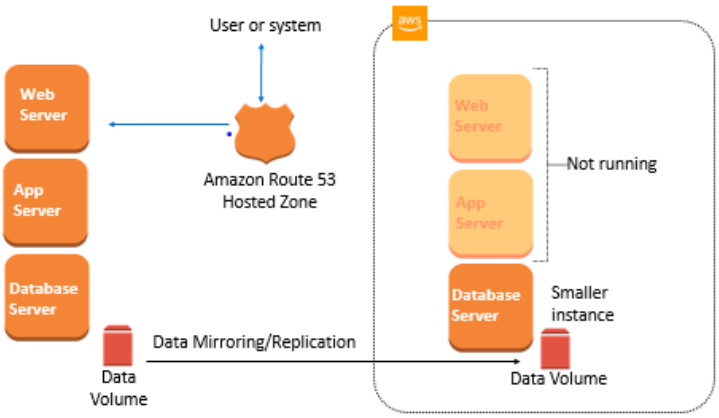
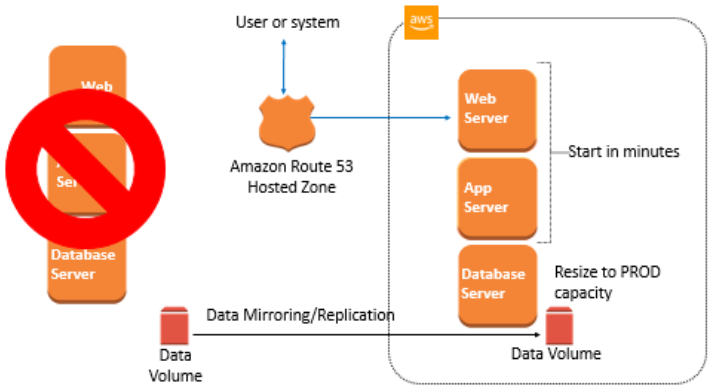
Pilot Light Architecture

In this lab, we are going to design a Disaster Recovery architecture. AWS has recommended four different types of architectures for Reliability which includes Backup & Restore, Pilot Light, Fully Working Low-Capacity Standby and Multi-Site Active-Active. In this lab, we are going to design a **Pilot Light** architecture for Disaster Recovery.

With reference to below image, let's consider that you've an on-premise Infrastructure which includes Web Server, App Server and Database Server. Consider that you've replicated the identical environment on AWS but it's not currently Powered ON except for the regular backup of Databases from on-prem Infrastructure to a DB Instance in AWS which is not running at the Production scale. Route 53 is currently pointing towards the on-premise Infrastructure.



Now with reference to below image, let's consider that the entire on-premise Infrastructure shuts down due to Power Supply issues or any catastrophic events. Now the Web Server and App Server Instances which are running on AWS gets powered ON and the Database Instance grows to the Production Capacity and Route 53 is now gets pointed to the AWS environment. Hence now your application is still Up and Running with the latest contents even though entire On-Premise Infrastructure is down.



This is called as Pilot Light Architecture for Disaster Recovery.

Below is the list of Tasks:

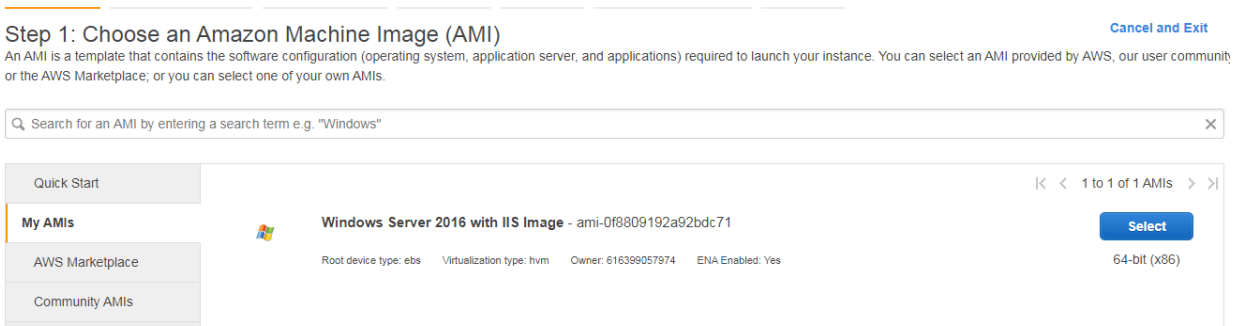
- Task 1: Launch EC2 Instances in two different Regions and allocate EIPs
- Task 2: Create a SNS Topic
- Task 3: Route 53 Configurations
- Task 4: Create an IAM Policy and Roles for Lambda Function
- Task 5: Configure a Lambda Function
- Task 6: Setup to test the Pilot Light Scenario
- Task 7: Test the Pilot Light Scenario

Task 1: Launch EC2 Instances in two different Regions and allocate EIPs

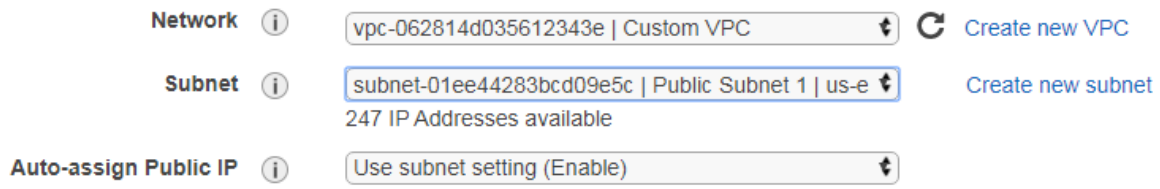
In this lab, we are going to consider **US East (N. Virginia) (us-east-1)** as the **Primary Region** and **Asia Pacific (Mumbai) (ap-south-1)** as the **Secondary Region**.

Navigate to AWS Management Console, select region as **US East (N. Virginia) (us-east-1)** and navigate to EC2 Service. Since in the Lab 2, we've created a template for the Windows Server 2016 IIS Server. Using that template launch a new EC2 Instance.

