Experiment: Container Communication Eavesdropping

Create Our Containers

For MacOS and Windows

For MacOS

Start a terminal and cd to ~/Projects

For Windows

Run Git Bash from the c:\Projects folder

For Windows and MacOS

Change to the kafka labs subfolder in the cloned repo lab folder.

For this simple experiment we'll use Docker Compose.

Note: The following docker-compose.yml is in the kafka folder in the repository, or you can paste this into a docker-compose.yml file

KubernetesNetworking/labs/kafka $ **docker-compose up -d**

```
[+] Running 15/15
 - zookeeper Pulled
91.6s
   - 4d0d850cd4ad Pull complete
16.6s
   - eb918a808c15 Pull complete
71.2s
   - 68bcf2239ce4 Pull complete
72.1s
   - 2e31c2ab64ea Pull complete
72.3s
   - 5dcd26e8f603 Pull complete
74.6s
   - 27267efe7f14 Pull complete
74.7s
 - kafka Pulled
91.6s
   - 96965a3a8424 Pull complete
16.5s
   - 7a73120408f4 Pull complete
70.8s
```

```
   - 5c2fffeabbf7 Pull complete
71.3s
   - b479bf09eedc Pull complete
72.2s
   - e5161e1fdbdc Pull complete
72.4s
   - e0f07497560d Pull complete
74.6s
   - b88780ca570c Pull complete
74.7s
[+] Running 3/3g
16.9s
 - Network kafka_default        Created
0.9s
 - Container kafka_zookeeper_1  Started
3.0s
 - Container kafka_kafka_1      Started
```

Validate our Zookeeper and Kafka services are running

If you have Netcat installed you could use that for service validation.

$ nc -z localhost 22181

```
Connection to localhost port 22181 [tcp/*] succeeded!
```

$ nc -z localhost 29092

```
Connection to localhost port 29092 [tcp/*] failed!
```

In the event that it is not installed you could also use curl for validation as noted below.

```
kubernetes@DESKTOP-1M2VN7E MINGW64 /c/projects/kind
$ curl -X POST -d "hello" localhost:22181
  % Total    % Received % Xferd  Average Speed   Time    Time
Time  Current
                                 Dload  Upload   Total   Spent
Left  Speed
100     5    0     0  100     5      0    384 --:--:-- --:--:--
--:--:--   416
curl: (52) Empty reply from server
```

The “Empty reply” shows that Zookeeper is running on that port

```
kubernetes@DESKTOP-1M2VN7E MINGW64 /c/projects/kind
$ curl -X POST localhost:29092
  % Total    % Received % Xferd  Average Speed   Time    Time
Time  Current
                                 Dload  Upload   Total   Spent
Left  Speed
  0     0    0     0    0     0      0      0 --:--:--  0:00:03
--:--:--     0
```

```
curl: (7) Failed to connect to localhost port 29092: Connection
refused
```

The “Failed to connect” shows that Kafka is not running on the expected port and we’ll need to do some troubleshooting of our environment.

Troubleshooting failure

Validate the container creation logs

$ docker-compose logs zookeeper | grep -i started

```
zookeeper_1  | [2021-07-26 13:28:56,308] INFO Started
o.e.j.s.ServletContextHandler@304bb45b{/,null,AVAILABLE}
(org.eclipse.jetty.server.handler.ContextHandler)
zookeeper_1  | [2021-07-26 13:28:56,324] INFO Started
ServerConnector@2a265ea9{HTTP/1.1, (http/1.1)}{0.0.0.0:8080}
(org.eclipse.jetty.server.AbstractConnector)
zookeeper_1  | [2021-07-26 13:28:56,324] INFO Started @746ms
(org.eclipse.jetty.server.Server)
zookeeper_1  | [2021-07-26 13:28:56,324] INFO Started AdminServer on address
0.0.0.0, port 8080 and command URL /commands
(org.apache.zookeeper.server.admin.JettyAdminServer)
zookeeper_1  | [2021-07-26 13:28:56,360] INFO PrepRequestProcessor (sid:0)
started, reconfigEnabled=false
(org.apache.zookeeper.server.PrepRequestProcessor)
```

$ docker-compose logs kafka | grep -i started

Notice that we’re not seeing kafka starting in our logs. We’ll need to do a bit more validation.

```
$ docker-compose logs
```

Browse through the logs, searching for ERROR related to Kafka startup. Notice that in our Docker Compose YAML file we set the configuration to 1f but that parameter requires a Short in the current implementation. Older version allowed for the float value, but no longer.

```
kafka_1      | [2021-07-26 13:14:10,250] ERROR Exiting Kafka due
to fatal exception (kafka.Kafka$)
kafka_1      | org.apache.kafka.common.config.ConfigException:
Invalid value 1f for configuration
offsets.topic.replication.factor: Not a number of type SHORT
```

Update Configuration, Flush and Reload our Containers

Update to remove the "**f**" from the **KAFKA_OFFSETS_TOPIC_REPLICATION_FACTOR** value. Edit the docker-compose.yml file with nano. Find that value.

