Scientific Platforms and Astronomical Data Access in the Era of (Scientific) Cloud Computing

Matias Carrasco Kind
Senior Research Scientist, NCSA/UIUC
Data Release Scientist, DES

AstroData2020s Science Workshop, Caltech
Dec 4th, 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
- Versioning
- 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
- Versioning
- Consistency
- Integrity
- Redundancy
- Data Model
- Storage
- Backups
- Recovery
- Hardware

Interfaces

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

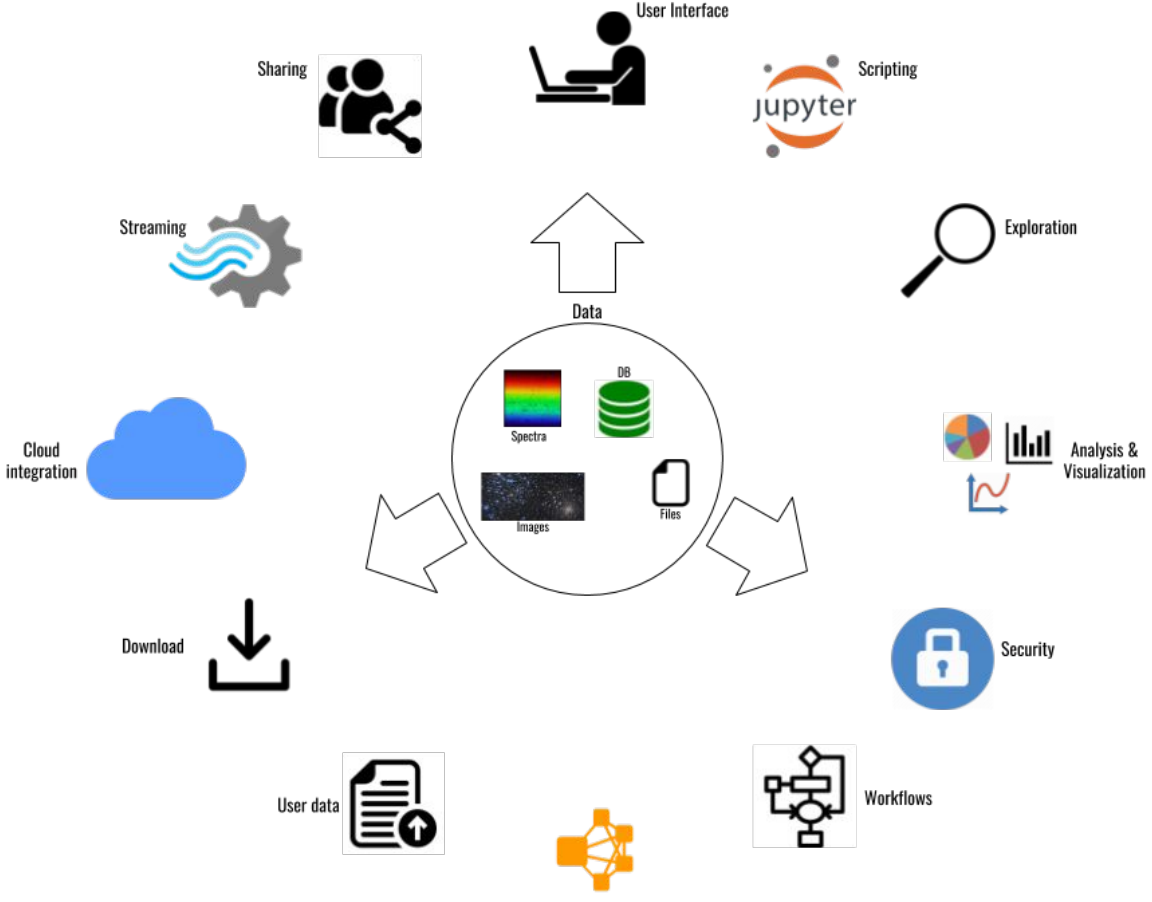
Documentation

- 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?

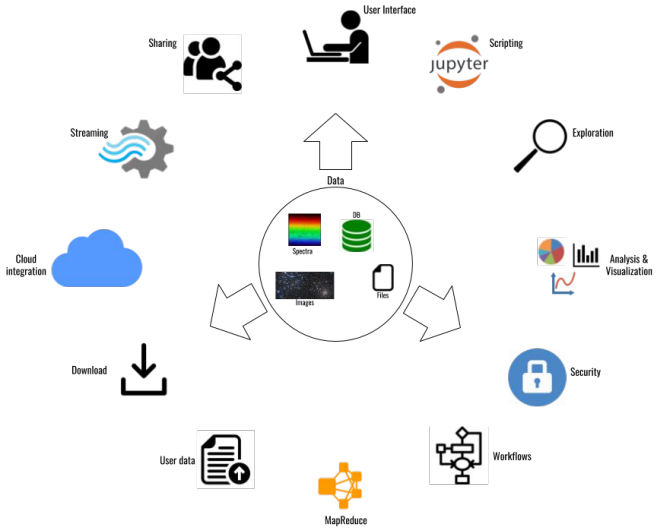


Several meanings around a central data archive, a.k.a “data lake”, repository with common components

- 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.)



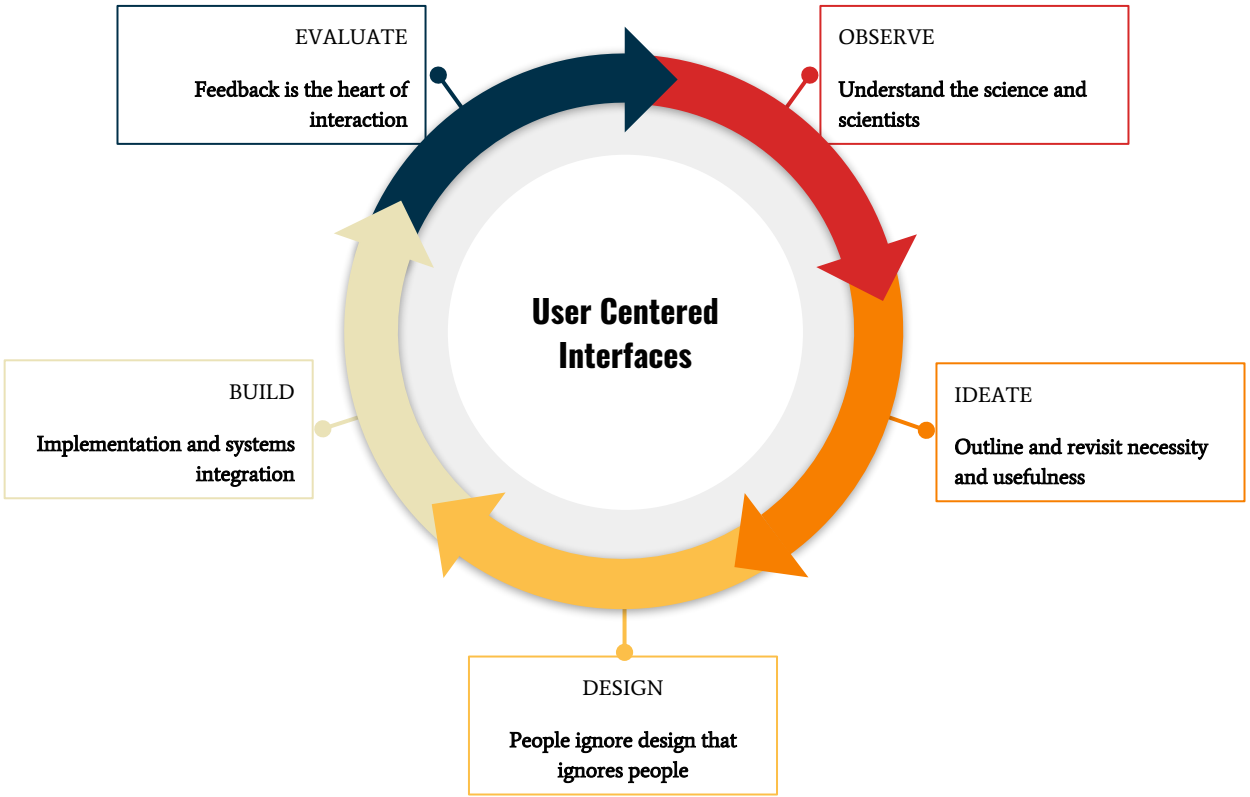
"Science gateways allow science & engineering communities to access shared data, software, computing services, instruments, educational materials, and other resources specific to their disciplines." (Science Gateways Institute)

"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.

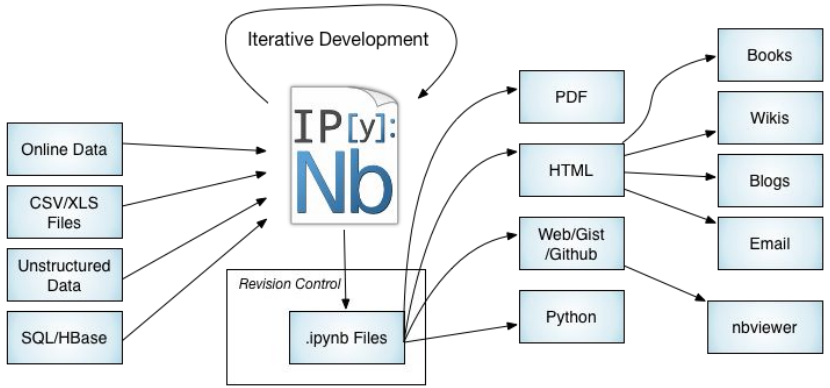
"... In an ideal world, a user would remember every function after only a single use, but we do not live in idealism. The reality is that familiarity and intuition must be consciously designed into the interface"

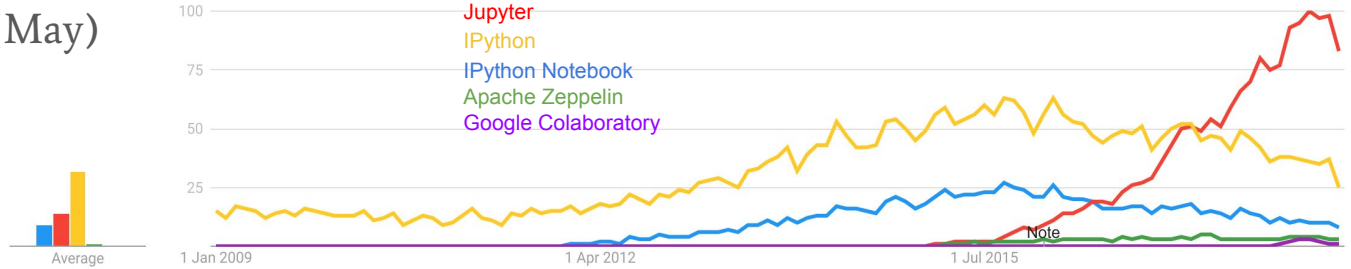


The Notebook Revolution

The Notebook Development

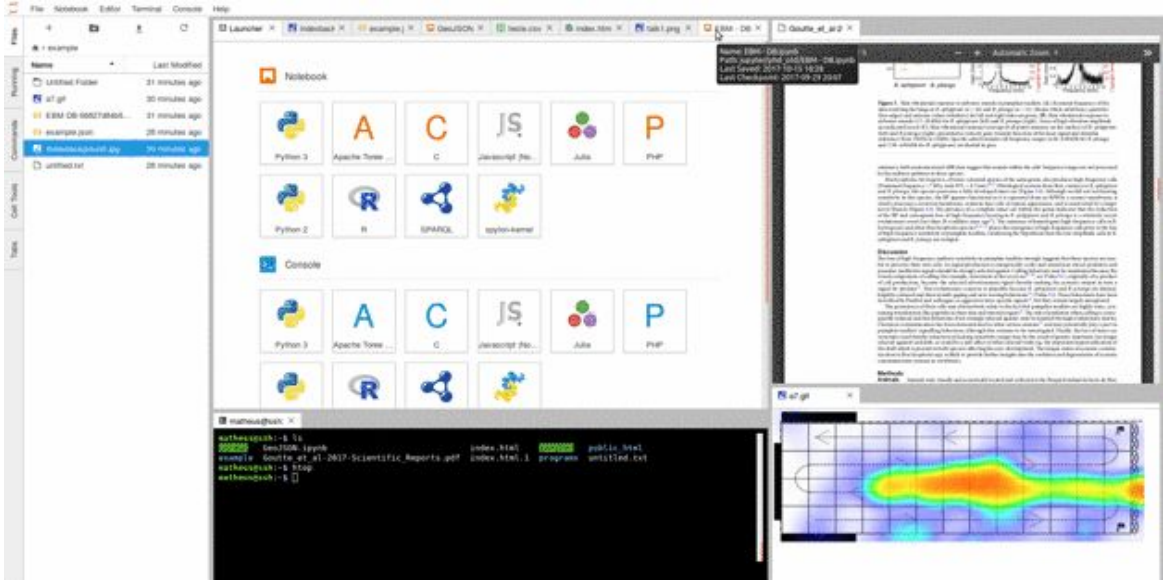
- Started from ideas like Matlab, Maple or Mathematica ~1988
- IPython has been around since 2001
- Sage Notebook released in 2005 (uses IPython)
- IPython Notebook was released in 2011
- IPython Notebook moved to Jupyter in 2014
- Apache Zeppelin created in 2015 (JVM and integrated with Apache Products)
- 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)





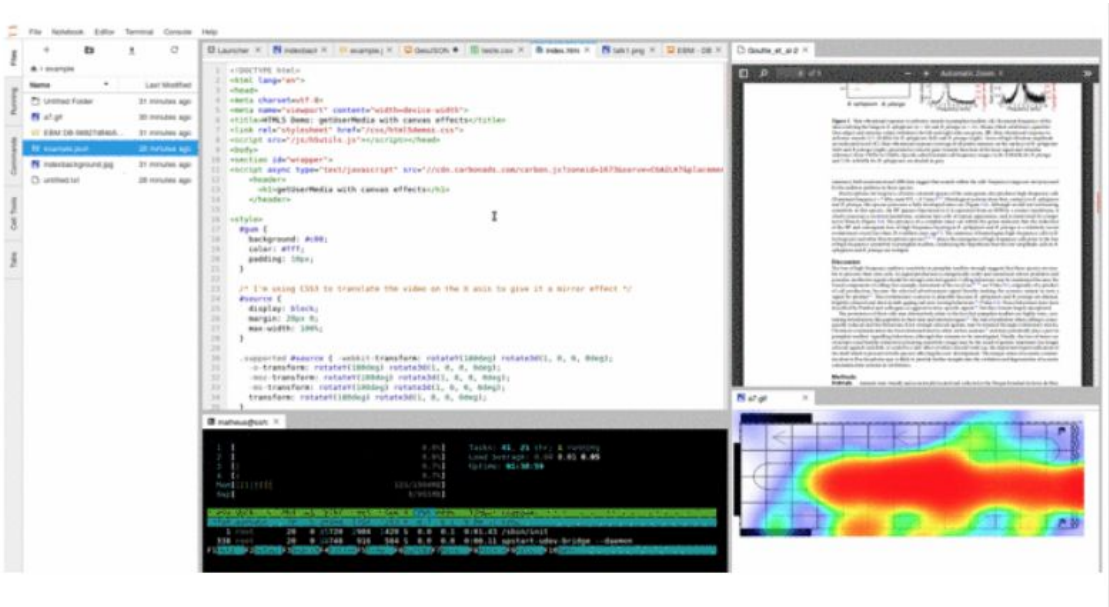
The Jupyter Notebook

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

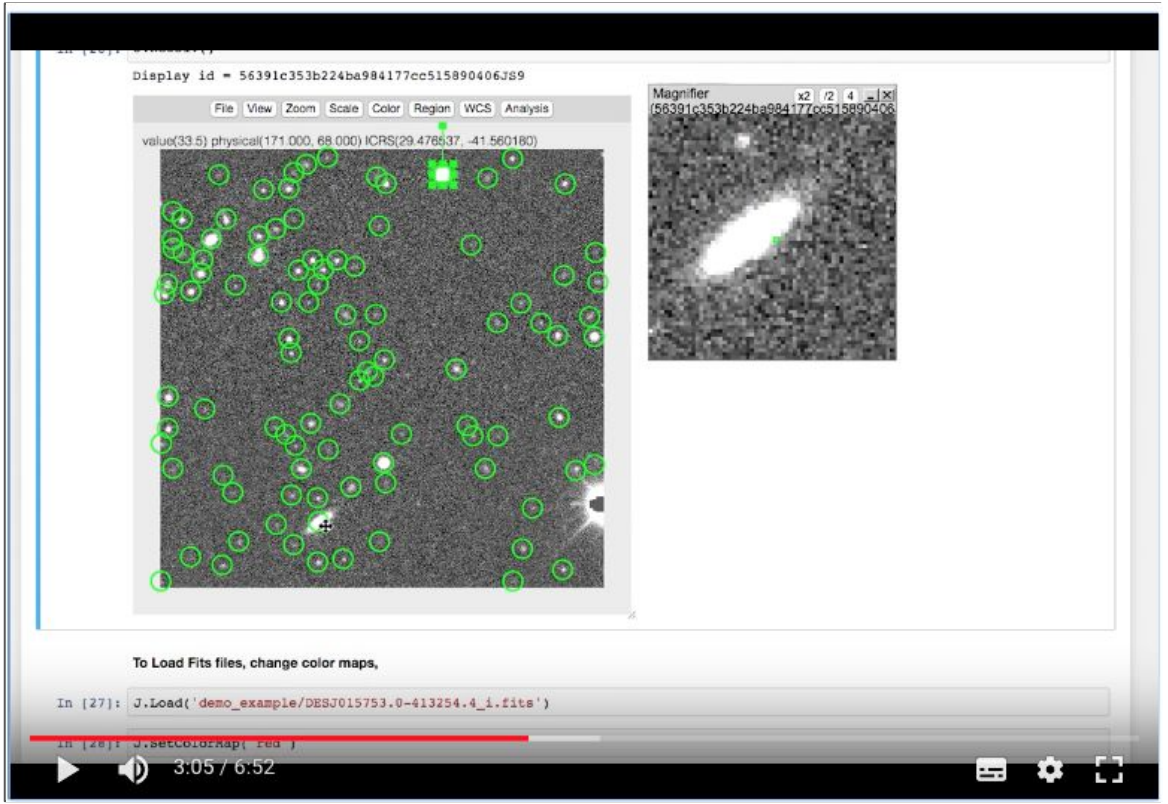


Jupyter in Astronomy

- 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, SciServer, etc)
- Tools and extensions developed by/for astronomers

