HOW TO ??

HOST A SERVER ON

WITH PROPER STEPS

singh-yuvraj002

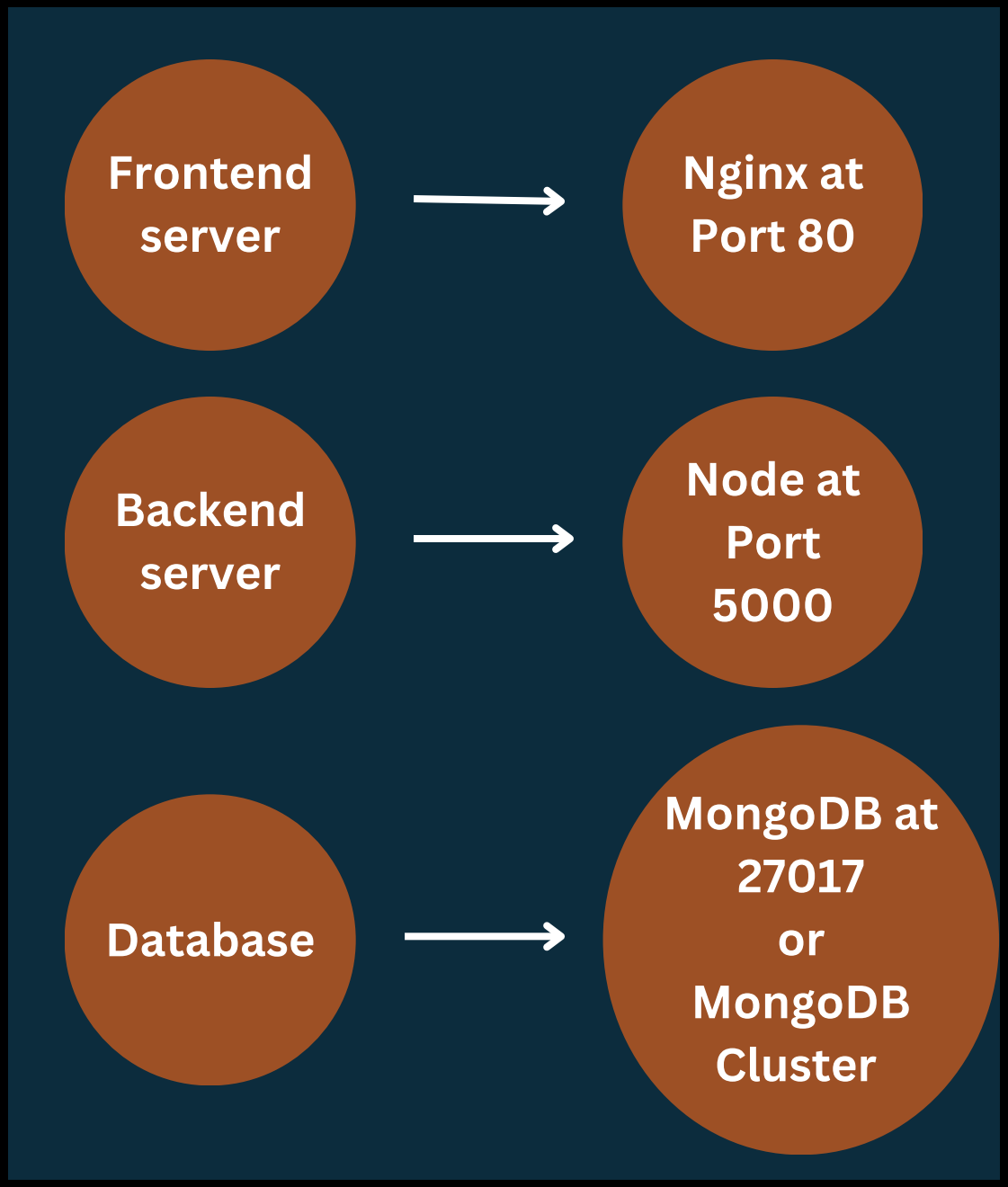
AMAZON EC2

Amazon Elastic Compute Cloud is a part of Amazon.com's cloud-computing platform, it provides you the feature to create an instance, which is a rented virtual computer on which you can run your own computer applications.

