

## SMOKE TEST

---

JupyterHub with Sparkmagic 2.1

Date Prepared: Oct 2019

Document Information

Project Name	<b>EPIC Accelerator Deployment &amp; Integration Services</b>		
Project Owner		Document Version No	0.1
Quality Review Method			
Prepared By	Priyanka	Preparation Date	Oct 2019
Reviewed By		Review Date	

**Table of Contents**

1   CONFIGURE AMBARI SPARK2 & LIVY SERVER.....4

2   TESTING SPARKMAGIC .....10

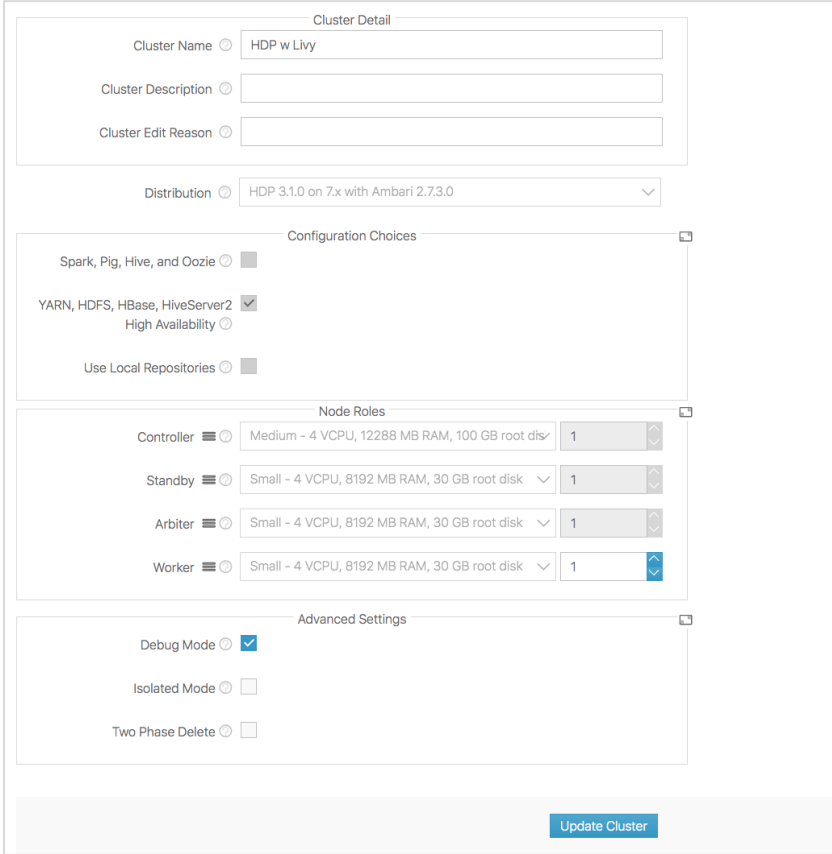
**Table of Tables**

**NO TABLE OF FIGURES ENTRIES FOUND.**

## 1 CONFIGURE AMBARI SPARK2 & LIVY SERVER

In this section, we will configure Ambari Spark2 and Livy Server.

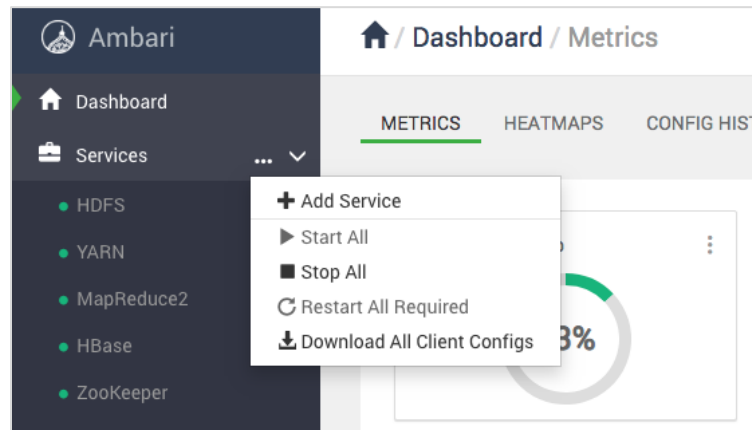
### 1. Create a Cluster (HDP 3.1.0)



