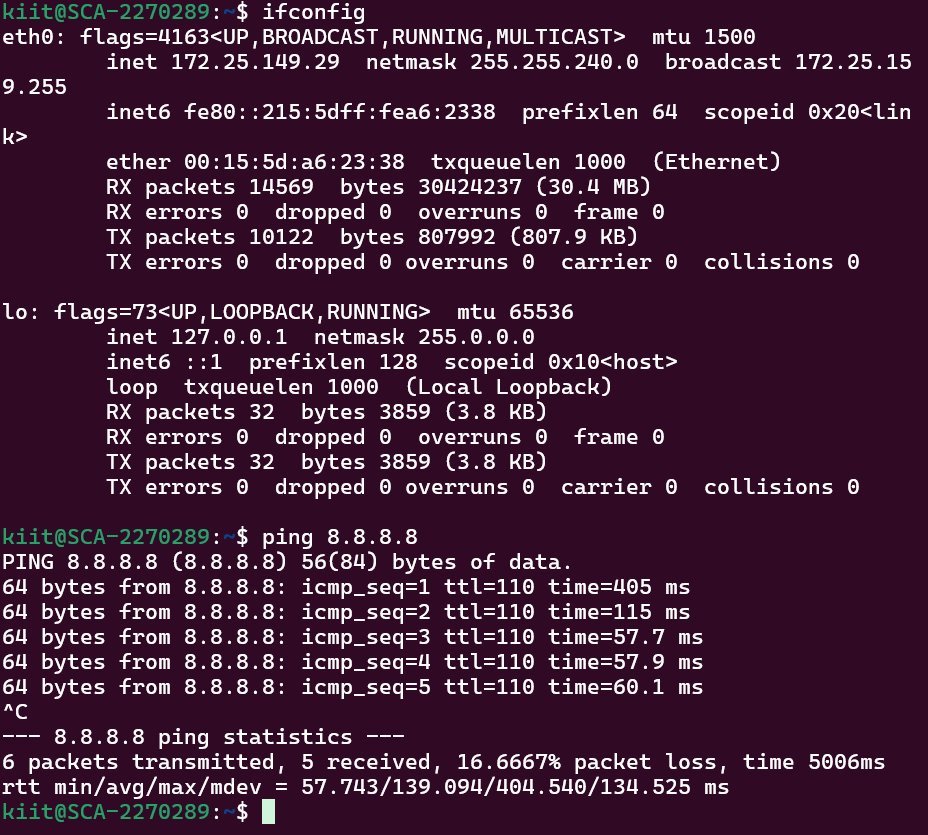
# Lab 1 : Setup and Verify Networking Configuration

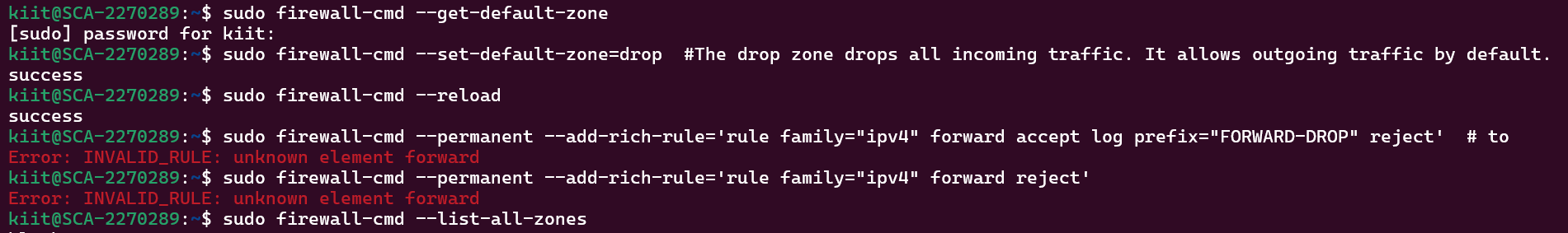
1. **Check network interfaces:**
   * Run ip addr or ifconfig to identify your network interfaces. Typically, your interface may be eth0, ens33, enp0s3, etc.
2. **Verify connectivity:**
   * Make sure you can reach the internet and other machines in your local network:
     + ping 8.8.8.8 (Google DNS server)
     + ping <your\_gateway\_ip>
     + ping <another\_machine\_in\_the\_network>
3. **Confirm that firewalld is installed:**



