

# Java-Success.com

Prepare to fast-track, choose & go places with 800+ Java & Big Data Q&As with lots of code & diagrams.

[Home](#) [Why? ▾](#) [300+ Java FAQs ▾](#) [300+ Big Data FAQs ▾](#) [Courses ▾](#)[👤 Membership ▾](#) [Your Career ▾](#)[Home](#) › [bigdata-success.com](#) › [Tutorials - Big Data](#) › [TUT - Spark Scala on Zeppelin](#) ›

01B: Spark on Zeppelin – custom Dockerfile

## 01B: Spark on Zeppelin – custom Dockerfile

 Posted on [July 22, 2018](#)

**Pre-requisite:** Docker is installed on your machine for Mac OS X (E.g. \$ brew cask install docker) or Windows 10. [Docker interview Q&As](#).

### What is Apache Zeppelin?

Zeppelin is a web based notebook to execute arbitrary code in Scala, SQL, Spark, etc. You can mix languages. Apache Zeppelin helps data analysts, data scientist, and business users to get better understanding of data. As described below you can quickly explore data, create visualizations and share

### 300+ Java Interview FAQs

300+ Java FAQs



16+ Java Key Areas Q&amp;As



150+ Java Architect FAQs



80+ Java Code Quality Q&amp;As



150+ Java Coding Q&amp;As



### 300+ Big Data Interview FAQs

300+ Big Data FAQs



Tutorials - Big Data

TUT -  Starting Big Data

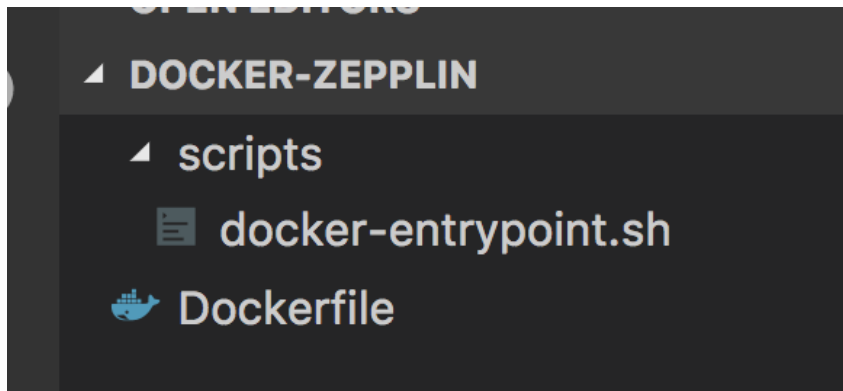
TUT - Starting Spark &amp; Scala

their insights, as web pages, with various stakeholders. For example

- 1) Prepare data using Shell by say downloading files with curl/wget, and then inject to HDFS.
- 2) Perform data analytics with Spark (i.e Scala) or pyspark (i.e. Python).
- 3) Perform simple visualizations in SQL.
- 4) Export the results with Shell, and publish to create graphs.

## How to install Apache Zeppelin on Docker?

You need around ~4GB disk space to create a Docker container with Ubuntu OS and Apache Zeppelin.



Apache Zeppelin on Docker

**Step 1:** Create a folder say “docker-zeppelin” under a folder named “projects”. Within the folder docker-zeppelin, create a file named “**Dockerfile**” and it should have the following contents. It installs Java on Ubuntu and then the Zeppelin note book.

TUT - Starting with Python

TUT - Kafka

TUT - Pig

TUT - Apache Storm

TUT - Spark Scala on Zeppelin

TUT - Cloudera

TUT - Cloudera on Docker

TUT - File Formats

TUT - Spark on Docker

TUT - Flume

TUT - Hadoop (HDFS)

TUT - HBase (NoSQL)

TUT - Hive (SQL)

TUT - Hadoop & Spark

TUT - MapReduce

TUT - Spark and Scala

TUT - Spark & Java

TUT - PySpark on Databricks

TUT - Zookeeper

## 800+ Java Interview Q&As

300+ Core Java Q&As



300+ Enterprise Java Q&As



150+ Java Frameworks Q&As



120+ Companion Tech Q&As



Tutorials - Enterprise Java



The Dockerfile shown below was simplified from the image “[apache/zeppelin](#)”, which is already available to be used from the Docker Hub. It installs all the conda packages, Python related packages, and “R” language related packages.

```
1 FROM ubuntu:16.04
2 MAINTAINER java-success.com
3
4 ENV ZEPPELIN_PORT 8080
5 ENV ZEPPELIN_HOME /usr/zeppelin
6
7 EXPOSE $ZEPPELIN_PORT
8
9 #install java
10 RUN apt-get update && \
11     apt-get install -y software-properties-common && \
12     add-apt-repository ppa:openjdk-r/ppa && \
13     apt-get update && \
14     apt-get install -y openjdk-8-jdk
15
16 #install other
17 RUN apt-get install -y \
18     npm \
19     vim \
20     wget
21
22 #install Zeppelin
23 RUN wget http://apache.mirror.amaaze.com.au/zeppelin-0.8.0-bin-all.tgz -C /usr/ && \
24     tar -zxf zeppelin-0.8.0-bin-all.tgz -C /usr/ && \
25     mv /usr/zeppelin* $ZEPPELIN_HOME
26
27 WORKDIR $ZEPPELIN_HOME
28
29 # Create a Zeppelin user
30 RUN useradd --home $ZEPPELIN_HOME --shell /bin/bash zeppelin
31 RUN chown zeppelin:zeppelin -R $ZEPPELIN_HOME
32
33
34 COPY scripts/docker-entrypoint.sh /docker-entrypoint.sh
35 RUN chmod 777 /docker-entrypoint.sh
36
37 ENTRYPOINT ["/docker-entrypoint.sh"]
38
```

The <https://zeppelin.apache.org/download.html> has the link to Zeppelin downloads. Get the binary package with all interpreters as shown above.

