

Name :- Anirudha Kurhade

Prn :- 22120150

## Assignment 3

---

**Aim:** - Deploy Web application on AWS Cloud (or any cloud)(PHP/Python/Node js any application).

---

### Description :-

#### 1. Cloud Computing Definition:

Cloud Computing refers to the delivery of computing services over the Internet. It enables organizations to access and utilize computing resources, such as servers, storage, databases, software, and analytics, without having to own, manage, and maintain the underlying infrastructure. Cloud computing enables organizations to access these resources on-demand, as they need them, providing flexible and scalable computing solutions that can be easily adapted to meet changing business needs. Additionally, cloud computing provides several benefits, such as reduced costs, improved agility, and increased innovation.

#### 2. Cloud Service Models:

Infrastructure as a Service (IaaS): IaaS is a cloud service model that provides virtualized computing resources, including storage, computing power, and networking, over the Internet. With IaaS, organizations can rent these resources, such as virtual machines and storage, as needed, and only pay for what they use. This eliminates the need for organizations to invest in and maintain their own infrastructure, allowing them to focus on their core business activities.

1. **Platform as a Service (PaaS):** PaaS is a cloud service model that provides a platform and environment for organizations to develop, run, and manage applications and services. This model typically includes the underlying infrastructure, such as servers and storage, as well as a development environment and tools to support the creation and deployment of applications. This allows organizations to focus on the development of their applications and services, without having to worry about the underlying infrastructure.

2. **Software as a Service (SaaS):** SaaS is a cloud service model that provides access to software applications over the Internet. With SaaS, organizations can subscribe to and use

software applications, such as email, customer relationship management, and project management, without having to install, configure, and maintain the software themselves.

This provides organizations with a cost-effective and convenient way to access and use software applications.

### Cloud Deployment Models:

1. **Public Cloud:** Public Cloud is a cloud deployment model where the cloud infrastructure is owned and operated by a third-party service provider and is made available to the general public over the Internet. Public Clouds are designed to provide a cost-effective and scalable solution for organizations that need to access and utilize computing resources. Public Clouds are typically operated by large companies, such as Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform.
  2. **Private Cloud:** Private Cloud is a cloud deployment model where the cloud infrastructure is owned, operated, and only used by a single organization. Private Clouds provide organizations with more control and customization options compared to public cloud services. They are typically used by organizations that have specific security, privacy, and compliance requirements.
  3. **Hybrid Cloud:** Hybrid Cloud is a cloud deployment model that combines the benefits of both public and private clouds. This model allows organizations to run their most critical applications and data in a private cloud environment, while leveraging the cost-effectiveness and scalability of public clouds for less critical workloads. Hybrid Clouds are often used to provide organizations with the flexibility and scalability they need to meet changing business needs.
  4. **Community Cloud:** Community Cloud is a cloud deployment model that is shared by several organizations and supports a specific community with shared concerns, such as security, compliance, jurisdiction, and data sovereignty. Community Clouds provide organizations with a way to collaborate and share resources, while also ensuring that their data and applications are protected.
-

## Steps Output :-

→ Login To Aws

---

### STEP 2

→ Go to E2C Instance via compute or access it directly from frequently used section

