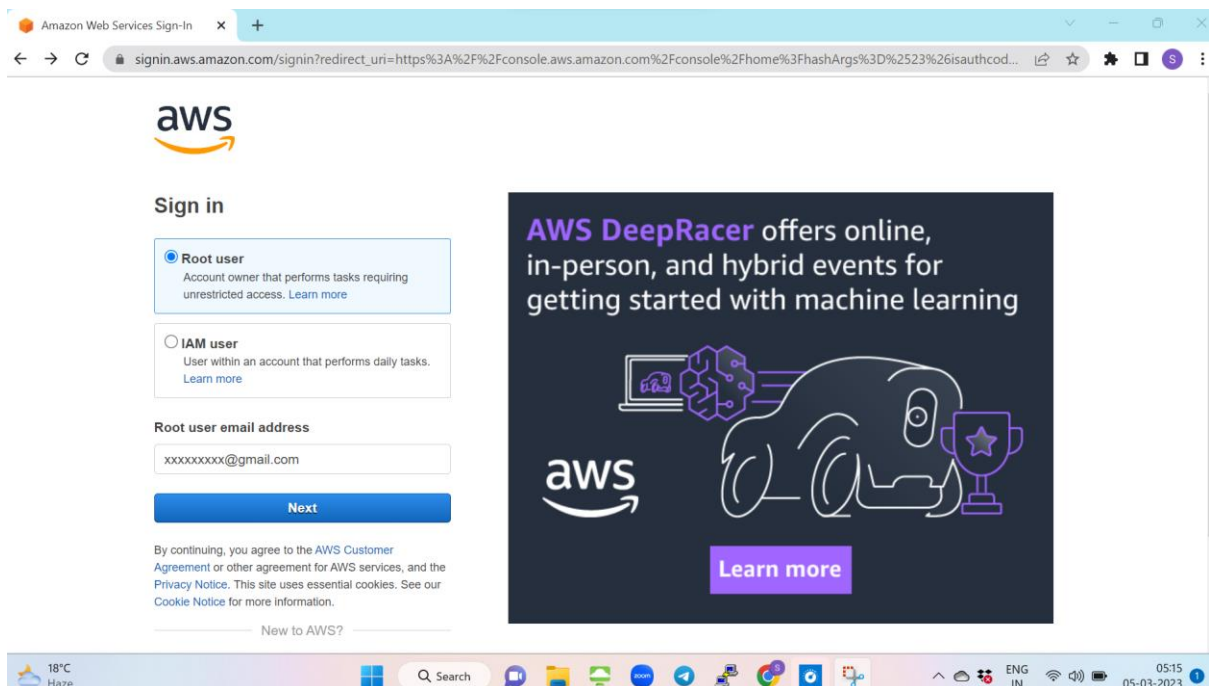


END TO END DEPLOYMENT OF ONE TIER WEB APPLICATION USING AWS LINUX INSTANCE

AWS EC2 Linux Instance: Step by step procedure to create Linux instance in AWS EC2 and deployment of one tier static

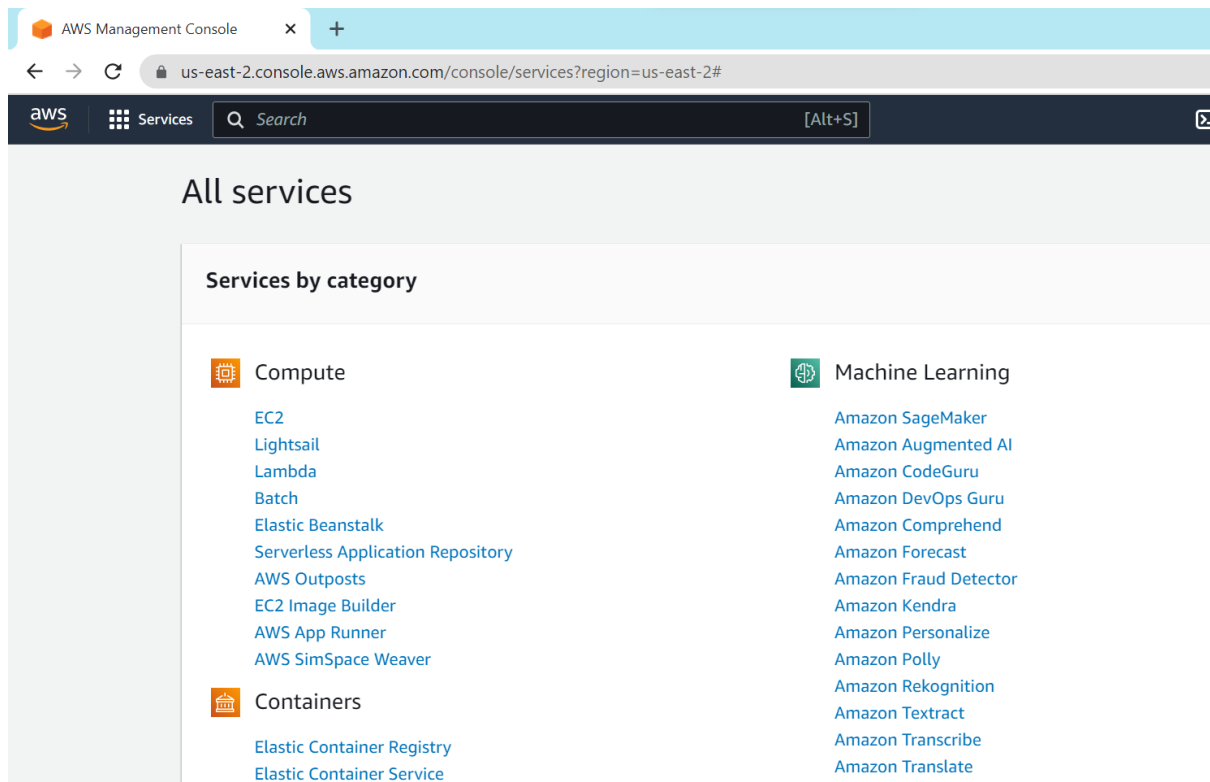
Web application in it.

Step1: Create one free tier aws account and login as root user.



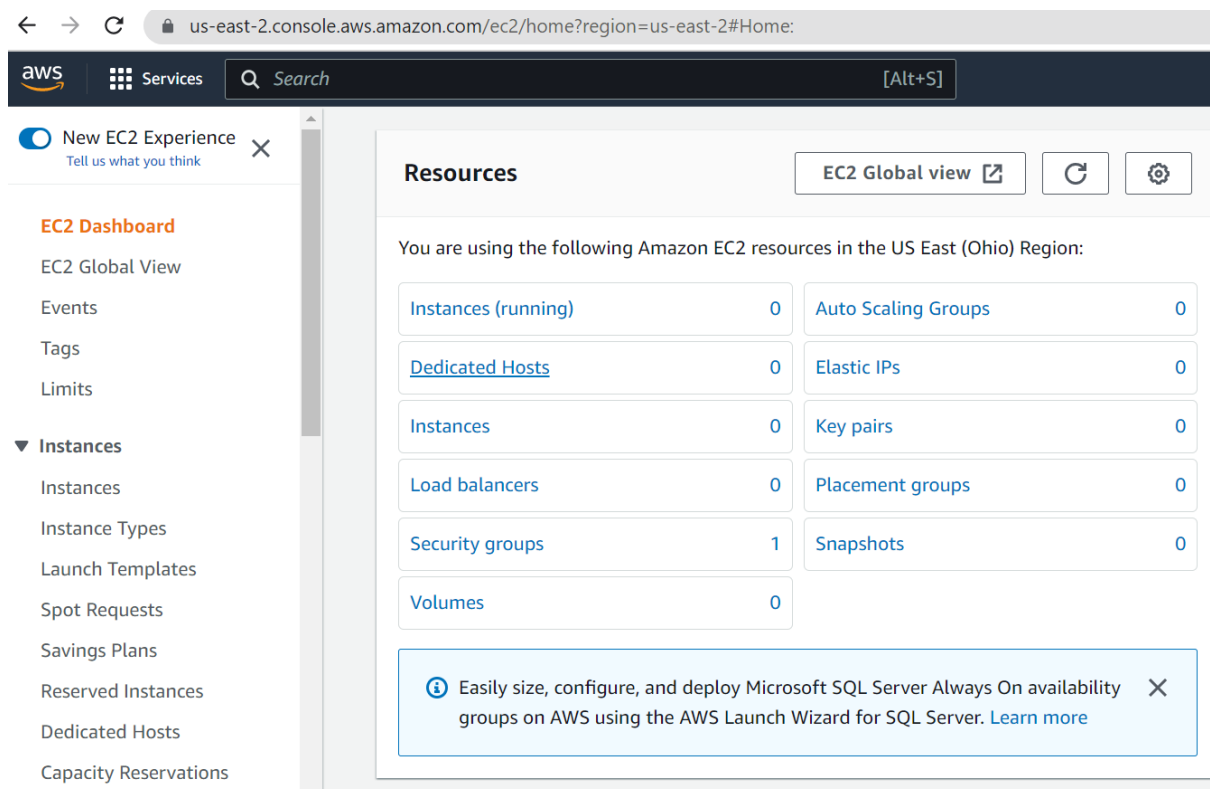
Step 2: go to AWS all services and search for compute and select EC2, which elastic compute cloud.

