



Fostering Inclusion

Project Submitted by—Ratna Sai Kumar

Gmail--Ratnasaikumar419@gmail.com

PROJECT - 1

Project: - To set up high availability and auto-scaling for a web application using EC2, ELB (specifically an Application Load Balancer or ALB), and Auto Scaling Group (ASG) in a new VPC and subnet, follow these step Process

step-by-Step Guide

1. Create a New VPC and Subnet

Create a New VPC:

Go to the VPC Dashboard in AWS Management Console.

Click on "Create VPC".

Provide a name for the VPC and an IPv4 CIDR block (e.g., 10.0.0.0/16).

Enable DNS hostnames and DNS resolution.

Telugu Movies (2024) - iBOMM | Gangs of Godavari (2024) DVD | Gangs of Godavari (2024) Telugu | vpcs | VPC Console

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

AWS Services Search [Alt+S] N. Virginia Reddy@2563

Your VPCs (5) Info

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR
-	vpc-0e9b06bda5c837e9a	Available	172.31.0.0/16	-
student_project-vpc	vpc-0b97a29f56ca6de9b	Available	10.0.0.0/16	-
-	vpc-0f12188d9711495de	Available	10.0.0.0/16	-
project_1 minor -website_vpc	vpc-0f3a7a1c0b8ab306c	Available	10.0.0.0/16	-
-	vpc-099f27763e0aa3e0	Available	10.0.0.0/16	-

Select a VPC above

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 38°C Partly sunny ENG IN 12:15 01-06-2024

Telugu Movies (2024) - iBOMM | Gangs of Godavari (2024) DVD | Gangs of Godavari (2024) Telugu | VpcDetails | VPC Console

us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#VpcDetails?VpcId=vpc-0f3a7a1c0b8ab306c

AWS Services Search N. Virginia Reddy@2563

VPC dashboard

EC2 Global View Filter by VPC: Select a VPC

Virtual private cloud Your VPCs Subnets Route tables Internet gateways Egress-only Internet gateways Carrier gateways DHCP option sets Elastic IPs

VPC > Your VPCs > vpc-0f3a7a1c0b8ab306c / project_1 minor -website_vpc Actions

Details

VPC ID vpc-0f3a7a1c0b8ab306c	State Available	DNS hostnames Disabled	DNS resolution Enabled
Tenancy Default	DHCP option set dopt-0280244ed3ab19618	Main route table rtb-0f504938fff6a95a1	Main network ACL acl-0c46dba2a9d7a4b86
Default VPC No	IPv4 CIDR 10.0.0.0/16	IPv6 pool -	IPv6 CIDR (Network border group) -
Network Address Usage metrics Disabled	Route 53 Resolver DNS Firewall rule groups -	Owner ID 533267448465	

Resource map CIDRs Flow logs Tags Integrations

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences SCO - AFG Game score ENG IN 12:20 01-06-2024

2.Create Subnets:

Create at least two subnets in different Availability Zones for high availability.

Provide a name, select the VPC you just created, and assign an IPv4 CIDR block (e.g., 10.0.1.0/24 and 10.0.2.0/24).

Name	Subnet ID	State	VPC
student_website_project1	subnet-0eff9727cc3806cf6	Available	vpc-099f277763e0aa3e0

Create an Internet Gateway:

Create and attach an Internet Gateway to the VPC.

Create a Route Table:

Create a route table and add a route for the Internet Gateway.

Associate the route table with the subnets created.

The screenshot shows the AWS VPC dashboard with the 'Internet gateways' section selected. There are four internet gateways listed:

Name	Internet gateway ID	State	VPC ID
-	igw-00a05db1606cecca2	Attached	vpc-0e9b06bda5c837e9a
-	igw-0e7dc8249e475336	Attached	vpc-0b97a29f56ca6de9b
my_website_gateway-project1	igw-021a8019db97eaa75	Attached	vpc-0f3a7a1c0b8ab306c
my_webhosting_internet_gateway	igw-0a96b2b15a5f59c6f	Attached	vpc-0f12188d9711495de

2. Launch EC2 Instances and Install Software

Launch EC2 Instances:

Go to the EC2 Dashboard and launch two EC2 instances.

Select the newly created VPC and subnets.

Assign a public IP for initial setup.

Choose an appropriate AMI (e.g., Amazon Linux 2).

Choose instance type (e.g., t2.micro).

Configure the security group to allow HTTP (port 80) and SSH (port 22) access.

The screenshot shows the AWS EC2 Instances page. The left sidebar includes links for EC2 Dashboard, EC2 Global View, Events, Console-to-Code Preview, Instances (selected), Instances Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, and Images. The main content area displays a table titled "Instances (3) Info" with columns for Name, Instance ID, Instance state, Instance type, Status check, and Alarm status. Three instances are listed:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status
	i-0451bafb7441ac2e6	Running	t2.micro	2/2 checks passed	View alarms +
My saikumar_ec...	i-0b71a014f0199f3ce	Pending	t2.micro	-	View alarms +
	i-06f6c24deb124a25d	Running	t2.micro	2/2 checks passed	View alarms +

A modal window titled "Select an instance" is open over the table, centered on the pending instance.

