15-Aug-2024

Internship Day - 1 Report:

What is AWS Cloud?

AWS Cloud, or Amazon Web Services Cloud, is a comprehensive platform of cloud computing services offered by Amazon. It allows individuals, companies, and governments to access computing resources (like servers, storage, databases, networking, and software) over the internet, rather than managing their own hardware and software.

Here's an overview of what AWS Cloud offers and how it works:

1. Infrastructure as a Service (IaaS):

- AWS provides raw computing resources, such as **Amazon EC2** (Elastic Compute Cloud) for virtual servers, **Amazon S3** (Simple Storage Service) for object storage, and **Amazon RDS** (Relational Database Service) for managed databases.
- With IaaS, users can scale up or down resources based on demand, pay only for what they use, and avoid managing physical infrastructure.

2. Platform as a Service (PaaS):

- AWS offers managed services for deploying and scaling applications, like **AWS Elastic Beanstalk** and **AWS Lambda** for serverless computing.
- With PaaS, users can focus on application code, while AWS handles provisioning, scaling, and maintenance.

3. Software as a Service (SaaS):

- AWS also provides managed software solutions, including **AWS Glue** for data integration and **Amazon Recognition** for image and video analysis.
- SaaS allows users to leverage AWS's advanced software offerings without needing to build and maintain them.

4. Global Infrastructure:

- AWS has data centers around the world, organized into **Regions** and **Availability Zones**, offering low-latency and highly resilient services.
- Users can deploy resources close to their end-users to reduce latency and improve performance.

5. Security and Compliance:

- AWS provides a secure platform with encryption, access management, and compliance certifications.
- Tools like **AWS Identity and Access Management (IAM)** help secure resources and control access.

6. Use Cases:

- AWS Cloud is used for a wide range of applications, from hosting websites and applications to running data analytics, machine learning, IoT, and more.

Key Benefits:

- Scalability: Instantly scale resources to match demand.
- Cost Efficiency: Pay-as-you-go pricing reduces the need for large upfront investments.
- Reliability: Highly available and resilient infrastructure with built-in redundancy.
- Flexibility: Support for a vast array of services, APIs, and integrations.

AWS is widely used by startups, large enterprises, and governments because of its flexibility, security, and ability to scale.

Create AWS Cloud Account:

Here are the steps to create an AWS account:

Step 1: Visit the AWS Sign-Up Page:

- Go to aws.amazon.com (https://aws.amazon.com/) and click on "Create an AWS Account" in the top right corner.

Step 2. Enter Your Email and Create a Password:

- Provide a valid email address, choose an account name, and create a strong password. This will be used as your root account.

Step 3. Select the AWS Account Type:

- Choose Personal (for individual use) or Professional (for business use), and fill in the required information, including contact details.

Step 4. Add Payment Information:

- Enter a valid credit or debit card. AWS will charge a small, refundable fee (around $1) to verify the payment method.

Step 5. Verify Your Identity:

- Complete the phone verification by receiving a call or SMS with a verification code.

Step 6. Select a Support Plan:

- Choose a support plan (Basic Support is free, with paid options for advanced support).

Step 7. Activate Your Account:

- After completing the steps, AWS will process your information and activate your account, which may take a few minutes. Once activated, you can sign in and start using AWS services.

AWS offers a Free Tier for new accounts, allowing you to try many services at no cost for the first 12 months.

16-Aug-2024

Internship Day - 2 Report:

What is PuTTy Software?

PuTTY is a free, open-source SSH (Secure Shell) and Telnet client used primarily to remotely connect and manage servers. Available for Windows, it provides a secure way to access command-line interfaces on remote computers, commonly used for managing Linux servers. PuTTY also supports other protocols, like SCP and SFTP, for secure file transfers, and includes tools like PuTTYgen for key generation.

To install PuTTY on Windows:

1. Download the Installer:

- Go to the official PuTTY website (https://www.putty.org/), navigate to "Download PuTTY," and download the latest Windows version.

2. Run the Installer:

- Locate the downloaded `.msi` file, double-click to run it, and follow the setup prompts.

3. Choose Installation Options:

- Select the installation location (default is usually fine), and choose to create a desktop shortcut if desired. You may need admin rights to complete the installation.

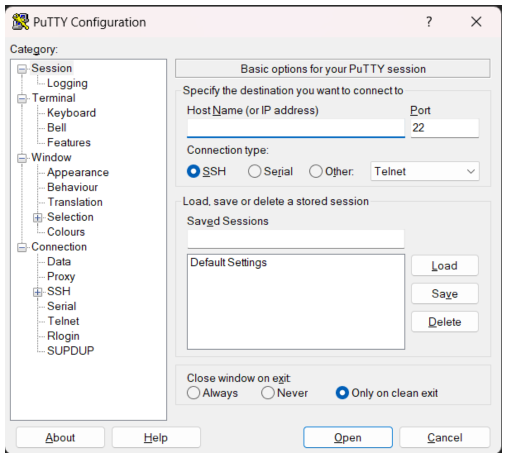
4. Complete Installation:

Once installation finishes, click “Finish.”

5. Verify and Launch PuTTY:

- Open PuTTY from your desktop or Start menu, set up your SSH connection (hostname, port, etc.), and click “Open” to connect.

PuTTY is now installed and ready for SSH connections on your Windows system.



Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address):

Port: 22

Connection type: [✓ SSH] Serial Other: Telnet

Load, save or delete a stored session

Saved Sessions

Default Settings

Close window on exit: Always Never [✓ Only on clean exit]

What is WinSCP Software?

WinSCP is a free, open-source file transfer client for Windows, used to securely transfer files between a local computer and a remote server. It supports SFTP (SSH File Transfer Protocol), SCP (Secure Copy Protocol), and FTP (File Transfer Protocol). WinSCP provides an intuitive graphical interface for file management, allowing users to drag and drop files, and it also integrates with PuTTY for SSH access to remote servers.

To install WinSCP on Windows:

1. Download the Installer:

- Visit the official WinSCP website (https://winscp.net/eng/download.php) and download the latest version for Windows.

2. Run the Installer:

Locate the downloaded `.exe` file, double-click it, and follow the installation prompts.

3. Choose Setup Options:

Select your preferred installation type (typical setup is recommended), installation location, and interface language.

