**37.AWS-B30-CloudWatchMonitoring**

--- in this session, we will discuss about cloudwatch.

--- In cloud watch we will learn below topics

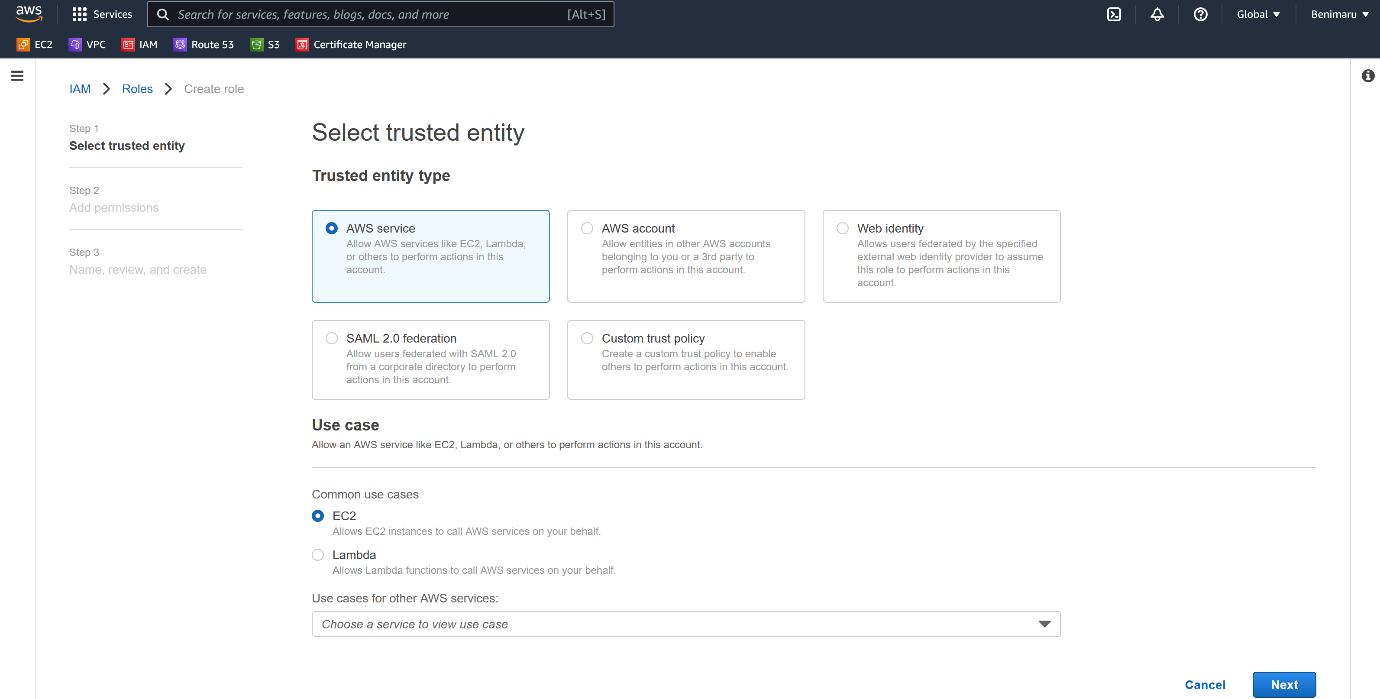
* + - Installing cloudwatch agent using SSM (system manager)
    - Configuring alarms
    - Monitoring logs using aws logs
    - Events (cloudwatch)

**Create a role with below policies**

--- create a role with below policies

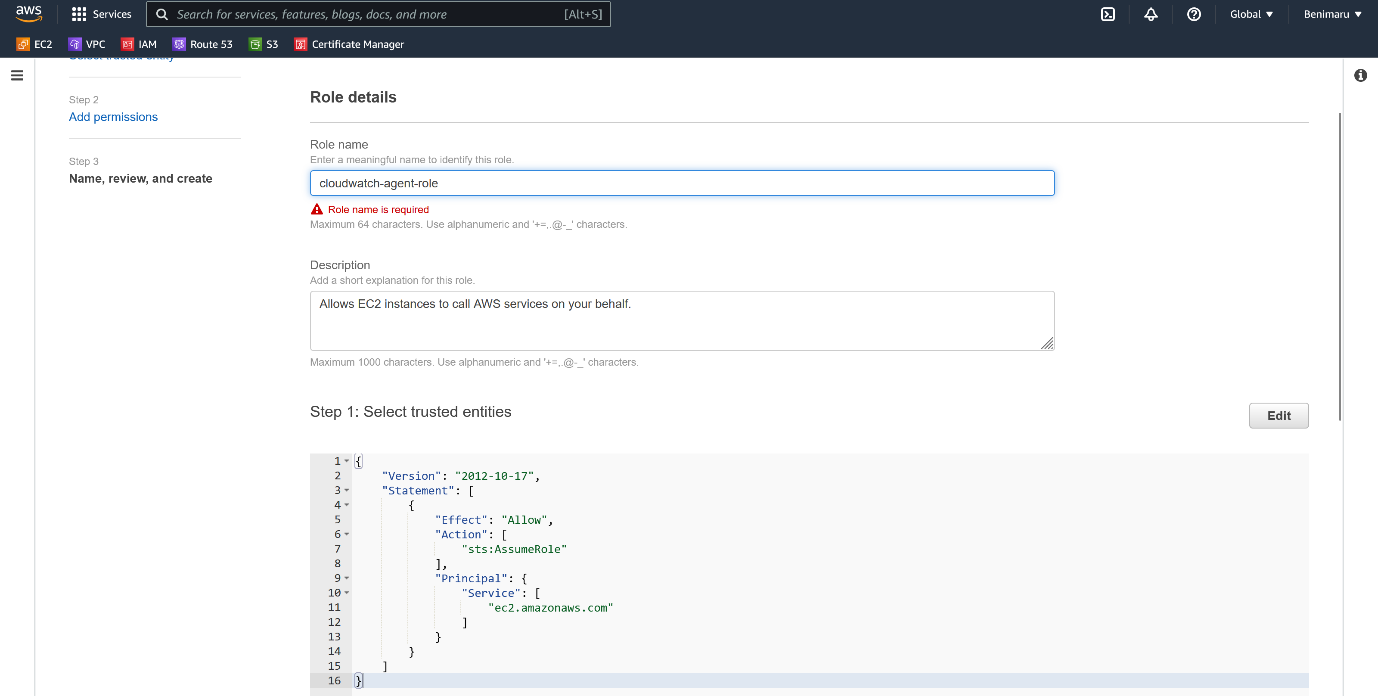
* AmazonEC2RoleforSSM
* CloudWatchAgentAdminPolicy
* CloudWatchAgentServerPolicy

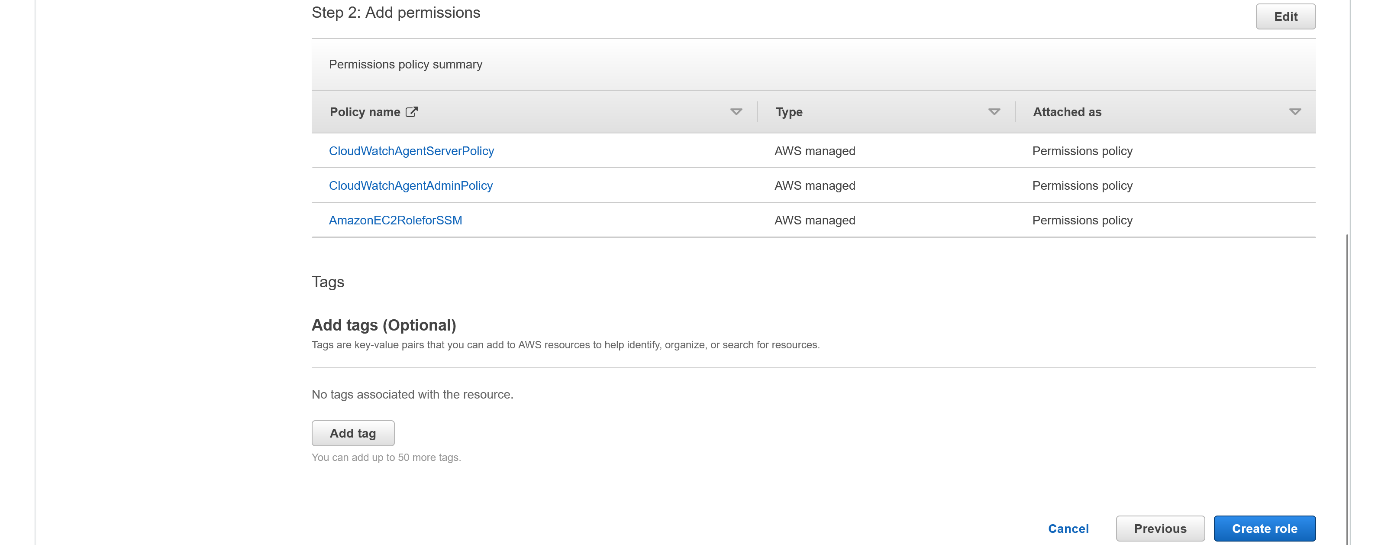
--- go to IAM and click on create role



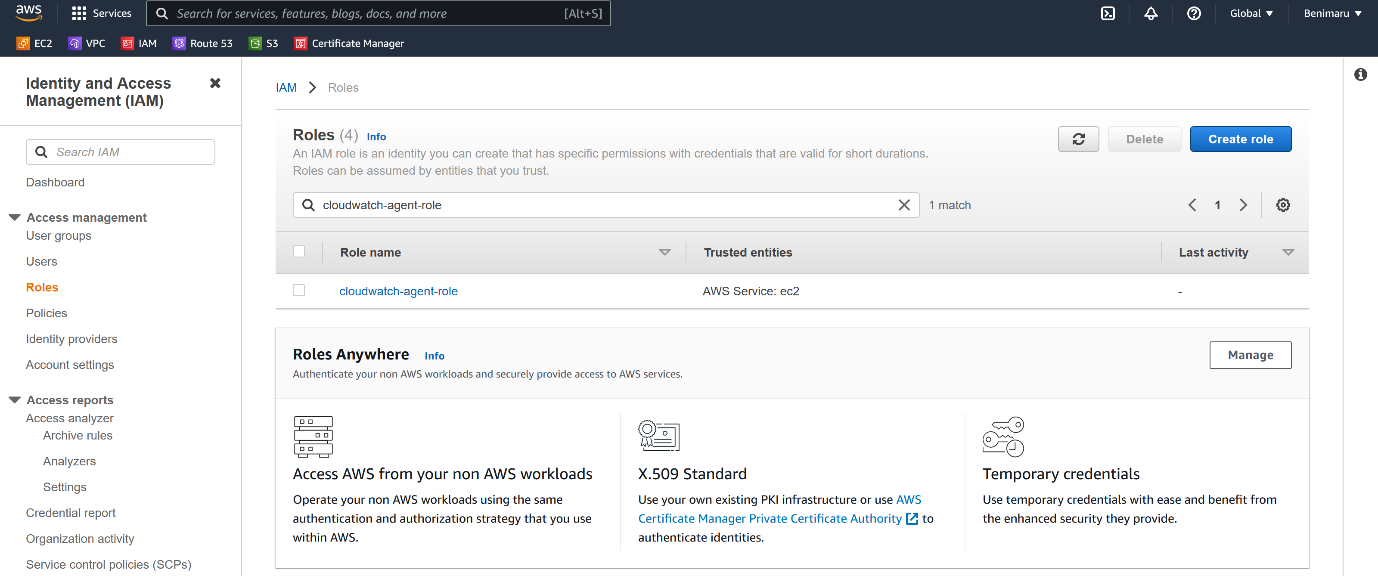
--- click on next and next page add below policies

* AmazonEC2RoleforSSM
* CloudWatchAgentAdminPolicy
* CloudWatchAgentServerPolicy





--- **note** – click on create role.



