RubiOne

APACHE SPARK AS THE BACKBONE OF A RETAIL ANALYTICS DEVELOPMENT ENVIRONMENT

rubikloud

Chief Engineer – Adrian Petrescu

RUBIKLOUD

CORPORATE FOUNDATIONS

Rubikloud is a venture-backed software and technology focused startup. Rubikloud gives retailers the power to lead their markets.

Rubikloud's investors include:

- Horizons Ventures
- Access Industries
- MaRS Investment Accelerator Fund
- TOM Group Limited

Suite 1100, 15 Toronto St,
Toronto, Canada, M5C 2E3

LAST 2 YEARS

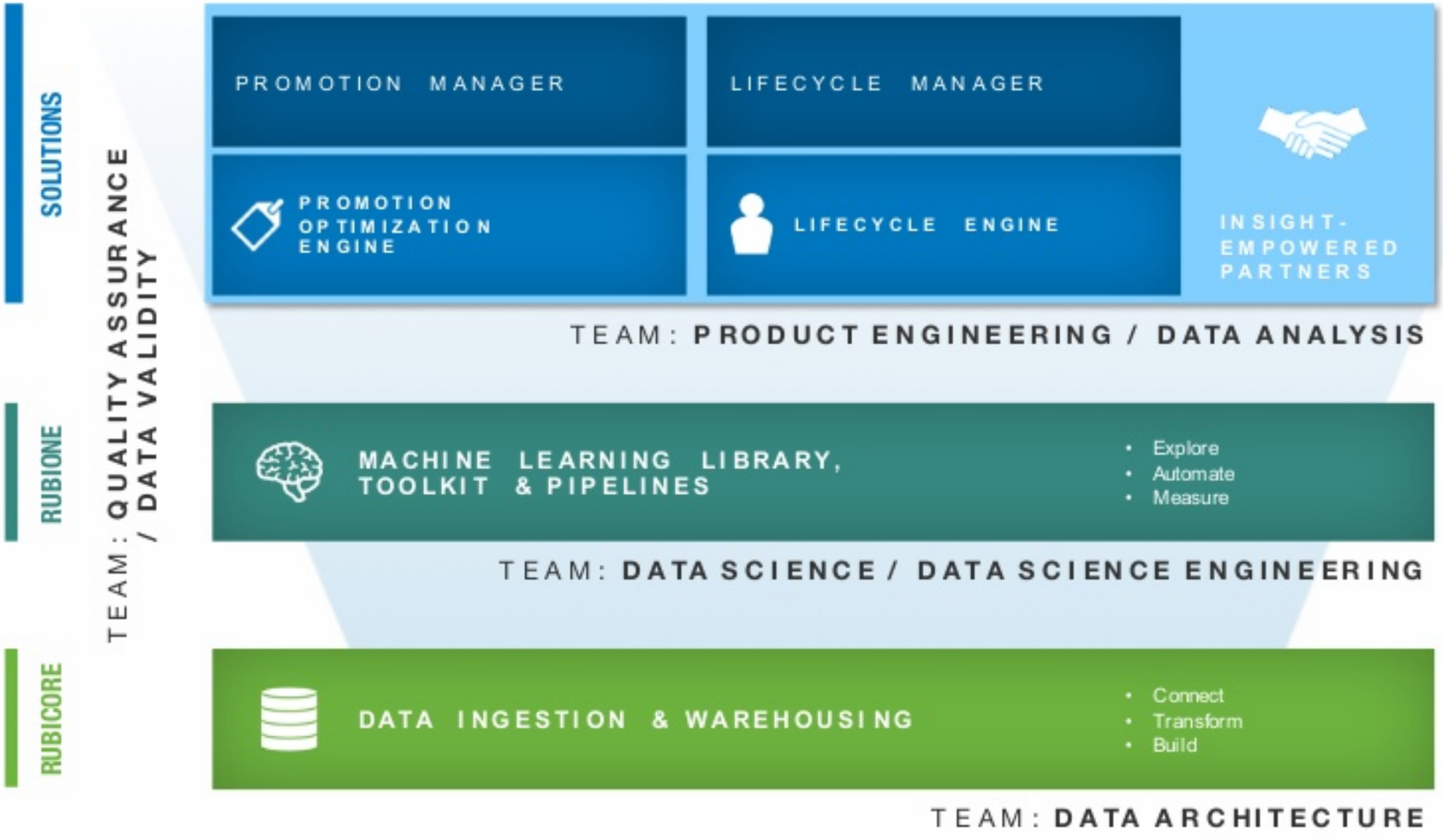
15-50 Teammates

GROWING TO

100+ (2017)

FOUNDED EARLY-2013

The Rubikloud Platform



Data Science Development Platform

RubiOne

MODELS

PRICE ELASTICITY

17

FORECASTING MODELS

PRICE & PROMOTION OPTIMIZATION

CUSTOMER VALUE METRICS

CAPABILITIES

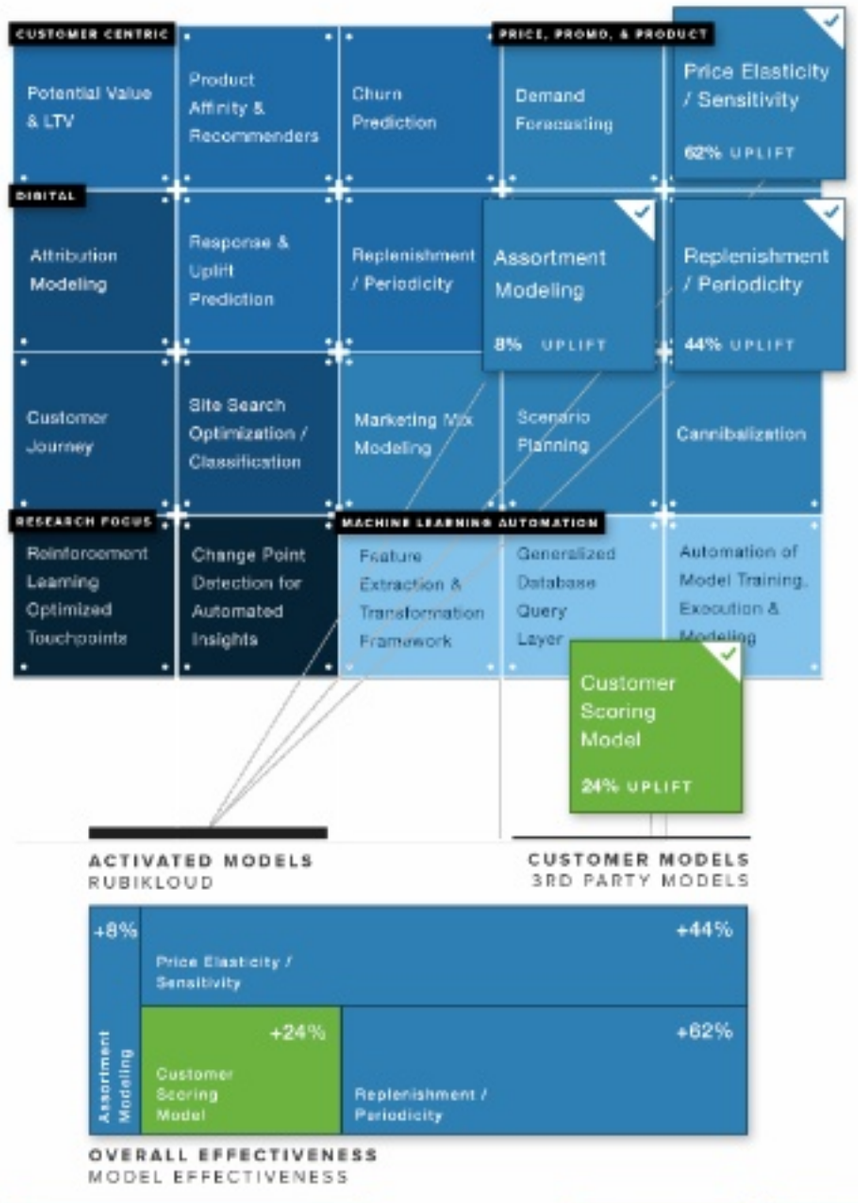
Develop your own models, at scale

Leverage our vertical specific feature-engineering framework

Chain together models in pipelines to deploy multi-stage systems into production

Incorporate or benchmark against Rubikloud's own models, trained on several years of retail data

Leverage various visualization libraries to compare and monitor the performance of models



RubiOne

Goals

- Completely self-contained deployment, to work around long IT turnaround times

RubiOne

Goals

- Completely self-contained deployment, to work around long IT turnaround times
- Integrations with common enterprise retail authentication systems, data backends, and execution layers

RubiOne

Goals

- Completely self-contained deployment, to work around long IT turnaround times
- Integrations with common enterprise retail authentication systems, data backends, and execution layers
- Integrate with familiar tools, while providing transparent scalability

RubiOne

Goals

- Completely self-contained deployment, to work around long IT turnaround times
- Integrations with common enterprise retail authentication systems, data backends, and execution layers
- Integrate with familiar tools, while providing transparent scalability
- Multi-cloud/cloud-agnostic

RubiOne

Goals

- Completely self-contained deployment, to work around long IT turnaround times
- Integrations with common enterprise retail authentication systems, data backends, and execution layers
- Integrate with familiar tools, while providing transparent scalability
- Multi-cloud/cloud-agnostic
- Have the ability to apply policy to *everything*

RubiOne

Goals

- Completely self-contained deployment, to work around long IT turnaround times
- Integrations with common enterprise retail authentication systems, data backends, and execution layers
- Integrate with familiar tools, while providing transparent scalability
- Multi-cloud/cloud-agnostic
- Have the ability to apply policy to *everything*
- Batteries included

RubiOne

NON-Goals

- Deep learning

RubiOne

NON-Goals

RubiOne

NON-Goals

- Deep learning
- Low-latency results

RubiOne

NON-Goals

- Deep learning
- Low-latency results
- Streaming input

RubiOne

NON-Goals

- Deep learning
- Low-latency results
- Streaming input
- “Big” data


rubikloud™


rubikloud

RubiOne

RubiOne

What It Is

- Interactive, ad-hoc development
- Backed by JupyterHub and Jupyter Notebooks
- Integrated with Rubikloud's retail vertical-specific visualizations, ORM, and machine learning library.

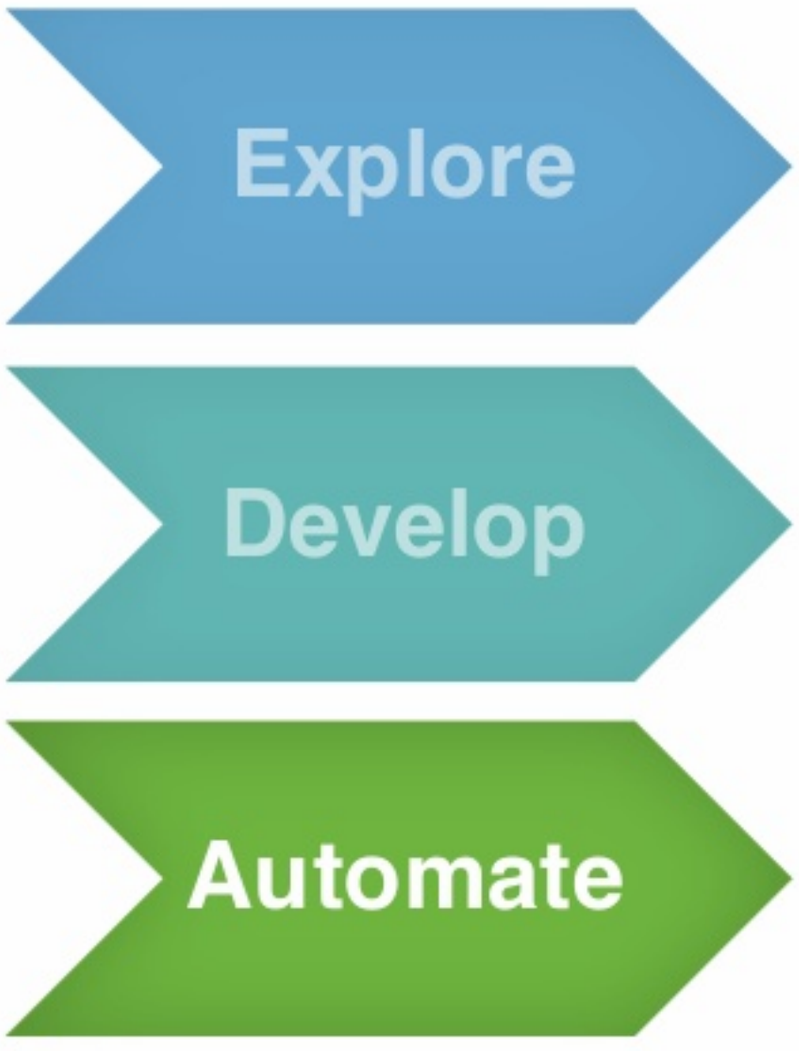
RubiOne

What It Is

- Powerful APIs for production-grade machine learning in retail
- Built-in feature extraction, dependency management, and parameter tuning

