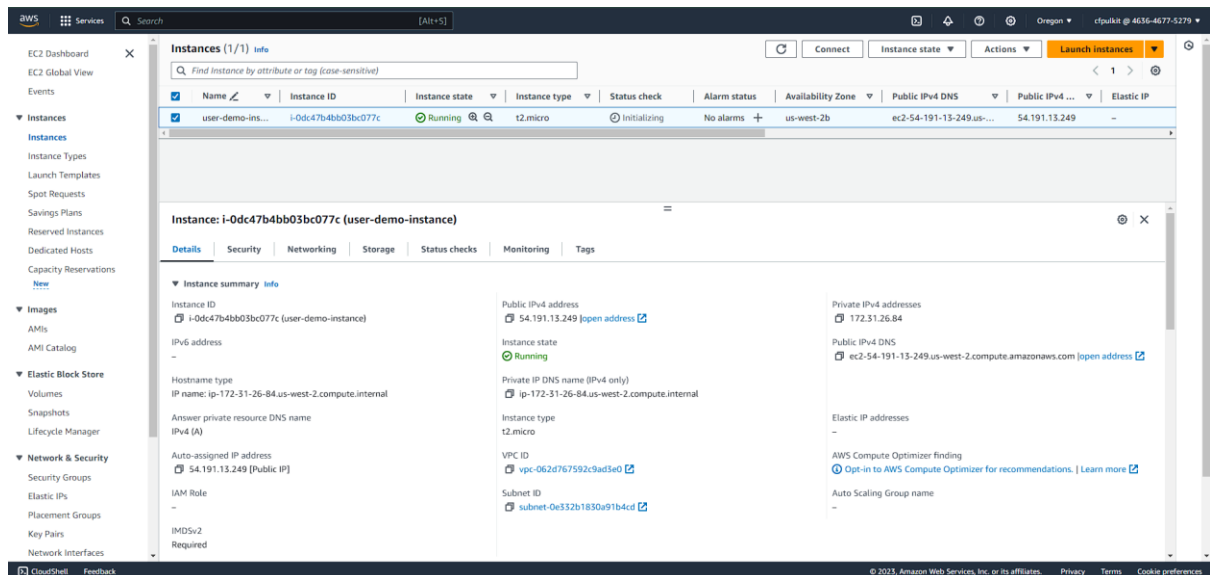
Here's a basic overview of how you can work with custom metrics in CloudWatch:

1. **Publishing Custom Metrics:** You can publish custom metrics using the CloudWatch API or the AWS Command Line Interface (CLI). Metrics are typically associated with a specific namespace and have a unique name within that namespace.
2. **Namespaces:** A namespace is a container for CloudWatch metrics. It is a way to group related metrics. You can use an existing namespace or create a new one for your custom metrics.
3. **Dimensions:** Dimensions are name/value pairs associated with a metric. They are used to uniquely identify a metric within a namespace. For example, if you are monitoring the performance of instances in different regions, you might use dimensions like "Region" and "InstanceId."
4. **Units and Timestamps:** When publishing custom metrics, you need to specify the unit of the data (e.g., Bytes, Count) and the timestamp for the data point.
5. **Setting Up Alarms:** Once you have custom metrics in CloudWatch, you can set up alarms based on these metrics to receive notifications when certain thresholds are breached.
6. **Viewing and Analysing Metrics:** You can view custom metrics on the CloudWatch console, create custom dashboards, and use CloudWatch API calls to retrieve metric data for analysis.
7. **Pricing:** Be mindful of CloudWatch pricing as custom metrics may incur additional charges. It's important to understand the cost implications of publishing and storing custom metric data.

# How to set Custom Metrics in Cloud Watch:

## Step 1: Launch Instance

1. Log in to Console and launch an EC2 instance with Amazon Linux.
2. Just launch a basic instance in which you can SSH easily.



## Step 2: Install Pearl and Memory Related Commands

1. Now SSH into it using command prompt or by directly connecting it on the browser.
2. Then write the command to get into the root user which is '**sudo su**'
3. Then to check the amount of free storage write '**free -h**'
4. And if you want to check the disk storage then write '**df -h**'

```
root@ip-172-31-26-84:/home, x + v
[ec2-user@ip-172-31-26-84 ~]$ sudo su
[root@ip-172-31-26-84 ec2-user]# free -h
              total        used        free      shared  buff/cache   available
Mem:           949Mi       152Mi       581Mi       2.0Mi       215Mi       657Mi
Swap:            0B           0B           0B
[root@ip-172-31-26-84 ec2-user]# df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        4.0M   0  4.0M   0% /dev
tmpfs           475M   0  475M   0% /dev/shm
tmpfs           190M  2.9M  188M   2% /run
/dev/xvda1       8.0G  1.6G  6.5G  20% /
tmpfs           475M   0  475M   0% /tmp
/dev/xvda128     10M  1.3M  8.7M  13% /boot/efi
tmpfs           95M   0   95M   0% /run/user/1000
[root@ip-172-31-26-84 ec2-user]#
```