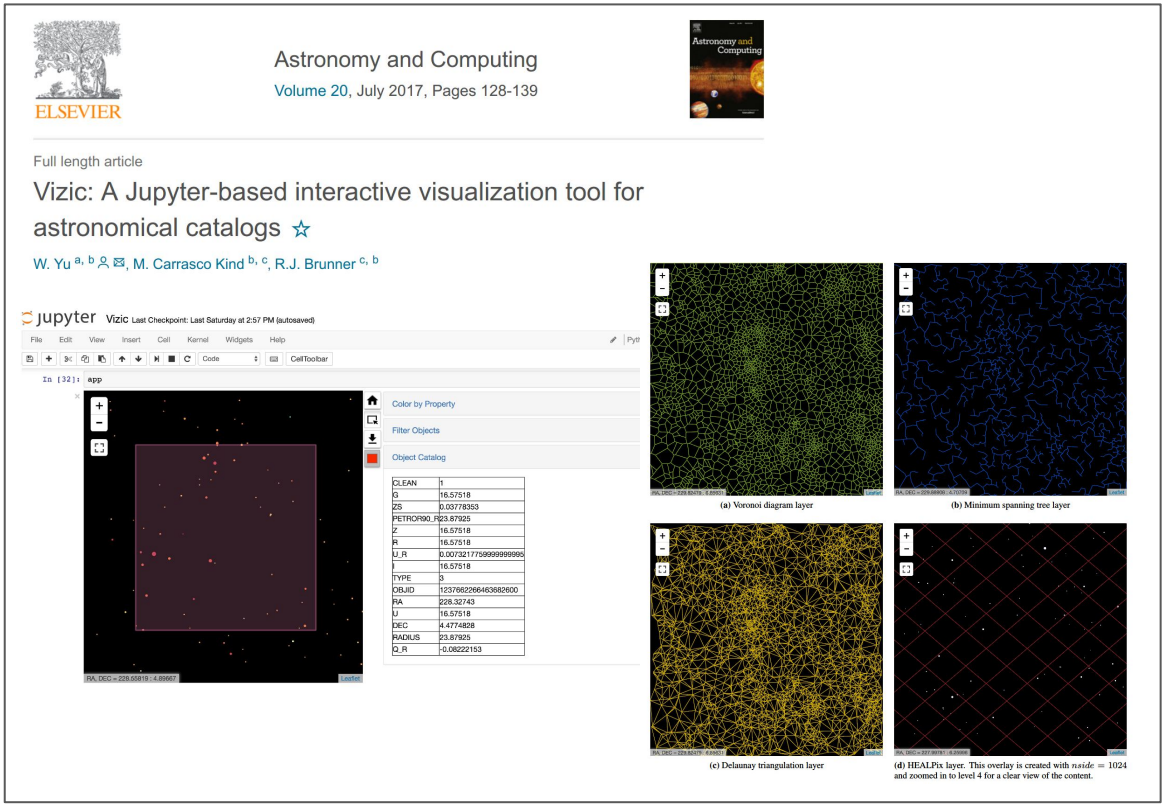
Jupyter in Astronomy

- 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, SciServer, etc)
- Tools and extensions developed by/for astronomers



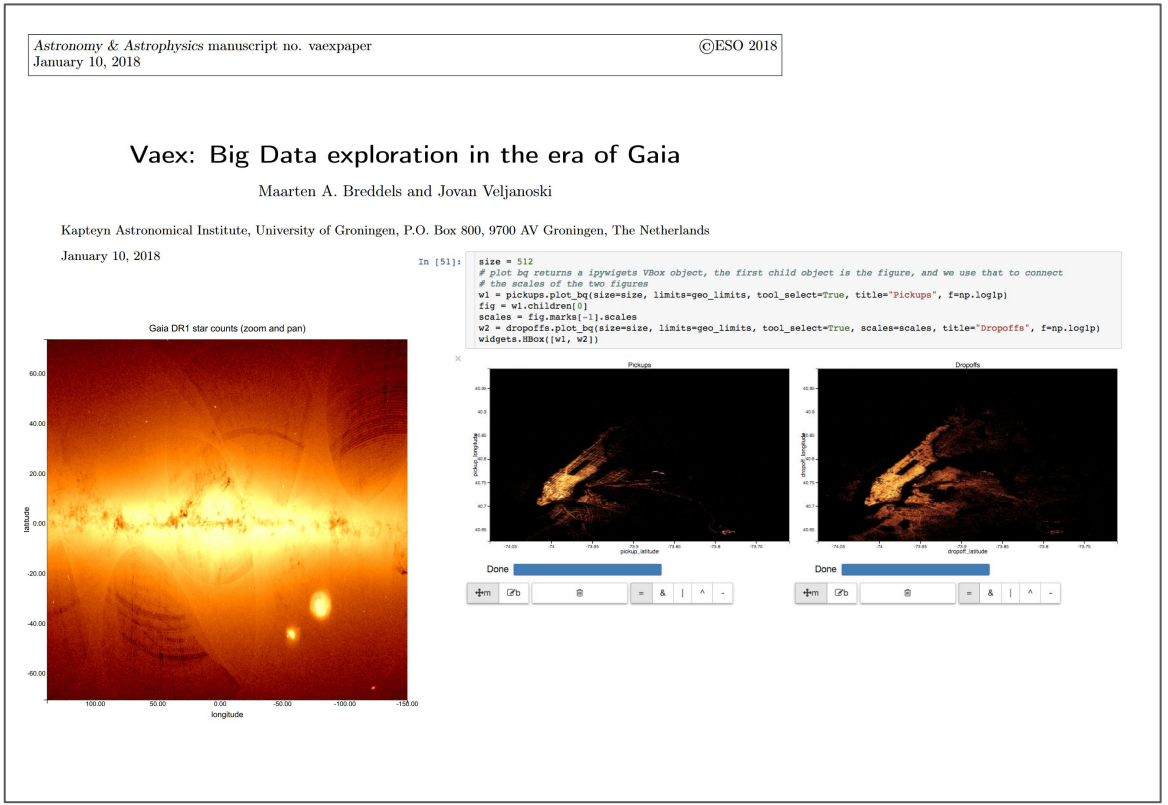
Jupyter in Astronomy

- 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, SciServer, 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, SciServer, etc)
- Tools and extensions developed by/for astronomers