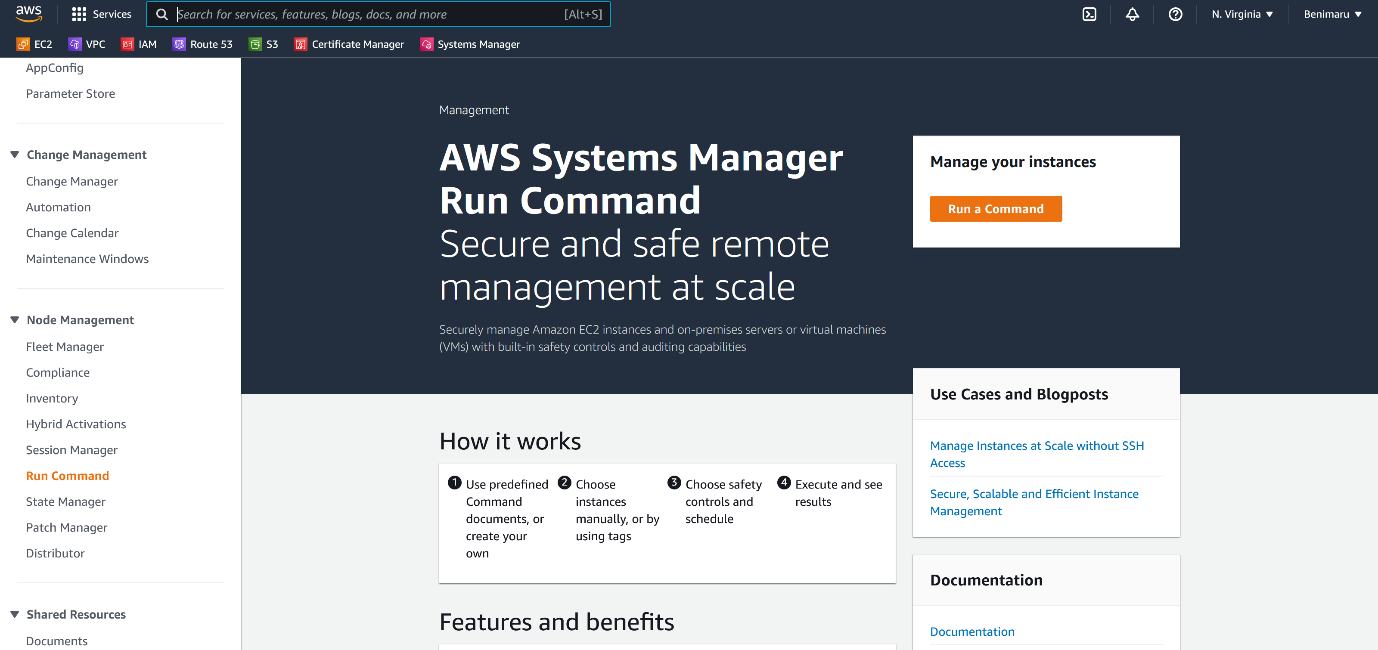
**Assign roles to machines**

--- Deploy t2.med or t2.large machine and assign above role while deploying for windows Deploy t2.micro for Amazon linux2

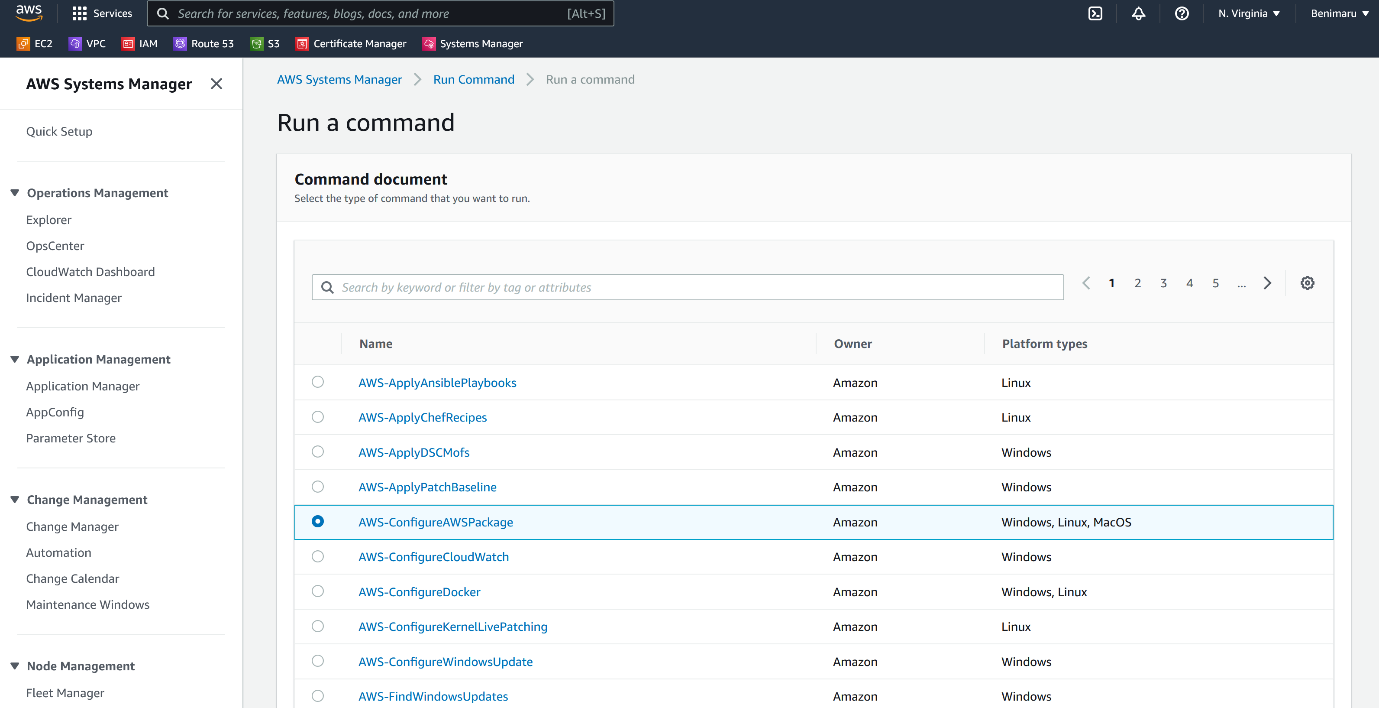
--- **note** – we will use AWS ssm to download the agent on the instance.

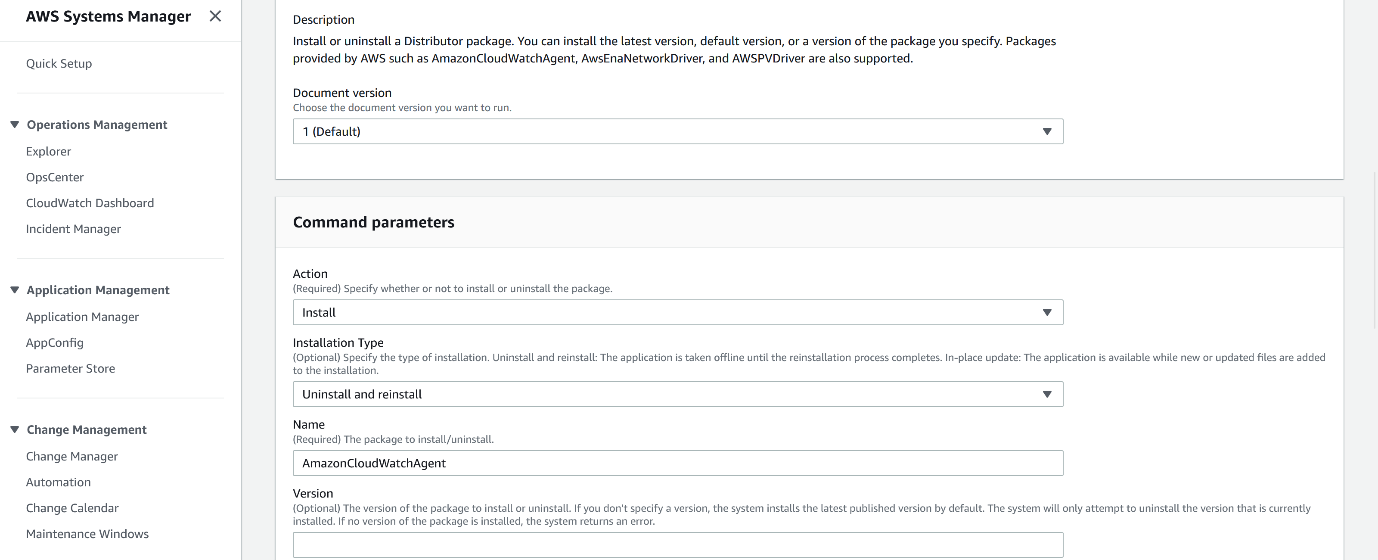
**AWS system manager**

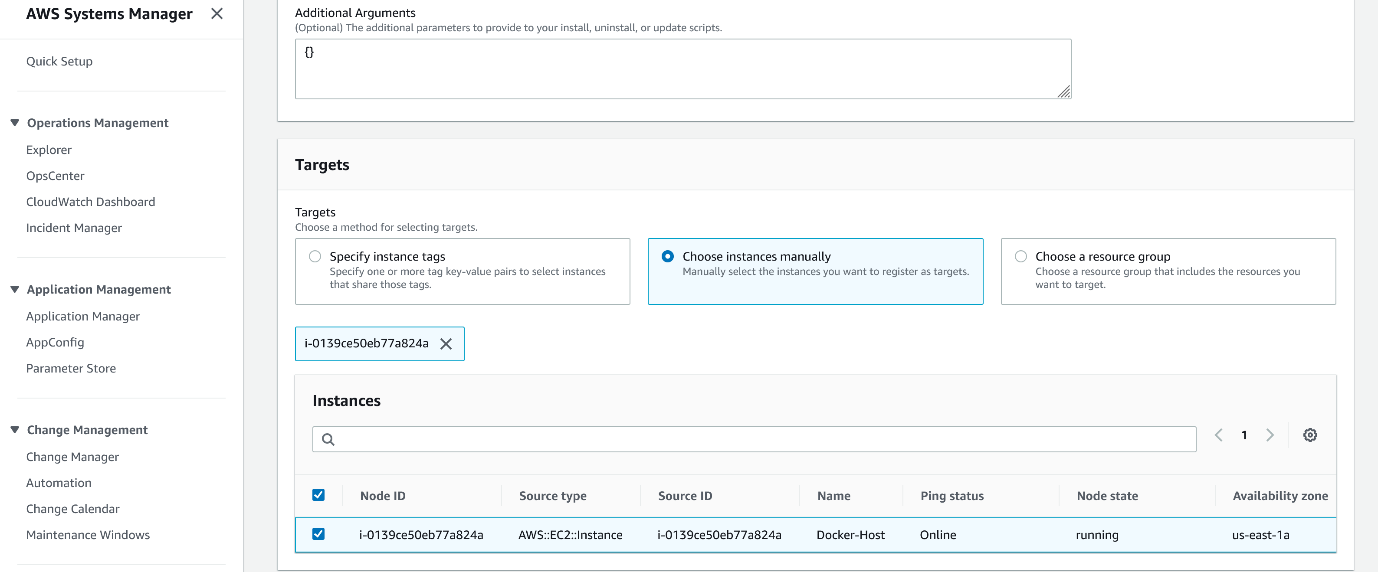
**RUN command**

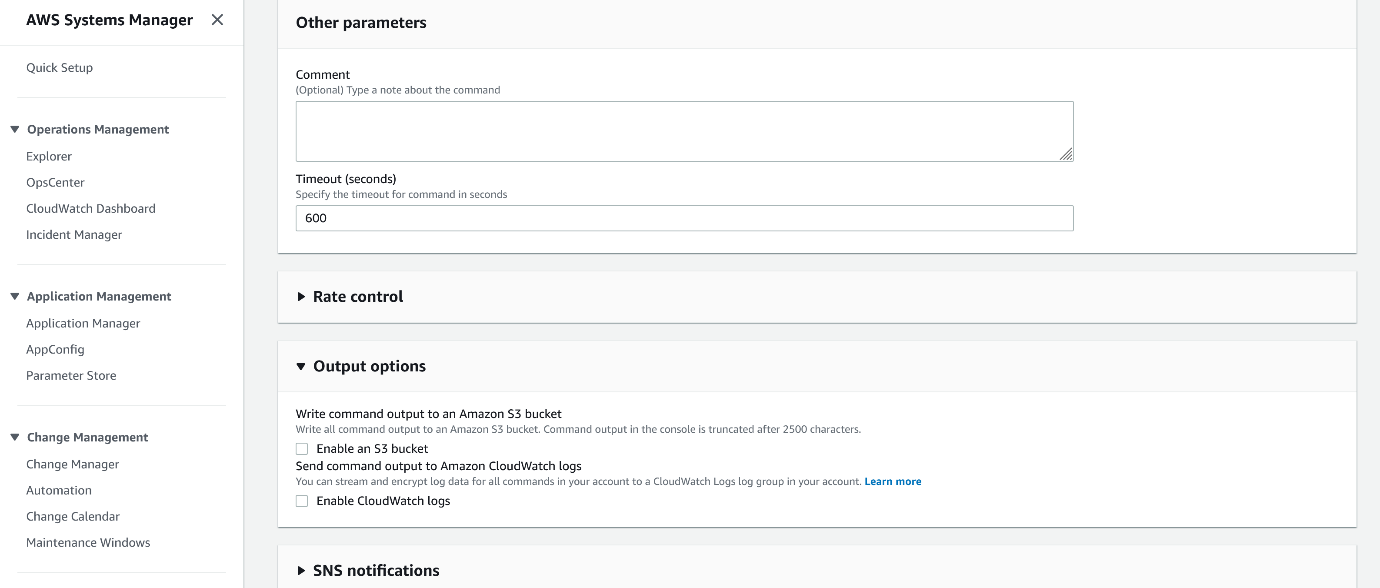


--- click on RUN a command.

