package PackageDemo;

import java.io.IOException;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.LongWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Job;

import org.apache.hadoop.mapreduce.Mapper;

import org.apache.hadoop.mapreduce.Reducer;

import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;

import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

import org.apache.hadoop.util.GenericOptionsParser;

public class WordCount {

public static void main(String [] args) throws Exception

{

Configuration c=new Configuration();

String[] files=new GenericOptionsParser(c,args).getRemainingArgs();

Path input=new Path(files[0]);

Path output=new Path(files[1]);

Job j=new Job(c,"wordcount");

j.setJarByClass(WordCount.class);

j.setMapperClass(MapForWordCount.class);

j.setReducerClass(ReduceForWordCount.class);

j.setOutputKeyClass(Text.class);

j.setOutputValueClass(IntWritable.class);

FileInputFormat.addInputPath(j, input);

FileOutputFormat.setOutputPath(j, output);

System.exit(j.waitForCompletion(true)?0:1);

}

public static class MapForWordCount extends Mapper<LongWritable, Text, Text, IntWritable>{

public void map(LongWritable key, Text value, Context con) throws IOException, InterruptedException

{

String line = value.toString();

String[] words=line.split(",");

for(String word: words )

{

Text outputKey = new Text(word.toUpperCase().trim());

IntWritable outputValue = new IntWritable(1);

con.write(outputKey, outputValue);

}

}

}

public static class ReduceForWordCount extends Reducer<Text, IntWritable, Text, IntWritable>

{

public void reduce(Text word, Iterable<IntWritable> values, Context con) throws IOException, InterruptedException

{

int sum = 0;

for(IntWritable value : values)

{

sum += value.get();

}

con.write(word, new IntWritable(sum));

}

}

}

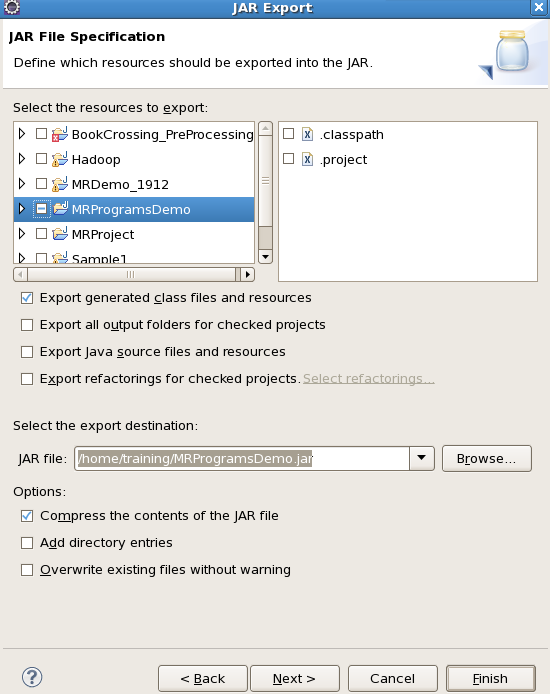
**Explanation**

The program consist of 3 classes:

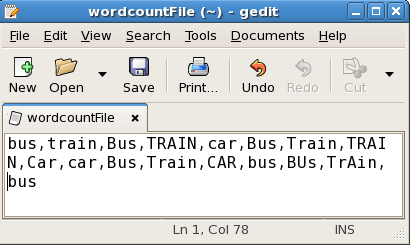
* Driver class (Public void static main- the entry point)
* Map class which **extends** public class Mapper<KEYIN,VALUEIN,KEYOUT,VALUEOUT>  and implements the Map function.
* Reduce class which extends public class Reducer<KEYIN,VALUEIN,KEYOUT,VALUEOUT> and implements the Reduce function.

