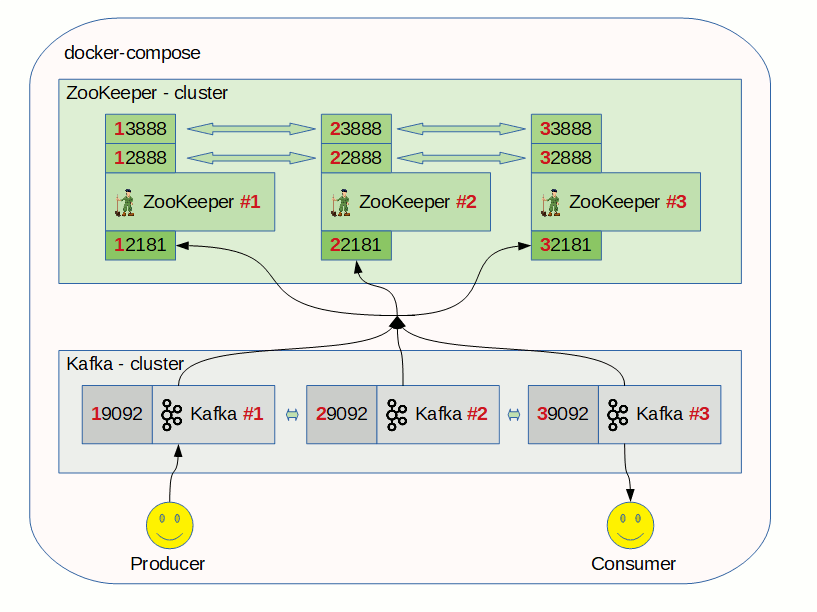
**Building Apache Kafka cluster using docker-compose and VirtualBox**



Apache Kafka is a an open-source stream-processing software platform, designed for high-throughput, low-latency and real-time data broadcasting. It’s provided by an easy-scalable and high-availability environment. Let’s see how to configure your own docker-compose recipe with fully functional Apache Kafka clustered environment just in few minutes.

**Overview**

* Preparing host machine for clustered environment using VirtualBox, docker and docker-compose.
* Creating docker-compose recipe file – step by step guide.
* The final version of Apache Kafka cluster docker-compose.yml file.
* Testing Apache Kafka cluster using kafkacat tool.

**1.Installing docker and docker-compose tools**

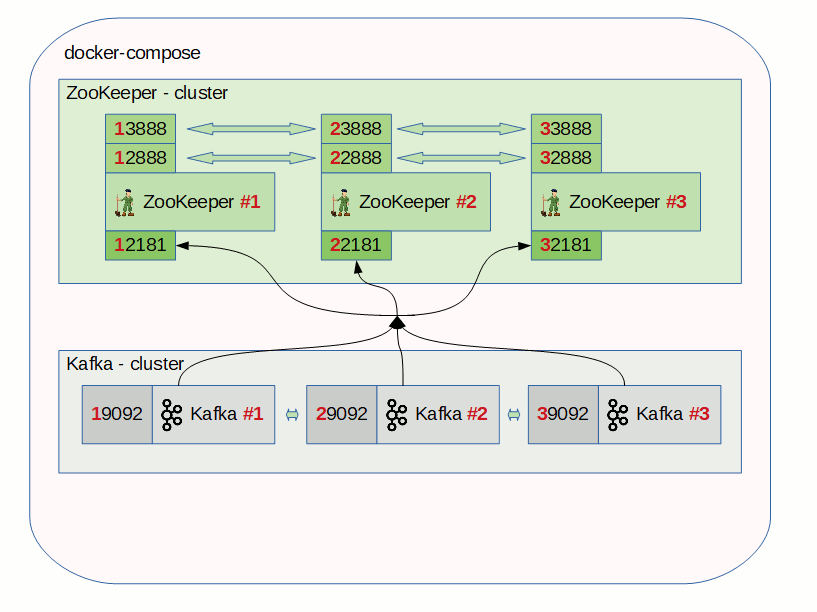
The next step is to install some necessary tools like docker and docker-compose which allow you to manage many different containers with complex dependencies in a very understandable and compact form.

sudo apt-get update

sudo apt-get install docker.io docker-compose

**2. Creating Apache Kafka cluster using docker-compose**

It is time to add three more containers to docker-compose.yml file which belongs to the Kafka cluster. Newly created servers are pointed to already prepared ZooKeeper cluster as it is shown on the image below.

