Author: Piotr Artman

Student No: C00220223

Course: Cloud Data Centres

Content: Lab 2 – notes and screenshots from creating a Nebula network

Nebula zero-trust network

A Certificate Authority

Certificate Authority can be set in any device with Nebula software installed. This software is necessary to create digital signatures used in an authentication process within Nebula mesh network.

Step by step guide

Follow quick start tutorial available here [1].

Instance in Google Cloud Platform

Creating an instance.

In order to avail a Free Tier in Google Cloud Platform it is necessary to select machine with following parameters[2]:

Machine type: **e2-micro VM instance**

Geographic regions: **Oregon: us-west1, Iowa: us-central1, South Carolina: us-east1**

Disk size: **30 GB-months standard persistent disk.**

Network traffic: **1 GB of outbound data transfer from North America to all region destinations (excluding China and Australia) per month.**

Note: **A free usage of e2-micro instances is based on total hours used across all instances.**

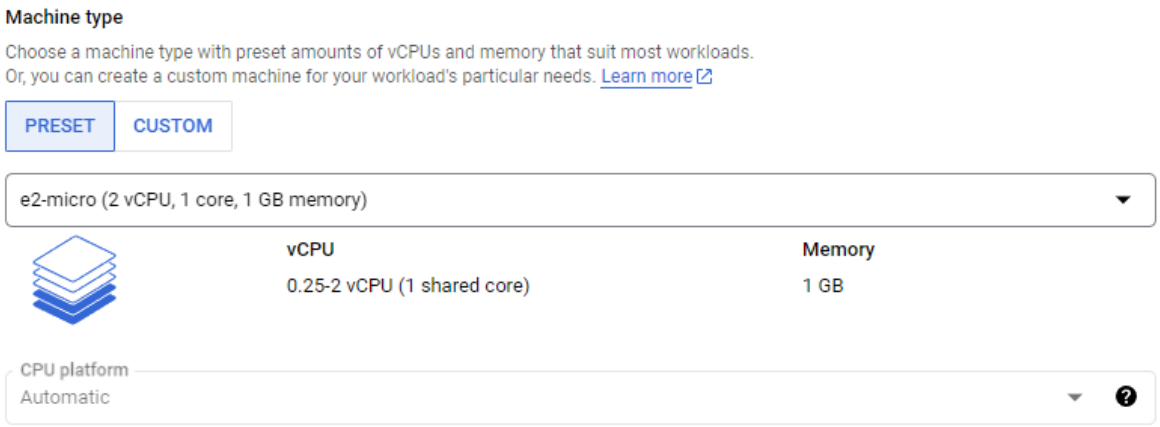

Fig. 1. Step 1 - select machine type.

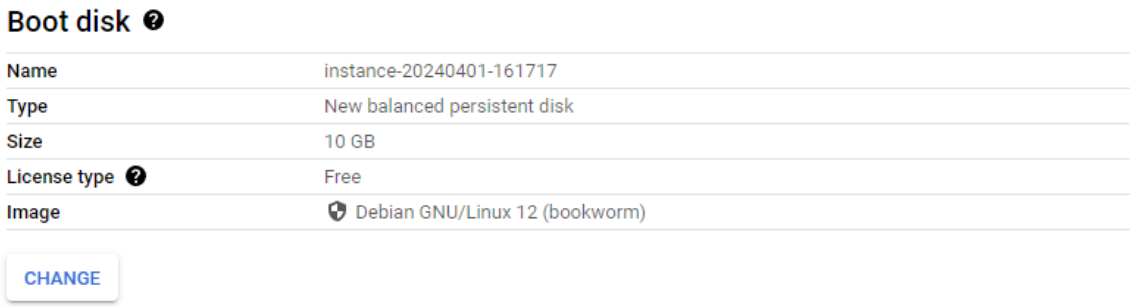

Fig. 2. Step 2 - select boot disk.

