**DevOps-task**

**Task :**

Using Lambda Function to Start & Stop the EC2 Instance Using Cloud watch Events.

- Create function to start ec2 instance at 10 AM IST if instance has tag ec2\_start = "true".

- Create function to stop ec2 instance at 10 PM IST if instance has tag ec2\_stop = "true".

- Outcome is ec2 instance should start or stop automatically based on cron job schedule.

- Keep the code stuff into git repository.

- Document all the execution steps.

**Services used :**

**AWS lambda**

**AWS cloudwatch**

**IAM (Identity and Access Management)**

**git**

**Steps to do task:**

**step1 : Create ec2-instance**

**step2: Create Iam policy and set roles**

**step3: Create lambda fuction to start and stop code and write code in python to automate start and stop**

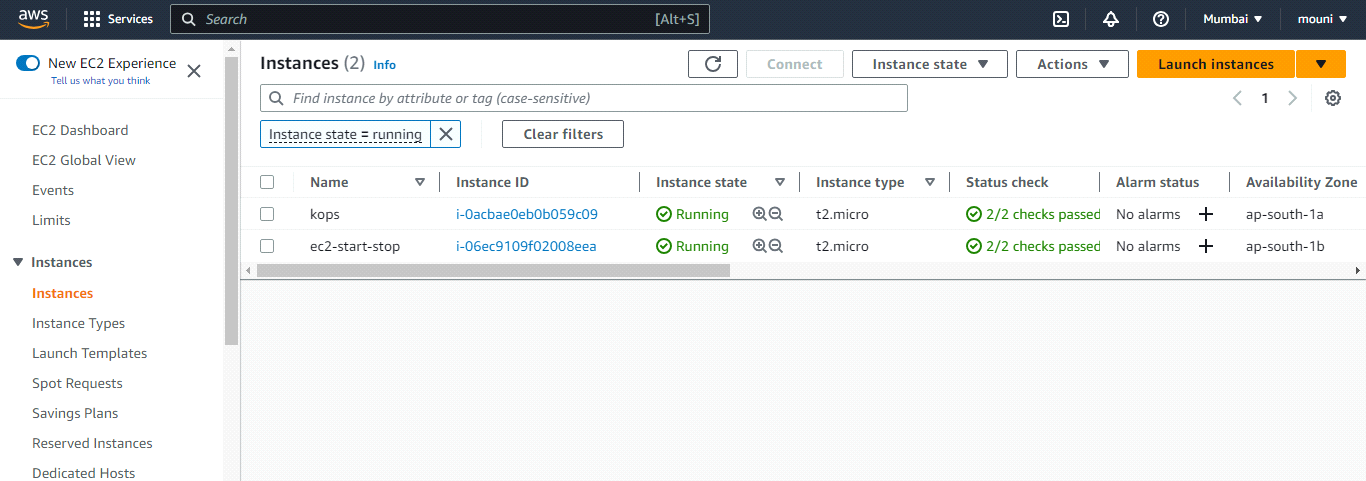
**step4: Depoly and test the aws lambda function**

**step5 : Add lambda trigger**

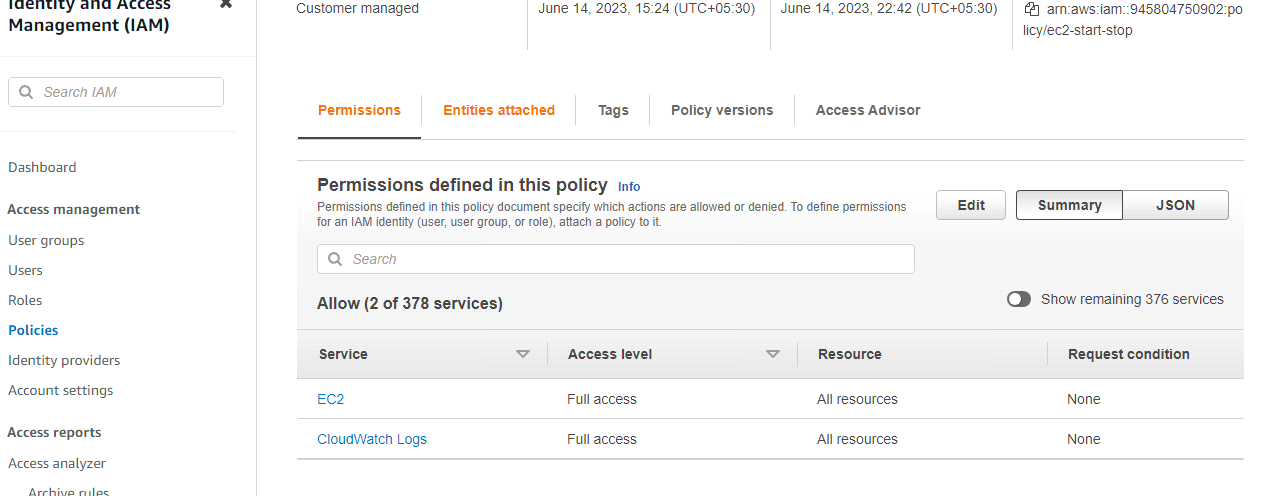
**step6: Add cloudwatch and set cronjob time to 10 am to start and 10pm to stop.**

