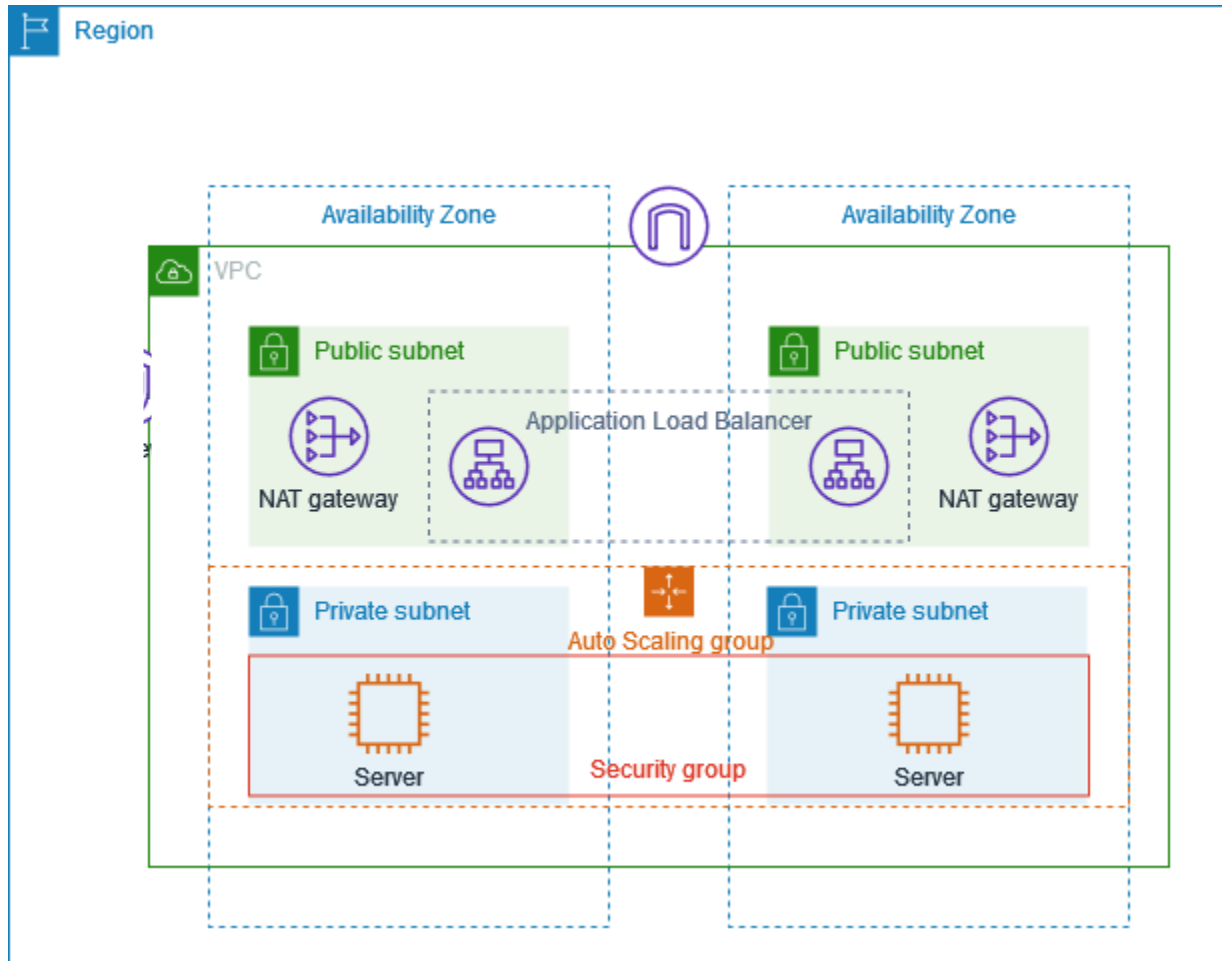


## VPC MINI PROJECT

The following diagram provides an overview of the resources included in this example. The VPC has public subnets and private subnets in two Availability Zones. Each public subnet contains a NAT gateway and a load balancer node. The servers run in the private subnets, are launched and terminated by using an Auto Scaling group, and receive traffic from the load balancer. The servers can connect to the internet by using the NAT gateway. The servers can connect to Amazon S3 by using a gateway VPC endpoint.