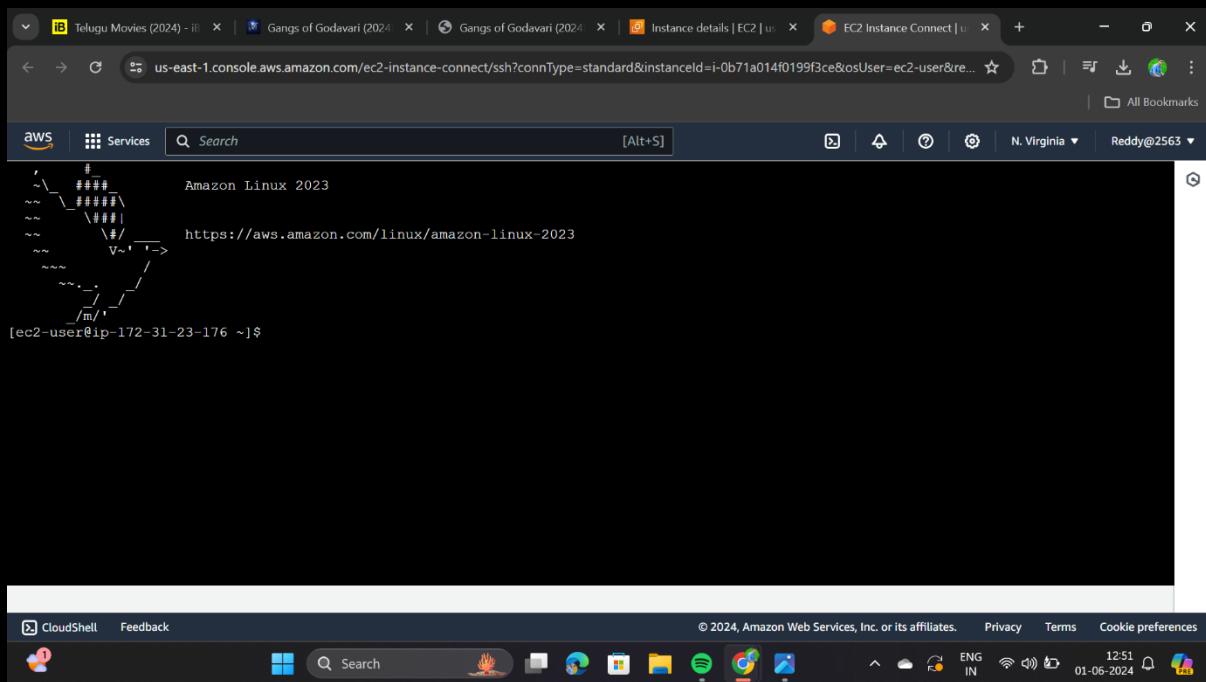
The screenshot shows the AWS EC2 Instance details page for instance i-0b71a014f0199f3ce. The left sidebar is identical to the previous screenshot. The main content area shows the "Instance summary for i-0b71a014f0199f3ce (My saikumar_ec2)" table:

Instance ID	Public IPv4 address	Private IPv4 addresses
i-0b71a014f0199f3ce (My saikumar_ec2)	34.235.129.98 open address	172.31.23.176
IPv6 address	Instance state	Public IPv4 DNS
-	Running	ec2-34-235-129-98.compute-1.amazonaws.com open address
Hostname type	Private IP DNS name (IPv4 only)	Elastic IP addresses
IP name: ip-172-31-23-176.ec2.internal	ip-172-31-23-176.ec2.internal	-
Answer private resource DNS name	Instance type	AWS Compute Optimizer findings
IPv4 (A)	t2.micro	-
Auto-assigned IP address	VPC ID	

Ec2 launch create file

After instances will be connect to the attach

Next amazon linux



```
Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023
[ec2-user@ip-172-31-23-176 ~]$
```

SSH into each EC2 instance and install your web application and required software.

Example for a simple Apache setup:

sh

Copy code

`sudo yum update -y`

`sudo yum install -y httpd`

`sudo systemctl start httpd`

`sudo systemctl enable httpd`

`echo "<h1>Web Server $(hostname -f)</h1>" | sudo tee /var/www/html/index.html`

Telugu Movies (2024) - iBOMM | Launch an instance | EC2 | us-east-1 | EC2 Instance Connect | us-east-1 | +

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

AWS Services Search [Alt+S] N. Virginia Reddy@2563

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name Add additional tags

Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Summary

Number of instances Info
1

Amazon Linux 2023 AMI 2023.4.2... read more
ami-00beae93a2d981137

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)

Cancel **Launch instance** Review commands

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 13:00 01-06-2024

Telugu Movies (2024) - iBOMM | Launch an instance | EC2 | us-east-1 | EC2 Instance Connect | us-east-1 | +

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

AWS Services Search [Alt+S] N. Virginia Reddy@2563

VPC - required Info

vpc-0e9b06bda5c837e9a (default)
172.31.0.0/16

Subnet Info
subnet-0bbfe5352fa6d3854
VPC: vpc-0e9b06bda5c837e9a Owner: 533267448465 Availability Zone: us-east-1a IP addresses available: 4091 CIDR: 172.31.80.0/20

Create new subnet

Auto-assign public IP Info
Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups) Info
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group Select existing security group

Security group name - required
launch-wizard-4

Summary

Number of instances Info
1

Software image (AMI)
Amazon Linux 2023 AMI
ami-00beae93a2d981137

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 13:03 01-06-2024

The screenshot shows two browser windows for the AWS Cloud Console.

