1. **Download Elasticsearch**

**Elasticsearch** is a search engine based on Lucene software. It provides a distributed, full-featured search engine with an HTTP web interface that supports JSON data. Elasticsearch is developed in Java and released as open source under the Apache license. Main functions:

-> Full-text search

-> JSON data storage

-> Big data analysis & statistics

* Add GPG key and Elastic source to APT:

curl -fsSL https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo gpg --dearmor -o /usr/share/keyrings/elastic.gpg

echo "deb [signed-by=/usr/share/keyrings/elastic.gpg] https://artifacts.elastic.co/packages/7.x/apt stable main" | sudo tee -a /etc/apt/sources.list.d/elastic-7.x.list

sudo apt update

sudo apt install elasticsearch

* Edit the configuration file for Elasticsearch:

nano /etc/elasticsearch/elasticsearch.yml

=====================

network.host: 0.0.0.0

discovery.type: single-node

=====================

\*with:

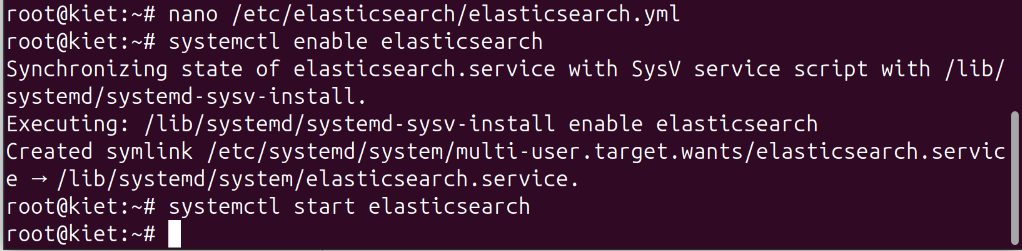
network.host: 0.0.0.0 -> Allow access from any IP.

discovery.type: single-node -> Skip cluster discovery, use simple mode.

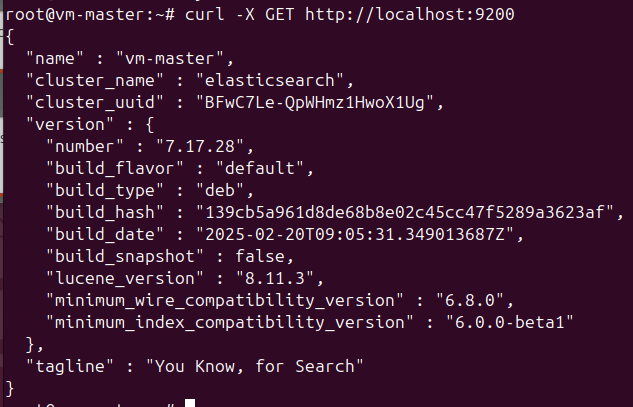
* Start and enable Elasticsearch

systemctl enable elasticsearch

systemctl start elasticsearch



* check: curl -X GET http://localhost:9200



1. **Test on Elasticsearch (CRUD + Cluster Info) with JSON format**

* Create index and add first document (Create):

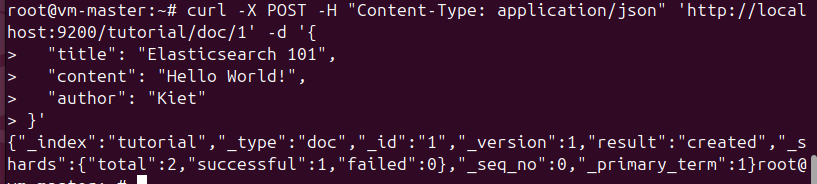
curl -X POST -H "Content-Type: application/json" 'http://localhost:9200/tutorial/doc/1' -d '{

"title": "Elasticsearch 101",

"content": "Hello World!",

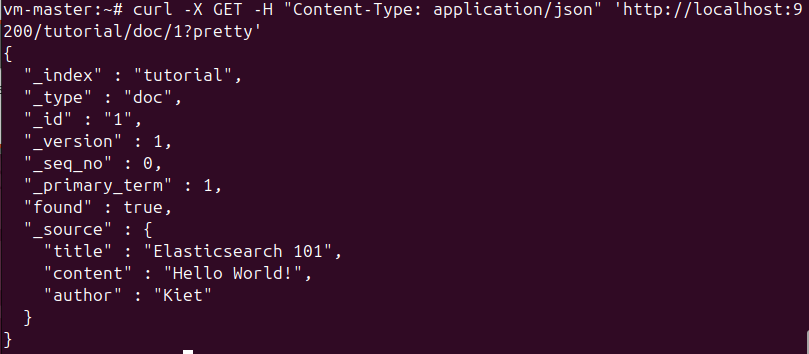
"author": "Kiet"

}'



* View document (Read):

curl -X GET -H "Content-Type: application/json" 'http://localhost:9200/tutorial/doc/1?pretty'



* Update document (Update):

curl -X POST -H "Content-Type: application/json" 'http://localhost:9200/tutorial/doc/1/\_update' -d '{

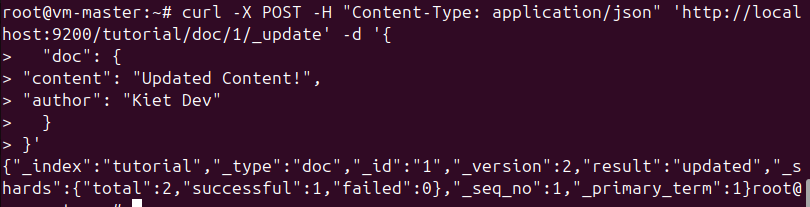
"doc": {

"content": "Updated Content!",

"author": "Kiet Dev"

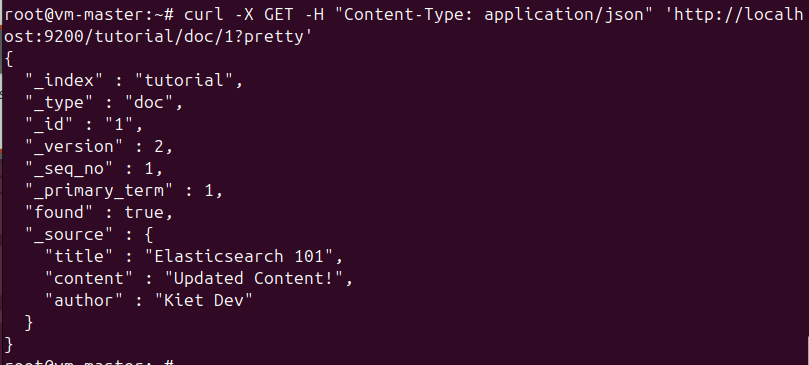
}

}'



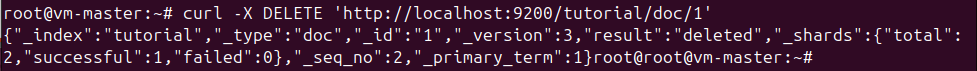
* check:

curl -X GET -H "Content-Type: application/json" 'http://localhost:9200/tutorial/doc/1?pretty'



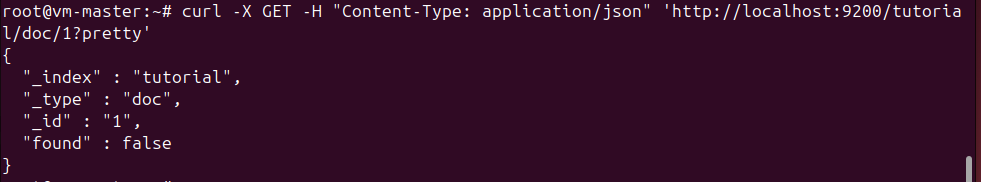
* Delete document (Delete):

curl -X DELETE 'http://localhost:9200/tutorial/doc/1'



* check:

curl -X GET -H "Content-Type: application/json" 'http://localhost:9200/tutorial/doc/1?pretty'



* Create multiple documents using Bulk API:

Create nano bulk\_data.json

content:  
{ "index": { "\_index": "tutorial", "\_type": "doc", "\_id": "2" } }

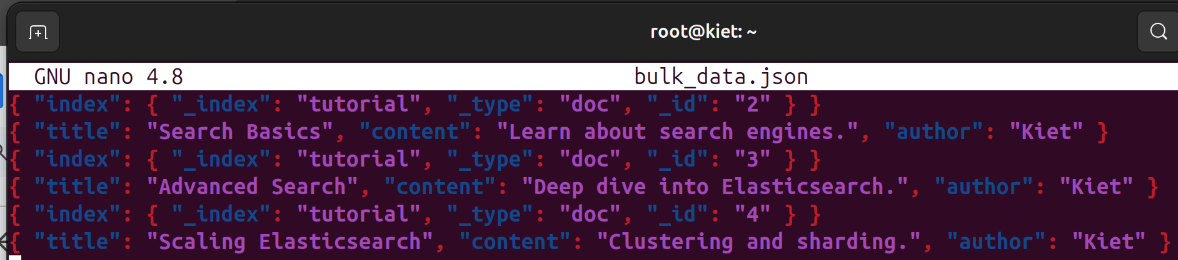
{ "title": "Search Basics", "content": "Learn about search engines.", "author": "Kiet" }

