# Enabling web-based interactive notebooks on geographically distributed HPC resources

Alexandre Beche <alexandre.beche@epfl.ch>

## Outlines

1. Context
2. Interactive notebook running on cluster(s)
3. Advanced use cases
4. Summary

## Context

# Blue Brain Project

**Host Institution**
Ecole Polytechnique Fédérale de Lausanne (EPFL)

**Director**
Henry Markram

**Co-Directors**
Sean Hill, Felix Schürmann

**Team today**
~ 100 scientists, engineers & staff

**Timeline**
2005 founded at EPFL
2011/2012 ETH Board funding
**2013-2021 Swiss National Research Infrastructure**

**Main International Collaborations**
Switzerland (CSCS, CERN)
Israel (HUJI)
USA (Yale, ANL, OLCF, Allen Brain)
Spain (UPM)
Saudi Arabia (KAUST)
Europe (HBP)

Deutschland
Frankreich
Österreich
Lausanne
Geneva
Italien
© Bacoo.de

# Data-Driven Modeling & Simulation!

## Neuronal anatomy

- ~ **2 mm** thick
- **55** morphological types
- **13** excitatory & **42** inhibitory m-types
- **31,000** neurons
- **111,700** neurons/mm$^3$
- Excitatory to inhibitory neuron ratio of **86:14** %
- **346** m of axon
- **211** m of dendrites
- Maximum branch order of m-types:

| | Axon | Dendrites |
|---|---|---|
| Excitatory | 24 | 35 |
| Inhibitory | 50 | 17 |

## Neuronal physiology

- **11** electrical types
- **207** morpho-electrical types
- **13** HH type ion channel models
- **bAP** & **EPSP** attenuation for **207** morpho-electrical types
- Ion channel density distribution profiles:

| | Axon | Soma | Dendrites |
|---|---|---|---|
| Uniform | 8 | 6 | 6 |

## Synaptic anatomy

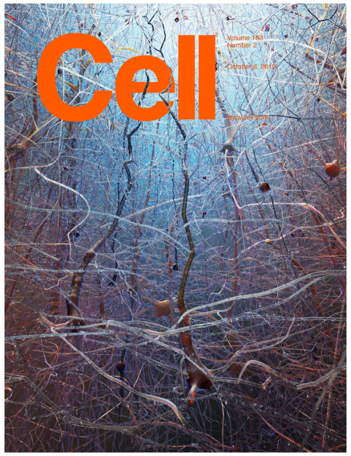
- **0.63** synapses/mm$^3$
- **Extrinsic** to **intrinsic** synapse ratio of **75:25** %
- **3025** possible synaptic pathways
- **2258** viable synaptic pathways
- **664** excitatory pathways
- **1594** inhibitory pathways
- **600** intra-laminar pathways
- **1658** inter-laminar pathways
- Mean synapses/connection

| Exc. | Inh. |
|---|---|
| 4.3 | 8.5 |

## Synaptic physiology

- **6** synapse types
- **207** synaptomes
- **Space clamp** corrected **synaptic conductances** for **607** pathways
- The per synapse conductance of **1.5** nS **for connections between L5TTPCs** is the highest in the microcircuit
- Mean conductance per synapse: **0.85** nS for excitatory & **0.66** nS for inhibitory synapses
- Total conductance in a single neuron is **971** nS

| Excitatory | Inhibitory |
|---|---|
| 697 nS | 274 nS |



Markram et al, Cell 2015

https://bbp.epfl.ch/nmc-portal

- ➢ 80 authors
- ➢ Joint effort between computer and neuro-scientists
- ➢ Reproducible work
- ➢ Extensible

# HPC Today's Infrastructure

## Elastic Compute

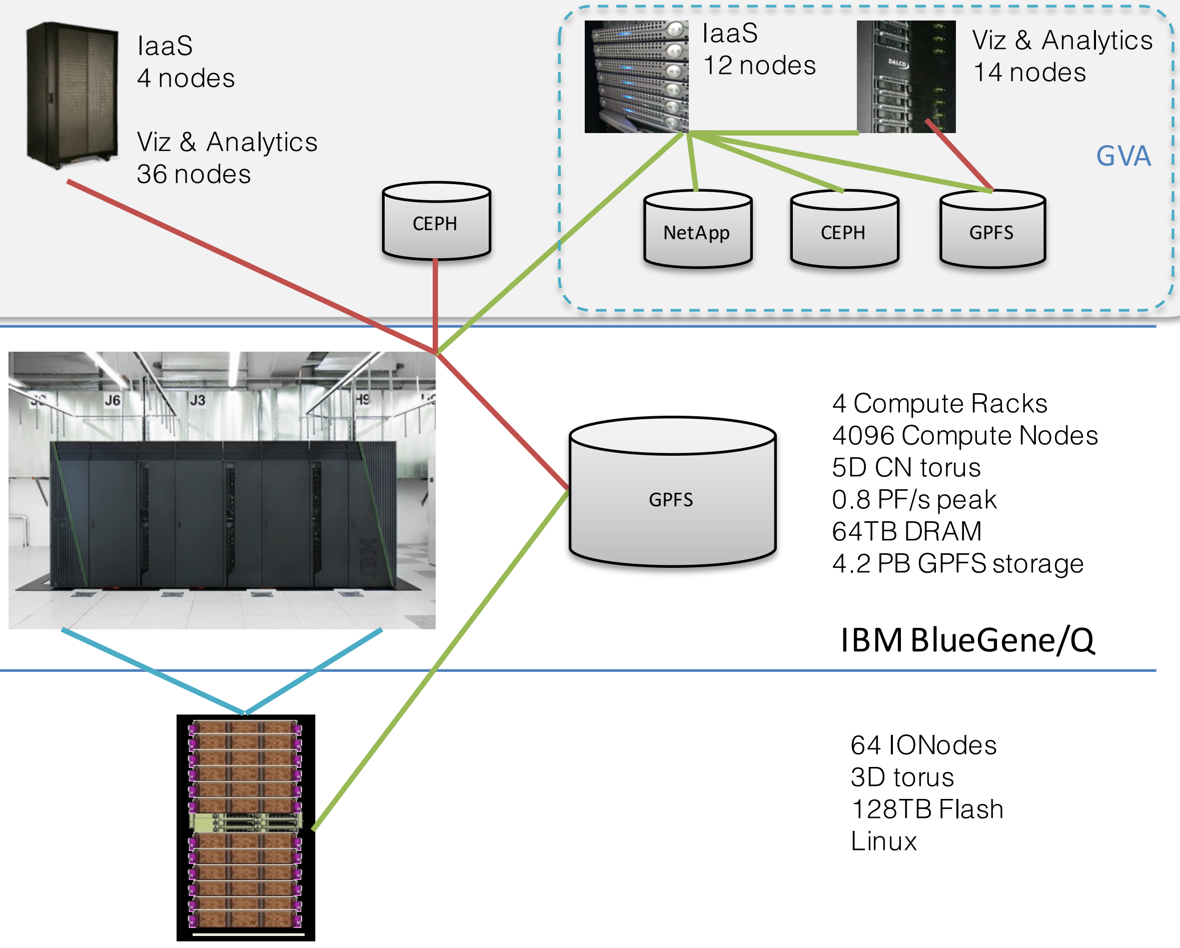
- Visualization and analysis
- Web services
- SW development
- Continuous Integration
- Continuous Deployment

## Production HPC

- Model development
- Reconstruction
- Simulation
- SW development

## Systems Research

- Memory extension
- Application coupling
- Interactive supercomputing
- Reproducibility

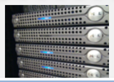


IBM BlueGene Active Storage

# HPC Today's Infrastructure

Elastic Compute

- Visualization and analysis
- Web services
-



GVA

Produ

es

e

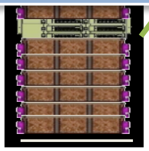
Q

Infrastructure renewal ongoing

New HPC system planned for Q1 2018

Syste

-
- Interactive supercomputing
- Reproducibility



128TB Flash
Linux

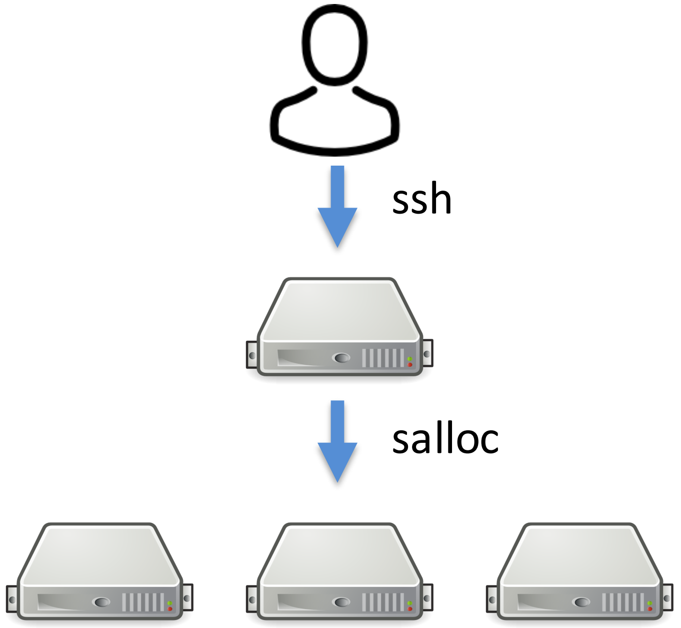
IBM BlueGene Active Storage

## Interactive notebook running on cluster(s)

# Problem description

Scientist wants to run a web-based interactive notebook

1) **Connect to a cluster frontend**
2) **Get an allocation**



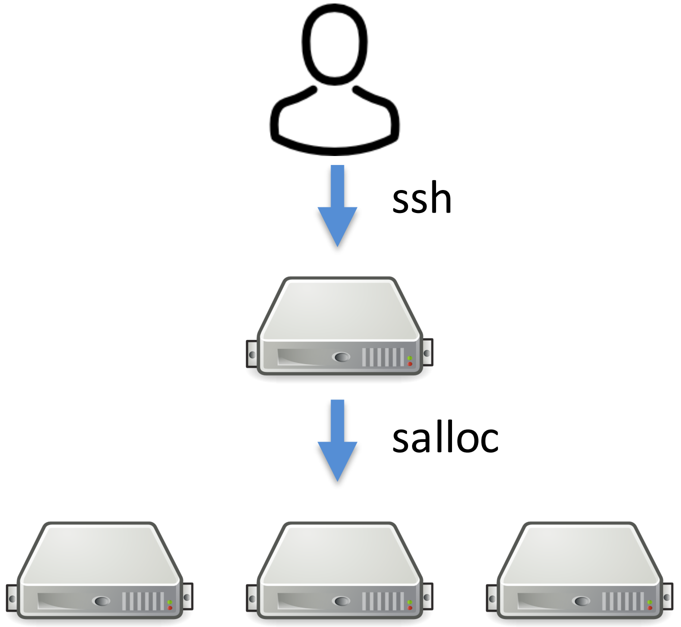
Blue Brain Project

# Problem description

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Scientist wants to run a web-based interactive notebook

