Using Conda and Anaconda

Conda is a package manager system for Python and other tools and is widely used in some areas such as bioinformatics and data science. On personal computers it is a useful way to install a stack of tools.

The full documentation can be found at

https://docs.conda.io/projects/conda/en/latest/user-guide/index.html

Warning: Conda, whilst convenient, is not designed to be installed on multi-user compute clusters and we are unable to guarantee that tools installed via it will work correctly. This is especially true for any parallel (MPI) tools.

Setting up Conda

First load the appropriate modules

```
$ module load gcc miniconda3
```

For getting the `conda` command to work with your `bash` shell, you need to type

```
eval "$(command conda 'shell.bash' 'hook' 2> /dev/null)"
```

You can automatize this to happen every time you log in, by simply typing the very first time you use it:

```
$ conda init bash
```

This command will hang on a sudo password input, just ignore it (ctrl-c)

You will now probably need to log out and back in again to "activate" the changes.

Once you log in again conda should be available.

However this is **not recommended**, especially if you are using different kind of environments (eg. Conda and Mamba). A convenient option is to define and alias inside your `~/.bashrc` by adding at the end the following line:

```
alias goconda="eval \"\$(command conda 'shell.bash' 'hook' 2> /dev/null)\""
```

Then each time you need Conda, after loading the module, you just type

```
goconda
```

Please ignore any messages about updating to a newer version of conda!

Configuring Conda

By default Conda will put everything including downloads in your home directory. Due to the limited space available this is probable not what you want.

We strongly recommend that you create a `.condarc` file in your home directory with the following options:

```
pkgs_dirs:
  - /work/path/to/my/project/space
```

where the path is the path to your project space on /work - we do not recommend installing things in /scratch as they might be automatically deleted.

You may also wish to add a non standard `env_dirs`

```
envs_dirs:
  - ~/myproject-envs
```

Please see the full `condarc` documentation for all the possible configuration options

https://docs.conda.io/projects/conda/en/latest/user-guide/configuration/use-condarc.html

Using Conda virtual environments

The basic commands for creating conda environments are:

Creation

```
$ conda create --name $MY_CONDA_ENV_NAME
```

Activation

```
$ conda activate $MY_CONDA_ENV_NAME
```

Deactivation

```
$ conda deactivate
```

Environment in specific location

If you need to create an environment in a non standard location:

```
$ conda create --prefix $MY_CONDA_ENV_PATH

$ conda activate $MY_CONDA_ENV_PATH

$ conda deactivate
```

Installing packages

The base commands are:

```
$ conda search $PACKAGE_NAME
$ conda install $PACKAGE_NAME
```

Running Slurm jobs with conda

Since Conda needs some initialization before being used, a Sbatch script must explicitly ask to run bash in *login* mode. This can be performed by adding `--login` option to the shebang. Here is an example of Sbatch script using Conda:

```
#!/bin/bash --login

#SBATCH --time 00-00:05:00
#SBATCH --nodes 1
#SBATCH --ntasks 1
#SBATCH --cpus-per-task 1
#SBATCH --mem 4G

module load gcc miniconda3
conda activate $MY_CONDA_ENV_PATH
```

...

Révision #2
Créé 26 mai 2021 06:49:23 par Emmanuel Jeanvoine
Mis à jour 15 juin 2023 13:03:19 par Flavio Calvo