Streaming Telemetry Demo

This is a containerized demo of streaming telemetry. Download xr-d control-plane image from cisco.com using your cco account https://www.cisco.com/c/en/us/support/routers/ios-xrd/series.html#~tab-downloads

Clone the repo

Clone the git repo

```
git clone https://wwwin-github.cisco.com/spa-ie/ios-xr-streaming-telemetry-
demo.git
cd ios-xr-streaming-telemetry-demo
```

Host Requirements

Regardless of which type of Docker host you are using, the following requirements apply in order to be able to use XRd successfully:

- Docker version 18+ (and permission to run Docker containers)
- Linux kernel version 4+
- Linux cgroups version 1 (unified hierarchy cgroups not yet supported)
- Resource requirement depends on load and intended scale
- 2GB RAM per XRd instance
- 2000 inotify user instances/watches per XRd instance

To verify that your chosen host meets all the requirements to support XRd in Docker, run the host-check script `./host-check`.

NOTE: Running `sudo sysctl -w fs.inotify.max_user_instances=64000` most likely will be the only fix needed :) .

Extract the contents of the tar file `tar -xvzf xrd-control-plane-container-x64.7.7.1.tgz`

Load the docker image by loading the downloaded tar file `docker load -i xrd-container-x64.dockerv1.tgz`

After the image is available locally and recommended fixes are applied. Luanch the simple bgp topology by running this command. Be sure to replace the image name with the name present when you run the command `docker images`.

```
sudo ./xr-compose -f samples/simple-bgp/docker-compose.xr.yml  -i
localhost/ios-xr:7.7.1 -l
```

This will bring up the topology. You can verify that the topology is up by running `docker ps`

```
richu@richu-VirtualBox:~/Desktop/ios-xr-streaming-telemetry-demo$ docker ps
CONTAINER ID   IMAGE                        COMMAND
CREATED          STATUS          PORTS     NAMES
2664a33be181   localhost/ios-xr:7.7.1.09I   "/bin/sh -c /sbin/xr…"   4
seconds ago   Up 1 second               xrd-1
2cc1f9ad737c   localhost/ios-xr:7.7.1.09I   "/bin/sh -c /sbin/xr…"   4
seconds ago   Up 1 second               xrd-2
d4a439c44f71   ubuntu:14.04                 "/bin/bash -c 'ip ro…"   4
seconds ago   Up 2 seconds              dest
4f44778487c2   ubuntu:14.04                 "/bin/bash -c 'ip ro…"   4
seconds ago   Up 2 seconds              source
```

Wait for atleast 5 minutes so that the images are booted up and running properly.

Streaming Telemetry Stack.

This demo has a custom stack consisting of Telegraf,InfluxDB,Grafana(TIG), With Prometheus and Kafka also alongside it.

Note: With InfluxDB 2.2, influx has moved into a token based authentication instead of username/password. If you want to run influxdb , you will need to login to the web portal , generate a token. Copy the token inside the telemetry/telegraf/telegraf.conf file, and restart telegraf for it to work properly

Note: Kafka takes some time to boot up and Telegraf has an issue where if it detects an ouput is not up (kafka in this case). It will quit the applicaton causing the container to stop. You can fix this by restarting telegraf after waiting for kafka to be up by running the command `docker restart telemetry_telegraf_1` and it would fix the problem.

You can view the telemetry in 2 ways via Grafana and Kafka. For sake of simplicity Grafana is configured to poll data from prometheus instead of influxDB due to token issue.

Bring the stack up by running the commands.

```
cd telemetry
docker-compose up -d
```

You can access grafana on localhost:3000. Look at the dashboards available to see packet counts on interfaces. To view the data being streamed into kafka , There is a consumer application written.

```
docker logs -f telemetry_consumer_1
```

There is also yansguite spun up available at localhost:8443

Cleanup

Run `docker-compose down` in the ios-xr-streaming-telemetry-demo folder to bring down the xr-d topology. Additionally run `./cleanup` script to remove volumes too.

Run

```
cd telemetry
docker-compose down
```

To bring down the TIG , Prometheus and Kafka stack.

Additional Notes:

If you need to bring up any other topology other than Simple BGP , you can launch other topology by replacing the file name with the any of the topology available in samples folder. Please note that the configuration in telegraf to for dial-in telemetry has been configured for 2 routers. Please add the gnmi configuration for additional routers which will be brought up by the new topology.

This demo makes heavy use of xr-d. Learn more about xr-d here https://wwwin-github.cisco.com/pages/xrd/