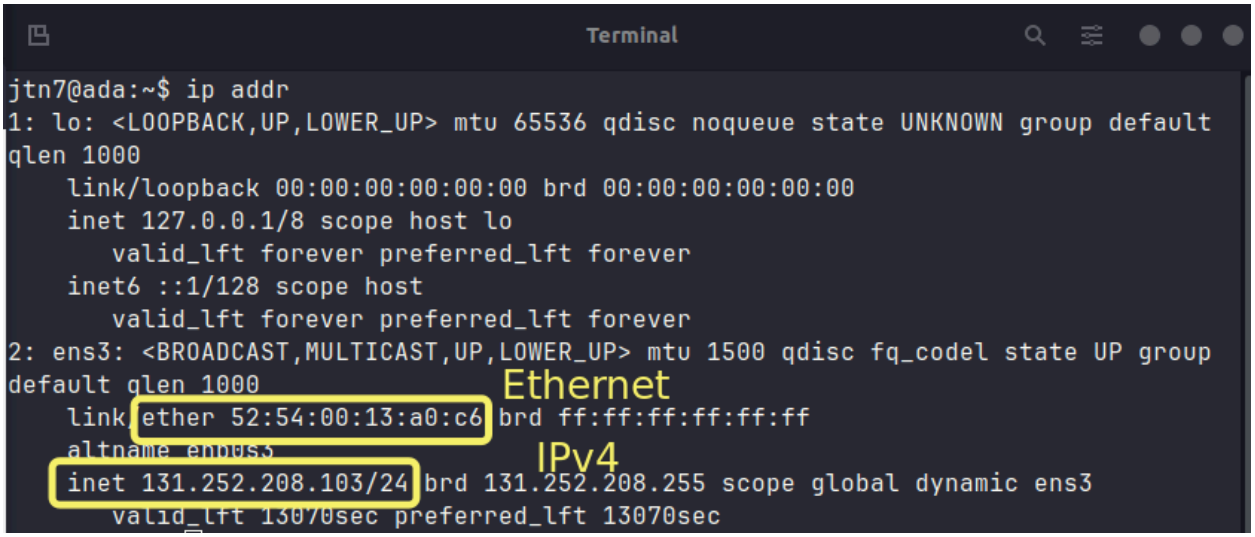
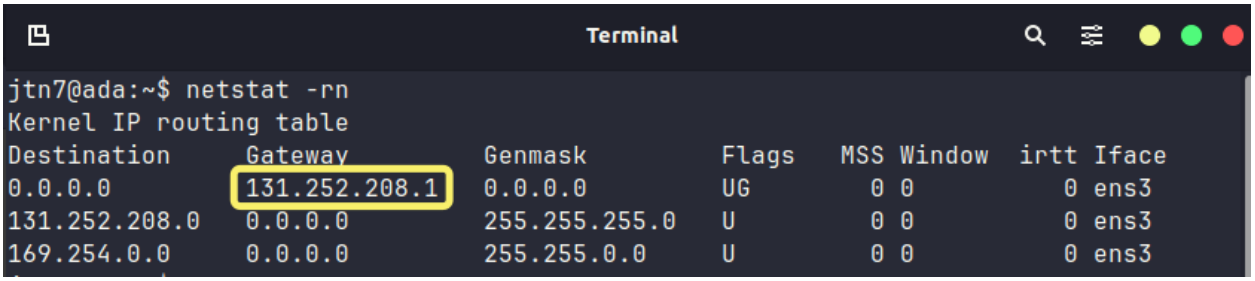
Odin ID: jtn7

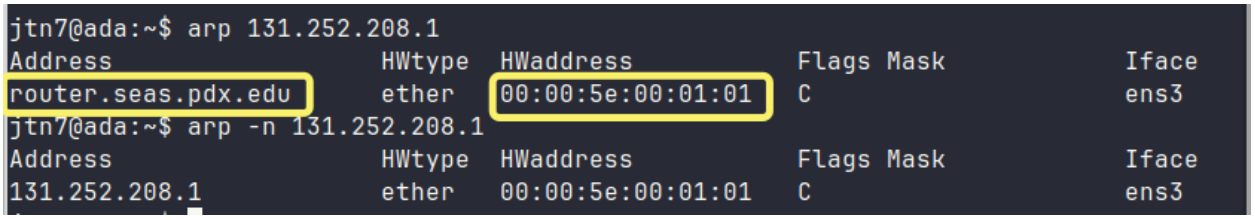
Use the ip address command to find the IPv4 address and hardware address of the local ethernet card interface (Typically beginning with eth, ens, or enp). Include both in your lab notebook.



