Astronomical Data Access in the Era of Scientific Cloud Computing

Matias Carrasco Kind
Senior Research Scientist, NCSA

LIneA Webinar
July 5th, 2018

Outline

- What does Data Access mean?
- Scientific Platforms and Gateways
- The Notebook revolution
- Scientific Cloud computing
- Containerization
- Kubernetes
- Applications

What is a Data Release?

Data Products

Interfaces

Documentation

Support

What is a Data Release?

Data Products

Preparation
Vetting
Checks
Consistency
Integrity
Redundancy
Data Model
Storage
Backups
Recovery
Hardware

Interfaces

Development
Version control
Licenses
Data Access
Languages
Sustainability
Guidelines
Scalability
Deployment
Hardware
Maintenance

Documentation

Papers
Web
Code
Data Model
Data Access
Data Format
Guidelines
Accessible
Maintenance
Contributions

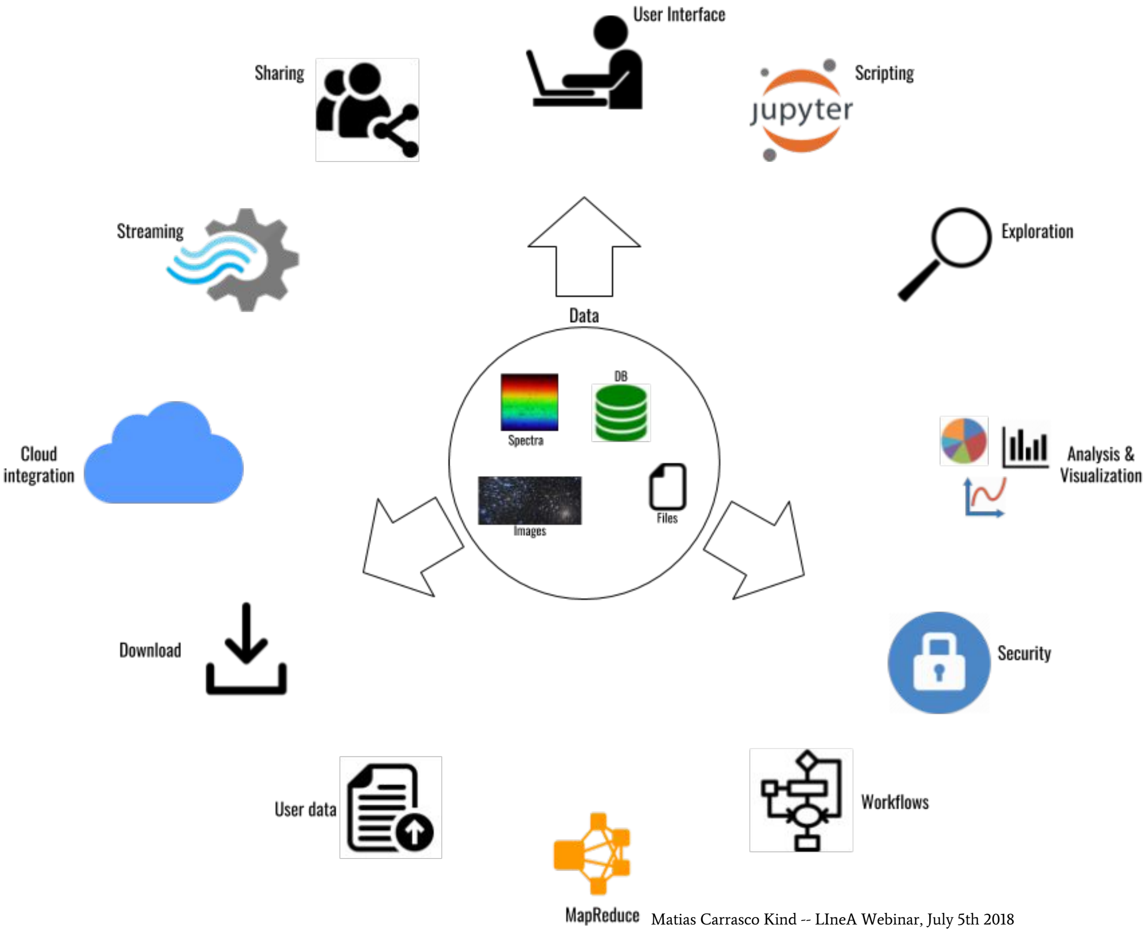
Support

Short Term
Long Term
Forum
Help
Understanding
Deployment
Privacy
Maintenance
Focused
Distributed

What is a Data Release?

Data Products	Preparation Vetting Checks Consistency Integrity Redundancy Data Model Storage Backups Recovery Hardware	**Interfaces**	Development Version control Licenses Data Access Languages Sustainability Guidelines Scalability Deployment Hardware Maintenance
Documentation	Papers Web Code Data Model Data Access Data Format Guidelines Accessible Maintenance Contributions	Support	Short Term Long Term Forum Help Understanding Deployment Privacy Maintenance Focused Distributed

What is Data Access?

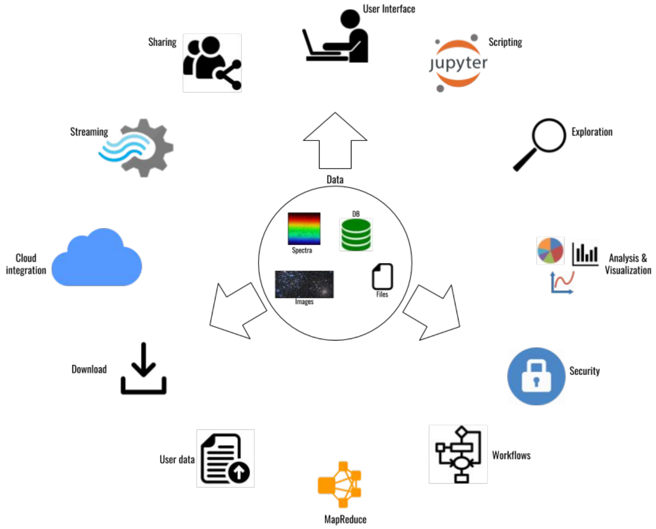


Several meanings around a central data repository with common components

- Storage
- Security
- Retrieving
- Interacting
- Modifying
- Understanding

Scientific Platforms and Gateways

... and many of these concepts are also associated with Scientific Platforms and Gateways (and Science portals, Science servers, etc.)



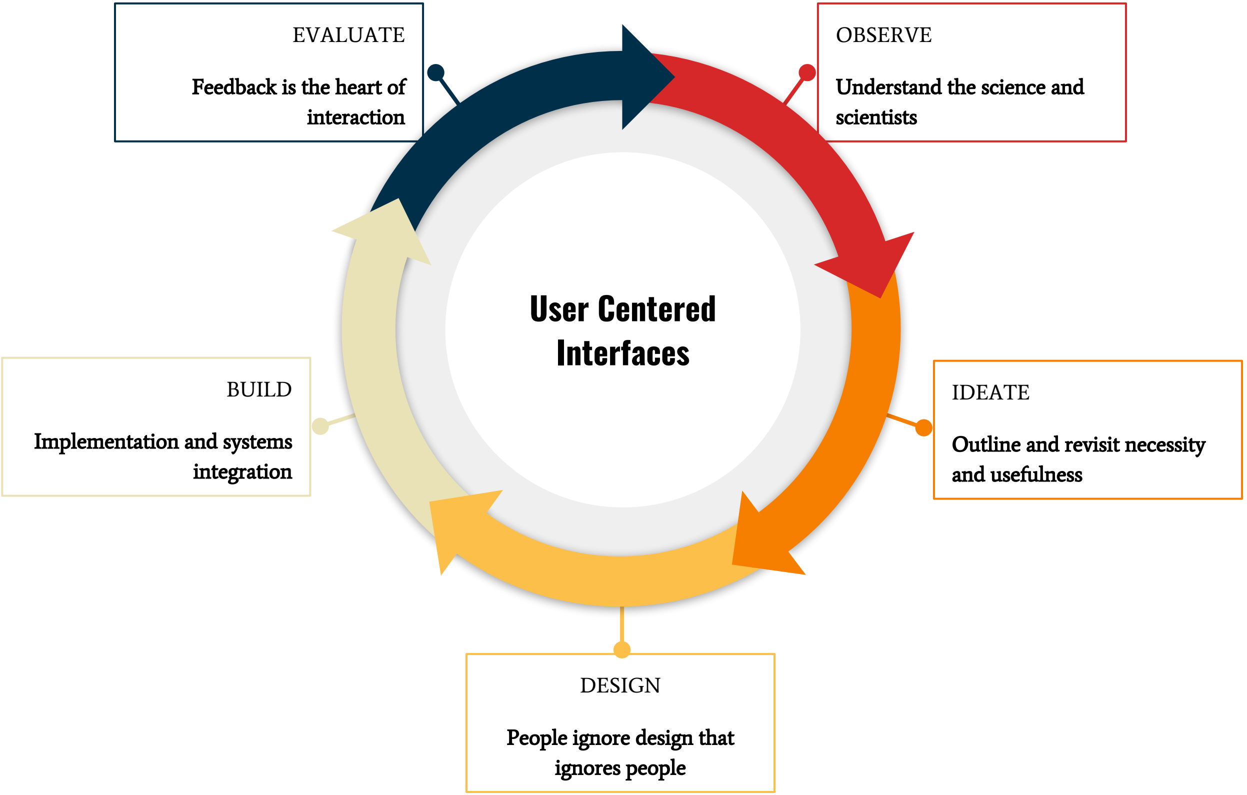
“Science gateways allow science & engineering communities to access shared data, software, computing services, instruments, educational materials, and other resources specific to their disciplines.” (Science Gateways Institute)

“Science gateways is a place to do collaborative scientific related activities” (Me)

User (Scientist) Centered Design

Data Access would not exists without a user interface, but will only succeed if it is user driven.

“… In an ideal world, a user would remember every function after only a single use, but we do not live in idealism. The reality is that familiarity and intuition must be consciously designed into the interface”



The Notebook Revolution

jupyter

Apache Zeppelin

Google Colaboratory

BeakerX

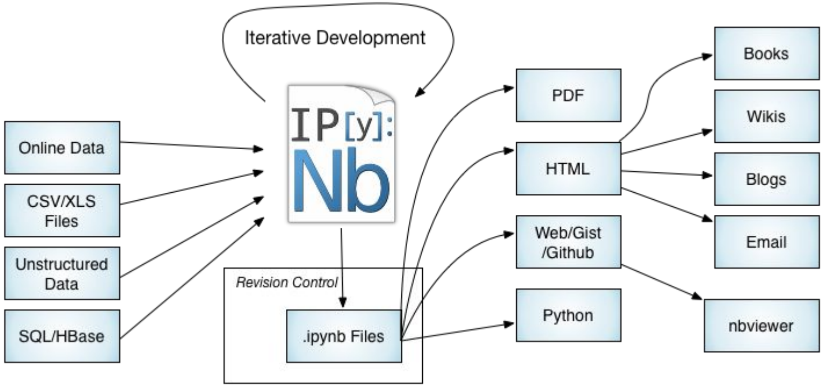
sage free open source mathematics software

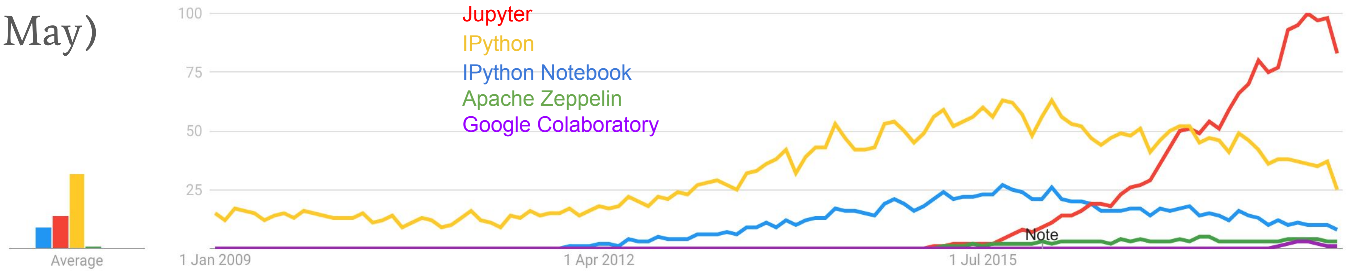
COCALC

9

The Notebook Development

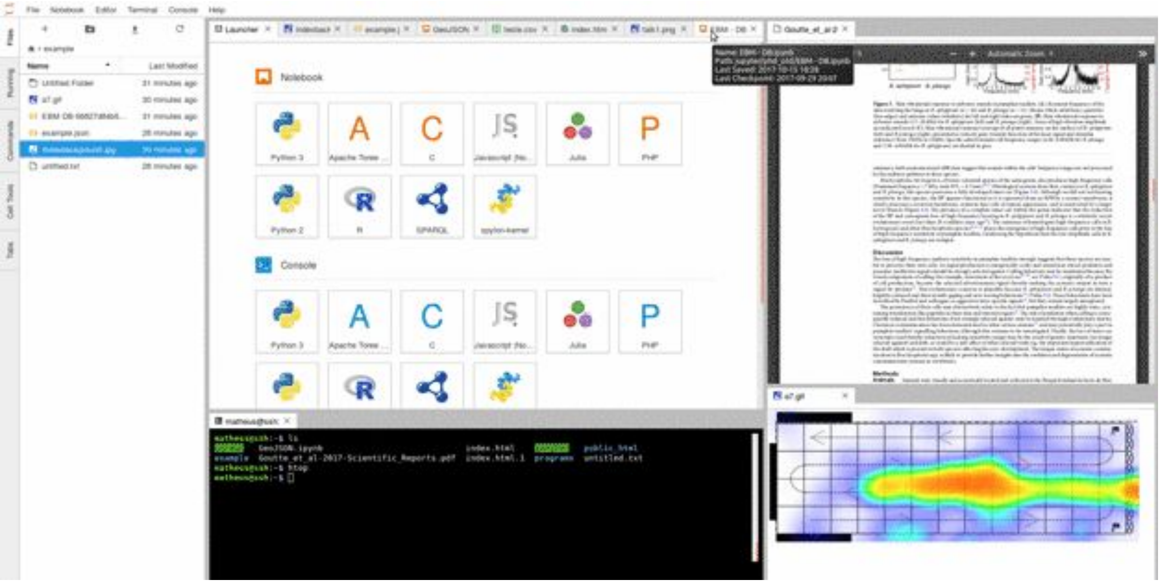
- Started from ideas like Matlab, Maple or Mathematica ~1988
- IPython has been around since 2001
- Sage Notebook released in 2005 (uses IPython)
- IPython Notebook was released in 2011
- IPython Notebook moved to Jupyter in 2014
- Apache Zeppelin created in 2015 (JVM and integrated with Apache Products)
- Beaker Notebook 2015 (moved to BeakerX)
- Google Colaboratory released in Oct 2017 (from ideas back in 2014)
- Cocalc (by SageMath) in 2018
- Jupyter Lab Beta 2.0 (May)





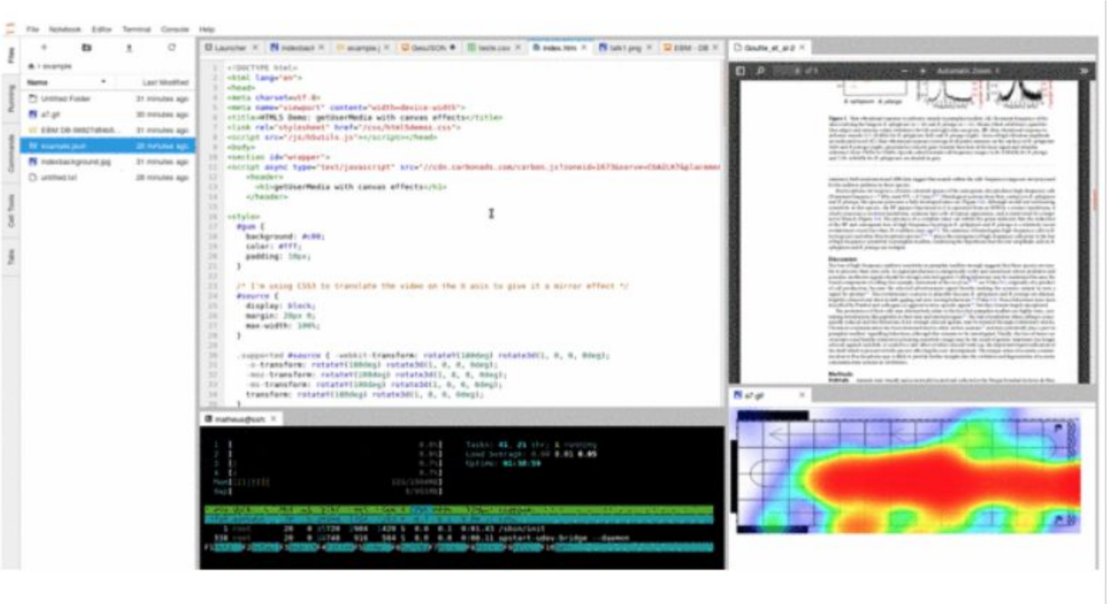
The Jupyter Notebook

- Computational narrative
- Scripting interface
- Scientific oriented interface
- Customizable
- Collaborative
- Adopted by many projects, DES, LSST
- Widgets
- Big Data Integration (Spark
- Interactive plots
- Multiple Kernels (Python, R, Julia, Scala, etc.)



