Assignment

Que.*Create VPC with public and private subnet*

Ans*.Following are the steps to create VPC with Public Subnet and Private Subnet*

1.Create New VPC (MyVPC)

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP options set	Main Route table
MyVPC	vpc-0eb70db6d6c4f2ddc	available	10.200.0....	-	dopt-75b1411e	rtb-09abb9fba1c5978b
	vpc-65818b0d	available	172.31.0....	-	dopt-75b1411e	rtb-25bb704e

State	available	Default VPC	No
IPv4 CIDR	10.200.0.0/16	IPv6 CIDR	-
IPv6 Pool	-	DNS resolution	Enabled
Network ACL	acl-09306a9f9dccf6e4f	DNS hostnames	Disabled
DHCP options set	dopt-75b1411e	Route table	rtb-09abb9fba1c5978b6
Owner	545033201860		

2.Create Internet Gateway (MyIGW) So VPC can talk with Internet

Name	ID	State	VPC	Owner
MyIGW	igw-0bef23020ad0...	attached	vpc-0eb70db6d6c...	545033201860
	igw-d3da7dbb	attached	vpc-65818b0d	545033201860

Internet gateway: igw-0bef23020ad060917

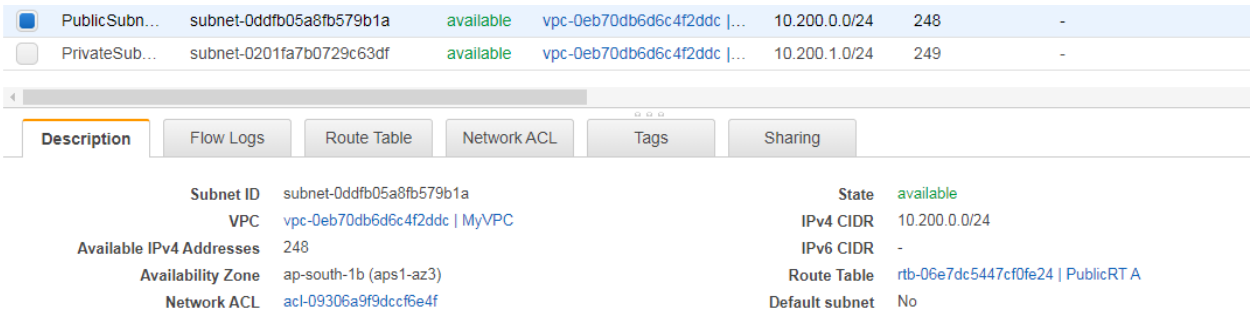
Description | Tags

ID	igw-0bef23020ad060917	Attached VPC ID	vpc-0eb70db6d6c4f2ddc | MyVPC
State	attached	Owner	545033201860

3.Attach InternetGateway to MyVPC

*4.**Public Subnets***

->Create Public Subnet

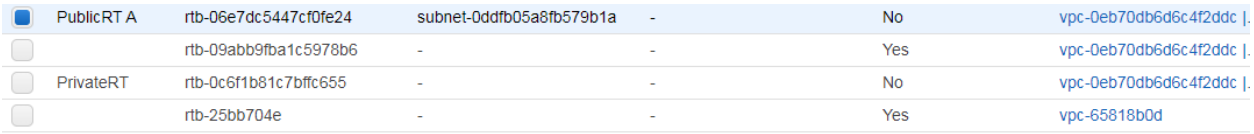


PublicSubn...	subnet-0ddfb05a8fb579b1a	available	vpc-0eb70db6d6c4f2ddc |...	10.200.0.0/24	248	-
PrivateSub...	subnet-0201fa7b0729c63df	available	vpc-0eb70db6d6c4f2ddc |...	10.200.1.0/24	249	-

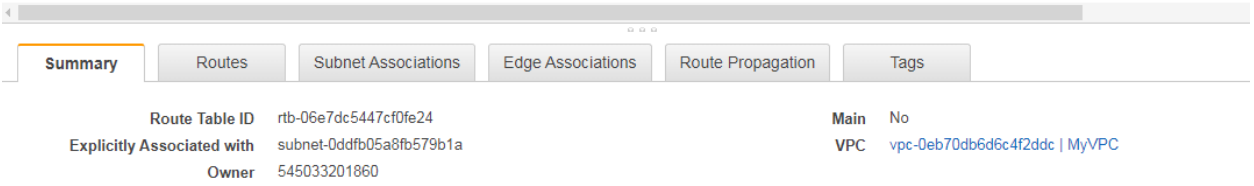
Description | Flow Logs | Route Table | Network ACL | Tags | Sharing

Subnet ID	subnet-0ddfb05a8fb579b1a	**State**	available
VPC	vpc-0eb70db6d6c4f2ddc | MyVPC	**IPv4 CIDR**	10.200.0.0/24
Available IPv4 Addresses	248	**IPv6 CIDR**	-
Availability Zone	ap-south-1b (aps1-az3)	**Route Table**	rtb-06e7dc5447cf0fe24 | PublicRT A
Network ACL	acl-09306a9f9dccf6e4f	**Default subnet**	No

->Create Route Table and associate with Public Subnet



PublicRT A	rtb-06e7dc5447cf0fe24	subnet-0ddfb05a8fb579b1a	-	No	vpc-0eb70db6d6c4f2ddc |.
	rtb-09abb9fba1c5978b6	-	-	Yes	vpc-0eb70db6d6c4f2ddc |.
PrivateRT	rtb-0c6f1b81c7bffc655	-	-	No	vpc-0eb70db6d6c4f2ddc |.
	rtb-25bb704e	-	-	Yes	vpc-65818b0d



Summary | Routes | Subnet Associations | Edge Associations | Route Propagation | Tags

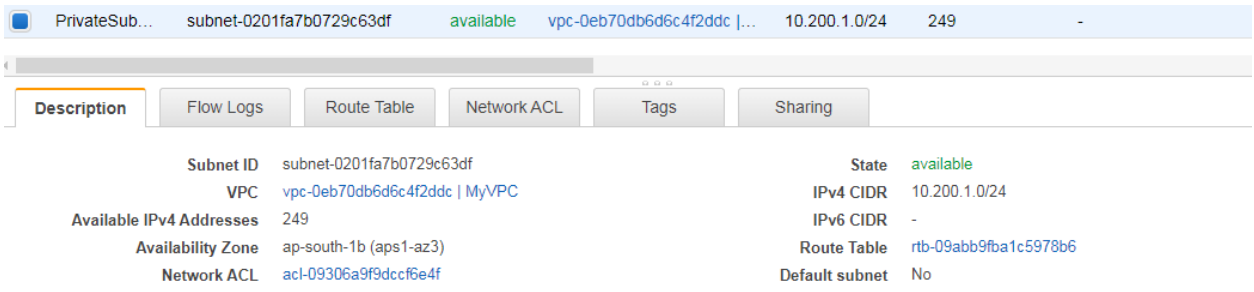
Route Table ID	rtb-06e7dc5447cf0fe24	**Main**	No
Explicitly Associated with	subnet-0ddfb05a8fb579b1a	**VPC**	vpc-0eb70db6d6c4f2ddc | MyVPC
Owner	545033201860		

->Add Route to the Internet

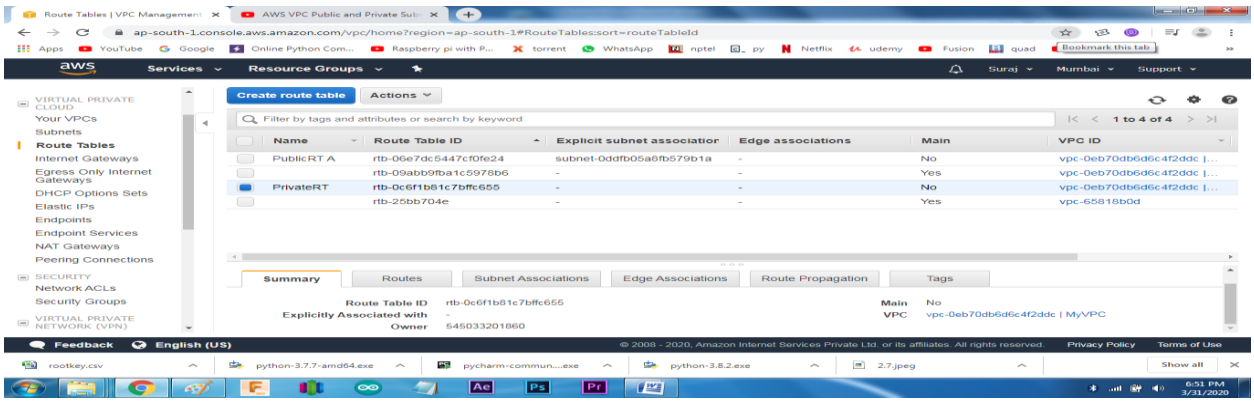
->Associate these Route table with Subnet So subnet become public subset

5 .***Private Subnet***

->Create Private Subnet



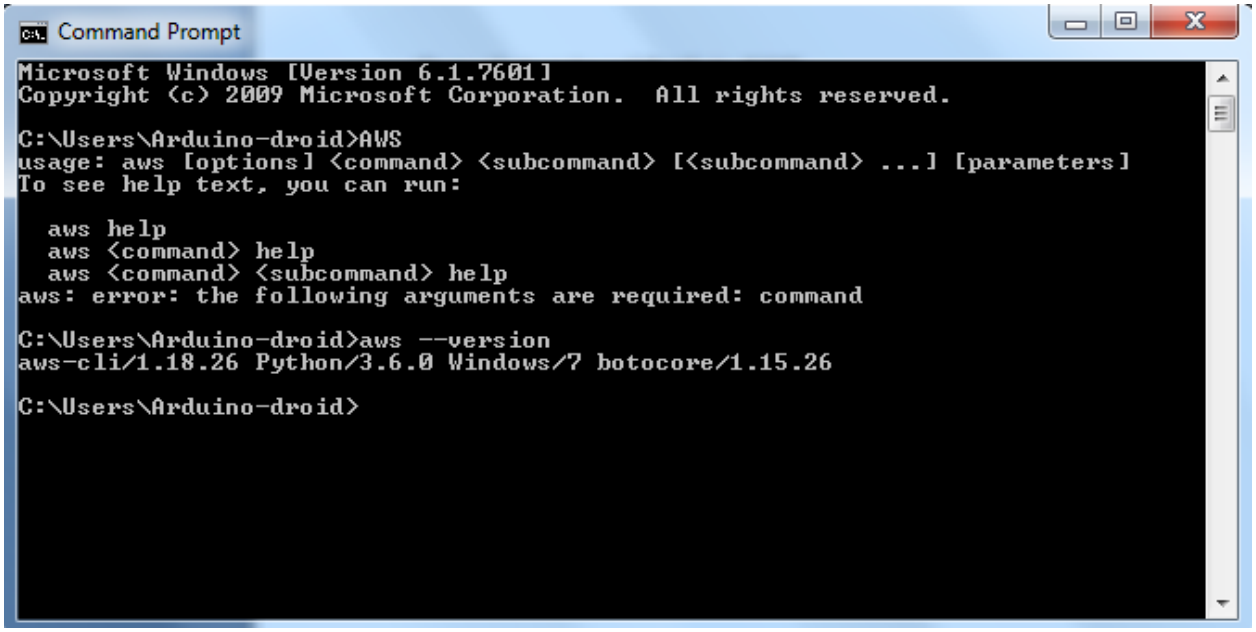
->Create Route Table with Private Subnet



->It has Route within VPC Dont't Modify because it is Private Subset

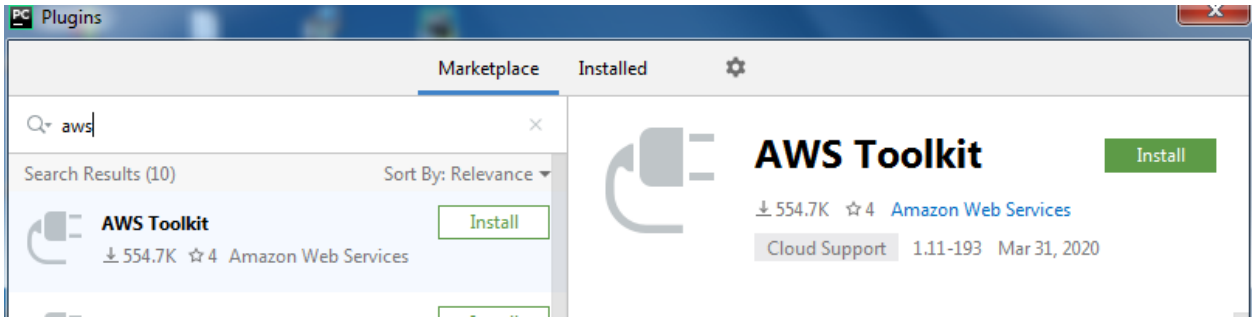
*Que.>**Setup Local Path in AWS***

ANS.