### STEP 3

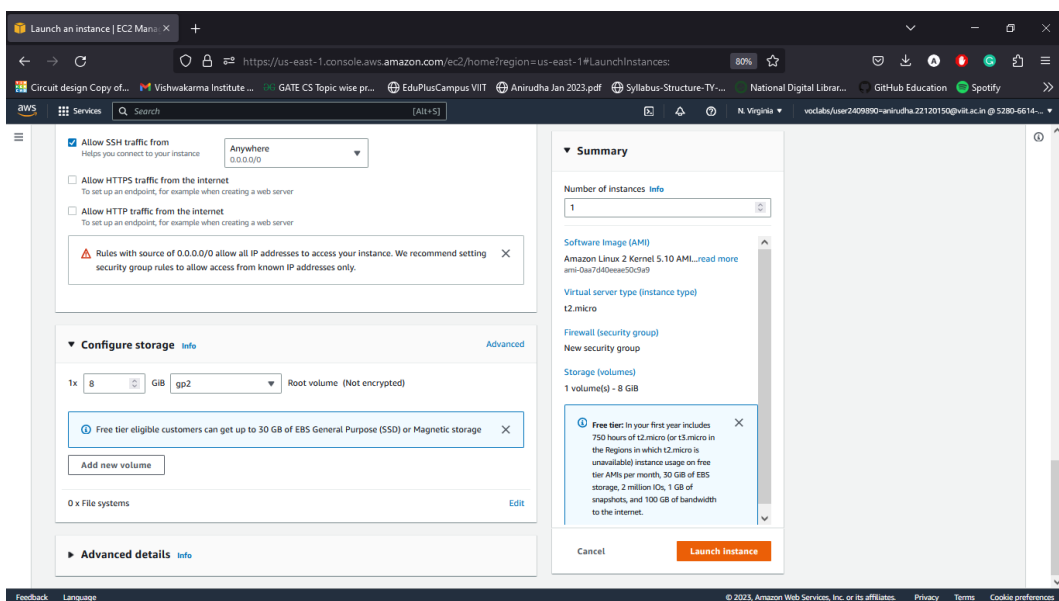
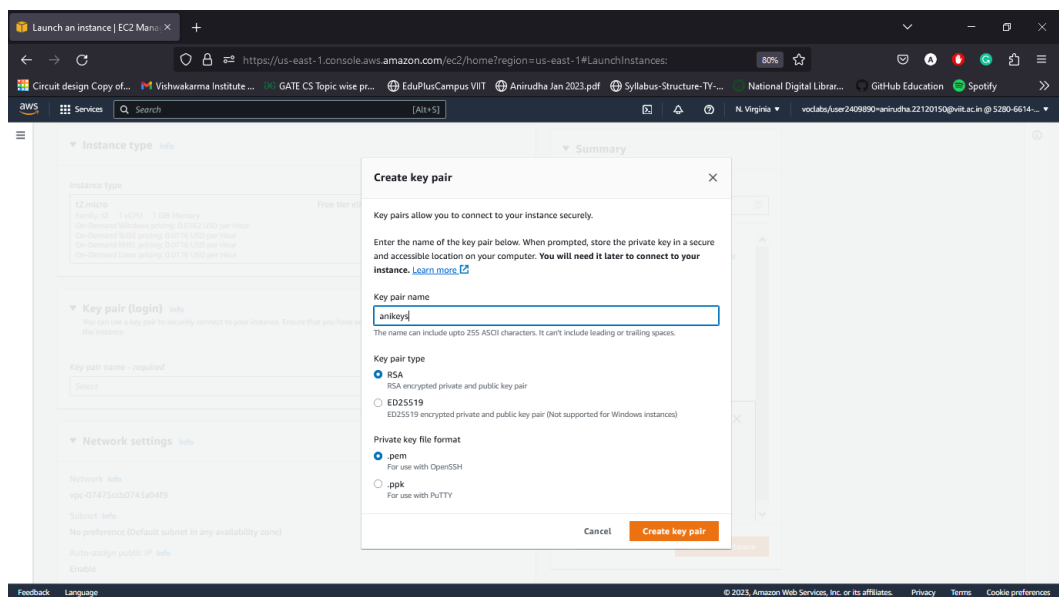
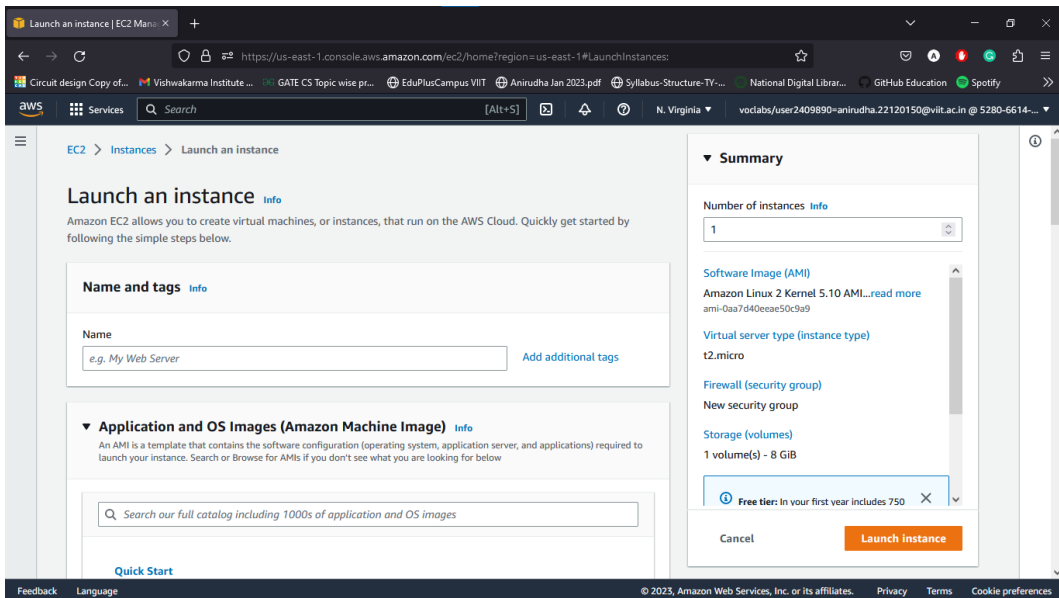
→ Configure the Instance

