

# Brainstorm Technical Assignment

First create an EC2 instance with the following Security Groups we will be modifying the inbound rules once we are done with the setup.

The image shows two screenshots from the AWS Management Console. The top screenshot displays the 'Instances' page, where an EC2 instance named 'Brainstorm' (ID: i-01c0ffa0ca3af5496) is shown in a 'Running' state. Below the instance list, the 'Inbound rules' section for the associated security group is visible, showing four rules with various protocols and port ranges. The bottom screenshot shows the 'Edit inbound rules' configuration page for the security group 'sg-072af835783a611c4 - test1-sql'. It displays a table of inbound rules with columns for Security group rule ID, Type, Protocol, Port range, Source, and Description. The rules are configured for MySQL/Aurora, HTTPS, SSH, and HTTP. A warning message at the bottom states: 'Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.'

**Instances (1/1) Info**

Find instance by attribute or tag (case-sensitive) All states

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic
Brainstorm	i-01c0ffa0ca3af5496	Running	t2.micro	Initializing	View alarms +	us-east-1c	ec2-54-152-204-113.co...	54.152.204.113	-

**i-01c0ffa0ca3af5496 (Brainstorm)**

**Inbound rules**

Name	Security group rule ID	Port range	Protocol	Source	Security groups	Description
-	sgr-023a0600fa4c7c63	22	TCP	115.98.96.89/32	test1-sql	-
-	sgr-0c8686f68f9947cb	443	TCP	0.0.0.0/0	test1-sql	-
-	sgr-02abca34408a2976b	80	TCP	0.0.0.0/0	test1-sql	-
-	sgr-0da35139f6620c3e6	3306	TCP	0.0.0.0/0	test1-sql	-

**Edit inbound rules Info**

Inbound rules control the incoming traffic that's allowed to reach the instance.

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-0da35139f6620c3e6	MySQL/Aurora	TCP	3306	Cu... 49.43.228.174/32	Delete
sgr-0c8686f68f9947cb	HTTPS	TCP	443	Cu... 0.0.0.0/0	Delete
sgr-023a0600fa4c7c63	SSH	TCP	22	Cu... 49.43.228.174/32	Delete
sgr-02abca34408a2976b	HTTP	TCP	80	Cu... 0.0.0.0/0	Delete

Add rule

Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Cancel Preview changes Save rules

# Brainstorm Technical Assignment

## Ssh into the EC2 instance

```
root@rahul-G3-3500:/home/rahul/Downloads# ssh -l ubuntu.pem ubuntu@54.152.204.113
The authenticity of host '54.152.204.113 (54.152.204.113)' can't be established.
ED25519 key fingerprint is SHA256:QC/+xLrnyNs6EQ/xt2Bz/F3w0/WHFUDh0JFb8Bg.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '54.152.204.113' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.8.0-1015-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Mon Dec  9 16:17:58 UTC 2024

System load:  0.0          Processes:      109
Usage of /:   21.1% of 7.57GB Users logged in:  0
Memory usage: 21%         IPv4 address for eth0: 172.31.27.18
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-27-18:~$ sudo su
```

## Then install nginx on our Ec2 instance and make sure is running

```
Setting up nginx-common (1.18.0-6ubuntu14.5) ...
Created symlink /etc/systemd/system/multi-user.target.wants/nginx.service → /lib/systemd/system/nginx.service.
Setting up libjbig0:amd64 (2.1-3.1ubuntu0.22.04.1) ...
Setting up libnghttp-mod-http-xslt-filter (1.18.0-6ubuntu14.5) ...
Setting up fonts-dejavu-core (2.37-2build1) ...
Setting up libjpeg-turbo0:amd64 (2.1.2-0ubuntu1) ...
Setting up libwebp7:amd64 (1.2.2-2ubuntu0.22.04.2) ...
Setting up libnghttp-mod-http-geoip2 (1.18.0-6ubuntu14.5) ...
Setting up libjpeg8:amd64 (8c-2ubuntu10) ...
Setting up libnghttp-mod-nccl (1.18.0-6ubuntu14.5) ...
Setting up fontconfig-config (2.13.1-4.2ubuntu5) ...
Setting up libnghttp-mod-stream (1.18.0-6ubuntu14.5) ...
Setting up libtiff5:amd64 (4.3.0-6ubuntu10) ...
Setting up libfontconfig1:amd64 (2.13.1-4.2ubuntu5) ...
Setting up libnghttp-mod-stream-geoip2 (1.18.0-6ubuntu14.5) ...
Setting up libgd3:amd64 (2.3.0-2ubuntu2.3) ...
Setting up libnghttp-mod-http-image-filter (1.18.0-6ubuntu14.5) ...
Setting up nginx-core (1.18.0-6ubuntu14.5) ...
 * Upgrading binary nginx
Setting up nginx (1.18.0-6ubuntu14.5) ...
Processing triggers for ufw (0.36.1-4ubuntu0.1) ...
Processing triggers for man-db (2.10-2.1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.8) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

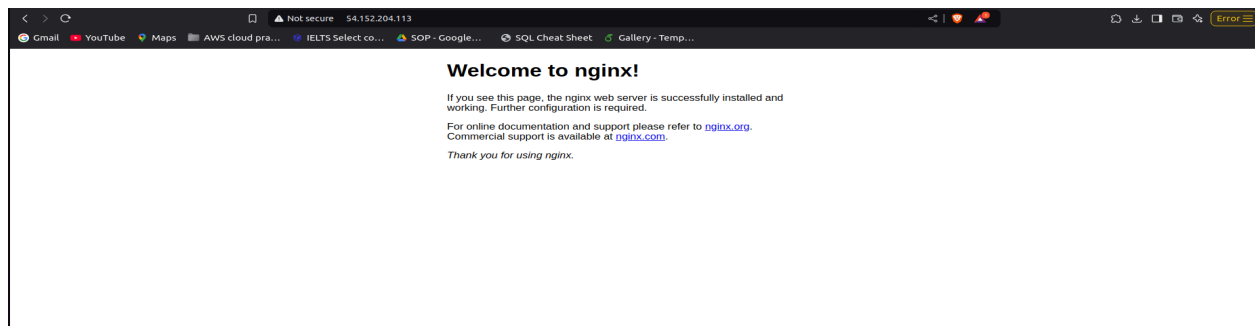
No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-27-18:/home/ubuntu# systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2024-12-09 16:23:02 UTC; 47s ago
     Docs: man:nginx(8)
   Process: 2028 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
   Process: 2029 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
   Main PID: 2129 (nginx)
     Tasks: 2 (limit: 1130)
    Memory: 4.6M
       CPU: 19ms
   CGroup: /system.slice/nginx.service
           └─2129 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             └─2132 "nginx: worker process"

Dec 09 16:23:02 ip-172-31-27-18 systemd[1]: Starting A high performance web server and a reverse proxy server...
Dec 09 16:23:02 ip-172-31-27-18 systemd[1]: Started A high performance web server and a reverse proxy server.
root@ip-172-31-27-18:/home/ubuntu#
```

