

PROJECT 1

(DEPLOYING A HIGH AVAILABLE WEB APPLICATION AND BASTION HOST IN AWS)

STEP 1 : Create VPC name (MyVPC) IPV4 CIDR = 10.0.0.0/16

The screenshot shows the AWS VPC Dashboard. On the left sidebar, under the 'VIRTUAL PRIVATE CLOUD' section, 'Your VPCs' is selected. In the main content area, the 'Your VPCs' table shows two entries: 'MyVPC' (selected) with VPC ID 'vpc-03c8685e07a43e526' and State 'Available', and another entry with VPC ID 'vpc-c57db1b8' and State 'Available'. Below the table, the 'Details' tab is selected for 'vpc-03c8685e07a43e526 / MyVPC', displaying various configuration details like VPC ID, State, DHCP options set, and route tables.

STEP 2 : Create Subnet (select AZ 1)

1) Public subnet IPV4 CIDR 10.0.1.0/24

The screenshot shows the AWS Subnets page. Under the 'Subnets' section in the sidebar, the 'Subnets' table lists two subnets: 'PrivateSubnet' (unchecked) with Subnet ID 'subnet-0c423bb15805696f8' and State 'Available', and 'PublicSubnet' (checked) with Subnet ID 'subnet-0598aba40a27a7eb8' and State 'Available'. Below the table, the 'Details' tab is selected for 'subnet-0598aba40a27a7eb8 / PublicSubnet', showing details like Subnet ID, State, IPv4 CIDR, Availability Zone, and Network ACL.

2) Private subnet IPV4 CIDR 10.0.2.0/24 (select AZ 2)

The screenshot shows the AWS VPC Subnets page. On the left sidebar, under the 'VIRTUAL PRIVATE CLOUD' section, 'Subnets' is selected. The main content area displays a table of subnets:

Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 CIDR	Available IPv4 addresses
PrivateSubnet	subnet-0c423bb15805696f8	Available	vpc-03c8685e07a43e526 MyVPC	10.0.2.0/24	-	248
PublicSubnet	subnet-0598aba40a27a7eb8	Available	vpc-03c8685e07a43e526 MyVPC	10.0.1.0/24	-	248

Below the table, there are tabs for 'Details', 'Flow logs', 'Route table', 'Network ACL', 'Tags', and 'Sharing'. The 'Details' tab is selected, showing the following information for the PrivateSubnet:

- Subnet ID: subnet-0c423bb15805696f8
- State: Available
- VPC: vpc-03c8685e07a43e526 | MyVPC
- IPv4 CIDR: 10.0.2.0/24
- Available IPv4 addresses: 248
- IPv6 CIDR: -
- Availability Zone: us-east-1b
- Availability Zone ID: use1-az2
- Network border group: us-east-1
- Route table: rtb-0091868941bb14a40 | PrivateRouteTable
- Auto-assign public IPv4 address: No
- Auto-assign IPv6 address: No
- Auto-assign customer-owned IPv4 address: No
- Default subnet: No
- Customer-owned IPv4 pool: -
- Outpost ID: -
- Owner: amawsse2us-east-1:774055385223:subnet-0c423bb15805696f8

STEP 3 : Create Internet Gateway (for internet access)

The screenshot shows the AWS Internet Gateways page. On the left sidebar, under the 'INTERNET GATEWAYS' section, 'Internet Gateways' is selected. The main content area displays a table of internet gateways:

Name	Internet gateway ID	State	VPC ID	Owner
MyIGW	igw-073abdaa3acce8069	Attached	vpc-03c8685e07a43e526 MyVPC	774055385223
-	igw-d8b487a3	Attached	vpc-c57db1b8	774055385223

Below the table, there are tabs for 'Details' and 'Tags'. The 'Details' tab is selected, showing the following information for the MyIGW internet gateway:

- Internet gateway ID: igw-073abdaa3acce8069
- State: Attached
- VPC ID: vpc-03c8685e07a43e526 | MyVPC
- Owner: 774055385223

STEP 4 : Create route table

1) Public route table (add route to internet gateway & modify auto assign IP & associate with public subnet)

The screenshot shows the AWS VPC Route Tables page. A new route table named "PublicRouteT..." is being created. The table is explicitly associated with the public subnet "subnet-0598aba40a27a7eb8". The route table ID is "rtb-055dc2edb6b663b21". The table is owned by the user and is associated with the VPC "vpc-03c8685e07a43e526 | MyVPC".