1) Connect to a cluster frontend
2) Get an allocation
3) **Run the Jupyter notebook**



```
(thalamus-venv-3.4)-bash-4.1$ . bin/activate
(thalamus-venv-3.4)-bash-4.1$ jupyter notebook --ip=0.0.0.0 --port 10080
[I 16:19:49.604 NotebookApp] The Jupyter Notebook is running at: http://0.0.0.0:10080/
```

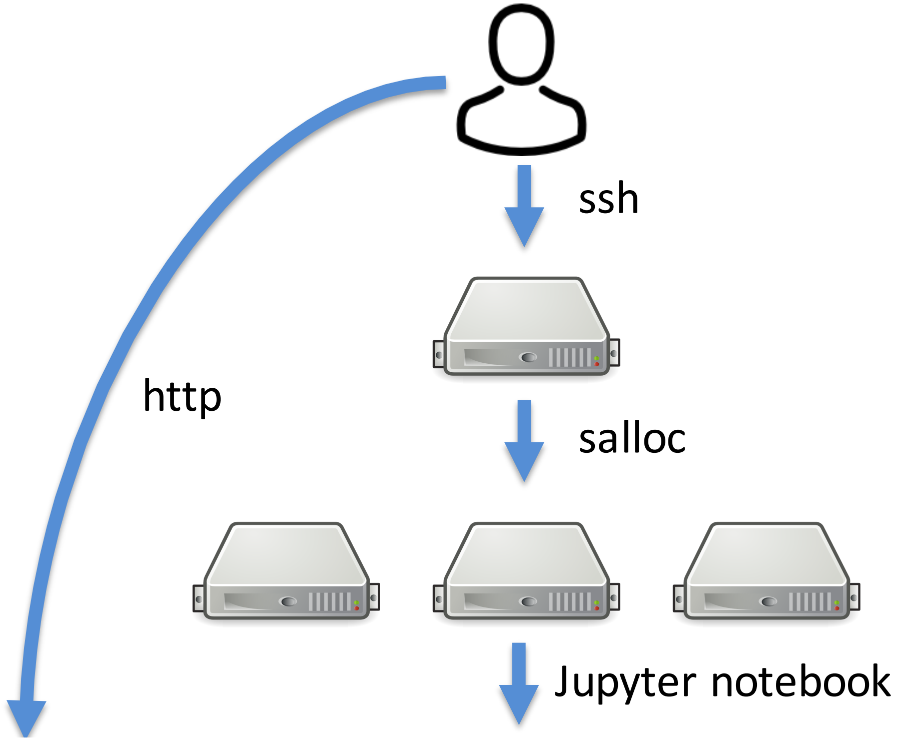
Blue Brain Project

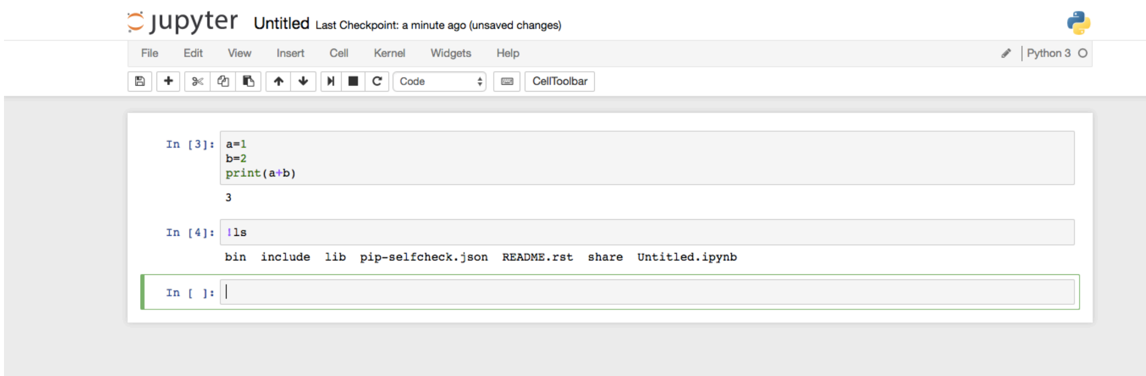
## Problem description

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Scientist wants to run a web-based interactive notebook

1) Connect to a cluster frontend
2) Get an allocation
3) Run the Jupyter notebook
4) **Connect to the notebook**



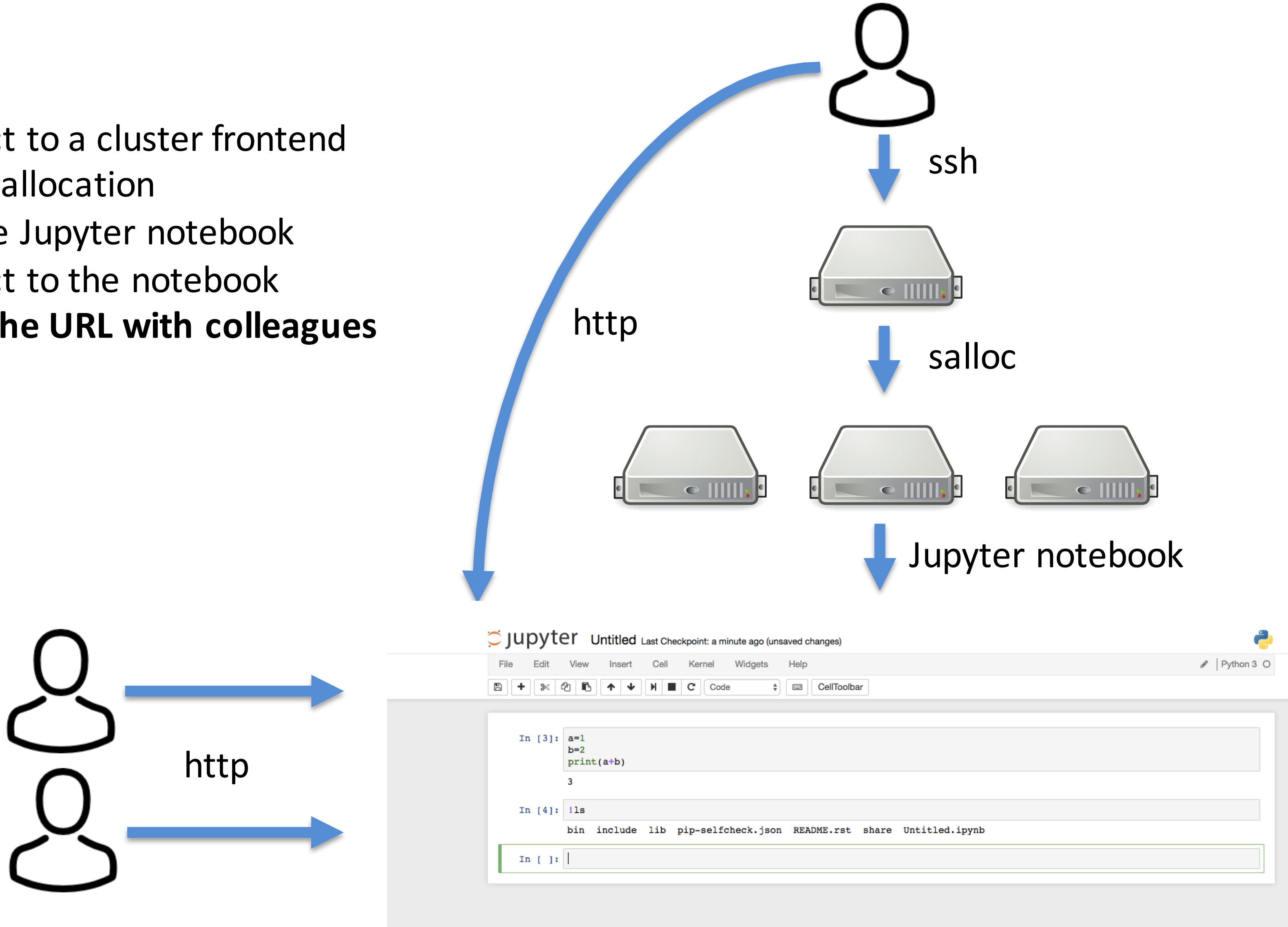

jupyter Untitled Last Checkpoint: a minute ago (unsaved changes)
File Edit View Insert Cell Kernel Widgets Help
Python 3
Code
CellToolbar

```
In [3]: a=1
        b=2
        print(a+b)

        3

In [4]: !ls

        bin  include  lib  pip-selfcheck.json  README.rst  share  Untitled.ipynb

In [ ]:
```