[ OK ]



# Brainstorm Technical Assignment

Next we will install mysql and assign one password to it as shown

```
root@ip-172-31-27-18:/home/ubuntu# mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 8.0.40-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH caching_sha2_password BY 'rahul';
Query OK, 0 rows affected (0.07 sec)

mysql> ^DBye
root@ip-172-31-27-18:/home/ubuntu#

root@ip-172-31-27-18:/home/ubuntu# mysqlq
Command 'mysqlq' not found, did you mean:
  command 'mysql' from deb mysql-client-core-8.0 (8.0.39-0ubuntu0.22.04.1)
  command 'mysql' from deb mariadb-client-core-10.6 (1:10.6.18-0ubuntu0.22.04.1)
Try: apt install <deb name>
root@ip-172-31-27-18:/home/ubuntu# mysql
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: NO)
root@ip-172-31-27-18:/home/ubuntu#
```

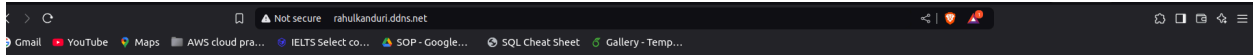
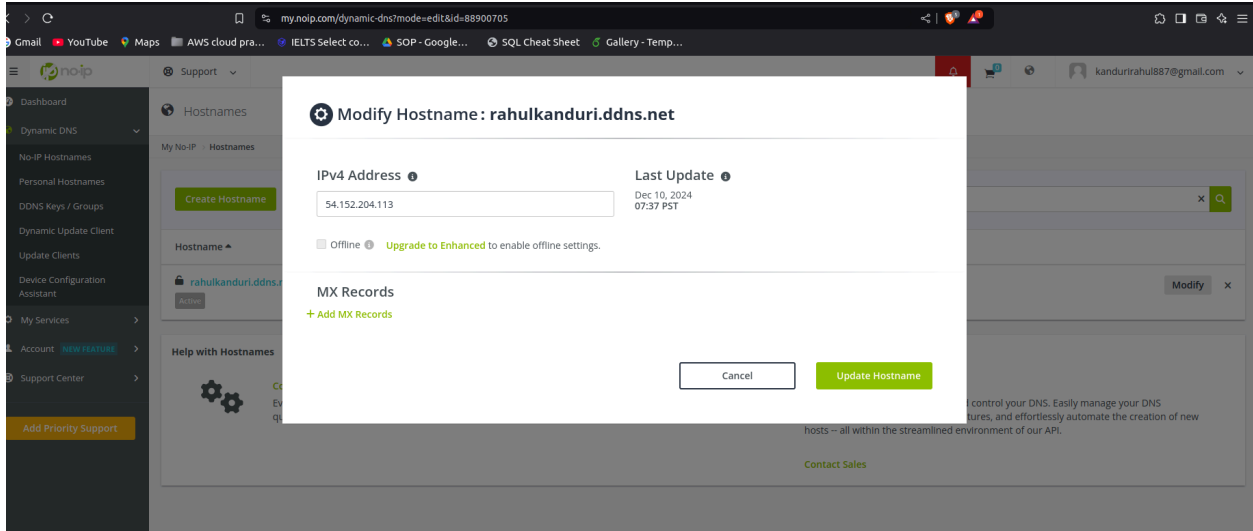
Next install PHP remember each WordPress plugin has its own set of requirements. Some may require additional PHP extension packages to be installed.

**sudo apt install php-curl php-gd php-intl php-mbstring php-soap php-xml php-xmlrpc php-zip**

Next we need to create a domain for our application, make sure that you select 'A' name in the type since we are directly routing traffic from here and in IPV4 address give your public IP.

