FIX URL USING ELASTIC IP & AUTOMATE START/STOP EC2 USING LAMBDA

Learn Lambda & Elastic IP Application

Indi Algo
https://twitter.com/indialgo https://t.me/IndiAlgo

Contents

Application of Unique URL in EC2

Typically, in ec2 whenever start/stop instance of EC2 every time a public IP get assigned. Which means URL of your hosted application is never be going to be same.

Above statement is in your favour if you're not willing to have unique address for your machine/application to be accessible to and secure from outside world from any sort of hacking.

But lookout below if you've any of the scenarios needed.

There're advantages as well of having unique address.

- If you've a python app running in Flask and wanted to host with fixed URL.
- If you want to connect to database server installed in ec2 to other applications.
- You can do SSH without going to AWS console on daily basis.

To have the Elastic IP created, assuming you've already created a EC2. Instance.

Step 1: Logon to https://console.aws.amazon.com

Step2: Change your region to Mumbai

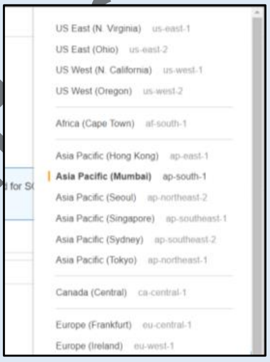
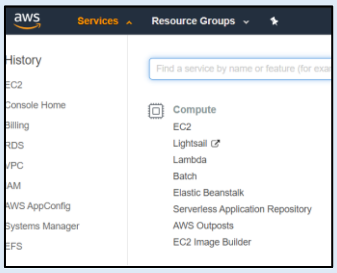

US East (N. Virginia) us-east-1
US East (Ohio) us-east-2
US West (N. California) us-west-1
US West (Oregon) us-west-2
Africa (Cape Town) af-south-1
Asia Pacific (Hong Kong) ap-east-1
Asia Pacific (Mumbai) ap-south-1
Asia Pacific (Seoul) ap-northeast-2
Asia Pacific (Singapore) ap-southeast-1
Asia Pacific (Sydney) ap-southeast-2
Asia Pacific (Tokyo) ap-northeast-1
Canada (Central) ca-central-1
Europe (Frankfurt) eu-central-1
Europe (Ireland) eu-west-1

Figure 1 Change region in AWS console

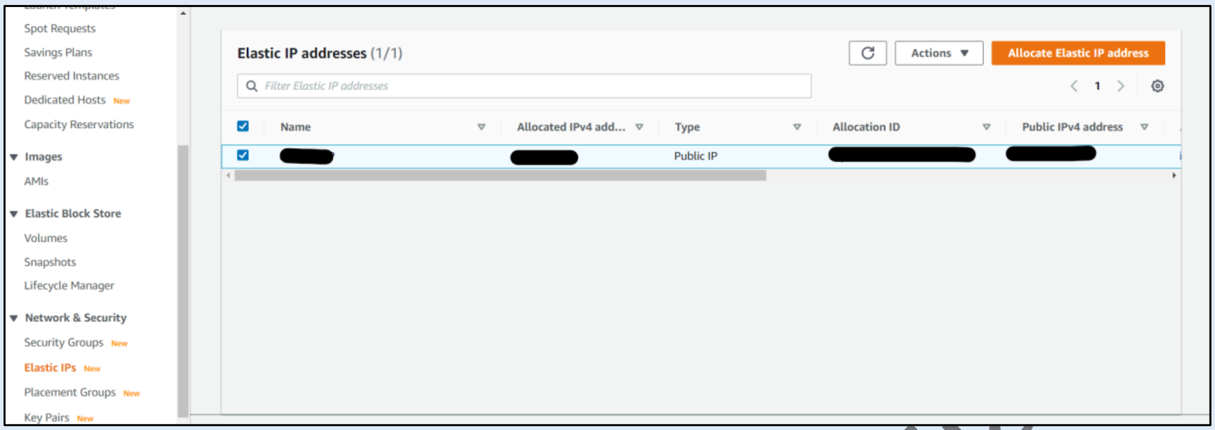
Step 3: Click on services > EC2



Step 4: Scroll a bit down in EC2 instances where you'll find

Network & Security → Elastic IPs

Click on Elastic IPs, you should be seeing something like below snap.



