

Lemay AI Assessment

Farid Faraji

01/10/2023

Building the images:

Using **docker-compose build** to build the images of the **producer_app** and **consumer_app** services

```
devops on  main  on  v20.10.23 using  default/iron-burner-389219 via  base took 34m 19.3s  
→ docker-compose build
```

```
[+] Building 0.0s (0/0)
[+] Building 0.1s (2/3)
=> [internal] load build definition from Dockerfile-producer 0.0s
=> => transferring dockerfile: 41B 0.0s
=> [internal] load .dockerignore 0.0s
=> => transferring context: 2B 0.0s
[+] Building 1.2s (9/9) FINISHED
=> [internal] load build definition from Dockerfile-producer 0.0s
=> => transferring dockerfile: 41B 0.0s
=> [internal] load .dockerignore 0.0s
=> => transferring context: 2B 0.0s
=> [internal] load metadata for docker.io/library/python:3.8-slim 1.0s
=> [1/4] FROM docker.io/library/python:3.8-slim@sha256:9187d27fd8f222a181292f24f8e7d6b22419d46bd9cc4506adbdf2dcfae68a56 0.0s
=> => resolve docker.io/library/python:3.8-slim@sha256:9187d27fd8f222a181292f24f8e7d6b22419d46bd9cc4506adbdf2dcfae68a56 0.0s
=> [internal] load build context 0.0s
=> => transferring context: 82B 0.0s
=> CACHED [2/4] RUN pip install kafka-python elasticsearch neo4j confluent-kafka 0.0s
[+] Building 1.2s (9/9) FINISHED
=> [internal] load build definition from Dockerfile-consumer 0.0s
=> => transferring dockerfile: 41B 0.0s
=> [internal] load .dockerignore 0.0s
=> => transferring context: 2B 0.0s
=> [internal] load metadata for docker.io/library/python:3.8-slim 1.0s
=> [internal] load build context 0.0s
=> => transferring context: 82B 0.0s
=> [1/4] FROM docker.io/library/python:3.8-slim@sha256:9187d27fd8f222a181292f24f8e7d6b22419d46bd9cc4506adbdf2dcfae68a56 0.0s
=> => resolve docker.io/library/python:3.8-slim@sha256:9187d27fd8f222a181292f24f8e7d6b22419d46bd9cc4506adbdf2dcfae68a56 0.0s
=> CACHED [2/4] RUN pip install kafka-python elasticsearch==7.10.0 neo4j 0.0s
=> CACHED [3/4] COPY ./consumer_app /app 0.0s
=> CACHED [4/4] WORKDIR /app 0.0s
=> exporting to image 0.0s
=> => exporting layers 0.0s
=> => writing image sha256:21a4a526ff3e8f2d738954fd1ef801e6ab4003ea2d88c4065e5509910e3d7cb0 0.0s
=> => naming to docker.io/library/devops-consumer_app 0.0s
```

Running the containers:

Using the **docker-compose up** to run the docker containers.

```
devops on  main [!] on  v20.10.23 using  default/iron-burner-389219 via  base
→ docker-compose up -d

[+] Running 6/6
  :: Container devops-neo4j-1      Started
  :: Container zookeeper          Started
  :: Container elasticsearch      Started
  :: Container kafka              Started
  :: Container devops-consumer_app-1 Started
  :: Container devops-producer_app-1 Started
```

Here are the running containers

```
devops on  main [!] on  v20.10.23 runs  DDDEKZ using  default/iron-burner-389219 via  base
+ docker container ls
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                               NAMES
281425b0bb60   devops-producer_app                "python producer_app.." 35 minutes ago Up 20 seconds                               devops-pro
ducer_app-1
da0e94ff01d4   devops-consumer_app                "python consumer_app.." 35 minutes ago Up 20 seconds                               devops-con
sumer_app-1
0f0280b889ce   neo4j:4.2.3                        "/sbin/tini -g -- /d.." 35 minutes ago Up 20 seconds   0.0.0.0:7474->7474/tcp, 7473/tcp, 0.0.0.0:7687->7687/tcp devops-neo
4j-1
9e88fa7af03b   docker.elastic.co/elasticsearch:7.10.2 "/sbin/tini -- /usr/local.." 39 minutes ago Up 20 seconds   0.0.0.0:9200->9200/tcp, 9300/tcp               elasticsea
rch
8b1d18e605c5   confluentinc/cp-kafka:latest       "/etc/confluent/dock.." 40 minutes ago Up 20 seconds   9092/tcp, 0.0.0.0:9093->9093/tcp               kafka
3fb07511957d   confluentinc/cp-zookeeper:latest   "/etc/confluent/dock.." 41 minutes ago Up 20 seconds   2888/tcp, 0.0.0.0:2181->2181/tcp, 3888/tcp     zookeeper
```

<input type="checkbox"/>	Name	Image	Status	Port(s)	Last started	Actions
<input type="checkbox"/>	devops	-	Running (6/6)		2 minutes ago	
<input type="checkbox"/>	zookeeper 3fb07511957d	confluentinc/cp-zookeeper:latest	Running	2181:2181	2 minutes ago	
<input type="checkbox"/>	kafka 8b1d18e605c5	confluentinc/cp-kafka:latest	Running	9093:9093	2 minutes ago	
<input type="checkbox"/>	elasticsearch 9e88fa7af03b	docker.elastic.co/elasticsearch/elasticsearch:7.10.2	Running	9200:9200	2 minutes ago	
<input type="checkbox"/>	neo4j-1 0f0280b889ce	neo4j:4.2.3	Running	7474:7474 Show all ports (2)	2 minutes ago	
<input type="checkbox"/>	consumer_app-1 da0e94ff01d4	devops-consumer_app	Running		2 minutes ago	
<input type="checkbox"/>	producer_app-1 281425b0bb60	devops-producer_app	Running		2 minutes ago	

The producer service creates the message sending it through Kafka to be consumed by the consumer service.