**Detailed process:**

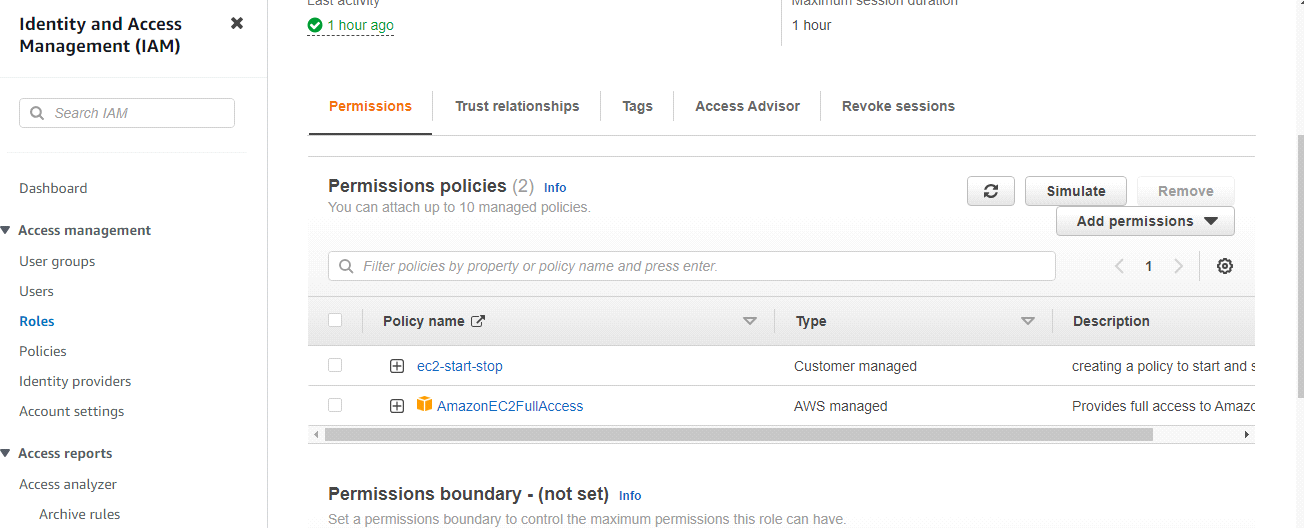
First of all we should create ec2 instance and we wan to give two tags ec2\_start = true and ec2\_stop = true to test the code



