High Availability Across Regions

We can achieve High Availability across multiple Availability Zones using Elastic Load Balancer.

However, if you want to achieve High Availability across different Regions then you can achieve this by using Route 53 service in AWS.

In the last lab, we've configured Highly Availability across different Availability Zones. In this lab, we are going to take this architecture to the next level by implementing High Availability across different AWS Regions.

Now even if entire AWS region gets destroyed due to any catastrophic event, your application will be still up and running 😊

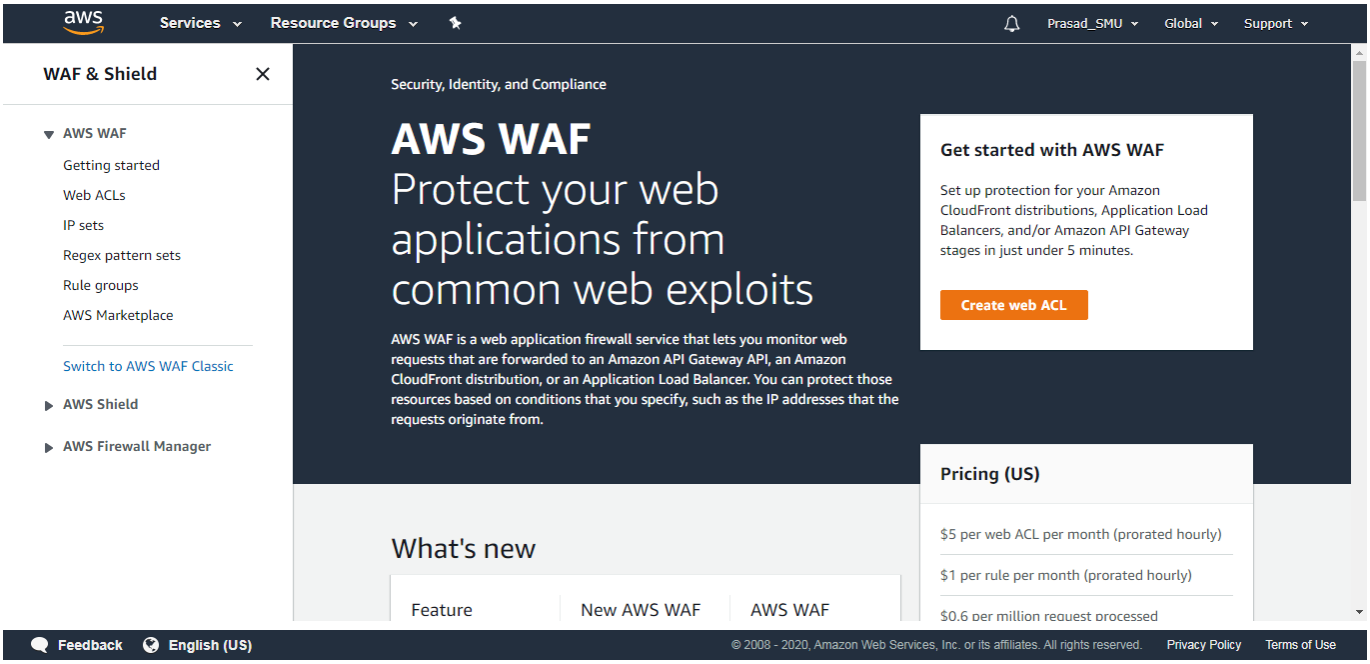
Below is the list of tasks that we are going to perform in this lab.

- Task 1: Create a Web Application Firewall (WAF).
- Task 2: Create CloudFront Distribution with the Source as Elastic Load Balancer (ELB).
- Task 3: Copy the Amazon Machine Image (AMI) to another Region (Mumbai).
- Task 4: Launch a new Windows Server 2016-IIS Web Server in Mumbai region using AMI.
- Task 5: Create a Hosted Zone on Route 53.
- Task 6: Create Health Checks for the Windows Server 2016-IIS Web Server.
- Task 7: Create A records and define Failover routing policies in Route 53.
- Task 8: Test the Failover.

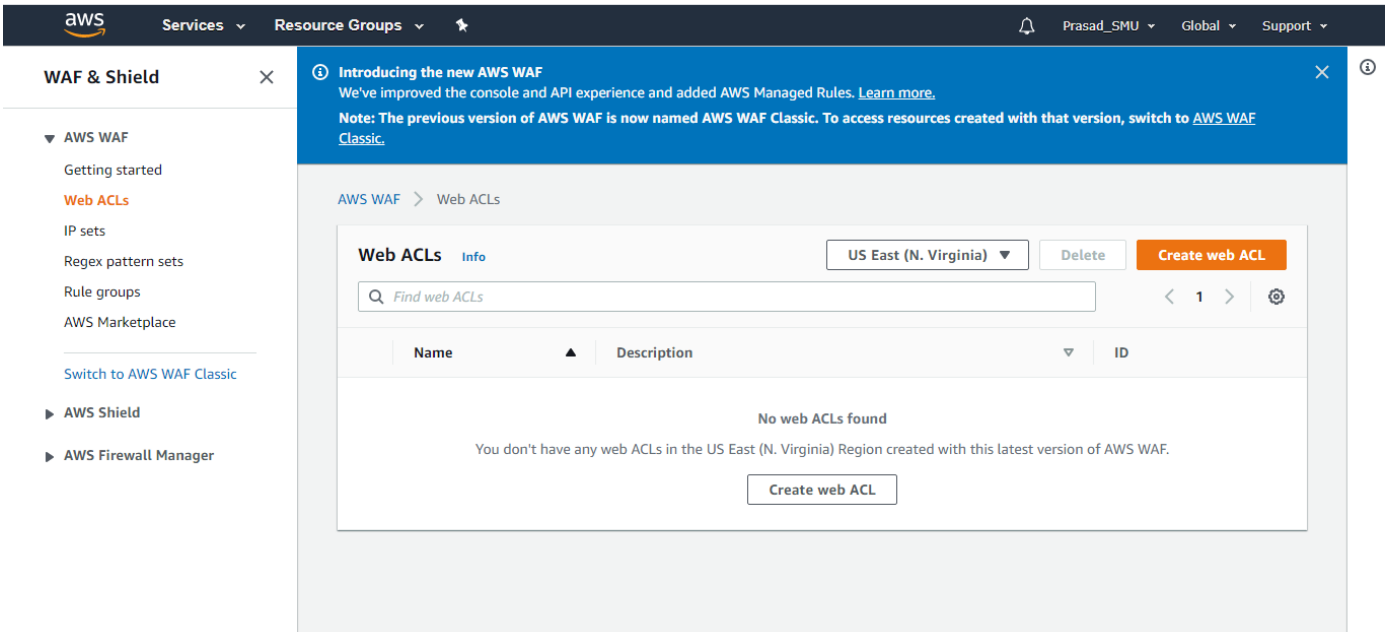
Task 1: Create a Web Application Firewall (WAF).

Web Application Firewall (WAF) is used to protect your application against Distributed Denial of Service (DDoS) attacks. We will make use of this Service to add additional layer of security to our IIS Web Server architecture.

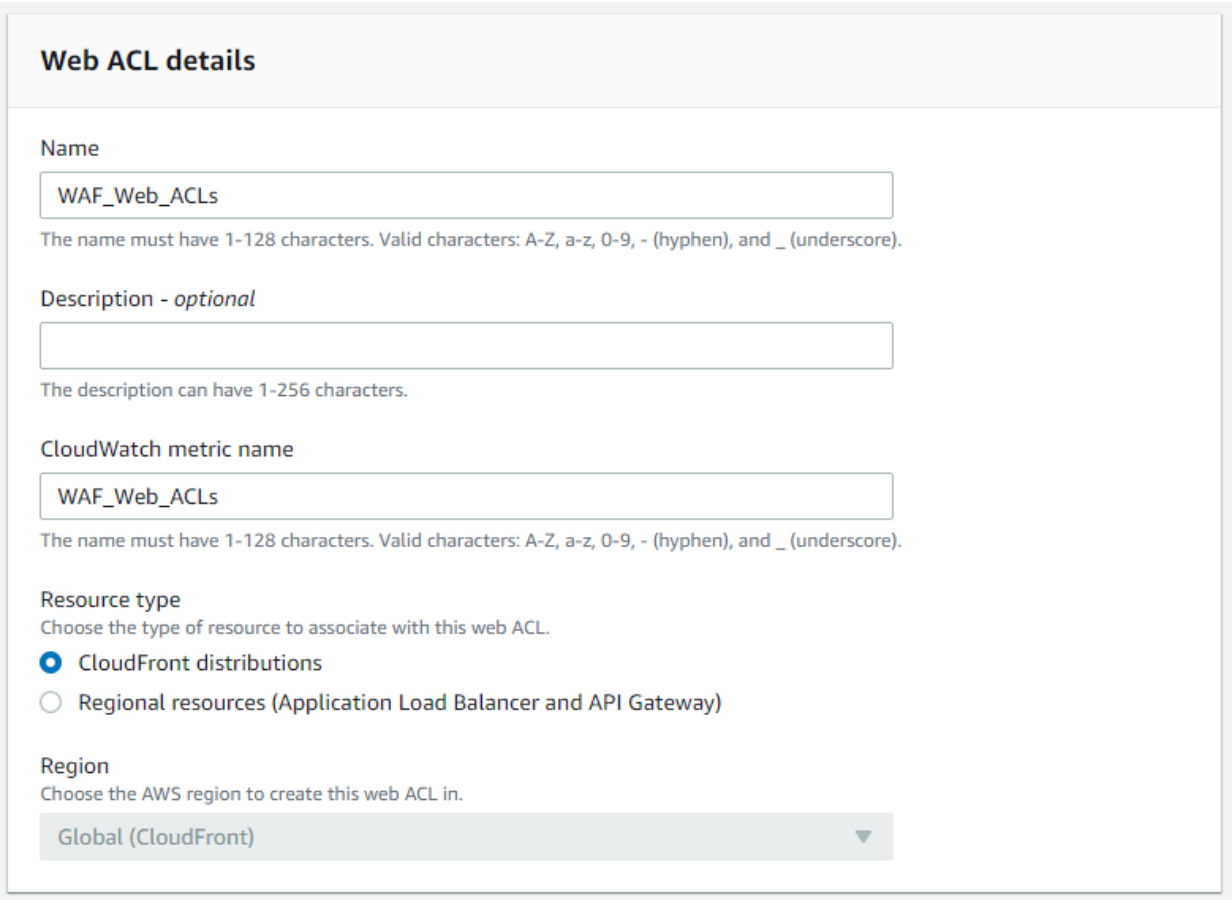
Open the AWS Management Console and navigate to WAF & Shield Service.



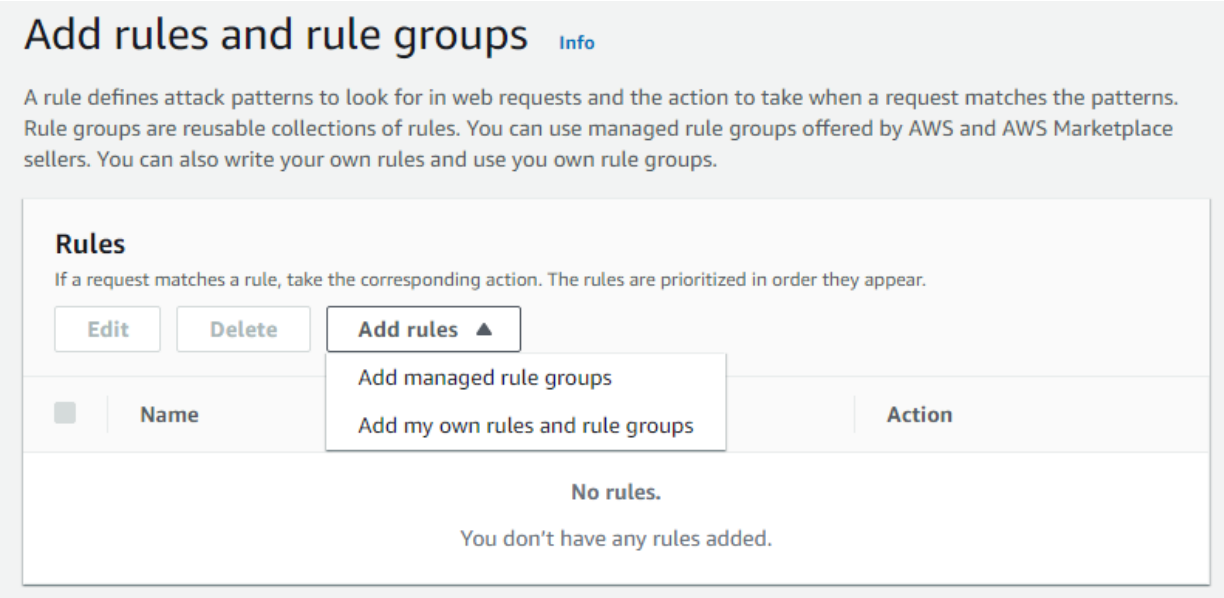
On right hand side, under Web Application Firewall, click on Web ACLs.



