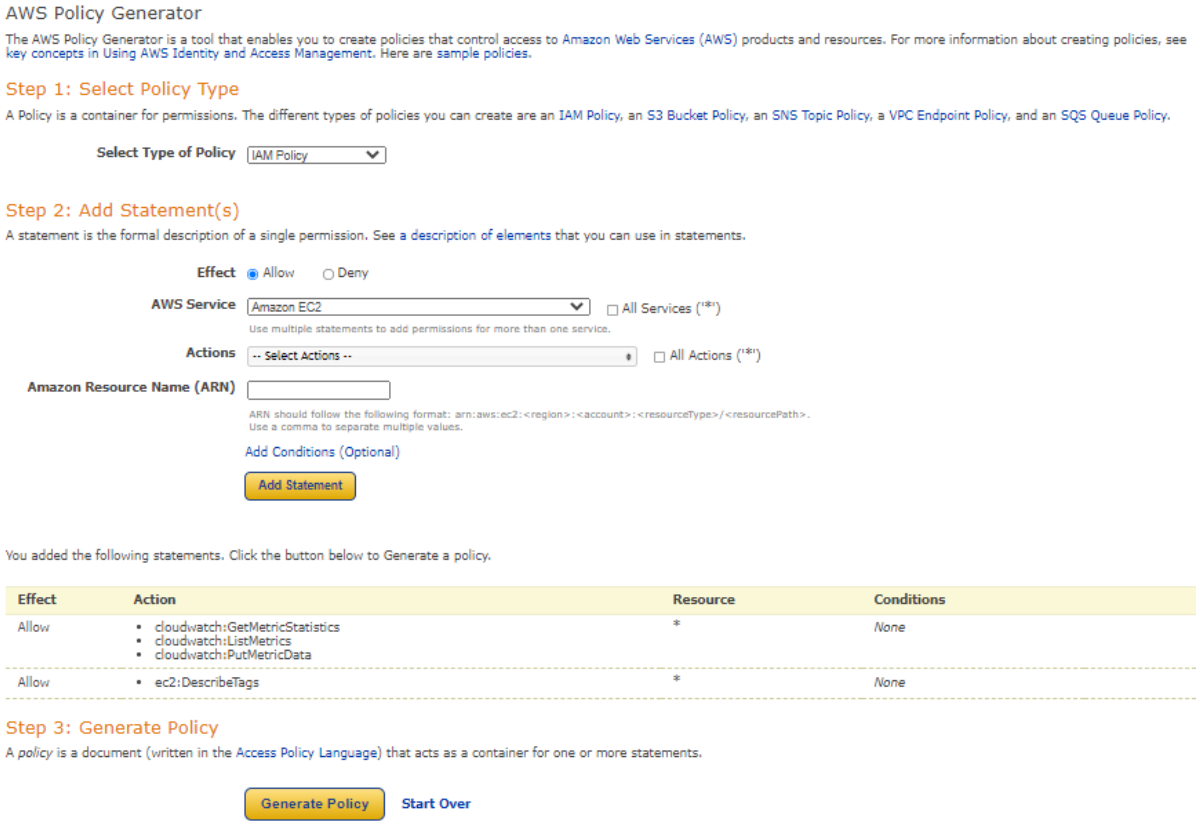
Cloud Metric using Cloud Watch

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.

