A WEB APPLICATION FOR INTERACTIVE DATA ANALYSIS WITH SPARK

Romain Rigaux

Spark Summit, Jul 1, 2014

cloudera

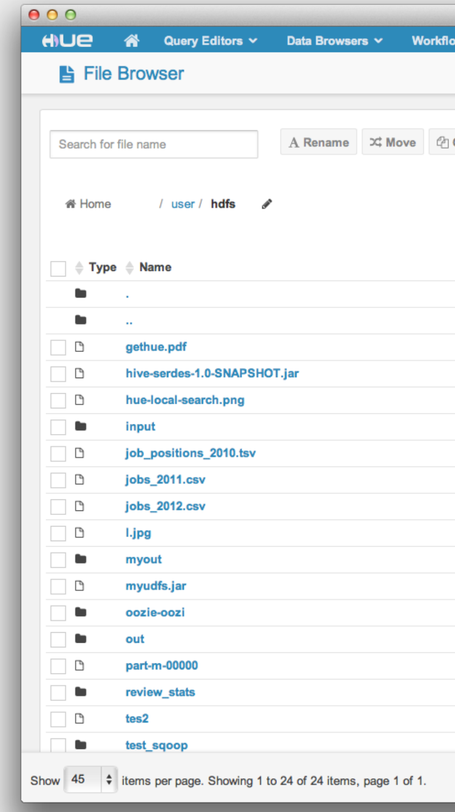
GOAL OF HUE

WEB INTERFACE FOR ANALYZING DATA WITH APACHE HADOOP

SIMPLIFY AND INTEGRATE

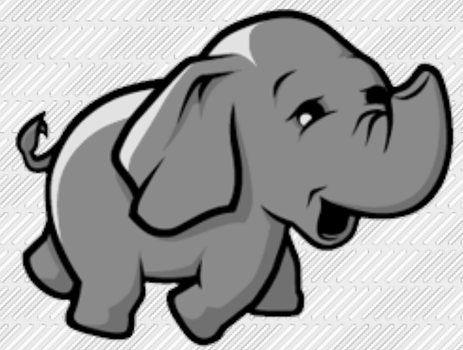
FREE AND OPEN SOURCE

—> OPEN UP BIG DATA

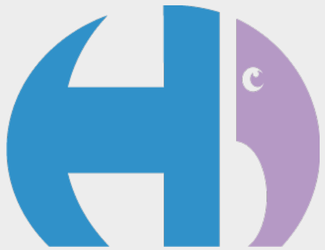


cloudera

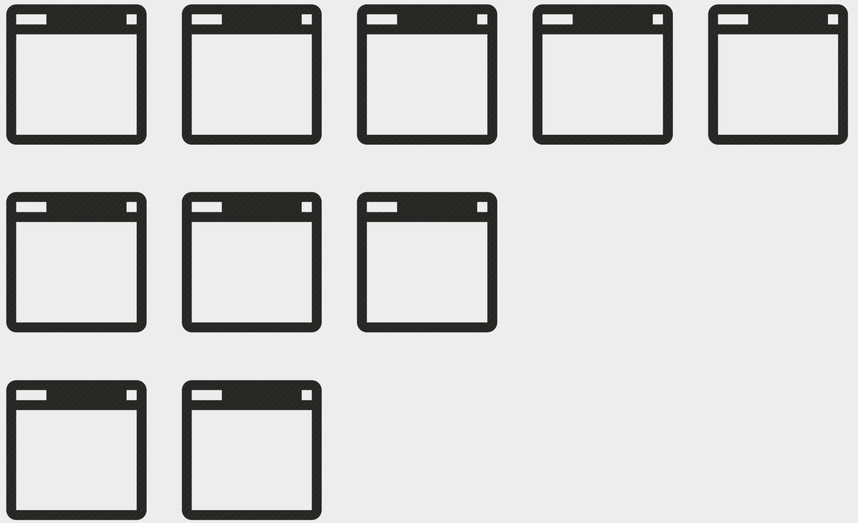
VIEW FROM 30K FEET



Hadoop



Web Server

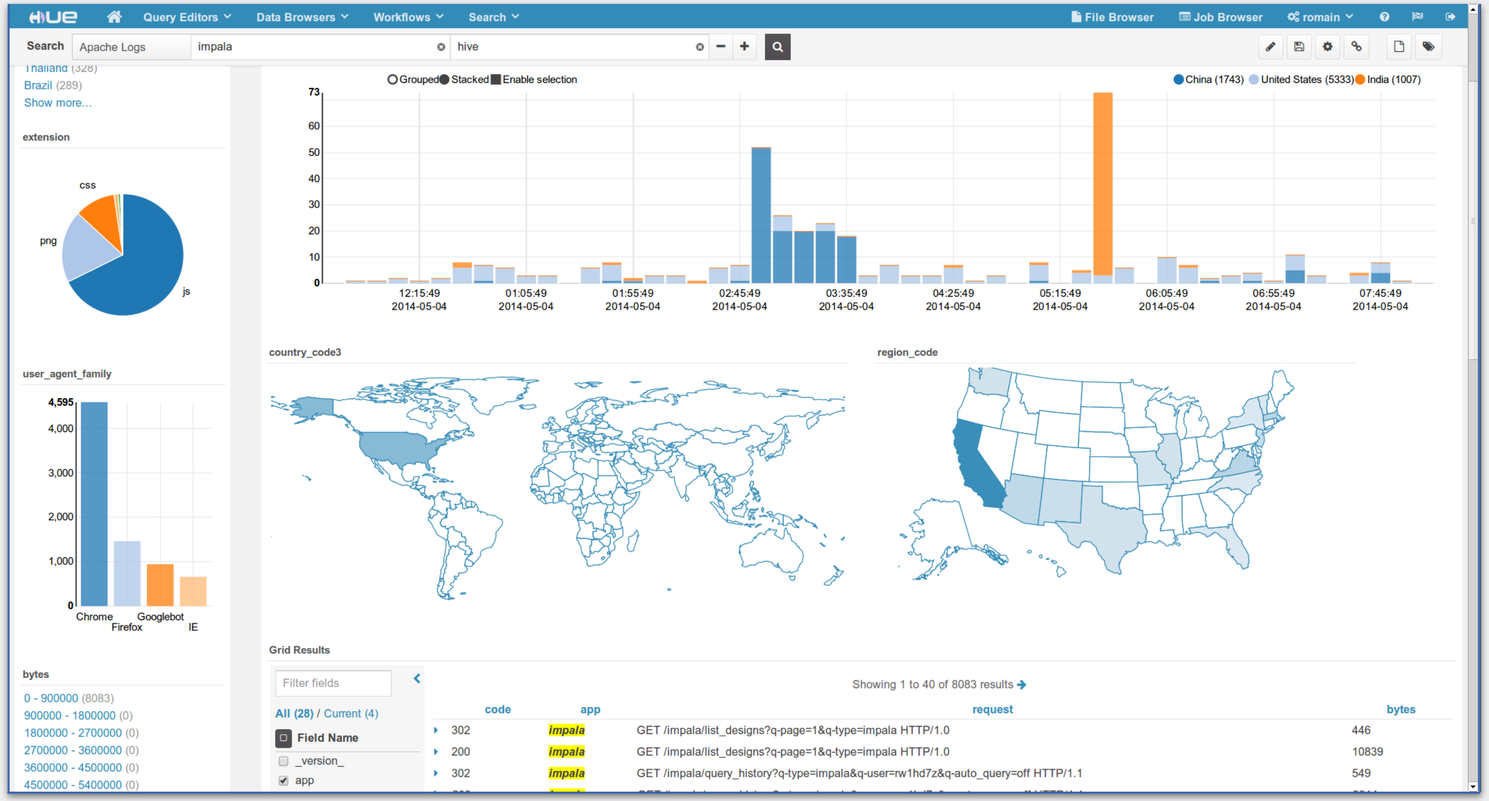


You, your colleagues and even that friend that uses IE9 ;)