RubiOne

What It Is



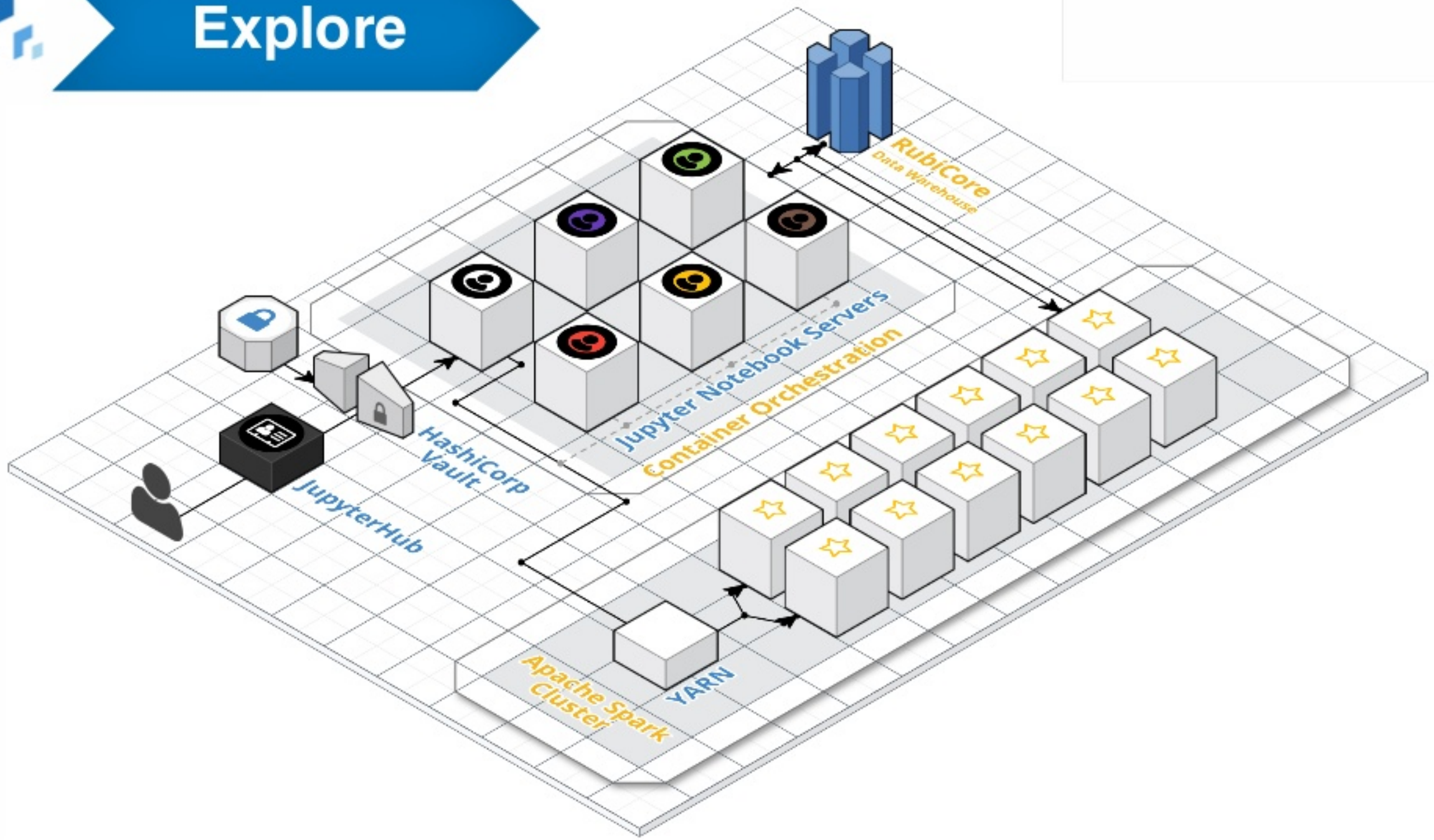
- Plugs in directly to the computational graph for our existing models
- Schedules batch jobs according to data freshness/staleness
- Output directly into execution tools

RubiOne

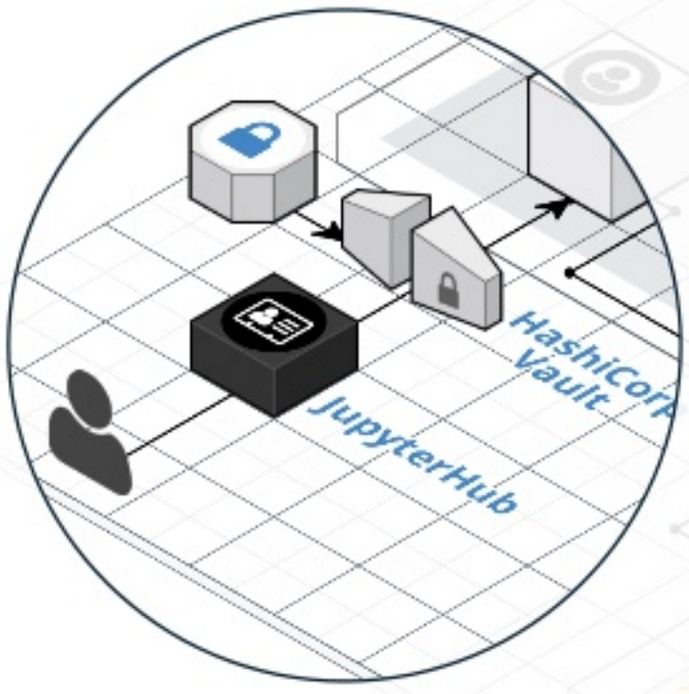
What It Is

Apache Spark is the backbone that makes all of this possible.

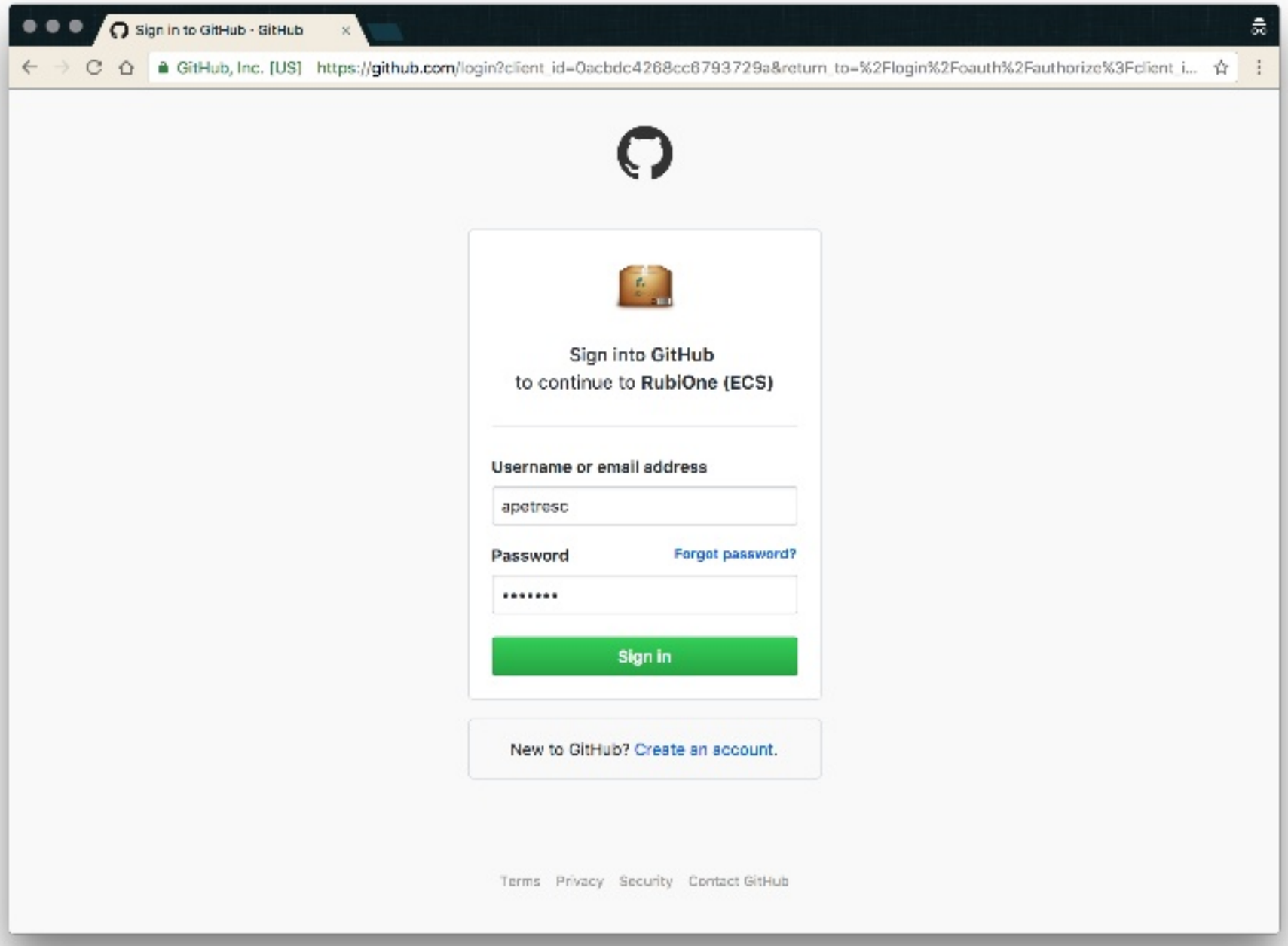
Explore

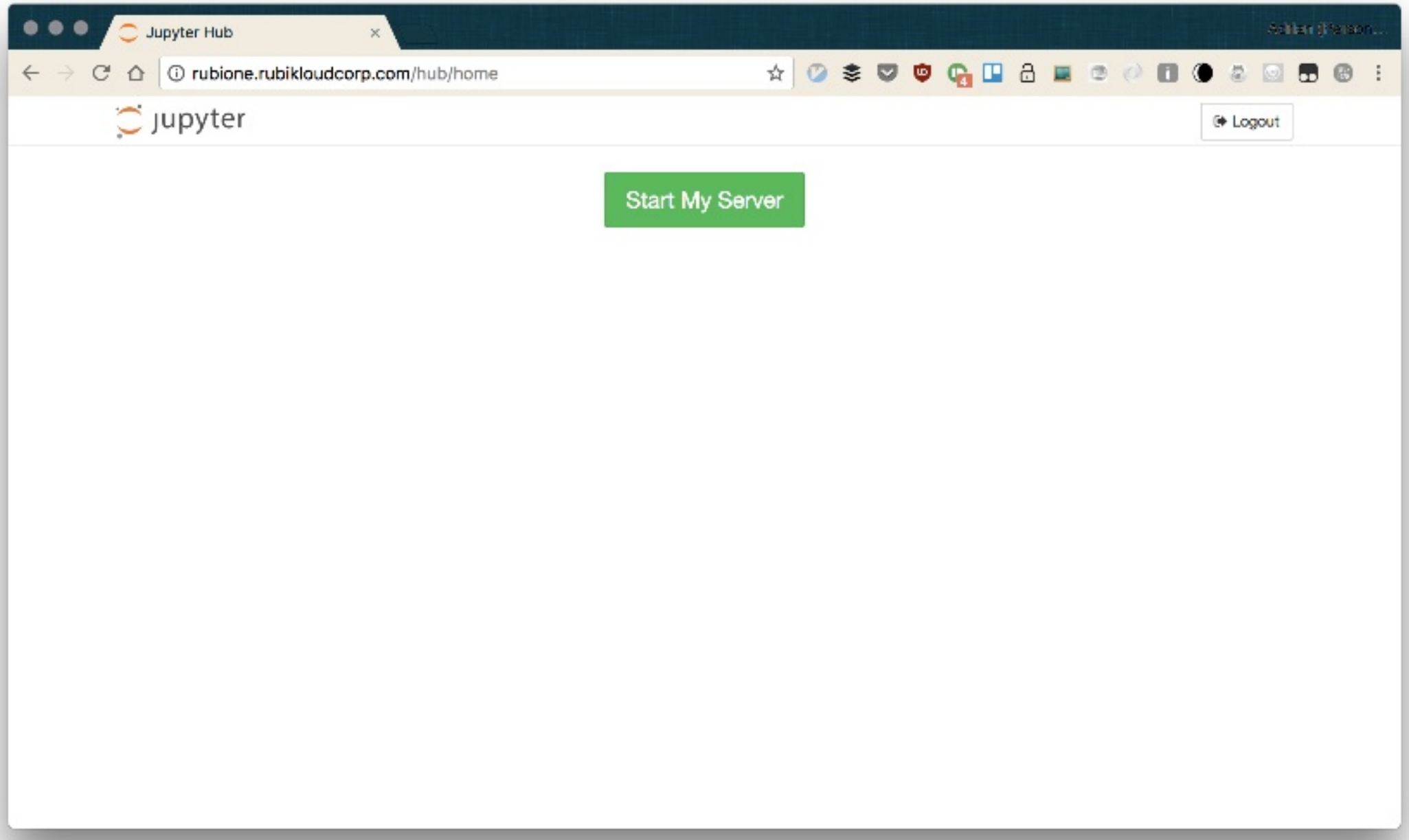

RubiCore
Data Warehouse
Jupyter Notebook Servers
Container Orchestration
HashiCorp
Vault
JupyterHub
YARN
Apache Spark
Cluster

Explore



- JupyterHub is a multi-user hub which spawns and proxies per-user Jupyter Notebook instances
- Integrates with LDAP, OAuth, SAML, etc through its Authenticator interface
- Integrates with multiple orchestration systems through its Spawner interface
- Responsible for provisioning individual workers with configuration and authentication tokens


Sign in to GitHub · GitHub
GitHub, Inc. [US] https://github.com/login?client_id=0acbdc4268cc6793729a&return_to=%2Flogin%2Foauth%2Fauthorize%3Fclient_i...
Sign into GitHub
to continue to RubiOne (ECS)
Username or email address
apetresc
Password
Forgot password?
Sign In
New to GitHub? Create an account.
Terms Privacy Security Contact GitHub

