

Deploying a Highly Available Web Application and Bastion Host in AWS

Introduction

Bastion Host

- A **bastion host** is a **system** that is exposed to the internet.
- In terms of security, Bastion is the only server that is exposed to the internet and should be highly protective to malicious attacks.
- A **Bastion host** is also **known as a Jump Box**. It is a computer that acts like a proxy server and that allows the client machine to connect to the remote server.
- It usually resides outside the firewall.
- The Bastion server filters the incoming traffic and prevents unwanted connections entering the network thus acting as a gateway to maintain the security of bastion hosts, all unnecessary software, and daemons.

High Availability

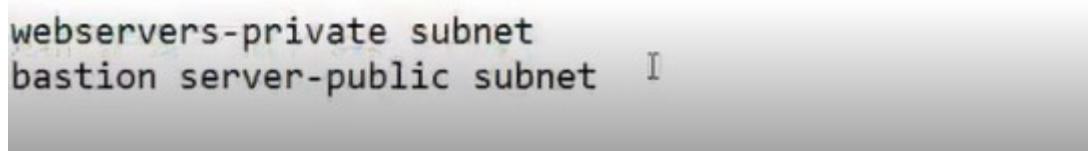
- Consider your application is running on a single EC2 instance. If the traffic to your application increases and you need further resources, we can launch multiple EC2 instances from an already running server and then **use Elastic Load Balancing** to distribute the traffic to your application among the newly-created servers.
- We can also **eliminate the Fault tolerance** in your application by placing the servers (EC2 instances) across different availability zones.
- In the event of **Failure of one Availability zone**, your application will serve or handle the **traffic from another availability zone**.
- High **Availability and fault tolerance** can be **achieved using Elastic Load balancers**.

Task Details

1. Launch a **Bastion Host instance** along with **two web application instances**, **two web application instances should be launched in the private subnet**.
2. Set up a **Load Balancer** and associate the two web instances to the Load Balancer.
3. **SSH** into the web servers **via the Bastion server**.
4. Publish a test **index.html** on both of the web servers.
5. Access the webpage using the load balancer's DNS endpoint.
6. Check the responses to see the **Load Distribution** between the 2 servers.
7. Stop or terminate one of the web servers.
8. Check the responses to see how the Load Distribution changes

Following Steps following making in this projects

- 1)vpc
- 2)public and private subnet
- 3)nat gateway deployment
- 4)Bastion host deployment
- 5)launching web servers securely in private subnet
- 6)ssh into remote servers in the private subnet
- 7)Creating customized security groups
- 8)creating and accessing a load balancer
- 9)RSA private key login procedure
- 10)changing file permissions in linux



Create VPC:

1. Log into AWS Management Console.
2. Make sure you are in the **US East (N. Virginia) us-east-1** region.

AWS Management Console

AWS services

Find Services

Recently visited services

- Cognito
- IAM
- EC2
- S3
- VPC
- S3 Glacier
- Billing
- Elastic Beanstalk
- ECR
- X-Ray

Stay connected to your AWS resources on-the-go

Explore AWS

Amazon Redshift

Run Serverless Containers with AWS Fargate

Feedback English (US) ▾ Type here to search © 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use 17:08 03/11/2020

3. Navigate to VPC under the services menu. Click on Your VPCs.

The screenshot shows the AWS VPC Management Console. In the left sidebar, under 'VIRTUAL PRIVATE CLOUD', 'Your VPCs' is selected. The main pane displays a table titled 'Your VPCs (1)'. The table has columns: Name, VPC ID, State, and IPv4 CIDR. One row is listed: Name is empty, VPC ID is 'vpc-4a0bf137', State is 'Available', and IPv4 CIDR is '172.31.0.0/16'. At the top right of the table, there is a red-bordered 'Create VPC' button. The bottom of the screen shows the Windows taskbar.

Click on Create VPC Button and add on following details,

Name Tag : **MyVPC**

IPv4 CIDR block : **10.0.0.0/16** (You can also put any other CIDR range)

IPv6 CIDR block : Select **No IPv6 CIDR Block**

Tenancy : **Default**

The screenshot shows the 'Create VPC' dialog box. The 'VPC settings' section is highlighted with a red box. It contains fields for 'Name tag - optional' (with value 'MYVPC'), 'IPv4 CIDR block' (with value '10.0.0.0/16'), and 'Tenancy' (set to 'Default'). Below the settings, there is a 'Tags' section with a table for adding key-value pairs. One tag is present: 'Key' is 'Name' and 'Value - optional' is 'MYVPC'. There is a 'Remove' button next to the tag entry. At the bottom right of the dialog is a red-bordered 'Create VPC' button.

Click on in above screen selected “Create VPC “Button.

4. The VPC is now created and showing VPC details in following screen.

VPC ID	State	DNS hostnames	DNS resolution
vpc-022b5487b6495ccda	Available	Disabled	Enabled

CIDR	Status
10.0.0.0/16	Associated

Create Public and Private Subnets:

1. Navigate to **Subnets** in the left panel of the VPC page.

Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR	Availability Zone	Availability Zone ID	Network
subnet-163f5c18	subnet-163f5c18	available	vpc-4a0bf137	172.31.64.0/20	4091	-	us-east-1f	use1-az5	us-e2
subnet-8d7b02c0	subnet-8d7b02c0	available	vpc-4a0bf137	172.31.16.0/20	4091	-	us-east-1b	use1-az4	us-e2
subnet-b1c61ee	subnet-b1c61ee	available	vpc-4a0bf137	172.31.32.0/20	4091	-	us-east-1c	use1-az6	us-e2
subnet-c6ad19e7	subnet-c6ad19e7	available	vpc-4a0bf137	172.31.80.0/20	4091	-	us-east-1a	use1-az2	us-e2
subnet-cc2987aa	subnet-cc2987aa	available	vpc-4a0bf137	172.31.0.0/20	4091	-	us-east-1d	use1-az1	us-e2
subnet-eace33db	subnet-eace33db	available	vpc-4a0bf137	172.31.48.0/20	4091	-	us-east-1e	use1-az3	us-e2

1. Let's create a Public subnet. Click on Create subnet button.

- Name tag : **MyPublicSubnet**
- VPC : **MyVPC**
- Availability Zone : **No Preference**

- o IPv4 CIDR block : 10.0.0.0/24

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	MyPublicsubnet
VPC*	vpc-022b5487b6495ccda
Availability Zone	No preference
VPC CIDRs	CIDR
	10.0.0.0/16 Status: associated
IPv4 CIDR block*	10.0.0.0/24

* Required

Create

- o Make sure add on above details click on Create button on above screen.

2. Let's enable Auto Assign public IP to Instances created within this subnet,

- Select **MyPublicSubnet**,
- Click on **Modify auto-assign IP settings**
- Auto-assign IPv4 : Check

Modify auto-assign IP settings

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.

Subnet ID: subnet-0a5bcb5c7b112f01f

Auto-assign IPv4 <input checked="" type="checkbox"/>	Enable auto-assign public IPv4 address
Auto-assign Co-IP <input type="checkbox"/>	Enable auto-assign customer-owned IPv4 address

* Required

Save



Click on save button and review below screen **Public Subnet details**

The screenshot shows the AWS VPC Dashboard. On the left, there's a sidebar with various VPC-related options like Route Tables, Internet Gateways, and Security Groups. The main area displays a table of subnets. One subnet, 'MyPublicsubnet', is selected and highlighted with a red box. The table columns include Name, Subnet ID, State, VPC, IPv4 CIDR, Available IPv4, IPv6 CIDR, and Availability Zone. Below the table, there's a detailed view of the selected subnet ('subnet-0a5bcb5c7b112f01f') with tabs for Description, Flow Logs, Route Table, Network ACL, Tags, and Sharing.

3. Let's create a private subnet. Click on Create subnet button

The screenshot shows the 'Create subnet' wizard. It has several input fields: 'Name tag' (MyPrivateSubnet), 'VPC' (vpc-022b5487b6495ccda), 'Availability Zone' (No preference), 'VPC CIDRs' (CIDR: 10.0.0.0/16, Status: associated), and 'IPv4 CIDR block' (10.0.1.0/24). The 'Create' button is at the bottom right. A note at the bottom left says '* Required'.

The screenshot shows the Windows taskbar. It includes a search bar, pinned icons for File Explorer, Edge, File Manager, and File History, and various system status icons like battery level and signal strength.

Added in above screen Private Subnet details

- Name tag : **MyPrivateSubnet**
- VPC : **MyVPC**
- Availability Zone : **No Preference**
- IPv4 CIDR block : **10.0.1.0/24**

Click on save button and review below screen **Private Subnet details**

The screenshot shows the AWS VPC Subnets page. On the left sidebar, under the 'VPC' section, 'Subnets' is highlighted. The main table lists subnets with columns for Name, Subnet ID, State, VPC, IPv4 CIDR, Available IPv4, and IPv6 CIDR. Two subnets are selected: 'MyPublicSubnet' and 'MyPrivateSubnet'. The 'MyPrivateSubnet' row is highlighted with a red box.

Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR
MyPublicSubnet	subnet-0a5bcb5c7b11201f	available	vpc-022b5487b6495ccda MYVPC	10.0.0.0/24	251	-
MyPrivateSubnet	subnet-0f6afb0bb75a17b05	available	vpc-022b5487b6495ccda MYVPC	10.0.1.0/24	251	-
	subnet-468d093d	available	vpc-77b3541c	172.31.16.0/20	4091	-
	subnet-9c4c3ed0	available	vpc-77b3541c	172.31.0.0/20	4091	-
	subnet-e9h56a6	available	vpc-77b3541c	172.31.32.0/20	4091	-

4. Now, two subnets are created

Create Internet Gateway:

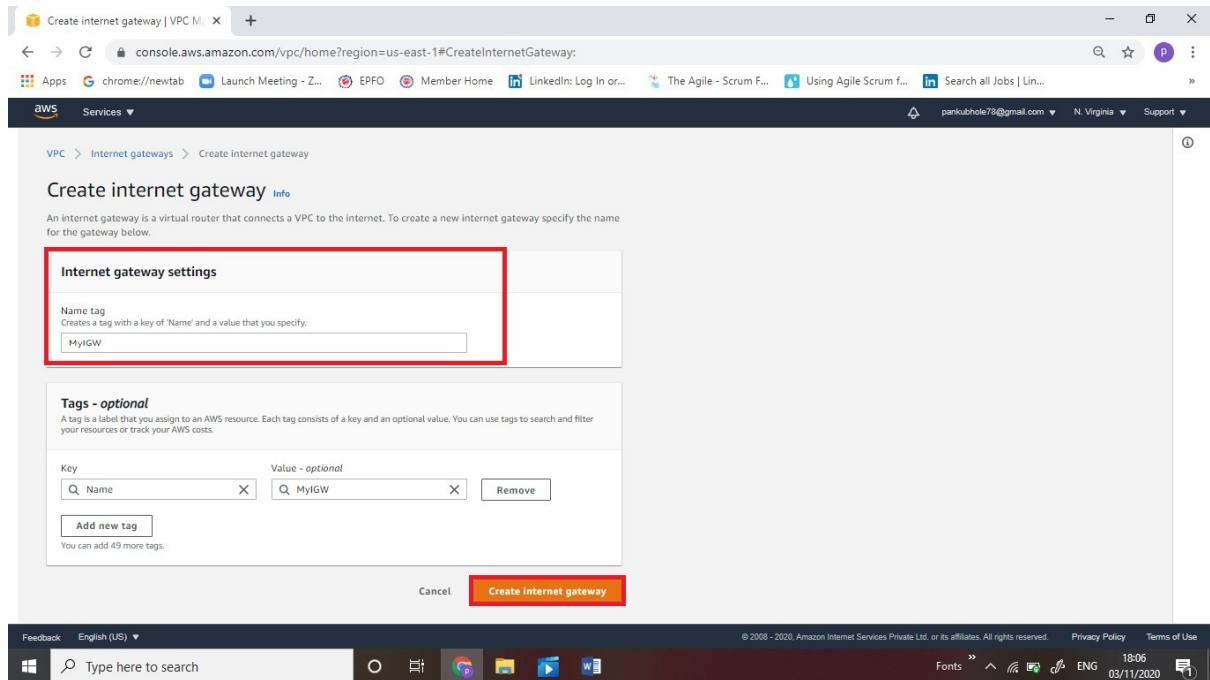
1. Navigate **Internet Gateways** in the left panel of the VPC page.

The screenshot shows the AWS Internet Gateways page. On the left sidebar, 'Internet Gateways' is highlighted. The main table lists one internet gateway with columns for Name, Internet gateway ID, State, VPC ID, and Owner. The gateway is attached to the VPC. The gateway's details are shown in a modal window below the table.

Name	Internet gateway ID	State	VPC ID	Owner
-	igw-90fffd6eb	Attached	vpc-4a0bf137	618639665310

2. Click on Create Internet Gateway button and add following gateway details.

- o Name tag : **MyIGW**
- o **Click on Create internet gateway button.**



3. An Internet Gateway is now created.

4. To attach Internet Gateway to a VPC,

- o Select the Internet Gateway *MyIGW*.
- o Click on **Actions**. Select **Attach to VPC**.
- o **VPC : MyVPC**

The screenshot shows the AWS VPC Management Console. On the left, there's a sidebar with 'Virtual Private Cloud' navigation. Under 'Internet Gateways', the 'igw-0a9316ad04cd5b449 / MyIGW' entry is selected. The main panel displays the gateway's details: Internet gateway ID (igw-0a9316ad04cd5b449), State (Detached), and VPC ID (-). To the right, the 'Actions' menu is open, with 'Attach to VPC' highlighted. Below the details, there's a 'Tags' section with one tag: Name (MyIGW).

Click on Attached VPC button.

5. Now *MyIGW* is attached to *MyVPC*.

The screenshot shows the 'Attach to VPC' dialog box. It has a 'VPC' section with instructions to attach the gateway to a VPC. A 'Available VPCs' dropdown is open, showing a single item: 'vpc-022eb545/b649bccc'. At the bottom, there's a 'Cancel' button and a prominent orange 'Attach internet gateway' button.

Create Public Route Table and Configure:

We will create a route table and attach a public subnet to it. Instances launched within this subnet will have access to the Internet.

1. Navigate to Route Table in the left panel of the VPC page.

The screenshot shows the AWS VPC Management console. On the left sidebar, under the 'Route Tables' section, the 'Route Tables' link is highlighted with a red box. The main content area displays a table of existing route tables, with two entries visible:

Name	Route Table ID	Explicit subnet association	Edge associations	Main	VPC ID	Owner
rtb-0d9c4965bad622c02	-	-	-	Yes	vpc-022b5487b6495ccda ...	618639665310
rtb-5dba1f36	-	-	-	Yes	vpc-77b3541c	618639665310

2. Click on to Create Route Table Button.

- o Name tag : ***PublicRouteTable***
- o VPC : ***MyVPC***
- o Click on Create button

The screenshot shows the 'Create route table' wizard. The 'Name tag' field is set to 'PublicRouteTable' and the 'VPC' dropdown is set to 'vpc-022b5487b6495ccda'. At the bottom right, the 'Create' button is highlighted with a red box.

3. A route table by name *PublicRouteTable* will be created.

4. To attach an Internet Gateway, select *PublicRouteTable*.
5. In the **Routes** tab below:



- o Click on Edit Route
- o On the next page, click on
- o Destination : Enter **0.0.0.0/0**
- o Target : Select **Internet Gateway**, and once the internet gateways have been created, select **MyIGW**
- o Click on Save Button.

Destination	Target	Status	Propagated
10.0.0.0/16	local	active	No
0.0.0.0/0	igw-0a9316ad04cd5b449		No

* Required Cancel **Save routes**



6. To associate the Public Subnet to the route table, Select *PublicRouteTable*.
- o Click on the Action and select **Edit Subnet Associations** tab.
 - o Click on Save Button.
 - o On the next page, select *MyPublicSubnet* from the list displayed.

The screenshot shows the AWS VPC Dashboard with the 'Route Tables' section selected. A specific route table, 'PublicRouteTable', is highlighted. The 'Routes' tab is selected, showing two entries:

Destination	Target	Status	Propagated
10.0.0.16	local	active	No
0.0.0.0	igw-a9316ad04cd5b449	active	No

Click on Save Button

The screenshot shows the 'Edit subnet associations' dialog box. It displays a table of subnets associated with the route table 'rtb-07e053c8d02140507'. One subnet, 'subnet-0a5bcb5c7b112f01f | MyPublicSubnet', is selected and highlighted.

Subnet ID	IPv4 CIDR	IPv6 CIDR	Current Route Table
subnet-0a5bcb5c7b112f01f MyPublicSubnet	10.0.0.0/24	-	Main
subnet-0f6af80bb75a17b05 MyPrivateSubnet	10.0.1.0/24	-	Main

7. Once all the configurations are completed, it should look like below .selected details are Main Route Table details.
8. Now the Instances launched within *MyPublicSubnet* will have access to the Internet.
9. As you can see, there is another existing route table already available for *MyVPC*. It is a main route table created at the time the VPC was created. We will use it while creating the **NAT Gateway**.

Name	Route Table ID	Explicit subnet association	Edge associations	Main	VPC ID
PublicRouteTable	rtb-0e053c8d02140507	subnet-0a5bcb5c7b112f01f	-	No	vpc-022b5487b6495ccda MYVPC
	rtb-0d9c4965bad622c02	-	-	Yes	vpc-022b5487b6495ccda MYVPC
	rtb-5dba1f36	-	-	Yes	vpc-77b3541c

Launching an EC2 Instance in Public Subnet

1. Make sure you are in the **Mumbai** region.
2. Navigate to the menu in the top, click on in the section.
3. Click on Launch Instance button.

4. Choose an Amazon Machine Image (AMI): Search for **Amazon Linux 2 AMI** in the search box and click on the **select** button.

Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Search for an AMI by entering a search term e.g. "Windows"

Quick Start

- My AMIs
- AWS Marketplace
- Community AMIs
- Free tier only

AMI Name	Description	Root device type	Virtualization type	ENI Enabled	Select
Amazon Linux 2 AMI (HVM, SSD Volume Type) - ami-0c306788ff2473ccb (64-bit x86) / ami-001c484a60bb07f8d (64-bit Arm)	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.	ebs	hvm	Yes	<input type="button" value="Select"/>
Red Hat Enterprise Linux 8 (HVM, SSD Volume Type) - ami-052c08d70def0ac62 (64-bit x86) / ami-0ad289a92ed067259 (64-bit Arm)	Red Hat Enterprise Linux version 8 (HVM), EBS General Purpose (SSD) Volume Type	ebs	hvm	Yes	<input type="button" value="Select"/>
SUSE Linux Enterprise Server 15 SP2 (HVM, SSD Volume Type) - ami-0d022ed4db1debd6 (64-bit x86) / ami-0b51f04425cd9683d (64-bit Arm)	SUSE Linux Enterprise Server 15 SP2 (HVM, SSD Volume Type)	ebs	hvm	Yes	<input type="button" value="Select"/>

5. Choose an Instance Type: select **Amazon Linux 2 AMI** and click on the
6. Configure Instance Details:

- o Network : **MyVPC**
- o Subnet : **MyPublicSubnet**
- o Auto-assign Public IP : *Use Subnet Setting(Enable)* - default
- o Leave all other settings as default.

7. Click on Configure Instance details button.

Step 3: Configure Instance Details

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.

Number of instances Launch into Auto Scaling Group

Purchasing option Request Spot instances

Network <input type="button" value="vpc-022b5487b6495ccda MYVPC"/>	<input type="button" value="Create new VPC"/>
Subnet <input type="button" value="subnet-0a5bcb5c7b112f01f MyPublicsubnet ap-south-1"/>	<input type="button" value="Create new subnet"/>
Auto-assign Public IP <input type="button" value="Use subnet setting (Enable)"/>	

Placement group Add instance to placement group

Capacity Reservation

Domain join directory

IAM role

CPU options Specify CPU options

8. Add Storage: No need to change anything in this step. Click on configure security button and click Add Tag button.

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		MyPublicEC2Server		<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

9. Configure Security Group:

- SSH is already available:
 - Choose Type: **SSH**
 - Source: **Anywhere**

Click on Review and Launch button.

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: launch-wizard-36

Description: launch-wizard-36 created 2020-11-04T12:27:37.953+05:30

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

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

Feedback English (US) ▾

Type here to search

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

Fonts ENG 12:28 04/11/2020

10. Review and Launch: Review all settings and click on Launch button.

11. Key Pair: Create a new key pair and click on .

- o Name KeyPair as **MyKey.pem**

12. Click on Instance details.

13. Launch Status: Your instance is now launching, Select the instance and wait for it to change status to **Running 2/2 check**.