Step 6. Make Jar File

Right Click on Project> Export> Select export destination as **Jar File**  > next> Finish



Step 7: Take a text file and move it in HDFS



To Move this into Hadoop directly, open the terminal and enter the following commands:

[training@localhost ~]$ hadoop fs -put wordcountFile wordCountFile

Step 8 . Run Jar file

*(hadoop jar jarfilename.jar packageName.ClassName  PathToInputTextFile PathToOutputDirectry)*

[training@localhost ~]$ hadoop jar MRProgramsDemo.jar PackageDemo.WordCount wordCountFile MRDir1

Step 9. Open Result

[training@localhost ~]$ hadoop fs -ls MRDir1

Found 3 items

-rw-r--r-- 1 training supergroup 0 2016-02-23 03:36 /user/training/MRDir1/\_SUCCESS

drwxr-xr-x - training supergroup 0 2016-02-23 03:36 /user/training/MRDir1/\_logs

-rw-r--r-- 1 training supergroup 20 2016-02-23 03:36 /user/training/MRDir1/part-r-00000

[training@localhost ~]$ hadoop fs -cat MRDir1/part-r-00000

BUS 7

CAR 4

TRAIN 6

[Hortonworks Sandbox](https://dzone.com/go?i=285441&u=https%3A%2F%2Fhortonworks.com%2Fproducts%2Fsandbox%2F%3Futm_campaign%3Ddzonepre%2Fpostroll%26utm_medium%3Ddisplay%26apos%3B%26utm_source%3Ddzone%26utm_id%3D2216633) for HDP and HDF is your chance to get started on learning, developing, testing and trying out new features. Each [download](https://dzone.com/go?i=285441&u=https%3A%2F%2Fhortonworks.com%2Fproducts%2Fsandbox%2F%3Futm_campaign%3Ddzonepre%2Fpostroll%26utm_medium%3Ddisplay%26apos%3B%26utm_source%3Ddzone%26utm_id%3D2216633) comes preconfigured with interactive tutorials, sample data and developments from the Apache community.

**Like This Article? Read More From DZone**

[Four MapReduce Design Patterns](https://dzone.com/articles/mapreduce-design-patterns?fromrel=true)

[Evolving Technologies for Driving Big Data](https://dzone.com/articles/evolving-technologies-for-driving-big-data?fromrel=true)

[BigTable Model with Cassandra and HBase](https://dzone.com/articles/bigtable-model-cassandra-and?fromrel=true)

[](https://dzone.com/refcardz/apache-spark?fromrel=true)

[Free DZone Refcard](https://dzone.com/refcardz/apache-spark?fromrel=true)

[Apache Spark](https://dzone.com/refcardz/apache-spark?fromrel=true)

Topics:

mapreduce ,haddop ,wordcount ,java

Like [(27)](https://dzone.com/articles/word-count-hello-word-program-in-mapreduce)

Comment (2)

Save

[Tweet](https://dzone.com/articles/word-count-hello-word-program-in-mapreduce)

114.6k Views

Opinions expressed by DZone contributors are their own.

**Big Data Partner Resources**

[TDWI Checklist Report](https://hortonworks.com/info/tdwi-checklist/)

Hortonworks

[Data Lake vs. Data Warehouse](https://hortonworks.com/info/data-lake-vs-data-warehouse/)

Hortonworks

[GDPR poses challenges to enterprises that need to manage data, but a comprehensive framework can help. Learn how.](https://hortonworks.com/info/path-gdpr-compliance-begins-data-governance/?utm_campaign=dzonegdprwp&utm_medium=3rd-party-resource&utm_source=dzone&utm_id=2217878)

Hortonworks

Let's be friends:

**Beginner's Guide to Data and Analytics**

**Analytics has become such a multi-faceted field that requires some basic knowledge. You don't need to spend years learning what data and analytics are, though.**

[](https://dzone.com/users/3017779/shelbyblitz.html)by

[Shelby Blitz](https://dzone.com/users/3017779/shelbyblitz.html)

·

Apr 06, 18 · [Big Data Zone](https://dzone.com/big-data-analytics-tutorials-tools-news) · Opinion

Like [(1)](https://dzone.com/articles/word-count-hello-word-program-in-mapreduce)

Comment (0)

Save

[Tweet](https://dzone.com/articles/word-count-hello-word-program-in-mapreduce)

249 Views

Join the DZone community and get the full member experience.

[Join For Free](https://dzone.com/articles/word-count-hello-word-program-in-mapreduce)

It's becoming quite common to find articles and think-pieces that tout the countless benefits and advantages big data analytics offer companies and organizations across all industries. What isn't as common is a clear idea of what data and analytics are. For most people, these two concepts were forgotten shortly after university and are used only tangentially. However, it's becoming quite clear that to be truly successful and find the best ways to constantly maintain an edge, understanding analytics is vital.

