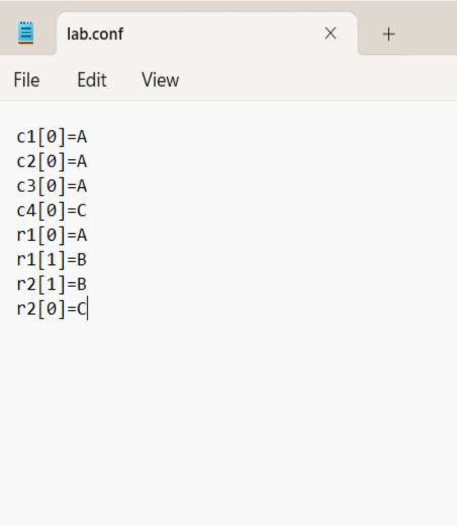
1. Creating lab topology

- Create the file named assignment_1.
- Create intermediate file named **lab.conf.**
- Content of the **lab.conf** as follows.



```
c1[0]=A
c2[0]=A
c3[0]=A
c4[0]=C
r1[0]=A
r1[1]=B
r2[1]=B
r2[0]=C
```

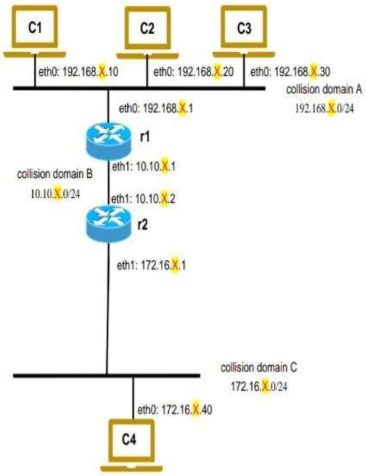
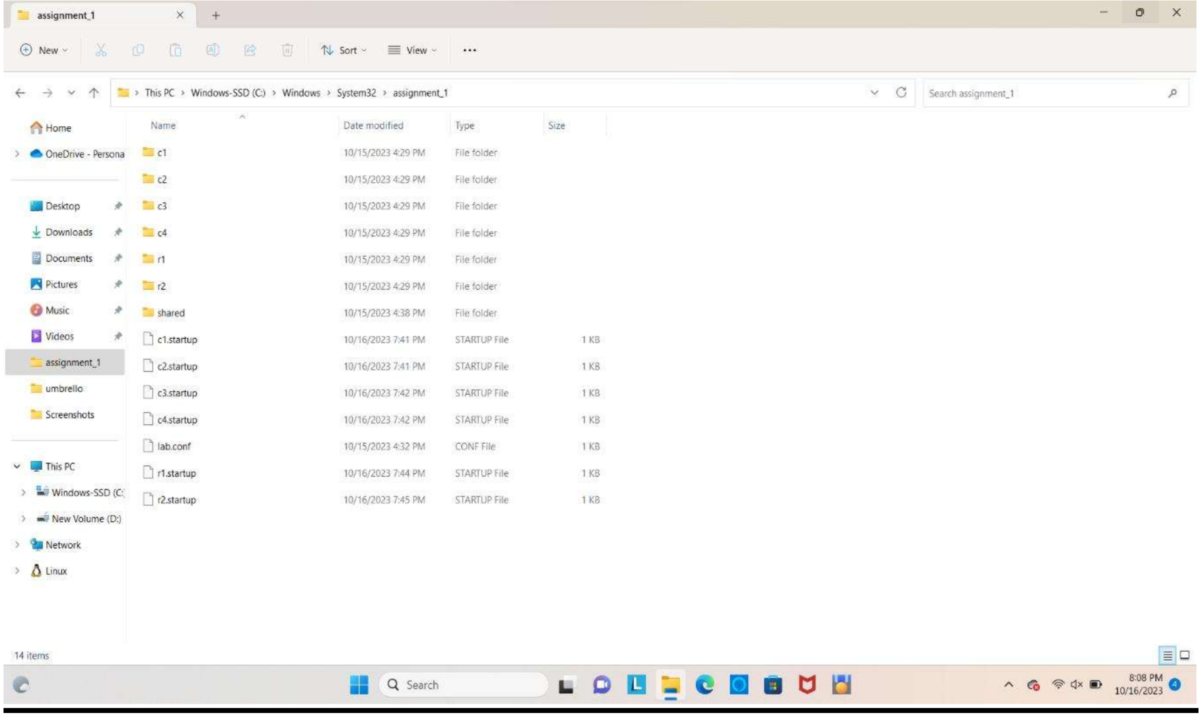