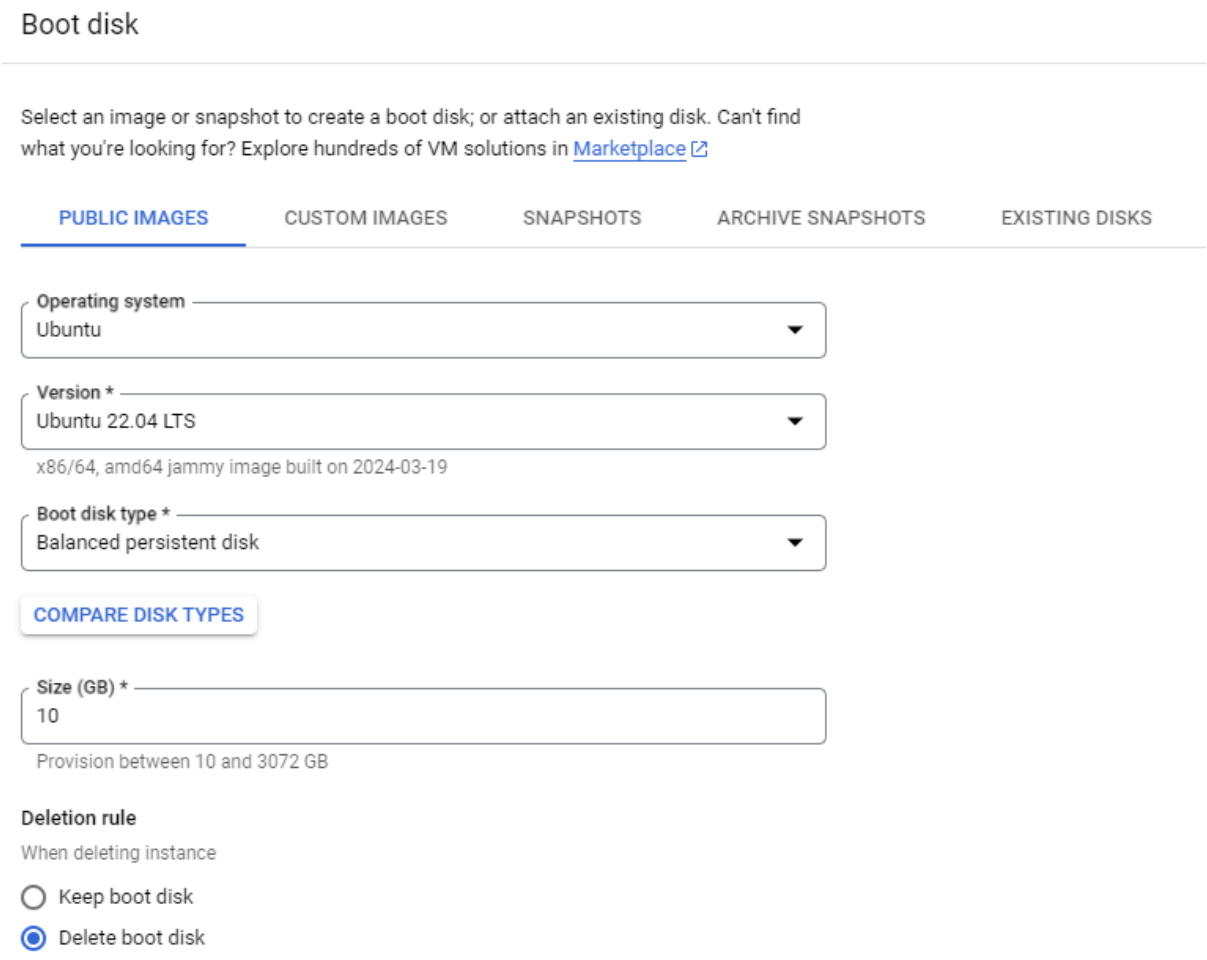

Fig. 3. Step 3 - select boot disk type.

