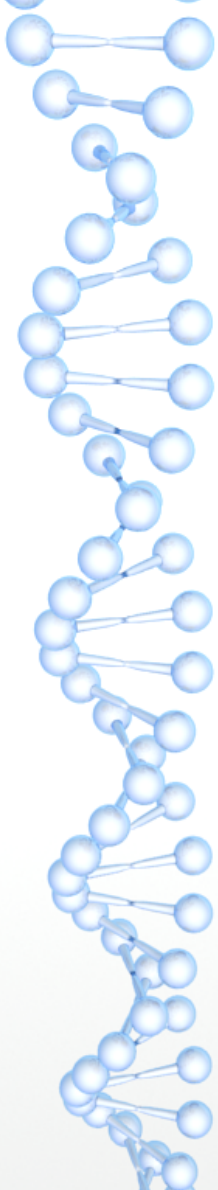


Running jupyter notebook remotely in a docker swarm cluster

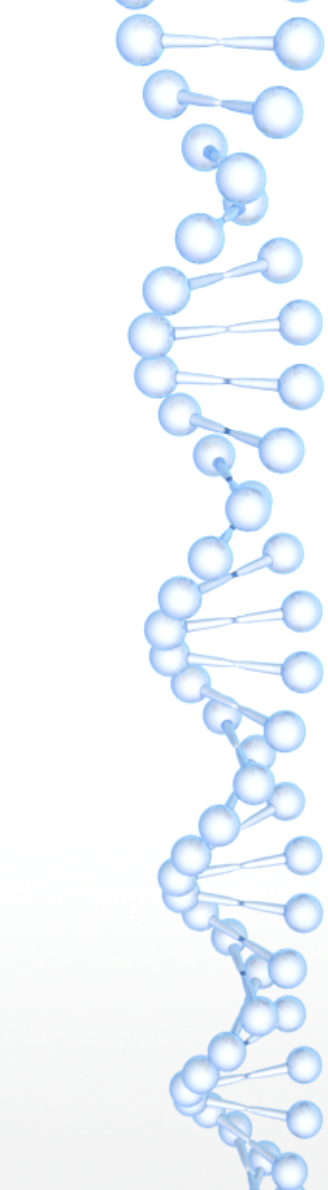
Jordi Deu-Pons
Barcelona 2017

PyData jupyter docker

BBGLab
BARCELONA BIOMEDICAL GENOMICS LAB

IRB BARCELONA
INSTITUTE FOR RESEARCH IN BIOMEDICINE

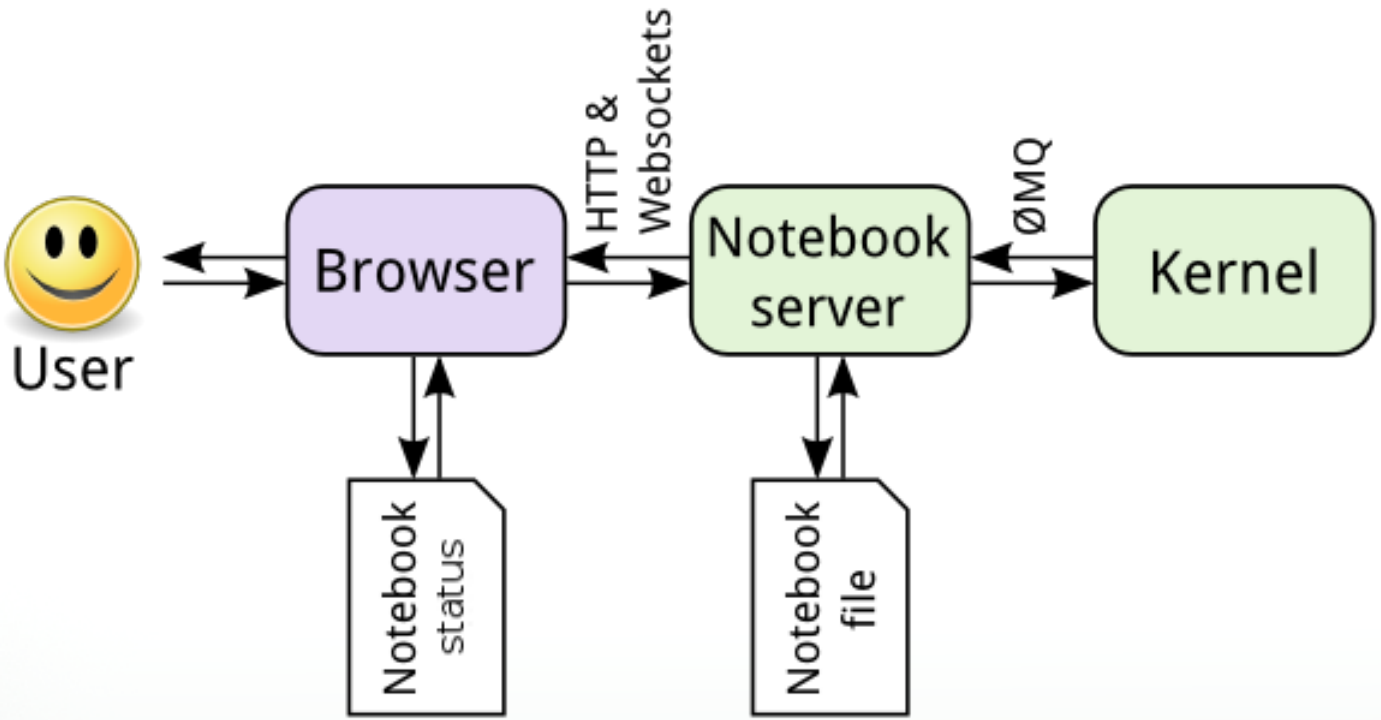

>>> from
coffee
import *

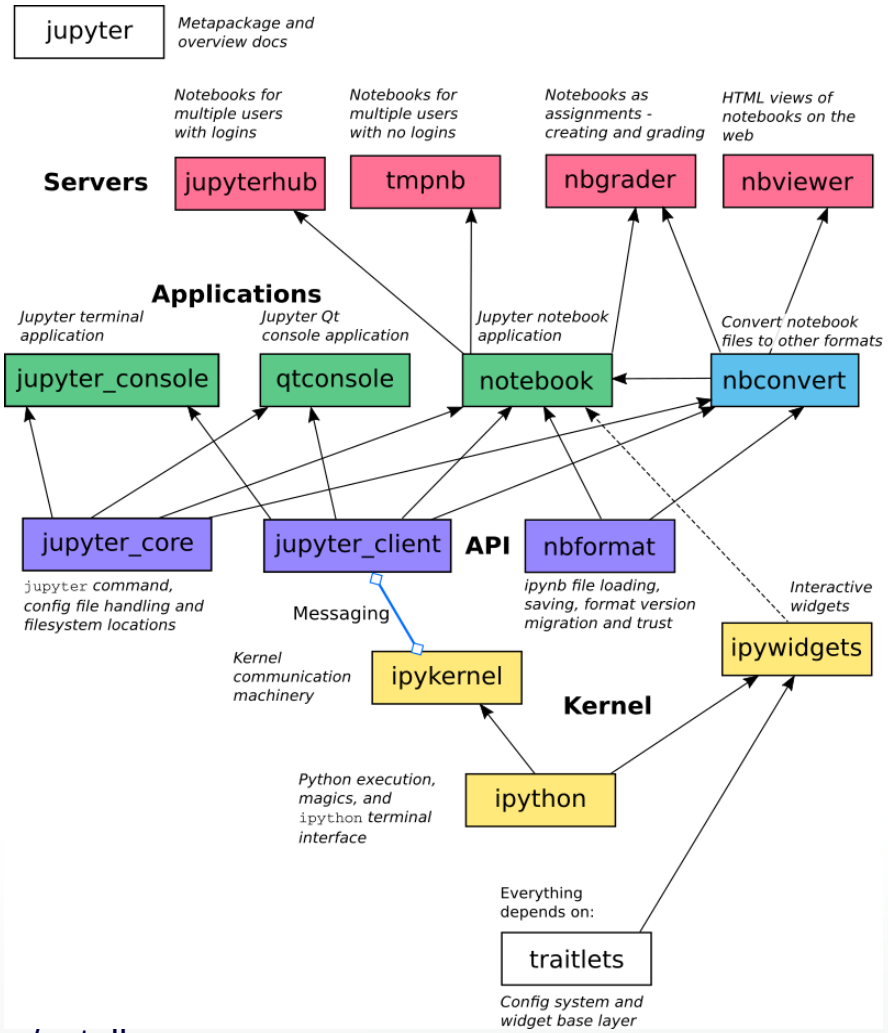


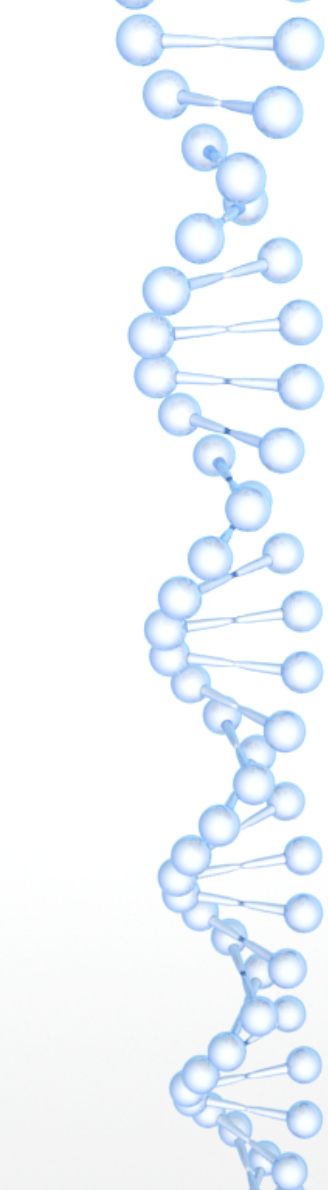
Content of this talk

- Jupyter architecture overview
- Running a notebook remotly
- JupyterHub solution
- Alternative solution using remote desktops
- Pros and cons
- Future

Jupyter architecture overview




jupyter
Metapackage and overview docs
Notebooks for multiple users with logins
Notebooks for multiple users with no logins
Notebooks as assignments - creating and grading
HTML views of notebooks on the web
Servers
jupyterhub
tmpnb
nbgrader
nbviewer
Applications
Jupyter terminal application
Jupyter Qt console application
