LAB 1 – Simulate unboxing and configuring Router

1. We will be using FRR Routing for this lab. This is a software based router that requires a Linux kernel
2. Our other labs will require “installation” and configuration of multiple, interconnected routers so we will be running each FRR router within a docker container.
3. Why Docker - Docker packages the “router” into a format where we can create multiple instances, and start/stop/configure them independently

Part 1 – Obtain the base FRR Routing image from a public repository like dockerhub

A screenshot of a computer

Description automatically generated

The name of the container we want to get is frrrouting/frr:v8.4.1. Notice the versioning – v8.4.1 is the latest version published. If you don’t specify a version it will use the “latest” version by default. Notice from above that the container with the “latest” flag is only built for the AMD architecture. Note that version 8.4.1 is built for both ARM and AMD so it’s a better solution.

To get the image, from the command line, execute: docker pull frrouting/frr:v8.4.1

Text

Description automatically generated

While you are at it, we will also need to simulate a host, for this we will use an alpine linux container. Get this also via docker: docker pull alpine.

To verify that you have these containers, execute: docker images

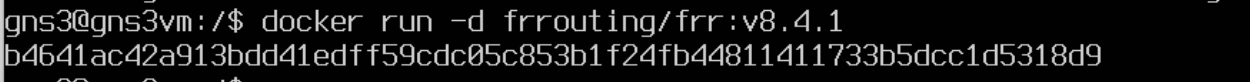
Text

Description automatically generated

Not that I have more containers than you, but you should have alpine:latest and frrouting/frr:v8.4.1. If so, move on.

Part 2 – Simulate powering on your router for the first time so that we can configure it.

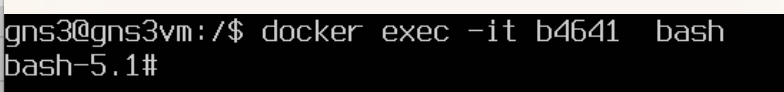
EXECUTE: docker run -d frrouting/frr:v8.4.1



Note the response will be a long hex number, this is the container ID that docker started in the background. Note that what I got starts with “b4641”, your number will be different.

That container is running, but its running in the background, so next we need to shell into it via executing

docker exec -it b4641 bash



Note use the first 4-5 characters from the container ID you got when you executed the docker run command. If everything is successful you will now be shelled into the container.

PART 3 – Configuring the container based router

Graphical user interface, text

Description automatically generated

The main configuration is in the daemons file in the /etc/frr directory.

Edit that file:

Text

Description automatically generated

Change bgpd and ospfd from no to yes, like shown above. This basically enables the router to support the bgp and ospf routing algorithms.

Save the file and exit.

Now set the default configuration for the router control software.



Basically you are creating a file named vtysh.conf that has one line in it “service integrated-vtysh-config”

Last step – change the owner and group of this file to frr:frr via the chown command

Text

Description automatically generated

The above is what the /etc/frr directory should look like

Part 4 – Save your changes into a new docker container

Exit the shell and go back to your vm – just type exit

Now query the running container, this is where your changes were saved

Graphical user interface, text

Description automatically generated

Finally we want to save the b4641 container that we just configured so that we can use it later

Text

Description automatically generated with medium confidence



To save the configured container we issue the docker commit command and give a name for our newly created container. From above, its cs472-router.

Part 5 – Verify Router Configuration

Step 1 – Launch GNS3 and create a new project called setup

Graphical user interface, text, application, email

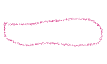
Description automatically generated

Now open preferences

Graphical user interface, text, application, chat or text message

Description automatically generated

Then pick “Docker Containers”

Graphical user interface, text, application, email

Description automatically generated



Then Pick NEW

Graphical user interface, application, Teams

Description automatically generated

Then select the cs472-router

Keep hitting next,

When you get to the number of adapters option, pick 4

Keep hitting next, and ultimately finish

Graphical user interface, application

Description automatically generated

Looks like I messed up and forgot to change the number of adapters, no sweat, we can fix this and need to make one other change.