Now click on Create Web ACL. Give the Name as of your choice and select Resource Type as CloudFront Distribution and click Next.



Click on Add Rules and select Add Managed Rule Groups.



Expand AWS managed rule groups, and select Windows Operating System rule at the bottom.

Windows operating system

Contains rules that block request patterns associated with exploiting vulnerabilities specific to Windows, (e.g., PowerShell commands). This can help prevent exploits that allow attacker to run unauthorized commands or execute malicious code.

200

Add to web ACL

Set rules action to count

This rule helps to protect our Windows based EC2 Instances against external Vulnerabilities.

Add rules and rule groups Info

A rule defines attack patterns to look for in web requests and the action to take when a request matches the patterns. Rule groups are reusable collections of rules. You can use managed rule groups offered by AWS and AWS Marketplace sellers. You can also write your own rules and use you own rule groups.

Rules

If a request matches a rule, take the corresponding action. The rules are prioritized in order they appear.

Edit | Delete | Add rules

Name	Capacity	Action
AWS-AWSManagedRulesWindowsRuleSet	200	Count

Click on Add Rules. Keep default settings for next steps. Finally, review and Click on Create Web ACLs.

Success
You successfully created the web ACL: WAF_Web_ACLs

AWS WAF > Web ACLs

Web ACLs Info

Global (CloudFront) | Delete | Create web ACL

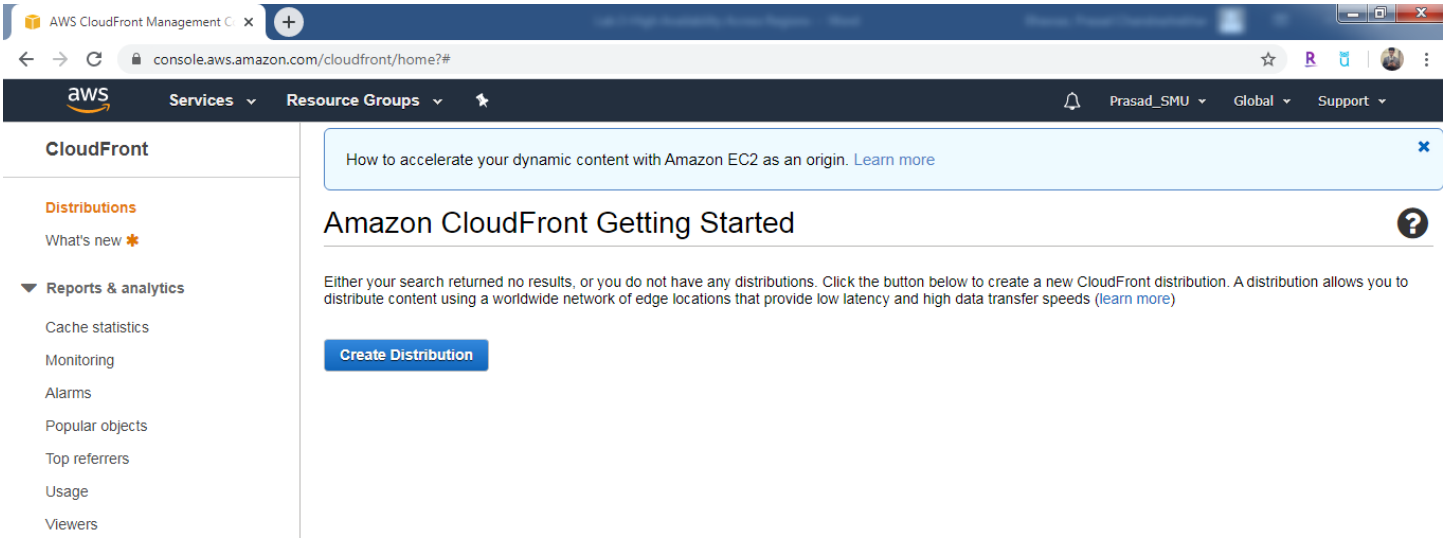
Find web ACLs

1

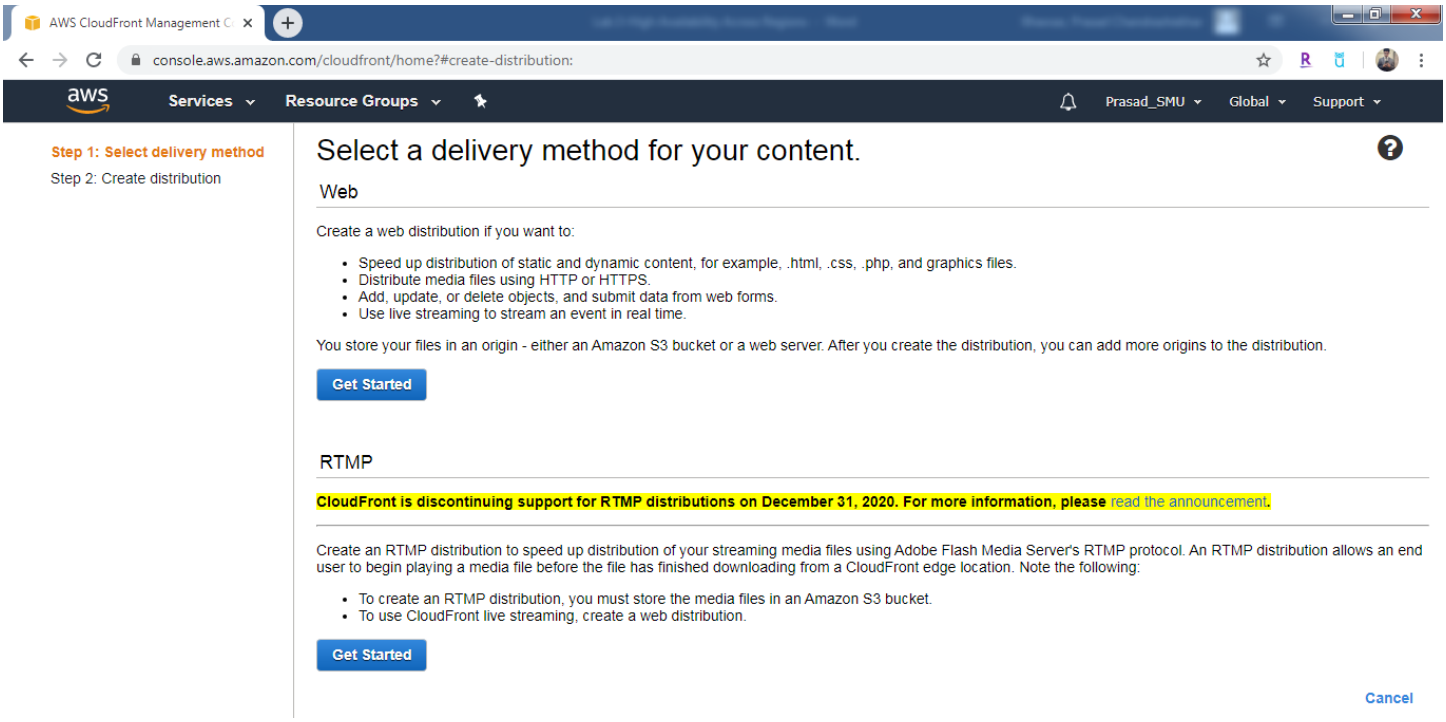
Name	Description	ID
WAF_Web_ACLs	-	3c770c78-ab70-4da2-b1cb-e5c4c5b88edd

Task 2: Create CloudFront Distribution.

Navigate to CloudFront Service.



Click on Create Distribution. Since we are going to distribute Web Application over HTTP and HTTPS, under Web select Get Started.



In the Origin Domain Name, select the Elastic Load Balancer that we've configured in the previous lab.

Origin Settings

Origin Domain Name	ELB-1731333003.us-east-1.elb.amazona
Origin Path	
Origin ID	ELB-ELB-1731333003
Minimum Origin SSL Protocol	○ TLSv1.2 ○ TLSv1.1 ◉ TLSv1 ○ SSLv3
Origin Protocol Policy	◉ HTTP Only ○ HTTPS Only ○ Match Viewer

Since the content which is going get served by CloudFront Edge Location is Static, we are keeping the Default TTL value as 86400 Seconds. It means for the specified TTL value, the content will be cached on the CloudFront Edge Locations.

Minimum TTL	0
Maximum TTL	31536000
Default TTL	86400

In-case of Dynamic Content, we can specify the TTL value as 0 Seconds. It means any request to the CloudFront Distribution will be directly forwarded to specified Source location.

Under Distribution Settings, in AWS WAF Web ACL, specify the Web ACLs which we've configured in the previous task.

Distribution Settings

Price Class Use All Edge Locations (Best Performance)

AWS WAF Web ACL WAF_Web_ACLs (wafv2)

Once all the settings are reviewed, click on Create Distribution.

Creation of your distribution on all the CloudFront edge locations takes time. Have some patience and keep observing the status of CloudFront Distribution.

Viewing 1 to 1 of 1 Items

Comment	Origin	CNAMEs	Status	State	Last Modified
-	ELB-1731333003.us-east-1.elb.amazonaws.com	-	Deployed	Enabled	2020-04-20 02:2

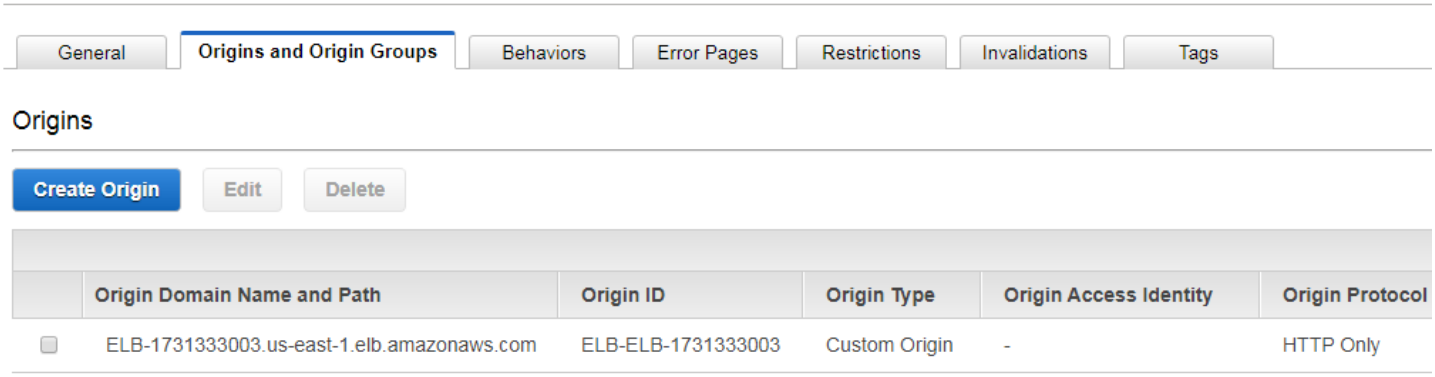
Once the CloudFront Distribution is created, click on Distribution ID. Under General Tab, verify the CloudFront Distribution Domain Name, WAF Web ACL etc.

