Process

1. **Install Nebula in WSL2**

   Command: *sudo snap nebula*

```
rosalind@Rosalind:/mnt/c/Users/陈芷薇$ sudo snap install nebula
2025-10-16T10:37:26+08:00 INFO Waiting for automatic snapd restart...
nebula v1.9.7 from Will Sinatra (durrendal) installed
```

2. **A Nebula certificate authority**

   It create files named ca.key and ca.cert in the current directory. The ca.key file is the key used to sign the certificates for individual nebula nodes/hosts.

   Command: ./nebula-cert ca -name “chenzhiwei”

```
rosalind@Rosalind:/mnt/e/nebula-linux-386$ ./nebula-cert ca -name "chenzhiwei"
rosalind@Rosalind:/mnt/e/nebula-linux-386$ ls
ca.crt  ca.key  nebula  nebula-cert
```

3. **Generate Nebula host keys and certificates from that certificate authority**

   Command: ./nebula-cert sign -name "lighthouse1" -ip "192.168.100.1/24"

   ./nebula-cert sign -name "MBP15" -ip "192.168.100.15/24"

   ./nebula-cert sign -name "mini" -ip "192.168.100.10/24"

   This assumes I have three nodes, named lighthouse1, MBP15 and mini, followed with the chosen IP addresses and the associated subnet.

```
rosalind@Rosalind:/mnt/e/nebula-linux-386$ ./nebula-cert sign -name "lighthouse" -ip "192.168.100.1/24"
rosalind@Rosalind:/mnt/e/nebula-linux-386$ ./nebula-cert sign -name "MBP15" -ip "192.168.100.15/24"
rosalind@Rosalind:/mnt/e/nebula-linux-386$ ./nebula-cert sign -name "mini" -ip "192.168.100.10/24"
```

4. **Run nebula on each host**

```
rosalind@Rosalind:/mnt/e/nebula-linux-386$ sudo ./nebula -config /mnt/e/nebula-linux-386/confih.yaml
INFO[0000] Firewall rule added                           firewallRule="map[caName: caSha: direction:outgoi
ng endPort:0 groups:[] host:any ip: proto:0 startPort:0]"
INFO[0000] Firewall rule added                           firewallRule="map[caName: caSha: direction:incomi
ng endPort:0 groups:[] host:any ip: proto:0 startPort:0]"
INFO[0000] Firewall started                              firewallHash=21716b47a7a140e448077fe66c31b4b42f23
2e996818d7dd1c6c4991e066dbdb
INFO[0000] Main HostMap created                          network=192.168.100.1/24 preferredRanges="[]"
INFO[0000] UDP hole punching enabled
INFO[0000] Nebula interface is active                    build=1.4.0 interface=nebula1 network=192.168.100
.1/24 udpAddr="[::]:4242"
INFO[0221] Handshake message received                    certName=MBP15 fingerprint=9ff84564a6699a4396a66b
0162a776bd8d61417ba94119c583dcb0762f171ce6 handshake="map[stage:1 style:ix_psk0]" initiatorIndex=161550090
5 issuer=46caa1e2ef880dfb1b9b087653c15a2aa3ef9558e095b093f053541c20f1fc77 remoteIndex=0 responderIndex=0 u
dpAddr="127.0.0.1:4243" vpnIp=192.168.100.15
INFO[0221] Handshake message sent                        certName=MBP15 fingerprint=9ff84564a6699a4396a66b
0162a776bd8d61417ba94119c583dcb0762f171ce6 handshake="map[stage:2 style:ix_psk0]" initiatorIndex=161550090
5 issuer=46caa1e2ef880dfb1b9b087653c15a2aa3ef9558e095b093f053541c20f1fc77 remoteIndex=0 responderIndex=279
8152680 sentCachedPackets=0 udpAddr="127.0.0.1:4243" vpnIp=192.168.100.15
```

```
rosalind@Rosalind:/mnt/e/nebula-linux-386$ sudo ./nebula -config /mnt/e/nebula-linux-386/config.yaml
INFO[0000] Firewall rule added                           firewallRule="map[caName: caSha: direction:outgoing endPo
rt:0 groups:[] host:any ip: proto:0 startPort:0]"
INFO[0000] Firewall rule added                           firewallRule="map[caName: caSha: direction:incoming endPo
rt:0 groups:[] host:any ip: proto:0 startPort:0]"
INFO[0000] Firewall started                              firewallHash=21716b47a7a140e448077fe66c31b4b42f232e996818
d7dd1c6c4991e066dbdb
INFO[0000] Main HostMap created                          network=192.168.100.15/24 preferredRanges="[]"
INFO[0000] UDP hole punching enabled
INFO[0000] Nebula interface is active                    build=1.4.0 interface=nebula2 network=192.168.100.15/24 u
dpAddr="[::]:4243"
INFO[0000] Handshake message sent                        handshake="map[stage:1 style:ix_psk0]" initiatorIndex=161
5500905 udpAddrs="[127.0.0.1:4242]" vpnIp=192.168.100.1
INFO[0000] Handshake message received                    certName=lighthouse durationNs=8366658 fingerprint=921082
c069988bbac1c84ba5c9d20886707b4d110cd40310ed18b1fb9b09da48 handshake="map[stage:2 style:ix_psk0]" initiatorIndex=1
615500905 issuer=46caa1e2ef880dfb1b9b087653c15a2aa3ef9558e095b093f053541c20f1fc77 remoteIndex=1615500905 responder
Index=2798152680 sentCachedPackets=1 udpAddr="127.0.0.1:4242" vpnIp=192.168.100.1
INFO[0256] Close tunnel received, tearing down.          certName=lighthouse udpAddr="127.0.0.1:4242" vpnIp=192.16
8.100.1
```

