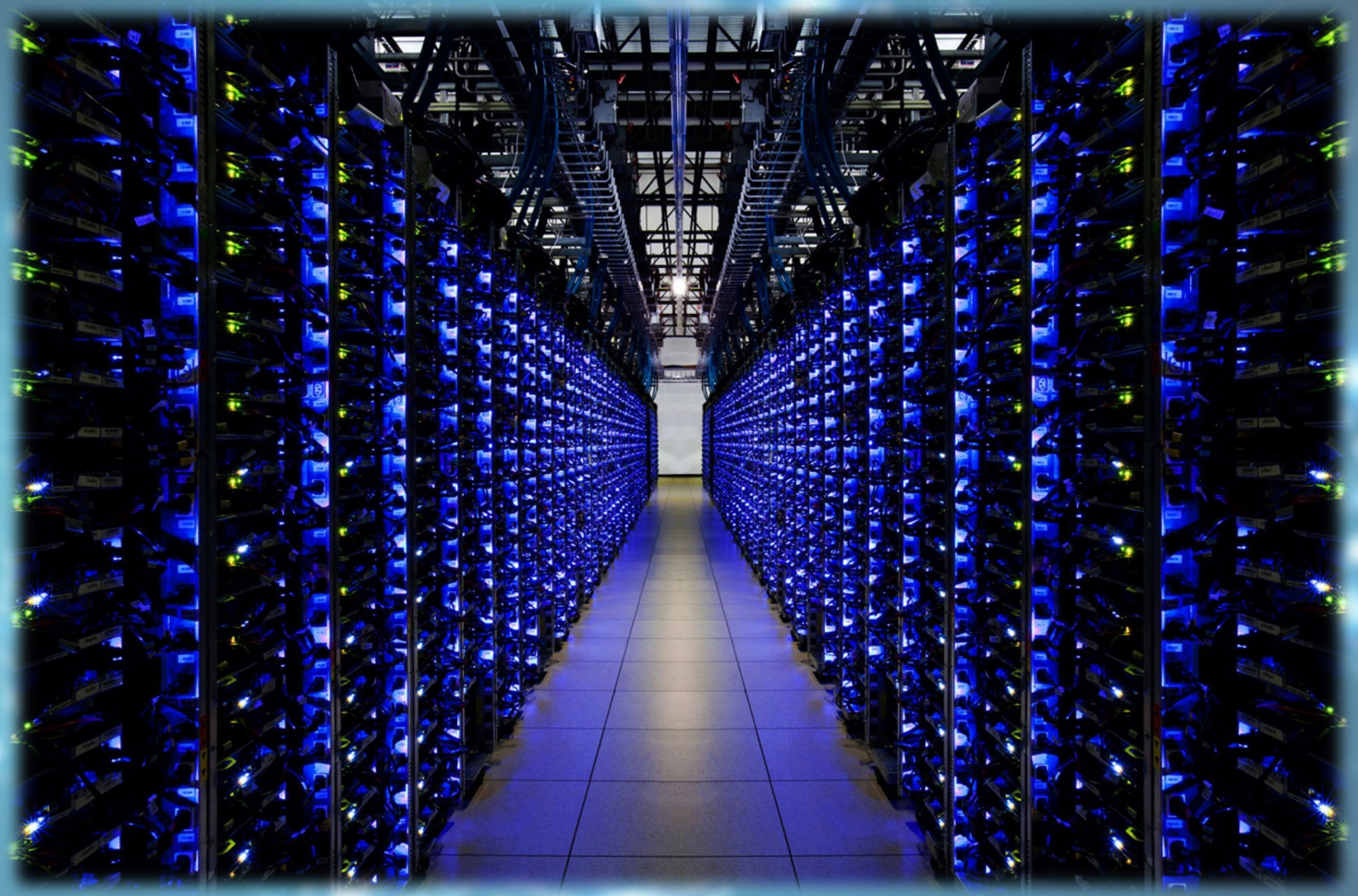
Introduction to HPC

Dimitrios Papadopoulos
Associate Proffesor, DTU Compute

Deep Learning



Training and deploying a deep learning model

GPU is a must!!!

