Hewlett Packard Enterprise

# SMOKE TEST

## JupyterHub with Sparkmagic 2.1

Date Prepared: Oct 2019

Hewlett Packard Enterprise

Document Information

| Project Name | **EPIC Accelerator Deployment & Integration Services** | | |
|---|---|---|---|
| Project Owner | | Document Version No | 0.1 |
| Quality Review Method | | | |
| Prepared By | Priyanka | Preparation Date | Oct 2019 |
| Reviewed By | | Review Date | |

## Table of Contents



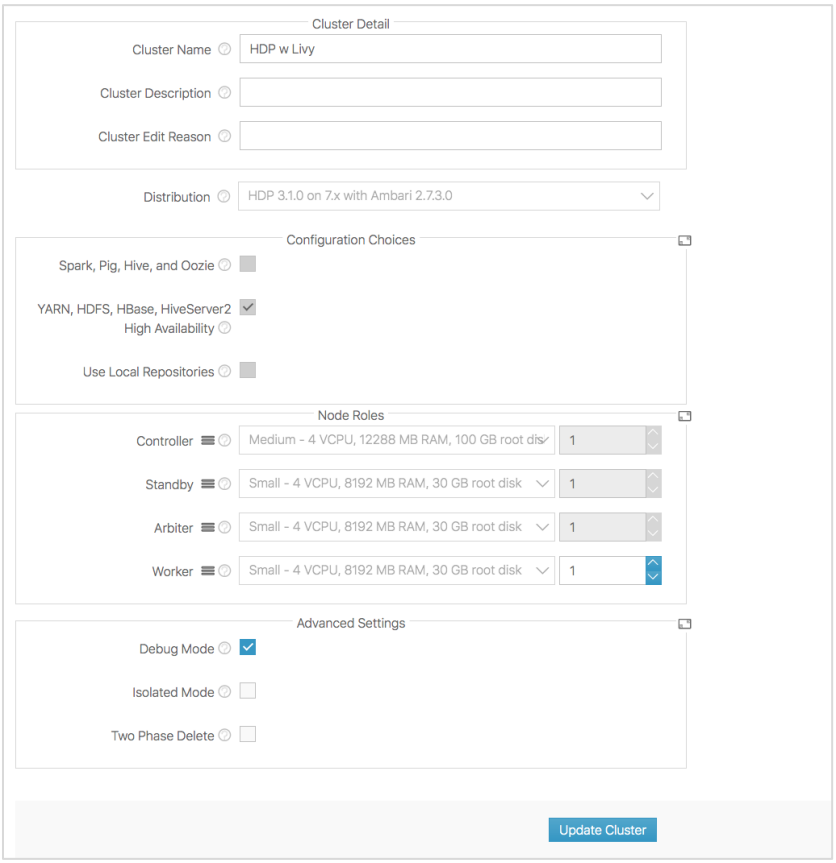
## Table of Tables

**NO TABLE OF FIGURES ENTRIES FOUND.**

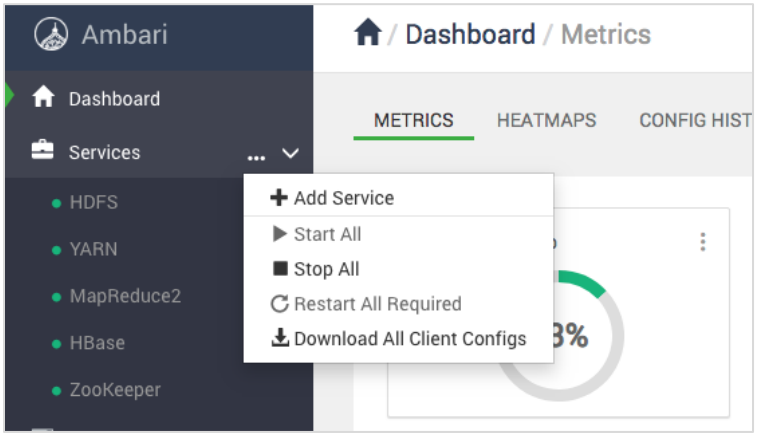
# 1 CONFIGURE AMBARI SPARK2 & LIVY SERVER

