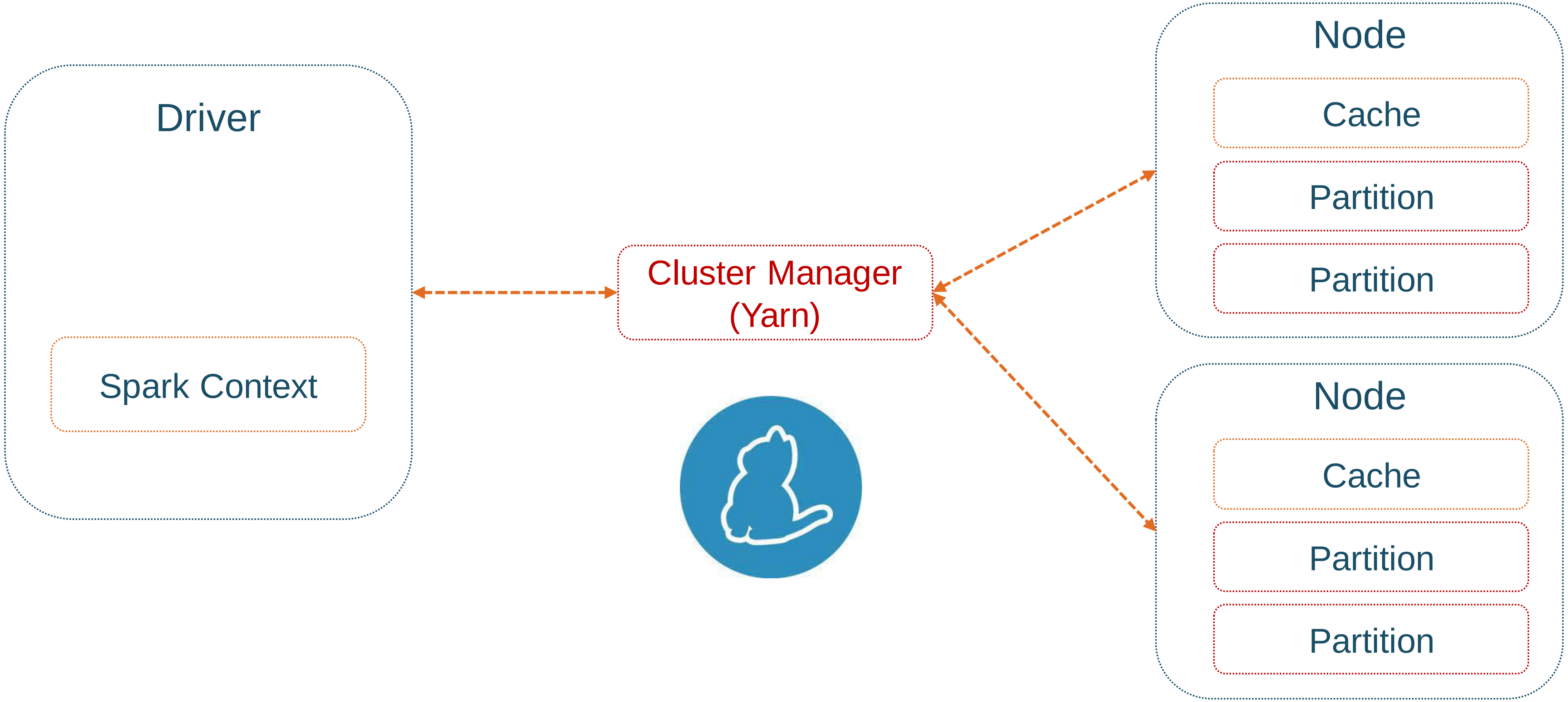
Spark cluster & production

SELEZNEV ARTEM
HEAD OF CVM ANALYTICS @ MAGNIT

SPARK CLUSTER

НА КЛАСТЕРЕ ДРУГОЕ УРАВЛЕНИЕ

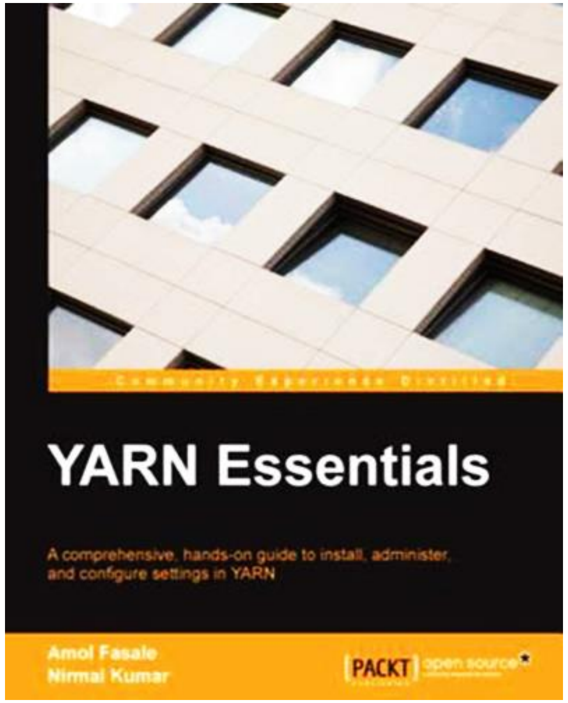


НА КЛАСТЕРЕ ДРУГОЕ УРАВЛЕНИЕ

- yarn app –list
- yarn app -status
- yarn app –appStates
- yarn app –destroy appId
- yarn app –kill appId

https://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YarnCommands.html

НА КЛАСТЕРЕ ДРУГОЕ УРАВЛЕНИЕ



Community Experience Distilled

YARN Essentials

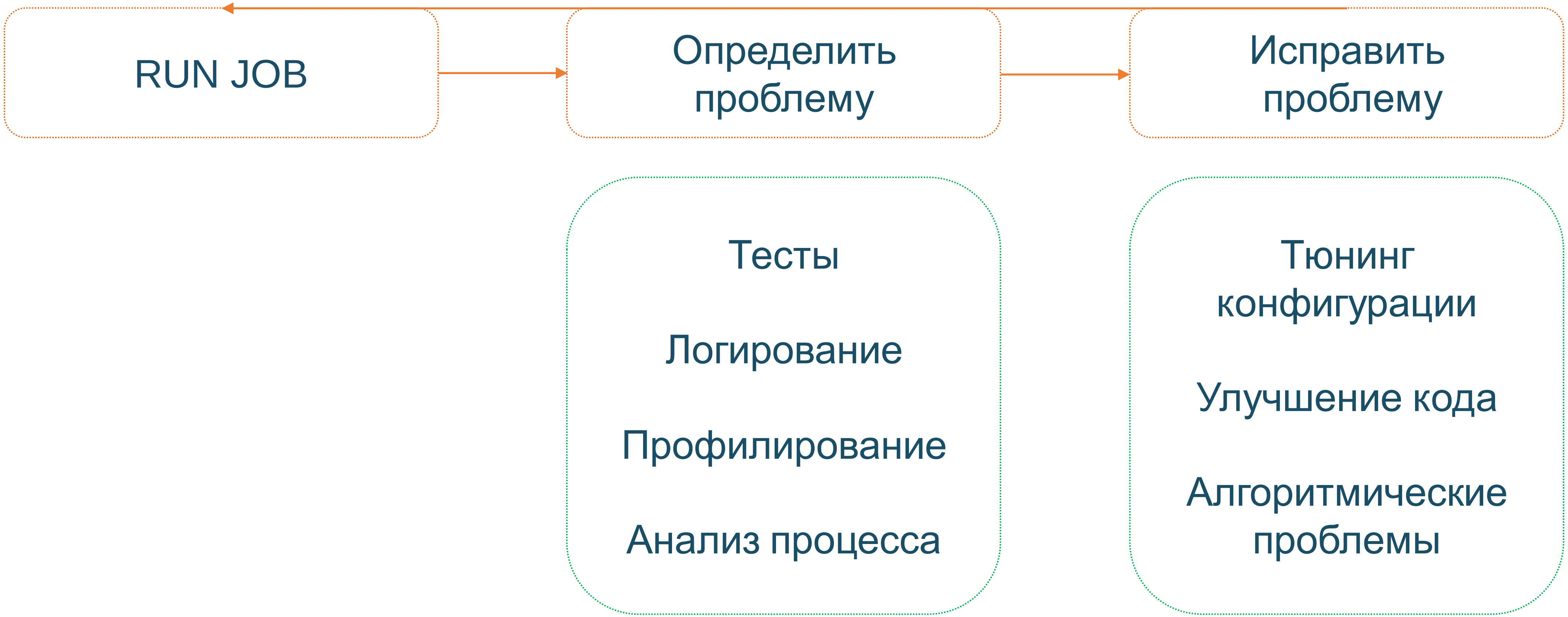
A comprehensive, hands-on guide to install, administer, and configure settings in YARN

Amol Fasale
Nirmal Kumar

[PACKT] open source

https://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YarnCommands.html

БОЛЬШЕ ДАННЫХ – СЛОЖНЕЕ ПРОЦЕССЫ



RUN JOB

Определить проблему

Исправить проблему

Тесты

Логирование

Профилирование

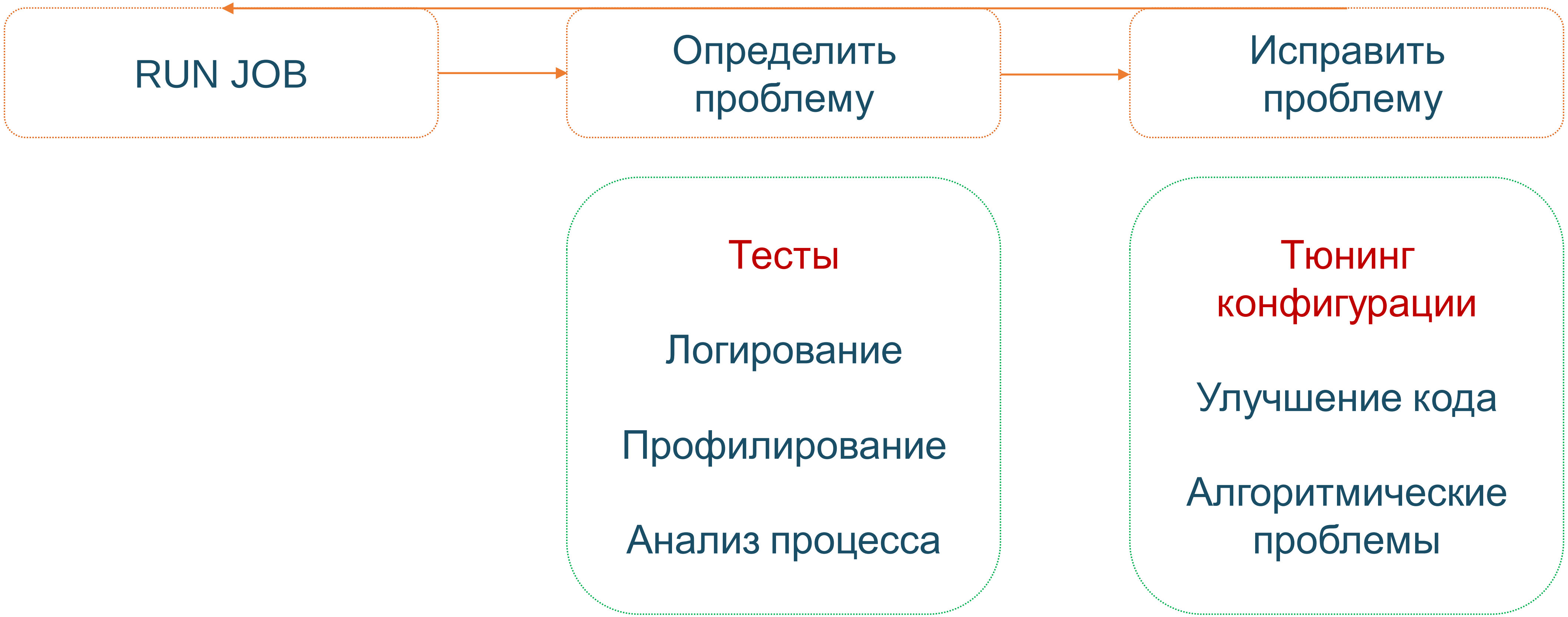
Анализ процесса

Тюнинг конфигурации

Улучшение кода

Алгоритмические проблемы

БОЛЬШЕ ДАННЫХ – СЛОЖНЕЕ ПРОЦЕССЫ



RUN JOB

Определить проблему

Исправить проблему

Тесты

Логирование

Профилирование

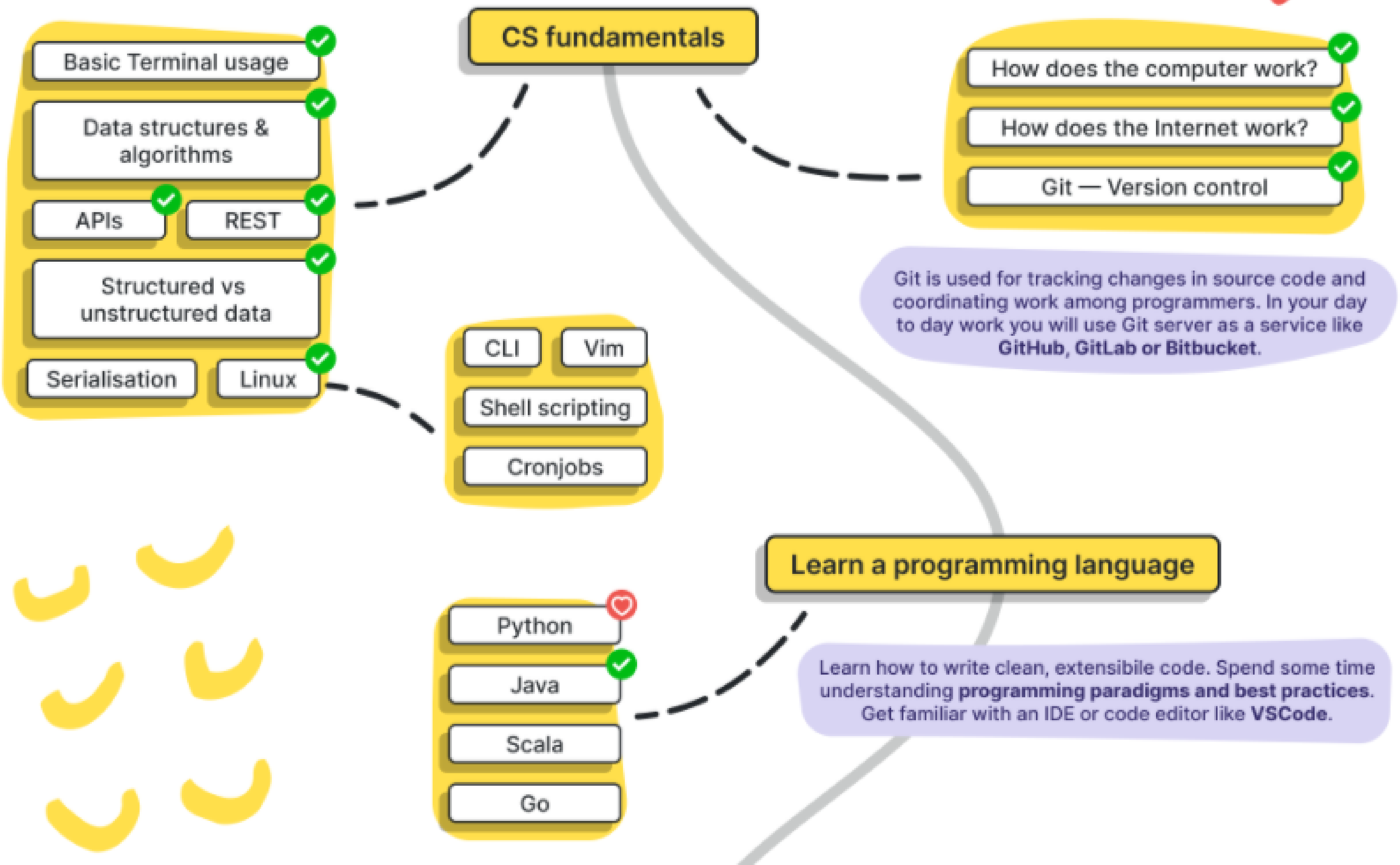
Анализ процесса

Тюнинг конфигурации

Улучшение кода

Алгоритмические проблемы

SPARK CLUSTER DEEPER…



CS fundamentals

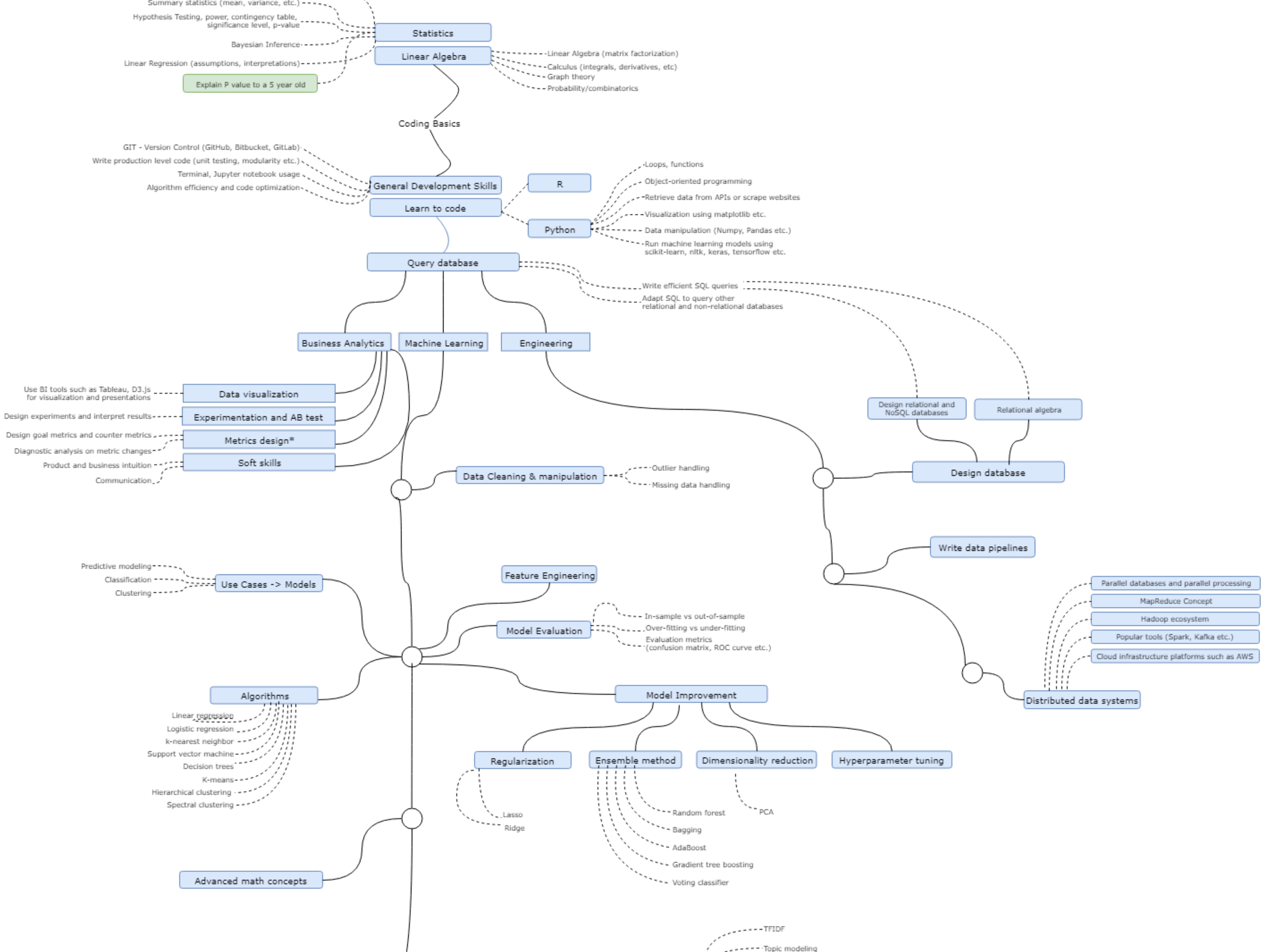
- Basic Terminal usage ✓
- Data structures & algorithms ✓
- APIs ✓
- REST ✓
- Structured vs unstructured data ✓
- Serialisation
- Linux ✓
  - CLI
  - Vim
  - Shell scripting
  - Cronjobs
- How does the computer work? ✓
- How does the Internet work? ✓
- Git — Version control ✓

Git is used for tracking changes in source code and coordinating work among programmers. In your day to day work you will use Git server as a service like **GitHub, GitLab or Bitbucket**.

