

using custom **Python distribution
with **Apache Spark****

PyconID 2017 - PENS

Python @ Bukalapak : Analytics & AI

- Interactive data analysis with **Jupyter/Zeppelin**
- Machine learning & AI models creation (**tensorflow, pytorch**)

Python @ Bukalapak : Pipelines

- **500+** custom data pipelines with **Airflow**
- TBs of **data processing** with **Spark**
- 200 cores, 1.2 TB of RAM
- Will scale at least 10x!

Challenges with Python : **Scaling**

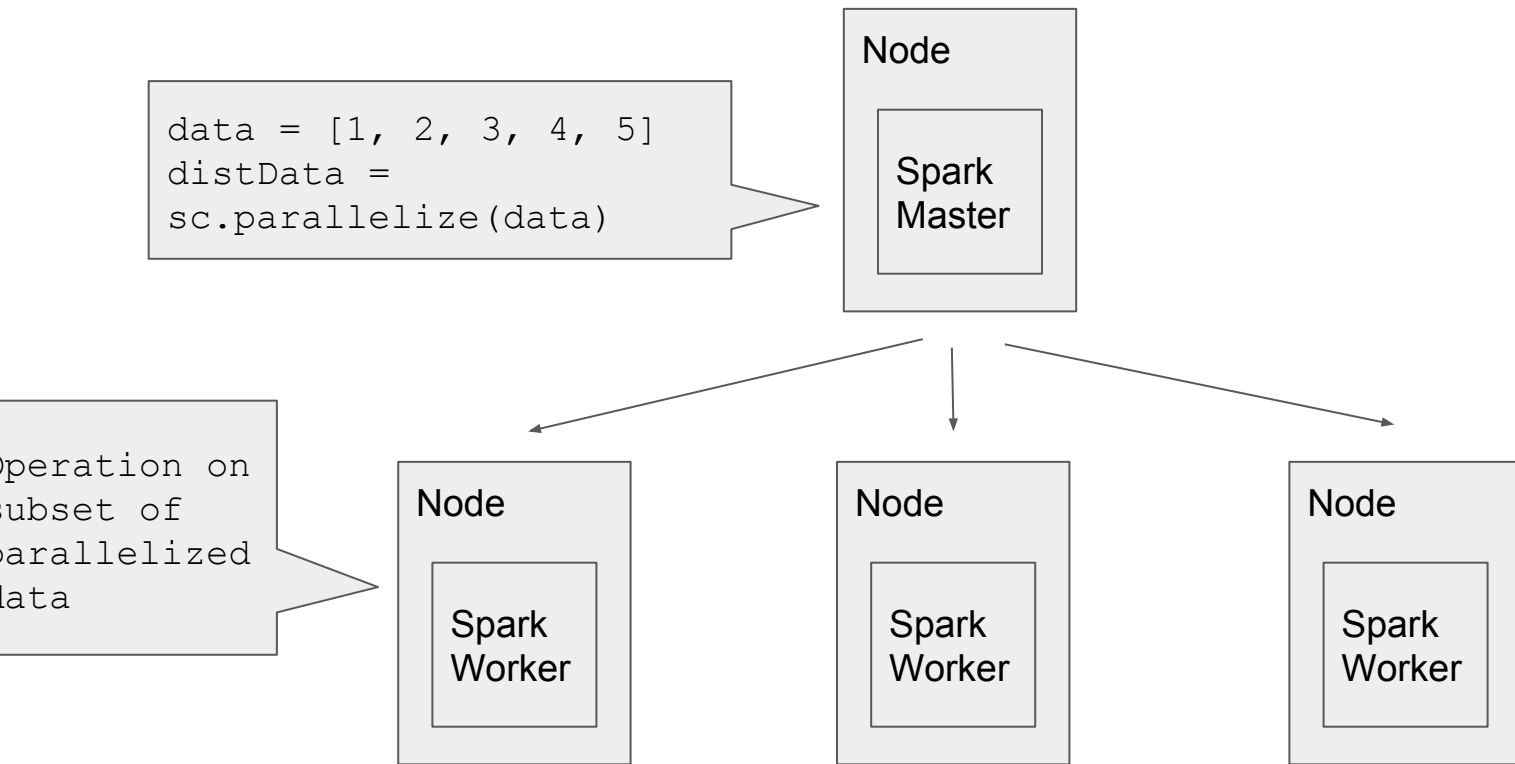
- Vertical scaling **vs** horizontal scaling
- Needs **abstraction** for easy Python horizontal scaling

Here comes the **Spark**

- **General purpose** cluster computing framework
- Support R, Java, Scala & **Python**
- Gives abstraction to **parallelize data processing** to n-nodes