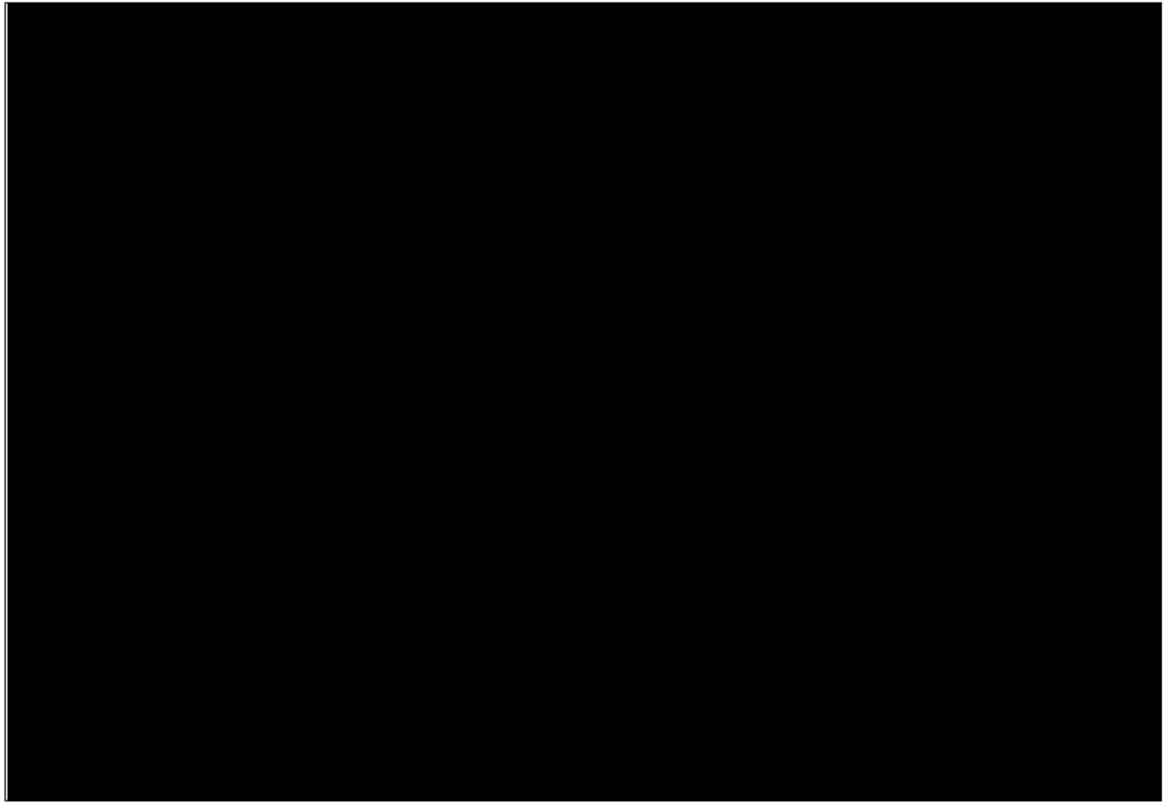
The Jupyter Notebook

- Computational narrative
- Scripting interface
- Scientific oriented interface
- Customizable
- Collaborative
- Adopted by many projects, DES, LSST
- Widgets
- Big Data Integration (Spark
- Interactive plots
- Multiple Kernels (Python, R, Julia, Scala, etc.)



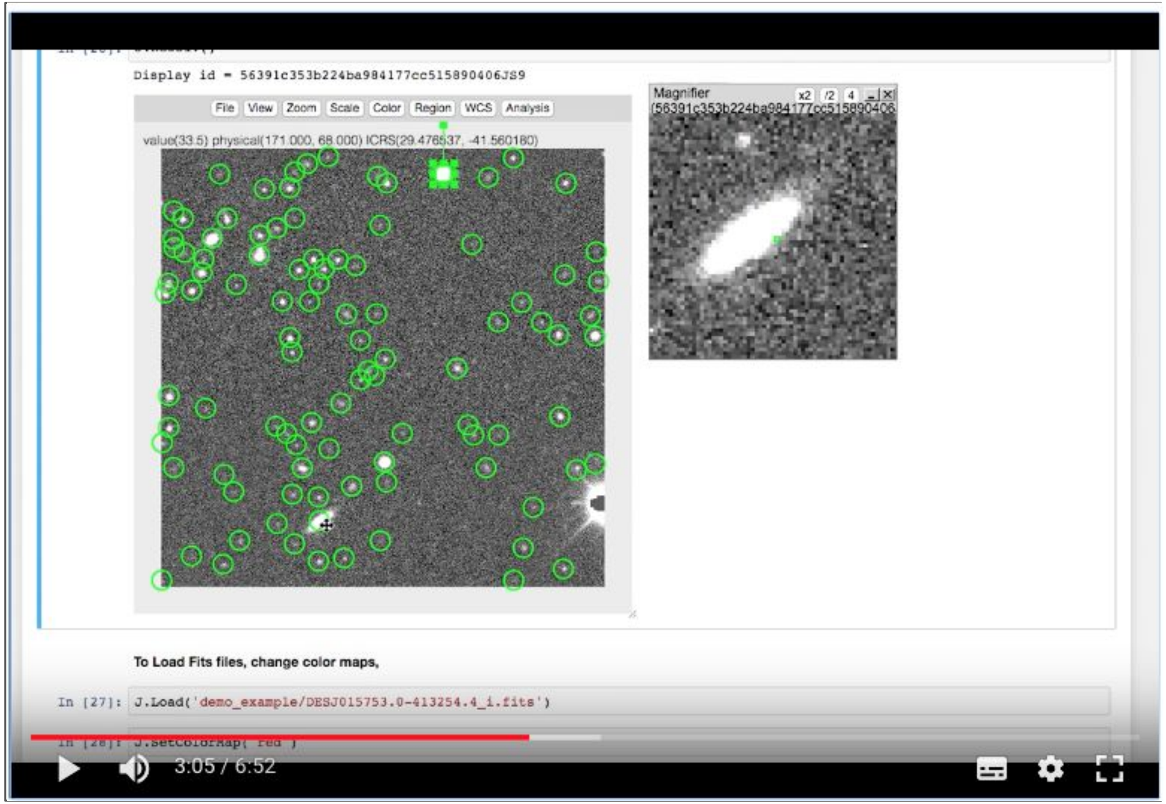
Jupyter in Astronomy

- Becoming standard practice to publish notebooks along with papers, including LIGO results (and many others)
- One of the most common tools used by Astronomers to do analysis
- … and education
- Multi user interface adopted by many projects (DES, LSST, NASA, STScI, NOAO, etc)
- Tools and extensions developed by/for astronomers



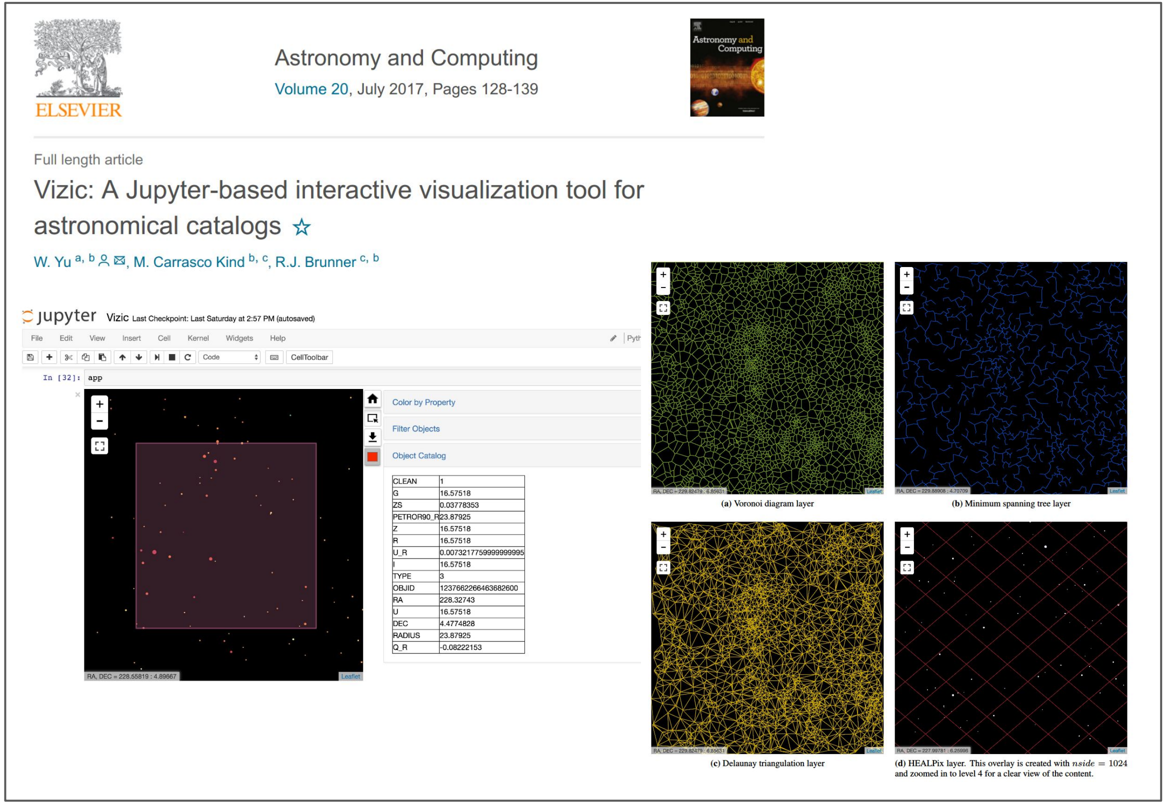
Jupyter in Astronomy

- Becoming standard practice to publish notebooks along with papers, including LIGO results (and many others)
- One of the most common tools used by Astronomers to do analysis
- ... and education
- Multi user interface adopted by many projects (DES, LSST, NASA, STScI, NOAO, etc)
- Tools and extensions developed by/for astronomers



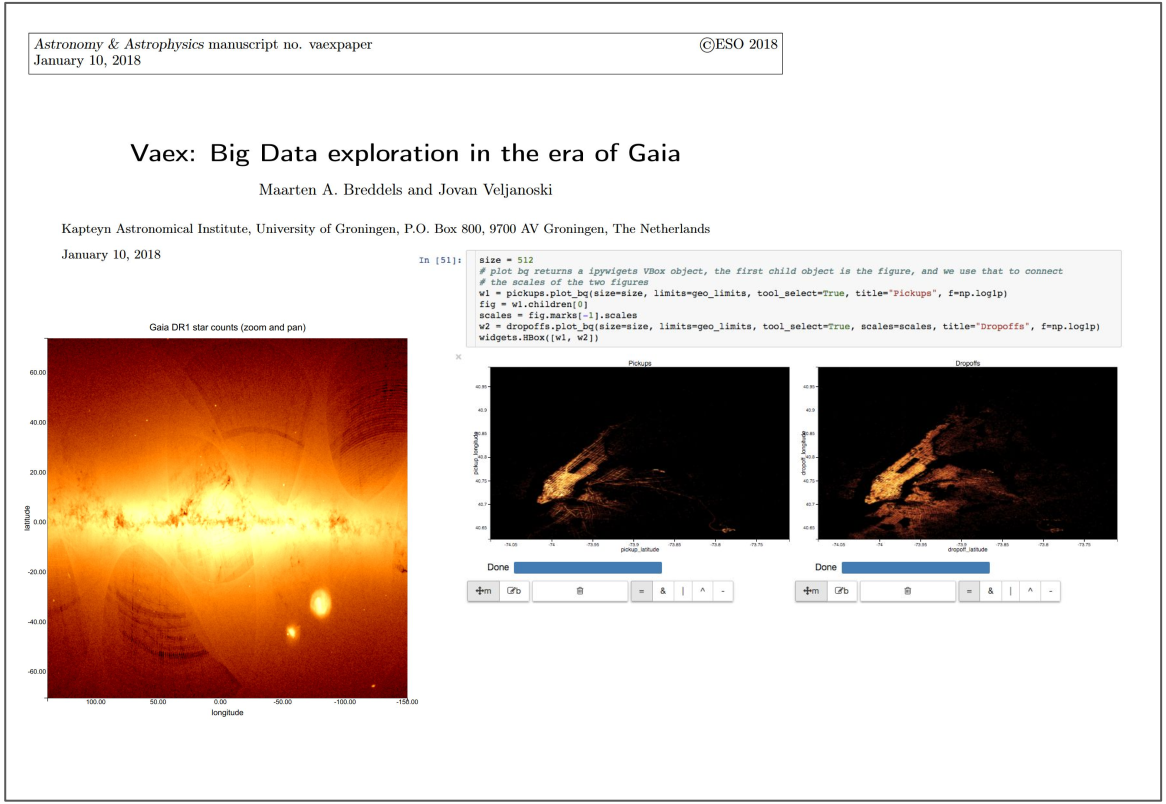
Jupyter in Astronomy

- Becoming standard practice to publish notebooks along with papers, including LIGO results (and many others)
- One of the most common tools used by Astronomers to do analysis
- ... and education
- Multi user interface adopted by many projects (DES, LSST, NASA, STScI, NOAO, etc)
- Tools and extensions developed by/for astronomers



Jupyter in Astronomy

- Becoming standard practice to publish notebooks along with papers, including LIGO results (and many others)
- One of the most common tools used by Astronomers to do analysis
- ... and education
- Multi user interface adopted by many projects (DES, LSST, NASA, STScI, NOAO, etc)
- Tools and extensions developed by/for astronomers