Options:

- Your CPU in your local machine (technically yes, but you will finish the assignment on July ☺)
- Google Colab (a free GPU, but not really…)
- The expensive GPU of your gaming desktop (technically yes, but you work in groups at campus)
- A GPU cluster (DTU has HPC cluster and we have allocated about 15 high-end GPUs (cost about 1M DKK) for this course)

Let’s do a quick poll

(raise your hand if you agree)

(1) I am familiar with basic UNIX commands

*ls -l, pwd, cp, scp, rm -rf *,cd .., mkdir, sudo, cat, rsync, df, grep, vim, find, netstat*

(2) I know how to access a server via ssh

`ssh user@awesomeserver.dtu.dk`

(3) I have used colab or jupyter notebook

```
jupyter notebook --no-browser --port=9999
```

(4) I have used pip, conda and/or virtualenv

```
conda create --name notmyfirstconda
python=3.5
```

(5) I am familiar with running scripts on a cluster (batch or interactive) using a GPU

Steps

1) Terminal
2) Learn Basic UNIX commands
3) Access HPC and open interactive node
4) Copy something to HPC filesystem
5) Monitoring commands and filesystem
6) Loading modules
7) Run a script (+GPU)
8) Conda or virtual environment
9) Open jupyter (graphics)
10) Open jupyter and forward it

(1) Terminal

Unix-based: default terminal or iTerm

Windows: deault in new windows (?) or Putty, Kitty, windows terminal

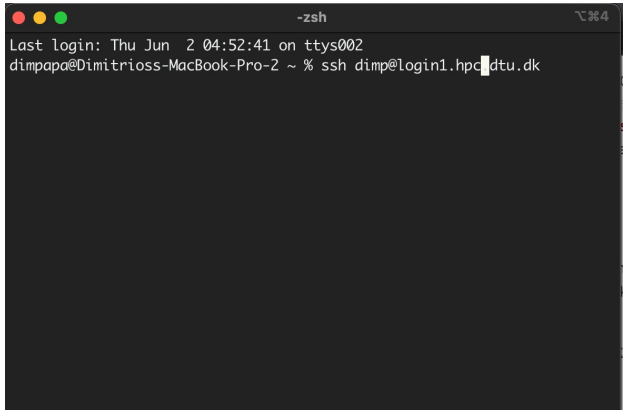
(2) Basic UNIX commands

- Navigating to the filesystem
- Copy-Move-Create
- df/ du
- top/ htop
- man/ help
- nvidia-smi/ gpu-stat
- quota
- Googling for the rest if needed

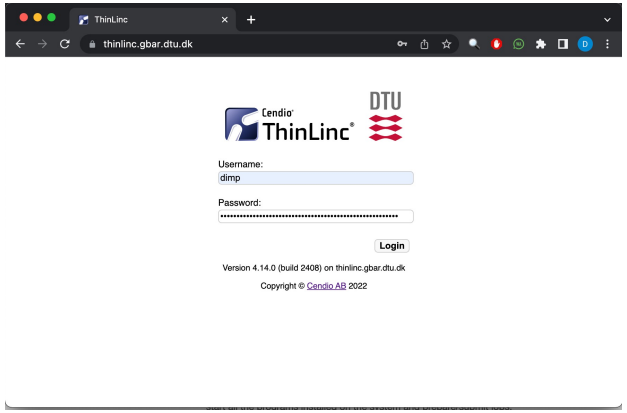
https://www.unixtutorial.org/basic-unix-commands

(3) Access HPC

- Everything about HPC: https://www.hpc.dtu.dk/
- Accessing the LSF 10 Cluster: https://www.hpc.dtu.dk/?page_id=2501



