**Script 1:**

**1. Retrieve a list of instances running in a specific AWS region.**

**2. Filter the instances based on their names.**

**3. Stop the matching instances.**

if [ -z "i-03bd1ddc45047827e" ]; then

-z→This is a test option that checks if the length of the string

if [ $? -eq 0 ]; then

$?-->This variable holds the exit status of the last command that was executed.

-eq 0→his checks if the value of $? is equal to 0

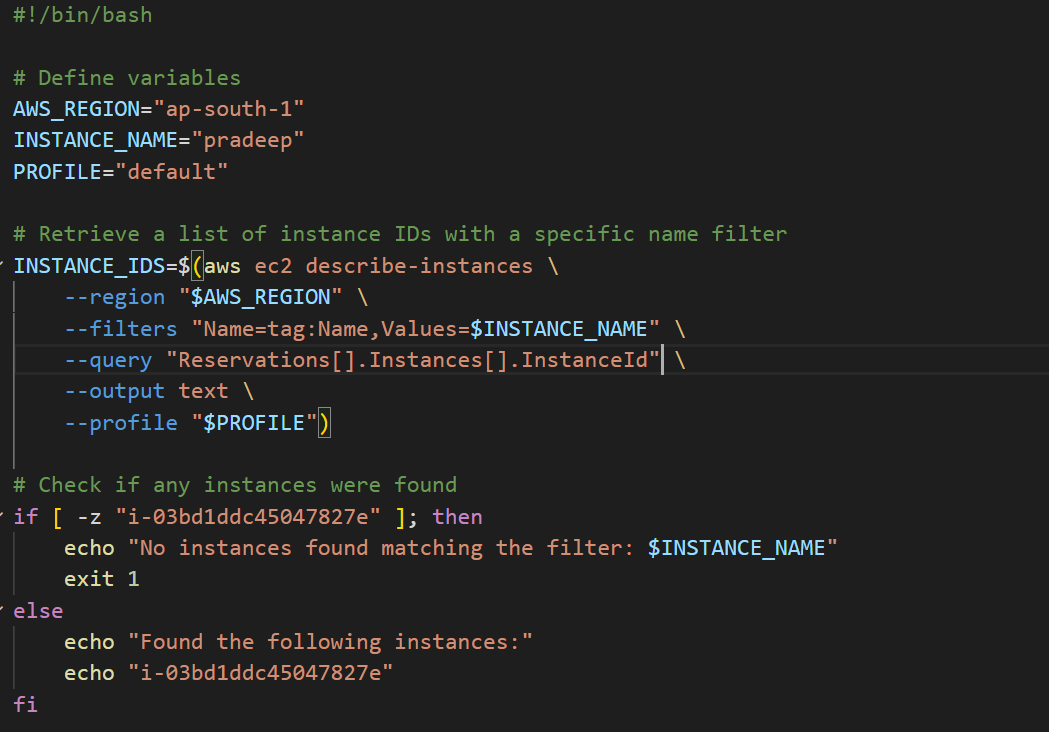
--query "Reservations[].Instances[].InstanceId"

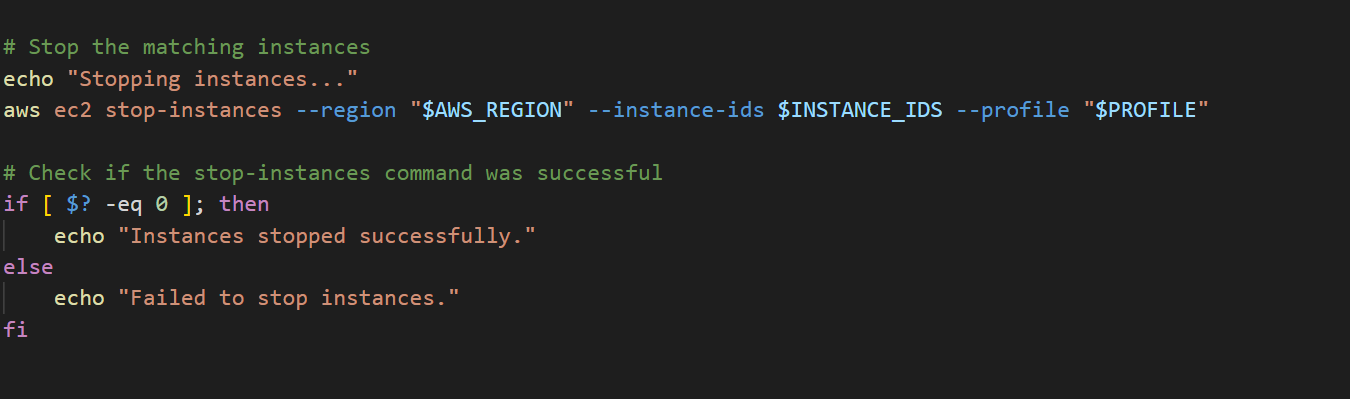
**--query**→this is used to filter and format the output

**Reservations[]**-->it typically corresponds to one or more instances.

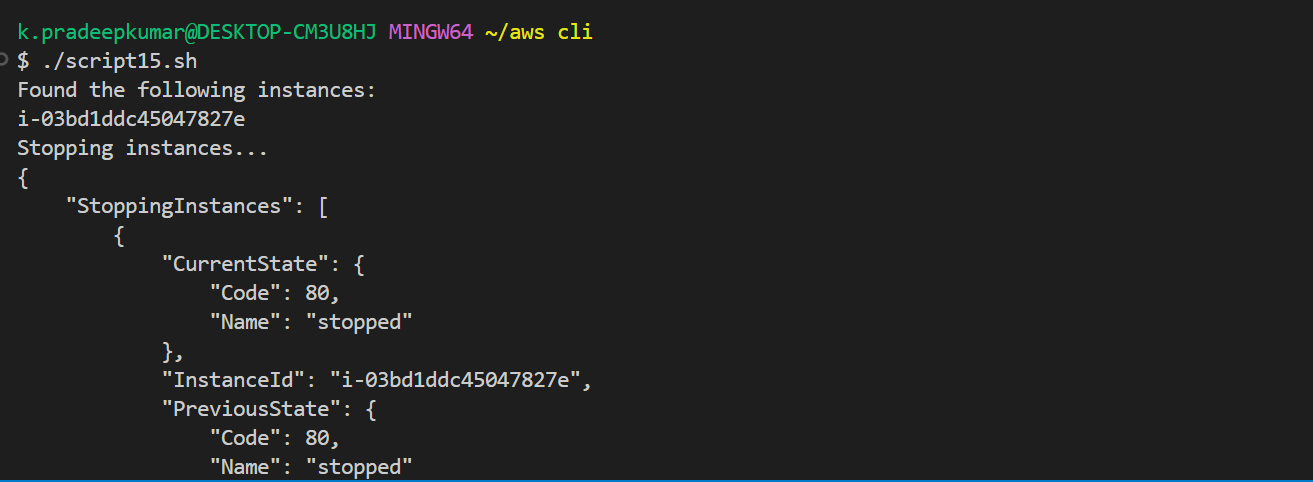
**Instances[]**-->which contains the details of the EC2 instances