Elastic IP addresses (1/1)

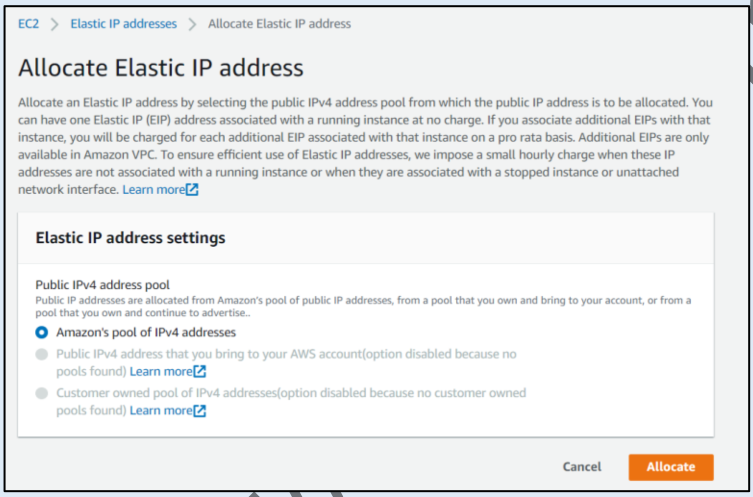
Filter Elastic IP addresses

✓	Name	Allocated IPv4 add…	Type	Allocation ID	Public IPv4 address
✓	[redacted]	[redacted]	Public IP	[redacted]	[redacted]

Figure 2 Elastic IPs default page

Step 5: Click on Allocate Elastic IP Address.

Step 6 : Follow as below



Allocate Elastic IP address

Allocate an Elastic IP address by selecting the public IPv4 address pool from which the public IP address is to be allocated. You can have one Elastic IP (EIP) address associated with a running instance at no charge. If you associate additional EIPs with that instance, you will be charged for each additional EIP associated with that instance on a pro rata basis. Additional EIPs are only available in Amazon VPC. To ensure efficient use of Elastic IP addresses, we impose a small hourly charge when these IP addresses are not associated with a running instance or when they are associated with a stopped instance or unattached network interface. Learn more

Elastic IP address settings

Public IPv4 address pool

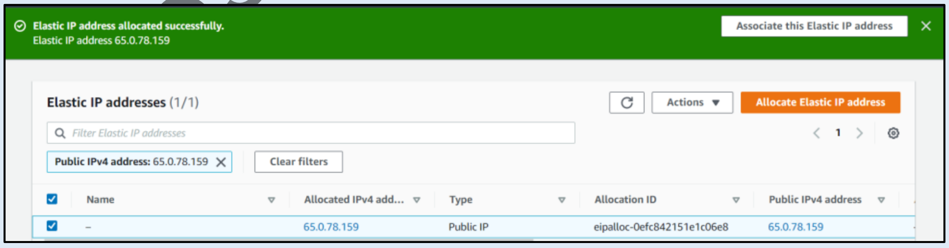
Public IP addresses are allocated from Amazon's pool of public IP addresses, from a pool that you own and bring to your account, or from a pool that you own and continue to advertise.

