Franklin Abodo

Reuben Hooley

COP5614

**11/30/2018**

**OS Final Project**

The final project was implemented in four stages:

1. Implementing two applications that utilize multiple containers to communicate over a network,
2. inserting a Loadable Kernel Module into the linux kernel that captures communications between those applications and logs pairs of communicating IP addresses to a file,
3. a Python program that loads the IP log file and visualizes it dynamically,
4. and a Docker image of the LKM Program.

Below are the step-by-step instructions to implement a Hadoop Instance with four containers and a website in Docker Compose with 5 containers. Instructions are also provided via a video at this Youtube link: https://youtu.be/UfoBqfeJGts

**Stage 1: Linux Kernel Module**

The LKM is to be inserted first so that it is available to containers when they are first instantiated.

1.1 Unzip the project archive.

1.2 Make and run the LKM:

1. Navigate to the /lkm folder
2. From the Terminal, run: make
3. To insert the module into the kernel, run: *sudo insmod* [*lkm.ko*](https://github.com/proboscisjoe/final-project/blob/master/lkm/lkm_process_info.ko)
4. To remove the module from the kernel (after completing network topology visualization in Stage 3), run: *sudo rmmod* [*lkm.ko*](https://github.com/proboscisjoe/final-project/blob/master/lkm/lkm_process_info.ko)

While the LKM program was running, a Python program was made to access the network packet information, to filter the information based upon the IP addresses of the containers within the running programs of section 1a and 1b. This program also provides a graphic visualization of the network topology.

**Stage 2: Applications**

* 1. Implement Website Linux Command Line Instructions

1. Navigate to the /composetest directory
2. Install docker compose:

*sudo curl -L "https://github.com/docker/compose/releases/download/1.23.1/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose*

1. Initialize it: *sudo chmod +x /usr/local/bin/docker-compose*
2. Verify the Version: *docker-compose –version*
3. Install docker:

sudo apt install docker.io

1. Make the LKM and copy it into the Docker build context:  
   *cd ../lkm*

*make*

*cp lkm.ko ../composetest*

1. Run the docker website image:

*sudo docker run -p 127.0.0.1:4000:80 rhool001/basicsomesomenot*

1. Create a swarm that uses 5 containers to run the basic website: *sudo docker swarm init*
2. Deploy the Website with the created docker compose files: *sudo docker stack deploy -c docker-compose.yml getstartedlab*
3. Check that the service is running: *sudo docker service ls*
4. Show the containers: *sudo docker container ls -q*
5. Open a web browser to check that the service is running at: http://127.0.0.1:4000/
   1. Implement Hadoop
6. Pull hadoop from cmd line: *sudo docker pull cloudsuite/hadoop*
7. Pull cloudsuite data analytics: *sudo docker pull cloudsuite/data-analytics*
8. Make a network: *sudo docker network create hadoop-net*
9. Run the master node: *sudo docker run -d --net hadoop-net --name master --hostname master cloudsuite/data-analytics master*
10. Run 3 slave nodes:
    1. *sudo docker run -d --net hadoop-net --name slave01 --hostname slave01 cloudsuite/hadoop slave*
    2. *sudo docker run -d --net hadoop-net --name slave02 --hostname slave02 cloudsuite/hadoop slave*
    3. *sudo docker run -d --net hadoop-net --name slave03 --hostname slave03 cloudsuite/hadoop slave*
11. Inspect the network for the containers: *sudo docker network inspect hadoop-net*
12. Run the Network: *sudo docker exec master benchmark*

**Stage 3 – Network Topology Visualization**

Now that there are two container networks running, a LKM output can be visualized. This program uses Netfilter hooks to listen in the kernel for all network packets and log their to/from IP addresses. Each IP instance is read from the log file and counted by the Python program.

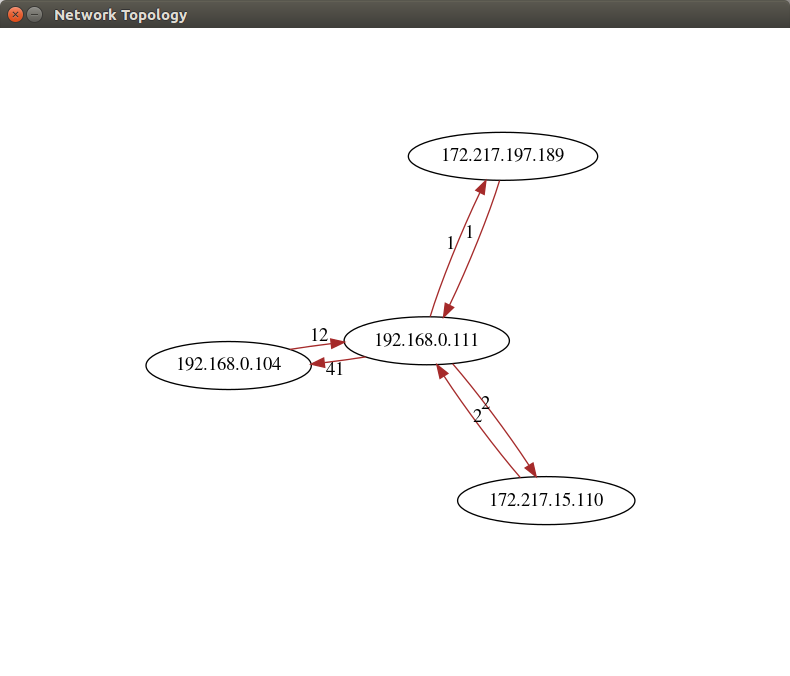
The visualization application was implemented using Python 3.6, which is expected to already be installed either natively or using a virtual environment like Anaconda.

3.1 Install Dependencies

1. sudo apt-get install pip
2. pip install pygraphviz
3. pip install pillow
4. pip install wxPython

3.2 Run Python Visualization

1. Change to the viz directory
2. Run python app.py



*Fig 1. Example network topology graph*

**Stage 4 – …**