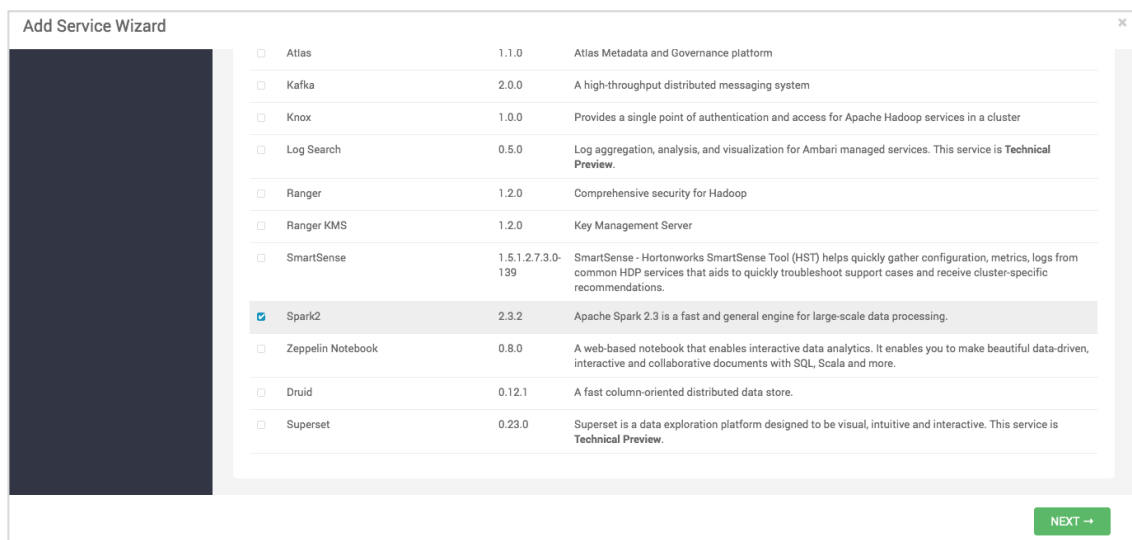
The image shows the Ambari Cluster Creation form. It is divided into several sections: Cluster Detail, Configuration Choices, Node Roles, and Advanced Settings. The Cluster Name is 'HDP w Livy'. The Distribution is 'HDP 3.1.0 on 7.x with Ambari 2.7.3.0'. Under Configuration Choices, 'YARN, HDFS, HBase, HiveServer2' is checked. Under Node Roles, there are four roles: Controller, Standby, Arbiter, and Worker, each with a dropdown menu for configuration and a numeric input for the number of nodes. Under Advanced Settings, 'Debug Mode' is checked. An 'Update Cluster' button is at the bottom right.

Role	Configuration	Count
Controller	Medium - 4 VCPU, 12288 MB RAM, 100 GB root disk	1
Standby	Small - 4 VCPU, 8192 MB RAM, 30 GB root disk	1
Arbiter	Small - 4 VCPU, 8192 MB RAM, 30 GB root disk	1
Worker	Small - 4 VCPU, 8192 MB RAM, 30 GB root disk	1

2. Once cluster is in Ready state, click on **Ambari Server** from the controller role
3. It will navigate you to the Ambari Login page, login using:
  - a. Username: admin
  - b. Password: admin
4. From the left-hand panel, click on **Services** and click on **Add Service**



5. From the Add Service Wizard, search for Spark2 and select it, then click on **NEXT**



6. Hive needed and Tez needed prompt will pop-up, click on **Ok** for both

7. Click on **NEXT** and accept all default in Assign Masters page. If Spark2 History Server is not adding on controller, manually map it to controller node.

# Smoke Test Document JupyterHub with Sparkmagic 2.1



Add Service Wizard

Choose Services

Assign Masters

Assign master components to hosts you want to run them on.