**InstanceId**→This specifies that you want to retrieve the InstanceId field from each element in the Instances





**output:**



**Script 2:**

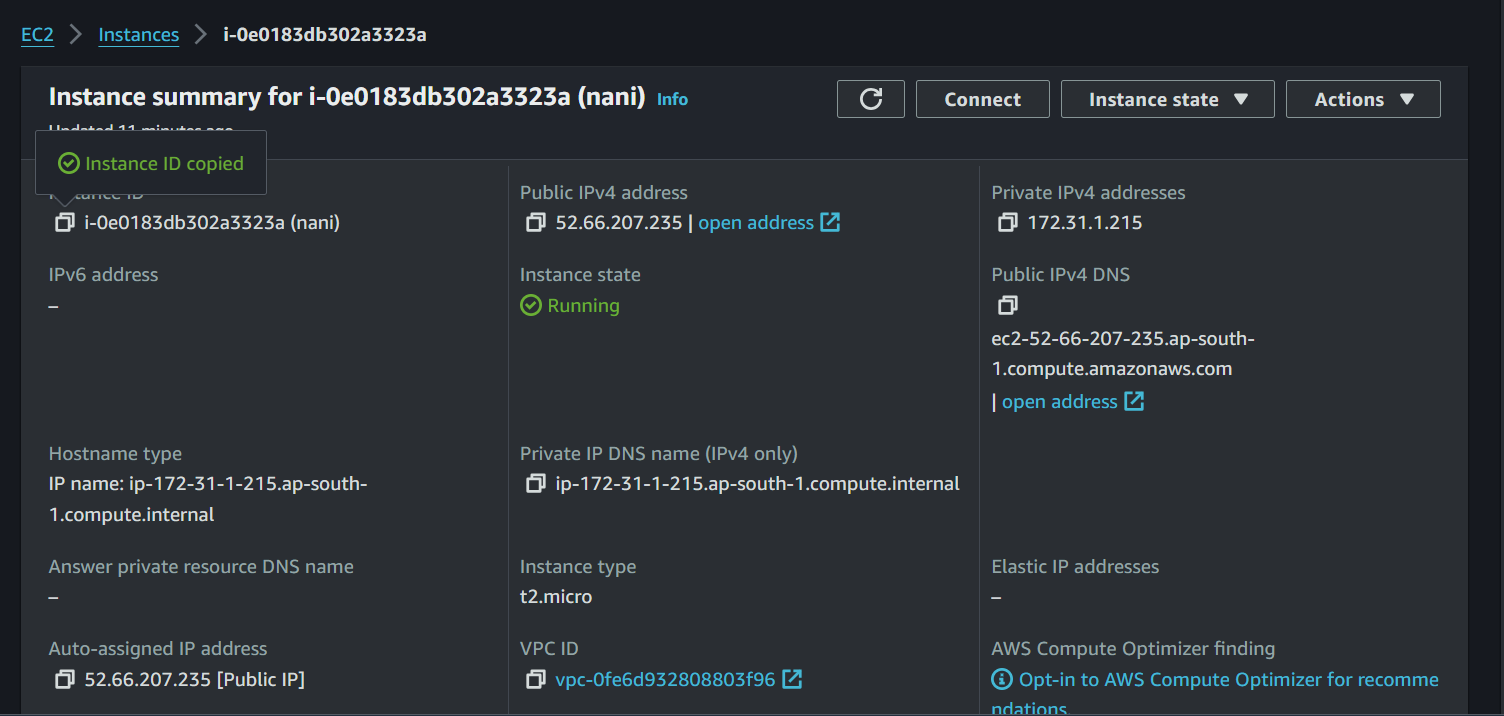
**1. Get a list of all instances.**

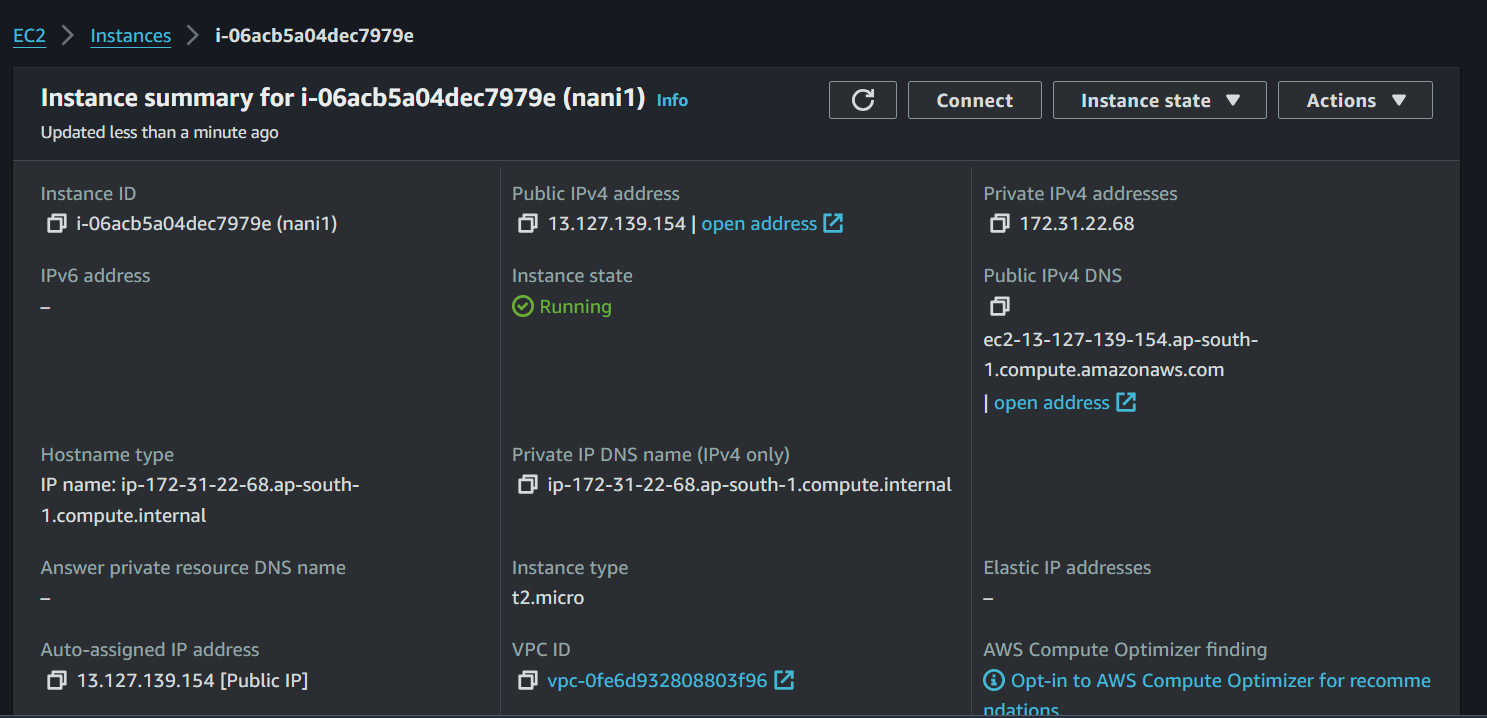
**2. Stop all running instances at specified times, for example:**

