Prashant Lakhera
Apr 19 · 4 min read · Listen

100 Days of AWS — Day 16- Stop/Start EC2 instance on a scheduled basis to save cost

To view the complete course, please enroll it using the below link(it's free)

https://www.101daysofdevops.com/courses/100-days-of-aws/

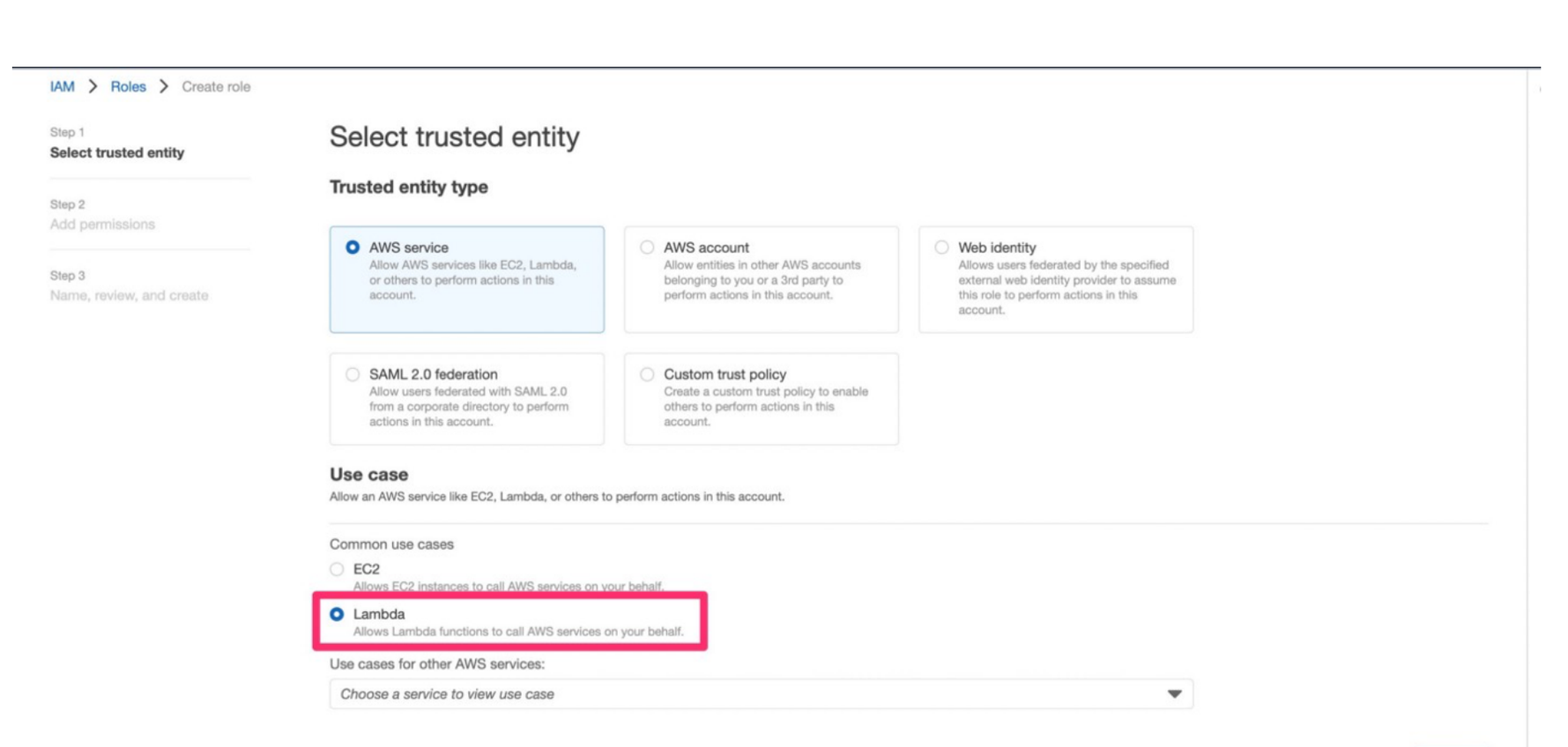
Welcome to Day 16 of 100 Days of AWS. The topic for today is Stop/Start EC2 instance on a scheduled basis to save cost.

This is one of the ask I came across in Dev env to save money where you need to shut down all the EC2 instance on a scheduled basis and bring it back the next day. To achieve that, we use Lambda in the combination of CloudWatch Events.

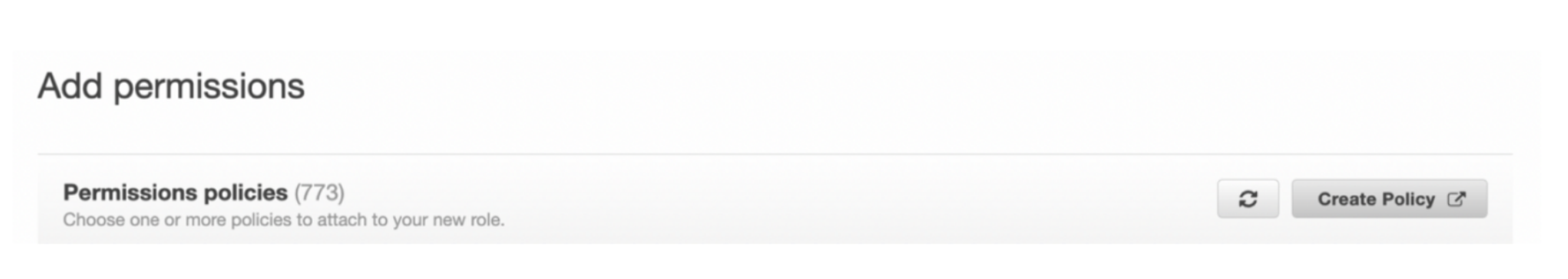
These are the steps we need to perform

***Step1:** Create IAM Role so that Lambda can interact with CloudWatch Events*

```
Go to IAM Console https://console.aws.amazon.com/iam --> Roles -->
Create role
```