Learn a programming language

- Python ♥
- Java ✓
- Scala
- Go

Learn how to write clean, extensibile code. Spend some time understanding **programming paradigms and best practices**. Get familiar with an IDE or code editor like **VSCode**.



Summary statistics (mean, variance, etc.)

Hypothesis Testing, power, contingency table, significance level, p-value

Bayesian Inference

Linear Regression (assumptions, interpretations)

Explain P value to a 5 year old

Statistics

Linear Algebra

Linear Algebra (matrix factorization)

Calculus (integrals, derivatives, etc)

Graph theory

Probability/combinatorics

Coding Basics

GIT - Version Control (GitHub, Bitbucket, GitLab)

Write production level code (unit testing, modularity etc.)

Terminal, Jupyter notebook usage

Algorithm efficiency and code optimization

General Development Skills

Learn to code

R

Python

Loops, functions

Object-oriented programming

Retrieve data from APIs or scrape websites

Visualization using matplotlib etc.

Data manipulation (Numpy, Pandas etc.)

Run machine learning models using scikit-learn, nltk, keras, tensorflow etc.

Query database

Write efficient SQL queries

Adapt SQL to query other relational and non-relational databases

Business Analytics

Machine Learning

Engineering

Use BI tools such as Tableau, D3.js for visualization and presentations

Data visualization

Design experiments and interpret results

Experimentation and AB test

Design goal metrics and counter metrics

Diagnostic analysis on metric changes

Metrics design*

Product and business intuition

Communication

Soft skills

Data Cleaning & manipulation

Outlier handling

Missing data handling

Design relational and NoSQL databases

Relational algebra

Design database

Write data pipelines

Parallel databases and parallel processing

MapReduce Concept

Hadoop ecosystem

Popular tools (Spark, Kafka etc.)

Cloud infrastructure platforms such as AWS

Distributed data systems

Predictive modeling

Classification

Clustering

Use Cases -> Models

Feature Engineering

Model Evaluation

In-sample vs out-of-sample

Over-fitting vs under-fitting

Evaluation metrics (confusion matrix, ROC curve etc.)

Algorithms

Linear regression

Logistic regression

k-nearest neighbor

Support vector machine

Decision trees

K-means

Hierarchical clustering

Spectral clustering

Model Improvement

Regularization

Lasso

Ridge

Ensemble method

Random forest

Bagging

AdaBoost

Gradient tree boosting

Voting classifier

Dimensionality reduction

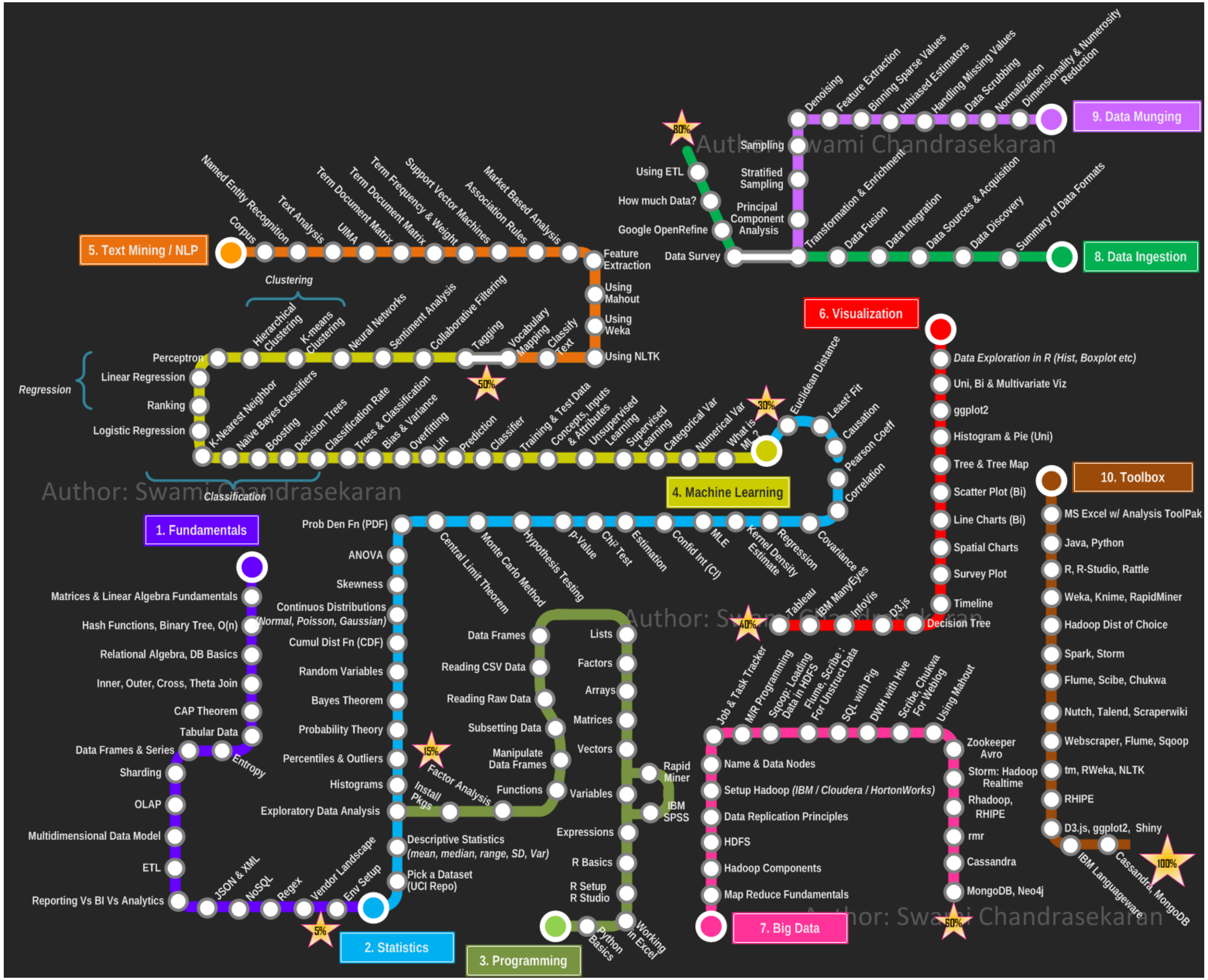
PCA

Hyperparameter tuning

Advanced math concepts

TFIDF

Topic modeling



github.com/MrMimic/data-scientist-roadmap

ЖИЗНЬ НА КЛАСТЕРЕ

VIRTUAL

VDS RUS

VDS OTHERS

ЖИЗНЬ НА КЛАСТЕРЕ

VIRTUAL

VDS RUS

VDS OTHERS

= 7499 атак

ЖИЗНЬ НА КЛАСТЕРЕ

- Ansible

ЖИЗНЬ НА КЛАСТЕРЕ

- Ansible
    - Apache Hadoop3

ЖИЗНЬ НА КЛАСТЕРЕ

- Ansible
    - Apache Hadoop3
        - Apache Spark 3

ЖИЗНЬ НА КЛАСТЕРЕ

- Ansible
- Apache Hadoop3
- Apache Spark 3
- Apache Drill

ЖИЗНЬ НА КЛАСТЕРЕ

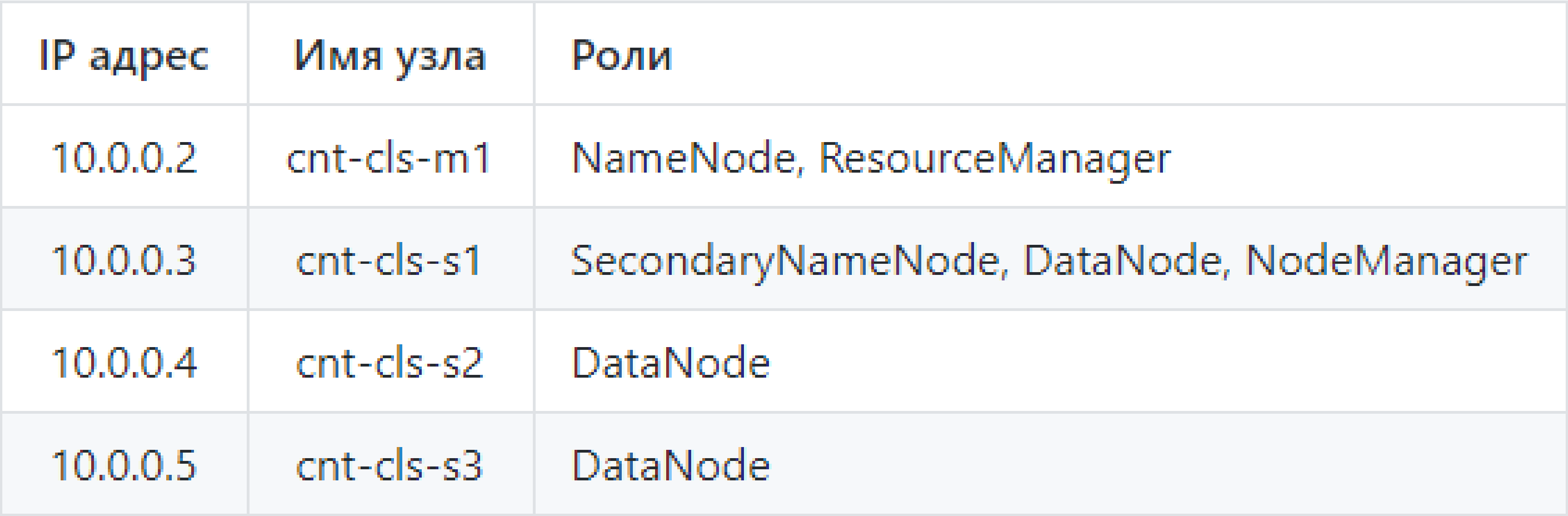
- Ansible
- Apache Hadoop3
- Apache Spark 3
- Apache Drill
- JupyterHub + Kernel

ЖИЗНЬ НА КЛАСТЕРЕ

- Ansible
- Apache Hadoop3
- Apache Spark 3
- Apache Drill
- JupyterHub + Kernel
- Feature Store

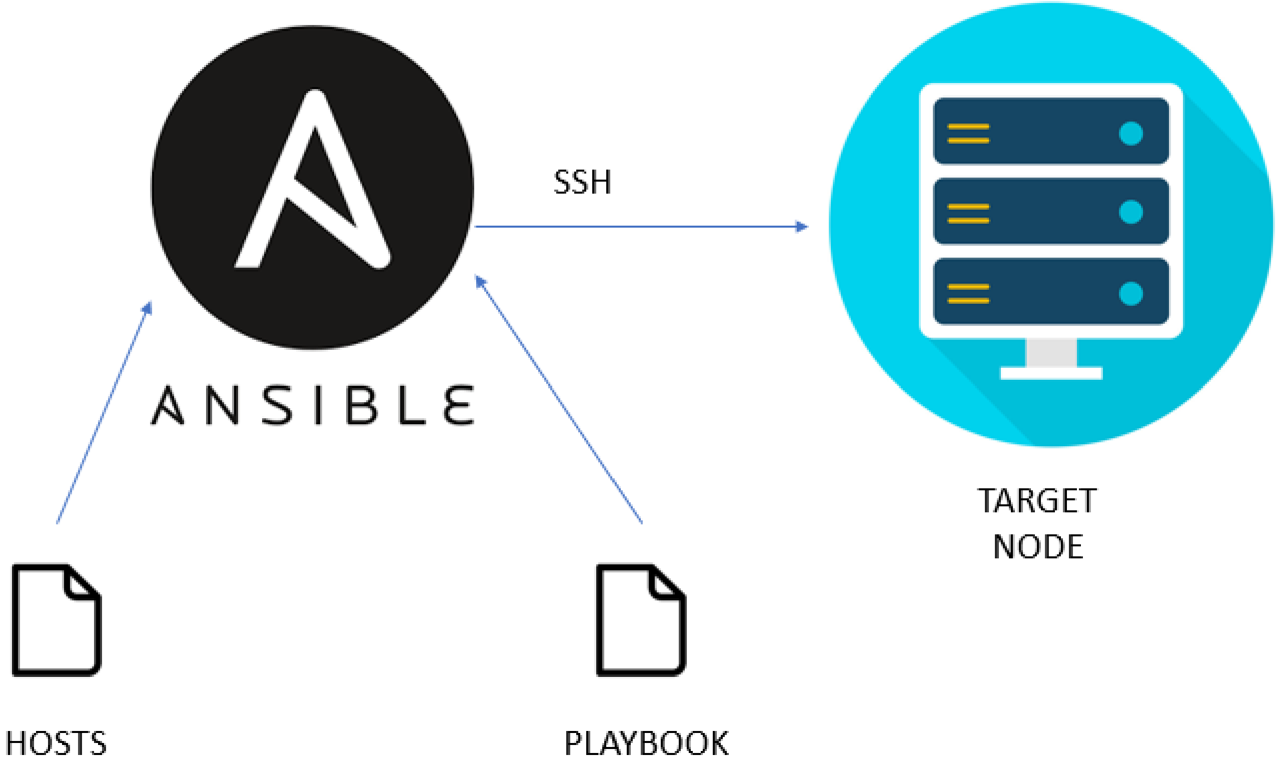
ЖИЗНЬ НА КЛАСТЕРЕ

- Linux Centos 7
- Java 8
- Scala 12
- Python 3
- Публичные IP / Внешние IP



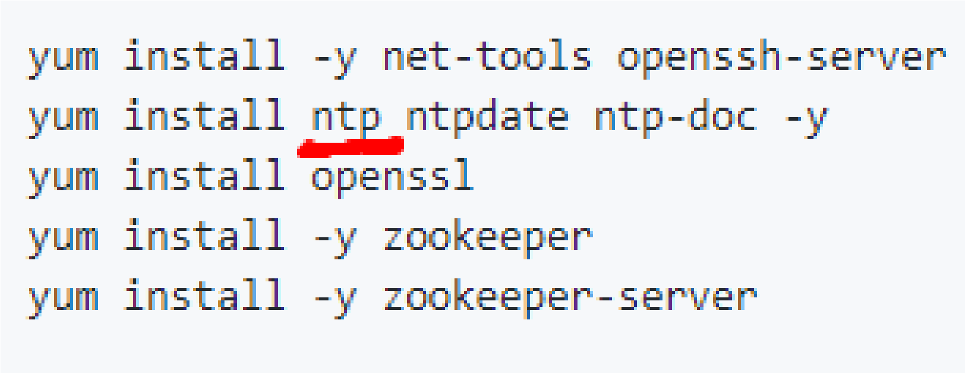
IP адрес	Имя узла	Роли
10.0.0.2	cnt-cls-m1	NameNode, ResourceManager
10.0.0.3	cnt-cls-s1	SecondaryNameNode, DataNode, NodeManager
10.0.0.4	cnt-cls-s2	DataNode
10.0.0.5	cnt-cls-s3	DataNode

ANSIBLE



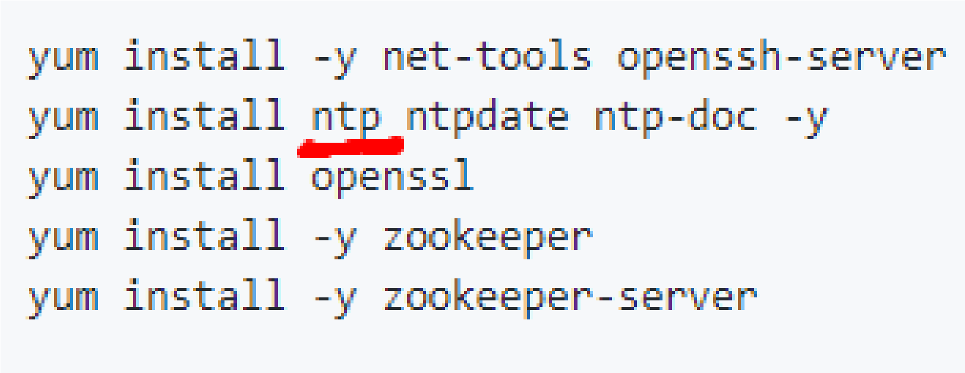
https://tproger.ru/translations/ansible-how-to-use/