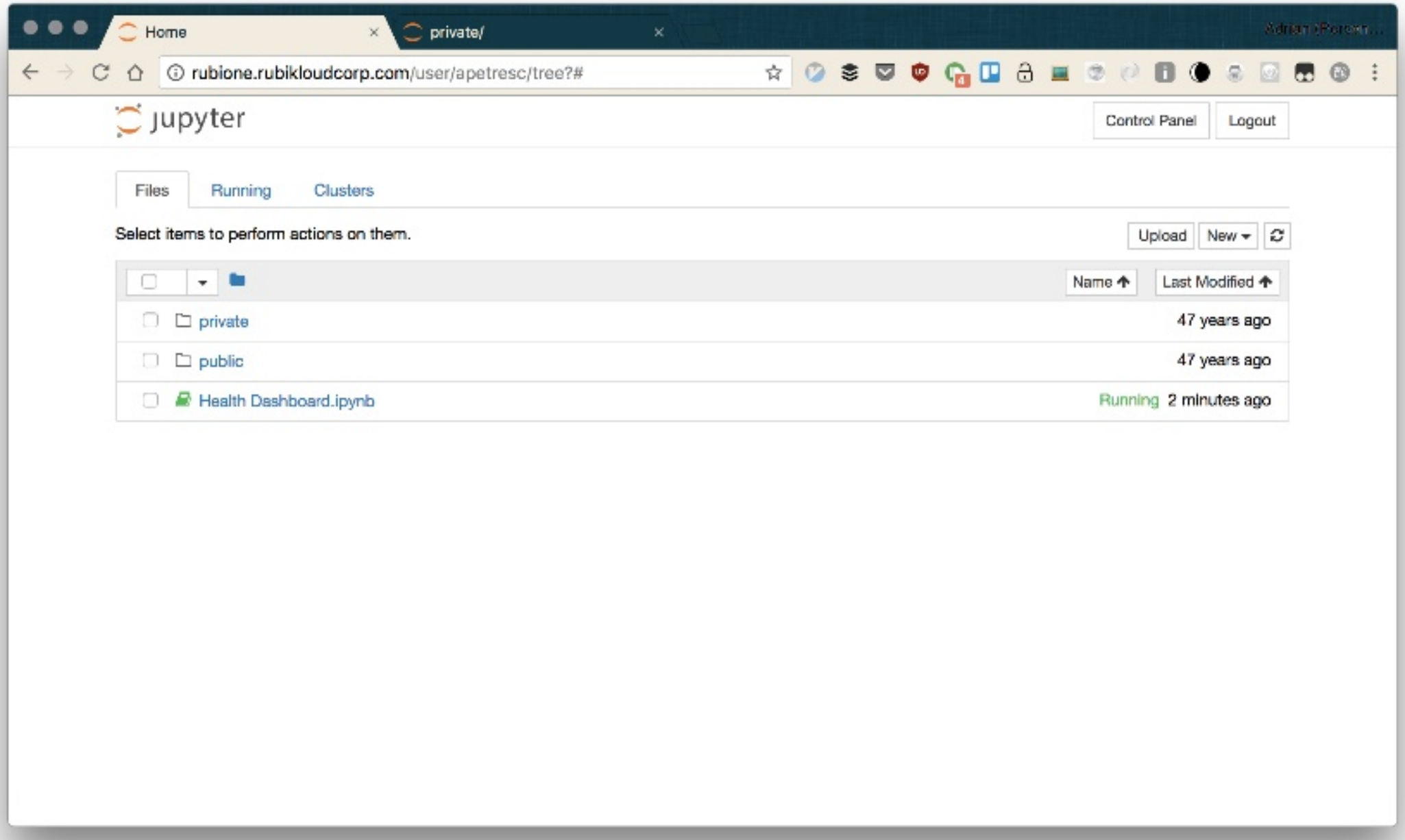

Jupyter Hub
rubione.rubikloudcorp.com/hub/home
jupyter
Logout
Start My Server

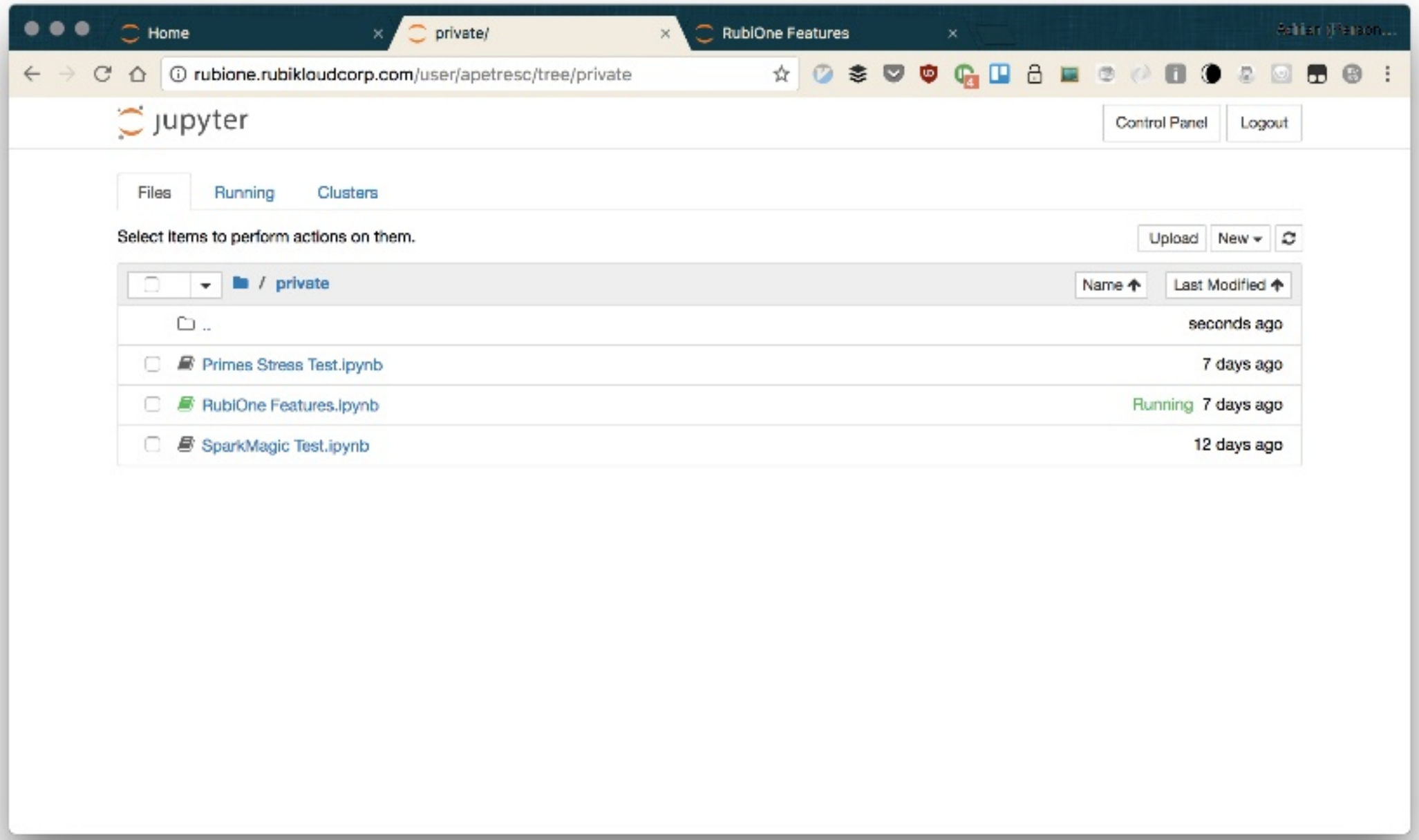
Explore

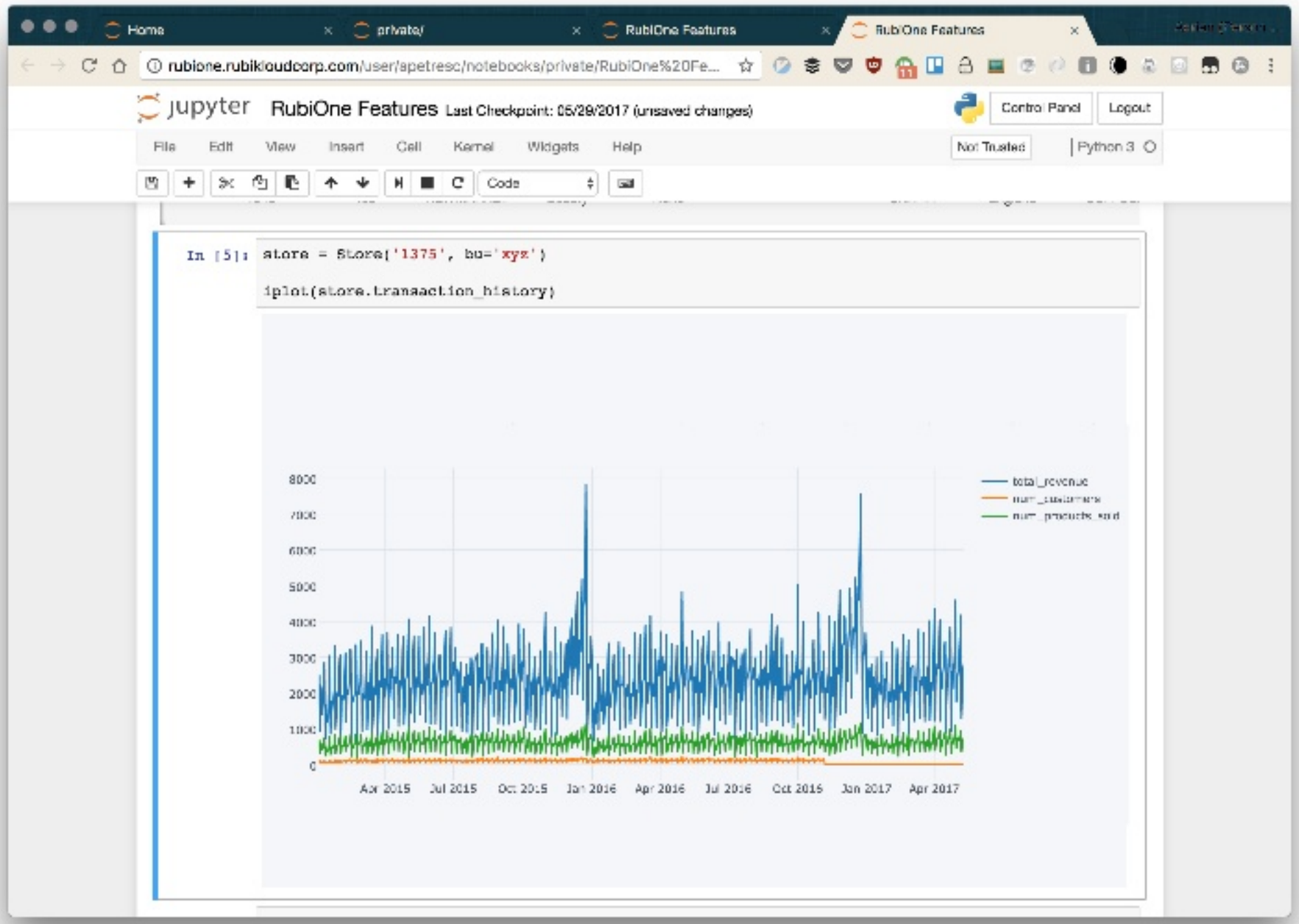
- Hosts Rubikloud's retail-specific data schema, RDM
- Also contains extension tables, model outputs, and measurement results specific to each client
- Row- and column-level policy enforced at the role level
- Convenient ORM takes advantage of the fact that schema is consistent across BUs and clients

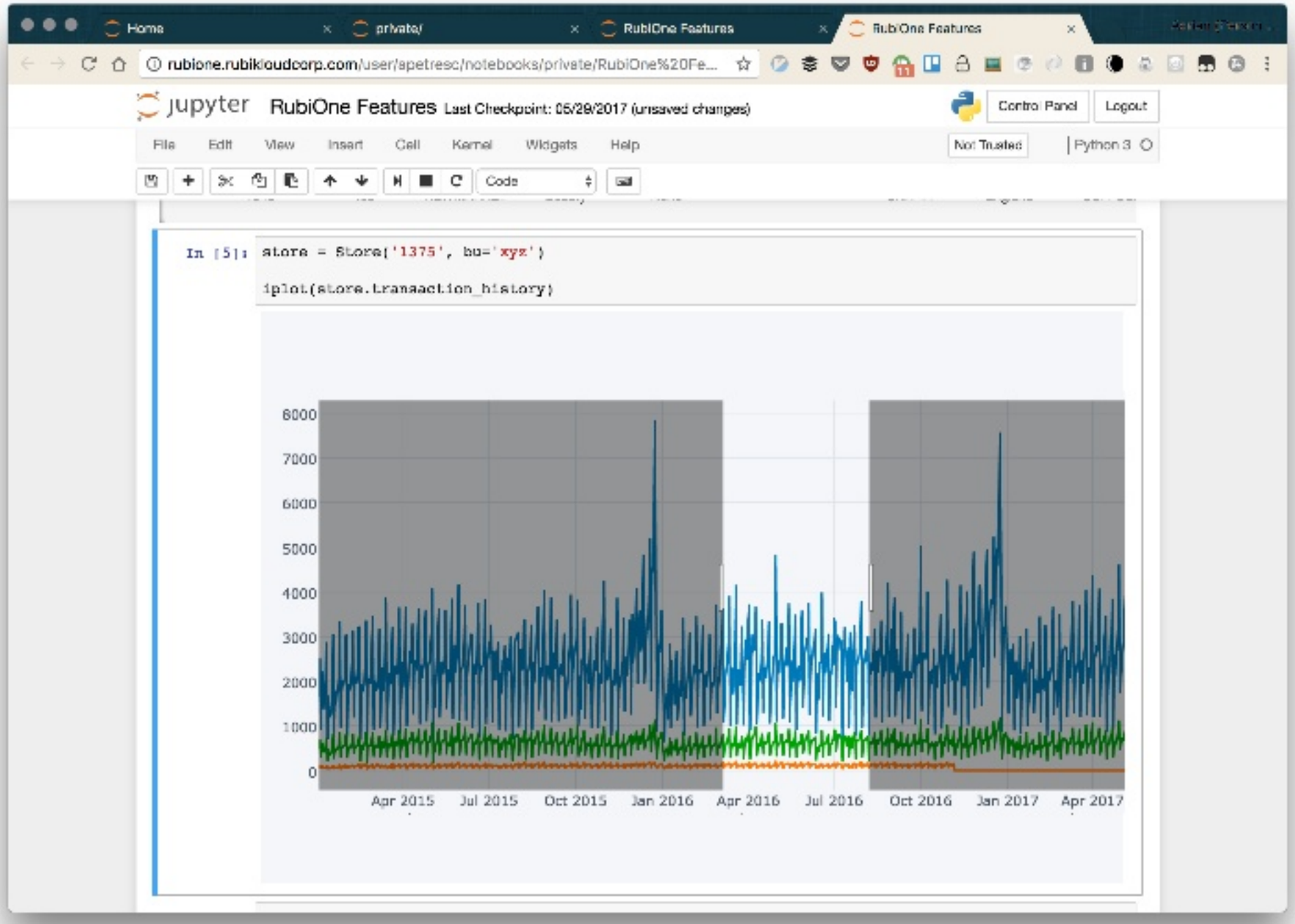
Explore

- Spawned as on-demand ephemeral containers behind a VPC – not accessible on the public internet
- Preloaded with all the tools of the trade:
  - Kernels for Python, R, and Scala
  - Stable version set of common libraries like NumPy, scikit-learn
- Mounted volumes to instantly sync notebooks


