

[◀ Back to Week 1](#)[✕ Lessons](#)[Prev](#)[Next](#)

To make the learning process more convenient, our team has built Docker containers for each block of assignments. These containers are everything that you should install to pass all our courses. Having these containers deployed you can work with the same Jupyter notebooks as in Coursera's interface and be independent from the internet connection problems or any other problems.

To use our containers with Jupyter notebooks and appropriate Hadoop services installed, you should do the steps below. Firstly you should install the latest version of Docker. Then you should deploy the necessary containers from our [repository](#) in DockerHub. And finally if you want you can use a container in a terminal. It is an optional part of the guide because this is a good decision if you are interested to learn how Hadoop cluster works and not only to practice API of Hadoop services.

I. Installing Docker

For installing Docker and further using it with our containers for this course, you should have a machine answering the following requirements.

The minimal requirements for running notebooks on your computer are:

- 2-core processor
- 4 GB of RAM
- 20 GB of free disk space

The recommended requirements are:

- 4-core processor
- 8GB of RAM
- 50 GB of free disk space

Installing Docker on Unix

To install Docker on any of the most popular Linux operating systems (Ubuntu, Debian, CentOS, fedora etc) you should run the following command with sudo privileges:

```
1 curl -sSL https://get.docker.com/ | sh
```

Then optionally you can add your user account to the ***docker*** group:

```
1 usermod -aG docker $USER
```

This allows you to deploy containers without sudo but we don't recommend doing this on CentOS for security reasons.

Checking the installation

Now run the test container to check the installation successful:

```
1 docker run hello-world
```

You can see some Docker deployment logs on the screen and then **"Hello from Docker!"** message from successfully deployed container.

II. Deploying containers

You can find all of them in our Dockerhub repository by the following [link](#). Appropriate **<container_name>** is specified at the bottom of self-reading to each assignment. E.g. you can see "If you want to deploy the environment on your own machine, please use [bigdatateam/spark-course1](#) Docker container."

To run a Jupyter notebook on you own machine you should do the following steps.

1. Pull the actual version of the container from DockerHub

```
1 docker pull <container_name>
```

2. Deploy the container

```
1 docker run --rm -it -p <your_port>:8888 <container_name>
```

3. Open Jupyter environment

Now Jupyter can be opened in a browser by typing **localhost:<your_port>**. If you're working on Windows, you should open **<container_host>:<your_port>** instead of localhost. You can find **<container_host>** in the deployment logs of the container.

So now you have full environment with installed and adjusted Hadoop services, prepared datasets and demo codes on your own machine.

III. Working with docker through a terminal

To start the container and work with it via Unix terminal you should do the following steps.

1. Start a Tmux session:

```
1  tmux new -s my_docker
```

2. Run your container in terminal opened:

```
1  docker run --rm -it -p 8888:8888 -p 50070:50070 -p 8088:8088 bigdatateam/hdfs
    -notebook
```

(please, take into account that you can forward some additional ports with Hadoop UIs, not only Jupyter's port).

3. Detach the Tmux session by typing "Ctrl+b" and then "d".

4. Check your container's id:

```
1  docker ps
```

In the output of the command you should find something like this "**da1d48ac25fc**". This is your container's id.

5. Finally, you can open the terminal within the container by executing:

```
1  docker exec -it <YOUR_CONTAINER_ID> /bin/bash
```

6. Now you've logged in the container as root and you can execute Hadoop shell commands.

```
1  root@da1d48ac25fc:~# hdfs dfs -ls /data
2  Found 1 items
3  drwxrwxrwx - jovyan supergroup 0 2017-10-15 16:30 /data/wiki
4  root@da1d48ac25fc:~#
```

Simultaneously, you can work with Jupyter via browser.

To stop the docker container you should do the following steps.

1. Attach the Tmux session.

```
1  tmux a -t my_docker
```

(if you have forgotten the name of Tmux session, you can type 'tmux ls')

2. Stop the container via Ctrl+C.

3. Exit from the Tmux session.

We recommend to learn Tmux, it has many interesting features (e.g., you can start from this guide). However these 3 commands will be enough to work with our docker containers:

```
1  tmux new -s <session>
2  tmux a -t <sesscion>
3  tmus ls
```

✓ Complete

