

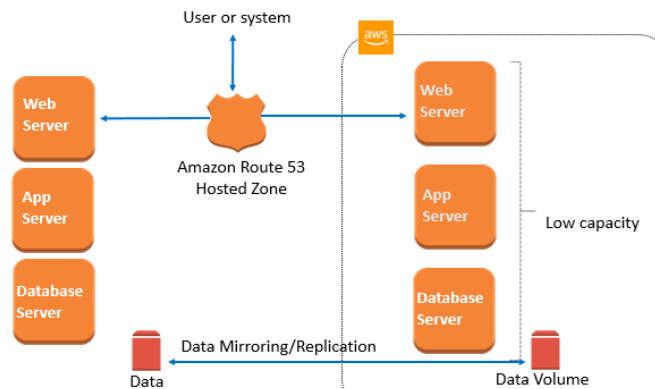
## Fully Working Low-Capacity Standby Architecture

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 **Fully Working Low-Capacity Standby** architecture for Disaster Recovery.

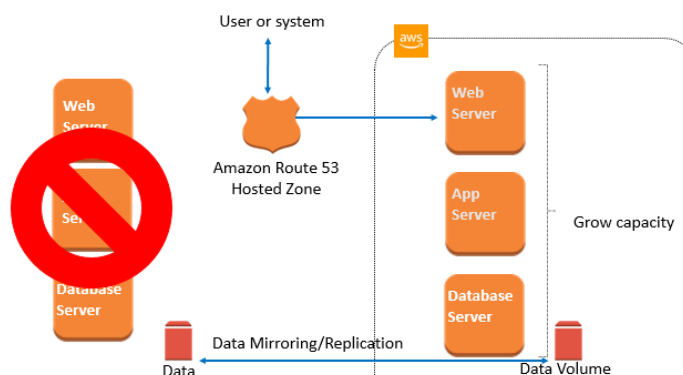
With reference to the below image, let's consider that you have an on-premise Infrastructure which includes Web Server, Application Server and Database Server. You've a similar environment on AWS but all the resources are running at low capacity. Also, there is a synchronous Data replication or mirroring is happening for the Databases from the On-premise Infrastructure to AWS EBS Volume. Lastly, the Route53 is primarily pointing towards the On-premise Infrastructure with Secondary Failover policy to the AWS environment.

### Can we achieve Load Balancing in this architecture?

Fully Working Low-Capacity Standby Architecture is designed for Reliability & High Availability. Since, the resources running on the AWS environment are not scale at the Production level rather they are running at the Low Capacity, hence it is not possible to achieve Load Balancing in this case.



With reference to the below image, if your entire Infrastructure shuts down due to Power Supply failures or any Catastrophic events, then we let the AWS resources to run at the Production level. It means we are scaling out our AWS environment. As the Route53 has the secondary Failover Policy, it will now point to the AWS environment. Finally, due to the latest backups of the on-premise Databases, your application will be still Up and Running with the latest web contents.



## Task 1: Inspect your AWS Environment

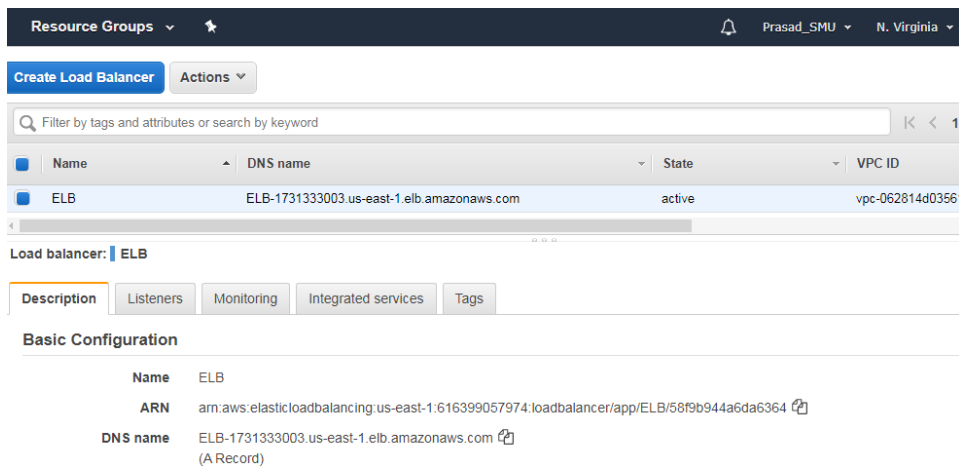
Since in the Lab 2-High Availability Across Availability Zone, we configured a Highly Available architecture across multiple Availability Zones. The entire architecture was hosted in US East (N. Virginia) (us-east-1) region. In this lab, we are going to consider this environment as the Secondary Environment.