КЛАСТЕР

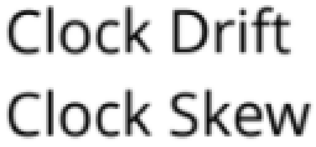


```
yum install -y net-tools openssh-server
yum install ntp ntpdate ntp-doc -y
yum install openssl
yum install -y zookeeper
yum install -y zookeeper-server
```

КЛАСТЕР

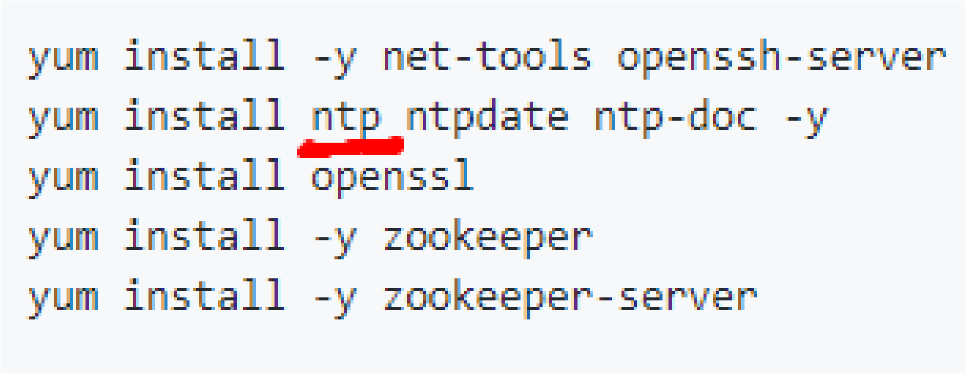


```
yum install -y net-tools openssh-server
yum install ntp ntpdate ntp-doc -y
yum install openssl
yum install -y zookeeper
yum install -y zookeeper-server
```

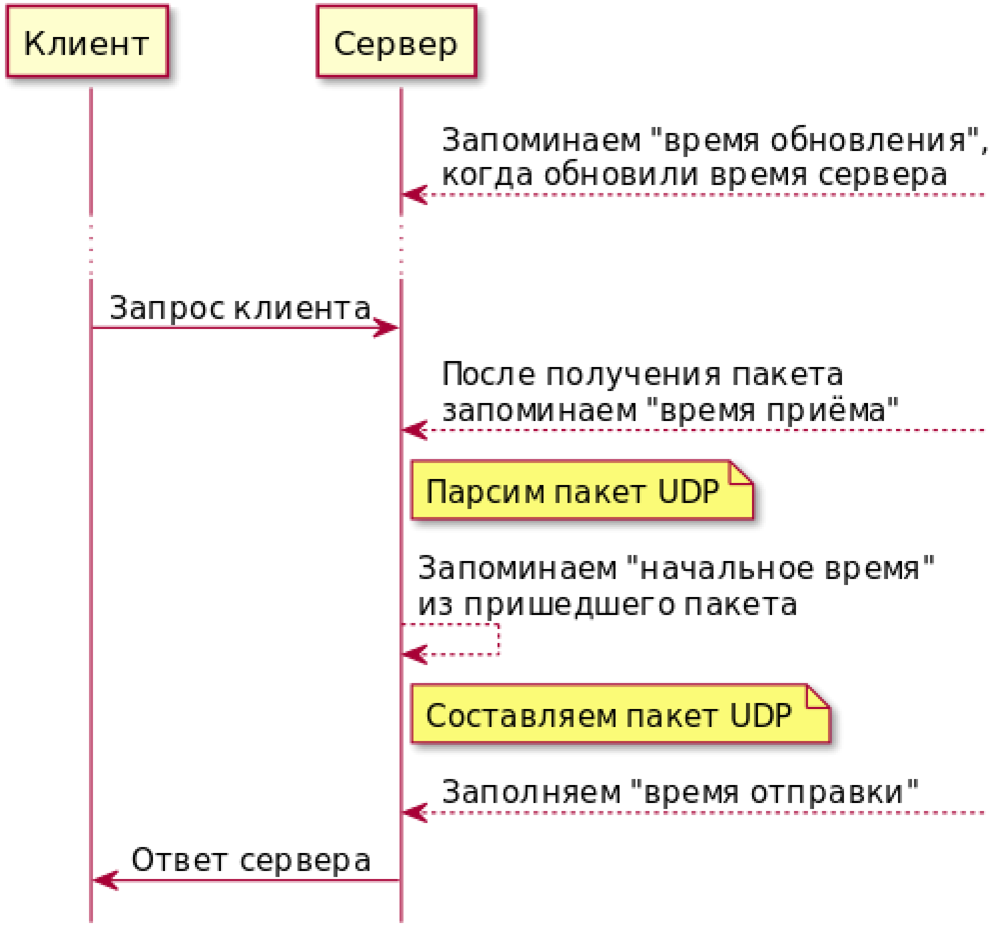


Clock Drift
Clock Skew

КЛАСТЕР



```
yum install -y net-tools openssh-server
yum install ntp ntpdate ntp-doc -y
yum install openssl
yum install -y zookeeper
yum install -y zookeeper-server
```



Клиент

Сервер

Запоминаем "время обновления",
когда обновили время сервера

Запрос клиента

После получения пакета
запоминаем "время приёма"

Парсим пакет UDP

Запоминаем "начальное время"
из пришедшего пакета

Составляем пакет UDP

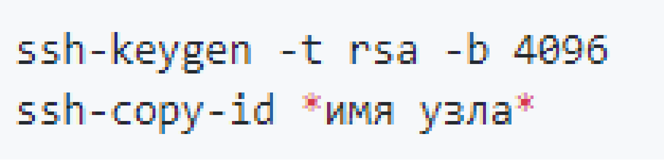
Заполняем "время отправки"

Ответ сервера

https://clck.ru/R87Kx

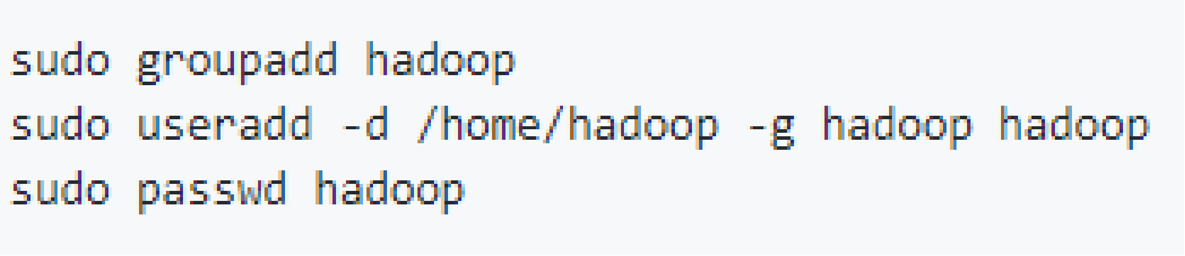
КЛАСТЕР

SSH KEY



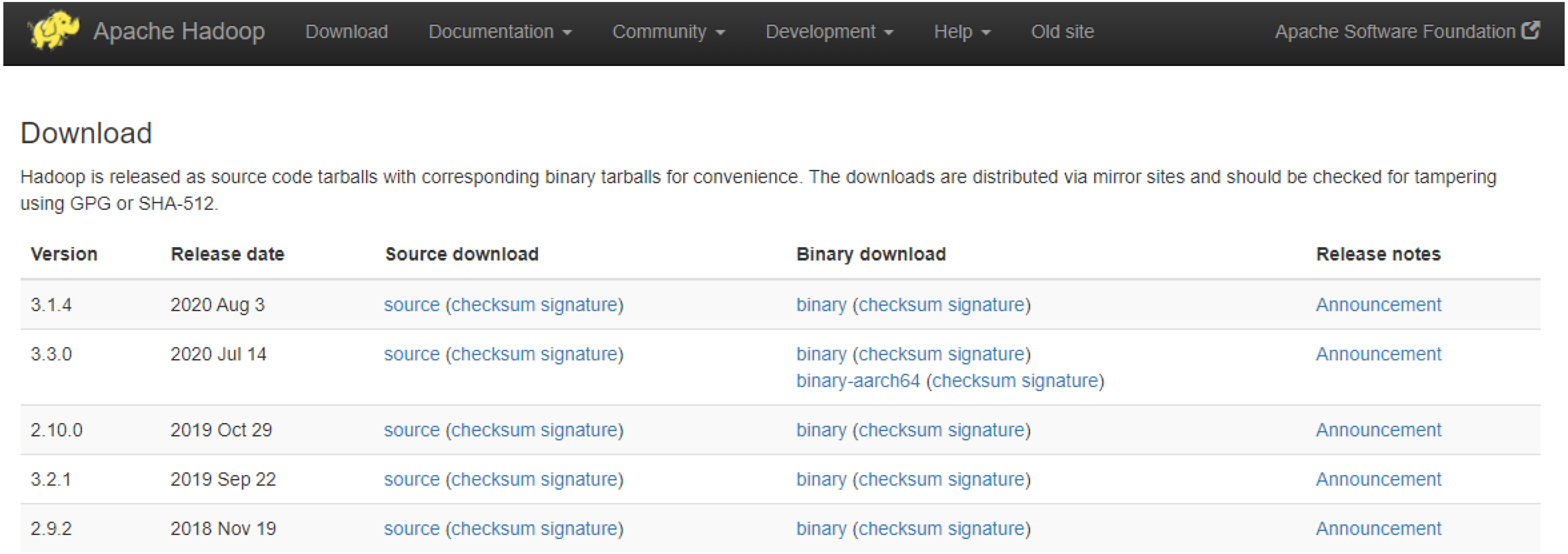
```
ssh-keygen -t rsa -b 4096
ssh-copy-id *имя узла*
```

ADD USER



```
sudo groupadd hadoop
sudo useradd -d /home/hadoop -g hadoop hadoop
sudo passwd hadoop
```

HADOOP

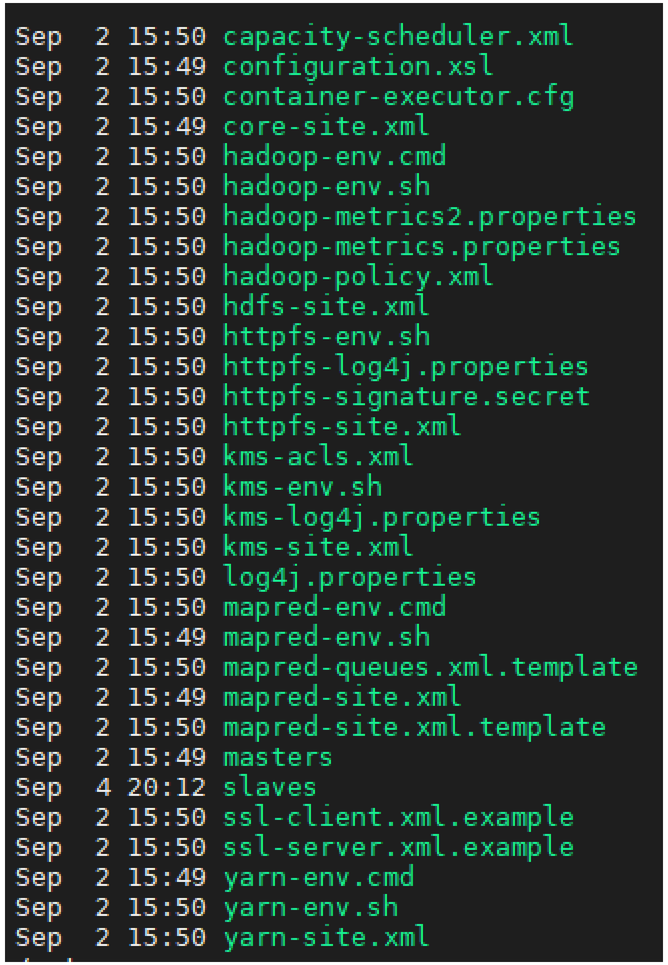


Download

Hadoop is released as source code tarballs with corresponding binary tarballs for convenience. The downloads are distributed via mirror sites and should be checked for tampering using GPG or SHA-512.

Version	Release date	Source download	Binary download	Release notes
3.1.4	2020 Aug 3	source (checksum signature)	binary (checksum signature)	Announcement
3.3.0	2020 Jul 14	source (checksum signature)	binary (checksum signature) binary-aarch64 (checksum signature)	Announcement
2.10.0	2019 Oct 29	source (checksum signature)	binary (checksum signature)	Announcement
3.2.1	2019 Sep 22	source (checksum signature)	binary (checksum signature)	Announcement
2.9.2	2018 Nov 19	source (checksum signature)	binary (checksum signature)	Announcement

HADOOP

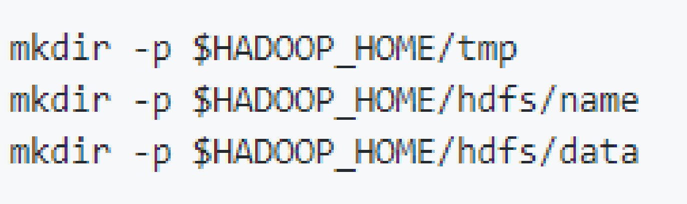


```
Sep  2 15:50 capacity-scheduler.xml
Sep  2 15:49 configuration.xsl
Sep  2 15:50 container-executor.cfg
Sep  2 15:49 core-site.xml
Sep  2 15:50 hadoop-env.cmd
Sep  2 15:50 hadoop-env.sh
Sep  2 15:50 hadoop-metrics2.properties
Sep  2 15:50 hadoop-metrics.properties
Sep  2 15:50 hadoop-policy.xml
Sep  2 15:50 hdfs-site.xml
Sep  2 15:50 httpfs-env.sh
Sep  2 15:50 httpfs-log4j.properties
Sep  2 15:50 httpfs-signature.secret
Sep  2 15:50 httpfs-site.xml
Sep  2 15:50 kms-acls.xml
Sep  2 15:50 kms-env.sh
Sep  2 15:50 kms-log4j.properties
Sep  2 15:50 kms-site.xml
Sep  2 15:50 log4j.properties
Sep  2 15:50 mapred-env.cmd
Sep  2 15:49 mapred-env.sh
Sep  2 15:50 mapred-queues.xml.template
Sep  2 15:49 mapred-site.xml
Sep  2 15:50 mapred-site.xml.template
Sep  2 15:49 masters
Sep  4 20:12 slaves
Sep  2 15:50 ssl-client.xml.example
Sep  2 15:50 ssl-server.xml.example
Sep  2 15:49 yarn-env.cmd
Sep  2 15:50 yarn-env.sh
Sep  2 15:50 yarn-site.xml
```

- core-site.xml
- hdfs-site.xml
- mapred-site.xml
- yarn-site.xml

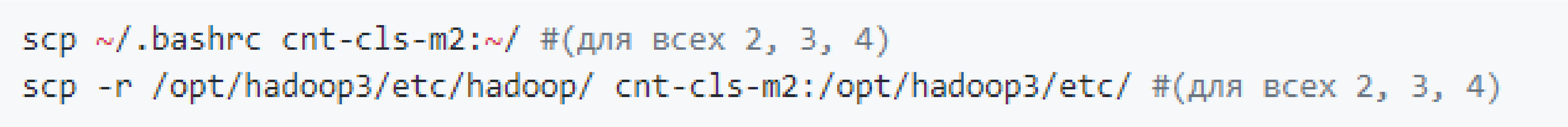
HADOOP

ADD DIRS



```
mkdir -p $HADOOP_HOME/tmp
mkdir -p $HADOOP_HOME/hdfs/name
mkdir -p $HADOOP_HOME/hdfs/data
```

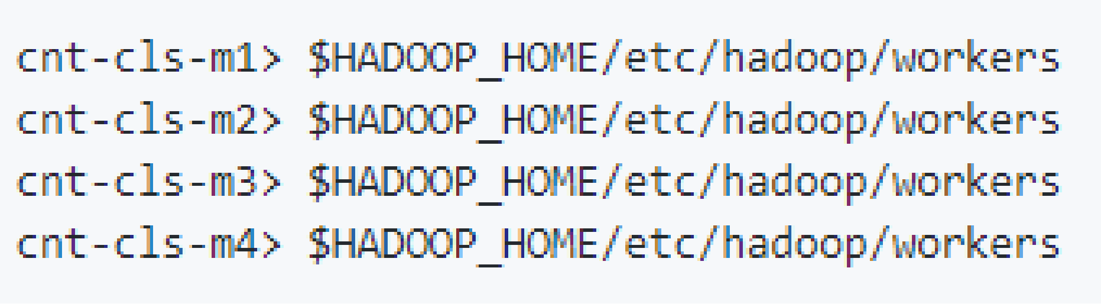
COPY SETTINGS



```
scp ~/.bashrc cnt-cls-m2:~/ #(для всех 2, 3, 4)
scp -r /opt/hadoop3/etc/hadoop/ cnt-cls-m2:/opt/hadoop3/etc/ #(для всех 2, 3, 4)
```