Astronomy & Astrophysics manuscript no. vaexpaper ©ESO 2018
January 10, 2018

Vaex: Big Data exploration in the era of Gaia

Maarten A. Breddels and Jovan Veljanoski

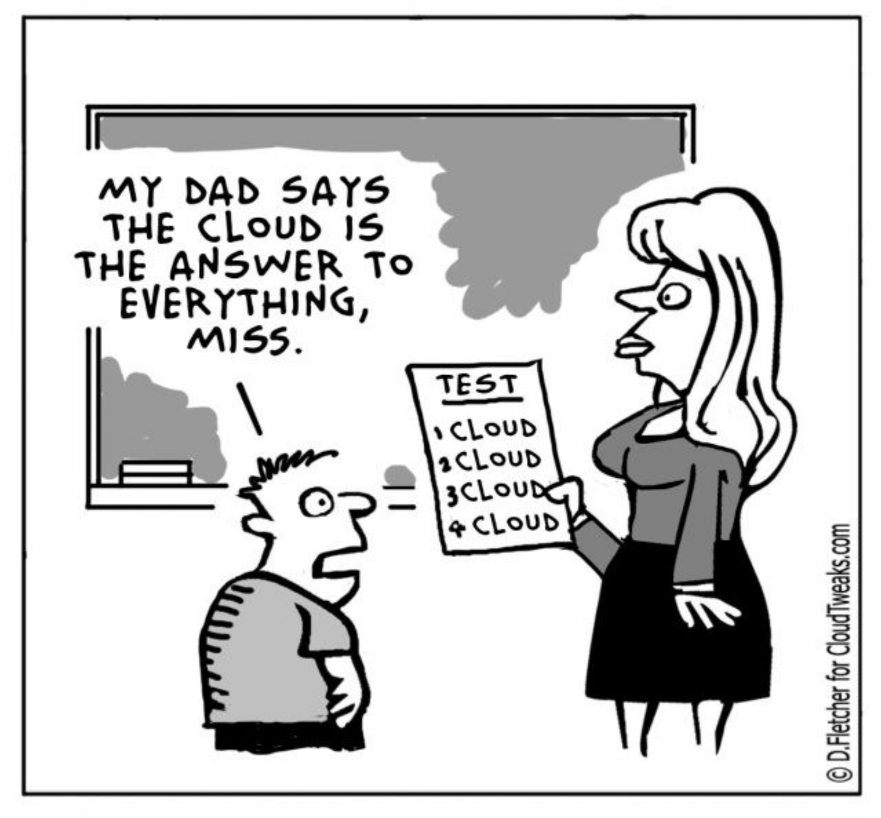
Kapteyn Astronomical Institute, University of Groningen, P.O. Box 800, 9700 AV Groningen, The Netherlands

January 10, 2018

```
In [51]: size = 512
# plot bq returns a ipywigets VBox object, the first child object is the figure, and we use that to connect
# the scales of the two figures
w1 = pickups.plot_bq(size=size, limits=geo_limits, tool_select=True, title="Pickups", f=np.log1p)
fig = w1.children[0]
scales = fig.marks[-1].scales
w2 = dropoffs.plot_bq(size=size, limits=geo_limits, tool_select=True, scales=scales, title="Dropoffs", f=np.log1p)
widgets.HBox([w1, w2])
```

Scientific Cloud Computing

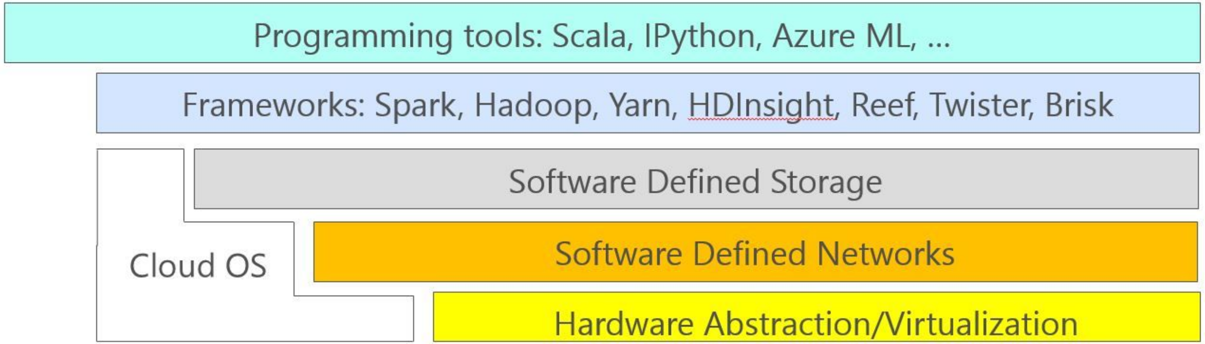
Cloud is about how you do computing, not where you do computing.



Why we should be doing science on the cloud

- Remote and dynamic data (!= Big data)
- Big data ⇒ Data Gravity
- Remote software/server
- Easy to deploy*
- Asynchronous
- Web applications / Shareable
- Serverless applications
- Tablets/ChromeOS
- more...

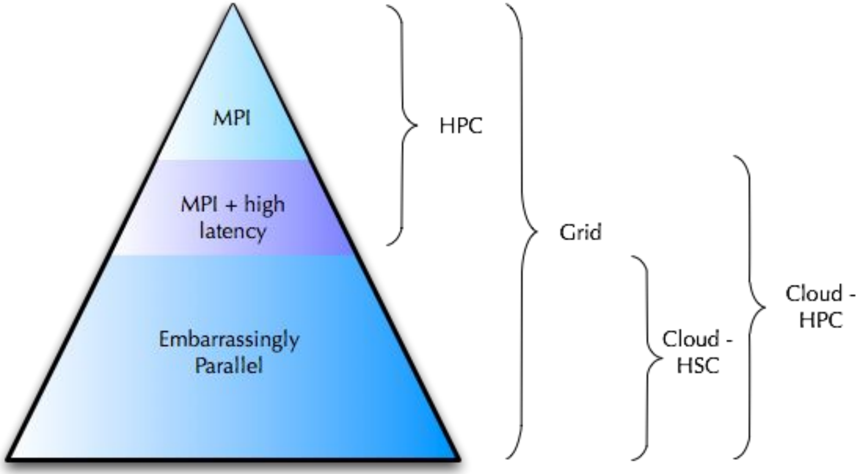
Will we get to have Science as a Service (SCIaaS?)



*arguable

Why we shouldn’t be doing science on the cloud

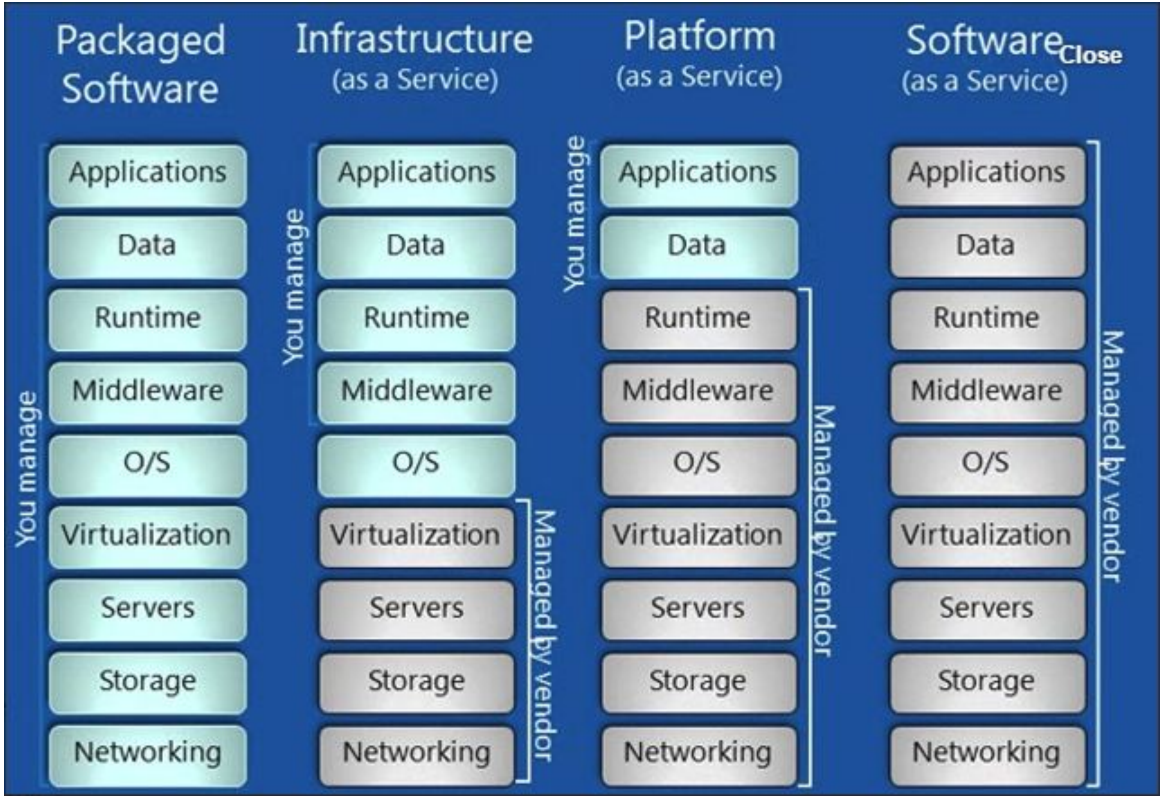
- Because there is no a real reason for it
- HPC is not there yet, large latencies and bad bisection bandwidth
  ... but HPC is adopting cloud technologies
- Full control on data and application
- Security concerns
- Faster development*
- Billing (if a commercial provider)
- more ...



*arguable (CI, CD)

What kind of science?

- HTC vs HPC
- Interactive
- Small projects
- Visualizations
- Short term projects*



*arguable

Which Clouds?

Amazon Web Services (AWS) – 40%

Microsoft Azure – about 50% of AWS

Google Cloud – 3rd place

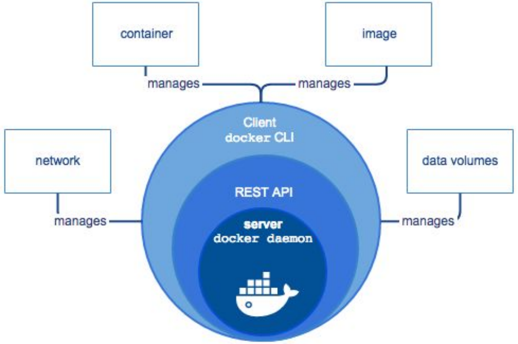
IBM Bluemix – growing fast

Salesforce, DigitalOcean, Rackspace, 1&1, UpCloud, CityCloud, CloudSigma, CloudWatt, Aruba, CloudFerro, Orange, OVH, T-Systems

Cloud for Research: Aristotle, Bionimbus, Jetstream, Chameleon, RedCloud

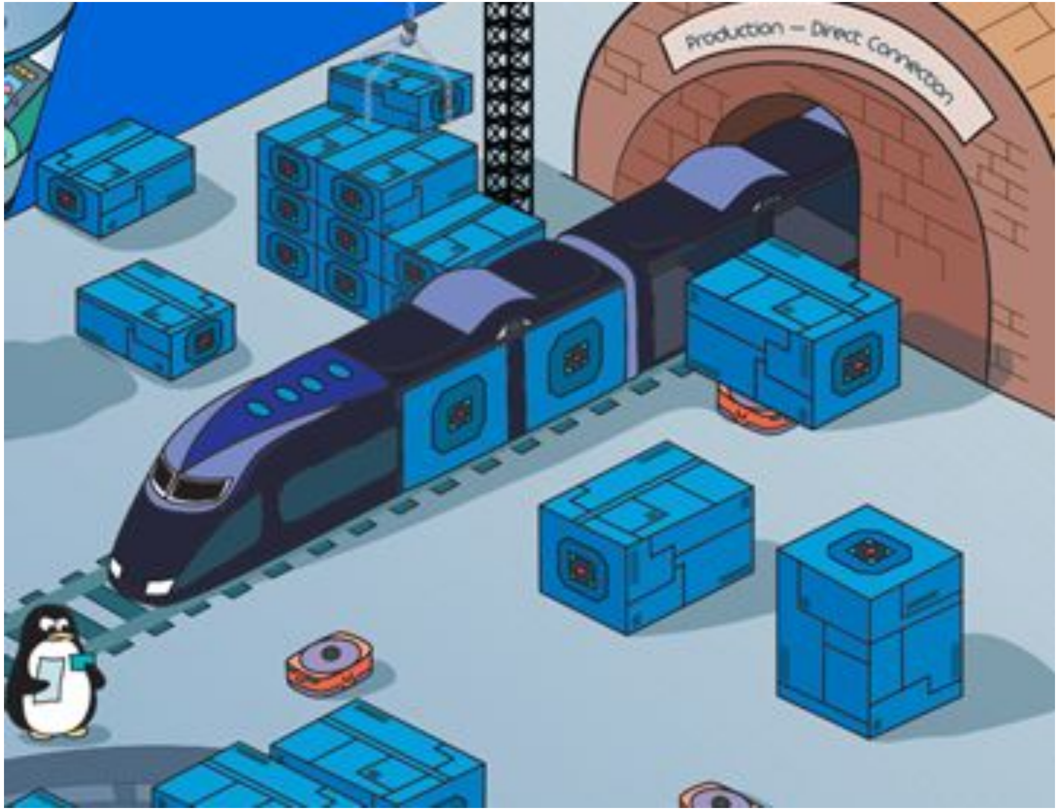
Containerization to the rescue

- It's been around for over 10 years, but popular since 2014 thanks to Docker
- Many other alternatives (rkt, kata, shifter, singularity, etc...)
- Lightweight, stand-alone, executable package of a piece of software that includes everything to run it
- Not just applications
- Software designed storage
- Software designed network



Container organization and orchestration

- We can create a container with an application inside, now what?
- Need to consider:
  - Resource needs
  - Fault tolerant
  - Load balancing
  - Storage management
  - Lifecycle
  - Service Discovery
  - Scalability

