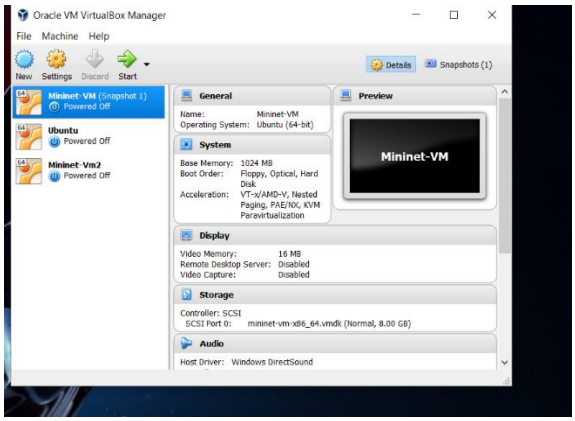
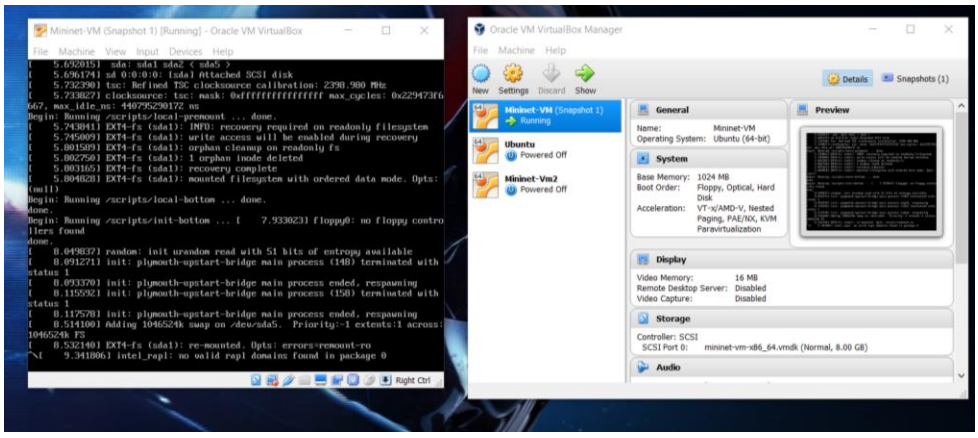
README

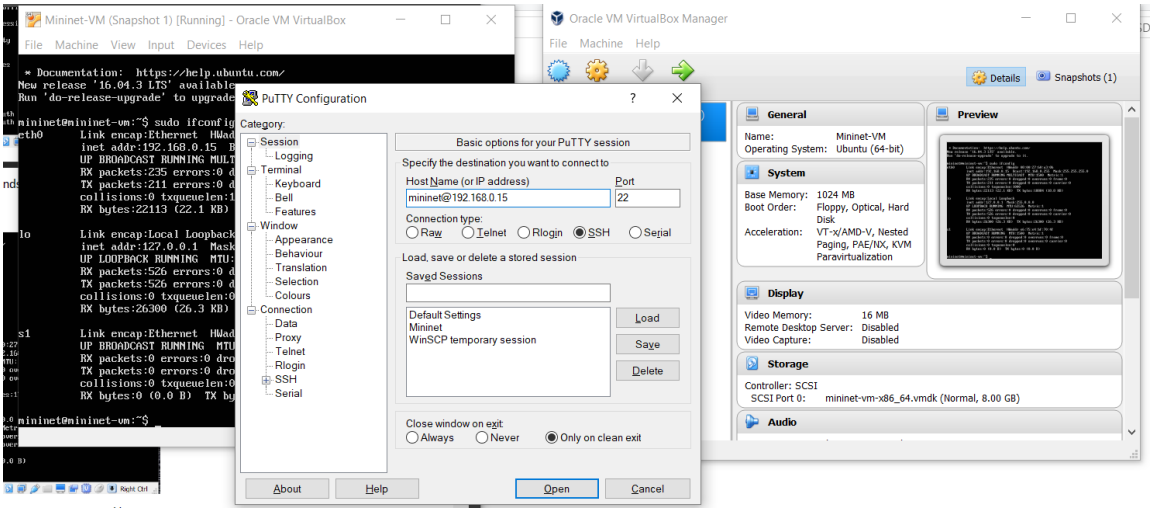
Instructions to run Advanced Load Balancer:

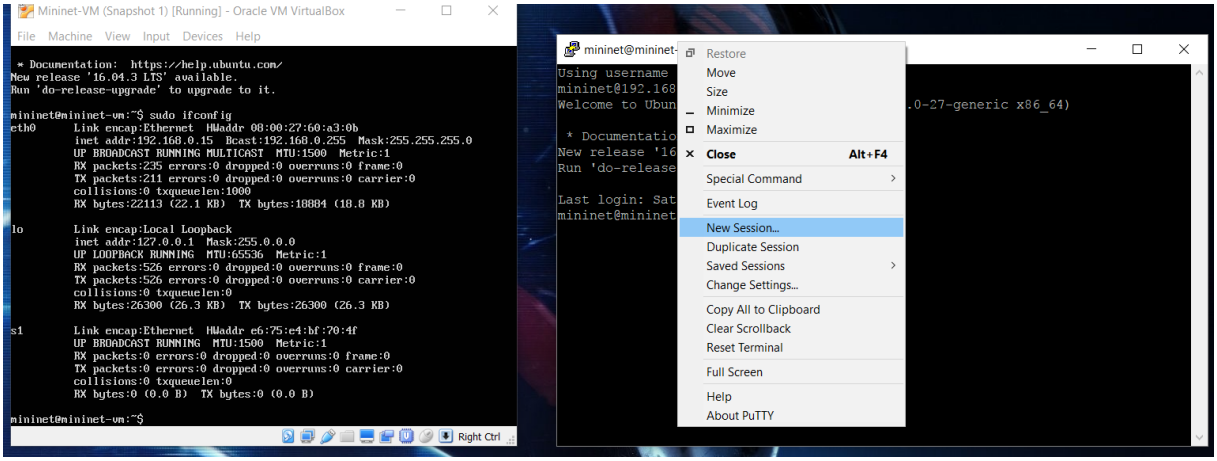
1. Run Virtual Machine in VirtualBox



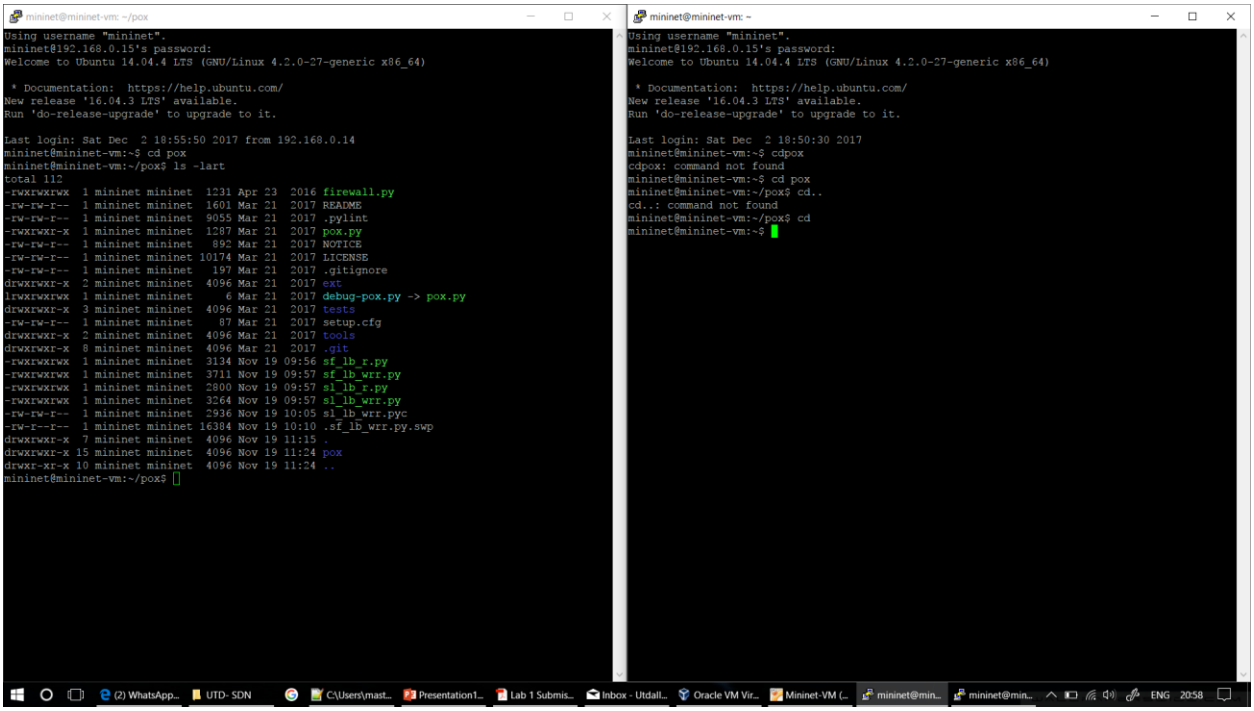


2. Run putty and login to mininet with the ip address of Virtual Machine and connection type SSH





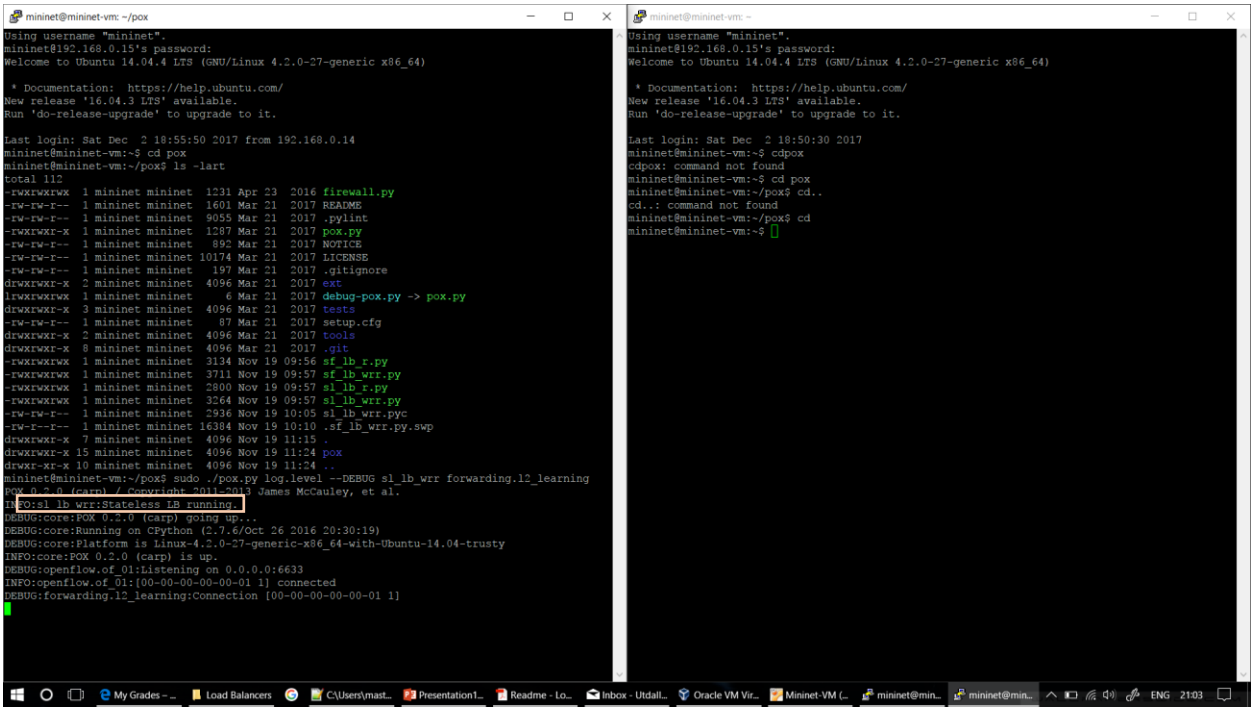
