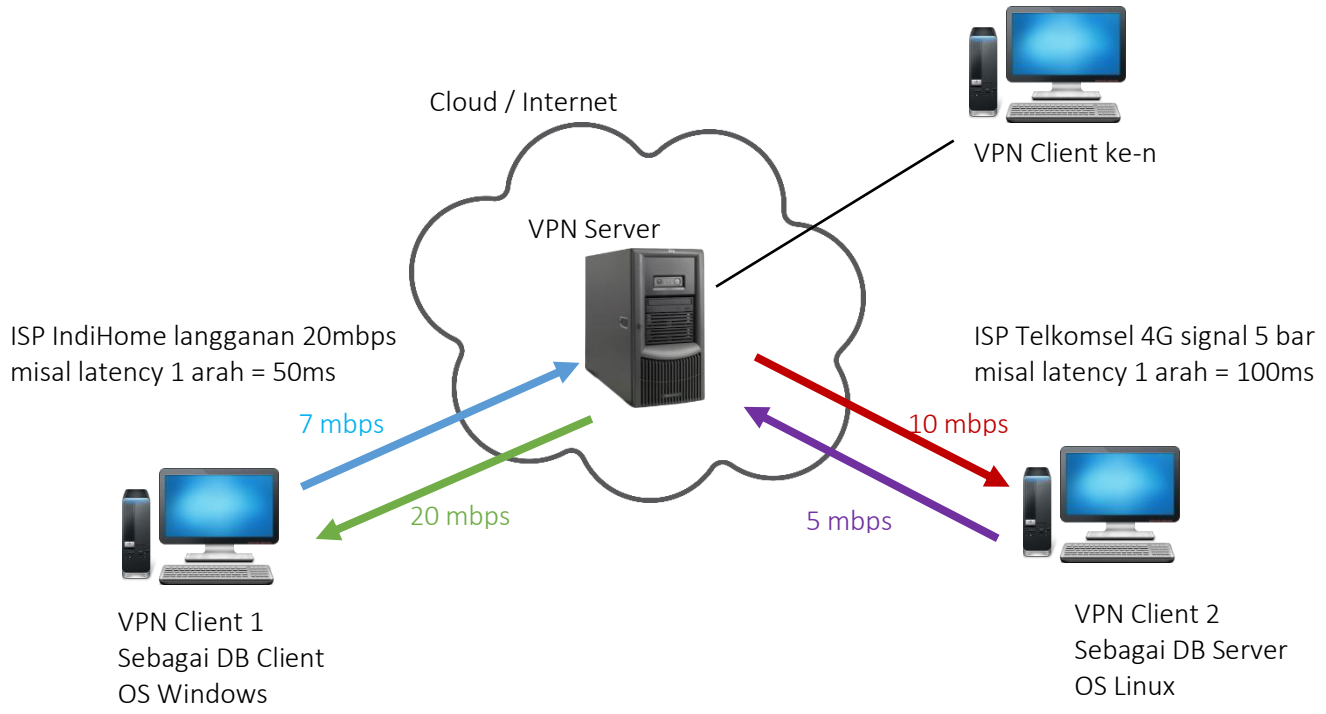
Tutorial VPN Client di Linux Debian-Based

Topology, Bandwidth & Latency



Topology

VPN Client 1 menghubungkan ke komputer VPN Client 2 untuk keperluan remote, DB, transfer file, dll.

Bandwidth

- Saat VPN Client 1 melakukan download file dari VPN Client 2, maka kecepatan bandwidth yang menjadi penentu adalah arah upstream VPN Client 2 yaitu 5 mbps. (karena 5 mbps terkecil terhadap 20 mbps)
- Saat VPN Client 1 melakukan upload file ke VPN Client 2, maka kecepatan bandwidth yang menjadi penentu adalah arah upstream VPN Client 1 yaitu 7 mbps. (karena 7 mbps terkecil terhadap 10 mbps)
- Kondisi bandwidth di atas juga dihitung saat koneksi sedang ideal, yaitu ISP sedang stabil, internet tidak digunakan untuk keperluan lain seperti buka YouTube, dll.

Latency

- Adalah waktu tempuh packet network mulai dari awal request sampai mendapatkan respon balik.
- Bila VPN Client 1 mencoba mengakses database di VPN Client 2, maka latency yang terjadi adalah: Total latency = 50 (biru) + 100 (merah) + 100 (ungu) + 50 (hijau) = 300 ms (mili detik).
- Latency untuk kebutuhan audio/video streaming harus di bawah < 200 ms agar tidak terjadi putus-putus suara dan gambar.
- Latency untuk transfer data non streaming tetap bisa dilakukan bahkan sampai 2.000 ms (2 detik) sekalipun tetapi akan berefek lamanya transfer file bila berukuran besar.

