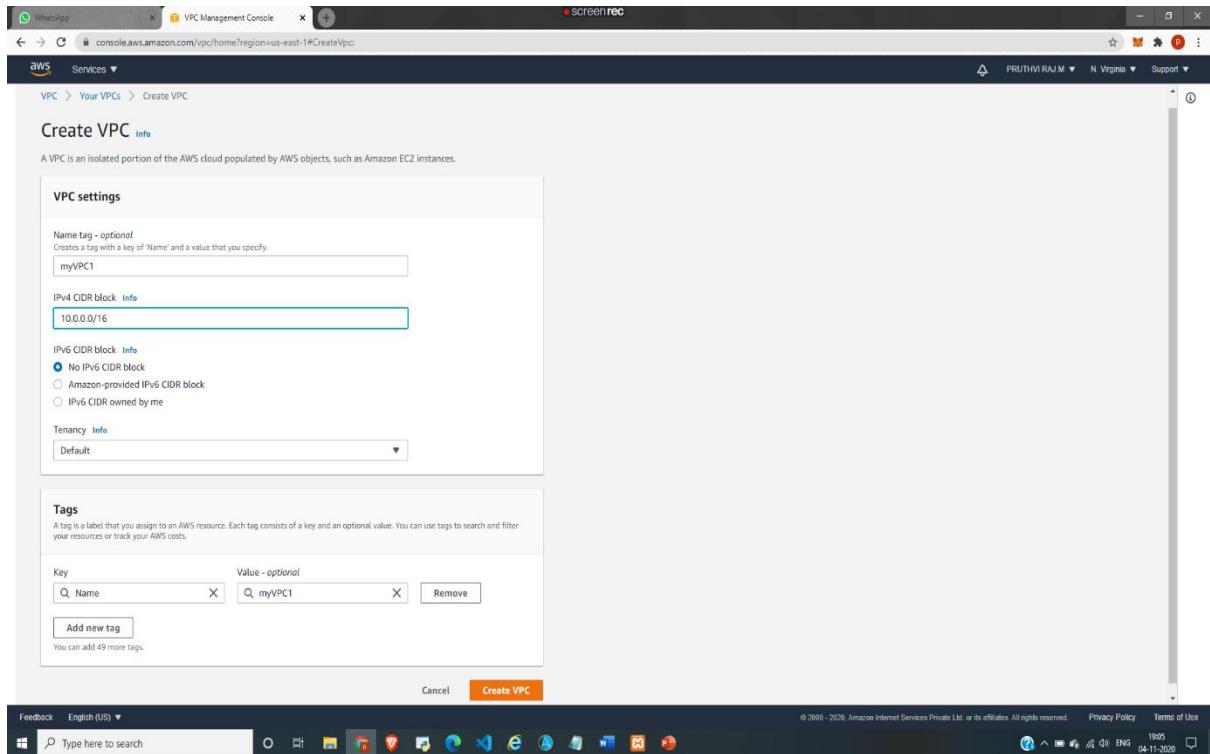


AWS PROJECT 1

DEPLOYMENT OF A HIGHLY AVAILABLE WEB APPLICATION

- PRUTHVI RAJ.M

VPC



VPC name : myVPC

CIDR Block : 10.0.0.0/16

Tenancy : Default

VPC DETAILS:

Your VPCs (1/2) Info

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR (Network border group)	IPv6 pool
vpc-a1539dc	vpc-0f8d4d4d5509f5117	Available	172.31.0.0/16	-	-
myVPC1	vpc-0f8d4d4d5509f5117	Available	10.0.0.0/16	-	-

Details | CIDRs | Flow logs | Tags

Details

VPC ID vpc-0f8d4d4d5509f5117	State Available	DNS hostnames Disabled	DNS resolution Enabled
Tenancy Default	DHCP options set dopt-5fabc325	Route table rtb-09c7cb4dd1994382d	Network ACL acl-0f748c1dea61ed5b8
Default VPC No	IPv4 CIDR 10.0.0.0/16	IPv6 pool -	IPv6 CIDR (Network border group) -
Owner ID 581731927073			

PUBLIC SUBNET

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.

Name tag	VPC*	Availability Zone	VPC CIDRs	IPv4 CIDR block*
mypublicsubnet	vpc-0f8d4d4d5509f5117	us-east-1a	CIDR 10.0.0.0/16 associated	10.0.0.0/24

* Required

Cancel 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 subnet ID is subnet-065404bd518677a96. The 'Auto-assign IPv4' checkbox is checked, indicating that public IPv4 addresses will be automatically assigned to instances launched in this subnet. 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.



PUBLIC SUBNET DETAILS:

The screenshot shows the 'Subnets | VPC Management Console' page. On the left, there's a navigation sidebar with options like 'New VPC Experience', 'VPC Dashboard', 'Filter by VPC', 'Select a VPC', and lists for 'VIRTUAL PRIVATE CLOUD', 'SECURITY', and 'VIRTUAL PRIVATE NETWORK (VPN)'. The main content area displays a table of subnets. One subnet is selected: mypublicsub... (subnet-065404bd518677a96). The table includes columns for Name, Subnet ID, State, VPC, IPv4 CIDR, Available IPv4, IPv6 CIDR, Availability Zone, Availability Zone ID, Network Border Group, and Route table. Below the table, there's a detailed view for the selected subnet, showing its ID, VPC, available IPv4 addresses (251), and configuration for auto-assigning customer-owned and auto-assigning IPv4 addresses. The bottom of the screen shows the Windows taskbar with various pinned icons and the date/time as 04-11-2020.

PRIVATE SUBNET

The screenshot shows the 'Create subnet' wizard in the AWS VPC Management Console. The form fields are as follows:

- Name tag: myprivatesubnet
- VPC: vpc-0f8d4d4d5509f5117
- Availability Zone: us-east-1b
- VPC CIDRs: 10.0.0.0/16
- IPv4 CIDR block: 10.0.1.0/24

Below the form, there is a note: "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 'Required' and 'Cancel' buttons at the bottom.



PRIVATE SUBNET DETAILS:

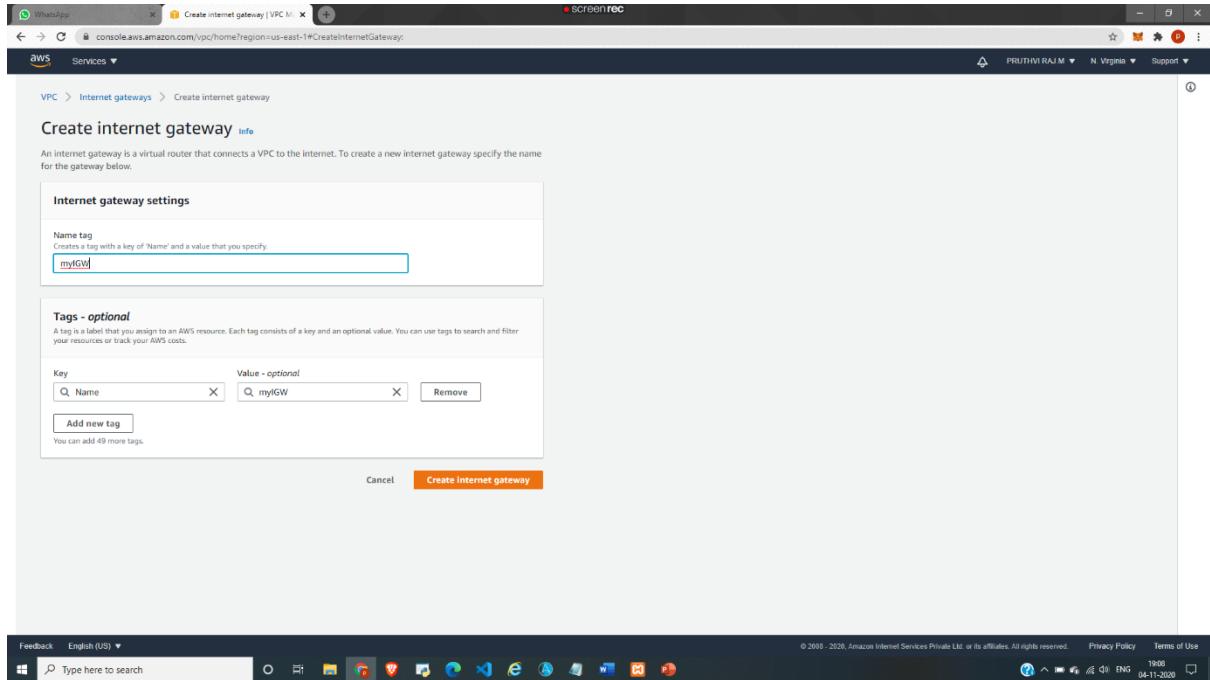
The screenshot shows the list of subnets in the AWS VPC Management Console. The table includes columns for Name, Subnet ID, State, VPC, IPv4 CIDR, Available IPv4, IPv6 CIDR, Availability Zone, Availability Zone ID, Network Border Group, and Route table. The subnet details for 'myprivatesubnet' are highlighted.

Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR	Availability Zone	Availability Zone ID	Network Border Group	Route table
myprivatesu...	subnet-05ed16861b678cb24	available	vpc-0f8d4d4d5509f5117 myVPC1	10.0.1.0/24	251	-	us-east-1b	use1-a22	us-east-1	rtb-09c7cb4dd195
mypublicsu...	subnet-0b136705	available	vpc-0f8d4d4d5509f5117 myVPC1	10.0.0.0/24	251	-	us-east-1a	use1-a21	us-east-1	rtb-09c7cb4dd195
	subnet-0b936444	available	vpc-0f8d4d4d5509f5117 myVPC1	172.31.64.0/20	4091	-	us-east-1f	use1-a25	us-east-1	rtb-139cd66d
	subnet-0832163ce	available	vpc-0f8d4d4d5509f5117 myVPC1	172.31.16.0/20	4091	-	us-east-1c	use1-a24	us-east-1	rtb-139cd66d
	subnet-c1e55ce0	available	vpc-0f8d4d4d5509f5117 myVPC1	172.31.80.0/20	4091	-	us-east-1b	use1-a22	us-east-1	rtb-139cd66d
	subnet-cc59eaaa	available	vpc-0f8d4d4d5509f5117 myVPC1	172.31.0.0/20	4091	-	us-east-1a	use1-a21	us-east-1	rtb-139cd66d
	subnet-c0e87ef	available	vpc-0f8d4d4d5509f5117 myVPC1	172.31.48.0/20	4091	-	us-east-1e	use1-a23	us-east-1	rtb-139cd66d

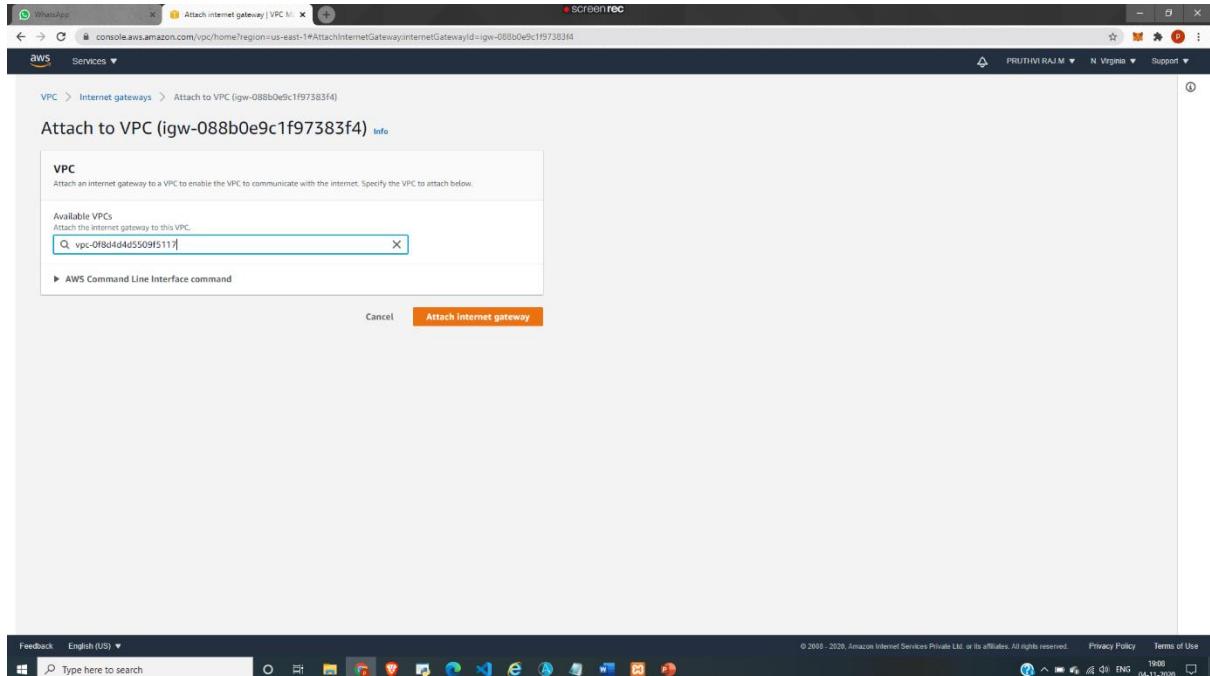
Below the table, the subnet details for 'myprivatesubnet' are shown:

Subnet ID	State	available
subnet-065404bd518677a96	IPV4 CIDR	10.0.0.0/24
VPC	IPv6 CIDR	-
vpc-0f8d4d4d5509f5117 myVPC1	Network Border Group	us-east-1
Available IPv4 Addresses	Network ACL	aci-0748c1deaf1ed5b8
251	Auto-assign public IPv4	Yes
Availability Zone	Customer-owned IPv4 pool	-
us-east-1a (use1-a21)	Output ID	-
Route Table	Customer-owned IPv4 address	-
rtb-09c7cb4dd194382d	Owner	581731927073
Default subnet		No

INTERNET GATEWAY



ATTACH IGW WITH VPC:



INTERNET GATEWAY DETAILS:

The screenshot shows the AWS VPC Internet Gateways page. On the left, there's a navigation sidebar with various VPC-related options like 'Your VPCs', 'Route Tables', and 'Internet Gateways'. The main content area displays a table of Internet Gateways. One gateway, 'myIGW' (ID: igw-088b0e9c1f97383f4), is listed as 'Attached' to 'vpc-0f8d4d4d5509f5117 | myVPC'. Another gateway, 'igw-fbb28080', is also attached to the same VPC. Below the table, a 'Details' section provides more information about the selected gateway.

PUBLIC ROUTE TABLE

The screenshot shows the AWS Route Tables page. A 'Create route table' dialog is open. It requires a 'Name tag' (set to 'publicroutetable') and a 'VPC' (set to 'vpc-0f8d4d4d5509f5117'). There's a 'Tags' section where a single tag 'Key: Value' is added. At the bottom, there are 'Cancel' and 'Create' buttons. The status bar at the bottom indicates the date as '04-11-2020'.

EDITING ROUTES IN PUBLIC TABLE:

The screenshot shows the AWS VPC Management console with the URL <https://console.aws.amazon.com/vpc/home?region=us-east-1#EditRouteTable:routeTableId=rtb-0bced558a5e0d557>. The page title is "Edit routes". A table lists two routes:

Destination	Target	Status	Propagated
10.0.0.0/16	local	active	No
0.0.0.0	igw-088b0e9c197383f4	No	

Buttons at the bottom include "Add route", "Cancel", and "Save routes". A note says "* Required".



ASSOCIATING PUBLIC ROUTE TABLE WITH PUBLIC SUBNET:

The screenshot shows the AWS VPC Management console with the URL <https://console.aws.amazon.com/vpc/home?region=us-east-1#EditRouteTableSubnetAssociations:routeTableId=rtb-025cf147011d0cd3>. The page title is "Edit subnet associations". A table lists subnet associations:

Associated subnets	Subnet ID	IPv4 CIDR	IPv6 CIDR	Current Route Table
subnet-05ed16861b678cb24	subnet-05404bd518577a66 mypublics...	10.0.0.0/24	-	Main
	subnet-05ed16861b678cb24 myprivate...	10.0.1.0/24	-	Main

Buttons at the bottom include "Cancel" and "Save". A note says "* Required".



DETAILS OF PUBLIC ROUTE TABLE:

The screenshot shows the AWS VPC Management console with the 'Route Tables' section selected. A table lists three route tables, with the first one, 'publicroutetb-08c6e558a5e0dd987', highlighted. The table includes columns for Name, Route Table ID, Explicit subnet association, Edge associations, Main, VPC ID, and Owner. Below the table, a detailed view of the selected route table shows its ID, association status, and owner.

Name	Route Table ID	Explicit subnet association	Edge associations	Main	VPC ID	Owner
publicroutetb-08c6e558a5e0dd987	rtb-08c6e558a5e0dd987	-	-	No	vpc-0fb4d4d5509f5117	581731927073
rtb-09c7c4dd1994362d	rtb-09c7c4dd1994362d	-	-	Yes	vpc-0fb4d4d5509f5117	581731927073
rtb-139cd96d	rtb-139cd96d	-	-	Yes	vpc-a1539cc0c	581731927073

PRIVATE ROUTE TABLE

The screenshot shows the 'Create route table' wizard. The first step, 'Create route table', is displayed. It asks for a name tag ('privateroutetable') and a VPC ('vpc-0fb4d4d5509f5117'). A note indicates that the resource currently has no tags. At the bottom, there are 'Cancel' and 'Create' buttons.

Name tag: privateroutetable

VPC: vpc-0fb4d4d5509f5117

Key (128 characters maximum) | Value (256 characters maximum)

This resource currently has no tags

Add Tag 50 remaining (Up to 50 tags maximum)

* Required

Create

ASSOCIATING PRIVATE ROUTE TABLE WITH PRIVATE SUBNET:

The screenshot shows the AWS VPC console with the URL <https://console.aws.amazon.com/vpc/home?region=us-east-1#EditRouteTableSubnetAssociationsrouteTableId:rtb-025cf147011d0cd3>. The page title is "Edit subnet associations". A table lists subnet associations:

Subnet ID	IPv4 CIDR	IPv6 CIDR	Current Route Table
subnet-05ed16861b678cb24	10.0.0.0/24	-	Main
subnet-05ed16861b678cb24	10.0.1.0/24	-	Main

Buttons at the bottom include "Required", "Cancel", and "Save".



