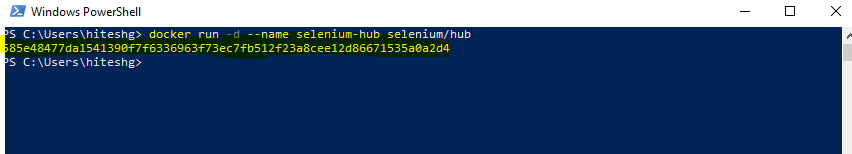
**Hub Starting:**

**Once all the images are installed, first steps is to start the selenium hub.**

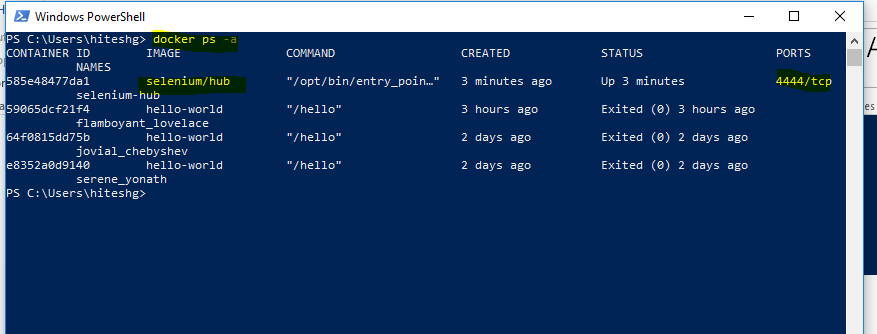
1. Issue the following command to run the hub (Note: Step # 3 is preferred to start the hub on a particular port).

$ docker run –d --name selenium-hub selenium/hub

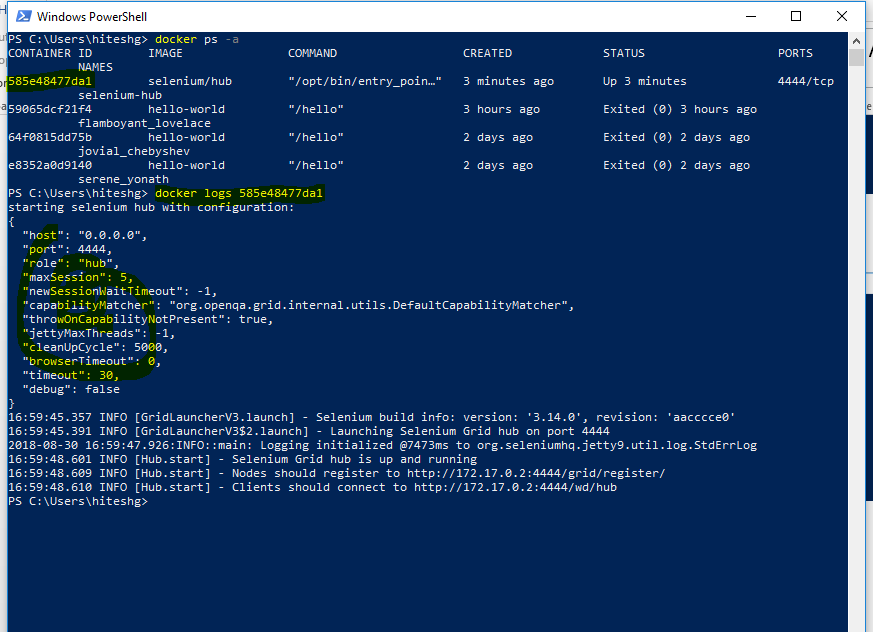


Now, the hub is started

1. Issue $ docker ps –a command to see whether hub process is running or not.



1. Issue $ docker logs ‘hubprocess id’ to see the running hub details



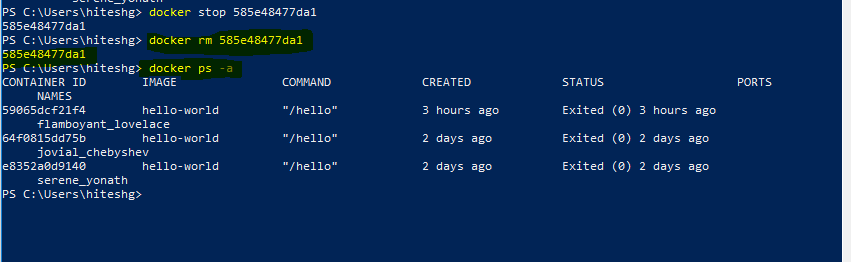
Important note: Above commands will run the hub on the local system? But we have to expose it outside world so following steps need to be followed.