[](https://better-coding.com/wp-content/uploads/2018/06/kafka-cluster-step2.2a.gif)

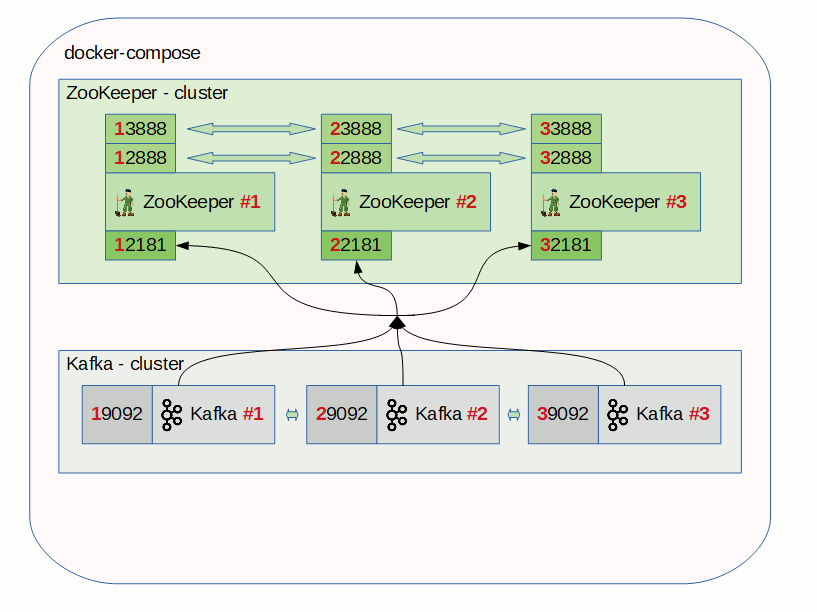
Each kafka node exposes #9092 client port.

**2.1 Apache Kafka cluster docker-compose.yml file**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82 | version: '2'  services:    zookeeper-1:      image: confluentinc/cp-zookeeper:latest      hostname: zookeeper-1      ports:        - "12181:12181"      environment:        ZOOKEEPER\_SERVER\_ID: 1        ZOOKEEPER\_CLIENT\_PORT: 12181        ZOOKEEPER\_TICK\_TIME: 2000        ZOOKEEPER\_INIT\_LIMIT: 5        ZOOKEEPER\_SYNC\_LIMIT: 2        ZOOKEEPER\_SERVERS: zookeeper-1:12888:13888;zookeeper-2:22888:23888;zookeeper-3:32888:33888      zookeeper-2:      image: confluentinc/cp-zookeeper:latest      hostname: zookeeper-2      ports:        - "22181:22181"      environment:        ZOOKEEPER\_SERVER\_ID: 2        ZOOKEEPER\_CLIENT\_PORT: 22181        ZOOKEEPER\_TICK\_TIME: 2000        ZOOKEEPER\_INIT\_LIMIT: 5        ZOOKEEPER\_SYNC\_LIMIT: 2        ZOOKEEPER\_SERVERS: zookeeper-1:12888:13888;zookeeper-2:22888:23888;zookeeper-3:32888:33888      zookeeper-3:      image: confluentinc/cp-zookeeper:latest      hostname: zookeeper-3      ports:        - "32181:32181"      environment:        ZOOKEEPER\_SERVER\_ID: 3        ZOOKEEPER\_CLIENT\_PORT: 32181        ZOOKEEPER\_TICK\_TIME: 2000        ZOOKEEPER\_INIT\_LIMIT: 5        ZOOKEEPER\_SYNC\_LIMIT: 2        ZOOKEEPER\_SERVERS: zookeeper-1:12888:13888;zookeeper-2:22888:23888;zookeeper-3:32888:33888      kafka-1:      image: confluentinc/cp-kafka:latest      hostname: kafka-1      ports:        - "19092:19092"      depends\_on:        - zookeeper-1        - zookeeper-2        - zookeeper-3      environment:        KAFKA\_BROKER\_ID: 1        KAFKA\_ZOOKEEPER\_CONNECT: zookeeper-1:12181,zookeeper-2:12181,zookeeper-3:12181        KAFKA\_ADVERTISED\_LISTENERS: PLAINTEXT://kafka-1:19092      kafka-2:      image: confluentinc/cp-kafka:latest      hostname: kafka-2      ports:        - "29092:29092"      depends\_on:        - zookeeper-1        - zookeeper-2        - zookeeper-3      environment:        KAFKA\_BROKER\_ID: 2        KAFKA\_ZOOKEEPER\_CONNECT: zookeeper-1:12181,zookeeper-2:12181,zookeeper-3:12181        KAFKA\_ADVERTISED\_LISTENERS: PLAINTEXT://kafka-2:29092      kafka-3:      image: confluentinc/cp-kafka:latest      hostname: kafka-3      ports:        - "39092:39092"      depends\_on:        - zookeeper-1        - zookeeper-2        - zookeeper-3      environment:        KAFKA\_BROKER\_ID: 3        KAFKA\_ZOOKEEPER\_CONNECT: zookeeper-1:12181,zookeeper-2:12181,zookeeper-3:12181        KAFKA\_ADVERTISED\_LISTENERS: PLAINTEXT://kafka-3:39092 |

