BIG DATA EUROPE

Empowering Communities
with Data Technologies

BDE Architecture

universitätbonn
Rheinische
Friedrich-Wilhelms-
Universität Bonn

Hajira Jabeen, University of Bonn

M1-M18 Review Meeting

Structure

- Evolution of BDE architecture
- User of BDE
- Working

Platform Description

Technology assessment

- Lessons learned:
  - A lot of technologies available
  - Big Data space moves fast
  - High barrier to entry

- Focus:
  - Ease of use
    - Installation, development, deployment, monitoring
  - Flexibility
    - Keep options open for future
  - Reuse effort of the community
    - Don't reinvent the wheel

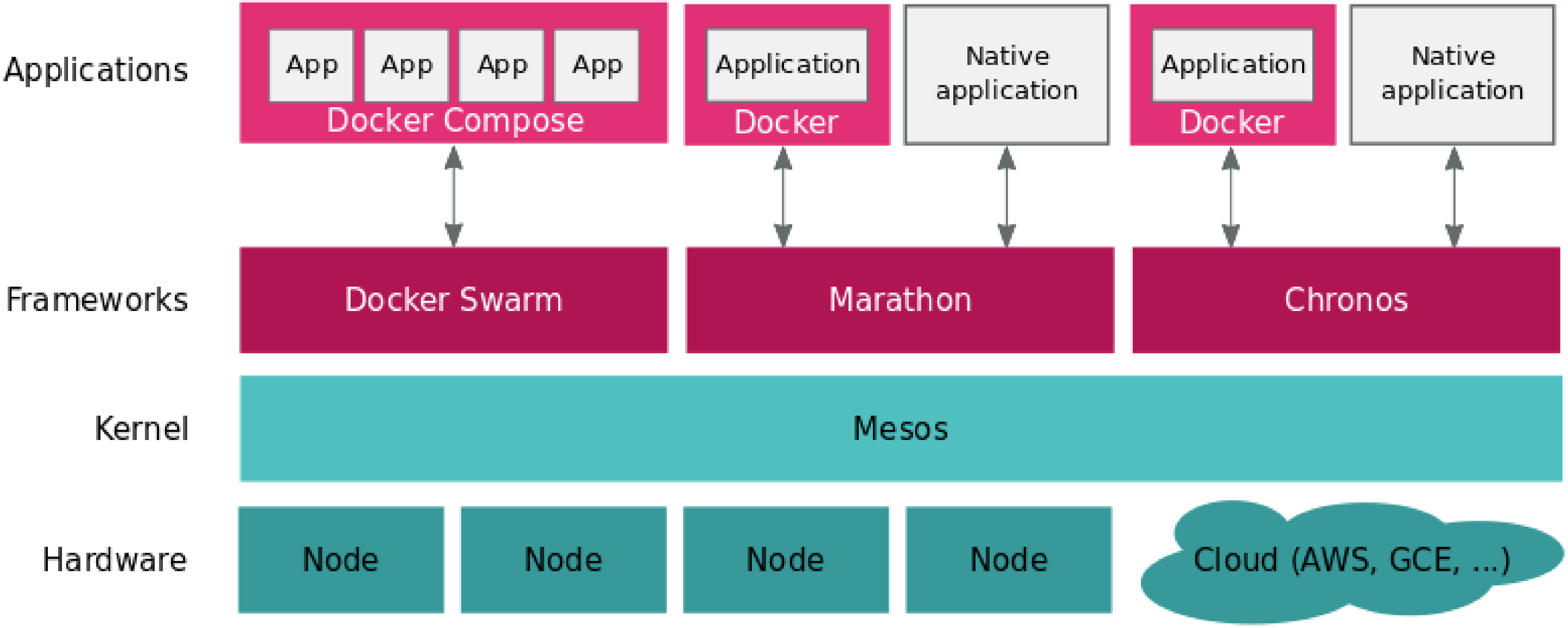
Technical requirements

- Input:
  - WP2: General requirements elicitation
  - WP5: Specific pilot requirements