General | Origins and Origin Groups | Behaviors | Error Pages | Restrictions | Invalidations | Tags

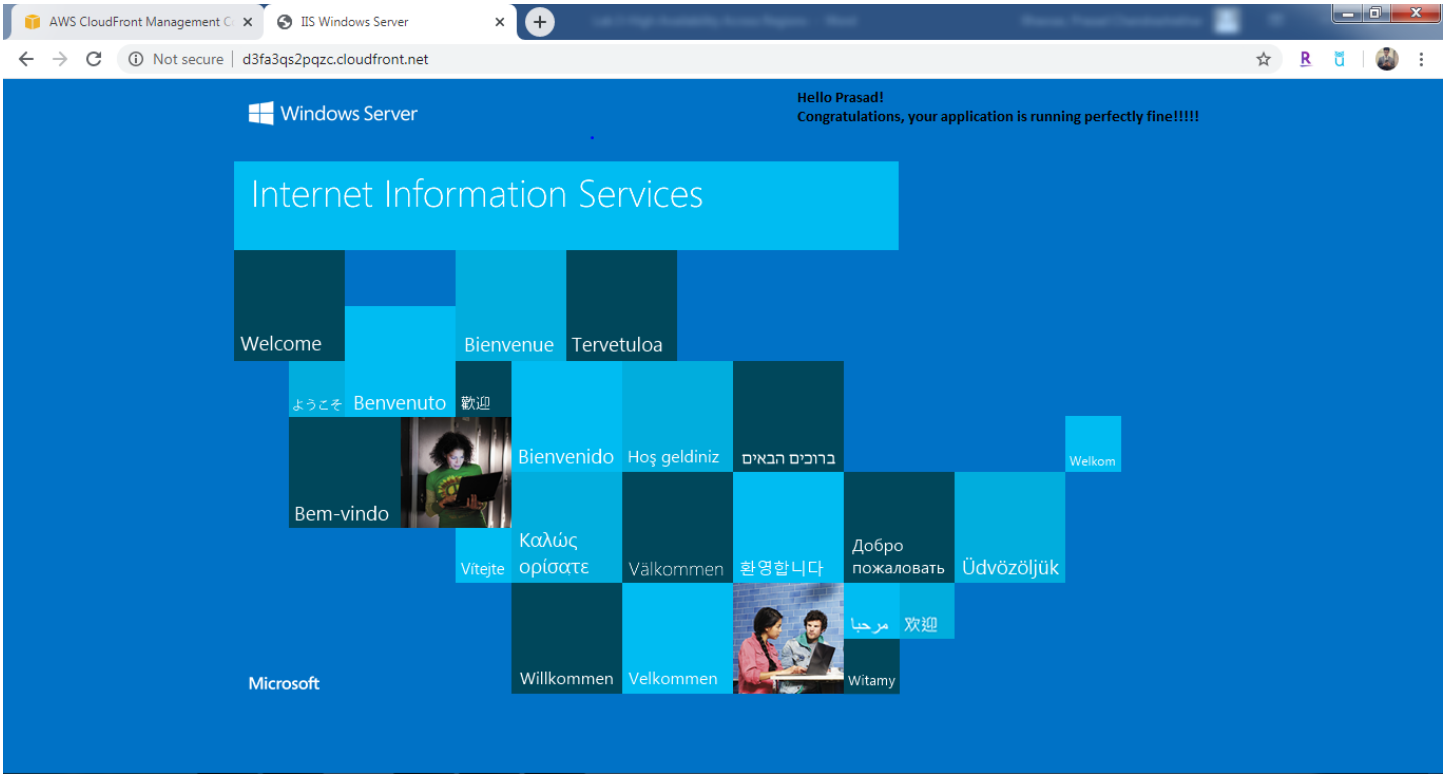
Edit

Distribution ID	ETS6R1DIEBILV
ARN	arn:aws:cloudfront::616399057974:distribution/ETS6R1DIEBILV
Log Prefix	-
Delivery Method	Web
Cookie Logging	Off
Distribution Status	Deployed
Comment	-
Price Class	Use All Edge Locations (Best Performance)
AWS WAF Web ACL	WAF_Web_ACLs (wafv2)
State	Enabled
Alternate Domain Names (CNAMEs)	-
SSL Certificate	Default CloudFront Certificate (*.cloudfront.net)
Domain Name	d3fa3qs2pqzc.cloudfront.net
Custom SSL Client Support	-

Now click on Origins and Origin Groups and verify the Source Location which is the Elastic Load Balancer (ELB).



Now Copy the CloudFront Distribution Name and put it in your Browser.

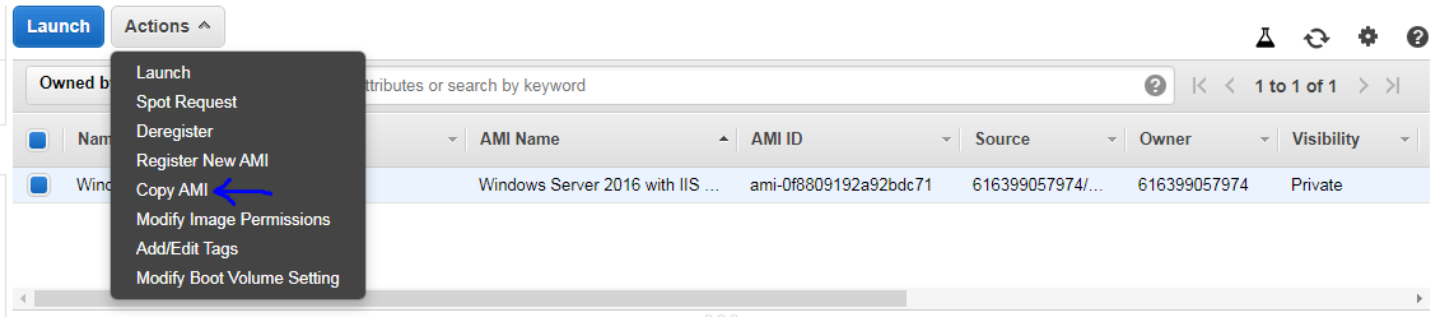


Website Opens, it means your application is now served by the CloudFront Distribution through Cache on Edge Locations. The resources utilization of base Infrastructure is now very minimal due to caching; hence we've now achieved Performance Efficiency successfully!!!!!!

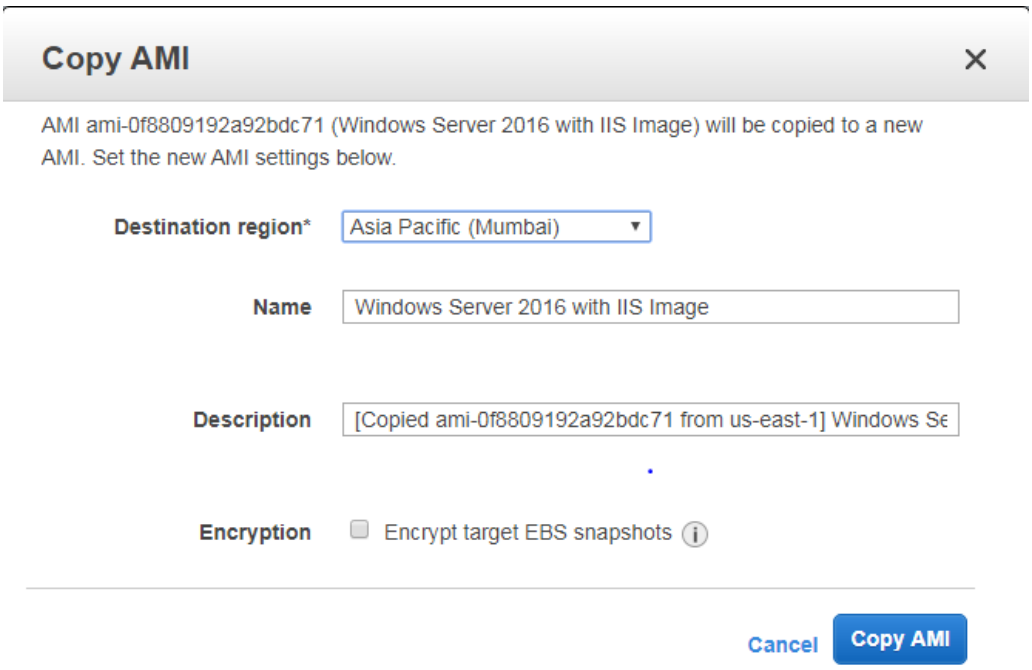
Task 3: Copy the Amazon Machine Image (AMI) to another Region (Mumbai).

We have one Amazon Machine Image (AMI) ready for the deployment of Windows Server 2016 IIS Web Server in US East (N. Virginial) region. We are now going to copy the same AMI Image to a new region Asia Pacific (Mumbai) and then we will deploy identical Web Server in Asia Pacific (Mumbai) region.

Navigate to EC2 Service and click on AMIs. Select the existing AMIs, click on Actions and click on Copy.



Select the Destination Region as Asia Pacific (Mumbai).



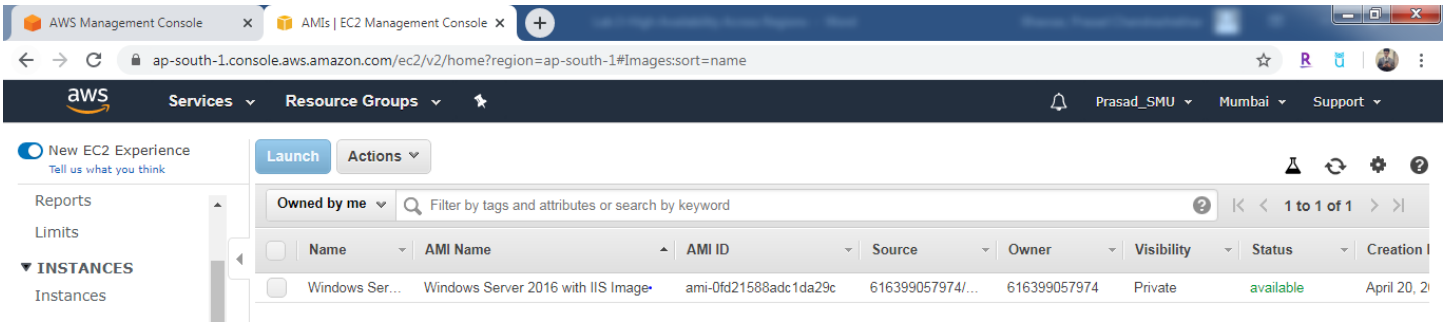
Click on Copy AMI. This will now deploy a new Amazon Machine Image (AMIs) in Asia Pacific (Mumbai) region.

Task 4: Launch a new Windows Server 2016-IIS Web Server in Mumbai region using AMI.

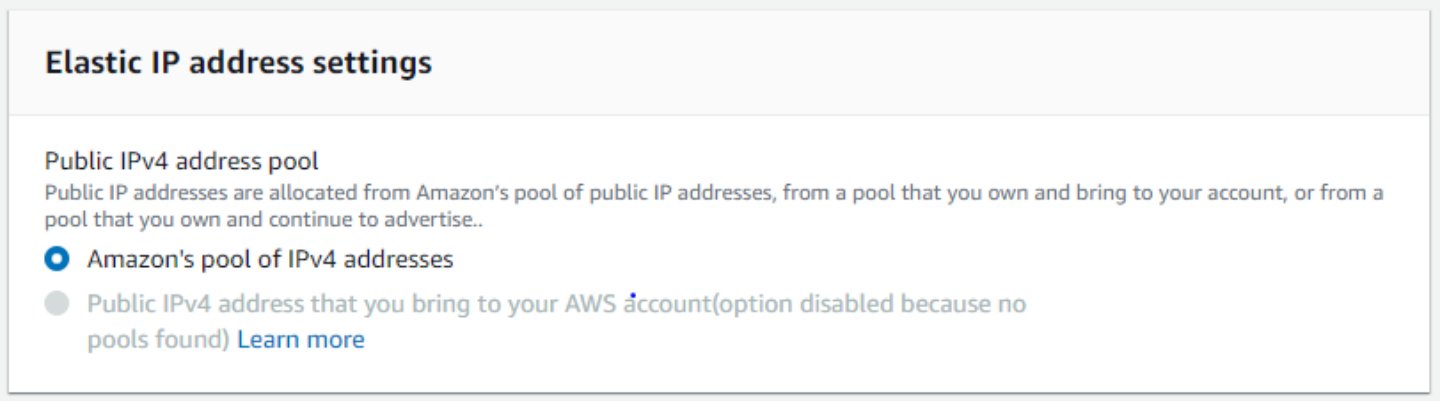
Now change the AWS region to Asia Pacific (Mumbai).