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

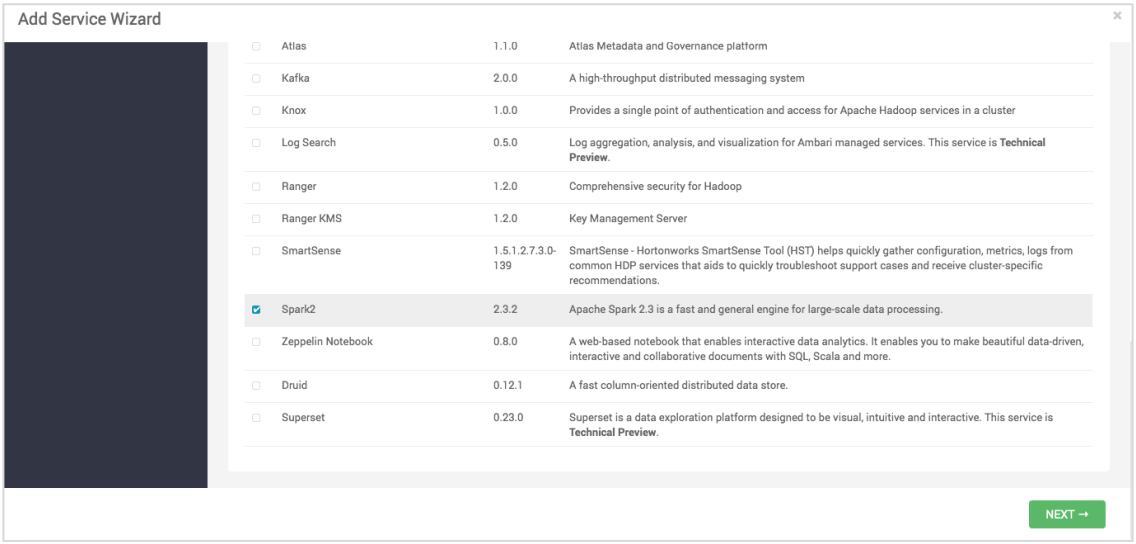
1. Create a Cluster (HDP 3.1.0)



