

AWS Final Project 1

Deployment of a Highly Available Web Application

-Manoj K

Started with creating a VPC:

The screenshot shows the 'Create VPC' wizard in the AWS VPC Management Console. The 'VPC settings' section is active, displaying the following configuration:

- Name tag - optional:** myvpc
- IPv4 CIDR block:** 10.0.0.0/16
- IPv6 CIDR block:** No IPv6 CIDR block selected.
- Tenancy:** Default

The 'Tags' section contains one tag: Name: myvpc. A note indicates that up to 49 more tags can be added. At the bottom right of the wizard are 'Cancel' and 'Create VPC' buttons.

VPC Name: myvpc

CIDR block: 10.0.0.0/16

Tenancy: Default

VPC Details:

The screenshot shows the AWS VPC Manager interface. On the left, a sidebar lists various VPC-related services: New VPC Experience, VPC Dashboard, Filter by VPC, VIRTUAL PRIVATE CLOUD, Your VPCs, Subnets, Route Tables, Internet Gateways, Egress Only Internet Gateways, Carrier Gateways, DHCP Options Sets, Elastic IPs, Managed Prefix Lists, Endpoints, Endpoint Services, NAT Gateways, Peering Connections, SECURITY, Network ACLs, Security Groups, VIRTUAL PRIVATE NETWORK (VPN), Customer Gateways, and Virtual Private Gateways. The 'Your VPCs' section is expanded, showing two entries: 'vpc-eb7bb996' and 'myvpc'. The 'myvpc' entry is selected, and its details are displayed in the main pane.

Your VPCs (1/2) Info

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR (Network Border Group)	IPv6 pool
vpc-eb7bb996	vpc-0ef0246c86bdff45e	Available	172.31.0.0/16	-	-
myvpc	vpc-0ef0246c86bdff45e	Available	10.0.0.0/16	-	-

Details | CIDs | Flow logs | Tags

Details

VPC ID vpc-0ef0246c86bdff45e	State Available	DNS hostnames Disabled	DNS resolution Enabled
Tenancy Default	DHCP options set dopt-3bfc9341	Route table rtb-064ea3ded65aae4c	Network ACL acl-03ea064c0ac4adcc5
Default VPC No	IPv4 CIDR 10.0.0.0/16	IPv6 pool -	IPv6 CIDR (Network Border Group) -
Owner ID 245024665952			

Creation of public subnet:

The screenshot shows the 'Create subnet' page under the 'Subnets' section. The page title is 'Create subnet'. It prompts the user to specify the subnet's IP address block in CIDR format. Below this, there are fields for 'Name tag' (set to 'publicsubnet'), 'VPC' (set to 'vpc-0ef0246c86bdff45e'), and 'Availability Zone' (set to 'us-east-1a'). A table displays the 'VPC CIDRs' with one entry: 'CIDR' 10.0.0.0/16 and 'Status' 'associated'. At the bottom, the 'IPv4 CIDR block' is set to '10.0.0.0/24'. A note at the bottom left indicates that the 'IPv4 CIDR block' is required. On the right, there are 'Cancel' and 'Create' buttons.

Create subnet

Specify your subnet's IP address block in CIDR format, for example, 10.0.0.0/24. IPv4 block sizes must be between a /16 netmask and /28 netmask, and can be the same size as your VPC. An IPv6 CIDR block must be a /64 CIDR block.

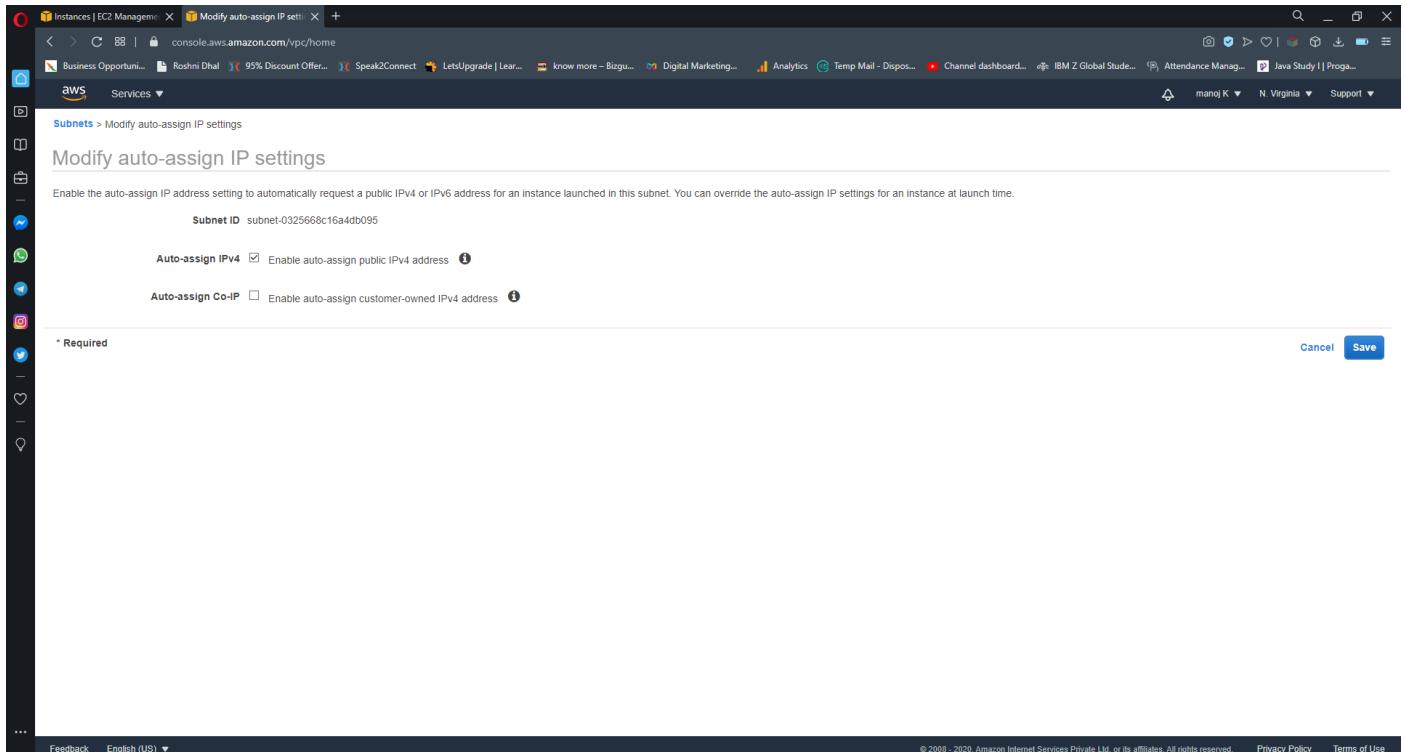
VPC CIDRs	CIDR	Status	Status Reason
	10.0.0.0/16	associated	

IPv4 CIDR block* 10.0.0.0/24

* Required

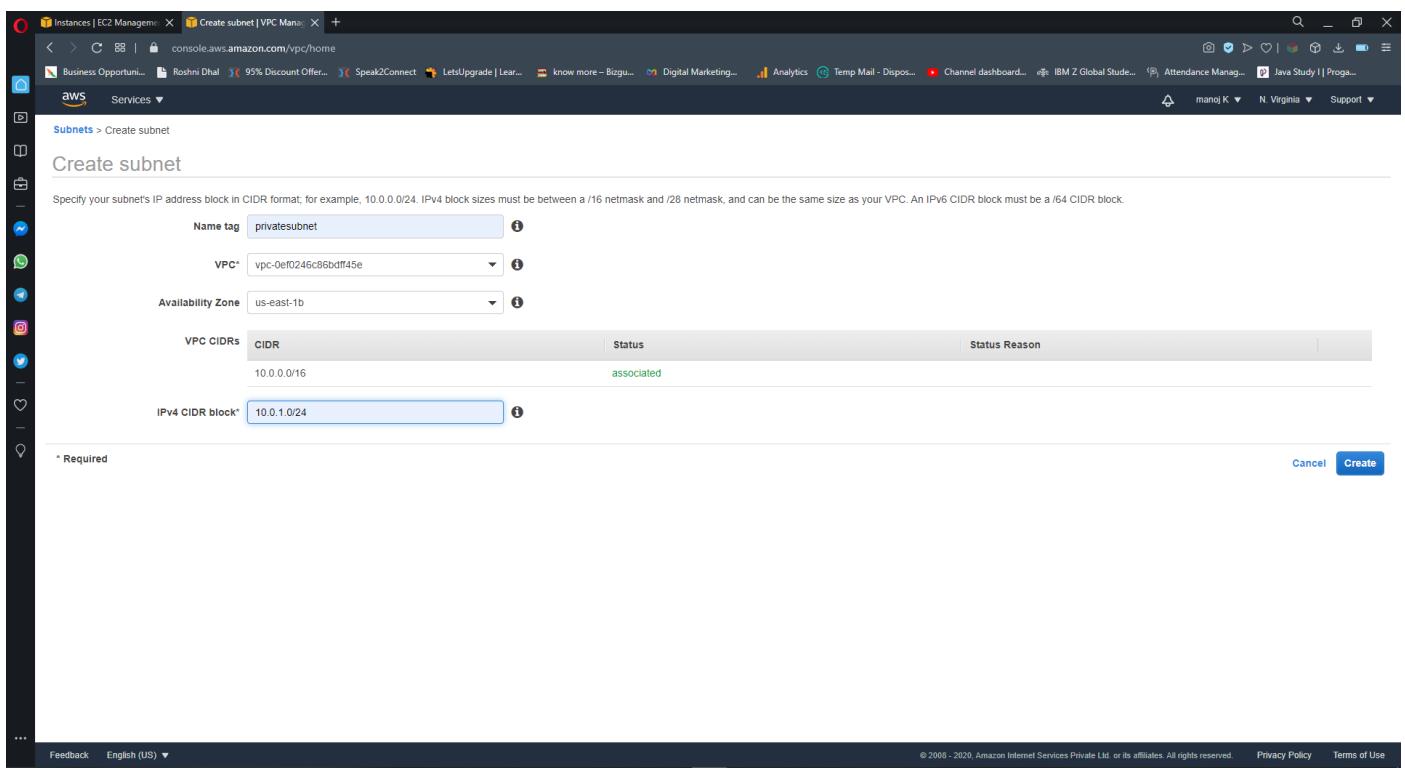
Create

Checking the auto-assign public IPv4 address for public subnet:



The screenshot shows the 'Modify auto-assign IP settings' page for a specific subnet. The 'Auto-assign IPv4' checkbox is checked, indicating that instances launched in this subnet will automatically receive a public IPv4 address. The 'Auto-assign Co-IP' checkbox is unchecked. A note at the top states: 'Enable the auto-assign IP address setting to automatically request a public IPv4 or IPv6 address for an instance launched in this subnet. You can override the auto-assign IP settings for an instance at launch time.' There are 'Cancel' and 'Save' buttons at the bottom right.

Creation of private subnet:



The screenshot shows the 'Create subnet' page. A new subnet is being created with the following details:

- Name tag: privatesubnet
- VPC: vpc-0ef0245c86bdf1f45e
- Availability Zone: us-east-1b
- VPC CIDRs: 10.0.0.0/16 (Status: associated)
- IPv4 CIDR block*: 10.0.1.0/24

A note at the top says: 'Specify your subnet's IP address block in CIDR format, for example, 10.0.0.0/24. IPv4 block sizes must be between a /16 netmask and /28 netmask, and can be the same size as your VPC. An IPv6 CIDR block must be a /64 CIDR block.' There are 'Cancel' and 'Create' buttons at the bottom right.

Creating public route table:

The screenshot shows the 'Create route table' page in the AWS VPC console. A route table named 'publicroutetable' is being created under VPC 'vpc-0ef0246c86bdff45e'. The 'Key' field is empty. There is one tag added: 'Name' with value 'publicroutetable'. The 'Create' button is visible at the bottom right.

Associating public route table with public subnet:

The screenshot shows the 'Edit subnet associations' page for route table 'rtb-0165bb5ae8e802253'. It lists two subnets: 'publicsubnet' (IPv4 CIDR: 10.0.0.0/24) and 'privatesubnet' (IPv4 CIDR: 10.0.1.0/24). Both are associated with the 'Main' route table. The 'Save' button is visible at the bottom right.

Subnet ID	IPv4 CIDR	IPv6 CIDR	Current Route Table
subnet-0325668c16a4db095 publicsubnet	10.0.0.0/24	-	Main
subnet-03023c15e588a279 privatesubnet	10.0.1.0/24	-	Main

Creating private route table :

The screenshot shows the 'Create route table' page in the AWS VPC console. A route table named 'privateroutetable' is being created under VPC 'vpc-0ef0246c86bdff45e'. The 'Name tag' field contains 'privateroutetable'. The 'VPC' dropdown is set to 'vpc-0ef0246c86bdff45e'. There are no tags added yet. The 'Create' button is visible at the bottom right.