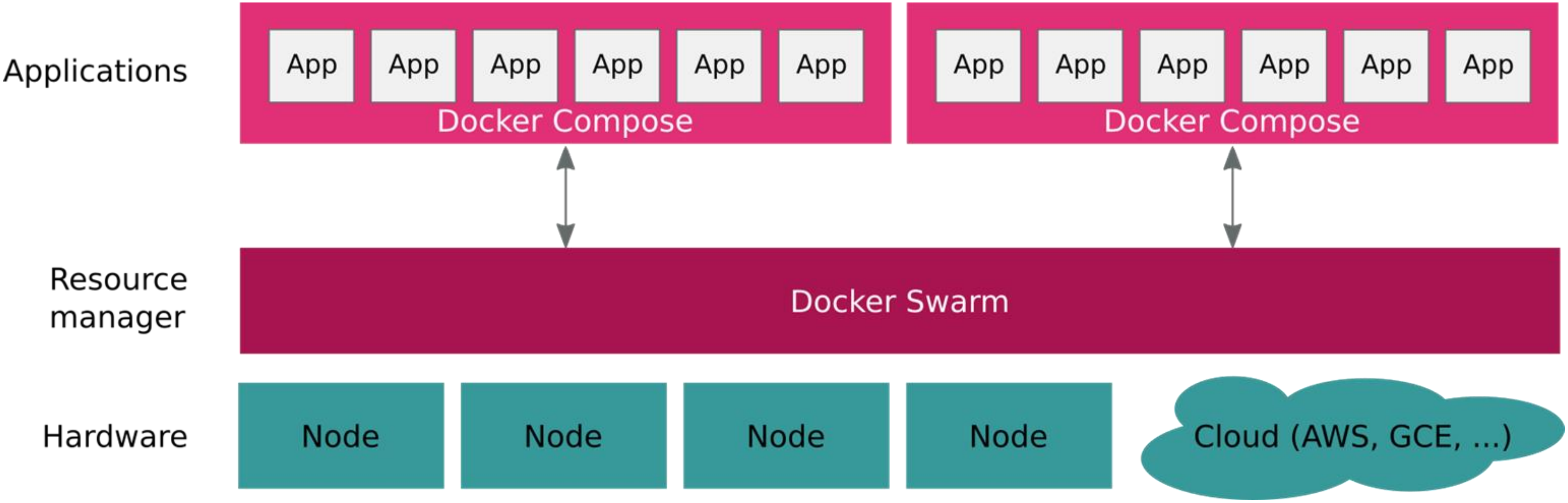
- Initial idea: platform profile per V
  - Not 1 V that overrules the others per SC