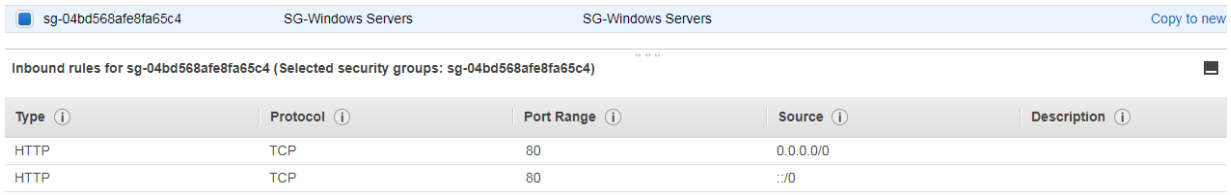
Click on Launch Instance, then click on My AMIs and select the below AMI.



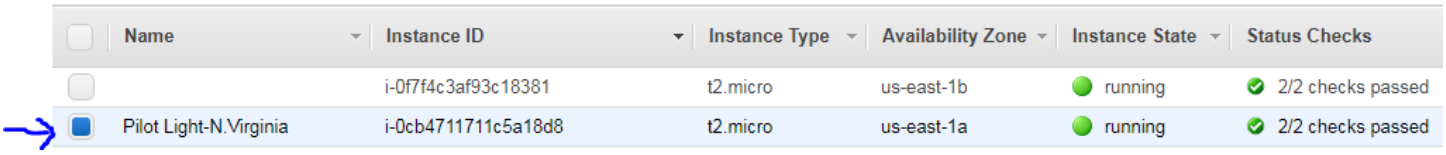
We are going to launch the EC2 Instance in our Original Architecture which was launched by CloudFormation. Make sure to select the Network as Custom VPC & Subnet as Public Subnet 1.



Select the Security Group that we've created for the Windows Server 2016 IIS Web Servers.



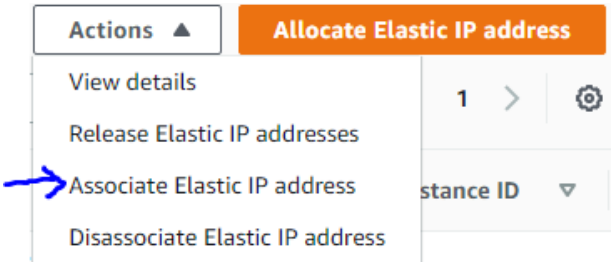
Choose an existing Key Pair and Click on launch EC2 Instance. Rename the Instance as per your accordance. Instance has been launched successfully.



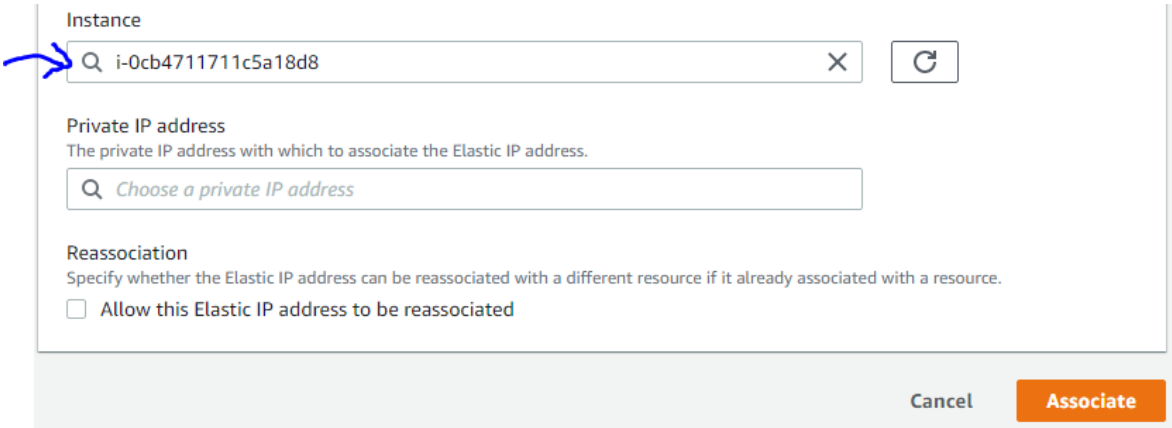
Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks
	i-0f7f4c3af93c18381	t2.micro	us-east-1b	running	2/2 checks passed
Pilot Light-N.Virginia	i-0cb4711711c5a18d8	t2.micro	us-east-1a	running	2/2 checks passed

Now click on Elastic IPs (EIP) and click on Allocate Elastic IP Address. Select the Amazon Pool of IPv4 Addresses and click on allocate.

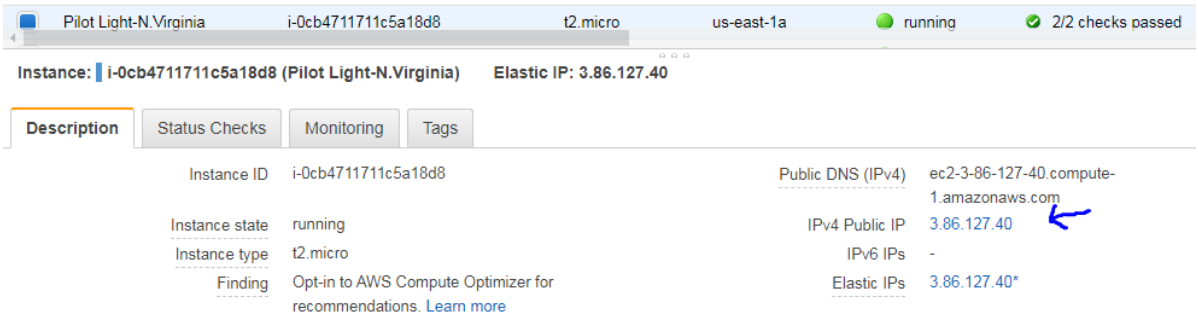
Select the Elastic IP Address and click on Actions and click on Associate Elastic IP Address.



Select the Instance from the drop-down menu to whom you want to allocate this Elastic IP Address and click on Allocate.