## Problem description

Scientist wants to run a web-based interactive notebook

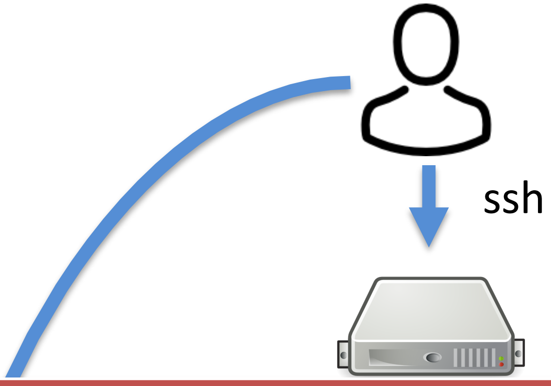
1) Connect to a cluster frontend
2) Get an allocation
3) Run the Jupyter notebook
4) Connect to the notebook
**5) Share the URL with colleagues**


ssh
salloc
Jupyter notebook
http
http
jupyter Untitled Last Checkpoint: a minute ago (unsaved changes)
File Edit View Insert Cell Kernel Widgets Help
Python 3
Code
CellToolbar

```
In [3]: a=1
        b=2
        print(a+b)

        3

In [4]: !ls

        bin  include  lib  pip-selfcheck.json  README.rst  share  Untitled.ipynb

In [ ]:
```

# Problem description

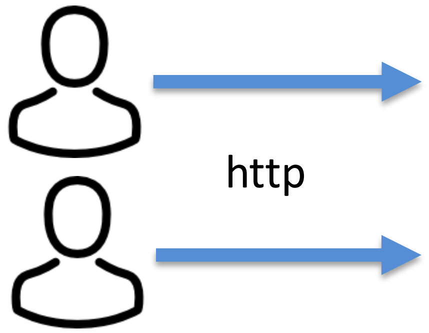
Scientist wants to run a web-based interactive notebook

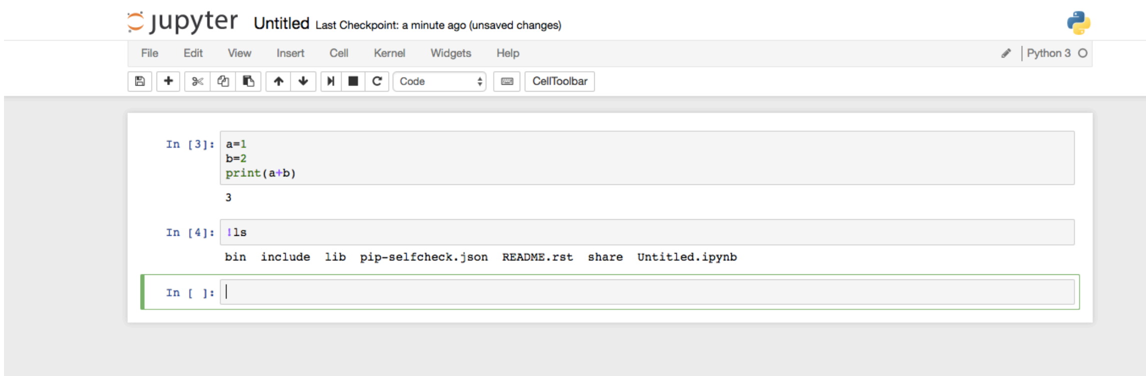


1) Connect to a cluster frontend
2) Get an allocation
3) Run the Jupyter notebook
4) Connect to the notebook

Arbitrary code can now be run on behalf of the user…

… And access to data



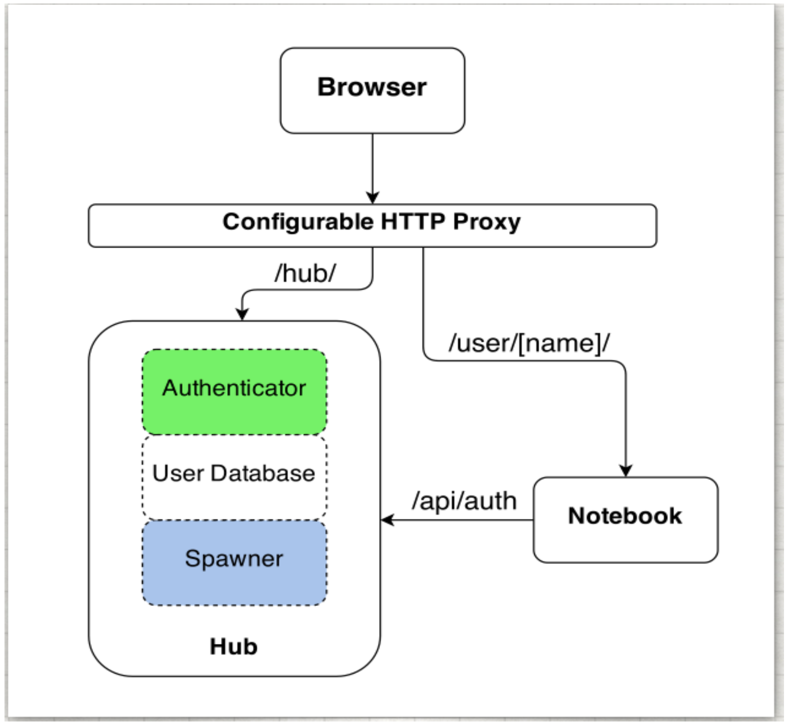

jupyter Untitled Last Checkpoint: a minute ago (unsaved changes)
File Edit View Insert Cell Kernel Widgets Help
Python 3
Code
CellToolbar

```
In [3]: a=1
        b=2
        print(a+b)

        3

In [4]: !ls

        bin  include  lib  pip-selfcheck.json  README.rst  share  Untitled.ipynb

In [ ]:
```

## JupyterHub

**Multi-user server for Jupyter notebooks**

- Authenticate users
- Spawn single-user Jupyter notebook
- Proxy user traffic to notebook securely



Blue Brain Project

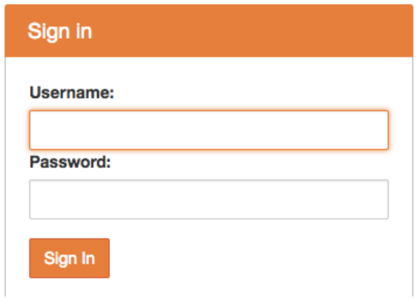
# JupyterHub

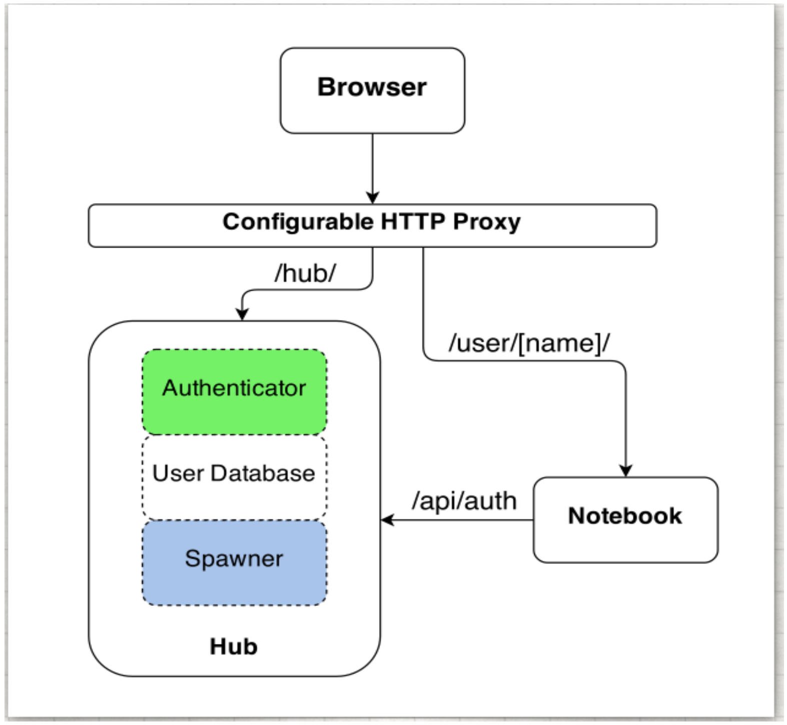
ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

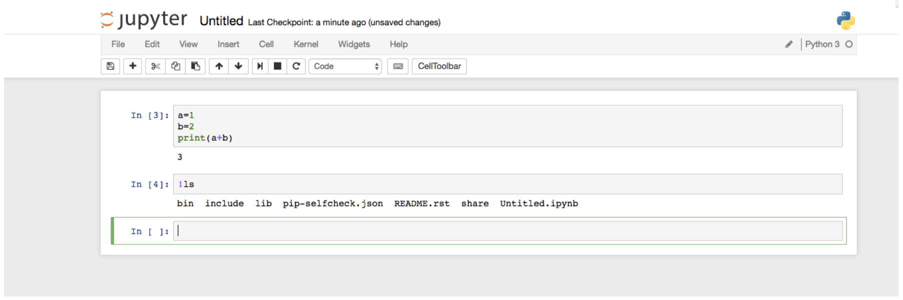
**Multi-user server for Jupyter notebooks**

- Authenticate users
- Spawn single-user Jupyter notebook
- Proxy user traffic to notebook securely

https






jupyter Untitled Last Checkpoint: a minute ago (unsaved changes)
File Edit View Insert Cell Kernel Widgets Help
Python 3
Code
CellToolbar

```
In [3]: a=1
        b=2
        print(a+b)

        3

In [4]: !ls

        bin  include  lib  pip-selfcheck.json  README.rst  share  Untitled.ipynb

In [ ]:
```

**Authenticator:** pam + sssd

**Spawner:** custom slurm spawner

## Basic approach

*Based on https://github.com/mkgilbert/slurmspawner*

**JupyterHub configuration file**

```
c = get_config()
c.JupyterHub.spawner_class = 'SlurmSpawner'
c.SlurmSpawner.job_name = 'JupyterHub'
c.SlurmSpawner.partition = 'batch'
c.SlurmSpawner.account = 'projXX'
c.SlurmSpawner.memory = '1024'
c.SlurmSpawner.time = '06:00:00'
```

