logocover

**SMOKE TEST DOCUMENT**

HDF Smoke Test Cases

Date Prepared: July 2019

**Document Information**

|  |  |  |  |
| --- | --- | --- | --- |
| **Project Name** | **HDF Smoke Test Document** | | |
| **Project Owner** |  | **Document Version No** | 1.0 |
| **Quality Review Method** | By email/HP SharePoint |  |  |
| **Prepared By** |  | **Preparation Date** | July 2019 |
| **Reviewed By** | Refer to version history | **Review Date** |  |

**Table of Contents**

[1 Testing NIFI 5](#_Toc15658670)

[1.1 Select a GenerateFlowFile processor in NIFI UI 5](#_Toc15658671)

[1.2 Configure GenerateFlowFile processor to generate the data 5](#_Toc15658672)

[1.3 Select a PutFile processor in NIFI UI 6](#_Toc15658673)

[1.4 Configure PutFile processor to store generated data from GenerateFlowFile processor 6](#_Toc15658674)

[1.5 Connect and run the processors 7](#_Toc15658675)

[1.6 Verify the data 7](#_Toc15658676)

[2 testing NIFI registry 8](#_Toc15658677)

[2.1 Create HelloWorld bucket in NIFI Registry UI 8](#_Toc15658678)

[2.2 Create a process group 8](#_Toc15658679)

[2.3 Configure a process group 9](#_Toc15658680)

[2.4 Verify version control 9](#_Toc15658681)

[3 testing grafana 10](#_Toc15658682)

[3.1 Selection of component 10](#_Toc15658683)

[4 testing solr 11](#_Toc15658684)

[4.1 Go to the Solr Admin UI in HDF cluster 11](#_Toc15658685)

[4.2 Go to the Solr container and create a new collection 11](#_Toc15658686)

[4.3 Verify HelloWorld collection creation in Solr Admin UI 12](#_Toc15658687)

[4.4 Place some example JSON data in HelloWorld collection 12](#_Toc15658688)

[4.5 Fetch JSON data from HelloWorld collection 13](#_Toc15658689)

[5 testing Kafka 14](#_Toc15658690)

[5.1 Create a new Kafka topic 14](#_Toc15658691)

[5.2 Create a new topic 14](#_Toc15658692)

[6 testing log search 15](#_Toc15658693)

[6.1 Log search dashboard for service logs 15](#_Toc15658694)

[6.2 Log search dashboard for audit logs 15](#_Toc15658695)

[7 testing Streaming Analytics Manager 16](#_Toc15658696)

[7.1 Go to the Streamline UI 16](#_Toc15658697)

[7.2 Create a service pool 16](#_Toc15658698)

[7.3 Create a new environment for HDF cluster from service pool 17](#_Toc15658699)

[7.4 Create a new application for your environment 17](#_Toc15658700)

[8 testing ranger 19](#_Toc15658701)

[9 testing storm 20](#_Toc15658702)

[9.1 Download Storm WordCount example 20](#_Toc15658703)

[9.2 Change directory to the storm-example 20](#_Toc15658704)

[9.3 Building jar 20](#_Toc15658705)

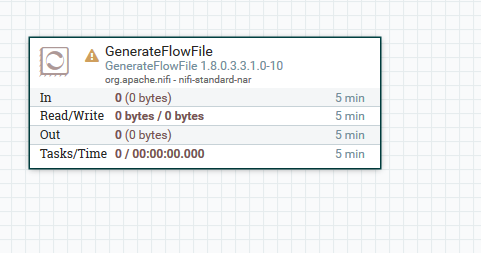
[9.4 Create topology for Storm WordCount example 21](#_Toc15658706)

[9.5 Verify WordCount topology from Storm UI 22](#_Toc15658707)

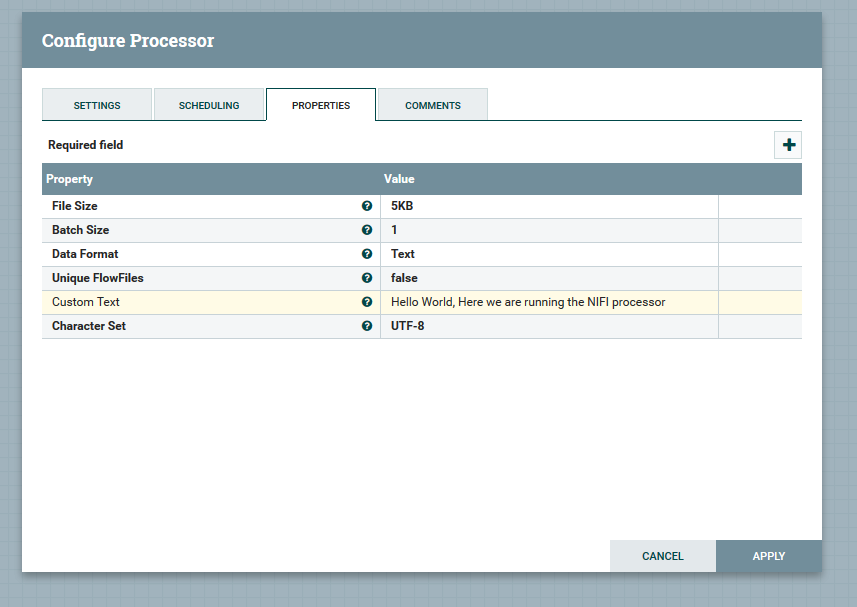
# Testing NIFI

We will use NIFI UI to create a HelloWorld dataflow

## Select a GenerateFlowFile processor in NIFI UI



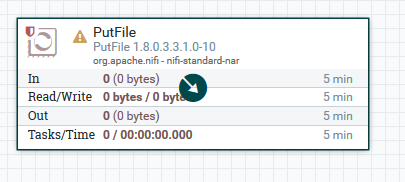
## Configure GenerateFlowFile processor to generate the data



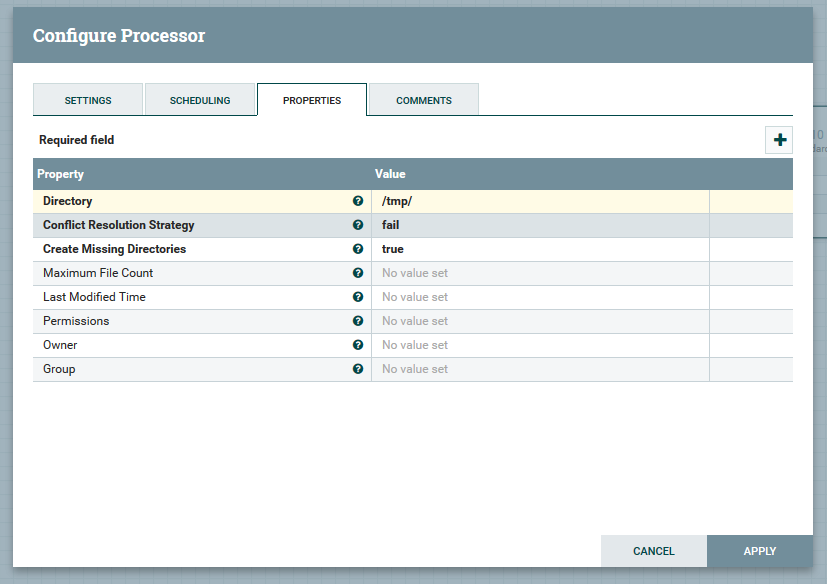
Following are the configuration you need to perform for GenerateFlowFile processor:

* Set the file size
* Enter custom text in Custom Text section

## Select a PutFile processor in NIFI UI



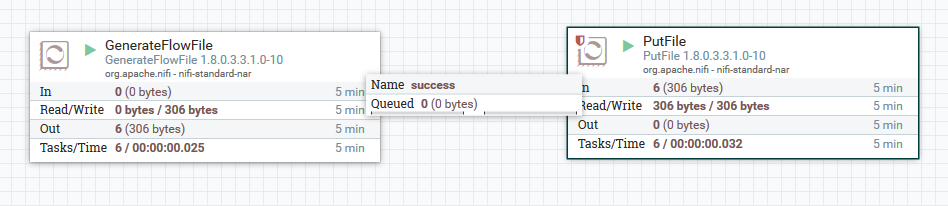
## Configure PutFile processor to store generated data from GenerateFlowFile processor



You need to perform following configuration for PutFile processor:

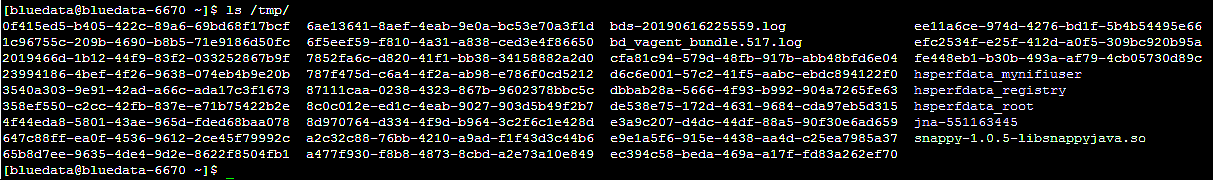
* Enter a location where you want to store generated data from GenerateFlowFile processor. (Ex. /tmp/)

