Python Jupyter notebooks @Prometheus

- Prerequisites
- How to use on Prometheus?
    - Via SSH tunnel
        - Submit jupyter job to a computing node
        - Make a tunnel
        - Start the notebook
        - Stop job
        - Status of jobs and finished jobs data
        - Additional software

Prerequisites

Python Jupyter notebooks are usually started on a localhost, which results in starting local webserver and using web browser to interact with the notebook.

On prometheus we cannot easily expose the web socket to external world, as calculations are done internally on a computing node, not visible from Internet.

How to use on Prometheus?

Via SSH tunnel

The trick is to start Jupyter via a job submitted to a computing node and creating a SSH tunnel to access it on a local PC.

Submit jupyter job to a computing node

Create a following file:

pyton-notebook.slurm

```
#!/bin/bash
#SBATCH --partition plgrid-testing
#SBATCH --nodes 1
#SBATCH --ntasks-per-node 6
#SBATCH --time 0:30:00
#SBATCH --job-name jupyter-notebook-tunnel
#SBATCH --output jupyter-log-%J.txt

## get tunneling info
XDG_RUNTIME_DIR=""
ipnport=$(shuf -i8000-9999 -n1)
ipnip=$(hostname -i)
user=$USER

## print tunneling instructions to jupyter-log-{jobid}.txt
echo -e "
    Copy/Paste this in your local terminal to ssh tunnel with remote
    -----------------------------------------------------------------
    ssh -o ServerAliveInterval=300 -N -L $ipnport:$ipnip:$ipnport ${user}@pro.cyfronet.pl
    -----------------------------------------------------------------

    Then open a browser on your local machine to the following address
    ------------------------------------------------------------------
    localhost:$ipnport  (prefix w/ https:// if using password)
    ------------------------------------------------------------------
    "

module load plgrid/tools/python

## start an ipcluster instance and launch jupyter server
jupyter-notebook --no-browser --port=$ipnport --ip=$ipnip
```

JupyterLab

In case you want to start JupyterLab just change last line in above `python-notebook.slurm` script from

Jupyter Notebook

```
jupyter-notebook --no-browser --port=$ipnport --ip=$ipnip
```

to

JupyterLab

```
jupyterlab --no-browser --port=$ipnport --ip=$ipnip
```

Save it as `pyton-notebook.slurm`.

Send job to queue using `sbatch` command on login node of Prometheus

```
sbatch pyton-notebook.slurm
```

Wait until your job enters running state.

To check status of submitted job using `squeue` command

```
squeue -j <JobID>
```

or all jobs of user

```
squeue -u $USER
```

which lists all current user jobs submitted to queue ($USER - is enviromental).

Common states of jobs:

- `PD - PENDING` - Job is awaiting resource allocation.
- `R - RUNNING` - Job currently has an allocation and is running.
- `CF - CONFIGURING` - Job has been allocated resources, but are waiting for them to become ready for use (e.g. booting). On Prometheus `CF` state could last for up to 8 minutes in case when nodes that have been in power save mode.
- `CG - COMPLETING` - Job is in the process of completing. Some processes on some nodes may still be active.

Make a tunnel

In your directory cat jupyter log file:

```
cat jupyter-log-XXXXXXX.txt
```

where `XXXXXXX` is your sbatch job id which is displayed after you run it f.e. `cat jupyter-log-7123485.txt`

It will show you something like this:

```
Copy/Paste this in your local terminal to ssh tunnel with remote
-----------------------------------------------------------------
ssh -o ServerAliveInterval=300 -N -L 8511:172.20.68.193:8511 plgusername@pro.cyfronet.pl
-----------------------------------------------------------------
Then open a browser on your local machine to the following address
-----------------------------------------------------------------
localhost:8511 (prefix w/ https:// if using password)
-----------------------------------------------------------------
## You exec in another shell given command:
> ssh -o ServerAliveInterval=300 -N -L 8511:172.20.68.193:8511 plgusername@pro.cyfronet.pl
## And you are set, you can open in browser:
> `localhost:8511`
```

Exec in another shell at your local computer given command to make a tunnel:

```
ssh -o ServerAliveInterval=300 -N -L 8511:172.20.68.193:8511 plgusername@pro.cyfronet.pl
```

Start the notebook

Open in browser: `` `localhost:8511` ``

If you need jupyter token '`cat jupyter-log-XXXXXXX.txt` once again, and you can copy your token from there.

All informations from jupyter will be stored in this log file.

Stop job

if you wish to to end your sbatch, use `scancel <JOBID>` command, where JOBID is your tunnel JOBID you can look it up with `pro-jobs` or `qsueue -u $USER` commands.

```
scancel <JOBID>
```

Status of jobs and finished jobs data

To check submitted and running jobs use `pro-jobs` or `qsueue -u $USER` commands.

To check information about finished and historic jobs use `pro-jobs-history` command. For example with option "-d 30" that command shows all user's jobs from last 30 days. More info in `pro-jobs-history -h`.

```
pro-jobs-history -d 30
```

Additional software

Python module idact (Interactive Data Analysis Convenience Tools) could be used to automate Jupyter deployment on Prometheus cluster.