That still leaves the question of what exactly these concepts are. Data may seem like a simple concept but is nevertheless accompanied by several ideas that can make the difference between success and failure. Similarly, analytics has become such a multi-faceted field that requires some basic knowledge. You don't need to spend years learning what data and analytics are, though. Starting from the beginning can help paint a much clearer picture.

**What Is Data?**

In its simplest form, data is just information. Everything companies and consumers do online and even offline creates information. This information can be incredibly specific — including demographic data, behavioral information, and other personal details — or can relate to larger groups as an aggregate.

Major corporations like Facebook and Google can gather data from even the most seemingly innocuous activities online and use it to better optimize their services. This new massive stream of information has led to the creation of the field of big data, which focuses on datasets (groupings of information) that are simply too large to analyze with traditional systems.

Data by itself doesn't reveal much, and especially when viewed without context or method. To really understand what data is saying, there must be a method to clean it, organize it, and interpret it better — and this is where analytics comes into the picture.

**What Is Analytics?**

Analytics is the way we turn thousands of data points into meaningful insights that can be applied to business processes. Today, analytics has outgrown the academic sphere to play a central role in most companies' development and growth strategies. This is because analytics helps find patterns in data that can highlight areas for improvement, successful tactics, and even show potential trends for the future.

The field of analytics is an umbrella term for a variety of more specific areas, and it can be applied to almost any industry today with success thanks to advanced data analytics software. Moreover, modern technology like [machine learning](https://www.sisense.com/blog/beyond-hype-machine-learning-unlocking-power-bi/) and [AI](https://www.sisense.com/blog/predictive-analytics-ai-can-give-competitive-edge/) have made analytics a much more accessible field and have expanded its uses outside of statistics.

**So, How Do Analytics and Data Help Businesses?**

Analytics needs data to work, and data that hasn't been sorted and analyzed is not very helpful. Data needs to be properly interpreted and organized to be useful, a process that includes removing data that isn't useful (scrubbing the data), organizing it into more logical groupings, and connecting data points to find patterns, insights, and useful information that might hint at future tendencies.

For most businesses, collecting data happens throughout several points in their production and value chains. [Retailers](https://www.sisense.com/blog/business-intelligence-case-study-retail-better-brand/), for example, collect data from suppliers, their own warehouses, points of sale, customer surveys, and even online visitors to their websites. All this information provides different insights and can be used for a variety of reasons including better marketing strategies, tailored promotions, value chain improvements, and operational fixes. Data analytics helps companies understand themselves better and lets them create better strategies based on their findings by turning data into actionable insights.

There are several key areas where implementing a regimen of analytics can be a major win for businesses. One of the first areas many companies look at with data analytics is inside their own operations. Companies consistently set goals, milestones, and objectives to reach, but tracking them isn't always easy. By focusing on the data teams and business segments produce, it's easy to create key performance indicators and have clear, tangible ways to measure them.

Similarly, analytics can significantly improve customer relations and support. Service, retention, and engagement are all based on the idea of understanding not just what customers want at a specific moment, but what they may want in the future. Using predictive analytics and visualizations, companies can see the areas where customers are satisfied, areas where they can improve, and more importantly, where these preferences may lead in the future. [Marketing](https://www.sisense.com/blog/top-benefits-business-intelligence-marketing/) can also be greatly enhanced thanks to analytics. It's much easier to track success by measuring correlated data such as conversions, click-throughs, interactions on social media, and other similar measures before implementing findings in future campaigns.

It doesn't require a Ph.D. in statistics to dive into analytics. By finding the right tools and platforms to analyze and better interpret the data your company produces, you can start making smarter decisions to get a let up on your competition.

**Like This Article? Read More From DZone**

[Embedded Analytics Are Powerful for Apps: Why Do You Need Them?](https://dzone.com/articles/embedded-analytics-are-powerful-for-apps-why-do-yo?fromrel=true)

[Demystifying the Intricacies of Advanced Data Visualization](https://dzone.com/articles/demystifying-the-intricacies-of-advanced-data-visu?fromrel=true)

[D3 Visualization: Take Your Dashboards to Another Level](https://dzone.com/articles/d3-visualization-take-your-dashboards-to-another-l?fromrel=true)

[](https://dzone.com/refcardz/apache-spark?fromrel=true)

[Free DZone Refcard](https://dzone.com/refcardz/apache-spark?fromrel=true)

[Apache Spark](https://dzone.com/refcardz/apache-spark?fromrel=true)

Topics:

big data,data analytics,data visualization,predictive analytics,business intelligence

Like [(1)](https://dzone.com/articles/word-count-hello-word-program-in-mapreduce)

Comment (0)

Save

[Tweet](https://dzone.com/articles/word-count-hello-word-program-in-mapreduce)

249 Views

Published at DZone with permission of Shelby Blitz , DZone MVB. [See the original article here.](https://www.sisense.com/blog/beginners-guide-to-data-and-analytics/?0)

Opinions expressed by DZone contributors are their own.

