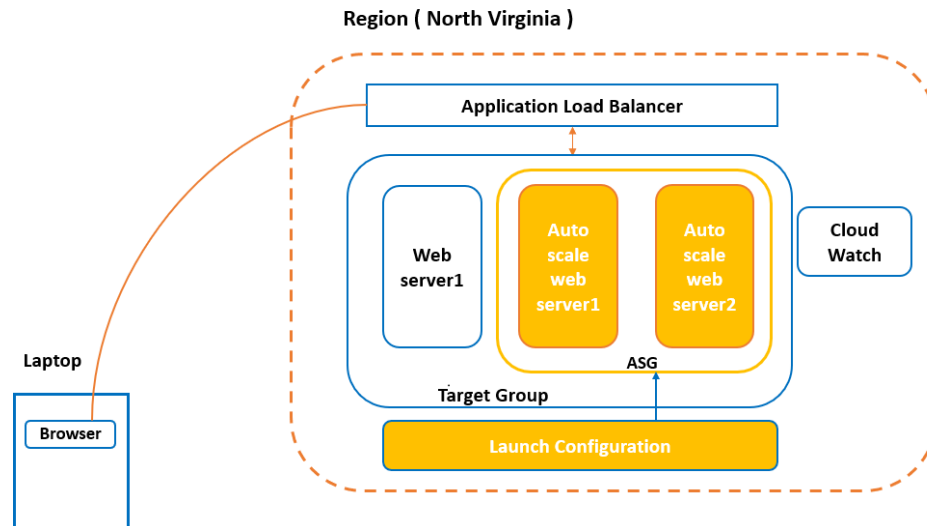


EC2 AUTOSCALING

Deploy multiple EC2 instances using Autoscaling with load balancing & target groups.

FINAL OUTCOME



How to do it?

Create 1 EC2 instance using the 7-step workflow

a) Use the Amazon Linux AMI in AZ1

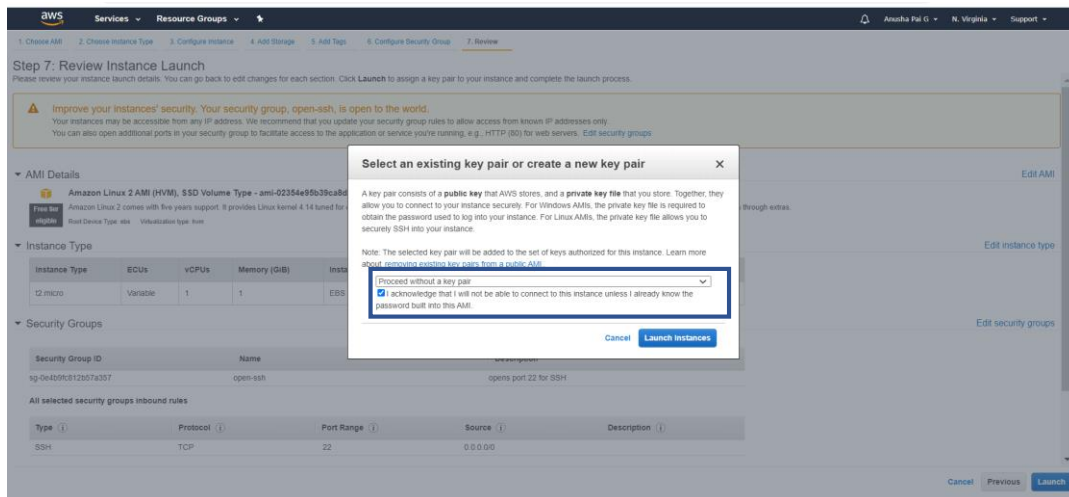
b) Use the bootstrap script to install the http server & to create the index.html and health.html

▼ Advanced Details

Metadata accessible	①	Enabled	▼
Metadata version	①	V1 and V2 (token optional)	▼
Metadata token response hop limit	①	1	▼
User data	①	<input checked="" type="radio"/> As text <input type="radio"/> As file <input type="checkbox"/> Input is already base64 encoded	

