

Spark Administration



We can set up and launch a standalone cluster or set up on a single machine for the personal development or testing purpose. In this lab, we will be providing steps to configure spark on single node.

Apache Spark Installation

Make sure you have compatible java installed on your machine. You can verify it by typing command:

```
java -version
```

Apache spark setup has been downloaded on the following path and added to \$PATH variable:

```
/headless/Downloads/spark-2.4.7-bin-hadoop2.7/
```

Start Master

```
[user@hostname spark-2.4.7-bin-hadoop2.7]$ cd /headless/Downloads/spark-2.4.7-bin-hadoop2.7/  
[user@hostname spark-2.4.7-bin-hadoop2.7]$ ./sbin/start-master.sh
```

Inspect the response:

```
starting org.apache.spark.deploy.master.Master, logging to /headless/Downloads/spark-2.4.7-bin-hadoop2.7/logs/spark--org.apache.spark.deploy.master.Master-1-SandboxHost-637493255759703205.out
```

The screenshot shows a Linux desktop environment with a terminal window and a web browser window. The terminal window displays the command to start the Apache Spark master and its output. The Firefox window shows the Spark Master UI at `localhost:8080`, displaying basic metrics and application logs.

Terminal Output:

```
bash-4.2# pwd  
/headless/Downloads/spark-2.4.7-bin-hadoop2.7/sbin  
bash-4.2# ./start-master.sh  
starting org.apache.spark.deploy.master.Master, logging to /headless/Downloads/spark-2.4.7-bin-hadoop2.7/logs/spark--org.apache.spark.deploy.master.Master-1-SandboxHost-637493255759703205.out
```

Spark Master UI (localhost:8080):

- Master Metrics:** Alive Workers: 0, Cores in use: 0 Total, 0 Used, Memory in use: 0.0 B Total, 0.0 B Used, Applications: 0 Running, 0 Completed, Drivers: 0 Running, 0 Completed, Status: ALIVE.
- Workers (0):** Worker ID, Address.
- Running Applications (0):** Application ID, Name, Cores, Mem.
- Completed Applications (0):** Application ID, Name, Cores, Mem.

The UI also includes a terminal window where the user runs commands like `ps -ef | grep -i spark` and `grep -i spark` in the background.

Logs Path

Now inspect the `.out` file

```
cd /home/user/spark-2.4.7-bin-hadoop2.7/logs/
```

```
ls -ltr
```

```
cat filename
```

```
bash-4.2# pwd  
/headless/Downloads/spark-2.4.7-bin-hadoop2.7/logs  
bash-4.2#  
bash-4.2# bash-4.2# ls -ltr  
total 20  
-rw-r--r-- 1 root root 10454 Feb 19 13:00 spark--org.apache.spark.deploy.worker.Worker-1-SandboxHost-637493255759703205.out  
-rw-r--r-- 1 root root 5053 Feb 19 13:00 spark--org.apache.spark.deploy.master.Master-1-SandboxHost-637493255759703205.out  
bash-4.2#
```

```
bash-4.2# cat spark--org.apache.spark.deploy.master.Master-1-SandboxHost-637493255759703205.out  
Spark Command: /usr/java/jdk1.8.0_241-amd64/jre/bin/java -cp /headless/Downloads/spark-2.4.7-bin-hadoop2.7/conf/:/headless/Downloads/spark-2.4.7-bin-hadoop2.7/jars/* -Xmx1g org.apache.spark.deploy.master.Master --host SandboxHost-637493255759703205 --port 7077 --webui-port 8080  
=====  
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties  
21/02/19 12:00:08 INFO Master: Started daemon with process name: 6372@SandboxHost-637493255759703205  
21/02/19 12:00:08 INFO SignalUtils: Registered signal handler for TERM  
21/02/19 12:00:08 INFO SignalUtils: Registered signal handler for HUP  
21/02/19 12:00:08 INFO SignalUtils: Registered signal handler for INT  
21/02/19 12:00:08 WARN Utils: Your hostname, SandboxHost-637493255759703205 resolves to a loopback address: 127.0.0.1; using 192.168.0.209 instead (on interface eth0)  
21/02/19 12:00:08 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address  
21/02/19 12:00:09 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable  
21/02/19 12:00:09 INFO SecurityManager: Changing view acls to: root  
21/02/19 12:00:09 INFO SecurityManager: Changing modify acls to: root  
21/02/19 12:00:09 INFO SecurityManager: Changing view acls groups to:  
21/02/19 12:00:09 INFO SecurityManager: Changing modify acls groups to:  
21/02/19 12:00:09 INFO SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users with view permissions: Set(root); groups with view permissions: Set(); users with modify permissions: Set(root); groups with modify permissions: Set()  
21/02/19 12:00:09 INFO Utils: Successfully started service 'sparkMaster' on port 7077.  
21/02/19 12:00:09 INFO Master: Starting Spark master at spark://SandboxHost-637493255759703205:7077  
21/02/19 12:00:09 INFO Master: Running Spark version 2.4.7  
21/02/19 12:00:10 INFO Utils: Successfully started service 'MasterUI' on port 8080.  
21/02/19 12:00:10 INFO MasterWebUI: Bound MasterWebUI to 0.0.0.0, and started at http://192.168.0.209:8080  
21/02/19 12:00:10 INFO Master: I have been elected leader! New state: ALIVE  
21/02/19 12:02:51 INFO Master: Registering worker 192.168.0.209:33781 with 3 cores, 6.9 GB RAM  
21/02/19 12:06:03 INFO Master: Registering app Spark shell  
21/02/19 12:06:03 INFO Master: Registered app Spark shell with ID app-20210219120603-0000  
21/02/19 12:06:03 INFO Master: Launching executor app-20210219120603-0000/0 on worker worker-20210219120249-192.168.0.209-33781  
21/02/19 12:06:44 INFO Master: Received unregister request from application app-20210219120603-0000  
21/02/19 12:06:44 INFO Master: Removing app app-20210219120603-0000  
21/02/19 12:06:44 INFO Master: 127.0.0.1:58338 got disassociated, removing it.  
21/02/19 12:06:44 INFO Master: 192.168.0.209:41351 got disassociated, removing it.
```