5. So, to use AWS Cloud Watch as custom metrics you need to install pearl on your instance because the script to do so is written in perl. The command to install pearl is
- ```
'sudo yum install -y perl-Switch perl-DateTime perl-Sys-Syslog perl-LWP-Protocol-https perl-Digest-SHA.x86_64'
```

```
[root@ip-172-31-26-84 ec2-user]# sudo yum install -y perl-Switch perl-DateTime perl-Sys-Syslog perl-LWP-Protocol-https perl-Digest-SHA.x86_64
Last metadata expiration check: 5:06:36 ago on Tue Dec 12 12:46:38 2023.
Dependencies resolved.
```

| Package                        | Architecture | Version                 | Repository  | Size  |
|--------------------------------|--------------|-------------------------|-------------|-------|
| Installing:                    |              |                         |             |       |
| perl-DateTime                  | x86_64       | 2.11.50-2.amzn2023.0.2  | amazonlinux | 128 k |
| perl-Digest-SHA                | x86_64       | 1.6.02-459.amzn2023.0.2 | amazonlinux | 62 k  |
| perl-LWP-Protocol-https        | noarch       | 6.10-2.amzn2023.0.2     | amazonlinux | 22 k  |
| perl-Switch                    | noarch       | 2.17-21.amzn2023.0.2    | amazonlinux | 26 k  |
| perl-Sys-Syslog                | x86_64       | 0.36-459.amzn2023.0.2   | amazonlinux | 47 k  |
| Installing dependencies:       |              |                         |             |       |
| mailcap                        | noarch       | 2.1.40-3.amzn2023.0.3   | amazonlinux | 33 k  |
| perl-AutoLoader                | noarch       | 5.74-477.amzn2023.0.5   | amazonlinux | 22 k  |
| perl-B                         | x86_64       | 1.00-477.amzn2023.0.5   | amazonlinux | 180 k |
| perl-B-Hooks-EndOfScope        | noarch       | 0.20-13.amzn2023.0.2    | amazonlinux | 38 k  |
| perl-Class-Data-Inheritable    | noarch       | 0.08-37.amzn2023.0.2    | amazonlinux | 14 k  |
| perl-Class-Inspector           | noarch       | 1.36-5.amzn2023.0.2     | amazonlinux | 31 k  |
| perl-Class-Method-Modifiers    | noarch       | 2.13-6.amzn2023.0.2     | amazonlinux | 44 k  |
| perl-Class-Singleton           | noarch       | 1.6-2.amzn2023.0.2      | amazonlinux | 26 k  |
| perl-Compress-Raw-Gzip2        | x86_64       | 2.101-3.amzn2023.0.2    | amazonlinux | 34 k  |
| perl-Compress-Raw-Zlib         | x86_64       | 2.101-3.amzn2023.0.2    | amazonlinux | 60 k  |
| perl-Data-Dump                 | noarch       | 1.23-16.amzn2023.0.2    | amazonlinux | 33 k  |
| perl-Data-Dumper               | x86_64       | 2.170-460.amzn2023.0.2  | amazonlinux | 55 k  |
| perl-Data-OptList              | noarch       | 0.110-15.amzn2023.0.2   | amazonlinux | 27 k  |
| perl-Date-ISO8601              | noarch       | 0.985-11.amzn2023.0.2   | amazonlinux | 21 k  |
| perl-DateTime-Locale           | noarch       | 1.32-1.amzn2023.0.2     | amazonlinux | 2.8 k |
| perl-DateTime-TimeZone         | noarch       | 2.51-1.amzn2023.0.2     | amazonlinux | 359 k |
| perl-DateTime-TimeZone-SystemV | noarch       | 0.010-12.amzn2023.0.2   | amazonlinux | 24 k  |
| perl-DateTime-TimeZone-Tzfile  | noarch       | 0.011-12.amzn2023.0.2   | amazonlinux | 21 k  |
| perl-Devel-CallChecker         | x86_64       | 0.006-12.amzn2023.0.2   | amazonlinux | 24 k  |
| perl-Devel-Caller              | x86_64       | 2.06-24.amzn2023.0.2    | amazonlinux | 20 k  |
| perl-Devel-LexAlias            | x86_64       | 0.05-25.amzn2023.0.2    | amazonlinux | 17 k  |
| perl-Devel-StackTrace          | noarch       | 1.2.04-8.amzn2023.0.2   | amazonlinux | 31 k  |
| perl-Digest                    | noarch       | 1.20-1.amzn2023.0.2     | amazonlinux | 26 k  |
| perl-Digest-MD5                | noarch       | 1.03-27.amzn2023.0.2    | amazonlinux | 17 k  |

```
Complete!
[root@ip-172-31-26-84 ec2-user]# |
```

