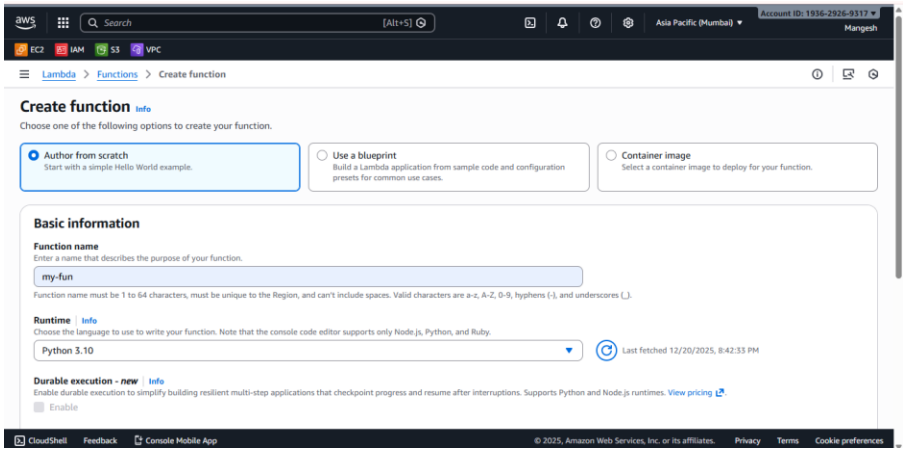
Auto Start Stop instance by using Lambda Function and Amazon EventBridge.

Steps:

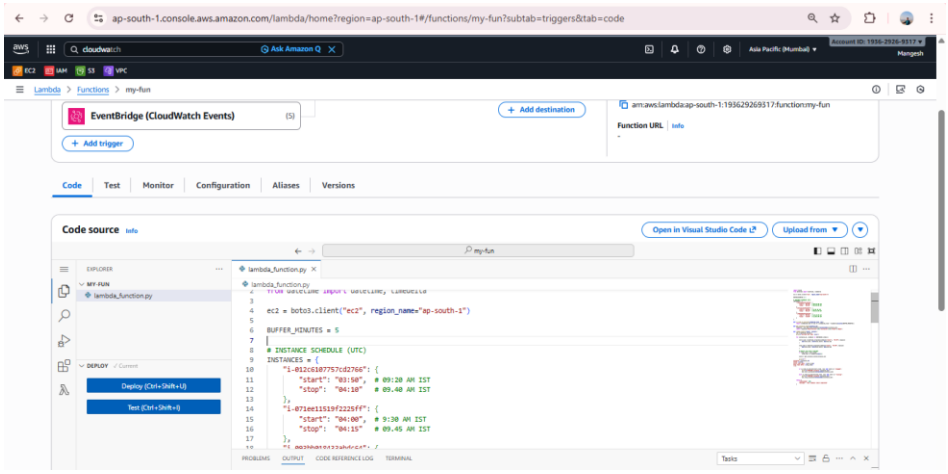
Create a Lambda Function. Select the language Python 3.10



Paste the code in code editor.