Perform a netstat -rn to list the route table for the machine.
What is the default router's IP address (e.g. the gateway address for the default route 0.0.0.0/0)?



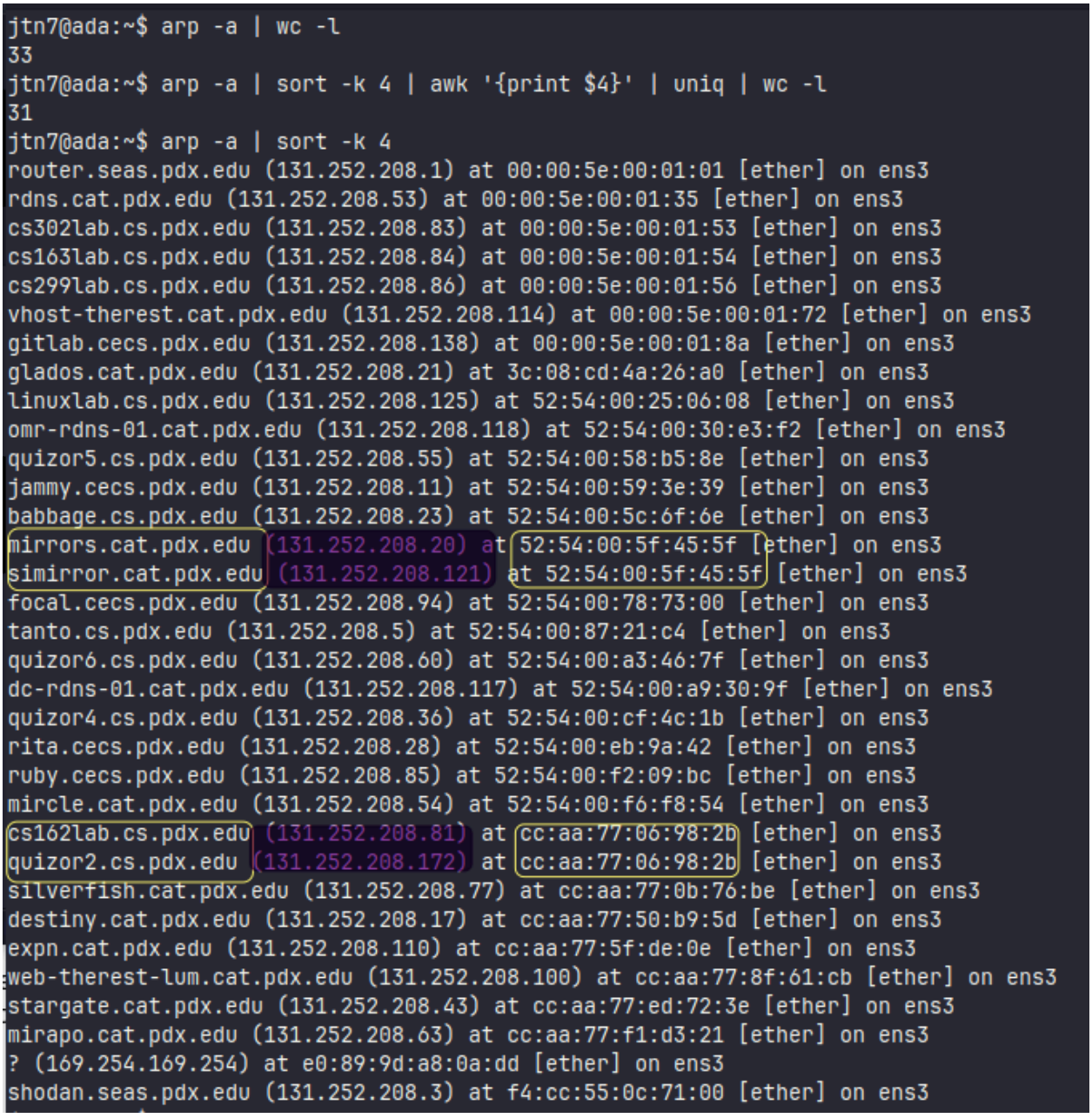
Perform an arp command (both with and without the -n flag) on the IP address of the router.
What is the name of the default router and its hardware address?