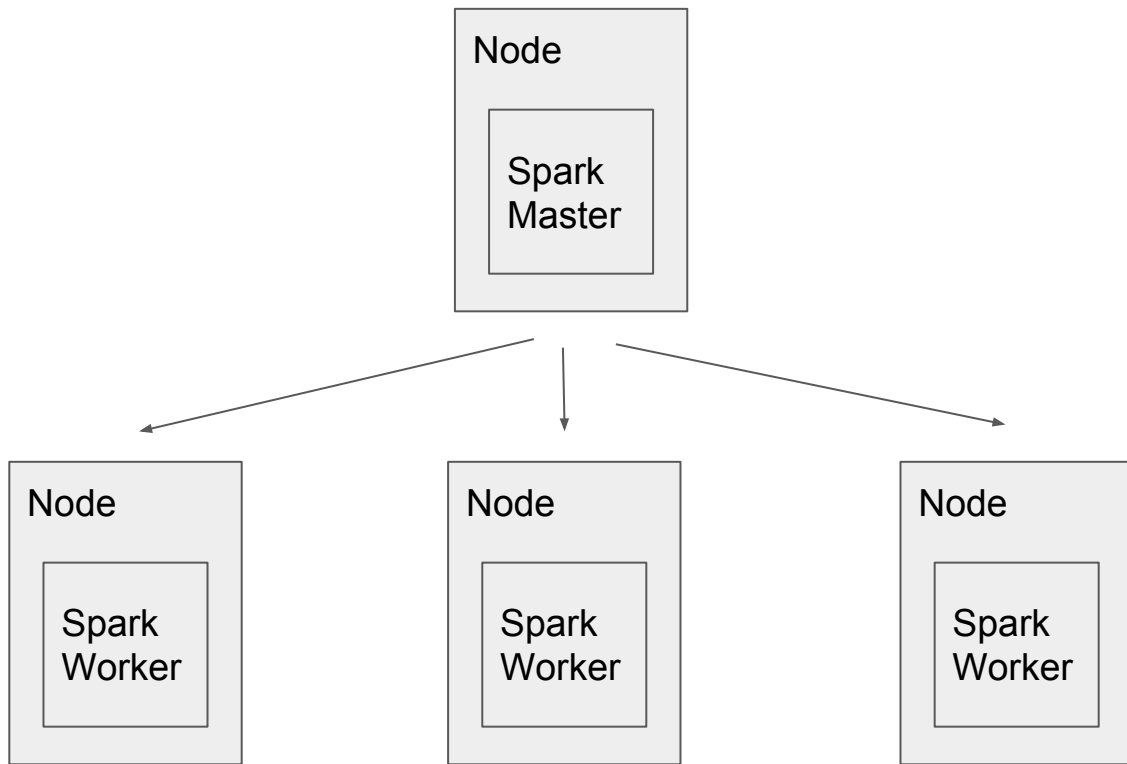


How spark **works**



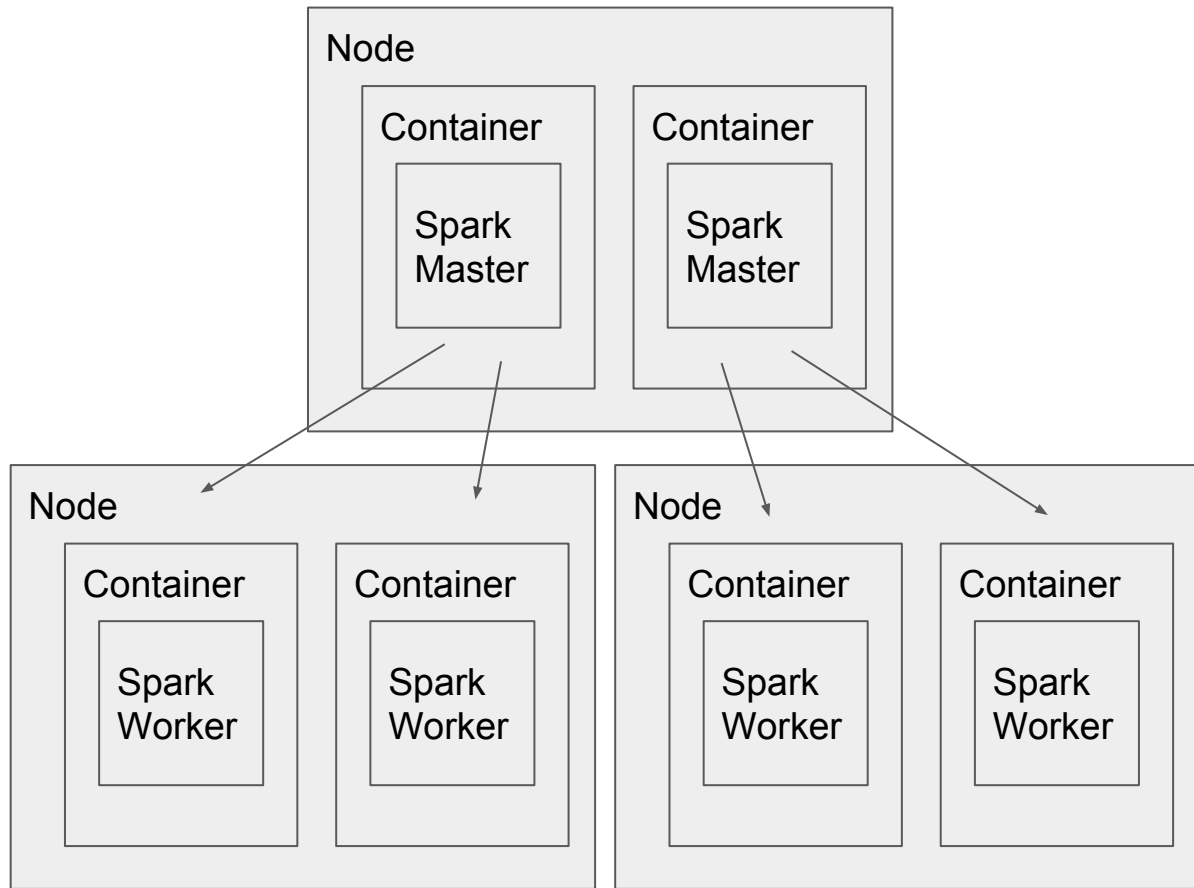
Ways to run Spark : **Standalone**

- Running spark master + worker **statically** on n-nodes