Step 1: First stop the hub

$ docker stop ‘hub process id’

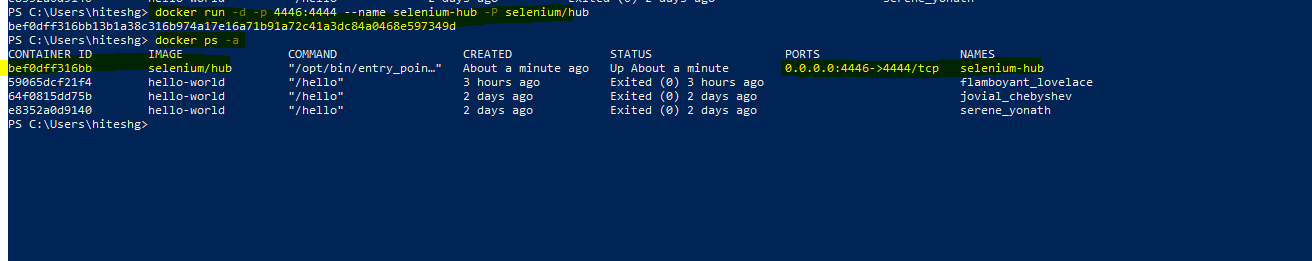


Step 2 : Remove the hub process



Step 3: Issue the following command which maps the local port with another port which exposes the hub outside world

$ docker run -d -p 4444:4444 --name selenium-hub -P selenium/hub



**Start NODES**

**After starting the hub, now the second step is to start the nodes.**

1. First node start – chrome debug node

Issue the following command to start the chrome debug node

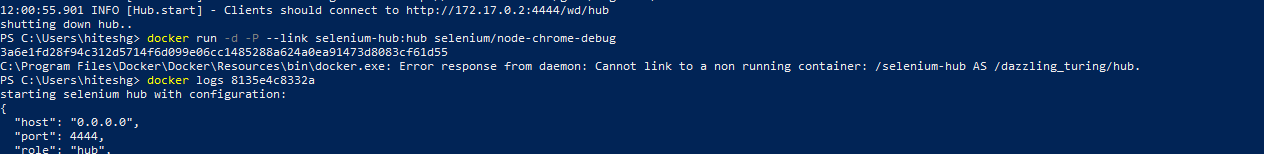
$ docker run –d –P --link selenium-hub:hub selenium/standalone-chrome-debug:3.6.0-bromine

Issue $ docker ps –a ---To know whether the node is started or not

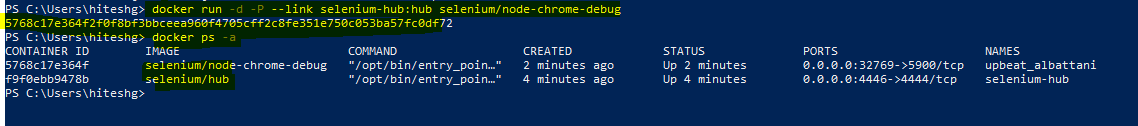
Troubleshooting 1 : It may be possible that while connecting the node to hub, we may encounter ‘invalid reference format’ error.

Solution: Update the docker version and restart the system.

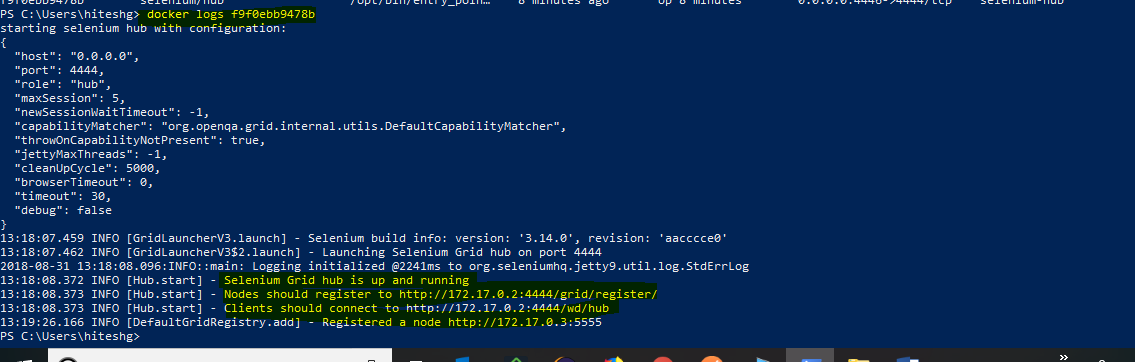
Troubleshooting 2: The following error ( cannot link to container ) may occur while connecting the node to hub