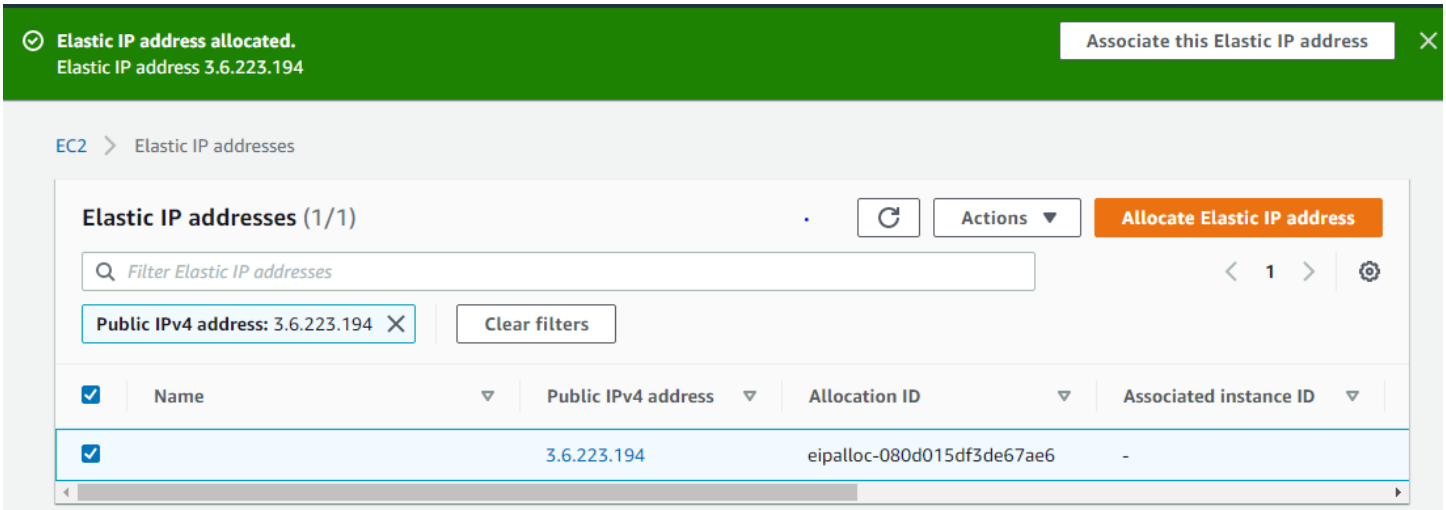
Navigate to EC2 Service and click on AMIs.

You'll notice that the AMI is now successfully copied to Asia Pacific (Mumbai) region from US East (N. Virginia) region.

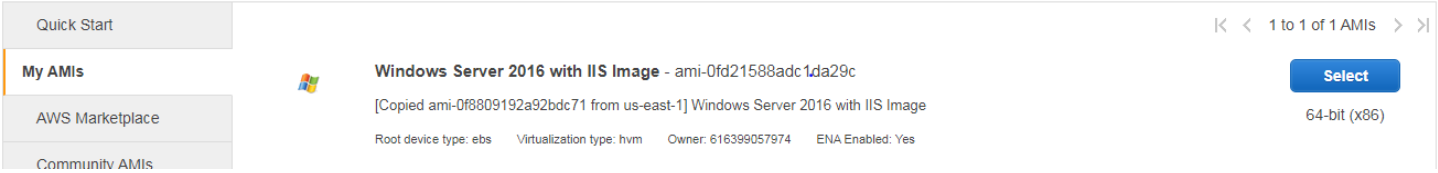


Navigate to Elastic IPs and Allocate an Elastic IP for your EC2 Instance from Amazon's pool of IPv4 addresses.



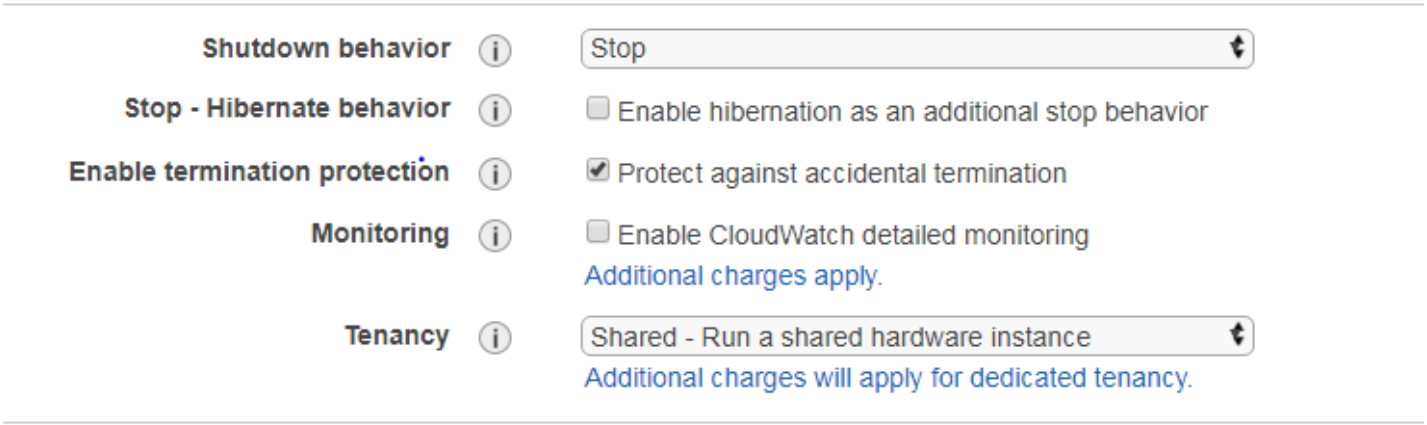


Navigate to Instances and click on Launch Instance. Click on My AMIs and select the existing AMI.



In Step 2: Choose Instance Type, keep the default Instance type as t2.micro and click Next.

In Step 3: Configure Instance Details, keep the Network as Default VPC and Subnet as of no preference. Make sure to select **enable protect against accidental termination** and **Shutdown behavior** as Stop and click Next.



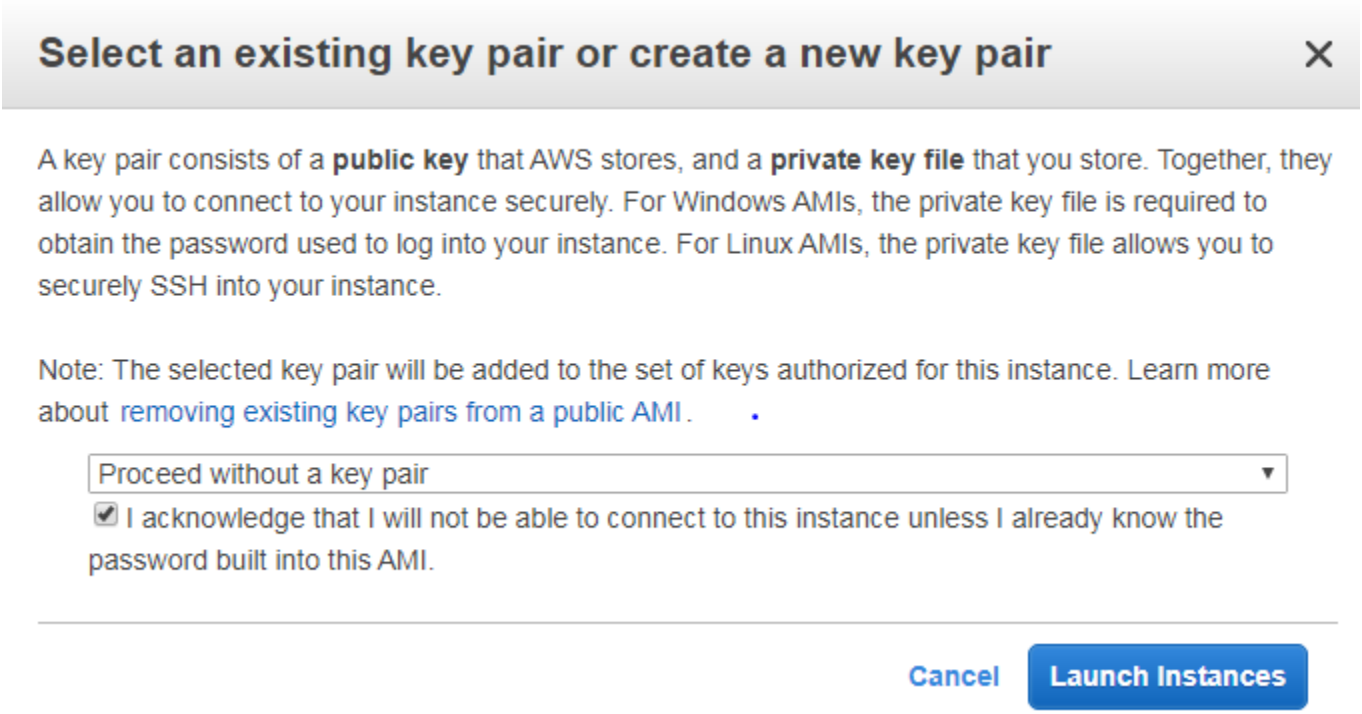
In Step 4: Add Storage, keep the default EBS storage settings.

In Step 5: Add Tags, you can add Tags If you wish else click Next.

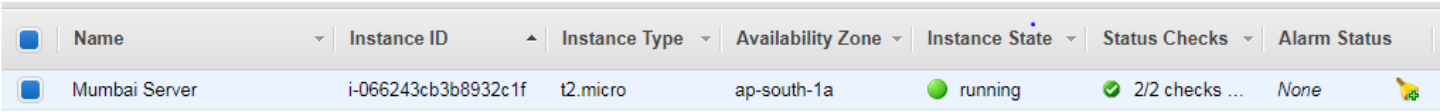
In Step 6: Configure Security Group, since we haven't configured any Security Group, you can either click a new Security Group for the Wen Server or click Next.

In Step 7: Review Instance Launch, review the configurations and click on Launch.

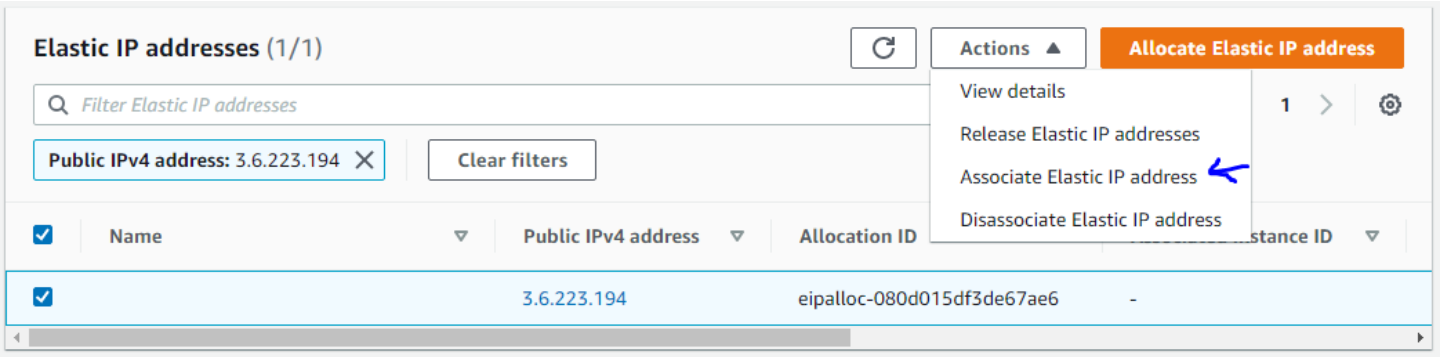
Here you can select proceed without a Key Pair since we are not going to take RDP of this Instance and click on Launch Instance.



