Miles ogrady

Network setup

1. To find network information on this server you can use commands such as

Ubuntu Server network information:
This Server uses port 2222

Ubuntu server is tool based some important networking Command Line Tools are:

Server Ip address: (127.0.0.1) use ifconfig, if you don't have it use sudo apt install net-tools, also the server uses DHCP to assign the ip address automatically.

Route: Use route to get show ip table

```
miles@milesserver1:~$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         _gateway        0.0.0.0         UG    100    0        0 enp0s3
10.0.2.0        0.0.0.0         255.255.255.0   U     100    0        0 enp0s3
_gateway        0.0.0.0         255.255.255.255 UH    100    0        0 enp0s3
10.1.45.192     0.0.0.0         255.255.255.192 U     0      0        0 *
10.1.45.205     0.0.0.0         255.255.255.255 UH    0      0        0 calid692c5e3ad2
10.1.45.206     0.0.0.0         255.255.255.255 UH    0      0        0 cali5a6ab965db9
cdns01.comcast. _gateway        255.255.255.255 UGH   100    0        0 enp0s3
cdns02.comcast. _gateway        255.255.255.255 UGH   100    0        0 enp0s3
172.17.0.0      0.0.0.0         255.255.0.0     U     0      0        0 docker0
miles@milesserver1:~$ _
```

Firewall: use sudo iptables -S to show ipv4 rules

```
miles@milesserver1:~$ sudo iptables -S
# Warning: iptables-legacy tables present, use iptables-legacy to see them
-P INPUT ACCEPT
-P FORWARD ACCEPT
-P OUTPUT ACCEPT
-N DOCKER
-N DOCKER-ISOLATION-STAGE-1
-N DOCKER-ISOLATION-STAGE-2
-N DOCKER-USER
-A FORWARD -j DOCKER-USER
-A FORWARD -j DOCKER-ISOLATION-STAGE-1
-A FORWARD -o docker0 -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -o docker0 -j DOCKER
-A FORWARD -i docker0 ! -o docker0 -j ACCEPT
-A FORWARD -i docker0 -o docker0 -j ACCEPT
-A FORWARD -s 10.1.0.0/16 -m comment --comment "generated for MicroK8s pods" -j ACCEPT
-A FORWARD -d 10.1.0.0/16 -m comment --comment "generated for MicroK8s pods" -j ACCEPT
-A DOCKER-ISOLATION-STAGE-1 -i docker0 ! -o docker0 -j DOCKER-ISOLATION-STAGE-2
-A DOCKER-ISOLATION-STAGE-1 -j RETURN
-A DOCKER-ISOLATION-STAGE-2 -o docker0 -j DROP
-A DOCKER-ISOLATION-STAGE-2 -j RETURN
-A DOCKER-USER -j RETURN
miles@milesserver1:~$
```

Configs: The server has many config files located in the /etc/ directory

DNS config file: to display the file enter cat /etc/resolv.conf

```
miles@milesserver1:~$ cat /etc/resolv.conf
# This is /run/systemd/resolve/stub-resolv.conf managed by man:systemd-resolved(8).
# Do not edit.
#
# This file might be symlinked as /etc/resolv.conf. If you're looking at
# /etc/resolv.conf and seeing this text, you have followed the symlink.
#
# This is a dynamic resolv.conf file for connecting local clients to the
# internal DNS stub resolver of systemd-resolved. This file lists all
# configured search domains.
#
# Run "resolvectl status" to see details about the uplink DNS servers
# currently in use.
#
# Third party programs should typically not access this file directly, but only
# through the symlink at /etc/resolv.conf. To manage man:resolv.conf(5) in a
# different way, replace this symlink by a static file or a different symlink.
#
# See man:systemd-resolved.service(8) for details about the supported modes of
# operation for /etc/resolv.conf.

nameserver 127.0.0.53
options edns0 trust-ad
search hsd1.nh.comcast.net
miles@milesserver1:~$ _
```

Hosts file: to display hosts file use cat /etc/hosts

```
miles@milesserver1:~$ cat /etc/hosts
127.0.0.1 localhost
127.0.1.1 miles_server1

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
miles@milesserver1:~$ _
```

Scripts:
No networking scripts for this server.

Manual changes:
No manual changes so far.

centOS server network information:
This Server uses port 2223

centOS server is tool based some important networking Command Line Tools are:

Server Ip address: (127.0.0.1) use ifconfig, also the server uses DHCP to assign the ip address automatically.

Route: Use route to get show ip table

```
[miles@localhost ~]$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         _gateway        0.0.0.0         UG    100    0        0 enp0s3
10.0.2.0        0.0.0.0         255.255.255.0   U     100    0        0 enp0s3
[miles@localhost ~]$
```

Firewall: use sudo iptables -S to show ipv4 rules

```
[miles@localhost ~]$ sudo iptables -S
-P INPUT ACCEPT
-P FORWARD ACCEPT
-P OUTPUT ACCEPT
[miles@localhost ~]$
```

Configs: The server has many config files located in the /etc/ directory

DNS config file: to display the file enter cat /etc/resolv.conf

```
[miles@localhost ~]$ cat /etc/resolv.conf
# Generated by NetworkManager
search hsd1.nh.comcast.net
nameserver 75.75.75.75
nameserver 75.75.76.76
[miles@localhost ~]$
```

Hosts file: to display hosts file use cat /etc/hosts

```
[miles@localhost ~]$ cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
[miles@localhost ~]$ _
```

Scripts:

My centos server has network scripts located at /etc/sysconfig/network-scripts/ and has one file in it

/etc/sysconfig/network-scripts/

Name	Size	Changed	Rights	Owner
..		12/9/2023 5:06:17 PM	rwxr-xr-x	root
ifcfg-enp0s3	1 KB	12/9/2023 5:06:17 PM	rw-r--r--	root

Manual changes:

No manual changes so far.

My servers use multiple networking tools like, ifconfig, routing tables. The servers use configuration files by default to use network features and commands. The network uses firewall settings to handle the security on the network/server that regulate incoming and outgoing traffic with tools like iptables.