**Graded Assignment On Serverless Architecture**

**Assignment 1: Automated Instance Management Using AWS Lambda and Boto3**

**Objective:** In this assignment, you will gain hands-on experience with AWS Lambda and Boto3, Amazon's SDK for Python. You will create a Lambda function that will automatically manage EC2 instances based on their tags.

**Task:** You're tasked to automate the stopping and starting of EC2 instances based on tags. Specifically:

1. Setup:

   - Create two EC2 instances.

   - Tag one of them as `Auto-Stop` and the other as `Auto-Start`.

2. Lambda Function Creation:

   - Set up an AWS Lambda function.

   - Ensure that the Lambda function has the necessary IAM permissions to describe, stop, and start EC2 instances.

3. Coding:

   - Using Boto3 in the Lambda function:

     - Detect all EC2 instances with the `Auto-Stop` tag and stop them.

     - Detect all EC2 instances with the `Auto-Start` tag and start them.

4. Testing:

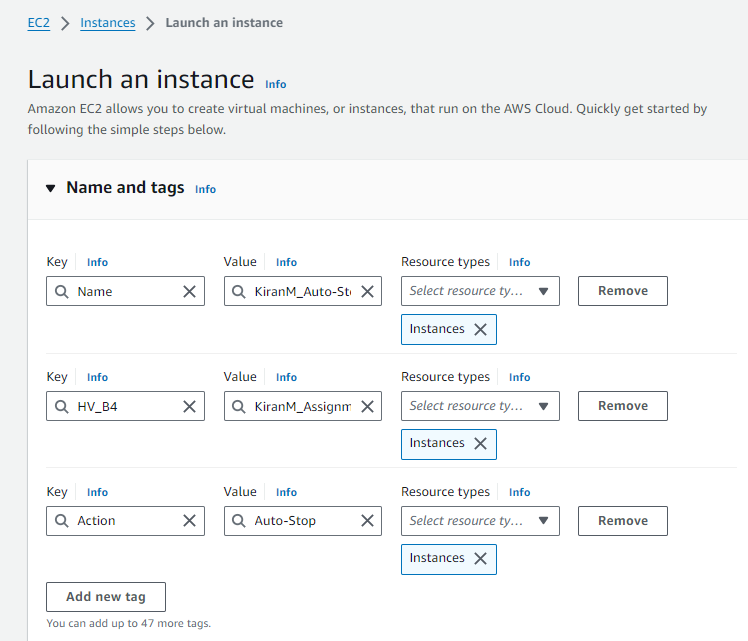
   - Manually invoke the Lambda function.

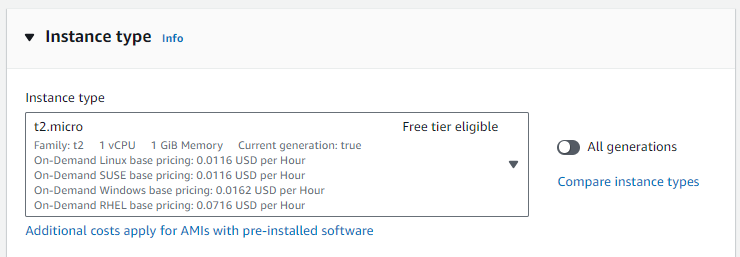
   - Confirm that the instance tagged `Auto-Stop` stops and the one tagged `Auto-Start` starts.

