

K2VIEW

FABRIC DOCKER INSTALLATION

fabric-6.4.2_47

4/5/2021

CONFIDENTIALITY

This document contains copyrighted work and proprietary information belonging to K2View.

This document and information contained herein are delivered to you as is, and K2View makes no warranty whatsoever as to its accuracy, completeness, fitness for a particular purpose, or use. Any use of the documentation and/or the information contained herein, is at the user's risk, and K2View is not responsible for any direct, indirect, special, incidental, or consequential damages arising out of such use of the documentation. Technical or other inaccuracies, as well as typographical errors, may occur in this Guide.

This document and the information contained herein and any part thereof are confidential and proprietary to K2View. All intellectual property rights (including, without limitation, copyrights, trade secrets, trademarks, etc.) evidenced by or embodied in and/or attached, connected, or related to this Guide, as well as any information contained herein, are and shall be owned solely by K2View. K2View does not convey to you an interest in or to this Guide, to information contained herein, or to its intellectual property rights, but only a personal, limited, fully revocable right to use the Guide solely for reviewing purposes. Unless explicitly set forth otherwise, you may not reproduce by any means any document and/or copyright contained herein.

Information in this Guide is subject to change without notice. Corporate and individual names and data used in examples herein are fictitious unless otherwise noted.

Copyright © 2021 K2View Ltd./K2VIEW LLC. All rights reserved.

The following are trademark of K2View:

K2View logo, K2View's platform.

K2View reserves the right to update this list from time to time.

Other company and brand products and service names in this Guide are trademarks or registered trademarks of their respective holders.

REVISION HISTORY

[illegible]

TABLE OF CONTENTS

1	PREREQUISITES	5
1.1	MINIMUM REQUIREMENT FOR DOCKER OVER CENTOS/REDHAT	5
1.2	MINIMUM REQUIREMENT FOR DOCKER-DESKTOP	6
2	USING THE CONTAINER	7
2.1	LOAD THE CONTAINER IMAGE	7
3	RUN SINGLE FABRIC, CASSANDRA INSTANCES:	8
4	APPENDIX A	9
4.1	HOW TO SAVE CHANGES THAT WARE DONE IN THE CONTAINER.....	10
4.2	HOW TO STOP/START CONTAINER.....	10

1 PREREQUISITES

The following is for Dev/training/Demo/small POV

the minimum requirement is:

- **Docker CE/EE** over **CentOs/RedHat 7.7** or higher
- Or **docker-desktop** for Windows or Mac

1.1 Minimum requirement for Docker over CentOs/RedHat

1. CentOS/RedHat 7.3 (or higher) Operating System with latest patches, for each:
 - Modern Xeon Processor.
 - 8 Physical Cores.
 - RAM 16.
 - HDD, free 200G for the Docker



NOTE:

- Minimum 10G RAM for running 1x fabric, 1x Cassandra
- it is RECOMMENDED that Docker host will be physical server and not a VM

2. **Docker CE/EE & docker-compose** (see Docker documentation online) must be installed, and assigned privileges to k2view user
3. NTP/chronyc should be enabled (see https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/system_administrators_guide/sect-using_chrony)
4. Installed the latest Docker CE/EE version (<https://docs.docker.com/install/linux/docker-ce/centos/>)



NOTE: the **/var/lib/docker** should have at list 200G of free

5. Add k2view user, and assigned the k2view user 50G of space

update the k2view user home directory and password!!

```
useradd -m -d /<update the path>/k2view/ k2view  
passwd k2view
```

6. Give the k2view user privileges to run **docker & docker-compose** via **SUDO**

```
## Allow root to run any commands anywhere
root    ALL=(ALL)        ALL
k2view  NOPASSWD: docker_ps
```

1.2 Minimum requirement for docker-desktop

1. Windows or Mac machine with 8 cores of CPU
2. Windows or Mac machine with minimum 16G of RAM
3. Windows or Mac machine with minimum 120G of free space