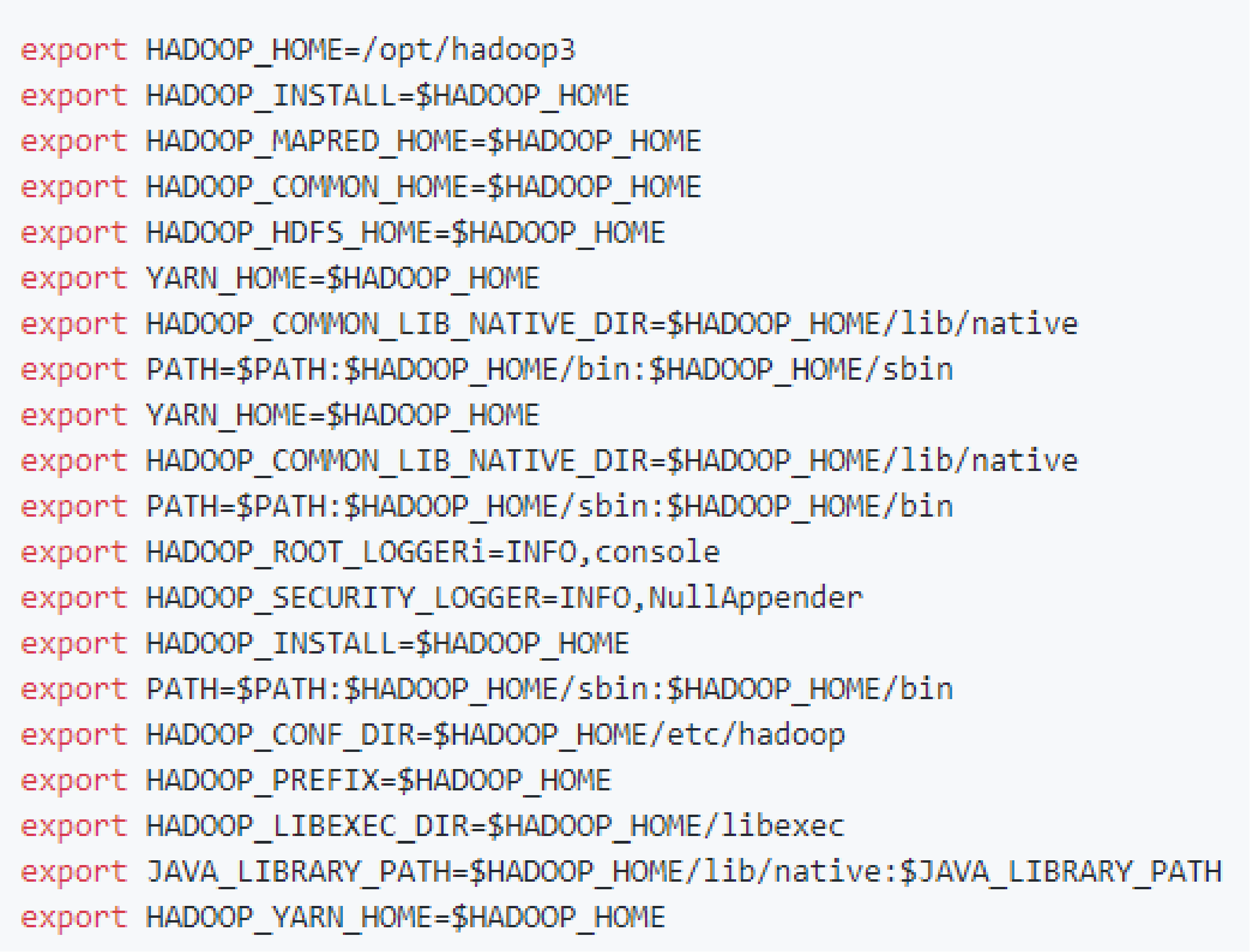
HADOOP

ADD HOSTS



```
cnt-cls-m1> $HADOOP_HOME/etc/hadoop/workers
cnt-cls-m2> $HADOOP_HOME/etc/hadoop/workers
cnt-cls-m3> $HADOOP_HOME/etc/hadoop/workers
cnt-cls-m4> $HADOOP_HOME/etc/hadoop/workers
```

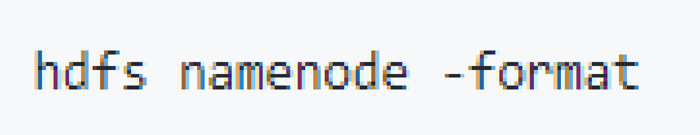
HADOOP



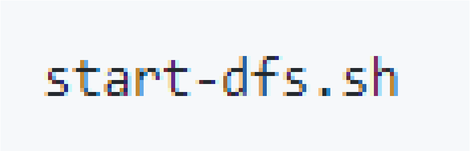
```
export HADOOP_HOME=/opt/hadoop3
export HADOOP_INSTALL=$HADOOP_HOME
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
export HADOOP_ROOT_LOGGERi=INFO,console
export HADOOP_SECURITY_LOGGER=INFO,NullAppender
export HADOOP_INSTALL=$HADOOP_HOME
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
export HADOOP_PREFIX=$HADOOP_HOME
export HADOOP_LIBEXEC_DIR=$HADOOP_HOME/libexec
export JAVA_LIBRARY_PATH=$HADOOP_HOME/lib/native:$JAVA_LIBRARY_PATH
export HADOOP_YARN_HOME=$HADOOP_HOME
```

HADOOP

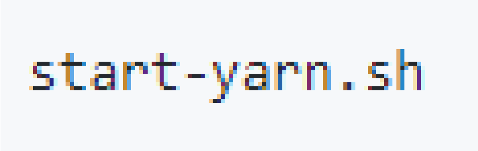
ПЕРВЫЙ ЗАПУСК



```
hdfs namenode -format
```

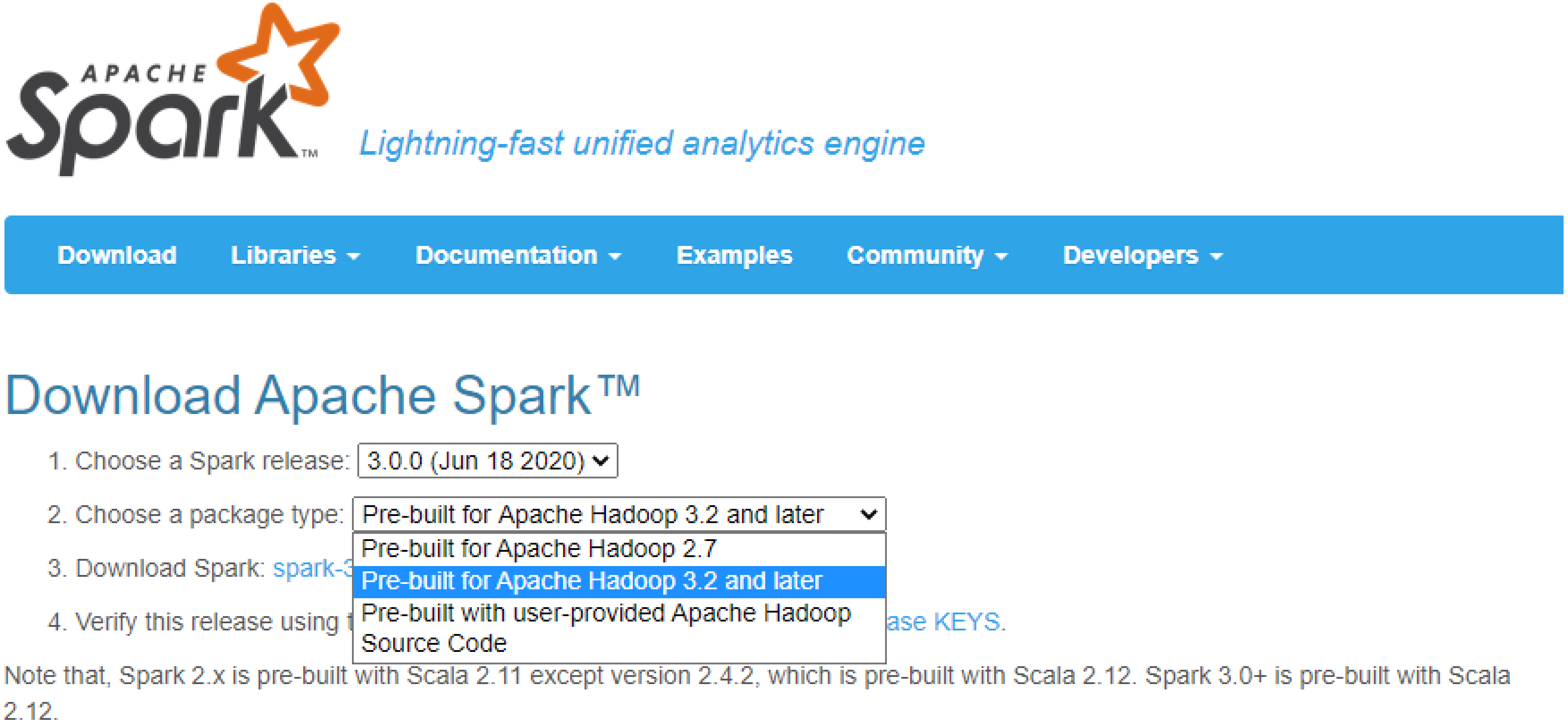


```
start-dfs.sh
```



```
start-yarn.sh
```

SPARK



Download Apache Spark™

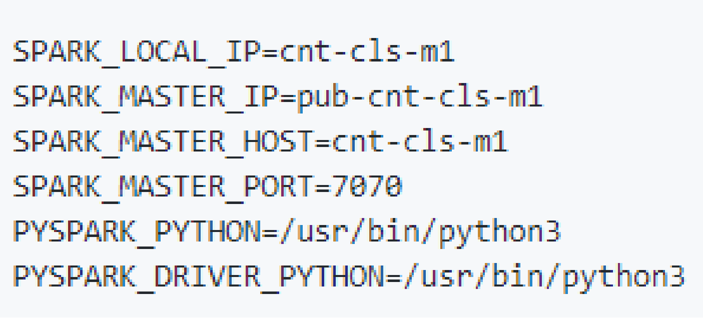
1. Choose a Spark release: 3.0.0 (Jun 18 2020)
2. Choose a package type: Pre-built for Apache Hadoop 3.2 and later
   - Pre-built for Apache Hadoop 2.7
   - Pre-built for Apache Hadoop 3.2 and later
   - Pre-built with user-provided Apache Hadoop
   - Source Code
3. Download Spark: spark-3
4. Verify this release using t ase KEYS.

Note that, Spark 2.x is pre-built with Scala 2.11 except version 2.4.2, which is pre-built with Scala 2.12. Spark 3.0+ is pre-built with Scala 2.12.

```
pip3 install pyspark
pip3 install py4j
```

SPARK

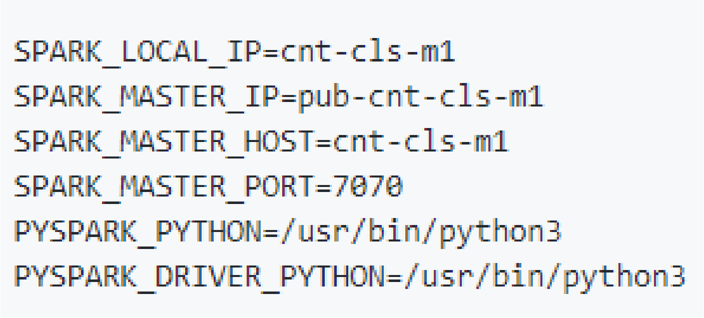
/opt/spark3/conf/spark-env.sh



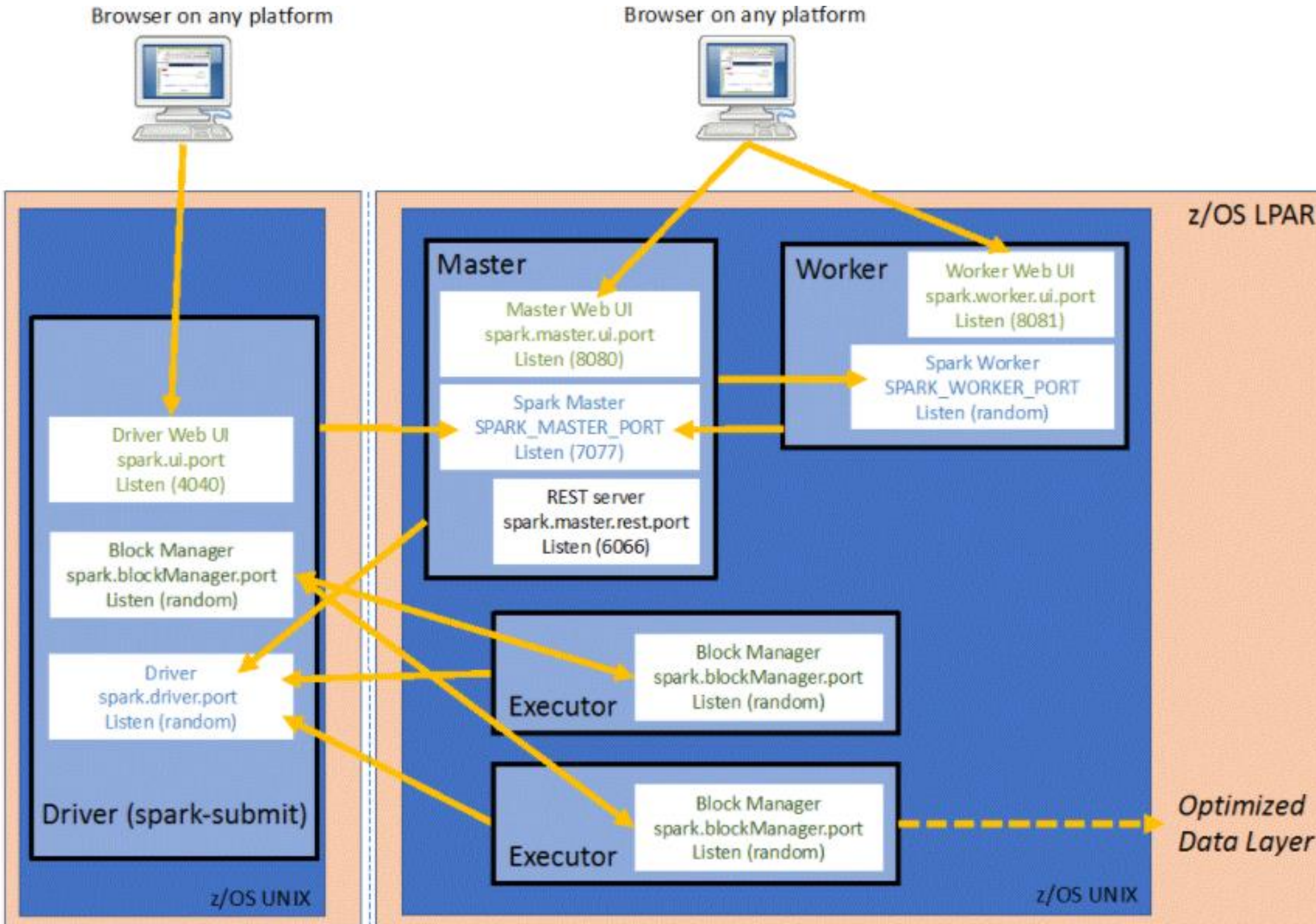
```
SPARK_LOCAL_IP=cnt-cls-m1
SPARK_MASTER_IP=pub-cnt-cls-m1
SPARK_MASTER_HOST=cnt-cls-m1
SPARK_MASTER_PORT=7070
PYSPARK_PYTHON=/usr/bin/python3
PYSPARK_DRIVER_PYTHON=/usr/bin/python3
```

SPARK

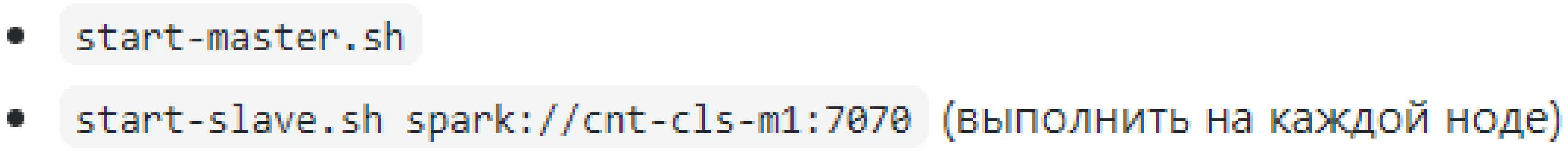
/opt/spark3/conf/spark-env.sh



```
SPARK_LOCAL_IP=cnt-cls-m1
SPARK_MASTER_IP=pub-cnt-cls-m1
SPARK_MASTER_HOST=cnt-cls-m1
SPARK_MASTER_PORT=7070
PYSPARK_PYTHON=/usr/bin/python3
PYSPARK_DRIVER_PYTHON=/usr/bin/python3
```



SPARK

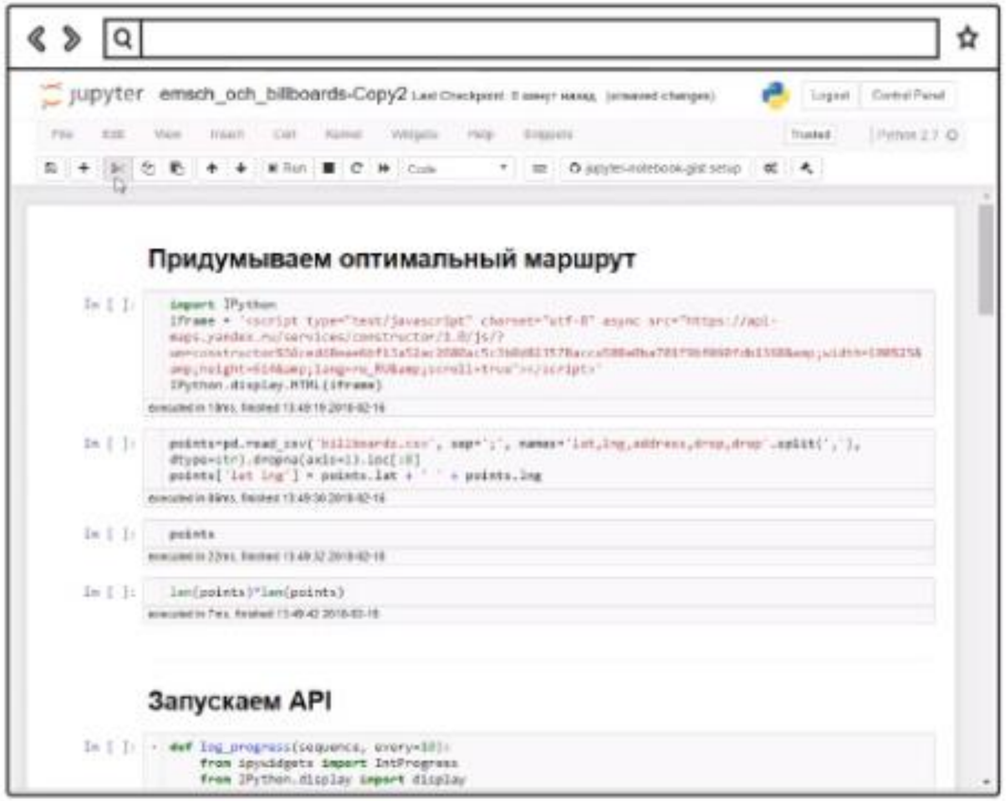


- `start-master.sh`
- `start-slave.sh spark://cnt-cls-m1:7070` (выполнить на каждой ноде)

