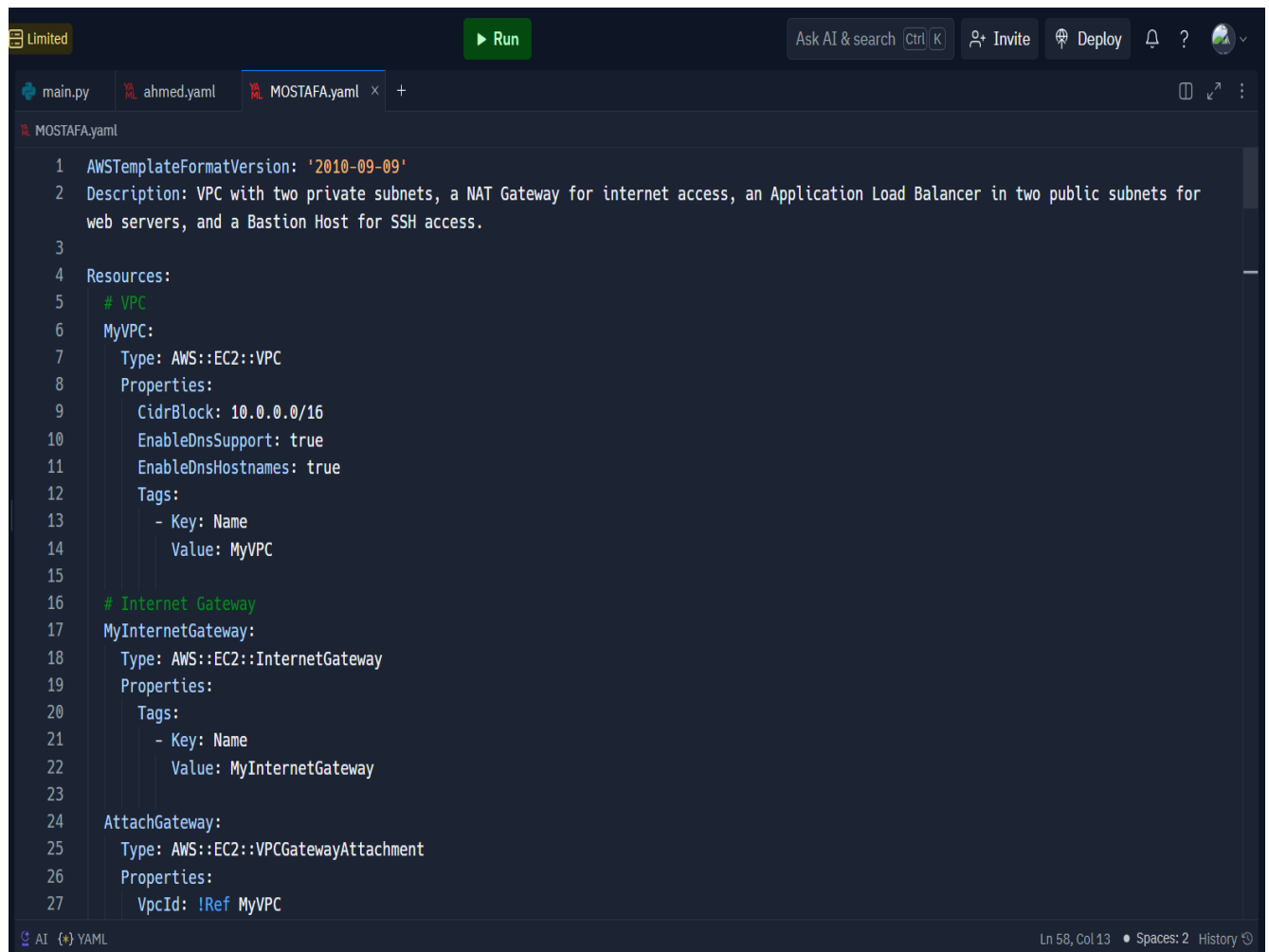


Final Project Documentation:

1. Create Network Environment with Infrastructure as Code (IaC)

Use AWS Cloud Formation or Terraform to define and manage our network infrastructure programmatically.