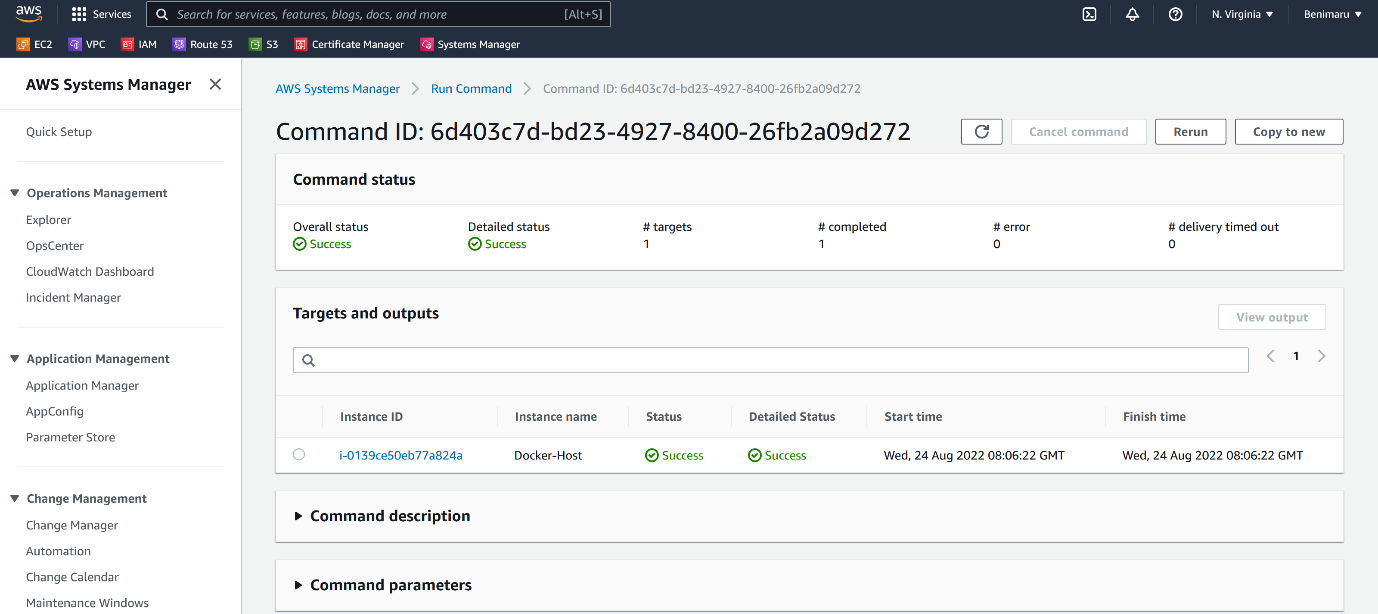








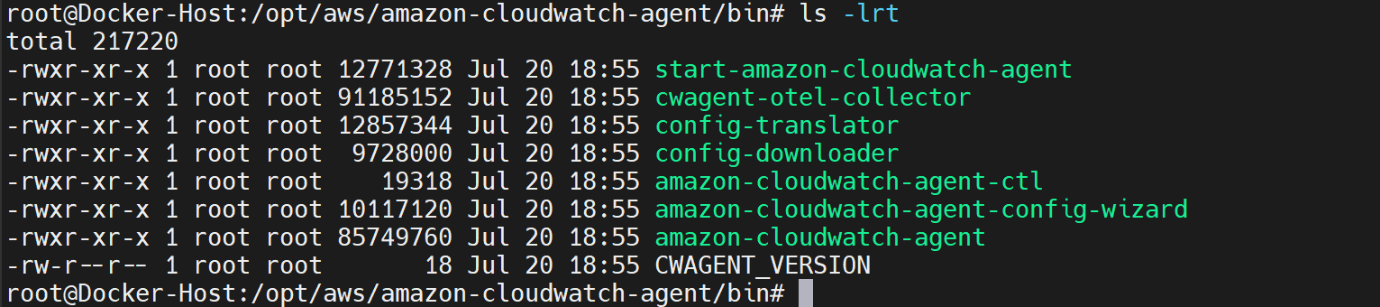
--- Click on run.



**Login into server**

# Cloud watch agent installation path

--- /opt/aws/amazon-cloudwatch-agent/bin



# Start the cloudwatch agent.

--- ./amazon-cloudwatch-agent-config-wizard

root@Docker-Host:/opt/aws/amazon-cloudwatch-agent/bin# ./amazon-cloudwatch-agent-config-wizard

================================================================

= Welcome to the Amazon CloudWatch Agent Configuration Manager =

=                                                              =

= CloudWatch Agent allows you to collect metrics and logs from =

= your host and send them to CloudWatch. Additional CloudWatch =

= charges may apply.                                           =

================================================================

On which OS are you planning to use the agent?

1. linux

2. windows

3. darwin

default choice: [1]:

Trying to fetch the default region based on ec2 metadata...

Are you using EC2 or On-Premises hosts?

1. EC2

2. On-Premises

default choice: [1]:

Which user are you planning to run the agent?

1. root

2. cwagent

3. others

default choice: [1]:

Do you want to turn on StatsD daemon?

1. yes

2. no

default choice: [1]:

Which port do you want StatsD daemon to listen to?

default choice: [8125]

What is the collect interval for StatsD daemon?

1. 10s

2. 30s

3. 60s

default choice: [1]:

2

What is the aggregation interval for metrics collected by StatsD daemon?

1. Do not aggregate

2. 10s

3. 30s

4. 60s

default choice: [4]:

3

Do you want to monitor metrics from CollectD? WARNING: CollectD must be installed or the Agent will fail to start

1. yes

2. no

default choice: [1]:

2

Do you want to monitor any host metrics? e.g. CPU, memory, etc.

1. yes

2. no

default choice: [1]:

Do you want to monitor cpu metrics per core?

1. yes

2. no

default choice: [1]:

Do you want to add ec2 dimensions (ImageId, InstanceId, InstanceType, AutoScalingGroupName) into all of your metrics if the info is available?

1. yes

2. no

default choice: [1]:

Do you want to aggregate ec2 dimensions (InstanceId)?

1. yes

2. no

default choice: [1]:

Would you like to collect your metrics at high resolution (sub-minute resolution)? This enables sub-minute resolution for all metrics, but you can customize for specific metrics in the output json file.

1. 1s

2. 10s

3. 30s

4. 60s

default choice: [4]:

3

Which default metrics config do you want?

1. Basic

2. Standard

3. Advanced

4. None

default choice: [1]:

3

Current config as follows:

{

        "agent": {

                "metrics\_collection\_interval": 30,

                "run\_as\_user": "root"

        },

        "metrics": {

                "aggregation\_dimensions": [

                        [

                                "InstanceId"

                        ]

                ],

                "append\_dimensions": {

                        "AutoScalingGroupName": "${aws:AutoScalingGroupName}",

                        "ImageId": "${aws:ImageId}",

                        "InstanceId": "${aws:InstanceId}",

                        "InstanceType": "${aws:InstanceType}"

                },

                "metrics\_collected": {

                        "cpu": {

                                "measurement": [

                                        "cpu\_usage\_idle",

                                        "cpu\_usage\_iowait",

                                        "cpu\_usage\_user",

                                        "cpu\_usage\_system"

                                ],

                                "metrics\_collection\_interval": 30,

                                "resources": [

                                        "\*"

                                ],

                                "totalcpu": false

                        },

                        "disk": {

                                "measurement": [

                                        "used\_percent",

                                        "inodes\_free"

                                ],

                                "metrics\_collection\_interval": 30,

                                "resources": [

                                        "\*"

                                ]

                        },

                        "diskio": {

                                "measurement": [

                                        "io\_time",

                                        "write\_bytes",

                                        "read\_bytes",

                                        "writes",

                                        "reads"

                                ],

                                "metrics\_collection\_interval": 30,

                                "resources": [

                                        "\*"

                                ]

                        },

                        "mem": {

                                "measurement": [

                                        "mem\_used\_percent"

                                ],

                                "metrics\_collection\_interval": 30

                        },

                        "netstat": {

                                "measurement": [

                                        "tcp\_established",

                                        "tcp\_time\_wait"

                                ],

                                "metrics\_collection\_interval": 30

                        },

                        "statsd": {

                                "metrics\_aggregation\_interval": 30,

                                "metrics\_collection\_interval": 30,

                                "service\_address": ":8125"

                        },

                        "swap": {

                                "measurement": [

                                        "swap\_used\_percent"

                                ],

                                "metrics\_collection\_interval": 30

                        }

                }

        }

}

Are you satisfied with the above config? Note: it can be manually customized after the wizard completes to add additional items.

1. yes

2. no

default choice: [1]:

Do you have any existing CloudWatch Log Agent (http://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/AgentReference.html) configuration file to import for migration?

1. yes

2. no

default choice: [2]:

Do you want to monitor any log files?

1. yes

2. no

default choice: [1]:

2

Saved config file to /opt/aws/amazon-cloudwatch-agent/bin/config.json successfully.

Current config as follows:

{

        "agent": {

                "metrics\_collection\_interval": 30,

                "run\_as\_user": "root"

        },

        "metrics": {

                "aggregation\_dimensions": [

                        [

                                "InstanceId"

                        ]

                ],

                "append\_dimensions": {

                        "AutoScalingGroupName": "${aws:AutoScalingGroupName}",

                        "ImageId": "${aws:ImageId}",

                        "InstanceId": "${aws:InstanceId}",

                        "InstanceType": "${aws:InstanceType}"

                },

                "metrics\_collected": {

                        "cpu": {

                                "measurement": [

                                        "cpu\_usage\_idle",

                                        "cpu\_usage\_iowait",

                                        "cpu\_usage\_user",

                                        "cpu\_usage\_system"

                                ],

                                "metrics\_collection\_interval": 30,

                                "resources": [

                                        "\*"

                                ],

                                "totalcpu": false

                        },

                        "disk": {

                                "measurement": [

                                        "used\_percent",

                                        "inodes\_free"

                                ],

                                "metrics\_collection\_interval": 30,

                                "resources": [

                                        "\*"

                                ]

                        },

                        "diskio": {

                                "measurement": [

                                        "io\_time",

                                        "write\_bytes",

                                        "read\_bytes",

                                        "writes",

                                        "reads"

                                ],

                                "metrics\_collection\_interval": 30,

                                "resources": [

                                        "\*"

                                ]

                        },

                        "mem": {

                                "measurement": [

                                        "mem\_used\_percent"

                                ],

                                "metrics\_collection\_interval": 30

                        },

                        "netstat": {

                                "measurement": [

                                        "tcp\_established",

                                        "tcp\_time\_wait"

                                ],

                                "metrics\_collection\_interval": 30

                        },

                        "statsd": {

                                "metrics\_aggregation\_interval": 30,

                                "metrics\_collection\_interval": 30,

                                "service\_address": ":8125"

                        },

                        "swap": {

                                "measurement": [

                                        "swap\_used\_percent"

                                ],

                                "metrics\_collection\_interval": 30

                        }

                }

        }

}

Please check the above content of the config.

The config file is also located at /opt/aws/amazon-cloudwatch-agent/bin/config.json.

Edit it manually if needed.

Do you want to store the config in the SSM parameter store?

1. yes

2. no

default choice: [1]:

What parameter store name do you want to use to store your config? (Use 'AmazonCloudWatch-' prefix if you use our managed AWS policy)

default choice: [AmazonCloudWatch-linux]

AmazonCloudWatch-linux-testing

Trying to fetch the default region based on ec2 metadata...

Which region do you want to store the config in the parameter store?

default choice: [us-east-1]

Which AWS credential should be used to send json config to parameter store?

1. ASIAZENOB23YJRH2V7WS(From SDK)

2. Other

default choice: [1]:

Successfully put config to parameter store AmazonCloudWatch-linux-testing.

Program exits now.

--- successfully enabled.

Which default metrics config do you want?

