

# Cloud-Watch Custom Matrix

## Aws Documentation link:-

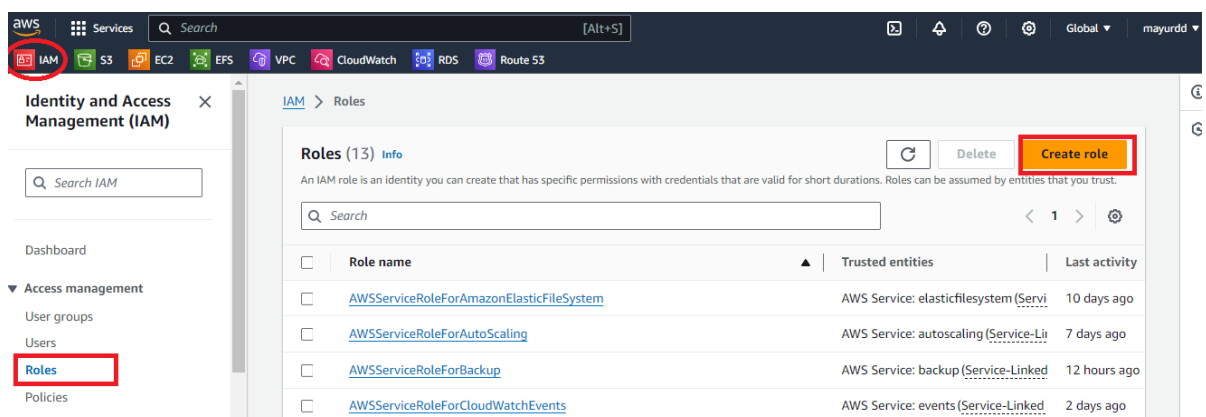
<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/monitoring-scripts-intro.html>

## What is Cloudwatch Custom matrix ??

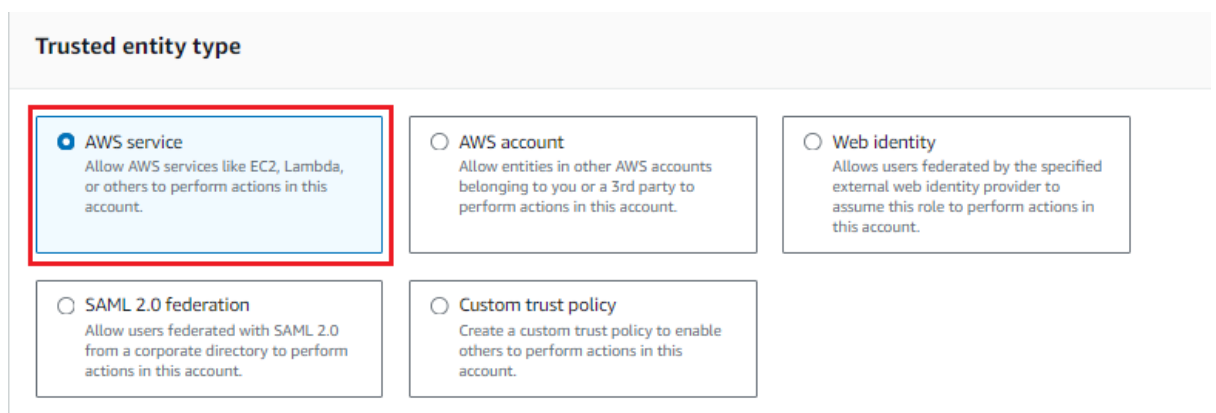
Custom metrics in CloudWatch allow you to monitor and collect data about your applications, services, and resources that are not automatically provided by AWS.