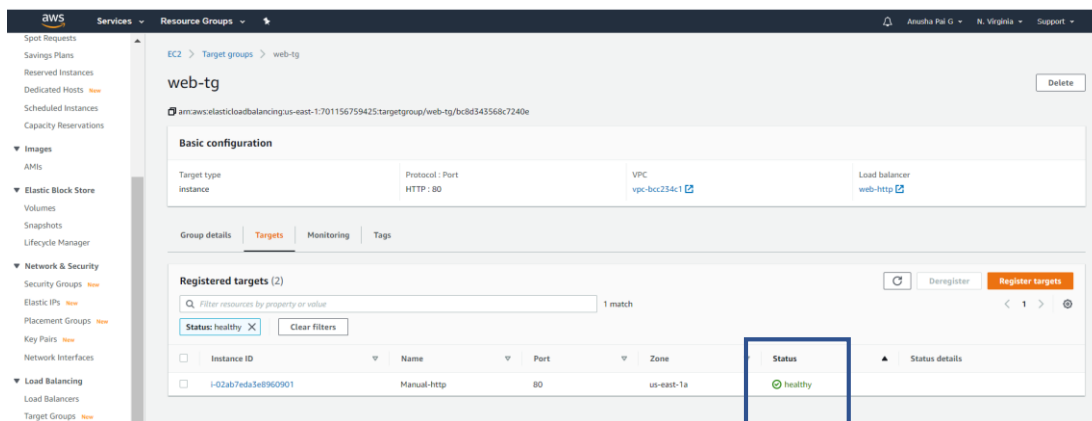
```
#!/bin/bash
yum update -y
yum install httpd -y
service httpd start
chkconfig httpd on
IP_ADDR=$(curl http://169.254.169.254/latest/meta-data/public-ipv4)
echo "Manual instance with IP $IP_ADDR" > /var/www/html/index.html
echo "ok" > /var/www/html/health.html
```

c) No PEM is needed as we are not going to SSH

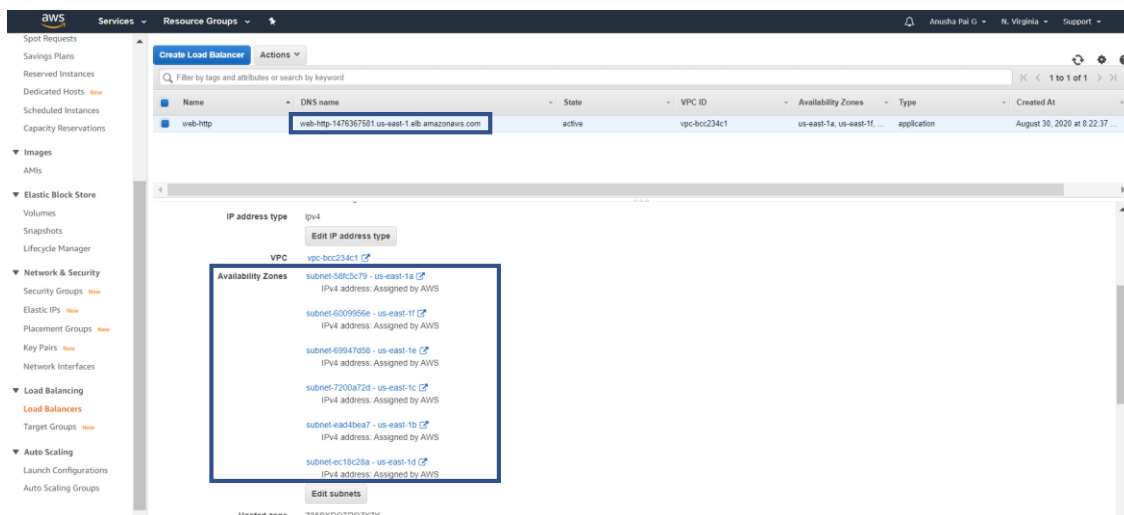


2. Create a LB with TG

a) Associate the EC2 instance to it



b) Ensure all AZ are selected when creating the TG



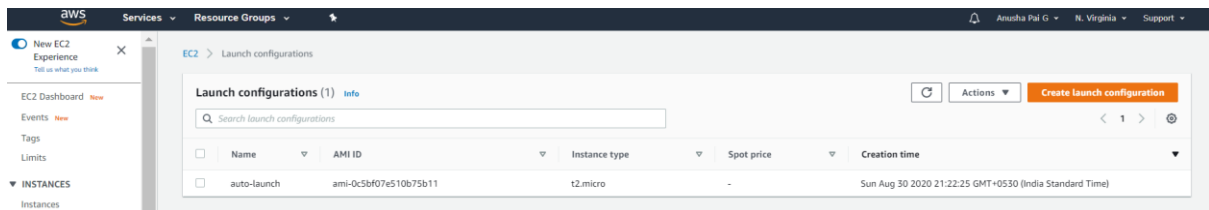
c) Once the instance turns "healthy" hit the LB to ensure the page is being served

