

Cloud Metric using Cloud Watch

Step 1: Generate policy using

<https://awspolicygen.s3.amazonaws.com/policygen.html>

AWS Policy Generator

The AWS Policy Generator is a tool that enables you to create policies that control access to Amazon Web Services (AWS) products and resources. For more information about creating policies, see [key concepts in Using AWS Identity and Access Management](#). Here are sample policies.

Step 1: Select Policy Type

A Policy is a container for permissions. The different types of policies you can create are an IAM Policy, an S3 Bucket Policy, an SNS Topic Policy, a VPC Endpoint Policy, and an SQS Queue Policy.

Select Type of Policy IAM Policy

Step 2: Add Statement(s)

A statement is the formal description of a single permission. See a [description of elements](#) that you can use in statements.

Effect ☒ Allow ☐ Deny

AWS Service Amazon EC2 ☐ All Services ^(*)

Use multiple statements to add permissions for more than one service.

Actions -- Select Actions -- ☐ All Actions ^(*)

Amazon Resource Name (ARN)

ARN should follow the following format: arn:aws:ec2:<region>:<account>:<resourceType>:<resourcePath>.
Use a comma to separate multiple values.

Add Conditions (Optional)

Add Statement

You added the following statements. Click the button below to Generate a policy.

Effect	Action	Resource	Conditions
Allow	<ul style="list-style-type: none">cloudwatch:GetMetricStatisticscloudwatch:ListMetricscloudwatch:PutMetricData	*	None
Allow	<ul style="list-style-type: none">ec2:DescribeTags	*	None

Step 3: Generate Policy

A policy is a document (written in the [Access Policy Language](#)) that acts as a container for one or more statements.

Generate Policy

Start Over

Step:2 Create a new policy and paste the json code there

Create policy

1 2 3

A policy defines the AWS permissions that you can assign to a user, group, or role. You can create and edit a policy in the visual editor and using JSON. [Learn more](#)

Visual editor

JSON

[Import managed policy](#)

```
8  ],
9  "Effect": "Allow",
10 "Resource": "*",
11 },
12 {
13   "Sid": "Stmt1625816859563",
14   "Action": [
15     "cloudwatch:GetMetricData",
16     "cloudwatch:ListMetrics",
17     "cloudwatch:PutMetricData"
18   ],
19   "Effect": "Allow",
20   "Resource": "*"
21 }
22 ]
23 }
```

Security: 0 Errors: 0 Warnings: 0 Suggestions: 0

Step 3: Review and Create the policy

Review policy

Name* CloudMetrics

Use alphanumeric and '+@,._-' characters. Maximum 128 characters.

Description CloudMetrics logs

Maximum 1000 characters. Use alphanumeric and '+@,._-' characters.

Summary

Filter

Service	Access level	Resource	Request condition
Allow (2 of 285 services) Show remaining 283			
CloudWatch	Limited: List, Read, Write	All resources	None
EC2	Limited: Read	All resources	None

Tags

Key	Value
-----	-------

No tags associated with the resource.

* Required

[Cancel](#)

[Previous](#)

[Create policy](#)

Step 4: Create a role in AWS Service EC2 and attach the particular policy

Create role

1 2 3 4

Attach permissions policies

Choose one or more policies to attach to your new role.

[Create policy](#)



Filter policies		Showing 1 result
CloudMetr		
	Policy name	Used as
<input checked="" type="checkbox"/>	CloudMetrics	None

Set permissions boundary

* Required

[Cancel](#)

[Previous](#)

[Next: Tags](#)

Step5:And Create a role

Create role



Review

Provide the required information below and review this role before you create it.

Role name* CloudMetriclogs
Use alphanumeric and '+=, @, _' characters. Maximum 64 characters.

Role description Allows EC2 instances to call AWS services on your behalf.
Maximum 1000 characters. Use alphanumeric and '+=, @, _' characters.

Trusted entities AWS service: ec2.amazonaws.com

Policies CloudMetrics [↗](#)

Permissions boundary Permissions boundary is not set

No tags were added.