## Connect and run the processors



## Verify the data

Go to NIFI container and check into /tmp directory if the data is stored or not



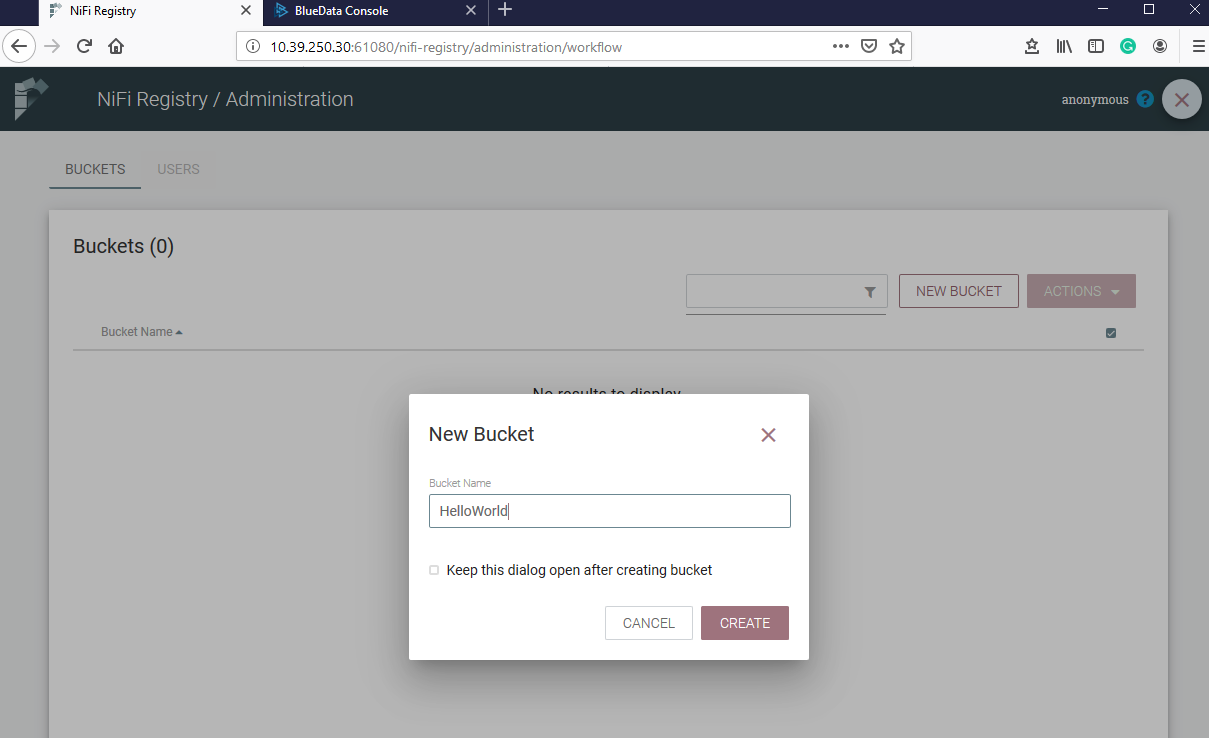
Use cat command to check the stored data

C:\Users\Vsamant\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\DF8B2E71.tmp

# testing NIFI registry

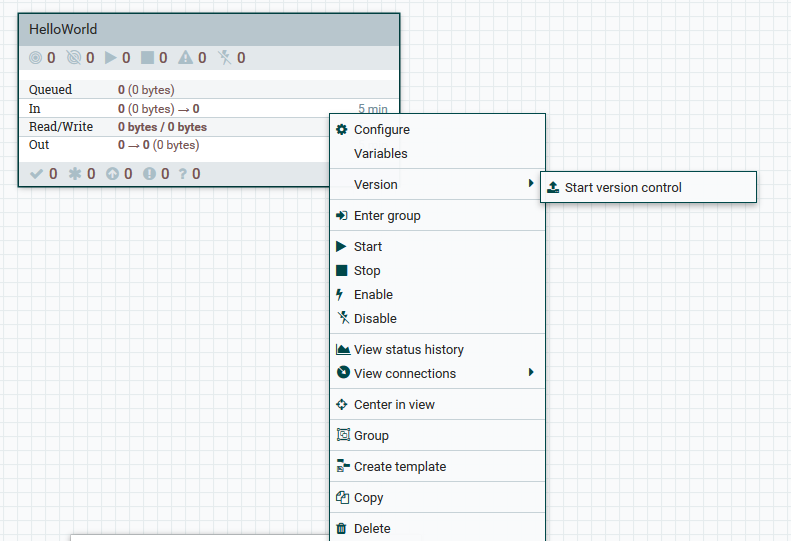
Here we will test the NIFI Registry service. Go to the NIFI Registry UI and create a bucket called HelloWorld.

## Create HelloWorld bucket in NIFI Registry UI



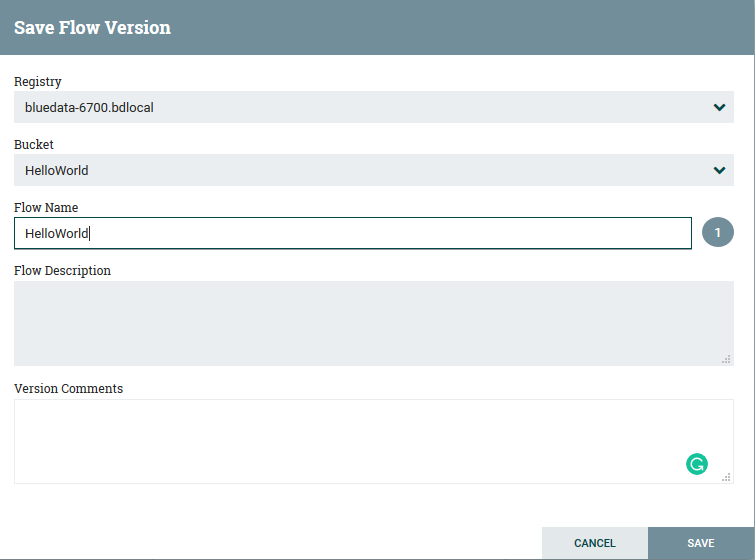
## Create a process group

Create a HelloWorld process group in NIFI UI and start version control



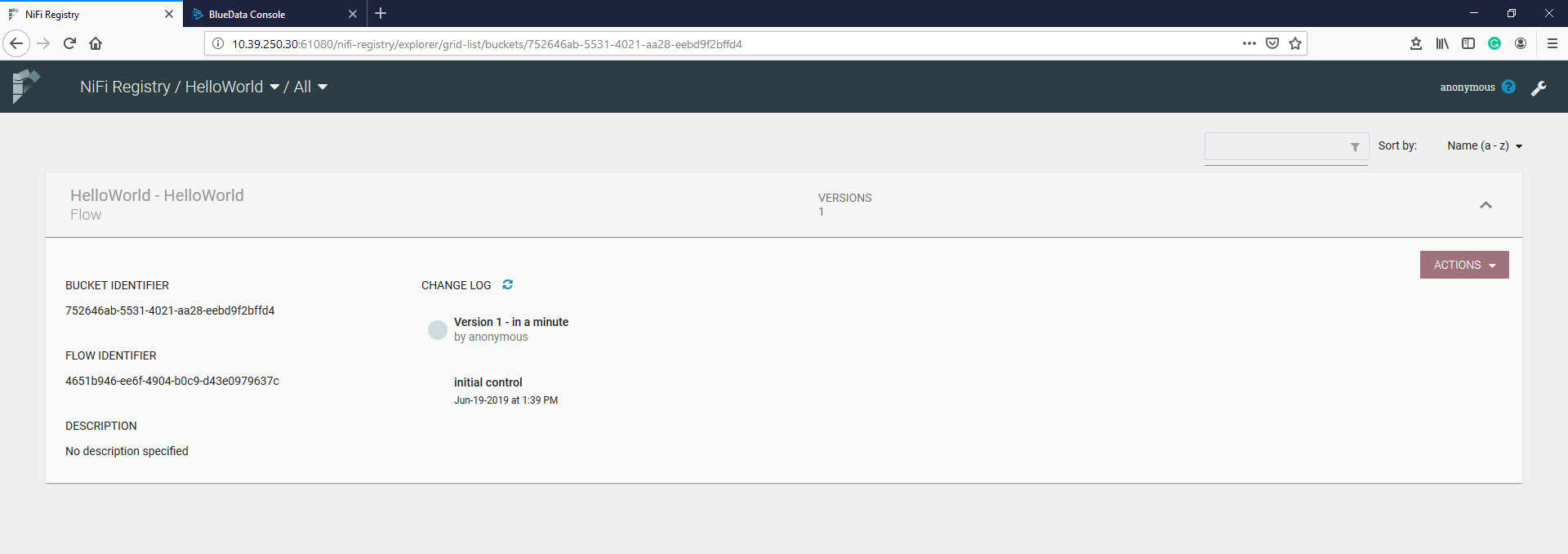
## Configure a process group

Configure HelloWorld process group for storing it into bucket HelloWorld in NIFI Registry