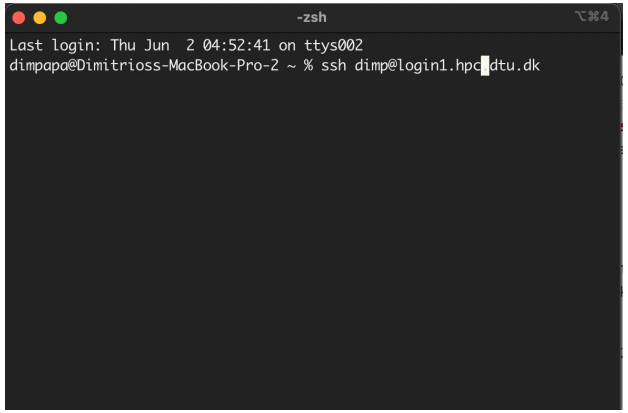
command line



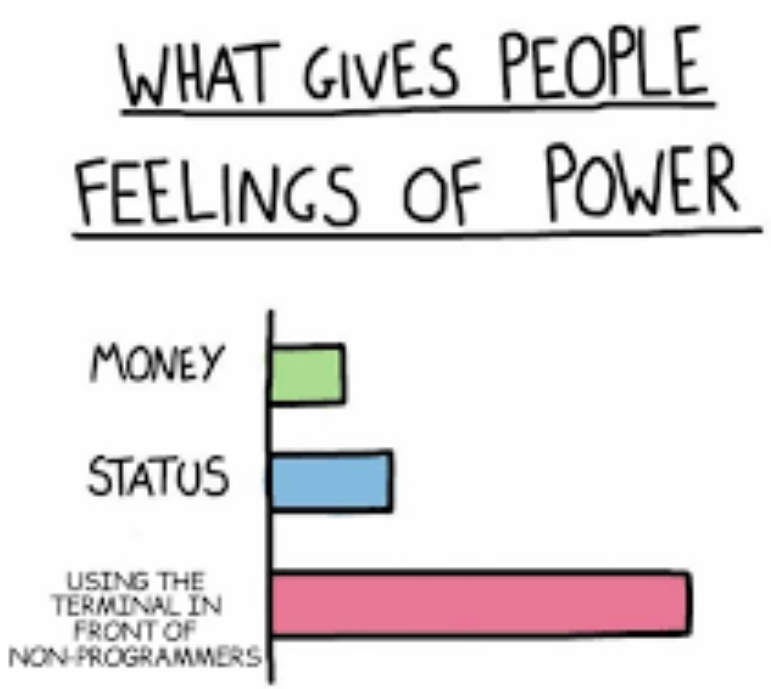
thinlinc graphics

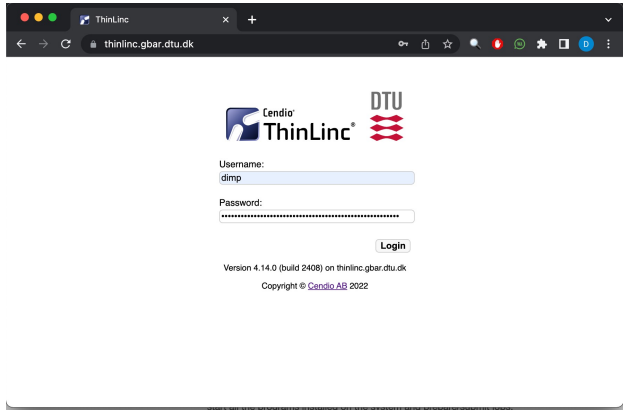
(3) Access HPC

- Everything about HPC: https://www.hpc.dtu.dk/
- Accessing the LSF 10 Cluster: https://www.hpc.dtu.dk/?page_id=2501



command line





thinlinc graphics

(3) Access HPC

ssh-keys and VPN: https://www.hpc.dtu.dk/?page_id=4317


```
When connecting from _outside_ of DTU (and not using DTU-VPN)
one is required to authenticate with a
ssh-key + ssh-key-keyphrase + DTU-password.

From inside of DTU (or connecting via DTU-VPN) one can also connect
with password or ssh-key.

https://www.hpc.dtu.dk/?page_id=4317

---
(For thinlinc-users: This is not an error-message, this is an
"information-message", please press "close" in the
right-bottom-corner to continue.)
---
dimp@login1.hpc.dtu.dk: Permission denied (publickey).
```