Home
private/
rubione.rubikloudcorp.com/user/apetresc/tree?#
jupyter
Control Panel
Logout
Files
Running
Clusters
Select items to perform actions on them.
Upload
New
Name
Last Modified
private
47 years ago
public
47 years ago
Health Dashboard.ipynb
Running 2 minutes ago


Home
private/
RubiOne Features
rubione.rubikloudcorp.com/user/apetresc/tree/private
jupyter
Control Panel
Logout
Files
Running
Clusters
Select items to perform actions on them.
Upload
New
/ private
Name
Last Modified
..
seconds ago
Primes Stress Test.ipynb
7 days ago
RubiOne Features.ipynb
Running 7 days ago
SparkMagic Test.ipynb
12 days ago


Home
private/
RubiOne Features
RubiOne Features
rubione.rubikloudcorp.com/user/apetresc/notebooks/private/RubiOne%20Fe...
jupyter
RubiOne Features
Last Checkpoint: 05/29/2017 (unsaved changes)
Control Panel
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Not Trusted
Python 3
Code
In [5]:
store = Store('1375', bu='xyz')
iplot(store.transaction_history)
8000
7000
6000
5000
4000
3000
2000
1000
0
Apr 2015
Jul 2015
Oct 2015
Jan 2016
Apr 2016
Jul 2016
Oct 2016
Jan 2017
Apr 2017
total_revenue


Home
private/
RubiOne Features
RubiOne Features
rubione.rubikloudcorp.com/user/apetresc/notebooks/private/RubiOne%20Fe...
jupyter
RubiOne Features
Last Checkpoint: 05/29/2017 (unsaved changes)
Control Panel
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Not Trusted
Python 3
Code
In [5]:
store = Store('1375', bu='xyz')
iplot(store.transaction_history)
8000
7000
6000
5000
4000
3000
2000
1000
0
Apr 2015
Jul 2015
Oct 2015
Jan 2016
Apr 2016
Jul 2016
Oct 2016
Jan 2017
Apr 2017

LaTeX

Execute timings

R, Scala, Python

File I/O Abstraction

Geocoders and GIS imagery

Credential management

Version control

Spark wrapper kernels

sparkmagic

Snippets

Plot.ly

Excel/CSV import

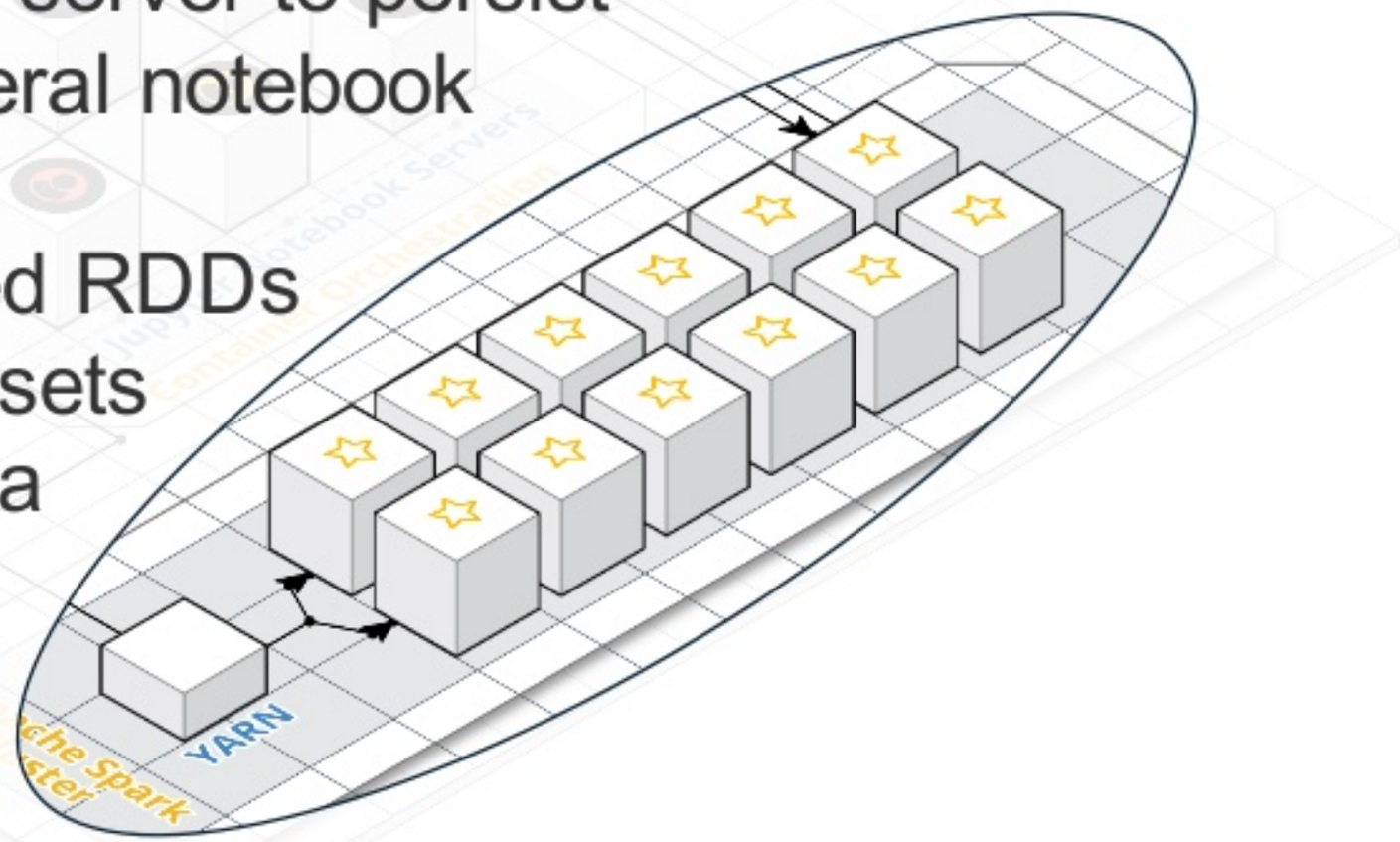
nbconvert

A/B Testing

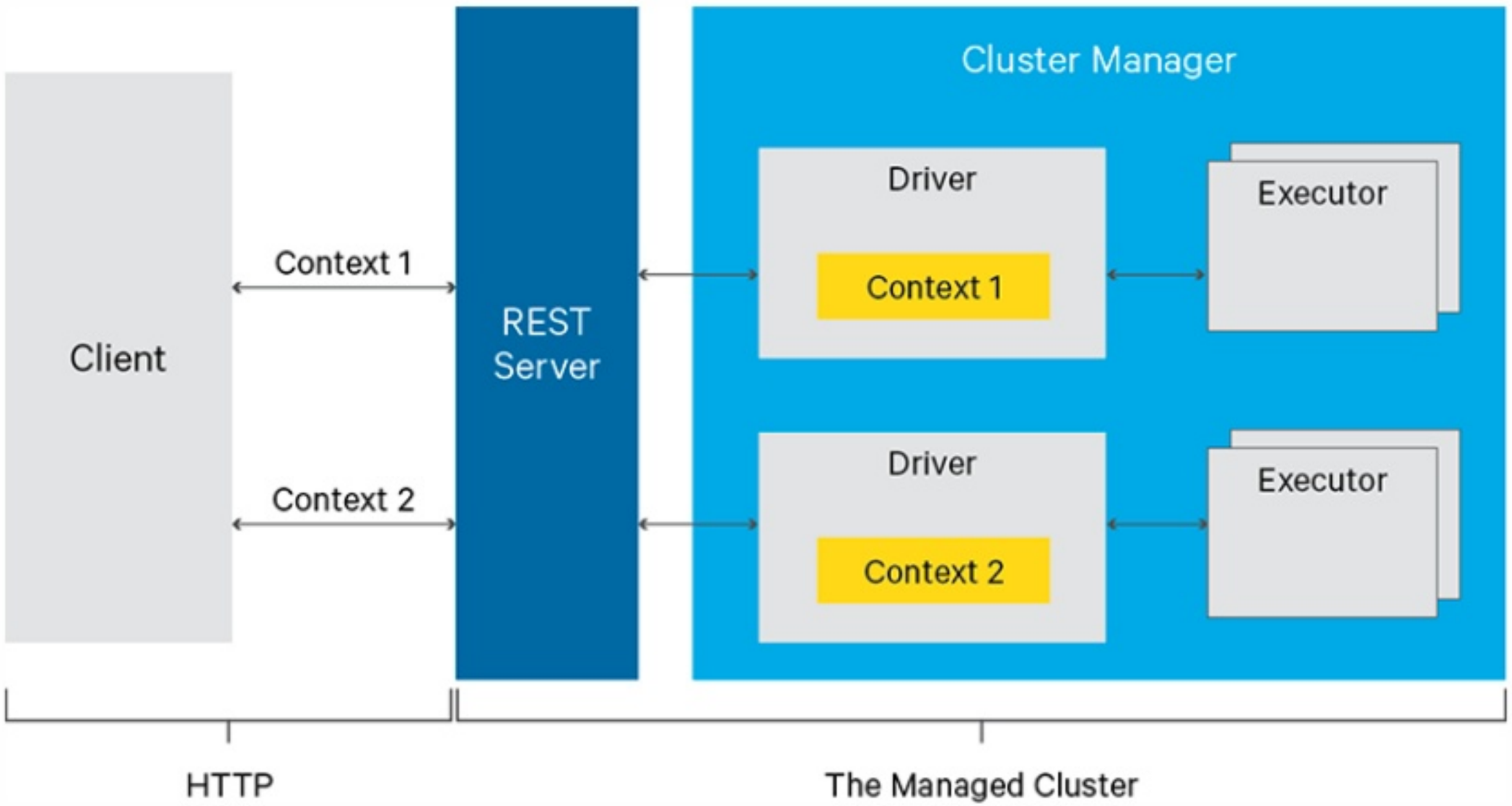
Keyboard shortcuts

Explore

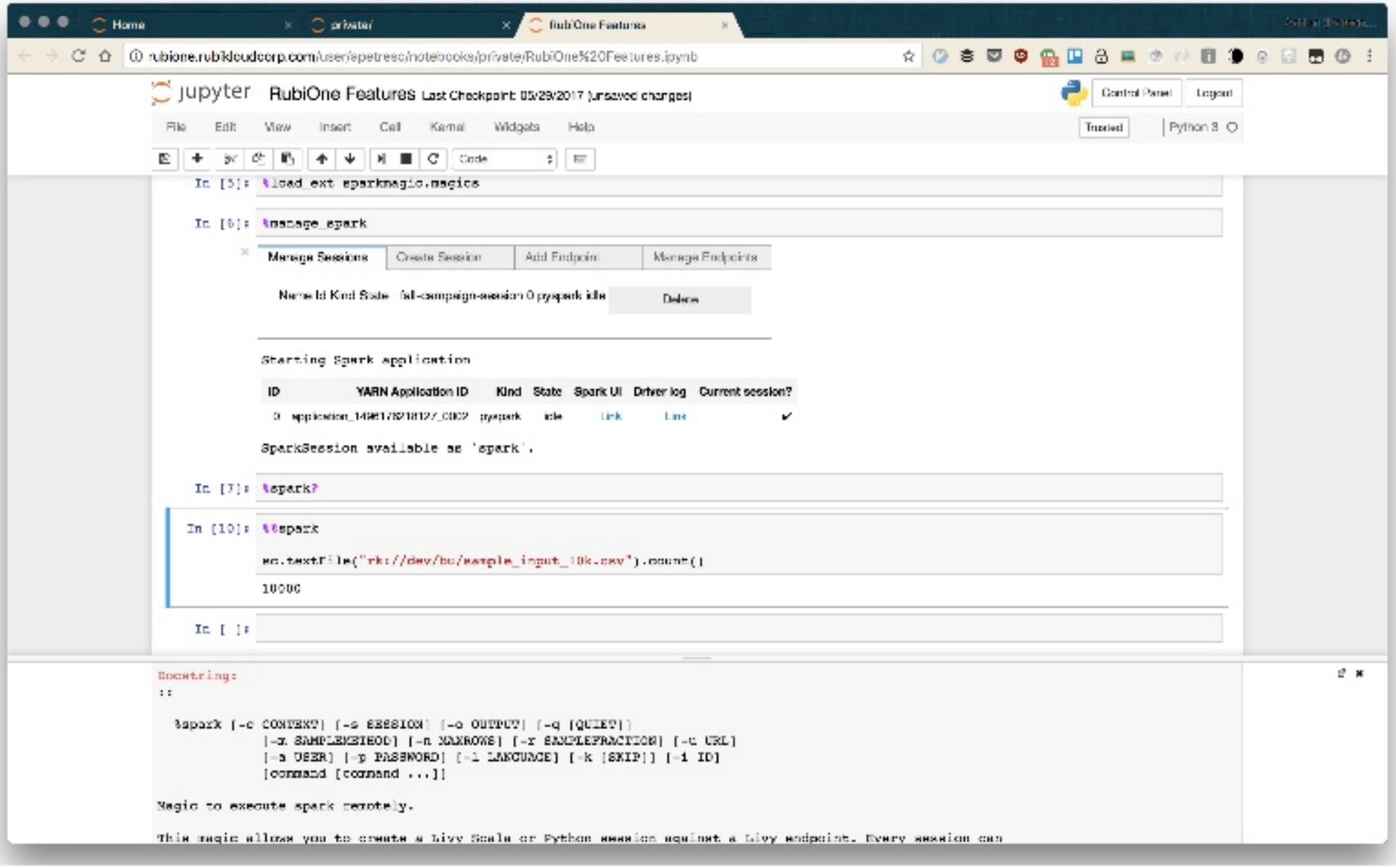
- Auto-scales based on current usage
- Master node runs a Livy server to persist sessions across ephemeral notebook servers
- Maintains a set of cached RDDs for commonly-used datasets
- Federated access to data warehouse and blob storage