- N-nodes are **'locked'** with spark



Ways to run Spark : **YARN**

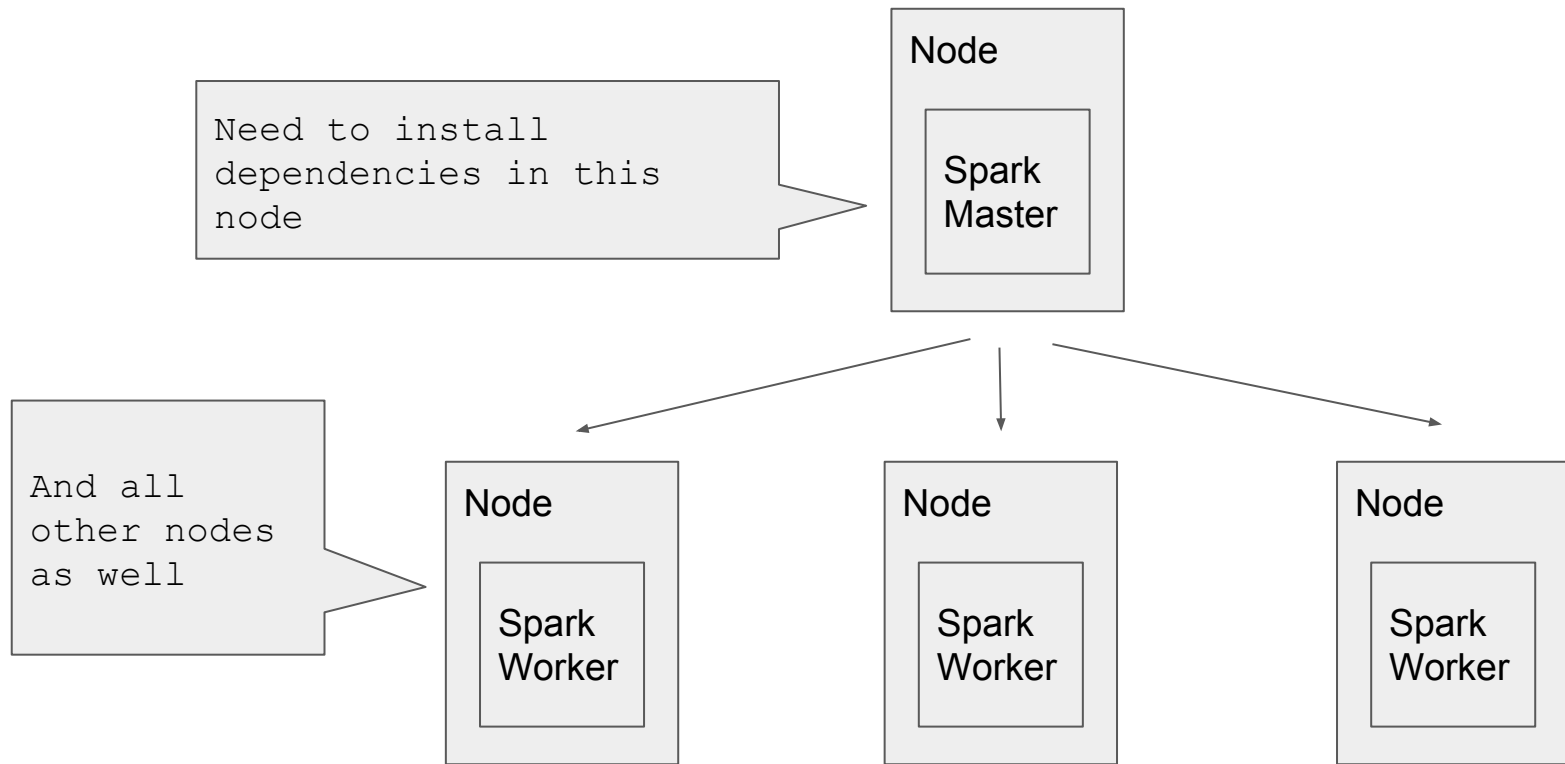
- Resource **manager** & job **scheduling** on n-nodes
- Putting apps into **containers**