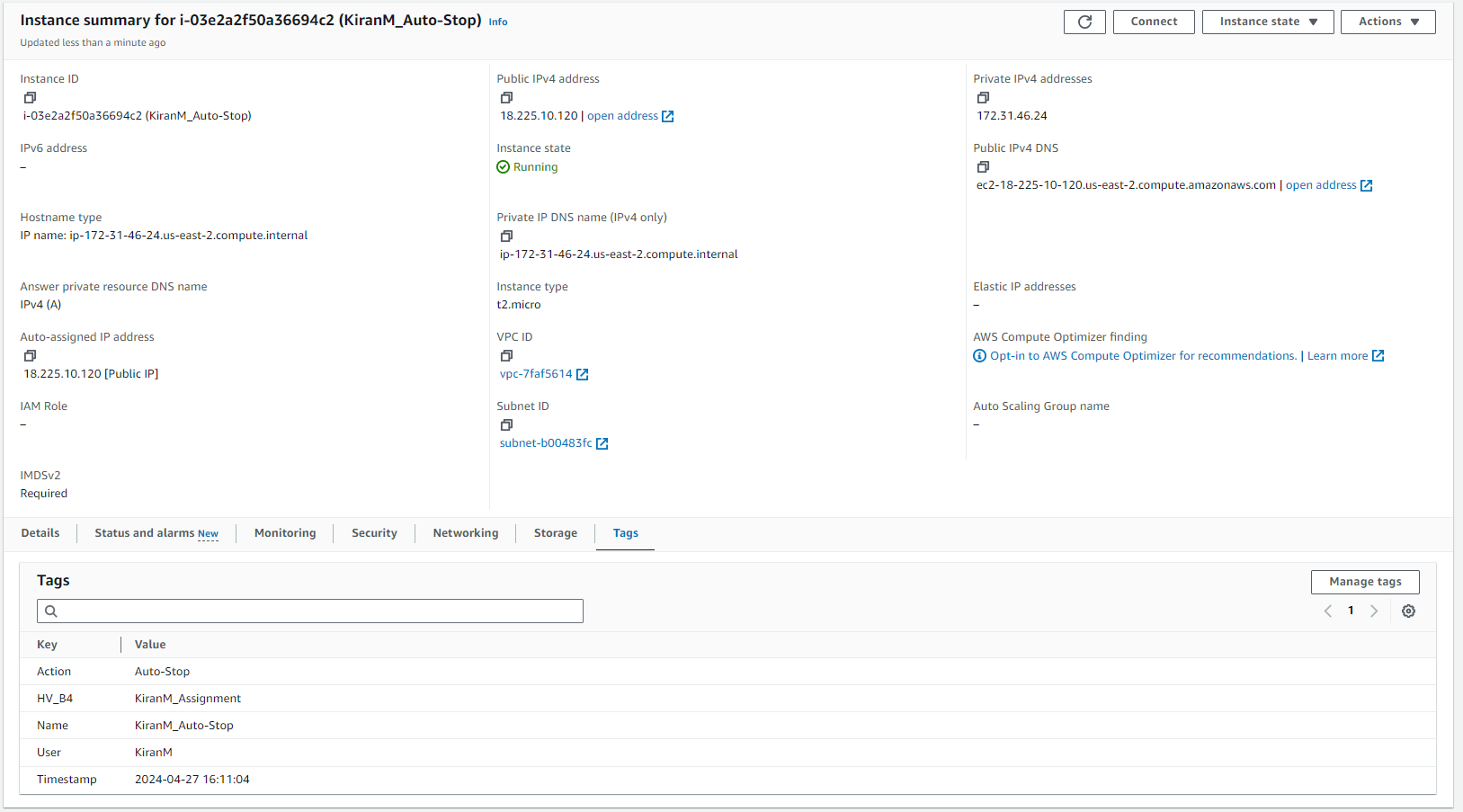
1. **SetUp:**

We will create 2 EC2 instances here.

