Linux Ağ Yönetimi 2020 Bahar Final Projesi

Abdulhamit Kumru 170202020

Kullanılan yazılımların versiyonları `Ubuntu Desktop / Server 18`, `Virtual Box 6.1`

Makinelerin Klonlanması

Base olarak kullanacağımız makineyi oluşturduktan sonra klon makineyi menüden sağ tıklayıp istediğimiz şekilde oluşturabiliyoruz. Biz daha az yer kullanmak ve ortak networklere bağlanacağımız için `Linked Clone` ve `Generate new MAC addresses for all network adapters` seçeneğini kullanacağız.

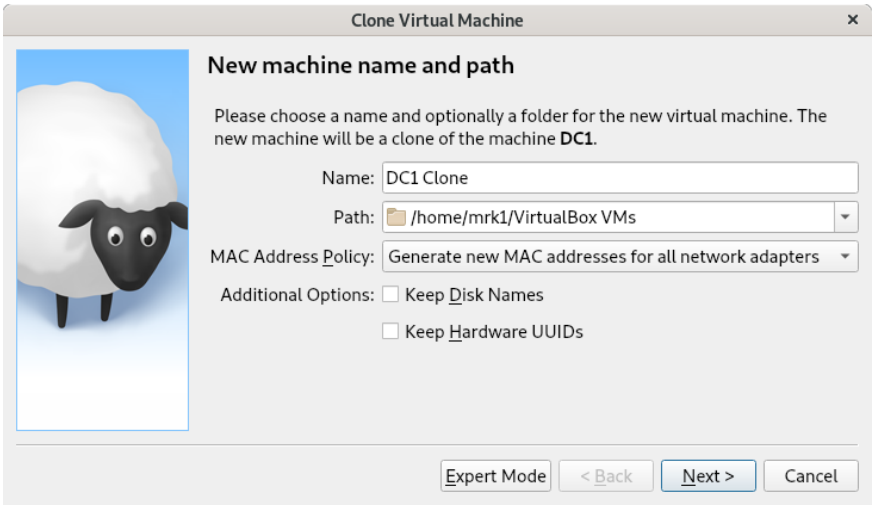

Figure 1: Klonlama

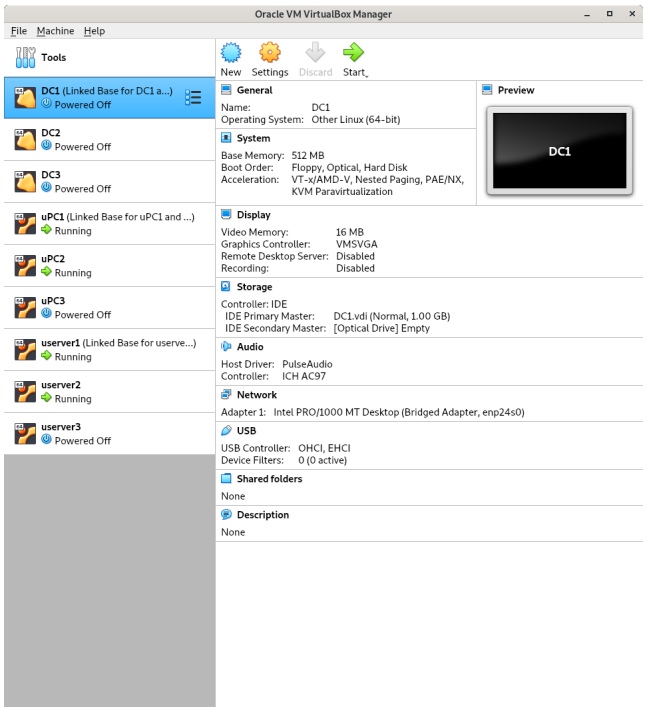

Figure 2: Klonlanmış Makineler

Senaryo 1

NAT ile Host-Guest Ubuntu PC Bağlantısı

Başlangıc olarak guest makinenin Network Adapterini NAT olarak seçiyoruz

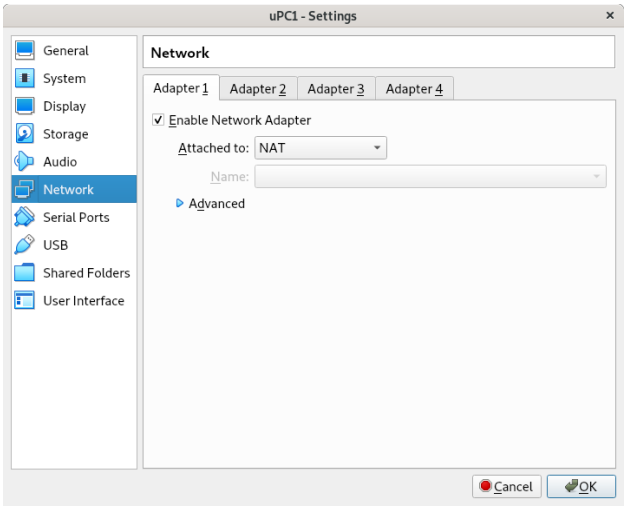

Figure 3: NAT adapter ayarı

SSH

host makineye NAT bağlantı ile erişebilmek için öncelikle gerekli gerekli portu yönlendiriyoruz daha sonra gueste ssh ile bağlanabilmek için open-ssh serveri apt ile yüklüyoruz.

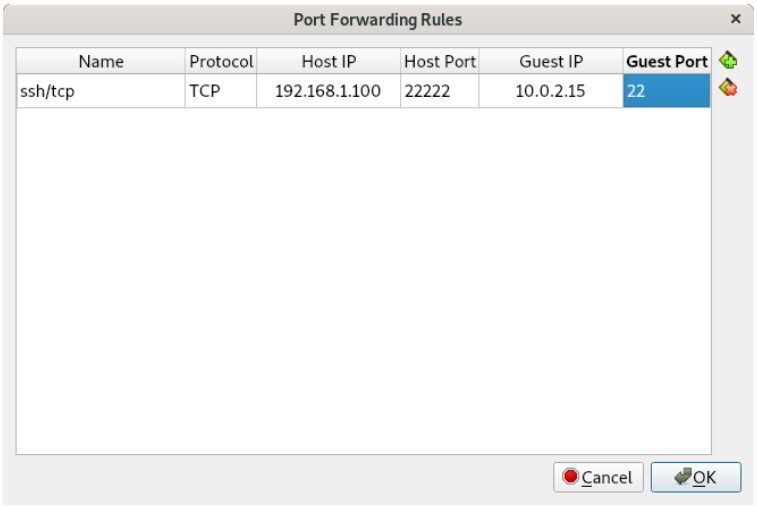

Figure 4: Port forwarding

```
$ apt install opessh-server
```

ssh server servisinin ayarlarını dosyasından 22 numaralı portu ve parolayı kabul edicek şekilde yapıyoruz.

```
$ echo "Port 22" >> /etc/ssh/sshd_config
$ echo "PasswordAuthentication yes" >> /etc/ssh/sshd_config
```