END TO END DEPLOYMENT OF ONE TIER WEB APPLICATION USING AWS LINUX INSTANCE



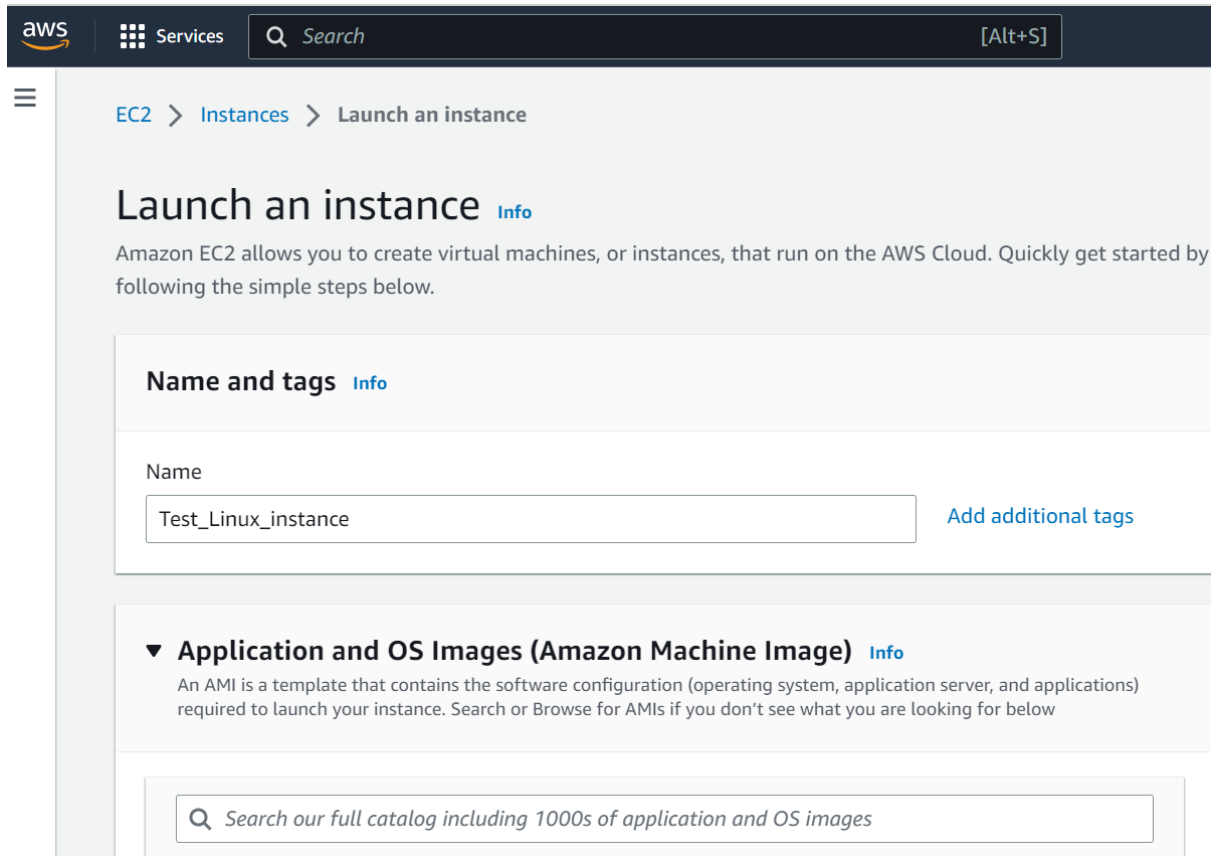
Step 3:

Go to instances.



END TO END DEPLOYMENT OF ONE TIER WEB APPLICATION USING AWS LINUX INSTANCE

Step 4: Give name to your Instance in Name and tags.

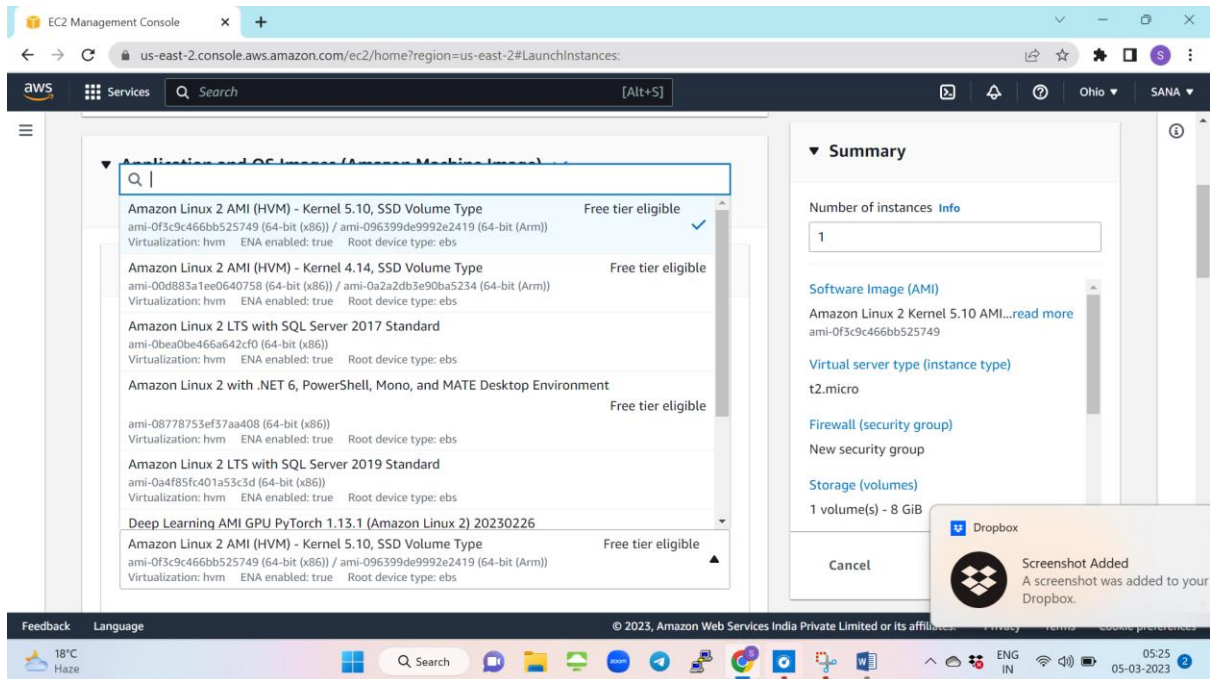


The screenshot displays the AWS Management Console interface for launching an EC2 instance. The breadcrumb navigation shows 'EC2 > Instances > Launch an instance'. The main heading is 'Launch an instance' with an 'Info' link. Below the heading, a descriptive paragraph states: 'Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.'

