Addison Wurtz
CS 530
notebooks/Week1

Lab Notebook – Week 1

Table of Contents:

Addison Wurtz
CS 530
notebooks/Week1

01.2: ARP, Wireshark, Netsim

1. ARP (linux.cs.pdx.edu)

IPv4 address and hardware address of local ethernet card interface:

IP address: 131.252.208.103/24

Hardware address: 52:54:00:13:a0:c6

```
2: ens3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 52:54:00:13:a0:c6 brd ff:ff:ff:ff:ff:ff
    altname enp0s3
    inet 131.252.208.103/24 brd 131.252.208.255 scope global dynamic ens3
       valid_lft 9292sec preferred_lft 9292sec
awurtz@ada:~$
```

The default router's IP address is 131.252.208.1

```
awurtz@ada:~$ netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
0.0.0.0         131.252.208.1   0.0.0.0         UG        0 0          0 ens3
131.252.208.0   0.0.0.0         255.255.255.0   U         0 0          0 ens3
169.254.0.0     0.0.0.0         255.255.0.0     U         0 0          0 ens3
awurtz@ada:~$
```

Default router name: router.seas.pdx.edu
Default router hardware address: 00:00:5e:00:01:01

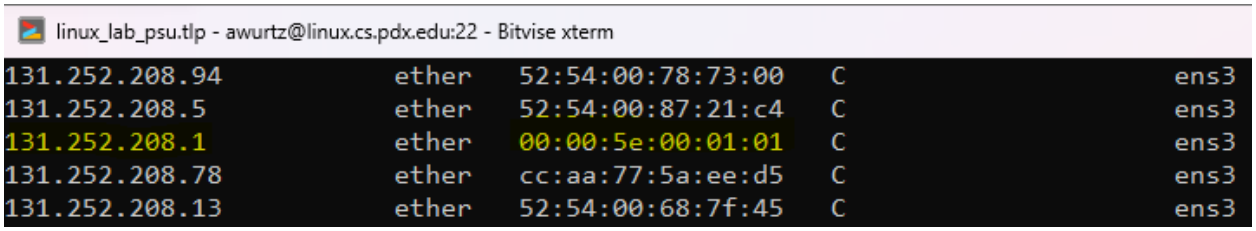
`arp`

linux_lab_psu.tlp - awurtz@linux.cs.pdx.edu:22 - Bitvise xterm

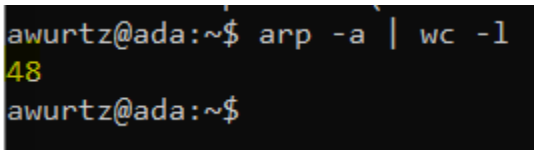
```
focal.cecs.pdx.edu       ether   52:54:00:78:73:00   C                     ens3
tanto.cs.pdx.edu         ether   52:54:00:87:21:c4   C                     ens3
router.seas.pdx.edu      ether   00:00:5e:00:01:01   C                     ens3
termite.cat.pdx.edu      ether   cc:aa:77:5a:ee:d5   C                     ens3
quizor3.cs.pdx.edu       ether   52:54:00:68:7f:45   C                     ens3
```

Addison Wurtz
CS 530
notebooks/Week1

`arp -n`



Number of entries in ARP table: 48



Addison Wurtz
CS 530
notebooks/Week1

2. -

IP addresses that share the same hardware address:

Hardware address	**IP addresses**
30:e4:db:f9:26:37	169.252.169.254 131.252.208.212
52:54:00:30:e3:f2	131.252.208.118 131.252.208.53
cc:aa:77:2e:16:a0	131.252.208.15 131.252.208.7

```
awurtz@ada:~$ arp -a | sort -k 4
router.seas.pdx.edu (131.252.208.1) at 00:00:5e:00:01:01 [ether] on ens3
mirrors.cat.pdx.edu (131.252.208.20) at 00:00:5e:00:01:14 [ether] on ens3
walt.ee.pdx.edu (131.252.208.38) at 00:00:5e:00:01:26 [ether] on ens3
vhost-users.cat.pdx.edu (131.252.208.59) at 00:00:5e:00:01:3b [ether] on ens3
cs162lab.cs.pdx.edu (131.252.208.81) at 00:00:5e:00:01:51 [ether] on ens3
cs302lab.cs.pdx.edu (131.252.208.83) at 00:00:5e:00:01:53 [ether] on ens3
cs163lab.cs.pdx.edu (131.252.208.84) at 00:00:5e:00:01:54 [ether] on ens3
vhost-therest.cat.pdx.edu (131.252.208.114) at 00:00:5e:00:01:72 [ether] on ens3
gitlab.cecs.pdx.edu (131.252.208.138) at 00:00:5e:00:01:8a [ether] on ens3
? (169.254.169.254) at 30:e4:db:f9:26:37 [ether] on ens3
radiant.seas.pdx.edu (131.252.208.212) at 30:e4:db:f9:26:37 [ether] on ens3
omr-rdns-01.cat.pdx.edu (131.252.208.118) at 52:54:00:30:e3:f2 [ether] on ens3
rdns.cat.pdx.edu (131.252.208.53) at 52:54:00:30:e3:f2 [ether] on ens3
quizor5.cs.pdx.edu (131.252.208.55) at 52:54:00:58:b5:8e [ether] on ens3
jammy.cecs.pdx.edu (131.252.208.11) at 52:54:00:59:3e:39 [ether] on ens3
babbage.cs.pdx.edu (131.252.208.23) at 52:54:00:5c:6f:6e [ether] on ens3
simirror.cat.pdx.edu (131.252.208.121) at 52:54:00:5f:45:5f [ether] on ens3
quizor3.cs.pdx.edu (131.252.208.13) at 52:54:00:68:7f:45 [ether] on ens3
focal.cecs.pdx.edu (131.252.208.94) at 52:54:00:78:73:00 [ether] on ens3
tanto.cs.pdx.edu (131.252.208.5) at 52:54:00:87:21:c4 [ether] on ens3
aarl-web.mme.pdx.edu (131.252.208.105) at 52:54:00:93:91:b9 [ether] on ens3
quizor6.cs.pdx.edu (131.252.208.60) at 52:54:00:a3:46:7f [ether] on ens3
omr-adns-01.cat.pdx.edu (131.252.208.112) at 52:54:00:a5:68:d1 [ether] on ens3
dc-rdns-01.cat.pdx.edu (131.252.208.117) at 52:54:00:a9:30:9f [ether] on ens3
gitlab-01.cecs.pdx.edu (131.252.208.137) at 52:54:00:c2:05:63 [ether] on ens3
quizor4.cs.pdx.edu (131.252.208.36) at 52:54:00:cf:4c:1b [ether] on ens3
rita.cecs.pdx.edu (131.252.208.28) at 52:54:00:eb:9a:42 [ether] on ens3
ruby.cecs.pdx.edu (131.252.208.85) at 52:54:00:f2:09:bc [ether] on ens3
mircle.cat.pdx.edu (131.252.208.54) at 52:54:00:f6:f8:54 [ether] on ens3
quizor2.cs.pdx.edu (131.252.208.172) at cc:aa:77:06:98:2b [ether] on ens3
quizor1.cs.pdx.edu (131.252.208.171) at cc:aa:77:07:f2:7a [ether] on ens3
silverfish.cat.pdx.edu (131.252.208.77) at cc:aa:77:0b:76:be [ether] on ens3
rocket-01.cat.pdx.edu (131.252.208.15) at cc:aa:77:2e:16:a0 [ether] on ens3
rocket.cat.pdx.edu (131.252.208.7) at cc:aa:77:2e:16:a0 [ether] on ens3
quizortest.cs.pdx.edu (131.252.208.124) at cc:aa:77:2f:fa:de [ether] on ens3
```

Addison Wurtz
CS 530
notebooks/Week1

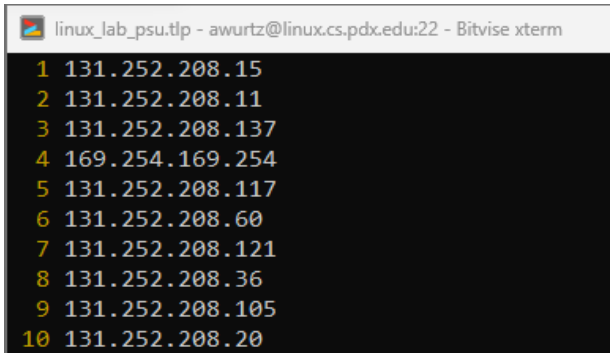
There are 3 fewer hardware addresses than IP addresses (45 hardware addresses, 48 IP)

```
awurtz@ada:~$ arp -a | sort -k 4 | awk '{print $4}' | uniq | wc -l
45
awurtz@ada:~$
```