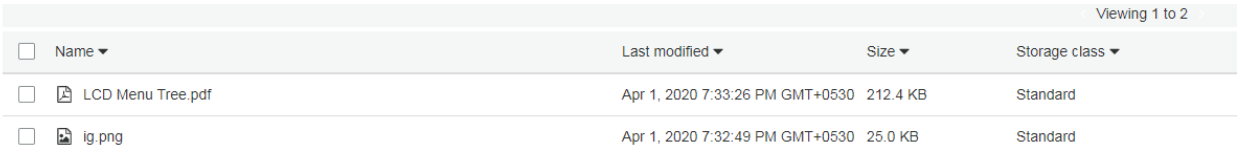
*Que.> **Install Pycharm and aws toolkit plugin in pycharm***

Ans.



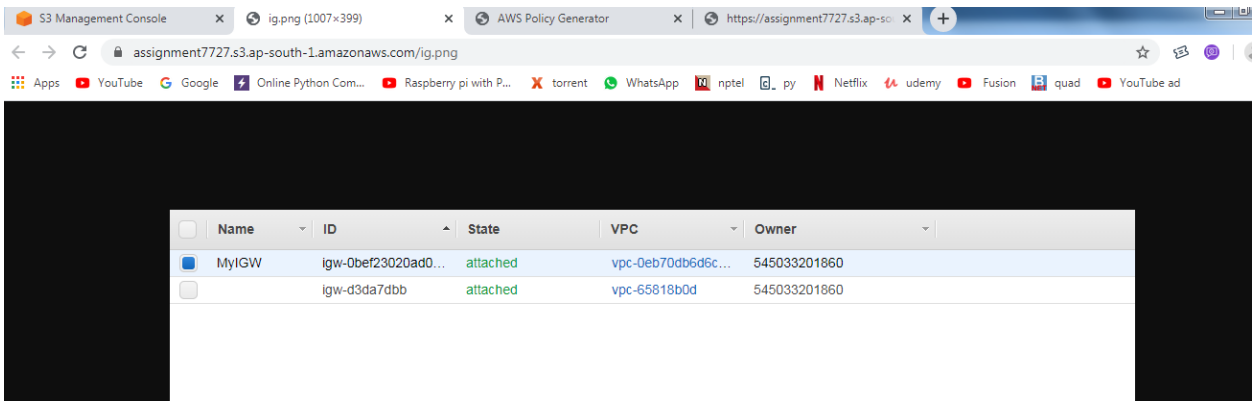
Que.Create Bucket Policy on your own upload some images and pdf which give public acess to images not to pdf

Ans. -> Upload pdf and png file in the S3 bucket

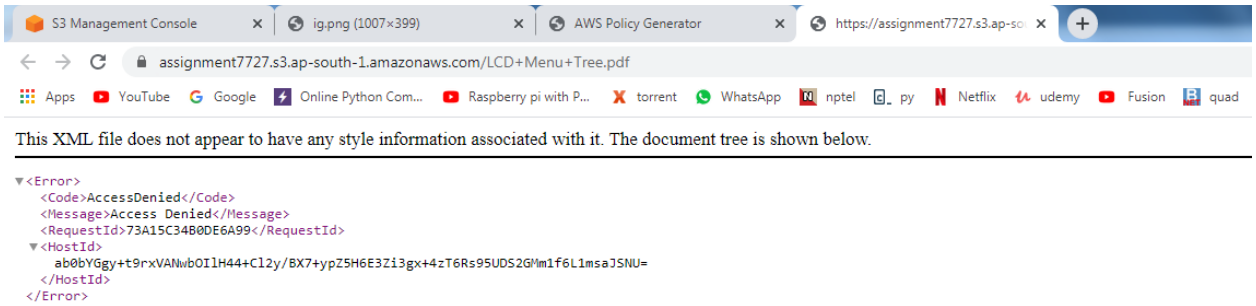


->Create bucket policy using policy generatot that allow png file and deny pdf file

-> Bucket Policy give acess to Png file



->Bucket Policy Deny acess to pdf file



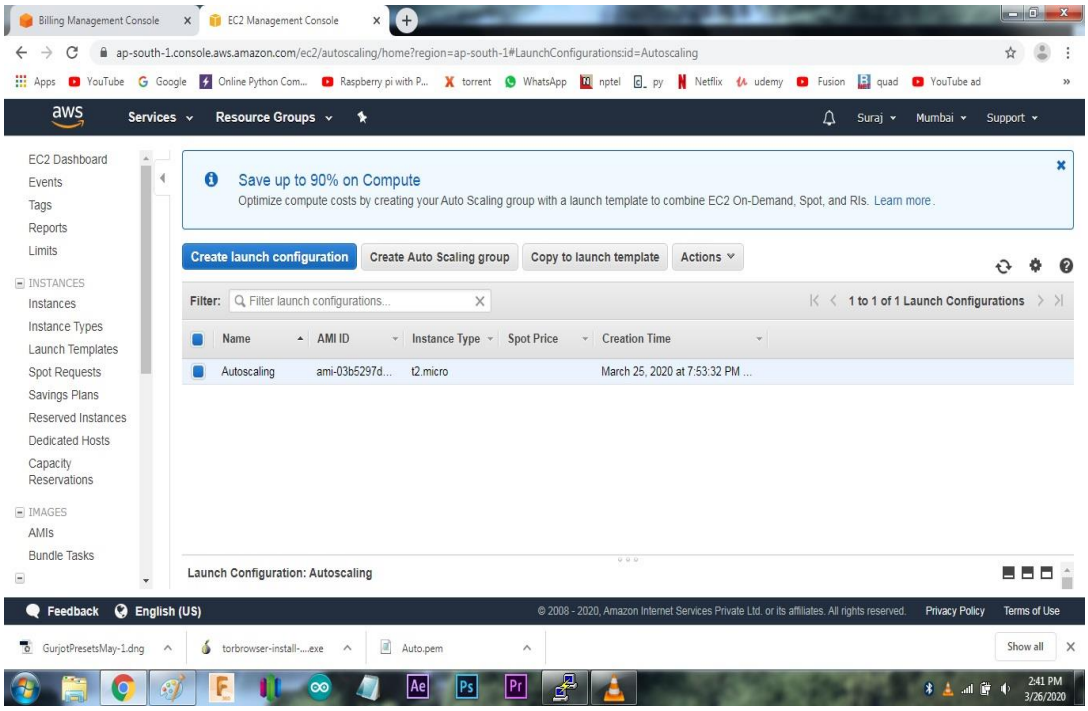
Que. Write the steps to create an autoscaling group and associate the EC2 instances of that autoscaling group with an Application load balancer. When you click on the URL of Application load balancer, it should return the content from EC2 instances.

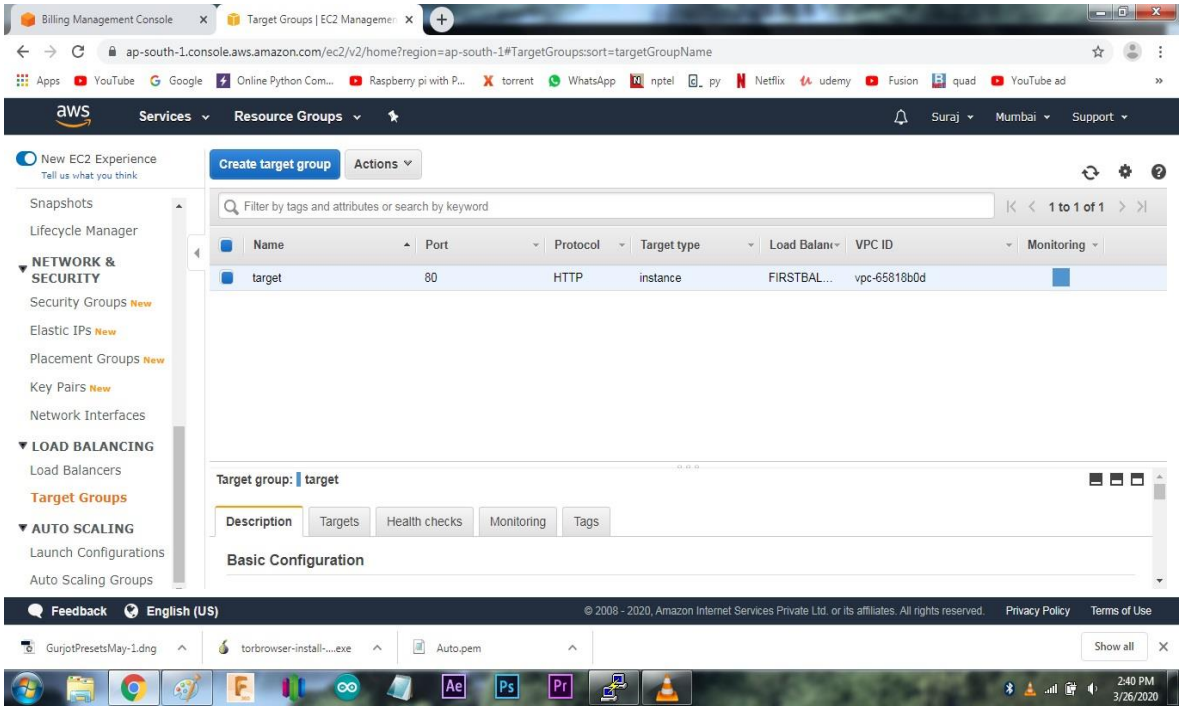
*Ans.**->Steps to create Autoscaling Group***

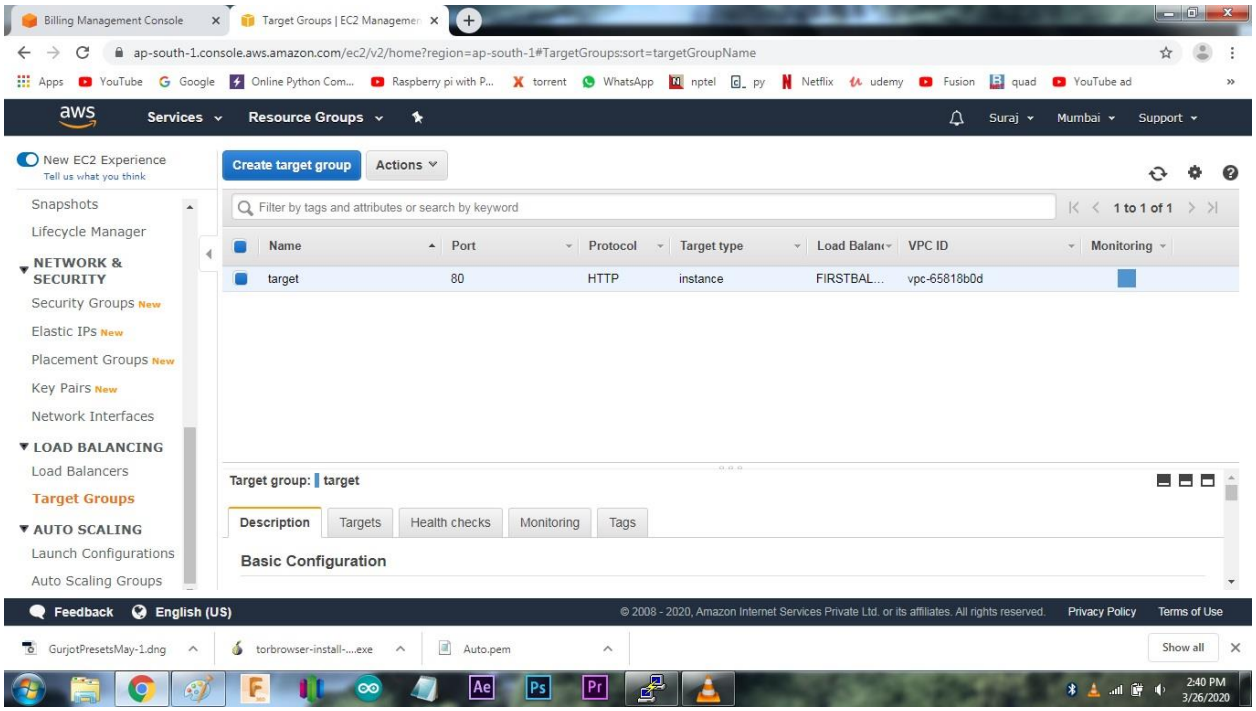
1. *On the Auto Scaling groups page, choose Create an Auto Scaling group.*
2. *On the Choose launch configuration page, for Auto Scaling group name, enter a name for your Auto Scaling group.*
3. *Launch template*
4. *Keep Network set to the default VPC for your chosen AWS Region, or select your own VPC..*
5. *For Subnet, choose a subnet*
6. *On the Review page, review the information for the group, and then choose Create Auto Scaling group.*