3. Once the mininet is running, create a new session of Putty and login with mininet credentials



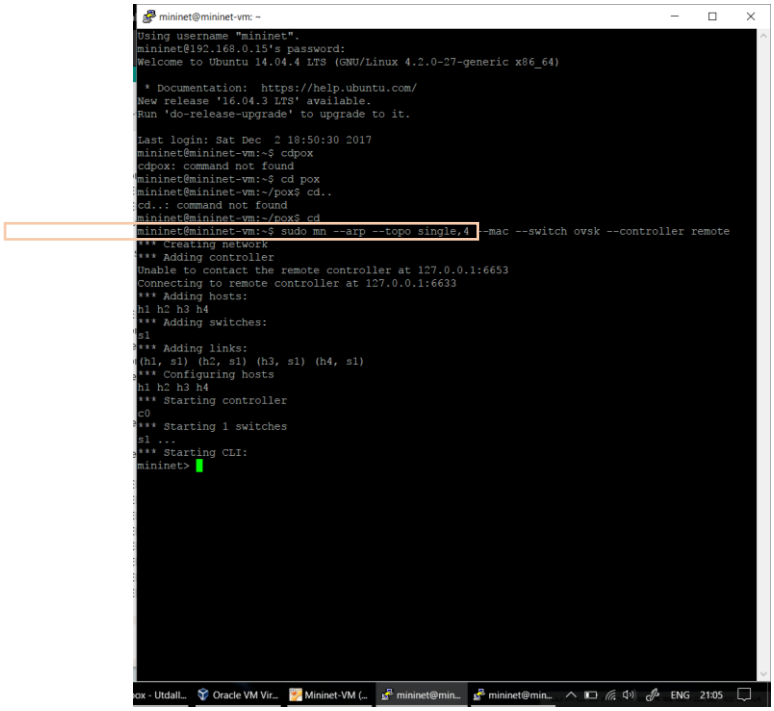
4. On one mininet instance, we run the POX controller and on the other mininet instance we run the topology-Switches, Hosts, Links