Login to the AWS Management Console.

### Part 1:

Select the Region as Asia Pacific (Mumbai) (ap-south-1).

Navigate to the EC2 Service and click on the Load Balancer.



Resource Groups ▾ ⚙️ Prasad\_SMU ▾ N. Virginia ▾

Create Load Balancer Actions ▾

Filter by tags and attributes or search by keyword

Name	DNS name	State	VPC ID
ELB	ELB-1731333003.us-east-1.elb.amazonaws.com	active	vpc-062814d03561

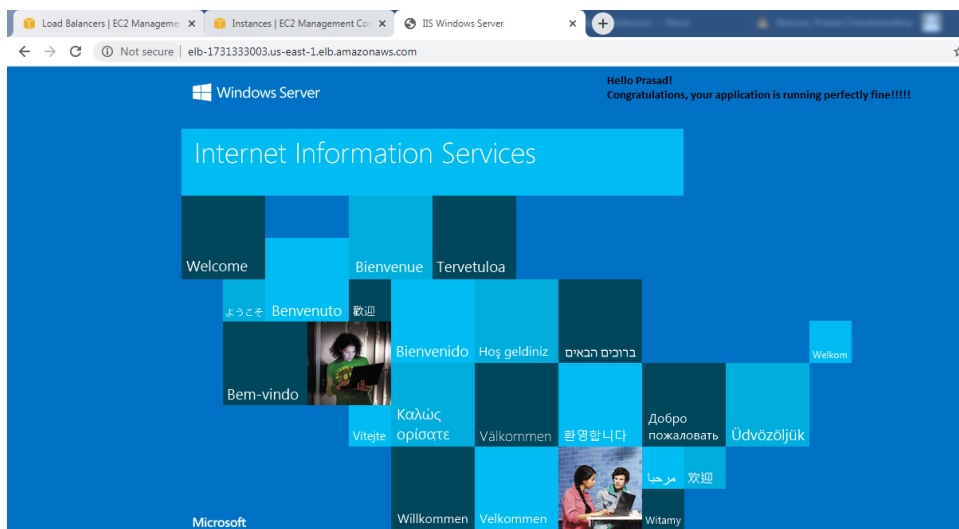
Load balancer: ELB

Description Listeners Monitoring Integrated services Tags

Basic Configuration

Name	ELB
ARN	arn:aws:elasticloadbalancing:us-east-1:616399057974:loadbalancer/app/ELB/58f9b944a6da6364
DNS name	ELB-1731333003.us-east-1.elb.amazonaws.com (A Record)

Your Load Balancer should be ACTIVE. Copy the DNS name and paste it in your browser. It should display your application's web page.



Now click on Target Group, you'll see the two EC2 Instances are running and load balancer is diverting the IIS traffic between these two EC2 Instances.

Resource Groups | Prasad\_SMU | N. Virginia | Support

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
TargetGroupofELB	80	HTTP	instance	ELB	vpc-062814d035612343e	

Target group: TargetGroupofELB

Description | **Targets** | Health checks | Monitoring | Tags

The load balancer starts routing requests to a newly registered target as soon as the registration process completes and the target passes the initial health checks. If demand on your targets increases, you can register additional targets. If demand on your targets decreases, you can deregister targets.

Edit

Registered targets

Instance ID	Name	Port	Availability Zone	Status	Description
i-07f4c3af93c18381		80	us-east-1b	healthy	This target is currently passing target group's health checks.
i-060878cf7930ebea6	Server 2	80	us-east-1a	healthy	This target is currently passing target group's health checks.