Use version control for IaC templates, segment our network using Subnets, and secure access with Security Groups and Network ACLs. Deploy a NAT Gateway to allow private instances internet access without exposing them to the public.



```
1 AWSTemplateFormatVersion: '2010-09-09'
2 Description: VPC with two private subnets, a NAT Gateway for internet access, an Application Load Balancer in two public subnets for
  web servers, and a Bastion Host for SSH access.
3
4 Resources:
5   # VPC
6   MyVPC:
7     Type: AWS::EC2::VPC
8     Properties:
9       CidrBlock: 10.0.0.0/16
10      EnableDnsSupport: true
11      EnableDnsHostnames: true
12      Tags:
13        - Key: Name
14          Value: MyVPC
15
16   # Internet Gateway
17   MyInternetGateway:
18     Type: AWS::EC2::InternetGateway
19     Properties:
20       Tags:
21        - Key: Name
22          Value: MyInternetGateway
23
24   AttachGateway:
25     Type: AWS::EC2::VPCGatewayAttachment
26     Properties:
27       VpcId: !Ref MyVPC
```

```
main.py  ahmed.yaml  MOSTAFA.yaml  +
MOSTAFA.yaml
27     VpcId: !Ref MyVPC
28     InternetGatewayId: !Ref MyInternetGateway
29
30 # NAT Gateway
31 NatEIP:
32   Type: AWS::EC2::EIP
33   Properties:
34     Domain: vpc
35
36 MyNatGateway:
37   Type: AWS::EC2::NatGateway
38   Properties:
39     AllocationId: !GetAtt NatEIP.AllocationId
40     SubnetId: !Ref PublicSubnet1
41     Tags:
42       - Key: Name
43         Value: MyNatGateway
44
45 # Route Tables
46 PublicRouteTable:
47   Type: AWS::EC2::RouteTable
48   Properties:
49     VpcId: !Ref MyVPC
50     Tags:
51       - Key: Name
52         Value: PublicRouteTable
53
54 PrivateRouteTable:
```

```
main.py  MOSTAFA.yaml  +
MOSTAFA.yaml
54 PrivateRouteTable:
55   Type: AWS::EC2::RouteTable
56   Properties:
57     VpcId: !Ref MyVPC
58     Tags:
59       - Key: Name
60         Value: PrivateRouteTable
61
62 # Public Route
63 PublicRoute:
64   Type: AWS::EC2::Route
65   Properties:
66     RouteTableId: !Ref PublicRouteTable
67     DestinationCidrBlock: 0.0.0.0/0
68     GatewayId: !Ref MyInternetGateway
69
70 # Private Route
71 PrivateRoute:
72   Type: AWS::EC2::Route
73   Properties:
74     RouteTableId: !Ref PrivateRouteTable
75     DestinationCidrBlock: 0.0.0.0/0
76     NatGatewayId: !Ref MyNatGateway
77
78 # Subnets
79 PublicSubnet1:
80   Type: AWS::EC2::Subnet
81   Properties:
82     VpcId: !Ref MyVPC
```

Ln 54, Col 21 • Spaces: 2 History

```
main.py MOSTAFA.yaml x +
MOSTAFA.yaml
80 Properties:
81   VpcId: !Ref MyVPC
82   CidrBlock: 10.0.1.0/24
83   AvailabilityZone: us-east-1a
84   MapPublicIpOnLaunch: true
85   Tags:
86     - Key: Name
87       Value: PublicSubnet1
88
89 PublicSubnet2:
90   Type: AWS::EC2::Subnet
91   Properties:
92     VpcId: !Ref MyVPC
93     CidrBlock: 10.0.2.0/24
94     AvailabilityZone: us-east-1b
95     MapPublicIpOnLaunch: true
96     Tags:
97       - Key: Name
98         Value: PublicSubnet2
99
100 PrivateSubnet1:
101   Type: AWS::EC2::Subnet
102   Properties:
103     VpcId: !Ref MyVPC
104     CidrBlock: 10.0.3.0/24
105     AvailabilityZone: us-east-1a
106     MapPublicIpOnLaunch: false
107     Tags:
```

```
main.py MOSTAFA.yaml x +
MOSTAFA.yaml
107   Tags:
108     - Key: Name
109       Value: PrivateSubnet1
110
111 PrivateSubnet2:
112   Type: AWS::EC2::Subnet
113   Properties:
114     VpcId: !Ref MyVPC
115     CidrBlock: 10.0.4.0/24
116     AvailabilityZone: us-east-1b
117     MapPublicIpOnLaunch: false
118     Tags:
119       - Key: Name
120         Value: PrivateSubnet2
121
122 PublicSubnet1RouteTableAssociation:
123   Type: AWS::EC2::SubnetRouteTableAssociation
124   Properties:
125     SubnetId: !Ref PublicSubnet1
126     RouteTableId: !Ref PublicRouteTable
127
128 PublicSubnet2RouteTableAssociation:
129   Type: AWS::EC2::SubnetRouteTableAssociation
130   Properties:
131     SubnetId: !Ref PublicSubnet2
132     RouteTableId: !Ref PublicRouteTable
133
134 PrivateSubnet1RouteTableAssociation:
```

Ln 134, Col 39 • Spaces: 2 History

```
main.py MOSTAFA.yaml x +
MOSTAFA.yaml
134 PrivateSubnet1RouteTableAssociation:
135   Type: AWS::EC2::SubnetRouteTableAssociation
136   Properties:
137     SubnetId: !Ref PrivateSubnet1
138     RouteTableId: !Ref PrivateRouteTable
139
140 PrivateSubnet2RouteTableAssociation:
141   Type: AWS::EC2::SubnetRouteTableAssociation
142   Properties:
143     SubnetId: !Ref PrivateSubnet2
144     RouteTableId: !Ref PrivateRouteTable
145
146 # Security Group for Bastion Host
147 BastionSecurityGroup:
148   Type: AWS::EC2::SecurityGroup
149   Properties:
150     GroupDescription: Allow SSH from anywhere
151     VpcId: !Ref MyVPC
152     SecurityGroupIngress:
153       - IpProtocol: tcp
154         FromPort: 22
155         ToPort: 22
156         CidrIp: 0.0.0.0/0 # Limit this to a specific IP for security
157
158 # Bastion EC2 Instance (in Public Subnet)
159 BastionHost:
160   Type: AWS::EC2::Instance
161   Properties:
```