2 USING THE CONTAINER

2.1 Load the container image

1. Grab the latest version of Fabric, Cassandra, Kafka & Configuration files at:

cassandra	https://owncloud-bkp2.s3.amazonaws.com/adminoc/fabricint/cassandra/3.11.6/openjdk/D_k2v_cassandra_3.11.6_2.tar.gz
kafka	https://owncloud-bkp2.s3.amazonaws.com/adminoc/fabricint/kafka/5.5.1/D_k2view_kafka_5.5.1_2.tar.gz
ES	https://owncloud-bkp2.s3.amazonaws.com/adminoc/fabricint/ES/D_k2view_elasticsearch_7.6.0.tar.gz
fabric	https://owncloud-bkp2.s3.us-east-1.amazonaws.com/adminoc/fabricint/fabric_6.5/6.5.1/Server/fabric-6.5.1_47/D_k2view_fabric_6.5.1_2_HF1.tar.gz
docker-compose	https://owncloud-bkp2.s3.us-east-1.amazonaws.com/adminoc/fabricint/fabric_6.5/6.5.1/Server/fabric-6.5.1_47/compose_fabric_6.5.1.zip

2. Copy all to the k2view directory

```
cd ~/
wget https://owncloud-bkp2.s3.amazonaws.com/adminoc/fabricint/cassandra/3.11.6/openjdk/D_k2v_cassandra_3.11.6_2.tar.gz
wget https://owncloud-bkp2.s3.amazonaws.com/adminoc/fabricint/kafka/5.5.1/D_k2view_kafka_5.5.1_2.tar.gz
wget https://owncloud-bkp2.s3.amazonaws.com/adminoc/fabricint/ES/D_k2view_elasticsearch_7.6.0.tar.gz
wget https://owncloud-bkp2.s3.us-east-1.amazonaws.com/adminoc/fabricint/fabric_6.5/6.5.1/Server/fabric-6.5.1_47/D_k2view_fabric_6.5.1_2_HF1.tar.gz
```

3. Load Docker images:

```
docker load -i D_k2v_cassandra_3.11.6_2.tar.gz
docker load -i D_k2view_kafka_5.5.1_2.tar.gz
docker load -i D_k2view_elasticsearch_7.6.0.tar.gz
docker load -i D_k2view_fabric_6.5.1_2_HF1.tar.gz
```

You can see the images names by using the command “docker images” (bellow is example)

```
k2view@TicketNoAdminTrainingLinux212:~$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
k2view/kafka         5.5.1_2            17aa10f29504       6 days ago         2.7GB
k2view/cassandra     3.11.6_02          de0372344243       7 days ago         1.14GB
k2view/fabric        6.3.0_64           2ec6e8de9cbb       8 days ago         1.71GB
k2view/elasticsearch 7.6.0              c65954c8fa26       9 months ago       1.18GB
```

4. UNZIP Config archives:

```
unzip compose_fabric_6.5.1.zip
```

open the “.env” file and update the names of the images as needed

3 RUN SINGLE FABRIC, CASSANDRA INSTANCES:

1. Run docker-compose

```
cd compose_fabric_6.5.1
# run:
docker-compose up -d

# or

sudo /usr/local/bin/docker-compose up -d
```

```
k2view@TicketMoAdminTrainingLinux212:~/compose_fabric_6.3.0$ docker-compose up -d
Creating network "compose_fabric_630_fab-net" with driver "bridge"
Creating cassandra1 ... done
Creating fabric1 ... done
Creating kafka1 ... done
k2view@TicketMoAdminTrainingLinux212:~/compose_fabric_6.3.0$ docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS
NAMES
ae92e40cc344   k2view/fabric:6.3.0_64              "sh -c /usr/local/k2..." 15 seconds ago Up 13 seconds 0.0.0.0:3213->3213/tcp, 0.0.0.0:5124->5124/tcp
fabric1
b32b43f7b431   k2view/kafka:5.5.1_2                "sh /opt/apps/kafka/..." 15 seconds ago Up 13 seconds 7000-7001/tcp, 7199/tcp, 0.0.0.0:2181->2181/tcp, 9042/tcp, 9142/tcp, 0.0.0.0:9093->9093/tcp, 9160/tcp
kafka1
2b7bc1f5d69a1   k2view/cassandra:3.11.6_02          "sh /opt/apps/cassan..." 15 seconds ago Up 13 seconds 0.0.0.0:7199->7199/tcp, 7000-7001/tcp, 0.0.0.0:9042->9042/tcp, 9142/tcp, 0.0.0.0:9160->9160/tcp
cassandra1
```