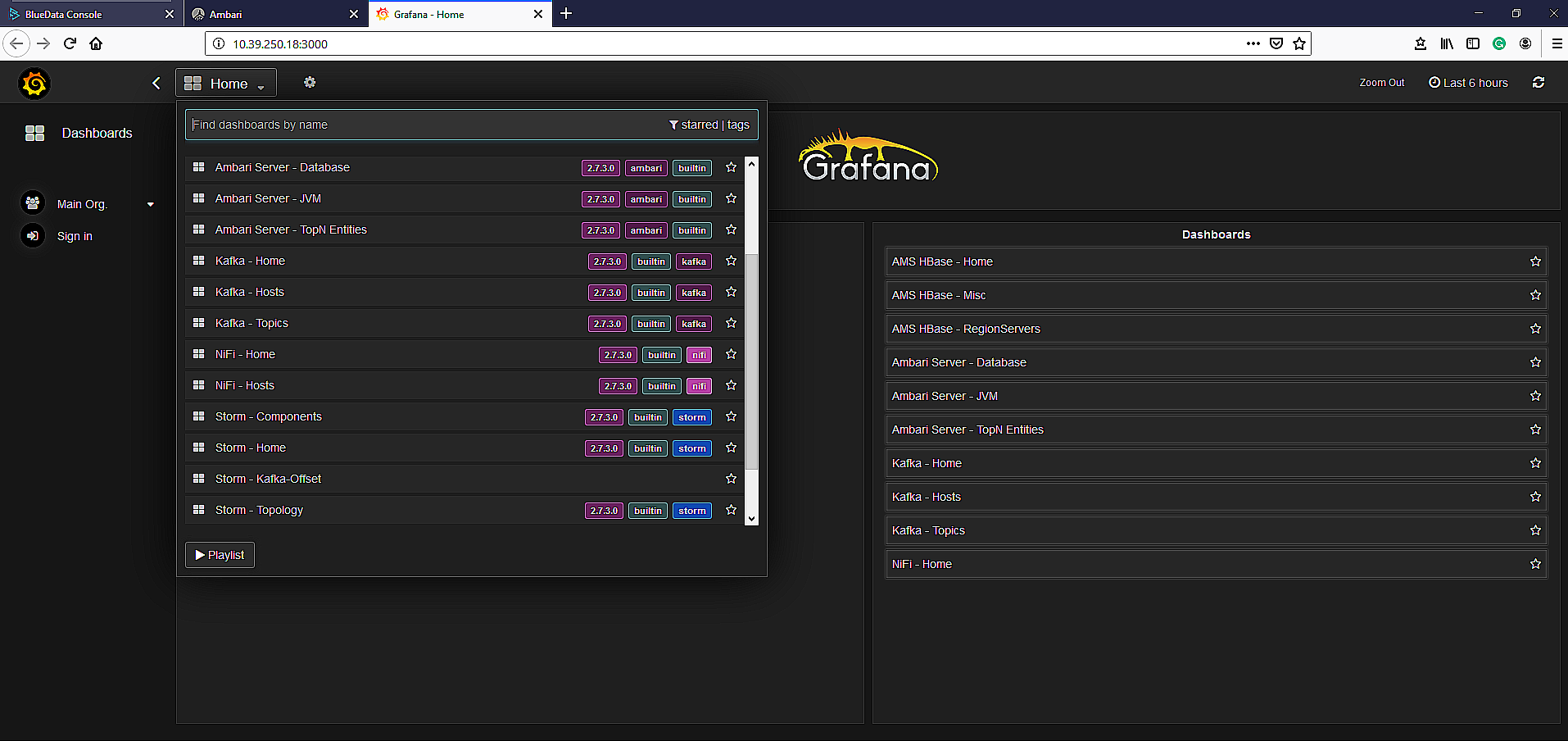
## Verify version control

Go to the NIFI Registry UI. Select HelloWorld bucket and check if version control started for process group HelloWorld



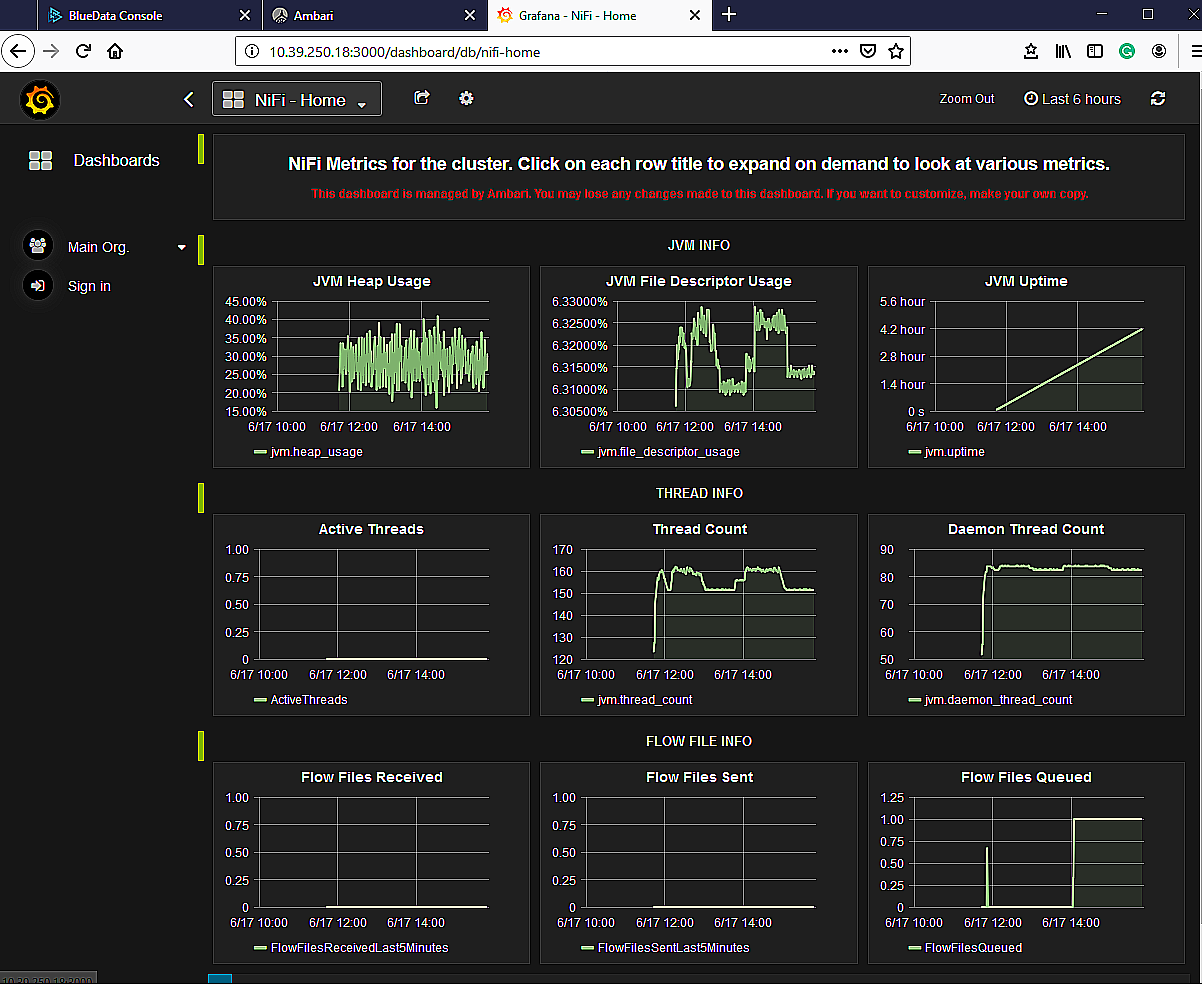
# testing grafana

Go to the Grafana UI and click on the search icon. Here you can see the components which is deployed into the HDF cluster



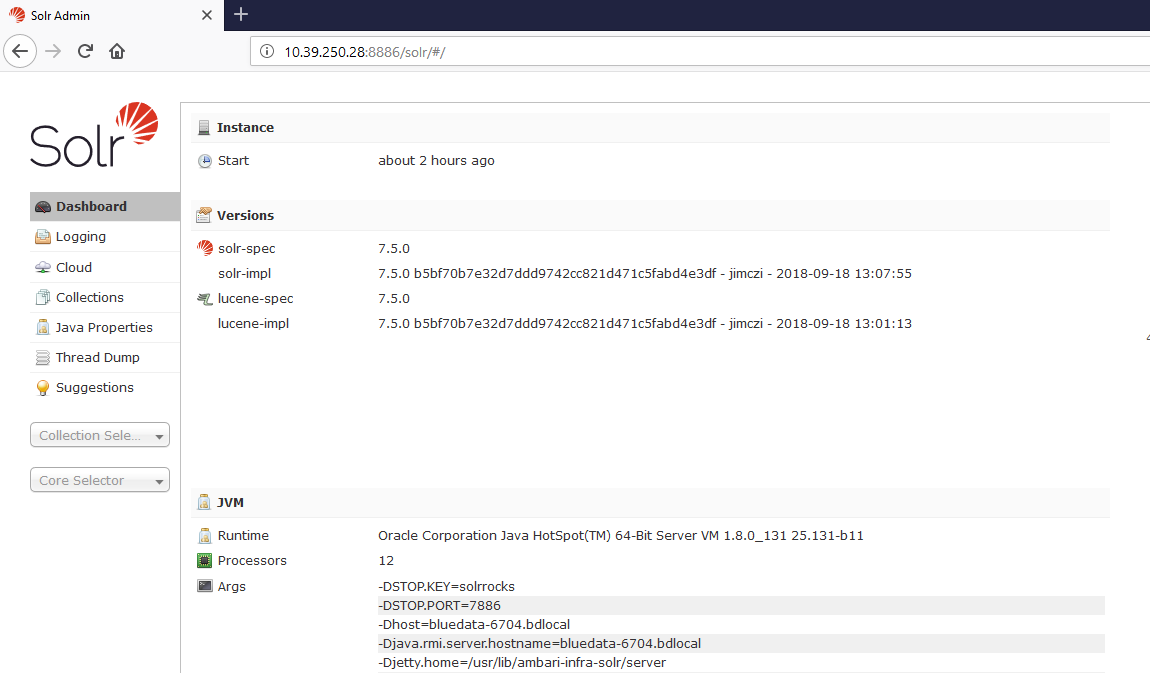
## Selection of component

NIFI – Home component is selected in Grafana UI, and you can see the resources usages by the NIFI service