Creating private route table with private subnet:

The screenshot shows the 'Edit subnet associations' page for route table 'rtb-0ad72ba06527f2b9a'. It lists two subnets: 'publicsubnet' and 'privatesubnet'. The 'privatesubnet' is selected and associated with the 'Main' route table. The 'Save' button is visible at the bottom right.

Subnet ID	IPv4 CIDR	IPv6 CIDR	Current Route Table
subnet-0325668c16a4db095 publicsubnet	10.0.0.0/24	-	rtb-0165bb5ae8e802253
subnet-03023c158e588a279 privatesubnet	10.0.1.0/24	-	Main

Attaching Internet Gateway with the VPC:

The screenshot shows the 'Attach to VPC' dialog box. At the top, it says 'Attach to VPC (igw-037429c961de560dd)'. Below that is a section titled 'VPC' with the sub-instruction 'Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below.' A search bar contains the text 'vpc-0ef0246c86bdff45e'. At the bottom right are 'Cancel' and 'Attach internet gateway' buttons.

Created Internet Gateway Details:

The screenshot shows the 'Internet gateways' list page. A green banner at the top states 'Internet gateway igw-037429c961de560dd successfully attached to vpc-0ef0246c86bdff45e'. The main table lists two entries: 'igw1' (attached to 'vpc-0ef0246c86bdff45e | myvpc') and 'igw-92a392e9' (attached to 'vpc-eb7bb996'). Below the table, a detailed view for 'igw-037429c961de560dd / igw1' is shown, with tabs for 'Details' and 'Tags'. The 'Details' tab displays information such as Internet gateway ID (igw-037429c961de560dd), State (Attached), VPC ID (vpc-0ef0246c86bdff45e | myvpc), and Owner (245024665952).

Editing routes in public route table i.e adding internet gateway:

The screenshot shows the 'Edit routes' page for a public route table. A new route is being added with a destination of 0.0.0.0/0 and a target of igw-037429c961de560dd. The 'Save routes' button is highlighted.

Public route table details:

The screenshot shows the 'Create route table' page. It lists existing route tables: publicroutetable and privateroutetable. The publicroutetable is selected. Below it, the 'Route Table: rtb-0165bb5ae8e802253' is shown with its routes. The 'Routes' tab is selected, showing two routes: one to 10.0.0.16 (target: local) and another to 0.0.0.0/0 (target: igw-037429c961de560dd).

Creating NAT gateway:

The screenshot shows the 'Create NAT gateway' wizard in the AWS VPC Management Console. The 'NAT gateway settings' section includes fields for 'Name' (mynat), 'Subnet' (subnet-0325668c16a4db095), and 'Elastic IP allocation ID' (eipalloc-090874b05a0492440). The 'Tags' section contains a single tag 'Name: mynat'. At the bottom are 'Cancel' and 'Create NAT gateway' buttons.

Successful Creation of NAT:

The screenshot shows the 'NAT gateways' page in the AWS VPC Management Console. A green success message at the top states 'NAT gateway nat-076cb377fa130ac0c | mynat was created successfully.' The main table displays the newly created NAT gateway details: NAT gateway ID (nat-076cb377fa130ac0c), State (Pending), Private IP address (-), Network interface ID (vpc-0ef0246c86bdff45e / myvpc), State message (-), Elastic IP address (-), Subnet (subnet-0325668c16a4db095 / publicsubnet), and Created (2020/11/01 15:33 GMT+5:30). Below the table are tabs for 'Monitoring' and 'Tags', and a chart showing network traffic metrics.

NAT Gateway Details:

The screenshot shows the AWS VPC Management Console with the 'NAT gateways' page. A single NAT gateway named 'mynat' is listed in the table. The table columns include Name, NAT gateway ID, State, State message, Elastic IP address, Private IP address, Network interface ID, and VPC. The 'mynat' entry has the following details:

Name	NAT gateway ID	State	State message	Elastic IP address	Private IP address	Network interface ID	VPC
mynat	nat-076cb377fa130ac0c	Available	-	107.20.114.66	10.0.0.178	eni-0d639f0868ede2cf0	vpc-0ef0246c86bdff45e / myvpc

Below the table, a detailed view for 'nat-076cb377fa130ac0c / mynat' is shown. The 'Details' tab is selected, displaying the following information:

NAT gateway ID	State	State message	Elastic IP address
nat-076cb377fa130ac0c	Available	-	107.20.114.66
Private IP address	Network interface ID	VPC	Subnet
10.0.0.178	eni-0d639f0868ede2cf0	vpc-0ef0246c86bdff45e / myvpc	subnet-0325668c16a4db095 / publicsubnet
Created	Deleted	-	
2020/11/01 15:33 GMT+5:30			

Editing Routes in private route Table i.e allowing NAT Gateway:

The screenshot shows the AWS VPC Management Console with the 'Edit routes' page for a route table. Two routes are listed in the table:

Destination	Target	Status	Propagated
10.0.0.0/16	local	active	No
0.0.0.0/0	nat-076cb377fa130ac0c	No	

At the bottom of the page, there is a 'Save routes' button.

Private Route table Details:

The screenshot shows the AWS VPC Management console. On the left, there's a sidebar with various VPC-related options like VPC Dashboard, Your VPCs, Route Tables, Security, and Virtual Private Network (VPN). The main area shows a table of route tables. One route table, 'privateroutetable', is selected and expanded to show its details. The table has columns for Name, Route Table ID, Explicit subnet association, Edge associations, Main, VPC ID, and Owner. The selected route table has rows for 'rtb-0ad72ba06527f2b9a' and 'rtb-f4f8b18a'. Below the table, there's a tabular view of routes with columns for Destination, Target, Status, and Propagated.

Destination	Target	Status	Propagated
10.0.0.0/16	local	active	No
0.0.0.0/0	nat-076cb37fa130ac0c	active	No

Now Let's create instance for Public Subnet.

Creating instance in Public Subnet which is called Bastion-Server:

Through which private instance can be accessed.

The screenshot shows the AWS Launch Instance Wizard, Step 3: Configure Instance Details. It's a form with various configuration options. At the top, it says 'Step 3: Configure Instance Details' and 'Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.' The form includes fields for Number of Instances (set to 1), Purchasing option (Request Spot Instances), Network (selected vpc-0ef0246c86bdff45e | myvc), Subnet (selected subnet-0325668c16a4db095 | publicsubnet | us-east-1), and IAM role (None). Other sections include CPU options, Shutdown behavior (Stop), Stop - Hibernate behavior (Enable hibernation as an additional stop behavior), Enable termination protection (Protect against accidental termination), Monitoring (Enable CloudWatch detailed monitoring), Tenancy (Shared - Run a shared hardware instance), and Elastic Inference (Add an Elastic Inference accelerator). At the bottom, there are buttons for Cancel, Previous, Review and Launch (highlighted in blue), and Next: Add Storage.

Storage for Bastion-Server:

Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/xvda	snap-0299d083f0ce6cd12	8	General Purpose SSD (gp2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Add New Volume

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

Cancel Previous Review and Launch Next: Add Tags

Adding tags for Bastion-server:

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. A copy of a tag can be applied to volumes, instances or both. Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key	(128 characters maximum)	Value	(256 characters maximum)	Instances	Volumes
Name		bastion-server		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Add another tag (Up to 50 tags maximum)

Cancel Previous Review and Launch Next: Configure Security Group

Configuring Security Group for Bastion Server:

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: Create a new security group Select an existing security group

Security group name:

Description:

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Anywhere	e.g. SSH for Admin Desktop

Add Rule

Warning
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Cancel Previous Review and Launch

Review of instance for Bastion Server:

Step 7: Review Instance Launch

Your instances may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only.
You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. [Edit security groups](#)

AMI Details [Edit AMI](#)

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0947d2ba12ee1ff75

Free tier eligible

Root Device Type: ebs Virtualization type: hvm

Instance Type [Edit instance type](#)

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	-	1	1	EBS only	-	Low to Moderate

Security Groups [Edit security groups](#)

Security group name: bastion-server-sg
Description: security group for bastion server

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	0.0.0.0/0	
SSH	TCP	22	::/0	

Instance Details [Edit instance details](#)

Number of instances: 1
Network: vpc-0ef0246c86bdff45e
Subnet: subnet-0325668c16a4db095
EBS-optimized: No
Monitoring: No

Cancel Previous Launch

Bastion Server details:

The screenshot shows the AWS EC2 Management Console. On the left, the navigation pane includes links for EC2 Dashboard, Events, Tags, Limits, Instances (selected), 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), and Feedback. The main content area displays the 'Instances (1/1)' page with a table showing one instance: 'bastion-server' (Instance ID: i-0c6f2020c705f71a1, State: Running, Type: t2.micro, Status check: 2/2 checks ...). Below the table is the 'Instance summary' section for the selected instance, showing details like Public IPv4 address (54.211.197.35), Private IPv4 addresses (10.0.0.142), and VPC ID (vpc-0ef0246c86bdff45e). The bottom of the page includes standard footer links for Feedback, English (US), Privacy Policy, and Terms of Use.

Creating Security Group for loadbalancer:

The screenshot shows the 'Create security group' wizard in the AWS EC2 Management Console. The 'Basic details' step is active, with fields for Security group name (loadbalancer-sg), Description (security group for loadbalancer), and VPC (vpc-0ef0246c86bdff45e (myvpc)). The 'Inbound rules' step is shown below, featuring a table with columns for Type (HTTP, TCP), Protocol (TCP, TCP), Port range (80), Source (Custom, 0.0.0.0/0), and Description - optional. An 'Add rule' button is available. The 'Outbound rules' step is also visible at the bottom. The bottom of the page includes standard footer links for Feedback, English (US), Privacy Policy, and Terms of Use.

Load balancer security group creation:

The screenshot shows the AWS EC2 Management Console with the 'Security Groups' section open. A success message at the top states: "Security group (sg-0015061d3bc61e679 | loadbalancer-sg) was created successfully". The main table displays the following information:

Security group name	Security group ID	Description	VPC ID
loadbalancer-sg	sg-0015061d3bc61e679	security group for loadbalancer	vpc-0ef0246c86bdff45e
Owner	Inbound rules count	Outbound rules count	
245024665952	1 Permission entry	1 Permission entry	

The 'Inbound rules' tab is selected, showing one rule:

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

Now it's time to create web-servers in private network

Creation of web-server 1 in private subnet:

The screenshot shows the AWS Launch Instance Wizard at Step 3: Configure Instance Details. The 'Configure Security Group' tab is selected. The configuration fields include:

- Number of instances: 1
- Purchasing option: Request Spot instances
- Network: vpc-0ef0246c86bdff45e | myvpc
- Subnet: Subnet-03023c158e589a279 | privatesubnet | us-east-1
- Auto-assign Public IP: Use subnet setting (Disable)
- Placement group: None
- Capacity Reservation: Open
- Domain join directory: No directory
- IAM role: None
- CPU options: None
- Shutdown behavior: Stop
- Stop - Hibernate behavior: None
- Enable termination protection: None
- Monitoring: None
- Tenancy: Shared - Run a shared hardware instance
- Elastic Inference: None

At the bottom, there are buttons for 'Cancel', 'Previous', 'Review and Launch', and 'Next: Add Storage'.

Adding storage to web-server 1:

Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/xvda	snap-0299d083f0ce6cd12	8	General Purpose SSD (gp2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted

Add New Volume

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

Cancel Previous **Review and Launch** Next: Add Tags

Configuring security group for web-server:

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: Create a new security group Select an existing security group

Security group name:
Description:

Type	Protocol	Port Range	Source	Description
SSH	TCP	[22]	Custom sg-0755301c5c9cd9f61	e.g. SSH for Admin Desktop
HTTP	TCP	[80]	Custom sg-0015061d3bc61e679	e.g. SSH for Admin Desktop

Add Rule

Cancel Previous **Review and Launch** Next: Add Tags

Launching the web-server 1:

The screenshot shows the AWS EC2 Management Console with the 'Launch instance wizard' open. The current step is 'Step 7: Review Instance Launch'. The main page displays 'AMI Details' (Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0947d2ba12ee1ff75), 'Instance Type' (t2.micro), and 'Security Groups' (web-server-sg). A modal window titled 'Select an existing key pair or create a new key pair' is overlaid on the page. The modal contains instructions about key pairs, a dropdown menu with 'Choose an existing key pair' and 'Select a key pair' options, and a note about acknowledging access to the private key file. At the bottom of the modal are 'Cancel' and 'Launch Instances' buttons.

