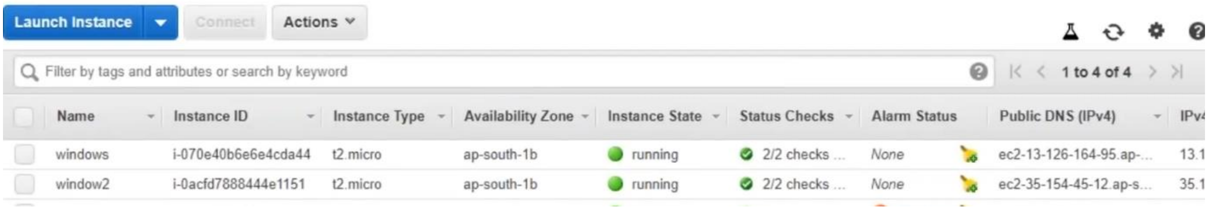
CloudWatch

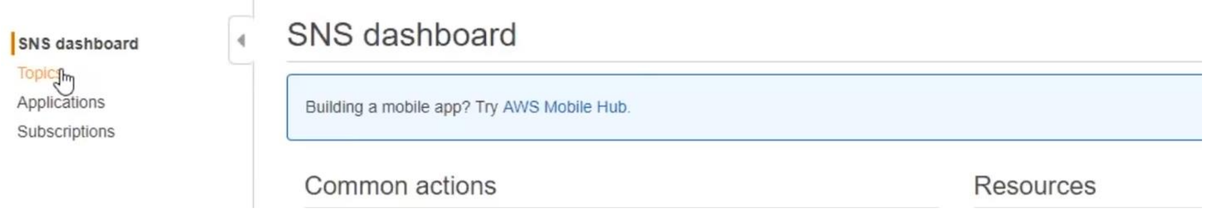
Configure CPU utilization

Create instance



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4
windows	i-070e40b6e6e4cda44	t2.micro	ap-south-1b	● running	✓ 2/2 checks ...	None	ec2-13-126-164-95.ap-...	13.1
window2	i-0acfd7888444e1151	t2.micro	ap-south-1b	● running	✓ 2/2 checks ...	None	ec2-35-154-45-12.ap-s...	35.1

Open SNS



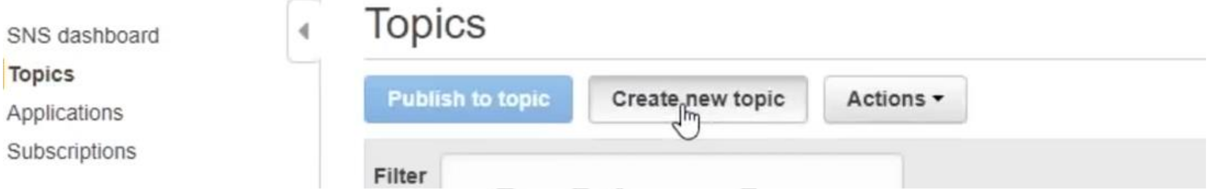
SNS dashboard

Building a mobile app? Try AWS Mobile Hub.

Common actions

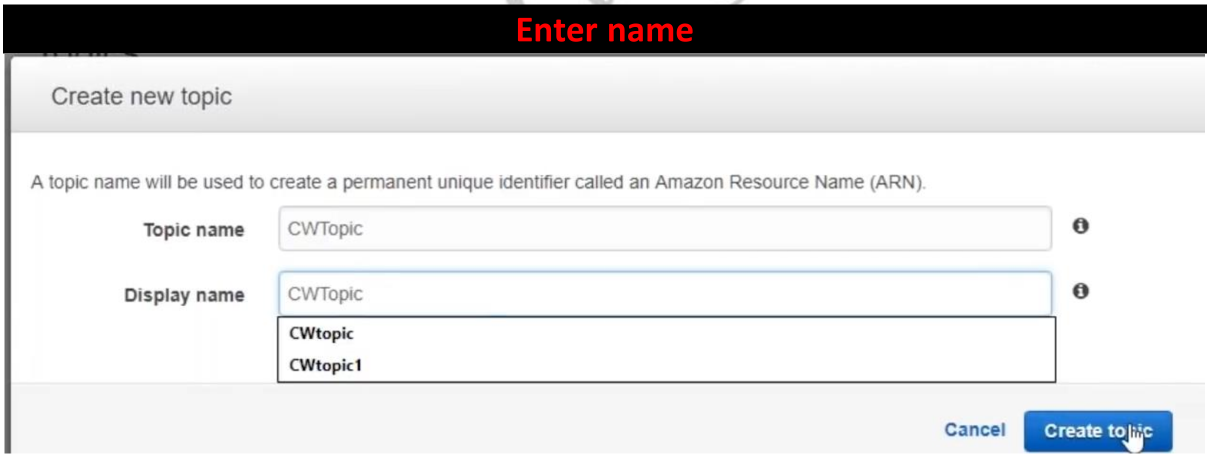
Resources

Create TOPIC



Topics

Enter name


Create new topic

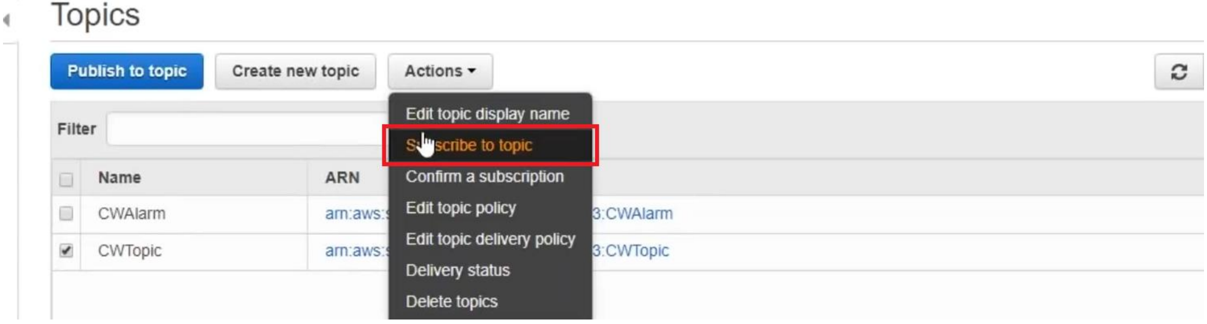
A topic name will be used to create a permanent unique identifier called an Amazon Resource Name (ARN).

Topic name: CWTopic

Display name: CWTopic

CWtopic
CWtopic1

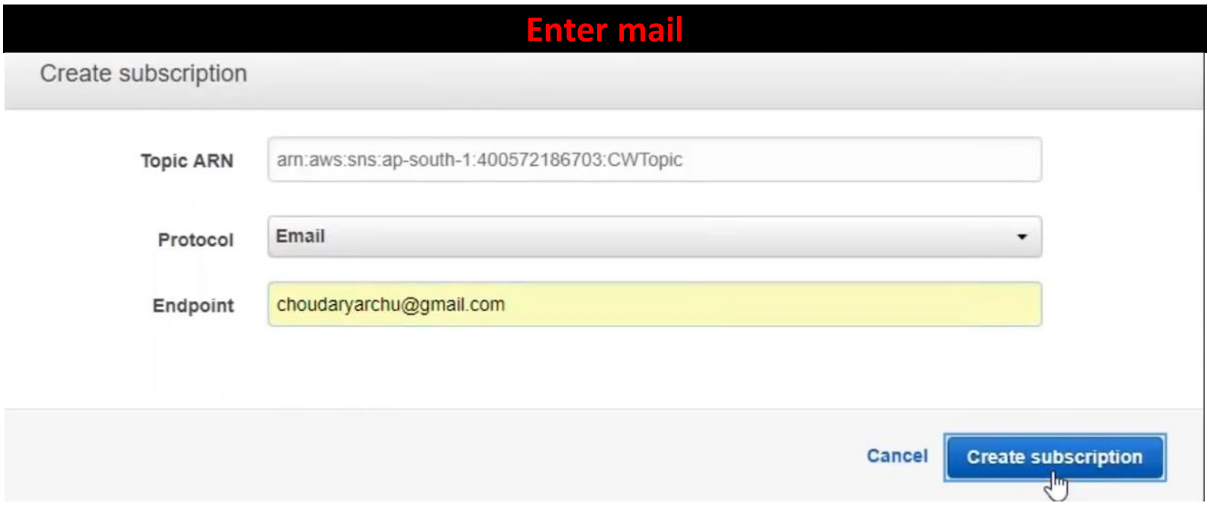
Create topic and subscribe to topic select


