

How to VALERIA

1. Get access to Valeria:

- supervisor has to ask for that (Venkata wrote a mail for getting me access)
- I then could sign into the Valeria browser and ssh to the server
- `ssh <your-IDUL>@login.valeria.science` + Valeria pw
- Was not allowed to use any of the tools in Valeria -> Venkata had to send a second mail to ask that I get access for them (including my Laval mail)

2. Folder structure of the account:

- Home -> personal area
- Project -> to work in a team
- Scratch -> working area
- Public -> tutorials etc

3. Bring data to server

- git-repository:
 - clone the repository into home, project or scratch (in the documentation there are more details about the best practice)
- Use Globus Connect Personal to create a node on your personal computer and being able to transfer the data to other points in the system
 - Download globus Connect Personal
 - create collection and say what folders the collection can access on my laptop
 - go to the webapp and look for my local collection (IUCPQ-Ocotech) and select the files I want to sync
 - find the VALERIA laval account and connect with credentials to it
 - start the transaction

4. Start JupyterHub/JupyterLab

- In Valeria (when you got all the permissions) you can start a JupyterHub Server (I just used the default settings for it)
- so far no python libraries are available in there but you can look at your code, open a Terminal
- How to run your code including all the libraries will be explained later

5. The way to submitting my first job to the server

- Jobs are submitted using a bash script which uses SLURM parameters

1. Slurm

- [Parameter descriptions](#)

2. Load Module and Python Environment

- To run python on Valeria you need to load modules `module load StdEnv/2020 python/3.8 scipy-stack/2022a`

- this module has to be loaded each time when you exit and reenter the server
- Then You need a virtual python environment
 - create environnement virtuel
 - `virtualenv --no-download ~/venvs/mon-projet`
 - activate environnement virtuel
 - `source ~/venvs/mon-projet/bin/activate`
 - always upgrade pip
 - `pip install --no-index --upgrade pip`
 - install other Python libraries via wheels
 - search for available wheels:
 - `avail_wheels pandas`
 - `avail_wheels --name "*cdf*"` (looks for library which include "cdf")
 - this virtual environment stays on your account and to activate it on a relogin you need to
 1. reload the modules
 - `module load StdEnv/2020 python/3.8 scipy-stack/2022a`
 2. activate the environemnt
 - `source ~/venvs/mon-projet/bin/activate`
 - all this will be put into a bash script in which the modules are loaded an the environment is created
 - build-env1.sh :

```
#!/bin/bash
module load StdEnv/2020 python/3.8 scipy-stack/2022a
module save mymodules-venv1

Create virtual env
virtualenv --no-download ~/venvs/venv1
source ~/venvs/venv1/bin/activate

pip install --no-index --upgrade pip
pip install --no-index scikit_learn lifelines boto3 s3fs
sqlalchemy psycopg2 pgpasslib xlrd openpyxl
pip install pymrmre
```

- To start a job a second bash script is needed:
 - declare all sbatch parameters, reload earlier saved module and activate virtual environment and finally start the python script
 - run-job.sh:

```
#!/bin/bash
#SBATCH --nodes=1
#SBATCH --partition=bigmem
#SBATCH --nodelist=ul-val-pr-cpu90
#SBATCH --cpus-per-task=32
#SBATCH --mem=200G
#SBATCH --job-name=my-batch
#SBATCH --output=%x-%j.out
```

```

echo "Restoring modules"
module restore mymodules-venv1

# Create virtual env
source ~/venvs/venv1/bin/activate
echo "venv activated"

echo "call python script next"
python script_to_train.py

```

- The python script

- I have to load and save data but the server is not able to get it from my local file paths. The following steps need to be performed:
 1. Go to the Valeria login browser -> My Stockage
 2. Start the S3 Navigateur with *lancer*
 3. Create a bucket (compartiment) with *ajouter compartiment*, name it *bucketname* (no upper letters and spaces)
 4. Upload the needed files directly to the bucket or create an extra folder
- Tell the python code where it can find that bucket
 - boto3 helps with identifying you with your credentials (I do not know exactly how it works but it does with this lines of code so I don't complain. 😊)

```

s3 = boto3.resource('s3', aws_access_key_id=
'YOUR_ACCESS_KEY_ID',
aws_secret_access_key='YOUR_SECRET_ACCESS_KEY')
ENDPOINT_URL = 'https://s3.valeria.science'
bucket = 'bucketname'
# Import
df =
pd.read_csv(f"s3://{bucket}/folder_name_if_you_created_one/file_name.csv", storage_options={"client_kwargs": {'endpoint_url':
ENDPOINT_URL}})
# Export
df.to_csv(f"s3://{bucket}/folder_name_if_you_created_one/file_name.csv", storage_options={"client_kwargs": {'endpoint_url':
ENDPOINT_URL}})

```

- You can also try the working of this in a JupyterHub if you loaded the modules, virtual environment and made it available for jupyterHub (command at the bottom of this page or explanation [here in french](#))
- To start the job:
 - `sbatch build-env1.sh`
 - `sbatch run-job.sh`
- To check if it is running and which jobs are running in general:
 - `sq`
- To see the output file

■ `cat my-batch-JOBID.out`

Make a python environment available for JupyterLab:

```
export NAME=mon-projet
module reset
module load python/3.9 scipy-stack/2022a
module save $NAME

virtualenv --no-download ~/venvs/$NAME
source ~/venvs/$NAME/bin/activate
pip install --no-index --upgrade pip
# install other libraries that you need

python -m ipykernel install --name ${NAME} --user
```

In JupyterLab:

- Go to the upper right corner and select the kernel -Y choose the name you named your project and you can execute code with all the libraries installed in this environment
1. I need data
 - Download globus Connect Personal
 - create collection and say what folders the collection can access on my laptop
 - go to the webapp and look for my local collection (IUCPQ-Ocotech) and select the files I want to sync
 - find the VALERIA laval account and connect with credentials to it
 - start the transaction
 2. ssh works into that account and the same account is also shown in the jupyterHub/lab window
- but it does not have scikit installed