Ayarların aktif olmasi için ssh servisimizi yeniden başlatıyoruz.

```
$ sudo service ssh restart
```

Forward ettiğimiz port üzerinden guest'e bağlabiliriz.

```
$ ssh kumru@192.168.1.100 -p 22222 # hostun ip adresi
```

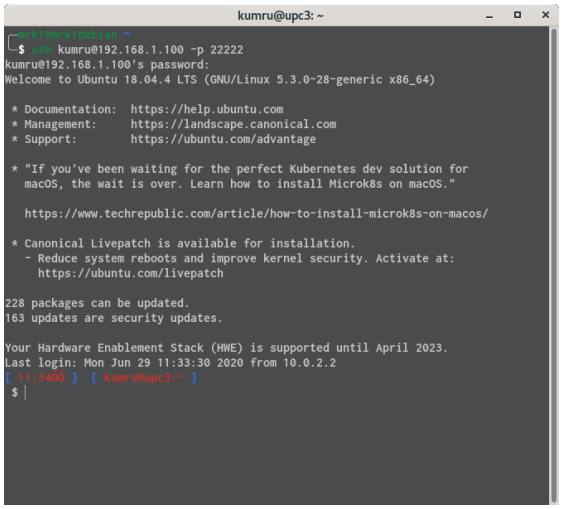

Figure 5: Gueste ssh ile baglanma

Ping

Eğer ICMP port ile çalışsaydı aynı şekilde onunda portunu yönlendirip gueste ping atabilirdik fakat ICMP TCP/UDP portları üzerinden çalışmıyor.

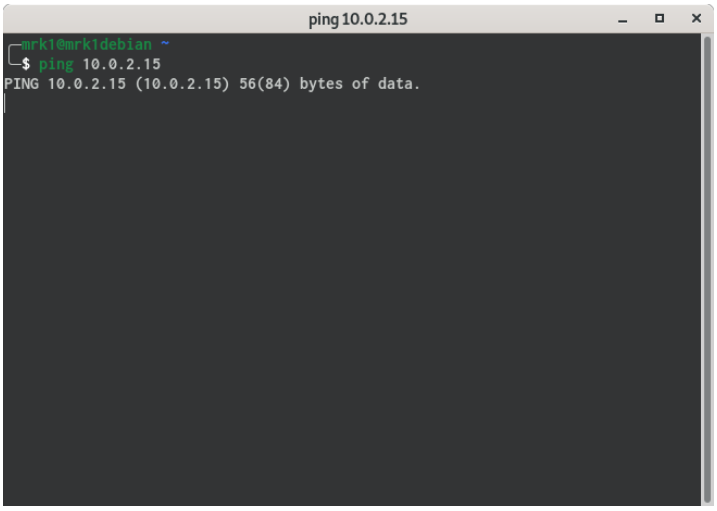

Figure 6: Gueste ping gönderilemiyor

Misafir Eklentileri

Misafir eklentilerini guest makinye sanal disk takarak yükleyebiliyoruz bu sayade hosttan gueste veya guestten hosta kopyala yapıştır ve dosya sürekle bırak yapabiliyoruz. Eklentiyi yüklelemek zor değil sanal makine penceresinden `Devices > Insert Guest Additions CD...` seçeneğini seçince bize autorun.sh'ı çalıştırayım mı diye soruyor evet diyip parolamızı girince eklentiler yükleniyor

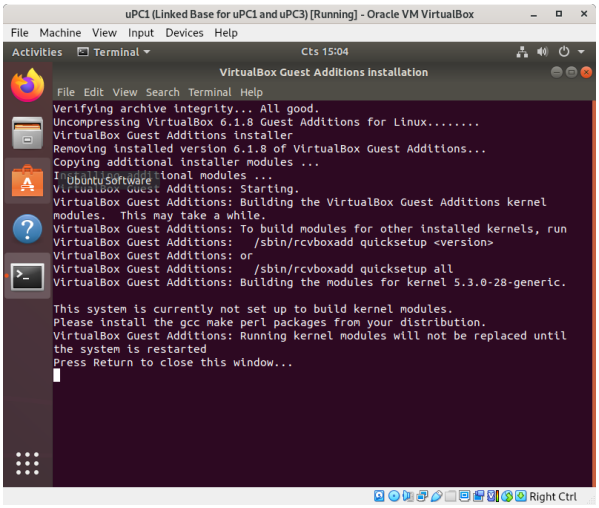

Figure 7: autorun.sh çıktısı

Sanal makinenin penceresinden `Devices > Drag and Drop`, `Devices > Shared Clipboard` seçeneklerinden detaylı ayar yapılabilir.

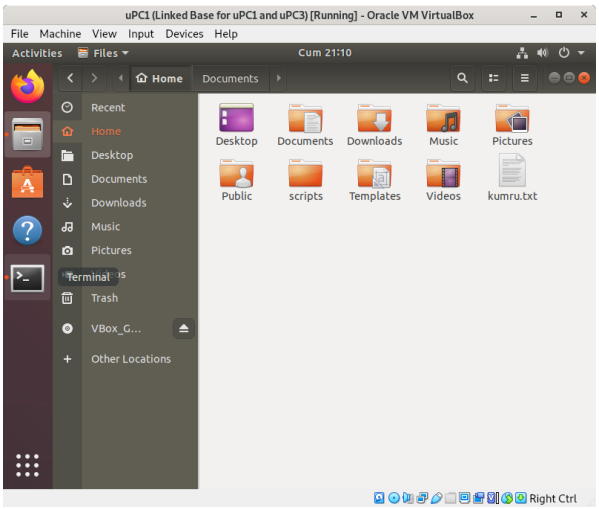

Figure 8: Gueste eklentileri ile sürükle bırak

gif linki (github)