Now click on Autoscaling Group.

You'll notice that the Autoscaling has the Desired Capacity set to 2, Minimum capacity set to 2 and Maximum capacity set to 2. Minimum and Maximum Capacity is only considered while Scale-In and Scale-Out operations. As the Desired Capacity is set to 2, you noticed that there currently 2 Instances are running in Target Groups under the Load Balancer.

Resource Groups | Prasad\_SMU | N. Virginia | Support

Try the new design for Amazon EC2 Auto Scaling  
This older console is being replaced with the new EC2 Auto Scaling console. No new features or improvements will be made in this older console. [Go to the new console.](#)

Create Auto Scaling group | Actions

Filter: Filter Auto Scaling groups... | 1 to 1 of 1 Auto Scaling Groups

Name	Launch Configuration /	Instances	Desired	Min	Max	Availability Zones	Default Cooldown	Health Check Grac
ASG	LC	2	2	2	3	us-east-1a, us-east-1b	300	300

Keep in mind that this regional architecture (N. Virginia) is our Secondary Architecture.

**Part 2:**

Change Region to Asia Pacific (Mumbai) (ap-south-1). This is going to be our PRIMARY REGION.

Navigate to EC2 Service.

Since in the Lab 2, we've copied a template for the Windows Server 2016 IIS Server to Mumbai region. Using this template, we'll launch a new EC2 Instance now.

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

**Step 1: Choose an Amazon Machine Image (AMI)**[Cancel and Exit](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community or the AWS Marketplace; or you can select one of your own AMIs.

Quick Start
My AMIs
AWS Marketplace
Community AMIs

Windows Server 2016 with IIS Image - ami-0f809192a92bdc71

Root device type: ebs    Virtualization type: hvm    Owner: 616399057974    ENA Enabled: Yes

Select

64-bit (x86)

Make sure to select the Network as Custom VPC & Subnet as Public Subnet and Select the Auto-assign Public IP as Enable.

Network

vpc-00e090e78818d54a9 | Mumbai-Custom-VPC

Create new VPC

Subnet

subnet-0b6c4a15fb543fa4c | Public Subnet | ap-sout

Create new subnet

248 IP Addresses available

Auto-assign Public IP

Enable

Select the Security Group if you've created for the Windows Server 2016 IIS Web Servers. Else click Next.

sg-04bd568afe8fa65c4

SG-Windows Servers

SG-Windows Servers

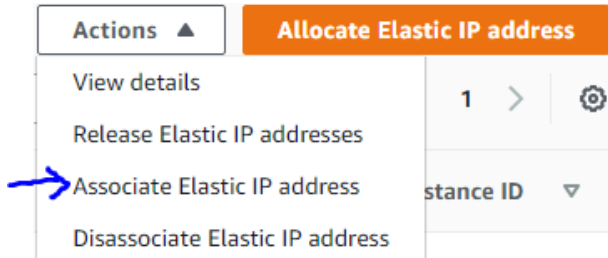
Copy to new

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

<input type="checkbox"/>	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status
<input type="checkbox"/>	test	i-0e11f191d80731b88	t2.micro	ap-south-1a	running	2/2 checks ...	None
<input checked="" type="checkbox"/>	Low Capacity Standby-Primary	i-066243cb3b8932c1f	t2.micro	ap-south-1a	running	2/2 checks ...	None

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.

The screenshot shows the 'Associate Elastic IP address' dialog box. It contains the following fields and options:

- Instance:** A search bar with the instance ID 'i-066243cb3b8932c1f' entered.
- Private IP address:** A section with the text 'The private IP address with which to associate the Elastic IP address.' and a search bar containing 'Choose a private IP address'.
- Reassociation:** A section with the text 'Specify whether the Elastic IP address can be reassociated with a different resource if it already associated with a resource.' and a checkbox labeled 'Allow this Elastic IP address to be reassociated'.
- Buttons:** 'Cancel' and 'Associate' buttons at the bottom right.