Creation of web-server 2:

The screenshot shows the AWS EC2 Management Console with the 'Launch instance wizard' open. The current step is 'Step 3: Configure Instance Details'. The page allows configuring 'Number of instances' (1), 'Purchasing option' (Request Spot instances), 'Network' (vpc-0ef0246c86bdf45e | mynyc), 'Subnet' (subnet-03023c158e589a279 | privatesubnet | us-eas), 'Auto-assign Public IP' (Use subnet setting (Disable)), 'Placement group' (Add instance to placement group), 'Capacity Reservation' (Open), 'Domain join directory' (No directory), 'IAM role' (None), 'CPU options' (Specify CPU options), 'Shutdown behavior' (Stop), 'Stop - Hibernate behavior' (Enable hibernation as an additional stop behavior), 'Enable termination protection' (Protect against accidental termination), 'Monitoring' (Enable CloudWatch detailed monitoring), 'Tenancy' (Shared - Run a shared hardware instance), and 'Elastic Inference' (Add an Elastic Inference accelerator). At the bottom of the page are 'Cancel', 'Previous', 'Review and Launch' (highlighted in blue), and 'Next: Add Storage' buttons.

Adding storage to web-server 2:

Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/xvda	snap-0299d083fce6cd12	8	General Purpose SSD (gp2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted

Add New Volume

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

Cancel Previous **Review and Launch** Next: Add Tags

Adding tags for web-server 2:

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. A copy of a tag can be applied to volumes, instances or both. Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key	(128 characters maximum)	Value	(256 characters maximum)	Instances	Volumes
Name		web-server-2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Add another tag (Up to 50 tags maximum)

Cancel Previous **Review and Launch** Next: Configure Security Group

Selecting Security group from previous web-server:

The screenshot shows the AWS EC2 Management Console with the 'Launch instance wizard' open. The current step is '6. Configure Security Group'. A sub-step titled 'Step 6: Configure Security Group' is displayed. It asks 'Assign a security group:' with two options: 'Create a new security group' (radio button not selected) and 'Select an existing security group' (radio button selected). Below this is a table of existing security groups:

Security Group ID	Name	Description	Actions
sg-0755301c5c9cd9f61	bastion-server-sg	security group for bastion server	Copy to new
sg-0ad93551e2374000	default	default VPC security group	Copy to new
sg-0015061d3bc61e679	loadbalancer-sg	security group for loadbalancer	Copy to new
sg-0b3101cf035e8cc37	web-server-sg	security group for web server	Copy to new

Below the table, it says 'Inbound rules for sg-0b3101cf035e8cc37 (Selected security groups: sg-0b3101cf035e8cc37)'. A table shows the inbound rules:

Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	sg-0015061d3bc61e679 (loadbalancer-sg)	
SSH	TCP	22	sg-0755301c5c9cd9f61 (bastion-server-sg)	

At the bottom right are 'Cancel', 'Previous', 'Review and Launch' buttons.

Launching the web-server 2:

The screenshot shows the AWS EC2 Management Console with the 'Launch instance wizard' open. The current step is '7. Review'. A sub-step titled 'Step 7: Review Instance Launch' is displayed. It says 'Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.' Below this are sections for 'AMI Details', 'Instance Type', and 'Security Groups'.

In the 'Security Groups' section, it shows 'All selected security groups inbound rules' with the same configuration as the previous screenshot.

A modal dialog titled 'Select an existing key pair or create a new key pair' is open. It contains instructions: 'A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.' It also says 'Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#)'. A dropdown menu shows 'Choose an existing key pair' and 'Select a key pair' with 'web-server' selected. A checkbox is checked with the text 'I acknowledge that I have access to the selected private key file (web-server.pem), and that without this file, I won't be able to log into my instance.' At the bottom are 'Cancel' and 'Launch Instances' buttons.

At the bottom right are 'Edit instance type', 'Edit security groups', 'Edit instance details', 'Edit storage', 'Cancel', 'Previous', and 'Launch' buttons.

Web-server 1 Details:

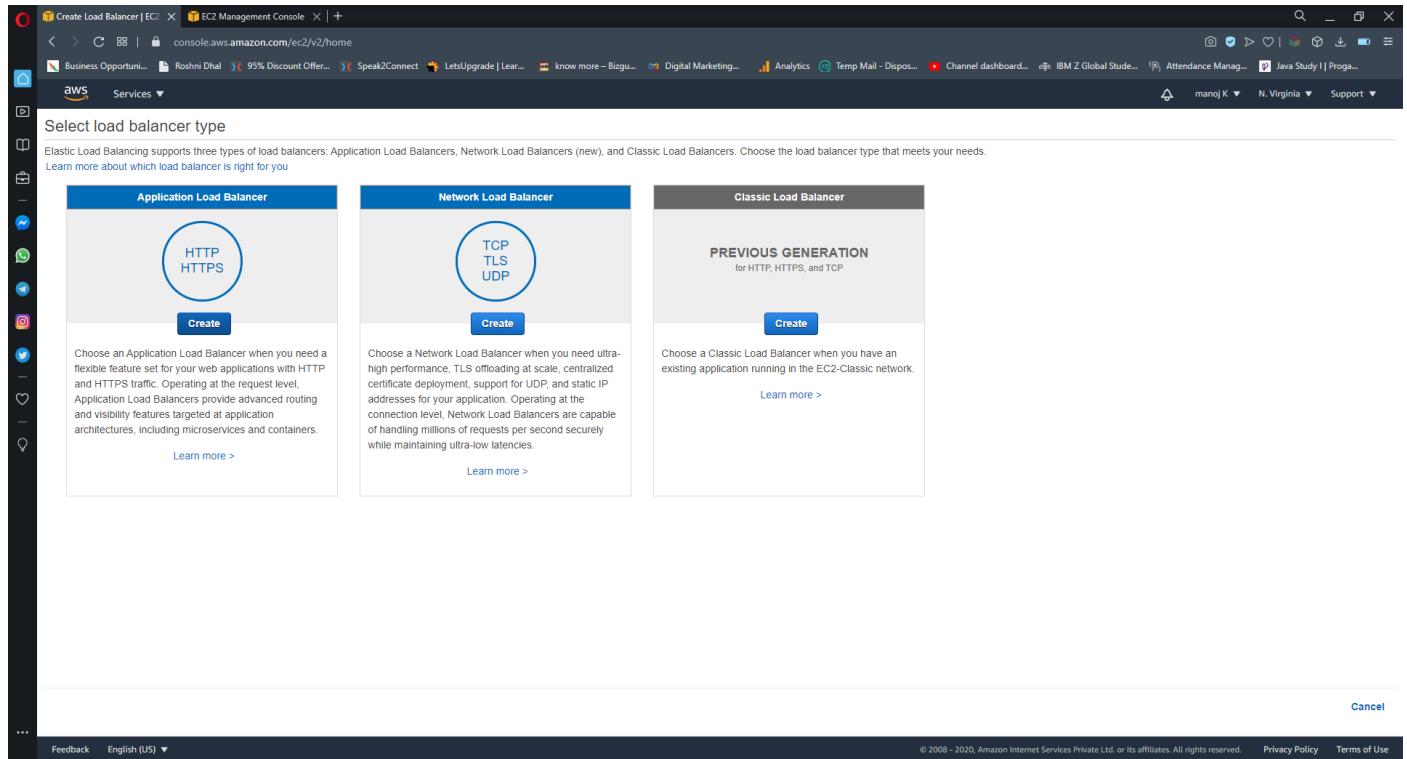
The screenshot shows the AWS EC2 Management Console. On the left, a sidebar navigation menu includes: New EC2 Experience, EC2 Dashboard, Events, Tags, Limits, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations, Images (AMIs), Elastic Block Store (Volumes, Snapshots, Lifecycle Manager), Network & Security (Security Groups, Elastic IPs, Placement Groups, Key Pairs), and Support. The main content area displays the 'Instances (1/3) Info' table with three rows: bastion-server, web-server-1 (selected), and web-server-2. The 'Details' tab is selected for web-server-1, showing its configuration: Instance ID (i-0302839db79ec6175), Instance state (Running), Instance type (t2.micro), Status check (2/2 checks ...), Alarm Status (No alarms), Availability zone (us-east-1a), Public IPv4 DNS (54.211.197.35), and Private IPv4 address (10.0.1.209). The 'Instance summary' section also lists Public IPv4 DNS (ip-10-0-1-209.ec2.internal), VPC ID (vpc-0ef0246c86bdff45e (myvpc)), and Subnet ID (subnet-03023c158e588a279 (privatesubnet)).

Web-server 2 Details:

The screenshot shows the AWS EC2 Management Console. The sidebar navigation menu is identical to the previous screenshot. The main content area displays the 'Instances (1/3) Info' table with three rows: bastion-server, web-server-1, and web-server-2 (selected). The 'Details' tab is selected for web-server-2, showing its configuration: Instance ID (i-07ecb048677d8ff6e), Instance state (Running), Instance type (t2.micro), Status check (2/2 checks ...), Alarm Status (No alarms), Availability zone (us-east-1b), Public IPv4 DNS (54.211.197.35), and Private IPv4 address (10.0.1.228). The 'Instance summary' section also lists Public IPv4 DNS (ip-10-0-1-228.ec2.internal), VPC ID (vpc-0ef0246c86bdff45e (myvpc)), and Subnet ID (subnet-03023c158e588a279 (privatesubnet)).

After creating web server

Let us creating a load balancer for web servers:



Select load balancer type

Elastic Load Balancing supports three types of load balancers: Application Load Balancers, Network Load Balancers (new), and Classic Load Balancers. Choose the load balancer type that meets your needs.

Learn more about which load balancer is right for you

Application Load Balancer

Choose an Application Load Balancer when you need a flexible feature set for your web applications with HTTP and HTTPS traffic. Operating at the request level, Application Load Balancers provide advanced routing and visibility features targeted at application architectures, including microservices and containers.

[Learn more >](#)

Network Load Balancer

Choose a Network Load Balancer when you need ultra-high performance, TLS offloading at scale, centralized certificate deployment, support for UDP, and static IP addresses for your application. Operating at the connection level, Network Load Balancers are capable of handling millions of requests per second securely while maintaining ultra-low latencies.

[Learn more >](#)

Classic Load Balancer

PREVIOUS GENERATION for HTTP, HTTPS, and TCP

Choose a Classic Load Balancer when you have an existing application running in the EC2-Classic network.

[Learn more >](#)

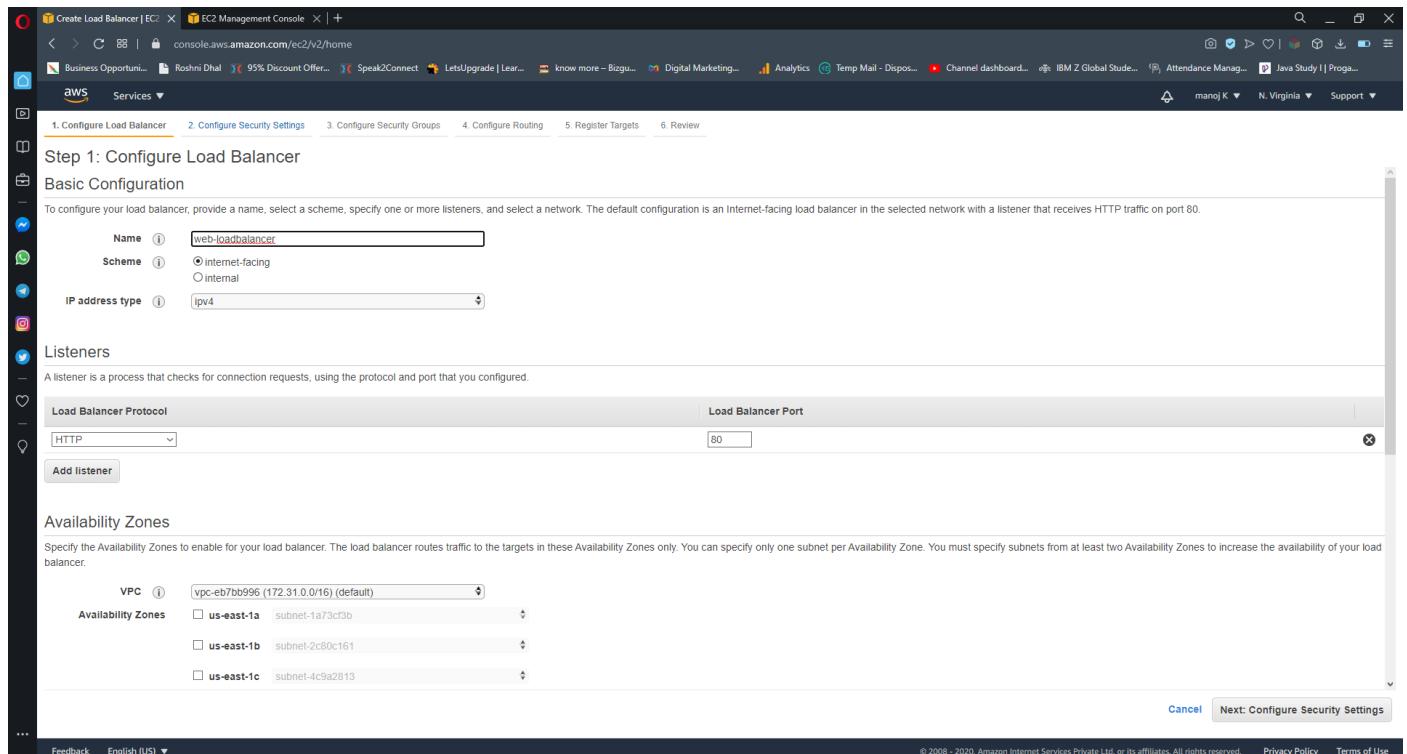
[Create](#)

[Create](#)

[Create](#)

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

Configuration of load-balancer:



1. Configure Load Balancer 2. Configure Security Settings 3. Configure Security Groups 4. Configure Routing 5. Register Targets 6. Review

Step 1: Configure Load Balancer

Basic Configuration

To configure your load balancer, provide a name, select a scheme, specify one or more listeners, and select a network. The default configuration is an Internet-facing load balancer in the selected network with a listener that receives HTTP traffic on port 80.