{ "index": { "\_index": "tutorial", "\_type": "doc", "\_id": "3" } }

{ "title": "Advanced Search", "content": "Deep dive into Elasticsearch.", "author": "Kiet" }

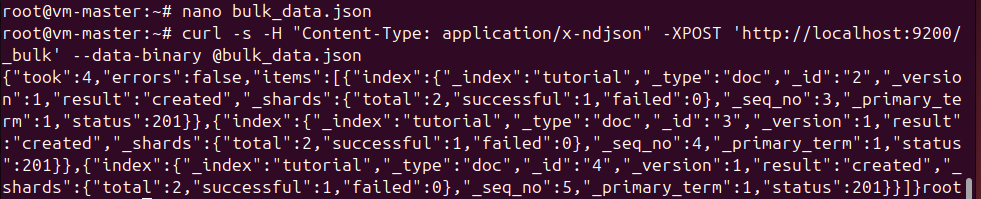
{ "index": { "\_index": "tutorial", "\_type": "doc", "\_id": "4" } }

{ "title": "Scaling Elasticsearch", "content": "Clustering and sharding.", "author": "Kiet" }



* add datas:

curl -s -H "Content-Type: application/x-ndjson" -XPOST 'http://localhost:9200/\_bulk' --data-binary @bulk\_data.json



* Find all document in index:

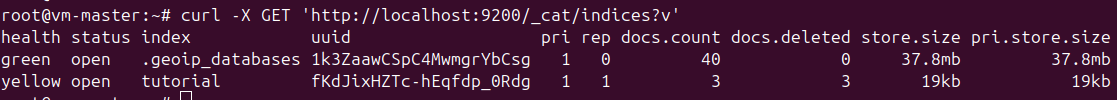
curl -X GET 'http://localhost:9200/tutorial/\_search?pretty'



**Check index, mapping and health**

* List index:

curl -X GET 'http://localhost:9200/\_cat/indices?v'



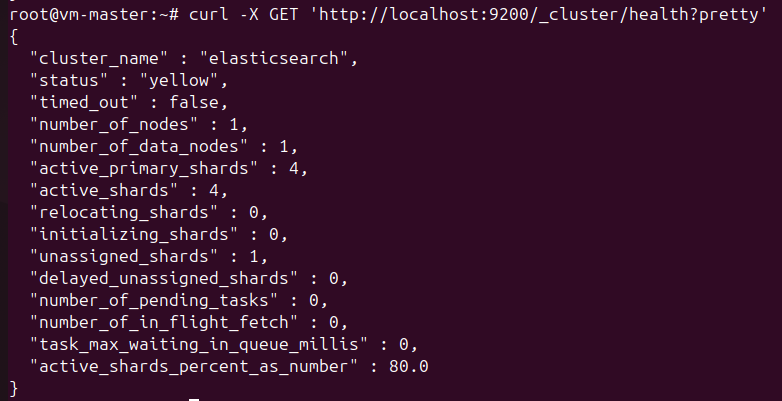
* Mapping of index tutorial:

curl -X GET '<http://localhost:9200/tutorial/_mapping?pretty>'



* Status cluster:

curl -X GET '<http://localhost:9200/_cluster/health?pretty>'



1. **Install and Work with Apache Cassandra**

**Apache Cassandra** is one of the most popular open source database systems. It is designed to handle large amounts of data stored in many distributed servers while providing high scalability and availability without a single failure. Cassandra systems can span multiple data centers, enabling low latency for all connected machines. Main functions:

-> Table-based data storage

-> Distributed across cluster nodes

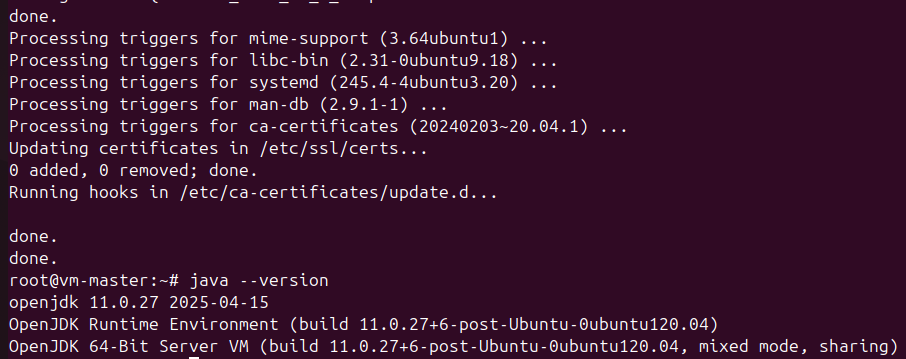
-> Optimized for large data reads/writes

* Install Java 11 (require of Cassandra):

apt install openjdk-11-jdk -y

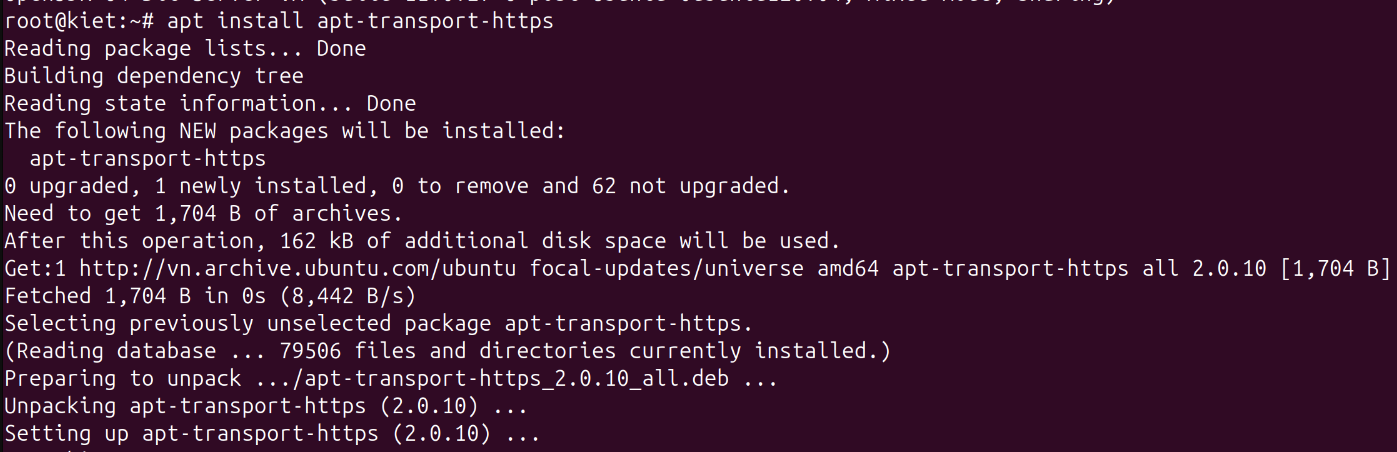
* check version java:

java --version



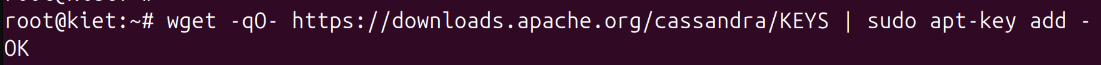
* Install apt-transport-https

apt install apt-transport-https



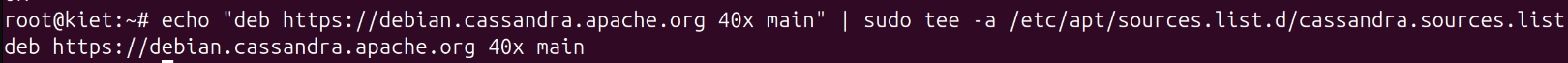
* Add repository Cassandra and import GPG key:
* Import key from Apache:

wget -qO- https://downloads.apache.org/cassandra/KEYS | sudo apt-key add -



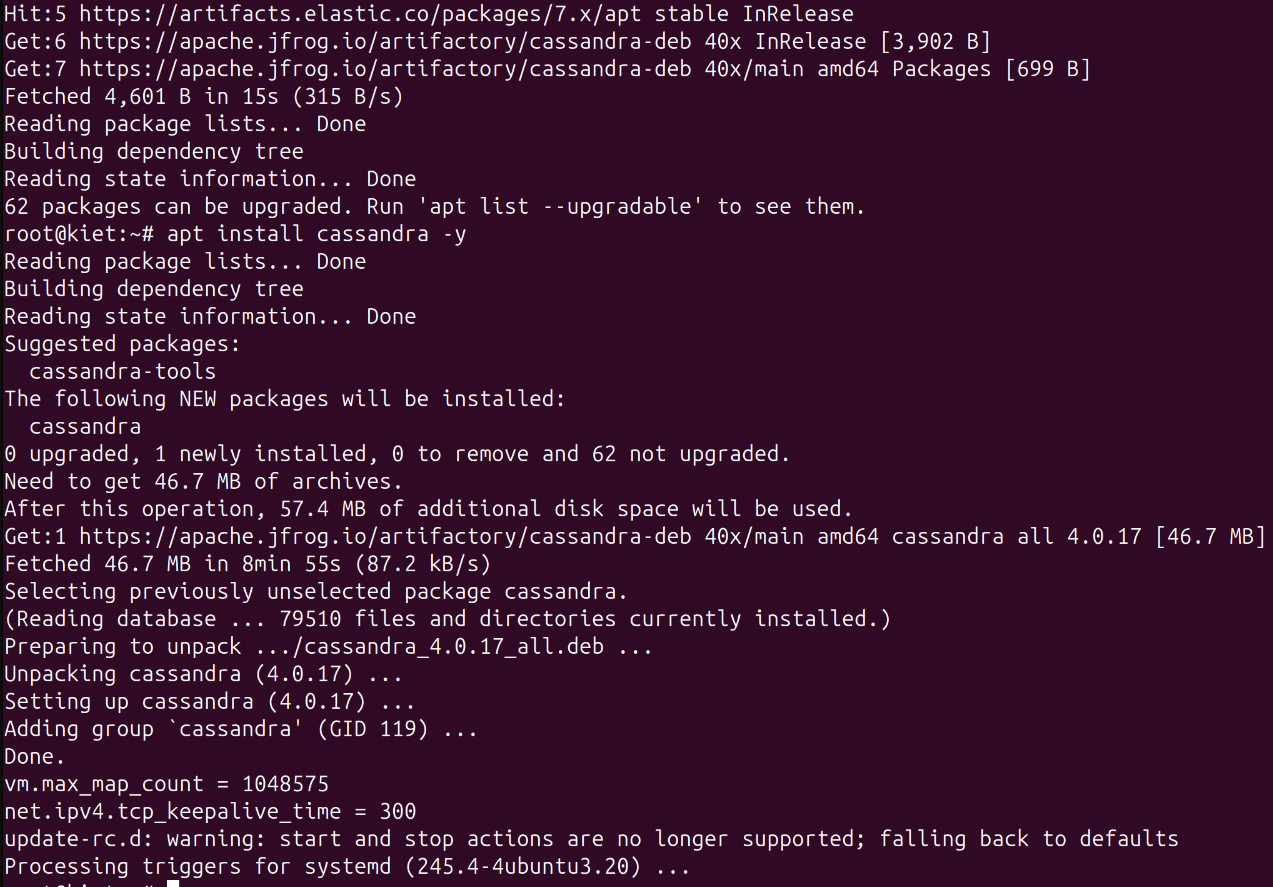
* Add repository Cassandra:

echo "deb https://debian.cassandra.apache.org 40x main" | sudo tee -a /etc/apt/sources.list.d/cassandra.sources.list



* Setup Apache Cassandra:
* reupdate repo: apt update
* Insstall Cassandra:

apt install cassandra -y



* enable and start service:

systemctl enable cassandra

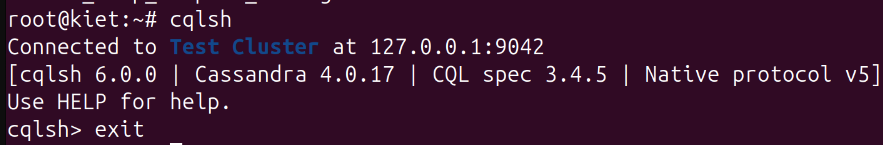
systemctl start cassandra

**Setting Cassandra**

* Backup before editing configuration file:

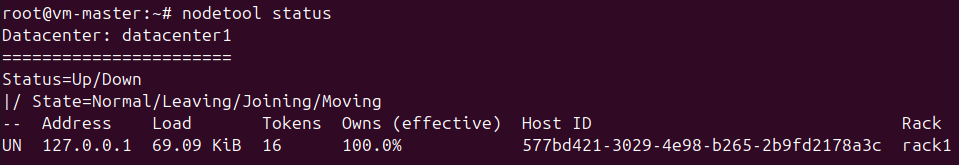
cp /etc/cassandra/cassandra.yaml /etc/cassandra/cassandra.yaml.backup

* access CLI – Cassandra Shell (**cqlsh**)



* Check active cluster:

nodetool status

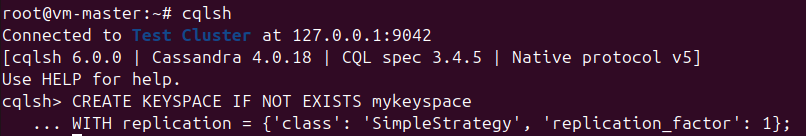


1. **Work with Cassandra – CRUD**

* create Keyspace:

CREATE KEYSPACE IF NOT EXISTS mykeyspace

WITH replication = {'class': 'SimpleStrategy', 'replication\_factor': 1};



\*with:

SimpleStrategy -> simple replication for dev environments

replication\_factor: 1 -> only keep a single copy per node

* Create table:

USE mykeyspace;

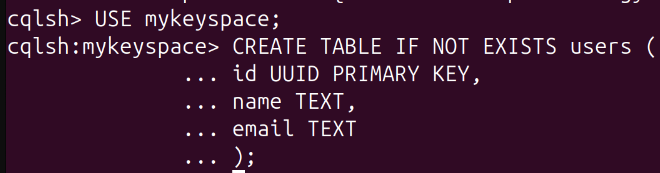
CREATE TABLE IF NOT EXISTS users (

id UUID PRIMARY KEY,

name TEXT,

email TEXT

);



* Create – Read data:

INSERT INTO users (id, name, email) VALUES (uuid(), 'Kiệt', 'kiet@example.com');

SELECT \* FROM users;



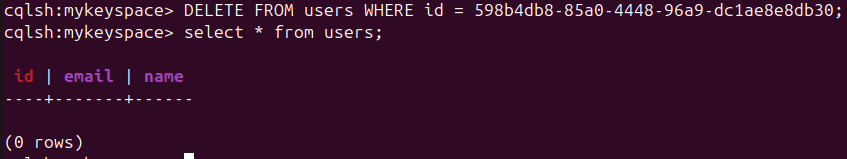
* Update data:

UPDATE users SET email = 'kiet@newdomain.com' WHERE id = **598b4db8-85a0-4448-96a9-dc1ae8e8db30**;



* Delete data:

DELETE FROM users WHERE id = **598b4db8-85a0-4448-96a9-dc1ae8e8db30**;



**Manage table:**

* Delete all data in table:

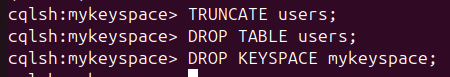
TRUNCATE users;

* delete tables:

DROP TABLE users;

* delete Keyspace:

DROP KEYSPACE mykeyspace;



1. **Download Kafka**

* download and unzip source:

wget <https://archive.apache.org/dist/kafka/3.7.0/kafka_2.13-3.7.0.tgz>

tar -xzf kafka\_2.13-3.7.0.tgz

mv kafka\_2.13-3.7.0 kafka

cd kafka

* config:

nano /root/kafka/config/server.properties

============================

listeners=PLAINTEXT://0.0.0.0:9092 # Listen to all IPs

log.dirs=/tmp/kafka-logs # Directory to save topic logs

zookeeper.connect=localhost:2181 # Connect to Zookeeper

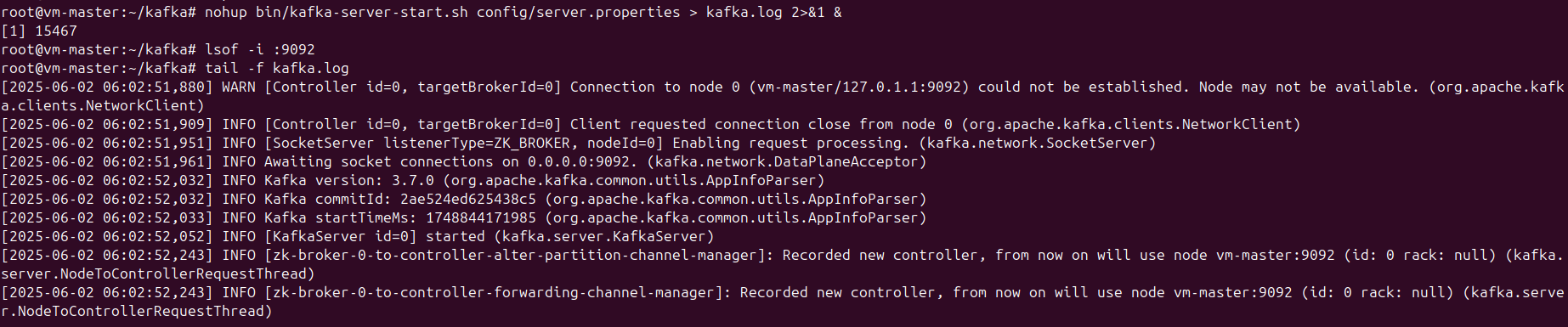
advertised.listeners=PLAINTEXT://**192.168.114.100**:9092 # IP for client to connect