Command to create file called arp_entires that contains each IP address in the machine's ARP table:

```
awurtz@ada:~$ arp -an | awk -F '[()]' '{print $2}' > arp_entries
```

Most of the IP addresses in arp_entries share the prefix 131.252.208



Addison Wurtz
CS 530
notebooks/Week1

3. ARP (Cloud)

IP address and hardware address of local ethernet card interface:

IP address: 10.138.0.2/32

Hardware address: 42:01:0a:8a:00:02

```
awurtz@course-vm:~$ ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1460 qdisc mq state UP group default qlen 1000
    link/ether 42:01:0a:8a:00:02 brd ff:ff:ff:ff:ff:ff
    inet 10.138.0.2/32 metric 100 scope global dynamic ens4
       valid_lft 86308sec preferred_lft 86308sec
    inet6 fe80::4001:aff:fe8a:2/64 scope link
       valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:78:df:40:49 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
       valid_lft forever preferred_lft forever
```

The default router's IP address is 10.138.0.1

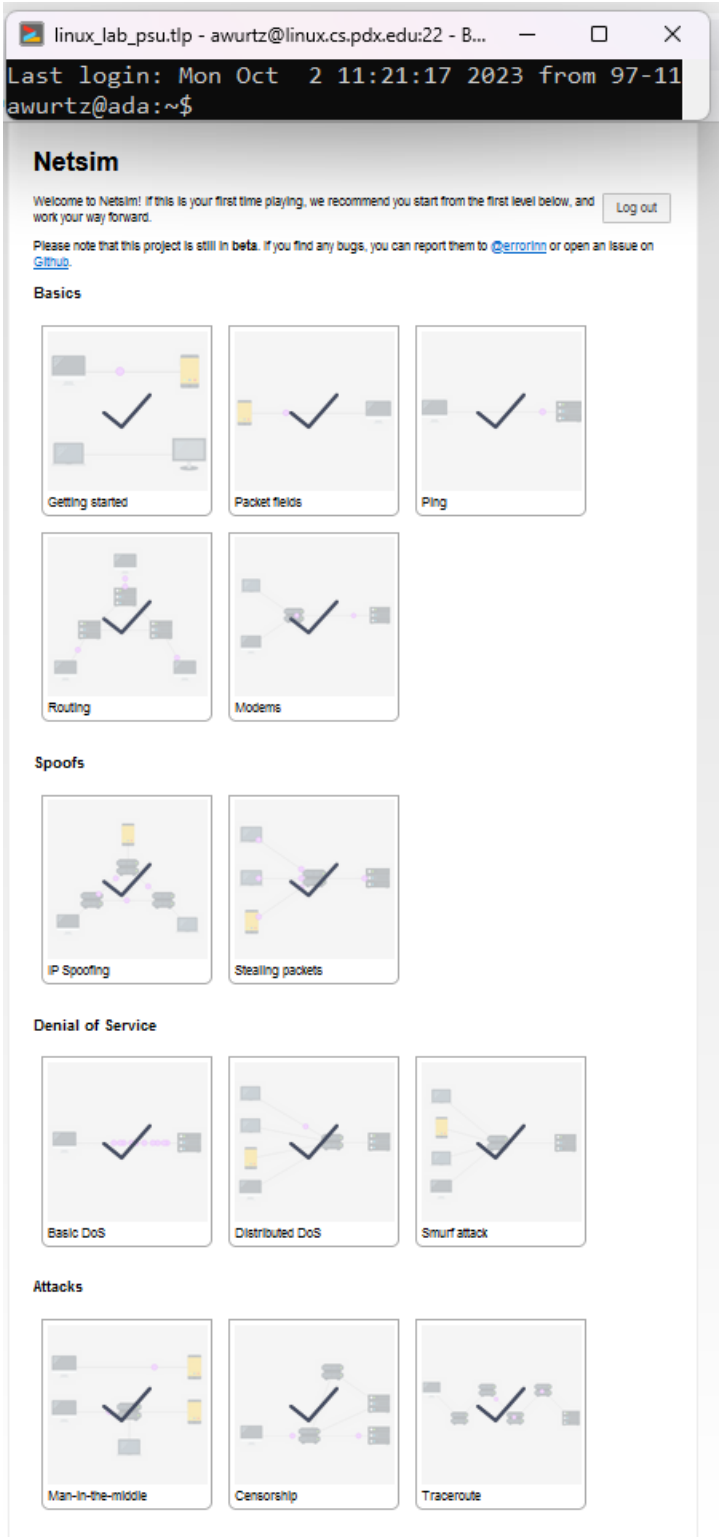
```
awurtz@course-vm:~$ netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
0.0.0.0         10.138.0.1      0.0.0.0         UG        0 0          0 ens4
10.138.0.1      0.0.0.0         255.255.255.255 UH        0 0          0 ens4
169.254.169.254 10.138.0.1      255.255.255.255 UGH       0 0          0 ens4
172.17.0.0      0.0.0.0         255.255.0.0     U         0 0          0 docker0
```