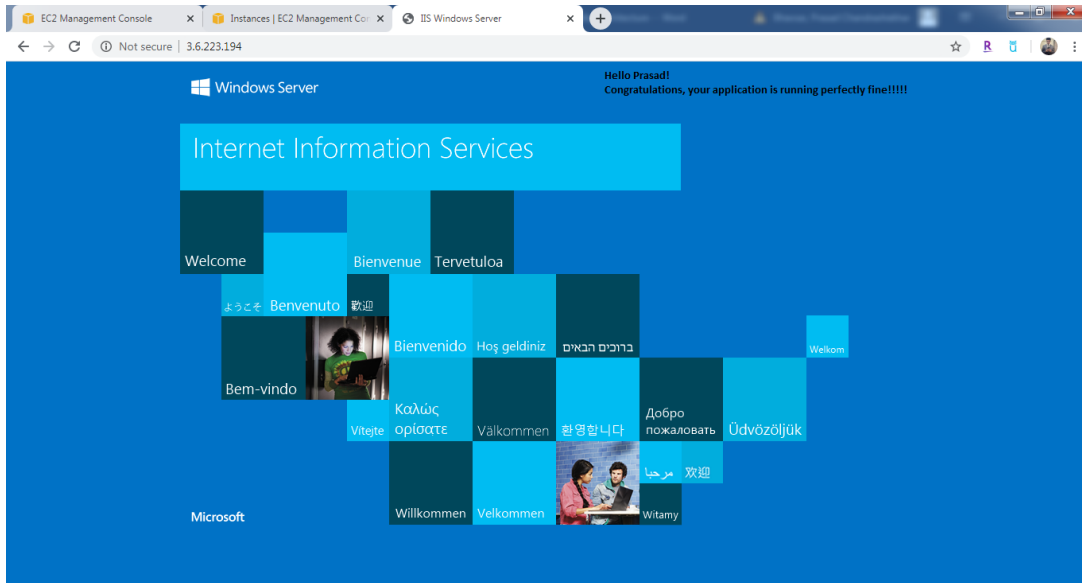
Your Instance is now configured with the Elastic IP Address.

The screenshot shows the AWS console details for an EC2 instance. The instance is named 'Low Capacity Standby-Primary' with ID 'i-066243cb3b8932c1f', type 't2.micro', and region 'ap-south-1a'. It is in a 'running' state. The Elastic IP address is '3.6.223.194'.

Instance: i-066243cb3b8932c1f (Low Capacity Standby-Primary)		Elastic IP: 3.6.223.194	
<b>Description</b>	<b>Status Checks</b>	<b>Monitoring</b>	<b>Tags</b>
Instance ID	i-066243cb3b8932c1f	Public DNS (IPv4)	ec2-3-6-223-194.ap-south-1.compute.amazonaws.com
Instance state	running	IPv4 Public IP	3.6.223.194
Instance type	t2.micro	IPv6 IPs	-
Finding	Opt-in to AWS Compute Optimizer for recommendations. <a href="#">Learn more</a>	Elastic IPs	3.6.223.194*

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

It should open the Webpage as follows.