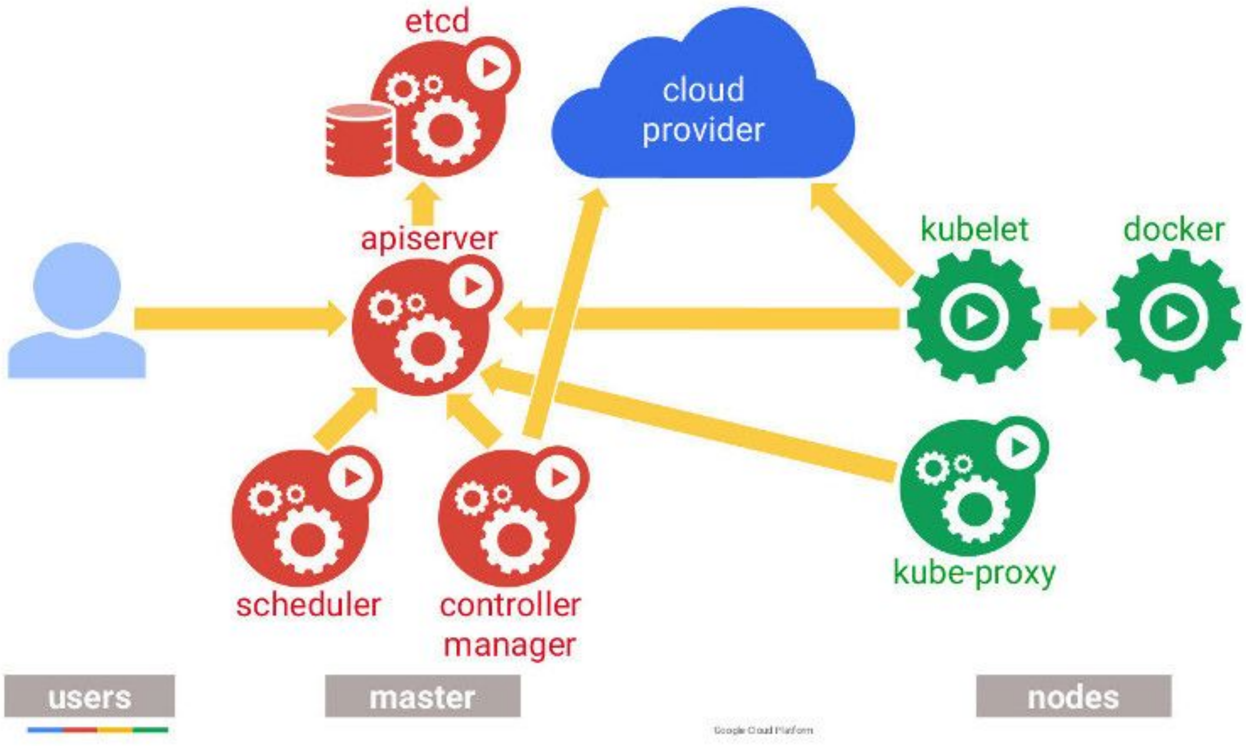


The Kubernetes Factor

- It solves all previous issues and more (not the only one but most popular)
- Open source container management and orchestration platform
- Developed by Google, made open sourced
- One of top 5 most commented open source repositories and #2 in number of pull request
- Standard within all cloud platforms
- Flexible and extensible, customize schedulers
- Is changing the cloud computing paradigm

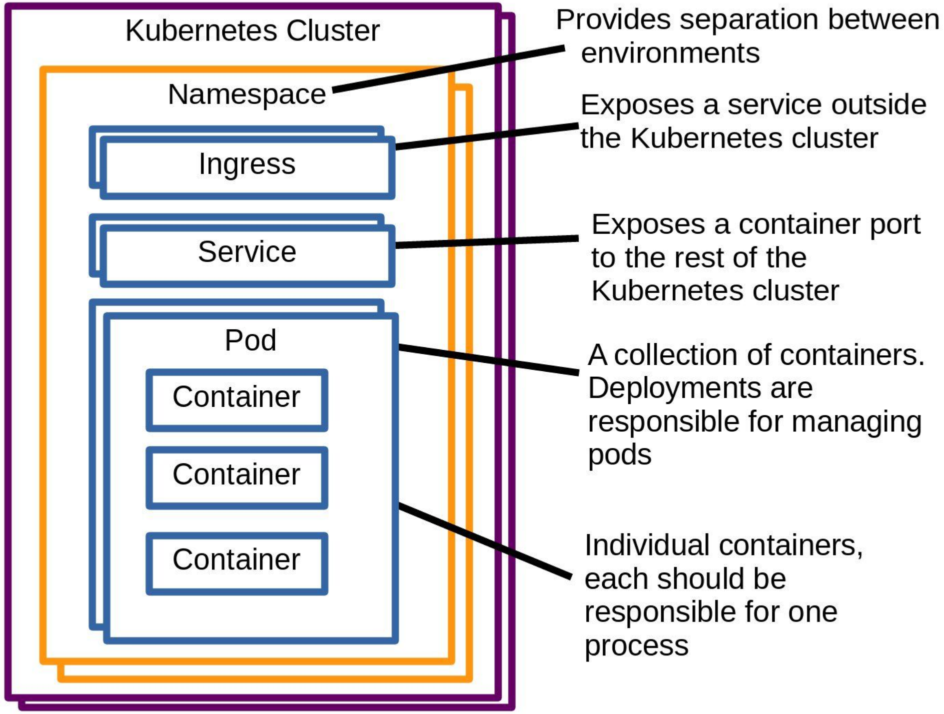
Kubernetes Overview

- Cloud democratization
- Easy deployment
- Controls most of the aspects
- Adopted at NCSA, CERN, LSST, NASA
- Edge Computing
- Scalability
- Federation
- Resource Manager

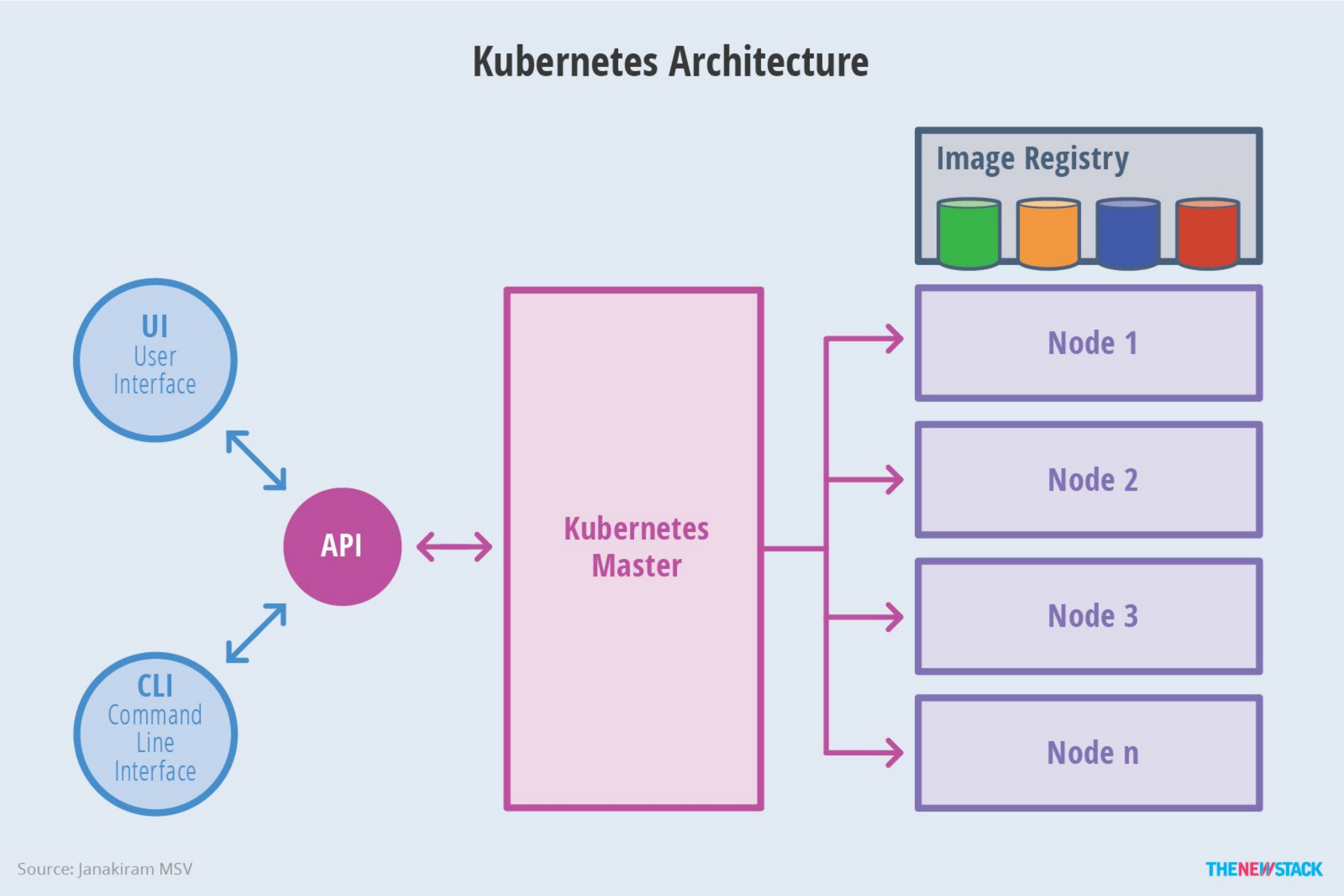


Kubernetes Key Concepts

- **Pod** - A group of Containers
- **Labels** - Labels for identifying pods
- **Kubelet** - Container Agent
- **Proxy** - A load balancer for Pods
- **etcd** - A metadata service
- **cAdvisor** - Container Advisor provides resource usage/performance statistics
- **Replication Controller** - Manages replication of pods
- **Scheduler** - Schedules pods in worker nodes
- **API Server** - Kubernetes API server

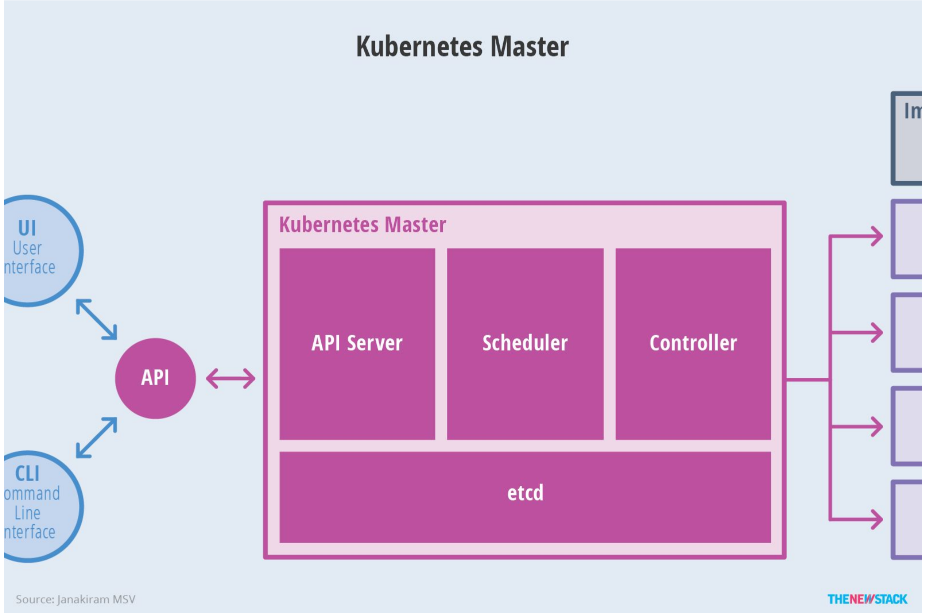


The Kubernetes Architecture

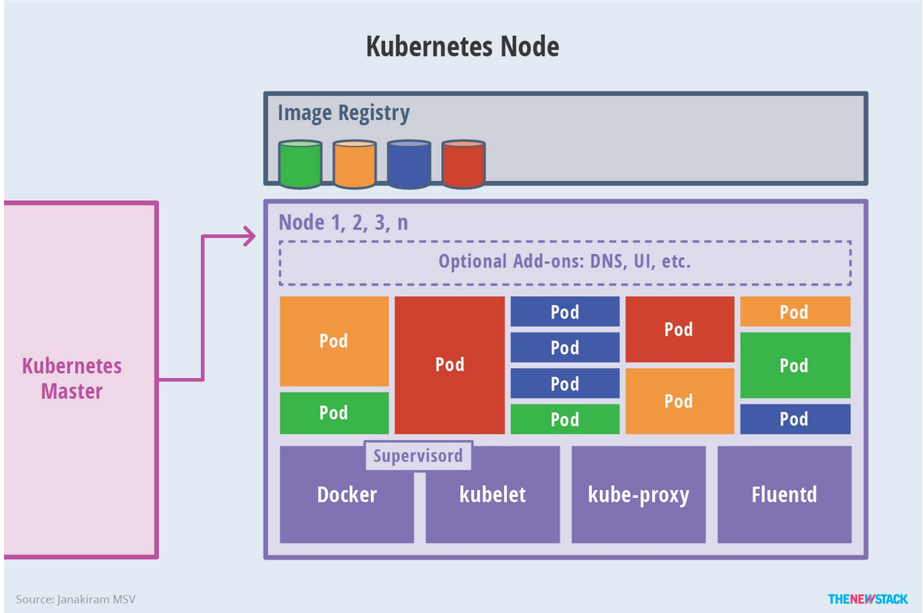


The Kubernetes Architecture

Master



Nodes



Applications

- DES Infrastructure
- LSST Science Platform (next week's talk)
- Anomaly detection service

The Dark Energy Survey

- 4 meters telescope, 520 Mpx camera
- 5 year survey, ⅛ of the sky, Telescope in Chile, data @ NCSA, about to start 6th season
- Main Goal: To constrain the models of the Universe regarding Dark Energy and Dark Matter.
- Many other Science Cases! (New dwarf planet, New galaxy satellites, Supernovae, etc)
- 1 - 3 TB of data per night, 1 PB of data
- Processing done at FermiGrid, Campus Cluster and Blue Waters
- Thousands of images and billions of rows, ~500 millions objects
- 1st Public Data Release in January 2018
- NCSA provide means to access and interact with data → Containers

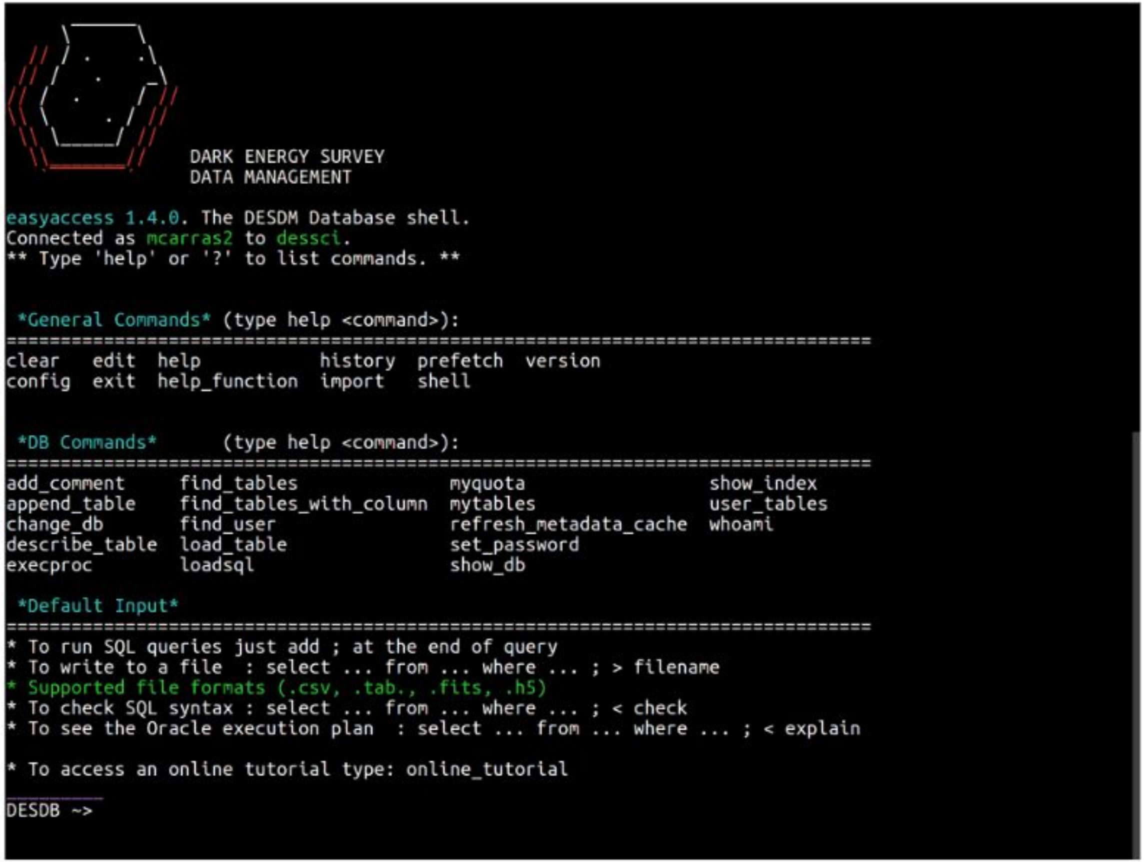
The DES Data Access

Challenges:

- Data access wasn't very clear in original proposal
- People
- Time
- Collaborations Needs
- All the rest of technical challenges

- DES Survey: Gold (Data) Mine
- DESDM: Excellent job at mining the data
- Consumers outside the mine
- Need to bring/expose gold (data) outside
- Tools and interfaces
- DES DR1 is out!

easyaccess: DES command line tool



- DES DB in Oracle
- Specifically designed for DES (internal and public)
- Enhanced SQL command line interpreter in Python
- Astronomer friendly
- Python API, web interface
- There are many other CLI and GUI clients.
- Needed a simple tool, easy to use and install
- Autocompletion
- Load/Save to hdf5, fits, csv

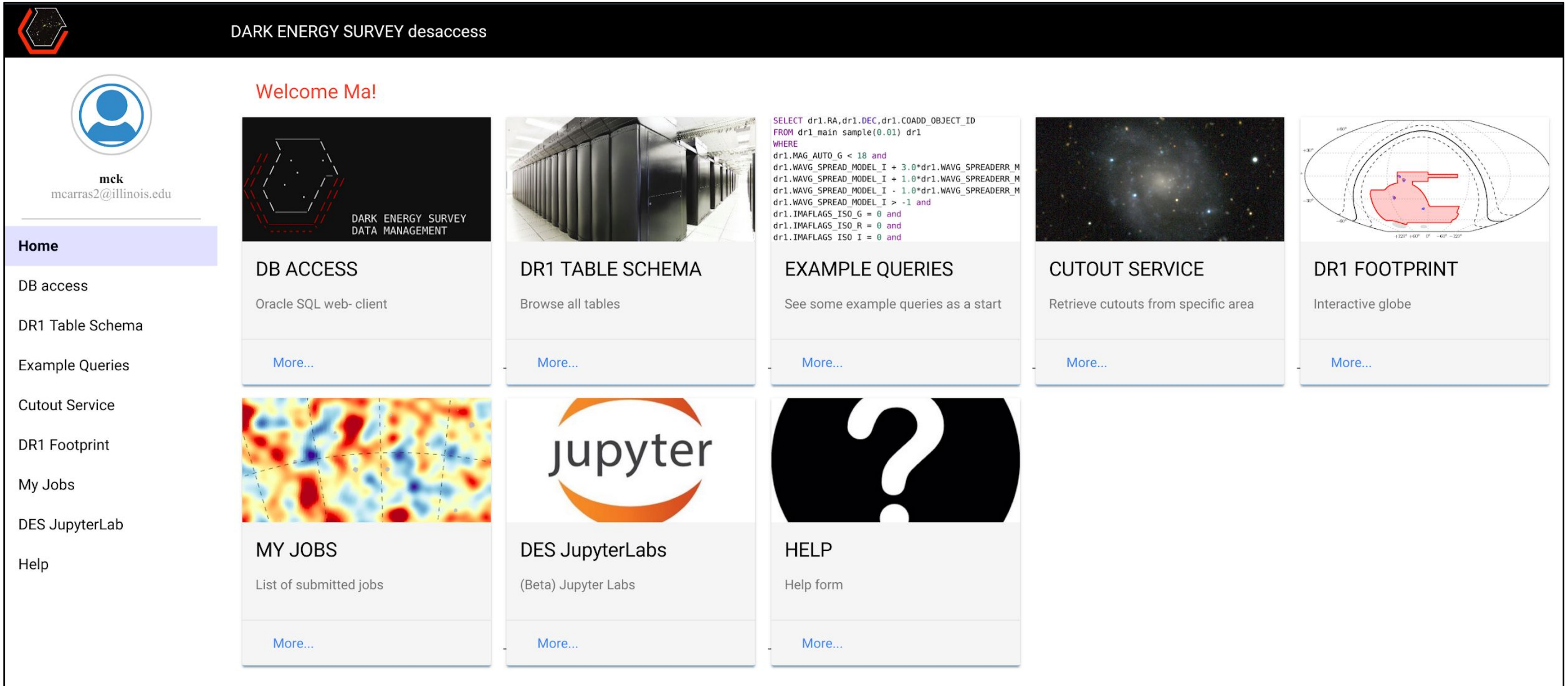
easyaccess: DES command line tool

- DES DB in Oracle
- Specifically designed for DES (internal and public)
- Enhanced SQL command line interpreter in Python
- Astronomer friendly
- Python API, web interface
- There are many other CLI and GUI clients.
- Needed a simple tool, easy to use and install
- Autocompletion
- Load/Save to hdf5, fits, csv

DES Labs: Collection of containerized tools for DES

- March 2015
- Used by the Collaboration
- Running using Kubernetes at NCSA cloud
- Currently being migrated to match DR1 Infrastructure

NCSA DESaccess: DR1 Infrastructure

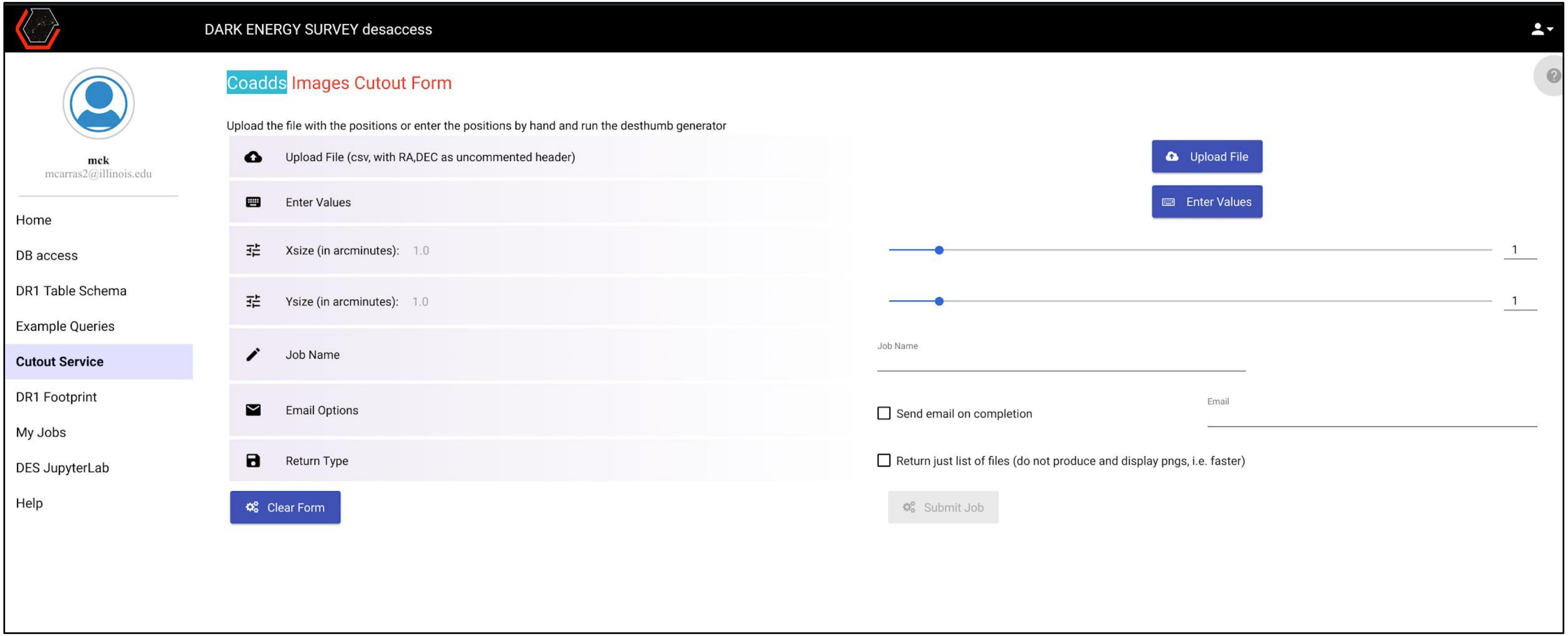


des.ncsa.illinois.edu/easyweb

NCSA DESaccess: DB access

des.ncsa.illinois.edu/easyweb

NCSA DESaccess: Cutouts Service



des.ncsa.illinois.edu/easyweb

NCSA DESaccess: Asynchronous Jobs

DARK ENERGY SURVEY desaccess

mck
mcarras2@illinois.edu

- Home
- DB access
- DR1 Table Schema
- Example Queries
- Cutout Service
- DR1 Footprint
- **My Jobs**
- DES JupyterLab
- Help

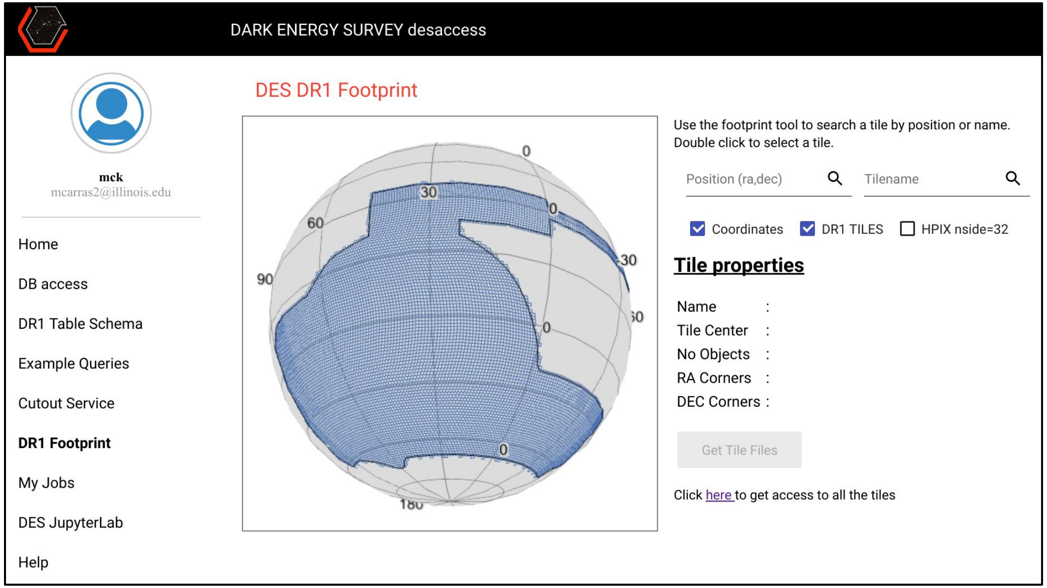
My Jobs

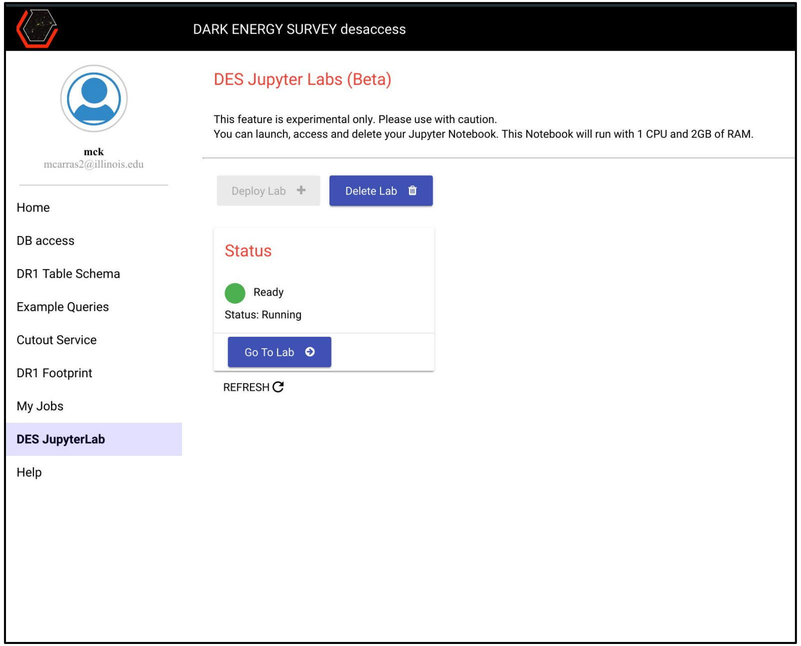
#	Status	Job Name	Job type	Execution time (s)	Cancel Job	Queries	Results	Files
0	green	Name: Job id: 6b4cac2b-b544-44e1-9bbf-58cd4968a338 6 days and 0 hours ago (Expired)	query	0	⊗	Query	Cutouts	Files
1	green	Name: Job id: daf5ee3c-461e-42ed-8efb-5fcfbf684047 6 days and 0 hours ago (Expired)	cutout	1	⊗	Query	Cutouts	Files
2	green	Name: testapi Job id: 0d6c5a58-b00a-4798-834f-9816c6fa98e5 7 days and 4 hours ago (Expired)	cutout	3	⊗	Query	Cutouts	Files
3	green	Name: testapi Job id: 12861656-8075-4629-8e4f-fd4378013634 7 days and 4 hours ago (Expired)	cutout	3	⊗	Query	Cutouts	Files
4	green	Name: testapi Job id: d9a37fe9-209b-4296-b87d-c6567cde0649 7 days and 4 hours ago (Expired)	cutout	1	⊗	Query	Cutouts	Files
5	green	Name: Job id: 6d10cf32-3cd6-4090-bb90-344268dd615e 7 days and 5 hours ago (Expired)	cutout	1	⊗	Query	Cutouts	Files
6	red	Name: testapi Job id: b85ea747-5201-4e49-a0eb-f2b6b7f266de 7 days and 5 hours ago (Expired)	cutout	-1	⊗	Query	Cutouts	Files
7	green	Name: Job id: 8bfea56a-4685-49f9-b7be-603310ccddeb 8 days and 16 hours ago (Expired)	query	577	⊗	Query	Cutouts	Files
8	green	Name: Job id: df8a57c4-b1d5-4332-80d5-a08a27b537d9 8 days and 16 hours ago (Expired)	query	1042	⊗	Query	Cutouts	Files
9	red	Name: Job id: 7ffdb550-4d38-441f-a037-ed659b3b79c9 8 days and 16 hours ago (Expired)	query	-1	⊗	Query	Cutouts	Files
10	green	Name: Job id: fcaacdec-9d63-45a4-92f2-4f847b9b415c 8 days and 16 hours ago (Expired)	query	9	⊗	Query	Cutouts	Files
11	green	Name: Job id: a88b79cc-fd71-4ed0-a33d-92b5be98106f 8 days and 17 hours ago (Expired)	query	9	⊗	Query	Cutouts	Files
		Name: demo1						