6. After installing pearl, now you need to create a folder to install the directories into it. And are creating the folder go into that folder. The command for creating a folder is **mkdir /cloudwatch** (cloudwatch is the folder name) and to go into the folder write **'cd /cloudwatch**. Here you can see you are in the cloudwatch folder.

```
Complete!
[root@ip-172-31-26-84 ec2-user]# mkdir /cloudwatch
[root@ip-172-31-26-84 ec2-user]# cd /cloudwatch
[root@ip-172-31-26-84 cloudwatch]# |
```

7. Now you need to download the script and the command or path to download that is **wget <https://aws-cloudwatch.s3.amazonaws.com/downloads/CloudWatchMonitoringScripts-1.2.2.zip>** (Wget is a networking command-line tool that lets you download files and interact with REST APIs)

```
[root@ip-172-31-26-84 cloudwatch]# wget https://aws-cloudwatch.s3.amazonaws.com/downloads/CloudWatchMonitoringScripts-1.2.2.zip
--2023-12-12 18:04:49-- https://aws-cloudwatch.s3.amazonaws.com/downloads/CloudWatchMonitoringScripts-1.2.2.zip
Resolving aws-cloudwatch.s3.amazonaws.com (aws-cloudwatch.s3.amazonaws.com)... 52.216.56.145, 52.217.170.185, 52.217.234.241, ...
Connecting to aws-cloudwatch.s3.amazonaws.com (aws-cloudwatch.s3.amazonaws.com)[52.216.56.145]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 24225 (24K) [application/zip]
Saving to: 'CloudWatchMonitoringScripts-1.2.2.zip'

CloudWatchMonitoringScripts-1.2.2.zip 100%[=====] 23.66K --.-KB/s in 0.06s

2023-12-12 18:04:50 (381 KB/s) - 'CloudWatchMonitoringScripts-1.2.2.zip' saved [24225/24225]

[root@ip-172-31-26-84 cloudwatch]# |
```

8. Now if you run command **'ll'** then you can see a zip folder inside cloudwatch.

```
[root@ip-172-31-26-84 cloudwatch]# ll
total 24
-rw-r--r--. 1 root root 24225 Mar 26 2018 CloudWatchMonitoringScripts-1.2.2.zip
[root@ip-172-31-26-84 cloudwatch]# |
```

9. If you see the script that you downloaded is in .zip file and you need to unzip it to move forward. So just type this command to unzip it

**'unzip CloudWatchMonitoringScripts-1.2.2.zip'**

Now if you 'll' into cloudwatch folder you'll see two files.

```
[root@ip-172-31-26-84 cloudwatch]# unzip CloudWatchMonitoringScripts-1.2.2.zip
Archive: CloudWatchMonitoringScripts-1.2.2.zip
  extracting: aws-scripts-mon/awscreds.template
   inflating: aws-scripts-mon/AwsSignatureV4.pm
   inflating: aws-scripts-mon/CloudWatchClient.pm
   inflating: aws-scripts-mon/LICENSE.txt
   inflating: aws-scripts-mon/mon-get-instance-stats.pl
   inflating: aws-scripts-mon/mon-put-instance-data.pl
   inflating: aws-scripts-mon/NOTICE.txt
[root@ip-172-31-26-84 cloudwatch]# ll
total 24
-rw-r--r--. 1 root root 24225 Mar 26 2018 CloudWatchMonitoringScripts-1.2.2.zip
drwxr-xr-x. 2 root root 185 Dec 12 18:10 aws-scripts-mon
[root@ip-172-31-26-84 cloudwatch]# |
```

10. This zipped file is not of your use so just remove it. The command to remove this file is 'rm' and the file name

**'rm CloudWatchMonitoringScripts-1.2.2.zip'**

This is how the file can be removed easily.

```
[root@ip-172-31-26-84 cloudwatch]# rm CloudWatchMonitoringScripts-1.2.2.zip
rm: remove regular file 'CloudWatchMonitoringScripts-1.2.2.zip'? yes
[root@ip-172-31-26-84 cloudwatch]# ll
total 0
drwxr-xr-x. 2 root root 185 Dec 12 18:10 aws-scripts-mon
[root@ip-172-31-26-84 cloudwatch]# |
```

11. Now you need to go into this file aws-scripts-mon. For that you need to 'cd' into it.

**'cd aws-scripts-mon'**

```
[root@ip-172-31-26-84 cloudwatch]# cd aws-scripts-mon
[root@ip-172-31-26-84 aws-scripts-mon]# |
```

12. So, now if you list its content by using 'll' command you can this much of content was in this folder. So, now you need to rename awscreds.template to awscreds.conf

```
[root@ip-172-31-26-84 aws-scripts-mon]# ll
total 96
-r--r--r--. 1 root root 17021 Mar 26 2018 AwsSignatureV4.pm
-r--r--r--. 1 root root 22519 Mar 26 2018 CloudWatchClient.pm
-rw-r--r--. 1 root root 9124 Mar 26 2018 LICENSE.txt
-rw-r--r--. 1 root root 138 Mar 26 2018 NOTICE.txt
-rw-r--r--. 1 root root 30 Mar 26 2018 awscreds.template
-rwxr-xr-x. 1 root root 9739 Mar 26 2018 mon-get-instance-stats.pl
-rwxr-xr-x. 1 root root 18144 Mar 26 2018 mon-put-instance-data.pl
[root@ip-172-31-26-84 aws-scripts-mon]# |
```