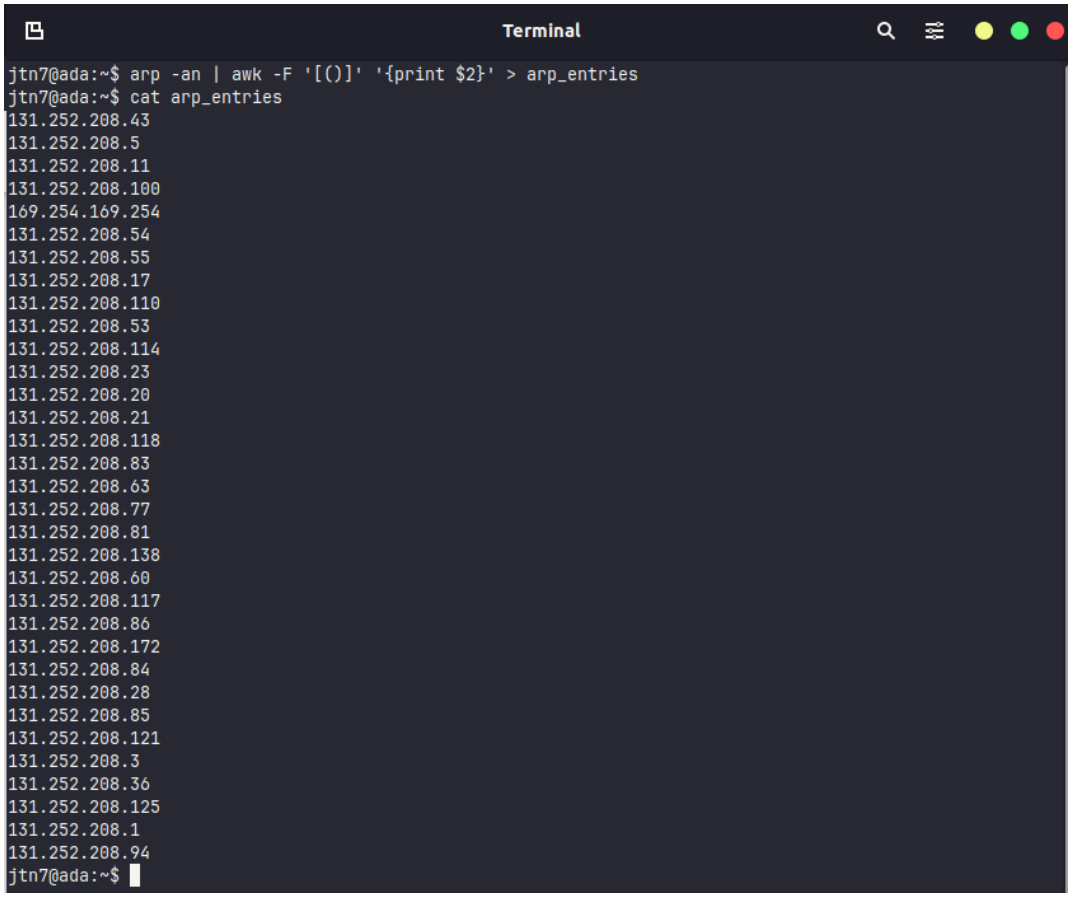
We can pipe the output of the command to wc -l to determine the number of entries in the table.
How many entries are there in the ARP table?
33