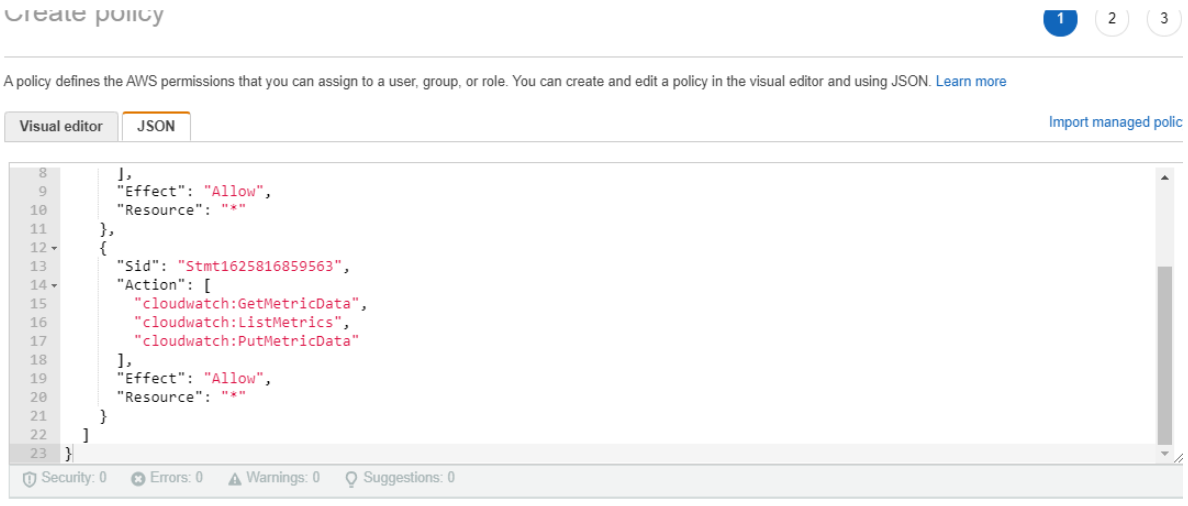
Effect	Action	Resource	Conditions
Allow	• cloudwatch:GetMetricStatistics • cloudwatch:ListMetrics • cloudwatch:PutMetricData	*	None
Allow	• ec2:DescribeTags	*	None

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

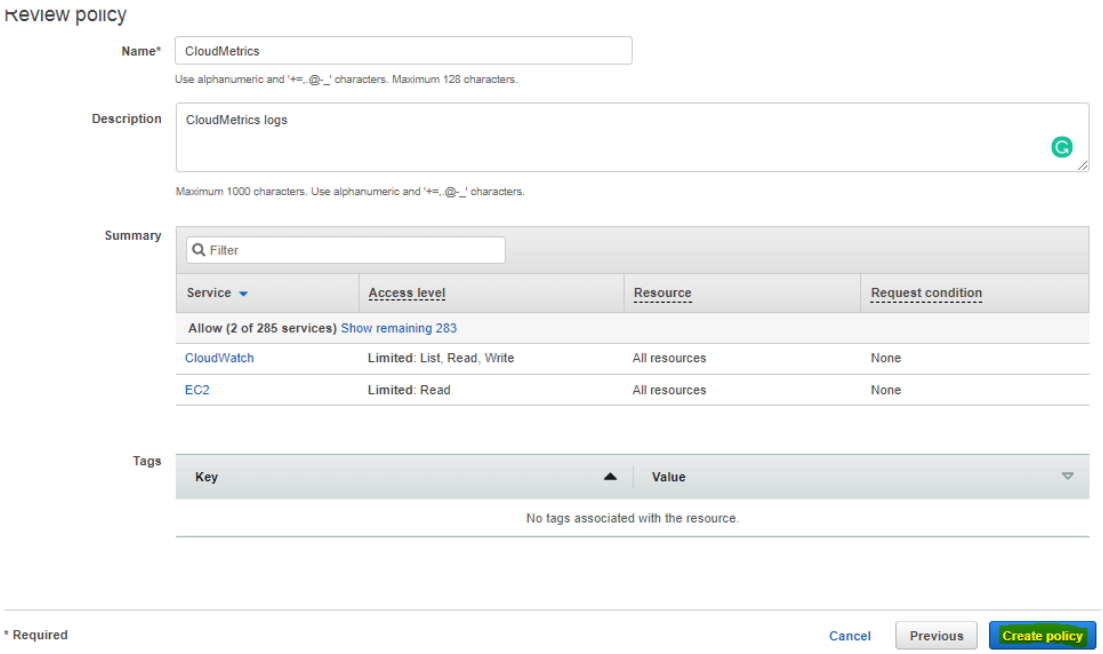
A policy defines the AWS permissions that you can assign to a user, group, or