**3. Testing Apache Kafka cluster using kafkacat tool**

In this step I will show you how to use kafkacat tool to test previously created Kafka cluster. We will send a message to the first node of the cluster and we will see if we will receive the same message from the third node of the cluster as it is shown on image below.

[](https://better-coding.com/wp-content/uploads/2018/06/kafka-cluster-step3.gif)

Please remember to add kafka-1, kafka-2 and kafka-3 hosts to the client /etc/hosts file.

/et/hosts file:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | 127.0.0.1 localhost      192.168.1.231 kafka-1 kafka-2 kafka-3    # The following lines are desirable for IPv6 capable hosts  ::1     ip6-localhost ip6-loopback |

Now install kafkacat using the following command:

Ubuntu: kafkacat installation

|  |  |
| --- | --- |
| 1 | sudo apt-get install kafkacat |
|  |  |

Run the following command to list all available brokers in the cluster:

|  |  |
| --- | --- |
| 1 | kafkacat -L -b kafka-1:19092 |

As you can see all of three nodes are accessible:

The result of command: kafkacat -L -b kafka-1:19092

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | better-coding@bc-vbox:~$ kafkacat -L -b kafka-1:19092  Metadata for all topics (from broker 1: kafka-1:19092/1):  3 brokers:    broker 2 at kafka-2:29092    broker 1 at kafka-1:19092    broker 3 at kafka-3:39092  2 topics:    topic "\_\_confluent.support.metrics" with 1 partitions:      partition 0, leader 2, replicas: 2,3,1, isrs: 2,3,1    topic "helloworld.t" with 1 partitions:      partition 0, leader 1, replicas: 1, isrs: 1 |

Open two instances of terminal and run:

Terminal #2: kafkacat in Producer mode

|  |  |
| --- | --- |
| 1 | kafkacat -P -b kafka-1:19092 -t helloworld\_topic |

Terminal #2: kafkacat in Consumer mode

Shell

|  |  |
| --- | --- |
| 1 | kafkacat -C -b kafka-3:39092 -t helloworld\_topic |

Then write some message to the first terminal, and you should see the same message in the second terminal.

Terminal #2: kafka Consumer output

|  |  |
| --- | --- |
| 1  2  3  4  5 | better-coding@bc-vbox:~$ kafkacat -C -b kafka-3:39092 -t helloworld\_topic  % Reached end of topic helloworld\_topic [0] at offset 0    test\_message  % Reached end of topic helloworld\_topic [0] at offset 1 |