List any IP addresses share the same hardware address:
131.252.208.20
131.252.208.121
131.252.208.121
131.252.208.121

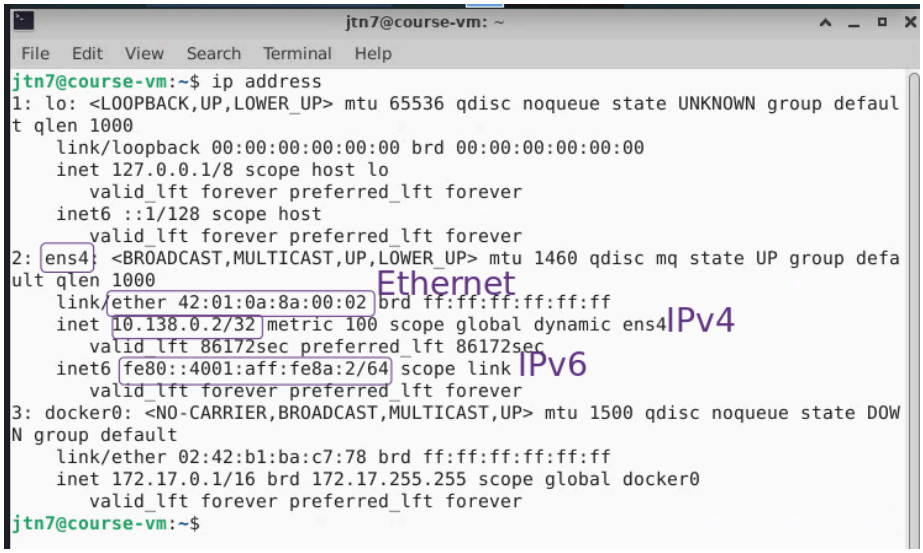
How many less hardware addresses are there than IP addresses in the ARP table? 2



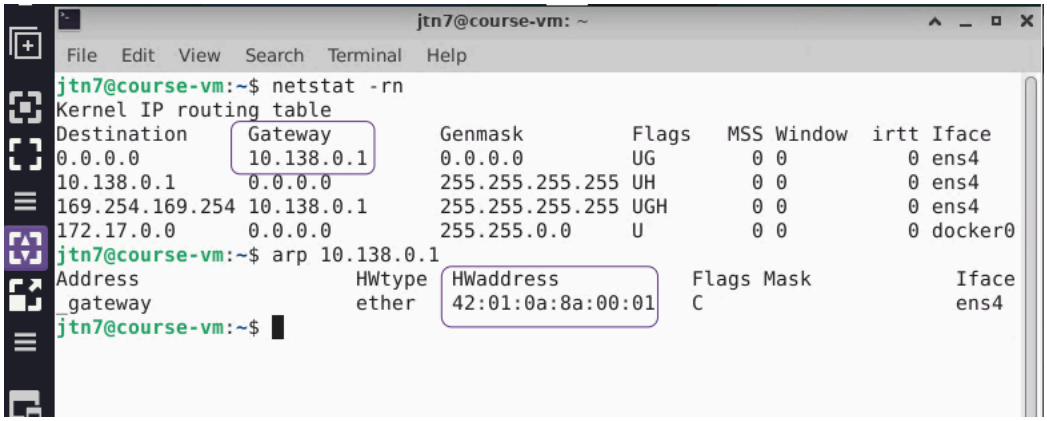
Use a single command-line to create a file that contains each IP address that appears in the machine's ARP table and places the results in a file called arp_entries. Include the command in your lab notebook. What network prefix do most of the IP addresses in the ARP table share? **131.252.208**