->Steps to Create Load Balancer

1.Create instance in EC2.
2.Create classic load balancer
3.Create target group .
4.Under target group in the target tab we need to add targets
5.Create application load balance

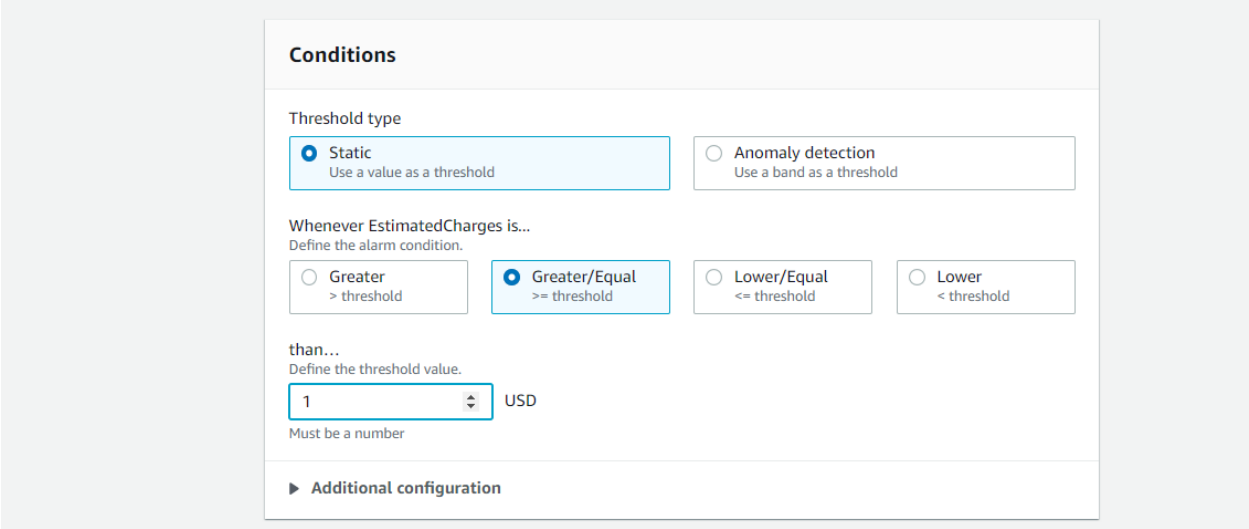

Billing Management Console
EC2 Management Console
ap-south-1.console.aws.amazon.com/ec2/autoscaling/home?region=ap-south-1#LaunchConfigurations:id=Autoscaling
Services
Resource Groups
Suraj
Mumbai
Support
EC2 Dashboard
Events
Tags
Reports
Limits
INSTANCES
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
IMAGES
AMIs
Bundle Tasks
Save up to 90% on Compute
Optimize compute costs by creating your Auto Scaling group with a launch template to combine EC2 On-Demand, Spot, and RIs. Learn more.
Create launch configuration
Create Auto Scaling group
Copy to launch template
Actions
Filter launch configurations...
1 to 1 of 1 Launch Configurations
Name
AMI ID
Instance Type
Spot Price
Creation Time
Autoscaling
ami-03b5297d...
t2.micro
March 25, 2020 at 7:53:32 PM ...
Launch Configuration: Autoscaling
Feedback
English (US)
© 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
GurjotPresetsMay-1.dng
torbrowser-install-....exe
Auto.pem
Show all
2:41 PM
3/26/2020

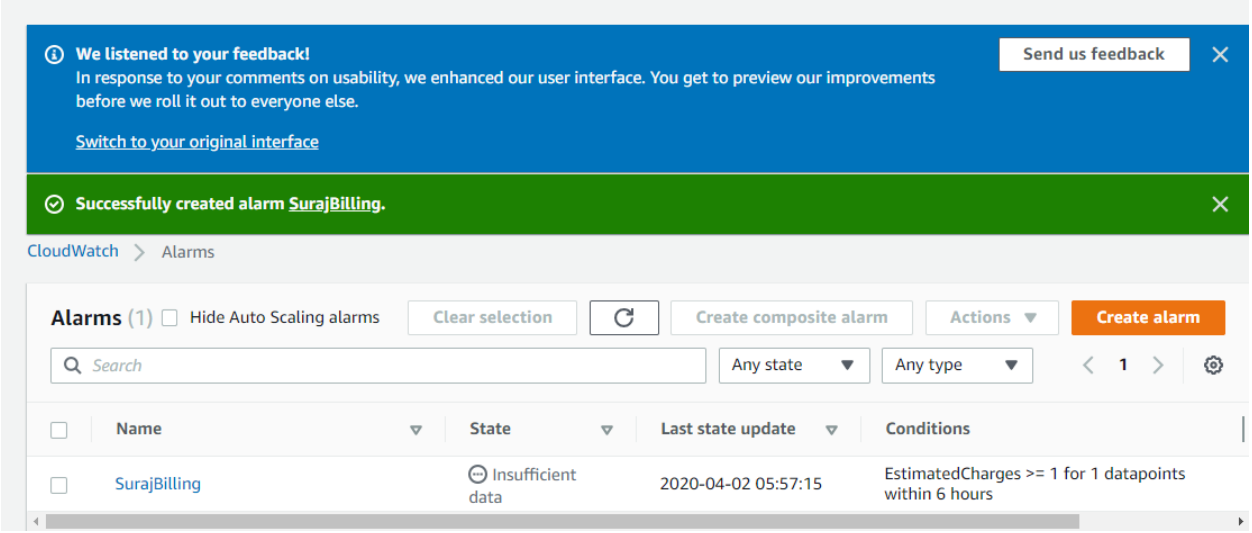

Billing Management Console
Target Groups | EC2 Manageme
ap-south-1.console.aws.amazon.com/ec2/v2/home?region=ap-south-1#TargetGroups:sort=targetGroupName
Services
Resource Groups
Suraj
Mumbai
Support
New EC2 Experience
Tell us what you think
Snapshots
Lifecycle Manager
NETWORK & SECURITY
Security Groups New
Elastic IPs New
Placement Groups New
Key Pairs New
Network Interfaces
LOAD BALANCING
Load Balancers
Target Groups
AUTO SCALING
Launch Configurations
Auto Scaling Groups
Create target group
Actions
Filter by tags and attributes or search by keyword
1 to 1 of 1
Name
Port
Protocol
Target type
Load Balanc
VPC ID
Monitoring
target
80
HTTP
instance
FIRSTBAL...
vpc-65818b0d
Target group: target
Description
Targets
Health checks
Monitoring
Tags
Basic Configuration
Feedback
English (US)
© 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
GurjotPresetsMay-1.dng
torbrowser-install-....exe
Auto.pem
Show all
2:40 PM
3/26/2020


Billing Management Console
Target Groups | EC2 Manageme
ap-south-1.console.aws.amazon.com/ec2/v2/home?region=ap-south-1#TargetGroups:sort=targetGroupName
Apps YouTube Google Online Python Com... Raspberry pi with P... torrent WhatsApp nptel py Netflix udemy Fusion quad YouTube ad
aws Services Resource Groups Suraj Mumbai Support
New EC2 Experience
Tell us what you think
Snapshots
Lifecycle Manager
NETWORK & SECURITY
Security Groups New
Elastic IPs New
Placement Groups New
Key Pairs New
Network Interfaces
LOAD BALANCING
Load Balancers
Target Groups
AUTO SCALING
Launch Configurations
Auto Scaling Groups
Create target group
Actions
Filter by tags and attributes or search by keyword
1 to 1 of 1
Name Port Protocol Target type Load Balanc VPC ID Monitoring
target 80 HTTP instance FIRSTBAL... vpc-65818b0d
Target group: target
Description Targets Health checks Monitoring Tags
Basic Configuration
Feedback English (US)
© 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy Terms of Use
GurjotPresetsMay-1.dng
torbrowser-install-....exe
Auto.pem
Show all
2:40 PM
3/26/2020

Que. Create a billing alarm on your account to check if your account billing exceeds $1 for the current month

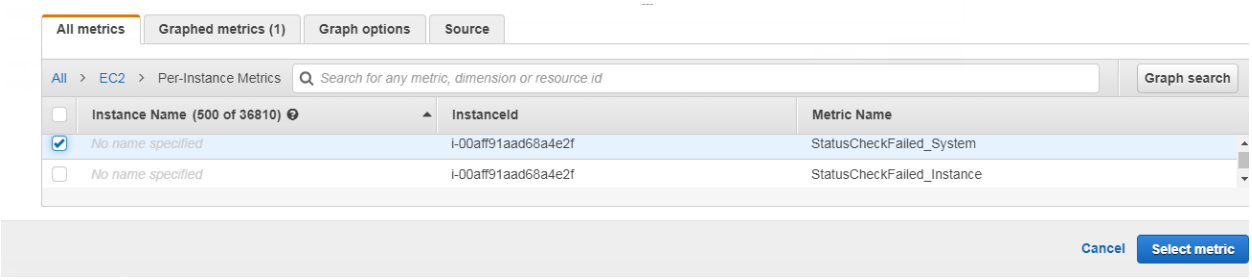
Ans.

