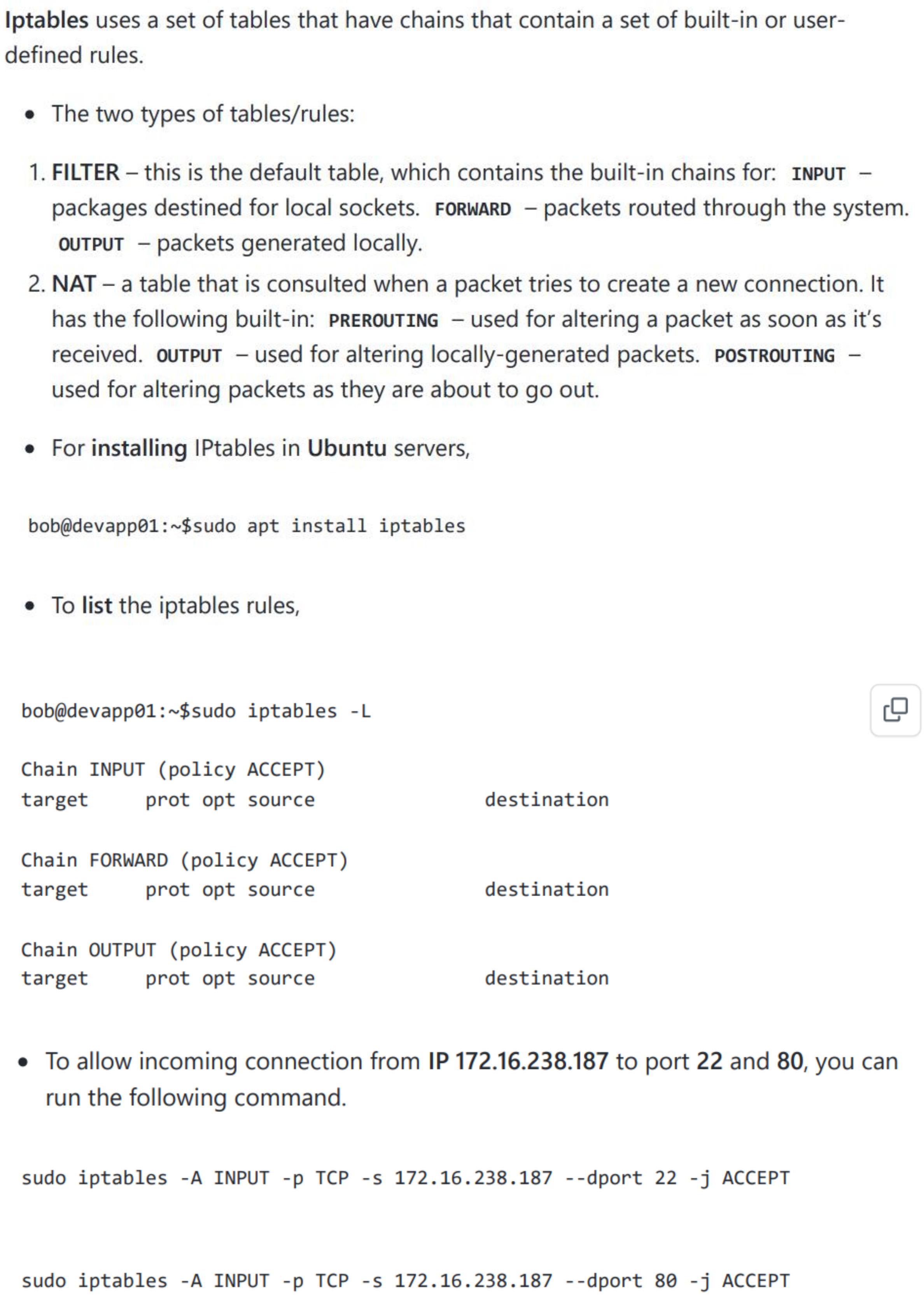
IP TABLES



Iptables uses a set of tables that have chains that contain a set of built-in or user-defined rules.

- The two types of tables/rules:

1. **FILTER** – this is the default table, which contains the built-in chains for: `INPUT` – packages destined for local sockets. `FORWARD` – packets routed through the system. `OUTPUT` – packets generated locally.
2. **NAT** – a table that is consulted when a packet tries to create a new connection. It has the following built-in: `PREROUTING` – used for altering a packet as soon as it's received. `OUTPUT` – used for altering locally-generated packets. `POSTROUTING` – used for altering packets as they are about to go out.

- For **installing** IPtables in **Ubuntu** servers,

```
bob@devapp01:~$sudo apt install iptables
```

- To **list** the iptables rules,

```
bob@devapp01:~$sudo iptables -L

Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

- To allow incoming connection from **IP 172.16.238.187** to port **22** and **80**, you can run the following command.