NameNode: bluedata-2.bdlocal (94.2 GB, 24 cores)

NameNode: bluedata-3.bdlocal (94.2 GB, 24 cores)

ResourceManager: bluedata-2.bdlocal (94.2 GB, 24 cores)

ResourceManager: bluedata-3.bdlocal (94.2 GB, 24 cores)

Timeline Service V1.5: bluedata-2.bdlocal (94.2 GB, 24 cores)

Timeline Service V2.0 Reader: bluedata-2.bdlocal (94.2 GB, 24 cores)

History Server: bluedata-2.bdlocal (94.2 GB, 24 cores)

Hive Metastore: bluedata-3.bdlocal (94.2 GB, 24 cores)

bluedata-2.bdlocal (94.2 GB, 24 cores)

NameNode ResourceManager

Timeline Service V1.5 Timeline Service V2.0 Reader

History Server HBase Master ZooKeeper Server

Spark2 History Server

bluedata-3.bdlocal (94.2 GB, 24 cores)

NameNode ResourceManager Hive Metastore

HiveServer2 HBase Master ZooKeeper Server

bluedata-4.bdlocal (94.2 GB, 24 cores)

ZooKeeper Server

1 hosts not running master services

8. Check on “**Livy for Spark2 Server & Spark2 Thrift Server**” on Controller host and click on **NEXT**

Add Service Wizard

Choose Services

Assign Masters

Assign Slaves and Clients

Assign slave and client components to hosts you want to run them on.  
Hosts that are assigned master components are shown with \*.  
“Client” will install Tez Client, Hive Client and Spark2 Client.

Host	none	all	none	all	none	all	none	all	none	all	none	all	none	
bluedata-2.bdlocal *	NFSGateway	<input type="checkbox"/>	NodeManager	<input type="checkbox"/>	RegionServer	<input type="checkbox"/>	Phoenix Query Server	<input type="checkbox"/>	Livy for Spark2 Server	<input checked="" type="checkbox"/>	Spark2 Thrift Server	<input checked="" type="checkbox"/>	Client	<input checked="" type="checkbox"/>
bluedata-3.bdlocal *	NFSGateway	<input checked="" type="checkbox"/>	NodeManager	<input checked="" type="checkbox"/>	RegionServer	<input checked="" type="checkbox"/>	Phoenix Query Server	<input type="checkbox"/>	Livy for Spark2 Server	<input type="checkbox"/>	Spark2 Thrift Server	<input type="checkbox"/>	Client	<input checked="" type="checkbox"/>
bluedata-4.bdlocal *	NFSGateway	<input checked="" type="checkbox"/>	NodeManager	<input checked="" type="checkbox"/>	RegionServer	<input checked="" type="checkbox"/>	Phoenix Query Server	<input type="checkbox"/>	Livy for Spark2 Server	<input type="checkbox"/>	Spark2 Thrift Server	<input type="checkbox"/>	Client	<input checked="" type="checkbox"/>
bluedata-5.bdlocal	NFSGateway	<input checked="" type="checkbox"/>	NodeManager	<input checked="" type="checkbox"/>	RegionServer	<input checked="" type="checkbox"/>	Phoenix Query Server	<input type="checkbox"/>	Livy for Spark2 Server	<input type="checkbox"/>	Spark2 Thrift Server	<input type="checkbox"/>	Client	<input checked="" type="checkbox"/>

Items per page: 25 1 - 4 of 4

← BACK

NEXT →

9. A error will occur on Hive, click on it

## Customize Services

We have come up with recommended configurations for the services you selected. Customize them as you see fit.

HDFS YARN MAPREDUCE2 TEZ HIVE 1 HBASE ZOOKEEPER SPARK2 MISC

There are 29 configuration changes in 4 services [Show Details](#)

10. You will get a page like below. Click on **DATABASE** tab and provide any password for **Database Password**

HDFS YARN MAPREDUCE2 TEZ HIVE HBASE ZOOKEEPER SPARK2 MISC

There are 29 configuration changes in 4 services [Show Details](#)

Group: Default (4) [Filter...](#)