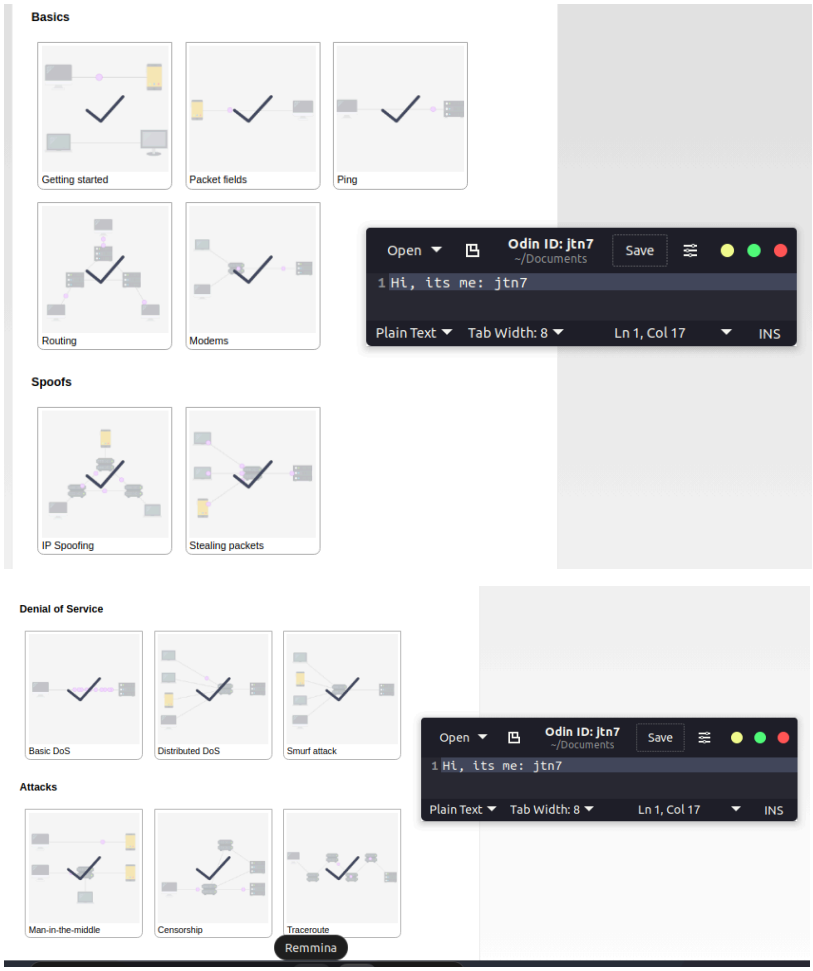
Find the IP address and hardware address of the local ethernet card interface (Typically beginning with eth, ens, or enp). Include both in your lab notebook



What is the default router's IP address (e.g. the gateway address for the default route 0.0.0.0/0)
10.138.0.1
What is the default router's hardware address? 42:01:0a:8a:00:01



Netsim: upon completion of all levels take a screenshot of the completed list of levels



Run nmap on the internal subnet the instances have been placed on. Show a screenshot of the output for the scan for your lab notebook.

```
jtn7@course-vm:~$ nmap 10.138.0.3
Starting Nmap 7.80 ( https://nmap.org ) at 2024-01-13 19:25 UTC
Nmap scan report for tikiwiki-20-07-2020-1-vm.c.cloud-nguyen-jtn7.internal (10.138.0.3)
Host is up (0.00022s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 0.06 seconds
jtn7@course-vm:~$ nmap 10.138.0.4
Starting Nmap 7.80 ( https://nmap.org ) at 2024-01-13 19:25 UTC
Nmap scan report for orocommerce-1-vm.c.cloud-nguyen-jtn7.internal (10.138.0.4)
Host is up (0.00027s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 0.06 seconds
jtn7@course-vm:~$ nmap 10.138.0.5
Starting Nmap 7.80 ( https://nmap.org ) at 2024-01-13 19:25 UTC
Nmap scan report for joomla-1-vm.c.cloud-nguyen-jtn7.internal (10.138.0.5)
Host is up (0.00031s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 0.06 seconds
jtn7@course-vm:~$
```

How many subnetworks are created initially on the default network? How many regions does this correspond to? (Use a pipe to pass output to grep in order to return specific lines of output and then another to pass output to wc to count them: | grep default | wc -l)
84 subnetworks were created and this corresponds to 84 regions.