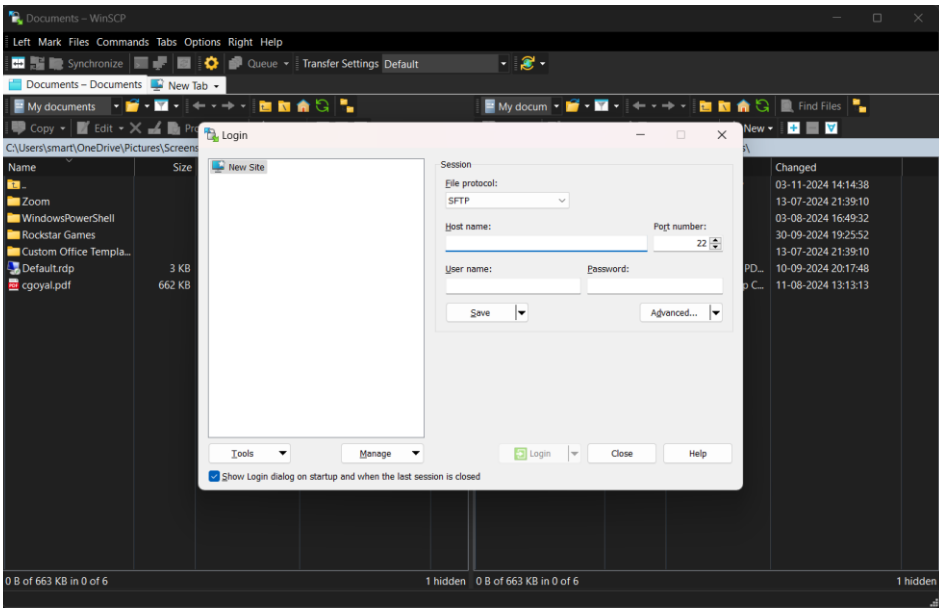
4. Complete Installation:

- Click “Finish” once the setup is complete. Optionally, choose to launch WinSCP immediately.

5. Launch WinSCP and Configure Connection:

- Open WinSCP from the desktop or Start menu, enter your server's hostname, username, password, and protocol (like SFTP or FTP), then click "Login" to connect.

WinSCP is now installed and ready for secure file transfers on your Windows system.



C:\Users\smart\OneDrive\Pictures\Screens

Name	Size	Changed
..		03-11-2024 14:14:38
Zoom		13-07-2024 21:39:10
WindowsPowerShell		03-08-2024 16:49:32
Rockstar Games		30-09-2024 19:25:52
Custom Office Templa...		13-07-2024 21:39:10
Default.rdp	3 KB	10-09-2024 20:17:48
cgoyal.pdf	662 KB	11-08-2024 13:13:13

File protocol: SFTP

Host name:

Port number: 22

User name:

Password:

[✓] Show Login dialog on startup and when the last session is closed

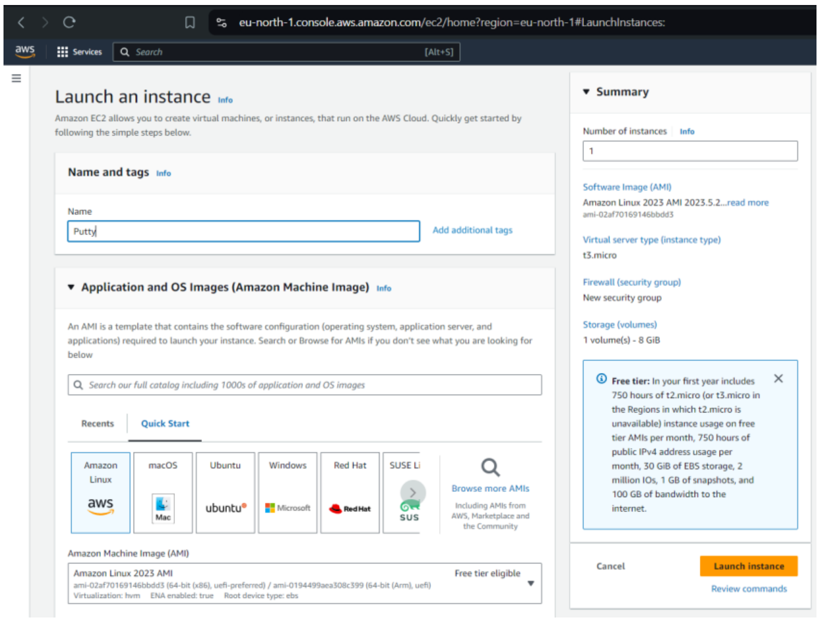
0 B of 663 KB in 0 of 6 1 hidden 0 B of 663 KB in 0 of 6 1 hidden

To launch and connect to an AWS EC2 instance using PuTTY, follow these steps:

1. Launch an EC2 Instance

- **Sign in to AWS Console:** Go to AWS Console (https://aws.amazon.com/console/) and open the **EC2 Dashboard**.

- **Launch Instance:** Select **Launch Instance**, then choose an **Amazon Machine Image (AMI)** (e.g., Amazon Linux 2).



eu-north-1.console.aws.amazon.com/ec2/home?region=eu-north-1#LaunchInstances:

Launch an instance

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.

Name and tags

Name: Putty

Add additional tags

Application and OS Images (Amazon Machine Image)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI — Free tier eligible
ami-02af70169146bbdd3 (64-bit (x86), uefi-preferred) / ami-0194499aea308c399 (64-bit (Arm), uefi)
Virtualization: hvm   ENA enabled: true   Root device type: ebs

Summary

Number of instances: 1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.5.2...read more
ami-02af70169146bbdd3

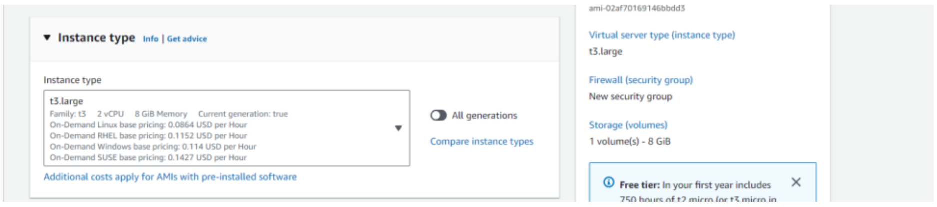
Virtual server type (instance type)
t3.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

- **Instance Type:** Choose an instance type (e.g., t2.micro for the free tier)



Instance type

Instance type
t3.large
Family: t3   2 vCPU   8 GiB Memory   Current generation: true
On-Demand Linux base pricing: 0.0864 USD per Hour
On-Demand RHEL base pricing: 0.1152 USD per Hour
On-Demand Windows base pricing: 0.114 USD per Hour
On-Demand SUSE base pricing: 0.1427 USD per Hour