Explore


Cluster Manager
Driver
Context 1
Executor
Context 1
REST
Server
Client
Driver
Context 2
Executor
Context 2
HTTP
The Managed Cluster

Explore


Home
private
RubiOne Features
rubione.rubikcloudcorp.com/user/epetrescl/notebooks/private/RubiOne%20Features.ipynb
jupyter RubiOne Features Last Checkpoint: 05/29/2017 (unsaved changes)
Control Panel
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
In [5]: %load_ext sparkmagic.magics
In [6]: %manage_spark
Manage Sessions
Create Session
Add Endpoint
Manage Endpoints
Name Id Kind State fall-campaign-session 0 pyspark idle
Delete
Starting Spark application
ID YARN Application ID Kind State Spark UI Driver log Current session?
0 application_1496178218127_0002 pyspark idle link link ✔
SparkSession available as 'spark'.
In [7]: %spark?
In [10]: %%spark
sc.textFile("rk://dev/bc/sample_input_10k.csv").count()
10000
In [ ]:
Docstring:
::
%spark [-c CONTEXT] [-s SESSION] [-o OUTPUT] [-q [QUIET]]
[-m SAMPLEMETHOD] [-n MAXROWS] [-r SAMPLEFRACTION] [-u URL]
[-a USER] [-p PASSWORD] [-l LANGUAGE] [-k [SKIP]] [-i ID]
[command [command ...]]
Magic to execute spark remotely.
This magic allows you to create a Livy Scala or Python session against a Livy endpoint. Every session can

Explore

Home | private/ | RubiOne Features | catherinedevlin/ipython-sql:

rubione.rubikloudcorp.com/user/apetresc/notebooks/private/RubiOne%...

jupyter **RubiOne Features** Last Checkpoint: 05/29/2017 (unsaved changes)

Control Panel | Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3

Code

ID	YARN Application ID	Kind	State	Spark UI	Driver log	Current session?
0	application_1496176218127_0002	pyspark	idle	Link	Link	✔

```
SparkSession available as 'spark'.
```

In [7]:

```
%%sql -n campaign_history

select
      cc.BU_CODE,
      x.campaign_name,
      x.campaign_code,
      x.TREATMENT_CHANNEL_NAME,
      final_load,
      count(distinct contact_key) as CONTACTS
from
            (
                  select
                        co.campaign_name,
                        co.campaign_code,
                        cc.camp_offer_key,
                        cc.TREATMENT_CHANNEL_NAME,
                        max(cc.load_num) as final_load
                  from
                        crm_target.b_camp_offer co
                        inner join
                        crm_target.b_camp_contact cc
                        on  co.camp_offer_key = cc.camp_offer_key
                  where
                        launched_ts >= '01-JAN-17'
```

Explore

Home | private/ | RubiOne Features | catherinedevlin/ipython-sql:

rubione.rubikloudcorp.com/user/apetresc/notebooks/private/RubiOne%...

jupyter **RubiOne Features** Last Checkpoint: 05/29/2017 (unsaved changes)

Control Panel | Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3

Code

ID	YARN Application ID	Kind	State	Spark UI	Driver log	Current session?
0	application_1496176218127_0002	pyspark	idle	Link	Link	✔

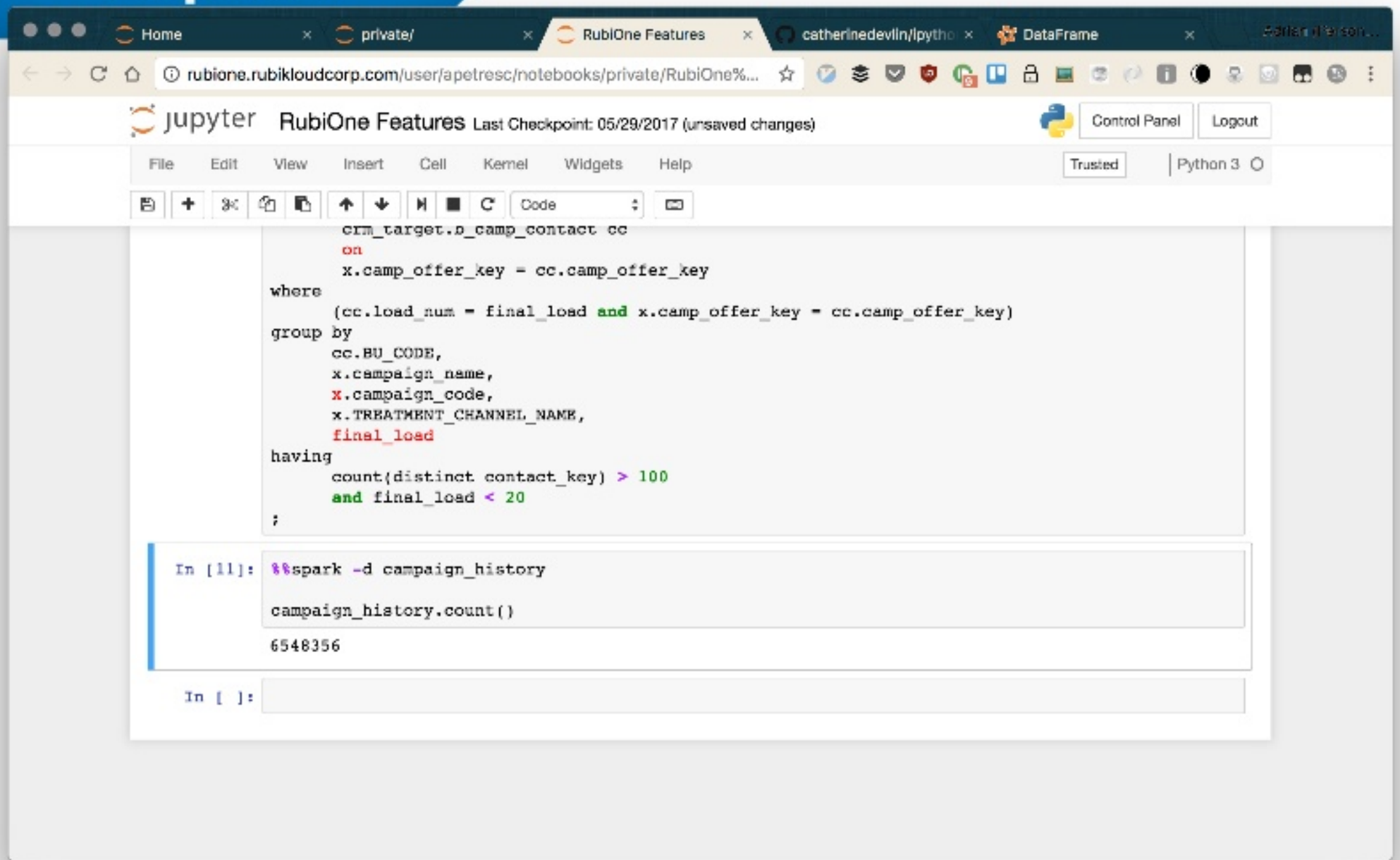
```
SparkSession available as 'spark'.
```

In [7]:

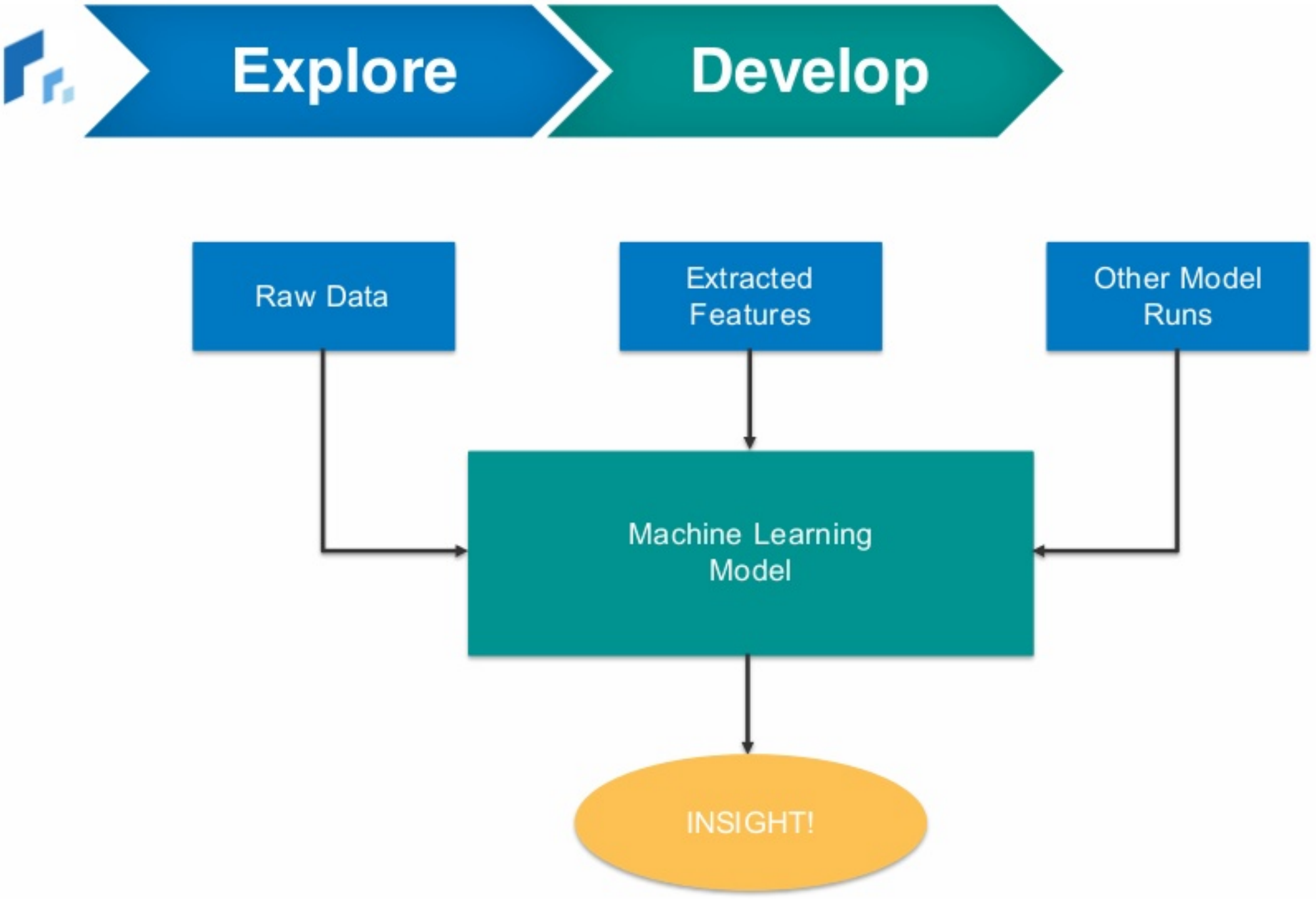
```
%%sql -n campaign_history

select
      cc.BU_CODE,
      x.campaign_name,
      x.campaign_code,
      x.TREATMENT_CHANNEL_NAME,
      final_load,
      count(distinct contact_key) as CONTACTS
from
            (
                  select
                        co.campaign_name,
                        co.campaign_code,
                        cc.camp_offer_key,
                        cc.TREATMENT_CHANNEL_NAME,
                        max(cc.load_num) as final_load
                  from
                        crm_target.b_camp_offer co
                        inner join
                        crm_target.b_camp_contact cc
                        on  co.camp_offer_key = cc.camp_offer_key
                  where
                        launched_ts >= '01-JAN-17'
```