    ⇒ Provide component suggestions per V

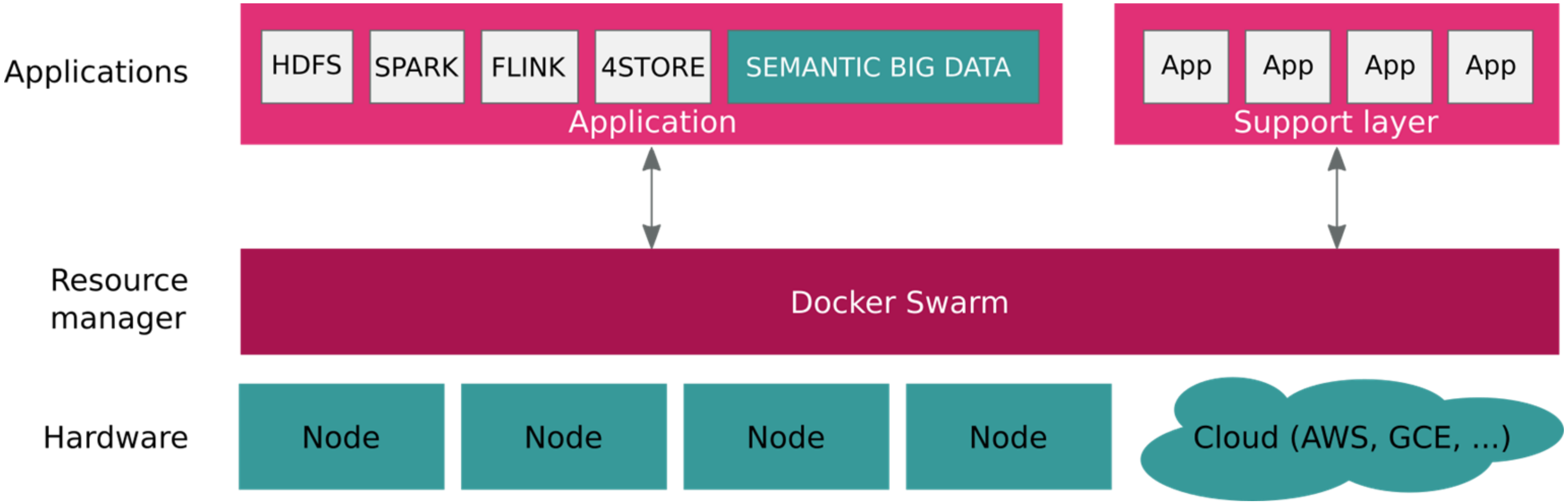
Architectural design



Architectural design



Architectural design

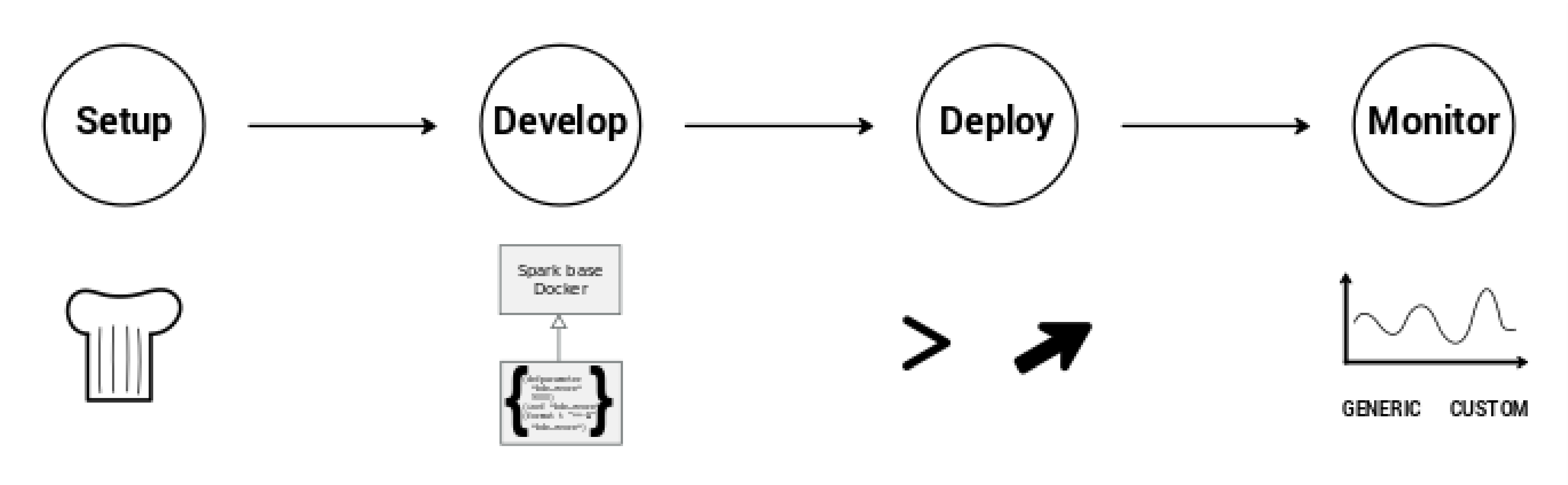


User of BDE

The minimum knowledge requirements for the BDE user are:

- Ability to write programs for his particular use case
- Inter connectivity of components, if he wants to create a pipeline of different components
- Basics of distributed systems and web-services
- However, this does not exclude experienced users or data scientists from using the platform with ease.