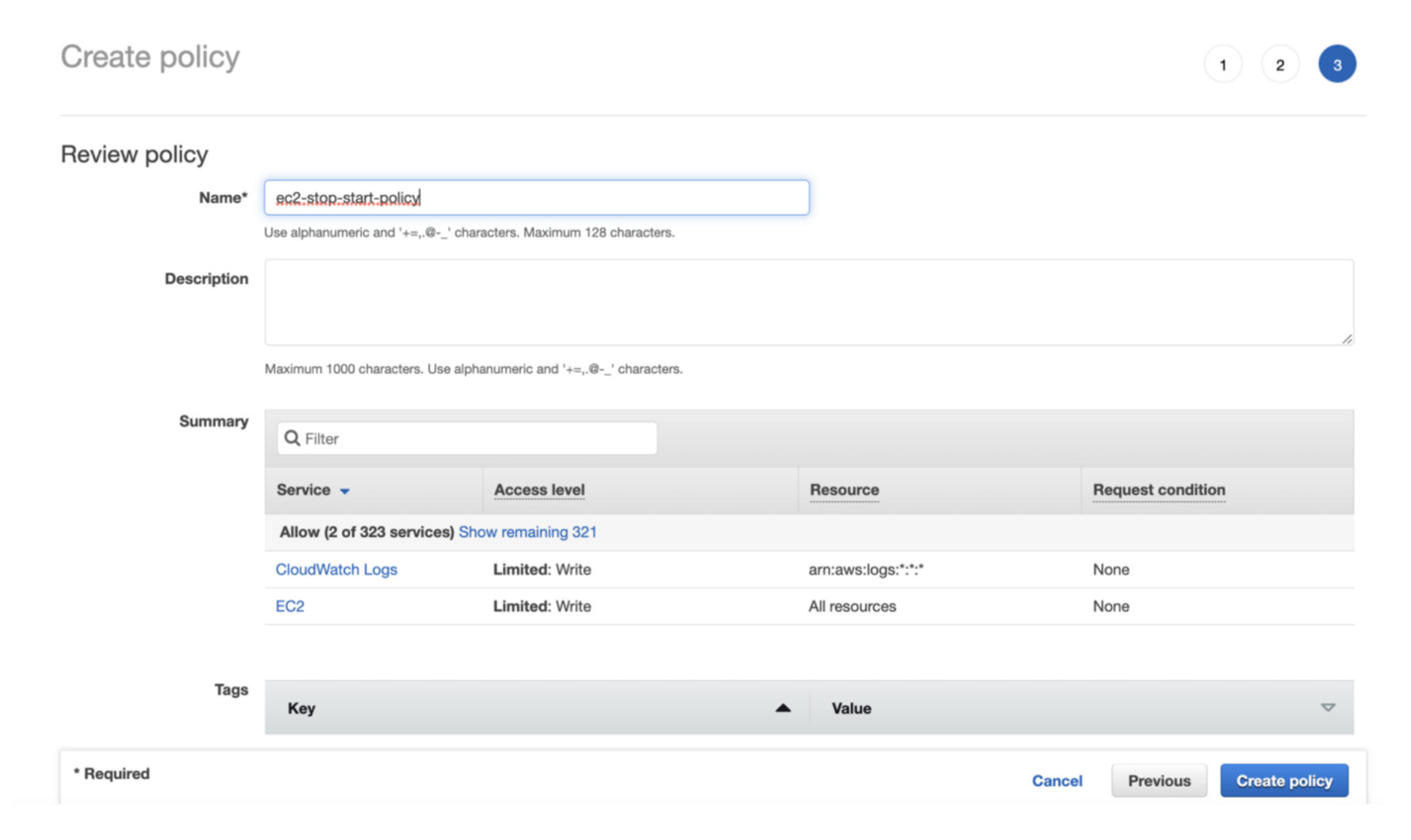


- *In the next screen select on Create Policy and paste the following policy*



```
{
"Version": "2012-10-17",
"Statement": [
{
"Effect": "Allow",
"Action": [
"logs:CreateLogGroup",
"logs:CreateLogStream",
"logs:PutLogEvents"
],
"Resource": "arn:aws:logs:*:*:*"
},
{
"Effect": "Allow",
"Action": [
"ec2:Start*",
"ec2:Stop*"
],
"Resource": "*"
}
]
}
```

