

1.

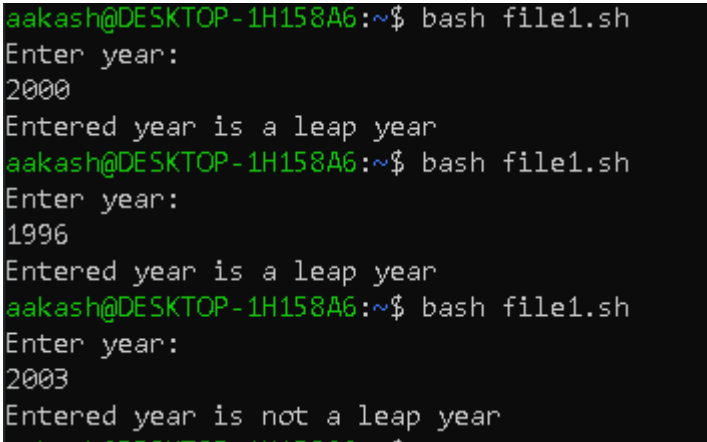
```
#!/bin/bash
```

```
echo "Enter year: "  
read y
```

```
if [[ $y%4 -eq 0 ]]  
then
```

```
    if [[ $y%100 -eq 0 ]]  
    then  
        if [[ $y%400 -eq 0 ]]  
        then  
            echo "Entered year is a leap year"  
        else  
            echo "Entered year is not a leap year"  
        fi  
    else  
        echo "Entered year is a leap year"  
    fi
```

```
else  
    echo "Entered year is not a leap year"  
fi
```



```
aakash@DESKTOP-1H158A6:~$ bash file1.sh  
Enter year:  
2000  
Entered year is a leap year  
aakash@DESKTOP-1H158A6:~$ bash file1.sh  
Enter year:  
1996  
Entered year is a leap year  
aakash@DESKTOP-1H158A6:~$ bash file1.sh  
Enter year:  
2003  
Entered year is not a leap year
```

## 2.

```
#!/bin/bash
echo "Shell program to find greatest of 3 number"

echo "Enter number 1 : "
read a

echo "Enter number 2: "
read b

echo "Enter number 3: "
read c

if [[ $a -gt $b ]]
then
if [[ $a -gt $c ]]
then
echo "Number 1 is greater than number 2 and 3"
else
echo "Number 3 is greater than number 1 and 2"
fi
else
if [[ $b -gt $c ]]
then
echo "Number 2 is greater than number 1 and 3"
else
echo "Number 3 is greater than number 1 and 2"
fi
fi
```

```
aakash@DESKTOP-1H158A6:~$ bash file2.sh
Shell program to find greatest of 3 number
Enter number 1 :
5
Enter number 2:
7
Enter number 3:
2
Number 2 is greater than number 1 and 3
```

3.

**#adduser nobita**

**#groupadd nobitag**

**#usermod -a -G nobitag nobita**

```
nobita@DESKTOP-1H158A6:~$ getent group
```

```
nobitag:x:1006:nobita
```

```
nobita@DESKTOP-1H158A6:~$ █
```

```
nobita@DESKTOP-1H158A6:~$ touch nobita_file.txt
```

```
nobita@DESKTOP-1H158A6:~$ ls -l
```

```
total 0
```

```
-rw-rw-r-- 1 nobita nobita 0 Sep 26 17:50 nobita_file.txt
```

```
nobita@DESKTOP-1H158A6:~$ chmod 400 nobita_file.txt
```

```
nobita@DESKTOP-1H158A6:~$ ls -l
```

```
total 0
```

```
-r----- 1 nobita nobita 0 Sep 26 17:50 nobita_file.txt
```

```
nobita@DESKTOP-1H158A6:~$
```

4.

```
aakash@DESKTOP-1H158A6:~$ mkdir dbda
aakash@DESKTOP-1H158A6:~$ cd dbda
aakash@DESKTOP-1H158A6:~/dbda$ touch Sep.txt
aakash@DESKTOP-1H158A6:~/dbda$ ls
Sep.txt
aakash@DESKTOP-1H158A6:~/dbda$ mv Sep.txt Batch2.txt
aakash@DESKTOP-1H158A6:~/dbda$ ls
Batch2.txt
aakash@DESKTOP-1H158A6:~/dbda$ cd ..
aakash@DESKTOP-1H158A6:~$ mkdir dbda_B2
aakash@DESKTOP-1H158A6:~$ tree
.
├── dbda
│   └── Batch2.txt
├── dbda_B2
├── file1.sh
└── file2.sh

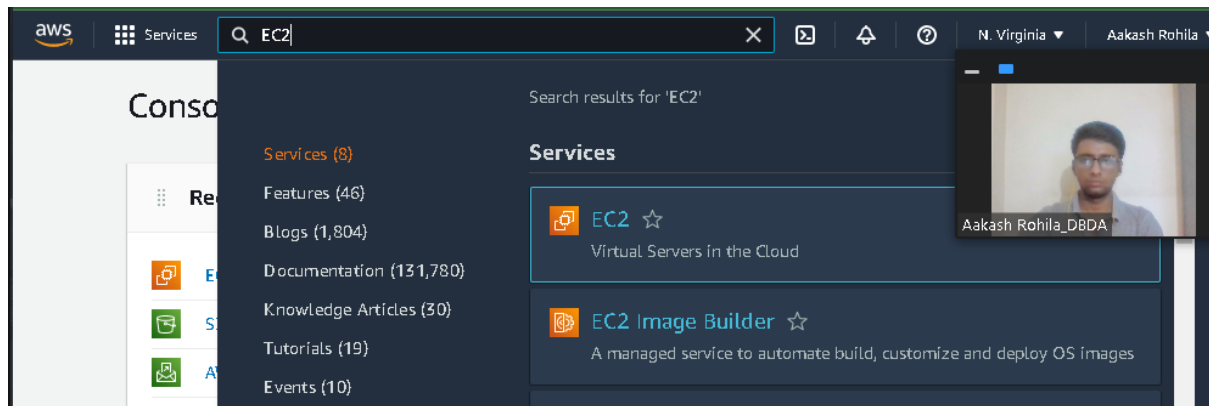
2 directories, 3 files
aakash@DESKTOP-1H158A6:~$ mv dbda/Batch2.txt dbda_B2
aakash@DESKTOP-1H158A6:~$ tree
.
├── dbda
├── dbda_B2
│   └── Batch2.txt
├── file1.sh
└── file2.sh

2 directories, 3 files
aakash@DESKTOP-1H158A6:~$
```