Let's be friends:

[](https://www.hortonworks.com/?utm_campaign=dzonelogo&utm_medium=display&utm_source=dzone&utm_id=2217877)

**Apache Ignite Cluster Together With Spring Boot**

**Spring Boot is taking the Java world by storm, so let's take a look at how it's helping make the Big Data world a bit more accessible!**

[](https://dzone.com/users/2918001/piotrminkowski.html)by

[Piotr Mińkowski](https://dzone.com/users/2918001/piotrminkowski.html)

·

Apr 06, 18 · [Big Data Zone](https://dzone.com/big-data-analytics-tutorials-tools-news) · Tutorial

Like [(1)](https://dzone.com/articles/word-count-hello-word-program-in-mapreduce)

Comment (0)

Save

[Tweet](https://dzone.com/articles/word-count-hello-word-program-in-mapreduce)

1,047 Views

Join the DZone community and get the full member experience.

[Join For Free](https://dzone.com/articles/word-count-hello-word-program-in-mapreduce)

[Hortonworks Sandbox](https://dzone.com/go?i=285437&u=https%3A%2F%2Fhortonworks.com%2Fproducts%2Fsandbox%2F%3Futm_campaign%3Ddzonepre%2Fpostroll%26utm_medium%3Ddisplay%26apos%3B%26utm_source%3Ddzone%26utm_id%3D2216633) for HDP and HDF is your chance to get started on learning, developing, testing and trying out new features. Each [download](https://dzone.com/go?i=285437&u=https%3A%2F%2Fhortonworks.com%2Fproducts%2Fsandbox%2F%3Futm_campaign%3Ddzonepre%2Fpostroll%26utm_medium%3Ddisplay%26apos%3B%26utm_source%3Ddzone%26utm_id%3D2216633) comes preconfigured with interactive tutorials, sample data and developments from the Apache community.

I have already introduced Apache Ignite in one of my previous articles, [In-Memory Data Grid With Apache Ignite](https://dzone.com/articles/in-memory-data-grid-with-apache-ignite). Apache Ignite can be easily launched locally together with a Spring Boot application. The only thing we have to do is include the artifact org.apache.ignite:ignite-spring-data in the project dependencies and then declare the Ignite instance @Bean. A sample @Bean declaration is below.

@Bean

public Ignite igniteInstance() {

IgniteConfiguration cfg = new IgniteConfiguration();

cfg.setIgniteInstanceName("ignite-cluster-node");

CacheConfiguration ccfg1 = new CacheConfiguration("PersonCache");

ccfg1.setIndexedTypes(Long.class, Person.class);

CacheConfiguration ccfg2 = new CacheConfiguration("ContactCache");

ccfg2.setIndexedTypes(Long.class, Contact.class);

cfg.setCacheConfiguration(ccfg1, ccfg2);

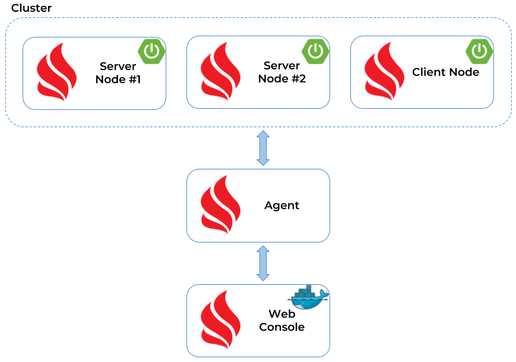
IgniteLogger log = new Slf4jLogger();

cfg.setGridLogger(log);

return Ignition.start(cfg);

}

In this article, I would like to show you a little more advanced sample where we will start multiple Ignite nodes inside the cluster, Ignite's web console for monitoring the cluster, and Ignite's agent for providing communication between the cluster's nodes and web console. Let's begin by looking at a picture with an architecture of our sample solution.



