

Linux Ağ Yönetimi Final Projesi

Abdulhamit Kumru 170202020

Makinelerin Klonlanması

!!! makinelerin klonlanmasını ekle !!! programlar hakkında metadata ekle

Senaryo 1

NAT ile Host-Guest Ubuntu PC Bağlantısı

Başlangıç olarak guest makinenin Network Adapterini NAT olarak seçiyoruz

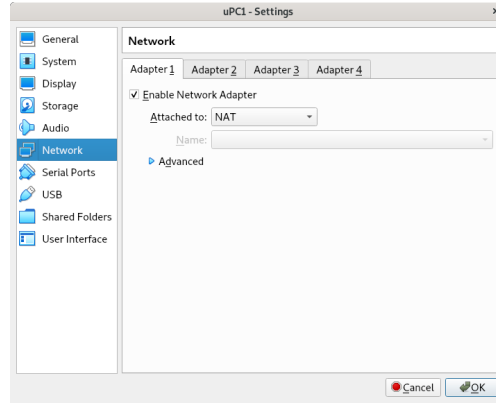


Figure 1: NAT adapter ayarı

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.

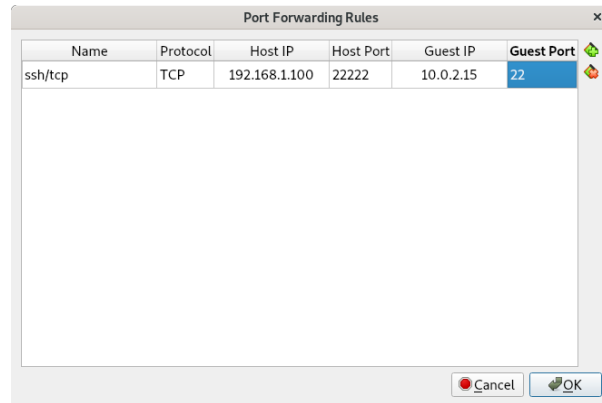


Figure 2: Port forwarding

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

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

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

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

Ayarların aktif olması için ssh servisimizi yeniden başlatıyoruz.

```
$ sudo service ssh restart
```

Forward ettiğimiz port üzerinden guest'e bağlanabiliriz.

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

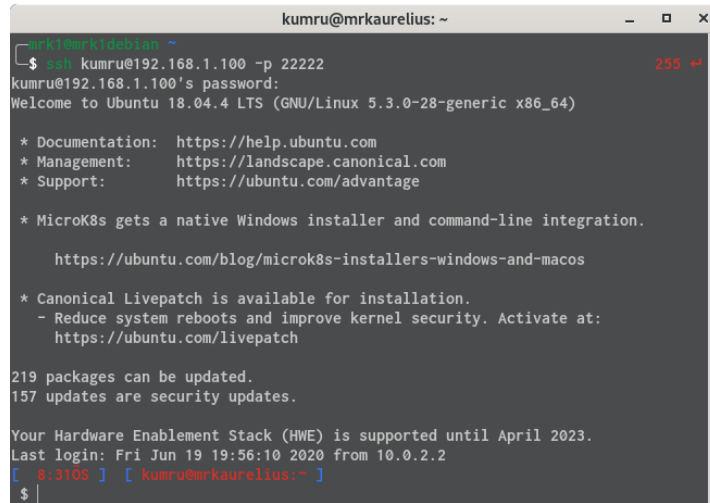


Figure 3: Gueste ssh ile baglanma

Ping

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

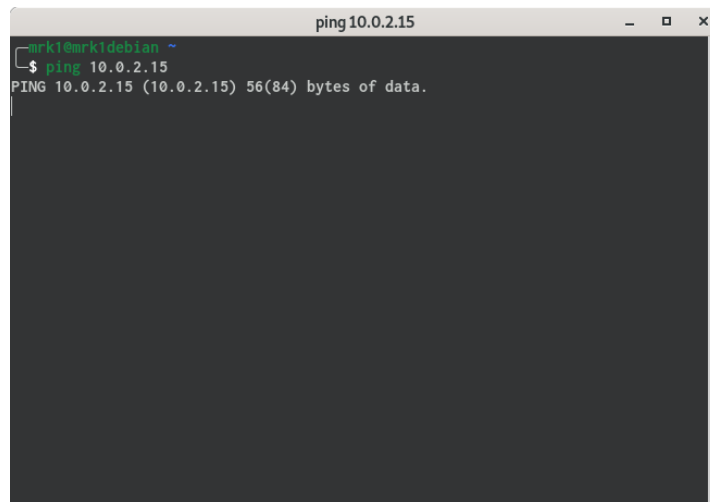


Figure 4: Gueste ping gönderilemiyor

Misafir Eklentileri

Misafir eklentilerini guest makineye sanal disk takarak yükleyebiliyoruz bu sayade hosttan gueste veya guestten hosta kopyala yapıdır ve dosya sürekli bırak yapabiliyoruz. Eklentiye yüklemek zor değil sanal makine penceresinden **Devices > Insert Guest Additions CD...** seçeneğini seçince bize `autorun.sh`'i çalıştırırım mı diye soruyor evet diyip parolamızı girince eklentiler yükleniyor