Name → Any name you like

OS → aws linux

Architecture → 64 bit

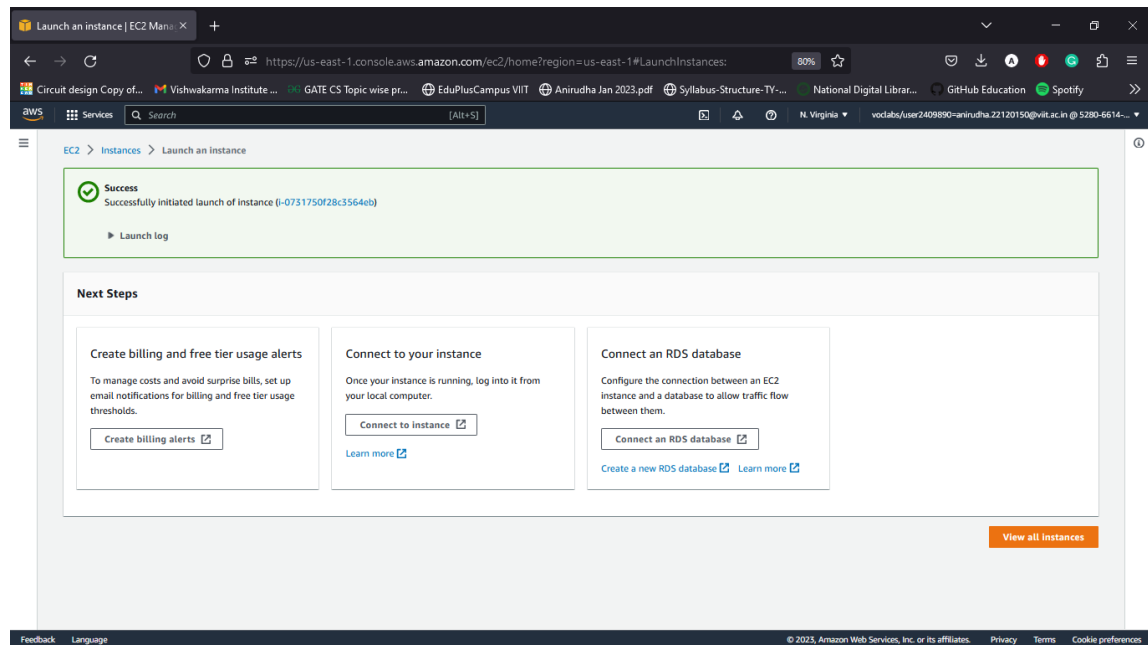
Create keys (Give any name to them and download it)