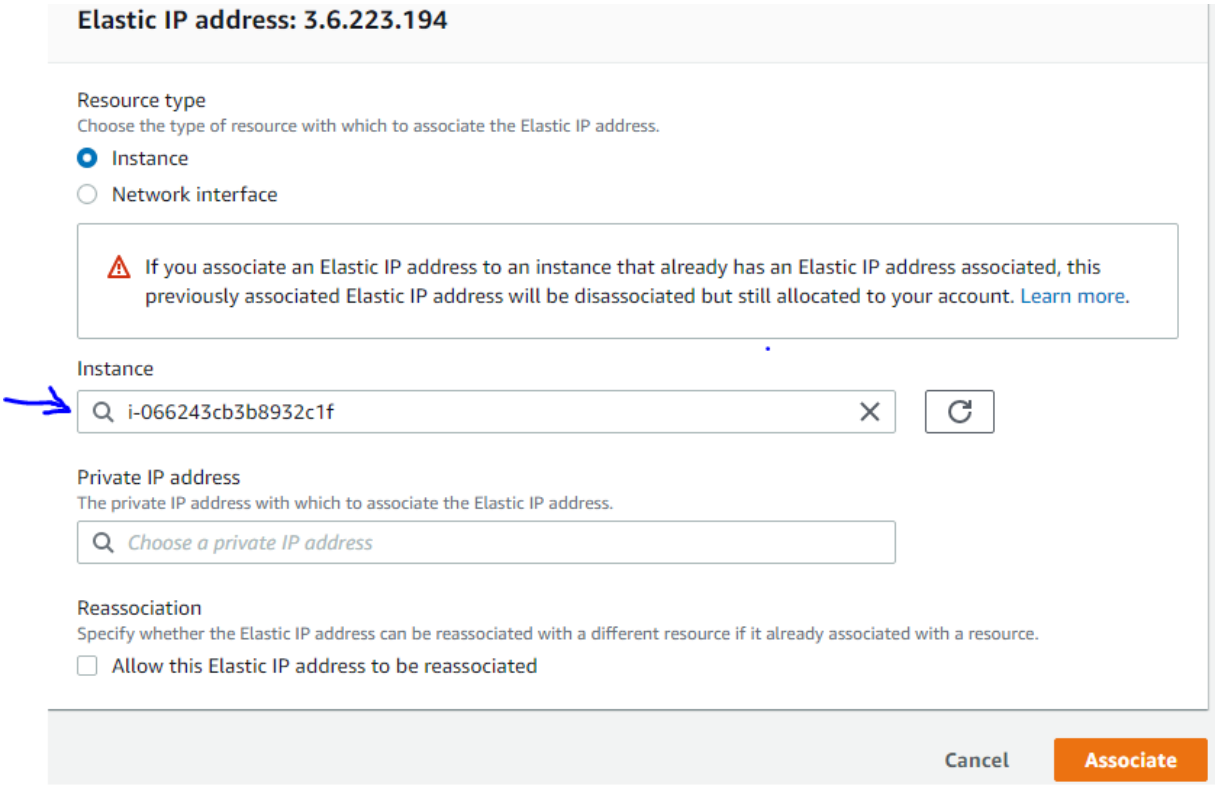
Wait till the EC2 Instance passes the 2/2 Status Check and Instance State as Running.



Once the Application Server is deployed, go back to Elastic IPs and click on the Allocated IP Address and click on Actions. Click on Associate Elastic IP Address.

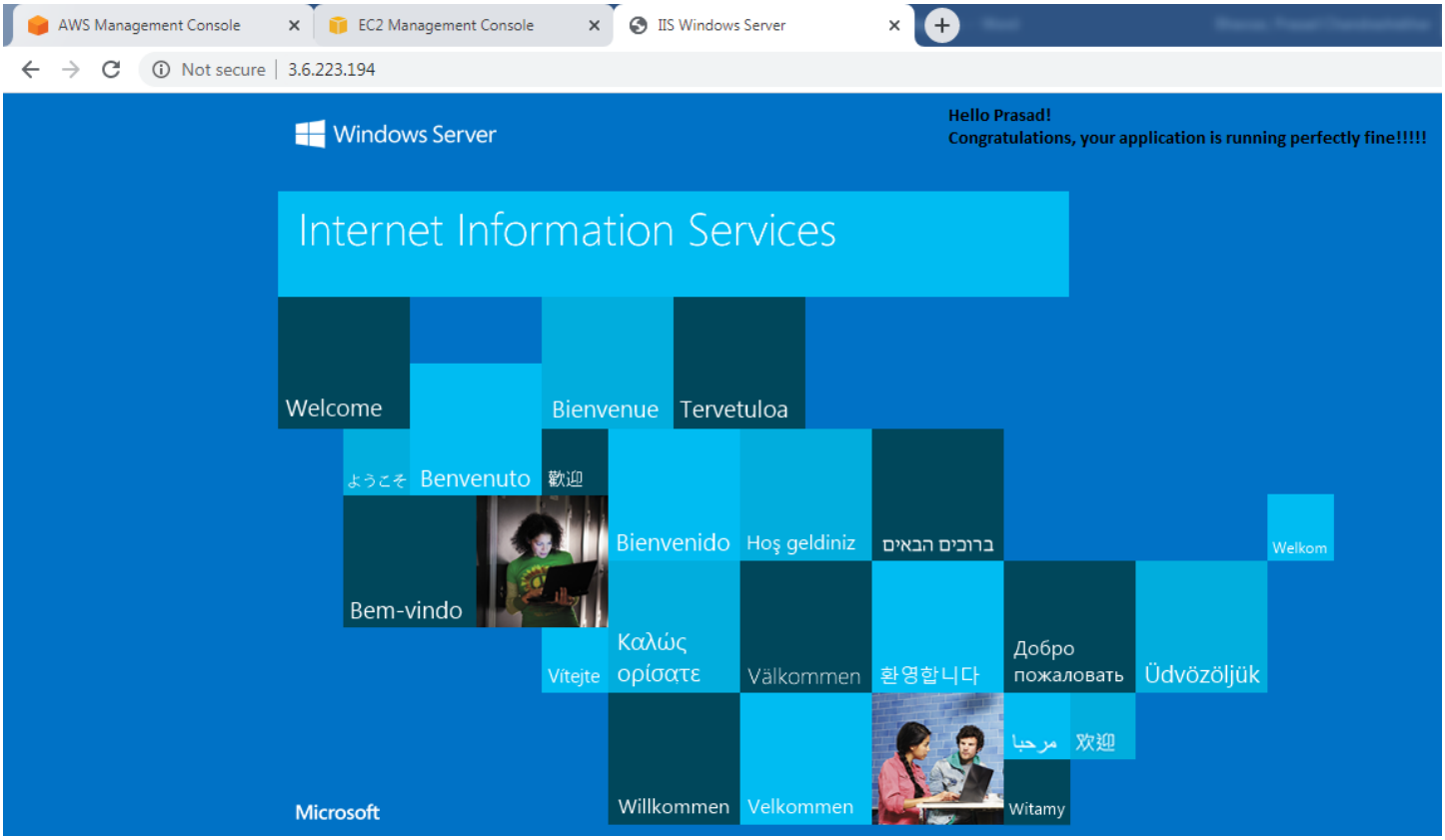


Select the Instance that you've recently launched and click on Associate.



You can now verify that the Elastic IP Address has been assigned to the Web Server running in the Asia Pacific (Mumbai) region.

Copy the EC2 Instance IP Address and put it in the browser, your application opens.



<u>NOTE:</u> In-case if the Web Page doesn't open then verify the Security Group associated with it.

It means your application is now also running on different Region which is Asia Pacific (Mumbai).

In the Summary, your application is running in two different regions as follows.

<u>Asia Pacific (Mumbai):</u> Application is hosted on a standalone EC2 Instance.

<u>US East (N. Virginia):</u> Application is running on Infrastructure wherein you've configured Elastic Load Balancer, Autoscaling etc.

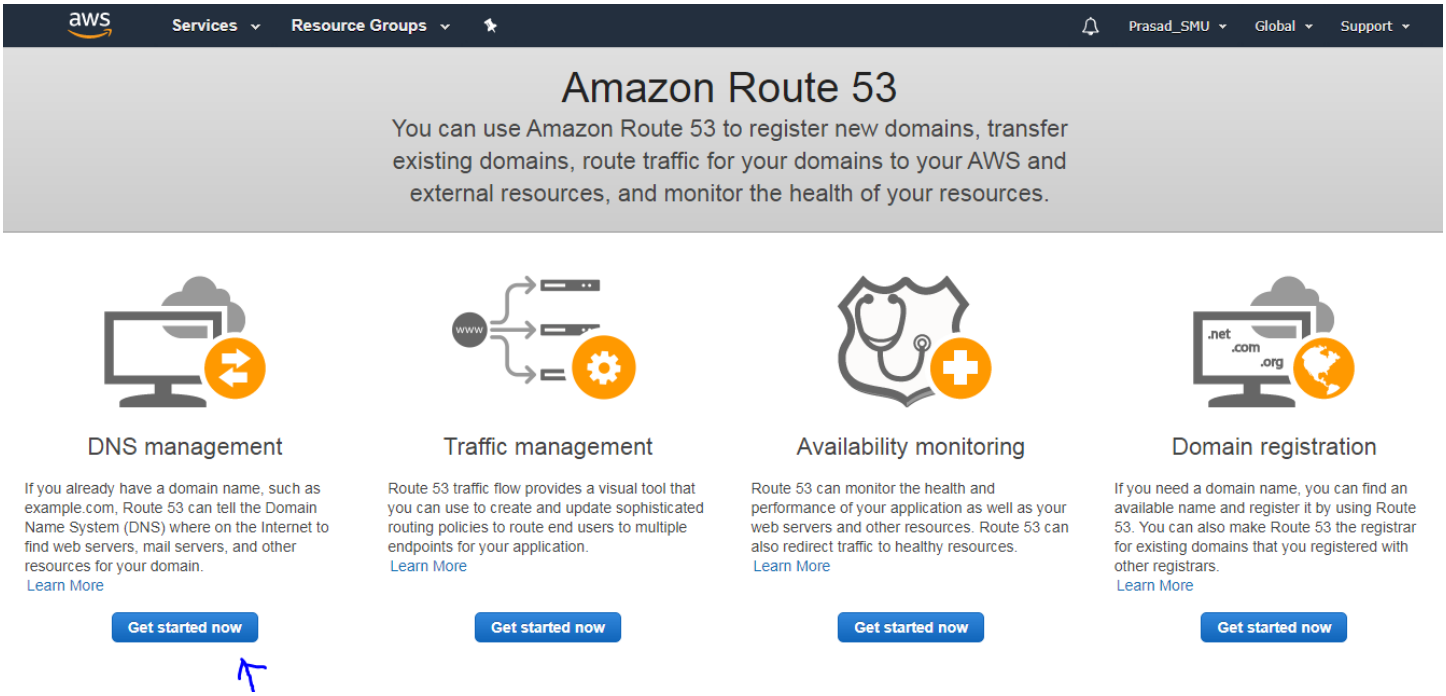
Now in the next tasks, you will configure Asia Pacific (Mumbai) as the Primary and US East (N. Virginia) as the Secondary region to run the Application.

In-case if the entire Asia Pacific (Mumbai) region goes down, your application will be still up through the US East (N. Virginia). This is what we called as High Availability Across Multiple Regions.

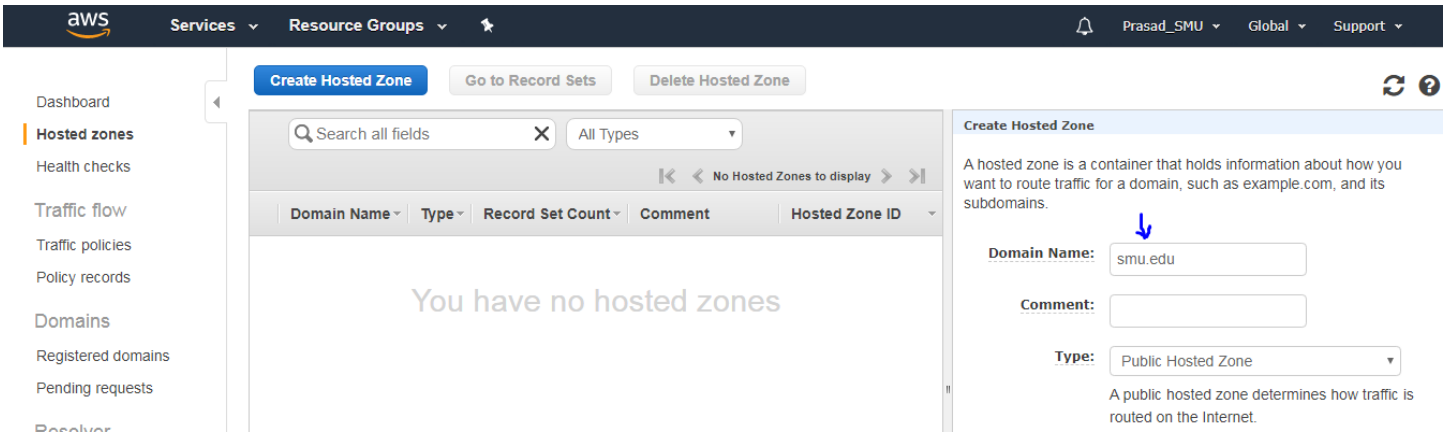
Task 5: Create a Hosted Zone on Route 53.