### 1. Creating role for ec2 instance

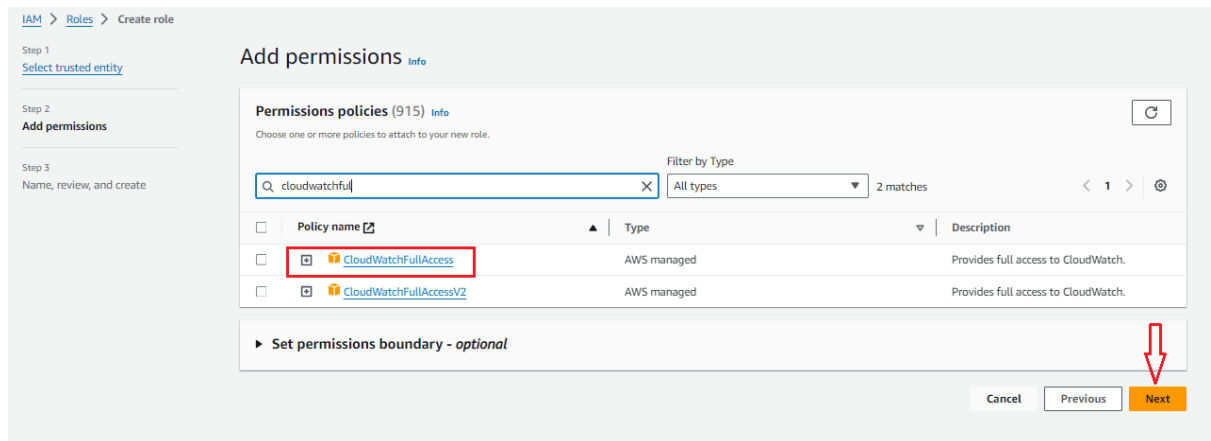
- Select IAM service & click on create role



- Select Ec2 Service & click on next



- Select The permission policy & click on next



- Assign name and click on next

**#policy created successfully.....**

## 2. Create Ec2 Instance

Select Amazon AMI 2 machine for performing this practical because The monitoring scripts were tested on instances using the following systems.....