Jupyter notebook application
Convert notebook files to other formats
jupyter_console
qtconsole
notebook
nbconvert
jupyter_core
jupyter_client
API
nbformat
jupyter command, config file handling and filesystem locations
Messaging
ipynb file loading, saving, format version migration and trust
Interactive widgets
ipywidgets
Kernel communication machinery
ipykernel
Kernel
Python execution, magics, and ipython terminal interface
ipython
Everything depends on:
traitlets
Config system and widget base layer



Running a notebook remotly

- Proxy requests
- Authenticate the users
- Central managment
- Share with other users
- Collaborative notebook editing
- Reconnect to a running notebook

is it possible to make a jupyter notebook run even if the page is closed? #2446

Open **laurentperrinet** opened this issue 24 days ago · 1 comment

laurentperrinet commented 24 days ago

I am running multiple notebooks at the same time, often in different browsers, sometimes on different remote clients. till now, when I close the tab corresponding to a running notebook, it warns that the corresponding run will be stopped.

My question: How do I make a jupyter notebook resume it's run even if the page is closed ?

From what I understand, the client-server architecture could make that possible, but that there may be issues with multiple concurrent runs...

PS: I also asked the question on SO and will make sure to make this issue updated if I get solutions.

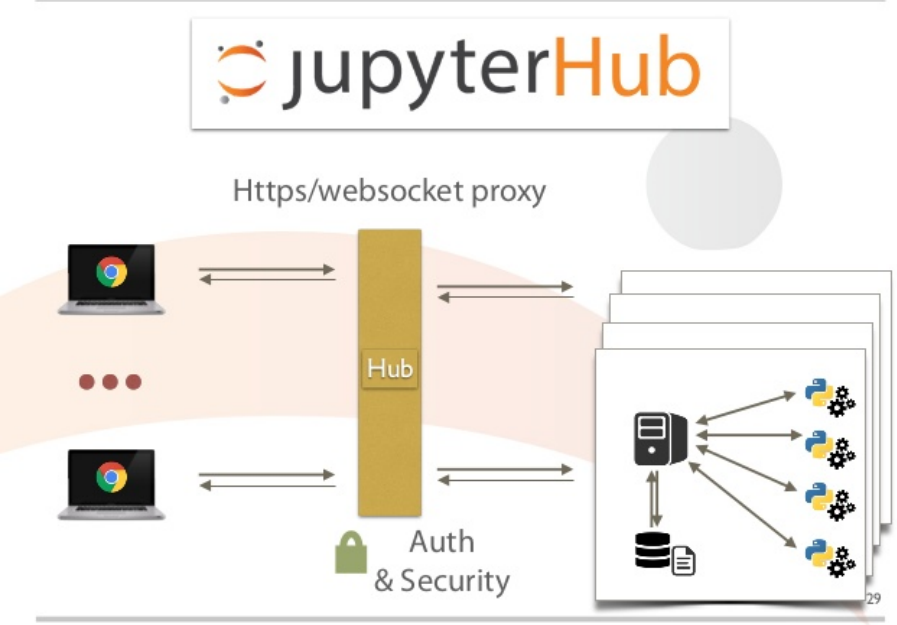
takluyver commented 24 days ago Owner

Anything already running in the notebook will keep running, and the kernel it started for that will stay running - so it won't lose your variables. However, any output produced while the notebook isn't open in a browser tab is lost; there isn't an easy way to change this until we have the notebook server able to track the document state, which has been on the plan for ages.