Additional costs apply for AMIs with pre-installed software

ami-02af70169146bbdd3

Virtual server type (instance type)
t3.large

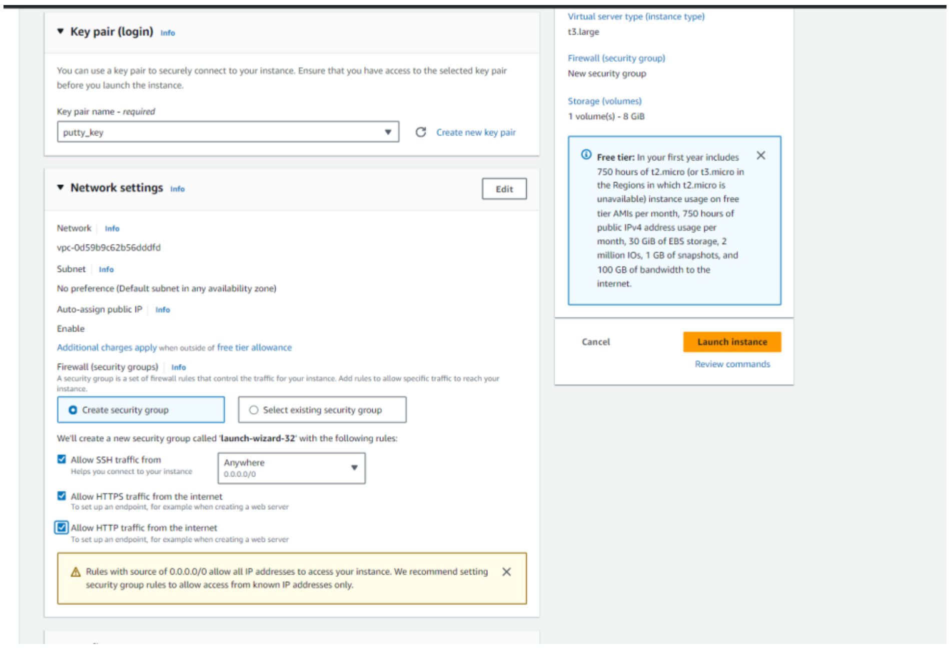
Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year includes

- **Configure Instance:** Set network and storage preferences as needed, then proceed to the **Key Pair** step.

- **Key Pair:** Enter a **Key Pair Name** (e.g., "main"), download the \`.pem\` file (AWS format), and Launch the instance.


Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

putty_key

Create new key pair

Network settings Info

Network | Info

vpc-0d59b9c62b56dddfd

Subnet | Info

No preference (Default subnet in any availability zone)

Auto-assign public IP | Info

Enable

Additional charges apply when outside of free tier allowance

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-32**' with the following rules:

- [✓] Allow SSH traffic from — Anywhere 0.0.0.0/0
  Helps you connect to your instance
- [✓] 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

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.

Virtual server type (instance type)

t3.large

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

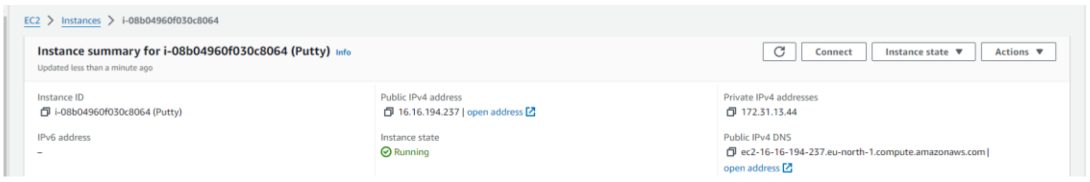
Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

2. Convert Key Pair for PuTTY

- Convert .pem to .ppk: Use PuTTYgen (included with PuTTY) to convert the `.pem` file to a `.ppk` file:
- Open PuTTYgen, click **Load**, and select your `.pem` file.
- Click **Save Private Key** and save the file as `main.ppk`.

3. Get EC2 Instance Details

- **Access Instance ID:** Go back to the EC2 dashboard, click on the **Instance ID** to open instance details.
- **Copy IPv4 Address:** Under **Public IPv4 address**, copy the instance's IP address for later use.


EC2 > Instances > i-08b04960f030c8064

Instance summary for i-08b04960f030c8064 (Putty) Info

Updated less than a minute ago

Instance ID: i-08b04960f030c8064 (Putty)

Public IPv4 address: 16.16.194.237 | open address

Private IPv4 addresses: 172.31.13.44

IPv6 address: –

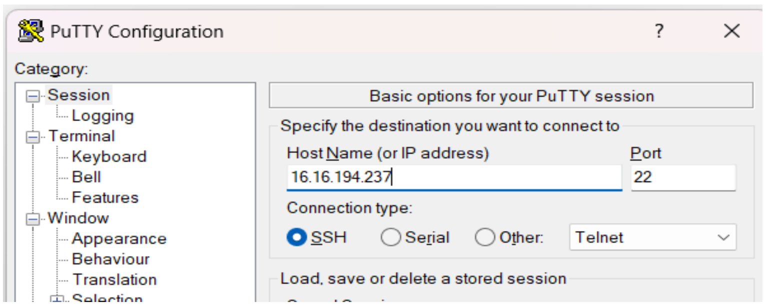
Instance state: Running

Public IPv4 DNS: ec2-16-16-194-237.eu-north-1.compute.amazonaws.com | open address

4. Connect to EC2 Using PuTTY

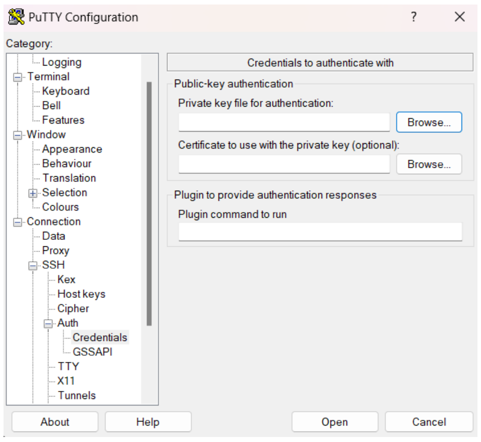
- **Open PuTTY:** Launch PuTTY on your computer.

- **Session Configuration:**
  In **Session**, enter the **IPv4 Address** in the **Host Name** field (e.g., `ec2-user@<public-ip>`).



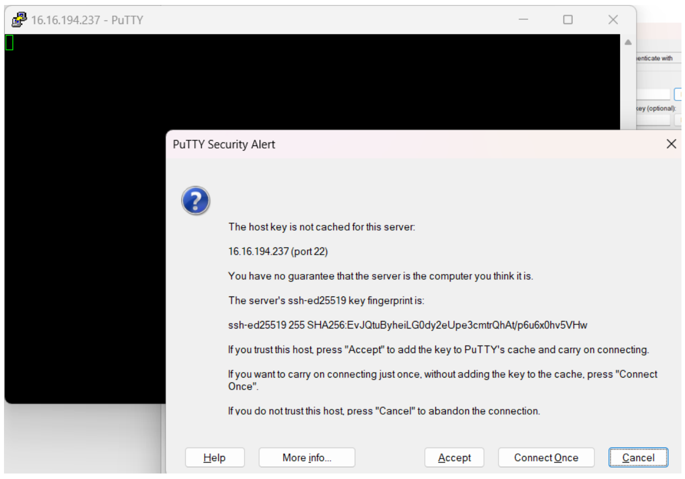
Host Name (or IP address): 16.16.194.237
Port: 22
Connection type: [✓ SSH]

- **Add Private Key:**
  Go to Connection > SSH > Auth.
  Under Credentials, click Browse and select your `main.ppk` file.



