



Linux Academy
Live! Lab

Route IP Traffic by Creating Static Routes

Contents

Introduction.....	1
Adding and Removing a Route.....	1
Persistent Routes.....	2

Related Courses

*Red Hat Certified
Engineer*

Related Videos

*Route IP Traffic
by Creating Static
Routes*

Need Help?

*Linux Academy
Community*

*... and you can
always send in a
support ticket on
our website to talk
to an instructor!*

Lab Connection Information

- Labs may take up to five minutes to build
- Username and password information is stored on the Live! Lab page.
- Labs will expire after a defined amount of time.

Introduction

Networks manage how traffic moves from a source server to a destination, with the initial route being dictated via the default gateway, which lets users off the origin box. Static routes can be manually configured to determine how traffic flows on your servers.

Adding and Removing a Route

Run `ip route list` to view the default network, as well as other available networks and gateways; you want to take note of both the `eth0` and `eth1` gateways for future use. These control how traffic flows on the system.

`ip route list` is a newer command replacing `netstat -rn`. You can discover your gateways through either command; although, `ip route list` may not be available on servers supported by older Red Hat versions.

We have control over these routes, and can add static routes to force traffic to traverse in an intended path, based on IP address. You need to be careful with static routing because the packets can become stuck if the path does not allow the traffic to access the destination.

Run `ping google.com` and copy the first three octets of the output IP address — `216.58.217` in this instance; Google has many IP addresses, and yours may differ. Whether you want to use our example or work with your own is up to you. We are going to force this into our routing path in such a way it does not function correctly.

To add the static route:

```
[root@red-hat]# ip route add 216.58.217.0/24 via 10.208.192.1 dev eth1
```

The route added is the subnet for Google's network (note the `0` in place of the fourth octet), and we are routing *via* our internal default gateway, `10.208.191.1`, which lives on `eth1`. Your default gateway may vary, refer to `ip route list` or `netstat -rn` to confirm your own. Should you run `ip route list` again, after running the above command, you can see the added route.

Now pull the full Google IP address (ours is `216.58.217.174`), and try to `ping` it. Data will not transfer. The `ping` fails.

We now need to remove the faulty route:

```
[root@red-hat]# ip route del 216.58.217.0/24 via 10.208.191.1 dev eth1
```

Notice that the command almost fully mimics the command we used to create a static route, but with *del* instead of *add* to signify we are deleting the route. Again, you need to replace the gateway information with

your own.

If you attempt to `ping` the IP again, it succeeds. You can also do a `traceroute` to see how the packets move to its destination, or run `ip route list` to see the available routes.

Persistent Routes

If a route needs to be persistent, then we need to create a file for static routes, located at `/etc/sysconfig/static-routes`. This is determined by outputting `/etc/init.d/network` and grepping for `route`.

In the new `static-routes` file, we use a Google IP address to route through our main gateway, `eth0`; as above, you need to retrieve your `eth0` gateway IP separately.

Add the below command, replacing the Google IP and default gateway as needed.

```
any net 216.58.217.0 netmask 255.255.255.0 gw 162.242.253.1 dev eth0
```

Essentially, this says “for any network that exists on the IP subnet with the netmask of 255, use the gateway of our default network (IP) on device `eth0`.”

Should we run `ip route list`, this new network is not yet added. We need to restart the network service for it to take effect:

```
[root@red-hat]# systemctl restart network
```

Rerun `ip route list` to see the newly-added persistent route. If we now `ping 216.58.217.174`, ping can send packets. Run `traceroute 216.58.217.174` to see the path used.