Note :- AWS provides 720 hours of free tier on an EC2 instance for new users.

singh-yuvraj002

BLUEPRINT OF SERVER...

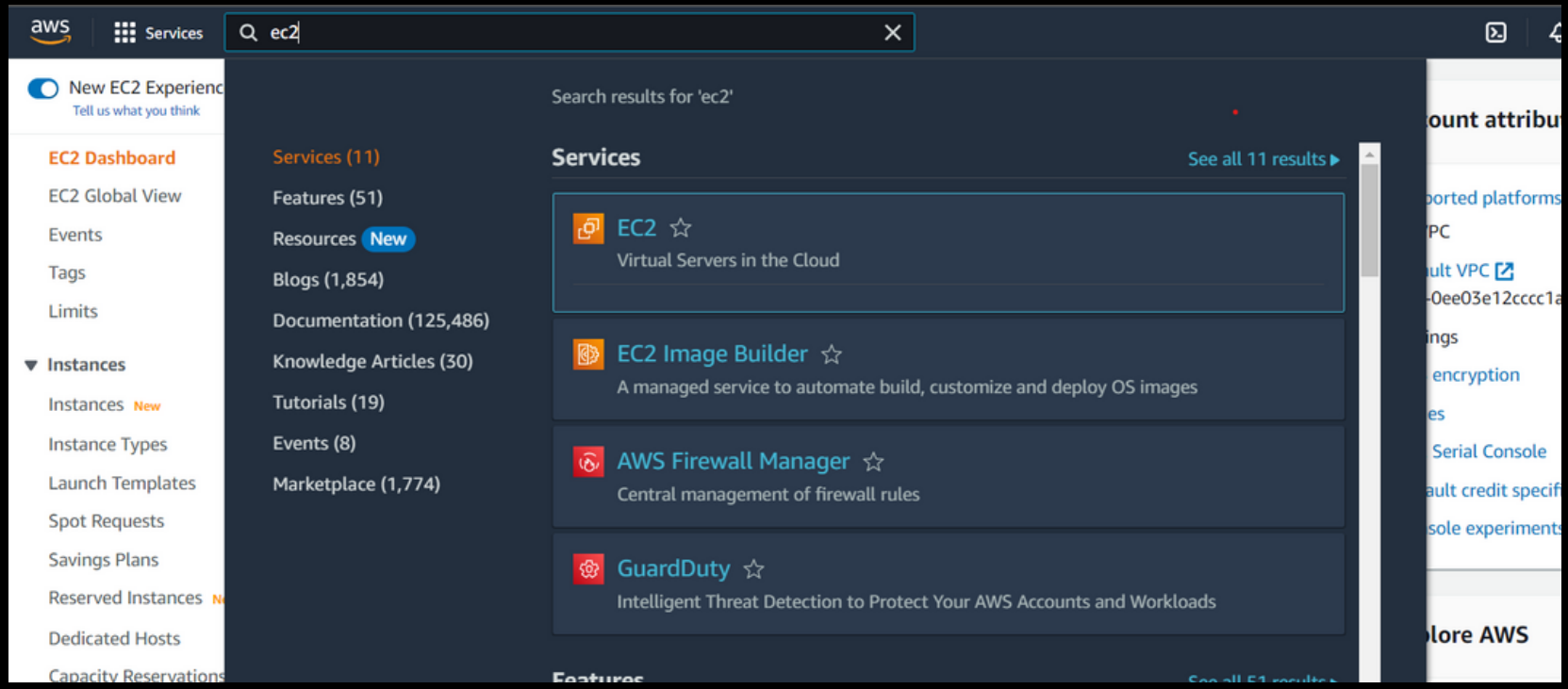


Operating system / System Image of EC2 instance (say Ubuntu)