DETAILS OF PRIVATE ROUTE TABLE:

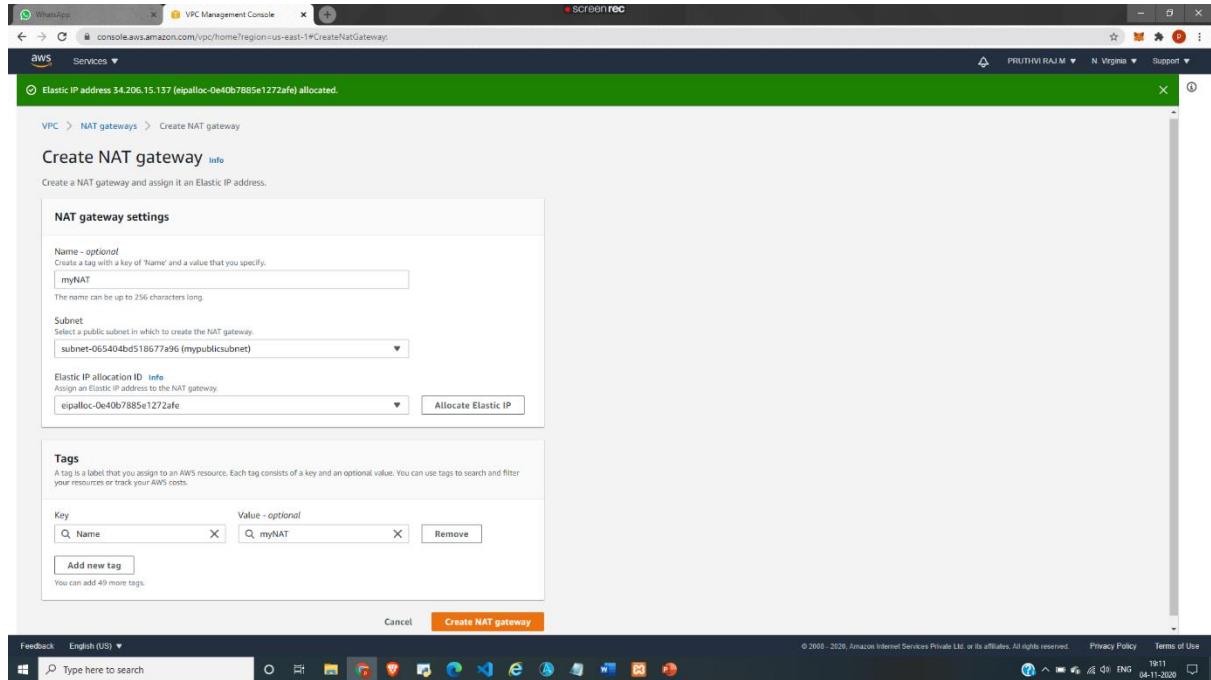
The screenshot shows the AWS VPC console with the URL <https://console.aws.amazon.com/vpc/home?region=us-east-1#RouteTablessort:routeTableId>. The left sidebar shows "Route Tables" selected. The main table displays route tables:

Name	Route Table ID	Explicit subnet association	Edge associations	Main	VPC ID	Owner
privateroute	rtb-025cf147011d0cd3	subnet-05ed16861b678cb24	-	No	vpc-0f8d4d4d5509f5117	581731927073
publicrout...	rtb-08c6e556a5e0dd987	-	-	No	vpc-0f8d4d4d5509f5117	581731927073
...	rtb-09c7b4dd1994382d	-	-	Yes	vpc-0f8d4d4d5509f5117	581731927073
...	rtb-139cd56d	-	-	Yes	vpc-a15390dc	581731927073

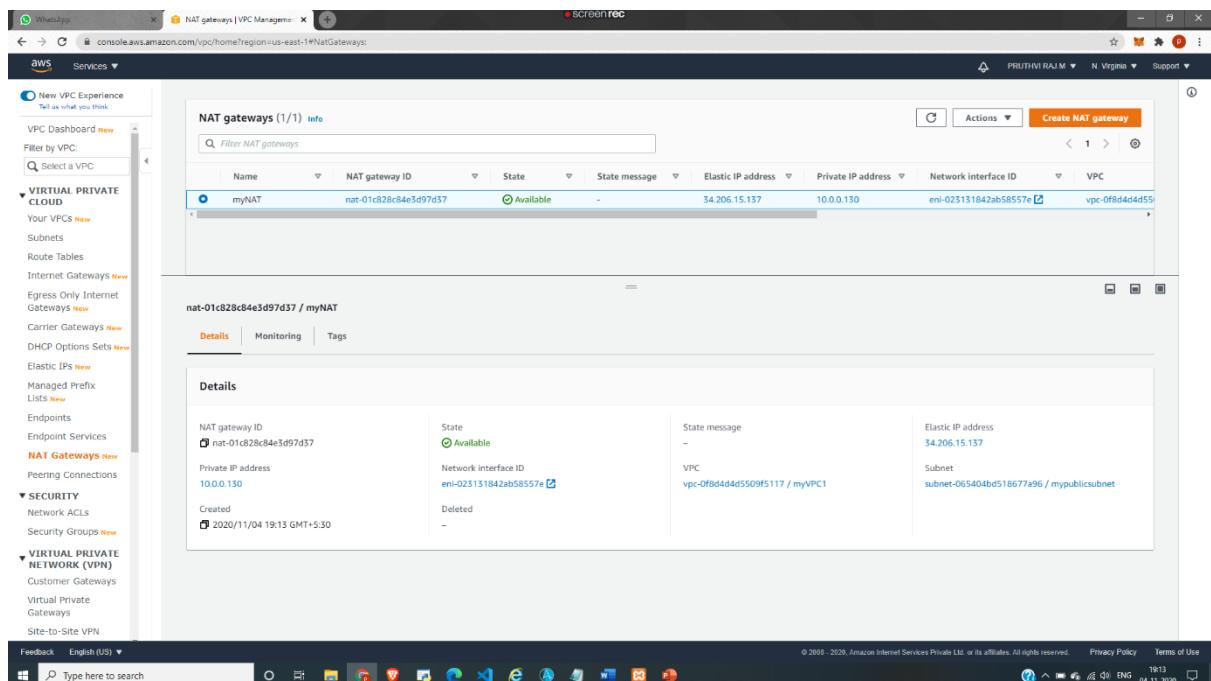
Below the table, a summary for the route table **rtb-025cf147011d0cd3** is shown:

Route Table ID	rtb-025cf147011d0cd3	Main	No
Explicitly Associated with	subnet-05ed16861b678cb24	VPC	vpc-0f8d4d4d5509f5117 myVPC1
Owner	581731927073		

NAT GATEWAY



DETAILS OF NAT GATEWAY:



EDITING ROUTES IN PRIVATE ROUTE TABLE:

WhatsApp

Edit routes | VPC Management C

screenrec

console.aws.amazon.com/vpc/home?region=us-east-1#editRoutes?routeTableId=rtb-025cf147011d0cd3

Services ▾

Route Tables > Edit routes

Edit routes

Destination	Target	Status	Propagated
10.0.0.0/16	local	active	No
0.0.0.0	nat-01c828c84e3d97d37	active	No

Add route

* Required

Cancel Save routes



SECURITY GROUP

WhatsApp VPC Management Console Screenrec

console.aws.amazon.com/vpc/home?region=us-east-1#CreateSecurityGroup; PRUTHVI RAJAM N Virani Support

VPC Services

VPC > Security Groups > Create security group

Create security group Info

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

Security group name Info
loadbalancer-SG
Name cannot be edited after creation.

Description Info
security group for load balancer

VPC Info
vpc-0f8d4d45509f5117 (myVPC1)

Inbound rules Info

Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Source <small>Info</small>	Description - optional <small>Info</small>	Delete
HTTP	TCP	80	Custom	<input type="text" value="0.0.0.0"/> <input type="button" value="X"/>	<input type="button" value="Delete"/>

Add rule

Outbound rules Info

Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Destination <small>Info</small>	Description - optional <small>Info</small>	Delete
All traffic	All	All	Custom	<input type="text"/> <input type="button" value="X"/>	<input type="button" value="Delete"/>

Feedback English (US) Type here to search © 2006–2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use 19:19 04-11-2020

DETAILS OF SECURITY GROUP:

The screenshot shows the 'Create security group' page in the AWS VPC Management Console. The security group name is 'loadbalancer-SG'. In the 'Inbound rules' section, there is a single rule for port 80 allowing HTTP traffic from 0.0.0.0/0. The 'Outbound rules' section shows all traffic allowed.

BASTION SERVER-USING PUBLIC SUBNET

CREATING INSTANCE IN PUBLIC SUBNET IS CALLED AS BASTION SERVER

CONFIGURE INSTANCE:

The screenshot shows the 'Step 3: Configure Instance Details' page of the AWS Launch Instance Wizard. The instance type is selected as 't2.micro'. The instance is launching into a public subnet ('Subnet: 065404b1e18677496 | mypublicsubnet (us-east-1)'). Other configuration options include 'Auto-assign Public IP' (selected), 'Placement group' (disabled), 'Capacity Reservation' (Open), 'Domain Join directory' (No directory), 'IAM role' (None), 'CPU options' (disabled), 'Shutdown behavior' (Stop), 'Stop - Hibernate behavior' (disabled), 'Enable termination protection' (disabled), 'Monitoring' (disabled), 'Tenancy' (Shared - Run a shared hardware instance), and 'Elastic Inference' (disabled). At the bottom, there are buttons for 'Cancel', 'Previous', 'Review and Launch', and 'Next: Add Storage'.

STORAGE FOR BASTION SERVER:

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Deletes on Termination	Encryption
Root	/dev/xvda	snap-0299d083f0ce5cd12	8	General Purpose SSD (gp2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted

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.

ADDING TAGS FOR BASTION-SERVER:

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

Add another tag (Up to 50 tags maximum)

CONFIGURING SECURITY GROUP FOR BASTION SERVER:

The screenshot shows the AWS Launch Instance wizard at Step 6: Configure Security Group. The user has selected 'Create a new security group'. The security group name is 'bastion-securitygroup' and the description is 'security group for bastion server'. A single rule is defined: Type: SSH, Protocol: TCP, Port Range: 22, Source: Custom (0.0.0.0/0), Description: e.g. SSH for Admin Desktop. A warning message at the bottom states: '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.'

A screenshot of a Windows desktop environment. The taskbar at the bottom shows various pinned icons including File Explorer, Edge browser, and other system icons. The system tray on the right displays network status, battery level, and system time (22:16, 04-11-2020).

