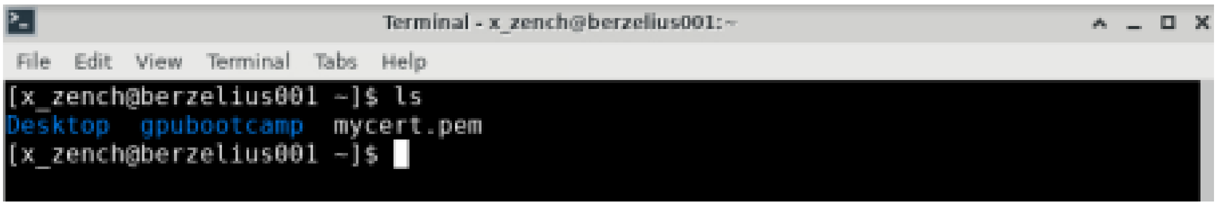
# ------------------- Step 1 : create certificate -----------

```
# Jupyter web server. Not _strictly_ necessary if you don't care for encrypted

# HTTP. Example:

username@laptop: $ openssl req -x509 -nodes -days 365 -newkey rsa:2048 \ -keyout mycert.pem
                                                              -out mycert.pem

Generating a RSA private key

...............................+++++

.................................................................+++++

writing new private key to 'mycert.pem'

-----

You are about to be asked to enter information that will be incorporated

into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN.

There are quite a few fields but you can leave some blank

For some fields there will be a default value,

If you enter '.', the field will be left blank.

-----

Country Name (2 letter code) [AU]:se

State or Province Name (full name) [Some-State]:Provincia

Locality Name (eg, city) []:Grönköping

Organization Name (eg, company) [Internet Widgits Pty Ltd]:Some University

Organizational Unit Name (eg, section) []:Dept. of Exceptionalism

Common Name (e.g. server FQDN or YOUR name) []:Example Name

Email Address []:example.name@someuniversity.se
```

# ------------------- Step 2 : copy the mycert.pem onto Berzelius compute node -----------

```
scp mycert.pem x_MyUserName@berzelius1.nsc.liu.se
```

double check you have the certificate copied onto Berzelius as below screenshot show



```
[x_zench@berzelius001 ~]$ ls
Desktop  gpubootcamp  mycert.pem
[x_zench@berzelius001 ~]$ 
```

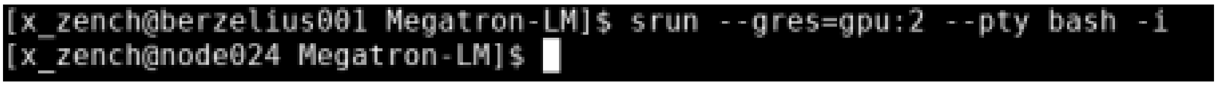
------------------- Step 3 : git clone the gpubootcamp repo ------------------------------- Open a terminal , git clone the gpubootcamp repo

```
## cd into your own user directory

cd /proj/megatron_bootcamp/users/$(id -un)

## clone the git repos we need

git clone https://github.com/gpuhackathons-org/gpubootcamp.git

git clone https://github.com/NVIDIA/Megatron-LM.git
```

------------------- Step 4 : get an interactive session -------------------------

```
srun --gres=gpu:2 --pty bash -i
```

Note: remember which node number you've been assigned, in this example it is node024



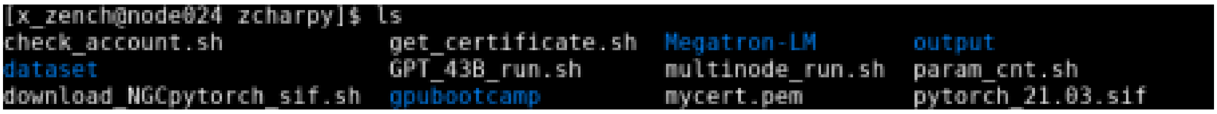
```
[x_zench@berzelius001 Megatron-LM]$ srun --gres=gpu:2 --pty bash -i
[x_zench@node024 Megatron-LM]$ █
```

