

Assessment Project - 1

Project Title:

Deploying a Highly Available Web Application and Bastion Host in AWS

Following are the sequence of steps & screenshots for the solution,

I. Created VPC - **samp_VPC** (IPV4 CIDR - **10.0.0.0/16**)

The screenshot shows the AWS Management Console interface for VPCs. The top navigation bar includes the AWS logo, 'Services' dropdown, and user information (Shiva Basava P, N. Virginia, Support). The main content area is titled 'Your VPCs (1/2) Info'. Below the title is a search bar and a table of VPCs. The table has columns: Name, VPC ID, State, IPv4 CIDR, and IPv6 CIDR (Network border group). One VPC, 'samp_VPC', is listed with VPC ID 'vpc-0c9081107866453a2', State 'Available', and IPv4 CIDR '10.0.0.0/16'. Below the table, the details for 'vpc-0c9081107866453a2' are shown in a grid format.

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR (Network border group)
samp_VPC	vpc-0c9081107866453a2	Available	10.0.0.0/16	-

Property	Value
VPC ID	vpc-0c9081107866453a2
State	Available
DNS hostnames	Disabled
DNS resolution	Enabled
Tenancy	Default
DHCP options set	dopt-ed9df097
Route table	rtb-0add175e93f307df5
Network ACL	acl-0d844677efb6cc2a8
Default VPC	No
IPv4 CIDR	10.0.0.0/16
IPv6 pool	-
Owner ID	973501320577

II. Created **Internet Gateway** - **my-IGW** & Attached it to the above VPC.

The screenshot shows the AWS Management Console interface for Internet Gateways. The top navigation bar includes the AWS logo, 'Services' dropdown, and user information (Shiva Basava P, N. Virginia, Support). The main content area is titled 'Internet gateways (1/2) Info'. Below the title is a search bar and a table of Internet Gateways. The table has columns: Name, Internet gateway ID, State, and VPC ID. One Internet Gateway, 'my-IGW', is listed with Internet gateway ID 'igw-0ded758fbd2791bcc', State 'Attached', and VPC ID 'vpc-0c9081107866453a2 | samp_VPC'. Below the table, the details for 'igw-0ded758fbd2791bcc / my-IGW' are shown in a grid format.

Name	Internet gateway ID	State	VPC ID
my-IGW	igw-0ded758fbd2791bcc	Attached	vpc-0c9081107866453a2 samp_VPC

Property	Value
Internet gateway ID	igw-0ded758fbd2791bcc
State	Attached
VPC ID	vpc-0c9081107866453a2 samp_VPC
Owner	973501320577

III. Created **Security Groups** in the above VPC for - Bastion Server, Web-Server & Load Balancer.

1. **Security Group** for Bastion Server - **SG_Batsion_server**.

Inbound Rules for **SG_Batsion_server**

The screenshot shows the AWS Management Console interface for the 'Security Groups' page. The 'Inbound rules' tab is selected for the security group 'SG_Batsion_server' (ID: sg-0cecc1a8f28a8bb77). The table below lists the inbound rules.

Type	Protocol	Port range	Source	Description - optional
SSH	TCP	22	0.0.0.0/0	-

Outbound Rules(Default) for **SG_Batsion_server**

The screenshot shows the AWS Management Console interface for the 'Security Groups' page, specifically the 'Outbound rules' tab for the security group 'SG_Batsion_server' (ID: sg-0cecc1a8f28a8bb77). The table below lists the outbound rules.

Type	Protocol	Port range	Destination	Description - optional
All traffic	All	All	0.0.0.0/0	-

2. Security Group for Load Balancer - SG_LB

Inbound Rules for SG_LB

The screenshot displays the AWS Management Console interface for the 'Security Groups (1/5)' page. The 'Inbound rules' tab is selected for the security group 'sg-008269d9039de4c40 - SG_LB'. The 'Inbound rules' table shows a single rule for HTTP traffic on port 80 from any source (0.0.0.0/0).

Type	Protocol	Port range	Source	Description - optional
HTTP	TCP	80	0.0.0.0/0	-

Outbound Rules(Default) for SG_LB

The screenshot displays the AWS Management Console interface for the 'Security Groups (1/5)' page. The 'Outbound rules' tab is selected for the security group 'sg-008269d9039de4c40 - SG_LB'. The 'Outbound rules' table shows a single default rule for all traffic to any destination (0.0.0.0/0).

Type	Protocol	Port range	Destination	Description - optional
All traffic	All	All	0.0.0.0/0	-

3. Security Group for Web-Server - SG_web-server

(A common Security Group for both Web-Server1 & Web-Server2)

Inbound Rules for **SG_web-server**, copy from **Security Group** of **Bastion Server** & **Load Balancer**

The screenshot shows the AWS Management Console interface for the Security Groups page. The 'Security Groups (1/5)' section is active, displaying a table with one entry: 'SG_web-server' with ID 'sg-0123b992e534e4495' and VPC ID 'vpc-0c9081107866453a2'. Below this, the 'Inbound rules' tab is selected for the selected security group. The 'Inbound rules' table shows two rules: HTTP (TCP, Port 80, Source: sg-008269d9039de4c40 (SG_LB)) and SSH (TCP, Port 22, Source: sg-0cecc1a8f28a8bb77 (SG_Bastion_server)).

Type	Protocol	Port range	Source	Description - optional
HTTP	TCP	80	sg-008269d9039de4c40 (SG_LB)	-
SSH	TCP	22	sg-0cecc1a8f28a8bb77 (SG_Bastion_server)	-

Outbound Rules(Default) for SG_web-server

The screenshot shows the AWS Management Console interface for the Security Groups page, specifically the 'Outbound rules' tab for the 'SG_web-server' security group. The 'Outbound rules' table shows a single default rule: All traffic (All, Port range: All, Destination: 0.0.0.0/0).