Top Window: The URL is `us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#InstanceDetails:instanceId=i-08324dd22c331bb73`. The title is "Instance summary for i-08324dd22c331bb73 (website_project_ec2)". The page displays instance details including:

- Instance ID: i-08324dd22c331bb73 (website_project_ec2)
- Public IPv4 address: 184.73.20.91
- Private IPv4 addresses: 172.31.80.87
- IPv6 address: -
- Instance state: Running
- Public IPv4 DNS: ec2-184-73-20-91.compute-1.amazonaws.com
- Hostname type: IP name: ip-172-31-80-87.ec2.internal
- Private IP DNS name (IPv4 only): ip-172-31-80-87.ec2.internal
- Answer private resource DNS name: -
- Instance type: t2.micro
- VPC ID: -
- Elastic IP addresses: -

Bottom Window: The URL is `us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#ConnectToInstance:instanceId=i-08324dd22c331bb73`. The title is "EC2 Instance Connect | us-east-1". It shows options to connect via EC2 Instance Connect, Session Manager, or SSH client. The SSH client tab is selected, showing the command to run: `ssh -i "website_project_ec2.pem" ec2-user@ec2-184-73-20-91.compute-1.amazonaws.com`. A note states: "Note: In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username." The command `ssh -i "website_project_ec2.pem" ec2-user@ec2-184-73-20-91.compute-1.amazonaws.com` is highlighted with a green border.