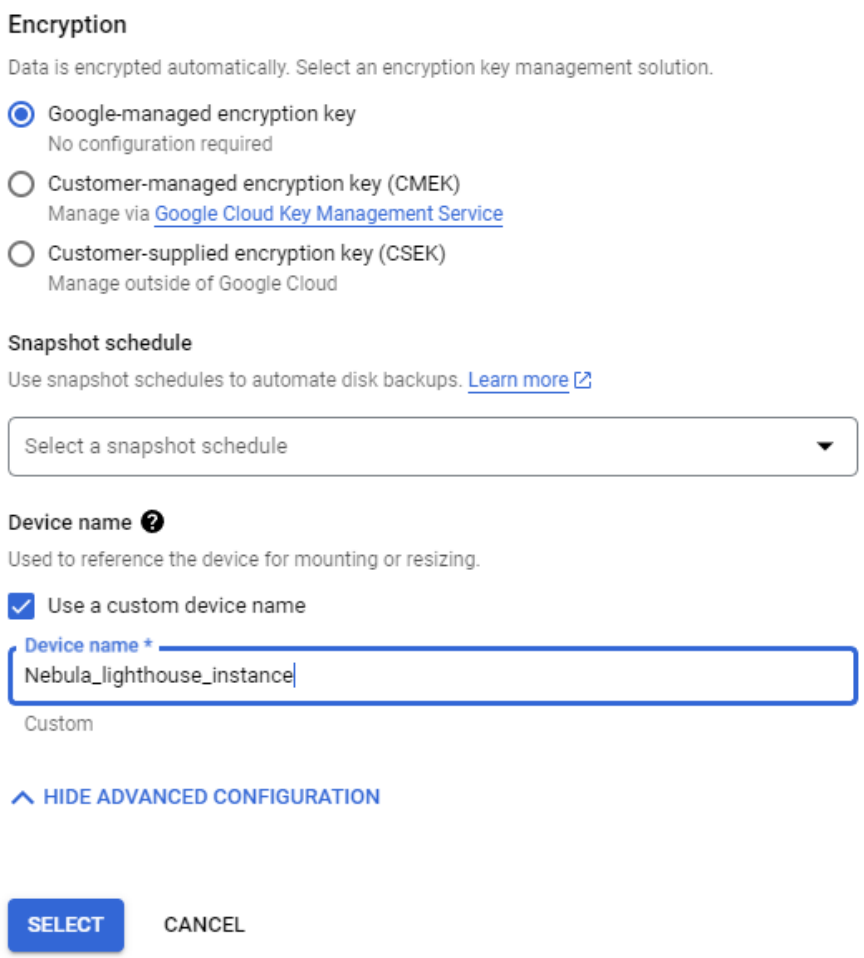

Fig. 4. Step 4 - change of device (instance) name.

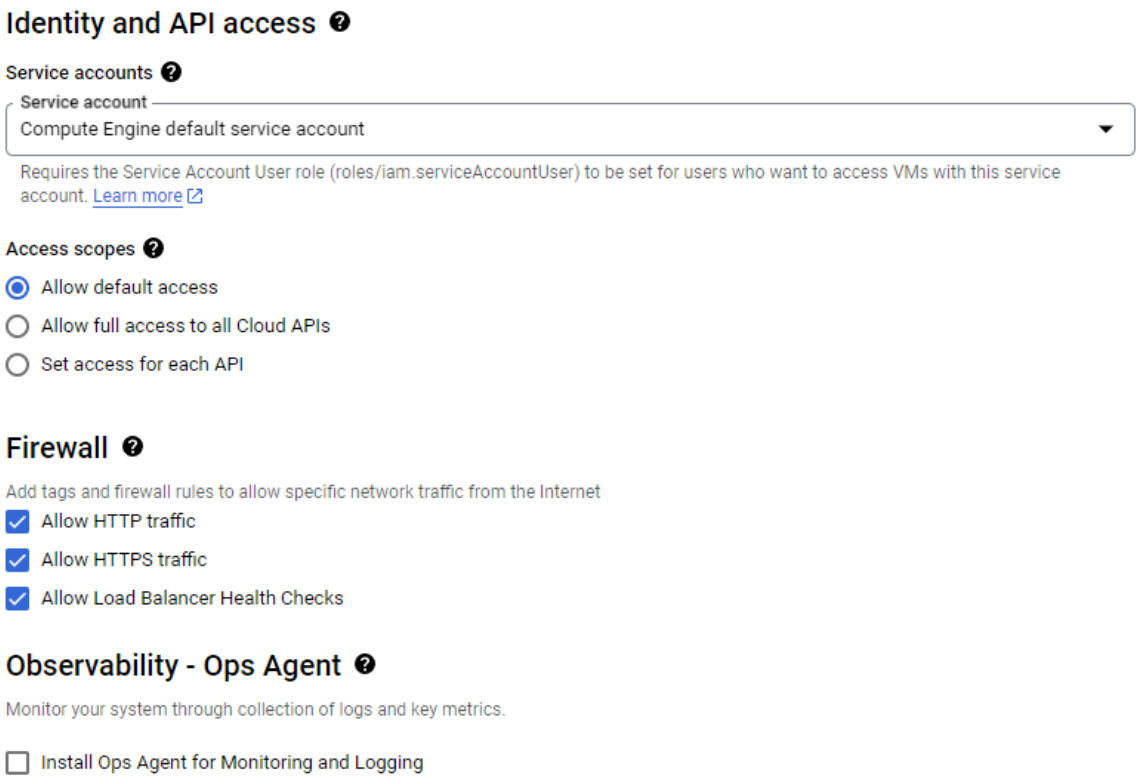

Fig. 5. Step 5 - select firewall and API related options.