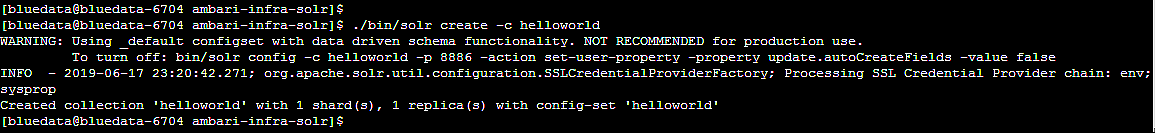


# testing solr

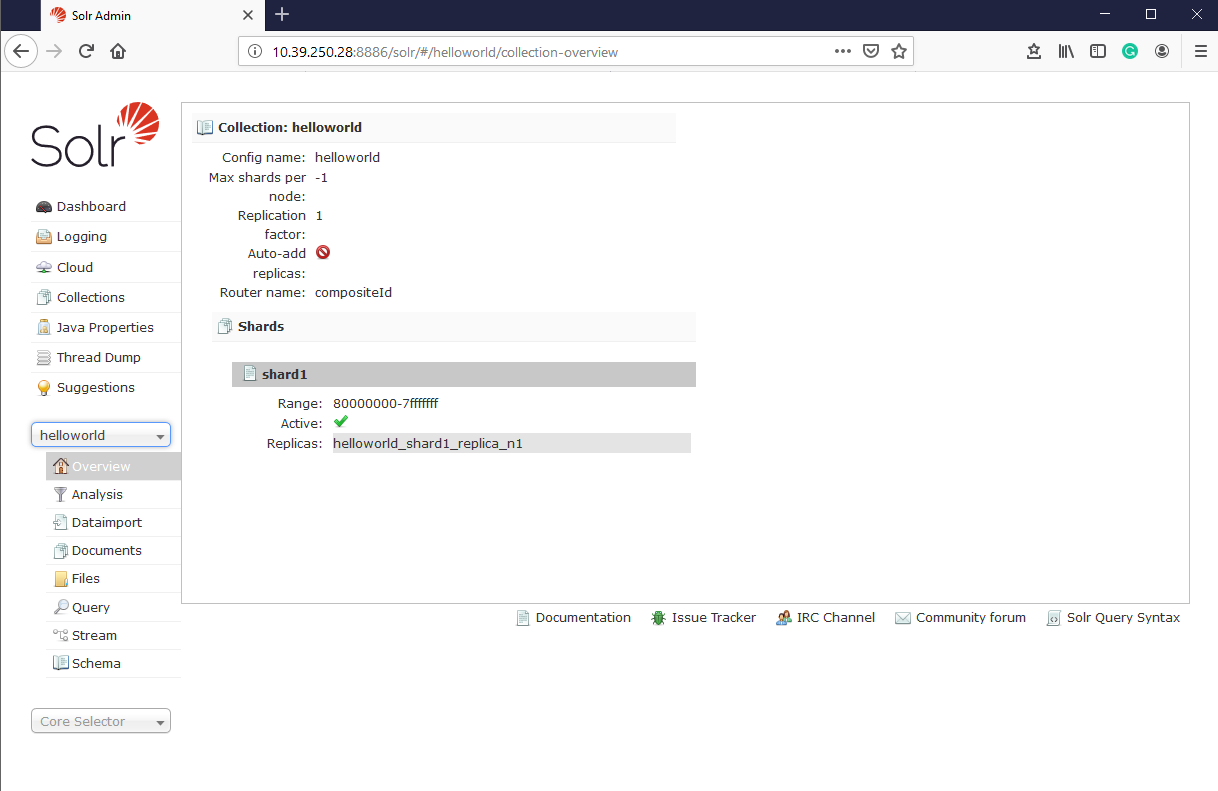
## Go to the Solr Admin UI in HDF cluster



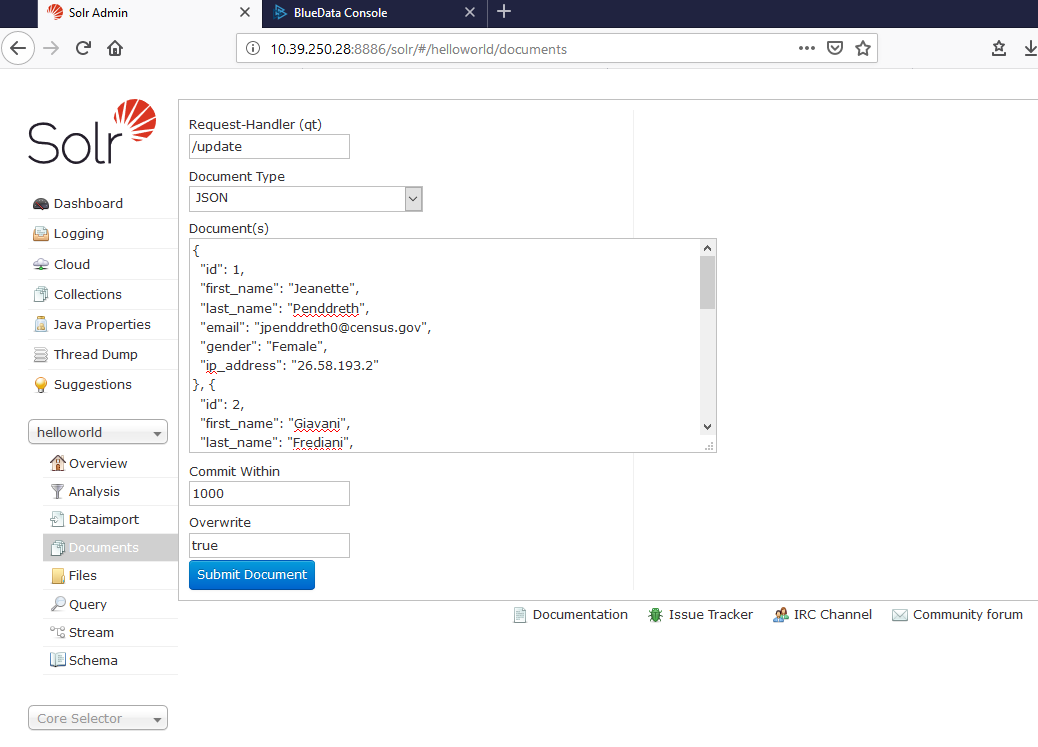
## Go to the Solr container and create a new collection



## Verify HelloWorld collection creation in Solr Admin UI



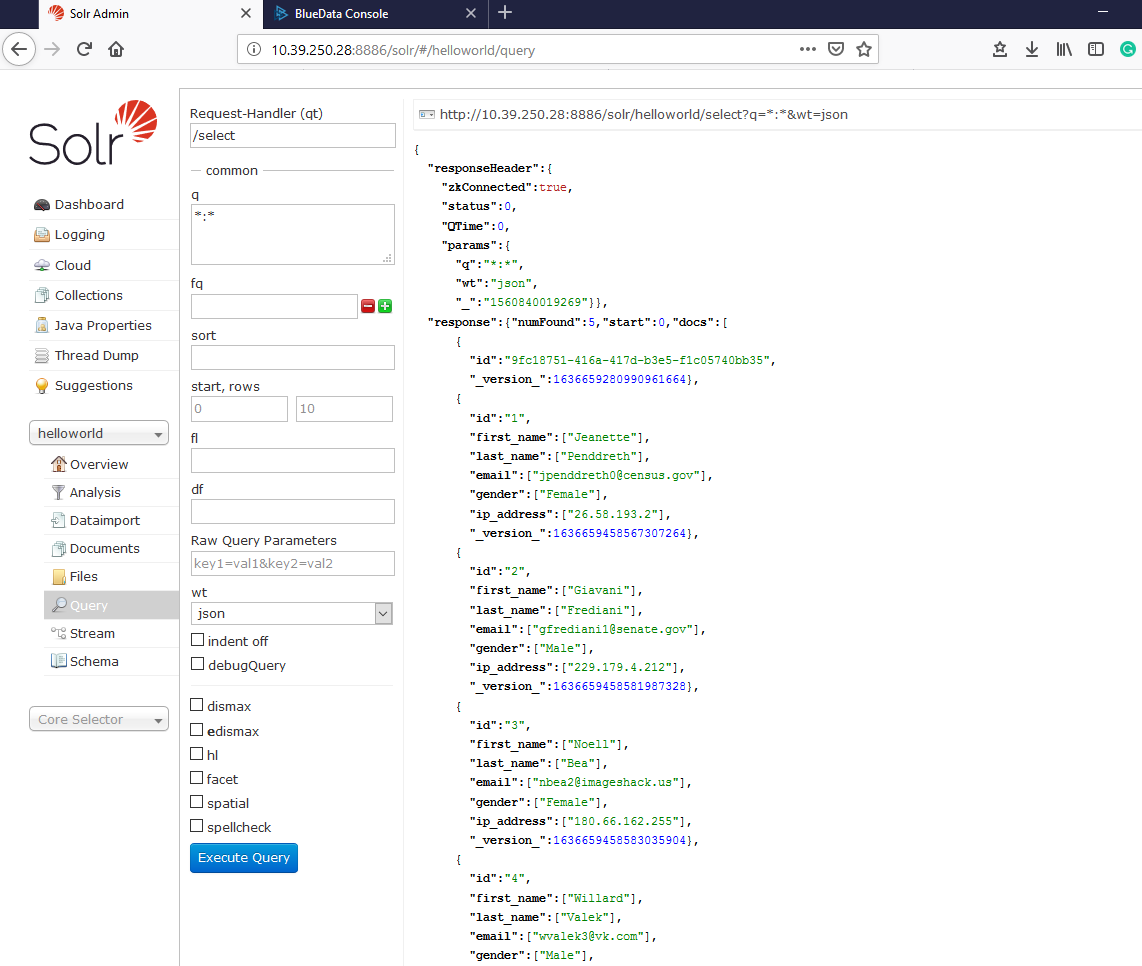
## Place some example JSON data in HelloWorld collection