Given the CIDR prefix associated with each subnetwork, how many hosts does each subnetwork support? **20**

Create two instances in different zones in separate regions of your choice. List both instances.

```
jtn7@cloudshell:~ (cloud-nguyen-jtn7)$ gcloud compute instances list
NAME: instance-2
ZONE: us-central1-a
MACHINE_TYPE: n1-standard-1
PREEMPTIBLE:
INTERNAL_IP: 10.128.0.2
EXTERNAL_IP: 34.171.224.44
STATUS: RUNNING

NAME: instance-1
ZONE: us-west1-a
MACHINE_TYPE: n1-standard-1
PREEMPTIBLE:
INTERNAL_IP: 10.138.0.6
EXTERNAL_IP: 34.168.230.196
STATUS: RUNNING

NAME: course-vm
ZONE: us-west1-b
MACHINE_TYPE: e2-medium
PREEMPTIBLE:
INTERNAL_IP: 10.138.0.2
EXTERNAL_IP: 34.145.38.62
STATUS: RUNNING
```

Which CIDR subnetworks are these instances brought up in? Do they correspond to the appropriate region based on the prior commands?
The /20 prefix CIDR and yes, they correspond to the appropriate region.

From instance-1, perform a ping to the Internal IP address of instance-2. Take a screenshot of the output.

```
jtn7@instance-1:~$ ping 10.128.02
PING 10.128.02 (10.128.0.2) 56(84) bytes of data.
64 bytes from 10.128.0.2: icmp_seq=1 ttl=64 time=39.1 ms
64 bytes from 10.128.0.2: icmp_seq=2 ttl=64 time=38.7 ms
64 bytes from 10.128.0.2: icmp_seq=3 ttl=64 time=38.6 ms
64 bytes from 10.128.0.2: icmp_seq=4 ttl=64 time=38.6 ms
64 bytes from 10.128.0.2: icmp_seq=5 ttl=64 time=38.6 ms
64 bytes from 10.128.0.2: icmp_seq=6 ttl=64 time=38.6 ms
64 bytes from 10.128.0.2: icmp_seq=7 ttl=64 time=38.3 ms
64 bytes from 10.128.0.2: icmp_seq=8 ttl=64 time=35.0 ms
64 bytes from 10.128.0.2: icmp_seq=9 ttl=64 time=35.1 ms
64 bytes from 10.128.0.2: icmp_seq=10 ttl=64 time=35.0 ms
64 bytes from 10.128.0.2: icmp_seq=11 ttl=64 time=35.0 ms
64 bytes from 10.128.0.2: icmp_seq=12 ttl=64 time=35.0 ms
^C
--- 10.128.02 ping statistics ---
12 packets transmitted, 12 received, 0% packet loss, time 11016ms
rtt min/avg/max/mdev = 34.959/37.139/39.091/1.797 ms
```

From the figure in the previous step. What facilitates this connectivity: the virtual switch or the VPN Gateway?
The VPN gateway because instance 1 and instance 2 are not in the same region, so the VPN is used to route traffic between GCP infrastructure (see highlighted below).

The infrastructure that is deployed to implement this is shown in red. Because these subnetworks were initially private, virtual switches that handle traffic within GCP infrastructure must be used to encrypt traffic between the 3 subnetworks. The figure also shows VPN gateways that must be used to encrypt and route traffic between GCP infrastructure and external destinations such as the customer site. Note that the CIDR prefixes for each subnetwork employ private IP address ranges that are not reachable externally (e.g. `10.240.0.0/24`, `192.168.1.0/24`, and `10.2.0.0/16`).

Explain why the result of this ping is different from when you performed the ping to instance-2.
Because instance-3 and instance-4 are on the custom network we created, custom-network-1 with the 24 CIDR prefix while instance-1 is on the default network with the 20 CIDR prefix, each subnetwork employ private IP address ranges that are not reachable externally. Thus instance-1 cannot reach instance-3 and instance-4 and the packets are dropped. (See highlighted text above).