Type	Protocol	Port range	Destination	Description - optional
All traffic	All	All	0.0.0.0/0	-

IV. Created **Subnets**(Public & Private) inside the above **VPC**.

1. **Private Subnet - private_sub** (IPv4 CIDR - **10.0.1.0/24**) & at Availability Zone - **us-east-1c**.

The screenshot shows the AWS Management Console interface for a subnet. The top navigation bar includes the user name 'Shiva Basava P', the region 'N. Virginia', and a 'Support' link. Below the navigation bar, there are buttons for 'Create subnet' and 'Actions'. A search bar is present with the text 'Filter by tags and attributes or search by keyword'. A table lists subnets, with 'private_sub' (subnet-05cd092438f3ee82e) selected. The details for this subnet are shown below the table, including its ID, VPC, state, IPv4 CIDR, and other attributes.

Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR
private_sub	subnet-05cd092438f3ee82e	available	vpc-0c9081107866453a2 samp_VPC	10.0.1.0/24	248	-

Subnet: subnet-05cd092438f3ee82e

Description	Flow Logs	Route Table	Network ACL	Tags	Sharing
Subnet ID	subnet-05cd092438f3ee82e				
VPC	vpc-0c9081107866453a2 samp_VPC				
Available IPv4 Addresses	248				
Availability Zone	us-east-1c (use1-az2)				
Route Table	rtb-0add175e93f307df5				
Default subnet	No				
Auto-assign customer-owned IPv4 address	No				
Auto-assign IPv6 address	No				
Owner	973501320577				
State	available				
IPv4 CIDR	10.0.1.0/24				
IPv6 CIDR	-				
Network Border Group	us-east-1				
Network ACL	acl-0d844677efb6cc2a8				
Auto-assign public IPv4 address	No				
Customer-owned IPv4 pool	-				
Outpost ID	-				

(after Route Table Association **Step VI**, through **NATGateway - NAT-GW** from **Step V**)

The screenshot shows the AWS Management Console interface for a subnet, similar to the previous one. The 'Route Table' tab is selected, and the 'Edit route table association' button is visible. The route table 'rtb-0add175e93f307df5' is associated with the subnet. The route table details are shown below the table.

Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR
private_sub	subnet-05cd092438f3ee82e	available	vpc-0c9081107866453a2 samp_VPC	10.0.1.0/24	248	-
subnet-3b8c3e1a	subnet-3b8c3e1a	available	vpc-834282fe	172.31.0.0/20	4091	-

Subnet: subnet-05cd092438f3ee82e

Description	Flow Logs	Route Table	Network ACL	Tags	Sharing
Route Table	rtb-0add175e93f307df5				

Edit route table association

Route Table: rtb-0add175e93f307df5

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	nat-08d809931d18a064d

2. **Public Subnet**, Created 3 public subnets with preference to Availability Zones.

- Public Subnet 1** - **public_subnet_1** (IPV4 CIDR - **10.0.0.0/24**) at Availability Zone - **us-east-1a**.

The screenshot shows the AWS Management Console interface for a subnet. At the top, there's a header with the user's name 'Shiva Basava P', region 'N. Virginia', and 'Support' link. Below the header, there's a 'Create subnet' button and an 'Actions' dropdown. A search bar is present with the text 'Filter by tags and attributes or search by keyword'. Below the search bar, there's a table listing subnets. The first subnet is 'public_subnet_1' with ID 'subnet-029d88c739c5a3e08', state 'available', VPC 'vpc-0c9081107866453a2 | samp_VPC', IPv4 CIDR '10.0.0.0/24', and 248 available IPv4 addresses. The second subnet is 'private_sub' with ID 'subnet-05cd092438f3ee82e', state 'available', VPC 'vpc-0c9081107866453a2 | samp_VPC', IPv4 CIDR '10.0.1.0/24', and 248 available IPv4 addresses. Below the table, there's a section for 'Subnet: subnet-029d88c739c5a3e08' with tabs for 'Description', 'Flow Logs', 'Route Table', 'Network ACL', 'Tags', and 'Sharing'. The 'Description' tab is selected, showing details for the subnet: Subnet ID, VPC, Available IPv4 Addresses, Availability Zone, Route Table, Default subnet, Auto-assign customer-owned IPv4 address, Auto-assign IPv6 address, Owner, State, IPv4 CIDR, IPv6 CIDR, Network Border Group, Network ACL, Auto-assign public IPv4 address, Customer-owned IPv4 pool, and Outpost ID.

Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR
public_subnet_1	subnet-029d88c739c5a3e08	available	vpc-0c9081107866453a2 samp_VPC	10.0.0.0/24	248	-
private_sub	subnet-05cd092438f3ee82e	available	vpc-0c9081107866453a2 samp_VPC	10.0.1.0/24	248	-

Subnet: subnet-029d88c739c5a3e08

Property	Value
Subnet ID	subnet-029d88c739c5a3e08
VPC	vpc-0c9081107866453a2 samp_VPC
Available IPv4 Addresses	248
Availability Zone	us-east-1a (use1-az6)
Route Table	rtb-083057542d5670d27 public_RT
Default subnet	No
Auto-assign customer-owned IPv4 address	No
Auto-assign IPv6 address	No
Owner	973501320577
State	available
IPv4 CIDR	10.0.0.0/24
IPv6 CIDR	-
Network Border Group	us-east-1
Network ACL	acl-0d844677efb6cc2a8
Auto-assign public IPv4 address	Yes
Customer-owned IPv4 pool	-
Outpost ID	-

(after Route Table Association next step, **Step VI**, through **Internet Gateway** - **my-IGW**)