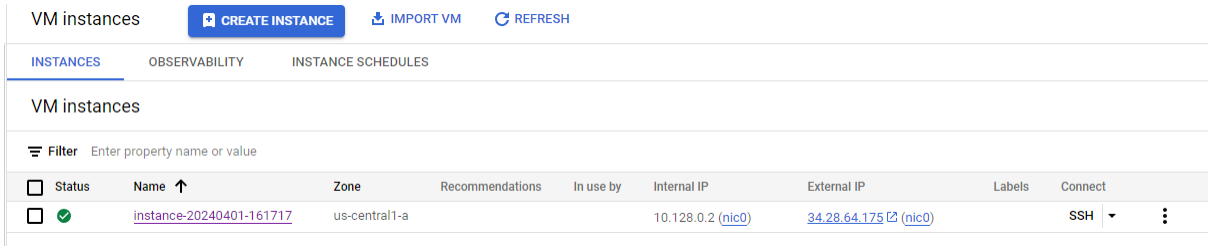

Fig. 6. New instance displayed on Google Cloud Compute Engine control panel.

Nebula lighthouse on Google Cloud Platform instance.

Step 1: Download Nebula

wget https://github.com/slackhq/nebula/releases/download/v1.8.2/nebula-linux-amd64.tar.gz

Step 2: Extract the Archive

tar -xzvf nebula-linux-amd64.tar.gz

Step 3: Move Nebula Binaries to '/usr/local/bin'

sudo mv nebula /usr/local/bin/

sudo mv nebula-cert /usr/local/bin/

Step 4: Configuration

lighthouse_config.yaml

D: > IT Carlow > Y4_2023_2024 > OneDrive - South East Technological University > Y4 2023_2024 > Cloud Data Centers > labs > Lab2 > lighthouse_config.yaml

```yaml
# This is the nebula example configuration file. You must edit, at a minimum, the static_host_map, lighthouse, and firewall sections
# Some options in this file are HUPable, including the pki section. (A HUP will reload credentials from disk without affecting existing tunnels)

# PKI defines the location of credentials for this node. Each of these can also be inlined by using the yaml ": |" syntax.
pki:
  # The CAs that are accepted by this node. Must contain one or more certificates created by 'nebula-cert ca'
  ca: /home/peter_artman/ca.crt
  cert: /home/peter_artman/lighthouse.crt
  key: /home/peter_artman/lighthouse.key
  # blocklist is a list of certificate fingerprints that we will refuse to talk to
  #blocklist:
  #  - c99d4e650533b92061b09918e838a5a0a6aaee21eed1d12fd937682865936c72
  # disconnect_invalid is a toggle to force a client to be disconnected if the certificate is expired or invalid.
  #disconnect_invalid: false

# The static host map defines a set of hosts with fixed IP addresses on the internet (or any network).
# A host can have multiple fixed IP addresses defined here, and nebula will try each when establishing a tunnel.
# The syntax is:
#   "{nebula ip}": ["{routable ip/dns name}:{routable port}"]
# Example, if your lighthouse has the nebula IP of 192.168.100.1 and has the real ip address of 100.64.22.11 and runs on port 4242:
static_host_map:
  "192.168.100.1": ["34.28.64.175:4242"]
lighthouse:
  # am_lighthouse is used to enable lighthouse functionality for a node. This should ONLY be true on nodes
  # you have configured to be lighthouses in your network
  am_lighthouse: true
```