Custom Spawner

Slurm options

## Basic approach

*Based on https://github.com/mkgilbert/slurmspawner*

**JupyterHub configuration file**

```
c = get_config()
c.JupyterHub.spawner_class = 'SlurmSpawner'
c.SlurmSpawner.job_name = 'JupyterHub'
c.SlurmSpawner.partition = 'batch'
c.SlurmSpawner.account = 'projXX'
c.SlurmSpawner.memory = '1024'
c.SlurmSpawner.time = '06:00:00'
```

Custom Spawner

Slurm options

**Steps**

1. Run as root on the frontend
2. Resolve uid from logged username
3. Submit job with #SBATCH --uid=$user
4. Job connect back to JupyterHub
5. Gracefully fails if no resources available

# Basic approach

*Based on https://github.com/mkgilbert/slurmspawner*

## JupyterHub configuration file

```
c = get_config()
c.JupyterHub.spawner_class = 'SlurmSpawner'
c.SlurmSpawner.job_name = 'JupyterHub'
c.SlurmSpawner.partition = 'batch'
c.SlurmSpawner.account = 'projXX'
c.SlurmSpawner.memory = '1024'
c.SlurmSpawner.time = '06:00:00'
```

Custom Spawner

Slurm options

## Steps

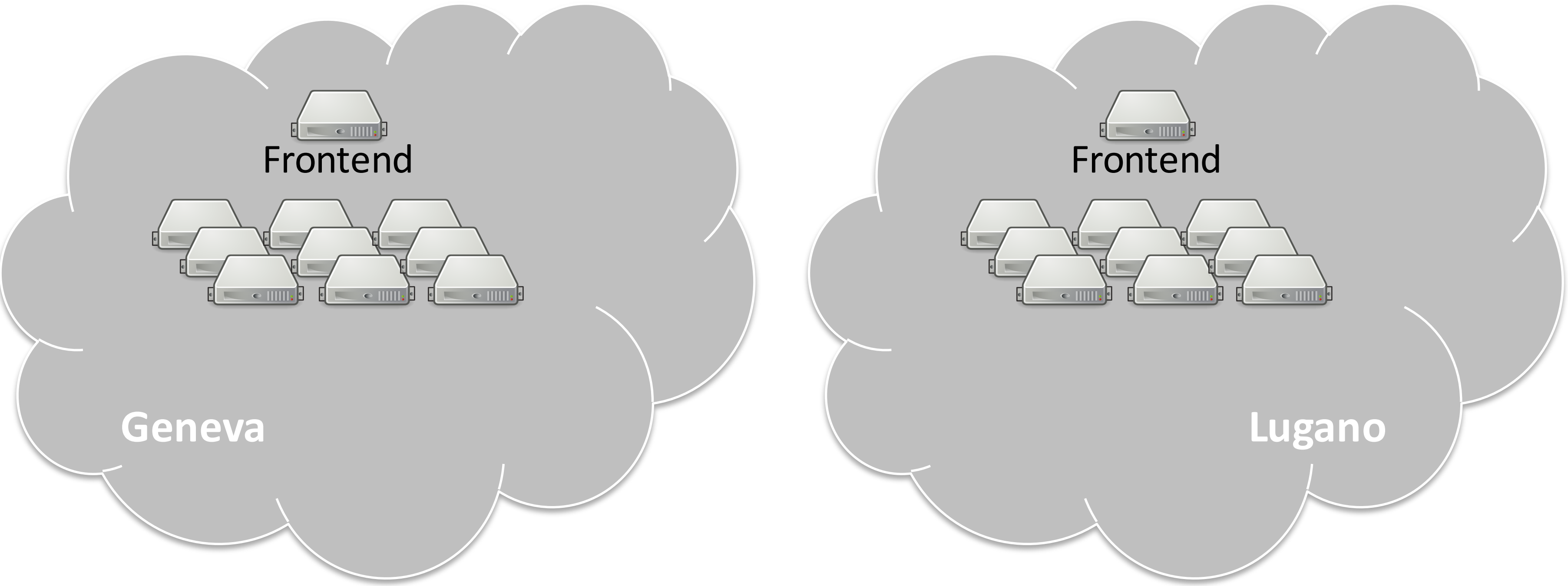
1. Run as root on the frontend
2. Resolve uid from logged username
3. Submit job with #SBATCH --uid=$user
4. Job connect back to JupyterHub
5. Gracefully fails if no resources available

## Limitations

- Single instance per user
- Hardcoded parameters
- No Kerberos credentials (FS required)

# Additional constraints

- BBP infrastructure is multi-sites (2 Slurm clusters)

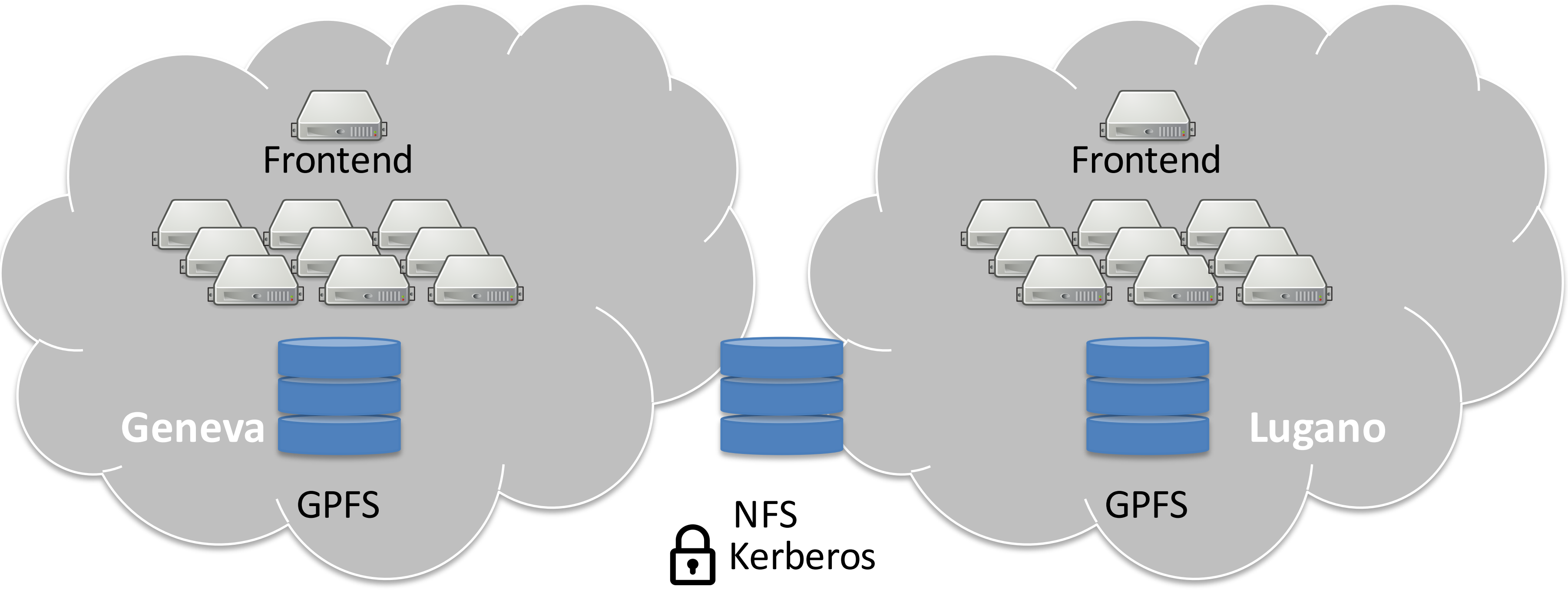


Blue Brain Project

# Additional constraints

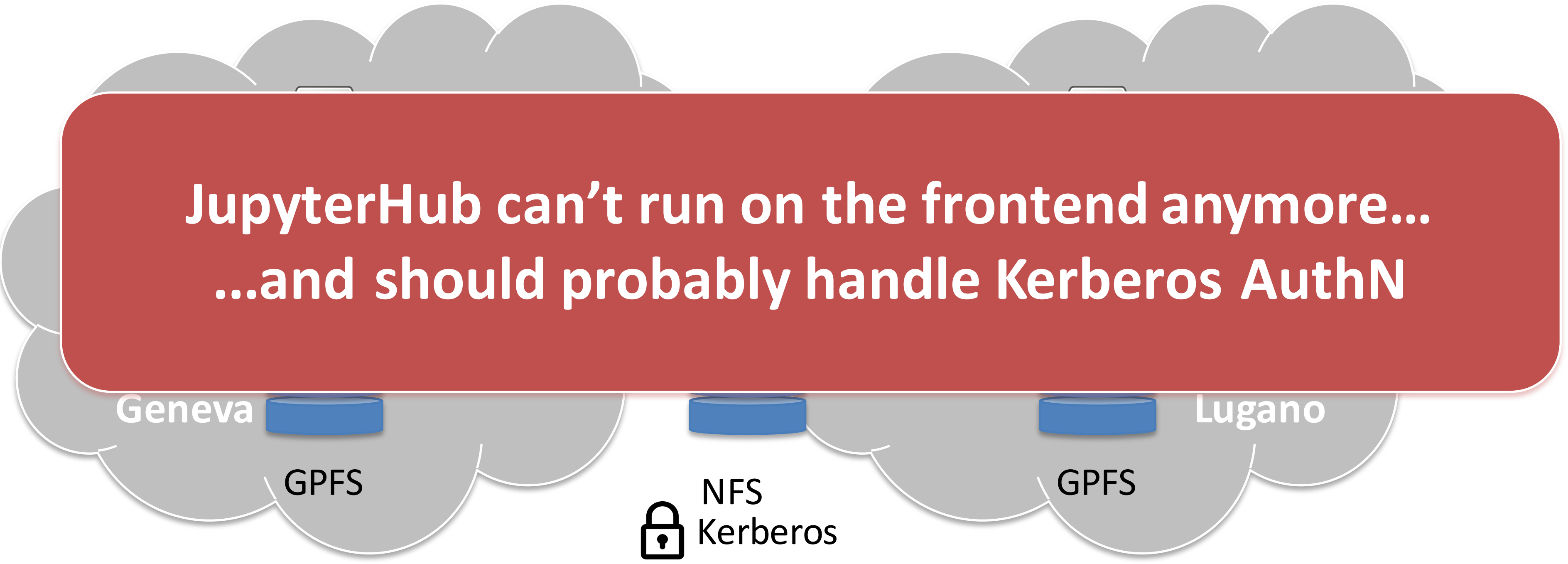
ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