Astronomy & Astrophysics manuscript no. vaexpaper
January 10, 2018

©ESO 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])
```

Gaia DR1 star counts (zoom and pan)

Pickups

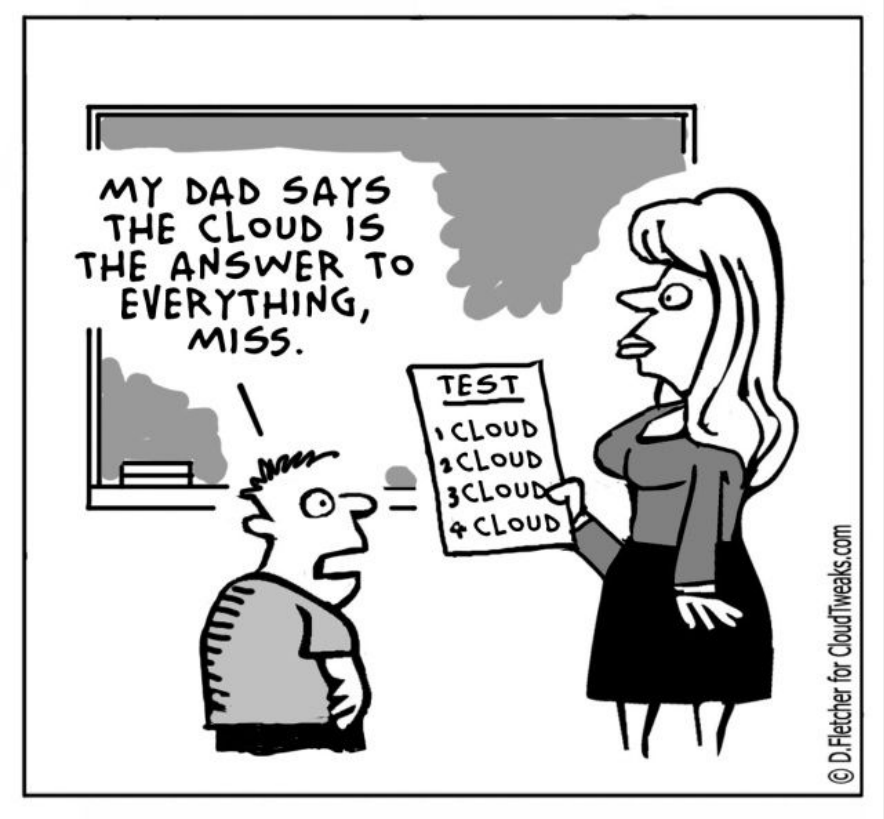
Dropoffs

Done

Done

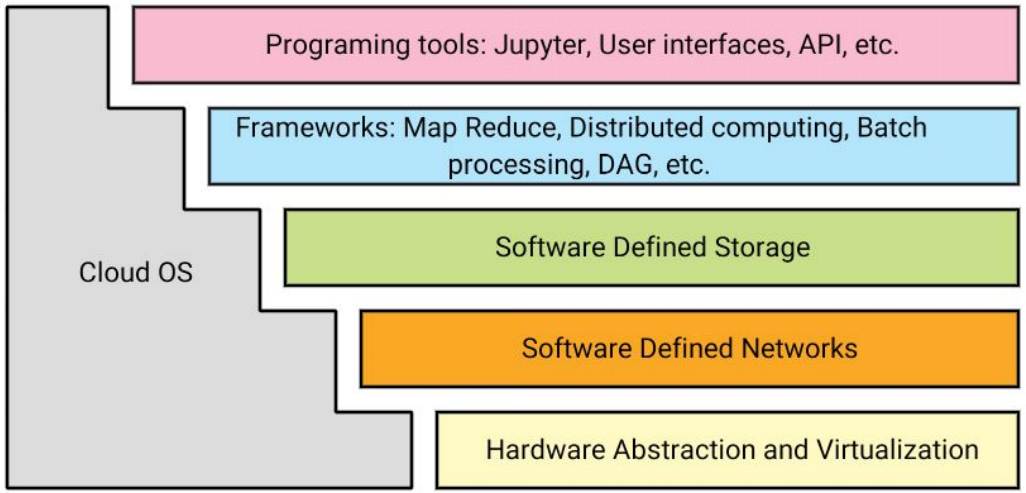
Scientific Cloud Computing

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



Why we should be doing science on the cloud

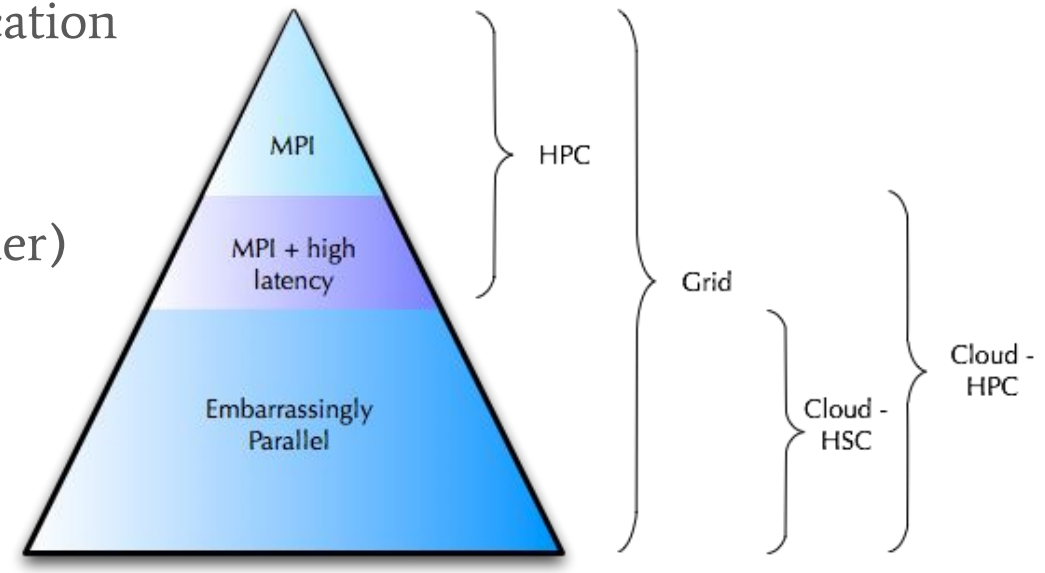
- Remote and dynamic data (!= Big data)
- Big data ⇒ Data Gravity, Data Lake, etc.
- Remote software/server
- Easy to deploy*
- Asynchronous
- Web applications / Shareable
- Serverless applications
- Federation of Services
- Tablets/ChromeOS
- more...



*arguable

Why we shouldn't be doing science on the cloud

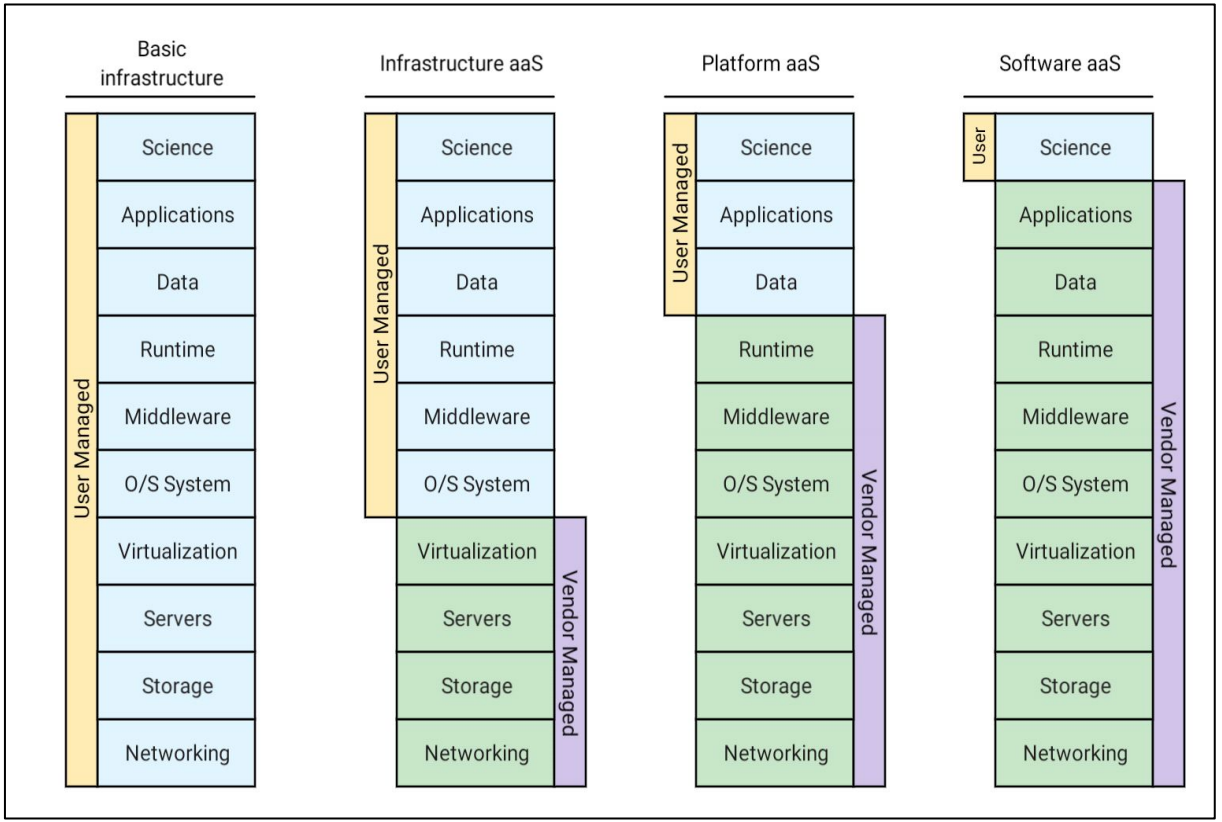
- Because there is no a real reason for it^
- HPC is not there yet, large latencies and bad bisection bandwidth
  ... but HPC is adopting cloud technologies
- Full control on data and application
- Security concerns*
- Faster development*
- Billing (if a commercial provider)
- more ...



^arguable
*arguable (CI, CD)

What kind of science/projects? → Which model

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



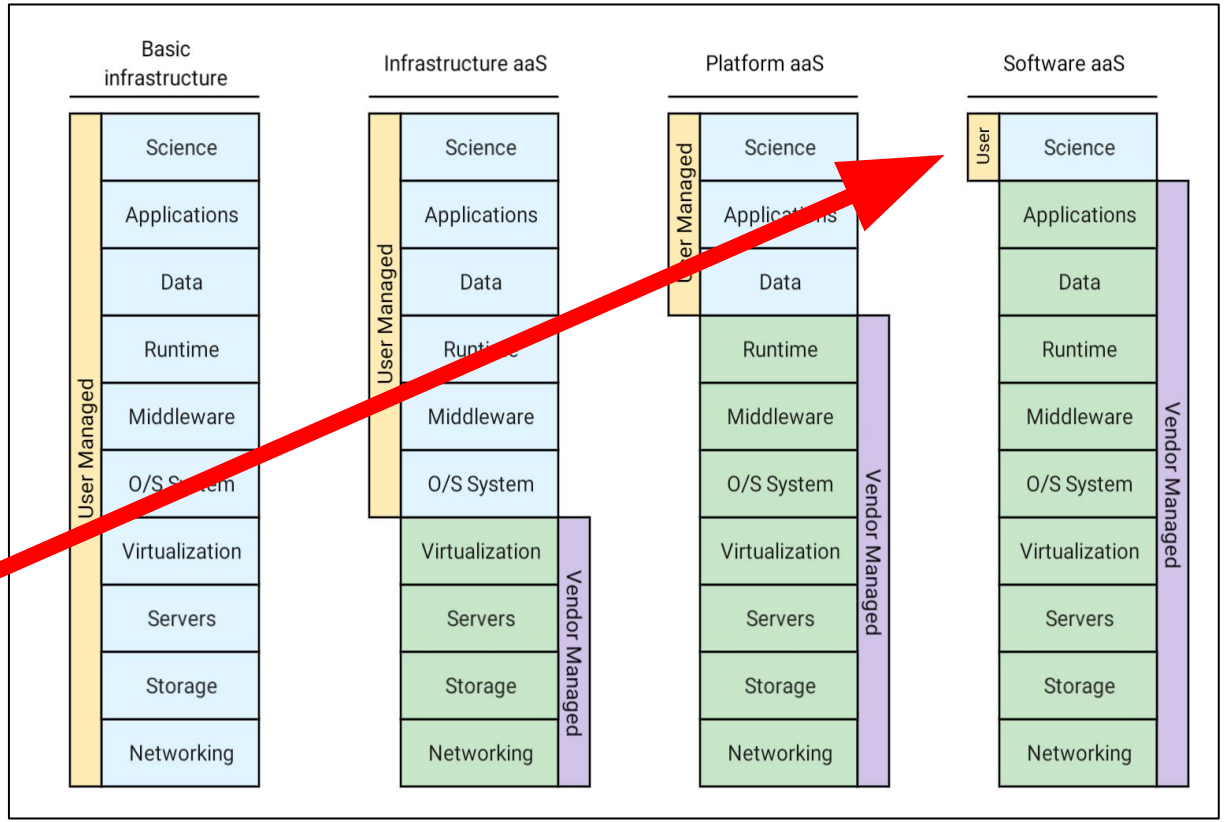
*arguable

What kind of science/projects? → Which model

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

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

*arguable



Which Clouds?

Amazon Web Services (AWS) – 40%

Microsoft Azure – about 50% of AWS

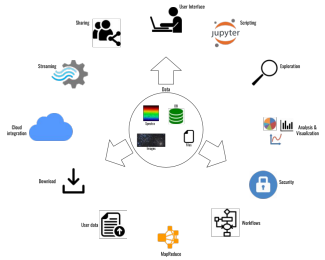
Google Cloud – 3rd place

IBM Bluemix – growing VERY fast

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

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

Why not both?



Scale on demand

User AWS account

Offered services

Processing

Local

Cloud

OSG

HPC

VMs

Remote storage

User Applications

Applications, services and user interfaces

NFS, GridFTP, Globus, etc

Edge VMs, Services

Storage and Storage Services

What kind of users/groups?

- Runaway/Abusive users (sleep infinity / over subscribed)
- Superuser a. → Data-In Intensive Jobs (e.g., ML)
- Superuser b. → Data-Out Intensive Jobs (e.g., Sims)
- Superuser c. → Resource Intensive Jobs (e.g., Modeling/Fitting)
- Super user d. → Any combination of the above
- User with no exposure or experience (Need good documentation/tutorial, Outreach)
- Users provisioning hardware
- Users with money/cloud credits to provide
- Ephemeral users (workshops, demos)
- “Normal” users

CLOUD NATIVE COMPUTING FOUNDATION

CLOUD NATIVE TRAIL MAP

The Cloud Native Landscape l.cncf.io has a large number of options. This Cloud Native Trail Map is a recommended process for leveraging open source, cloud native technologies. At each step, you can choose a vendor-supported offering or do it yourself, and everything after step #3 is optional based on your circumstances.

HELP ALONG THE WAY

A. Training and Certification

Consider training offerings from CNCF and then take the exam to become a Certified Kubernetes Administrator or a Certified Kubernetes Application Developer

cncf.io/training

B. Consulting Help

If you want assistance with Kubernetes and the surrounding ecosystem, consider leveraging a Kubernetes Certified Service Provider

cncf.io/kcsp

C. Join CNCF's End User Community

For companies that don't offer cloud native services externally

cncf.io/enduser

WHAT IS CLOUD NATIVE?

Cloud native technologies empower organizations to build and run scalable applications in modern, dynamic environments such as public, private, and hybrid clouds. Containers, service meshes, microservices, immutable infrastructure, and declarative APIs exemplify this approach.

These techniques enable loosely coupled systems that are resilient, manageable, and observable. Combined with robust automation, they allow engineers to make high-impact changes frequently and predictably with minimal toil.

The Cloud Native Computing Foundation seeks to drive adoption of this paradigm by fostering and sustaining an ecosystem of open source, vendor-neutral projects. We democratize state-of-the-art patterns to make these innovations accessible for everyone.

l.cncf.io

v20181206

1. CONTAINERIZATION

- Commonly done with Docker containers
- Any size application and dependencies (even PDP-11 code running on an emulator) can be containerized
- Over time, you should aspire towards splitting suitable applications and writing future functionality as microservices

2. CI/CD

- Setup Continuous Integration/Continuous Delivery (CI/CD) so that changes to your source code automatically result in a new container being built, tested, and deployed to staging and eventually, perhaps, to production
- Setup automated rollouts, roll backs and testing

3. ORCHESTRATION & APPLICATION DEFINITION

- Kubernetes is the market-leading orchestration solution
- You should select a Certified Kubernetes Distribution, Hosted Platform, or Installer: cncf.io/ck
- Helm Charts help you define, install, and upgrade even the most complex Kubernetes application

kubernetes CNCF Graduated

HELM CNCF Incubating

4. OBSERVABILITY & ANALYSIS

- Pick solutions for monitoring, logging and tracing
- Consider CNCF projects Prometheus for monitoring, Fluentd for logging and Jaeger for Tracing
- For tracing, look for an OpenTracing-compatible implementation like Jaeger

Prometheus CNCF Graduated

fluentd CNCF Incubating

OPENTRACING CNCF Incubating

JAEGER CNCF Incubating

5. SERVICE PROXY, DISCOVERY, & MESH

- CoreDNS is a fast and flexible tool that is useful for service discovery
- Envoy and Linkerd each enable service mesh architectures
- They offer health checking, routing, and load balancing

envoy CNCF Graduated

CoreDNS CNCF Incubating

LINKERD CNCF Incubating

6. NETWORKING

To enable more flexible networking, use a CNI-compliant network project like Calico, Flannel, or Weave Net.

CNI CNCF Incubating

7. DISTRIBUTED DATABASE & STORAGE

