Jupyter Notebooks for Performing & Sharing Bioinformatics Analyses

Jonathan Dursi, OICR
jonathan@dursi.ca

https://github.com/ljdursi/glbio-jupyter-workshop

GLBIO/CCBC GREAT LAKES BIOINFORMATICS AND CANADIAN COMPUTATIONAL BIOLOGY CONFERENCE

OICR
Ontario Institute
for Cancer Research

The plan for this morning

- Run Jupyter notebooks on your laptop, in R and Python

- How to perform interactive analyses in a web browser using Jupiter

- How does it all work?

- Using markdown and latex to format notebooks nicely

- "Port" an R bioinformatics workflow from scripts into a Jupyter notebook

- Sharing a Jupyter notebook online:

  - SageMathCloud

  - GitHub and git; nbviewer

  - mybinder.org

OICR
Ontario Institute
for Cancer Research

The plan for this morning

Perfectly reasonable and interesting things to discuss that we *won't* be talking about today:

- Combining R and Python: cool and possible but requires a little setup
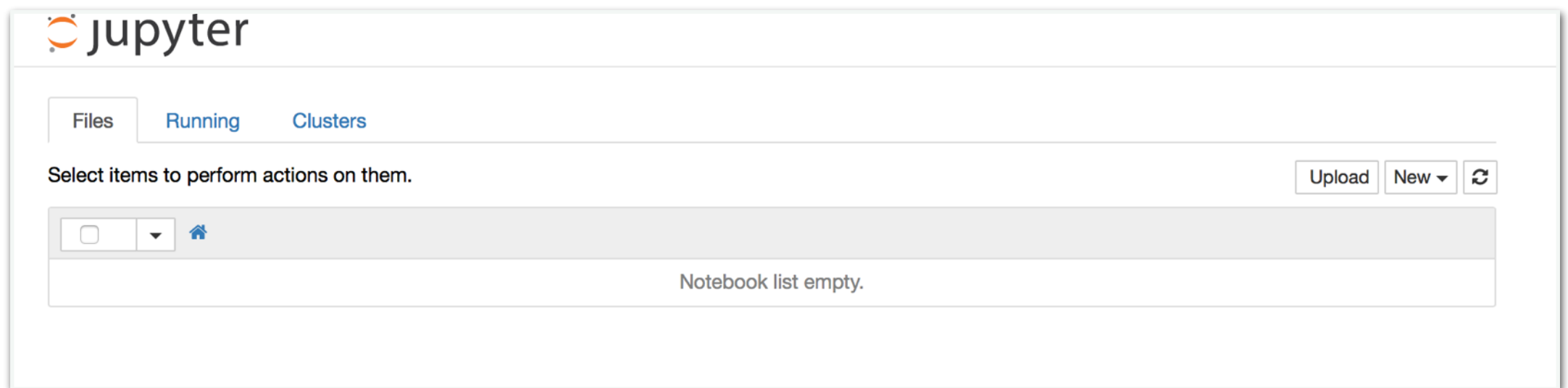
- IPython magics (probably)

Let's start up a notebook

- Make a Notebooks directory and start from there

- Command Line: Linux, Mac

Let's start up a notebook

- Make a Notebooks directory and start from there
- Command Line: Linux, Mac
- Browser should open (or go to localhost:**8888**)