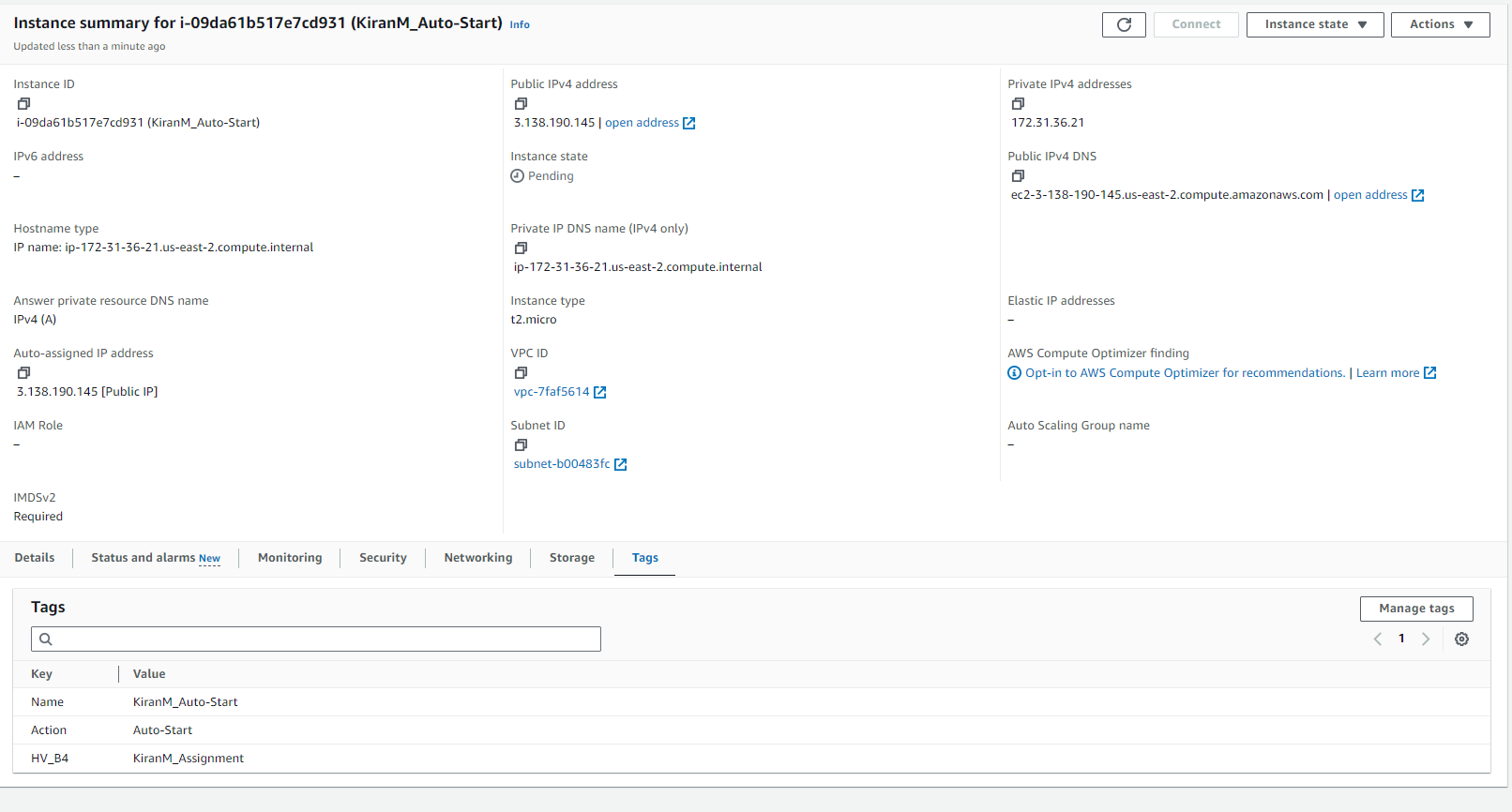
**Step 1: Create Auto-Stop instance**







**Step 2: Create another EC2 instance Auto-Start**



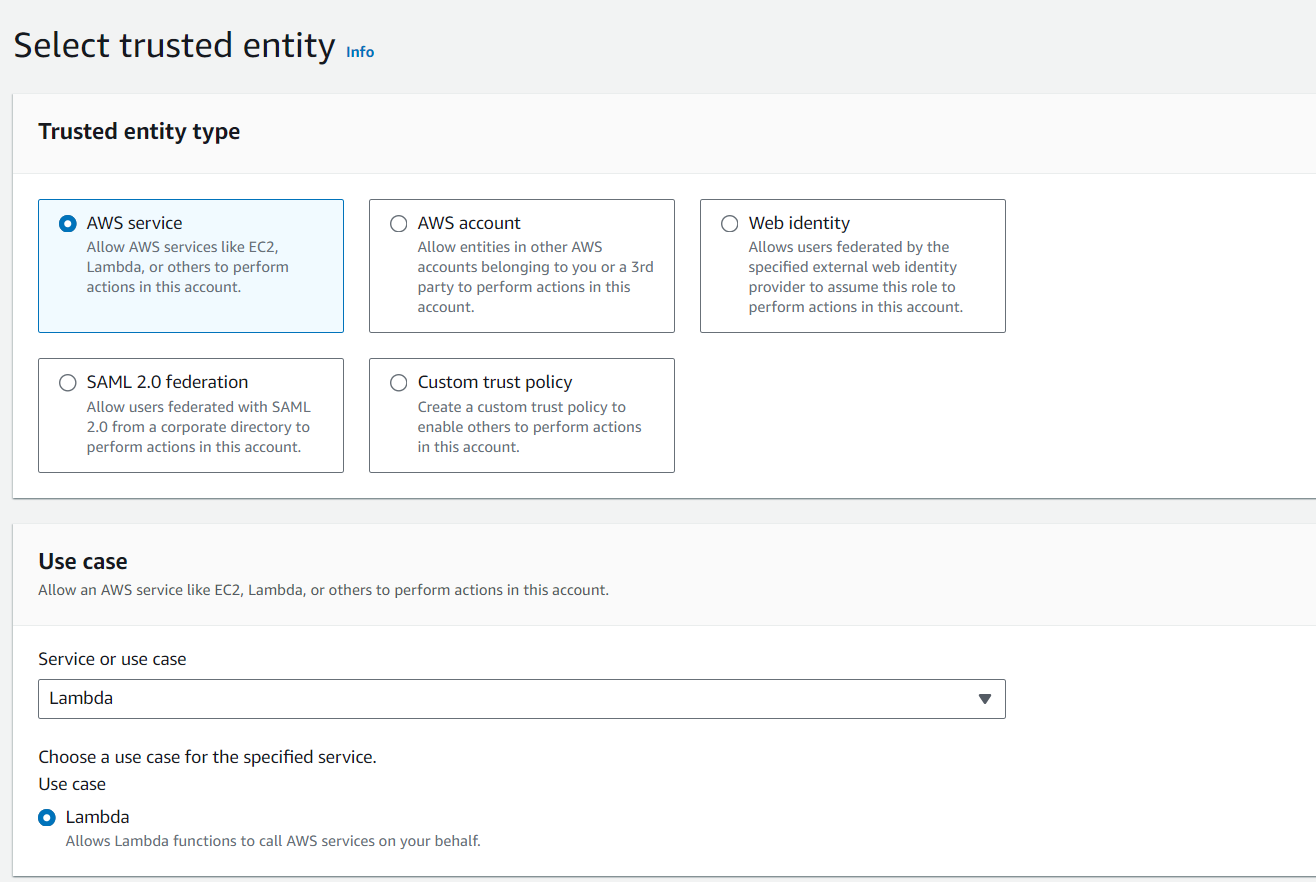
1. **Lambda Function Creation:**