[✓ Amazon's pool of IPv4 addresses]

Public IPv4 address that you bring to your AWS account(option disabled because no pools found) Learn more

Customer owned pool of IPv4 addresses(option disabled because no customer owned pools found) Learn more

Step 7: After clicking on Allocate, you should see below snap



Elastic IP address allocated successfully.
Elastic IP address 65.0.78.159

Associate this Elastic IP address
Actions ▼
Allocate Elastic IP address
Clear filters

Elastic IP addresses (1/1)

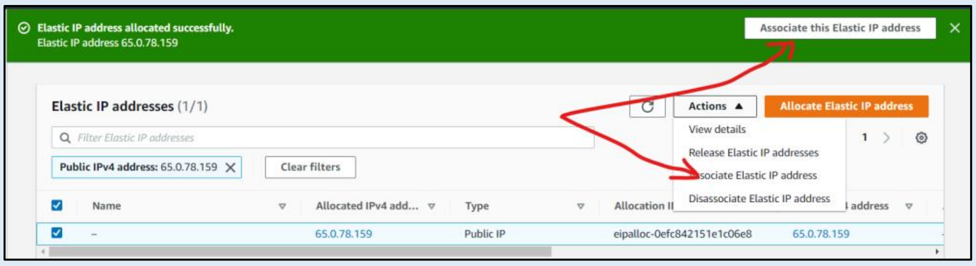
Filter Elastic IP addresses

Public IPv4 address: 65.0.78.159

✓	Name	Allocated IPv4 add…	Type	Allocation ID	Public IPv4 address
✓	–	65.0.78.159	Public IP	eipalloc-0efc842151e1c06e8	65.0.78.159

Step 8: Here comes the part where we're going to use it.

Click on Associate Elastic IP Address.


Elastic IP address allocated successfully.
Elastic IP address 65.0.78.159

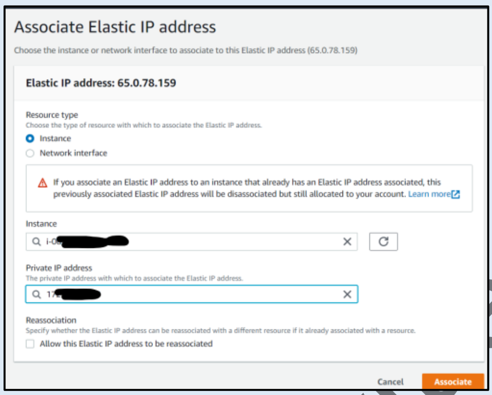
Associate this Elastic IP address
Elastic IP addresses (1/1)
Actions
Allocate Elastic IP address
Filter Elastic IP addresses
View details
Release Elastic IP addresses
ssociate Elastic IP address
Disassociate Elastic IP address
Public IPv4 address: 65.0.78.159
Clear filters

Name	Allocated IPv4 add...	Type	Allocation I	address
–	65.0.78.159	Public IP	eipalloc-0efc842151e1c06e8	65.0.78.159

Step 9: Click associate to your create instance id as shown in below snap

You've to choose

- Instance id
- Private IP address of that instance


Associate Elastic IP address

Choose the instance or network interface to associate to this Elastic IP address (65.0.78.159)

Elastic IP address: 65.0.78.159

Resource type
Choose the type of resource with which to associate the Elastic IP address.
- [✓ Instance]
- Network interface

If you associate an Elastic IP address to an instance that already has an Elastic IP address associated, this previously associated Elastic IP address will be disassociated but still allocated to your account. Learn more