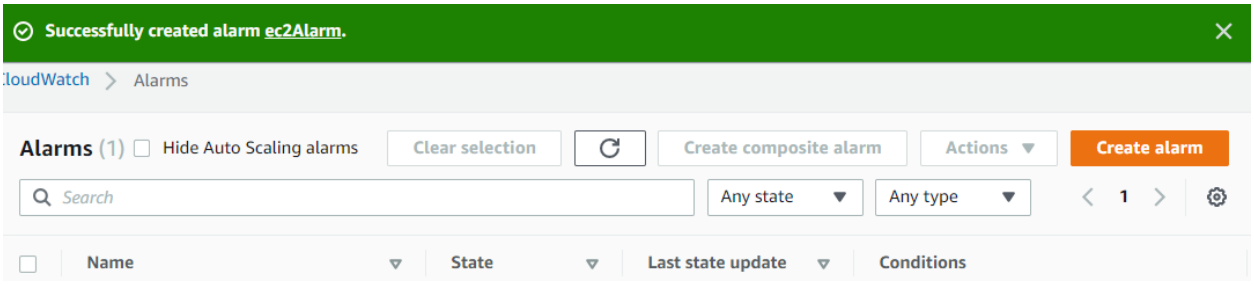




Que. Create an alarm on EC2 instance, to notify you if the instance is in stopped state or terminated state

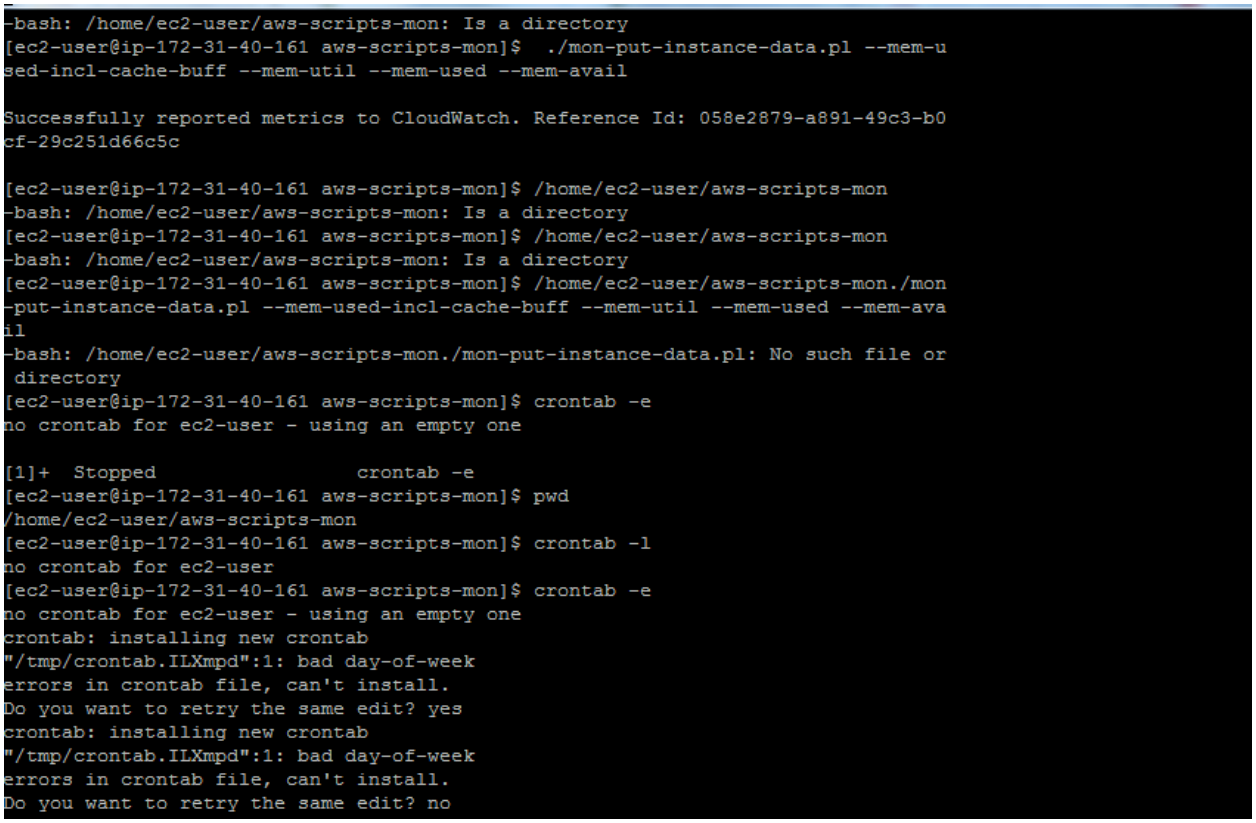
Ans.



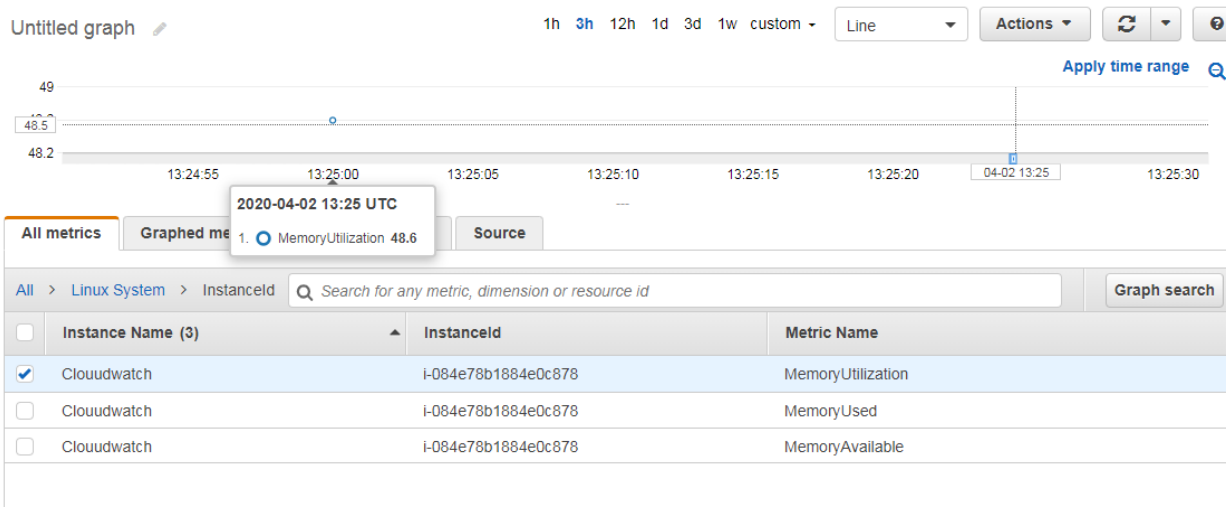


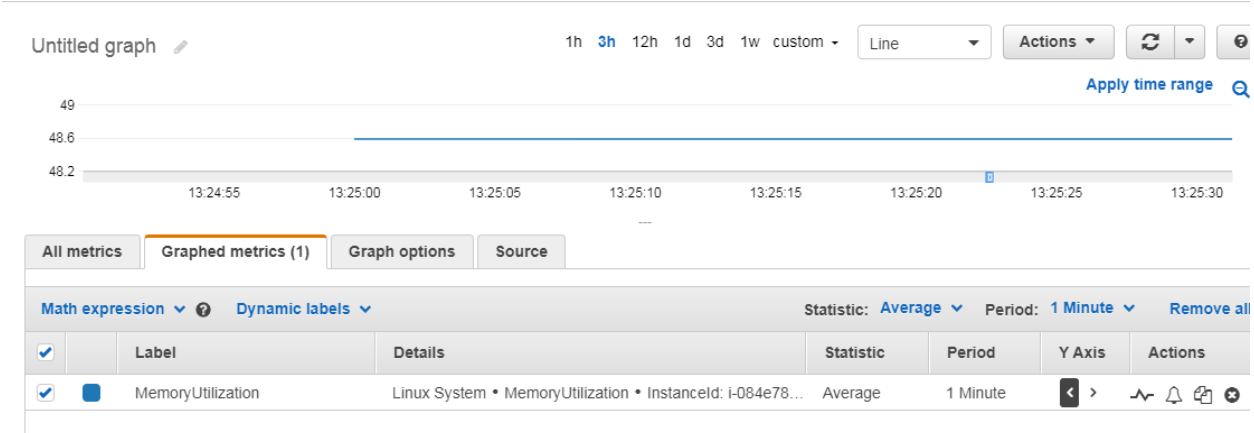
Que.Create a custom metric on EC2, to find memory utilization of a running EC2 instance.

Ans.



```
-bash: /home/ec2-user/aws-scripts-mon: Is a directory
[ec2-user@ip-172-31-40-161 aws-scripts-mon]$  ./mon-put-instance-data.pl --mem-u
sed-incl-cache-buff --mem-util --mem-used --mem-avail

Successfully reported metrics to CloudWatch. Reference Id: 058e2879-a891-49c3-b0
cf-29c251d66c5c

[ec2-user@ip-172-31-40-161 aws-scripts-mon]$ /home/ec2-user/aws-scripts-mon
-bash: /home/ec2-user/aws-scripts-mon: Is a directory
[ec2-user@ip-172-31-40-161 aws-scripts-mon]$ /home/ec2-user/aws-scripts-mon
-bash: /home/ec2-user/aws-scripts-mon: Is a directory
[ec2-user@ip-172-31-40-161 aws-scripts-mon]$ /home/ec2-user/aws-scripts-mon./mon
-put-instance-data.pl --mem-used-incl-cache-buff --mem-util --mem-used --mem-ava
il
-bash: /home/ec2-user/aws-scripts-mon./mon-put-instance-data.pl: No such file or
 directory
[ec2-user@ip-172-31-40-161 aws-scripts-mon]$ crontab -e
no crontab for ec2-user - using an empty one

[1]+  Stopped                 crontab -e
[ec2-user@ip-172-31-40-161 aws-scripts-mon]$ pwd
/home/ec2-user/aws-scripts-mon
[ec2-user@ip-172-31-40-161 aws-scripts-mon]$ crontab -l
no crontab for ec2-user
[ec2-user@ip-172-31-40-161 aws-scripts-mon]$ crontab -e
no crontab for ec2-user - using an empty one
crontab: installing new crontab
"/tmp/crontab.ILXmpd":1: bad day-of-week
errors in crontab file, can't install.
Do you want to retry the same edit? yes
crontab: installing new crontab
"/tmp/crontab.ILXmpd":1: bad day-of-week
errors in crontab file, can't install.
Do you want to retry the same edit? no
```

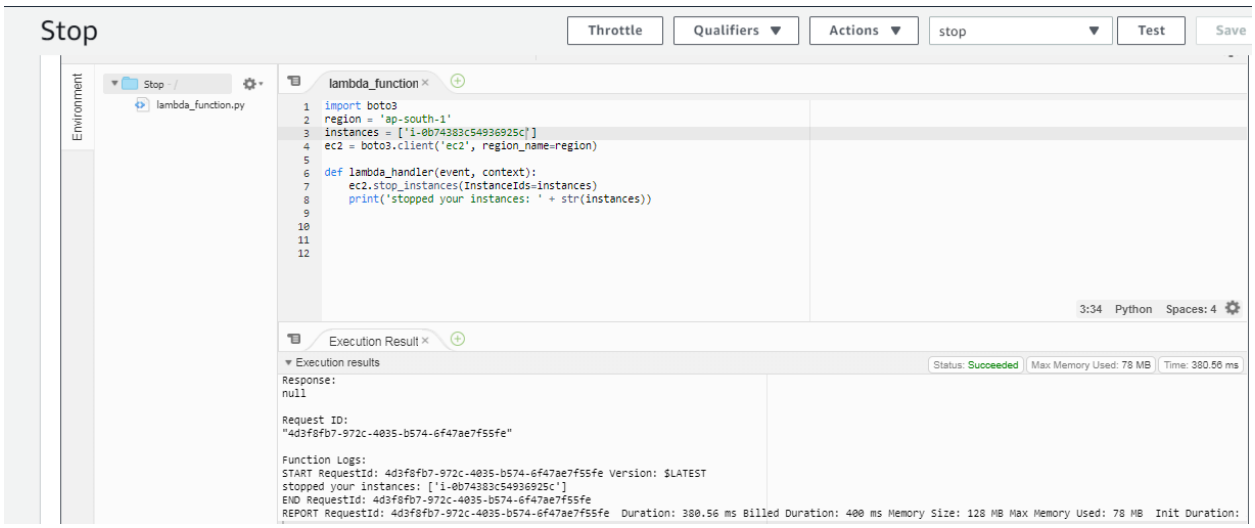

Untitled graph
1h 3h 12h 1d 3d 1w custom
Line
Actions
Apply time range
49
48.5
48.2
13:24:55
13:25:00
13:25:05
13:25:10
13:25:15
13:25:20
04-02 13:25
13:25:30
2020-04-02 13:25 UTC
1. MemoryUtilization 48.6
All metrics
Graphed me
Source
All > Linux System > InstanceId
Search for any metric, dimension or resource id
Graph search
Instance Name (3)
InstanceId
Metric Name
Clouudwatch
i-084e78b1884e0c878
MemoryUtilization
Clouudwatch
i-084e78b1884e0c878
MemoryUsed
Clouudwatch
i-084e78b1884e0c878
MemoryAvailable