The 'Name and tags' section is expanded, showing a 'Name' label and a text input field containing 'Test_Linux_instance'. To the right of the input field is a link labeled 'Add additional tags'. Below this section, the 'Application and OS Images (Amazon Machine Image)' section is also expanded, showing a search bar with the placeholder text 'Search our full catalog including 1000s of application and OS images'.

Step 5: Choose free tier AMI, i.e one need to choose the OS for their server. (linux/Windows/any other linux flavours) .I have choosed Amazon linux.

END TO END DEPLOYMENT OF ONE TIER WEB APPLICATION USING AWS LINUX INSTANCE



Step 6: Give Keypair name , download PPK putty private key file for your linux instance.

Key pair name

Linux_key_05MAR

The name can include upto 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

Cancel

Create key pair

Step 7: For linux instance, in network settings,

END TO END DEPLOYMENT OF ONE TIER WEB APPLICATION USING AWS LINUX INSTANCE

A protocol is a set of standard rules which must be followed by two systems (client&server) in order to make communication possible,default protocol for linux is ssh. i.e: SSH, also known as Secure Socket Shell, is a network protocol that provides administrators with a secure way to access a remote computer.

In this document, we are going to launch simple webserver over EC2,for that Http protocol is required.(**HTTP** is a protocol for fetching resources such as HTML documents.)

▼ Network settings [Info](#)

Edit

Network [Info](#)

vpc-00a6af4f188fc928d

Subnet [Info](#)

No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)

Enable

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 your instance.

☒ Create security group

☐ Select existing security group

We'll create a new security group called '**launch-wizard-1**' 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

Step 8: give default storage provided by AWS for free tier.preview the summary and launch as shown below.

END TO END DEPLOYMENT OF ONE TIER WEB APPLICATION USING AWS LINUX INSTANCE

▼ Configure storage [Info](#)

[Advanced](#)

1x GiB ▼ Root volume (Not encrypted)

[i](#) Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage [X](#)

Add new volume

0 x File systems

[Edit](#)

▼ Summary

Number of instances [Info](#)

Amazon Linux 2 Kernel 5.10 AMI...[read more](#)
ami-0f3c9c466bb525749

[Virtual server type \(instance type\)](#)

t2.micro

[Firewall \(security group\)](#)

New security group

[Storage \(volumes\)](#)

1 volume(s) - 8 GiB

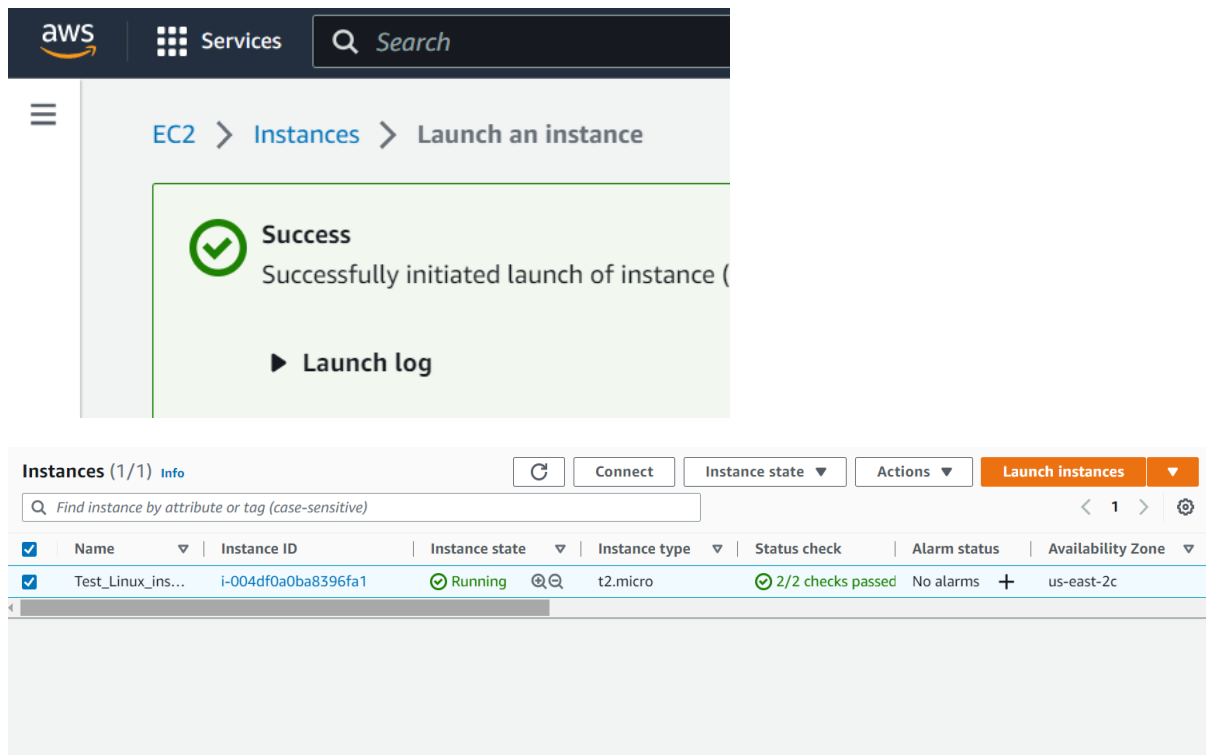
[i](#) **Free tier:** In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is [X](#)

Cancel

Launch instance

Step 9: Instance successfully launched in your required region. Make sure you are searching in the region where you created your actual instance.

END TO END DEPLOYMENT OF ONE TIER WEB APPLICATION USING AWS LINUX INSTANCE



Step 10: Ensure 2/2 checks for your EC2 instance is passed.

When an EC2 instance is launched, AWS checks whether that ec2 is able to connect to the network or not. It does so in two steps.

Step 10a: System check, this checks for proper hardware configuration of the instance at host level. It should be noted that you can not check these. These are aws managed. The best you can do is just stop and start the Instance in case system checks fail. It simply deallocates the previous VM and a new VM on different hardware is provisioned. If you want to retain public IP and other related stuff, you can configure Auto Recovery of instance from cloudwatch console.

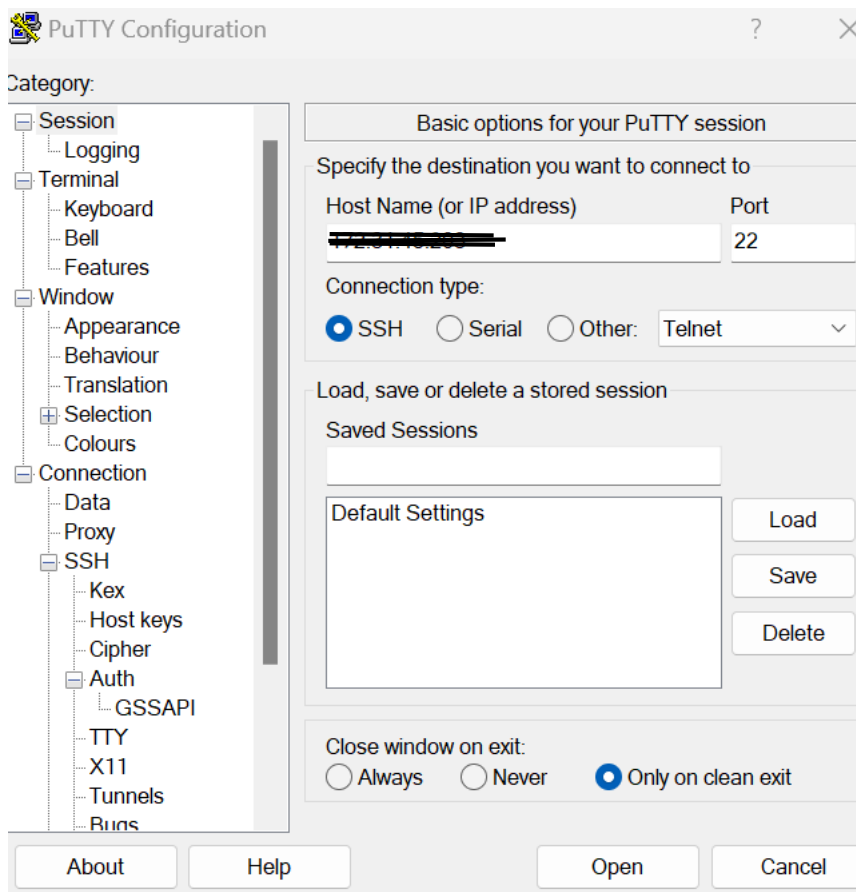
Step 10b : Instance check, this check is done to check software(os) level configuration of the instance. This can be looked into by you. If this check fails then you will need to check either user data/ launch configuration or AMI/OS configuration.