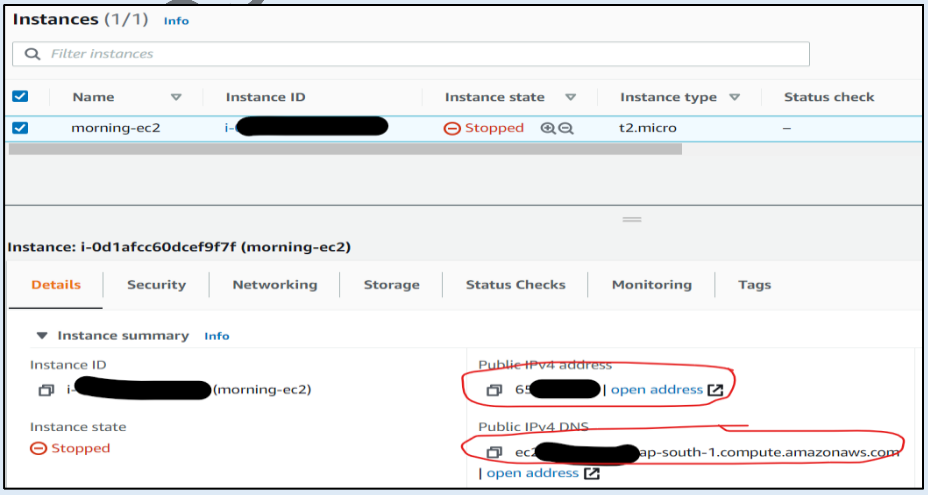
Instance
i-0[redacted]

Private IP address
The private IP address with which to associate the Elastic IP address.
17[redacted]

Reassociation
Specify whether the Elastic IP address can be reassociated with a different resource if it already associated with a resource.
Allow this Elastic IP address to be reassociated

And you're done.

To verify it's working. If you go to the instance details you should see DNS and Public IP are fixed and they'll never change unless you release Elastic IP. Even if you stop your instance.


Instances (1/1) Info

Name	Instance ID	Instance state	Instance type	Status check
morning-ec2	i-[redacted]	Stopped	t2.micro	–

Instance: i-0d1afcc60dcef9f7f (morning-ec2)

Instance summary Info

Instance ID
i-[redacted] (morning-ec2)

Public IPv4 address
65[redacted] | open address

Instance state
Stopped

Public IPv4 DNS
ec2[redacted]ap-south-1.compute.amazonaws.com
| open address

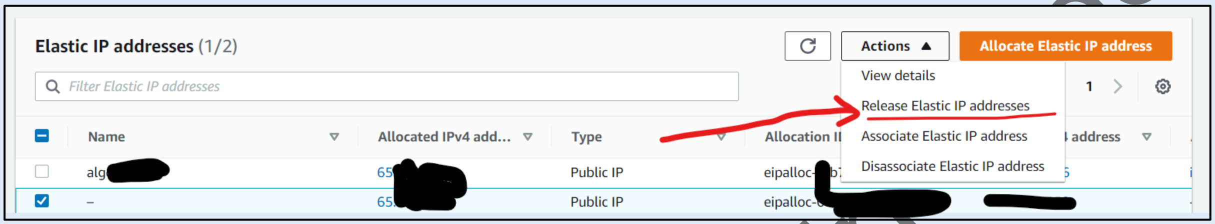
Delete Elastic IP mapped to EC2

In case if you’re not willing to have Elastic IP mapped you can always delete it.

But be cautioned if any applications are relying on that address, they’ll shall fail.

Step 1: Again, go to Services > EC2 > Instance > Networking & Security > Elastic IPs

Click on IP which you want to release and click on Release as shown in below snap.



Elastic IP addresses (1/2)

	Name	Allocated IPv4 add...	Type	Allocation ID
☐	alg[redacted]	65[redacted]	Public IP	eipalloc[redacted]
✓	–	65[redacted]	Public IP	eipalloc-[redacted]

Actions
Allocate Elastic IP address
Filter Elastic IP addresses
View details
Release Elastic IP addresses
Associate Elastic IP address
Disassociate Elastic IP address

After you’re done with this every time a new public IP would be assigned to your EC2 Instance.

