

## Personal Home Cloud with Secure Multi-Level Access (Systems & Networking)

For that, I am using raspberypie 3b and for reliable access provided with ssd and a network card which supports monitor mode

using rasp os/ubuntu on that top of that for more easier access

### Set Up Disk Encryption for security part

**\*\*sudo apt install cryptsetup**

```
Processing triggers for man-db (2.10.2-1) ...
web@web-ThinkCentre-neo-50t-Gen-3:~$ sudo apt install cryptsetup
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Progress: [ 8%] [#####]
libflashrom1 libftdi1-2 liblvm13
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  cryptsetup-bin cryptsetup-initramfs
Suggested packages:
  keyutils
The following NEW packages will be installed:
  cryptsetup cryptsetup-bin cryptsetup-initramfs
0 upgraded, 3 newly installed, 0 to remove and 14 not upgraded.
Need to get 365 kB of archives.
After this operation, 1,246 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 cryptsetup-bin amd64 2:2.4.3-1ubuntu1.3 [145 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 cryptsetup amd64 2:2.4.3-1ubuntu1.3 [194 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 cryptsetup-initramfs all 2:2.4.3-1ubuntu1.3 [25.6 kB]
Fetched 365 kB in 5s (71.6 kB/s)
Preconfiguring packages ...
Selecting previously unselected package cryptsetup-bin.
(Reading database ... 217343 files and directories currently installed.)
Preparing to unpack .../cryptsetup-bin_2%3a2.4.3-1ubuntu1.3_amd64.deb ...
Unpacking cryptsetup-bin (2:2.4.3-1ubuntu1.3) ...
Selecting previously unselected package cryptsetup.
Preparing to unpack .../cryptsetup_2%3a2.4.3-1ubuntu1.3_amd64.deb ...
Unpacking cryptsetup (2:2.4.3-1ubuntu1.3) ...
Selecting previously unselected package cryptsetup-initramfs.
Preparing to unpack .../cryptsetup-initramfs_2%3a2.4.3-1ubuntu1.3_all.deb ...
Unpacking cryptsetup-initramfs (2:2.4.3-1ubuntu1.3) ...
Setting up cryptsetup-bin (2:2.4.3-1ubuntu1.3) ...
Setting up cryptsetup (2:2.4.3-1ubuntu1.3) ...
Setting up cryptsetup-initramfs (2:2.4.3-1ubuntu1.3) ...
update-initramfs: deferring update (trigger activated)
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for initramfs-tools (0.140ubuntu13.5) ...
update-initramfs: Generating /boot/initrd.img-6.8.0-79-generic
Progress: [ 92%] [#####]
```

## Install and Configure the VPN Server (WireGuard)

Why WireGuard?

- Fast and lightweight
- Easy to set up
- Strong encryption

**\*\*sudo apt install wireguard -y**

```

web@web-ThinkCentre-neo-50t-Gen-3:~$ sudo apt install wireguard
[sudo] password for web:
Sorry, try again.
[sudo] password for web:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libflashrom1 libftdi1-2 libllvm13
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  wireguard-tools
Suggested packages:
  openresolv | resolvconf
The following NEW packages will be installed:
  wireguard wireguard-tools
0 upgraded, 2 newly installed, 0 to remove and 14 not upgraded.
Need to get 90.0 kB of archives.
After this operation, 345 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 wireguard-tools amd64 1.0.20210914-1ubuntu2 [86.9 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu jammy/universe amd64 wireguard all 1.0.20210914-1ubuntu2 [3,114 B]
Fetched 90.0 kB in 7s (12.2 kB/s)
Selecting previously unselected package wireguard-tools.
(Reading database ... 217263 files and directories currently installed.)
Preparing to unpack .../wireguard-tools_1.0.20210914-1ubuntu2_amd64.deb ...
Unpacking wireguard-tools (1.0.20210914-1ubuntu2) ...
Selecting previously unselected package wireguard.
Preparing to unpack .../wireguard_1.0.20210914-1ubuntu2_all.deb ...
Unpacking wireguard (1.0.20210914-1ubuntu2) ...
Setting up wireguard-tools (1.0.20210914-1ubuntu2) ...
wg-quick.target is a disabled or a static unit not running, not starting it.
Setting up wireguard (1.0.20210914-1ubuntu2) ...
Processing triggers for man-db (2.10.2-1) ...
web@web-ThinkCentre-neo-50t-Gen-3:~$ sudo apt install cryptsetup

