



lab



lab title

**Monitoring AWS Services**

**V1.00**



Course title

**BackSpace Academy**  
**AWS Certified Associate**



# Table of Contents

## Contents

Table of Contents.....	1
About the Lab .....	2
Implementing CloudWatch Monitoring Scripts on EC2 .....	3
Create IAM Role for EC2 .....	3
Create EC2 Server .....	7
Running the EC2 Monitoring Scripts.....	10
Creating a CRON task to Push Metrics to CloudWatch .....	11
Viewing Metrics in CloudWatch .....	11
Clean Up.....	13

## About the Lab

**Please note that not all AWS services are supported in all regions. Please use the US-East-1 (North Virginia) region for this lab.**

These lab notes are to support the hands on instructional videos of the Monitoring section of the AWS Certified Associate Course.

**Please note that AWS services change on a weekly basis and it is extremely important you check the version number on this document to ensure you have the latest version with any updates or corrections.**

# 🎬 Implementing CloudWatch Monitoring Scripts on EC2

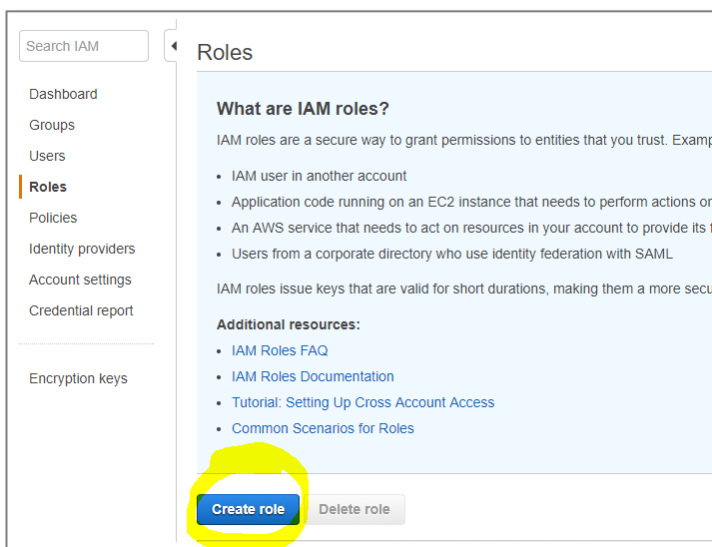
In this section we will create an EC2 server and assign it a role to access the CloudWatch service. We will then install CloudWatch monitoring scripts to produce custom CloudWatch metrics.

Create IAM Role for EC2

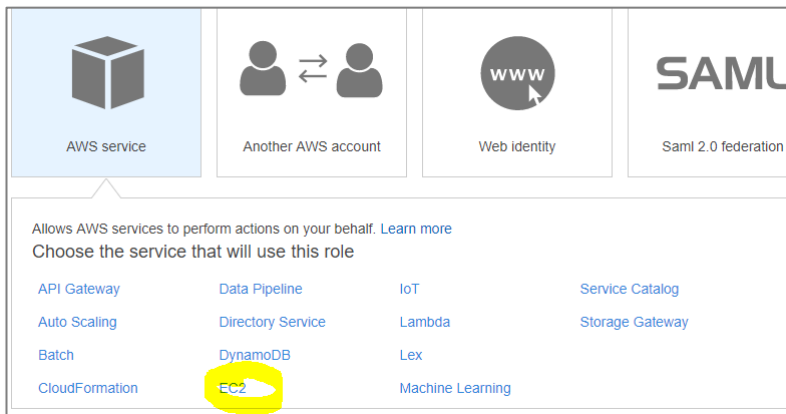
Select “Services” “IAM” to go to the IAM console

Select Roles

Click “Create Role”