2) Private route table (associate with private subnet)

The screenshot shows the AWS VPC Route Tables page. A new route table named "PrivateRouteT..." is being created. The table is explicitly associated with the private subnet "subnet-0423bb15805696f8". The route table ID is "rtb-0091868941bb14a40". The table is owned by the user and is associated with the VPC "vpc-03c8685e07a43e526 | MyVPC".

STEP 5 : launch instance for bastion server in public subnet (SG custom)

The screenshot shows the AWS EC2 Instances page. An instance named "Bastion" is listed, with the instance ID "i-0cc367edab5c78080". The instance is running in the "us-east-1a" availability zone and has a public IPv4 address of "3.238.105.125". The instance type is "t2.micro". The instance is associated with the VPC "vpc-03c8685e07a43e526 | MyVPC" and the subnet "subnet-0598aba40a27a7eb8". The instance has an elastic IP assigned.

STEP 6 : Create Security Group for Load Balancer (SG http)

The screenshot shows the AWS Management Console with the search bar at the top containing 'Search for services, features, marketplace products, and docs [Alt+S]'. The left sidebar includes sections for Events, Tags, Limits, Instances (with sub-options like Instances, Launch Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations), Images (AMIs), Elastic Block Store (Volumes, Snapshots, Lifecycle Manager), Network & Security (Security Groups, Elastic IPs), and Load Balancing (Load Balancers, Target Groups). The main content area displays the 'Security Groups (1/5) Info' table. A new security group, 'sg-014c034a31db24a62 - LoadBalancerSG', is being created. The table has columns for Name, Security group ID, Security group name, VPC ID, Description, Owner, Inbound rules count, and Outbound rules count. The 'LoadBalancerSG' row is highlighted in blue. Below the table, the details for 'sg-014c034a31db24a62 - LoadBalancerSG' are shown, including its name, security group ID, description ('Security group for load balancer'), owner ('774055385223'), and rule counts.

STEP 7 : Launch 2 instance for webserver in private subnet (in SG 1st source ssh for bastion SG & 2nd source http for loadbalancer SG)

Webserver 1

The screenshot shows the AWS Management Console with the search bar at the top containing 'Search for services, features, marketplace products, and docs [Alt+S]'. The left sidebar includes sections for Instances (with sub-options like Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations), Images (AMIs), Elastic Block Store (Volumes, Snapshots, Lifecycle Manager), Network & Security (Security Groups, Elastic IPs), Placement Groups, Key Pairs, Network Interfaces, and Auto Scaling (Launch Configurations). The main content area displays the 'Instances (1/3) Info' table. Three instances are listed: 'Bastion' (i-0cc367edab5c78080, Running, t2.micro, 2/2 checks, No alarms, us-east-1a, 3.238.105.125, -), 'w2' (i-06486ee198c6c1cd, Running, t2.micro, 2/2 checks, No alarms, us-east-1b, -, -), and 'w1' (i-09dd98f9802b28b10, Running, t2.micro, 2/2 checks, No alarms, us-east-1b, -, -). Below the table, the details for instance 'w1' are shown, including its instance ID (i-09dd98f9802b28b10), state (Running), type (t2.micro), and various network and monitoring configurations. It is associated with a VPC ID (vpc-03c8685e07a43e526) and a subnet (subnet-0c423bb1580569f8).

