# Lab 3

## Static Routing, OSPF & Routing Policy

## Overview

This lab demonstrates configuration and monitoring of Layer 3 routing on JUNOS platforms. In this lab, you use the CLI to configure and monitor interfaces, static routing, and basic OSPF. Throughout these configuration tasks, you will become familiar with and describe the contents of the routing and forwarding tables.

The second section of this lab demonstrates configuration and monitoring of routing policy on JUNOS devices. You will use the command-line interface (CLI) to define, apply, and monitor basic routing policy.

All devices are connected to a common management network which facilitates access to the CLI. These exercises assume you already have some basic understanding of the JUNOS CLI interfaces or you have read the IJS documentation or similar.

Note that your `lab` login (password given to you separately) grants you all permissions needed to complete this lab; however, some restrictions have been made to prevent loss of connectivity to the devices. Please be careful, and have fun!

By completing this lab, you will perform the following tasks:

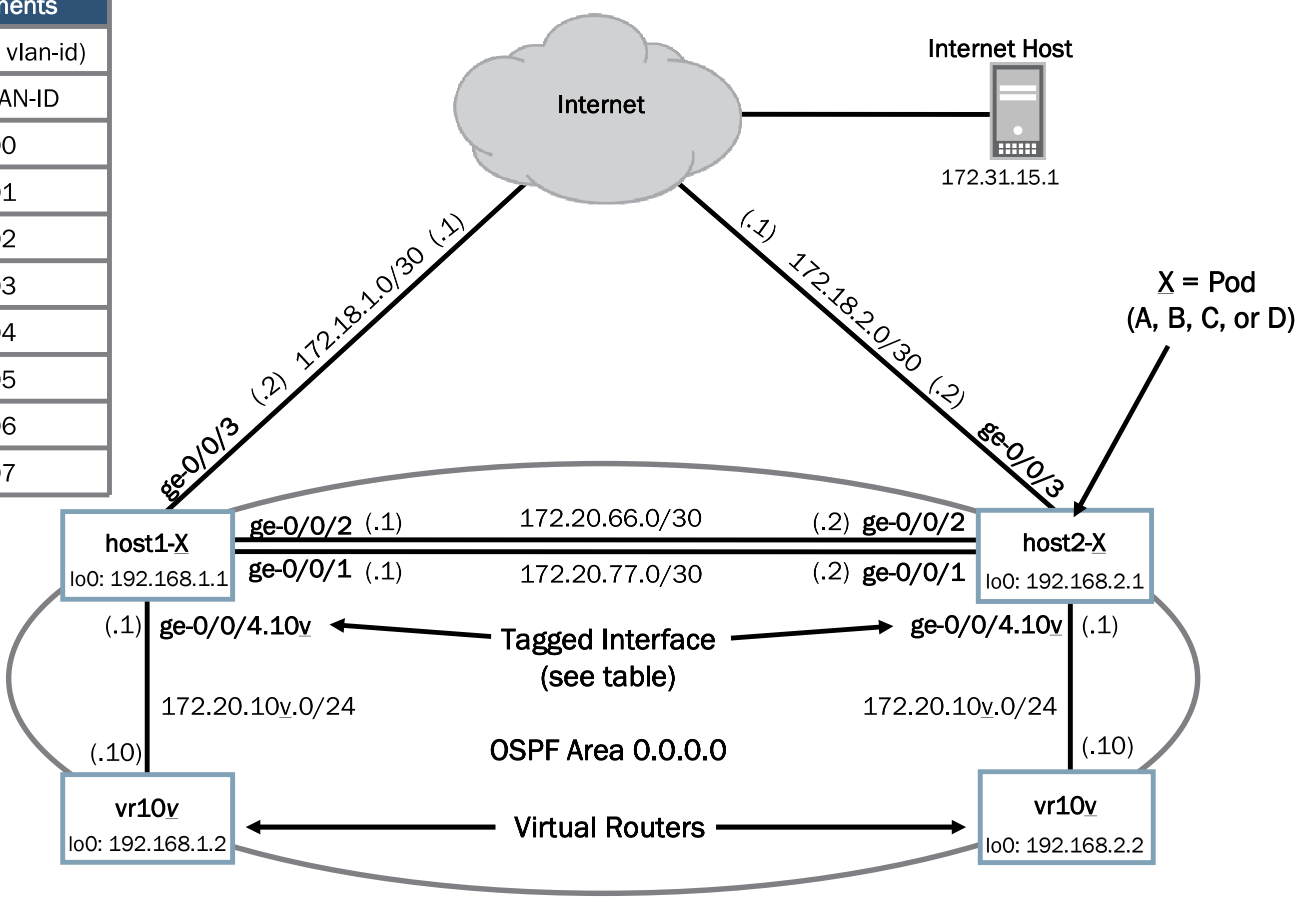
- Configure and verify proper operation of network interfaces.
- Configure and monitor static routing.
- Configure and monitor OSPF.
- Configure and monitor routing policy.

Please refer to the lab 3 diagram to perform sections 1, 2 & 3 of this exercise:

## Lab Diagram

# Network Diagram: Statics & OSPF

| VLAN Assignments | |
|---|---|
| (v=remainder of vlan-id) | |
| Hostname | VLAN-ID |
| host1-a | 100 |
| host2-a | 101 |
| host1-b | 102 |
| host2-b | 103 |
| host1-c | 104 |
| host2-c | 105 |
| host1-d | 106 |
| host2-d | 107 |



# Key Commands

Key *operational* mode commands used in this lab include the following:

```
configure
show configuration
show interfaces terse
show route
show ospf neighbor
```

## Part 1: Configuring and Monitoring Interfaces

In this lab part, you will configure network interfaces on your assigned device. You will then verify that the interfaces are operational and display to corresponding route table entries.

To do this lab you must configure the two devices that have been assigned to you (host1-x and host2-x). You need to configure both systems

**Note**

Please do NOT delete interface ge-0/0/0 as this is your management interface which provides access to your session and the J-Web !!

Do NOT delete either the security section of your configurations. This allows your system to allow any traffic in/out.

---

**Note 2**

Preferably use the console connection to access your assigned station. Using the console connection ensures persistent connectivity even when the management network access is unavailable. This lab could be done also via management intf

---

**Note 3**

Remember that the exercise proposed in this documentation is generic and the examples given here apply only to one particular pod of devices. Please adapt the example to your assigned set of devices (host1-a & host2-a, or host1-b & host2-b, or host1-c & host2-c, or host1-d & host2-d).
Look at you lab diagram and mind the pod of systems that you have been assigned!