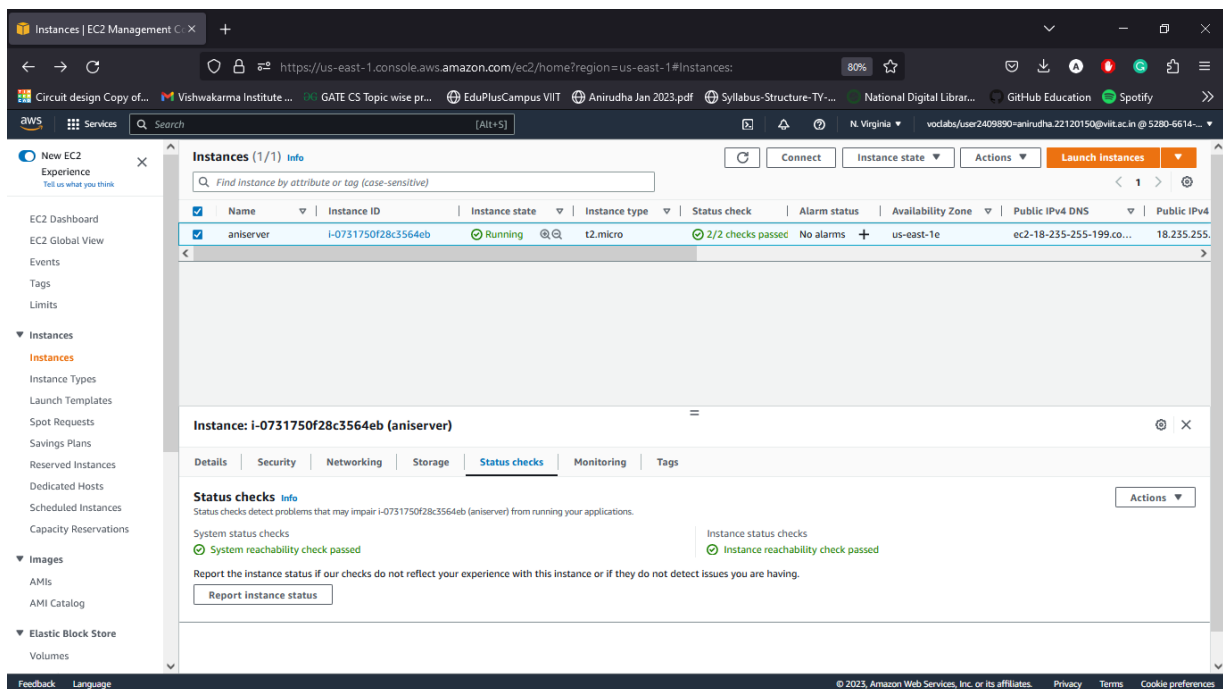
## STEP 4

→ Launch the instance

→ After clicking at right bottom launch instance button it show look like this

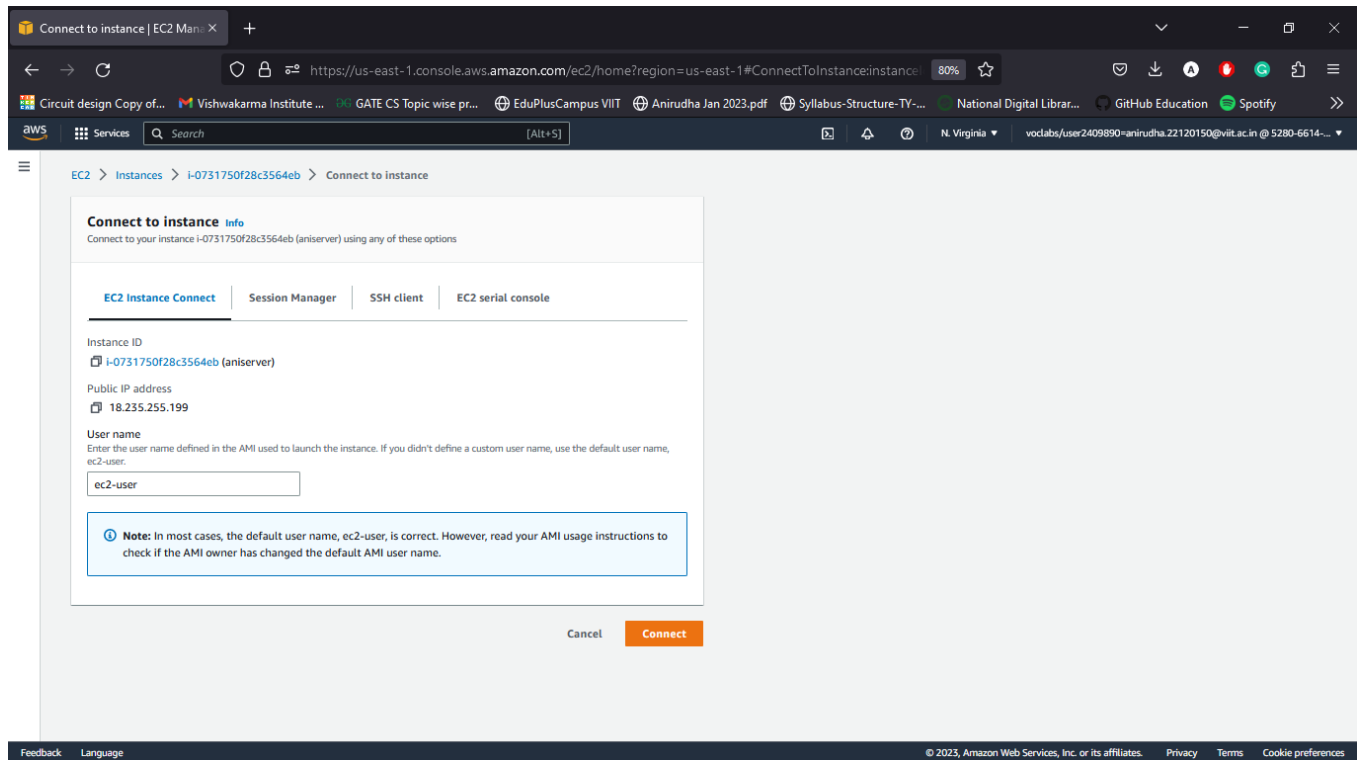


We will begin launching the instance after 2/2 status checks

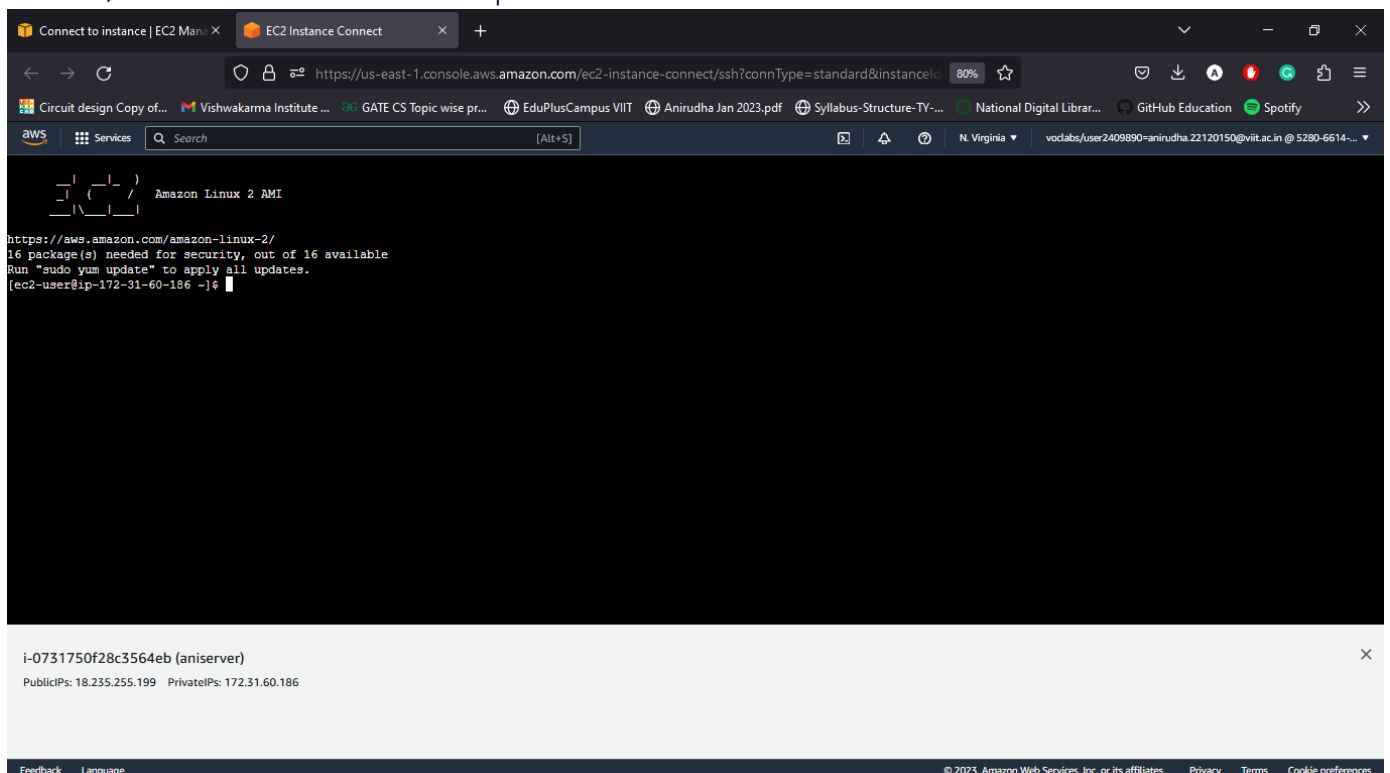


## STEP 5

→ Launch the instance (Connect to the instance)



→ This terminal window shall open



→ Put this commands in order

- sudo su -
- yum update -y

```

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

https://aws.amazon.com/amazon-linux-2/
16 package(s) needed for security, out of 16 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-60-186 ~]$ sudo su -
[root@ip-172-31-60-186 ~]# yum update -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
| 3.7 kB 00:00:00

Resolving Dependencies
--> Running transaction check
--> Package curl.x86_64 0:7.79.1-7.amzn2.0.1 will be updated
--> Package curl.x86_64 0:7.87.0-2.amzn2.0.1 will be an update