Topics

	Name	ARN
□	CWAlarm	arn:aws: ... 3:CWAlarm
☑	CWTopic	arn:aws: ... 3:CWTopic

CloudWatch

Enter mail



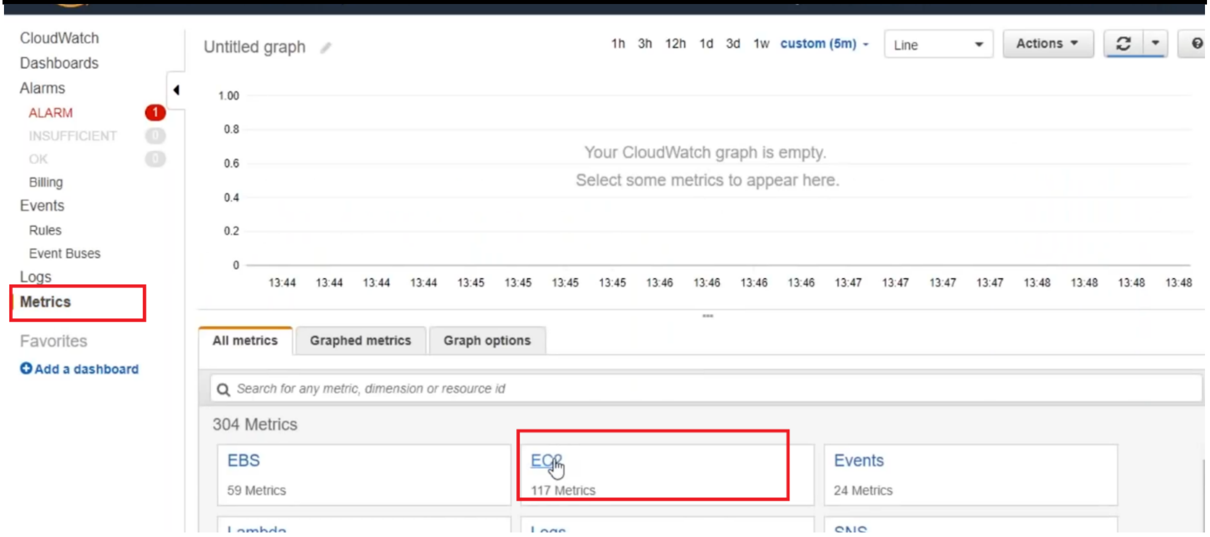
Create subscription

Topic ARN: arn:aws:sns:ap-south-1:400572186703:CWTopic

Protocol: Email

Endpoint: choudaryarchu@gmail.com

Open CloudWatch and click metrics and ec2



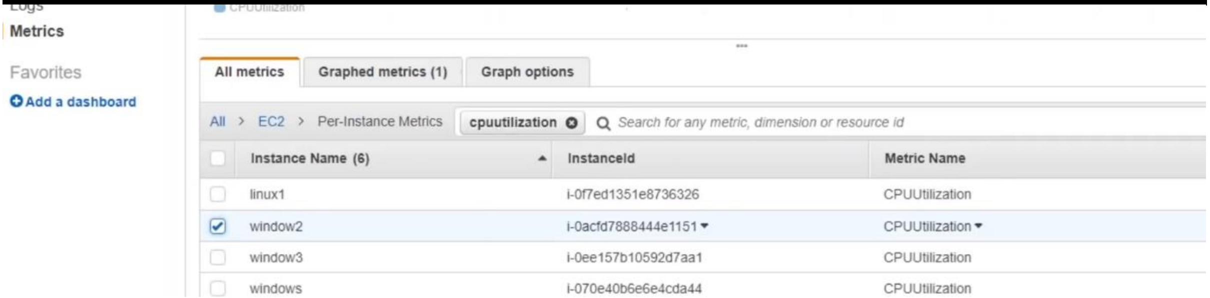
Your CloudWatch graph is empty.
Select some metrics to appear here.

Search for any metric, dimension or resource id

304 Metrics

EBS 59 Metrics	EC2 117 Metrics	Events 24 Metrics
Lambda	Logs	SNS

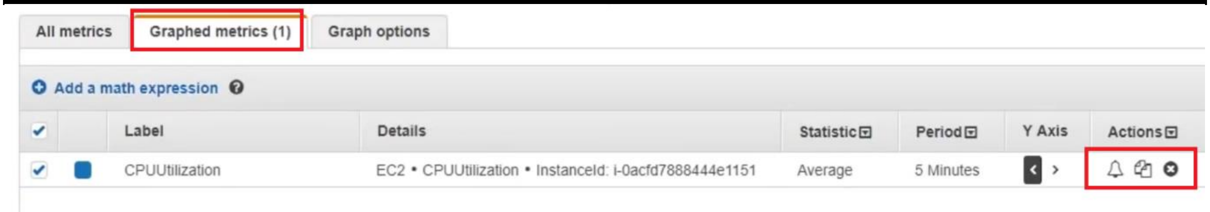
Choose instance



All > EC2 > Per-Instance Metrics   cpuutilization   Search for any metric, dimension or resource id

	Instance Name (6)	InstanceId	Metric Name
□	linux1	i-0f7ed1351e8736326	CPUUtilization
☑	window2	i-0acfd7888444e1151	CPUUtilization
□	window3	i-0ee157b10592d7aa1	CPUUtilization
□	windows	i-070e40b6e6e4cda44	CPUUtilization

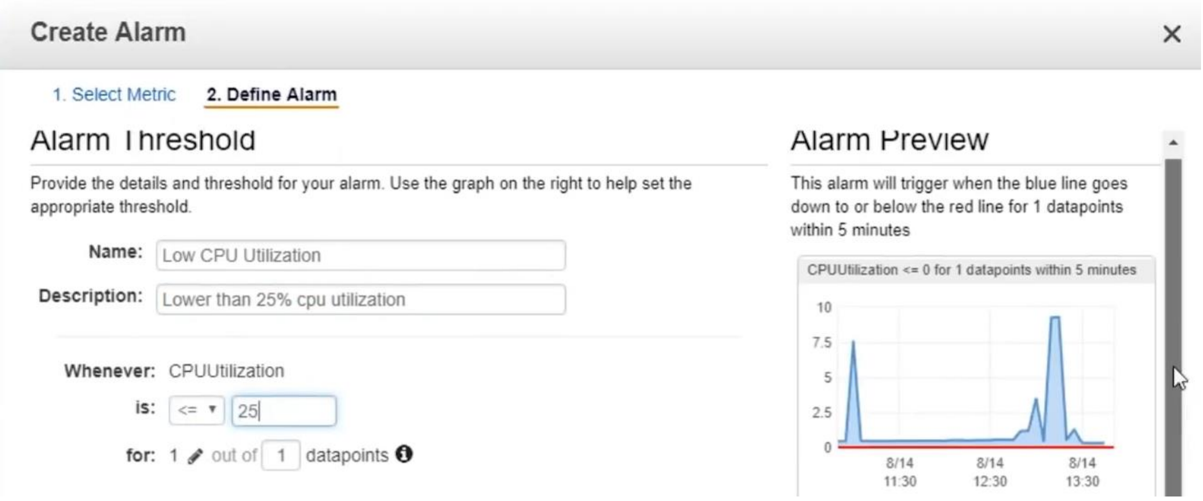
Select graphed and click notification symbol



✓		Label	Details	Statistic	Period	Y Axis	Actions
✓	🟦	CPUUtilization	EC2 • CPUUtilization • InstanceId: i-0acfd7888444e1151	Average	5 Minutes	‹ ›	🔔 ⧉ ⊗

CloudWatch

Create alarms



Create Alarm

1. Select Metric 2. Define Alarm

Alarm Threshold

Provide the details and threshold for your alarm. Use the graph on the right to help set the appropriate threshold.

Name: Low CPU Utilization

Description: Lower than 25% cpu utilization

Whenever: CPUUtilization

is: <= 25

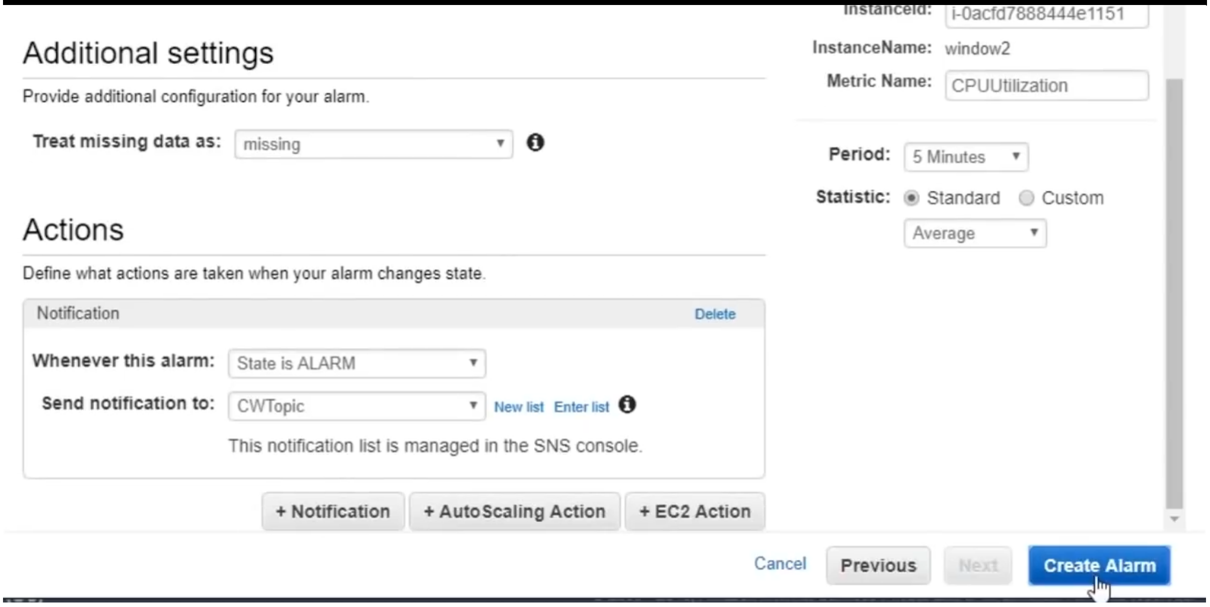
for: 1 out of 1 datapoints

Alarm Preview

This alarm will trigger when the blue line goes down to or below the red line for 1 datapoints within 5 minutes

CPUUtilization <= 0 for 1 datapoints within 5 minutes

Select SNS



Additional settings

Provide additional configuration for your alarm.

Treat missing data as: missing

Actions

Define what actions are taken when your alarm changes state.

Notification

Whenever this alarm: State is ALARM

Send notification to: CWTopic

This notification list is managed in the SNS console.