If everything goes right when you copy and paste the domain name we need to see nginx welcome page that we got when we opened it with instance public IP as shown

# Brainstorm Technical Assignment



Now we will need to change the nginx conf as follows, and lets understand the code

```
root@ip-172-31-27-18:/home/ubuntu# cat /etc/nginx/sites-available/wordpress
server {
    listen 80;
    server_name rahulkanduri.ddns.net;

    root /var/www/wordpress;
    index index.php index.html index.htm;

    location / {
        try_files $uri $uri/ /index.php?$args;
    }

    location ~ \.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/var/run/php/php7.4-fpm.sock;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        include fastcgi_params;
    }

    location ~ /\.ht {
        deny all;
    }
}

root@ip-172-31-27-18:/home/ubuntu# nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
root@ip-172-31-27-18:/home/ubuntu#
```

=====

# ***Brainstorm Technical Assignment***

```
server {  
    listen 80;  
    server_name rahulkanduri.ddns.net;  
  
    root /var/www/wordpress;  
    index index.php index.html index.htm;  
  
    location / {  
        try_files $uri $uri/ /index.php?$args;  
    }  
  
    location ~ \.php$ {  
        include snippets/fastcgi-php.conf;  
        fastcgi_pass unix:/var/run/php/php7.4-fpm.sock;  
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;  
        include fastcgi_params;  
    }  
  
    location ~ /\.ht {  
        deny all;  
    }  
}
```

In server block we have given the details like nginx port which is 80 and in servername we have given our domain name so that nginx can send the traffic to localhost according to the config we are defining

Next in root we are defining the working directory where our entire code and files will be there

Then in index we have given the files that it has to try by order inside the working directory

Next we are using location block for root which means all request coming to nginx url

Try\_files this is a tricky part so nginx has some pretty URLs that let's make it simple with an example:

## ***Brainstorm Technical Assignment***

\$uri : if I do <http://rahulkanduri.ddns.net/somepage> nginx will search for somepage.html and serves it.

\$uri/ this means if we do not have somepage.html it will search for somepage folder inside root folder of nginx and serves the .html file inside of it

If none of the got satisfied it will show index.php

Next we have another location block like “location ~ \.php\$ ”

First thing nginx cannot serve .php files, let's say in our root folder I have somepage.html then we are hitting this file then we can see contents in it but if it is somepage.php nginx cannot serve that

So we are including fastcgi-php.conf which have some common configs to handle .php code

Next we have fastcgi\_pass which is **Unix socket** where PHP-FPM (FastCGI Process Manager) is running this will help to process php code

Now we need to install wordpress we can do it with following commands

```
14 nano /etc/nginx/sites-available/rahulkanduri.ddns.net
15 sudo wget https://wordpress.org/latest.tar.gz
16 tar -xvzf latest.tar.gz
17 mv wordpress /var/www/wordpress
18 history
root@ip-172-31-27-18: /home/ubuntu#
```

Next we need an database so that word press can store content and we will create an user as well

# Brainstorm Technical Assignment

```
root@ip-172-31-27-18:/home/ubuntu# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 16
Server version: 8.0.40-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database wordpress;
Query OK, 1 row affected (0.02 sec)

mysql> CREATE USER 'rahul'@'localhost' IDENTIFIED BY 'rahul';
Query OK, 0 rows affected (0.03 sec)

mysql>
mysql> GRANT ALL PRIVILEGES ON wordpress.* TO 'rahul'@'localhost';
Query OK, 0 rows affected (0.01 sec)

mysql> GRANT ALL PRIVILEGES ON *.* TO 'rahul'@'localhost';
Query OK, 0 rows affected (0.01 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.01 sec)

mysql>
```

Update this values in config so that wordpress can authenticate the credentials

```
*
* * Database settings
* * Secret keys
* * Database table prefix
* * ABSPATH
*
* @link https://developer.wordpress.org/advanced-administration/wordpress/wp-config/
*
* @package WordPress
*/

/** Database settings - You can get this info from your web host ** */
/** The name of the database for WordPress */
define( 'DB_NAME', 'wordpress' );

/** Database username */
define( 'DB_USER', 'rahul' );

/** Database password */
define( 'DB_PASSWORD', 'rahul' );

/** Database hostname */
define( 'DB_HOST', 'localhost' );

/** Database charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );

/** The database collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );

/**#@+
 * Authentication unique keys and salts.
```

Next please do <http://rahulkanduri.ddns.net/wordpress>

If you get 502 bad gateway make sure that php and php-fpm version is same  
remember php-fpm will process the code for you

# Brainstorm Technical Assignment

The first screenshot shows the WordPress installation welcome screen. It lists the following items that need to be known before getting started:

1. Database name
2. Database username
3. Database password
4. Database host
5. Table prefix (if you want to run more than one WordPress in a single database)

Below this list, there is a note: "This information is being used to create a wp-config.php file. If for any reason this automatic file creation does not work, do not worry. All this does is fill in the database information to a configuration file. You may also simply open wp-config-sample.php in a text editor, fill in your information, and save it as wp-config.php. Read more help here: [WordPress database configuration](#)." A "Let's go!" button is at the bottom.

The second screenshot shows the "Step 1: Database" configuration screen. It contains the following fields and instructions:

- Database Name:** . The name of the database you want to use with WordPress.
- Username:** . Your database username.
- Password:** . Your database password. (There is a "Hide" button next to the password field.)
- Database Host:** . You should be able to get this info from your web host. If localhost does not work.
- Table Prefix:** . If you want to run multiple WordPress installations in a single database, change this.

A "Submit" button is at the bottom of the form.

SSL:

First install certbot for getting the ssl certificates

***sudo apt install certbot python3-certbot-nginx -y***

Now we need to get ssl certificates right so we run the following command

***sudo certbot --nginx -d rahulkanduri.ddns.net***

Then it will ask some questions like email address etc



# Brainstorm Technical Assignment

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-27-18:/home/ubuntu# sudo certbot --nginx -d rahulkanduri.ddns.net
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Enter email address (used for urgent renewal and security notices)
(Enter 'c' to cancel): kandurirahul887@gmail.com

- - - - -
Please read the Terms of Service at
https://letsencrypt.org/documents/LE-SA-v1.4-April-3-2024.pdf. You must agree in
order to register with the ACME server. Do you agree?
- - - - -
(Y)es/(N)o: y

- - - - -
Would you be willing, once your first certificate is successfully issued, to
share your email address with the Electronic Frontier Foundation, a founding
partner of the Let's Encrypt project and the non-profit organization that
develops Certbot? We'd like to send you email about our work encrypting the web,
EFF news, campaigns, and ways to support digital freedom.
- - - - -
(Y)es/(N)o: n
Account registered.
Requesting a certificate for rahulkanduri.ddns.net