5. **Ping from one host to another one**

```
rosalind@Rosalind:~$ ping 192.168.100.1
PING 192.168.100.1 (192.168.100.1) 56(84) bytes of data.
64 bytes from 192.168.100.1: icmp_seq=1 ttl=64 time=0.039 ms
64 bytes from 192.168.100.1: icmp_seq=2 ttl=64 time=0.076 ms
64 bytes from 192.168.100.1: icmp_seq=3 ttl=64 time=0.075 ms
64 bytes from 192.168.100.1: icmp_seq=4 ttl=64 time=0.073 ms
64 bytes from 192.168.100.1: icmp_seq=5 ttl=64 time=0.069 ms
64 bytes from 192.168.100.1: icmp_seq=6 ttl=64 time=0.087 ms
64 bytes from 192.168.100.1: icmp_seq=7 ttl=64 time=0.082 ms
64 bytes from 192.168.100.1: icmp_seq=8 ttl=64 time=0.091 ms
64 bytes from 192.168.100.1: icmp_seq=9 ttl=64 time=0.080 ms
64 bytes from 192.168.100.1: icmp_seq=10 ttl=64 time=0.224 ms
64 bytes from 192.168.100.1: icmp_seq=11 ttl=64 time=0.068 ms
64 bytes from 192.168.100.1: icmp_seq=12 ttl=64 time=0.084 ms
64 bytes from 192.168.100.1: icmp_seq=13 ttl=64 time=0.072 ms
64 bytes from 192.168.100.1: icmp_seq=14 ttl=64 time=0.067 ms
64 bytes from 192.168.100.1: icmp_seq=15 ttl=64 time=0.090 ms
64 bytes from 192.168.100.1: icmp_seq=16 ttl=64 time=0.075 ms
64 bytes from 192.168.100.1: icmp_seq=17 ttl=64 time=0.076 ms
64 bytes from 192.168.100.1: icmp_seq=18 ttl=64 time=0.080 ms
64 bytes from 192.168.100.1: icmp_seq=19 ttl=64 time=0.065 ms
64 bytes from 192.168.100.1: icmp_seq=20 ttl=64 time=0.076 ms
64 bytes from 192.168.100.1: icmp_seq=21 ttl=64 time=0.074 ms
64 bytes from 192.168.100.1: icmp_seq=22 ttl=64 time=0.081 ms
64 bytes from 192.168.100.1: icmp_seq=23 ttl=64 time=0.070 ms
```