InstanceId: i-0acfd7888444e1151

InstanceName: window2

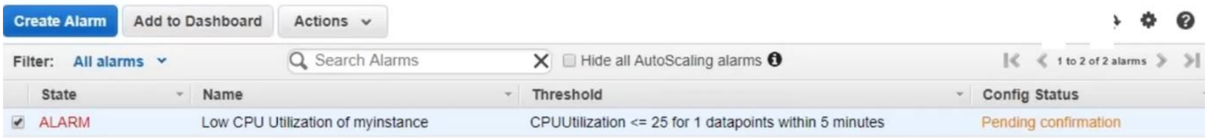
Metric Name: CPUUtilization

Period: 5 Minutes

Statistic: ● Standard ○ Custom

Average

CREATE SUCCESSFULLY

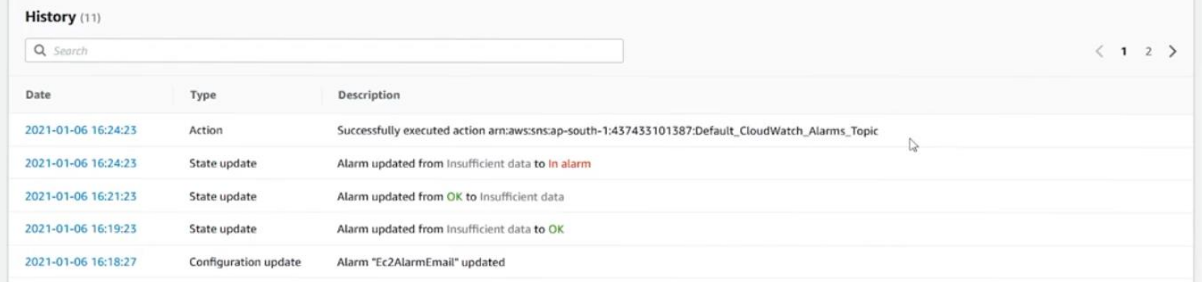


□ Hide all AutoScaling alarms

1 to 2 of 2 alarms

	State	Name	Threshold	Config Status
☑	ALARM	Low CPU Utilization of myinstance	CPUUtilization <= 25 for 1 datapoints within 5 minutes	Pending confirmation

If CPU utilization high than notification given



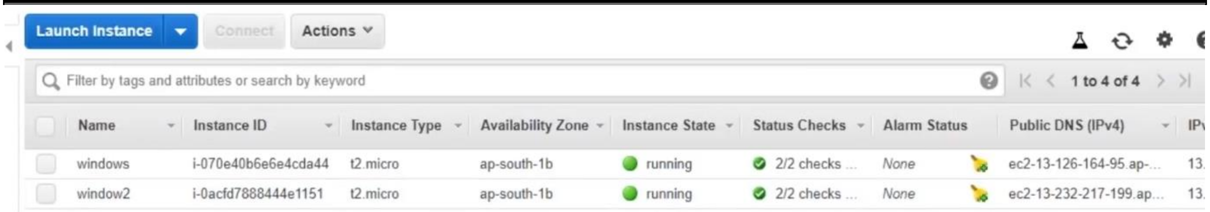
History (11)

Date	Type	Description
2021-01-06 16:24:23	Action	Successfully executed action arn:aws:sns:ap-south-1:437433101387:Default_CloudWatch_Alarms_Topic
2021-01-06 16:24:23	State update	Alarm updated from Insufficient data to In alarm
2021-01-06 16:21:23	State update	Alarm updated from OK to Insufficient data
2021-01-06 16:19:23	State update	Alarm updated from Insufficient data to OK
2021-01-06 16:18:27	Configuration update	Alarm "Ec2AlarmEmail" updated

CloudWatch

Configuration Schedule Events automatic shutdown (18 : 30) EC2 and Automatic Start-up (6:01 AM) EC2 instances

Create EC2

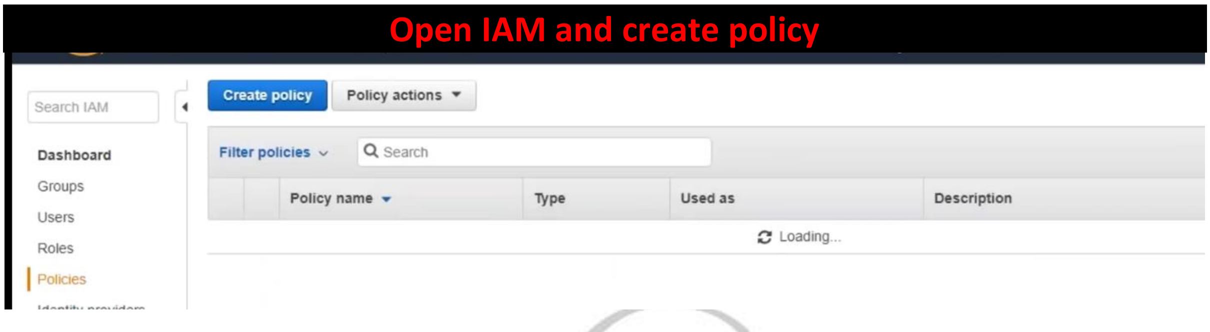


Launch Instance | Connect | Actions

Filter by tags and attributes or search by keyword — 1 to 4 of 4

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv
windows	i-070e40b6e6e4cda44	t2.micro	ap-south-1b	● running	✓ 2/2 checks ...	None	ec2-13-126-164-95.ap-...	13.
window2	i-0acfd7888444e1151	t2.micro	ap-south-1b	● running	✓ 2/2 checks ...	None	ec2-13-232-217-199.ap...	13.

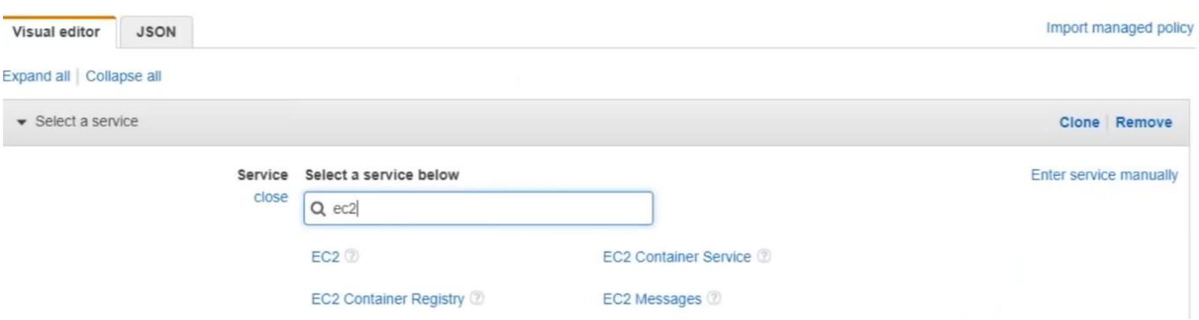
Open IAM and create policy



Policy name	Type	Used as	Description

Loading...

Select EC2



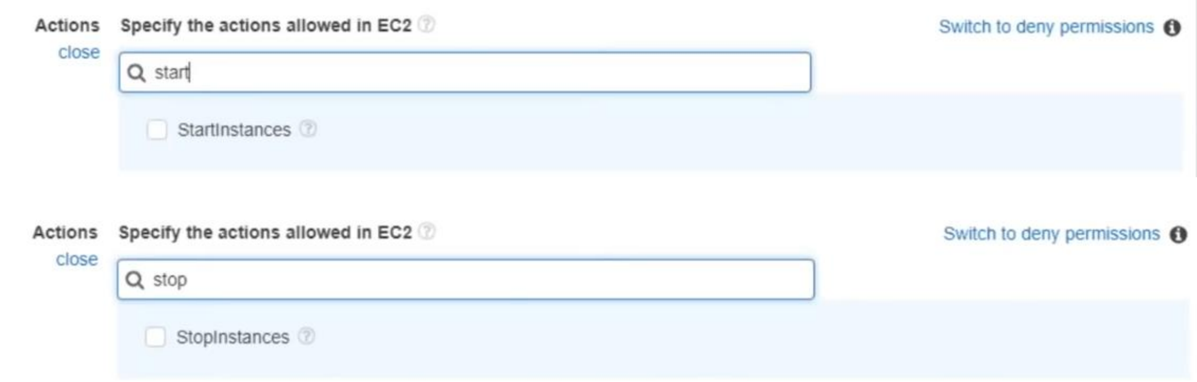
Service close — Select a service below

ec2

EC2 | EC2 Container Service

EC2 Container Registry | EC2 Messages

ACTION SELECT



Actions close — Specify the actions allowed in EC2 — Switch to deny permissions

start

□ StartInstances

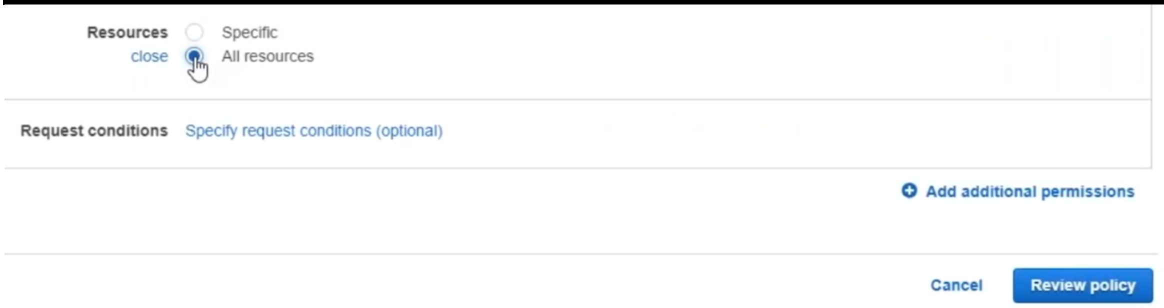
Actions close — Specify the actions allowed in EC2 — Switch to deny permissions

stop

□ StopInstances

CloudWatch

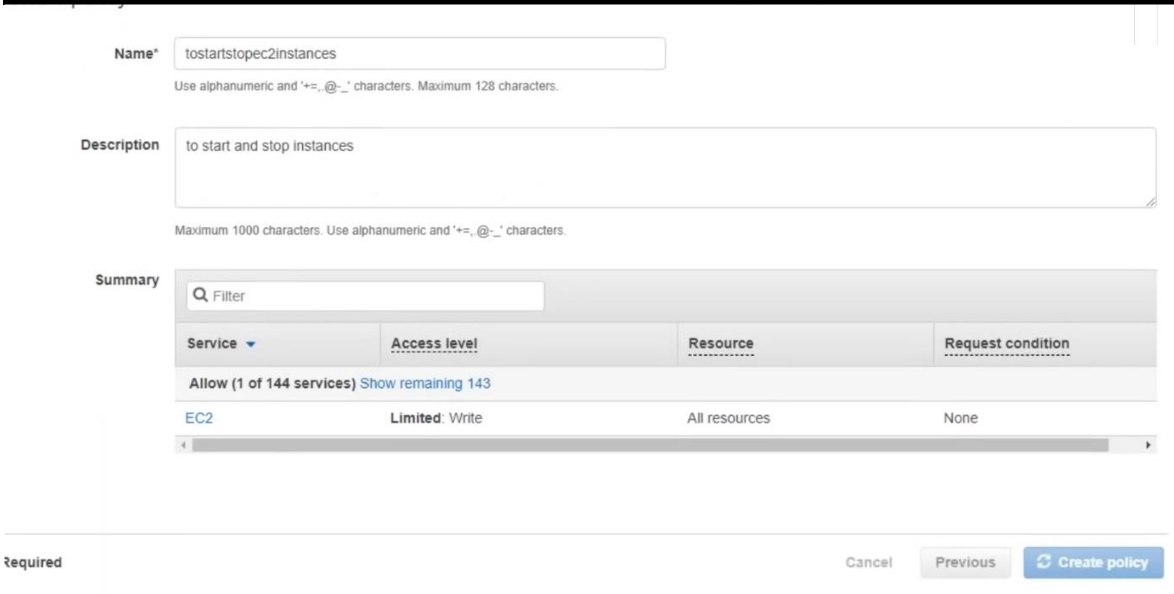
SELECT RESOURCES



Resources ○ Specific
close ● All resources

Request conditions: Specify request conditions (optional)