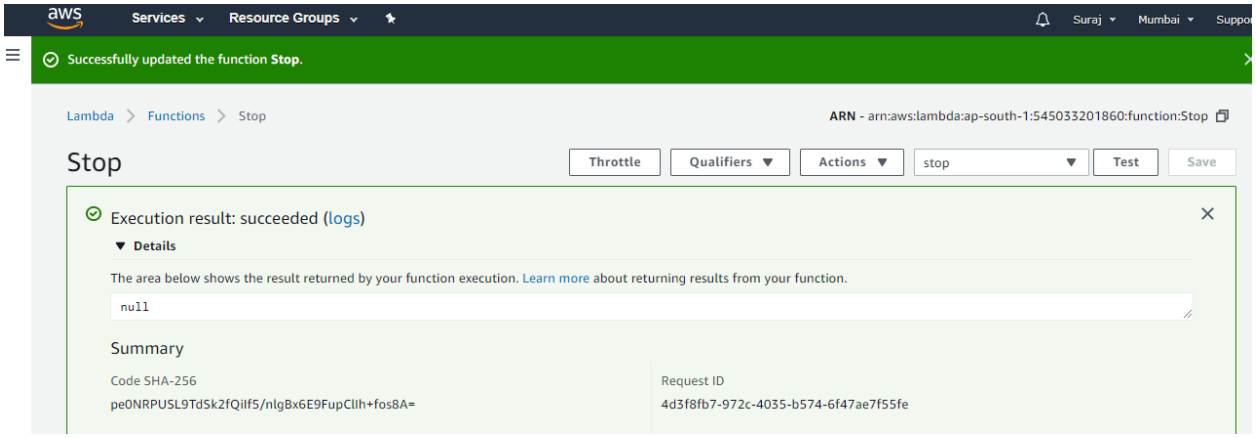

Untitled graph
1h 3h 12h 1d 3d 1w custom
Line
Actions
Apply time range
49
48.6
48.2
13:24:55
13:25:00
13:25:05
13:25:10
13:25:15
13:25:20
13:25:25
13:25:30
All metrics
Graphed metrics (1)
Graph options
Source
Math expression
Dynamic labels
Statistic: Average
Period: 1 Minute
Remove all
Label
Details
Statistic
Period
Y Axis
Actions
MemoryUtilization
Linux System • MemoryUtilization • InstanceId: i-084e78...
Average
1 Minute

Que. Create a lambda function to start, stop and terminate a running EC2 instance. Also, verify the logs generated by the lambda in the Cloudwatch logs to check whether the lambda execution is done properly

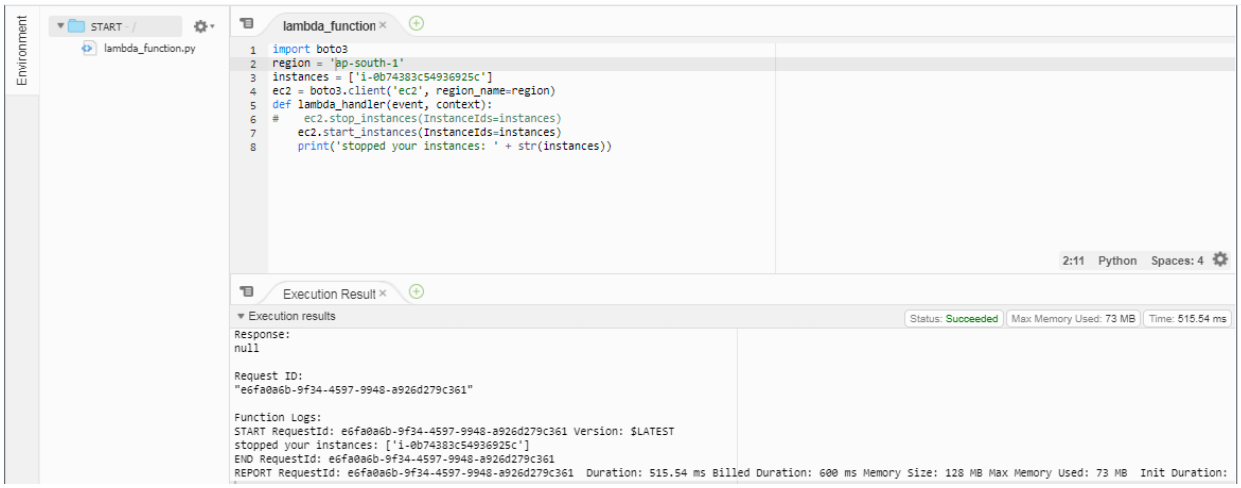
Ans.

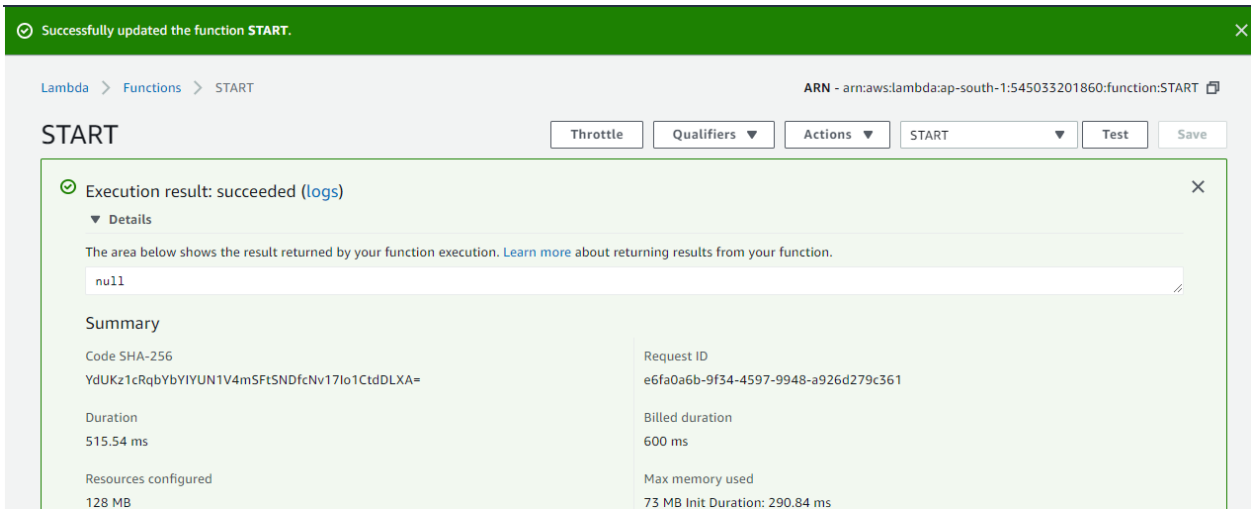
->Stop EC2 Instance Via Lamda





->Start EC2 Instance via Lamda

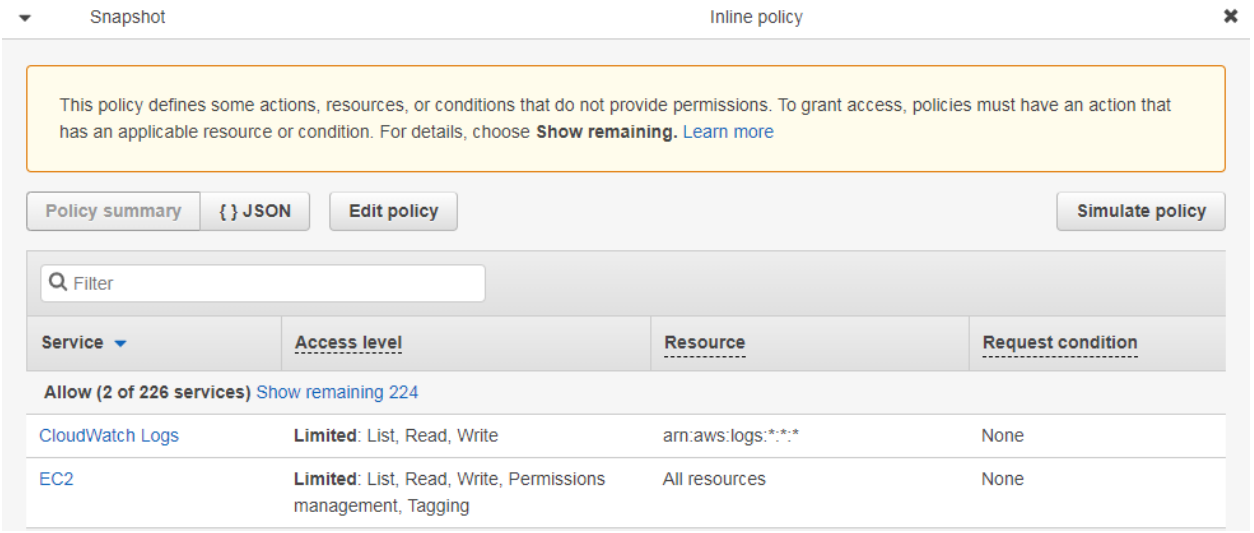




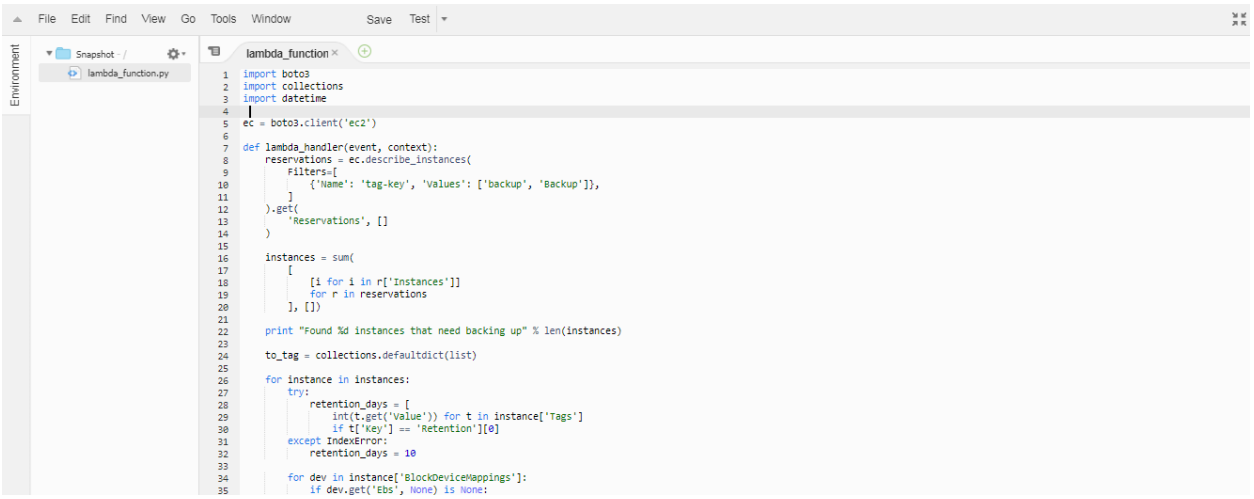
Que. When you terminate an EC2 instance, a snapshot should be created from the EBS volume attached to the EC2 instance automatically and store it in an S3 bucket. Create a lambda function to automate the above

Ans.Follwing are the steps to create Lamda Function for Ebs Snapshot-

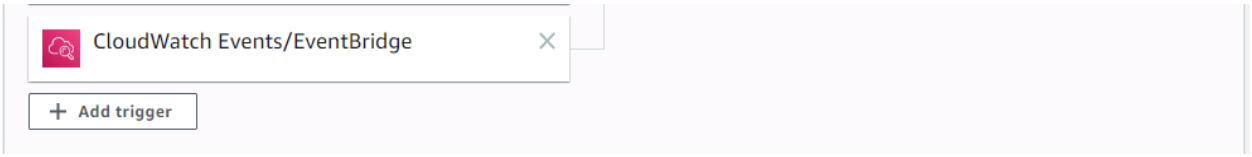
->Create IAM Role



->Create Snapshot Lamda Function



->Schedule Trigger as CloudWatch Rule



Output:>

Filter events | all | 2020-04-10 (19:00:04)

Time (UTC +00:00)	Message
2020-04-11	
	No older events found at the moment. Retry.
19:00:03	START RequestId: 02a8c930-2615-423d-8b45-f2ba21bd52cd Version: $LATEST
19:00:04	Found 2 instances that need backing up
19:00:04	Found EBS volume vol-06eab2de06864d7bc on instance i-08f87ef35457c9724
19:00:04	Retaining snapshot snap-0c536d380d958b78c of volume vol-06eab2de06864d7bc from instance i-08f87ef35457c9724 for 10 days
19:00:04	Found EBS volume vol-0fa33b1bac776e3bc on instance i-08f87ef35457c9724
19:00:04	Retaining snapshot snap-00b46f1c08cbe4c8f of volume vol-0fa33b1bac776e3bc from instance i-08f87ef35457c9724 for 10 days
19:00:04	Found EBS volume vol-0d5634127dbeaf42b on instance i-0b74383c54936925c
19:00:04	Retaining snapshot snap-0b970fc3fd5f1efd6 of volume vol-0d5634127dbeaf42b from instance i-0b74383c54936925c for 10 days
19:00:04	Will delete 3 snapshots on 2020-04-21
19:00:04	END RequestId: 02a8c930-2615-423d-8b45-f2ba21bd52cd
19:00:04	REPORT RequestId: 02a8c930-2615-423d-8b45-f2ba21bd52cd Duration: 1008.36 ms Billed Duration: 1100 ms Memory Size: 128 MB Max Memory Used: 88 MB Init Durati
	No newer events found at the moment. Retry.