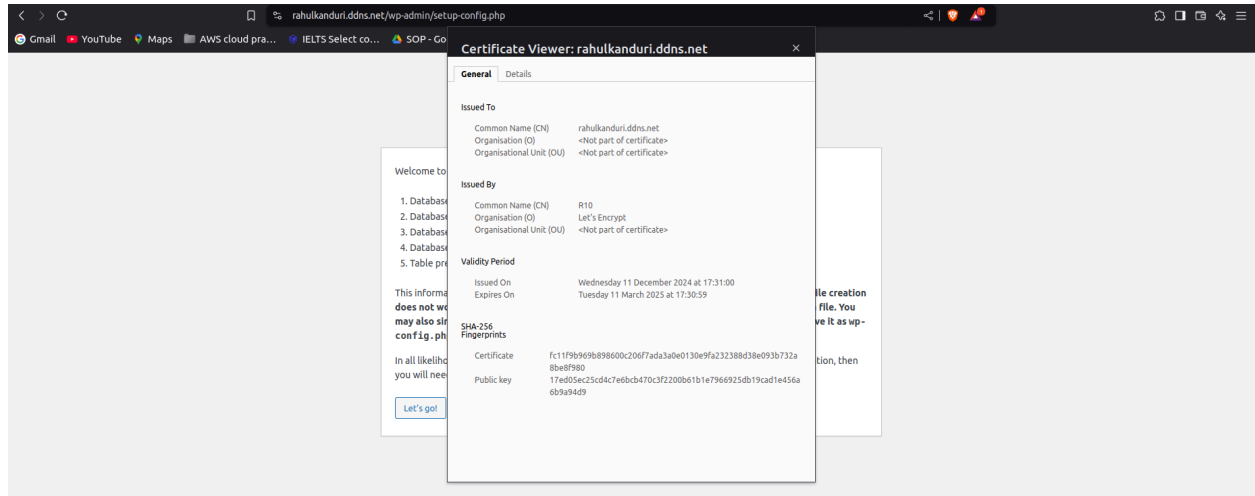
Successfully received certificate.
Certificate is saved at: /etc/letsencrypt/live/rahulkanduri.ddns.net/fullchain.pem
Key is saved at: /etc/letsencrypt/live/rahulkanduri.ddns.net/privkey.pem
This certificate expires on 2025-03-11.
These files will be updated when the certificate renews.
Certbot has set up a scheduled task to automatically renew this certificate in the background.

Deploying certificate
Successfully deployed certificate for rahulkanduri.ddns.net to /etc/nginx/sites-enabled/wordpress
Congratulations! You have successfully enabled HTTPS on https://rahulkanduri.ddns.net

- - - - -
If you like Certbot, please consider supporting our work by:
* Donating to ISRG / Let's Encrypt: https://letsencrypt.org/donate
* Donating to EFF: https://eff.org/donate-le
- - - - -
root@ip-172-31-27-18:/home/ubuntu#
```

Now you open your URL with https, if you get any error remember we have to allow traffic to instance with https 443 as well

# Brainstorm Technical Assignment



But we haven't changed anything in the server block to listen to 443 right? Then how can we open the URL?

The answer is certbot is nginx extension so it will change the nginx.conf (in our case /etc/nginx/sites-available/wordpress)

```
root@ip-172-31-27-18:/home/ubuntu#
root@ip-172-31-27-18:/home/ubuntu# cat /etc/nginx/sites-available/wordpress
server {
    server_name rahulkanduri.ddns.net;

    root /var/www/wordpress;
    index index.php index.html index.htm;

    location / {
        try_files $uri $uri/ /index.php?$args;
    }

    location ~ /\.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/var/run/php/php8.1-fpm.sock;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        include fastcgi_params;
    }

    location ~ /\.ht {
        deny all;
    }

    listen 443 ssl; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/rahulkanduri.ddns.net/fullchain.pem; # managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/rahulkanduri.ddns.net/privkey.pem; # managed by Certbot
    include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot
}

server {
    if ($host = rahulkanduri.ddns.net) {
        return 301 https://$host$request_uri;
    } # managed by Certbot