5. Copy the Python (Load Balancer) scripts to POX controller mininet session

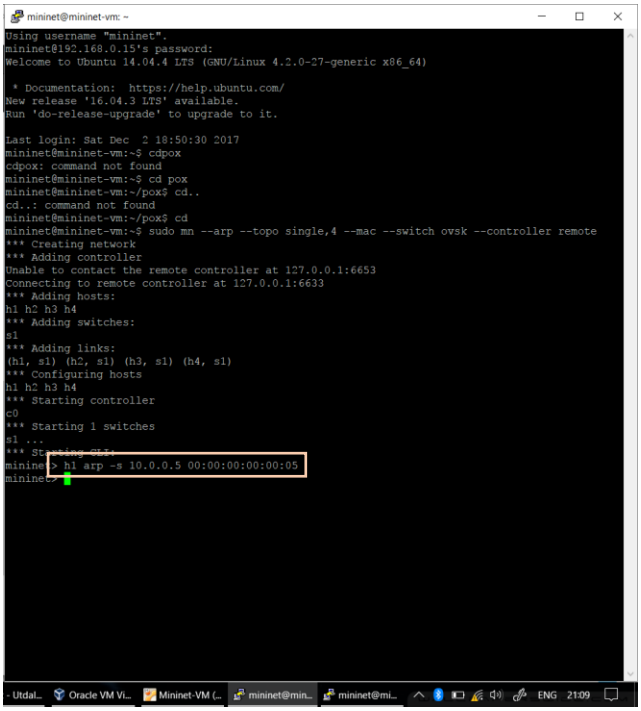
6. On POX controller, run the Command: *sudo ./pox.py log.level --DEBUG sl_lb_wrr forwarding.l2_learning* .This will start the LoadBalancer (Stateless, WeightedRoundRobin).



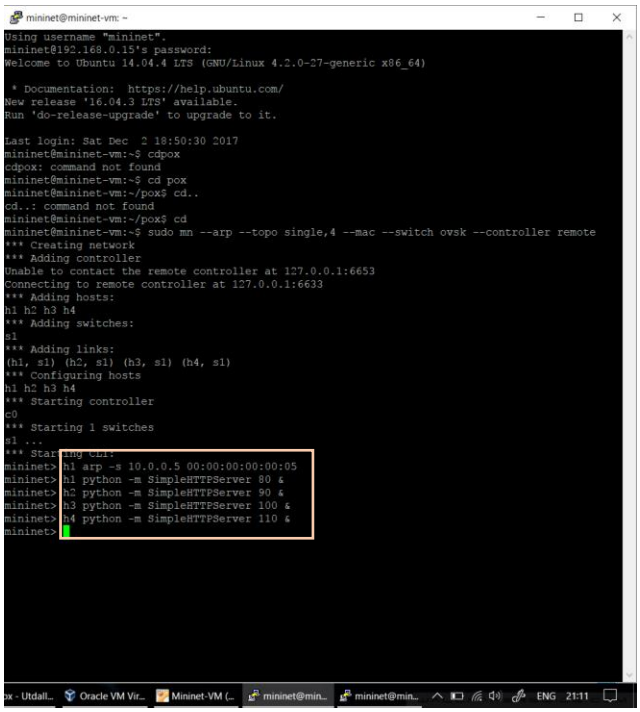
7. On the other mininet session, we create the topology using the following command:
$ sudo mn --arp --topo single,4 --mac --switch ovsk --controller remote .