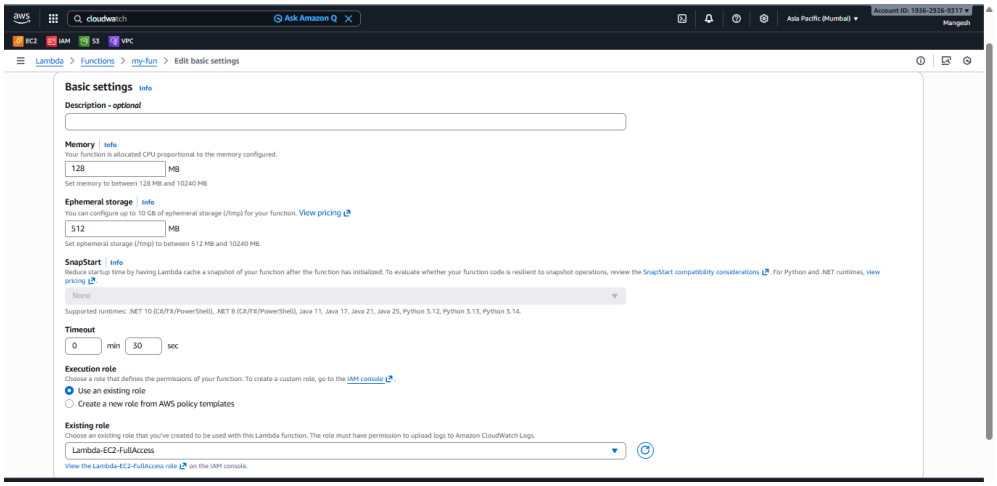
```python
import boto3
from datetime import datetime, timedelta

ec2 = boto3.client("ec2", region_name="ap-south-1")

BUFFER_MINUTES = 5
# INSTANCE SCHEDULE (UTC)
INSTANCES = {
    "i-012c6107757cd2766": {
        "start": "03:50",  # 09:20 AM IST
        "stop":  "04:10"   # 09.40 AM IST
    },
    "i-071ee11519f2225ff": {
        "start": "04:00",  # 9:30 AM IST
        "stop":  "04:15"   # 09.45 AM IST
    },
    "i-093bb018433abdc64": {
        "start": "04:40",  # 10:10 AM IST
        "stop":  "18:40"   # 12:05 AM IST
    }
}

def is_time_to_execute(scheduled_time, now):
    return scheduled_time <= now <= (scheduled_time + timedelta(minutes=BUFFER_MINUTES))

def get_instance_state(instance_id):
    response = ec2.describe_instances(InstanceIds=[instance_id])
    return response["Reservations"][0]["Instances"][0]["State"]["Name"]

def lambda_handler(event, context):
    now = datetime.utcnow()
    print(f"Current UTC Time: {now}")

    for instance_id, schedule in INSTANCES.items():

        start_time = datetime.strptime(schedule["start"], "%H:%M").replace(
            year=now.year, month=now.month, day=now.day
        )

        stop_time = datetime.strptime(schedule["stop"], "%H:%M").replace(
            year=now.year, month=now.month, day=now.day
        )
        # Handle stop after midnight
        if stop_time < start_time:
            stop_time += timedelta(days=1)

        state = get_instance_state(instance_id)

        print(f"""
Instance: {instance_id}
State: {state}
Start Time (UTC): {start_time}
Stop Time (UTC): {stop_time}
""")

        if is_time_to_execute(start_time, now) and state == "stopped":
            print(f"STARTING instance {instance_id}")
            ec2.start_instances(InstanceIds=[instance_id])

        elif is_time_to_execute(stop_time, now) and state == "running":
            print(f"STOPPING instance {instance_id}")
            ec2.stop_instances(InstanceIds=[instance_id])

    return {
        "statusCode": 200,
        "message": "EC2 schedule check completed"
    }
```

Deploy the code.

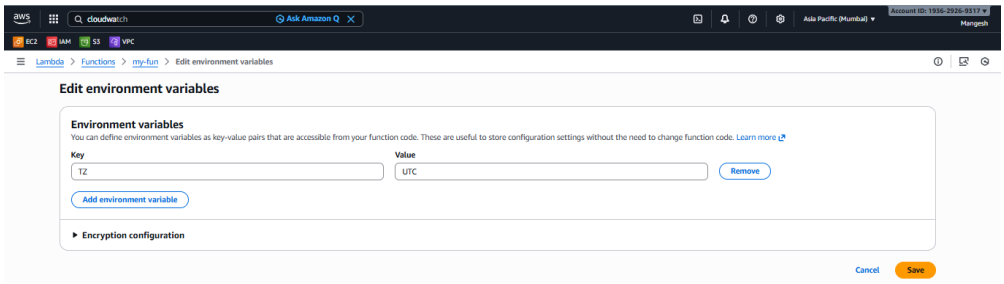


Create a new test event.

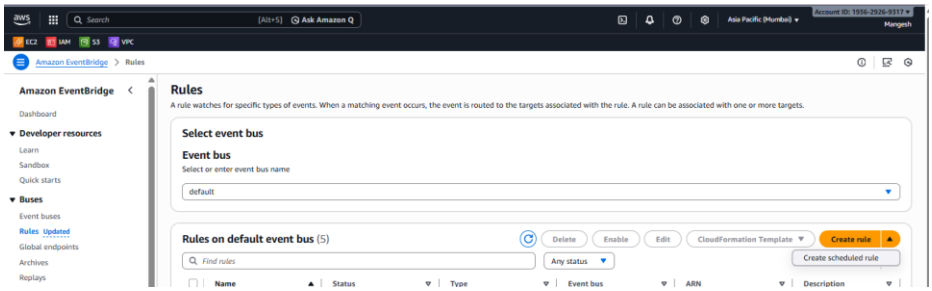
Go into Configuration → General Configuration → Edit → Time-Out → increase upto 30 second.