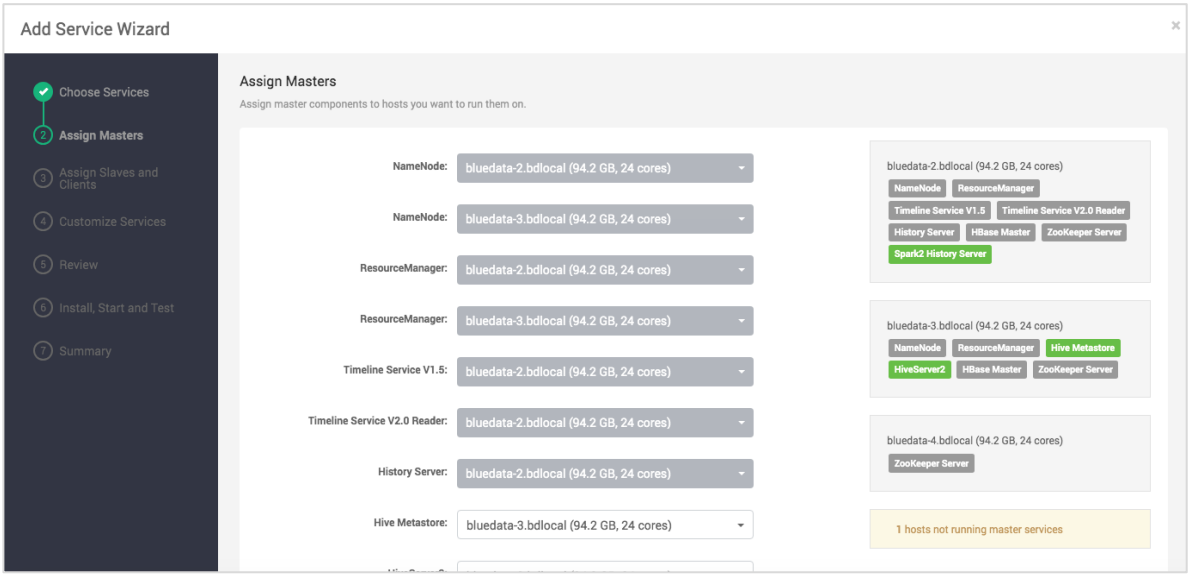

2. Once cluster is in Ready state, click on **Ambari Server** from the controller role
3. It will navigate you to the Ambari Login page, login using:
   a. Username: admin
   b. Password: admin
4. From the left-hand panel, click on **Services** and click on **Add Service**



