



DTU-HPC Anaconda tutorial

Getting conda environments to run on the DTU-HPC is simple but there are a few pitfalls to be aware of and a few tips that the senior users of the system can pass on. Find them in this tutorial.

1 Ensure sufficient space to run and analyze data

For long-term usage it is important to make sure one has all the space one needs to run, and then analyze the output.

Start by making sure to get allocated a so-called work directory (also called scratch as it is not backed up to the same extent that the home dir is).

E-mail support@hpc.dtu.dk, and ask for a work folder, something like 1 TB is not unusual. It is good to keep in mind that the home dir is 30 GB which can fill up quite quickly.

2 Connect to the DTU-HPC-cluster

Having made sure that storage space won't be an issue, the next step is to connect to the DTU-HPC-cluster. There are many options to do this:

- **ssh directly**
If one is comfortable with working in a terminal—powershell, BASH and such—one can directly ssh into it via login1.gbar.dtu.dk, using the DTU username and password. See other options and the official guide here: https://www.hpc.dtu.dk/?page_id=2501
- **Using an interface**
Many would rather use an interface that does this in the background and also transfers files easily, for example:
 - For those that already use an IDE (integrated development environment) like Visual Studio Code or the like, this would be the best choice.
 - For others that are perhaps not looking for all the benefits of an IDE and would prefer a simpler interface, the program MobaXterm is recommended: <https://mobaxterm.mobatek.net/>
 - Another good option is: <https://www.putty.org/>

3 Install conda on the cluster

Having acquired a good set up, the next step is to install (Ana/mini) conda on the cluster. Note that you want to install it in the work dir that you got rather than your home dir as the envs you install will also take space in this folder and with a few envs you can quickly get in to space problems.



If you are unsure which of these you would need, miniconda vs. Anaconda, choose the Anaconda installation as this is more complete and versatile (but takes more space):

<https://docs.conda.io/projects/conda/en/latest/user-guide/install/linux.html>

Take note of where you install your conda as you need this path to use conda in the queuing system.

4 Installing conda environments

After this installation, one can conda install envs directly on the cluster or if that becomes problematic you have the option of simply creating an env yourself, uploading the env file and creating the env on the cluster that way. If you already have the environment in question activated, the command would be something like this:

```
conda env export > base.yml
```

This example uses base as the name of the file but this can of course be anything you like. This file can then be uploaded to the cluster. Note that this method just lists the dependencies used and the channels involved and so this is mostly a good way to speed up making the env but not necessarily an explicit copy of the setup you used on your hardware.

If you find the need to be more thorough, you can make an explicit list of dependencies used: `conda list --explicit > base.txt`

Demonstration of transferring files

A nice demo on transferring files with MobaXterm can be seen here (transfer starts being mentioned around the 60% mark):

<https://mobaxterm.mobatek.net/demo.html>

The file is then used to create an env with:

```
conda env create -f base.yml
```

Or if you want to use the explicit text file:

```
conda create --name base --file base.txt
```

This leads to using conda on the cluster. One option would be to activate the env you want to work with and running your computation on the login node, which can be very unpopular with computationally heavy stuff but should be fine for quick testing and troubleshooting. The better option is to submit the computation to the cluster via the queuing system which is used via the command `bsub` and a script for that command, e.g. `job.sh`.

Examples of job.sh scripts and the usage of them

- The short version: https://www.hpc.dtu.dk/?page_id=2759
- The long and detailed version: https://www.hpc.dtu.dk/?page_id=1416



5 Enabling the HPC to use your conda

The important difference in using conda on the cluster's node/s is making the node/s which receive the job (computation/calculation) aware of your conda, which is achieved by including the location which you put anaconda in the job script, that is, include these lines in the job script:

```
source [your_conda_install_path]/anaconda3/bin/activate  
conda activate base
```

6 Submitting the job

Put the command you usually use in the job script and submit the job.