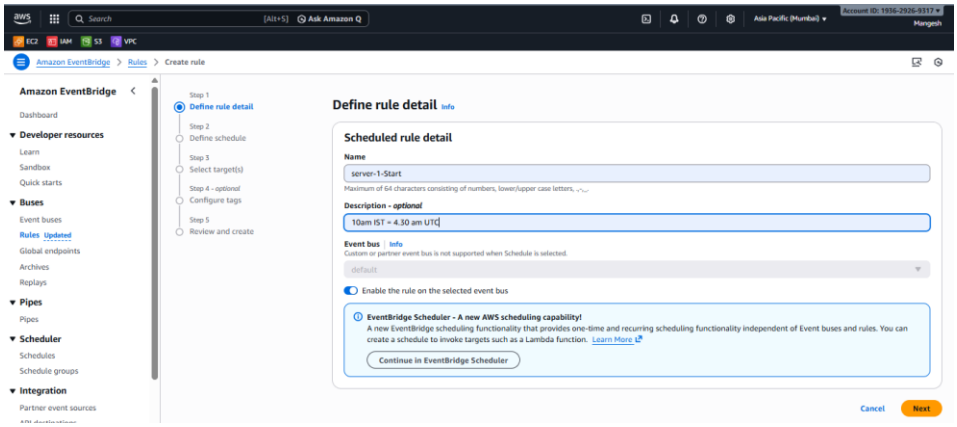
Go into Configuration → Environment Variables → Edit → Key = TZ (Time Zone) , Pair = UTC



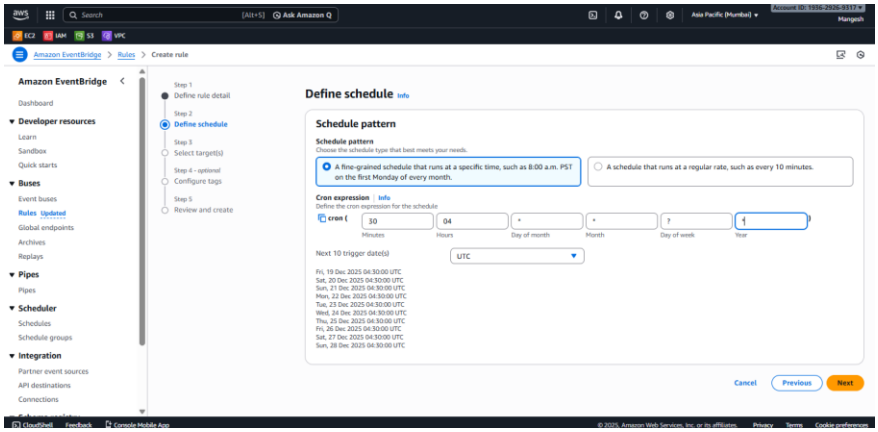
Go into another service Amazon EventBridge → Go into Rules → Create Scheduled Rule



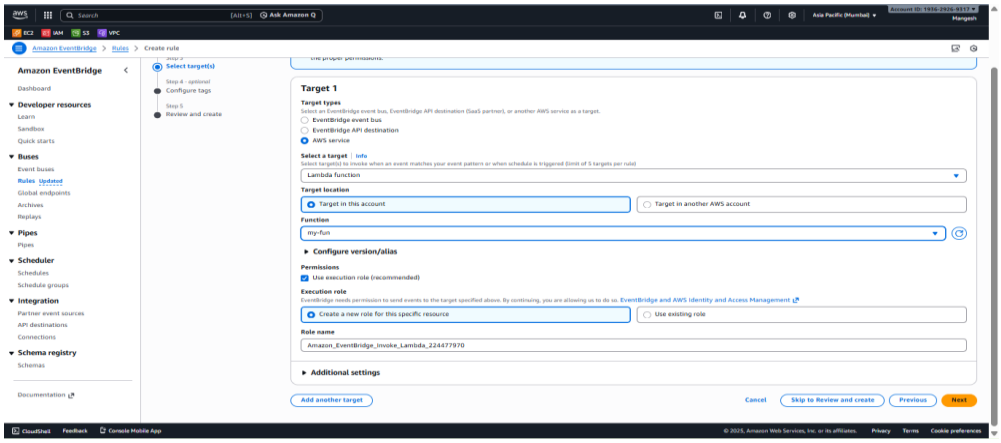
Define a rule → Put the details in step-1



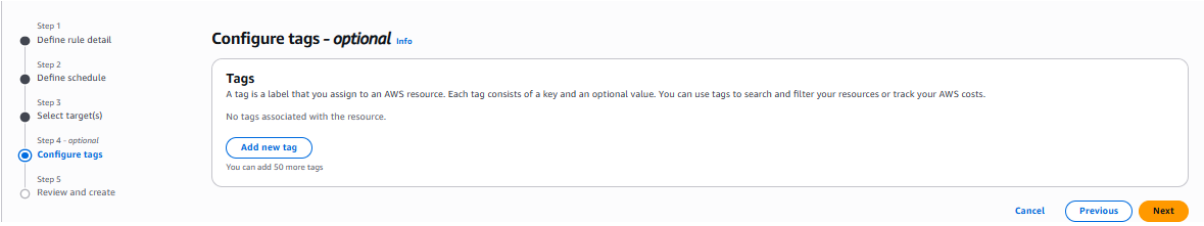
Write a crontab expression in next step that trigger an lambda function on Specific time in UTC.