When you need more resiliency and scalability than you can get from a single database, Vitess is a good option for running MySQL at scale through sharding. Rook is a storage orchestrator that integrates a diverse set of storage solutions into Kubernetes.

Vitess CNCF Incubating

ROOK CNCF Incubating

8. STREAMING & MESSAGING

When you need higher performance than JSON-REST, consider using gRPC or NATS. gRPC is a universal RPC framework. NATS is a multi-modal messaging system that includes request/reply, pub/sub and load balanced queues.

gRPC CNCF Incubating

NATS CNCF Incubating

9. CONTAINER REGISTRY & RUNTIME

Harbor is a registry that stores, signs, and scans content. You can use alternative container runtimes. The most common, all of which are OCI-compliant, are containerd, rkt and CRI-O.

HARBOR CNCF Incubating

containerd CNCF Incubating

rkt CNCF Incubating

10. SOFTWARE DISTRIBUTION

If you need to do secure software distribution, evaluate Notary, an implementation of The Update Framework.

Notary CNCF Incubating

TUF CNCF Incubating

Cloud Native Landscape

2018-12-06T16:57:37Z 503faf2

See the interactive landscape at l.cncf.io

Greyed logos are not open source

App Definition and Development

Database

CNCF Incubating: Vitess

Streaming & Messaging

CNCF Incubating: NATS

Application Definition & Image Build

CNCF Incubating: HELM

Continuous Integration & Delivery

Orchestration & Management

Scheduling & Orchestration

CNCF Graduated: kubernetes

Coordination & Service Discovery

CNCF Incubating: CoreDNS

Remote Procedure Call

CNCF Incubating: GRPC

Service Proxy

CNCF Graduated: envoy

API Gateway

Service Mesh

CNCF Incubating: LINKERD

Runtime

Cloud-Native Storage

CNCF Incubating: ROOK

Container Runtime

CNCF Incubating: cri-o, rkt

Cloud-Native Network

CNCF Incubating: CNI

Provisioning

Automation & Configuration

Container Registry

CNCF Incubating: HARBOR

Security & Compliance

CNCF Incubating: Notary, TUF

Key Management

Cloud

Public

Platform

Certified Kubernetes - Distribution

Certified Kubernetes - Hosted

Certified Kubernetes - Installer

PaaS/Container Service

Observability and Analysis

Monitoring

CNCF Graduated: Prometheus

Logging

CNCF Incubating: fluentd

Tracing

CNCF Incubating: JAEGER, OPENTRACING

Chaos Engineering

Serverless

Special

Kubernetes Certified Service Provider

Kubernetes Training Partner

This landscape is intended as a map through the previously uncharted terrain of cloud native technologies. There are many routes to deploying a cloud native application, with CNCF Projects representing a particularly well-traveled path

l.cncf.io

CLOUD NATIVE Landscape

CLOUD NATIVE COMPUTING FOUNDATION

Redpoint Amplify

Serverless Cloud Native Landscape

2018-12-06T16:57:37Z 503faf2

See the serverless interactive display at s.cncf.io

Greyed logos are not open source

Tools

CLOUDZERO, dashbird, Epsagon, event gateway, Gloo, IO|pipe, Iron.io, Node Lambda, python-λ, SCAR, SIGMA, STACKERY, THUNDRA

Security

intrinsic, Protego, PURESEC, snyk, threat stack

Framework

APEX, Architect, aws Chalice, AWS SAM, Claudia.js, FLOGO, gun.io, serverless, SPARTA, Spring Cloud Function

Hosted

Installable

Platform

ALGORITHMIA, Alibaba Cloud Function Compute, AWS Lambda, Azure Functions, BINARIS, cloudboost, CLOUDFLARE, Google Cloud Functions, HUAWEI, HYPER.SH, IBM Cloud, netlify, PubNub FUNCTIONS, spotinst, stdlib, Tencent Cloud, twilio, ZEIT

OpenWhisk, AppScale, DISPATCH, fission, fn, GALACTIC FOG, Knative, Kubeless, Kyma, LunchBadger, nuclio, OPENFAAS, PipelineAI, riff, Virtual Kubelet

s.cncf.io

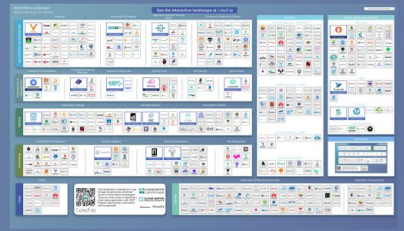
Serverless computing refers to a new model of cloud native computing, enabled by architectures that do not require server management to build and run applications. This landscape illustrates a finer-grained deployment model where applications, bundled as one or more functions, are uploaded to a platform and then executed, scaled, and billed in response to the exact demand needed at the moment

CLOUD NATIVE Landscape

CLOUD NATIVE COMPUTING FOUNDATION

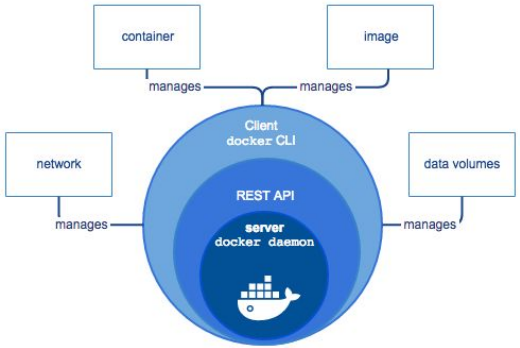
Redpoint

Cloud Native Landscape



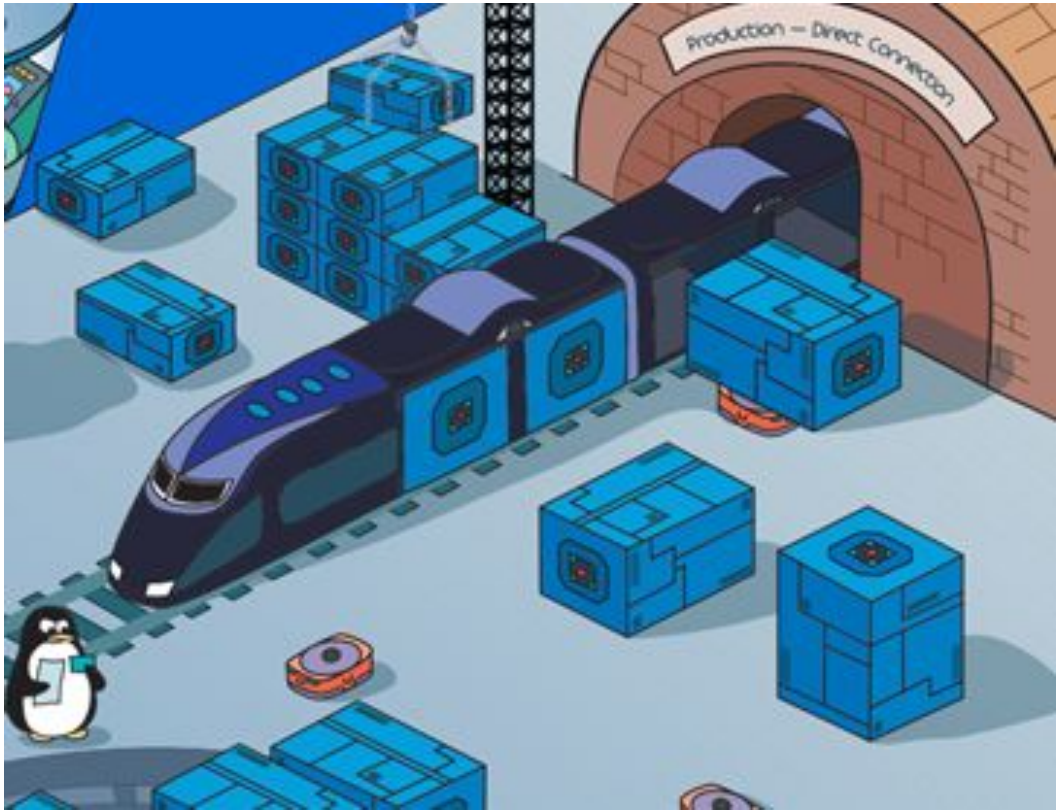
Containerization to the rescue

- 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

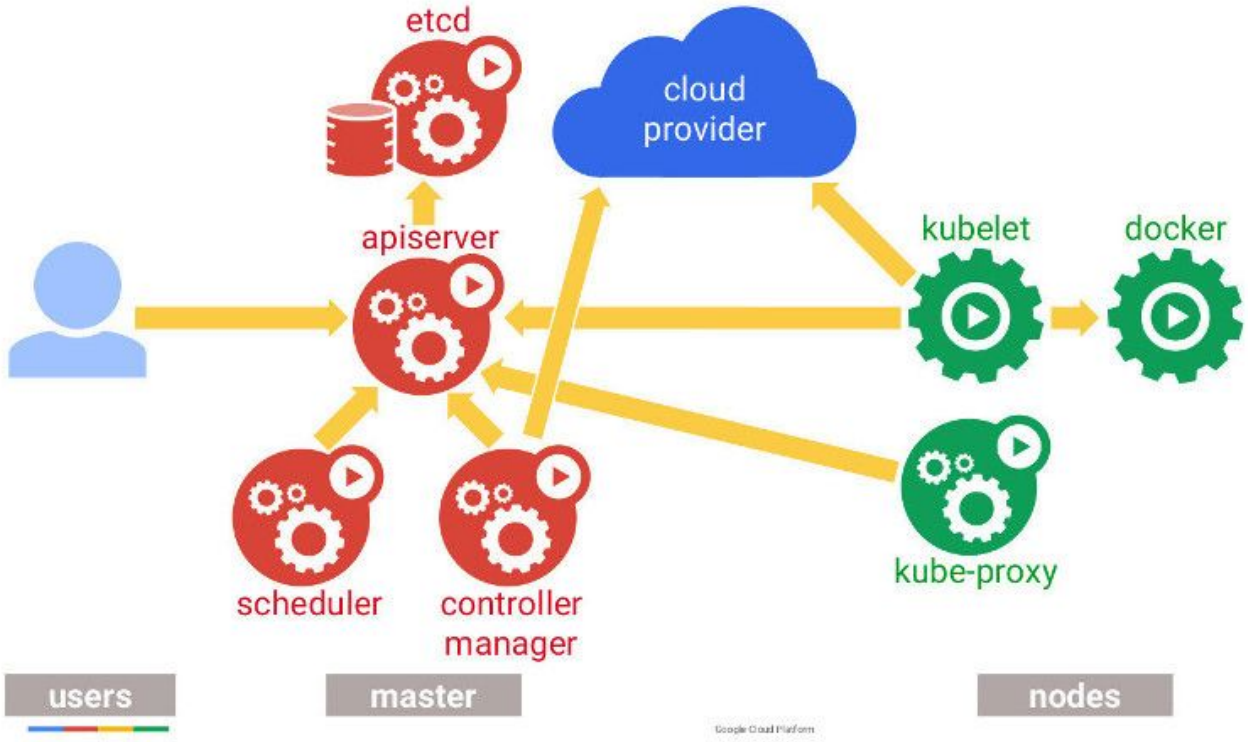


The Kubernetes Factor

- 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

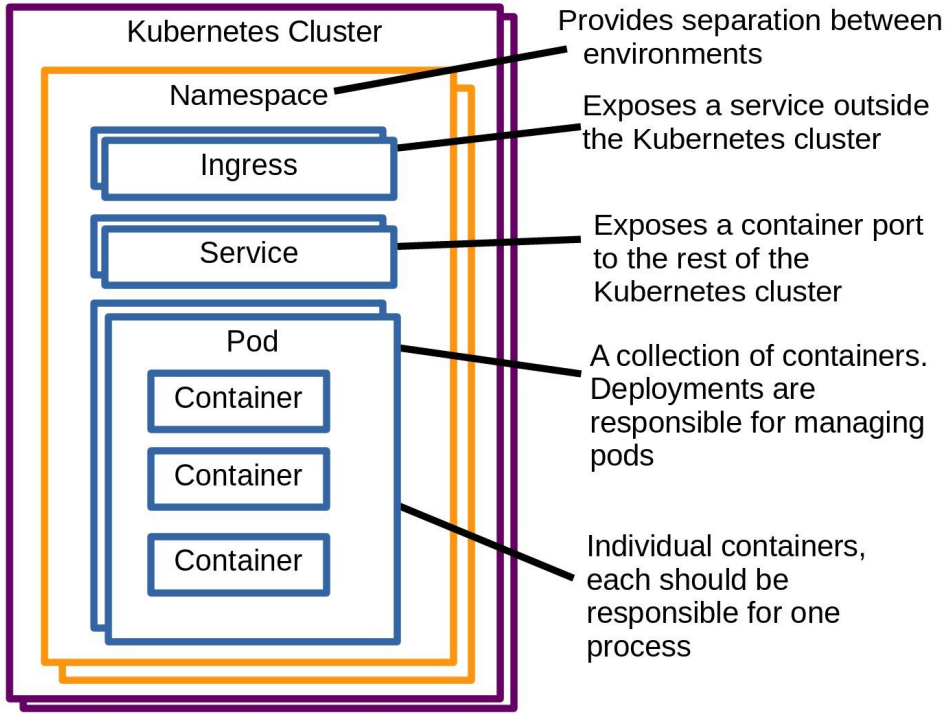
Kubernetes Overview

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

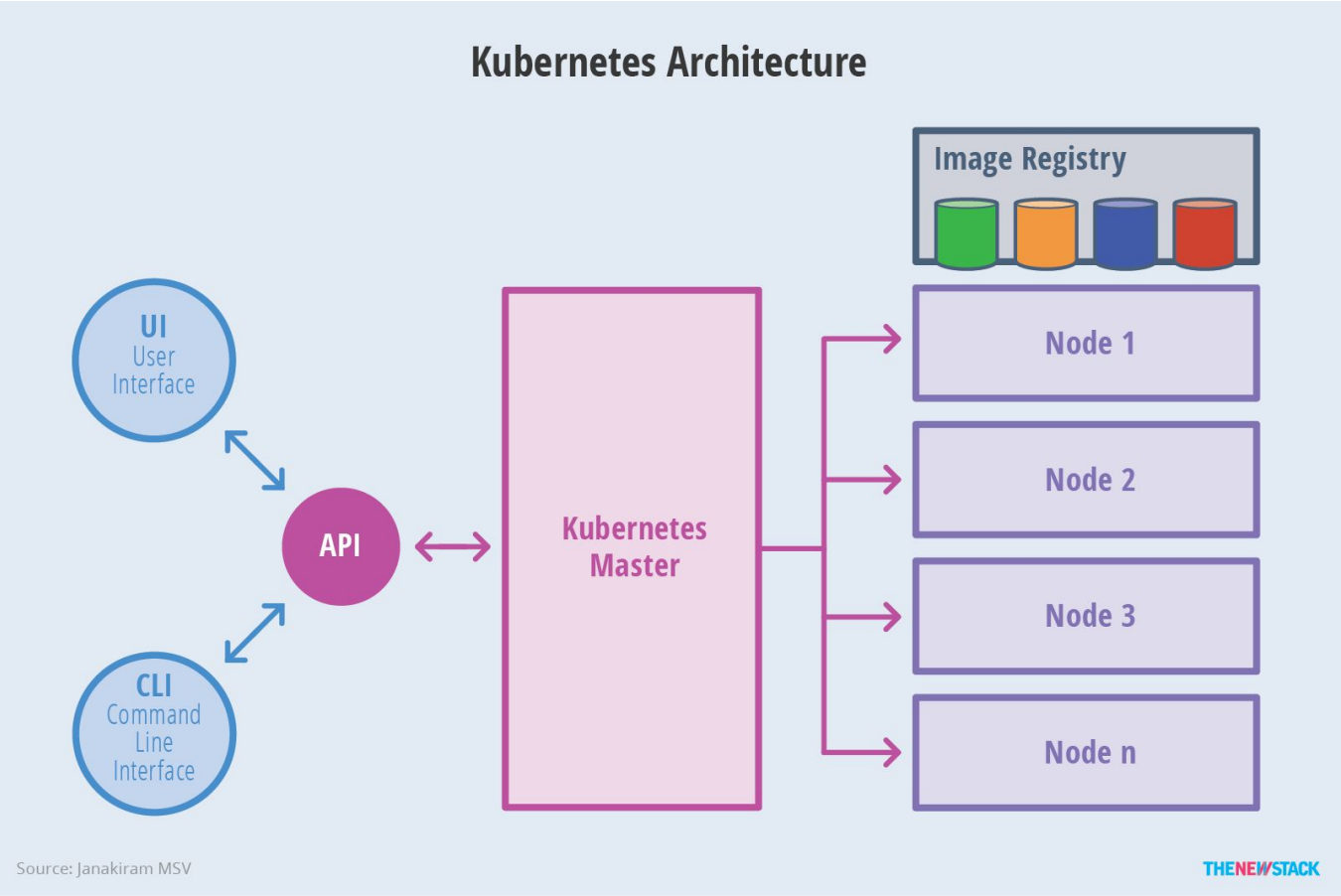


Kubernetes Key Concepts

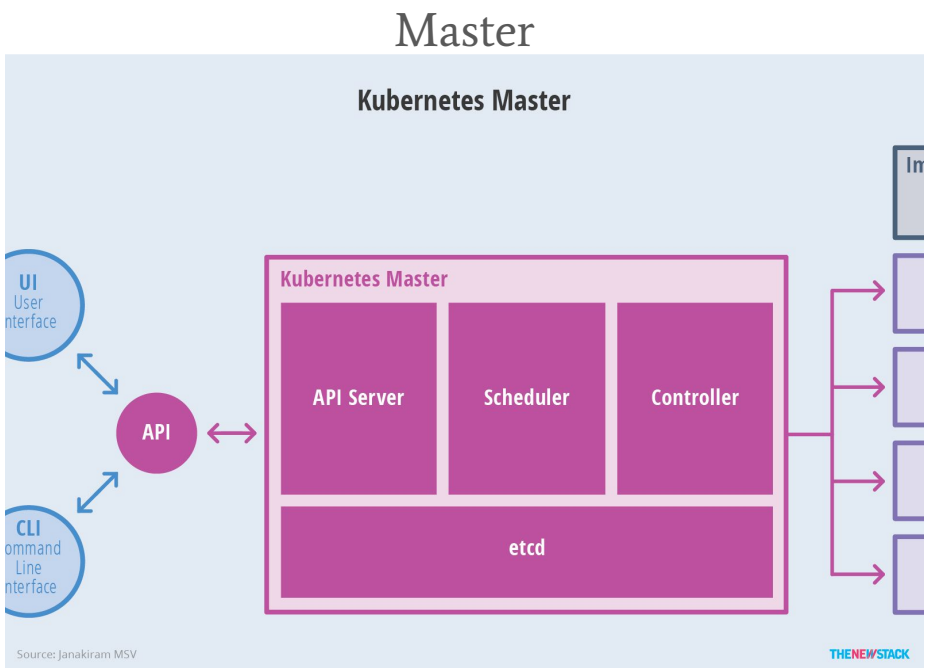
- **Pod** - A group of Containers
- **Labels** - Labels for identifying pods
- **Kubelet** - Container Agent
- **Proxy** - A load balancer for Pods
- **etcd** - A metadata service
- **cAdvisor** - Container Advisor provides resource usage/performance statistics
- **Replication Controller** - Manages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

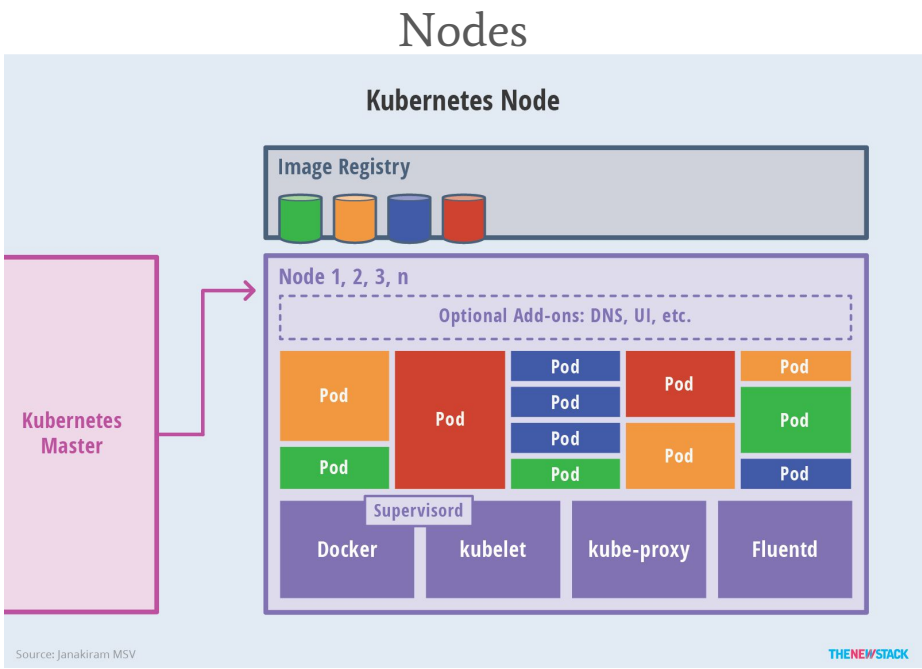


The Kubernetes Architecture



The Kubernetes Architecture





Applications

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

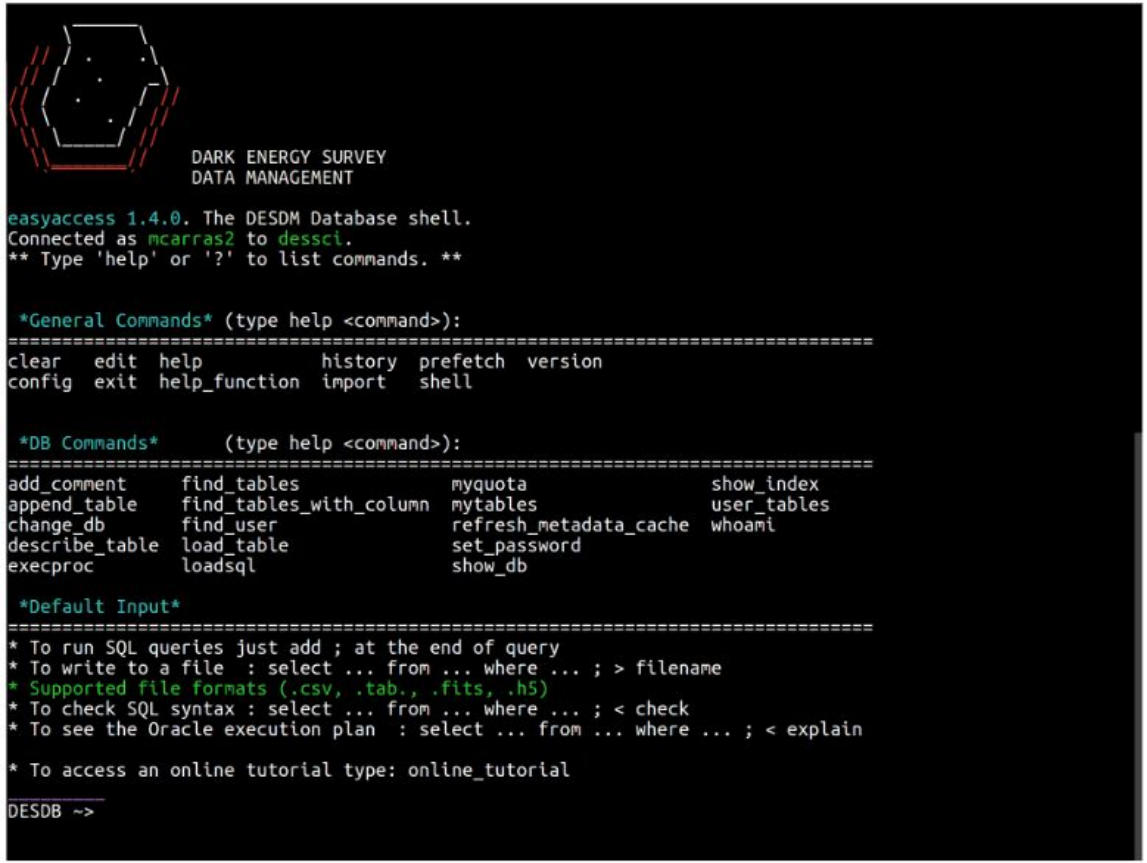
The DES Data Access

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



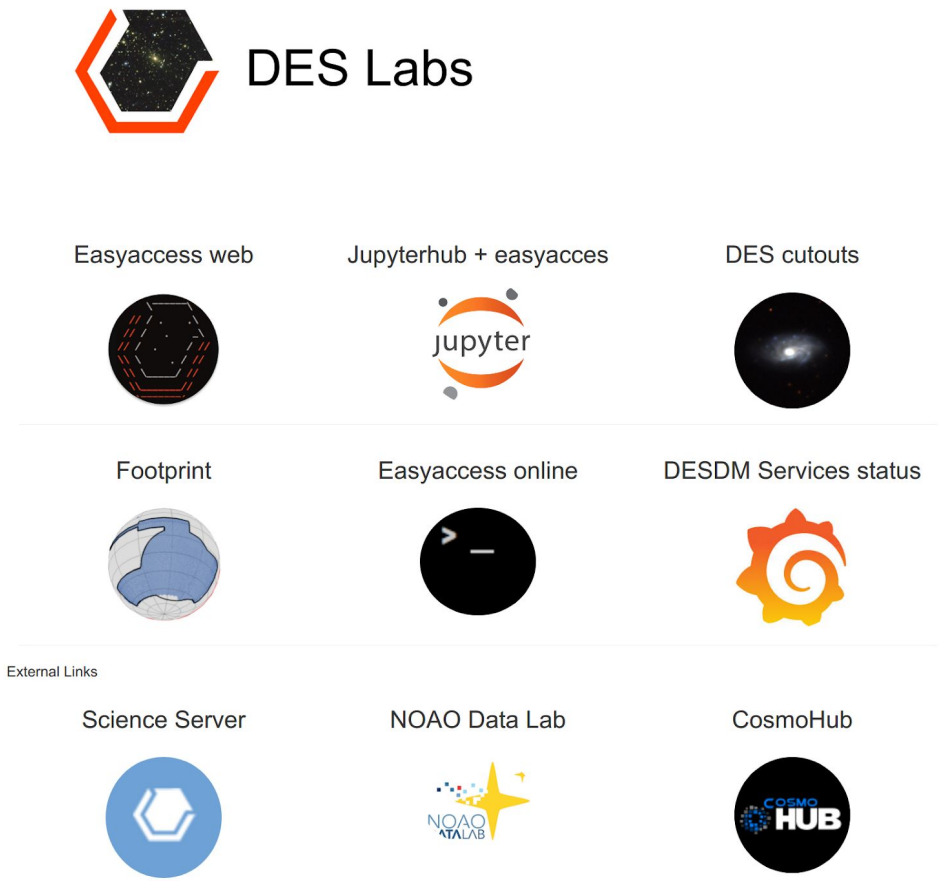
- DES DB in Oracle
- Specifically designed for DES (internal and public)
- Enhanced SQL command line interpreter in Python
- Astronomer friendly
- Python API, web interface
- There are many other CLI and GUI clients.
- Needed a simple tool, easy to use and install
- Autocompletion
- Load/Save to hdf5,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

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



NCSA DESaccess: Services

DARK ENERGY SURVEY desaccess

mcarras2
mcarras2@ncsa.illinois.edu

- Home
- DB access
- DES Table Schema
- Example Queries
- Cutouts Service
- DES JupyterLab
- Finding Chart
- DES Footprint
- Data Analysis
- My Jobs
- Help

Welcome to dessci, Matias!

DB ACCESS

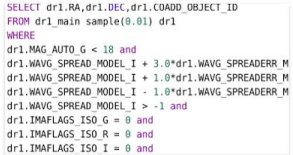
Oracle SQL web- client

More...

DES TABLE SCHEMA

Browse all tables

More...