$ nano docker-compose.yml

Save the file.

Stop the containers we started initially.

$ **docker-compose stop**

Restart the containers for Zookeeper and Kafka

KubernetesNetworking/labs/kafka $ **docker-compose up -d**

Validate our services

```
kubernetes@DESKTOP-1M2VN7E MINGW64 /c/projects/kind
$ curl -X POST -d "hello" localhost:22181

  % Total    % Received % Xferd  Average Speed   Time    Time
Time  Current
                                 Dload  Upload   Total   Spent
Left  Speed
100     5    0     0  100     5      0    384 --:--:-- --:--:--
--:--:--   416
curl: (52) Empty reply from server

kubernetes@DESKTOP-1M2VN7E MINGW64 /c/projects/kind
$ curl -X POST localhost:29092

  % Total    % Received % Xferd  Average Speed   Time    Time
Time  Current
                                 Dload  Upload   Total   Spent
Left  Speed
  0     0    0     0    0     0      0      0 --:--:--  0:00:03
--:--:--     0
curl: (52) Empty reply from server
```

Verify Container startup in the logs

$ docker-compose logs zookeeper | grep -i started

```
zookeeper_1  | [2021-07-26 13:28:56,308] INFO Started
o.e.j.s.ServletContextHandler@304bb45b{/,null,AVAILABLE}
(org.eclipse.jetty.server.handler.ContextHandler)
```

```
zookeeper_1  | [2021-07-26 13:28:56,324] INFO Started
ServerConnector@2a265ea9{HTTP/1.1, (http/1.1)}{0.0.0.0:8080}
(org.eclipse.jetty.server.AbstractConnector)
zookeeper_1  | [2021-07-26 13:28:56,324] INFO Started @746ms
(org.eclipse.jetty.server.Server)
zookeeper_1  | [2021-07-26 13:28:56,324] INFO Started AdminServer on address
0.0.0.0, port 8080 and command URL /commands
(org.apache.zookeeper.server.admin.JettyAdminServer)
zookeeper_1  | [2021-07-26 13:28:56,360] INFO PrepRequestProcessor (sid:0)
started, reconfigEnabled=false
(org.apache.zookeeper.server.PrepRequestProcessor)
```

$ docker-compose logs kafka | grep -i started

```
kafka_1  | [2021-07-26 13:29:03,020] DEBUG [ReplicaStateMachine
controllerId=1] Started replica state machine with initial state -> HashMap()
(kafka.controller.ZkReplicaStateMachine)
kafka_1  | [2021-07-26 13:29:03,026] DEBUG [PartitionStateMachine
controllerId=1] Started partition state machine with initial state ->
HashMap() (kafka.controller.ZkPartitionStateMachine)
kafka_1  | [2021-07-26 13:29:03,072] INFO [SocketServer
listenerType=ZK_BROKER, nodeId=1] Started data-plane acceptor and
processor(s) for endpoint : ListenerName(PLAINTEXT)
(kafka.network.SocketServer)
kafka_1  | [2021-07-26 13:29:03,078] INFO [SocketServer
listenerType=ZK_BROKER, nodeId=1] Started data-plane acceptor and
processor(s) for endpoint : ListenerName(PLAINTEXT_HOST)
(kafka.network.SocketServer)
kafka_1  | [2021-07-26 13:29:03,079] INFO [SocketServer
listenerType=ZK_BROKER, nodeId=1] Started socket server acceptors and
processors (kafka.network.SocketServer)
kafka_1  | [2021-07-26 13:29:03,090] INFO [KafkaServer id=1] started
(kafka.server.KafkaServer)
```

Connect tcpdump to our container

In this case we're interested in the containers we just started for zookeeper and Kafka

View container processes

$ docker ps

```
CONTAINER ID   IMAGE                  COMMAND                  CREATED
STATUS              PORTS
NAMES
f3402b942291   ca0dbcd0244c           "/etc/confluent/dockrÇª"   37 minutes
ago      Up 37 minutes      9092/tcp, 0.0.0.0:29092->29092/tcp, :::29092-
>29092/tcp           kafka_kafka_1
bcde193d9756   04999d93068f           "/etc/confluent/dockrÇª"   37 minutes
ago      Up 37 minutes      2888/tcp, 3888/tcp, 0.0.0.0:22181->2181/tcp,
:::22181->2181/tcp   kafka_zookeeper_1
9ef6e9140604   kindest/node:v1.21.1   "/usr/local/bin/entrrÇª"   About an
hour ago   Up About an hour   127.0.0.1:61038->6443/tcp
kind-control-plane
```

Connect tcpdump to our Apache Kafka container network

Use the names value from the "**docker ps**"

