

1. Job Script for Launching the Jupyter Notebook Job in the Cluster

```
[@login1 ~]$ cat jupyter.sh
#!/bin/bash
#PBS -N Notebook_job
#PBS -l select=1:ncpus=2:ngpus=1
#PBS -l walltime=24:00:00
#PBS -q serialq
#PBS -j oe

cd $PBS_O_WORKDIR

NOTEBOOK_LOGFILE=jupyterlog.out

# get tunneling info
node=$(hostname -s)
user=$(whoami)
cluster="AA.BB.CC.D"          # Enter cluster IP address
port=XXXX                    # Enter a port number like 9000
export JUPYTER_PORT=XXXX
### --After job submission open the port_forwarding.txt file for port forwarding details --###
echo -e "
Run the following command from your local machine terminal with local machine port YYYY:
$ ssh -N -f -L YYYY:${node}:${port} ${user}@${cluster}
" > port_forwarding.txt

module load anaconda/3
# launch the Jupyter Notebook run
jupyter notebook --no-browser --ip=${node} --port=${port} > ${NOTEBOOK_LOGFILE} 2>&1
```

2. Submitting the Job

```
[@login1 ~]$ qsub jupyter.sh
# A job id should be generated like 136407.cluster
```

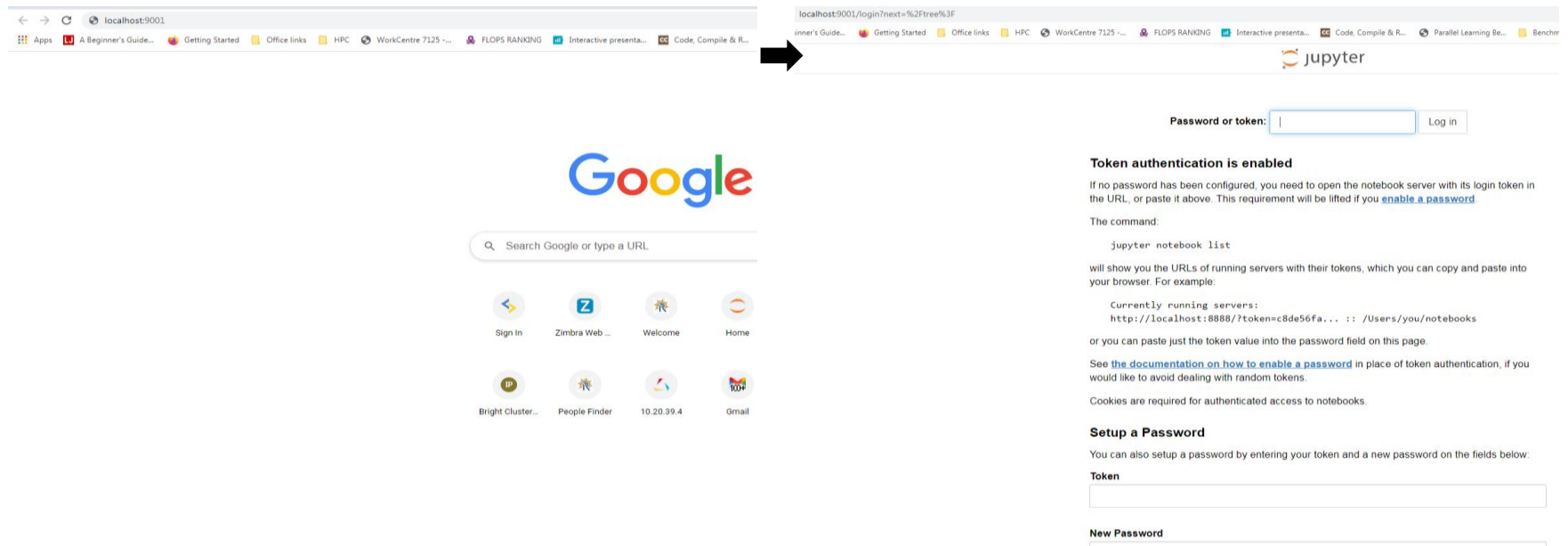
3. Port forwarding via ssh to local machine port no. YYYY

```
# Check the "port_forwarding.txt" file made after the job submission
# Check the compute node, port, username and cluster name in the file.
# You can copy paste the command from "port_forwarding.txt" file
# The command syntax will be like the following:
```

```
[@localhost]$ ssh -N -f -L YYYY:node:XXXX username@AA.BB.CC.D
```

XXXX: Cluster Port No. [like 8888 (default port) or 9000 or any other port number not in use in the cluster]
YYYY: Local Machine Port No. [like 8888 or 9000 or any other port number not in use in the local machine]
node: Compute node name [like cn001 or gn01 or any other compute node on which the job is submitted]
username: Your HPC Username [like ganesh]
AA.BB.CC.D: Cluster IP Address [like 10.20.40.8]

4. Open web browser in local machine and type http://localhost:YYYY



The screenshot shows a web browser window with the address bar set to 'localhost:9001'. The page content is split into two main sections. On the left, there is a Google search interface with the Google logo and a search bar. On the right, the JupyterLab interface is displayed. At the top of the JupyterLab interface, there is a 'Password or token:' field followed by a 'Log in' button. Below this, a message states 'Token authentication is enabled' and explains that if no password has been configured, the user needs to open the notebook server with its login token. It provides the command 'jupyter notebook list' and shows the output: 'Currently running servers: http://localhost:8888/?token=c8de56fa... :: /Users/you/notebooks'. It also mentions that cookies are required for authenticated access to notebooks. At the bottom, there is a 'Setup a Password' section with a 'You can also setup a password by entering your token and a new password on the fields below.' and two input fields labeled 'Token' and 'New Password'.