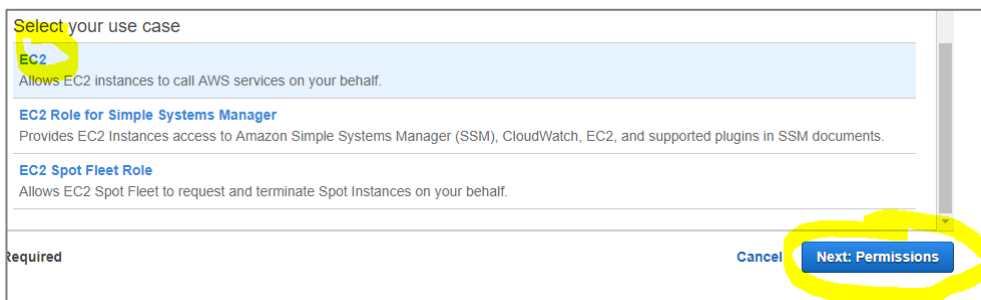


Select EC2

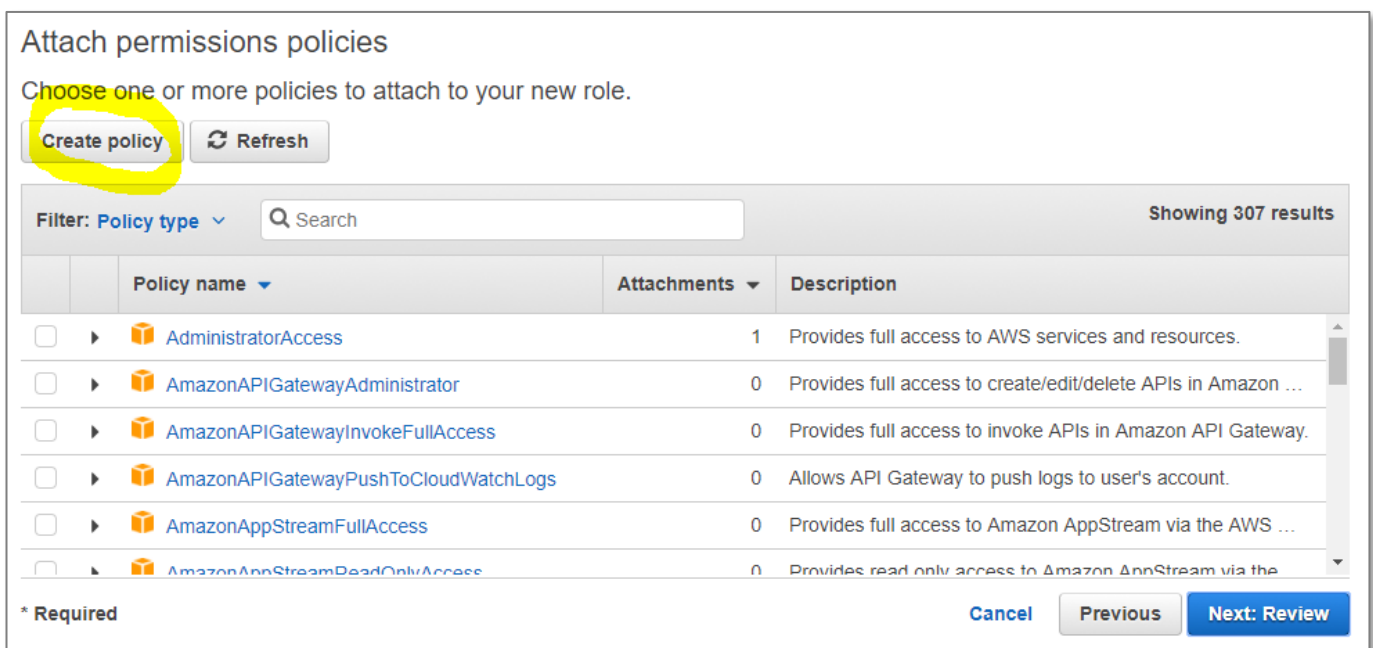


Select EC2 again

Click "Next: Permissions"



Click "Create Policy"



Select the "Create Your Own Policy"

### Create Policy

A policy is a document that formally states one or more permissions. Create a policy by copying an AWS Managed Policy, using the Policy Generator, or typing your own custom policy.

---

Copy an AWS Managed Policy

Start with an AWS Managed Policy, then customize it to fit your needs.