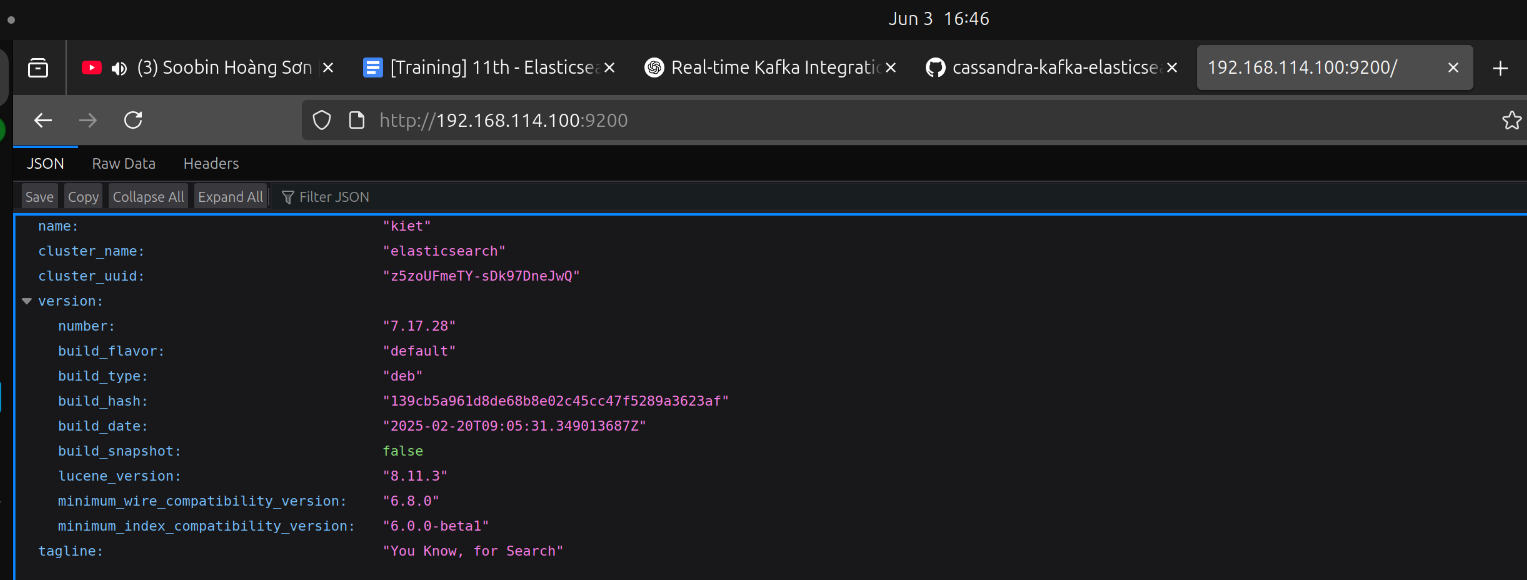
============================

* start zookeeper and kafka:

nohup /root/kafka/bin/zookeeper-server-start.sh /root/kafka/config/zookeeper.properties > zookeeper.log 2>&1 &

nohup /root/kafka/bin/kafka-server-start.sh /root/kafka/config/server.properties > /root/kafka/kafka.log 2>&1 &





* create a topic for exam:

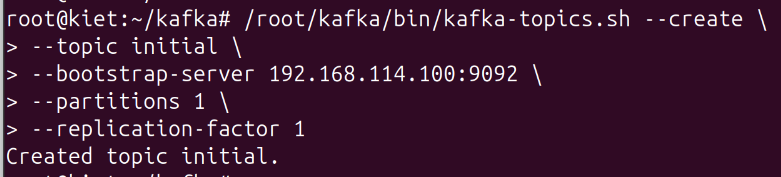
/root/kafka/bin/kafka-topics.sh --create \

--topic initial \

--bootstrap-server **192.168.114.100**:9092 \

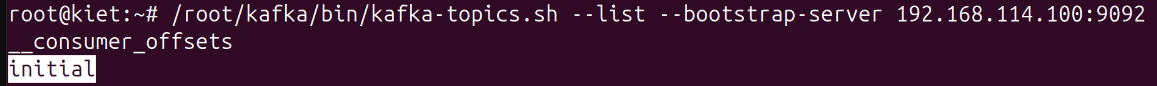
--partitions 1 \

--replication-factor 1



* check list topic:

/root/kafka/bin/kafka-topics.sh --list --bootstrap-server **192.168.114.100**:9092



1. **Kafka Connect Sink: Elasticsearch**

* install connector:

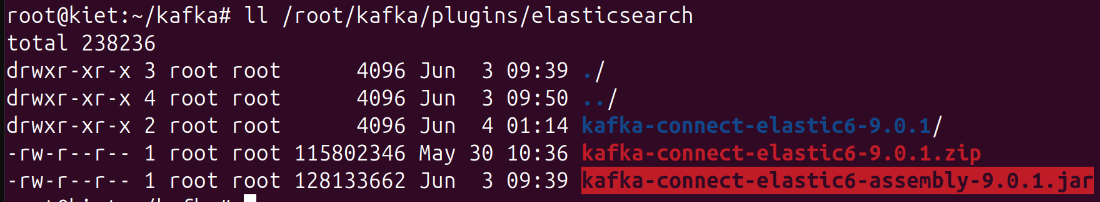
mkdir -p /root/kafka/plugins/elasticsearch

cd /root/kafka/plugins/elasticsearch

wget [https://github.com/lensesio/stream-reactor/releases/download/9.0.1/kafka-connect-elastic6-9.0.1.zi](https://github.com/lensesio/stream-reactor/releases/download/9.0.1/kafka-connect-elastic6-9.0.1.zip)p

unzip kafka-connect-elastic6-9.0.1.zip

* check path **.jar** file kafka connect elasticsearch:



* config **sink** for elasticsearch:

nano /root/kafka/config/elasticsearch-sink.properties

======================================================================

name=elastic6-sink

connector.class=io.lenses.streamreactor.connect.elastic6.ElasticSinkConnector

tasks.max=1

topics=initial

connect.elasticsearch.url=http://**192.168.114.100**:9200

connect.elasticsearch.cluster.name=elasticsearch

connect.elastic.write.ignore.key=true

connect.elastic.write.ignore.schema=true

connect.elastic.kcql=INSERT INTO initial SELECT \* FROM initial STOREAS JSON

======================================================================

\*with:

* kcql language for declaring how data will be transferred
* STOREAS JSON writes data to ES as raw JSON

1. **Kafka Connect Sink: Cassandra**

* connect to cassandra, use key space and create table to check:

cqlsh

> USE mykeyspace;

CREATE TABLE IF NOT EXISTS initial (

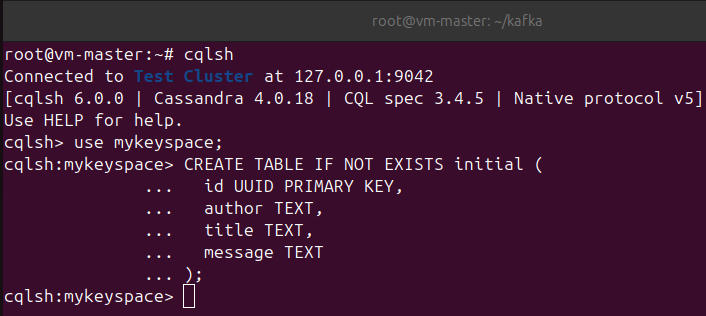
id UUID PRIMARY KEY,

author TEXT,

title TEXT,

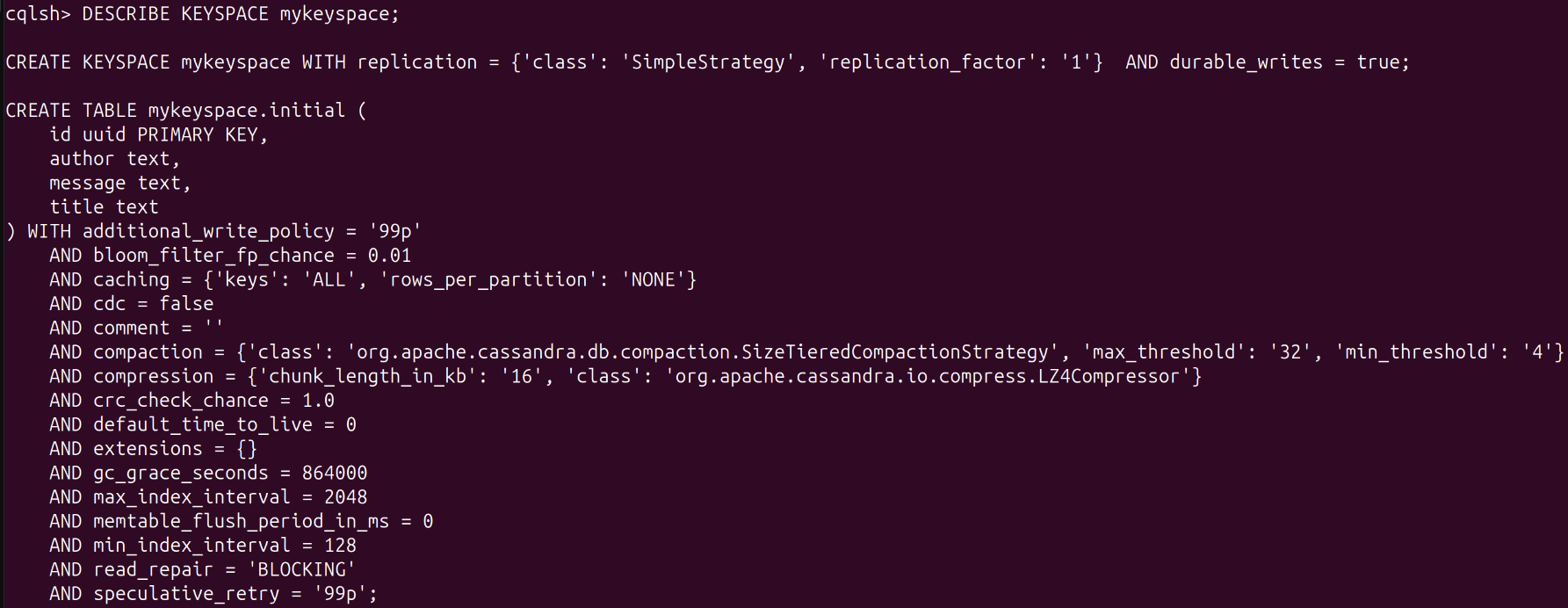
message TEXT

);