---

### Step 1.1

Log in to the system using the console port with the username `lab` using the password given to you. Please use the console connection to access your system.

```
[luis@js2 ~]$ telnet 192.168.2.11 7001
Trying 192.168.2.11...
Connected to 192.168.2.11 (192.168.2.11).
Escape character is '^]'.
```

```
host1-a (ttyp2)

login: lab
Password:


--- JUNOS 9.6R1.13 built 2009-08-01 09:23:09 UTC
lab@host1-a>
```

## Step 1.2

Issue the **show route** command to display the contents of the route table.

```
lab@host1-a> show route

inet.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.210.14.128/27   *[Direct/0] 2d 20:02:51
                    > via ge-0/0/0.0
10.210.14.131/32   *[Local/0] 2d 20:02:51
                      Local via ge-0/0/0.0
```

- What route table is displayed with the **show route** command?

________________________________________

❖ The output should show the `inet.0` route table, which is the primary IPv4 route table for the master routing instance. You can display all route tables and their respective entries using the **show route all** command, as shown in the following output:

```
lab@host1-a> show route all

inet.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.210.14.128/27   *[Direct/0] 2d 20:04:06
                    > via ge-0/0/0.0
10.210.14.131/32   *[Local/0] 2d 20:04:06
                      Local via ge-0/0/0.0

__juniper_private1__.inet.0: 4 destinations, 4 routes (2 active, 0 holddown, 2
hidden)
+ = Active Route, - = Last Active, * = Both

10.0.0.1/32        *[Direct/0] 3w4d 19:57:51
                    > via lo0.16385
10.0.0.16/32       *[Direct/0] 3w4d 19:57:51
                    > via lo0.16385
128.0.0.1/32        [Direct/0] 3w4d 19:57:51
                    > via lo0.16385
128.0.1.16/32       [Direct/0] 3w4d 19:57:51
                    > via lo0.16385

__juniper_private2__.inet.0: 1 destinations, 1 routes (0 active, 0 holddown, 1
```

```
hidden)
+ = Active Route, - = Last Active, * = Both

127.0.0.1/32        [Direct/0] 3w4d 19:57:51
                    > via lo0.16384


__juniper_private1__.inet6.0: 1 destinations, 1 routes (1 active, 0 holddown, 0
hidden)
+ = Active Route, - = Last Active, * = Both

fe80::226:88ff:fee9:d280/128
                   *[Direct/0] 3w4d 19:57:51
                    > via lo0.16385
```

- What route entries are present in the `inet.0` route table?

---

❖ The `inet.0` route table should currently show a single `Direct` route and a single `Local` route. Both routes are associated with the ge-0/0/0 interface. The `Direct` route matches the IP address assigned to the ge-0/0/0 interface while the `Local` route matches the management network.

## Step 1.3

Refer to the network diagram for this lab and configure the listed interfaces. Use logical unit 0 on the interfaces `ge-0/0/1`, `ge-0/0/2` and `ge-0/0/3`, using a /30 as a subnet mask.

Please observe that `ge-0/0/4` is a tagged interface! Use the VLAN-ID as the logical unit value for the tagged interface. Wherever this lab example indicates *10v* you will have to replace it for your system vlan-id. Use a mask of /24 for this interface. Please refer to the following table:

| **VLAN Assignments** | |
|---|---|
| **(v=remainder of vlan-id)** | |
| **Hostname** | **VLAN-ID** |
| host1-a | 100 |
| host2-a | 101 |
| host1-b | 102 |
| host2-b | 103 |
| host1-c | 104 |
| host2-c | 105 |
| host1-d | 106 |
| host2-d | 107 |

At last, do not forget to configure the loopback interface with a logical unit 0 and subnet mask /30

```
[edit]
lab@host1-a# edit interfaces

[edit interfaces]
lab@host1-a# set ge-0/0/1 unit 0 family inet address 172.20.77.x/30

[edit interfaces]
lab@host1-a# set ge-0/0/2 unit 0 family inet address 172.20.66.x/30

[edit interfaces]
lab@host1-a# set ge-0/0/3 unit 0 family inet address 172.18.x.2/30

[edit interfaces]
lab@host1-a# set lo0 unit 0 family inet address 192.168.x.1/32

[edit interfaces]
lab@host1-a# set ge-0/0/4 vlan-tagging

[edit interfaces]
lab@host1-a# set ge-0/0/4 unit 10v vlan-id 10v

[edit interfaces]
lab@host1-a# set ge-0/0/4 unit 10v family inet address 172.20.10v.1/24
```

Your configuration should look like this example taken from host1-a. The following sample configuration is taken from `host1-a` in an environment using SRX Series devices. If your environment is using J Series devices, your output will be different. Please refer to the lab diagram given to you

```
[edit interfaces]
lab@host1-a# show

ge-0/0/0 {
    description "MGMT Interface - DO NOT DELETE";
    unit 0 {
        family inet {
            address 10.210.14.131/27;
        }
    }
}
ge-0/0/1 {
    unit 0 {
        family inet {
            address 172.20.77.1/30;
        }
    }
}
ge-0/0/2 {
    unit 0 {
        family inet {
            address 172.20.66.1/30;
        }
    }
}
```

```
ge-0/0/3 {
    unit 0 {
        family inet {
            address 172.18.1.2/30;
        }
    }
}
ge-0/0/4 {
    vlan-tagging;
    unit 100 {
        vlan-id 100;
        family inet {
            address 172.20.100.1/24;
        }
    }
}
lo0 {
    unit 0 {
        family inet {
            address 192.168.1.1/32;
        }
    }
}
```

### Step 1.4

Activate the configuration and return to operational mode.

```
[edit interfaces]
lab@host1-a# commit and-quit

commit complete
Exiting configuration mode

lab@host1-a>
```

**Note**

Please log into the other device assigned to you (`host2-X`) and configure the relevant interfaces too!

### Step 1.5

Issue the **`show interfaces terse`** CLI command to verify the state of the configured interfaces.

```
lab@host1-a> show interfaces terse

Interface               Admin Link Proto    Local                 Remote

ge-0/0/0                up    up
ge-0/0/0.0              up    up   inet     10.210.14.131/27
...TRIMMED...
```

```
ge-0/0/1                up    up
ge-0/0/1.0              up    up   inet     172.20.77.1/30
ge-0/0/2                up    up
ge-0/0/2.0              up    up   inet     172.20.66.1/30
ge-0/0/3                up    up
ge-0/0/3.0              up    up   inet     172.18.1.2/30
ge-0/0/4                up    up
ge-0/0/4.100            up    up   inet     172.20.100.1/24
ge-0/0/4.32767          up    up
ge-0/0/5                up    down
ge-0/0/6                up    down
ge-0/0/7                up    down
ge-0/0/8                up    down
ge-0/0/9                up    down
ge-0/0/10               up    down
ge-0/0/11               up    down
ge-0/0/12               up    down
ge-0/0/13               up    down
ge-0/0/14               up    down
ge-0/0/15               up    down
gre                     up    up
ipip                    up    up
lo0                     up    up
lo0.0                   up    up   inet     192.168.1.1         --> 0/0
...TRIMMED ...
```

- What is the `Admin` and `Link` state of the recently configured interfaces?