Enter name and create policy



Name*: tostartstopec2instances

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

Description: to start and stop instances

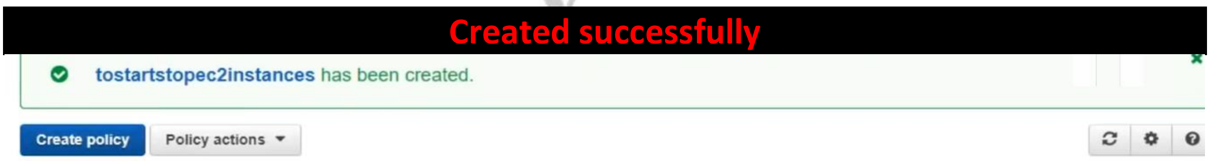
Maximum 1000 characters. Use alphanumeric and '+=,.@-_' characters.

Summary

Allow (1 of 144 services) Show remaining 143

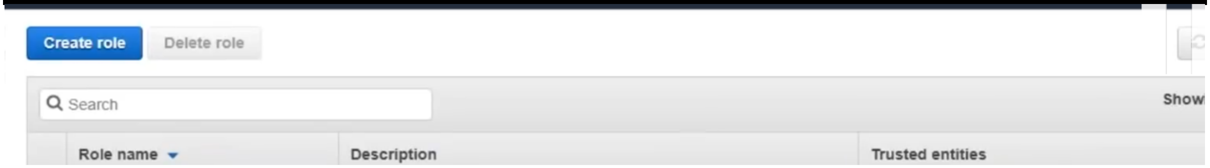
Service	Access level	Resource	Request condition
EC2	**Limited**: Write	All resources	None

Created successfully

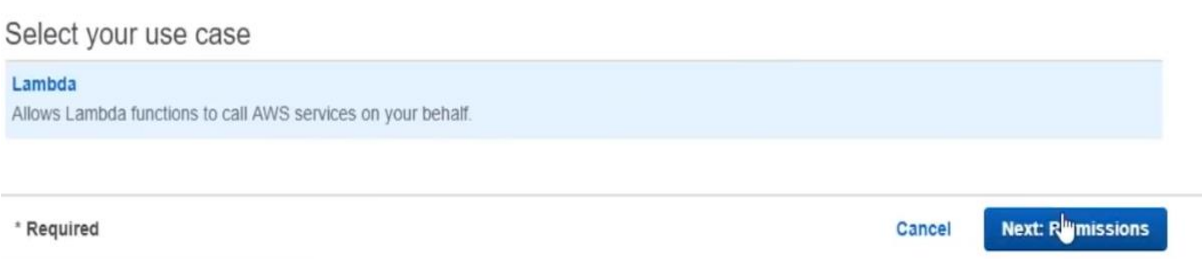


tostartstopec2instances has been created.

Create roles



Choose lambda function

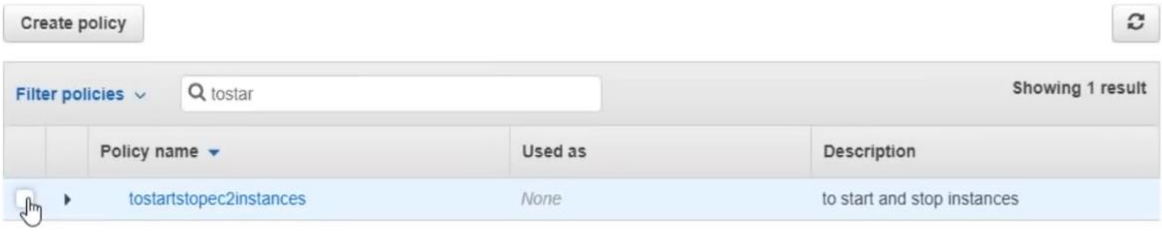


Select your use case

Lambda
Allows Lambda functions to call AWS services on your behalf.

CloudWatch

Select policys

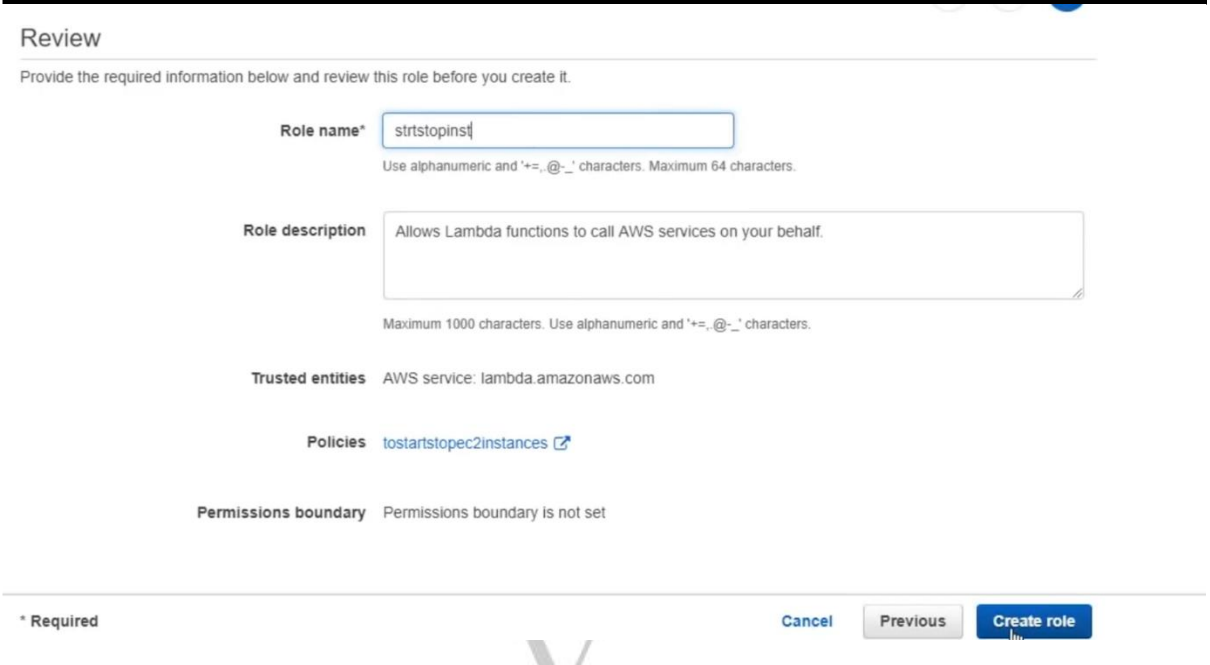


tostar

Showing 1 result

Policy name	Used as	Description
tostartstopec2instances	None	to start and stop instances

Review and create



Review

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

Role name* strtstopinst

Use alphanumeric and '+=,.@-_' characters. Maximum 64 characters.

Role description Allows Lambda functions to call AWS services on your behalf.

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

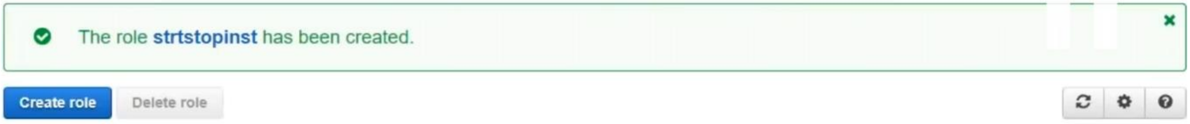
Trusted entities AWS service: lambda.amazonaws.com

Policies tostartstopec2instances

Permissions boundary Permissions boundary is not set

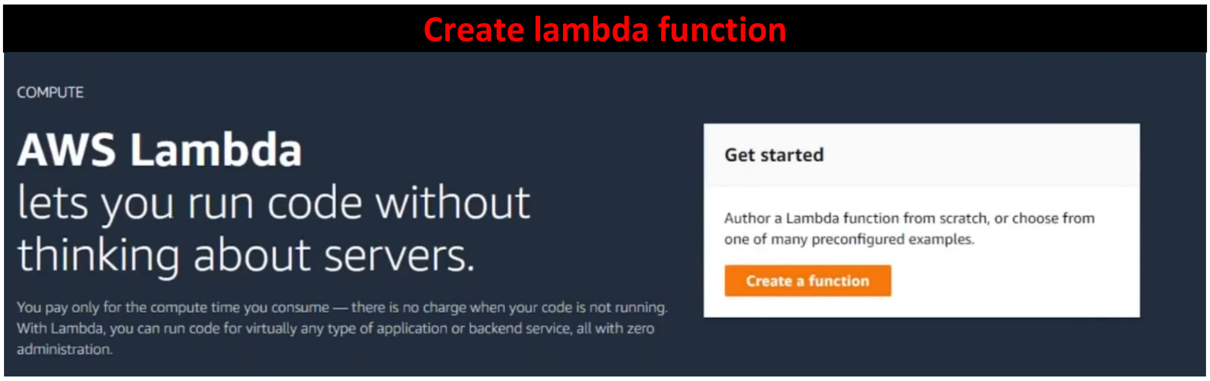
* Required

See successfully create role



The role strtstopinst has been created.