Challenges with Python : **Dependencies**

- `pip -r requirements.txt`
- Will not work on non python dependencies
- Ex. library `hdfs3` - needs `libhdfs3.so`
- `pip install hdfs3` only install python wrapper
- Need to install `.so` with `apt-get`
- Or copy & add `LD_LIBRARY_PATH`
- Or use `conda -c conda-forge install hdfs3`

Challenges with Python : **Distributed dependencies**



Solution : use **automation**

`Ansible-playbook pip.yml`

- Auto install to a n-nodes
- Have an integrated pip & venv modules



Challenges with Python : **Different Distributed dependencies**

Compatibility

The library is compatible with all Elasticsearch versions since `0.90.x` but you **have to use a matching major version**:

For **Elasticsearch 6.0** and later, use the major version 6 (`6.x.y`) of the library.

For **Elasticsearch 5.0** and later, use the major version 5 (`5.x.y`) of the library.

For **Elasticsearch 2.0** and later, use the major version 2 (`2.x.y`) of the library, and so on.

It's super **complicated**

- Switching env before running each Spark job ?
 - Spark can't run shell commands
- Installing/uninstalling different library as base
 - Scheduling / blocking mess
- But, spark can specify what Python **BINARY** path to run during startup

Building a custom Python **binary**

download and extract Python (using 2.7.12 here as an example)

```
export PYTHON_ROOT=~/.Python
```

```
curl -O https://www.python.org/ftp/python/2.7.12/Python-2.7.12.tgz
```

```
tar -xvf Python-2.7.12.tgz
```

compile into local PYTHON_ROOT

```
cd Python-2.7.12
```

```
./configure --prefix="${PYTHON_ROOT}" --enable-unicode=ucs4
```

```
make
```

```
make install
```

Building a custom Python **binary + dependencies**

install pip

```
cd ${PYTHON_ROOT}
```

```
curl -O https://bootstrap.pypa.io/get-pip.py
```

```
bin/python get-pip.py
```

```
rm get-pip.py
```

Note: add any extra dependencies here, e.g. pydoop

```
${PYTHON_ROOT}/bin/pip install pydoop
```

https://github.com/yahoo/TensorFlowOnSpark/wiki/GetStarted_YARN

It **works** with YARN containers !

- Files put in hdfs will can be pulled into YARN containers and extracted

zip the custom python distribution

```
cd "${PYTHON_ROOT}"
```

```
zip -r Python.zip *
```

copy this Python distribution into HDFS

```
hadoop fs -put ${PYTHON_ROOT}/Python.zip
```


It **works** with YARN containers !

pass the HDFS url when submitting spark jobs

```
${SPARK_HOME}/bin/spark-submit \  
--master yarn \  
--deploy-mode cluster \  
--num-executors 4 \  
--executor-memory 4G \  
--archives hdfs:///new-python-binary-hdfs/Python.zip#Python
```

and pass the new path to spark variable

```
--conf spark.executorEnv.PYSPARK_PYTHON=./Python/bin/python
```

Problem **solved** !

- Create **multiple** python distributions zip file in HDFS depending your configurations need
- If you **update** dependencies, just need to **update** file in HDFS
- Note : don't forget to use the binary in the node where you **submitting** spark jobs as well

Lessons **learned**

- Things got complicated when want to scale **horizontally**
- **Decoupling** between base python in node vs runtime python in containers
- You might **need** or might **not need** to use custom Python binary
- Comparison **vs** using a custom docker image

Interested ? We're **hiring** !

<https://careers.bukalapak.com>

recruitment@bukalapak.com

Thanks :)