# MODUL KASPA SERVICE & CLIENT – DEPLOY SENSOR

## Docker

*Docker* diperlukan untuk beberapa *case*, yaitu untuk menjalankan sensor *Snort*, menjalankan *user interface* untuk *Kafka-topics* dan *Kafka-connect-schema-registry*.Lakukan instalasi docker pada server yang digunakan sebagai sensor. Langkah instalasinya adalah sebagai berikut,

### Instalasi Docker

Jalankan perintah berikut untuk melakukan instalasi docker

|  |
| --- |
| # apt-get install apt-transport-https ca-certificates curl gnupg-agent software-properties-common  # curl -fsSL https://download.docker.com/linux/ubuntu/gpg | apt-key add -  # apt-key fingerprint 0EBFCD88  # add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb\_release -cs) stable"  # apt update  # apt install docker-ce docker-ce-cli containerd.io |

## Supervisord

**Supervisord**adalah sebuah package process manager untuk sistem client/server yang memungkinkan kita lebih mudah dalam memonitor dan mengontrol sejumlah proses program yang kita buat pada sistem operasi UNIX

### Instalasi Supervisord

Jalankan perintah berikut untuk melakukan instalasi supervisord

|  |
| --- |
| # apt install supervisor |

## Deploy Kaspa Service & UI

Pertama buat direktori baru bernama **snoqtt** pada home direktori user hduser. Direktori **snoqtt** digunakan untuk menampung semua program yang digunakan nantinya, sehingga sebagai default direktori proyek. Semua konfigurasi IP address mengacu pada *IP address dari service yang telah dipasang pada server*, IP address dibawah hanya contoh.

|  |
| --- |
| $ cd ~  $ mkdir snoqtt  $ cd snoqtt |

Install python virtual environment menggunakan perintah berikut

|  |
| --- |
| $ pip install virtualenv |

### Instalasi Kaspa Service

Clone repository proyek kaspa service

|  |
| --- |
| $ git clone https://github.com/dimasmamot/KaspaServices.git |

Setelah itu

Masuk ke dalam direktori proyek dan buat virtual environment

|  |
| --- |
| $ cd KaspaServices  $ python -m virtualenv venv |

Lalu aktifkan virtual environment dengan menjalankan perintah berikut dan install dependency yang diperlukan program.

Perlu diketahui **(venv)** menunjukkan kita berada dalam virtual environment dan setiap menjalankan apapun harus menggunakan environment tersebut.

|  |
| --- |
| $ source venv/bin/activate |

Jika terdapat **pkg-resources==0.0.0** dalam file requirements.txt hapus baris tersebut terlebih dahulu.

|  |
| --- |
| (venv) pip install -r requirements.txt |

Pada direktori Kaspa Service buat file environment baru

|  |
| --- |
| $ nano .env  CASSANDRA\_USERNAME=admin  CASSANDRA\_PASSWORD=<password\_cassandra>  CASSANDRA\_CLUSTER\_HOST=127.0.0.1  DEFAULT\_OINKCODE=9769c7431c3fadc5efdf7e28610815c5xxxxxxxx  SECRET\_KEY=06f492bdad114fb79cf9757b9cce7441  FLASK\_APP=kaspaservices.py |

### Instalasi Kaspa Client

Tetap pada direktori /home/hduser/snoqtt

Clone repository proyek kaspa client

|  |
| --- |
| $ git clone https://github.com/dimasmamot/KaspaClient.git |

Setelah itu

Masuk ke dalam direktori proyek dan buat virtual environment

|  |
| --- |
| $ cd KaspaClient  $ python -m virtualenv venv |

Lalu aktifkan virtual environment dengan menjalankan perintah berikut dan install dependency yang diperlukan program.

Perlu diketahui **(venv)** menunjukkan kita berada dalam virtual environment dan setiap menjalankan apapun harus menggunakan environment tersebut.

|  |
| --- |
| $ source venv/bin/activate |

Install requirement

|  |
| --- |
| (venv) pip install -r requirements.txt |

Pada direktori Kaspa Client buat file environment baru

|  |
| --- |
| $ nano .env  API\_HOST=10.130.122.127:5000  API\_USER=admin  API\_PASS=admin  SECRET\_TOKEN=eyJhbGciOiJIUzI1NiIsImlhdCI6MTU1MTU0MDczMywiZXhwIjoxNTgzMDc2NzMzfQ.eyJ1c2VybmFtZSI6ImFkbWluIn0.VC577rSpgTa80EPoJ6qYYW5LhBhGjv-gybYVHvftVHE  SECRET\_PASSWORD=wumboakuwumbo  SECRET\_KEY=6d4fdf7c805949be88b5470dedbcb061  FLASK\_ENV=development  FLASK\_APP=kaspaclient.py  FLASK\_DEBUG=1 |

### Setup Supervisord

Buat file konfigurasi untuk membuat kaspa service dan kaspa client berjalan di background.