role. You can create and edit a policy in the visual editor and using JSON. Learn more

Visual editor | JSON Import managed polic

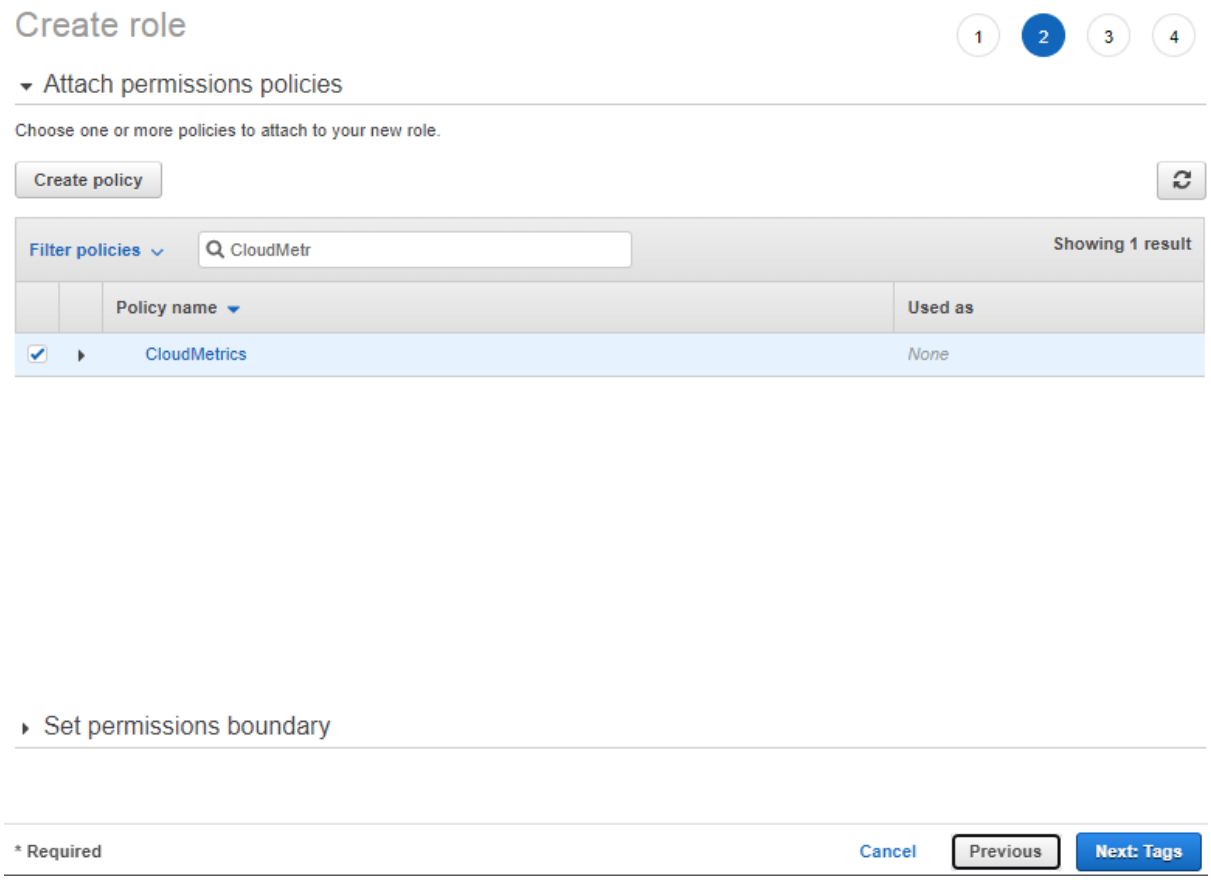
```
      ],
      "Effect": "Allow",
      "Resource": "*"
    },
    {
      "Sid": "Stmt1625816859563",
      "Action": [
        "cloudwatch:GetMetricData",
        "cloudwatch:ListMetrics",
