**Docker Container Setup Commands On Linux Machine (Sample for TensorFlow container) - Neel Shah**

1. docker pull tensorflow/tensorflow
2. docker run -it -p 8888:8888 tensorflow/tensorflow
3. apt update
4. apt install git
5. pip install jupyterlab
6. mkdir neel\_tmp (Make the tmp directory and make the work init, It is okay to make the directory by any other name also)
7. cd neel\_tmp
8. jupyter lab --ip 0.0.0.0 --port 8888 --allow-root
9. Open the link from the terminal in the browser
10. Upload / clone the repo of the code and start working on it.
11. Now the whole work done in the directory will be deleted, Once the container is stoped, So we have to save the copy of that container before closing the container.
12. Now open new terminal and write the following commands, without closing the iterative container terminal:
    1. docker ps (And identify the id number of current running docker container)
    2. docker commit <old-running-container-name> <docker-user-name/new-container-name>
    3. docker login (complete docker login process)
    4. docker push <docker-user-name/new-container-name>:<tag-name>

→ Example of the tag name is the latest, 0.0.2, 0.0.3, and etc.

* 1. And then, when you want to get the container, You have to write the,

docker pull <docker-user-name/new-container-name>:<tag-name>

→ And you would get the docker container, with the whole code, environment, model and each and every other thing that was in the last container.

1. (Optional) For locally saving the container and loading the container from the .tar file
   1. For saving the container in .tar format locally
   2. docker save -o <tobe-saved-file-name>.tar <container-image-name>
   3. For loading the saved container from locally
   4. docker load -i my\_custom\_image.tar
   5. For running the container
   6. docker run -it -p 8888:8888 <container-image-name>
2. To save the model and it’s weight on the huggingFace follow the following steps
   1. TODO