Select

---

Policy Generator

Use the policy generator to select services and actions from a list. The policy generator uses your selections to create a policy.

Select

---

Create Your Own Policy

Use the policy editor to type or paste in your own policy.

Select

Give the policy a name and description

Paste the following JSON into Policy Document.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:GetMetricStatistics",
        "cloudwatch:ListMetrics",
        "cloudwatch:PutMetricData"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeTags"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

Click "Validate Policy to check it is OK

Click "Create Policy"

### Review Policy

Customize permissions by editing the following policy document. For more information about the access policy language, see [Overview of Policies](#) in the *Using IAM* guide. To test the effects of this policy before applying your changes, use the [IAM Policy Simulator](#).

This policy is valid.

**Policy Name**  
EC2-AWS-Monitoring-Scripts

**Description**  
Permissions required for the AWS EC2 Perl monitoring scripts

**Policy Document**

```
14     },
15     {
16       "Effect": "Allow",
17       "Action": [
18         "ec2:DescribeTags"
19       ],
20       "Resource": [
21         "*"
22       ]
23     }
24   ]
}
```

☒ Use autoformatting for policy editing

[Cancel](#) [Validate Policy](#) [Previous](#) [Create Policy](#)

Go back to the Roles page

Click on the role you created

Search IAM

- Dashboard
- Groups
- Users
- Roles**
- Policies
- Identity providers
- Account settings
- Credential report
- Encryption keys

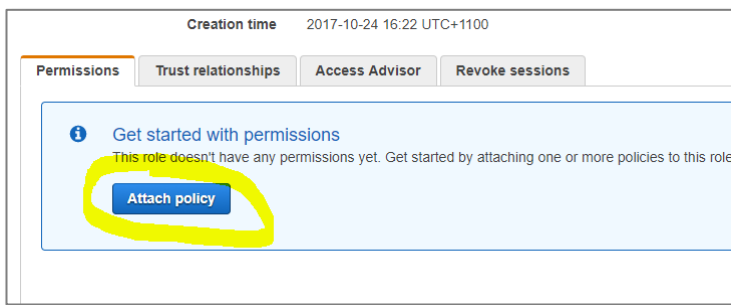
Common Scenarios for Roles

[Create role](#) [Delete role](#)

Search

Role name	Description
<input type="checkbox"/> AWS-CodePipeline-Service	
<input type="checkbox"/> aws-codestar-service-role	
<input type="checkbox"/> aws-elasticbeanstalk-ec2-r...	
<input type="checkbox"/> aws-elasticbeanstalk-servi...	
<input type="checkbox"/> code-build-backspace-aws...	
<input type="checkbox"/> dms-cloudwatch-logs-role	
<input type="checkbox"/> dms-vpc-role	
<input type="checkbox"/> <b>EC2CloudWatch</b>	
<input type="checkbox"/> rds-monitoring-role	

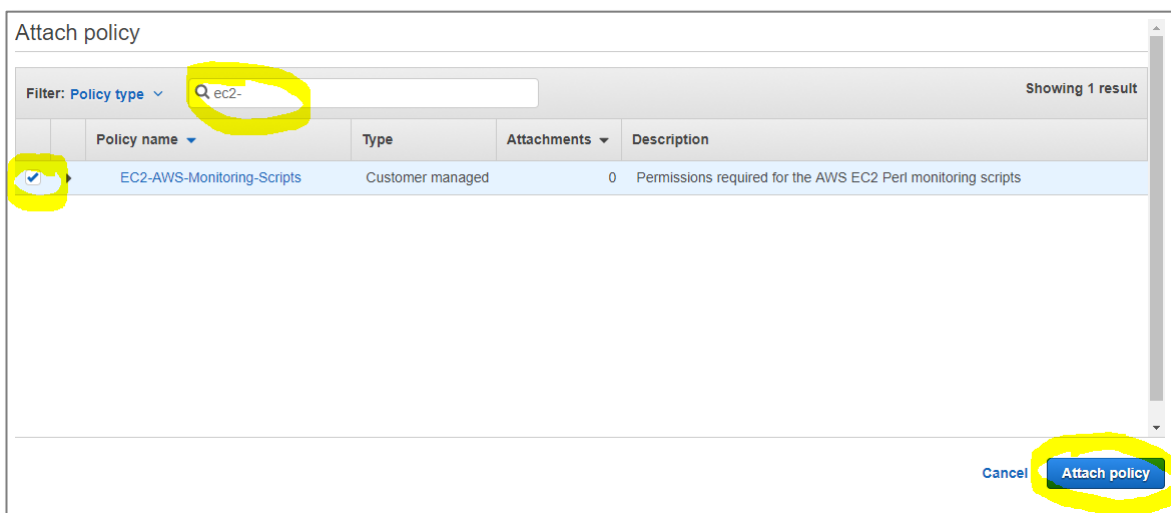
Click "Attach policy"