Let's start up a notebook

- Or you can start from home (eg, launch from start menu: Win) and navigate folders, and create new folders:
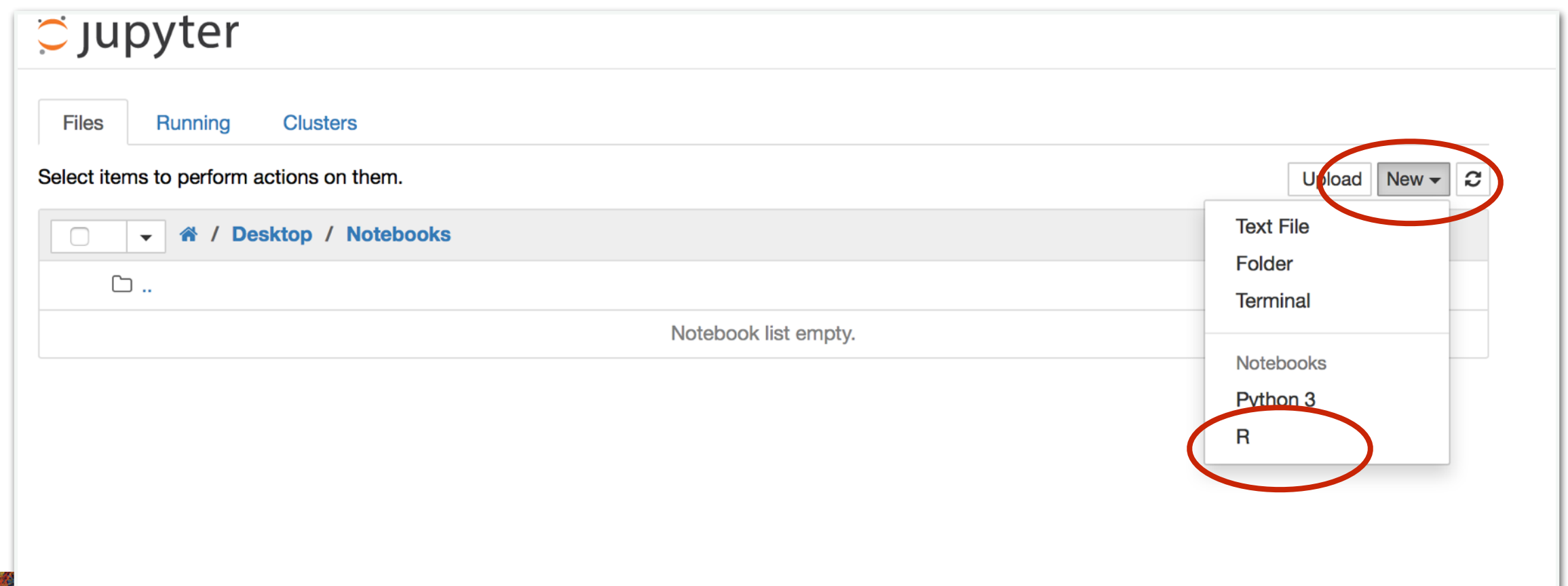
Let's start up a notebook