REVIEW OF INSTANCE FOR BASTION-SERVER:

The screenshot shows the AWS Launch Instance wizard at Step 7: Review Instance Launch. The summary includes:

- AMI Details:** Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0947d2ba12ee1ff7f. It is a Free tier eligible instance.
- Instance Type:** t2.micro
- Security Groups:** bastion-securitygroup (description: security group for bastion server). A rule allows SSH (TCP port 22) from 0.0.0.0/0.
- Launch Details:** The 'Launch' button is present at the bottom right.

A note at the top of the review section 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.'

BASTION SERVER DETAILS:

The screenshot shows the AWS EC2 Management Console. The left sidebar navigation includes 'Instances', 'Images', 'Elastic Block Store', 'Network & Security', 'Load Balancing', and 'Auto Scaling'. The main content area displays the 'Instances (1/1) info' section. One instance is listed: 'bastion_server' (Instance ID: i-0a466f051bb756b34), which is 'Running' and has an 't2.micro' instance type. It is located in the 'us-east-1a' availability zone with a public IPv4 address of 3.91.174.236. The 'Details' tab of the instance summary is selected, showing details like Instance ID, Instance state, Instance type, IAM Role, Platform, Public IPv4 address, Private IPv4 addresses, Public IPv4 DNS, Private IPv4 DNS, VPC ID, and Subnet ID.

WEB-SERVER1 – USING PRIVATE SUBNET

CONFIGURE INSTANCE DETAILS :

The screenshot shows the 'Launch instance wizard | EC2 M...' step 3: 'Configure Instance Details'. The configuration includes:

- Number of instances: 1
- Purchasing option: Request Spot Instances
- Network: vpc-05d4dd5509f5117 | myVPC1
- Subnet: subnet-05ed16661b678c24 | myprivatesubnet | us-east-1
- Auto-assign Public IP: Use subnet setting (Disable)
- Placement group: Add instance to placement group (unchecked)
- Capacity Reservation: Open
- Domain join directory: No directory
- IAM role: None
- CPU options: Specify CPU options (unchecked)
- Shutdown behavior: Stop
- Stop - Hibernate behavior: Enable hibernation as an additional stop behavior (unchecked)
- Enable termination protection: Protect against accidental termination (unchecked)
- Monitoring: Enable CloudWatch detailed monitoring (Additional charges apply) (unchecked)
- Tenancy: Shared - Run a shared hardware instance (Additional charges will apply for dedicated tenancy)
- Elastic Inference: Add an Elastic Inference accelerator (Additional charges apply) (unchecked)

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

ADDING STORAGE TO WEB-SERVER1:

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)	Deletes 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.

ADDING TAGS:

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		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Add another tag (Up to 50 tags maximum)

ADDING CONFIGURE SECURITY GROUP:

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 <input type="text" value="sg-0548e1b910ef87c4c"/>	e.g. SSH for Admin Desktop
HTTP	TCP	80	Custom <input type="text" value="sg-04da56d50243bd402"/>	e.g. SSH for Admin Desktop

[Add Rule](#)

[Cancel](#) [Previous](#) [Review and Launch](#)

REVIEW OF INSTANCE FOR WEB-SERVER1:

Step 7: Review Instance Launch

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.

AMI Details

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0947d2ba12ee1ff77

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

Security group name	Description
web-server-SG	security group for web server

Instance Details

Storage

Tags

[Cancel](#) [Previous](#) [Launch](#)

DETAILS OF WEB-SERVER1:

The screenshot shows the AWS EC2 Management Console. On the left, there's a navigation sidebar with sections like Instances, Images, Elastic Block Store, Network & Security, Load Balancing, and Auto Scaling. The main area is titled 'Instances (1/1) info' and shows a single instance named 'web-server1'. The instance is running, has an 't2.micro' type, and is in the 'us-east-1b' availability zone. It has a public IPv4 address of '10.0.1.145' and a private IPv4 address of 'ip-10-0-1-145.ec2.internal'. The VPC ID is 'vpc-0f8d4d4d5509f5117 (myVPC1)'. The instance was launched on '04-11-2020' at '22:25'. The interface includes tabs for Details, Security, Networking, Storage, Status Checks, Monitoring, and Tags.

WEB-SERVER-2 – USING PRIVATE SUBNET

CONFIGURE INSTANCE DETAILS:

The screenshot shows the 'Launch instance wizard | EC2 Management Console' on step 3: 'Configure Instance Details'. The page is titled 'Step 3: Configure Instance Details' and says '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.' There are several configuration sections: 'Number of instances' (set to 1), 'Purchasing option' (Request Spot instances), 'Network' (VPC: 'vpc-0f8d4d4d5509f5117 | myVPC1'), 'Subnet' ('subnet-05e1d1961b678c324 | myprivatesubnet | us-east-1'), '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, there are buttons for 'Cancel', 'Previous', 'Review and Launch', and 'Next: Add Storage'.

ADDING STORAGE:

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

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.](#)

ADDING TAGS:

Key	Value
Name	[web-server1]

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.](#)

CONFIGURE SECURITY GROUP:

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 ID	Name	Description	Actions
sg-0546e10910ef67c4c	bastion-securitygroup	security group for bastion server	Copy to new
sg-0f03e47ea9bf151c3	bastion-SG	security group for bastion server	Copy to new
sg-097cer178930515ec	default	default VPC security group	Copy to new
sg-04da56d502438b402	loadbalancer-SG	security group for load balancer	Copy to new
sg-0a2f9e34f977e32ac	web-server-SG	security group for web server	Copy to new

Inbound rules for sg-0a2f9e34f977e32ac (Selected security groups: sg-0a2f9e34f977e32ac)

Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	sg-04da56d502438b402 (loadbalancer-SG)	
SSH	TCP	22	sg-0546e10910ef67c4c (bastion-securitygroup)	

REVIEW LAUNCH INSTANCE:

Step 7: Review Instance Launch

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.

AMI Details

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

Free tier: Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras.

Root Device Type: ebs Virtualization type: hvm

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

Security Group ID	Name	Description
sg-0a2f9e34f977e32ac	web-server-SG	security group for web server

All selected security groups inbound rules

Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	sg-04da56d502438b402 (loadbalancer-SG)	
SSH	TCP	22	sg-0546e10910ef67c4c (bastion-securitygroup)	

Instance Details

Storage

Tags

Cancel Previous **Launch**

DETAILS OF WEB-SERVER2:

The screenshot shows the AWS EC2 Management Console. On the left, a sidebar lists various services: Instances, Images, Elastic Block Store, Network & Security, Load Balancing, and Auto Scaling. The 'Instances' section is expanded, showing a table of instances. One instance, 'web-server2', is selected and highlighted in blue. The main pane displays the details for 'Instance: i-0d219da3fc1f36c21 (web-server2)'. The 'Details' tab is selected, showing the following information:

Attribute	Value
Instance ID	i-0d219da3fc1f36c21
Instance state	Running
Instance type	t2.micro
AMI ID	ami-0947d2ba12ee1ff75
Platform	Amazon Linux (inferred)

Below the table, the 'Monitoring' section indicates that monitoring is disabled.

LOAD BALANCER – FOR TWO WEB-SERVERS

LOAD BALANCER TYPE:

This screenshot is identical to the one above, showing the AWS EC2 Management Console. It displays the same instance details for 'web-server2' (i-0d219da3fc1f36c21). The 'Details' tab is selected, showing the same configuration and monitoring status as the previous screenshot.

CONFIGURE LOAD BALANCER:

The screenshot shows the AWS Create Load Balancer wizard at Step 1: Configure Load Balancer. The basic configuration section includes a Name field set to "web-application-LB", a Scheme field set to "internet-facing", and an IP address type field set to "IPv4". The listeners section shows a single listener for the "HTTP" protocol on port 80. The availability zones section lists two zones: "us-east-1a" and "us-east-1b", each associated with a specific subnet. The bottom of the screen shows a Windows taskbar with various icons.

CONFIGURE SECURITY SETTINGS:

The screenshot shows the AWS Create Load Balancer wizard at Step 2: Configure Security Settings. A warning message in a yellow box states: "⚠ 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." The "Next: Configure Security Groups" button is visible at the bottom right.

CONFIGURE SECURITY GROUPS:

The screenshot shows the AWS Lambda configuration interface. The top navigation bar includes tabs for WhatsApp, Create Load Balancer | EC2 Manager, Assignment - Google Drive, and screenrec. The main title is "Step 3: Configure Security Groups". A sub-instruction states: "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." Below this, there are two radio button options: "Create a new security group" (unchecked) and "Select an existing security group" (checked). A table lists existing security groups with columns for Security Group ID, Name, Description, and Actions (Copy to new). One row is selected: "sg-04da56d502438b402" (loadbalancer-SG), which is highlighted in blue.

CONFIGURE ROUTING:

The screenshot shows the AWS Lambda configuration interface. The top navigation bar includes tabs for WhatsApp, Create Load Balancer | EC2 Manager, Assignment - Google Drive, and screenrec. The main title is "Step 4: Configure Routing". A sub-instruction states: "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." The "Target group" section is expanded, showing fields for Target group (New target group), Name (web-app-TG), Target type (Instance, selected), Protocol (HTTP, selected), Port (80), and Protocol version (HTTP1, selected). The "Health checks" section is also visible, showing Protocol (HTTP) and Path (index.html). At the bottom, there is a link for "Advanced health check settings".

REGISTER TARGETS:

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.

Remove	Instance	Name	Port	Status	Security groups	Zone
	i-07e500ae65251b995	web-server1	80	running	web-server-SG	us-east-1b
	i-0d219da3fc1f36c21	web-server2	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.

Add to registered	on port	80				
Search Instances						
Instance	Name	State	Security groups	Zone	Subnet ID	Subnet CIDR
i-046f051b756b34	bastion_server	running	bastion-securitygroup	us-east-1a	subnet-065404bd518677a96	10.0.0.0/24
i-07e500ae65251b995	web-server1	running	web-server-SG	us-east-1b	subnet-05ed16861b678cb24	10.0.1.0/24
i-0d219da3fc1f36c21	web-server2	running	web-server-SG	us-east-1b	subnet-05ed16861b678cb24	10.0.1.0/24

Cancel Previous Next: Review

REVIEW LOAD BALANCER:

Step 6: Review

Please review the load balancer details before continuing.