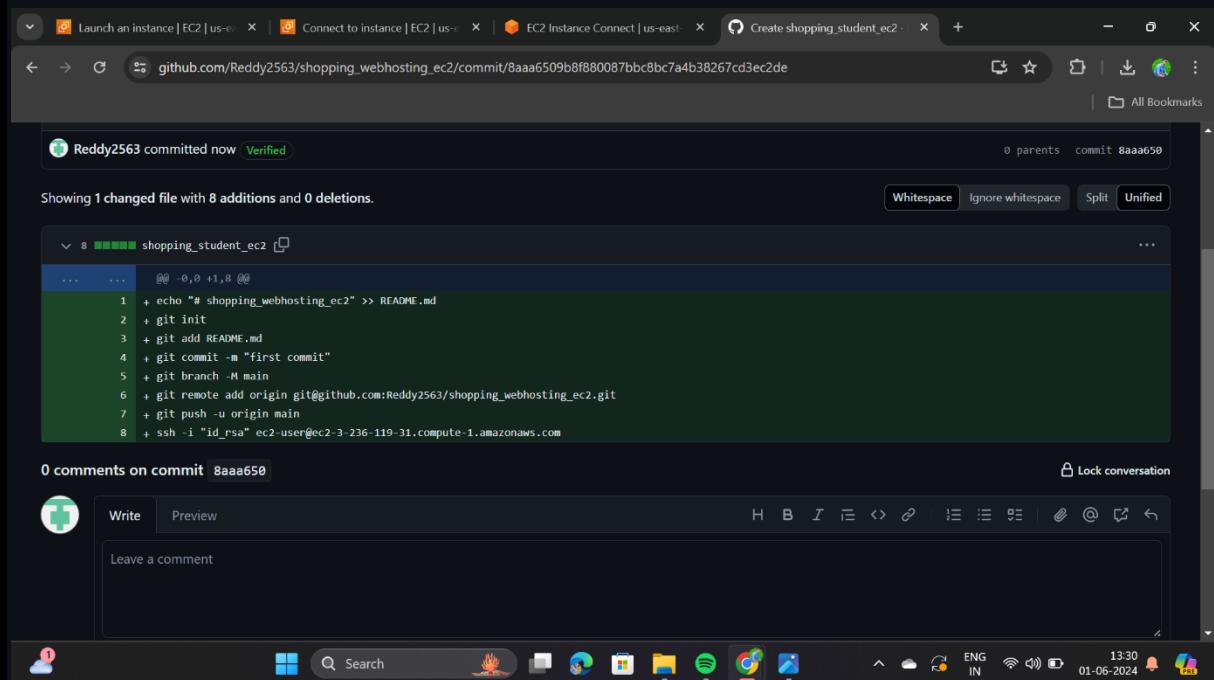
Command prompt for connecting to github

New file code create in command changes

Repository name new create
After public

Repository create

Github:shopping_website_project_ec2

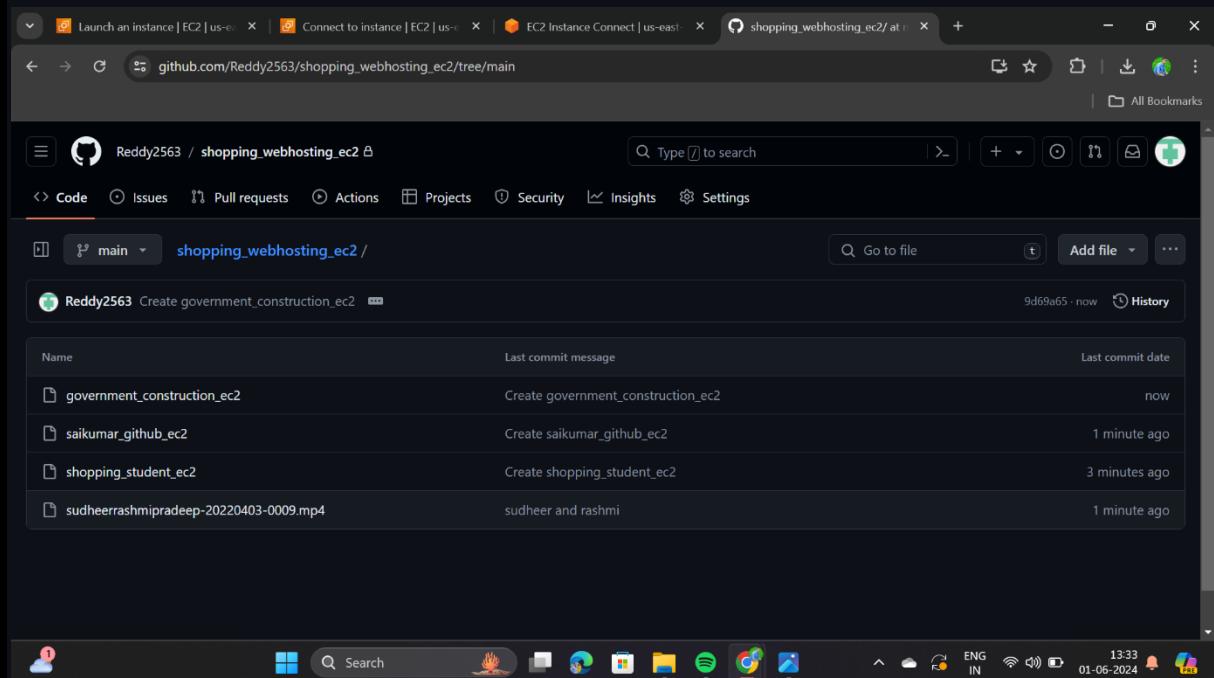


Showing 1 changed file with 8 additions and 0 deletions.

```
diff --git a/README.md b/README.md
@@ -0,0 +1,8 @@
1+ echo "# shopping_webhosting_ec2" >> README.md
2+ git init
3+ git add README.md
4+ git commit -m "first commit"
5+ git branch -M main
6+ git remote add origin git@github.com:Reddy2563/shopping_webhosting_ec2.git
7+ git push -u origin main
8+ ssh -i "id_rsa" ec2-user@ec2-3-236-119-31.compute-1.amazonaws.com
```

0 comments on commit 8aaa650

Leave a comment



Type / to search

Go to file

Add file

Name	Last commit message	Last commit date
government_construction_ec2	Create government_construction_ec2	now
saikumar.github_ec2	Create saikumar.github_ec2	1 minute ago
shopping_student_ec2	Create shopping_student_ec2	3 minutes ago
sudheerashmipradeep-20220403-0009.mp4	sudheer and rashmi	1 minute ago

Github: new create file command changes Shopping_webhosting

3. Configure Application Load Balancer (ALB)

Create an ALB:

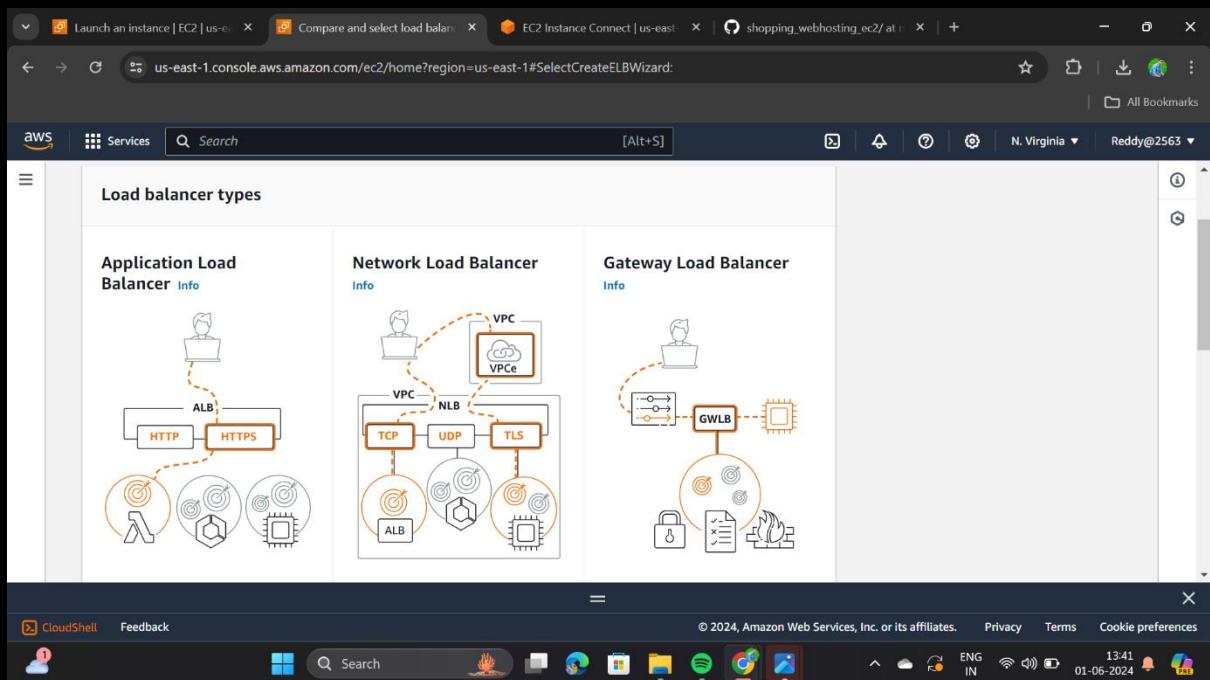
Go to the EC2 Dashboard and select "Load Balancers".