JUPYTERHUB

```
yum install install npm nodejs-legacy
pip3 install jupyterhub
npm install -g configurable-http-proxy
```

- Классический «ноутбук»
- Различные языки программирования
- Интерактивный код, легко менять на лету
- Визуализации, произвольный output


jupyter
emsch_och_billboards-Copy2 Last Checkpoint: 8 минут назад (unsaved changes)
Logout
Control Panel
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Snippets
Trusted
Python 2.7
Run
Code
jupyter-notebook-gist setup

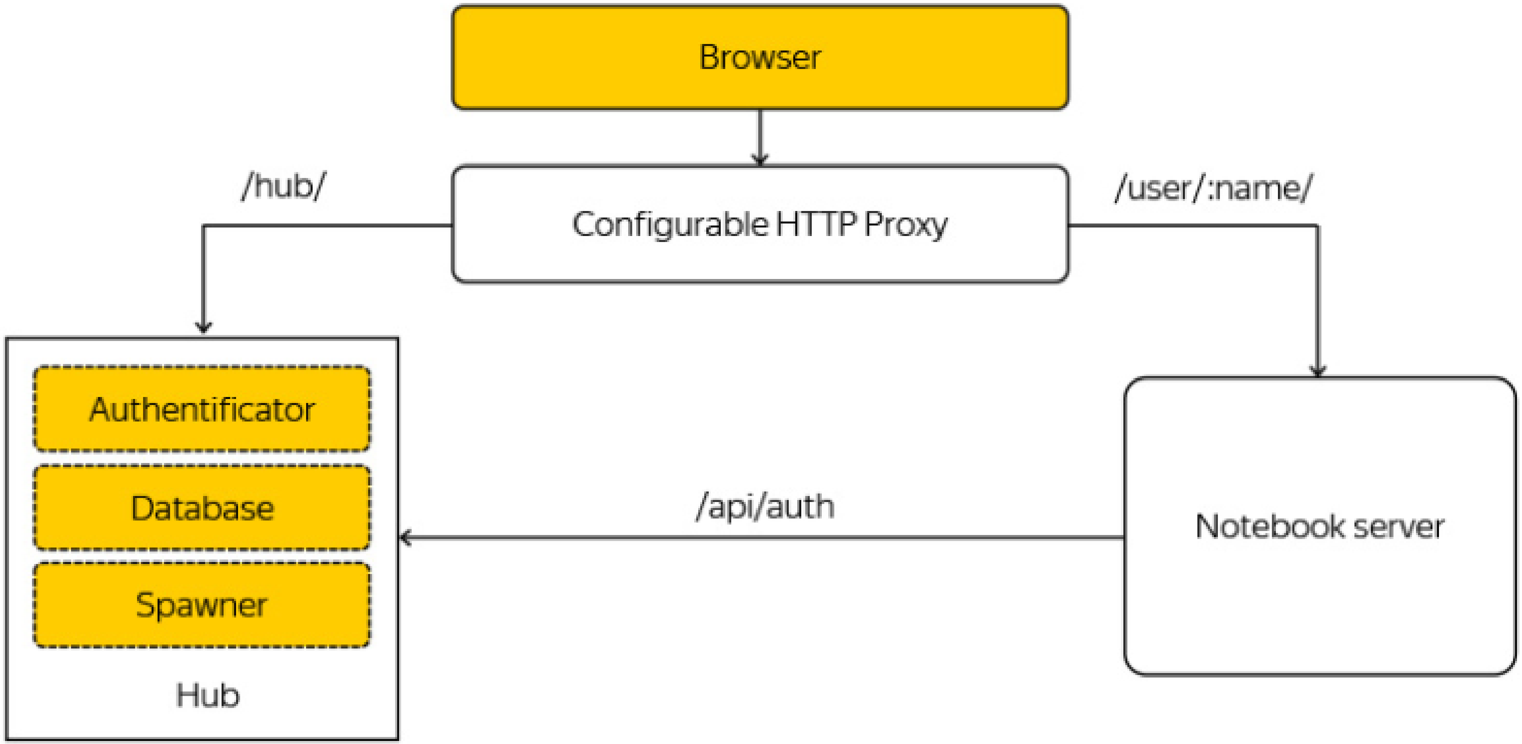
Придумываем оптимальный маршрут

```
In [ ]:   import IPython
          iframe = '<script type="text/javascript" charset="utf-8" async src="https://api-
          maps.yandex.ru/services/constructor/1.0/js/?
          um=constructor[illegible]&amp;width=100%25&
          amp;height=614&amp;lang=ru_RU&amp;scroll=true"></script>'
          IPython.display.HTML(iframe)
        executed in 10ms, finished 13:49:19 2018-02-16

In [ ]:   points=pd.read_csv('billboards.csv', sep=';', names='lat,lng,address,drop,drop'.split(','),
          dtype=str).dropna(axis=1).loc[:8]
          points['lat lng'] = points.lat + ' ' + points.lng
        executed in 8ms, finished 13:49:30 2018-02-16

In [ ]:   points
        executed in 22ms, finished 13:49:32 2018-02-16

In [ ]:   len(points)*len(points)
        executed in 7ms, finished 13:49:42 2018-02-16
```

Запускаем API

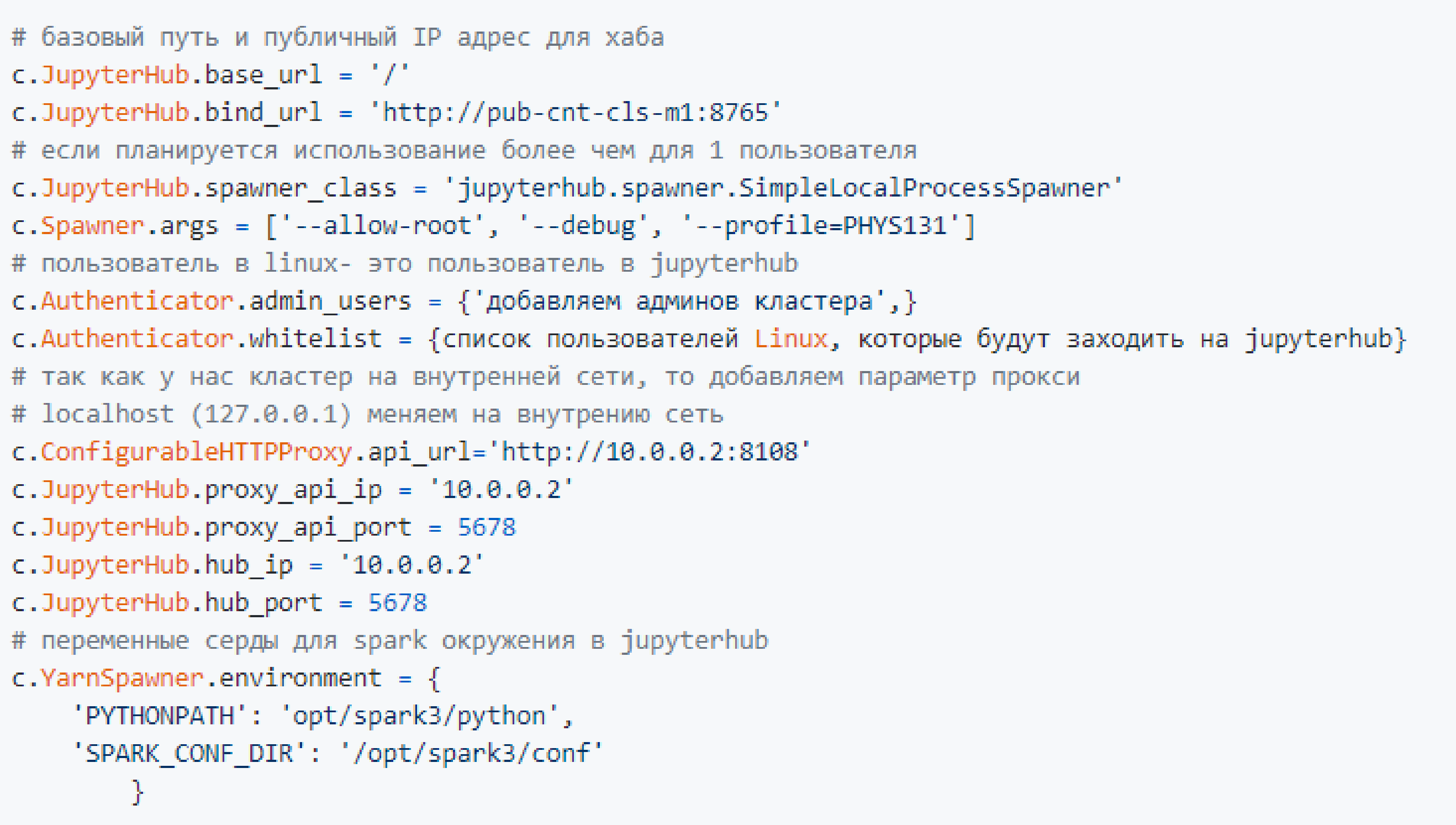
```
In [ ]: - def log_progress(sequence, every=10):
              from ipywidgets import IntProgress
              from IPython.display import display
```

JUPYTERHUB



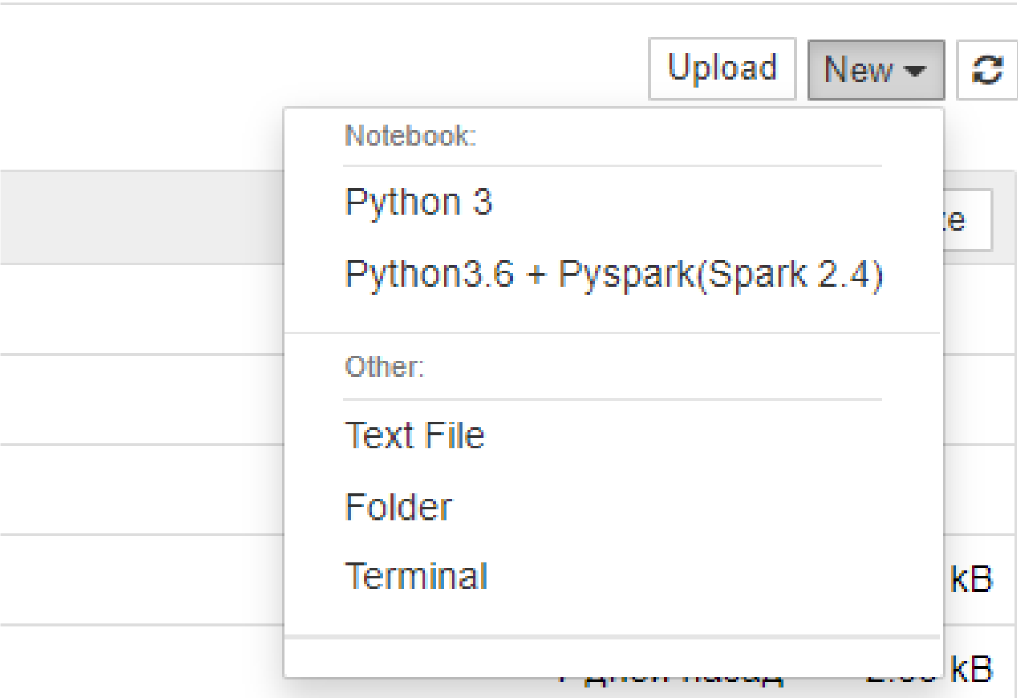
Browser
/hub/
Configurable HTTP Proxy
/user/:name/
Authentificator
Database
Spawner
Hub
/api/auth
Notebook server

JUPYTERHUB



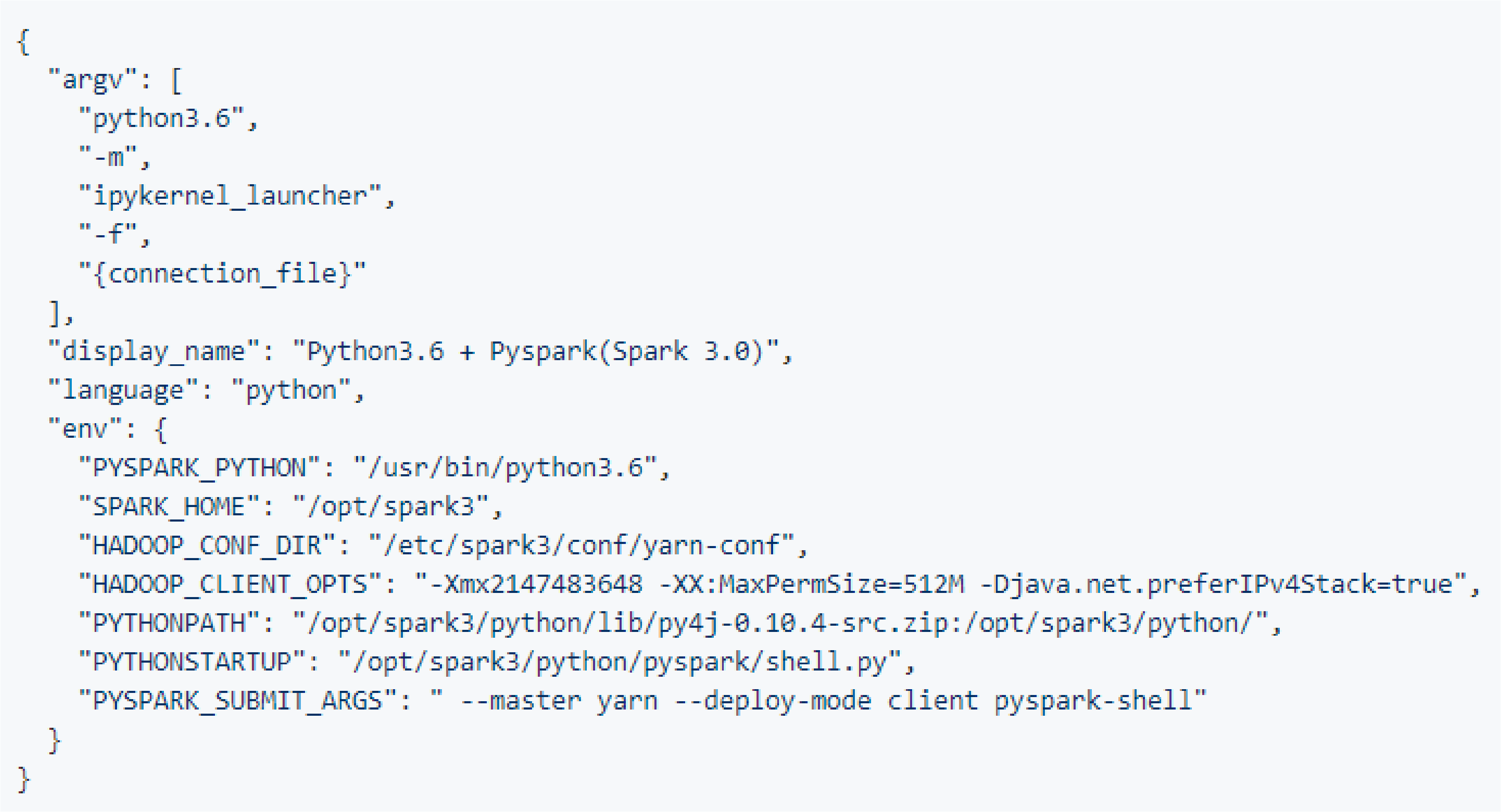
```
# базовый путь и публичный IP адрес для хаба
c.JupyterHub.base_url = '/'
c.JupyterHub.bind_url = 'http://pub-cnt-cls-m1:8765'
# если планируется использование более чем для 1 пользователя
c.JupyterHub.spawner_class = 'jupyterhub.spawner.SimpleLocalProcessSpawner'
c.Spawner.args = ['--allow-root', '--debug', '--profile=PHYS131']
# пользователь в linux- это пользователь в jupyterhub
c.Authenticator.admin_users = {'добавляем админов кластера',}
c.Authenticator.whitelist = {список пользователей Linux, которые будут заходить на jupyterhub}
# так как у нас кластер на внутренней сети, то добавляем параметр прокси
# localhost (127.0.0.1) меняем на внутрению сеть
c.ConfigurableHTTPProxy.api_url='http://10.0.0.2:8108'
c.JupyterHub.proxy_api_ip = '10.0.0.2'
c.JupyterHub.proxy_api_port = 5678
c.JupyterHub.hub_ip = '10.0.0.2'
c.JupyterHub.hub_port = 5678
# переменные серды для spark окружения в jupyterhub
c.YarnSpawner.environment = {
    'PYTHONPATH': 'opt/spark3/python',
    'SPARK_CONF_DIR': '/opt/spark3/conf'
        }
```