The default router's hardware address is 42:01:0a:8a:00:01

```
awurtz@course-vm:~$ arp 10.138.0.1
Address                  HWtype  HWaddress           Flags Mask            Iface
_gateway                 ether   42:01:0a:8a:00:01   C                     ens4
awurtz@course-vm:~$
```

Addison Wurtz
CS 530
notebooks/Week1

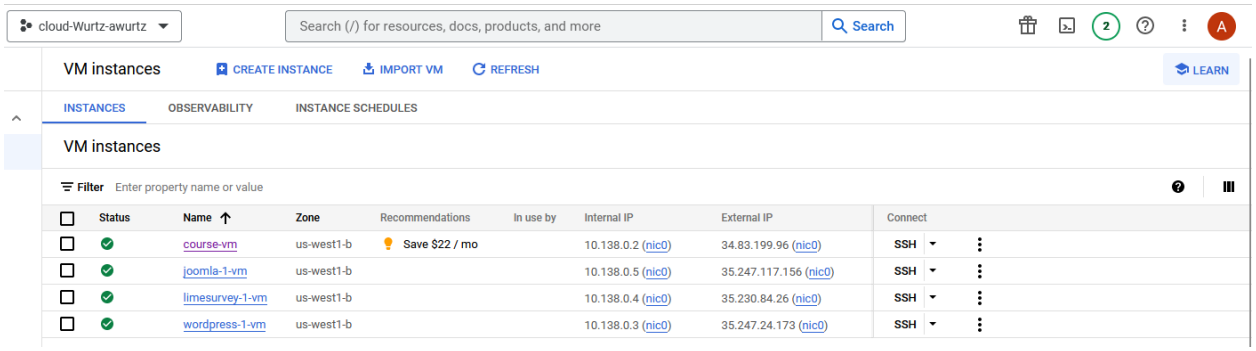
4. Netsim



Addison Wurtz
CS 530
notebooks/Week1

01.3: Cloud Networking

1. Network Scanning (nmap) #1
2. Launch Targets



3. Scan targets for service
4. CIDR and subnets #2
5. Navigating default networks

   *How many subnetworks are created initially on the **default** network? How many regions does this correspond to?*

   Initially, 40 subnets are created:

```
awurtz@cloudshell:~ (cloud-wurtz-awurtz)$ gcloud compute networks subnets list | grep 'NETWORK: default' | wc -l
40
```

   There is one subnetwork for each region:

```
awurtz@cloudshell:~ (cloud-wurtz-awurtz)$ gcloud compute networks subnets list | grep REGION: | sort -k 2 | uniq | wc -l
40
```

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

   Each subnetwork can support 2^12 or 4096 hosts.

Addison Wurtz
CS 530
notebooks/Week1

List of networks in project:

```
awurtz@cloudshell:~ (cloud-wurtz-awurtz)$ gcloud compute instances list
NAME: instance-1
ZONE: us-central1-a
MACHINE_TYPE: n1-standard-1
PREEMPTIBLE:
INTERNAL_IP: 10.128.0.2
EXTERNAL_IP: 34.171.145.46
STATUS: RUNNING

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

NAME: instance-2
ZONE: us-west1-c
MACHINE_TYPE: n1-standard-1
PREEMPTIBLE:
INTERNAL_IP: 10.138.0.6
EXTERNAL_IP: 35.230.84.26
STATUS: RUNNING
awurtz@cloudshell:~ (cloud-wurtz-awurtz)$
```

Which CIDR subnetworks are these instances brought up in? Do they correspond to the appropriate region based on the prior commands?

The subnetworks were brought up in subnetworks us-central1 and us-west1, respectively. They do correspond to the appropriate region. The screenshot below shows the IP range of the corresponding subnetworks.