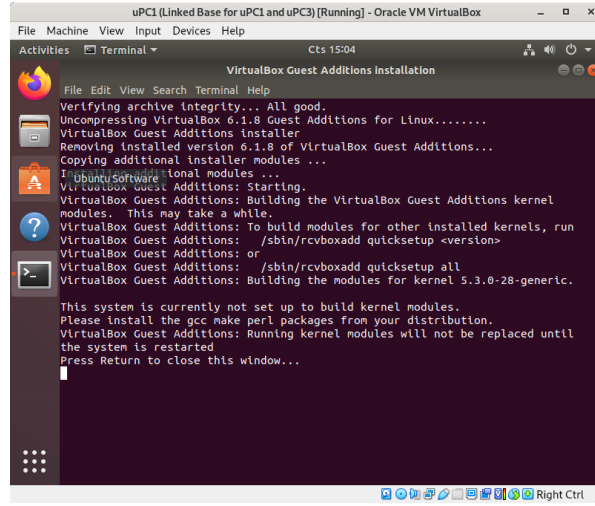


Figure 5: autorun.sh çıktısı

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

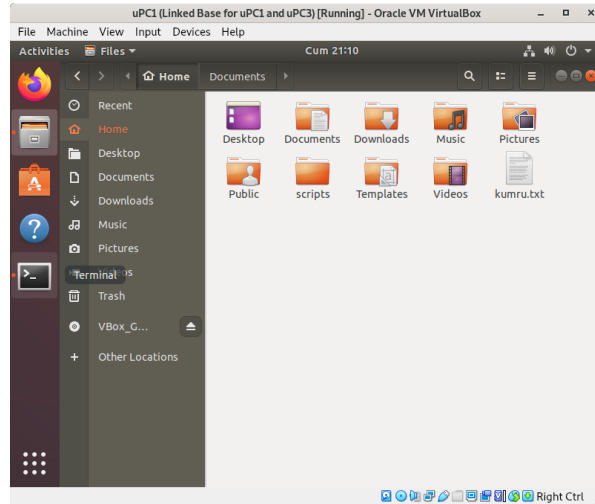


Figure 6: Gueste eklentileri ile sürükle bırak

Network Bridge ile Host, TinyCore Guest Bağlantısı

Guestin network adapter ayarını Bridge Adapter olarak seçiyoruz

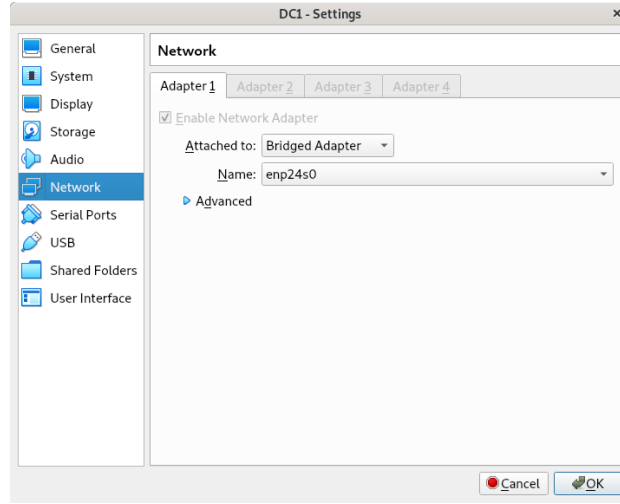


Figure 7: Network Bridge Ayarı

TinyCore'da IP'mizi ifconfig komutu ile kontrol edebiliyoruz.