JUPYTERHUB KERNEL



JUPYTERHUB KERNEL

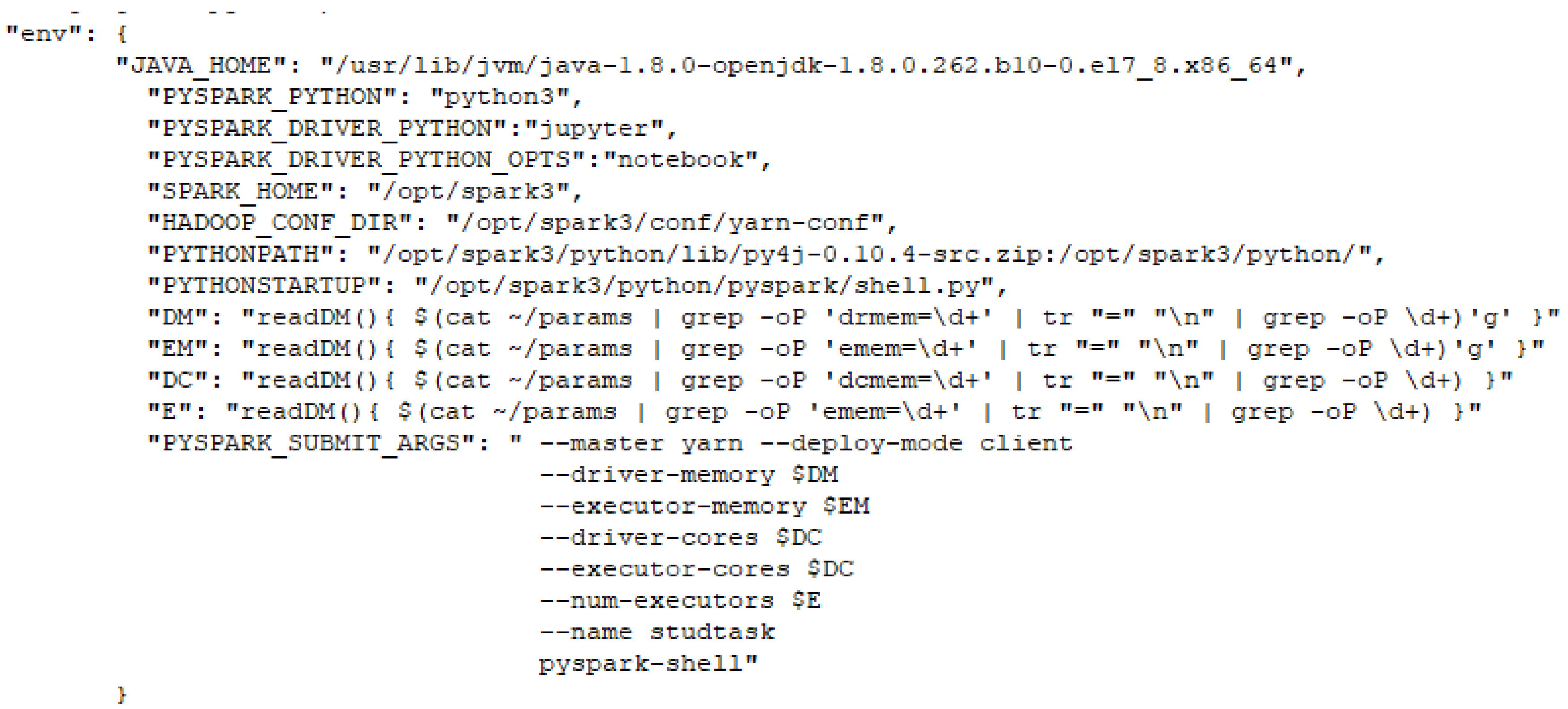
/usr/share/jupyter/kernels/



```
{
  "argv": [
    "python3.6",
    "-m",
    "ipykernel_launcher",
    "-f",
    "{connection_file}"
  ],
  "display_name": "Python3.6 + Pyspark(Spark 3.0)",
  "language": "python",
  "env": {
    "PYSPARK_PYTHON": "/usr/bin/python3.6",
    "SPARK_HOME": "/opt/spark3",
    "HADOOP_CONF_DIR": "/etc/spark3/conf/yarn-conf",
    "HADOOP_CLIENT_OPTS": "-Xmx2147483648 -XX:MaxPermSize=512M -Djava.net.preferIPv4Stack=true",
    "PYTHONPATH": "/opt/spark3/python/lib/py4j-0.10.4-src.zip:/opt/spark3/python/",
    "PYTHONSTARTUP": "/opt/spark3/python/pyspark/shell.py",
    "PYSPARK_SUBMIT_ARGS": " --master yarn --deploy-mode client pyspark-shell"
  }
}
```

JUPYTERHUB KERNEL DYNAMIC

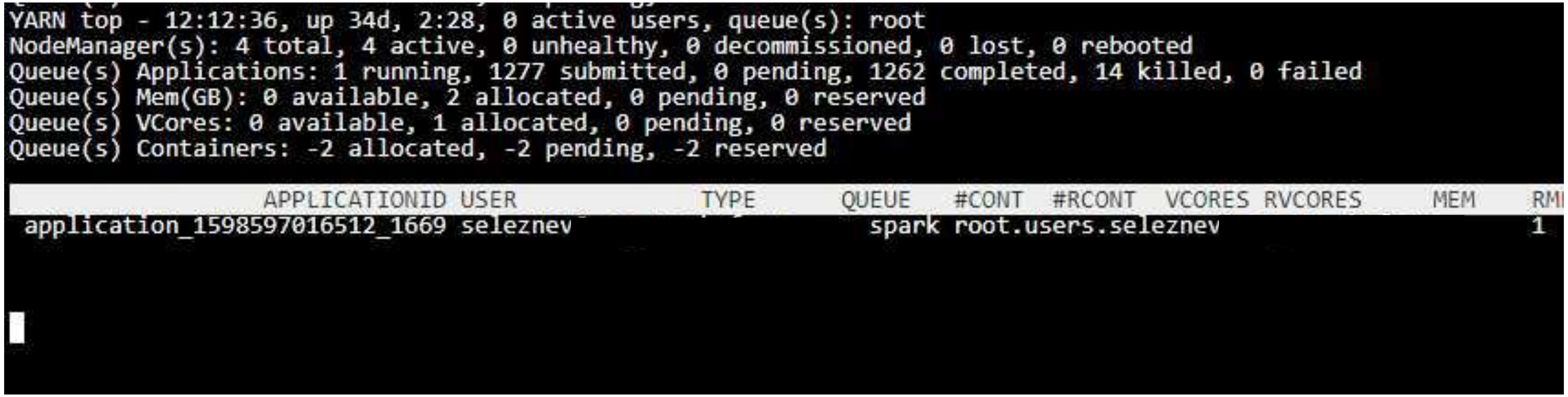
/usr/share/jupyter/kernels/dynamicone



```
"env": {
       "JAVA_HOME": "/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.262.b10-0.el7_8.x86_64",
         "PYSPARK_PYTHON": "python3",
         "PYSPARK_DRIVER_PYTHON":"jupyter",
         "PYSPARK_DRIVER_PYTHON_OPTS":"notebook",
         "SPARK_HOME": "/opt/spark3",
         "HADOOP_CONF_DIR": "/opt/spark3/conf/yarn-conf",
         "PYTHONPATH": "/opt/spark3/python/lib/py4j-0.10.4-src.zip:/opt/spark3/python/",
         "PYTHONSTARTUP": "/opt/spark3/python/pyspark/shell.py",
         "DM": "readDM(){ $(cat ~/params | grep -oP 'drmem=\d+' | tr "=" "\n" | grep -oP \d+)'g' }"
         "EM": "readDM(){ $(cat ~/params | grep -oP 'emem=\d+' | tr "=" "\n" | grep -oP \d+)'g' }"
         "DC": "readDM(){ $(cat ~/params | grep -oP 'dcmem=\d+' | tr "=" "\n" | grep -oP \d+) }"
         "E": "readDM(){ $(cat ~/params | grep -oP 'emem=\d+' | tr "=" "\n" | grep -oP \d+) }"
         "PYSPARK_SUBMIT_ARGS": " --master yarn --deploy-mode client
                                  --driver-memory $DM
                                  --executor-memory $EM
                                  --driver-cores $DC
                                  --executor-cores $DC
                                  --num-executors $E
                                  --name studtask
                                  pyspark-shell"
       }
```

JUPYTERHUB KERNEL DYNAMIC

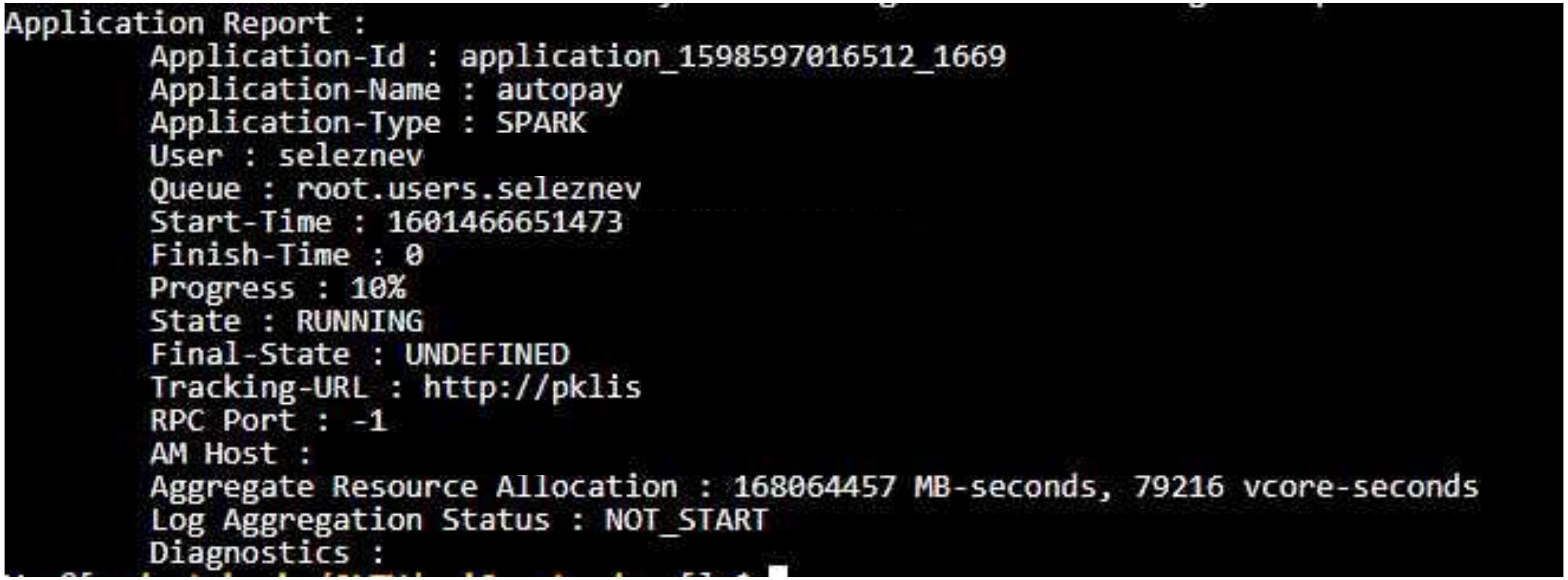
yarn top



```
YARN top - 12:12:36, up 34d, 2:28, 0 active users, queue(s): root
NodeManager(s): 4 total, 4 active, 0 unhealthy, 0 decommissioned, 0 lost, 0 rebooted
Queue(s) Applications: 1 running, 1277 submitted, 0 pending, 1262 completed, 14 killed, 0 failed
Queue(s) Mem(GB): 0 available, 2 allocated, 0 pending, 0 reserved
Queue(s) VCores: 0 available, 1 allocated, 0 pending, 0 reserved
Queue(s) Containers: -2 allocated, -2 pending, -2 reserved

                  APPLICATIONID USER                 TYPE        QUEUE    #CONT   #RCONT   VCORES RVCORES     MEM     RM
 application_1598597016512_1669 seleznev                           spark root.users.seleznev                            1
```

JUPYTERHUB KERNEL DYNAMIC

yarn application –status app_id



```
Application Report :
        Application-Id : application_1598597016512_1669
        Application-Name : autopay
        Application-Type : SPARK
        User : seleznev
        Queue : root.users.seleznev
        Start-Time : 1601466651473
        Finish-Time : 0
        Progress : 10%
        State : RUNNING
        Final-State : UNDEFINED
        Tracking-URL : http://pklis
        RPC Port : -1
        AM Host :
        Aggregate Resource Allocation : 168064457 MB-seconds, 79216 vcore-seconds
        Log Aggregation Status : NOT_START
        Diagnostics :
```

JUPYTERHUB KERNEL DYNAMIC

ResourceManager REST API

(curl –v –X GET –H “Content-Type: application/json”)

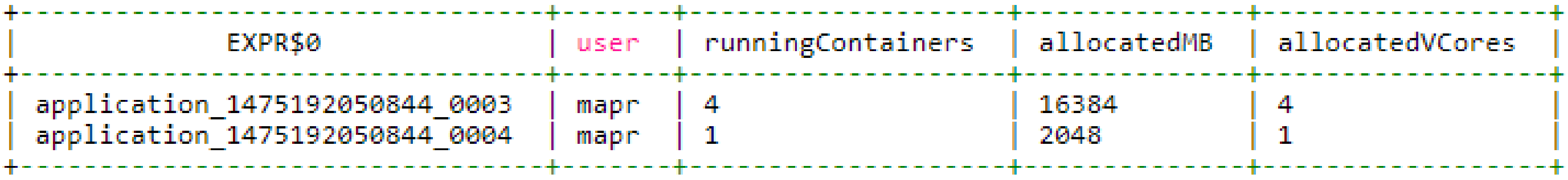
+

Apache DRILL

```
with tmp as
(
select flatten(t.apps.app) as col
from dfs.tmp.`restapi/data.json` t
)

select tmp.col.id
        , tmp.col.`user` as `user`
        , tmp.col.runningContainers as `runningContainers`
        , tmp.col.allocatedMB as `allocatedMB`
        , tmp.col.allocatedVCores as `allocatedVCores`
from tmp
where tmp.col.state='RUNNING'
order by tmp.col.runningContainers desc;
```

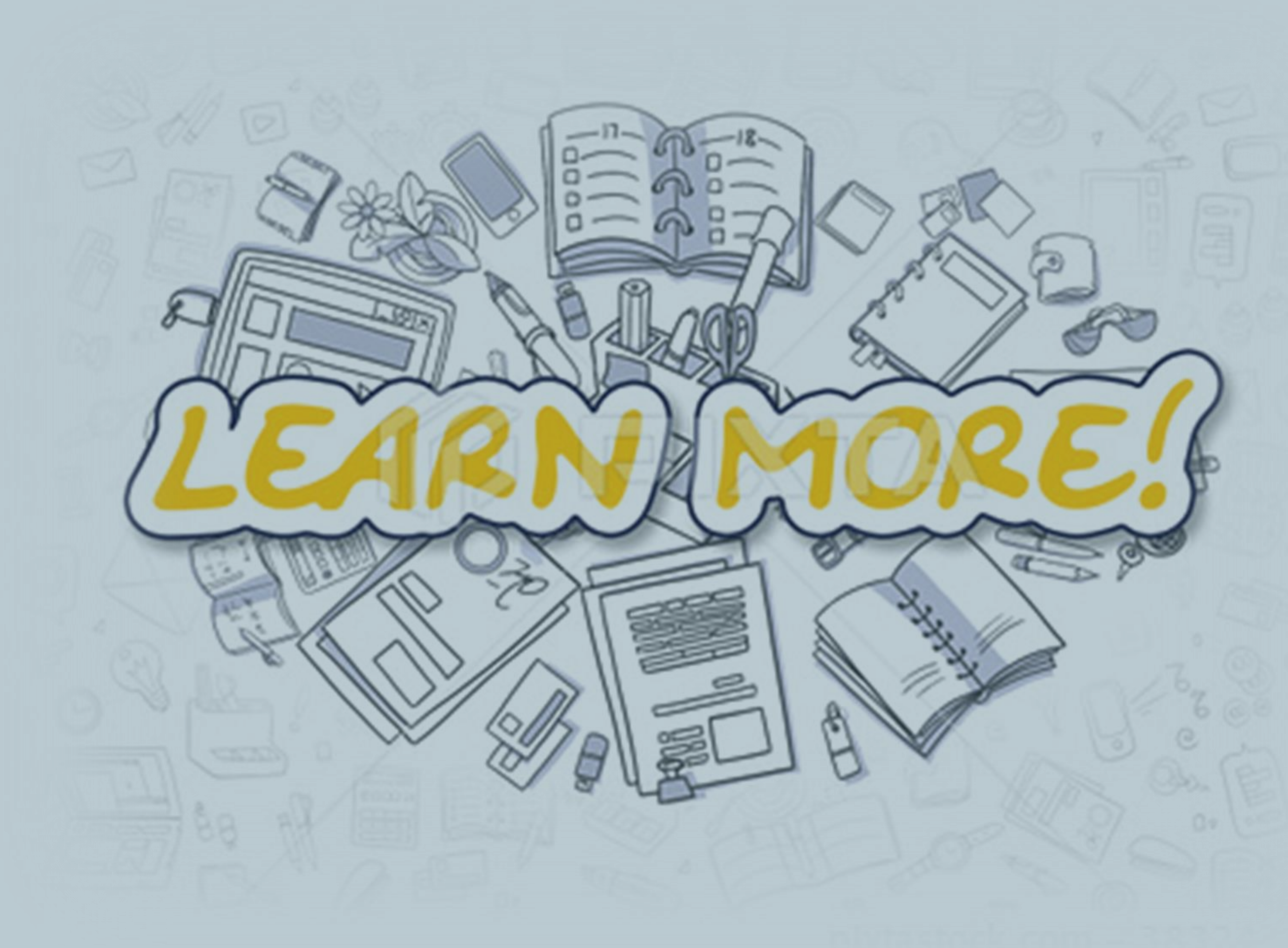
JUPYTERHUB KERNEL DYNAMIC

```
with tmp as
(
select  flatten(t.apps.app)  as col
from  dfs.tmp.`restapi/data.json`  t
)

select  tmp.col.id
          , tmp.col.`user`  as `user`
          , tmp.col.runningContainers  as  `runningContainers`
          , tmp.col.allocatedMB  as  `allocatedMB`
          , tmp.col.allocatedVCores  as  `allocatedVCores`
from  tmp
where tmp.col.state='RUNNING'
order  by tmp.col.runningContainers  desc;
```