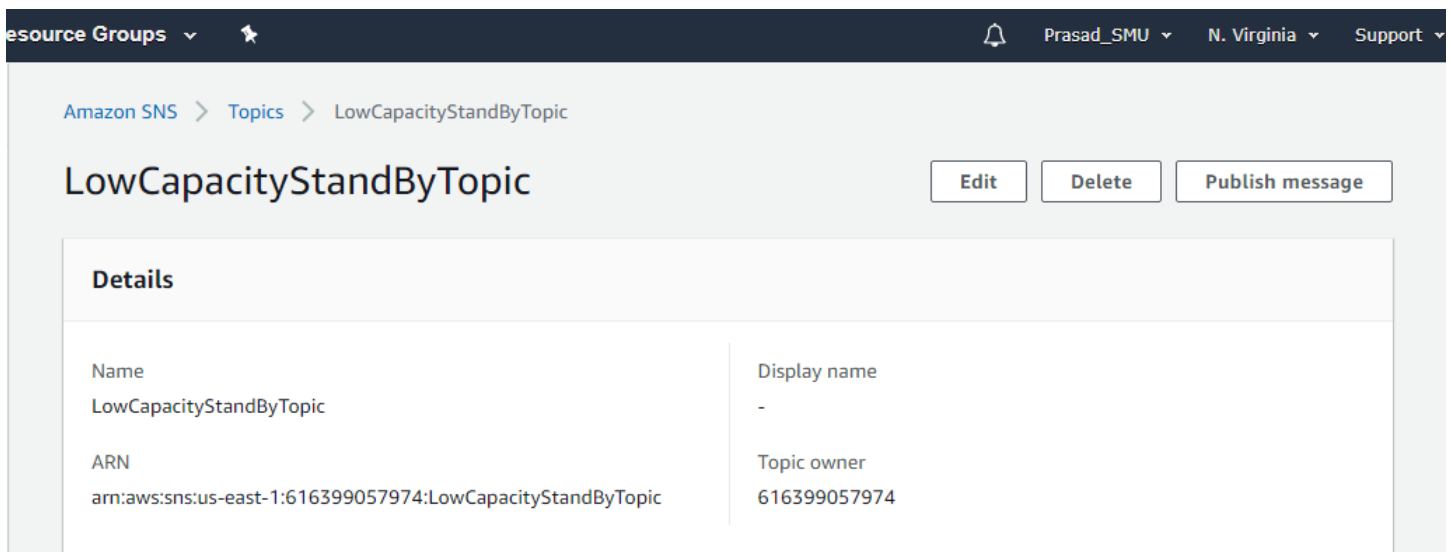


Keep in mind that this regional architecture (Mumbai) is our Primary Architecture.

## 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 Mumbai region i.e. 3.6.223.194. Make sure that Health Check sends Alarms to the SNS topic.

The screenshot shows the AWS Route 53 Health Checks console. At the top, there's a search bar with the text 'Filter by keyword' and a navigation bar with '<< < 1 to 1 of 1 health check'. Below this is a table with columns: Name, Status, Description, and Alarms. The table contains one entry: 'Low Capacity Check' with a status of 'Healthy' (indicated by a green bar and '18 hours ago' to 'now'), a description of 'http://3.6.223.194:80/', and '1 of 1 in OK'. Below the table are tabs for 'Info', 'Monitoring', 'Alarms', 'Tags', 'Health checkers', and 'Latency'. The 'Info' tab is selected, showing the 'ID' as 'c6996245-38e7-47e1-9af0-90af033502fc' and the 'URL' as 'http://3.6.223.194:80/'. To the right, under 'Advanced configuration', the 'Request interval' is set to '30 seconds'.


### 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 (3.6.223.194), Select the Routing Policy as Failover-Primary and Select the Health Check which you've configured.

The screenshot shows the 'Edit Record Set' configuration page in the AWS Route 53 console. The 'Name' field is 'www.smu.edu.' and the 'Type' is 'A - IPv4 address'. The 'Alias' is set to 'No'. The 'TTL (Seconds)' is set to '300'. The 'Value' field contains '3.6.223.194'. Below the value field, there's a note: 'IPv4 address. Enter multiple addresses on separate lines. Example: 192.0.2.235, 198.51.100.234'. On the right side, the 'Routing Policy' is set to 'Failover'. Below this, a description states: 'Route 53 responds to queries using primary record sets if any are healthy, or using secondary record sets otherwise. Learn More'. The 'Failover Record Type' is set to 'Primary'. The 'Set ID' is 'www-Primary'. Below that, 'Associate with Health Check' is set to 'Yes'. A note states: 'When responding to queries, Route 53 can omit resources that fail health checks. Learn More'. Finally, the 'Health Check to Associate' is set to 'Low Capacity Check'.

A Record for the Secondary Region: Put the Alias Target as Load Balancer's DNS Name. Select the Routing Policy as Failover-Secondary and no need to select the Health Check here.