Create lambda function



COMPUTE

AWS Lambda

lets you run code without thinking about servers.

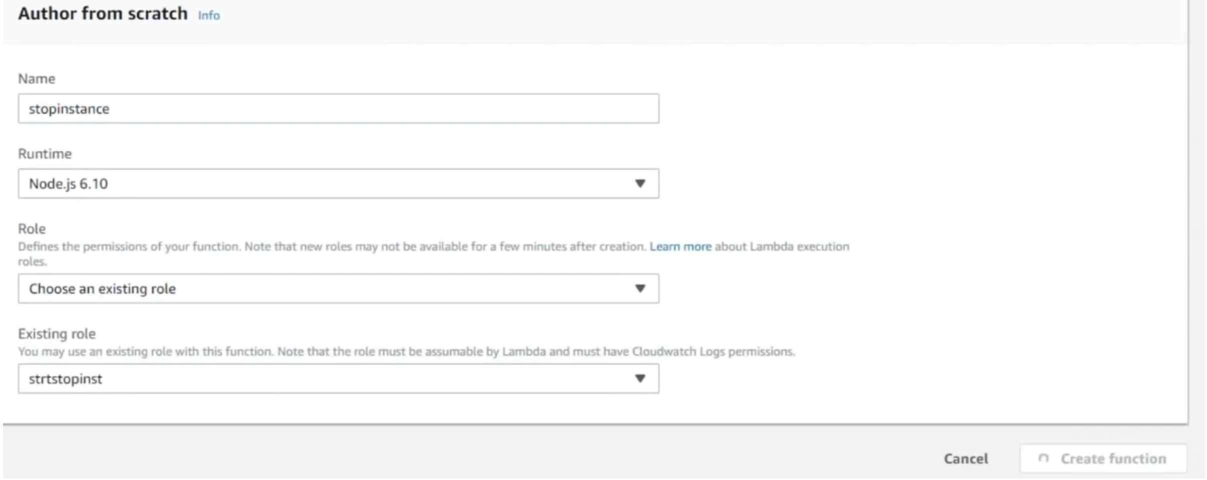
You pay only for the compute time you consume — there is no charge when your code is not running. With Lambda, you can run code for virtually any type of application or backend service, all with zero administration.

Get started

Author a Lambda function from scratch, or choose from one of many preconfigured examples.

CloudWatch

Enter name and select role stop



Author from scratch Info

Name: stopinstance

Runtime: Node.js 6.10

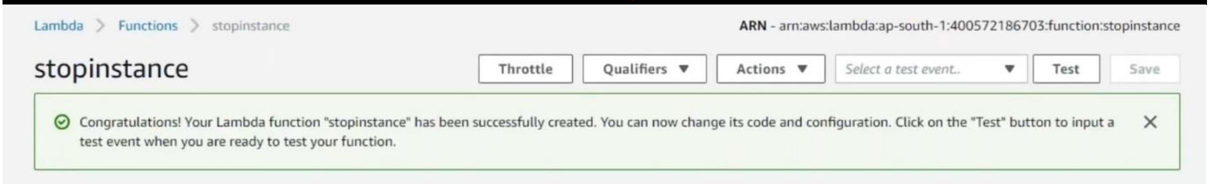
Role
Defines the permissions of your function. Note that new roles may not be available for a few minutes after creation. Learn more about Lambda execution roles.

Choose an existing role

Existing role
You may use an existing role with this function. Note that the role must be assumable by Lambda and must have Cloudwatch Logs permissions.

strtstopinst

Successfully create

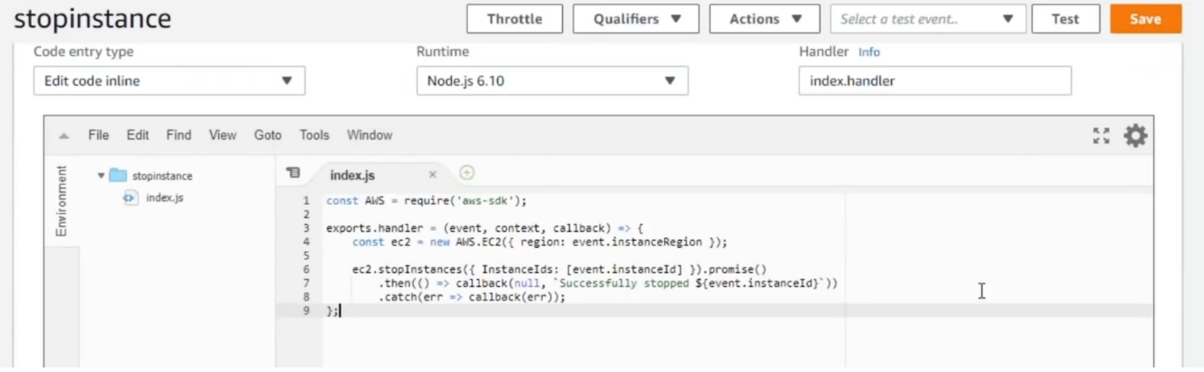


ARN - arn:aws:lambda:ap-south-1:400572186703:function:stopinstance

stopinstance

Congratulations! Your Lambda function "stopinstance" has been successfully created. You can now change its code and configuration. Click on the "Test" button to input a test event when you are ready to test your function.

Configure stop ec2 query and save



stopinstance

Code entry type: Edit code inline

Runtime: Node.js 6.10

Handler: index.handler

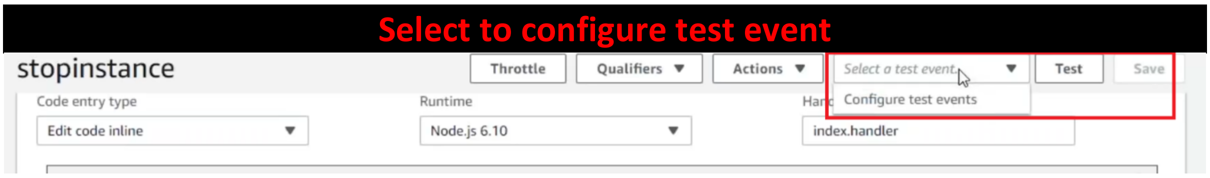
index.js

```
const AWS = require('aws-sdk');

exports.handler = (event, context, callback) => {
    const ec2 = new AWS.EC2({ region: event.instanceRegion });

    ec2.stopInstances({ InstanceIds: [event.instanceId] }).promise()
        .then(() => callback(null, `Successfully stopped ${event.instanceId}`))
        .catch(err => callback(err));
};
```

Select to configure test event



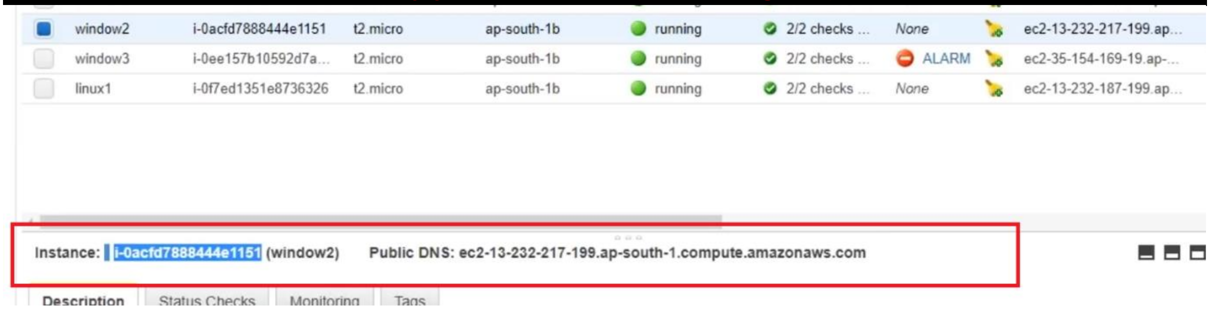
stopinstance

Code entry type: Edit code inline

Runtime: Node.js 6.10

Handler: index.handler

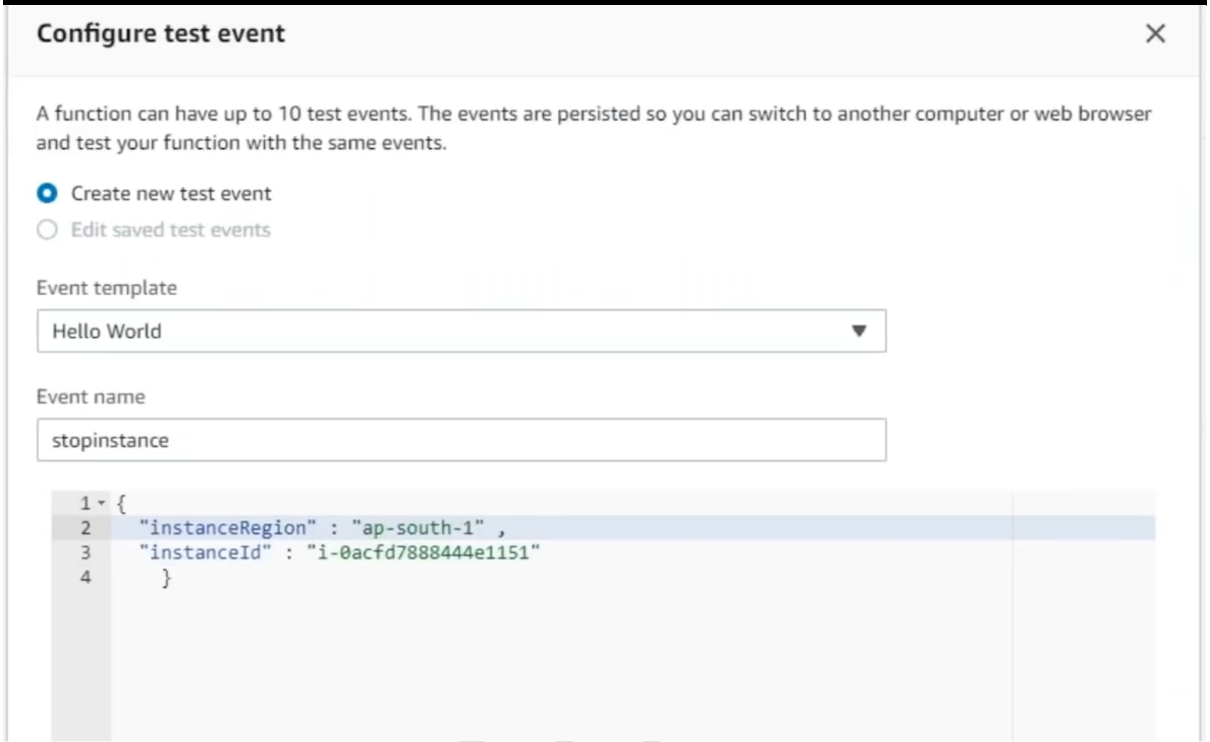
Copy windows 2 (ID & Region)