```

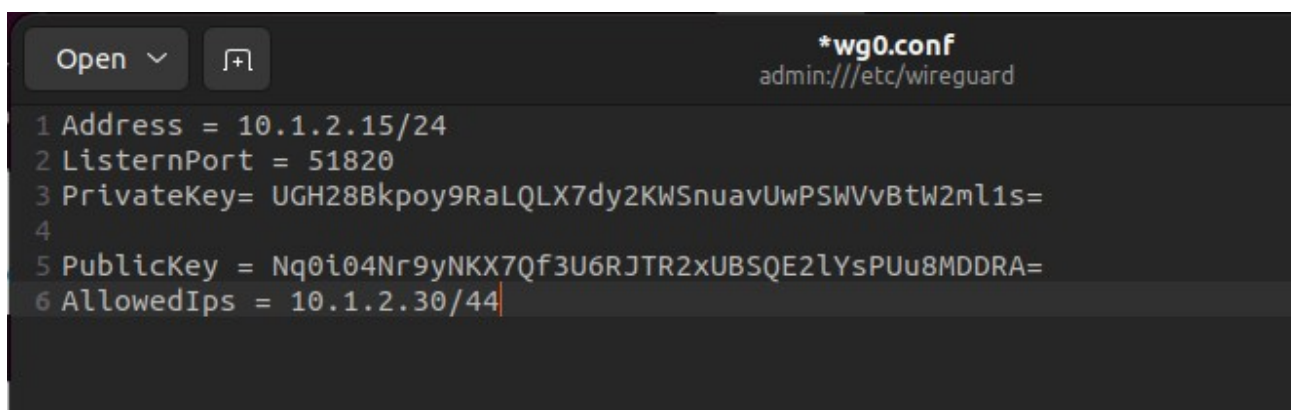
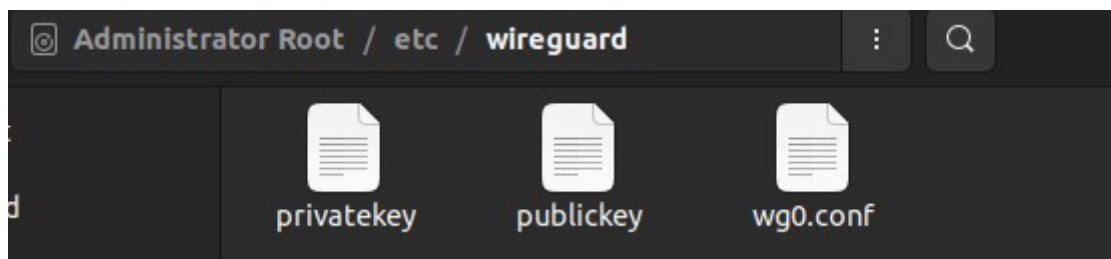
## Generating keys:

`wg genkey | tee privatekey | wg pubkey > publickey`

```

web@web-ThinkCentre-neo-50t-Gen-3:~$ wg genkey | tee privatekey | wg pubkey > publickey
web@web-ThinkCentre-neo-50t-Gen-3:~$