(3) Access HPC

ssh-keys and VPN: https://www.hpc.dtu.dk/?page_id=4317

SSH and ThinLinc login using ssh-keys

To increase the security of the DTU infrastructure, the whole DTU network setup is changing, and this affects also the way users can access G-bar and the DCC HPC clusters ***from outside the DTU network***.

Nothing changes, if you are accessing G-bar or the HPC cluster either on Campus, or via the DTU VPN. The instructions below are only relevant, if you need to access those resources from **outside** the DTU network, and the VPN is not a viable option, or it affects the performance critically (e.g. increased latency, incompatibility with a pre-existent pipeline, significant data transfer, etc).

If you only work on Campus or via the DTU VPN

Nothing changes for you. Just make sure that your VPN setup is up to date, and the Multi Factor Authentication is working.
You can safely skip the rest of the page.

If you work remotely, and always using the VPN is not a convenient option

SSH access will from now on require the use of ssh-keys + ssh-key passphrase + DTU password.
Instructions on how to setup your machine(s), and how to configure SSH for your G-bar/HPC account follow below.

Note: the guide below requires that you – during the setup process – are either connected to a DTU network on Campus, or via VPN (see DTU Inside for the VPN setup). Remember that the CISCO VPN requires the Multi Factor Authentication.

(3) Interactive nodes and Batch jobs

02514sh Is All You Need

Only 1 at a time →
if you can't open one→
kill the open session →
if not sure, *bkill XXXXX*

Reference: https://arxiv.org/abs/1706.03762

(4) Filesystem and copy

- **Monitoring:**

  `getquota_zhome.sh`

- **Local machine to cluster:**

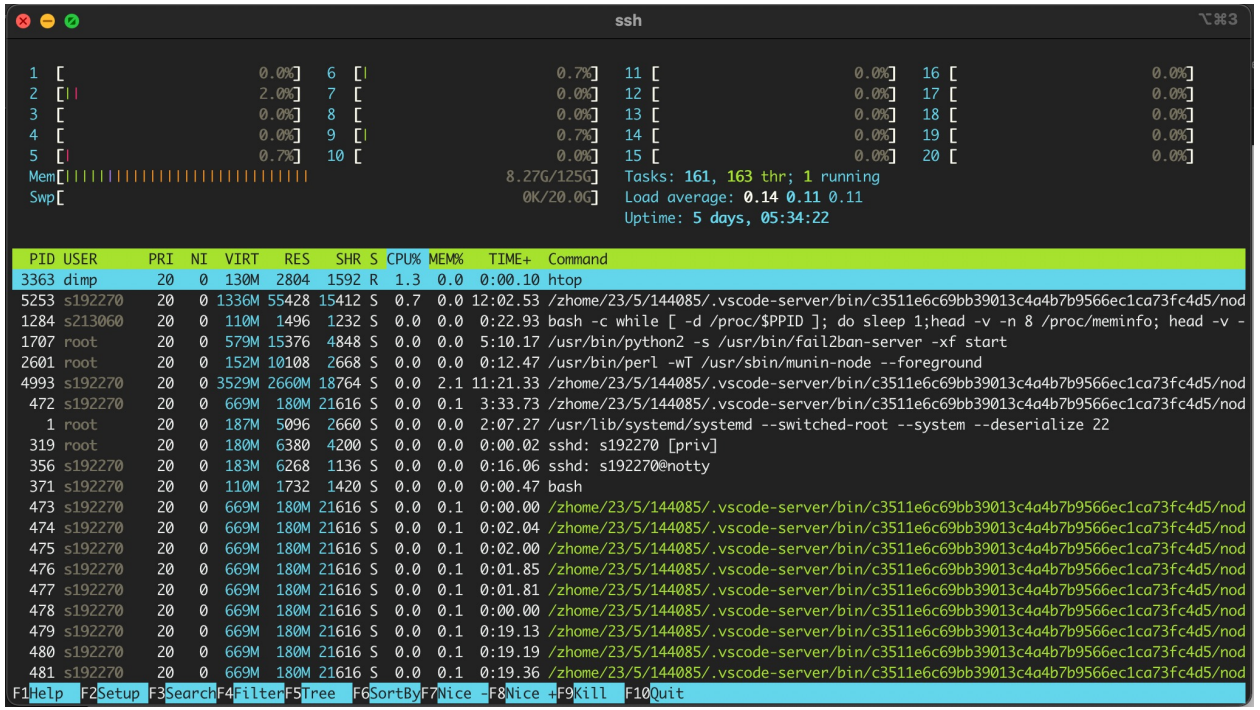
  ```
  scp randomimage.png
  dimp@login1.hpc.dtu.dk:/path/in/the/cluster/myfolder/bla/bla/
  ```