The provided image depicts a simplified diagram of a Virtual Private Cloud (VPC) on Amazon Web Services (AWS). It illustrates the key components and their relationships to form a secure and scalable cloud computing environment.

This simplified VPC diagram demonstrates the fundamental structure of a secure and scalable cloud computing environment on AWS. It highlights the use of public and private subnets, load balancing, network access control, and auto scaling to manage resources effectively and protect sensitive data.

## CREATING VPC

### Step 1: Create a VPC

1. Navigate to the AWS Management Console.
2. Go to "Services" > "VPC."
3. Click on "Your VPCs" in the left navigation pane.
4. Click the "Create VPC" button.
5. Click on vpc and more.
6. Follow the below vpc setting for creating vpc

### VPC settings

**Resources to create** [Info](#)  
Create only the VPC resource or the VPC and other networking resources.

☐ VPC only

☒ VPC and more

---

**Name tag auto-generation** [Info](#)  
Enter a value for the Name tag. This value will be used to auto-generate Name tags for all resources in the VPC.

☒ Auto-generate

mc-vpc

---

**IPv4 CIDR block** [Info](#)  
Determine the starting IP and the size of your VPC using CIDR notation.

10.0.0.0/1665,536 IPs

CIDR block size must be between /16 and /28.

---

**IPv6 CIDR block** [Info](#)

☒ No IPv6 CIDR block

☐ Amazon-provided IPv6 CIDR block

Tenancy [Info](#)

Default ▼

Number of Availability Zones (AZs) [Info](#)

Choose the number of AZs in which to provision subnets. We recommend at least two AZs for high availability.

1

2

3

## ► Customize AZs

Number of public subnets [Info](#)

The number of public subnets to add to your VPC. Use public subnets for web applications that need to be publicly accessible over the internet.

0

2

Number of private subnets [Info](#)

The number of private subnets to add to your VPC. Use private subnets to secure backend resources that don't need public access.

0

2

4

## ► Customize subnets CIDR blocks

## ► Customize subnets CIDR blocks

NAT gateways (\$) [Info](#)

Choose the number of Availability Zones (AZs) in which to create NAT gateways. Note that there is a charge for each NAT gateway.

None

In 1 AZ

1 per AZ

VPC endpoints [Info](#)

Endpoints can help reduce NAT gateway charges and improve security by accessing S3 directly from the VPC. By default, full access policy is used. You can customize this policy at any time.