Name:

Scheme: internet-facing internal

IP address type:

Listeners

A listener is a process that checks for connection requests, using the protocol and port that you configured.

Load Balancer Protocol	Load Balancer Port
HTTP	80

Add listener

Availability Zones

Specify the Availability Zones to enable for your load balancer. The load balancer routes traffic to the targets in these Availability Zones only. You can specify only one subnet per Availability Zone. You must specify subnets from at least two Availability Zones to increase the availability of your load balancer.

VPC	Subnet
vpc-eb7bb996 (172.31.0.0/16) (default)	us-east-1a: subnet-1a7cfcb
	us-east-1b: subnet-2c80c161
	us-east-1c: subnet-4c9a2813

[Cancel](#) [Next: Configure Security Settings](#)

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

Configuration of load-balancer:

The screenshot shows the 'Create Load Balancer' wizard in the AWS EC2 Management Console. The current step is 'Step 1: Configure Load Balancer'. The configuration includes:

- Scheme:** internet-facing
- IP address type:** ipv4
- Listeners:** A listener for port 80 using the Load Balancer Protocol (HTTP).
- Availability Zones:** us-east-1a and us-east-1b, both assigned by AWS.

A note at the bottom states: "You are creating an internet-facing Load Balancer, but there is no Internet Gateway attached to these subnets you have selected: subnet-03023c158e588a279".

Security setting of load-balancer:

The screenshot shows the 'Create Load Balancer' wizard in the AWS EC2 Management Console, currently at Step 2: Configure Security Settings. A warning message is displayed:

⚠ Improve your load balancer's security. Your load balancer is not using any secure listener.
If your traffic to the load balancer needs to be secure, use the HTTPS protocol for your front-end connection. You can go back to the first step to add/configure secure listeners under [Basic Configuration](#) section. You can also continue with current settings.

At the bottom right, there are buttons for 'Cancel', 'Previous', and 'Next: Configure Security Groups'.

Selecting Security group for load balancer:

Step 3: Configure Security Groups

A security group is a set of firewall rules that control the traffic to your load balancer. On this page, you can add rules to allow specific traffic to reach your load balancer. First, decide whether to create a new security group or select an existing one.

Assign a security group:

- Create a new security group
- Select an existing security group

Security Group ID	Name	Description	Actions
sg-0755301c5c9cd9f61	bastion-server-sg	security group for bastion server	Copy to new
sg-0adb93551e2374000	default	default VPC security group	Copy to new
<input checked="" type="checkbox"/> sg-0015061d3b061ee679	loadbalancer-sg	security group for loadbalancer	Copy to new
sg-0b3101cf035e8cc37	web-server-sg	security group for web server	Copy to new

Filter [VPC security groups]

Cancel Previous Next: Configure Routing

Configuring routing for load balancer:

Step 4: Configure Routing

Your load balancer routes requests to the targets in this target group using the protocol and port that you specify, and performs health checks on the targets using these health check settings. The target group you specify in this step will apply to all of the listeners configured on this load balancer; you can edit the listeners and add listeners after the load balancer is created.

Target group

Target group: New target group

Name: new target 1

Target type:

- Instance
- IP
- Lambda function

Protocol: HTTP

Port: 80

Protocol version:

- HTTP1 Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2.
- HTTP2 Send requests to targets using HTTP/2. Supported when the request protocol is HTTP/2 or gRPC, but gRPC-specific features are not available.
- gRPC Send requests to targets using gRPC. Supported when the request protocol is gRPC.

Health checks

Protocol: HTTP

Path: /index.html

Advanced health check settings

Cancel Previous Next: Register Targets

Registering targets for load-balancer:

Step 5: Register Targets

Register targets with your target group. If you register a target in an enabled Availability Zone, the load balancer starts routing requests to the targets as soon as the registration process completes and the target passes the initial health checks.

Registered targets

To deregister instances, select one or more registered instances and then click Remove.

Instance	Name	Port	State	Security groups	Zone
i-0302839db79ec6175	web-server-1	80	running	web-server-sg	us-east-1b
i-07ecb048677d8ff6e	web-server-2	80	running	web-server-sg	us-east-1b

Instances

To register additional instances, select one or more running instances, specify a port, and then click Add. The default port is the port specified for the target group. If the instance is already registered on the specified port, you must specify a different port.

Instance	Name	State	Security groups	Zone	Subnet ID	Subnet CIDR
i-0c6f2020c705f71a1	bastion-server	running	bastion-server-sg	us-east-1a	subnet-0325668c16a4db095	10.0.0.0/24
i-0302839db79ec6175	web-server-1	running	web-server-sg	us-east-1b	subnet-03023c158e588a279	10.0.1.0/24
i-07ecb048677d8ff6e	web-server-2	running	web-server-sg	us-east-1b	subnet-03023c158e588a279	10.0.1.0/24

Add to registered on port 80

Search Instances

Cancel Previous Next: Review

Reviewing Load balancer:

Step 6: Review

Please review the load balancer details before continuing

Load balancer

Name	web-loadbalancer
Scheme	internet-facing
Listeners	Port 80 - Protocol:HTTP
IP address type	IPv4
VPC	vpc-0e0246c86bd745e (myvpc)
Subnets	subnet-0325668c16a4db095 (publicsubnet), subnet-03023c158e588a279 (privatesubnet)
Tags	

Security groups

Security groups	sg-0015061d3bc61e679
-----------------	----------------------

Routing

Target group	New target group
Target group name	new-target-1
Port	80
Target type	instance
Protocol	HTTP
Protocol version	HTTP1
Health check protocol	HTTP
Path	/index.html
Health check port	traffic port
Healthy threshold	5
Unhealthy threshold	10
Timeout	10
Interval	30
Success codes	200

Targets

Instances	I-0302839db79ec6175 (web-server-1):80, I-07ecb048677d8ff6e (web-server-2):80
-----------	--

Cancel Previous Create

Successful Creation of Load Balancer:

The screenshot shows the AWS EC2 Management Console with a success message: "Successfully created load balancer". The message states that the load balancer "web-loadbalancer" was successfully created and notes that it might take a few minutes for the load balancer to be fully set up and ready to route traffic. It also suggests integrating with other services like AWS Global Accelerator.

Suggested next steps

- Discover other services that you can integrate with your load balancer. Visit the **Integrated services** tab within [web-loadbalancer](#).
- Consider using AWS Global Accelerator to further improve the availability and performance of your applications. [AWS Global Accelerator console](#)

Close

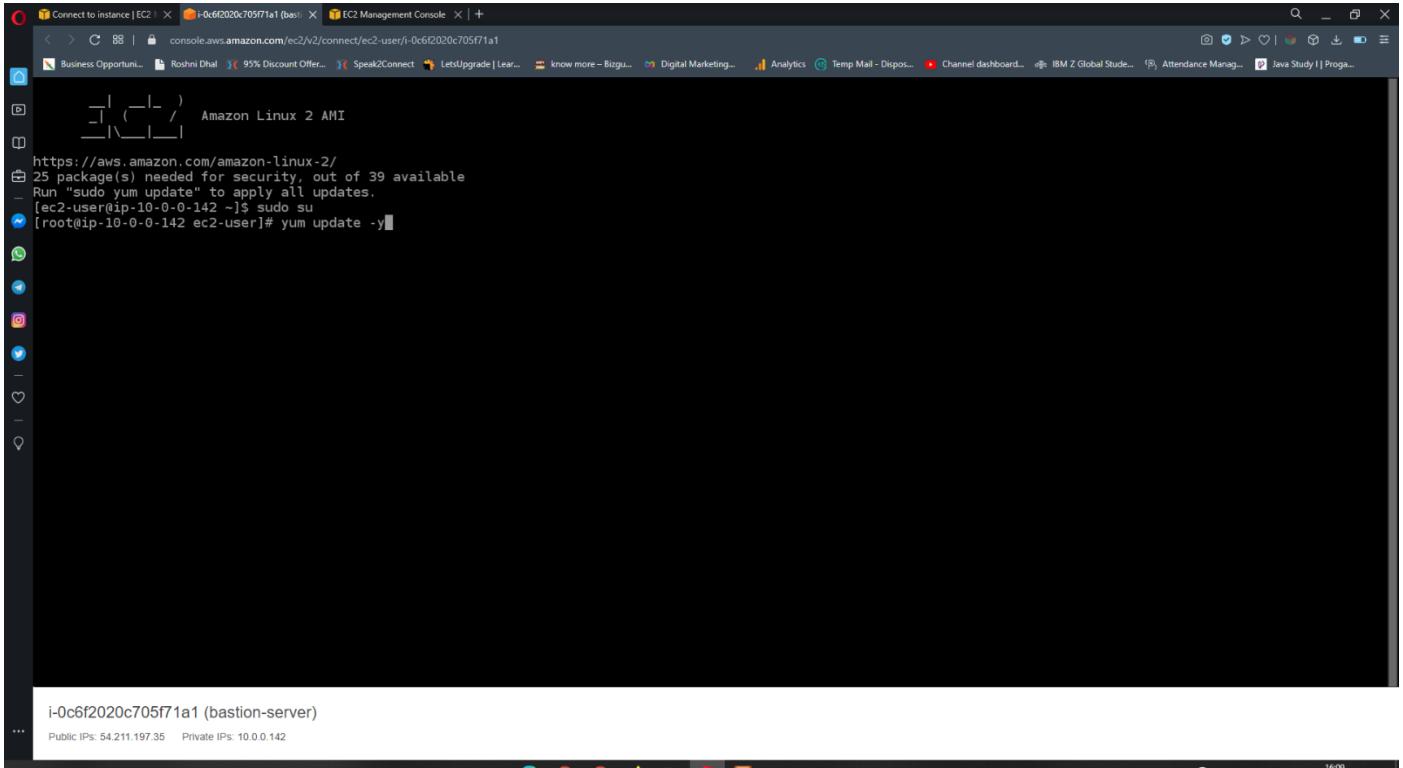
Load Balancer details:

The screenshot shows the AWS EC2 Management Console displaying the details of the newly created load balancer "web-loadbalancer". The basic configuration includes:

- Name:** web-loadbalancer
- ARN:** arn:aws:elasticloadbalancing:us-east-1:245024665952:loadbalancer/app/web-loadbalancer/f6e247e78d362c16
- DNS name:** web-loadbalancer-1445974582.us-east-1.elb.amazonaws.com (A Record)
- Type:** application
- Scheme:** internet-facing
- IP address type:** ipv4

Under the **Network & Security** section, the VPC is set to "vpc-0ef0246c86bdff45e" and the Availability Zones are "subnet-03023c158e58a279 - us-east-1b" and "subnet-0325668c16a4db095 - us-east-1a".

