Setup WireGuard VPN in Google Cloud Platform

dhanangw · Follow
5 min read · Nov 18, 2020

WireGuard VPN running on Google Compute Engine (GCE) Virtual Machine (VM) with Ubuntu 20.04 LTS. We're going to put our VPN server in either one of these Google Cloud Platform regions because it's free:

- Oregon: us-west1
- Iowa: us-central1
- South Carolina: us-east1

GCE free tier specification is good enough for our VPN server.

Part 1: Set up GCP Networking

Open a port through Google Cloud Platform's firewall

- On GCP console, click the hamburger menu on the top-left corner, click Networks -> VPC Network -> Firewall -> Create Firewall.
- Create a name for our firewall rule.
- in "Direction of traffic" choose"ingress".
- in "Targets" choose "All instances in the network".
- in "Source filter" choose"IP Ranges".
- in "IP Range" fill "0.0.0.0/0" (allow all traffic).
- in "Protocols and ports" click "Specified protocol and ports".

- Tick “udp”, enter port number “51820”.
- click “Create”.

Create external static IP address for VM

- in Networks menu, click VPC Network -> External IP Address -> Reserve Static Address.
- Create a name for IP address.
- in “Network Service Tier” choose “Premium”.
- in “IP version” select IPv4.
- in “Type” select “Regional”.
- Select your preferred Google Cloud Platform Region to run VPN server in.
- click “Reserve”.

Part 2: Set up GCP Compute Engine VM instance

- in the top-left hamburger menu click “Compute Engine”, then click “VM Instances” -> “Create Instance”.
- Choose “New VM Instance” option in the left sidebar.
- Create a name for our VM (we’ll use: “wireguard”).
- Choose your preferred Region to run VPN server in, the Zone does not matter.
- in “Machine family” choose General Purpose.
- in “Series” choose N1.
- in “Machine type” choose “f1-micro”.
- in “Boot disk”, choose “public image”.
- under “Operating system” choose “Ubuntu”.
- under “version” choose “20.04 LTS”.

- click “Management, security, disks, networking, sole tenancy”
- click Networking tab -> default network interface.
- in External IP, select the IP addressed you’ve created in Part 1.
- check IP forwarding.
- click Create.

Part 3: Set up WireGuard Server

- After you’ve created a new VM instance, you should be navigated back to list of GCE instances page. Wait for the status of your newly created VM instance to turn green.
- Under “Connect” column, click “SSH”. A new browser window with a SSH terminal should appear. The following steps is going to be done in the SSH terminal.
- Update packages, enter Y if prompted.
  ```
  sudo apt update && sudo apt upgrade
  ```
- Check whether your machine needs, a reboot.
  ```
  cat /var/run/reboot-required
  ```
  if it returns `*** System restart required ***`, reboot your machine
  ```
  sudo reboot
  ```
- Turn on IP forwarding for IPv4.
  edit `/etc/sysctl.conf` file, uncomment this line:
  ```
  net.ipv4.ip_forward=1
  ```
  then apply the changes.
  ```
  sudo sysctl -p
  ```
- Install WireGuard, enter Y if prompted.
  ```
  sudo apt install wireguard
  ```
- Generate server keys with this chained commands:
  ```
  sudo mkdir -p /etc/wireguard/keys; wg genkey | sudo tee
  ```

```
/etc/wireguard/keys/server.key | wg pubkey | sudo tee
/etc/wireguard/keys/server.key.pub
```

That command will create a new directory `/etc/wireguard/keys`, generate and save server's private key in `mobile.key`, and finally generate and save server's public key in `mobile.key.pub`

We're going to need those keys further along in the setup. Use this command to see your server's private key:

```
cat /etc/wireguard/keys/server.key
```

DO NOT show your server's private key to anyone!

- Check your default network interface.

  ```
  ip -o -4 route show to default | awk '{print $5}'
  ```

  take note of the command result.

- Configure WireGuard interface. Create a new file `/etc/wireguard/wg0.conf`. Fill it with this below. Replace `<YOUR_NETWORK_INTERFACE>` with your server's network interface.

  Replace `<YOUR_SERVER_PRIVATE_KEY>` with your server's private key:

```
[Interface]
Address = 10.0.0.1/24
ListenPort = 51820
PrivateKey = <YOUR_SERVER_PRIVATE_KEY>
PostUp = iptables -A FORWARD -i %i -j ACCEPT; iptables -t nat -A
POSTROUTING -o <YOUR_NETWORK_INTERFACE> -j MASQUERADE
PostDown = iptables -D FORWARD -i %i -j ACCEPT; iptables -t nat -D
POSTROUTING -o <YOUR_NETWORK_INTERFACE> -j MASQUERADE
SaveConfig = true
```

- Set permission for `wg0.conf`, `server.key`, and `server.key.pub` to be accessible only by root.

  ```
  sudo chmod 600 /etc/wireguard/wg0.conf
  sudo chmod 600 /etc/wireguard/keys/server.key
  sudo chmod 600 /etc/wireguard/keys/server.key.pub
  ```

- Turn on `wg0` WireGuard interface:

  ```
  sudo wg-quick up wg0
  ```

  make sure wg0 already running, and it's public key is equal to content of

`server.key.pub` .