--> Package kernel.x86_64 0:5.10.165-143.735.amzn2 will be installed
--> Package kernel-tools.x86_64 0:5.10.162-141.675.amzn2 will be updated
--> Package kernel-tools.x86_64 0:5.10.165-143.735.amzn2 will be an update
--> Package libblkid.x86_64 0:2.30.2-2.amzn2.0.10 will be updated
--> Package libblkid.x86_64 0:2.30.2-2.amzn2.0.11 will be an update
--> Package libcurl.x86_64 0:7.79.1-7.amzn2.0.1 will be updated
--> Package libcurl.x86_64 0:7.87.0-2.amzn2.0.1 will be an update
--> Package libfdisk.x86_64 0:2.30.2-2.amzn2.0.10 will be updated
--> Package libfdisk.x86_64 0:2.30.2-2.amzn2.0.11 will be an update
--> Package libmount.x86_64 0:2.30.2-2.amzn2.0.10 will be updated
--> Package libmount.x86_64 0:2.30.2-2.amzn2.0.11 will be an update
--> Package libsmartcols.x86_64 0:2.30.2-2.amzn2.0.10 will be updated
--> Package libsmartcols.x86_64 0:2.30.2-2.amzn2.0.11 will be an update
--> Package libuuid.x86_64 0:2.30.2-2.amzn2.0.10 will be updated
--> Package libuuid.x86_64 0:2.30.2-2.amzn2.0.11 will be an update