Step 11:Copy Public Ip under networking. Paste it in Putty as shown below.

END TO END DEPLOYMENT OF ONE TIER WEB APPLICATION USING AWS LINUX INSTANCE

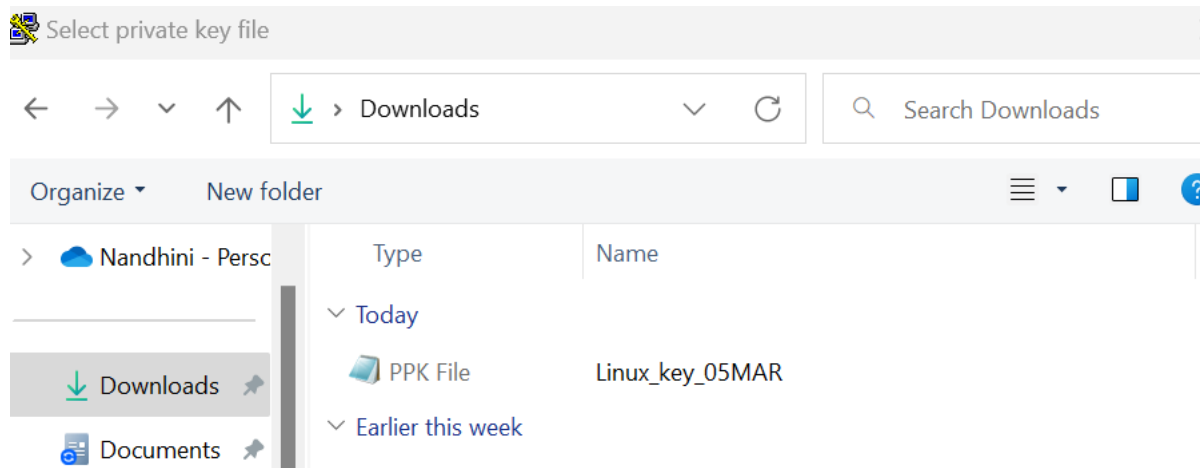
Instance: i-004df0a0ba8396fa1 (Test_Linux_instance)

Details	Security	Networking	Storage	Status checks	Monitoring	Tags
You can now check network connectivity with Reachability Analyzer.						
▼ Networking details Info						
Public IPv4 address 3.19.29.39 open address			Private IPv4 addresses 172.31.45.203			V I
Public IPv4 DNS ec2-3-19-29-39.us-east-			Private IP DNS name (IPv4 only) ip-172-31-45-203.us-east-2.compute.internal			

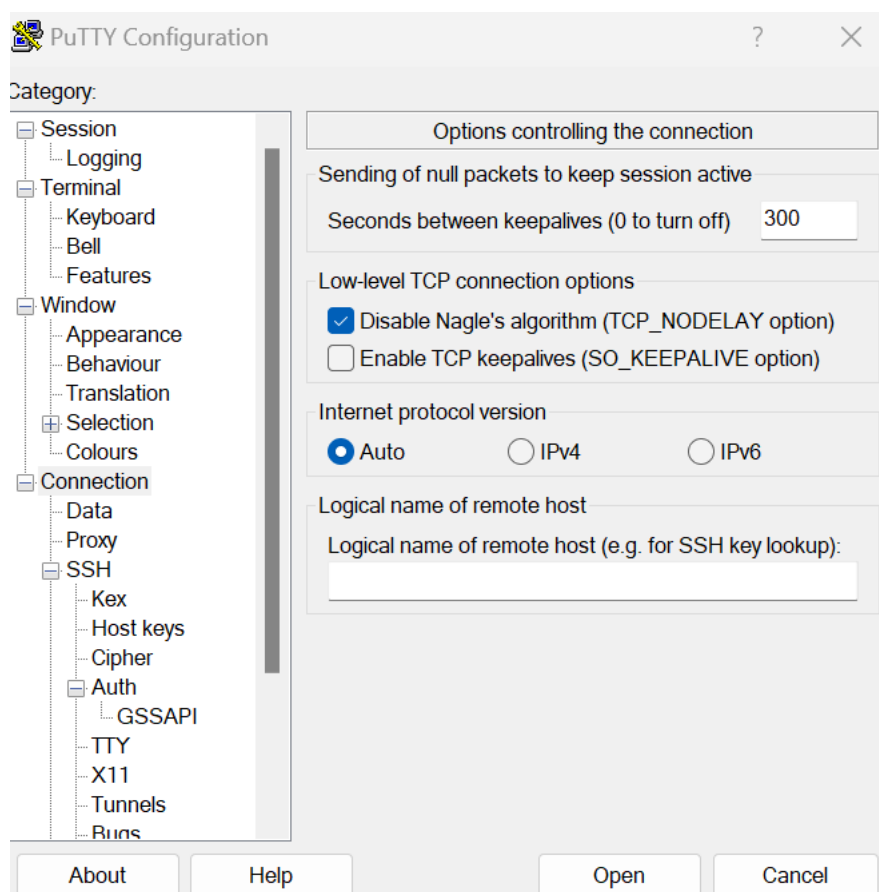


Go to SSH then choose Auth in that and browse and give your PPK file.