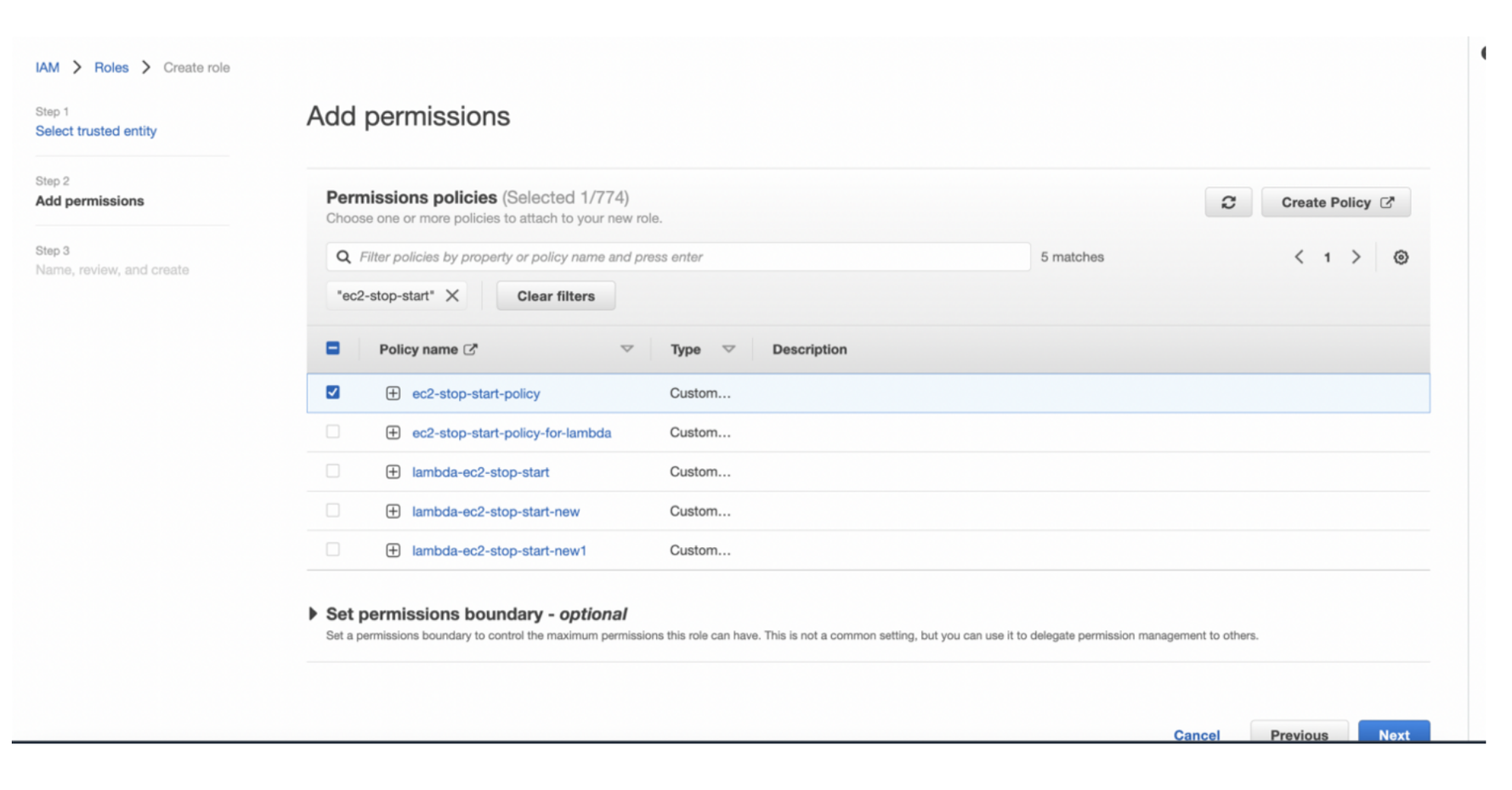
- *Give your policy some name and click on Create policy*