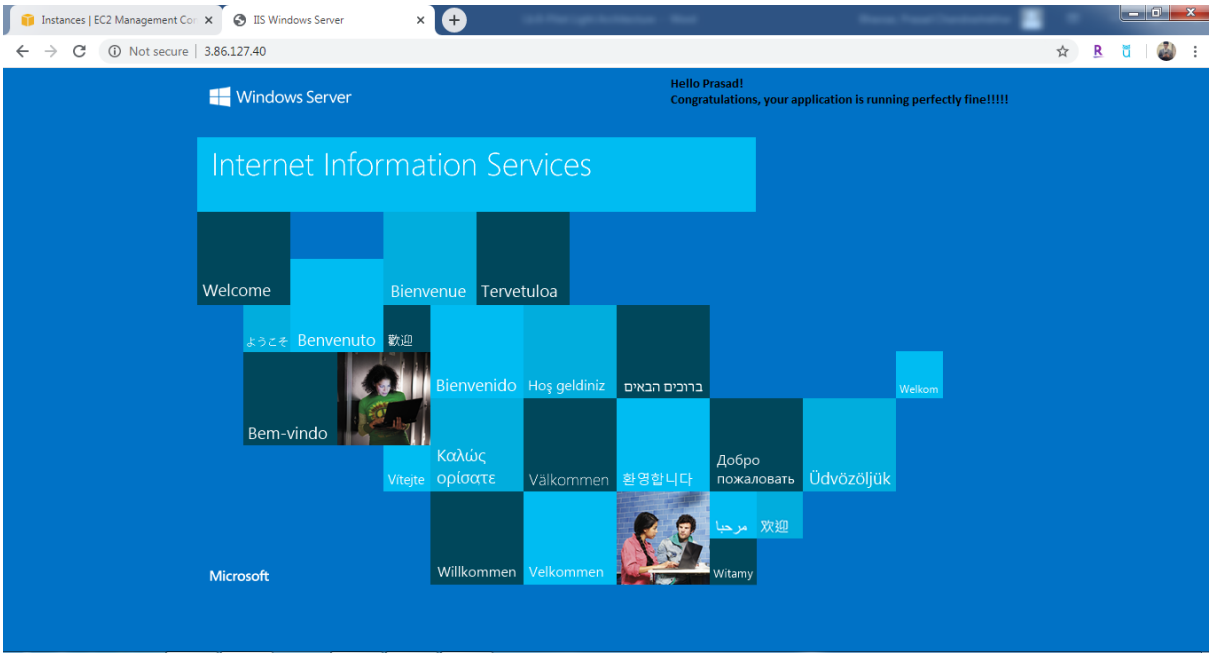


Your Instance is now configured with the Elastic IP Address.



Copy the Elastic IP Address and paste it in your browser.

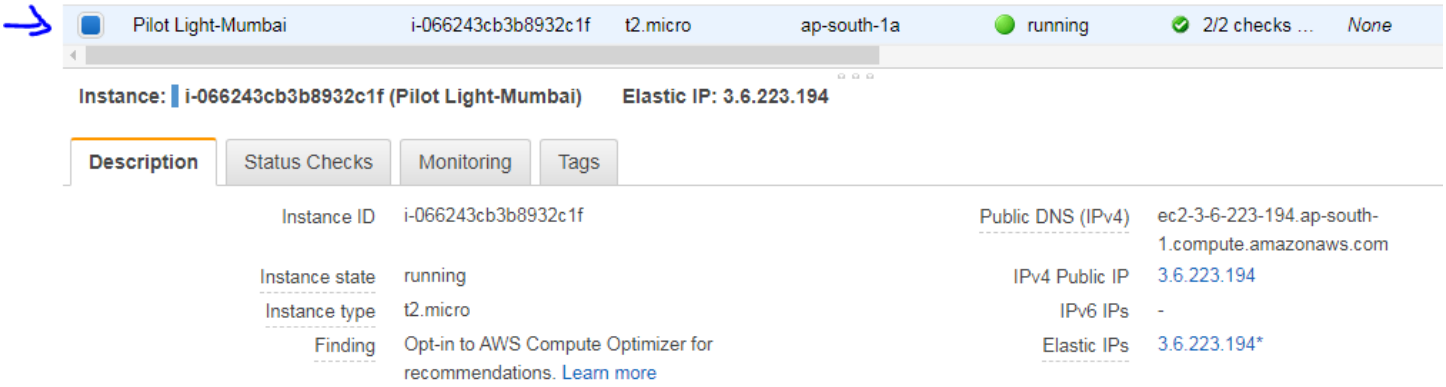
It should open the Webpage as follows.



Now switch it to the **Asia Pacific (Mumbai) (ap-south-1) Region**, and launch a new EC2 Instance.

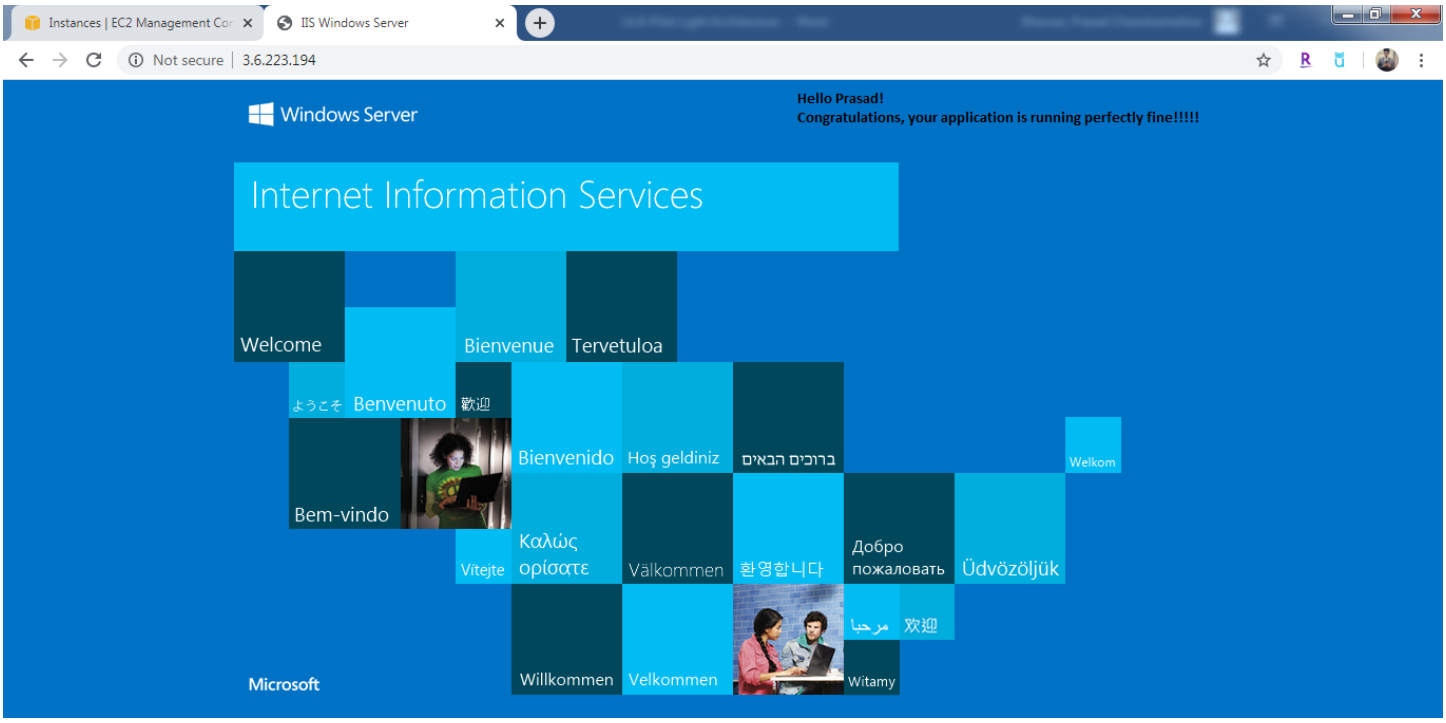
Launch this Instance using the same AMI Image which we've copied earlier in Lab 3. Make sure to select the VPC as Custom VPC and Subnet as Public Subnet. Allocate an Elastic IP Address and assign it to the EC2 Instance.