Addison Wurtz
CS 530
notebooks/Week1

List of automatically created subnetworks:

```
awurtz@cloudshell:~ (cloud-wurtz-awurtz)$ gcloud compute networks subnets list
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: 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: default
REGION: us-west1
NETWORK: default
RANGE: 10.138.0.0/20
STACK_TYPE: IPV4_ONLY
IPV6_ACCESS_TYPE:
INTERNAL_IPV6_PREFIX:
EXTERNAL_IPV6_PREFIX:
```

Pinging instance-2 from instance-1:

```
awurtz@instance-1:~$ ping 10.138.0.6
PING 10.138.0.6 (10.138.0.6) 56(84) bytes of data.
64 bytes from 10.138.0.6: icmp_seq=1 ttl=64 time=39.5 ms
64 bytes from 10.138.0.6: icmp_seq=2 ttl=64 time=38.6 ms
64 bytes from 10.138.0.6: icmp_seq=3 ttl=64 time=38.6 ms
64 bytes from 10.138.0.6: icmp_seq=4 ttl=64 time=38.6 ms
64 bytes from 10.138.0.6: icmp_seq=5 ttl=64 time=38.6 ms
64 bytes from 10.138.0.6: icmp_seq=6 ttl=64 time=38.6 ms
64 bytes from 10.138.0.6: icmp_seq=7 ttl=64 time=38.6 ms
64 bytes from 10.138.0.6: icmp_seq=8 ttl=64 time=38.8 ms
64 bytes from 10.138.0.6: icmp_seq=9 ttl=64 time=32.0 ms
64 bytes from 10.138.0.6: icmp_seq=10 ttl=64 time=32.0 ms
64 bytes from 10.138.0.6: icmp_seq=11 ttl=64 time=32.0 ms
64 bytes from 10.138.0.6: icmp_seq=12 ttl=64 time=31.9 ms
64 bytes from 10.138.0.6: icmp_seq=13 ttl=64 time=31.9 ms
64 bytes from 10.138.0.6: icmp_seq=14 ttl=64 time=32.0 ms
64 bytes from 10.138.0.6: icmp_seq=15 ttl=64 time=32.0 ms
64 bytes from 10.138.0.6: icmp_seq=16 ttl=64 time=31.9 ms
64 bytes from 10.138.0.6: icmp_seq=17 ttl=64 time=32.0 ms
64 bytes from 10.138.0.6: icmp_seq=18 ttl=64 time=32.0 ms
64 bytes from 10.138.0.6: icmp_seq=19 ttl=64 time=32.0 ms
64 bytes from 10.138.0.6: icmp_seq=20 ttl=64 time=32.0 ms
64 bytes from 10.138.0.6: icmp_seq=21 ttl=64 time=31.9 ms
^C
--- 10.138.0.6 ping statistics ---
21 packets transmitted, 21 received, 0% packet loss, time 20029ms
rtt min/avg/max/mdev = 31.935/34.546/39.502/3.285 ms
awurtz@instance-1:~$
```

Addison Wurtz
CS 530
notebooks/Week1

6. Creating custom networks

Create custom network:

```
awurtz@cloudshell:~ (cloud-wurtz-awurtz)$ gcloud compute networks create custom-network1 --subnet-mode custom
Created [https://www.googleapis.com/compute/v1/projects/cloud-wurtz-awurtz/global/networks/custom-network1].
NAME: custom-network1
SUBNET_MODE: CUSTOM
BGP_ROUTING_MODE: REGIONAL
IPV4_RANGE:
GATEWAY_IPV4:

Instances on this network will not be reachable until firewall rules
are created. As an example, you can allow all internal traffic between
instances as well as SSH, RDP, and ICMP by running:

$ gcloud compute firewall-rules create <FIREWALL_NAME> --network custom-network1 --allow tcp,udp,icmp --source-ranges <IP_RANGE>
$ gcloud compute firewall-rules create <FIREWALL_NAME> --network custom-network1 --allow tcp:22,tcp:3389,icmp
```

Project now has default and custom networks:

```
awurtz@cloudshell:~ (cloud-wurtz-awurtz)$ gcloud compute networks list
NAME: custom-network1
SUBNET_MODE: CUSTOM
BGP_ROUTING_MODE: REGIONAL
IPV4_RANGE:
GATEWAY_IPV4:

NAME: default
SUBNET_MODE: AUTO
BGP_ROUTING_MODE: REGIONAL
IPV4_RANGE:
GATEWAY_IPV4:
awurtz@cloudshell:~ (cloud-wurtz-awurtz)$
```