Solution: Stop and remove the hub (by using the commands lists in hub starting section) and then restart the hub, it will connect the node to hub like below:



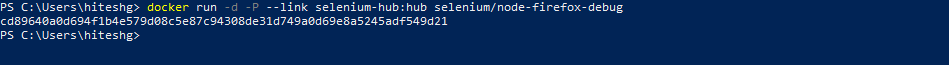
Now see the logs for hub process to see whether the node is registered to hub or not

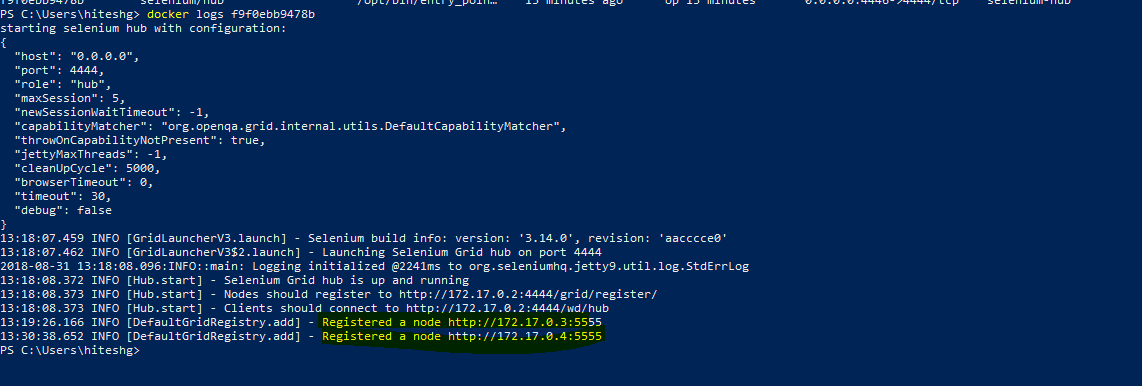


Start second node – Firefox debug

Issue the following command

docker run -d -P --link selenium-hub:hub selenium/node-firefox-debug





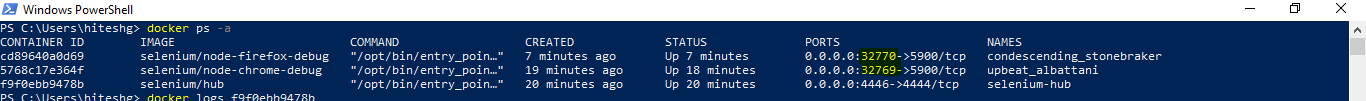
**Check the running of configured nodes by using VNC**

Now, we need to see the configured nodes. VNC viewer can be used to see the configured nodes

Step 1: Open VNC and enter the address of node as below:

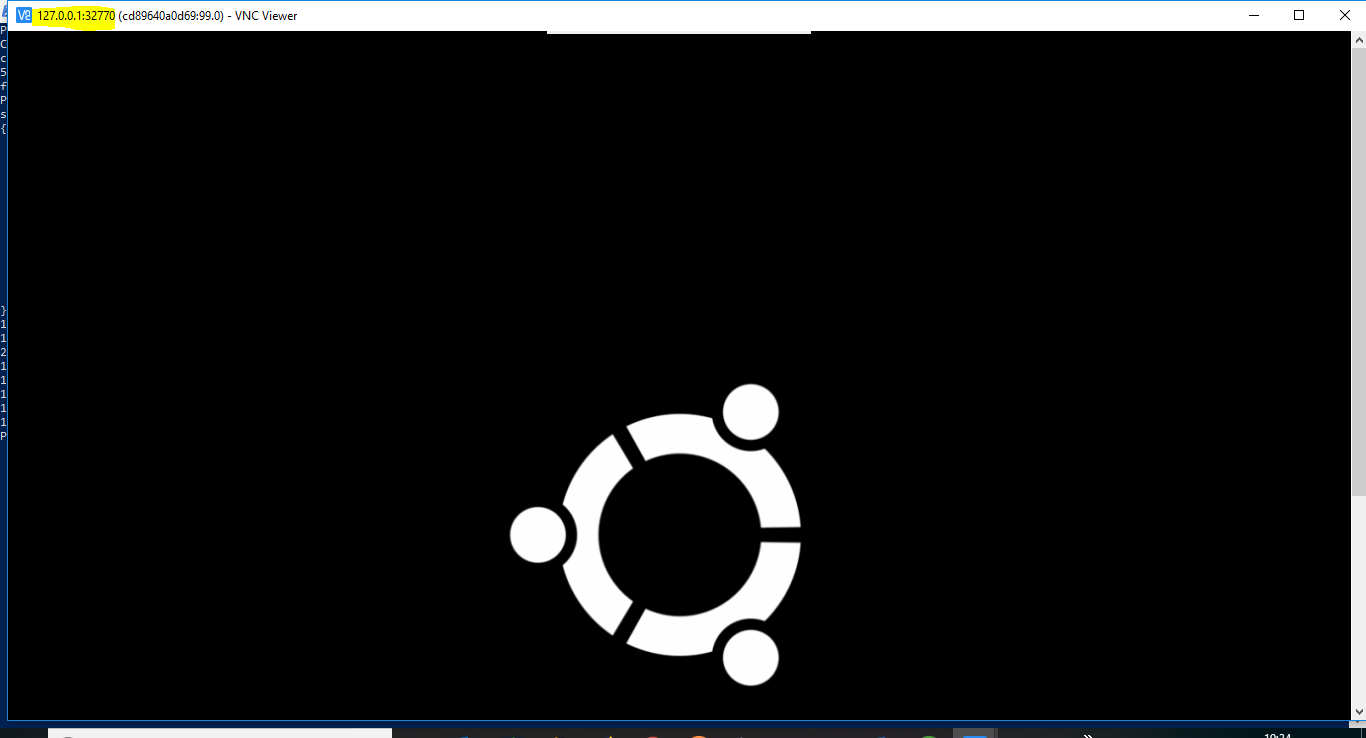
127.0.0.1:32770

Here, port number 32770 can be picked by issuing the following command

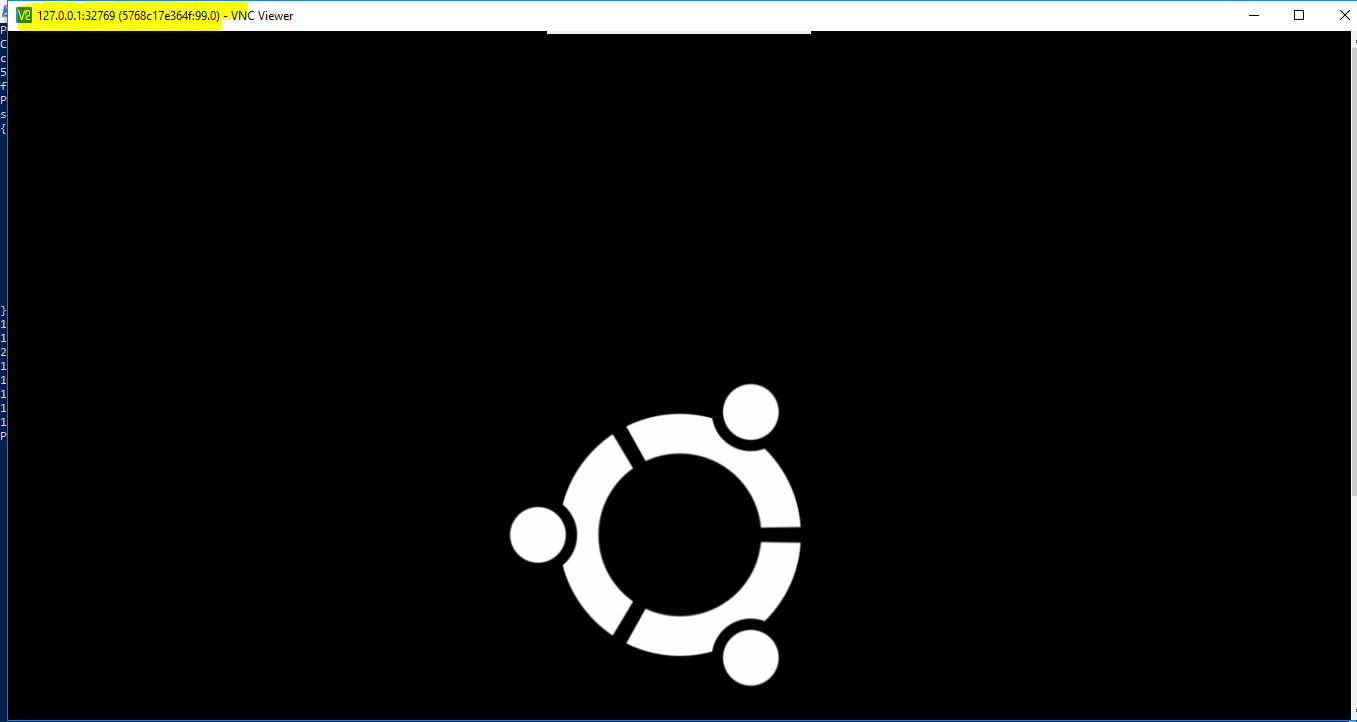


Use ‘secret’ as a password to login into vnc

After login into VNC, the following window of node should be displayed.



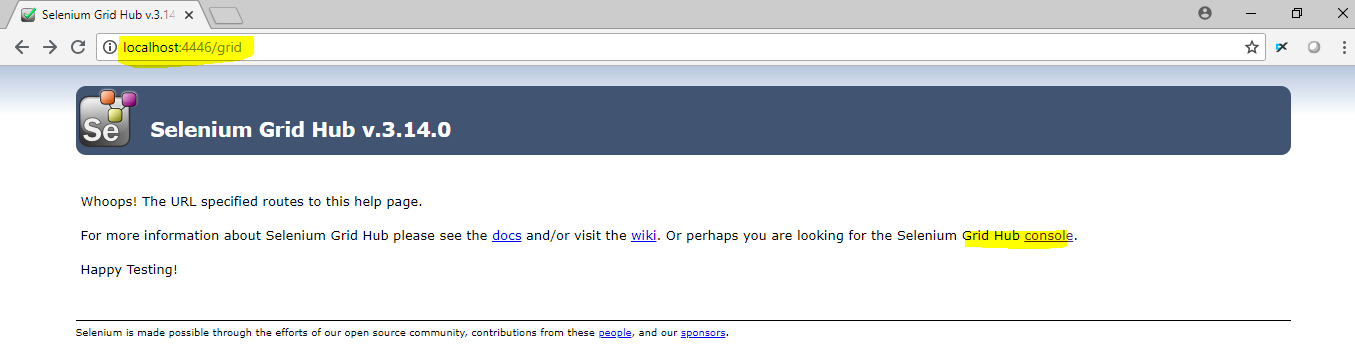
Similarly, open the another node window by using VNC



**Checking the node configuration:**

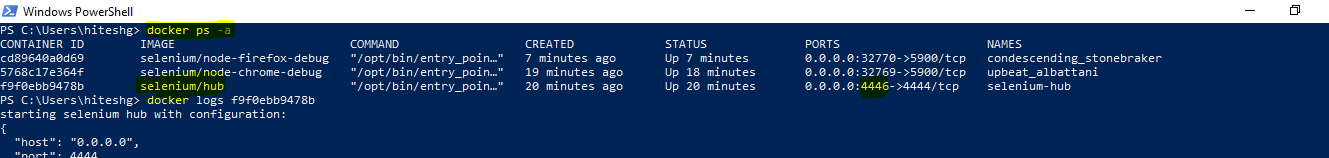
