

# Using Algo VPN to access sdnog Infrastructure

This page was written by Manhal Mohamed, sdnog team, on 8 August 2024.

Algo VPN simplifies deploying a secure VPN server across multiple platforms. This guide provides a step-by-step walkthrough for setting up Algo VPN on a local Ubuntu server to securely access the sdnog infrastructure.

## Prerequisites

Before starting, ensure the following:

- **Operating System:** Ubuntu Server (18.04 or later)
- **Privileges:** Sudo access on the server
- **Skills:** Basic familiarity with command-line operations

## Step-by-Step Setup

### 1. Update Your System

Before installing Algo VPN, ensure that your system is up-to-date. Open a terminal and run the following commands:

```
sudo apt update
sudo apt upgrade -y
```

### 2. Install Dependencies

Algo VPN requires certain dependencies to be installed. Use the following commands to install them:

```
apt-get install git apparmor build-essential python3-dev python3-pip python3-setuptools python3-virtualenv
libffi-dev libssl-dev -y
```

### 3. Clone the Algo VPN Repository

Clone the Algo VPN repository from GitHub to your local server:

```
git clone https://github.com/trailofbits/algo.git
cd algo
```

## 4. Create and Activate a Python Virtual Environment

Create a Python virtual environment and activate it:

```
cd algo
python3 -m virtualenv --python=/usr/bin/python3 .env
source .env/bin/activate
```

## 5. Install Algo VPN

Install Algo VPN and its dependencies using pip:

```
python3 -m pip install -U pip virtualenv
python3 -m pip install -r requirements.txt
```

## 6. Configure Algo VPN

Run the Algo VPN setup script to create a configuration file:

```
./algo
```

Follow the prompts to configure your VPN. You will need to provide details such as:

The VPN server's public IP address or domain name Your preferred VPN protocol (e.g., WireGuard or IPsec) User accounts for VPN access

## 7. Deploy Algo VPN

Once the configuration is complete, deploy Algo VPN with the following command:

```
./algo
```

The deployment process will set up the VPN server according to the configuration you provided.

```
TASK [Set required ansible version as a fact] *****
ok: [localhost] => (item=ansible==2.9.7)

TASK [Verify Python meets Algo VPN requirements] *****
ok: [localhost] => {
  "changed": false,
  "msg": "All assertions passed"
}

TASK [Verify Ansible meets Algo VPN requirements] *****
```

```
ok: [localhost] => {
  "changed": false,
  "msg": "All assertions passed"
}
[WARNING]: Found variable using reserved name: no_log
```

PLAY [Ask user for the input] \*\*\*\*\*

TASK [Gathering Facts] \*\*\*\*\*

```
ok: [localhost]
[Cloud prompt]
What provider would you like to use?
```

1. DigitalOcean
2. Amazon Lightsail
3. Amazon EC2
4. Microsoft Azure
5. Google Compute Engine
6. Hetzner Cloud
7. Vultr
8. Scaleway
9. OpenStack (DreamCompute optimised)
10. CloudStack (Exoscale optimised)
11. Linode
12. Install to existing Ubuntu 18.04 or 20.04 server (for more advanced users)

Enter the number of your desired provider

```
:
```

12

Type 12 and hit Enter to setup Algo VPN on Ubuntu 20.04 server. You will be asked for several questions as shown

TASK [Set facts based on the input] \*\*\*\*\*

```
ok: [localhost]
[Cellular On Demand prompt]
Do you want macOS/iOS IPsec clients to enable "Connect On Demand" when connected to cellular networks?
[y/N]
:y
```

TASK [Cellular On Demand prompt] \*\*\*\*\*

```
ok: [localhost]
[Wi-Fi On Demand prompt]
Do you want macOS/iOS IPsec clients to enable "Connect On Demand" when connected to Wi-Fi?
[y/N]
:y
```

TASK [Wi-Fi On Demand prompt] \*\*\*\*\*

```
ok: [localhost]
[Trusted Wi-Fi networks prompt]
List the names of any trusted Wi-Fi networks where macOS/iOS IPsec clients should not use "Connect On Demand"
(e.g., your home network. Comma-separated value, e.g., HomeNet,OfficeWifi,AlgoWiFi)
:HomeNet
```