Here are the logs from the producer service.

```
2023-11-01 14:24:01 INFO:root:Sent message: b'Message 9024'
2023-11-01 14:24:02 INFO:root:Sent message: b'Message 9025'
2023-11-01 14:24:03 INFO:root:Sent message: b'Message 9026'
2023-11-01 14:24:04 INFO:root:Sent message: b'Message 9027'
2023-11-01 14:24:05 INFO:root:Sent message: b'Message 9028'
2023-11-01 14:24:06 INFO:root:Sent message: b'Message 9029'
2023-11-01 14:24:07 INFO:root:Sent message: b'Message 9030'
2023-11-01 14:24:08 INFO:root:Sent message: b'Message 9031'
```

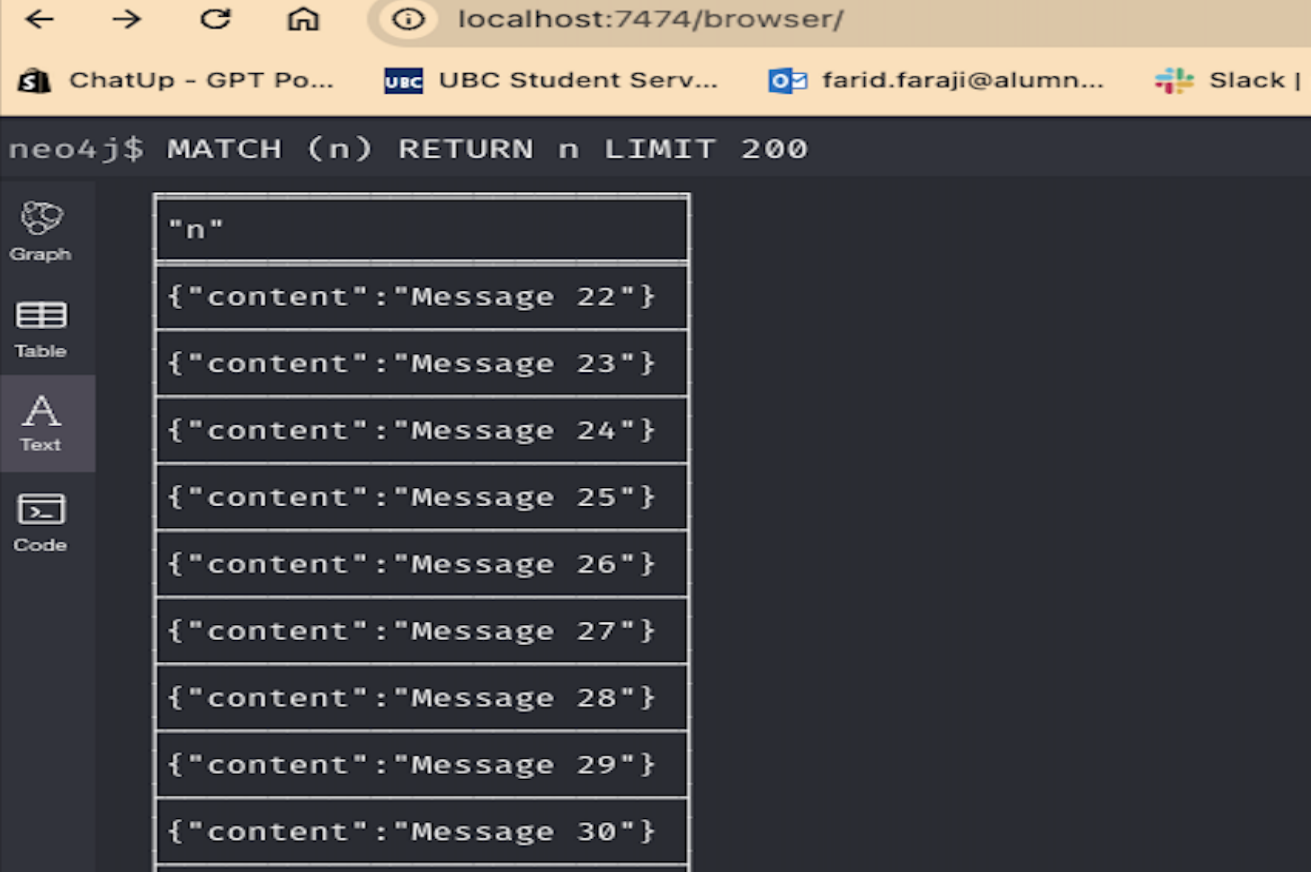
And here are the logs from the consumer service.

```
2023-11-01 14:23:12 INFO:elasticsearch:POST http://elasticsearch:9200/messages-test/_doc [status:201 request:0.008s]
2023-11-01 14:23:13 INFO:root:Received message: Message 8976
2023-11-01 14:23:13 INFO:elasticsearch:POST http://elasticsearch:9200/messages-test/_doc [status:201 request:0.007s]
2023-11-01 14:23:14 INFO:root:Received message: Message 8977
2023-11-01 14:23:14 INFO:elasticsearch:POST http://elasticsearch:9200/messages-test/_doc [status:201 request:0.008s]
2023-11-01 14:23:15 INFO:root:Received message: Message 8978
2023-11-01 14:23:15 INFO:elasticsearch:POST http://elasticsearch:9200/messages-test/_doc [status:201 request:0.006s]
2023-11-01 14:23:16 INFO:root:Received message: Message 8979
2023-11-01 14:23:16 INFO:elasticsearch:POST http://elasticsearch:9200/messages-test/_doc [status:201 request:0.007s]
```

ElasticSearch and Neo4j

The consumer service then inserts the messages into the ElasticSearch and the Neo4j services.