Load balancer

Name: web-application-LB
Scheme: internet-facing
Listeners: Port 80 - Protocol: HTTP
IP address type: ipv4
VPC: vpc-0fbdd4d45509f5117 (myVPC1)
Subnets: subnet-065404bd518677a96 (mypublicsubnet), subnet-05ed16861b678cb24 (myprivatesubnet)

Security groups

Security groups: sg-04da56d502438b402

Routing

Target group: New target group
Target group name: web-app-TG
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: 2
Timeout: 5s
Interval: 30s
Success codes: 200

Targets

Instances: i-07e500ae65251b995 (web-server1:80), i-0d219da3fc1f36c21 (web-server2:80)

Cancel Previous Create

DETAILS OF WEB-APPLICATION-LB:

The screenshot shows the AWS EC2 Management Console with the Load Balancers page open. A table lists one load balancer named "web-application-LB". The details page shows the following configuration:

- Name:** web-application-LB
- ARN:** arn:aws:elasticloadbalancing:us-east-1:581731927073:loadbalancer/app/web-application-LB/2b99a70bfe822614
- DNS name:** web-application-LB-511780239.us-east-1.elb.amazonaws.com (A Record)
- State:** provisioning
- Type:** application
- Scheme:** internet-facing
- IP address type:** ipv4
- VPC:** vpc-0f8d44d5509f5117
- Availability Zones:** subnet-05e168f1b678c24 - us-east-1b, subnet-06540ab5b18677996 - us-east-1a
- Hosted zone:** Z365XKDOTRQ7X7K

CONNECT BASHION SERVER:

The screenshot shows the AWS EC2 Management Console with the Connect to instance page open for an instance with ID i-0a466f051bb756b34. The page displays the following information:

- EC2 Instance Connect** (selected)
- Session Manager**
- SSH client**

Instance ID: i-0a466f051bb756b34 (bastion_server)

Public IP address: 3.91.174.236

User name: ec2-user

Note: In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

UPDATING LINUX:

```
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-10-0-0-225 ~]$ sudo su
[root@ip-10-0-0-225 ec2-user]# yum update -y
```

i-0a466f051bb756b34 (bastion_server)
Public IP: 3.91.174.236 Private IP: 10.0.0.225

INSTALLING HTTPD:

```
Verifying : cpio-2.11-27.amzn2.x86_64
Verifying : glibc-2.26-35.amzn2.x86_64
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.amzn2.2.2.x86_64
Verifying : glibc-all-langpacks-2.26-35.amzn2.x86_64
Verifying : e2fsprogs-1.42.9-12.amzn2.0.2.x86_64
Verifying : libcroco-0.6.11-1.amzn2.0.2.x86_64
Verifying : awscli-bootstraps-1.4.32.amzn2.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 0:3.0.1.161.0-1.amzn2
  bash.x86_64 0:4.2.46-34.amzn2
  e2fsprogs-libs.x86_64 0:1.42.9-19.amzn2
  expat.x86_64 0:2.1-12.amzn2
  glibc-common.x86_64 0:2.26-37.amzn2
  hunspell.x86_64 0:1.1.6-16.amzn2
  libcom_err.x86_64 0:1.42.9-19.amzn2
  libcroco.x86_64 0:1.4.3-12.amzn2
  libssh2.x86_64 0:1.4.3-12.amzn2
  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.g7d1cfdb8.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-10-0-0-225 ec2-user]# yum install httpd -y
```

i-0a466f051bb756b34 (bastion_server)
Public IP: 3.91.174.236 Private IP: 10.0.0.225

CREATED FILE USING "VI WEB-SERVERKEY.PEM" COMMAND:

-COPY PASTED .PEM FILE INT IT

CHANGING PERMISSIONS OF WEB-SERVERKEY.PEM FILE:



SSH INTO PRIVATE SUBNET USING PUBLIC SUBNET:

```
(D) WhatsApp Instances: EC2 Management Con - o-2466051bb75b34 (bastion) - [Assignment - Google ScreenToGo] console.amazonaws.com:ec2/v2/connect/ec2-user/-/o-2466051bb75b34
BBf3c74z-2m3XHgkA85d5qzLzR53TzIrLH5k/y+7h/qg20FvcfCbRoA3H0jYjt+ntPms4Y5am561WHLKAlApJES/1qZBnHqs14aaAOXkx2/7WkrnhXshhacS7bv1gEwDX72wMPR23xv8g1zLbdxSA5PrFyzs0MNT2Jo6pT7jsGU07s6a1Oh1FxQo3c1ZPn0a9VNmmt07q;cbu3C3w784dg90ZTYHs9nD2pd+pTs+ss+25FHts;YhykfnJ9SamRo4G79K0WnL4+96nTsSz97oeKjYzUjdp+6eaal0TDAA0ABaIBAHkTwIt2vFvPUje3z+13jTr096n7b017KA1Yf7uN1L2zsp3r34as8RKnbgkLGxkLAR+jE01kzd/x3rsxD34Cj15j6eOpDB485y5csEoC0waUkti7sMchm78g1LUY70HdbVpCazym0vmsWtfH2z1p3g748G00h/TH5/Tj;jqP01goZ12zirZ3c0u5+0tvsDFY1r7uyUTChAId15xaDrZ7t1ccTMpBp271BaDR12Bv97W80/joupSD094dbhas2+zc4ukev2548Fn0eq1uh0.1G84sm3EBH17YtaWadgJAg60j1kwL5frZ7z2o0eo2po0l0thgeCgAyai9rsnuHx8fj9az/1Y7gd+COpTiF7n810Eb5J1rwsjnaCdaUvB+16BH7/fdFMkp rh68H2n0tD1aCg76gbezo8IBs302871wHzy2+4KYm7d1r*xzCnPWhvb10gNzq165sttuDkByW8Y65s9j7oEg/XUKHx6zobekhY58TECgjYEAH1u3zH1769jR0wGvAjty/duw)uj13zHmHn/ZUR0T1B61dc1YnszcEhrUp3R=5m0VAb4ZYBmBysFTbgaH1P+2kpxm1gUm9eqzXlMwK56...59/1UDH131MmgE+Hfy0q014d51K+1K+1bA071aBd7f1x5bsAb/18v1Lc49AsJ1xb0s5VaCoC3xWPBq45K5W5fPf2RDYXk1AP761ETKvduY1M+5t+kfbJlojH+Kz+VjogUdYwMvryJ1zU4N4WmkJd_f0f1/pwzRbQgBwEl02055kuHRP8/P91B14Awb+0rY7k1lD2G5v5vp1qPhBvBtZ0y+rfo+HduoPqh+pwuL3F1L45ZvNrhkX7F+o1n2h7qwpV0283n1RbtEcT1AKUJthrt+0YzmpYTe1gotp2frfr+s776mEb0n0d776sCBCCExVRvsgJaogAMDW#F2L1skdAO1jTu70g00ddeCm8Pw7Pjor8kdpPK7/oN5y1Av+xxH0aoXScd3/HPT4XvB+4BfSt+dix9VsANci8k3fjd3LDj+0pnvmTNjPbB9ZnwxTjkGzB1g3wkvRpCAX1z0r7XU3t7yls5KEIKJFk1Ijy+vtfjic=-----END RSA PRIVATE KEY-----
```

i-0a466f051bb756b34 (bastion_server)

Public IPs: 3.91.174.236 Private IPs: 10.0.0.225



i-0a466f051bb756b34 (bastion_server)

Public IPs: 3.91.174.236



UPDATE TO LINUX IN WEBSERVER1:

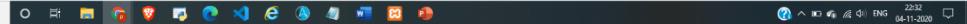
```
0 WhatsApp Instances | EC2 Management Con i-0a466f051bb756b34 (bastion_ Assignment - Google D screenrec
← C consoleaws.amazon.com/ec2/v2/connect/ec2-user/i-0a466f051bb756b34 bastion_ Assignment - Google D screenrec
Verifying : cpio-2.11-27.amzn2.x86_64
Verifying : glibc-2.26-35.amzn2.x86_64
Verifying : 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 : zlib-1.2.11-23.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.amzn2.2.2.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-botcore-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.gid1c6db8.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-10-0-1-145 ec2-user]#
```

i-0a466f051bb756b34 (bastion_server)
Public IP: 3.91.174.236 Private IP: 10.0.225



INSTALLING HTTPD IN WEB-SERVER1:

```
0 WhatsApp Instances | EC2 Management Con i-0a466f051bb756b34 (bastion_ Assignment - Google D screenrec
← C consoleaws.amazon.com/ec2/v2/connect/ec2-user/i-0a466f051bb756b34 bastion_ Assignment - Google D screenrec
Verifying : cpio-2.11-27.amzn2.x86_64
Verifying : glibc-2.26-35.amzn2.x86_64
Verifying : e2fsprogs-libs-1.42.9-12.amzn2.0.2.x86_64
Verifying : gzip-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.amzn2.2.2.x86_64
Verifying : glibc-all-langpacks-2.26-35.amzn2.x86_64
Verifying : e2fsprogs-1.42.9-12.amzn2.0.2.x86_64
Verifying : libxml2-2.9.1-6.amzn2.x86_64
Verifying : aws-cfn-bootstrap-1.4-32.amzn2.0.1.noarch
Verifying : python2-botcore-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.gid1c6db8.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-10-0-1-145 ec2-user]# yum install httpd -y
```

i-0a466f051bb756b34 (bastion_server)
Public IP: 3.91.174.236 Private IP: 10.0.225