* check details key space with cassandra:

> DESCRIBE KEYSPACE mykeyspace;



* check details table in key space with cassandra:

> USE mykeyspace;

> DESCRIBE KEYSPACE mykeyspace;



* install connector :

mkdir -p /root/kafka/plugins/cassandra

cd /root/kafka/plugins/cassandra

wget <https://github.com/lensesio/stream-reactor/releases/download/9.0.1/kafka-connect-cassandra-9.0.1.zip>

unzip kafka-connect-cassandra-9.0.1.zip



* config **sink** for cassandra:

nano /root/kafka/config/cassandra-sink.properties

======================================================================

name=cassandra-sink

connector.class=io.lenses.streamreactor.connect.cassandra.sink.CassandraSinkConnector

tasks.max=1

topics=initial

connect.cassandra.contact.points=localhost

connect.cassandra.port=9042

connect.cassandra.key.space=mykeyspace

connect.cassandra.consistency.level=ONE

connect.cassandra.kcql=INSERT INTO initial SELECT id, author, title, message FROM initial

======================================================================

1. **Config connect standalone - Insert data into topic initial to test:**

uuidgen -> e1f7ac84-40e5-11f0-964a-4f97696080bc

uuidgen -> 8c754c80-4c06-4363-8136-c06866539c18

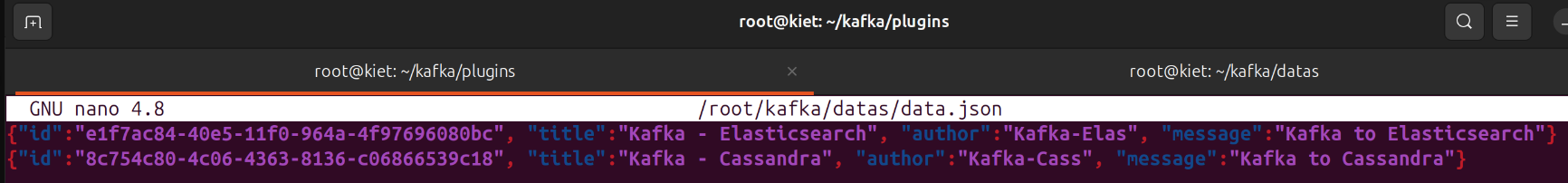
nano /root/kafka/datas/data.json

=====================================================================

{"id":"e1f7ac84-40e5-11f0-964a-4f97696080bc", "title":"Kafka - Elasticsearch", "author":"Kafka-Elas", "message":"Kafka to Elasticsearch"}

{"id":"8c754c80-4c06-4363-8136-c06866539c18", "title":"Kafka - Cassandra", "author":"Kafka-Cass", "message":"Kafka to Cassandra"}

=====================================================================



* insert data to kafka console producer with **data.json** file:

cat ~/kafka/datas/data.json | /root/kafka/bin/kafka-console-producer.sh --broker-list **192.168.114**.100:9092 --topic initial

* check datas in kafka console consumer:

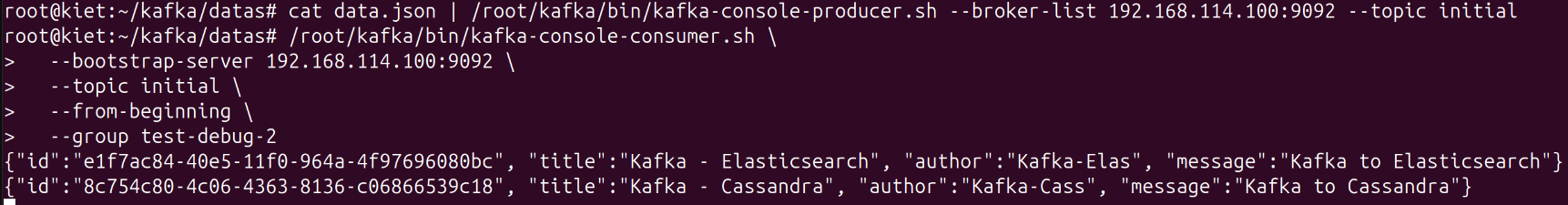
/root/kafka/bin/kafka-console-consumer.sh \

--bootstrap-server **192.168.114.100**:9092 \

--topic initial \

--from-beginning

--group test-debug-2



# Config **connect standalone** for cassandra + elasticsearch:

nano /root/kafka/config/connect-standalone.properties

======================================================================

bootstrap.servers=**192.168.114.100**:9092

key.converter=org.apache.kafka.connect.storage.StringConverter

value.converter=org.apache.kafka.connect.json.JsonConverter

key.converter.schemas.enable=false

value.converter.schemas.enable=false

offset.storage.file.filename=/tmp/connect.offsets

plugin.path=/root/kafka/plugins/elasticsearch,/root/kafka/plugins/cassandra

======================================================================

* run composite with 2 Sink Connectors:

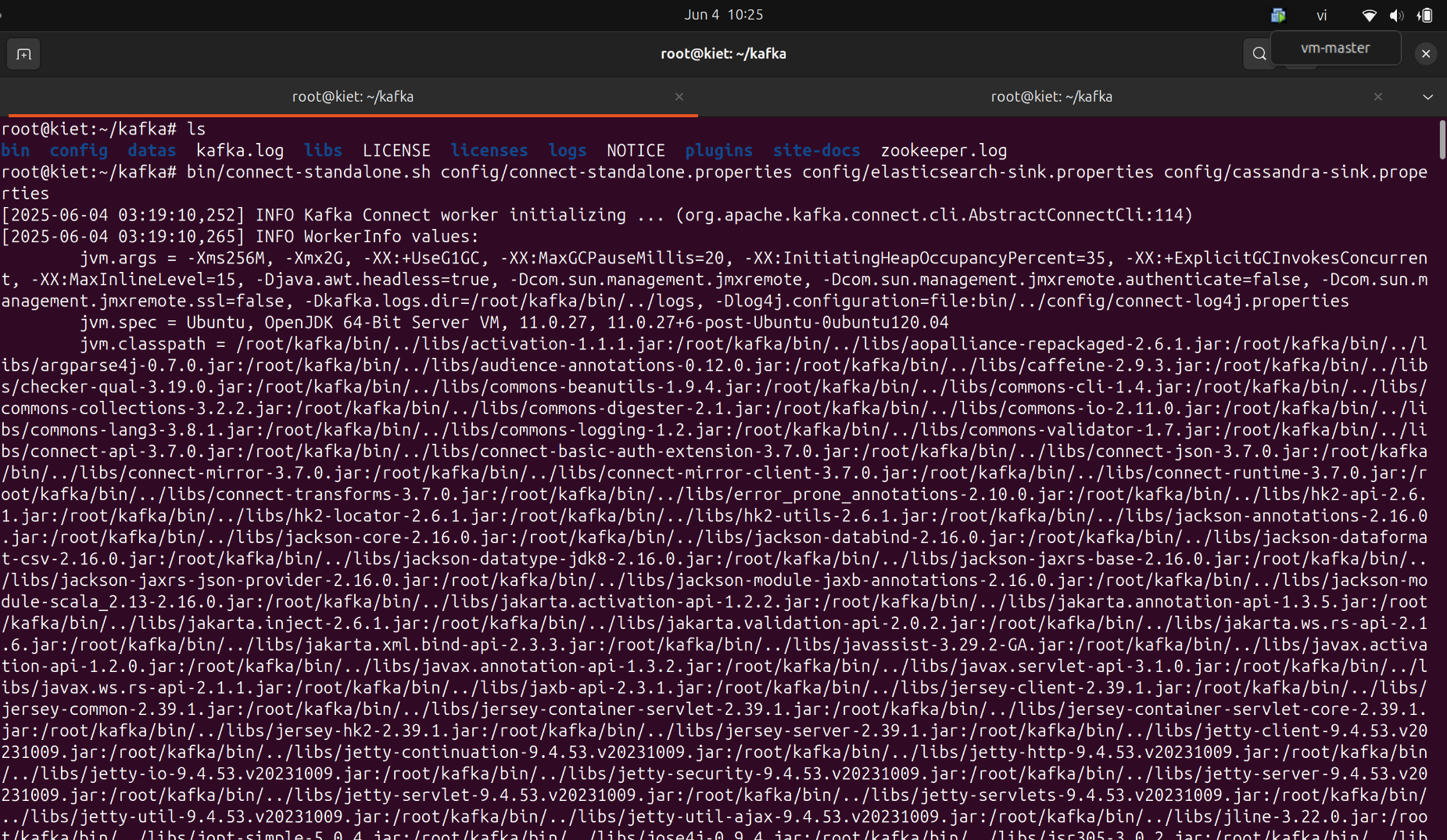
cd ~/kafka

bin/connect-standalone.sh \

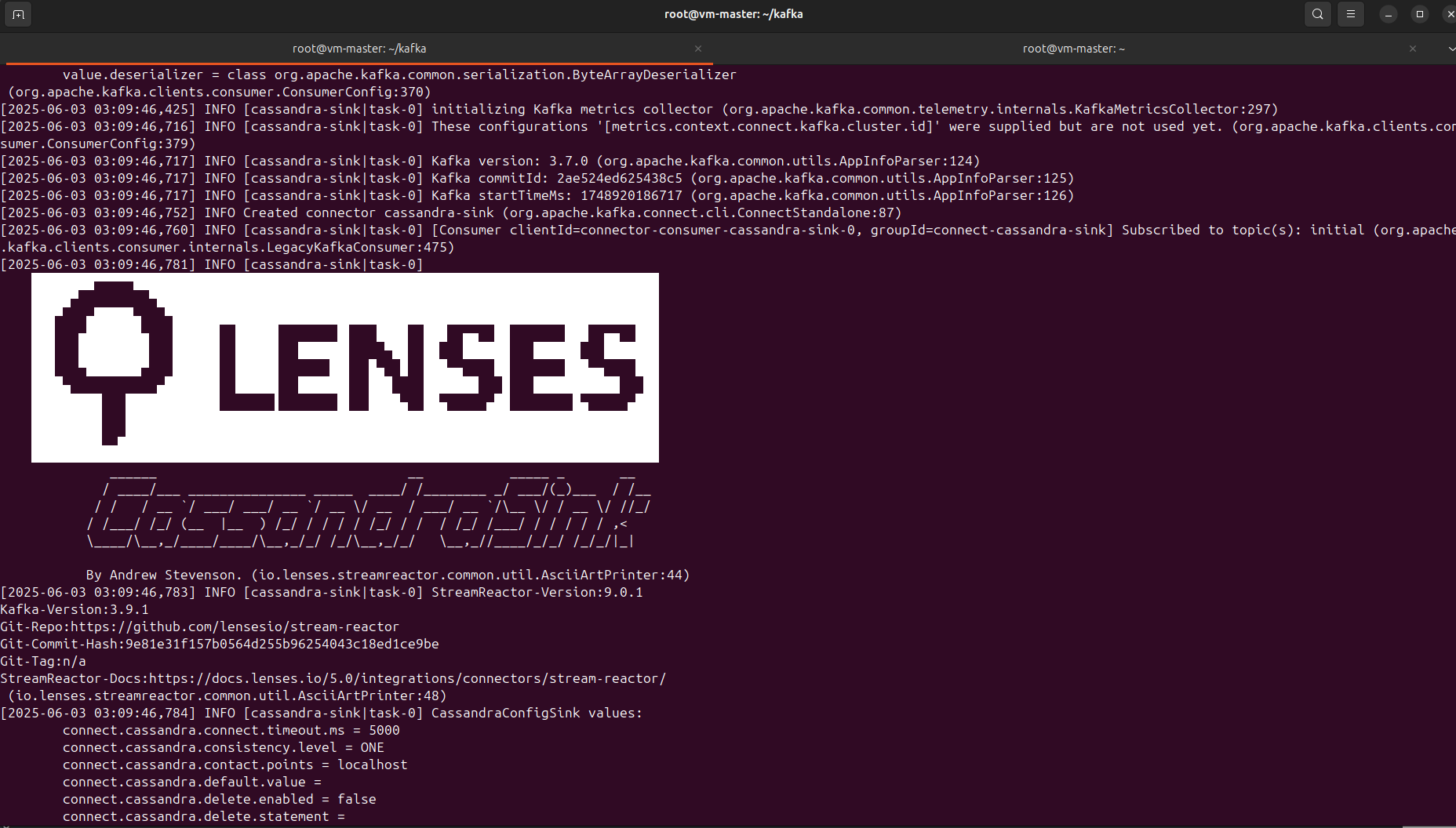
config/connect-standalone.properties \

config/elasticsearch-sink.properties \

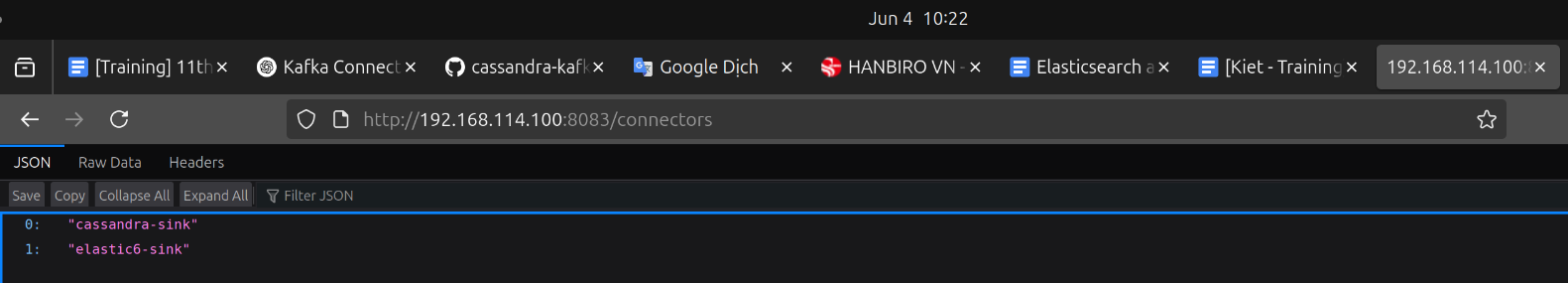
config/cassandra-sink.properties



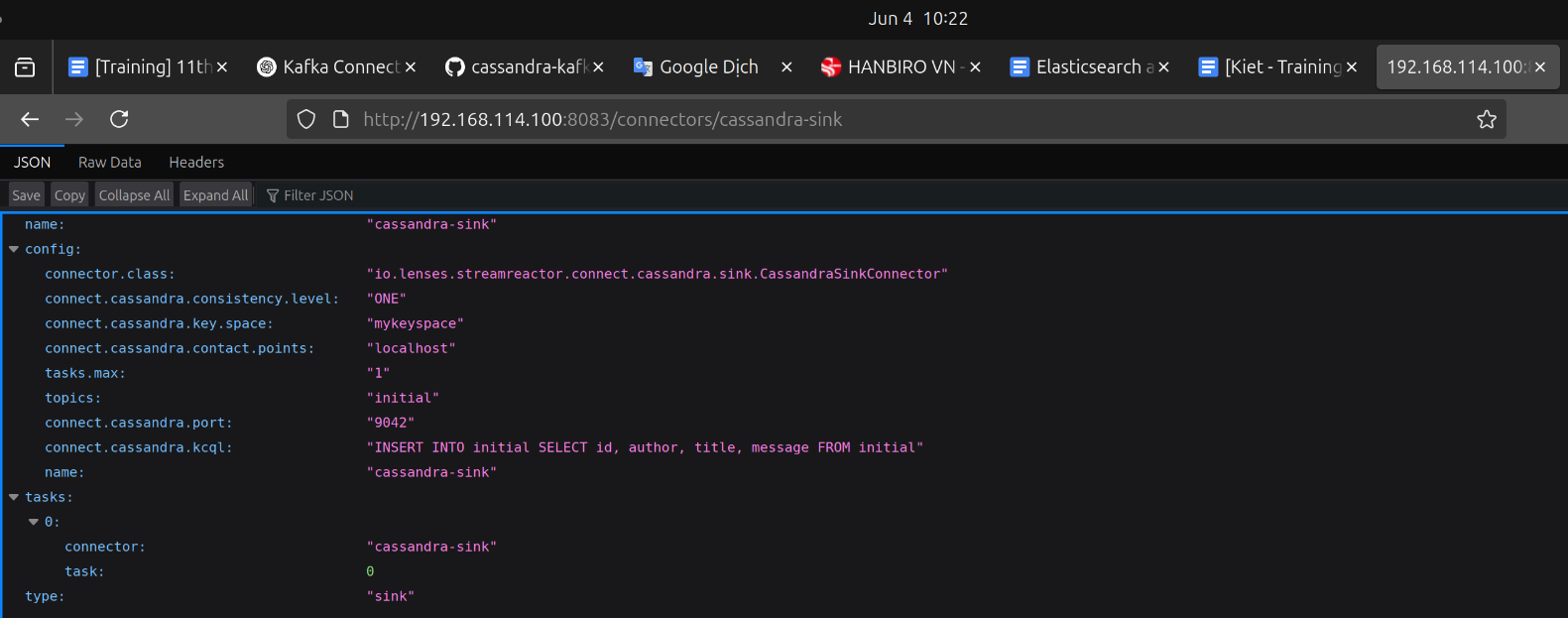


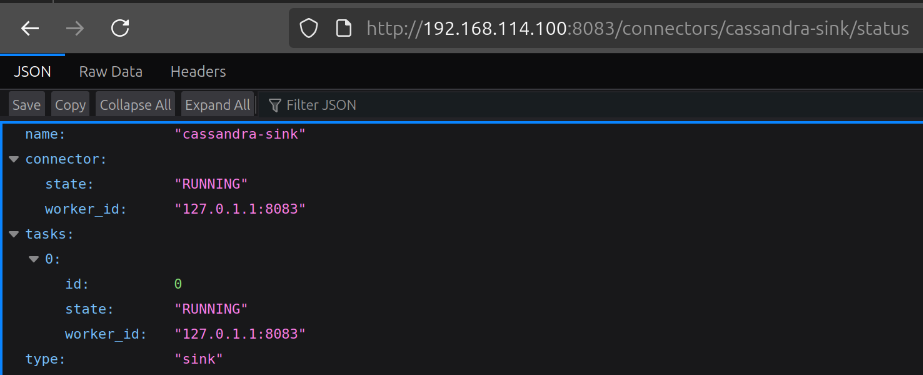


* check all connectors:

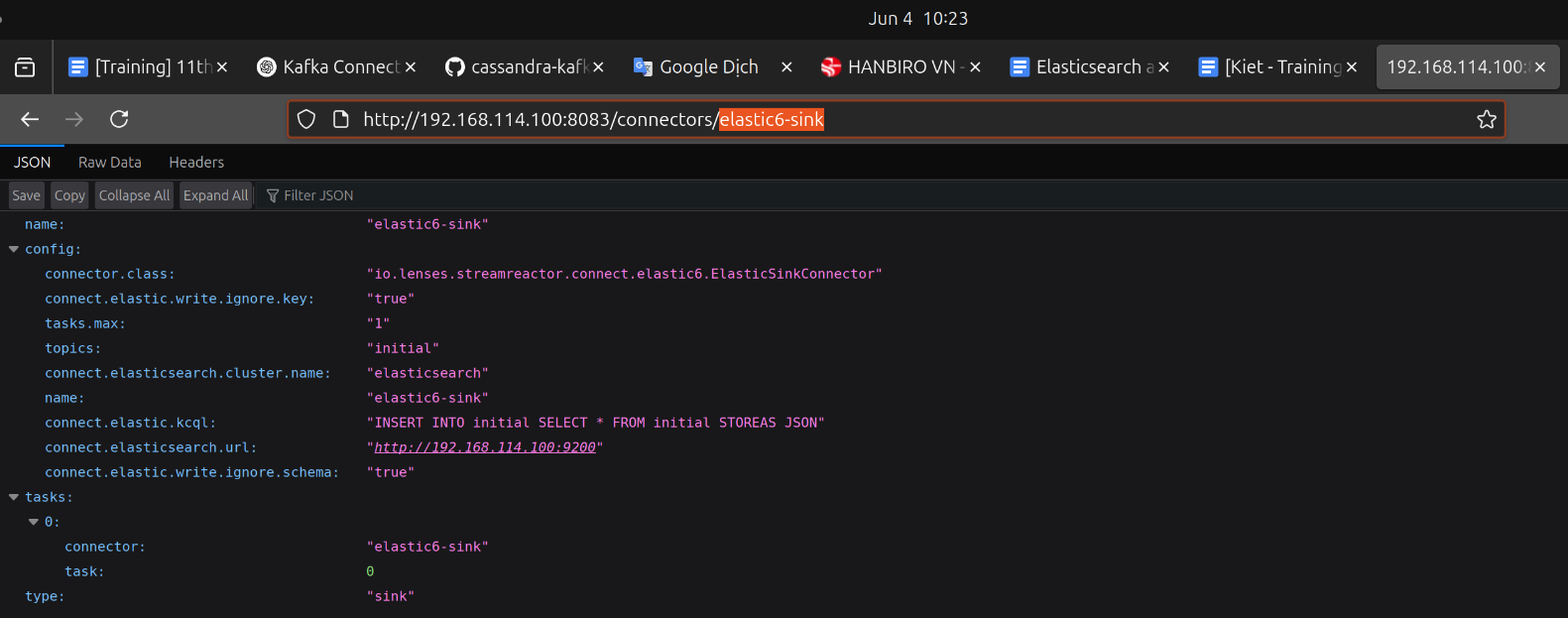


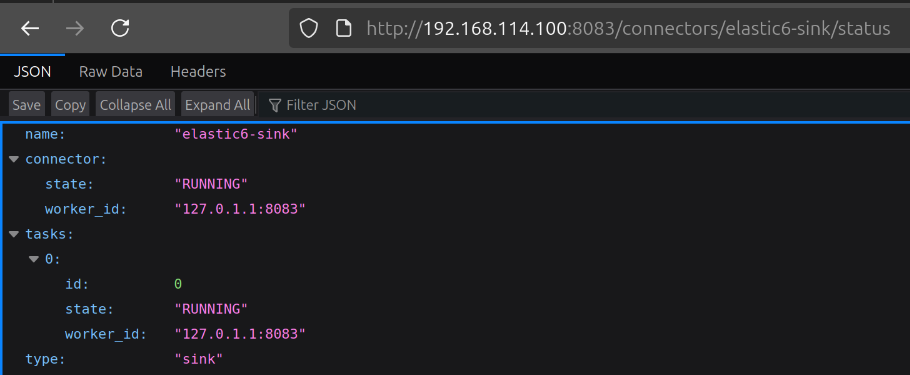
* check connector and status “cassandra-sink”:





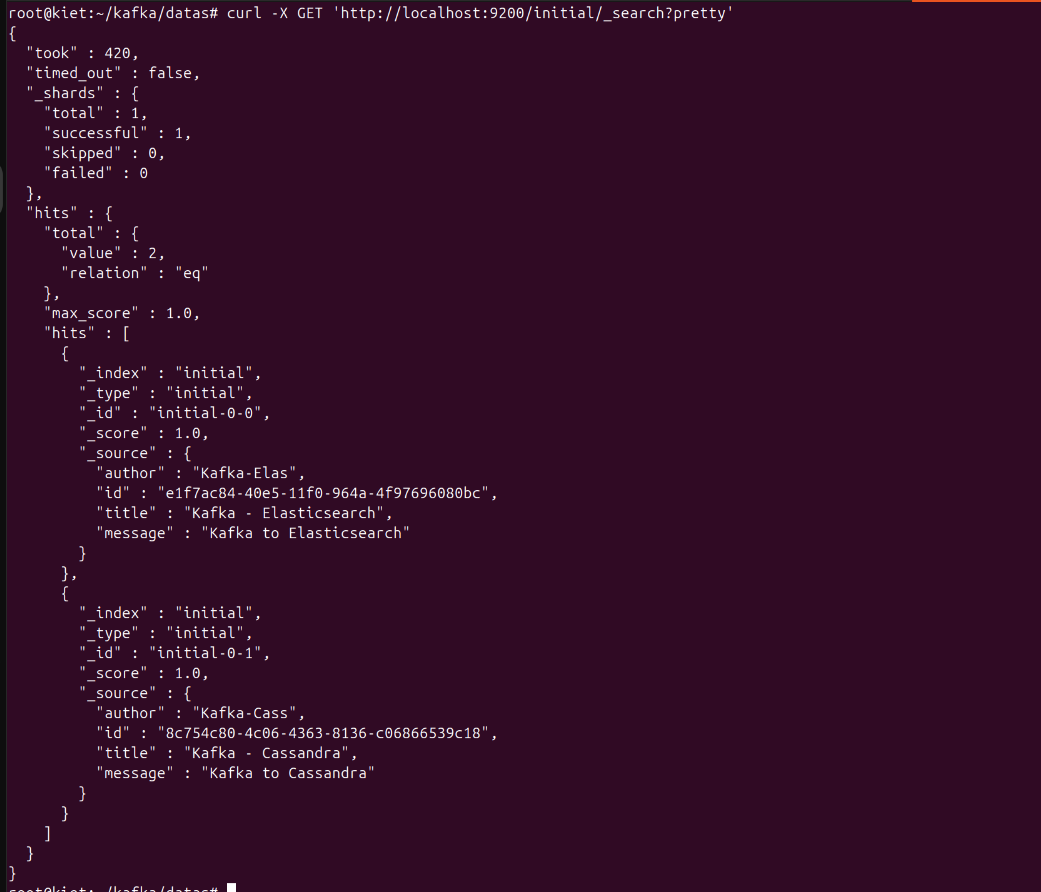
* check connector and status “elastic6-sink”:





* Check if data in **Elasticsearch** is **written from Kafka**:

curl -X GET 'http://localhost:9200/initial/\_search?pretty'



* Check if data in **Cassandra** is **written from Kafka**:

cqlsh> select \* from mykeyspace.initial;



* create new data to RETEST while **connector is still running with two sinks (elasticsearch + cassandra)** to check if they are simultaneously adding data:

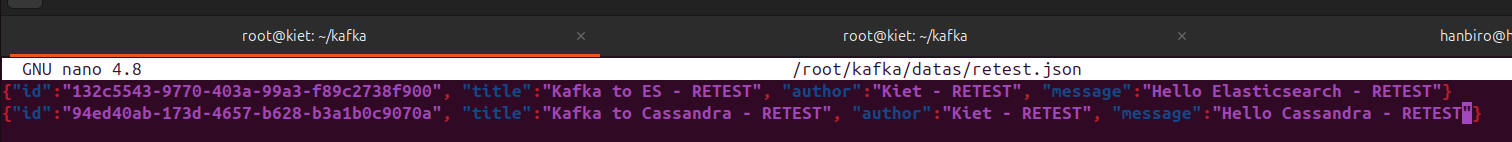
nano /root/kafka/datas/retest.json

======================================================================

{"id":"132c5543-9770-403a-99a3-f89c2738f900", "title":"Kafka to ES - RETEST", "author":"Kiet - RETEST", "message":"Hello Elasticsearch - RETEST"}

{"id":"94ed40ab-173d-4657-b628-b3a1b0c9070a", "title":"Kafka to Cassandra - RETEST", "author":"Kiet - RETEST", "message":"Hello Cassandra - RETEST"}

======================================================================



* insert data with json file:

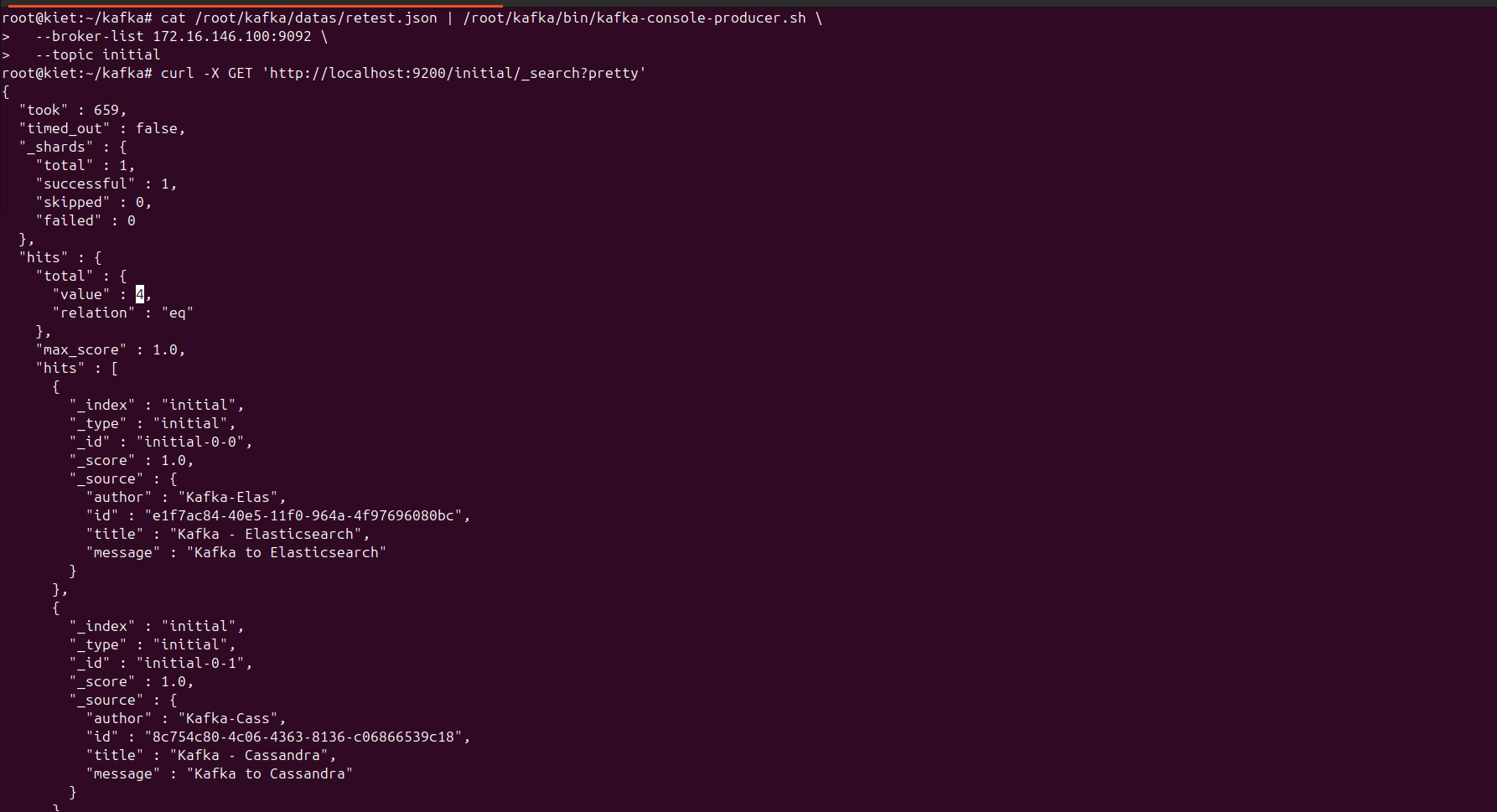
cat /root/kafka/datas/retest.json | /root/kafka/bin/kafka-console-producer.sh \

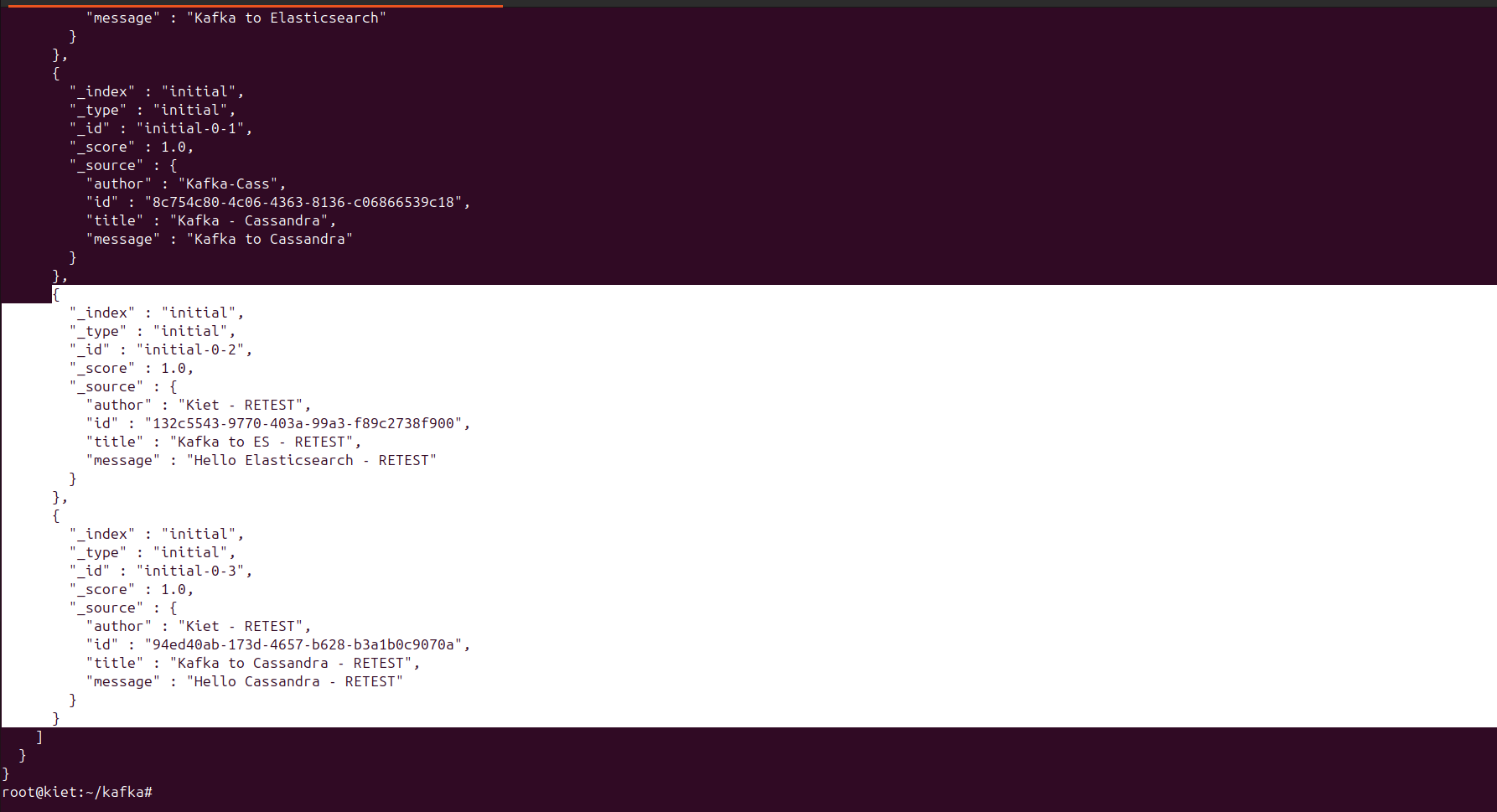
--broker-list **172.16.146.100**:9092 \

--topic initial

* check data in **elasticsearch**:

curl -X GET 'http://localhost:9200/initial/\_search?pretty'





* check data in **cassandra**:

**cqlsh**> select \* from mykeyspace.initial;