This will ask for the token which is available in the jupyterlog.out file in your working directory on the cluster.

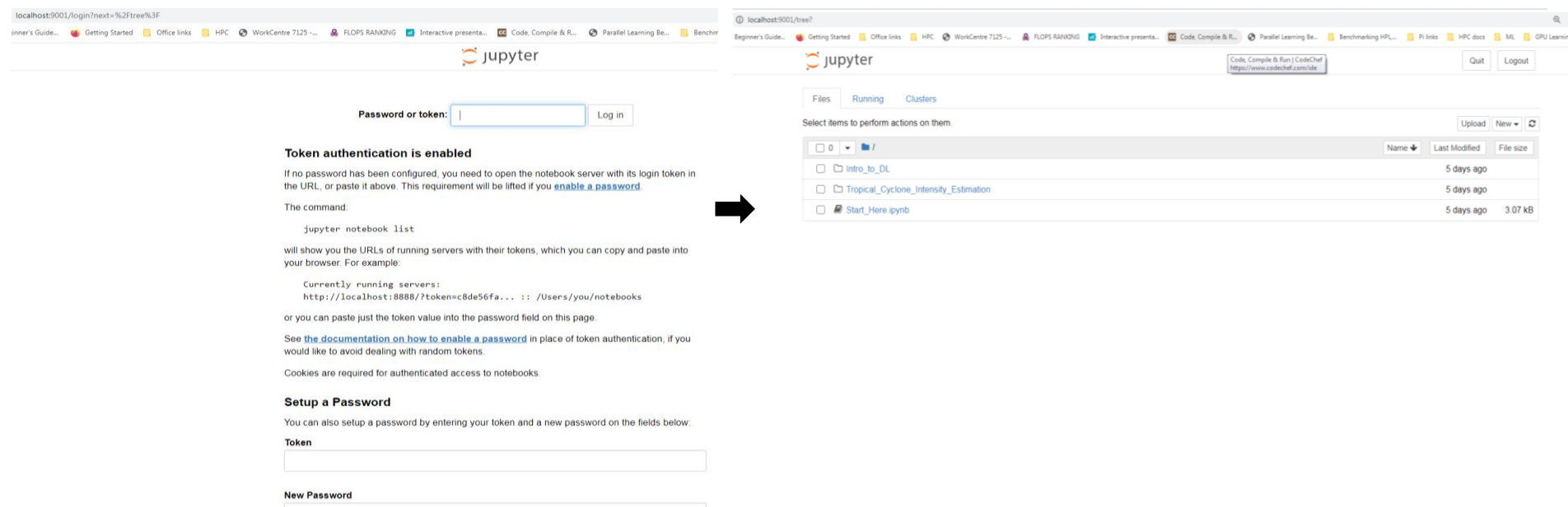
5. To get the token from the jupyterlog.out file in the cluster working directory

`[@login1 ~]$ tailf jupyterlog.out`

```
login1 ~]$ cat jupyterlog.out
04:21:11.007290: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library libcudart.so.10.2
[04:21:16.082 NotebookApp] jupyter_tensorboard extension loaded.
[04:21:16.083 NotebookApp] JupyterLab extension loaded from /usr/local/lib/python3.6/dist-packages/jupyterlab
[04:21:16.083 NotebookApp] JupyterLab application directory is /usr/local/share/jupyter/lab
[04:21:16.087 NotebookApp] [JupyterText Server Extension] NotebookApp.contents_manager_class is (a subclass of) jupyterlab.TextFileContentsManager already - OK
[04:21:16.087 NotebookApp] Serving notebooks from local directory: /workspace/python/jupyter_notebook
[04:21:16.087 NotebookApp] The Jupyter Notebook is running at:
[04:21:16.087 NotebookApp] http://hostname:9090/?token=30c6e3a7542268661f04aabc0b6e4eb101f9050ff29273
[04:21:16.087 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).

To access the notebook, open this file in a browser:
file:///home/deepakaga/.local/share/jupyter/runtime/nbserver-154462-open.html
Or copy and paste one of these URLs:
http://hostname:9090/?token=30c6e3a7542268661f04aabc0b6e4eb101f9050ff29273
[04:24:47.528 NotebookApp] 302 GET /tree? (172.21.3.104) 1.94ms
login1 ~]$
```

#Just copy and paste the token without the equal sign in the token window in browser.



The image shows a transition from a terminal window to a JupyterLab web interface. On the left, a terminal window displays the output of `tailf jupyterlog.out`, showing the Jupyter server's log. A specific line highlights the token: `http://hostname:9090/?token=30c6e3a7542268661f04aabc0b6e4eb101f9050ff29273`. An arrow points from this token to the JupyterLab interface on the right. The JupyterLab interface shows the 'Token authentication is enabled' message and a 'Password or token:' field. Below this, it lists 'Currently running servers' with the URL `http://localhost:8888/?token=c8de56fa...`. The 'Files' tab is active, showing a file browser with a table of files:

	Name	Last Modified	File size
<input type="checkbox"/>	/		
<input type="checkbox"/>	Intro_to_DL	5 days ago	
<input type="checkbox"/>	Tropical_Cyclone_Intensity_Estimation	5 days ago	
<input type="checkbox"/>	Start_Here.ipynb	5 days ago	3.07 kB