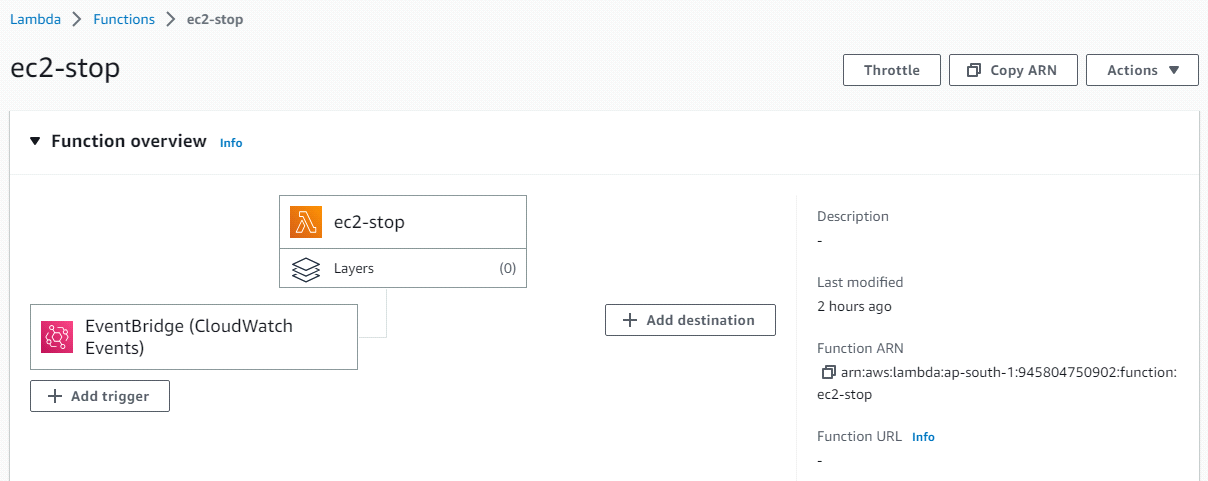
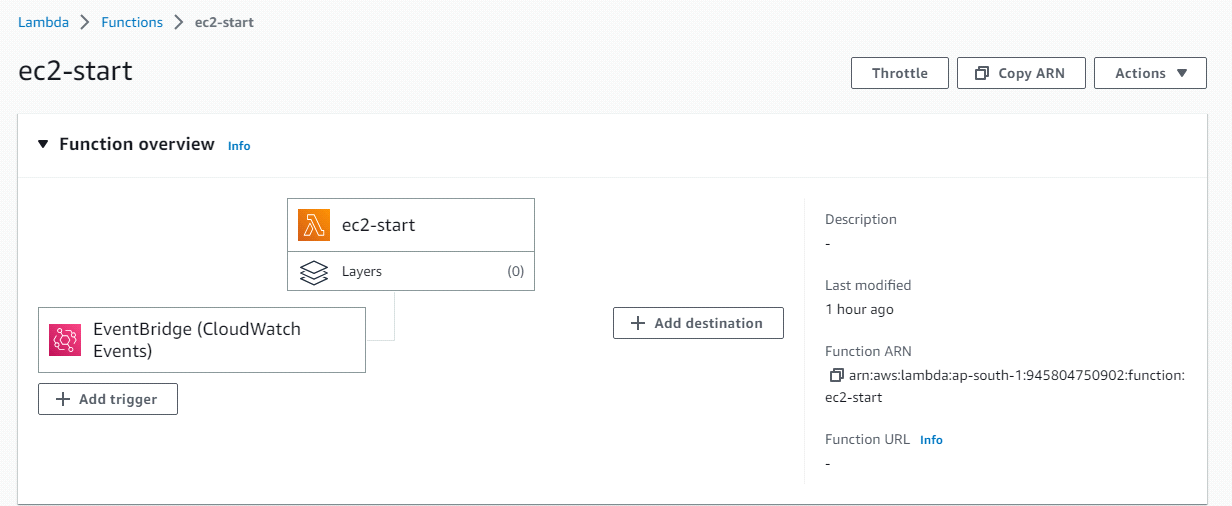
then switch to aws iam , and then create i am policies and give full permissions for ec2 and cloudwatch logs



after creating policies , create role and attach these policies by giving amazonEc2full access