Come back to the Original Region (N. Virginia).

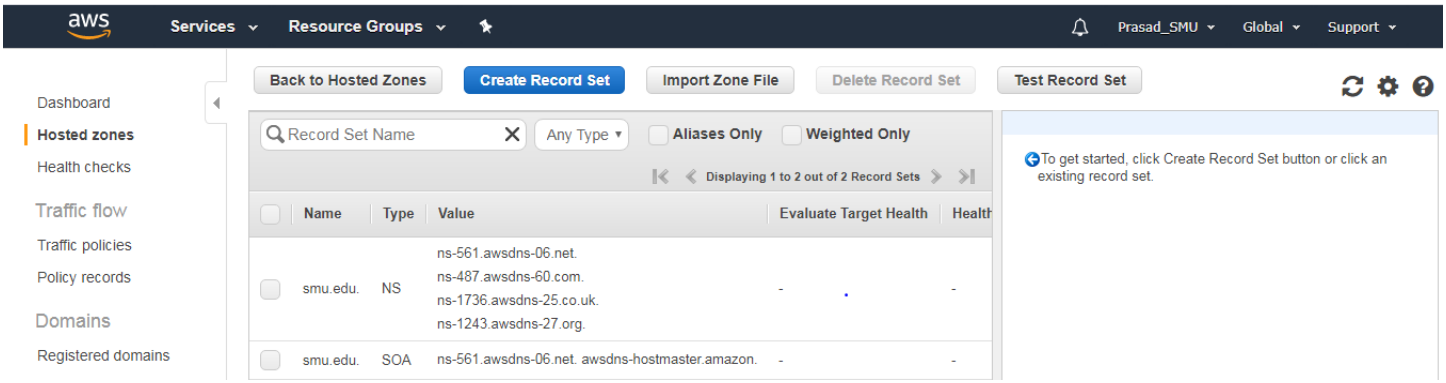
Navigate to Route 53 Service and under DNS Management click on Get Started.



Click on Create Hosted Zone. Give Domain Name and keep type as Public Hosted Zone.



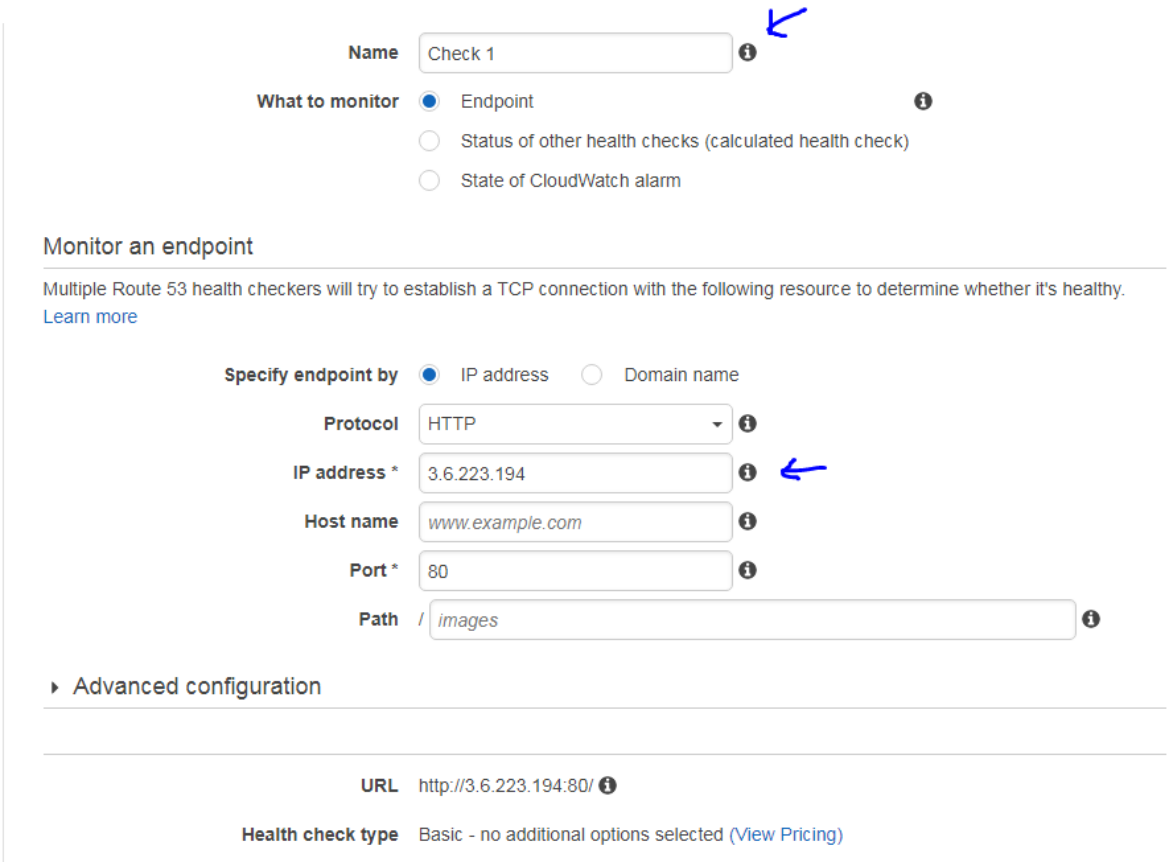
Click on Create. Your Domain is now configured.



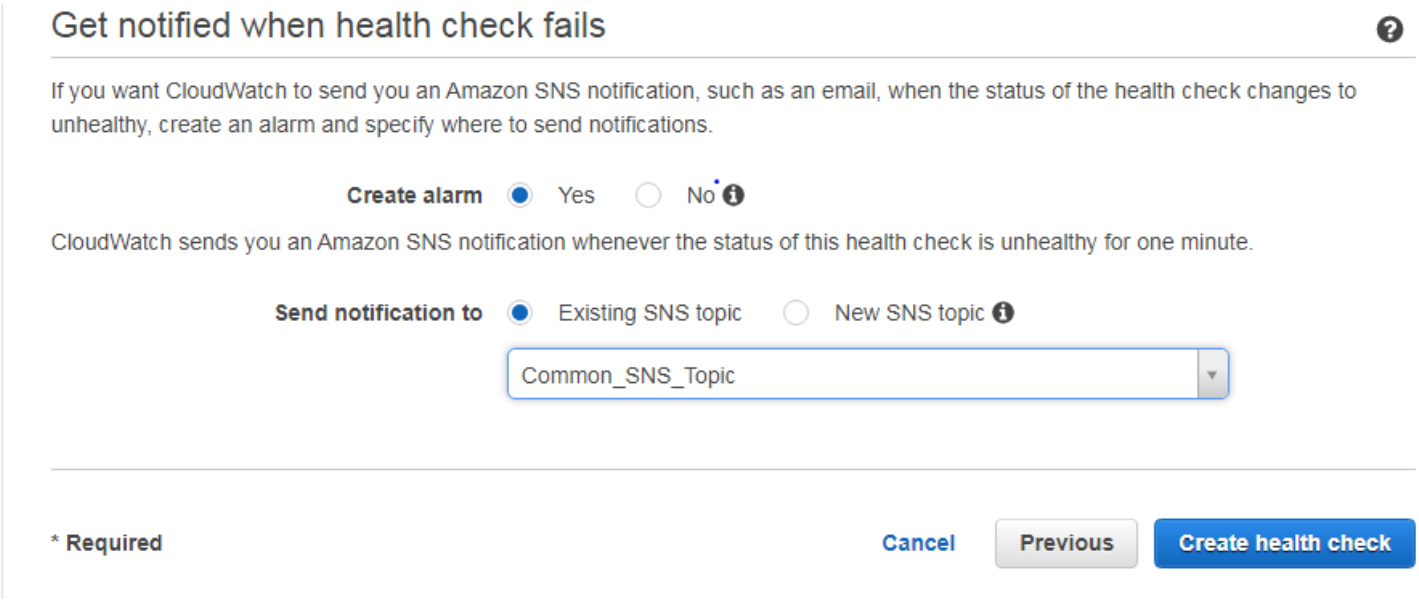
Task 6: Create Health Checks for the Windows Server 2016 IIS Web Server.

Now on left side, click on Health Checks. You'll now configure Health Check for the EC2 Instance in Mumbai region.

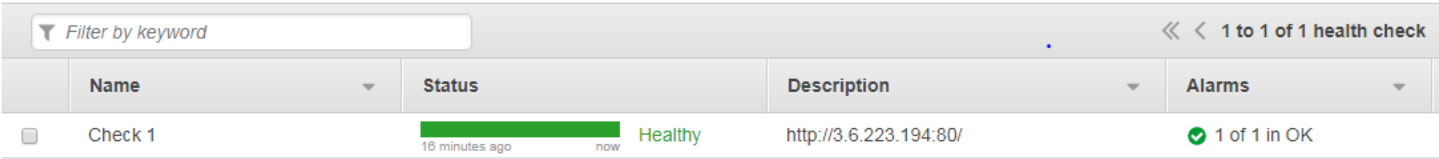
Click on Health Check. Give the Health Check Name, IP Address of your EC2 Instance running in Asia Pacific (Mumbai) region and click Next.



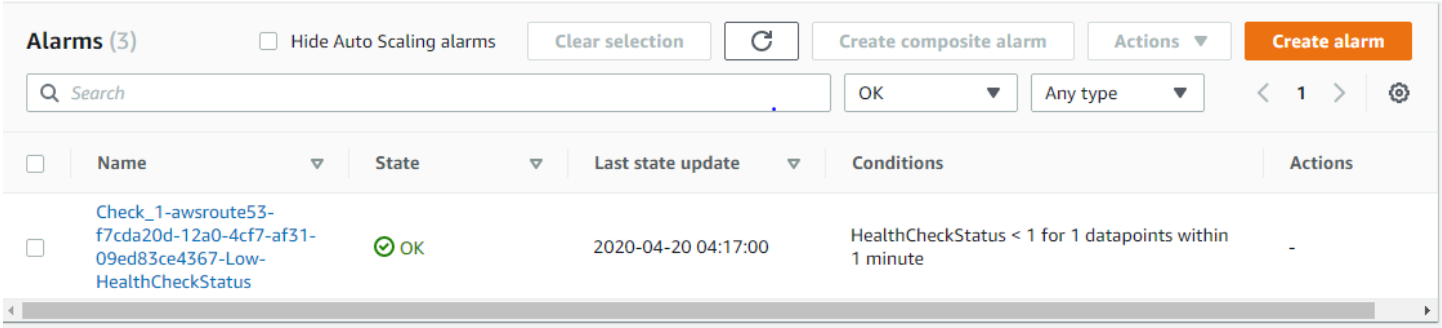
Create a CloudWatch Alarm which sends the notification to SNS Topic Subscribers in-case of failure and click on Create Health Check.



You'll observe that Health Check has been successfully configured which keeps on checking the Health of EC2 Instance launched in Asia Pacific (Mumbai) region.



If you navigate to CloudWatch service, you'll observe that the CloudWatch alarm has been successfully created for above Health Check.