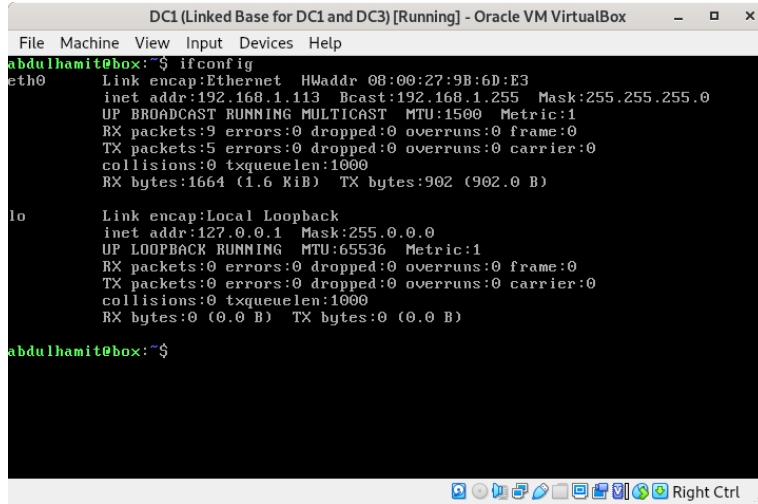


Figure 8: ifconfig çıktısı

Network Bridge ile Guest LAN'daki makine gibi kullanılabilir.

SSH

tce-load programı ile TinyCore'da program yükleyebiliyoruz.

```
# openssh client/server paketini yükleme
tce-load -w -i openssh.tcz
# ssh/sshd ayarları
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
```

```
ssh abdulhamit@192.168.1.113
mrklmrkldebian ~
$ uname -a
Linux mrklmrkldebian 4.19.0-9-amd64 #1 SMP Debian 4.19.118-2+deb10u1 (2020-06-07) x86_64 GNU/Linux
mrklmrkldebian ~
$ ssh abdulhamit@192.168.1.113
abdulhamit@192.168.1.113's password:
( '>')
/) TC (\   Core is distributed with ABSOLUTELY NO WARRANTY.
(/_--_\)   www.tinycorelinux.net

abdulhamit@box:~$ uname -a
Linux box 5.4.3-tinycore #2020 SMP Tue Dec 17 17:00:50 UTC 2019 i686 GNU/Linux
abdulhamit@box:~$
```

Figure 9: Gueste ssh bağlantısı

Ping

```
ping 192.168.1.113
mrklmrkldebian ~
$ ping 192.168.1.113
PING 192.168.1.113 (192.168.1.113) 56(84) bytes of data.
64 bytes from 192.168.1.113: icmp_seq=1 ttl=64 time=0.244 ms
64 bytes from 192.168.1.113: icmp_seq=2 ttl=64 time=0.237 ms
```

Figure 10: 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.