_______________________________________________

_______________________________________________

  - ❖ All configured interfaces should show an `Admin` and `Link` state of up, as shown in the sample capture.

## Step 1.6

Once you have configured the other device (i.e. *host2-a)* use the ping utility to verify reachability to the neighboring IPs connected to your device.. The following sample capture shows ping tests from *host1-a* to the Internet gateway, *host2-a*, and *vr100*, which are all directly connected:

- Were the pings to all directly connected neighbors successful?

_______________________________________________

_______________________________________________

  - ❖ The output sample shows that you can successfully ping all directly connected neighbor interfaces

```
lab@host1-a> ping 172.20.77.2 count 3

PING 172.20.77.2 (172.20.77.2): 56 data bytes
64 bytes from 172.20.77.2: icmp_seq=0 ttl=64 time=1.261 ms
64 bytes from 172.20.77.2: icmp_seq=1 ttl=64 time=1.153 ms
64 bytes from 172.20.77.2: icmp_seq=2 ttl=64 time=1.153 ms

--- 172.20.77.2 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.153/1.189/1.261/0.051 ms


lab@host1-a> ping 172.20.66.2 count 3

PING 172.20.66.2 (172.20.66.2): 56 data bytes
64 bytes from 172.20.66.2: icmp_seq=0 ttl=64 time=1.235 ms
64 bytes from 172.20.66.2: icmp_seq=1 ttl=64 time=1.135 ms
64 bytes from 172.20.66.2: icmp_seq=2 ttl=64 time=1.180 ms

--- 172.20.66.2 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.135/1.183/1.235/0.041 ms


lab@host1-a> ping 172.18.1.1 count 3

PING 172.18.1.1 (172.18.1.1): 56 data bytes
64 bytes from 172.18.1.1: icmp_seq=0 ttl=64 time=2.323 ms
64 bytes from 172.18.1.1: icmp_seq=1 ttl=64 time=3.618 ms
64 bytes from 172.18.1.1: icmp_seq=2 ttl=64 time=4.274 ms
--- 172.18.1.1 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 2.323/3.405/4.274/0.811 ms


lab@host1-a> ping 172.20.100.10 count 3

PING 172.20.100.10 (172.20.100.10): 56 data bytes
64 bytes from 172.20.100.10: icmp_seq=0 ttl=64 time=4.017 ms
64 bytes from 172.20.100.10: icmp_seq=1 ttl=64 time=8.165 ms
64 bytes from 172.20.100.10: icmp_seq=2 ttl=64 time=2.146 ms
--- 172.20.100.10 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 2.146/4.776/8.165/2.515 ms

lab@host1-a>
```

## Step 1.7

Issue a show route command to confirm that your configured interfaces appear as Direct or Local in the routing table inet.0

```
lab@host1-a> show route

inet.0: 9 destinations, 9 routes (9 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
10.210.14.128/27   *[Direct/0] 05:44:13
                    > via ge-0/0/0.0
10.210.14.131/32   *[Local/0] 05:44:13
                      Local via ge-0/0/0.0
172.18.1.0/30      *[Direct/0] 00:31:17
                    > via ge-0/0/3.0
172.18.1.2/32      *[Local/0] 00:31:17
                      Local via ge-0/0/3.0
172.20.66.0/30     *[Direct/0] 00:31:17
                    > via ge-0/0/2.0
172.20.66.1/32     *[Local/0] 00:31:17
                      Local via ge-0/0/2.0
172.20.77.0/30     *[Direct/0] 00:31:17
                    > via ge-0/0/1.0
172.20.77.1/32     *[Local/0] 00:31:17
                      Local via ge-0/0/1.0
172.20.100.0/24    *[Direct/0] 00:09:10
                    > via ge-0/0/4.100
172.20.100.1/32    *[Local/0] 00:09:10
                      Local via ge-0/0/4.100
192.168.1.1/32     *[Direct/0] 00:31:17
                    > via lo0.0
```

- Does the route table display an entry for all local interface addresses and directly connected networks?

    - The answer should be yes. If needed, you can refer back to the network diagram and compare it with the displayed route entries.

- What is the route preference for the `Local` and `Direct` route entries?

    - The `Local` and `Direct` route entries should both show a route preference of `0`, as shown in the sample output.

- Are any routes currently hidden?

    - No routes should be hidden at this time. The summary line towards the top of the sample output makes this lack of hidden routes evident.

# Part 2: Configuring and Monitoring Static Routing

In this lab part, you will configure and monitor static routing

## Step 2.1

Attempt to ping the Internet host referenced on the network diagram for this lab.

**Note**

Use Ctrl+c to stop a continuous ping operation.

```
lab@host1-a> ping 172.31.15.1
PING 172.31.15.1 (172.31.15.1): 56 data bytes
ping: sendto: No route to host
ping: sendto: No route to host
ping: sendto: No route to host
ping: sendto: No route to host
^C
--- 172.31.15.1 ping statistics ---
   4 packets transmitted, 0 packets received, 100% packet loss
```

♦ What does the result from the ping operation indicate?

❖ The results from the ping operation indicate that no route to the specified host currently exists.

♦ Based on the network diagram, what IP address would your device use as a next hop to reach the Internet host?

❖ The answer depends on your assigned device. For all `host1-x` devices, the next-hop IP address would be 172.18.1.1. For all `host2-x` devices, the next-hop IP address would be 172.18.2.1.

## Step 2.2

Enter configuration mode and define a default static route. Use the IP address identified in the last step as the next hop for the default static route.

```
lab@host1-a> configure
Entering configuration mode
```

```
[edit]
lab@host1-a# edit routing-options

[edit routing-options]
lab@host1-a# set static route 0.0.0.0/0 next-hop 172.18.x.1

[edit routing-options]
lab@host1-a#
```

## Step 2.3

Activate the newly added static route and issue the **run show route 172.31.15.1** command.

```
[edit routing-options]
lab@host1-a# commit

commit complete

[edit routing-options]
lab@host1-a# run show route 172.31.15.1

inet.0: 12 destinations, 12 routes (12 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0          *[Static/5] 00:00:14
                    > to 172.18.1.1 via ge-0/0/3.0
```

- Does the IP address associated with the Internet host now show a valid route entry?

  ___

  ___

  - Yes, at this point the default static route should be active and all destinations that do not have a more specific route entry, would use the default route.

- What is the route preference of the default static route?

  ___

  ___

  - The default static route uses the route preference value of 5, which is the default route preference for static routes

## Step 2.4

Issue the **run ping 172.31.15.1** command to ping the Internet host.

```
[edit routing-options]
lab@host1-a# run ping 172.31.15.1
```

```
PING 172.31.15.1 (172.31.15.1): 56 data bytes
64 bytes from 172.31.15.1: icmp_seq=0 ttl=64 time=2.892 ms
64 bytes from 172.31.15.1: icmp_seq=1 ttl=64 time=2.425 ms
64 bytes from 172.31.15.1: icmp_seq=2 ttl=64 time=2.595 ms
^C
--- 172.31.15.1 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 2.425/2.637/2.892/0.193 ms
```

- Does the ping operation succeed this time?