```



## Running the wireguard service and as we can that my ip is listed here (10.1.2.15)

```
root@web-ThinkCentre-neo-50t-Gen-3:/etc/wireguard# ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eno2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether e0:be:03:7c:5a:50 brd ff:ff:ff:ff:ff:ff
    altnam enp0s31f6
    inet 10.1.2.15/8 brd 10.255.255.255 scope global noprefixroute eno2
        valid_lft forever preferred_lft forever
    inet6 fe80::2d56:9fb2:721e:d6bd/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: wlo1: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default qlen 1000
    link/ether c4:75:ab:02:d1:09 brd ff:ff:ff:ff:ff:ff
    altnam wlp0s20f3
root@web-ThinkCentre-neo-50t-Gen-3:/etc/wireguard# sudo nano /etc/sysctl.conf
root@web-ThinkCentre-neo-50t-Gen-3:/etc/wireguard# ^[[200~sudo sysctl -p
sudo: command not found
root@web-ThinkCentre-neo-50t-Gen-3:/etc/wireguard# sudo sysctl -p
net.ipv4.ip_forward = 1
root@web-ThinkCentre-neo-50t-Gen-3:/etc/wireguard# sudo systemctl start wg-quick@wg0
sudo systemctl enable wg-quick@wg0
root@web-ThinkCentre-neo-50t-Gen-3:/etc/wireguard# sudo systemctl start wg-quick@wg0
sudo systemctl enable wg-quick@wg0
root@web-ThinkCentre-neo-50t-Gen-3:/etc/wireguard# sudo systemctl status wg-quick@wg0
● wg-quick@wg0.service - WireGuard via wg-quick(8) for wg0
   Loaded: loaded (/lib/systemd/systemd/wg-quick@.service; enabled; vendor preset: enabled)
   Active: active (exited) since Mon 2025-09-15 12:59:53 IST; 11s ago
     Docs: man:wg-quick(8)
           man:wg(8)
           https://www.wireguard.com/
           https://www.wireguard.com/quickstart/
           https://git.zx2c4.com/wireguard-tools/about/src/man/wg-quick.8
           https://git.zx2c4.com/wireguard-tools/about/src/man/wg.8
   Main PID: 24435 (code=exited, status=0/SUCCESS)
     CPU: 35ms

Sep 15 12:59:53 web-ThinkCentre-neo-50t-Gen-3 systemd[1]: Starting WireGuard via wg-quick(8) for wg0>
Sep 15 12:59:53 web-ThinkCentre-neo-50t-Gen-3 wg-quick[24435]: [#] ip link add wg0 type wireguard
Sep 15 12:59:53 web-ThinkCentre-neo-50t-Gen-3 wg-quick[24435]: [#] wg setconf wg0 /dev/fd/63
Sep 15 12:59:53 web-ThinkCentre-neo-50t-Gen-3 wg-quick[24435]: [#] ip -4 address add 10.1.2.1/24 dev>
Sep 15 12:59:53 web-ThinkCentre-neo-50t-Gen-3 wg-quick[24435]: [#] ip link set mtu 1420 up dev wg0
Sep 15 12:59:53 web-ThinkCentre-neo-50t-Gen-3 wg-quick[24435]: [#] iptables -A FORWARD -i wg0 -j ACC>
Sep 15 12:59:53 web-ThinkCentre-neo-50t-Gen-3 systemd[1]: Finished WireGuard via wg-quick(8) for wg0.
lines 1-19/19 (END)
```

## Set Up HTTPS with Nginx and Encryption stage

### Install Certbot:

```

web@web-ThinkCentre-neo-50t-Gen-3:~$ sudo apt install certbot python3-certbot-nginx -y
[sudo] password for web:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libflashrom1 libftdi1-2 libllvm13
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libnginx-mod-http-geoip2 libnginx-mod-http-image-filter
  libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream
  libnginx-mod-stream-geoip2 nginx nginx-common nginx-core python3-acme
  python3-certbot python3-configargparse python3-configobj python3-icu
  python3-josepy python3-openssl python3-parsedatetime
  python3-requests-toolbelt python3-zope.component python3-zope.event
  python3-zope.hookable python3-zope.interface
Suggested packages:
  python-certbot-doc python3-certbot-apache fcgiwrap nginx-doc python-acme-doc
  python-certbot-nginx-doc python-configobj-doc python-openssl-doc
  python3-openssl-dbg
The following NEW packages will be installed:
  certbot libnginx-mod-http-geoip2 libnginx-mod-http-image-filter
  libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream
  libnginx-mod-stream-geoip2 nginx nginx-common nginx-core python3-acme
  python3-certbot python3-certbot-nginx python3-configargparse
  python3-configobj python3-icu python3-josepy python3-openssl
  python3-parsedatetime python3-requests-toolbelt python3-zope.component
  python3-zope.event python3-zope.hookable python3-zope.interface
0 upgraded, 24 newly installed, 0 to remove and 14 not upgraded.
Need to get 1,906 kB of archives.
After this operation, 8,827 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 nginx-common all 1.18.0-6ubuntu14.7 [40.1 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libnginx-mod-http-geoip2 amd64 1.18.0-6ubuntu14.7 [12.0 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libnginx-mod-http-image-filter amd64 1.18.0-6ubuntu14.7 [15.5 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libnginx-mod-http-xslt-filter amd64 1.18.0-6ubuntu14.7 [13.8 kB]
Get:5 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libnginx-mod-mail amd64 1.18.0-6ubuntu14.7 [45.8 kB]
Get:6 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libnginx-mod-stream amd64 1.18.0-6ubuntu14.7 [73.0 kB]
Get:7 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libnginx-mod-stream-geoip2 amd64 1.18.0-6ubuntu14.7 [10.1 kB]
Get:8 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 nginx-core amd64 1.18.0-6ubuntu14.7 [483 kB]
Get:9 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 nginx amd64 1.18.0-6ubuntu14.7 [3,

