

BIG DATA EUROPE

*Coordination and Support Action*

## Big Data Europe – Empowering Communities with Data Technologies

Project Number: 644564

Start Date of Project: 01/01/2015

Duration: 36 months

### Deliverable 4.2: Second Big Data Integrator Platform Release

Dissemination Level	Public
Due Date of Deliverable	M20, 31/08/2016
Actual Submission Date	M20, 04/11/2016
Work Package	WP4, Big Data Integrator Platform & Implementation
Task	T4.2
Type	Other
Approval Status	Final
Version	V1.0
Number of Pages	12
Filename	D4.2_Second_Big_Data_Integrator_Platform_Release.pdf

**Abstract:** This document presents a brief overview of the BDE platform released publically on 31/08/2016.

The information in this document reflects only the author's views and the European Community is not liable for any use that may be made of the information contained therein. The information in this document is provided "as is" without guarantee or warranty of any kind, express or implied, including but not limited to the fitness of the information for a particular purpose. The user thereof uses the information at his/ her sole risk and liability.





## History

Version	Date	Reason	Revised by
0.1	21/10/2016	Initial version	Hajira Jabeen, Gezim Sejdiu and Jens Lehmann (UBO), Aad Verstaden (Tenforce), Ivan Ermilov (InfAI)
0.2	31/10/2016	Peer-review	Hossain Abroshan (CESSDA)
1.0	04/11/2016	Final version	Hajira Jabeen, Gezim Sejdiu and Jens Lehmann (UBO), Aad Verstaden (Tenforce), Ivan Ermilov (InfAI)

## Author List

Organization	Name	Contact Information
UBO	Jens Lehmann	jens.lehmann@cs.uni-bonn.de
UBO	Hajira Jabeen	jabeen@cs.uni-bonn.de
InfAI	Ivan Ermilov	iermilov@informatik.uni-leipzig.de
Tenforce	Aad Verstaden	aad.versteden@tenforce.com
CESSDA	Hossain Abroshan	hossein.abroshan@cessda.net



# Executive Summary

The first public release of the open source [BigDataEurope Platform](#) has been successfully announced on August 31, 2016. This platform is designed to help communities to solve societal challenges and problems by accelerating the process of getting started with the big data technology-zoo.

The BDE platform is an easy-to-deploy, easy-to-use and adaptable (cluster-based and standalone) platform for the execution of big data frameworks and tools like [Apache Hadoop](#), [Apache Spark](#), [Apache Flink](#) and many others. We have selected these latter based upon the requirements gathered from the seven different societal challenges. Thus, the platform allows to perform a variety of big data flow tasks like message passing ([Kafka](#), [Flume](#)), storage ([Hive](#), [Cassandra](#)) or publishing ([geotriples](#)).

The user can follow a simple set of [instructions](#) to install, or follow the installation [video](#) for getting started with the platform. The user can then pull ready-to-use images from the [BDE repository](#) or create a [dataflow pipeline](#) to realize a full data value chain using the components available at the BDE repository. This platform will be refined and expanded collaboratively with the users and the societal challenge (use case) partners. This public release of our platform represents the beginning of the operational phase, with which the use cases will be materialized and tested on the developed platform.

In the future, the BDE platform aims to introduce the concept of Smart Data that would support a range of processing tools for the semantic web and knowledge graphs. This includes the concept of semantic data lake, a semantic analytics stack engine and a semantic platform for logging and integration for cluster resilience.

Overall, the platform has lowered the barrier to entry for new big data users and scientists from different domains to experiment with a variety of big data tools in a plug-and-play fashion.

This release was publically announced on several channels including

- [Twitter](#)
- [BDE Blog](#)
- [LinkedIn](#)
- [Press Release](#)
- And several mailing lists.



# Table of Contents

1. Introduction .....	6
1.1 Platform Overview .....	6
1.2 Support Layer.....	7
1.3 User Interfaces.....	8
1.3.1 Workflow Builder .....	8
1.3.2 Workflow Monitor .....	9
1.3.3 Swarm UI .....	10
1.3.4 Integrator UI .....	11
2. Conclusion and Future Work.....	11
3. References.....	12



# List of Figures

Figure 1: BDE Platform Architecture .....	7
Figure 2: Platform Components .....	8
Figure 3: BDE WorkFlow Builder .....	9
Figure 4: BDE WorkFlow Monitor .....	10
Figure 5: Swarm UI displaying a running pipeline .....	10
Figure 6: BDE Integrator UI (Dashboard).....	11



# 1. Introduction

The platform presented in D 4.1 (December, 2015) used Mesos as the distributed kernel and we were planning to deploy dockers of big data tools through marathon and chronos frameworks. However, based on our research, rapid development of the Big Data Tools horizon, and several factors detailed below, we have updated the previous architecture of the BDE platform.