Ln 161, Col 16 • Spaces: 2 History

```
main.py MOSTAFA.yaml x +
MOSTAFA.yaml
159 BastionHost:
160   Type: AWS::EC2::Instance
161   Properties:
162     InstanceType: t2.micro
163     KeyName: vockey # Replace with your KeyPair name
164     ImageId: ami-04d40dba56f6e1303 # Replace with the latest Amazon Linux AMI for your region
165     NetworkInterfaces:
166       - AssociatePublicIpAddress: true
167         DeviceIndex: 0
168         SubnetId: !Ref PublicSubnet1
169         GroupSet:
170           - !Ref BastionSecurityGroup
171     Tags:
172       - Key: Name
173         Value: BastionHost
174
175 # Security Group for Web Servers (allow SSH from Bastion Host)
176 WebServerSG:
177   Type: AWS::EC2::SecurityGroup
178   Properties:
179     GroupDescription: Allow HTTP, HTTPS, and SSH from Bastion
180     VpcId: !Ref MyVPC
181     SecurityGroupIngress:
182       - IpProtocol: tcp
183         FromPort: 80
184         ToPort: 80
185         SourceSecurityGroupId: !Ref ALBSecurityGroup
186       - IpProtocol: tcp
```

Ln 186, Col 26 • Spaces: 2 History

```
main.py MOSTAFA.yaml x +
MOSTAFA.yaml
186     - IpProtocol: tcp
187       FromPort: 443
188       ToPort: 443
189       SourceSecurityGroupId: !Ref ALBSecurityGroup
190     - IpProtocol: tcp
191       FromPort: 22
192       ToPort: 22
193       SourceSecurityGroupId: !Ref BastionSecurityGroup
194
195 # ALB Security Group
196 ALBSecurityGroup:
197   Type: AWS::EC2::SecurityGroup
198   Properties:
199     GroupDescription: Allow HTTP and HTTPS access
200     VpcId: !Ref MyVPC
201     SecurityGroupIngress:
202       - IpProtocol: tcp
203         FromPort: 80
204         ToPort: 80
205         CidrIp: 0.0.0.0/0
206       - IpProtocol: tcp
207         FromPort: 443
208         ToPort: 443
209         CidrIp: 0.0.0.0/0
210
211 # Web Server Launch Template
212 WebServerLaunchTemplate:
213   Type: AWS::EC2::LaunchTemplate
```

Ln 213, Col 26 • Spaces: 2 History

```
main.py MOSTAFA.yaml x +
MOSTAFA.yaml
210
211 # Web Server Launch Template
212 WebServerLaunchTemplate:
213   Type: AWS::EC2::LaunchTemplate
214   Properties:
215     LaunchTemplateName: WebServerLaunchTemplate
216     LaunchTemplateData:
217       KeyName: vockey
218       ImageId: ami-04d40dba56f6e1303
219       InstanceType: t2.micro
220       SecurityGroupIds:
221         - !Ref WebServerSG
222       UserData:
223         Fn::Base64: !Sub |
224           #!/bin/bash
225           yum update -y
226           yum install -y httpd
227           systemctl start httpd
228           systemctl enable httpd
229           echo "<h1>Web Server launched by Auto Scaling Group</h1>" > /var/www/html/index.html
230
231 # Auto Scaling Group for Web Servers
232 WebServerAutoScalingGroup:
233   Type: AWS::AutoScaling::AutoScalingGroup
234   Properties:
235     VPCZoneIdentifier:
236       - !Ref PrivateSubnet1
237       - !Ref PrivateSubnet2
```

Ln 213, Col 26 • Spaces: 2 History

```
main.py MOSTAFA.yaml x +
MOSTAFA.yaml
230     - !Ref PrivateSubnet1
237     - !Ref PrivateSubnet2
238 LaunchTemplate:
239   LaunchTemplateId: !Ref WebServerLaunchTemplate
240   Version: !GetAtt WebServerLaunchTemplate.LatestVersionNumber
241   MinSize: '2'
242   MaxSize: '4'
243   DesiredCapacity: '2'
244   TargetGroupARNs:
245     - !Ref ALBTargetGroup
246   HealthCheckType: ELB
247   HealthCheckGracePeriod: 300
248   Tags:
249     - Key: Name
250       Value: WebServer
251     PropagateAtLaunch: true
252
253 # ALB
254 ALB:
255   Type: AWS::ElasticLoadBalancingV2::LoadBalancer
256   Properties:
257     Name: MyALB
258     Subnets:
259       - !Ref PublicSubnet1
260       - !Ref PublicSubnet2
261     Scheme: internet-facing
262     LoadBalancerAttributes:
263       - Key: idle_timeout.timeout_seconds
264         Value: '60'
```