```

- c. `yum install -y httpd`

```
aws
Services
Search
[Alt+S]
N. Virginia
voclabs/user2409890=anirudha.22120150@viit.ac.in @ 5280-6614...

libfdisk.x86_64 0:2.30.2-2.amzn2.0.11 libmount.x86_64 0:2.30.2-2.amzn2.0.11 libsmarttools.x86_64 0:2.30.2-2.amzn2.0.11 libuuid.x86_64 0:2.30.2-2.amzn2.0.11
sysstat.x86_64 0:10.1.5-18.amzn2.0.2 util-linux.x86_64 0:2.30.2-2.amzn2.0.11 vim-common.x86_64 2:9.0.1160-1.amzn2.0.1 vim-data.noarch 2:9.0.1160-1.amzn2.0.1
vim-enhanced.x86_64 2:9.0.1160-1.amzn2.0.1 vim-filesystem.noarch 2:9.0.1160-1.amzn2.0.1 vim-minimal.x86_64 2:9.0.1160-1.amzn2.0.1

Complete!
[root@ip-172-31-60-186 ~]# yum install -y httpd
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
--> Package httpd.x86_64 0:2.4.54-1.amzn2 will be installed
--> Processing Dependency: httpd-tools = 2.4.54-1.amzn2 for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: httpd filesystem = 2.4.54-1.amzn2 for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: system-logos-httpd for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: mod_http2 for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: httpd filesystem for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: /etc/mime.types for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: libaprutil-1.so.0()(64bit) for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: libapr-1.so.0()(64bit) for package: httpd-2.4.54-1.amzn2.x86_64
--> Running transaction check
--> Package apr.x86_64 0:1.7.0-9.amzn2 will be installed
--> Package apr-util.x86_64 0:1.6.1-5.amzn2.0.2 will be installed
--> Processing Dependency: apr-util-bdb(x86-64) = 1.6.1-5.amzn2.0.2 for package: apr-util-1.6.1-5.amzn2.0.2.x86_64
--> Package generic-logos-httpd.noarch 0:18.0.0-4.amzn2 will be installed
--> Package httpd filesystem.noarch 0:2.4.54-1.amzn2 will be installed
--> Package httpd-tools.x86_64 0:2.4.54-1.amzn2 will be installed
--> Package mailcap.noarch 0:2.1.41-2.amzn2 will be installed
--> Package mod_http2.x86_64 0:1.15.19-1.amzn2.0.1 will be installed
--> Running transaction check
--> Package apr-util-bdb.x86_64 0:1.6.1-5.amzn2.0.2 will be installed
--> Finished Dependency Resolution

i-0731750f28c3564eb (aniserver)
PublicIPs: 18.235.255.199 PrivateIPs: 172.31.60.186
```

- d. `systemctl status httpd`
- e. `mkdir temp | cd temp`

The screenshot shows the AWS Management Console interface for an EC2 instance. The terminal window displays the following output:

```
Verifying : mod_http2-1.15.19-1.amzn2.0.1.x86_64 4/9
Verifying : httpd-2.4.54-1.amzn2.x86_64 5/9
Verifying : mailcap-2.1.41-2.amzn2.noarch 6/9
Verifying : generic-logos-httpd-18.0.0-4.amzn2.noarch 7/9
Verifying : httpd-filesystem-2.4.54-1.amzn2.noarch 8/9
Verifying : apr-1.7.0-9.amzn2.x86_64 9/9

Installed:
  httpd.x86_64 0:2.4.54-1.amzn2

Dependency Installed:
  apr.x86_64 0:1.7.0-9.amzn2          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 0:2.4.54-1.amzn2  httpd-tools.x86_64 0:2.4.54-1.amzn2  mailcap.noarch 0:2.1.41-2.amzn2  mod_http2.x86_64 0:1.15.19-1.amzn2.0.1

Complete!
[root@ip-172-31-60-186 ~]# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)
   Active: inactive (dead)
     Docs: man:httpd.service(8)
[root@ip-172-31-60-186 ~]# mkdir temp | cd temp
-bash: cd: temp: No such file or directory
[root@ip-172-31-60-186 ~]# mkdir temp | cd temp
mkdir: cannot create directory 'temp': File exists
[root@ip-172-31-60-186 ~]# rm temp
rm: cannot remove 'temp': Is a directory
[root@ip-172-31-60-186 ~]# rmdir temp
[root@ip-172-31-60-186 ~]# mkdir temp | cd temp
-bash: cd: temp: No such file or directory
[root@ip-172-31-60-186 ~]# cd temp
[root@ip-172-31-60-186 temp]#
```

Below the terminal window, the instance ID is shown as i-0731750f28c3564eb (aniserver) with public and private IP addresses.

- f. wget <https://www.free-css.com/assets/files/free-css-templates/download/page288/startup.zip>
1. As I'm no web developer so I took a template from <https://www.free-css.com/free-css-templates> remember to place the download link in front of wget

The screenshot shows the AWS Management Console interface for the same EC2 instance. The terminal window displays the following output:

```
[root@ip-172-31-60-186 temp]# wget https://www.free-css.com/assets/files/free-css-templates/download/page288/startup.zip
--2023-02-13 11:16:57-- https://www.free-css.com/assets/files/free-css-templates/download/page288/startup.zip
Resolving www.free-css.com (www.free-css.com)... 217.160.0.242, 2001:8d8:100f:f000::28f
Connecting to www.free-css.com (www.free-css.com)|217.160.0.242|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 684565 (669K) [application/zip]
Saving to: 'startup.zip'

0.00%[=====>] 684,565 1.05MB/s in 0.6s

2023-02-13 11:16:59 (1.05 MB/s) - 'startup.zip' saved [684565/684565]

[root@ip-172-31-60-186 temp]#
```

Below the terminal window, the instance ID is shown as i-0731750f28c3564eb (aniserver) with public and private IP addresses.