Search for your policy

Select the policy

Click "Attach policy"



Create EC2 Server

Click on the services menu and select EC2.

Click "Launch Instance"

Select the Amazon Linux AMI



**Step 1: Choose an Amazon Machine Image (AMI)** [Cancel and Exit](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

**Quick Start**

My AMIs

AWS Marketplace

Community AMIs

☐ Free tier only ⓘ

1 to 33 of 33 AMIs >|

**Amazon Linux AMI 2017.09.0 (HVM), SSD Volume Type** - ami-8c1be5f6

**Free tier eligible**

The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.

Root device type: ebs    Virtualization type: hvm

**Select**

64-bit

**Red Hat Enterprise Linux 7.4 (HVM), SSD Volume Type** - ami-c998b6b2

**Free tier eligible**

Red Hat Enterprise Linux version 7.4 (HVM), EBS General Purpose (SSD) Volume Type

Root device type: ebs    Virtualization type: hvm

**Select**

64-bit

Select t2 micro instance type

Click “Next: Configure Instance Details”

Select our EC2 Role

**Step 3: Configure Instance Details**

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of lower prices, and more.

**Number of instances** ⓘ  [Launch into Auto Scaling Group](#) ⓘ

**Purchasing option** ⓘ ☐ Request Spot instances

**Network** ⓘ  [Create new VPC](#)

**Subnet** ⓘ  [Create new subnet](#)

**Auto-assign Public IP** ⓘ

**IAM role** ⓘ  [Create new IAM role](#)

Expand the “Advanced Details”

Add the following bash script in “User Data”

```
#!/bin/bash
yum -y update
sudo yum install perl-Switch perl-DateTime perl-Sys-Syslog perl-LWP-Protocol-https -y
curl http://aws-cloudwatch.s3.amazonaws.com/downloads/CloudWatchMonitoringScripts-1.2.1.zip -O
unzip CloudWatchMonitoringScripts-1.2.1.zip
```

Click Review and Launch”

### Step 3: Configure Instance Details

**IAM role** EC2CloudWatch [Create new IAM role](#)

**Shutdown behavior** Stop

**Enable termination protection** ☐ Protect against accidental termination

**Monitoring** ☐ Enable CloudWatch detailed monitoring  
[Additional charges apply.](#)

**Tenancy** Shared - Run a shared hardware instance  
[Additional charges will apply for dedicated tenancy.](#)

▼ **Advanced Details**

**User data** ☒ As text ☐ As file ☐ Input is already base64 encoded

```
Protocol=https -y
curl http://aws-
cloudwatch.s3.amazonaws.com/downloads/CloudWatchMonitoringScripts-
1.2.1.zip -O
unzip CloudWatchMonitoringScripts-1.2.1.zip
```

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Add Storage](#)

Scroll down and click “Edit tags”

### Step 7: Review Instance Launch

t2.micro	Variable	1	1	EBS only	-	Low to Moderate
----------	----------	---	---	----------	---	-----------------

▼ **Security Groups** [Edit security groups](#)

**Security group name** launch-wizard-10  
**Description** launch-wizard-10 created 2017-10-24T18:57:01.175+11:00

Type	Protocol	Port Range	Source	Description
This security group has no rules				

► **Instance Details** [Edit instance details](#)