Figure 1: Network topology

2. Creating startup files

- Create startup files for each device.
- Name of the file shold be the file extension(**.startup**) followed by the device name (eg: **c1.startup**).

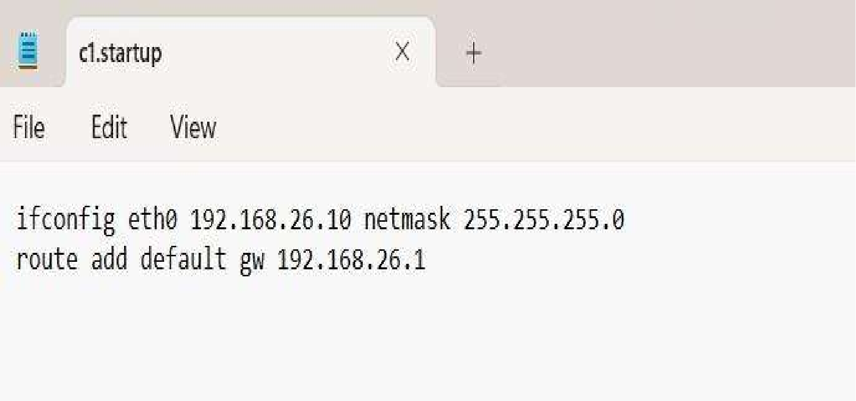


Name	Date modified	Type	Size
c1	10/15/2023 4:29 PM	File folder	
c2	10/15/2023 4:29 PM	File folder	
c3	10/15/2023 4:29 PM	File folder	
c4	10/15/2023 4:29 PM	File folder	
r1	10/15/2023 4:29 PM	File folder	
r2	10/15/2023 4:29 PM	File folder	
shared	10/15/2023 4:38 PM	File folder	
c1.startup	10/16/2023 7:41 PM	STARTUP File	1 KB
c2.startup	10/16/2023 7:41 PM	STARTUP File	1 KB
c3.startup	10/16/2023 7:42 PM	STARTUP File	1 KB
c4.startup	10/16/2023 7:42 PM	STARTUP File	1 KB
lab.conf	10/15/2023 4:32 PM	CONF File	1 KB
r1.startup	10/16/2023 7:44 PM	STARTUP File	1 KB
r2.startup	10/16/2023 7:45 PM	STARTUP File	1 KB

3. Contents of startup files

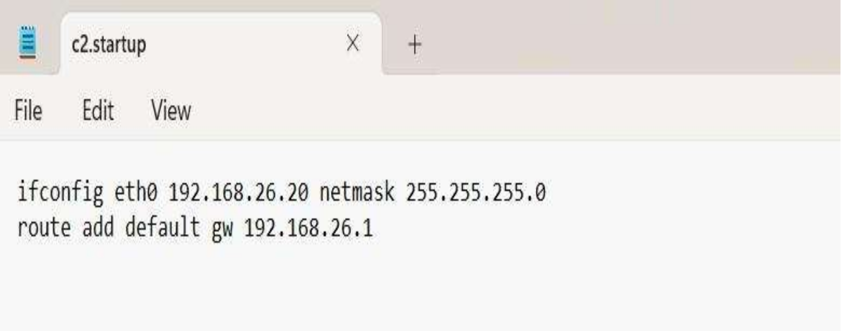
- The content of startup file for each device is their respective IP configurations,interface,netmask and routing address.