Here are examples of the data being populated in Neo4j

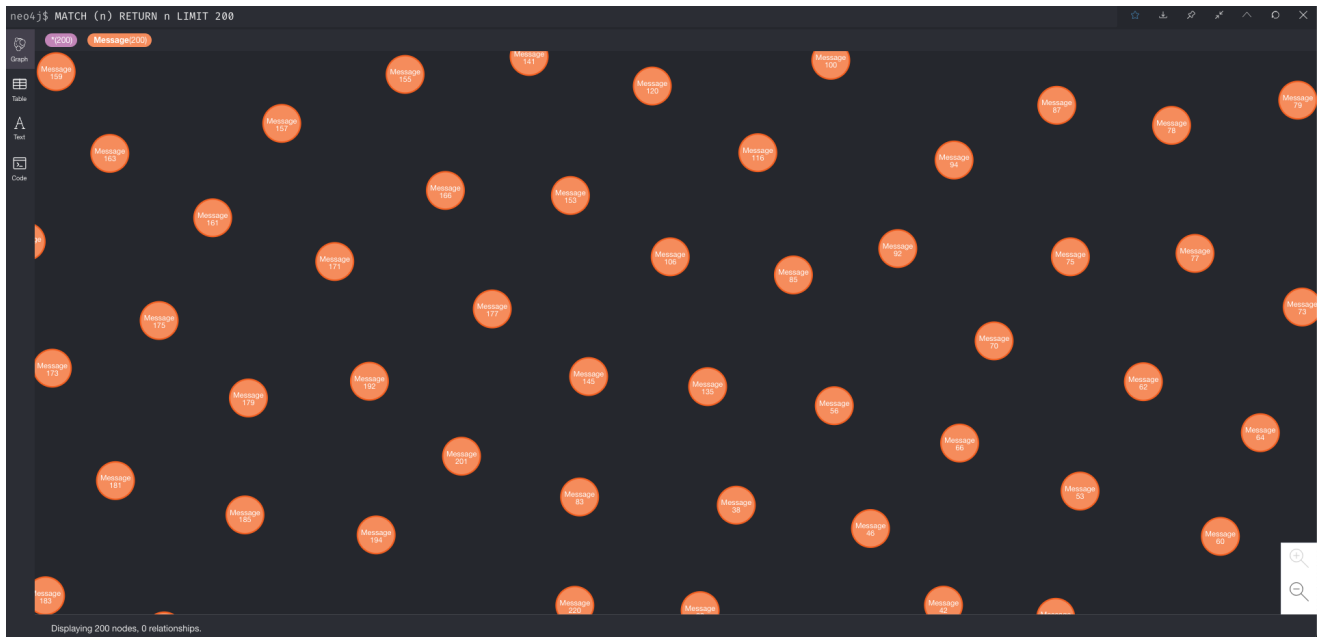


The screenshot shows the Neo4j Browser interface in a web browser. The address bar shows `localhost:7474/browser/`. The browser has several tabs open: "ChatUp - GPT Po...", "UBC UBC Student Serv...", "farid.faraji@alumn...", and "Slack |".

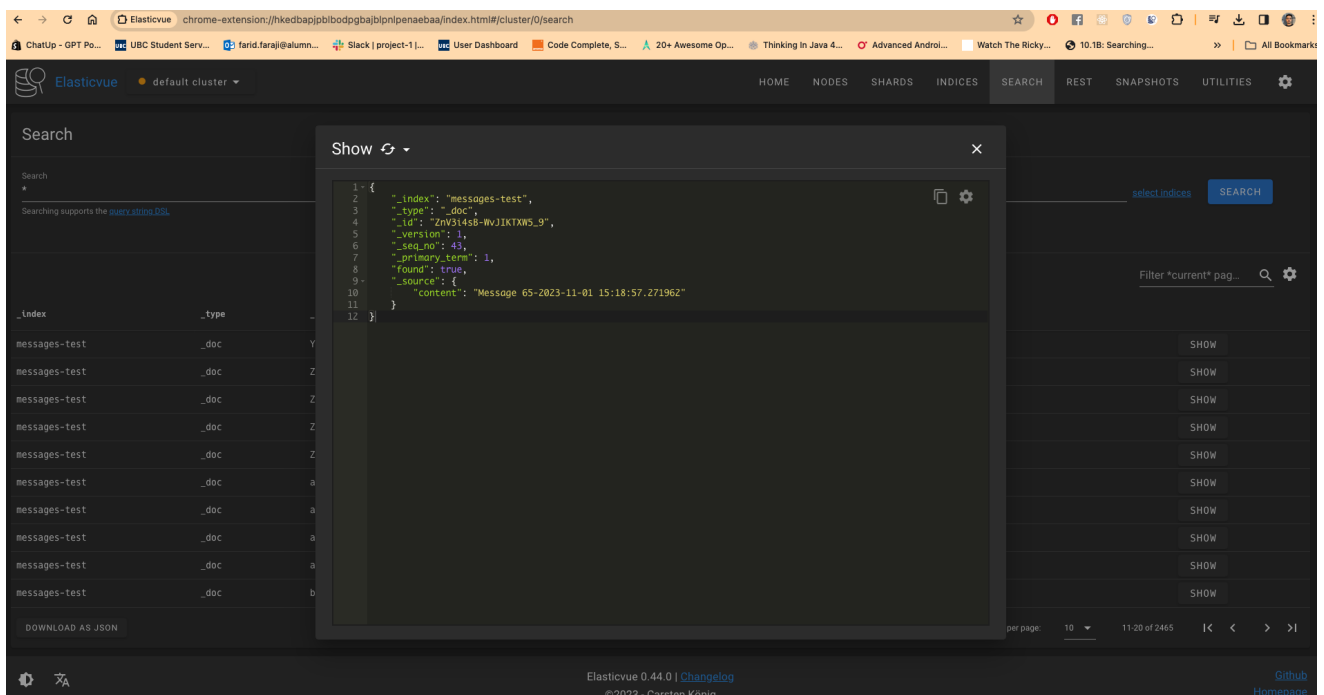
The main interface displays a Cypher query in the top bar: `neo4j$ MATCH (n) RETURN n LIMIT 200`. On the left sidebar, there are four view options: "Graph", "Table", "Text", and "Code". The "Table" view is selected.

The results are displayed in a table with the following data:

"n"
{"content": "Message 22"}
{"content": "Message 23"}
{"content": "Message 24"}
{"content": "Message 25"}
{"content": "Message 26"}
{"content": "Message 27"}
{"content": "Message 28"}
{"content": "Message 29"}
{"content": "Message 30"}



Below shows an example of content in the elasticsearch index being populated by the consumer service.