Network Bridge ile Host, TinyCore Guest Bağlantısı

Guestin network adapter ayarını Bridge Adapter olarak seçiyoruz

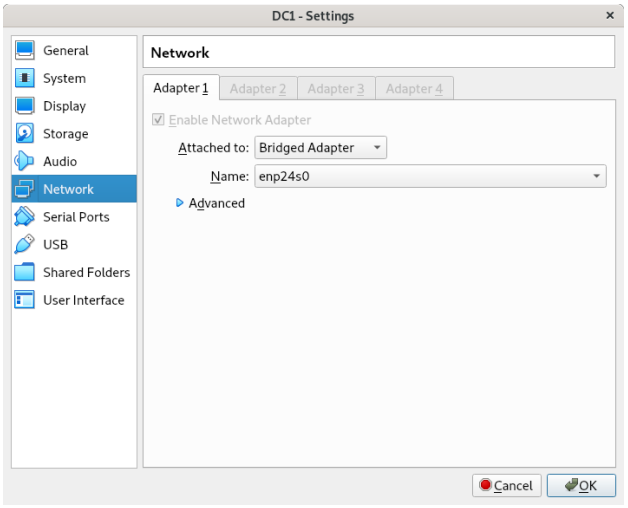

Figure 9: Network Bridge Ayarı

TinyCoreda IP'mizi ifconfig komutu ile kontol edebiliyoruz.

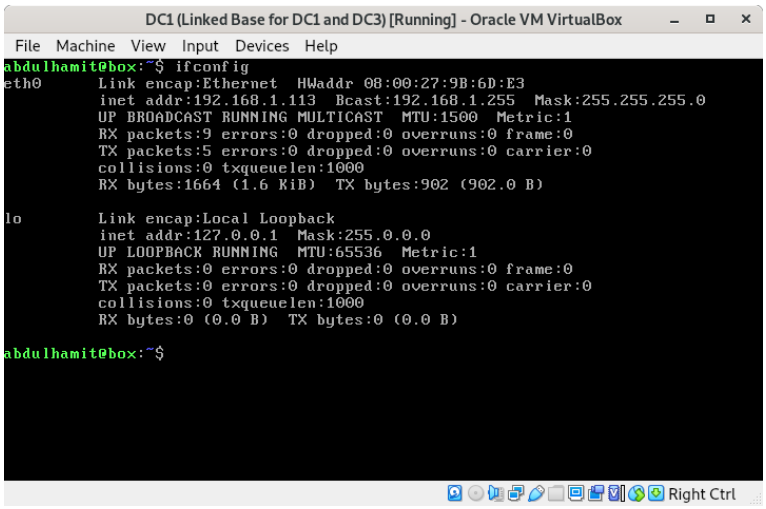

Figure 10: ifconfig çıktısı

Network Bridge ile Guest LANdaki makine gibi kullanılabilir.

SSH

tce-load programı ile TinyCoreda program yükleyebiliyoruz.

```
# openssh client/server paketini yukleme
tce-load -w -i openssh.tcz
# ssh/sshd ayarlari
cp /usr/local/etc/ssh/ssh_config.orig /usr/local/etc/ssh/ssh_config
cp /usr/local/etc/ssh/sshd_config.orig /usr/local/etc/ssh/sshd_config
echo "Port 22" >> /usr/local/etc/ssh/sshd_config
echo "PasswordAuthentication yes" >> /usr/local/etc/ssh/sshd_config
# servisi calistirma
/usr/local/etc/init.d/openssh start
```

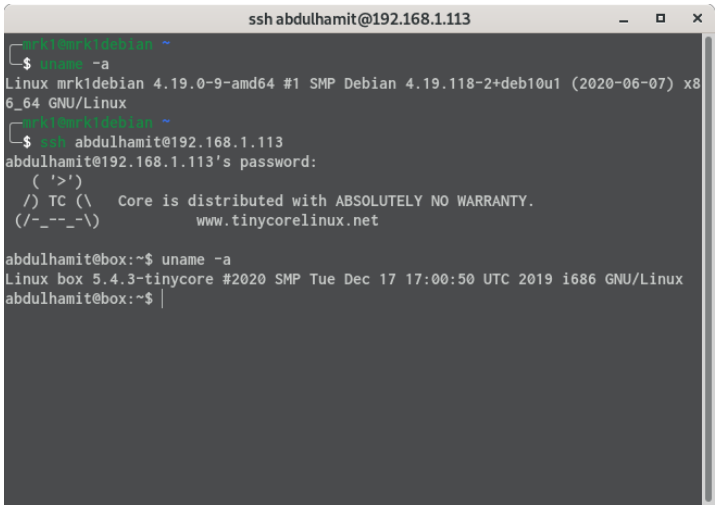

Figure 11: Gueste ssh bağlantısı

Ping

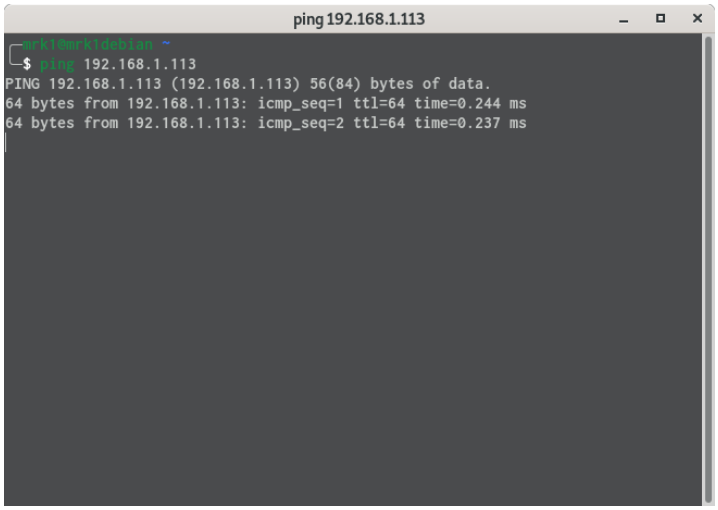

Figure 12: Gueste ping gönderebiliyoruz

Host only Adapter ile Host, Ubuntu Server Guest Bağlantısı

Host only Adapter kullanabilmek için öncelikle Host Network oluşturmak gerekli. Ana Menüden `File > Host Network Manager...`i seçip create tıklıyoruz ve Network Oluşuyor.