5. Save and Open Session

- **Save Session:** Go back to **Session** and enter a name (e.g., "My EC2 Instance") in **Saved Sessions,** then click **Save**.
- **Connect:** Click **Open** to establish an SSH connection to the EC2 instance.

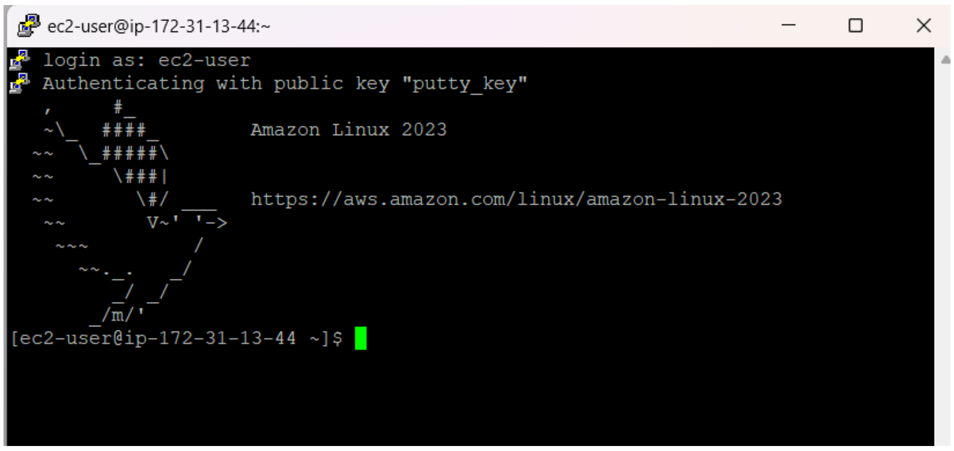


```
The host key is not cached for this server:

16.16.194.237 (port 22)

You have no guarantee that the server is the computer you think it is.

The server's ssh-ed25519 key fingerprint is:

ssh-ed25519 255 SHA256:EvJQtuByheiLG0dy2eUpe3cmtrQhAt/p6u6x0hv5VHw

If you trust this host, press "Accept" to add the key to PuTTY's cache and carry on connecting.

If you want to carry on connecting just once, without adding the key to the cache, press "Connect
Once".

If you do not trust this host, press "Cancel" to abandon the connection.
```

6. SSH Verification and Access

- **Accept Security Alert:** If prompted with a security alert, click **Yes** to accept the host key.

- **Use the Shell:** You are now connected via SSH to your EC2 instance and can use the command-line interface for managing your instance.



```
login as: ec2-user
Authenticating with public key "putty_key"
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-13-44 ~]$ 
```

Day – 18 Aug, 2024

Internship Day - 3 Report:

What is Web Server?

A web server is a software or hardware that serves content to the web. It processes incoming requests from clients (usually web browsers) and delivers web pages, images, videos, and other content over the Internet using the Hypertext Transfer Protocol (HTTP). The main function of a web server is to store, process, and deliver web pages to users.

Introduction of Apache web server:

Apache HTTP Server, commonly referred to as Apache, is one of the oldest and most widely used web servers in the world. It was developed by the Apache Software Foundation and first released in 1995. Here are some key features and characteristics of Apache:

Key Features:

1. **Open Source:** Apache is free to use and distribute, and its source code is available for modification.

2. **Cross-Platform:** It runs on various operating systems, including Unix, Linux, Windows, and macOS.

3. **Modular Architecture:** Apache supports a modular architecture, allowing users to enable or disable specific features through modules. This includes support for various programming languages (e.g., PHP, Python) and features like URL rewriting, authentication, and caching.

4. **Configuration Flexibility**: Apache uses configuration files (httpd.conf, .htaccess) that allow for fine-grained control over server settings, directory permissions, and URL handling.

5. **Robust Community Support:** Due to its long history, Apache has a large community and extensive documentation, making it easier to find support and resources.

6. **Virtual Hosting:** Apache supports both name-based and IP-based virtual hosting, allowing multiple domains to be hosted on a single server.

7. **Security Features**: Apache offers various security modules and features, including SSL/TLS support, authentication, and access control.

Use Cases:

- Ideal for serving dynamic content and applications, especially when combined with languages like PHP or Python.
- Commonly used for hosting websites, web applications, and content management systems (CMS) like WordPress.

Introduction of Nginx web server:

Nginx (pronounced "engine-x") is a high-performance web server and reverse proxy server created by Igor Sysoev and first released in 2004. It has gained significant popularity due to its speed and efficiency. Here are some key features and characteristics of Nginx:

Key Features:

1. **Event-Driven Architecture:** Nginx uses an asynchronous, event-driven architecture, which allows it to handle many connections simultaneously with low resource consumption. This makes it well-suited for high-traffic websites.

2. **Reverse Proxy and Load Balancing:** Nginx is commonly used as a reverse proxy server, distributing incoming traffic among multiple backend servers. It can also perform load balancing, caching, and SSL termination.

3. **Static File Serving:** Nginx excels at serving static content (e.g., images, CSS, JavaScript) quickly and efficiently, making it a popular choice for serving static websites.

4. **Configuration Simplicity:** Nginx configuration files are straightforward and easy to read, which simplifies server setup and management.

5. **High Performance:** Nginx is known for its speed and ability to handle thousands of concurrent connections with minimal resource usage.

6. **Support for HTTP/2 and HTTP/3**: Nginx supports modern web protocols like HTTP/2 and HTTP/3, enhancing performance and security.

7. **Modular Design**: While Nginx does not have the same level of modularity as Apache, it does support various modules for extended functionality, including third-party modules.

Use Cases:

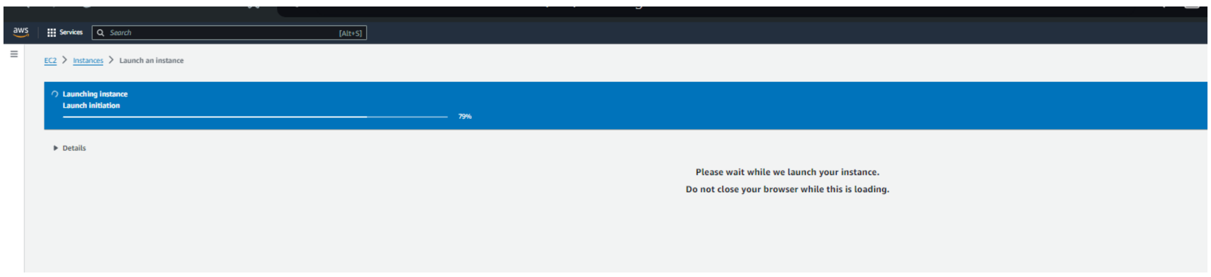
- Often used for serving static websites and as a reverse proxy for dynamic applications.
- Commonly employed in high-traffic environments, such as content delivery networks (CDNs) and large-scale web applications.