LATEST HUE

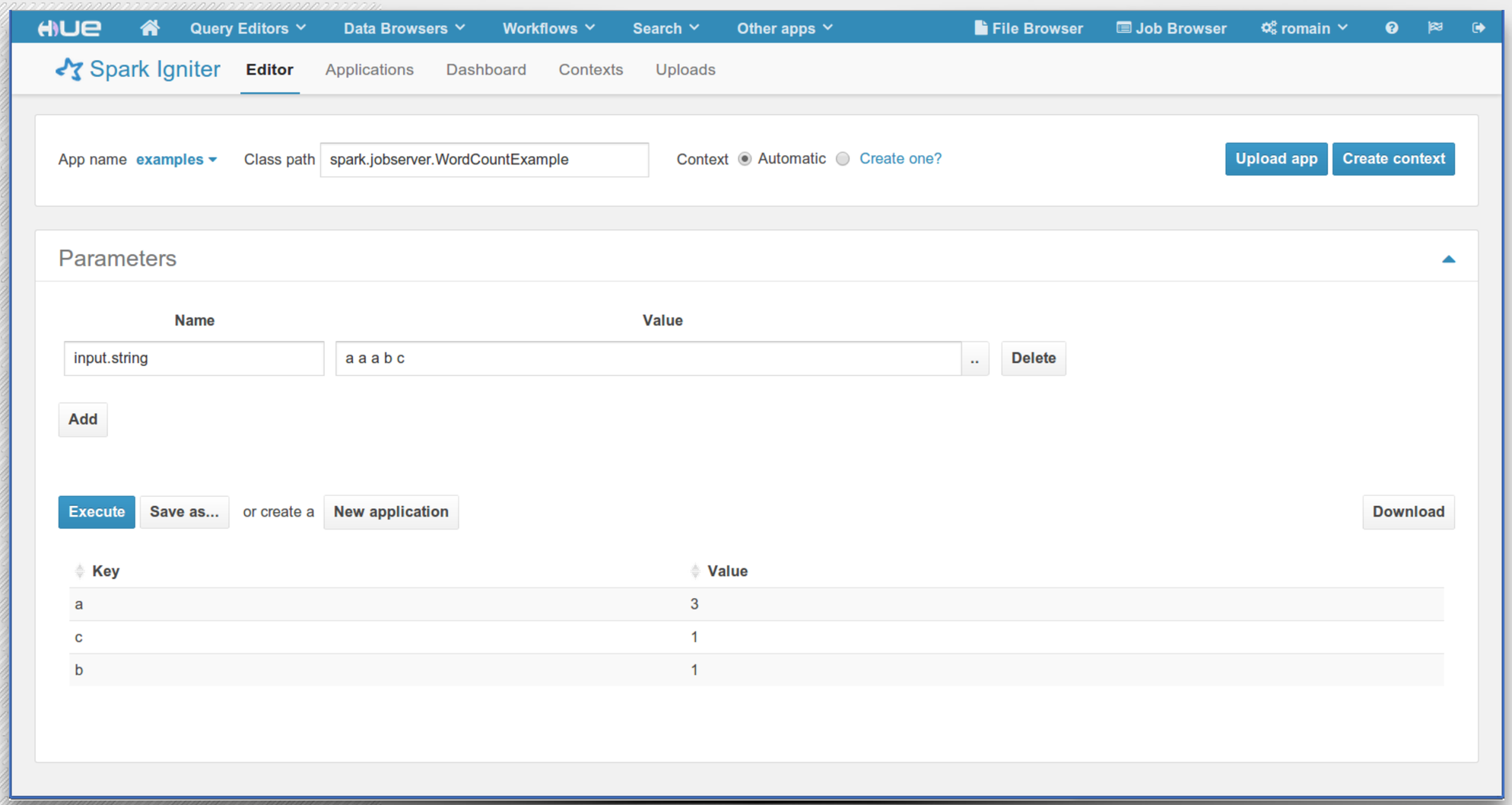
HUE 3.6+

Where we are now, a brand new way to search and explore your data.



