Hadoop service update

Recent service changes

- 8 new machines have been added to ANALYTIX recently
  - Private IPs
  - 24 cores, 512GB of RAM, 264TB of disk space

Cluster Name	Configuration	Primary Usage
lxhadoop	18 nodes (Cores – 288,Mem – 912GB,Storage – 1.29 PB)	ATLAS EventIndex project
analytix	48 nodes (Cores – 892,Mem – 7.5TB,Storage – 6 PB)	General Purpose
hadalytic	14 nodes (Cores – 196,Mem – 768GB,Storage – 2.15 PB)	BE development. Will be decommissioned
nxcals	20 nodes (Cores 480, Mem - 8 TB, Storage – 5 PB, 96GB in SSD)	Accelerator logging (NXCALS) project dedicated cluster

- Spark authentication enabled
  - spark.authentication=true is needed in the client config (if client config is not kept in sync)
  - When running in LOCAL mode spark.authentication.secretKey="phrase" is also needed

Hadoop High Availability

- Already enabled for 1 year for HDFS and YARN
  - 2 masters, one active at a time, automatic failover

- Referencing HDFS (do not use machine names)
  - hdfs://analytix/user/z/zbaranow # CORRECT
  - hdfs://analytix.cern.ch/user/z/zbaranow #NOT CORRECT
  - hdfs://ithdp1101.cern.ch/user/z/zbaranow #NOT CORRECT
  - /user/z/zbaranow #IS ALSO CORRECT IF…

- Please keep your core-site.xml, hdfs-site.xml and yarn-site.xml in sync with the target cluster

Interacting with Hadoop clusters

- What is needed to use the Hadoop clusters?
  - config files, binaries, e-group-based ACL granted, valid Kerberos ticket

- Supported ways of accessing the service
  - Puppet client module
    - Will be refactored/simplified – will use RPMs
  - Sourcing environments from CVMFS **(HDFS,HBASE, YARN)**
    - LXPLUS or own machines with CVMFS (see KB0004426)
  - Using SWAN for interactive analysis **(PYSPARK)**
  - RPM-based configuration syncing and binaries installation **(HDFS, HBASE, YARN) (will be available in June)**
  - ~~Ssh to a cluster machine~~

Closing ssh access to cluster machines

- Why?
  - Security
    - multitenant environment
    - separation of servers from clients
  - Consolidate service access
  - Better isolation of the service resources

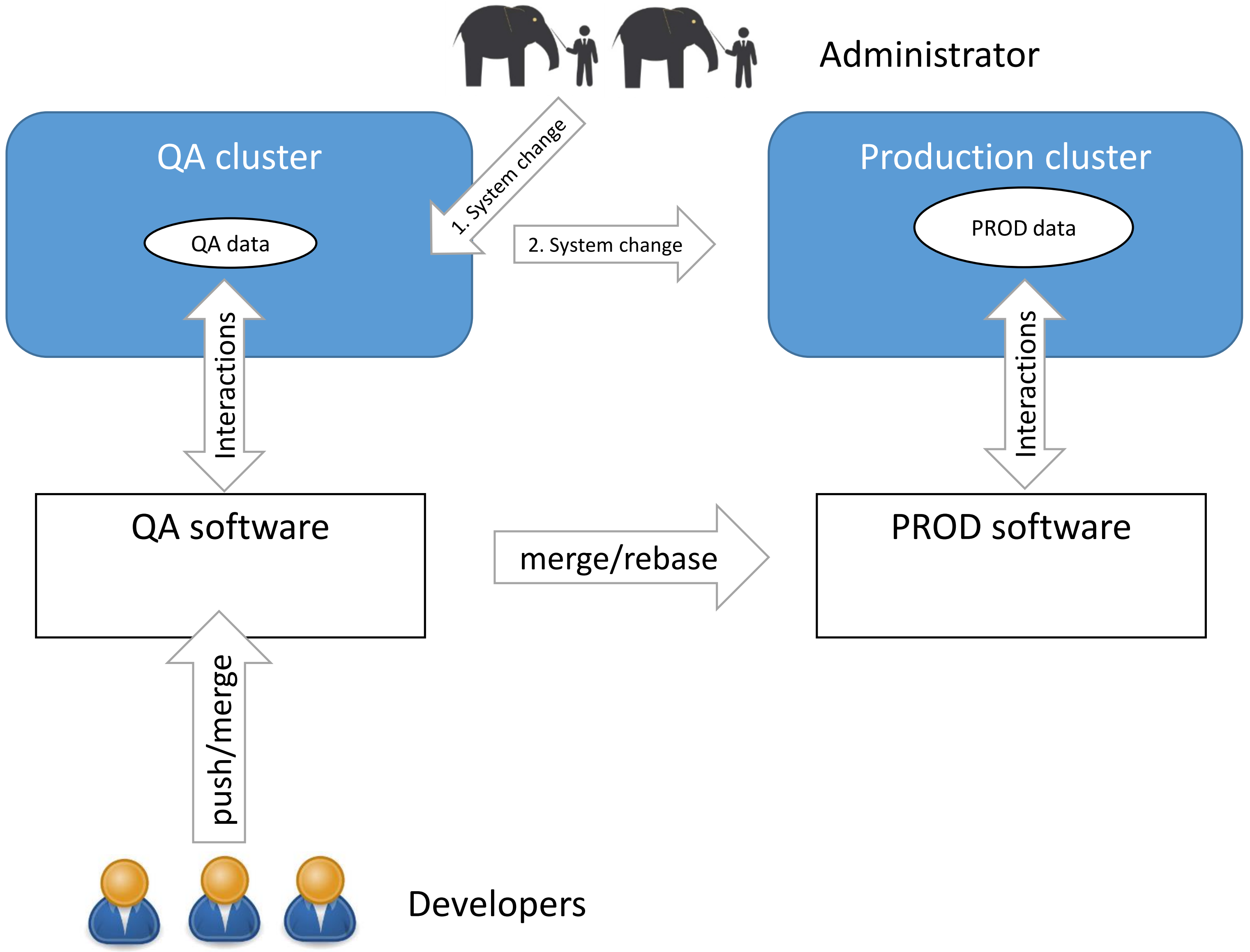
- Impact
  - Client has to use alternative environment
    - LXPLUS, Own machine, SWAN (for interactive pyspark)