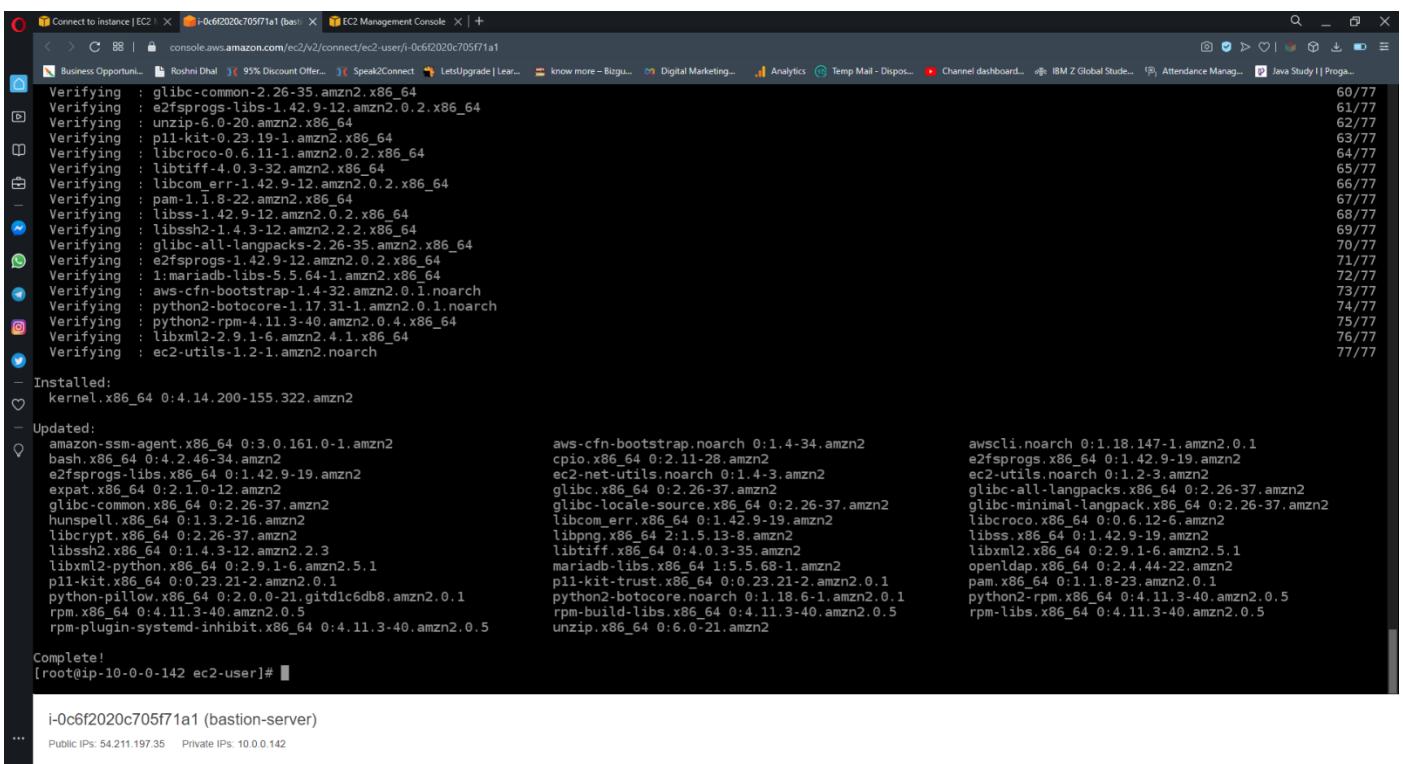
SSH into Public and Private EC2 Instance and Test Internet Connectivity:



```
https://aws.amazon.com/amazon-linux-2/
25 package(s) needed for security, out of 39 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-10-0-0-142 ~]$ sudo su
[root@ip-10-0-0-142 ec2-user]# yum update -y
```

i-0c6f2020c705f71a1 (bastion-server)
Public IPs: 54.211.197.35 Private IPs: 10.0.0.142

Updating linux:



```
Verifying : glibc-common-2.26-35.amzn2.x86_64
Verifying : e2fsprogs-libs-1.42.9-12.amzn2.0.2.x86_64
Verifying : unzip-6.0-20.amzn2.x86_64
Verifying : p11-kit-0.23.19-1.amzn2.x86_64
Verifying : libcroco-0.6.11-1.amzn2.0.2.x86_64
Verifying : libtiff-4.0.3-32.amzn2.x86_64
Verifying : libcom_err-1.42.9-12.amzn2.0.2.x86_64
Verifying : pam-1.1.8-22.amzn2.x86_64
Verifying : libss-1.42.9-12.amzn2.0.2.x86_64
Verifying : libssh2-1.4.3-12.amzn2.2.x86_64
Verifying : glibc-all-langpacks-2.26-35.amzn2.x86_64
Verifying : e2fsprogs-1.42.9-12.amzn2.0.2.x86_64
Verifying : mariadb-libs-5.5.64-1.amzn2.x86_64
Verifying : aws-cfn-bootstrap-1.4-32.amzn2.0.1.noarch
Verifying : python2-botocore-1.17.31-1.amzn2.0.1.noarch
Verifying : python2-rpm-4.11.3-40.amzn2.0.4.x86_64
Verifying : libxml2-2.9.1-6.amzn2.4.1.x86_64
Verifying : ec2-utils-1.2-1.amzn2.noarch
-
Kernel: kernel.x86_64 0:4.14.200-155.322.amzn2
-
Updated:
amazon-ssm-agent.x86_64 0:3.0.161.0-1.amzn2
bash.x86_64 0:4.2.46-34.amzn2
e2fsprogs-libs.x86_64 0:1.42.9-19.amzn2
expat.x86_64 0:2.1.0-12.amzn2
glibc-common.x86_64 0:2.26-37.amzn2
hunspell.x86_64 0:1.3.2-16.amzn2
libcrypt.x86_64 0:2.26-37.amzn2
libssh2.x86_64 0:1.4.3-12.amzn2.2.3
libxml2-python.x86_64 0:2.9.1-6.amzn2.5.1
p11-kit.x86_64 0:0.23.21-2.amzn2.0.1
python-pillow.x86_64 0:2.0.0-21.gitd1c6db8.amzn2.0.1
rpm.x86_64 0:4.11.3-40.amzn2.0.5
rpm-plugin-systemd-inhibit.x86_64 0:4.11.3-40.amzn2.0.5
-
Complete!
[root@ip-10-0-0-142 ec2-user]#
```

i-0c6f2020c705f71a1 (bastion-server)
Public IPs: 54.211.197.35 Private IPs: 10.0.0.142

Installing HTTPD:

Connect to instance | EC2 | i-0c6f2020c705f71a1 (bast) | EC2 Management Console | +

console.aws.amazon.com/ec2/v2/connect/ec2-user/i-0c6f2020c705f71a1

Business Opportunity... Roshni Dhal 95% Discount Offer... Speak2Connect Let's Upgrade! Learn... know more - Bizgu... Digital Marketing... Analytics Temp Mail - Dispos... Channel dashboard... IBM Z Global Studie... Attendance Manag... Java Study | Progra...

Verifying : glibc-common-2.26-35.amzn2.x86_64 60/77
Verifying : e2fsprogs-libs-1.42.9-12.amzn2.0.2.x86_64 61/77
Verifying : unzip-6.0-20.amzn2.x86_64 62/77
Verifying : p11-kit-0.23.19-1.amzn2.x86_64 63/77
Verifying : libcroco-0.6.11-1.amzn2.0.2.x86_64 64/77
Verifying : libtiff-4.0.3-32.amzn2.x86_64 65/77
Verifying : libcom_err-1.42.9-12.amzn2.0.2.x86_64 66/77
Verifying : pam-1.1.8-22.amzn2.x86_64 67/77
Verifying : libss-1.42.9-12.amzn2.0.2.x86_64 68/77
Verifying : libssh2-1.4.3-12.amzn2.2.2.x86_64 69/77
Verifying : glibc-all-langpacks-2.26-55.amzn2.x86_64 70/77
Verifying : e2fsprogs-1.42.9-12.amzn2.0.2.x86_64 71/77
Verifying : 1:mariadb-libs-5.5.64-1.amzn2.x86_64 72/77
Verifying : aws-cfn-bootstrap-noarch 1.4-32.amzn2.0.1.noarch 73/77
Verifying : python2-botocore-1.17.31-1.amzn2.0.1.noarch 74/77
Verifying : python2-rpm-4.11.3-40.amzn2.0.4.x86_64 75/77
Verifying : libxml2-2.9.1-6.amzn2.4.1.x86_64 76/77
Verifying : ec2-utils-1.2-1.amzn2.noarch 77/77

- Installed:
kernel.x86_64 0:4.14.200-155.322.amzn2

- Updated:
amazon-ssm-agent.x86_64 0:3.0.161.0-1.amzn2
bash.x86_64 0:4.2.46-34.amzn2
e2fsprogs-libs.x86_64 0:1.42.9-19.amzn2
expat.x86_64 0:2.1.0-12.amzn2
glibc-common.x86_64 0:2.26-37.amzn2
hunspell.x86_64 0:1.3.2-16.amzn2
libcrypt.x86_64 0:2.26-37.amzn2
libssh2.x86_64 0:1.4.3-12.amzn2.2.3
libxml2-python.x86_64 0:2.9.1-6.amzn2.5.1
p11-kit.x86_64 0:0.23.21-2.amzn2.0.1
python-pillow.x86_64 0:2.0.0-21.gidtclc6db8.amzn2.0.1
rpm.x86_64 0:4.11.3-40.amzn2.0.5
rpm-plugin-systemd-inhibit.x86_64 0:4.11.3-40.amzn2.0.5

aws-cfn-bootstrap.noarch 0:1.4-34.amzn2
cpio.x86_64 0:2.11-28.amzn2
ec2-net-utils.noarch 0:1.4-3.amzn2
glibc.x86_64 0:2.26-37.amzn2
glibc-locale-source.x86_64 0:2.26-37.amzn2
libcom_err.x86_64 0:1.42.9-19.amzn2
libpng.x86_64 2:1.5.13-8.amzn2
libtiff.x86_64 0:4.0.3-35.amzn2
mariadb-libs.x86_64 1:5.5.68-1.amzn2
p11-kit-trust.x86_64 0:0.23.21-2.amzn2.0.1
python2-botocore.noarch 0:1.18.6-1.amzn2.0.1
rpm-build-libs.x86_64 0:4.11.3-40.amzn2.0.5
unzip.x86_64 0:6.0-21.amzn2

awscli.noarch 0:1.18.147-1.amzn2.0.1
e2fsprogs.x86_64 0:1.42.9-19.amzn2
ec2-utils.noarch 0:1.2-3.amzn2
glibc-all-langpacks.x86_64 0:2.26-37.amzn2
glibc-minimal-langpack.x86_64 0:2.26-37.amzn2
libcroco.x86_64 0:0.6.12-6.amzn2
libss.x86_64 0:1.42.9-19.amzn2
libxml2.x86_64 0:2.9.1-6.amzn2.5.1
openldap.x86_64 0:2.4.44-22.amzn2
pam.x86_64 0:1.1.8-23.amzn2.0.1
python2-rpm.x86_64 0:4.11.3-40.amzn2.0.5
rpm-libs.x86_64 0:4.11.3-40.amzn2.0.5

Complete!
[root@ip-10-0-0-142 ec2-user]# yum install httpd -y

i-0c6f2020c705f71a1 (bastion-server)

Public IPs: 54.211.197.35 Private IPs: 10.0.0.142

Created file using “vi webserver.pem” command

And copy pasted pem file into it:

Changing permission of webserver.pem file:

```
Connect to instance | EC2 < x-0c6f2020c705f71a1 (be1) < x- EC2 Management Console < +  
https://console.aws.amazon.com/ec2/v2/connect/ec2-user/x-0c6f2020c705f71a1  
Business Opportunities... Rohini Dhal... 95% Discount Offer... Speak Connect... LetsUpgrade | Lear... know more - Bizgu... Digital Marketing... Analytics Temp Mail - Dispos... Channel dashboard... IBM Z Global Studie... Attendance Manag... Java Study | Progra...  
0qqkjt2061tTJvzn1RDKmnbkW+DhHhnz+rsm7zcj9WsPjTct4tbVbW7VKnRhb/7H7pG6ap91mDkmc4zJh10sKbdAubGrLECYIkq9KXyaaoyl1NbS3+dGDLXAnP3NPe59xUfLASHw1svrbX2o1t9MB2F2Xse2aX9qZt1PU7sgNqSHxGw1Zcgig16ZHESnRhe3evgQoHFy1A4DsBjXwCPPtL8Zryggp0m6VvGgBefqwgNlsUUxhokpDjTN8oJ27YzmjCR61FP6qz5kqjZn9LjPeqIDoAQBaoIBAA8mqdG7tkyG/blyqSsINwujXj8/U6Ag+d0907nz146MnChal9w+yBSjWkudCzbAbt0Zpg0ok5KJ00TbCmukwLszqmAvzRjua1v5qA1vnu22JwL4EmIShMeJ6Kj/0iUBRT0ywbk1EPa1Z2Zq+jt1ThwFBonArSk1NbkbX6EtDlNPq33sVz14/77s4pbtlrdiMTQSIfBpRz+ut3/6A9IXAUvurcC2NPpeDfxoXevgrkwB34f48g2zBFtRF3R7xLushomQgdvz+mKquV2Gezevc+lXcLcJrIh4W1vHscnChluoF5vg3PpE/5N9jGFdk9hX/kPAMEoKDSYEcgYEAv9xSkeRBnlsoWov0a4zIuInvh24v41Wm+Dsnldgtj;YHxQYvdLnTf1876Rtm59xdLyhW17PR1xobyoFL19JLLkxxsUsv97bwfSyP8/q84Gcf0xe52yjC3Xvgvv2ydl0ahpB310gvLj1n60Imog7MaXu5w7gMARH2kbtl53uekCgYEaubCr5k4WidHP9utIkL5N/6u6hP0j8zr4G0n8faar+Pgr57eBo1VCCwEvSR2hAV8vH1s8qrjHntV9/wOx9-Hg+1g+1v/5HlQ90wlc1KrNhDzsDav8DNu+Clz5EY2htLLzCzW07eXzbayhGGBGAs5BUkzJB1YkvPDRsEcgyAMVuug154dNb0oLChja3t98I0LQnjdgNrAjC0zg/nScypDw2FuYd/jqJYe6th0ElZBz+Lhe50uzTMiq/E4jnykExRgrt2NINh25550NeWHD2w5KmxvLLchlsZj;Ipu08E8g75fk7y81/yVdgwsmplxf5v0Cdcy0Gi_mzq20K8bg0CzU6DQp+z0Yz0xg1riaNz9woow0cYdf1PEjh-G9YH0x0+HlmpqJ1jCNfy1mZ6wxdGcmLa3G6hrvn3XFCf0KMfDm4u5J+j+YDxxXawds6sZUeduvGgv1ShgbhffHAWyN9RwgvN0KhwyEt7+PxVs5021jNF3vAkiVdbrvt4vgKBgGyVNgpizrVKku1jy1v1x8dvf4wM5Y9D6/GH6GN+zyTX1mtzA0yrUO-3l1srF0s0KwP0l7e1h0wtrhPPy34q2pw3c2LUUzuZ9nh/i1Qh0Wnc8w3YTR3C4hVbK18wH5Wxt1Rviobtp/m04/8b5D0oMB7GCFA4YaCYARe5yqUBK-----END RSA PRIVATE KEY-----  
"webserver.pem" [New] 27L, 1675C written  
[root@ip-10-0-0-142 ec2-user]# chmod 400 webserver.pem  
[root@ip-10-0-0-142 ec2-user]#
```

SSH into to private subnet using public subnet:

Instances | EC2 Management | i-0c6f2020c705f71a1 (basti...) | EC2 Management Console | +

console.aws.amazon.com/ec2/v2/connect/ec2-user/i-0c6f2020c705f71a1

Business Opportuni... Roshni Dhal | 95% Discount Offer... | Speak2Connect | LetsUpgrade | Lear... | know more - Bizgu... | Digital Marketing... | Analytics | Temp Mail - Dispos... | Channel dashboard... | IBM Z Global Studie... | Attendance Manag... | Java Study | Progra...

qrYJNjT9vJ0wX9H+Hg+1Ge1v/5HLq9W0lcr1KnhHDzDAv8DNu+8CLzb5EY2HtL
4zcZoW7eXzbayYhGGBA5sBUkIzJB1pYXvPDRsEcgYAMVug154dNb0LcMja3tB
[i] 10LQnjDgNrA1cOrZg/nScyp3gOfU2YqJyvYe6tboEIZZBe+LheF50uzTMq/E4
jnykExRgr2W2INlx25550NEwHD25kMxvllchMsZzJIpou08E8g75fk7y8tV/yd
wsmplxf5vOcdyoGimzq20KBgOCz9U6D0p+z0Yz0xg1riahZ9vocgw0c:Ydf1PEjh
G9hqqzy+5Y0x0+LmqpJL1CnfyInz6wxGcm1a3Ghhrvn3FCf0KMfDm4u5J+jYDxx
xawds6s7UeDuUVggvISgbhffAWyN9RwgVn0KhyyEt7+PxSv5021NF3VaAkiVdbr
vt4vg0KBgQgVlpizRVKkuJy1xvdvfa4w5M5Y9DG/6Gh6GN+zYTx1ltz1a0rU0
3L1s1Fs0kWP0lve7i1h0vrriPPry3Hq2pV3c2UZuZ9nh/i0h0Wncl8sYTr3C4h
VbkI8wH5WxtLRw1oBtp/m014/8b50QmB7GCF44YaCYARe5yqUBK
-----END RSA PRIVATE KEY-----

"webserver.pem" [New] 27L, 1675C written

[root@ip-10-0-0-142 ec2-user]# chmod 400 webserver.pem

[root@ip-10-0-0-142 ec2-user]# ssh -i webserver.pem ec2-user@10.0.1.209
The authenticity of host '10.0.1.209' (10.0.1.209) can't be established.
ECDSA key fingerprint is SHA256:XHw2mAp1LZRes0795CT74DXXLHM/dIOsYb1ck1JL+0.
ECDSA key fingerprint is MD5:5:c6:92:2b:b3:42:ea:07:50:c4:90:48:97:9b:cf:ed.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.1.209' (ECDSA) to the list of known hosts.

Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
25 package(s) needed for security, out of 39 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-10-0-1-209 ~]\$ █

Updated to linux in webser 1:

```
Instances | EC2 Management | i-0c6f2020c705f71a1 (bast) | EC2 Management Console | + | https://console.aws.amazon.com/ec2/v2/connect/ec2-user/i-0c6f2020c705f71a1
Business Opportunities | Roshni Dhal | 95% Discount Offer... | Speak2Connect | Let'sUpgrade | Learn... | know more - Bizgu... | Digital Marketing... | Analytics | Temp Mail - Dispos... | Channel dashboard... | IBM Z Global Studie... | Attendance Manag... | Java Study | Progra...
Verifying : glibc-common-2.26-35.amzn2.x86_64
Verifying : e2fsprogs-libs-1.42.9-12.amzn2.0.2.x86_64
Verifying : unzip-6.0-20.amzn2.x86_64
Verifying : p11-kit-0.23.19-1.amzn2.x86_64
Verifying : libcroco-0.6.11-1.amzn2.0.2.x86_64
Verifying : libtiff-4.0.3-32.amzn2.x86_64
Verifying : libcom_err-1.42.9-12.amzn2.0.2.x86_64
Verifying : pam-1.8-22.amzn2.x86_64
Verifying : libss-1.42.9-12.amzn2.0.2.x86_64
Verifying : libssh2-1.4.3-12.amzn2.2.2.x86_64
Verifying : glibc-all-langpacks-2.26-35.amzn2.x86_64
Verifying : e2fsprogs-1.42.9-12.amzn2.0.2.x86_64
Verifying : mariadb-libs-5.5.64-1.amzn2.x86_64
Verifying : aws-cfn-bootstrap-1.4-32.amzn2.0.1.noarch
Verifying : python2-botocore-1.17.31-1.amzn2.0.1.noarch
Verifying : python2-rpm-4.11.3-40.amzn2.0.4.x86_64
Verifying : libxml2-2.9.1-6.amzn2.4.1.x86_64
Verifying : ec2-utils-1.2-1.amzn2.noarch
- Installed:
kernel.x86_64 0:4.14.200-155.322.amzn2
- Updated:
amazon-ssm-agent.x86_64 0:3.0.161.0-1.amzn2
bash.x86_64 0:4.2.46-34.amzn2
e2fsprogs-libs.x86_64 0:1.42.9-19.amzn2
expat.x86_64 0:2.1.0-12.amzn2
glibc-common.x86_64 0:2.26-37.amzn2
hunspell.x86_64 0:1.3.2-16.amzn2
libcrypt.x86_64 0:2.26-37.amzn2
libssh2.x86_64 0:1.4.3-12.amzn2.2.3
libxml2-python.x86_64 0:2.9.1-6.amzn2.5.1
p11-kit.x86_64 0:0.23.21-2.amzn2.0.1
python-pillow.x86_64 0:2.0.0-21.gitd1c6db8.amzn2.0.1
rpm.x86_64 0:4.11.3-40.amzn2.0.5
rpm-plugin-systemd-inhibit.x86_64 0:4.11.3-40.amzn2.0.5
unzip.x86_64 0:6.0-21.amzn2
awscli.noarch 0:1.18.147-1.amzn2.0.1
e2fsprogs.x86_64 0:1.42.9-19.amzn2
ec2-utils.noarch 0:1.2-3.amzn2
glibc-all-langpacks.x86_64 0:2.26-37.amzn2
glibc-minimal-langpack.x86_64 0:2.26-37.amzn2
libcroco.x86_64 0:0.6.12-6.amzn2
libss.x86_64 0:1.42.9-19.amzn2
libxml2.x86_64 0:2.9.1-6.amzn2.5.1
openldap.x86_64 0:2.4.44-22.amzn2
pam.x86_64 0:1.1.8-23.amzn2.0.1
python2-rpm.x86_64 0:4.11.3-40.amzn2.0.5
rpm-libs.x86_64 0:4.11.3-40.amzn2.0.5
Complete!
[root@ip-10-0-1-209 ec2-user]#
```

i-0c6f2020c705f71a1 (bastion-server)

Public IPs: 54.211.197.35 Private IPs: 10.0.0.142

Installing HTTPD in web-server 1:

```
Instances | EC2 Management | i-0c6f2020c705f71a1 (bast) | EC2 Management Console | + | https://console.aws.amazon.com/ec2/v2/connect/ec2-user/i-0c6f2020c705f71a1
Business Opportunities | Roshni Dhal | 95% Discount Offer... | Speak2Connect | Let'sUpgrade | Learn... | know more - Bizgu... | Digital Marketing... | Analytics | Temp Mail - Dispos... | Channel dashboard... | IBM Z Global Studie... | Attendance Manag... | Java Study | Progra...
Verifying : glibc-common-2.26-35.amzn2.x86_64
Verifying : e2fsprogs-libs-1.42.9-12.amzn2.0.2.x86_64
Verifying : unzip-6.0-20.amzn2.x86_64
Verifying : p11-kit-0.23.19-1.amzn2.x86_64
Verifying : libcroco-0.6.11-1.amzn2.0.2.x86_64
Verifying : libtiff-4.0.3-32.amzn2.x86_64
Verifying : libcom_err-1.42.9-12.amzn2.0.2.x86_64
Verifying : pam-1.8-22.amzn2.x86_64
Verifying : libss-1.42.9-12.amzn2.0.2.x86_64
Verifying : libssh2-1.4.3-12.amzn2.2.2.x86_64
Verifying : glibc-all-langpacks-2.26-35.amzn2.x86_64
Verifying : e2fsprogs-1.42.9-12.amzn2.0.2.x86_64
Verifying : mariadb-libs-5.5.64-1.amzn2.x86_64
Verifying : aws-cfn-bootstrap-1.4-32.amzn2.0.1.noarch
Verifying : python2-botocore-1.17.31-1.amzn2.0.1.noarch
Verifying : python2-rpm-4.11.3-40.amzn2.0.4.x86_64
Verifying : libxml2-2.9.1-6.amzn2.4.1.x86_64
Verifying : ec2-utils-1.2-1.amzn2.noarch
- Installed:
kernel.x86_64 0:4.14.200-155.322.amzn2
- Updated:
amazon-ssm-agent.x86_64 0:3.0.161.0-1.amzn2
bash.x86_64 0:4.2.46-34.amzn2
e2fsprogs-libs.x86_64 0:1.42.9-19.amzn2
expat.x86_64 0:2.1.0-12.amzn2
glibc-common.x86_64 0:2.26-37.amzn2
hunspell.x86_64 0:1.3.2-16.amzn2
libcrypt.x86_64 0:2.26-37.amzn2
libssh2.x86_64 0:1.4.3-12.amzn2.2.3
libxml2-python.x86_64 0:2.9.1-6.amzn2.5.1
p11-kit.x86_64 0:0.23.21-2.amzn2.0.1
python-pillow.x86_64 0:2.0.0-21.gitd1c6db8.amzn2.0.1
rpm.x86_64 0:4.11.3-40.amzn2.0.5
rpm-plugin-systemd-inhibit.x86_64 0:4.11.3-40.amzn2.0.5
unzip.x86_64 0:6.0-21.amzn2
awscli.noarch 0:1.18.147-1.amzn2.0.1
e2fsprogs.x86_64 0:1.42.9-19.amzn2
ec2-utils.noarch 0:1.2-3.amzn2
glibc-all-langpacks.x86_64 0:2.26-37.amzn2
glibc-minimal-langpack.x86_64 0:2.26-37.amzn2
libcroco.x86_64 0:0.6.12-6.amzn2
libss.x86_64 0:1.42.9-19.amzn2
libxml2.x86_64 0:2.9.1-6.amzn2.5.1
openldap.x86_64 0:2.4.44-22.amzn2
pam.x86_64 0:1.1.8-23.amzn2.0.1
python2-rpm.x86_64 0:4.11.3-40.amzn2.0.5
rpm-libs.x86_64 0:4.11.3-40.amzn2.0.5
Complete!
[root@ip-10-0-1-209 ec2-user]# yum install httpd
```

i-0c6f2020c705f71a1 (bastion-server)