The screenshot shows the AWS Management Console interface for a subnet. At the top, there's a header with the user's name 'Shiva Basava P', region 'N. Virginia', and 'Support' link. Below the header, there's a 'Create subnet' button and an 'Actions' dropdown. A search bar is present with the text 'Filter by tags and attributes or search by keyword'. Below the search bar, there's a table listing subnets. The first subnet is 'public_subnet_1' with ID 'subnet-029d88c739c5a3e08', state 'available', VPC 'vpc-0c9081107866453a2 | samp_VPC', IPv4 CIDR '10.0.0.0/24', and 248 available IPv4 addresses. The second subnet is 'private_sub' with ID 'subnet-05cd092438f3ee82e', state 'available', VPC 'vpc-0c9081107866453a2 | samp_VPC', IPv4 CIDR '10.0.1.0/24', and 248 available IPv4 addresses. Below the table, there's a section for 'Subnet: subnet-029d88c739c5a3e08' with tabs for 'Description', 'Flow Logs', 'Route Table', 'Network ACL', 'Tags', and 'Sharing'. The 'Route Table' tab is selected, showing the 'Edit route table association' button. Below the button, there's a section for 'Route Table: rtb-083057542d5670d27 | public_RT' with a table showing the route table entries. The first entry is for destination '10.0.0.0/16' with target 'local'. The second entry is for destination '0.0.0.0/0' with target 'igw-0ded758fbd2791bcc'.

Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR
public_subnet_1	subnet-029d88c739c5a3e08	available	vpc-0c9081107866453a2 samp_VPC	10.0.0.0/24	248	-
private_sub	subnet-05cd092438f3ee82e	available	vpc-0c9081107866453a2 samp_VPC	10.0.1.0/24	248	-

Subnet: subnet-029d88c739c5a3e08

Property	Value
Subnet ID	subnet-029d88c739c5a3e08
VPC	vpc-0c9081107866453a2 samp_VPC
Available IPv4 Addresses	248
Availability Zone	us-east-1a (use1-az6)
Route Table	rtb-083057542d5670d27 public_RT
Default subnet	No
Auto-assign customer-owned IPv4 address	No
Auto-assign IPv6 address	No
Owner	973501320577
State	available
IPv4 CIDR	10.0.0.0/24
IPv6 CIDR	-
Network Border Group	us-east-1
Network ACL	acl-0d844677efb6cc2a8
Auto-assign public IPv4 address	Yes
Customer-owned IPv4 pool	-
Outpost ID	-

Edit route table association

Route Table: rtb-083057542d5670d27 | public_RT

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	igw-0ded758fbd2791bcc

- b. Public Subnet 2 - **public_subnet_2** (IPv4 CIDR - **10.0.20.0/24**) & at Availability Zone - **us-east-1b**.

The screenshot shows the AWS Management Console interface for a subnet. The top navigation bar includes the user name 'Shiva Basava P', the region 'N. Virginia', and a 'Support' link. Below the navigation bar, there are buttons for 'Create subnet' and 'Actions'. A search bar is present with the text 'Filter by tags and attributes or search by keyword'. A table lists subnets, with 'public_subnet_2' selected. The details for 'Subnet: subnet-04dde2c35d7e31575' are shown below the table. The 'Description' tab is active, displaying various attributes of the subnet.

Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR
public_subnet_2	subnet-04dde2c35d7e31575	available	vpc-0c9081107866453a2 samp_VPC	10.0.20.0/24	250	-
public_subnet_1	subnet-029d88c739c5a3e08	available	vpc-0c9081107866453a2 samp_VPC	10.0.0.0/24	248	-

Subnet: subnet-04dde2c35d7e31575

Description		Flow Logs		Route Table		Network ACL		Tags		Sharing	
Subnet ID	subnet-04dde2c35d7e31575	State	available	Subnet ID	subnet-04dde2c35d7e31575	State	available	Subnet ID	subnet-04dde2c35d7e31575	State	available
VPC	vpc-0c9081107866453a2 samp_VPC	IPv4 CIDR	10.0.20.0/24	VPC	vpc-0c9081107866453a2 samp_VPC	IPv4 CIDR	10.0.20.0/24	VPC	vpc-0c9081107866453a2 samp_VPC	IPv4 CIDR	10.0.20.0/24
Available IPv4 Addresses	250	IPv6 CIDR	-	Available IPv4 Addresses	250	IPv6 CIDR	-	Available IPv4 Addresses	250	IPv6 CIDR	-
Availability Zone	us-east-1b (use1-az1)	Network Border Group	us-east-1	Availability Zone	us-east-1b (use1-az1)	Network Border Group	us-east-1	Availability Zone	us-east-1b (use1-az1)	Network Border Group	us-east-1
Route Table	rtb-083057542d5670d27 public_RT	Network ACL	acl-0d844677efb6cc2a8	Route Table	rtb-083057542d5670d27 public_RT	Network ACL	acl-0d844677efb6cc2a8	Route Table	rtb-083057542d5670d27 public_RT	Network ACL	acl-0d844677efb6cc2a8
Default subnet	No	Auto-assign public IPv4 address	Yes	Default subnet	No	Auto-assign public IPv4 address	Yes	Default subnet	No	Auto-assign public IPv4 address	Yes
Auto-assign customer-owned IPv4 address	No	Customer-owned IPv4 pool	-	Auto-assign customer-owned IPv4 address	No	Customer-owned IPv4 pool	-	Auto-assign customer-owned IPv4 address	No	Customer-owned IPv4 pool	-
Auto-assign IPv6 address	No	Outpost ID	-	Auto-assign IPv6 address	No	Outpost ID	-	Auto-assign IPv6 address	No	Outpost ID	-
Owner	973501320577			Owner	973501320577			Owner	973501320577		

(after Route Table Association next step, **Step VI**, through **Internet Gateway - my-IGW**)

The screenshot shows the AWS Management Console interface for a subnet, specifically the 'Route Table' tab. The top navigation bar includes the user name 'Shiva Basava P', the region 'N. Virginia', and a 'Support' link. Below the navigation bar, there are buttons for 'Create subnet' and 'Actions'. A search bar is present with the text 'Filter by tags and attributes or search by keyword'. A table lists subnets, with 'public_subnet_2' selected. The details for 'Subnet: subnet-04dde2c35d7e31575' are shown below the table. The 'Route Table' tab is active, displaying the route table association for the subnet.

Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR
public_subnet_2	subnet-04dde2c35d7e31575	available	vpc-0c9081107866453a2 samp_VPC	10.0.20.0/24	250	-
public_subnet_1	subnet-029d88c739c5a3e08	available	vpc-0c9081107866453a2 samp_VPC	10.0.0.0/24	248	-

Subnet: subnet-04dde2c35d7e31575

Description		Flow Logs		Route Table		Network ACL		Tags		Sharing	
Subnet ID	subnet-04dde2c35d7e31575	State	available	Subnet ID	subnet-04dde2c35d7e31575	State	available	Subnet ID	subnet-04dde2c35d7e31575	State	available
VPC	vpc-0c9081107866453a2 samp_VPC	IPv4 CIDR	10.0.20.0/24	VPC	vpc-0c9081107866453a2 samp_VPC	IPv4 CIDR	10.0.20.0/24	VPC	vpc-0c9081107866453a2 samp_VPC	IPv4 CIDR	10.0.20.0/24
Available IPv4 Addresses	250	IPv6 CIDR	-	Available IPv4 Addresses	250	IPv6 CIDR	-	Available IPv4 Addresses	250	IPv6 CIDR	-
Availability Zone	us-east-1b (use1-az1)	Network Border Group	us-east-1	Availability Zone	us-east-1b (use1-az1)	Network Border Group	us-east-1	Availability Zone	us-east-1b (use1-az1)	Network Border Group	us-east-1
Route Table	rtb-083057542d5670d27 public_RT	Network ACL	acl-0d844677efb6cc2a8	Route Table	rtb-083057542d5670d27 public_RT	Network ACL	acl-0d844677efb6cc2a8	Route Table	rtb-083057542d5670d27 public_RT	Network ACL	acl-0d844677efb6cc2a8
Default subnet	No	Auto-assign public IPv4 address	Yes	Default subnet	No	Auto-assign public IPv4 address	Yes	Default subnet	No	Auto-assign public IPv4 address	Yes
Auto-assign customer-owned IPv4 address	No	Customer-owned IPv4 pool	-	Auto-assign customer-owned IPv4 address	No	Customer-owned IPv4 pool	-	Auto-assign customer-owned IPv4 address	No	Customer-owned IPv4 pool	-
Auto-assign IPv6 address	No	Outpost ID	-	Auto-assign IPv6 address	No	Outpost ID	-	Auto-assign IPv6 address	No	Outpost ID	-
Owner	973501320577			Owner	973501320577			Owner	973501320577		

Route Table: rtb-083057542d5670d27 | public_RT

Destination		Target	
10.0.0.0/16	local	10.0.0.0/16	local
0.0.0.0/0	igw-0ded758fbd2791bcc	0.0.0.0/0	igw-0ded758fbd2791bcc

- c. Public Subnet 2 - **public_subnet_2** (IPv4 CIDR - **10.0.40.0/24**) & at Availability Zone - **us-east-1d**.

Subnet: subnet-007475cfd979e46b4

Property	Value	Property	Value
Subnet ID	subnet-007475cfd979e46b4	State	available
VPC	vpc-0c9081107866453a2 samp_VPC	IPv4 CIDR	10.0.40.0/24
Available IPv4 Addresses	251	IPv6 CIDR	-
Availability Zone	us-east-1d (use1-az4)	Network Border Group	us-east-1
Route Table	rtb-083057542d5670d27 public_RT	Network ACL	acl-0d844677efb6cc2a8
Default subnet	No	Auto-assign public IPv4 address	Yes
Auto-assign customer-owned IPv4 address	No	Customer-owned IPv4 pool	-
Auto-assign IPv6 address	No	Outpost ID	-
Owner	973501320577		

(after Route Table Association next step, **Step VI**, through **Internet Gateway - my-IGW**)

Route Table: rtb-083057542d5670d27 | public_RT

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	igw-0ded758fbd2791bcc

V. Created a NAT Gateway - NAT-GW

The screenshot displays the AWS Management Console interface for NAT gateways. At the top, the user is logged in as Shiva Basava P in the N. Virginia region. The main heading is "NAT gateways (1/1)". Below this is a search bar and a table with one entry:

Name	NAT gateway ID	State	State message	Elastic IP address
NAT-GW	nat-08d809931d18a064d	Available	-	50.19.115.190

Below the table is a "Details" section with the following information:

- NAT gateway ID:** nat-08d809931d18a064d
- State:** Available
- State message:** -
- Elastic IP address:** 50.19.115.190
- Private IP address:** 10.0.0.219
- Network interface ID:** eni-01bf7a2898cfd7ba
- VPC:** vpc-0c9081107866453a2 / samp_VPC
- Subnet:** subnet-029d88c739c5a3e08 / public_subnet_1
- Created:** 2020/11/08 20:06 GMT+5:30
- Deleted:** -

Associated Elastic IP addresses

The screenshot displays the AWS Management Console interface for Elastic IP addresses. The user is logged in as Shiva Basava P in the N. Virginia region. The main heading is "Elastic IP addresses (1/1)". Below this is a search bar and a table with one entry:

Name	Allocated IPv4 address	Type	Allocation ID
-	50.19.115.190	Public IP	eipalloc-02422c117ebfa5393

Below the table is a "Summary" section with the following information:

- Allocated IPv4 address:** 50.19.115.190
- Type:** Public IP
- Allocation ID:** eipalloc-02422c117ebfa5393
- Association ID:** eipassoc-0261420ec3f22c441
- Scope:** VPC
- Associated instance ID:** -
- Private IP address:** 10.0.0.219
- Network interface ID:** eni-01bf7a2898cfd7ba
- Network interface owner account ID:** 973501320577
- Public DNS:** -
- NAT Gateway ID:** nat-08d809931d18a064d (NAT-GW)
- Address pool:** Amazon
- Network Border Group:** us-east-1

VI. Created Route table

- Public Subnets(public_subnet_1, public_subnet_2, public_subnet_3) - public_RT & Main route table = No.

Route Table: rtb-083057542d5670d27

Destination	Target	Status	Propagated
10.0.0.0/16	local	active	No
0.0.0.0/0	igw-0ded758fdb2791bcc	active	No

All associated & unassociated Subnets

The following subnets have not been explicitly associated with any route tables and are therefore associated with the main route table:

Subnet ID	IPv4 CIDR	IPv6 CIDR
subnet-05cd092438f3e...	10.0.1.0/24	-

- b. This is for communicating with Private Subnet, through **NAT Gateway(NAT-GT) & Main route table = Yes.**

(**Note:** We don't explicitly create this, Its Implicitly Created by route table during creation of **public_RT**, But we create **NAT Gateway** & map that to this)

aws Services

Shiva Basava P N. Virginia Support

Create route table Actions

Filter by tags and attributes or search by keyword

Name	Route Table ID	Explicit st	Edge as	Main	VPC ID	Owner
rtb-0add175e93f307df5		-	-	Yes	vpc-0c9081107866453a2 samp_VPC	973501320577
rtb-eee8a790		-	-	Yes	vpc-834282fe	973501320577
public_RT	rtb-083057542d5670d27	3 subnets	-	No	vpc-0c9081107866453a2 samp_VPC	973501320577

Route Table: rtb-0add175e93f307df5

Summary Routes Subnet Associations Edge Associations Route Propagation Tags

Edit routes

View All routes

Destination	Target	Status	Propagated
10.0.0.0/16	local	active	No
0.0.0.0/0	nat-08d809931d18a064d	active	No

Subnet Association

aws Services

Shiva Basava P N. Virginia Support

Create route table Actions

Filter by tags and attributes or search by keyword

Name	Route Table ID	Explicit st	Edge as	Main	VPC ID	Owner
rtb-0add175e93f307df5		-	-	Yes	vpc-0c9081107866453a2 samp_VPC	973501320577
rtb-eee8a790		-	-	Yes	vpc-834282fe	973501320577
public_RT	rtb-083057542d5670d27	3 subnets	-	No	vpc-0c9081107866453a2 samp_VPC	973501320577

Summary Routes Subnet Associations Edge Associations Route Propagation Tags

Edit subnet associations

None found

You do not have any subnet associations.

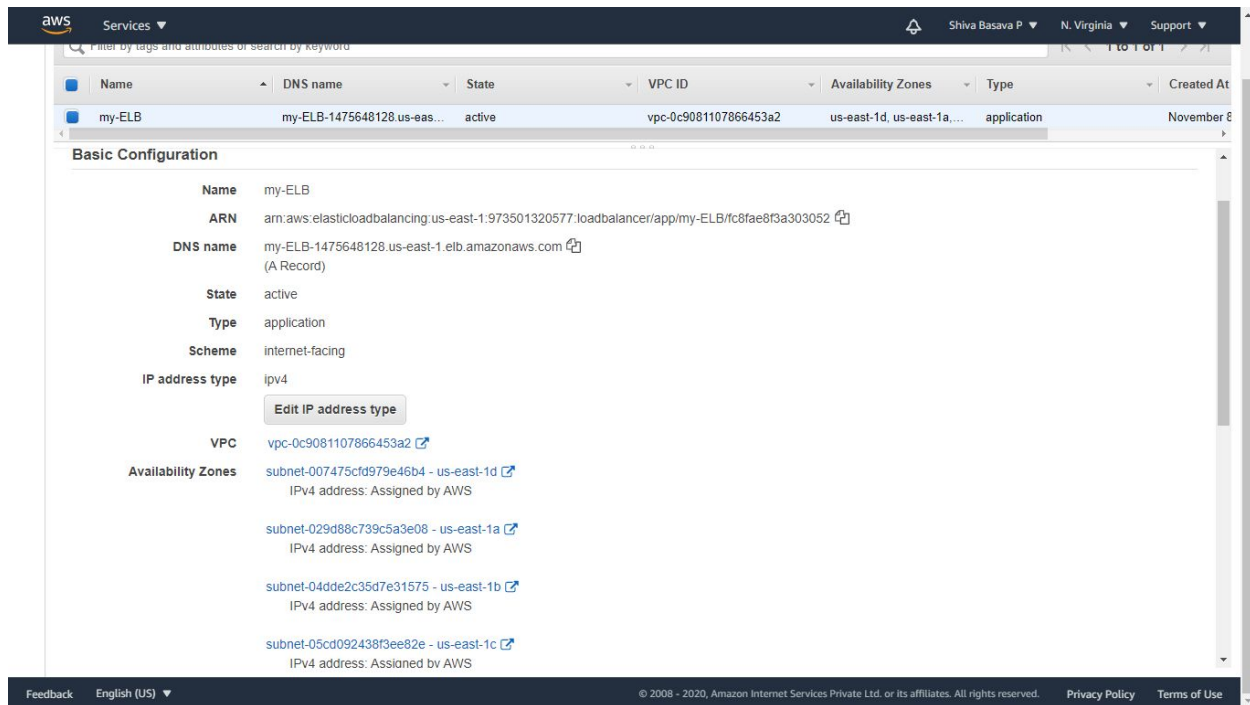
The following subnets have not been explicitly associated with any route tables and are therefore associated with the main route table:

Subnet ID	IPv4 CIDR	IPv6 CIDR
subnet-05cd092438f3e...	10.0.1.0/24	-

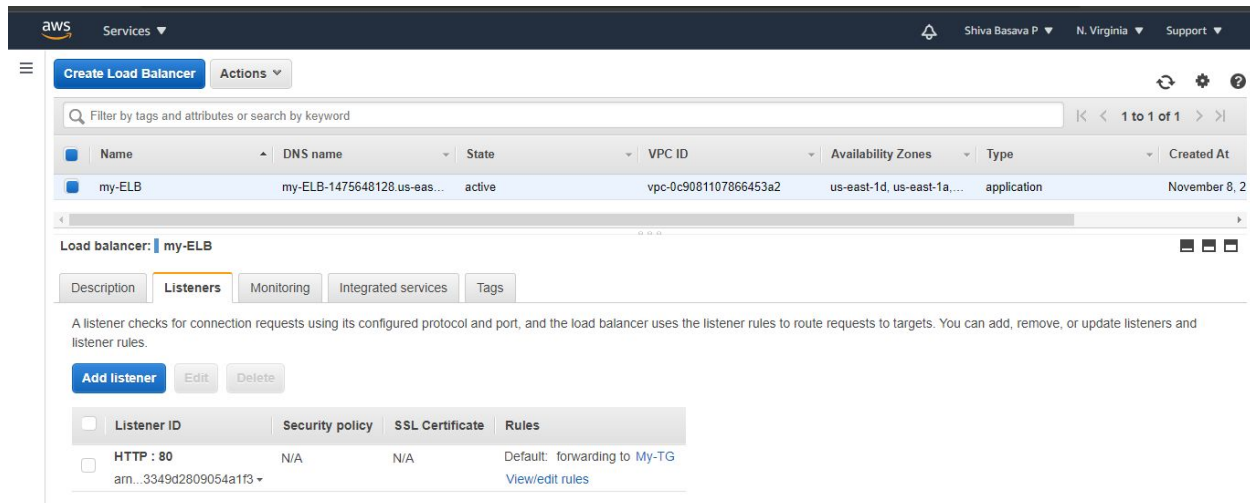
Feedback English (US)

© 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

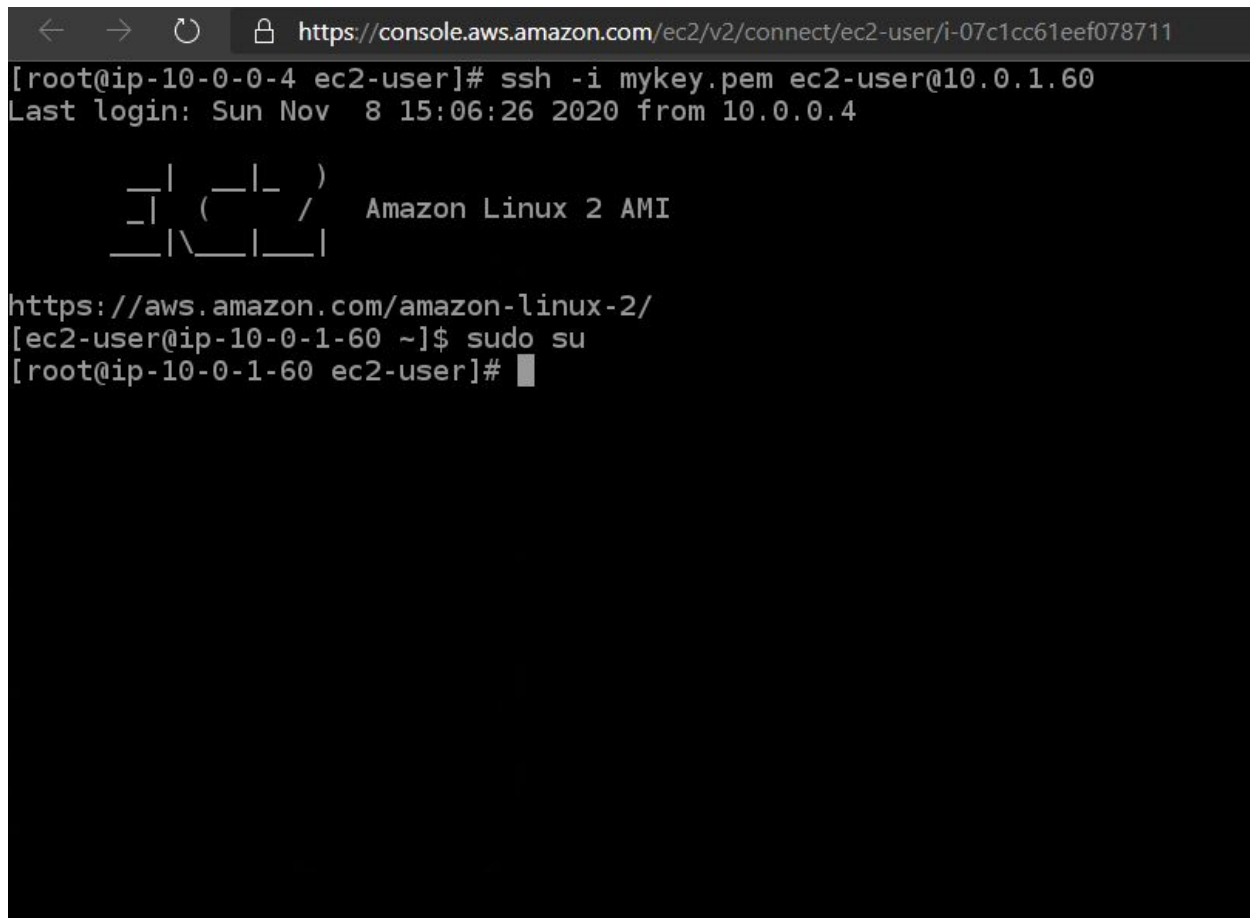
VII. Created Load Balancer - myELB, with DNS Name - my-ELB-1475648128.us-east-1.elb.amazonaws.com