___

___

- Yes, the ping operation should now succeed.

## Step 2.5

**Note**

Refer to the network diagram, as needed, for the subsequent lab steps.

Add a static route to the loopback address of the directly attached virtual router.

```
[edit routing-options]
lab@host1-a# set static route 192.168.x.2/32 next-hop 172.20.10v.10
```

## Step 2.6

Define the required static routes to allow end-to-end connectivity to the remote subnet and loopback addresses within your assigned pod. Use the IP address assigned to the remote device on the 172.20.66.0/30 subnet as the next hop for these static routes.

```
[edit routing-options]
lab@host1-a# set static route 192.168.x.1/32 next-hop 172.20.66.x

[edit routing-options]
lab@host1-a# set static route 192.168.x.2/32 next-hop 172.20.66.x

[edit routing-options]
lab@host1-a# set static route 172.20.10v.0/24 next-hop 172.20.66.x
```

So far, your configuration should look similar to this output taken from `host1-a`:

```
[edit routing-options]
lab@host1-a# show
static {
    route 0.0.0.0/0 next-hop 172.18.1.1;
    route 192.168.1.2/32 next-hop 172.20.100.10;
    route 192.168.2.1/32 next-hop 172.20.66.2;
```

```
    route 192.168.2.2/32 next-hop 172.20.66.2;
    route 172.20.101.0/24 next-hop 172.20.66.2;
}
```

## Step 2.7

Use the IP address assigned to the remote student device on the 172.20.77.0/30 subnet as a qualified next hop for the recently added static routes to the remote subnet and loopback addresses. Use a route preference of 6 for these definitions.

```
[edit routing-options]
lab@host1-a# set static route 192.168.x.1/32 qualified-next-hop 172.20.77.x
preference 6

[edit routing-options]
lab@host1-a# set static route 192.168.x.2/32 qualified-next-hop 172.20.77.x
preference 6

[edit routing-options]
lab@host1-a# set static route 172.20.10v.0/24 qualified-next-hop 172.20.77.x
preference 6
```

## Step 2.8

Display the resulting configuration to review your work. Once satisfied, activate the configuration changes and return to operational mode. This example is taken from `host1-a`

```
[edit routing-options]
lab@host1-a# show
static {
    route 0.0.0.0/0 next-hop 172.18.1.1;
    route 192.168.1.2/32 next-hop 172.20.100.10;
    route 192.168.2.1/32 {
        next-hop 172.20.66.2;
        qualified-next-hop 172.20.77.2 {
            preference 6;
        }
    }
    route 192.168.2.2/32 {
        next-hop 172.20.66.2;
        qualified-next-hop 172.20.77.2 {
            preference 6;
        }
    }
    route 172.20.101.0/24 {
        next-hop 172.20.66.2;
        qualified-next-hop 172.20.77.2 {
            preference 6;
        }
    }
```

```
}

[edit routing-options]
lab@host1-a# commit and-quit
commit complete
Exiting configuration mode

lab@host1-a>
```

## Step 2.9

Issue the **show route protocol static** command to view the current static routes in your device route table.

```
lab@host1-a> show route protocol static

inet.0: 16 destinations, 19 routes (16 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0          *[Static/5] 00:28:57
                    > to 172.18.1.1 via ge-0/0/3.0
172.20.101.0/24    *[Static/5] 00:02:16
                    > to 172.20.66.2 via ge-0/0/2.0
                    [Static/6] 00:02:16
                    > to 172.20.77.2 via ge-0/0/1.0
192.168.1.2/32     *[Static/5] 00:02:16
                    > to 172.20.100.10 via ge-0/0/4.100
192.168.2.1/32     *[Static/5] 00:02:16
                    > to 172.20.66.2 via ge-0/0/2.0
                    [Static/6] 00:02:16
                    > to 172.20.77.2 via ge-0/0/1.0
192.168.2.2/32     *[Static/5] 00:02:16
                    > to 172.20.66.2 via ge-0/0/2.0
                    [Static/6] 00:02:16
                    > to 172.20.77.2 via ge-0/0/1.0
```

- How many static routes display?

______________________________________________

______________________________________________

  - Each device should show five static routes.

- Are both next hops displayed for the remote subnet and loopback destinations? Which next hop is active? Why?

______________________________________________

______________________________________________

❖ You should see both next hops associated with the remote subnet and loopback destinations. The routes using the next hop associated with the 10.210.66.0/30 subnet should be active due to a lower route preference of 5.

## Step 2.10

Ping the loopback address of all internal devices to verify reachability.

**Note**

The virtual routers have a preconfigured default static route using their directly connected devices as the next hop.

```
lab@host1-a> ping 192.168.1.2 rapid count 25
PING 192.168.1.2 (192.168.1.2): 56 data bytes
!!!!!!!!!!!!!!!!!!!!!!!!!
--- 192.168.1.2 ping statistics ---
25 packets transmitted, 25 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.885/5.442/64.180/12.088 ms

lab@host1-a> ping 192.168.2.2 rapid count 25
PING 192.168.2.2 (192.168.2.2): 56 data bytes
!!!!!!!!!!!!!!!!!!!!!!!!!
--- 192.168.2.2 ping statistics ---
25 packets transmitted, 25 packets received, 0% packet loss
round-trip min/avg/max/stddev = 2.208/2.959/4.508/0.808 ms

lab@host1-a> ping 192.168.2.1 rapid count 25
PING 192.168.2.1 (192.168.2.1): 56 data bytes
!!!!!!!!!!!!!!!!!!!!!!!!!
--- 192.168.2.1 ping statistics ---
25 packets transmitted, 25 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.970/2.469/7.018/2.469 ms
```

♦ Did the ping tests succeed?

________________________________________

________________________________________

❖ The ping tests should succeed as long as the remote device has the required configuration in place. If the tests fail, check the remote station to ensure that the required configuration is in place.

## Part 3: Configuring and Monitoring OSPF

In this part, you will configure and monitor OSPF. You will configure a single OSPF area based on the network diagram for this lab. Finally, you will perform some verification tasks to ensure that OSPF works properly.

### Step 3.1

Enter configuration mode and navigate to the `[edit protocols ospf]` hierarchy level.

```
lab@host1-a> configure
Entering configuration mode

[edit]
lab@host1-a# edit protocols ospf

[edit protocols ospf]
lab@host1-a#
```

### Step 3.2

Define OSPF Area 0 and include all internal interfaces that connect to the remote device and the directly connected virtual router. Ensure that you also include the lo0 interface. Issue the **`show`** command to view the resulting configuration.

