**IAM Role**

1. Create Iam Role Using Aws Iam Service
2. Open Dashboard Role>Create Role>Aws Service>Use Case(lambda)
3. Attach policy **AWSLambdaBasicExecutionRole** , **AmazonEC2ReadOnlyAccess** , **AmazonEC2FullAccess** , [AWSLambda\_FullAccess](https://us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/policies/details/arn%3Aaws%3Aiam%3A%3Aaws%3Apolicy%2FAWSLambda_FullAccess) Then Click On Create.

**Lambda**

1. Create lambda Using Aws Lambda Function.
2. Open Dashboard>Create function>Author From Scratch>Set Function Name>Python>**Change default execution role**>Existing Rule>Use An Existing Role>Select Created Role>Create Function
3. Peste Following Code

import boto3

import datetime

def lambda\_handler(event, context):

ec2 = boto3.client('ec2')

# Specify the instance ID you want to monitor

instance\_id = 'i-xxxxxxxxxxxxxxxxx' # Replace with your instance ID

# Describe the instance

response = ec2.describe\_instances(InstanceIds=[instance\_id])

instance = response['Reservations'][0]['Instances'][0]

# Get the instance state and launch time

state = instance['State']['Name']

launch\_time = instance['LaunchTime']

# Check if the instance is running

if state == 'running':

# Calculate the running time

running\_time = datetime.datetime.now(datetime.timezone.utc) - launch\_time

# Check if the instance has been running for more than 5 minutes

if running\_time.total\_seconds() > 300: # 5 minutes = 300 seconds

# Stop the instance

ec2.stop\_instances(InstanceIds=[instance\_id])

print(f'Stopped instance: {instance\_id}')

else:

print(f'Instance {instance\_id} has been running for {running\_time.total\_seconds()} seconds.')

else:

print(f'Instance {instance\_id} is not running. Current state: {state}')

1. Click On Deploy Then Cilck On Test Create Event > Set Event Name > Save Again Click On Test > Code Running > Check Bugs / Success > If Get Error Of Time – Open Configuration Settings > Edit > Increase Timeout >Save.

**EventBridge**

1. Then Set Trigger Of Event Bridge First Create Event Bridge Using Following Steps. Open Event Bridge > Dashboard > Create Rule > Set Name > Description > Rule Type (schedule) > Continue To Create Rule > Set Schedule Pattern > Rate or Cron > Select Rate > Set Trigger Time >Next > Target Type (Aws Service) > Ec2 Stop Instance Api Call > Next >Next >Done.
2. Open Lambda > Set Trigger > Select Source (Eventbridge) > Select Created Schedule > Done.
3. Check Logs in Cloudwatch >Log > Log Group And Monitoring in Lambda.
4. Its Working Ec2 Instance Automatically Stop After 5 Min.

* Create Policy in AWS lambda to stop ec2 instance when I upload file on S3 ?

**IAM Policy**

{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

"Action": "ec2:StopInstances",

"Resource": "\*"

},

{

"Effect": "Allow",

"Action": [

"logs:CreateLogGroup",

"logs:CreateLogStream",

"logs:PutLogEvents"

],

"Resource": "\*"

}

]

}

Attach This Policy To IAM Role.

**LAMBDA CODE**

import boto3

def lambda\_handler(event, context):

ec2 = boto3.client('ec2')

# Replace with your actual EC2 instance ID

instance\_id = 'i-0123456789abcdef0'

try:

response = ec2.stop\_instances(InstanceIds=[instance\_id])

print(f"Stopping instance: {instance\_id}")

return response

except Exception as e:

print(f"Error stopping instance: {str(e)}")

raise