**- Start instances at 10 AM.**

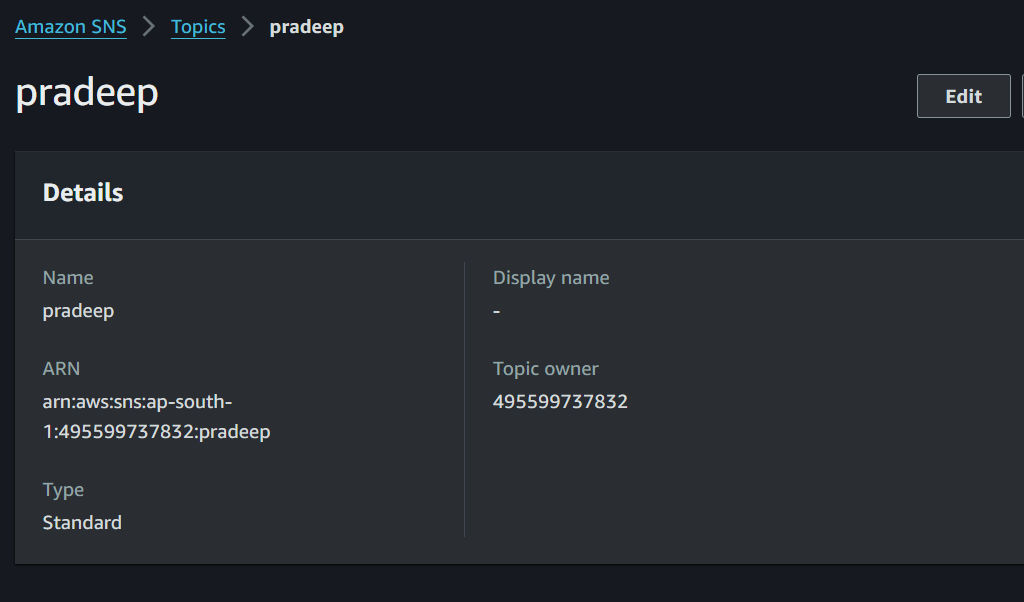
**- Stop instances at 7 PM.**

**1)Created two instances:**

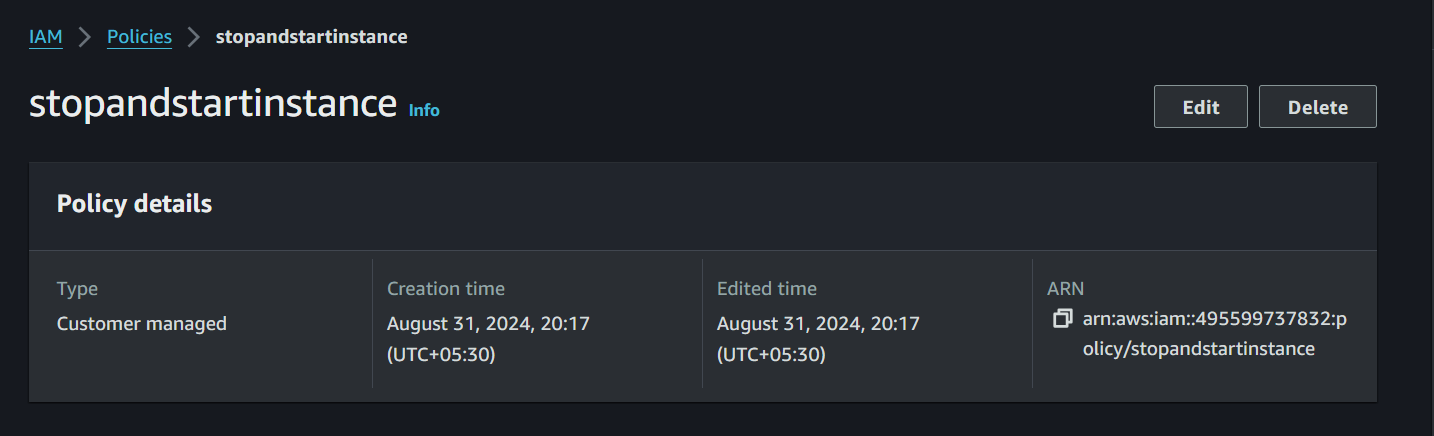
****

****

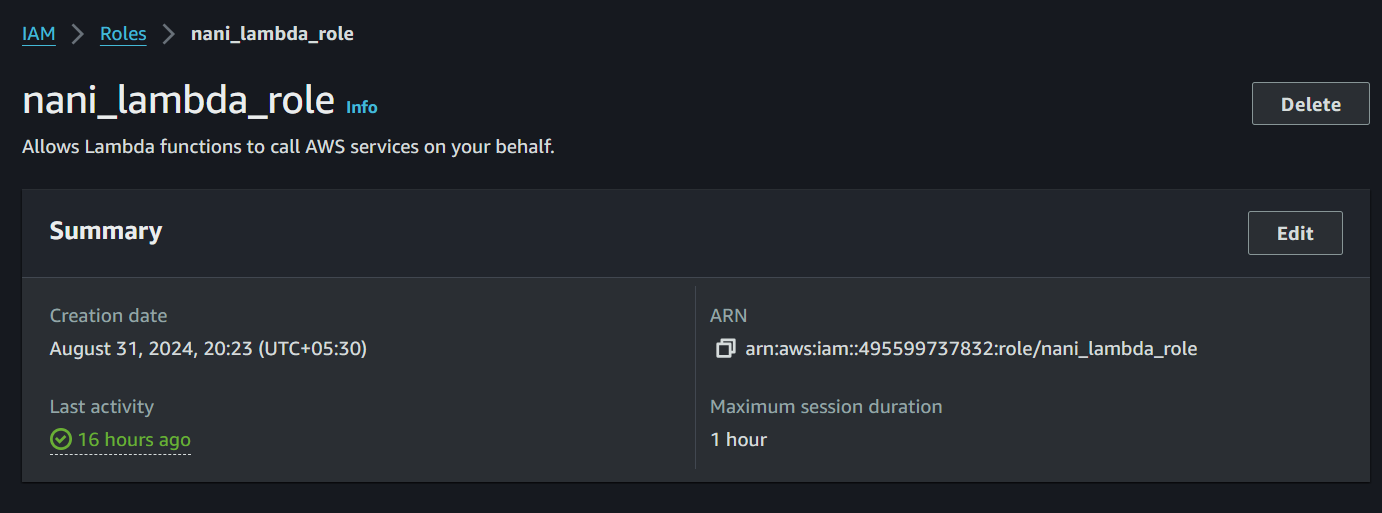
**2)Created an SNS topic:**

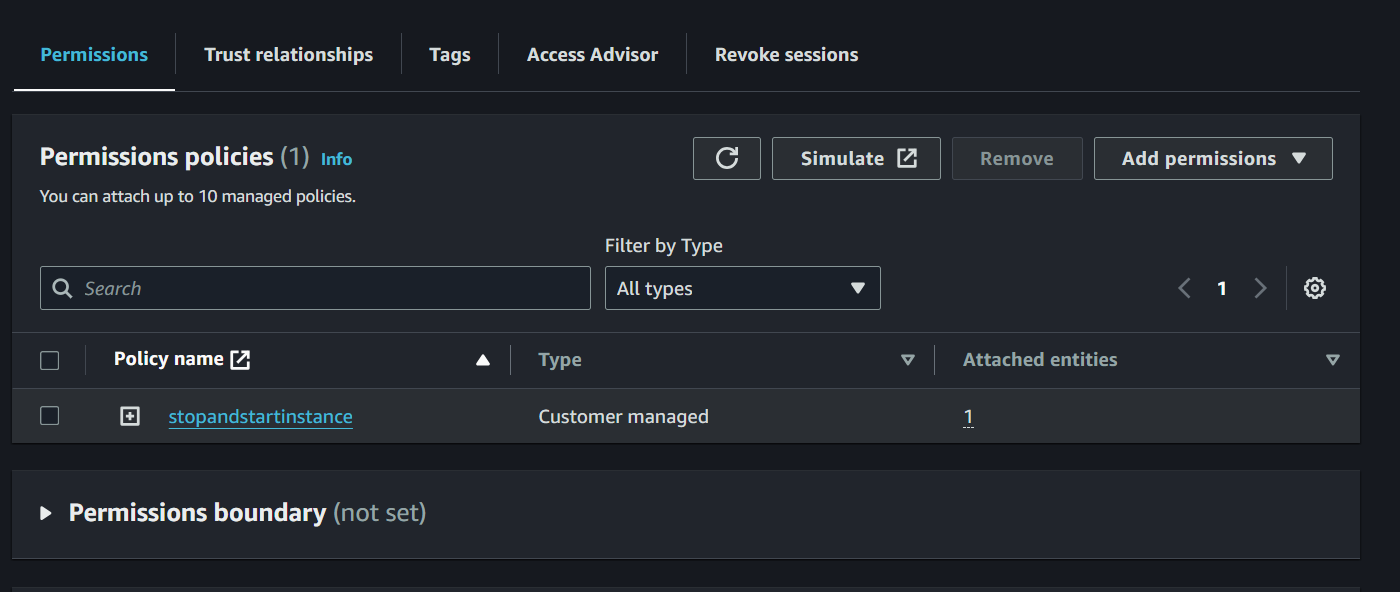