## 5.

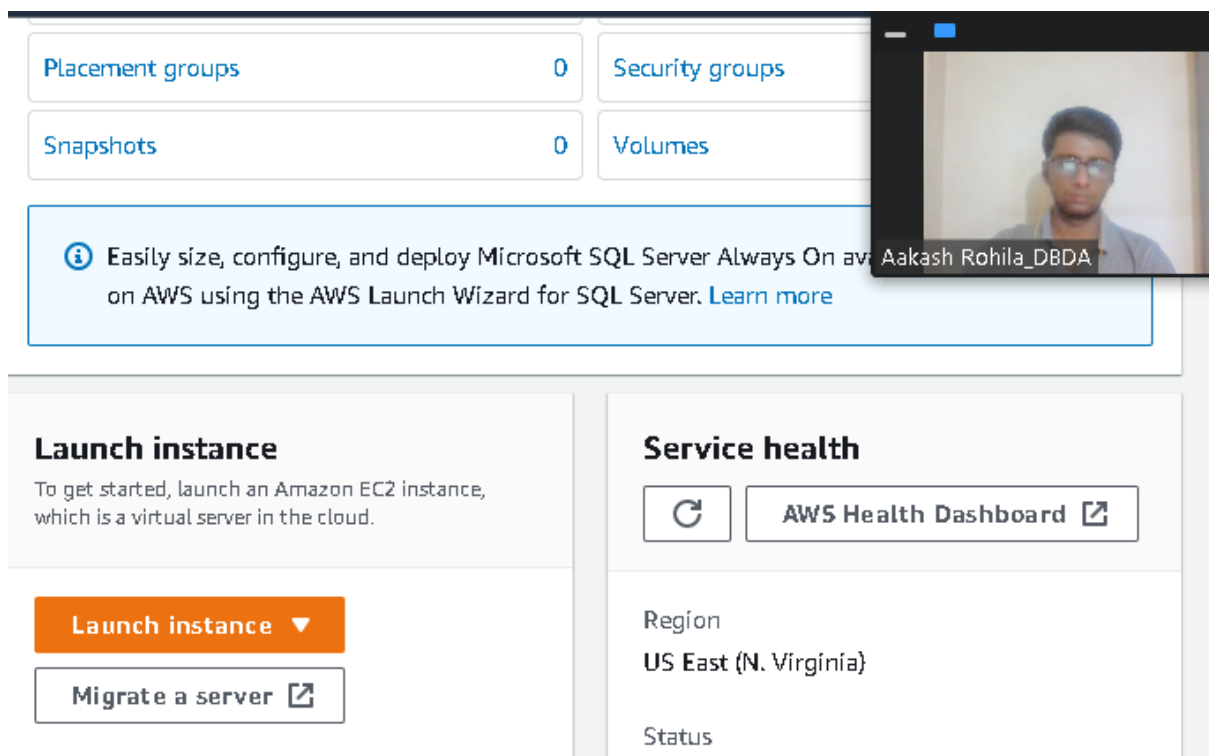
Step1>>

Go to EC2 service on Amazon AWS



Step2>>

Launch an E2 instance



### Step3>>Name your instance

## Launch an instance [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started with the simple steps below.

### Name and tags [Info](#)

Name

[Add additional tags](#)

### Step4>> Make sure you have selected the correct OS and version

Amazon Linux  
aws

macOS  
Mac

Ubuntu  
ubuntu

Windows  
Microsoft

Red Hat  
Red Hat

SUSE  
SUSE

Amazon Machine Image (AMI)

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type  
ami-026b57f3c383c2eec (64-bit (x86)) / ami-0636eac5d73e0e5d7 (64-bit (Arm))  
Virtualization: hvm   ENA enabled: true   Root device type: ebs

Free tier eligible ▼

Description

Amazon Linux 2 Kernel 5.10 AMI 2.0.20220912.1 x86\_64 HVM gp2

Architecture

AMI ID

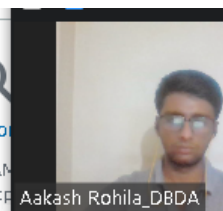
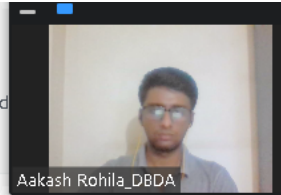
64-bit (x86) ▼

ami-026b57f3c383c2eec

Verified provider

Browse more

Including AMIs from AWS, Marketplace, and the Community



## Step 5>>

Create new keypair with .pem extension for linux machines

### Create key pair

Key pairs allow you to connect to your instance securely.