2. Start Cassandra

```
docker exec -u=cassandra -it cassandra1 bash -l
# run:

cassandra
```

3. Start Kafka:

```
docker exec -it -u=kafka kafka1 bash -l -c '$K2_HOME/kafka/bin/zookeeper-server-start -daemon
$K2_HOME/kafka/zookeeper.properties'
sleep 3
docker exec -it -u=kafka kafka1 bash -l -c '$K2_HOME/kafka/bin/kafka-server-start -daemon
$K2_HOME/kafka/server.properties'
sleep 3
docker exec -it -u=kafka kafka1 bash -l -c '~/kafka/bin/zookeeper-shell localhost:2181 <<< "ls
/brokers/ids"'
```

4. Start Fabric:

```
#run:
docker exec -it fabric1 bash -l

cp -r fabric/config.template config && \
sed -i 's@#HOSTS=.*@HOSTS=172.29.0.116@' $K2_HOME/config/config.ini && \
sed -i 's@#USER=cassandra@USER=cassandra@' $K2_HOME/config/config.ini && \
sed -i 's@#BOOTSTRAP_SERVERS=localhost:9093@BOOTSTRAP_SERVERS=172.29.0.115:9093@' $K2_HOME/config/config.ini && \
sed -i 's@#BOOTSTRAP_SERVERS=localhost:9093@BOOTSTRAP_SERVERS=172.29.0.115:9093@' $K2_HOME/config/iifConfig.ini && \
k2fabric start && k2fabric cluster-status && exit
```


4 APPENDIX A

1. stop docker-compose

```
cd compose_fabric_6.5.1
run: docker-compose stop
```

2. Stop Cassandra service

```
docker exec -u=root -it cassandra bash
su - cassandra

## stop cassandra
nodetool -u k2view -pw Q1w2e3r4t5 flush && kill -9 $(ps aux | grep -i $INSLATT_DIR |
grep -v "pts/7" |awk {'print $2'})

## restart cassandra
nodetool -u k2view -pw Q1w2e3r4t5 flush && kill -9 $(ps aux | grep -i $INSLATT_DIR |
grep -v "pts/7" |awk {'print $2'}) && $INSLATT_DIR/cassandra/bin/cassandra
```

3. Stop / Restart Fabric:

```
docker exec -it fabric1 bash -l
k2fabric stop / restart
```

4.1 How to save changes that were done in the container

Changes in running container (like in ini files) are not save to the **docker images**, so there is case that you want to create new image in order to be able to start a new container from it in the future

1. Connect as k2view user, identified the container ID or name:

```
sudo docker ps
```

```
[root@docker01 gabio]# sudo docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
a8d0c4953648	k2view/etl:v2.5.5.4	"/bin/bash"	34 minutes ago	Up 34 minutes	0.0.0.0:3000->3000/tcp, 0.0.0.0:3210->3210/tcp, 0.0.0.0:4001->4001/tcp, 0.0.0.0:5432->5432/tcp, 0.0.0.0:10389->10389/tcp
etl-01					

```
[root@docker01 gabio]#
```

2. Run:

```
sudo docker commit adi1 k2view/adi:v2.6.4.2_6
```

```
## check the docker images for the new image:  
sudo docker images
```

```
[root@docker01 gabio]# sudo docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
k2view/etl	v2.5.5.3_1	3b8d508debd6	About a minute ago	5.2 GB
k2view/etl	v2.5.5.4	b5f0ca90ef56	3 hours ago	5.2 GB

4.2 How to stop/start container

1. Run **stop**:
`sudo docker stop [containerID/container name]`
2. Run **start**:
`sudo docker start [containerID/container name]`