None

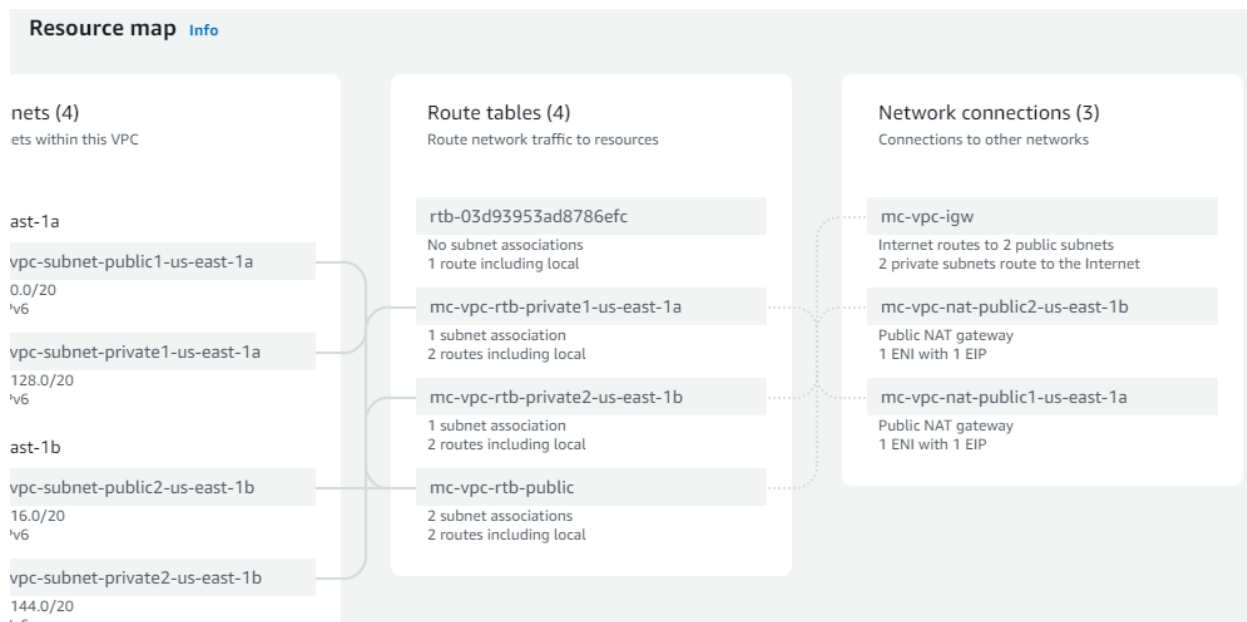
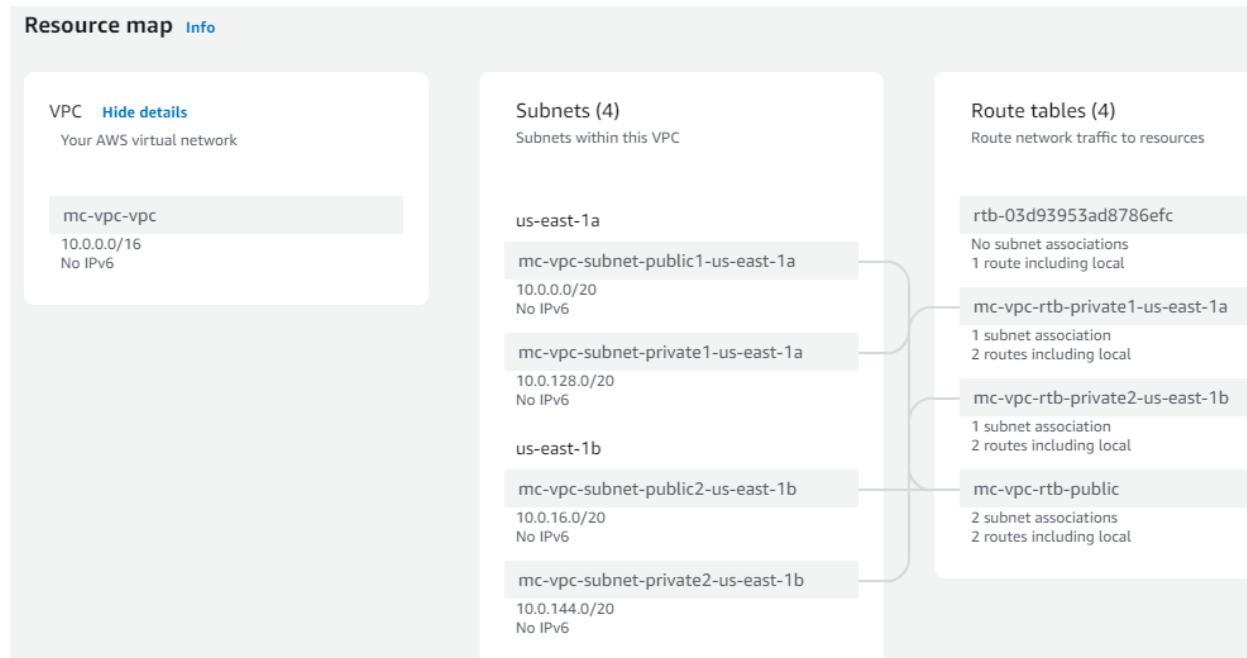
S3 Gateway

DNS options [Info](#)

- ☒ Enable DNS hostnames
- ☒ Enable DNS resolution

## ► Additional tags

Now, you have a VPC with 2 public subnets, 2 private subnets, 3 route tables (2 public and 1 private), 1 internet gateway, and 2 NAT gateways.



