# User Software Installation

Tips and examples on how to install software locally


- Allowed directories for users software installations
- Install Miniconda
- Conda
- Example of a user application installation
- User customization with module

# Allowed directories for users software installations

There are a few options for local users installation locations as showed on the next table, create appropriate paths bellow the base directory.

| Site | Base Directory | Comments |
|---|---|---|
| **INCD-Lisbon** | /home/GROUP/USERNAME | |
| **INCD-Lisbon** | /data/GROUP/USERNAME | |
| **INCD-Lisbon** | /data/GROUP | *available on request* and a responsible must be appointed |
| **INCD-Lisbon** | /exper-sw | legacy location, to be moved whenever possible |
| **INCD-Minho** | /home/GROUP/USERNAME | |
| **ISEC-COIMBRA** | ***none so far*** | |

- GROUP: User group name
- USERNAME: User login name

> **NOTE** Some applications may have dependencies requiring cluster wide installation of packages, please contact the **INCD support helpdesk** on those cases.

# Install Miniconda

Small tutorial on how to install and run miniconda on the HPC clusters.

## **1.** Login into your login machine

## **2.** Download Miniconda into your local home

```
$ wget https://repo.anaconda.com/miniconda/Miniconda2-latest-Linux-x86_64.sh
```

## **3.** Execute Miniconda

```
$ chmod +x Miniconda2-latest-Linux-x86_64.sh (give execution permission fo file)

$ ./Miniconda2-latest-Linux-x86_64.sh

Welcome to Miniconda2 4.6.14

In order to continue the installation process, please review the license
agreement.
Please, press ENTER to continue

....

Do you wish the installer to initialize Miniconda2
by running conda init? [yes|no]
[no] >>> no

...
```

## **4.** Run miniconda

```
$ ./miniconda2/bin/conda
```

## **5.** Load conda environment

```
. /home/csys/jpina/miniconda2/etc/profile.d/conda.sh
```

## **6.** Load conda environment in your submission script

```
$ cat test_submit.sh

#!/bin/bash
# Load user environment during job excution
#$ -V

# Call parallell environment "mp", and excute in 4 cores
#$ -pe mp 4

# Queue selection
#$ -q hpc

# Load miniconda
.  /home/csys/jpina/miniconda2/etc/profile.d/conda.sh
```

> **NOTE** Loading the conda environment can lead to conflits with the 'module load' command, therefore users should test the environment on a running job when using both conda and modules environments. If possible, use only conda environment.

# Conda

Another example of **conda** environment setup.

## Login on the submit node

Login on the cluster submission node, check the page *How to Access* for more information:

```
$ ssh -l <username> cirrus.ncg.ingrid.pt
[username@cirrus ~]$ _
```

## Prepare a conda virtual environment

The default **python** version for *CentOS 7.x* is ***2.7.5*** which is not suitable for many applications. So, we will create a **python** virtual environment:

```
[username@cirrus ~]$ conda create -n myconda python=3.6
[username@cirrus ~]$ conda activate myconda
```

On the first command, where we create the **conda** virtual environment, you can include a list of applications to include on your environmnet, for example:

```
[username@cirrus ~]$ conda create -n myconda python=3.6 ipython-notebook numpy=1.6
```

## Manage the conda virtual environment

It is possible to include additional packages to your **conda** environment, for example:

```
[username@cirrus ~]$ conda activate myconda
[username@cirrus ~]$ conda install numpy
```

You can update your software bandle on the **conda** virtual environment with the command:

```
[username@cirrus ~]$ conda update [scipy ...]
```

or remove a specific application:

```
[username@cirrus ~]$ conda uninstall tensorflow-gpu
```

Check the **conda** help for more information:

```
[username@cirrus ~]$ conda help
[username@cirrus ~]$ conda install --help
```

## Manage the conda packages list with pip

It is possible to complemment the **conda** virtual environment packages list with **pip**. For example:

```
[username@cirrus ~]$ conda activate myconda
[username@cirrus ~]$ pip install --upgrade pip
[username@cirrus ~]$ pip install --upgrade setuptools
[username@cirrus ~]$ pip install tensorflow-gpu
[username@cirrus ~]$ pip install keras
```

## Manage packages versions

If the applications available on **conda** virtual environment do not match your version requirements you may need to use packages from **pip** repostory; check the availability of ***conda search*** and ***pip search*** command line interfaces.

As an example we have the **tensorflow-gpu** package, when used with **keras**, the **conda** repository downgrades ***tensorflow-gpu** to version *1.15*, but you most likely will prefer version *2.0* . The **pip** repository has the right combination of **tensorflow-gpu** and **keras** packages.

***We advise the user to install a package from only one repository in order to guarantee perfect behaviour.***

## Load conda environment on a batch job

Create a submit script:

```
[username@cirrus ~]$ cat submit.sh

#!/bin/bash

#SBATCH --job-name=MyFirstSlurmJob
#SBATCH --time=0:10:0
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=16

# Be sure to request the correct partition to avoid the job to be held in the queue, furthermore
#□on CIRRUS-B (Minho)  choose for example HPC_4_Days
#□on CIRRUS-A (Lisbon) choose for example hpc
#SBATCH --partition=HPC_4_Days
```

```
# check python version
python --version

# Load conda environment
conda activate myconda

# recheck python version
python --version

# your job payload
#....
```

Submit:

```
[username@cirrus ~]$ sbatch submit.sh
Your job 2037792 ("submit.sh") has been submitted
```

After completion:

```
[username@hpc7 ~]$ ls -l
-rwxr-----+ 1 username hpc    668 Jan  7 12:19 submit.sh
-rw-r-----+ 1 username hpc     44 Jan  7 12:18 submit.sh.e2037792
-rw-r-----+ 1 username hpc      0 Jan  7 12:18 submit.sh.o2037792

[username@cirrus ~]$ cat submit.sh.e2037792
Python 2.7.5
Python 3.6.9 :: Anaconda, Inc.
```

## Some References

- GitHub OS-agnostic
- Anaconda Documentation
- Conda Documentation

# Example of a user application installation

This example will show how to install **Octave** on the hipotetical user *username* home directory **HOME/soft/octave/5.1.0**. The example will use the interactive host to install the software but if you can also write a script and submit a job as long the command line instructions could be automatized.