User profiles



Setup

Develop

Deploy

Monitor

Spark base Docker

GENERIC

CUSTOM

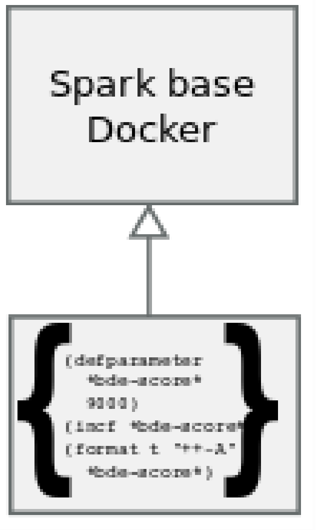
Platform installation

- Manual installation guide
- Using Docker Machine
  - On local machine (VirtualBox)
  - In cloud (AWS, DigitalOcean, Azure)
  - Bare metal
- Screencast

Developing a component

- Base Docker images
  - Serve as a template for a (Big Data) technology
  - Easily extendable custom algorithm/data

- Published components
  - Responsibilities divided b/w partners
  - Image repositories on GitHub
  - Automated builds on DockerHub
  - Documentation on BDE Wiki



```
(defparameter
  *bde-score*
  9000)
(incf *bde-score*
(format t "**-A"
  *bde-score*)
```

Deploying a Big Data pipeline

- Pipeline:

  collection of communicating components
  to solve a specific problem

- Described in Docker Compose
  - Component configuration
  - Application topology

- Orchestrator required for initialization process
  - Components may depend on each other