cloudera

SPARK IGNITER



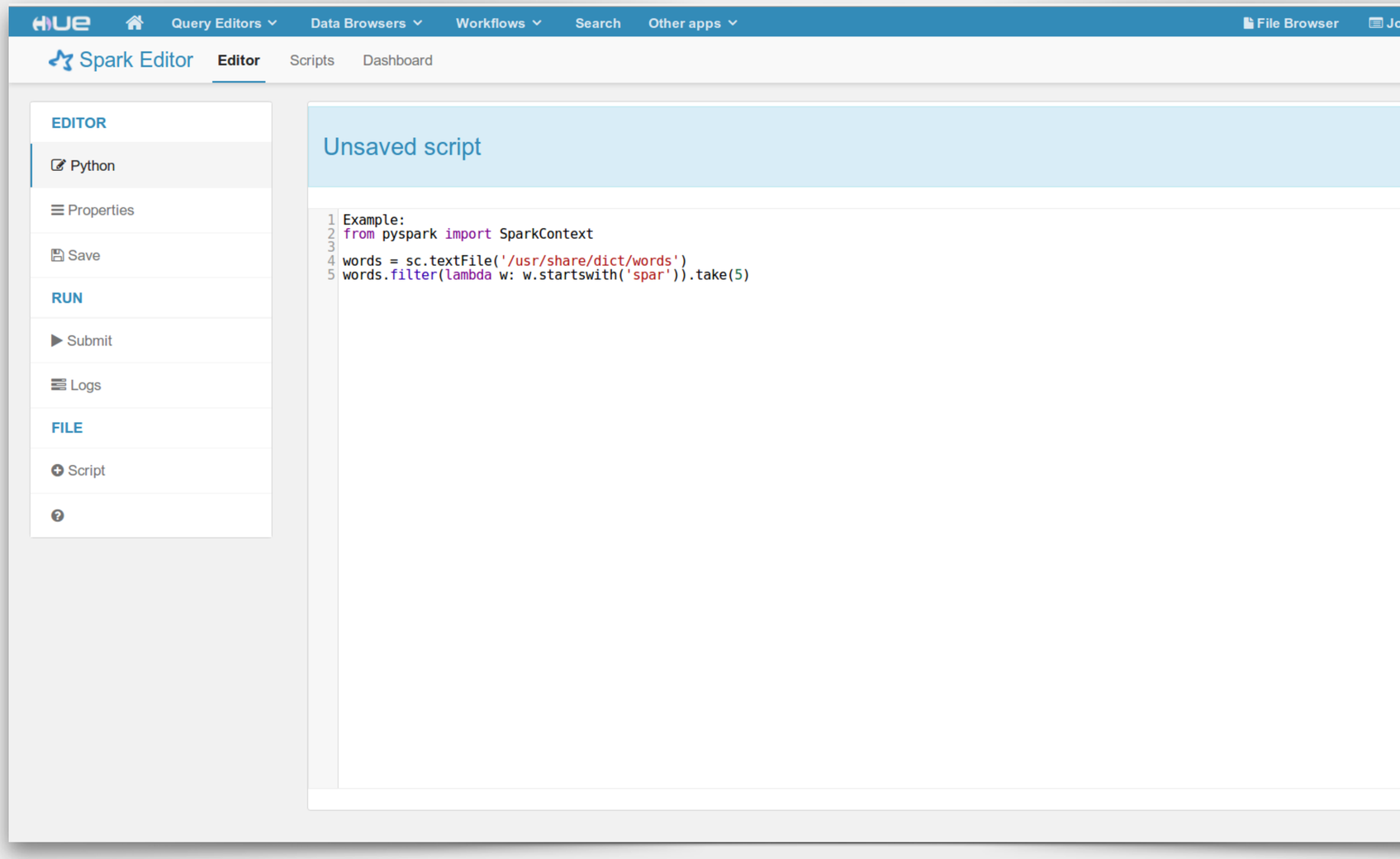
cloudera

HISTORY

OCT 2013

Submit through Oozie

Shell like for Java, Scala, Python



cloudera

HISTORY

JAN 2014

V2 Spark Igniter

Spark 0.8

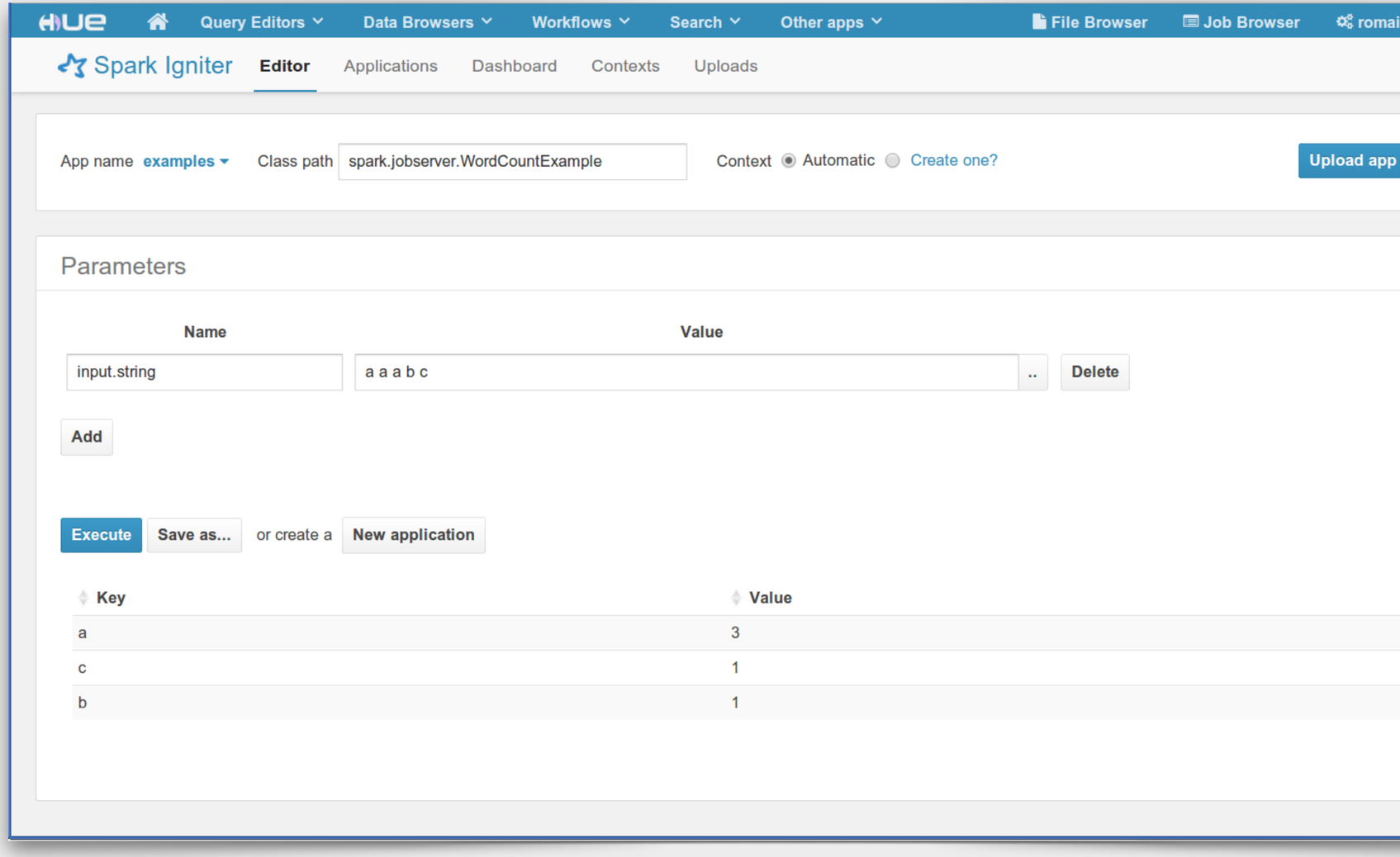
Java, Scala with Spark Job Server

APR 2014

Spark 0.9

JUN 2014

Ironing + How to deploy

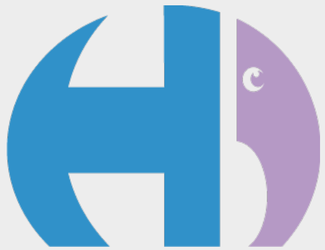


cloudera

“JUST A VIEW” ON TOP OF SPARK

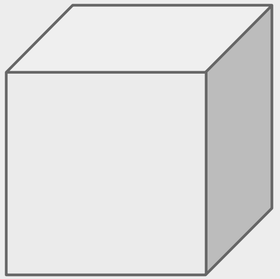
Saved script metadata

eg. name, args, classname, jar name…



Hue

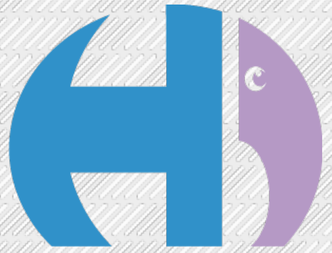
submit
list apps
list jobs
list contexts



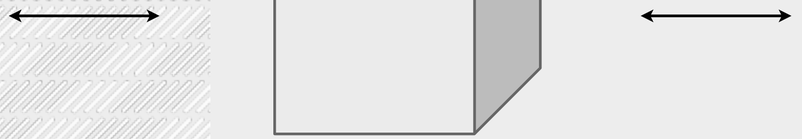
Job Server

cloudera

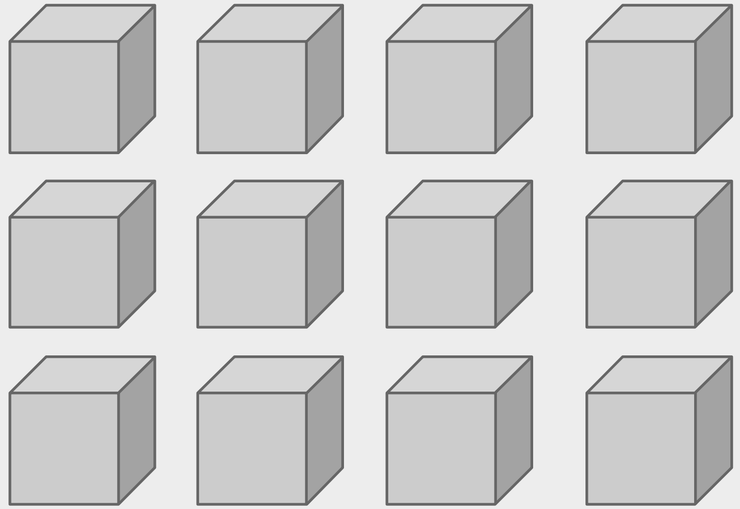
HOW TO TALK TO SPARK?