- **Shared space in cluster (read-only):**

  `/dtu/datasets1/02514/`

(5) Monitoring

htop



nvidia-smi

```
n-62-20-1(dimp) $ nvidia-smi
Thu Jun  2 05:13:02 2022
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 515.43.04    Driver Version: 515.43.04    CUDA Version: 11.7     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla V100-PCIE...  On   | 00000000:37:00.0 Off |                    0 |
| N/A   31C    P0    25W / 250W |      0MiB / 16384MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+
|   1  Tesla V100-PCIE...  On   | 00000000:AF:00.0 Off |                    0 |
| N/A   29C    P0    25W / 250W |      0MiB / 16384MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

getquota_zhome.sh

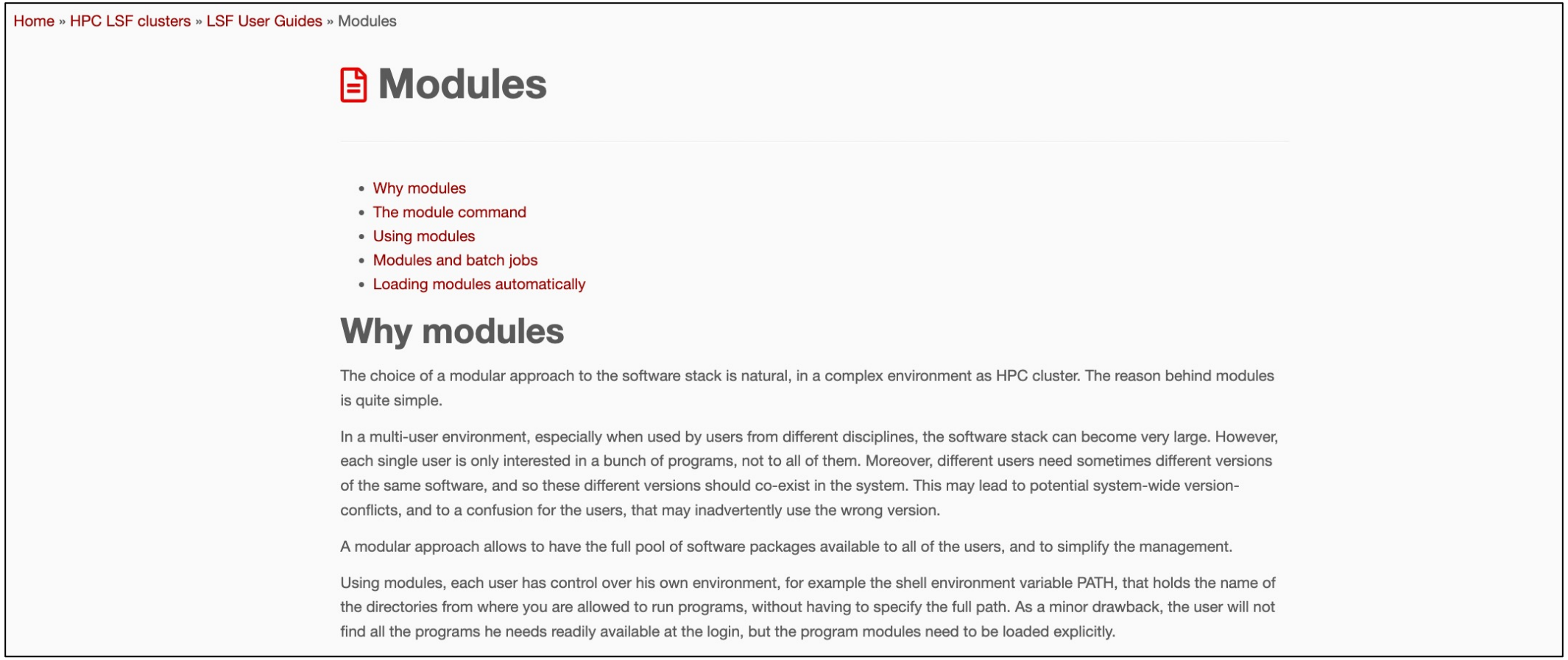
```
n-62-20-1(dimp) $ getquota_zhome.sh
You are using 817.51M of 30.00G quota