**Note**

Remember to specify the appropriate logical interface! If the logical unit is not specified, JUNOS Software assumes a logical unit of zero (0).

```
[edit protocols ospf]
lab@host1-a# set area 0.0.0.0 interface ge-0/0/1.0

[edit protocols ospf]
lab@host1-a# set area 0.0.0.0 interface ge-0/0/2.0

[edit protocols ospf]
lab@host1-a# set area 0.0.0.0 interface ge-0/0/4.10v

[edit protocols ospf]
lab@host1-a# set area 0.0.0.0 interface lo0.0
```

Your configuration should look similar to this example taken from `host1-a`:

```
[edit protocols ospf]
lab@host1-a# show
area 0.0.0.0 {
    interface ge-0/0/1.0;
    interface ge-0/0/2.0;
    interface ge-0/0/4.100;
    interface lo0.0;
}
```

♦ With the OSPF configuration in place, how many OSPF neighbor adjacencies should form?

---

---

❖ Although four interfaces are present in the configuration, only three of those interfaces are capable of forming OSPF neighbor adjacencies.

Before proceeding to the next section ensure you do a similar configuration in the other system assigned to you (host2-a in this case).

## Step 3.3

Activate the candidate configuration using the **commit** command. Issue the **run show ospf neighbor** command to verify OSPF neighbor adjacency state information.

**Note**

The OSPF adjacency state for each neighbor is dependent on that neighbor's configuration. Ensure that the neighboring system contains the required OSPF configuration and committed the changes. The virtual routers contain preconfigured settings.

```
[edit protocols ospf]
lab@host1-a# commit
commit complete

[edit protocols ospf]
lab@host1-a# run show ospf neighbor

Address          Interface              State     ID               Pri  Dead
172.20.77.2      ge-0/0/1.0             Full      192.168.2.1      128    38
172.20.66.2      ge-0/0/2.0             Full      192.168.2.1      128    33
172.20.100.10    ge-0/0/4.100           Full      192.168.1.2      128    33
```

♦ What state do the OSPF neighbor adjacencies show?

---

---

❖ Although you might see some transitional states, the state for all three OSPF neighbors should eventually show `Full`.

## Step 3.4

Issue the **run show route protocol ospf** to view the active OSPF routes in your device's route table.

```
[edit protocols ospf]
lab@host1-a# run show route protocol ospf

inet.0: 17 destinations, 24 routes (17 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

172.20.101.0/24     [OSPF/10] 00:18:55, metric 2
                      to 172.20.77.2 via ge-0/0/1.0
                    > to 172.20.66.2 via ge-0/0/2.0
192.168.1.2/32      [OSPF/10] 00:19:00, metric 1
                    > to 172.20.100.10 via ge-0/0/4.100
192.168.2.1/32      [OSPF/10] 00:18:55, metric 1
                      to 172.20.77.2 via ge-0/0/1.0
                    > to 172.20.66.2 via ge-0/0/2.0
192.168.2.2/32      [OSPF/10] 00:18:55, metric 2
                      to 172.20.77.2 via ge-0/0/1.0
                    > to 172.20.66.2 via ge-0/0/2.0
224.0.0.5/32       *[OSPF/10] 00:19:10, metric 1
                      MultiRecv
```

♦ Are all of the OSPF routes for the remote subnet and loopback destinations active? Why?

---

---

❖ No, all of the OSPF routes for the remote subnet and loopback destinations should not be active (Note the * is missing on most of the OSPF routes). As you might remember, we still have the previously defined static routes in place. The active static routes use a route preference of 5, which makes them more preferred than OSPF routes. Internal OSPF routes use a route preference of 10, by default.

## Step 3.5

Delete all static routes used for internal connectivity. Ensure that you do <u>not</u> delete the default static route used to route traffic to the Internet.

```
[edit protocols ospf]
lab@host1-a# top edit routing-options

[edit routing-options]
lab@host1-a# show
static {
    route 0.0.0.0/0 next-hop 172.18.1.1;
    route 192.168.1.2/32 next-hop 172.20.100.10;
    route 192.168.2.1/32 {
        next-hop 172.20.66.2;
```

```
            qualified-next-hop 172.20.77.2 {
                preference 6;
            }
        }
        route 192.168.2.2/32 {
            next-hop 172.20.66.2;
            qualified-next-hop 172.20.77.2 {
                preference 6;
            }
        }
        route 172.20.101.0/24 {
            next-hop 172.20.66.2;
            qualified-next-hop 172.20.77.2 {
                preference 6;
            }
        }
    }

[edit routing-options]
lab@host1-a# delete static route 192.168.1.2/32

[edit routing-options]
lab@host1-a# delete static route 192.168.2.1/32

[edit routing-options]
lab@host1-a# delete static route 192.168.2.2/32

[edit routing-options]
lab@host1-a# delete static route 172.20.10v.0/24

[edit routing-options]
lab@host1-a# show
static {
    route 0.0.0.0/0 next-hop 172.18.1.1;
}
```

## Step 3.6

Activate the configuration and return to operational mode. Issue the **show route protocol ospf** command to verify that the OSPF routes are now active.

```
[edit routing-options]
lab@host1-a# commit and-quit
commit complete
Exiting configuration mode

lab@host1-a> show route protocol ospf

inet.0: 17 destinations, 17 routes (17 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

172.20.101.0/24    *[OSPF/10] 00:26:58, metric 2
                      to 172.20.77.2 via ge-0/0/1.0
                    > to 172.20.66.2 via ge-0/0/2.0
```

```
192.168.1.2/32      *[OSPF/10] 00:27:03, metric 1
                     > to 172.20.100.10 via ge-0/0/4.100
192.168.2.1/32      *[OSPF/10] 00:26:58, metric 1
                       to 172.20.77.2 via ge-0/0/1.0
                     > to 172.20.66.2 via ge-0/0/2.0
192.168.2.2/32      *[OSPF/10] 00:26:58, metric 2
                       to 172.20.77.2 via ge-0/0/1.0
                     > to 172.20.66.2 via ge-0/0/2.0
224.0.0.5/32        *[OSPF/10] 00:27:13, metric 1
                       MultiRecv

lab@host1-a>
```

♦ Are all of the OSPF routes for the remote subnet and loopback destinations active now?

___

___

❖ Yes, as illustrated in the sample output, all OSPF routes should now be active. (Note the `*` is now present for all of the OSPF routes.)