  - Components may require manual intervention

Scalability of BDE

- 1000 Nodes
- 3000 Containers
- 1 Swarm Manager
- Docker swarm V 1.0

BDE vs Hadoop distributions

BDE vs Hadoop distributions

	Hortonworks	Cloudera	MapR	Bigtop	BDE
File System	HDFS	HDFS	NFS	HDFS	**HDFS**
Installation	Native	Native	Native	Native	**lightweight virtualization**
Plug & play components (no rigid schema)	no	no	no	no	**yes**
High Availability	Single failure recovery (yarn)	Single failure recovery (yarn)	Self healing, mult. failure rec.	Single failure recovery (yarn)	**Multiple Failure recovery**
Cost	Commercial	Commercial	Commercial	Free	**Free**
Scaling	Freemium	Freemium	Freemium	Free	**Free**
Addition of custom components	Not easy	No	No	No	**Yes**
Integration testing	yes	yes	yes	yes	--
Operating systems	Linux	Linux	Linux	Linux	**All**
Management tool	Ambari	Cloudera manager	MapR Control system	-	**Docker swarm UI+ Custom**

BDE vs Hadoop distributions

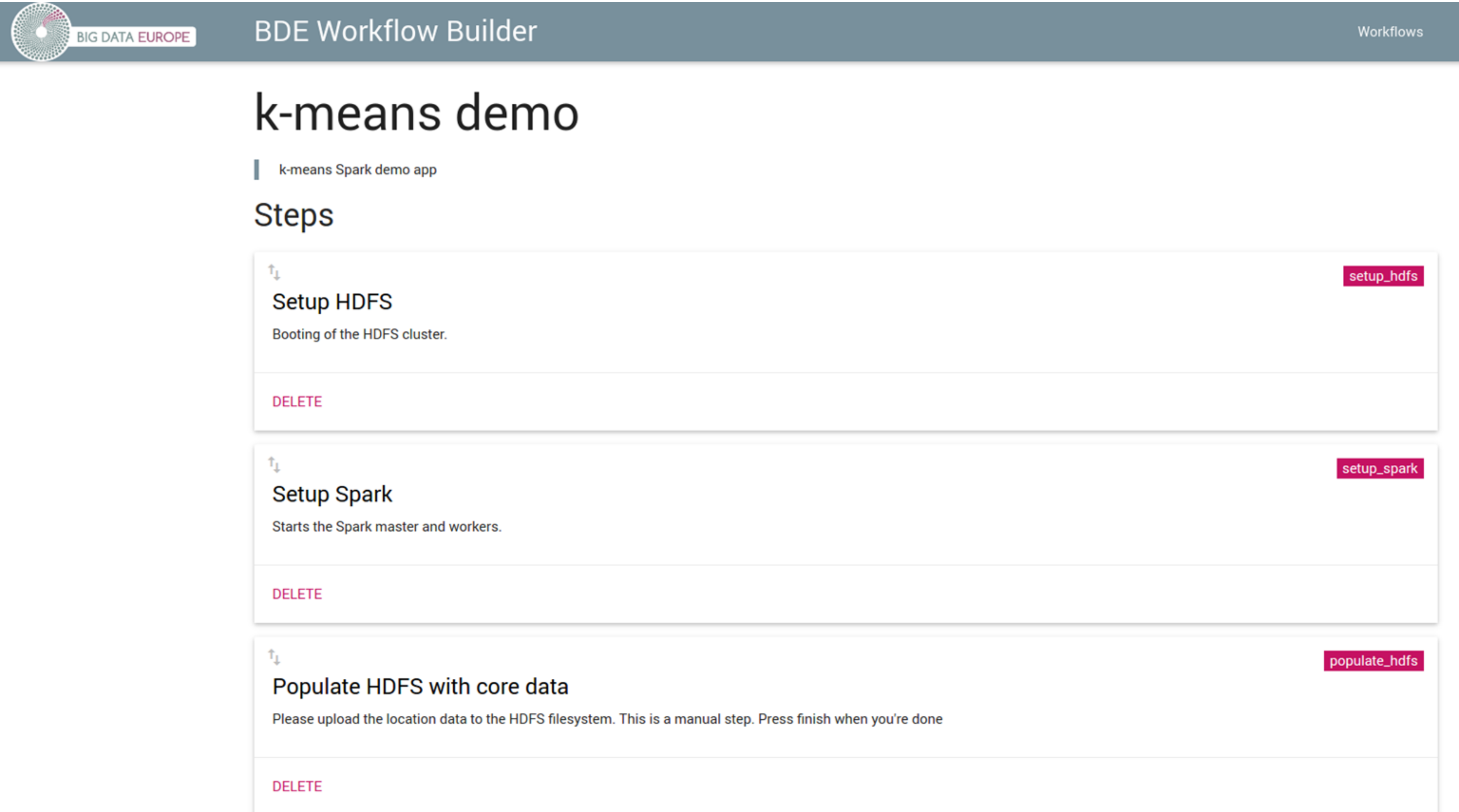
BDE is:

- Not built on top of existing distributions
- Targets
  - Communities
  - Research institutions
- Bridges scientists and open data
- Multi Tier research efforts towards Smart Data

User interfaces

- Target: facilitate use of the platform
- Available interfaces
  - Workflow UIs
    - Workflow Builder
    - Workflow Monitor
  - Swarm UI
  - Integrator UI

BDE Workflow builder



BDE Workflow Builder

k-means demo

> k-means Spark demo app

Steps

setup_hdfs

Setup HDFS

Booting of the HDFS cluster.

DELETE

setup_spark

Setup Spark

Starts the Spark master and workers.

DELETE

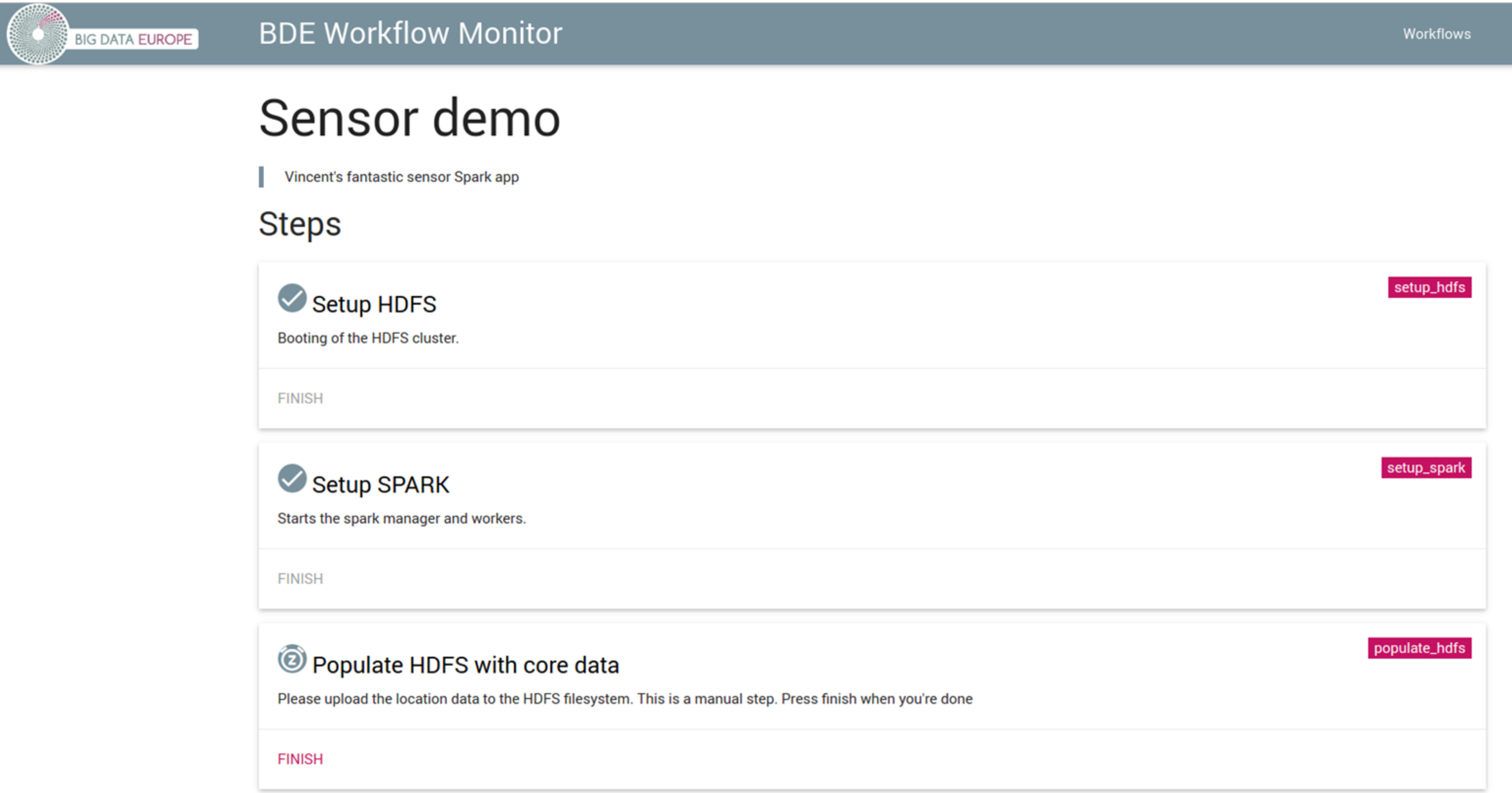
populate_hdfs

Populate HDFS with core data

Please upload the location data to the HDFS filesystem. This is a manual step. Press finish when you're done

DELETE

BDE Workflow monitor



Sensor demo

> Vincent's fantastic sensor Spark app

Steps

✓ Setup HDFS — setup_hdfs

Booting of the HDFS cluster.

FINISH

✓ Setup SPARK — setup_spark

Starts the spark manager and workers.