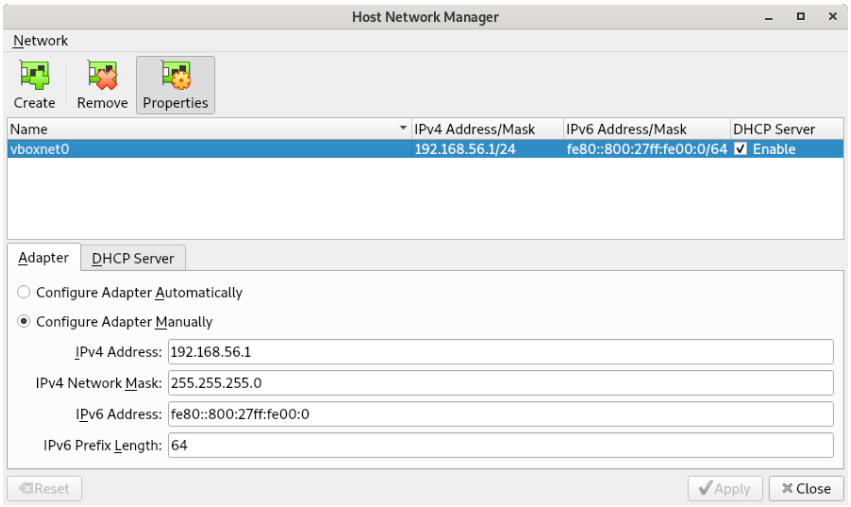

Figure 13: Host Network

Guestimizin Network adapterini Host only Adapter seçip alt seçenekten oluşturduğumuz Host Network'ü seçiyoruz.

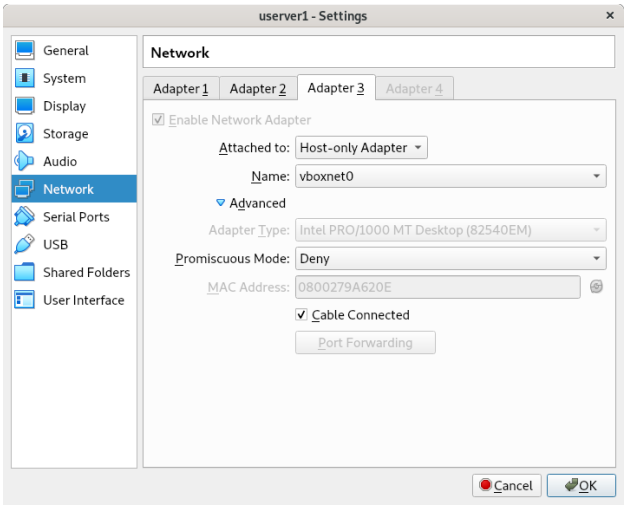

Figure 14: Host Only Adapter Ayarı

yaml fromatındaki `/etc/netplan/50-cloud-init.yaml` dosyasını Network Interfacemizi DHCP ile yönetilmesi için ayarlıyoruz

```
# /etc/netplan/50-cloud-init.yaml
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s9:
      dhcp4: yes
```

Yeni ayarların kullanması için netplan komutunu çalıştırıyoruz.

```
$ sudo netplan --debug apply
```

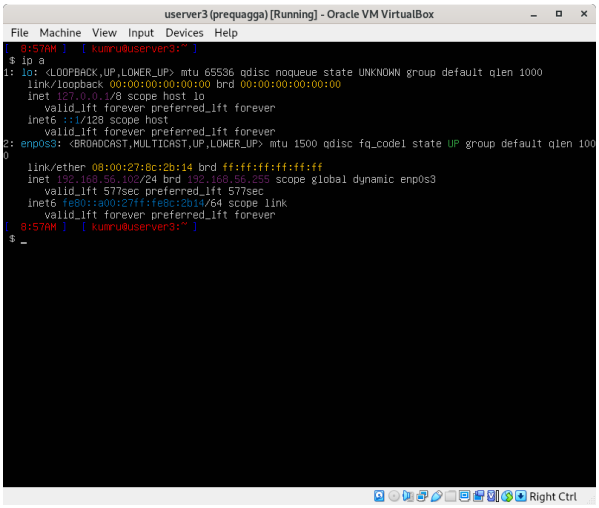

Figure 15: Host-only Adapterin Host Networkten aldığı IP.

SSH

NAT ile Host, Guest Ubuntu PC Bağlantısı bölümünde yaptığımız komutları burada tekrar ediyoruz.

```
$ apt install opessh-server
$ echo "Port 22" >> /etc/ssh/sshd_config
$ echo "PasswordAuthentication yes" >> /etc/ssh/sshd_config
$ sudo service ssh restart
```

userver1'e SSH ile bağlanıyoruz.

```
$ ssh kumru@192.168.56.101
```

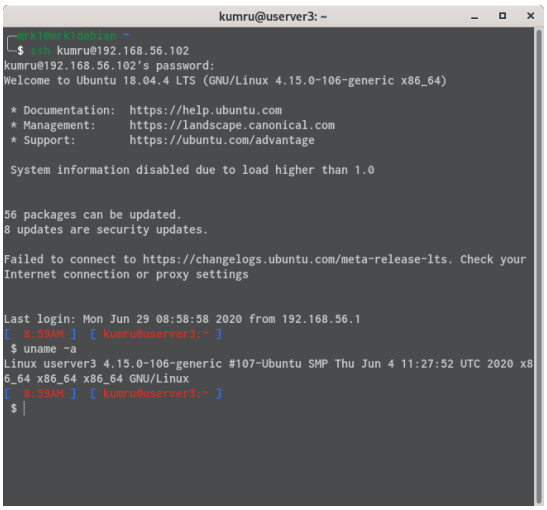

Figure 16: Gueste ssh ile bağlanma

Ping

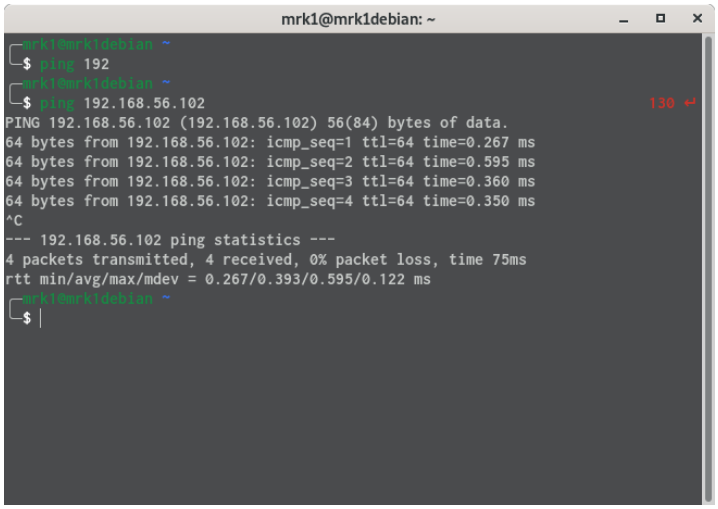

Figure 17: Gueste Ping gönderme

Senaryo 2

Internal Network ile Host-Guest, Guest-Guest Bağlatısı

Internal Network Ayarı

Senaryo geregi ip Ubuntu Serverlarin ipleri

`host: 192.168.0.1 userver1: 192.168.0.2 userver2: 192.168.0.3 userver3: 192.168.0.4`

Internal network için Network Adapterin modunu seçmek yeterli.