------------------- Step 5: copy over pytorch_21.03.sif and other assets-------------------------

open another terminal, use below script to copy pytorch_21.03.sif and other assets

```
cd /proj/megatron_bootcamp/users/$(id -un) && tar xf
/proj/megatron_bootcamp/assets.tar
mkdir ./output/sv_gpt3_ckpt
mkdir ./profiles
```

Verify pytorch_21.03.sif exist under the directory of **/proj/megatron_bootcamp/users/$(id -un)**



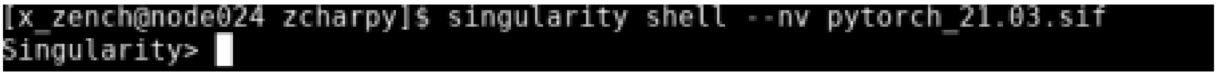
```
[x_zench@node024 zcharpy]$ ls
check_account.sh          get_certificate.sh  Megatron-LM       output
dataset                   GPT_43B_run.sh      multinode_run.sh  param_cnt.sh
download_NGCpytorch_sif.sh  gpubootcamp       mycert.pem        pytorch_21.03.sif
```

------------------- Step 6: run singularity with the downloaded pytorch_21.03.sif ----------------

```
export SINGULARITY_BINDPATH="/proj/megatron_bootcamp/users/$(id -un)"
singularity shell --nv pytorch_21.03.sif
```

Note1: making sure to specify the SINGULARITY_BINDPATH to ensure you are in the directory which you have read/write permission, on BerzeLiUs it is under the ="/proj/megatron_bootcamp/users/$(id -un)"

Note2: --nv is an important flag to add, otherwise nvidia-smi will not work

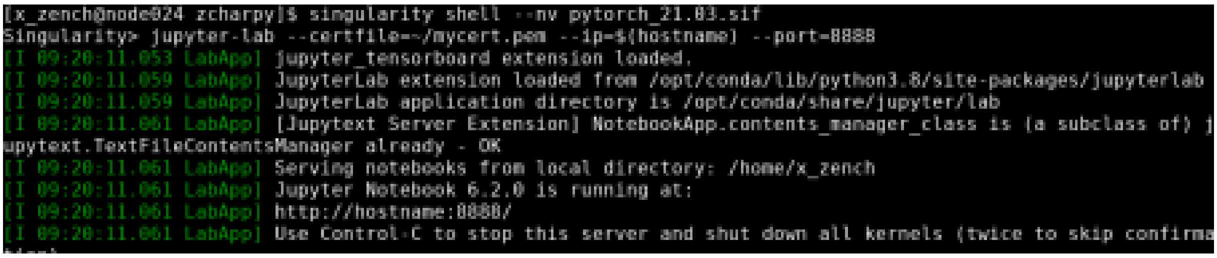


```
[x_zench@node024 zcharpy]$ singularity shell --nv pytorch_21.03.sif
Singularity> 
```

# ------------------ Step 7: call out jupyter lab and specify a port -----------------------------------

```
    jupyter-lab --certfile=~/mycert.pem --ip=$(hostname) --port=<YOUR_ASSIGNED_PORT>
```

Note: each user will be assigned a port, mine is 8888 , each port should be unique