`sudo wg show wg0` .

- Set `wg0` to be started at boot

  `sudo systemctl enable wg-quick@wg0`

- Open WireGuard port through firewall

  `sudo ufw allow 51820/udp`

  open port for SSH as well

  `sudo ufw allow 22/tcp`

- Turn on firewall

  `sudo ufw enable`

- Check firewall status, make sure the port for WireGuard and SSH are opened.

  `sudo ufw status verbose`

- Set MTU size to 1360 due to limitation in Google Cloud Platform.

  `sudo ip link set dev wg0 mtu 1360`

Part 4: Set Up WireGuard Client

We're going to setup WireGuard client on Android by generating a QR code to be scanned by the client app.

- On the server, install `qrencode` .

  `sudo apt install qrencode`

- Run this chained commands.

  `sudo mkdir -p /etc/wireguard/clients; wg genkey | sudo tee /etc/wireguard/clients/mobile.key | wg pubkey | sudo tee /etc/wireguard/clients/mobile.key.pub`

  That command will create a new directory `/etc/wireguard/clients` , generate and save our client's private key in `mobile.key` , and finally generate and save our client's public key in `mobile.key.pub` .

- Create an interface for your WireGuard client. Createa new file `/etc/wireguard/mobile.conf` . Fill with this below. Replace `<CLIENT'S-PRIVATE-KEY>`

with content of `mobile.key`. Replace `<PRIVATE-IP-OF-WIREGUARD-SERVER>` to "10.0.0.1/24". Replace `<YOUR-SERVER'S-PUBLIC-KEY>` with your server's public key (check out `server.key.pub` or run `sudo wg show wg0`). Replace `<STATIC-IP-OF-GCP-INSTANCE>` with IP address you've created in Part 1.

```
[Interface]
PrivateKey = <CLIENT'S-PRIVATE-KEY>
Address = <PRIVATE-IP-OF-WIREGUARD-SERVER>/24
DNS = 1.1.1.1, 1.0.0.1
MTU = 1360

[Peer]
PublicKey = <YOUR-SERVER'S-PUBLIC-KEY>
AllowedIPs = 0.0.0.0/0
Endpoint = <STATIC-IP-OF-GCP-INSTANCE>:51820
```

- Download the official WireGuard Android app in your phone. It's this one.
- Add client to server's interface. Run this command:

  ```
  sudo wg set wg0 peer <YOUR_CLIENT_PUBLIC_KEY> allowed-ips <YOUR_CLIENT_VPN_IP>
  ```

  Replace `<YOUR_CLIENT_VPN_IP>` to 10.0.0.2/32

  Make sure your client already added to server.

  ```
  sudo wg show wg0
  ```

- **TEMPORARILY** make `/etc/wireguard/` accessible:

  ```
  sudo chmod 777 /etc/wireguard/
  sudo chmod 777 /etc/wireguard/clients/
  sudo chmod 777 /etc/wireguard/clients/mobile.conf
  ```

- Generate QR code to transfer client's interface config to your phone.

  ```
  qrencode -t ansiutf8 < /etc/wireguard/clients/mobile.conf
  ```

- Make `/etc/wireguard/` accessible to only root again.

  ```
  sudo chmod 600 /etc/wireguard/
  sudo chmod 600 /etc/wireguard/clients/
  sudo chmod 600 /etc/wireguard/clients/mobile.conf
  ```

  DO NOT FORGET TO DO THIS!

- Open WireGuard app on your phone, tap the floating plus("+") button on the bottom-right corner, and choose "SCAN FROM QR CODE".

- Scan the generated QR code, and give the tunnel a name.
- Connect to the tunnel, and check your ip (search this keyword on Google: “whatsmyip”). Your IP should be equal to the external IP you’ve created in Part 1.

You can find how to connect more types of client (e.g. Windows PC, MacOS, iOS) in Reference no 2.

References

1. Jacob Marble’s WireGuard server setup guide on GCP.
2. Jay Rogers’ WireGuard setup course in ServerSideUp.
3. howyay’s MTU solution in Github.
4. Limitations of MTU in Google Cloud Platform.

Wireguard | Google Cloud Platform | VPN