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with options like New EC2 Experience, EC2 Dashboard, Events, Tags, Instances (with Instances selected), Images, Elastic Block Store, Network & Security, and Feedback. The main area shows a table of instances. One instance, 'MyPublicEC2Server' with ID i-05108de41b14d68b1, is selected and highlighted with a red box. Its details are shown in the 'Details' tab, including its Public IPv4 address (13.127.90.47) which is also highlighted with a red box. Other tabs like Security, Networking, Storage, Status Checks, Monitoring, and Tags are also visible.

14. Note down the Public IP address of *MyPublicEC2Server*:**13.127.90.47**.

Launching an EC2 Instance in Private Subnet:

1. Click on Launch Instance button.
2. Choose an Amazon Machine Image (AMI): Search for Amazon Linux 2 AMI in the search box and click on the select button.
3. Choose an Instance Type: select Amazon Linux 2 AMI and click on the



4. Configure Instance Details:

- o Network : MyVPC
- o Subnet : MyPrivateSubnet
- o Auto-assign Public IP : Use Subnet Setting(Enable) - default
- o Leave all other settings as default.

5. Click on Configure Instance details button.

The screenshot shows the AWS Launch Instance Wizard at Step 3: Configure Instance Details. The 'Auto-assign Public IP' dropdown is highlighted with a red box. Other visible fields include Network (vpc-022b5487b6495ccda | MYVPC), Subnet (subnet-0f6afb0bb75a17b05 | MyPrivateSubnet | ap-south-1), and IAM role (None). Buttons at the bottom include 'Cancel', 'Previous', 'Review and Launch' (highlighted in blue), and 'Next: Add Storage'.

- Add Storage: No need to change anything in this step. Click on configure security button and click Add Tag button.

The screenshot shows the AWS Launch Instance Wizard at Step 5: Add Tags. A tag named 'Name' with the value 'MyPrivateEC2Server' is selected. Other visible fields include 'Key' (Name) and 'Value' (MyPrivateEC2Server). Buttons at the bottom include 'Cancel', 'Previous', 'Review and Launch' (highlighted in blue), and 'Next: Configure Security Group'.

- Configure Security Group: this step same as above instance Publicserver.

- Review and Launch: Review all settings and click on Launch button.

- Key Pair: Create a new key pair and click on.

O Name KeyPair as MyKey.pem

- Click on Instance details

- Launch Status: Your instance is now launching, Select the instance and wait for it to change status to **Running 2/2 check**.

12. Note down the Private IP address of MyPrivateEC2Server:10.0.1.152.

13. Two servers are now launched and ready.

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

1. Using the public IP address, SSH into *MyPublicEC2Server*.
 2. Please follow the steps in [SSH into EC2 Instance](#).
 3. Switch to root user: **sudo –su**

```
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-10-0-0-71 ~]$ sudo su
```

i-05108de41b14d68b1 (MyPublicEC2Server)

Public IPs: 13.127.90.47 Private IPs: 10.0.0.71

4. Run the updates using the following command:

- **yum -y update**

```
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-10-0-0-71 ~]$ sudo su
[root@ip-10-0-0-71 ec2-user]# yum -y update
```

i-05108de41b14d68b1 (MyPublicEC2Server)

Public IPs: 13.127.90.47 Private IPs: 10.0.0.71



Verifying	Installed	Updated	Complete!
glibc-all-langpacks-2.26-35.amzn2.x86_64	kernel.x86_64 0:4.14.200-155.322.amzn2	aws-cfn-bootstrap.noarch 0:1.4-34.amzn2	awscli.noarch 0:1.18.147-1.amzn2.0.1
e2fsprogs-1.42.9-12.amzn2.0.2.x86_64	aws-ssm-agent.x86_64 0:3.0.161.0-1.amzn2	cpio.x86_64 0:2.11-28.amzn2	e2fsprogs.x86_64 0:1.42.9-19.amzn2
lvm2-utils-libs-5.5.64-1.amzn2.x86_64	bash.x86_64 0:4.2.46-34.amzn2	ec2-net-utils.noarch 0:1.4-3.amzn2	ec2-utils.noarch 0:1.2-3.amzn2
Verifying : mariadb-libs-5.5.64-1.amzn2.noarch	expat.x86_64 0:2.1.0-12.amzn2	glibc.x86_64 0:2.26-37.amzn2	glibc-all-langpacks.x86_64 0:2.26-37.amzn2
Verifying : aws-cfn-bootstrap-1.4-32.amzn2.0.1.noarch	glibc-common.x86_64 0:2.17.31-1.amzn2.0.1.noarch	libcom_err.x86_64 0:1.42.9-19.amzn2	glibc-minimal-langpack.x86_64 0:2.26-37.amzn2
Verifying : python2-botocore-1.17.31-1.amzn2.0.1.noarch	python2-rpm-4.11.3-40.amzn2.0.4.x86_64	libcrypt.x86_64 0:2.26-37.amzn2	libcroco.x86_64 0:0.6.12-6.amzn2
Verifying : libxml2-2.9.1-6.amzn2.4.1.x86_64	python2-boto-1.17.31-1.amzn2.0.1.noarch	libdb-openssl.x86_64 0:2.15.13-8.amzn2	libss.x86_64 0:1.42.9-19.amzn2
Verifying : ec2-utils-1.2-1.amzn2.noarch	rpm-build-libs.x86_64 0:4.11.3-40.amzn2.0.5	libtiff.x86_64 0:4.0.3-35.amzn2	libxml2.x86_64 0:2.9.1-6.amzn2.5.1
Installed:	kernel.x86_64 0:4.14.200-155.322.amzn2	mariaDB-libs.x86_64 1:5.5.68-1.amzn2	openldap.x86_64 0:2.4.44-22.amzn2
Updated:	amazon-ssm-agent.x86_64 0:3.0.161.0-1.amzn2	p11-kit.x86_64 0:0.23.21-2.amzn2.0.1	pam.x86_64 0:1.1.8-23.amzn2.0.1
	bash.x86_64 0:4.2.46-34.amzn2	python2-boto-core.noarch 0:1.18.6-1.amzn2.0.1	python2-rpm.x86_64 0:4.11.3-40.amzn2.0.5
	e2fsprogs-libs.x86_64 0:1.42.9-19.amzn2	rpm-plugin-systemd-inhibit.x86_64 0:4.11.3-40.amzn2.0.5	rpm-libs.x86_64 0:4.11.3-40.amzn2.0.5
	expat.x86_64 0:2.1.0-12.amzn2	unzip.x86_64 0:6.0-21.amzn2	
	glibc-common.x86_64 0:2.17.31-1.amzn2.0.1.noarch		
	python2-boto-1.17.31-1.amzn2.0.1.noarch		
	rpm-build-libs.x86_64 0:4.11.3-40.amzn2.0.5		
	mariaDB-libs.x86_64 1:5.5.68-1.amzn2		
	p11-kit-trust.x86_64 0:0.23.21-2.amzn2.0.1		
	python2-boto-core.noarch 0:1.18.6-1.amzn2.0.1		
	rpm-plugin-systemd-inhibit.x86_64 0:4.11.3-40.amzn2.0.5		
	unzip.x86_64 0:6.0-21.amzn2		

i-05108de41b14d68b1 (MyPublicEC2Server)

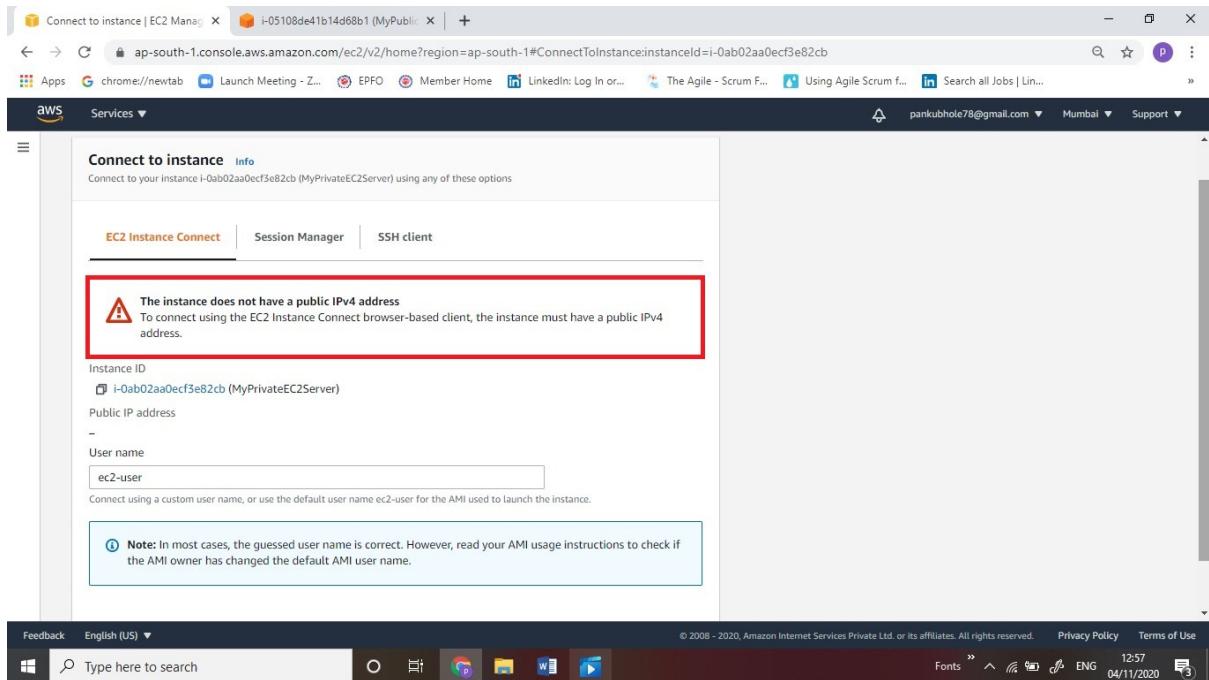
Public IPs: 13.127.90.47 Private IPs: 10.0.0.71



5. Since the Internet Gateway *MyIGW* is connected to *MyPublicSubnet*, updates will be completed successfully.

6. Let's SSH into ***MyPrivateEC2Server*** from ***MyPublicEC2Server***.

7. In order to SSH into ***MyPrivateEC2Server***, first we need to create the PEM file in the public EC2 ie, ***MyPrivateEC2Server*** and copy the data from our ***MyKey.pem*** in the local machine.