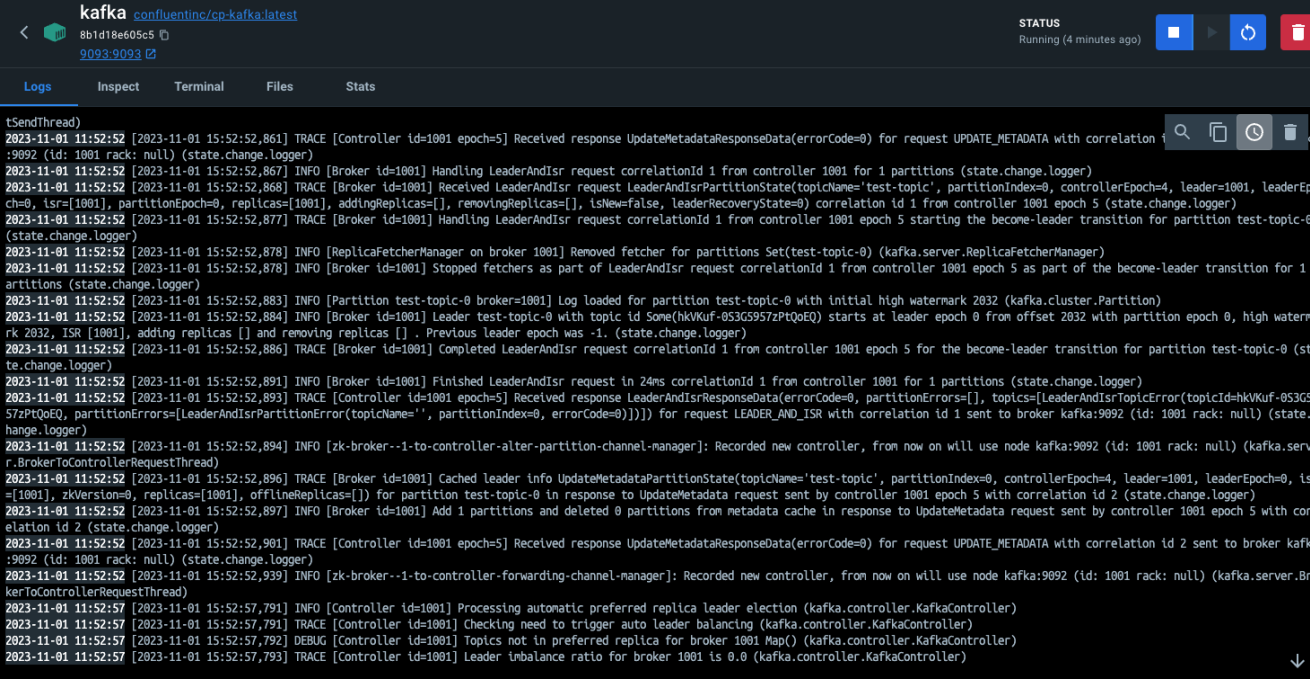


Logs Usage:

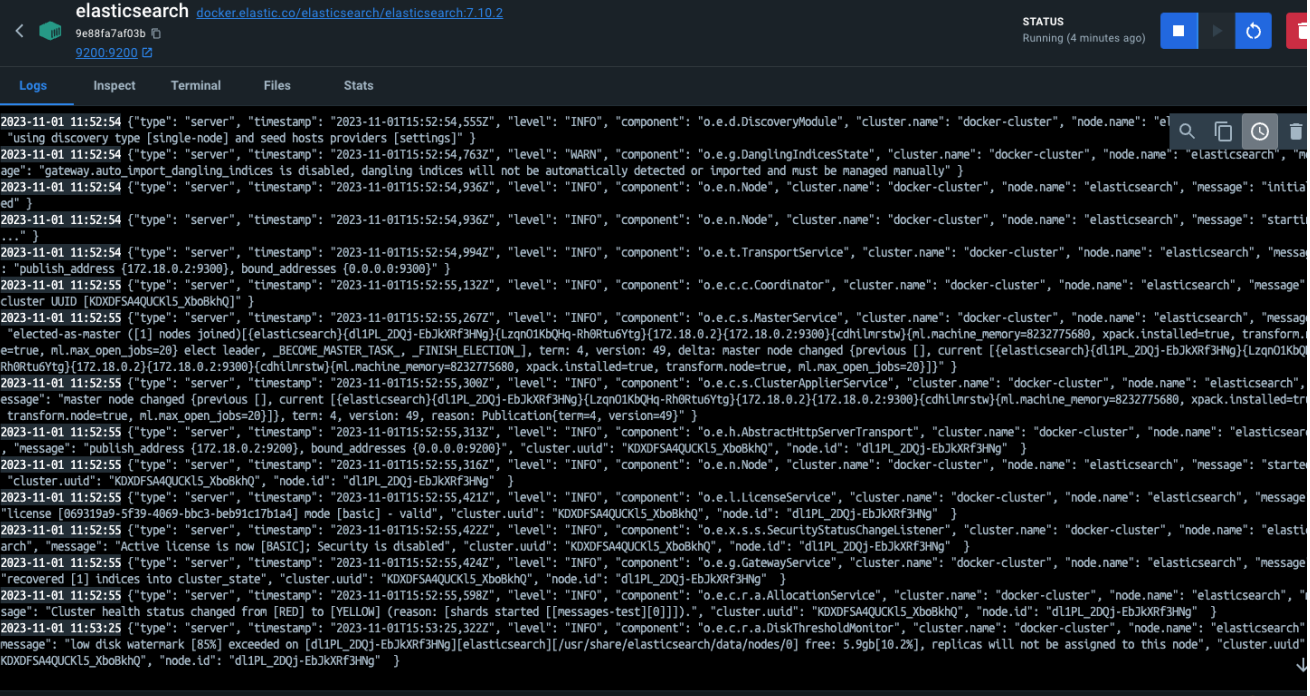
The log files from each container play a critical role in troubleshooting issues within the pipeline. Consider an instance where unexpected data is detected in the Neo4j graph.

In such a case, it is standard procedure to examine the consumer service's log files to confirm the integrity of the messages being recorded. If the messages are not as anticipated, the investigation would proceed to review the producer's log files to determine the content being generated.