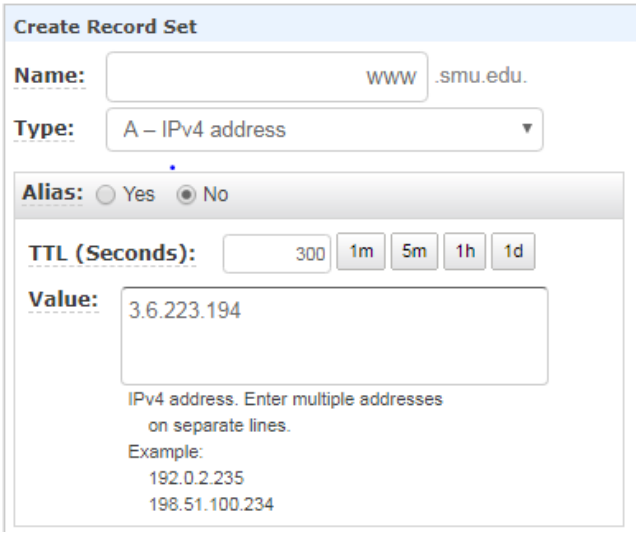
Task 7: Create A records and define Failover routing policies in Route 53.

Navigate to Route 53 Service, click on Hosted Zone that you've created and click on Create Record Set.

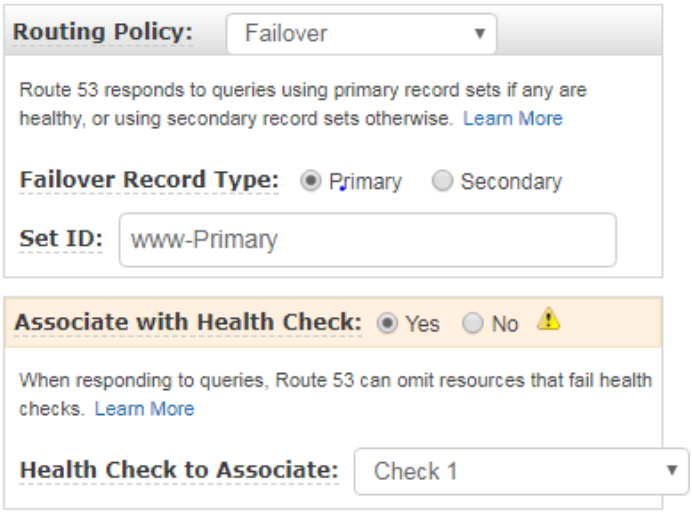
We'll now specify Asia Pacific (Mumbai) as the Primary and US East (N. Virginia) as the Secondary region for our IIS Web Application.

Create primary record set as follows.

Name will be "www"; hence the website becomes www.smu.edu. Value field contains the IP Address of the EC2 Instance which is running in the Asia Pacific (Mumbai) region.

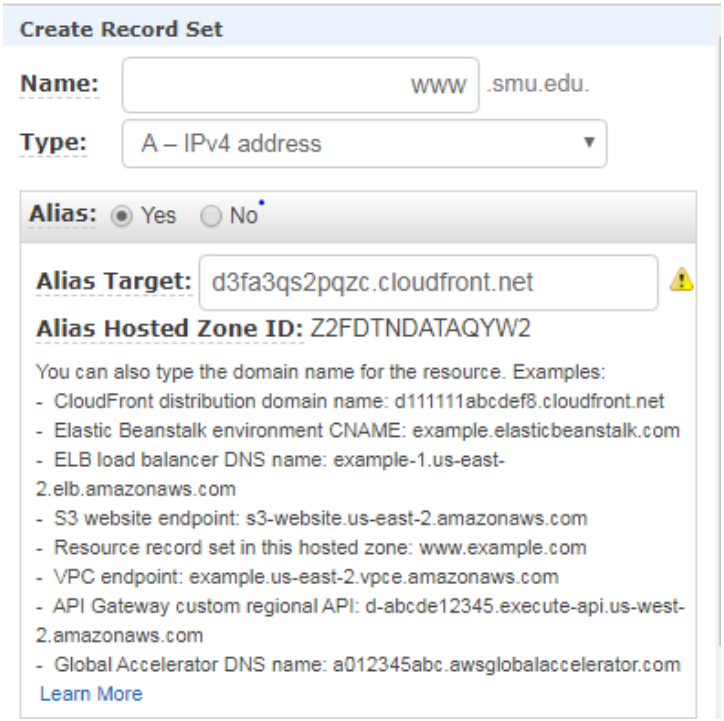


Select the Routing Policy as Failover. This Failover Record Type will be PRIMARY. Also associate Health Check that we've configured to this A Record and Click on Create Record.

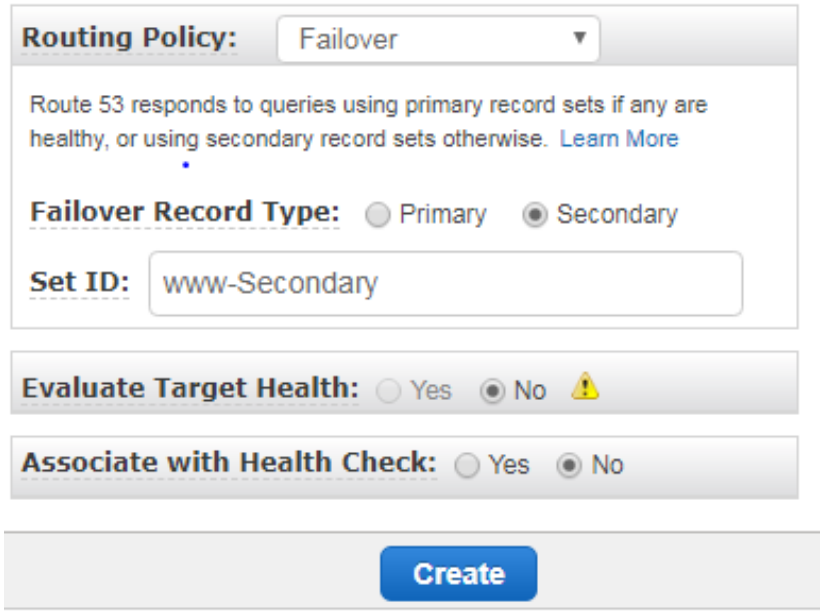


Similarly, again click on Create Record Set and create A Record as follows.

Name will be "www"; hence the website becomes www.smu.edu. Select Alias as YES and give Alias Target as CloudFront Domain which is caching the Website from US East (N. Virginia) region.



Select the Failover Routing Policy and select Failover Record Type as Secondary. No need to specify Health Check here and click on Create.

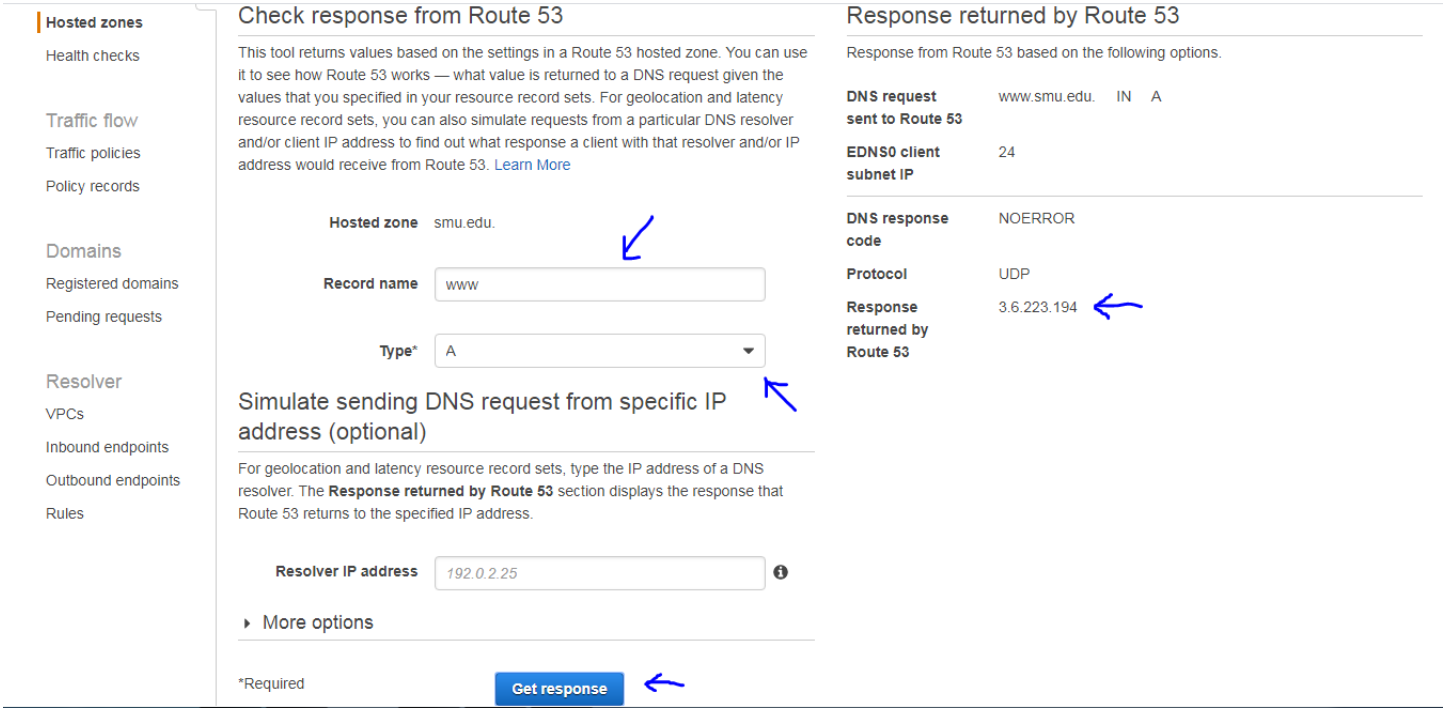


Your Records Sets should look like the below Screenshot.

Name	Type	Value	Evaluate Target Health
smu.edu.	NS	ns-561.awsdns-06.net. ns-487.awsdns-60.com. ns-1736.awsdns-25.co.uk. ns-1243.awsdns-27.org.	-
smu.edu.	SOA	ns-561.awsdns-06.net. awsdns-hostmaster.amazon.	-
www.smu.edu.	A	3.6.223.194	-
www.smu.edu.	A	ALIAS d3fa3qs2pqzc.cloudfront.net. (z2fdtndataqyw	No

Task 8: Test the Failover.

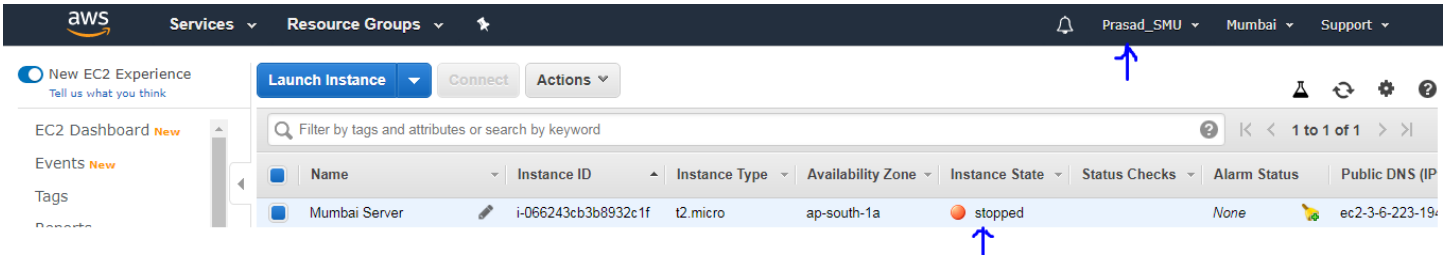
On the Route 53 Service, click on Test Record Set. Select Record Name as “www” and Record Type as “A” and click on Get Response.



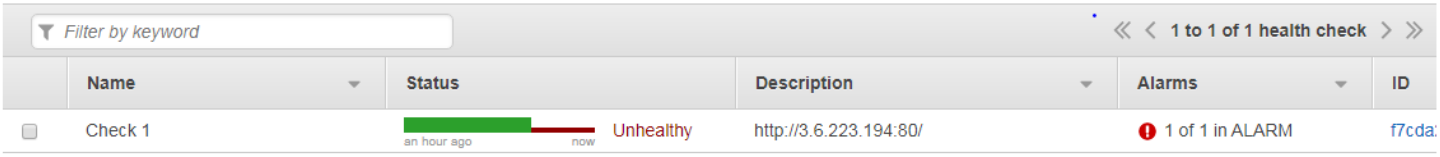
You'll notice that Response returned by Route 53 displays the IP Address of the EC2 Instance which is running in Asia Pacific (Mumbai) region. Which is absolutely Right 😉

Now let's STOP this server and test the Failover.