When connect the privateEC2 server then system is showing on above screen selected error message.

8. We need *MyKey.pem* inorder to SSH. We shall copy key details from the *MyKey.pem* from your local machine (which was downloaded earlier while launching EC2 instances).

9. To create the *MyKey.pem* in ***MyPublicEC2Server***, run

- **nano MyKey.pem**

10. In the editor, copy and paste the key that looks similar to the example below

```

Instances | EC2 Management Con... i-05108de41b14d68b1 (MyPublic) +
← → C ap-south-1.console.aws.amazon.com/ec2/v2/connect/ec2-user/i-05108de41b14d68b1
Apps G chrome://newtab Launch Meeting - Z... EPO Member Home LinkedIn: Log In or... The Agile - Scrum F... Using Agile Scrum f... Search all Jobs | Lin...
Services ▾
Connect to instance Info
Connect to your instance i-0ab02aa0ecf3e82cb (MyPrivateEC2Server) using any of these options

EC2 Instance Connect Session Manager SSH client

⚠ The instance does not have a public IPv4 address
To connect using the EC2 Instance Connect browser-based client, the instance must have a public IPv4 address

Instance ID
i-0ab02aa0ecf3e82cb (MyPrivateEC2Server)
Public IP address -
User name
ec2-user
Connect using a custom user name, or use the default user name ec2-user for the AMI used to launch the instance.

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.

Feedback English (US) ▾
Type here to search Fonts 12:57 ENG 04/11/2020

```

i-05108de41b14d68b1 (MyPublicEC2Server)

Public IPs: 13.127.90.47 Private IPs: 10.0.0.71



11. Update Permissions for the *MyKey.pem*

- **chmod 400 MyKey.pem**

12. Use the Private IP address 10.0.1.152 to SSH.

- **ssh -i MyKey.pem ec2-user@10.0.1.152**
- **Note:** Incase if this message shows **Are you sure you want to continue connecting (yes/no)?** : Enter yes

13. Switch to root user

- **Sudo su**

14. Run the updates using the following command:

- **yum -y update**

15. Since no internet access is provided for EC2 instances in a private subnet, you will not be able to get updates. After some time, it will fail with the following message.

```
Verifying : glibc-all-langpacks-2.26-35.amzn2.x86_64 72/79
Verifying : e2fsprogs-1.42.9-12.amzn2.0.2.x86_64 73/79
Verifying : mariadb-libs-5.5.64-1.amzn2.x86_64 74/79
Verifying : aws-cfn-bootstrap-1.4-32.amzn2.0.1.noarch 75/79
Verifying : python2-boto3core-1.17.31-1.amzn2.0.1.noarch 76/79
Verifying : python2-rpm-4.11.3-40.amzn2.0.4.x86_64 77/79
Verifying : libxml2-2.9.1-6.amzn2.4.1.x86_64 78/79
Verifying : ec2-utils-1.2-1.amzn2.noarch 79/79

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
  libcroco.x86_64 0:0.6.12-6.amzn2
  libcss.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-10-0-1-152 ec2-user]#
```

i-05108de41b14d68b1 (MyPublicEC2Server)
Public IPs: 13.127.90.47 Private IPs: 10.0.0.71

16. You can see that the updates have been completed successfully in the terminal.

17. This shows that **MyPrivateEC2Server** has internet access.

18. Use **exit** command to close the private server connection.

Steps to Create Bastion Server:

1. Make sure you are in the **N.Virginia** Region.
2. Navigate to EC2 by clicking on the menu at the top, then click on in the section and click on Launch instance button.
3. Choose the first Amazon Machine Image (AMI): **Amazon Linux 2 AMI (HVM), SSD Volume Type** click on the **Select** button
4. Instance Type : **t2.micro**

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
t2	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/> t2	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
	t2.small	1	2	EBS only	-	Low to Moderate	Yes
	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
	t2.large	2	8	EBS only	-	Low to Moderate	Yes
	t2.xlarge	4	16	EBS only	-	Moderate	Yes
	t2.2xlarge	8	32	FRS only	-	Moderate	Yes

Cancel Previous Review and Launch Next: Configure Instance Details

5. In the **Configure Instance Details**, leave all fields with the default values and then click on configure instance details button.

Step 3: Configure Instance Details

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.

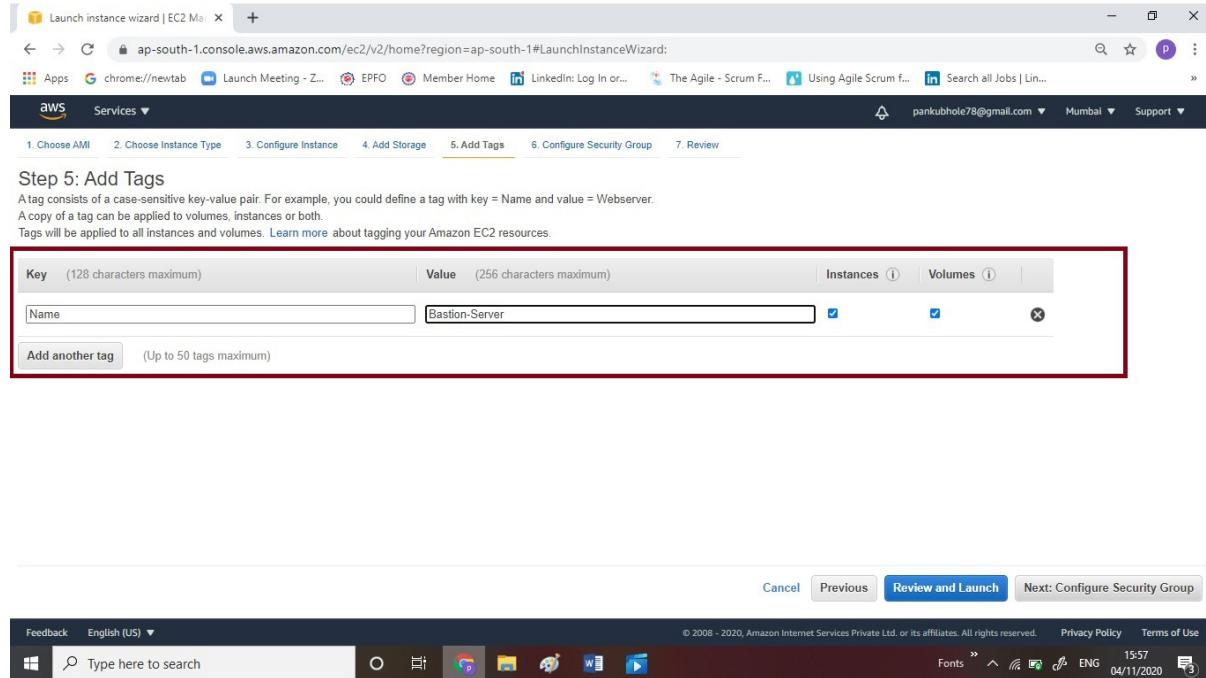
Number of instances	1	Launch into Auto Scaling Group
Purchasing option	<input type="checkbox"/> Request Spot instances	
Network	vpc-77b3541c (default)	<input type="checkbox"/> Create new VPC
Subnet	No preference (default subnet in any Availability Zone)	<input type="checkbox"/> Create new subnet
Auto-assign Public IP	Use subnet setting (Enable)	
Placement group	<input type="checkbox"/> Add instance to placement group	
Capacity Reservation	Open	
Domain join directory	No directory	<input type="checkbox"/> Create new directory
IAM role	None	<input type="checkbox"/> Create new IAM role
CPU options	<input type="checkbox"/> Specify CPU options	

Cancel Previous Review and Launch Next: Add Storage

6. No need to change anything in this step, click on Add storage details.

7. **Add Tags:** Click on

- Key : **Name**
- Value : **Bastion-Server**



The screenshot shows the 'Add Tags' step of the EC2 Launch Instance Wizard. It displays a table where a single tag is being defined. The 'Key' column contains 'Name' and the 'Value' column contains 'Bastion-Server'. Below the table, there's a link to 'Add another tag'. At the bottom of the page, there are navigation buttons: 'Cancel', 'Previous', 'Review and Launch' (which is highlighted in blue), and 'Next: Configure Security Group'.

8. **Configure Security Group:**

- Assign a security group: Choose to **Create a new security group**
- Security group name: **Bastion-SG**
 - Description: **Security group for Bastion-server**
 - To add **SSH**:
 - Choose Type : **SSH**
- Source : **Custom(Allow specific IP address) - 0.0.0.0/0**

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="0.0.0.0/0"/>	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**

10. Review and Launch: Review all settings and click on .

11. Key Pair: Create a new key pair named **bastionkey**, click on Launch button.

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-0e30678ff2473ccb
 Free tier eligible
 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

Cancel Previous **Launch**

12. Navigate to **Instances** and wait for 1-2 minutes (until the Bastion-server's status changes from pending to running).

The screenshot shows the AWS EC2 Instances page. On the left sidebar, under 'Instances', 'Instances' is selected. In the main content area, there are three instances listed:

Name	Instance ID	Instance state	Instance type	Status check	Alarm Status	Availability zone
MyPublicEC2Server	i-05108de41b14d68b1	Stopped	t2.micro	-	No alarms	ap-south-1a
MyPrivateEC2Server	i-0ab02aa0ecf3e82cb	Stopped	t2.micro	-	No alarms	ap-south-1a
Bastion-Server	i-0666f94950da905ac	Running	t2.micro	2/2 checks ...	No alarms	ap-south-1a

In the bottom panel, the details for the Bastion-Server are shown. The 'Public IPv4 address' field is highlighted with a red box and contains the value 13.232.156.97 | open address.

- **Bastion-server Public IP: 13.232.156.97.**

Creating a Security Group for the Load Balancer:

1. Navigate to the Ec2 Dashboard, scroll down to in left menu and click on Create Load Balancer button

The screenshot shows the AWS EC2 Management Console. On the left sidebar, under 'Load Balancing', 'Load Balancers' is selected. In the main content area, there is a message: 'You do not have any load balancers in this region.'