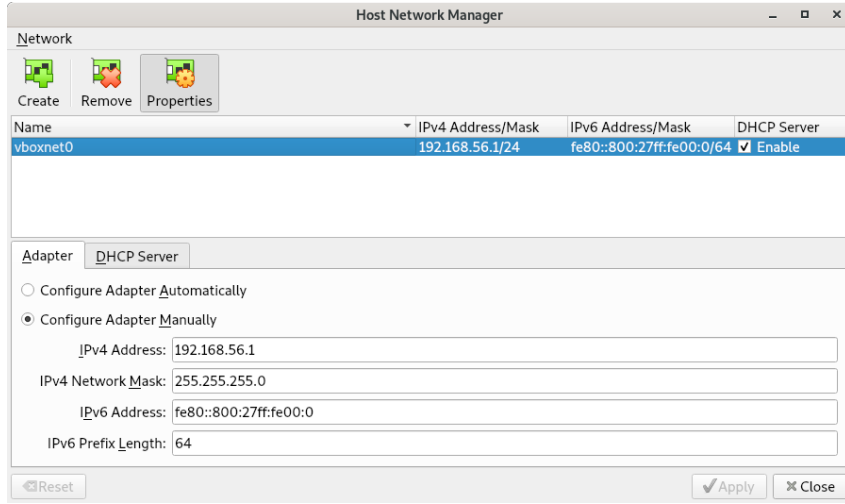


Figure 11: Host Network

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

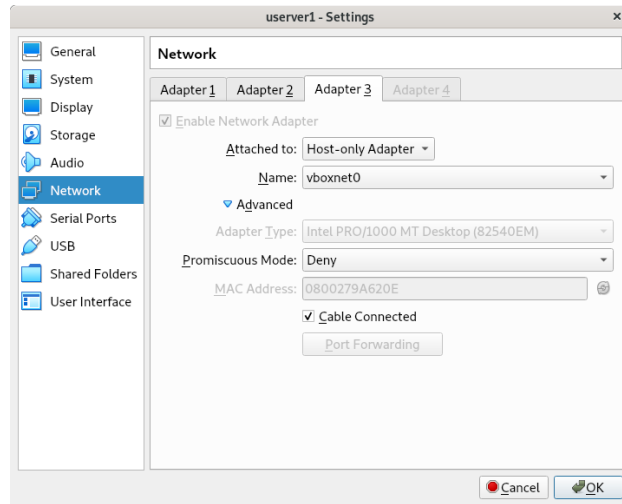


Figure 12: Host Only Adapter Ayarı

yaml formatındaki `/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
```

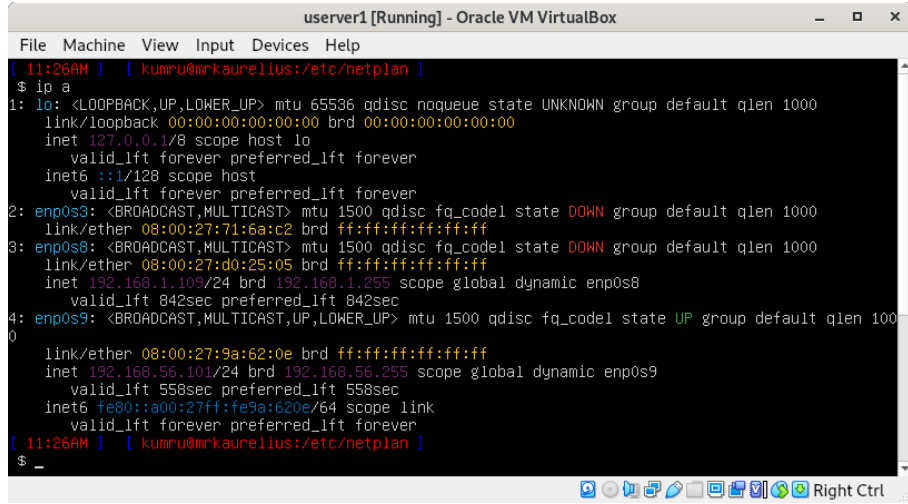


Figure 13: Host-only Adapterin Host Networkten aldığı IP.

SSH

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

```
$ apt install openssh-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
```

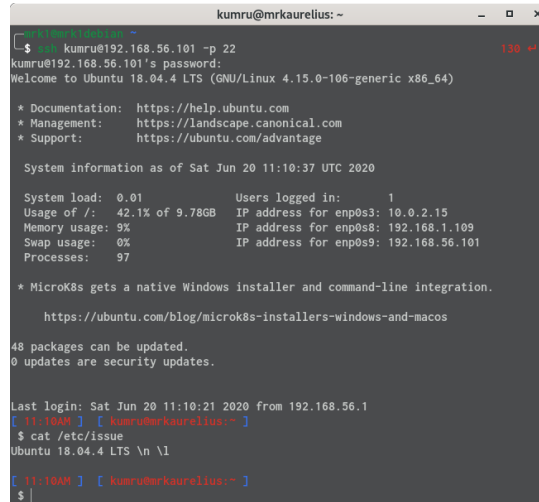


Figure 14: Gueste ssh ile bağlanma

Ping


```
mrk1@mrk1debian: ~  
$ ping 192.168.56.101  
PING 192.168.56.101 (192.168.56.101) 56(84) bytes of data.  
64 bytes from 192.168.56.101: icmp_seq=1 ttl=64 time=0.322 ms  
64 bytes from 192.168.56.101: icmp_seq=2 ttl=64 time=0.267 ms  
64 bytes from 192.168.56.101: icmp_seq=3 ttl=64 time=0.320 ms  
64 bytes from 192.168.56.101: icmp_seq=4 ttl=64 time=0.255 ms  
^C  
--- 192.168.56.101 ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 82ms  
rtt min/avg/max/mdev = 0.255/0.291/0.322/0.030 ms  
mrk1@mrk1debian: ~  
$
```

Figure 15: Geste Ping gönderme

Senaryo 2

Internal Network ile Host-Guest, Guest-Guest Bağlatısı

Internal Network Ayarı

Senaryo gereği ip Ubuntu Serverların 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.

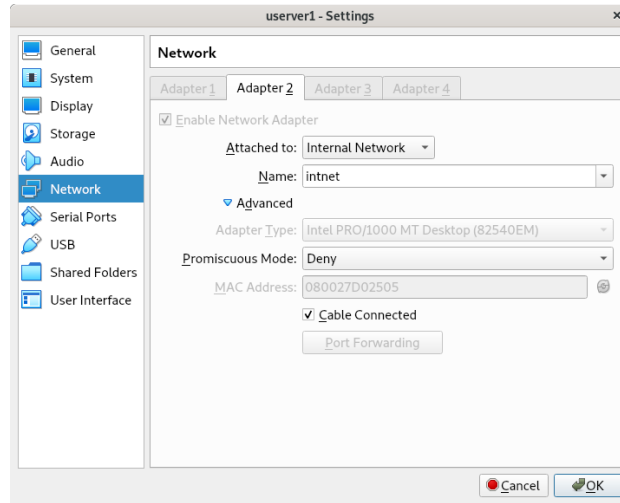


Figure 16: Internal Network Ayarları

Netplan Ayarları

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