flying-sheep referenced this issue 3 days ago

keep notebook running after the browser tab closed #1647 Closed

JupyterHub solution

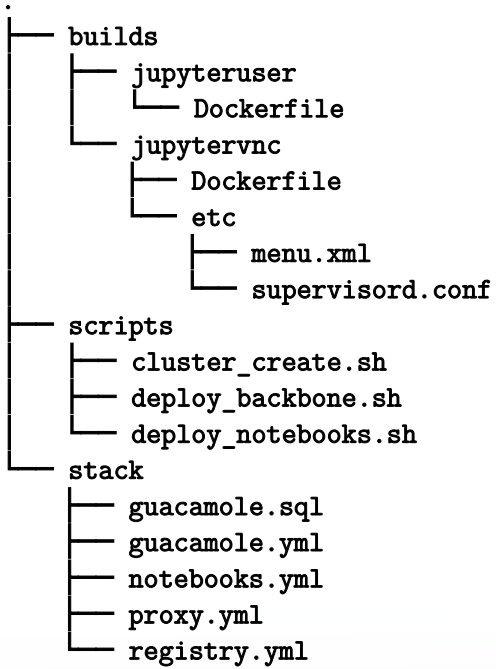


Alternative solution

Technologies used:

- Docker swarm cluster
- VNC server + Firefox + Jupyter
- Apache Guacamole *(clientless remote desktop gateway)*

Alternative solution



Docker cluster

```
# Create 3 machines
for i in 1 2 3; do
    docker-machine create -d virtualbox node-$i
done
eval $(docker-machine env node-1)

# Initialize docker manager
docker swarm init --advertise-addr $(docker-machine ip node-1)

# Join workers to the cluster
TOKEN=$(docker swarm join-token -q worker)
for i in 2 3; do
  eval $(docker-machine env node-$i)
  docker swarm join --token $TOKEN --advertise-addr $(docker-machine ip node-$i) \
  $(docker-machine ip node-1):2377
done
```

Deploy backbone services

```
# Create proxy and notebooks networks

docker network create --driver overlay proxy

docker network create --driver overlay notebooks


# Start proxy, registry and guacamole

docker stack deploy -c ../stack/proxy.yml proxy

docker stack deploy -c ../stack/guacamole.yml guacamole

docker stack deploy -c ../stack/registry.yml registry
```

Build and push images

- Common image (builds/jupytervnc/Dockerfile)

```
$ docker build -t localhost:5000/jupytervnc:5.0 .
$ docker push localhost:5000/jupytervnc:5.0
```

```
FROM debian:jessie-slim
RUN apt-get -y update \
 && apt-get install -y --no-install-recommends x11vnc xvfb supervisor openbox firefox-esr wget bzip2 ca-certificates \
...
# Configure environment
ENV CONDA_DIR /opt/conda
...
COPY etc/supervisord.conf /etc/supervisor/supervisord.conf
COPY etc/menu.xml /etc/xdg/openbox/menu.xml
...
RUN wget --quiet https://repo.continuum.io/miniconda/Miniconda3-4.3.11-Linux-x86_64.sh && \
    echo "b9fe70ce7b6fa8df05abfb56995959b897d0365299f5046063bc236843474fb8 *Miniconda3-4.3.11-Linux-x86_64.sh" | sha256sum -c - && \
…
USER $BG_USER
EXPOSE 5900
WORKDIR /home/$BG_USER
CMD ["/usr/bin/supervisord"]
```

Build and push images

- User image (builds/jupyteruser/Dockerfile)

```
$ docker build -t localhost:5000/jupyteruser:5.0 .
$ docker push localhost:5000/jupyteruser:5.0
```

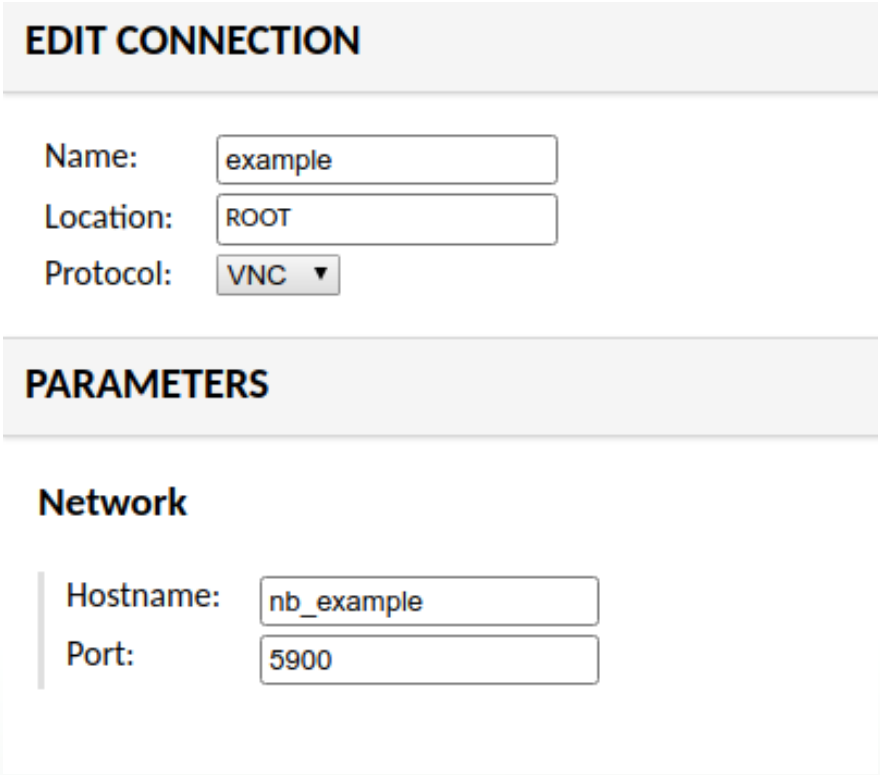
```
FROM localhost:5000/jupytervnc:5.0
ENV BG_NEW_USER username
ENV BG_NEW_UID 1391

# Change user UID
USER root
RUN usermod -u $BG_NEW_UID $BG_USER \
 && usermod -l $BG_NEW_USER $BG_USER \
 && chown -R -h $BG_NEW_USER /var/log/supervisor

USER $BG_NEW_USER
```