Hue

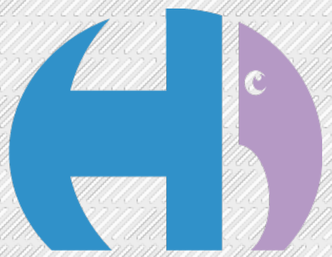


Spark Job Server

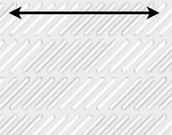


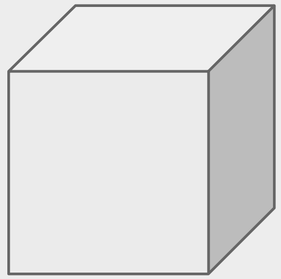
Spark

APP LIFE CYCLE

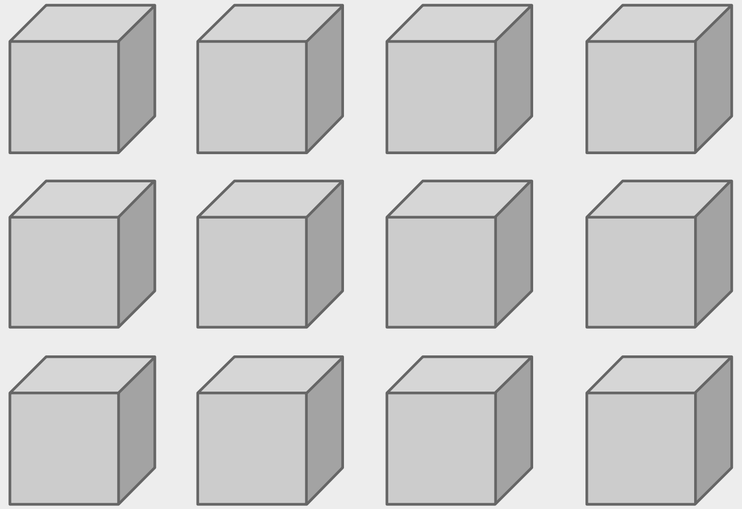


Hue





Spark Job Server



Spark

APP LIFE CYCLE

```scala
package spark.jobserver

import com.typesafe.config.{Config, ConfigFactory}
import org.apache.spark._
import scala.util.Try
import java.util.{Random, Date}

/**
 * A long job for stress tests purpose.
 * Iterative and randomized algorithm to compute Pi.
 * Imagine a square centered at (1,1) of length 2 units.
 * It tightly encloses a circle centered also at (1,1) of radius 1 unit.
 * Randomly throw darts to them.
 * We can use the ratio of darts inside the square and circle to approximate the Pi.
 *
 * stress.test.longpijob.duration controls how long it run in seconds.
 * Longer duration increases precision.
 *
 */
object LongPiJob extends SparkJob {
  private val rand = new Random(now)

  def main(args: Array[String]) {
    val sc = new SparkContext("local[4]", "LongPiJob")
    val config = ConfigFactory.parseString("")
    val results = runJob(sc, config)
    println("Result is " + results)
  }

  override def validate(sc: SparkContext, config: Config): SparkJobValidation = {
    SparkJobValid
  }

  override def runJob(sc: SparkContext, config: Config): Any = {
    val duration = Try(config.getInt("stress.test.longpijob.duration")).getOrElse(5)
    var hit:Long = 0
    var total:Long = 0
```

