|  |
| --- |
| Getting started with KafkaZookeeper and Docker |

Abstract

Kafka is a platform that can be used to produce and consume Messages/Events.

The elements of Kafka are: Brokers Topics, Partitions, Consumers, Producers.

Whether running within Docker or not, Kafka requires a separate Zookeeper server to track these elements.

Kafka can be used to support an Event-driven Messaging Platform (sometimes called an Event-sourcing Architecture).

To enable this architecture, an Overlay Network is necessary.

****

Establishing a Kafka Overlay Network (theory)

When you install Docker Engine it creates a default bridge network automatically.  Bridge networks are isolated networks on a single Engine installation.

To create a network that spans multiple Docker Hosts (each running a Docker Engine) instead create an Overlay Network, which sits atop of (overlays) the host-specific networks.

This Overlay Network allows containers connected to it – including swarm service containers – to communicate securely (when encryption is enabled).

Unlike a bridge network) an Overlay Network has a few prerequisites:

An Overlay Network requires a K/V store, such as Consul, Etcd, or Zookeeper.

Additionally, you need a cluster of hosts with connectivity to that K/V store.

Finally, a properly configured Engine daemon on each host in the cluster.

NB These are the dockerd options that support an Overlay Network:

--cluster-store

--cluster-store-opt

--cluster-advertise

### Create overlay networks with /24 blocks only (which is the default. This limits you to 256 IP addresses. If you need more than 256 IP addresses, do not increase the IP block size.

### Instead, use the dnsrr endpoint mode w/ an external Load Balancer, or use multiple small Overlay Networks.

ref <https://docs.docker.com/engine/reference/commandline/network_create/>

ref <https://docs.docker.com/network/overlay/>

****

Establishing a Kafka Overlay Network (practice)

The command to create an (overlay) network:

docker network create -d overlay --attachable kafka-net

… may fail initially:

A screenshot of a cell phone

Description automatically generated

Follow the interactive feedback/corrective instructions in response to any issues.

a. They involve first setting up Docker Swarm:

docker swarm init

b. Then, joining the manager for Docker Swarm:

docker swarm join-token manager

c. Issuing the above command should result in a response resembling:

docker swarm join --token SWMTKN-1-

5pg2wz03npf83foijp3417ks157j34l3eajwlutj1tafrkrlq6-

6yg32jn8r9gul1te1yuyu3ryl 192.168.65.3:2377

d. Now, rerun the command to create an (overlay) network:

docker network create -d overlay --attachable kafka-net

… with a response resembling:

zhd4d9ecqvnnz1ebev4wzwqb5

Installing Zookeeper

The first service to install on the Overlay Network is a single Zookeeper Container (has a WEB UI monitor).

Zookeeper is installed before Kafka because Kafka has a dependency on Zookeeper.

Here is the Zookeeper installation command:

docker service create \

--network kafka-net \

--name=zkui \

--publish 9090:9090 \

qnib/plain-zkui@sha256:30c4aa1236ee90e4274a9059a5fa87de2ee778d9bfa3cb48c4c9aafe7cfa1a13s8xydg5kw9n83fswj0o5fo703

A screenshot of a cell phone

Description automatically generated

****

Next, issue this Docker CLI command to verify that both Zookeeper images are running:

docker service ls --format 'table {{.Name}}\t{{.Replicas}}\t{{.Ports}}'

A screenshot of a cell phone

Description automatically generated

****

Next, open localhost:9090 to see the Zookeeper Web UI.

The login creds: admin/manager

A screenshot of a cell phone

Description automatically generated

****

Brokers

Bring up one broker to populate Zookeeper using these CLI options:

--hostname

Uses Docker Templates to derive the hostname from placement decisions.

This makes the containers identifiable

KAFKA\_BROKER\_ID

Pins the identifier of the broker to its slot-id – if a container fails, it’s replaced.

Since the KAFKA\_BROKER\_ID is set to the slot-id, it takes the place of the earlier one.

Otherwise, the broker-id is increased with each new broker.

ZK\_SERVERS points Kafka to the Zookeeper service.

Here is the CLI command to create a Broker:

docker service create \

--network kafka-net \

--name broker \

--hostname="{{.Service.Name}}.{{.Task.Slot}}.{{.Task.ID}}" \

-e KAFKA\_BROKER\_ID={{.Task.Slot}} -e ZK\_SERVERS=tasks.zookeeper \

qnib/plain-kafka:2018-04-25\_1.1.0qnib/plain-kafka:2018-04-25\_1.1.0 \

f9qcjupwr923mojm1rs2hvls9

****

Yahoo Kafka Manager

Here is the command to create a Yahoo (better) Kafka Manager:

docker service create --network kafka-net --name manager -e ZOOKEEPER\_HOSTS=tasks.zookeeper \

--publish=9000:9000 qnib/plain-kafka-manager:2018-04-25

A screenshot of a cell phone

Description automatically generated

****

Kafka Topics

Use the Kafka tooling within the broker image to create a first Topic:

****

Addendum

D

<https://success.docker.com/article/getting-started-with-kafka>

<https://docs.docker.com/engine/reference/commandline/network_create/>

****