```

## Starting nginx for https

```

web@web-ThinkCentre-neo-50t-Gen-3:~$ sudo apt install nginx -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
nginx is already the newest version (1.18.0-6ubuntu14.7).
nginx set to manually installed.
The following packages were automatically installed and are no longer required:
  libflashrom1 libftdi1-2 libllvm13
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 14 not upgraded.
web@web-ThinkCentre-neo-50t-Gen-3:~$ sudo systemctl start nginx
sudo systemctl enable nginx
Synchronizing state of nginx.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable nginx
web@web-ThinkCentre-neo-50t-Gen-3:~$ sudo 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 2025-09-15 13:03:12 IST; 1min 39s ago
     Docs: man:nginx(8)
  Main PID: 25407 (nginx)
    Tasks: 13 (limit: 18702)
   Memory: 11.3M
      CPU: 49ms
  CGroup: /system.slice/nginx.service
           └─25407 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             └─25409 "nginx: worker process"
             └─25410 "nginx: worker process"
             └─25411 "nginx: worker process"
             └─25412 "nginx: worker process"
             └─25413 "nginx: worker process"
             └─25414 "nginx: worker process"
             └─25415 "nginx: worker process"
             └─25416 "nginx: worker process"
             └─25417 "nginx: worker process"
             └─25418 "nginx: worker process"
             └─25419 "nginx: worker process"
             └─25420 "nginx: worker process"

Sep 15 13:03:12 web-ThinkCentre-neo-50t-Gen-3 systemd[1]: Starting A high performance web server and
Sep 15 13:03:12 web-ThinkCentre-neo-50t-Gen-3 systemd[1]: Started A high performance web server and
lines 1-25/25 (END)

```

## Allow HTTPS through the firewall

```

web@web-ThinkCentre-neo-50t-Gen-3:~$ sudo ufw allow 'Nginx Full'
[sudo] password for web:
Rules updated
Rules updated (v6)
web@web-ThinkCentre-neo-50t-Gen-3:~$

```

## Configure Nginx for your domain



```
wg0.conf  ×  privatekey  ×  publickey  ×  *homecloud  ×
1 server {
2     listen 80;
3     server_name your-domain.com www.your-domain.com;
4
5     root /var/www/homecloud;
6     index index.html index.htm;
7
8     location / {
9         try_files $uri $uri/ =404;
10    }
11
12    location /wg/ {
13        proxy_pass http://localhost:51820; # Or your WireGuard service if needed
14        proxy_set_header Host $host;
15        proxy_set_header X-Real-IP $remote_addr;
16    }
17 }
18
19
20
```

**Enable the site:** (and as we can see that the site is successfully enabled)

```
web@web-ThinkCentre-neo-50t-Gen-3:~$ sudo ln -s /etc/nginx/sites-available/homecloud /etc/nginx/sites-enabled/
sudo nginx -t
sudo systemctl reload nginx
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
web@web-ThinkCentre-neo-50t-Gen-3:~$
```