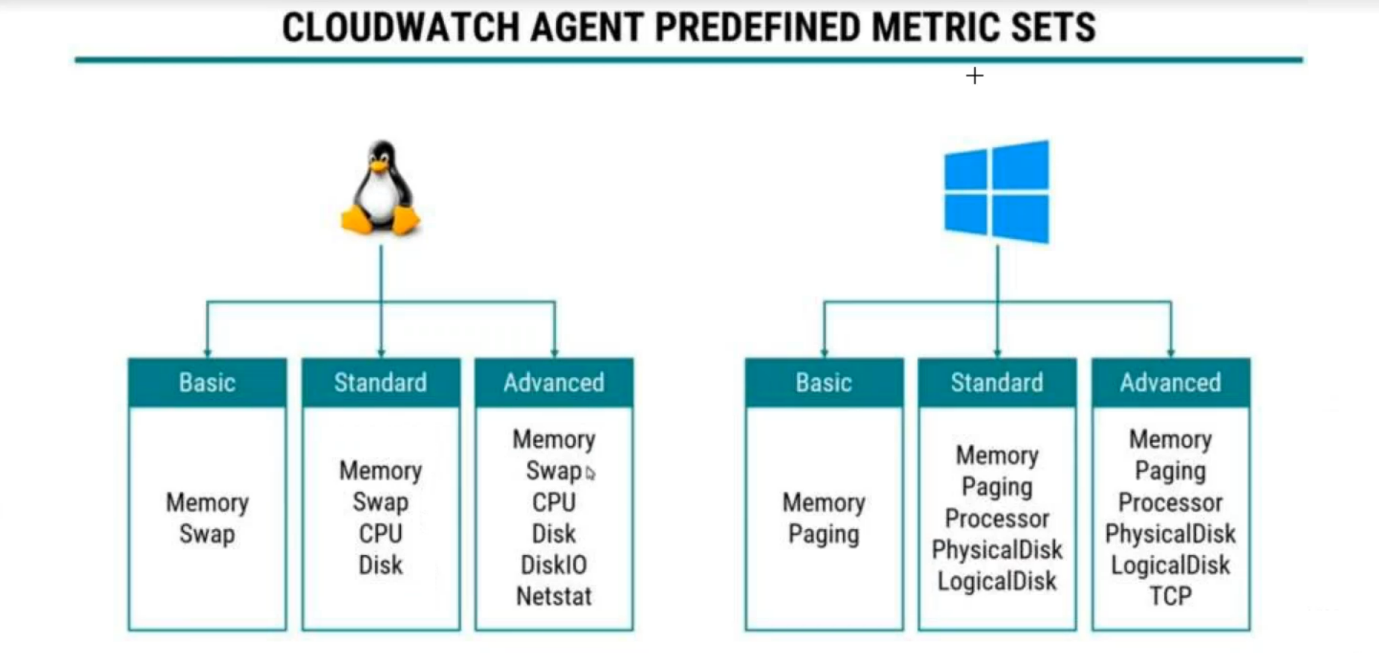
1. Basic

2. Standard

3. Advanced

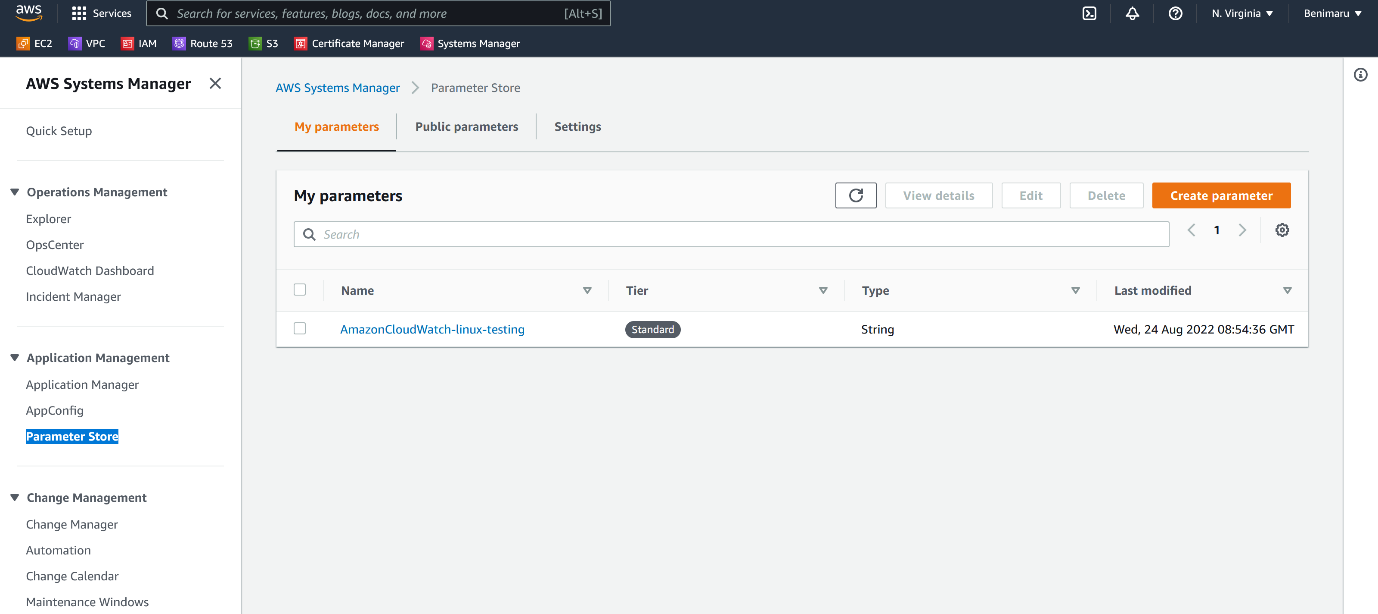
4. None

--- **note** – if you select 3 option, then it will collect below metrics.

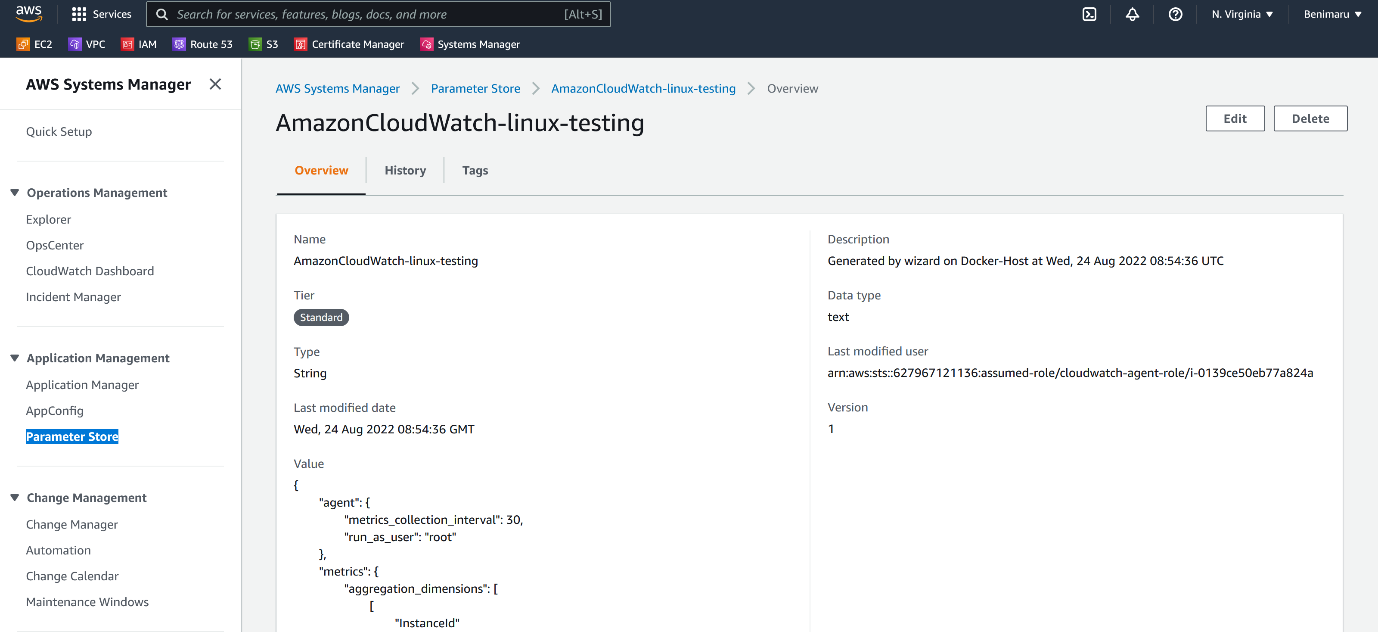


[**Parameter Store**](https://us-east-1.console.aws.amazon.com/systems-manager/parameters?region=us-east-1)

--- ssm



--- click on the name

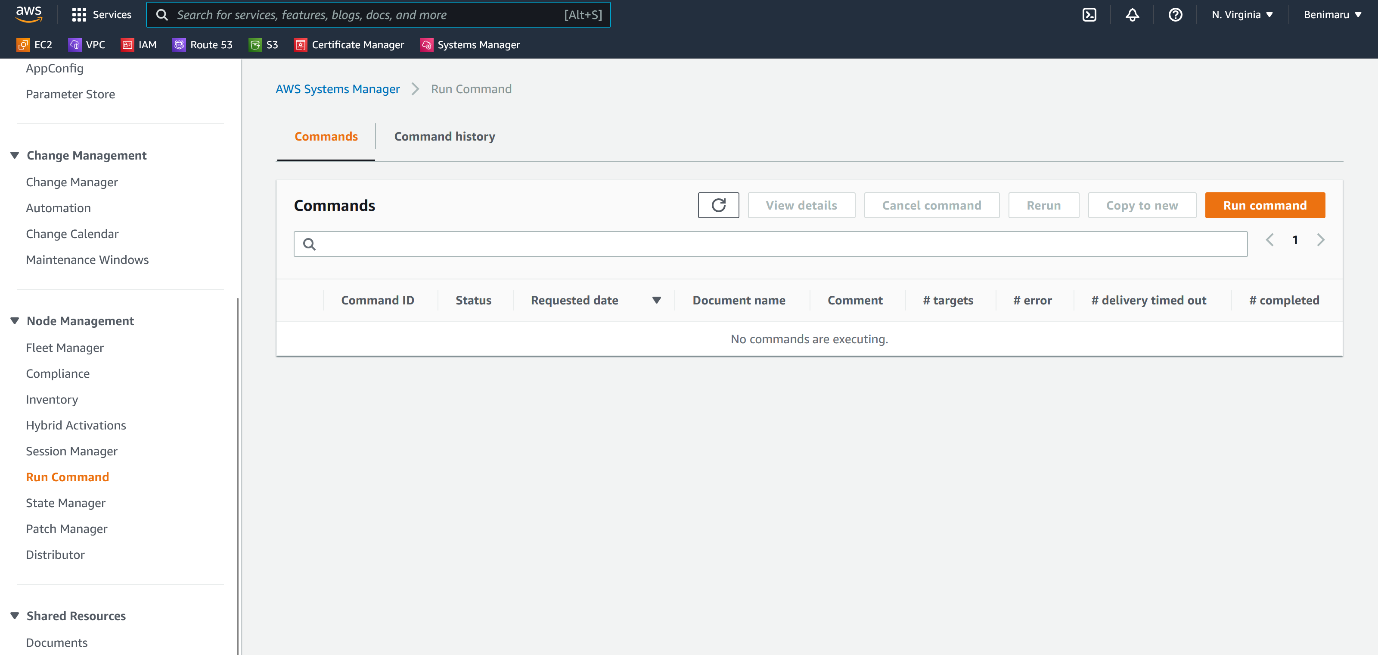


--- **note** – whatever options we selected in the instance while creating, those you can see here.