## Step 3.7

Ping the loopback address of all internal devices to verify reachability through the OSPF routes.

```
lab@host1-a> ping 192.168.1.2 rapid count 15
PING 192.168.1.2 (192.168.1.2): 56 data bytes
!!!!!!!!!!!!!!!
--- 192.168.1.2 ping statistics ---
15 packets transmitted, 15 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.938/9.447/94.242/22.726 ms

lab@host1-a> ping 192.168.2.1 rapid count 15
PING 192.168.2.1 (192.168.2.1): 56 data bytes
!!!!!!!!!!!!!!!
--- 192.168.2.1 ping statistics ---
15 packets transmitted, 15 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.957/2.988/7.086/2.715 ms

lab@host1-a> ping 192.168.2.2 rapid count 15
PING 192.168.2.2 (192.168.2.2): 56 data bytes
!!!!!!!!!!!!!!!
--- 192.168.2.2 ping statistics ---
15 packets transmitted, 15 packets received, 0% packet loss
round-trip min/avg/max/stddev = 2.231/4.845/12.030/2.926 ms
```

- Do the ping tests succeed?

---

- Yes, as illustrated in the sample capture, the ping tests succeed compliments of the current OSPF routes in your device's route table.

STOP

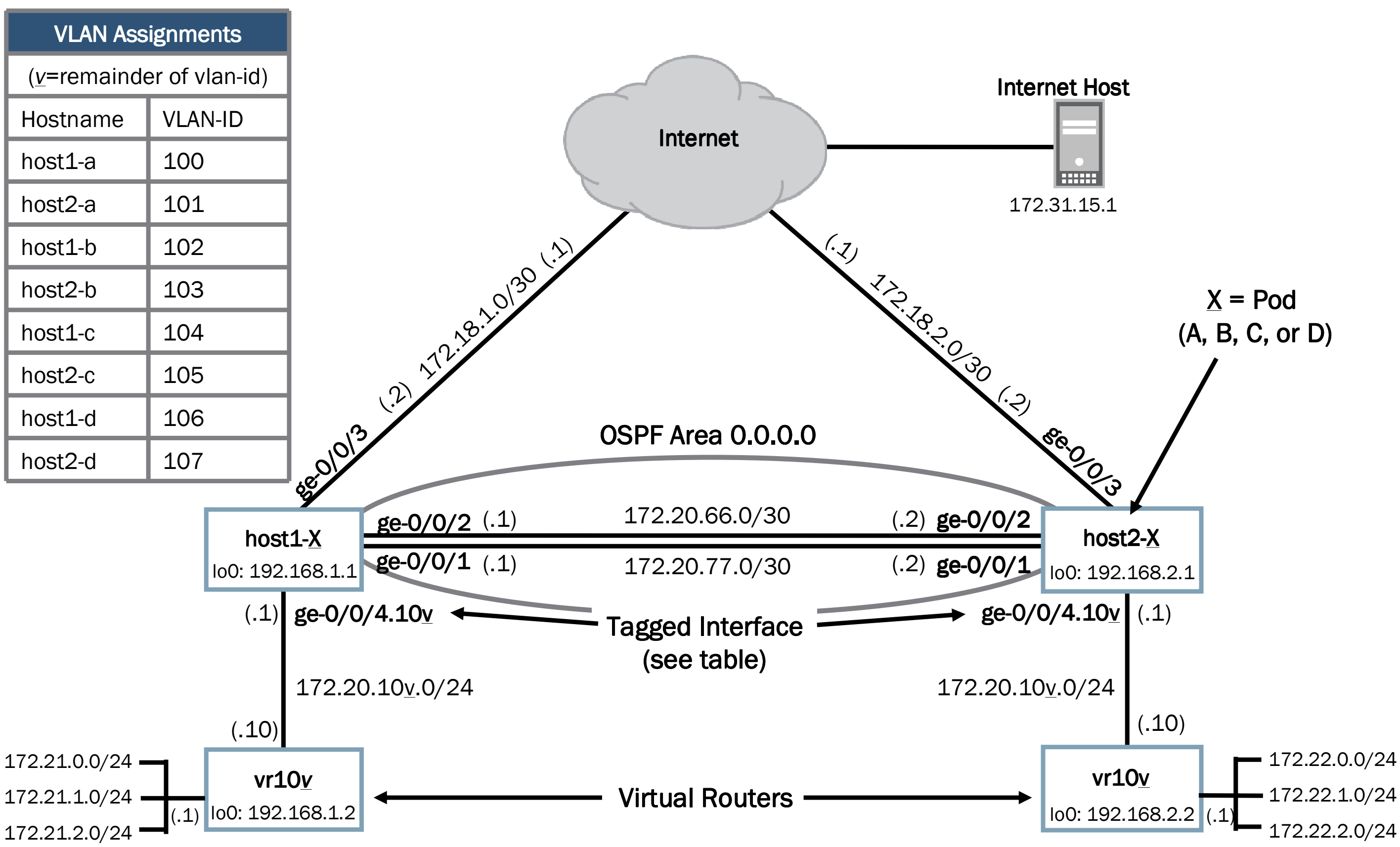
**Make sure you are done in both devices with your OSPF configurations. Now we will proceed to alter the current topolgy**

## Part 4: Re-configuring and Monitoring OSPF

In this part, you will configure a single OSPF area based on the network diagram for this lab. Finally, you will perform some verification tasks to ensure that OSPF works properly.
Please refer to the lab diagram 3b to perform this exercise:

# Network Diagram: Statics, OSPF & Policy

| VLAN Assignments | |
|---|---|
| (v=remainder of vlan-id) | |
| Hostname | VLAN-ID |
| host1-a | 100 |
| host2-a | 101 |
| host1-b | 102 |
| host2-b | 103 |
| host1-c | 104 |
| host2-c | 105 |
| host1-d | 106 |
| host2-d | 107 |



## Step 4.1

Enter configuration mode and navigate to the `[edit protocols ospf]` hierarchy level.

```
lab@host1-a> configure
Entering configuration mode

[edit]
lab@host1-a# edit protocols ospf

[edit protocols ospf]
lab@host1-a#
```

## Step 4.2

Identify the interface that connects your assigned device to the directly connected virtual router. If needed, refer to the network diagram for this lab. Remove the identified interface from the OSPF configuration and activate the configuration change.

```
[edit protocols ospf]
lab@host1-a# show
area 0.0.0.0 {
    interface ge-0/0/1.0;
    interface ge-0/0/2.0;
    interface ge-0/0/4.100;
    interface lo0.0;
}

[edit protocols ospf]
lab@host1-a# delete area 0.0.0.0 interface ge-0/0/4.10v

[edit protocols ospf]
lab@host1-a# commit
commit complete
```

## Step 4.3

Navigate to the `[edit routing-options]` hierarchy level. Define a static route for each of the three subnets connected to the virtual router attached to your device. Refer to the network diagram for the destination subnet and next-hop information.

```
[edit protocols ospf]
lab@host1-a# top edit routing-options

[edit routing-options]
lab@host1-a# set static route 172.2x.0.0/24 next-hop 172.20.10v.10

[edit routing-options]
lab@host1-a# set static route 172.2x.1.0/24 next-hop 172.20.10v.10

[edit routing-options]
lab@host1-a# set static route 172.2x.2.0/24 next-hop 172.20.10v.10

[edit routing-options]
lab@host1-a#
```