TASK [Trusted Wi-Fi networks prompt] \*\*\*\*\*

```

ok: [localhost]
[Compatible ciphers prompt]
Do you want the VPN to support Windows 10 or Linux Desktop clients? (enables compatible ciphers and key exchange)
[y/N]
:y

TASK [Compatible ciphers prompt] *****
ok: [localhost]
[Retain the CA key prompt]
Do you want to retain the CA key? (required to add users in the future, but less secure)
[y/N]
:y

TASK [Retain the CA key prompt] *****
ok: [localhost]
[DNS adblocking prompt]
Do you want to install an ad blocking DNS resolver on this VPN server?
[y/N]
:y

TASK [DNS adblocking prompt] *****
ok: [localhost]
[SSH tunneling prompt]
Do you want each user to have their own account for SSH tunneling?
[y/N]
:N
Enter the IP address of your server: (or use localhost for local installation):
[localhost]
:
localhost
TASK [local : pause] *****
ok: [localhost]

TASK [local : Set the facts] *****
ok: [localhost]
[local : pause]
What user should we use to login on the server? (note: passwordless login required, or ignore if you're deploying to root)
[root]
:
root

Enter the public IP address or domain name of your server: (IMPORTANT! This is used to verify the certificate)
vpn.jnb.sdnog.sd

```

## 8. Access sdnog Infrastructure

Once the installation has been completed successfully, you should get the following output:

```

TASK [debug] *****
ok: [localhost] => {
  "msg": [
    [

```

```

"\#          Congratulations!          #\",
"\#          Your Algo server is running.          #\",
"\# Config files and certificates are in the ./configs/ directory.  #\",
"\#          Go to https://whoer.net/ after connecting          #\",
"\#          and ensure that all your traffic passes through the VPN.  #\",
"\#          Local DNS resolver 172.18.7.104          #\",
""
],
"  \"#      The p12 and SSH keys password for new users is 7OefSUZt0  #\"\\n\",
"  \"#      The CA key password is g5AvchZygjV@4AN  #\"\\n\",
"  \"
]
}

```

```

PLAY RECAP *****
localhost      : ok=125  changed=39  unreachable=0  failed=0  skipped=53  rescued=0  ignored=0

```

After the installation, you should see the configuration file for each VPN profile using the following command:

```
ls configs/your-server-ip/wireguard/
```

You should see all the profile in the following output:

```
apple desktop.conf desktop.png laptop.conf laptop.png phone.conf phone.png user1.conf user1.png
```

You can use any of the above files on your client device to connect to the Algo VPN server.

To access sdnog infrastructure via the VPN, you need to configure your local machine to connect to the VPN server.

Download the VPN client configuration files from the Algo VPN setup and import them into your VPN client.

For WireGuard, you can use the wg-quick tool to connect:

```
sudo wg-quick up /path/to/your/configuration.conf
```

For IPsec, follow the instructions specific to your operating system to import the configuration and connect.

## 9. Adding new VPN users

- Update the users list in your config.cfg.

```

vim config.cfg
users:
- laptop
- desktop
- sdnog

```

- Sara
- Nishal
- Manhal
- Hafiz

- Open a terminal, cd to the algo directory, and activate the virtual environment with :

```
source .env/bin/activate
```

Run the command and it will require password , us the output password from step 8

```
./algo update-user
```

After this process completes, the Algo VPN server will contain only the users listed in the config.cfg file.

## Troubleshooting

- If you encounter issues during installation or configuration:

```
cd algo/  
sudo rm -rf /etc/wireguard/*  
rm -rf configs/*
```

Then immediately re-run ./algo.

- Check the Algo VPN documentation for troubleshooting tips.
- Ensure that your firewall rules allow VPN traffic.
- Verify that your VPN client is correctly configured.

## Conclusion

By following these steps, you should have a functioning Algo VPN setup on your local Ubuntu server, providing secure access to the sdnog infrastructure. For more advanced configurations and additional features, refer to the Algo VPN GitHub repository.

---

Revision #3

Created 30 November 2024 10:58:18 by sara

Updated 17 May 2025 17:12:59 by sara