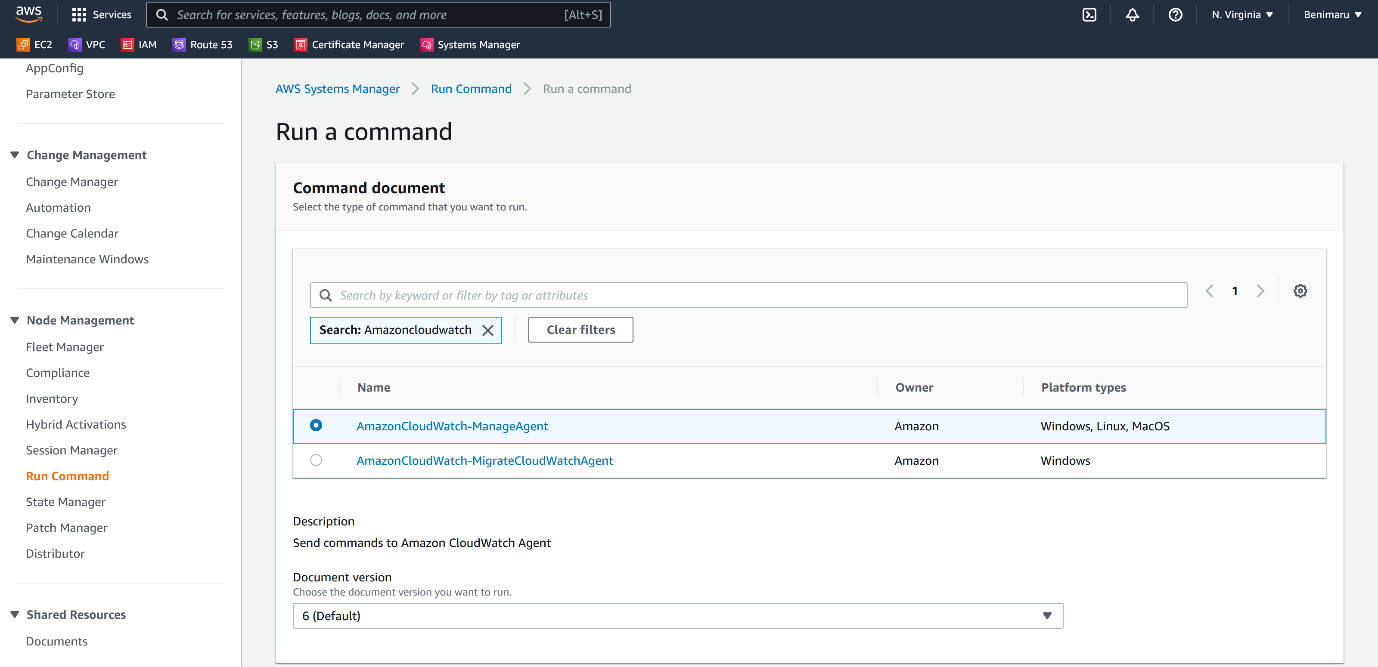
**RUN command**

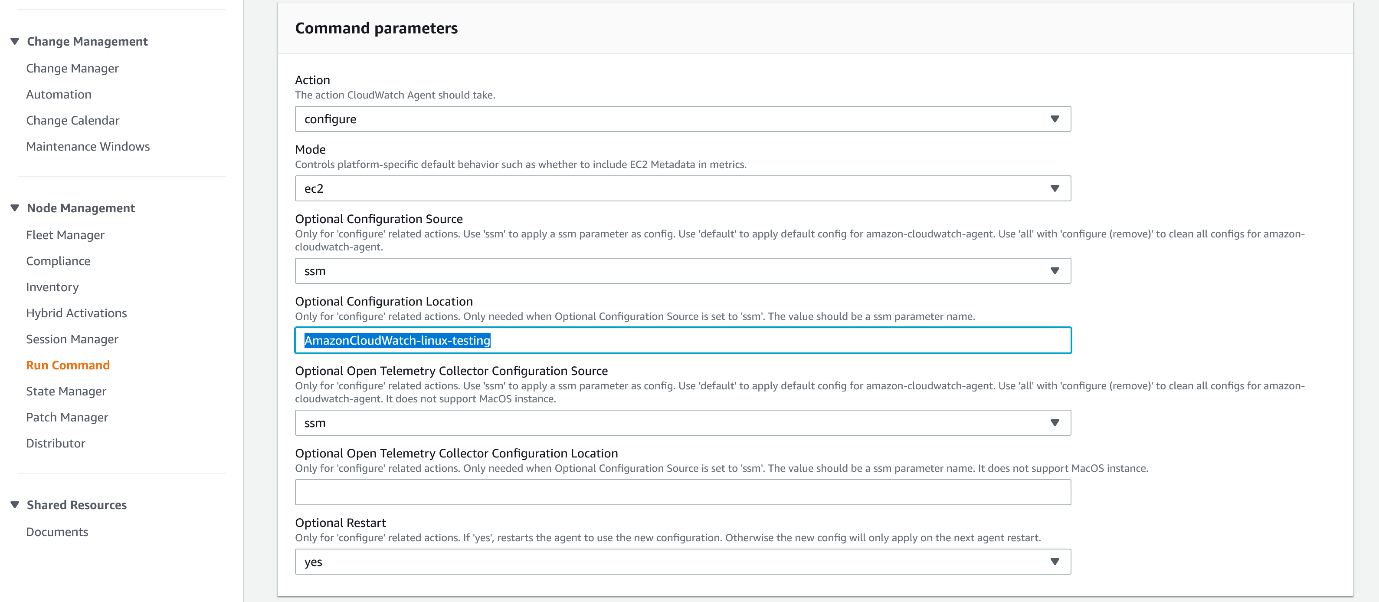
--- **note** – we installed agent on the server and now we need to configure the agent.

--- go to SSM

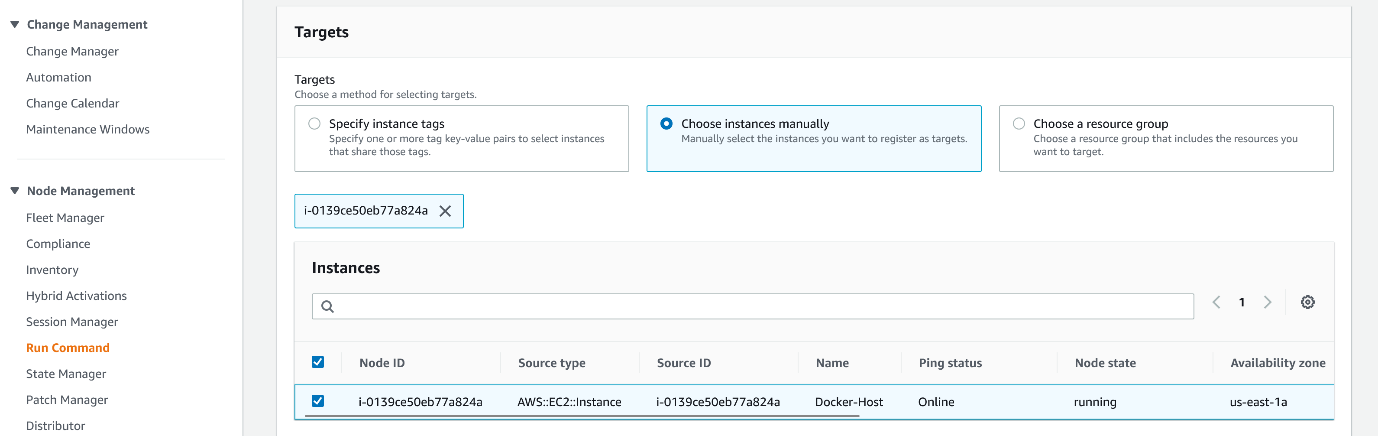


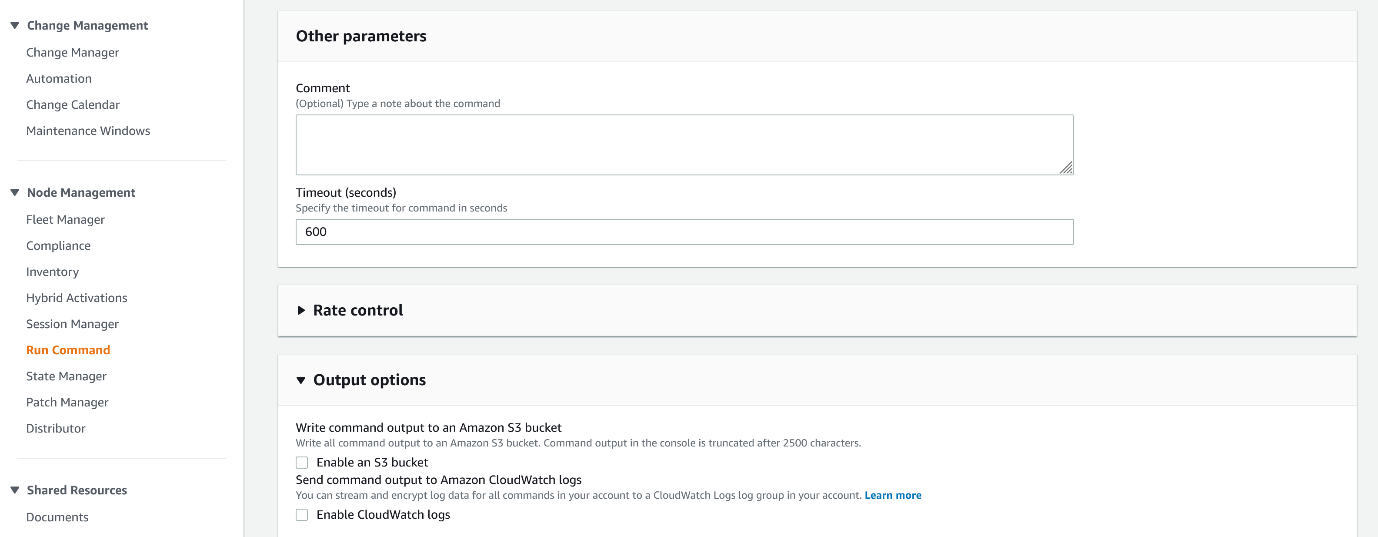
--- click on Run command.



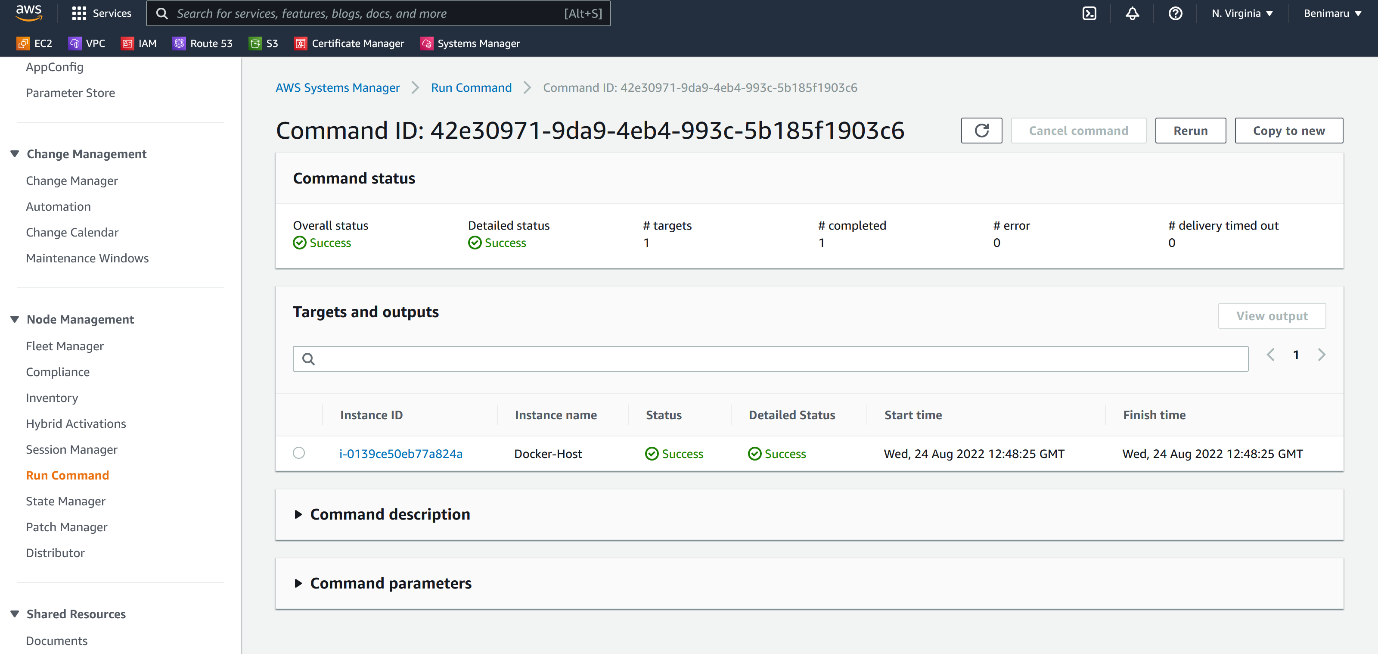


--- **note** – while configuring the cloud watch agent, we have given this name.





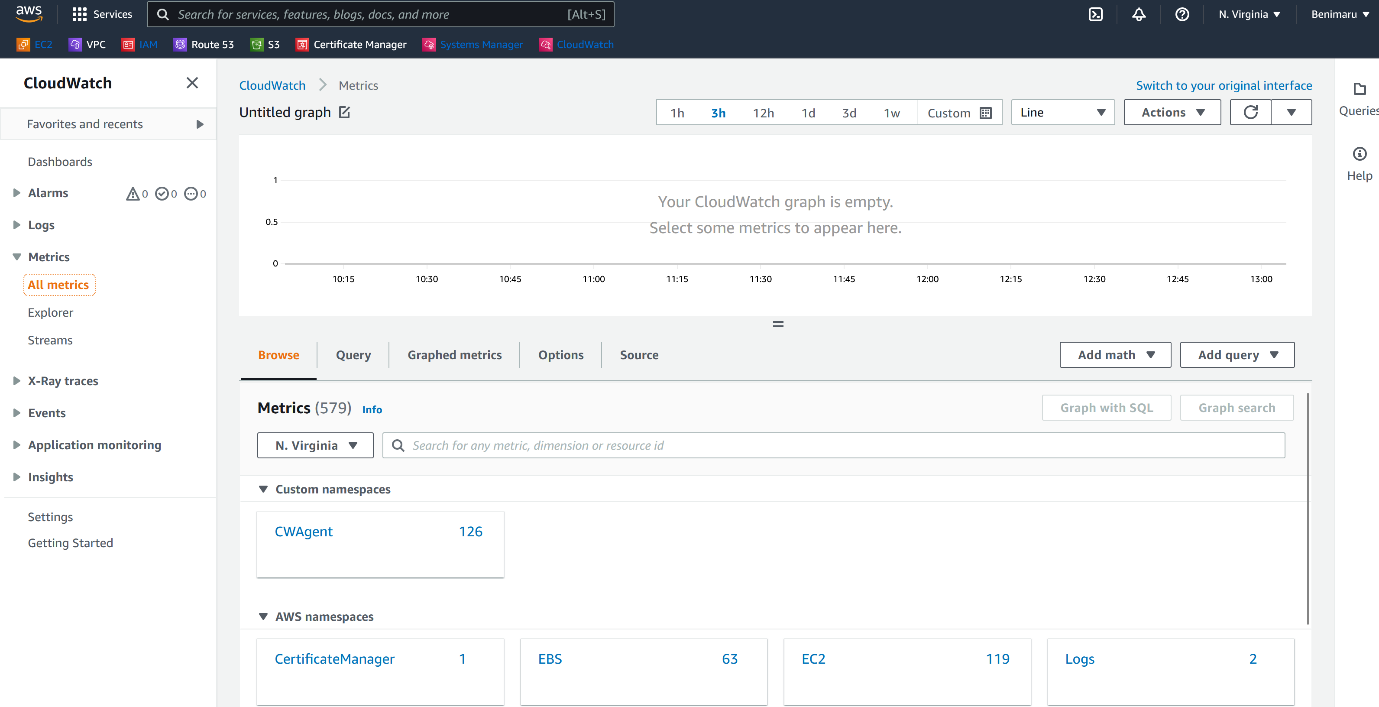
--- **note** – click on run.



