

# Setting up Jethro Server using Docker - On Hadoop

[Docker](#) is a software container platform, which bundles only the libraries and settings, required to make a software run isolated on a shared operating system.

Running Jethro Server using Docker, guarantees that it will always run in the same manner, regardless of where it is deployed.

This article will explain in detail, step by step, how to configure a Jethro Server Docker container on a Linux machine:

- [Setup Jethro Docker](#)
  - [1. Install and start Docker](#)
  - [2. Download and Load the Image](#)
  - [3. Prepare folders to mount with the docker image file system](#)
  - [4. Plan the preferred configurations for running the image](#)
  - [5. Plan the preferred environment variables for Jethro](#)
  - [6. Create a file for the environment variables](#)
  - [7. Collect the image information](#)
  - [8. Create and start a Container](#)
- [Connecting to Jethro containers](#)
- [Maintenance](#)
- [Troubleshooting](#)
- [About the Images Content](#)
- [See Also](#)

## Setup Jethro Docker

### 1. Install and start Docker

```
yum install docker
service docker start
```

### 2. Download and Load the Image

In order to run a docker container, you should first have the image loaded into your local docker repository.

1. Sign in as user root (or a sudoer)
2. Download the image as .tar, according to your environment:

Environment	Command/Link
Hadoop - HDP	wget <a href="http://jethro.io/latest-docker-hdp">http://jethro.io/latest-docker-hdp</a>
Hadoop - CDH	wget <a href="http://jethro.io/latest-docker-cdh">http://jethro.io/latest-docker-cdh</a>

3. Run 'docker load' with the full name of the tar file downloaded. For example:

```
docker load --input jethro_docker-HDP-3.0.5-16389.tar
```



#### Please note

If you are not using Hadoop, please refer to the relevant guides for your environment:

[Setting up Jethro Server using Docker - On a Local File System](#)

[Setting up Jethro Server using Docker - On NFS](#)

### 3. Prepare folders to mount with the docker image file system

Since a docker container is a stateless independent file system, separated from the host's file system, it is recommended to create folders on the host's file system, and to mount them to the container's file system.

That way it would keep the information collected by Jethro persistent, even if the container will be lost.

The following code block will suggest a set of folder names to be used for the needs of Jethro's persistency, but you can also use other paths if you prefer so.

Please note that the 'cache' folder that Jethro uses for its instances, might be very big, depending on the data you will load, and the limits on cache size, which will be set during each instance creation/attachment. Make sure that the paths chosen will have enough space available for your current needs, and the future ones.

```
# create a main folder for all the sub folders described below #
mkdir /jethro_docker_volume

# create a folder for the instances configuration files #
mkdir /jethro_docker_volume/instances_opt

# create a folder for the instances cache #
mkdir /jethro_docker_volume/instances_cache

# create a folder for the instances logs #
mkdir /jethro_docker_volume/instances_logs

# create a folder for any other file you would like to keep persisted when the image will be removed/lost #
mkdir /jethro_docker_volume/persist

# give all users the permission to read write and execute the files within those folders #
chmod -R 777 /jethro_docker_volume/
```

## 4. Plan the preferred configurations for running the image

Docker allows multiple parameters of configuration (called 'OPTIONS'), to be set when running the image.

For Jethro Docker image, the following parameters needs to be defined, when the image will start to run:

1. **Container Name** - Decide on a name for the image container. Specifying a name gives the ability to use it when referencing the container within a Docker network, instead of using a long generated ID.  
Recommended name: '**jethroDocker**'.
2. **Ports Mapping** - Jethro exposes its services to external connections through ports. The ports which are exposed within the Jethro Docker image, needs to be mapped to ports that can be exposed on the host.
  - a. Normally, Jethro uses the following ports:
    - i. 9100 - For Jethro Manager.
    - ii. 9111-9200 - For the query engines of each instance.
  - b. SSH connections normally uses the port 22 (Not related to Jethro specifically, this is a port commonly used on most Linux environments for establishing a secured log in to the machine).
    - i. Since the SSH port used by the Host, is the same port used by the Docker image (22), it is recommended to map the Docker image SSH port, to a port address which is not in conflict with the Host one's (for example 9322).