We have three nodes that are part of the cluster. If you carefully take a look at the picture illustrating an architecture, you have probably noticed that there are two nodes: a Server node and a Client node. By default, all Ignite nodes are started as Server nodes. The Client node needs to be explicitly enabled. Server nodes participate in caching, compute execution, and stream processing, while Client nodes provide the ability to connect to the servers remotely. However, they allow using the whole set of Ignite APIs, including near caching, transactions, compute, and streaming.

Here's Ignite's client instance @Bean declaration.

@Bean

public Ignite igniteInstance() {

IgniteConfiguration cfg = new IgniteConfiguration();

cfg.setIgniteInstanceName("ignite-cluster-node");

cfg.setClientMode(true);

CacheConfiguration ccfg1 = new CacheConfiguration("PersonCache");

ccfg1.setIndexedTypes(Long.class, Person.class);

CacheConfiguration ccfg2 = new CacheConfiguration("ContactCache");

ccfg2.setIndexedTypes(Long.class, Contact.class);

cfg.setCacheConfiguration(ccfg1, ccfg2);

return Ignition.start(cfg);

}

The fact is that we don't have to do anything more to make our nodes work together within the cluster. Every new node is automatically detected by all other cluster's nodes using multicast communication. When starting our sample application, we only have to guarantee that each instance's server would listen of a different port by overriding the server.port Spring Boot property. Here's command that starts the sample application, which is available on [GitHub](https://github.com/piomin/sample-ignite-jpa.git) under the [branch cluster](https://github.com/piomin/sample-ignite-jpa/tree/cluster). Each node exposes the same REST API, which may be easily tested using Swagger2 just by opening its dashboard available under address *http://localhost:port/swagger-ui.html*.

java -jar -Dserver.port=8901 -Xms512m -Xmx1024m -XX:+UseG1GC -XX:+DisableExplicitGC -XX:MaxDirectMemorySize=256m target/ignite-rest-service-1.0-SNAPSHOT.jar

If you have successfully started a new node, you should see similar information in your application logs.

>>> +----------------------------------------------------------------------+

>>> Ignite ver. 2.4.0#20180305-sha1:aa342270b13cc1f4713382a8eb23b2eb7edaa3a5

>>> +----------------------------------------------------------------------+

>>> OS name: Windows 10 10.0 amd64

>>> CPU(s): 4

>>> Heap: 1.0GB

>>> VM name: 14132@piomin

>>> Ignite instance name: ignite-cluster-node

>>> Local node [ID=9DB1296A-7EEC-4564-BAAD-14E5D4A3A08D, order=2, clientMode=false]

>>> Local node addresses: [piomin/0:0:0:0:0:0:0:1, piomin/127.0.0.1, piomin/192.168.1.102, piomin/192.168.116.1, /192.168.226.1, /192.168.99.1]

>>> Local ports: TCP:8082 TCP:10801 TCP:11212 TCP:47101 UDP:47400 TCP:47501

Let's move back for a moment to the source code of our sample application. I assume you have already cloned a given repository from GitHub. There are two Maven modules available. The module ignite-rest-service is responsible for starting Ignite's cluster node in the Server node, while ignite-client-service for starting node in the Client node. Because we run only a single instance of the Client node, we would not override its default port set inside the application.yml file. You can build the project using the mvn clean install command and then start with java -jar or just run the main class IgniteClientApplication from your IDE.

There is also a JUnit test class inside module ignite-client-service, which defines one test responsible for calling HTTP endpoints (POST /person, POST /contact) that puts data into Ignite's cache. This test performs two operations: it puts some data into Ignite's in-memory cluster by calling endpoints exposed by the Client node and then checks if that data has been propagated through the cluster by calling the GET /person/{id}/withContacts endpoint exposed by one of the selected server nodes.

public class TestCluster {

TestRestTemplate template = new TestRestTemplate();

Random r = new Random();

int[] clusterPorts = new int[] {8901, 8902};

@Test

public void testCluster() throws InterruptedException {

for (int i=0; i<1000; i++) {

Person p = template.postForObject("http://localhost:8090/person", createPerson(), Person.class);

Assert.notNull(p, "Create person failed");