Scheduling the EC2 to Auto Start/Stop

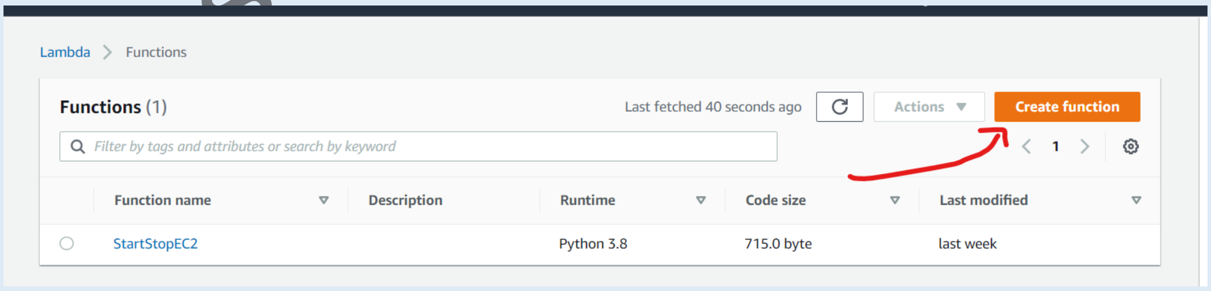
Let’s say you’ve a scenario where you don’t your machine to be running continuously backtoback to burn pocket money and you want to make to it to be on during specific time duration only.

In our scenario, It’s obviously let’s say market time from Monday to Friday from 9AM to 4PM let’s say.

So let’s see in below steps how can we achieve this.

Step1: Services(find in top black ribbon) → Lambda

Step2: Click on Create function as shown in snap.