```
SELECT dr1.RA,dr1.DEC,dr1.COADD_OBJECT_ID
FROM dr1_main sample(0.01) dr1
WHERE
dr1.MAG_AUTO_G < 18 and
dr1.WAVG_SPREAD_MODEL_I + 3.0*dr1.WAVG_SPREADERR_M
dr1.WAVG_SPREAD_MODEL_I + 1.0*dr1.WAVG_SPREADERR_M
dr1.WAVG_SPREAD_MODEL_I - 1.0*dr1.WAVG_SPREADERR_M
dr1.WAVG_SPREAD_MODEL_I > -1 and
dr1.IMAFLAGS_ISO_G = 0 and
dr1.IMAFLAGS_ISO_R = 0 and
dr1.IMAFLAGS_ISO_I = 0 and
```

EXAMPLE QUERIES

See some example queries as a start

More...

CUTOUTS SERVICE

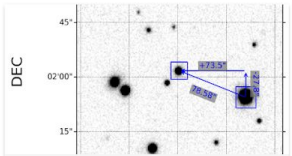
Generate cutouts for positions or ids

More...

DES JupyterLabs

(Beta) Jupyter Labs

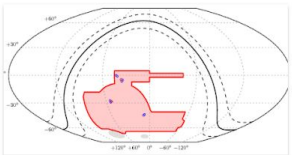
More...



FINDING CHART

Find your object

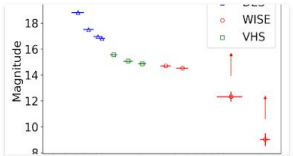
More...



DES FOOTPRINT

Interactive globe

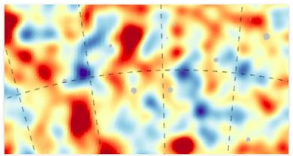
More...



DATA ANALYSIS

SEDs and color-color diagrams

More...



MY JOBS

List of submitted jobs

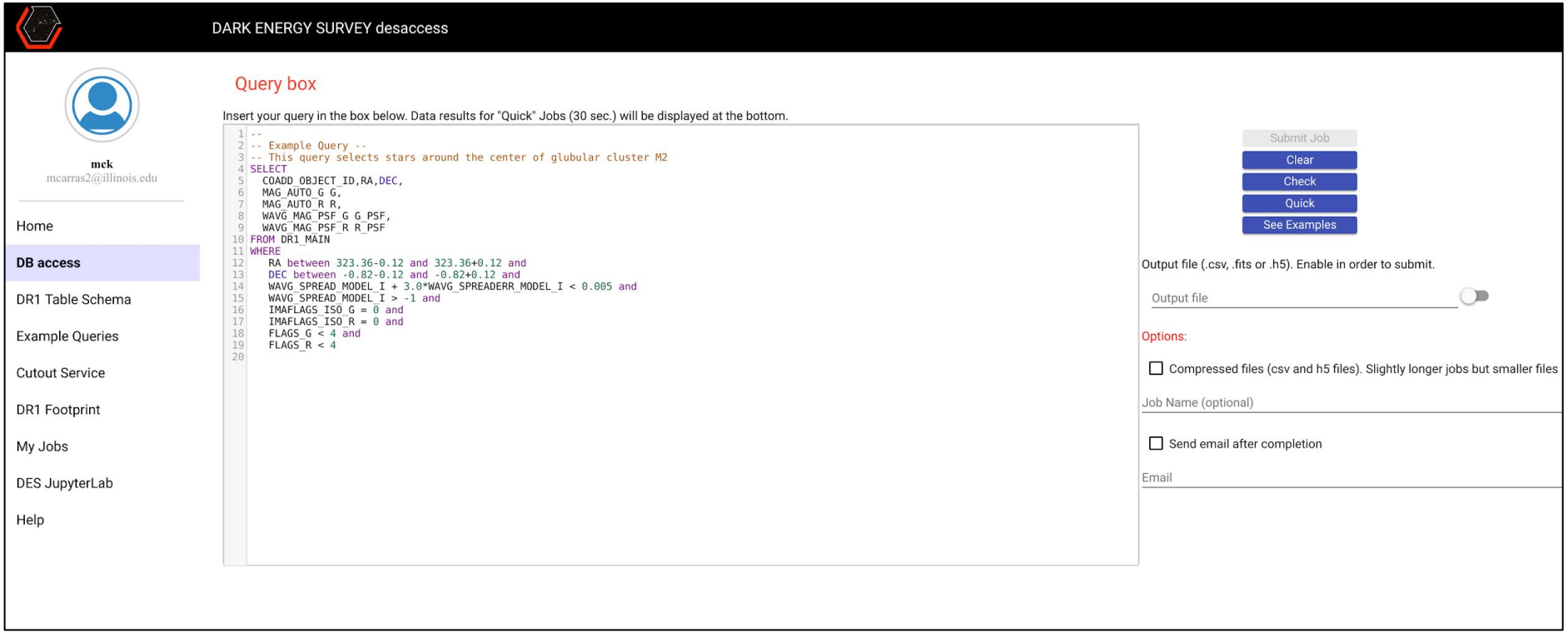
More...

HELP

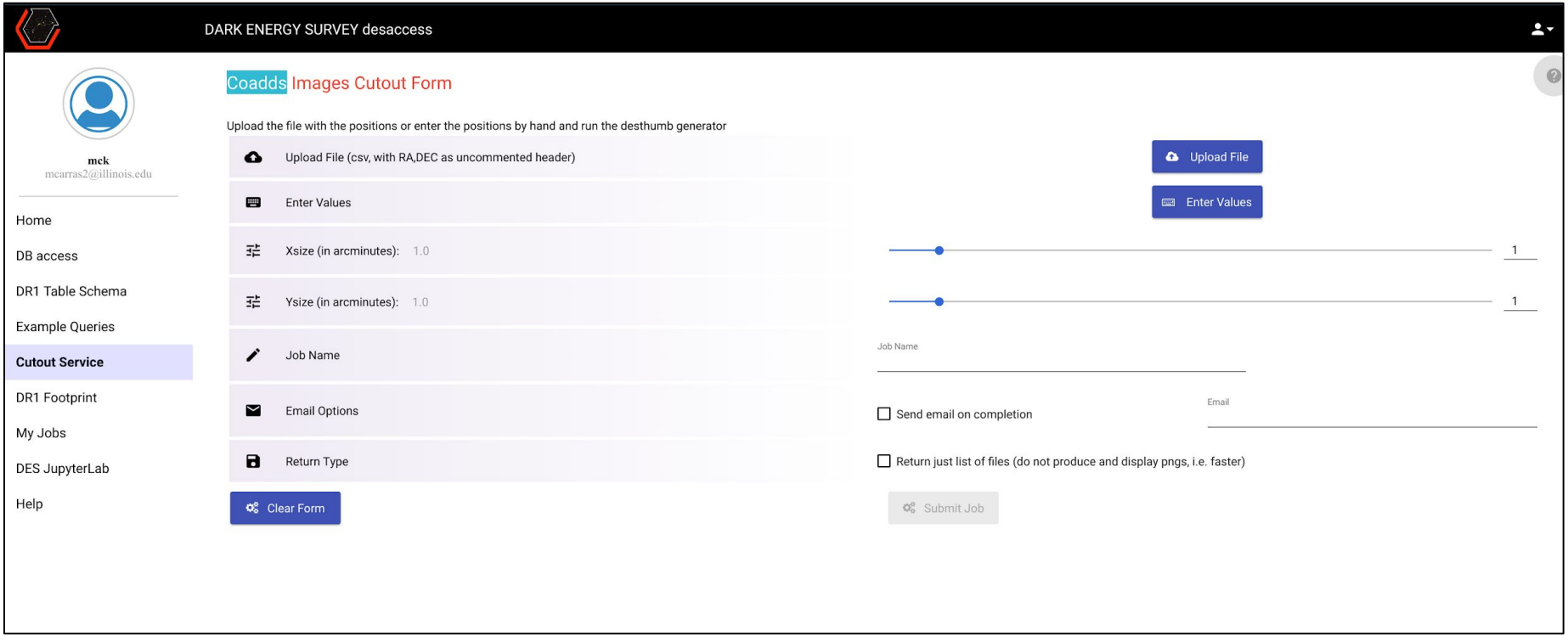
Help form

More...

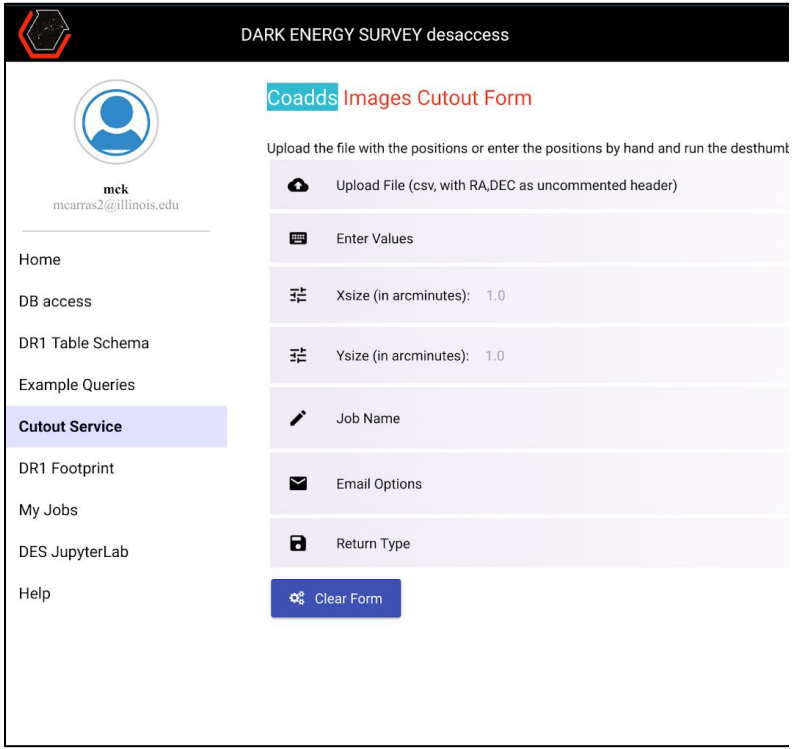
NCSA DESaccess: DB access



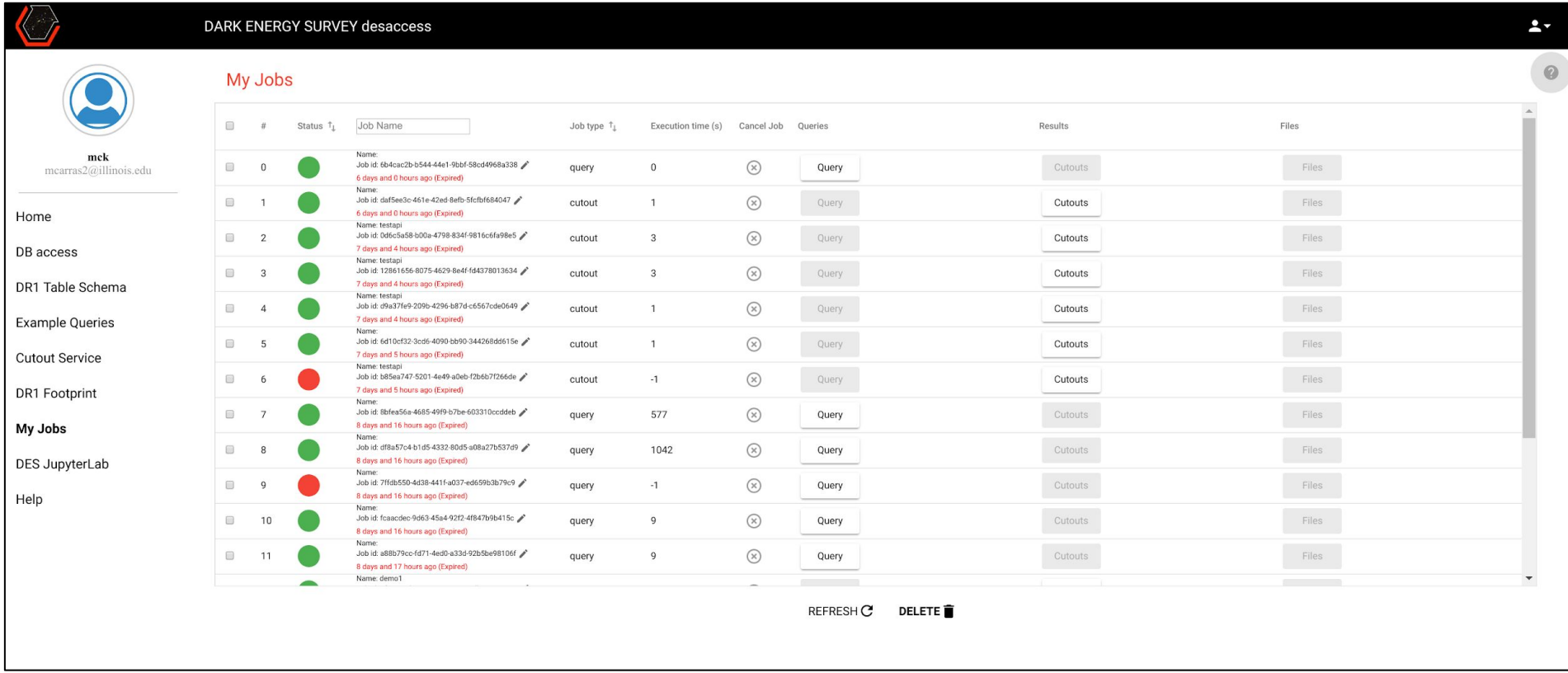
NCSA DESaccess: Cutouts Service



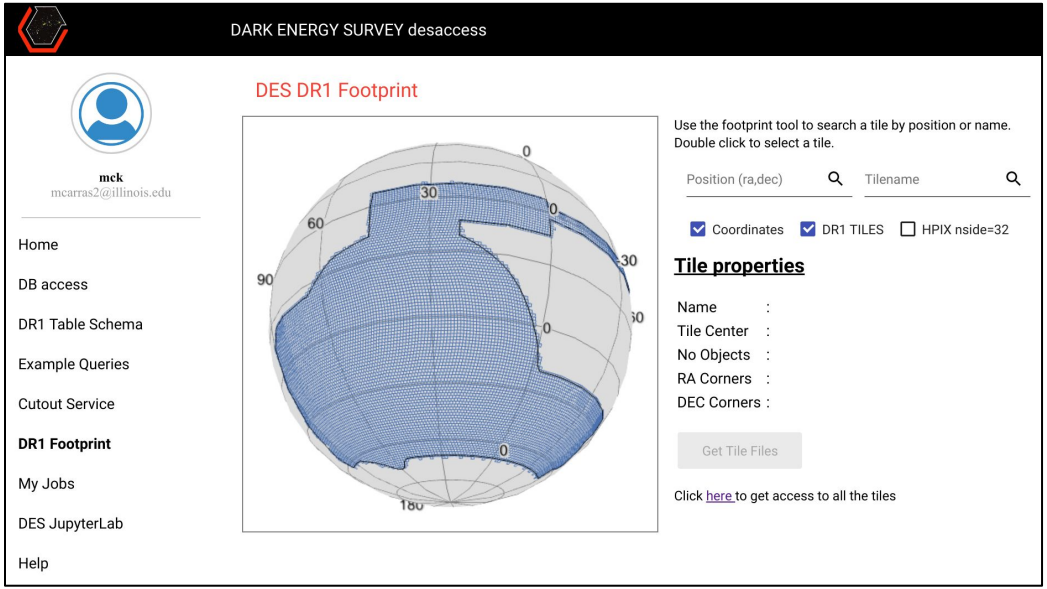
NCSA DESaccess: Cutouts Service

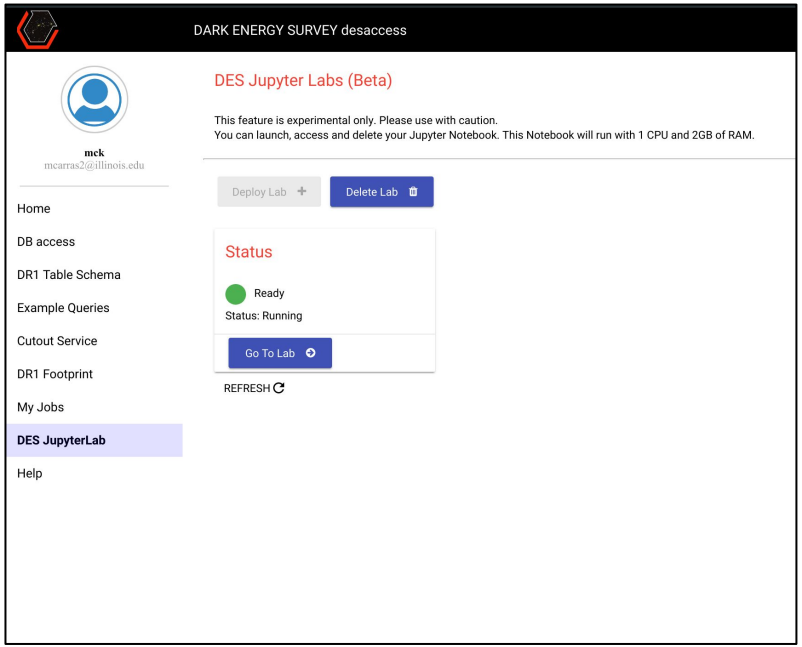


NCSA DESaccess: Asynchronous Jobs

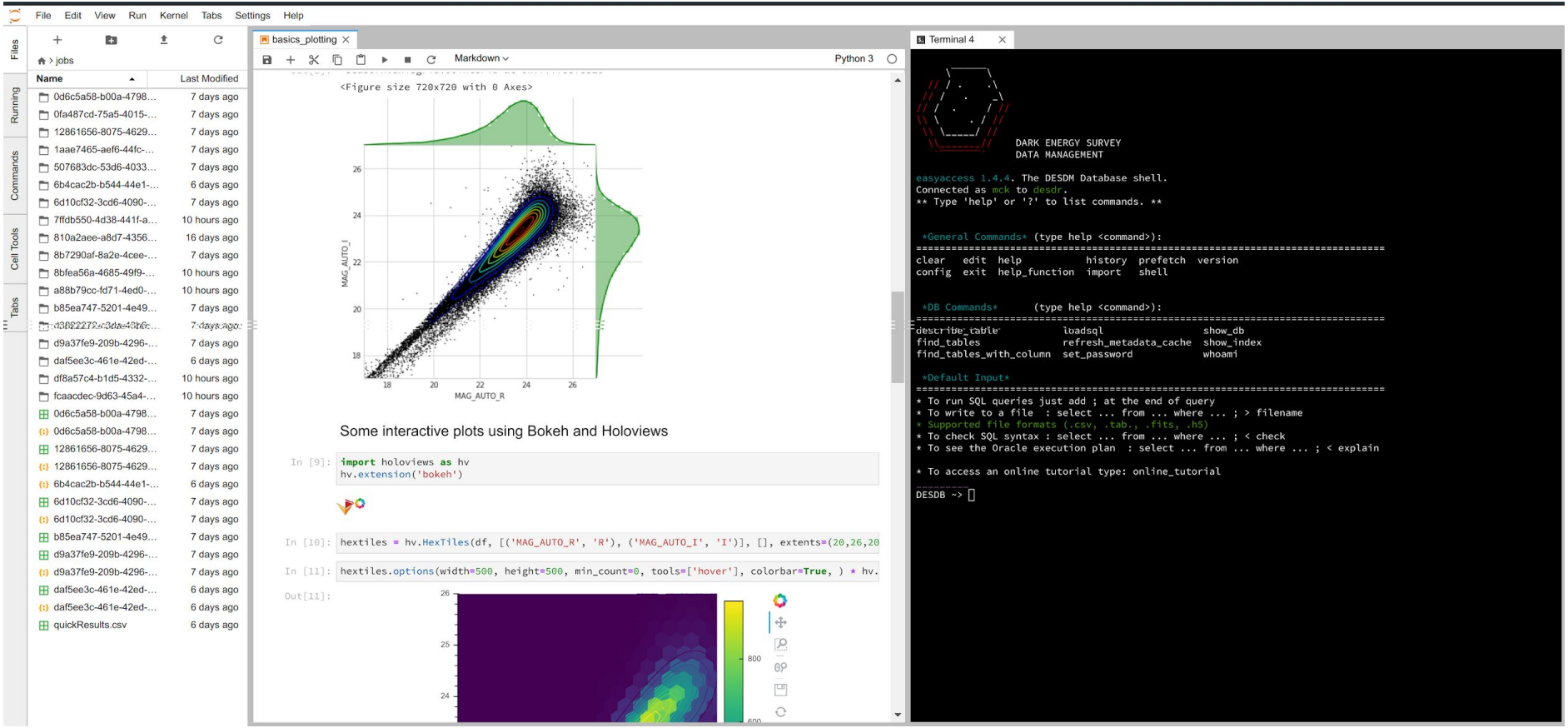


NCSA DESaccess: Footprint and Jupyter Labs

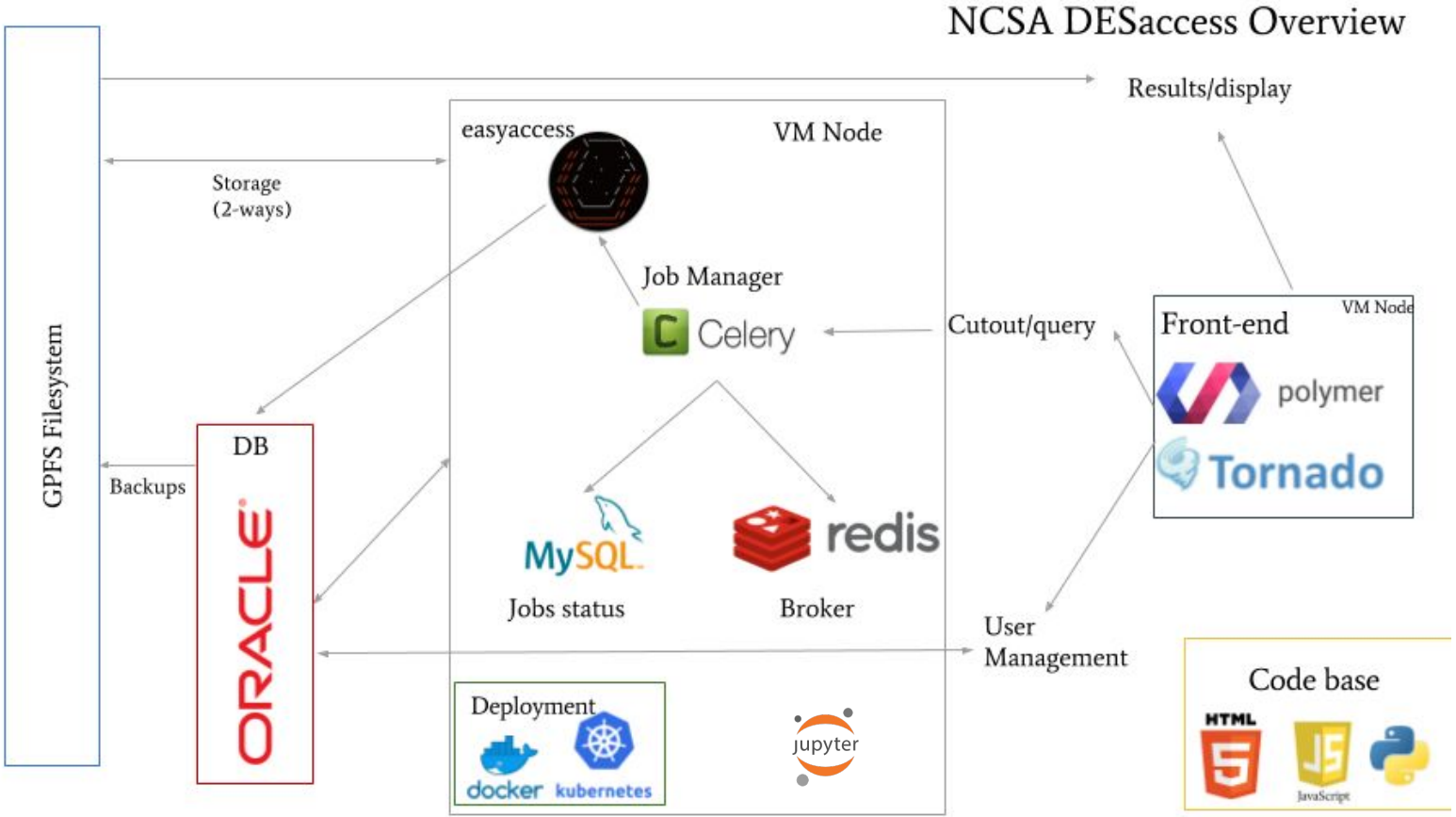




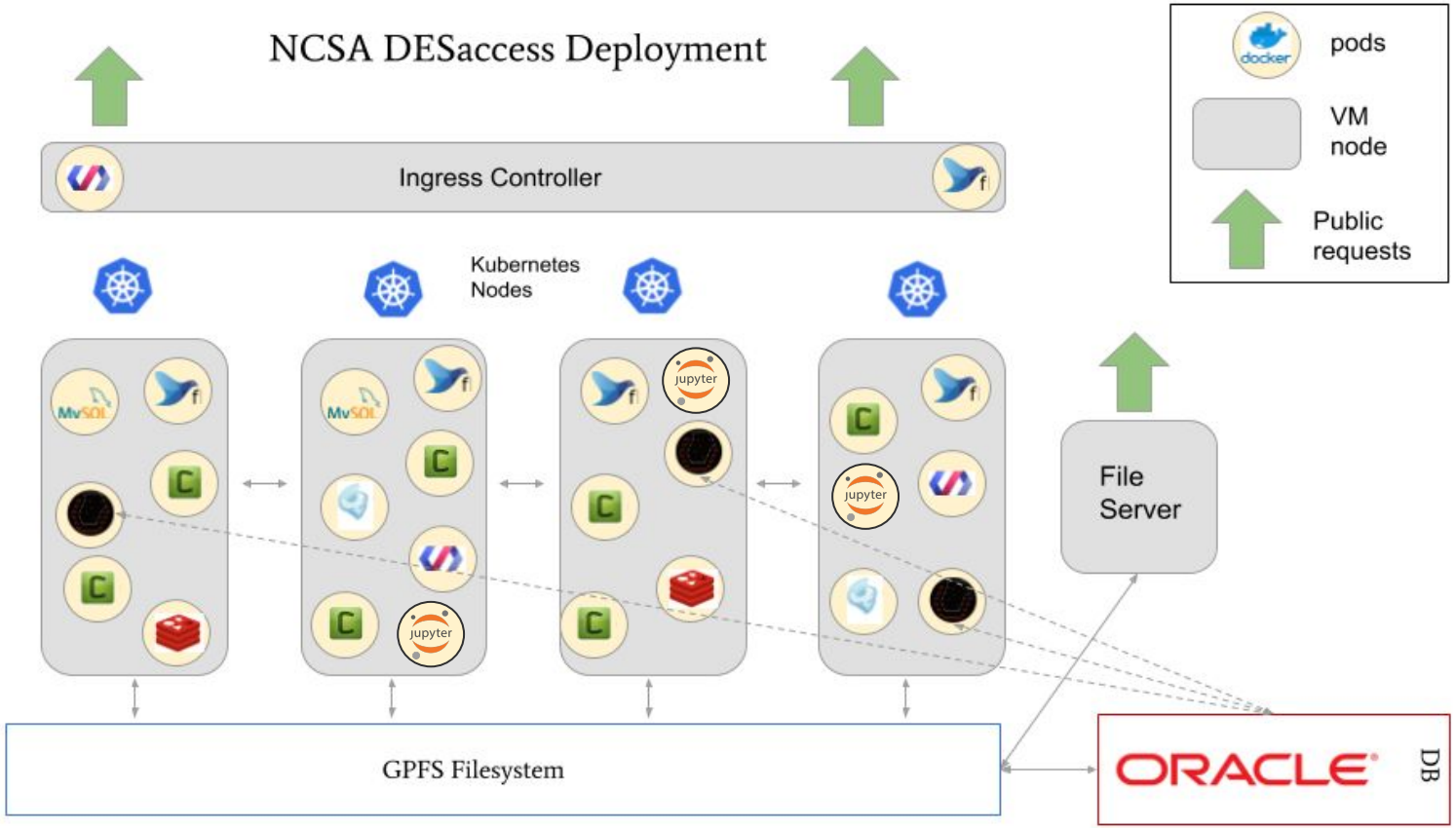
NCSA DESaccess: Labs with access to Jobs and easyaccess



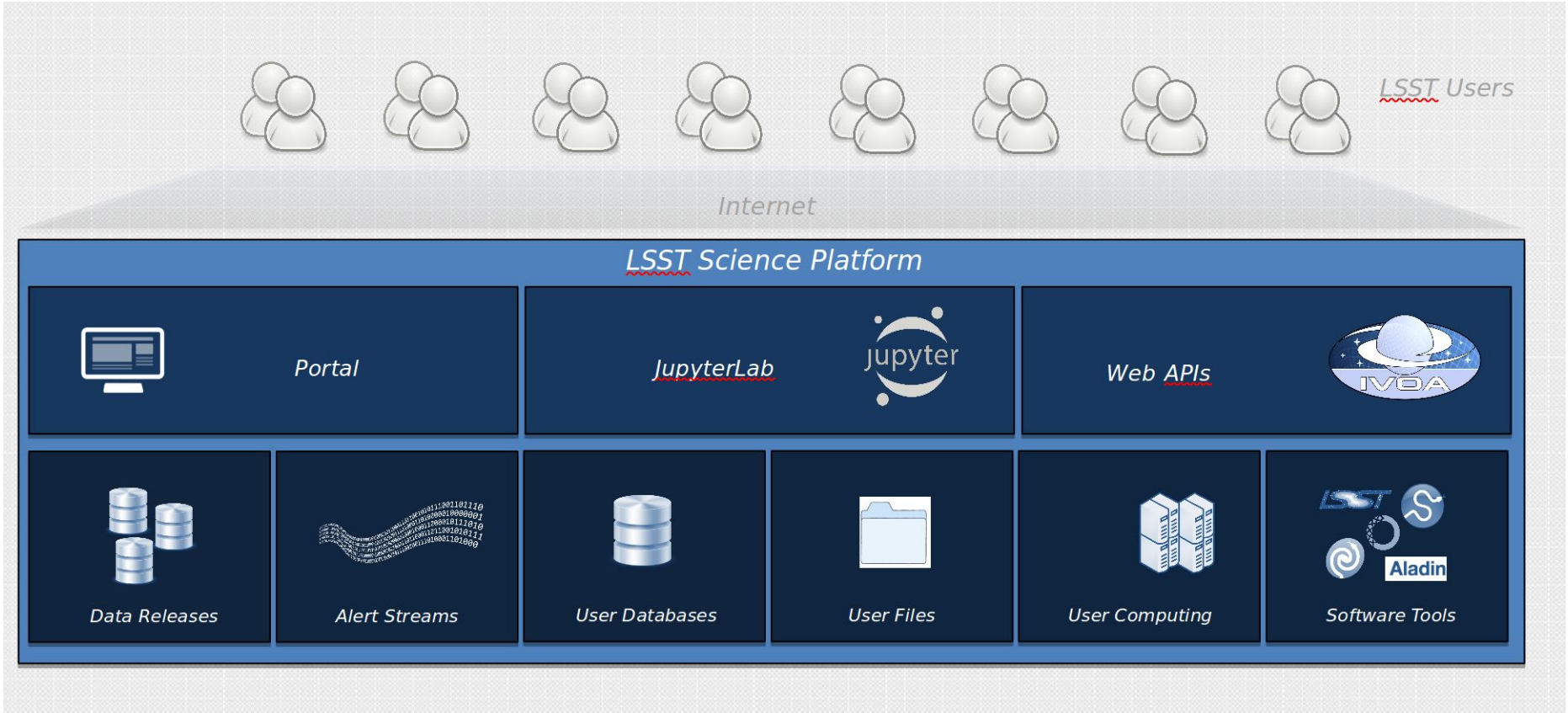
NCSA DESaccess: Technology Overview