Contact c1 = template.postForObject("http://localhost:8090/contact", createContact(p.getId(), 0), Contact.class);

Assert.notNull(c1, "Create contact failed");

Contact c2 = template.postForObject("http://localhost:8090/contact", createContact(p.getId(), 1), Contact.class);

Assert.notNull(c2, "Create contact failed");

Thread.sleep(10);

Person result = template.getForObject("http://localhost:{port}/person/{id}/withContacts", Person.class, clusterPorts[r.nextInt(2)], p.getId());

Assert.notNull(result, "Person not found");

Assert.notEmpty(result.getContacts(), "Contacts not found");

}

}

private Contact createContact(Long personId, int index) {

...

}

private Person createPerson() {

...

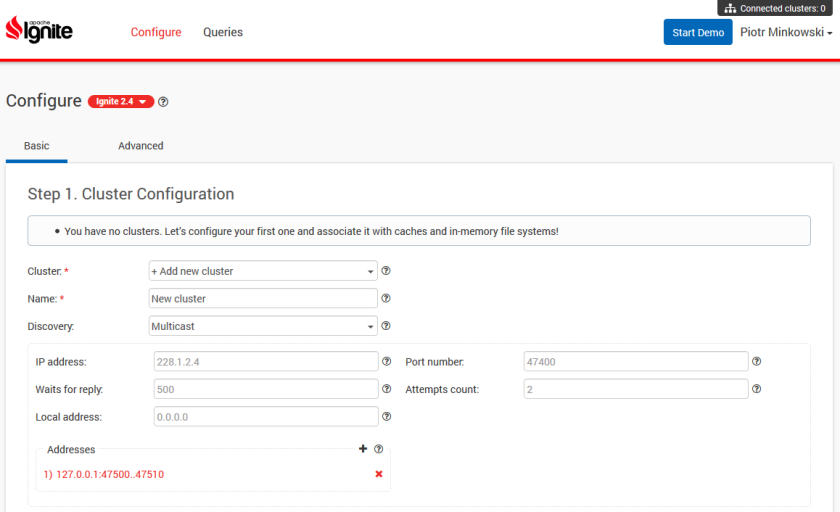
}

}

Before running any tests, we should launch two additional elements being a part of our architecture: Ignite's web console and agent. The most suitable way to run Ignite's web console on the local machine is through its Docker image *apacheignite/web-console-standalone*. Here's Docker command that starts Ignite's web console and exposes it on *port 80*. Because I run Docker on Windows, it is now available under the default VM address *http://192.168.99.100/*.

docker run -d -p 80:80 -p 3001:3001 -v /var/data:/var/lib/mongodb --name ignite-web-console apacheignite/web-console-standalone

In order to access it, you should first register your user. Although the mail server is not available on the Docker container, you will be logged in after it. You can configure your cluster using Ignite's web console and also run some SQL queries on that cluster. Of course, we still need to connect our cluster consisting of three nodes with the instance of web console started on Docker container. To achieve it, you have to download a web agent. It is probably not very intuitive, but you have to click **Start Demo**, which is located in the right corner of Ignite's web console. Then, you will be redirected to the download page, where you can accept the download of the *ignite-web-agent-2.4.0.zip* file, which contains all needed libraries and configuration to start the web agent locally.



After downloading and unpacking the web agent, go to its main directory and change property server-uri to *http://192.168.99.100* inside the *default.properties*file. Then, you may run the script *ignite-web-agent.bat* (or *.sh* if you are testing it on Linux), which starts the web agent. Unfortunately, it's not all that has to be done. Every server node's application should include the artifact ignite-rest-http in order to be able to communicate with the agent. It is responsible for exposing the HTTP endpoint that is accessed by a web agent. It is based on Jetty server, which causes some problems in conjunction with Spring Boot. Spring Boot sets default versions of Jetty libraries used inside the project. The problem is that ignite-rest-http requires older versions of that libraries, so we also have to override some default managed versions in the *pom.xml* file, according to the code below.