****

**3)In IAM user created an stopandstartinstance policy**

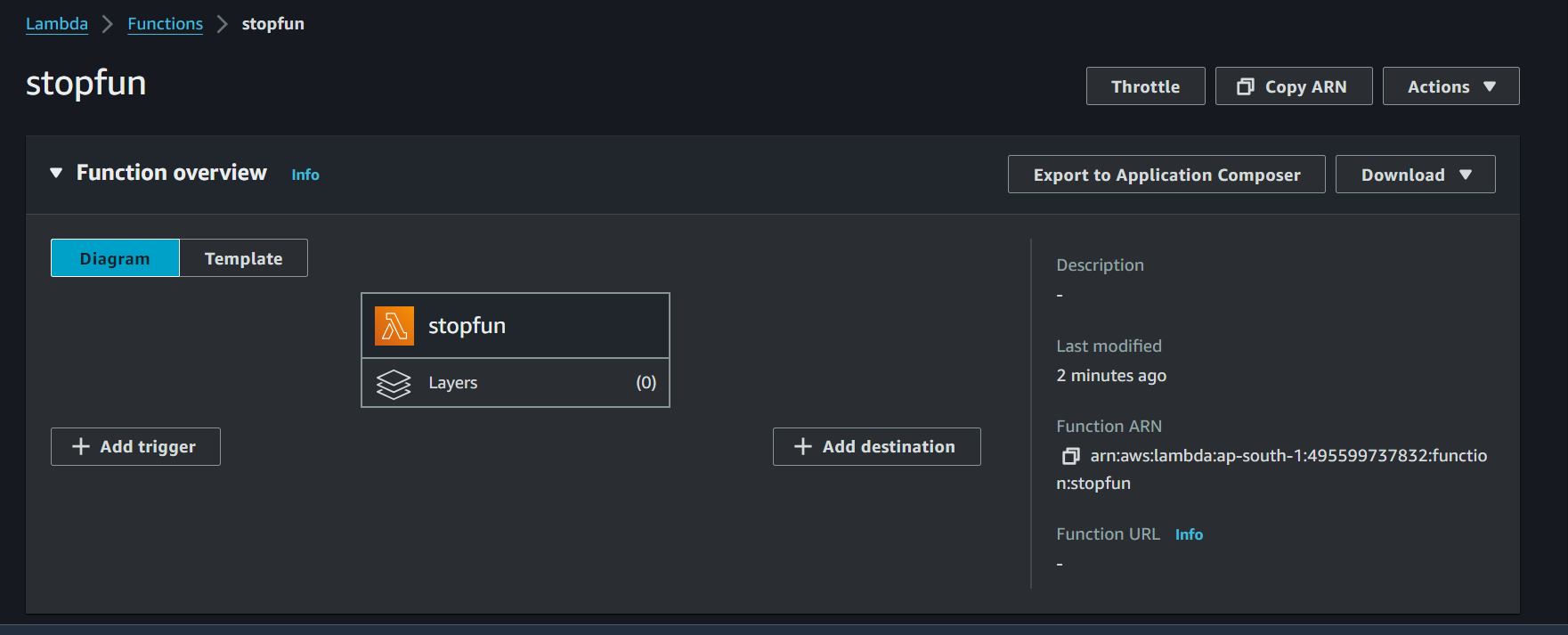
****

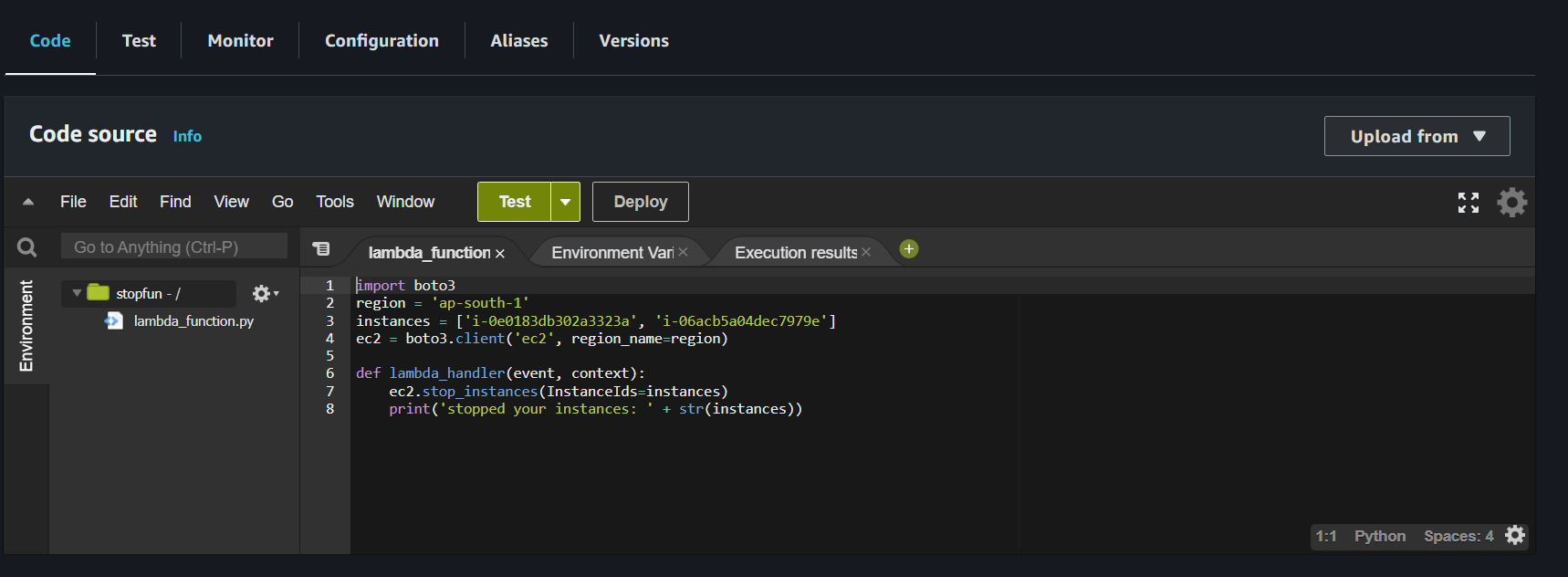
**4)created an lambda role in IAM user and attach policy to the lambda role**