► **Storage** [Edit storage](#)

► **Tags** [Edit tags](#)

[Cancel](#) [Previous](#) [Launch](#)

Give the instance a “Name” tag

### Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.  
A copy of a tag can be applied to volumes, instances or both.  
Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key  (127 characters maximum)	Value  (255 characters maximum)	Instances	Volumes
Name	EC2 CloudWatch Monitoring	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

[Add another tag](#) (Up to 50 tags maximum)

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Configure Security Group](#)

Click Review and Launch”

Click “Launch”

## Select a key pair launch the instance

## Running the EC2 Monitoring Scripts

Open your command line.

## Navigate to the location of your key pair

## Connect to your instance using the CLI

```
G:\OneDrive\Documents\KeyPairs\BackSpace
λ ssh -i "pcoady-us-east-1.pem" ec2-user@ec2-52-207-109-192.compute-1.amazonaws.com
The authenticity of host 'ec2-52-207-109-192.compute-1.amazonaws.com (52.207.109.192)' can't be established.
ECDSA key fingerprint is SHA256:3D+WibAgR9sff7+GIiFP5E9rwr7mR2NvNwHDgxv17Pk.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'ec2-52-207-109-192.compute-1.amazonaws.com,52.207.109.192' (ECDSA) to the list of known hosts.

 _ _ | _ _ | _ _ |
 _ | ( _ _ | _ _ | /
 _ | \ _ _ | _ _ |

Amazon Linux AMI

https://aws.amazon.com/amazon-linux-ami/2017.09-release-notes/
[ec2-user@ip-172-31-81-206 ~]$
```

Change directory to where the scripts are located

```
cd ~/../../aws-scripts-mon
```

Check the scripts are working

```
./mon-put-instance-data.pl --mem-util --verify --verbose
```

```
[ec2-user@ip-172-31-81-206 aws-scripts-mon]$ ./mon-put-instance-data.pl --mem-util --verify --verbose
MemoryUtilization: 7.96874938564312 (Percent)
No credential methods are specified. Trying default IAM role.
Using IAM role <EC2CloudWatch>
Endpoint: https://monitoring.us-east-1.amazonaws.com
Payload: {"MetricData":[{"Timestamp":1508823931,"Dimensions":[{"Value":"i-065f24f3730d0542","Name":"InstanceId"}],"Value":7.96874938564312,"Unit":"Percent","MetricName":"MemoryUtilization"}],"Namespace":"System/Linux","__type":"com.amazonaws.cloudwatch.v2010_08_01.PutMetricDataInput"}
Verification completed successfully. No actual metrics sent to CloudWatch.

[ec2-user@ip-172-31-81-206 aws-scripts-mon]$
```

### Run the script to send metric data to CloudWatch

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

## Creating a CRON task to Push Metrics to CloudWatch

Edit the Linux CRON Table using the command

```
crontab -e
```

Press insert key



We will push metric data to Cloudwatch every minute

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

After you have copied and pasted the CRON statement you can save and exit

Press ESC to exit insert mode

```
:wq!
```

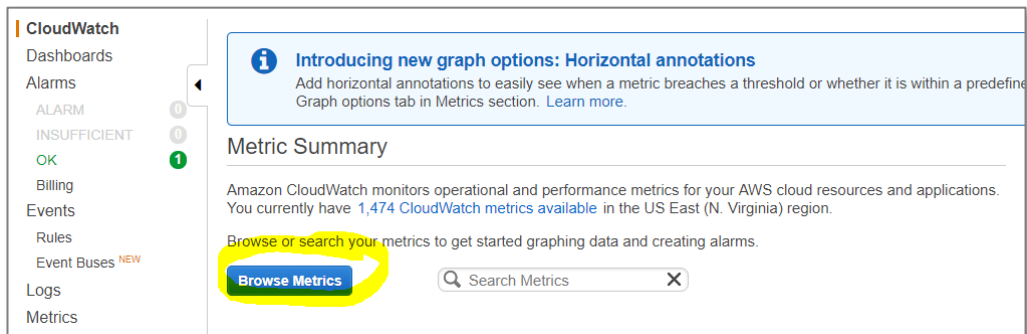
```
[ec2-user@ip-172-31-39-252 aws-scripts-mon]$ crontab -e
no crontab for ec2-user - using an empty one
crontab: installing new crontab
[ec2-user@ip-172-31-39-252 aws-scripts-mon]$ |
```