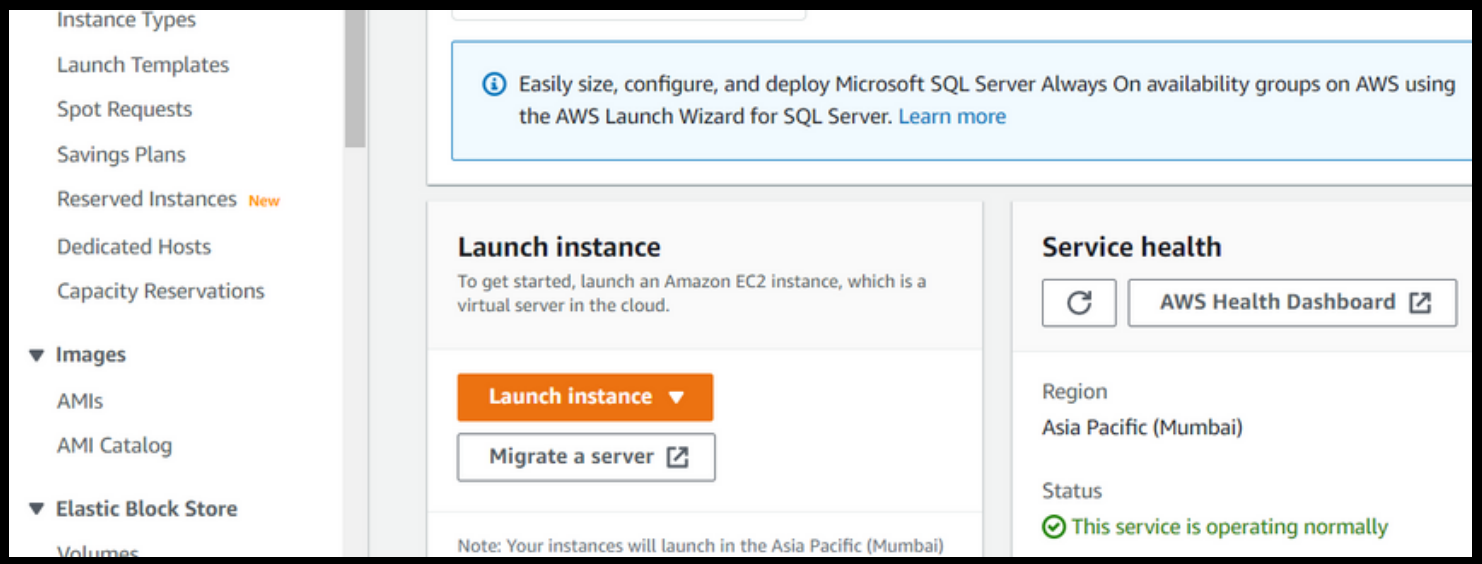
singh-yuvraj002

1. SEARCH FOR EC2 AND CREATE AN INSTANCE ....

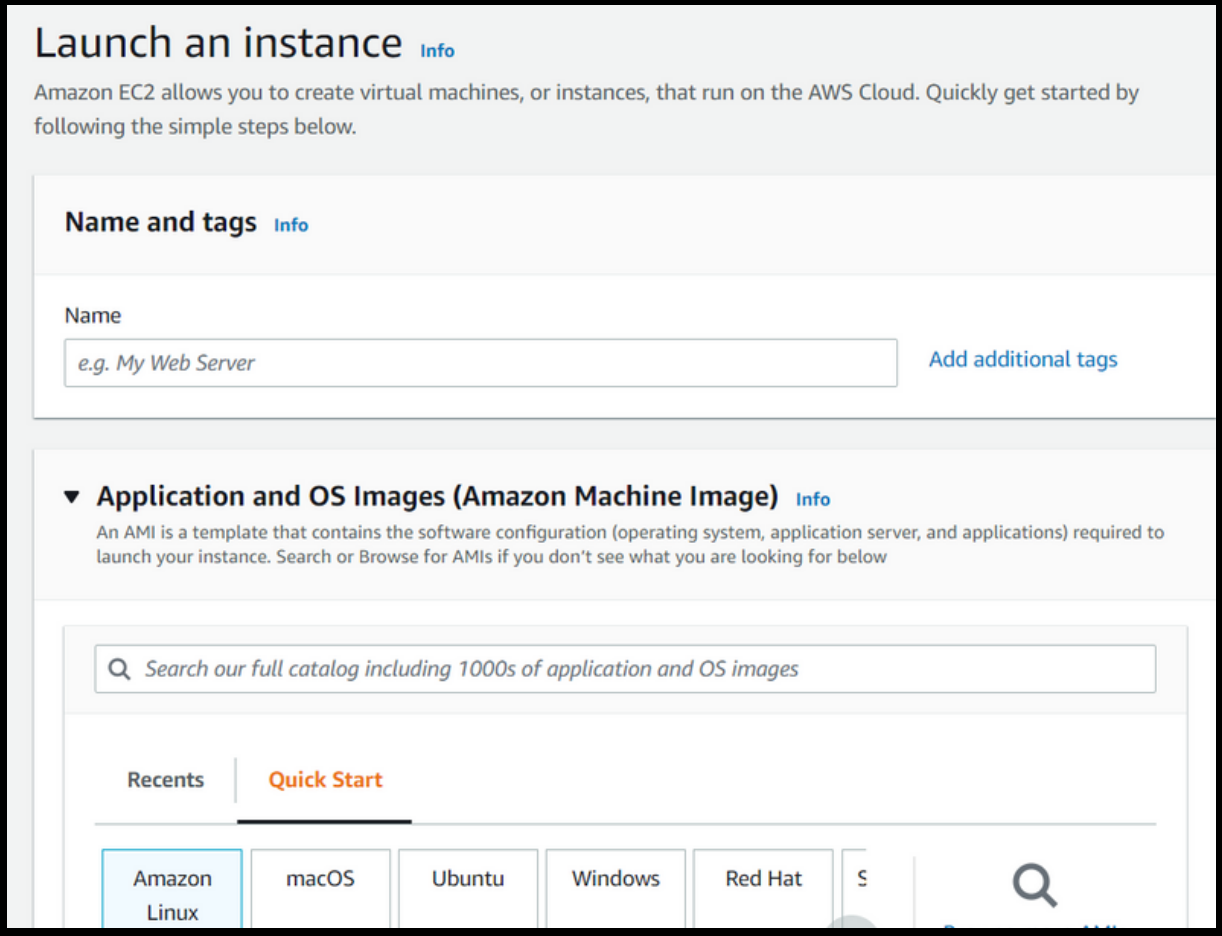
- Sign in into AWS , choose location of server from top right side and then open EC2.



