**Monitoring Memory and Disk Metrics for AWS EC2 Instances**

Amazon Web Services(AWS) reports some good metrics on the console by default, like CPU, but some key metrics like memory usage or disk space are missing; these are important to monitor to ensure instance up-time and health.

In this post we'll look at how we can use [CloudWatch](https://aws.amazon.com/cloudwatch/) to monitor these extended metrics, allowing you to build reports, dashboards, and alerts.

**📘 NOTE**  
Before we begin, note that standard Amazon CloudWatch usage charges will be applicable for these scripts. For more information, see the [Amazon CloudWatch](https://aws.amazon.com/cloudwatch/pricing/) pricing page.

✋ For simplicity, let's assume that we are using EC2 with Amazon Linux operating system. For other operating systems, you can use respective commands to achieve the results.

**Creating an IAM role to access the metrics**

In order to pass metrics data from EC2 to AWS Cloudwatch, we will have to create a user with the following IAM role access:

* cloudwatch:PutMetricData
* cloudwatch:GetMetricStatistics
* cloudwatch:ListMetrics
* ec2:DescribeTags

Got confused? Just create a policy (say - cloudwatch-ec2-access) with the above permissions. After that, create a user (cloudwatch-stats-user) and attach the created policy(cloudwatch-ec2-access) to the user. Also, store the generated AWSAccessKeyId and AWSSecretKey of this user which will be required at a later stage.

**SSH to EC2**

SSH to your EC2 instance and perform the following steps:

**1. Create a script folder**

I generally prefer to keep everything in a separate folder for clarity, though not mandatory. As we are using Amazon Linux 2 AMI, steps will look similar to this:

# current folder /home/ec2-user

$ mkdir cloudwatch\_logs

$ cd cloudwatch\_logs

**2. Install the required packages**

In order to be able to run the AWS scripts, we will have to install some packages. Also, note that the command may change based on your operating system. Check this [link](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/mon-scripts.html) for more information.

sudo yum install -y perl-Switch perl-DateTime perl-Sys-Syslog perl-LWP-Protocol-https perl-Digest-SHA.x86\_64

Refer link for different OS :

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/mon-scripts.html

### 3. Download the scripts from AWS

Now that we have installed all the packages, we need to download the Perl scripts provided by AWS.

curl https://aws-cloudwatch.s3.amazonaws.com/downloads/CloudWatchMonitoringScripts-1.2.2.zip -O

### 4. Unzip the scripts

unzip CloudWatchMonitoringScripts-1.2.2.zip

# remove the zip

rm CloudWatchMonitoringScripts-1.2.2.zip

# move to the unzipped folder

cd aws-scripts-mon

The package for the monitoring scripts contains the following files:

* CloudWatchClient.pm – Shared Perl module that simplifies calling Amazon CloudWatch from other scripts.
* mon-put-instance-data.pl – Collects system metrics on an Amazon EC2 instance (memory, swap, disk space utilization) and sends them to Amazon CloudWatch.
* mon-get-instance-stats.pl – Queries Amazon CloudWatch and displays the most recent utilization statistics for the EC2 instance on which this script is executed.
* awscreds.template – File template for AWS credentials that stores your access key ID and secret access key.
* LICENSE.txt – Text file containing the Apache 2.0 license.
* NOTICE.txt – Copyright notice.

**5. Add Access ID and Secret Key of cloudwatch user (cloudwatch-stats-user)**

As mentioned in the earlier step, amazon provides a template file awscreds.template which can be used to create a conf file to store AWS credentials.

# creates a conf file from the template. Make sure the filename is as is

cp awscreds.template awscreds.conf

Next, add your AWS Access ID and secret key in this file and save it.

### 6. Verify if statistics are captured correctly

We will now verify if everything is in place and works smoothly with following command:

/home/ec2-user/cloudwatch\_logs/aws-scripts-mon/mon-put-instance-data.pl --mem-used-incl-cache-buff --mem-util --mem-used --mem-avail --disk-space-util --disk-space-avail --disk-path=/ --verify --verbose

### 7. Final Step

Congratulations! 👏 You have successfully configured Cloudwatch metrics on your EC2 instance. Now one last step is to add it to cron so it can send metrics after every 5 mins.