Should an error or complication arise within Kafka or any ancillary service, the logs provide essential diagnostics. By analyzing these logs, one can identify the issue and take appropriate measures to address it.



```
tSendThread)
2023-11-01 11:52:52 [2023-11-01 15:52:52,861] TRACE [Controller id=1001 epoch=5] Received response UpdateMetadataResponseData(errorCode=0) for request UPDATE_METADATA with correlation id 1 from controller 1001 epoch 5 (state.change.logger)
2023-11-01 11:52:52 [2023-11-01 15:52:52,867] INFO [Broker id=1001] Handling LeaderAndIsr request correlationId 1 from controller 1001 for 1 partitions (state.change.logger)
2023-11-01 11:52:52 [2023-11-01 15:52:52,868] TRACE [Broker id=1001] Received LeaderAndIsr request LeaderAndIsrPartitionState(topicName='test-topic', partitionIndex=0, controllerEpoch=4, leader=1001, leaderEpoch=0, isr=[1001], partitionEpoch=0, replicas=[1001], addingReplicas=[], removingReplicas=[], isNew=false, leaderRecoveryState=0) correlation id 1 from controller 1001 epoch 5 (state.change.logger)
2023-11-01 11:52:52 [2023-11-01 15:52:52,877] TRACE [Broker id=1001] Handling LeaderAndIsr request correlationId 1 from controller 1001 epoch 5 starting the become-leader transition for partition test-topic-0 (state.change.logger)
2023-11-01 11:52:52 [2023-11-01 15:52:52,878] INFO [ReplicaFetcherManager on broker 1001] Removed fetcher for partitions Set(test-topic-0) (kafka.server.ReplicaFetcherManager)
2023-11-01 11:52:52 [2023-11-01 15:52:52,878] INFO [Broker id=1001] Stopped fetchers as part of LeaderAndIsr request correlationId 1 from controller 1001 epoch 5 as part of the become-leader transition for 1 partitions (state.change.logger)
2023-11-01 11:52:52 [2023-11-01 15:52:52,883] INFO [Partition test-topic-0 broker=1001] Log loaded for partition test-topic-0 with initial high watermark 2032 (kafka.cluster.Partition)
2023-11-01 11:52:52 [2023-11-01 15:52:52,884] INFO [Broker id=1001] Leader test-topic-0 with topic id Some(hkVkuF-053G5957zPtQoEQ) starts at leader epoch 0 from offset 2032 with partition epoch 0, high watermark 2032, ISR [1001], adding replicas [] and removing replicas []. Previous leader epoch was -1. (state.change.logger)
2023-11-01 11:52:52 [2023-11-01 15:52:52,886] TRACE [Broker id=1001] Completed LeaderAndIsr request correlationId 1 from controller 1001 epoch 5 for the become-leader transition for partition test-topic-0 (state.change.logger)
2023-11-01 11:52:52 [2023-11-01 15:52:52,891] INFO [Broker id=1001] Finished LeaderAndIsr request in 24ms correlationId 1 from controller 1001 for 1 partitions (state.change.logger)
2023-11-01 11:52:52 [2023-11-01 15:52:52,893] TRACE [Controller id=1001 epoch=5] Received response LeaderAndIsrResponseData(errorCode=0, partitionErrors=[], topics=[LeaderAndIsrTopicError(topicId=hkVkuF-053G5957zPtQoEQ, partitionErrors=[LeaderAndIsrPartitionError(topicName='', partitionIndex=0, errorCode=0)])]) for request LEADER_AND_ISR with correlation id 1 sent to broker kafka:9092 (id: 1001 rack: null) (state.change.logger)
2023-11-01 11:52:52 [2023-11-01 15:52:52,894] INFO [zk-broker-1-to-controller-alter-partition-channel-manager]: Recorded new controller, from now on will use node kafka:9092 (id: 1001 rack: null) (kafka.server.BrokerToControllerRequestThread)
2023-11-01 11:52:52 [2023-11-01 15:52:52,896] TRACE [Broker id=1001] Cached leader info UpdateMetadataPartitionState(topicName='test-topic', partitionIndex=0, controllerEpoch=4, leader=1001, leaderEpoch=0, isr=[1001], zkVersion=0, replicas=[1001], offlineReplicas=[]) for partition test-topic-0 in response to UpdateMetadata request sent by controller 1001 epoch 5 with correlation id 2 (state.change.logger)
2023-11-01 11:52:52 [2023-11-01 15:52:52,897] INFO [Broker id=1001] Add 1 partitions and deleted 0 partitions from metadata cache in response to UpdateMetadata request sent by controller 1001 epoch 5 with correlation id 2 (state.change.logger)
2023-11-01 11:52:52 [2023-11-01 15:52:52,901] TRACE [Controller id=1001 epoch=5] Received response UpdateMetadataResponseData(errorCode=0) for request UPDATE_METADATA with correlation id 2 sent to broker kafka:9092 (id: 1001 rack: null) (state.change.logger)
2023-11-01 11:52:52 [2023-11-01 15:52:52,939] INFO [zk-broker-1-to-controller-forwarding-channel-manager]: Recorded new controller, from now on will use node kafka:9092 (id: 1001 rack: null) (kafka.server.BrokerToControllerRequestThread)
2023-11-01 11:52:57 [2023-11-01 15:52:57,791] INFO [Controller id=1001] Processing automatic preferred replica leader election (kafka.controller.KafkaController)
2023-11-01 11:52:57 [2023-11-01 15:52:57,791] TRACE [Controller id=1001] Checking need to trigger auto leader balancing (kafka.controller.KafkaController)
2023-11-01 11:52:57 [2023-11-01 15:52:57,792] DEBUG [Controller id=1001] Topics not in preferred replica for broker 1001 Map() (kafka.controller.KafkaController)
2023-11-01 11:52:57 [2023-11-01 15:52:57,793] TRACE [Controller id=1001] Leader imbalance ratio for broker 1001 is 0.0 (kafka.controller.KafkaController)
```



```
2023-11-01 11:52:54 {"type": "server", "timestamp": "2023-11-01T15:52:54.555Z", "level": "INFO", "component": "o.e.d.DiscoveryModule", "cluster.name": "docker-cluster", "node.name": "elasticsearch", "message": "Using discovery type [single-node] and seed hosts providers [settings]"}
2023-11-01 11:52:54 {"type": "server", "timestamp": "2023-11-01T15:52:54.763Z", "level": "WARN", "component": "o.e.g.DanglingIndicesState", "cluster.name": "docker-cluster", "node.name": "elasticsearch", "message": "gateway.auto_import_dangling_indices is disabled, dangling indices will not be automatically detected or imported and must be managed manually"}
2023-11-01 11:52:54 {"type": "server", "timestamp": "2023-11-01T15:52:54.936Z", "level": "INFO", "component": "o.e.n.Node", "cluster.name": "docker-cluster", "node.name": "elasticsearch", "message": "Initialised"}
2023-11-01 11:52:54 {"type": "server", "timestamp": "2023-11-01T15:52:54.936Z", "level": "INFO", "component": "o.e.n.Node", "cluster.name": "docker-cluster", "node.name": "elasticsearch", "message": "Starting"}
2023-11-01 11:52:54 {"type": "server", "timestamp": "2023-11-01T15:52:54.994Z", "level": "INFO", "component": "o.e.t.TransportService", "cluster.name": "docker-cluster", "node.name": "elasticsearch", "message": "publish_address [172.18.0.2:9300], bound_addresses [0.0.0.0:9300]"}
2023-11-01 11:52:55 {"type": "server", "timestamp": "2023-11-01T15:52:55.132Z", "level": "INFO", "component": "o.e.c.c.Coordinator", "cluster.name": "docker-cluster", "node.name": "elasticsearch", "message": "Elasticsearch cluster UUID [KXDFSA4QUCKL5_Xbo8khQ]"}
2023-11-01 11:52:55 {"type": "server", "timestamp": "2023-11-01T15:52:55.267Z", "level": "INFO", "component": "o.e.c.s.MasterService", "cluster.name": "docker-cluster", "node.name": "elasticsearch", "message": "elected-as-master ([1] nodes joined){elasticsearch[d1LPL_2DQj-EbJkXRF3HNg]{LzqN01KbQh-Rh8Rtu6Ytg}{172.18.0.2}{172.18.0.2:9300}{cdhlnrstw}[nL.machine.memory=8232775680, xpack.installed=true, transform.node=true, m.max.open_jobs=20] elect leader, BECOME_MASTER_TASK, FINISH_ELECTION, term: 4, version: 49, delta: master node changed (previous [], current [{elasticsearch[d1LPL_2DQj-EbJkXRF3HNg]{LzqN01KbQh-Rh8Rtu6Ytg}{172.18.0.2}{172.18.0.2:9300}{cdhlnrstw}[nL.machine.memory=8232775680, xpack.installed=true, transform.node=true, m.max.open_jobs=20]})"}
2023-11-01 11:52:55 {"type": "server", "timestamp": "2023-11-01T15:52:55.308Z", "level": "INFO", "component": "o.e.c.s.ClusterApplierService", "cluster.name": "docker-cluster", "node.name": "elasticsearch", "message": "master node changed (previous [], current [{elasticsearch[d1LPL_2DQj-EbJkXRF3HNg]{LzqN01KbQh-Rh8Rtu6Ytg}{172.18.0.2}{172.18.0.2:9300}{cdhlnrstw}[nL.machine.memory=8232775680, xpack.installed=true, transform.node=true, m.max.open_jobs=20]})"}
2023-11-01 11:52:55 {"type": "server", "timestamp": "2023-11-01T15:52:55.313Z", "level": "INFO", "component": "o.e.h.AbstractHttpServerTransport", "cluster.name": "docker-cluster", "node.name": "elasticsearch", "message": "publish_address [172.18.0.2:9200], bound_addresses [0.0.0.0:9200]"}
2023-11-01 11:52:55 {"type": "server", "timestamp": "2023-11-01T15:52:55.316Z", "level": "INFO", "component": "o.e.n.Node", "cluster.name": "docker-cluster", "node.name": "elasticsearch", "message": "Starting Elasticsearch"}
2023-11-01 11:52:55 {"type": "server", "timestamp": "2023-11-01T15:52:55.421Z", "level": "INFO", "component": "o.e.l.LicenseService", "cluster.name": "docker-cluster", "node.name": "elasticsearch", "message": "license [069319a9-5f39-4069-bbc3-beb91c17b1a4] mode [basic] - valid"}
2023-11-01 11:52:55 {"type": "server", "timestamp": "2023-11-01T15:52:55.422Z", "level": "INFO", "component": "o.e.x.s.s.SecurityStatusChangeListener", "cluster.name": "docker-cluster", "node.name": "elasticsearch", "message": "Active license is now [BASIC]; Security is disabled"}
2023-11-01 11:52:55 {"type": "server", "timestamp": "2023-11-01T15:52:55.424Z", "level": "INFO", "component": "o.e.g.GatewayService", "cluster.name": "docker-cluster", "node.name": "elasticsearch", "message": "Recovered [1] indices into cluster_state"}
2023-11-01 11:52:55 {"type": "server", "timestamp": "2023-11-01T15:52:55.598Z", "level": "INFO", "component": "o.e.c.r.a.AllocationService", "cluster.name": "docker-cluster", "node.name": "elasticsearch", "message": "Cluster health status changed from [RED] to [YELLOW] (reason: [shards started [[messages-test][0]]])"}
2023-11-01 11:53:25 {"type": "server", "timestamp": "2023-11-01T15:53:25.322Z", "level": "INFO", "component": "o.e.c.r.a.DiskThresholdMonitor", "cluster.name": "docker-cluster", "node.name": "elasticsearch", "message": "Low disk watermark [85%] exceeded on [d1LPL_2DQj-EbJkXRF3HNg][elasticsearch]/usr/share/elasticsearch/data/nodes/0 free: 5.9Gb[10.2%], replicas will not be assigned to this node"}
KXDFSA4QUCKL5_Xbo8khQ, "node.id": "d1LPL_2DQj-EbJkXRF3HNg" }
```