****

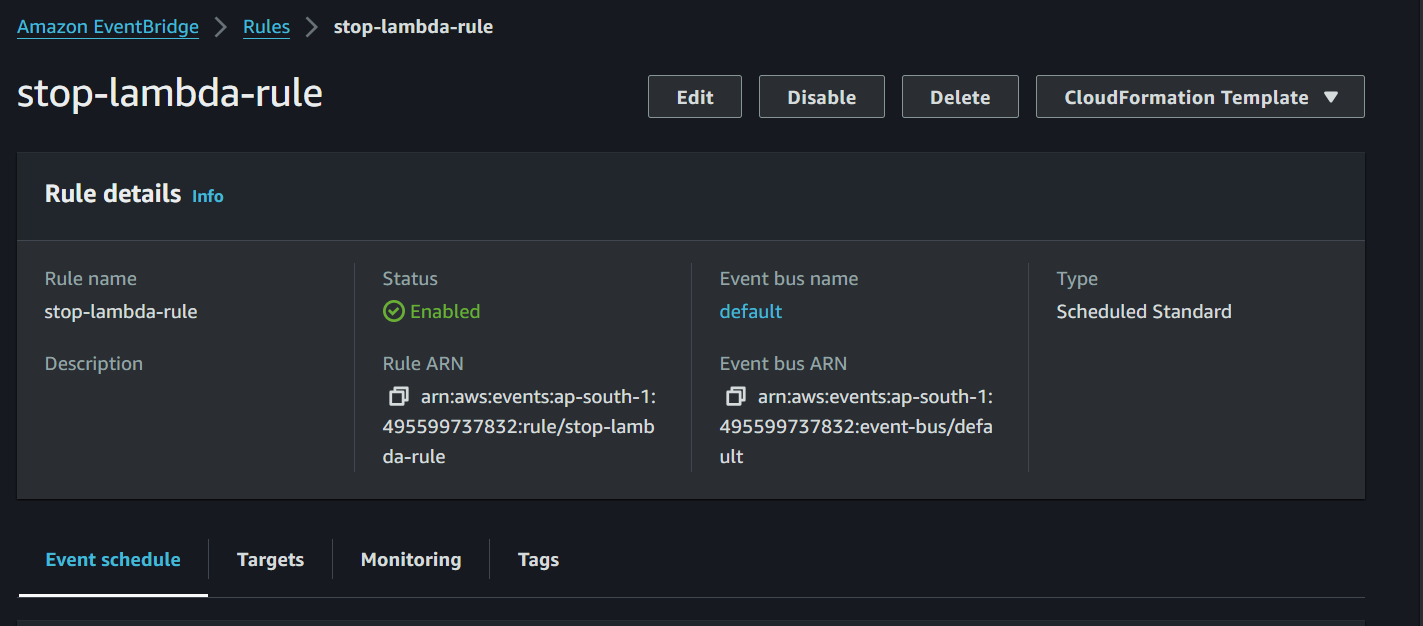
****

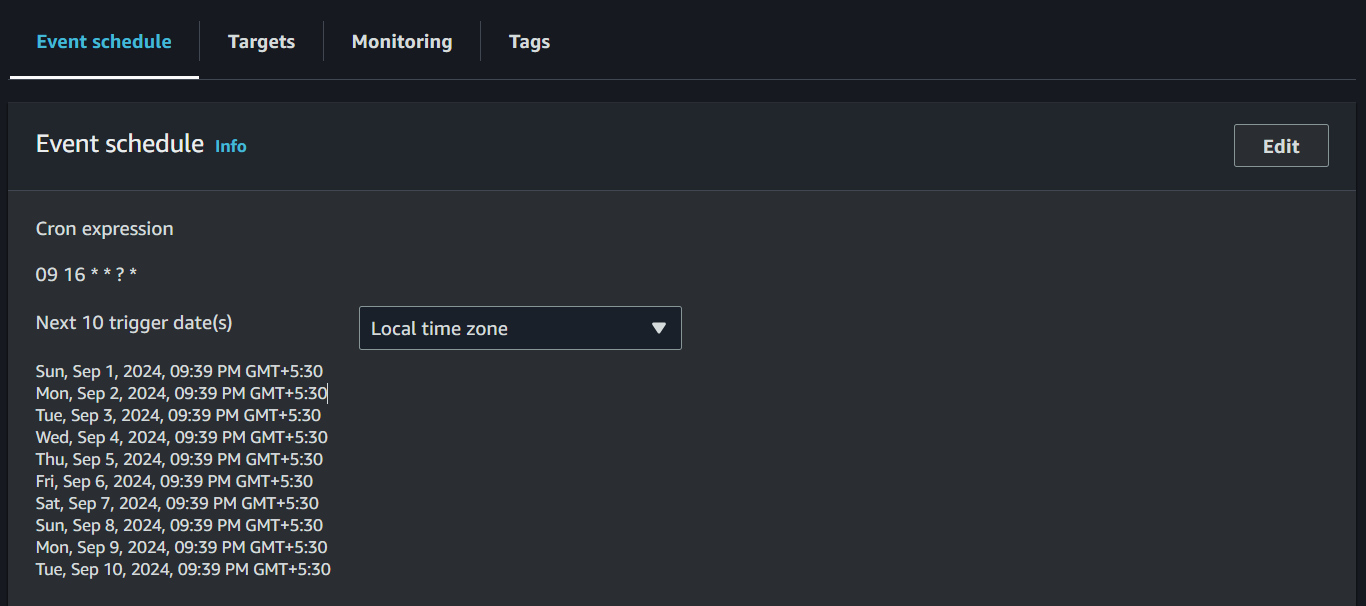
**5)Create the stop function in lambda and deploy the code in the stop fun**



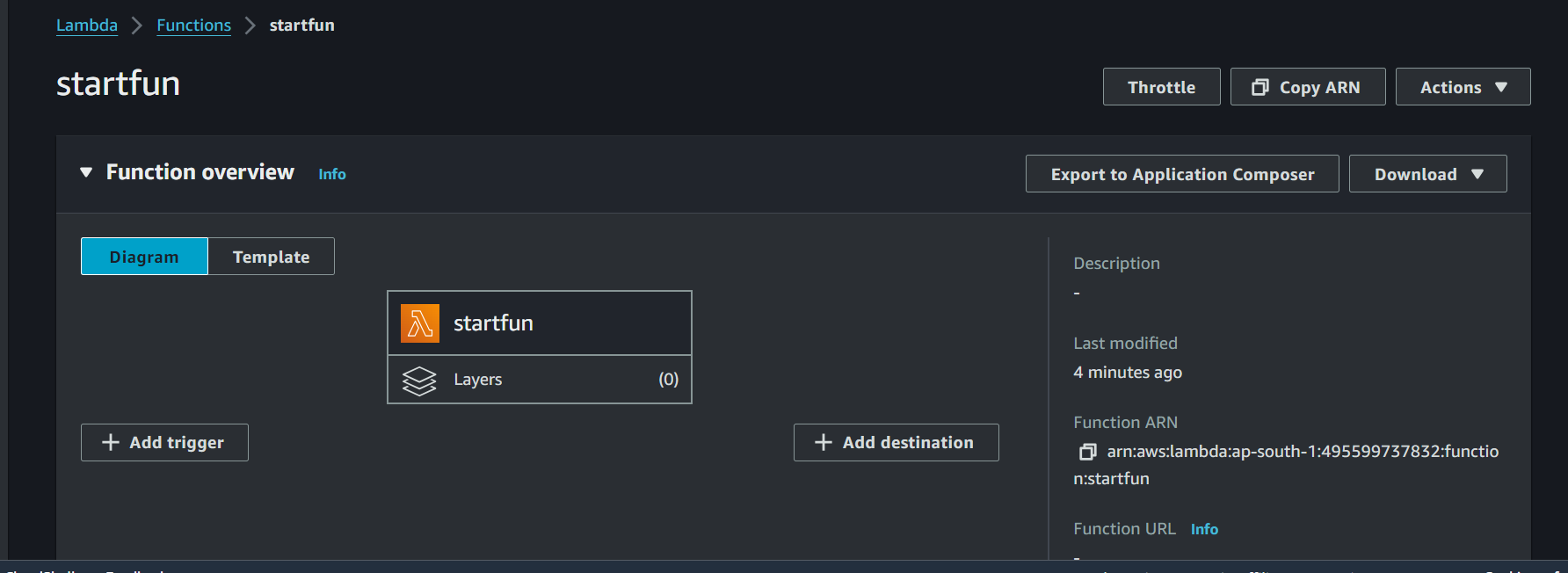


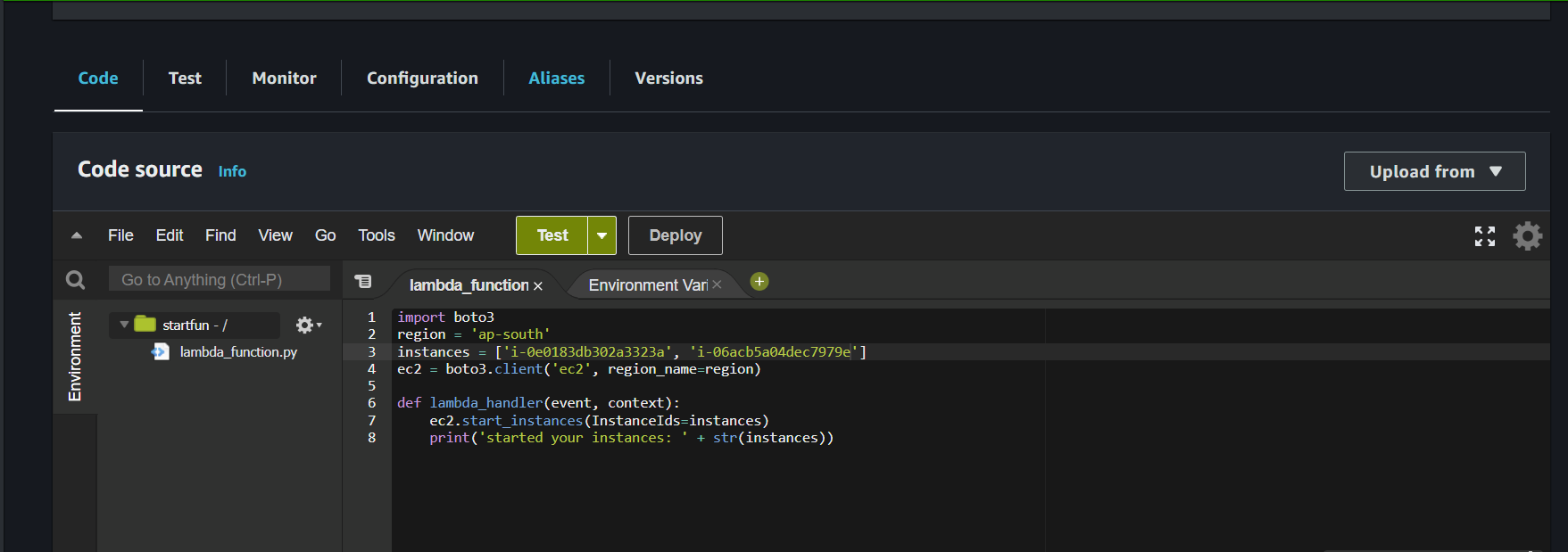
**6)Create a stop rule in cloudwatch and schedule to stop the instance using cron expression**



