iu-ne-lab-02-Iskander_Nafikov

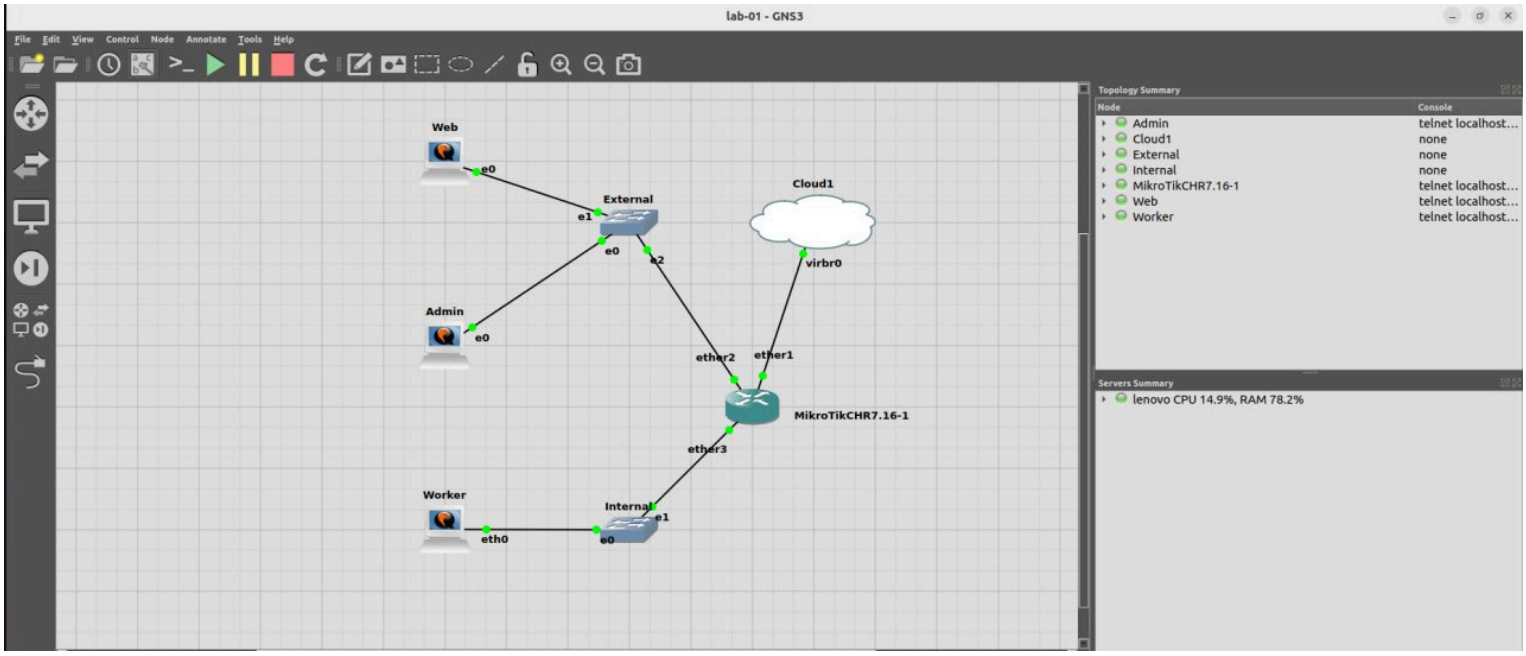
- **Name**: Iskander Nafikov
- **E-mail**: i.nafikov@innopolis.university
- **Username**: iskanred
- **Hostname**: `lenovo`

Overview

> In this lab, you will get familiar with some of the layers of the OSI model, mainly the layers from 3 (Networking) to 7 (Application). You will learn about some protocols relying on the IP protocol (version 4 and 6) and then learn some of the skills required to troubleshoot your networks in the case of some problems.

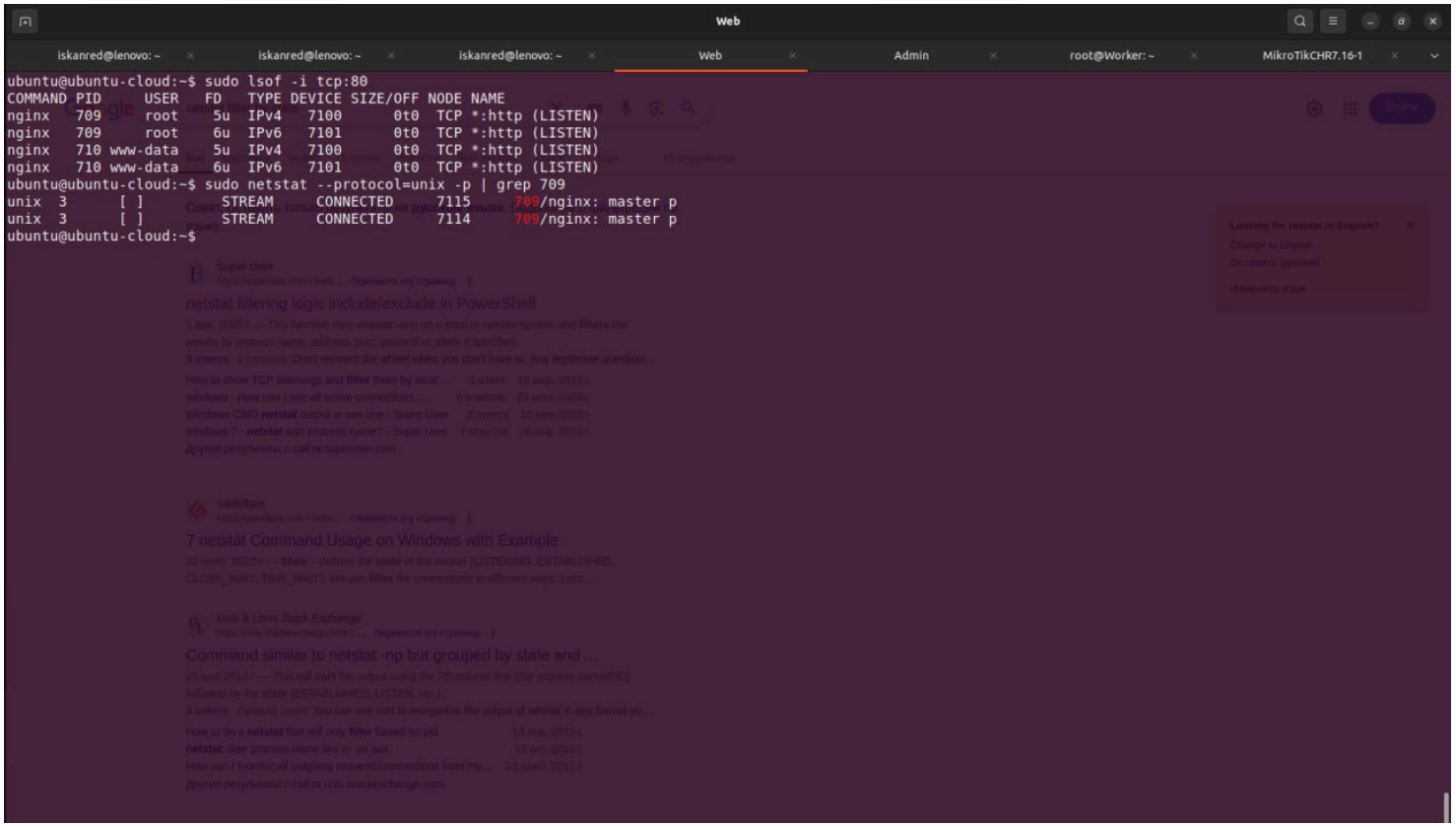
Task 1 - Ports and Protocols

My topology is:

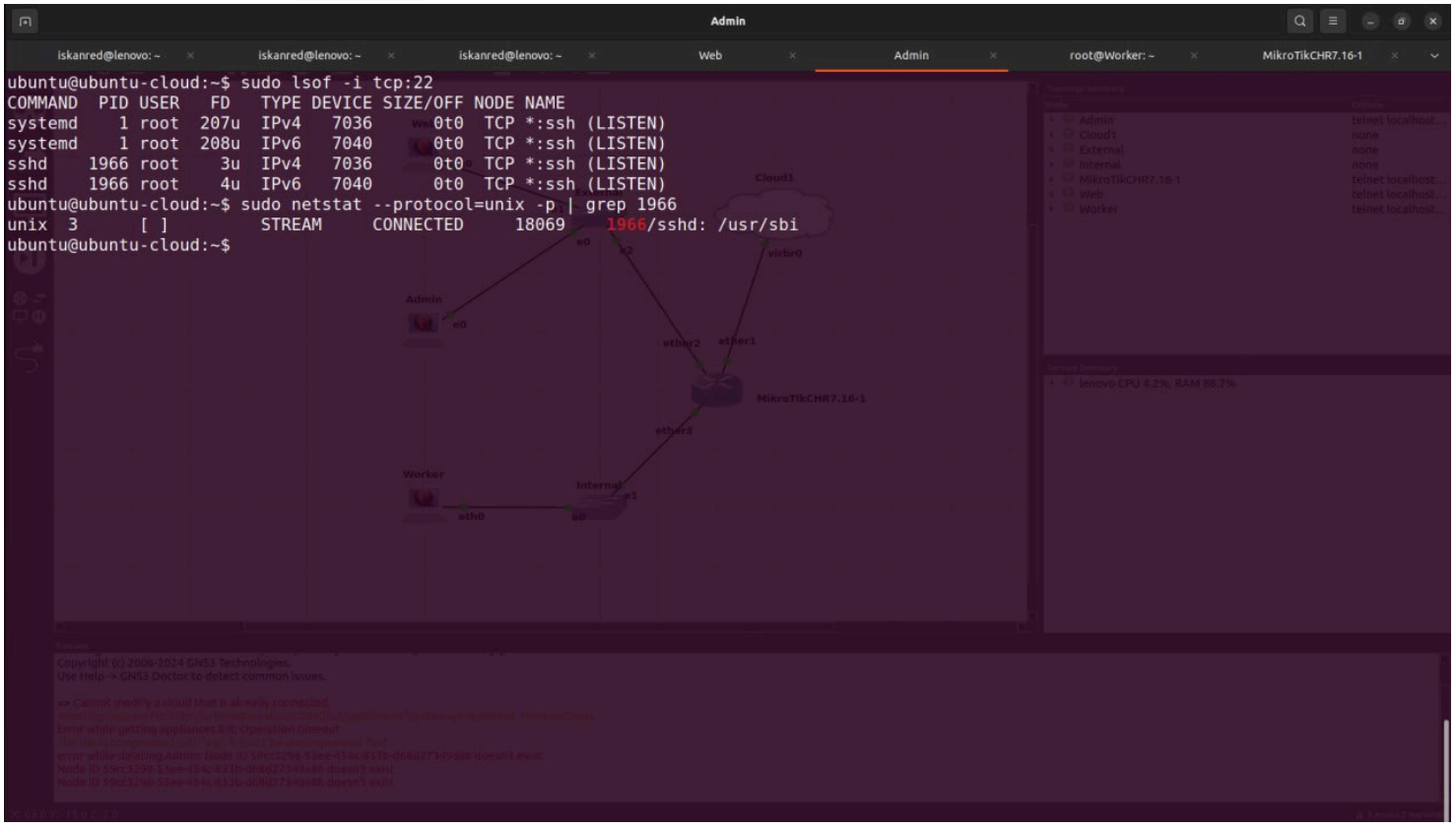


> 1. Check the open ports and listening Unix sockets against ssh (22) and http (80) on Admin and Web respectively.