# Lab 2: Configure a Basic Firewall

## **Set default policies:**

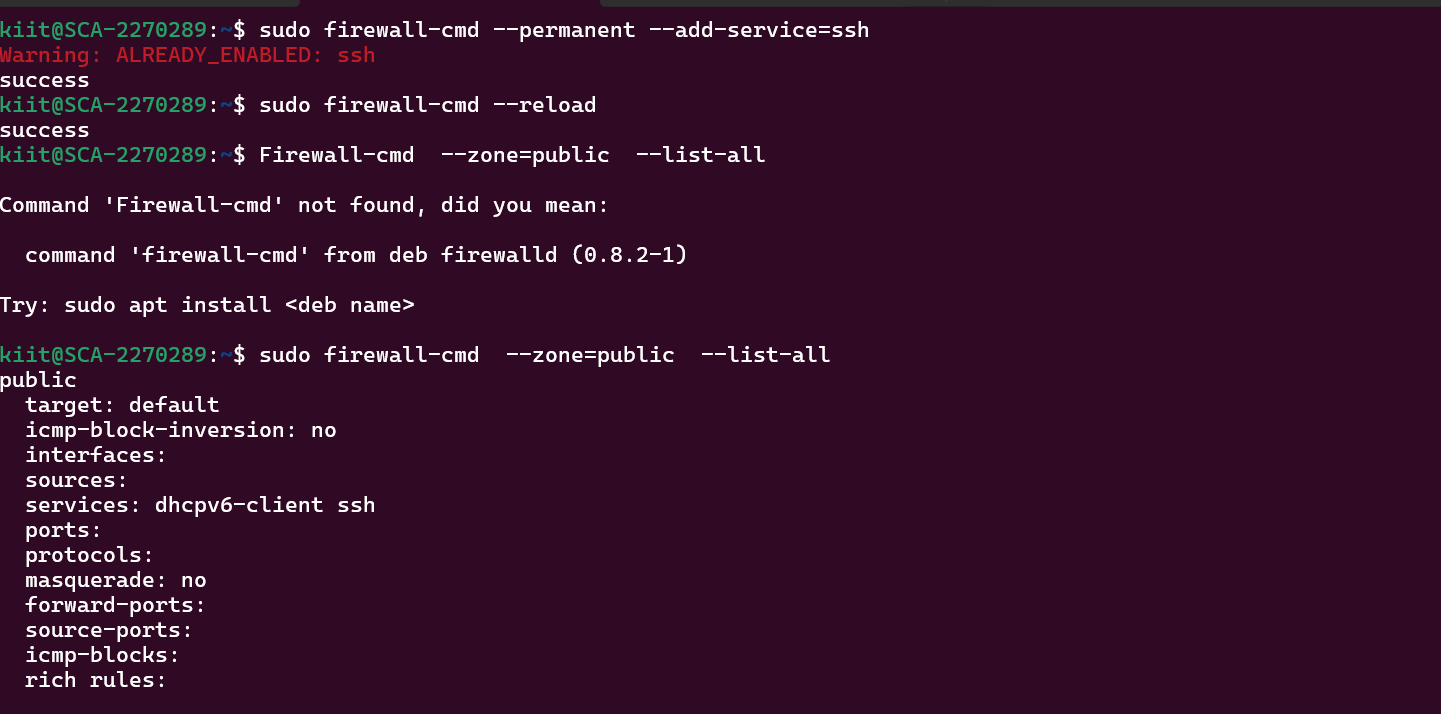
By default, we want to deny all incoming traffic and allow outgoing traffic. This can be set as follows: Block all incoming traffic, Block forwarding, Allow all outgoing traffic



## Allow established connections:

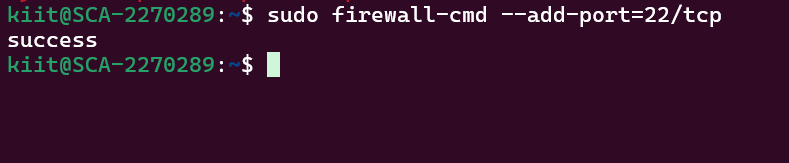
To maintain established connections (like active SSH sessions), you need to allow the related and established connections:

sudo iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT



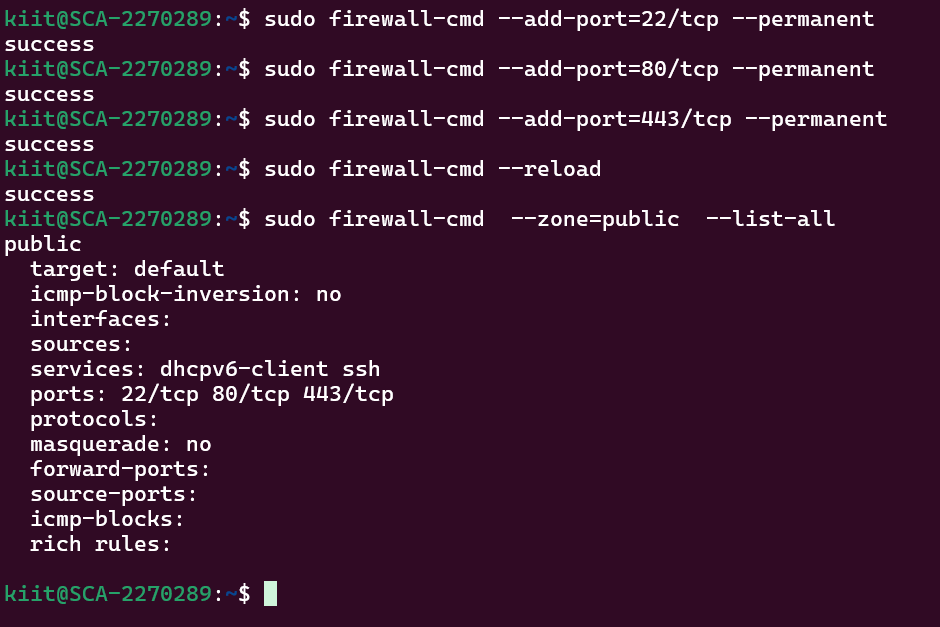
**Allow SSH access (port 22):**

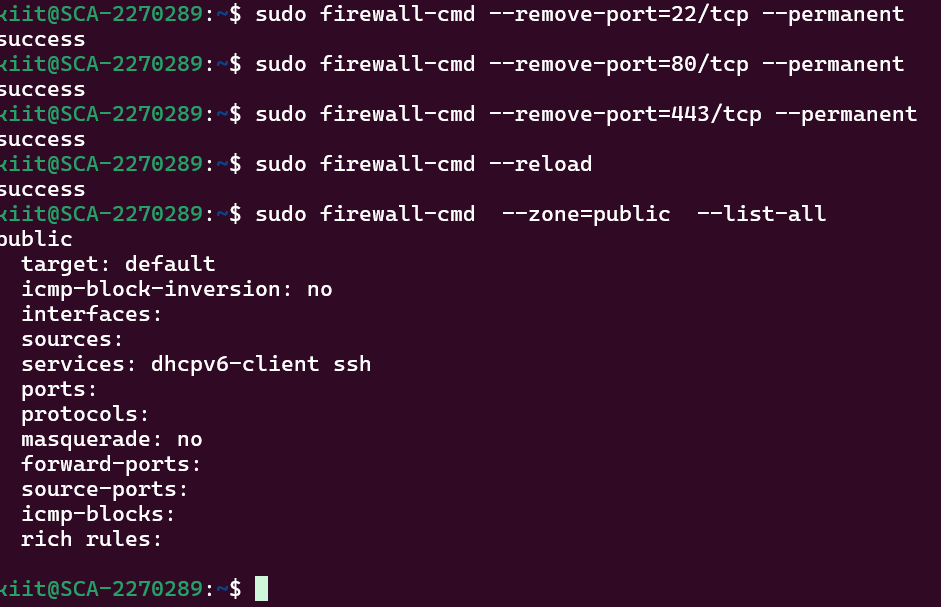
* You need to allow SSH traffic to connect remotely to the system. This is done by allowing inbound traffic on port 22:



## Allow HTTP/HTTPS (ports 80, 443):

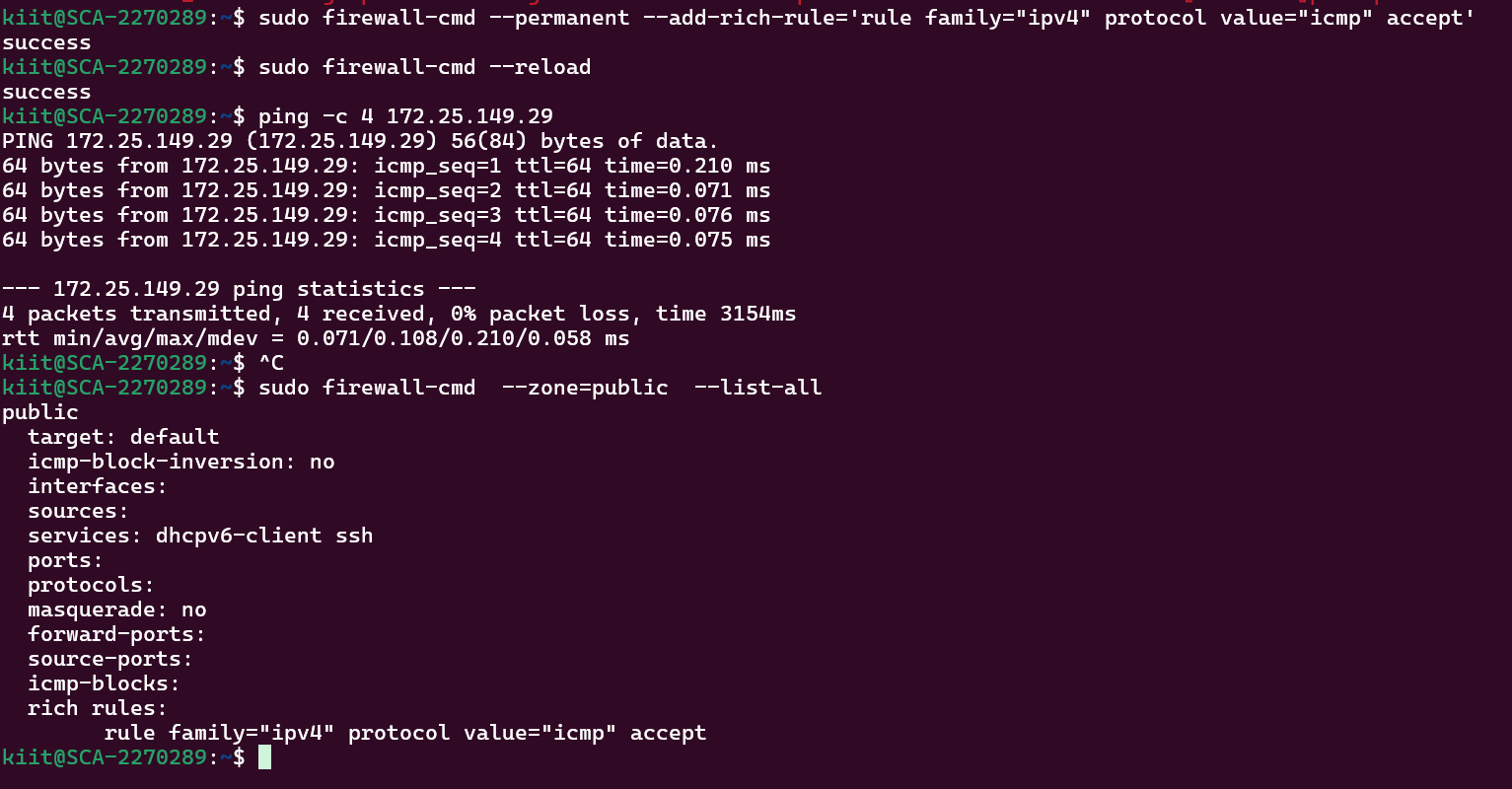
If your server will serve web pages, open HTTP and HTTPS ports:





## Allow ICMP (ping) requests:

🡪 You can allow ICMP traffic for ping functionality:



## Save your rules:

To ensure the firewall rules persist after reboot,

Solution :

sudo firewall-cmd --zone=public --permanent --save

# Part 5: Monitor and Manage Firewall :

**View current firewall rules:**

* + Check the current rules using:

sudo iptables -L

**Flush all rules (reset firewall):**

* + If you want to reset the firewall to its default state (deny all traffic):