## Viewing Metrics in CloudWatch

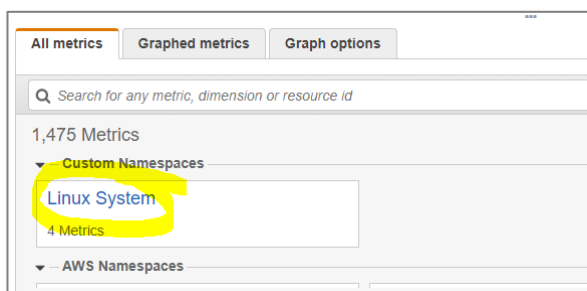
Wait a few minutes for data to be pushed to CloudWatch

Go to the CloudWatch Console

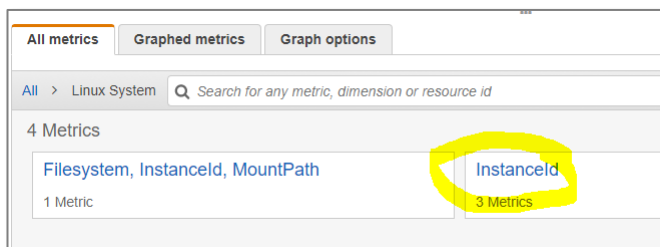
Click "Browse Metrics"



### Select "Linux System"

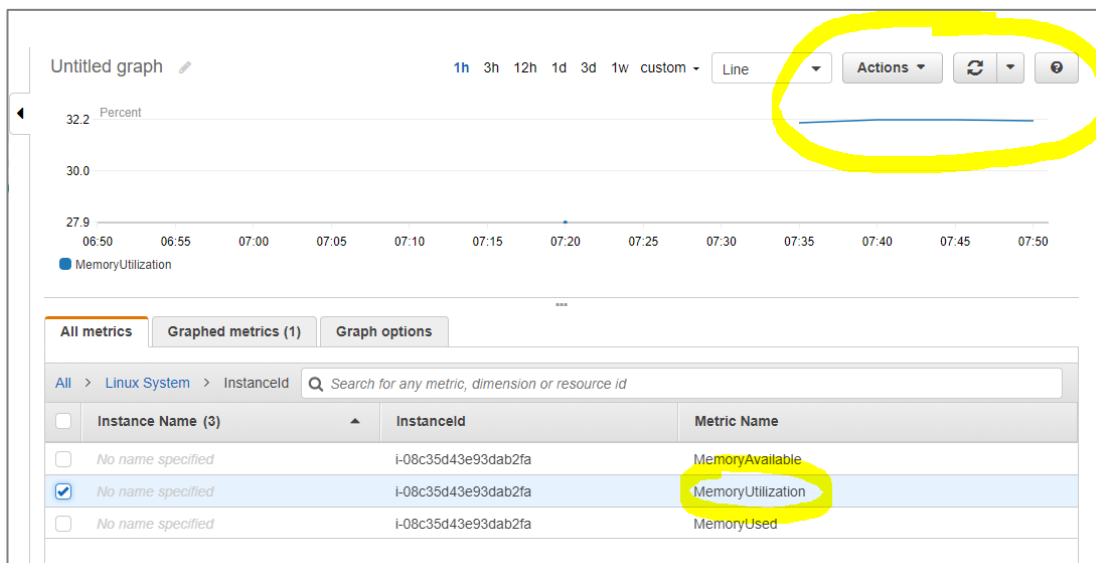


### Select "Instance ID"



### Select "Memory Utilization"

You will now see a graph of the metric pushed from the EC2 instance every minute



## Clean Up

Now delete the EC2 Instance so that you don't get billed.