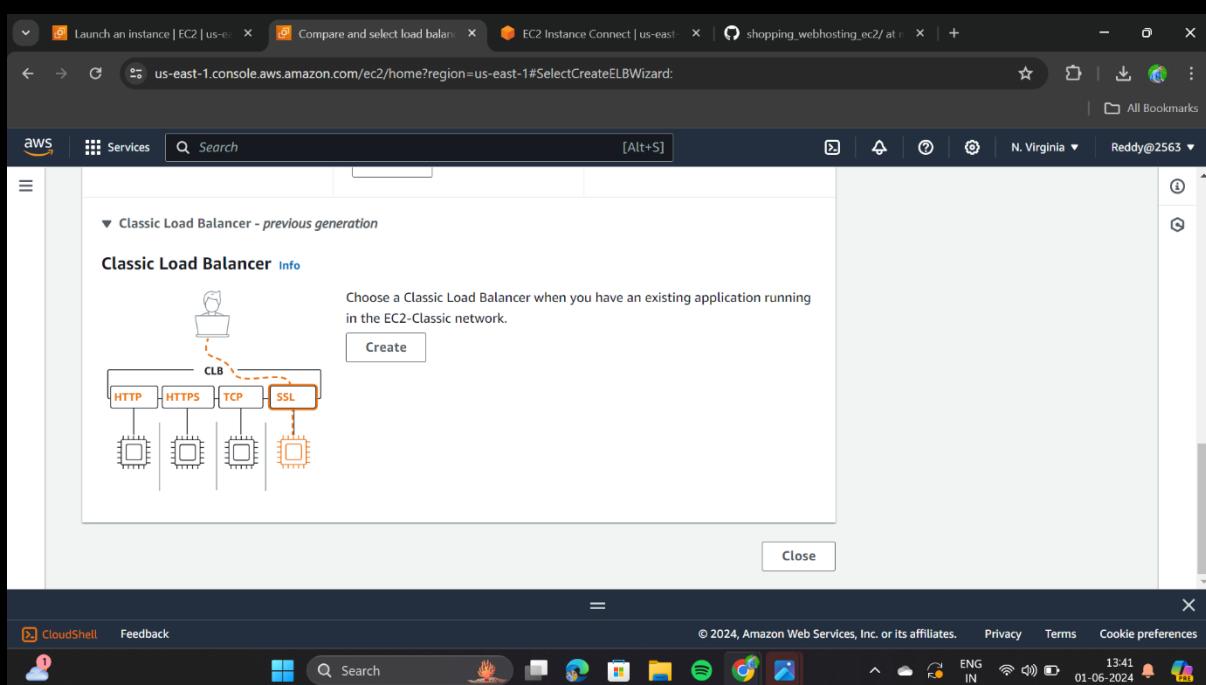
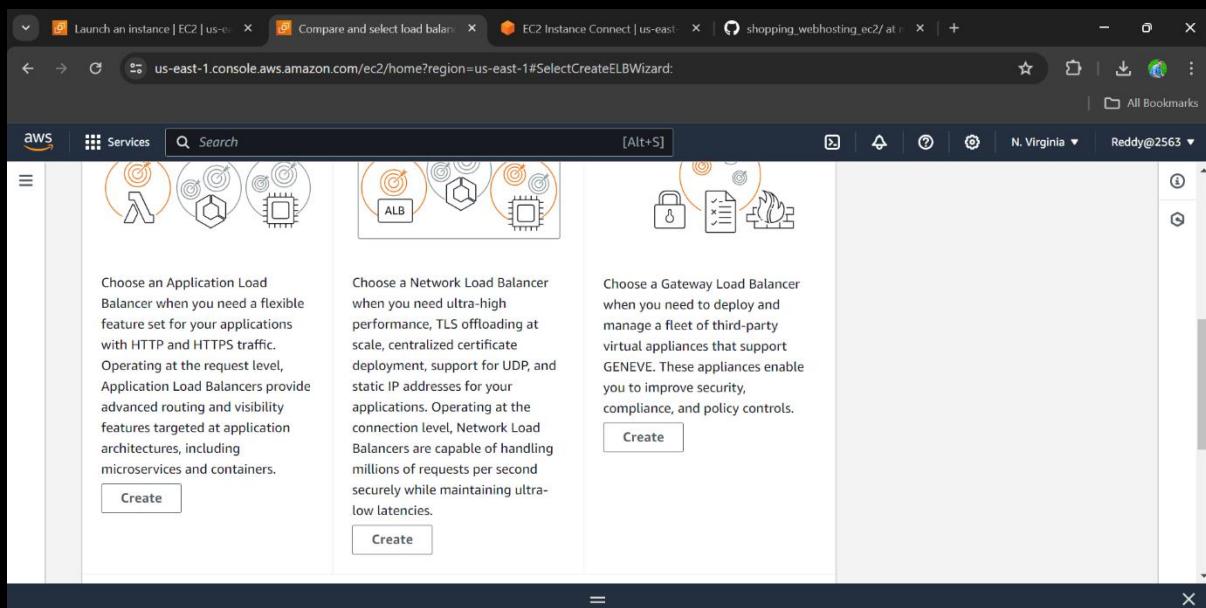
Create a new Application Load Balancer.

