How to Point Domain and Deploy ReactJS and VueJS Project using Github on Nginx Remote Server or VPS

- Get Access to Remote Server via SSH

```
Syntax:- ssh -p PORT USERNAME@HOSTIP
Example:- ssh -p 1034 raj@216.32.44.12
```

- Verify that all required softwares are installed

```
nginx -v
node -v
npm -v
git --version
```

- Install Software (If required)

```
sudo apt install nginx
sudo apt install git
curl -fsSL https://deb.nodesource.com/setup_18.x | sudo -E bash - &&\
sudo apt-get install -y nodejs
```

- Verify Nginx is Active and Running

```
sudo service nginx status
```

- Verify Web Server Ports are Open and Allowed through Firewall

```
sudo ufw status verbose
```

- Exit from Remote Server

```
exit
```

- Login to Your Domain Provider Website
- Navigate to Manage DNS
- Add Following Records:

Type	Host/Name	Value
A	@	Your Remote Server IP
A	www	Your Remote Server IP
AAAA	@	Your Remote Server IPv6
AAAA	www	Your Remote Server IPv6

- Copy Project from Local Machine to Remote Server or VPS. There are two ways to do it:-
  1. Using Command Prompt
     - On Local Windows Machine Make Your Project Folder a Zip File using any of the software e.g. winzip
     - Open Command Prompt

  - Copy Zip File from Local Windows Machine to Linux Remote Server

  ```
  Syntax:- scp -P Remote_Server_Port Source_File_Path Destination_Path
  Example:- scp -P 1034 miniblog.zip raj@216.32.44.12:
  ```

  - Copied Successfully
  - Get Access to Remote Server via SSH

  ```
  Syntax:- ssh -p PORT USERNAME@HOSTIP
  Example:- ssh -p 1034 raj@216.32.44.12
  ```

  - Unzip the Copied Project Zip File

  ```
  Syntax:- unzip zip_file_name
  Example:- unzip miniblog.zip
  ```

2. Using Github
  - Open Project on VS Code then add .gitignore file (If needed)
  - Push your Poject to Your Github Account as Private Repo
  - Make Connection between Remote Server and Github Repo via SSH Key
  - Generate SSH Keys

  ```
  Syntax:- ssh-keygen -t ed25519 -C "your_email@example.com"
  ```

  - If Permission Denied then Own .ssh then try again to Generate SSH Keys

  ```
  Syntax:- sudo chown -R user_name .ssh
  Example:- sudo chown -R raj .ssh
  ```

  - Open Public SSH Keys then copy the key

  ```
  cat ~/.ssh/id_ed25519.pub
  ```

  - Go to Your Github Repo
  - Click on Settings Tab
  - Click on Deploy Keys option from sidebar
  - Click on Add Deploy Key Button and Paste Remote Server's Copied SSH Public Key then Click on Add Key
  - Clone Project from your github Repo using SSH Path It requires to setup SSH Key on Github

  ```
  Syntax:- git clone ssh_repo_path
  Example:- git clone git@github.com:geekyshow1/miniblog.git
  ```

- Move Project Folder to Web Server public directory

```
Syntax:- sudo mv project_folder_name /var/www
Example:- sudo mv miniblog /var/www
```

- Install Dependencies

```
npm install
```

- Create Production Build

```
npm run build
```

- Create Virtual Host File

```
Syntax:- sudo nano /etc/nginx/sites-available/your_domain
```

```
Example:- sudo nano /etc/nginx/sites-available/sonamkumari.com
```

- Write following Code in Virtual Host File

```
server{
    listen 80;
    listen [::]:80;
    server_name your_domain www.your_domain;
    root /var/www/project_folder_name/production_build_folder_name;
    index index.html;
    location / {
        try_files $uri $uri/ =404;
    }
}
```

- Enable Virtual Host or Create Symbolic Link of Virtual Host File

```
cd /etc/nginx/sites-available/
Syntax:- sudo ln -s /etc/nginx/sites-available/virtual_host_file /etc/nginx/sites-enabled/vi
Example:- sudo ln -s /etc/nginx/sites-available/sonamkumari.com /etc/nginx/sites-enabled/sor
```