**proof that wireguard vpn is running**

```
web@web-ThinkCentre-neo-50t-Gen-3:~$ sudo systemctl status wg-quick@wg0
[sudo] password for web:
● wg-quick@wg0.service - WireGuard via wg-quick(8) for wg0
   Loaded: loaded (/lib/systemd/system/wg-quick@.service; enabled; vendor pre>
   Active: active (exited) since Mon 2025-09-15 12:59:53 IST; 12min ago
     Docs: man:wg-quick(8)
           man:wg(8)
           https://www.wireguard.com/
           https://www.wireguard.com/quickstart/
           https://git.zx2c4.com/wireguard-tools/about/src/man/wg-quick.8
           https://git.zx2c4.com/wireguard-tools/about/src/man/wg.8
   Main PID: 24435 (code=exited, status=0/SUCCESS)
      CPU: 35ms

Sep 15 12:59:53 web-ThinkCentre-neo-50t-Gen-3 systemd[1]: Starting WireGuard vi>
Sep 15 12:59:53 web-ThinkCentre-neo-50t-Gen-3 wg-quick[24435]: [#] ip link add >
Sep 15 12:59:53 web-ThinkCentre-neo-50t-Gen-3 wg-quick[24435]: [#] wg setconf w>
Sep 15 12:59:53 web-ThinkCentre-neo-50t-Gen-3 wg-quick[24435]: [#] ip -4 addres>
Sep 15 12:59:53 web-ThinkCentre-neo-50t-Gen-3 wg-quick[24435]: [#] ip link set >
Sep 15 12:59:53 web-ThinkCentre-neo-50t-Gen-3 wg-quick[24435]: [#] iptables -A >
Sep 15 12:59:53 web-ThinkCentre-neo-50t-Gen-3 systemd[1]: Finished WireGuard vi>
lines 1-19/19 (END)
```

vpn successfully configured and listening

```
valld_tlt forever preferred_tlt forever
web@web-ThinkCentre-neo-50t-Gen-3:~$ sudo wg
Interface: wg0
public key: Nq0i04Nr9yNKX7Qf3U6RJTR2xUBSQE2LYsPUu8MDDRA=
private key: (hidden)
listening port: 51820
web@web-ThinkCentre-neo-50t-Gen-3:~$
```

\  
\

checking vpn locally and traffic is flowing or not through local

```
web@web-ThinkCentre-neo-50t-Gen-3:~$ ping -I wg0 10.1.2.1
PING 10.1.2.1 (10.1.2.1) from 10.1.2.1 wg0: 56(84) bytes of data.
64 bytes from 10.1.2.1: icmp_seq=1 ttl=64 time=0.042 ms
64 bytes from 10.1.2.1: icmp_seq=2 ttl=64 time=0.032 ms
64 bytes from 10.1.2.1: icmp_seq=3 ttl=64 time=0.035 ms
64 bytes from 10.1.2.1: icmp_seq=4 ttl=64 time=0.035 ms
64 bytes from 10.1.2.1: icmp_seq=5 ttl=64 time=0.035 ms
64 bytes from 10.1.2.1: icmp_seq=6 ttl=64 time=0.034 ms
```