Perform below steps to store JSON data into HelloWorld collection:

* Select document type as JSON
* Place some example JSON data in document(s) section in HelloWorld collection and click on Submit document

## Fetch JSON data from HelloWorld collection



Perform below step to fetch JSON data from HelloWorld collection:

* Select wt as JSON and click on execute query

# testing Kafka

Go inside the Kafka container and follow the below steps

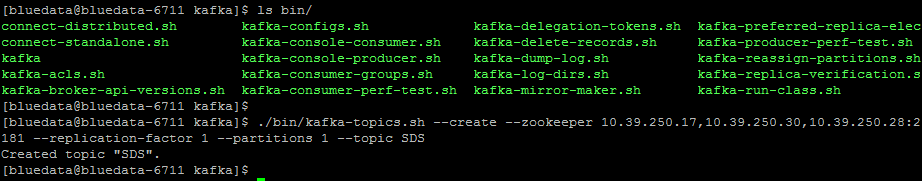
## Create a new Kafka topic

Create new Kafka topic inside the Kafka container

## Create a new topic

Execute the below command to create a new topic “SDS”

bin/Kafka-topics --create --zookeeper <ip address of zookeeper>:2181 --replication-factor 1 --partitions 1 --topic SDS

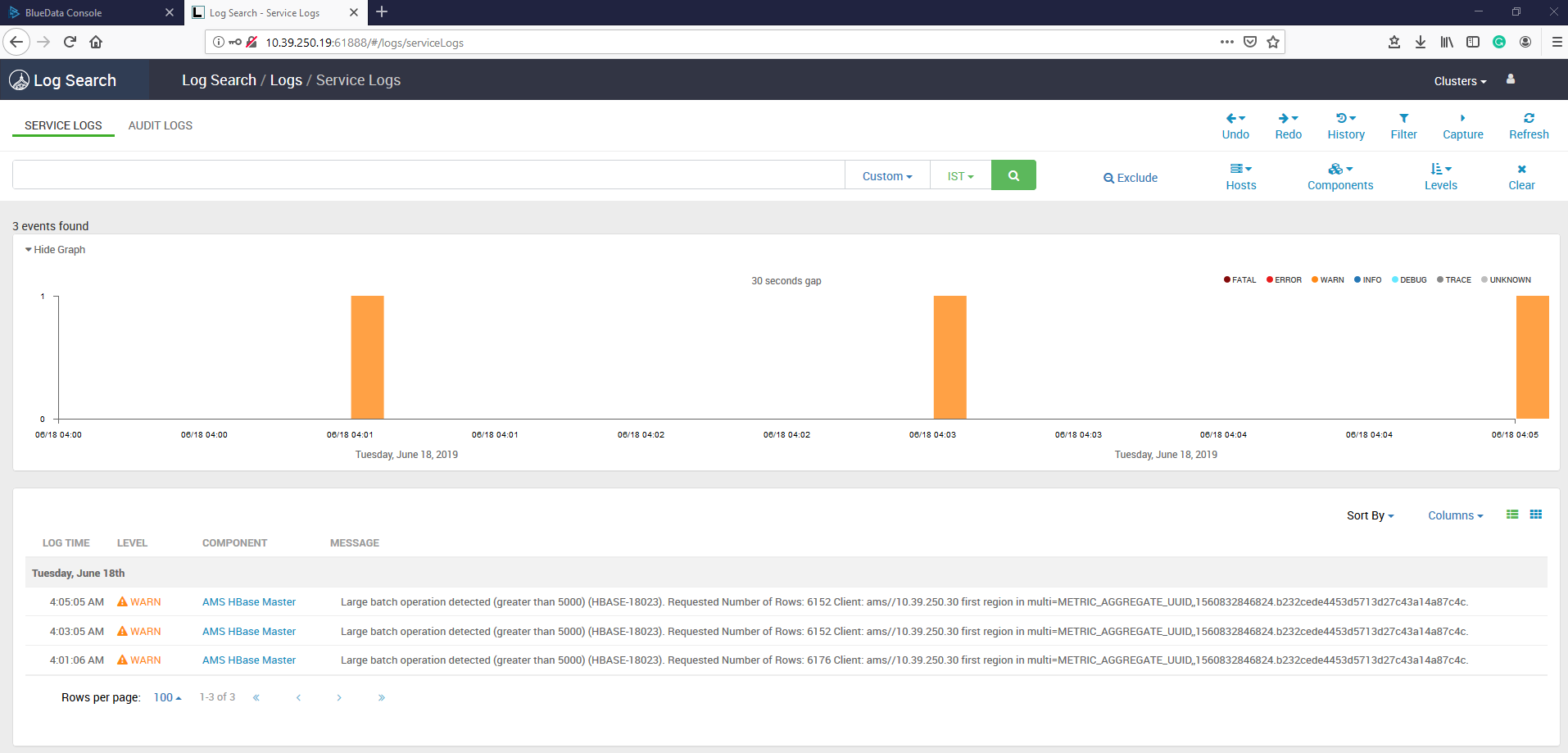


# testing log search

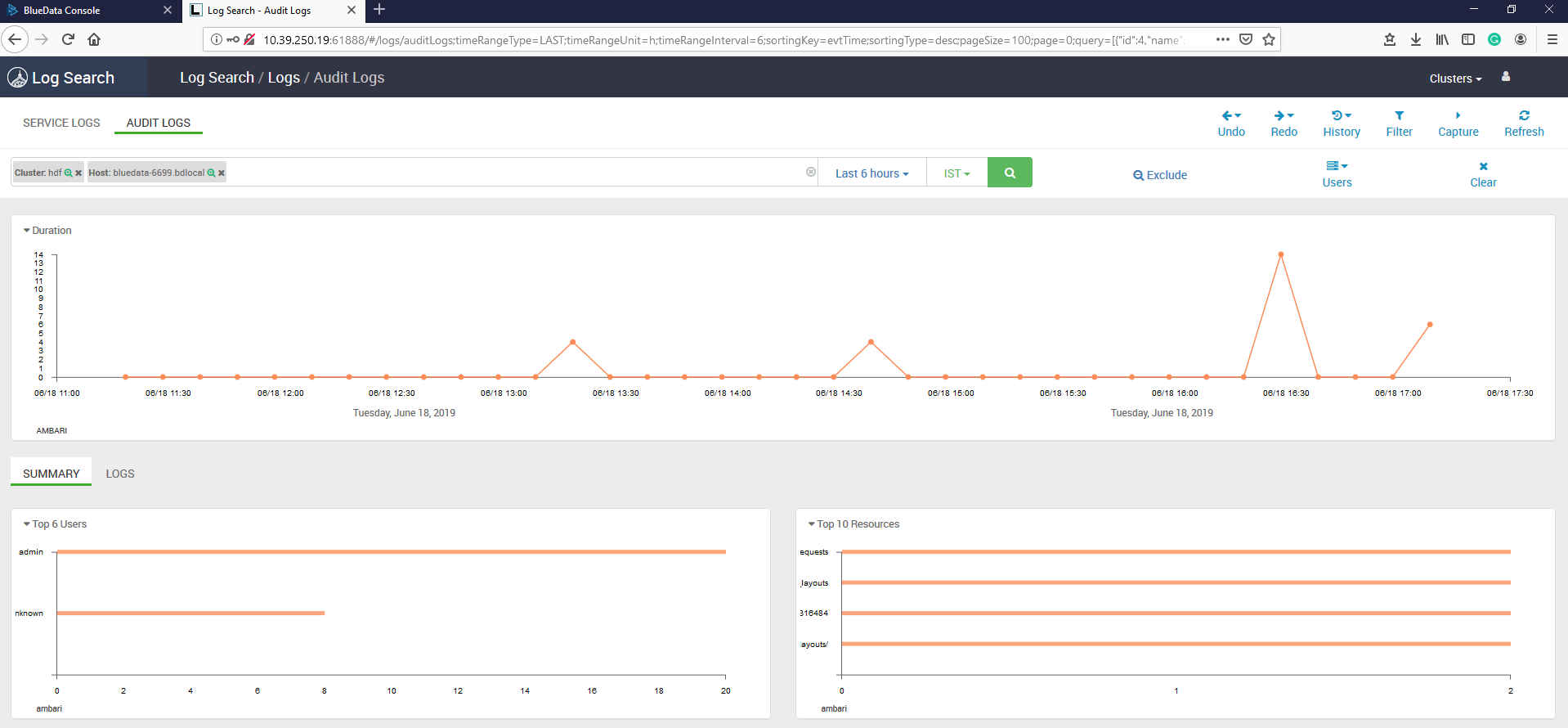
Go to the Log search UI in HDF cluster, enter username/password – admin/admin