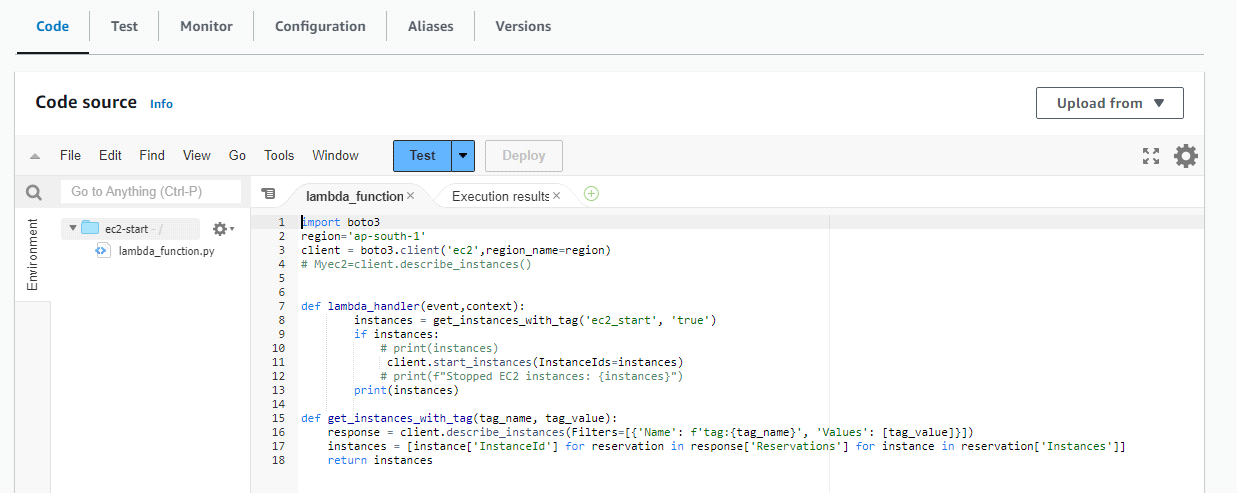


after setting up of i am we want to create lambda functions for both ec2\_start and ec2\_stop



then we want to write python code to ec2-instance start and stop , then we want to depoly and test the code

**Code to start ec2-instance in lambda function using python 3.9**



**CODE:**

**import boto3**

**region='ap-south-1'**

**client = boto3.client('ec2',region\_name=region)**

**# Myec2=client.describe\_instances()**

**def lambda\_handler(event,context):**

**instances = get\_instances\_with\_tag('ec2\_start', 'true')**

**if instances:**

**# print(instances)**

**client.start\_instances(InstanceIds=instances)**

**# print(f"Stopped EC2 instances: {instances}")**

**print(instances)**

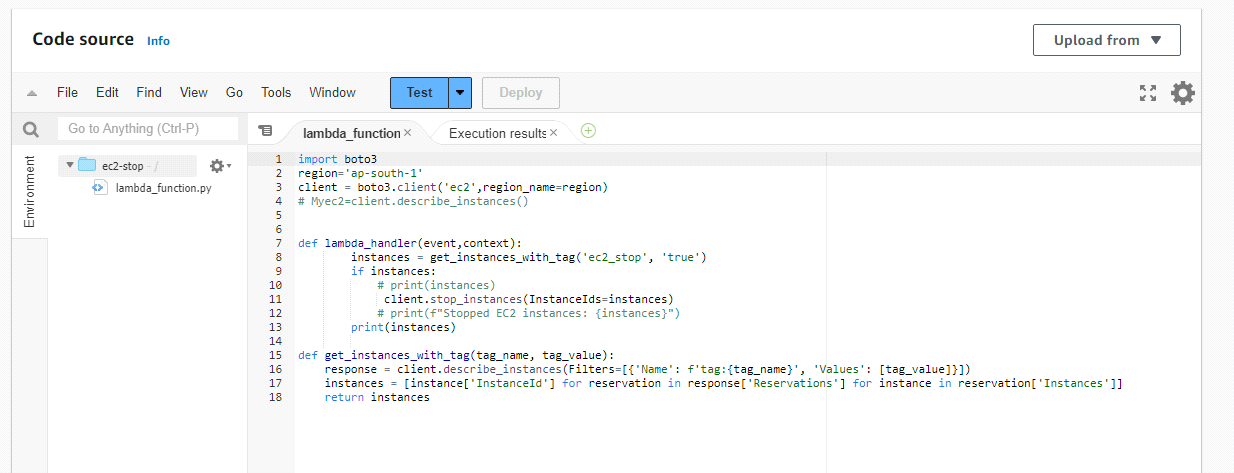
**def get\_instances\_with\_tag(tag\_name, tag\_value):**

**response = client.describe\_instances(Filters=[{'Name': f'tag:{tag\_name}', 'Values': [tag\_value]}])**

**instances = [instance['InstanceId'] for reservation in response['Reservations'] for instance in reservation['Instances']]**

**return instances**

**Code to stop ec2-instance in lambda function using python 3.9**



**code:**

**import boto3**

**region='ap-south-1'**

**client = boto3.client('ec2',region\_name=region)**

**# Myec2=client.describe\_instances()**

**def lambda\_handler(event,context):**

**instances = get\_instances\_with\_tag('ec2\_start', 'true')**

**if instances:**

**# print(instances)**

**client.start\_instances(InstanceIds=instances)**

**# print(f"Stopped EC2 instances: {instances}")**

**print(instances)**

**def get\_instances\_with\_tag(tag\_name, tag\_value):**

**response = client.describe\_instances(Filters=[{'Name': f'tag:{tag\_name}', 'Values': [tag\_value]}])**

**instances = [instance['InstanceId'] for reservation in response['Reservations'] for instance in reservation['Instances']]**

**return instances**

After testing the code ec2-instance will start by testing ec2-start lambda function and then code ec2-instance will stop by testing ec2-stop lambda function, this is manual process to automate the process we want to setup cloudwatch.