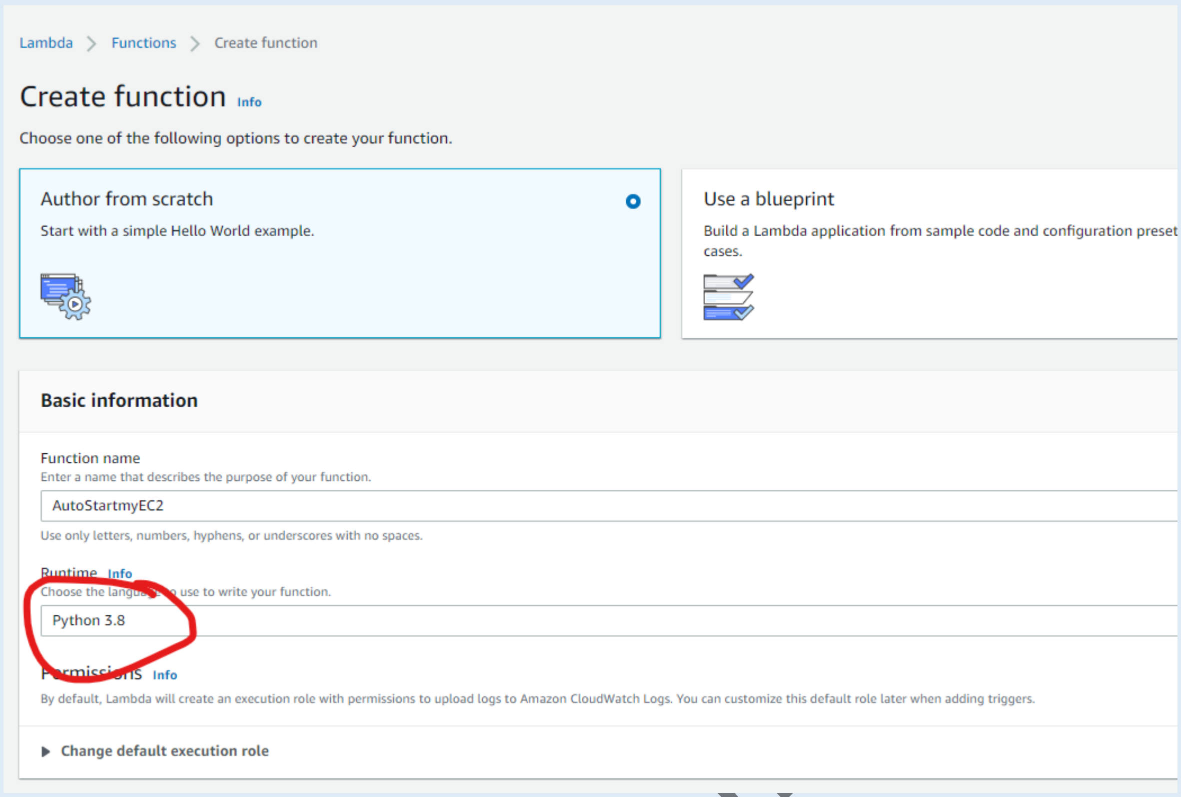


Lambda > Functions

Functions (1)   Last fetched 40 seconds ago

	Function name	Description	Runtime	Code size	Last modified
○	StartStopEC2		Python 3.8	715.0 byte	last week

Step3: Select python as runtime info as shown in below snap and click on create function.


Lambda > Functions > Create function

Create function Info

Choose one of the following options to create your function.

Author from scratch
Start with a simple Hello World example.

Use a blueprint
Build a Lambda application from sample code and configuration preset cases.

Basic information

Function name
Enter a name that describes the purpose of your function.
AutoStartmyEC2
Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime Info
Choose the language to use to write your function.
Python 3.8

Permissions Info
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▸ Change default execution role

Step 4: *****Important*** : Giving the proper permissions is very important to this function. So that it could have access to turn on/off your EC2 instance.**

https://www.howtoforge.com/aws-lambda-function-to-start-and-stop-ec2-instance/

Follow simple steps above to create policies and attach link above to assign the role.

Make sure you assign AmazonEC2FullAccess

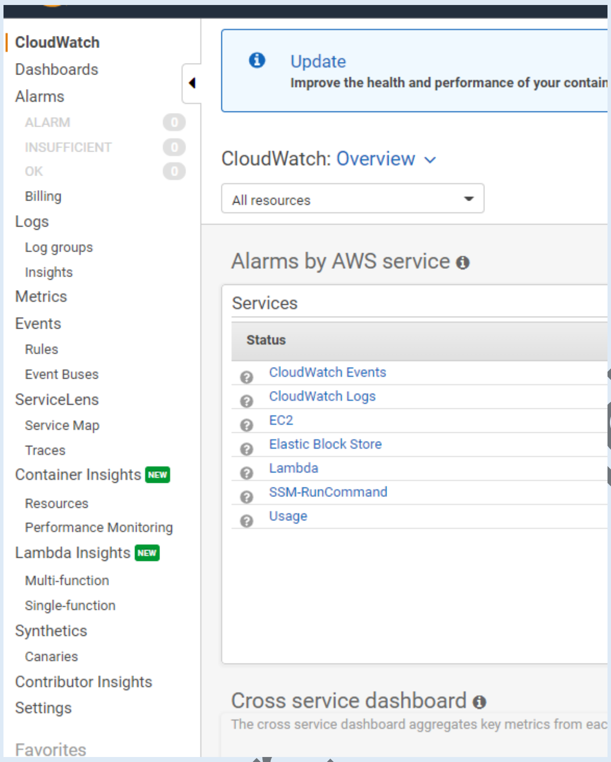
Step 5: our code for lambda function:

```
import json
import boto3
import time
import urllib3

region = 'ap-south-1'
instances = ['i-0adcf094b3e497e99']
#ec2 = boto3.client('ec2', region_name=region)
ec2 = boto3.resource('ec2')
client = boto3.client('ec2')


def lambda_handler(event, context):
   instance = ec2.Instance(id='i-0adcf094b3e49799')
   print("starting instance " + 'i-0adcf094b3e49799')
   waiter = client.get_waiter('instance_running')
   instance.start()  # If you want write for stop write as instance.stop()
   # time.sleep(200)

   waiter.wait(InstanceIds=instances)
   # Reload the instance attributes
   instance.load()
   print(instance.public_dns_name)
   return {

      "statusCode":200,
      "status":instance.public_dns_name
   }
```

Step 6: now comes the part of scheduling the above lambda.