SETTINGS DATABASE ADVANCED

### Database Configurations

Hive Database  
New MySQL

Hive Database Type  
mysql

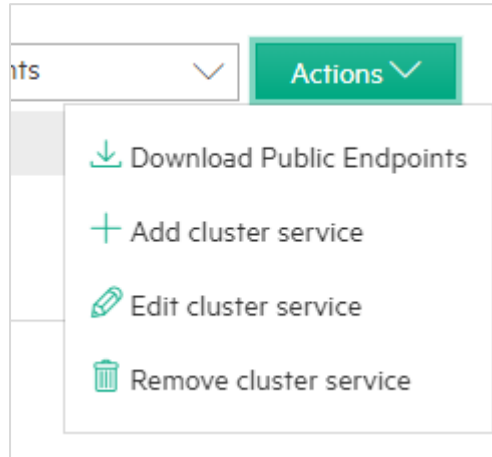
JDBC Driver Class  
com.mysql.jdbc.Driver

Database Name  
hive

Database Password  
\*\*\*\*\*

To use MySQL with Hive, you must download the <https://dev.mysql.com/downloads/connector/j/> from MySQL. Once downloaded to the Ambari Server host, run:  
`ambari-server setup --jdbc-db=mysql --jdbc-driver=/path/to/mysql/com.mysql.jdbc.Driver`

11. Ignore the Configuration warning. Click on **Proceed Anyway** and then click on **Deploy**
12. Once all components are installed, click on **Next > Complete > Restart > all Required services**
13. In order to get Livy server on EPIC cluster as Service, click on **Actions**
14. A drop-down menu will appear, click on **Add cluster service**



15. Provide the following details and click on **Submit**

Add Cluster Service

Node(s) ⓘ  
bluedata-2.bdlocal (controller) ▼

Service ID ⓘ  
Livy

Exported Service ⓘ

Service Name ⓘ  
Livy

Service Description ⓘ

Port ⓘ  
8999

Dashboard ⓘ  
☒

Path ⓘ  
/

Protocol ⓘ  
http ▼

✕ Cancel

✓ Submit

16. Once submitted, you can see Livy service on Controller



HDP 3.1.0 on 7.x with Ambari 2.7.3.0	controller	172.18.0.7	<a href="#">Livy</a> , <a href="#">Ambari Server</a> , <a href="#">HBASE Master</a> , <a href="#">HistoryServer</a> , <a href="#">NameNode</a> , <a href="#">ResourceManager</a> APP Timeline Server: mip-bd-vm67.mip.storage.hpccorp.net:10002 Zookeeper Server: mip-bd-vm67.mip.storage.hpccorp.net:10007 SSH: mip-bd-vm67.mip.storage.hpccorp.net -p 10006
--------------------------------------	------------	------------	--

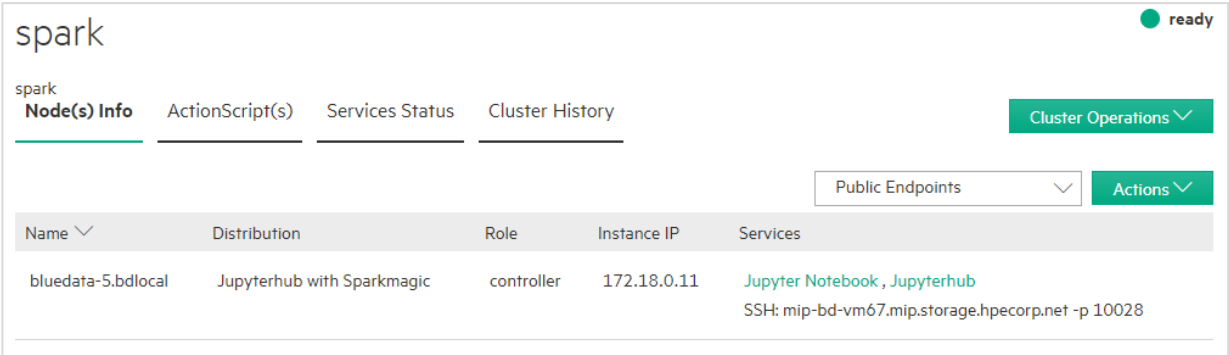
17. Click on Livy, it will navigate you to a new window