We can also view which nodes are running and configured with hub by using the following steps

Open browser and hit <http://localhost:4446/grid/console>





Note: We have to use the right port to view the grid console. Here we have used ‘4446’ port which can be picked from the following screen shot.



**Batch file creation**

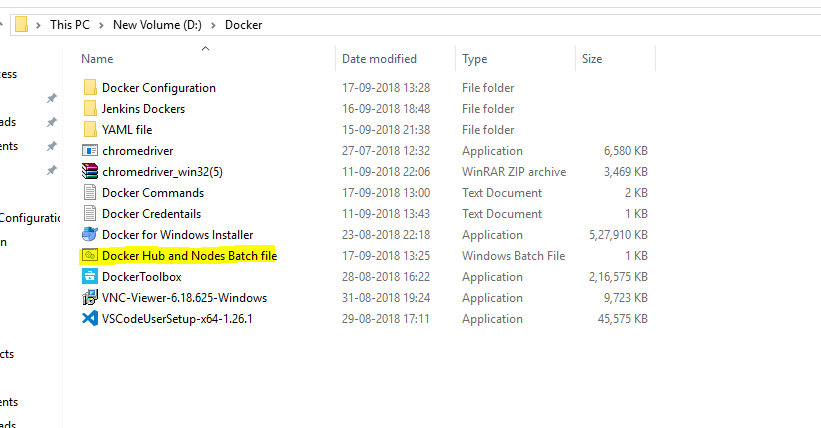
Another good way is to create a batch file consisting of all docker commands as follow:

Command 1 – Hub : docker run -d -p 4444:4444 --name selenium-hub -P selenium/hub

Command 2 – Firefox command: docker run -d -P --link selenium-hub:hub selenium/node-firefox-debug

Command 3 – Chrome command: docker run -d -p 5900:5900 --link selenium-hub:hub -v /dev/shm:/dev/shm selenium/node-chrome-debug

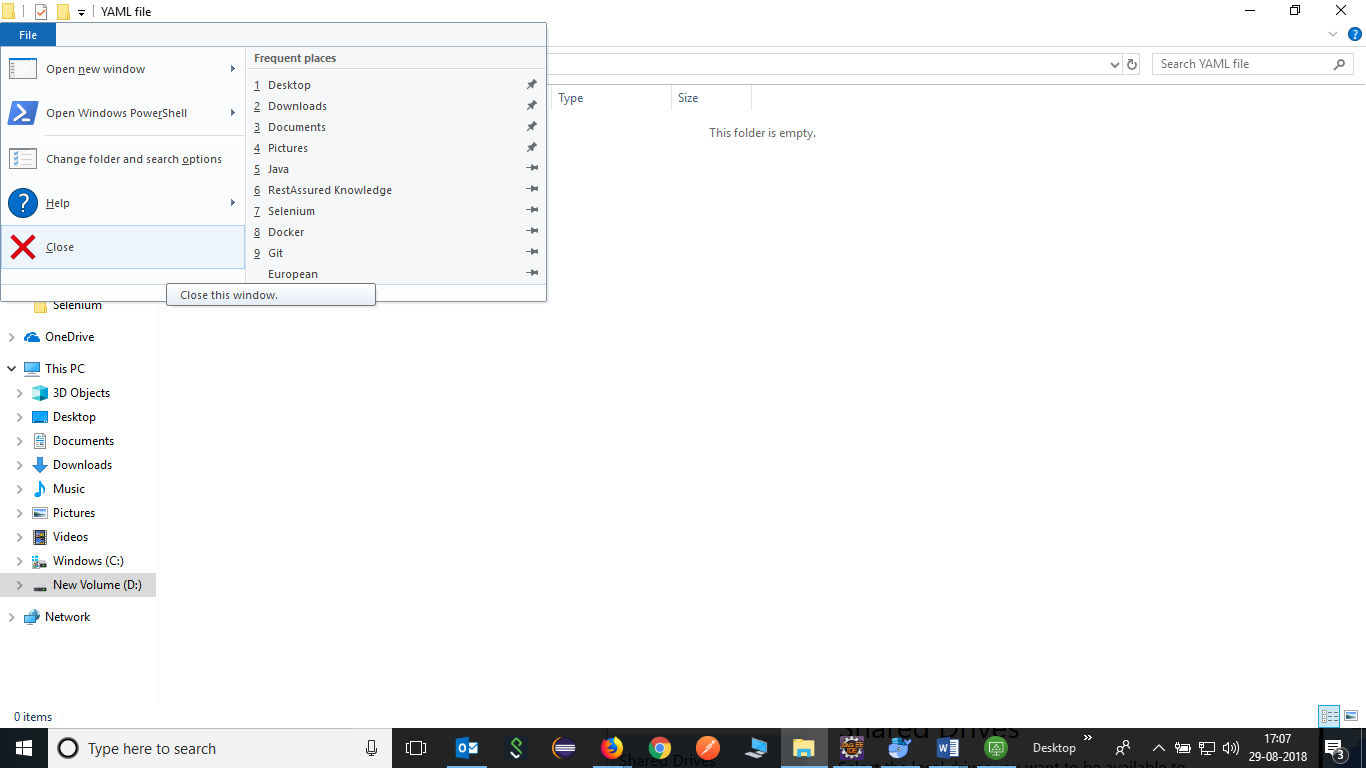
So , a file with the name ‘***Docker Hub and Nodes Batch file’*** in ‘D:\Docker’ location has been created. Just run it. After running it the hub and nodes will be configured automatically.