Untuk kaspa service seperti berikut ini.

|  |
| --- |
| # nano /etc/supervisor/conf.d/kaspaservices.conf  [program:kaspaservices]  command=/home/hduser/snoqtt/KaspaServices/venv/bin/gunicorn -b 0.0.0.0:5000 -w 4 kaspaservices:app  directory=/home/hduser/snoqtt/KaspaServices  user=root  autostart=true  autorestart=true  stopasgroup=true  killasgroup=true |

Untuk kaspa client seperti berikut ini.

|  |
| --- |
| # nano /etc/supervisor/conf.d/kaspaclient.conf  [program:kaspaclient]  Command=/home/hduser/snoqtt/KaspaClient/venv/bin/gunicorn -b 0.0.0.0:5001 -w 4 kaspaclient:app  directory=/home/hduser/snoqtt/KaspaClient  user=root  autostart=true  autorestart=true  stopasgroup=true  killasgroup=true |

### Menjalankan Kaspa

Pertama lakukan restart daemon dari supervisor

|  |
| --- |
| # systemctl restart supervisor |

Untuk menjalankan supervisor yang telah dibuat lakukan perintah berikut

|  |
| --- |
| # supervisorctl start kaspaservices  # supervisorctl start kaspaclient |

Untuk melihat status

|  |
| --- |
| # supervisorctl status |

Untuk menghentikan supervisor

|  |
| --- |
| # supervisorctl stop kaspaservices  # supervisorctl stop kaspaclient |

### Membuat User

Anda masih dapat menggunakan user sebelumnya yaitu admin / admin.

**Menggunakan Command Line**

Tetapi jika ingin membuat membuat user baru lakukan request POST pada kaspa service dengan parameter berikut ke http://ip\_server:5000/api/users/v1.0/createuser

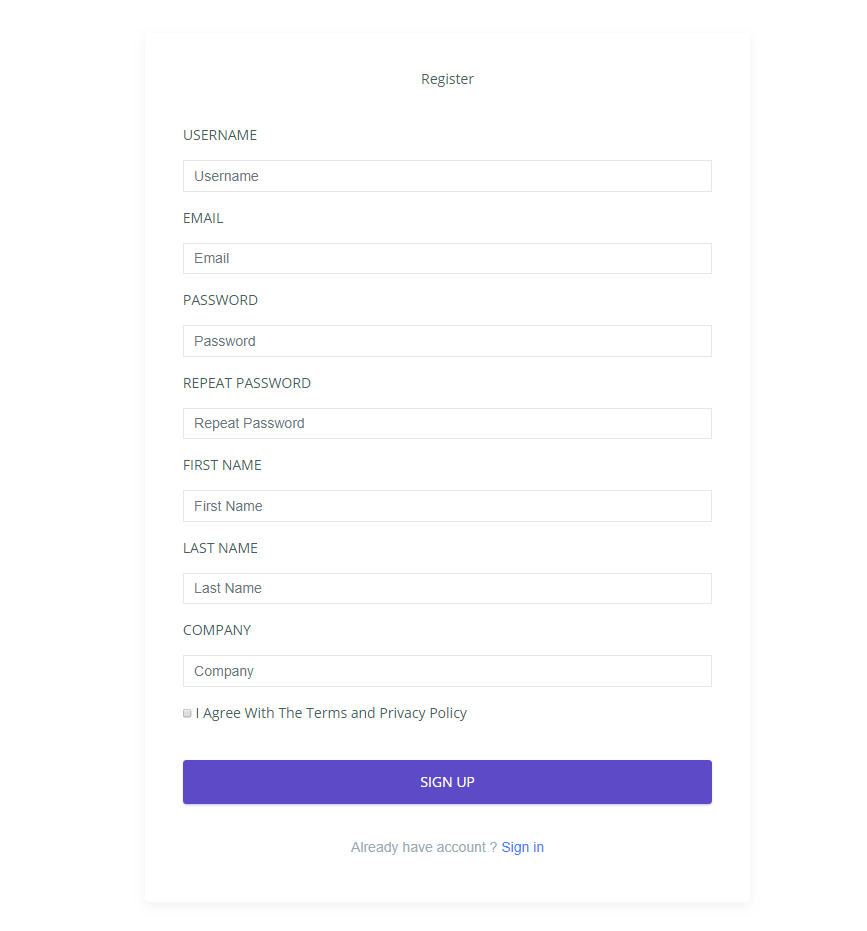
* username
* password
* first\_name
* last\_name
* email
* company

Contohnya dengan menggunakan cURL, untuk membuat user baru diatas.

|  |
| --- |
| curl -X POST \  http://157.230.241.208:5000/api/users/v1.0/createuser \  -H 'Content-Type: application/json' \  -d '{  "username": "userbaru",  "password": "userbaru",  "first\_name": "Hduser",  "last\_name": "user",  "email": "hduser@gmail.com",  "company": "C307"  }' |