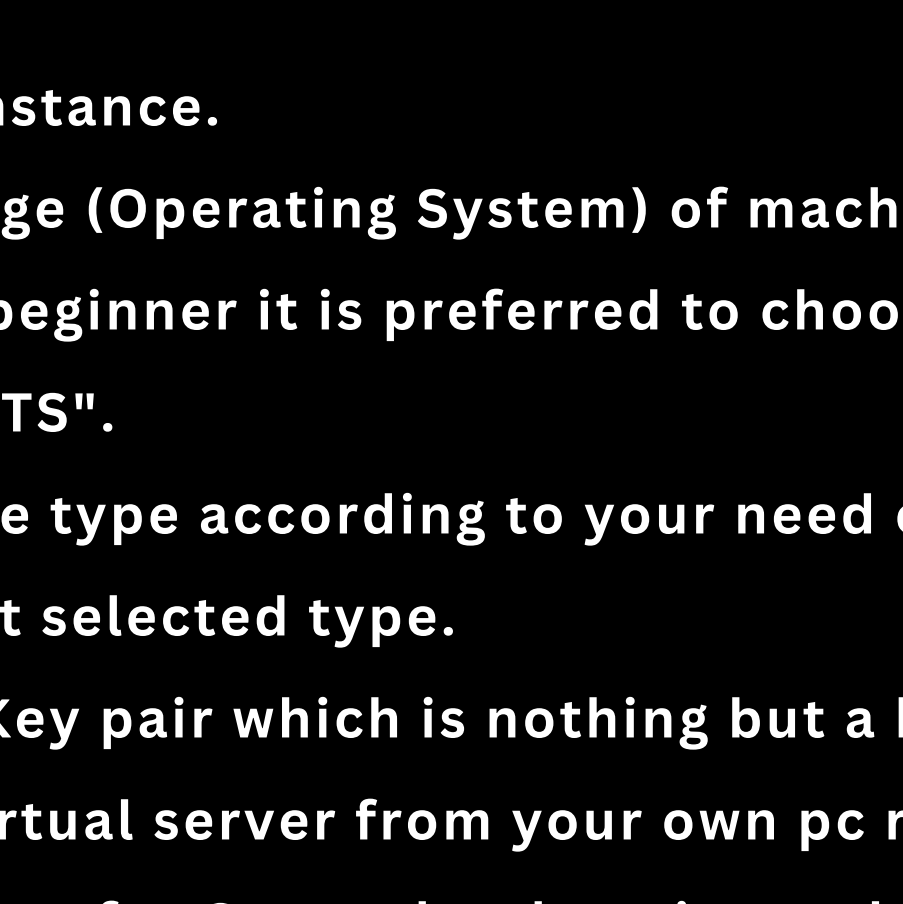
- **Launch new instance.**



- **Fill out the details of the instance one by one carefully as mentioned.**



- Name of the instance.
- Select the image (Operating System) of machine you are renting. As a beginner it is preferred to choose "Ubuntu Server 22.04 LTS".
- Select Instance type according to your need otherwise use the default selected type.
- Create a new Key pair which is nothing but a key to access your virtual server from your own pc remotely (use .pem format for Openssh otherwise .ppk for Putty). Beginner's keep the default settings.



- **Set the security groups settings as follows.**

Type	Protocol	Port	Source	Why??
SSH	TCP	22	0.0.0.0/0	OpenSSH
HTTP	TCP	80	0.0.0.0/0, ::/0	Nginx
Custom TCP	TCP	5000	0.0.0.0/0	Node
Custom TCP	TCP	27017	0.0.0.0/0	Database

- **Finally Launch the instance, wait for some time while the instance gets started.**

2. ACCESSING THE SERVER REMOTELY FROM YOUR SIDE USING TERMINAL....

- Open terminal in the folder where you had stored the key pair file.
- Run the mentioned commands to access cloud server from your device everytime you want.

instance name like ubuntu

```
ssh -i file-name.pem ubuntu@public-dns-of-instance
```