(Information updated every 30 minutes. Last Update: 2022-06-02 04:49:15)
```

Live demo Part 1!

(6) Modules

Modules: https://www.hpc.dtu.dk/?page_id=282



Home » HPC LSF clusters » LSF User Guides » Modules

Modules

- Why modules
- The module command
- Using modules
- Modules and batch jobs
- Loading modules automatically

Why modules

The choice of a modular approach to the software stack is natural, in a complex environment as HPC cluster. The reason behind modules is quite simple.

In a multi-user environment, especially when used by users from different disciplines, the software stack can become very large. However, each single user is only interested in a bunch of programs, not to all of them. Moreover, different users need sometimes different versions of the same software, and so these different versions should co-exist in the system. This may lead to potential system-wide version-conflicts, and to a confusion for the users, that may inadvertently use the wrong version.

A modular approach allows to have the full pool of software packages available to all of the users, and to simplify the management.

Using modules, each user has control over his own environment, for example the shell environment variable PATH, that holds the name of the directories from where you are allowed to run programs, without having to specify the full path. As a minor drawback, the user will not find all the programs he needs readily available at the login, but the program modules need to be loaded explicitly.

(7) Run a script

```
python i_can_run_hello_world_remotely.py
```

```
CUDA_VISIBLE_DEVICES=1 python i_can_even_run_it_on_a_gpu_that_i_select_wow.py
```

(8) Conda or Virtualenv

Virtual environments

Using virtual environments lets you have different stacks of software packages in different folders. Please search the web for additional information regarding virtual environments.

First, select the Python version you want, and load the corresponding module. Here we choose python 3.8.2. If you use versions older than 3.6, the instructions to create the virtual environment may differ.

```
module load python3/3.8.2
```

Then create a virtual environment. We choose the name `venv_1`

```
python3 -m venv venv_1
```

That's all. Now, activate the environment with the command

```
source venv_1/bin/activate
```

When the environment is active the shell notes this to the left of the cursor. It will look something like this:

```
(venv_1) ~
gbarlogin1$
```

This indicates the virtual environment name in parenthesis.

Note that *both* the commands `python` *and* `python3` point to the python version of your original module. Now, when the environment is active, you can install all the packages you need with the normal `python3 -m pip install ...` syntax, for example (note `--user` should **NOT** be added)

```
python -m pip install scikit-learn
```

All those packages will be installed locally under the directory

```
venv_1/lib/python3.8/site-packages/
```

When you have finished to use the environment, just deactivate it with the command