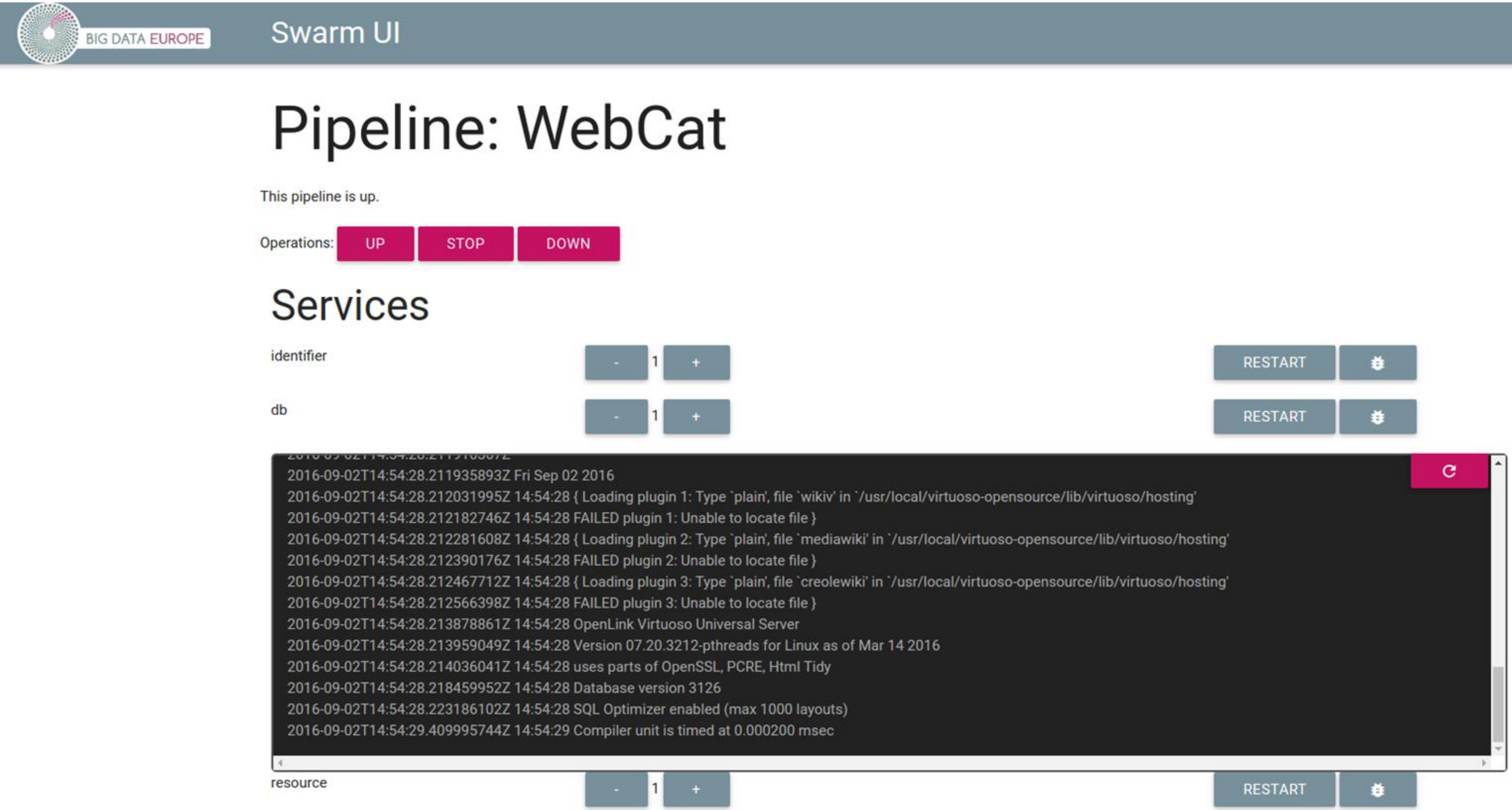
FINISH

Populate HDFS with core data — populate_hdfs

Please upload the location data to the HDFS filesystem. This is a manual step. Press finish when you're done

FINISH

Swarm UI



Swarm UI

Pipeline: WebCat

This pipeline is up.

Operations: UP STOP DOWN

Services

identifier

db

```
2016-09-02T14:54:28.211935893Z Fri Sep 02 2016
2016-09-02T14:54:28.212031995Z 14:54:28 { Loading plugin 1: Type `plain', file `wikiv' in `/usr/local/virtuoso-opensource/lib/virtuoso/hosting'
2016-09-02T14:54:28.212182746Z 14:54:28 FAILED plugin 1: Unable to locate file }
2016-09-02T14:54:28.212281608Z 14:54:28 { Loading plugin 2: Type `plain', file `mediawiki' in `/usr/local/virtuoso-opensource/lib/virtuoso/hosting'
2016-09-02T14:54:28.212390176Z 14:54:28 FAILED plugin 2: Unable to locate file }
2016-09-02T14:54:28.212467712Z 14:54:28 { Loading plugin 3: Type `plain', file `creolewiki' in `/usr/local/virtuoso-opensource/lib/virtuoso/hosting'
2016-09-02T14:54:28.212566398Z 14:54:28 FAILED plugin 3: Unable to locate file }
2016-09-02T14:54:28.213878861Z 14:54:28 OpenLink Virtuoso Universal Server
2016-09-02T14:54:28.213959049Z 14:54:28 Version 07.20.3212-pthreads for Linux as of Mar 14 2016
2016-09-02T14:54:28.214036041Z 14:54:28 uses parts of OpenSSL, PCRE, Html Tidy
2016-09-02T14:54:28.218459952Z 14:54:28 Database version 3126
2016-09-02T14:54:28.223186102Z 14:54:28 SQL Optimizer enabled (max 1000 layouts)
2016-09-02T14:54:29.409995744Z 14:54:29 Compiler unit is timed at 0.000200 msec
```

resource