## Creating an Auto Scaling Group (ASG)

Creating an Auto Scaling Group (ASG) in the previously created VPC involves defining launch configurations, setting up the Auto Scaling Group, and configuring scaling policies.

Below are step-by-step instructions:

### Step 1: Create a Launch Template

In the AWS Management Console, go to "Services" > "EC2."

1. In the left navigation pane, under "Instances," click on "Launch Templates."
2. Click the "Create launch template" button.
3. Enter a name for your launch template.
4. **Version:** Enter a version number or accept the default.
5. **AMI (Amazon Machine Image):** Choose an AMI for your instances.
6. **Instance Type:** Choose the type of instance you want to launch.
7. **Key Pair:** Select a key pair for SSH access.
8. **Security Groups:** Choose one or more security groups for your instances.
9. **Network Settings:**
  - **Network:** Choose the VPC.
  - **Subnet:** Choose a subnet.
10. **Tags:**
  - Optionally, add tags to your instances.
11. Click "Create launch template."

### Step 2: Create an Auto Scaling Group

1. After creating the launch template, click "Create Auto Scaling group" from the confirmation page.
2. Configure the Auto Scaling group settings:
  - **Group name:** Enter a name for your Auto Scaling group.

- **Group size:** Set the desired and minimum number of instances.
  - **VPC:** Choose the VPC you created.
  - **Subnets:** Choose the private subnets.
  - **Load balancing:** If you have a load balancer, you can configure it here.
3. Configure the Advanced details:
- **Health check type:** Choose EC2 for this example.
  - **Health check grace period:** Set a suitable grace period.
4. Click "Next" until you reach the "Review" page.
5. Review your configuration and click "Create Auto Scaling group."

### Step 3: Test the Auto Scaling Group

1. Wait for the Auto Scaling group to launch instances based on the desired capacity.
2. Monitor the Auto Scaling group in the AWS Management Console to ensure that instances are launched and terminated based on your scaling policies.

### Important Notes:

- The instances launched by the Auto Scaling group will use the launch configuration settings.
- If you want instances to be part of a public subnet (for internet access), you may need to adjust the subnet configuration and associate an Elastic IP or use a public IP address.
- Make sure that your instances are appropriately configured to handle dynamic scaling.

This setup creates an Auto Scaling Group within your VPC, which can automatically adjust the number of instances based on defined scaling policies. Ensure that your launch configuration and scaling policies are well-tailored to your application's requirements. Adjustments can be made based on the specific needs of your environment and application.


EC2 > Auto Scaling groups > mcasg

## mcasg

Details | Activity | Automatic scaling | Instance management | Monitoring | Instance refresh




### Group details

Edit

Auto Scaling group name mcasg	Desired capacity 2	Desired capacity type Units (number of instances)	Amazon Resource Name (ARN)  arn:aws:autoscaling:us-east-1:937351429912:autoScalingGroup:3615f1ad-8eb2-4653-8270-2cdda5b41eab:autoScalingGroup/mcasg
Date created Wed Nov 22 2023 14:54:33 GMT+0530 (India Standard Time)	Minimum capacity 1	Status -	
	Maximum capacity 5		

### Launch template

Edit

Launch template  lt-019a596338d44686a mc-template	AMI ID  ami-0230bd60aa48260c6	Instance type t2.micro	Owner arn:aws:iam::937351429912:root
Version Default	Security groups -	Security group IDs  sg-01907c668b45edbe1	Create time Wed Nov 22 2023 14:51:48 GMT+0530 (India Standard Time)
Description mc-template	Storage (volumes) -	Key pair name awskeypair	Request Spot Instances No

[View details in the launch template console](#)

## Network [Info](#)

For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across the zones. The default VPC and default subnets are suitable for getting started quickly.