```
# /etc/netplan/50-cloud-init.yaml
# S:0 R:2
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s8:
      dhcp4: no
      addresses:
        - 192.168.0.2/24 # userver1 için
      gateway4: 192.168.0.1
      nameservers:
        # aslında nameserver ayarlamasının anlamı yok ama adet yerini bulsun
        addresses: [8.8.8.8, 1.1.1.1]
```

SSH

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

```
userver2 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
6:34PM | kumru@userver2:~$ ssh 192.168.0.2
The authenticity of host '192.168.0.2 (192.168.0.2)' can't be established.
ECDSA key fingerprint is SHA256:fEn1Q2McENj2KX05Fuk03Y1nByRk8X4MAYQ3bnZc0Y.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.0.2' (ECDSA) to the list of known hosts.
kumru@192.168.0.2's password:
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-106-generic x86_64)

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

System information as of Sun Jun 21 18:34:51 UTC 2020

System load: 0.0          Processes:           93
Usage of /:  43.9% of 9.7GB Users logged in:       1
Memory usage: 6%         IP address for enp0s8: 192.168.0.2
Swap usage:  0%

48 packages can be updated.
0 updates are security updates.

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Sun Jun 21 18:28:54 2020
zsh: corrupt history file /home/kumru/.zsh_history
6:34PM | kumru@userver1:~$
```

Figure 17: Guestler Arasında SSH

```
userver1 (Linked Base for userver1 and userver3) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
6:40PM | kumru@userver1:~$ ssh 192.168.0.4
The authenticity of host '192.168.0.4 (192.168.0.4)' can't be established.
ECDSA key fingerprint is SHA256:fEn1Q2McENj2KX05Fuk03Y1nByRk8X4MAYQ3bnZc0Y.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.0.4' (ECDSA) to the list of known hosts.
kumru@192.168.0.4's password:
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-106-generic x86_64)

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

System information as of Sun Jun 21 18:41:55 UTC 2020

System load: 0.08         Processes:           88
Usage of /:  43.9% of 9.7GB Users logged in:       1
Memory usage: 7%         IP address for enp0s8: 192.168.0.4
Swap usage:  0%

48 packages can be updated.
0 updates are security updates.

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Sun Jun 21 18:30:25 2020
zsh: corrupt history file /home/kumru/.zsh_history
6:41PM | kumru@userver3:~$
```

Figure 18: Guestler Arasında SSH

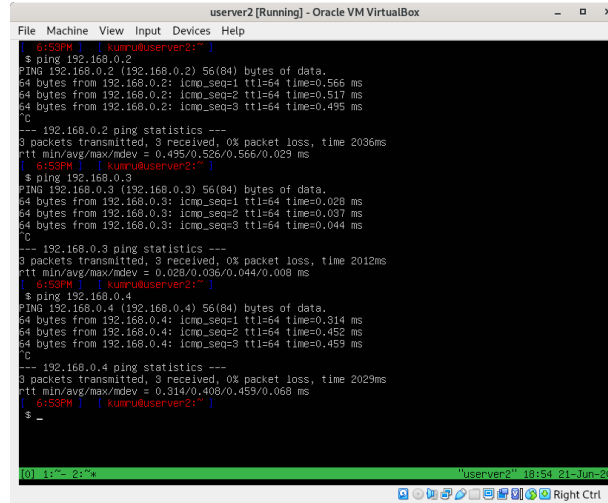
Internal Networkte hostdan guestlere ulaşmak mümkün değil.