--- **note** – it is successful.

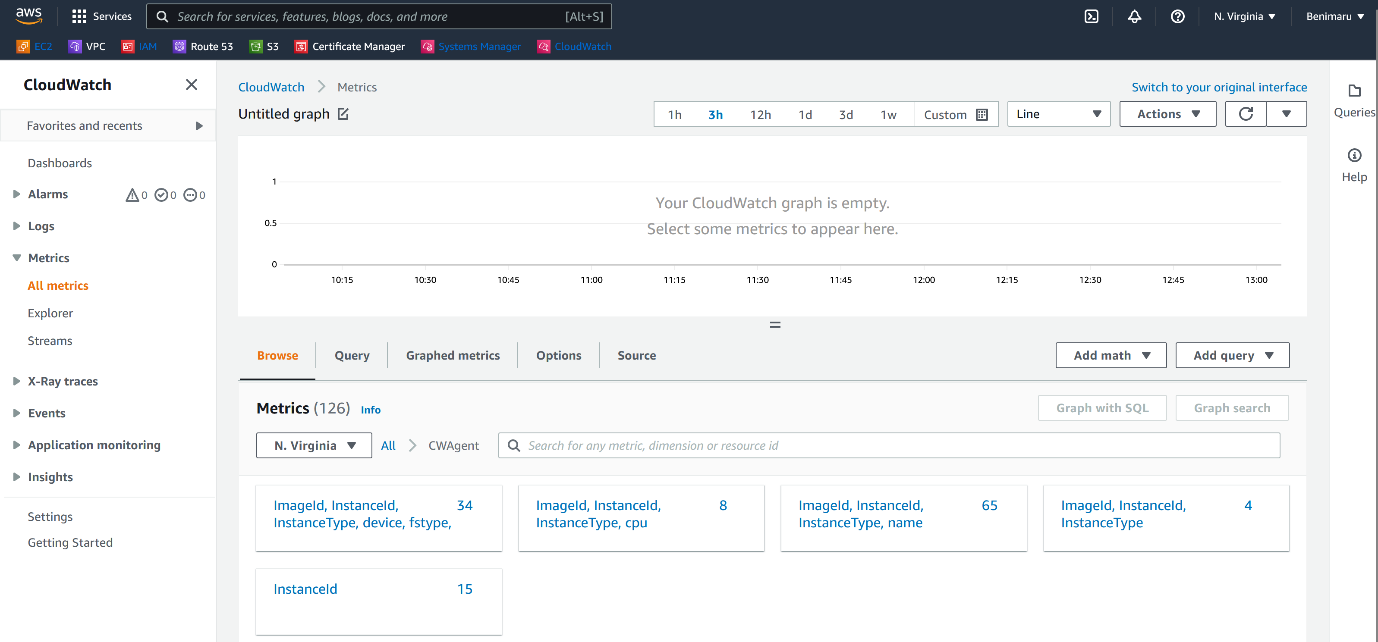
--- **note** – go to the cloudwatch metrics.

**AWS cloudwatch**



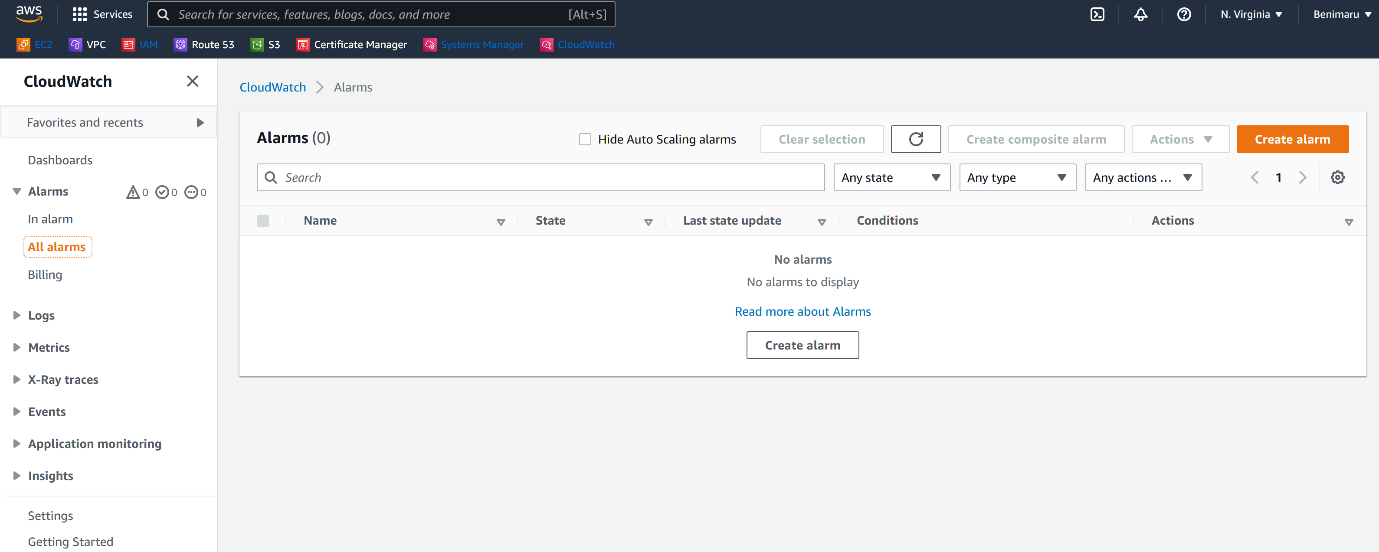
--- **note** – click on all metrics and under custom namespaces you will find **CWAgent**.

--- click on CWAgent.

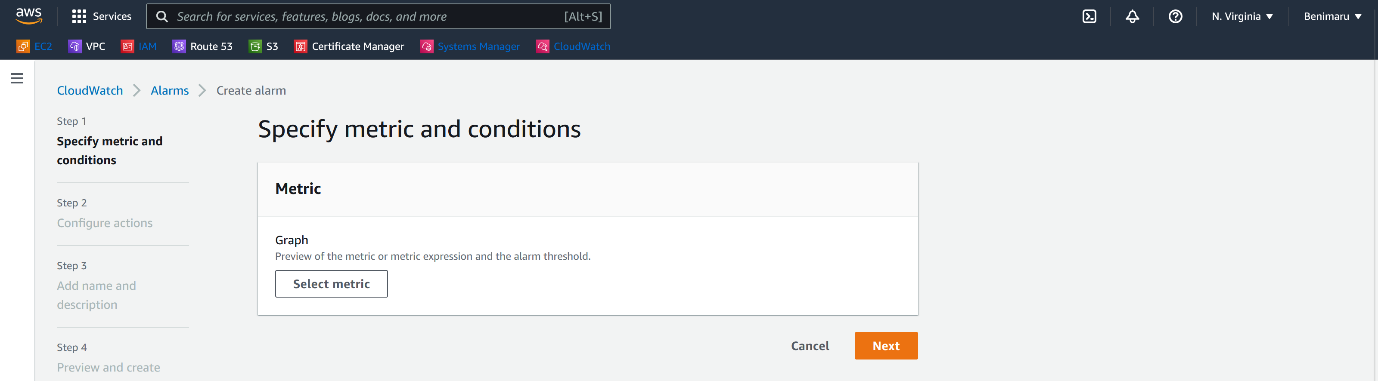


--- **note** – now we can create alarms.

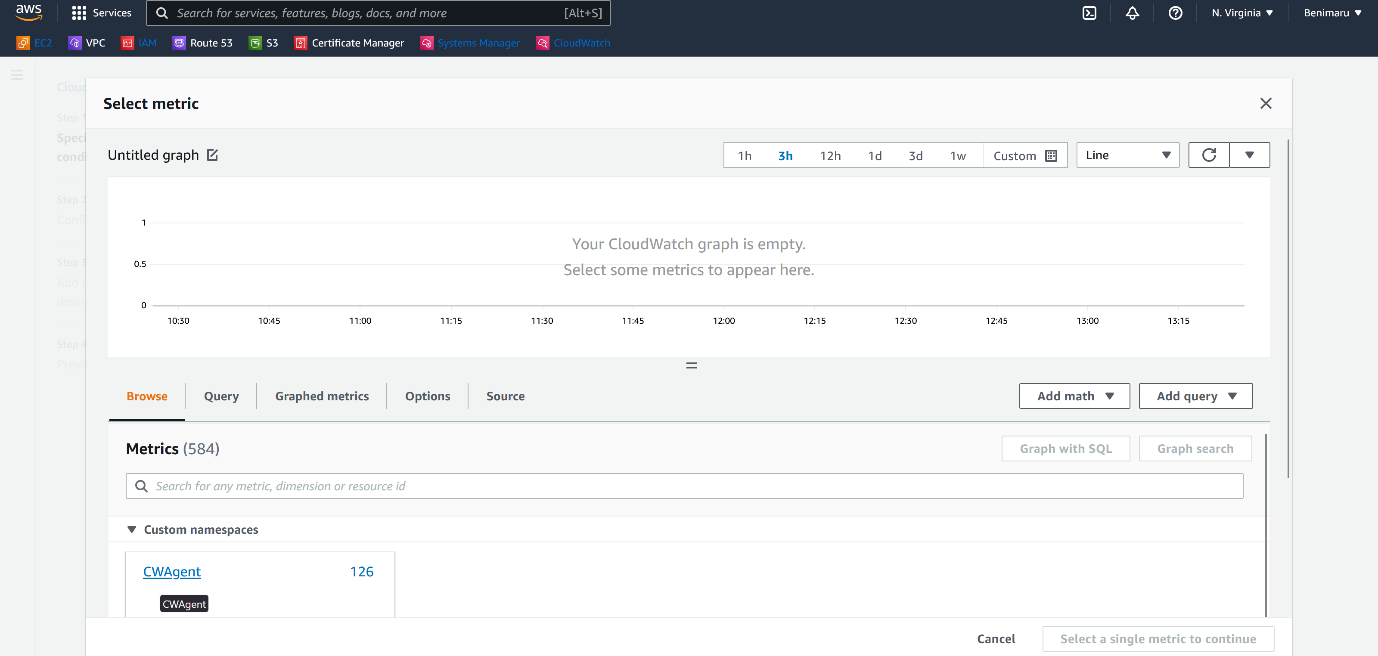
**Create alarms for memory utilization**