### VPC

Choose the VPC that defines the virtual network for your Auto Scaling group.

vpc-0085ea99478e939cd (mc-vpc-vpc)  
10.0.0.0/16



[Create a VPC](#)

### Availability Zones and subnets

Define which Availability Zones and subnets your Auto Scaling group can use in the chosen VPC.

Select Availability Zones and subnets



us-east-1a | subnet-00998c11026c3cb81 (mc-vpc-subnet-private1-us-east-1a)  
10.0.128.0/20



us-east-1b | subnet-0f4a6abf51566a939 (mc-vpc-subnet-private2-us-east-1b)  
10.0.144.0/20

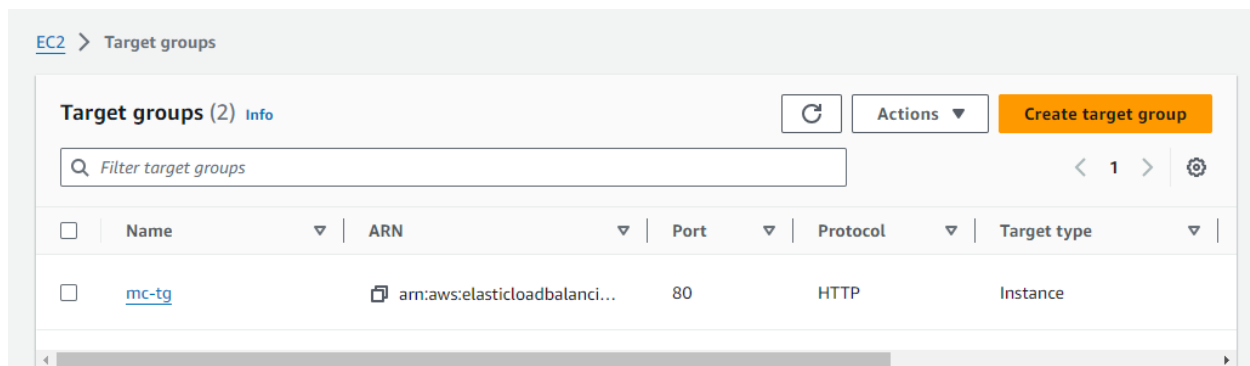


[Create a subnet](#)

## Creating Elastic Load Balancer

### Step 1: Create a Target Group

1. In the AWS Management Console, go to "Services" > "EC2."
2. Under "Load Balancing," click on "Target Groups" in the left navigation pane.
3. Click the "Create target group" button.
4. Enter a name for your target group.
5. **Protocol:** Choose the protocol for your target group (e.g., HTTP or HTTPS).
6. **Port:** Specify the port for your target group.
7. **VPC:** Choose the VPC you created.
8. **Health checks:**
  - Configure health check settings, including the protocol, path, and health check interval.
9. Click "Create."



### Step 2: Create an Application Load Balancer (ALB)

1. In the EC2 Dashboard, under "Load Balancing," click on "Load Balancers" in the left navigation pane.
2. Click the "Create Load Balancer" button.
3. Choose "Application Load Balancer."
4. Configure the load balancer settings:
  - **Name:** Enter a name for your load balancer.



- **Scheme:** Choose "internet-facing" for a public-facing load balancer.
  - **IP address type:** Choose "ipv4."
5. Configure security settings, listeners, and routing as needed.
  6. Under "Availability Zones," choose the VPC you created and select the subnets where your instances are running.
  7. Configure security groups for your load balancer.
  8. Click "Next: Configure Routing."
  9. For "Configure Routing," create a new target group and select the target group you created in Step 1.
  10. Click "Next: Register Targets" and "Next: Review."
  11. Review your settings and click "Create."