    listen 80;
    server_name rahulkanduri.ddns.net;
    return 404; # managed by Certbot
}
```

Optimizing Nginx server configuration:

# Brainstorm Technical Assignment

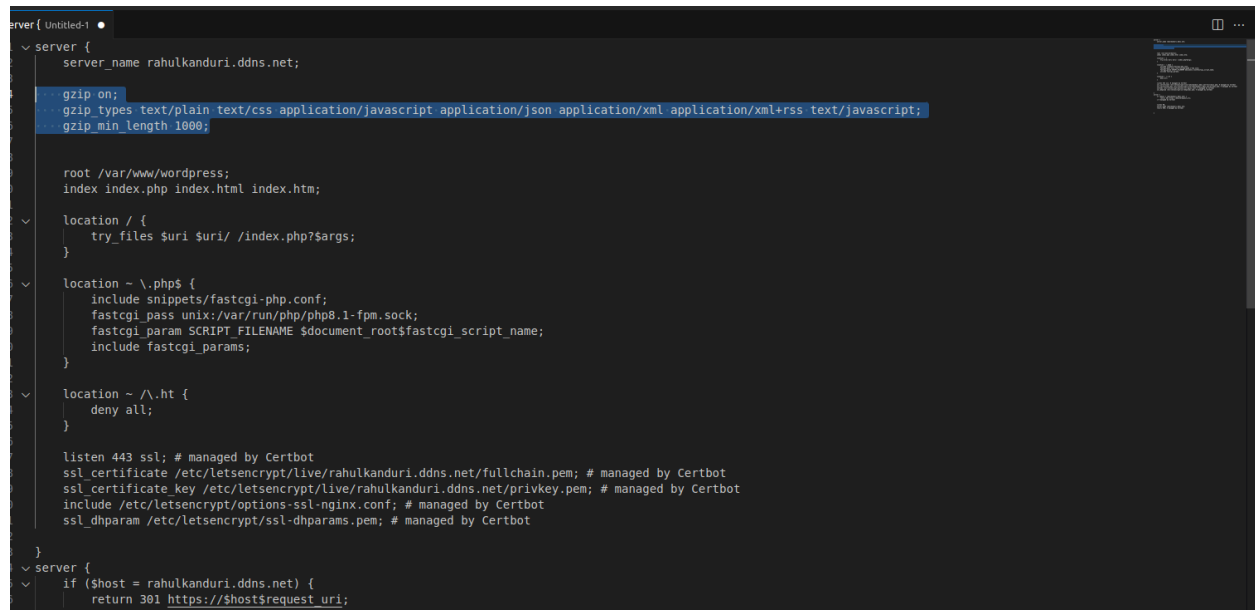
We are going to enable gzip compression, but what is the use of the gzip. Everytime when you do <https://rahulkanduri.ddns.net/somepage.html> remember you will be executing some .html or .php in simple terms we will be transferring from local machine to browser what is we can compress the file so that the transferring of the file to browser will take less time then we will use gzip

Add the following lines for enabling gzip

gzip on; ⇒ This is to enable the gzip

gzip\_types text/plain text/css application/javascript application/json application/xml application/xml+rss text/javascript; ⇒ This files types are going to be compressed

gzip\_min\_length 1000; ⇒ To compress this files it has to be this much size in bytes



```
server {
    server_name rahulkanduri.ddns.net;

    gzip on;
    gzip_types text/plain text/css application/javascript application/json application/xml application/xml+rss text/javascript;
    gzip_min_length 1000;

    root /var/www/wordpress;
    index index.php index.html index.htm;

    location / {
        try_files $uri $uri/ /index.php?$args;
    }

    location ~ \.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/var/run/php/php8.1-fpm.sock;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        include fastcgi_params;
    }

    location ~ /\.ht {
        deny all;
    }

    listen 443 ssl; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/rahulkanduri.ddns.net/fullchain.pem; # managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/rahulkanduri.ddns.net/privkey.pem; # managed by Certbot
    include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot
}

server {
    if ($host = rahulkanduri.ddns.net) {
        return 301 https://$host$request_uri;
    } # managed by Certbot
```

Caching: Here we will tell nginx to send and store some of the files on user's browser so that next time the user hits the files with URL instead of coming to server it can show the content from stored files from browser

Add the following lines in conf file

```
location ~*
\.(\.jpg|jpeg|png|gif|svg|ico|css|js|pdf|txt|woff|woff2|ttf|eot|otf|webp|avif)$ {
    expires 30d;
    add_header Cache-Control "no-transform";
```

# Brainstorm Technical Assignment

}

```
server {
    server_name rahulkanduri.ddns.net;

    gzip on;
    gzip_types text/plain text/css application/javascript application/json application/xml application/xml+rss text/javascript;
    gzip_min_length 1000;

    root /var/www/wordpress;
    index index.php index.html index.htm;

    location / {
        try_files $uri $uri/ /index.php?$args;
    }

    location ~* \.(jpg|jpeg|png|gif|svg|ico|css|js|pdf|txt|woff|woff2|ttf|eot|otf|webp|avif)$ {
        expires 30d;
        add_header Cache-Control "no-transform";
    }

    location ~ \.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/var/run/php/php8.1-fpm.sock;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        include fastcgi_params;
    }

    location ~ /\.ht {
        deny all;
    }

    listen 443 ssl; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/rahulkanduri.ddns.net/fullchain.pem; # managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/rahulkanduri.ddns.net/privkey.pem; # managed by Certbot
    include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot
}
```

**location ~\***

**\.(jpg|jpeg|png|gif|svg|ico|css|js|pdf|txt|woff|woff2|ttf|eot|otf|webp|avif)\$**

In this line we are telling that any file with this extension has to be copied to users browser so that he can access that file with less latency

**expires 30d;**

This we are telling the users browser to check for the changes in the stored files only after 30 days

**add\_header Cache-Control "no-transform";**

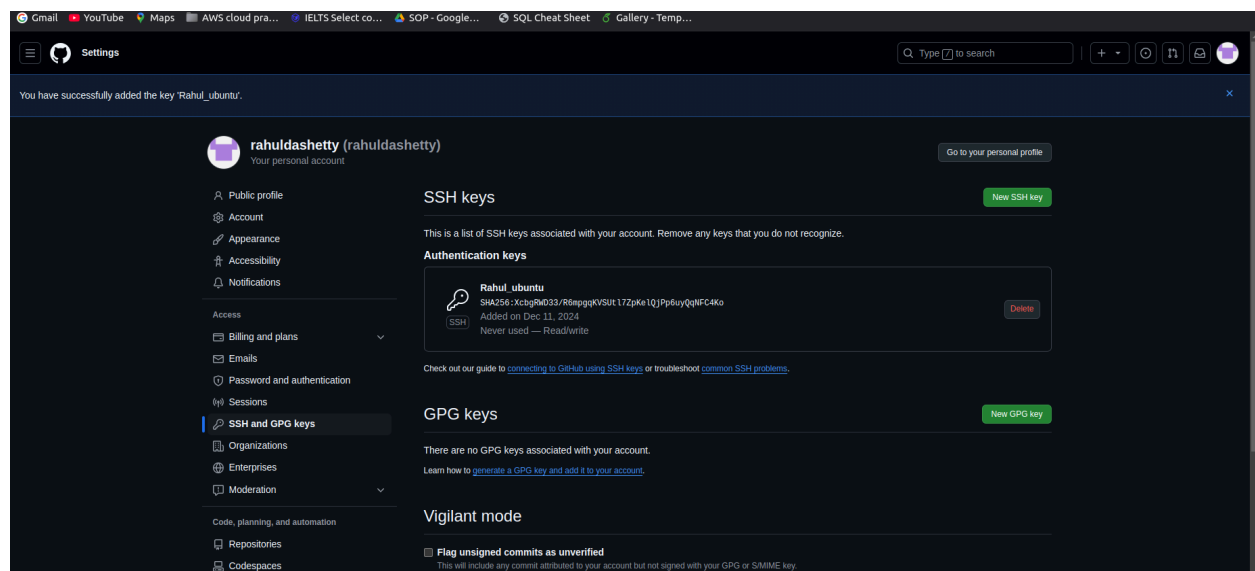
We are telling the browser to cache the data not any intermediate cdn's only browser and without modification

# Brainstorm Technical Assignment

## GitHub & CI/CD:

Now in your local system we need to initialize git, we have generated the ssh key and saved the key in the github's SSH KEYS as shown, so that we are authenticated by the git hub