```
mrk1@mrk1debian: ~$ ssh kumru@192.168.0.3
ssh: connect to host 192.168.0.3 port 22: No route to host
mrk1@mrk1debian: ~$
```

Figure 19: Hosttan Geste SSH denemesi

Ping

Guestler birbirlerine ulaşabildiği için birbirlerine ping göndermeleri mümkün.



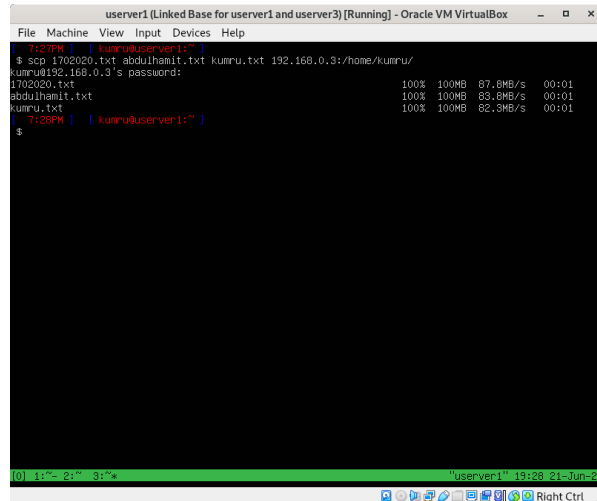
```
userver2 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
$ ping 192.168.0.2
PING 192.168.0.2 (192.168.0.2) 56(84) bytes of data:
64 bytes from 192.168.0.2: icmp_seq=1 ttl=64 time=0.566 ms
64 bytes from 192.168.0.2: icmp_seq=2 ttl=64 time=0.517 ms
64 bytes from 192.168.0.2: icmp_seq=3 ttl=64 time=0.495 ms
--- 192.168.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2036ms
rtt min/avg/max/mdev = 0.495/0.526/0.566/0.029 ms
$ ping 192.168.0.3
PING 192.168.0.3 (192.168.0.3) 56(84) bytes of data:
64 bytes from 192.168.0.3: icmp_seq=1 ttl=64 time=0.028 ms
64 bytes from 192.168.0.3: icmp_seq=2 ttl=64 time=0.037 ms
64 bytes from 192.168.0.3: icmp_seq=3 ttl=64 time=0.044 ms
--- 192.168.0.3 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2012ms
rtt min/avg/max/mdev = 0.028/0.036/0.044/0.008 ms
$ ping 192.168.0.4
PING 192.168.0.4 (192.168.0.4) 56(84) bytes of data:
64 bytes from 192.168.0.4: icmp_seq=1 ttl=64 time=0.314 ms
64 bytes from 192.168.0.4: icmp_seq=2 ttl=64 time=0.452 ms
64 bytes from 192.168.0.4: icmp_seq=3 ttl=64 time=0.459 ms
--- 192.168.0.4 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2029ms
rtt min/avg/max/mdev = 0.314/0.408/0.459/0.068 ms
$ _
```

Figure 20: Guestler Arası Ping

SCP

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

```
# dosyaları hazırlama
$ dd if=/dev/zero of=abdulhamit.txt count=100 bs=1M
$ cat abdulhamit.txt > kumru.txt > 1702020.txt
# scp komutu
$ scp *.txt 192.168.0.3:/home/kumru/
```



```
userver1 (Linked Base for userver1 and userver3) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
$ scp 1702020.txt abdulhamit.txt kumru.txt 192.168.0.3:/home/kumru/
kumru192.168.0.3's password:
1702020.txt          100% 100MB 87.8MB/s   00:01
abdulhamit.txt       100% 100MB 83.8MB/s   00:01
kumru.txt            100% 100MB 82.3MB/s   00:01
$
```

Figure 21: scp Komutu Çıktısı

Senaryo 3

Network Ayarları

Virtual Box Ayarları

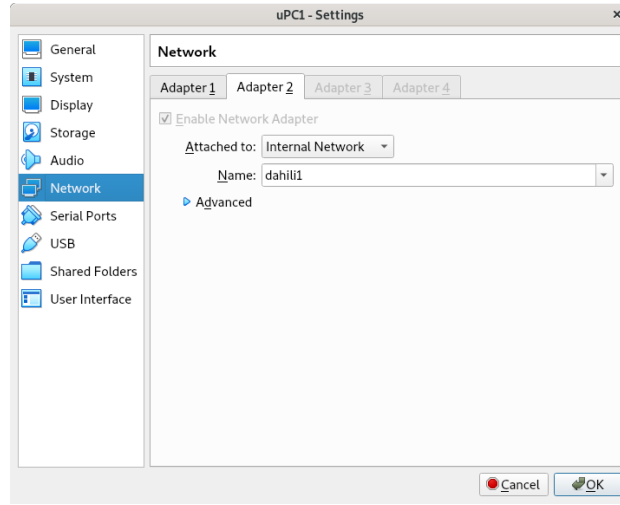


Figure 22: Internal Network Adapter

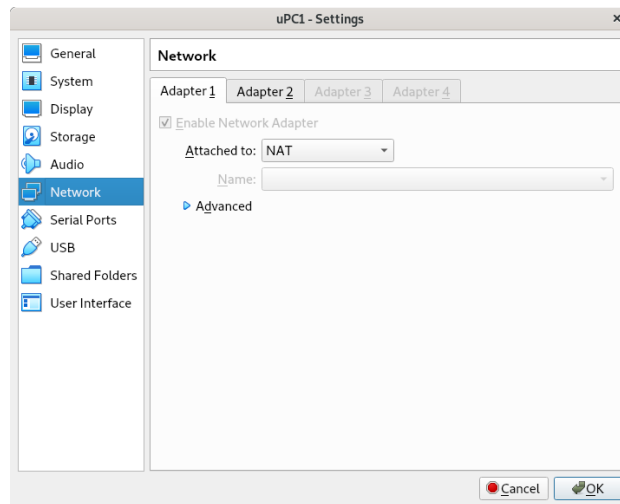


Figure 23: NAT Adapter

Network Şeması

dahili1

uPC1 NAT dhcp, dahili1 192.168.0.5

userver2 NAT dhcp, dahili1 192.168.0.3

dahili2

uPC2 NAT, dahili2 192.168.0.6

userver1 NAT, dahili2 192.168.0.2

Netplan Ayarları

Ubuntu Server için netplan ayarı

Internal Network için 50-cloud-init.yaml dosyasını kullandık.