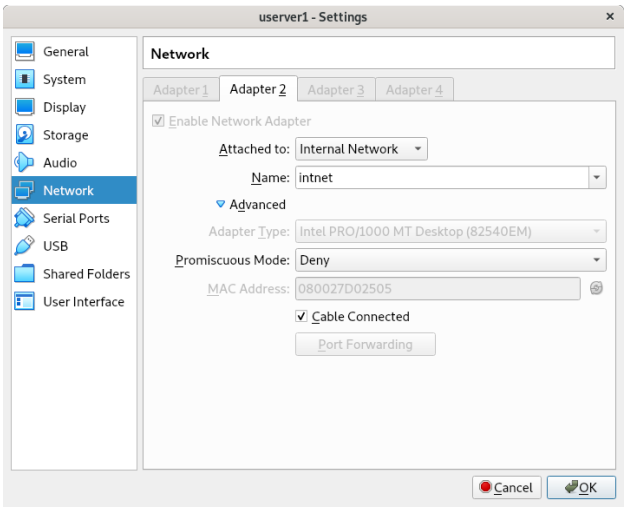

Figure 18: Internal Network Ayarları

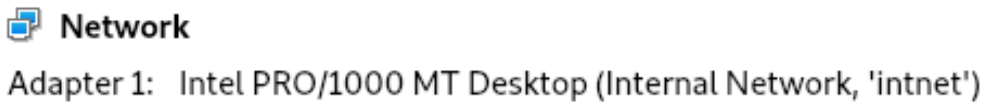

Figure 19: userver1

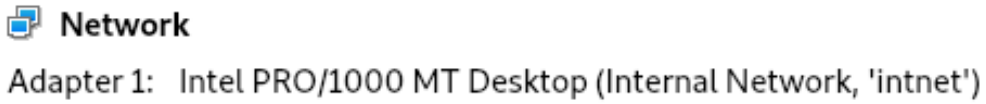

Figure 20: userver2

Figure 21: userver3

Netplan Ayarları

netplan ayarlarını Senaryo 1 de yaptığımız gibi yapıyoruz. Farklı olarak dhcp olmadan statik bir şekilde IP alıyoruz

- userver1

```
# /etc/netplan/50-cloud-init.yaml
# S:0 R:2 userver1
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s8:
      dhcp4: no
      addresses:
        - 192.168.0.2/24
      gateway4: 192.168.0.1
```

- userver2

```
# /etc/netplan/50-cloud-init.yaml
# S:0 R:2 userver2
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s8:
      dhcp4: no
      addresses:
        - 192.168.0.3/24
      gateway4: 192.168.0.1
```

- userver3

```
# /etc/netplan/50-cloud-init.yaml
# S:0 R:2 userver3
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s8:
      dhcp4: no
      addresses:
        - 192.168.0.4/24
      gateway4: 192.168.0.1
```

SSH

SSH server yüklediğimiz için ve diğer makineler klon olduğu için bir daha SSH yükleme ve ayarlamaya gerek yok. Guestler arasında SSH bağlantısı yapılabiliyor.

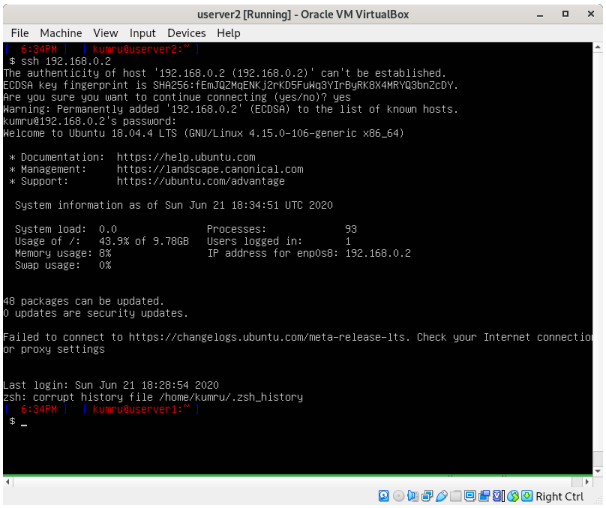

Figure 22: Guestler Arasında SSH

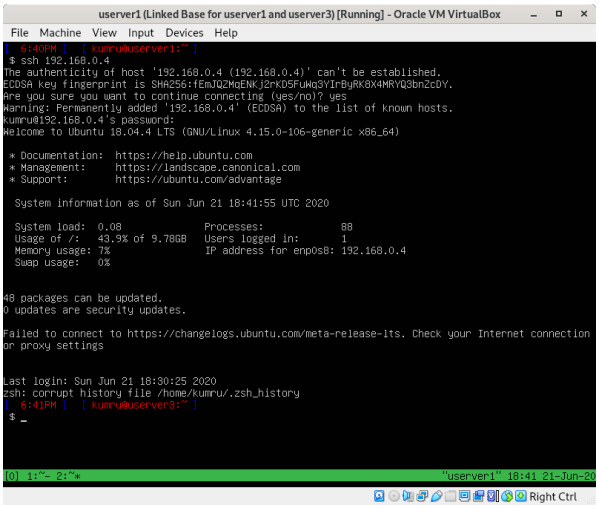

Figure 23: Guestler Arasında SSH

Internal Networkte hostdan guestlere ulaşmak mümkün değil.

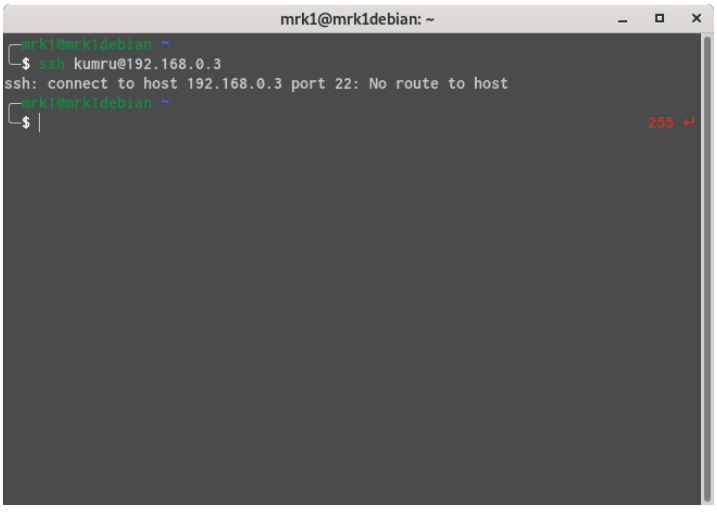

Figure 24: Hosttan Gueste SSH denemesi

Ping

Guestler birbirlerine ulaşabildiği için birbirlerine ping göndermeleri mümkün.