Enter the name of the key pair below. When prompted, store the private key in a secure and accessible location on your computer. **You will need it later to connect to your instance.** [Learn more](#)

Key pair name

demokey

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

☒ RSA

RSA encrypted private and public key pair

☐ ED25519

ED25519 encrypted private and public key pair (Not supported for Windows instances)

Private key file format

☒ .pem

For use with OpenSSH

☐ .ppk

For use with PuTTY



## Step6>>Select traffic from all source

### Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach you.

☒ Create security group

☐ Select existing security group

We'll create a new security group called 'launch-wizard-7' with the following rules:

☒ Allow SSH traffic from

Helps you connect to your instance

Anywhere

0.0.0.0/0

☒ Allow HTTPS traffic from the internet

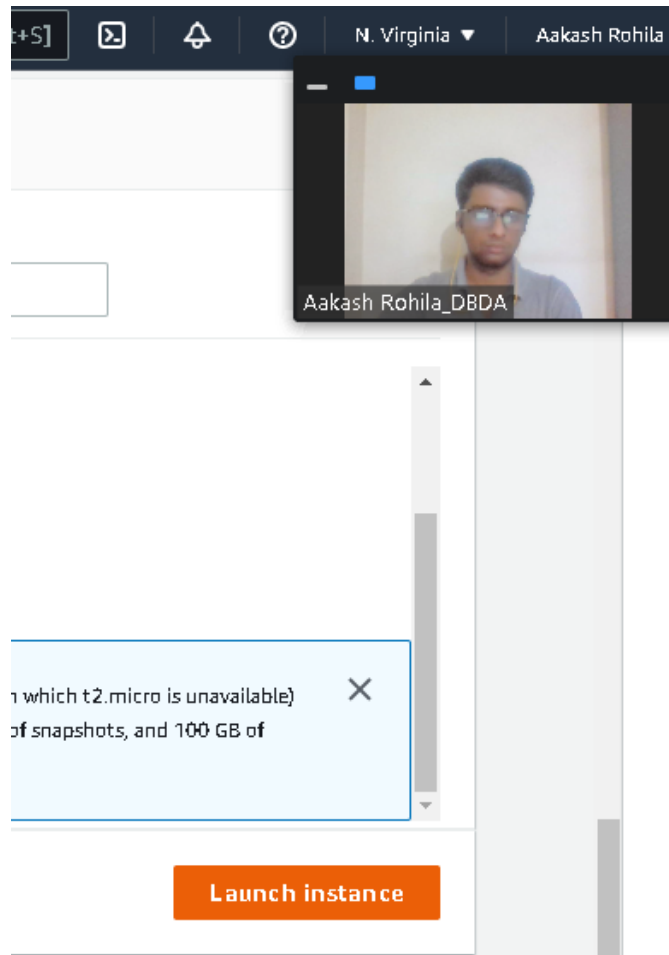
To set up an endpoint, for example when creating a web server

☒ Allow HTTP traffic from the internet

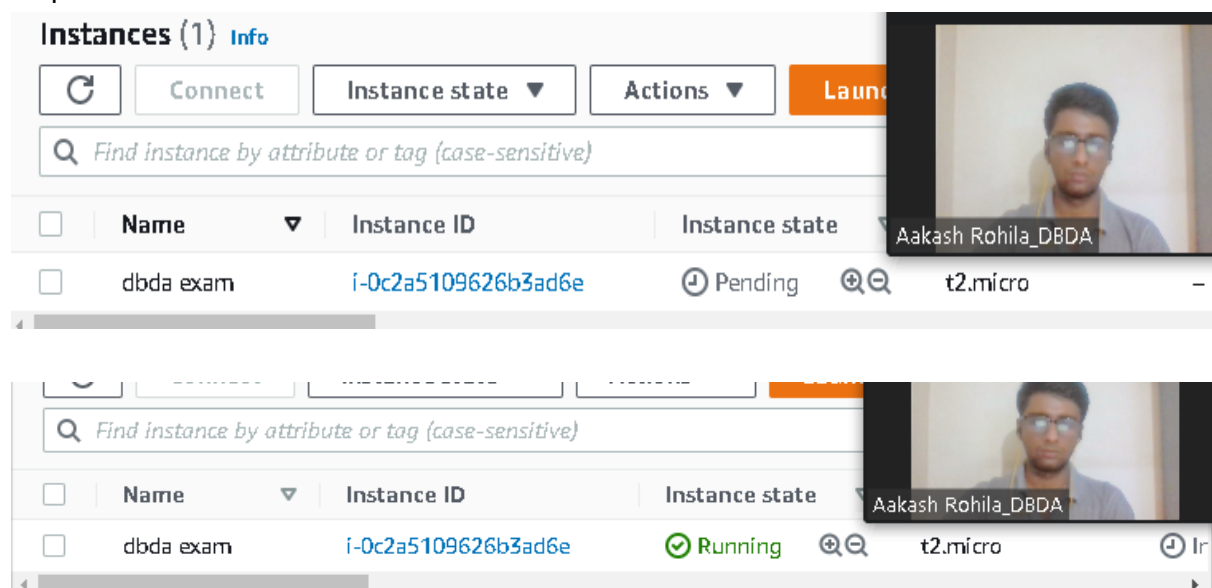
To set up an endpoint, for example when creating a web server