- BBP infrastructure is multi-sites (2 Slurm clusters)
- User needs access to Shared File Systems (Kerberos required)



# Additional constraints

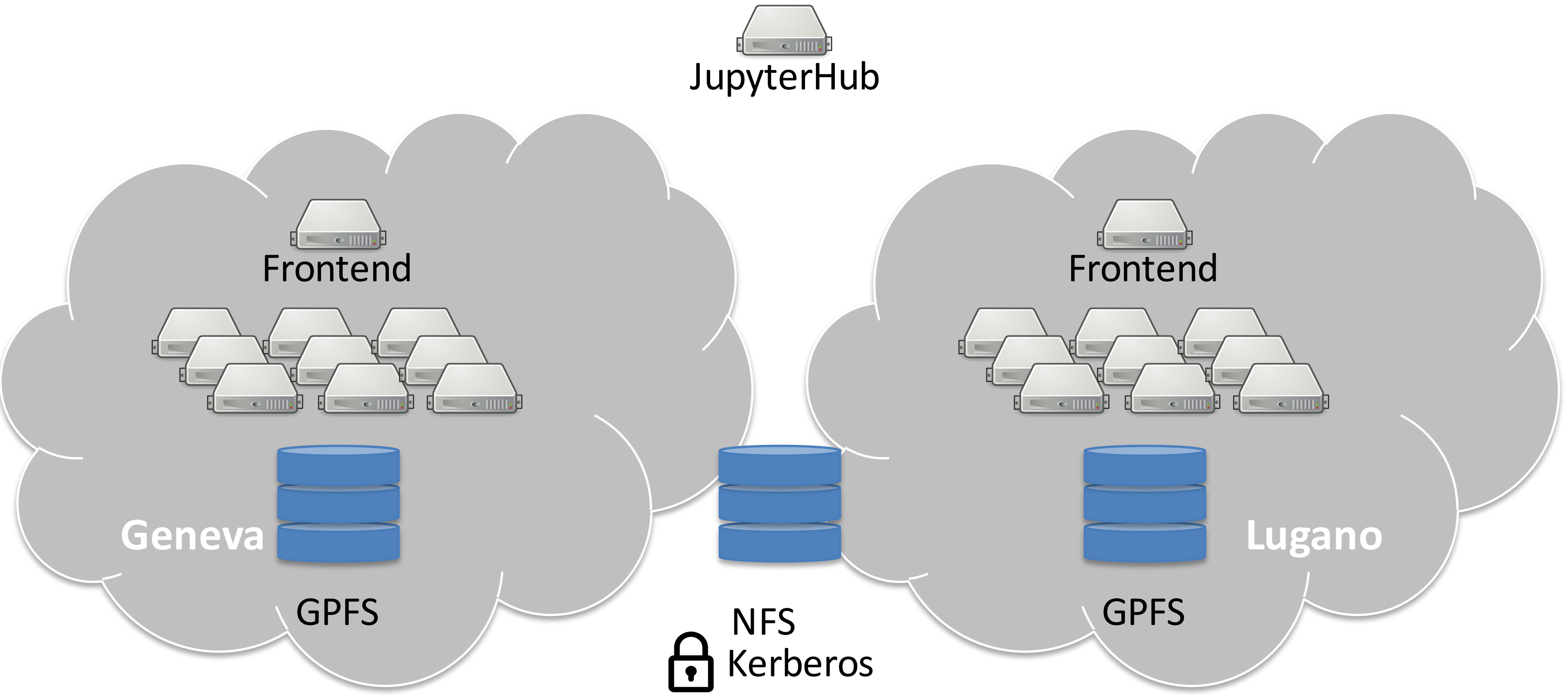
- BBP infrastructure is multi-sites (2 Slurm clusters)
- User needs access to Shared File Systems (Kerberos required)



**JupyterHub can't run on the frontend anymore…**
**…and should probably handle Kerberos AuthN**

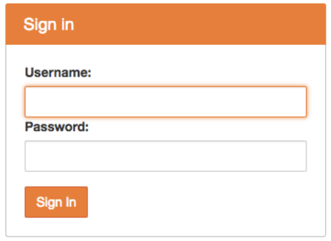
# Site agnostic approach

- JupyterHub is installed out of any cluster
- And should securely submit a job (ssh as user + sbatch)



# Authentication mechanism

Step 1. Authentication

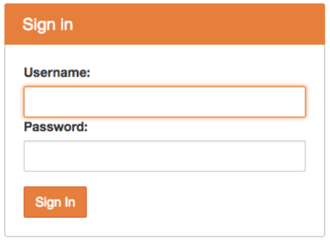


KerberosPAMAuthenticator

Username + password => kerberos token

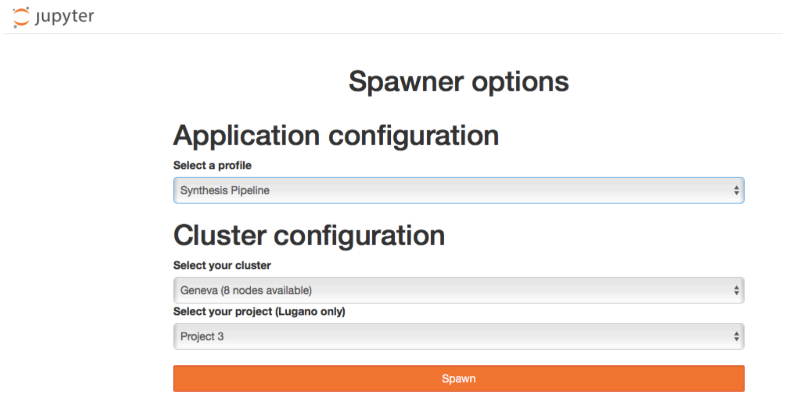
# Authentication mechanism

Step 1. Authentication



KerberosPAMAuthenticator
Username + password => kerberos token

Step 2. Job Submission


jupyter

Spawner options

Application configuration

Select a profile: Synthesis Pipeline

Cluster configuration

Select your cluster: Geneva (8 nodes available)

Select your project (Lugano only): Project 3

**Kerberos token forwarded using**

- ssh (to frontend)
- auks (to compute)

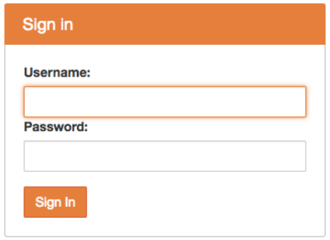
```
def set_cache_file(username):
    """ This function finds a kerberos cache file for the given username,
    file ownership is checked to avoid impersonation """
    userid = pwd.getpwnam(username).pw_uid
    cache_pattern = '/tmp/krb5cc_{0}*'.format(userid)

    # Finding the newest cache file belongin to the user
    cache_file = None
    for f in glob.glob(cache_pattern):
        if os.stat(f).st_uid != userid:
            continue
        if not cache_file or os.stat(f).st_ctime > os.stat(cache_file).st_ctime:
            cache_file = f

    if cache_file:
        os.environ['KRB5CCNAME'] = 'FILE:{0}'.format(cache_file)
        return 'KRB5CCNAME=FILE:{0}'.format(cache_file)
```

Blue Brain Project

# Authentication mechanism

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

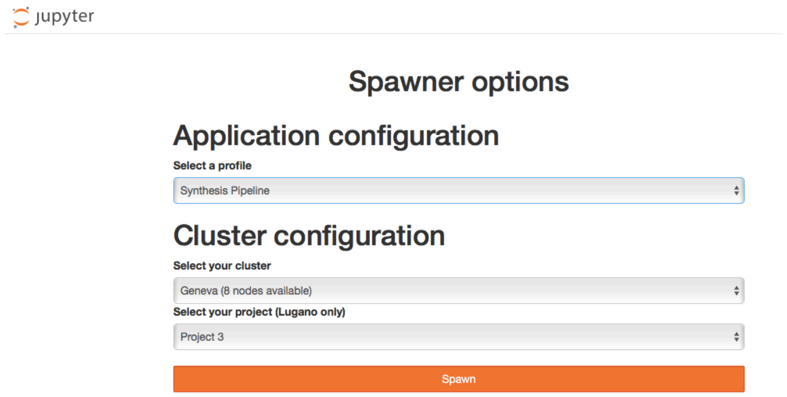
## Step 1. Authentication



KerberosPAMAuthenticator

Username + password => kerberos token

## Step 2. Job Submission


jupyter

Spawner options

Application configuration

Select a profile: Synthesis Pipeline

Cluster configuration

Select your cluster: Geneva (8 nodes available)