```
deactivate
```

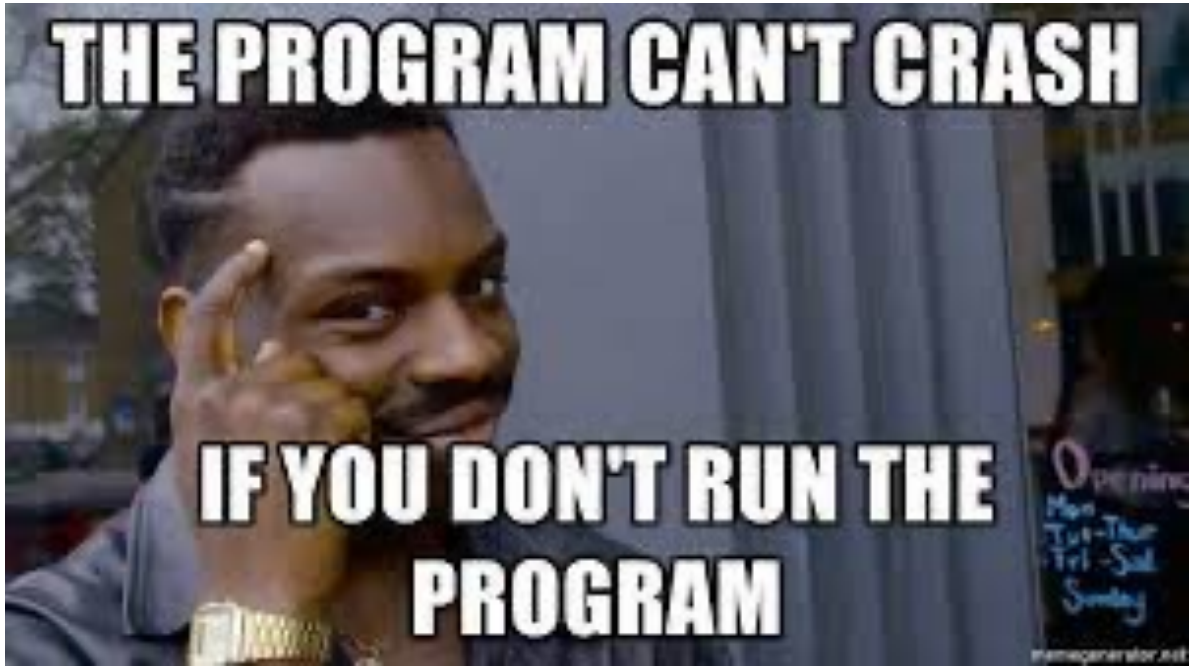
https://www.hpc.dtu.dk/?page_id=3678

(9) Jupyter

