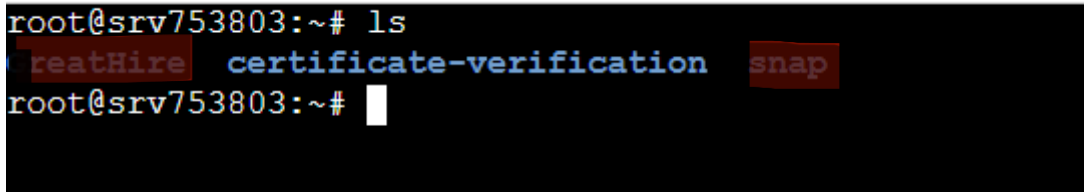
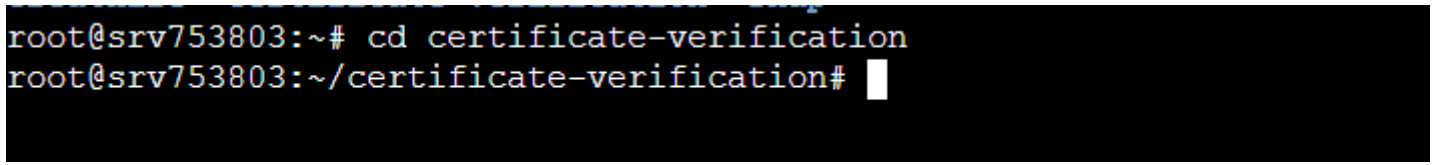
GREATHIRE EMPLOYEE VERIFICATION DEPLOYMENT DOCUMENTATION

1. Login into the vps dashboard in the Hostinger.
2. After logging to the server through the vps browser terminal.
3. Navigate to the root directory of the server.
4. Perform the **git cone <command>** to copy the entire code from the GitHub.
5. After that it appears like this.



6. Now enter into the certificate-verification folder through the CLI
7. Cd <certificate-verification> now you will be in the inside the certificate-verification folder.



1. Perform ls command in the certificate-verification .
2. And navigate to the frontend and backend folders and perform the npm install on both folders.

```
root@srv753803:~/certificate-verification# ls
README.md  backend  frontend  node_modules
root@srv753803:~/certificate-verification# cd frontend
root@srv753803:~/certificate-verification/frontend# npm install

up to date, audited 229 packages in 1s

45 packages are looking for funding
  run `npm fund` for details

2 moderate severity vulnerabilities

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
root@srv753803:~/certificate-verification/frontend# cd ..
root@srv753803:~/certificate-verification# cd backend
root@srv753803:~/certificate-verification/backend# npm install

up to date, audited 122 packages in 843ms

20 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
root@srv753803:~/certificate-verification/backend#
```

3. Navigate to the frontend folder and perform the npm run build command --> to build the static files for production level.

```
root@srv753803:~/certificate-verification# cd frontend
root@srv753803:~/certificate-verification/frontend# npm run build

> frontend@0.1.0 build
> vite build

vite v5.4.19 building for production...
✓ 1496 modules transformed.
dist/index.html                   0.47 kB | gzip:  0.31 kB
dist/assets/index-mzkx2Gky.css   19.53 kB | gzip:  4.51 kB
dist/assets/index-DxY2kjj_.js   216.00 kB | gzip: 65.28 kB
✓ built in 4.03s
root@srv753803:~/certificate-verification/frontend#
```

4. Navigate to the backend/src folder …. And run the backend index.js through the CLI.
   - pm2 start index.js
   - pm2 save
   - pm2 startup

* Here the pm2 commands which are used to start the backend automatically when the server is rebooted or restarted.

```
root@srv753803:~/certificate-verification/backend/src# ls
index.js  models  routes
root@srv753803:~/certificate-verification/backend/src# pm2 start index.js
```

1. Make sure that .env files of backend present at the backend/src/.env like in this path.
2. You can verify through the cat command in the terminal

-> inside the backend/src/

->perform the cat .env

- To see the env files of the backend

Nginx setup:-

1. Navigate to the root hierarchy of the server.

```
root@srv753803:~/certificate-verification/backend/src# cd ../../..
root@srv753803:~# l
GreatHire/  certificate-verification/  snap/
root@srv753803:~# cd ..
root@srv753803:/# ls
bin  boot  dev  etc  home  lib  lib32  lib64  libx32  lost+found  media  mnt  opt  proc  root  run  sbin  snap  srv  sys  tmp  usr  var
root@srv753803:/#
```

2. Now Navigate to the etc/nginx/sites-available folders through the command
   - cd etc/nginx/sites-available
3. Inside the **sites-available folder** there is a **certificate file**.

```
root@srv753803:/# ls
bin  boot  dev  etc  home  lib  lib32  lib64  libx32  lost+found  media  mnt  opt  proc  root  run  sbin  snap  srv  sys  tmp  usr  var
root@srv753803:/# cd etc/nginx
root@srv753803:/etc/nginx# ls
conf.d       fastcgi_params  koi-win     modules-available  nginx.conf    scgi_params      sites-enabled  uwsgi_params
fastcgi.conf  koi-utf        mime.types  modules-enabled    proxy_params  sites-available  snippets       win-utf
root@srv753803:/etc/nginx# cd sites-available
root@srv753803:/etc/nginx/sites-available# ls
certificate
root@srv753803:/etc/nginx/sites-available#
```

4. Open the certificate file through the nano command.

-> nano certificate

```
server {
    listen 80;
    server_name emp.greathire.in;
    return 301 https://$host$request_uri;
}

server {
    listen 443 ssl;
    server_name emp.greathire.in;

    ssl_certificate /etc/letsencrypt/live/emp.greathire.in/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/emp.greathire.in/privkey.pem;

    location /api/ {
        proxy_pass http://localhost:5000;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }

    location / {
        root /root/certificate-verification/frontend/dist;
        index index.html;
        try_files $uri /index.html;
    }
}
```

4. Save the file with the help of commands

-> ctrl+o,enter,ctrl+x

After that perform the **<nginx -t>** to know the any syntax issue's in the template

- nginx –t

```
root@srv753803:/etc/nginx/sites-available# nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
root@srv753803:/etc/nginx/sites-available#
```

5. After that reload the nginx through the command.

- systemctl reload nginx

6. Restart nginx

- systemctl restart nginx

7. If you want the SSL certificate for the website … request the SSL certificate through the certbot Installation.

-

```bash
sudo apt update
sudo apt install certbot python3-certbot-nginx
```

8. If you want the SSL certificate for the website … request the SSL certificate through the certbot

-
```
sudo certbot --nginx -d emp.greathire.in
```

9. Make sure that you allow the necessary ports for the application through the **ufw command.**

-ufw allow <port number>

10. Finally our websites get’s live -> in the browser.