  - **c1.startup**



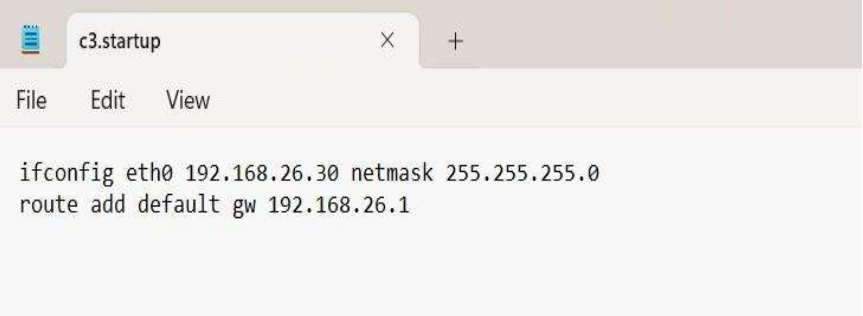
```
ifconfig eth0 192.168.26.10 netmask 255.255.255.0
route add default gw 192.168.26.1
```

  - **c2.startup**



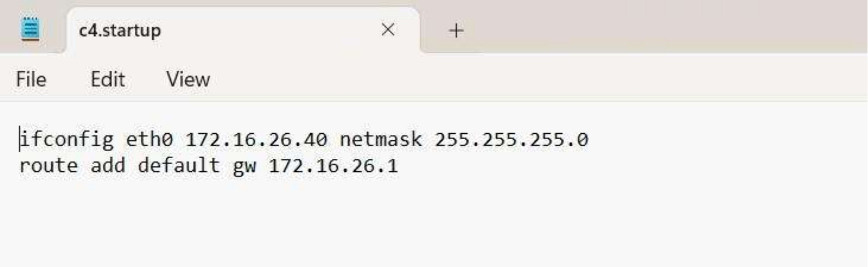
```
ifconfig eth0 192.168.26.20 netmask 255.255.255.0
route add default gw 192.168.26.1
```

  - **c3.startup**



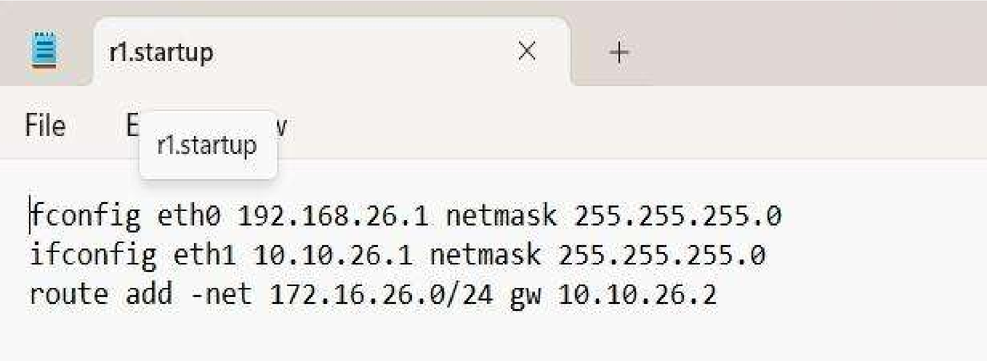
```
ifconfig eth0 192.168.26.30 netmask 255.255.255.0
route add default gw 192.168.26.1
```