Certificates used inside Nebula network.

```
nebula  nebula-cert  nebula-linux-amd64.tar.gz

┌──(peter㉿LAPTOP-IO62M5BI)-[/mnt/c/Users/peter/Desktop/nebula]
└─$ ./nebula-cert ca -name "Piotr Artman, Y4, Cloud Data Centres"

┌──(peter㉿LAPTOP-IO62M5BI)-[/mnt/c/Users/peter/Desktop/nebula]
└─$ ls
ca.crt  ca.key  nebula  nebula-cert  nebula-linux-amd64.tar.gz

┌──(peter㉿LAPTOP-IO62M5BI)-[/mnt/c/Users/peter/Desktop/nebula]
└─$ ./nebula-cert sign -name "lighthouse" -ip "192.168.100.1/24"

┌──(peter㉿LAPTOP-IO62M5BI)-[/mnt/c/Users/peter/Desktop/nebula]
└─$ ./nebula-cert sign -name "laptop_P50" -ip "192.168.100.3/24" -groups "laptop,ssh"

┌──(peter㉿LAPTOP-IO62M5BI)-[/mnt/c/Users/peter/Desktop/nebula]
└─$ ./nebula-cert sign -name "laptop_SurfacePro6" -ip "192.168.100.5/24" -groups "laptop,ssh"

┌──(peter㉿LAPTOP-IO62M5BI)-[/mnt/c/Users/peter/Desktop/nebula]
└─$ ls
ca.crt          laptop_P50.key          lighthouse.crt  nebula-cert
ca.key          laptop_SurfacePro6.crt  lighthouse.key  nebula-linux-amd64.tar.gz
laptop_P50.crt  laptop_SurfacePro6.key  nebula

┌──(peter㉿LAPTOP-IO62M5BI)-[/mnt/c/Users/peter/Desktop/nebula]
└─$ rm laptop_P50.key  lighthouse.crt laptop_SurfacePro6.crt lighthouse.key laptop_P50.crt  lapto
p_SurfacePro6.key

┌──(peter㉿LAPTOP-IO62M5BI)-[/mnt/c/Users/peter/Desktop/nebula]
└─$ ls
ca.crt  ca.key  nebula  nebula-cert  nebula-linux-amd64.tar.gz

┌──(peter㉿LAPTOP-IO62M5BI)-[/mnt/c/Users/peter/Desktop/nebula]
└─$ ./nebula-cert sign -name "lighthouse" -ip "192.168.100.1/24"

┌──(peter㉿LAPTOP-IO62M5BI)-[/mnt/c/Users/peter/Desktop/nebula]
└─$ ./nebula-cert sign -name "laptop_SurfacePro6" -ip "192.168.100.5/24"

┌──(peter㉿LAPTOP-IO62M5BI)-[/mnt/c/Users/peter/Desktop/nebula]
└─$ ./nebula-cert sign -name "laptop_P50" -ip "192.168.100.3/24"

┌──(peter㉿LAPTOP-IO62M5BI)-[/mnt/c/Users/peter/Desktop/nebula]
└─$ ./nebula-cert sign -name "mobile_SamsungS21" -ip "192.168.100.7/24"

┌──(peter㉿LAPTOP-IO62M5BI)-[/mnt/c/Users/peter/Desktop/nebula]
└─$ ls
ca.crt          laptop_SurfacePro6.crt  mobile_SamsungS21.crt  nebula-linux-amd64.tar.gz
ca.key          laptop_SurfacePro6.key  mobile_SamsungS21.key
laptop_P50.crt  lighthouse.crt          nebula
laptop_P50.key  lighthouse.key          nebula-cert

┌──(peter㉿LAPTOP-IO62M5BI)-[/mnt/c/Users/peter/Desktop/nebula]
└─$
```

Settings on client machine

laptop_P50_config.yaml

D: > IT Carlow > Y4_2023_2024 > OneDrive - South East Technological University > Y4 2023_2024 > Cloud Data Centers > labs > Lab2 > laptop_P50_config.yaml

```yaml
# This is the nebula example configuration file. You must edit, at a minimum, the static_host_map, lighthouse, and firewall sections
# Some options in this file are HUPable, including the pki section. (A HUP will reload credentials from disk without affecting existing tunnels)

# PKI defines the location of credentials for this node. Each of these can also be inlined by using the yaml ": |" syntax.
pki:
  # The CAs that are accepted by this node. Must contain one or more certificates created by 'nebula-cert ca'
  ca: /home/peter/ca.crt
  cert: /home/peter/laptop_P50.crt
  key: /home/peter/laptop_P50.key
  # blocklist is a list of certificate fingerprints that we will refuse to talk to
  #blocklist:
  #  - c99d4e650533b92061b09918e838a5a0a6aaee21eed1d12fd937682865936c72
  # disconnect_invalid is a toggle to force a client to be disconnected if the certificate is expired or invalid.
  #disconnect_invalid: false

# The static host map defines a set of hosts with fixed IP addresses on the internet (or any network).
# A host can have multiple fixed IP addresses defined here, and nebula will try each when establishing a tunnel.
# The syntax is:
#   "{nebula ip}": ["{routable ip/dns name}:{routable port}"]
# Example, if your lighthouse has the nebula IP of 192.168.100.1 and has the real ip address of 100.64.22.11 and runs on port 4242:
static_host_map:
  "192.168.100.1": ["34.28.64.175:4242"]
lighthouse:
  # am_lighthouse is used to enable lighthouse functionality for a node. This should ONLY be true on nodes
  # you have configured to be lighthouses in your network
  am_lighthouse: false
```

REFERENCES:

[1] "Quick Start - Nebula Docs." Accessed: Apr. 02, 2024. [Online]. Available: https://nebula.defined.net/docs/guides/quick-start/

[2] "Free cloud features and trial offer - Google Cloud Free Program." Accessed: Apr. 01, 2024. [Online]. Available: https://cloud.google.com/free/docs/free-cloud-features#compute