- When?
  - Tentatively May

New QA cluster

- Goal of the cluster: <u>isolate production from QA systems</u>
  - USERS: pre-production version of users' code
    - Test and validate at 'small' scale
    - Integrate
    - Develop
  - ADMINS: validate changes before applying them on PROD
- Cluster specification
  - HA enabled
  - 10 physical machines
    - 16 cores, 64GB of RAM, 64TB of disk space
- When: end of April

Enabling standard system/application development process with QA cluster



Decoupling Spark from Cloudera’s distribution

- Why?
  - Cloudera ‘free’ distribution is significantly lagging behind – currently it is still on v1.6
  - Users wants to use the latest Spark versions
  - Infrastructure problems: spark history server 1.6 does not support ‘well’ jobs from >= 2.0
- IT Spark rpm packages
  - Repo: http://linuxsoft.cern.ch/internal/repos/hdp7-stable/x86_64/os/
  - RPM name: spark-bin-[version]
- When
  - End of March

Integration with user groups

- Idea
    - Service level user groups will be mapped to e-groups
    - Nested e-groups supported

- Advantages
    - Fine grained segmentation of user communities
    - ACLs on service resources per user community
        - HDFS, YARN, HBASE…
    - Isolation of the projects
- When tentatively April

Apache-CERN Vanilla available

- Due to the needs of NXCALS project we have rolled out Apache Hadoop distribution (instead of Cloudera)

- Impressions
  - No problems so far
  - Stable

- Advantages
  - Better control on the software stack
  - No patches/feature mixing
  - Rapid bug fixing and roll-out
  - Customizations possible
- Disadvantages
  - We have to maintain it

- On NXCLAS now, considering full service migration to the Apache distribution

pyspark – Spark Python API

- User Requirement: need full stack of Python software suitable to perform Data Science tasks, e.g. ML studies, dataset operations, analytics

- PROBLEM: Limited set of packages available on cluster

- Hadoop service recommends the following options depending on needs

- OPTION1: Use the python distribution from CVMFS
  - Prerequisites: Users need to mount CVMFS on client/edge nodes and install HEP_OSlibs
  - Hadoop service is committed to making CVMFS available on all cluster nodes
  - Advantages: Availability of widely used python packages and experiment software

pyspark – Spark Python API

- OPTION2: Install python distribution (e.g Anaconda2) on AFS public
  - Hadoop service is committed to making AFS available on cluster nodes
  - Advantages: Full user control over python distribution, able to install any python modules, supports agile software development