### Step7>>Launch Instance

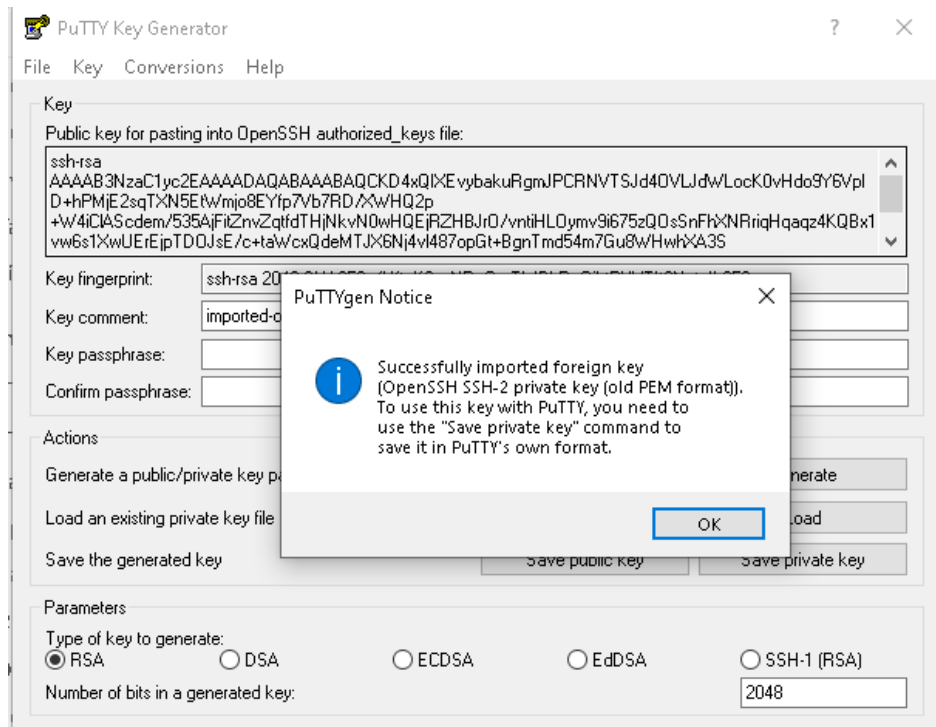


### Step8>>Instance Created

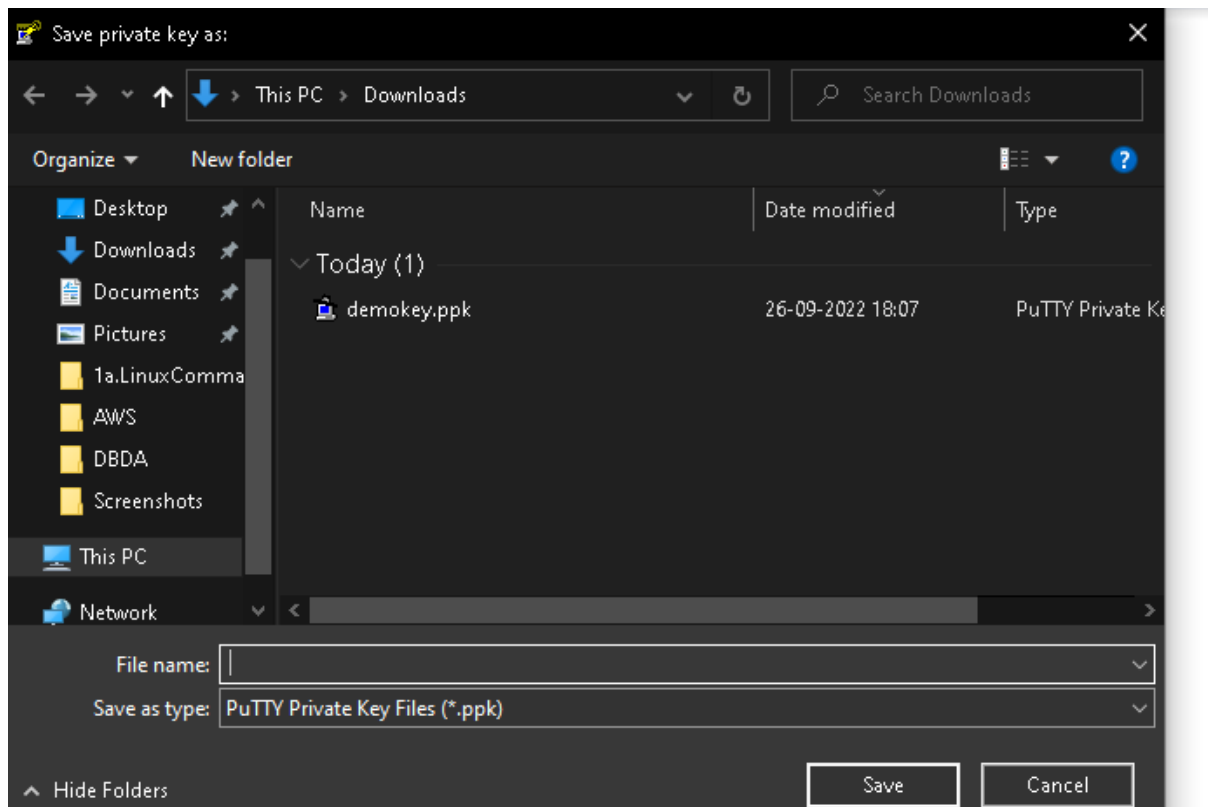




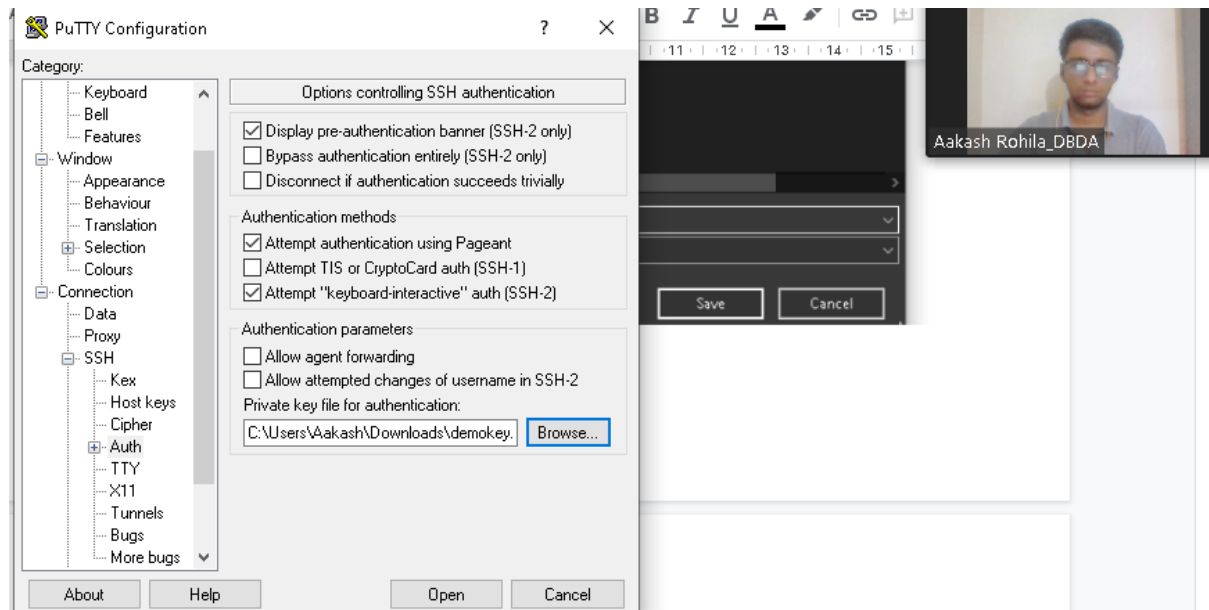