Problems that I met

1. This image indicates that my Lab.2.lighthouse.config.yaml file has syntax errors in its YAML format, especially issues with the colon, indentation, or quotation marks near line 8.

```
rosalind@Rosalind:/mnt/e/nebula-linux-386$ ./nebula -config /mnt/e/nebula-linux-386/Lab.2.lighthouse.config.yaml
failed to load config: yaml: line 8: mapping values are not allowed in this contextrosalind@Rosalind:/mnrrosalind
@Rosalind:/mnt/e/nebula-linux-386$
```

Solution:

The modified conflh.yaml

```
pki:
  ca: /mnt/e/nebula-linux-386/ca.crt
  cert: /mnt/e/nebula-linux-386/lighthouse.crt
  key: /mnt/e/nebula-linux-386/lighthouse.key

lighthouse:
  am_lighthouse: true
  interval: 60

listen:
  host: "[::]"
  port: 4242

punchy:
```

```
  punch: true

cipher: chachapoly

tun:
  disabled: false
  dev: nebula1
  drop_local_broadcast: false
  drop_multicast: false
  tx_queue: 500
  mtu: 1300

logging:
  level: info
  format: text

firewall:
  conntrack:
    tcp_timeout: 12m
    udp_timeout: 3m
    default_timeout: 10m
    max_connections: 100000

  outbound:
    - port: any
      proto: any
      host: any

  inbound:
    - port: any
      proto: any
      host: any
```

2. This indicates that Nebula failed to create the virtual network interface (TUN/TAP) during startup. TUN/TAP is a virtual network interface at the operating system kernel level; ordinary users do not have permission to create it.
   Therefore, start Nebula with **sudo** instead.

3. Lighthouse has already started successfully and is listening on port 4242 normally. I am currently running two Nebula instances (Lighthouse + Node) on the same machine, both of which are trying to create a network interface with the same name, nebula1, causing a resource conflict.

Solution:

Change the dev in

“

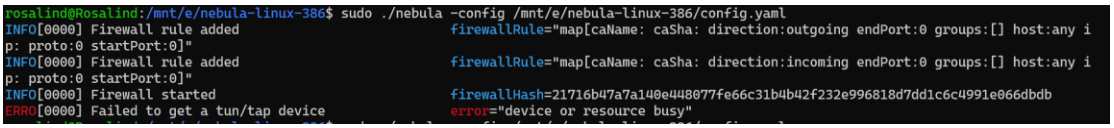
tun:

disabled: false

dev: nebula1

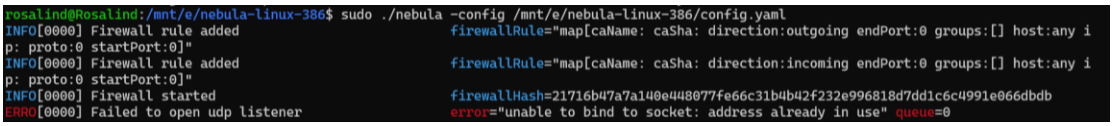
“

to nebula2



4. Each Nebula node needs to bind to a UDP port for communication (4242 by default). A Lighthouse instance is already running on the same machine, occupying UDP port 4242. Therefore, when a second instance (config.yaml) also tries to listen on the same port, a conflict occurs. This is because a UDP port cannot be bound to by two programs simultaneously on the same IP address.

   Solution: change the listening port of config to 4243.



5. My node (192.168.100.15) wants to contact Lighthouse (192.168.100.1), but I don't know its real IP/port because static_host_map is not configured.

   Solution: The static_host_map was modified to use 127.0.0.1 because it runs on the same host.