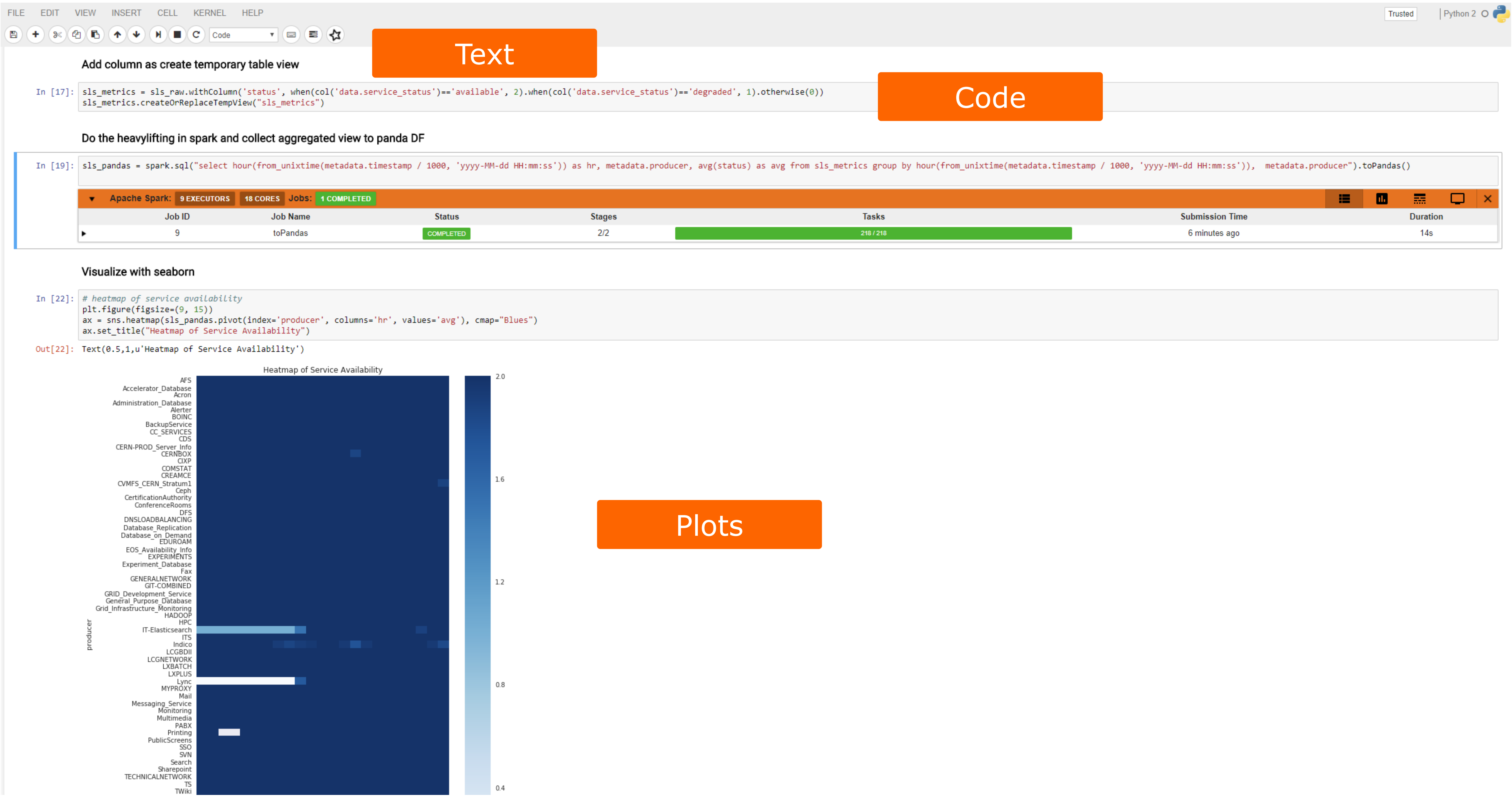
- OPTION3: Shipping custom python packages to pyspark executors
  - Recommended only if users are unable to use CVMFS or AFS python distributions
  - Advantages: Full user control over python packages without any need for shared file system

- KB Article - How to set python distribution for pyspark
  - https://cern.service-now.com/service-portal/article.do?n=KB0005361