END TO END DEPLOYMENT OF ONE TIER WEB APPLICATION USING AWS LINUX INSTANCE

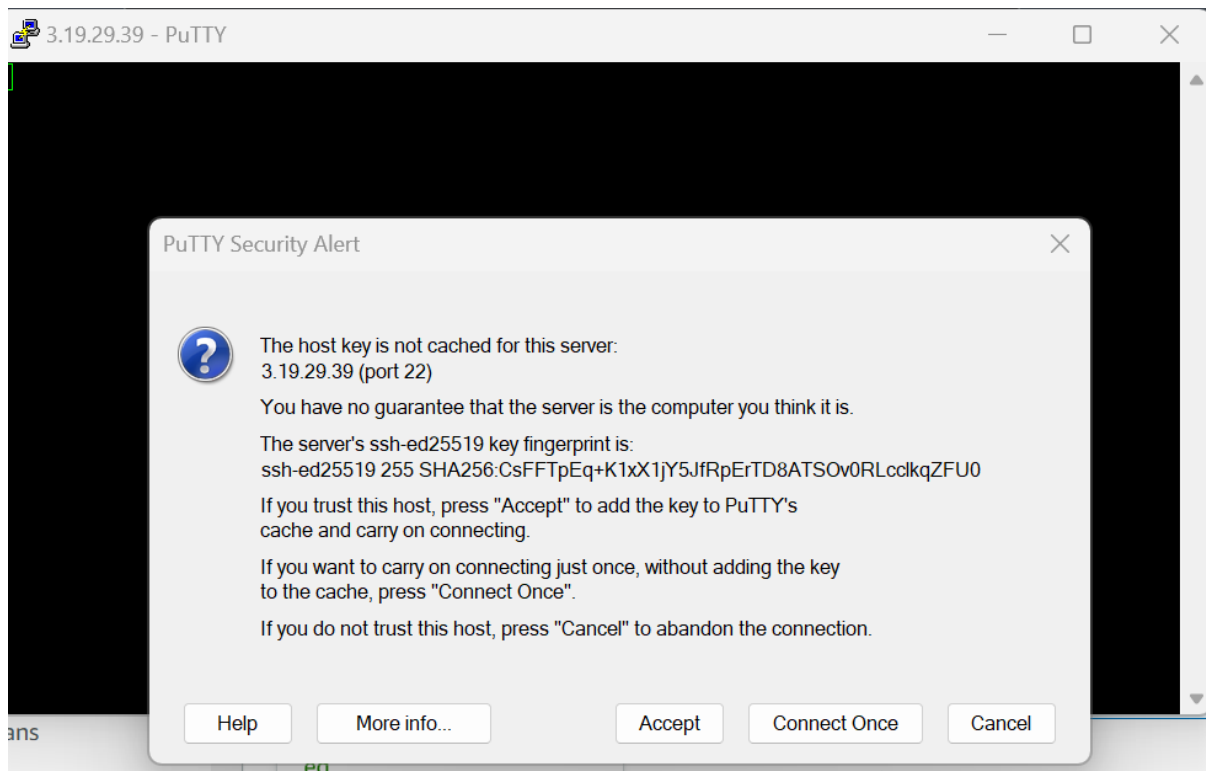


Under connection ,give the timings ,how long one need to keep your session alive. One can customise the screen settings in putty.



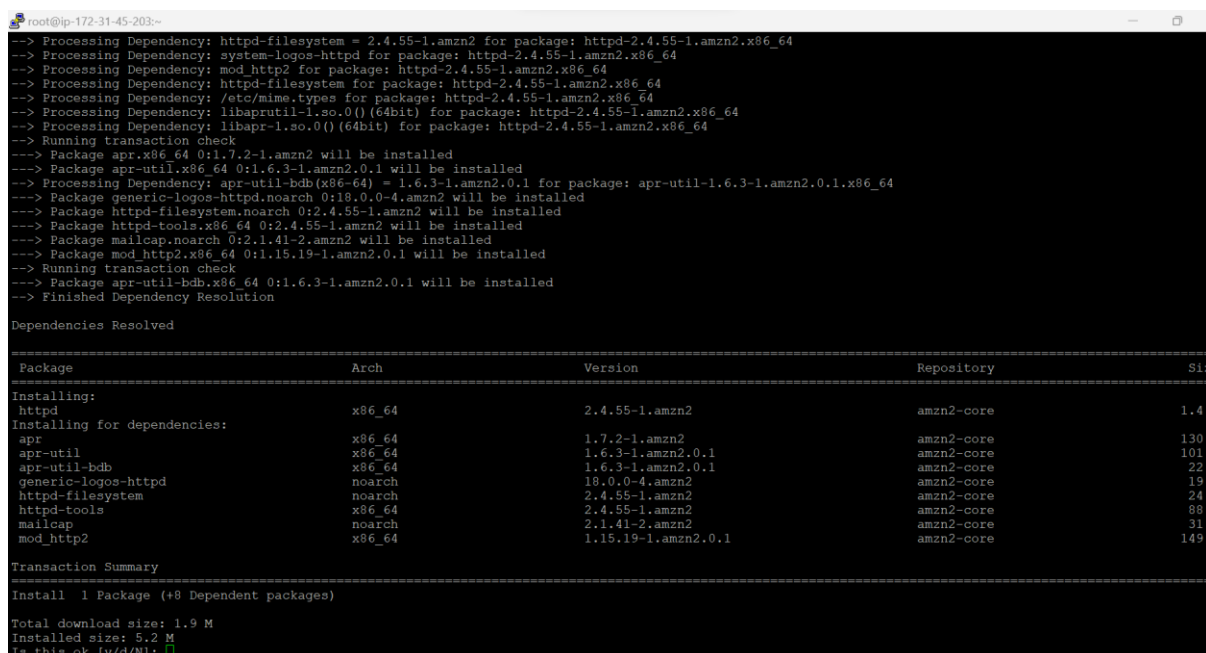
Give open and you will get the below window.accept and open it.

END TO END DEPLOYMENT OF ONE TIER WEB APPLICATION USING AWS LINUX INSTANCE



Now Linux machine Virtualisation has been done with the help of Putty ssh client. In linux machine one need to be root user to install any thing.

Give sudo -i and switch to root user,give yum install httpd -y to install apache web server,in the below Screenshot,I have provide without -y for your reference.



END TO END DEPLOYMENT OF ONE TIER WEB APPLICATION USING AWS LINUX INSTANCE

Give y to install further. Once done you will get the below page.

```
(1/9): apr-1.7.2-1.amzn2.x86_64.rpm | 130 kB 00:00:00
(2/9): apr-util-1.6.3-1.amzn2.0.1.x86_64.rpm | 101 kB 00:00:00
(3/9): apr-util-bdb-1.6.3-1.amzn2.0.1.x86_64.rpm | 22 kB 00:00:00
(4/9): generic-logos-httpd-18.0.0-4.amzn2.noarch.rpm | 19 kB 00:00:00
(5/9): httpd-filesystem-2.4.55-1.amzn2.noarch.rpm | 24 kB 00:00:00
(6/9): httpd-2.4.55-1.amzn2.x86_64.rpm | 1.4 MB 00:00:00
(7/9): httpd-tools-2.4.55-1.amzn2.x86_64.rpm | 88 kB 00:00:00
(8/9): mailcap-2.1.41-2.amzn2.noarch.rpm | 31 kB 00:00:00
(9/9): mod_http2-1.15.19-1.amzn2.0.1.x86_64.rpm | 149 kB 00:00:00
-----
Total | 10 MB/s | 1.9 MB 00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Installing : apr-1.7.2-1.amzn2.x86_64 1/9
Installing : apr-util-1.6.3-1.amzn2.0.1.x86_64 2/9
Installing : apr-util-bdb-1.6.3-1.amzn2.0.1.x86_64 3/9
Installing : httpd-tools-2.4.55-1.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-2.4.55-1.amzn2.noarch 7/9
Installing : mod_http2-1.15.19-1.amzn2.0.1.x86_64 8/9
Installing : httpd-2.4.55-1.amzn2.x86_64 9/9
Verifying : apr-util-bdb-1.6.3-1.amzn2.0.1.x86_64 1/9
Verifying : httpd-2.4.55-1.amzn2.x86_64 2/9
Verifying : apr-1.7.2-1.amzn2.x86_64 3/9
Verifying : httpd-filesystem-2.4.55-1.amzn2.noarch 4/9
Verifying : mailcap-2.1.41-2.amzn2.noarch 5/9
Verifying : generic-logos-httpd-18.0.0-4.amzn2.noarch 6/9
Verifying : mod_http2-1.15.19-1.amzn2.0.1.x86_64 7/9
Verifying : httpd-tools-2.4.55-1.amzn2.x86_64 8/9
Verifying : apr-util-1.6.3-1.amzn2.0.1.x86_64 9/9

Installed:
httpd.x86_64 0:2.4.55-1.amzn2

Dependency Installed:
apr.x86_64 0:1.7.2-1.amzn2          apr-util.x86_64 0:1.6.3-1.amzn2.0.1          apr-util-bdb.x86_64 0:1.6.3-1.amzn2.0.1
generic-logos-httpd.noarch 0:18.0.0-4.amzn2      httpd-filesystem.noarch 0:2.4.55-1.amzn2          httpd-tools.x86_64 0:2.4.55-1.amzn2
mailcap.noarch 0:2.1.41-2.amzn2      mod_http2.x86_64 0:1.15.19-1.amzn2.0.1

Complete!
[root@ip-172-31-45-203 ~]#
```