13. You need to run copy command which will change its name to what you need.

**'cp awscreds.template awscreds.conf'**

```
[root@ip-172-31-26-84 aws-scripts-mon]# cp awscreds.template awscreds.conf
[root@ip-172-31-26-84 aws-scripts-mon]# ll
total 100
-r--r--r--. 1 root root 17021 Mar 26 2018 AwsSignatureV4.pm
-r--r--r--. 1 root root 22519 Mar 26 2018 CloudWatchClient.pm
-rw-r--r--. 1 root root 9124 Mar 26 2018 LICENSE.txt
-rw-r--r--. 1 root root 138 Mar 26 2018 NOTICE.txt
-rw-r--r--. 1 root root 30 Dec 12 18:24 awscreds.conf
-rw-r--r--. 1 root root 30 Mar 26 2018 awscreds.template
-rwxr-xr-x. 1 root root 9739 Mar 26 2018 mon-get-instance-stats.pl
-rwxr-xr-x. 1 root root 18144 Mar 26 2018 mon-put-instance-data.pl
[root@ip-172-31-26-84 aws-scripts-mon]# |
```

14. Now you need to go into awscreds.conf folder where you need to put your Access key and Secret access key of your AWS account.

Type the command **"vim awscreds.conf"**

15. As you have saved your access keys into 'awscreds.conf' folder. Now you have to run this command to send your data to AWS Cloud Watch.

**"/mon-put-instance-data.pl --mem-used-incl-cache-buff --mem-util --mem-used --mem-avail"**

```
[ec2-user@ip-172-31-22-252 aws-scripts-mon]$ ./mon-put-instance-data.pl --mem-used-incl-cache-buff --mem-util --mem-used --mem-avail
Successfully reported metrics to CloudWatch. Reference Id: 499a618d-1149-4642-aed8-3932e6976170
[ec2-user@ip-172-31-22-252 aws-scripts-mon]$ |
```

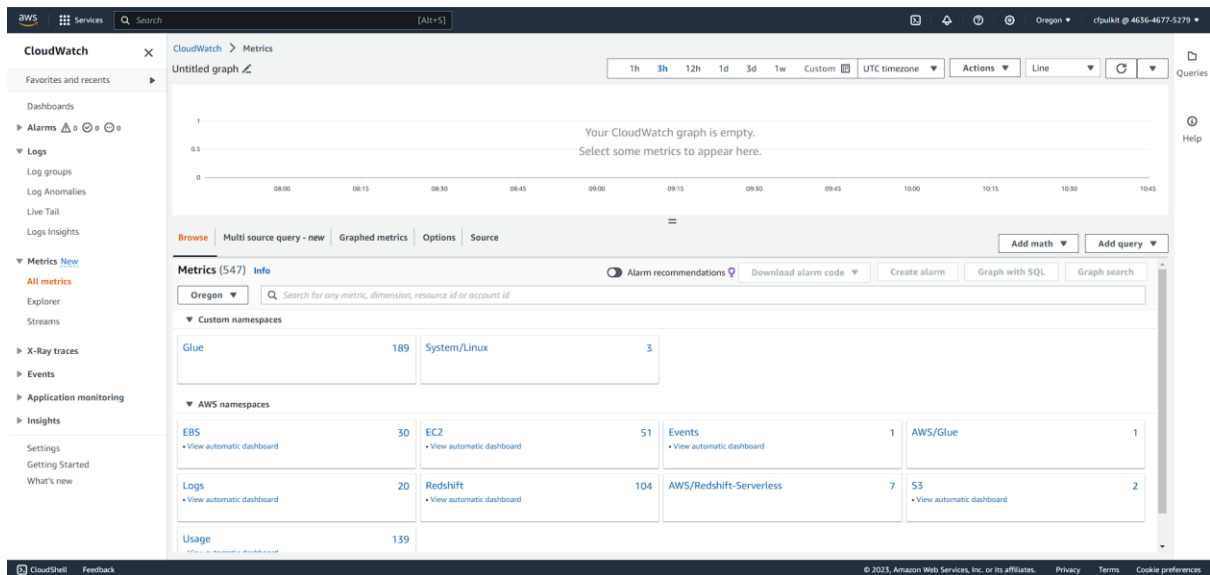
16. To see your metrics, you need to navigate to AWS Cloud Watch on the console. Then click on Metrics > All metrics. You will this kind of interface, now you need to copy the instance ID of the EC2 instance and paste it in the search bar (you can find the search bar in the middle of your screen.)