- Using `lsof` and `netstat` I checked ports and Unix sockets against `http (80)` for Web node:

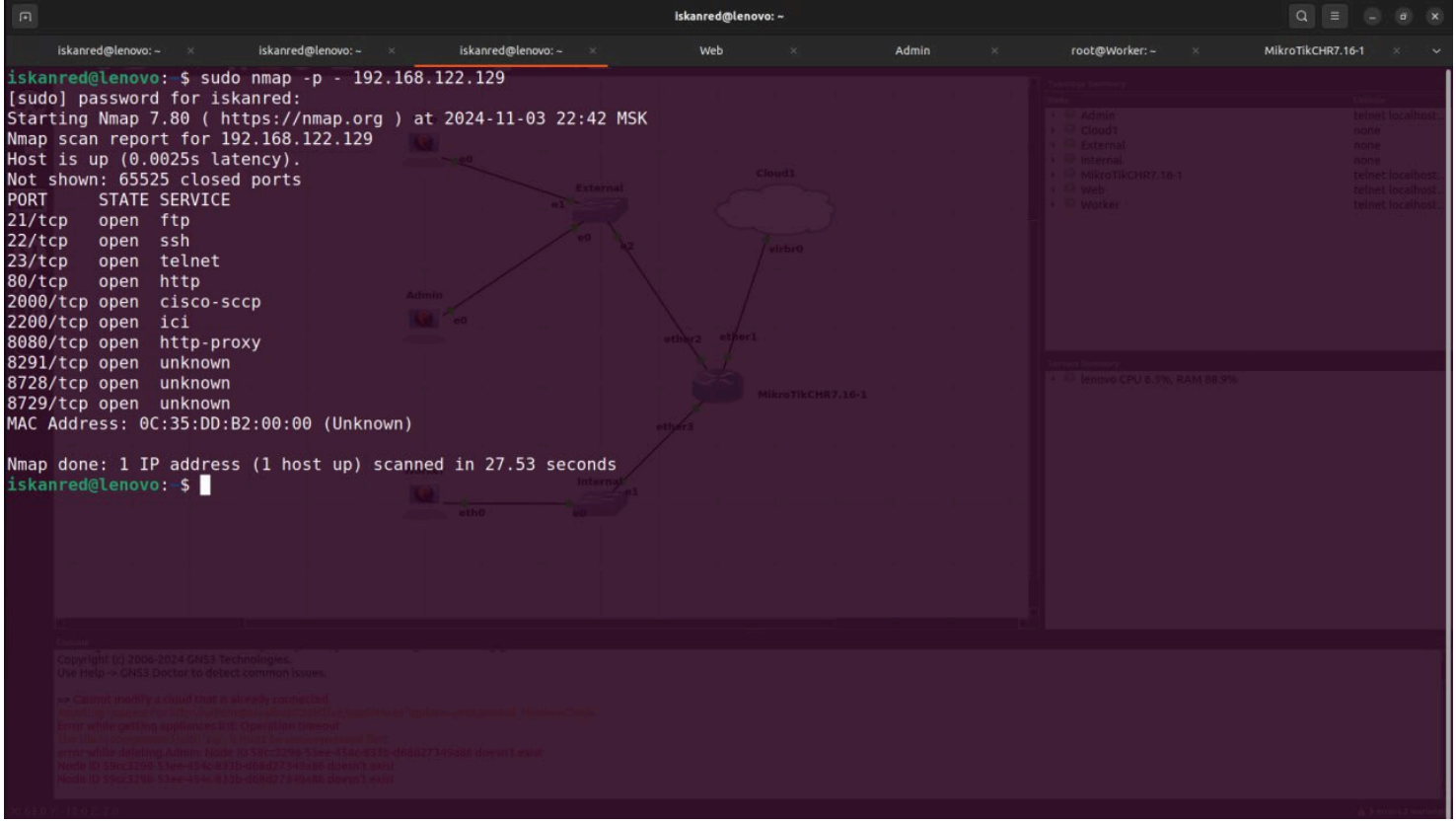


- And the samfe against `ssh (22)` for Admin node:



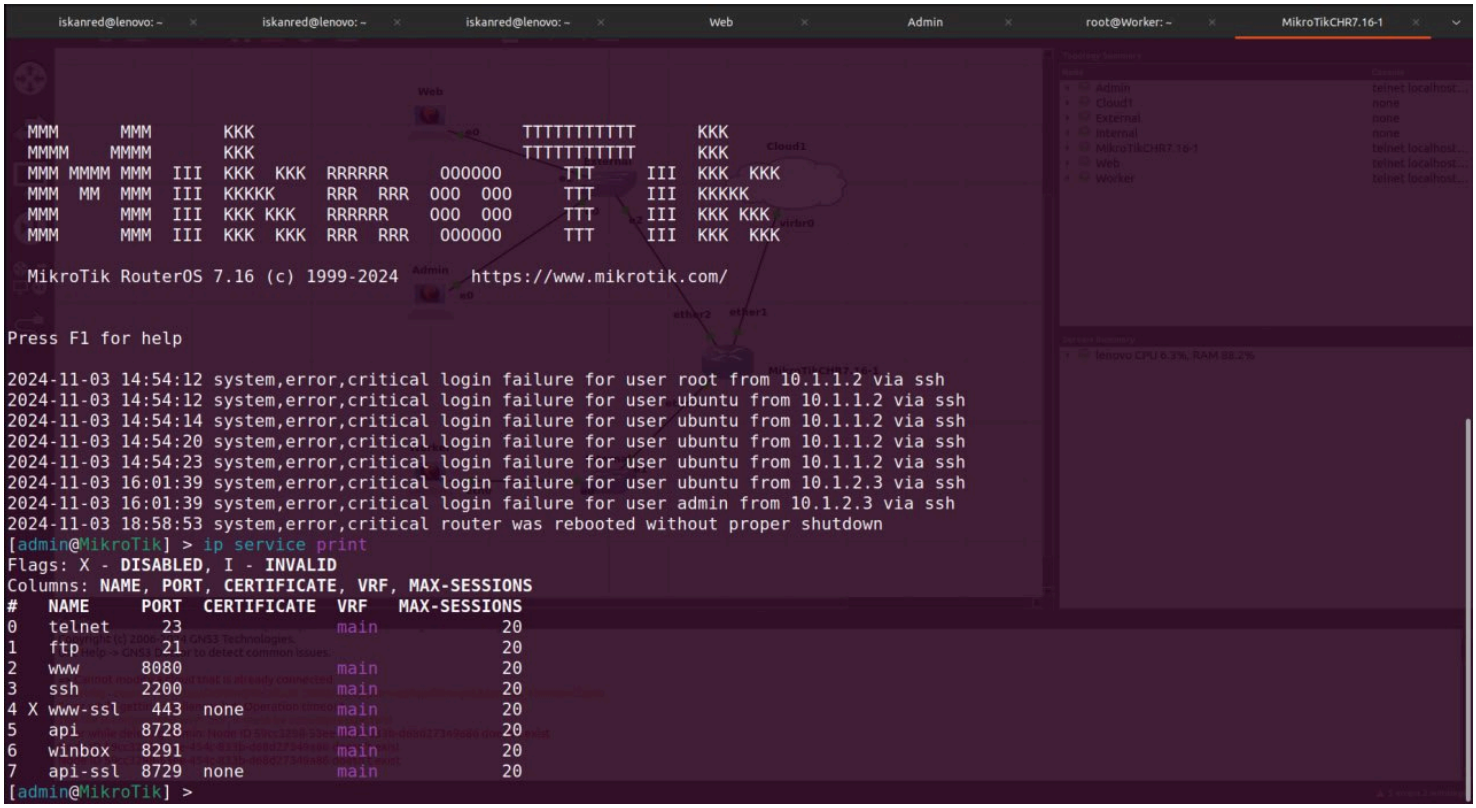
> 2. Scan your gateway from the outside. What are the known open ports?

- First, my virtual gateway is at `192.168.122.169`. The port-forwarding is enabled for ports `80` and `22`. Scanning this address for open ports gives the following results:

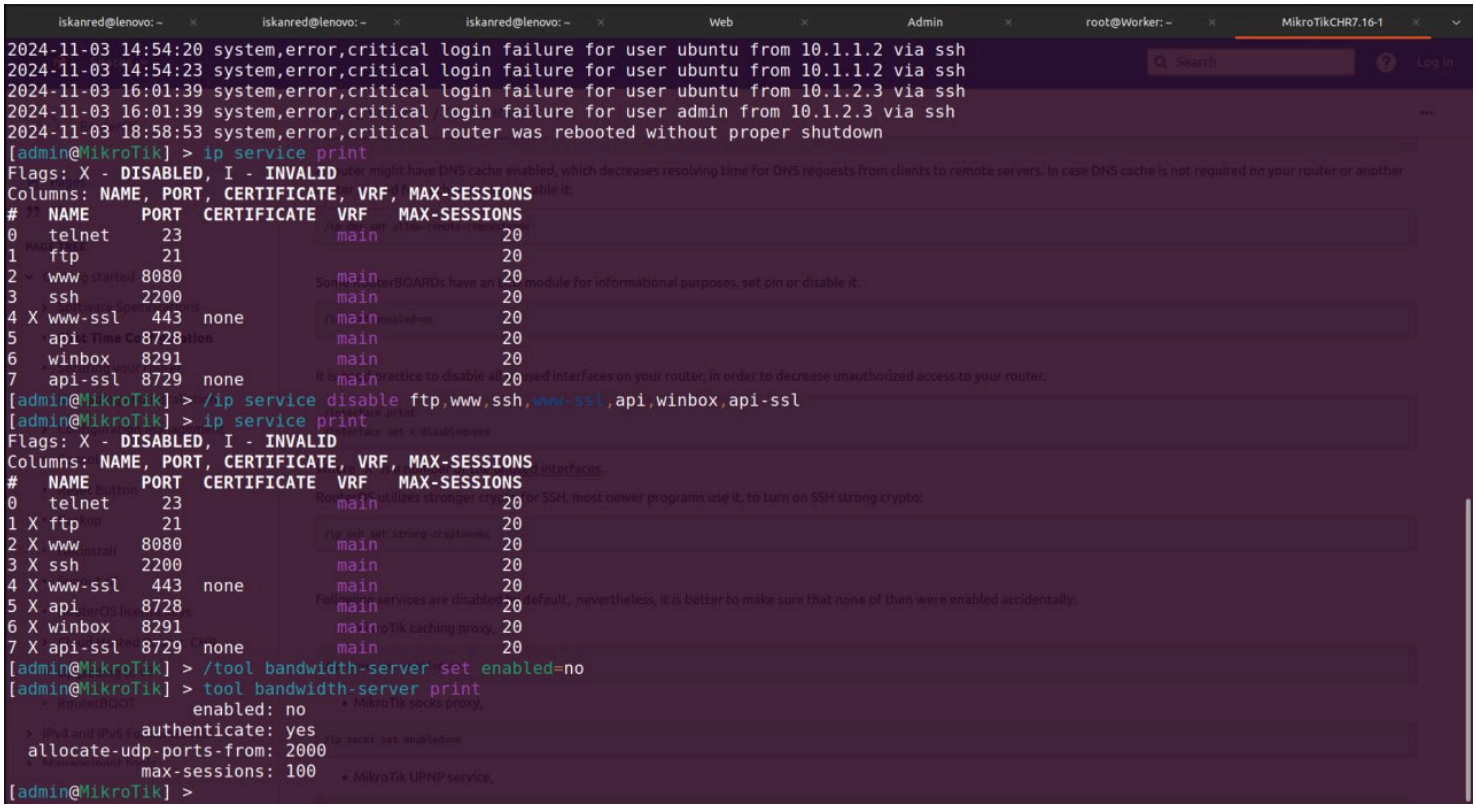


3. A gateway has to be transparent, you should not see any port that is not specifically forwarded. Adjust your firewall rules to make this happen. Disable any unnecessary services and scan again.

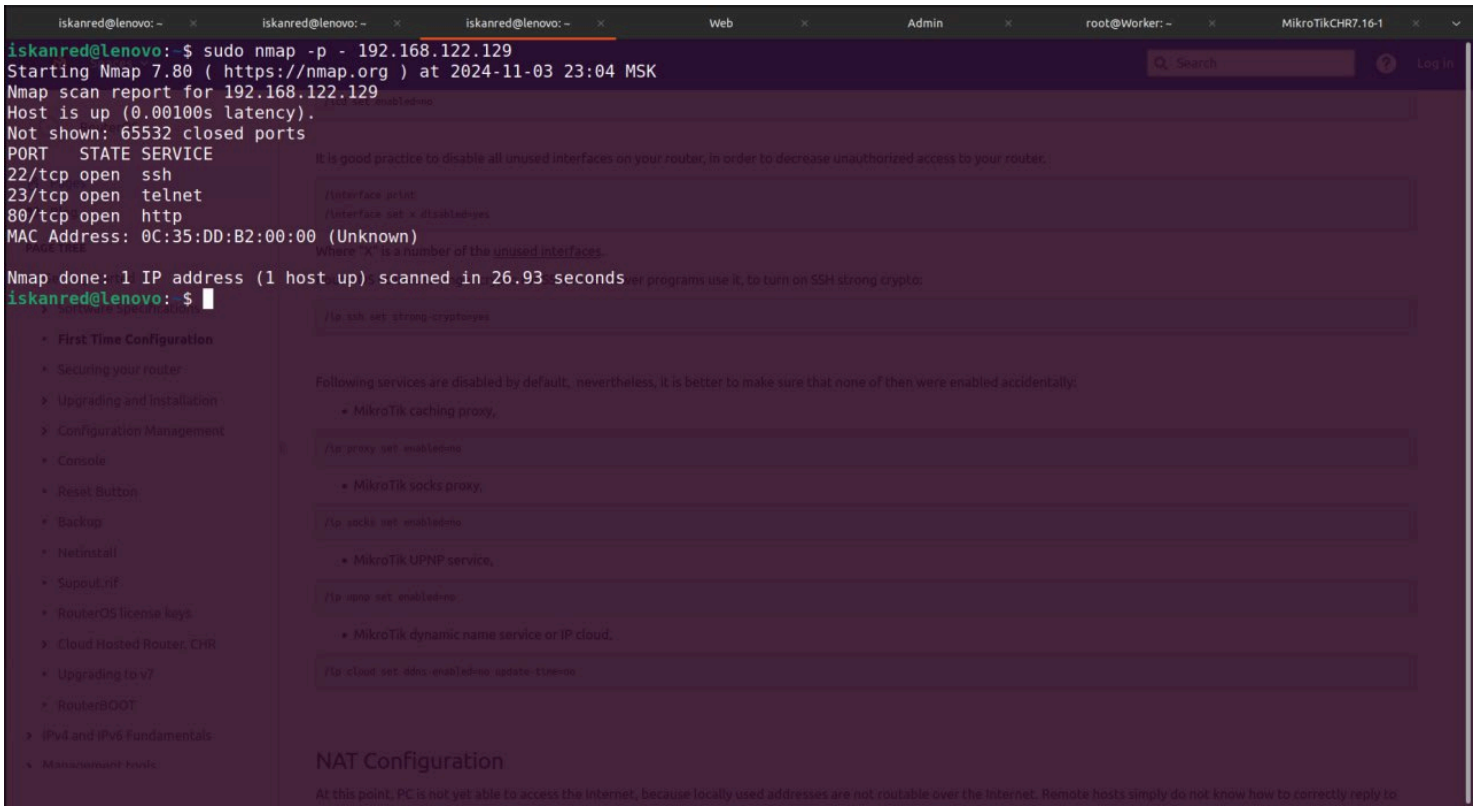
- We see there are several ports opened and some of them are seemed to be unknown for the `nmap`. Let's check the running services on these ports:



- We can see now what those ports mean. Let's disable some of them:



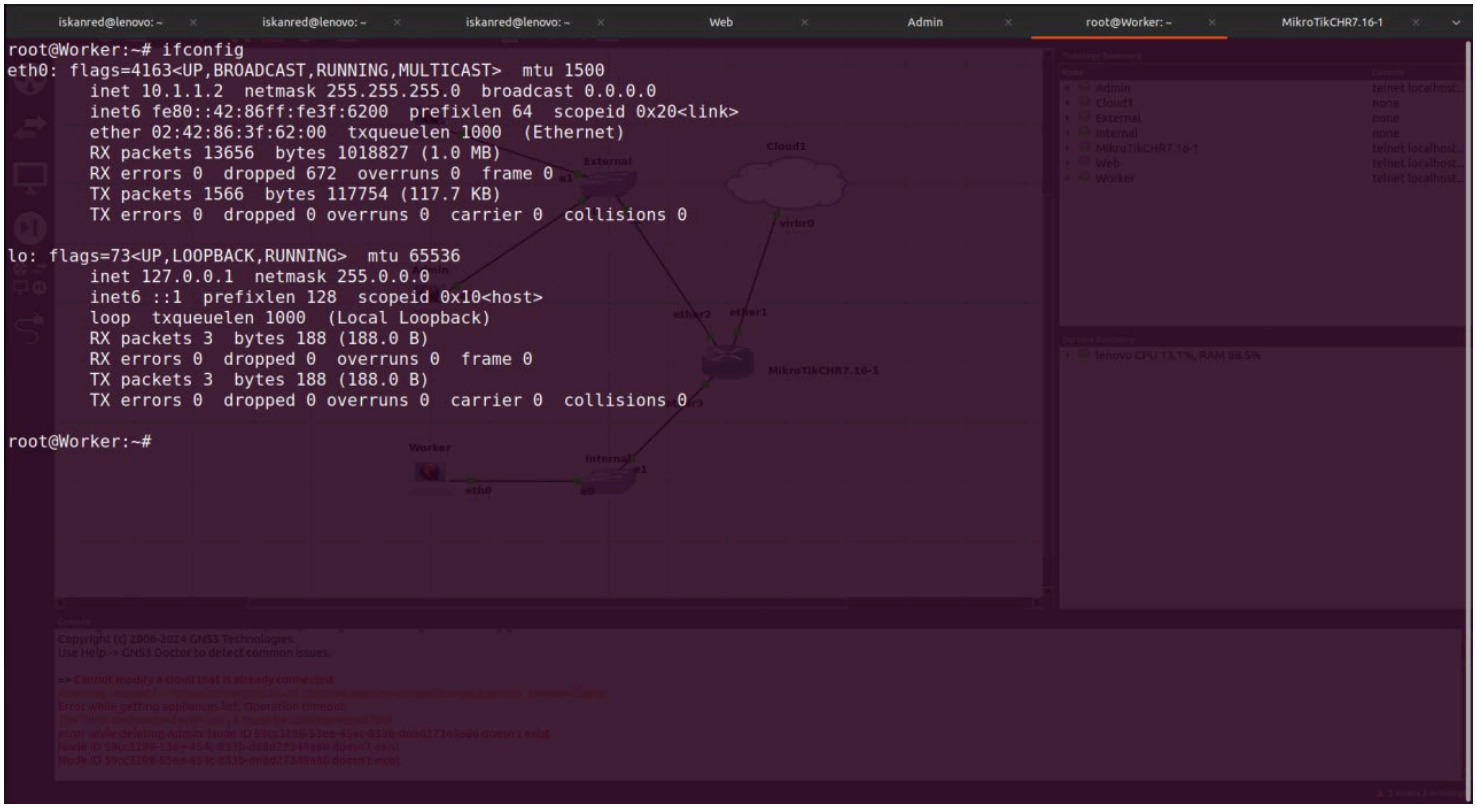
- Besides `ssh` and `http` I left only `telnet` since I prefer this way for interact with the router from a console. Overall, now only needed ports are opened from the outside:



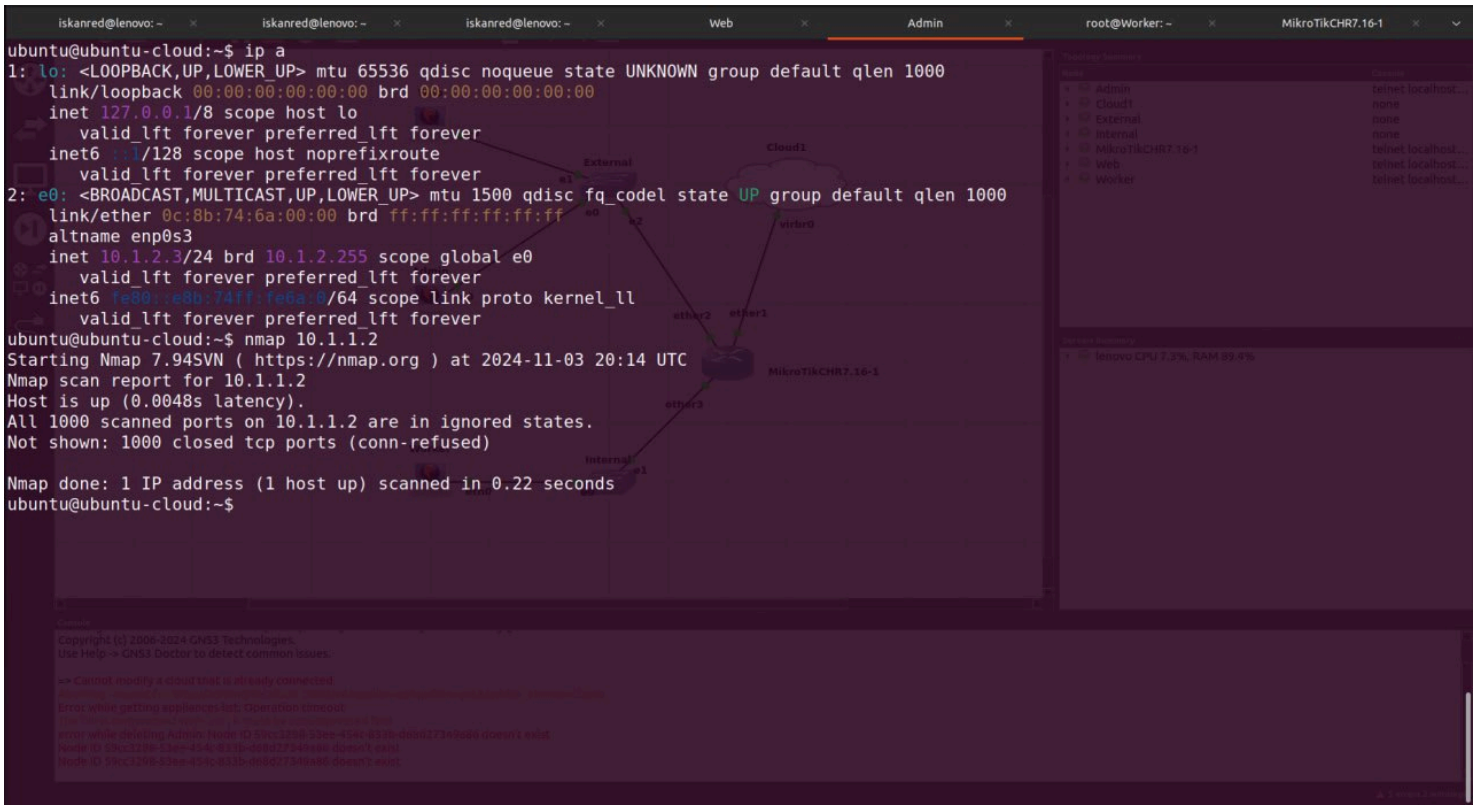
> 4. It supposed that some scanners start by scanning the known ports and pinging a host to see if it is alive

> 4.1 Scan the Worker VM from Admin. Can you see any ports?

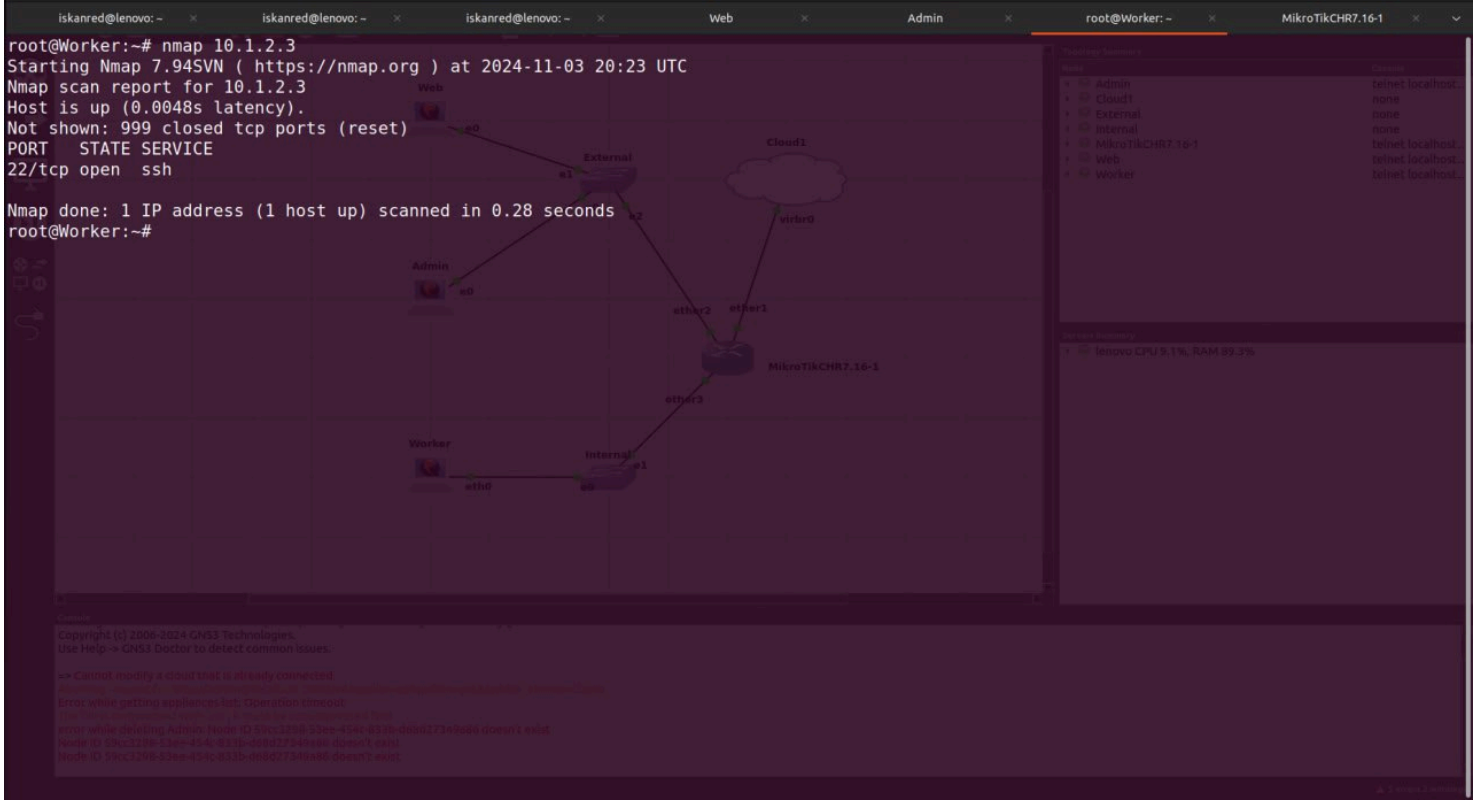
- First, my worker is located at `10.1.1.2`:



- So, if I make just `nmap 10.1.1.2` from my Admin node (which is at `10.1.2.3`) I got only the following picture without ports:



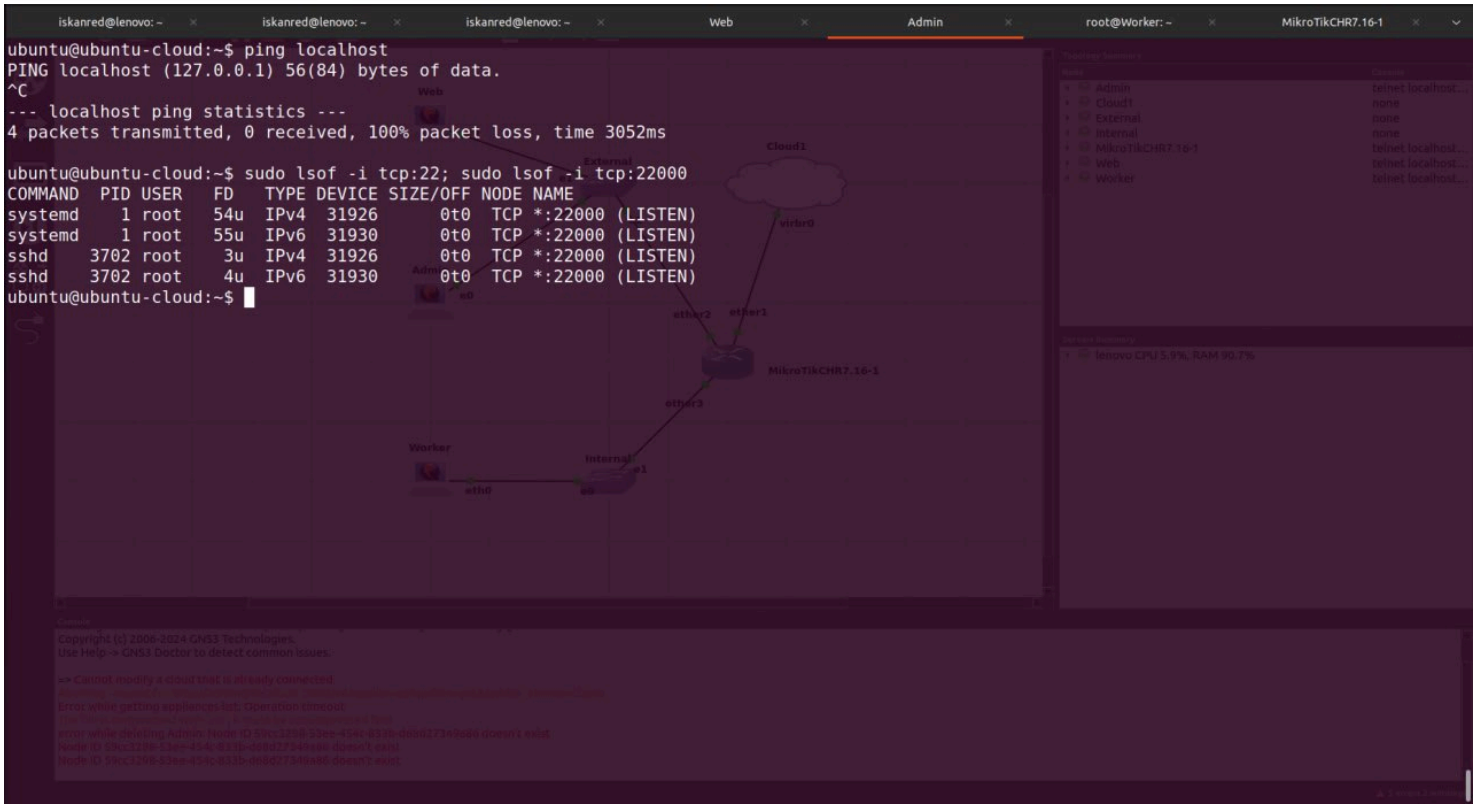
- Since we had no requirements to install SSH I decided to reverse the task description and perform all the operations from the `Admin` to `Worker` nodes. So, now I can show the result of scanning Admin node (`10.1.2.3/24`) from Worker (`10.1.1.2/24`):



Now we can see that `ssh/22` port is open

> 4.2 Block ICMP traffic on Worker and change the port for SSH to one that is above 10000

- I blocked ICMP echo responses on Admin node with the help of `/etc/sysctl.conf` - `net.ipv4.icmp_echo_ignore_all=1` and I changed SSH port to `22000` instead of default `22` inside the `/etc/ssh/sshd_config` - `Port 22000`.



> 4.3 Scan it without extra arguments.

- Afterwards, I performed `nmap 10.1.2.3` from Worker (`10.1.1.2/24`) to Admin (`10.1.2.3/24`):

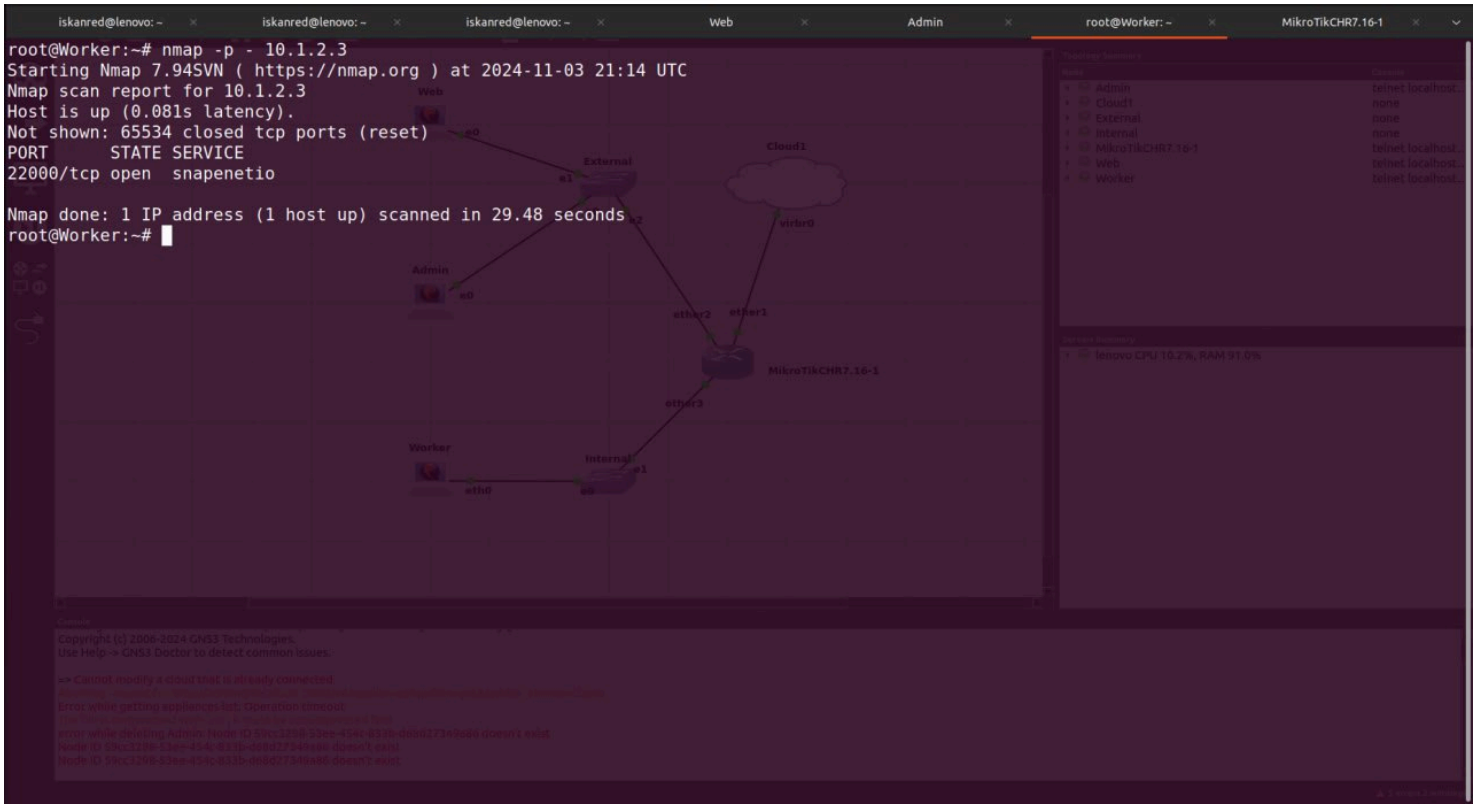
As we can see `nmap` still can recognise that the host is UP because host discovery includes the ping scan which is from the docs:

> `Despite the name ping scan, this goes well beyond the simple ICMP echo request packets associated with the ubiquitous ping tool.`

However, now it cannot find the newly assigned port `22000` for the ssh server

> 4.4. Now make necessary changes to the command to force the scan on all possible ports.

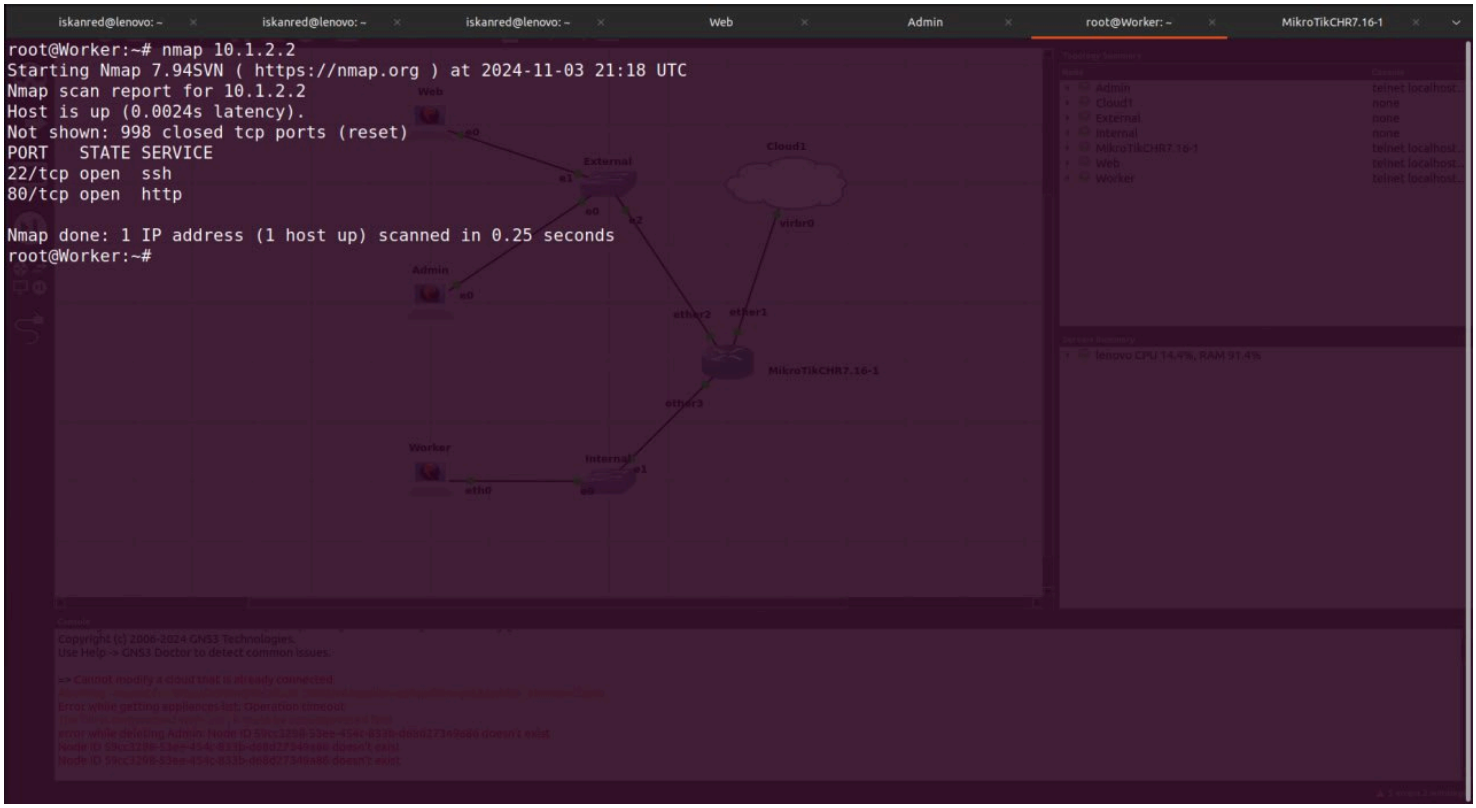
- Now we can make full port scan using `nmap -p - {ip}`



Finally, port `22000` is visible

> 4.5. Gather some information about your open ports on Web (ssh and http).

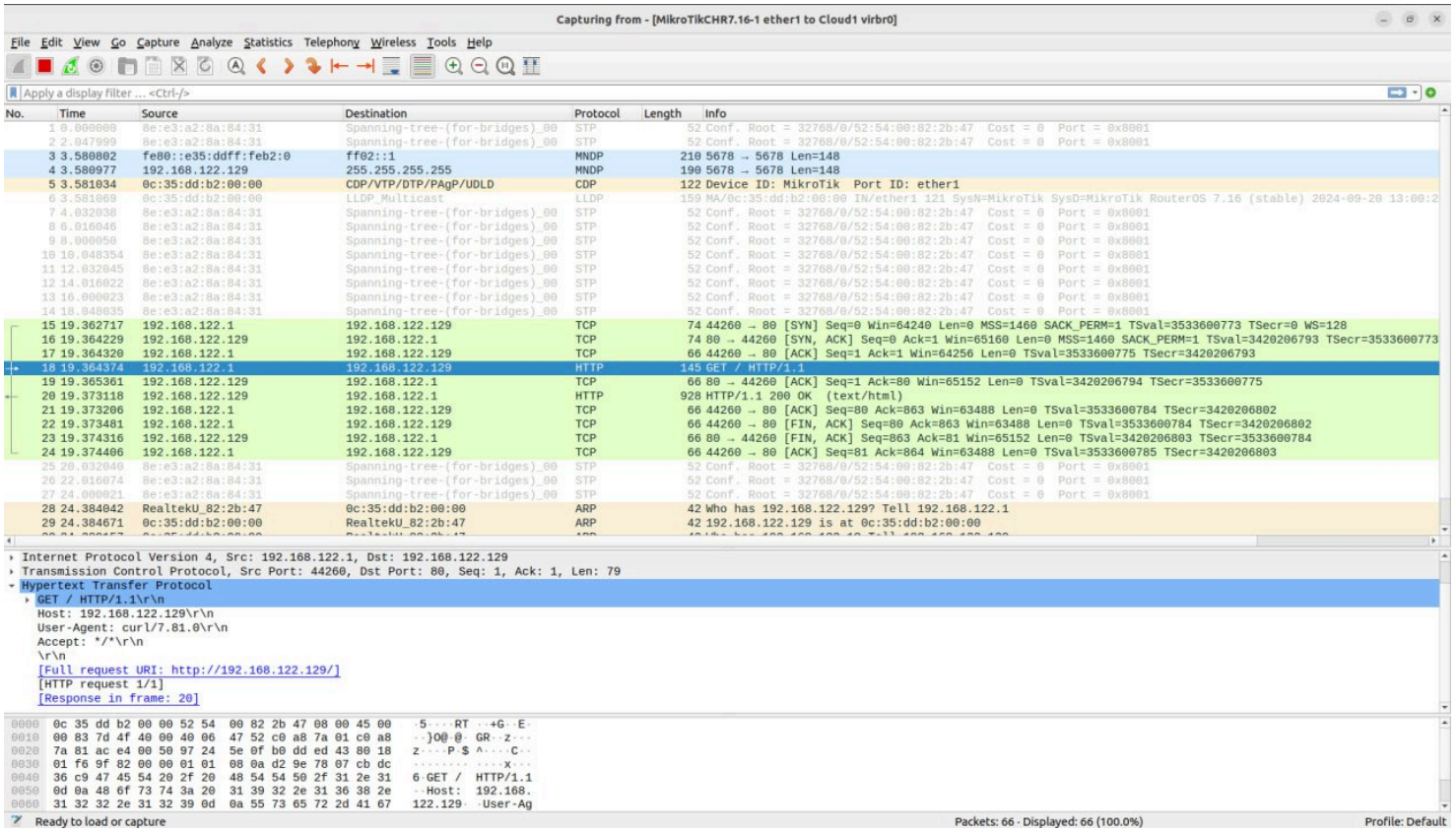
- It seems easy. Everything is clear and I see no reason to move it to appendix



Task 2 - Traffic Captures

> 1. Access your Web's http page from outside and capture the traffic between the gateway and the bridged interface.