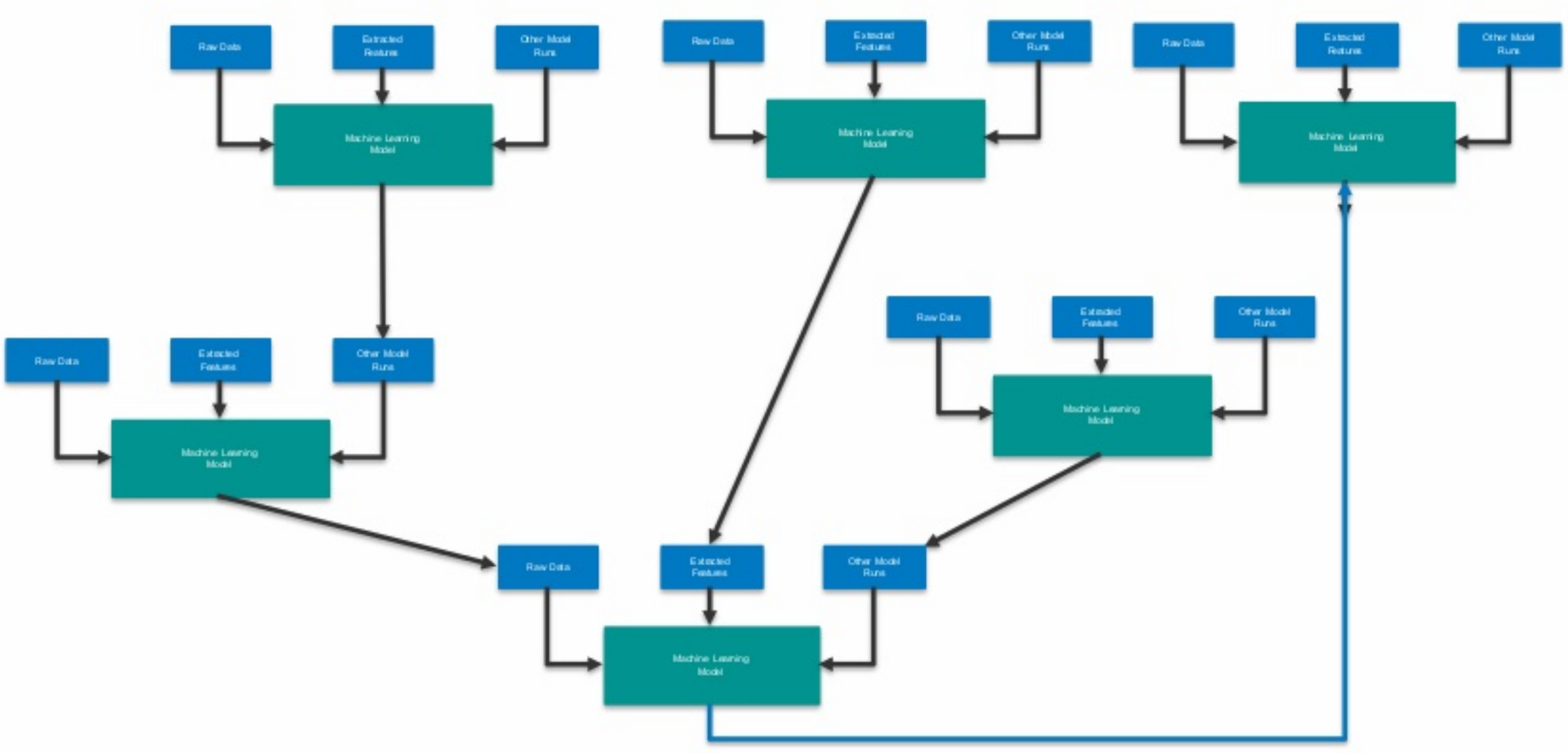
Explore


Home
private/
RubiOne Features
catherinedevlin/ipytho
DataFrame
rubione.rubikloudcorp.com/user/apetresc/notebooks/private/RubiOne%...
jupyter
RubiOne Features
Last Checkpoint: 05/29/2017 (unsaved changes)
Control Panel
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Code
crm_target.b_camp_contact cc
on
x.camp_offer_key = cc.camp_offer_key
where
(cc.load_num = final_load and x.camp_offer_key = cc.camp_offer_key)
group by
cc.BU_CODE,
x.campaign_name,
x.campaign_code,
x.TREATMENT_CHANNEL_NAME,
final_load
having
count(distinct contact_key) > 100
and final_load < 20
;
In [11]:
%%spark -d campaign_history
campaign_history.count()
6548356
In [ ]:


Explore
Develop


Explore
Develop
Raw Data
Extracted Features
Other Model Runs
Machine Learning Model
INSIGHT!

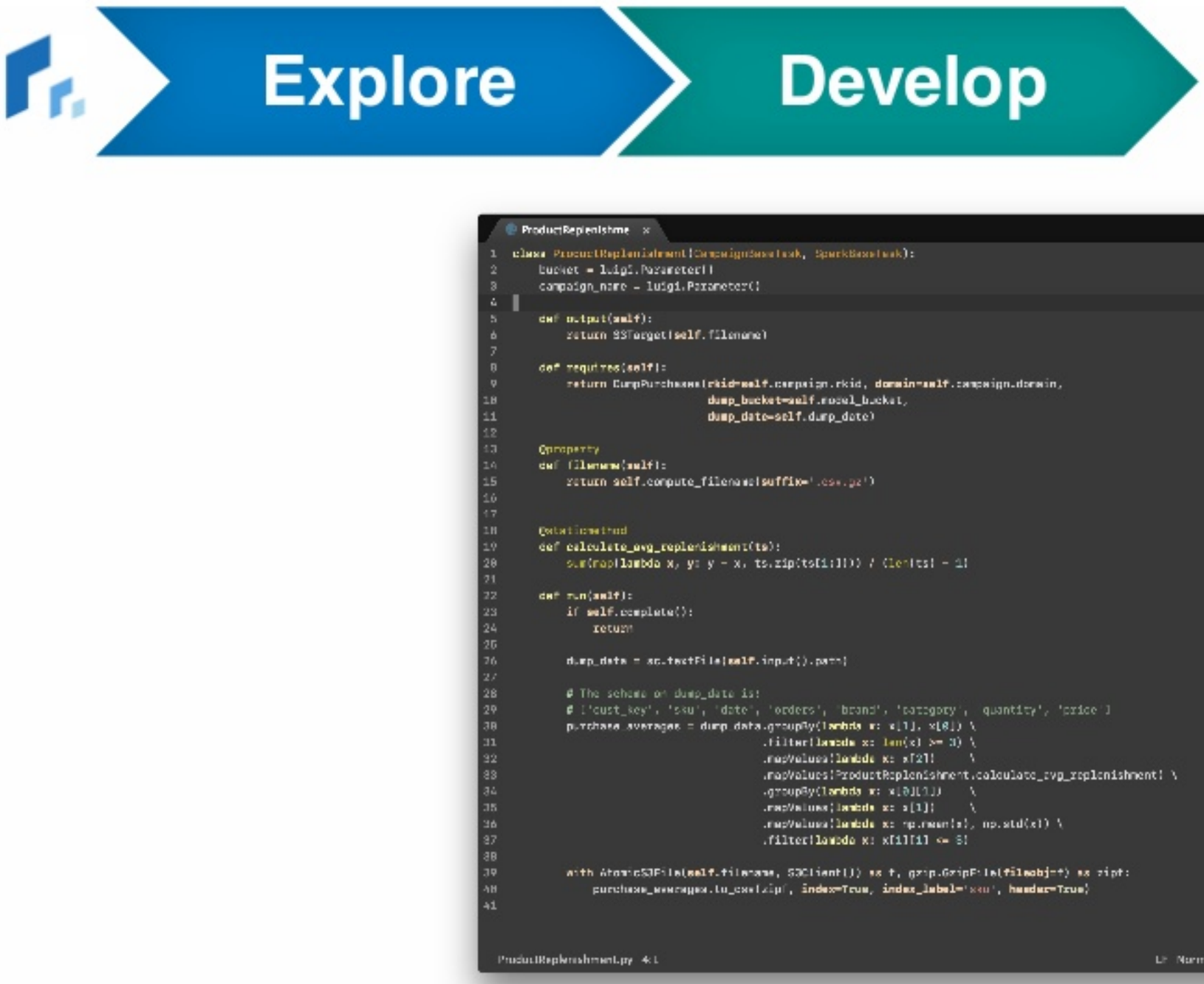
Explore Develop

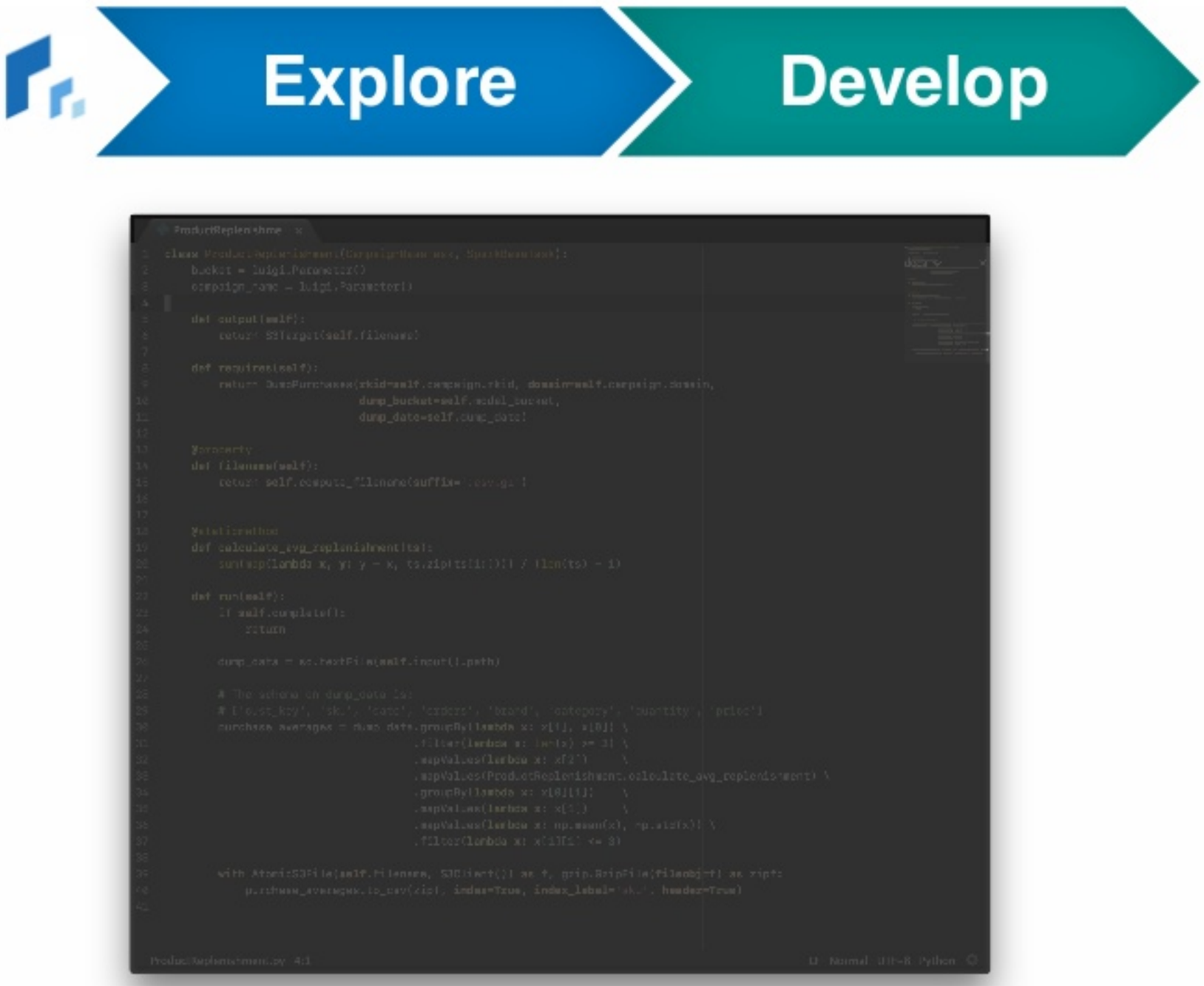


Explore Develop

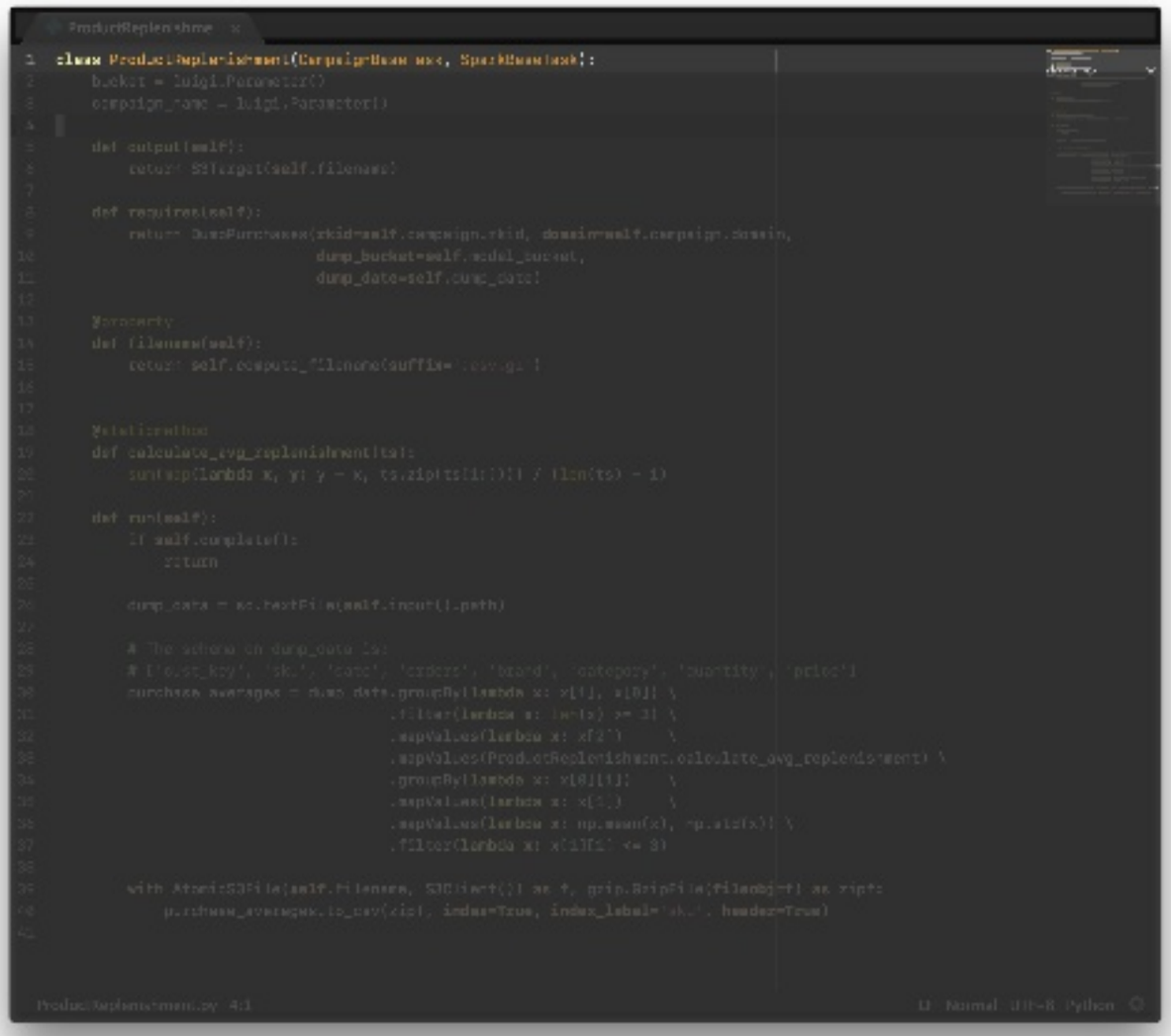
- Pipeline flow is fixed by the task definition