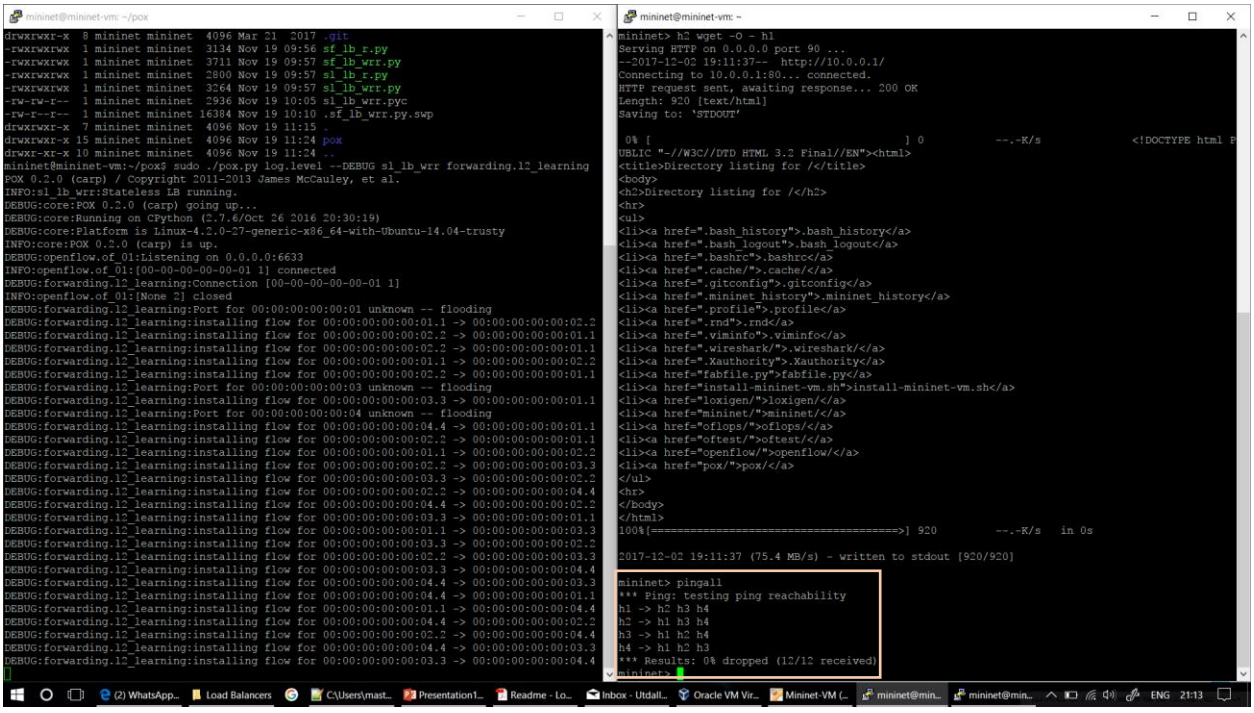
8. Now we select a virtual IP (VIP) and MAC for the load-balancer. This is the IP address to which the clients will make a HTTP request. The controller will push rules to rewrite the VIP with the selected HTTP server. To make this work, you need to static set an ARP entry for the VIP in the client. If ‘h1’ is the client and 10.0.0.5 is the VIP, the following command will add the static ARP entry: *h1 arp -s 10.0.0.5 00:00:00:00:00:05*



9. Now traffic is generated by running SimpleHTTPServer webservers on the hosts by using following command.
    a. h1 python -m SimpleHTTPServer 80 &
    b. h2 python -m SimpleHTTPServer 90 &
    c. h3 python -m SimpleHTTPServer 100 &
    d. h4 python -m SimpleHTTPServer 110 &



10. We can test pingability using the command: *pingall*



11. Now when a client performs the following command, we will receive the IP address of the handling server. Command: *h1 wget* http://10.0.0.5:8000
    Techniques used:
    a. **Stateless Weighted Round Robin**: All requests coming from the client (h1) will be assigned to different servers (h2, h3, h4) on the order of their weights. Order of weights assigned: h2>h3>h4. So, the first request will be assigned to h2 followed by h3 and h4
    b. **Stateless Random**: All requests coming from the client (h1) will be assigned to different servers (h2, h3, h4) on a random order.
    c. **Stateful Weighted Round robin**: All requests coming from the client (h1) will be assigned to different servers (h2, h3, h4) on a random order. Order of weights assigned: h2>h3>h4. So, the first request will be assigned to h2 followed by h3 and h4. Also, as it is stateful so all requests from the same client will be directed to the same server.
    d. **Stateful Random:** All requests coming from the client (h1) will be assigned to different servers (h2, h3, h4) on a random order. But all the requests from the same client will be directed to the same server as it is stateful.

12. Repeat the steps 6-11 for the other Python scripts to Implement different load balancing techniques.