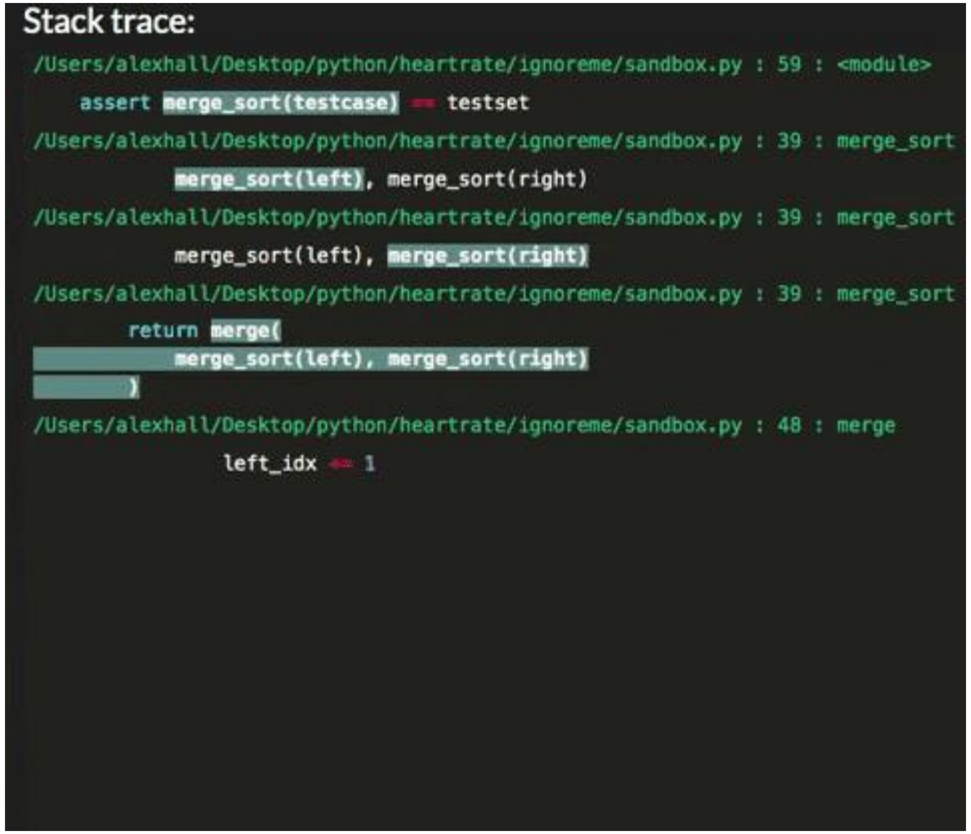
```
+-----------------------------------+--------+---------------------+---------------+------------------+
|              EXPR$0               |  user  | runningContainers   | allocatedMB   | allocatedVCores  |
+-----------------------------------+--------+---------------------+---------------+------------------+
| application_1475192050844_0003    | mapr   | 4                   | 16384         | 4                |
| application_1475192050844_0004    | mapr   | 1                   | 2048          | 1                |
+-----------------------------------+--------+---------------------+---------------+------------------+
```

SPARK PERFORMANCE



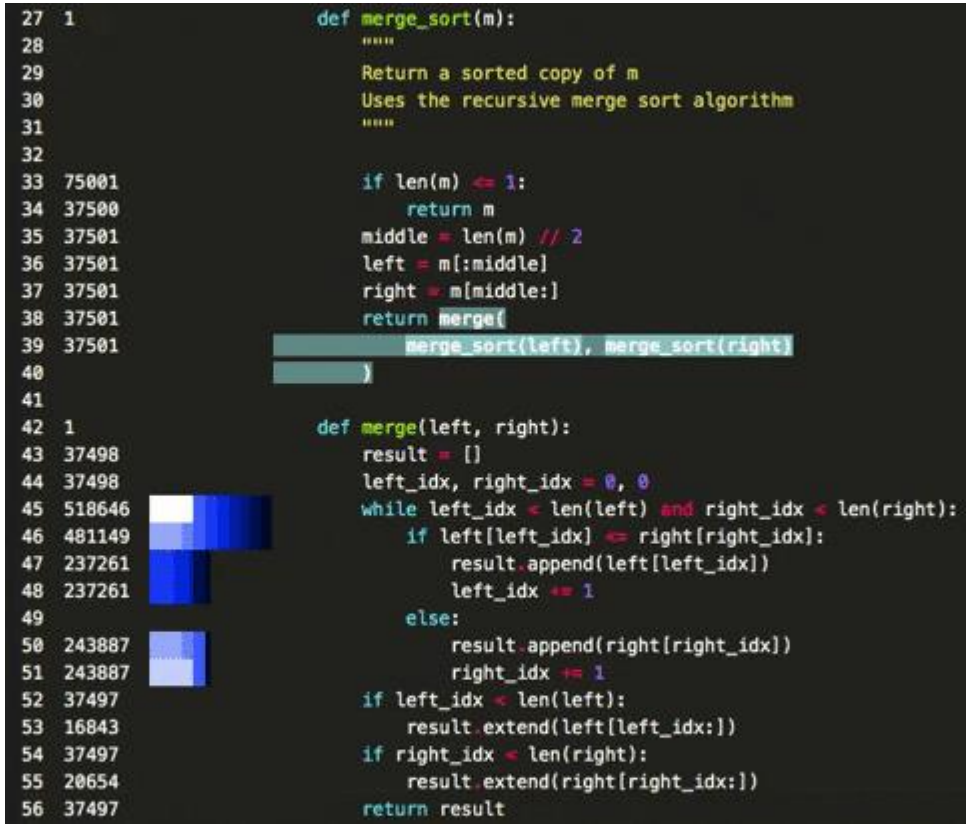
UNDERSTANDABLE ENV

STACKTRACE / DEBUG



```
Stack trace:
 /Users/alexhall/Desktop/python/heartrate/ignoreme/sandbox.py : 59 : <module>
     assert merge_sort(testcase) == testset
 /Users/alexhall/Desktop/python/heartrate/ignoreme/sandbox.py : 39 : merge_sort
             merge_sort(left), merge_sort(right)
 /Users/alexhall/Desktop/python/heartrate/ignoreme/sandbox.py : 39 : merge_sort
             merge_sort(left), merge_sort(right)
 /Users/alexhall/Desktop/python/heartrate/ignoreme/sandbox.py : 39 : merge_sort
         return merge(
             merge_sort(left), merge_sort(right)
         )
 /Users/alexhall/Desktop/python/heartrate/ignoreme/sandbox.py : 48 : merge
                 left_idx += 1
```

ВЫПОЛНЕНИЕ КОДА

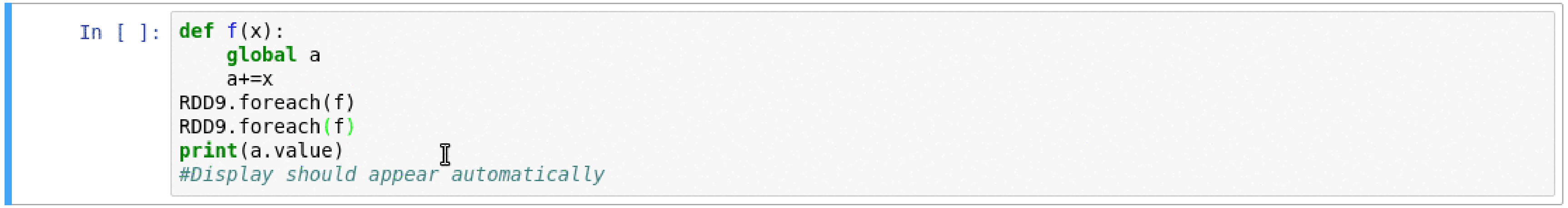


```
27  1             def merge_sort(m):
28                    """
29                    Return a sorted copy of m
30                    Uses the recursive merge sort algorithm
31                    """
32
33  75001             if len(m) <= 1:
34  37500                 return m
35  37501             middle = len(m) // 2
36  37501             left = m[:middle]
37  37501             right = m[middle:]
38  37501             return merge(
39  37501                 merge_sort(left), merge_sort(right)
40                    )
41
42  1             def merge(left, right):
43  37498             result = []
44  37498             left_idx, right_idx = 0, 0
45  518646            while left_idx < len(left) and right_idx < len(right):
46  481149                if left[left_idx] <= right[right_idx]:
47  237261                    result.append(left[left_idx])
48  237261                    left_idx += 1
49                        else:
50  243887                    result.append(right[right_idx])
51  243887                    right_idx += 1
52  37497             if left_idx < len(left):
53  16843                 result.extend(left[left_idx:])
54  37497             if right_idx < len(right):
55  20654                 result.extend(right[right_idx:])
56  37497             return result
```

https://github.com/alexmojaki/heartrate

UNDERSTANDABLE ENV

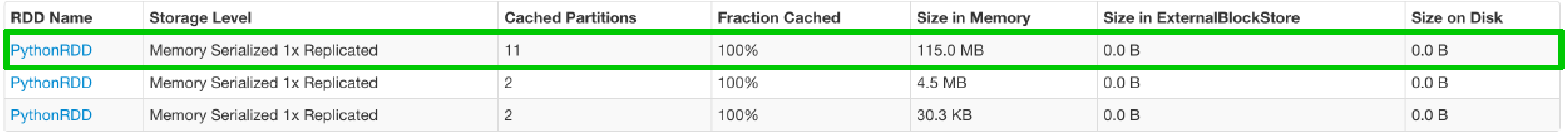
ART OF SPARK



```
def f(x):
    global a
    a+=x
RDD9.foreach(f)
RDD9.foreach(f)
print(a.value)
#Display should appear automatically
```

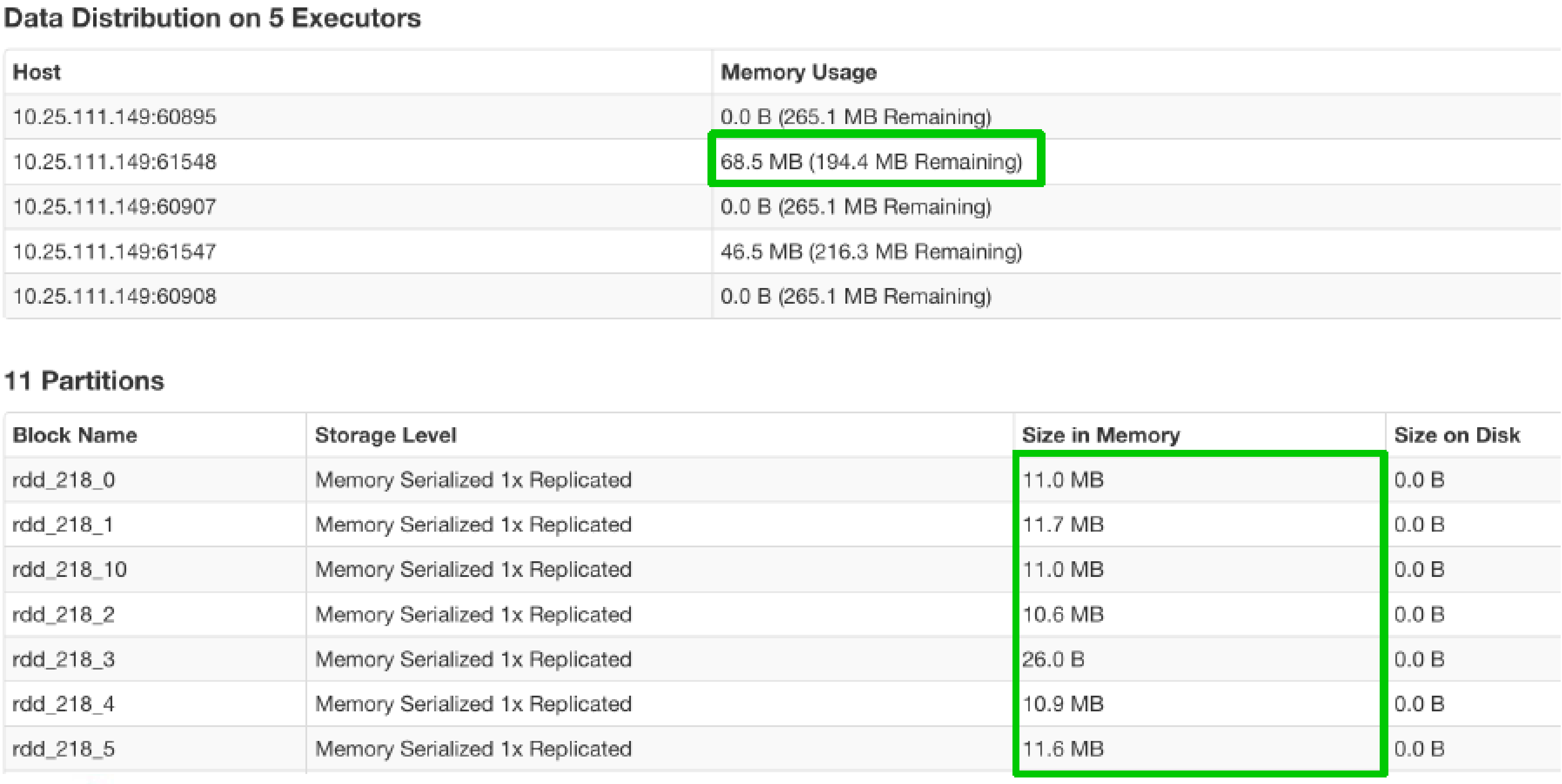
https://github.com/krishnan-r/sparkmonitor

КЛАСТЕР – ЭТО ПОСТОЯННАЯ ПЕРЕДАЧА ДАННЫХ



RDD Name	Storage Level	Cached Partitions	Fraction Cached	Size in Memory	Size in ExternalBlockStore	Size on Disk
PythonRDD	Memory Serialized 1x Replicated	11	100%	115.0 MB	0.0 B	0.0 B
PythonRDD	Memory Serialized 1x Replicated	2	100%	4.5 MB	0.0 B	0.0 B
PythonRDD	Memory Serialized 1x Replicated	2	100%	30.3 KB	0.0 B	0.0 B

КЛАСТЕР – ЭТО ПОСТОЯННАЯ ПЕРЕДАЧА ДАННЫХ



Data Distribution on 5 Executors

Host	Memory Usage
10.25.111.149:60895	0.0 B (265.1 MB Remaining)
10.25.111.149:61548	68.5 MB (194.4 MB Remaining)
10.25.111.149:60907	0.0 B (265.1 MB Remaining)
10.25.111.149:61547	46.5 MB (216.3 MB Remaining)
10.25.111.149:60908	0.0 B (265.1 MB Remaining)

11 Partitions

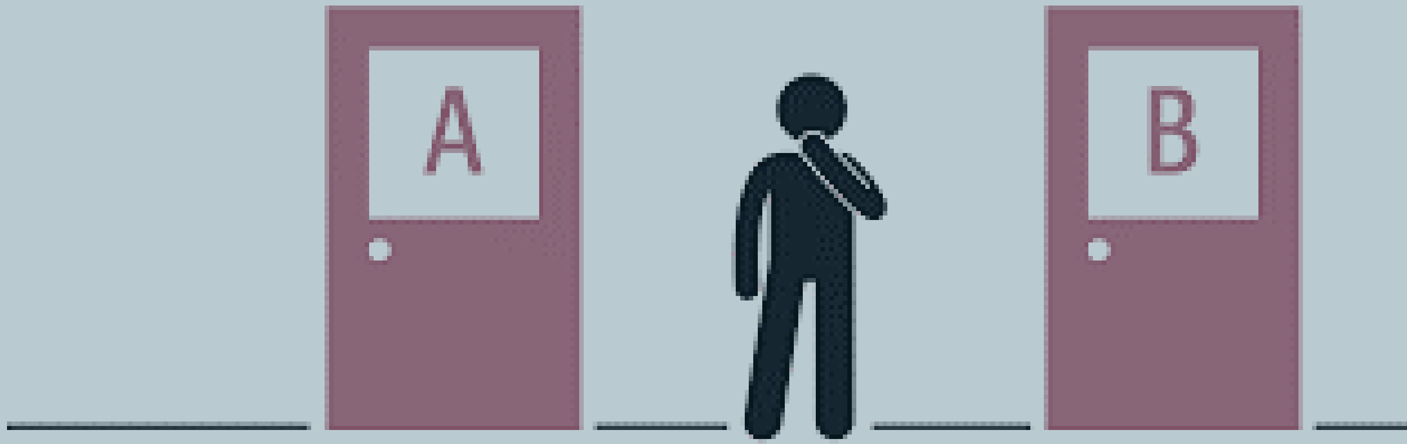
Block Name	Storage Level	Size in Memory	Size on Disk
rdd_218_0	Memory Serialized 1x Replicated	11.0 MB	0.0 B
rdd_218_1	Memory Serialized 1x Replicated	11.7 MB	0.0 B
rdd_218_10	Memory Serialized 1x Replicated	11.0 MB	0.0 B
rdd_218_2	Memory Serialized 1x Replicated	10.6 MB	0.0 B
rdd_218_3	Memory Serialized 1x Replicated	26.0 B	0.0 B
rdd_218_4	Memory Serialized 1x Replicated	10.9 MB	0.0 B
rdd_218_5	Memory Serialized 1x Replicated	11.6 MB	0.0 B

КЛАСТЕР – ЭТО ПОСТОЯННАЯ ПЕРЕДАЧА ДАННЫХ

Serialization/Deserialization

```
conf.set(‘spark.serializer”,
            ‘org.apache.spark.serializer.KryoSerializer’)
```