▼ Metrics [New](#)

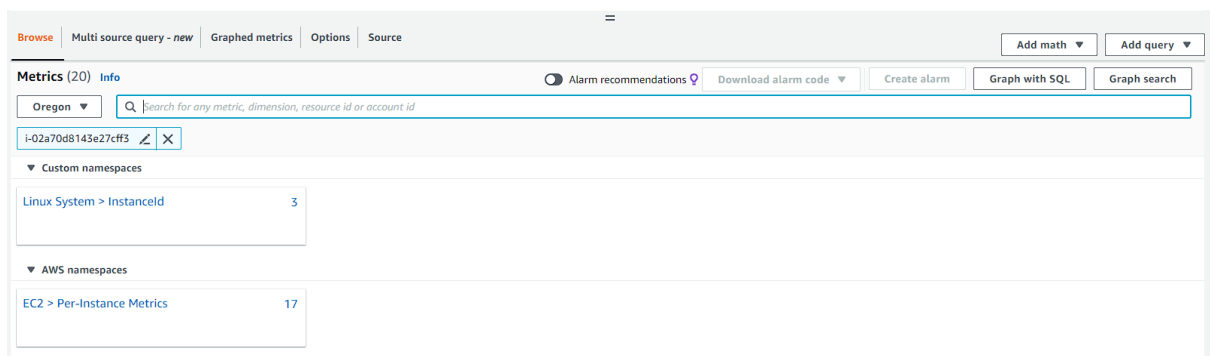
All metrics

Explorer

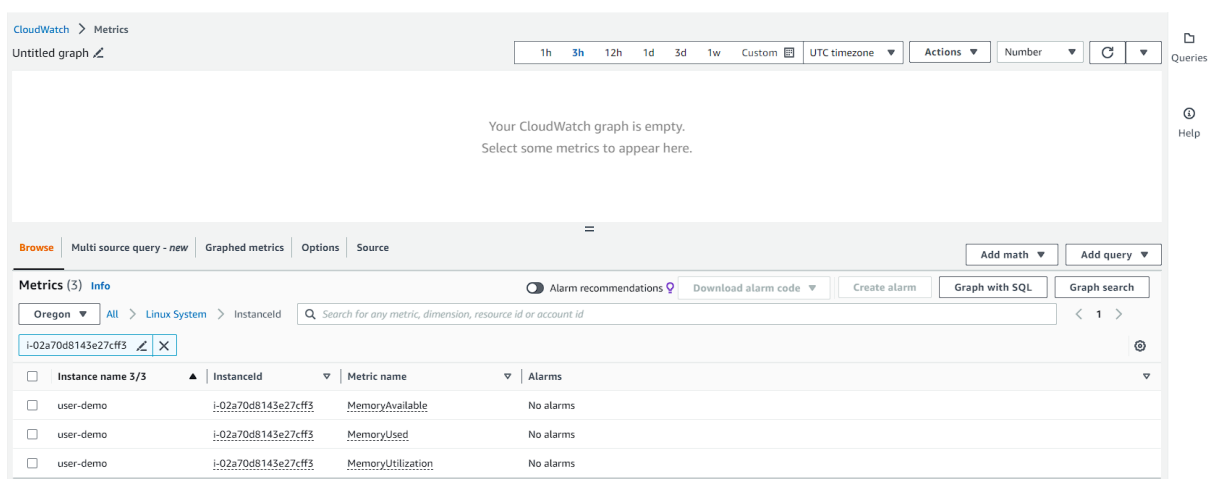
Streams



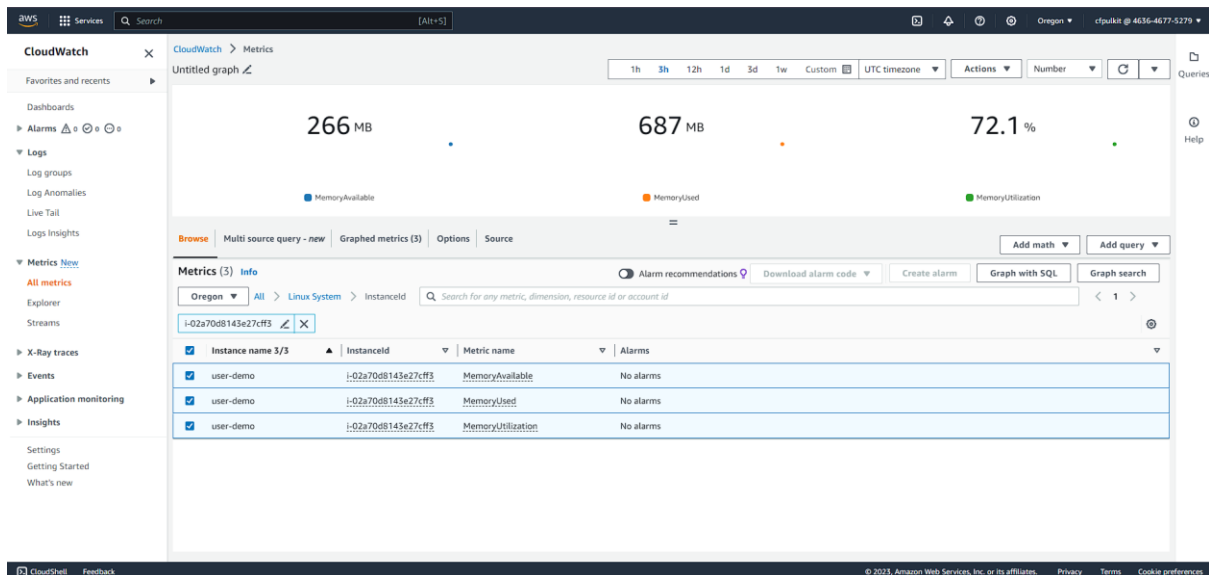
17. Once you have pasted the instance ID, you'll see some data in custom namespace. There you need to click on **Linux System>InstanceID**.