Listeners



- VIII. Connected to **Bastion Server** (Bastion_Server)
 In Super user mode, Performed the update of the server, then created **mykey.pem**
- From **Bastion_Server** SSH to Web-Server 1 with its **private IP - 10.0.1.60**

A screenshot of the AWS Management Console terminal window. The browser address bar shows the URL: https://console.aws.amazon.com/ec2/v2/connect/ec2-user/i-07c1cc61eef078711. The terminal output shows a successful SSH connection to an EC2 instance. The prompt is [root@ip-10-0-0-4 ec2-user]#. The command executed is ssh -i mykey.pem ec2-user@10.0.1.60. The output shows the last login time: Sun Nov 8 15:06:26 2020 from 10.0.0.4. Below this is the Amazon Linux 2 AMI logo, which consists of a stylized 'A' made of lines and the text 'Amazon Linux 2 AMI'. The terminal then shows the URL https://aws.amazon.com/amazon-linux-2/. The prompt changes to [ec2-user@ip-10-0-1-60 ~]\$. The command executed is sudo su. The output shows the prompt changing to [root@ip-10-0-1-60 ec2-user]#.

i-07c1cc61eef078711 (Batsion_server)

Public IPs: 75.101.193.142 Private IPs: 10.0.0.4

Installing Apache:

```
sudo su
```

```
yum update -y
```

```
yum install httpd -y
```

```
cd /var/www/html
```

```
systemctl start httpd
```

```
systemctl enable httpd
```

```
echo "REQUEST HANDLING BY SERVER 1" > index.html
```

```
systemctl status httpd
```

(Again to connect to next web-server, come-out of the current and follow **step b**)

```

https://console.aws.amazon.com/ec2/v2/connect/ec2-user/i-07c1cc61eef078711
https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-0-1-60 ~]$ sudo su
[root@ip-10-0-1-60 ec2-user]# cd /var/www/html/
[root@ip-10-0-1-60 html]# cat index.html
"REQUEST HANDLING BY SERVER 1"
[root@ip-10-0-1-60 html]# systemctl status https
Unit https.service could not be found.
[root@ip-10-0-1-60 html]# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
   Active: active (running) since Sun 2020-11-08 15:08:50 UTC; 1h 3min ago
     Docs: man:httpd.service(8)
  Main PID: 12599 (httpd)
    Status: "Total requests: 352; Idle/Busy workers 100/0; Requests/sec: 0.0918; Bytes served/sec: 44 B/sec"
    CGroup: /system.slice/httpd.service
            └─12599 /usr/sbin/httpd -DFOREGROUND
              └─12600 /usr/sbin/httpd -DFOREGROUND
                └─12601 /usr/sbin/httpd -DFOREGROUND
                  └─12602 /usr/sbin/httpd -DFOREGROUND
                    └─12603 /usr/sbin/httpd -DFOREGROUND
                      └─12604 /usr/sbin/httpd -DFOREGROUND
                        └─12683 /usr/sbin/httpd -DFOREGROUND

Nov 08 15:08:50 ip-10-0-1-60.ec2.internal systemd[1]: Starting The Apache HTTP Server...
Nov 08 15:08:50 ip-10-0-1-60.ec2.internal systemd[1]: Started The Apache HTTP Server.
[root@ip-10-0-1-60 html]#

```

i-07c1cc61eef078711 (Bastion_server)

Public IPs: 75.101.193.142 Private IPs: 10.0.0.4

- b. From **Bastion_Server** SSH to Web-Server_2 with its **private IP - 10.0.1.154**

```

https://console.aws.amazon.com/ec2/v2/connect/ec2-user/i-07c1cc61eef078711
[root@ip-10-0-0-4 ec2-user]# ssh -i mykey.pem ec2-user@10.0.1.154
Last login: Sun Nov 8 16:15:34 2020 from 10.0.0.4

 _ _ | ( _ _ )
 _ | ( _ _ ) /   Amazon Linux 2 AMI
 _ \| \ _ _ | _ |

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-0-1-154 ~]$ sudo su
[root@ip-10-0-1-154 ec2-user]#

```

i-07c1cc61eef078711 (Bastion_server)

Public IPs: 75.101.193.142 Private IPs: 10.0.0.4

Installing Apache: (Same commands till, systemctl enable httpd), then we are tying this -