Step9>>Load .pem file in PuTTYGen to convert to .ppk format



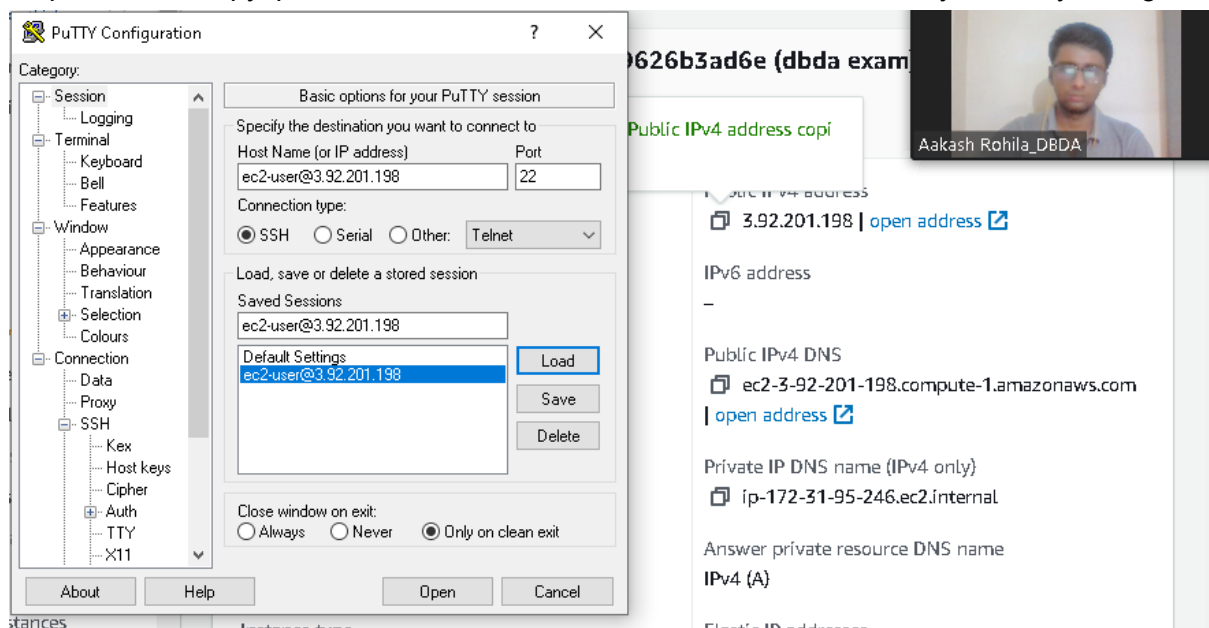
Step10>>Then Click on save private key and enter name of .ppk file and it should save your file in ..ppk format