Choose "internet-facing", select the VPC, and add the subnets.

Configure security groups to allow HTTP (port 80).

Create a target group for the EC2 instances.





Register EC2 Instances with the ALB:

Register the launched EC2 instances to the target group.

Configure Health Checks:

Configure health checks to ensure traffic is only routed to healthy instances.

The screenshot shows the AWS CloudShell interface with multiple tabs open. The active tab is titled "Create application load balancer" and displays the "Basic configuration" section of the wizard. It asks for the "Load balancer name" (set to "application"), "Scheme" (set to "Internet-facing"), and other details. Below this, there's a "How Application Load Balancers work" section and a "Next Step" button.

The screenshot shows the AWS CloudShell interface with multiple tabs open. The active tab is titled "Step 1 Create target group" and displays the "Target type" section of the wizard. It lists three options: "IP addresses", "Lambda function", and "Application Load Balancer". The "Application Load Balancer" option is selected. Below it, there's a "Target group name" input field and a "Next Step" button.

The screenshot shows the AWS CloudShell interface with a browser window open to the EC2 home page. A modal dialog is displayed for creating a new Application Load Balancer. The configuration includes:

- Load balancer type: Application
- Status: Provisioning
- VPC: [vpc-0e9b06bda5c837e9a](#)
- IP address type: IPv4
- Scheme: Internet-facing
- Hosted zone: Z35SXDOTRQ7X7K
- Availability Zones:
 - [subnet-0bbfe5352fa6d3854](#) us-east-1a (use1-az2)
 - [subnet-0633535a69891a407](#) us-east-1a (use1-az2)
- Date created: June 1, 2024, 13:51 (UTC+05:30)
- Load balancer ARN: [arn:aws:elasticloadbalancing:us-east-1:533267448465:loadbalancer/app/Application/a63e46900c1ef5ef](#)
- DNS name copied: Application-388142079.us-east-1.elb.amazonaws.com (A Record)

The screenshot shows the AWS CloudShell interface with a browser window open to the EC2 home page. A modal dialog is displayed for selecting an Amazon Machine Image (AMI). The selected AMI is:

- Amazon Machine Image (AMI): al2023-ami-2023.4.20240528.0-kernel-6.1-
x86_64
ami-00beae93a2d981137
- Free tier eligible
- Verified provider
- Browse more AMIs
- Including AMIs from AWS, Marketplace and the Community

Details for the AMI:

Catalog	Published	Architecture	Virtualization	Root device	ENI Enabled
Quickstart AMIs	2024-05-24T03:27:53.00Z	x86_64	hvm	type ebs	Yes

Instance type settings:

- Instance type: t2.micro
- Info | Get advice
- Advanced
- All generations

Summary section:

- Firewall (security group): default
- Storage (volumes): 1 volume(s) - 8 GiB

A tooltip for the Free tier information:

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GiB of

4. Create an Auto Scaling Group (ASG)

Create a Launch Template:

Go to the EC2 Dashboard and create a launch template with the configuration of your EC2 instances.

Create an Auto Scaling Group:

Go to the Auto Scaling Groups section and create a new Auto Scaling Group.

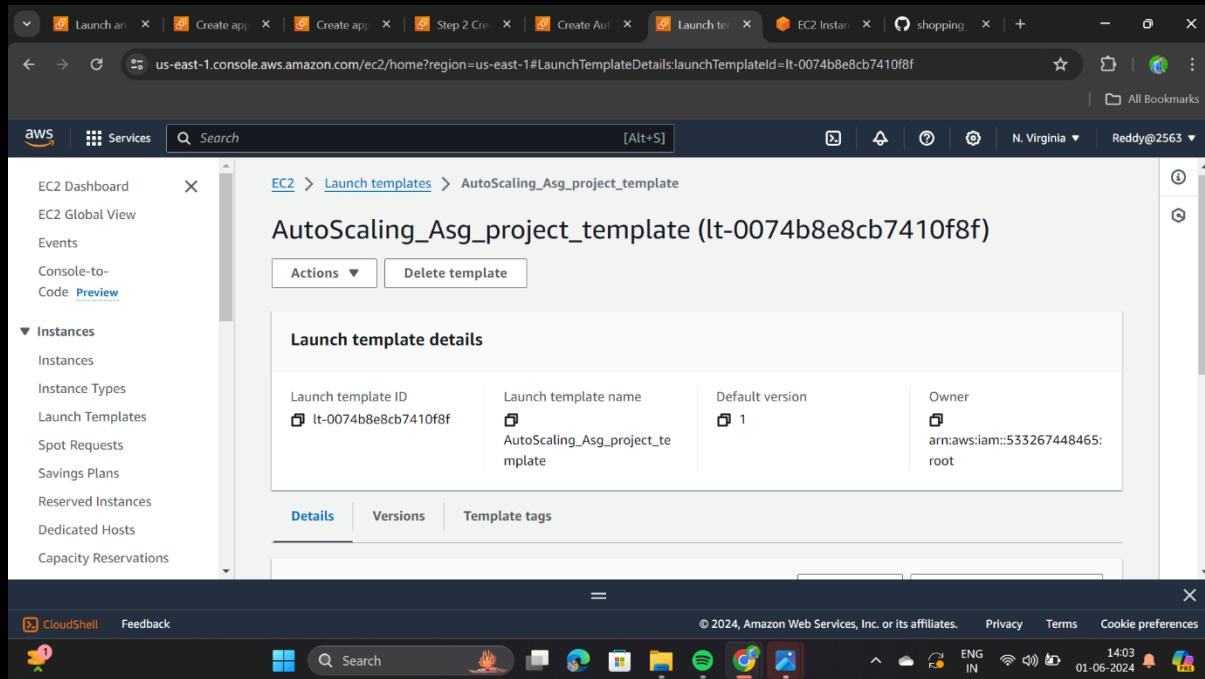
Select the launch template created.