Ln 213, Col 26 • Spaces: 2 History

```
main.py MOSTAFA.yaml x +
MOSTAFA.yaml
262 LoadBalancerAttributes:
263   - Key: idle_timeout.timeout_seconds
264     Value: '60'
265 SecurityGroups:
266   - !Ref ALBSecurityGroup
267 Tags:
268   - Key: Name
269     Value: MyALB
270
271 ALBTargetGroup:
272   Type: AWS::ElasticLoadBalancingV2::TargetGroup
273   Properties:
274     VpcId: !Ref MyVPC
275     Port: 80
276     Protocol: HTTP
277     TargetType: instance
278     HealthCheckProtocol: HTTP
279     HealthCheckPort: 80
280     HealthCheckPath: /
281     HealthCheckIntervalSeconds: 30
282     HealthCheckTimeoutSeconds: 5
283     HealthyThresholdCount: 3
284     UnhealthyThresholdCount: 2
285     Tags:
286       - Key: Name
287         Value: ALBTargetGroup
288
289 ALBListener:
```

```
main.py MOSTAFA.yaml x +
MOSTAFA.yaml
288
289 ALBListener:
290   Type: AWS::ElasticLoadBalancingV2::Listener
291   Properties:
292     LoadBalancerArn: !Ref ALB
293     Port: 80
294     Protocol: HTTP
295     DefaultActions:
296     - Type: forward
297       TargetGroupArn: !Ref ALBTargetGroup
298
299 Outputs:
300   VPCId:
301     Description: The ID of the VPC
302     Value: !Ref MyVPC
303
304   PublicSubnet1Id:
305     Description: The ID of Public Subnet 1
306     Value: !Ref PublicSubnet1
307
308   PublicSubnet2Id:
309     Description: The ID of Public Subnet 2
310     Value: !Ref PublicSubnet2
311
312   PrivateSubnet1Id:
313     Description: The ID of Private Subnet 1
314     Value: !Ref PrivateSubnet1
315
Generate Ctrl I
AI (Y) YAML Ln 315, Col 1 • Spaces: 2 History
```

```
main.py MOSTAFA.yaml x +
MOSTAFA.yaml
312 PrivateSubnet1Id:
313   Description: The ID of Private Subnet 1
314   Value: !Ref PrivateSubnet1
315
316 PrivateSubnet2Id:
317   Description: The ID of Private Subnet 2
318   Value: !Ref PrivateSubnet2
319
320 LoadBalancerDNSName:
321   Description: The DNS name of the ALB
322   Value: !GetAtt ALB.DNSName
323
324 BastionHostPublicIp:
325   Description: Public IP address of the Bastion Host
326   Value: !GetAtt BastionHost.PublicIp
327

Ln 294, Col 21 • Spaces: 2 History
```

aws

Services

Search

[Alt+S]

N. Virginia

voclabs/user3383544=Bilal_Ali @ 9765-3269-0140

Capacity Reservations

Images

AMIs

AMI Catalog

Elastic Block Store

Volumes

Snapshots

Lifecycle Manager

Network & Security

Security Groups

Elastic IPs

Placement Groups

Key Pairs

Network Interfaces

Load Balancing

Load Balancers

Target Groups

Trust Stores

Auto Scaling

Auto Scaling Groups

EC2 > Target groups

Target groups (1) Info

Filter target groups

try2-ALBTarg-OG4HJSHLLTPHarn:aws:elasticloadbalanci...80HTTPInstanceMyALB

0 target groups selected

Select a target group above.

Activate Windows

Go to Settings to activate Windows.

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Instances:

aws

Services

Search

[Alt+S]

N. Virginia

voclabs/user3383544=Bilal_Ali @ 9765-3269-0140

EC2 Dashboard

EC2 Global View

Events

Console-to-Code

Instances

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity Reservations

Images

AMIs

AMI Catalog

Elastic Block Store

Volumes

Snapshots

Lifecycle Manager

Instances (4) Info

Last updated less than a minute ago

Find Instance by attribute or tag (case-sensitive)