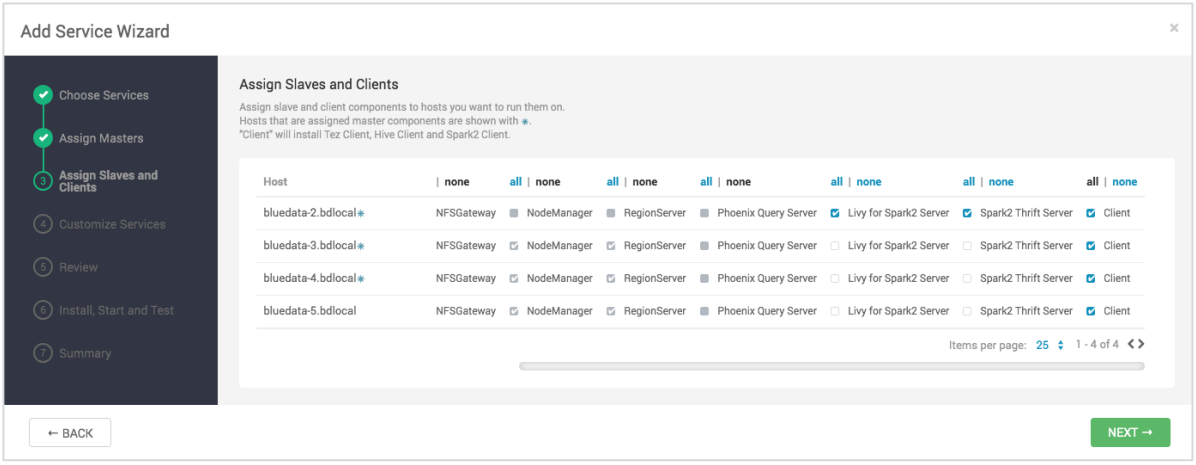

5. From the Add Service Wizard, search for Spark2 and select it, then click on **NEXT**



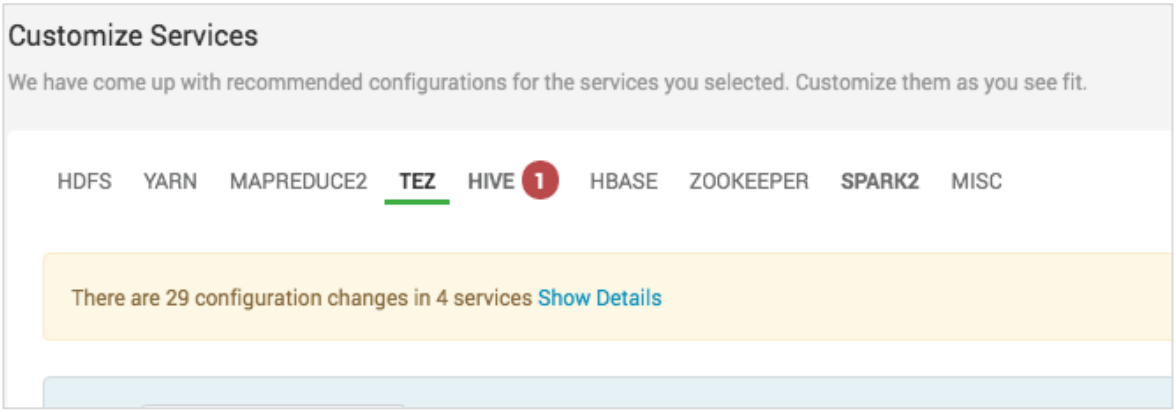

6. Hive needed and Tez needed prompt will pop-up, click on **Ok** for both
7. Click on **NEXT** and accept all default in Assign Masters page. If Spark2 History Server is not adding on controller, manually map it to controller node.