Config and start a notebook

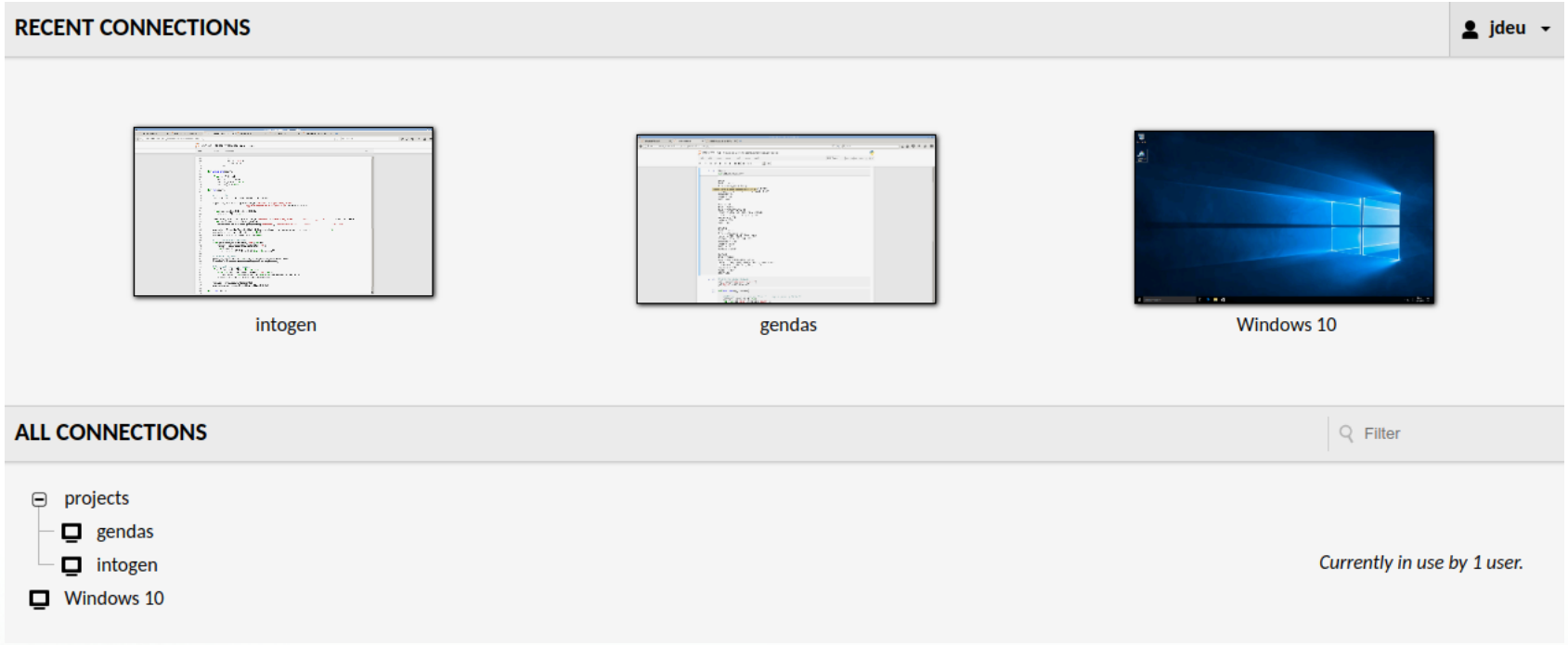
```
$ docker stack deploy -c ../stack/notebooks.yml nb
```