  - **c4.startup**



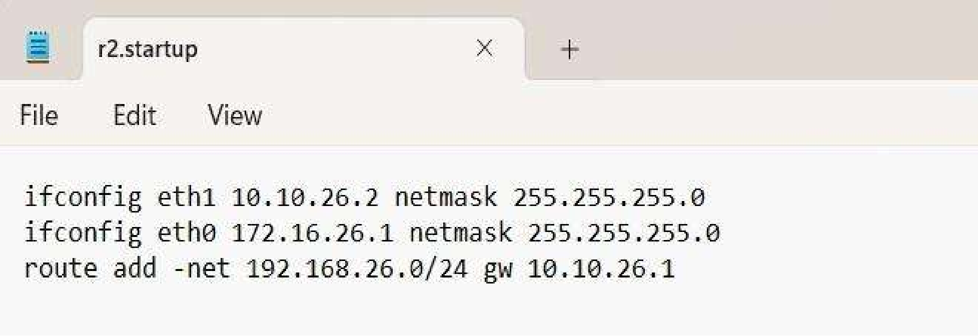
```
ifconfig eth0 172.16.26.40 netmask 255.255.255.0
route add default gw 172.16.26.1
```

- ### **r1.startup**



```
ifconfig eth0 192.168.26.1 netmask 255.255.255.0
ifconfig eth1 10.10.26.1 netmask 255.255.255.0
route add -net 172.16.26.0/24 gw 10.10.26.2
```

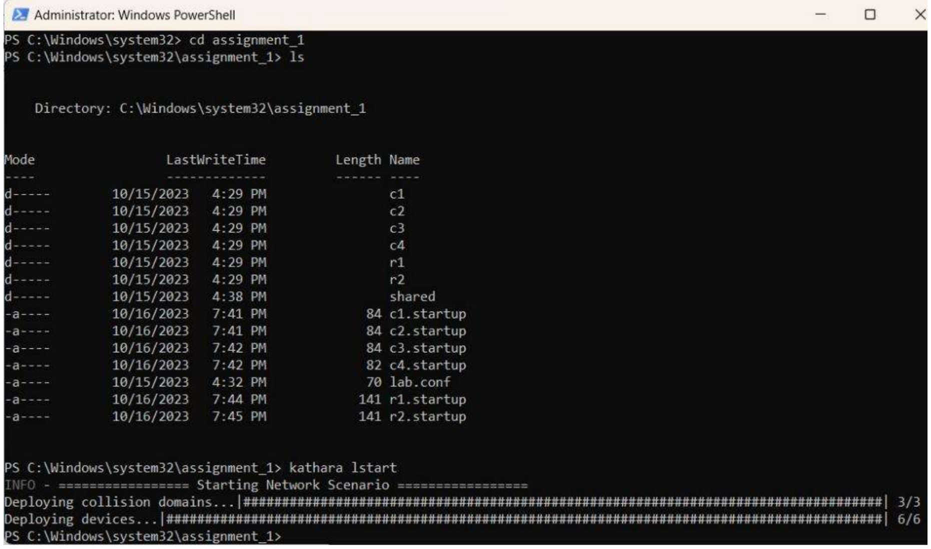
- ### **r2.startup**



```
ifconfig eth1 10.10.26.2 netmask 255.255.255.0
ifconfig eth0 172.16.26.1 netmask 255.255.255.0
route add -net 192.168.26.0/24 gw 10.10.26.1
```