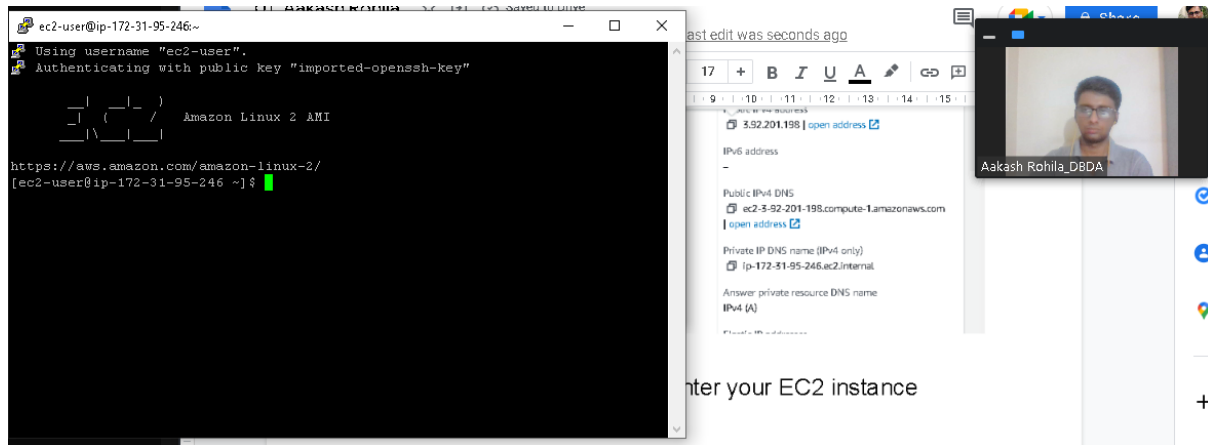
Step11>>Now start Putty on windows and load demokey.ppk file



Step12>>Now copy ip address from ec2 instance created and save it in your Putty config.



Step13>>Accept the next window and click enter your EC2 instance should start:



```
[root@ip-172-31-95-246 ec2-user]# yum update -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core                               | 3.7 kB      00:00
No packages marked for update
[root@ip-172-31-95-246 ec2-user]# yum install httpd -y
```

```
[root@ip-172-31-95-246 ec2-user]# systemctl start httpd
[root@ip-172-31-95-246 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-172-31-95-246 ec2-user]# 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 Mon 2022-09-26 12:47:45 UTC; 18s ago
     Docs: man:httpd.service(8)
  Main PID: 3468 (httpd)
    Status: "Total requests: 0; Idle/Busy workers 100/0; Requests/sec: 0; Bytes served/sec: 0 B/sec"
    CGroup: /system.slice/httpd.service
            └─3468 /usr/sbin/httpd -DFOREGROUND
              └─3469 /usr/sbin/httpd -DFOREGROUND
                └─3470 /usr/sbin/httpd -DFOREGROUND
                  └─3471 /usr/sbin/httpd -DFOREGROUND
                    └─3472 /usr/sbin/httpd -DFOREGROUND
                      └─3473 /usr/sbin/httpd -DFOREGROUND

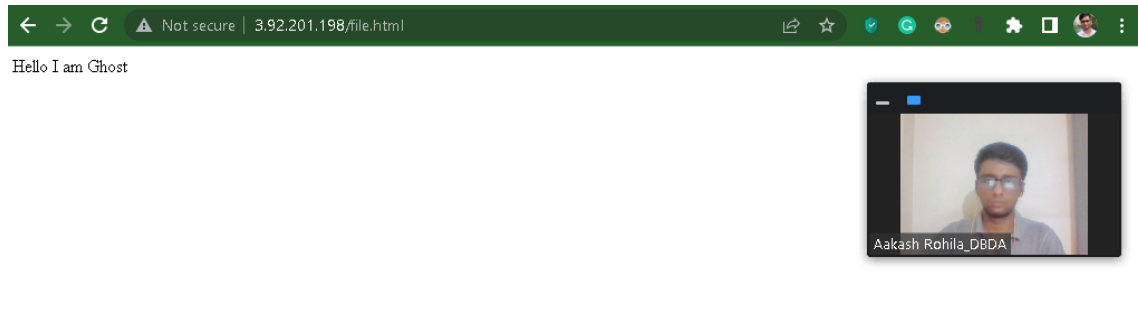
Sep 26 12:47:44 ip-172-31-95-246.ec2.internal systemd[1]: Starting The Apache.
Sep 26 12:47:45 ip-172-31-95-246.ec2.internal systemd[1]: Started The Apache.
Hint: Some lines were ellipsized, use -l to show in full.
```

```
[root@ip-172-31-95-246 ec2-user]# cd /var/www/html
[root@ip-172-31-95-246 html]# nano file.html
[root@ip-172-31-95-246 html]#
```

In the above command, we have started HTTP server and hosted our file.html context on it.

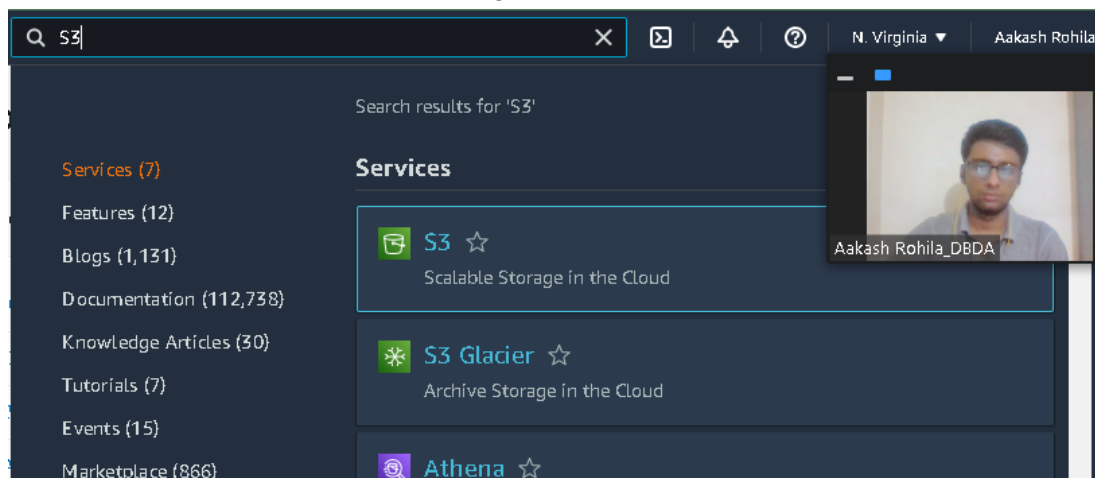
```
GNU nano 2.9.8 file.html
Hello I am Ghost
```