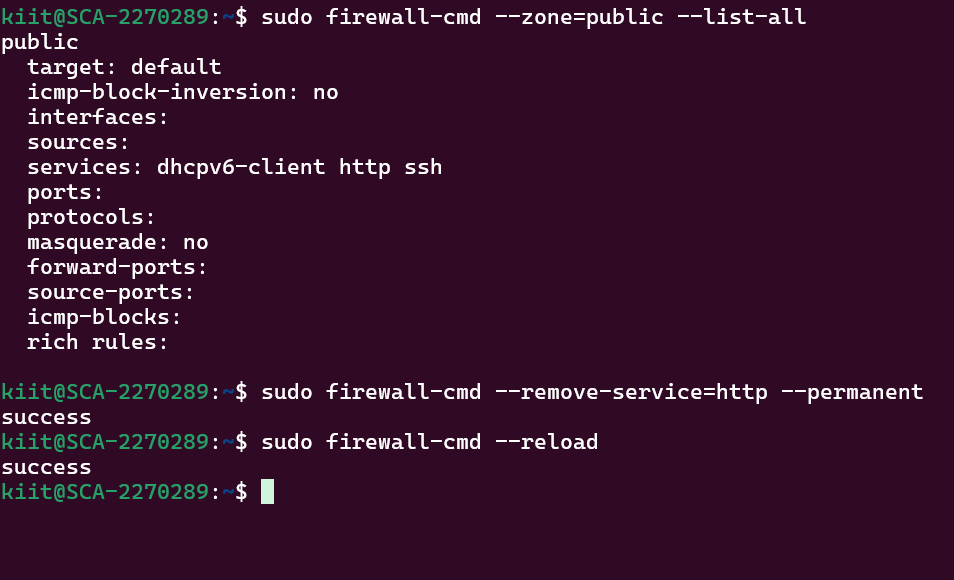
sudo iptables -F

**Delete a specific rule:**

* + If you need to delete a specific rule, use:

sudo iptables -D INPUT -p tcp --dport 80 -j ACCEPT

Solution :



# **2. Block Specific Ports**

You can block incoming traffic on specific ports using iptables.

1. **Block incoming traffic on port 80 (HTTP):**
   * This will block all HTTP traffic from reaching your server.

sudo iptables -A INPUT -p tcp --dport 80 -j DROP

1. **Block incoming traffic on port 443 (HTTPS):**
   * This will block all HTTPS traffic.

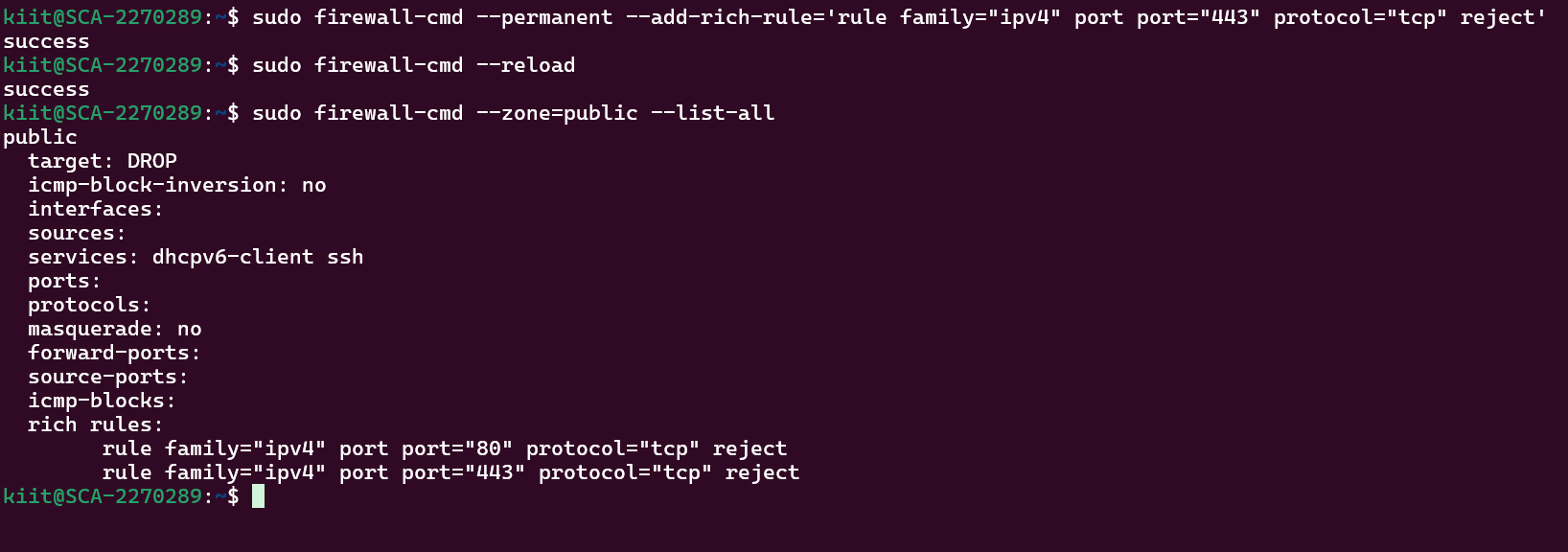
sudo iptables -A INPUT -p tcp --dport 443 -j DROP