■	window2	i-0acfd7888444e1151	t2.micro	ap-south-1b	● running	✓ 2/2 checks ...	None		ec2-13-232-217-199.ap...
□	window3	i-0ee157b10592d7a...	t2.micro	ap-south-1b	● running	✓ 2/2 checks ...	ALARM		ec2-35-154-169-19.ap-...
□	linux1	i-0f7ed1351e8736326	t2.micro	ap-south-1b	● running	✓ 2/2 checks ...	None		ec2-13-232-187-199.ap...

Instance: i-0acfd7888444e1151 (window2) Public DNS: ec2-13-232-217-199.ap-south-1.compute.amazonaws.com

Configure test page



Configure test event

A function can have up to 10 test events. The events are persisted so you can switch to another computer or web browser and test your function with the same events.

● Create new test event

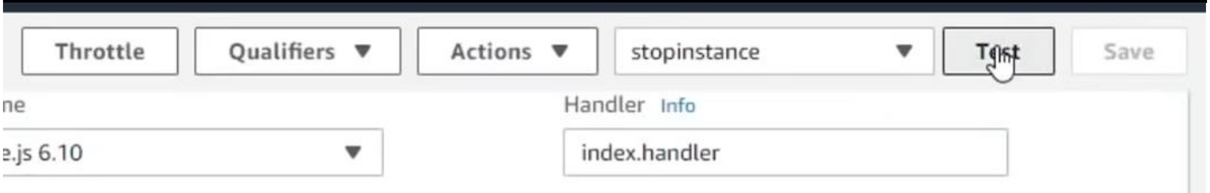
○ Edit saved test events

Event template: Hello World

Event name: stopinstance

```
{
  "instanceRegion" : "ap-south-1" ,
  "instanceId" : "i-0acfd7888444e1151"
    }
```

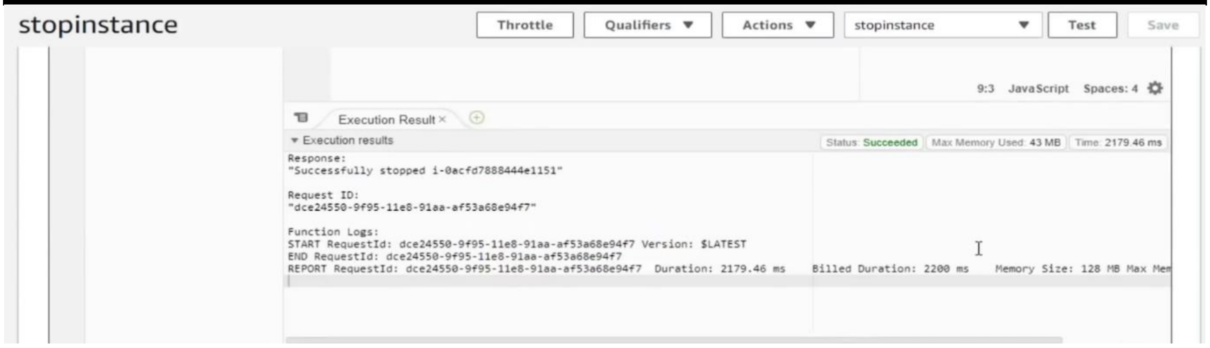
Click to test



ne: …js 6.10

Handler Info: index.handler

See event successfully run



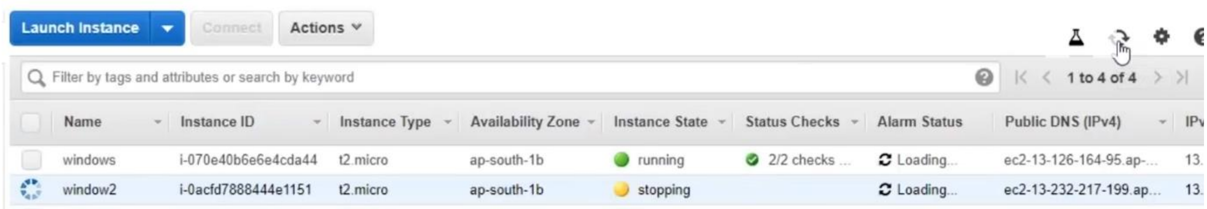
Status: Succeeded | Max Memory Used: 43 MB | Time: 2179.46 ms

```
Response:
"Successfully stopped i-0acfd7888444e1151"

Request ID:
"dce24550-9f95-11e8-91aa-af53a68e94f7"

Function Logs:
START RequestId: dce24550-9f95-11e8-91aa-af53a68e94f7 Version: $LATEST
END RequestId: dce24550-9f95-11e8-91aa-af53a68e94f7
REPORT RequestId: dce24550-9f95-11e8-91aa-af53a68e94f7  Duration: 2179.46 ms    Billed Duration: 2200 ms    Memory Size: 128 MB Max Mem
```

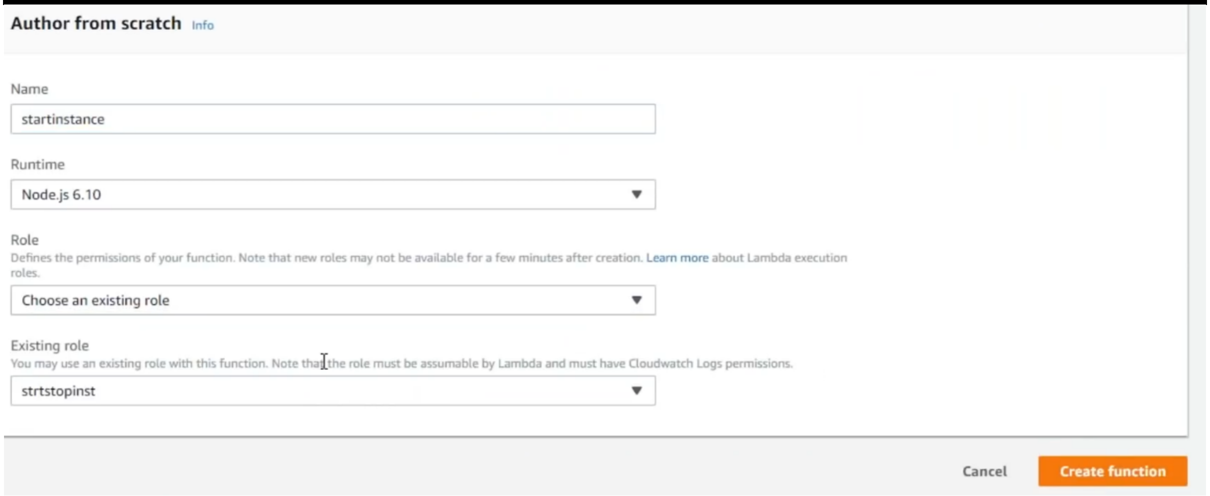
See window 2 ec2 instance stop



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv
windows	i-070e40b6e6e4cda44	t2.micro	ap-south-1b	● running	✓ 2/2 checks ...	Loading...	ec2-13-126-164-95.ap-...	13.
window2	i-0acfd7888444e1151	t2.micro	ap-south-1b	● stopping		Loading...	ec2-13-232-217-199.ap...	13.

CloudWatch

Enter name and select role start



Author from scratch Info

Name

startinstance

Runtime

Node.js 6.10

Role

Defines the permissions of your function. Note that new roles may not be available for a few minutes after creation. Learn more about Lambda execution roles.

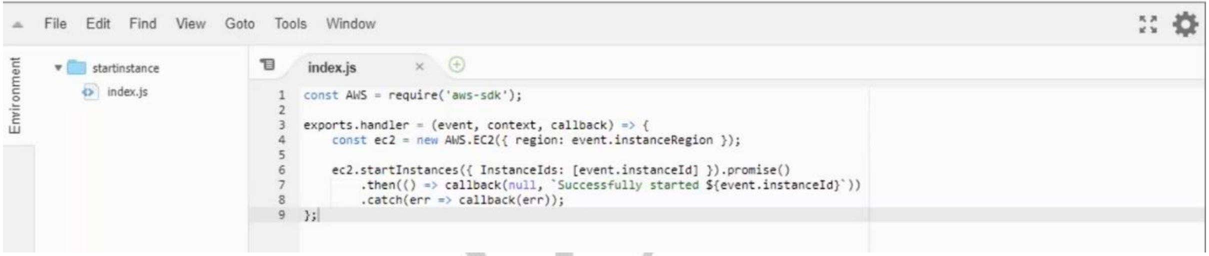
Choose an existing role

Existing role

You may use an existing role with this function. Note that the role must be assumable by Lambda and must have Cloudwatch Logs permissions.