# Open the crontab file

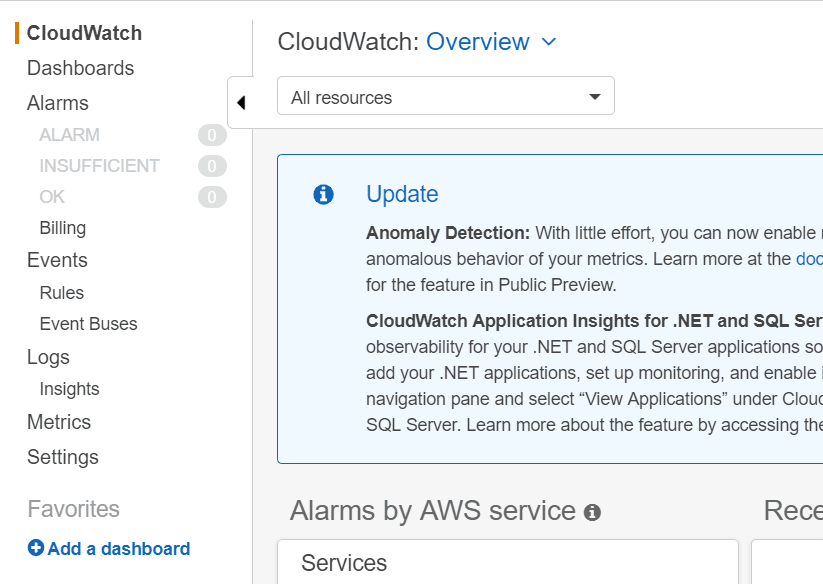
crontab -e

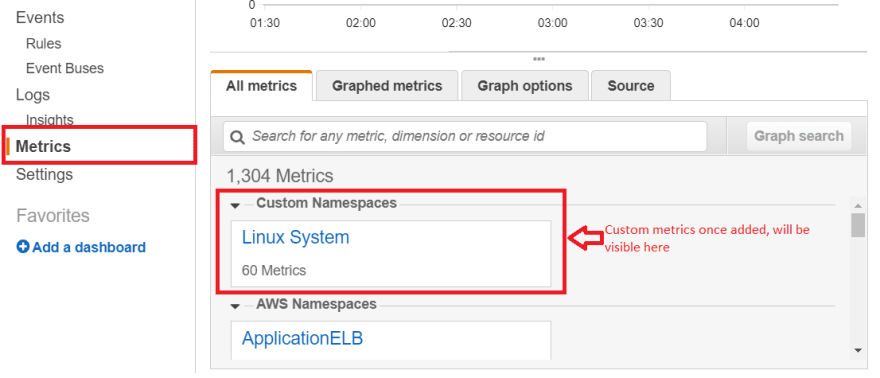
# Add the following line and save it

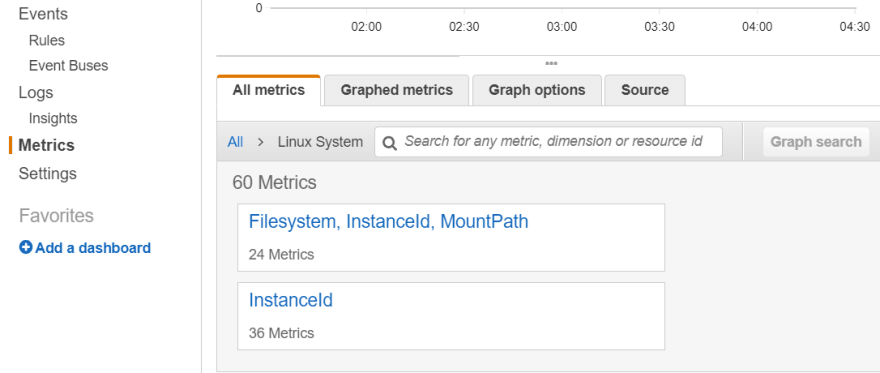
# Cloudwatch Monitoring Metrics (AWS)

\*/5 \* \* \* \* /home/ec2-user/cloudwatch\_logs/aws-scripts-mon/mon-put-instance-data.pl --mem-used-incl-cache-buff --mem-util --mem-used --mem-avail --disk-space-util --disk-space-avail --disk-path=/ --from-cron

# Where to find these metrics on AWS?

In your AWS console, go to Cloudwatch service.  
[](https://res.cloudinary.com/practicaldev/image/fetch/s---gQzT8RR--/c_limit%2Cf_auto%2Cfl_progressive%2Cq_auto%2Cw_880/https:/res.cloudinary.com/idr/image/upload/v1567136936/dev.to/cloudwatch_screen_1_jc48in.png)

All your metrics are available in Metrics menu. A new Custom Namespaces will be added in your metrics view once Cloudwatch starts receiving metrics from the scripts.  
[](https://res.cloudinary.com/practicaldev/image/fetch/s--XEYIlj5B--/c_limit%2Cf_auto%2Cfl_progressive%2Cq_auto%2Cw_880/https:/res.cloudinary.com/idr/image/upload/v1567393095/dev.to/cloudwatch_screen_2_jxorgj.png)

Click on it, and you will get 2 options as depicted in the image below:  
[](https://res.cloudinary.com/practicaldev/image/fetch/s--UA0fmrFJ--/c_limit%2Cf_auto%2Cfl_progressive%2Cq_auto%2Cw_880/https:/res.cloudinary.com/idr/image/upload/v1567136936/dev.to/cloudwatch_screen_3_wcxzhs.png)

1. Filesystem, InstanceId, MountPath - All your EC2 disk metrics will be available inside this option.
2. InstanceId - All your EC2 memory metrics will be available inside this option.

**Points to Remember**

* Whenever you reboot or allocate/deallocate disk space, your EC2 disk filesystem path may change and hence you might require to reconfigure any alarms or dashboards you had created for monitoring EC2.
* If you have launched an EC2 instance from an AMI which had these metrics configured, then the above setup is already present. Just clear the cache via command: rm /var/tmp/aws-mon/instance-id
* AWS has recently launched CloudWatch Agent to collect both system metrics and log files from Amazon EC2 instances. Hence, it is recommended to use CloudWatch Agent to collect metrics and logs in place of these monitoring scripts. However, there might be certain cases where you would require to use these monitoring scripts and this post might help you to configure it.