Log output

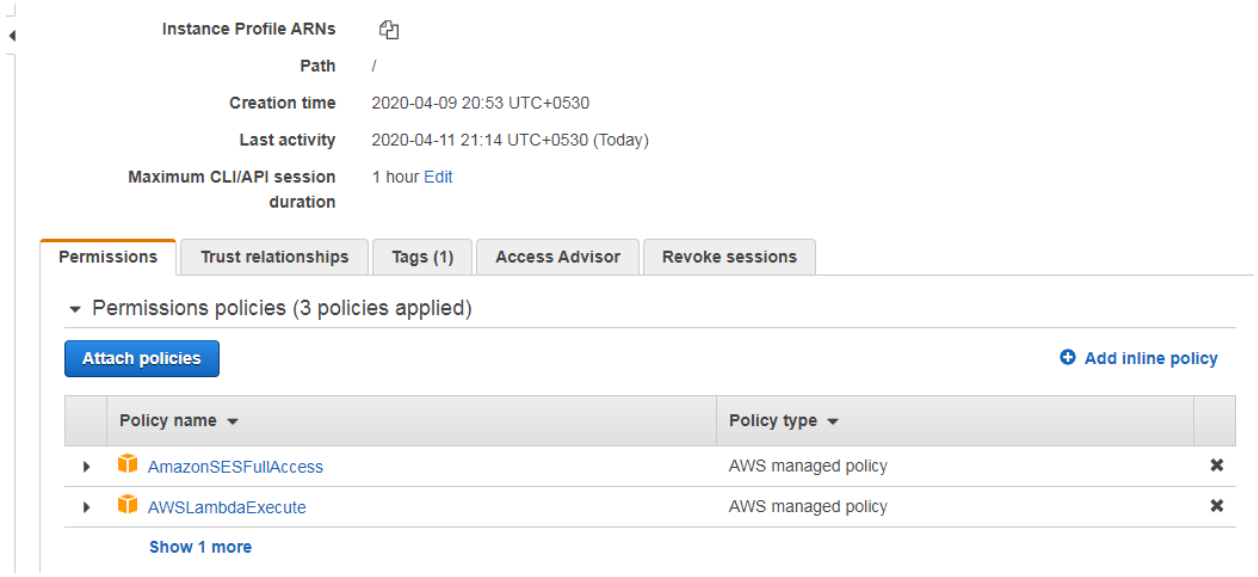
The section below shows the logging calls in your code. These correspond to a single row within the CloudWatch log group corresponding to this Lambda function. Click here to view the CloudWatch log group.

```
START RequestId: 02a8c930-2615-423d-8b45-f2ba21bd52cd Version: $LATEST
Found 2 instances that need backing up
Found EBS volume vol-06eab2de06864d7bc on instance i-08f87ef35457c9724
Retaining snapshot snap-0c536d380d958b78c of volume vol-06eab2de06864d7bc from instance i-08f87ef35457c9724 for 10
days
Found EBS volume vol-0fa33b1bac776e3bc on instance i-08f87ef35457c9724
Retaining snapshot snap-00b46f1c08cbe4c8f of volume vol-0fa33b1bac776e3bc from instance i-08f87ef35457c9724 for 10
days
Found EBS volume vol-0d5634127dbeaf42b on instance i-0b74383c54936925c
Retaining snapshot snap-0b970fc3fd5f1efd6 of volume vol-0d5634127dbeaf42b from instance i-0b74383c54936925c for 10
```

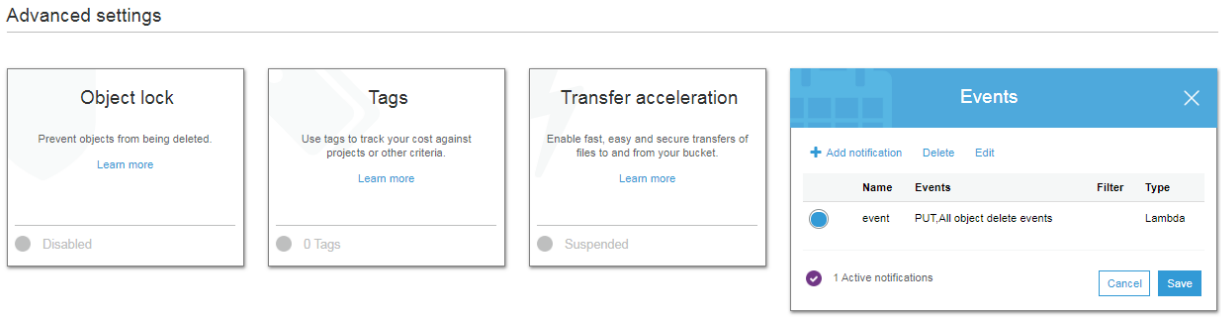
Que. Create a lambda function to send an email notification to your email id of your AWS account, as soon as you delete a file from an S3 bucket, mentioning the file name which is deleted.

Ans.Following are the steps for Lamda Function

->Create IAM Role



->Create Event in S3 Bucket

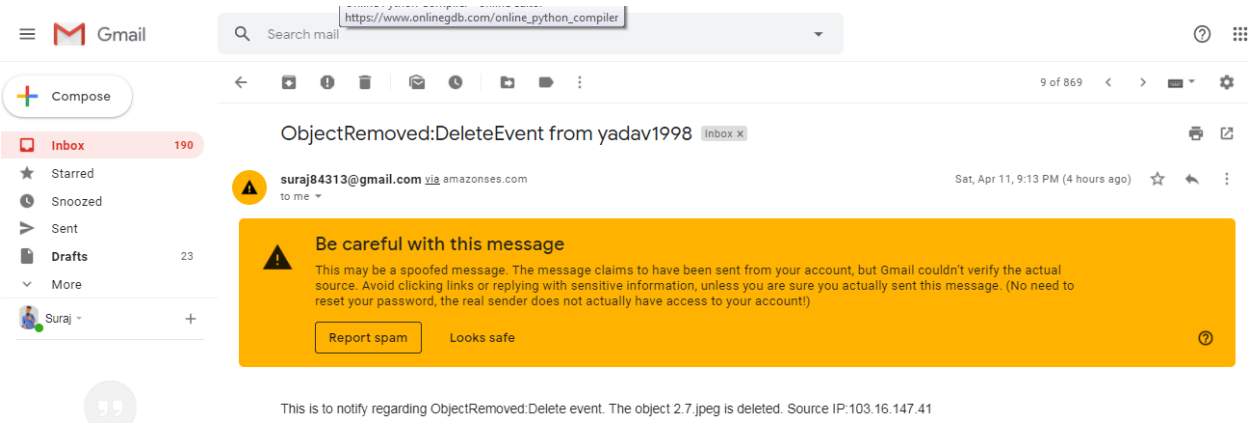


-> Create Lamda Function to Trigger Mail

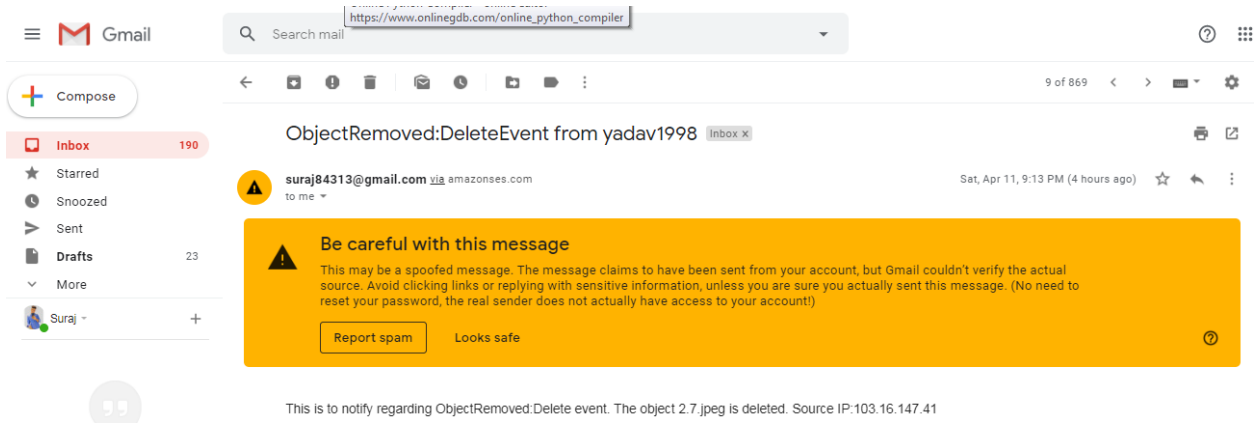
->Schedule Trigger as S3 Bucket

Output->

>When Object is Removed From Bucket



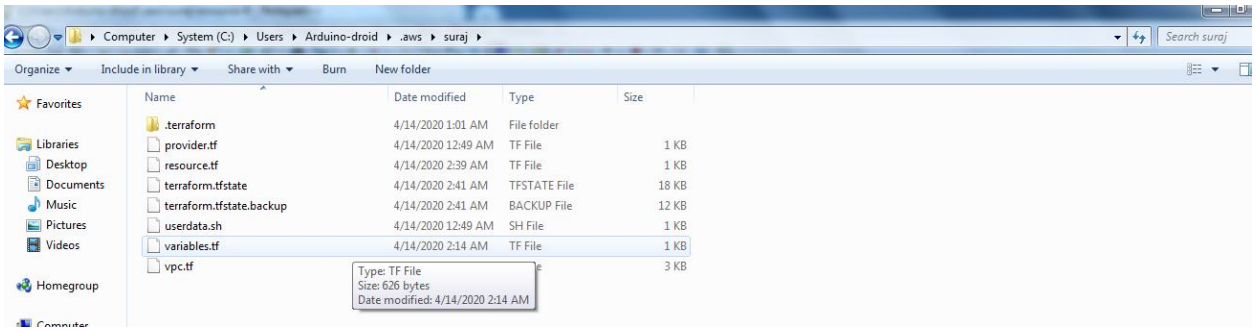
> When Object is Uploaded to Bucket



Que. Create a VPC with one public subnet, one private subnet. Create variable files to set CIDR range, vpc name, subnet names, subnet CIDR range etc. Create an EC2 instance on the public subnet of the above VPC, and install an apache webserver on the EC2 instance.