devops-neo4j-1

neo4j-4.2.3

0f0280b889ce

7474:74747687:7687

STATUS

Running (4 minutes ago)

Logs

Inspect

Terminal

Files

Stats

2023-11-01 11:51:54 at org.neo4j.server.web.Jetty9WebServer.stop(Jetty9WebServer.java:170) ~[neo4j-server-4.2.3.jar:4.2.3]

2023-11-01 11:51:54 at org.neo4j.server.AbstractNeoWebServer.stopWebServer(AbstractNeoWebServer.java:372) ~[neo4j-server-4.2.3.jar:4.2.3]

2023-11-01 11:51:54 at org.neo4j.server.AbstractNeoWebServer\$ServerComponentsLifecycleAdapter.stop(AbstractNeoWebServer.java:475) ~[neo4j-server-4.2.3.jar:4.2.3]

2023-11-01 11:51:54 at org.neo4j.kernel.lifecycle.LifeSupport\$LifecycleInstance.stop(LifeSupport.java:477) ~[neo4j-common-4.2.3.jar:4.2.3]

2023-11-01 11:51:54 at org.neo4j.kernel.lifecycle.LifeSupport.stopInstances(LifeSupport.java:260) ~[neo4j-common-4.2.3.jar:4.2.3]

2023-11-01 11:51:54 at org.neo4j.kernel.lifecycle.LifeSupport.stop(LifeSupport.java:146) ~[neo4j-common-4.2.3.jar:4.2.3]

2023-11-01 11:51:54 at org.neo4j.server.AbstractNeoWebServer.stop(AbstractNeoWebServer.java:347) ~[neo4j-server-4.2.3.jar:4.2.3]

2023-11-01 11:51:54 at org.neo4j.kernel.lifecycle.LifeSupport\$LifecycleInstance.stop(LifeSupport.java:477) ~[neo4j-common-4.2.3.jar:4.2.3]

2023-11-01 11:51:54 at org.neo4j.kernel.lifecycle.LifeSupport.stopInstances(LifeSupport.java:260) ~[neo4j-common-4.2.3.jar:4.2.3]

2023-11-01 11:51:54 at org.neo4j.kernel.lifecycle.LifeSupport.stop(LifeSupport.java:146) ~[neo4j-common-4.2.3.jar:4.2.3]

2023-11-01 11:51:54 at org.neo4j.kernel.lifecycle.LifeSupport.shutdown(LifeSupport.java:169) ~[neo4j-common-4.2.3.jar:4.2.3]

2023-11-01 11:51:54 at org.neo4j.dbms.database.DatabaseManagementServiceImpl.shutdown(DatabaseManagementServiceImpl.java:131) ~[neo4j-kernel-4.2.3.jar:4.2.3]

2023-11-01 11:51:54 at org.neo4j.server.NeoBootstrapper.doShutdown(NeoBootstrapper.java:267) ~[neo4j-4.2.3.jar:4.2.3]

2023-11-01 11:51:54 at org.neo4j.server.NeoBootstrapper.lambda\$addShutdownHook\$1(NeoBootstrapper.java:284) ~[neo4j-4.2.3.jar:4.2.3]

2023-11-01 11:51:54 at java.lang.Thread.run(Thread.java:829) [?:?]