**Cloudwatch setup:**

For our final step, let’s go to CloudWatch in the AWS Management Console. We’re going to create a rule that will start our instances at a specific time.

**setup cron job:**

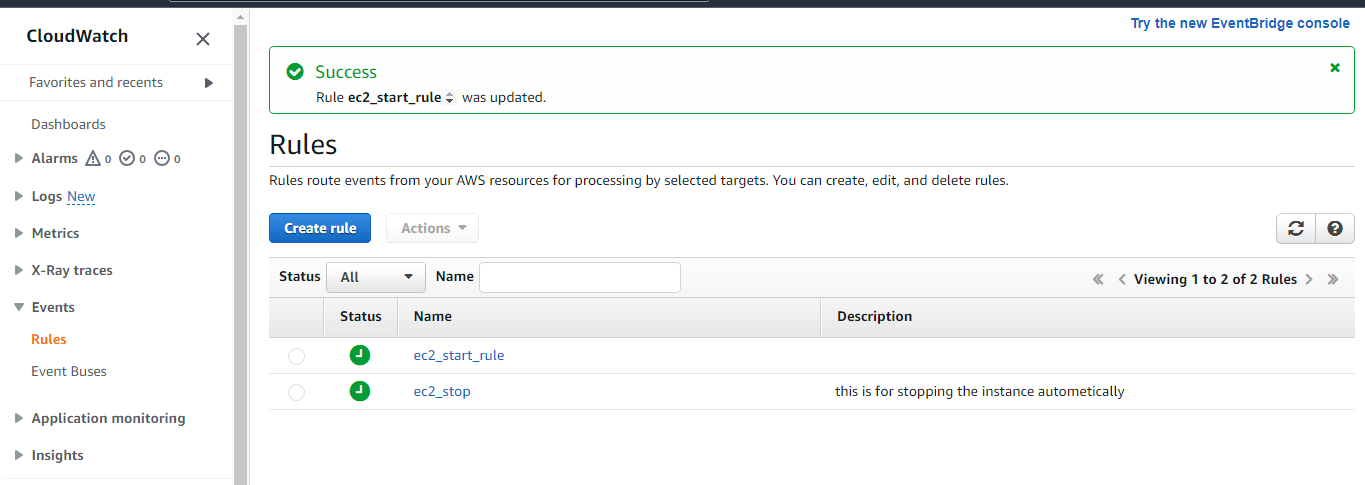
Here's how you can set up a cron job using AWS Lambda and EventBridge:

Create an AWS Lambda function: If you haven't already, create an AWS Lambda function that contains the code you want to execute as part of your cron job. This function can be written in a supported programming language such as Python, Node.js, Java, or others.

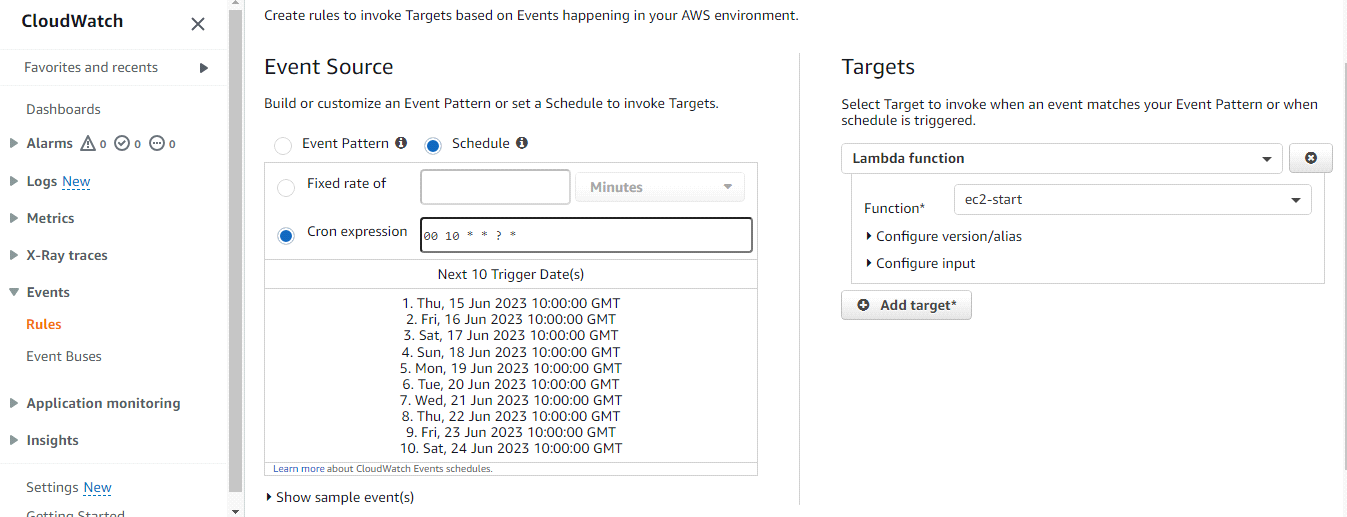
Set up an EventBridge rule: Go to the EventBridge service in the AWS Management Console and create a new rule. Specify the scheduling pattern using the cron syntax. For example, to run a Lambda function every day at 9:30 AM, you would set the cron expression to cron(30 9 \* \* ? \*).