SWAN – Jupyter Notebooks On Demand

- SWAN – Service for web based analysis
  - collaboration between EP-SFT, IT-DB and IT-ST
- A web-based interactive interface and platform that combines code, equations, text and visualisations
  - Ideal for exploration, reproducibility, collaboration
- Fully Integrated with IT Hadoop Clusters
  - Modern, powerful and scalable platform for data analysis
  - pyspark driver stays in the SWAN container
  - Pyspark executors on the IT Hadoop Clusters (doing the heavy lifting)

NEW



FILE EDIT VIEW INSERT CELL KERNEL HELP

Trusted | Python 2

Text

Code

Plots

Add column as create temporary table view

```
In [17]: sls_metrics = sls_raw.withColumn('status', when(col('data.service_status')=='available', 2).when(col('data.service_status')=='degraded', 1).otherwise(0))
         sls_metrics.createOrReplaceTempView("sls_metrics")
```

Do the heavylifting in spark and collect aggregated view to panda DF

```
In [19]: sls_pandas = spark.sql("select hour(from_unixtime(metadata.timestamp / 1000, 'yyyy-MM-dd HH:mm:ss')) as hr, metadata.producer, avg(status) as avg from sls_metrics group by hour(from_unixtime(metadata.timestamp / 1000, 'yyyy-MM-dd HH:mm:ss')),  metadata.producer").toPandas()
```

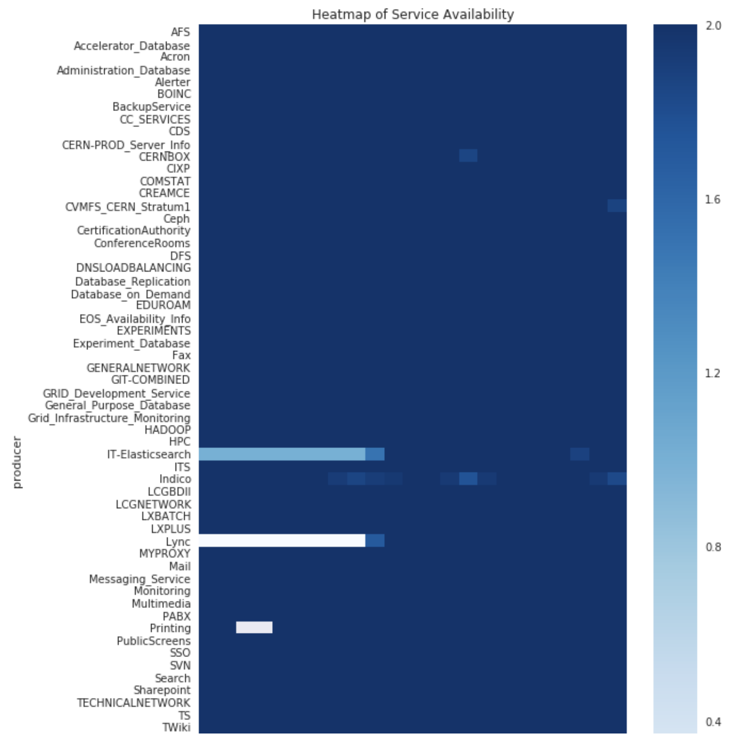
Apache Spark: 9 EXECUTORS 18 CORES Jobs: 1 COMPLETED

Job ID	Job Name	Status	Stages	Tasks	Submission Time	Duration
9	toPandas	COMPLETED	2/2	218 / 218	6 minutes ago	14s