In Log search you can track service and audit logs for HDF component

## Log search dashboard for service logs

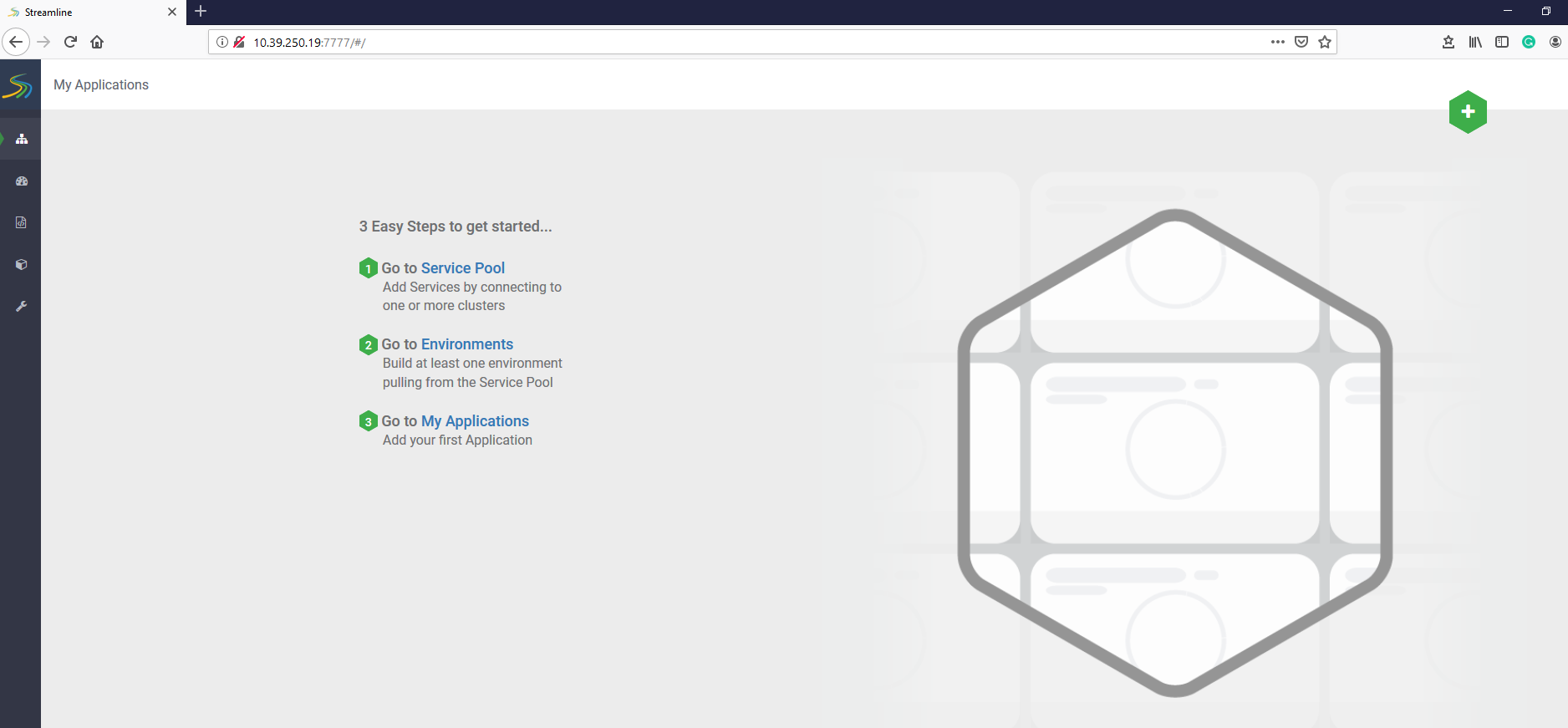


## Log search dashboard for audit logs

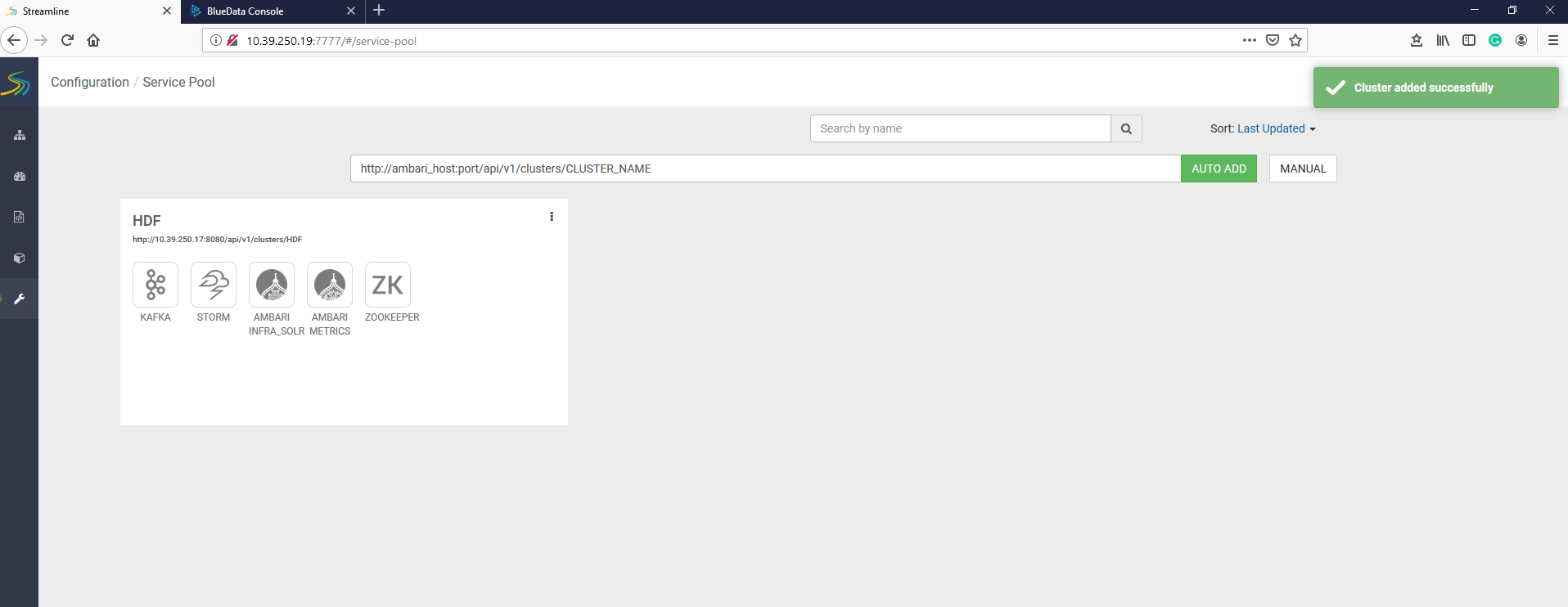


# testing Streaming Analytics Manager

## Go to the Streamline UI



## Create a service pool

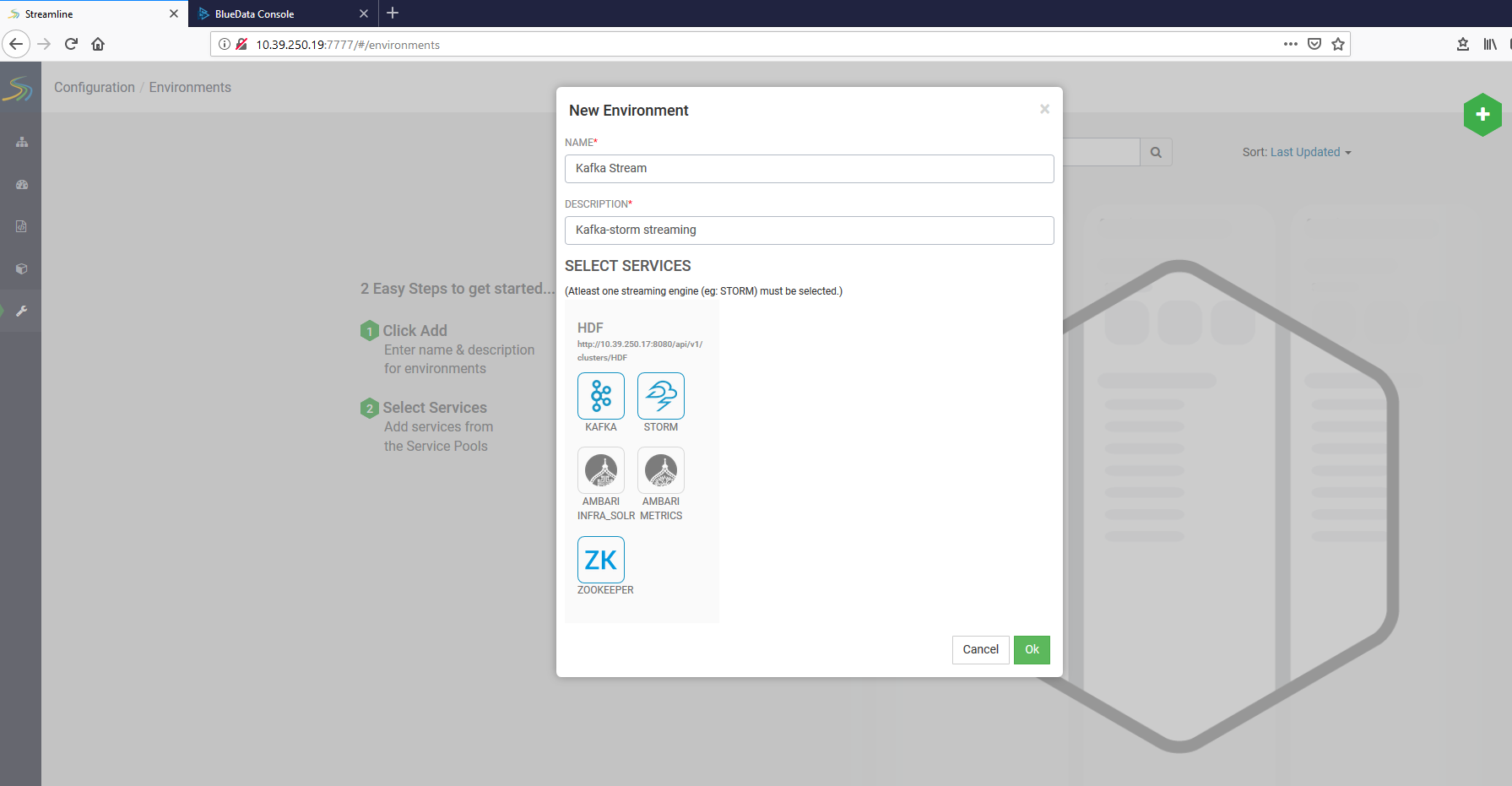


Perform below steps to create a service pool for HDF Cluster:

Enter url for HDF cluster into url bar and click on AUTO ADD button.