## 1. Dockerization<sup>1</sup> of technologies

During the last months we have dockerized several (Big Data) technologies. None of them posed major problems to be run in a Docker container. Based on our experience and the movement around Docker, including major parties like Microsoft, we are getting more confident that all components can be dockerized in reasonable time. This evolvment erases the platform constraint to have a native application deployment mechanism.

## 2. Integration of Mesos and Dockers

Docker is one of the supported containerization technologies. However, Docker support is limited to Docker Engine only. The frameworks support the deployment of single, isolated Docker containers. They do not support the deployment of a collection of communicating containers through Docker Compose, which is the preferred way of describing and deploying applications in the BDE platform as described in D3.4 [1].

The integration between Mesos and Swarm is not production-ready yet. Moreover, while the Swarm maintainers actively contributed to the integration during the second half of 2015, the integration now seems to have lost traction. The code base did not majorly change since March 2016. As a consequence, several unresolved issues make deploying Docker containers through Docker Swarm on a Mesos cluster in production near impossible today.

## 3. Development of Docker Swarm

Swarm is an orchestration tool that allows to deploy Docker containers on a cluster in a transparent way. Each node in the cluster runs a Swarm agent. Together the nodes constitute a swarm which is managed by a Swarm manager. The Swarm manager serves the same API as the Docker Engine. As such, using the same commands, one can deploy dockerized applications on a single machine or on a cluster. Docker Swarm has evolved a lot since our initial research on the BDE project. A stable production-ready v1.0.0 has been released in November 2015. Ever since, Swarm kept - and still keeps - improving based on the input and feedback from the community.

## 1.1 Platform Overview

The new architecture is illustrated in Figure 1. The Docker Swarm, with its built-in scheduler, offers most of the features required by the platform i.e. scalability, interlinking the containers, networking between different containers, resource management between containers, load balancing, fault tolerance, failure recovery, log based monitoring etc.

---

<sup>1</sup> Creation of a docker container of an application

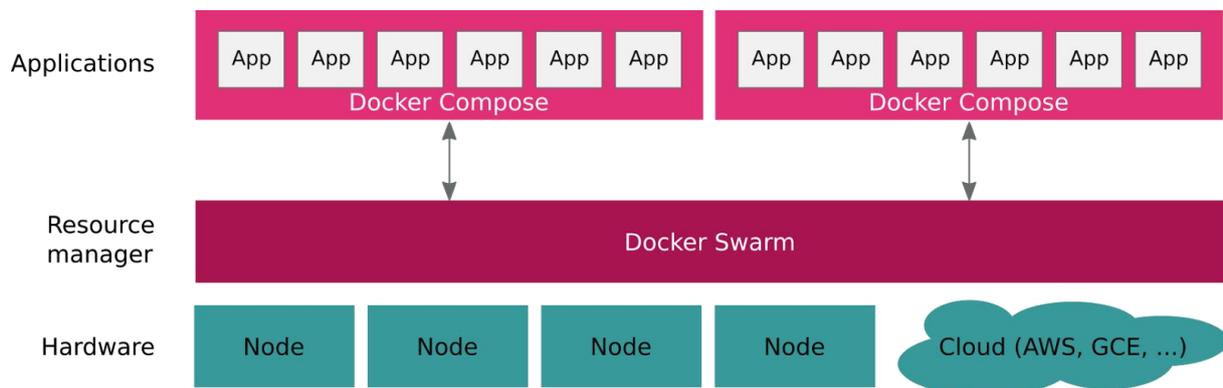


Figure 1: BDE Platform Architecture

Docker Swarm operates act as a resource manager directly on top of the hardware layer as a replacement of Mesos. This hardware layer can be on premises machines or any cloud provider service. On top of Swarm, applications can be deployed easily through Docker as a single container, or through Docker Compose as a collection (cluster) of communicating containers.

## 1.2 Support Layer

The technical aim of the BDE is to reuse the components wherever possible and build the tools which are necessary to fit the societal challenge needs. In this regard, we discovered that when starting a Docker Compose application, the services defined in the docker-compose.yml start running all at once. This is not always the intended behaviour. Some applications may depend on each other, or on a human intervention. For example, a Flink worker requires the Flink master to be available before it can register with the master. Another example is a Flink MapReduce algorithm that requires a file to be available on HDFS before starting the computation. At the moment, these dependencies cannot be expressed in Docker Compose.

Awaiting the general solution of the Docker community<sup>2</sup>, we have developed a semantic alternative. We provide an init daemon service that, given an application-specific workflow, orchestrates the initialization process of the several components. On one hand, the init daemon service, a microservice built on the mu.semte.ch<sup>3</sup> platform, provides requests through which the components can report their initialization progress. Moreover, because the init daemon knows the startup workflow, it can validate whether a specific component can start based on the initialization statuses reported by the other components.