**Step 1: Create IAM Role**

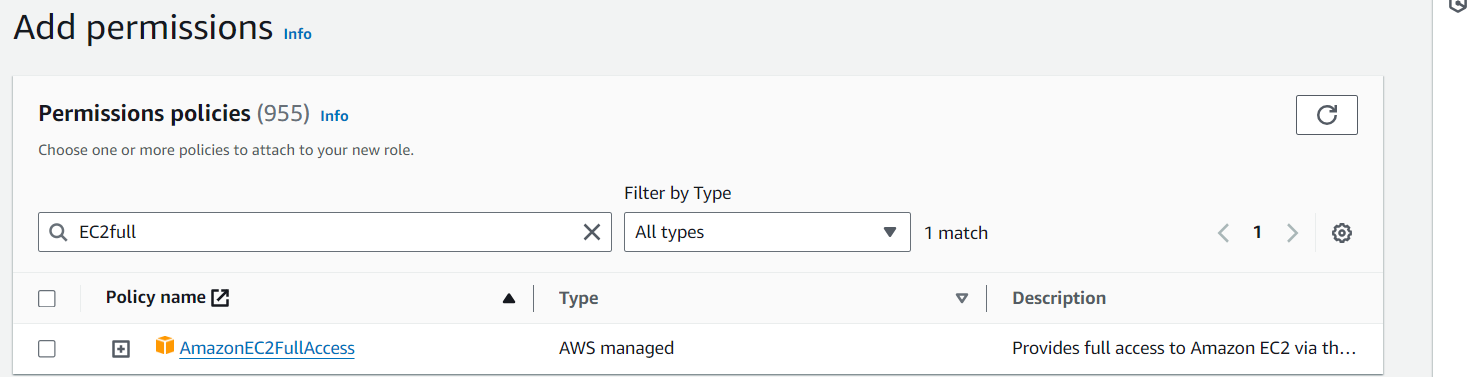
Go to IAM service

Create Role by going to navigation Roles -> Create Role

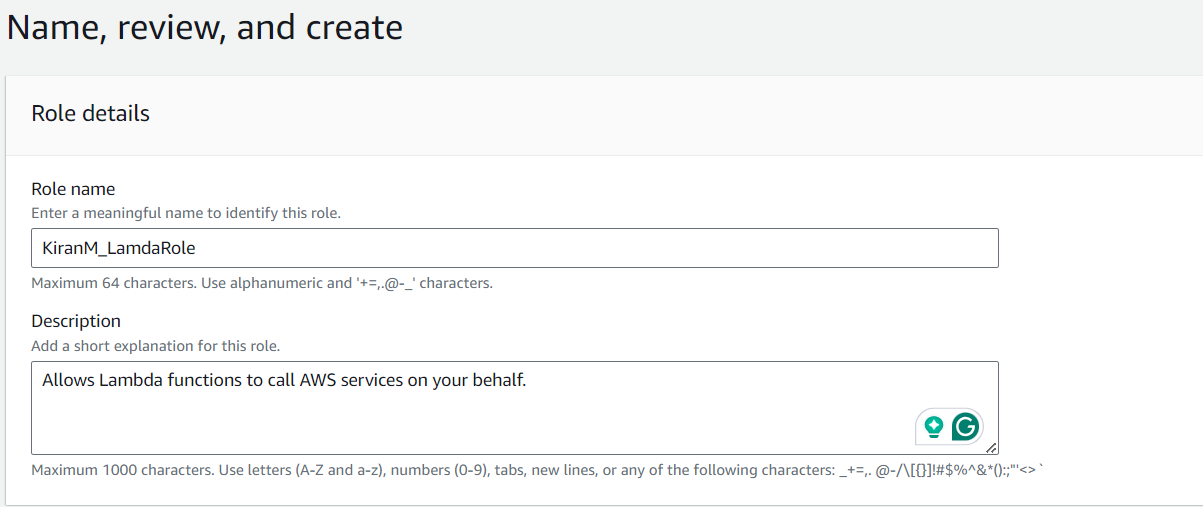
Select AWS service and Service as Lambda in Step 1 Select trusted entity