2. Configure the security group as follows:

Security group name: **LoadBalancer-SG**

- Description: **Security group for the Load balancer**
- VPC: **Leave as the default**
- Click on Create Security group and add the port as follows:
 - Type: **HTTP**
 - Make sure, Protocol is **TCP** and **Port range** is **80**
 - Source: Custom and enter **0.0.0.0/0**

The screenshot shows the AWS EC2 Management Console with the 'Security Groups' page open. The left sidebar shows various services like Elastic Block Store, Network & Security, Load Balancing, and Auto Scaling. Under 'Network & Security', 'Security Groups' is selected. The main area displays a table of existing security groups, including 'load-balancer-wizard-1' and 'Bastion-SG'. At the top right of the table is a red-bordered button labeled 'Create security group'. Below the table, there's a search bar and some navigation controls.

The screenshot shows the AWS EC2 Management Console with the 'Security Groups' page open. The left sidebar shows various services like Elastic Block Store, Network & Security, Load Balancing, and Auto Scaling. Under 'Network & Security', 'Security Groups' is selected. The main area displays a table of existing security groups, including 'load-balancer-wizard-1' and 'Bastion-SG'. A new security group named 'LoadBalancer-SG' has been added to the list, highlighted with a blue selection box. Below the table, there's a detailed view of the selected security group, showing its ID as 'sg-0a4c6f7870453dd39' and its name as 'LoadBalancer-SG'. It also lists other security groups like 'load-balancer-wizard-2', 'load-balancer-wizard-3', and 'default'. At the bottom of the screen, there are tabs for 'Details', 'Inbound rules', 'Outbound rules', and 'Tags'.

3. Leave everything by default in Outbound rules and Tags - optional

4. Click on save button.
5. The security group for the load balancer will be created see as above screen.

Steps to create Web-servers:

1. Click on Launch **instance button**.
2. Choose the first Amazon Machine Image (AMI): **Amazon Linux 2 AMI (HVM), SSD Volume Type** click on the **Select** button.

Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Search for an AMI by entering a search term e.g. "Windows"

Quick Start

My AMIs	Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0c306788ff2473ccb (64-bit x86) / ami-001c484a60bb07f8d (64-bit Arm)	Select
AWS Marketplace	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.	<input checked="" type="radio"/> 64-bit (x86) <input type="radio"/> 64-bit (Arm)
Community AMIs	Root device type: ebs Virtualization type: hvm ENA Enabled: Yes	
<input checked="" type="checkbox"/> Free tier only	Red Hat Enterprise Linux 8 (HVM), SSD Volume Type - ami-052c08d70def0ac62 (64-bit x86) / ami-0ad289a92ed067259 (64-bit Arm)	Select
	Red Hat Enterprise Linux version 8 (HVM), EBS General Purpose (SSD) Volume Type	<input checked="" type="radio"/> 64-bit (x86) <input type="radio"/> 64-bit (Arm)
	Root device type: ebs Virtualization type: hvm ENA Enabled: Yes	
	SUSE Linux Enterprise Server 15 SP2 (HVM), SSD Volume Type - ami-0d0522ed4db1debd6 (64-bit x86) / ami-0b51f04425cd9683d (64-bit Arm)	Select
	SUSE Linux Enterprise Server 15 SP2 (HVM), EBS General Purpose (SSD) Volume Type	<input checked="" type="radio"/> 64-bit (x86)
	Root device type: ebs Virtualization type: hvm ENA Enabled: Yes	

3. Instance Type : t2.micro

Step 2: Choose an Instance Type

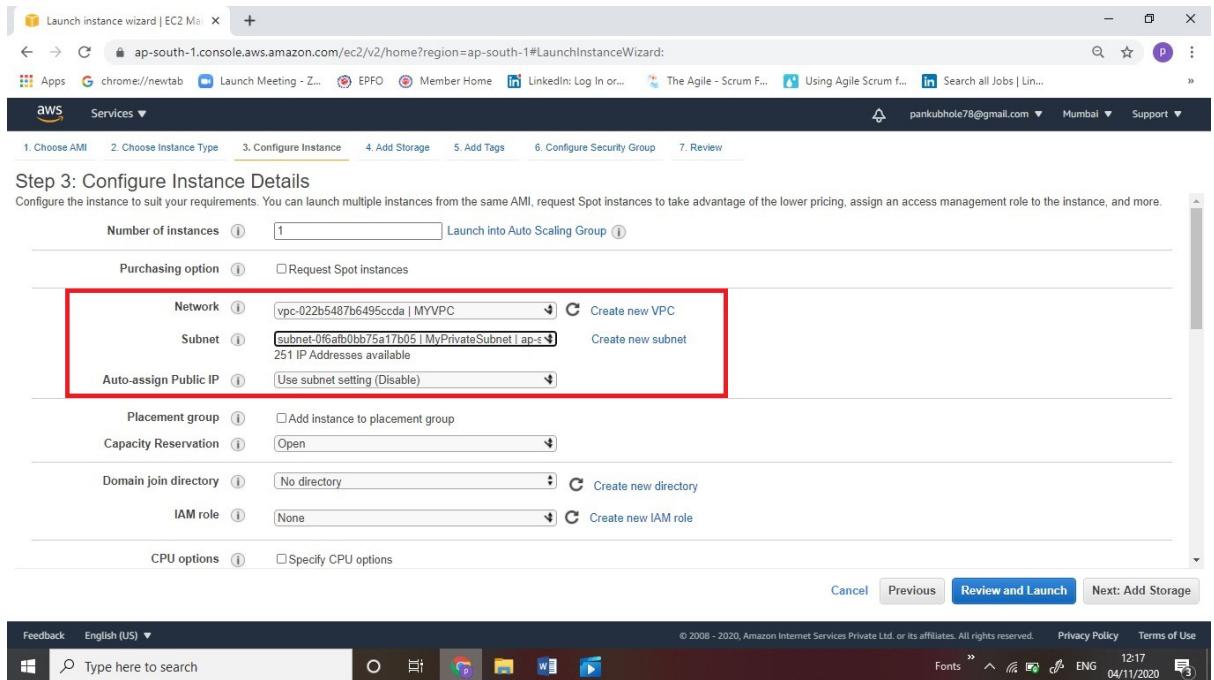
Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Filter by: All instance families ▾ Current generation ▾ Show/Hide Columns

Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
t2	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
	t2.small	1	2	EBS only	-	Low to Moderate	Yes
	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
	t2.large	2	8	EBS only	-	Low to Moderate	Yes
	t2.xlarge	4	16	EBS only	-	Moderate	Yes
	t2.2xlarge	8	32	EBS only	-	Moderate	Yes

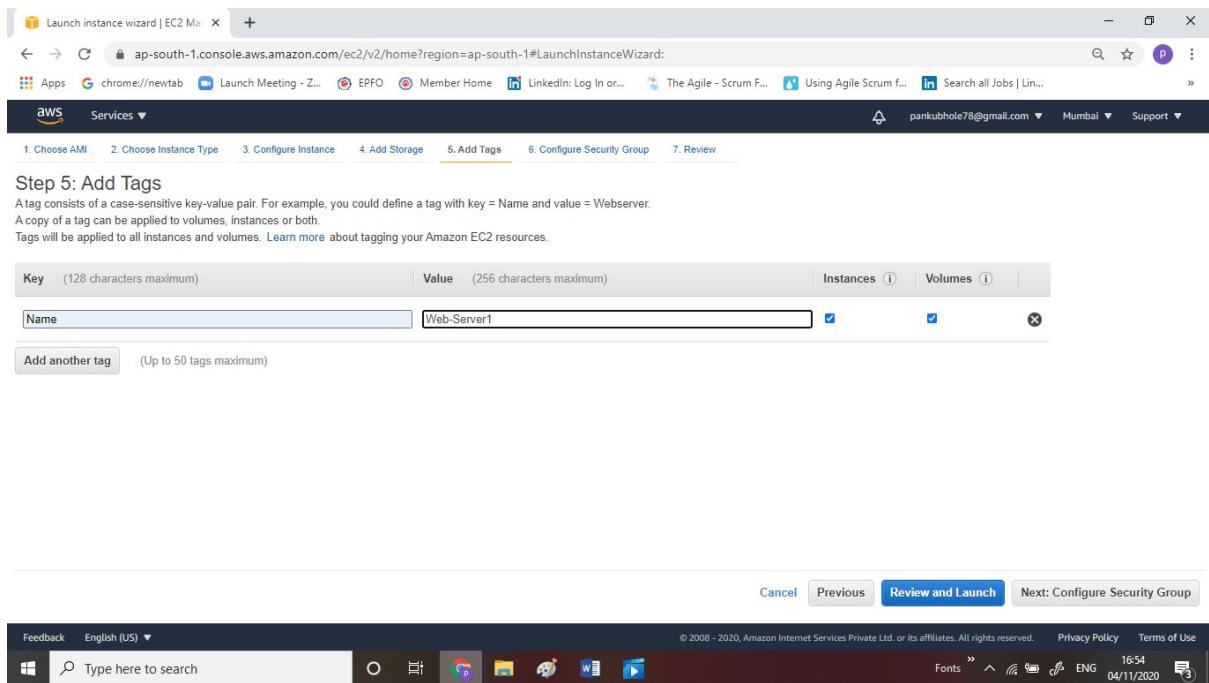
Cancel Previous Review and Launch Next: Configure Instance Details

4. In Configure Instance Details scroll down to Subnet and choose the Private subnet as shown below:



- Leave other fields default and click on Add Storage button.

6. Add Storage: No need to change anything in this step, click on Configure Security Group.



7. Add Tags: Click on

- Key : **Name**
- Value : **Web-server-1**
- Click on : **Configure Security Group**

8. 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 sg-018f3dbba895ff60e	e.g. SSH for Admin Desktop
HTTP	TCP	80	Custom sg-0a4c6f7870453dd39	e.g. SSH for Admin Desktop

Add Rule

Create New Security Group details as follow:

- Name: **web-server-SG**
- Description: **Security group for web servers**
- On port 22, we choose the **Bastion-SG security group as its source to allow SSH connection to web servers from only the bastion server** by restricting the public SSH connection. Type bastion in source and select the **Bastion-SG**.
- On port 80, choose the **Load Balancer-SG as its source** to serve the traffic coming through the load balancer. Type Load in source and select **Load Balancer-SG**.
- To add **SSH**:
 - Choose Type : **SSH**
 - Source: **Bastion-SG**
- To add **HTTP**:
 - Choose Type: **HTTP**
 - Source: **Load Balancer-SG**
- After that, click on Review and Launch.

9. **Key Pair:** Create a new key pair named **web-server key**, click download button and the **key will be downloaded to your local system**. After that click on .
10. After a few minutes, you will see the new instance named **web-server-1** running along with the **Bastion-server** created in the earlier step.