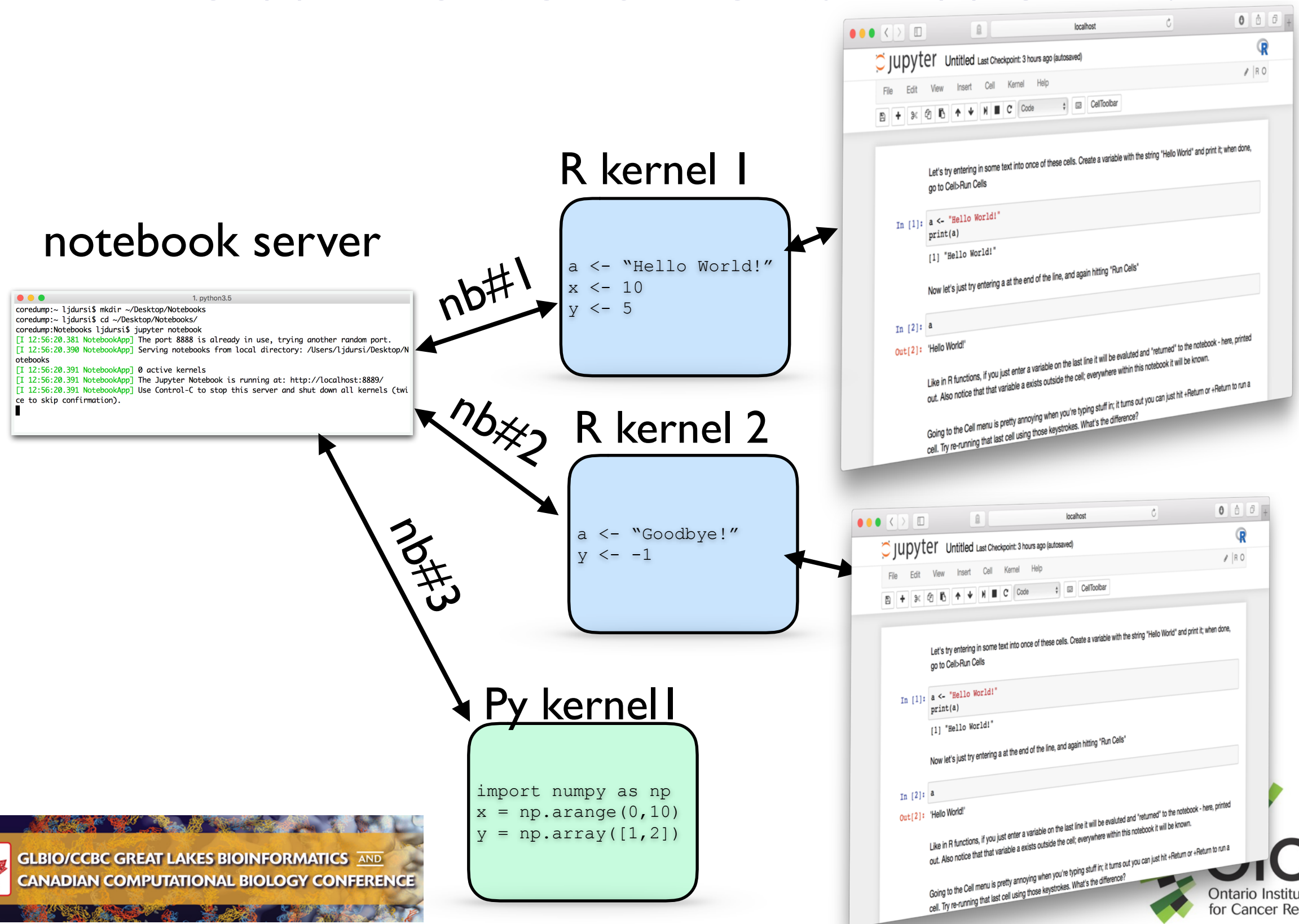- Let's start up an R notebook

Demo - starting notebook

How does this all work?

notebook server

R kernel 1

```
a <- "Hello World!"
x <- 10
y <- 5
```

nb#1

nb#2

R kernel 2

```
a <- "Goodbye!"
y <- -1
```

nb#3

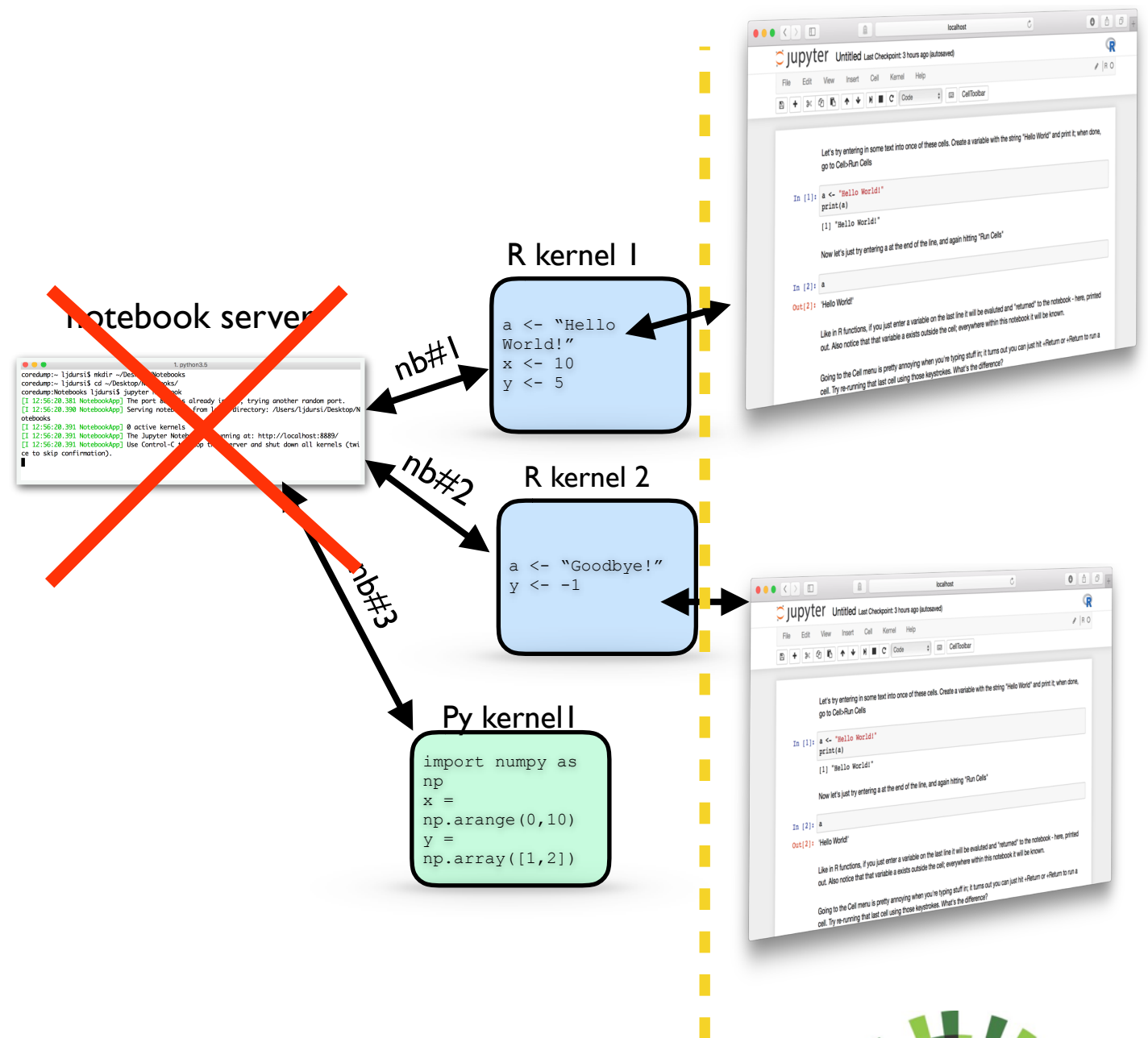**Py kernel 1**

```
import numpy as np
x = np.arange(0,10)
y = np.array([1,2])
```