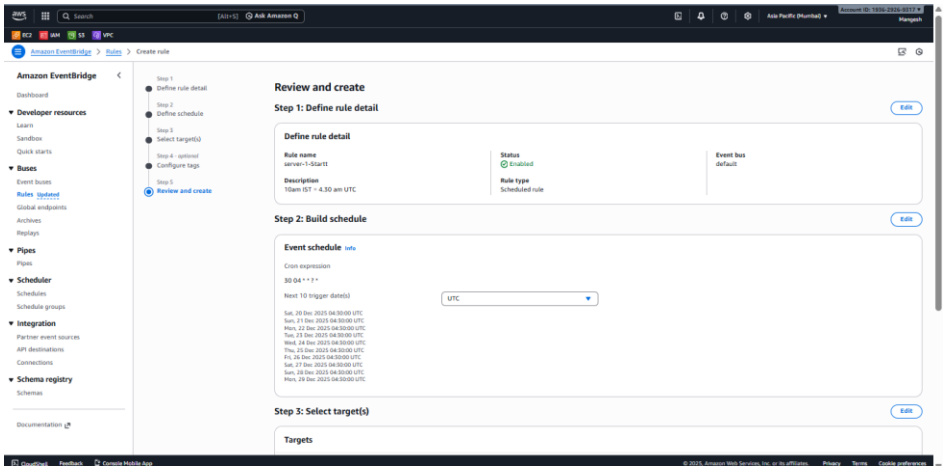
In Step 3. Select Target. In the execution role Select → Create new role for this specific resource.



In step-4 . Tags

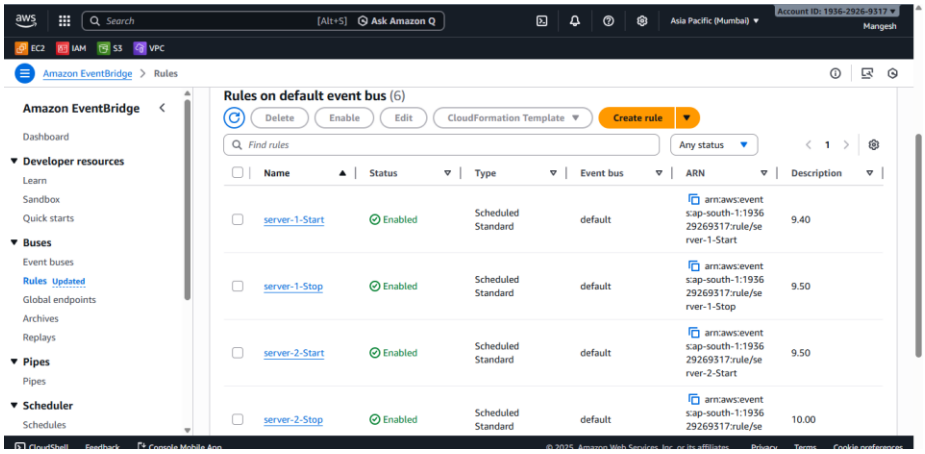


In step-5. Review and create.

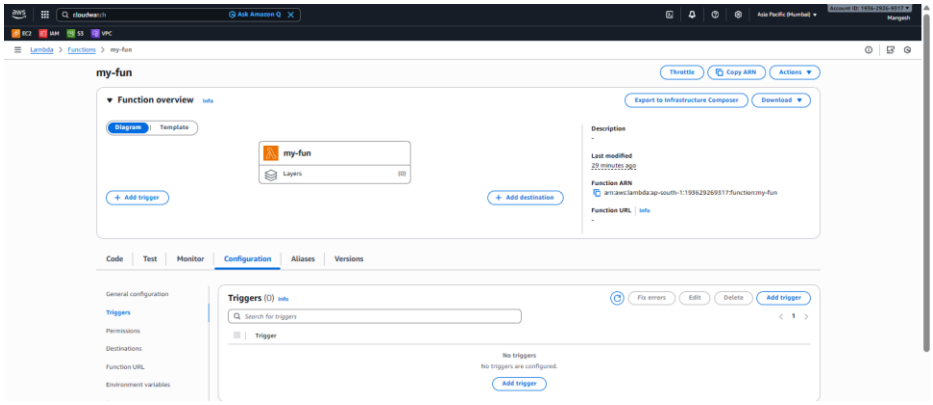


Rule created.