One can check whether httpd is installed or not using below commands.httpd -V or yum info httpd

```
root@ip-172-31-45-203 ~]# httpd -V
server version: Apache/2.4.55 ()
server built: Feb 9 2023 18:42:11
server's Module Magic Number: 20120211:126
server loaded: APR 1.7.2, APR-UTIL 1.6.3, PCRE 8.32 2012-11-30
compiled using: APR 1.7.0, APR-UTIL 1.6.3, PCRE 8.32 2012-11-30
architecture: 64-bit
server MPM: prefork
  threaded: no
    forked: yes (variable process count)
server compiled with....
-D APR_HAS_SENDFILE
-D APR_HAS_MMAP
-D APR_HAVE_IPV6 (IPv4-mapped addresses enabled)
-D APR_USE_PROC_PTHREAD_SERIALIZE
-D APR_USE_PTHREAD_SERIALIZE
-D SINGLE_LISTEN_UNSERIALIZED_ACCEPT
-D APR_HAS_OTHER_CHILD
-D AP_HAVE_RELIABLE_PIPED_LOGS
-D DYNAMIC_MODULE_LIMIT=256
-D HTTPD_ROOT="/etc/httpd"
-D SUEXEC_BIN="/usr/sbin/suexec"
-D DEFAULT_PIDLOG="/run/httpd/httpd.pid"
-D DEFAULT_SCOREBOARD="logs/apache_runtime_status"
-D DEFAULT_ERRORLOG="logs/error_log"
-D AP_TYPES_CONFIG_FILE="conf/mime.types"
-D SERVER_CONFIG_FILE="conf/httpd.conf"
root@ip-172-31-45-203 ~]#
```

END TO END DEPLOYMENT OF ONE TIER WEB APPLICATION USING AWS LINUX INSTANCE

Or

```
[root@ip-172-31-45-203 ~]# yum info httpd
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Installed Packages
Name           : httpd
Arch            : x86_64
Version        : 2.4.55
Release        : 1.amzn2
Size           : 4.1 M
Repo           : installed
From repo      : amzn2-core
Summary        : Apache HTTP Server
URL            : https://httpd.apache.org/
License        : ASL 2.0
Description    : The Apache HTTP Server is a powerful, efficient, and extensible
                  : web server.

[root@ip-172-31-45-203 ~]#
```

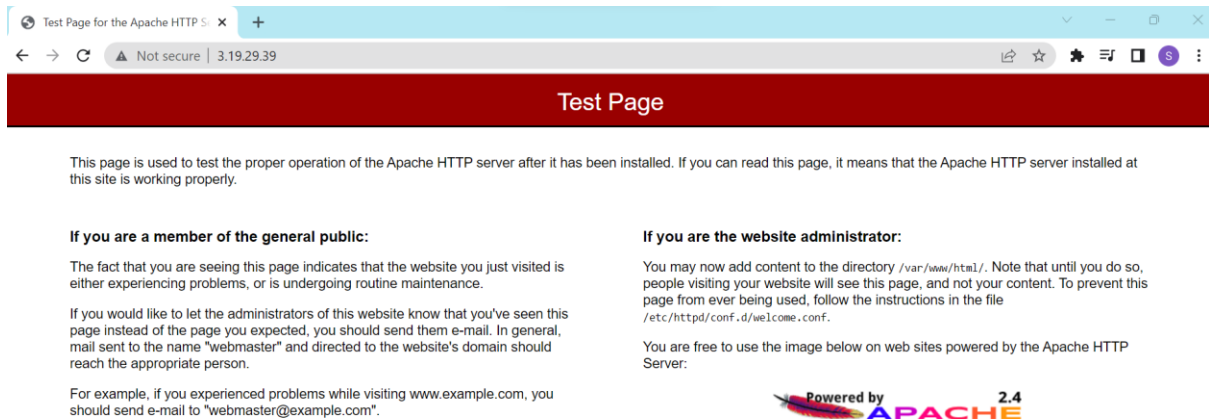
Step 12: Start the installed httpd service using ‘systemctl start httpd’. After starting check the status to ensure it by using command “service status httpd”

```
[root@ip-172-31-45-203 ~]# systemctl start httpd
[root@ip-172-31-45-203 ~]# service httpd status
Redirecting to /bin/systemctl status httpd.service
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)
   Active: active (running) since Sun 2023-03-05 01:18:50 UTC; 38s ago
     Docs: man:httpd.service(8)
  Main PID: 3743 (httpd)
   Status: "Total requests: 0; Idle/Busy workers 100/0; Requests/sec: 0; Bytes served/sec: 0 B/sec"
    CGroup: /system.slice/httpd.service
            └─3743 /usr/sbin/httpd -DFOREGROUND
              └─3744 /usr/sbin/httpd -DFOREGROUND
                └─3745 /usr/sbin/httpd -DFOREGROUND
                  └─3746 /usr/sbin/httpd -DFOREGROUND
                    └─3747 /usr/sbin/httpd -DFOREGROUND
                      └─3748 /usr/sbin/httpd -DFOREGROUND

Mar 05 01:18:50 ip-172-31-45-203.us-east-2.compute.internal systemd[1]: Starting The Apache HTTP Server...
Mar 05 01:18:50 ip-172-31-45-203.us-east-2.compute.internal systemd[1]: Started The Apache HTTP Server.
[root@ip-172-31-45-203 ~]#
```

Now copy and past the Public ip of your Ec2 Linux instance and paste it in browser. Apache webserver is launched successfully.