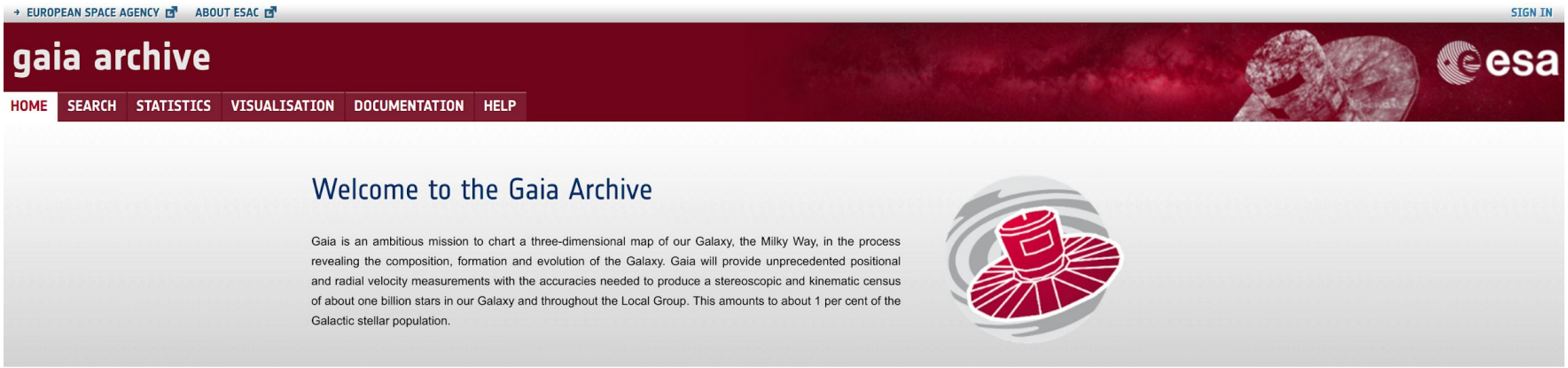
NCSA DESacces: Deployment



LSST Science Platform



GAIA Archive



SciServer


SciServer
Collaborative data-driven science
A new vision for science
A collaborative research environment for large-scale data-driven science
SciServer Betelgeuse v2.0.3
Login to SciServer
About
Bringing the Analysis to the Data
Tools
A modular system of independent components.
Hosted Datasets
SciServer makes a number of datasets available online in the…
Science
Full datasets
Common formats
Common sets of interfaces
Education
Building relationships with universities, institutes, and government organizations.
Help
Workshop supporting material, online documentation, provide feedback, report bugs.
SciServer is administered by
idies
JOHNS HOPKINS UNIVERSITY
SciServer is funded by National Science Foundation award ACI-1261715
NSF

SClaaS Example: Anomaly detection service

Goal: Build a resilient scalable anomaly detection service.

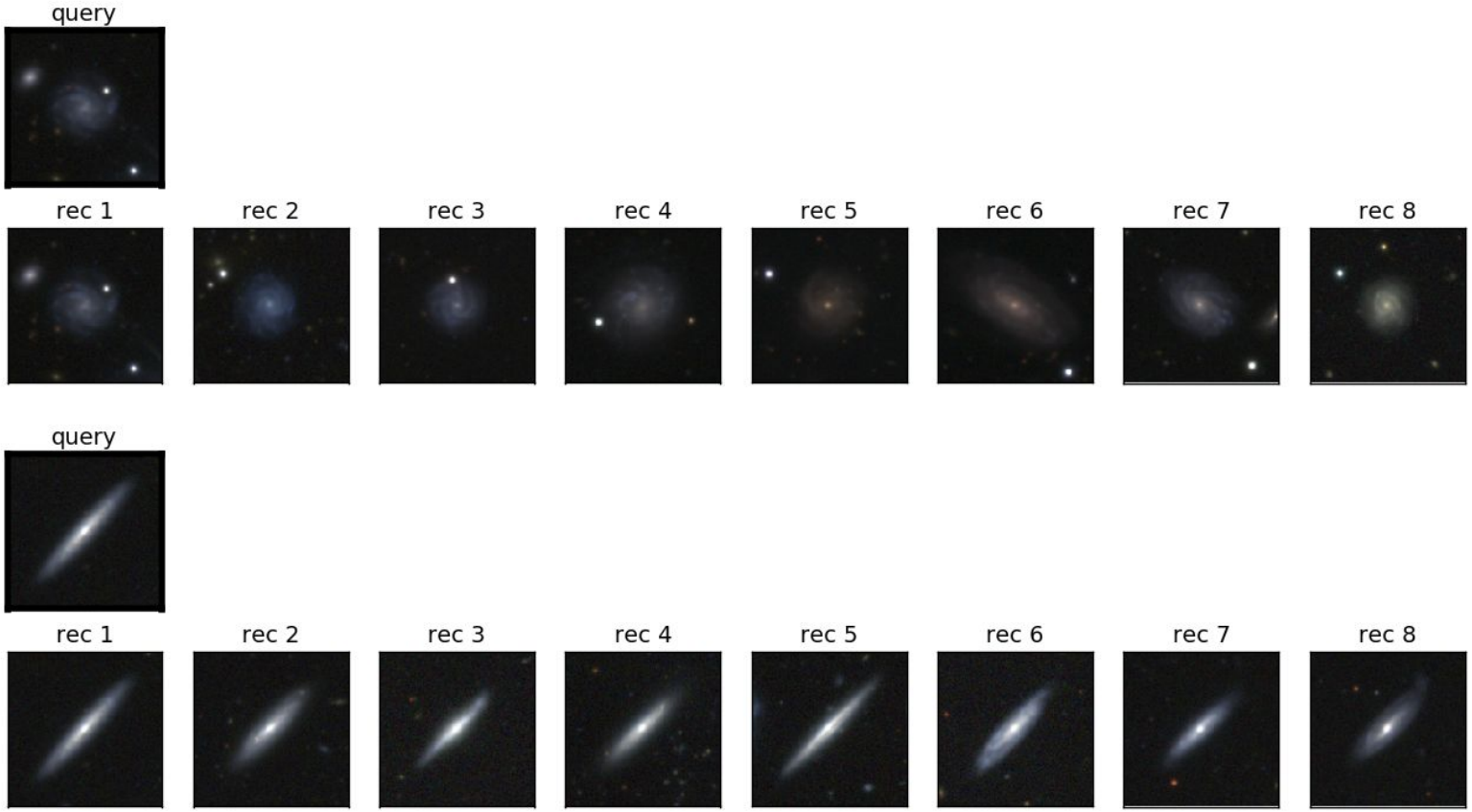
Motivation: Astronomical data (both literal and figurative)

Algorithm: Extended Isolation Forest

Infrastructure: Kubernetes cluster

MapReduce package: Spark

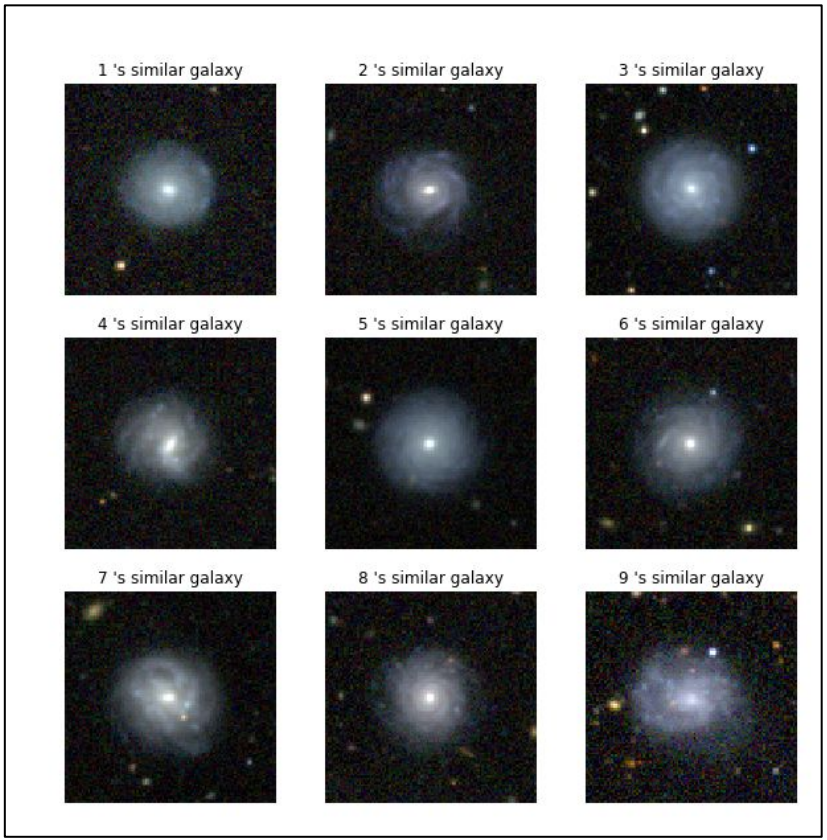
SClaaS Example: Galaxy selection and similarity search

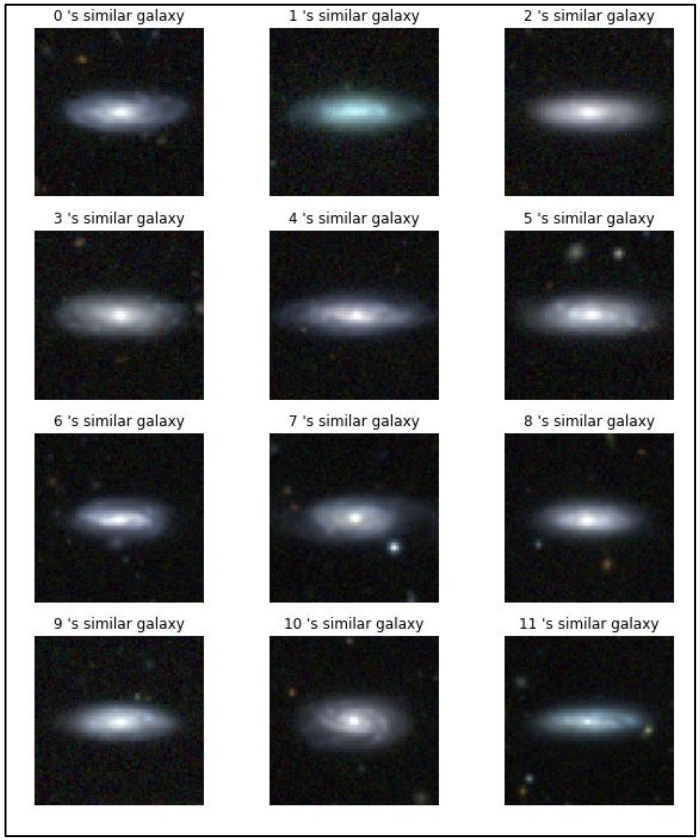


Compress images from 200x200 to 50 or less, for fast search

Bowen, Carrasco-Kind, et al. in prep.

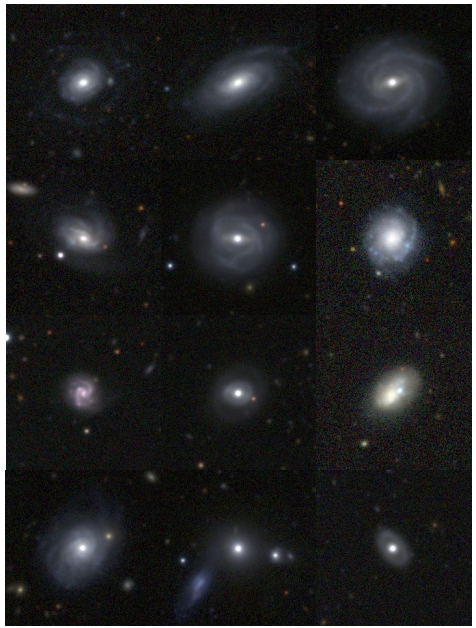
SClaaS Example: Galaxy selection and similarity search





Bowen, Carrasco-Kind, et al. in prep.

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

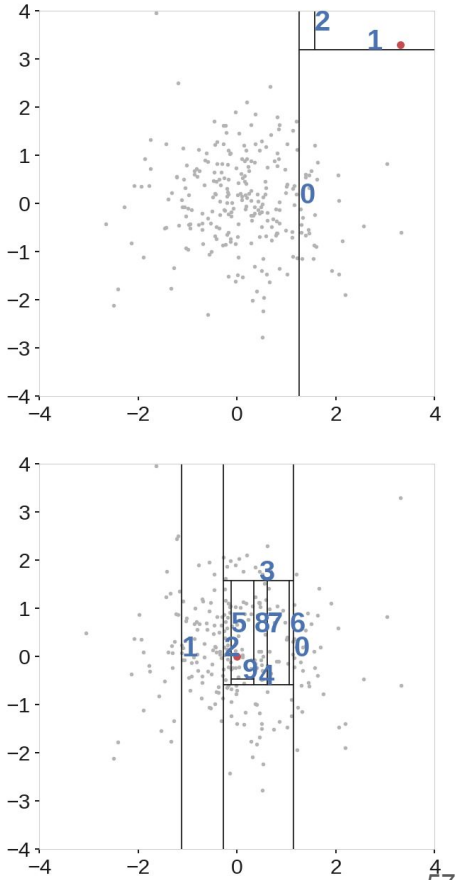