… extend SparkJob

.scala

sbt _/package

JAR

Upload

cloudera

APP LIFE CYCLE

```scala
package spark.jobserver

import com.typesafe.config.{Config, ConfigFactory}
import org.apache.spark._
import scala.util.Try
import java.util.{Random, Date}

/**
 * A Long job for stress tests purpose.
 * Iterative and randomized algorithm to compute Pi.
 * Imagine a square centered at (1,1) of length 2 units.
 * It tightly encloses a circle centered also at (1,1) of radius 1 unit.
 * Randomly throw darts to them.
 * We can use the ratio of darts inside the square and circle to approximate the Pi.
 *
 * stress.test.longpijob.duration controls how long it run in seconds.
 * Longer duration increases precision.
 *
 */
object LongPiJob extends SparkJob {
  private val rand = new Random(now)

  def main(args: Array[String]) {
    val sc = new SparkContext("local[4]", "LongPiJob")
    val config = ConfigFactory.parseString("")
    val results = runJob(sc, config)
    println("Result is " + results)
  }

  override def validate(sc: SparkContext, config: Config): SparkJobValidation = {
    SparkJobValid
  }

  override def runJob(sc: SparkContext, config: Config): Any = {
    val duration = Try(config.getInt("stress.test.longpijob.duration")).getOrElse(5)
    var hit:Long = 0
    var total:Long = 0
```

… extend SparkJob

.scala

sbt _/package

JAR

Upload

Context

create context: auto or manual

cloudera

SPARK JOB SERVER

WHERE

https://github.com/ooyala/spark-jobserver

WHAT

REST job server for Spark

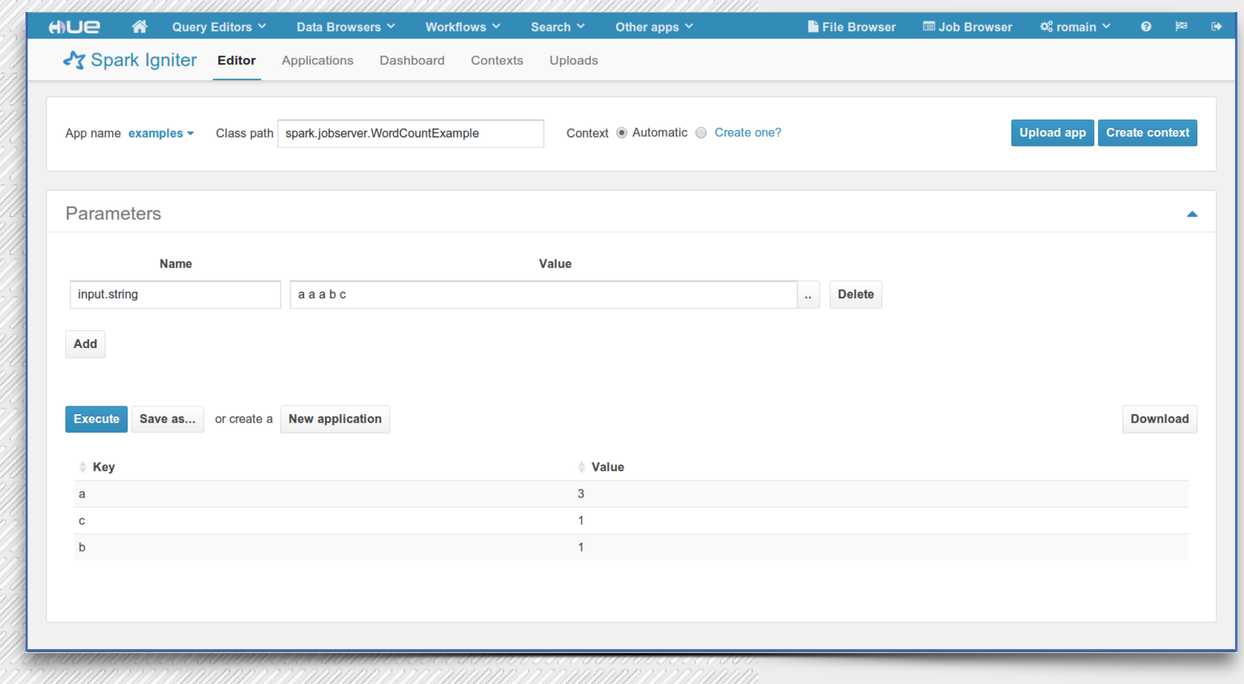
WHEN

Spark Summit talk Monday 5:45pm:
Spark Job Server: Easy Spark Job Management by Ooyala

```
curl -d "input.string = a b c a b see" 'localhost:8090/jobs?
appName=test&classPath=spark.jobserver.WordCountExample'
{
  "status": "STARTED",
  "result": {
    "jobId": "5453779a-f004-45fc-a11d-a39dae0f9bf4",
    "context": "b7ea0eb5-spark.jobserver.WordCountExample"
  }
}
```