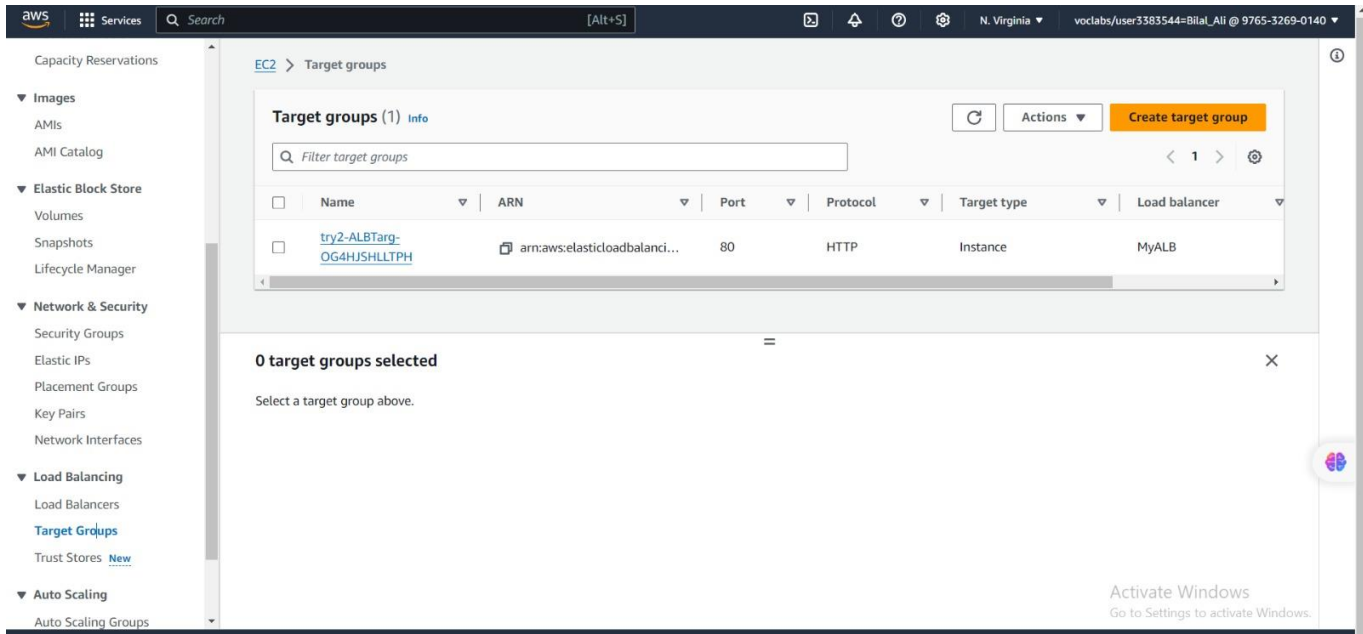
All states

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
	WebServer	i-0bbd88382257283b5	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1b	-
	BastionHost	i-066408876e638a27d	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	ec2-34-195
	WebServer	i-0f4b69a82d4dde96e	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	-
	webserver1	i-07c4533b86a449f6a	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	-

Select an instance

Activate Windows

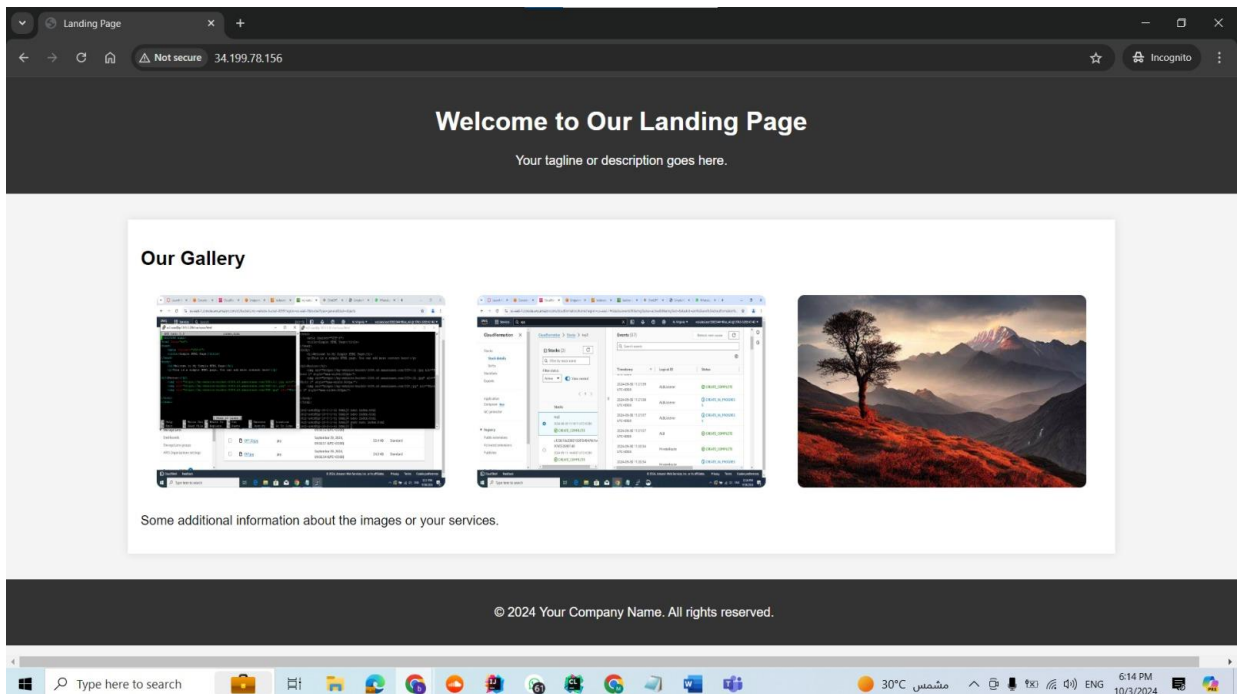
Go to Settings to activate Windows.