        "cloudwatch:PutMetricData"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

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

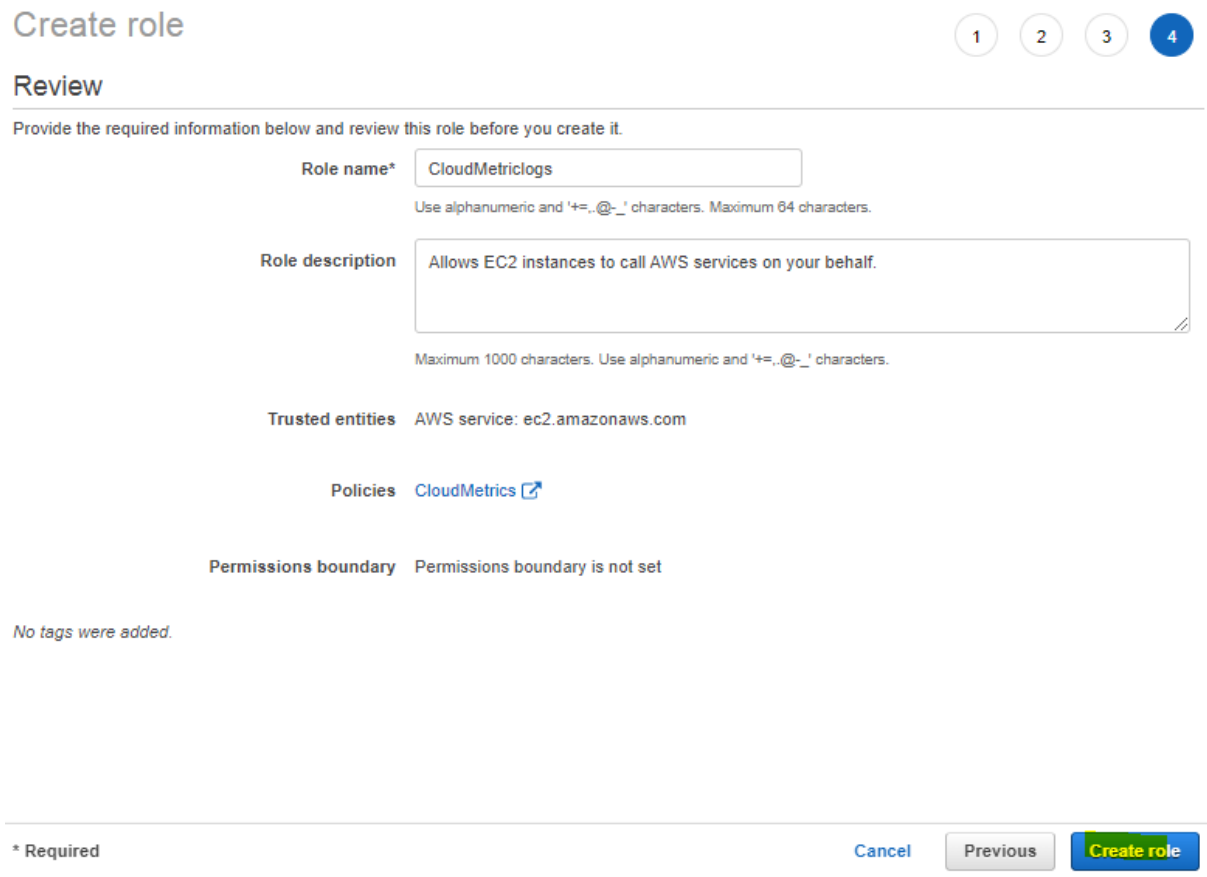
Step 3: Review and Create the policy



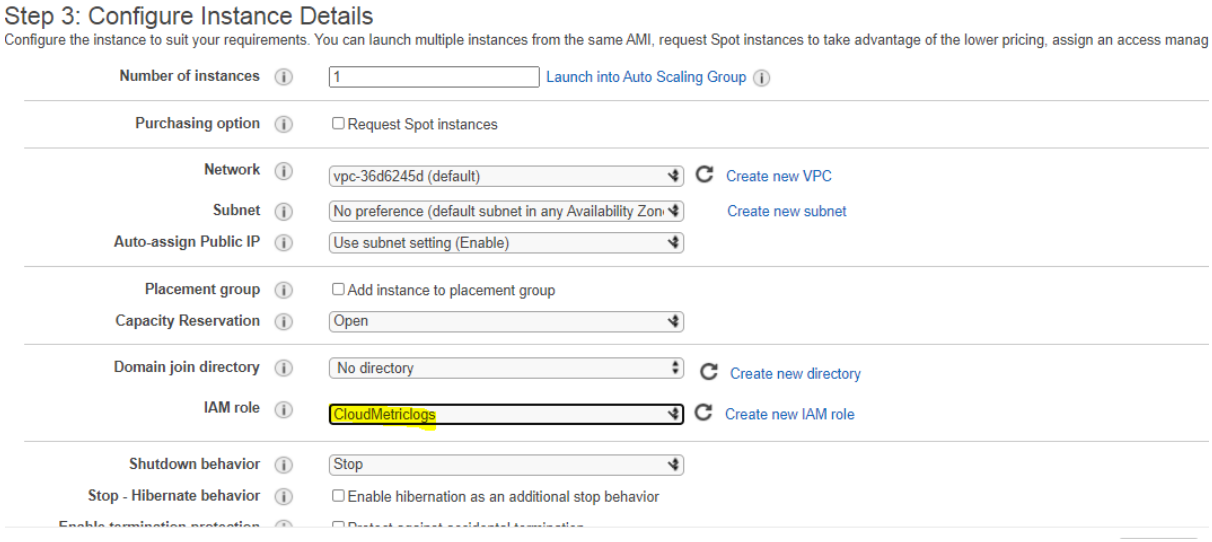
Step4: Create a role in **AWS Service EC2** and attach the particular policy



Step5:And Create a role