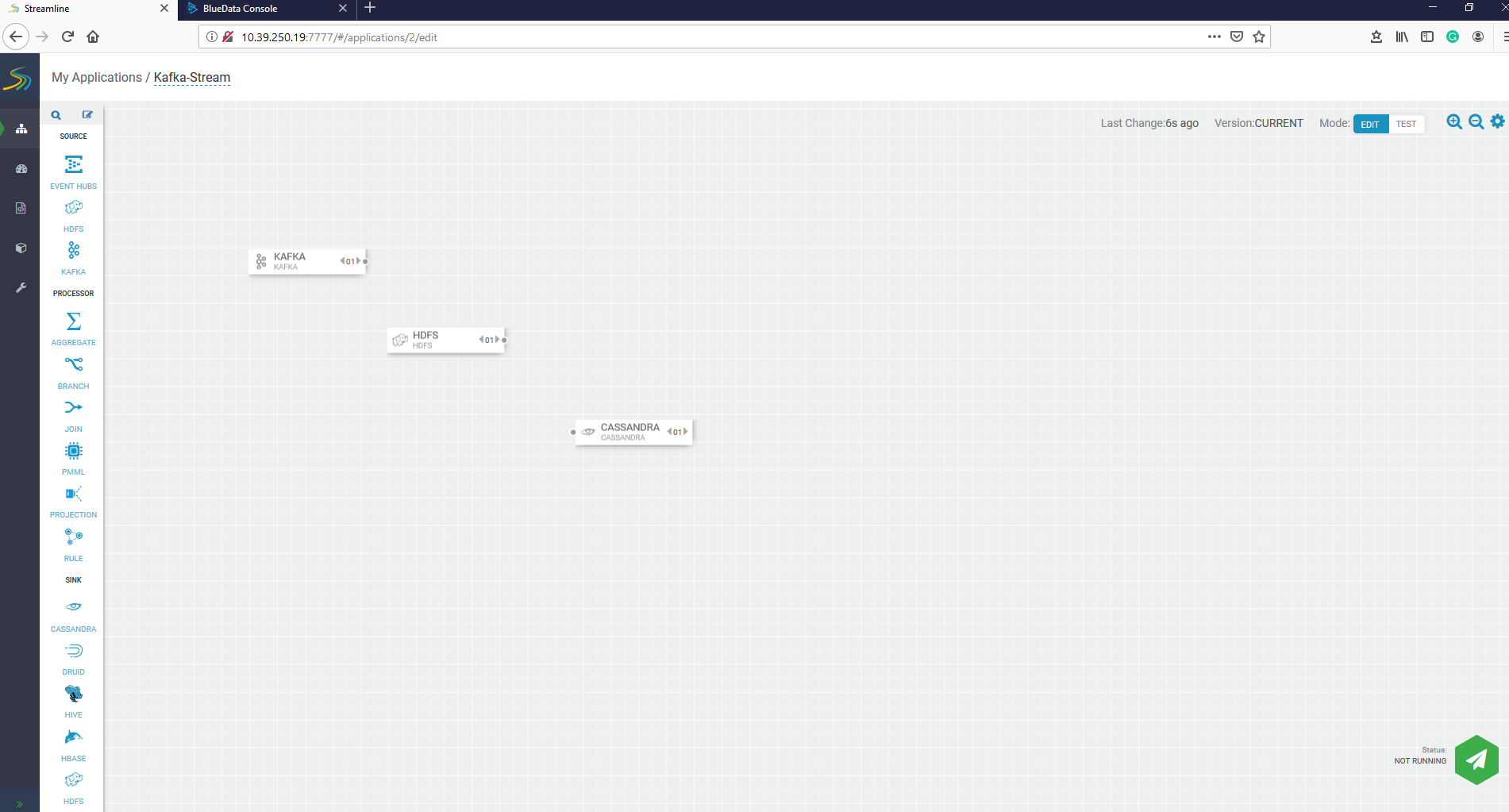
(Ex.: http://<ip address of ambari server>:8080/api/v1/clusters/HDF)

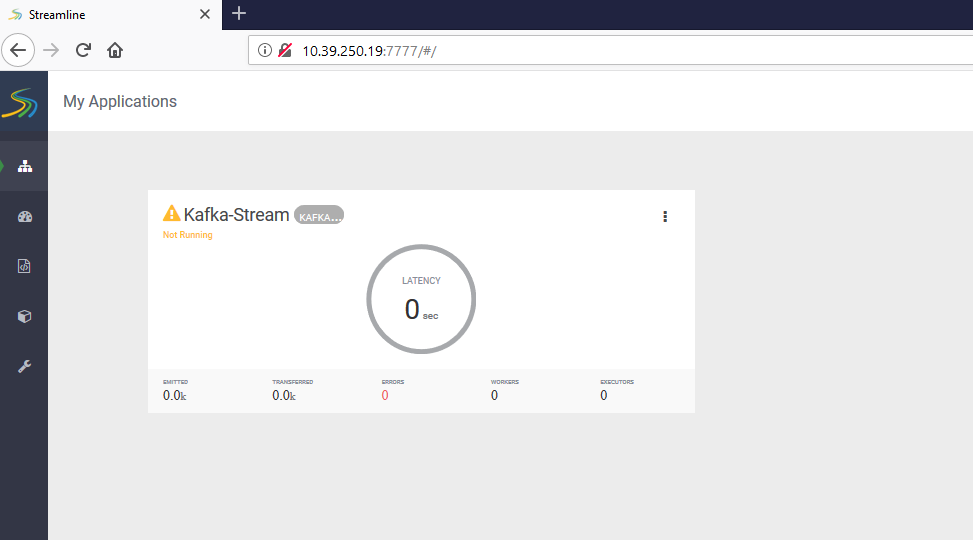
## Create a new environment for HDF cluster from service pool



## Create a new application for your environment

Here you can create and design your application. After running your appilcation you can verify your application under application section.

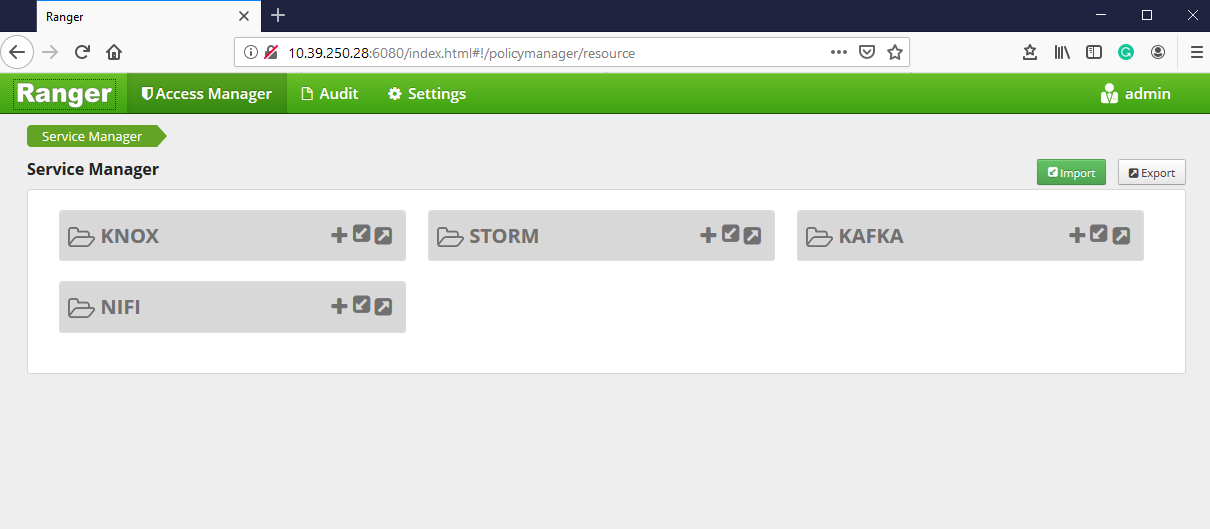


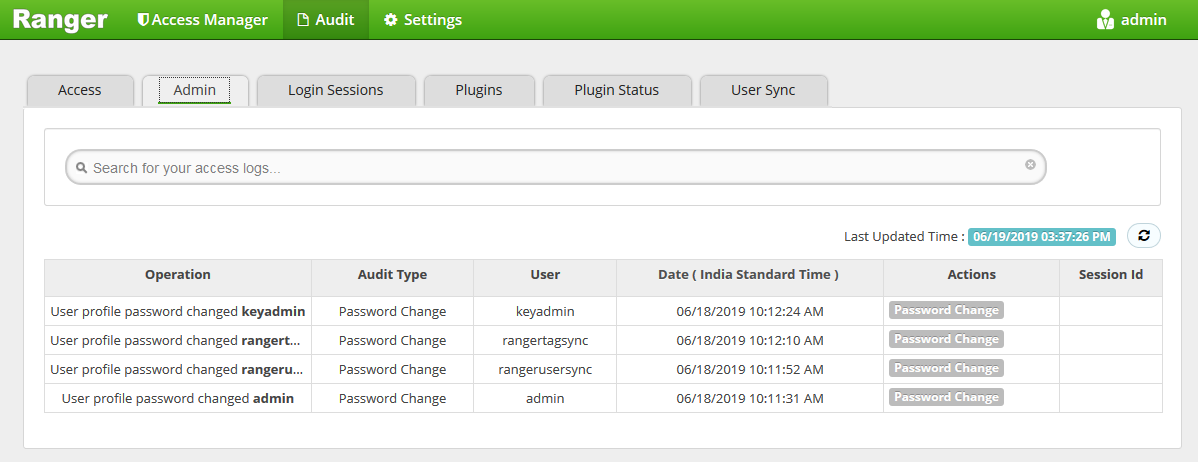


# testing ranger

Go to the Ranger UI in HDF cluster, enter username and password in login screen.