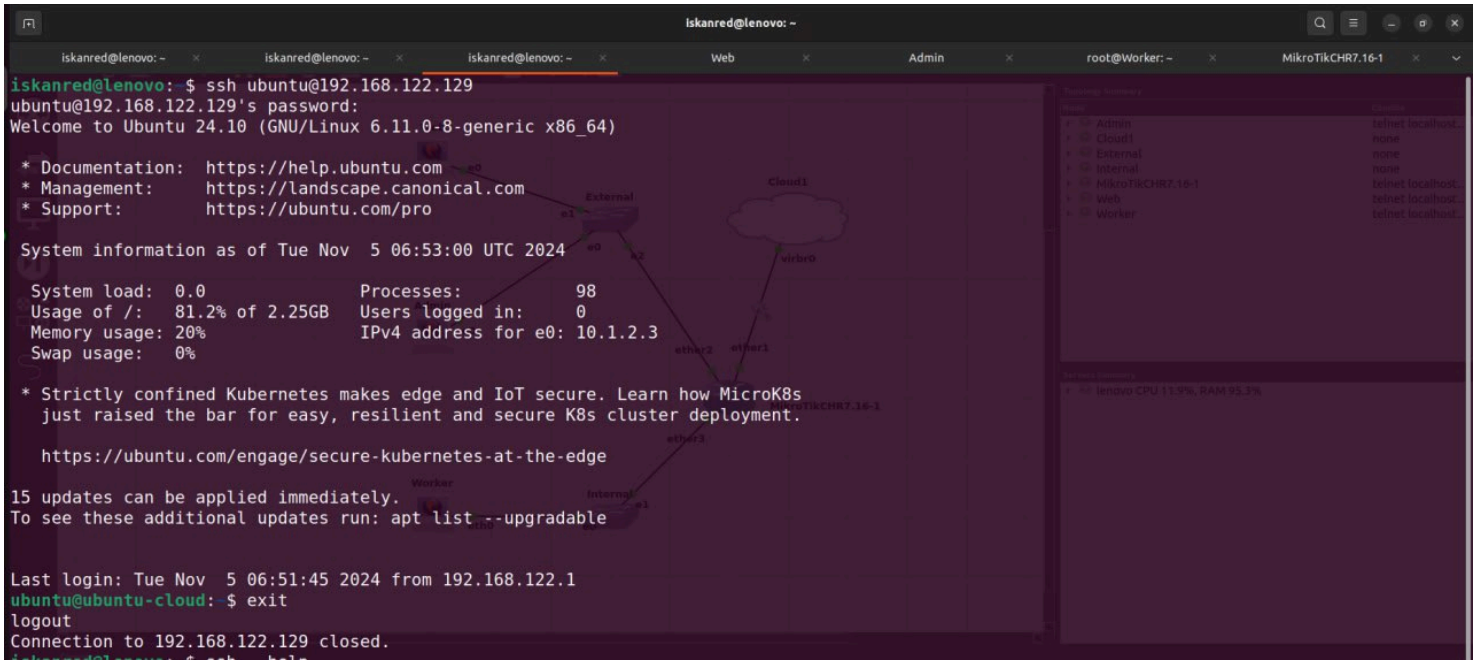
- I accessed Web's http page from outside and got the following



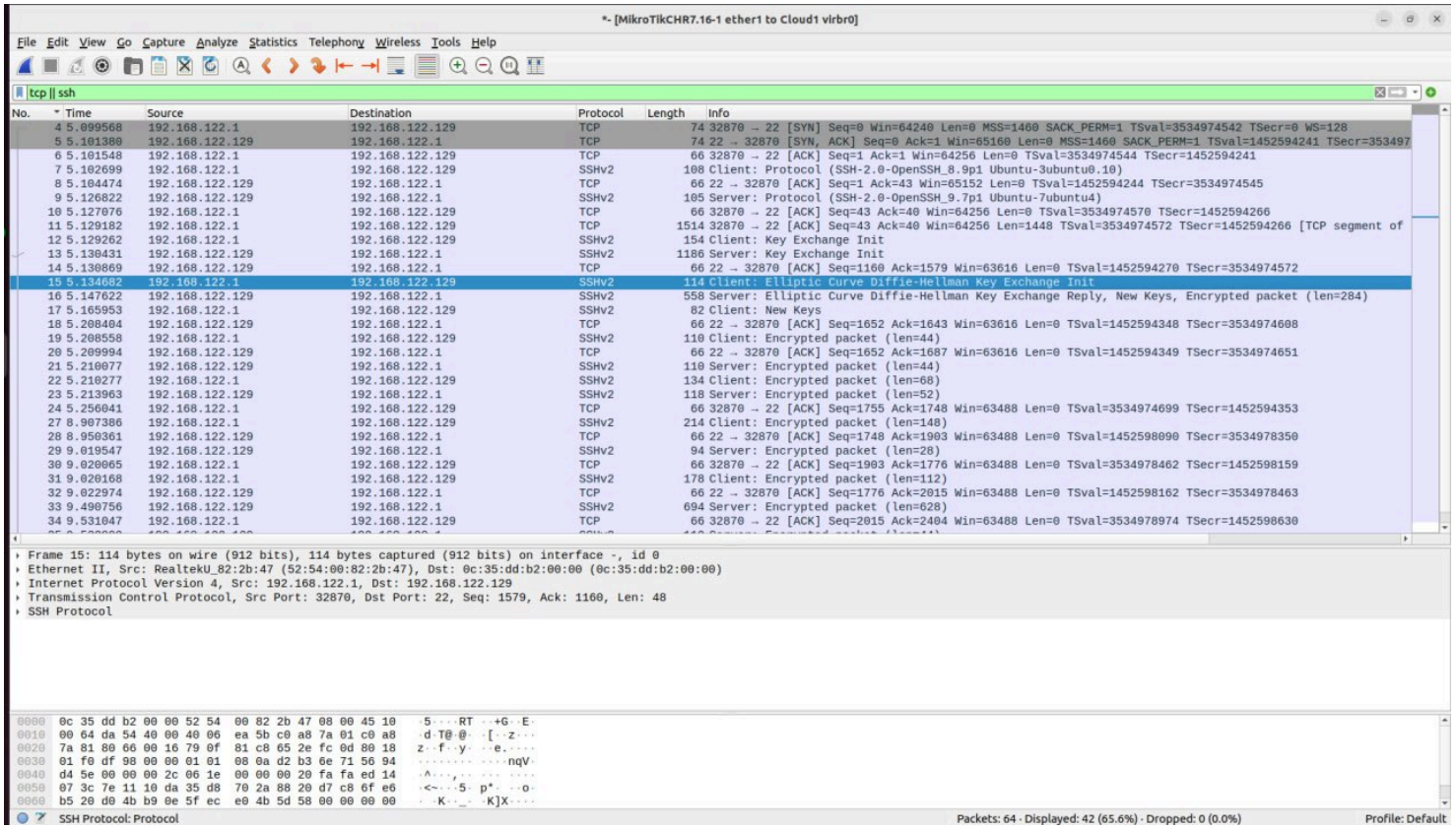
  - We can see there is a standard HTTP request and a successful response
  - We can see here TCP three-way handshake between the client (my host machine) and the server (the router that forwards requests to Web node): SYN -> SYN+ACK -> ACK,
  - Also, we can notice well-knpwn HTTP headers inside the HTTP packes: `Content-length`, `Content-type`, `Server: nginx/1.26.0 (Ubuntu)`, `Connection: keep-alive`, `Date` and etc
  - Finally, TCP closes connection: FIN+ACK -> FIN+ACK -> ACK

2. SSH to the Admin from outside and capture the traffic (make sure to start capturing before connecting to the server).

- First, I set the port for Admin SSH Server back to `22` instead of `22000`
- Now we can connect to the ssh server from outside

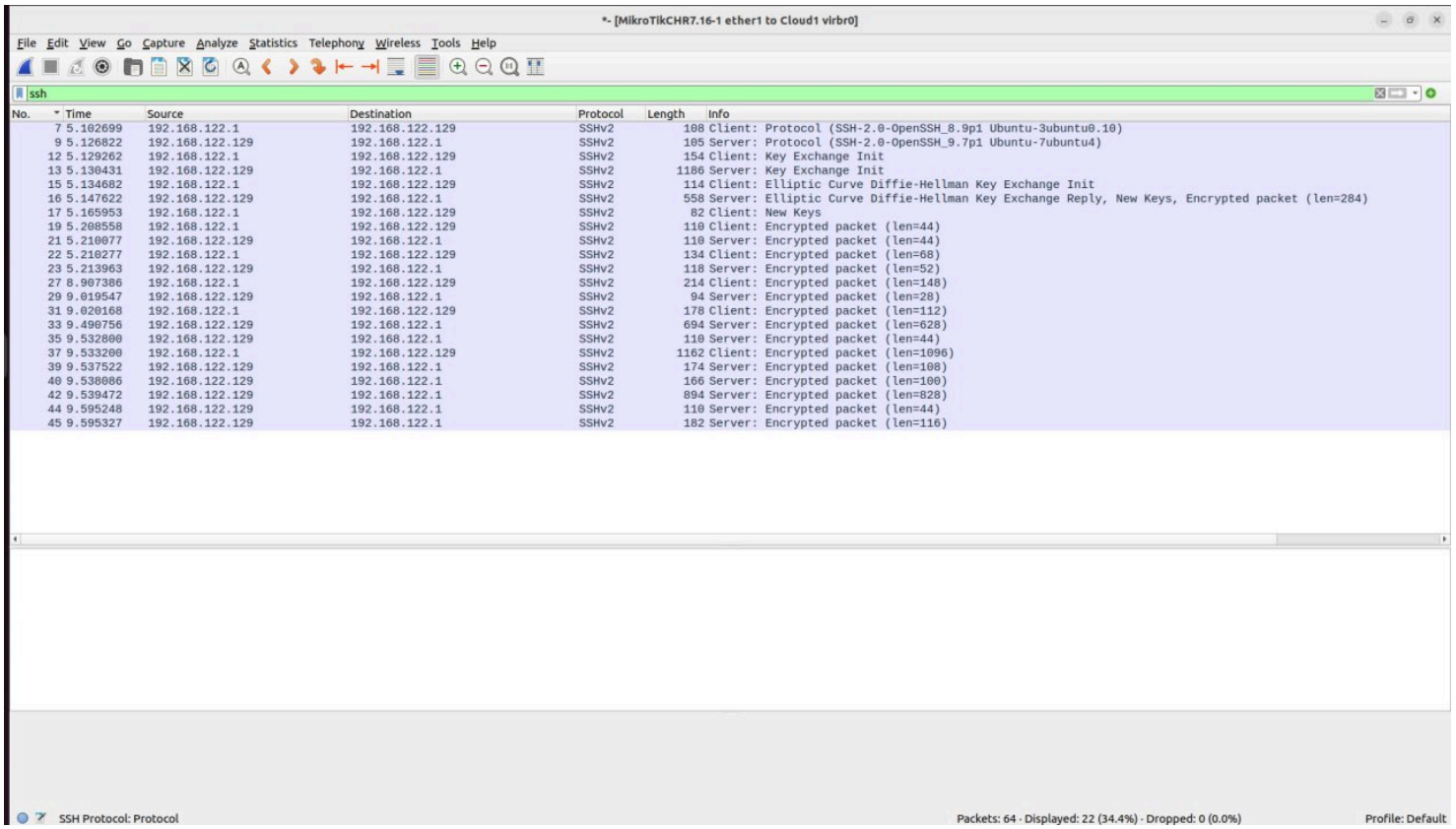


- Capturing the traffic I could observe how the SSH connection had been established