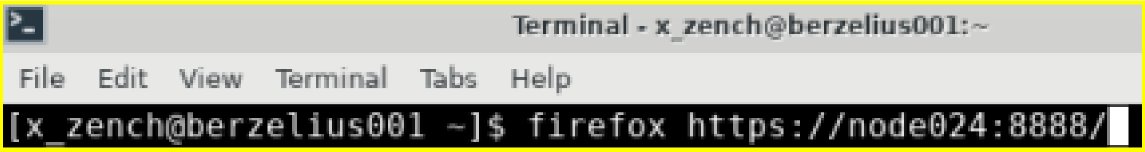


```
[x_zench@node024 zcharpy]$ singularity shell --nv pytorch_21.03.sif
Singularity> jupyter-lab --certfile=~/mycert.pem --ip=$(hostname) --port=8888
[I 09:20:11.053 LabApp] jupyter_tensorboard extension loaded.
[I 09:20:11.059 LabApp] JupyterLab extension loaded from /opt/conda/lib/python3.8/site-packages/jupyterlab
[I 09:20:11.059 LabApp] JupyterLab application directory is /opt/conda/share/jupyter/lab
[I 09:20:11.061 LabApp] [Jupytext Server Extension] NotebookApp.contents_manager_class is (a subclass of) j
upytext.TextFileContentsManager already - OK
[I 09:20:11.061 LabApp] Serving notebooks from local directory: /home/x_zench
[I 09:20:11.061 LabApp] Jupyter Notebook 6.2.0 is running at:
[I 09:20:11.061 LabApp] http://hostname:8888/
[I 09:20:11.061 LabApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirma
```

# ------------------ Step 8 : access the jupyter notebook via a browser ------------------------------

open another terminal , call out firefox , specifying the node number(here it is node041) and the port number (here it is 8888) , your node number and the port number should not be the same as the below.

```
   firefox https://node041:8888/
```