Therefore, the recommended values to be used for the mapping are: **9100-9200:9100-9200** and **9322:22**

## 5. Plan the preferred environment variables for Jethro

In addition to the parameters of configuration, each specific docker image can also offer/require it's own environment variables. Jethro's variables are optional, but must be used in groups, according to the following groups of variables:

1. **Instance Details** - Jethro Docker image allows using environment variables to set the container already running, with a new instance, or with an existing instance attached. To do that, define the following variables:
  - a. **HADOOP\_NAME\_NODE\_ADDRESS** - The address of your Hadoop name node and the port (Usually 8020).
  - b. **INSTANCE\_NAME** - The desired/existing name of the instance. If this instance name already exist on the storage path provided (next variable), the instance will be to be attached. Otherwise, it will be created.
  - c. **INSTANCE\_STORAGE\_PATH** - The desired/exiting path of the instance storage on Hadoop. (For example: /user/jethro)
  - d. **INSTANCE\_CACHE\_PATH** - A local folder within the container's file system, that will be used locally for the caching needs of the instance on that image. If not provided, the following default (and recommended) path will be used: '/jethro/instance\_cache'.
  - e. **INSTANCE\_CACHE\_SIZE** - The maximum size of storage allowed for the Jethro Docker Image to be used for Instance caching. If not provided, the default value used will be 10GB.
2. **RUN\_JETHRO\_MANAGER** - TRUE/FALSE variable, which defines if Jethro Manager will also run within the image or not.
3. **SSH key for multiple containers** - If you want to assign the same Jethro SSH key for multiple containers (can be useful for Jethro Manager), you can set a path from which the private SSH key will be taken from, into the image container (Public SSH key will be generated based on the private one provided). The relevant environment variables to be set:
  - a. **KEY\_PATH** - The full path + file name of the Private SSH key. If the path of the key is on HDFS, make sure to provide a path that includes the ip and the port for HDFS (for example: `hdfs://127.0.0.1:8020/user/jethro/id_rsa`), otherwise the path should be '`/jethro/persist/<file-name>`', and the file should be placed ahead on the host's folder which is mapped to `/jethro/persist`.
  - b. **GENERAT\_KEY\_IF\_NOT\_EXIST** - If the path provided won't work, the container will fail to load up. But if the generate variable is set to TRUE, it will not fail, and it will generate a new key instead (both in the container, and on the provided key path, if the permissions allows it).
4. **Kerberos Intergration** - In order to run Jethro docker container on a kerberised Hadoop cluster, the following parameters must be set:
  - a. **KERBEROS\_SERVER** - kerberso server IP.
  - b. **KERBEROS\_DEFAULT\_RLM** - default Kerberos RLM to use.
  - c. **KERBEROS\_PRINCIPAL** - Jethro prinicipal (**must be created in advance**).

d. **KERBEROS\_KEYTAB\_PATH** - Jethro keytab file path (**must be created in advance and stored on one of the available docker volumes, e.g.: /jethro/persist**).

5. **Hive Intergration** - In order to set an Hive client configuration inside Jethro docker container, to be used for loading data in Jethro Manager, the following parameters must be set:

- a. **HIVE\_SERVER**- Hive server IP.
- b. **HIVE\_META\_STORE\_URI** - Hive meta-store URI.
- c. **HIVE\_USER** - user name ('hive' by default).

These Hive properties can be found on any Hive machine, inside the hive-site.xml file (usually located under /etc/hive/conf/). Within the file, look for the following properties respectively:

- javax.jdo.option.ConnectionURL
- hive.metastore.uris
- javax.jdo.option.ConnectionUserName

## 6. Create a file for the environment variables

Creating a file for the environment variables, allows the users to centralize all used variables, in a single persistent place.

The file should be created and stored, under the folder which was created for persistency purposes in step 3:

```
/jethro_docker_volume/persist/jethro_env.txt
```

Its content should be formed as in the example below (comments within the file are allowed):

```
# Jethro Docker Env Variables

# HADOOP parameters
HADOOP_NAME_NODE_ADDRESS=10.1.1.144:8020

# Auto create/attach instance parameters
INSTANCE_NAME=myinstance
INSTANCE_STORAGE_PATH=/user/jethro
INSTANCE_CACHE_PATH=/jethro/instance_cache
INSTANCE_CACHE_SIZE=20G

# SSH parameters
KEY_PATH=hdfs://10.1.1.144:8020/user/jethro/id_rsa
GENERAT_KEY_IF_NOT_EXIST=TRUE

# Jethro Manager parameters
RUN_JETHRO_MANAGER=true

# Kerberos parameters
KERBEROS_SERVER=12.345.678.90
KERBEROS_DEFAULT_RLM=EXAMPLE.COM
KERBEROS_PRINCIPAL=jethro
KERBEROS_KEYTAB_PATH=/jethro/persist/jethro.hadoop.keytab

# Hive parameters
HIVE_SERVER=98.765.43.21
HIVE_META_STORE_URI=thrift://98.765.43.21:9083
HIVE_USER=hive
```

## 7. Collect the image information

To run the Docker container, we will need to collect two parameters:

- 'IMAGE REPOSITORY'
- 'TAG'

Those can be found by running the following command:

```
docker images
```

The result should look like:

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
jethrodata/jethro	HDP-3.0.5-16389	7e34b3ebce49	3 months ago	736MB

## 8. Create and start a Container

Now that we have prepared the folders for mounting, the instance name, the ports mapping, the values for the volumes mount, and the image information, we are ready to hit the 'run' command. The basic 'docker run' command takes this form:

```
docker run [OPTIONS] IMAGE[:TAG|@DIGEST] [COMMAND] [ARG...]
```

On this document, only the parameters required for a Jethro Server docker will be described. A full documentation of the command, can be found [here](#).