Choose the VPC and subnets.

Attach the ALB to the Auto Scaling Group.

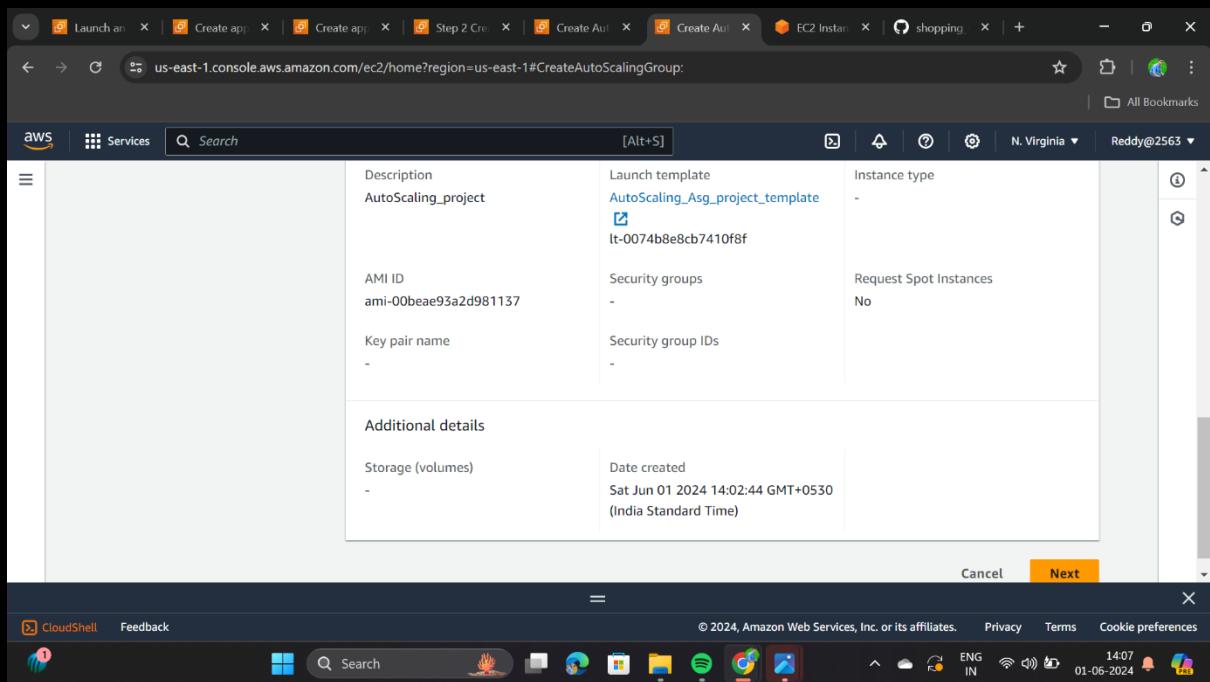
Configure the desired, minimum, and maximum number of instances.

Set up scaling policies based on metrics like CPU utilization.



The screenshot shows the AWS Auto Scaling Group creation wizard at Step 5. The main configuration pane is titled "Scaling" with an "Info" link. It asks for the "Desired capacity" (set to 2) and "Scaling limits". Under "Scaling limits", the "Min desired capacity" is set to 1 and the "Max desired capacity" is set to 2. Below this, there are two options: "Equal or less than desired capacity" and "Equal or greater than desired capacity". A note states: "You can resize your Auto Scaling group manually or automatically to meet changes in demand." On the left sidebar, steps 5 through 7 are listed: "Step 5 - optional Add notifications", "Step 6 - optional Add tags", and "Step 7 Review". The status bar at the bottom indicates the user is in the N. Virginia region and has 14:07 remaining.

The screenshot shows the AWS Auto Scaling Group creation wizard at Step 3. The main configuration pane is titled "Auto Scaling group name" with an "Info" link. It asks for a name ("AutoScaling_project1") which must be unique. A note states: "For accounts created after May 31, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023." Below this, the "Launch template" section is shown, with a note: "Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups." A dropdown menu shows "AutoScaling_Asg_project_template". On the left sidebar, steps 3 through 7 are listed: "Step 3 - optional Configure advanced options", "Step 4 - optional Configure group size and scaling", "Step 5 - optional Add notifications", "Step 6 - optional Add tags", and "Step 7 Review". The status bar at the bottom indicates the user is in the N. Virginia region and has 14:07 remaining.



5. Attach EBS Volumes

Create EBS Volumes:

Create EBS volumes in the same Availability Zone as your EC2 instances.

Attach EBS Volumes to EC2 Instances:

Go to the EC2 Dashboard, select an instance, and attach the EBS volume.