cloudera

FOCUS ON UX



VS

```
curl -d "input.string = a b c a b see" 'localhost:8090/jobs?
appName=test&classPath=spark.jobserver.WordCountExample'
{
  "status": "STARTED",
  "result": {
    "jobId": "5453779a-f004-45fc-a11d-a39dae0f9bf4",
    "context": "b7ea0eb5-spark.jobserver.WordCountExample"
  }
}
```

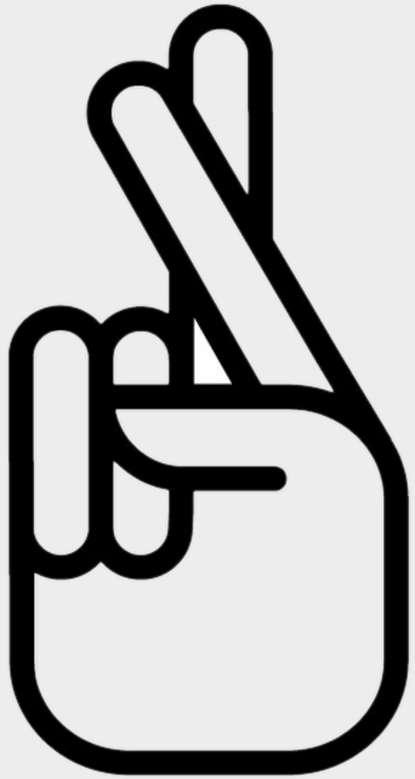
cloudera

TRAIT SPARKJOB

```
/**
*  This trait is the main API for Spark jobs submitted to the Job Server.
*/
trait SparkJob {
  /**
  * This is the entry point for a Spark Job Server to execute Spark jobs.
  * */
  def runJob(sc: SparkContext, jobConfig: Config): Any

  /**
  * This method is called by the job server to allow jobs to validate their input and reject
  * invalid job requests.  */
  def validate(sc: SparkContext, config: Config): SparkJobValidation
}
```

