Lab 2 Report

Course: cloud_computing

Student: ZhengYang

GitHub Repo: NPC00101/Cloud-Comupting-Labs: The repository is created to store my courses lab solutions

Date: 2025.9.16

1. Components and Tools .

Public IP: "192.168.100.1": ["104.243.20.247:554"]

Personal Computer (Node): laptop

Software: Nebula v1.8.1

Configuration Files:

ca.crt : Certificate Authority provided by lecturer)

ZhengYang.crt(Client certificate)

ZhengYang.key(Private key)

Tools:SSH, Terminal, Text Editor (for config comparison)

2. Procedures

Step 1: Nebula Installation

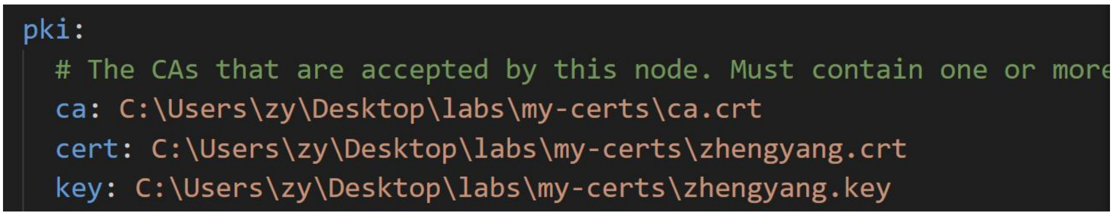
Nebula was installed on the personal laptop through github (Release Release v1.9.6 · slackhq/nebula)

Step 2: Configuration File Setup

The provided `config.yaml`was customized for the node. The key changes involved:

Ensuring the correct paths to the certificate and key files were specified under `pki`.

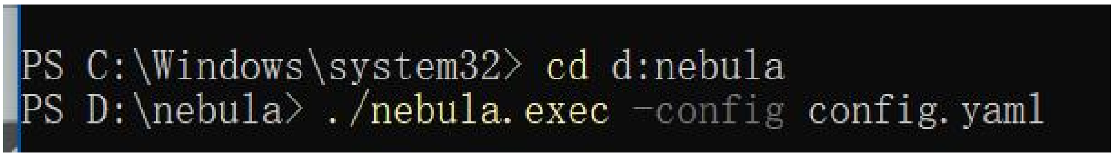
Screenshot:



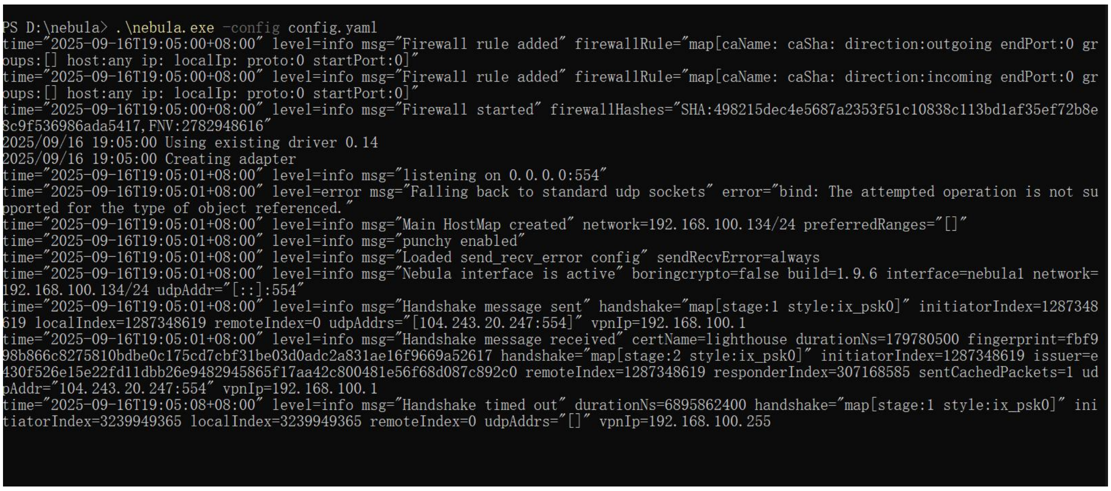
```
pki:
  # The CAs that are accepted by this node. Must contain one or more
  ca: C:\Users\zy\Desktop\labs\my-certs\ca.crt
  cert: C:\Users\zy\Desktop\labs\my-certs\zhengyang.crt
  key: C:\Users\zy\Desktop\labs\my-certs\zhengyang.key
```

Step 3: Running the Nebula Service

```
.\nebula.exe -config config.yaml
```



```
PS C:\Windows\system32> cd d:nebula
PS D:\nebula> ./nebula.exec -config config.yaml
```