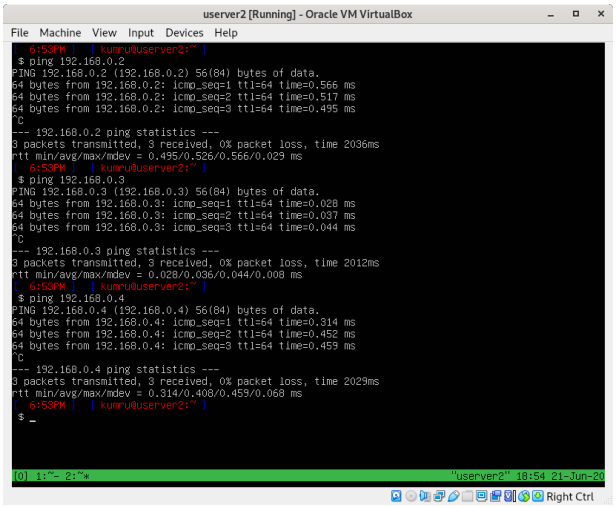

Figure 25: Guestler Arası Ping

SCP

Gerek olmasada boş dosya göndermek yerine sıfır yazılmış 100M büyüklüğünde dosyalar gönderelim.

```
# dosyalari hazirlama
$ dd if=/dev/zero of=abdulhamit.txt count=100 bs=1M
$ cat abdulhamit.txt > kumru.txt > 170202020.txt
# scp komutu
$ scp *.txt 192.168.0.3:/home/kumru/
```

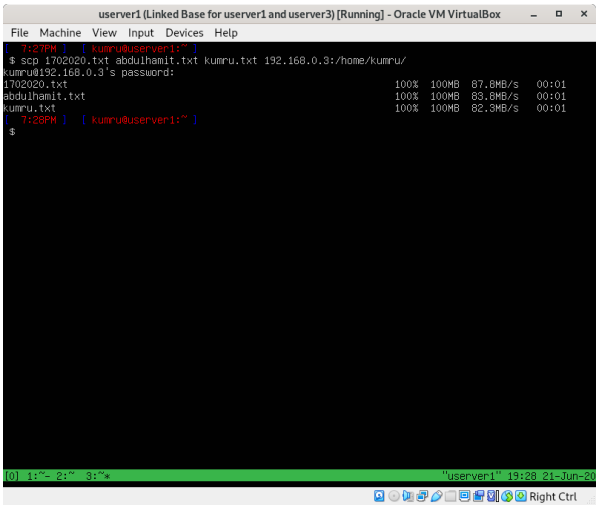

Figure 26: scp Komutu Çıktısı

Senaryo 3

Bu senaryoda bir ağdan diğer ağa routing yapmamız istenmekte. Router yazılımı olarak Quaggayı kullanacağız.

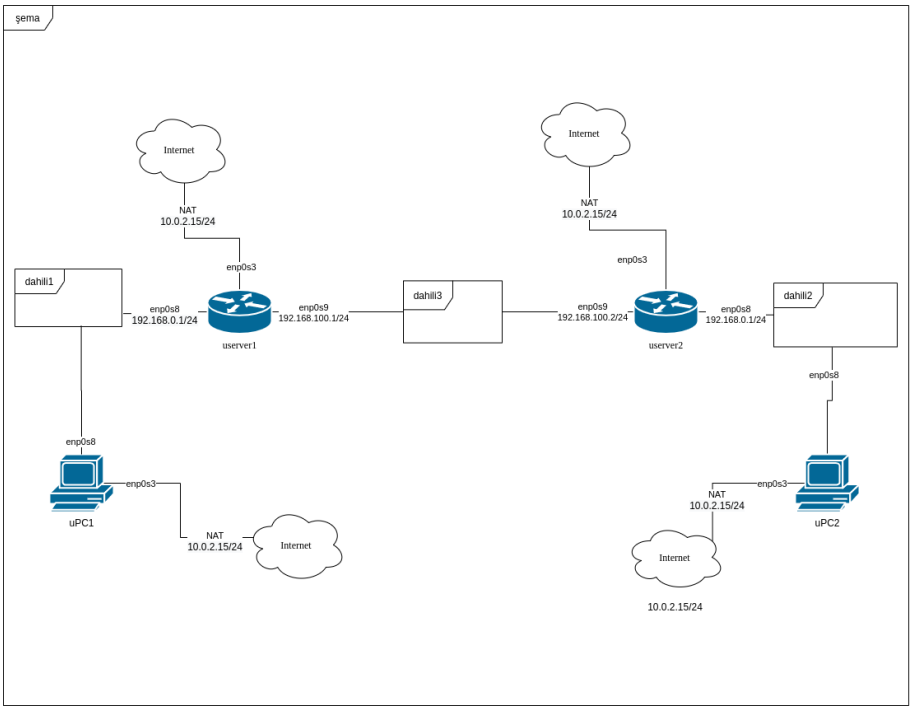

Figure 27: Şema

- dahili1
  `uPC1 192.168.1.1`
  `userver1 192.168.1.254`
- dahili2
  `uPC2 192.168.2.1`
  `userver2 192.168.2.254`
- dahili 3 `userver1 192.168.100.1`
  `userver2 192.168.100.2`

Network Ayarları

Virtual Box Network Ayarları

Serverlardaki adapter4 hariç diğer adaptörler senaryo gerekleri için kullandık. Adapter4 ü ise sunucuları yönetmek için kullandık.