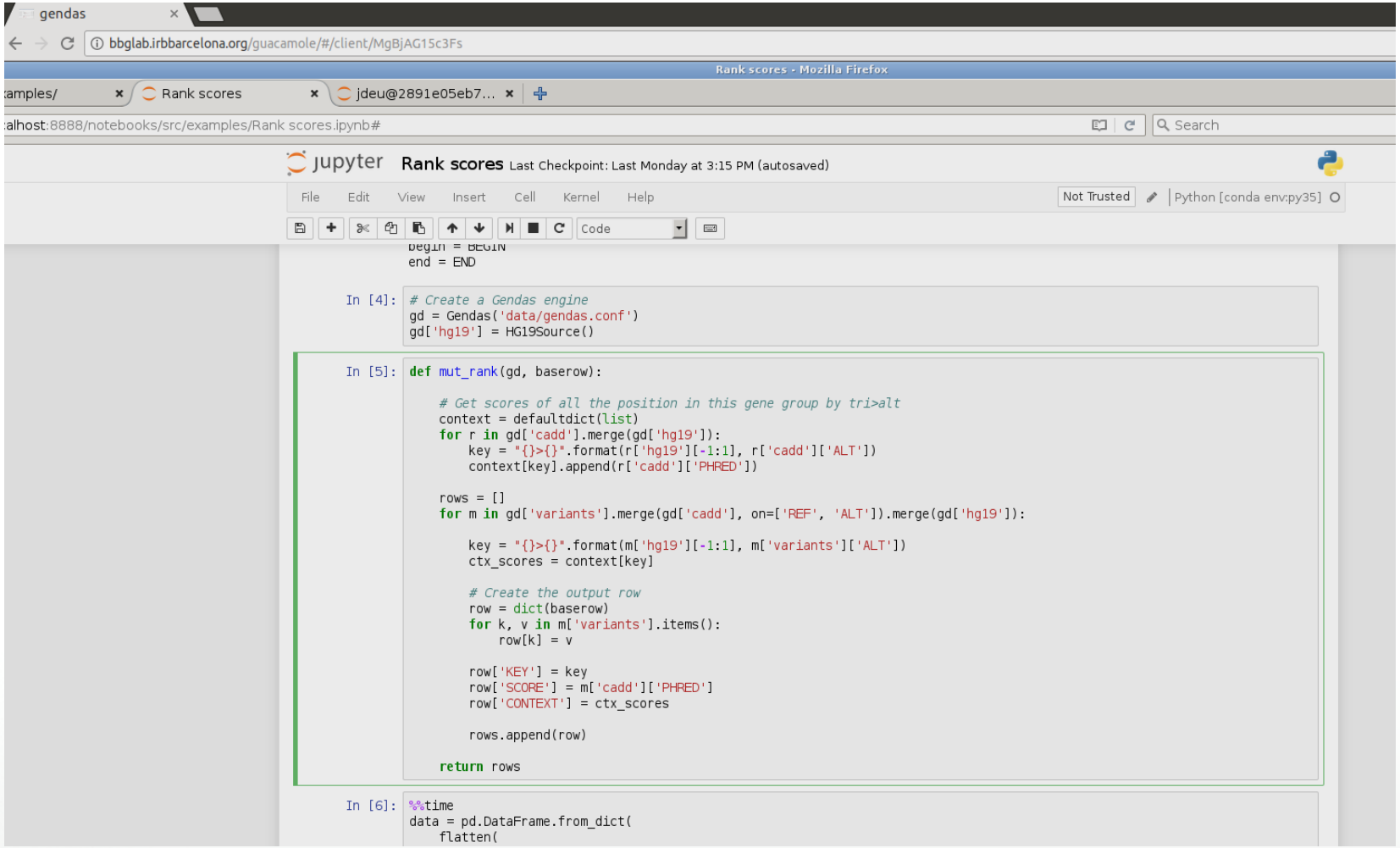
```
version: '3'

services:
  example:
    image: localhost:5000/jupyteruser:5.0
    networks:
      - notebooks
    volumes:
      - ${REPO_HOME}:/workspace
    working_dir: /workspace
    environment:
      - CONDA_ENVS_PATH=${REPO_HOME}/envs
```