```
# /etc/netplan/50-cloud-init.yaml
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s8:
      dhcp4: no
      addresses:
        - 192.168.0.2/24
      #gateway4: 192.168.0.1
      nameservers:
        # aslında nameserver ayarlamamızın anlamı yok ama adet yerini bulsun
        addresses: [8.8.8.8, 1.1.1.1]
```

NAT için yüklemeye oluşturulan 00-installer-config.yaml dosyasını kullandık.

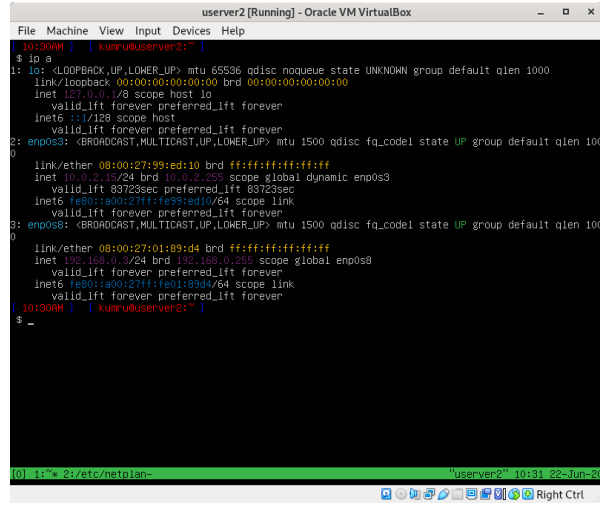
```
# /etc/netplan/00-installer-config.yaml
network:
  ethernets:
    enp0s3:
      dhcp4: yes
  version: 2
```

Ubuntu Desktop için netplan ayarı.

```
# /etc/netplan/01-network-manager-all.yaml
# Let NetworkManager manage all devices on this system
network:
  version: 2
  renderer: NetworkManager
  ethernets:
    enp0s3:
      dhcp4: yes
    enp0s8:
      dhcp4: no
      addresses:
        - 192.168.0.5/24
      #gateway4: 192.168.0.1
      nameservers:
        addresses: [8.8.8.8, 1.1.1.1]
```

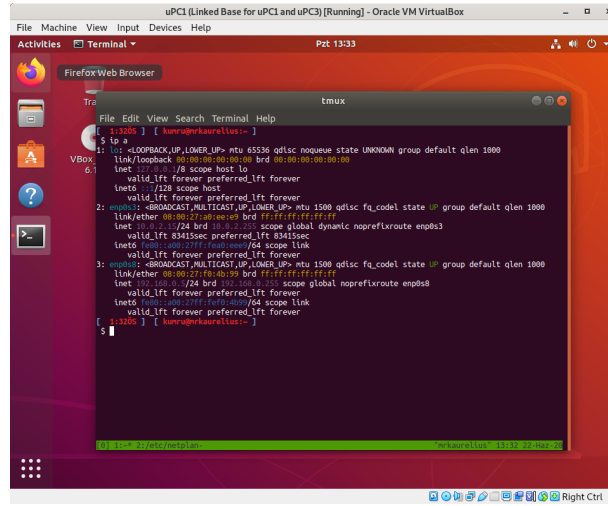
Yapılan ayarları uygulama.

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