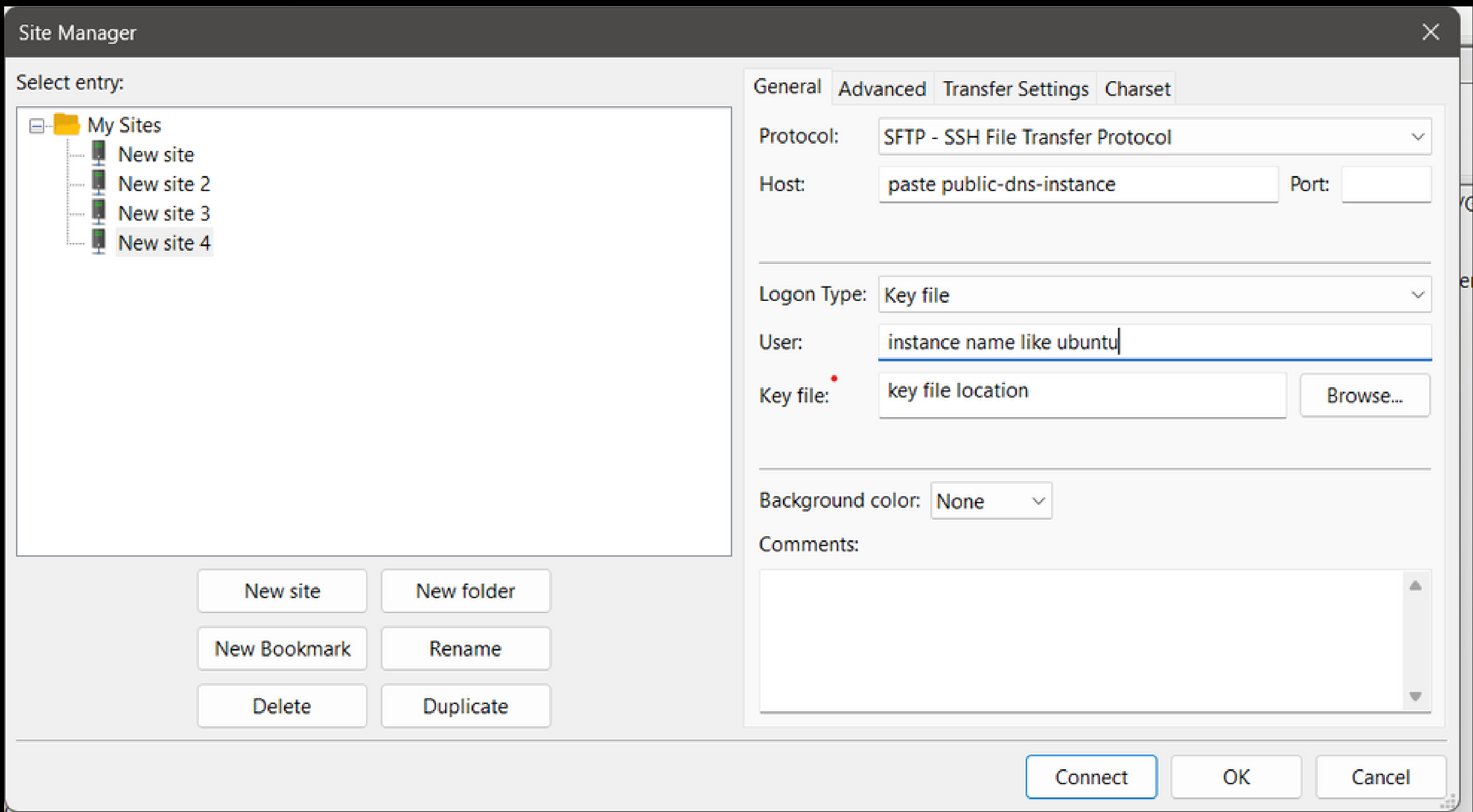
Key File Name

from aws instance

- Confirm to establish fingerprint in your device
- If error occurred like ""permission 0644"" run the command ""chmod 600 ./file-name.pem"".