Figure 28: uPC1

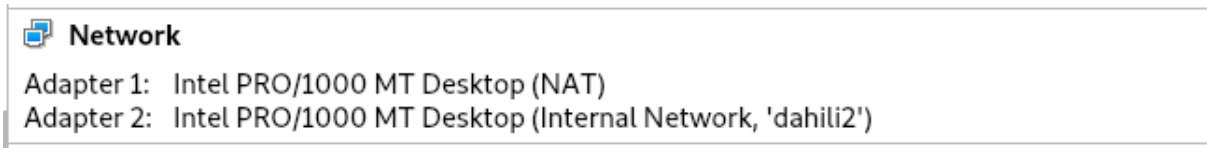

Figure 29: uPC2

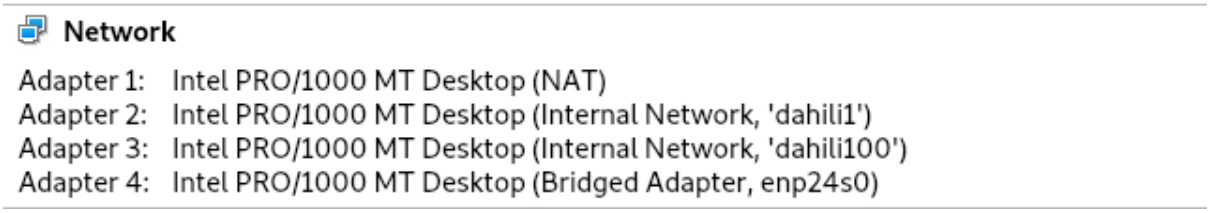

Figure 30: userver1

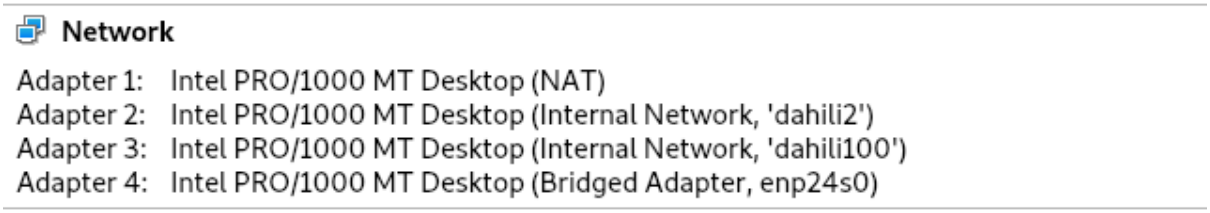

Figure 31: userver2

PC'lerin netplan Ayarları

uPC1 netplan ayarları

```
# uPC1
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s3:
      dhcp4: yes
    enp0s8:
      dhcp4: no
      addresses:
      - 192.168.1.1/24
      gateway4: 192.168.1.254
      nameservers:
          addresses: [8.8.8.8, 1.1.1.1]
      routes:
      - to: 192.168.0.0/16
        via: 192.168.1.254
```

uPC2 netplan ayarları

```
# uPC2
network:
  version: 2
  renderer: networkd
```

```
  ethernets:
    enp0s3:
      dhcp4: yes
    enp0s8:
      dhcp4: no
      addresses:
        - 192.168.2.1/24
      gateway4: 192.168.2.254
      nameservers:
          addresses: [8.8.8.8, 1.1.1.1]
      routes:
      - to: 192.168.0.0/16
        via: 192.168.1.254
```

Server'lerin Network Ayarları

Serverlerin nat ayarları

```
# serverler için ortak nat ayarı
network:
  ethernets:
    enp0s3:
      dhcp4: true
  version: 2
```

userver2

```
# userver2
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s8:
      dhcp4: no
      addresses:
        - 192.168.0.1/24
      #gateway4: 192.168.0.1
      nameservers:
        addresses: [8.8.8.8, 8.8.4.4]
    enp0s9:
      dhcp4: no
      addresses:
        - 192.168.100.2/24
      #gateway4: 192.168.0.1
      nameservers:
        addresses: [8.8.8.8, 8.8.4.4]
    enp0s10:
      dhcp4: no
      addresses:
        - 192.168.1.201/24
      #gateway4: 192.168.0.1
      nameservers:
        addresses: [8.8.8.8, 8.8.4.4]
```

userver1

```
# userver1
network:
  version: 2
  renderer: networkd
  ethernets:
```

```
    enp0s8:
      dhcp4: no
      addresses:
        - 192.168.0.1/24
      #gateway4: 192.168.0.1
      nameservers:
        addresses: [8.8.8.8, 8.8.4.4]
    enp0s9:
      dhcp4: no
      addresses:
        - 192.168.100.1/24
      #gateway4: 192.168.0.1
      nameservers:
        addresses: [8.8.8.8, 8.8.4.4]
    enp0s10:
      dhcp4: no
      addresses:
        - 192.168.1.200/24
      #gateway4: 192.168.0.1
      nameservers:
        addresses: [8.8.8.8, 8.8.4.4]
```

Serverlera Quagga Kurulumu ve Ayarları

Buradaki bilgiler brianlinkletter.com , ixnfo.com dan alınmıştır.

Quagga Kurulum Scripti

```
#!/bin/bash
# quagga installer
if [ "$EUID" -ne 0 ]
  then echo "Please run as root"
  exit
fi

sudo apt install quagga quagga-doc
sudo cat > /etc/quagga/daemons << EOF
zebra=yes
bgpd=no
ospfd=yes
ospf6d=no
ripd=no
ripngd=no
isisd=no
babeld=no
EOF

echo "net.ipv4.ip_forward=1" >> /etc/sysctl.conf
cp /usr/share/doc/quagga-core/examples/vtysh.conf.sample /etc/quagga/vtysh.conf
cp /usr/share/doc/quagga-core/examples/zebra.conf.sample /etc/quagga/zebra.conf
cp /usr/share/doc/quagga-core/examples/bgpd.conf.sample /etc/quagga/bgpd.conf
chown quagga:quagga /etc/quagga/*.conf
chown quagga:quaggavty /etc/quagga/vtysh.conf
chmod 640 /etc/quagga/*.conf
service zebra start
service bgpd start
systemctl enable zebra.service
systemctl enable bgpd.service
echo 'VTYSH_PAGER=more' >>/etc/environment
```

```
echo 'export VTYSH_PAGER=more' >>/etc/bash.bashrc
```

Quagga Ayar Scriptleri

- userver1

```
#!/bin/bash
if [ "$EUID" -ne 0 ]
  then echo "Please run as root"
  exit
fi
cat >> /etc/quagga/ospfd.conf << EOF
interface enp0s8
interface enp0s9
interface lo
router ospf
 passive-interface enp0s8
 network 192.168.1.0/24 area 0.0.0.0
 network 192.168.100.0/24 area 0.0.0.0
line vty
EOF
cat >> /etc/quagga/zebra.conf << EOF
interface enp0s8
 ip address 192.168.1.254/24
 ipv6 nd suppress-ra
interface enp0s9
 ip address 192.168.100.1/24
 ipv6 nd suppress-ra
interface lo
ip forwarding
line vty
EOF
sudo service zebra restart
sudo service bgpd restart
```