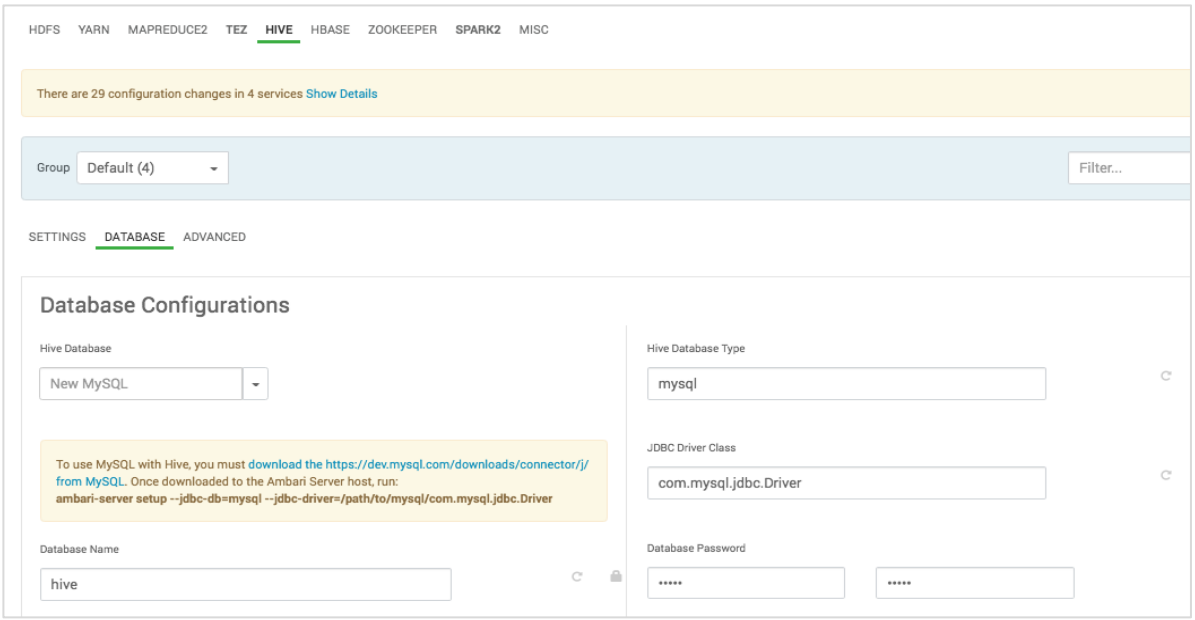

8. Check on “**Livy for Spark2 Server & Spark2 Thrift Server**” on Controller host and click on **NEXT**



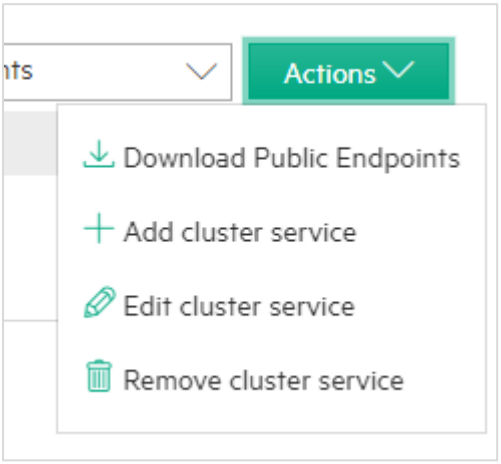

9. A error will occur on Hive, click on it



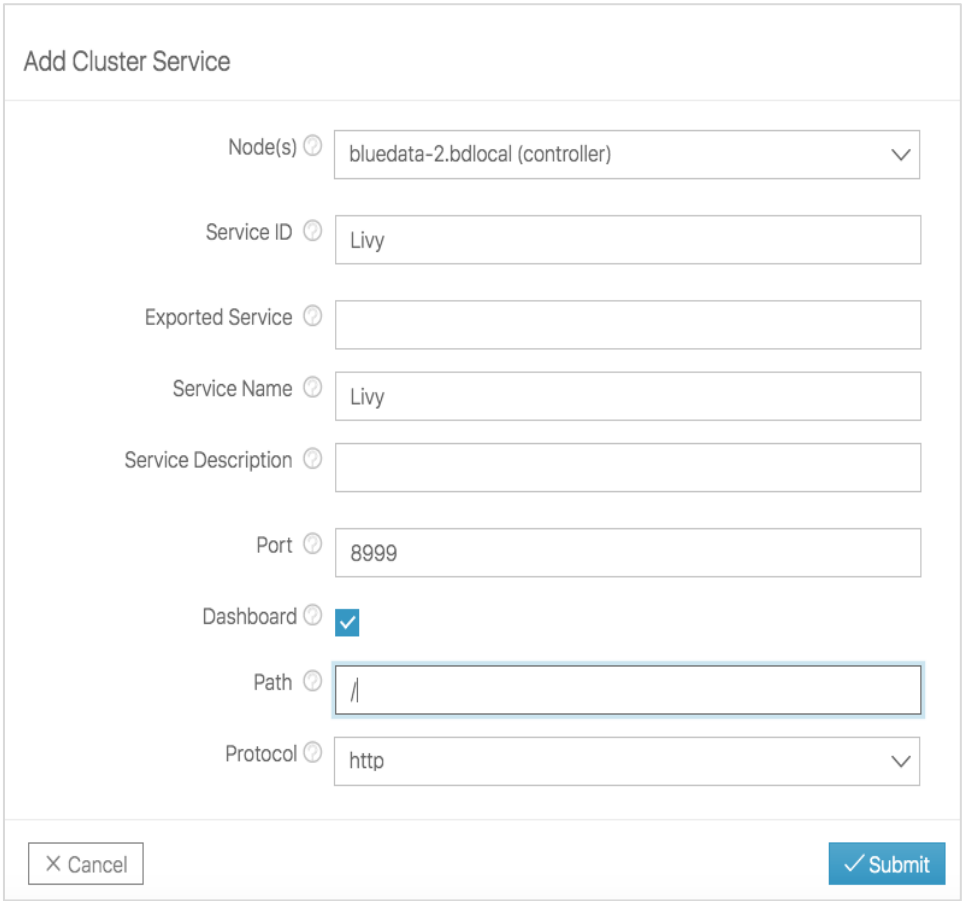

10. You will get a page like below. Click on **DATABASE** tab and provide any password for **Database Password**




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**




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

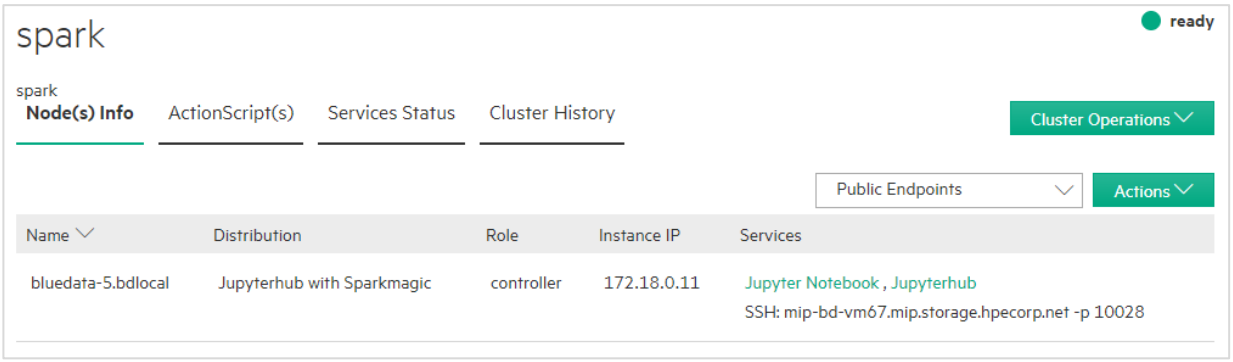