strtstopinst

After created Enter query start ec2



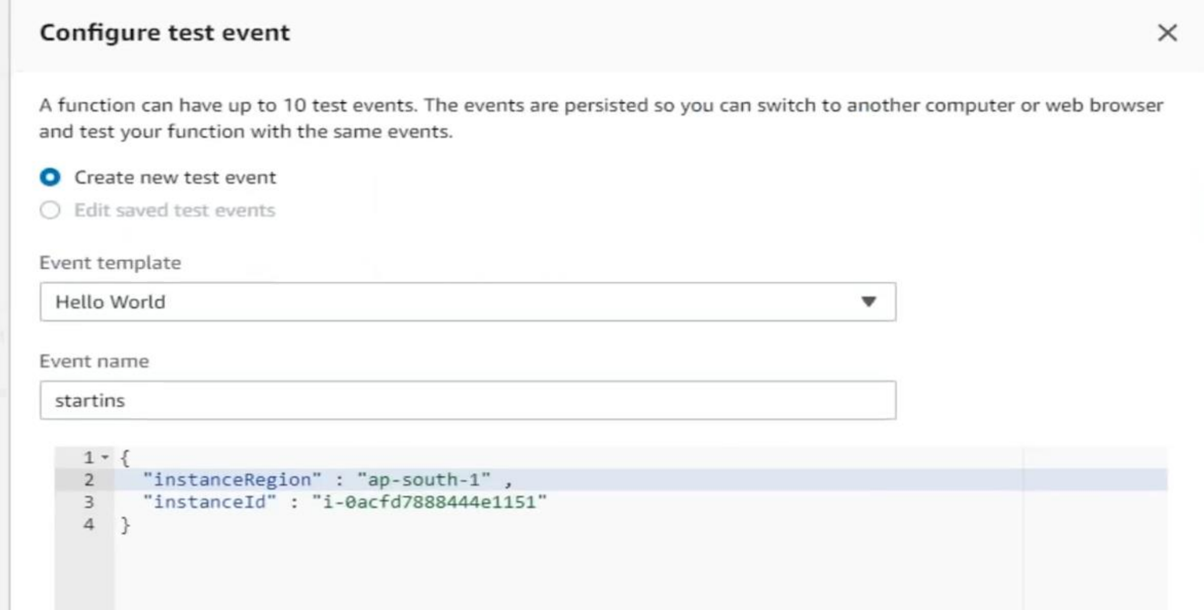
index.js

```
const AWS = require('aws-sdk');

exports.handler = (event, context, callback) => {
    const ec2 = new AWS.EC2({ region: event.instanceRegion });

    ec2.startInstances({ InstanceIds: [event.instanceId] }).promise()
        .then(() => callback(null, `Successfully started ${event.instanceId}`))
        .catch(err => callback(err));
};
```

Check test configure



Configure test event

A function can have up to 10 test events. The events are persisted so you can switch to another computer or web browser and test your function with the same events.

● Create new test event

○ Edit saved test events

Event template

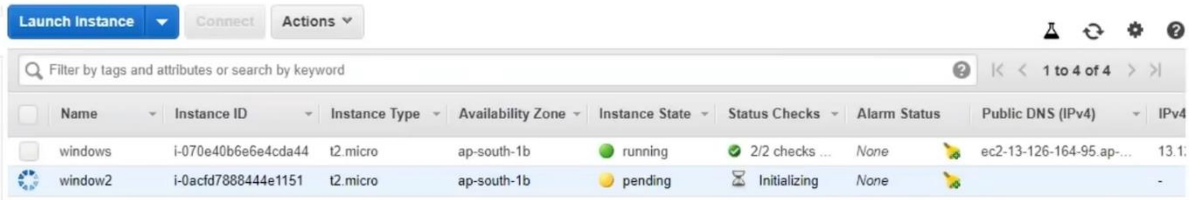
Hello World

Event name

startins

```
{
   "instanceRegion" : "ap-south-1" ,
   "instanceId" : "i-0acfd7888444e1151"
}
```

See successfully start EC2

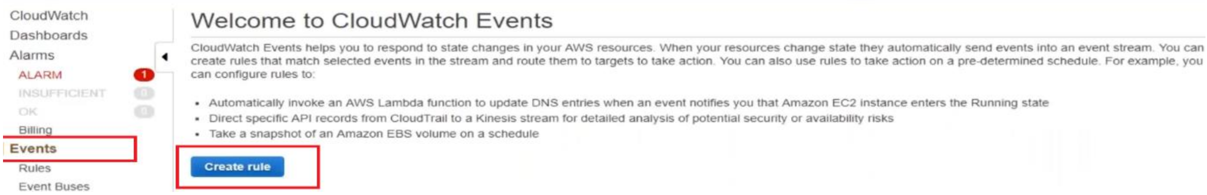


Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4
windows	i-070e40b6e6e4cda44	t2.micro	ap-south-1b	running	2/2 checks ...	None	ec2-13-126-164-95.ap-...	13.1:
window2	i-0acfd7888444e1151	t2.micro	ap-south-1b	pending	Initializing	None		-

CloudWatch

AUTOMATIC PROCESS THROUGH CLOUDWATCH

Open CloudWatch service select events and create rule

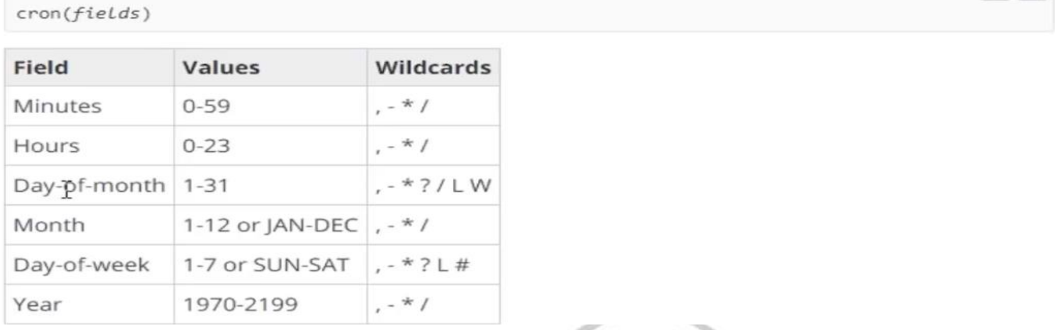


Welcome to CloudWatch Events

CloudWatch Events helps you to respond to state changes in your AWS resources. When your resources change state they automatically send events into an event stream. You can create rules that match selected events in the stream and route them to targets to take action. You can also use rules to take action on a pre-determined schedule. For example, you can configure rules to:

- Automatically invoke an AWS Lambda function to update DNS entries when an event notifies you that Amazon EC2 instance enters the Running state
- Direct specific API records from CloudTrail to a Kinesis stream for detailed analysis of potential security or availability risks
- Take a snapshot of an Amazon EBS volume on a schedule

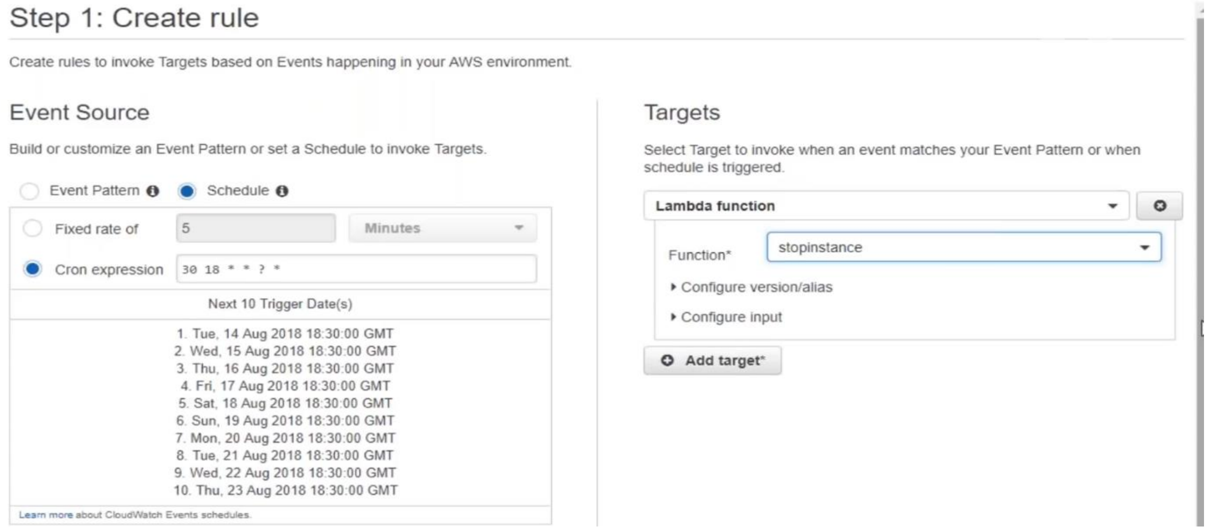
First read corn job format



```
cron(fields)
```

Field	Values	Wildcards
Minutes	0-59	, - * /
Hours	0-23	, - * /
Day-of-month	1-31	, - * ? / L W
Month	1-12 or JAN-DEC	, - * /
Day-of-week	1-7 or SUN-SAT	, - * ? L #
Year	1970-2199	, - * /

Configure stop EC2 Instance schedule 18:30



Step 1: Create rule

Create rules to invoke Targets based on Events happening in your AWS environment.

Event Source

Build or customize an Event Pattern or set a Schedule to invoke Targets.

○ Event Pattern ● Schedule

○ Fixed rate of 5 Minutes

● Cron expression `30 18 * * ? *`

Next 10 Trigger Date(s)

1. Tue, 14 Aug 2018 18:30:00 GMT
2. Wed, 15 Aug 2018 18:30:00 GMT
3. Thu, 16 Aug 2018 18:30:00 GMT
4. Fri, 17 Aug 2018 18:30:00 GMT
5. Sat, 18 Aug 2018 18:30:00 GMT
6. Sun, 19 Aug 2018 18:30:00 GMT
7. Mon, 20 Aug 2018 18:30:00 GMT
8. Tue, 21 Aug 2018 18:30:00 GMT
9. Wed, 22 Aug 2018 18:30:00 GMT
10. Thu, 23 Aug 2018 18:30:00 GMT

Learn more about CloudWatch Events schedules.

Targets

Select Target to invoke when an event matches your Event Pattern or when schedule is triggered.