Webserver 2

STEP 8 : Create Load Balancer (in myvpc & register webserver with loadbalancer)

The screenshot shows the AWS CloudWatch Metrics interface. At the top, there's a search bar with placeholder text "Search for services, features, marketplace products, and docs" and a keyboard shortcut "[Alt+S]". The top navigation bar includes links for "Ganesh Chandrakant Dandavate", "N. Virginia", and "Support". On the left, a sidebar lists various AWS services with their respective icons and names: Instances, Images, Elastic Block Store, Network & Security, Load Balancing, Auto Scaling, and more. The "Load Balancing" section is currently selected, showing a table of existing load balancers. One entry, "WebAppLB", is highlighted. Below the table, detailed configuration settings for "WebAppLB" are displayed under sections like "Basic Configuration" and "Security".

Load Balancer: WebAppLB

Basic Configuration

Name	WebAppLB
ARN	arn:aws:elasticloadbalancing:us-east-1:774055385223:loadbalancer/app/WebAppLB/1d4ba4263764c221
DNS name	WebAppLB-424069305.us-east-1.elb.amazonaws.com (A Record)
State	active
Type	application
Scheme	internet-facing
IP address type	ipv4
Edit IP address type	
VPC	vpc-03c8685e07a43e526
Availability Zones	subnet-059abab40a27a7eb8 - us-east-1a IPv4 address: Assigned by AWS
	subnet-0c423bb15805696f8 - us-east-1b IPv4 address: Assigned by AWS
Edit subnets	
Hosted zone	Z35SXDOTRQ7X7K
Creation time	January 1, 2021 at 4:07:15 PM UTC+5:30

Security

Security groups	sg-014c034a31db24a62, LoadBalancerSG • Security group for load balancer
Edit security groups	

STEP 9 : Create NAT gateway in public subnet (route with main route table & private subnet)

The screenshot shows the AWS VPC Management Console. On the left, there's a sidebar with various VPC-related options like 'New VPC Experience', 'VPC Dashboard', and 'Virtual Private Cloud'. The main area is titled 'NAT gateways (1/1) Info' and shows a single entry for 'MyNAT'. The table includes columns for Name, NAT gateway ID, State, State message, Elastic IP address, Private IP address, Network interface ID, VPC, and Subnet. The 'Details' tab is selected, showing the NAT gateway ID (nat-0ca13c8dbf1f0709e), State (Available), State message (-), Elastic IP address (34.194.191.19), Private IP address (10.0.1.12), Network interface ID (eni-0a0120d6a18b93490), VPC (vpc-03c8685e07a43e526 / MyVPC), and Subnet (subnet-0598aba40a27a7eb8 / PublicSubnet). There are also tabs for 'Monitoring' and 'Tags'.

STEP 10 :

1) Connect bastion host and run below command

```
* sudo su
* yum update -y
* copy downloaded pem file content
* vi key.pem ( and paste )
* ssh -i key.pem ec2-user@privateIPwebserver1
* after connected server1
* sudo su
* yum update -y
* yum install httpd -y
* systemctl start httpd
* systemctl enable httpd
* vi /var/www/html/index.html ( write here html code )
* above same process do for server 2
```

STEP 11 : Navigate to Load balancer & copy DNS address & check

The screenshot shows a browser window with several tabs open. The tabs include 'NAT gateways | VPC Management', 'i-0cc367edab5c78080 (Bastion)', 'webapplb-424069305.us-east-1.elb.amazonaws.com', and 'Folder - Google Drive'. The main content area shows a message 'REQUEST HANDLING BY SERVER 1'.

The screenshot shows a browser window with several tabs open. The tabs include 'NAT gateways | VPC Management', 'i-0cc367edab5c78080 (Bastion)', 'webapplb-424069305.us-east-1.elb.amazonaws.com', and 'Folder - Google Drive'. The main content area shows a message 'REQUEST HANDLING BY SERVER 2'.