Create the above infrastructure using terraform scripts

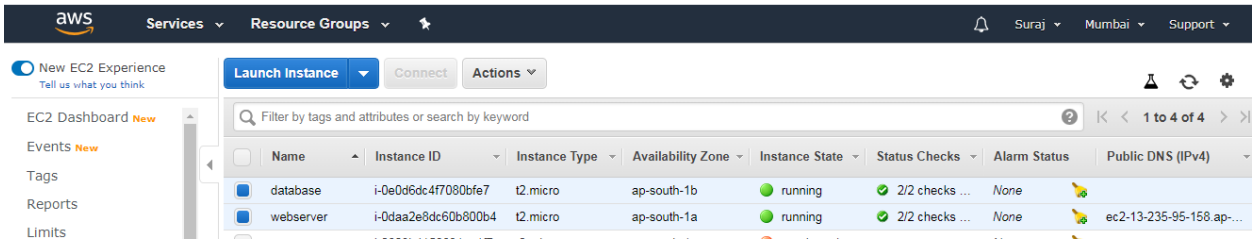
Ans.



```
          + volume_size           = (known after apply)
          + volume_type           = (known after apply)
        }
    }

  # aws_instance.wb will be created
  + resource "aws_instance" "wb" {
      + ami                          = "ami-0470e33cd681b2476"
      + arn                          = (known after apply)
      + associate_public_ip_address  = true
      + availability_zone            = (known after apply)
      + cpu_core_count               = (known after apply)
      + cpu_threads_per_core         = (known after apply)
      + get_password_data            = false
      + host_id                      = (known after apply)
      + id                           = (known after apply)
      + instance_state               = (known after apply)
      + instance_type                = "t2.micro"
      + ipv6_address_count           = (known after apply)
      + ipv6_addresses               = (known after apply)
      + key_name                     = (known after apply)
      + network_interface_id         = (known after apply)
      + password_data                = (known after apply)
      + placement_group              = (known after apply)
      + primary_network_interface_id = (known after apply)
      + private_dns                  = (known after apply)
      + private_ip                   = (known after apply)
      + public_dns                   = (known after apply)
      + public_ip                    = (known after apply)
      + security_groups              = (known after apply)
      + source_dest_check            = false
      + subnet_id                    = "subnet-0271579295f823e1e"
      + tags                         = {
          + "Name" = "webserver"
        }
      + tenancy                      = (known after apply)
      + user_data                    = "ad71878e45999d8d80fd59a0e63c80d53d4aa7b8"
      + volume_tags                  = (known after apply)
      + vpc_security_group_ids       = [
          + "sg-0295256f53f0fb9c0",
        ]

      + ebs_block_device {
          + delete_on_termination = (known after apply)
          + device_name           = (known after apply)
```


```
          + volume_size           = (known after apply)
          + volume_type           = (known after apply)
        }
    }

  # aws_instance.wb will be created
  + resource "aws_instance" "wb" {
      + ami                          = "ami-0470e33cd681b2476"
      + arn                          = (known after apply)
      + associate_public_ip_address  = true
      + availability_zone            = (known after apply)
      + cpu_core_count               = (known after apply)
      + cpu_threads_per_core         = (known after apply)
      + get_password_data            = false
      + host_id                      = (known after apply)
      + id                           = (known after apply)
      + instance_state               = (known after apply)
      + instance_type                = "t2.micro"
      + ipv6_address_count           = (known after apply)
      + ipv6_addresses               = (known after apply)
      + key_name                     = (known after apply)
      + network_interface_id         = (known after apply)
      + password_data                = (known after apply)
      + placement_group              = (known after apply)
      + primary_network_interface_id = (known after apply)
      + private_dns                  = (known after apply)
      + private_ip                   = (known after apply)
      + public_dns                   = (known after apply)
      + public_ip                    = (known after apply)
      + security_groups              = (known after apply)
      + source_dest_check            = false
      + subnet_id                    = "subnet-0271579295f823e1e"
      + tags                         = {
          + "Name" = "webserver"
        }
      + tenancy                      = (known after apply)
      + user_data                    = "ad71878e45999d8d80fd59a0e63c80d53d4aa7b8"
      + volume_tags                  = (known after apply)
      + vpc_security_group_ids       = [
          + "sg-0295256f53f0fb9c0",
        ]

      + ebs_block_device {
          + delete_on_termination = (known after apply)
          + device_name           = (known after apply)
```

```
Warning: Interpolation-only expressions are deprecated

  on resource.tf line 10, in resource "aws_instance" "wb":
  10:     user_data = "${file("userdata.sh")}"

Terraform 0.11 and earlier required all non-constant expressions to be
provided via interpolation syntax, but this pattern is now deprecated. To
silence this warning, remove the "${ sequence from the start and the }"
sequence from the end of this expression, leaving just the inner expression.

Template interpolation syntax is still used to construct strings from
expressions when the template includes multiple interpolation sequences or a
mixture of literal strings and interpolations. This deprecation applies only
to templates that consist entirely of a single interpolation sequence.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_instance.wb: Creating...
aws_instance.db: Creating...
aws_instance.wb: Still creating... [10s elapsed]
aws_instance.db: Still creating... [10s elapsed]
aws_instance.db: Still creating... [20s elapsed]
aws_instance.wb: Still creating... [20s elapsed]
aws_instance.db: Creation complete after 23s [id=i-0e0d6dc4f7080bfe7]
aws_instance.wb: Creation complete after 26s [id=i-0daa2e8dc60b800b4]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
PS C:\Users\Arduino-droid\.aws\suraj>
```



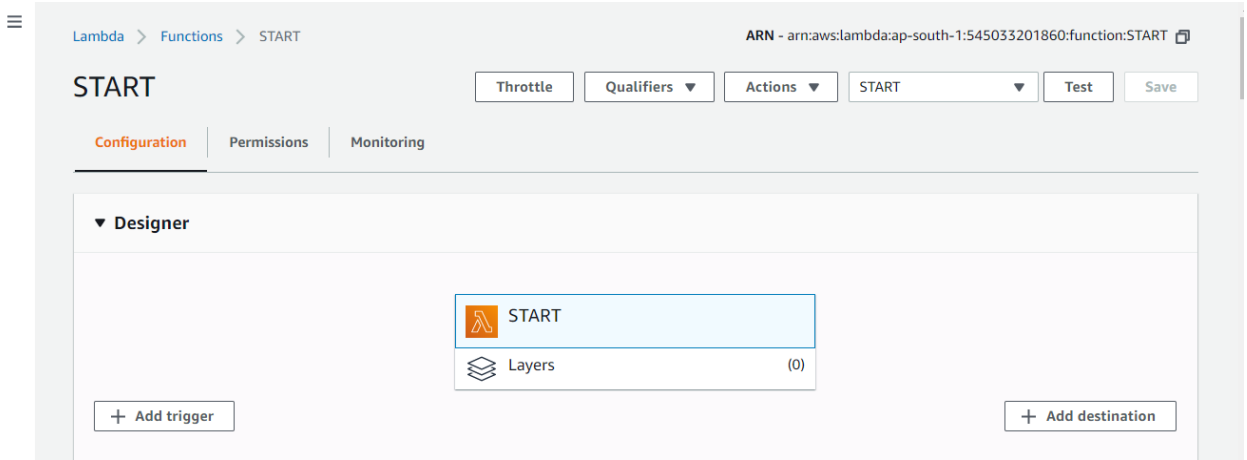
Que.Create a step function with the below states:

1. ***Starting state -> a pass type state with user input as 'start', or 'stop' or 'terminate' or 'exit'***
2. ***Choice state -> 3 choices : Based on the input value, it will execute 3 lambda functions (start ec2 lambda, storp ec2 lamba, terminate ec2 lambda).***
3. ***Also add a fail state which will be executed if you enter 'exit' in your input json.***
4. ***End state***

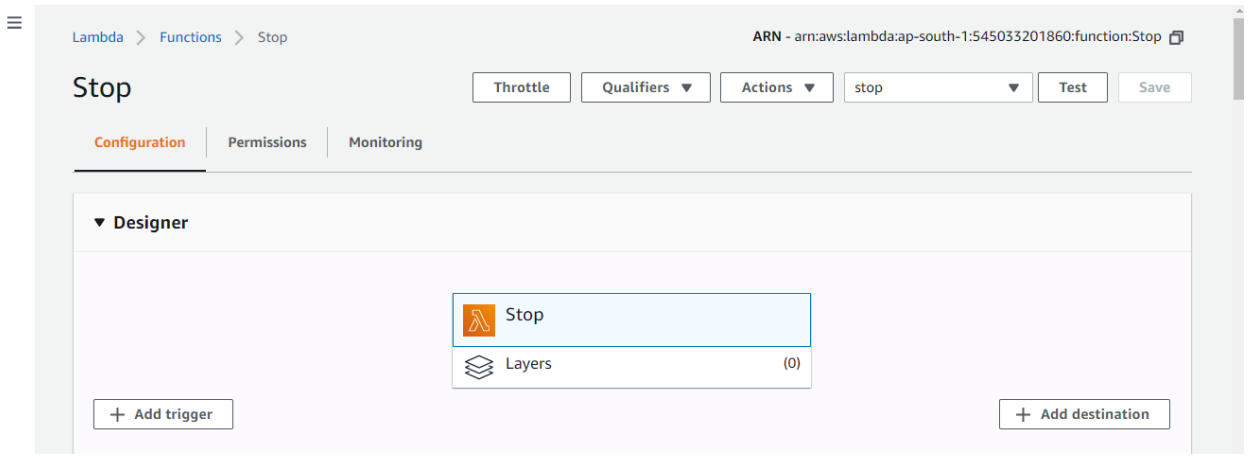
Ans*. Following are the steps to create step Function*