18. There you'll see that no data is present but if you select all the instance names then you'll have the data of your instance.



19. This is how the data will look like. You can also change the view option for the data (if you want)



## Step 2: Disk Related Commands

1. What you have done so far relates to the memory and how to view its availability, consumption, and utilisation.
2. Now you need to check all of these for the disk. For that first you need to check your disk name. So, the command to check disk name is **"df -h"**  
**/dev/xvda1** is the disk that you want.

```
[ec2-user@ip-172-31-22-252 ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        468M   0  468M   0% /dev
tmpfs           477M   0  477M   0% /dev/shm
tmpfs           477M 440K  476M   1% /run
tmpfs           477M   0  477M   0% /sys/fs/cgroup
/dev/xvda1      8.0G  1.7G  6.4G  22% /
tmpfs           96M   0   96M   0% /run/user/1000
```

3. Now that you are aware of the name of your disk, all you need to do is write down each command, execute it, and then visit Cloud Watch to see how much disk space is being used.

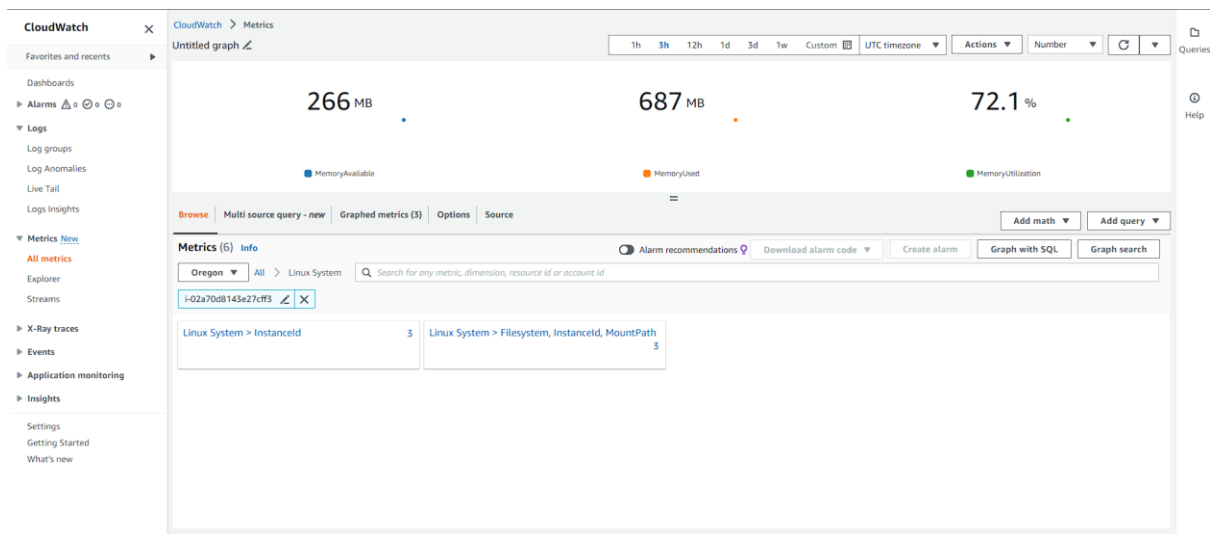
**./mon-put-instance-data.pl --disk-space-util --disk-path=/dev/xvda1**