```
sudo iptables -A INPUT -p TCP -s 172.16.238.187 --dport 22 -j ACCEPT


sudo iptables -A INPUT -p TCP -s 172.16.238.187 --dport 80 -j ACCEPT
```

Deepak

The -A or --append option appends the rule at the end of the selected chain. The -s or --source option Source specification. The -j, --jump option specifies the target of the rule. The -p, --protocol option defines protocol of the rule or the packet to check The --dport or --destination-port refers to the destination port. The --sport or --source-port refers to source port.

- To list the **iptables rules**,

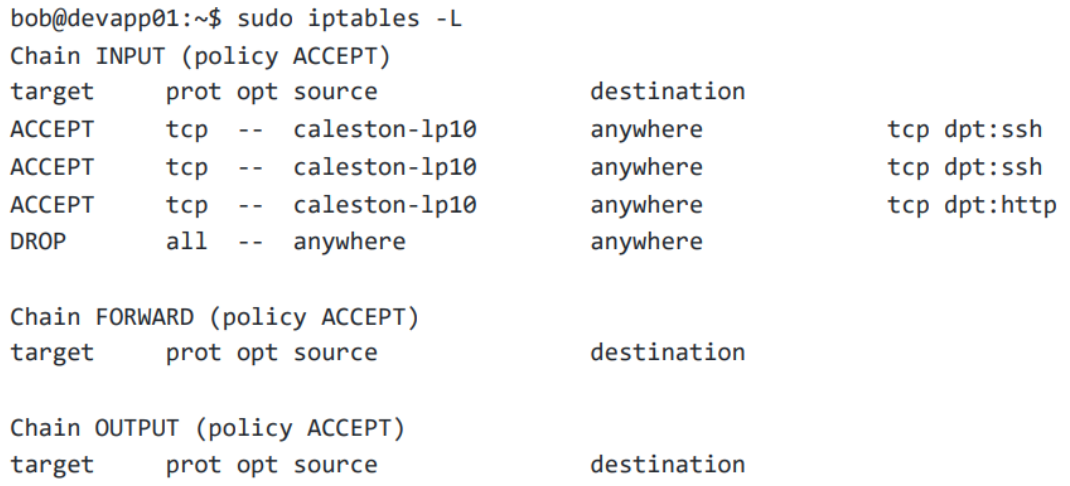
```
bob@devapp01:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp  --  caleston-lp10        anywhere             tcp dpt:ssh
ACCEPT     tcp  --  caleston-lp10        anywhere             tcp dpt:http

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

- To drop **incoming connections** from any **source** on any **destination port** for any **protocol**

```
bob@devapp01:~$sudo iptables -A INPUT -j DROP
```



```
bob@devapp01:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp  --  caleston-lp10        anywhere             tcp dpt:ssh
ACCEPT     tcp  --  caleston-lp10        anywhere             tcp dpt:ssh
ACCEPT     tcp  --  caleston-lp10        anywhere             tcp dpt:http
DROP       all  --  anywhere             anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

Difference between **DROP** and **REJECT** Both DROP and REJECT prohibits packets from passing through the firewall. But, the main difference between them is the response message.

When we use the DROP command, it will not forward the packet or answer it. But, simply drops the packet silently.

And, no indication is sent to the client or server.

But, the REJECT command sends an error message back to the source indicating a connection failure.

- To block outgoing traffic to any destination on **port 80**

```
bob@devapp01:~$sudo iptables -A OUTPUT -p tcp --dport 80 -j DROP
```

This will add rule in the **OUTPUT** chain

```
bob@devapp01:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp  --  caleston-lp10        anywhere             tcp dpt:ssh
ACCEPT     tcp  --  caleston-lp10        anywhere             tcp dpt:ssh
ACCEPT     tcp  --  caleston-lp10        anywhere             tcp dpt:http
DROP       all  --  anywhere             anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
DROP       tcp  --  anywhere             anywhere             tcp dpt:http
```

- To allow https connection from the server to **`google.com`**

```
bob@devapp01:~$ sudo iptables -I OUTPUT -p tcp -d google.com --dport 443 -j
ACCEPT
```

- **Unblock IP Address** or to **delete** a rule in IPtables Firewall
- First find the **line-number** of the rule using the command below

```
bob@devapp01:~$ sudo iptables -L --line-numbers
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
1    ACCEPT     tcp  --  caleston-lp10        anywhere             tcp dpt:ssh
2    ACCEPT     tcp  --  caleston-lp10        anywhere             tcp dpt:ssh
3    DROP       all  --  anywhere             anywhere