Goto : https://ap-south-1.console.aws.amazon.com/cloudwatch/home

Step 7: in left menu choose Rules Events > Rules as shown in below snap



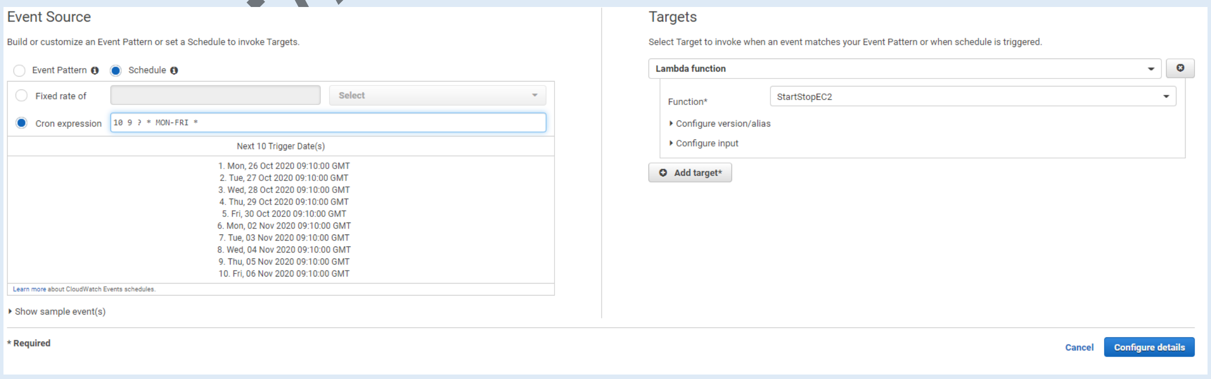
Step 8: Click on create rule

Select CRON expression and paste below code

```
10 9 ? * MON-FRI *
```

Refer here if you want your own timings.

https://docs.aws.amazon.com/AmazonCloudWatch/latest/events/ScheduledEvents.html



Event Source

Build or customize an Event Pattern or set a Schedule to invoke Targets.

Event Pattern [✓ Schedule]

Fixed rate of | Select

[✓ Cron expression] `10 9 ? * MON-FRI *`

Next 10 Trigger Date(s)

1. Mon, 26 Oct 2020 09:10:00 GMT
2. Tue, 27 Oct 2020 09:10:00 GMT
3. Wed, 28 Oct 2020 09:10:00 GMT
4. Thu, 29 Oct 2020 09:10:00 GMT
5. Fri, 30 Oct 2020 09:10:00 GMT
6. Mon, 02 Nov 2020 09:10:00 GMT
7. Tue, 03 Nov 2020 09:10:00 GMT
8. Wed, 04 Nov 2020 09:10:00 GMT
9. Thu, 05 Nov 2020 09:10:00 GMT
10. Fri, 06 Nov 2020 09:10:00 GMT

Learn more about CloudWatch Events schedules.

Show sample event(s)

Targets

Select Target to invoke when an event matches your Event Pattern or when schedule is triggered.