You've now successfully launched an EC2 Instance in **Asia Pacific (Mumbai) (ap-south-1) Region.**



Copy the Elastic IP Address and paste it in your browser.

It should open the Webpage as follows.



Finally, make a note of IP Addresses of EC2 Instances running in two different Regions.

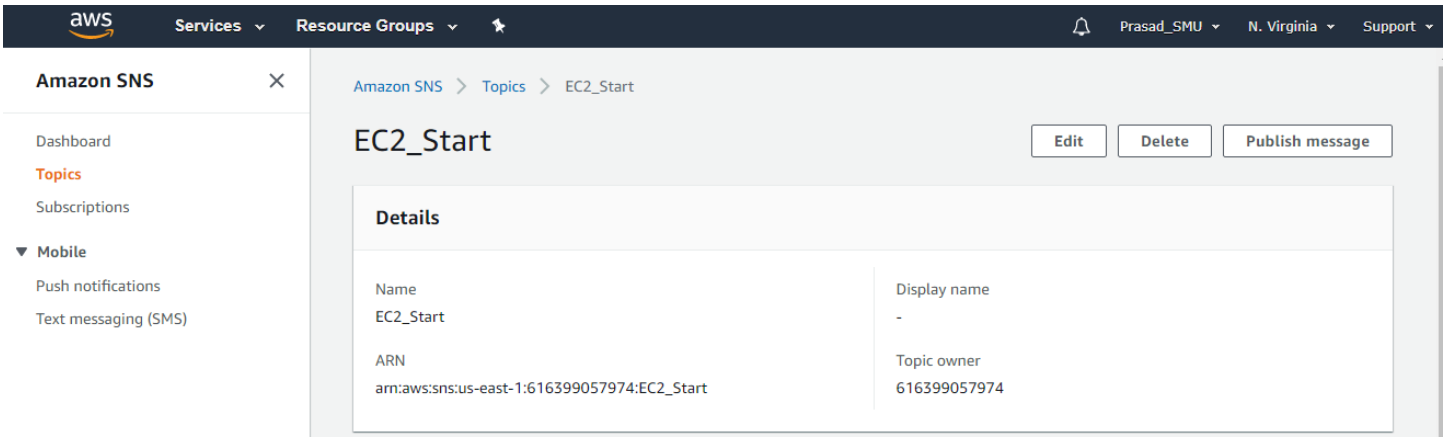
N. Virginia: 3.86.127.40

Mumbai: 3.6.223.194

Task 2: Create a SNS Topic

Come back to US East (N. Virginia) (us-east-1)

Navigate to Simple Notification Service and create a SNS Topic with the name of your choice.



Do not add any subscriptions for now.

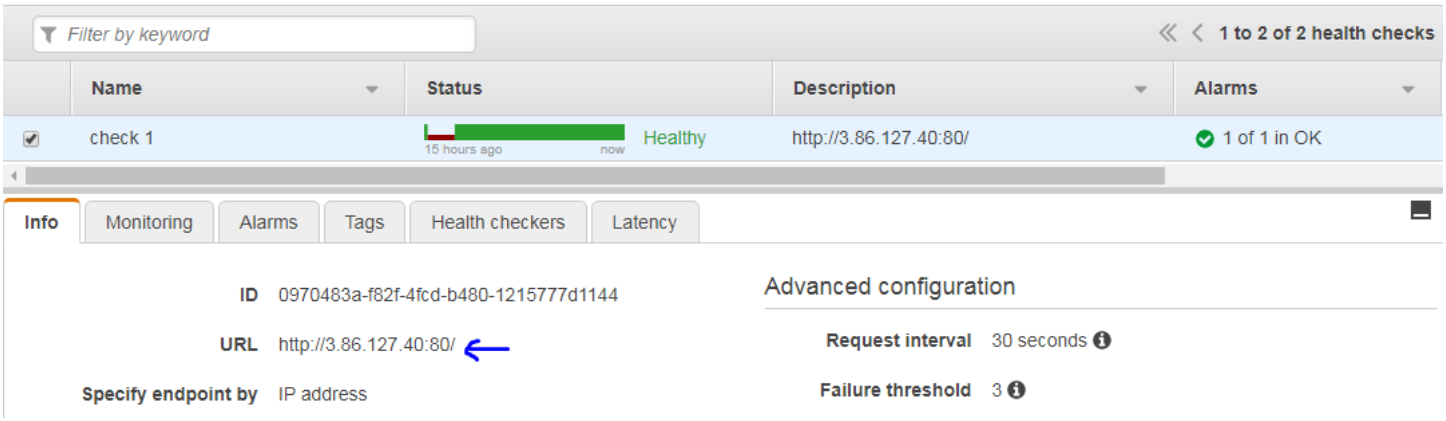
Task 3: Route 53 Configurations

Navigate to the Route 53 Service.