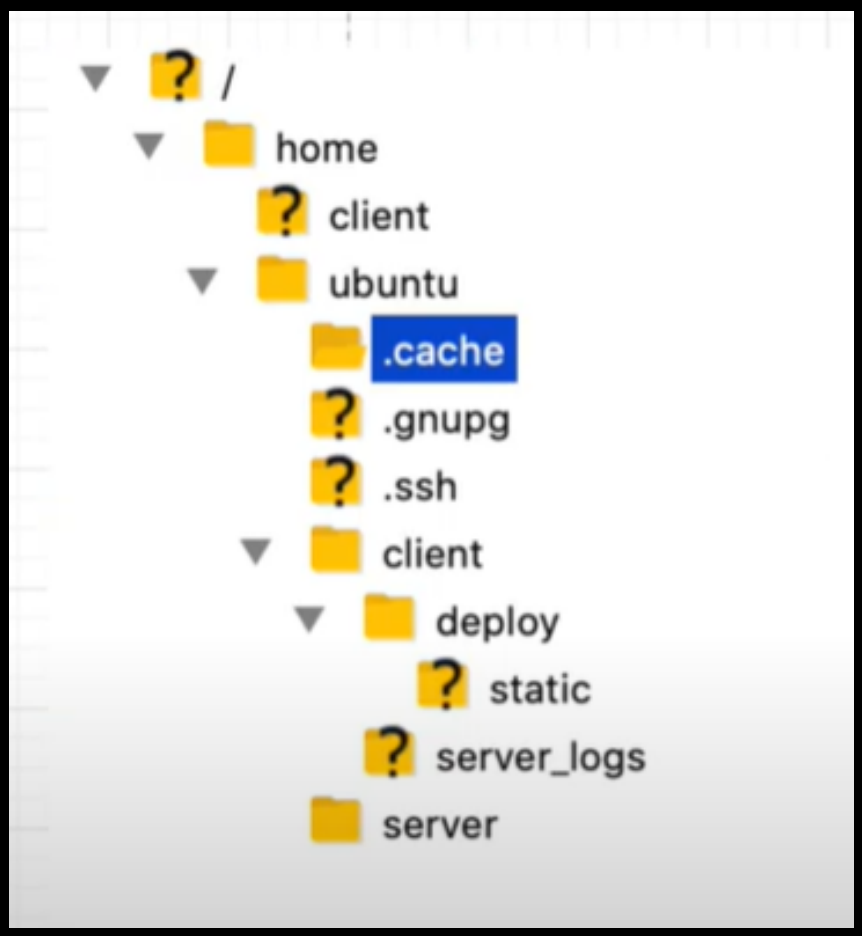
- Again repeat the first command every time you want to start the server to access it from your device.
- Run the command to update all the existing system -- ""sudo apt-get update""