Amazon Linux 2

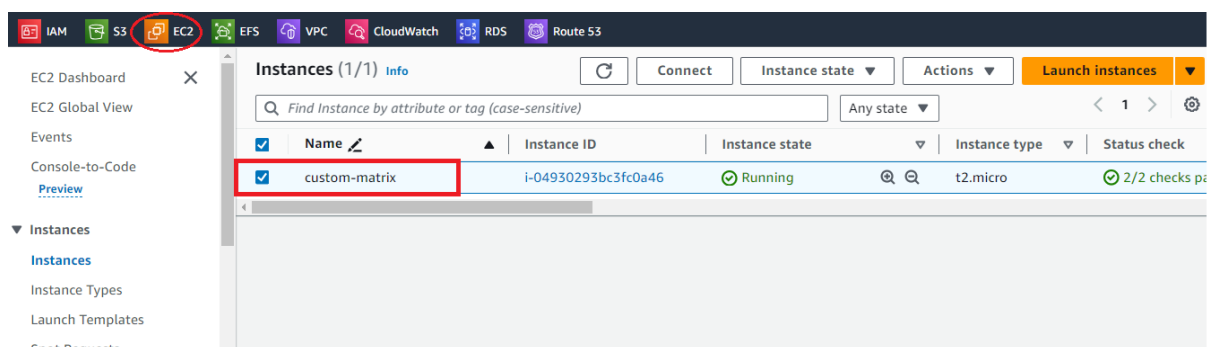
Amazon Linux AMI 2014.09.2 and later

Red Hat Enterprise Linux 6.9 and 7.4

SUSE Linux Enterprise Server 12

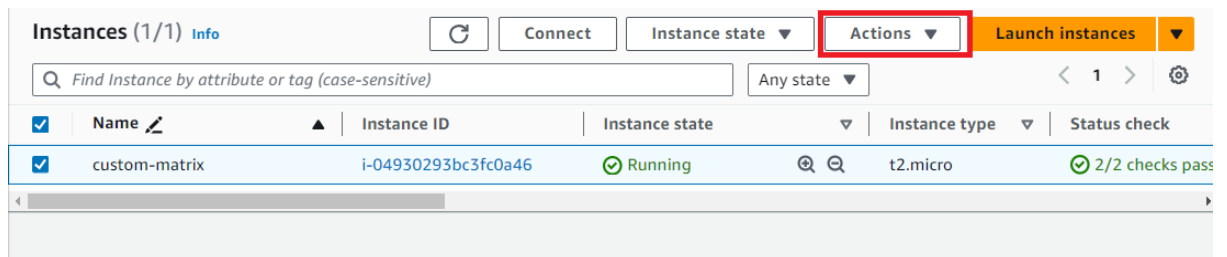
Ubuntu Server 14.04 and 16.04

- **Amazon Linux 2 Instance created successfully.....**

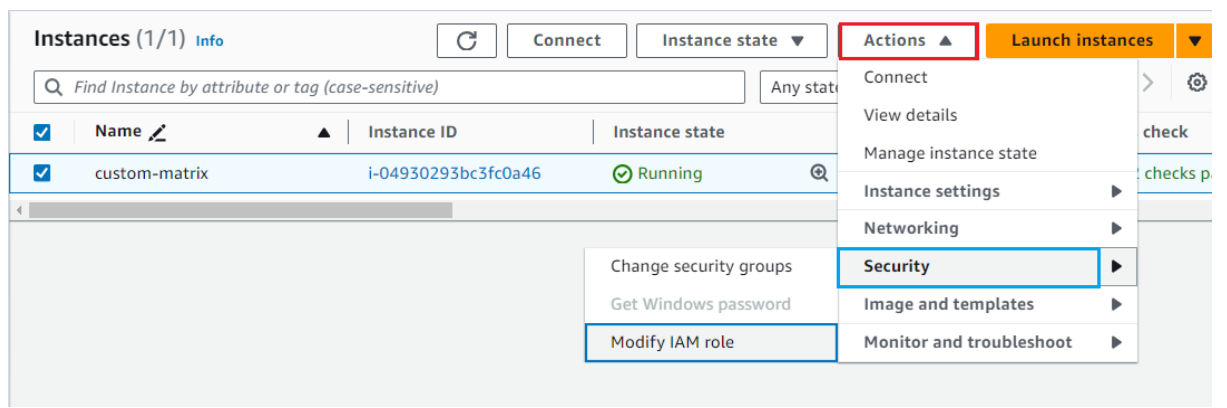


### 3. Assign role To Ec2 Instance

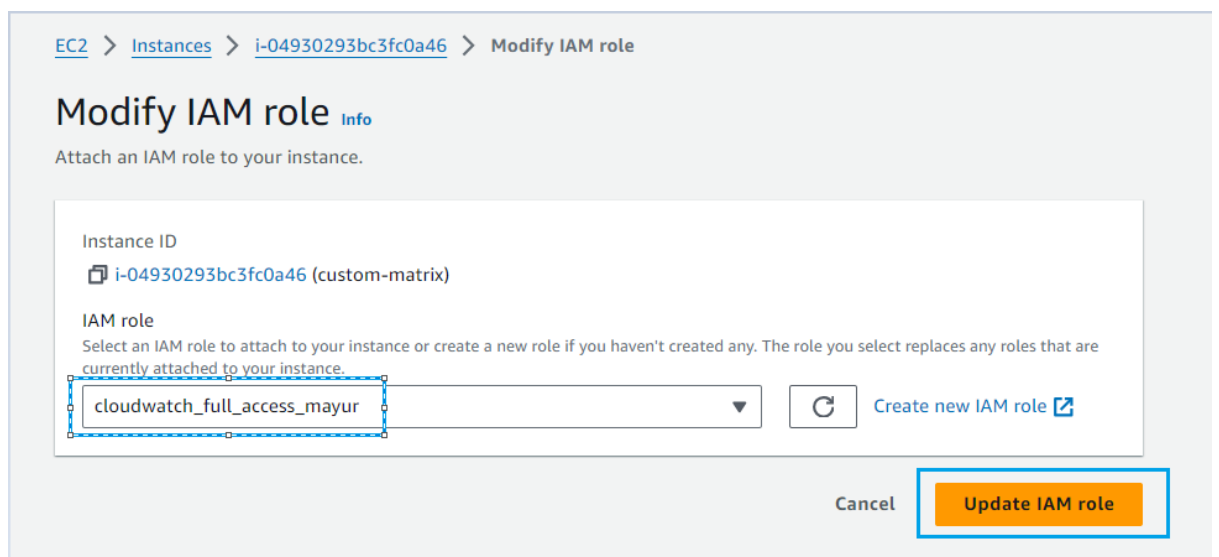
- Select the Created instance & click on Action option



- Under Security option Click on Modify IAM role



- Select The role And click on update IAM role



➤ **Role attached successfully.....**

(Now we can able to perform all actions on Cloudwatch service using EC2 instance...)