Click on the Hosted Zone that we've created i.e. SMU.EDU

Health Checks:

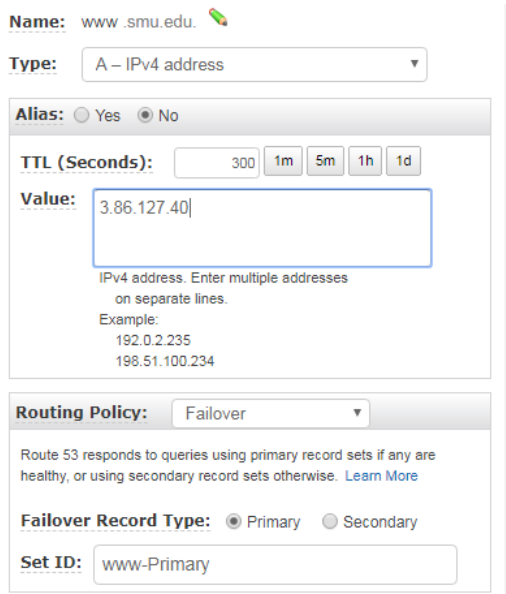
Click on Health Checks and create Health Check for our Primary EC2 Instance in N. Virginia region i.e. 3.86.127.40

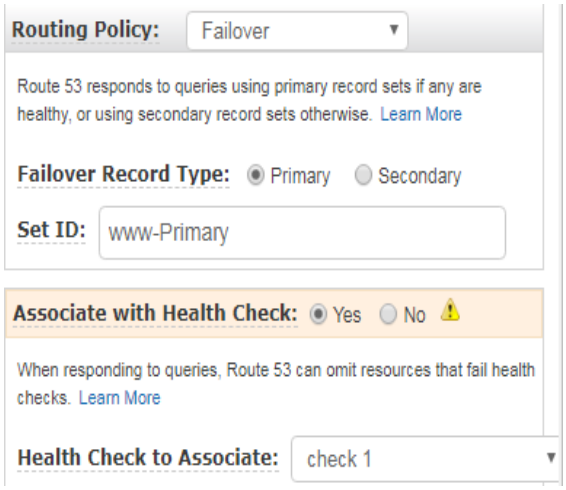


A-Records:

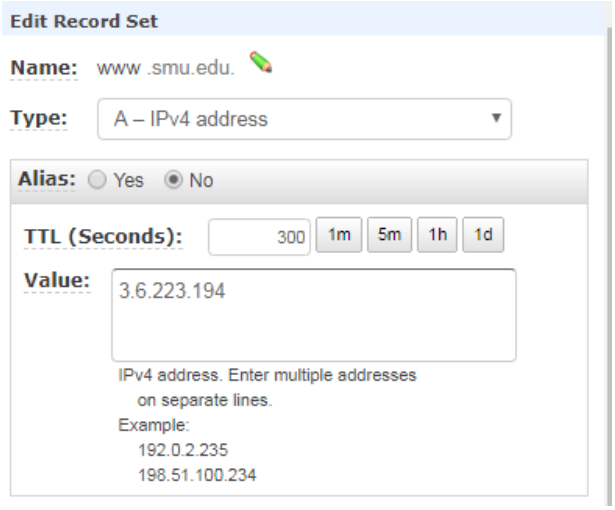
Now configure the Route 53 records with the Failover Policy.

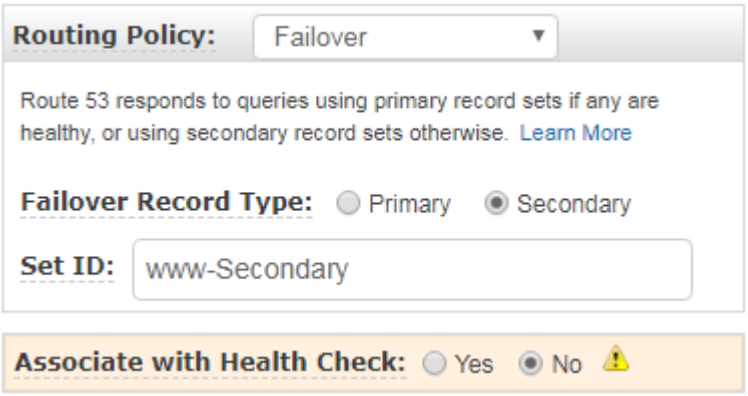
A Record for the Primary Region: Put value as IP Address of the EC2 Instance (3.86.127.40), Select the Routing Policy as Failover and Select the Health Check which you've configured.





A Record for the Secondary Region: Put value as IP Address of the EC2 Instance (3.6.223.194), Select the Routing Policy as Failover and no need to select the Health Check here.





Route 53 Configuration is now completed.

Task 4: Create an IAM Policy and Roles for Lambda Function

IAM Policy:

Navigate to the IAM Service., Click on Policies on the Left-Hand side and click on Create Policy.