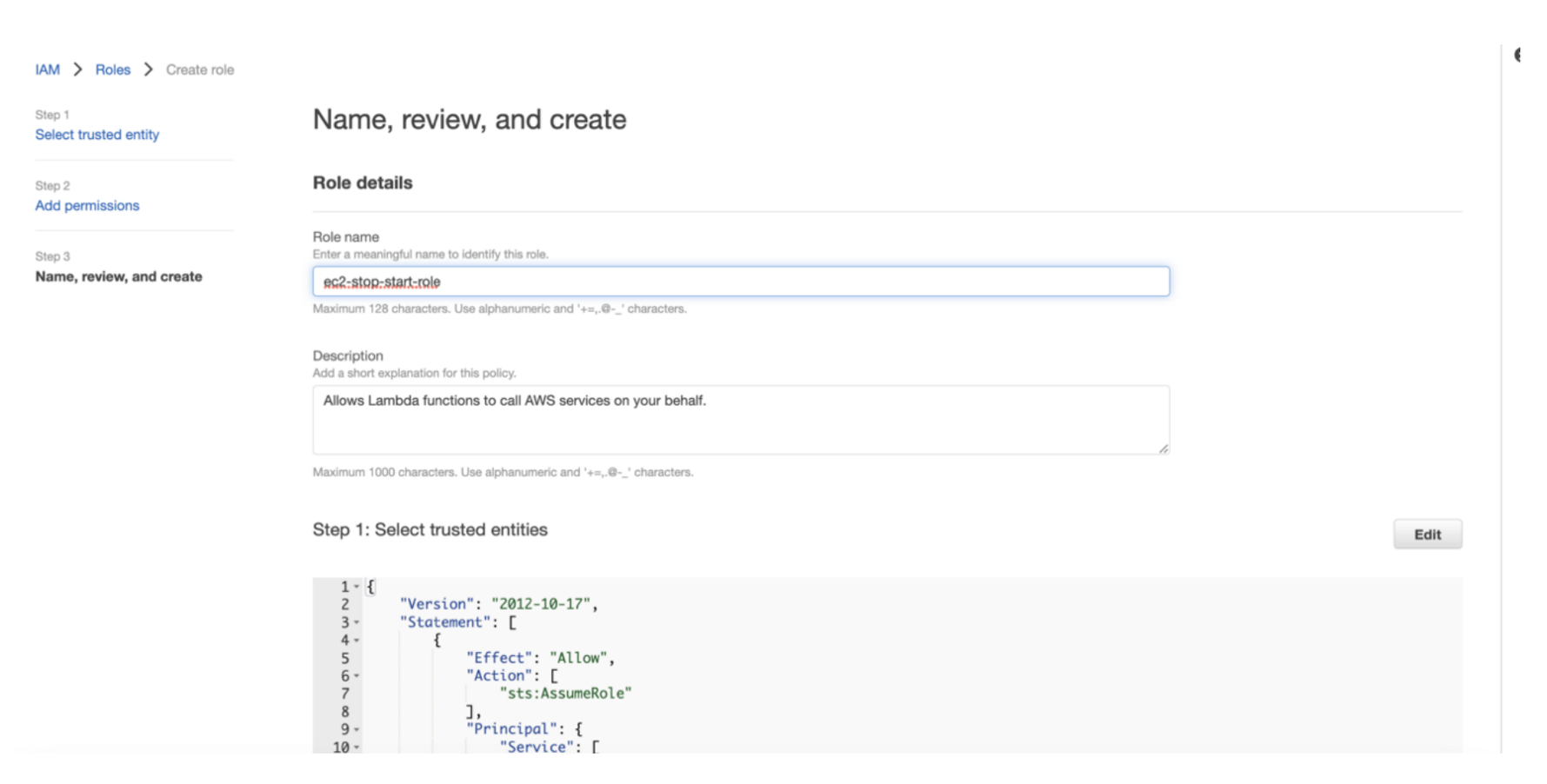
- *Click back on Roles and Create role*

```
Go to IAM Console https://console.aws.amazon.com/iam --> Roles -->
Create role
```

- *Select AWS service as Trusted entity type and Use case Lambda*
- *This time select ec2-stop-start-policy we created in the previous steps*

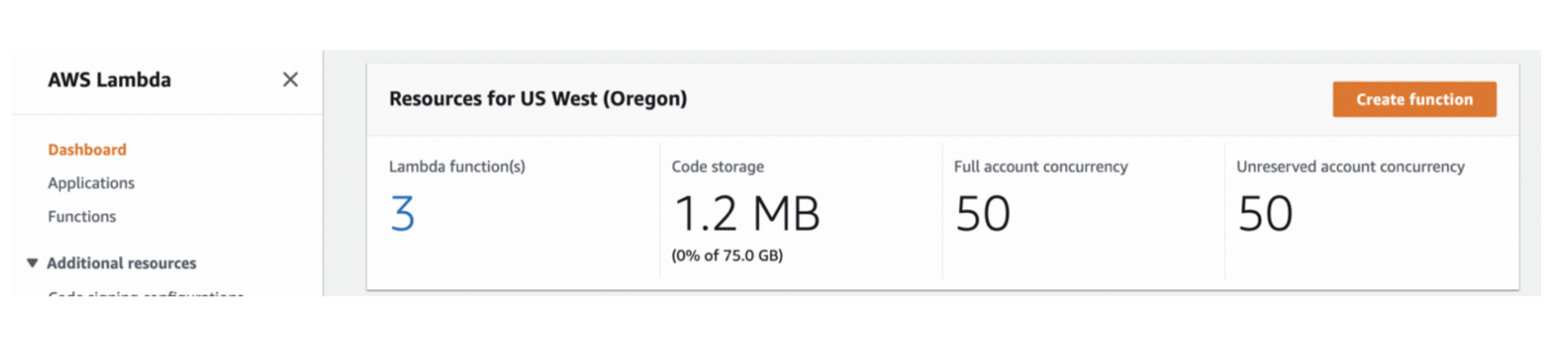


- *Give your role some meaningful name and click Create role at the bottom of screen.*