- Pipeline flow is fixed by the task definition
- Parameterization gets to be hugely verbose and redundant


Explore
Develop
ProductReplenishment
class ProductReplenishment(CampaignDaskTask, SparkDaskTask):
bucket = luigi.Parameter()
campaign_name = luigi.Parameter()
def output(self):
return S3Target(self.filename)
def requires(self):
return DumpPurchases(rkid=self.campaign.rkid, domain=self.campaign.domain,
dump_bucket=self.input_bucket,
dump_date=self.dump_date)
@property
def filename(self):
return self.compute_filename(suffix='.csv.gz')
@staticmethod
def calculate_avg_replenishment(ts):
sum(map(lambda x, y: y - x, ts.zip(ts[1:])) / (len(ts) - 1)
def run(self):
if self.complete():
return
dump_data = sc.textFile(self.input()).part()
# The schema of dump_data is:
# ['cust_key', 'ssu', 'date', 'orders', 'brand', 'category', 'quantity', 'price']
purchase_averages = dump_data.groupBy(lambda x: x[1], x[0]) \
.filter(lambda x: len(x) >= 3) \
.mapValues(lambda x: x[2]) \
.mapValues(ProductReplenishment.calculate_avg_replenishment) \
.groupBy(lambda x: x[0][1]) \
.mapValues(lambda x: x[1]) \
.mapValues(lambda x: np.mean(x), np.std(x)) \
.filter(lambda x: x[1][1] <= 5)
with AtomicS3File(self.filename, S3Client()) as f, gzip.GzipFile(fileobj=f) as zipf:
purchase_averages.to_csv(zipf, index=True, index_label='ssu', header=True)
ProductReplenishment.py 4:1
Normal
UTF-8
Python

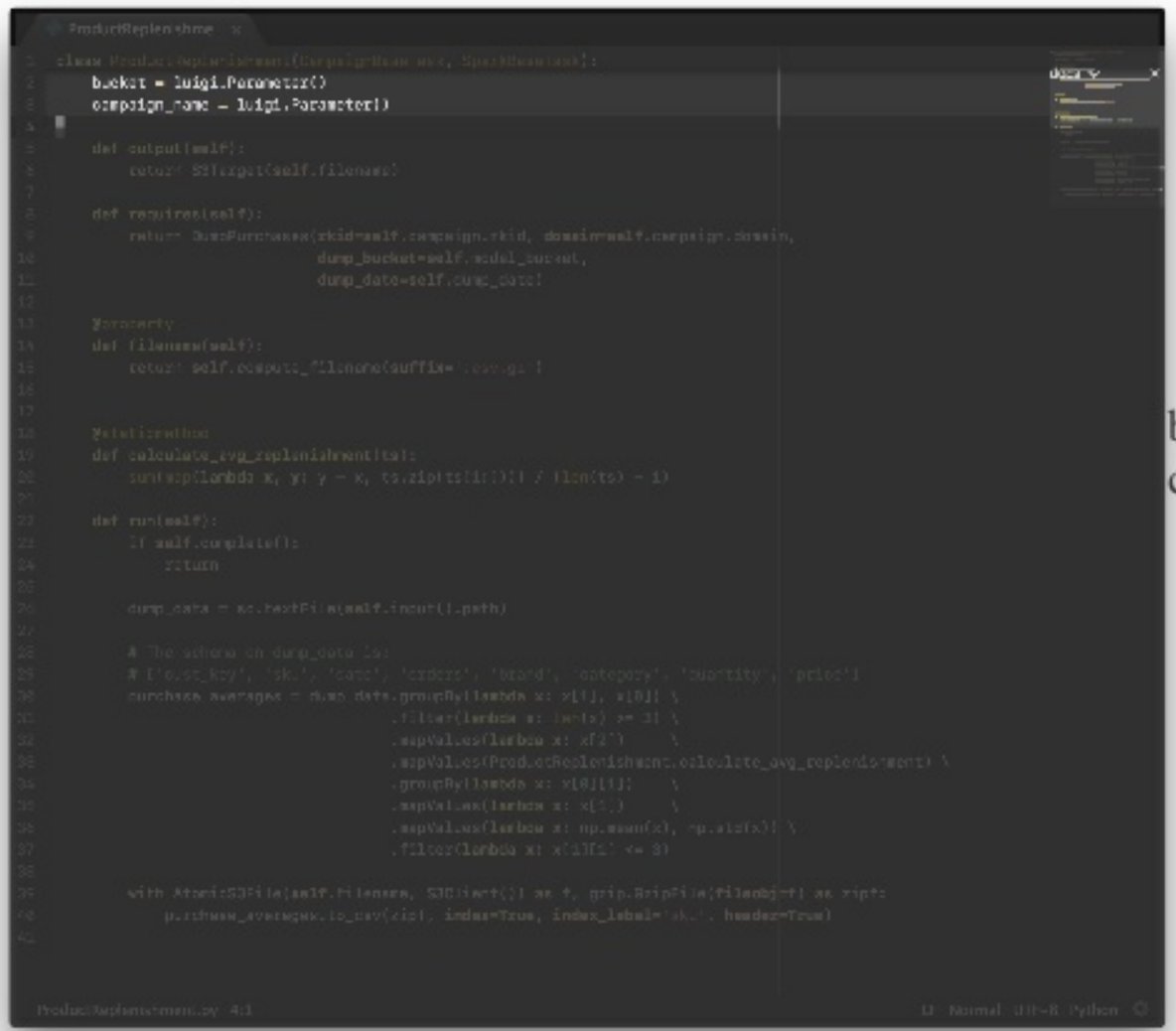

Explore
Develop

Explore Develop



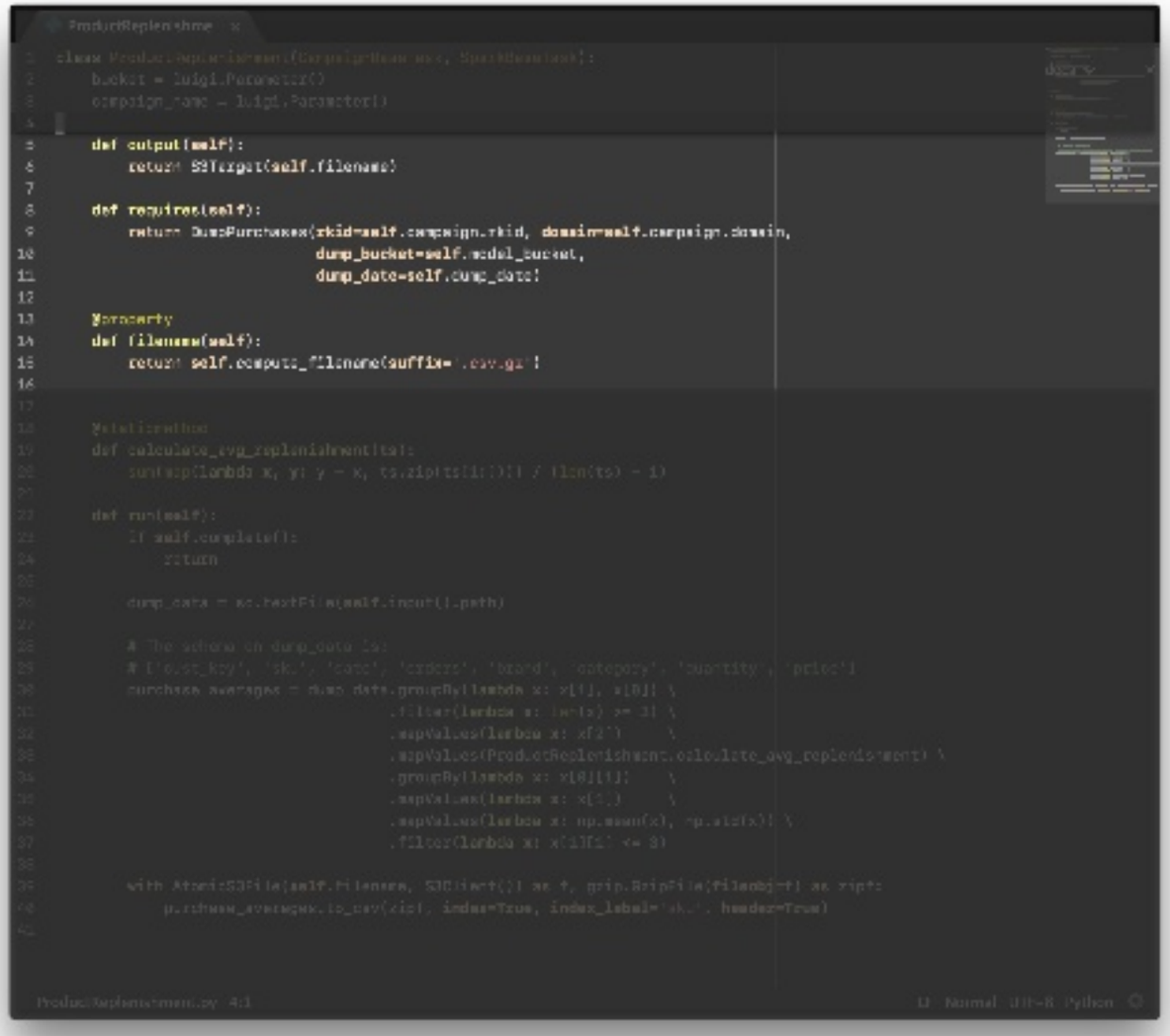

```
class ProductReplenishment(
                        CampaignBaseTask,
                        SparkBaseTask):
```

Explore Develop



```
bucket = luigi.Parameter()
campaign_name = campaign.Parameter()
```

Explore Develop



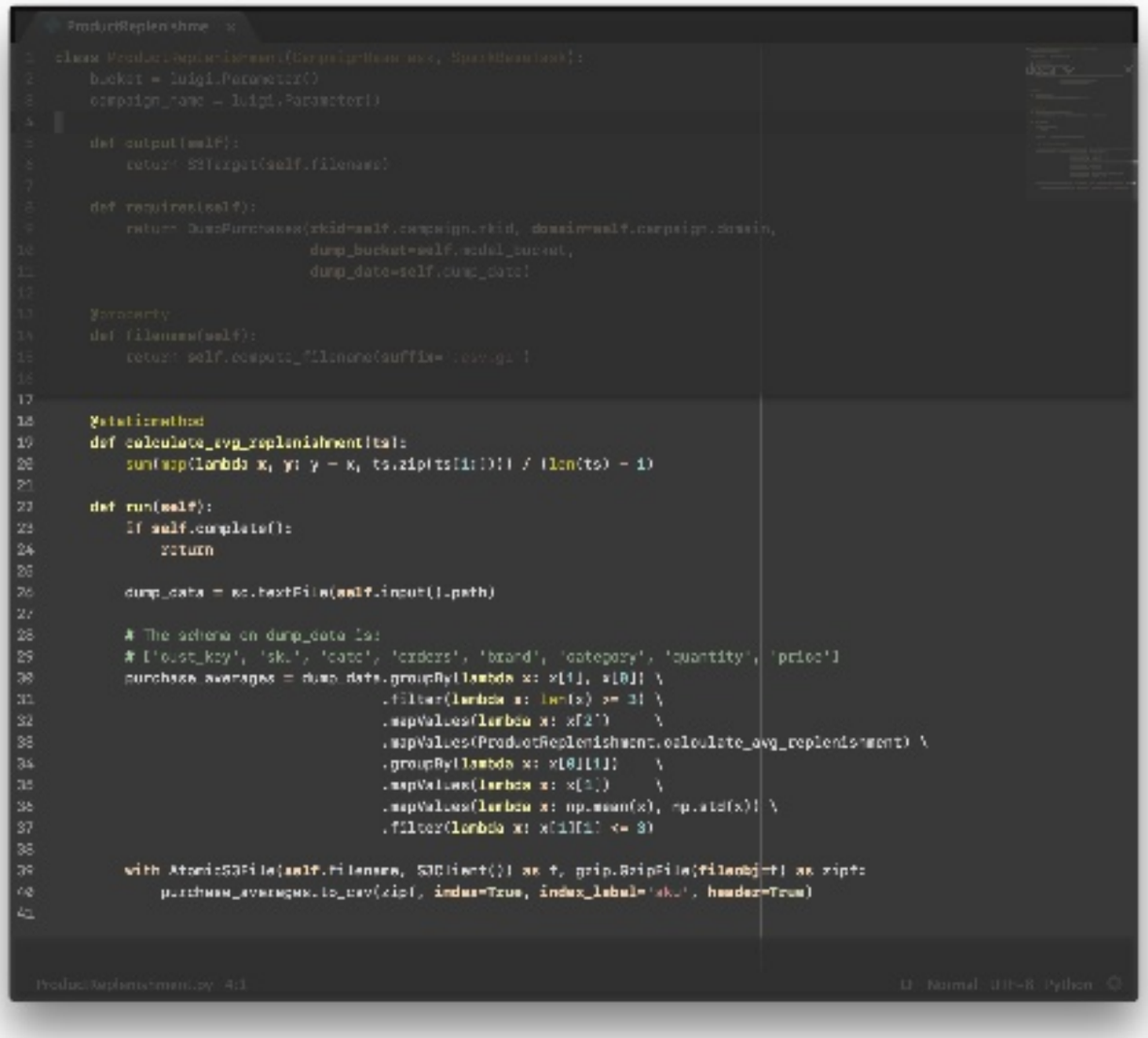
```
def output(self):
  return S3Target(self.filename)

def requires(self):
  return DumpPurchases(
      rkid=self.campaign.rkid,
      domain=self.campaign.domain,
      dump_bucket=self.model_bucket,
      dump_date=self.dump_date)

@property
def filename(self):
  return self.compute_filename(
      suffix='.csv.gz')
```