Now click on JSON format and put the below code in the JSON Editor.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": "arn:aws:logs:*:*:*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:Start*"
            ],
            "Resource": "*"
        }
    ]
}
```

I'll give you this code in a .txt file.

Click on review the Policy, Give the Policy Name of your Choice.

At the bottom, look the resources on which this policy is going to give an access.

Filter

Service	Access level	Resource	Request condition
Allow (2 of 228 services) Show remaining 226			
CloudWatch Logs	**Limited**: Write	arn:aws:logs:*:*:*	None
EC2	**Limited**: Write	All resources	None

Finally, Click on Create Policy.

IAM Policy has been created successfully.

Filter policies Start

Policy name	Type	Used as	Description
StartEC2Instance	Customer managed	Permissions policy (1)	This policy starts an Instance

IAM Roles:

On the left-hand side, click on ROLES and click on Create Roles.

Select the Type of Trusted Entity as LAMBDA.

Common use cases

EC2
Allows EC2 instances to call AWS services on your behalf.

Lambda
Allows Lambda functions to call AWS services on your behalf.

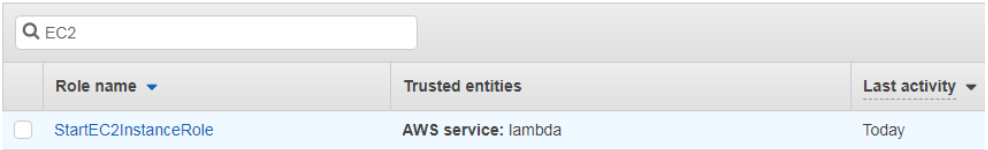
Under Permissions Policies, select the Policy that you've created.

Filter policies Start

Policy name
StartEC2Instance