- In operations 2020
- Every night for 10 years
- 18 billions objects (first year), ~40 billions by the end of survey
- ~1500 images per night
- Stream and static data
- Target to capture new physics (moving and variable objects)

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

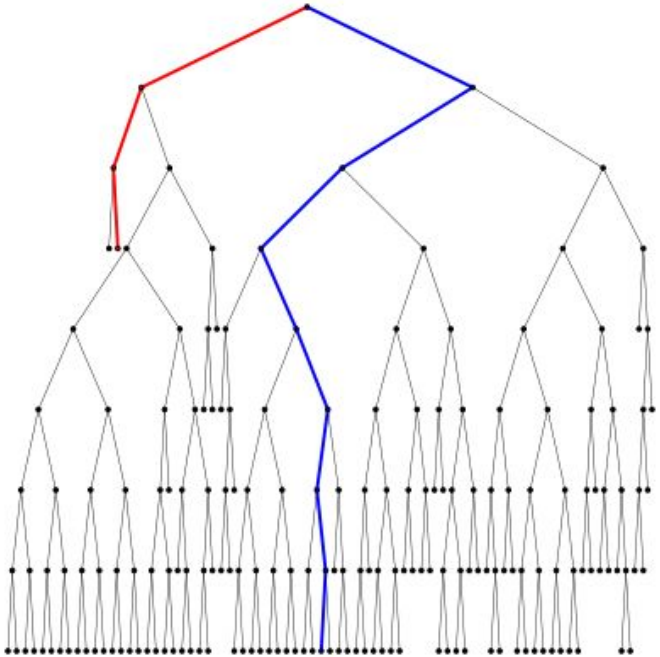
Anomaly Detection with Isolation Forest

- 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 ⟶ 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

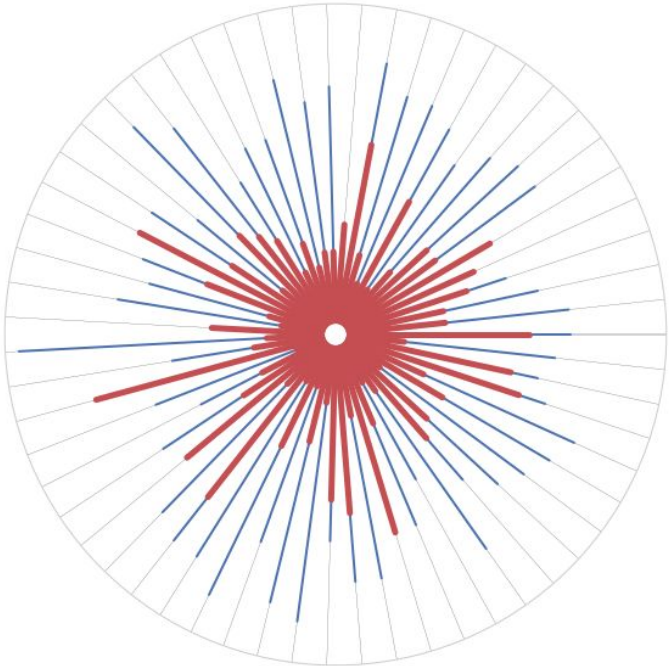


Anomaly Detection with Isolation Forest

Single Tree scores for
anomaly and nominal points



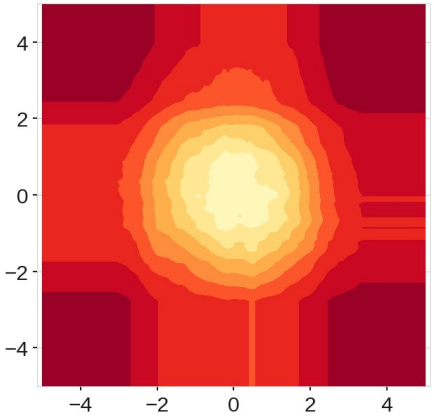
Forest plotted radially.
Scores for anomaly and
nominal shown as lines

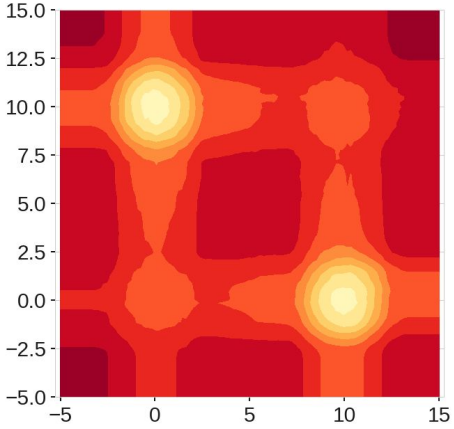


Anomaly Detection with Extended Isolation Forest

Isolation Forest:

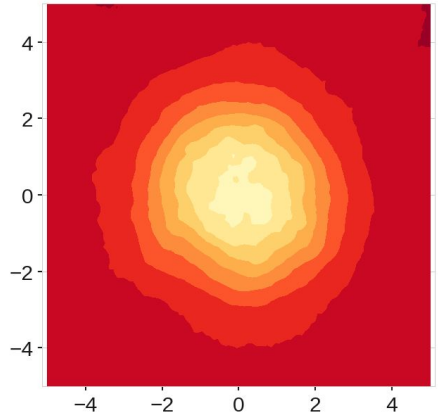
- ✓ Model free
- ✓ Computationally efficient
- ✓ Readily applicable to parallelization
- ✓ Readily application to high dimensional data
- ✗ Inconsistent scoring seen in score maps

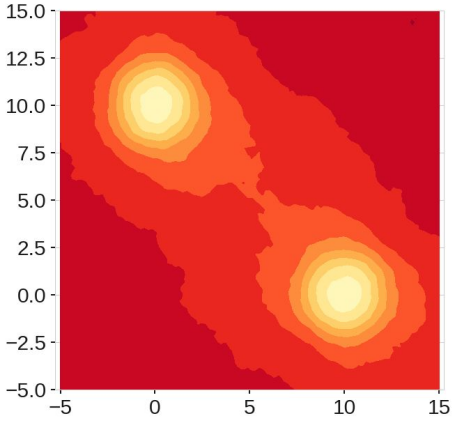




Extended Isolation Forest:

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





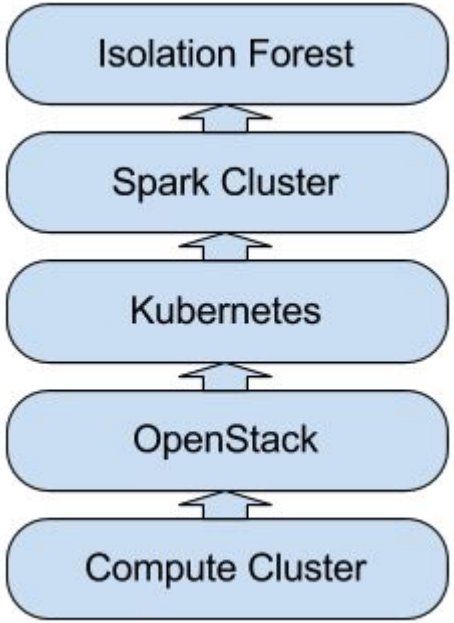
Technology Stack For Anomaly Service

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

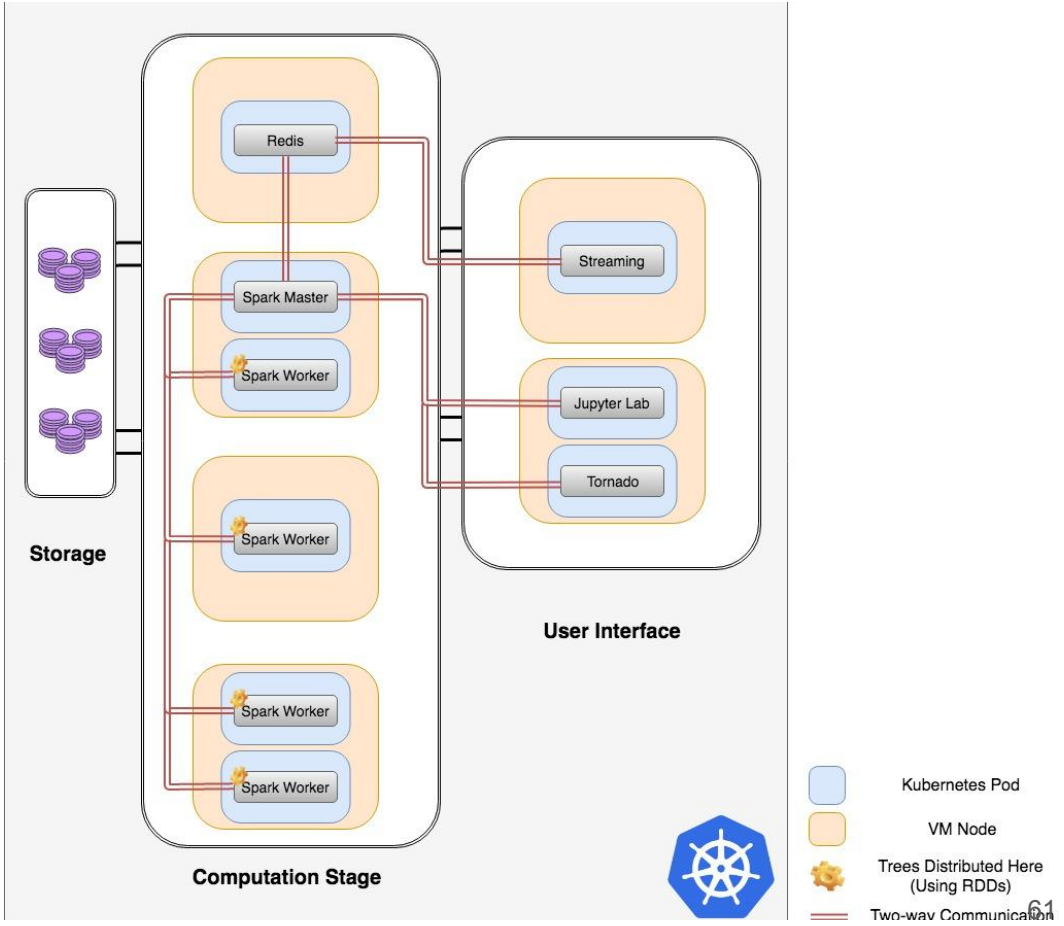
- Use Extended Isolation Forest as core algorithm
- Use Spark to parallelize trees and scoring
- Use Redis as a broker communicator
- To easily deploy in any environment, use Docker
- For orchestration of Docker containers, use Kubernetes