Mount the EBS Volumes:

SSH into each EC2 instance and mount the EBS volume:

sh

Copy code

```
sudo mkfs -t xfs /dev/xvdh
```

```
sudo mkdir /mnt/data
```

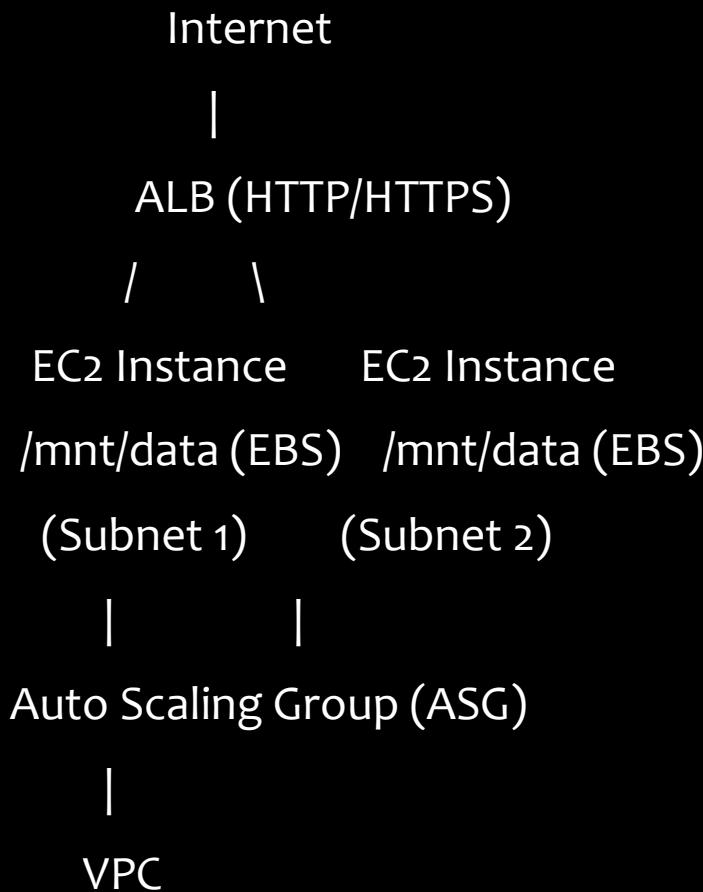
```
sudo mount /dev/xvdh /mnt/data
```

Diagram

Here's a high-level architecture diagram for the setup:

SCSS

Copy code



Screenshot of the AWS CloudShell interface showing the EC2 Volumes page.

The top navigation bar shows the URL: `us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Volumes:`

The sidebar menu includes:

- EC2 Dashboard
- EC2 Global View
- Events
- Console-to-Code [Preview](#)
- Instances
- Images
 - AMIs
 - AMI Catalog
- Elastic Block Store
 - Volumes**
 - Snapshots
 - Lifecycle Manager

The main content area displays the "Volumes (7) Info" table:

Name	Volume ID	Type	Size	IOPS	Throughput	Snapshot
-	vol-0f67a8709ec277195	gp3	8 GiB	3000	125	snap-0d10865
-	vol-03415d986e401b698	gp3	8 GiB	3000	125	snap-029f30d...
-	vol-0c42b43adea5ca170	gp3	100 GiB	3000	125	-
-	vol-0c8d4c21086c4e866	gp3	8 GiB	3000	125	snap-0d10865
-	vol-0aa4b2a5a3279ed6e	gp3	100 GiB	3000	125	-
-	vol-05814e908bfab9038	gp3	100 GiB	3000	125	-
-	vol-05c19efb296f7da24	gp3	8 GiB	3000	125	snap-029f30d...

A summary message at the bottom states: "Summary for all volumes in this Region".

The bottom navigation bar includes CloudShell, Feedback, and various system icons.

The screenshot shows the AWS EC2 Volumes page. The left sidebar includes links for EC2 Dashboard, EC2 Global View, Events, Instances, Images, AMIs, AMI Catalog, and Elastic Block Store (with Volumes selected). The main content area displays a table titled 'Volumes (7) Info' with columns: resources, Volume state, Encryption, KMS key ID, KMS key alias, Fast snapshots, and Multi-Attache. All volumes are listed as 'Okay' and not encrypted. A summary at the bottom states 'Summary for all volumes in this Region'.

resources	Volume state	Encryption	KMS key ID	KMS key alias	Fast snapshots	Multi-Attache
141019941611... /de...	Okay	Not encrypted	-	-	No	No
fb7441ac2e6: /de...	Okay	Not encrypted	-	-	No	No
d22c331bb73 (we...	Okay	Not encrypted	-	-	No	No
d22c331bb73 (we...	Okay	Not encrypted	-	-	No	No
d22c331bb73 (we...	Okay	Not encrypted	-	-	No	No
deb124a25d (stu...	Okay	Not encrypted	-	-	No	No

Conclusion:

By following these steps, you will have a web application setup that is highly available and scalable, using EC2 instances distributed across multiple subnets, an Application Load Balancer to distribute traffic, and an Auto Scaling Group to handle dynamic scaling. EBS volumes ensure that each instance has its own storage for application-specific data. Conclusion

By following these steps, you will have a web application setup that is highly available and scalable, using EC2 instances distributed across multiple subnets, an Application Load Balancer to distribute traffic, and an Auto Scaling Group to handle dynamic scaling. EBS volumes ensure that each instance has its own storage for application-specific data