**Step 2:** Create a “scripts” folder, and create the “docker-entrpoint.sh” file with the following contents.

```
1 #!/bin/bash
2
3 # Start Zeppelin as "zeppelin" user.
4 su - zeppelin -c "$ZEPELIN_HOME/bin/zeppelin-daemon.sh start"
5
6 # Non-ending command to keep the container alive.
7 tail -F n0 /dev/null
8
```

**Step 3:** Create a docker image with the following command. The docker command looks for a “Dockerfile” in the current (i.e “.”) folder.

```
1 $ docker build -t zeppelin-simple:0.8.0 .
2
```

This may take a while to download the 1 GB zeppelin-0.8.0-bin-all.tgz. Once the image is created, you can verify it with:

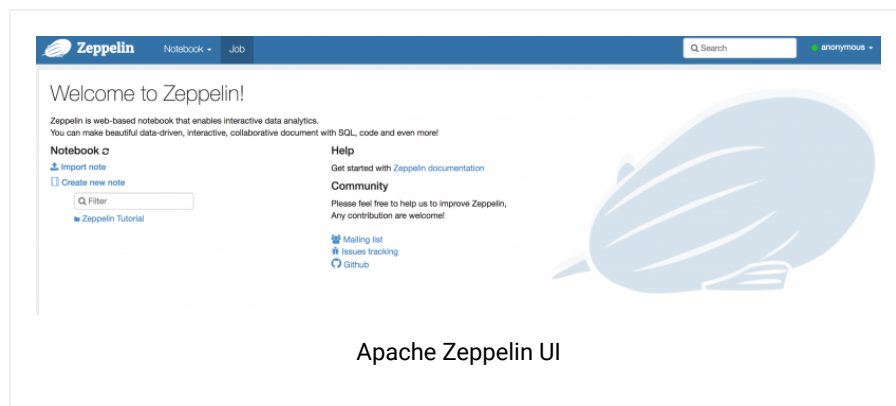
```
1 $ docker images
2
```

```
1
2 REPOSITORY          TAG
3 zeppelin-simple     0.8.0
4
```

**Step 4:** You can now run a docker container with this image as shown below:

```
1 $ docker run -it -p 8080:8080 zeppelin-simple:0.8.0
2
```

**Step 5:** Go to a browser and type:  
“http://localhost:8080”.



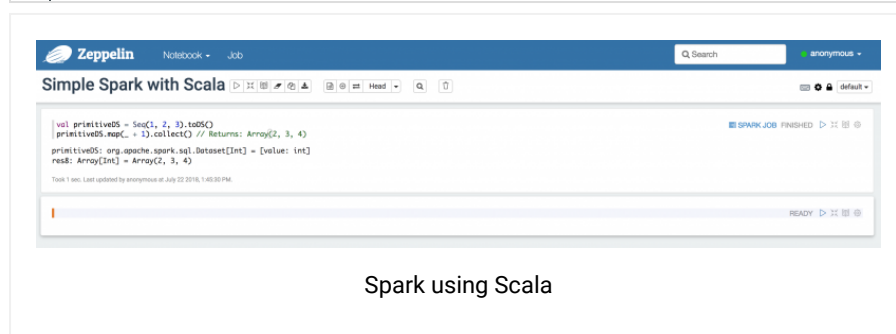
**Step 6:** Select the link “Create new note”, and name it “Simple Spark with Scala” and select the **interpreter** as “spark”.

Type the following simple Spark code to add 1 to the given set of numbers.

```
1 val primitiveDS = Seq(1, 2, 3).toDS()
2 primitiveDS.map(_ + 1).collect() // Returns: Array(2, 3, 4)
3
```

Press the play button, and the output will be:

```
1 Array(2, 3, 4)
2
```



Spark using Scala

◀ Event sourcing & CQRS interview Q&As

01A: Spark on Zeppelin – Docker pull from Docker hub ▶

## Disclaimer

The contents in this Java-Success are copyrighted and from EmpoweringTech pty ltd. The EmpoweringTech pty ltd has the right to correct or enhance the current content without any prior notice. These are general advice only, and one needs to take his/her own circumstances into consideration. The EmpoweringTech pty ltd will not be held liable for any damages caused or alleged to be caused either directly or indirectly by these materials and resources. Any trademarked names or labels used in this blog remain the property of their respective trademark owners. Links to external sites do not imply endorsement of the linked-to sites. [Privacy Policy](#).

© 2022 [java-success.com](https://www.java-success.com)