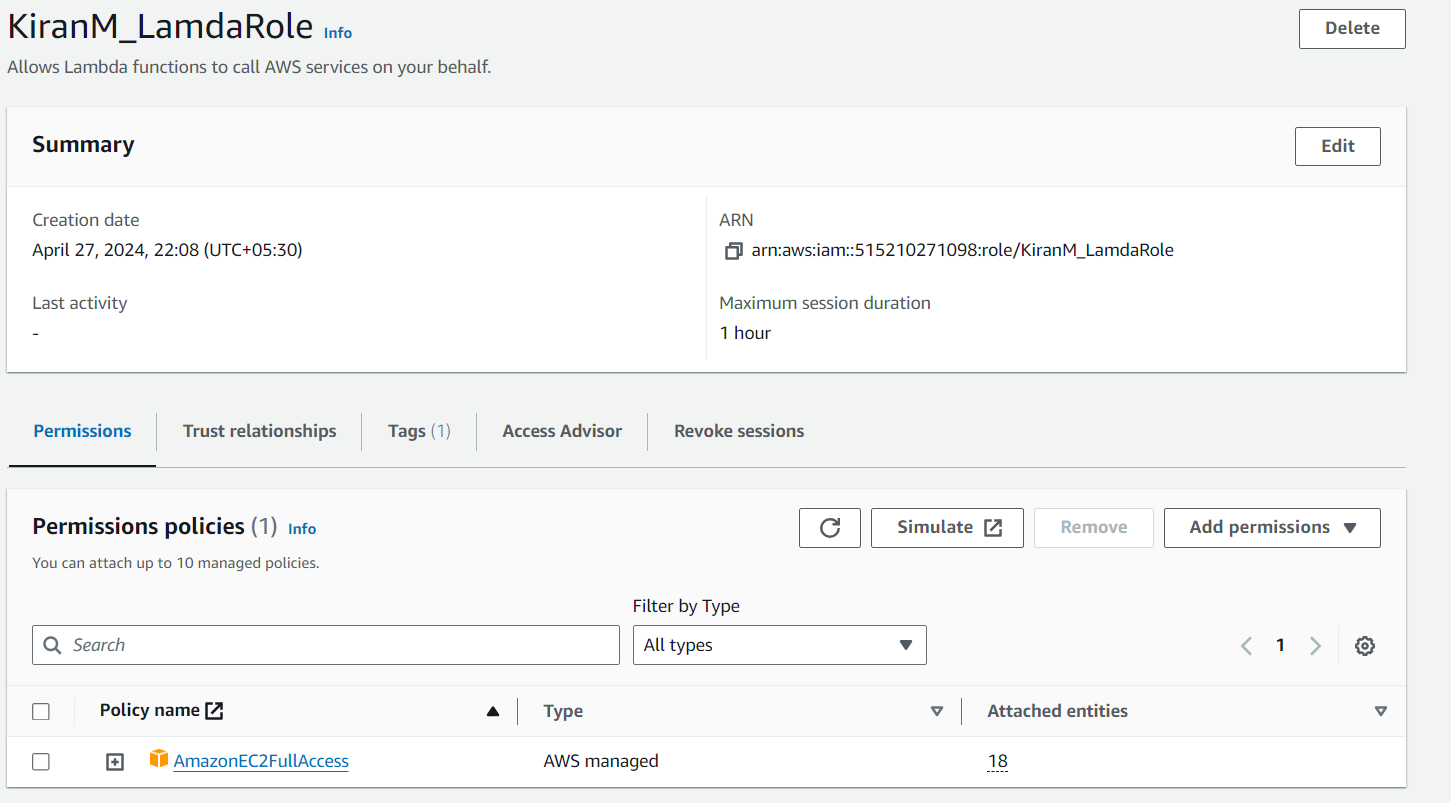
Next Step 2 Add permissions



Next Step 3 Name, review, and create