1. **Verify the changes:**
   * List the rules to ensure that the ports are blocked.

sudo iptables -L

1. **Test port blocking:**
   * From a different machine, try to access the blocked port using curl or a browser.
   * You should not be able to reach the server on ports 80 or 443.

# 



# **3. Allow Traffic from Specific IP Addresses**

You can allow or deny traffic based on specific IP addresses.

1. **Allow SSH (port 22) only from a specific IP address (e.g., 192.168.1.100):**
2. **Block SSH access from all other IP addresses:**
3. **Verify the changes:**
   * List the rules again to confirm the changes.
4. **Test the configuration:**
   * Try to SSH into the server from 192.168.1.100 — it should work.
   * Try from any other IP — it should be blocked.

## **4. Allow Traffic from a Specific IP Range**

You can also allow traffic from a specific range of IPs. For example, if you want to allow access to your server from a range of IP addresses within the 192.168.1.0/24 subnet:

1. **Allow traffic from the IP range 192.168.1.0/24:**

sudo iptables -A INPUT -p tcp -s 192.168.1.0/24 --dport 22 -j ACCEPT

1. **Block traffic from all other IP ranges:**

sudo iptables -A INPUT -p tcp --dport 22 -j DROP

1. **Verify the rules:**

sudo iptables -L

1. **Test the configuration:**
   * Try accessing the server from an IP within the 192.168.1.0/24 range — it should work.
   * Try accessing from an outside range — it should be blocked.

# 

## **5. Allow Specific Protocols (TCP, UDP, ICMP)**

You can allow or block specific network protocols (e.g., TCP, UDP, ICMP).

**Allow all incoming ICMP (Ping) requests:**

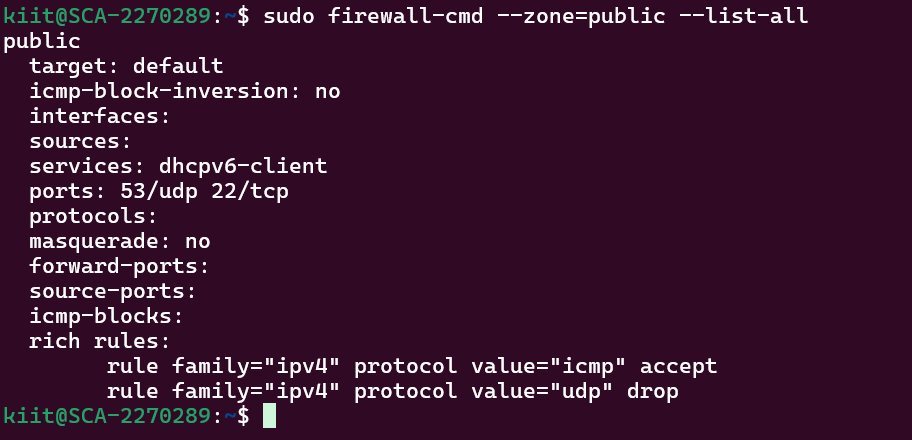
**Allow incoming UDP traffic on port 53 (DNS):**

**Allow incoming TCP traffic on port 22 (SSH):**

**Block UDP traffic**

1. **Verify the changes:** List the rules again to ensure all protocols and ports are configured as needed.
2. **Test the configurations:**
   * Test ICMP by pinging the server.
   * Test UDP and TCP services using tools like nc, ping, or curl to ensure proper functionality.

# 



## **Allow Specific IP and Port Combinations**

You can also allow specific combinations of IP and port.

1. **Allow traffic from 192.168.1.100 to port 22 (SSH):**
2. **Allow traffic from 192.168.1.0/24 to port 80 (HTTP):**
3. **Block all other IP addresses from accessing port 80**

# 