Public IPs: 54.211.197.35 Private IPs: 10.0.0.142

Creating “index.html” file in webserver 1:

The screenshot shows the AWS EC2 Management Console interface. In the terminal window, the user is navigating to the /var/www/html directory and creating an index.html file. The file is created with a size of 38C. The user then starts the httpd service and checks its status, which shows it is active (running). The Apache logs at the bottom indicate the server has started.

```
i-0c6f2020c705f71a1 (bastion-server)
...
Public IPs: 54.211.197.35 Private IPs: 10.0.0.142

[Complete!
[root@ip-10-0-1-209 ec2-user]# cd /var/www/html
[root@ip-10-0-1-209 html]# ls
[root@ip-10-0-1-209 html]# vi index.html

i-0c6f2020c705f71a1 (bastion-server)
...
Public IPs: 54.211.197.35 Private IPs: 10.0.0.142]
```

Started httpd service and checking status in web server 1:

The screenshot shows the AWS EC2 Management Console interface. The user has run the command 'service httpd status' to check the status of the httpd service. The output shows the service is active (running). The Apache logs at the bottom show the server starting and accepting requests.

```
"index.html" [New] 1L, 38C written
[root@ip-10-0-1-209 html]# service httpd start
- Redirecting to /bin/systemctl start httpd.service
[root@ip-10-0-1-209 html]# service httpd status
Redirecting to /bin/systemctl status httpd.service
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)
   Active: active (running) since Sun 2020-11-01 10:49:54 UTC; 14s ago
     Docs: man:httpd.service(8)
   Main PID: 12556 (httpd)
  Status: "Total requests: 0; Idle/Busy workers 100/0; Requests/sec: 0; Bytes served/sec: 0 B/sec"
  CGroup: /system.slice/httpd.service
          └─12556 /usr/sbin/httpd -DFOREGROUND
          ├─12557 /usr/sbin/httpd -DFOREGROUND
          ├─12558 /usr/sbin/httpd -DFOREGROUND
          ├─12559 /usr/sbin/httpd -DFOREGROUND
          ├─12560 /usr/sbin/httpd -DFOREGROUND
          └─12561 /usr/sbin/httpd -DFOREGROUND

Nov 01 10:49:54 ip-10-0-1-209.ec2.internal systemd[1]: Starting The Apache HTTP Server...
Nov 01 10:49:54 ip-10-0-1-209.ec2.internal systemd[1]: Started The Apache HTTP Server.
[root@ip-10-0-1-209 html]#
```

i-0c6f2020c705f71a1 (bastion-server)
...
Public IPs: 54.211.197.35 Private IPs: 10.0.0.142

SSH into web server 2:

The screenshot shows the AWS EC2 Management Console interface. A terminal window is open, showing the command-line session. The user has run 'ssh -i webserver.pem ec2-user@10.0.1.228' and is prompted to connect to host '10.0.1.228'. They accept the connection and are then prompted to update their known hosts. The user runs 'sudo yum update' and is informed that 25 packages are needed for security, out of 39 available. Finally, they run 'ls' to list the contents of the current directory.

```
Last login: Sun Nov 1 10:42:29 2020 from ec2-18-206-107-24.compute-1.amazonaws.com
[ec2-user@ip-10-0-0-142 ~]$ sudo su
[root@ip-10-0-0-142 ec2-user]# ssh -i webserver.pem ec2-user@10.0.1.228
The authenticity of host '10.0.1.228 (10.0.1.228)' can't be established.
ECDSA key fingerprint is SHA256:ZCW01LW/mTqWUkAig9bmfxYiJgkuPKcIMFzSWGieus.
ECDSA key fingerprint is MD5:fb:d7:77:7d:51:4a:3d:e4:7b:21:22:11:50:4f:56:00.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.1.228' (ECDSA) to the list of known hosts.

[ec2-user@ip-10-0-0-142 ~]$ ls
[ec2-user@ip-10-0-0-142 ~]$ 25 package(s) needed for security, out of 39 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-10-0-1-228 ~]$
```

i-0c6f2020c705f71a1 (bastion-server)

Public IPs: 54.211.197.35 Private IPs: 10.0.0.142

Updating the web server 2:

The screenshot shows the AWS EC2 Management Console interface. A terminal window is open, showing the command-line session. The user runs 'sudo yum update' and the output shows the progress of the update process. The update installs the kernel and multiple packages, including glibc-common, libcom_err, libssh2, libcrypt, libxml2, python2, pam, libtiff, mariadb-libs, aws-cfn-bootstrap, and others. The update process is completed successfully.

```
Verifying : glibc-common-2.26-35.amzn2.x86_64
Verifying : e2fsprogs-libs-1.42.9-12.amzn2.0.2.x86_64
Verifying : unzip-6.0-20.amzn2.x86_64
Verifying : p11-kit-0.23.19-1.amzn2.x86_64
Verifying : libcroco-0.6.11-1.amzn2.0.2.x86_64
Verifying : libtiff-4.0.3-32.amzn2.x86_64
Verifying : libcom_err-1.42.9-12.amzn2.0.2.x86_64
Verifying : pam-1.1.8-22.amzn2.x86_64
Verifying : libss-1.42.9-12.amzn2.0.2.x86_64
Verifying : libssh2-1.4.3-12.amzn2.0.2.x86_64
Verifying : glibc-all-langpacks-2.26-35.amzn2.x86_64
Verifying : e2fsprogs-1.42.9-12.amzn2.0.2.x86_64
Verifying : mariadb-libs-5.5.64-1.amzn2.x86_64
Verifying : aws-cfn-bootstrap-1.4-32.amzn2.0.1.noarch
Verifying : python2-botocore-1.17.31-1.amzn2.0.1.noarch
Verifying : python2-rpm-4.11.3-40.amzn2.0.4.x86_64
Verifying : libxml2-2.9.1-6.amzn2.4.1.x86_64
Verifying : ec2-utils-1.2-1.amzn2.noarch

Installed:
  kernel.x86_64 0:4.14.200-155.322.amzn2

Updated:
  amazon-ssm-agent.x86_64 0:3.0.161.0-1.amzn2
  bash.x86_64 0:4.2.46-34.amzn2
  e2fsprogs-libs.x86_64 0:2.29.19.amzn2
  expat.x86_64 0:2.1.0-12.amzn2
  glibc-common.x86_64 0:2.26-37.amzn2
  hunspell.x86_64 0:1.3.2-16.amzn2
  libcrypt.x86_64 0:2.26-37.amzn2
  libssh2.x86_64 0:1.4.3-12.amzn2.2.3
  libxml2-python.x86_64 0:2.9.1-6.amzn2.5.1
  p11-kit.x86_64 0:0.23.21-2.amzn2.0.1
  python2-pillow.x86_64 0:2.0.0-21.gitd1c6db8.amzn2.0.1
  rpm.x86_64 0:4.11.3-40.amzn2.0.5
  rpm-plugin-systemd-inhibit.x86_64 0:4.11.3-40.amzn2.0.5

Complete!
[ec2-user@ip-10-0-1-228 ~]$
```

i-0c6f2020c705f71a1 (bastion-server)

Public IPs: 54.211.197.35 Private IPs: 10.0.0.142

Installing HTTPD in web server 2:

```
Instances | EC2 Manager... × 0-06f2020c705f71a1 (bast... × EC2 Management Console × +  
console.aws.amazon.com/ec2/v2/connect/ec2-user/i-0c6f2020c705f71a1  
Business Opportuni... Roshni Dhal 95% Discount Offer... Speak2Connect Let'sUpgrade | Lear... know more - Biz... Digital Marketing... Analytics Temp Mail - Dispos... Channel dashboard... IBM Z Global Studie... Attendance Manag... Java Study I | Progra...  
(5/9): httpd-filesystem-2.4.46-1.amzn2.noarch.rpm | 23 kB 00:00:00  
(6/9): httpd-2.4.46-1.amzn2.x86_64.rpm | 1.3 kB 00:00:00  
(7/9): httpd-tools-2.4.46-1.amzn2.x86_64.rpm | 87 kB 00:00:00  
(8/9): mailcap-2.1.41-2.amzn2.noarch.rpm | 31 kB 00:00:00  
(9/9): mod_http2-1.15.14-2.amzn2.x86_64.rpm | 147 kB 00:00:00  
-----  
Total 9.2 MB/s | 1.8 MB 00:00:00  
- Running transaction check  
- Running transaction test  
Transaction test succeeded  
Running transaction  
  Installing : apr-1.6.3-5.amzn2.0.2.x86_64 2/9  
  Installing : apr-util-bdb-1.6.1-5.amzn2.0.2.x86_64 3/9  
  Installing : apr-util-1.6.1-5.amzn2.0.2.x86_64 4/9  
  Installing : httpd-tools-2.4.46-1.amzn2.x86_64 5/9  
  Installing : generic-logos-httpd-18.0.0-4.amzn2.noarch 6/9  
  Installing : mailcap-2.1.41-2.amzn2.noarch 7/9  
  Installing : httpd-filesystem-2.4.46-1.amzn2.noarch 8/9  
  Installing : mod_http2-1.15.14-2.amzn2.x86_64 9/9  
  Installing : httpd-2.4.46-1.amzn2.x86_64 1/9  
  Verifying : apr-util-1.6.1-5.amzn2.0.2.x86_64 2/9  
  Verifying : httpd-filesystem-2.4.46-1.amzn2.noarch 3/9  
  Verifying : apr-util-bdb-1.6.1-5.amzn2.0.2.x86_64 4/9  
  Verifying : httpd-tools-2.4.46-1.amzn2.x86_64 5/9  
  Verifying : mod_http2-1.15.14-2.amzn2.x86_64 6/9  
  Verifying : apr-1.6.3-5.amzn2.0.2.x86_64 7/9  
  Verifying : mailcap-2.1.41-2.amzn2.noarch 8/9  
  Verifying : generic-logos-httpd-18.0.0-4.amzn2.noarch 9/9  
  Verifying : httpd-2.4.46-1.amzn2.x86_64  
  
Installed:  
  httpd.x86_64 0:2.4.46-1.amzn2  
  
Dependency Installed:  
  apr.x86_64 0:1.6.3-5.amzn2.0.2  apr-util.x86_64 0:1.6.1-5.amzn2.0.2  apr-util-bdb.x86_64 0:1.6.1-5.amzn2.0.2  generic-logos-httpd.noarch 0:18.0.0-4.amzn2  
  httpd-filesystem.noarch 0:2.4.46-1.amzn2  httpd-tools.x86_64 0:2.4.46-1.amzn2  mailcap.noarch 0:2.1.41-2.amzn2  mod_http2.x86_64 0:1.15.14-2.amzn2  
  
Complete!  
[root@ip-10-0-1-228 ec2-user]#
```

Creating “index.html” file in webserver 2:

```
"index.html" 1L, 36C written
[root@ip-10-0-1-228 html]# service
```

Message when web-server 2 is accessed:

```
"index.html" 1L, 36C written
[root@ip-10-0-1-228 html]# more index.html
" REQUEST HANDLING BY SERVER 2"
[root@ip-10-0-1-228 html]#
```

i-0c6f2020c705f71a1 (bastion-server)
Public IPs: 54.211.197.35 Private IPs: 10.0.0.142

Started httpd service and checked status in web-server 2:

Finally exited from sudo user and also log out from webserver 2

Using exit command.

```
"index.html" 1L, 36C written
[root@ip-10-0-1-228 html]# more index.html
" REQUEST HANDLING BY SERVER 2"
[root@ip-10-0-1-228 html]# service httpd start
Redirecting to /bin/systemctl start httpd.service
[root@ip-10-0-1-228 html]# service httpd status
Redirecting to /bin/systemctl status httpd.service
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)
   Active: active (running) since Sun 2020-11-01 10:57:19 UTC; 13s ago
     Docs: man:httpd.service(8)
Main PID: 12581 (httpd)
   Status: "Total requests: 1; Idle/Busy workers 100/0;Requests/sec: 0.111; Bytes served/sec: 52 B/sec"
   CGroup: /system.slice/httpd.service
           └─12581 /usr/sbin/httpd -DFOREGROUND
           ├─12582 /usr/sbin/httpd -DFOREGROUND
           ├─12583 /usr/sbin/httpd -DFOREGROUND
           ├─12584 /usr/sbin/httpd -DFOREGROUND
           ├─12585 /usr/sbin/httpd -DFOREGROUND
           └─12586 /usr/sbin/httpd -DFOREGROUND

Nov 01 10:57:19 ip-10-0-1-228.ec2.internal systemd[1]: Starting The Apache HTTP Server...
Nov 01 10:57:19 ip-10-0-1-228.ec2.internal systemd[1]: Started The Apache HTTP Server.
[root@ip-10-0-1-228 html]# exit
exit
[ec2-user@ip-10-0-1-228 ~]$ exit
logout
Connection to 10.0.1.228 closed.
[root@ip-10-0-0-142 ec2-user]#
```

i-0c6f2020c705f71a1 (bastion-server)
Public IPs: 54.211.197.35 Private IPs: 10.0.0.142

Load-balancer Details:

The screenshot shows the AWS EC2 Management Console with the 'Load Balancing' section selected. On the left, there's a sidebar with various AWS services like Instances, Images, and Network & Security. The main area shows a table of existing load balancers:

Name	DNS name	State	VPC ID	Availability Zones	Type	Created At
web-loadbalancer	web-loadbalancer-14459745...	active	vpc-0ef0246c86bdff45e	us-east-1b, us-east-1a	application	November 1, 2020 at 4:06:5...