****

Click on Create role



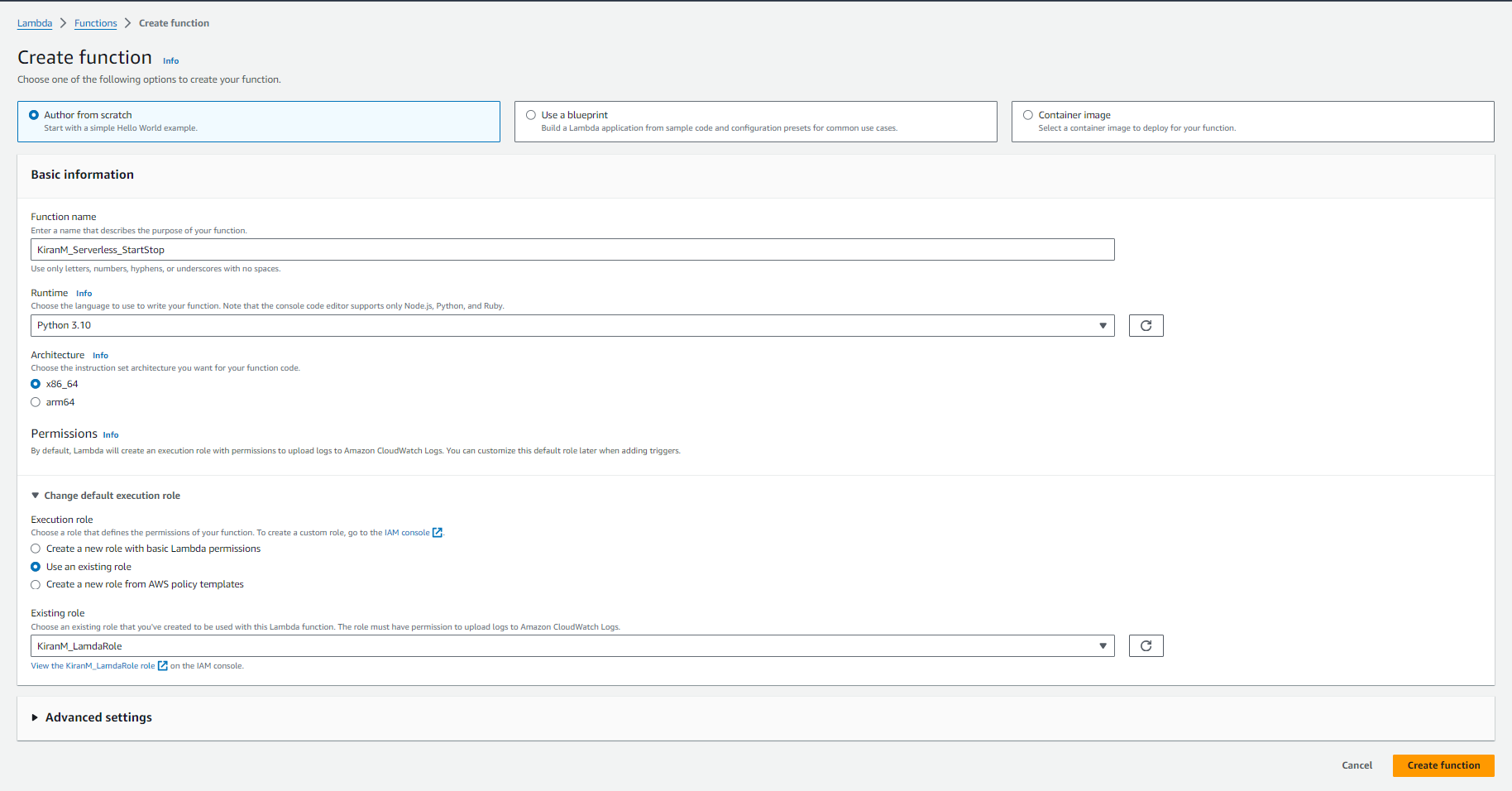
**Step 2: Go to Lambda Service and click on Create Function**