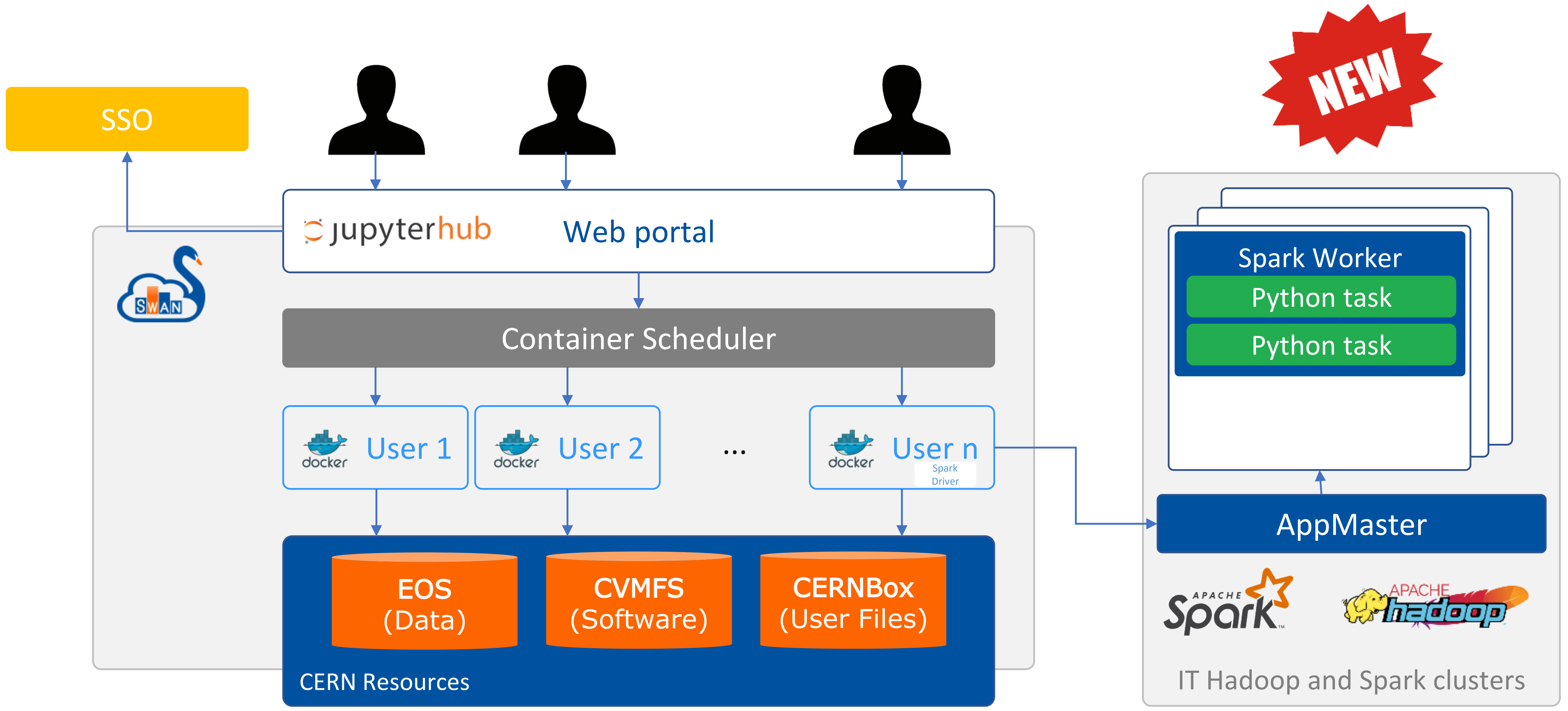
Visualize with seaborn

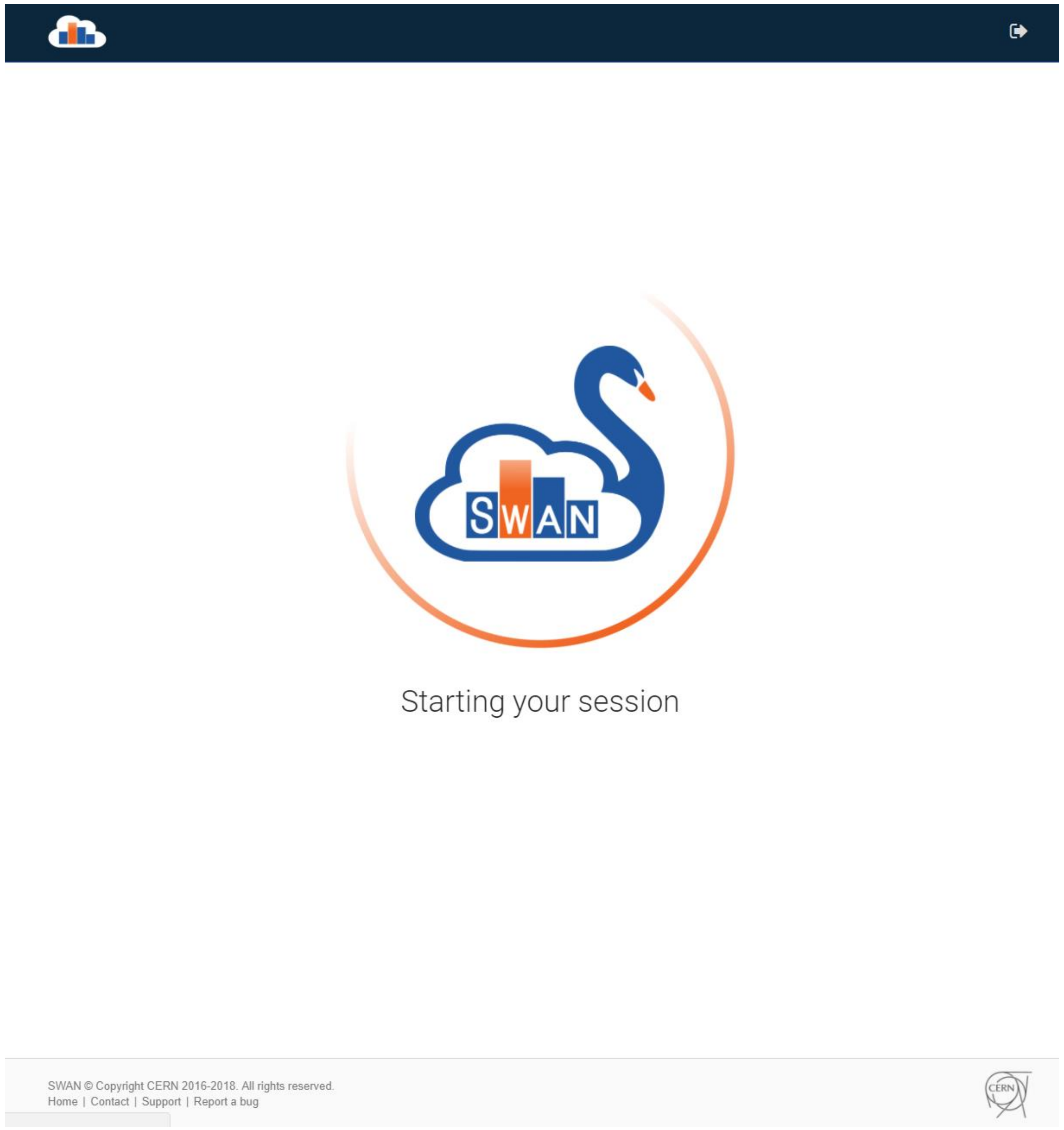
```
In [22]: # heatmap of service availability
         plt.figure(figsize=(9, 15))
         ax = sns.heatmap(sls_pandas.pivot(index='producer', columns='hr', values='avg'), cmap="Blues")
         ax.set_title("Heatmap of Service Availability")

Out[22]: Text(0.5,1,u'Heatmap of Service Availability')
```