## Login on the submit node

Login on the cluster submition node, check the page *How to Access* for more information:

```
$ ssh -l <username> hpc7.ncg.ingrid.pt
[username@hpc7 ~]$ _
```

## Download the source code

```
[username@hpc7 ~]$ wget ftp://ftp.gnu.org/gnu/octave/octave-5.1.0.tar.xz
[username@hpc7 ~]$ tar Jxf octave-5.1.0.tar.xz
[username@hpc7 ~]$ cd octave-5.1.0
```

## Configure and install

```
[username@hpc7 ~]$ ./configure --prefix=/home/mygroup/username/soft/octave/5.1.0 --enable-shared
[username@hpc7 ~]$ make
[username@hpc7 ~]$ make check
[username@hpc7 ~]$ make install
```

## Setup environment

The most basic would be to configure the appropriate environment variables, or better, create a shell script to load when needed:

```
[username@hpc7 ~]$ cat .octave.bash
export OCTAVE_HOME=/home/mygroup/username/soft/octave/5.1.0
[ -z "$PATH" ] && export PATH=$OCTAVE_HOME/bin || export PATH=$OCTAVE_HOME/bin:$PATH
[ -z "$CPATH" ] && export PATH=$OCTAVE_HOME/include || export CPATH=$OCTAVE_HOME/include:$CPATH
```

```
[ -z "$LD_LIBRARY_PATH" ] && export LD_LIBRARY_PATH=$OCTAVE_HOME/lib || export
LD_LIBRARY_PATH=$OCTAVE_HOME/lib:$LD_LIBRARY_PATH
[ -z "$PKG_CONFIG_PATH" ] && export PAG_CONFIG_PATH=$OCTAVE_HOME/lib/pkgconfig || export
PKG_CONFIG__PATH=$OCTAVE_HOME/lib/pkgconfig:$PKG_CONFIG_PATH
```

## Activate environment for application

```
[username@hpc7 ~]$ . .octave.bash
```

## Run the application

```
[username@hpc7 ~]$ which octave
~/soft/octave/5.1.0/bin/octave

[username@hpc7 ~]$ octave
octave:1> _
```

## Better way to setup environment: USE MODULES

A better way to provide configuration would be using the *module environment tool" customized for the user, check the User Customization With module page. It would be easier to manage and share with other users if needed.

# User customization with module

Example of environment configuration for *Octave* application installed on the example ***Example of a User application Installation*** page.

## Login on the submit node

Login on the cluster submition node, check the page *How to Access* for more information:

```
$ ssh -l <username> hpc7.ncg.ingrid.pt
[username@hpc7 ~]$ _
```

## Select a directory to store your modules environments

In this example we will use **~/.module** on the user *Home* directory:

```
[username@hpc7 ~]$ mkdir .module
```

## Create a modules resource file

Create a file named ***~/.modulerc*** with the following content:

```
[username@hpc7 ~]$ cat .modulerc
#%Module1.0#####################################################################
##########
##
## User prefer modules at session startup
##
module use $env(HOME)/.module
```

## Create a configuration file for Octave application

Lets create a simple configution file for the **Octave** application installed on the *Home* directory:

```
[username@hpc7 ~]$ mkdir .module/octave
[username@hpc7 ~]$ cat .module/octave/5.1.0
#%Module1.0
##
## octave/5.1.0 modulefile
##

proc ModulesHelp { } {
        global version
        puts stderr "\tSets up Octave"
        puts stderr "\n\tVersion $version\n"
}

module-whatis "Sets up Octave"

set version 5.1.0
set modroot /home/hpc/jmartins/soft/octave/$version

setenv OCTAVE_HOME $modroot
prepend-path PATH $modroot/bin
prepend-path CPATH $modroot/include
prepend-path LD_LIBRARY_PATH $modroot/lib
prepend-path PKG_CONFIG_PATH $modroot/lib/pkgconfig
```

## Check the new modules environment

The next time you login on server you will find your environment profile available for normal usage:

```
[username@hpc7 ~]$ module avail
------------------- /home/hpc/jmartins/.module -------------------
octave/5.1.0

------------------- /cvmfs/sw.el7/modules/soft -------------------
aster-13.1.0                gromacs-4.6.7
blat-36.2                   hdf5-1.8.16
....
```

## Manage your new modules environment

```
[username@hpc7 ~]$ module load octave/5.1.0

[username@hpc7 ~]$ which octave
/home/mygroup/username/soft/octave/5.1.0

[username@hpc7 ~]$ octave
octave:1> grid□□# a GRID plot will popup
octave:2> exit

[username@hpc7 ~]$ module unload octave/5.1.0

[username@hpc7 ~]$ which octave
<empty>
```

## Some Links

- Environment Modules Documentation