- Check Configuration is Correct or Not

```
sudo nginx -t
```

- Restart Nginx

```
sudo service nginx restart
```

- Now you can make some changes in your project local development VS Code and Pull it on Remote Server (Only if you have used Github)
- Go to Your Project Directory

```
Syntax:- cd /var/www/project_folder_name
Example:- cd /var/www/miniblog
```

- Pull the changes from github repo

```
git pull
```

- Create Production Build

```
npm run build
```

How to Automate ReactJS and VueJS Project Deployment using Github Action

- On Your Local Machine, Open Your Project using VS Code or any Editor
- Create A Folder named .scripts inside your root project folder e.g. miniblog/.scripts
- Inside .scripts folder Create A file with .sh extension e.g. miniblog/.scripts/deploy.sh

- Write below script inside the created .sh file

```bash
#!/bin/bash
set -e

echo "Deployment started..."

# Pull the latest version of the app
git pull origin master
echo "New changes copied to server !"

echo "Installing Dependencies..."
npm install --yes

echo "Creating Production Build..."
npm run build

echo "Deployment Finished!"
```

- Go inside .scripts Folder then Set File Permission for .sh File

```
git update-index --add --chmod=+x deploy.sh
```

- Create Directory Path named .github/workflows inside your root project folder e.g. miniblog/.github/workflows
- Inside workflows folder Create A file with .yml extension e.g. miniblog/.github/workflows/deploy.yml
- Write below script inside the created .yml file

```yaml
name: Deploy

# Trigger the workflow on push and
# pull request events on the master branch
on:
  push:
    branches: ["master"]
  pull_request:
    branches: ["master"]

# Authenticate to the the server via ssh
# and run our deployment script
jobs:
  deploy:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v3
      - name: Deploy to Server
        uses: appleboy/ssh-action@master
        with:
          host: ${{ secrets.HOST }}
          username: ${{ secrets.USERNAME }}
          port: ${{ secrets.PORT }}
          key: ${{ secrets.SSHKEY }}
          script: "cd /var/www/project_folder_name && ./.scripts/deploy.sh"
```

- Go to Your Github Repo Click on Settings
- Click on Secrets and Variables from the Sidebar then choose Actions
- On Secret Tab, Click on New Repository Secret
- Add Four Secrets HOST, PORT, USERNAME and SSHKEY as below

```
Name: HOST
Secret: Your_Server_IP
```

```
Name: PORT
Secret: Your_Server_PORT
```

```
Name: USERNAME
Secret: Your_Server_User_Name
```

- You can get Server User Name by loging into your server via ssh then run below command

```
whoami
```

- Generate SSH Key for Github Action by Login into Remote Server then run below Command

```
Syntax:- ssh-keygen -f key_path -t ed25519 -C "your_email@example.com"
Example:- ssh-keygen -f /home/raj/.ssh/gitaction_ed25519 -t ed25519 -C "gitactionautodep"
```

- Open Newly Created Public SSH Keys then copy the key

```
cat ~/.ssh/gitaction_ed25519.pub
```

- Open authorized_keys File which is inside .ssh/authroized_keys then paste the copied key in a new line

```
cd .ssh
nano authorized_keys
```

- Open Newly Created Private SSH Keys then copy the key, we will use this key to add New Repository Secret On Github Repo

```
cat ~/.ssh/gitaction_ed25519
```

```
Name: SSHKEY
Secret: Private_SSH_KEY_Generated_On_Server
```

- Commit and Push the change to Your Github Repo
- Get Access to Remote Server via SSH

```
Syntax:- ssh -p PORT USERNAME@HOSTIP
Example:- ssh -p 22 raj@216.32.44.12
```

- Go to Your Project Directory

```
Syntax:- cd /var/www/project_folder_name
Example:- cd /var/www/miniblog
```

- Pull the changes from github just once this time.

```
git pull
```

- Your Deployment should become automate.
- On Local Machine make some changes in Your Project then Commit and Push to Github Repo It will automatically deployed on Live Server
- You can track your action from Github Actions Tab
- If you get any File Permission error in the action then you have to change file permission accordingly.
- All Done