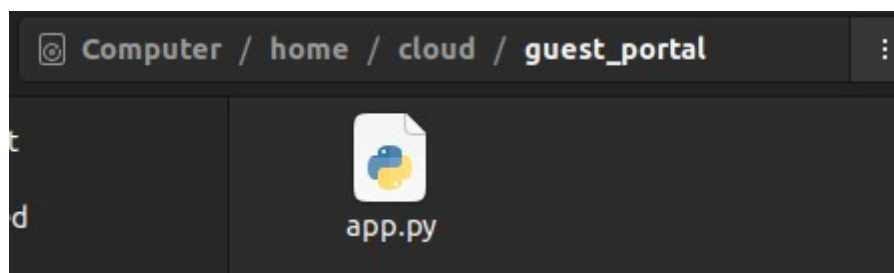
//

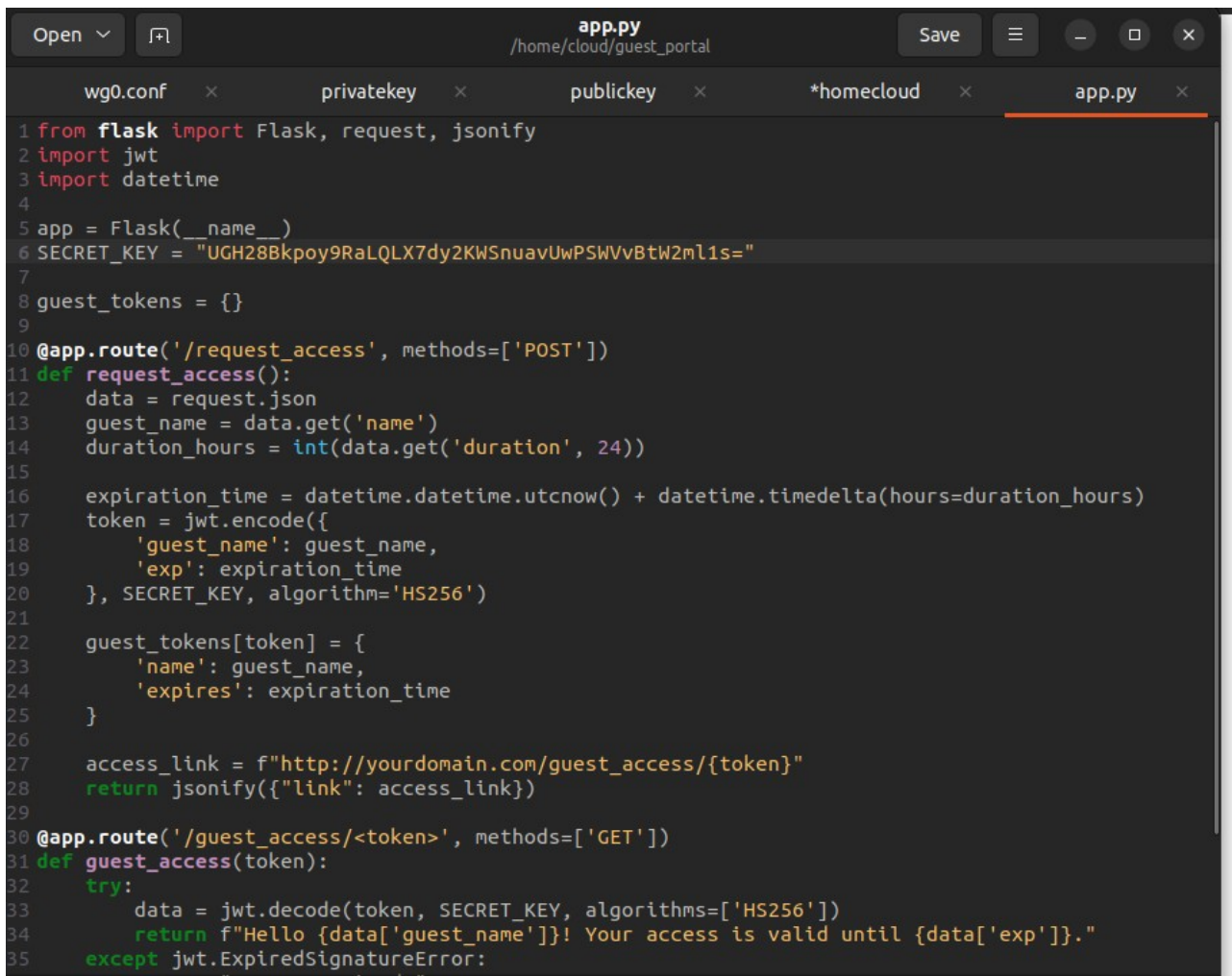
Now, there are two methods for adding guest with restricted, time bound access