Service manager is available for Knox, Storm, Kafka and NIFI services. With the help of Ranger, you can create resource and tag based polices for different-different services.





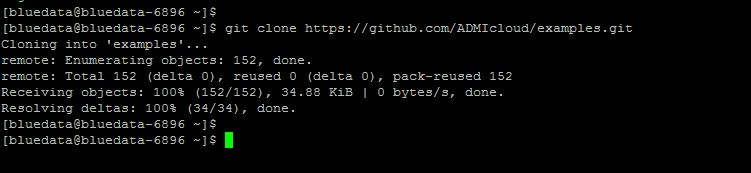
# testing storm

Here we will create a WordCount topology for Storm. We will download a WordCount example and create a WordCount topology.

## Download Storm WordCount example

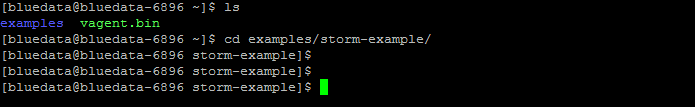
Execute the below command to download Storm WordCount example

**git clone** <https://github.com/ADMIcloud/examples.git>



## Change directory to the storm-example

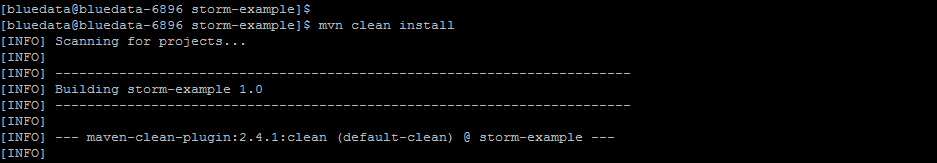
**cd examples/storm-example/**



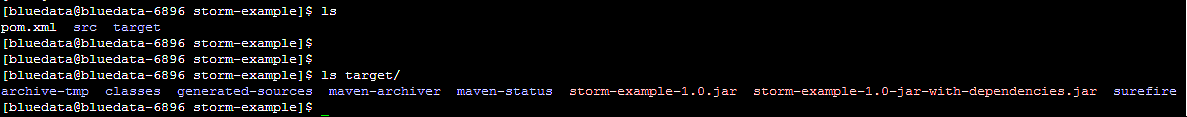
## Building jar

Execute the below command to build jar for Storm WordCount example

**mvn clean install**



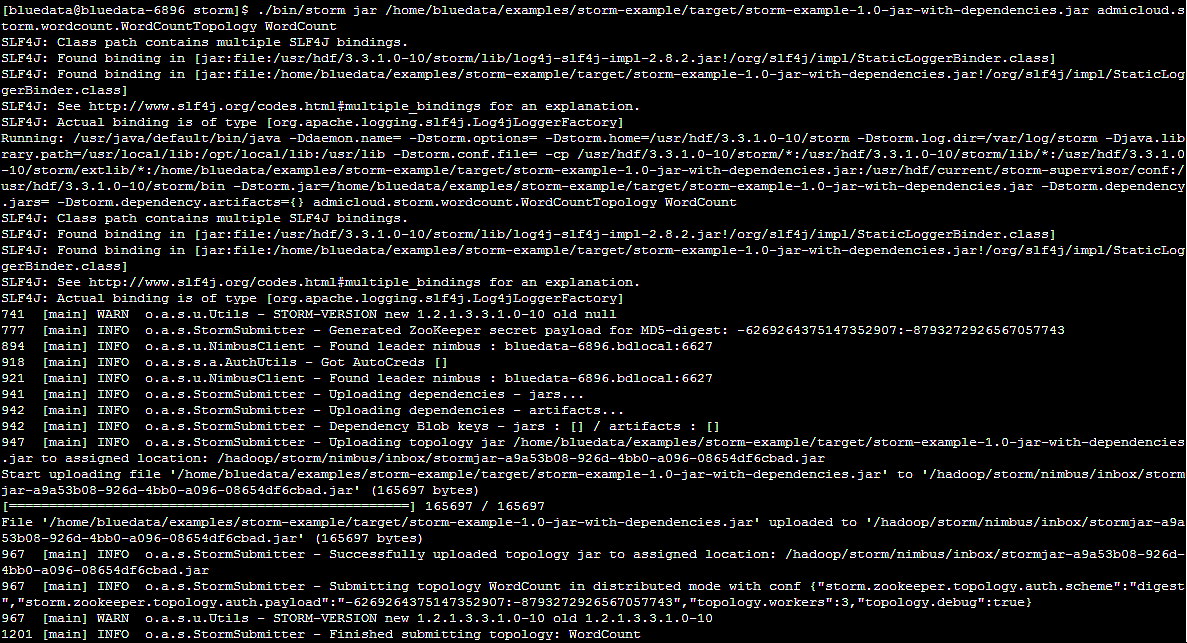
**Note:** After executing this command a new directory target will be created where you can find Strom WordCount example jar file. Use ls command to verify.



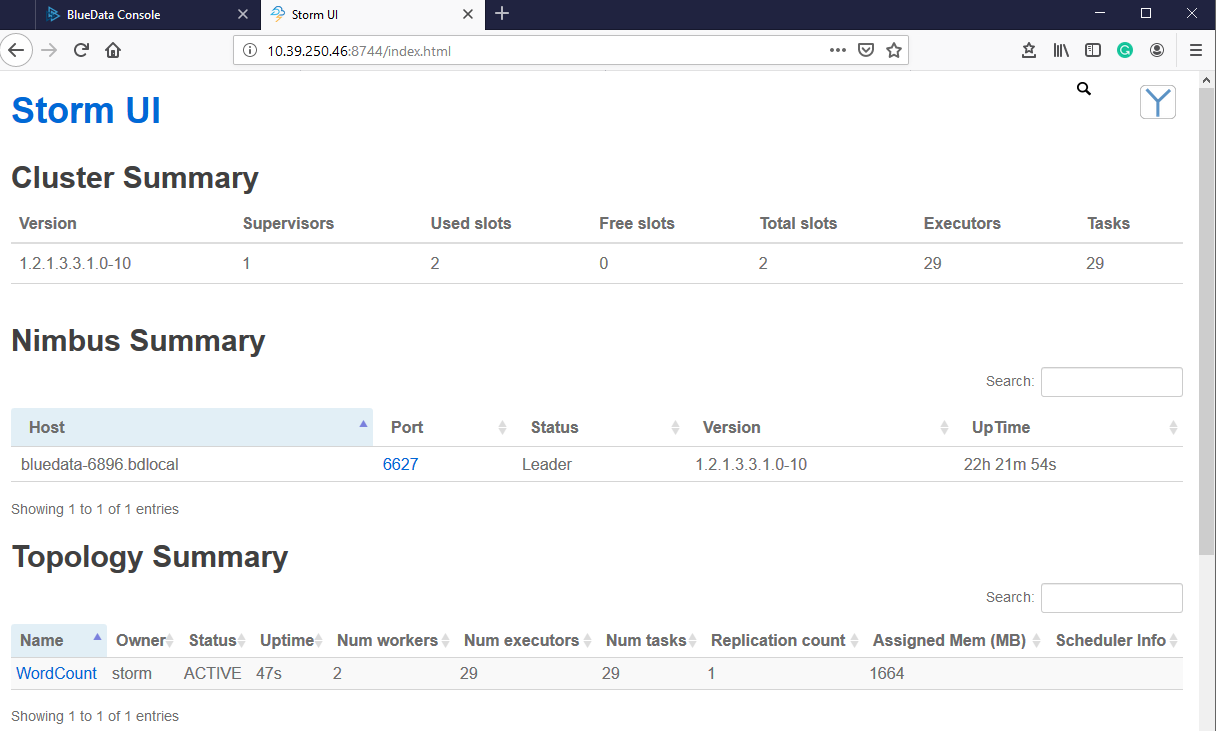
## Create topology for Storm WordCount example

Execute the below command to create topology for Storm WordCount example

**./bin/storm jar /home/bluedata/examples/storm-example/target/storm-example-1.0-jar-with-dependencies.jar admicloud.storm.WordCount.WordCountTopology WordCount**



## Verify WordCount topology from Storm UI



**Note:** Under Topology Summary WordCount topology is created