```
root@rahul-G3-3500:/home/rahul/Desktop/Brainstorm# ssh-keygen -t rsa -b 4096 -C "kandurirahul887@gmail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:XcbgB033/R6mpgqKV5utL7ZpK61QJPP6uyQqNFC4ko kandurirahul887@gmail.com
The key's randomart image is:
+--[RSA 4096]-----+
|      .+..+      |
|      +O  +..+   |
|      .  O.O.=+ . |
|      .  B.+..+ oo|
|      .  =S+..+  +|
|      .  .+ .+ O  |
|      .  O.O. .O  |
|      .  oo O .O +|
|      .  O+ .+..O O|
|      .  O+ .+..O O|
+-----[SHA256]-----+
root@rahul-G3-3500:/home/rahul/Desktop/Brainstorm# cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCAkA7gQ4JULZ8KGU5jJg4AodvBFfT2bv6Ub26KpYqQ8mNAXd8vgSMOTh3FpaY5L87GG/2hk071bBM8uGBoF4/W7ytpqINvULjGtWj3OP4GR5RA212oufM0iabUdBokAtVIRBA6KGz1+49AzILdCMCaCXvsAtyVf0oV3
ekosse48vCQ2IGL2x1PAdeBee8RW/Kj4dCQ01SXj1+dd5jCZQ8NLA54fe+19MFk4x9q1f+s41WdpHn+2003V5J1Dm8p5DwbT7VByE89500ukVXkEocaRwKufAwXoxEjyeAqCZXFvWx1rKSH4Qw1H1LUNOK14N8dk1A9HsV9QrdUyu2CYFCN9dwyF3E2qJLNdvoPT2ATrW
s8D0McHw7C70k+duj0Z2fGd9CvFuxh8vayWZnbpISN1JLD/580whhd/PHTR75F3NSLeohV0tBUNg4/8RMGzVALLN4Pox/b66f7s8fJBL9+tdA0rduh2Hw24eFL52BxqL5L864g0Xh6INGv3BrRHL5RC8ErnrgrH0wrXN2rZy2N9LjhcP3zbTv0Fk3jY42ZLn
rjYyzx00Dgax390pAnv71ZdXP0gNrxpX84kRwJ3N2hL17006j6F6tc9ZmesdkSh1v0oZa4N3yKwXuoq5/buN0HfXaAc00khH1Z603E1aG== kandurirahul887@gmail.com
root@rahul-G3-3500:/home/rahul/Desktop/Brainstorm# git remote set-url origin git@github.com:rahuldashetty/Brainstorm.git
root@rahul-G3-3500:/home/rahul/Desktop/Brainstorm#
root@rahul-G3-3500:/home/rahul/Desktop/Brainstorm# git remote -v
origin  git@github.com:rahuldashetty/Brainstorm.git (fetch)
origin  git@github.com:rahuldashetty/Brainstorm.git (push)
root@rahul-G3-3500:/home/rahul/Desktop/Brainstorm#
root@rahul-G3-3500:/home/rahul/Desktop/Brainstorm# ssh -T git@github.com
Hi rahuldashetty! You've successfully authenticated, but GitHub does not provide shell access.
root@rahul-G3-3500:/home/rahul/Desktop/Brainstorm# git branch
* main
root@rahul-G3-3500:/home/rahul/Desktop/Brainstorm# git push origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 228 bytes | 228.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:rahuldashetty/Brainstorm.git
 * [new branch] main -> main
root@rahul-G3-3500:/home/rahul/Desktop/Brainstorm#
```



# Brainstorm Technical Assignment

Now we need to copy key from our local machine to ec2 instance so first we generate the key then we will copy the content into ec2

```
ssh-keygen -t rsa -b 4096 -C "kandurirahul887@gmail.com" -f  
~/.ssh/github-action-ec2  
cat ~/.ssh/github-action-ec2
```