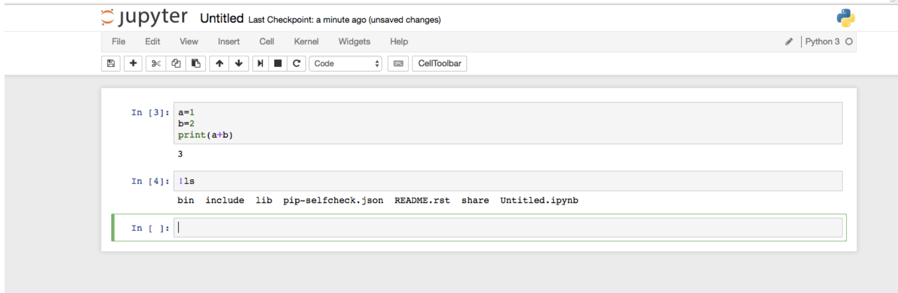
Select your project (Lugano only): Project 3

**Kerberos token forwarded using**

- ssh (to frontend)
- auks (to compute)

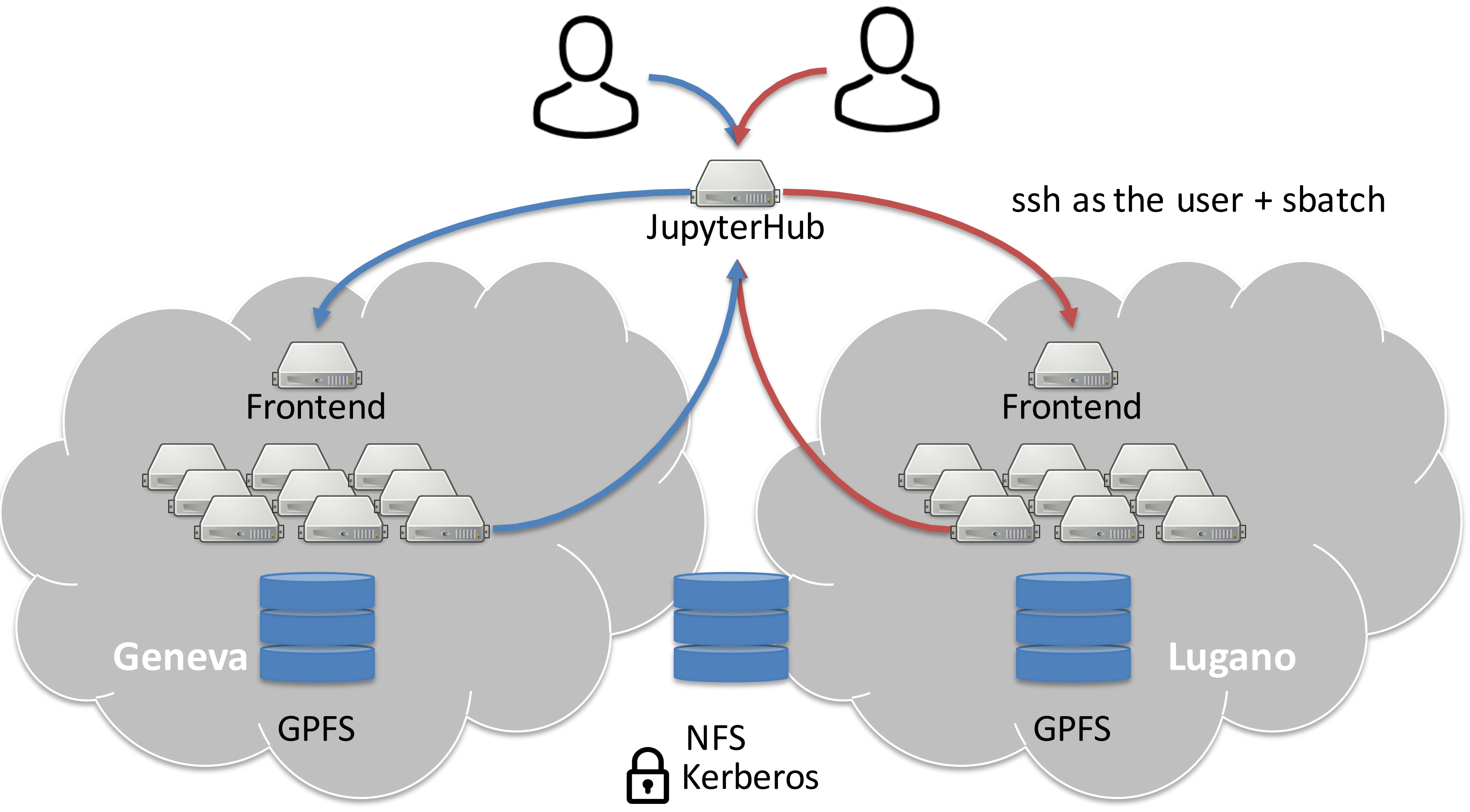
```python
def set_cache_file(username):
    """ This function finds a kerberos cache file for the given username,
    file ownership is checked to avoid impersonation """
    userid = pwd.getpwnam(username).pw_uid
    cache_pattern = '/tmp/krb5cc_{0}*'.format(userid)

    # Finding the newest cache file belongin to the user
    cache_file = None
    for f in glob.glob(cache_pattern):
        if os.stat(f).st_uid != userid:
            continue
        if not cache_file or os.stat(f).st_ctime > os.stat(cache_file).st_ctime:
            cache_file = f

    if cache_file:
        os.environ['KRB5CCNAME'] = 'FILE:{0}'.format(cache_file)
        return 'KRB5CCNAME=FILE:{0}'.format(cache_file)
```

## Step 3. Access notebook


jupyter Untitled Last Checkpoint: a minute ago (unsaved changes)

```
In [3]: a=1
        b=2
        print(a+b)

        3

In [4]: !ls

        bin  include  lib  pip-selfcheck.json  README.rst  share  Untitled.ipynb

In [ ]:
```

# Site agnostic approach



**Limitation:** User can no longer collaboratively work on a notebook

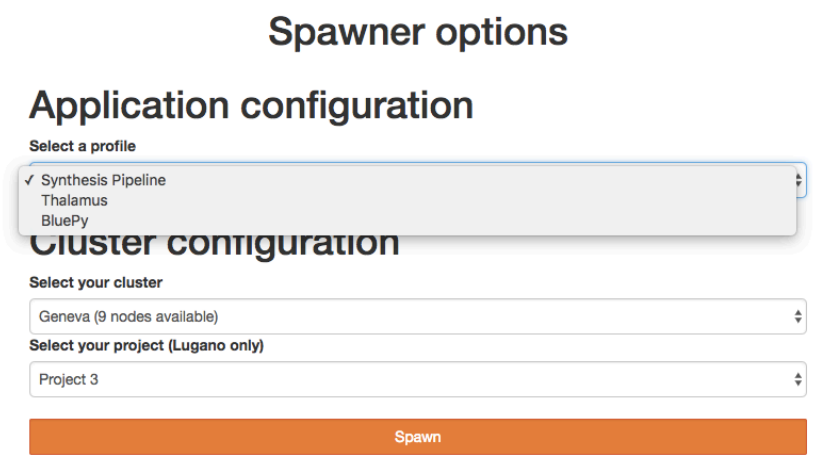
## Advance use cases:
## Extending JupyterHub with custom kernels and modules

Blue Brain Project

## JupyterHub profiles

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

BBP extension to abstract environment complexity and improve reproducibility


Spawner options

Application configuration

Select a profile

✓ Synthesis Pipeline
Thalamus
BluePy

Cluster configuration

Select your cluster

Geneva (9 nodes available)

Select your project (Lugano only)

Project 3

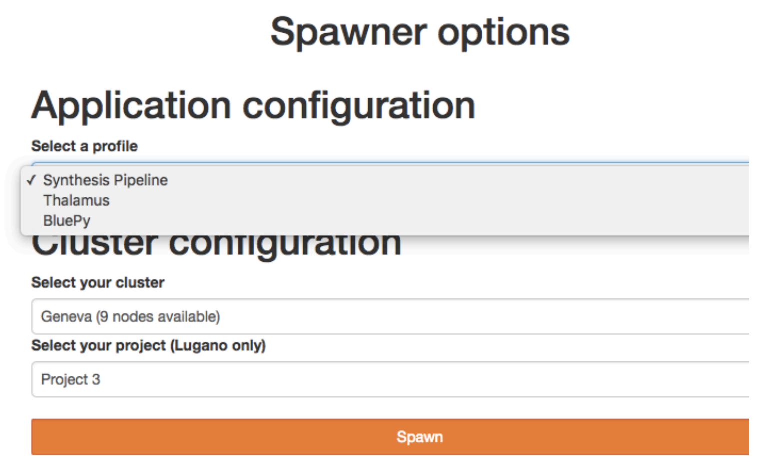
Spawn

Blue Brain Project

# JupyterHub profiles

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

BBP extension to abstract environment complexity and improve reproducibility



Spawner options

Application configuration

Select a profile

✓ Synthesis Pipeline
Thalamus
BluePy

Cluster configuration

Select your cluster

Geneva (9 nodes available)

Select your project (Lugano only)

Project 3

Spawn

**Arbitrary name**

```
PROFILES = {
  'synthesis_pipeline': {
    'name': 'Synthesis Pipeline',
    'core': 1,
    'memory': 1024,
    'kernel': '/gpfs/project/proj1/synthesis/install.sh',
    'modules': [ 'nix/nse/morphsyn/9040c' ]
  },
  'bluepy': {
    'name': 'BluePy',
    'core': 8,
    'memory': 65536,
    'kernel': 'source /opt/rh/python27/enable && /gpfs/project/proj2/bluepy/install.sh',
    'modules': []
  },
  'thalamus': {
    'name': 'Thalamus',
    'core': 2,
    'memory': 4096,
    'kernel': '/gpfs/project/proj3/thalamus/2.7.sh && /gpfs/project/proj3/thalamus/3.4.sh',
    'modules': [ 'nix/hpc/neuron/7.5-201707', 'nix/viz/rtneuron/2.13.0-201707' ]
  },
}
```