Config guacamole connection

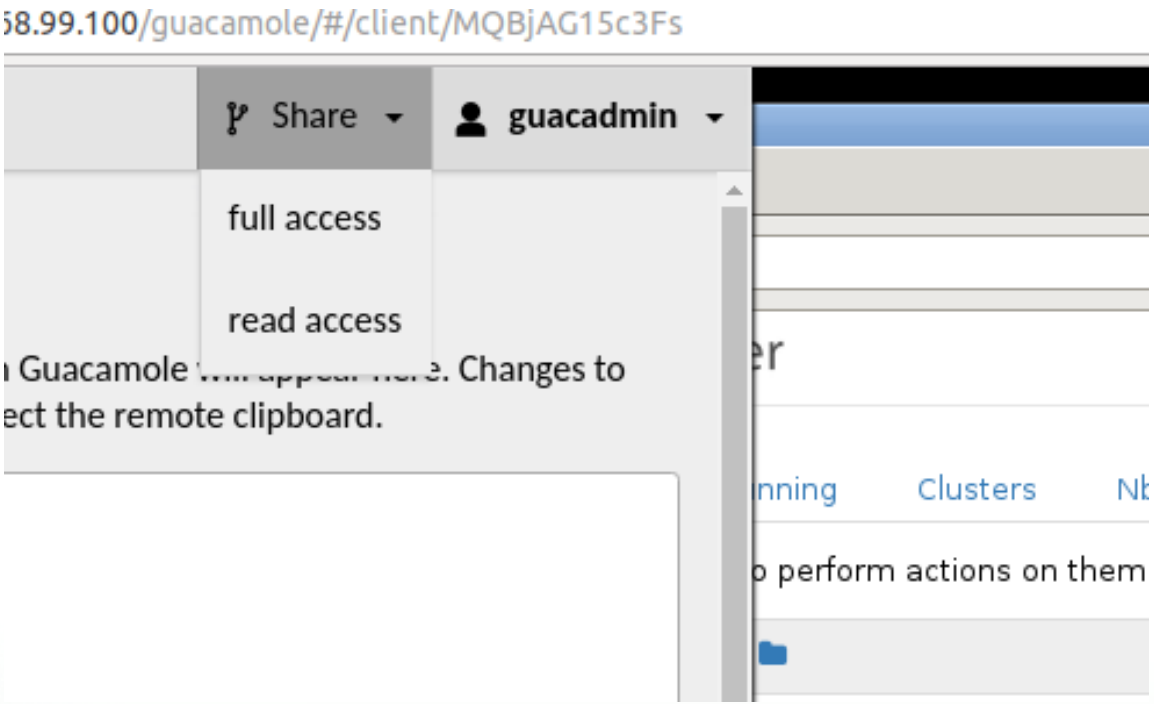


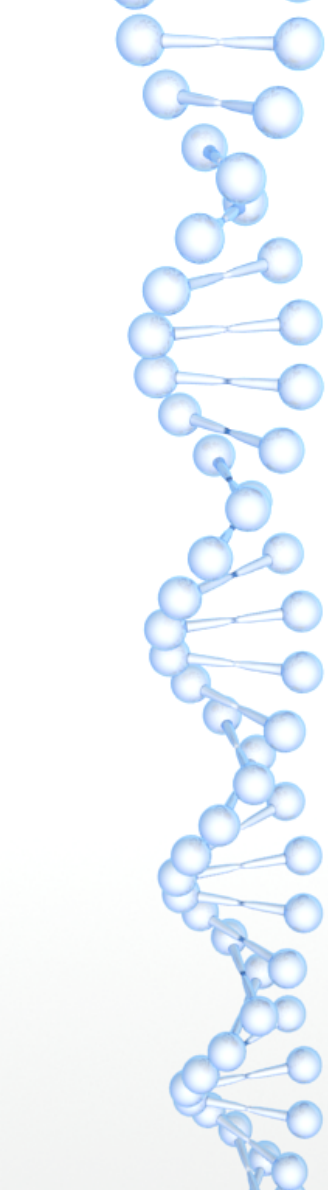
Connect to the notebook




gendas
bbglab.irbbarcelona.org/guacamole/#/client/MgBjAG15c3Fs
Rank scores - Mozilla Firefox
Rank scores
jdeu@2891e05eb7...
alhost:8888/notebooks/src/examples/Rank scores.ipynb#
Search
jupyter
Rank scores
Last Checkpoint: Last Monday at 3:15 PM (autosaved)
File Edit View Insert Cell Kernel Help
Not Trusted
Python [conda env:py35]
Code
begin = BEGIN
end = END
In [4]:
# Create a Gendas engine
gd = Gendas('data/gendas.conf')
gd['hg19'] = HG19Source()
In [5]:
def mut_rank(gd, baserow):
# Get scores of all the position in this gene group by tri>alt
context = defaultdict(list)
for r in gd['cadd'].merge(gd['hg19']):
key = "{}>{}".format(r['hg19'][-1:1], r['cadd']['ALT'])
context[key].append(r['cadd']['PHRED'])
rows = []
for m in gd['variants'].merge(gd['cadd'], on=['REF', 'ALT']).merge(gd['hg19']):
key = "{}>{}".format(m['hg19'][-1:1], m['variants']['ALT'])
ctx_scores = context[key]
# Create the output row
row = dict(baserow)
for k, v in m['variants'].items():
row[k] = v
row['KEY'] = key
row['SCORE'] = m['cadd']['PHRED']
row['CONTEXT'] = ctx_scores
rows.append(row)
return rows
In [6]:
%%time
data = pd.DataFrame.from_dict(
flatten(