Select Auto from Scratch

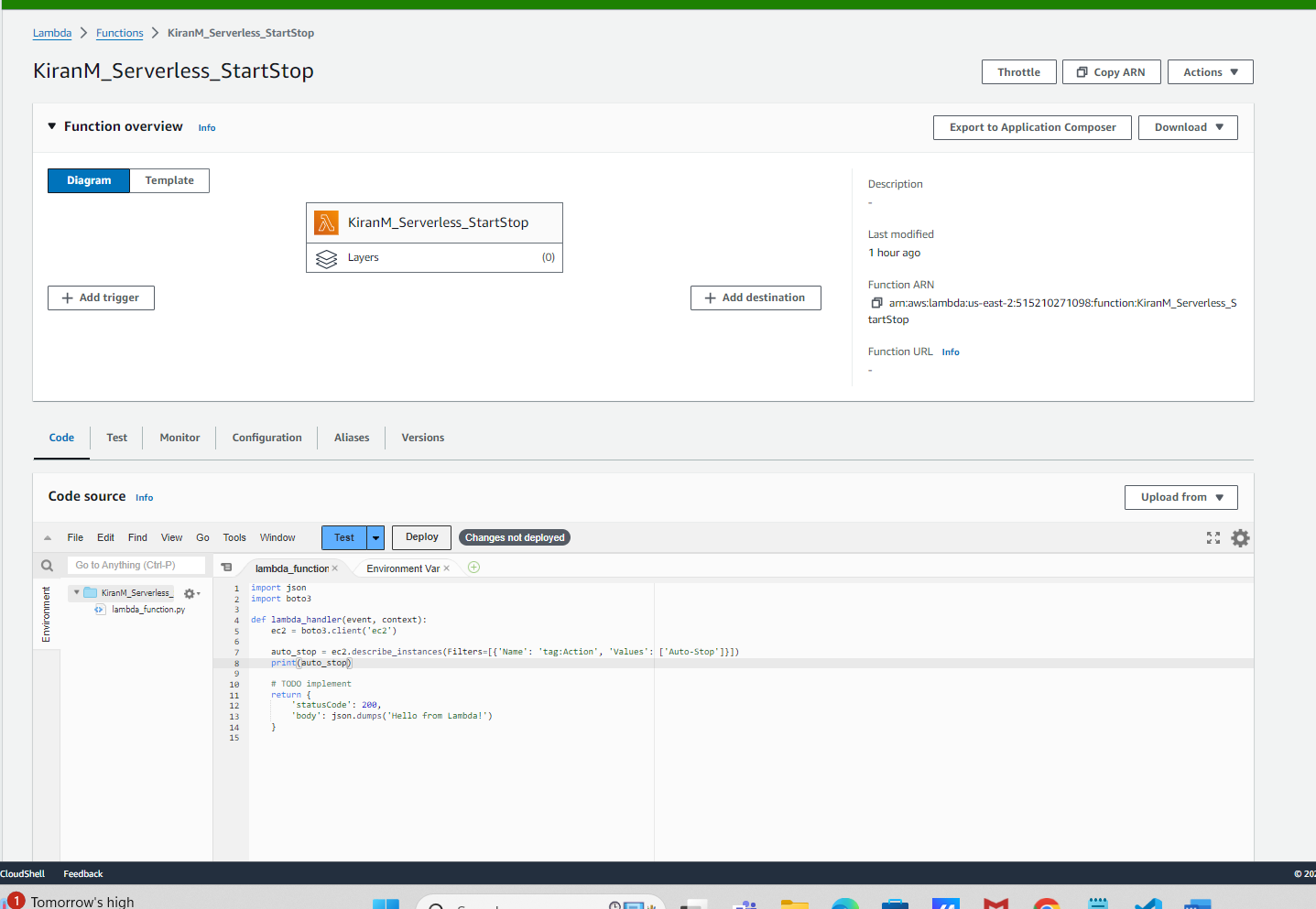
Function Name as KiranM\_Serverless\_StartStop

Runtime: Python 3.10

Execution Role: The role we created in above step KiranM\_LamdaRole



Click on Create Function



Add the code to start stop the instances based on the tags.

Code:

import json

import boto3

def lambda\_handler(event, context):

ec2 = boto3.client('ec2')

auto\_stop = ec2.describe\_instances(Filters=[{'Name': 'tag:Action', 'Values': ['Auto-Stop']}])

print(auto\_stop)

for reservation in auto\_stop['Reservations']:

for instance in reservation['Instances']:

instance\_id = instance['InstanceId']

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

print(f"Stopped instance {instance\_id}")

auto\_start = ec2.describe\_instances(Filters=[{'Name': 'tag:Action', 'Values': ['Auto-Start']}])

for reservation in auto\_start['Reservations']:

for instance in reservation['Instances']:

instance\_id = instance['InstanceId']

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

print(f"Started instance {instance\_id}")