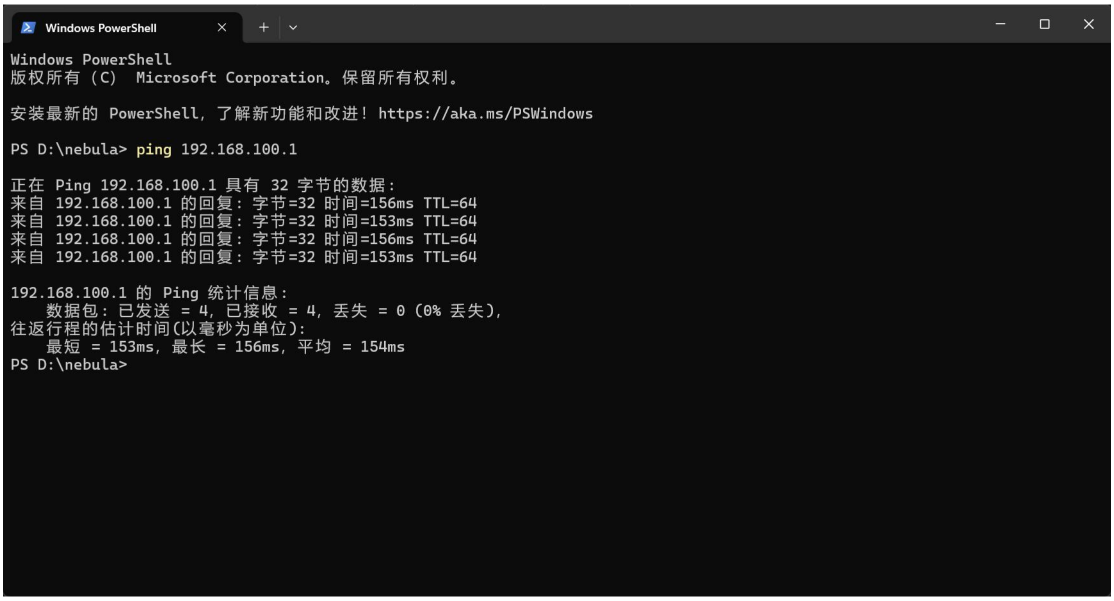


```
PS D:\nebula> .\nebula.exe -config config.yaml
time="2025-09-16T19:05:00+08:00" level=info msg="Firewall rule added" firewallRule="map[caName: caSha: direction:outgoing endPort:0 gr
oups:[] host:any ip: localIp: proto:0 startPort:0]"
time="2025-09-16T19:05:00+08:00" level=info msg="Firewall rule added" firewallRule="map[caName: caSha: direction:incoming endPort:0 gr
oups:[] host:any ip: localIp: proto:0 startPort:0]"
time="2025-09-16T19:05:00+08:00" level=info msg="Firewall started" firewallHashes="SHA:498215dec4e5687a2353f51c10838c113bd1af35ef72b8e
8c9f536986ada5417,FNV:2782948616"
2025/09/16 19:05:00 Using existing driver 0.14
2025/09/16 19:05:00 Creating adapter
time="2025-09-16T19:05:01+08:00" level=info msg="listening on 0.0.0.0:554"
time="2025-09-16T19:05:01+08:00" level=error msg="Falling back to standard udp sockets" error="bind: The attempted operation is not su
pported for the type of object referenced."
time="2025-09-16T19:05:01+08:00" level=info msg="Main HostMap created" network=192.168.100.134/24 preferredRanges="[]"
time="2025-09-16T19:05:01+08:00" level=info msg="punchy enabled"
time="2025-09-16T19:05:01+08:00" level=info msg="Loaded send_recv_error config" sendRecvError=always
time="2025-09-16T19:05:01+08:00" level=info msg="Nebula interface is active" boringcrypto=false build=1.9.6 interface=nebula1 network=
192.168.100.134/24 udpAddr="[::]:554"
time="2025-09-16T19:05:01+08:00" level=info msg="Handshake message sent" handshake="map[stage:1 style:ix_psk0]" initiatorIndex=1287348
619 localIndex=1287348619 remoteIndex=0 udpAddrs="[104.243.20.247:554]" vpnIp=192.168.100.1
time="2025-09-16T19:05:01+08:00" level=info msg="Handshake message received" certName=lighthouse durationNs=179780500 fingerprint=fbf9
98b866c8275810bdbe0c175cd7cbf31be03d0adc2a831ae16f9669a52617 handshake="map[stage:2 style:ix_psk0]" initiatorIndex=1287348619 issuer=e
430f526e15e22fd11dbb26e9482945865f17aa42c800481e56f68d087c892c0 remoteIndex=1287348619 responderIndex=307168585 sentCachedPackets=1 ud
pAddr="104.243.20.247:554" vpnIp=192.168.100.1
time="2025-09-16T19:05:08+08:00" level=info msg="Handshake timed out" durationNs=6895862400 handshake="map[stage:1 style:ix_psk0]" ini
tiatorIndex=3239949365 localIndex=3239949365 remoteIndex=0 udpAddrs="[]" vpnIp=192.168.100.255
```

Step 4: Testing Connectivity

With both services running, connectivity was tested by pinging the lighthouse's overlay IP address from the personal node.

```
ping 192.168.100.1
```