Take a screenshot of the new subnets created in custom-network1 alongside the default subnetworks in those regions assigned to the default network.

```
jtn7@cloudshell:~ (cloud-nguyen-jtn7)$ gcloud compute networks subnets list --regions=us-central1,europe-west1
NAME: default
REGION: europe-west1
NETWORK: default
RANGE: 10.132.0.0/20
STACK_TYPE: IPV4_ONLY
IPV6_ACCESS_TYPE:
INTERNAL_IPV6_PREFIX:
EXTERNAL_IPV6_PREFIX:

NAME: subnet-europe-west-192
REGION: europe-west1
NETWORK: custom-network1
RANGE: 192.168.5.0/24
STACK_TYPE: IPV4_ONLY
IPV6_ACCESS_TYPE:
INTERNAL_IPV6_PREFIX:
EXTERNAL_IPV6_PREFIX:

NAME: default
REGION: us-central1
NETWORK: default
RANGE: 10.128.0.0/20
STACK_TYPE: IPV4_ONLY
IPV6_ACCESS_TYPE:
INTERNAL_IPV6_PREFIX:
EXTERNAL_IPV6_PREFIX:

NAME: subnet-us-central-192
REGION: us-central1
NETWORK: custom-network1
RANGE: 192.168.1.0/24
STACK_TYPE: IPV4_ONLY
IPV6_ACCESS_TYPE:
INTERNAL_IPV6_PREFIX:
EXTERNAL_IPV6_PREFIX:
```

Take screenshots of all 4 instances in the UI including the network they belong to.

☐	✓	instance-1	us-west1-a	10.138.0.6 (nic0)	34.168.230.196 (nic0)	default	SSH	▾	⋮
☐	✓	instance-2	us-central1-a	10.128.0.2 (nic0)	34.171.224.44 (nic0)	default	SSH	▾	⋮
☐	✓	instance-3	us-central1-a	192.168.1.2 (nic0)	34.173.104.3 (nic0)	custom-network1	SSH	▾	⋮
☐	✓	instance-4	europe-west1-d	192.168.5.2 (nic0)	34.34.183.41 (nic0)	custom-network1	SSH	▾	⋮

Take a screenshot of the subnetworks created for the custom-network1 network and some of the subnetworks of the default network showing their regions, internal IP ranges and Gateways.

custom-network1

OVERVIEW | SUBNETS | STATIC INTERNAL IP ADDRESSES | FIREWALLS | FIREWALL ENDPOINTS | ROUTES | VPC NETWORK PEERING | PRIVATE SERVICES A(

Subnets ADD SUBNET FLOW LOGS

Filter Enter property name or value

Name	Region	Stack Type	Internal IP ranges	External IP ranges	Secondary IPv4 ranges	Gateway	Private Google Access	Flow logs
subnet-europe-west-192	europe-west1	IPv4	192.168.5.0/24	None	None	192.168.5.1	Off	Off
subnet-us-central-192	us-central1	IPv4	192.168.1.0/24	None	None	192.168.1.1	Off	Off

default

OVERVIEW | SUBNETS | STATIC INTERNAL IP ADDRESSES | FIREWALLS | FIREWALL ENDPOINTS | ROUTES | VPC NETWORK PEERING | PRIVATE SERVICES

Subnets ADD SUBNET FLOW LOGS

Filter Enter property name or value

Name	Region	Stack Type	Internal IP ranges	External IP ranges	Secondary IPv4 ranges	Gateway	Private Google Access	Flo
default	us-central1	IPv4	10.128.0.0/20	None	None	10.128.0.1	Off	Off
default	europe-west1	IPv4	10.132.0.0/20	None	None	10.132.0.1	Off	Off
default	us-west1	IPv4	10.138.0.0/20	None	None	10.138.0.1	Off	Off
default	asia-east1	IPv4	10.140.0.0/20	None	None	10.140.0.1	Off	Off
default	us-east1	IPv4	10.142.0.0/20	None	None	10.142.0.1	Off	Off
default	asia-northeast1	IPv4	10.146.0.0/20	None	None	10.146.0.1	Off	Off