Step14>> Now enter IP address in the browser to check if our site is up and running

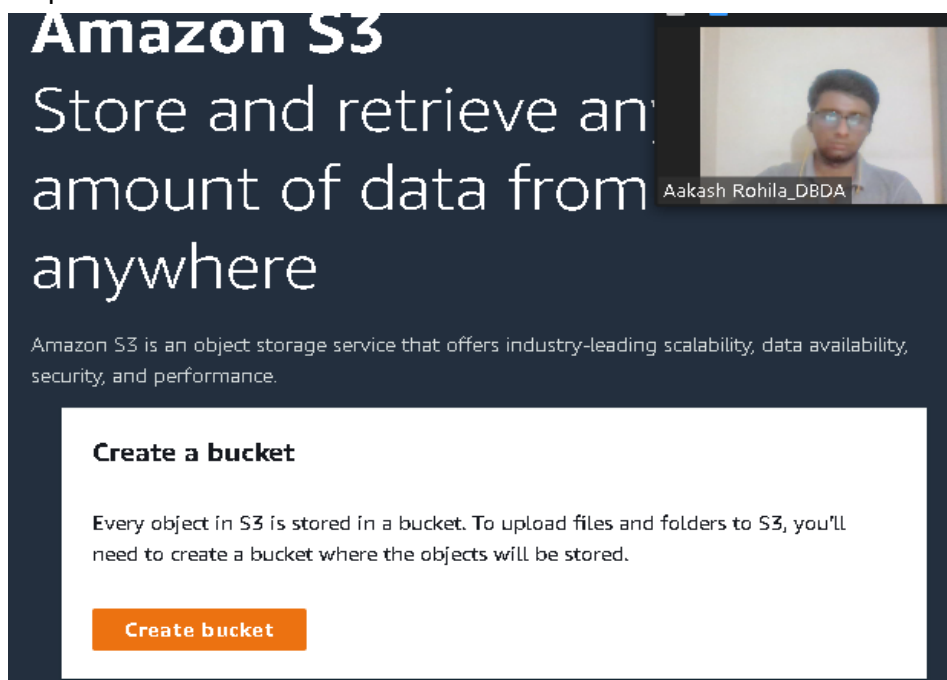


## 6.

Step1>>Now to create an S3 bucket, go to AWS website and search for S3 and click on it...




Step2>>Click on create bucket



Step3>>Enter your unique bucket name...

## Create bucket [Info](#)

Buckets are containers for data stored in S3. [Learn more](#)



### General configuration

Bucket name

Bucket name must be globally unique and must not contain spaces or uppercase letters. [See rules for bucket naming](#)

AWS Region

US East (N. Virginia) us-east-1

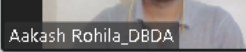
Copy settings from existing bucket - *optional*  
Only the bucket settings in the following configuration are copied.

Choose bucket

Step4>>Make ACL Enabled and make public traffic accessible...

## Object Ownership [Info](#)

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.



☐ ACLs disabled (recommended)  
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

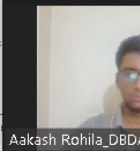
☒ ACLs enabled  
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.


### Object Ownership

☒ Bucket owner preferred  
If new objects written to this bucket specify the bucket-owner-full-control canned ACL, they are

☐ Block all public access  
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of each other.

- ☐ Block public access to buckets and objects granted through *new* access control lists (ACLs)  
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to buckets and objects using ACLs.
- ☐ Block public access to buckets and objects granted through *any* access control lists (ACLs)  
S3 will ignore all ACLs that grant public access to buckets and objects.
- ☐ Block public access to buckets and objects granted through *new* public bucket or access point policies  
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- ☐ Block public and cross-account access to buckets and objects through *any* public bucket or access point policies  
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