Give the Role Name as per your choice and click on Create Role.

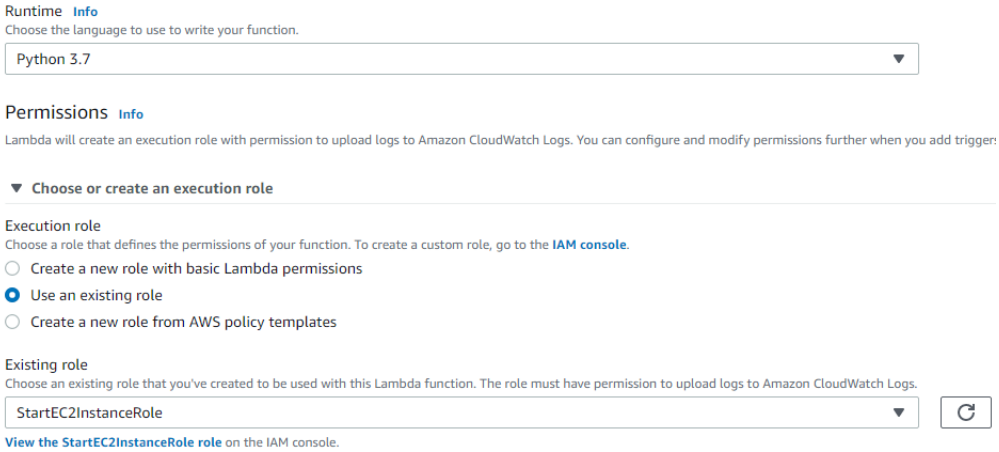
Role has been created Successfully.



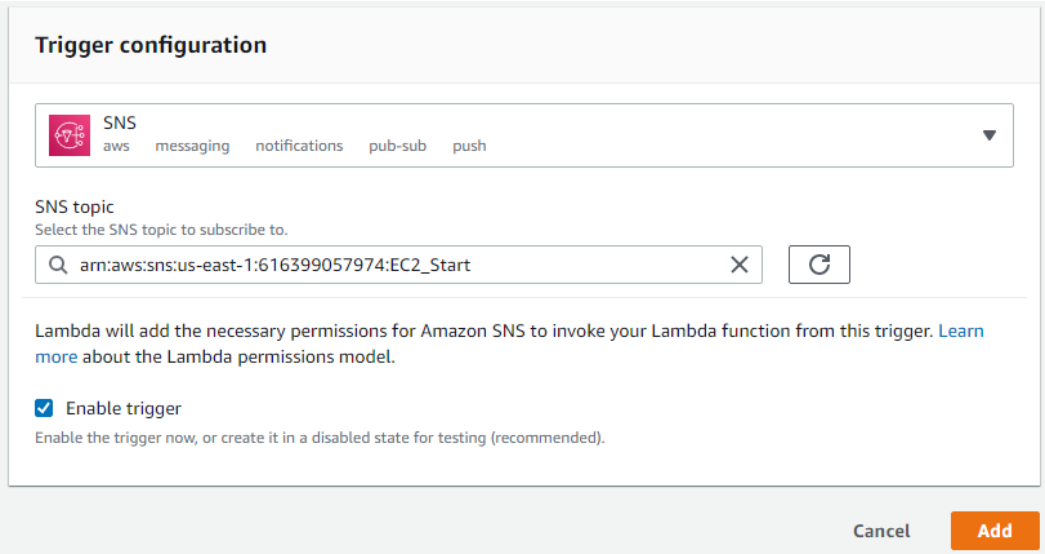
Task 5: Configure a Lambda Function

Navigate to Lambda Service and click on Create Function.

Give the Function Name of your Choice, select Runtime as Python 3.7 and select the existing Role which you've created.

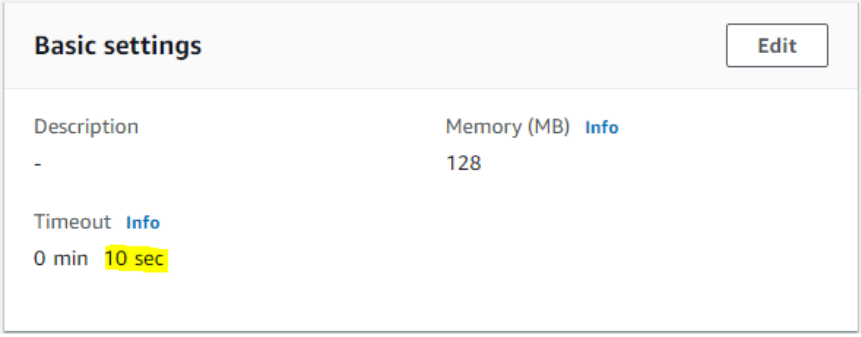


Once the Lambda Function is Created, op top left corner, click on Add Trigger. Select the SNS Service and select the Topic which we've created in Task 2 and click on Add.

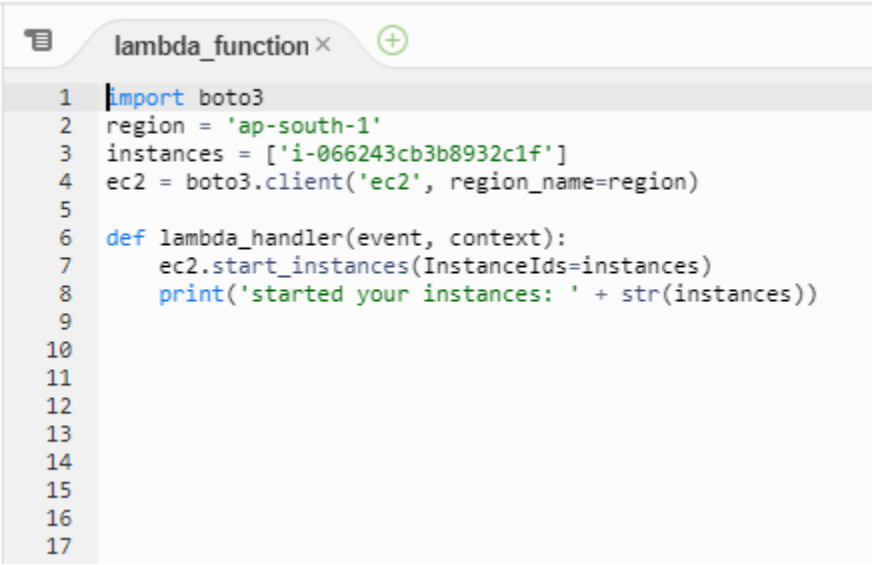


This adds Lambda as a Subscriber to the SNS Topic.

Now under basic settings, click on Edit and set the Timeout Intervals as 10 Seconds.



Now write down the Lambda Function Code as follows:



You just have to change the Instance ID in the Line Number 3. Copy the Instance ID of your EC2 Instance running in Mumbai (ap-south-1) region and Paste it in Line 3.

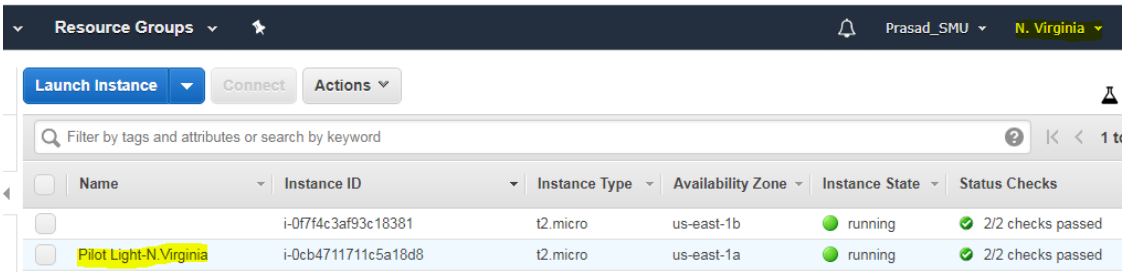
Finally, click on SAVE.

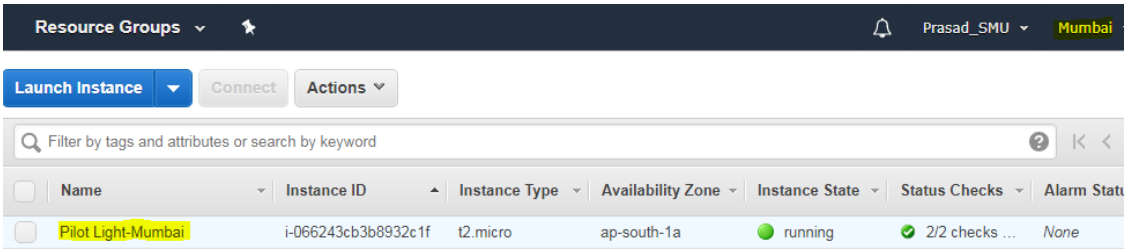
Lambda function is now configured properly.

Task 6: Setup to test the Pilot Light Scenario

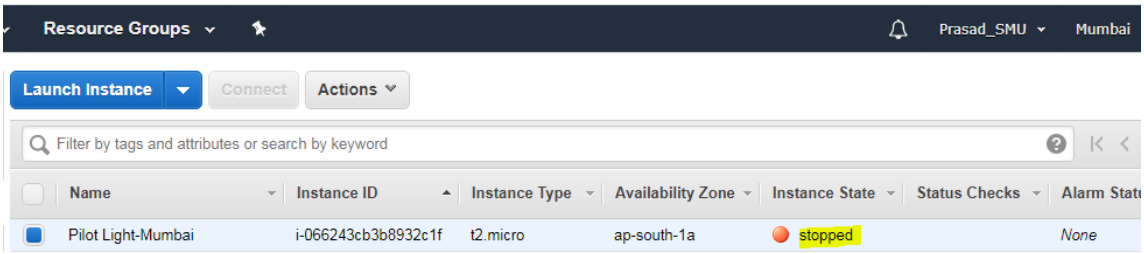
Follow the below steps to make setup to test the Pilot Light Scenario.

1. Verify the both the running EC2 Instances.





