JupyterLab on the curnagl cluster

JupyterLab can be run on the curnagl cluster for testing purposes, only as an intermediate step in the porting of applications from regular workstations to curnagl.

The installation is made inside a python virtual environment, and this tutorial covers the installation of the following kernels: IPyKernel (**python**), IRKernel (**R**), IJulia (**julia**), MATLAB kernel (**matlab**), IOctave (**octave**), stata_kernel (**stata**) and sas_kernel (**sas**).

If the workstation is outside of the campus, first connect to the VPN.

Creating the virtual environment

First create/choose a folder ${WORK} under the **/scratch** or the **/work** filesystems under your project (ex. WORK=*/work/FAC/.../my_project*). The following needs to be run only once on the cluster (preferably on an interactive computing node):

```
module load gcc python
python -m venv ${WORK}/jlab_venv
${WORK}/jlab_venv/bin/pip install jupyterlab ipykernel numpy matplotlib
```

The IPyKernel is automatically available. The other kernels need to be installed according to your needs.

Installing the kernels

Each time you start a new session on the cluster, remember to define the variable ${WORK} according to the path you chose when creating the virtual environment.

IRKernel

```
module load gcc r
export R_LIBS_USER=${WORK}/jlab_venv/lib/Rlibs
mkdir -p ${R_LIBS_USER}
```

```
echo "install.packages('IRkernel', repos='https://stat.ethz.ch/CRAN/', lib=Sys.getenv('R_LIBS_USER'))" | R --no-save
source ${WORK}/jlab_venv/bin/activate
echo "IRkernel::installspec()" | R --no-save
deactivate
```

IJulia

```
module load gcc julia
export JULIA_DEPOT_PATH=${WORK}/jlab_venv/lib/Jlibs
julia -e 'using Pkg; Pkg.add("IJulia")'
```

MATLAB kernel

```
${WORK}/jlab_venv/bin/pip install matlab_kernel matlabengine==9.11.19
```

IOctave

```
${WORK}/jlab_venv/bin/pip install octave_kernel
echo "c.OctaveKernel.plot_settings = dict(backend='gnuplot')" > ~/.jupyter/octave_kernel_config.py
```

stata_kernel

```
module load stata-se
${WORK}/jlab_venv/bin/pip install stata_kernel
${WORK}/jlab_venv/bin/python -m stata_kernel.install
sed -i "s/^stata_path = None/stata_path = $(echo ${STATA_SE_ROOT} | sed 's/\/\\\/g')\/stata-se/"
~/.stata_kernel.conf
sed -i 's/stata_path = \(.*\)stata-mp/stata_path = \1stata-se/' ~/.stata_kernel.conf
```

sas_kernel

```
module load sas
${WORK}/jlab_venv/bin/pip install sas_kernel
sed -i "s/'\/opt\/sasinside\/SASHome/'$(echo ${SAS_ROOT} | sed 's/\/\\\/g')/g"
${WORK}/jlab_venv/lib64/python3.9/site-packages/saspy/sascfg.py
```

Running JupyterLab

Before running JupyterLab, you need to start an interactive session!

```
Sinteractive
```

Take note of the name of the running node, that you will later need. On curnagl, you can type:

```
hostname
```

If you didn't install all of the kernels, the corresponding lines should be ignored in the commands below. **The execution order is important, in the sense that loading the gcc module should always be done before activating virtual environments.**

```
# Load python
module load gcc python

# IOctave (optional)
module load octave gnuplot

# IRKernel (optional)
export R_LIBS_USER=${WORK}/jlab_venv/lib/Rlibs

# IJulia (optional)
export JULIA_DEPOT_PATH=${WORK}/jlab_venv/lib/Jlibs

# JupyterLab environment
source ${WORK}/jlab_venv/bin/activate

# Launch JupyterLab (on the shell a link that can be copied on the browser will appear)
cd ${WORK}
jupyter-lab

deactivate
```

Before you can copy and paste the link into your favorite browser, you will need to establish an SSH tunnel to the interactive node. From a UNIX-like workstation, you can establish the SSH tunnel to the curnagl node with the following command (replace <username> with your user name, and <hostname> with the name of the node you obtained above, and the <port> number is obtained from the link, it is typically 8888):

```
ssh -n -N -J <username>@curnagl.dcsr.unil.ch -L <port>:localhost:<port> <username>@<hostname>
```

You will be prompted for your password. When you have finished, you can close the tunnel with Ctrl-C.

Note on Python/R/Julia modules and packages

The modules you install manually from JupyterLab in Python, R or Julia end up inside the JupyterLab virtual environment (${WORK}/jlab_venv). They are hence isolated and independent from your Python/R/Julia instances outside of the virtual environment.

Révision #19
Créé 24 février 2023 16:33:02 par Flavio Calvo
Mis à jour 5 juin 2023 13:36:48 par Margot Sirdey