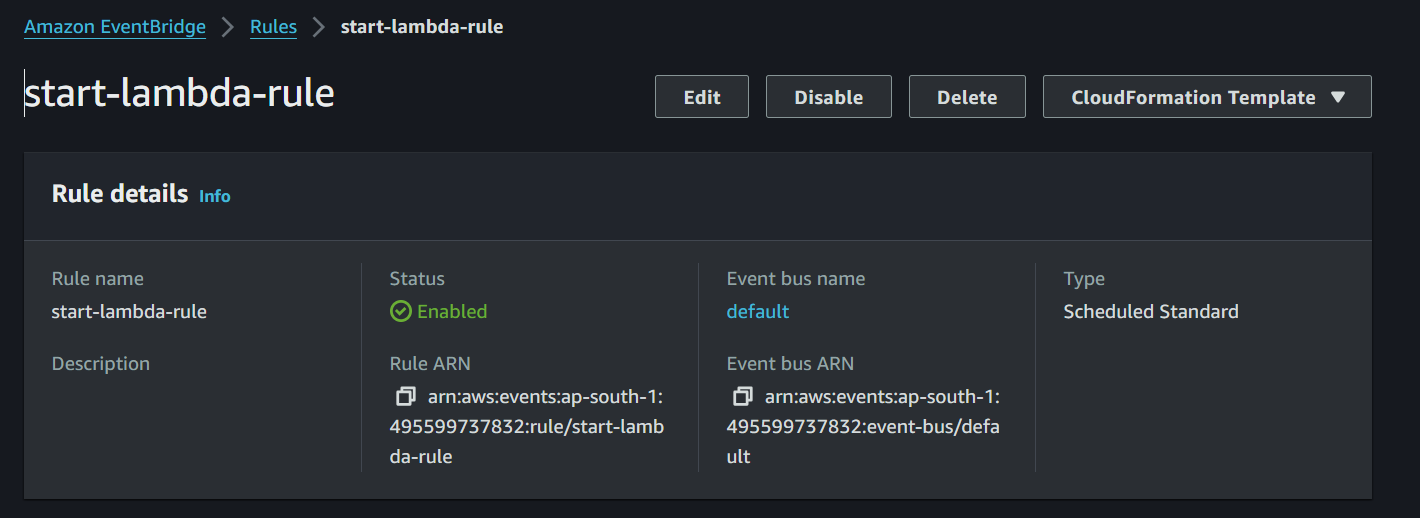


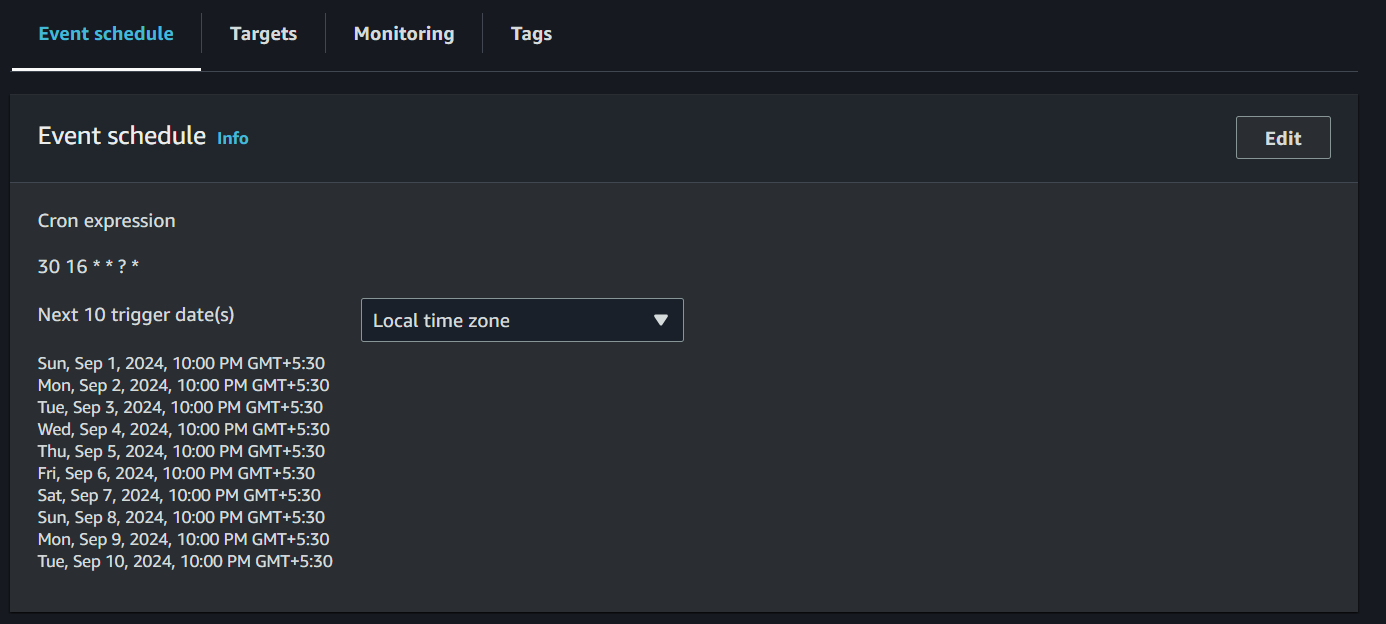
**7)Create the start function in lambda and deploy the code in the start fun**