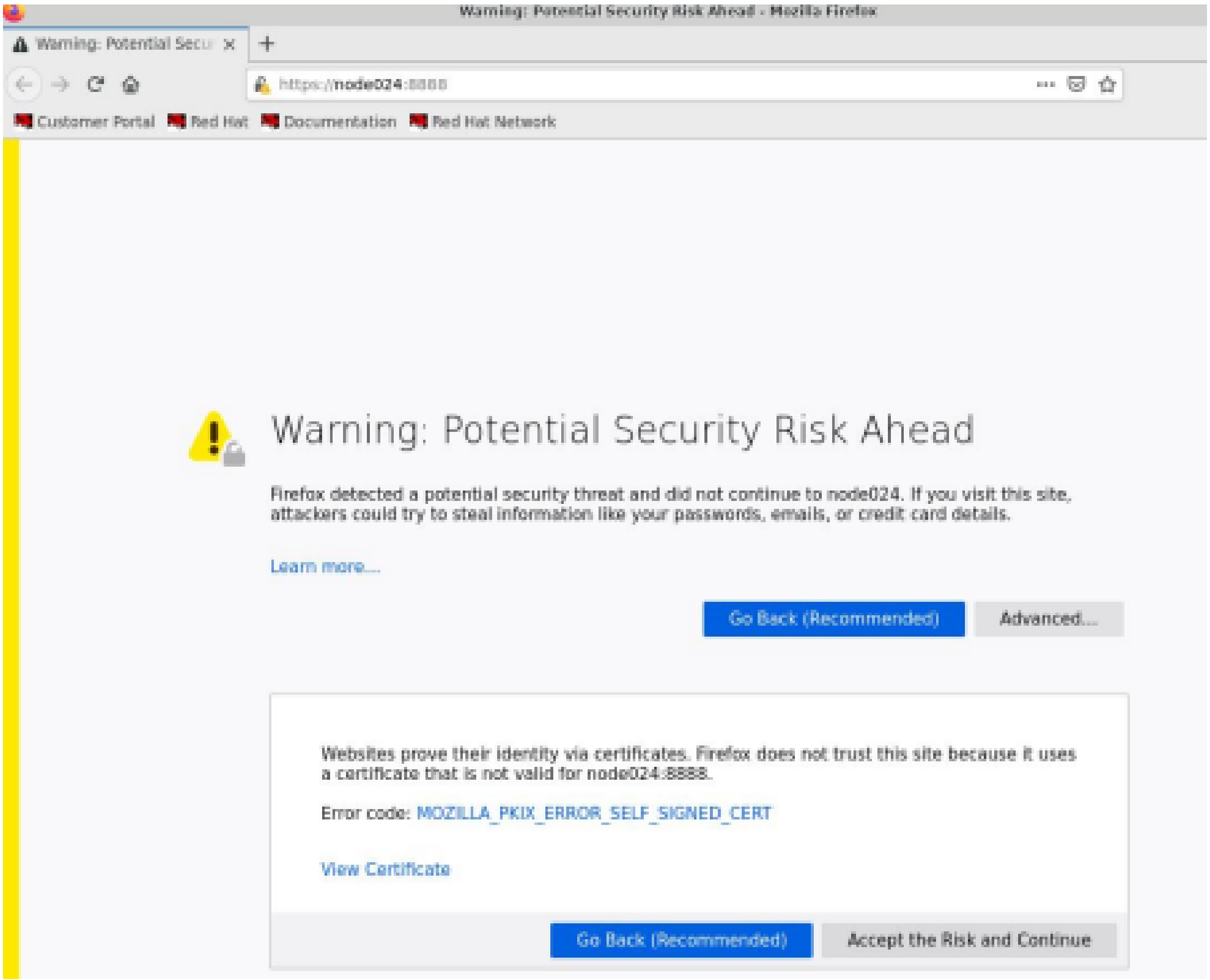


Terminal - x_zench@berzelius001:~
File Edit View Terminal Tabs Help

```
[x_zench@berzelius001 ~]$ firefox https://node024:8888/
```

This will automatically open up a browser

Note: if the browser complaints , simply click the advance button , then Accept the Risk and Continue



Warning: Potential Security Risk Ahead

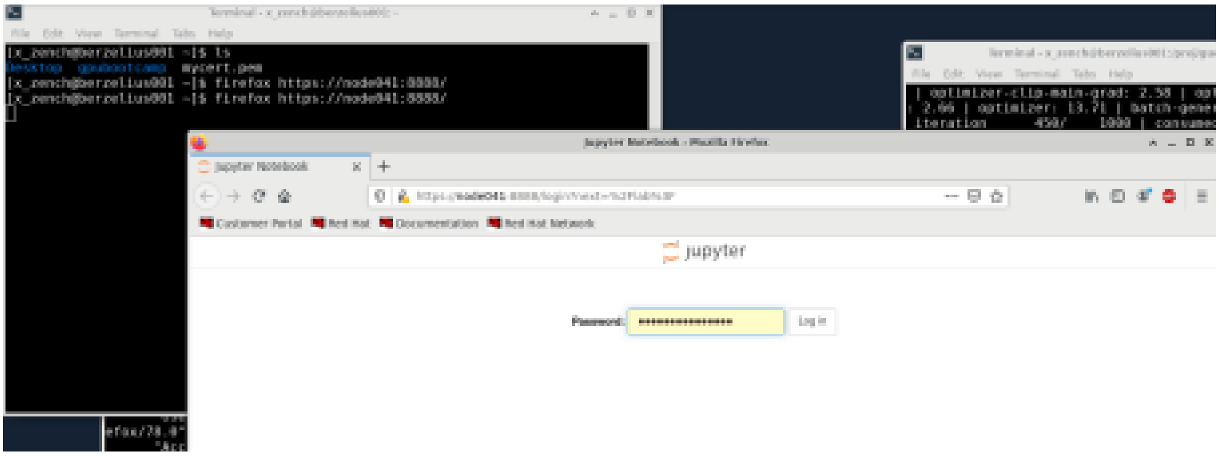
Firefox detected a potential security threat and did not continue to node024. If you visit this site, attackers could try to steal information like your passwords, emails, or credit card details.

Learn more…

Websites prove their identity via certificates. Firefox does not trust this site because it uses a certificate that is not valid for node024:8888.