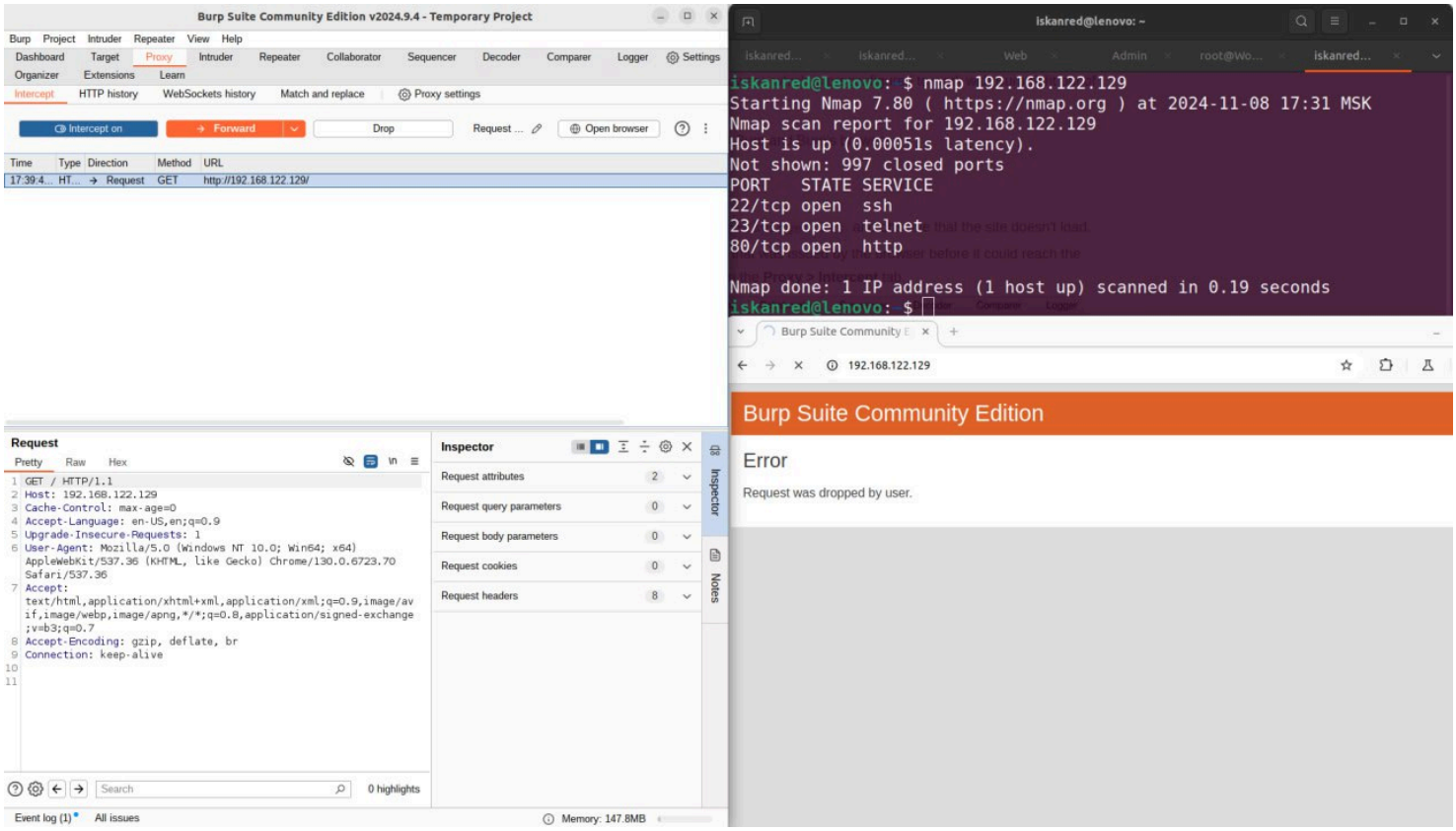
  - First, again TCP 3-way handshake
  - Then we see exchange of supported SSH protocol versions (`SSH-2.0-OpenSSH_8.9p1` and `SSH-2.0-OpenSSH_9.7p`)
  - Afterwards, there was a key exchange: client and server made a consensus on a key exchange protocol: `Elliptic Curve Diffie-Hellman Key Exchange Init`

- Now we can see the whole traffic with encrypted packets

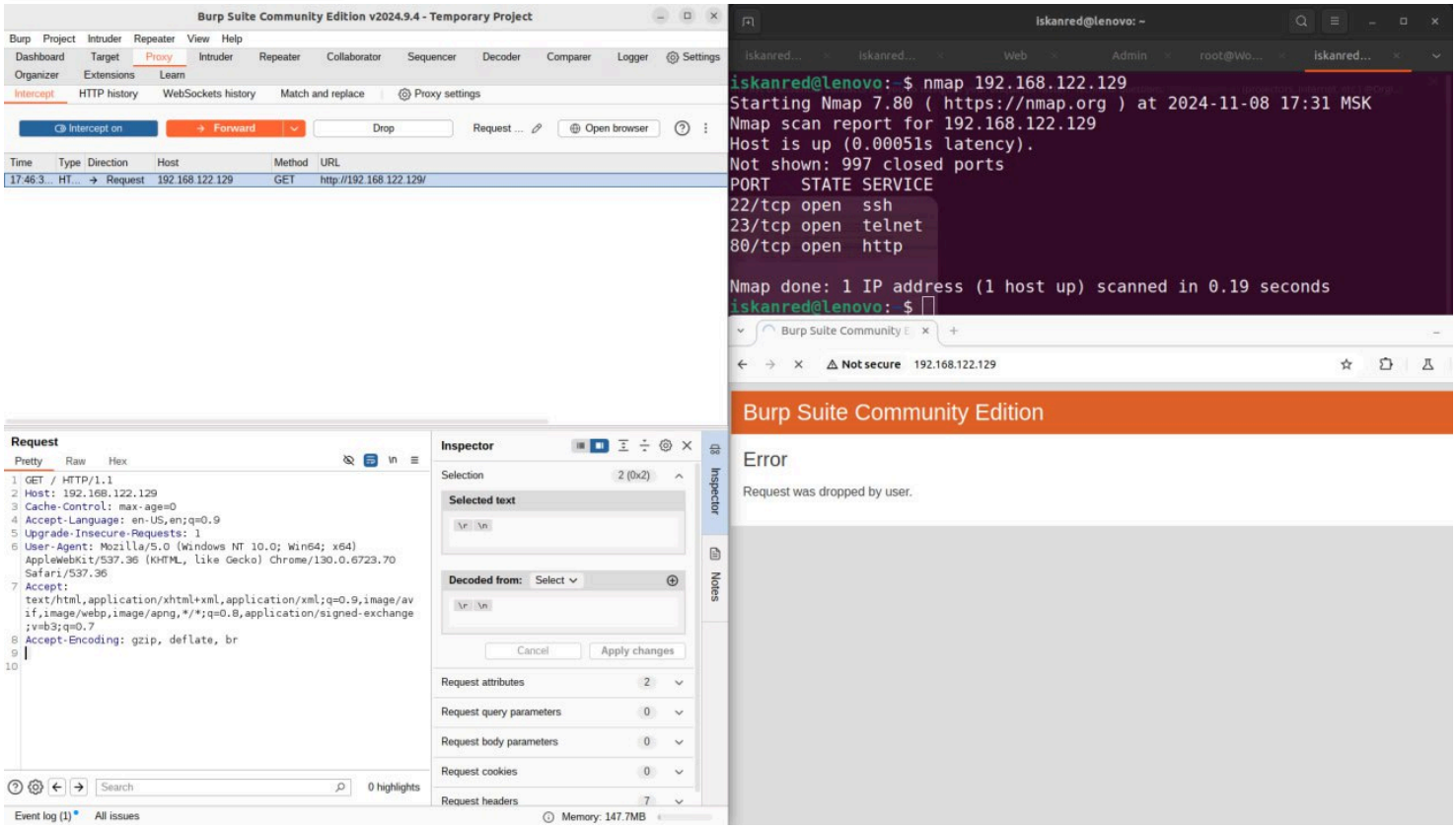


3. Configure the Burp suite as a proxy on your machine and intercept your HTTP traffic

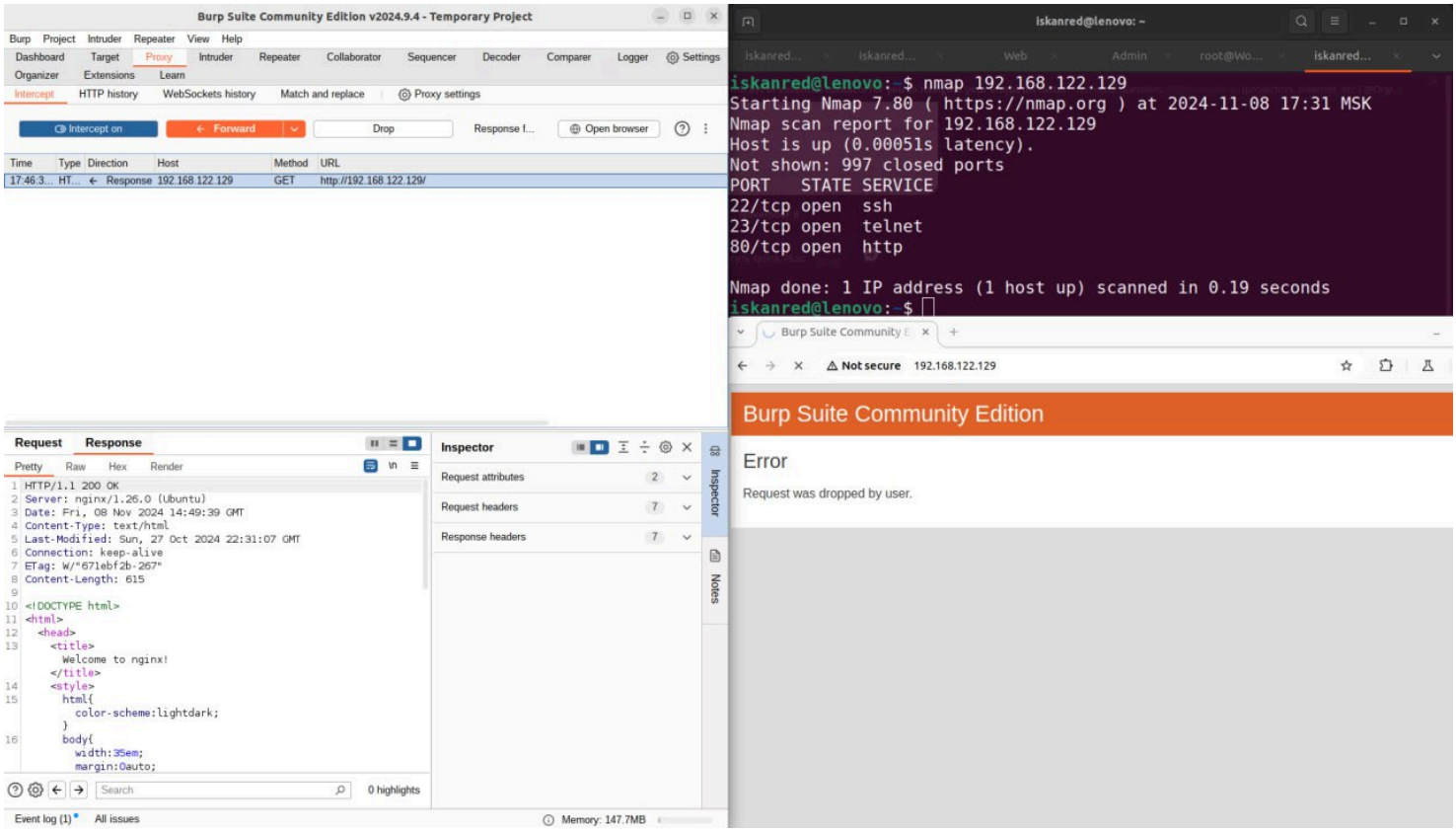
- I installed Burp Suite as Proxy and opened its browser. From this browser I tried to access the Web's page that is accessible from the outside by addressing the router:



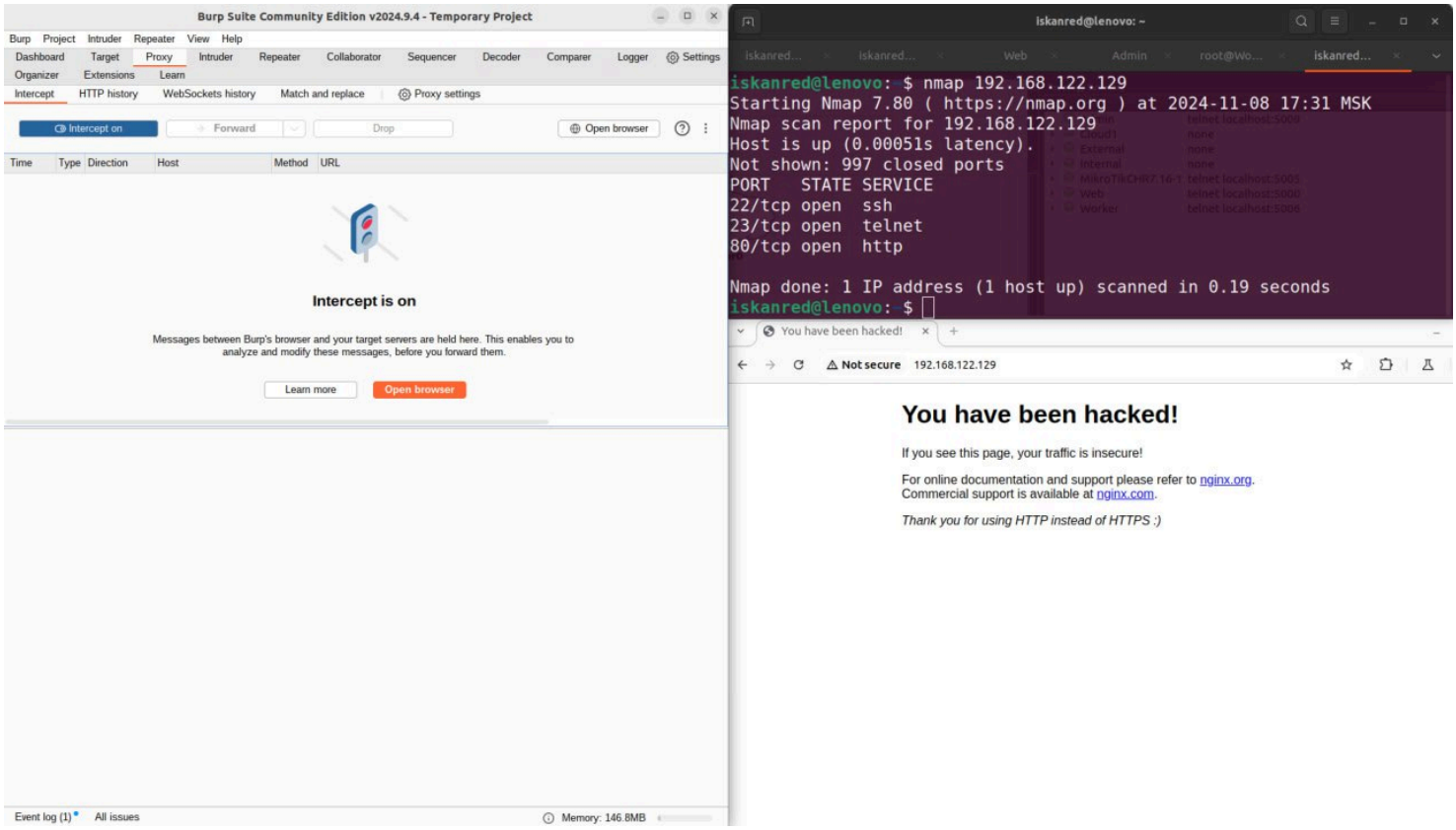
- As we can see, we can intercept the traffic and then change it as we want. For example, I removed the header `Connection: keep-alive`



- What is more, I can configure Burp Suite to intercept the response too:



- Finally, I can change the response HTML and the client will get the following output



> Why are you able to do this here and not in an SSH connection?

It's possible to change any information in a HTTP request because HTTP traffic is not secure as SSH. SSH is designed to use public key cryptography to prevent MiTM attacks and keep Integrity property.

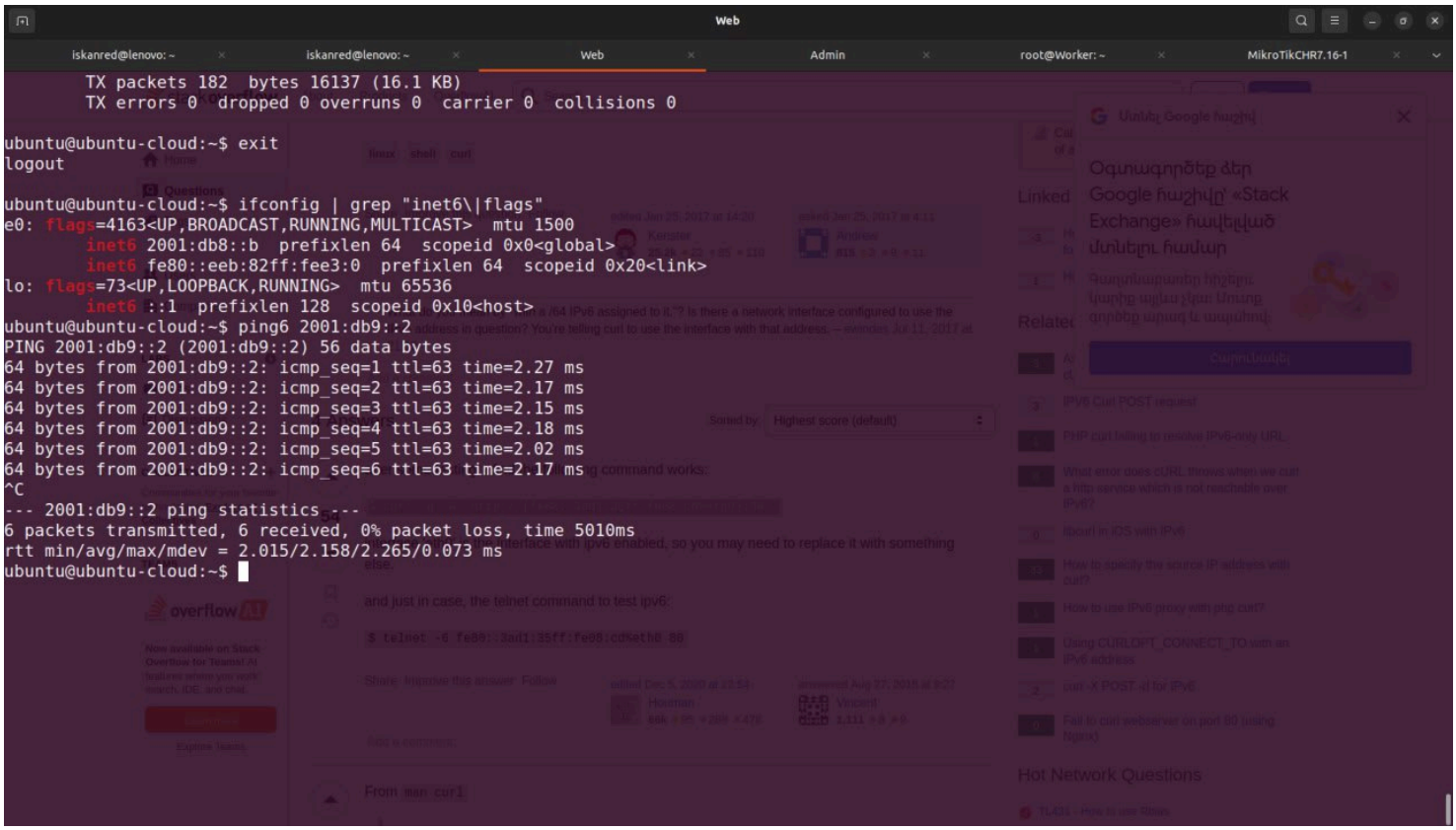
Task 3 - IPv6

Useful note:

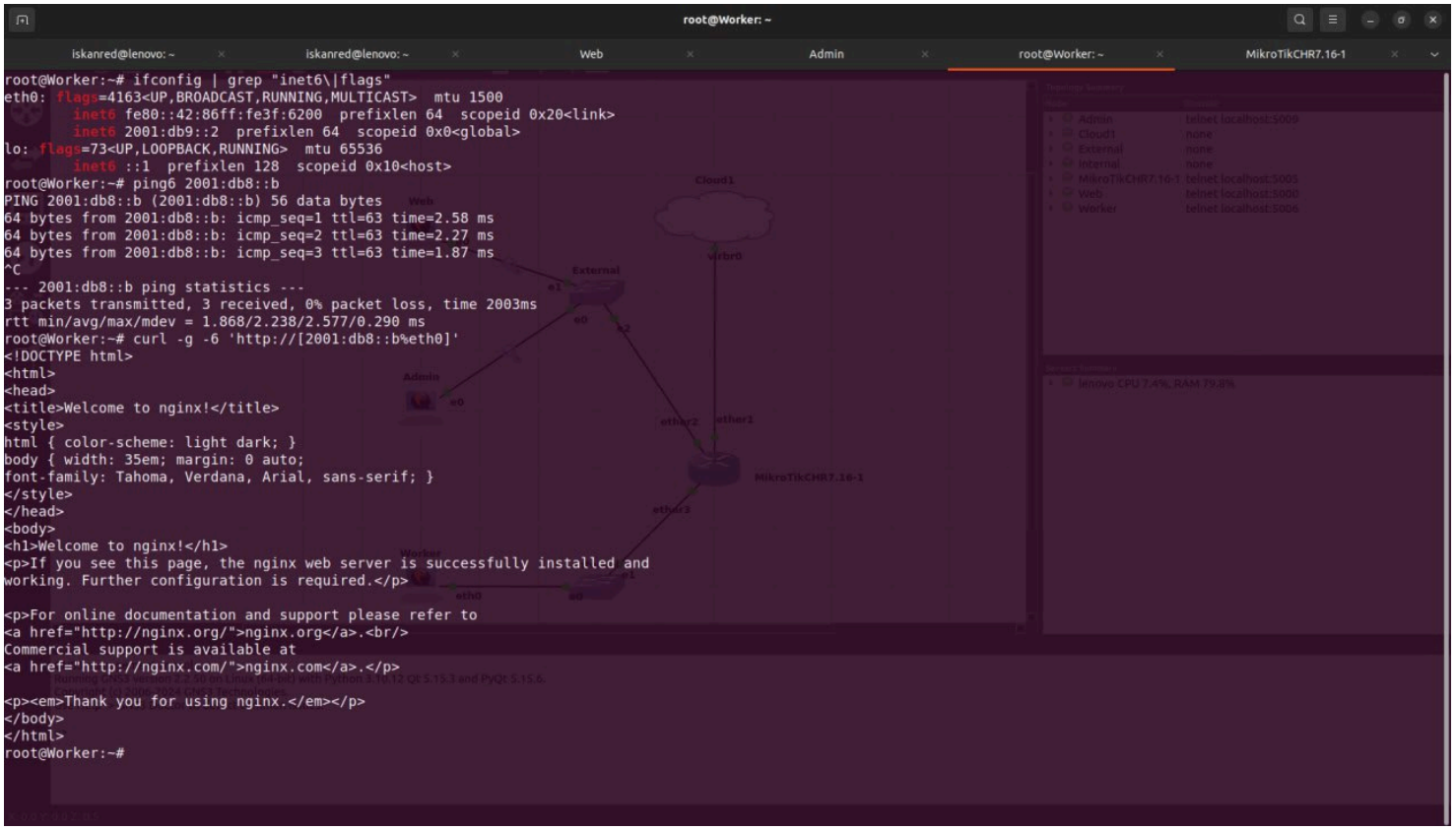
```
multicast (with prefix ff00::/8)
link-local (with prefix fe80::/10)
unique local addresses (with prefix fc00::/7)
```

> 1. Configure IPv6 from the Web Server to the Worker. This includes IPs on the servers and the default gateway

- I set the IPv6 global addresses for all the necessary nodes: Gateway (`2001:db8::1` for `bridge1` and `2001:db9::1` for `bridge2`), Worker (`2001:db9::2`) and Web (`2001:db8::a`).
- So now, I can ping Worker node from Web:

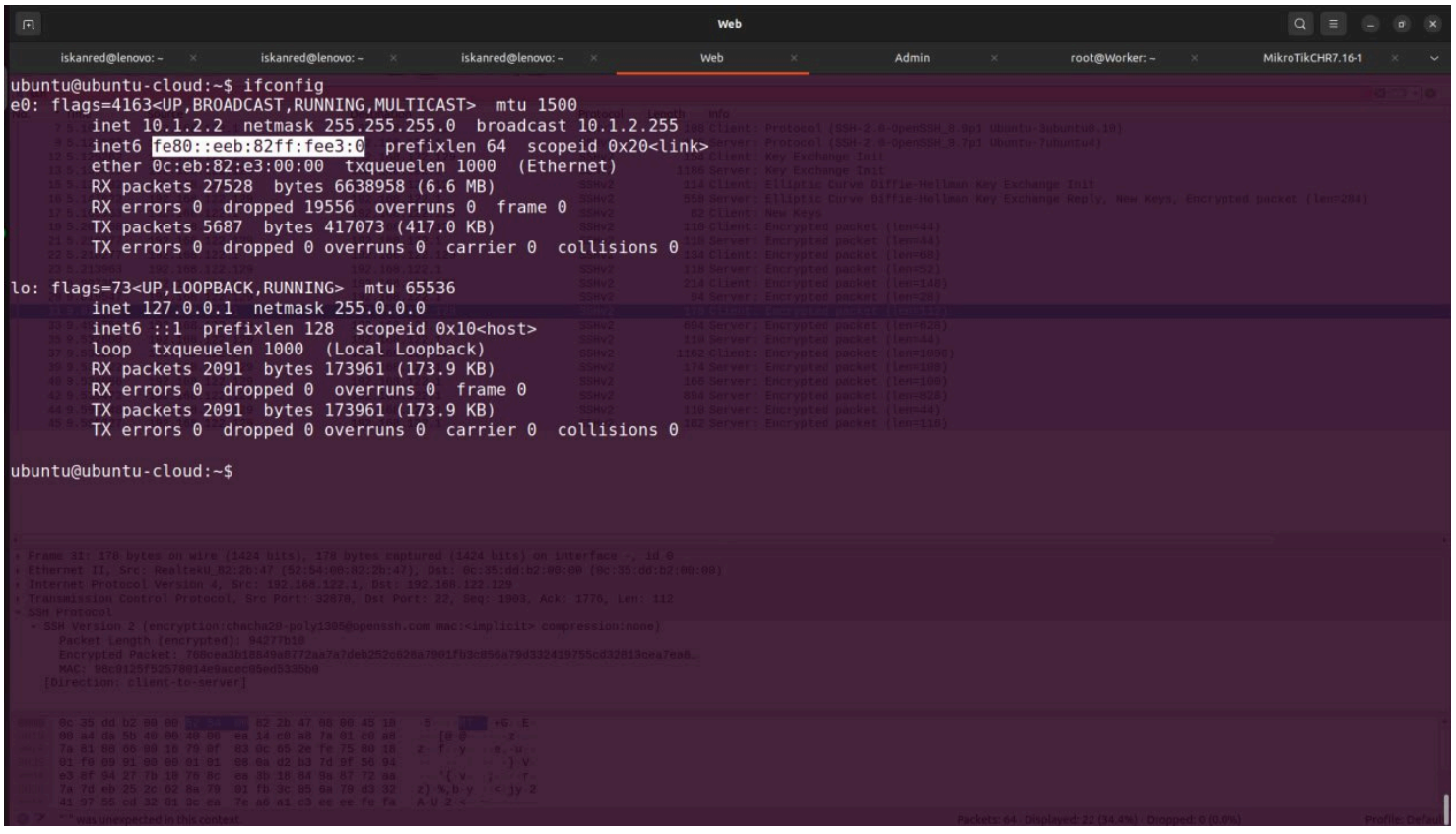


- And vice verse, not only ping though, but also access the Web's http page:

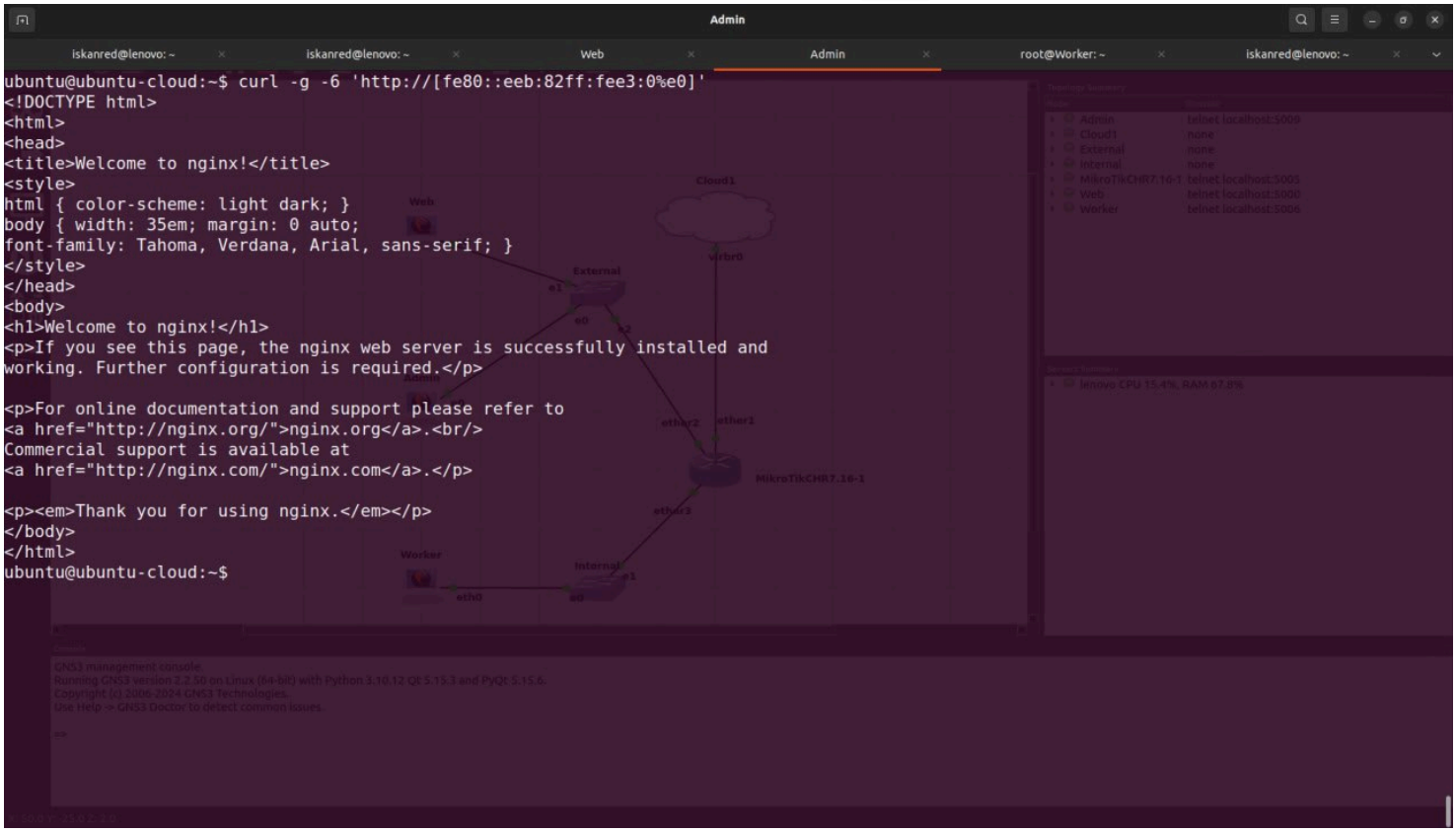


> 2. Access the Web's http page using IPv6 from Admin while capturing the traffic again. Can you see the difference? What's the difference in packages?

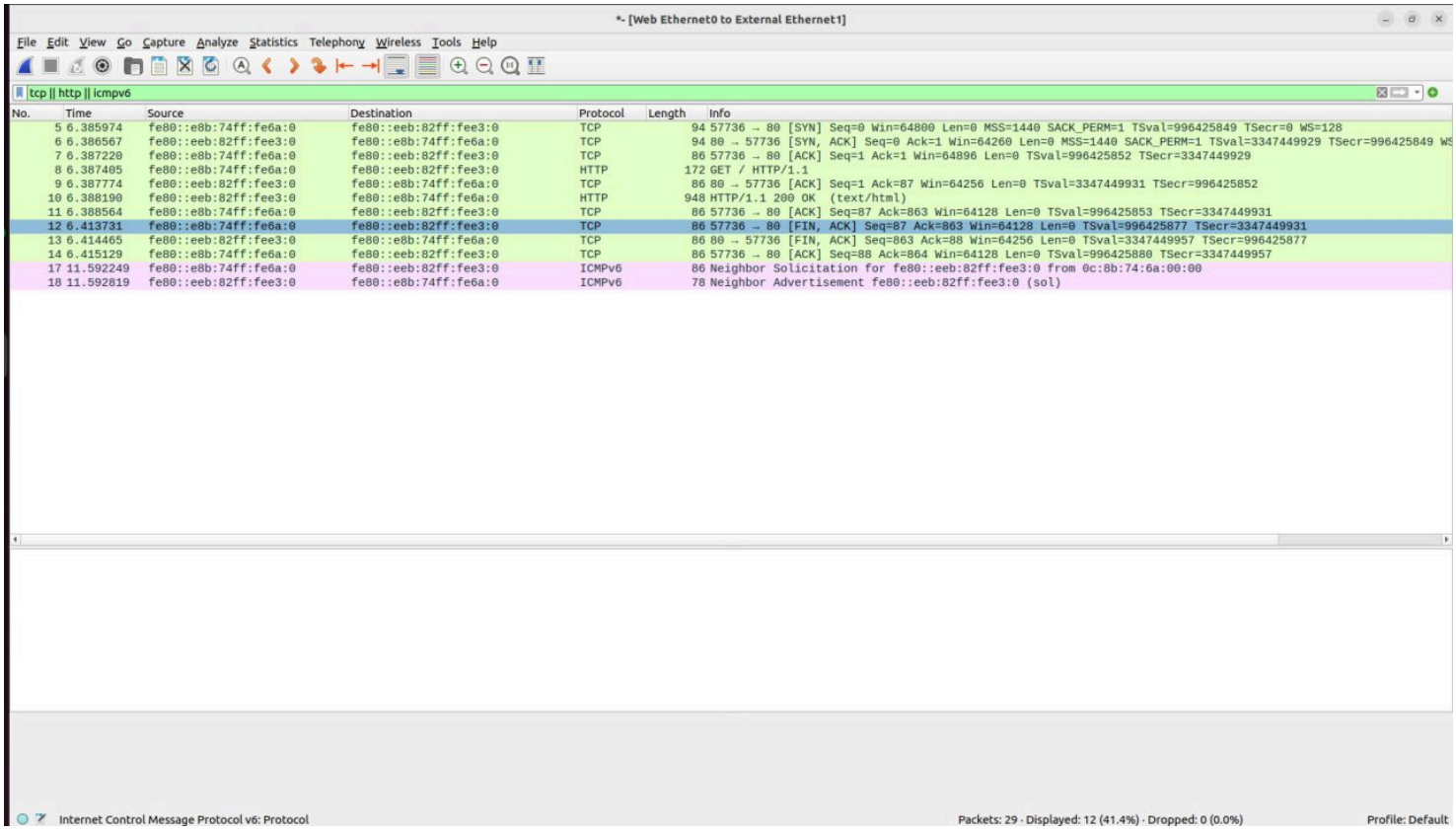
- First. let's obtain the IPv6 link-local address of the Web node (I did this part before the 1st task, so at that moment I had no Global or Unique Local addresses):



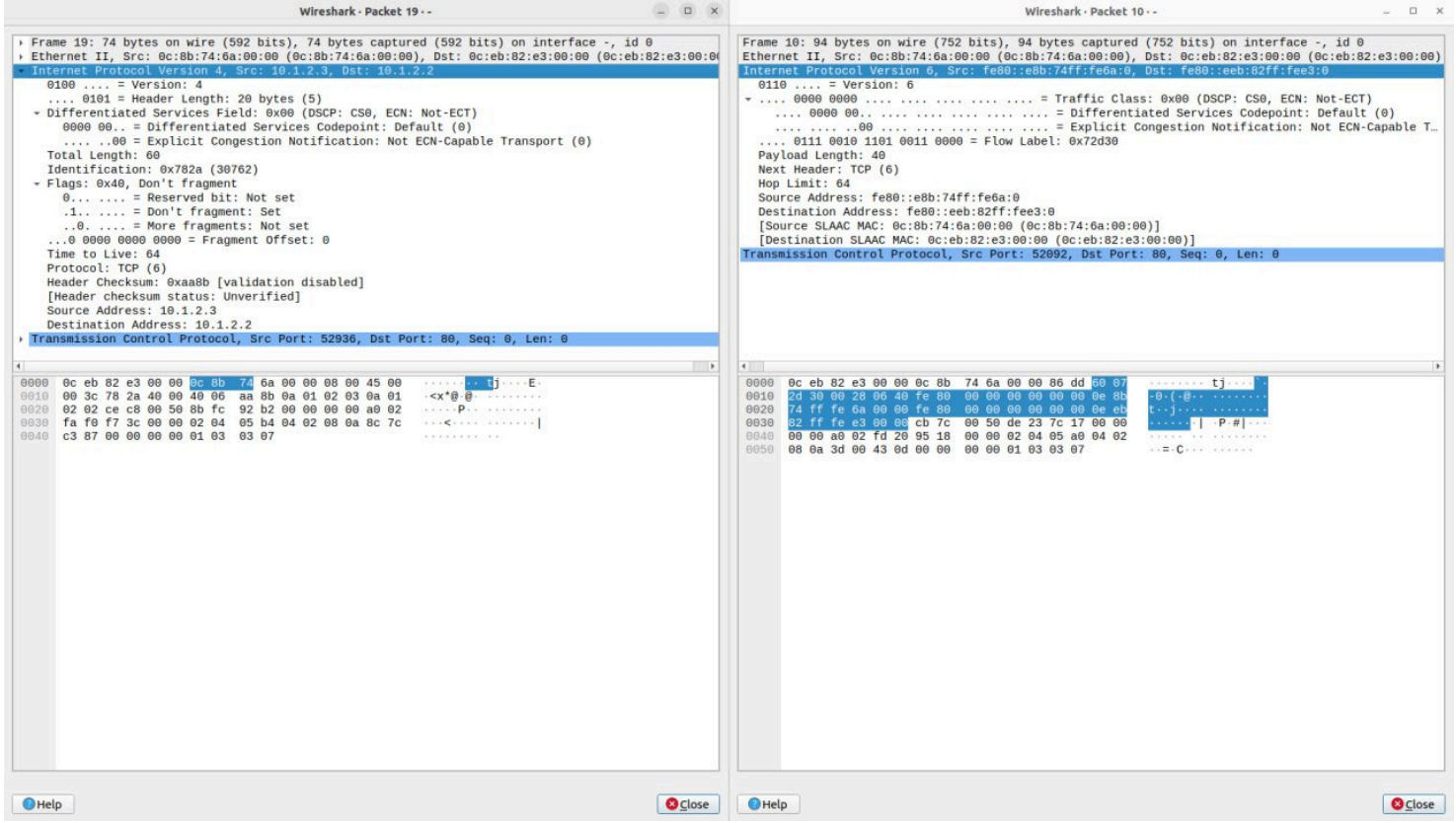
- After it we can request HTTP Web page from the Admin node using `curl`:



- Finally, we can observe the captured traffic inside the Wireshark:



- Here we can see that the `Source` and the `Destination` addresses are represented in the form of IPv6 (16 bytes). Also, we can notice using `ICMPv6` protocol for Neighbour Advertisement.
- Diving deep into the IP packet we can see that the length of `IPv6` packet is less because of absence some fields that are presented in `IPv4`:



References

- https://help.mikrotik.com/docs/spaces/ROS/pages/328151/First+Time+Configuration
- https://nmap.org/book/host-discovery.html
- https://ru.wikipedia.org/wiki/Lsof
- https://putty.org.ru/articles/netstat-linux-examples
- https://help.mikrotik.com/docs/spaces/ROS/pages/328247/IP+Addressing