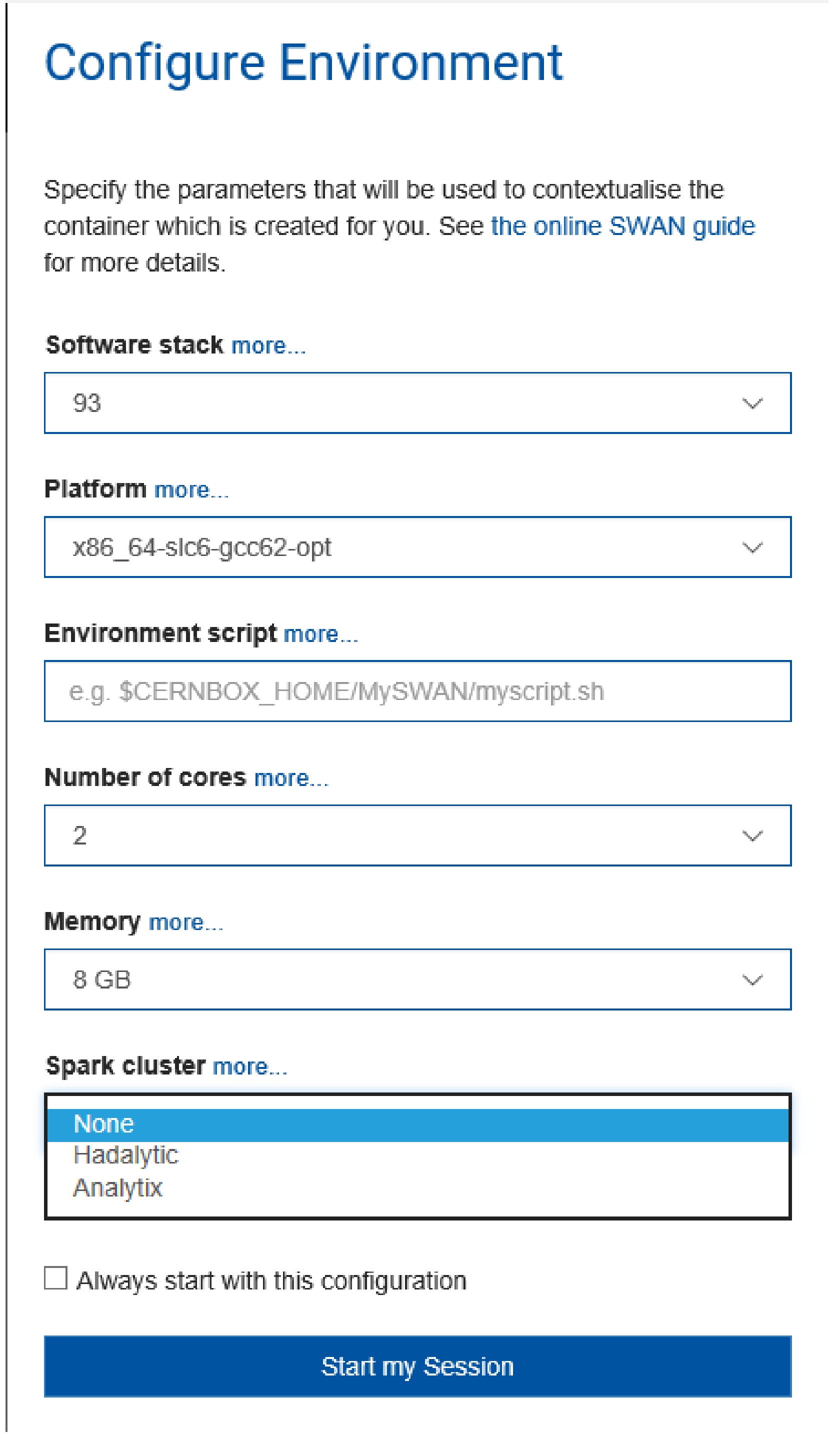
SWAN_Spark – Architecture





Starting your session

SWAN © Copyright CERN 2016-2018. All rights reserved.
Home | Contact | Support | Report a bug



Configure Environment

Specify the parameters that will be used to contextualise the container which is created for you. See the online SWAN guide for more details.