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with options like New EC2 Experience, EC2 Dashboard, Instances (selected), Images, Elastic Block Store, Network & Security, and more. The main area displays a table of instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm Status	Availability zone	Pu
MyPublicEC2Server	i-05108de41b14d68b1	Stopped	t2.micro	-	No alarms +	ap-south-1a	-
MyPrivateEC2Server	i-0ab02aa0ecf3e82cb	Stopped	t2.micro	-	No alarms +	ap-south-1a	-
Bastion-Server	i-0666f94950da905ac	Running	t2.micro	2/2 checks ...	No alarms +	ap-south-1a	ec
Web-Server1	i-04a83782fb9d07c9	Termina...	t2.micro	-	No alarms +	ap-south-1a	-
Web-Server1	i-067608ba8ef8d6455	Running	t2.micro	2/2 checks ...	No alarms +	ap-south-1b	ec

The instance details for **Web-Server1** are shown in a modal window, also highlighted with a red box:

Instance: i-067608ba8ef8d6455 (Web-Server1)

- Details** tab is selected.
- Instance summary** section shows:

Instance ID: i-067608ba8ef8d6455 (Web-Server1)	Public IPv4 address: 65.0.125.179 open address	Private IPv4 addresses: 172.31.5.191
Instance state: Running	Public IPv4 DNS: ec2-65-0-125-179.ap-south-1.compute.amazonaws.com open address	Private IPv4 DNS: ip-172-31-5-191.ap-south-1.compute.internal

12. Repeat the above steps from Step1 to create **Web-server-2**. You need 2 instances for this lab.

13. On **Add Tags** page, Click on **Add Tag** button, and enter below details.

Name: **Web-server-2**

The screenshot shows the AWS Launch Instance Wizard - Step 5: Add Tags. The page has a header with tabs: 1. Choose AMI, 2. Choose Instance Type, 3. Configure Instance, 4. Add Storage, 5. Add Tags (selected), 6. Configure Security Group, 7. Review.

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.

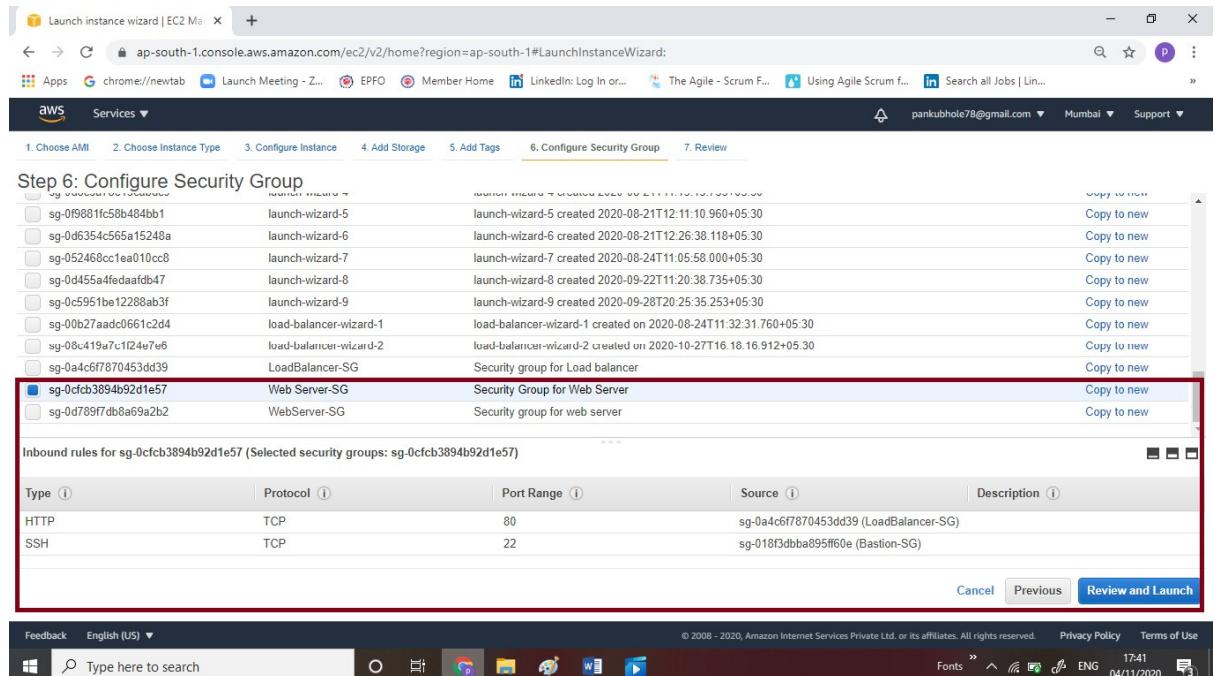
A copy of a tag can be applied to volumes, instances or both.

Tags will be applied to all instances and volumes. Learn more about tagging your Amazon EC2 resources.

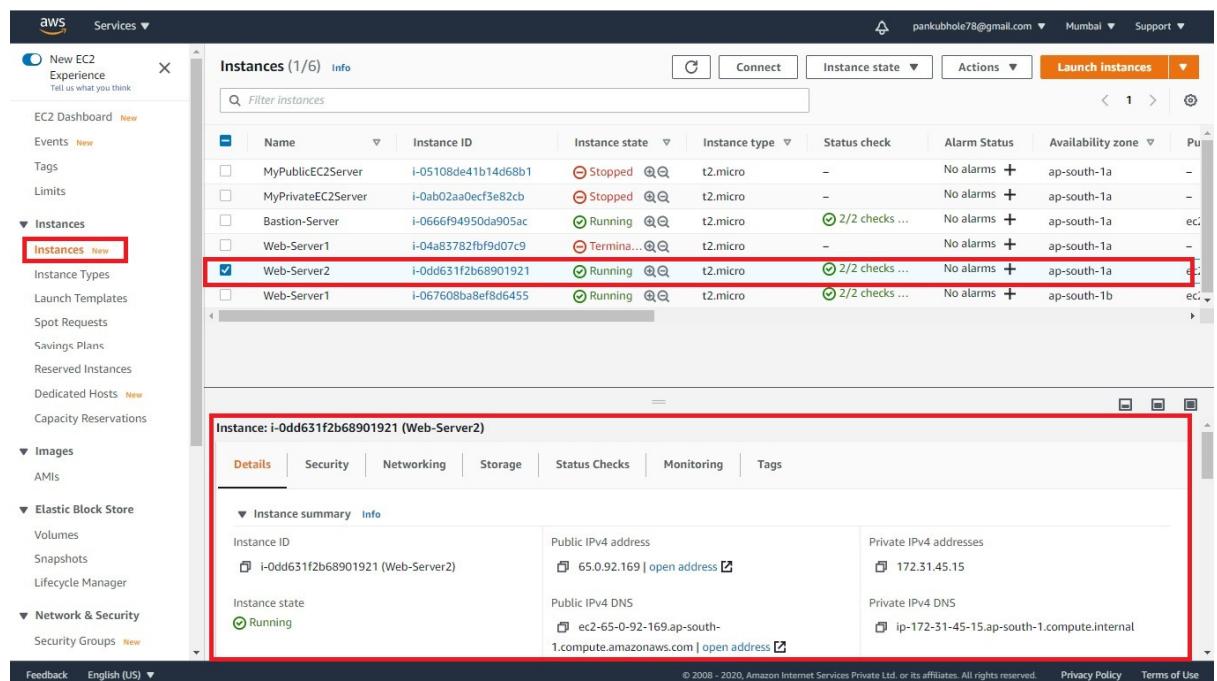
Key (128 characters maximum) | Value (256 characters maximum) | Instances | Volumes

Name	Web-Server2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Add another tag	(Up to 50 tags maximum)		

14. In, Security Group section, by selecting existing security group **web-server-SG**.



15. Click on **Review and Launch** and then Click on **Launch** and select a key pair, **web-server key**, and **Launch**. **Web server 2**



16. Now you will see three servers running namely **Bastion-server**, **Web-server-1**, and **Web-server-2**.

- **Web-server-1 Private IP : 172.31.5.191**
- **Web-server-2 Private IP : 172.31.45.15**

Creating a load balancer:

1. In the EC2 console, navigate to in Network & Security-> Load balancer link the left side panel.

The screenshot shows the AWS EC2 Management Console. On the left, there is a navigation sidebar with several sections: Elastic Block Store, Network & Security (Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces), Load Balancing (Load Balancers, Target Groups), and Auto Scaling (Launch Configurations, Auto Scaling Groups). The 'Load Balancers' section is highlighted with a red box. At the top center, there is a large blue button labeled 'Create Load Balancer'. Below it, there is a search bar and a table with columns for Name, DNS name, State, VPC ID, Availability Zones, and Type. A message at the bottom of the table says 'You do not have any load balancers in this region.' The status bar at the bottom right shows the date and time as 04/11/2020 16:12.

2. Click on at the top left to create a new load balancer for our web servers.

The screenshot shows the 'Select load balancer type' wizard. It displays three options: Application Load Balancer, Network Load Balancer, and Classic Load Balancer. The 'Application Load Balancer' section is highlighted with a red box. It features a circular icon with 'HTTP HTTPS' and a 'Create' button. Below the icon, there is a brief description: '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 integrated with application architectures, including microservices and containers.' There is also a 'Learn more >' link. The other two sections show similar layouts with 'Create' buttons and brief descriptions. The status bar at the bottom right shows the date and time as 04/11/2020 17:54.

3. On the next screen, choose since we are testing the high availability of the web app.

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	Web Application - LB
Scheme	<input checked="" type="radio"/> internet-facing <input type="radio"/> Internal
IP address type	IPv4

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	vpc-77b3541c (172.31.0.0/16) (default)
-----	--

Cancel Next: Configure Security Settings

4. In **configure the load balancer** enter the details below:

- Name: Enter **Web-application-LB**
- Scheme: Select **Internet-facing**
- IP address type: Choose **Ipv4**
- Listener: **Default (HTTP:80)**
- **Availability Zones**
- VPC: Choose **Default**
- Availability Zones: Select **All Availability Zones**,
- Make sure you select the **Public Subnet in the Availability zone us-east-1a**.

Note: we must specify the availability zones in which your load balancer needs to be enabled, making it routing the traffic only to the targets launched in those availability zones. You must include **subnets from a minimum of two Availability zones** to make our Load balancer **Highly Available**.