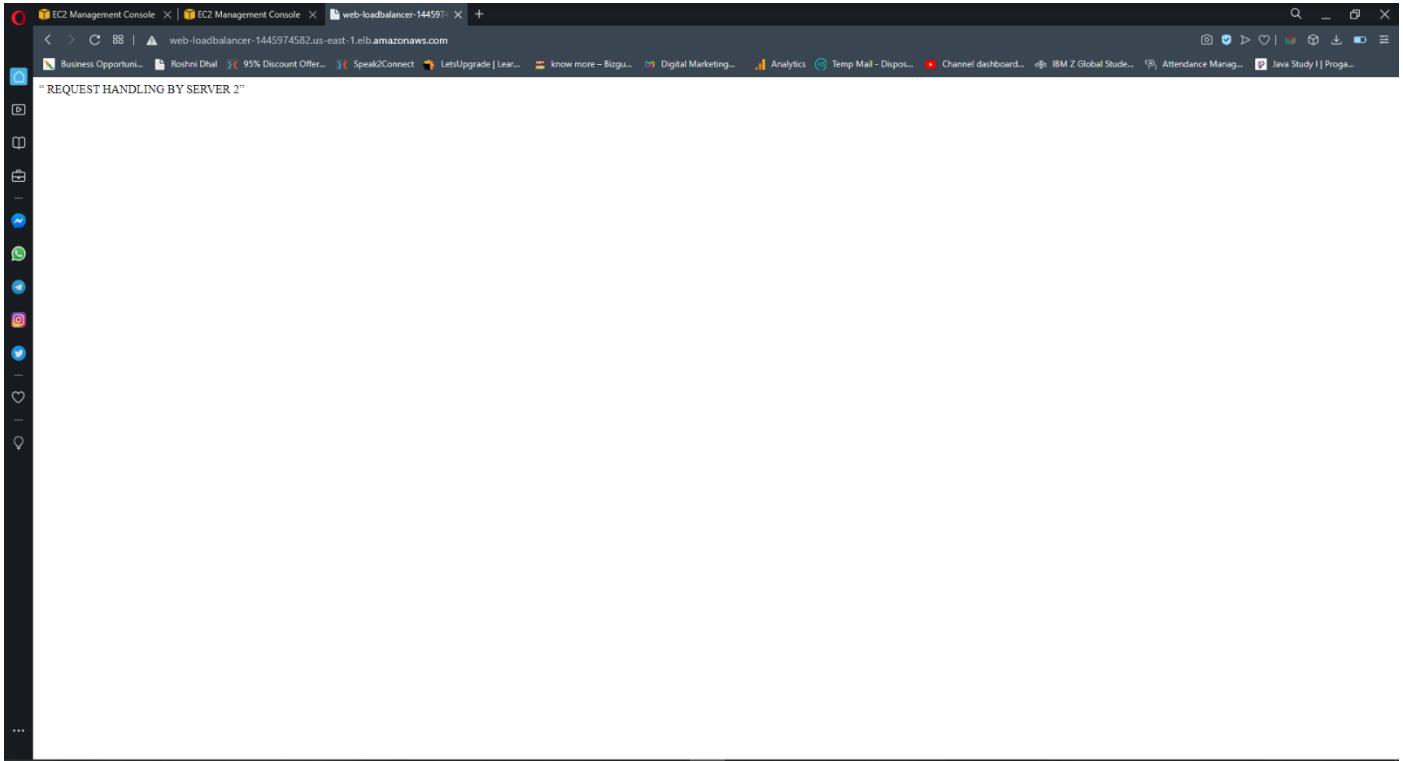
Below the table, a detailed view for the 'web-loadbalancer' is shown under the 'Basic Configuration' tab. The configuration includes:

- Name:** web-loadbalancer
- ARN:** arn:aws:elasticloadbalancing:us-east-1:245024665952:loadbalancer/app/web-loadbalancer/f5e247e78d362c16
- DNS name:** web-loadbalancer-1445974582.us-east-1.elb.amazonaws.com (A Record)
- State:** active
- Type:** application
- Scheme:** internet-facing
- IP address type:** ipv4
- VPC:** vpc-0ef0246c86bdff45e
- Availability Zones:** subnet-03023c158e58a279 - us-east-1b (IPv4 address: Assigned by AWS)
subnet-0325668c16a4db095 - us-east-1a (IPv4 address: Assigned by AWS)

Request from web server 1 when copy pasted DNS of Load balancer:

The screenshot shows a browser window with the URL <http://web-loadbalancer-1445974582.us-east-1.elb.amazonaws.com>. The page content is "REQUEST HANDLING BY SERVER 1".

Request from web server 2 when copy pasted DNS of Load balancer:



Health of web-server before being stopped:

A screenshot of the AWS Target groups console. The left sidebar shows navigation options like 'Target groups', 'EC2', 'AWS Lambda', and 'CloudWatch Metrics'. The main panel shows a target group named 'new-target-1'. Under 'Basic configuration', it lists 'Target type: instance', 'Protocol: Port HTTP : 80', 'Protocol version: HTTP1', 'VPC: vpc-0ef0246c86bdff45e', and 'Load balancer: web-loadbalancer'. Below this, the 'Targets' tab is selected, showing two registered targets: 'i-07ecb048677d8ff6e' (web-server-2) and 'i-0302839db79ec6175' (web-server-1), both marked as 'healthy'. There is also a 'Monitoring' tab and a 'Tags' tab.

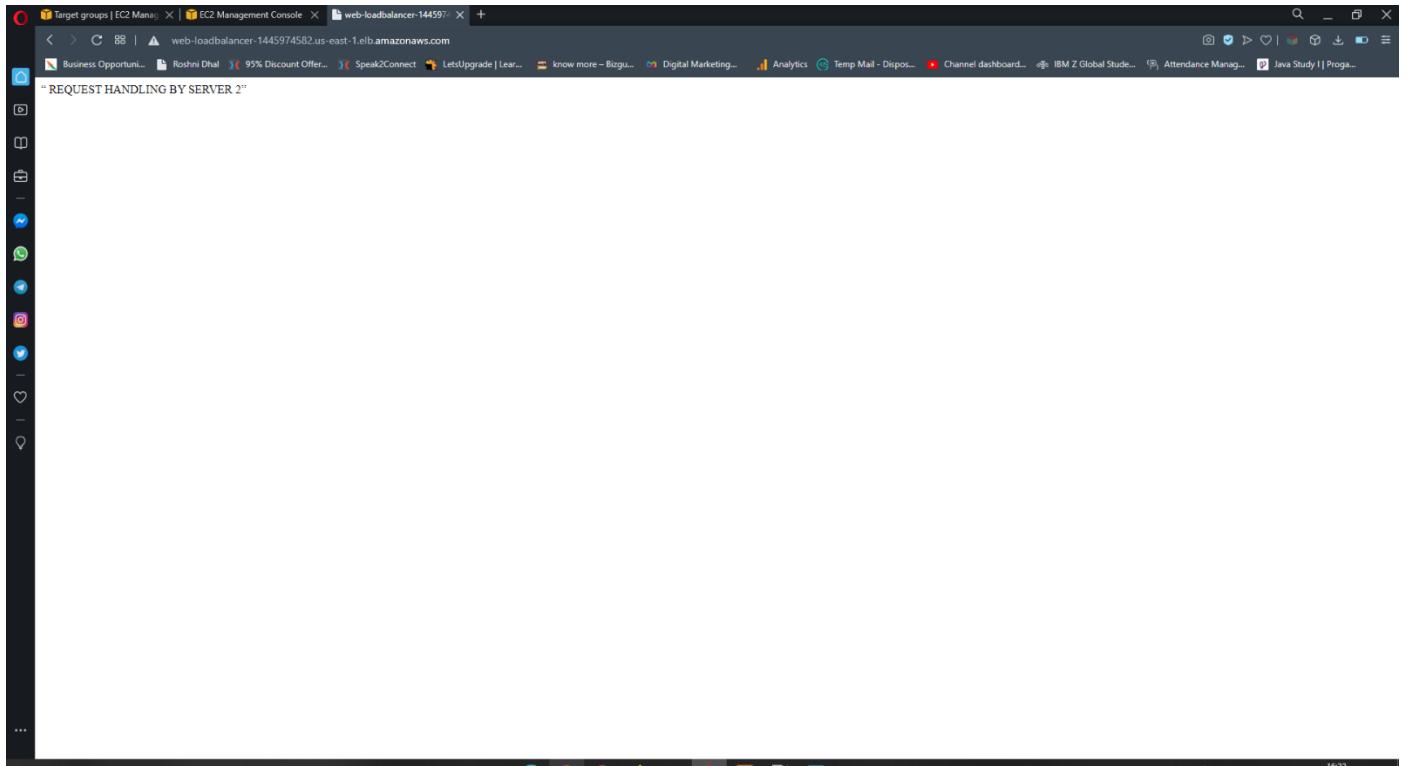
Stopping web server 1:

The screenshot shows the AWS EC2 Management Console. In the left sidebar, under the 'Instances' section, there is a list of three instances: 'bastion-server', 'web-server-1', and 'web-server-2'. The 'web-server-1' instance is selected and has a blue checkmark next to it. Its status is shown as 'Stopping'. The main pane displays a table of instances with columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm Status, Availability zone, Public IPv4 DNS, and Public IPv4 IP. The 'web-server-1' row is highlighted in blue. Below the table, a detailed view for 'Instance: i-0302839db79ec6175 (web-server-1)' is shown, with tabs for Details, Security, Networking, Storage, Status Checks, Monitoring, and Tags. The 'Details' tab is selected. It provides information such as Instance ID, Instance state (Stopping), Instance type (t2.micro), IAM Role (none), and VPC details like Subnet ID and VPC ID.

Health of webserver after being stopped:

The screenshot shows the AWS Target Groups page. In the left sidebar, under the 'Load Balancing' section, 'Target Groups' is selected. A target group named 'new-target-1' is listed. The 'Targets' tab is selected in the navigation bar. The 'Basic configuration' section shows the target type as 'instance', protocol as 'HTTP : 80', and VPC as 'vpc-0ef0246c86bdff45e'. The 'Registered targets' section lists two targets: 'web-server-2' (Instance ID: i-07ecb048677d8ff6e) and 'web-server-1' (Instance ID: i-0302839db79ec6175). Both targets are marked as 'healthy'.

When copy pasted DNS of Load balancer after Stopping of webserver 1:



Conclusion:

- Successfully created a new VPC from scratch and created both public and private subnets.
- Created an Internet Gateway and configured a new route table.
- Launched 1 EC2 instance each of the Public and Private subnets and tested Internet access from them.
- Provided Internet access to the EC2 instance in the Private subnet, you created a NAT Gateway and configured a Route table.
- Confirmed that the instance in the private subnet is able to connect to the internet.