

Workspace Setup and Build

These instructions will help you to setup a workspace that can run both unit and integration tests (topotests in FR) for FRR development.

Table of Contents

Workspace Setup and Build

Table of Contents

Setting up a workspace for the first time

Workspace Pull

Build the development container

Error: unable to remove repository reference

First launch of the development container

Developing and testing FRR changes

Subsequent Launches of the Development Container

When to rebuild the development container

Attaching to the development container

Detaching from the container without stopping it

Stopping the development container

Removing the development container

Building changes

Build and run the unit tests

Running all topotest suites

Launching a topology with virtual shells

Pausing a topotest on failure

Installing new packages

Setting up a workspace for the first time

Workspace Pull

Starting from your root, create a /src folder where you will clone the FRR repository.

```
git clone git@github.com:FRRouting/frr.git <workspace folder name>
cd <workspace folder name>
git checkout -b <branch>
BRANCH =
In the naming format dev/<userid>.<description>

Example: git checkout -b dev/pbrisset.hr_topotest
```

Build the development container

The development container can be created using the command:

```
docker build -t $(whoami)-$(basename ` /bin/pwd `)-frr-ubuntu20:latest --
build-arg=UBUNTU_VERSION=20.04 --build-arg=UID=$(id -u) -f docker/ubuntu-
ci/Dockerfile .
```

This is slightly different from the standard FRR container creation command (<http://docs.frrouting.org/projects/dev-guide/en/latest/building-docker.html#building-ubuntu-20-04-image>) in two important ways:

1. Your user name is part of the container name to prevent container name collisions with other users on the same host
2. Your linux user ID is specified so that the FRR user within the container has the same ID. This allows you to build FRR within the container using the workspace as mounted on the host. This means any edits to the files on the host will be immediately reflected in the container, saving you from needing to copy files to/from the container.

This should take approximately 5-10 minutes to complete and only needs to be done once for most users

Error: unable to remove repository reference

If you see the following error message, Error response from daemon: conflict: unable to remove repository reference "frr-ubuntu20" (must force) - container 59b45ce05ecb is using its referenced image f507a8e46df0 , you need to stop and

remove other containers that point to it first. In this case `docker rm 59b45ce05ecb` would allow the prior `docker rmi` command to go through.

First launch of the development container

Run the following command from the root of the workspace to launch your development container:

```
docker run --init -it --privileged --name $(whoami)-$(basename `~/bin/pwd`) -
frr-ubuntu20 -v /lib/modules:/lib/modules -v `~/bin/pwd`:/home/frr/frr
$(whoami)-$(basename `~/bin/pwd`)-frr-ubuntu20:latest bash
```

This will drop you into a bash shell with your workspace mounted at `~/frr/frr`.

As the workspace on the host has not been used to build yet, you will need to trigger another build of FRR to finish your workspace setup:

```
cd ~/frr
./bootstrap.sh
./configure --prefix=/usr \
  --localstatedir=/var \
  --sbindir=/usr/lib/frr \
  --sysconfdir=/etc \
  --enable-vtysh \
  --enable-pimd \
  --enable-sharped \
  --enable-multipath=64 \
  --enable-user=frr \
  --enable-group=frr \
  --enable-vty-group=frrvty \
  --enable-snmp=agentx \
  --enable-scripting \
  --with-pkg-extra-version=-my-manual-build
make -j16
sudo make install
exit
```

If you want to use `gdb` and avoid falling into optimized-out code, you can add the following to the `./configure` command:

```
--enable-dev-build
```

Developing and testing FRR changes

Make sure to go through all of the first time development steps in the prior sections.

All of the following commands/sections are intended to be **run from within the development container shell**.

More topotests documentation is available here: <https://docs.frrouting.org/projects/dev-guide/en/latest/topotests.html>

Subsequent Launches of the Development Container

From the root of the workspace run:

```
docker start $(whoami)-$(basename `~/bin/pwd`)-frr-ubuntu20
```

When to rebuild the development container

- When a patch modifies the Dockerfile initially used to build the container

Attaching to the development container

```
docker attach $(whoami)-$(basename `~/bin/pwd`)-frr-ubuntu20  
cd ~/frr
```

Detaching from the container without stopping it

```
Ctrl+p, Ctrl+q
```

Stopping the development container

```
docker stop $(whoami)-$(basename `pwd`)-frr-ubuntu20
```

Removing the development container

You need to run both of the following commands; `docker rm` removes the container filesystem and `docker rmi` removes the base image itself.

```
docker rm $(whoami)-$(basename `pwd`)-frr-ubuntu20
docker rmi $(whoami)-$(basename `pwd`)-frr-ubuntu20
```

In case there are any test files in the workspace that you cannot remove, use the following command from the root of the workspace to delete them:

```
docker run --rm -v `pwd`:/mount alpine sh -c 'rm -rf /mount/tests'
```

Building changes

```
make -j16 && sudo make install
```

Build and run the unit tests

```
make check
```

Running all topotest suites

```
cd tests/topotests && sudo pytest -n16 --dist=loadfile
```

Launching a topology with virtual shells

Ex. To launch the topology `bgp_evpn_mh` and connect to the router `torm11`

Note: This ****MUST**** be run from within `tmux`. An error will be seen if not.

[tmux cheatsheet](#)

```
cd tests/topotests && sudo -E pytest -s --pause --vtysh=torm11 bgp_evpn_mh
```

From the VTU, the following outputs can be seen:

```
torm11# show evpn es
Type: B bypass, L local, R remote, N non-DF
ESI                               Type ES-IF                               VTEPs
03:44:38:39:ff:ff:01:00:00:01     LR    hostbond1                               192.168.100.16
03:44:38:39:ff:ff:01:00:00:02     LR    hostbond2                               192.168.100.16
03:44:38:39:ff:ff:02:00:00:01     R     -
192.168.100.17,192.168.100.18
03:44:38:39:ff:ff:02:00:00:02     R     -
192.168.100.17,192.168.100.18
torm11# show evpn es-evi
Type: L local, R remote
VNI      ESI                               Type
1000     03:44:38:39:ff:ff:01:00:00:01     L
1000     03:44:38:39:ff:ff:01:00:00:02     L
```

Pausing a topotest on failure

```
cd tests/topotests && sudo -E pytest --vtysh-on-error bgp_evpn_mh
```

Instal

ling new packages

```
sudo -E apt-get ...
```