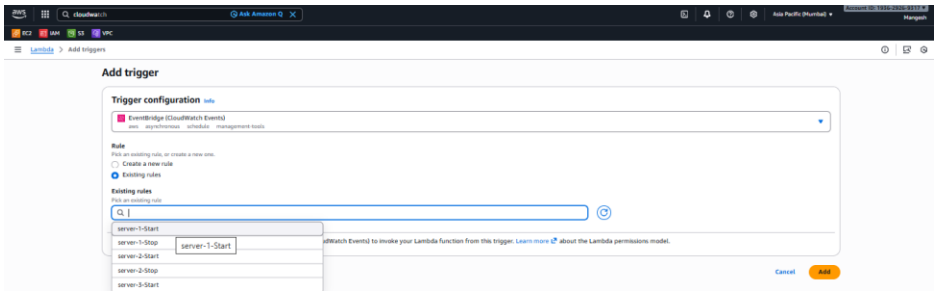
Do same steps for creating 6 Rules in EventBridge.



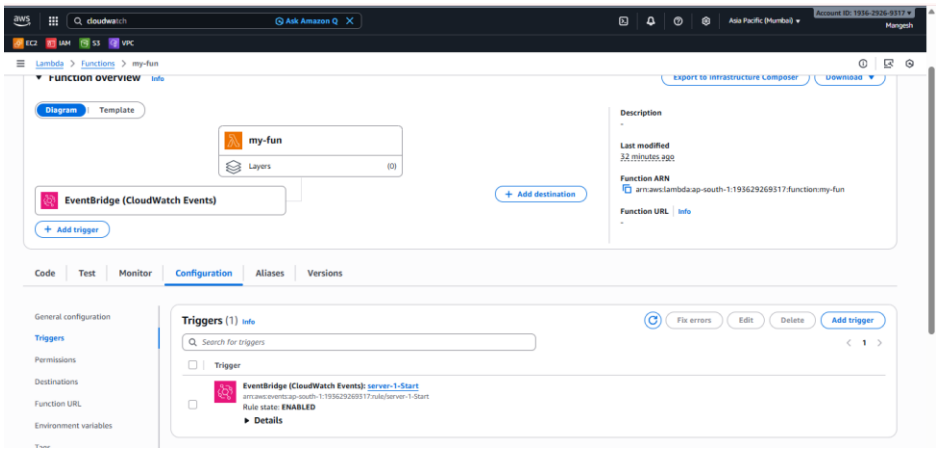
Go into Configuration → Trigger →Select EventBridge (CloudWatch Events) → Add Rules to trigger the functions.



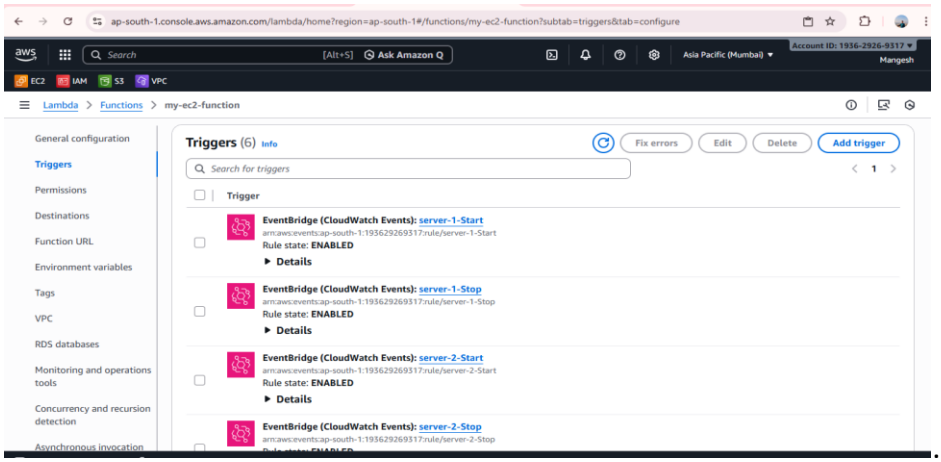
Add trigger here . Select the eventbridge rule here for triggering and add.



Trigger added.



Add next trigger do same process. Here we add 5 trigger means the rules we created in Amazon EventBridge service.



Here We done all the process.

Lets check.