Chain FORWARD (policy ACCEPT)
num  target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination
1    ACCEPT     tcp  --  anywhere             google.com           tcp
dpt:https
2    ACCEPT     tcp  --  anywhere             devdb01              tcp
dpt:postgresql
3    ACCEPT     tcp  --  anywhere             caleston-repo-01     tcp
dpt:http
4    DROP       tcp  --  anywhere             anywhere             tcp
dpt:http
5    DROP       tcp  --  anywhere             anywhere             tcp
dpt:https
```

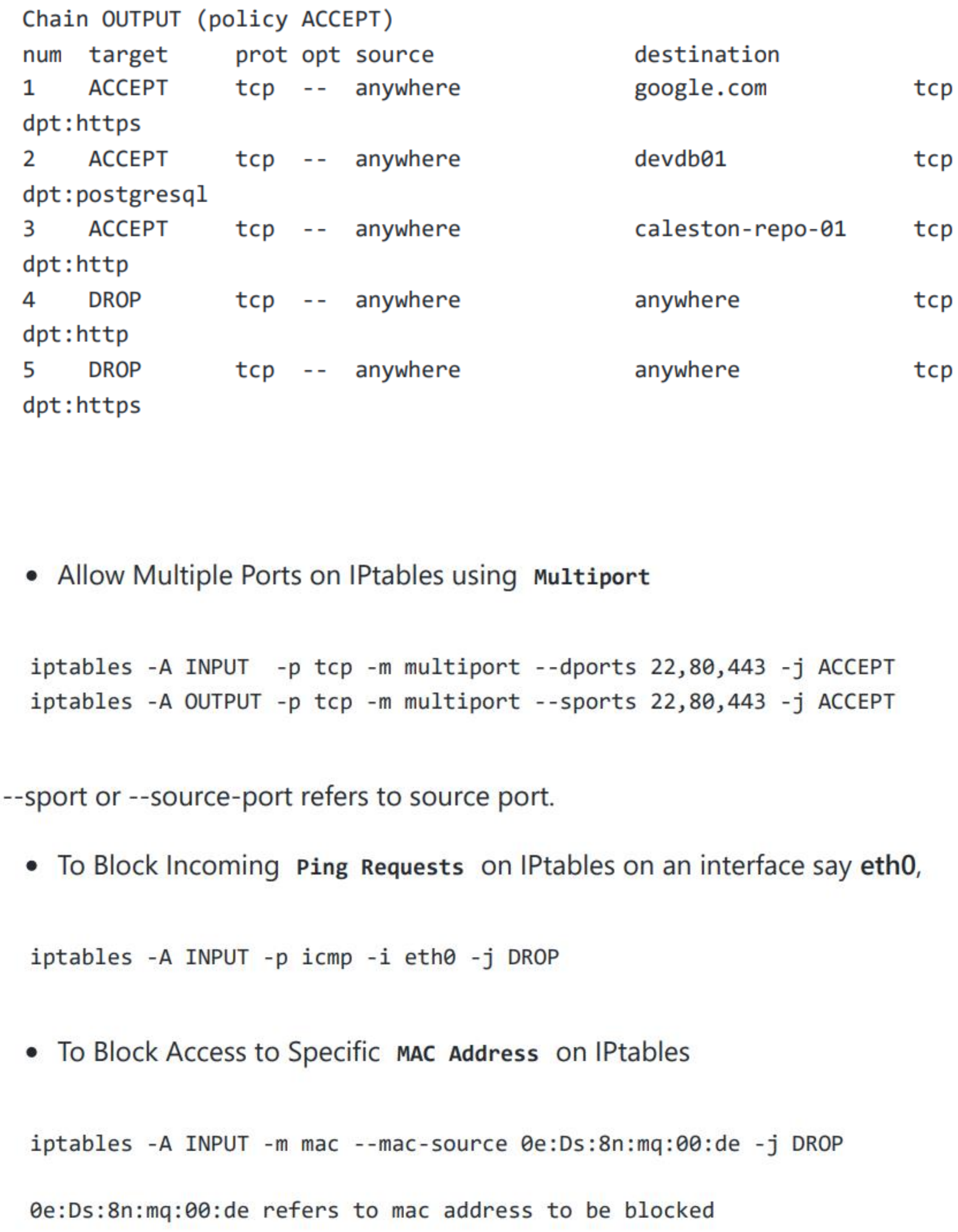
- Now if you want to delete the **INPUT** rule number 3, run

```
sudo iptables -D INPUT 3
```

- To display the **line number** for the rules,

```
bob@devapp01:~$ sudo iptables -L --line-numbers
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
1    ACCEPT     tcp  --  caleston-lp10        anywhere             tcp dpt:ssh
2    ACCEPT     tcp  --  caleston-lp10        anywhere             tcp dpt:ssh

Chain FORWARD (policy ACCEPT)
num  target     prot opt source               destination
```



```
Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination
1    ACCEPT     tcp  --  anywhere             google.com           tcp
dpt:https
2    ACCEPT     tcp  --  anywhere             devdb01              tcp
dpt:postgresql
3    ACCEPT     tcp  --  anywhere             caleston-repo-01     tcp
dpt:http
4    DROP       tcp  --  anywhere             anywhere             tcp
dpt:http
5    DROP       tcp  --  anywhere             anywhere             tcp
dpt:https
```

- Allow Multiple Ports on IPtables using **`Multiport`**

```
iptables -A INPUT  -p tcp -m multiport --dports 22,80,443 -j ACCEPT
iptables -A OUTPUT -p tcp -m multiport --sports 22,80,443 -j ACCEPT
```

--sport or --source-port refers to source port.

- To Block Incoming **`Ping Requests`** on IPtables on an interface say **eth0**,

```
iptables -A INPUT -p icmp -i eth0 -j DROP
```

- To Block Access to Specific **`MAC Address`** on IPtables

```
iptables -A INPUT -m mac --mac-source 0e:Ds:8n:mq:00:de -j DROP
```

```
0e:Ds:8n:mq:00:de refers to mac address to be blocked
```