4. Implementing

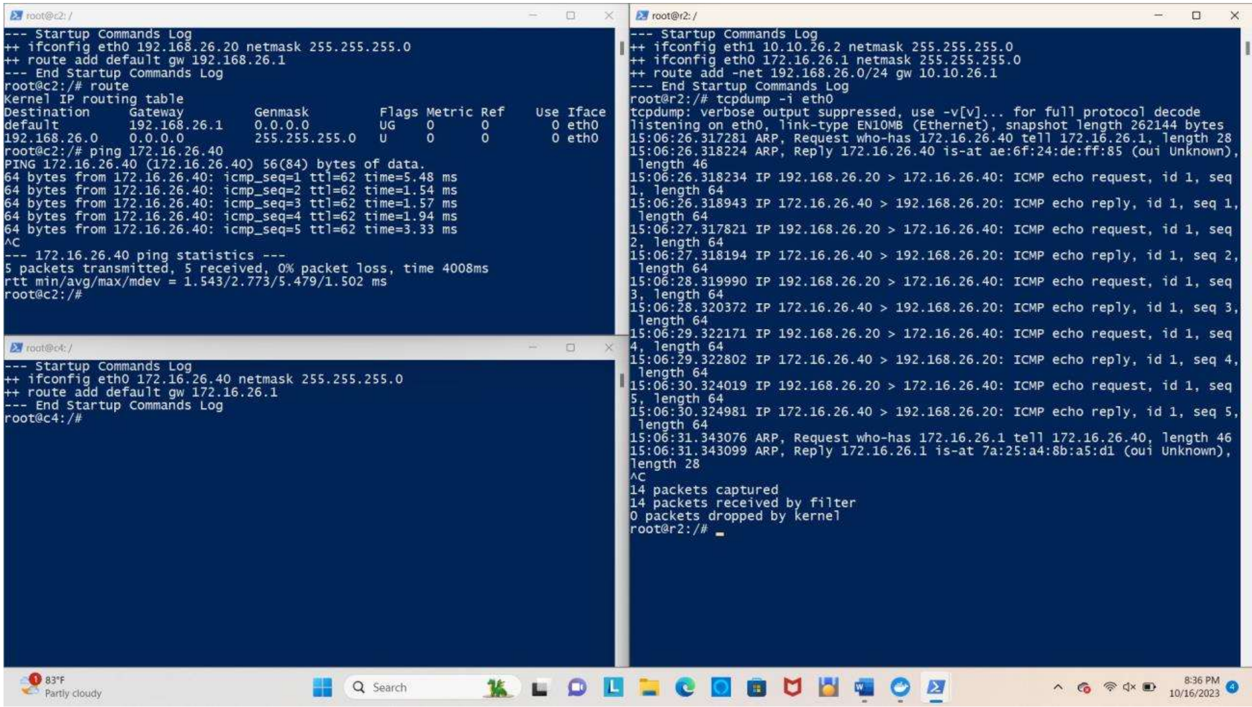
- To implement the network topology open docker and run in background.
- Now,open windows power shell and change directory(**cd**) to **assignment_1**.
- Then start kathara application by using (**kathara lstart**).



```
PS C:\Windows\system32> cd assignment_1
PS C:\Windows\system32\assignment_1> ls


    Directory: C:\Windows\system32\assignment_1


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        10/15/2023   4:29 PM                c1
d-----        10/15/2023   4:29 PM                c2
d-----        10/15/2023   4:29 PM                c3
d-----        10/15/2023   4:29 PM                c4
d-----        10/15/2023   4:29 PM                r1
d-----        10/15/2023   4:29 PM                r2
d-----        10/15/2023   4:38 PM                shared
-a----        10/16/2023   7:41 PM             84 c1.startup
-a----        10/16/2023   7:41 PM             84 c2.startup
-a----        10/16/2023   7:42 PM             84 c3.startup
-a----        10/16/2023   7:42 PM             82 c4.startup
-a----        10/15/2023   4:32 PM             70 lab.conf
-a----        10/16/2023   7:44 PM            141 r1.startup
-a----        10/16/2023   7:45 PM            141 r2.startup


PS C:\Windows\system32\assignment_1> kathara lstart
INFO - ================= Starting Network Scenario =================
Deploying collision domains...|##################################################################################| 3/3
Deploying devices...|############################################################################################| 6/6
PS C:\Windows\system32\assignment_1>
```

5. **Observation**

- Capture the packet with the ping command triggered on c2 with destination c4.
- Check whether the ping is working or not.
- Capture with the tcpdump command on the interface eth0. Started on r2,while the ping from c2 to c4 is working.