echo "REQUEST HANDLING BY SERVER 2" > index.html

systemctl status httpd

```

https://console.aws.amazon.com/ec2/v2/connect/ec2-user/i-07c1cc61eef078711
https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-0-1-154 ~]$ sudo su
[root@ip-10-0-1-154 ec2-user]# cd /var/www/html/
[root@ip-10-0-1-154 html]# cat index.html
"REQUEST HANDLING BY SERVER 2"
[root@ip-10-0-1-154 html]# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
   Active: active (running) since Sun 2020-11-08 15:11:52 UTC; 1h 6min ago
     Docs: man:httpd.service(8)
  Main PID: 12607 (httpd)
    Status: "Total requests: 380; Idle/Busy workers 100/0; Requests/sec: 0.0956; Bytes served/sec: 45 B/sec"
    CGroup: /system.slice/httpd.service
            └─12607 /usr/sbin/httpd -DFOREGROUND
              └─12608 /usr/sbin/httpd -DFOREGROUND
                └─12609 /usr/sbin/httpd -DFOREGROUND
                  └─12610 /usr/sbin/httpd -DFOREGROUND
                    └─12611 /usr/sbin/httpd -DFOREGROUND
                      └─12612 /usr/sbin/httpd -DFOREGROUND
                        └─12683 /usr/sbin/httpd -DFOREGROUND
                          └─12689 /usr/sbin/httpd -DFOREGROUND

Nov 08 15:11:52 ip-10-0-1-154.ec2.internal systemd[1]: Starting The Apache HTTP Server...
Nov 08 15:11:52 ip-10-0-1-154.ec2.internal systemd[1]: Started The Apache HTTP Server.
[root@ip-10-0-1-154 html]#

```

i-07c1cc61eef078711 (Batsion_server)

Public IPs: 75.101.193.142 Private IPs: 10.0.0.4

IX. Checking the **Health** of the **Load Balancer**, under Target group -> Registered Targets. Both servers are showing 'Healthy'.

The screenshot shows the AWS Management Console interface for a Target Group named 'My-TG'. The 'Basic configuration' section shows the target type as 'instance', protocol as 'HTTP', port as '80', and VPC as 'vpc-0c9081107866453a2'. The 'Targets' tab is active, showing two registered targets:

Instance ID	Name	Port	Zone	Status	Status details
i-06e355be278d0ef86	Web-Server-1	80	us-east-1c	healthy	
i-0cce3671152087f46	Web-Server-2	80	us-east-1c	healthy	

X. Now Checking the Application by accessing the DNS of the Load Balancer - **my-ELB-1475648128.us-east-1.elb.amazonaws.com**

a. At browser we get the following response from **Web-Server-1**



b. If we keep-on refreshing, we get the Reply from **Web-Server-2**



(We have successfully created a **Bastion Server**, 2 web-server & an Application Load Balancer.)

XI. Test Case for **High Availability**

a. Stopped the **Web-Server-1** & **Web-Server-2** is Running.

Following is the Screenshot for the same

A screenshot of the AWS Management Console 'Instances' page. It shows a table with three instances. The 'Web-Server-1' instance is in a 'Stopped' state, while the others are 'Running'.

	Name	Instance ID	Instance state	I...	Status check	Alarm Status	Availability zone
<input type="checkbox"/>	Web-Server-2	i-0cce3671152087f46	Running	t2...	2/2 checks pa...	No alarms +	us-east-1c
<input type="checkbox"/>	Web-Server-1	i-06e355be278d0ef86	Stopped	t2...	-	No alarms +	us-east-1c
<input type="checkbox"/>	Bastion_server	i-07c1cc61eef078711	Running	t2...	2/2 checks pa...	No alarms +	us-east-1a

At Load Balancer's Target Group we can check for its health, It'll be showing as **unused**

& Target Group is **stopped** status for **Web-Server-1**

My-TG Delete

arn:aws:elasticloadbalancing:us-east-1:973501320577:targetgroup/My-TG/7be693eae7d1c2b

Basic configuration

Target type instance	Protocol : Port HTTP : 80 Protocol version HTTP1	VPC vpc-0c9081107866453a2	Load balancer my-ELB
-------------------------	---	------------------------------	-------------------------

Group details **Targets** Monitoring Tags

Registered targets (2) Refresh Deregister Register targets

Filter resources by property or value

<input type="checkbox"/>	Instance ID	Name	Port	Zone	Status	Status details
<input type="checkbox"/>	i-06e355be278d0ef86	Web-Server-1	80	us-east-1c	unused	Target is in the stopped state
<input type="checkbox"/>	i-0cce3671152087f46	Web-Server-2	80	us-east-1c	healthy	

Feedback English (US) © 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Now if we access application through the link of Load Balancer - **my-ELB-1475648128.us-east-1.elb.amazonaws.com**

We always receive response from **Web-Server-2**

← → ↻ ⓘ Not secure | my-elb-1475648128.us-east-1.elb.amazonaws.com

“REQUEST HANDLING BY SERVER 2”

b. Stopped the **Web-Server-2** & **Web-Server-1** is Running.
Following is the Screenshot for the same

	Name	Instance ID	Instance state	I...	Status check	Alarm Status	Availability zone
<input type="checkbox"/>	Web-Server-2	i-0cce3671152087f46	Stopped	t2...	-	No alarms +	us-east-1c
<input type="checkbox"/>	Web-Server-1	i-06e355be278d0ef86	Running	t2...	2/2 checks pa...	No alarms +	us-east-1c
<input type="checkbox"/>	Batsion_server	i-07c1cc61eef078711	Running	t2...	2/2 checks pa...	No alarms +	us-east-1a

At Load Balancer's Target Group we can check for its health, It'll be showing as **unused** & Target Group is **stopped** status for **Web-Server-2**

Basic configuration							
Target type	Protocol : Port		VPC		Load balancer		
instance	HTTP : 80		vpc-0c9081107866453a2		my-ELB		
	Protocol version						
	HTTP1						

Registered targets (2)							
<input type="checkbox"/>	Instance ID	Name	Port	Zone	Status	Status details	
<input type="checkbox"/>	i-06e355be278d0ef86	Web-Server-1	80	us-east-1c	healthy		
<input type="checkbox"/>	i-0cce3671152087f46	Web-Server-2	80	us-east-1c	unused	Target is in the stopped state	

Now if we access application through the link of Load Balancer - **my-ELB-1475648128.us-east-1.elb.amazonaws.com**

We always receive response from **Web-Server-1**