Step-by-Step Guide to Set Up and Deploy an HTML Website on an AWS EC2 Instance with Apache

This guide walks through setting up an EC2 instance, installing Apache, and uploading a website template using WinSCP.

1. Launch an EC2 Instance

- **Login to AWS: Access** your AWS console and open **EC2** from the dashboard.
- **Launch Instance:** Choose an Amazon Machine Image (AMI), such as Amazon Linux 2, and select an instance type (e.g., t2.micro).
- **Key Pair:** Enter a **Key Pair Name** (e.g., "main"), download the `.pem` file, and launch the instance.

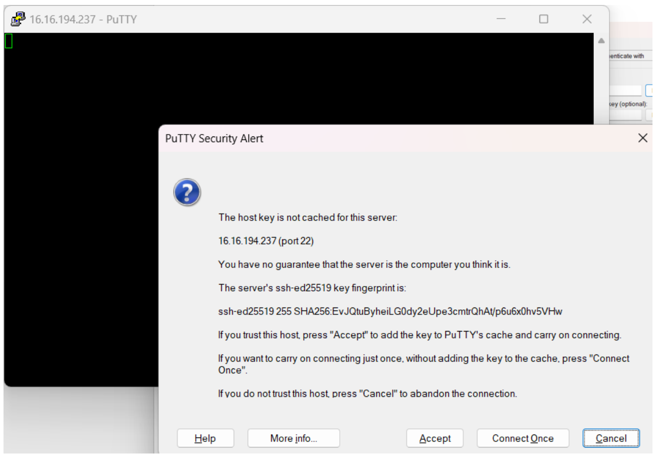


Launching instance
Launch initiation
79%

Please wait while we launch your instance.
Do not close your browser while this is loading.

2. Connect to the EC2 Instance Using PuTTY

- **Convert Key Pair:** Use PuTTYgen to convert your `.pem` file to `.ppk`.
- **Open PuTTy:**
  Enter the **IPv4 address** of your instance in **Host Name** (e.g., `ec2-user@<public-ip>`).
  Go to Connection > SSH >Authand upload the `main.ppk` file.
- **Login as ec2-user:** When the SSH session opens, type `ec2-user` and press **Enter**.



The host key is not cached for this server:

16.16.194.237 (port 22)

You have no guarantee that the server is the computer you think it is.

The server's ssh-ed25519 key fingerprint is:

ssh-ed25519 255 SHA256:EvJQtuByheiLG0dy2eUpe3cmtrQhAt/p6u6x0hv5VHw

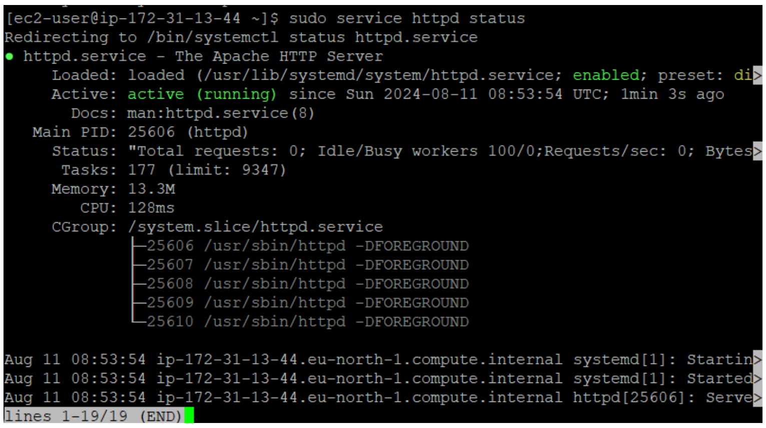
If you trust this host, press "Accept" to add the key to PuTTY's cache and carry on connecting.

If you want to carry on connecting just once, without adding the key to the cache, press "Connect Once".

If you do not trust this host, press "Cancel" to abandon the connection.

3. Install Apache Web Server

- **Update and Install Apache:**
  Run `sudo yum install httpd -y` to install Apache on your EC2 instance.
- **Status Apache Service:**
  Use `sudo service httpd status` to start Apache.

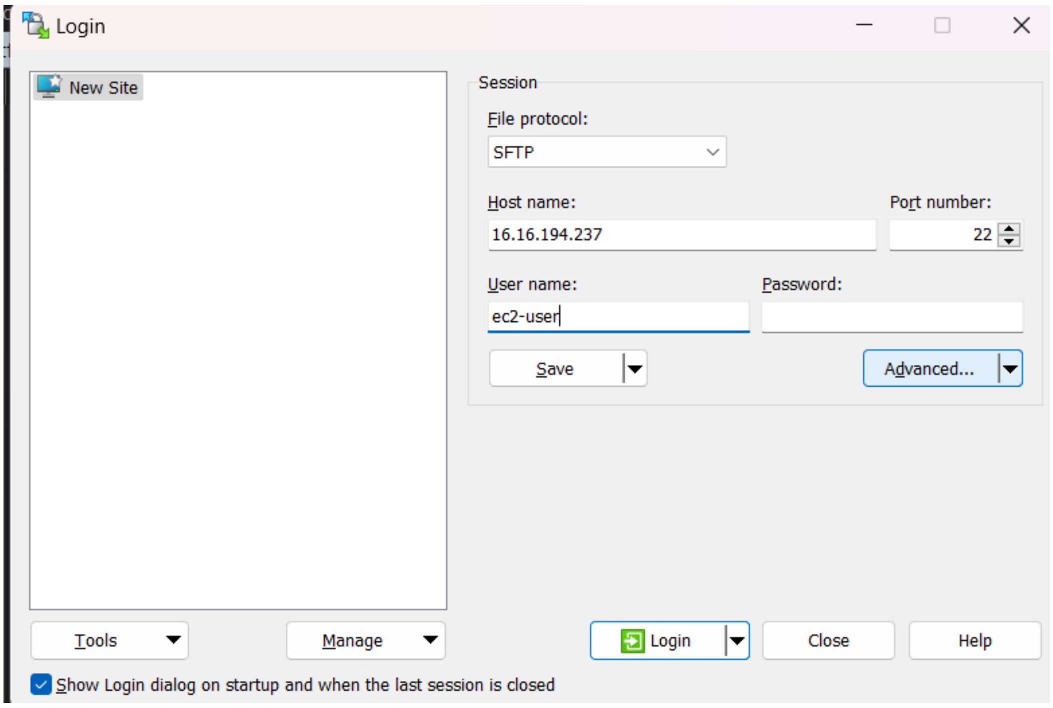


```
[ec2-user@ip-172-31-13-44 ~]$ sudo service httpd status
Redirecting to /bin/systemctl status httpd.service
● httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: di>
     Active: active (running) since Sun 2024-08-11 08:53:54 UTC; 1min 3s ago
       Docs: man:httpd.service(8)
   Main PID: 25606 (httpd)
     Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes>
      Tasks: 177 (limit: 9347)
     Memory: 13.3M
        CPU: 128ms
     CGroup: /system.slice/httpd.service
             ├─25606 /usr/sbin/httpd -DFOREGROUND
             ├─25607 /usr/sbin/httpd -DFOREGROUND
             ├─25608 /usr/sbin/httpd -DFOREGROUND
             ├─25609 /usr/sbin/httpd -DFOREGROUND
             └─25610 /usr/sbin/httpd -DFOREGROUND

Aug 11 08:53:54 ip-172-31-13-44.eu-north-1.compute.internal systemd[1]: Startin>
Aug 11 08:53:54 ip-172-31-13-44.eu-north-1.compute.internal systemd[1]: Started>
Aug 11 08:53:54 ip-172-31-13-44.eu-north-1.compute.internal httpd[25606]: Serve>
lines 1-19/19 (END)
```