| | | | |
|---|---|---|---|
| HDP 3.1.0 on 7.x with Ambari 2.7.3.0 | controller | 172.18.0.7 | Livy , Ambari Server , HBASE Master , HistoryServer , NameNode , ResourceManager<br>APP Timeline Server: mip-bd-vm67.mip.storage.hpecorp.net:10002<br>Zookeeper Server: mip-bd-vm67.mip.storage.hpecorp.net:10007<br>SSH: mip-bd-vm67.mip.storage.hpecorp.net -p 10006 |

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

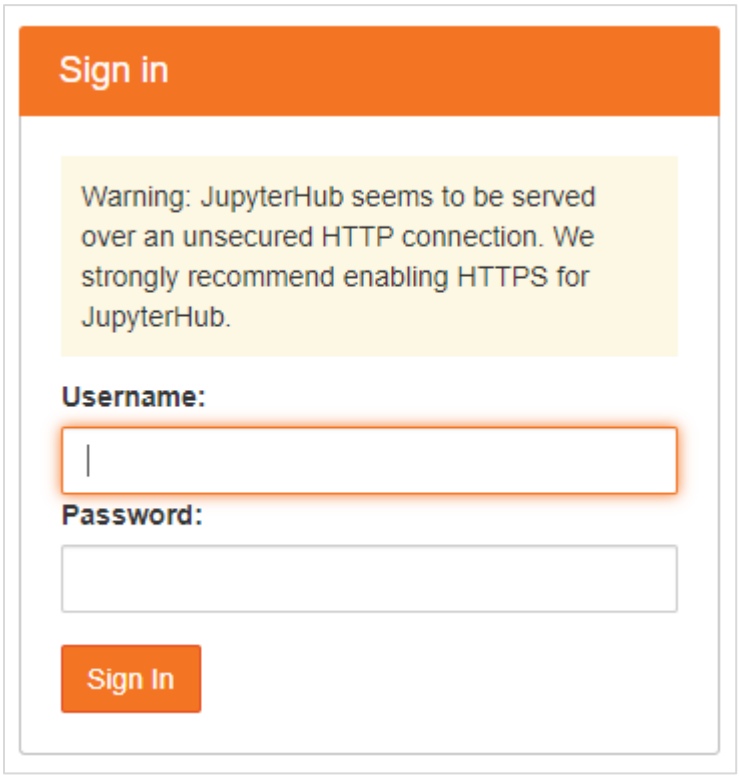
# 2 TESTING SPARKMAGIC

In this section, we will test Sparkmagic.