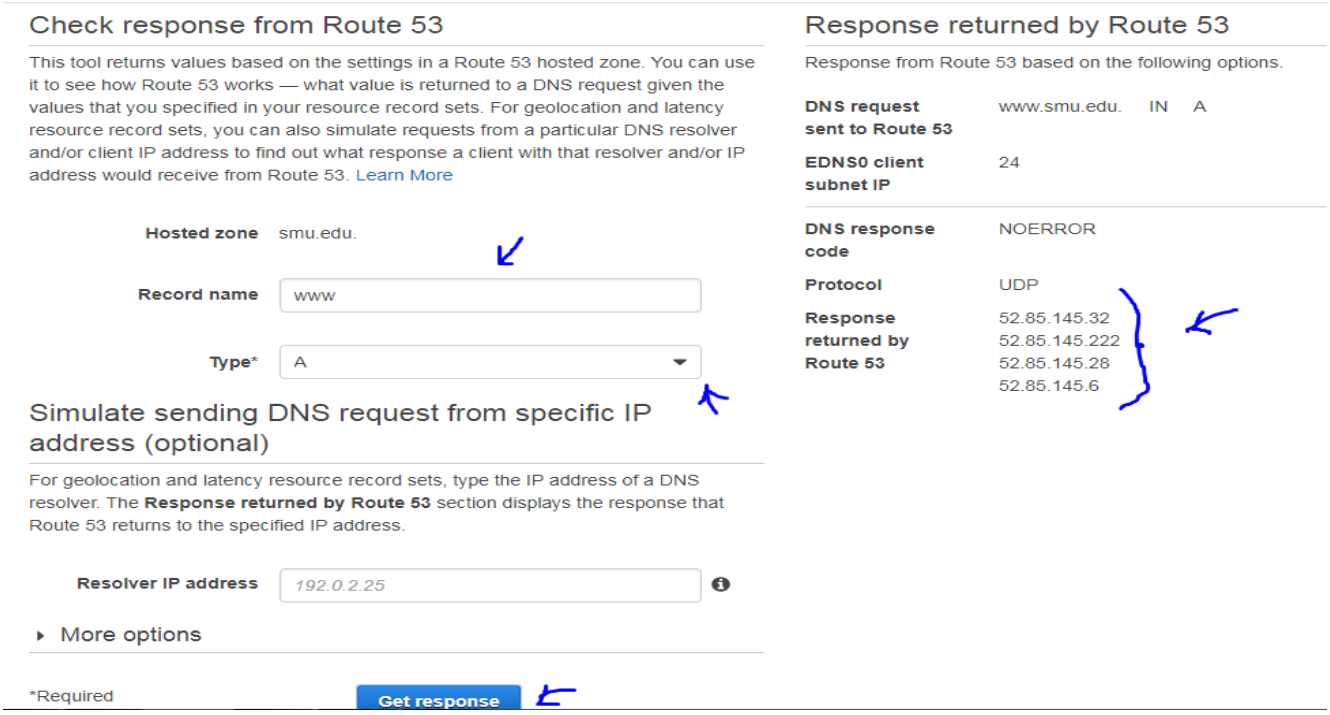
Go to Asia Pacific (Mumbai) region, navigate to EC2 Service and Stop the EC2 Instance.



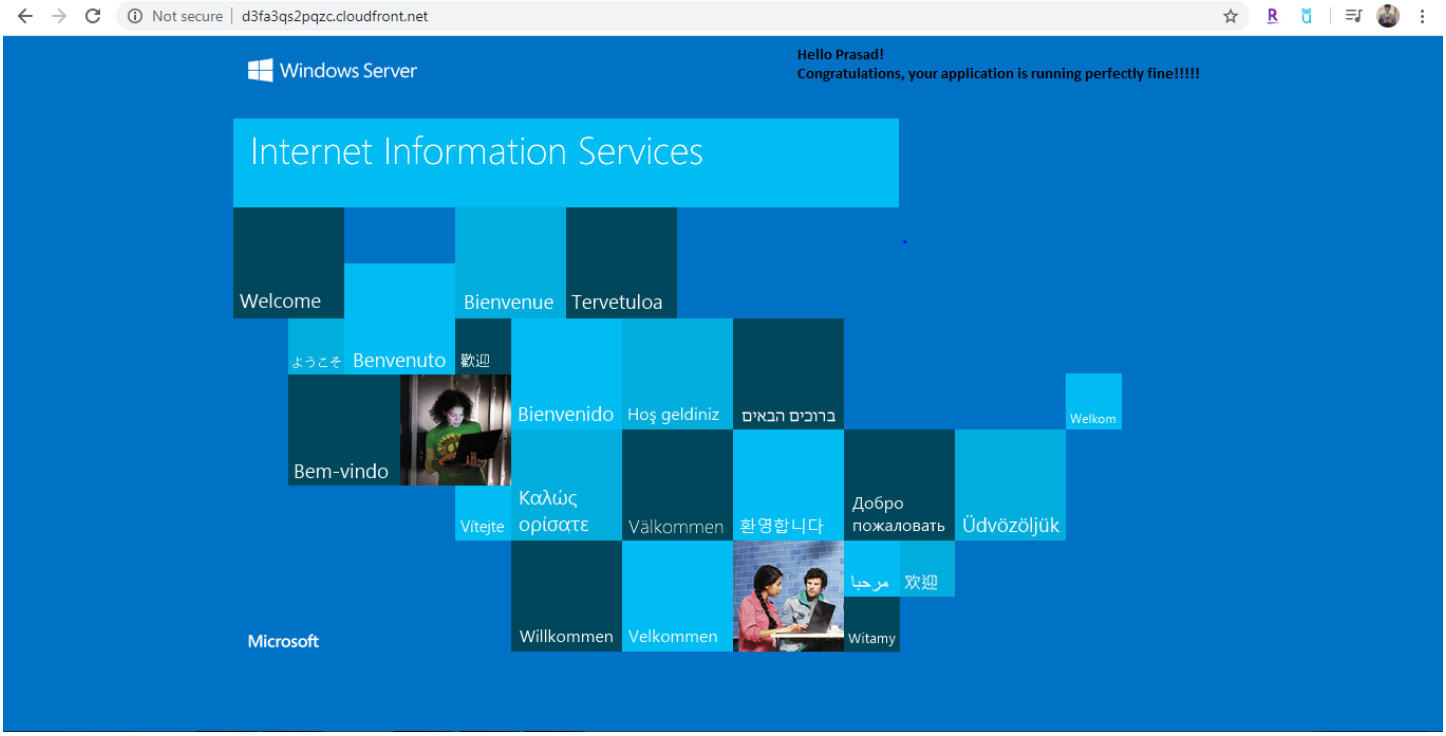
In US EAST (N. Virginia) region, navigate to Route 53 and click on Health Check. You'll see the Health Check as UNHEALTHY. You now have received SNS notification on your Email



Click on Hosted Zones, select your Hosted Zone and click on Test Record Set. You'll now see the Response returned by Route 53 indicates that the Website is now serving by the CloudFront Edge Locations.

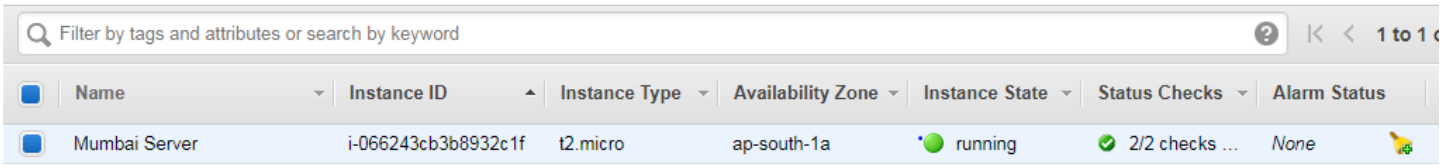


Your website is running perfectly fine even though the entire primary region is down.



We have successfully achieved High Availability across the REGIONS.

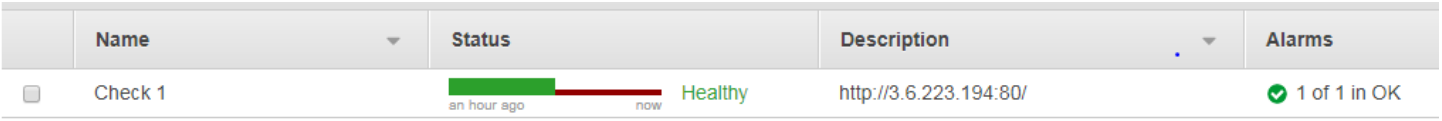
You can now turn on the Web Server in Asia Pacific (Mumbai) region and Test the Record Set again. It should point out to the Primary Web Server which is hosted in Asia Pacific (Mumbai) region.



Filter by tags and attributes or search by keyword — 1 to 1

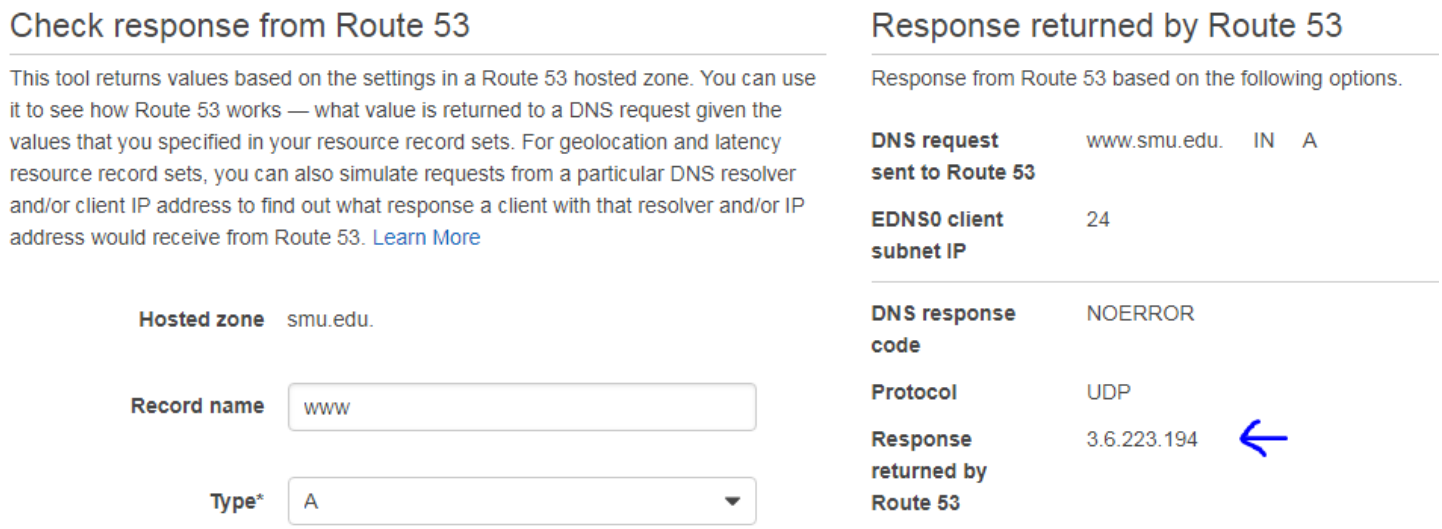
Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status
Mumbai Server	i-066243cb3b8932c1f	t2.micro	ap-south-1a	running	2/2 checks ...	None

Health Check Status is now back to Healthy. You now have received SNS notification on your Email.



Name	Status	Description	Alarms
Check 1	an hour ago — now: Healthy	http://3.6.223.194:80/	1 of 1 in OK

Finally, DNS is now pointing back to the Primary EC2 Instance which is hosted in Asia Pacific (Mumbai) region.



<u>Recommendations:</u>

I real world scenario, if you have a registered Domain then you can host a Website on S3 Bucket and create Secondary Failover Policy for it in Route 53. Hence, even if the Primary region goes down, your website will be still up and running through the S3 Bucket.

This completes the lab on achieving High Availability across Regions.

If you have any questions, contact me on pbhavsar@smu.edu.