Error code: MOZILLA_PKIX_ERROR_SELF_SIGNED_CERT

View Certificate

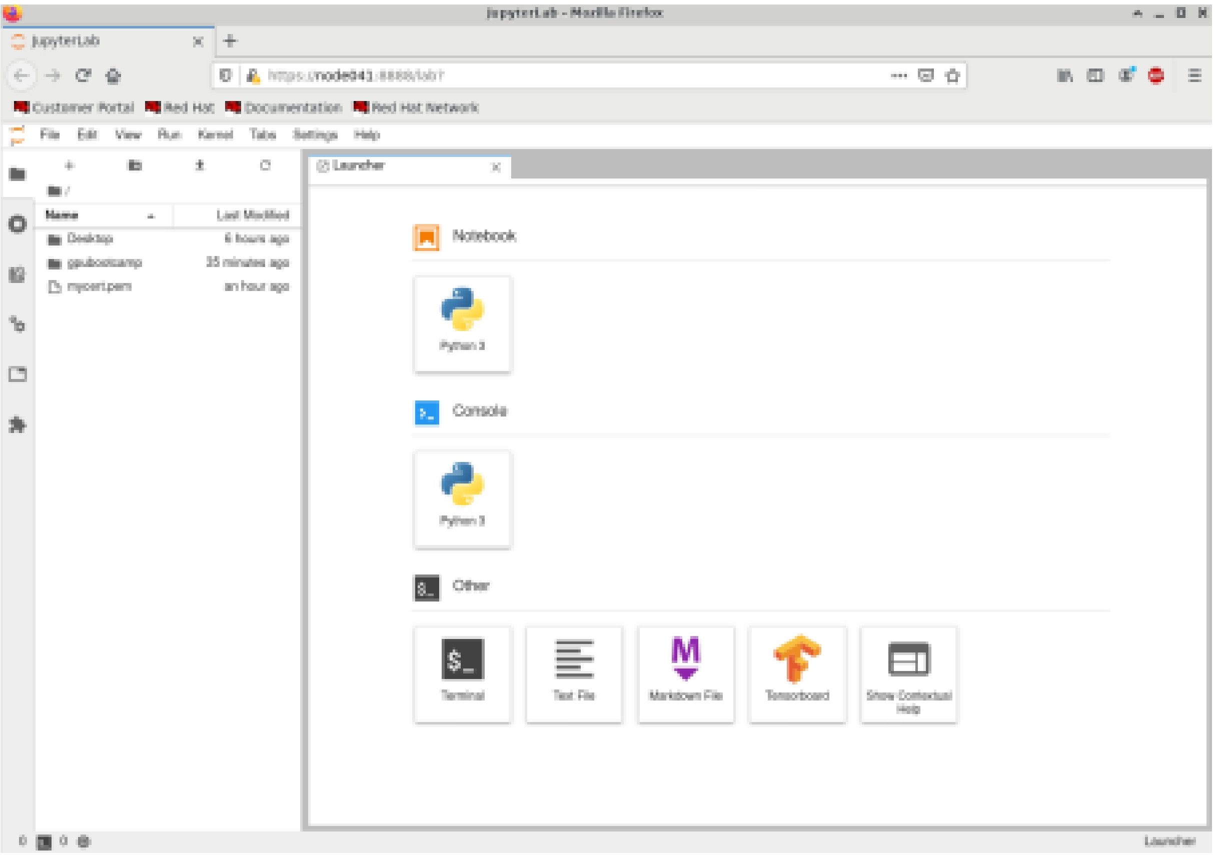


```
[x_zench@berzelius001 ~]$ ls
Desktop  gpubootcamp  mycert.pem
[x_zench@berzelius001 ~]$ firefox https://node041:8888/
[x_zench@berzelius001 ~]$ firefox https://node041:8888/
```

```
| optimizer-clip-main-grad: 2.58 | op
: 2.06 | optimizer: 13.71 | batch-gene
iteration      450/     1000 | consume
```

Password: ••••••••••••••••

# ----------step 9 : verify you can see the jupyter lab UI ---------------------------



# congratulations – you can now work on the tutorials !