- Kubernetes cluster built on top of OpenStack, but it can be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

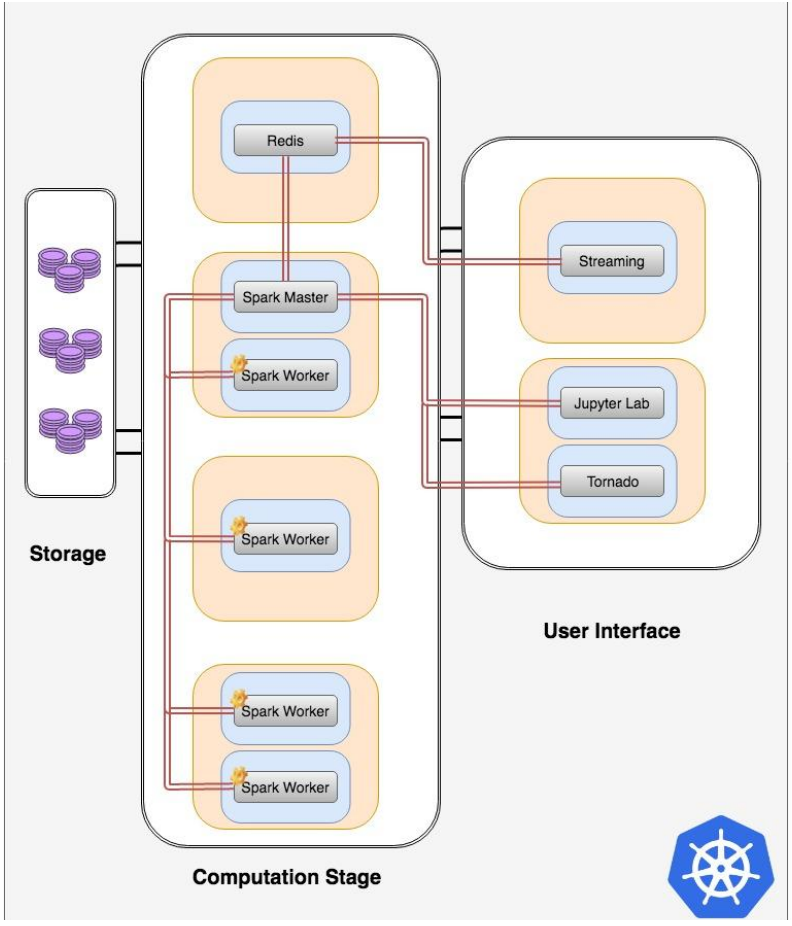
- NFS (Kubernetes PV/PVC)
- Redis
- RDD for Trees and Spark

User Interface:

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

Computation Stage:

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

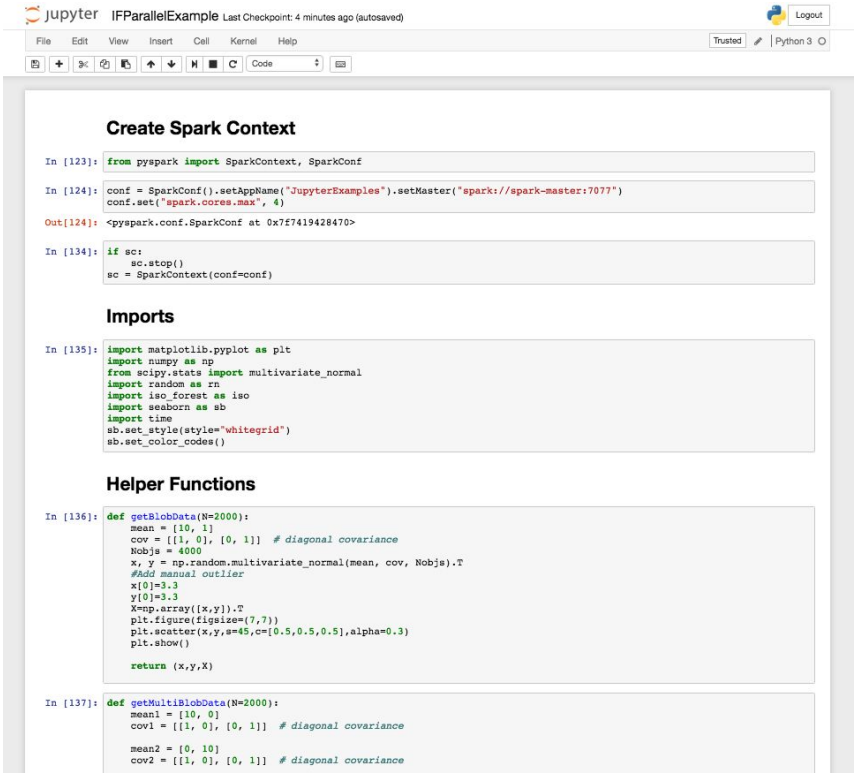


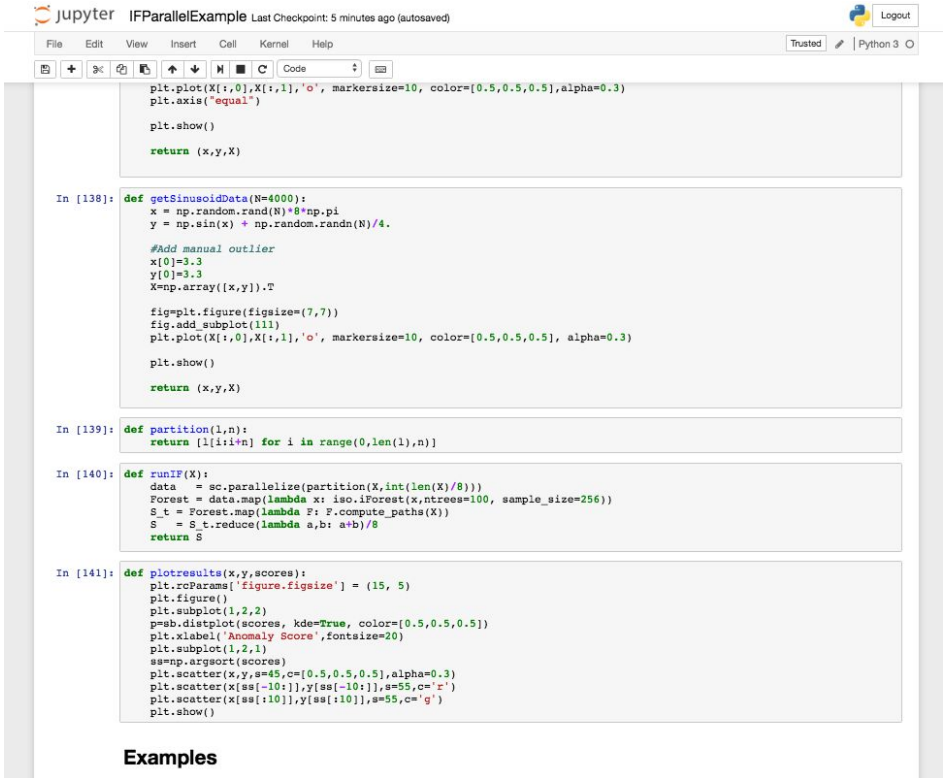
Matias Carrasco Kind, /

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





Example: Jupyter Notebooks

jupyter IFParallelExample Last Checkpoint: 4 minutes ago (autosaved)

Logout

File Edit View Insert Cell Kernel Help

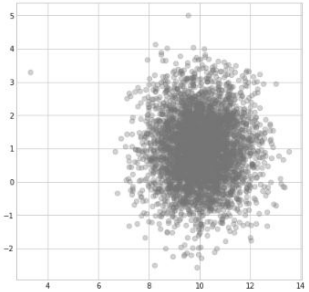
Trusted Python 3

Code

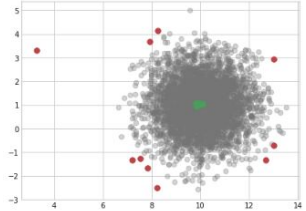
Examples

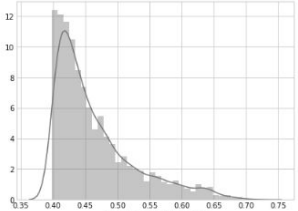
Blob

In [148]:
```
x,y,X = getBlobData()
```



In [149]:
```
S = runIF(X)
plotresults(x,y,S)
```





jupyter IFParallelExample Last Checkpoint: 4 minutes ago (autosaved)

Logout

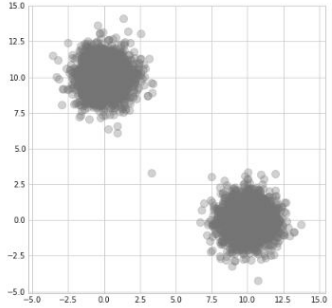
File Edit View Insert Cell Kernel Help

Trusted Python 3

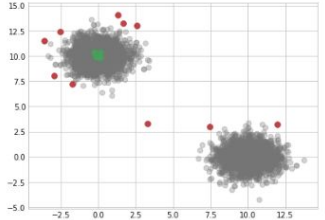
Code

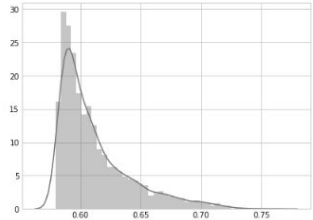
MultiBlob

In [150]:
```
x,y,X = getMultiBlobData()
```



In [151]:
```
S = runIF(X)
plotresults(x,y,S)
```





Final Remarks

- It's all about the user
- Jupyter as Scientific tool but not only solution
- 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
- YOU are not alone

... this is changing the way we do astronomy

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

Thank you!

Questions?

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