2023-11-01 11:52:48 Warning: Some files inside "/data" are not writable from inside container. Changing folder owner to neo4j.

2023-11-01 11:52:57 Changed password for user 'neo4j'.

2023-11-01 11:52:58 Directories in use:

2023-11-01 11:52:58 home: /var/lib/neo4j

2023-11-01 11:52:58 config: /var/lib/neo4j/conf

2023-11-01 11:52:58 logs: /logs

2023-11-01 11:52:58 plugins: /var/lib/neo4j/plugins

2023-11-01 11:52:58 import: /var/lib/neo4j/import

2023-11-01 11:52:58 data: /var/lib/neo4j/data

2023-11-01 11:52:58 certificates: /var/lib/neo4j/certificates

2023-11-01 11:52:58 run: /var/lib/neo4j/run

2023-11-01 11:52:58 Starting Neo4j.

2023-11-01 11:53:03 2023-11-01 15:53:03.723+0000 INFO Starting...

2023-11-01 11:53:11 2023-11-01 15:53:11.304+0000 INFO ===== Neo4j 4.2.3 =====

2023-11-01 11:53:15 2023-11-01 15:53:15.419+0000 INFO Performing postInitialization step for component 'security-users' with version 2 and status CURRENT

2023-11-01 11:53:15 2023-11-01 15:53:15.421+0000 INFO Updating the initial password in component 'security-users'

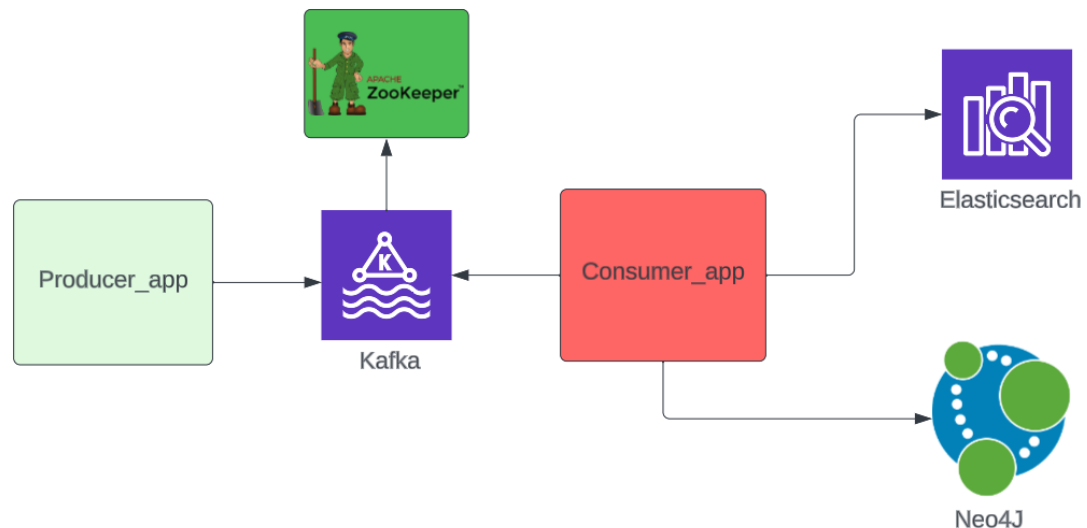
2023-11-01 11:53:22 2023-11-01 15:53:22.522+0000 INFO Bolt enabled on 0.0.0.0:7687.

2023-11-01 11:53:26 2023-11-01 15:53:26.506+0000 INFO Remote interface available at http://localhost:7474/

2023-11-01 11:53:26 2023-11-01 15:53:26.510+0000 INFO Started.

RAM 1.24 GB CPU 1.40% Disk 5.66 GB mem 659.27 GB -X Not connected to Hub

Explaining What the code does:



This project is composed of a suite of six microservices, exemplifying the quintessential consumer-producer paradigm. Within this architecture, one microservice is responsible for generating messages, while another is tasked with processing those messages. The system leverages a robust messaging broker—specifically, Kafka, coupled with ZooKeeper—for efficient message handling, ensuring the system's scalability and its capability to manage substantial message sizes. This configuration enables the system to proficiently process a high volume of sizeable requests.

Upon reception, the consumer microservice archives the incoming messages from Kafka into a Neo4j graph database, which allows for sophisticated relationship-driven data insights. Concurrently, Elasticsearch is employed to facilitate comprehensive full-text search capabilities and to perform complex data analytics.

Using GKE and GCR instead of docker-compose:

In order to run the services in a kubernetes cluster (using GKE in my example)

We will still need to build the docker images and in our case we tag the images with the correct tag to be pushed into GCR then kubernetes will pull the images from the GCR.

We will have to write a kubernetes manifest for each service. You can see screenshots of these manifests shown in the next page

```
# Project configuration
PROJECT_NAME =

ifeq ($(shell uname -p), arm)
DOCKER_PLATFORM = --platform linux/arm64
else
DOCKER_PLATFORM =
endif

default: help

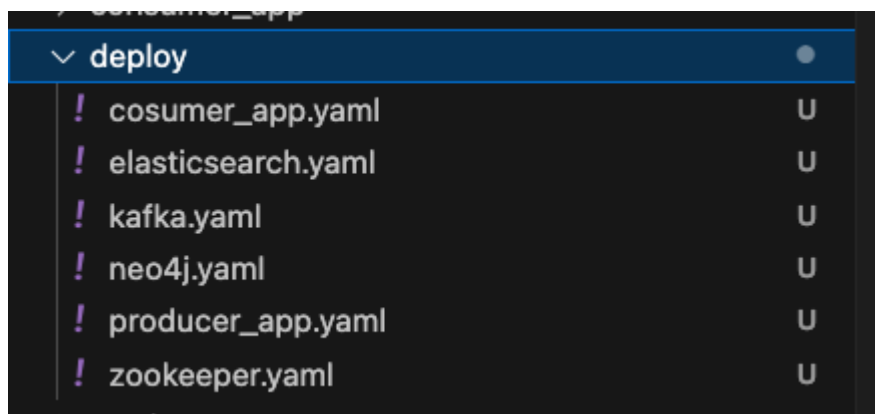
help: # Display help
    @awk -F ':' | grep '^#' \
        '/^[^:]*.+:.*/ { \
            printf "\033[36m%-30s\033[0m %s\n", $$1, $$NF \
        }' $(MAKEFILE_LIST) | sort

build-docker: ## Build the docker images
    docker build $(DOCKER_PLATFORM) -t producer-app -f Dockerfile-producer .
    docker build $(DOCKER_PLATFORM) -t consumer-app -f Dockerfile-consumer .

tag-docker: ## Tag the docker images
    docker tag producer-app gcr.io/lemay-project/producer-app:latest
    docker tag consumer-app gcr.io/lemay-project/consumer-app:latest

push-docker: ## push the images to registry
    docker push gcr.io/lemay-project/producer-app:latest
    docker push gcr.io/lemay-project/consumer-app:latest

k8-cluster: ## deploy service to cluster
    kubectl apply -f deploy/zookeeper.yaml
    kubectl apply -f deploy/kafka.yaml
    kubectl apply -f deploy/elasticsearch.yaml
    kubectl apply -f deploy/Neo4j.yaml
    kubectl apply -f deploy/producer.yaml
    kubectl apply -f deploy/consumer.yaml
```




