**Setting up Spark cluster in Multiple Computers**

Here is a step-by-step guide on how to set up a Spark cluster on individual systems:

1. Install Java 8 or higher on each system that will be part of the cluster. You can download and install Java from the [Oracle website](https://www.oracle.com/java/technologies/javase-downloads.html).
2. Install Spark on each system. You can download the latest version of Spark from the [Spark website](https://spark.apache.org/downloads.html).
3. Extract the downloaded Spark archive and move it to the directory where you want to install Spark. For example, you can extract it to the /opt directory:

**tar xvf spark-2.4.5-bin-hadoop2.7.tgz -C /opt**

1. Set the SPARK\_HOME environment variable on each system to point to the directory where you installed Spark. For example, you can add the following line to your .bashrc file:

**export SPARK\_HOME=/opt/spark-2.4.5-bin-hadoop2.7**

1. Add the $SPARK\_HOME/bin directory to your PATH environment variable. This will allow you to run Spark commands from the command line. For example, you can add the following line to your .bashrc file:

**export PATH=$SPARK\_HOME/bin:$PATH**

1. Configure the network settings on each system to allow communication between the systems. You will need to configure the firewall on each system to allow incoming and outgoing traffic on the ports used by Spark. The default port for the Spark master is 7077, and the default port for the Spark worker is 8888.
2. Start the Spark master on one of the systems. You can do this by running the following command:

**$SPARK\_HOME/sbin/start-master.sh**

1. Start the Spark worker on each system that will be part of the cluster. You can do this by running the following command, replacing MASTER\_IP with the IP address of the system where the Spark master is running:

**./sbin/start-worker.sh spark://10.0.0.51:7077 -c 6 -m 8G (execute this in all worker computers)**

The ./sbin/start-worker.sh command is used to start a Spark worker node on a cluster. The spark://10.0.0.51:7077 argument specifies the address of the Spark master node, which the worker will connect to in order to receive instructions and execute tasks. The -c 6 and -m 8G arguments specify the number of CPU cores and amount of memory, respectively, that the worker will use.

In this case, the command is being run on three computers, which means that three worker nodes will be started and connected to the Spark master. Each worker will use 6 CPU cores and 8 GB of memory. This will enable the worker nodes to execute tasks assigned by the master and process data in parallel, improving the performance of the Spark cluster.

1. You can now submit Spark jobs to the cluster using the spark-submit command. For example, to run a Spark shell on the cluster, you can run the following command:

**$SPARK\_HOME/bin/spark-shell --master spark://MASTER\_IP:7077**

**To Execute the pyspark in the master node**

**spark-submit --master spark://10.0.0.51:7077 --conf spark.ui.port=5051 sample2.py**

The spark-submit command is used to submit a Spark application for execution on a cluster. The --master flag specifies the address of the Spark master node in the cluster, and the --conf flag is used to set configuration options for the application. In this case, the spark.ui.port option is being set to 5051, which means that the user interface for the application will be available on port 5051. The sample2.py argument specifies the name of the Python file containing the application code that will be executed on the cluster.

When you run this command, it will submit the Spark application in the sample2.py file for execution on the cluster with the specified master node and configuration settings. The application will then be executed on the cluster, and the user interface will be available on the specified port.

**To start the worker node**

**./sbin/start-worker.sh spark://10.0.0.51:7077 -c 6 -m 8G (execute this in all worker computers)**

The ./sbin/start-worker.sh command is used to start a Spark worker node on a cluster. The spark://10.0.0.51:7077 argument specifies the address of the Spark master node, which the worker will connect to in order to receive instructions and execute tasks. The -c 6 and -m 8G arguments specify the number of CPU cores and amount of memory, respectively, that the worker will use.

In this case, the command is being run on three computers, which means that three worker nodes will be started and connected to the Spark master. Each worker will use 6 CPU cores and 8 GB of memory. This will enable the worker nodes to execute tasks assigned by the master and process data in parallel, improving the performance of the Spark cluster.