```
root@rahul-G3-3500:/home/rahul/Desktop/Brainstorm# ssh-keygen -t rsa -b 4096 -C "kandurirahul887@gmail.com" -f ~/.ssh/github-action-ec2
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/github-action-ec2
Your public key has been saved in /root/.ssh/github-action-ec2.pub
The key fingerprint is:
SHA256:GURStFwIsK6shbDny837xDb09SKlTeSyfv6VOVTI kandurirahul887@gmail.com
The key's randomart image is:
+--[RSA 4096]--+
|.+++=+0|
|o+oo+|
|.+++= E |
|.+++= o.|
|So.= . .|
|..+ +|
|..+ = .|
|. * oo.o|
|. ooo=.|
+-----[SHA256]-----+
root@rahul-G3-3500:/home/rahul/Desktop/Brainstorm#
root@rahul-G3-3500:/home/rahul/Desktop/Brainstorm#
root@rahul-G3-3500:/home/rahul/Desktop/Brainstorm# cat ~/.ssh/github-action-ec2
-----BEGIN OPENSSH PRIVATE KEY-----
oB1bnNzck1rZktdjEAAAABGSwbmUAAAABm9uZQAAAAAAAAAABAAACFwAAADzc2gtcn
NHAAAAAwEAAQAAgEAWd3dxt4N0e6ZM4ffo6FFE2nL2P2OE7nhS0Ovqc06tuyQ+JsoE5W
d13uvSLk6j5InWgS1889UOYZ3E0SxhugnXP0EV/v+kBuPBAargunBvducaulgM1LJLZGT
EuqdtUHOAJrRFPwOhLBCKbdgFDRxR/gvWoyoxnatuzGkW3gJb9Z1U2ob1L+rJ5Juq6NnQR
j098n1o8oq6QtsuyZF98NMUGl4+jFFKLhLX3g/AEDMq1JEjrNwLdJLdGa03jKcpaZ4Se
cR101m6Mx1gY8o16n3E4u3ZKcPncSNLxUR/UM71k114qMFTcuJ8YQm2R2CHLjYsNT7q/p
Z1Wsa/H9j5b6quGGL74b0tpVDRRD9dG5BPXZR+L+8g4mcsA0Hhns0R495NV87YvVAn2bw
lN6SG1VRd0d+GutLUL1PSh/MoRPRFJ04DuchFZf1FyEh1+goukXU13hsF8xyFFQPaLjn
fcd+0MOuBPZKwct91+Vh51DF1JY1ietYMoAKCvWVAKJnHg67p2g1KearlyRT30xb
7LzqLaxFIWELfWcctMDDBtwEwPMG+Qa5TEAwBTxHE4U0wnRQIRCFc0phxvUHQX/r60VY
K5o1Cb50b4un3Zz+A9BcQQfAb2g+kxNhrv+NtboofYAD0JCVCUNyBk+pgnthoQ8zVg0eEK
CAAAQFUqL6xVXousAAAAHC3NGLXJZYQAAgEAWd3dxt4N0e6ZM4ffo6FFE2nL2P2OE7n
hS0Ovqc06tuyQ+JsoE5Wd13uvSLk6j5InWgS1889UOYZ3E0SxhugnXP0EV/v+kBuPBAar
gunBvducaulgM1LJLZGTUeQdtUHOAJrRFPwOhLBCKbdgFDRxR/gvWoyoxnatuzGkW3gJb9
Z1U2ob1L+rJ5Juq6NnQRj098n1o8oq6QtsuyZF98NMUGl4+jFFKLhLX3g/AEDMq1JEjrN
wLdJLdGa03jKcpaZ4SecR101m6Mx1gY8o16n3E4u3ZKcPncSNLxUR/UM71k114qMFTcu
J8YQm2R2CHLjYsNT7q/pZ1Wsa/H9j5b6quGGL74b0tpVDRRD9dG5BPXZR+L+8g4mcsA0Hh
ns0R495NV87YvVAn2bw1N6SG1VRd0d+GutLUL1PSh/MoRPRFJ04DuchFZf1FyEh1+gou
kXU13hsF8xyFFQPaLjnLfcd+0MOuBPZKwct91+Vh51DF1JY1ietYMoAKCvWVAKJnHg6
7p2g1KearlyRT30xb7LzqLaxFIWELfWcctMDDBtwEwPMG+Qa5TEAwBTxHE4U0wnRQIRCF
c0phxvUHQX/r60VYK5o1Cb50b4un3Zz+A9BcQQfAb2g+kxNhrv+NtboofYAD0JCVCUNyBk
+pgnthoQ8zVg0eEKCAAAQABAAAAQABAAACABhplS3omqew0ozptmshvkSW08rUYTgaPN
jQNXtF5dGcTecNHNhJGne+TywPsSnt3WK/XR7pxT4C2jztwbT4do1zXKCRZLUf/gweD
/0b1nc9VqfLF+cKJhA+90AnnbnA1k5und1JqAvgn1BxKY4gpzJ8KX6rNfLmvP92Mp1kx
0ncYcct126zrFVJjwppqFU107b1VXB1gkFI0u1RezxKAXzJFF7MwZ6a1nvZuH2p01r1
+1DLfztHqGvVny7cdv1KQ4Fz+o/rScpS1w6w8115nr80zjpfJ55F81NZ0y1W3Y2
zg6Thx2nQ1VEzcQIQ9PCK1LT8+Ha2ZnNvPRUqS41/y2rprb1ru6dR3X0pWxQ5dEz4Dcm
```

Then in ec2 instance paste the contents into .ssh folder as shown

[illegible]

Then we add required repository secrets as shown

General

Access

Collaborators

Moderation options

Code and automation

Branches

Tags

Rules

Actions

Webhooks

Environments

Codespaces

Pages

Security

Code security

Deploy keys

Secrets and variables

Actions

Codespaces

Dependabot

Integrations

GitHub Apps

Email notifications

Actions secrets and variables

SecretsVariables

Environment secrets

This environment has no secrets.

Manage environment secrets

New repository secret

Name ↴	Last updated		
DEPLOY_PATH	now		
EC2_IP	2 minutes ago		
EC2_PRIVATE_KEY	now		
EC2_USER	2 minutes ago		

© 2024 GitHub, Inc.

Terms

Privacy

Security

Status

Docs

Contact

Manage cookies

Do not share my personal information

We have added an simple .php file and we also created deploy.yml file inside the github/workflows directory so that if any push see the actions being done

# Brainstorm Technical Assignment