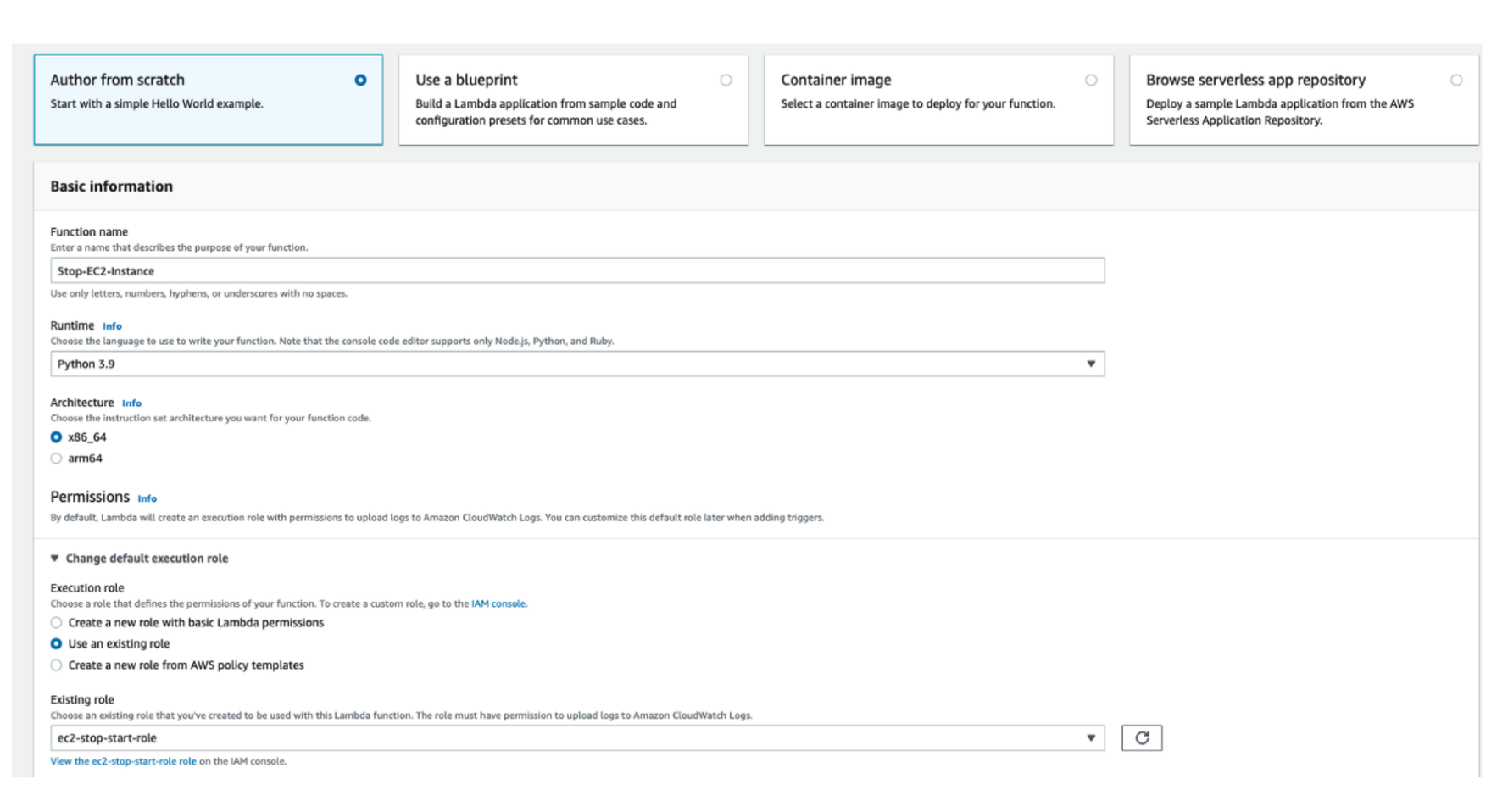


***Step2:** Create Lambda function*

- *Go to Lambda https://us-west-2.console.aws.amazon.com/lambda/home?region=us-west-2#/home*
- *Select Create function*



```
* Select Author from scratch
* Name: Give your Lambda function any name
* Runtime: Select Python3.9 as runtime
* Role: Choose the role we create in first step(ec2-stop-start-role)
* Click on Create function
```