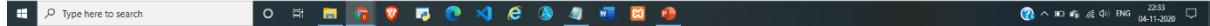


CREATING “INDEX.HTML” FILE IN WEB-SERVER1:

```
[5/9]: httpd-filesystem-2.4.46-1.amzn2.noarch.rpm          23 kB 00:00:00
[6/9]: httpd-tools-2.4.46-1.amzn2.x86_64.rpm             87 kB 00:00:00
[7/9]: mailcap-2.1.43-2.amzn2.noarch.rpm                31 kB 00:00:00
[8/9]: httpd-2.4.46-1.amzn2.x86_64.rpm                 1.3 MB 00:00:00
[9/9]: mod_http2-1.15.14-2.amzn2.x86_64.rpm            147 kB 00:00:00
Total                                         5.6 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           1/9
  Installing : apr-util-bdb-1.6.1-5.amzn2.0.2.x86_64   2/9
  Installing : apr-util-1.6.1-5.amzn2.0.2.x86_64       3/9
  Installing : httpd-tools-2.4.46-1.amzn2.x86_64      4/9
  Installing : generic-logos-httd-18.0.0-4.amzn2.noarch 5/9
  Installing : mailcap.noarch                         6/9
  Installing : httpd-filesystem-2.4.46-1.amzn2.noarch 7/9
  Installing : mod_http2-1.15.14-2.amzn2.x86_64       8/9
  Installing : httpd-2.4.46-1.amzn2.x86_64            9/9
  Verifying  : apr-1.6.3-5.amzn2.0.2.x86_64           1/9
  Verifying  : apr-util-bdb-1.6.1-5.amzn2.0.2.x86_64   2/9
  Verifying  : httpd-tools-2.4.46-1.amzn2.x86_64       3/9
  Verifying  : mod_http2-1.15.14-2.amzn2.x86_64       4/9
  Verifying  : apr-1.6.3-5.amzn2.0.2.x86_64            5/9
  Verifying  : mailcap-2.1.43-2.amzn2.noarch           6/9
  Verifying  : generic-logos-httd-18.0.0-4.amzn2.noarch 7/9
  Verifying  : httpd-2.4.46-1.amzn2.x86_64            8/9
  Verifying  : httpd-2.4.46-1.amzn2.x86_64            9/9
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-httd.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 0:2.1.43-2.amzn2  mod_http2.x86_64 0:1.15.14-2.amzn2

Complete!
[root@ip-10-0-1-145 ec2-user]# cd /var/www/html
[root@ip-10-0-1-145 html]# ls
[root@ip-10-0-1-145 html]# vi index.html
i-0a466f051bb756b34 (bastion_server)
Public IP: 3.91.174.236 Private IP: 10.0.0.225
```



STARTED HTTPD SERVICE AND CHECKING STATUS IN WEB-SERVER1:

```
"index.html" [New] 1L, 27C written
[root@ip-10-0-1-145 html]# service httpd start
Redirecting to /bin/systemctl start httpd.service
[root@ip-10-0-1-145 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 Wed 2020-11-04 17:05:54 UTC; 9s ago
       Docs: man:httpd.service(8)
 Main PID: 12579 (httpd)
    Status: "Processing requests..."
   Group: /system.slice/httpd.service
           └─12579 /usr/sbin/httpd -DFOREGROUND
           ├─12580 /usr/sbin/httpd -DFOREGROUND
           ├─12581 /usr/sbin/httpd -DFOREGROUND
           ├─12582 /usr/sbin/httpd -DFOREGROUND
           ├─12583 /usr/sbin/httpd -DFOREGROUND
           └─12584 /usr/sbin/httpd -DFOREGROUND
Nov 04 17:05:54 ip-10-0-1-145.ec2.internal systemd[1]: Starting The Apache HTTP Server...
Nov 04 17:05:54 ip-10-0-1-145.ec2.internal systemd[1]: Started The Apache HTTP Server.
[root@ip-10-0-1-145 html]#
```



```
"index.html" [New] 1L, 27C written
[root@ip-10-0-1-145 html]# service httpd start
Redirecting to /bin/systemctl start httpd.service
[root@ip-10-0-1-145 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 Wed 2020-11-04 17:05:54 UTC; 9s ago
       Docs: man:httpd.service(8)
     Main PID: 12579 (httpd)
      Status: "Processing requests..."
      CGrou...
```

Nov 04 17:05:54 ip-10-0-1-145.ec2.internal systemd[1]: Starting The Apache HTTP Server...
Nov 04 17:05:54 ip-10-0-1-145.ec2.internal systemd[1]: Started The Apache HTTP Server.

```
[root@ip-10-0-1-145 html]# exit
exit
[ec2-user@ip-10-0-1-145 ~]$ exit
logout
Connection to 10.0.1.145 closed.
[root@ip-10-0-225 ec2-user]#
```

SSH INTO WEB-SERVER2:

```
"index.html" [New] 1L, 27C written
[root@ip-10-0-1-145 html]# service httpd start
Redirecting to /bin/systemctl start httpd.service
[root@ip-10-0-1-145 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 Wed 2020-11-04 17:05:54 UTC; 9s ago
       Docs: man:httpd.service(8)
     Main PID: 12579 (httpd)
      Status: "Processing requests..."
```

Nov 04 17:05:54 ip-10-0-1-145.ec2.internal systemd[1]: Starting The Apache HTTP Server...
Nov 04 17:05:54 ip-10-0-1-145.ec2.internal systemd[1]: Started The Apache HTTP Server.

```
[root@ip-10-0-1-145 html]# exit
exit
[ec2-user@ip-10-0-1-145 ~]$ exit
logout
Connection to 10.0.1.145 closed.
[root@ip-10-0-225 ec2-user]# ssh -i web-serverkey.pem ec2-user@10.0.1.125
```

```

"index.html" [New] ll, 27C written
[root@ip-10-0-1-145 ~]# service httpd start
Redirecting to /bin/systemctl start httpd.service
[root@ip-10-0-1-145 ~]# systemctl status httpd.service
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)
     Docs: man:httpd.service(8)
   Main PID: 12579 (httpd)
      Status: "Processing requests..."
    CPU usage: 0.000 CPU seconds total
      Tasks: 0 (limit: 4900)
     Group: /system.slice/httpd.service
         ├─12579 /usr/sbin/httpd -DFOREGROUND
         ├─12580 /usr/sbin/httpd -DFOREGROUND
         ├─12581 /usr/sbin/httpd -DFOREGROUND
         ├─12582 /usr/sbin/httpd -DFOREGROUND
         └─12583 /usr/sbin/httpd -DFOREGROUND
           ├─12584 /usr/sbin/httpd -DFOREGROUND

Nov 04 17:05:54 ip-10-0-1-145.ec2.internal systemd[1]: Starting The Apache HTTP Server...
Nov 04 17:05:54 ip-10-0-1-145.ec2.internal systemd[1]: Started The Apache HTTP Server.
[root@ip-10-0-1-145 ~]# exit
exit
[ec2-user@ip-10-0-1-145 ~]$ exit
logout
Connection to 10.0.1.145 closed.
[root@ip-10-0-0-225 ec2-user]# ssh -i web-serverkey.pem ec2-user@10.0.1.125
The authenticity of host '10.0.1.125' (10.0.1.125) can't be established.
ECDSA key fingerprint is SHA256:SislyAOiv0ke3lcylo8WVcE4fnJLvoZkmxs+mEjLM.
ECDSA key fingerprint is MD5:61:6b:ee:bc:df:1f:4b:6a:19:02:0c:3b:2f:8a:a5:f6.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.1.125' (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-10-0-1-125 ~]$ 

```

i-0a466f051bb756b34 (bastion_server)

Public IPs: 3.91.174.296 Private IPs: 10.0.0.225

UPDATING THE WEB-SERVER2:

```

"index.html" [New] ll, 27C written
[root@ip-10-0-1-145 ~]# service httpd start
Redirecting to /bin/systemctl start httpd.service
[root@ip-10-0-1-145 ~]# systemctl status httpd.service
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)
     Docs: man:httpd.service(8)
   Main PID: 12579 (httpd)
      Status: "Processing requests..."
    CPU usage: 0.000 CPU seconds total
      Tasks: 0 (limit: 4900)
     Group: /system.slice/httpd.service
         ├─12579 /usr/sbin/httpd -DFOREGROUND
         ├─12580 /usr/sbin/httpd -DFOREGROUND
         ├─12581 /usr/sbin/httpd -DFOREGROUND
         ├─12582 /usr/sbin/httpd -DFOREGROUND
         └─12583 /usr/sbin/httpd -DFOREGROUND
           ├─12584 /usr/sbin/httpd -DFOREGROUND

Nov 04 17:05:54 ip-10-0-1-145.ec2.internal systemd[1]: Starting The Apache HTTP Server...
Nov 04 17:05:54 ip-10-0-1-145.ec2.internal systemd[1]: Started The Apache HTTP Server.
[root@ip-10-0-1-145 ~]# exit
exit
[ec2-user@ip-10-0-1-145 ~]$ exit
logout
Connection to 10.0.1.145 closed.
[root@ip-10-0-0-225 ec2-user]# ssh -i web-serverkey.pem ec2-user@10.0.1.125
The authenticity of host '10.0.1.125' (10.0.1.125) can't be established.
ECDSA key fingerprint is SHA256:SislyAOiv0ke3lcylo8WVcE4fnJLvoZkmxs+mEjLM.
ECDSA key fingerprint is MD5:61:6b:ee:bc:df:1f:4b:6a:19:02:0c:3b:2f:8a:a5:f6.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.1.125' (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-10-0-1-125 ~]$ sudo su
[root@ip-10-0-1-125 ec2-user]# yum update -y

```

i-0a466f051bb756b34 (bastion_server)