Verify by visiting `http://<instance-public-ip>` in a browser.

4. Prepare to Upload HTML Files with WinSCP

- **Download an HTML Template:**
  Visit ThemeWagon (https://themewagon.com/theme-tag/html5-css3/) and download an HTML5 template to your computer.
- **Configure WinSCP:**
  Open **WinSCP** and enter your **EC2 instance's IPv4 address** with **port 22**.



Login

Session

File protocol: SFTP

Host name: 16.16.194.237

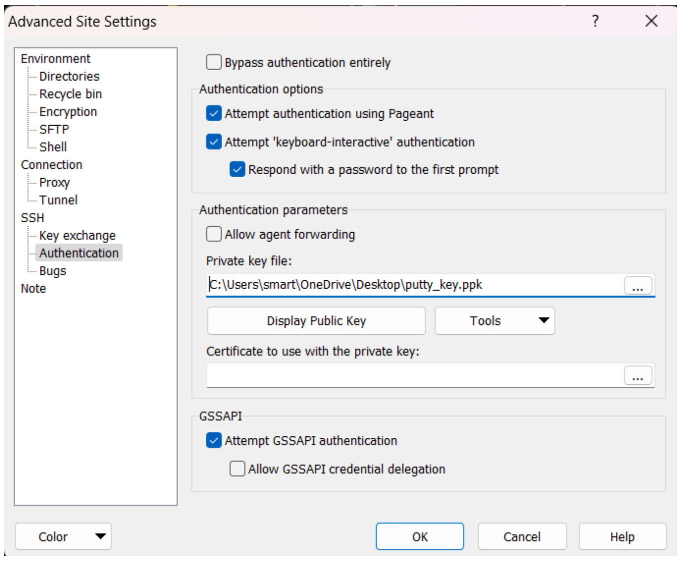
Port number: 22

User name: ec2-user

Password:

New Site
Save
Advanced...
Tools
Manage
Login
Close
Help
Show Login dialog on startup and when the last session is closed

- Go to **Advanced Settings** > **Authentication** and upload your `.ppk` file.


Advanced Site Settings

- [ ] Bypass authentication entirely

Authentication options
- [✓] Attempt authentication using Pageant
- [✓] Attempt 'keyboard-interactive' authentication
  - [✓] Respond with a password to the first prompt

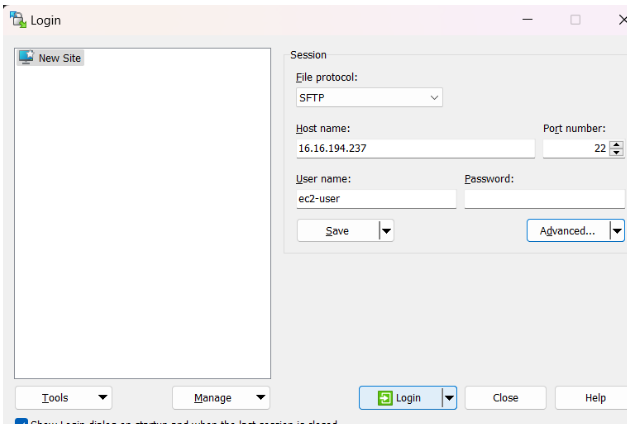
Authentication parameters
- [ ] Allow agent forwarding

Private key file: C:\Users\smart\OneDrive\Desktop\putty_key.ppk

Certificate to use with the private key:

GSSAPI
- [✓] Attempt GSSAPI authentication
  - [ ] Allow GSSAPI credential delegation

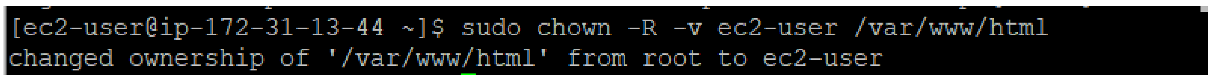
- **Connect:** Log in, and your local files will appear on the left (LHS), and the server files on the right (RHS).


Login

Session
File protocol: SFTP
Host name: 16.16.194.237
Port number: 22
User name: ec2-user
Password:

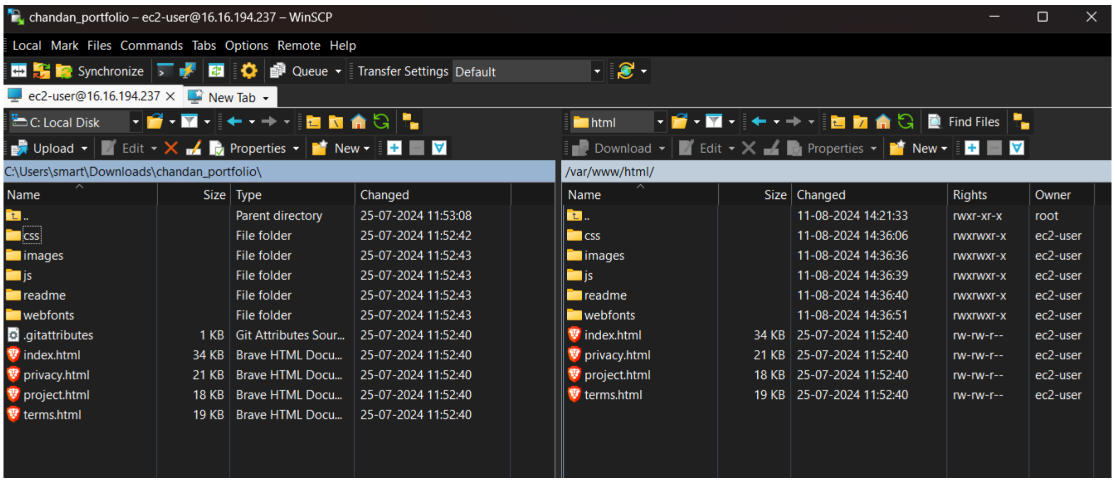
5. Set Up the Directory for HTML Files

- **Modify Directory Permissions:**
  In the SSH terminal, run `sudo chown -R -v ec2-user /var/www/html` to change permissions.


```
[ec2-user@ip-172-31-13-44 ~]$ sudo chown -R -v ec2-user /var/www/html
changed ownership of '/var/www/html' from root to ec2-user
```

- **Upload HTML Template:**
  In WinSCP, navigate to `/var/www/html` on the RHS.