The workflow needs to be described per application. It specifies the dependencies between services and indicates where human interaction is required.

The GUIs described in the next section together with the base Docker images and the init daemon service provide additional support to the user. It facilitates the tasks of building, deploying and monitoring Big Data pipelines on the BDE platform. This support is illustrated in Figure 2 as an additional layer in the platform architecture.

<sup>2</sup> <https://github.com/docker/compose/issues/374>

<sup>3</sup> See <https://mu.semte.ch>

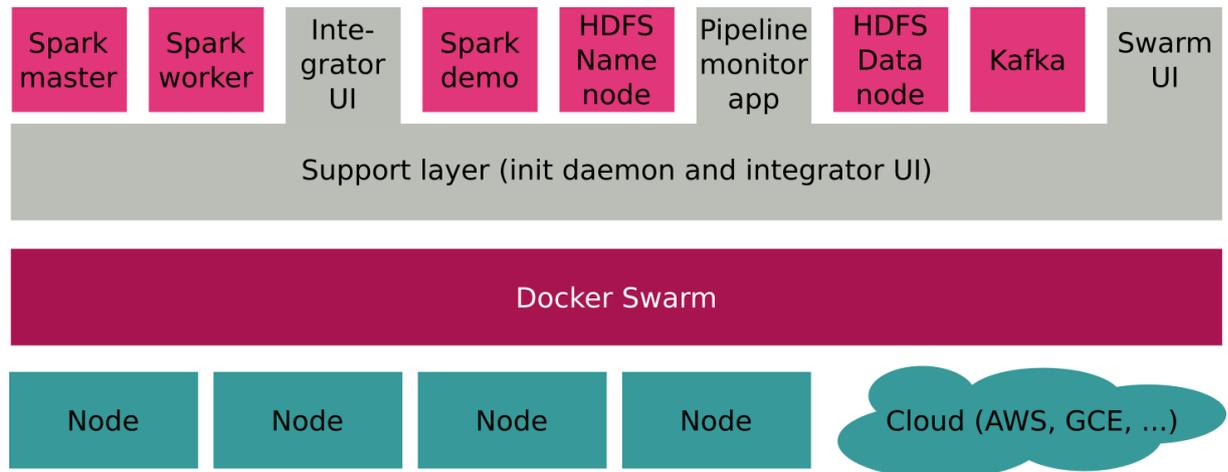


Figure 2: Platform Components

## 1.3 User Interfaces

As we want to lower the barrier to Big Data technologies, we have invested time to specify and implement GUIs. This has resulted in a pipeline builder application, a pipeline monitor application, an integrator UI and a Swarm UI. The GUIs serve different purposes, but in general make it easier for the user to build, deploy and monitor applications on the BDE platform. Most of the GUI applications are built on mu.semte.ch, a microservices framework backed by Linked Data. Each of the applications is described in more detail in the next sections.

### 1.3.1 Workflow Builder

This is a web page that allows user to create a new workflow. Each workflow has a title and a description. After creating a workflow, steps can be added to the workflow (see Figure 3). The steps reflect the dependencies between the images and/or manual actions in the application. Which container depends on another container? Which (manual) actions must be executed before a container can start? Example dependencies could be:

- The Spark master needs to be started before the Spark worker such that the Spark worker can register itself at the Spark master.
- The input data needs to be loaded in HDFS before the MapReduce algorithm starts computing.

The order of the steps can be rearranged by dragging-and-dropping the step panels. Once finished the workflow can be exported in Turtle format to be fed into the init daemon service.



# k-means demo

k-means Spark demo app

## Steps

 <b>Setup HDFS</b> Booting of the HDFS cluster.	
	
 <b>Setup Spark</b> Starts the Spark master and workers.	
	
 <b>Populate HDFS with core data</b> Please upload the location data to the HDFS filesystem. This is a manual step. Press finish when you're done	
	

Figure 3: BDE WorkFlow Builder

## 1.3.2 Workflow Monitor

Once an application is running on the BDE platform, the Workflow monitor application allows a user to follow-up the initialization process. It displays the workflow as defined in the pipeline builder application (see Figure 4). For each step in the workflow, the corresponding status (not started, running or finished) is shown as retrieved from the init daemon service. The interface automatically updates when a status changes, due to an update through the init daemon service by one of the pipeline components. As such, the user can follow-up the initialization process. The interface also offers the option to the user to manually finish a step in the pipeline if necessary.