← → ↻ Not secure | web-http-1476367581.us-east-1.elb.amazonaws.com

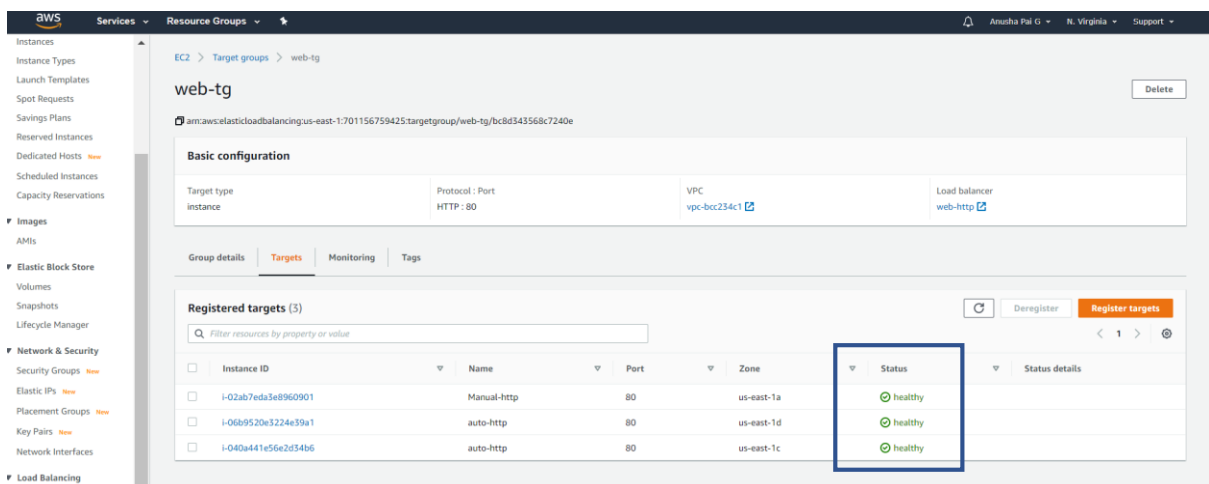
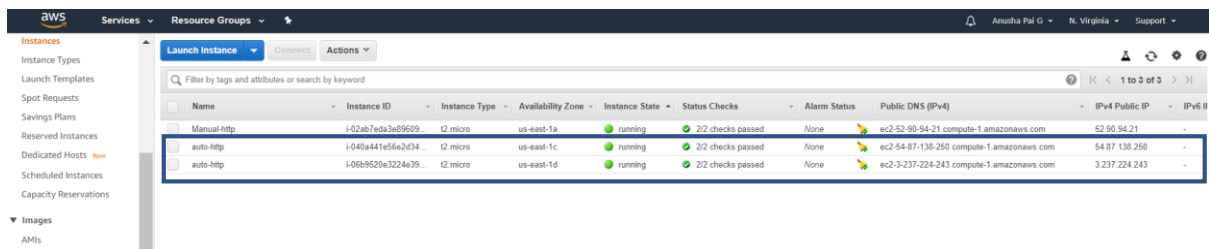
Manual instance with IP 52.90.94.21

3. Create a launch configuration & autoscaling group

a) Use the bootstrap script for the autoscale launch config



b) ASG : Min instance = 1, max = 3 , Desired = 2



4.Hit the LB again and observe the autoscale instance being added to the TG – The request is served by 3 instances i.e. 1 static instance and 2 auto scale instances in Round Robin fashion

5. Manually terminate the autoscale EC2 instance and observe – The desired number of instances will always be provisioned whenever the condition is breached.

aws Services Resource Groups

Launch Instance Connect Actions

Filter by tags and attributes or search by keyword

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP
Manual-http	i-0e384ec9c79aa05e	t2.micro	us-east-1a	pending	Initializing	None	ec2-54-236-41-254.compute-1.amazonaws.com	54.236.41.254
auto-http	i-02ab7eda3e896091	t2.micro	us-east-1a	running	2/2 checks passed	None	ec2-52-90-94-21.compute-1.amazonaws.com	52.90.94.21
auto-http	i-06b9520e3224e39a1	t2.micro	us-east-1d	running	2/2 checks passed	None	ec2-3-237-224-243.compute-1.amazonaws.com	3.237.224.243
auto-http	i-040a441e56a2e34...	t2.micro	us-east-1c	terminated		None		

aws Services Resource Groups

Launch Instance Connect Actions

Filter by tags and attributes or search by keyword

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 IP
Manual-http	i-02ab7eda3e896091	t2.micro	us-east-1a	running	2/2 checks passed	None	ec2-52-90-94-21.compute-1.amazonaws.com	52.90.94.21	
auto-http	i-06b9520e3224e39a1	t2.micro	us-east-1d	running	2/2 checks passed	None	ec2-3-237-224-243.compute-1.amazonaws.com	3.237.224.243	
auto-http	i-0e384ec9c79aa05e	t2.micro	us-east-1e	running	2/2 checks passed	None	ec2-54-236-41-254.compute-1.amazonaws.com	54.236.41.254	
auto-http	i-040a441e56a2e34...	t2.micro	us-east-1c	terminated		None			

aws Services Resource Groups

EC2 > Target groups > web-tg

web-tg Delete

arn:aws:elasticloadbalancing:us-east-1:701156759425:targetgroup/web-tg/bc8d343568c7240e

Basic configuration

Target type	Protocol : Port	VPC	Load balancer
Instance	HTTP : 80	vpc-bcc234c1	web-http

Group details Targets Monitoring Tags

Registered targets (3)

Instance ID	Name	Port	Zone	Status	Status details
i-02ab7eda3e896091	Manual-http	80	us-east-1a	healthy	
i-0e384ec9c79aa05e	auto-http	80	us-east-1e	healthy	
i-06b9520e3224e39a1	auto-http	80	us-east-1d	healthy	