consider web based port is easy and fast will move forward to that

installing tools for it as python3-pip, nginx, flask, pyjwt

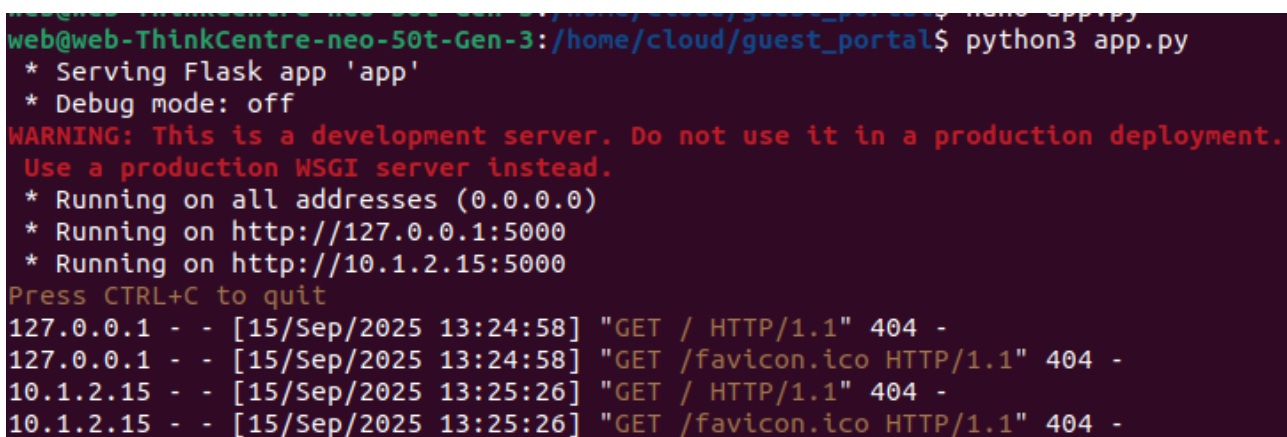
and creating directory for guest portal and creating flask app





```
1 from flask import Flask, request, jsonify
2 import jwt
3 import datetime
4
5 app = Flask(__name__)
6 SECRET_KEY = "UGH28Bkpoy9RaLQLX7dy2KWSnuavUwPSWVvBtW2m1s="
7
8 guest_tokens = {}
9
10 @app.route('/request_access', methods=['POST'])
11 def request_access():
12     data = request.json
13     guest_name = data.get('name')
14     duration_hours = int(data.get('duration', 24))
15
16     expiration_time = datetime.datetime.utcnow() + datetime.timedelta(hours=duration_hours)
17     token = jwt.encode({
18         'guest_name': guest_name,
19         'exp': expiration_time
20     }, SECRET_KEY, algorithm='HS256')
21
22     guest_tokens[token] = {
23         'name': guest_name,
24         'expires': expiration_time
25     }
26
27     access_link = f"http://yourdomain.com/guest_access/{token}"
28     return jsonify({"link": access_link})
29
30 @app.route('/guest_access/<token>', methods=['GET'])
31 def guest_access(token):
32     try:
33         data = jwt.decode(token, SECRET_KEY, algorithms=['HS256'])
34         return f"Hello {data['guest_name']}! Your access is valid until {data['exp']}."
35     except jwt.ExpiredSignatureError:
```

and running it



```
web@web-ThinkCentre-neo-50t-Gen-3:/home/cloud/guest_portal$ python3 app.py
* Serving Flask app 'app'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://10.1.2.15:5000
Press CTRL+C to quit
127.0.0.1 - - [15/Sep/2025 13:24:58] "GET / HTTP/1.1" 404 -
127.0.0.1 - - [15/Sep/2025 13:24:58] "GET /favicon.ico HTTP/1.1" 404 -
10.1.2.15 - - [15/Sep/2025 13:25:26] "GET / HTTP/1.1" 404 -
10.1.2.15 - - [15/Sep/2025 13:25:26] "GET /favicon.ico HTTP/1.1" 404 -
```

While adding backend for user policies, session and access control

and basic frontend via flask

(all files are uploaded on github with supporting screenshots)



## **Conclusion**

**Personal Home cloud is set with multi level acces**

**// Home Cloud Server & Segmentation – done with encrypted storage via crptdisk**

**// Role-Based Access Control – done via routing only home ips**

**// Guest Access Portal – Done via basic backend tokens and frontend via flask**

**// Network Security Controls -tested and running successfully**

**// Encryption & Monitoring – Done and tested via above ss**

**// Secure Storage – Done via routing tables and crptdisk setup**

**// Backend Management – Simple backend is implemented via sql and flask apis**

*Chaitanya Lade*

*github - Chaitanyx*