## 2 TESTING SPARKMAGIC

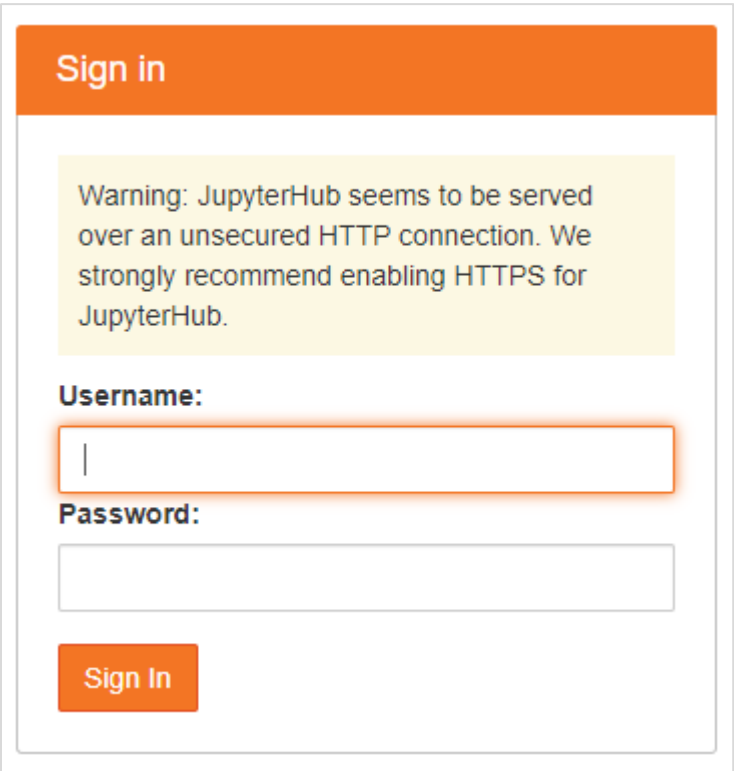
In this section, we will test Sparkmagic.

1. From JupyterHub with Sparkmagic cluster, click on **JupyterHub** service



Name	Distribution	Role	Instance IP	Services
bluedata-5.bdlocal	Jupyterhub with Sparkmagic	controller	172.18.0.11	Jupyter Notebook , Jupyterhub SSH: mip-bd-vm67.mip.storage.hpccorp.net -p 10028

2. It will navigate you to JupyterHub login page, login using your credentials



**Sign in**

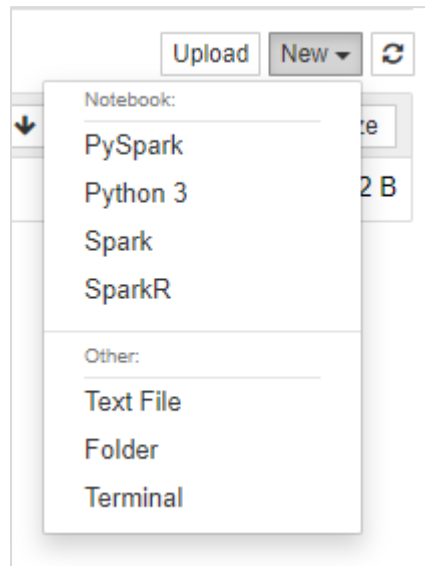
Warning: JupyterHub seems to be served over an unsecured HTTP connection. We strongly recommend enabling HTTPS for JupyterHub.

**Username:**

**Password:**

**Sign In**

3. Click on **New**, a drop-down menu will appear, click on **PySpark**. It will navigate you to a new Jupyter Notebook



4. Execute the below command:

```
%load_ext sparkmagic.magics
```

```
%manage_spark
```

In [1]: `%load_ext sparkmagic.magics`

In [2]: `%manage_spark`

Manage Sessions | **Create Session** | Add Endpoint | Manage Endpoints

Endpoint:  Name:  Language:

In [ ]:

In [1]: `%load_ext sparkmagic.magics`

In [2]: `%manage_spark`

int | Manage Endpoints

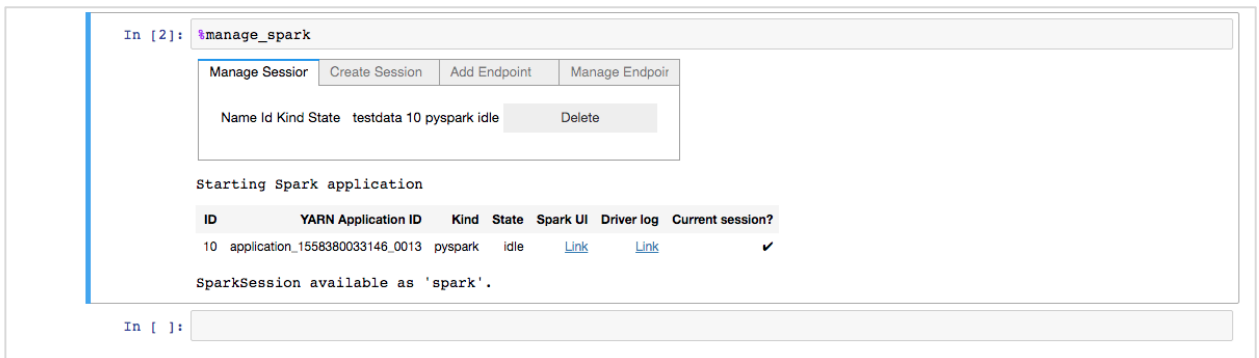
Language:

Starting Spark application

In [ ]:

**Note:** If you don't see Endpoint created automatically, you can add Endpoint manually or we can define in config.json (Here, Endpoint URL is: <Livy\_server\_URL> from the EPIC Cluster)

5. Click on **Create Session** (You may have to scroll right to see the option), in some time Spark session will be available



In [2]: %manage\_spark

Manage Session Create Session Add Endpoint Manage Endpoir

Name	Id	Kind	State	testdata	10	pyspark	idle	Delete
Starting Spark application								
ID	YARN Application ID	Kind	State	Spark UI	Driver log	Current session?		
10	application_1558380033146_0013	pyspark	idle	<a href="#">Link</a>	<a href="#">Link</a>	✓		

SparkSession available as 'spark'.

In [ ]:

6. Use sample PySpark code to load the data from HDFS

```
%%spark

df =
sqlContext.read.format('com.databricks.spark.csv').options(header='true', inferschema='true').load('/tmp/Iris.csv')
```

```
%%spark

df.registerTempTable("Iris")

df.show()
```

## Smoke Test Document JupyterHub with Sparkmagic 2.1



```
In [3]: %%spark
df = sqlContext.read.format('com.databricks.spark.csv').options(header='true', inferschema='true').load('/tmp/Iris.csv')
```

```
In [4]: %%spark
df.registerTempTable("Iris")
df.show()
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
1	1	5.1	3.5	1.4	0.2	Iris-setosa
2	2	4.9	3.0	1.4	0.2	Iris-setosa
3	3	4.7	3.2	1.3	0.2	Iris-setosa
4	4	4.6	3.1	1.5	0.2	Iris-setosa
5	5	5.0	3.6	1.4	0.2	Iris-setosa
6	6	5.4	3.9	1.7	0.4	Iris-setosa
7	7	4.6	3.4	1.4	0.3	Iris-setosa
8	8	5.0	3.4	1.5	0.2	Iris-setosa
9	9	4.4	2.9	1.4	0.2	Iris-setosa
10	10	4.9	3.1	1.5	0.1	Iris-setosa
11	11	5.4	3.7	1.5	0.2	Iris-setosa
12	12	4.8	3.4	1.6	0.2	Iris-setosa
13	13	4.8	3.0	1.4	0.1	Iris-setosa
14	14	4.3	3.0	1.1	0.1	Iris-setosa
15	15	5.8	4.0	1.2	0.2	Iris-setosa
16	16	5.7	4.4	1.5	0.4	Iris-setosa
17	17	5.4	3.9	1.3	0.4	Iris-setosa
18	18	5.1	3.5	1.4	0.3	Iris-setosa
19	19	5.7	3.8	1.7	0.3	Iris-setosa
20	20	5.1	3.8	1.5	0.3	Iris-setosa

only showing top 20 rows

```
In [ ]: |
```

**Note:** You should have **Iris.csv** in controller Node.

**Note:** In order to use the curl command to submit jobs in Notebook use the ! (Bang) in the beginning.