- *To stop the instance, the code will look like this*

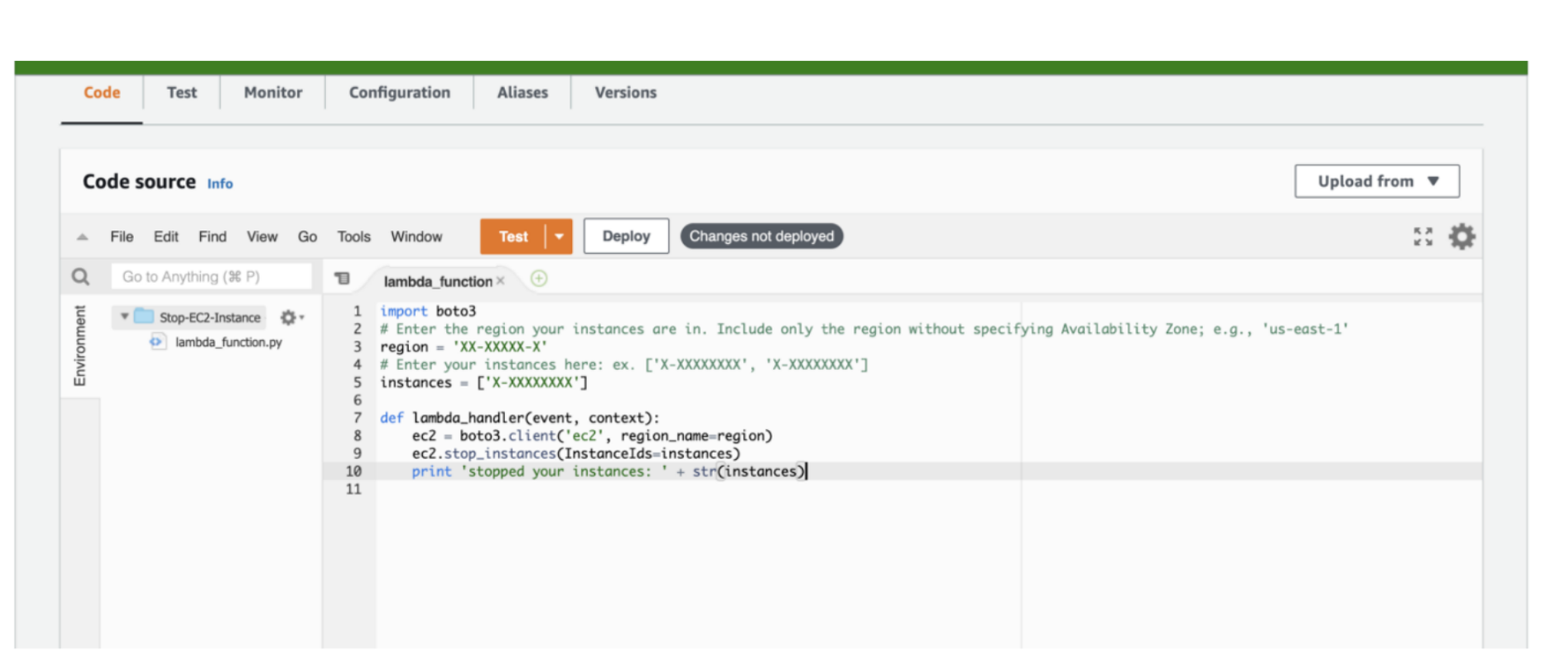
```
import boto3
# Enter the region your instances are in. Include only the region without specifying Availability Zone; e.g., 'us-east-1'
region = 'XX-XXXXX-X'
# Enter your instances here: ex. ['X-XXXXXXXX', 'X-XXXXXXXX']
instances = ['X-XXXXXXXX']

def lambda_handler(event, context):
    ec2 = boto3.client('ec2', region_name=region)
    ec2.stop_instances(InstanceIds=instances)
    print 'stopped your instances: ' + str(instances)
```

https://gist.githubusercontent.com/100daysofdevops/82b7db36251ca4dbcc2c5bfcc6105639/raw/83c46ee153ff58a73470880f571be3dde179d734/ec2_instance_stop.py

```
* Change the Value of region
* In the instance field specify instance id
```

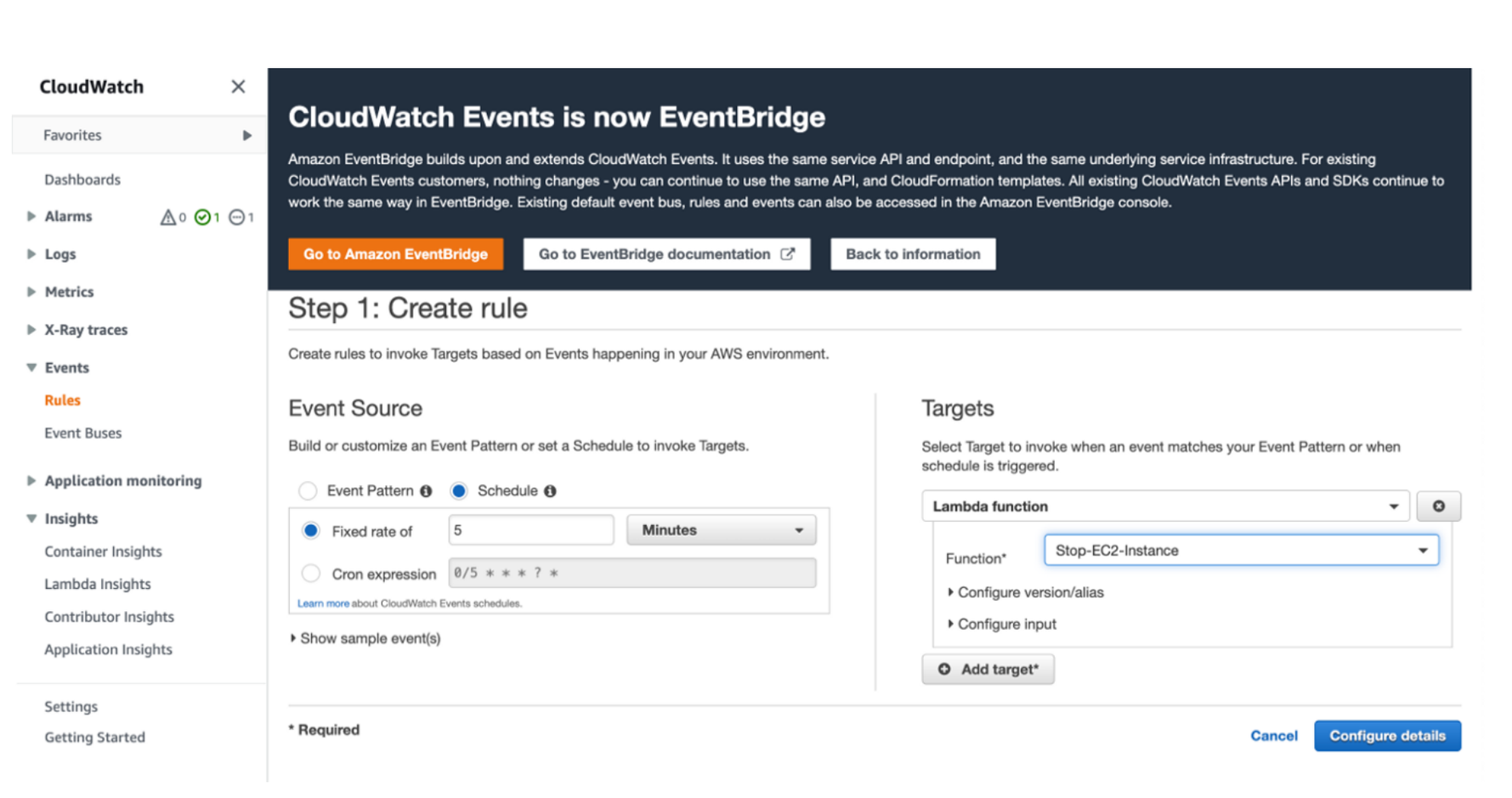
- *Keep all the settings as default, just change the timeout value to 10sec*
- *Now we need to perform the same steps for starting the instance*