Addison Wurtz
CS 530
notebooks/Week1

Custom subnets alongside regional subnets on the default network:

```
awurtz@cloudshell:~ (cloud-wurtz-awurtz)$ 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:
awurtz@cloudshell:~ (cloud-wurtz-awurtz)$
```

Pinging instance-3 and instance-4 from instance-1:

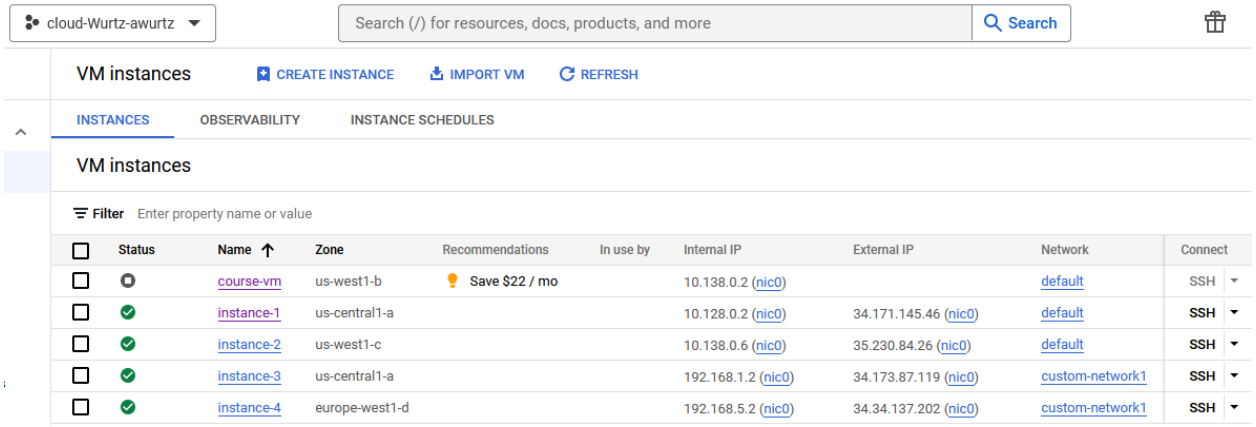
```
awurtz@instance-1:~$ ping 192.168.1.2
PING 192.168.1.2 (192.168.1.2) 56(84) bytes of data.
^C
--- 192.168.1.2 ping statistics ---
34 packets transmitted, 0 received, 100% packet loss, time 33778ms

awurtz@instance-1:~$ ping 192.168.5.2
PING 192.168.5.2 (192.168.5.2) 56(84) bytes of data.
^C
--- 192.168.5.2 ping statistics ---
5 packets transmitted, 0 received, 100% packet loss, time 4076ms
```

Explain why the result of this ping is different from when you performed the ping to instance-2.

This ping is different because instance-1 and instance-2 are on the default network. Instance-3 and instance-4 are on custom-network1. The instances are unable to communicate with each other because they are on different networks.