<dependencyManagement>

<dependencies>

<dependency>

<groupId>org.eclipse.jetty</groupId>

<artifactId>jetty-http</artifactId>

<version>9.2.11.v20150529</version>

</dependency>

<dependency>

<groupId>org.eclipse.jetty</groupId>

<artifactId>jetty-server</artifactId>

<version>9.2.11.v20150529</version>

</dependency>

<dependency>

<groupId>org.eclipse.jetty</groupId>

<artifactId>jetty-io</artifactId>

<version>9.2.11.v20150529</version>

</dependency>

<dependency>

<groupId>org.eclipse.jetty</groupId>

<artifactId>jetty-continuation</artifactId>

<version>9.2.11.v20150529</version>

</dependency>

<dependency>

<groupId>org.eclipse.jetty</groupId>

<artifactId>jetty-util</artifactId>

<version>9.2.11.v20150529</version>

</dependency>

<dependency>

<groupId>org.eclipse.jetty</groupId>

<artifactId>jetty-xml</artifactId>

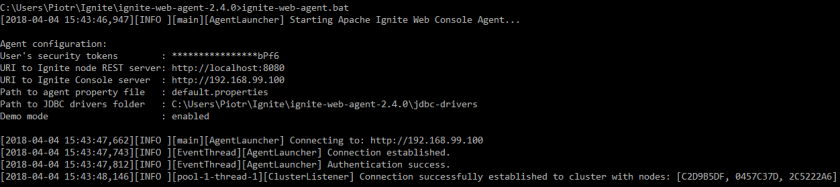
<version>9.2.11.v20150529</version>

</dependency>

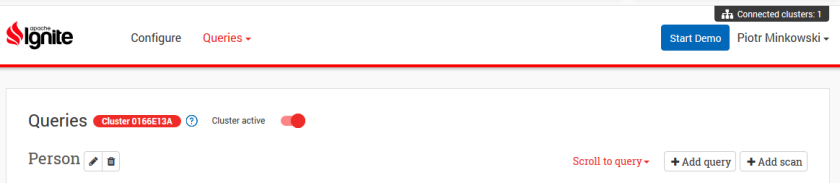
</dependencies>

</dependencyManagement>

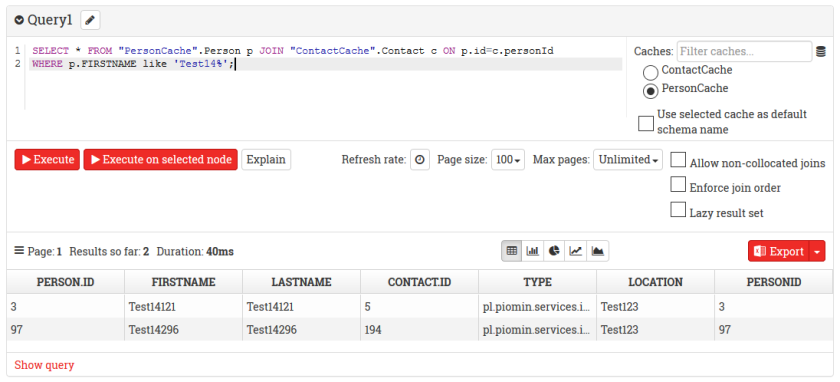
After implementing the changes described above, we may finally proceed to run all the elements that are part of our sample system. If you start the Ignite web agent locally, it should automatically detect all running cluster nodes. Here's the screen with the logs displayed by the agent after startup.



At the same time, you should see that a new cluster has been detected by the Ignite web console.



You can configure a new or a currently existing cluster using the web console or just run a test query on the selected managed cluster. You have to include a name of the cache as a prefix to the table name when defining a query.



Similar queries have to be declared inside a repository interface. Here are additional methods used for finding entities stored in PersonCache. If you would like to include results stored in other caches, you have to explicitly declare its name together with the table name.

@RepositoryConfig(cacheName = "PersonCache")

public interface PersonRepository extends IgniteRepository {

List findByFirstNameAndLastName(String firstName, String lastName);

@Query("SELECT p.id, p.firstName, p.lastName, c.id, c.type, c.location FROM Person p JOIN \"ContactCache\".Contact c ON p.id=c.personId WHERE p.id=?")

List<List> findByIdWithContacts(Long id);

@Query("SELECT c.\* FROM Person p JOIN \"ContactCache\".Contact c ON p.id=c.personId WHERE p.firstName=? and p.lastName=?")