**4. Install required packages**

```
{{ sudo yum install -y perl-Switch perl-DateTime perl-Sys-Syslog perl-LWP-Protocol-https perl-Digest-SHA.x86_64 }}
```

**5. Install monitoring scripts**

**Run the following commands For installing the monitoring script**

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

**Run the following commands to install the monitoring scripts you downloaded:**

```
unzip CloudWatchMonitoringScripts-1.2.2.zip && \  
rm CloudWatchMonitoringScripts-1.2.2.zip && \  
cd aws-scripts-mon
```

**6. Run The script as per your Requirement**

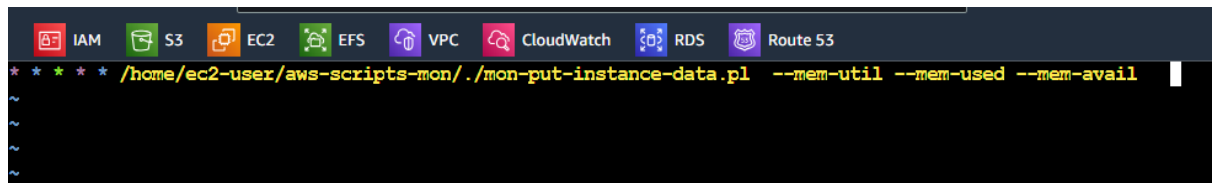
```
{{ ./mon-put-instance-data.pl --mem-util --mem-used --mem-avail }}
```

(note: for every time we need to push the data from our instance to cloudwatch service. For avoiding this issue we use crontab tool for automation....)

## 7. Automation using Crontab

- Add this script in crontab using (Crontab -e )

```
* * * * * /home/ec2-user/aws-scripts-mon/./mon-put-instance-data.pl  
--mem-util --mem-used --mem-avail
```