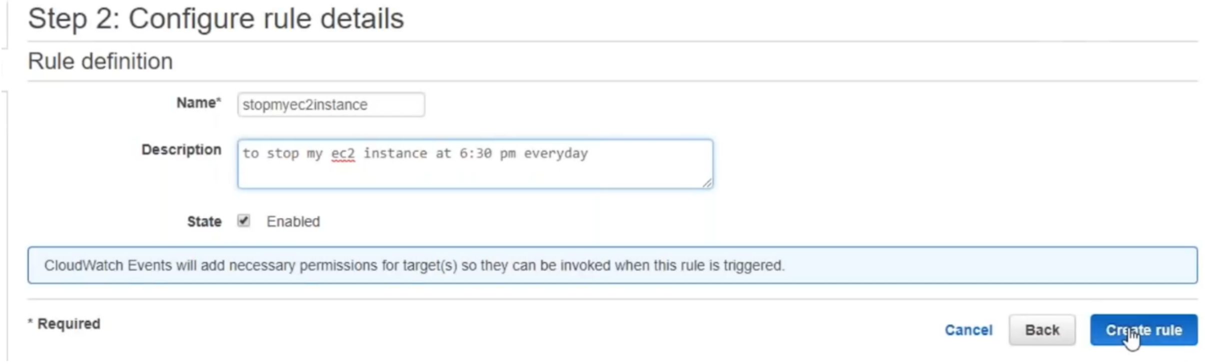
Lambda function

Function* stopinstance

▸ Configure version/alias

▸ Configure input

Configure rule details



Step 2: Configure rule details

Rule definition

Name* stopmyec2instance

Description to stop my ec2 instance at 6:30 pm everyday

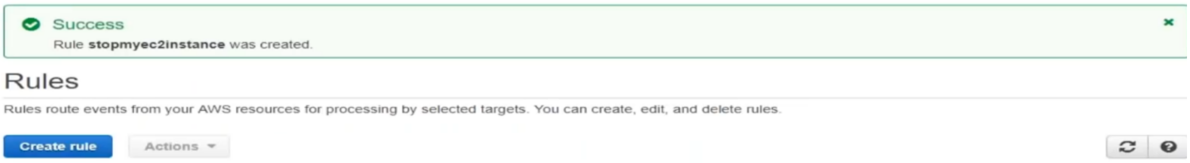
State ☑ Enabled

CloudWatch Events will add necessary permissions for target(s) so they can be invoked when this rule is triggered.

* Required

CloudWatch

Create successfully



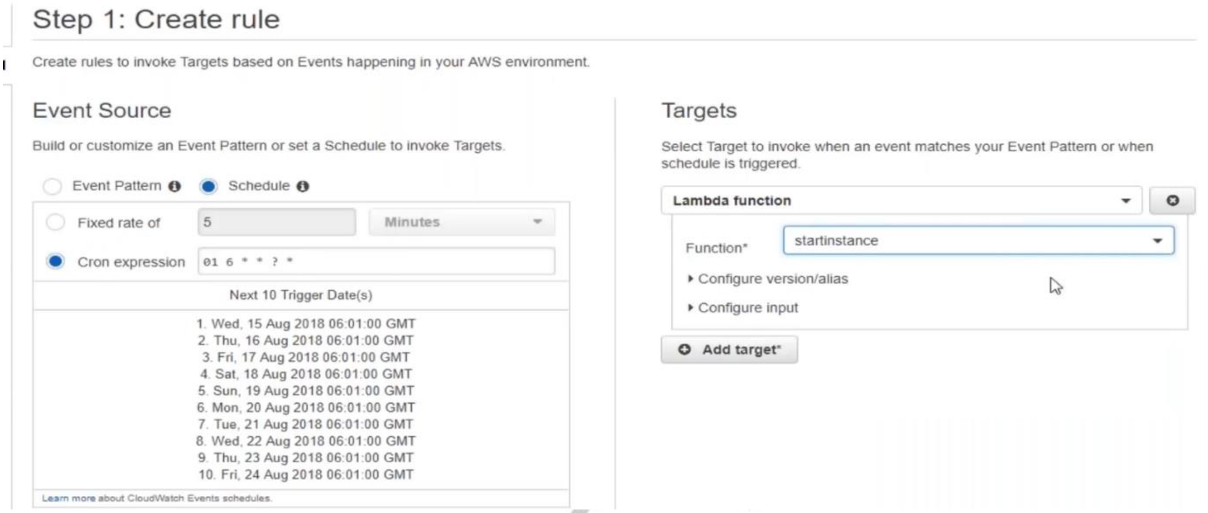
Success

Rule **stopmyec2instance** was created.

Rules

Rules route events from your AWS resources for processing by selected targets. You can create, edit, and delete rules.

Configure start EC2 Instance schedule 06:01 AM



Step 1: Create rule

Create rules to invoke Targets based on Events happening in your AWS environment.

Event Source

Build or customize an Event Pattern or set a Schedule to invoke Targets.

○ Event Pattern ● Schedule

○ Fixed rate of 5 Minutes

● Cron expression `01 6 * * ? *`

Next 10 Trigger Date(s)

1. Wed, 15 Aug 2018 06:01:00 GMT
2. Thu, 16 Aug 2018 06:01:00 GMT
3. Fri, 17 Aug 2018 06:01:00 GMT
4. Sat, 18 Aug 2018 06:01:00 GMT
5. Sun, 19 Aug 2018 06:01:00 GMT
6. Mon, 20 Aug 2018 06:01:00 GMT
7. Tue, 21 Aug 2018 06:01:00 GMT
8. Wed, 22 Aug 2018 06:01:00 GMT
9. Thu, 23 Aug 2018 06:01:00 GMT
10. Fri, 24 Aug 2018 06:01:00 GMT

Learn more about CloudWatch Events schedules.

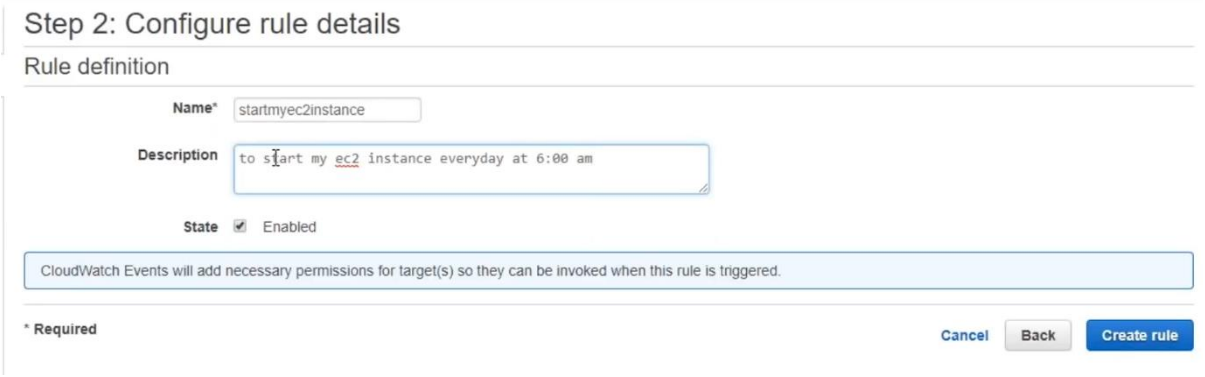
Targets

Select Target to invoke when an event matches your Event Pattern or when schedule is triggered.

Lambda function

Function* startinstance

Configure rule details



Step 2: Configure rule details

Rule definition

Name* startmyec2instance

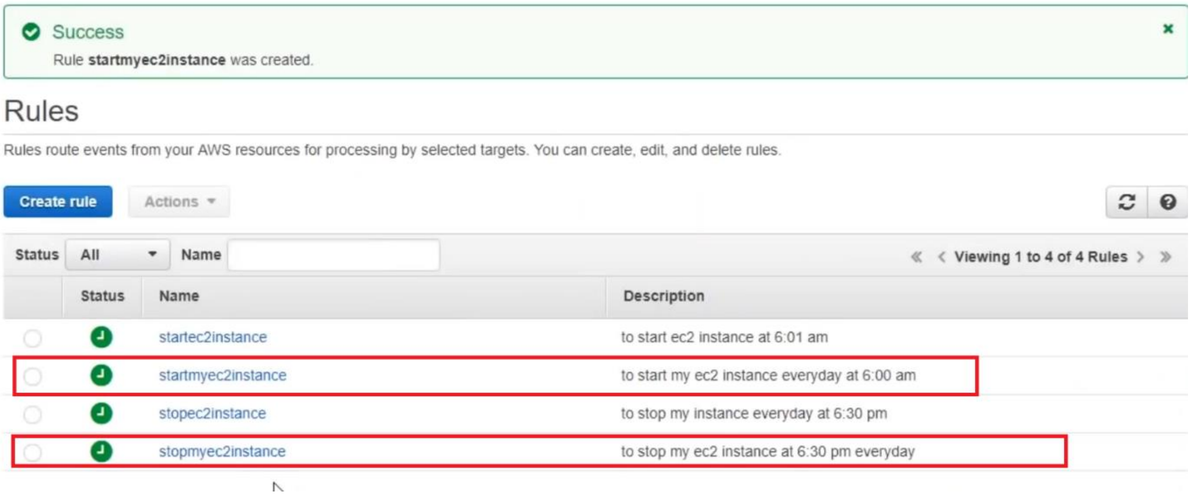
Description `to start my ec2 instance everyday at 6:00 am`

State ☑ Enabled

CloudWatch Events will add necessary permissions for target(s) so they can be invoked when this rule is triggered.

* Required

Successfully create schedule rules



Success

Rule **startmyec2instance** was created.

Rules

Rules route events from your AWS resources for processing by selected targets. You can create, edit, and delete rules.

Status All Name

Viewing 1 to 4 of 4 Rules

	Status	Name	Description
○	●	startec2instance	to start ec2 instance at 6:01 am
○	●	startmyec2instance	to start my ec2 instance everyday at 6:00 am
○	●	stopec2instance	to stop my instance everyday at 6:30 pm
○	●	stopmyec2instance	to stop my ec2 instance at 6:30 pm everyday