Delete (Flush) existing rules

Let's start by clearing out all pre-existing rules in the firewall. You can use the "Flush" command to do this.

```
iptables -F
```

Set the default chain policies

Now that the firewall is empty, we can initialize the default policies. Any traffic that does not match a rule in the firewall will fallback on the default policy (in this case, we will block all traffic by default).

```
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT DROP
```

Show status of your firewall

Now we can get a quick look at the firewall so far, what the policies are set to, and any rules which might exist.

```
iptables -L -n -v --line-numbers
```

Block an IP address

We can block individual IPs using a simple command.

```
iptables -A INPUT -s 1.2.3.4 -j DROP
```

Similarly, you could have specified a subnet to block as well (.e.g., 10.0.0.0/8)

Block access to remote site

If we wanted to block access from the inside of the network from being able to reach a remote resource, we can also do so easily. Feel free to use IPs, subnets, or even domain names (they will be automatically resolved).

```
iptables -A OUTPUT -p tcp -d www.facebook.com -j DROP
iptables -A OUTPUT -p tcp -d facebook.com -j DROP
```

Allow ping from the outside

We can allow ICMP echo replies to allow for testing

```
iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT

iptables -A OUTPUT -p icmp --icmp-type echo-reply -j ACCEPT
```

Allow ping to the outside

Since our default policies are blocking, we will want to specify outbound ICMP as well.

```
iptables -A OUTPUT -p icmp --icmp-type echo-request -j ACCEPT

iptables -A INPUT -p icmp --icmp-type echo-reply -j ACCEPT
```

Allow all incoming SSH tunnels to eth0

In addition to setting the type of connection and ports to be used in a rule, you can specify which interface adapter is allowed as well. In this case, we will allow new SSH connections to be stablished from the outside, but only over eth0.

```
iptables -A INPUT -i eth0 -p tcp --dport 22 -m state --state NEW,ESTABLISHED -j ACC
EPT

iptables -A OUTPUT -o eth0 -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT
```

Allow incoming SSH tunnels only from a specific source

Traffic can further be limited to a specific source.

```
iptables -A INPUT -i eth0 -p tcp -s 192.168.0.0/24 --dport 22 -m state --state NEW,
ESTABLISHED -j ACCEPT

iptables -A OUTPUT -o eth0 -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT
```

Allow HTTP web traffic

If we wanted to run a web server, we would use the following.

```
iptables -A INPUT -p tcp --dport 80 -m state --state NEW,ESTABLISHED -j ACCEPT

iptables -A OUTPUT -p tcp --sport 80 -m state --state ESTABLISHED -j ACCEPT
```

Load balance web traffic

NetFilter can also be used for round-robin load balancing. In this example, we will route all port 80 traffic between 3 web servers on the inside.

```
iptables -A PREROUTING -p tcp --dport 80 -m state -m nth --every 3 --packet 0 -j DN
AT --to-destination 10.0.0.4
```

```
iptables -A PREROUTING -p tcp --dport 80 -m state -m nth --every 3 --packet 1 -j DN
AT --to-destination 10.0.0.5

iptables -A PREROUTING -p tcp --dport 80 -m state -m nth --every 3 --packet 2 -j DN
AT --to-destination 10.0.0.6
```

Allow outbound DNS

Since outbound traffic is blocked by default, we will want to open a few things. Let's start with DNS.

```
iptables -A OUTPUT -p udp -o eth0 --dport 53 -j ACCEPT

iptables -A INPUT -p udp -i eth0 --sport 53 -j ACCEPT
```

Allow email traffic (inbound SMTP)

If you want to serve email, you should open SMTP.

```
iptables -A INPUT -p tcp --dport 25 -m state --state NEW,ESTABLISHED -j ACCEPT

iptables -A OUTPUT -p tcp --sport 25 -m state --state ESTABLISHED -j ACCEPT
```

Allow inbound POP3

Similarly, we should allow POP3 access to this machine.

```
iptables -A INPUT -p tcp --dport 110 -m state --state NEW,ESTABLISHED -j ACCEPT

iptables -A OUTPUT -p tcp --sport 110 -m state --state ESTABLISHED -j ACCEPT
```

Port forwarding

Ports can be forwarded from the external address to a machine on the private network easily.

```
iptables -t nat -A PREROUTING -p tcp -d 1.2.3.4 --dport 422 -j DNAT --to 192.168.0.
100:22
```