## Step 4.4

Issue the **show** command to display the resulting configuration. Once satisfied with your configuration, activate the changes and return to operational mode using the **commit and-quit** command.

```
[edit routing-options]
lab@host1-a# show
static {
    route 0.0.0.0/0 next-hop 172.18.1.1;
    route 172.21.0.0/24 next-hop 172.20.100.10;
    route 172.21.1.0/24 next-hop 172.20.100.10;
    route 172.21.2.0/24 next-hop 172.20.100.10;
}

[edit routing-options]
lab@host1-a# commit and-quit
commit complete
Exiting configuration mode

lab@host1-a>
```

## Step 4.5

Issue the **show route protocol static** command to display the current static route entries.

```
lab@host1-a> show route protocol static

inet.0: 17 destinations, 17 routes (17 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0          *[Static/5] 02:50:27
                    > to 172.18.1.1 via ge-0/0/3.0
172.21.0.0/24      *[Static/5] 00:05:00
                    > to 172.20.100.10 via ge-0/0/4.100
172.21.1.0/24      *[Static/5] 00:05:00
                    > to 172.20.100.10 via ge-0/0/4.100
172.21.2.0/24      *[Static/5] 00:05:00
                    > to 172.20.100.10 via ge-0/0/4.100
```

- Are all static route entries active?

______________________________________________

______________________________________________

❖ The answer should be yes. As displayed in the sample capture, the default static route and the three newly defined static routes should all be active. If you do not see four active static routes, check your configuration.

## Step 4.6

Use the **ping** utility to verify reachability to the subnets connected to the attached virtual router. Use the IP addresses assigned to the virtual router as the destination addresses when performing the ping tests.

```
lab@host1-a> ping 172.21.0.1 rapid count 40
PING 172.21.0.1 (172.21.0.1): 56 data bytes
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
--- 172.21.0.1 ping statistics ---
40 packets transmitted, 40 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.937/3.233/10.260/2.022 ms
```

```
lab@host1-a> ping 172.21.1.1 rapid count 40
PING 172.21.1.1 (172.21.1.1): 56 data bytes
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
--- 172.21.1.1 ping statistics ---
40 packets transmitted, 40 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.964/2.868/7.546/1.214 ms
```

```
lab@host1-a> ping 172.21.2.1 rapid count 40
PING 172.21.2.1 (172.21.2.1): 56 data bytes
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
--- 172.21.2.1 ping statistics ---
40 packets transmitted, 40 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.920/3.317/8.719/2.077 ms
```

- Do the ping tests succeed?

_______________________________________________

_______________________________________________

❖ The answer should be yes. As displayed in the sample capture, the ping tests to all three remote destination IP addresses should succeed.

## Step 4.7

Issue the **show ospf neighbor** command to display the current OSPF neighbour adjacencies on your device.

```
lab@host1-a> show ospf neighbor

Address          Interface              State     ID               Pri  Dead
172.20.77.2      ge-0/0/1.0             Full      192.168.2.1      128    38
172.20.66.2      ge-0/0/2.0             Full      192.168.2.1      128    30
```

- How many OSPF adjacencies exist? What is the current state of the OSPF neighbor adjacencies?

❖ Your system should show two OSPF neighbour adjacencies with the remote device. The state should be `Full` for both OSPF neighbor adjacencies, as shown in the sample capture.

## Part 5: Configuring and Monitoring Routing Policy

In this lab part, you will configure and monitor routing policy. First, you will create a routing policy designed to advertise routes in to OSPF. Next, you will apply the routing policy as an export policy under the `[edit protocols ospf]` hierarchy level. You will then use operational mode commands to verify that the policy is working properly. It should be noted that JUNOS routing policy is extremely flexible. Because of this flexibility, you can generally accomplish the same objective in multiple ways. The example configurations provided in the lab guide illustrate one way of accomplishing the stated tasks. Your configuration might vary.

### Step 5.1

Enter configuration mode and navigate to the `[edit policy-options]` hierarchy level.

```
lab@host1-a> configure
Entering configuration mode

[edit]
lab@host1-a# edit policy-options

[edit policy-options]
lab@host1-a#
```

### Step 5.2

Create a new policy named *`default-route`* that matches and accepts the existing default static route. Name the term *`match-default-static-route`*.

```
[edit policy-options]
lab@host1-a# edit policy-statement default-route

[edit policy-options policy-statement default-route]
lab@host1-a# set term match-default-static-route from protocol static

[edit policy-options policy-statement default-route]
lab@host1-a# set term match-default-static-route from route-filter 0/0 exact

[edit policy-options policy-statement default-route]
lab@host1-a# set term match-default-static-route then accept

[edit policy-options policy-statement default-route]
lab@host1-a# show
```

```
term match-default-static-route {
    from {
        protocol static;
        route-filter 0.0.0.0/0 exact;
    }
    then accept;
}
```

## Step 5.3

Navigate to the `[edit protocols ospf]` hierarchy level and apply the recently defined policy as an OSPF export policy. Activate the configuration change.

```
[edit policy-options policy-statement default-route]
lab@host1-a# top edit protocols ospf

[edit protocols ospf]
lab@host1-a# set export default-route

[edit protocols ospf]
lab@host1-a# commit
commit complete

[edit protocols ospf]
lab@host1-a#
```

**Note**

The next lab step requires coordination between both stations assigned to you. Ensure that you have done the above configuration also in the other station (host2-x).

## Step 5.4

Issue the **`run show route 0/0 exact`** command to verify that your device now shows a default OSPF route in the routing table. Check in the other station host2-X to ensure that they also see a default OSPF route in their device's routing table.

```
[edit protocols ospf]
lab@host1-a# run show route 0/0 exact

inet.0: 17 destinations, 18 routes (17 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0          *[Static/5] 03:34:59
                    > to 172.18.1.1 via ge-0/0/3.0
                    [OSPF/150] 00:00:29, metric 0, tag 0
                    > to 172.20.77.2 via ge-0/0/1.0
                      to 172.20.66.2 via ge-0/0/2.0
```

- Does your device show a default OSPF route in the route table?

______________________________________________________________

______________________________________________________________

  - ❖ At this point, both student devices should show a default OSPF route. If you do not see a default OSPF route, check with the remote stations to ensure that you have properly defined and applied the required policy.

- Is the default OSPF route active? Why?

______________________________________________________________

______________________________________________________________

  - ❖ As shown in the sample capture, the default OSPF route is not active due to its higher preference. Because policy injected the route into OSPF, this route is considered an external OSPF route. As you might remember, OSPF external routes use a default preference of 150 whereas internal OSPF routes use a default preference of 10.

- Based on the current default route entry, what would happen if your device's physical connection to the Internet failed?