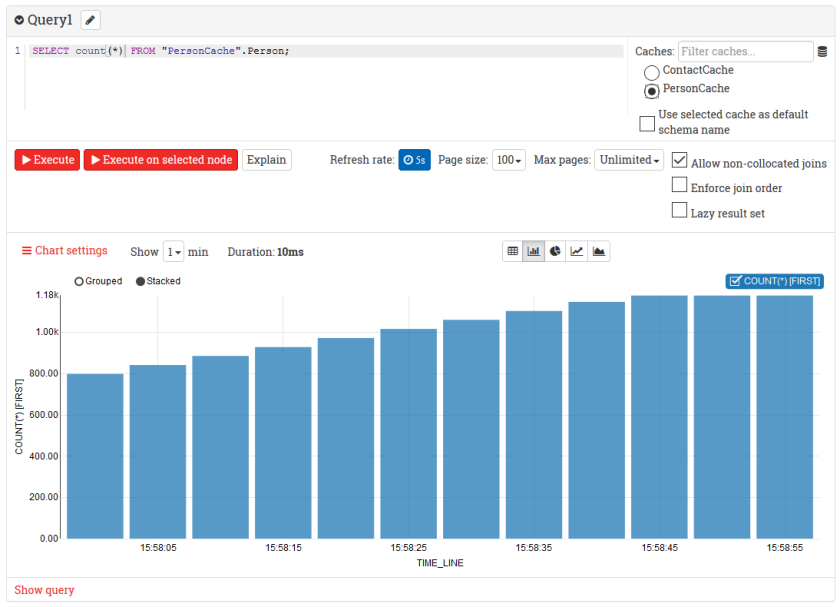
List selectContacts(String firstName, String lastName);

@Query("SELECT p.id, p.firstName, p.lastName, c.id, c.type, c.location FROM Person p JOIN \"ContactCache\".Contact c ON p.id=c.personId WHERE p.firstName=? and p.lastName=?")

List<List> selectContacts2(String firstName, String lastName);

}

We are nearing the end. Now, let's run our JUnit test TestCluster in order to generate some test data and put it into the clustered cache. You can monitor the size of a cache using the web console. All you have to do is to run the SELECT COUNT(\*) query and set graph mode as the default mode for displaying results. The chart visible below illustrates the number of entities stored inside Ignite's cluster at 5s intervals.



[Hortonworks Sandbox](https://dzone.com/go?i=285441&u=https%3A%2F%2Fhortonworks.com%2Fproducts%2Fsandbox%2F%3Futm_campaign%3Ddzonepre%2Fpostroll%26utm_medium%3Ddisplay%26apos%3B%26utm_source%3Ddzone%26utm_id%3D2216633) for HDP and HDF is your chance to get started on learning, developing, testing and trying out new features. Each [download](https://dzone.com/go?i=285441&u=https%3A%2F%2Fhortonworks.com%2Fproducts%2Fsandbox%2F%3Futm_campaign%3Ddzonepre%2Fpostroll%26utm_medium%3Ddisplay%26apos%3B%26utm_source%3Ddzone%26utm_id%3D2216633) comes preconfigured with interactive tutorials, sample data and developments from the Apache community.

**Like This Article? Read More From DZone**

[Apache Ignite With Spring Data](https://dzone.com/articles/apache-ignite-with-spring-data?fromrel=true)

[Getting Started With Apache Ignite (Part 7)](https://dzone.com/articles/getting-started-with-apache-ignite-part-7?fromrel=true)

[Optimizing Spark Job Performance With Apache Ignite (Part 1)](https://dzone.com/articles/increasing-spark-job-performances-with-apache-igni?fromrel=true)

[](https://dzone.com/refcardz/apache-spark?fromrel=true)

[Free DZone Refcard](https://dzone.com/refcardz/apache-spark?fromrel=true)

[Apache Spark](https://dzone.com/refcardz/apache-spark?fromrel=true)

Topics:

java,spring boot,apache ignite,tutorial,big data

Like [(1)](https://dzone.com/articles/word-count-hello-word-program-in-mapreduce)

Comment (0)

Save

[Tweet](https://dzone.com/articles/word-count-hello-word-program-in-mapreduce)

1,047 Views

Published at DZone with permission of Piotr Mińkowski , DZone MVB. [See the original article here.](https://piotrminkowski.wordpress.com/2018/04/04/apache-ignite-cluster-together-with-spring-boot/)

Opinions expressed by DZone contributors are their own.