**6)Create a start rule in cloudwatch and schedule to start the instance using cron expression**

****

****

**Script-2 using cli**

**Script 2:**

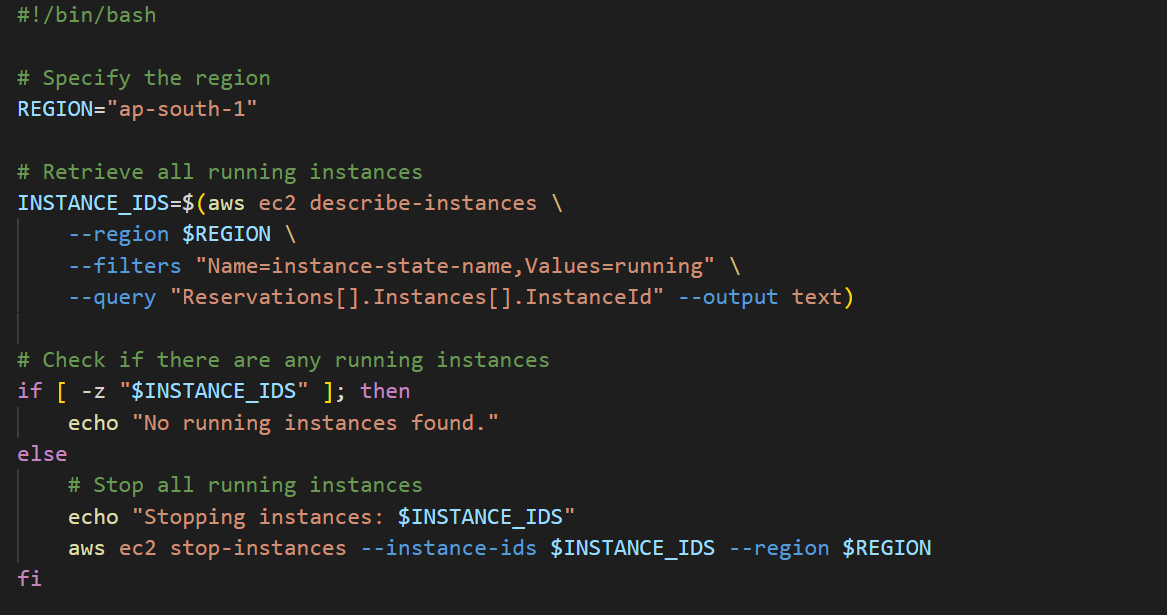
**1. Get a list of all instances.**

**2. Stop all running instances at specified times, for example:**

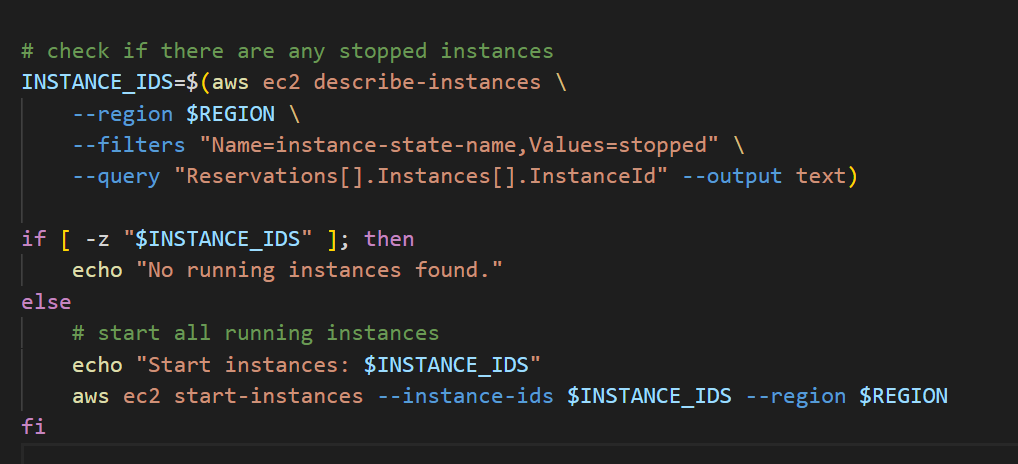
**- Start instances at 10 AM.**

**- Stop instances at 7 PM.**

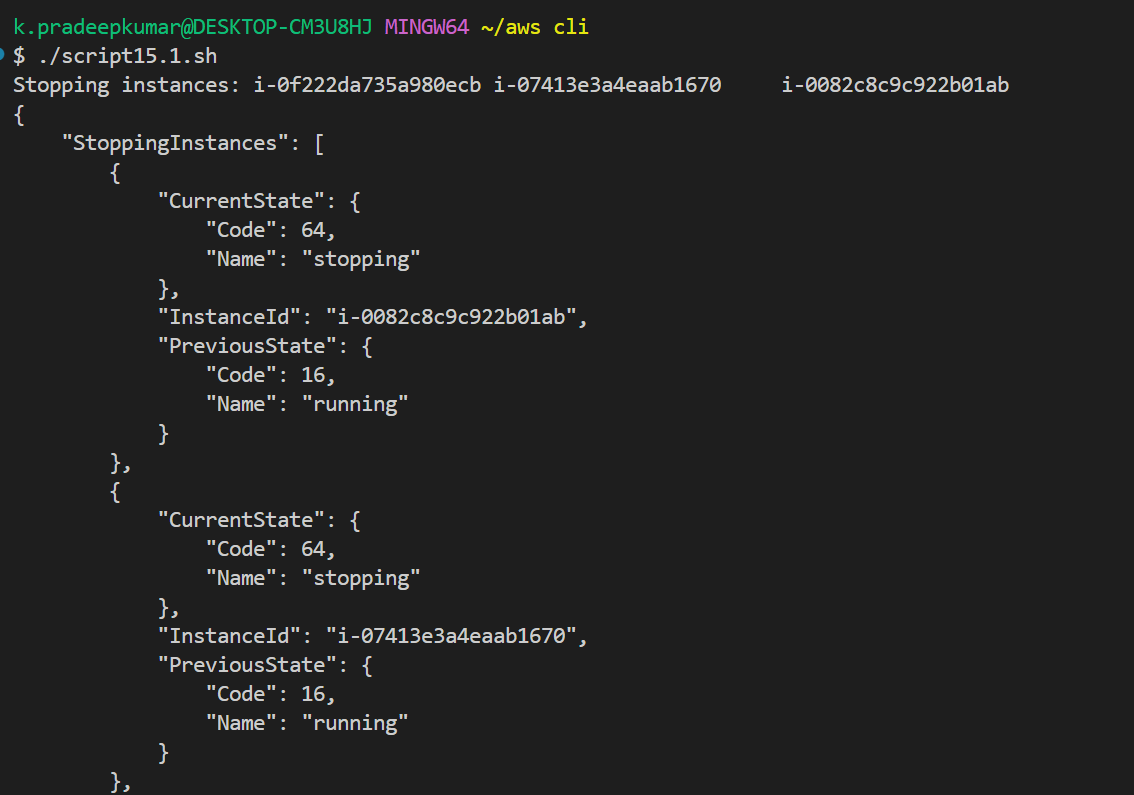
**1)**

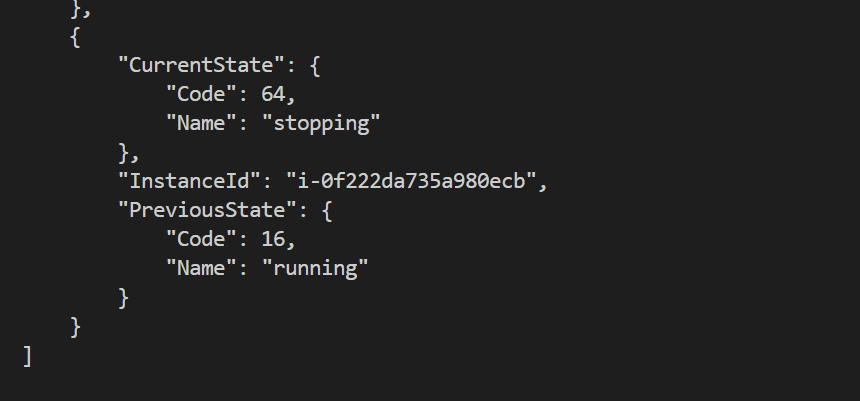
****

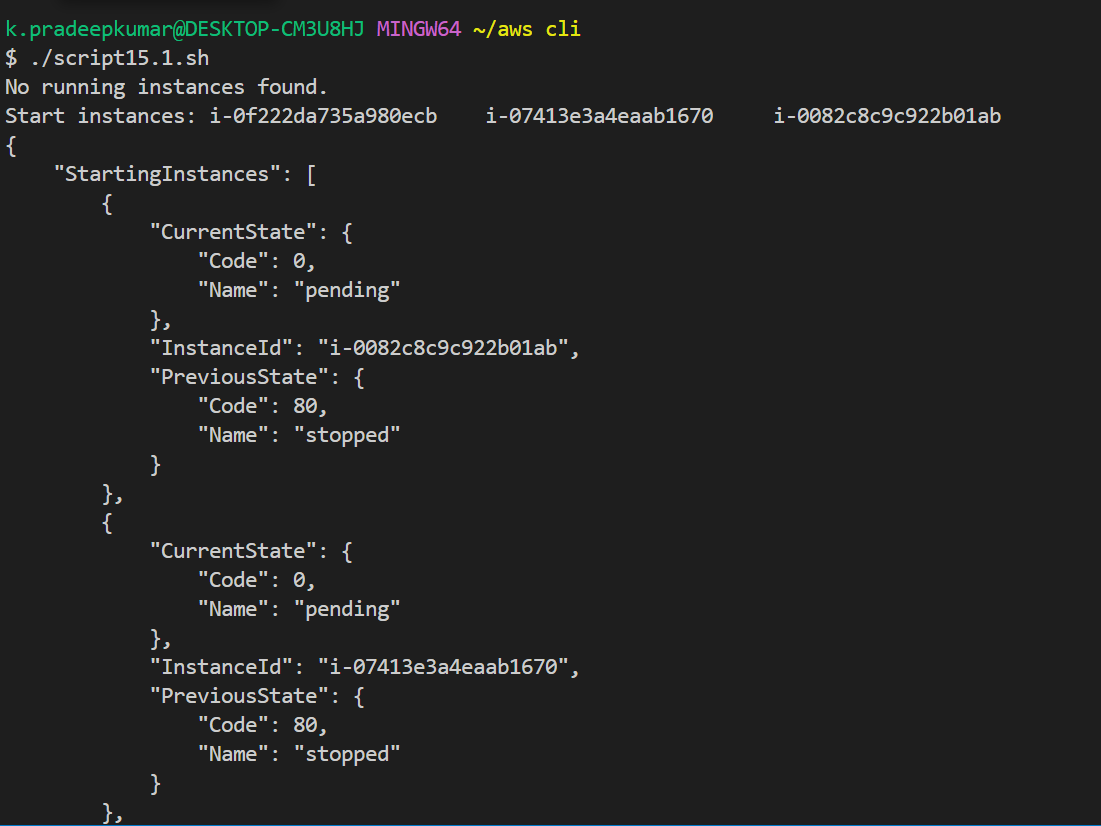
**2)**

****

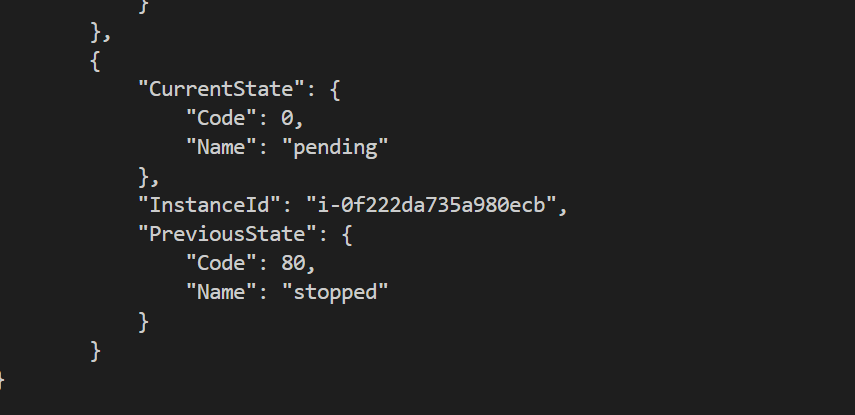
**3)**

****

**4)**

**5)**

**6)**

****

**--filters "Name=instance-state-name,Values=running" \**

**instance-state-name→** which corresponds to the current state of the EC2 instances.

**Values=running**—>It indicates that only instances that are currently in the "running" state should be returned.