______________________________________________________________

______________________________________________________________

  - ❖ The current design provides redundancy for this failure scenario. If the physical connection to the Internet fails, your device marks the OSPF default route as active and begins forwarding Internet-bound traffic to the remote student device.

## Step 5.5

Return to the `[edit policy-options]` hierarchy level.

```
[edit protocols ospf]
lab@host1-a# top edit policy-options

[edit policy-options]
lab@host1-a#
```

## Step 5.6

Define a new policy named *`interface-routes`* that matches and accepts the routes associated with your device's interfaces that connect to the Internet and to the directly attached virtual router. Name the term *`match-interface-routes`*.

```
[edit policy-options]
lab@host1-a# edit policy-statement interface-routes
```

```
[edit policy-options policy-statement interface-routes]
lab@host1-a# set term match-interface-routes from route-filter 172.18.x.0/30
exact

[edit policy-options policy-statement interface-routes]
lab@host1-a# set term match-interface-routes from route-filter 172.20.10v.0/24
exact

[edit policy-options policy-statement interface-routes]
lab@host1-a# set term match-interface-routes then accept


[edit policy-options policy-statement interface-routes]
lab@host1-a# show

term match-interface-routes {
    from {
        route-filter 172.18.1.0/30 exact;
        route-filter 172.20.100.0/24 exact;
    }
    then accept;
}
```

## Step 5.7

Navigate to the `[edit protocols ospf]` hierarchy level and apply the *`interface-routes`* policy as an OSPF export policy. Activate the configuration change.

```
[edit protocols ospf]
lab@host1-a# set export interface-routes

[edit protocols ospf]
lab@host1-a# commit
commit complete
```

**Note**

Ensure that you do an equivalent policy also in the other station (host2-x).

## Step 5.8

Issue the **`run show route protocol ospf`** command. Verify that your device shows the OSPF external routes associated with the interfaces of the remote device. Check in the other device to ensure that they also see the proper OSPF routes in their device's routing table.

```
[edit]
lab@host1-a# run show route protocol ospf

inet.0: 19 destinations, 20 routes (19 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0          [OSPF/150] 04:28:05, metric 0, tag 0
                   > to 172.20.77.2 via ge-0/0/1.0
```

```
                      to 172.20.66.2 via ge-0/0/2.0
172.18.2.0/30      *[OSPF/150] 00:01:11, metric 0, tag 0
                      to 172.20.77.2 via ge-0/0/1.0
                    > to 172.20.66.2 via ge-0/0/2.0
172.20.101.0/24    *[OSPF/150] 00:01:11, metric 0, tag 0
                    > to 172.20.77.2 via ge-0/0/1.0
                      to 172.20.66.2 via ge-0/0/2.0
192.168.2.1/32     *[OSPF/10] 06:07:11, metric 1
                      to 172.20.77.2 via ge-0/0/1.0
                    > to 172.20.66.2 via ge-0/0/2.0
224.0.0.5/32       *[OSPF/10] 06:07:26, metric 1
                      MultiRecv
```

♦ Does your device show the expected OSPF routes in the route table?

___

___

❖ At this point, both *host1-x* and *host2-x* devices should show the expected OSPF routes.

## Step 5.9

Return to the `[edit policy-options]` hierarchy level.

```
[edit protocols ospf]
lab@host1-a# top edit policy-options

[edit policy-options]
lab@host1-a#
```

## Step 5.10

Define a third policy named *other-static-routes* that matches and accepts the three recently defined static routes that include destination subnets attached to the virtual router connected to your device. Name the term *match-other-static-routes*.

```
[edit policy-options]
lab@host1-a# edit policy-statement other-static-routes

[edit policy-options policy-statement other-static-routes]
lab@host1-a# set term match-other-static-routes from protocol static

[edit policy-options policy-statement other-static-routes]
lab@host1-a# set term match-other-static-routes from route-filter 172.2x.0.0/24
exact

[edit policy-options policy-statement other-static-routes]
lab@host1-a# set term match-other-static-routes from route-filter 172.2x.1.0/24
exact
```

```
[edit policy-options policy-statement other-static-routes]
lab@host1-a# set term match-other-static-routes from route-filter 172.2x.2.0/24
exact

[edit policy-options policy-statement other-static-routes]
lab@host1-a# set term match-other-static-routes then accept


[edit policy-options policy-statement other-static-routes]
lab@host1-a# show

term match-other-static-routes {
    from {
        protocol static;
        route-filter 172.21.0.0/24 exact;
        route-filter 172.21.1.0/24 exact;
        route-filter 172.21.3.0/24 exact;
    }
    then accept;
}
```

## Step 5.11

Navigate to the `[edit protocols ospf]` hierarchy level and apply the *other-static-routes* policy as an OSPF export policy. Activate the configuration change.

```
[edit policy-options policy-statement other-static-routes]
lab@host1-a# top edit protocols ospf

[edit protocols ospf]
lab@host1-a# set export other-static-routes

[edit protocols ospf]
lab@host1-a# commit
commit complete
```

## Step 5.12

Issue the **`run show route protocol ospf`** command. Verify that your device shows the OSPF external routes associated with the static routes defined on the remote device. Check in the other device (`host2-X`) to ensure that the proper OSPF routes are also seen in the device's routing table.

```
[edit]
lab@host1-a# run show route protocol ospf

inet.0: 21 destinations, 22 routes (21 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0           [OSPF/150] 04:47:16, metric 0, tag 0
                    > to 172.20.77.2 via ge-0/0/1.0
                      to 172.20.66.2 via ge-0/0/2.0
172.18.2.0/30      *[OSPF/150] 00:20:22, metric 0, tag 0
                      to 172.20.77.2 via ge-0/0/1.0
```

```
                        > to 172.20.66.2 via ge-0/0/2.0
172.20.101.0/24        *[OSPF/150] 00:20:22, metric 0, tag 0
                        > to 172.20.77.2 via ge-0/0/1.0
                          to 172.20.66.2 via ge-0/0/2.0
172.22.0.0/24          *[OSPF/150] 00:00:38, metric 0, tag 0
                          to 172.20.77.2 via ge-0/0/1.0
                        > to 172.20.66.2 via ge-0/0/2.0
172.22.1.0/24          *[OSPF/150] 00:00:38, metric 0, tag 0
                          to 172.20.77.2 via ge-0/0/1.0
                        > to 172.20.66.2 via ge-0/0/2.0
192.168.2.1/32         *[OSPF/10] 06:26:22, metric 1
                          to 172.20.77.2 via ge-0/0/1.0
                        > to 172.20.66.2 via ge-0/0/2.0
224.0.0.5/32           *[OSPF/10] 06:26:37, metric 1
                          MultiRecv
```

- Does your device show the expected OSPF routes in the route table?

____________________________________________

____________________________________________

  - At this point, both devices should show the expected OSPF routes.

STOP **You have completed Lab 3 !**