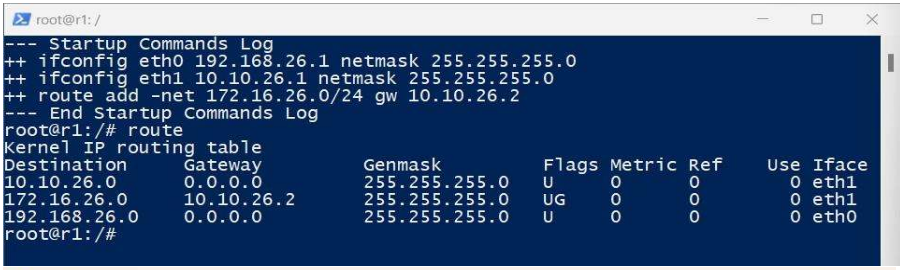
```
--- Startup Commands Log
++ ifconfig eth0 192.168.26.20 netmask 255.255.255.0
++ route add default gw 192.168.26.1
--- End Startup Commands Log
root@c2:/# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         192.168.26.1    0.0.0.0         UG    0      0        0 eth0
192.168.26.0    0.0.0.0         255.255.255.0   U     0      0        0 eth0
root@c2:/# ping 172.16.26.40
PING 172.16.26.40 (172.16.26.40) 56(84) bytes of data.
64 bytes from 172.16.26.40: icmp_seq=1 ttl=62 time=5.48 ms
64 bytes from 172.16.26.40: icmp_seq=2 ttl=62 time=1.54 ms
64 bytes from 172.16.26.40: icmp_seq=3 ttl=62 time=1.57 ms
64 bytes from 172.16.26.40: icmp_seq=4 ttl=62 time=1.94 ms
64 bytes from 172.16.26.40: icmp_seq=5 ttl=62 time=3.33 ms
^C
--- 172.16.26.40 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4008ms
rtt min/avg/max/mdev = 1.543/2.773/5.479/1.502 ms
root@c2:/#
```

```
--- Startup Commands Log
++ ifconfig eth0 172.16.26.40 netmask 255.255.255.0
++ route add default gw 172.16.26.1
--- End Startup Commands Log
root@c4:/#
```

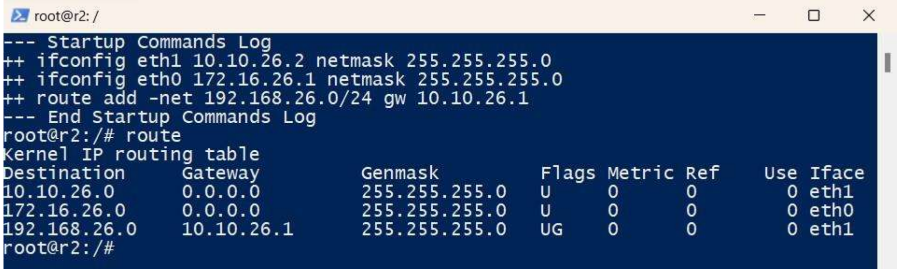
```
--- Startup Commands Log
++ ifconfig eth1 10.10.26.2 netmask 255.255.255.0
++ ifconfig eth0 172.16.26.1 netmask 255.255.255.0
++ route add -net 192.168.26.0/24 gw 10.10.26.1
--- End Startup Commands Log
root@r2:/# tcpdump -i eth0
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
15:06:26.317281 ARP, Request who-has 172.16.26.40 tell 172.16.26.1, length 28
15:06:26.318224 ARP, Reply 172.16.26.40 is-at ae:6f:24:de:ff:85 (oui Unknown),
 length 46
15:06:26.318234 IP 192.168.26.20 > 172.16.26.40: ICMP echo request, id 1, seq
1, length 64
15:06:26.318943 IP 172.16.26.40 > 192.168.26.20: ICMP echo reply, id 1, seq 1,
 length 64
15:06:27.317821 IP 192.168.26.20 > 172.16.26.40: ICMP echo request, id 1, seq
2, length 64
15:06:27.318194 IP 172.16.26.40 > 192.168.26.20: ICMP echo reply, id 1, seq 2,
 length 64
15:06:28.319990 IP 192.168.26.20 > 172.16.26.40: ICMP echo request, id 1, seq
3, length 64
15:06:28.320372 IP 172.16.26.40 > 192.168.26.20: ICMP echo reply, id 1, seq 3,
 length 64
15:06:29.322171 IP 192.168.26.20 > 172.16.26.40: ICMP echo request, id 1, seq
4, length 64
15:06:29.322802 IP 172.16.26.40 > 192.168.26.20: ICMP echo reply, id 1, seq 4,
 length 64
15:06:30.324019 IP 192.168.26.20 > 172.16.26.40: ICMP echo request, id 1, seq
5, length 64
15:06:30.324981 IP 172.16.26.40 > 192.168.26.20: ICMP echo reply, id 1, seq 5,
 length 64
15:06:31.343076 ARP, Request who-has 172.16.26.1 tell 172.16.26.40, length 46
15:06:31.343099 ARP, Reply 172.16.26.1 is-at 7a:25:a4:8b:a5:d1 (oui Unknown),
length 28
^C
14 packets captured
14 packets received by filter
0 packets dropped by kernel
root@r2:/# _
```