Share connection



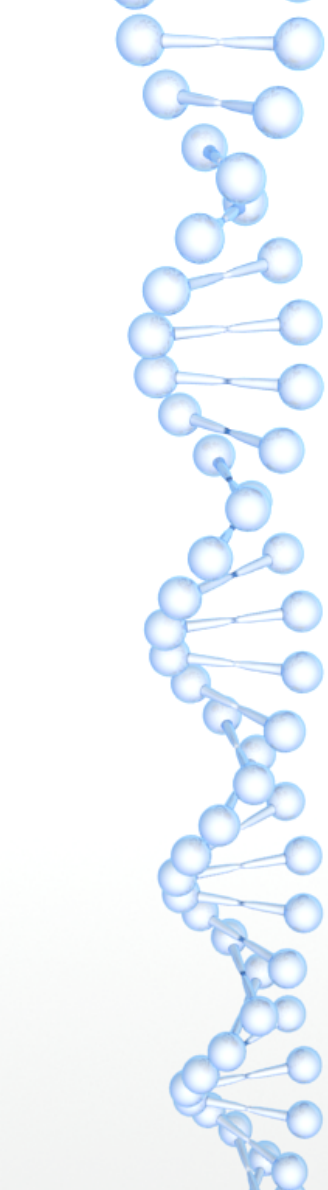


Pros and cons

- **Pros**
  - You can disconnect at any time
  - Concurrent notebook editing
  - Valid for any tool
- **Cons**
  - Indirect copy&paste
  - Scrolling less responsive
  - Different look&feel

Future: to do list

- Add a GUI to notebook creation
- Integrate the solution in JupyterHub
- Or add a HTTP/Websockets proxy protocol in Apache Guacamole
- Create a Jupyter Lab image



Future: collaborative editing

Brian E. Granger, 6 Feb 2017

Yes, we are doing this work in JupyterLab. We are building these features in a manner that will initially support Google Drive Real Time API, but could also have other real time backends plugged into it.

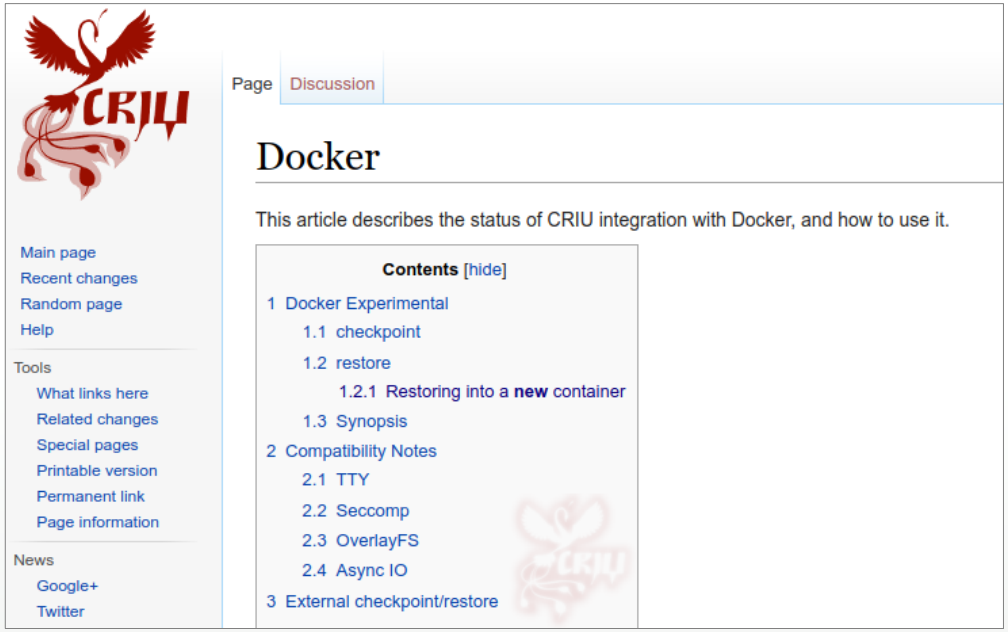
We have a full time post doc at UC Berkeley, Ian Rose, working on these things. The real time stuff isn't quite ready for public usage, but it is moving pretty fast at this point.

https://github.com/ian-r-rose/jupyterlab-google-drive

Future: notebook hibernation

docker checkpoint create …

(already available with experimental enable)



github.com/jordeu/pytalks