* Required

Cancel

Previous

Create role

Step:6 Create Ami Linux instance and attach IAM 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 the lower pricing, assign an access manag

Number of instances	1	Launch into Auto Scaling Group
Purchasing option	<input type="checkbox"/> Request Spot instances	
Network	vpc-36d6245d (default)	Create new VPC
Subnet	No preference (default subnet in any Availability Zone)	Create new subnet
Auto-assign Public IP	Use subnet setting (Enable)	
Placement group	<input type="checkbox"/> Add instance to placement group	
Capacity Reservation	Open	
Domain join directory	No directory	Create new directory
IAM role	CloudMetriclogs	Create new IAM role
Shutdown behavior	Stop	
Stop - Hibernate behavior	<input type="checkbox"/> Enable hibernation as an additional stop behavior	
Enable termination protection	<input type="checkbox"/> Protect against accidental terminatio	

Step:7 Install all this in Amazon Linux AMI

Install Curl

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

you want to store the monitoring scripts and run the following command to download them:

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

Step:8 The following example performs a simple test run without posting data to CloudWatch.

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

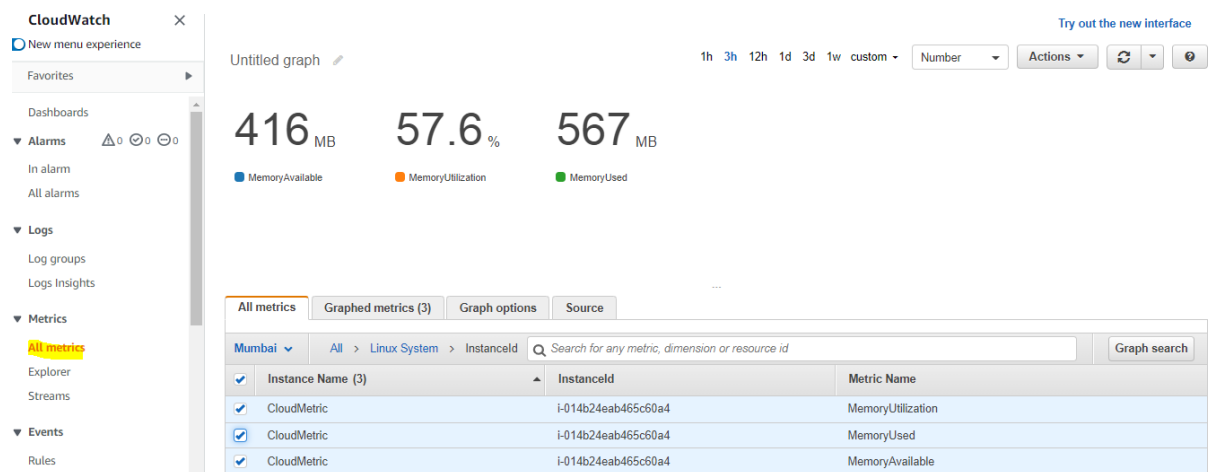
```
[ec2-user@ip-172-31-3-71 aws-scripts-mon]$ ./mon-put-instance-data.pl --mem-util --verify --verbose
MemoryUtilization: 14.3880376445537 (Percent)
No credential methods are specified. Trying default IAM role.
Using IAM role <CloudMetriclogs>
Endpoint: https://monitoring.ap-south-1.amazonaws.com
Payload: {"MetricData":[{"Timestamp":1625821598,"Dimensions":[{"Value":"i-014b24eab465c60a4","Name":"InstanceId"}],"Value":14.3880376445537,"Unit":"Percent","MetricName":"MemoryUtilization"},"Namespace":"System/Linux","__type":"com.amazonaws.cloudwatch.v2010_08_01#PutMetricDataInput"]}
Verification completed successfully. No actual metrics sent to CloudWatch.
```

Step:9 To report the log one time in cloudwatch

```
[ec2-user@ip-172-31-3-71 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: 1c84f326-9b9a-4eea-bde3-693f37477505

[ec2-user@ip-172-31-3-71 aws-scripts-mon]$ free
              total        used        free      shared  buff/cache   available
Mem:      1006892      95608      442056          440       469228       771108
Swap:              0              0
[ec2-user@ip-172-31-3-71 aws-scripts-mon]$ free -m
              total        used        free      shared  buff/cache   available
Mem:           983           93         431          0         458         752
Swap:              0              0
```

Step:10 Go to cloud watch log metric -System Linux



Additional Commands: -

You can add in contab for continuous monitoring

For ex

```
*/5 * * * * ~/aws-scripts-mon/mon-put-instance-data.pl --mem-used-incl-cache-buff --mem-util --disk-space-util --disk-path=/ --from-cron
```