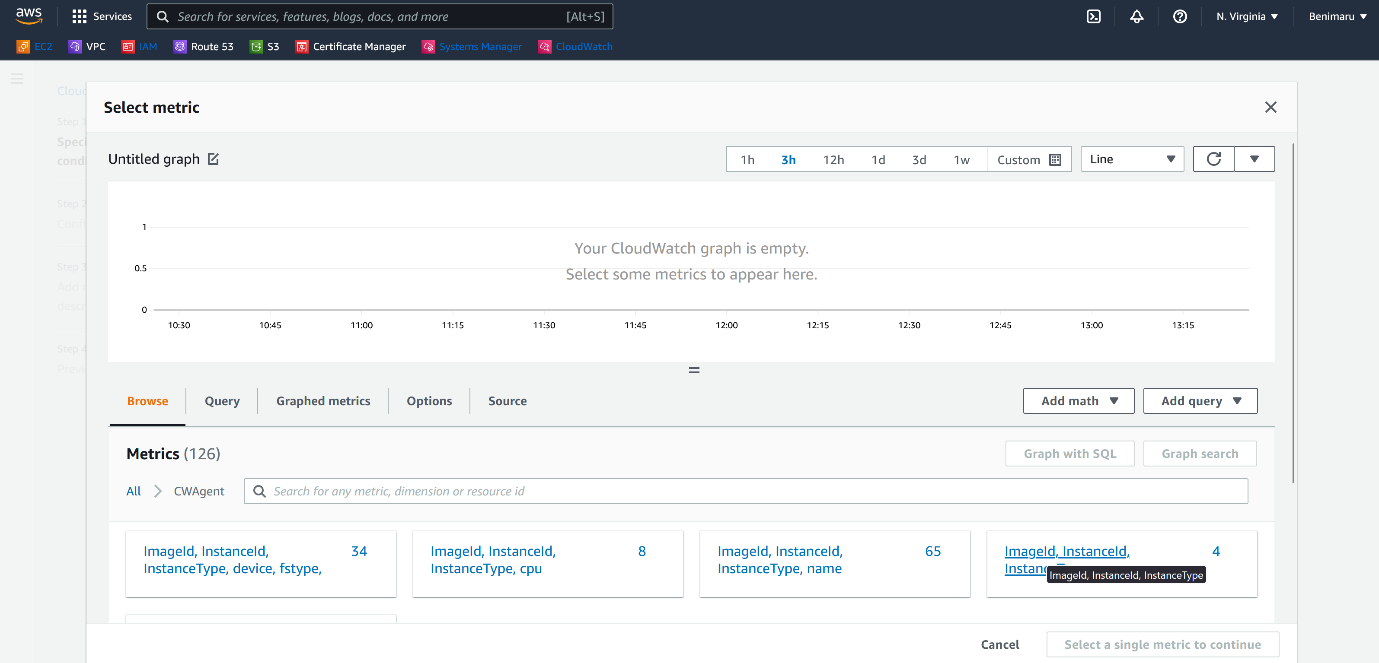
--- click on create alarm



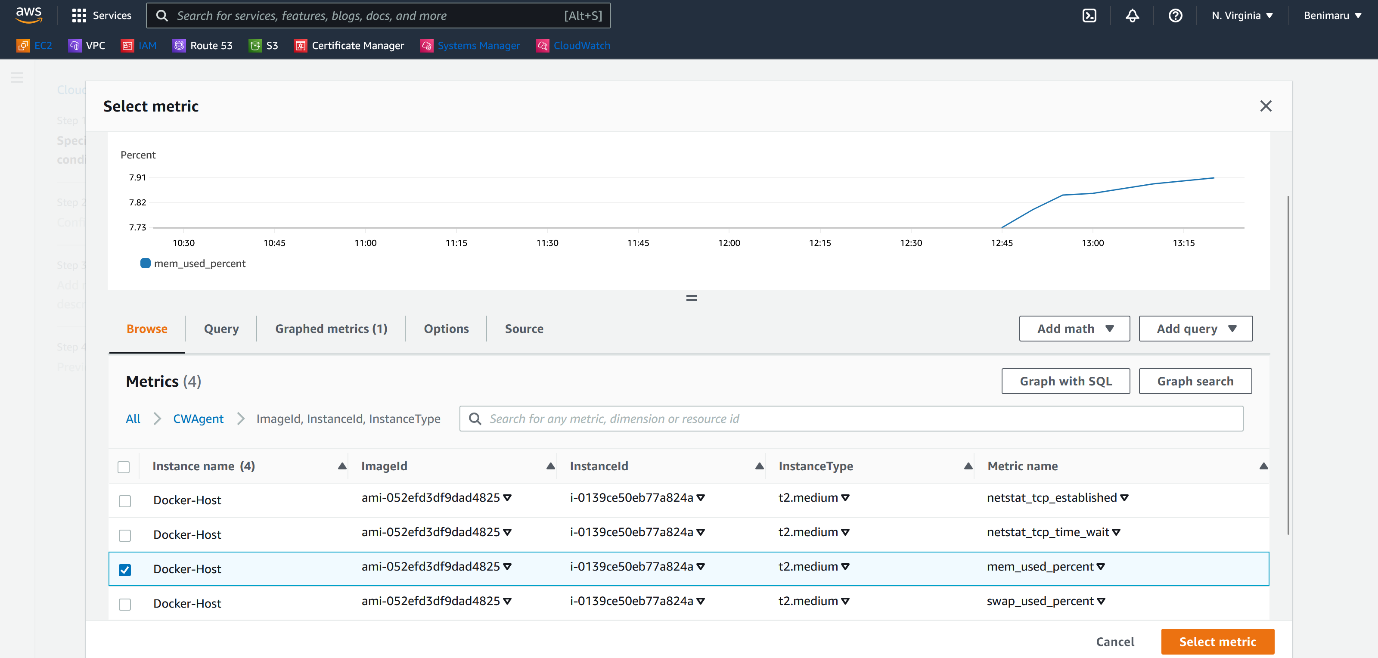
--- click on select metric.



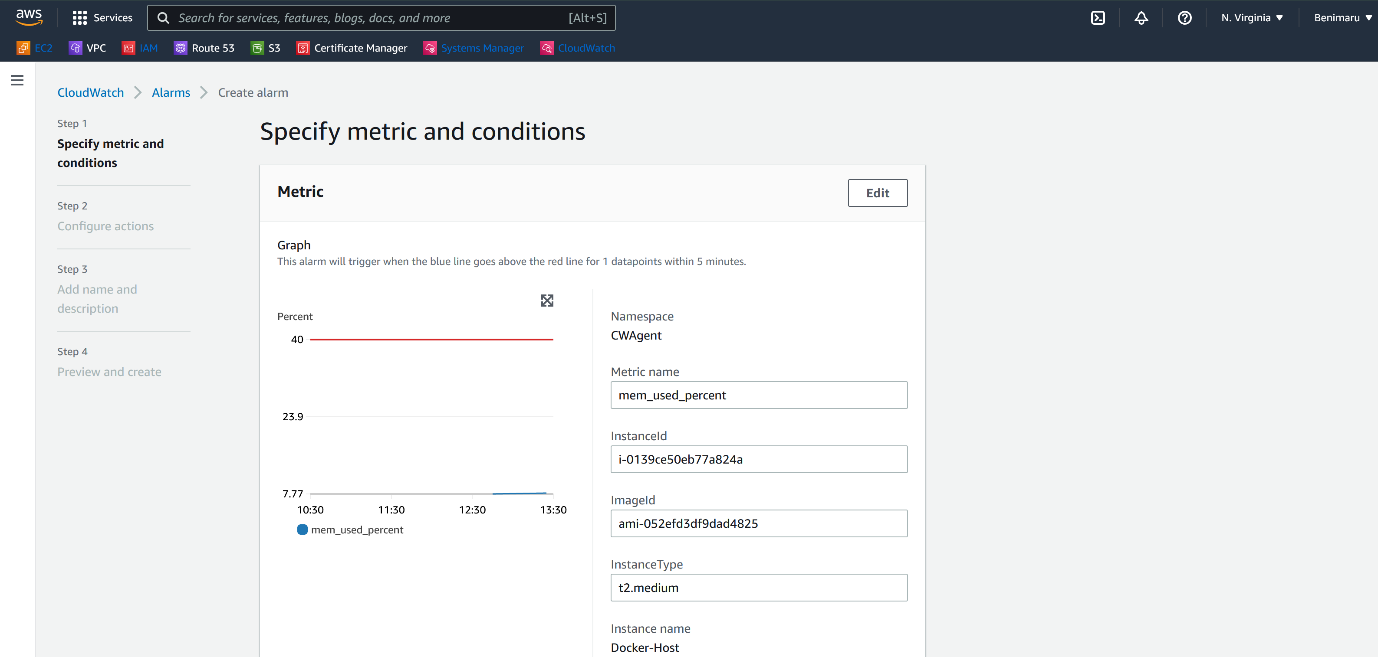
--- click on CWAgent

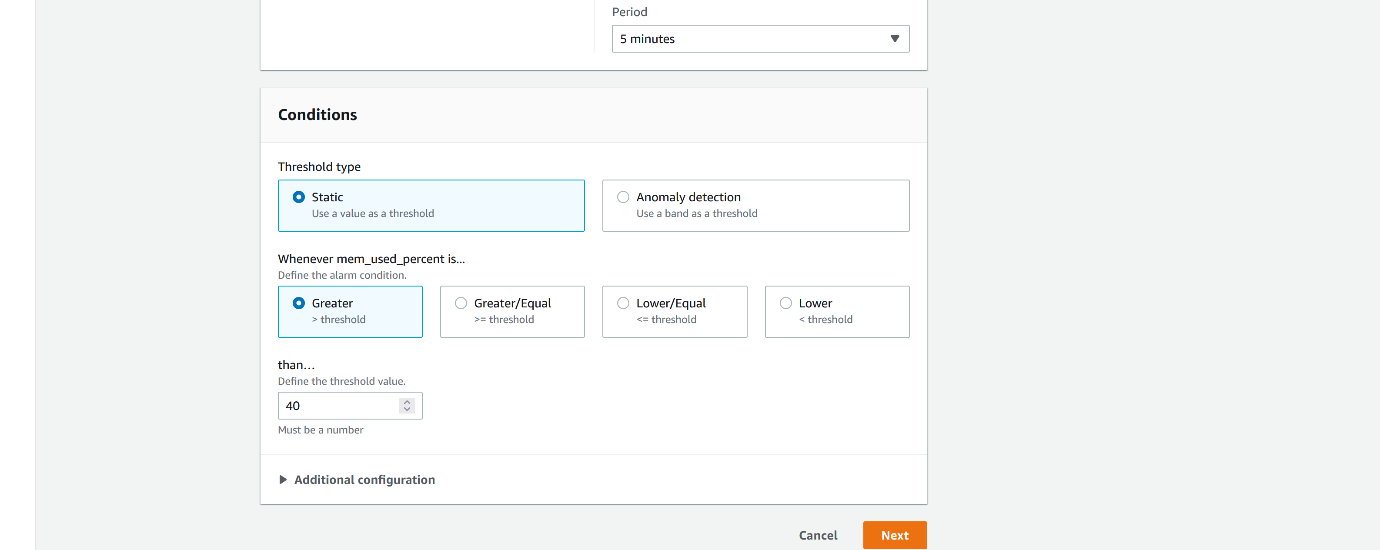


--- click on (ImagId, instanceid) this.