**Optimal resources**

**Environment:**

- Python environment
- Set of modules

Blue Brain Project

# Custom virtual environments

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

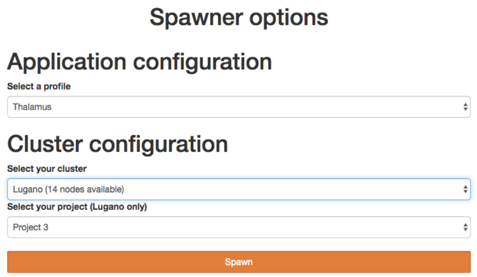
```
'thalamus': {
  'name': 'Thalamus',
  'core': 1,
  'memory': 1024,
  'kernel': '/gpfs/project/proj1/thalamus_venv-3.4.sh',
  'modules': [
    'nix/hpc/neuron/7.5-201707',
    'nix/viz/rtneuron/2.13.0-201707'
  ]
}
```

```
#!/bin/bash

VENV_PATH="${HOME}/thalamus-venv-3.4"

if [ -d $VENV_PATH ]; then exit; fi

module purge
module load nix/python/3.4-light

echo "Using `which python`"
echo "Creating environement $VENV_PATH"
virtualenv --clear $VENV_PATH -p `which python3`

source $VENV_PATH/bin/activate
echo "Now using `which python`"
echo $PATH

pip install six
pip install jupyter
pip install GitPython
pip install numpy
pip install scipy
pip install pandas
pip install seaborn
pip install ipykernel

`which python3` -m ipykernel install --user --name "Thalamus_34"
```

# Custom virtual environments

```
'thalamus': {
  'name': 'Thalamus',
  'core': 1,
  'memory': 1024,
  'kernel': '/gpfs/project/proj1/thalamus_venv-3.4.sh',
  'modules': [
    'nix/hpc/neuron/7.5-201707',
    'nix/viz/rtneuron/2.13.0-201707'
  ]
}
```

```
#!/bin/bash

VENV_PATH="${HOME}/thalamus-venv-3.4"

if [ -d $VENV_PATH ]; then exit; fi

module purge
module load nix/python/3.4-light

echo "Using `which python`"
echo "Creating environement $VENV_PATH"
virtualenv --clear $VENV_PATH -p `which python3`

source $VENV_PATH/bin/activate
echo "Now using `which python`"
echo $PATH

pip install six
pip install jupyter
pip install GitPython
pip install numpy
pip install scipy
pip install pandas
pip install seaborn
pip install ipykernel

`which python3` -m ipykernel install --user --name "Thalamus_34"
```



Spawner options

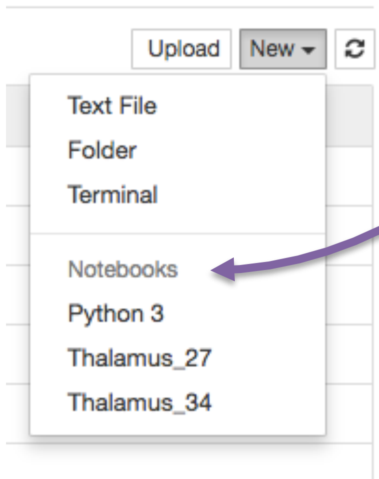
Application configuration

Select a profile: Thalamus

Cluster configuration

Select your cluster: Lugano (14 nodes available)

Select your project (Lugano only): Project 3

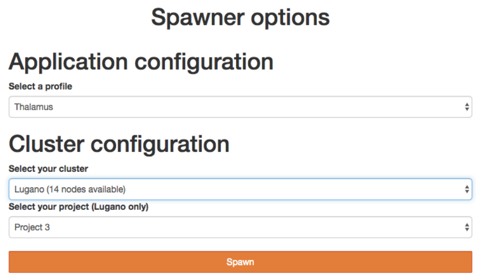


Kernels

# Custom virtual environments

```
'thalamus': {
  'name': 'Thalamus',
  'core': 1,
  'memory': 1024,
  'kernel': '/gpfs/project/proj1/thalamus_venv-3.4.sh',
  'modules': [
    'nix/hpc/neuron/7.5-201707',
    'nix/viz/rtneuron/2.13.0-201707'
  ]
}
```

```
#!/bin/bash

VENV_PATH="${HOME}/thalamus-venv-3.4"

if [ -d $VENV_PATH ]; then exit; fi

module purge
module load nix/python/3.4-light

echo "Using `which python`"
echo "Creating environement $VENV_PATH"
virtualenv --clear $VENV_PATH -p `which python3`

source $VENV_PATH/bin/activate
echo "Now using `which python`"
echo $PATH

pip install six
pip install jupyter
pip install GitPython
pip install numpy
pip install scipy
pip install pandas
pip install seaborn
pip install ipykernel

`which python3` -m ipykernel install --user --name "Thalamus_34"
```



Spawner options

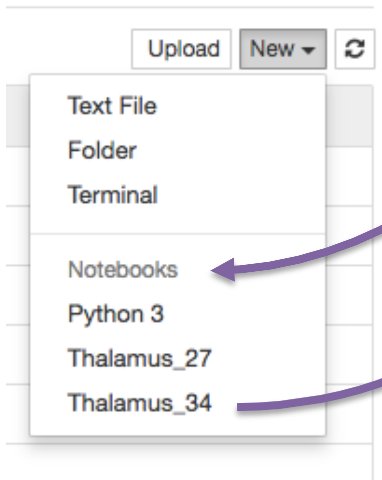
Application configuration

Select a profile: Thalamus

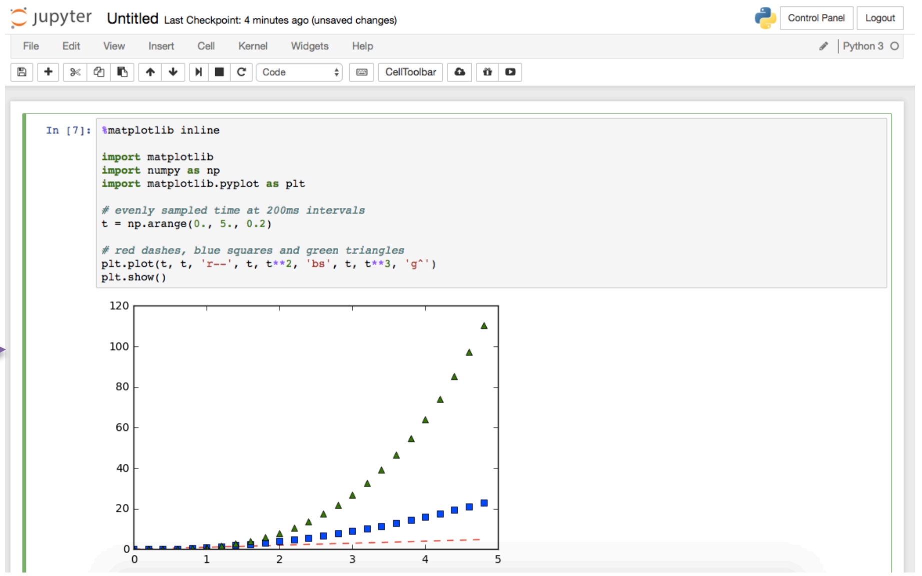
Cluster configuration

Select your cluster: Lugano (14 nodes available)

Select your project (Lugano only): Project 3

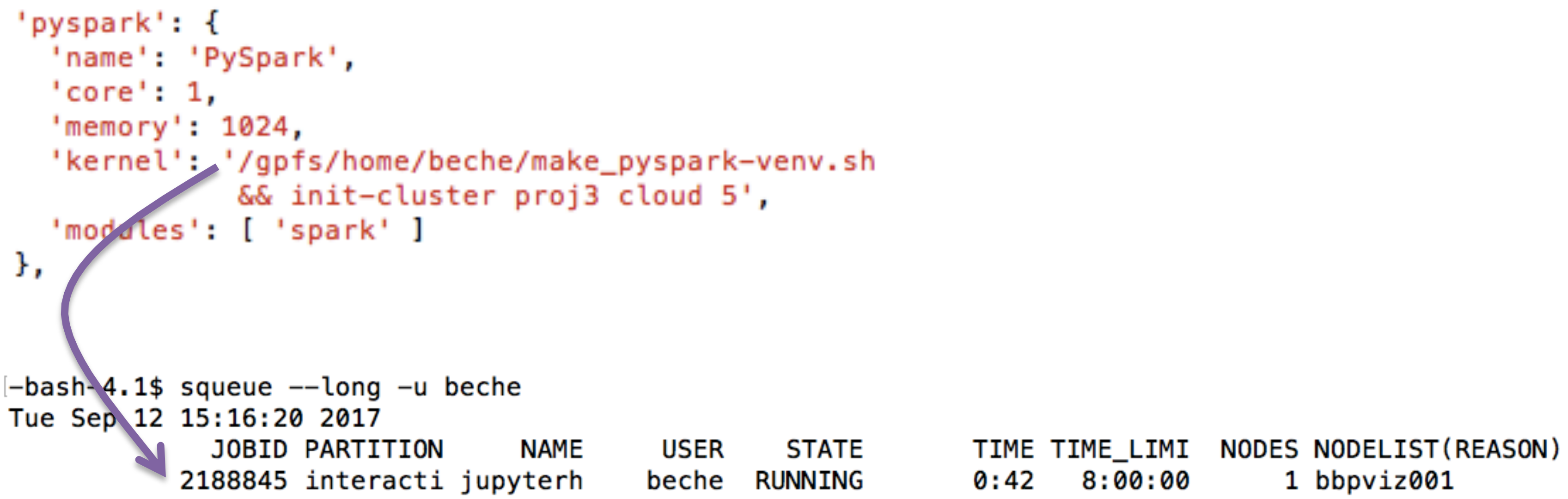


Kernels



```
In [7]: %matplotlib inline

        import matplotlib
        import numpy as np
        import matplotlib.pyplot as plt

        # evenly sampled time at 200ms intervals
        t = np.arange(0., 5., 0.2)

        # red dashes, blue squares and green triangles
        plt.plot(t, t, 'r--', t, t**2, 'bs', t, t**3, 'g^')
        plt.show()
```