How does this all work?

- There's no rule that says the notebook server/kernels has to be on same machine as browser!
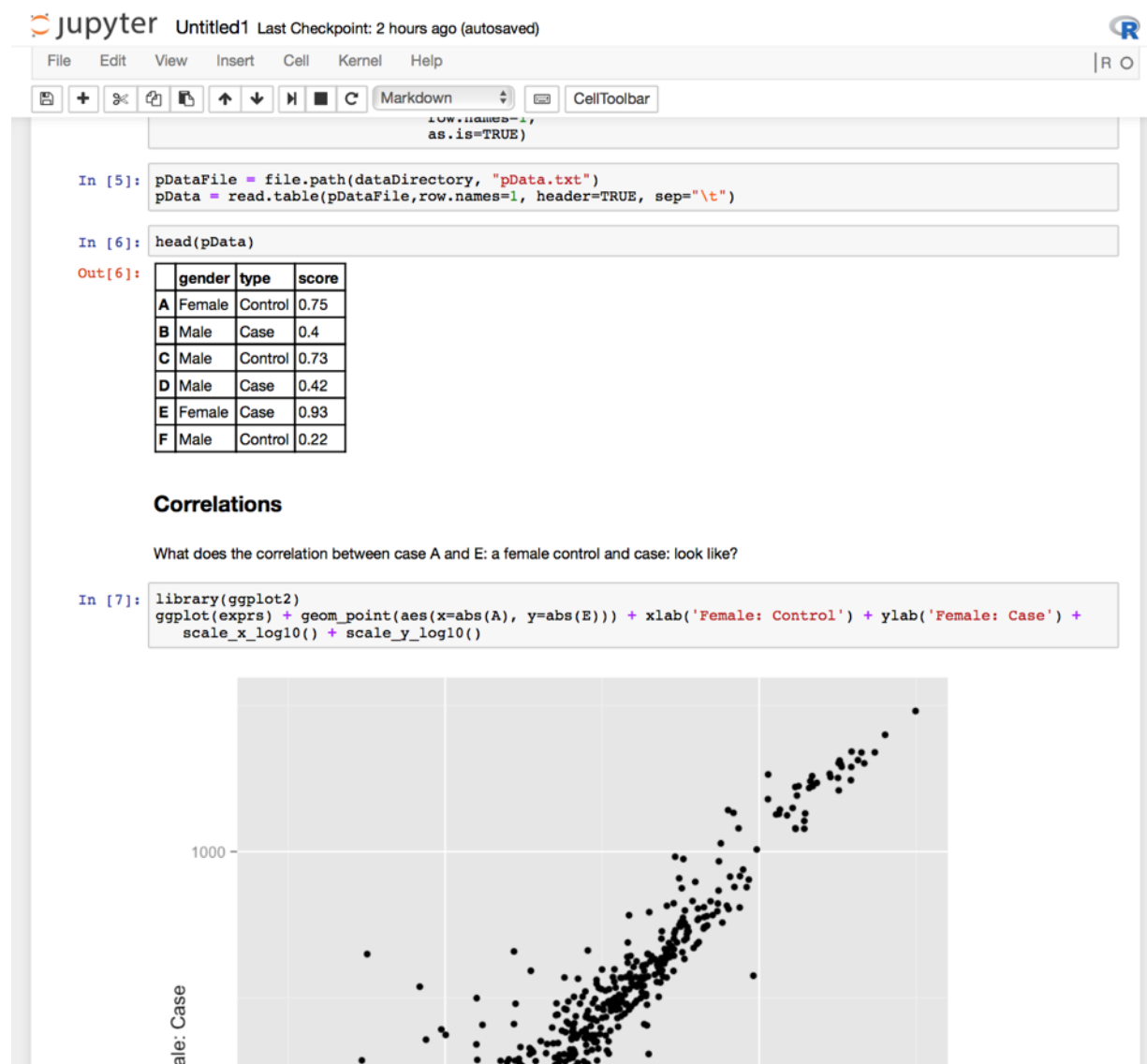
- Can be running on remote server (or cluster)