  Drag and drop your HTML files from your local system (LHS) into the `/var/www/html` directory on the server.

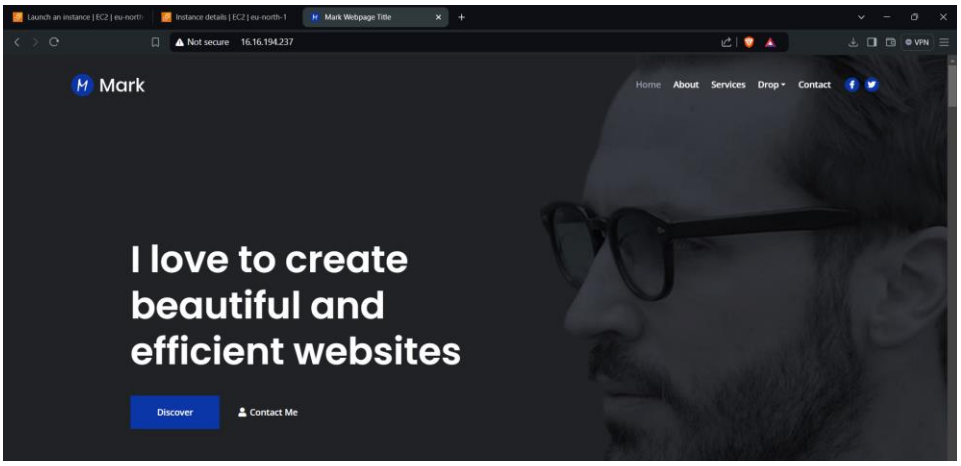


```
chandan_portfolio – ec2-user@16.16.194.237 – WinSCP
Local  Mark  Files  Commands  Tabs  Options  Remote  Help
Synchronize   Queue   Transfer Settings  Default
ec2-user@16.16.194.237   New Tab

C: Local Disk                                          html
Upload  Edit  Properties  New                          Download  Edit  Properties  New   Find Files
C:\Users\smart\Downloads\chandan_portfolio\            /var/www/html/

Name            Size   Type                Changed               Name           Size   Changed               Rights      Owner
..                     Parent directory    25-07-2024 11:53:08   ..                    11-08-2024 14:21:33   rwxr-xr-x   root
css                    File folder         25-07-2024 11:52:42   css                   11-08-2024 14:36:06   rwxrwxr-x   ec2-user
images                 File folder         25-07-2024 11:52:43   images                11-08-2024 14:36:36   rwxrwxr-x   ec2-user
js                     File folder         25-07-2024 11:52:43   js                    11-08-2024 14:36:39   rwxrwxr-x   ec2-user
readme                 File folder         25-07-2024 11:52:43   readme                11-08-2024 14:36:40   rwxrwxr-x   ec2-user
webfonts               File folder         25-07-2024 11:52:43   webfonts              11-08-2024 14:36:51   rwxrwxr-x   ec2-user
.gitattributes  1 KB   Git Attributes Sour...25-07-2024 11:52:40 index.html    34 KB  25-07-2024 11:52:40   rw-rw-r--   ec2-user
index.html     34 KB   Brave HTML Docu...  25-07-2024 11:52:40   privacy.html  21 KB   25-07-2024 11:52:40   rw-rw-r--   ec2-user
privacy.html   21 KB   Brave HTML Docu...  25-07-2024 11:52:40   project.html  18 KB   25-07-2024 11:52:40   rw-rw-r--   ec2-user
project.html   18 KB   Brave HTML Docu...  25-07-2024 11:52:40   terms.html    19 KB   25-07-2024 11:52:40   rw-rw-r--   ec2-user
terms.html     19 KB   Brave HTML Docu...  25-07-2024 11:52:40
```

6. Start/Stop Apache and View Your Website

1 **Control Apache Service:**
Start with `sudo service httpd start` and stop with `sudo service httpd stop` as needed.

2 **Verify the Website:**
Visit `http://<instance-public-ip>` to see your uploaded HTML template live.


Launch an instance | EC2 | eu-north
Instance details | EC2 | eu-north-1
Mark Webpage Title
Not secure 16.16.194.237
Mark
Home
About
Services
Drop
Contact
I love to create beautiful and efficient websites
Discover
Contact Me

Your website should now be hosted on AWS EC2 and accessible via the public IP address!

19-Aug-2024

Internship Day - 4 Report:

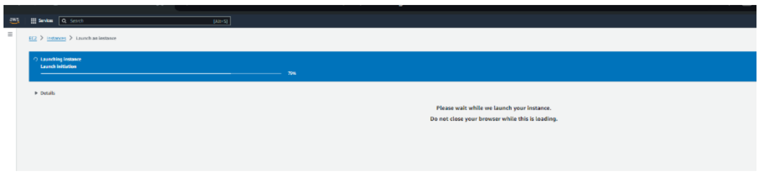
AWS EC2 Setup with Nginx Website Deployment

1. Creating an EC2 Instance

- **Login** to AWS Console.
- **Navigate** to EC2 Dashboard.
- **Click** “Launch Instance”.
- **Configure Instance:**

a) **Name**: nginx-webserver.
b) **AMI**: Amazon Linux 2023.
c) **Instance Type**: t2.micro (free tier).
d) **Key Pair:** Create new (nginx-key), download .pem.
e) **Network Settings:** Allow SSH (22), HTTP (80), HTTPS (443).

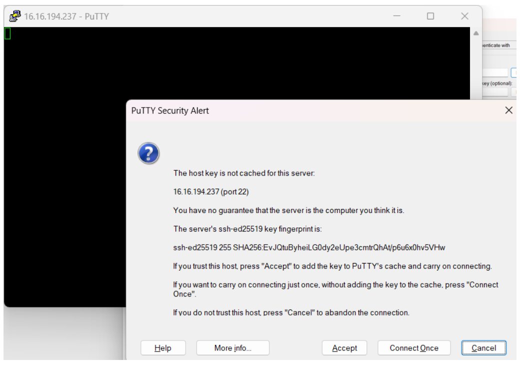
- **Click** “Launch Instance”.



Please wait while we launch your instance.
Do not close your browser while this is loading.