Name:  

Type:

Alias: ☒ Yes ☐ No

Alias Target:

Alias Hosted Zone ID: Z35SXDOTRQ7X7K

You can also type the domain name for the resource. Examples:

- CloudFront distribution domain name: d1111111abcdef8.cloudfront.net
  - Elastic Beanstalk environment CNAME: example.elasticbeanstalk.com
  - ELB load balancer DNS name: example-1.us-east-2.elb.amazonaws.com
  - S3 website endpoint: s3-website.us-east-2.amazonaws.com
  - Resource record set in this hosted zone: www.example.com
  - VPC endpoint: example.us-east-2.vpc.amazonaws.com
  - API Gateway custom regional API: d-abcde12345.execute-api.us-west-2.amazonaws.com
  - Global Accelerator DNS name: a012345abc.awsglobalaccelerator.com
- [Learn More](#)

Routing Policy:

Route 53 responds to queries using primary record sets if any are healthy, or using secondary record sets otherwise. [Learn More](#)

Failover Record Type: ☐ Primary ☒ Secondary

Set ID:

Evaluate Target Health: ☐ Yes ☒ No


Associate with Health Check: ☐ Yes ☒ No

Route 53 Configuration is now completed.

## Task 4: IAM Policy and Roles for Lambda Function

### IAM Policy:

In this lab, we are going to use default IAM Policy which is **AutoScalingFullAccess**.

Filter policies <input type="text" value="AutoscalingFu"/>				
	Policy name	Type	Used as	Description
<input type="radio"/>	 AutoScalingFullAccess	AWS managed	Permissions policy (1)	Provides full access to Auto Scaling.

### 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 default IAM Policy **AutoScalingFullAccess**.

Filter policies ▾		Autoscalingfull	Showing 1 result
	Policy name ▾	Used as	
<input checked="" type="checkbox"/>	AutoScalingFullAccess	Permissions policy (1)	

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

Role has been created Successfully.

low		
Role name ▾	Trusted entities	Last activity ▾
<input type="checkbox"/> LowCapacityStandbyRole	AWS service: lambda	Today

## Task 5: Configure a Lambda Function

Make sure you're in US East (N. Virginia) (us-east-1) region.

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 and click on Create Function.

### Runtime [Info](#)

Choose the language to use to write your function.

Python 3.7 ▾

### Permissions [Info](#)

Lambda will create an execution role with permission to upload logs to Amazon CloudWatch Logs. You can configure and modify permissions further when you add triggers.

#### ▼ Choose or create an execution role

#### Execution role

Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).

- ☐ Create a new role with basic Lambda permissions
- ☒ Use an existing role
- ☐ Create a new role from AWS policy templates

#### Existing role

Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.


LowCapacityStandbyRole ▾



[View the LowCapacityStandbyRole role](#) on the IAM console.

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.

**Trigger configuration**

 SNS  
aws   messaging   notifications   pub-sub   push ▼

SNS topic  
Select the SNS topic to subscribe to.

Lambda will add the necessary permissions for Amazon SNS to invoke your Lambda function from this trigger. [Learn more](#) about the Lambda permissions model.

☒ Enable trigger  
Enable the trigger now, or create it in a disabled state for testing (recommended).

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.

**Basic settings**

Description	Memory (MB) <a href="#">Info</a>
-	128

Timeout [Info](#)  
0 min **10 sec**

Now write down the Lambda Function Code as follows:

```
lambda_function x   
1 import boto3  
2 def lambda_handler(event, context):  
3  
4     client = boto3.client('autoscaling')  
5     response = client.set_desired_capacity(  
6         AutoScalingGroupName='ASG',  
7         DesiredCapacity=3,  
8         HonorCooldown=True,  
9  
10    )
```

In line number 6, make sure to replace the **AutoScalingGroupName**.

This Lambda Function sets the Autoscaling Group's Desired capacity to 3.

Finally, click on SAVE.

Lambda function is now configured properly.

	Function name	Description	Runtime	Code size	Last modified
<input type="radio"/>	LowCapacityStandBy		Python 3.7	297 bytes	18 hours ago

## Task 6: Setup to test the Fully Working Low-Capacity Standby Scenario

Select US East (N. Virginia) (US-EAST-1) region.

Navigate to EC2 Service and click on Autoscaling Group.

Make a note of Desired Capacity which is currently 2.