- AWS demo
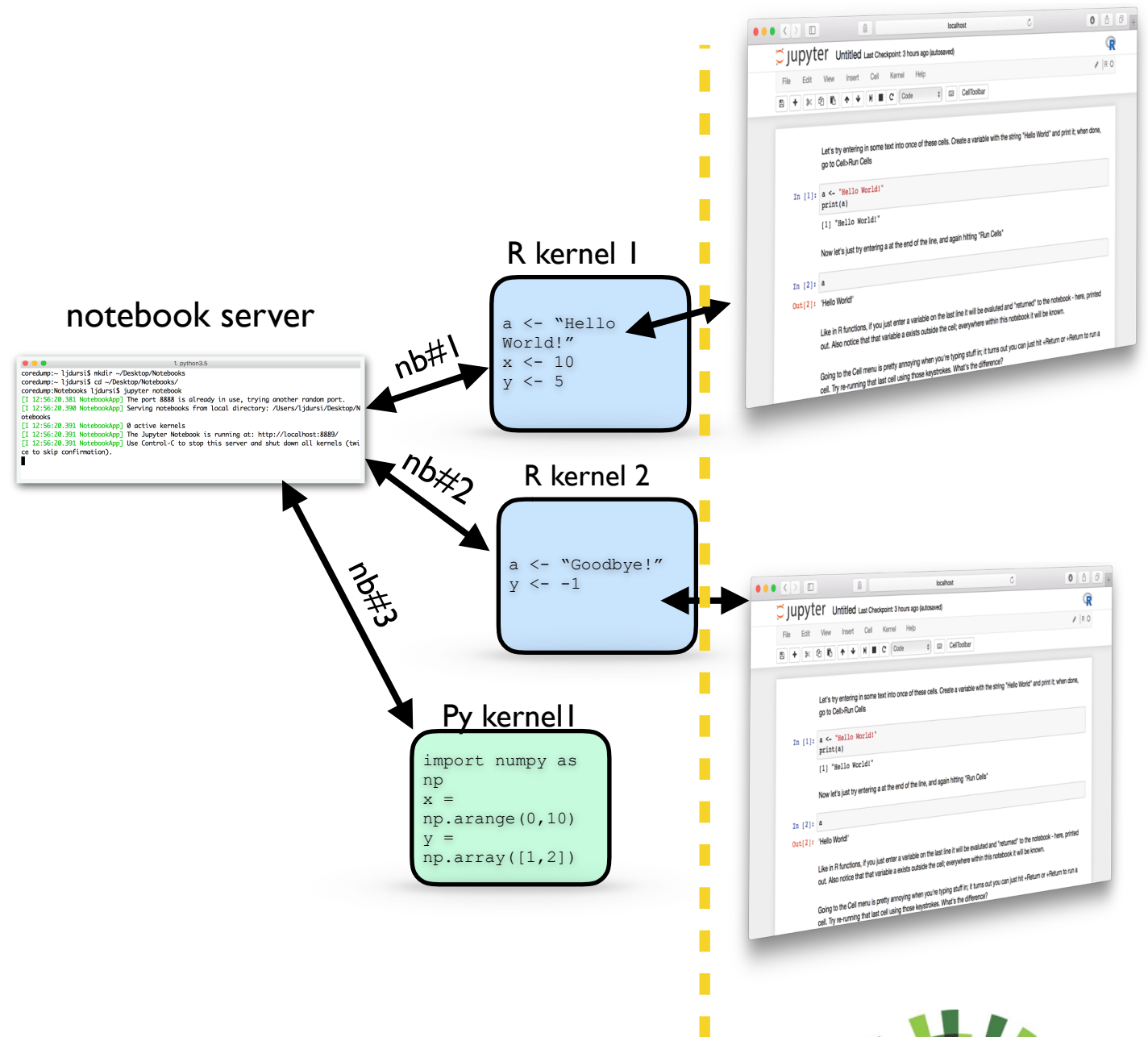
How does this all work?

- AWS demo…
- What happens if the notebook server dies?
- Demo

notebook server

nb#1
nb#2
nb#3

R kernel 1
```
a <- "Hello
World!"
x <- 10
y <- 5
```

R kernel 2
```
a <- "Goodbye!"
y <- -1
```

Py kernel1
```
import numpy as
np
x =
np.arange(0,10)
y =
np.array([1,2])
```

All input+output in ipynb file

- All the input+output is located in the .ipynb file