6. **Routing tables**

- To verify the routing table use command **route** for the given routers (ie: **r1** and **r2**).



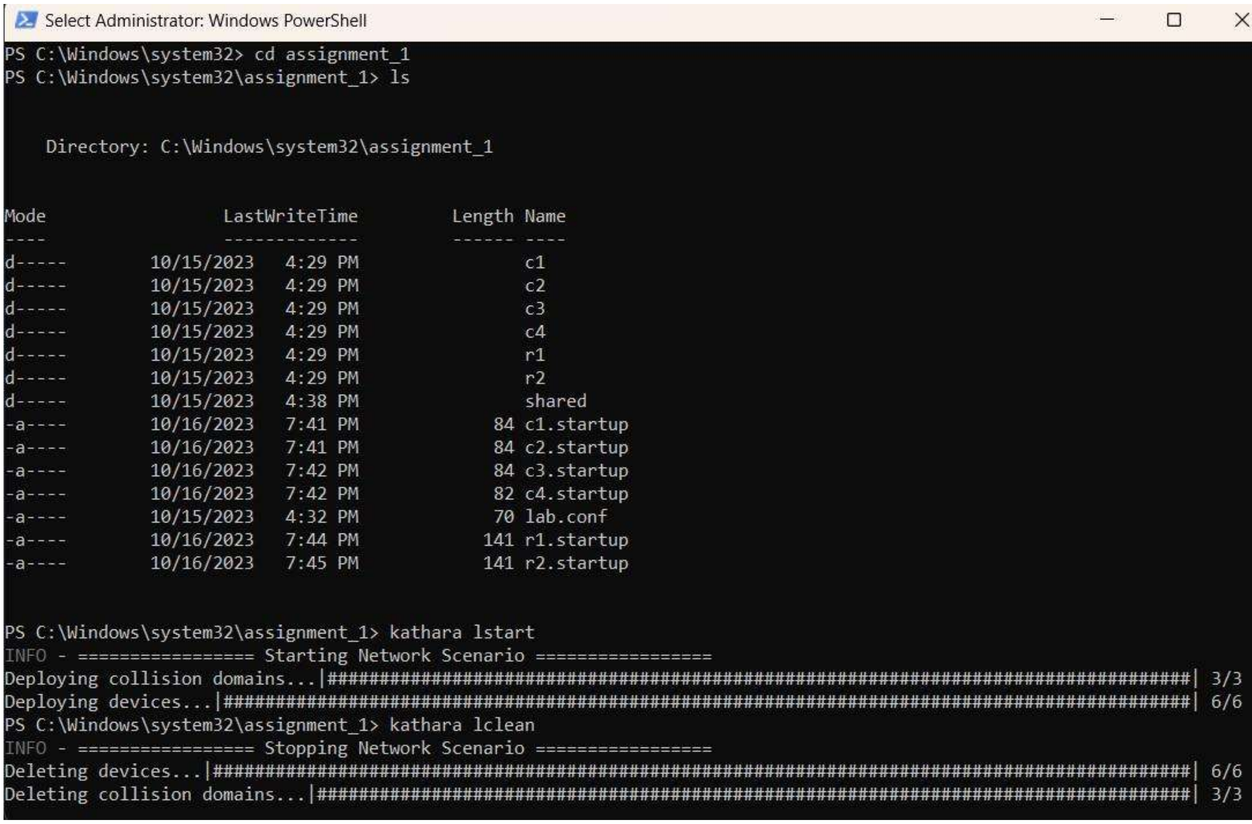
```
--- Startup Commands Log
++ ifconfig eth0 192.168.26.1 netmask 255.255.255.0
++ ifconfig eth1 10.10.26.1 netmask 255.255.255.0
++ route add -net 172.16.26.0/24 gw 10.10.26.2
--- End Startup Commands Log
root@r1:/# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.10.26.0      0.0.0.0         255.255.255.0   U     0      0        0 eth1
172.16.26.0     10.10.26.2      255.255.255.0   UG    0      0        0 eth1
192.168.26.0    0.0.0.0         255.255.255.0   U     0      0        0 eth0
root@r1:/#
```



```
--- Startup Commands Log
++ ifconfig eth1 10.10.26.2 netmask 255.255.255.0
++ ifconfig eth0 172.16.26.1 netmask 255.255.255.0
++ route add -net 192.168.26.0/24 gw 10.10.26.1
--- End Startup Commands Log
root@r2:/# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.10.26.0      0.0.0.0         255.255.255.0   U     0      0        0 eth1
172.16.26.0     0.0.0.0         255.255.255.0   U     0      0        0 eth0
192.168.26.0    10.10.26.1      255.255.255.0   UG    0      0        0 eth1
root@r2:/#
```