Lambda function

Function* StartStopEC2

Configure version/alias

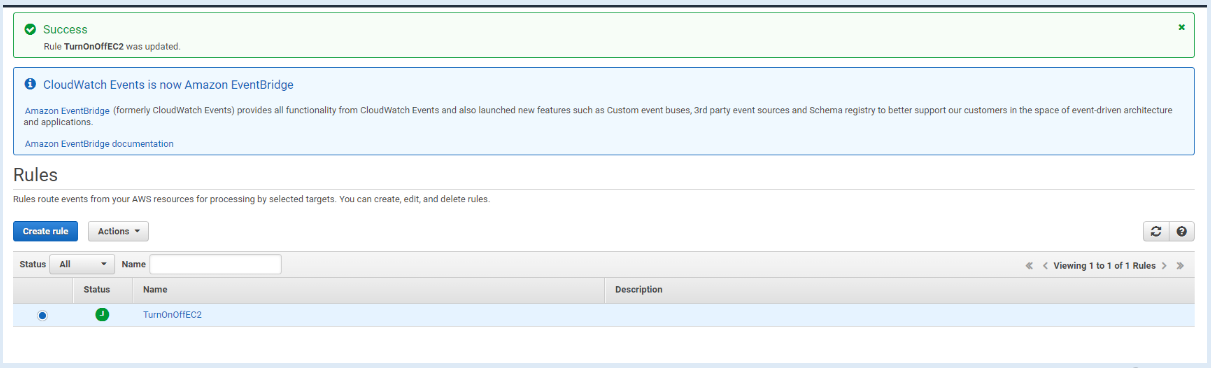
Configure input

Add target*

* Required

Cancel | Configure details

Click on Configure details and you’re done



Success

Rule **TurnOnOffEC2** was updated.

CloudWatch Events is now Amazon EventBridge

Amazon EventBridge (formerly CloudWatch Events) provides all functionality from CloudWatch Events and also launched new features such as Custom event buses, 3rd party event sources and Schema registry to better support our customers in the space of event-driven architecture and applications.

Amazon EventBridge documentation

Rules

Rules route events from your AWS resources for processing by selected targets. You can create, edit, and delete rules.

	Status	Name	Description
●	●	TurnOnOffEC2	

In case if you want to run for American time just play around changing running hours.

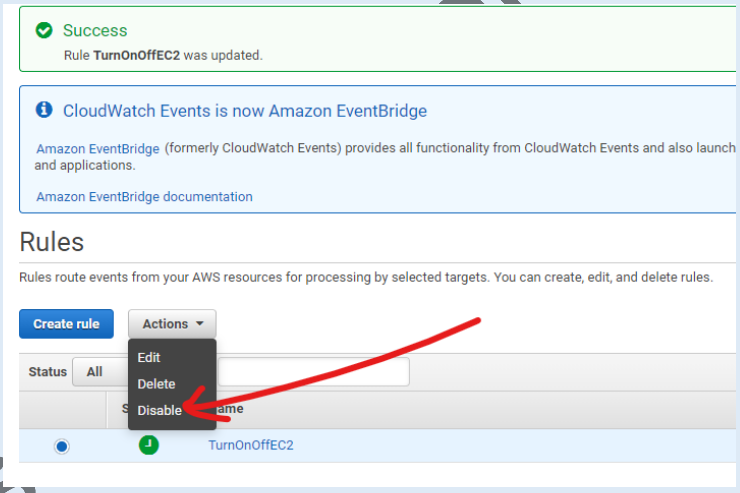
You’ll follow same to process to writing the stop function.

Disabling the scheduler.

Again go to Events > Rules

Select rule you’ve created and from above menu click on disable.

You don’t need to delete your lambda function as you would NOT be charged.



Success

Rule **TurnOnOffEC2** was updated.

CloudWatch Events is now Amazon EventBridge

Amazon EventBridge (formerly CloudWatch Events) provides all functionality from CloudWatch Events and also launch
and applications.

Amazon EventBridge documentation

Rules

Rules route events from your AWS resources for processing by selected targets. You can create, edit, and delete rules.

References:

- https://aws.amazon.com/premiumsupport/knowledge-center/start-stop-lambda-cloudwatch/
- https://www.howtoforge.com/aws-lambda-function-to-start-and-stop-ec2-instance/
- https://docs.bitnami.com/aws/faq/configuration/configure-static-address/
- https://aws.amazon.com/premiumsupport/knowledge-center/elastic-ip-charges/
- https://blog.shikisoft.com/3-ways-to-schedule-aws-lambda-and-step-functions-state-machines