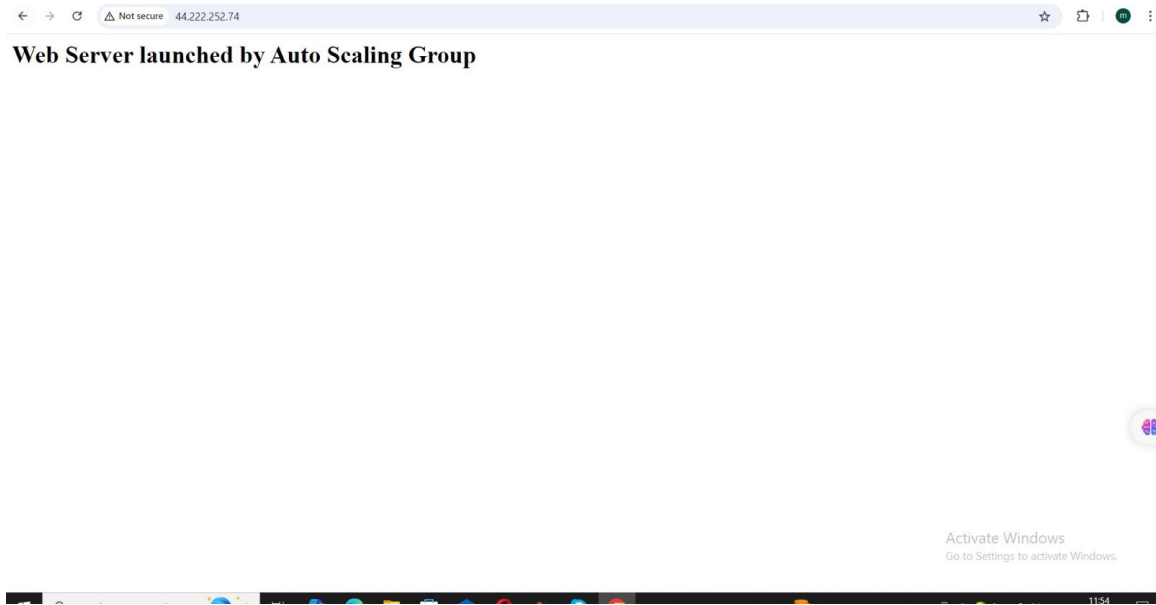


2. Host Web App on EC2 or Using Microservices

Use Amazon EC2 or Amazon ECS/EKS for deploying web apps or containerized microservices.

Use Auto Scaling to ensure high availability and scalability. Deploy Elastic Load Balancers across multiple Availability Zones for redundancy and traffic distribution.





3. Store App Static Content on External Storage (S3)

Enable S3 Versioning for backup and recovery and use CloudFront as a CDN to improve latency and user experience globally.

```
root@ip-10-0-4-155:/var/www/html
GNU nano 5.8 index.html
</head>
<body>

<header>
  <h1>Welcome to Our Landing Page</h1>
  <p>Your tagline or description goes here.</p>
</header>

<div class="container">
  <h2>Our Gallery</h2>
  <div class="image-gallery">
    
    
    
  </div>
  <p>Some additional information about the images or your services.</p>
</div>

<footer>
  <p><strong>copy; 2024 Your Company Name. All rights reserved.</strong></p>
</footer>

</body>
</html>
```

Objects (6) Info

↻

Copy S3 URI

Copy URL

Download

Open

Delete

Actions

Create folder

Upload

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Show versions

< 1 > ⚙

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	bastion+ec2.jpg	jpg	September 30, 2024, 13:32:49 (UTC+03:00)	223.8 KB	Standard
<input type="checkbox"/>	cloudformation.jpg	jpg	September 30, 2024, 13:34:53 (UTC+03:00)	186.3 KB	Standard
<input type="checkbox"/>	newfile.html	html	October 3, 2024, 17:03:14 (UTC+03:00)	14.0 B	Standard
<input type="checkbox"/>	OIP (1).jpg	jpg	September 29, 2024, 09:56:52 (UTC+03:00)	23.6 KB	Standard
<input type="checkbox"/>	OIP (2).jpg	jpg	September 29, 2024, 09:56:51 (UTC+03:00)	33.4 KB	Standard
<input type="checkbox"/>	OIP.jpg	jpg	September 29, 2024, 09:56:54 (UTC+03:00)	24.9 KB	Standard

4. Secure, Scalable, High Availability, and Disaster Recovery

Use IAM for security, Auto Scaling for scalability, and Multi-AZ deployment for high availability.

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Instances:

Services

Search

[Alt+S]

N. Virginia

voclabs/user3383544-Bilal_Ali @ 9765-3269-0140

EC2 Dashboard

EC2 Global View

Events

Console-to-Code

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity Reservations

Images

AMIs

AMI Catalog

Elastic Block Store

Volumes

Snapshots

Lifecycle Manager

Instances (4) Info

Last updated less than a minute ago

Connect

Instance state

Actions

Launch Instances

< 1 > ⚙

Find Instance by attribute or tag (case-sensitive)