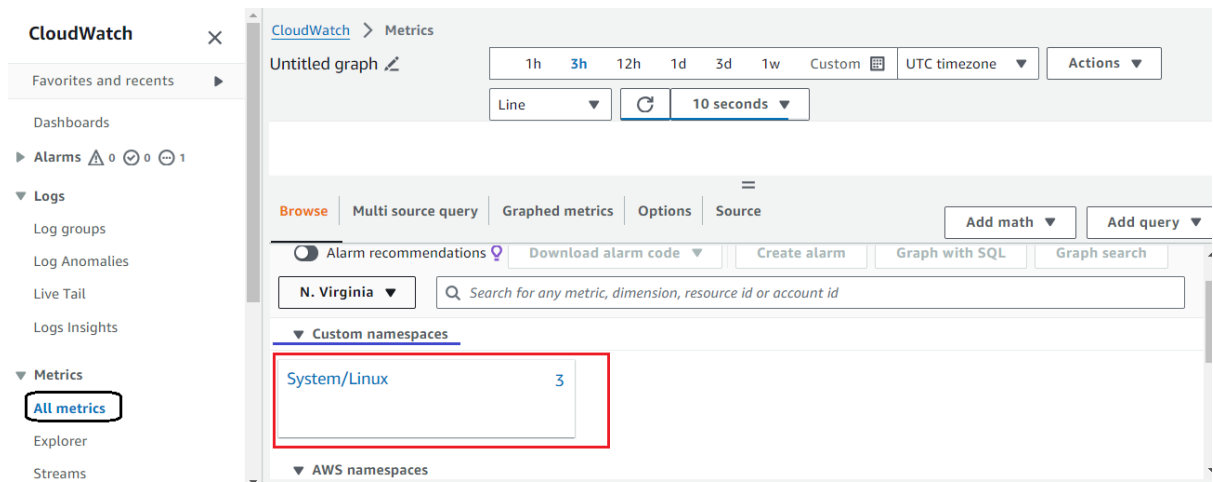
- Save & restart the crond service

sudo systemctl restart crond

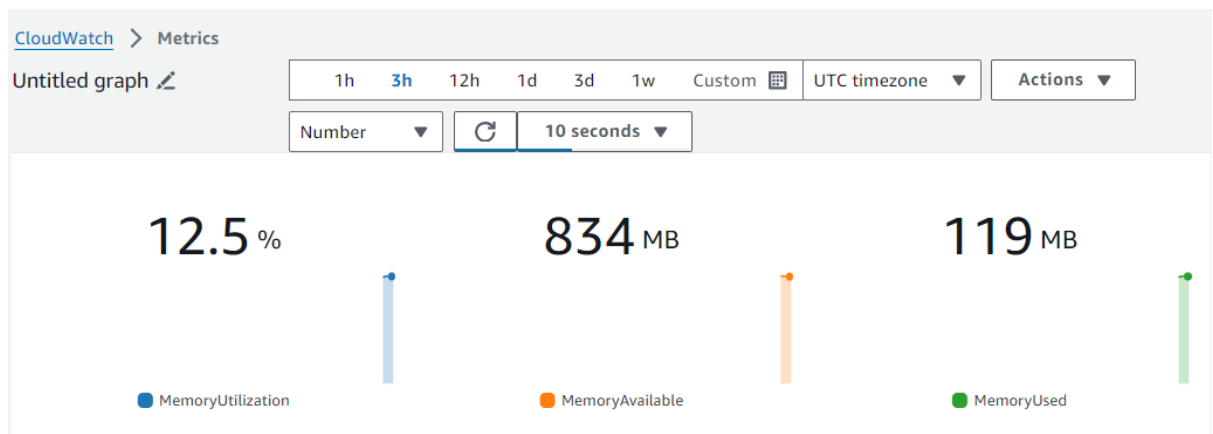
```
[ec2-user@ip-172-31-37-162 aws-scripts-mon]$ sudo systemctl restart crond  
You have new mail in /var/spool/mail/ec2-user  
[ec2-user@ip-172-31-37-162 aws-scripts-mon]$ sudo systemctl status crond  
● crond.service - Command Scheduler  
   Loaded: loaded (/usr/lib/systemd/system/crond.service; enabled; vendor preset: enabled)  
   Active: active (running) since Thu 2024-03-14 21:08:21 UTC; 9s ago  
     Main PID: 3928 (crond)  
       CGroup: /system.slice/crond.service  
              └─3928 /usr/sbin/crond -n  
  
Mar 14 21:08:21 ip-172-31-37-162.ec2.internal systemd[1]: Stopped Command Scheduler.  
Mar 14 21:08:21 ip-172-31-37-162.ec2.internal systemd[1]: Started Command Scheduler.  
Mar 14 21:08:21 ip-172-31-37-162.ec2.internal crond[3928]: (CRON) INFO (RANDOM_DELAY will be scaled with factor 76% if used.  
Mar 14 21:08:21 ip-172-31-37-162.ec2.internal crond[3928]: (CRON) INFO (running with inotify support)  
Mar 14 21:08:21 ip-172-31-37-162.ec2.internal crond[3928]: (CRON) INFO (@reboot jobs will be run at computer's startup.)  
[ec2-user@ip-172-31-37-162 aws-scripts-mon]$
```

## 8. Result

- Successfully pushed the data from Ec2 instance to cloudwatch service....



- Result is shown in numbers



## CloudWatch Agent on Our servers

### **\*\*Configuring Cloudwatch agent on Ubuntu operation system\*\***

#### **1. Launch the Ubuntu instance**

While launching the instance add this user-data script or we can do after installing as well....

&& make sure to add http port (80) in security group

```
#!/bin/bash
sudo apt install apache2 -y
sudo systemctl start apache2
sudo systemctl enable apache2
|
```

**Successfully launch the Ubuntu instance....**

Instances (1) Info

Refresh

Connect

Instance state ▼

Actions ▼

Launch instances

▼

Find Instance by attribute or tag (case-sensitive)

Running ▼

< 1 >

⚙

<input type="checkbox"/>	Name <div>✎</div> ▼	Instance ID	Instance state ▼	Instance type ▼	Status check
<input type="checkbox"/>	agenting	i-05d84eabb6ed173f9	<div>✔ Running</div>	t2.micro	<div>🔄 Initializing</div>

## 2. Configuration on Ubuntu instance

### ➤ Downloading amazon cloudwatch agent

```
sudo wget https://s3.amazonaws.com/amazoncloudwatch-agent/debian/amd64/latest/amazon-cloudwatch-agent.deb
```

### ➤ Install the agent

```
sudo dpkg -i -E ./amazon-cloudwatch-agent.deb
```

### ➤ Start Configuring the agent

```
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-config-wizard
```

### ➤ Some important options we need to assign

```
"default_user": "root"
```

```
"file_path": "/var/log/apache2/access.log",
```

```
"log_group_class": "STANDARD",
```

```
"log_group_name": "demo-ec2-apache.logs",
```

```
"log_stream_name": "apache.access.log",
```

```
"retention_in_days": -1
```

### ➤ Result is stored here

Go through all the steps in the wizard (The result is saved here:  
/opt/aws/amazon-cloudwatch-agent/bin/config.json)

### ➤ For checking configuration

```
cat /opt/aws/amazon-cloudwatch-agent/bin/config.json
```



### ➤ Installing collectd

- `sudo apt-get update -y`
- `sudo apt-get install collectd`
- `sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m ec2 -c file:/opt/aws/amazon-cloudwatch-agent/bin/config.json -s`
- `sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a status -----> to check status`

### 3. Result

- After successfully completing the configuration we get the access.log file in Log Group section ....