- userver2

```
#!/bin/bash
if [ "$EUID" -ne 0 ]
  then echo "Please run as root"
  exit
fi
cat >> /etc/quagga/ospfd.conf << EOF
interface enp0s8
interface enp0s9
interface lo
router ospf
 passive-interface enp0s8
 network 192.168.2.0/24 area 0.0.0.0
 network 192.168.100.0/24 area 0.0.0.0
line vty
EOF
cat >> /etc/quagga/zebra.conf << EOF
interface enp0s8
 ip address 192.168.2.254/24
 ipv6 nd suppress-ra
interface enp0s9
 ip address 192.168.100.2/24
```

```
 ipv6 nd suppress-ra
interface lo
ip forwarding
line vty
EOF
sudo service zebra restart
sudo service bgpd restart
```

Ayar sontrası cihazların durumu

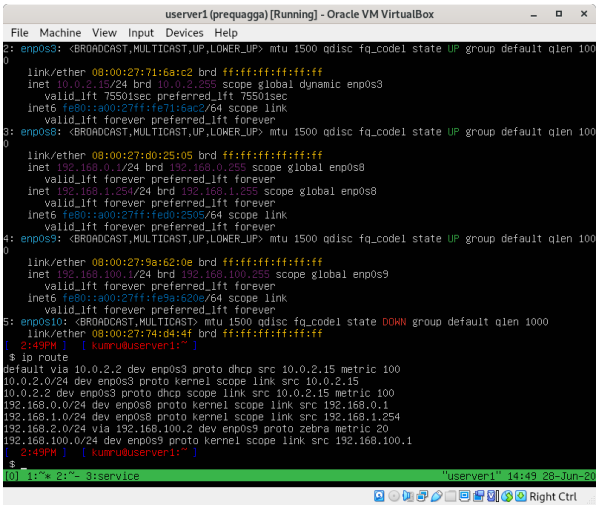

Figure 32: Ubuntu Server1 ip a, ip route Komut çıktısı

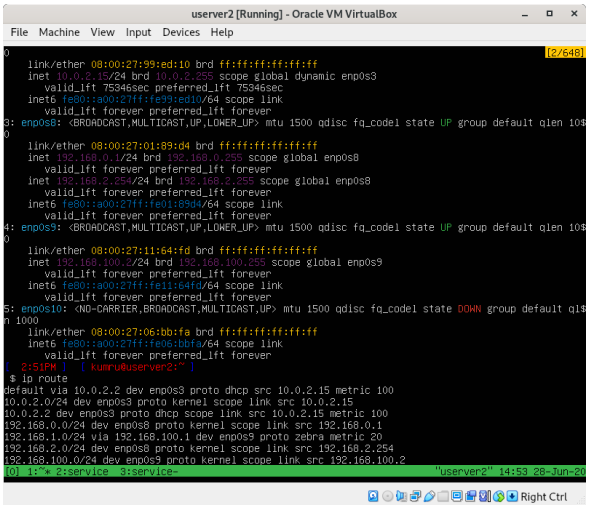

Figure 33: Ubuntu Server2 ip a, ip route Komut çıktısı

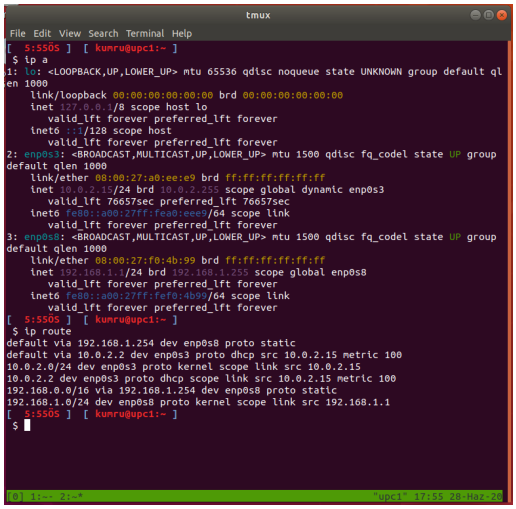

Figure 34: Ubuntu Desktop1 (uPC1) ip a, ip route Komut çıktısı

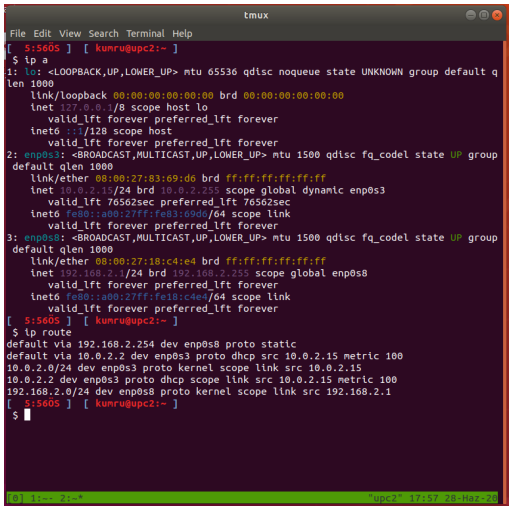

Figure 35: Ubuntu Desktop2 (uPC2) ip a, ip route Komut çıktısı

traceroute, paket takibi

Traceroute komutu ile paketlerin izlediği yolun takibini yapabiliriz.

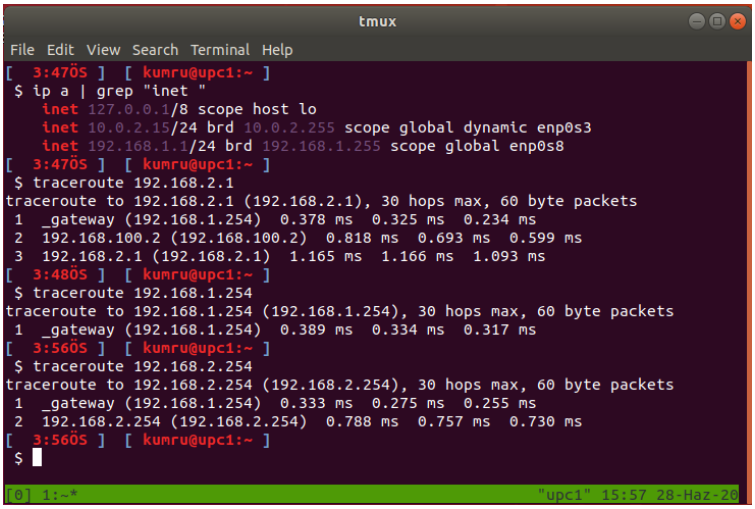

Figure 36: uPC1 -> uPC2, uPC1 -> userver1, uPC1 -> userver2