S.

! zookeeper.yaml U X

! cosumer_app.yaml U

 LICENSE

 Makefile U

deploy > ! zookeeper.yaml

```
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: zookeeper
5  spec:
6    replicas: 1
7    selector:
8      matchLabels:
9        app: zookeeper
10   template:
11     metadata:
12       labels:
13         app: zookeeper
14     spec:
15       containers:
16         - name: zookeeper
17           image: confluentinc/cp-zookeeper:latest
18           ports:
19             - containerPort: 2181
20           env:
21             - name: ZOOKEEPER_CLIENT_PORT
22               value: "2181"
23
24   ---
25   apiVersion: v1
26   kind: Service
27   metadata:
28     name: zookeeper
29   spec:
30     type: ClusterIP
31     ports:
32       - port: 2181
33     selector:
34       app: zookeeper
35
```

← → devops

! cosumer_app.yaml U ! LICENSE M Makefile U ! kafka.yaml U ! elasticsearch.yaml U

```
deploy > ! producer_app.yaml
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: producer-app
5  spec:
6    replicas: 1
7    selector:
8      matchLabels:
9        app: producer-app
10   template:
11     metadata:
12       labels:
13         app: producer-app
14     spec:
15       containers:
16         - name: producer-app
17           image: gcr.io/your-google-cloud-project-id/producer-app:latest
18           # Define any ports, environment variables, etc., as needed.
19
20   ---
21   apiVersion: v1
22   kind: Service
23   metadata:
24     name: producer-app
25   spec:
26     type: ClusterIP
27     # Define any ports as needed.
28     selector:
29       app: producer-app
30
```

```
devops
U ! cosumer_app.yaml U LICENSE Makefile U kafka.yaml U elasticsearch.yaml U
deploy > ! neo4j.yaml
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: neo4j
5  spec:
6    replicas: 1
7    selector:
8      matchLabels:
9        app: neo4j
10   template:
11     metadata:
12       labels:
13         app: neo4j
14     spec:
15       containers:
16         - name: neo4j
17           image: neo4j:4.2.3
18           ports:
19             - containerPort: 7474
20             - containerPort: 7687
21           env:
22             - name: NEO4J_AUTH
23               value: "neo4j/some_password"
24
25   ---
26   apiVersion: v1
27   kind: Service
28   metadata:
29     name: neo4j
30   spec:
31     type: ClusterIP
32     ports:
33       - port: 7474
34       - port: 7687
35     selector:
36       app: neo4j
37
```

```
! cosumer_app.yaml U  LICENSE  Makefile U  ! kafka.yaml U X  ! elasticsearch.yaml U  ! neo4j.j...

deploy > ! kafka.yaml
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: kafka
5  spec:
6    replicas: 1
7    selector:
8      matchLabels:
9        app: kafka
10   template:
11     metadata:
12       labels:
13         app: kafka
14     spec:
15       containers:
16         - name: kafka
17           image: confluentinc/cp-kafka:latest
18           ports:
19             - containerPort: 9093
20           env:
21             - name: KAFKA_LISTENER_SECURITY_PROTOCOL_MAP
22               value: "PLAINTEXT:PLAINTEXT,PLAINTEXT_HOST:PLAINTEXT"
23             - name: KAFKA_ADVERTISED_LISTENERS
24               value: "PLAINTEXT://kafka:9092,PLAINTEXT_HOST://localhost:9093"
25             - name: KAFKA_ZOOKEEPER_CONNECT
26               value: "zookeeper:2181"
27             - name: KAFKA_AUTO_CREATE_TOPICS_ENABLE
28               value: "true"
29
30   ---
31   apiVersion: v1
32   kind: Service
33   metadata:
34     name: kafka
35   spec:
36     type: ClusterIP
37     ports:
38       - port: 9093
39     selector:
40       app: kafka
41
42
```

```
U ! cosumer_app.yaml U LICENSE Makefile U kafka.yaml U ! elastics

deploy > ! elasticsearch.yaml
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: elasticsearch
5  spec:
6    replicas: 1
7    selector:
8      matchLabels:
9        app: elasticsearch
10   template:
11     metadata:
12       labels:
13         app: elasticsearch
14     spec:
15       containers:
16         - name: elasticsearch
17           image: docker.elastic.co/elasticsearch/elasticsearch:7.10.2
18           ports:
19             - containerPort: 9200
20           env:
21             - name: discovery.type
22               value: "single-node"
23
24   ---
25   apiVersion: v1
26   kind: Service
27   metadata:
28     name: elasticsearch
29   spec:
30     type: ClusterIP
31     ports:
32       - port: 9200
33     selector:
34       app: elasticsearch
35
```

! cosumer_app.yaml U X

🔒 LICENSE

M Makefile U

! kafka.yaml U

! elasticsearch.yaml U

deploy > ! cosumer_app.yaml

```
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: consumer-app
5  spec:
6    replicas: 1
7    selector:
8      matchLabels:
9        app: consumer-app
10   template:
11     metadata:
12       labels:
13         app: consumer-app
14     spec:
15       containers:
16       - name: consumer-app
17         image: gcr.io/your-google-cloud-project-id/consumer-app:v1
18         # Define any ports, environment variables, etc., as needed.
19
20   ---
21   apiVersion: v1
22   kind: Service
23   metadata:
24     name: consumer-app
25   spec:
26     type: ClusterIP
27     # Define any ports as needed.
28     selector:
29       app: consumer-app
30
```