$ docker run --tty --net=container:kafka_kafka_1 tcpdump

```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
14:12:47.094277 IP f3402b942291.34598 > kafka_zookeeper_1.kafka_default.2181: Flags
[P.], seq 204:216, ack 341, win 501, options [nop,nop,TS val 3493571334 ecr
4283699278], length 12
14:12:47.094873 IP kafka_zookeeper_1.kafka_default.2181 > f3402b942291.34598: Flags
[P.], seq 341:361, ack 216, win 506, options [nop,nop,TS val 4283705277 ecr
3493571334], length 20
14:12:47.094895 IP f3402b942291.34598 > kafka_zookeeper_1.kafka_default.2181: Flags
[.], ack 361, win 501, options [nop,nop,TS val 3493571335 ecr 4283705277], length 0
14:12:47.094260 IP f3402b942291.34598 > kafka_zookeeper_1.kafka_default.2181: Flags
[P.], seq 718501331:718501343, ack 4148012586, win 501, options [nop,nop,TS val
3493571334 ecr 4283699278], length 12
14:12:47.094871 IP kafka_zookeeper_1.kafka_default.2181 > f3402b942291.34598: Flags
[P.], seq 1:21, ack 12, win 506, options [nop,nop,TS val 4283705277 ecr 3493571334],
length 20
14:12:47.094895 IP f3402b942291.34598 > kafka_zookeeper_1.kafka_default.2181: Flags
[.], ack 21, win 501, options [nop,nop,TS val 3493571335 ecr 4283705277], length 0
```

We see that Kafka is communicating with our Zookeeper instance.

Kill that by Ctrl-C, if for some reason it does not response, use Ctrl-Z to put the process in the background and then kill it.

$ kill -9 %

```
0 [sig] sh 2368! sigpacket::process: Suppressing signal 18 to win32 process (pid 0)
[1]+  Stopped                 docker run --tty --net=container:kafka_kafka_1 tcpdump
```

You'll notice that doesn't actually kill the output for long, rather in this case we have to stop the container. Of course killing a container just causes another one to be restarted by the similar mechanism used in both Docker and Kubernetes to maintain our desired container state.

Retrieve the name of our container running tcpdump.

```
$ docker ps

CONTAINER ID   IMAGE                  COMMAND                  CREATED          STATUS
PORTS                                                  NAMES
caa3f287954c   tcpdump                "/bin/sh -c 'tcpdumpГÇª"   6 minutes ago    Up 6
minutes
friendly_lumiere
```

The docker kill subcommand kills one or more containers. The main process inside the container is sent SIGKILL signal (default), or the signal that is specified with the --signal option. You can kill a container using the container's ID, ID-prefix, or name.

$ docker kill friendly_lumiere
friendly_lumiere

Validate that our tcpdump container is toasted.

$ docker ps

Connect tcpdump to our Apache Zookeeper container network

Use the names value from the "**docker ps**"

$ docker run --tty --net=container:kafka_kafka_1 tcpdump

```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
14:30:47.903168 IP kafka_kafka_1.kafka_default.34598 > bcde193d9756.2181:
Flags [P.], seq 718503491:718503503, ack 4148016186, win 501, options
[nop,nop,TS val 3494652143 ecr 4284780080], length 12
14:30:47.904268 IP bcde193d9756.2181 > kafka_kafka_1.kafka_default.34598:
Flags [P.], seq 1:21, ack 12, win 506, options [nop,nop,TS val 4284786087 ecr
3494652143], length 20
14:30:47.904302 IP kafka_kafka_1.kafka_default.34598 > bcde193d9756.2181: Flags [.],
ack 21, win 501, options [nop,nop,TS val 3494652144 ecr 4284786087], length 0
```

We see the communication that our Zookeeper instance is having with our Apache Kafka container.

Use Ctrl-Z to put the process in the background and then kill it with the "docker kill" as we did previously. Remember that if we kill the tcpdump container process that results in another container being hydrated.

$ docker ps

```
CONTAINER ID   IMAGE                  COMMAND                  CREATED
STATUS             PORTS
NAMES
0e1118f8a03f   tcpdump                "/bin/sh -c 'tcpdumpΓÇª"   2 minutes ago
Up 2 minutes
reverent_euclid
f3402b942291   ca0dbcd0244c           "/etc/confluent/dockΓÇª"   About an hour ago
Up About an hour   9092/tcp, 0.0.0.0:29092->29092/tcp, :::29092->29092/tcp
kafka_kafka_1
bcde193d9756   04999d93068f           "/etc/confluent/dockΓÇª"   About an hour ago
Up About an hour   2888/tcp, 3888/tcp, 0.0.0.0:22181->2181/tcp, :::22181->2181/tcp
kafka_zookeeper_1
9ef6e9140604   kindest/node:v1.21.1   "/usr/local/bin/entrΓÇª"   2 hours ago
Up 2 hours         127.0.0.1:61038->6443/tcp
kind-control-plane
```

Use the NAMES value for our tcpdump image to kill the eavesdropping container attached to the Zookeeper container network

$ docker kill reverent_euclid

Experiment Cleanup

Remove the Zookeeper and Kafka containers with Docker Compose

$ docker-compose stop

```
Running 2/2
 - Container kafka_kafka_1      Stopped
4.6s
 - Container kafka_zookeeper_1  Stopped
```