Program design:

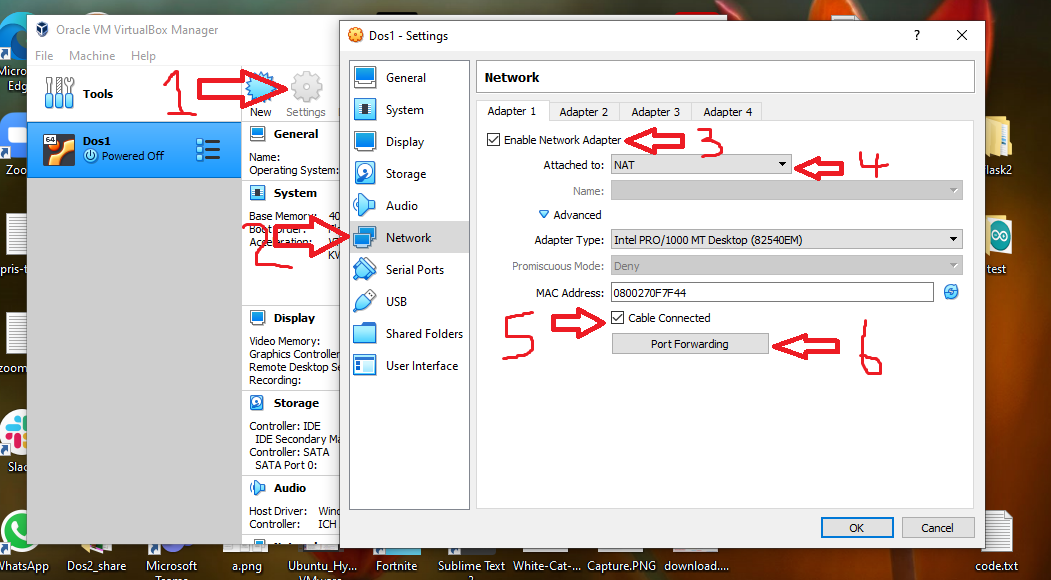
The program consists of 3 main flask microservices (client, catalog-server and order-server) each server has one replica and these microservices talk to each other using REST API.

How it works:

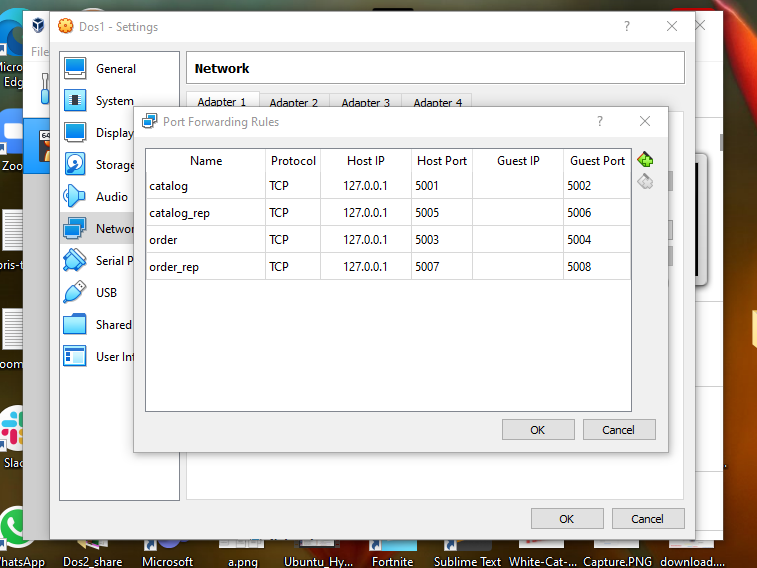
First, we run all flask microservices using command prompt (cmd) in windows and terminal in ubuntu. Second, open web browser like Firefox, google chrome, etc. And type the URL (search, lookup and buy) then the client flask microservice handles these requests. In Search and lookup requests the client checks the result of these requests in cache or not, in the cache hit the client sends the response directly without contacting with catalog-server and in the cache miss then the client sends a request to one of the two catalog servers based on round-robin algorithm for load balancing and put the response in the cache based on Least Recently Used (LRU) replacement policy and send the response. In Buy request the client flask microservice sends a request to one of the two order servers based on round-robin algorithm for load balancing then order server send a query request to catalog-server to check if the product’s quantity exist, if ran out of the stock then the order server receives from catalog server a message including that the product does not exist then order server sends a response to the client that the process of buy request failed. If not out of the stock then order server sends an update request to catalog server then catalog server decreases the product’s quantity by 1 then order server receives from catalog server a message including that the product does exist then order server sends a response to the client that the process of buy request successes then client flask microservice remove the result of lookup requests from the cache because the results become invalid.

How to run the program:

1. In host machine create a new flask microservice and copy the client code from GitHub and put it in this microservice.
2. Make sure that virtual machine turns off.
3. Open virtual box then goes to settings then go to network and Mack sure that you check Enable Network adapter , select Nat and check the Cable Connected then pressed on port forwarding as image below:



1. Add 4 rules as image below:



1. Run the virtual machine.
2. Create 4 flask microservices (catalog-server, catalog-server-replica, order-server and order-server-replica) and copy the code of these microservices from GitHub.
3. Run all microservices and test the program.

Performance results:

|  |  |  |
| --- | --- | --- |
| Cache/Request | Search/Lookup | Buy |
| No cache | **Time response > 0** | **Time response > 0** |
| With cache | **Time response = 0** | **Time response > 0** |

We conclude that in search/lookup requests the cache improve the performance but in buy request the cache does not improve performance because the buy request update the data on database and search/lookup requests read only from database so we can store data in cache.