Once started, the master will print out a spark://HOST:PORT URL for itself, which you can use to connect workers to it, or pass as the "master" argument to SparkContext. You can also find this URL on the master's web UI, which is <http://localhost:8080> by default.

Similarly, you can start one or more workers and connect them to the master via:

Start Worker

Start Worker and register the worker with master

Open <http://localhost:8080/> in browser and copy the master url

```
spark://hostname:7077
```

Connected (unencrypted) to SandboxHost-637493255759703205:1

Applications [Lab1 - Lab_1.md [Lab1]] Spark Master at spark://... Terminal [Terminal] 12:01 root

Spark Master at spark://SandboxHost-637493255759703205:7077 - Mozilla Firefox

localhost:8080

Spark 2.4.7

Spark Master at spark://SandboxHost-637493255759703205:7077

URL: [spark://SandboxHost-637493255759703205:7077](http://SandboxHost-637493255759703205:7077)

Alive Workers: 0

Cores in use: 0 Total, 0 Used

Memory in use: 0.0 B Total, 0.0 B Used

Applications: 0 Running, 0 Completed

Drivers: 0 Running, 0 Completed

Status: ALIVE

Workers (0)

Worker ID	Address	State	Cores	Memory

Running Applications (0)

Application ID	Name	Cores	Memory per Executor	Submitted Time	User	State	Duration

Completed Applications (0)

Application ID	Name	Cores	Memory per Executor	Submitted Time	User	State	Duration

Note: You can also use **localhost** because spark master is running on the same node.

now start the worker and register it with master using following command

```
./sbin/start-slave.sh spark://hostname:7077
```

Connected (unencrypted) to SandboxHost-637493255759703205:1

Applications [Lab1 - Lab_1.md [Lab1]] Spark Worker at 192.168.0.209:33781 Terminal 12:03 root

Spark Worker at 192.168.0.209:33781 - Mozilla Firefox

localhost:8080

Apache Spark 2.4.7

Spark Worker at 192.168.0.209:33781

ID: worker-20210219120249-192.168.0.209-33781

Master URL: [spark://SandboxHost-637493255759703205:7077](http://SandboxHost-637493255759703205:7077)

Cores: 3 (0 Used)

Memory: 6.9 GB (0.0 B Used)

[Back to Master](#)

Running Executors (0)

ExecutorID	Cores

bash-4.2# pwd
/headless/Downloads/spark-2.4.7-bin-hadoop2.7/sbin
bash-4.2# ./start-slave.sh spark://SandboxHost-637493255759703205:7077
starting org.apache.spark.deploy.worker.Worker, logging to /headless/Downloads/spark-2.4.7-bin-hadoop2.7/logs/spark--org.apache.spark.deploy.worker.Worker-1-SandboxHost-637493255759703205.out
bash-4.2#

Worker webUI: <http://localhost:8081>

Once you have started a worker, look at the master's web UI (<http://localhost:8080> by default). You should see the new node listed there, along with its number of CPUs and memory (minus one gigabyte left for the OS).

