Blocking an IP address on a Linux system can be done in several ways, depending on what exactly you want to achieve and the tools you prefer to use. Here are some common methods:

1. **Using `iptables`**

`iptables` is a powerful tool for managing network traffic on Linux. To block an IP address using `iptables`, follow these steps:

1. **Check Current Rules**:
   ```bash
   sudo iptables -L -n
   ```

2. **Block the IP Address**:
   Replace `X.X.X.X` with the IP address you want to block.
   ```bash
   sudo iptables -A INPUT -s X.X.X.X -j DROP
   ```

3. **Save the Rules**:
   To ensure that the rules persist after a reboot, save the rules to a file. The method for doing this can vary depending on your Linux distribution:

   - **Debian/Ubuntu**:
     ```bash
     sudo iptables-save > /etc/iptables/rules.v4
     ```
   - **Red Hat/CentOS**:
     ```bash
     sudo service iptables save
     ```

4. **Verify the Rule**:
   ```bash
   sudo iptables -L -n
   ```

2. **Using `nftables`**

`nftables` is the successor to `iptables` and is used in modern Linux distributions. Here's how to block an IP using `nftables`:

1. **Check if `nftables` is Installed**:
   ```bash
   nft --version
   ```

2. **Create a Table and Chain** (if not already created):
   ```bash
   sudo nft add table ip filter
   sudo nft add chain ip filter input { type filter hook input priority 0; }
   ```

3. **Add the Rule to Block the IP**:
   ```bash
   sudo nft add rule ip filter input ip saddr X.X.X.X drop
   ```

4. **Save the Rules**:
   Save the configuration to a file so that it persists across reboots:
   ```bash
   sudo nft list ruleset > /etc/nftables.conf
   ```

5. **Verify the Rule**:
   ```bash
   nft list ruleset
   ```

3. **Using `firewalld`**

If your system uses `firewalld`, you can block an IP address with it:

1. **Check the Current Rules**:
   ```bash
   sudo firewall-cmd --list-all
   ```

2. **Block the IP Address**:
   ```bash
   sudo firewall-cmd --permanent --add-rich-rule='rule family="ipv4" source address="X.X.X.X" drop'
   ```

3. **Reload `firewalld` to Apply Changes**:
   ```bash
   sudo firewall-cmd --reload
   ```

4. **Verify the Rule**:
   ```bash
   sudo firewall-cmd --list-all
   ```

4. **Using `/etc/hosts.deny`**

For blocking access to specific services (e.g., SSH), you can use `/etc/hosts.deny`:

1. **Edit the File**:
   ```bash
   sudo nano /etc/hosts.deny
   ```

2. **Add the Blocking Rule**:
   To block an IP address for a service (e.g., SSH):
   ```
   sshd: X.X.X.X
   ```

3. **Save and Exit**.

4. **Verify the Rule**:
   Check that the service is indeed denied from the specified IP.

Notes:
- Always ensure that you do not lock yourself out of your system when applying firewall rules, especially if you are working remotely.
- Depending on the Linux distribution and version, the exact commands and configuration file paths might differ. Always refer to the official documentation for your specific distribution.

Choose the method that best fits your needs and the tools you are using on your system.

To unblock an IP address that you previously blocked on a Linux system, you need to remove the rule that was used to block it. The steps depend on the tool or method you used to block the IP. Here's how you can unblock an IP address using different tools:

1. **Using `iptables`**

If you blocked the IP using `iptables`, you can remove the rule as follows:

1. **List Current Rules**:
   ```bash
   sudo iptables -L -n --line-numbers
   ```

2. **Identify the Rule Number**:
   Find the rule number associated with the IP you want to unblock.

3. **Delete the Rule**:
   Replace `NUM` with the rule number you identified.
   ```bash
   sudo iptables -D INPUT NUM
   ```

4. **Save the Changes**:
   Save the updated rules to ensure they persist after reboot:
   - **Debian/Ubuntu**:
     ```bash
     sudo iptables-save > /etc/iptables/rules.v4
     ```
   - **Red Hat/CentOS**:
     ```bash
     sudo service iptables save
     ```

2. **Using `nftables`**

If you used `nftables` to block the IP, you can remove the rule like this:

1. **List the Current Ruleset**:
   ```bash
   nft list ruleset
   ```

2. **Remove the Rule**:
   Replace `CHAIN_NAME` with the name of your chain, and `X.X.X.X` with the IP address you blocked.
   ```bash
   sudo nft delete rule ip filter input ip saddr X.X.X.X drop
   ```

3. **Save the Changes**:
   Save the updated configuration:
   ```bash
   sudo nft list ruleset > /etc/nftables.conf
   ```

3. **Using `firewalld`**

If you used `firewalld` to block the IP address, you can remove the rule as follows:

1. **Remove the Rich Rule**:
   Replace `X.X.X.X` with the IP address you want to unblock.
   ```bash
   ```

   sudo firewall-cmd --permanent --remove-rich-rule='rule family="ipv4" source address="X.X.X.X" drop'

2. **Reload `firewalld`**:
   ```bash
   sudo firewall-cmd --reload
   ```

3. **Verify the Change**:
   ```bash
   sudo firewall-cmd --list-all
   ```

4. **Using `/etc/hosts.deny`**

If you blocked the IP using `/etc/hosts.deny`, you can unblock it by removing the entry from the file:

1. **Edit the File**:
   ```bash
   sudo nano /etc/hosts.deny
   ```

2. **Remove the Blocking Rule**:
   Find and delete the line that blocks the IP. For example, remove:
   ```
   sshd: X.X.X.X
   ```

3. **Save and Exit**.

Notes:
- Always double-check the rules and ensure you're modifying the correct entries to avoid accidentally affecting other network traffic.
- Remember to review your configuration and verify that the IP address is no longer blocked after making changes.

Each method's commands and file paths might vary slightly based on your specific Linux distribution and version, so adjust accordingly.