Software stack more...
93

Platform more...
x86_64-slc6-gcc62-opt

Environment script more...
e.g. $CERNBOX_HOME/MySWAN/myscript.sh

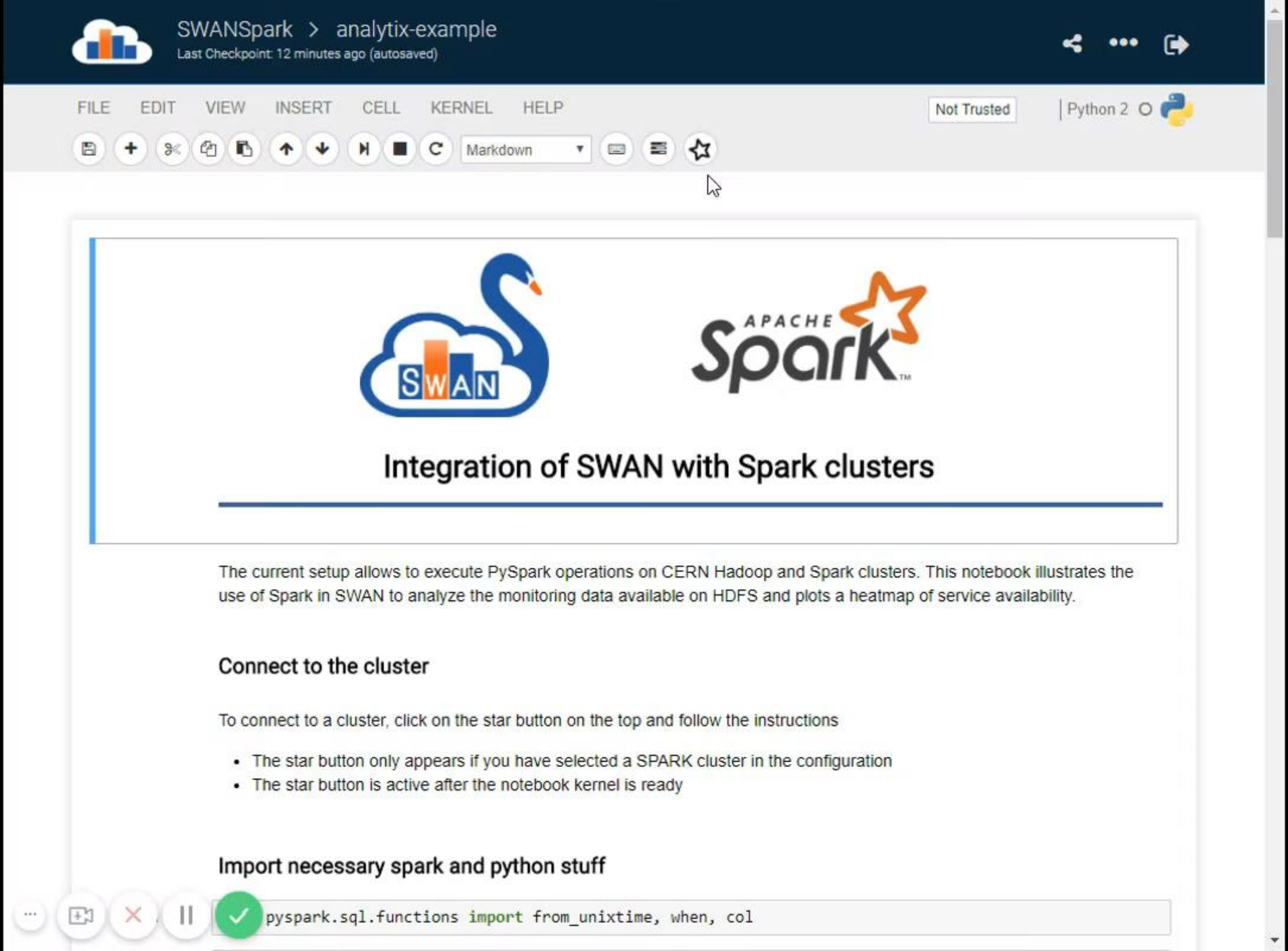
Number of cores more...
2

Memory more...
8 GB

Spark cluster more...
None
Hadalytic
Analytix

☐ Always start with this configuration

Start my Session


SWANSpark > analytix-example
Last Checkpoint: 12 minutes ago (autosaved)
FILE
EDIT
VIEW
INSERT
CELL
KERNEL
HELP
Not Trusted
Python 2
Markdown

Integration of SWAN with Spark clusters

The current setup allows to execute PySpark operations on CERN Hadoop and Spark clusters. This notebook illustrates the use of Spark in SWAN to analyze the monitoring data available on HDFS and plots a heatmap of service availability.

Connect to the cluster

To connect to a cluster, click on the star button on the top and follow the instructions

- The star button only appears if you have selected a SPARK cluster in the configuration
- The star button is active after the notebook kernel is ready

Import necessary spark and python stuff

```
pyspark.sql.functions import from_unixtime, when, col
```

SWAN_Spark features

- Spark Monitor – jupyter notebook extension
  - For live monitoring of spark jobs spawned from the notebook
  - Access to Spark WEB UI from the notebook
  - Several other features to debug Spark application

- Spark Connector – hiding the spark configuration complexity
  - User is presented with Spark Session (Spark) and Spark Context (sc)
  - Ability to bundle configurations specific to user communities
  - Ability to add configuration

- Concept of projects and sharing projects within SWAN

- Features with the goal of lowering the barrier for large scale distributed analysis with Apache Spark (PySpark)

SWAN_Spark – Demo

SWAN_Spark

- Pre-release of SWAN_Spark is fully available to HADOOP users
  - Targeting production release by mid-April
- URL – http://swan006.cern.ch
- Example Notebooks - <analytix-example>
- Try it and send feedback to
  - ai-hadoop-admins@cern.ch
  - swan-admins@cern.ch

Recap

- Provisioning of new QA cluster
- Proposal to discontinue ssh access to Hadoop boxes
- New Apache-CERN Hadoop distribution
- Streaming Hadoop client configuration and Setup
- Service integration with egroups
- Solutions to set Python distribution for PySpark
- New hosted notebook solution for PySpark (SWAN) available