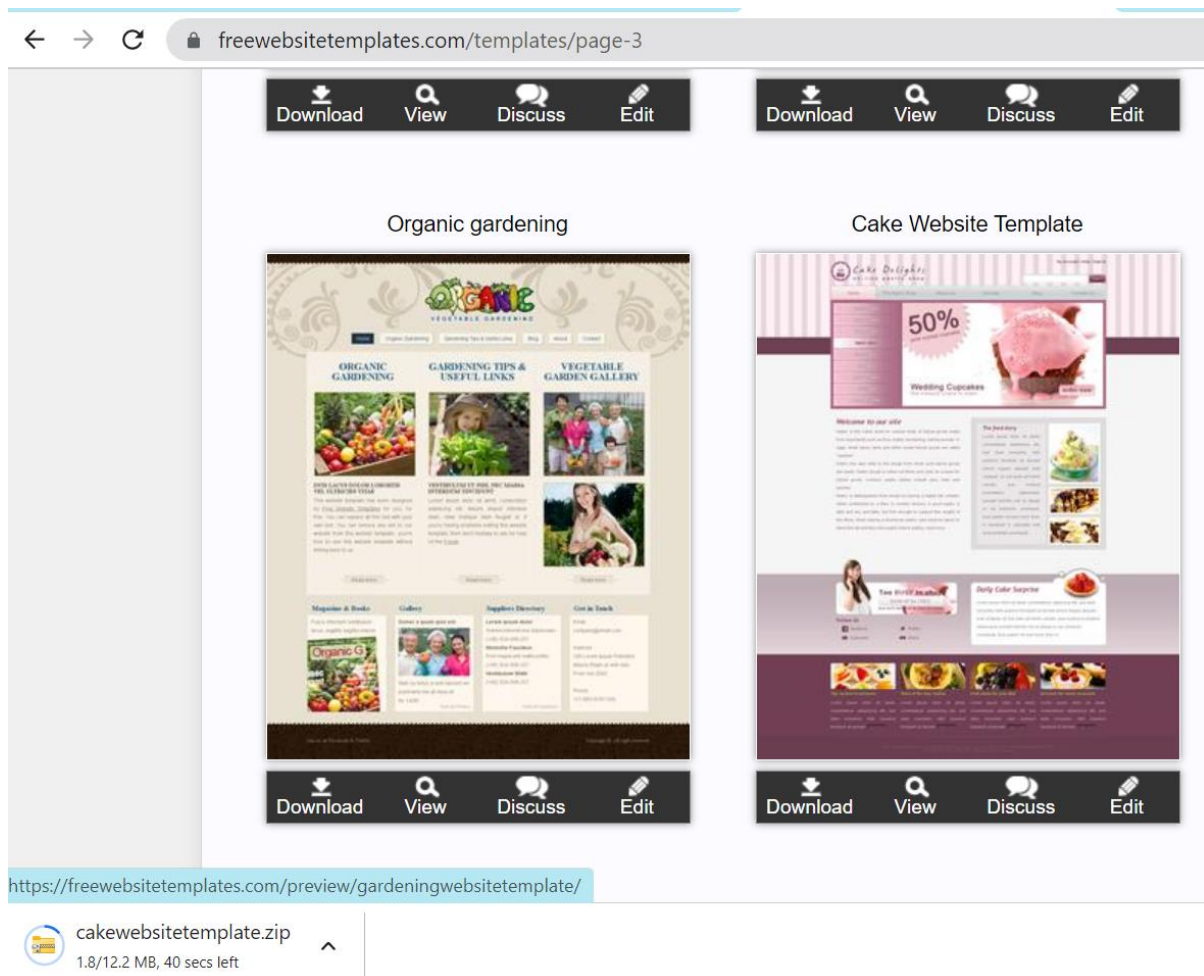
END TO END DEPLOYMENT OF ONE TIER WEB APPLICATION USING AWS LINUX INSTANCE



We have successfully launched default webserver for linux instance. Next, In realtime over this real application/developer written app will be launched. Here I'm going to show simple freeware application to launch/deploy over EC2 Linux server.

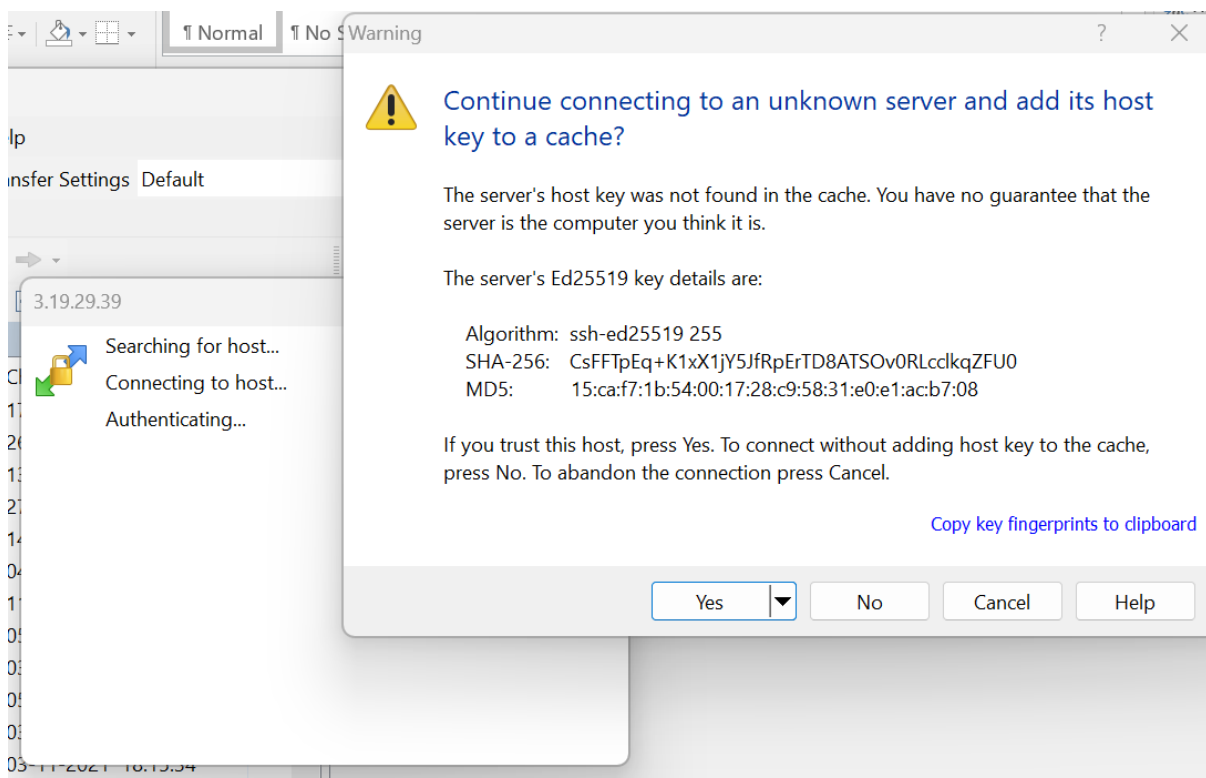
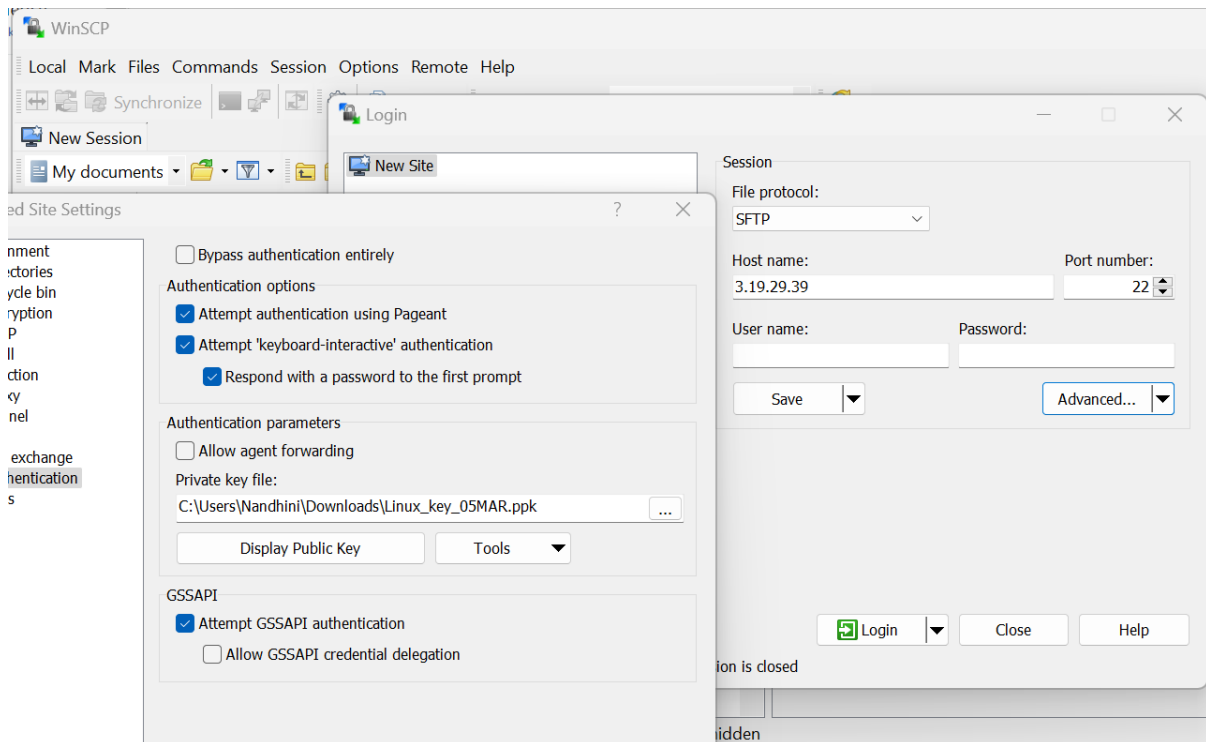
Step 14: To have opensource application to deploy over your server/EC2 instance ,go to this site <https://freewebsitetemplates.com/> and download any Application.