Install OpenVPN Client di OS Linux Debian (dan turunannya seperti Ubuntu, Kali Linux, dll)

- Cek dulu berapa interface yang ada di komputer VPN Client 2 ini

```
ifconfig
```

```
ens32: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.22.22.211  netmask 255.255.255.0  broadcast 172.22.22.255
        ether 00:0c:29:79:33:df  txqueuelen 1000  (Ethernet)

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        loop  txqueuelen 1000  (Local Loopback)
```

  - Saat ini hanya ada 2 interface: **`ens32`** dan **`lo`** (loopback)

- Cek dulu apakah komputer VPN Client 2 sudah ada MySQL jalan:

```
netstat -nlp
```

```
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address    Foreign Address  State    PID/Program name
tcp        0      0 127.0.0.1:5432   0.0.0.0:*        LISTEN   707/postgres
tcp        0      0 0.0.0.0:3306     0.0.0.0:*        LISTEN   229764/mariadbd
tcp6       0      0 :::2295          :::*             LISTEN   180297/sshd: /usr/s
tcp6       0      0 ::1:5432         :::*             LISTEN   707/postgres
```

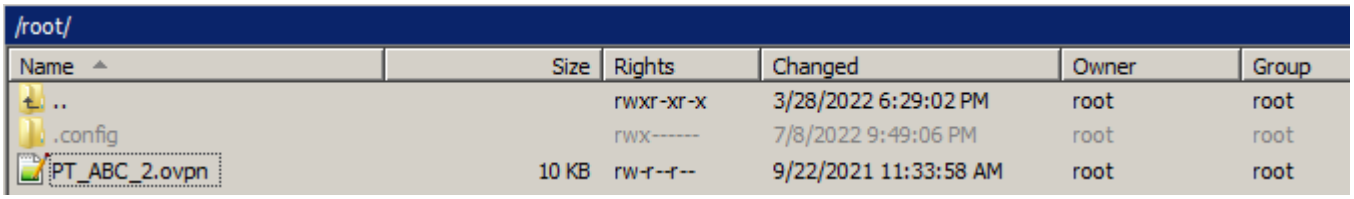
  - Terlihat MySQL / MariaDB sudah listen semua interface yaitu IP Address 0.0.0.0 di port 3306

    PENTING!!! Program SSH, MYSQL, dan lainnya yang ingin dihubungkan HARUS listening pada IP Address 0.0.0.0 agar bisa dinamis digunakan saat OpenVPN yang tersambung setelah program-program tersebut aktif, karena interface TUN milik VPN akan muncul setelah OpenVPN client tersambung.

- Install program OpenVPN Client bawaan Linux:

```
apt-get install openvpn
```

- Copy file OpenVPN Client Profile ke komputer VPN Client 2.



  - Di contoh ini di upload pakai program WinSCP ke folder **`/root`** (sangat disarankan agar secure)

- Lalu menjalankan OpenVPN client adalah

```
/usr/sbin/openvpn --daemon --config /root/PT_ABC_2.ovpn
```

  - Note: menjalankan command ini berulang kali ternyata tidak masalah. Yang terjadi program OpenVPN hanya akan memutus koneksi VPN yang ada, lalu menyambungkan kembali.

- Agar OpenVPN Client langsung tersambung saat OS Linux booting adalah dengan menggunakan CRON.

Harus login sebagai root, buka CRON untuk user root, dengan menjalankan command:

```
crontab -e
```

Lalu copy paste command berikut di akhir baris:

```
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
@reboot /usr/sbin/openvpn --daemon --config /root/PT_ABC_2.ovpn
```

```
@reboot /usr/sbin/openvpn --daemon --config /root/PT_ABC_2.ovpn
```

Jangan lupa save file CRON yang sudah di edit ini. Bila pakai program nano, tekan CTRL+O lalu ENTER

- Buktikan bahwa OpenVPN client sudah tersambung.

```
ifconfig
```

```
ens32: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.22.22.211  netmask 255.255.255.0  broadcast 172.22.22.255
        ether 00:0c:29:79:33:df  txqueuelen 1000  (Ethernet)

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        loop  txqueuelen 1000  (Local Loopback)

tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  mtu 1500
        inet 10.8.0.2  netmask 255.255.255.0  destination 10.8.0.3
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 500  (UNSPEC)
        RX packets 700  bytes 42000 (41.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
```

  - Bukti OpenVPN Client sudah tersambung HARUS ADA interface TUN (tun0, tun1, dst).
  - Di contoh ini dengan IP address VPN Client 10.8.0.2

- Tes Reachability pakai ping antara VPN Client – pastikan firewall sedang off untuk protocol ICMP

Saat ini kami sudah coba melakukan simulasi komputer VPN Client 2 terputus dengan cara:

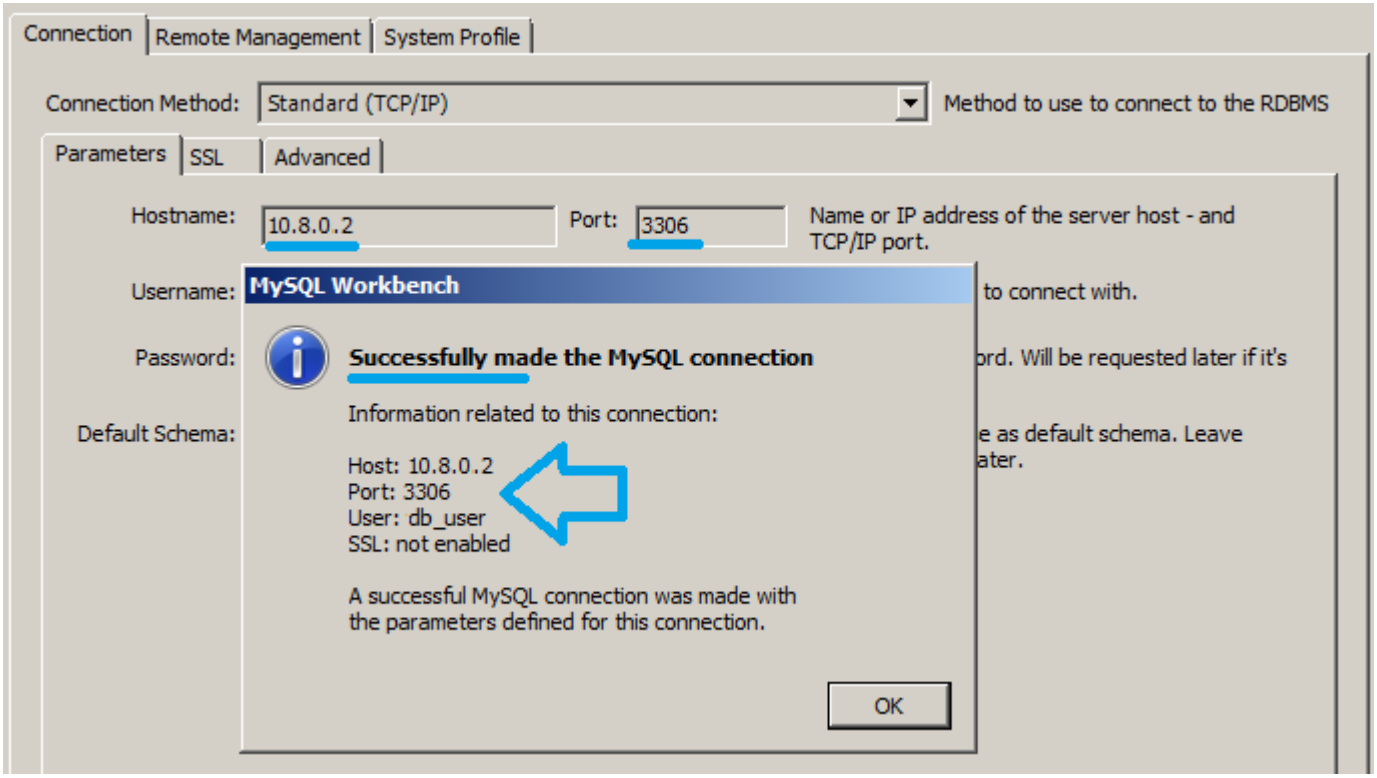
  - Reboot komputer
  - Memutus jaringan ISP sementara
  - Memutuskan koneksi WiFi
  - Men-disable Network Adapter
  - Mencabut kabel LAN

Lalu setelah disambungkan kembali, maka OpenVPN Client tersambung otomatis maksimal dalam 30 detik (biasanya lebih cepat).

```
Request timed out.
Request timed out.
Reply from 10.8.0.2: bytes=32 time=42ms TTL=64
Reply from 10.8.0.2: bytes=32 time=75ms TTL=64
```

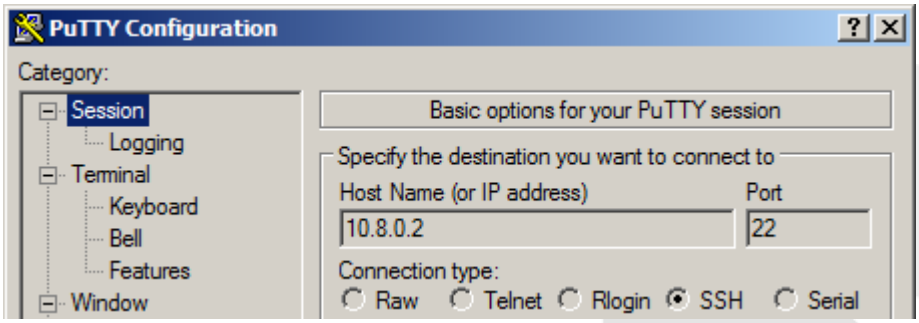
Ping dari VPN Client 1 ke VPN Client 2, tersambung kembali secara otomatis.

- Tes Koneksi DB MySQL – pastikan firewall sedang off untuk protocol TCP port 3306



VPN IP Address 10.8.0.2 port 3306 terbukti tersambung menggunakan MySQL Workbench

- Tes Koneksi SSH – pastikan firewall sedang off untuk protocol TCP port 22



IP VPN Client 2 adalah 10.8.0.2