1.Crate Lamda Function to Start , Stop and Terminate Ec2 Instance

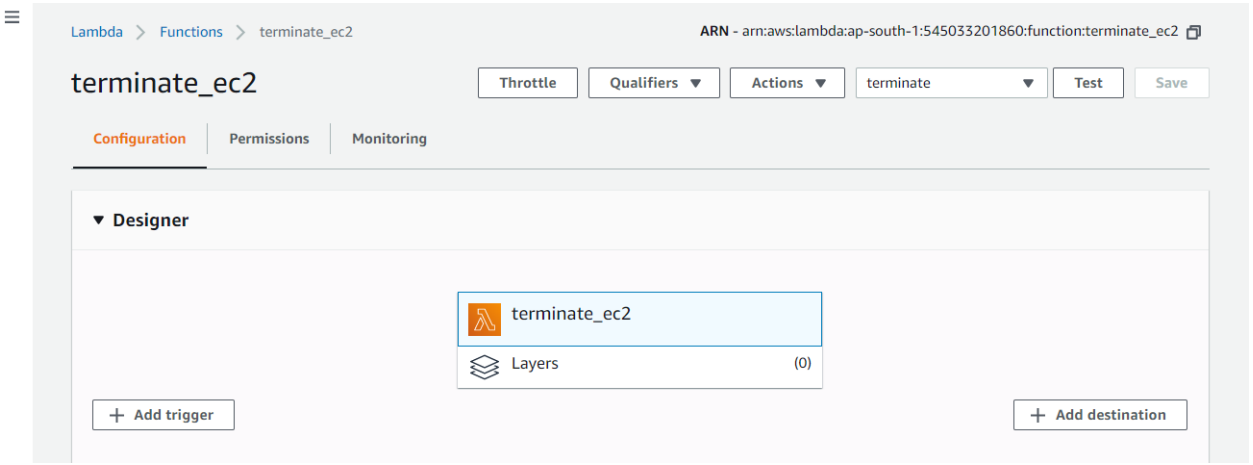
->Start Function



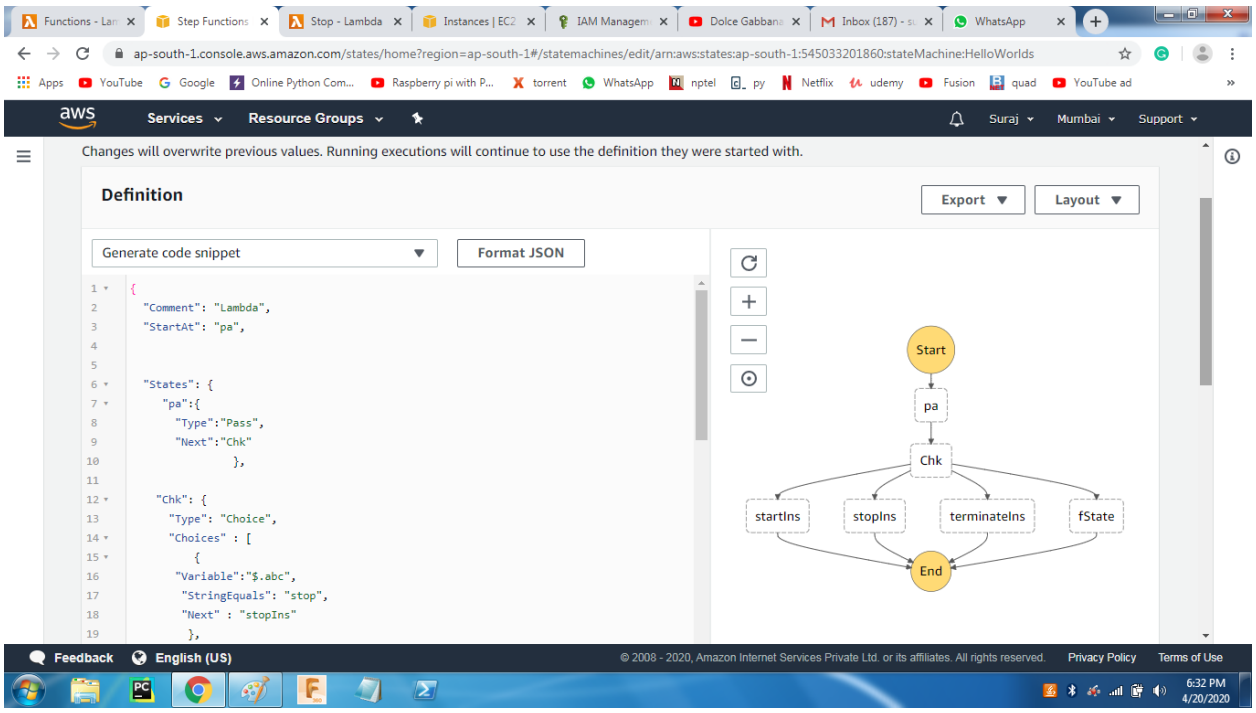
->Stop Function



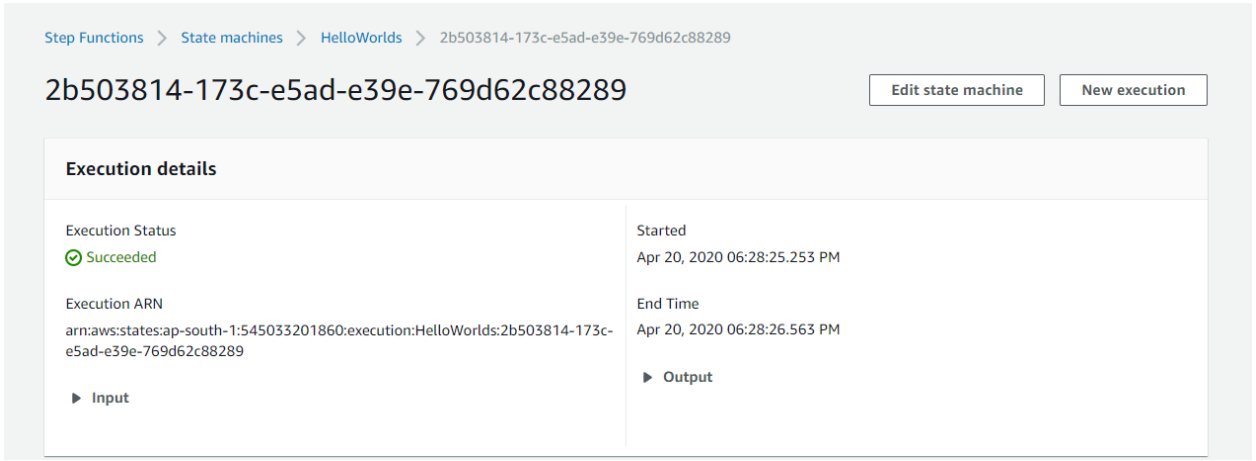
->Terminate Function

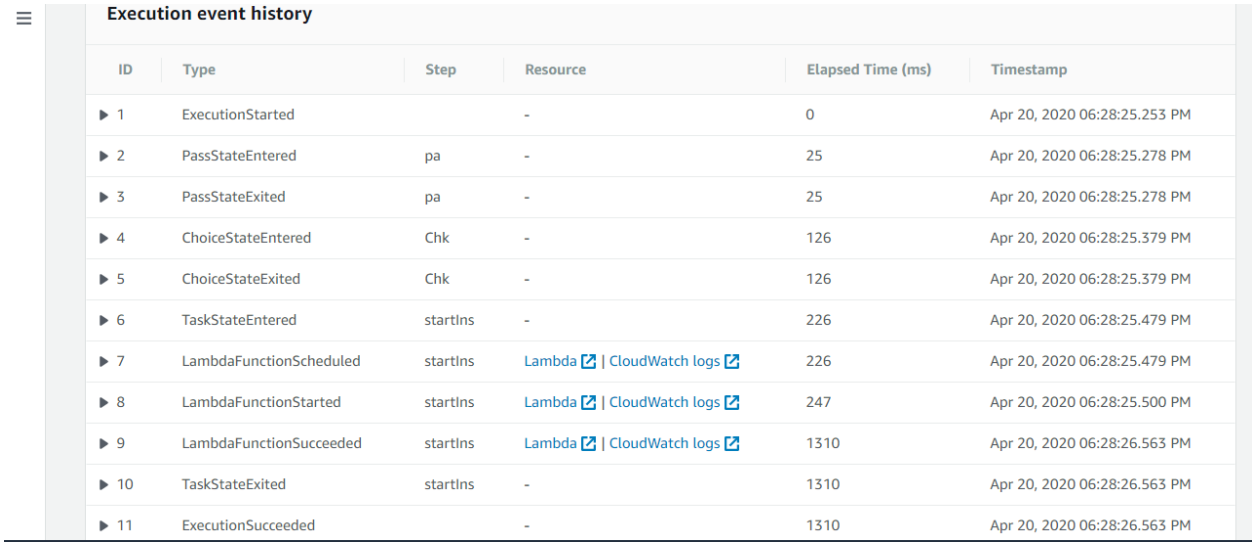


2.Execution of Step Function



(i)Start State

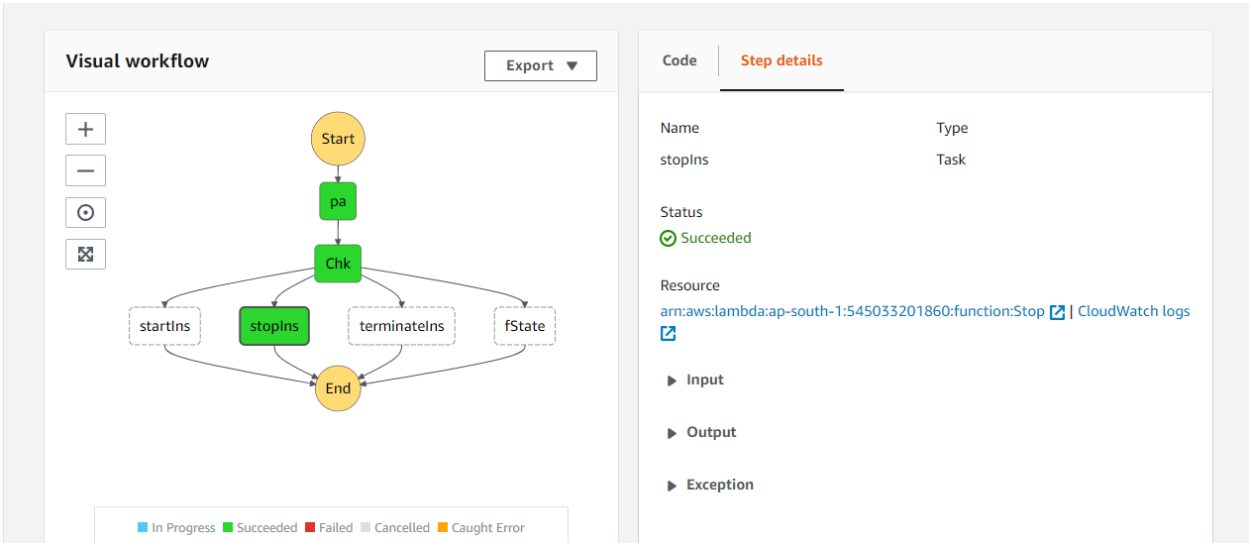




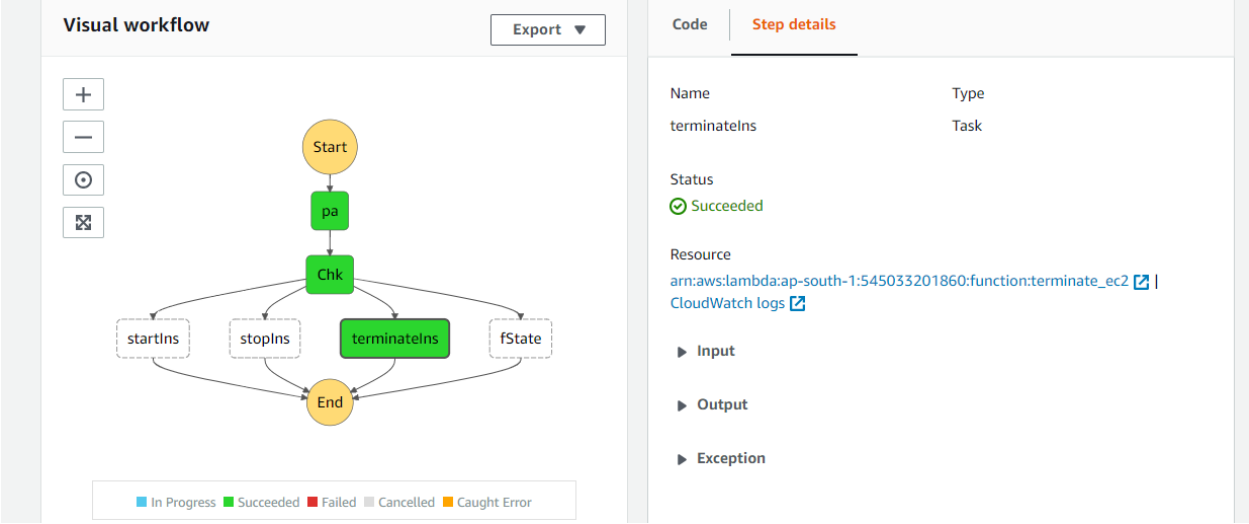
Execution event history

ID	Type	Step	Resource	Elapsed Time (ms)	Timestamp
1	ExecutionStarted		-	0	Apr 20, 2020 06:28:25.253 PM
2	PassStateEntered	pa	-	25	Apr 20, 2020 06:28:25.278 PM
3	PassStateExited	pa	-	25	Apr 20, 2020 06:28:25.278 PM
4	ChoiceStateEntered	Chk	-	126	Apr 20, 2020 06:28:25.379 PM
5	ChoiceStateExited	Chk	-	126	Apr 20, 2020 06:28:25.379 PM
6	TaskStateEntered	startIns	-	226	Apr 20, 2020 06:28:25.479 PM
7	LambdaFunctionScheduled	startIns	Lambda | CloudWatch logs	226	Apr 20, 2020 06:28:25.479 PM
8	LambdaFunctionStarted	startIns	Lambda | CloudWatch logs	247	Apr 20, 2020 06:28:25.500 PM
9	LambdaFunctionSucceeded	startIns	Lambda | CloudWatch logs	1310	Apr 20, 2020 06:28:26.563 PM
10	TaskStateExited	startIns	-	1310	Apr 20, 2020 06:28:26.563 PM
11	ExecutionSucceeded		-	1310	Apr 20, 2020 06:28:26.563 PM

(ii)Stop state



(iii)Terminate Ec2



(iv)Fail state