5. After filling in all the details above, click on **Confirm Security Settings button**.

6. On the next page, ignore the warning and click on.

7. **Configure Security Settings:**

- Select an **existing** security group and chose the security group **Load Balancer-SG** (we created this one in the step above)

Step 3: Configure Security Groups

Security Group ID	Name	Last modified	Action
sg-0602a6ccb1fba4aa1f	launch-wizard-30	2020-10-19T14:03:28.445+05:30	Copy to new
sg-013429573798d430d	launch-wizard-31	2020-10-19T14:05:29.951+05:30	Copy to new
sg-009c750b2b306148b	launch-wizard-32	2020-10-21T11:07:33.887+05:30	Copy to new
sg-0dbe3ac789178d3e3	launch-wizard-33	2020-10-26T20:32:01.494+05:30	Copy to new
sg-02cd6ccdf5f582201c	launch-wizard-34	2020-10-26T20:34:02.561+05:30	Copy to new
sg-0d8c3a78e15cabdc9	launch-wizard-4	2020-08-21T11:15:13.735+05:30	Copy to new
sg-0f9001fc50b404bb1	launch-wizard-5	2020-00-21T12:11:10.960+05:30	Copy to new
sg-0d6354c565a15248a	launch-wizard-6	2020-08-21T12:26:38.118+05:30	Copy to new
sg-052468cc1ea010cc8	launch-wizard-7	2020-08-24T11:05:58.000+05:30	Copy to new
sg-0d455a4fedafdb47	launch-wizard-8	2020-09-22T11:20:38.735+05:30	Copy to new
sg-0c5951be1228ab3f	launch-wizard-9	2020-09-28T20:25:35.253+05:30	Copy to new
sg-00b27aad0661c2d4	load-balancer-wizard-1	2020-08-24T11:32:31.760+05:30	Copy to new
sg-08c419a7c1f24e7e6	load-balancer-wizard-2	2020-10-27T16:18:16.912+05:30	Copy to new
sg-0a4c6f7870453dd39	LoadBalancer-SG	Security group for Load balancer	Next: Configure Routing
sg-0fcfb3894b2d1e57	Web Server-SG	Security Group for Web Server	Copy to new
sg-0d789f7db8a69a2b2	WebServer-SG	Security group for web server	Copy to new

8. Configure Routing

Step 4: Configure Routing

Protocol version: HTTP
 HTTP2
 gRPC

Health checks

Protocol: HTTP
Path: /index.html

Advanced health check settings

Port: traffic port
Unhealthy threshold: 2
Healthy threshold: 5
Timeout: 5 seconds
Interval: 30 seconds
Success codes: 200

- Target Group: Select New target group (default)
 - Name : Enter **web-app-TG**
 - Target Type: Select **Instance**
 - Protocol : Choose **HTTP**
 - Port : Enter **80**

Note: The target group is used to route requests to one or more registered targets.

- Health check:
 - Protocol : **HTTP**
 - Path : **/index.html**

Note: The load balancer periodically sends pings, attempts connections, or sends requests to test the EC2 instances. These tests are called health checks.

- In the upcoming steps, we will create an **index.html** in the root directory of the Apache web servers (/var/www/html) to pass this health check.

9. Registering Targets:

- Choose the two web instances and then click on and click on.

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-0dd631f2b68901921	Web-Server2	80	running	Web Server-SG	ap-south-1a
i-067608ba8ef8d6455	Web-Server1	80	running	Web Server-SG	ap-south-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-0666f94950da905ac	Bastion-Server	running	Bastion-SG	ap-south-1a	subnet-ce9b96a6	172.31.32.0/20
i-0dd631f2b68901921	Web-Server2	running	Web Server-SG	ap-south-1a	subnet-ce9b96a6	172.31.32.0/20
i-067608ba8ef8d6455	Web-Server1	running	Web Server-SG	ap-south-1b	subnet-9c4c3ed0	172.31.0.0/20

Cancel Previous Next: Review

10. Once you reviewed the settings, click on **Review Button**.

Step 6: Review

Please review the load balancer details before continuing

Load balancer

- Name: webapplicationLB1
- Scheme: internet-facing
- Listeners: Port:80 - Protocol:HTTP
- IP address type: ipv4
- VPC: vpc-77b3541c
- Subnets: subnet-ce9b96a6, subnet-9c4c3ed0, subnet-468d093d
- Tags

Security groups

- Security groups: sg-0a4c6f7870453dd39

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: 5
- Interval: 30

Cancel Previous Create

11. You have successfully created the Application Load balancer. Please wait for 3-4 minutes to make this ALB into Active state.

Connecting to web server via Bastion (Optional, if you have pasted the commands in User data):

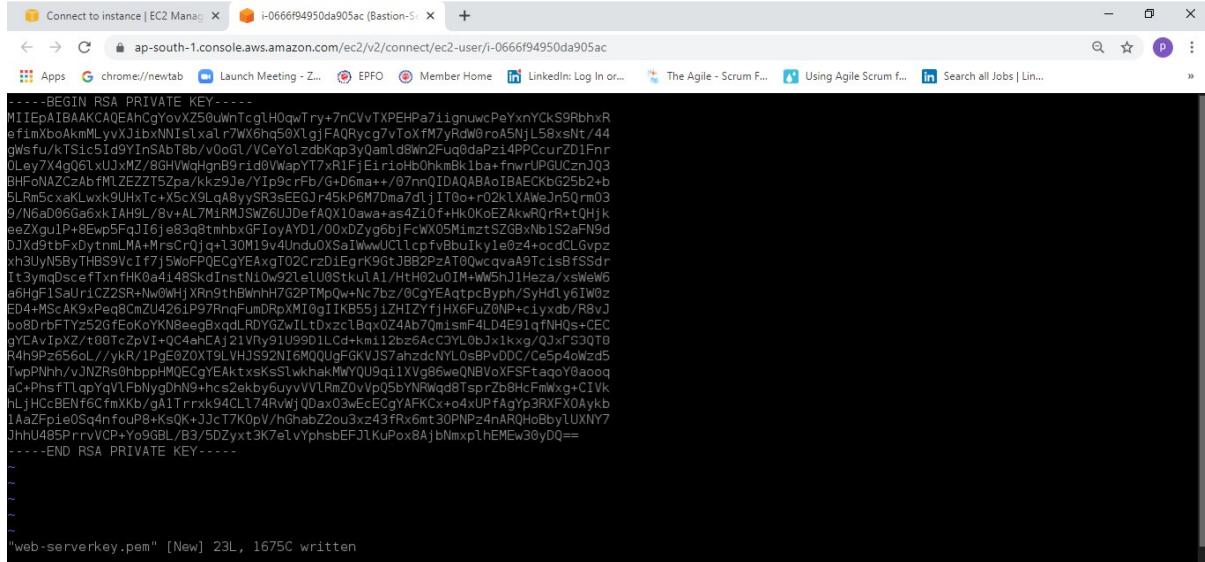
1. SSH into the Bastion server using the Bastion PEM key: **bastionkey.pem**

```
ssh-keygen -t rsa -b 2048 -f web-serverkey.pem
Generating public/private rsa key pair.
web-serverkey.pem 256, 1075C written
root@ip-172-31-41-226:~# chmod 400 web-serverkey.pem
root@ip-172-31-41-226:~#
```

Public IPs: 13.232.156.97 Private IPs: 172.31.41.226

2. To SSH into web servers via Bastion server, we need the web server key that we used to launch the previous web servers (**web-serverkey.pem**).
3. Open the **web-serverkey.pem** file on your local system and then **copy the text content**.
4. Navigate to the Bastion server and create a file named **web-serverkey.pem** using below command:

▪ **vi web-serverkey.pem**



```
-----BEGIN RSA PRIVATE KEY-----
MIIEpaIBAAKCAQEAnCgoyXZ50uwhTcgLH0qwTry+7nCVwTXPEHPa7iignuwcPeYxnYCKs9RbhxEfimxboknMlyvJ1bxN1s1xa1r7Wx6hq56XLgiFA0Rycg7v1oxfM7yRdw0roA5NjL58xsNt/44gWsfu/kTSic51d9YInSb8v/v0oG1/VceYolzdkbop3y0aml8wn2Fu0daPz14PPCcurzD1Fnrl0ley7x4q06LxJxMz/8GHWqQlnB9riid0vWapYT7xR1FjEiricb0hkmk1ba+frwrUPGUcznJ03BHFnAZCzAbfM1ZEZT5Zpa/kkz93e/YIp9cfrFb/G+06ma++/07nnQIDAQABoIBAECKbG25b2+b5LRm5cxLWx9UHxTc/X5cX9LqA8yySR3sEEGJr45kPGM7md7d1JT0+o+r02kLXAWeJh5Qrm039/7na06Ga6xkIAH5/AL7mRJM5w26UJDefAQX10awa+as4210f+hk0KE6ZAKwR0R+rt(H)keezXgu1P+8ewp5fqrI6je83o8tmhbxFIoYd1/00xDzyg6bjFcWX05MinzTSZ0BNbN1S2afN9dDJXdx9tbf+8ewp5fqrI6je83o8tmhbxFIoYd1/00xDzyg6bjFcWX05MinzTSZ0BNbN1S2afN9dXt3ympDsefTxnfHk0a1485kd1nstl1o921el1U0Stku1A1/Ith02u0TM+wW5h1Heza/xswE6a6hgf1SaUric25R-Nw0WhjXRn9thBwnhT762PTMpQw+Nc7bz/0cgyEAqtpByph/SyHdLy6IW0zED4+MsC4K9xPq9CmZu4261P97hndFumDRpKM0g1IKB5j1ZHIZYfjHX6Fu0NP+c1yxdB/Rv1bo8DrFTY252gtEokYKNeeggbxqdLRDYGzwILtDxctbqkx24Ab/qmsmf=4D4E91qTNHQs+CECgYCAv1pxZ/t80TcZpVi+qc4ahCAj21VRy91U99D1Lcd+m1.2bz=AcC3YLobJx1kg/J3x1S3QT8R4h9Pz656oL/ykR/1Pe0Z0XT91HJS92N16M0QQUgGKVJS7hzcNYL0sBPvDDC/Ces5p4wzd5TwpPNh/vJNZRs0hbpHM0EcGyeAktxsksLwkakMwYQ9q1XVgB6weQNBNv0XFSTfaqoY0aoaqoC+PhsftLqapYqVfL74RvWijQ0ax03wEcEcgYAFKCr+o4xUpfAqYp3RXFX0Aykb1AaZFpi05q4nfouP8+Ks0K+jCt7K0pV/hGhabZ2ou3x/z43fRx6mt30NPz4nARQh0BbyLUXNY7Jhnr485PrvvCP-Y9gBL/B3/5dZyx3K7e@tViphnsbEFJtKuPox8AjbnNxplhEMEv30yDQ=-----END RSA PRIVATE KEY-----
```