Finally, the following configuration options can be passed to the master and worker:

Argument	Meaning
<code>-h HOST, --host HOST</code>	Hostname to listen on
<code>-i HOST, --ip HOST</code>	Hostname to listen on (deprecated, use -h or --host)
<code>-p PORT, --port PORT</code>	Port for service to listen on (default: 7077 for master, random for worker)
<code>--webui-port PORT</code>	Port for web UI (default: 8080 for master, 8081 for worker)
<code>-c CORES, --cores CORES</code>	Total CPU cores to allow Spark applications to use on the machine (default: all available); only on worker
<code>-m MEM, --memory MEM</code>	Total amount of memory to allow Spark applications to use on the machine, in a format like 1000M or 2G (default: your machine's total RAM minus 1 GiB); only on worker
<code>-d DIR, --work-dir DIR</code>	Directory to use for scratch space and job output logs (default: SPARK_HOME/work); only on worker
<code>--properties-file FILE</code>	Path to a custom Spark properties file to load (default: conf/spark-defaults.conf)

Logs Path

```
cd /home/user/spark-2.4.7-bin-hadoop2.7/logs/
ls -ltr
cat filename
```

Now inspect the `.out` file, you will see the log like this:

```

2019-09-12 13:41:07 INFO Worker:2612 - Started daemon with process name:
144697@hostname 2019-09-12 13:41:07 INFO SignalUtils:54 - Registered signal handler
for TERM 2019-09-12 13:41:07 INFO SignalUtils:54 - Registered signal handler for HUP
2019-09-12 13:41:07 INFO SignalUtils:54 - Registered signal handler for INT 2019-09-12
13:41:08 WARN NativeCodeLoader:62 - Unable to load native-hadoop library for your
platform... using builtin-java classes where applicable 2019-09-12 13:41:08 INFO
SecurityManager:54 - Changing view acls to: user 2019-09-12 13:41:08 INFO
SecurityManager:54 - Changing modify acls to: user 2019-09-12 13:41:08 INFO
SecurityManager:54 - Changing view acls groups to: 2019-09-12 13:41:08 INFO
SecurityManager:54 - Changing modify acls groups to: 2019-09-12 13:41:08 INFO
SecurityManager:54 - SecurityManager: authentication disabled; ui acls disabled; users
with view permissions: Set(user); groups with view permissions: Set(); users with
modify permissions: Set(user); groups with modify permissions: Set() 2019-09-12
13:41:08 INFO Utils:54 - Successfully started service 'sparkWorker' on port 35633.
2019-09-12 13:41:08 INFO Worker:54 - Starting Spark worker 100.2.101.101:35633 with 32
cores, 124.6 GB RAM 2019-09-12 13:41:08 INFO Worker:54 - Running Spark version 2.3.4
2019-09-12 13:41:08 INFO Worker:54 - Spark home: /home/user/spark-2.4.7-bin-hadoop2.7
2019-09-12 13:41:08 INFO log:192 - Logging initialized @1510ms 2019-09-12 13:41:08
INFO Server:351 - jetty-9.3.z-SNAPSHOT, build timestamp: unknown, git hash: unknown
2019-09-12 13:41:08 INFO Server:419 - Started @1576ms 2019-09-12 13:41:08 INFO
AbstractConnector:278 - Started ServerConnector@3f9e3902{HTTP/1.1,[http/1.1]}
{0.0.0.0:8081} 2019-09-12 13:41:08 INFO Utils:54 - Successfully started service
'WorkerUI' on port 8081. 2019-09-12 13:41:08 INFO ContextHandler:781 - Started
o.s.j.s.ServletContextHandler@1dc21140{/logPage,null,AVAILABLE,@Spark} 2019-09-12
13:41:08 INFO ContextHandler:781 - Started
o.s.j.s.ServletContextHandler@5896ed4f{/logPage/json,null,AVAILABLE,@Spark} 2019-09-12
13:41:08 INFO ContextHandler:781 - Started
o.s.j.s.ServletContextHandler@1d9a25f0{/null,AVAILABLE,@Spark} 2019-09-12 13:41:08
INFO ContextHandler:781 - Started
o.s.j.s.ServletContextHandler@1ad57f24{/json,null,AVAILABLE,@Spark} 2019-09-12
13:41:08 INFO ContextHandler:781 - Started
o.s.j.s.ServletContextHandler@754605a4{/static,null,AVAILABLE,@Spark} 2019-09-12
13:41:08 INFO ContextHandler:781 - Started
o.s.j.s.ServletContextHandler@c5e9251{/log,null,AVAILABLE,@Spark} 2019-09-12 13:41:08
INFO WorkerWebUI:54 - Bound WorkerWebUI to 0.0.0.0, and started
athttp://hostname.com:8081 2019-09-12 13:41:08 INFO Worker:54 - Connecting to master
hostname.com:7077... 2019-09-12 13:41:08 INFO ContextHandler:781 - Started
o.s.j.s.ServletContextHandler@4ac9255f{/metrics/json,null,AVAILABLE,@Spark} 2019-09-12
13:41:08 INFO TransportClientFactory:267 - Successfully created connection to
hostname.com/199.6.212.152:7077 after 40 ms (0 ms spent in bootstraps) 2019-09-12
13:41:09 INFO Worker:54 - Successfully registered with master
spark://hostname.com:7077