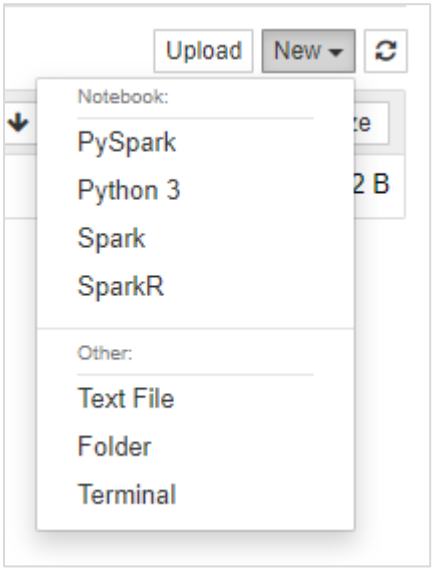
1. From JupyterHub with Sparkmagic cluster, click on **JupyterHub** service




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



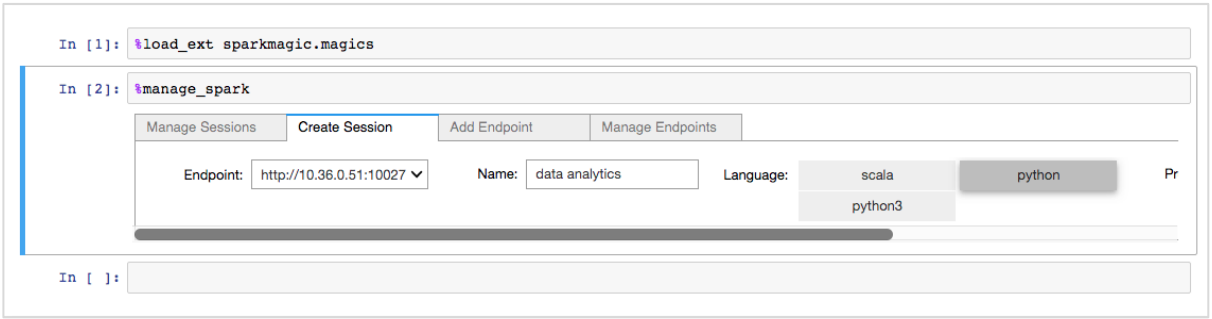

3. Click on **New**, a drop-down menu will appear, click on **PySpark**. It will navigate you to a new Jupyter Notebook

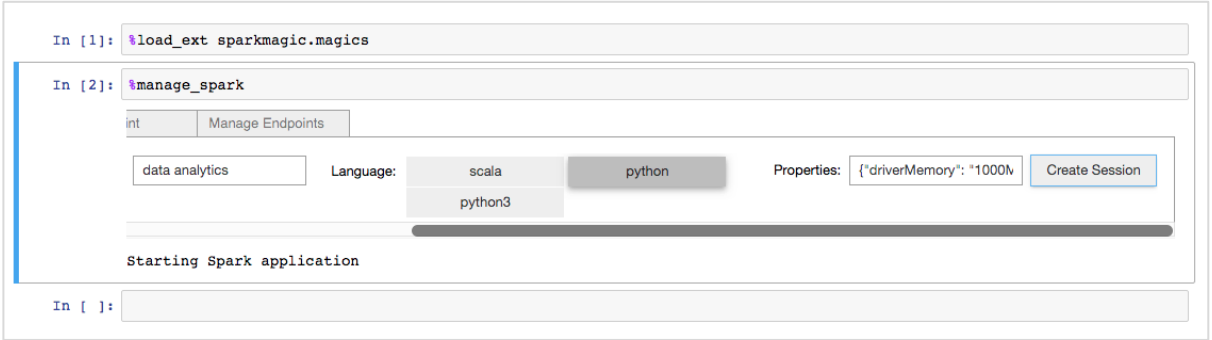



4. Execute the below command:

```