REFRESH DELETE

des.ncsa.illinois.edu/easyweb

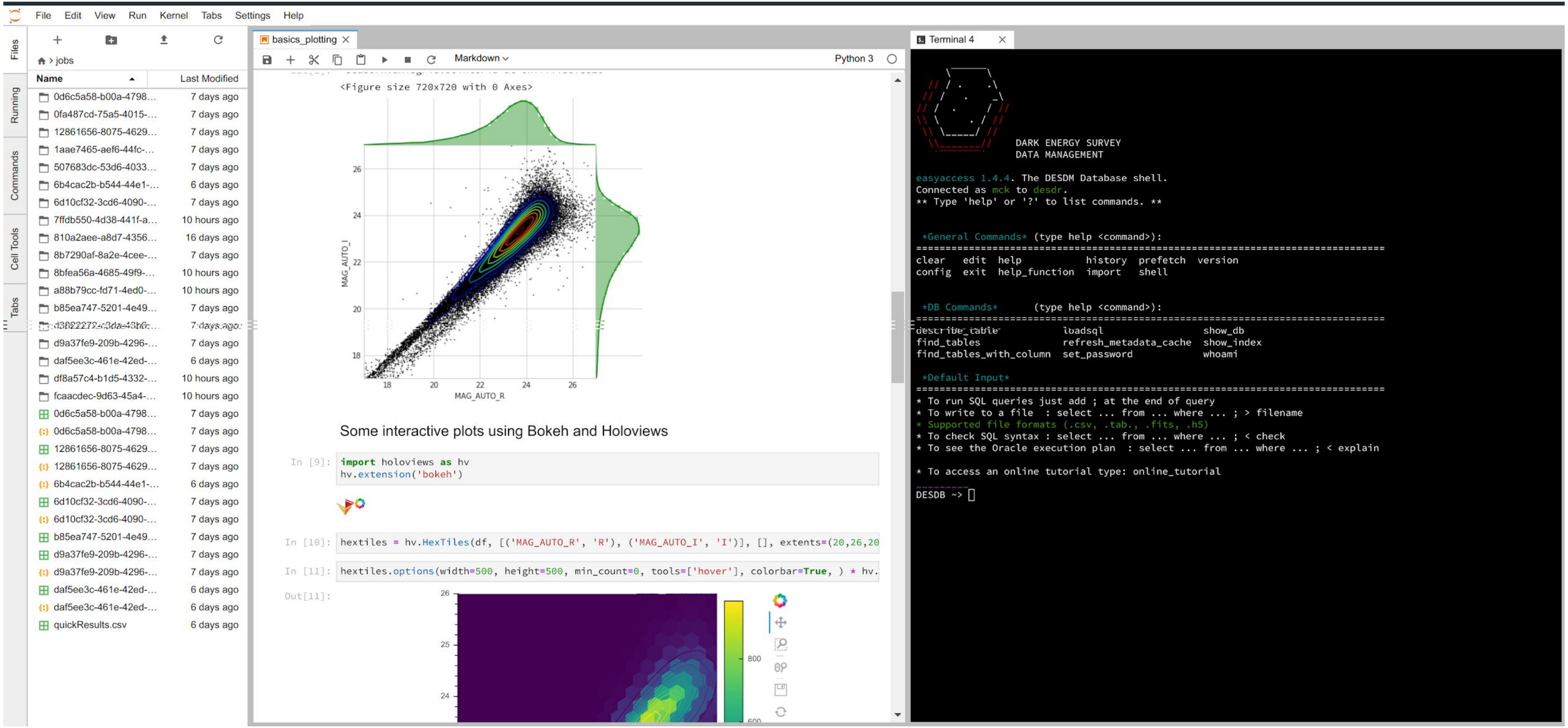
NCSA DESaccess: Footprint and Jupyter Labs



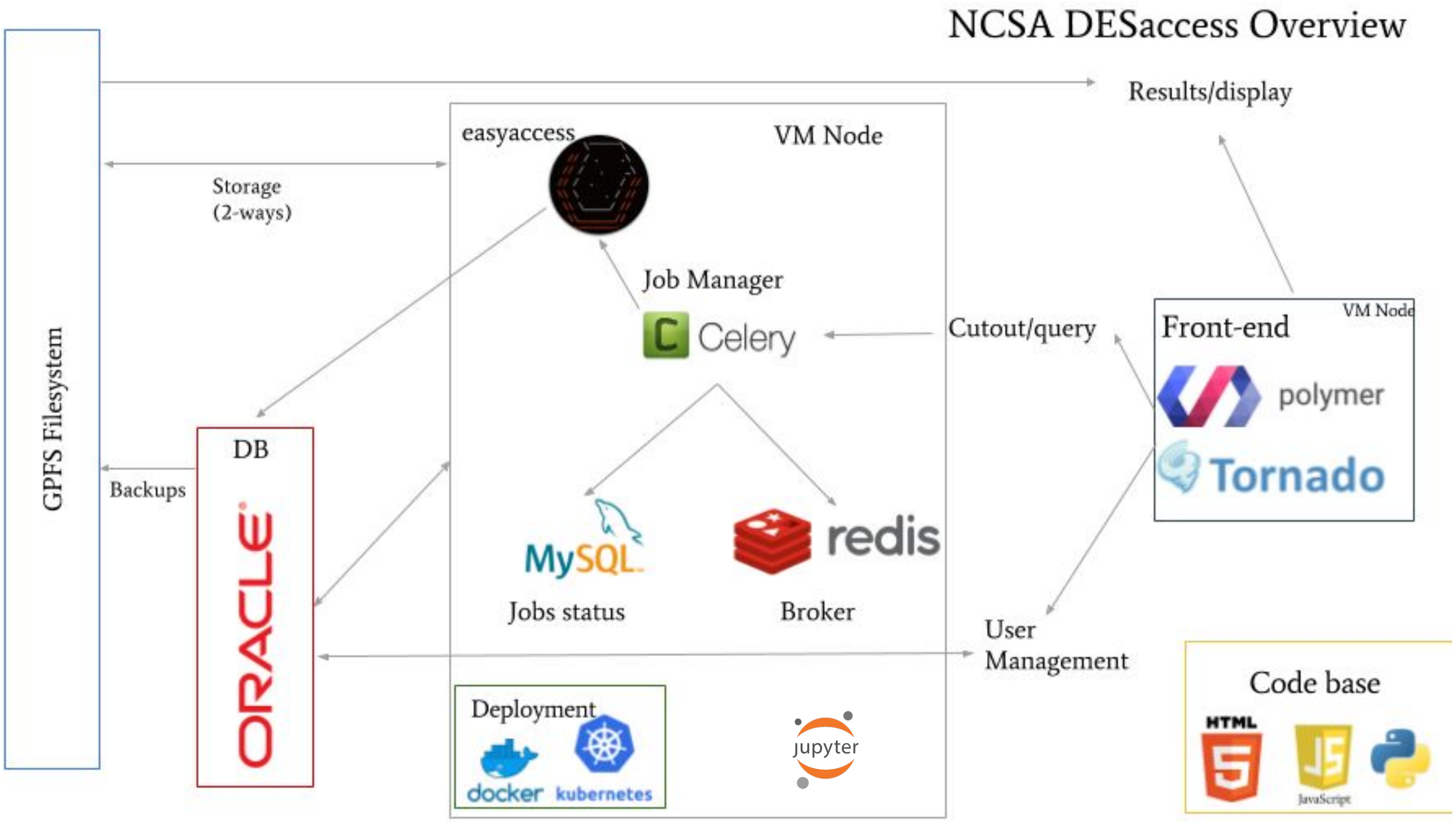


des.ncsa.illinois.edu/easyweb

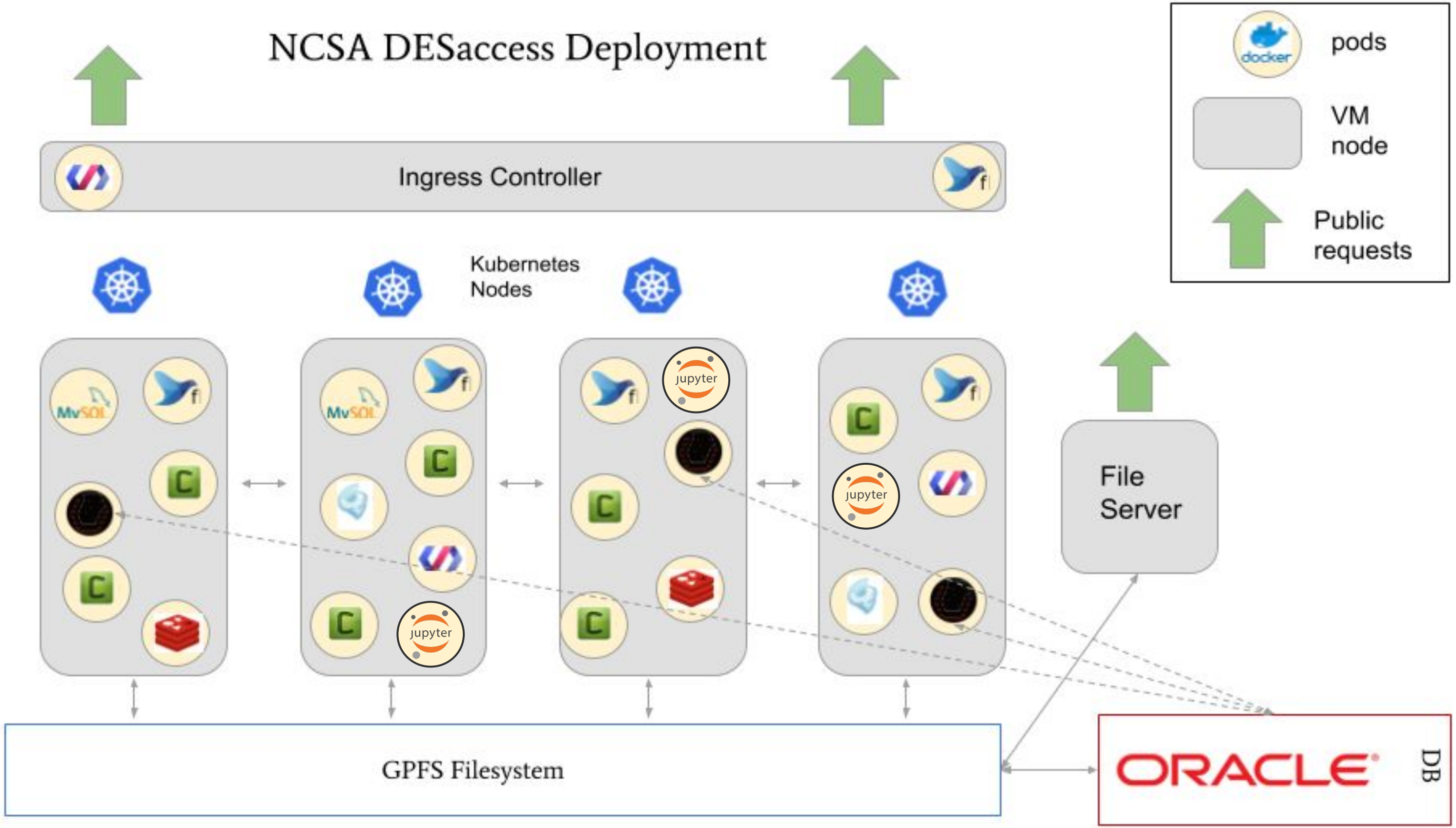
NCSA DESaccess: Labs with access to Jobs and easyaccess



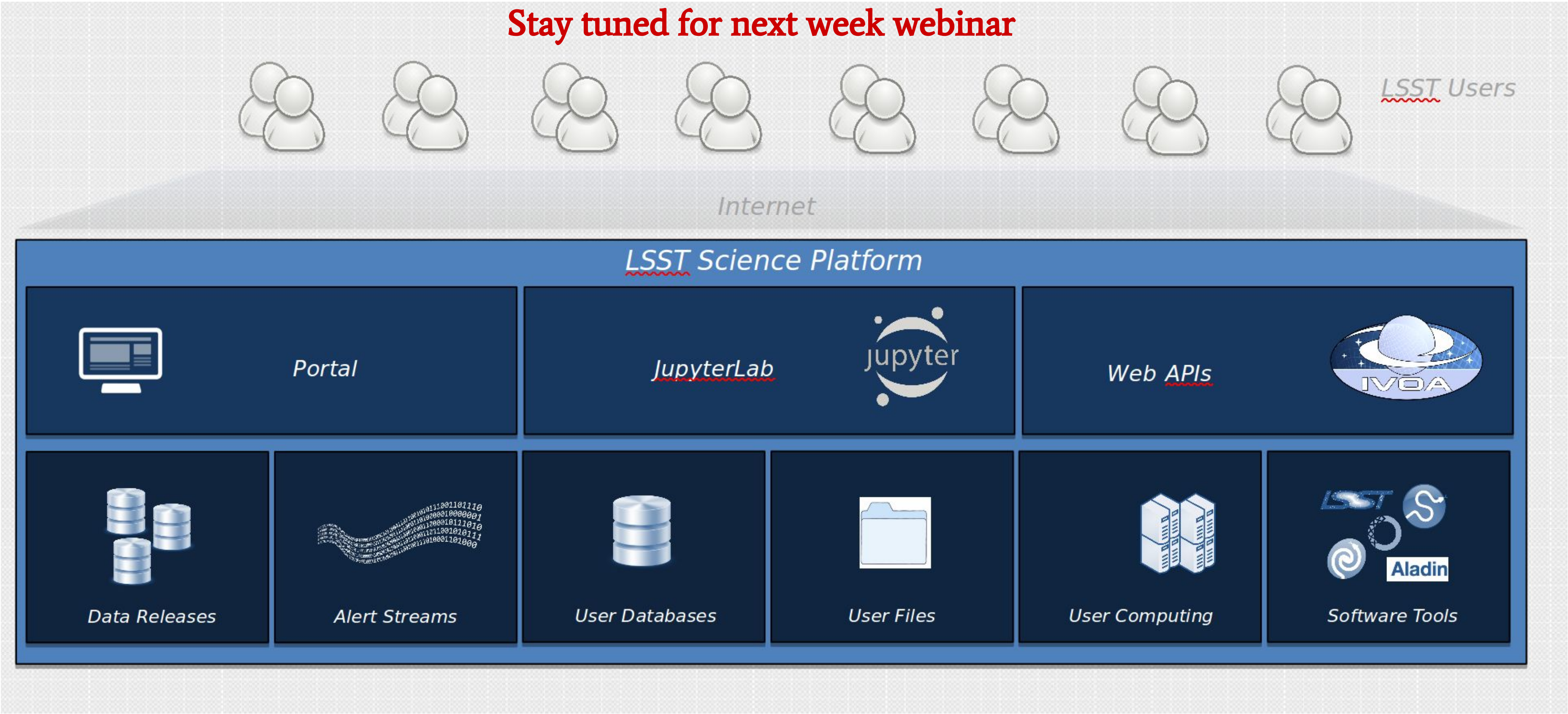
NCSA DESaccess: Technology Overview



NCSA DESacces: Deployment



LSST Science Platform



SCIaaS Example: Anomaly detection service

Goal: Build a resilient scalable anomaly detection service.

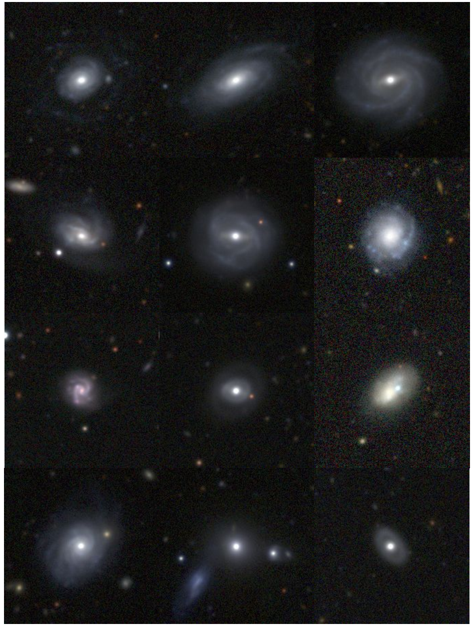
Motivation: Astronomical data (both literal and figurative)

Algorithm: Extended Isolation Forest

Infrastructure: Kubernetes cluster

MapReduce package: Spark

Part of the Motivation



Astronomy is just one example where data exploration needs to be automated.