The screenshot shows the 'BDE Workflow Monitor' interface. At the top, there is a header with the 'BIG DATA EUROPE' logo and the text 'BDE Workflow Monitor' and 'Workflows'. Below the header, the main title is 'Sensor demo'. Underneath, it says 'Vincent's fantastic sensor Spark app'. The 'Steps' section contains three items:

- Setup HDFS**: Booting of the HDFS cluster. Status: FINISH. Action button: setup\_hdfs.
- Setup SPARK**: Starts the spark manager and workers. Status: FINISH. Action button: setup\_spark.
- 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. Status: FINISH. Action button: populate\_hdfs.

Figure 4: BDE WorkFlow Monitor

### 1.3.3 Swarm UI

The Swarm UI allows to clone a Git repository containing a pipeline (i.e. containing a docker-compose.yml) and to deploy this pipeline on a Swarm cluster. Once the pipeline is running, the user can inspect the status and logs of the several services in the pipeline (see Figure 5). He/she can also scale up/down one or more services or start/stop/restart them. All these operations can be carried out through a graphical interface by just clicking some buttons.

The screenshot shows the 'Swarm UI' interface for a 'Pipeline: WebCat'. It indicates 'This pipeline is up.' and provides 'Operations: UP STOP DOWN' buttons. Below, a 'Services' table is displayed:

Service	Scale	Restart	Refresh
db	- 1 +	RESTART	Refresh
resource	- 3 +	RESTART	Refresh
dispatcher	- 3 +	RESTART	Refresh
identifier	- 2 +	RESTART	Refresh
webcat	- 1 +	RESTART	Refresh

Figure 5: Swarm UI displaying a running pipeline



### 1.3.4 Integrator UI

Most components, like Spark, Flink and HDFS provide dashboards to monitor their status. As each of the components runs in a separate Docker container, they all have a different IP address. Moreover, the ports and the paths on which the dashboards are available differ per technology. That is the reason why the BDE platform provides an integrator UI. This GUI displays each of the component's dashboards in a unified interface topped with a BDE style based on Google's Material Design<sup>4</sup>.

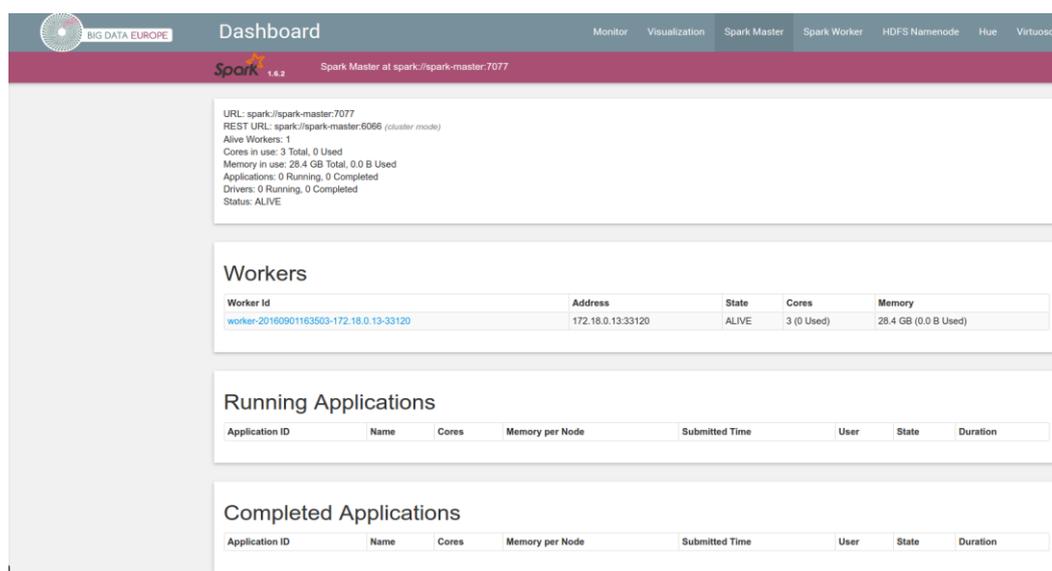


Figure 6: BDE Integrator UI (Dashboard)

## 2. Conclusion and Future Work

The platform architecture mainly focuses on two aspects: ease-of-use and flexibility. A user should be able to develop and deploy a Big Data pipeline with little effort. At the same time, the platform needs to be flexible to embrace future changes in the fast moving space of Big Data. Compared to the platform architecture presented in D3.3, the platform has been subject to changes. The removal of Mesos is the most important one. Docker Swarm is now used as the only tool to deploy applications on the platform.

At the moment, the existing platform is being tested by the use cases for the pilot pipelines, any lessons learnt from the practical deployment of the platform for the pilots cases will be reflected the next release of the platform

<sup>4</sup> See <https://material.google.com/>



## 3. References

- [1]. Big Data Europe, “WP3 Deliverable 3.4: Assessment on Application of Generic Data Management Technologies II,” Big Data Europe, 2016.