**./mon-put-instance-data.pl --disk-space-avail --disk-path=/dev/xvda1**

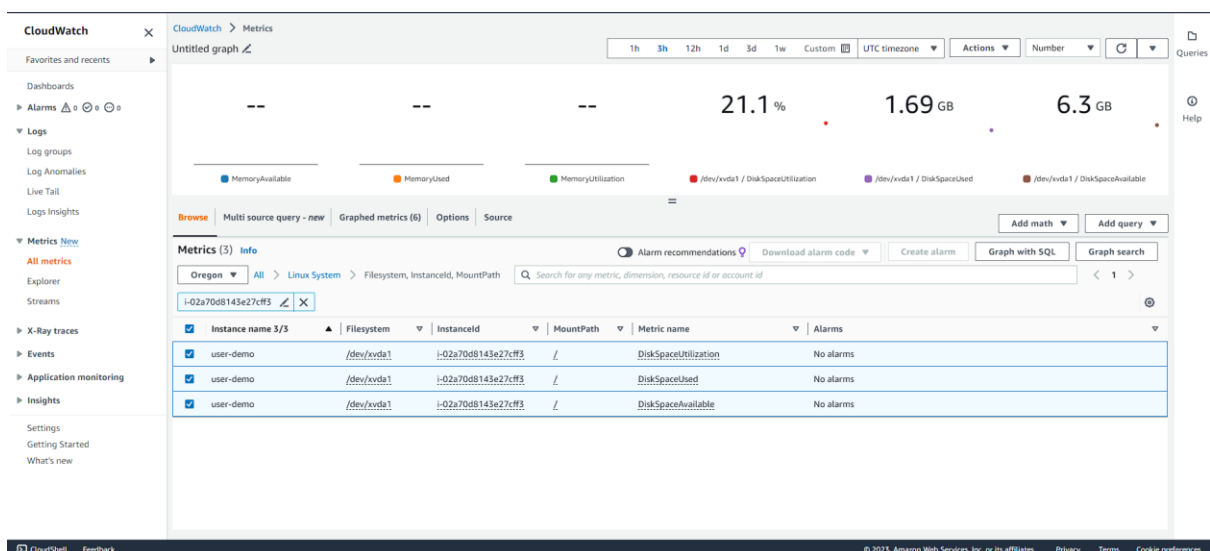
**./mon-put-instance-data.pl --disk-space-used --disk-path=/dev/xvda1**

```
[root@ip-172-31-22-252 aws-scripts-mon]# ./mon-put-instance-data.pl --disk-space-util --disk-path=/dev/xvda1
Successfully reported metrics to CloudWatch. Reference Id: 21115300-e383-4d2c-a6be-0d941a6eb03a
[root@ip-172-31-22-252 aws-scripts-mon]# ./mon-put-instance-data.pl --disk-space-avail --disk-path=/dev/xvda1
Successfully reported metrics to CloudWatch. Reference Id: f5a4d077-f26a-47cf-b15e-2ad6c8a512a0
[root@ip-172-31-22-252 aws-scripts-mon]# ./mon-put-instance-data.pl --disk-space-used --disk-path=/dev/xvda1
Successfully reported metrics to CloudWatch. Reference Id: a63e3bb2-9656-4e3d-bfd8-5b97911bfd87
[root@ip-172-31-22-252 aws-scripts-mon]#
```

- Afterwards go to Cloud Watch and see that 3 new files have been created. Now just click on **Linux System>Filesystem, InstanceId, MountPath**



- Now select the Instance Names and you'll have your data.



## Step 2: Setting Cron Tab

- The initial action taken was to push disk and memory commands. You now need to configure a cron tab to update your data once every minute. First you need write this command '**crontab -e**'

```
[root@ip-172-31-22-252 aws-scripts-mon]# crontab -e
```

```
*/1 * * * * /cloudwatch/aws-scripts-mon/mon-put-instance-data.pl --mem-used-  
incl-cache-buff --mem-util --disk-space-util --disk-path=/dev/xvda1 --from-cron
```

```
*/1 * * * * /cloudwatch/aws-scripts-mon/mon-put-instance-data.pl --mem-used-  
incl-cache-buff --mem-used --disk-space-used --disk-path=/dev/xvda1 --from-cron
```