Filter: <input type="text" value="Filter Auto Scaling groups..."/>		1 to 1 of 1 Auto Scaling Groups							
<input checked="" type="checkbox"/>	Name	Launch Configuration	Instances	Desired	Min	Max	Availability Zones	Default Cooldown	Health Check
<input checked="" type="checkbox"/>	ASG	LC	2	2	2	3	us-east-1a, us-east-1b	300	300

## Task 7: Test the Fully Working Low-Capacity Standby Architecture

Select the Asia Pacific (Mumbai) (ap-south-1) region.

Navigate to the EC2 Service, click on our Primary EC2 Instance.

Click on Actions > Instance State > STOP

Launch Instance  Actions

☐ test  
☐ Mumbai-Front End Server  
☒ Low Capacity Standby-Primary

Instance ID: i-066243cb3b8932c1f (Low Capacity Standby-Primary) Elastic IP: 3.6.223.194

Description Status Checks Monitoring Tags

Instance ID: i-066243cb3b8932c1f  
 Instance state: running  
 Public DNS (IPv4): ec2-3-6-223-194.ap-south-1.compute.amazonaws.com  
 IPv4 Public IP: 3.6.223.194

Actions menu:  
 Connect  
 Get Windows Password  
 Create Template From Instance  
 Launch More Like This  
 Instance State (selected)  
 Instance Settings  
 Image  
 Networking  
 CloudWatch Monitoring

Instance State sub-menu:  
 Start  
 Stop (selected)  
 Stop - Hibernate  
 Reboot  
 Terminate

Name	Type	Availability Zone	Instance State	Status Checks	Alarm Status
test			running	2/2 checks ...	None
Mumbai-Front End Server			running	2/2 checks ...	None
Low Capacity Standby-Primary			running	2/2 checks ...	None

Resource Groups							
<div> <div>Launch Instance</div> <div>Connect</div> <div>Actions</div> </div> <div> <input type="text"/> Filter by tags and attributes or search by keyword         </div>							
<input type="checkbox"/>	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Stat
<input type="checkbox"/>	test	i-0e11f191d80731b88	t2.micro	ap-south-1a	running	2/2 checks ...	None
<input type="checkbox"/>	Mumbai-Front End Server	i-0b09aec945f449db3	t2.micro	ap-south-1a	running	2/2 checks ...	None
<input checked="" type="checkbox"/>	Low Capacity Standby-Primary	i-066243cb3b8932c1f	t2.micro	ap-south-1a	stopped		None

Let's look at the Route53 Health Check now.

Health Check shows as Unhealthy, this would have triggered our SNS Topic.

Filter by keyword				
Name	Status	Description	Alarms	
<input checked="" type="checkbox"/> Low Capacity Check	18 hours ago <div><div></div></div> Unhealthy	http://3.6.223.194:80/	1 of 1 in ALARM	

Info

Monitoring

Alarms

Tags

Health checkers

Latency

ID

c6996245-38e7-47e1-9af0-90af033502fc

URL

http://3.6.223.194:80/

Specify endpoint by

IP address

Advanced configuration

Request interval

30 seconds

Failure threshold

1

SNS Topic then would have triggered the Lambda Function and Lambda Function would have changed the Desired Capacity of Autoscaling Group to 3 in N. Virginia region.

Resource Groups							
<div> <div>Try the new design for Amazon EC2 Auto Scaling</div> <div>This older console is being replaced with the new EC2 Auto Scaling console. No new features or improvements will be made in this older console. <a href="#">Go to the new console.</a></div> </div> <div> <div>Create Auto Scaling group</div> <div>Actions</div> </div>							
Filter: <input type="text"/> Filter Auto Scaling groups...							
<input checked="" type="checkbox"/>	Name	Launch Configuration /	Instances	Desired	Min	Max	Availability Zones
<input checked="" type="checkbox"/>	ASG	LC	3	3	2	3	us-east-1a, us-east-1b

You can also notice that a new Instance got launched in the N. Virginia region to main the desired capacity of 3.