```
A) jupyter notebook –no-browser --port=XXXXX  --ip=$HOSTNAME

B) browser thinlinc: n00-00-0:XXXXX and token or password
```

It seems the easiest way. We don’t recommend it at all. It will create you several problem



(10) Jupyter and forward port to my machine

Very handy command to ssh tunneling (port forwarding)!!!

Terminal 1 (local):

(1) ssh to hpc: ssh **USER@login1.hpc.dtu.dk**

(2) open interactive node: **02514h**

(3) run jupyter: **jupyter notebook --no-browser --port=XXXXX --ip=$HOSTNAME**

Terminal 2 (local):

(1) port forwarding: **ssh USER@login1.hpc.dtu.dk -g -LXXXXX:n-00-00-00:XXXXX –N**

Browser (local):

(1) copy paste command from Terminal 1:
http://127.0.0.1:XXXXX/?token=xyxyxyxyxyxyxyxyxyxyxyxyyxyxyxyxyxxyx

Live demo Part 2!

Bonus 1!!

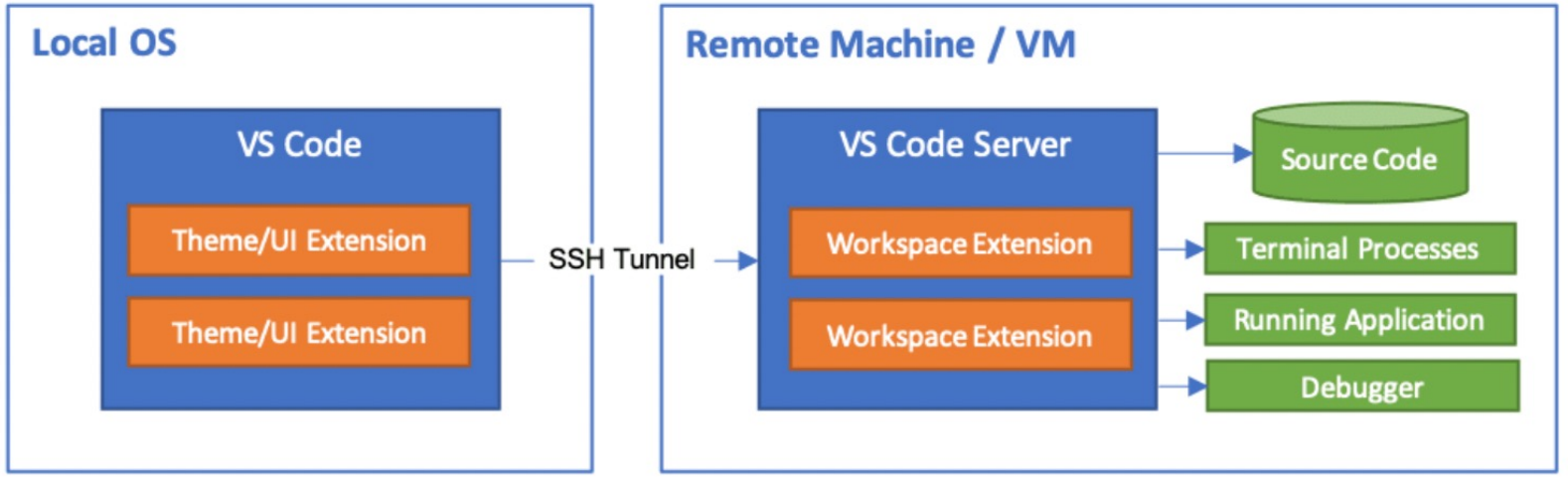
Scripts / Classes/ Functions >>> Notebooks

BUT

Editors>>>vim

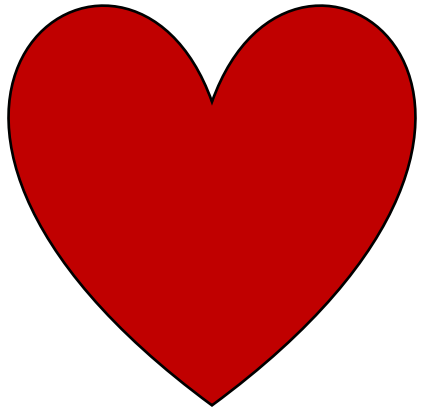
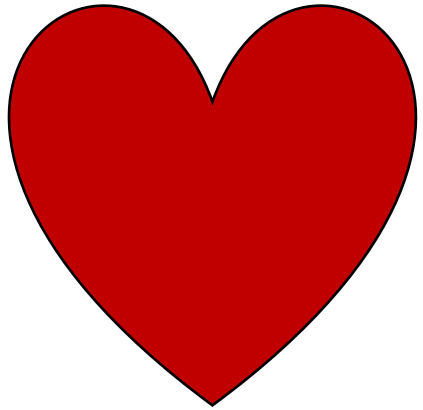
Vscode and Vscode with remote ssh:

https://code.visualstudio.com/docs/remote/ssh



Bonus 2!!

“My terminal is awesome!”

 **tmux** 

Questions???

Exercise 1.1

11.00-12.00

Good Luck!