### Testing the Configuration:

12. Ensure that your target group has healthy instances registered.
13. Access the DNS name or public IP of your Elastic Load Balancer. You can find this information on the "Description" tab of the load balancer.
14. The load balancer should distribute traffic to the instances registered with the target group based on the configured rules.

EC2 > Load balancers > mcelb

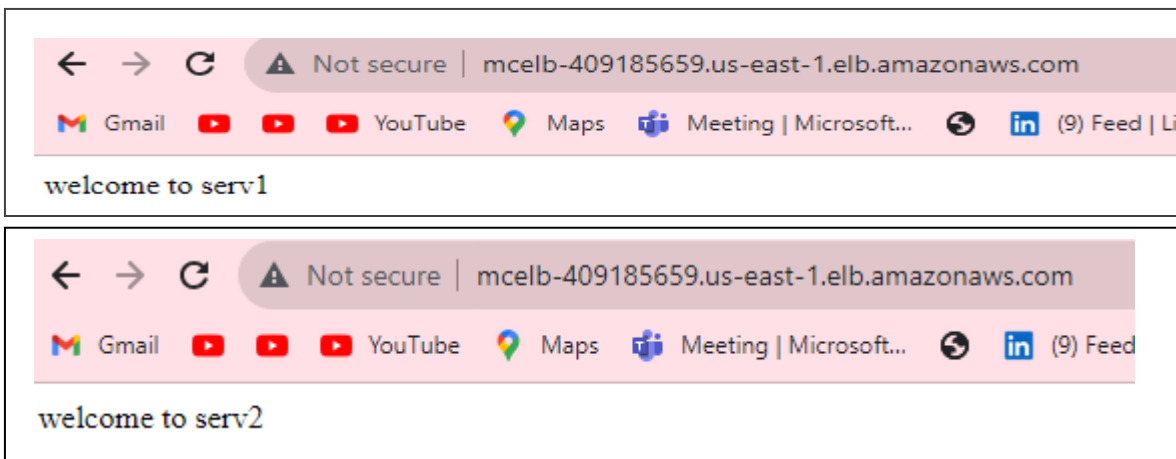
**mcelb** Refresh Actions

▼ Details

Load balancer type Application	Status Active	VPC <a href="#">vpc-0085ea99478e939cd</a>	IP address type IPv4
Scheme Internet-facing	Hosted zone Z35SXDOTRQ7X7K	Availability Zones <a href="#">subnet-068ce3cae4096e015</a> us-east-1a (use1-az4) <a href="#">subnet-0f617b422253a603f</a> us-east-1b (use1-az6)	Date created November 22, 2023, 15:33 (UTC+05:30)
Load balancer ARN <a href="#">arn:aws:elasticloadbalancing:us-east-1:937351429912:loadbalancer/app/mcelb/60a3b41105624a29</a>		DNS name <a href="#">Info</a> <a href="#">mcelb-409185659.us-east-1.elb.amazonaws.com</a> (A Record)	

[illegible]

- Now you access serv1 and install httpd package :
  - ✓ Sudo yum install -y httpd
  - ✓ Sudo systemctl start httpd
  - ✓ Sudo systemctl enable httpd
  - ✓ Sudo vi /var/www/html/index.html write your html code save and exit  
:wq
- For connecting serv2 now give command ssh -i awskeypair.pem ec2-user@serv1-ip-address
- Now you access serv2 and install httpd package :
  - ✓ Sudo yum install -y httpd
  - ✓ Sudo systemctl start httpd
  - ✓ Sudo systemctl enable httpd
  - ✓ Sudo vi /var/www/html/index.html write your html code save and exit  
:wq
- Now copy the DNS of elb and access in the web
- Required output is displayed



### !! Caution !!

- **Delete all services which are created by you**
- **First delete auto scaling group, elastic load balancer, Target group, Terminate instances, delete elastic ips, vpc**