## TODO implement

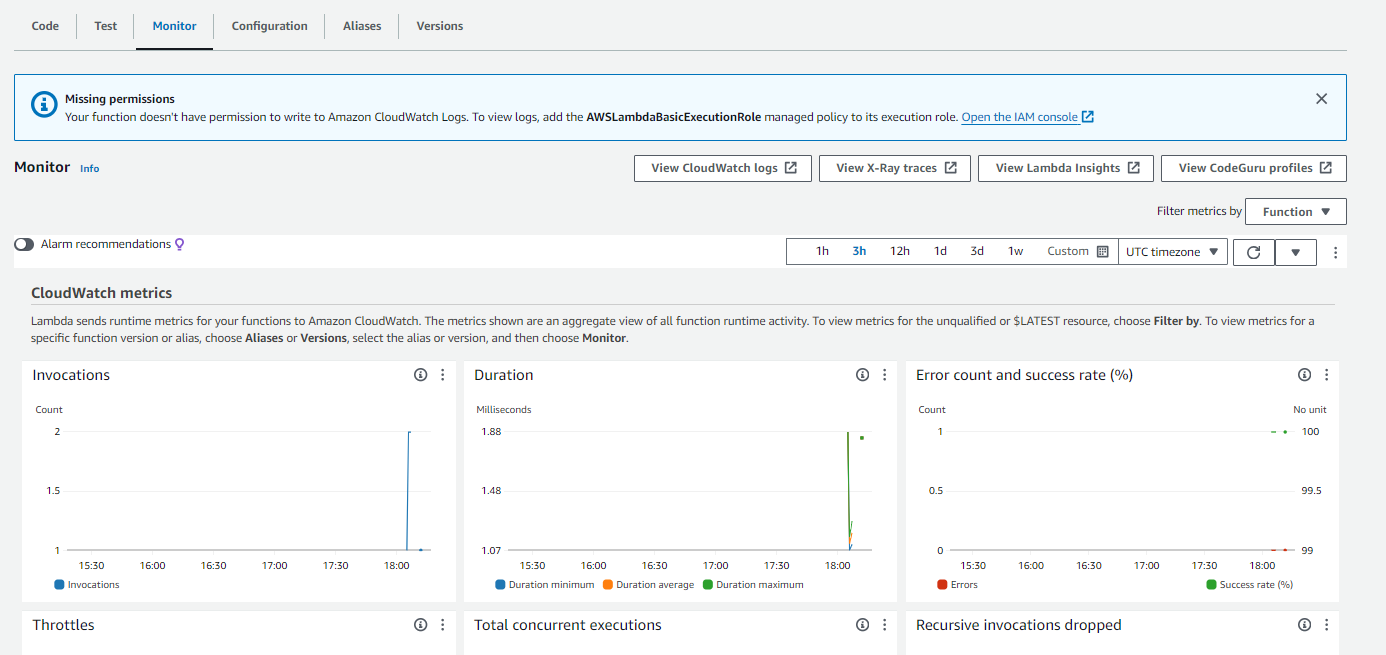
#return {

# 'statusCode': 200,

# 'body': json.dumps('Hello from Lambda!')

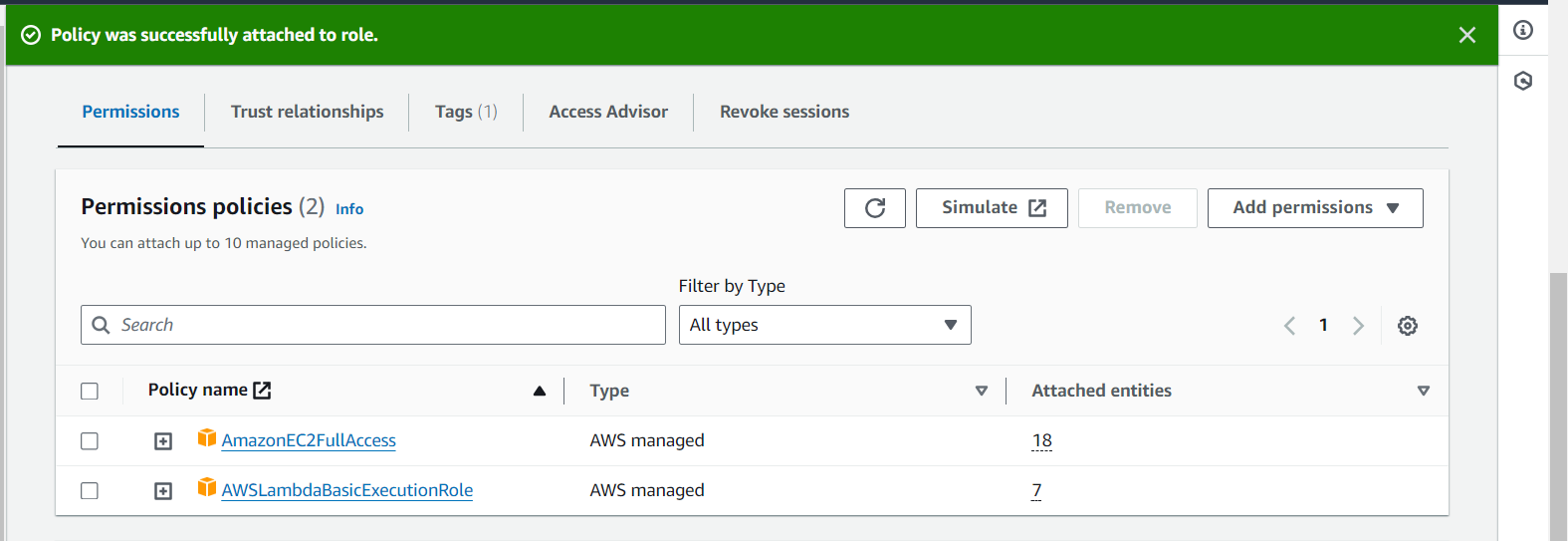
#}

Go to Monitor tab to check the logs in CloudWatch.



As we can see in the above image, it is saying Missing permission. We will add this permission.

We will go to IAM service again and add this policy to our IAM service.



Also to watch logs, go to Configuration-> Permissions.

Select Amazon CloudWatch logs in Resource summary.