- g. unzip startup.zip



S3 Management Console x EC2 Instance Connect x +

https://us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0731750f28c3564eb 80%

Circuit design Copy of... Vishwakarma Institute... GATE CS Topic wise pr... EduPlusCampus VIIT Anirudha Jan 2023.pdf Syllabus-Structure-TY... National Digital Librar... GitHub Education Spotify

aws Services Search [Alt+S]

Connecting to www.free-css.com (www.free-css.com) | 217.160.0.242 | :443... connected.  
 HTTP request sent, awaiting response... 200 OK  
 Length: 684565 (669K) [application/zip]  
 Saving to: 'startup.zip'

100%[=====] 684,565 1.05MB/s in 0.6s

2023-02-13 11:16:59 (1.05 MB/s) - 'startup.zip' saved [684565/684565]

```
[root@ip-172-31-60-186 temp]# unzip startup.zip
Archive:  startup.zip
  inflating: startup-website-template/about.html
  inflating: startup-website-template/blog.html
  inflating: startup-website-template/contact.html
  creating: startup-website-template/css/
  inflating: startup-website-template/css/bootstrap.min.css
  inflating: startup-website-template/css/style.css
  inflating: startup-website-template/detail.html
  inflating: startup-website-template/feature.html
  creating: startup-website-template/img/
  inflating: startup-website-template/img/about.jpg
  inflating: startup-website-template/img/blog-1.jpg
  inflating: startup-website-template/img/blog-2.jpg
  inflating: startup-website-template/img/blog-3.jpg
  inflating: startup-website-template/img/carousel-1.jpg
  inflating: startup-website-template/img/carousel-2.jpg
  inflating: startup-website-template/img/feature.jpg
  inflating: startup-website-template/img/team-1.jpg
  inflating: startup-website-template/img/team-2.jpg
  inflating: startup-website-template/img/team-3.jpg
  inflating: startup-website-template/img/testimonial-1.jpg
```

i-0731750f28c3564eb (aniserver)  
 PublicIPs: 18.235.255.199 PrivateIPs: 172.31.60.186

Feedback Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

h. `cd startup-website-template/`

```
[root@ip-172-31-9-124 temp]# cd startup-website-template/
[root@ip-172-31-9-124 startup-website-template]# ls-lrt
-bash: ls-lrt: command not found
[root@ip-172-31-9-124 startup-website-template]# ls -lrt
total 400
-rw-r--r-- 1 root root 89601 Jul 27 2021 startup-website-template.jpg
-rw-r--r-- 1 root root 538 Aug 11 2021 READ-ME.txt
drwxr-xr-x 8 root root 99 Aug 11 2021 lib
drwxr-xr-x 2 root root 21 Aug 11 2021 js
drwxr-xr-x 2 root root 4096 Aug 11 2021 img
```

i. `mv * /var/www/html/`

```
[root@ip-172-31-9-124 startup-website-template]# mv * /var/www/html/
```

j. `cd /var/www/html/`

```
[root@ip-172-31-9-124 startup-website-template]# cd /var/www/html/
[root@ip-172-31-9-124 html]# ls -lrt
total 400
-rw-r--r-- 1 root root 89601 Jul 27 2021 startup-website-template.jpg
-rw-r--r-- 1 root root 538 Aug 11 2021 READ-ME.txt
drwxr-xr-x 8 root root 99 Aug 11 2021 lib
drwxr-xr-x 2 root root 21 Aug 11 2021 js
drwxr-xr-x 2 root root 4096 Aug 11 2021 img
drwxr-xr-x 2 root root 48 Aug 11 2021 css
-rw-r--r-- 1 root root 1456 Aug 16 2021 LICENSE.txt
-rw-r--r-- 1 root root 22799 Oct 19 2021 about.html
-rw-r--r-- 1 root root 38423 Oct 19 2021 blog.html
-rw-r--r-- 1 root root 20237 Oct 19 2021 contact.html
-rw-r--r-- 1 root root 30401 Oct 19 2021 detail.html
-rw-r--r-- 1 root root 19294 Oct 19 2021 testimonial.html
```

k. `ls -lrt`

```
[root@ip-172-31-9-124 html]# ls -lrt
total 400
-rw-r--r-- 1 root root 89601 Jul 27 2021 startup-website-template.jpg
-rw-r--r-- 1 root root 538 Aug 11 2021 READ-ME.txt
drwxr-xr-x 8 root root 99 Aug 11 2021 lib
drwxr-xr-x 2 root root 21 Aug 11 2021 js
drwxr-xr-x 2 root root 4096 Aug 11 2021 img
drwxr-xr-x 2 root root 48 Aug 11 2021 css
-rw-r--r-- 1 root root 1456 Aug 16 2021 LICENSE.txt
-rw-r--r-- 1 root root 22799 Oct 19 2021 about.html
-rw-r--r-- 1 root root 38423 Oct 19 2021 blog.html
-rw-r--r-- 1 root root 20237 Oct 19 2021 contact.html
-rw-r--r-- 1 root root 30401 Oct 19 2021 detail.html
-rw-r--r-- 1 root root 19294 Oct 19 2021 testimonial.html
-rw-r--r-- 1 root root 19029 Oct 19 2021 feature.html
-rw-r--r-- 1 root root 52988 Oct 19 2021 index.html
-rw-r--r-- 1 root root 24975 Oct 19 2021 price.html
-rw-r--r-- 1 root root 19602 Oct 19 2021 quote.html
-rw-r--r-- 1 root root 24329 Oct 19 2021 service.html
-rw-r--r-- 1 root root 20123 Oct 19 2021 team.html
```