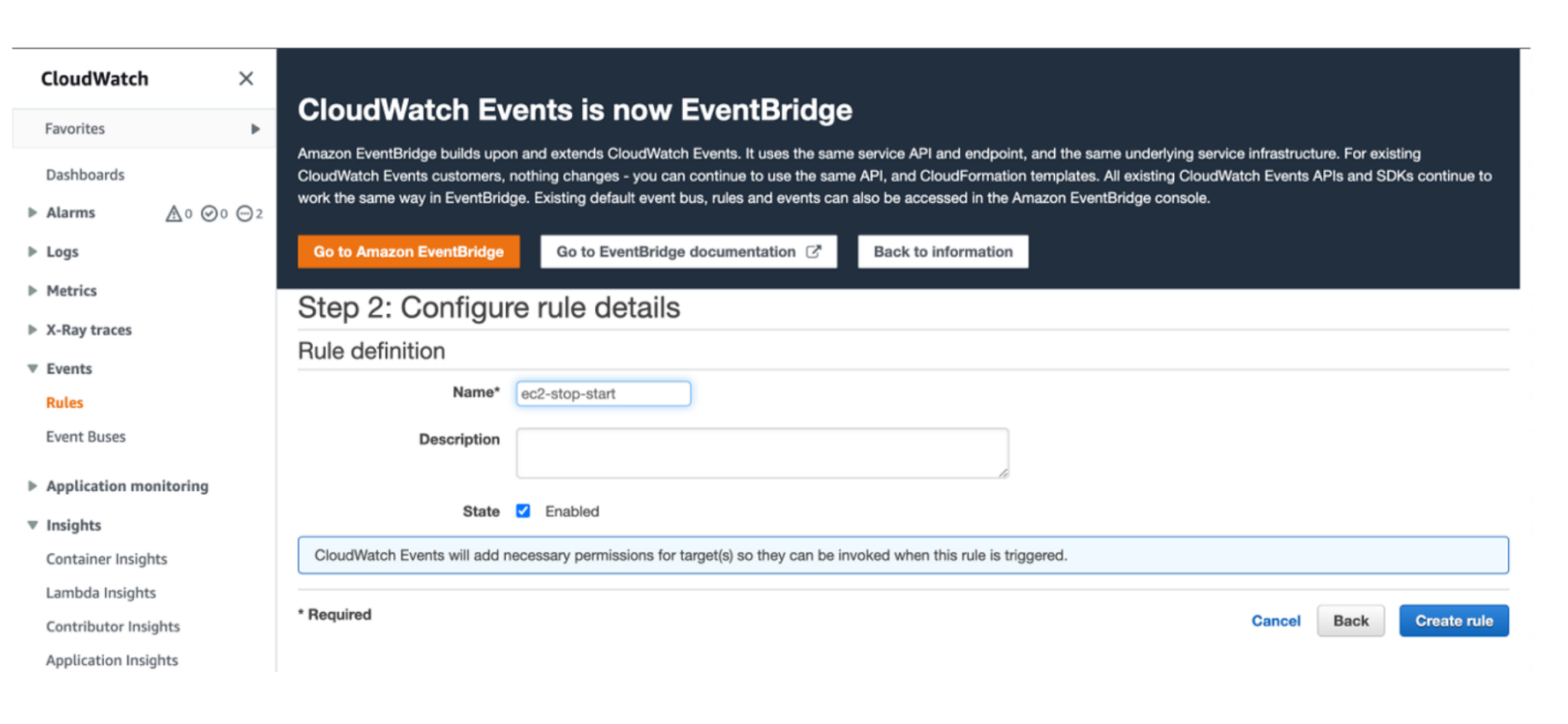
- *Click on Deploy*

***Step3:** Create the CloudWatch event to trigger this Lambda function*

- *Open the Amazon CloudWatch console.*
- *Choose Events, and then choose Create rule.*
- *Choose Schedule under Event Source.*