```
root@rahul-G3-3500:/home/rahul/Desktop/Brainstorm# ls
README.md
root@rahul-G3-3500:/home/rahul/Desktop/Brainstorm# nano mycode.php
root@rahul-G3-3500:/home/rahul/Desktop/Brainstorm# nano mycode.php
root@rahul-G3-3500:/home/rahul/Desktop/Brainstorm# git add .
root@rahul-G3-3500:/home/rahul/Desktop/Brainstorm# nano .github/workflows/deploy.yml
root@rahul-G3-3500:/home/rahul/Desktop/Brainstorm# git add .
root@rahul-G3-3500:/home/rahul/Desktop/Brainstorm# git commit -m "test workflow cicd"
[main a5702ca] test workflow cicd
2 files changed, 44 insertions(+)
create mode 100644 .github/workflows/deploy.yml
create mode 100644 mycode.php
root@rahul-G3-3500:/home/rahul/Desktop/Brainstorm# git push origin main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (6/6), 873 bytes | 873.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:rahuldashetty/brainstorm.git
 3644179..a5702ca  main -> main
root@rahul-G3-3500:/home/rahul/Desktop/Brainstorm#
```

After pushing we can see the actions that is being done

The screenshot displays the GitHub Actions interface for the repository 'rahuldashetty / Brainstorm'. The 'Actions' tab is selected, showing a list of workflows. The workflow 'test workflow cicd' is highlighted, indicating it is in progress. The interface includes a sidebar with navigation options like 'Code', 'Issues', 'Pull requests', 'Actions', 'Projects', 'Wiki', 'Security', 'Insights', and 'Settings'. The main area shows the workflow run details, including the commit hash 'a5702ca' and the status 'In progress'.

GitHub Actions interface showing the workflow run for 'test workflow cicd'.

The workflow run is titled 'test workflow cicd' and is currently in progress. The status is 'In progress' and the commit hash is 'a5702ca'.

The workflow run is triggered by a push to the 'main' branch.

The workflow run is currently in progress, as indicated by the 'In progress' status.



# Brainstorm Technical Assignment

```
root@rahul-G3-3500: /home/rahul/Desktop/Brainstorm# cat .github/workflows/deploy.yml
name: Deploy to EC2

on:
  push:
    branches:
      - main

jobs:
  deploy:
    runs-on: ubuntu-latest # Runs the job on an Ubuntu runner

    steps:
      - name: Checkout repository
        uses: actions/checkout@v3

      - name: Set up SSH key
        run: |
          mkdir -p ~/.ssh
          echo "SEC2_PRIVATE_KEY" > ~/.ssh/id_rsa
          chmod 600 ~/.ssh/id_rsa
          env:
            EC2_PRIVATE_KEY: ${ secrets.EC2_PRIVATE_KEY }

      - name: Add EC2 host to known_hosts
        run: |
          ssh-keyscan -H ${ secrets.EC2_IP } >> ~/.ssh/known_hosts

      - name: SCP mycode.php to EC2
        run: |
          scp -i ~/.ssh/id_rsa ./mycode.php ${ secrets.EC2_USER }@${ secrets.EC2_IP }:/home/ubuntu/
          env:
            EC2_USER: ${ secrets.EC2_USER }
            EC2_IP: ${ secrets.EC2_IP }

      - name: Move mycode.php to /var/www/wordpress
        run: |
          ssh -i ~/.ssh/id_rsa ${ secrets.EC2_USER }@${ secrets.EC2_IP } << 'EOF'
          sudo mv /home/ubuntu/mycode.php /var/www/wordpress/
          sudo chown www-data:www-data /var/www/wordpress/mycode.php
          sudo chmod 644 /var/www/wordpress/mycode.php
          EOF
          env:
            EC2_USER: ${ secrets.EC2_USER }
            EC2_IP: ${ secrets.EC2_IP }

      - name: Restart Nginx (if needed)
        run: |
          ssh -i ~/.ssh/id_rsa ${ secrets.EC2_USER }@${ secrets.EC2_IP } 'sudo systemctl restart nginx'
          env:
            EC2_USER: ${ secrets.EC2_USER }
            EC2_IP: ${ secrets.EC2_IP }
```

This is the deploy.yml file

After really brainstorming with the error we can finally fix the issue

# Brainstorm Technical Assignment

rahuldashetty

Brainstorm

Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

Actions

All workflows

Deploy to EC2

Management

Caches

Attestations

Runners

Usage metrics

Performance metrics

All workflows

Showing runs from all workflows

Help us improve GitHub Actions

Give feedback

7 workflow runs

	Event	Status	Branch	Actor
test workflow cicd attempt-7	Deploy to EC2 #7: Commit d41df30 pushed by rahuldashetty	main	1 minute ago	...
test workflow cicd attempt-6	Deploy to EC2 #6: Commit 8ae612b pushed by rahuldashetty	main	2 minutes ago	...
test workflow cicd attempt-5	Deploy to EC2 #5: Commit 9f664d1 pushed by rahuldashetty	main	5 minutes ago	...
test workflow cicd attempt-4	Deploy to EC2 #4: Commit cd580d3 pushed by rahuldashetty	main	6 minutes ago	...
test workflow cicd attempt-3	Deploy to EC2 #3: Commit add498f pushed by rahuldashetty	main	11 minutes ago	...
test workflow cicd attempt-2	Deploy to EC2 #2: Commit 96c4ee2 pushed by rahuldashetty	main	13 minutes ago	Failure
test workflow cicd	Deploy to EC2 #1: Commit a5702ca pushed by rahuldashetty	main	22 minutes ago	...

rahulkanduri.ddns.net/mycode.php

Hello this is Rahul