```

Note that above scripts must be executed on the machine you want to run the Spark master on, not your local machine.

You can optionally configure the cluster further by setting environment variables in conf/spark-env.sh. Create this file by starting with the conf/spark-env.sh.template, and copy it to on your worker machine for the settings to take effect. The following settings are available:

Environment Variable	Meaning
SPARK_MASTER_HOST	Bind the master to a specific hostname or IP address, for example a public one.
SPARK_MASTER_PORT	Start the master on a different port (default: 7077).
SPARK_MASTER_WEBUI_PORT	Port for the master web UI (default: 8080).
SPARK_MASTER_OPTS	Configuration properties that apply only to the master in the form "-Dx=y" (default: none). See below for a list of possible options.
SPARK_LOCAL_DIRS	Directory to use for "scratch" space in Spark, including map output files and RDDs that get stored on disk. This should be on a fast, local disk in your system. It can also be a comma-separated list of multiple directories on different disks.
SPARK_WORKER_CORES	Total number of cores to allow Spark applications to use on the machine (default: all available cores).
SPARK_WORKER_MEMORY	Total amount of memory to allow Spark applications to use on the machine, e.g. 1000m, 2g (default: total memory minus 1 GiB); note that each application's <i>individual</i> memory is configured using its spark.executor.memory property.
SPARK_WORKER_PORT	Start the Spark worker on a specific port (default: random).
SPARK_WORKER_WEBUI_PORT	Port for the worker web UI (default: 8081).
SPARK_WORKER_DIR	Directory to run applications in, which will include both logs and scratch space (default: SPARK_HOME/work).
SPARK_WORKER_OPTS	Configuration properties that apply only to the worker in the form "-Dx=y" (default: none). See below for a list of possible options.
SPARK_DAEMON_MEMORY	Memory to allocate to the Spark master and worker daemons themselves (default: 1g).
SPARK_DAEMON_JAVA_OPTS	JVM options for the Spark master and worker daemons themselves in the form "-Dx=y" (default: none).
SPARK_DAEMON_CLASSPATH	Classpath for the Spark master and worker daemons themselves (default: none).
SPARK_PUBLIC_DNS	The public DNS name of the Spark master and workers (default: none).

Spark Shell

Your cluster on single node is ready now, you can test it using running command **spark-shell**

```
[user@hostname ~]$ spark-shell --master spark://hostname:7077
```

Note: You can also use **localhost** because spark master is running on the same node.

Reload the master webui. You will get one running application:

The screenshot shows a desktop environment with a terminal window and a web browser. The terminal window is titled 'Terminal' and shows a spark shell session. The browser window is titled 'Spark Master at spark://SandboxHost-637493255759703205:7077 - Mozilla Firefox' and displays the Apache Spark 2.4.7 master UI.

Spark Master UI Details:

- URL:** spark://SandboxHost-637493255759703205:7077
- Alive Workers:** 1
- Cores in use:** 3 Total, 3 Used
- Memory in use:** 6.9 GB Total, 1024.0 MB Used
- Applications:** 1 Running, 0 Completed
- Drivers:** 0 Running, 0 Completed
- Status:** ALIVE

Workers (1)

Worker Id
worker-20210219120249-192.168.0.209-33781

Running Applications (1)

Application ID	Name	Cores	Memory per Executor	Submitted Time	User	State	Duration
app-20210219120603-0000	(kill) Spark shell	3	1024.0 MB	2021/02/19 12:06:03	root	PENDING	0 s

Completed Applications (0)

Application ID	Name	Cores	Memory per Executor	Submitted Time	User	State	Duration

You can exit the spark shell using typing `:q` then enter

Task

- Run spark-shell command after removing `--master` parameter from the spark-shell command and verify that it does not appear in master or worker UI. Because we did not specify master url while running spark-shell command.