OPTION	DESCRIPTION
-d	<b>Run the container in the background in a “detached” mode</b>
--privileged	<b>Give extended privileges to this container</b> By default a container is not allowed to access any device. A “privileged” container is given access to all devices on the host (as well as set some configuration in AppArmor or SELinux) to allow the container nearly all the same access to the host as processes running outside containers on the host.
--name	<b>Specifying a name gives the ability to use it when referencing the container within a Docker network</b>
-p hostPort:containerPort	<b>Publish a containers port or a range of ports to the host.</b> Format: hostPort:containerPort . Both hostPort and containerPort can be specified as a range of ports. When specifying ranges, the number of container ports in the range must match the number of host ports in the range.
--v host-src:container-dest	<b>Bind mount a volume</b>
-e	<b>Sets environment variables</b>
--env-file	<b>Read in a file of environment variables</b>

Jethro Docker image is required to run in 'privileged' mode (--privileged), and in a 'detached' mode (-d).

The rest of the information, parameters and variables that were collected, should be executed within the 'run' command, according to the syntax above.

**For example (HDP):**

```
docker run -d --privileged --name jethroDocker -p 9100-9200:9100-9200 -p 9322:22
--env-file /jethro_docker_volume/persist/jethro_env.txt
-v /jethro_docker_volume/persist:/jethro/persist
-v /jethro_docker_volume/instances_opt:/opt/jethro/instances
-v /jethro_docker_volume/instances_cache:/jethro/instance_cache
-v /jethro_docker_volume/instances_logs:/var/log/jethro
jethrodata/jethro:HDP-3.0.5-16389
```

## Connecting to Jethro containers

To connect to the container, or to interact with it, there are two methods available:

- 1) SSH - use the IP of the machine, port 9322 (unless if you decided to change it), and the credentials: user jethro, password jethro.
- 2) Bash - You can use the local machine to connect to the Docker machine, and run shell or bash commands on it. To do so:
  - Run 'docker ps' and get the container-name, or container-id
  - Run 'docker exec -it <container-name-or-id> bash' or 'docker exec -it jethroDocker sh'For example:

```
docker exec -it jethroDocker bash
docker exec -it 4e51f73265a7 sh
```

## Maintenance

**docker stop <CONTAINER>** - Stop a Container

**docker start <CONTAINER>** - Start a Container

**docker rm <CONTAINER>** - Remove a Container

**docker rmi <IMAGE>** - Remove an Image

To collect information about the list of images loaded on the host, Run:

```
docker images
```

It will show all top level images, their repository and tags, when they were created, and their size.

The tag column will include the Jethro Server version.

To collect information about the list of containers running on the host, Run:

```
docker ps
```

It will show only running containers by default. To see all containers: docker ps -a

## Troubleshooting

If you can't connect to the server or to any of the instances, make sure that:

- 1) The mapped ports of these instances are open.
- 2) The server is open for SSH communication on the mapped port for SSH.

## About the Images Content

HDP

Subject	Package
OS	Centos 7.x
Java	java-1.8.0-openjdk
Services	systemd
	initscripts
SSH	openssl
	openssh
	openssh-server
	openssh-clients

Hadoop	HDP hadoop-client
Hive	HDP hive
Volumes	/jethro/persist
	/opt/jethro/instances
	/jethro/instance_storage
	/jethro/instance_cache
	/var/log/jethro
Exposed Ports	9100-9200 22

**CDH**

Subject	Package
OS	Centos 7.x
Java	java-1.8.0-openjdk
Services	systemd
	initscripts
SSH	openssl
	openssh
	openssh-server
	openssh-clients
Hadoop	CDH hadoop-client
Hive	CDH hive-jdbc
Volumes	/jethro/persist
	/opt/jethro/instances
	/jethro/instance_storage
	/jethro/instance_cache
	/var/log/jethro
Exposed Ports	9100-9200 22

**POSIX**

Subject	Package
OS	Centos 7.x
Java	java-1.8.0-openjdk
Utilities	which
Services	systemd
	initscripts
SSH	openssl
	openssh
	openssh-server
	openssh-clients

Volumes	/jethro/persist
	/opt/jethro/instances
	/jethro/instance_storage
	/jethro/instance_cache
	/var/log/jethro
Exposed Ports	9100-9200 22

## See Also

[Getting started](#)

[Installing Jethro](#)

[Installing Jethro on Hadoop](#)

[Setting up Jethro Server using Docker - On NFS](#)

[Setting up Jethro Server using Docker - On a Local File System](#)