With CS472-router selected, hit the “Edit” Button

Graphical user interface, application

Description automatically generated

Change Icon if you want



Then go to the advanced tab

Graphical user interface, application

Description automatically generated

Set /etc/frr directory here. If you don’t do this your changes will be lost everytime you start and stop the router



Part 6 setup validation

Diagram

Description automatically generated

Create this topology and connect router-1 to router-2 using eth0

A picture containing diagram

Description automatically generated

Bring up the aux console for each router

On router-1, configure it. For this we use the vtysh utility.

1. vtysh
2. config t
3. int lo
4. ip address 1.1.1.1/32
5. int eth0
6. ip address 10.1.1.1/30

On router-2, configure it. For this we use the vtysh utility.

1. vtysh
2. config t
3. int lo
4. ip address 2.2.2.2/32
5. int eth0
6. ip address 10.1.1.2/30

The above sets a loopback address for router 1 to 1.1.1.1 and a loopback address for router 2 to 2.2.2.2. It also sets the IP address of eth0 on both routers, 10.1.1.1/30 for Router 1, and 10.1.1.2 for Router 2.

Make sure the routers can see each other via pinging each other. To do this issue the “do ping” command. See below:  
  
Text

Description automatically generated

Now have router-2 try to ping router-1 loopback via do ping 1.1.1.1 from router 2. Notice this does not work, but router-2 can ping its own loopback via “do ping 2.2.2.2”

Save your configuration, on each router type “do write” to save the current configurations.

Other commands:

do show interface br

do show interface

do show ip route

Finally, lets setup an alpine desktop to connect a client machine to our network.

Graphical user interface, text, application

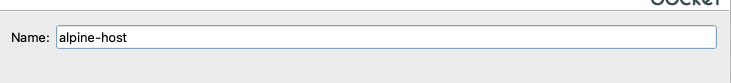
Description automatically generated



Go to GNS3->Preferences->Docker Containers and hit new

Graphical user interface, text, application

Description automatically generated



Then take the rest of the defaults.

Drag an alpine-host to the screen and right click on it and hit edit-config. Change the config like below to assign the host an ip address of 10.1.2.10/24 with gateway of 10.1.2.1

Graphical user interface, text, email

Description automatically generated

Graphical user interface, application, Word

Description automatically generated

Now connect alpine-host-1 with cs-472-router-1

Diagram

Description automatically generated

For anything to happen we now need to configure interface eth1 in router-1 to be the alpine host gateway with IP address of 10.1.2.1/24

Go to the console for router-1

A picture containing schematic

Description automatically generated

Now start the alpine-host, right-click and hit start, then bring up an auxiliary console. Try pinging all of the interfaces in router-1, this should work. Try pinging anything in router-2, this should fail. See below.

Text

Description automatically generated

Try on your own

See the picture below

Diagram

Description automatically generated

Based on this picture, configure interface eth1 with IP address 10.1.3.0/24 on cs472-router-2, add a new host alpine-host-2 with IP address 10.1.3.10.

From alpine-host-2

Ping router-2 interfaces (2.2.2.2 and 10.1.1.2), this should work.

Now try to ping host-1 at 10.1.2.10, this will not work.

Add static route on the routers to fix this.

Router-1, add static route: ip route 10.1.3.0/24 10.1.1.2

Router-2, add static route: ip route 10.1.2.0/24 10.1.1.1

Now check everything.

YOU HAVE NOW SUCCESSFULLY UNBOXED AND SETUP 2 ROUTERS, CONGRATS

Save your configuration with this command: do write

QUESTIONS/DISCUSSION (What to hand in)

1. Why do you think that the 2 routers can see each other via pinging each others IP address on the connected interface?
2. Why do you think that one router cannot ping the other routers local interface?
3. What is the purpose of configuring a local interface on a router?
4. Submit a screen print showing each router pinging the other routers interface.
5. Why did we have to add the static routes at the end. What do you think these commands do?