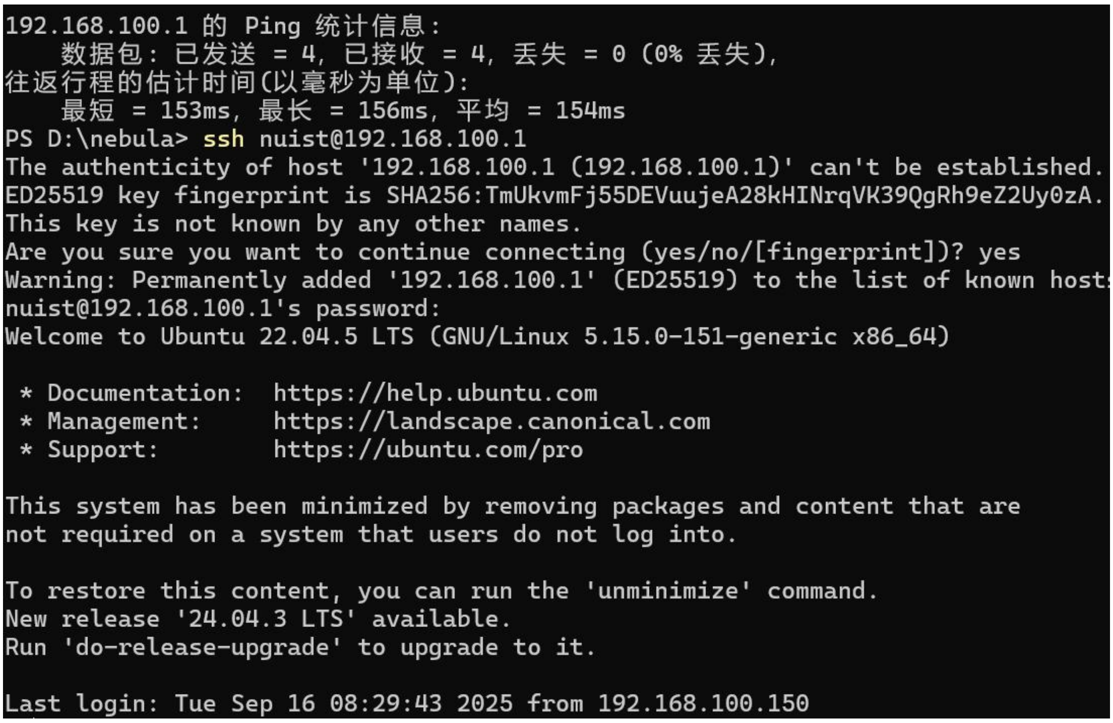


```
Windows PowerShell
版权所有 (C)  Microsoft Corporation。保留所有权利。

安装最新的 PowerShell，了解新功能和改进！https://aka.ms/PSWindows

PS D:\nebula> ping 192.168.100.1

正在 Ping 192.168.100.1 具有 32 字节的数据:
来自 192.168.100.1 的回复: 字节=32 时间=156ms TTL=64
来自 192.168.100.1 的回复: 字节=32 时间=153ms TTL=64
来自 192.168.100.1 的回复: 字节=32 时间=156ms TTL=64
来自 192.168.100.1 的回复: 字节=32 时间=153ms TTL=64

192.168.100.1 的 Ping 统计信息:
    数据包: 已发送 = 4，已接收 = 4，丢失 = 0 (0% 丢失)，
往返行程的估计时间(以毫秒为单位):
    最短 = 153ms，最长 = 156ms，平均 = 154ms
PS D:\nebula>
```

Step 5: Establishing SSH Connection to Lighthouse

A SSH connection was established to the lighthouse node **through the Nebula overlay network** using its overlay IP. This proves that higher-layer protocols work seamlessly over the established tunnel.

```
ssh nuist@192.168.100.1
```



```
192.168.100.1 的 Ping 统计信息:
    数据包: 已发送 = 4，已接收 = 4，丢失 = 0 (0% 丢失)，
往返行程的估计时间(以毫秒为单位):
    最短 = 153ms，最长 = 156ms，平均 = 154ms
PS D:\nebula> ssh nuist@192.168.100.1
The authenticity of host '192.168.100.1 (192.168.100.1)' can't be established.
ED25519 key fingerprint is SHA256:TmUkvmFj55DEVuujeA28kHINrqVK39QgRh9eZ2Uy0zA.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.100.1' (ED25519) to the list of known host
nuist@192.168.100.1's password:
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 5.15.0-151-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
New release '24.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Tue Sep 16 08:29:43 2025 from 192.168.100.150
```

3. Conclusion

This lab was successfully completed by deploying a Nebula lighthouse on a cloud VM and connecting a client node from behind a NAT. The key takeaways are:

Lighthouse Role: The lighthouse, with its public IP, is crucial for initial peer discovery and coordination, enabling direct P2P connections (NAT traversal) where possible

Certificate-Based Security: Nebula relies heavily on a PKI for node authentication, ensuring only authorized devices join the network.

Operational Simplicity: Once configured, the network operates transparently, providing a secure, virtual LAN layer over the public internet.

The successful ping and SSH connection confirm that the overlay network is functioning correctly, providing secure and reliable connectivity between the nodes.