%load_ext sparkmagic.magics
```

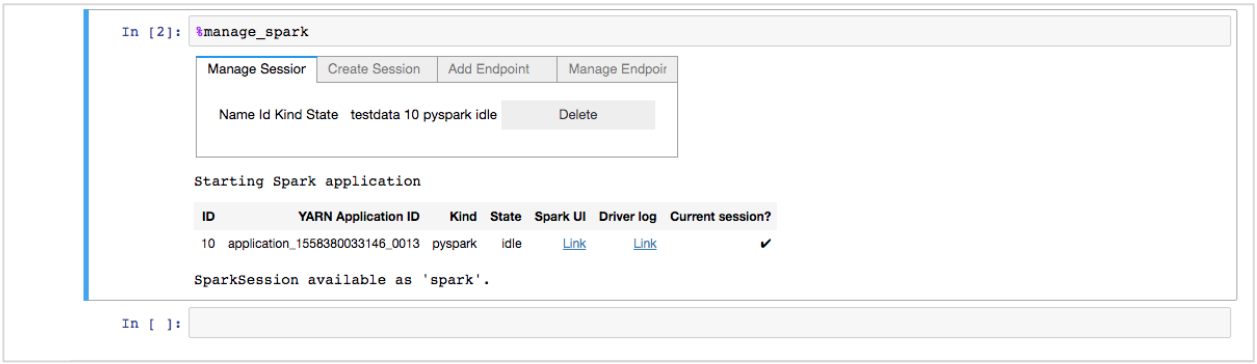
```
%manage_spark
```



**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




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

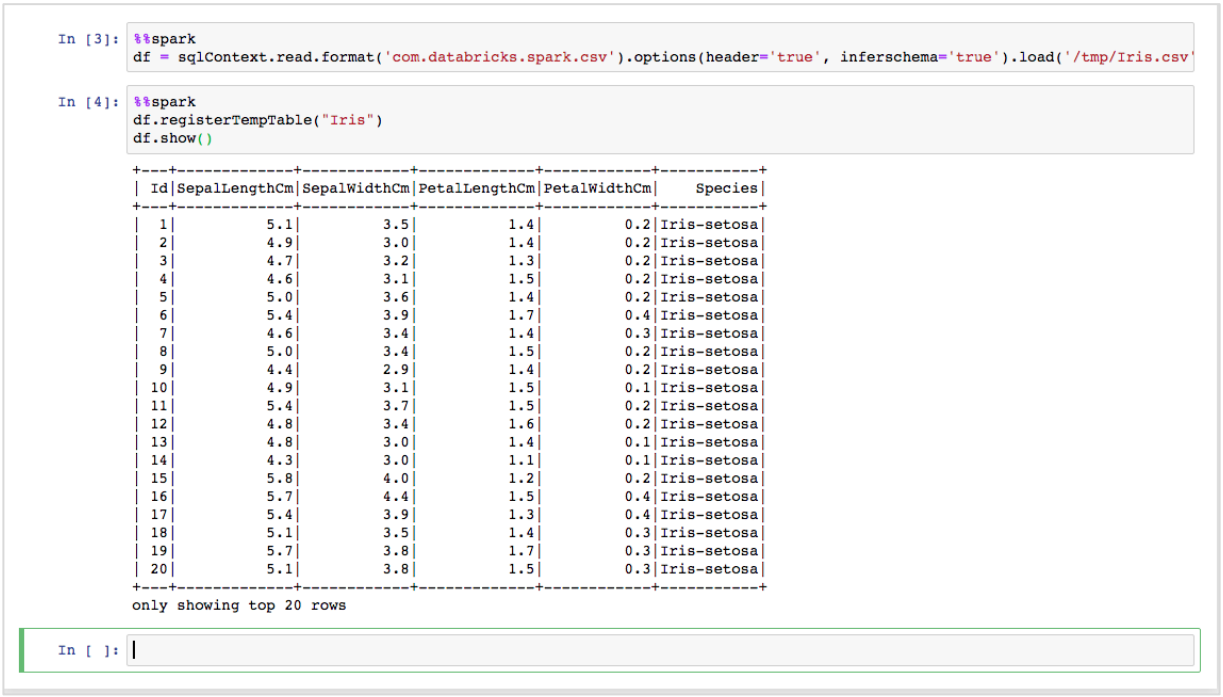
```
%%spark

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

```
%%spark

df.registerTempTable("Iris")

df.show()
```



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