END TO END DEPLOYMENT OF ONE TIER WEB APPLICATION USING AWS LINUX INSTANCE



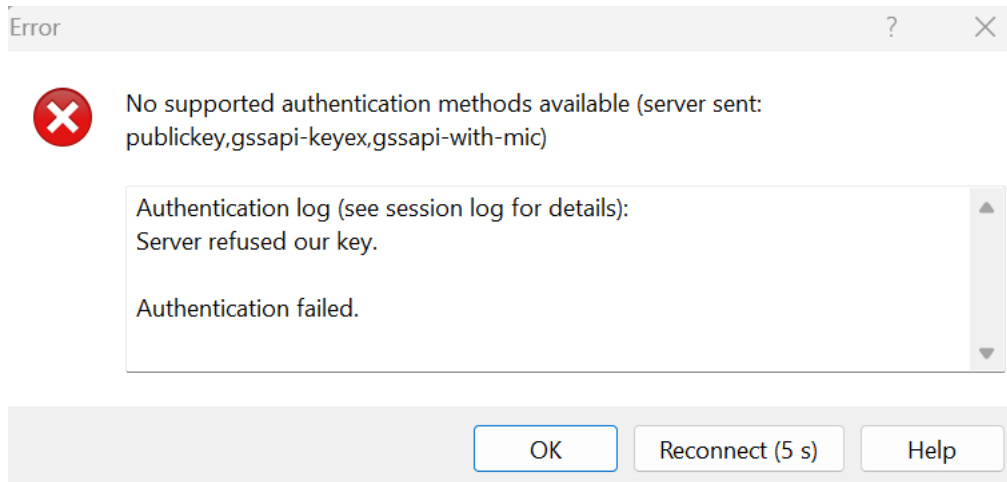
One need to extract and can Copy the downloaded application to Linux server using Winscp ssh client. Give public ip as winscp's username and select advanced option and go to ssh option on your left most corner and chose auth and provide your ppk file. And give yes to further proceed.

END TO END DEPLOYMENT OF ONE TIER WEB APPLICATION USING AWS LINUX INSTANCE



END TO END DEPLOYMENT OF ONE TIER WEB APPLICATION USING AWS LINUX INSTANCE

One may face this error ,



One need to place that downloaded application files in the path where apache is launched. i.e /var/www/html ,so using winscp,one need to copy those downloaded free app in this path /var/www/html.for that /var/www/html give full permission to allow copy from external machine.i.e from your laptop/machine where you downloaded your application.

(Note:Apache installed path is shown in your launched page from there one can refer the installed path)

Test Page

er it has been installed. If you can read this page, it means that the Apache HTTP server installed at

If you are the website administrator:

ed is You may now add content to the directory `/var/www/html/`. Note that until you do so, people visiting your website will see this page, and not your content. To prevent this page from ever being used, follow the instructions in the file `/etc/httpd/conf.d/welcome.conf`.

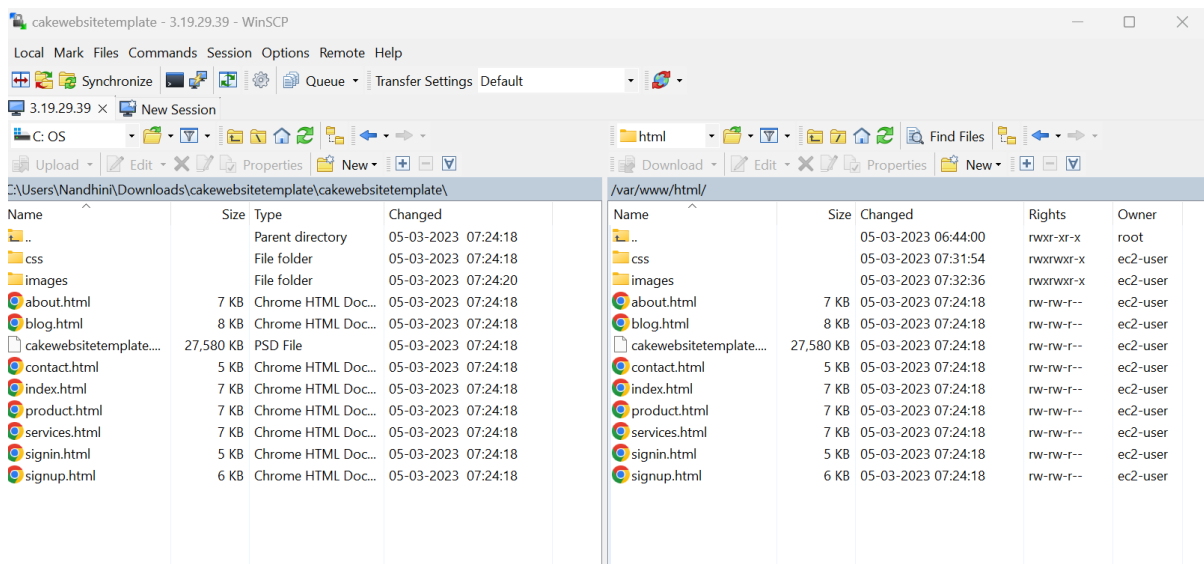
Before launching your application no files present in that path.

END TO END DEPLOYMENT OF ONE TIER WEB APPLICATION USING AWS LINUX INSTANCE

```
drwxr-xr-x 2 root root 6 Feb  9 18:42 cgi-bin
[root@ip-172-31-45-203 www]# chmod 777 /var/www/html/
[root@ip-172-31-45-203 www]# ls -lrt
total 0
drwxrwxrwx 2 root root 6 Feb  9 18:42 html
drwxr-xr-x 2 root root 6 Feb  9 18:42 cgi-bin
[root@ip-172-31-45-203 www]#
```

Now after giving full permission, one can scp the downloaded application file from local machine to remote. By default you will be dropped in home path of ec2 machine. i.e /home/ec2-user

Step 15: Navigate to /var/www/html/ and drag and drop your application. Now one can see the application deployed in the path /var/www/html.



```
[root@ip-172-31-45-203 html]# pwd
/var/www/html
[root@ip-172-31-45-203 html]# ls -lrt
total 27648
-rw-rw-r-- 1 ec2-user ec2-user 5241 Mar  5 01:54 signup.html
-rw-rw-r-- 1 ec2-user ec2-user 4951 Mar  5 01:54 signin.html
-rw-rw-r-- 1 ec2-user ec2-user 6752 Mar  5 01:54 services.html
-rw-rw-r-- 1 ec2-user ec2-user 6466 Mar  5 01:54 product.html
-rw-rw-r-- 1 ec2-user ec2-user 6982 Mar  5 01:54 index.html
-rw-rw-r-- 1 ec2-user ec2-user 5097 Mar  5 01:54 contact.html
-rw-rw-r-- 1 ec2-user ec2-user 28241047 Mar  5 01:54 cakewebsitetemplate.psd
-rw-rw-r-- 1 ec2-user ec2-user 8156 Mar  5 01:54 blog.html
-rw-rw-r-- 1 ec2-user ec2-user 7132 Mar  5 01:54 about.html
drwxrwxr-x 2 ec2-user ec2-user 68 Mar  5 02:01 css
drwxrwxr-x 2 ec2-user ec2-user 4096 Mar  5 02:02 images
[root@ip-172-31-45-203 html]#
```

END TO END DEPLOYMENT OF ONE TIER WEB APPLICATION USING AWS LINUX INSTANCE

Step 16: Now go back to browser and refresh the page. One tier static web Application is deployed successfully