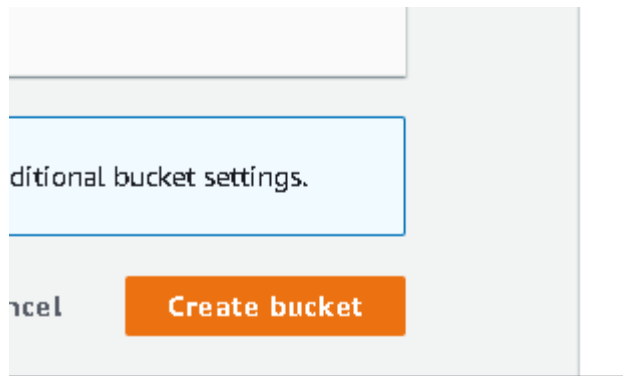


**Turning off block all public access might result in this bucket and the objects within becoming public**

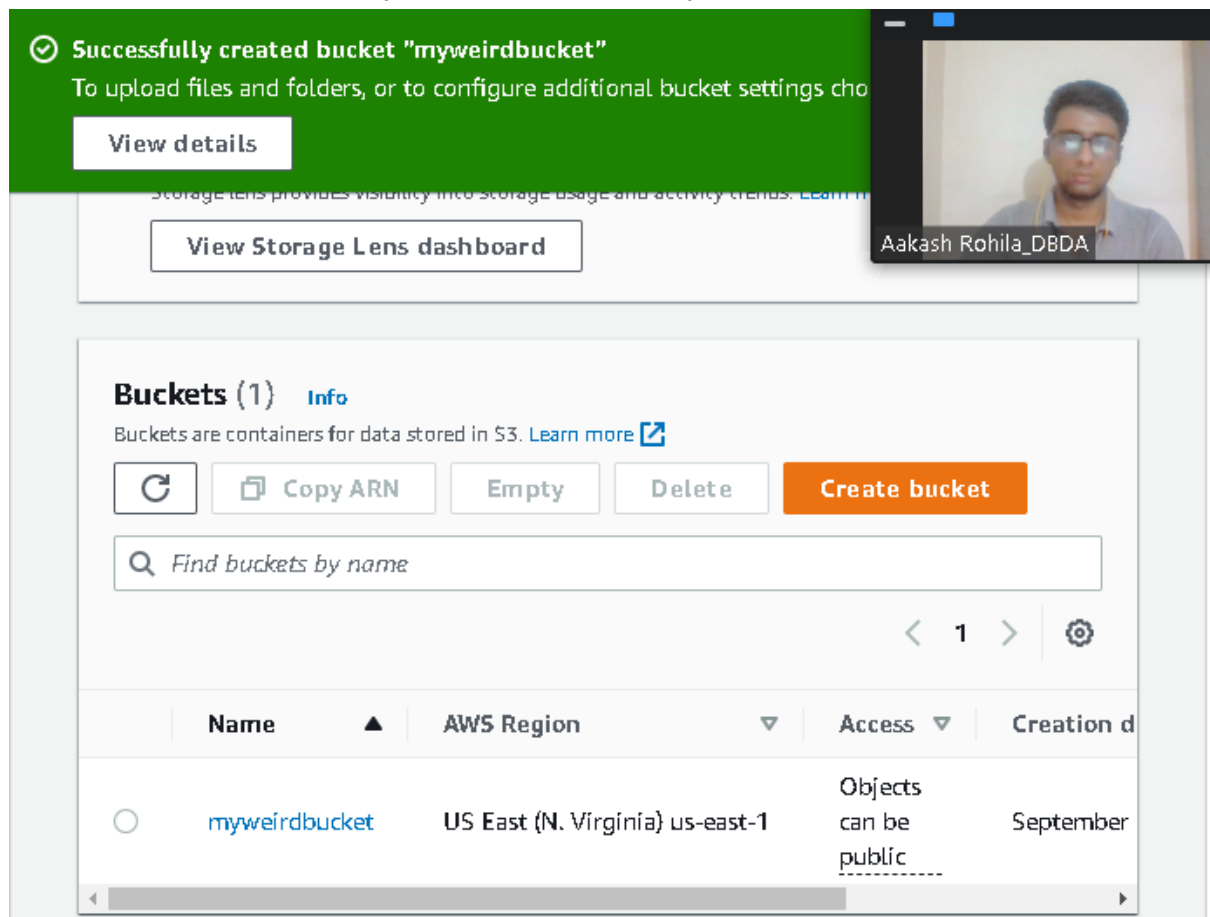
AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.

☒ I acknowledge that the current settings might result in this bucket and the objects within becoming public.

Step5>>Click on create bucket



Step6>>And it should create your bucket successfully



Step7>>To upload a file click on your bucket name link and click on upload file.

## Upload [Info](#)

Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the [AWS CLI](#), [AWS S3 CLI](#), or [S3 REST API](#). [Learn more](#)

Drag and drop files and folders you want to upload here, or choose **Add files**, or **Add folders**.

Files and folders (1 Total, 656.0 KB)  
All files and folders in this table will be uploaded.

Remove

Add files

Add folder

< 1 >

<input type="checkbox"/>	Name	Folder	Type	Size
<input type="checkbox"/>	peakpx.jpg	-	image/jpeg	656.0 KB

Step8>>It will show upload successfully

Upload succeeded  
View details below.

## Upload: status

The information below will no longer be available after you navigate away from this page.

### Summary

Destination <a href="#">s3://myweirdbucket</a>	Succeeded ✔ 1 file, 656.0 KB (100.00%)	Failed ⌛ 0 files, 0 B (0%)
---	---	-------------------------------

Step9>>Now in action centre make your file accessible i.e. Make public using ACL.

Edit storage class

Edit server-side encryption

Edit metadata

Edit tags

Make public using ACL

Copy URL

Download

Open

Delete

Actions

Create folder

Upload

< 1 >

<input checked="" type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input checked="" type="checkbox"/>	peakpx.jpg	jpg	September 26, 2022, 18:27:51 (UTC+05:30)	656.0 KB	Standard



Step10>>Through the object link now we can access the file:

**Object overview**

Owner	aakash.rohila.dbda.kh
AWS Region	US East (N. Virginia) us-east-1
Last modified	September 26, 2022, 18:27:51 (UTC+05:30)
Size	656.0 KB
Type	jpg
S3 URI	s3://myweirdbucket/peakpx.jpg
Amazon Resource Name (ARN)	arn:aws:s3:::myweirdbucket/peakpx.jpg
Entity tag (Etag)	1c163c7ff759bdeaa319416e791ca9fc
Object URL	<a href="https://myweirdbucket.s3.amazonaws.com/peakpx.jpg">https://myweirdbucket.s3.amazonaws.com/peakpx.jpg</a>

Step11>>Now this image is accessible through public ip...