#### l. systemctl enable httpd | systemctl start httpd

```
[root@ip-172-31-9-124 html]# systemctl enable httpd
Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to /usr/lib/systemd/system/httpd.service.
[root@ip-172-31-9-124 html]# system start httpd
-bash: system: command not found
[root@ip-172-31-9-124 html]# sysytemctl start httpd
-bash: sysytemctl: command not found
[root@ip-172-31-9-124 html]# sytemctl start httpd
-bash: sytemctl: command not found
[root@ip-172-31-9-124 html]# systemctl start httpd
```

#### m. systemctl status httpd

```
[root@ip-172-31-9-124 html]# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
   Active: active (running) since Sun 2023-02-05 12:37:24 UTC; 22s ago
     Docs: man:httpd.service(8)
   Main PID: 4034 (httpd)
   Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec: 0 B/sec"
   CGroup: /system.slice/httpd.service
           └─4034 /usr/sbin/httpd -DFOREGROUND
             └─4035 /usr/sbin/httpd -DFOREGROUND
               └─4036 /usr/sbin/httpd -DFOREGROUND
                 └─4037 /usr/sbin/httpd -DFOREGROUND
                   └─4038 /usr/sbin/httpd -DFOREGROUND
                     └─4039 /usr/sbin/httpd -DFOREGROUND

Feb 05 12:37:24 ip-172-31-9-124.ap-northeast-1.compute.internal systemd[1]: Starting The Apache HTTP Server...
Feb 05 12:37:24 ip-172-31-9-124.ap-northeast-1.compute.internal systemd[1]: Started The Apache HTTP Server.
```

## STEP 6

→ Change the inbound rules from security by simply clicking on launch wizard first

Instances | EC2 Management Console

EC2 Management Console

EC2 Instance Connect

https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Instances:

Find instance by attribute or tag (case-sensitive)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4
aniserver	i-0731750f28c3564eb	Running	t2.micro	2/2 checks passed	No alarms	us-east-1e	ec2-18-235-255-199.co...	18.235.255...

Instance: i-0731750f28c3564eb (aniserver)

Security groups

sg-05c8a5068018f506f (launch-wizard-1)

Inbound rules

Name	Security group rule ID	Port range	Protocol	Source	Security groups
-	sgr-020aefaad8f76734e	22	TCP	0.0.0.0/0	launch-wizard-1

Outbound rules

→ Click on edit inbound rules

Instances | EC2 Management Console

EC2 Management Console

EC2 Instance Connect

https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#SecurityGroups:group-name

Filter security groups

Security group name: launch-wizard-1

Name	Security group ID	Security group name	VPC ID	Description	Owner	Inbound rules count
-	sg-05c8a5068018f506f	launch-wizard-1	vpc-07475ccb0743a04f9	launch-wizard-1 create...	528066144619	1 Permission entry

You can now check network connectivity with Reachability Analyzer

Run Reachability Analyzer

Inbound rules (1/1)

Name	Security group rule...	IP version	Type	Protocol	Port range	Source
-	sgr-020aefaad8f76734e	IPv4	SSH	TCP	22	0.0.0.0/0

→ Add HTTP and HTTPS like the configuration given below

**Security Groups (1/2)**

Name	Security group ID	Security group name	VPC ID	Description	Owner	Inbound rules
launch-wizard-1	sg-05c8a5068018f506f	launch-wizard-1	vpc-07475ccb0743a04f9	launch-wizard-1 create...	528066144619	3 Permission ent...
default	sg-0e76ed5bdddd65ba93	default	vpc-07475ccb0743a04f9	default VPC security gr...	528066144619	1 Permission ent...

**Inbound rules (3)**

Name	Security group rule...	IP version	Type	Protocol	Port range	Source
-	sgr-020aefaad8f76734e	IPv4	SSH	TCP	22	0.0.0.0/0
-	sgr-07f7f3a6e88a4a82c	IPv4	HTTPS	TCP	443	0.0.0.0/0
-	sgr-09ac849f02d093d92	IPv4	HTTP	TCP	80	0.0.0.0/0

## STEP 7

→ Copy and paste the public IP to the browser to see the web template hosted



## STEP 8

→ Stop the instance & terminate it

---

## Conclusion :-

Thus, We Have Studied And Deployed Web application on AWS Cloud.