The screenshot shows a desktop environment with a terminal window and a web browser. The terminal window is titled 'Terminal' and shows a spark shell session. The browser window is titled 'Spark Master at spark://SandboxHost-637493255759703205:7077 - Mozilla Firefox' and displays the Apache Spark 2.4.7 master UI.

Spark Master UI Details:

- URL:** spark://SandboxHost-637493255759703205:7077
- Alive Workers:** 1
- Cores in use:** 3 Total, 0 Used
- Memory in use:** 6.9 GB Total, 0.0 B Used
- Applications:** 0 Running, 1 Completed
- Drivers:** 0 Running, 0 Completed
- Status:** ALIVE

Workers (1)

Worker Id
worker-20210219120249-192.168.0.209-33781

Running Applications (0)

Application ID	Name	Cores	Memory per Executor	Submitted Time	User	State	Duration

Completed Applications (1)

Application ID	Name	Cores	Memory per Executor	Submitted Time	User	State	Duration
app-20210219120603-0000	Spark shell	3	1024.0 MB	2021/02/19 12:06:03	root	FINISHED	41 s

Spark Submit

Now submit example application using spark-submit. Replace `spark://hostname:7077` with hostname of master node first.

```
spark-submit --class org.apache.spark.examples.SparkPi --master spark://hostname:7077  
/headless/Downloads/spark-2.4.7-bin-hadoop2.7/examples/jars/spark-examples_2.11-  
2.4.7.jar
```

Note: You can also use **localhost** as shown below because spark master is running on the same node:

```
spark-submit --class org.apache.spark.examples.SparkPi --master spark://localhost:7077  
/headless/Downloads/spark-2.4.7-bin-hadoop2.7/examples/jars/spark-examples_2.11-  
2.4.7.jar
```

If this example execute successfully, your spark installation is fine. You can see the results in console log

```
2019-09-12 13:53:27 INFO DAGScheduler:54 - Job 0 finished: reduce at SparkPi.scala:38,  
took 0.615754 sPi is roughly 3.1416557082785412 2019-09-12 13:53:27 INFO  
AbstractConnector:318 - Stopped Spark@6914bc2c{HTTP/1.1,[http/1.1]}{0.0.0.0:4040}
```

The screenshot shows a terminal window titled "Terminal" connected to a Spark Master at port 7077. The terminal displays the following command and its execution:

```
bash-4.2# spark-submit --class org.apache.spark.examples.SparkPi --master spark://SandboxHost-637493255759703205:7077 /headless/Downloads/spark-2.4.7-bin-hadoop2.7/examples/jars/spark-examples_2.11-2.4.7.jar
```

The output shows logs from the Spark Master and Worker, indicating the job has finished. The terminal also shows the Spark UI interface with sections for Workers, Running Applications, and Completed Applications.

Application ID	Name	Cores	Memory	Submitted Time	User	State	Duration
app-20210219120829-0001	Spark Pi	3	1024.0 MB	2021/02/19 12:08:29	root	FINISHED	6 s
app-20210219120603-0000	Spark shell	3	1024.0 MB	2021/02/19 12:06:03	root	FINISHED	41 s

Task

- 1) Run above spark-submit command but assign '2g' memory. You will get output after running spark-submit command as shown in the screenshot below:

Connected (unencrypted) to SandboxHost-637493255759703205:1 ()

12:10 root

Applications [Lab1 - Lab_1.md [Lab1]... Spark Worker at 192.168... Terminal

Spark Worker at 192.168.0.209:33781 - Mozilla Firefox

localhost:8081

Apache Spark 2.4.7

Spark Worker at 192.168.0.209:33781

ID: worker-20210219120249-192.168.0.209-33781
Master URL: spark://SandboxHost-637493255759703205:7077
Cores: 3 (0 Used)
Memory: 6.9 GB (0.0 B Used)

[Back to Master](#)

▼ Running Executors (0)

ExecutorID	Cores	State	Memory	Job Details	Logs
------------	-------	-------	--------	-------------	------

▼ Finished Executors (3)

ExecutorID	Cores	State	Memory	Job Details	Logs
0	3	KILLED	1024.0 MB	ID: app-20210219120603-0000 Name: Spark shell User: root	stdout stderr
0	3	KILLED	1024.0 MB	ID: app-20210219120829-0001 Name: Spark Pi User: root	stdout stderr
0	3	KILLED	2.0 GB	ID: app-20210219121014-0002 Name: Spark Pi User: root	stdout stderr

2) Run spark-submit command after removing `--master` parameter from the spark-submit command and verify that it does not appear in master or worker UI after refreshing the browser window.