## Setting Up Lab Environment Using Docker Compose

Containers are used to set up the network and run different experiments. Docker provides a tool called Compose, which simplifies the entire process. To build, start, and shutdown the lab environment, the following commands can be used:

* docker-compose build
* docker-compose up
* docker-compose down

Other useful Docker Commands [3]:

* docker network ls # List all the networks
* docker container restart <id> # Restart a container
* docker container stop <id> # Stop a container
* docker container start <id> # Start a container
* docker ps # List all the running containers, including their IDs
* docker ps -a # List all the containers (running or not)
* docker exec -it <container id> /bin/bash # Execute the /bin/bash command in a running container
* docker rm <container id> # Remove a container

## Lan setup

To setup the environment, download the tcp\_syn\_flood\_attcah folder. Compose is used to create a lab environment setup that consists of a LAN and containers, which are shown in Fig. below. The docker is already created and it should be up by using Compose command.



To setup network, we should create the containers by Docker in the follow steps:

* In the same folder, where the *docker-compose.yml* file is, run *$ docker-compose up*
* Now a network (net-10.9.0.0) has been setup and it can be shown by *docker network ls* or *ifconfig*
* In this network there are Node A (A-10.9.0.5), Node B (B-10.9.0.6), Attacker (M-10.9.0.105), and your machine (10.9.0.1)

## Attack & Detection Tasks

In this task, the victim (A-10.9.0.5) starts the detection module and waits for any incoming syn attacks.

* 1. To launch the module, we should do it on containers as follows:
* Find attacker & victim’s container’s ID by: *# docker ps*
* Execute the /bin/bash command on attacker container by: *# docker exec -it <attacker container id> /bin/bash*
* Execute the /bin/bash command on Host A container by: *# docker exec -it <A container id> /bin/bash*
* Launch detection on A’s container by: *$ ./volumes/tcp\_syn\_detect.py* (this code should be at *volumes* folder in VM)
* Launch attack on Attacker container by *$ ./volumes/tcp\_flooding.py* (this code should be at *volumes* folder in VM)
* The prevention commands of iptables should also be done on A’s container.