- Now install "file Zilla" in order to access the folder structure of your virtual server.
- Follow the following steps to setup your virtual server in it :-
  - Create a new host in file zilla (click top left side site manager icon).
  - Fill out the following details of your server and connect.



singh-yuvraj002

- Create the following folder structure in the virtual machine using file ZIlla once you access your server on it . Note that some of the folders would be present already and some need to be created.



- Here the client/deploy folder will contain the production build files of the frontend and the server folder will contain the server files.

3. SETTING THE CONFIGURATION FOR NGINX....

- In order to use nginx as a frontend server and node js as your backend server you need to set some configuration settings.
- Run the following commands / steps on the terminal:-
  - sudo apt-get install nodejs.
  - sudo apt-get install nginx.
  - sudo service nginx start (To start the frontend server)
  - cd /etc/nginx
  - sudo vi nginx.conf (To chnage the config. of nginx server)
  - press "d" till every pre written line is deleted

- **Write the following configuration in it.**

```
user ubuntu;
worker_processes  1;

error_log  /var/log/nginx/error.log warn;
pid        /var/run/nginx.pid;

events {
    worker_connections  1024;
}

http {
    include       /etc/nginx/mime.types;
    default_type  application/octet-stream;

    log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"';

    access_log  /var/log/nginx/access.log  main;

    sendfile        on;
    #tcp_nopush     on;

    client_body_buffer_size 100k;
    client_header_buffer_size 1k;
    client_max_body_size 100k;
    large_client_header_buffers 2 1k;
    client_body_timeout 10;
    client_header_timeout 10;
    keepalive_timeout 5 5;
    send_timeout 10;
    server_tokens off;
    #gzip  on; on;

    include /etc/nginx/conf.d/*.conf;
}
```

- **Press "esc" + ":wq" + "enter"**
- **cd conf.d/**
- **sudo touch default.conf**

- sudo vi default.conf (open default.conf)
- Write the following configuration in it.

```
server {
    #listen       80;
    listen 80 default_server;
    listen [::]:80 default_server;
    server_name  yourdomain.com;

    access_log /home/ubuntu/client/server_logs/host.access.log main;

    location / {
        root   /home/ubuntu/client/deploy;
        index  index.html index.htm;
        try_files $uri /index.html;
        add_header X-Frame-Options SAMEORIGIN;
        add_header X-Content-Type-Options nosniff;
        add_header X-XSS-Protection "1; mode=block";
        add_header Strict-Transport-Security "max-age=31536000; includeSubdomains;";
    }

    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
        root   /usr/share/nginx/html;
    }

    server_tokens off;

    location ~ /\.ht {
        deny  all;
    }

}
```

- Replace the word yourdomain with the domain if you have it.
- Press "esc" + ":wq" + "enter"
- sudo service nginx restart (restart the frontend with new changes)

4. CONNECTING SERVER TO DATABASE(MONGODB)....

- Database Connection could be done by using two ways :-
  a. By installing mongo on the virtual server and running the database continuously on the server at port 27017.
  b. By using MongoDB cluster which is nothing but a database running on a virtual server & mostly for free.
- Use the database connection string accordingly in the backend code.

5. UPLOADING FILES AND FINALLY RUNNING THE SERVER....

- As told earlier , drag and drop the contents of the production build folder into the deploy folder of the folder structure using file Zilla
- Then for applying the new frontend features run the command - - "sudo service nginx restart"
- Drag and drop the server(backend) files (except the node modules folder) into the server folder of the virtual server using file Zilla.
- Run command - "cd /home/ubuntu/server/.. (path to backend)
- Run - "npm install" (to install all the packages)
- Run - "node index.js" (to start the backend server) also you can install forever module to run the backend server continuously.
- Finally the server has started on the IP address of the server .

FOLLOW FOR MORE SUCH CONTENTS RELATED TO WEB DEV AND PROGRAMMING

singh-yuvraj002