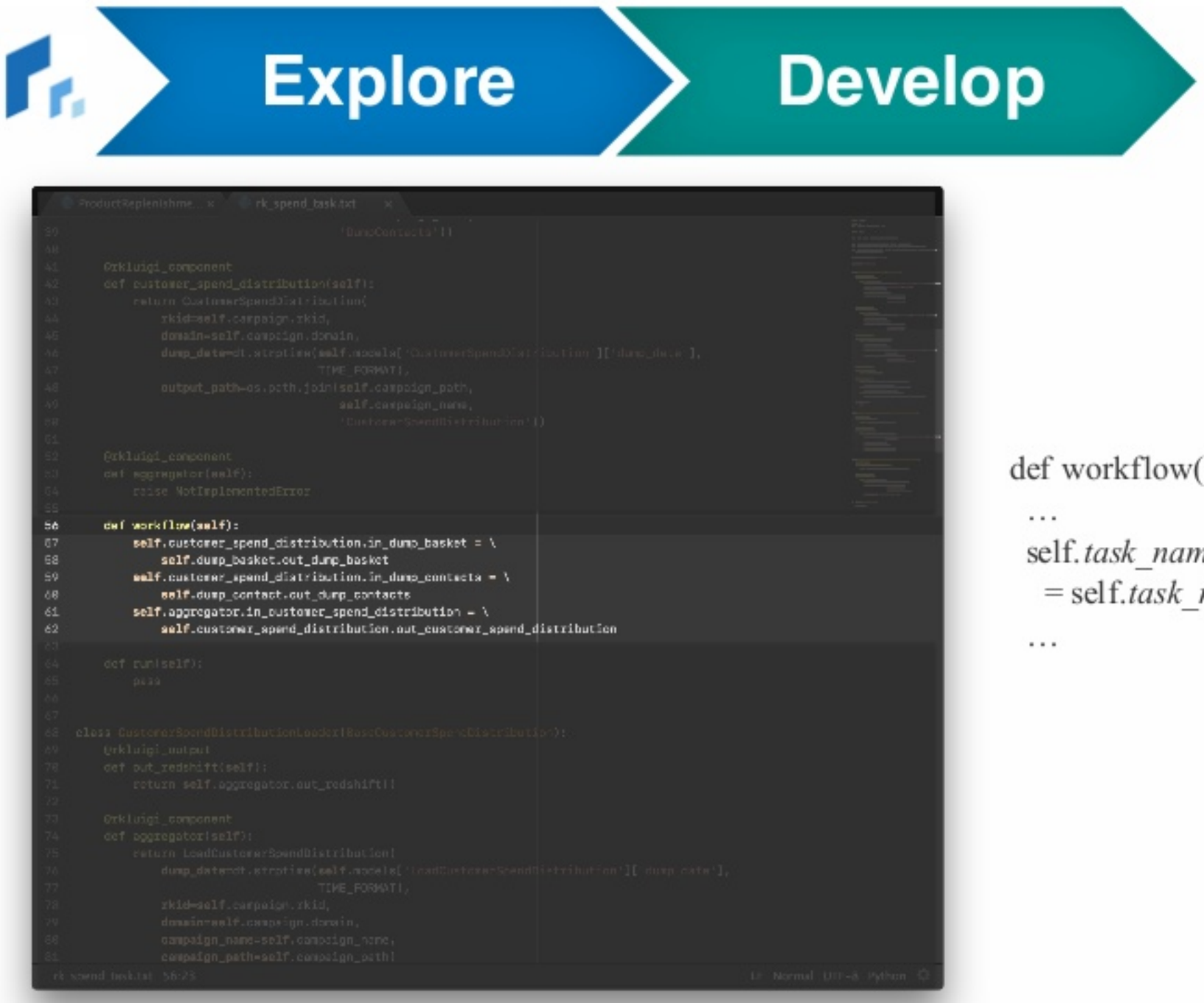
Explore

Develop




```
@staticmethod
def calculate_avg_replenishment(ts):
  return sum(map(-, zip(ts[1:], ts))
              \ (len(ts) - 1)

def run(self):
  dump_data = sc.textFile(
              self.input().path)
purchase_averages = dump_data
              .groupBy(…)
              .mapValues(…)
              …
              .filter(lambda x: x[1][1]<=D)
```

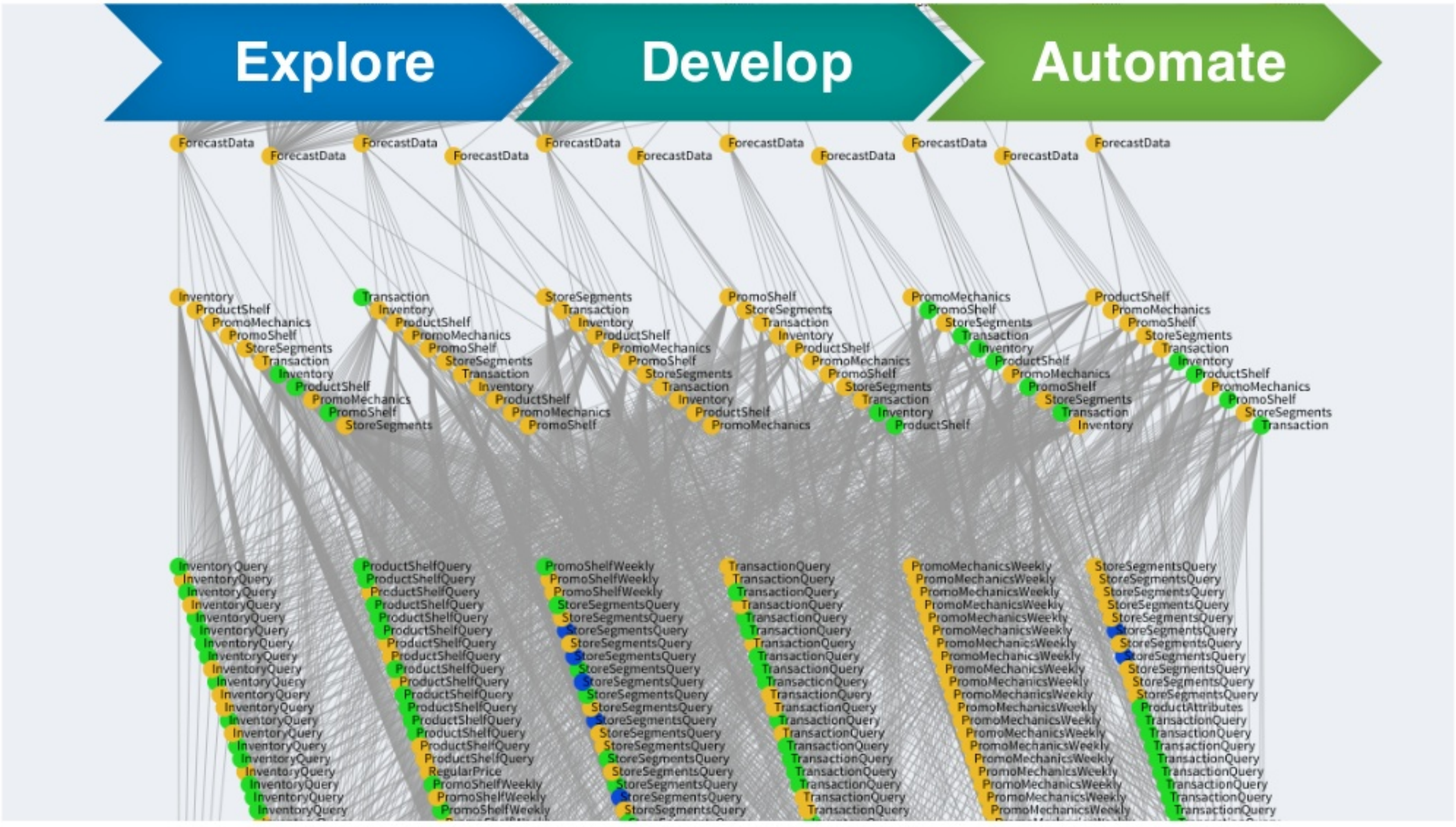


Explore

Develop

```
def workflow(self):
  ...
  self.task_name_1.input_param
    = self.task_name_2.output_param
  ...
```


Explore
Develop
Automate


Explore
Develop
Automate
ForecastData
Inventory
ProductShelf
PromoMechanics
PromoShelf
StoreSegments
Transaction
RegularPrice
ProductAttributes
InventoryQuery
ProductShelfQuery
PromoShelfWeekly
StoreSegmentsQuery
TransactionQuery
PromoMechanicsWeekly

Case Study: RubiOne

Personalized Search Results based on Relevancy Scores

$$W_f = aM_w + bR_w$$

$$\mathbf{R} = \begin{bmatrix} r_{1,1} & \cdots & r_{1,N} \\ \cdots & r_{i,j} & \cdots \\ r_{M,1} & \cdots & r_{M,N} \end{bmatrix}$$

$$\mathbf{R} = \mathbf{UP} = \begin{bmatrix} u_{1,1} & \cdots & u_{1,F} \\ \cdots & u_{i,j} & \cdots \\ u_{M,1} & \cdots & u_{M,F} \end{bmatrix} \begin{bmatrix} p_{1,1} & \cdots & p_{1,N} \\ \cdots & p_{i,j} & \cdots \\ p_{F,1} & \cdots & p_{F,N} \end{bmatrix}$$

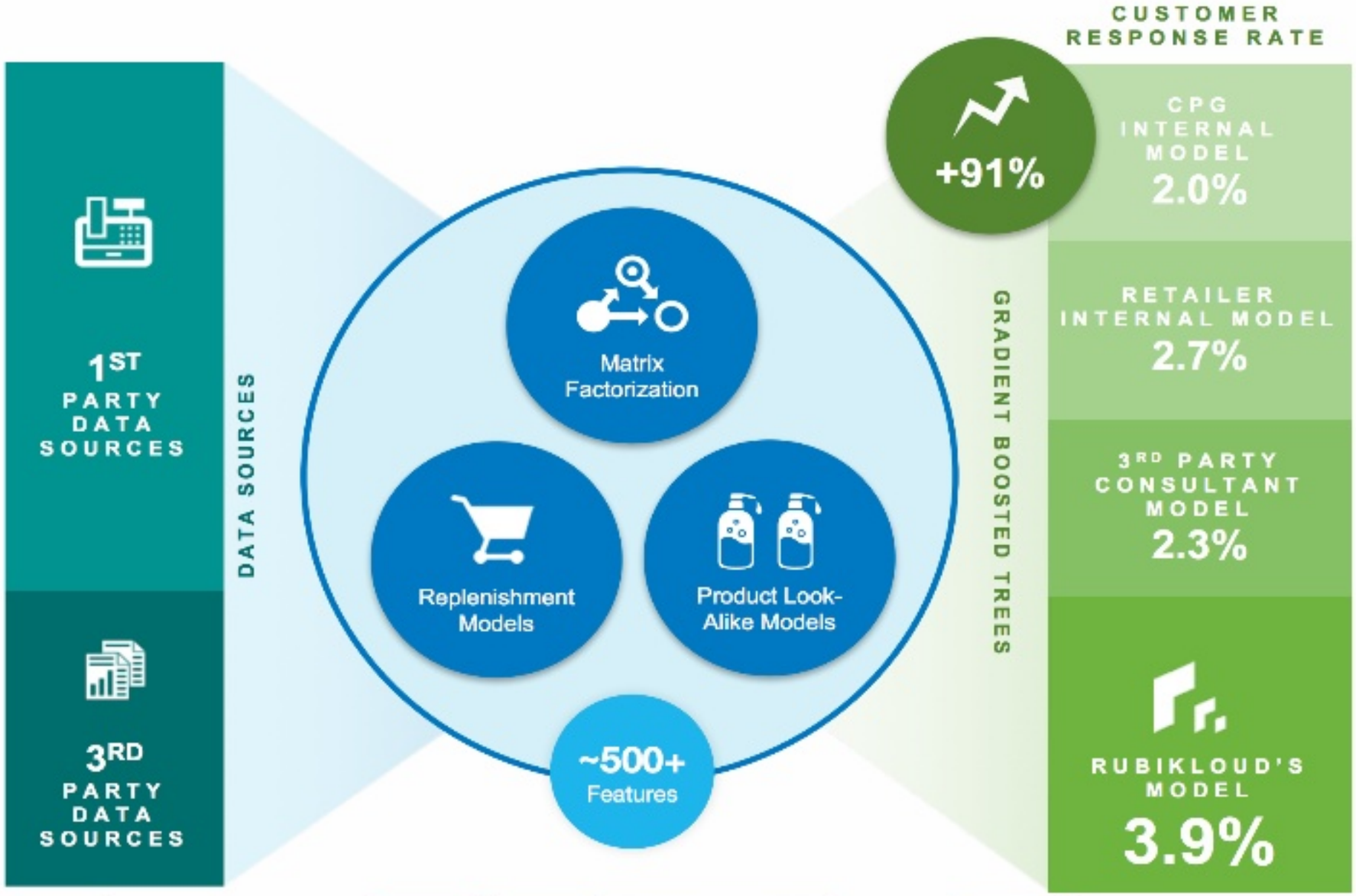
$$\min \sum_{r_{i,j} \text{ is known}} (r_{i,j} - x_i^T y_j) + \lambda(\|x_i\|^2 + \|y_j\|^2)$$

ORACLE

ENDECA

CASE STUDY: RUBIONE

Use Machine Learning to Solve Business Problems – At Scale



rubikloud®

FIN

That's All, Folks!