Large catalogs, Large number of images, many unexpected objects/problems → Anomaly detection

Large Synoptic Survey Telescope

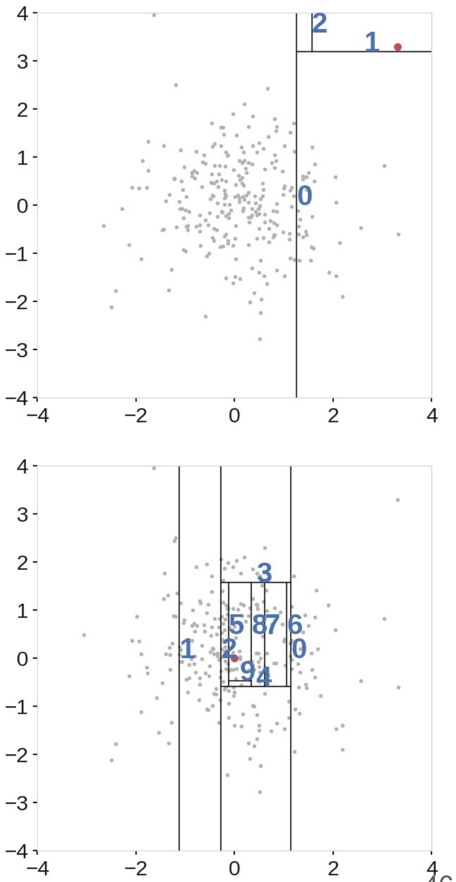
- In operations 2020
- Every night for 10 years
- 18 billions objects (first year), ~40 billions by the end of survey
- ~1500 images per night
- Stream and static data
- Target to capture new physics (moving and variable objects)

DARK ENERGY SURVEY

- More than 500 nights of observation over 5 years
- 500 millions cataloged galaxies and 100 millions stars
- Many open problems: Systematics, new objects, new physics, etc.
- Almost completed

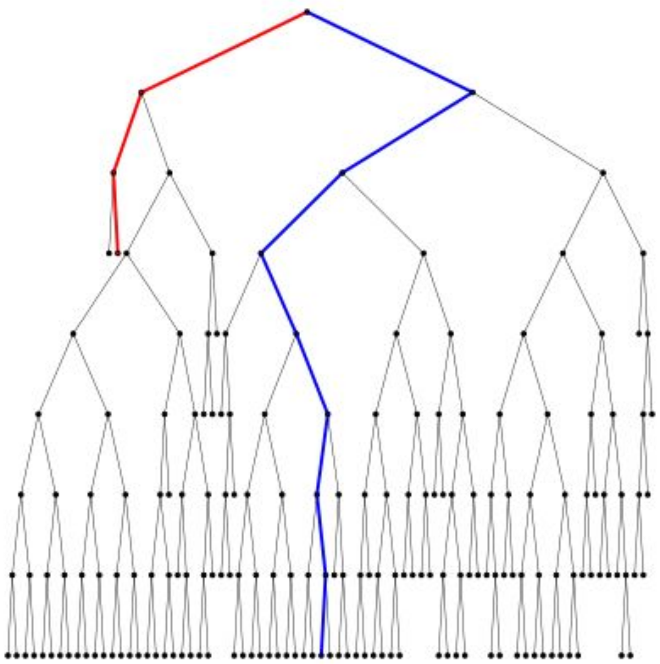
Anomaly Detection with Isolation Forest

- Few and different to be isolated quicker
- For each tree:
  - Get a sample of the data
  - Randomly select a dimension
  - Randomly pick a value in that dimension
  - Draw a straight line through the data at that value and split data
  - Repeat until tree is complete
- Generate multiple trees $\longrightarrow$ forest
- Anomalies will be isolated in only a few steps
- Nominal points in more
- To score points:
  - Run point down tree, record path
  - Repeat for each tree, aggregate scores $s(x,n) = 2^{-\frac{E(h(x))}{c(n)}}$
  - Score distribution

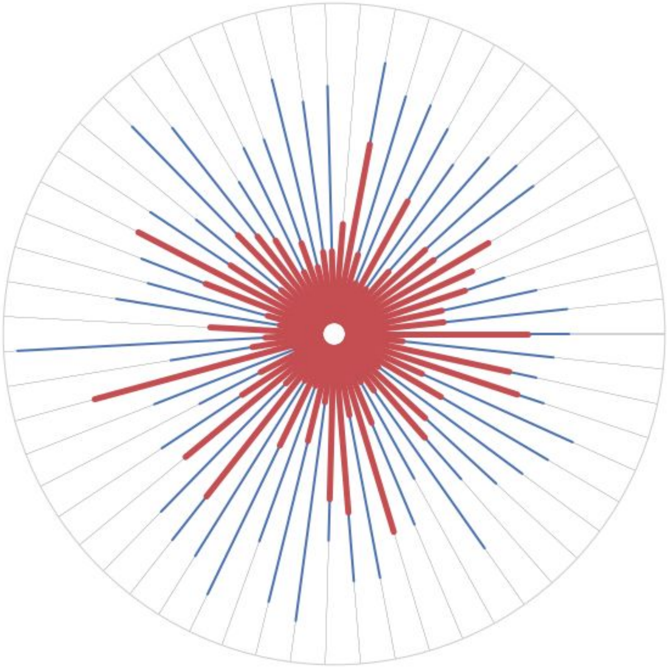


Anomaly Detection with Isolation Forest

Single Tree scores for anomaly and nominal points



Forest plotted radially. Scores for anomaly and nominal shown as lines



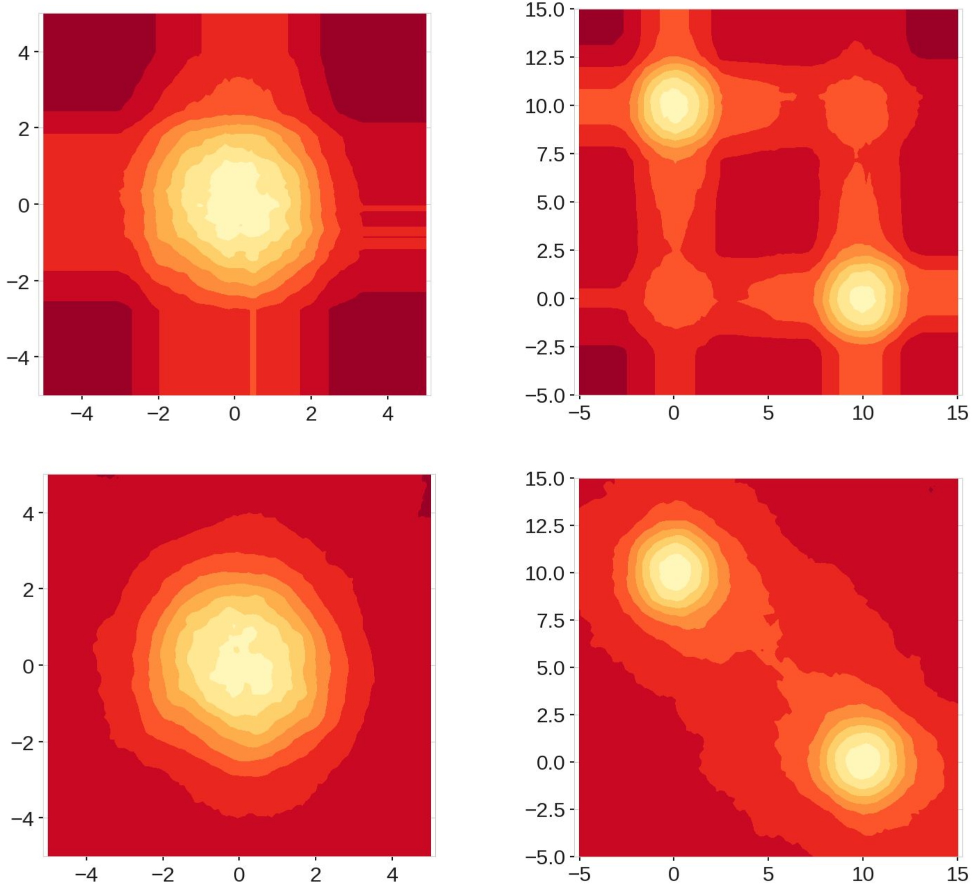
Anomaly Detection with Extended Isolation Forest

Isolation Forest:

- ✓ Model free
- ✓ Computationally efficient
- ✓ Readily applicable to parallelization
- ✓ Readily application to high dimensional data
- ✗ Inconsistent scoring seen in score maps

Extended Isolation Forest:

- ✓ Model free
- ✓ Computationally efficient
- ✓ Readily applicable to parallelization
- ✓ Readily application to high dimensional data
- ✓ Consistent scoring



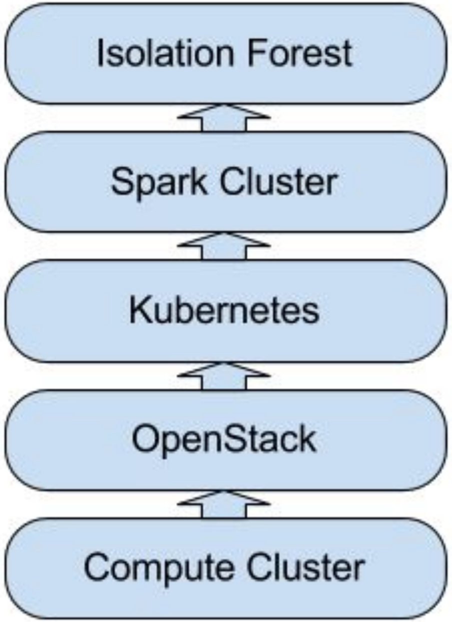
Technology Stack For Anomaly Service

Batch and online anomaly detection for scientific applications in a Kubernetes environment

Sahand Hariri*
University of Illinois at Urbana-Champaign
sahandha@gmail.com

Matias Carrasco Kind†
National Center for Supercomputing Applications
mcarras2@illinois.edu

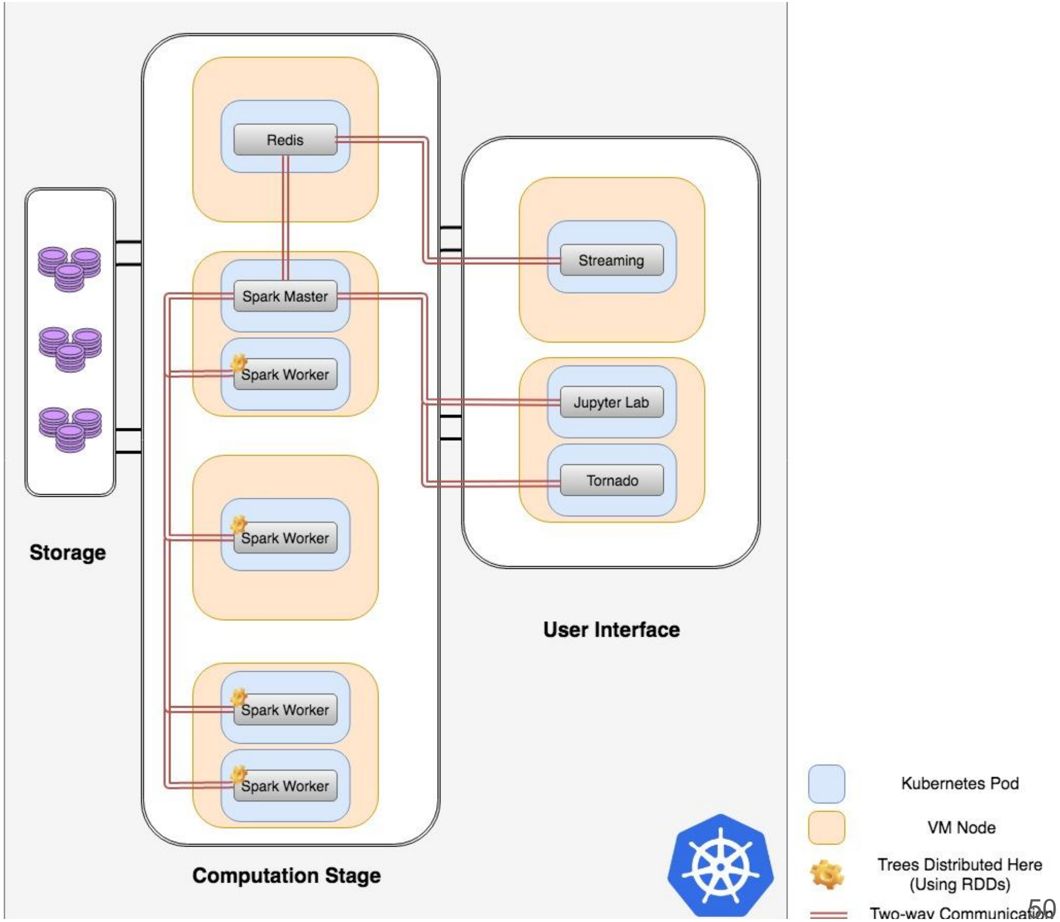
- Use Extended Isolation Forest as core algorithm
- Use Spark to parallelize trees and scoring
- Use Redis as a broker communicator
- To easily deploy in any environment, use Docker
- For orchestration of Docker containers, use Kubernetes
- Kubernetes cluster built on top of OpenStack, but it can be deployed also in AWS, GKE, etc.