Step 7: Now create the yaml file. Below are the steps

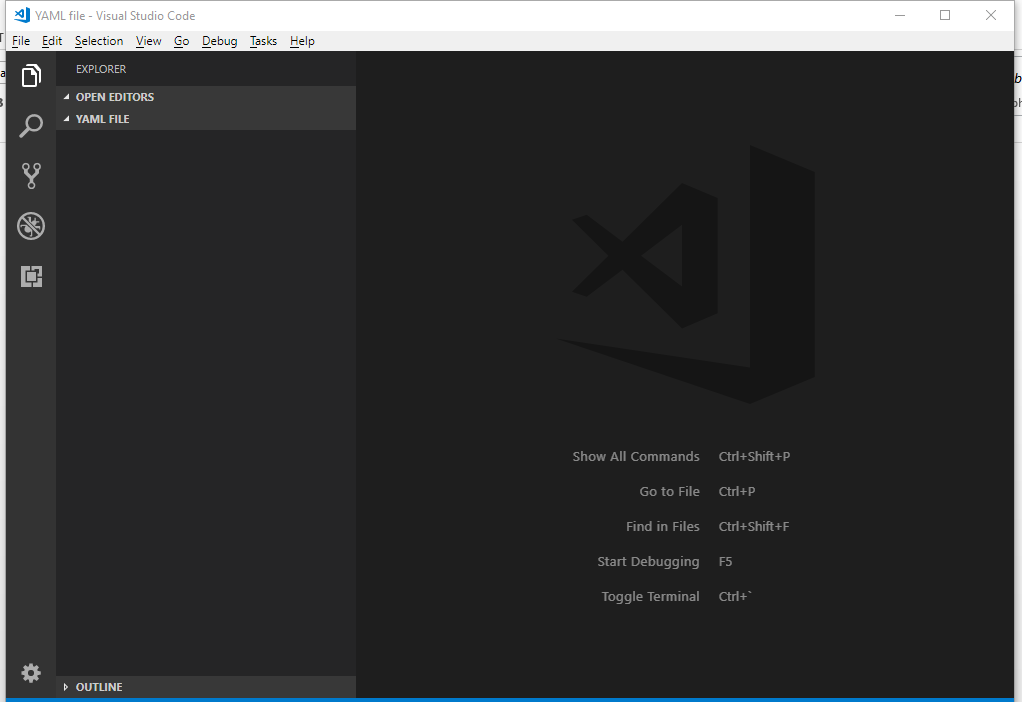
How to create yaml file (for docker composer)

Step 7.1: Create a folder and navigate to that folder and open the power shell or command window as described below

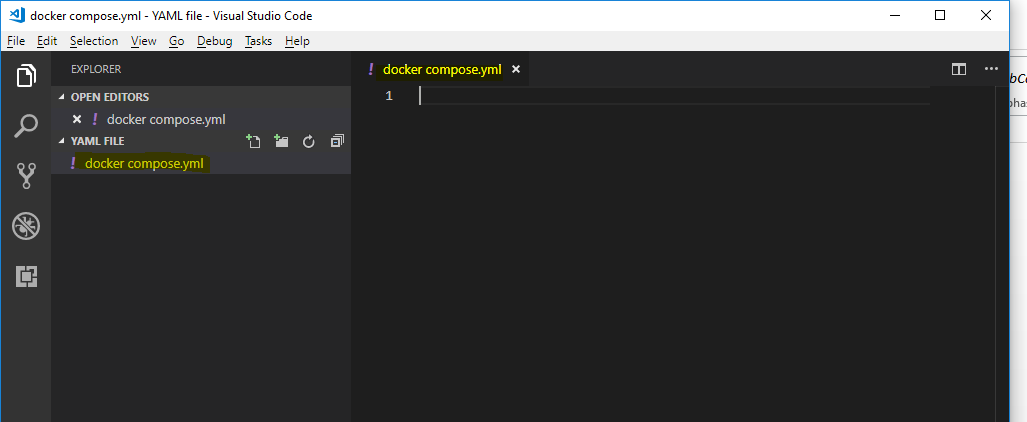


Step 7.2 : Enter code . and press enter and visual studio code window will open.

Note: Visual studio code needs to be installed on system.



Step 7.3: Now click on new file button and create the yaml file with .yml file extension.



Step 7.4