2. Connecting via PuTTY

- Open PuTTY.
- Host Name: ec2-user@[Your-EC2-Public-IP].
- Port: 22.
- SSH → Auth: Browse for your .ppk file.
- Click “Open” to connect.



The host key is not cached for this server:

16.16.194.237 (port 22)

You have no guarantee that the server is the computer you think it is.

The server's ssh-ed25519 key fingerprint is:

ssh-ed25519 255 SHA256:EvJQtuByheiLG0dy2eUpe3cmtrQhAt/p6u6x0hv5VHw

If you trust this host, press "Accept" to add the key to PuTTY's cache and carry on connecting.

If you want to carry on connecting just once, without adding the key to the cache, press "Connect Once".

If you do not trust this host, press "Cancel" to abandon the connection.

3. Installing and Configuring Nginx

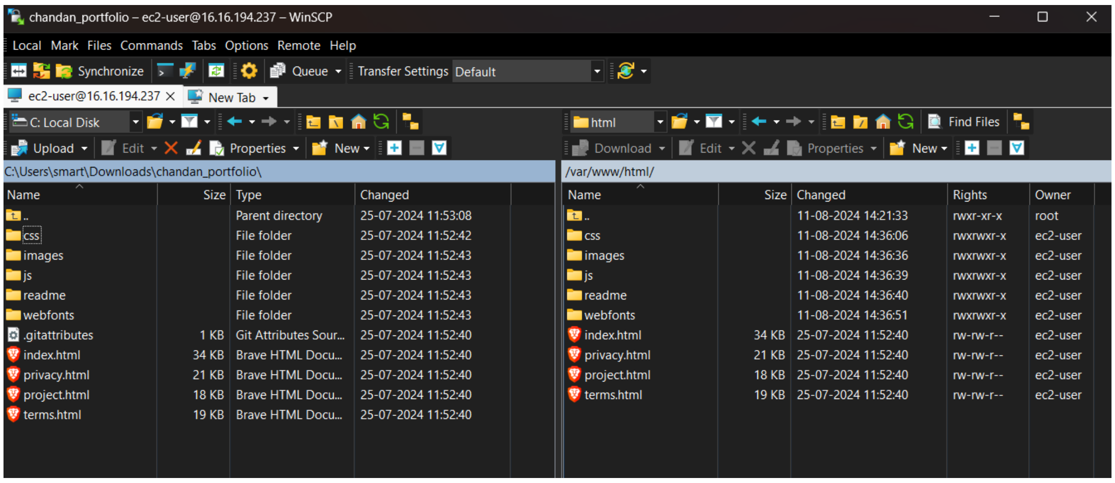
- sudo yum update -y.
- sudo yum install nginx -y.
- sudo systemctl start nginx.
- sudo systemctl enable nginx.
- sudo systemctl status nginx.

4. Setting Up Website Directory

- **sudo chmod -R 755 /usr/share/nginx/html**

5. Uploading Website Files Using WinSCP

- Open WinSCP
- Protocol: SFTP.
- Host name: [Your-EC2-Public-IP].
- Port: 22.
- Username: ec2-user.
- Advanced → Authentication: Browse for your .ppk file.
- Connect and upload files to /usr/share/nginx/html.


chandan_portfolio – ec2-user@16.16.194.237 – WinSCP
Local Mark Files Commands Tabs Options Remote Help
Synchronize
Queue
Transfer Settings Default
ec2-user@16.16.194.237
New Tab
C: Local Disk
Upload
Edit
Properties
New
html
Find Files
Download
Edit
Properties
New

C:\Users\smart\Downloads\chandan_portfolio\

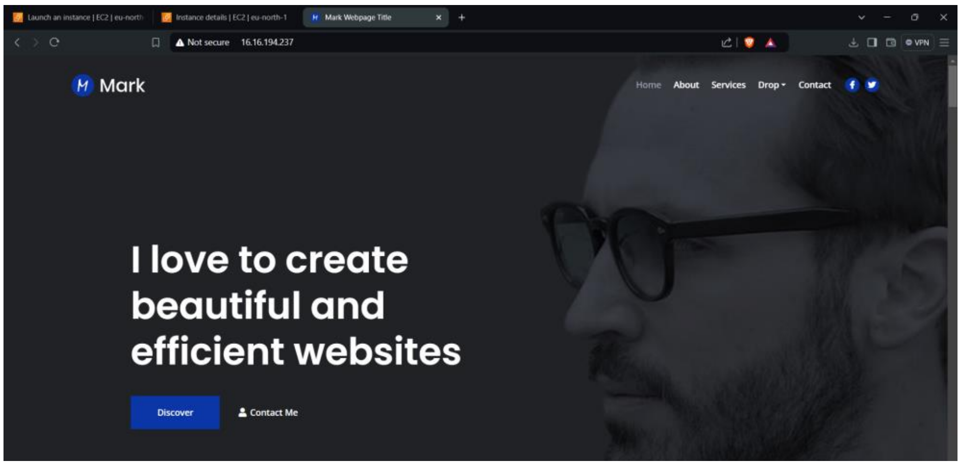
Name	Size	Type	Changed
..		Parent directory	25-07-2024 11:53:08
css		File folder	25-07-2024 11:52:42
images		File folder	25-07-2024 11:52:43
js		File folder	25-07-2024 11:52:43
readme		File folder	25-07-2024 11:52:43
webfonts		File folder	25-07-2024 11:52:43
.gitattributes	1 KB	Git Attributes Sour...	25-07-2024 11:52:40
index.html	34 KB	Brave HTML Docu...	25-07-2024 11:52:40
privacy.html	21 KB	Brave HTML Docu...	25-07-2024 11:52:40
project.html	18 KB	Brave HTML Docu...	25-07-2024 11:52:40
terms.html	19 KB	Brave HTML Docu...	25-07-2024 11:52:40

/var/www/html/

Name	Size	Changed	Rights	Owner
..		11-08-2024 14:21:33	rwxr-xr-x	root
css		11-08-2024 14:36:06	rwxrwxr-x	ec2-user
images		11-08-2024 14:36:36	rwxrwxr-x	ec2-user
js		11-08-2024 14:36:39	rwxrwxr-x	ec2-user
readme		11-08-2024 14:36:40	rwxrwxr-x	ec2-user
webfonts		11-08-2024 14:36:51	rwxrwxr-x	ec2-user
index.html	34 KB	25-07-2024 11:52:40	rw-rw-r--	ec2-user
privacy.html	21 KB	25-07-2024 11:52:40	rw-rw-r--	ec2-user
project.html	18 KB	25-07-2024 11:52:40	rw-rw-r--	ec2-user
terms.html	19 KB	25-07-2024 11:52:40	rw-rw-r--	ec2-user

6. Verifying Setup

- Open Web Browser
- Enter your EC2 public IP.
- Verify your website is visible.



I love to create beautiful and efficient websites