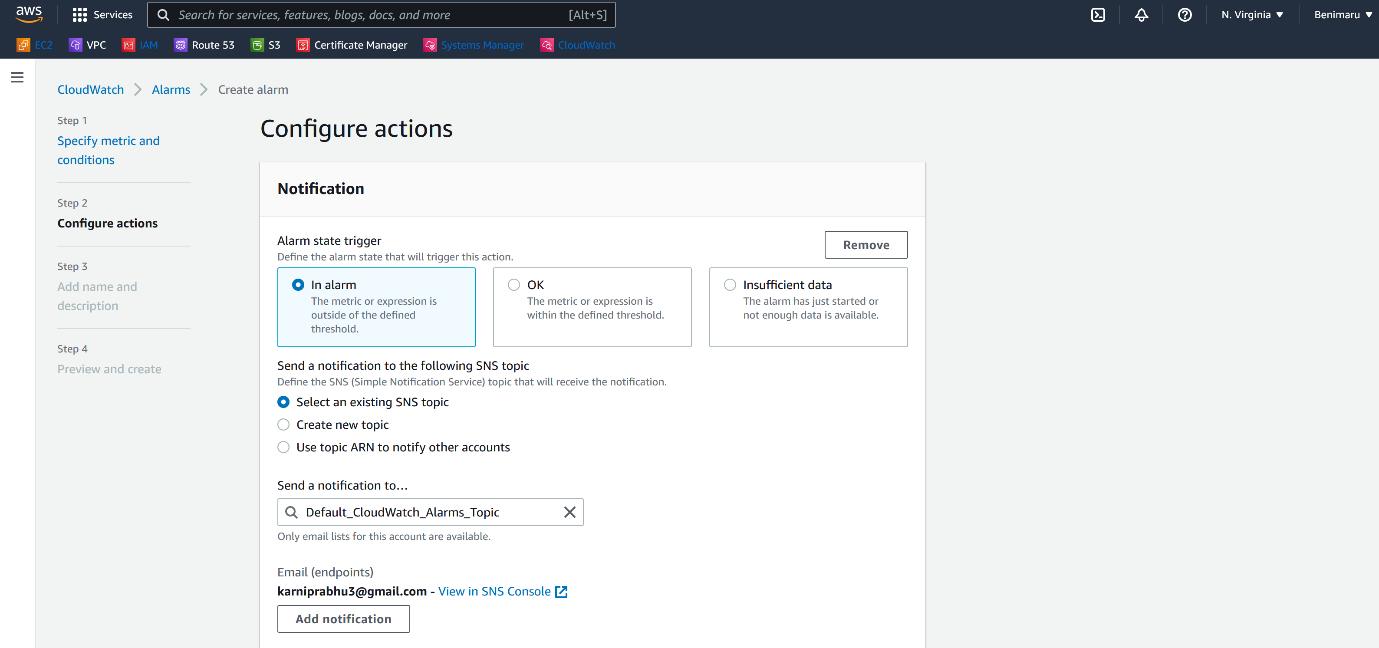


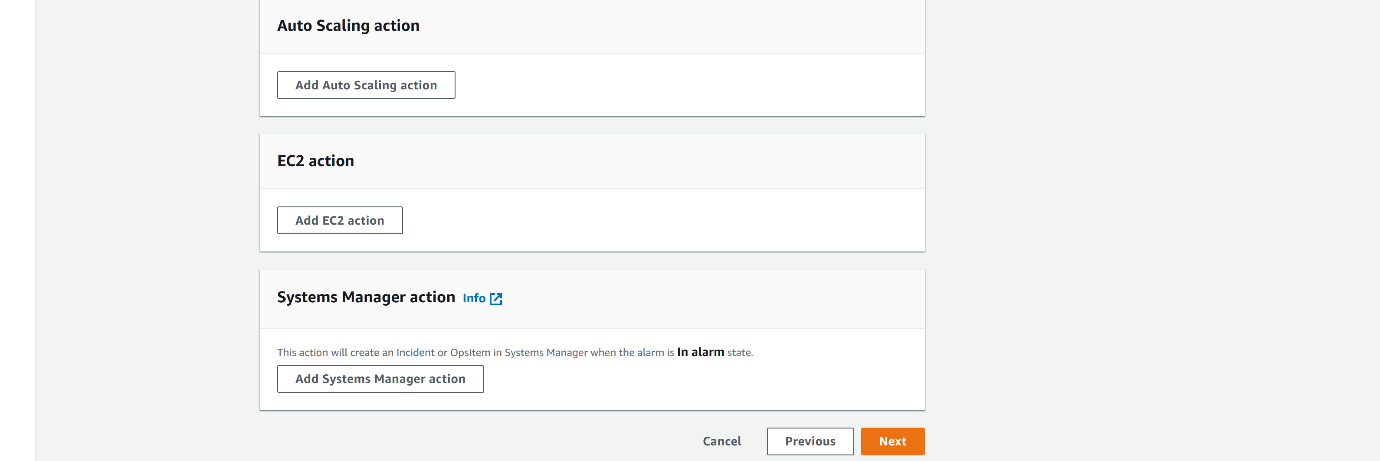
--- click select metric.



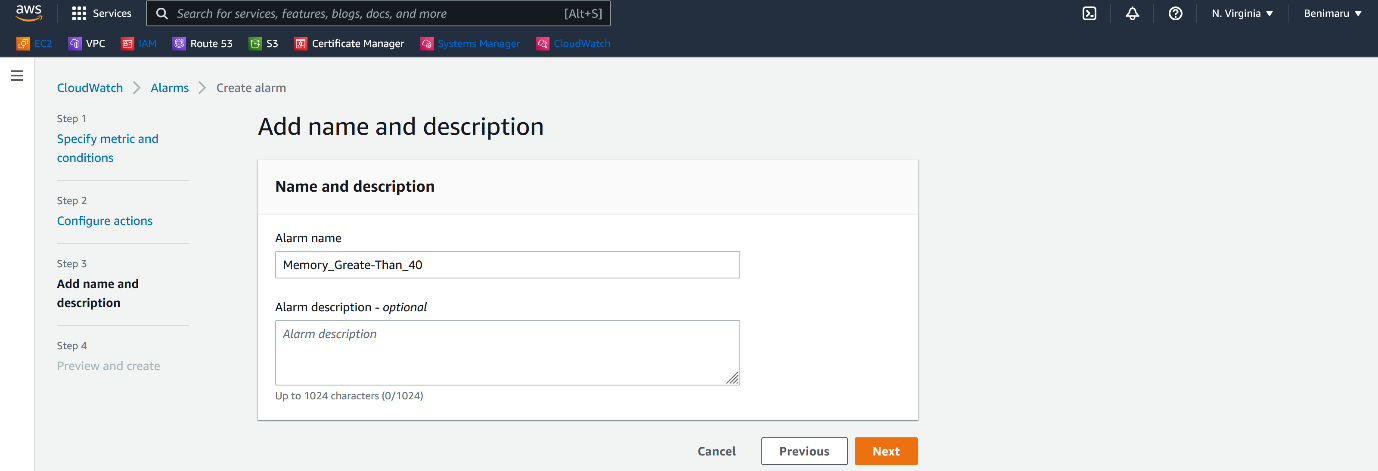


--- click on Next.



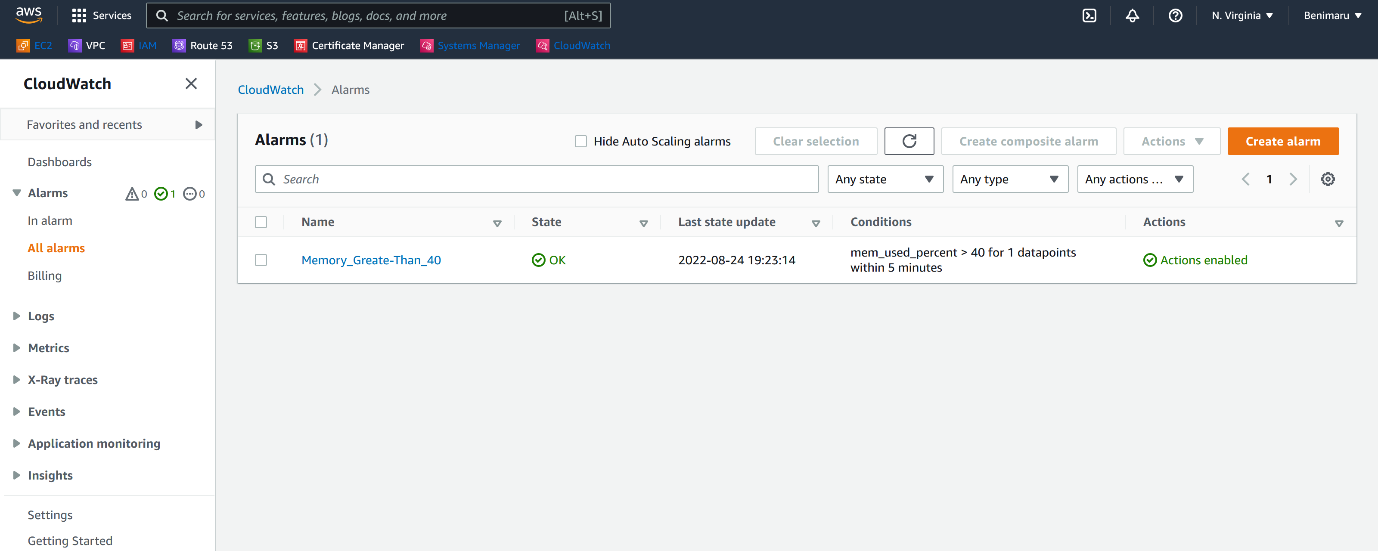


--- click on next.



--- give Alarm name here and click next.

--- **note** – click on create alarm.



--- **note** – the alarm got created successfully.

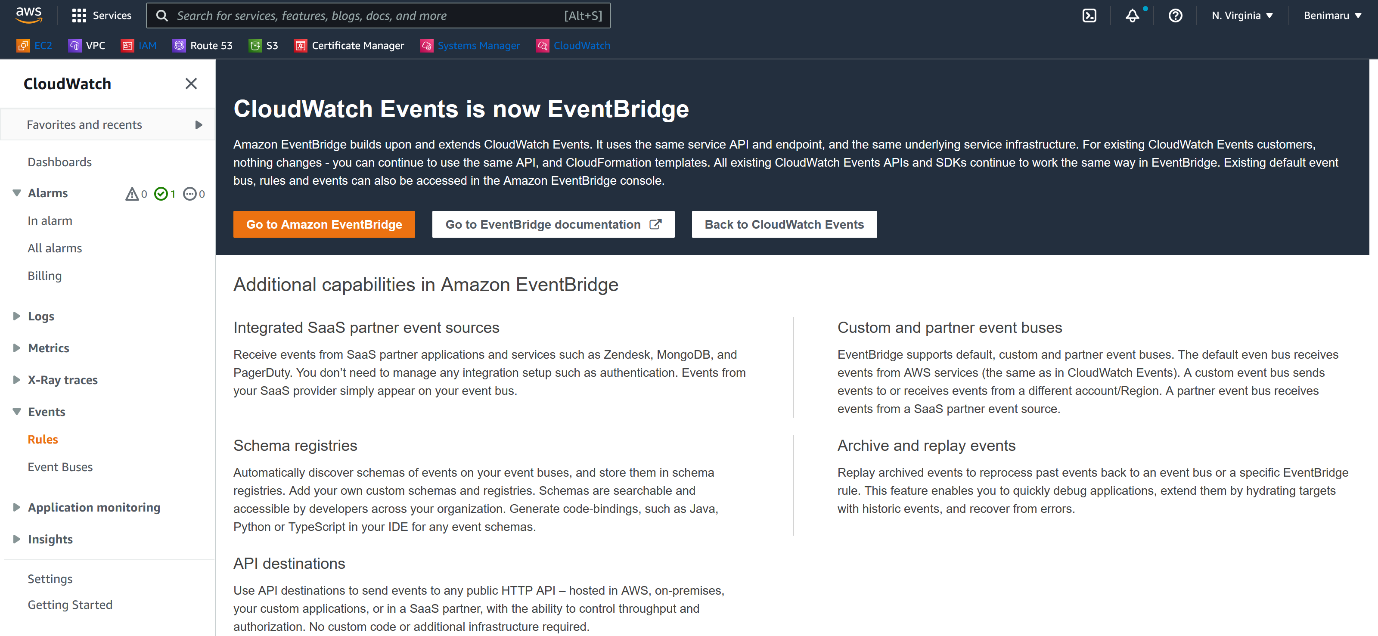
--- **important** - same way, we can monitor disk use percentage.

**Events**

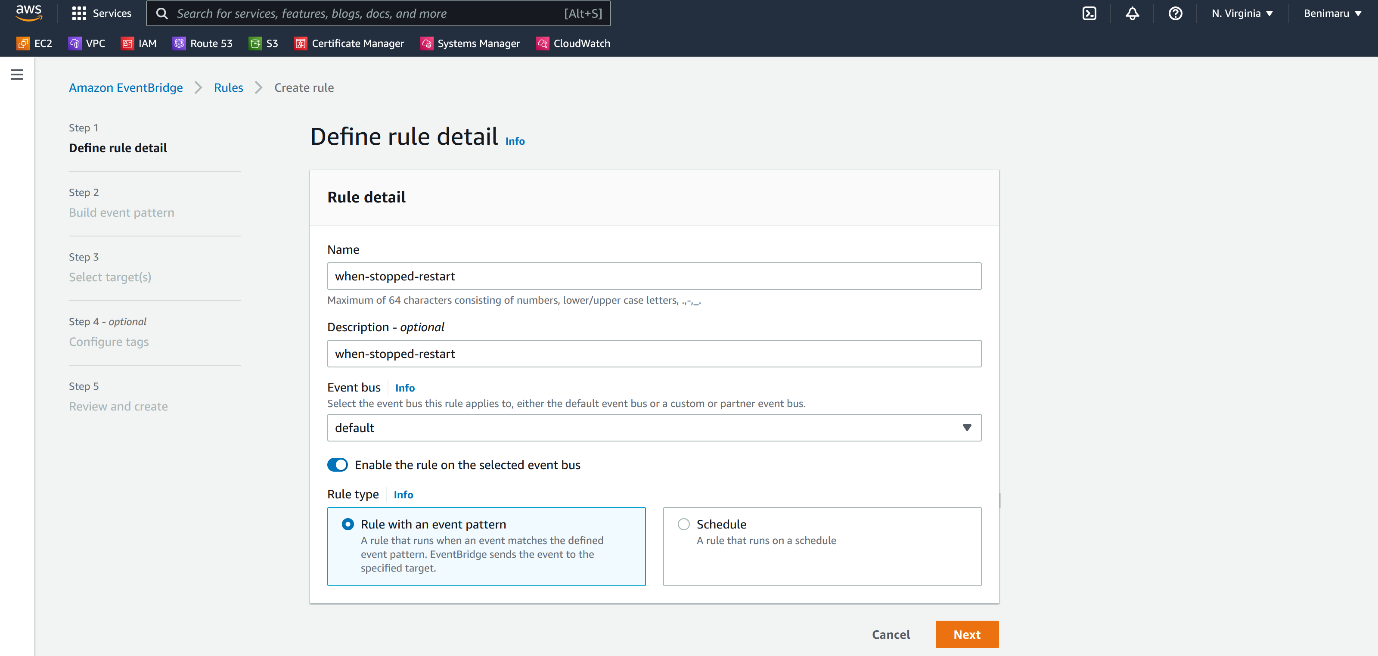
--- **scenario** – we have an instance running on aws and this is very important server, if any one stopped that instance then an event should be triggered and that instance needs to be started again.

--- go to cloudwatch

**Create a rule**



--- click on Go to Amazon EventBridge.



--- click on next.