**To run alluxio on localhost, it is supposed that Hadoop and Spark have already being installed on the system.**

1. **First install Java 7 or newer**

* sudo add-apt-repository ppa:webupd8team/java
* sudo apt get update;
* sudo apt install oracle-java8-installer

1. **Install Alluxio**

* Download alluxio 1.3.0 from http://www.alluxio.org/download
* Extract the file where you want to extract by using the following command   
  🡪 tar -xzf alluxio-1.3.0-bin.tar.gz
* Go inside the extracted folder by using the following command :  
  🡪 cd alluxio-1.3.0
* Configure the files as per the commands  
  🡪 cp conf/alluxio-env.sh.template alluxio.env.sh

🡪 gedit conf/alluxio-env.sh  
Here change the master namenode at ALLUXIO\_MASTER\_HOSTNAME to localhost for standalone mode. Do remove the ‘#’ sign. Save & Exit.

* Configure the files as per the commands  
  🡪 cp conf/alluxio.site.prioperties.template  
  🡪 conf/alluxio.site.properties

🡪 gedit conf/alluxio-site.properties  
Change the alluxio.underfs.address=hdfs://localhost:9000/alluxio/data  
Do remove the ‘#’ sign. Save & Exit.

1. **Running Hadoop**

* Starting Hadoop on another terminal by following the commands.  
  🡪 cd /usr/local/hadoop  
  🡪 start-dfs.sh  
  Type jps to check hdfs is running or not.

1. **Running Alluxio**

* Now we have to run Alluxio on our system by following commands.  
  🡪 cd Desktop/alluxio  
  🡪 ./bin/alluxio format  
  🡪 ./bin/alluxio-start.sh local

1. **Linking Alluxio with HDFS**

* Now we are linking HDFS with Alluxio. For this install maven.  
  🡪 ./sudo apt-get install maven  
  🡪 mvn clean package -DskipsTests

The last command links HDFS with Alluxio.

* Configuring HDFS with Alluxio  
  🡪 mvn install –Phadoop-2.7 –Dhadoop.version=2.7.3 –DskipTests  
  The last command builds Alluxio server binaries to link with HDFS.

Inside the Alluxio folder  
🡪 cp conf/alluxio-site.properties.template conf/alluxio-site.properties.  
Here change the following  
🡪 alluxio.underfs.address=hdfs://NAMENODE:PORT  
Here the namenode port is of the hdfs master.

* Testing that alluxio is running or not by following commands. If the tests are passed then Alluxio master and Alluxio worker are running effieciently with Hadoop.  
  🡪 ./bin/alluxio runTests
* Creating a directory in hdfs for the linking of alluxio with HDFS by following commands in Hadoop terminal.  
  🡪 cd /hadoop\_data  
  🡪 hdfs dfs -mkdir /alluxio/data

Go in browser and type [https://localhost:19999](https://localhost:19999/) to check for alluxio in WebUI.

Go in browser and type [https://localhost:50070](https://localhost:50070/) to check for hadoop file system.  
Go to Utilities in that and go to Browser Directory. Check for /alluxio/data.  
For this test, you should see files named like: /default\_tests\_files/BasicFile\_STORE\_SYNC\_PERSIST

1. **Running Spark**

* Now we are running Spark on another terminal by typing the following commands.  
  🡪 cd /usr/local/spark  
  🡪 spark-shell

1. **Linking Spark with Alluxio**

* By following the command your alluxio will be linked to Spark.  
  🡪 mvn clean package -Pspark -DskipTests

🡪 cp spark/conf/spark-defaults.conf.template spark/conf/spark-defaults.conf

* Add the following line to spark/conf/spark-defaults.conf  
    
  spark.driver.extraJavaOptions –Dalluxio.zookeeper.address=zookeeperHost1:2181,zookeeperHost2:2181 –Dalluxio.zookeeper.enabled=true  
  spark.executor.extraJavaOptions –Dalluxio.zookeeperHost1:2181,zookeeperHost2:2181 –Dalluxio.zookeeper.enabled=true

Now Alluxio is linked with Spark and HDFS. You can do anything but faster than Spark which is already considered really fast.

1. **Stopping Spark**

* To stop Spark you to be inside spark folder and then type  
  🡪 ./sbin/stop-all.sh

1. **Stopping Alluxio**

* To stop Alluxio write the following command  
  🡪 ./bin/alluxio-stp.sh all

1. **Stopping Hadoop**

* You can stop Hadoop by typing the following command.  
  🡪 stop-dfs.sh