```
userver2 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
[ 10:30AM ] [ kunru@userver2:~ ]
$ ip a
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: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:99:ed:10 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.15/24 brd 192.168.0.255 scope global dynamic enp0s3
        valid_lft 83723sec preferred_lft 83723sec
    inet6 fe80::a00:27ff:fe99:ed10/64 scope link
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:01:89:d4 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.1/24 brd 192.168.0.255 scope global enp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe01:89d4/64 scope link
        valid_lft forever preferred_lft forever
[ 10:30AM ] [ kunru@userver2:~ ]
$ _
```

Figure 24: Ubuntu Server ip Komut çıktısı



```
uPC1 [Linked Base for uPC1 and uPC3] [Running] - Oracle VM VirtualBox
File Edit View Search Terminal Help
[ 11:32AM ] [ kunru@uPC1:~ ]
$ ip a
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: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:99:ed:10 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.15/24 brd 192.168.0.255 scope global dynamic noprefixroute enp0s3
        valid_lft 83415sec preferred_lft 83415sec
    inet6 fe80::a00:27ff:fe99:ed10/64 scope link
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:01:89:d4 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.1/24 brd 192.168.0.255 scope global noprefixroute enp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe01:89d4/64 scope link
        valid_lft forever preferred_lft forever
[ 11:32AM ] [ kunru@uPC1:~ ]
$
```

Figure 25: Ubuntu Desktop ip Komut çıktısı

traceroute

Traceroute komutu ile paketlerin izlediği yolun takibini yapabiliriz.

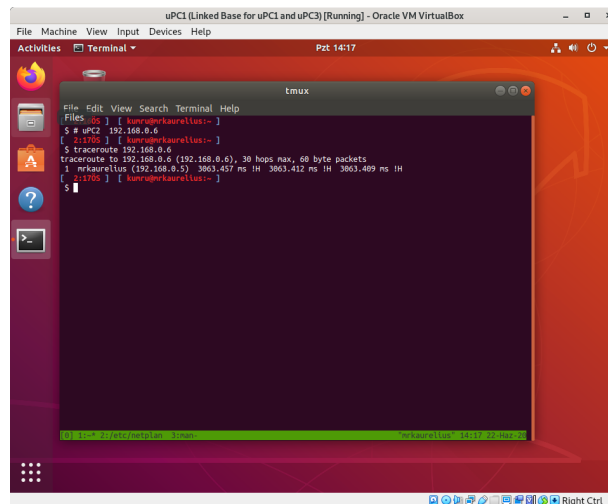


Figure 26: uPC1 -> uPC2

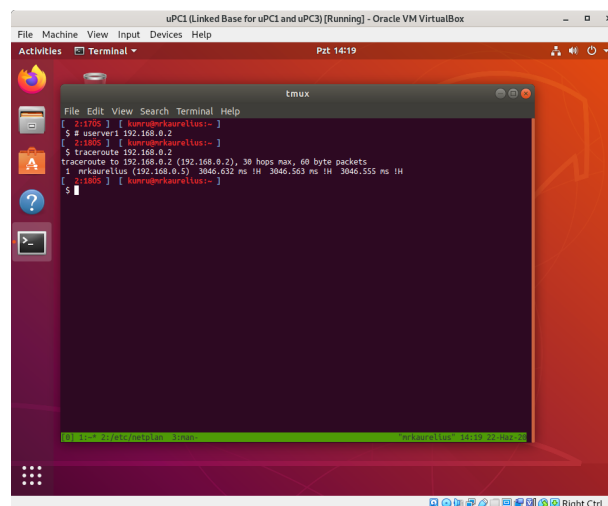


Figure 27: uPC1 -> userver1

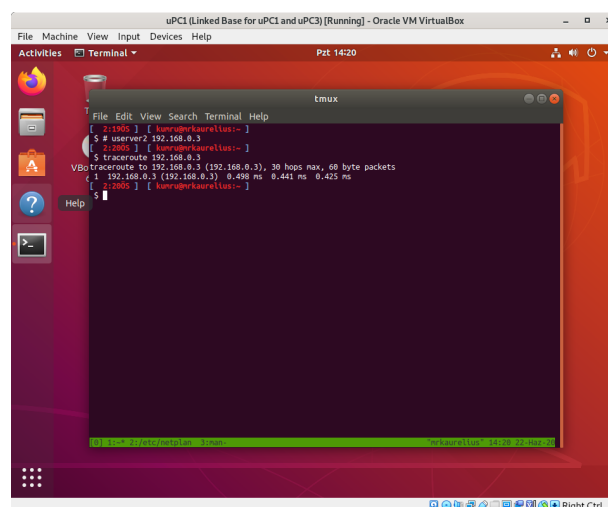


Figure 28: uPC1 -> userver2