"web-serverkey.pem" [New] 23L, 1675C written

i-0666f94950da905ac (Bastion-Server)

Public IPs: 13.232.156.97 Private IPs: 172.31.41.226



5. Paste the content and save it by pressing **shift+colon followed by :wq!** and then enter to save your private key.
 6. Make sure you have changed the **permission of the key file to 400**. You can change the permission using below command:
 - **chmod 400 web-serverkey.pem**
 7. Now **you can log into the web servers** using the private key copied to the bastion server with the help of below commands.
 - **Note:** You **don't have a public IPs** for the web servers since we them in a private **subnet**.
 - Syntax : **ssh -i web-serverkey.pem ec2-user@<Web-server-2 private IP>**
 - Example: **ssh -i web-serverkey.pem ec2-user@ 172.31.45.15**
-
8. Now **install the apache service** using the below commands and **create a test index.html file**, which will be **used for a health check**.
 - **Installing Apache:**
 - **sudo su**
 - **yum update -y**

- yum install httpd -y
 - systemctl start httpd
 - systemctl enable httpd
- **Creating the example homepage :**
 - echo “ REQUEST HANDLING BY SERVER 1 ” > index.html
 - **?Exit from webserver to Bastion server**
 - ?To come out of 2nd instance, type **exit** command for coming out of root user, and **exit** command again for coming out of the instance.
9. Repeat steps 7 & 8 for web server 2 **with its respective private IP** , making sure to change the content of index.html to “**REQUEST HANDLING BY SERVER 2**”
 10. To come out of 1st instance, type **exit** command for coming out of root user, and **exit** command again for coming out of the instance.



```

Instances | EC2 Management Con... X i-0666f94950da905ac (Bastion-S...
← → C ap-south-1.console.aws.amazon.com/ec2/v2/connect/ec2-user/i-0666f94950da905ac + 
Search all Jobs | Lin...
Apps G chrome://newtab L Launch Meeting - Z... EPOFO Member Home LinkedIn: Log In or... The Agile - Scrum F... Using Agile Scrum f...
Installing : httpd-tools-2.4.46-1.amzn2.x86_64 4/9
Installing : generic-logos-httpd-18.0.0-4.amzn2.noarch 5/9
Installing : mailcap-2.1.41-2.amzn2.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-util-1.6.1-5.amzn2.0.2.x86_64 1/9
Verifying : httpd-filesystem-2.4.46-1.amzn2.noarch 2/9
Verifying : apr-util-bdb-1.6.1-5.amzn2.0.2.x86_64 3/9
Verifying : httpd-tools-2.4.46-1.amzn2.x86_64 4/9
Verifying : mod_http2-1.15.14-2.amzn2.x86_64 5/9
Verifying : apr-1.6.3-5.amzn2.0.2.x86_64 6/9
Verifying : mailcap-2.1.41-2.amzn2.noarch 7/9
Verifying : generic-logos-httpd-18.0.0-4.amzn2.noarch 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-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-172-31-41-226 ec2-user]# systemctl start httpd
[root@ip-172-31-41-226 ec2-user]# systemctl enable httpd
Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to /usr/lib/systemd/system/httpd.service.
[root@ip-172-31-41-226 ec2-user]#

```

i-0666f94950da905ac (Bastion-Server)
Public IPs: 13.232.156.97 Private IPs: 172.31.41.226

Checking the health of the load balancer

1. Navigate to the Load balancer and created the one Myloadbalancer for the system health checkup.

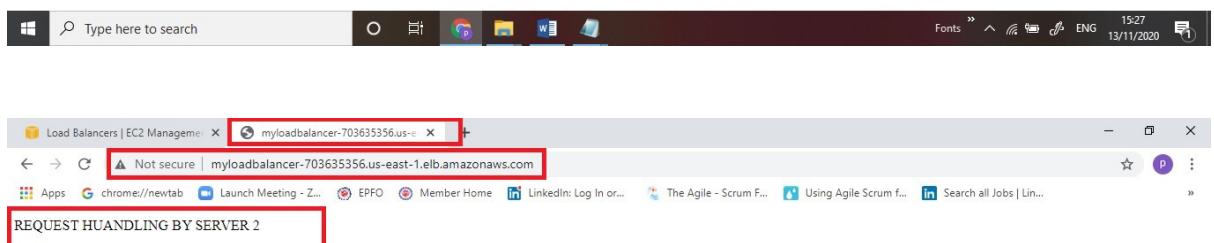
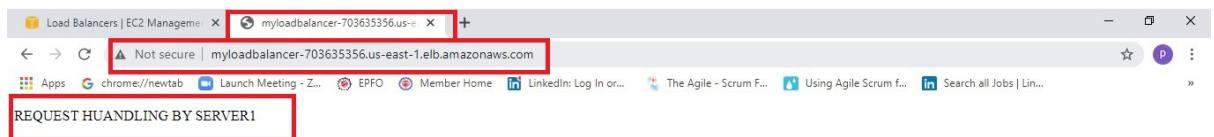
The screenshot shows the AWS Load Balancers console. On the left sidebar, under 'LOAD BALANCING', 'Load Balancers' is selected. In the main area, a table lists a single load balancer named 'MyLoadbalancer'. Below the table, a detailed view of the load balancer 'MyLoadbalancer' is shown. The 'Basic Configuration' section includes fields for Name (MyLoadbalancer), ARN (arn:aws:elasticloadbalancing:us-east-1:670363535610:loadbalancer/app/MyLoadbalancer/fe372c9eed4096b2), DNS name (MyLoadbalancer-703635356.us-east-1.elb.amazonaws.com), State (active), Type (application), and Scheme (internet-facing). The ARN and DNS name fields are highlighted with a red box.

2. Select the target group you created Mytagrget and then click on **Registered Target** and to see the **Status** of the attached targets.

The screenshot shows the AWS Target Groups console. On the left sidebar, under 'LOAD BALANCING', 'Target Groups' is selected. In the main area, a table lists a single target group named 'MyTagrget'. Below the table, a detailed view of the target group 'my tagrget' is shown. The 'Targets' tab is selected. The 'Registered targets' section displays two instances: 'Web-Server1' (Instance ID: i-02574d9c290ffdef) and 'Web-Server2' (Instance ID: i-0b3cb3530ed137c8f), both of which are healthy. The 'Availability Zones' section shows 'us-east-1b' with a target count of 2 and a healthy status. The 'Registered targets' section is highlighted with a red box.

3. It should show **Healthy** for the Load Balancer to work properly. You may need to wait for 2-5 minutes before the load balancer's status updates to "Healthy"

4. Now navigate to **load balancer** and select **DNS Link** that you created earlier.
5. Click on **DNS Link**, **copy the link** and paste it into the browser.
 - o DNS URL: MyLoadbalancer-703635356.us-east-1.elb.amazonaws.com



5. Refresh the browser a couple of times to see the requests being served from both servers. Seeing output similar to **REQUEST HANDLING BY SERVER 1 & REQUEST HANDLING BY SERVER 2** implies that load is shared between the two web servers via Application Load Balancer.
6. Now we have successfully created a **bastion server, two web servers and an Application Load balancer**, registered the targets to the load balancer and tested the working of Load Balancer.

Test case for High Availability

1. To check for high availability, we will make one of the instances unhealthy and test whether we get response from the other server.
2. If your instance is shown as Unhealthy then it's status would be one of the following:
 - stopping
 - stopped
 - terminating
 - Terminated
3. Navigate to the EC2 dashboard and select **Web-server-1**. Click on Action-> Instance state -> select and then click on **stop**.

The screenshot shows the AWS EC2 Target Groups interface. The left sidebar has links for EC2 Dashboard, Events, Tags, Limits, Instances, AMIs, and Elastic Block Store. The main area has tabs for 'Create target group' and 'Actions'. A search bar at the top says 'Filter by tags and attributes or search by keyword'. Below it is a table with columns: Name, Port, Protocol, Target type, Load Balancer, VPC ID, and Monitoring. One row is selected: 'MyTagrget' with port 80, protocol HTTP, target type instance, load balancer 'MyLoadbal...', and VPC ID 'vpc-0e7f2ad6957be8056'. Below the table is a section titled 'Target group: MyTagrget' with tabs for Description, Targets, Health checks, Monitoring, and Tags. It says: 'The load balancer starts routing requests to a newly registered target as soon as the registration process completes and the target passes the initial health checks. If demand on your targets increases, you can register additional targets. If demand on your targets decreases, you can deregister targets.' There is a blue 'Edit' button. Below this is a table titled 'Registered targets' with columns: Instance ID, Name, Port, Availability Zone, Status, and Description. It lists two targets: 'Web-Server1' (Instance ID i-02574d9c290ffdef, Port 80, us-east-1b, status unused, description 'Target is in the stopped state') and 'Web-Server2' (Instance ID i-0b3cb3530ed137c8f, Port 80, us-east-1b, status healthy, description 'This target is currently passing target group's health checks'). At the bottom are sections for 'Availability Zones' and 'Health checks'.

4. Navigate to **Target group** and click on **targets**. Here you will find the status of Web-server-1 (which should be unhealthy because it is unused).

5. Navigate to **Load balancers-->Description-->DNS name**. Copy the DNS name and paste it into your browser. You should see the response "REQUEST HANDLING BY SERVER 2" FROM WEB-SERVER-2.



6. If you refresh a few times, you will continue to see the response only from Web-server-2
 7. Repeat step 3 by stopping *Web-server-2* and starting *Web-server-1* back up. This time you should see the response "REQUEST HANDLING BY SERVER" from *Web-server-1*.

Instance ID	Name	Port	Availability Zone	Status	Description
i-02574d9c290ffdef	Web-Server1	80	us-east-1b	healthy	This target is currently passing target group's health checks.
i-0b3cb3530ed137c8f	Web-Server2	80	us-east-1b	unused	Target is in the stopped state



Completion and Conclusion

1. We have launched a Bastion server and two web-servers. We were able to SSH into the servers via Bastion Server successfully.
2. We launched an Application Load Balancer and associated our web servers with the load balancer.
3. We tested the load sharing between web servers.
4. We successfully tested the high availability of the web application by making one of the web servers unhealthy.