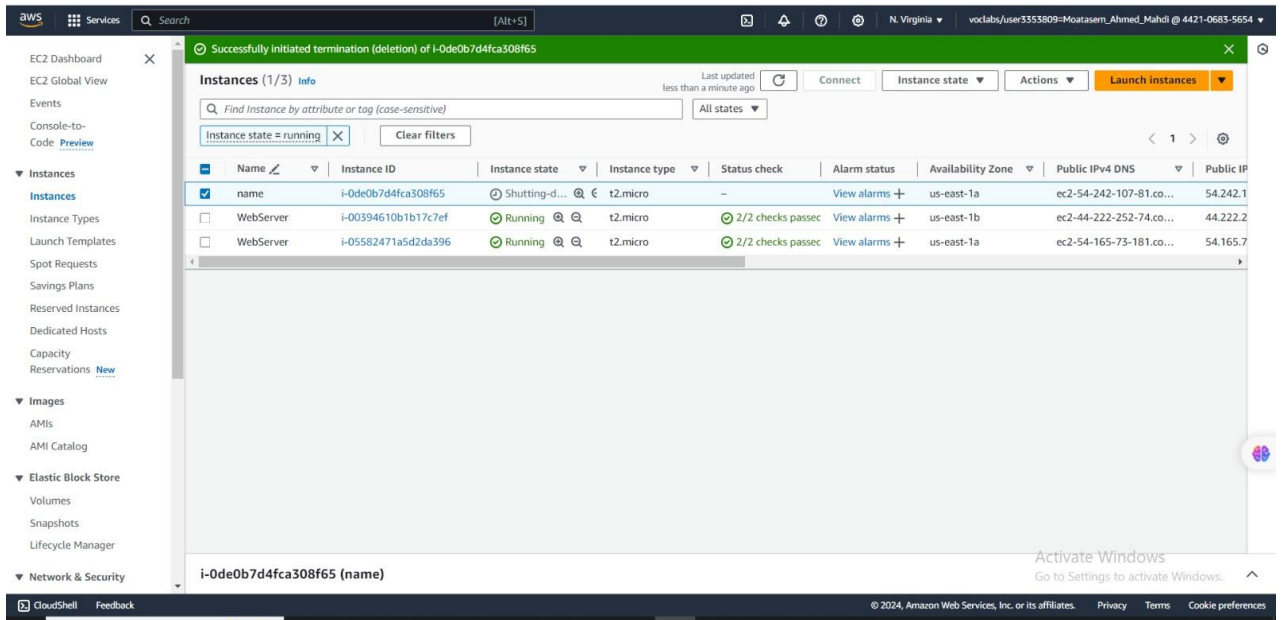
All states

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
<input type="checkbox"/>	WebServer	i-0bbd88582257283b5	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1b	-
<input type="checkbox"/>	BastionHost	i-066408876e638a27d	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	ec2-34-19...
<input type="checkbox"/>	WebServer	i-0f4b69a82d4dde96e	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	-
<input type="checkbox"/>	webserver1	i-07c4533b86a449f6a	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	-

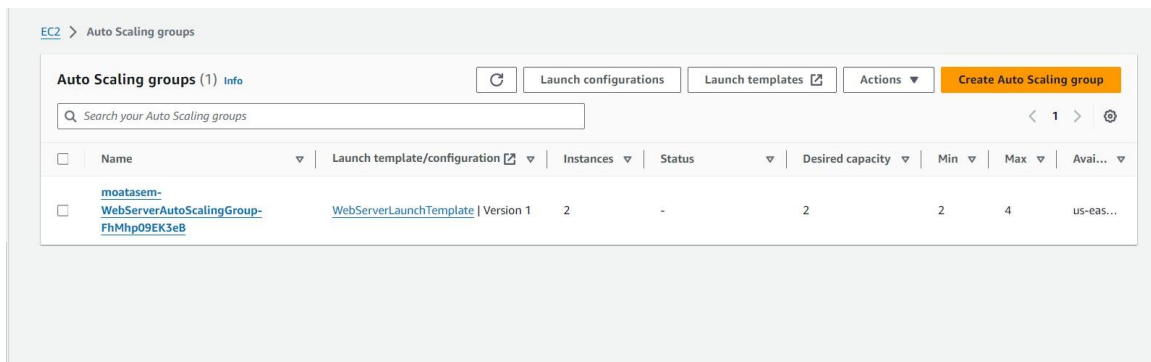
Select an instance

Activate Windows
Go to Settings to activate Windows.