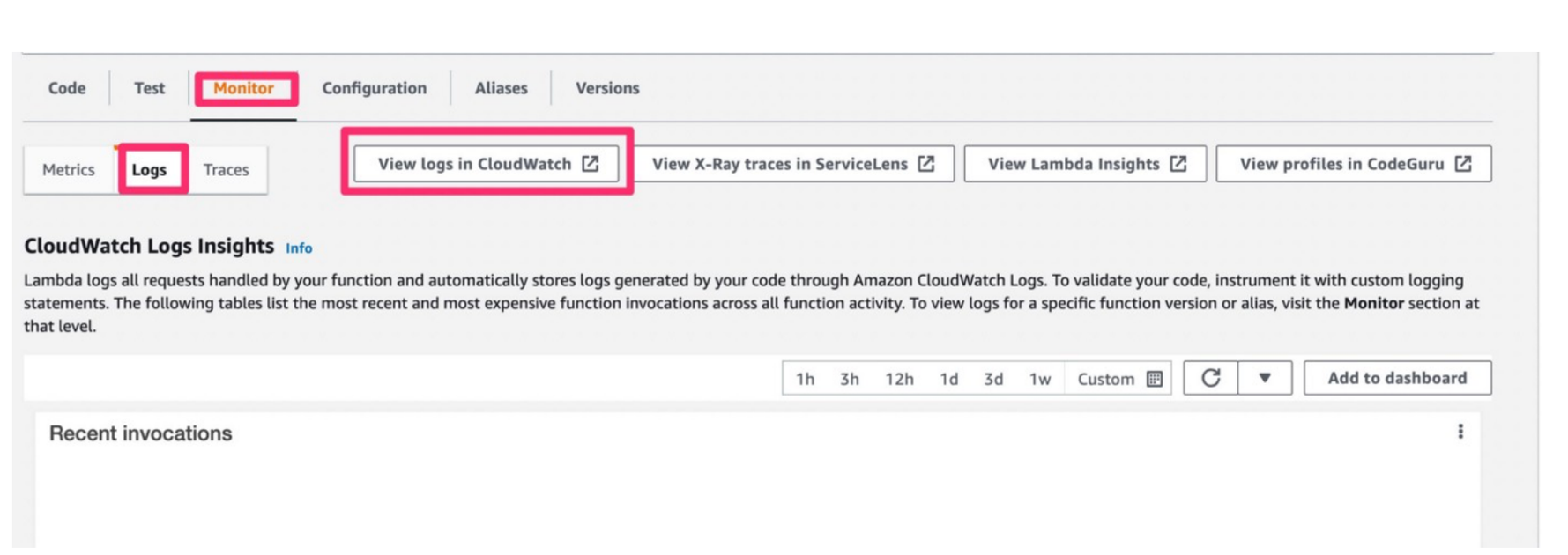
- *Under Cron expression choose * 18 * * ? * (If you want to shutdown your instance at 6pm everyday)*
- *Choose Add target, and then choose Lambda function that you created earlier to stop the instance*
- *Click on Configure details*

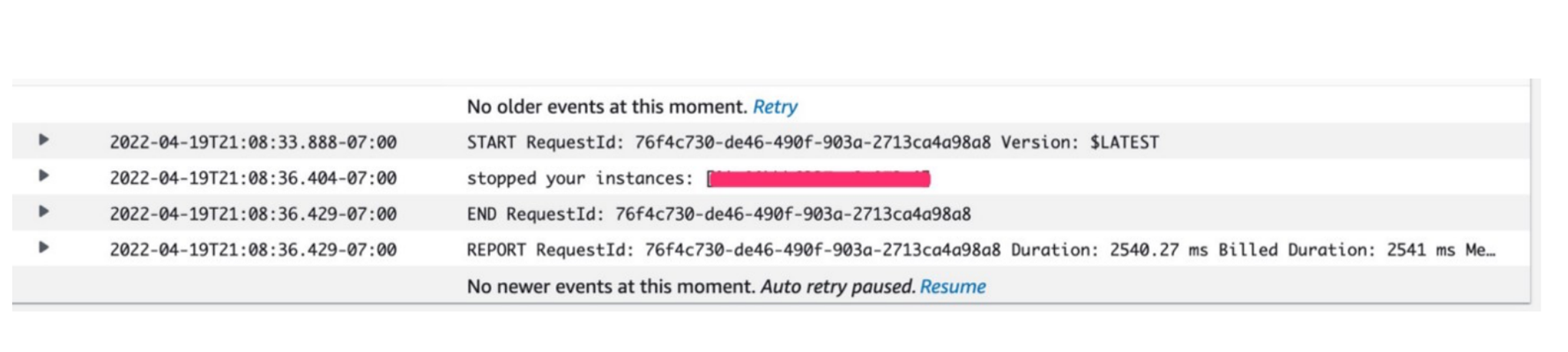


- *Give your rule some name and click on Create rule.*

***NOTE:** One very important point to note is that all scheduled event is in **UTC timezone**, so you need to customize it based on your timezone.*

- *Go back to your Lambda and click on View logs in CloudWatch*





- *The simple automation system is ready in stopping/starting the instance and to save some company money.*