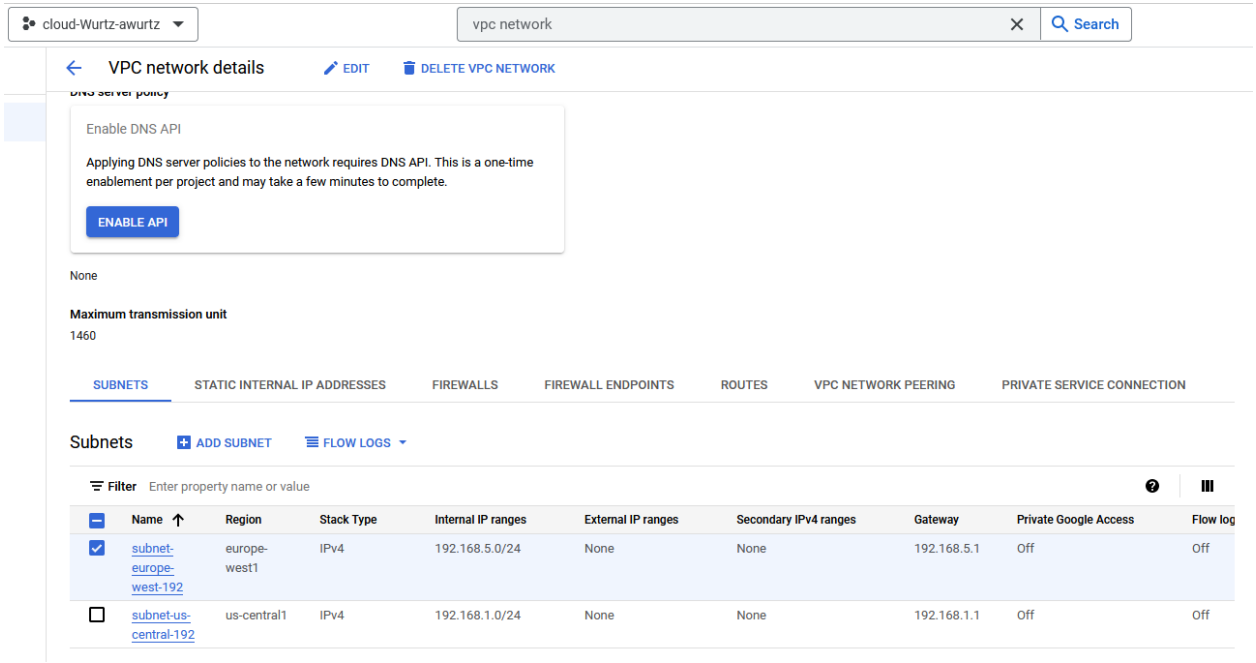
Addison Wurtz
CS 530
notebooks/Week1

Screenshot of all 4 instances in the UI:



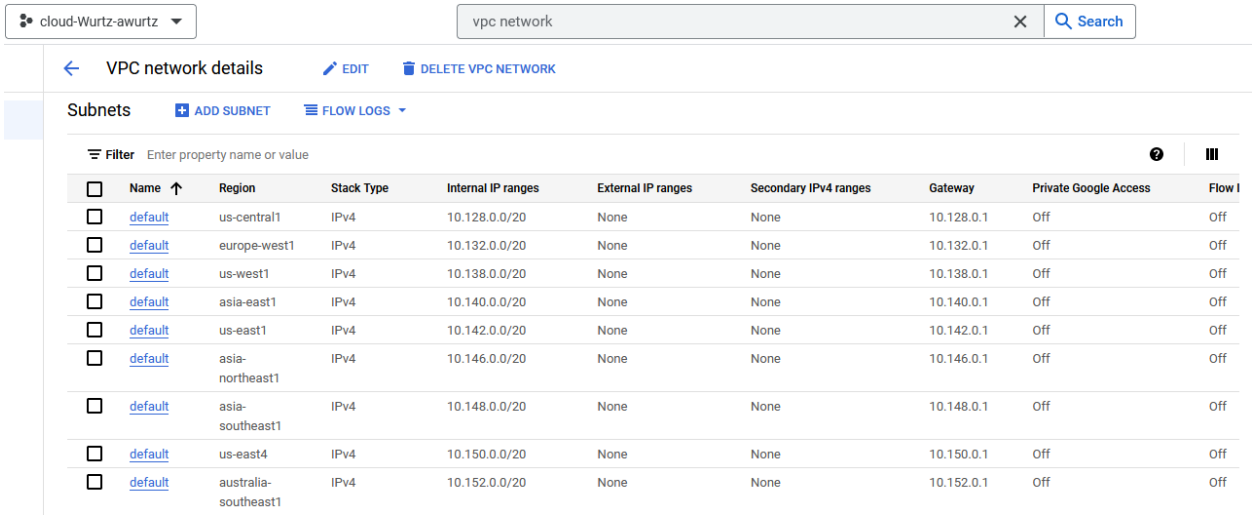
VPC Network Screenshots:

Custom-network1 and its subnetworks:



Addison Wurtz
CS 530
notebooks/Week1

Some subnetworks of default network:



Name	Region	Stack Type	Internal IP ranges	External IP ranges	Secondary IPv4 ranges	Gateway	Private Google Access	Flow l
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
default	asia-southeast1	IPv4	10.148.0.0/20	None	None	10.148.0.1	Off	Off
default	us-east4	IPv4	10.150.0.0/20	None	None	10.150.0.1	Off	Off
default	australia-southeast1	IPv4	10.152.0.0/20	None	None	10.152.0.1	Off	Off

---

7. Clean up

Deleted all VMs and networks.