Highly available with two availability zones



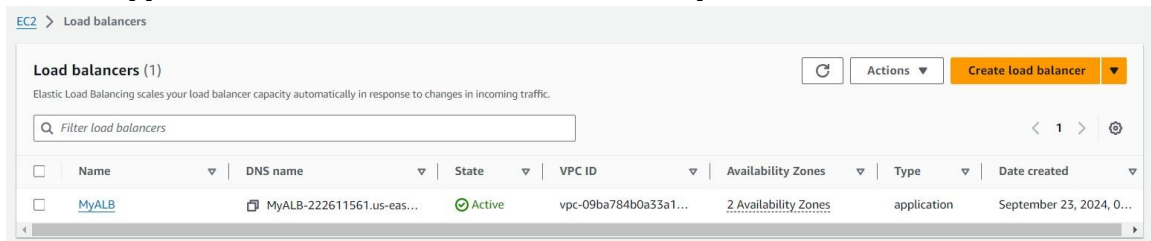
Web server multi-AZ



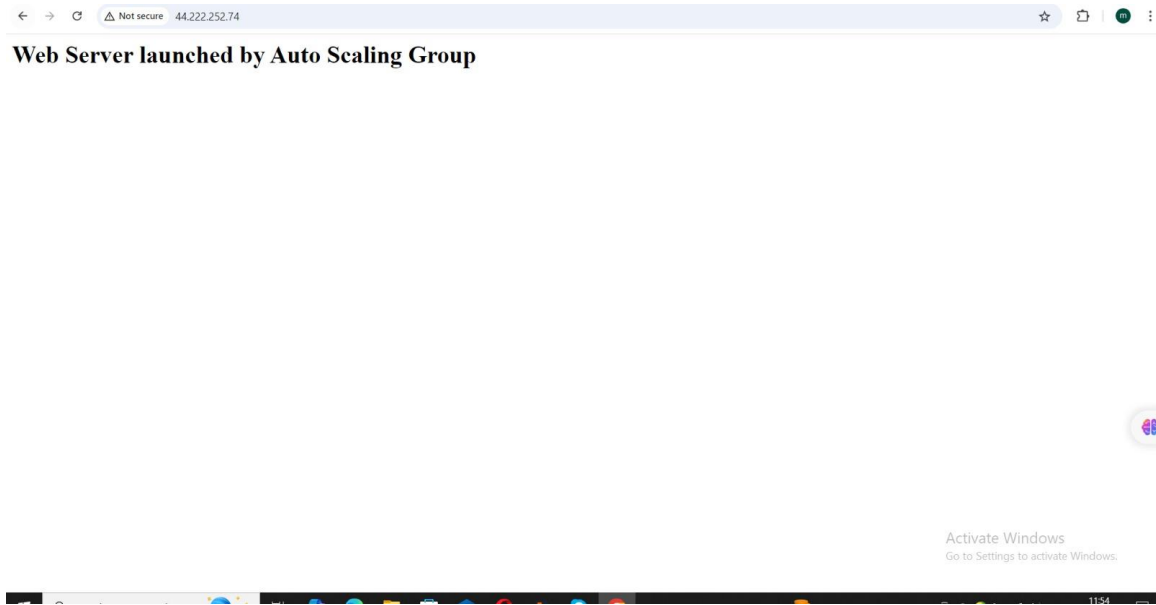
Auto scaling

5. Enable Redirection on Load Balancer (HTTP to HTTPS)

: Use an Application Load Balancer to redirect HTTP requests to HTTPS.



Use SSL/TLS certificates self-signed certificate to ensure encrypted communications.



6. Mount S3 on EC2

: Use s3fs or AWS CLI to mount an S3 bucket on your EC2 instance.

```
ec2-user@ip-10-0-4-155:/var/www/html

Last login: Thu Oct  3 14:32:44 2024 from 41.46.205.232
[ec2-user@ip-10-0-1-61 ~]$ ssh -i labsuser.pem ec2-user@10.0.4.155

A newer release of "Amazon Linux" is available.
  Version 2023.5.20241001:
  Run "/usr/bin/dnf check-release-update" for full release and version update info

#_
~\#####
~~\#####\
~~\####|
~~\#/      https://aws.amazon.com/linux/amazon-linux-2023
~~V~'-'>
~~~
~~~.
~~~\
~~~\m/'

Last login: Thu Oct  3 13:49:42 2024 from 10.0.1.61
[ec2-user@ip-10-0-4-155 ~]$ cd /var/www/html/
[ec2-user@ip-10-0-4-155 html]$ ls
ls: cannot access 'mounted': Permission denied
index.html  mount-s3.rpm  mounted
[ec2-user@ip-10-0-4-155 html]$
```

```
root@ip-10-0-4-155:/var/www/html/mounted
Run "/usr/bin/dnf check-release-update" for full release and version update info ^
      #
    _#_
   _###_
  _####_
 _#####_
#####
\#####\
 \####|
  \###|
   \#/
    V~' '->
   ~~~
  ~~~
 ~~~
  ~~~
 _/m/' '->
Last login: Thu Oct  3 13:49:42 2024 from 10.0.1.61
[ec2-user@ip-10-0-4-155 ~]$ cd /var/www/html/
[ec2-user@ip-10-0-4-155 html]$ ls
ls: cannot access 'mounted': Permission denied
index.html  mount-s3.rpm  mounted
[ec2-user@ip-10-0-4-155 html]$ cd mounted
-bash: cd: mounted: Permission denied
[ec2-user@ip-10-0-4-155 html]$ sudo su
[root@ip-10-0-4-155 html]# cd mounted/
[root@ip-10-0-4-155 mounted]# ls
'OIP (1).jpg'  OIP.jpg      cloudformation.jpg
'OIP (2).jpg'  bastion+ec2.jpg  newfile.html
[root@ip-10-0-4-155 mounted]#
```

user@34.199.78.156

sudo nano /etc/httpd/conf.d/reverse-proxy.conf

sudo apachectl configtest

<VirtualHost *:80>

ServerName 34.199.78.156 # Replace with your Bastion host's public IP

ProxyPreserveHost On

ProxyPass / http://10.0.3.84:80/ # Replace with your EC2 private IP

ProxyPassReverse / http://10.0.3.84:80/

</VirtualHost>

```
s3fs my-website-bucket-0099 ~/s3-bucket -o passwd_file=~/.passwd-s3fs -o allow_other
```

```
sudo s3fs my-website-bucket-0099 /mnt/s3bucket1 -o iam_role=EMR_DefaultRole -o  
use_cache=/tmp -o allow_other -o uid=1001 -o mp_umask=002 -o multireq_max=5 -o  
use_path_request_style -o url=https://s3-{{us-east-1}}.amazonaws.com
```