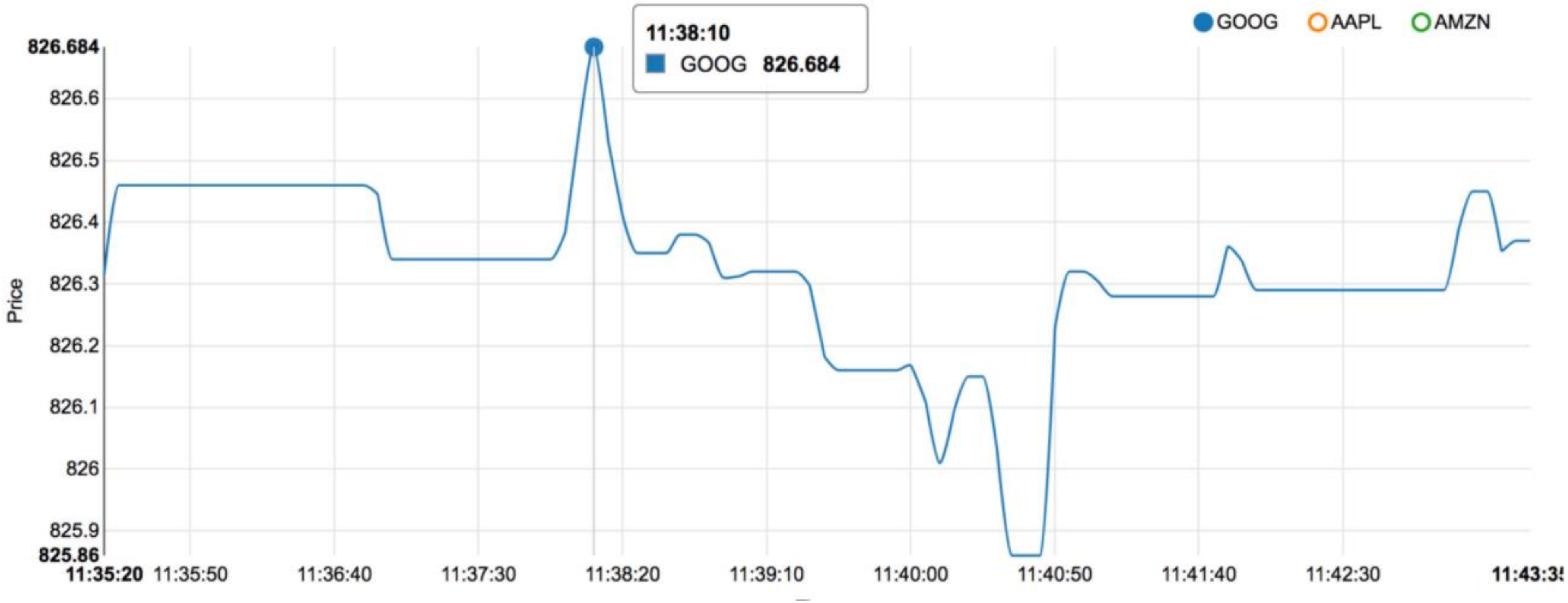
SPARK TEST



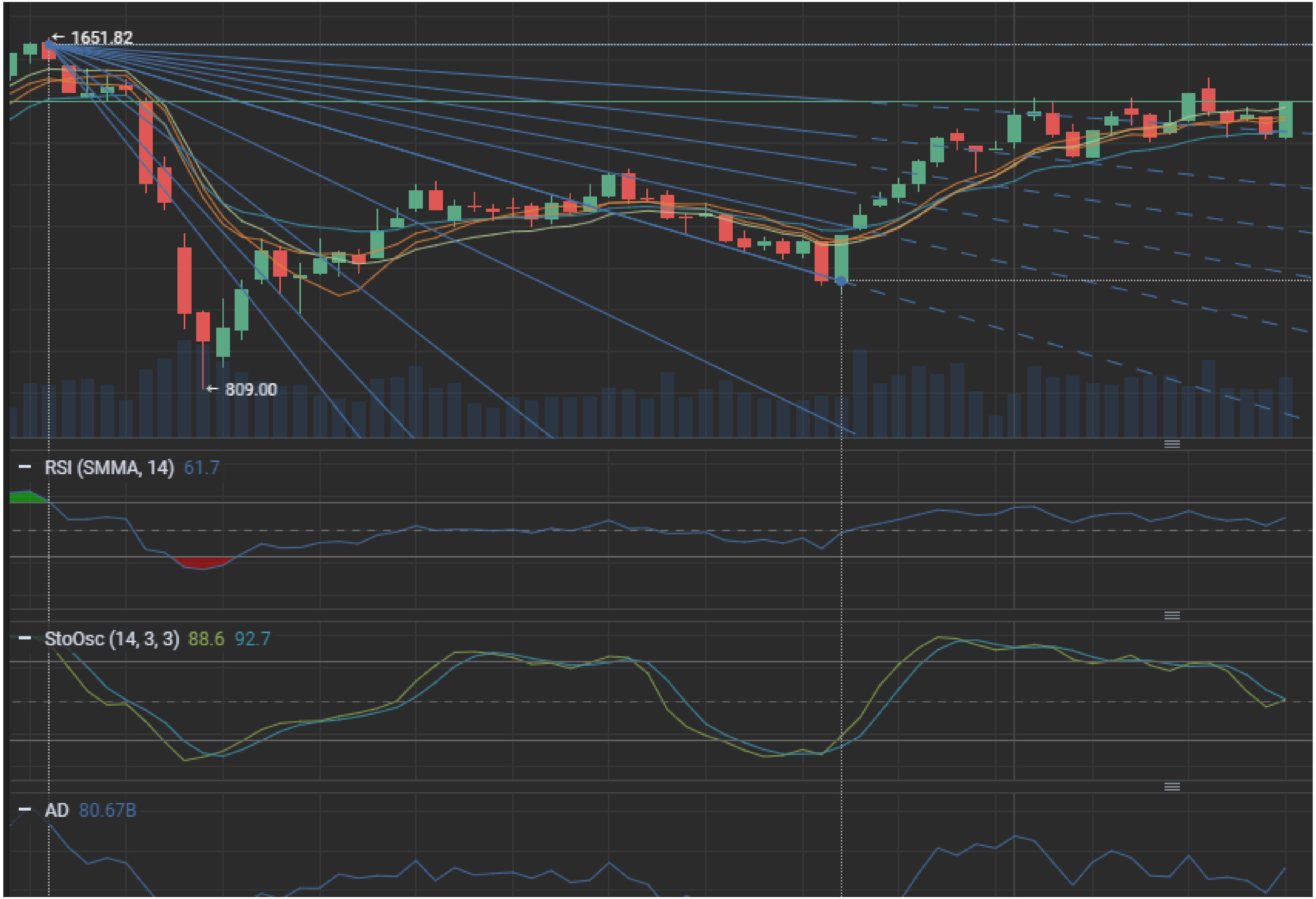
SPARK PROJECT STRUCTURE

SPARK ПРОЕКТ

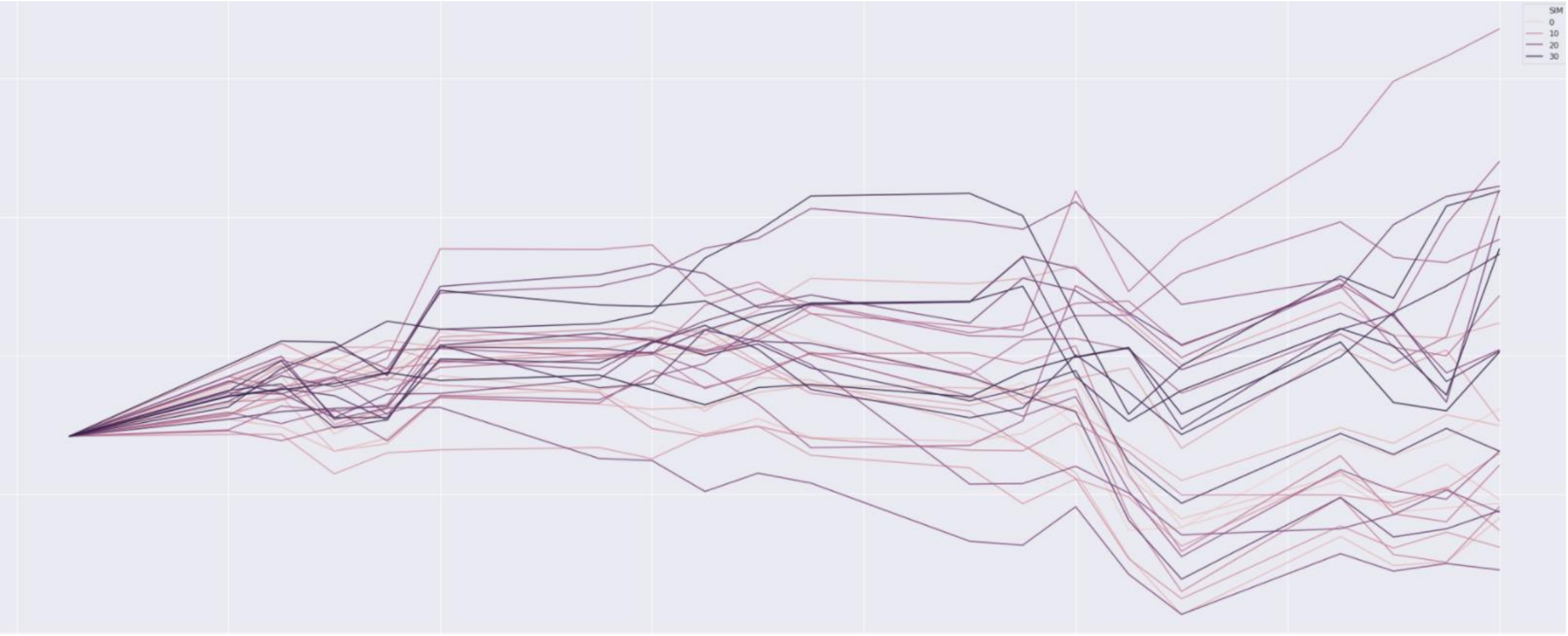
РЕШЕНИЕ
В КАРТИНКАХ



РЕШЕНИЕ В КАРТИНКАХ

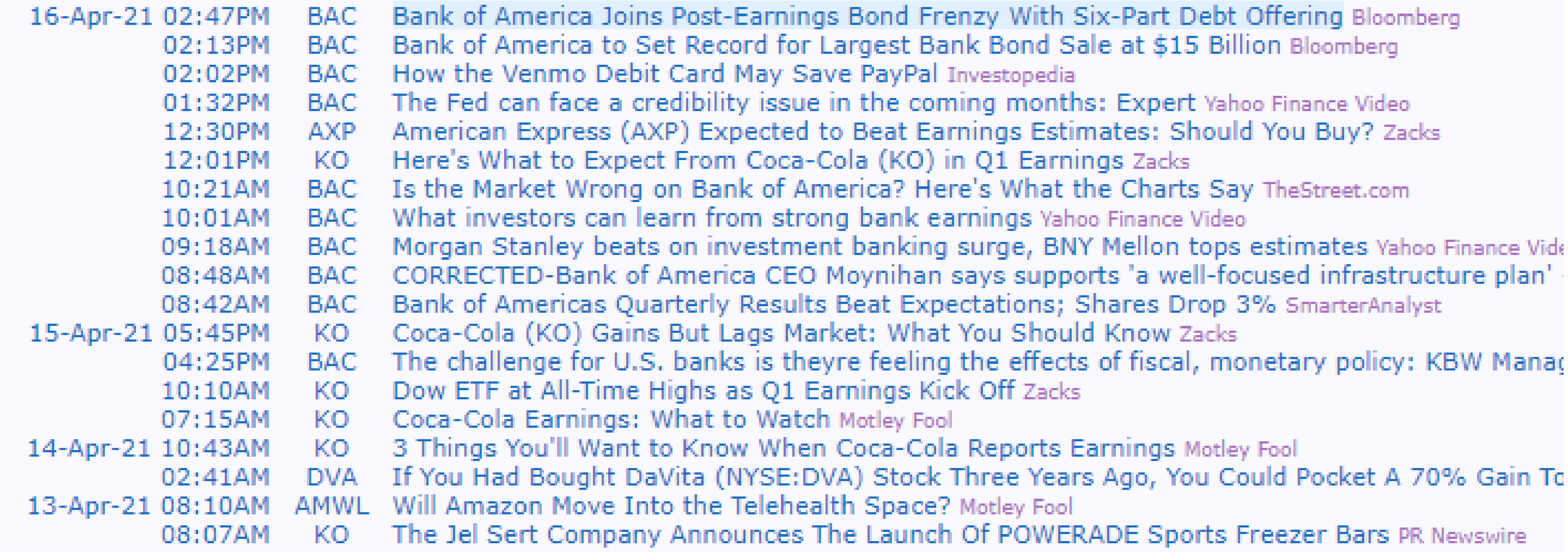


НА КАЖДУЮ МЕТРИКУ – МОНТЕ-КАРЛО



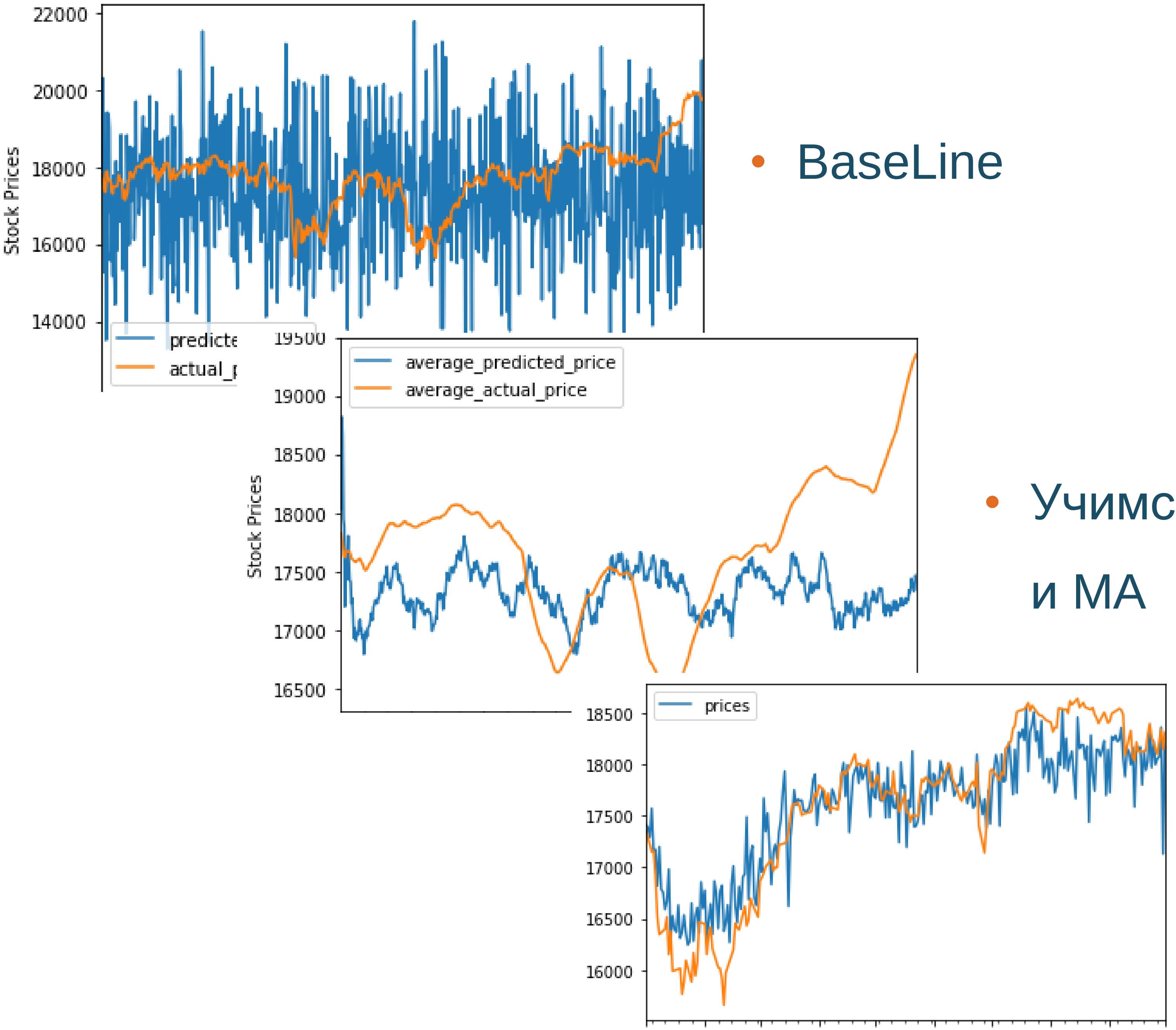
РАЗБОР ТЕКСТА

- TF-IDF
- Word2Vec
- Sentiment
- Key person | Company
- Relation



16-Apr-21 02:47PM	BAC	Bank of America Joins Post-Earnings Bond Frenzy With Six-Part Debt Offering Bloomberg
02:13PM	BAC	Bank of America to Set Record for Largest Bank Bond Sale at $15 Billion Bloomberg
02:02PM	BAC	How the Venmo Debit Card May Save PayPal Investopedia
01:32PM	BAC	The Fed can face a credibility issue in the coming months: Expert Yahoo Finance Video
12:30PM	AXP	American Express (AXP) Expected to Beat Earnings Estimates: Should You Buy? Zacks
12:01PM	KO	Here's What to Expect From Coca-Cola (KO) in Q1 Earnings Zacks
10:21AM	BAC	Is the Market Wrong on Bank of America? Here's What the Charts Say TheStreet.com
10:01AM	BAC	What investors can learn from strong bank earnings Yahoo Finance Video
09:18AM	BAC	Morgan Stanley beats on investment banking surge, BNY Mellon tops estimates Yahoo Finance Vide
08:48AM	BAC	CORRECTED-Bank of America CEO Moynihan says supports 'a well-focused infrastructure plan'
08:42AM	BAC	Bank of Americas Quarterly Results Beat Expectations; Shares Drop 3% SmarterAnalyst
15-Apr-21 05:45PM	KO	Coca-Cola (KO) Gains But Lags Market: What You Should Know Zacks
04:25PM	BAC	The challenge for U.S. banks is theyre feeling the effects of fiscal, monetary policy: KBW Manag
10:10AM	KO	Dow ETF at All-Time Highs as Q1 Earnings Kick Off Zacks
07:15AM	KO	Coca-Cola Earnings: What to Watch Motley Fool
14-Apr-21 10:43AM	KO	3 Things You'll Want to Know When Coca-Cola Reports Earnings Motley Fool
02:41AM	DVA	If You Had Bought DaVita (NYSE:DVA) Stock Three Years Ago, You Could Pocket A 70% Gain To
13-Apr-21 08:10AM	AMWL	Will Amazon Move Into the Telehealth Space? Motley Fool
08:07AM	KO	The Jel Sert Company Announces The Launch Of POWERADE Sports Freezer Bars PR Newswire

ИЩЕМ:
ЦЕНА ОТКРЫТИЯ
ОКНО 3, 5, 8 ДНЕЙ



- BaseLine
- Учимся с трендами и MA
- Финальная модель

APACHE SPARK

SELEZNEV ARTEM
HEAD OF CVM ANALYTICS @ MAGNIT

tg: @SeleznevArtem

/NameArtem

/seleznev-artem

/seleznev.artem.info