**--query "Reservations[].Instances[].InstanceId"**

**Reservations[]--->**Each reservation can contain one or more instances.

**Instances[]-->**This refers to the list of instance objects within each reservation.

**InstanceId**→This is the specific field that contains the ID of the EC2 instance

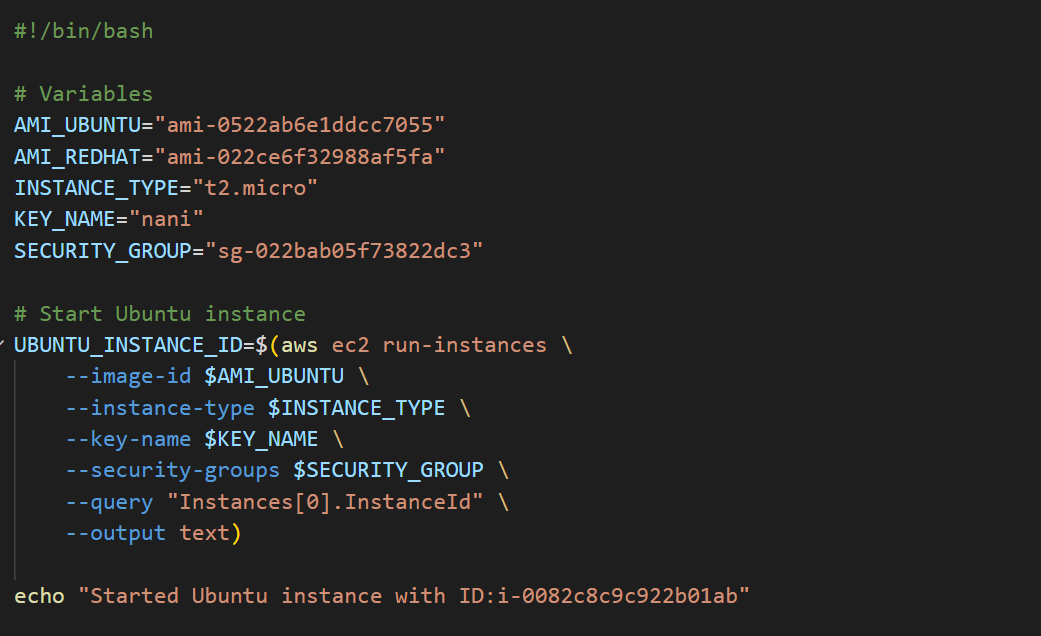
**if [ -z "$INSTANCE\_IDS" ]; then**

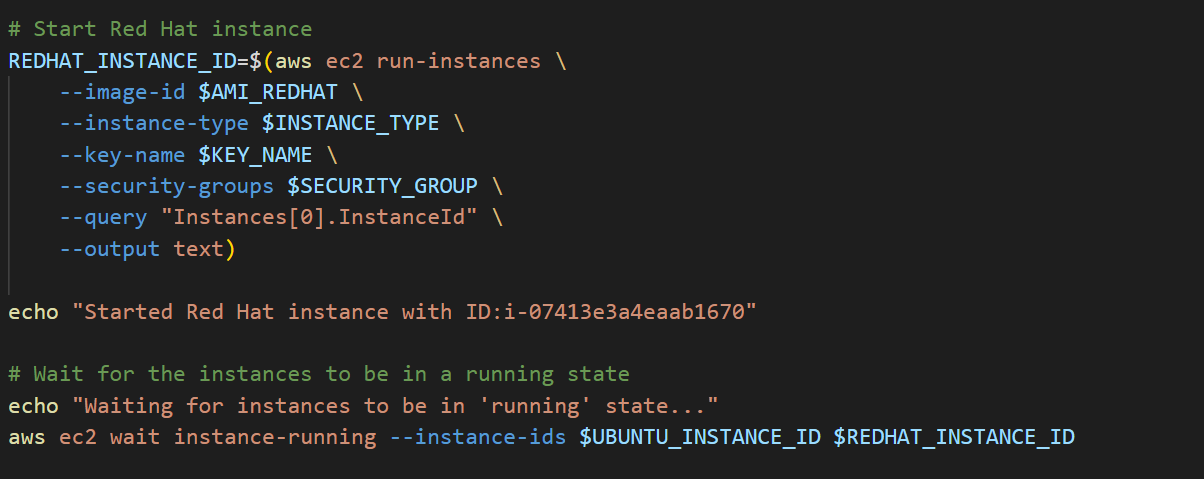
**-z—->**This is a unary test operator in Bash that checks if the length of the string is zero,

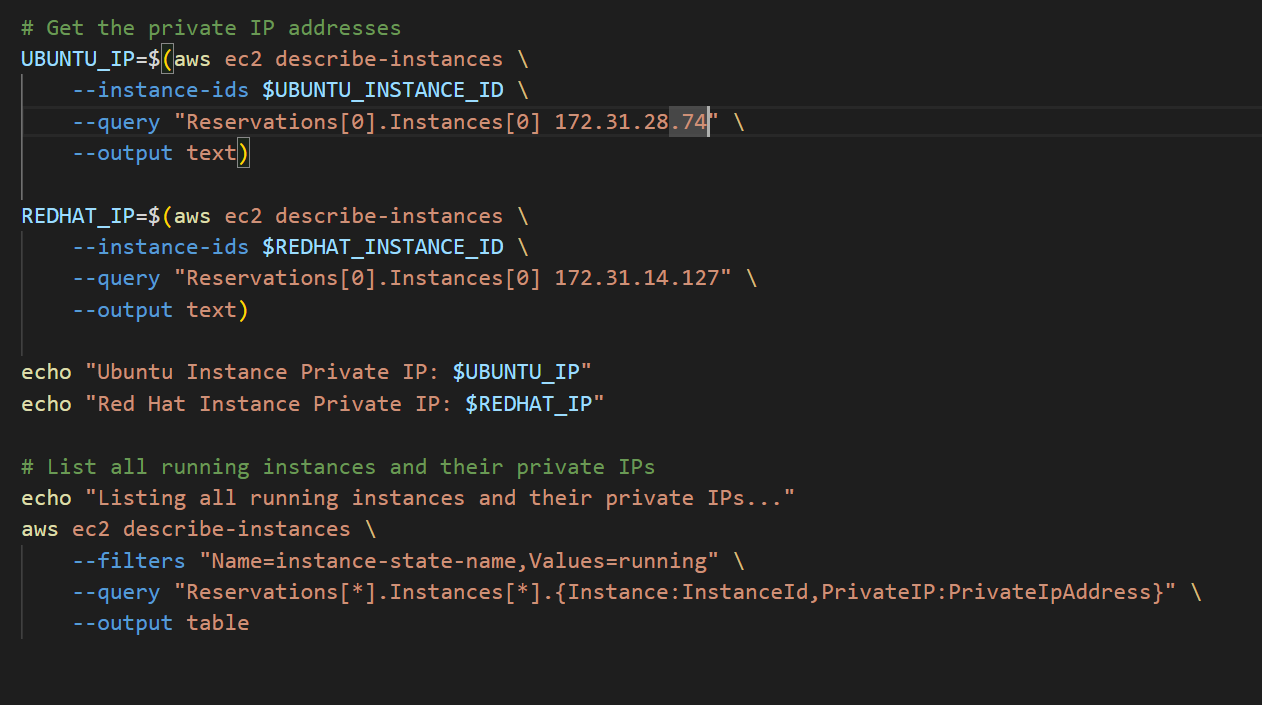
**24)**

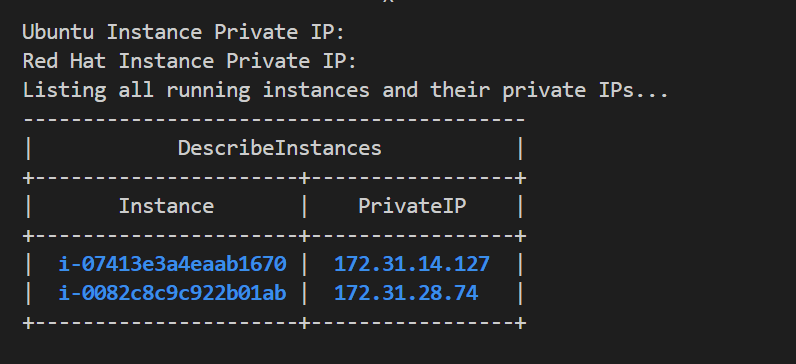
**Script**

**Use the AWS CLI to retrieve the list of private IP addresses of running instances. Start two instances with different operating systems (Red Hat, Ubuntu, CentOS) and get their private IP addresses.**

****

****

****

****