7. <u>Stopping kathara</u>

- To stop kathara programm after completion of network topology verification, use the command **kathara lclean** to clear all network collision domains and devices.



```
PS C:\Windows\system32> cd assignment_1
PS C:\Windows\system32\assignment_1> ls


    Directory: C:\Windows\system32\assignment_1


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        10/15/2023   4:29 PM                c1
d-----        10/15/2023   4:29 PM                c2
d-----        10/15/2023   4:29 PM                c3
d-----        10/15/2023   4:29 PM                c4
d-----        10/15/2023   4:29 PM                r1
d-----        10/15/2023   4:29 PM                r2
d-----        10/15/2023   4:38 PM                shared
-a----        10/16/2023   7:41 PM             84 c1.startup
-a----        10/16/2023   7:41 PM             84 c2.startup
-a----        10/16/2023   7:42 PM             84 c3.startup
-a----        10/16/2023   7:42 PM             82 c4.startup
-a----        10/15/2023   4:32 PM             70 lab.conf
-a----        10/16/2023   7:44 PM            141 r1.startup
-a----        10/16/2023   7:45 PM            141 r2.startup


PS C:\Windows\system32\assignment_1> kathara lstart
INFO - ================= Starting Network Scenario =================
Deploying collision domains...|###################################################################################| 3/3
Deploying devices...|#############################################################################################| 6/6
PS C:\Windows\system32\assignment_1> kathara lclean
INFO - ================= Stopping Network Scenario =================
Deleting devices...|##############################################################################################| 6/6
Deleting collision domains...|####################################################################################| 3/3
```

8. <u>Result</u>

- Therefore, the verification of the network topology is completed.
- Verification of routing tables of **r1** and **r2** is completed.
- And the connection between **c2** and **c4** is implemented successfully.