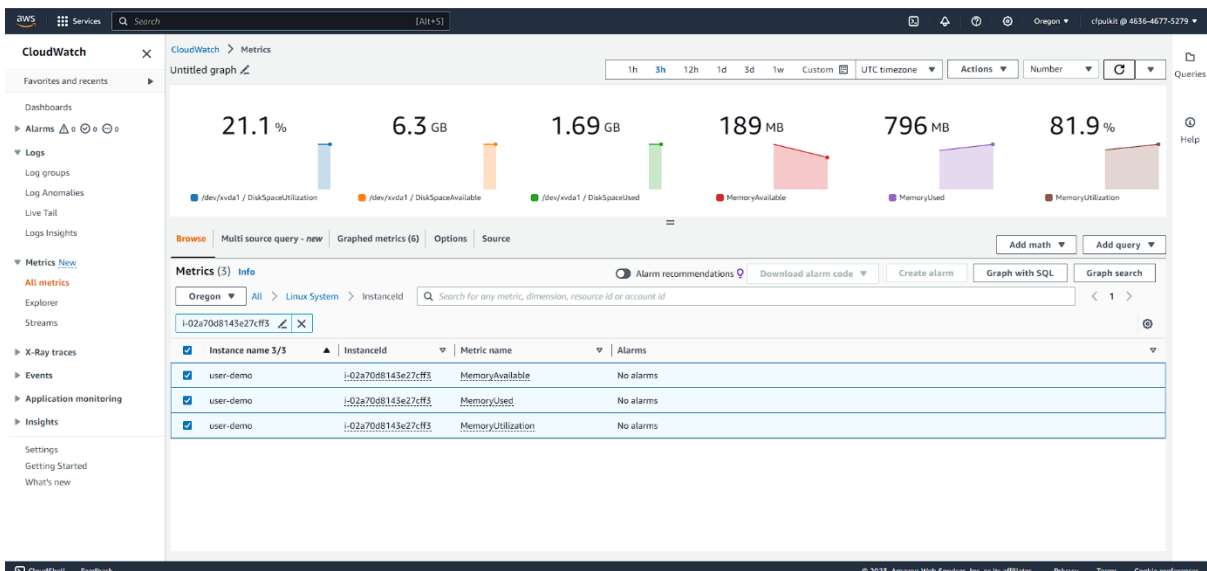
**\* /1 \* \* \* \* /cloudwatch/aws-scripts-mon/mon-put-instance-data.pl --mem-used-incl-cache-buff --mem-avail --disk-space-avail --disk-path=/dev/xvda1 --from-cron**  
Copy all these commands and paste them in one go. In this window you need to press **insert** first then paste all these commands. Then press **ESC** write this command to save this file.  
**':wq!'**

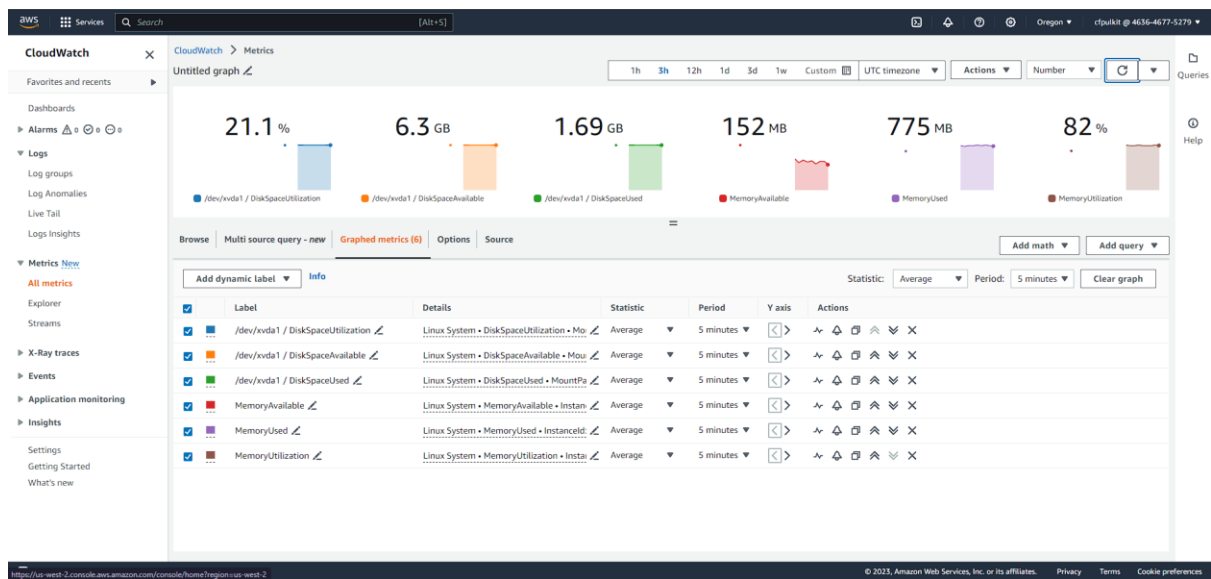
```
root@ip-172-31-22-252:/home
*/1 * * * * /cloudwatch/aws-scripts-mon/mon-put-instance-data.pl --mem-used-incl-cache-buff --mem-avail --disk-space-avail --disk-path=/dev/xvda1 --from-cron
*/1 * * * * /cloudwatch/aws-scripts-mon/mon-put-instance-data.pl --mem-used-incl-cache-buff --mem-used --disk-space-used --disk-path=/dev/xvda1 --from-cron
*/1 * * * * /cloudwatch/aws-scripts-mon/mon-put-instance-data.pl --mem-used-incl-cache-buff --mem-avail --disk-space-avail --disk-path=/dev/xvda1 --from-cron

~/tmp/crontab.Nuivlw* 5L, 472B 1,1 All
```

```
[root@ip-172-31-22-252 aws-scripts-mon]# crontab -e
crontab: installing new crontab
[root@ip-172-31-22-252 aws-scripts-mon]# |
```

2. Now go back to Cloud Watch and refresh your data.





3. As you can see the data has updated and it has changed.