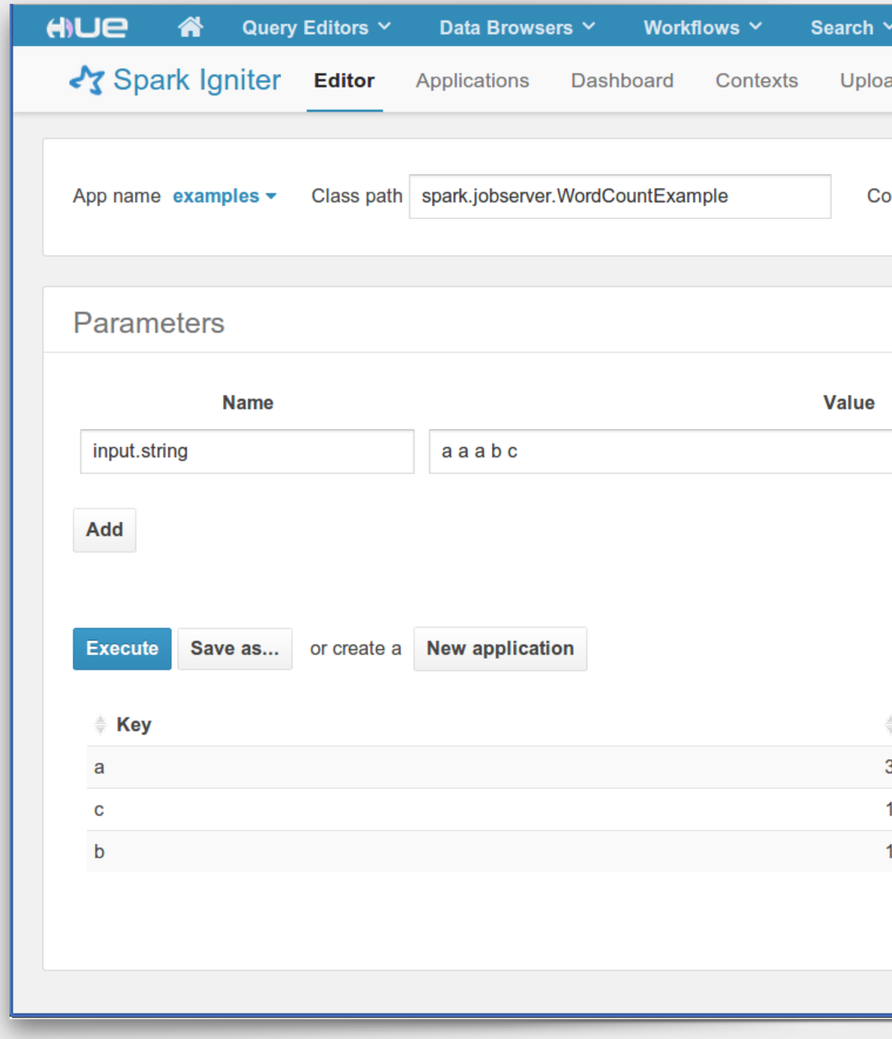

DEMO TIME



cloudera

LIVE DEMO

demo.gethue.com/spark



cloudera

CURRENT TECH
SUM-UP

STANDALONE APP

SCALA 2.10

SPARK 0.9

HUE C5+

cloudera

ROADMAP

WHAT

- **YARN**
- HUE-2134 [spark] App revamp and Job Server needs
  - Impersonation
  - Status report
  - Fetch N from result set
  - Python?
- Full Hue integration with HDFS, JobBrowser, Hive, charts…
- On the fly compile of Scala, Java?
- ?

cloudera

THANK YOU!

WEBSITE

http://gethue.com

LEARN

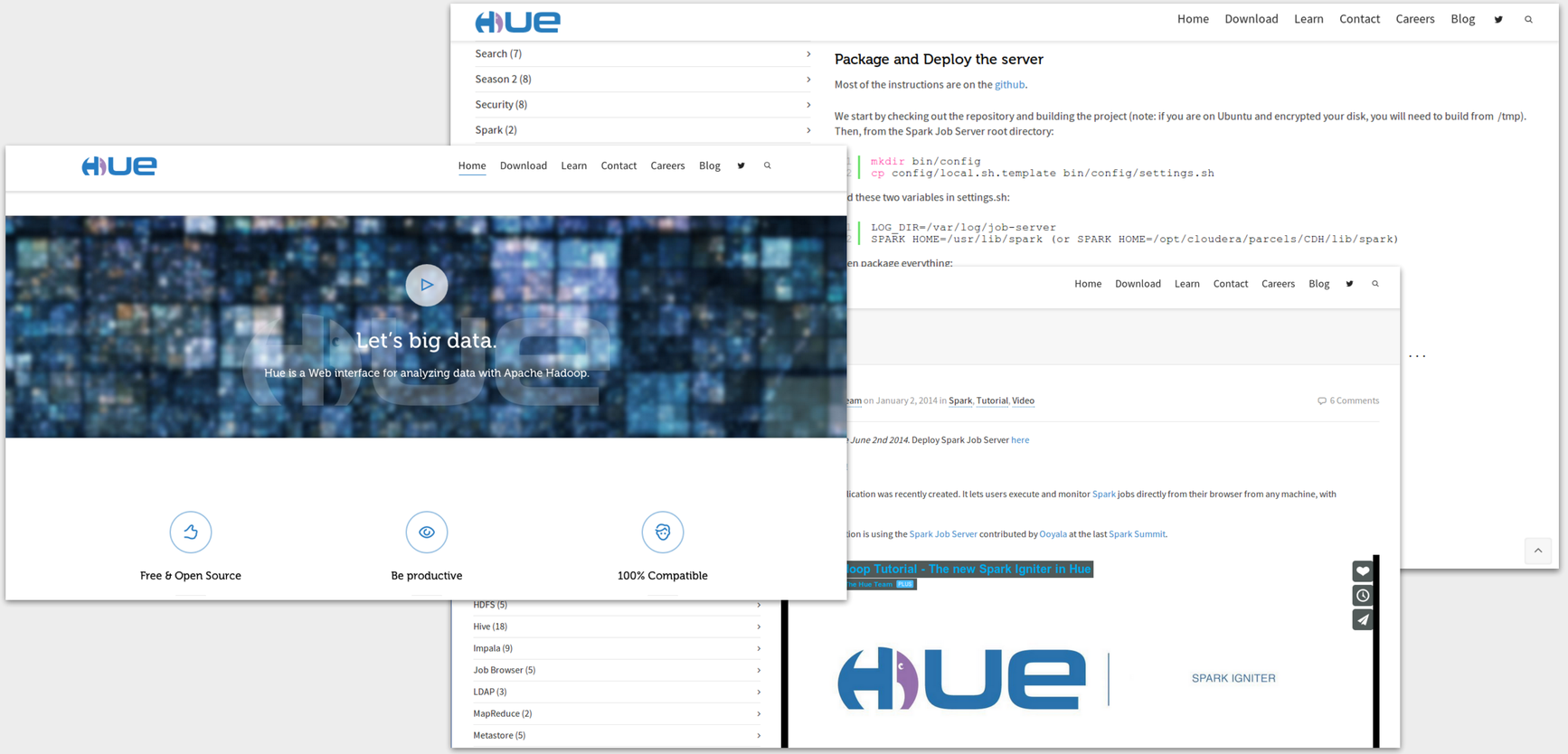
http://gethue.com/category/spark/

TWITTER

@gethue

USER GROUP

hue-user@



http://gethue.com/get-started-with-spark-deploy-spark-server-and-compute-pi-from-your-web-browser/

cloudera