Framework Architecture

There are three main components:

1. Storage
2. Computation Stage
3. User Interface / Streaming



Framework Architecture

Storage:

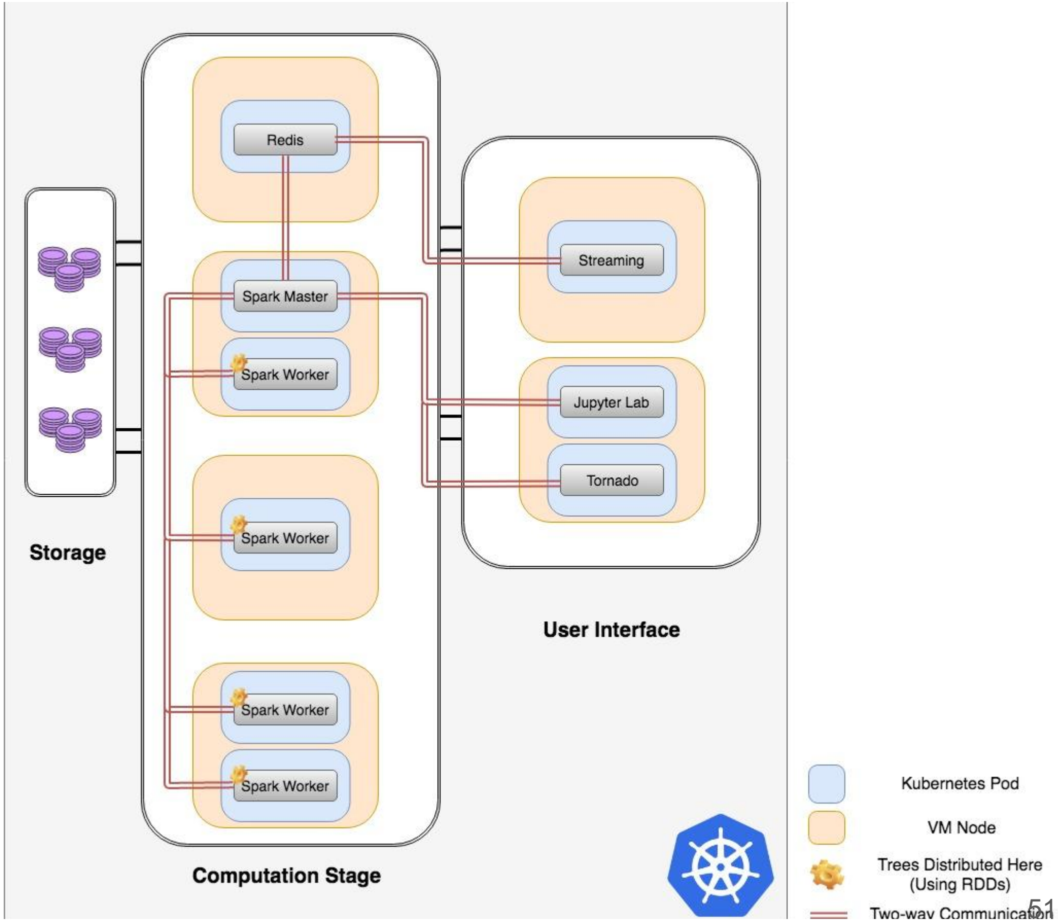
- NFS (Kubernetes PV/PVC)
- Redis
- RDD for Trees and Spark

User Interface:

- Jupyter notebooks
- Interactive web app for submitting jobs
- Streaming service

Computation Stage:

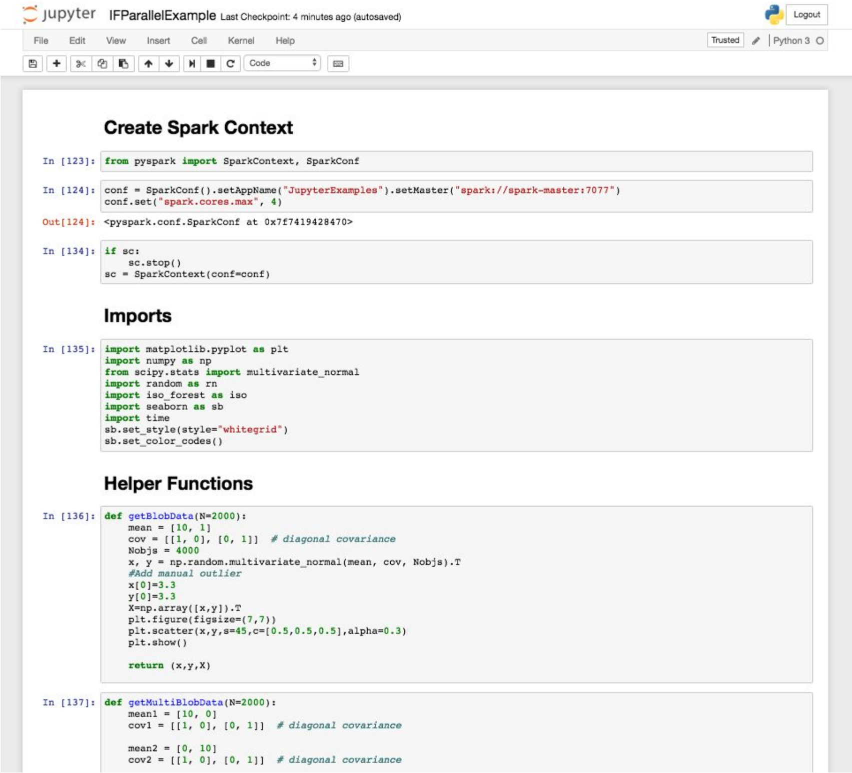
- Spark Master and Workers
- Communicator with Spark Master
- Subscription



Deployment

- Kubernetes allows very easy deployment, orchestration, scalability, resilience, replication, workloads and more
- Federation of services and Jobs
- From 0 to anomaly service → in minutes and config files
- Scale up/down (spark cluster and front-end) → Auto-scaling as an option
- Prototype support multiple users/projects, batch and streaming process
- Fault tolerant, disaster recovery

Example: Jupyter Notebooks



jupyter IFParallelExample Last Checkpoint: 4 minutes ago (autosaved)

File Edit View Insert Cell Kernel Help

Trusted | Python 3

Logout

Create Spark Context

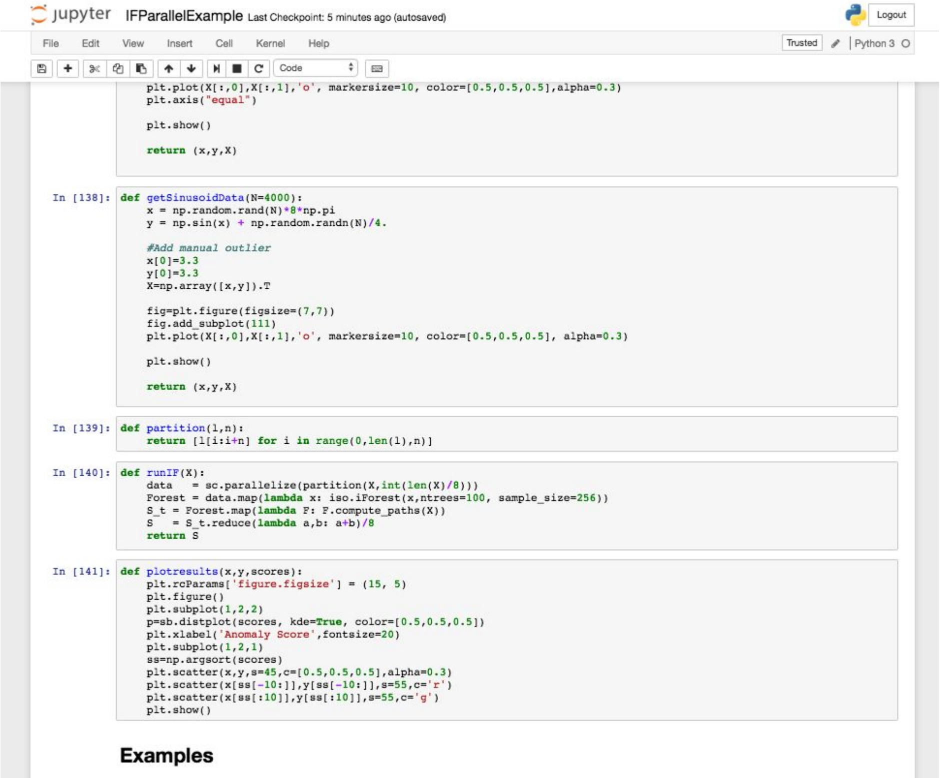
```
In [123]: from pyspark import SparkContext, SparkConf

In [124]: conf = SparkConf().setAppName("JupyterExamples").setMaster("spark://spark-master:7077")
          conf.set("spark.cores.max", 4)
Out[124]: <pyspark.conf.SparkConf at 0x7f7419428470>

In [134]: if sc:
              sc.stop()
          sc = SparkContext(conf=conf)
```

Imports

```
In [135]: import matplotlib.pyplot as plt
          import numpy as np
          from scipy.stats import multivariate_normal
          import random as rn
          import iso_forest as iso
          import seaborn as sb
          import time
          sb.set_style(style="whitegrid")
          sb.set_color_codes()
```

Helper Functions

```
In [136]: def getBlobData(N=2000):
              mean = [10, 1]
              cov = [[1, 0], [0, 1]]  # diagonal covariance
              Nobjs = 4000
              x, y = np.random.multivariate_normal(mean, cov, Nobjs).T
              #Add manual outlier
              x[0]=3.3
              y[0]=3.3
              X=np.array([x,y]).T
              plt.figure(figsize=(7,7))
              plt.scatter(x,y,s=45,c=[0.5,0.5,0.5],alpha=0.3)
              plt.show()

              return (x,y,X)

In [137]: def getMultiBlobData(N=2000):
              mean1 = [10, 0]
              cov1 = [[1, 0], [0, 1]]  # diagonal covariance

              mean2 = [0, 10]
              cov2 = [[1, 0], [0, 1]]  # diagonal covariance
```



jupyter IFParallelExample Last Checkpoint: 5 minutes ago (autosaved)

File Edit View Insert Cell Kernel Help

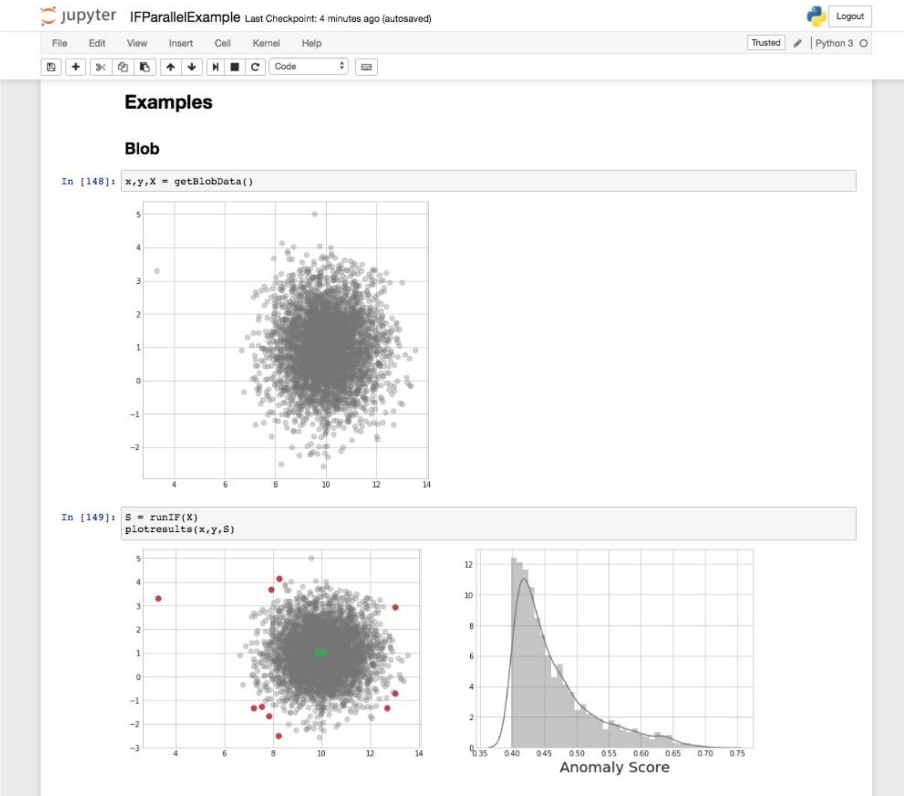
Trusted | Python 3

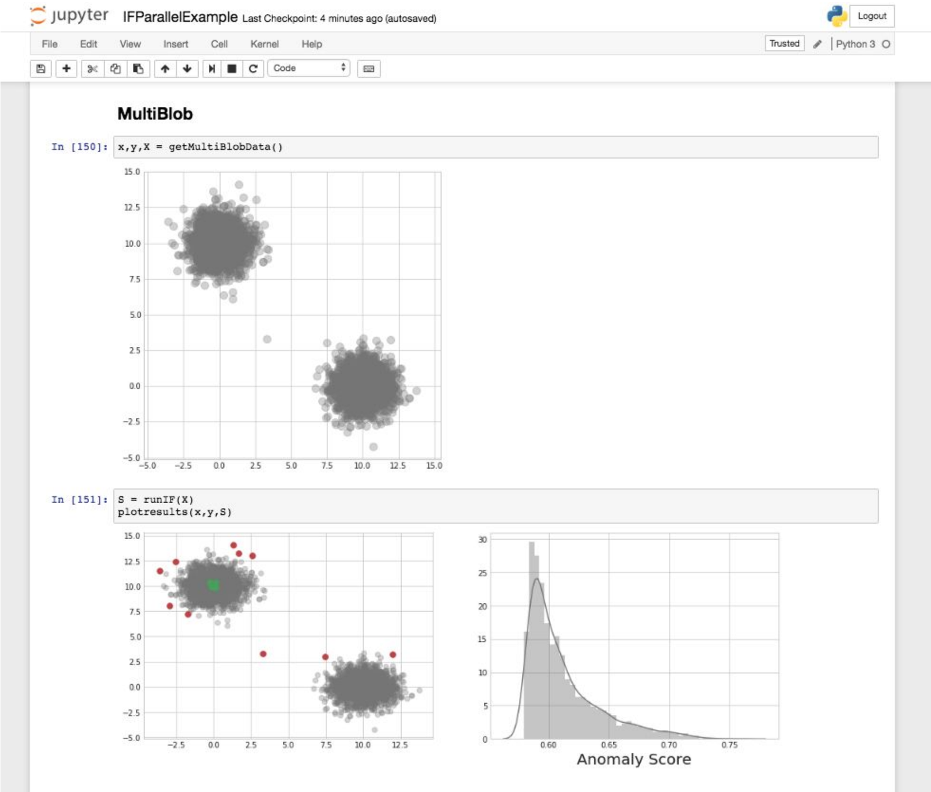
Logout

```
              plt.plot(X[:,0],X[:,1],'o', markersize=10, color=[0.5,0.5,0.5],alpha=0.3)
              plt.axis("equal")

              plt.show()

              return (x,y,X)

In [138]: def getSinusoidData(N=4000):
              x = np.random.rand(N)*8*np.pi
              y = np.sin(x) + np.random.randn(N)/4.

              #Add manual outlier
              x[0]=3.3
              y[0]=3.3
              X=np.array([x,y]).T

              fig=plt.figure(figsize=(7,7))
              fig.add_subplot(111)
              plt.plot(X[:,0],X[:,1],'o', markersize=10, color=[0.5,0.5,0.5], alpha=0.3)

              plt.show()

              return (x,y,X)

In [139]: def partition(l,n):
              return [l[i:i+n] for i in range(0,len(l),n)]

In [140]: def runIF(X):
              data   = sc.parallelize(partition(X,int(len(X)/8)))
              Forest = data.map(lambda x: iso.iForest(x,ntrees=100, sample_size=256))
              S_t = Forest.map(lambda F: F.compute_paths(X))
              S   = S_t.reduce(lambda a,b: a+b)/8
              return S

In [141]: def plotresults(x,y,scores):
              plt.rcParams['figure.figsize'] = (15, 5)
              plt.figure()
              plt.subplot(1,2,2)
              p=sb.distplot(scores, kde=True, color=[0.5,0.5,0.5])
              plt.xlabel('Anomaly Score',fontsize=20)
              plt.subplot(1,2,1)
              ss=np.argsort(scores)
              plt.scatter(x,y,s=45,c=[0.5,0.5,0.5],alpha=0.3)
              plt.scatter(x[ss[-10:]],y[ss[-10:]],s=55,c='r')
              plt.scatter(x[ss[:10]],y[ss[:10]],s=55,c='g')
              plt.show()
```

Examples

Example: Jupyter Notebooks





Final Remarks

Matias Carrasco Kind -- NCSA
mcarras2@illinois.edu
github.com/mgckind
matias-ck.com

- It's all about the user
- Jupyter as Scientific tool
- Science on the cloud is happening in many scientific fields including Astronomy
- Containerized solutions to ease management of the applications
- HPC is adopting cloud technologies to leverage the benefits of both worlds
- Kubernetes provide means to have 'the cloud' outside the commercial world
- Production services for large datasets

... this is changing the way we do astronomy

Thank you!

Questions?

Matias Carrasco Kind -- NCSA
mcarras2@illinois.edu
github.com/mgkind
matias-ck.com

Extra Slides