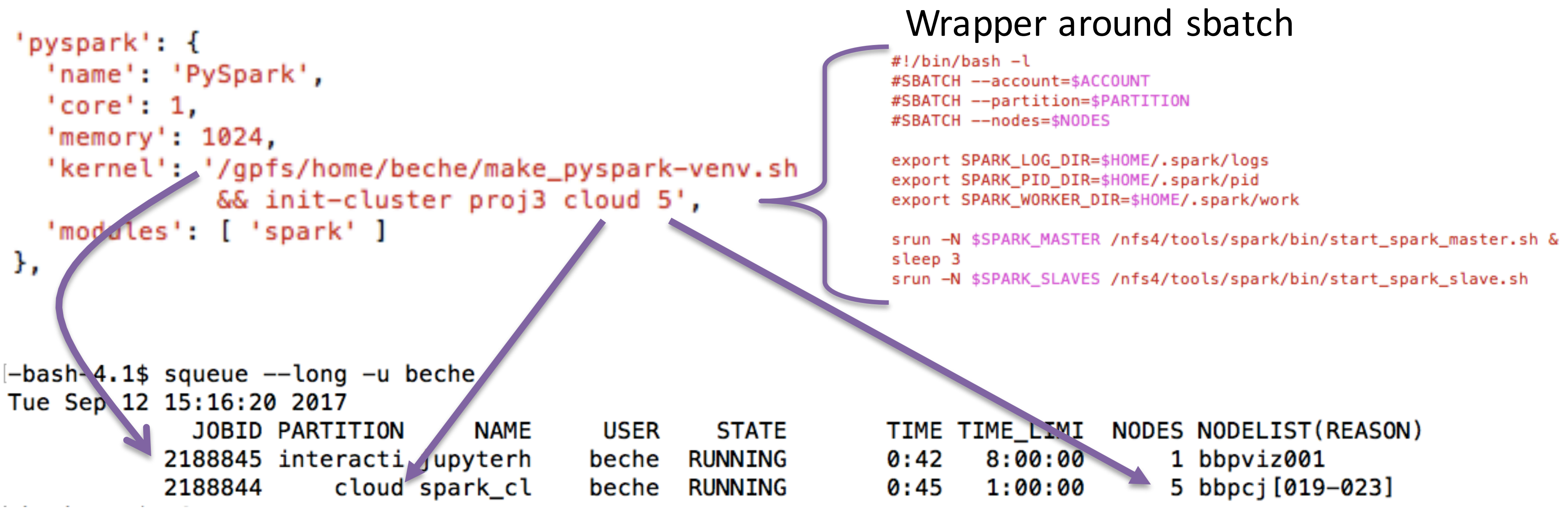
# Integration with Spark



```
'pyspark': {
   'name': 'PySpark',
   'core': 1,
   'memory': 1024,
   'kernel': '/gpfs/home/beche/make_pyspark-venv.sh
             && init-cluster proj3 cloud 5',
   'modules': [ 'spark' ]
},
```

```
-bash-4.1$ squeue --long -u beche
Tue Sep 12 15:16:20 2017
             JOBID PARTITION     NAME     USER    STATE       TIME TIME_LIMI  NODES NODELIST(REASON)
           2188845 interacti jupyterh    beche  RUNNING       0:42   8:00:00      1 bbpviz001
```

# Integration with Spark



Wrapper around sbatch

```
'pyspark': {
   'name': 'PySpark',
   'core': 1,
   'memory': 1024,
   'kernel': '/gpfs/home/beche/make_pyspark-venv.sh
              && init-cluster proj3 cloud 5',
   'modules': [ 'spark' ]
},
```

```
#!/bin/bash -l
#SBATCH --account=$ACCOUNT
#SBATCH --partition=$PARTITION
#SBATCH --nodes=$NODES

export SPARK_LOG_DIR=$HOME/.spark/logs
export SPARK_PID_DIR=$HOME/.spark/pid
export SPARK_WORKER_DIR=$HOME/.spark/work

srun -N $SPARK_MASTER /nfs4/tools/spark/bin/start_spark_master.sh &
sleep 3
srun -N $SPARK_SLAVES /nfs4/tools/spark/bin/start_spark_slave.sh
```

```
-bash-4.1$ squeue --long -u beche
Tue Sep 12 15:16:20 2017
             JOBID PARTITION     NAME     USER    STATE       TIME TIME_LIMI  NODES NODELIST(REASON)
           2188845 interacti jupyterh    beche  RUNNING       0:42   8:00:00      1 bbpviz001
           2188844     cloud spark_cl    beche  RUNNING       0:45   1:00:00      5 bbpcj[019-023]
```

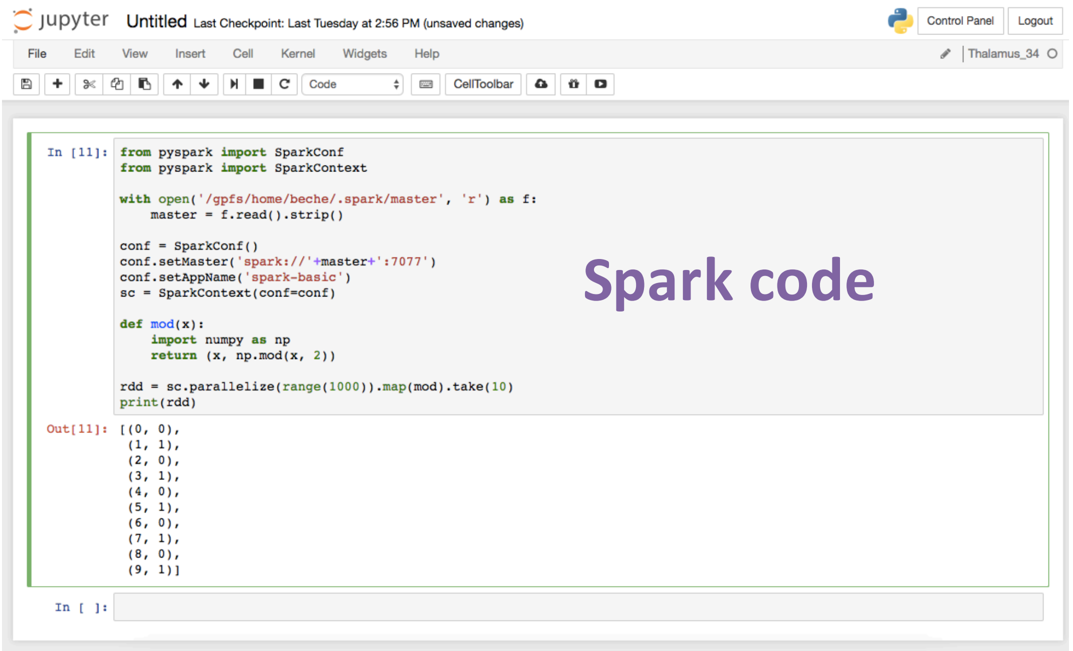
# Integration with Spark

```
'pyspark': {
   'name': 'PySpark',
   'core': 1,
   'memory': 1024,
   'kernel': '/gpfs/home/beche/make_pyspark-venv.sh
              && init-cluster proj3 cloud 5',
   'modules': [ 'spark' ]
},
```

Wrapper around sbatch

```
#!/bin/bash -l
#SBATCH --account=$ACCOUNT
#SBATCH --partition=$PARTITION
#SBATCH --nodes=$NODES

export SPARK_LOG_DIR=$HOME/.spark/logs
export SPARK_PID_DIR=$HOME/.spark/pid
export SPARK_WORKER_DIR=$HOME/.spark/work

srun -N $SPARK_MASTER /nfs4/tools/spark/bin/start_spark_master.sh &
sleep 3
srun -N $SPARK_SLAVES /nfs4/tools/spark/bin/start_spark_slave.sh
```

```
[-bash-4.1$ squeue --long -u beche
Tue Sep 12 15:16:20 2017
          JOBID PARTITION     NAME     USER    STATE       TIME TIME_LIMI  NODES NODELIST(REASON)
        2188845 interacti jupyterh    beche  RUNNING       0:42   8:00:00      1 bbpviz001
        2188844     cloud spark_cl    beche  RUNNING       0:45   1:00:00      5 bbpcj[019-023]
```



```
In [11]: from pyspark import SparkConf
         from pyspark import SparkContext

         with open('/gpfs/home/beche/.spark/master', 'r') as f:
             master = f.read().strip()

         conf = SparkConf()
         conf.setMaster('spark://'+master+':7077')
         conf.setAppName('spark-basic')
         sc = SparkContext(conf=conf)

         def mod(x):
             import numpy as np
             return (x, np.mod(x, 2))

         rdd = sc.parallelize(range(1000)).map(mod).take(10)
         print(rdd)

Out[11]: [(0, 0),
          (1, 1),
          (2, 0),
          (3, 1),
          (4, 0),
          (5, 1),
          (6, 0),
          (7, 1),
          (8, 0),
          (9, 1)]

In [ ]:
```

Spark code

## Summary

## Limitations and solutions

**JupyterHub does not support multiple “singleuser” process**

- Non blocking limitation (and may be soon supported upstream)

**JupyterHub notebook are user specific (not collaborative)**

- “copy on open” notebooks

**Sharing and versioning of the notebook**

- Nbgallery (https://github.com/nbgallery/nbgallery**)**

## Summary

- Provide a secure way to interact with remote web-based interactive notebooks
- Extend the tool to lower entry barrier to scientists and improve environment reproducibility
- Integrate the tools with other technology to leverage parallel computing such as Spark

# We are hiring

- Devops
- HPC specialists
- Storage specialists

jobs.bbp@epfl.ch

## Acknowledgements

BBP core services & HPC teams