Public IPs: 3.91.174.296 Private IPs: 10.0.0.225

INSTALLING HTTPD IN WEB-SERVER2:

```

Verifying : cpio-2.11-27.amzn2.x86_64
Verifying : glibc-2.26-35.amzn2.x86_64
Verifying : glibc-common-2.26-35.amzn2.x86_64
Verifying : e2fsprogs-libs-1.42.9-12.amzn2.0.2.x86_64
Verifying : unRAR-6.0-20.amzn2.x86_64
Verifying : pil-kit-trust-0.1.18.147-1.amzn2.x86_64
Verifying : libcroco-0.1.18.147-1.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.amzn2.2.2.x86_64
Verifying : q libc-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-botcore-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
  pil-kit.x86_64 0:0.23.21-2.amzn2.0.1
  python-pillow.x86_64 0:2.0.0-21.gid1c6db8.amzn2.0.1
  rpm.x86_64 0:4.11.3-40.amzn2.0.5
  rpm-plugin-systemd-inhibitit.x86_64 0:4.11.3-40.amzn2.0.5

Complete!
[root@ip-10-0-1-125 ec2-user]# yum install httpd -y

```

i-0a466f051bb756b34 (bastion_server)
Public IP: 3.91.174.236 Private IP: 10.0.0.225

CREATING “INDEX.HTML” FILE IN WEB-SERVER2:

```

(7/9): httpd-tools-2.4.46-1.amzn2.x86_64.rpm
(8/9): mailcap-2.1.41-2.amzn2.noarch.rpm
(9/9): mod_http2-1.15.14-2.amzn2.x86_64.rpm
Total
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : apr-1.6.3-5.amzn2.0.2.x86_64
  Installing : apr-util-bdb-1.6.1-5.amzn2.0.2.x86_64
  Installing : apr-util-1.6.1-5.amzn2.0.2.x86_64
  Installing : httpd-tools-2.4.46-1.amzn2.x86_64
  Installing : generic-logos-httdp-18.0.0-4.amzn2.noarch
  Installing : mailcap-2.1.41-2.amzn2.noarch
  Installing : httpd-filesystem-2.4.46-1.amzn2.noarch
  Installing : mod_http2-1.15.14-2.amzn2.x86_64
  Installing : httpd-2.4.46-1.amzn2.x86_64
  Verifying : apr-util-1.6.1-5.amzn2.0.2.x86_64
  Verifying : httpd-filesystem-2.4.46-1.amzn2.noarch
  Verifying : apr-util-bdb-1.6.1-5.amzn2.0.2.x86_64
  Verifying : apr-util-2.4.46-1.amzn2.x86_64
  Verifying : mod_http2-1.15.14-2.amzn2.x86_64
  Verifying : apr-1.6.3-5.amzn2.0.2.x86_64
  Verifying : mailcap-2.1.41-2.amzn2.noarch
  Verifying : generic-logos-httdp-18.0.0-4.amzn2.noarch
  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-httdp.noarch 0:18.0.0-4.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-10-0-1-125 ec2-user]# cd /var/www/html
[root@ip-10-0-1-125 html]# ls
[root@ip-10-0-1-125 html]# index.html
bash: index.html: command not found
[root@ip-10-0-1-125 html]# vi index.html

```

i-0a466f051bb756b34 (bastion_server)
Public IP: 3.91.174.236 Private IP: 10.0.0.225

STARTED HTTPD SERVICE AND CHECKING STATUS IN WEB-SERVER2:

```
"index.html" [New] 1L, 27C written
[root@ip-10-0-1-125 ~]# service httpd start
Redirecting to /bin/systemctl start httpd.service
[root@ip-10-0-1-125 ~]# 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 Wed 2020-11-04 17:09:40 UTC; 11s ago
       Docs: man:httpd.service(8)
    Main PID: 12571 (httpd)
   Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec: 0 B/sec"
      Group: /system.slice/httpd.service
         ├─12571 /usr/sbin/httpd -DFOREGROUND
         ├─12572 /usr/sbin/httpd -DFOREGROUND
         ├─12573 /usr/sbin/httpd -DFOREGROUND
         ├─12574 /usr/sbin/httpd -DFOREGROUND
         ├─12575 /usr/sbin/httpd -DFOREGROUND
         └─12576 /usr/sbin/httpd -DFOREGROUND

Nov 04 17:09:40 ip-10-0-1-125.ec2.internal systemd[1]: Starting The Apache HTTP Server...
Nov 04 17:09:40 ip-10-0-1-125.ec2.internal systemd[1]: Started The Apache HTTP Server.
[root@ip-10-0-1-125 ~]#
```

i-0a466f051bb756b34 (bastion_server)
Public IP: 3.91.174.236 Private IP: 10.0.0.225

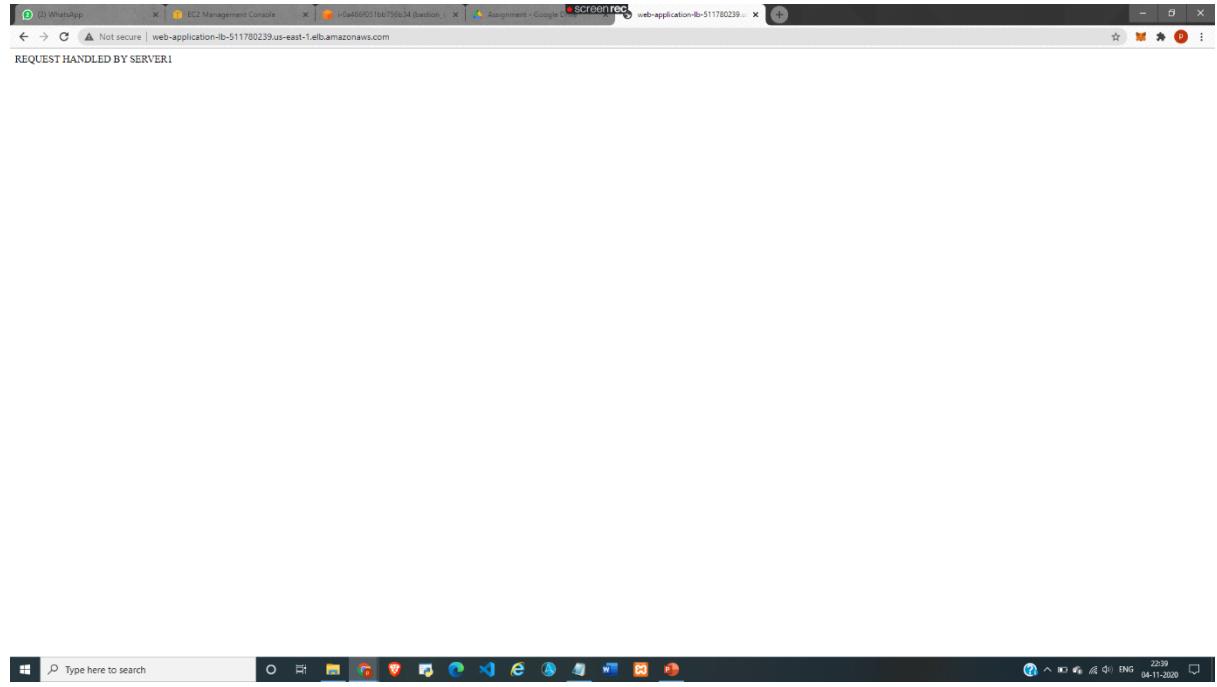
```
"index.html" [New] 1L, 27C written
[root@ip-10-0-1-125 ~]# service httpd start
Redirecting to /bin/systemctl start httpd.service
[root@ip-10-0-1-125 ~]# service httpd status
Redirecting to /bin/systemctl status httpd.service
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)
     Active: active (running) since Wed 2020-11-04 17:09:40 UTC; 11s ago
       Docs: man:httpd.service(8)
    Main PID: 12571 (httpd)
   Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec: 0 B/sec"
      Group: /system.slice/httpd.service
         ├─12571 /usr/sbin/httpd -DFOREGROUND
         ├─12572 /usr/sbin/httpd -DFOREGROUND
         ├─12573 /usr/sbin/httpd -DFOREGROUND
         ├─12574 /usr/sbin/httpd -DFOREGROUND
         ├─12575 /usr/sbin/httpd -DFOREGROUND
         └─12576 /usr/sbin/httpd -DFOREGROUND

Nov 04 17:09:40 ip-10-0-1-125.ec2.internal systemd[1]: Starting The Apache HTTP Server...
Nov 04 17:09:40 ip-10-0-1-125.ec2.internal systemd[1]: Started The Apache HTTP Server.
[root@ip-10-0-1-125 ~]# exit
exit
[ec2-user@ip-10-0-1-125 ~]$ exit
logout
Connection to 10.0.1.125 closed.
[root@ip-10-0-0-225 ec2-user]#
```