2. Let's Power OFF the EC2 Instance running in the Mumbai Region.



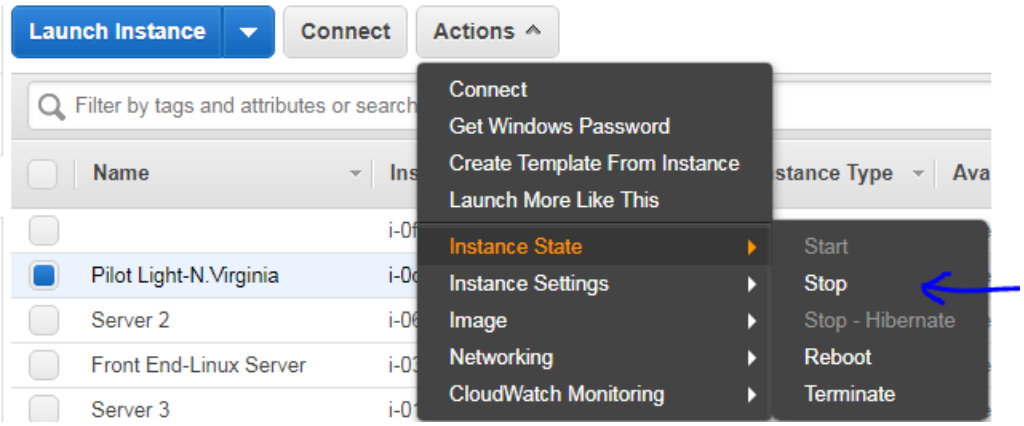
Task 7: Test the Pilot Light Scenario

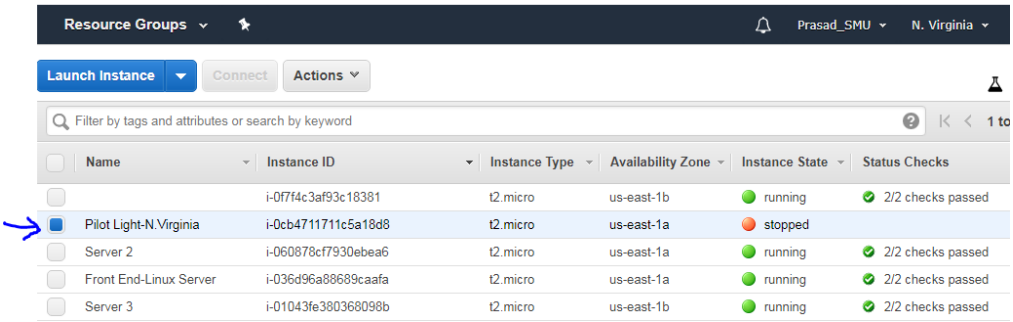
What's going to happen now?

If we Power OFF our Primary EC2 Instance which is running in the N. Virginia region, then our Secondary EC2 Instance which is currently Powered OFF in the Mumbai Region is going to get Powered ON and RUNNING.

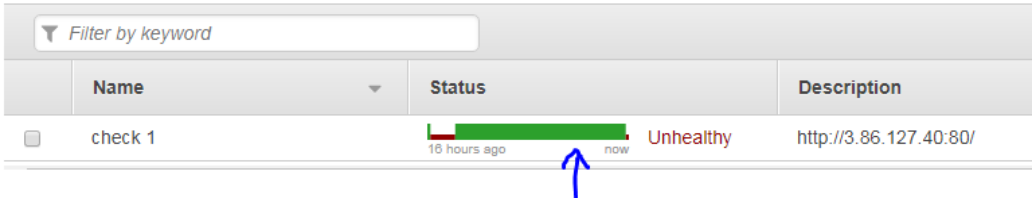
Let's verify it.

STOP the EC2 Instance which is running in the N. Virginia Region.





Let's look at the Route 53 Health Check.



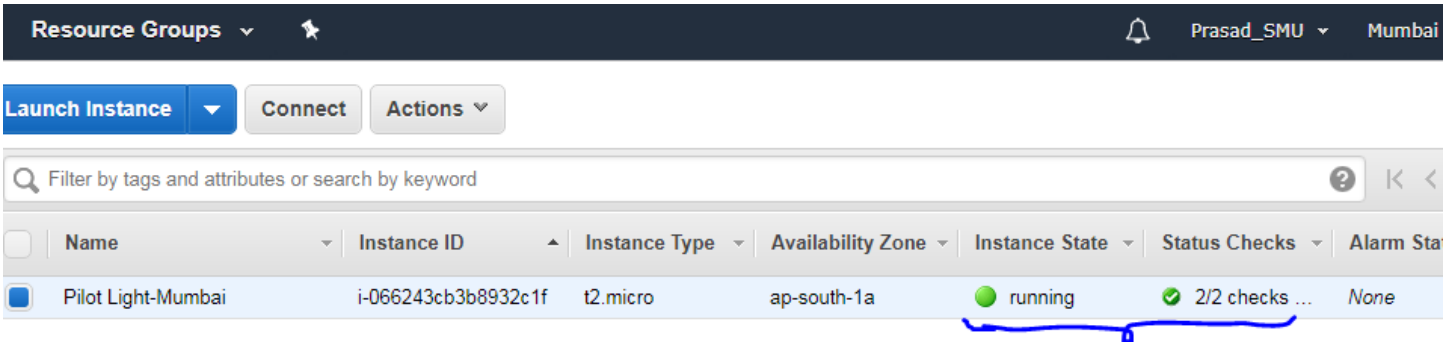
You can see that the Health Status is now UNHEALTHY.

This has now triggered the SNS Topic and since the SNS Topic has Lambda Subscription, it has triggered the Lambda Function.

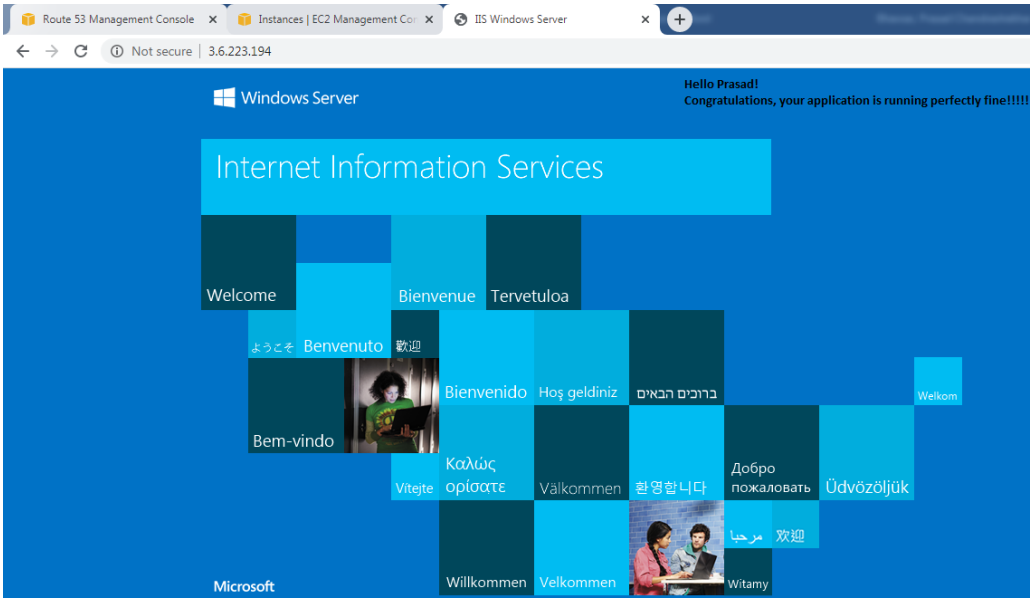
Lambda Function will now Power ON the EC2 Instance which is currently STOPPED in the Mumbai Region.

You can see that the EC2 Instance in Mumbai Region is now Initializing.

Now the EC2 Instance which is in Mumbai region is fully operational automatically.



Also, the Route 53 Failover Policy has now detected the Failover and routed the traffic towards the EC2 Instance in Mumbai Region.



This proves the Pilot Light Architecture.

For questions, contact me on pbhavsar@smu.edu .