Resource Groups

Prasad\_SMU

N. Virginia

Support

Launch Instance

Connect

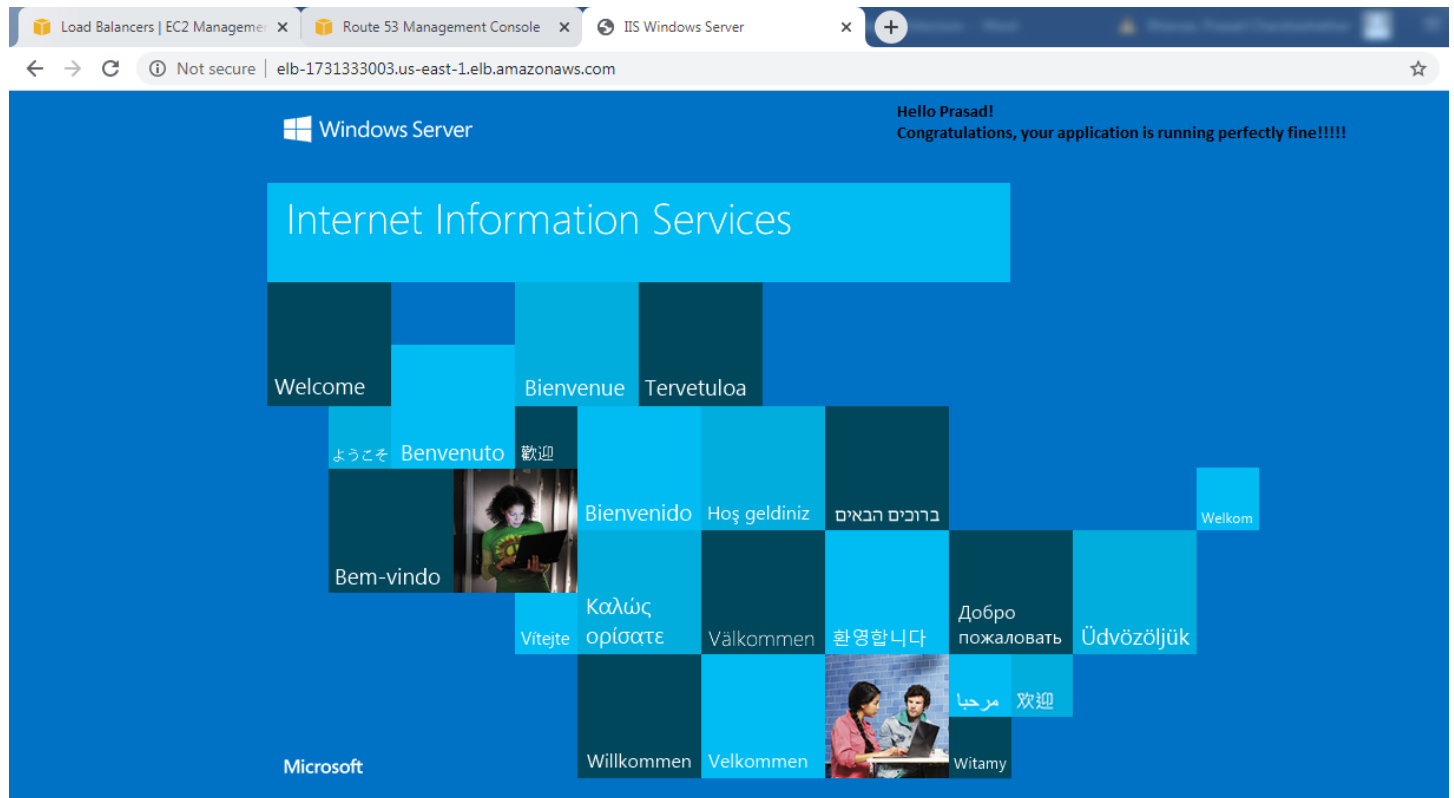
Actions

Filter by tags and attributes or search by keyword

1 to 5 of 5

	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm S
		i-0fd0a9b58b5604fcf	t2.micro	us-east-1b	running	Initializing	None

Simultaneously, the Route 53 Failover Policy has now detected the Failover and routed the traffic towards the Load Balancer in N. Virginia Region.



This proves the Fully Working Low-Capacity Standby Architecture.

For questions, contact me on [pbhavsar@smu.edu](mailto:pbhavsar@smu.edu) .