Configure the target: In the same EventBridge rule configuration, specify the AWS Lambda function as the target for the rule. This means that when the cron job triggers, it will invoke the Lambda function.

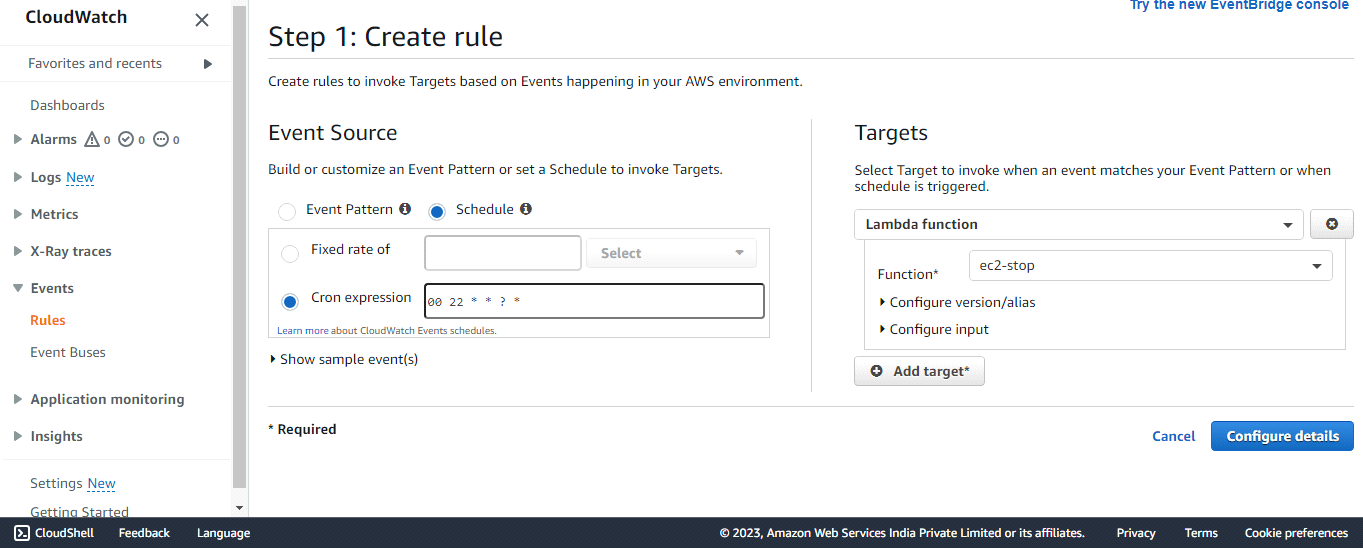
Save and activate the rule: Save the EventBridge rule and activate it. The cron job is now scheduled to run at the specified intervals.



**cronjob to start ec2-instance everyday at 10:00am**



**Cron job to stop ec2-instance everyday at 10:00pm**



**Git push :**

After completing the task we have created folder to store python codes to start and stop ec2 instances , then we pushed our code from local repository to central repository

step by step commands to push code from local to central

* git init
* git add .
* git status
* git commit -m "my\_new"
* git remote add origin "<https://github.com/Mounika7997/aws_lambda.git>"
* git remote -v
* git branch
* git push -v origin master