Step:6 Create Ami Linux instance and attach IAM role:



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.
```

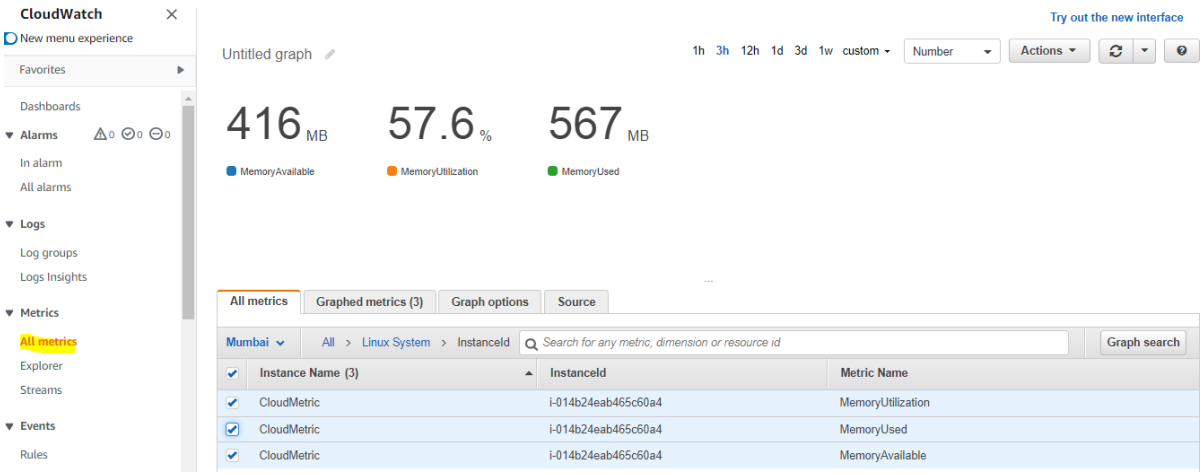
Step:9 To report the log one time in cloudwatch

```
[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           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           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