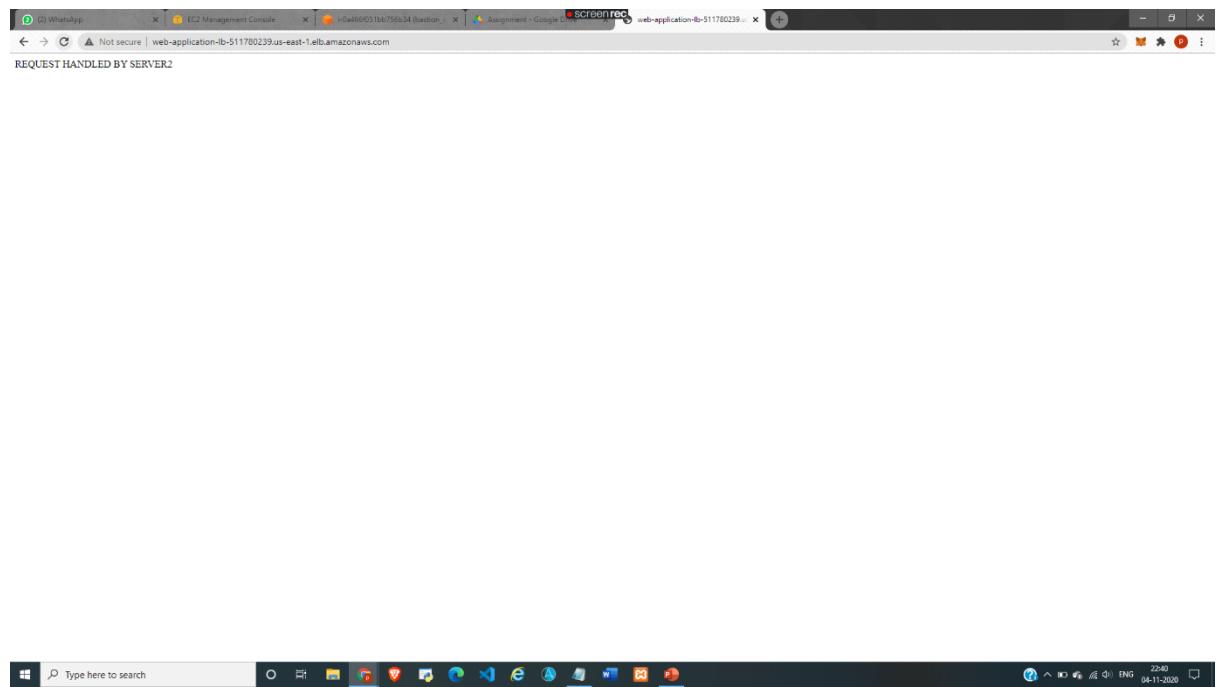
i-0a466f051bb756b34 (bastion_server)
Public IP: 3.91.174.236 Private IP: 10.0.0.225

-Copy paste load balancer DNS in browser

REQUEST FROM WEB-SERVER1:



REQUEST FROM WEB-SERVER2:



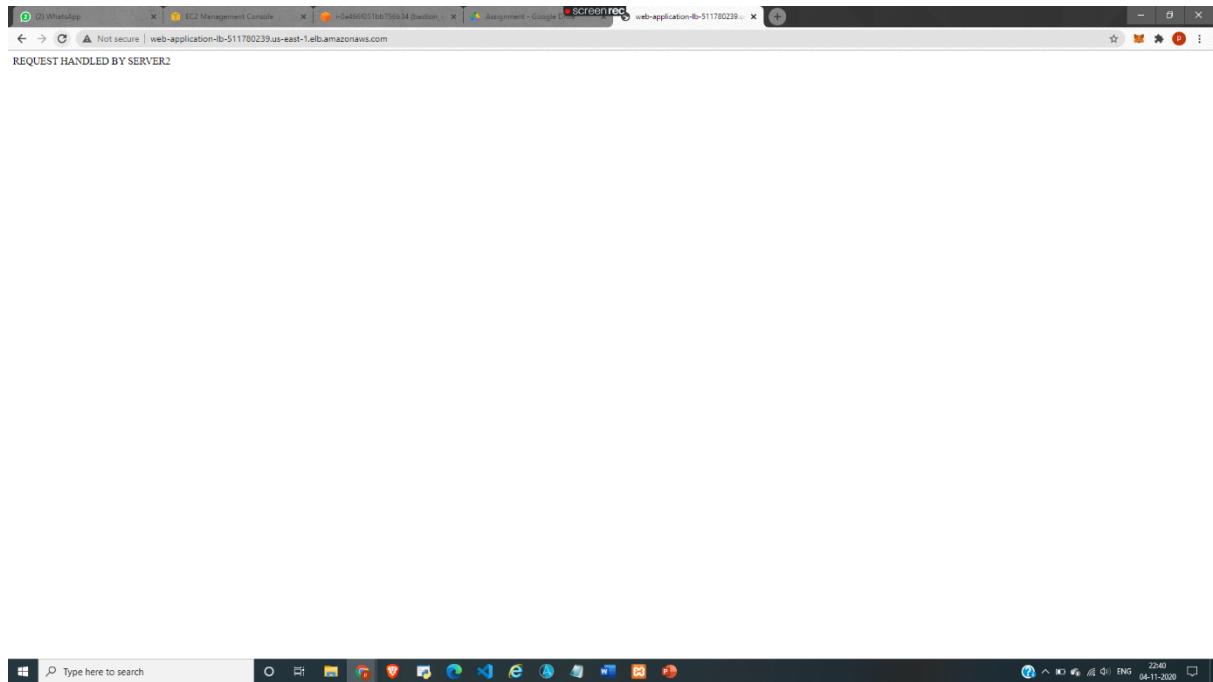
HEALTH OF WEB-SERVER BEFORE BEING STOPPED:

The screenshot shows the AWS EC2 Target Groups console. On the left, a sidebar lists various services like Launch Templates, Images, and Network & Security. The main area shows a target group named "web-app-TG". Under "Basic configuration", it specifies a target type of "instance", protocol of "HTTP: 80", and a VPC of "vpc-0f8d4d4d5509f5117". A "Load balancer" is set to "web-application-LB". Below this, the "Targets" tab is selected, showing two registered targets: "web-server1" and "web-server2", both marked as "healthy". The status details for both show a green circle and the word "healthy".

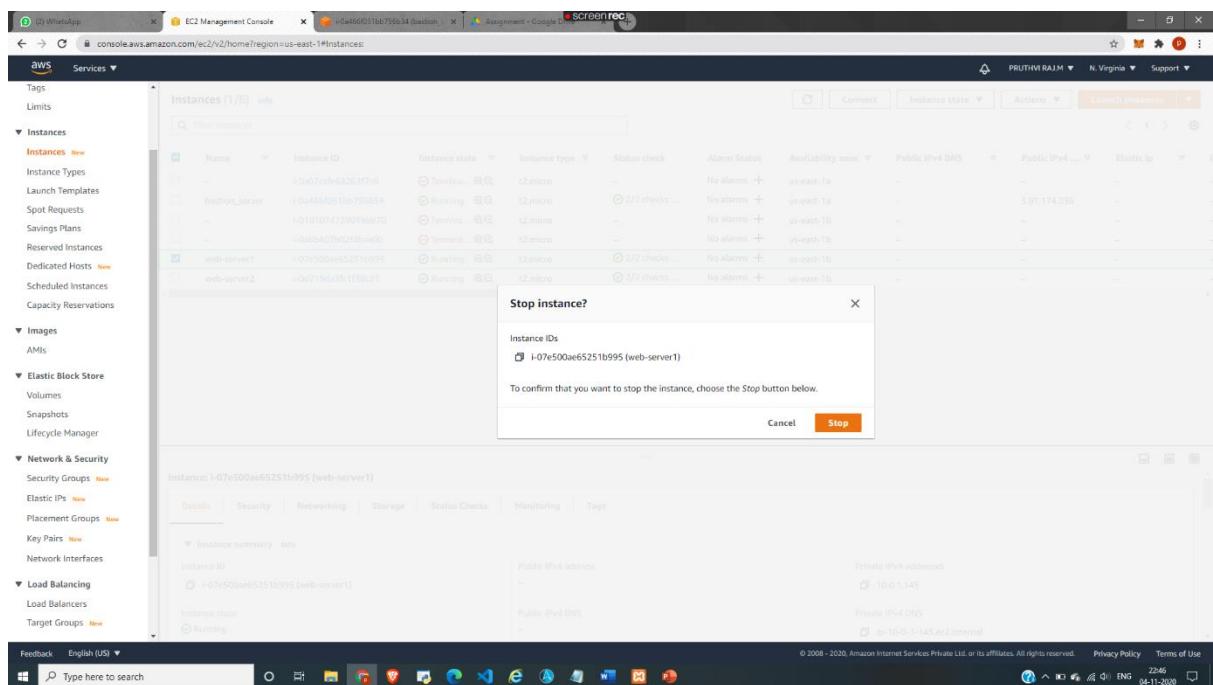
STOPPING WEB-SERVER1:

The screenshot shows the AWS EC2 Instances console. The left sidebar includes sections for Instances, Images, and Network & Security. In the main area, a table lists several instances, including "web-server1" and "web-server2". A modal dialog titled "Stop instance?" is open over instance "i-07e500ae65251b995 (web-server1)". It asks for confirmation to stop the instance. The "Cancel" button is visible, and the "Stop" button is highlighted in orange.

RESULT OF LOAD BALANCER AFTER STOPPING WEB-SERVER1:



STOPPING WEB-SERVER2 AND STARTING WEB-SERVER1:



HEALTH OF WEB-SERVER AFTER BEING STOPPED:

The screenshot shows the AWS EC2 Target Groups console. On the left, there's a navigation sidebar with sections like Instances, Images, Elastic Block Store, Network & Security, Load Balancing, and Auto Scaling. The main area shows a target group named 'web-app-TG'. Under 'Basic configuration', it shows the target type as 'instance', protocol as 'HTTP:80', and VPC as 'vpc-0f8dd4d45509f5117'. The 'Targets' tab is selected, displaying a table of registered targets:

Instance ID	Name	Port	Zone	Status	Status details
i-07e500ae65251b995	web-server1	80	us-east-1b	healthy	
i-0d219da3fc1f36c21	web-server2	80	us-east-1b	unused	Target is in the stopped state

RESULT OF LOAD BALANCER AFTER STOPPING WEB-SERVER2:

The screenshot shows a web browser window with the URL 'web-application-lb-511780239.us-east-1.elb.amazonaws.com'. The page content is 'REQUEST HANDLED BY SERVER1'. At the bottom, there's a Windows taskbar with various icons and system status information.

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.

THE END...