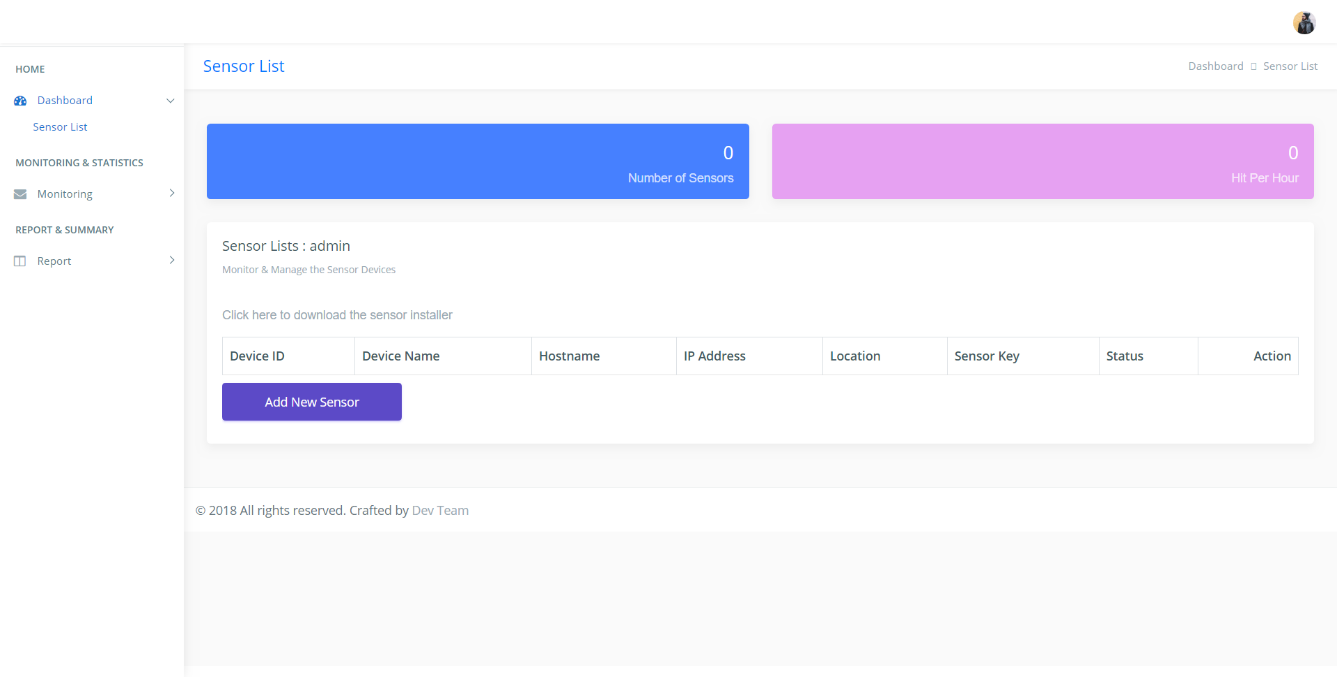
**Menggunakan Kaspa Client**

Buka kaspa client menggunakan browser http://ip\_server:5000 dan pilih menu Don't have account? Sign Up Here. Setelah masuk halaman registrasi, isi form sesuai yang diminta dan submit form.



Gambar 3.3.1 Tampilan menu registrasi (Kaspa Client)

Setelah selesai, langkah selanjutnya yaitu login ke aplikasi. Jika berhasil maka akan muncul tampilan seperti berikut.

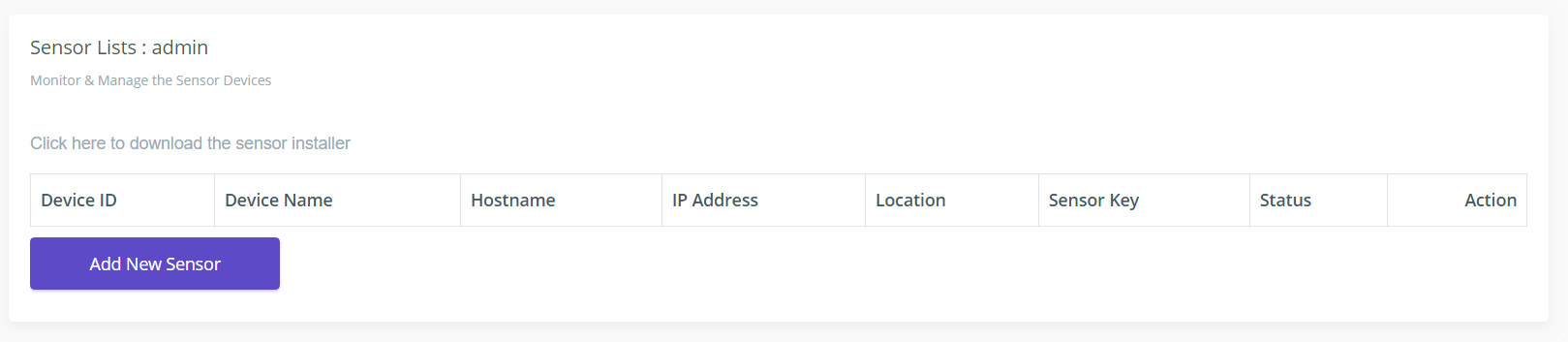


Gambar 3.3.2 Tampilan menu dashboard (Kaspa Client)

## Deploy Sensor

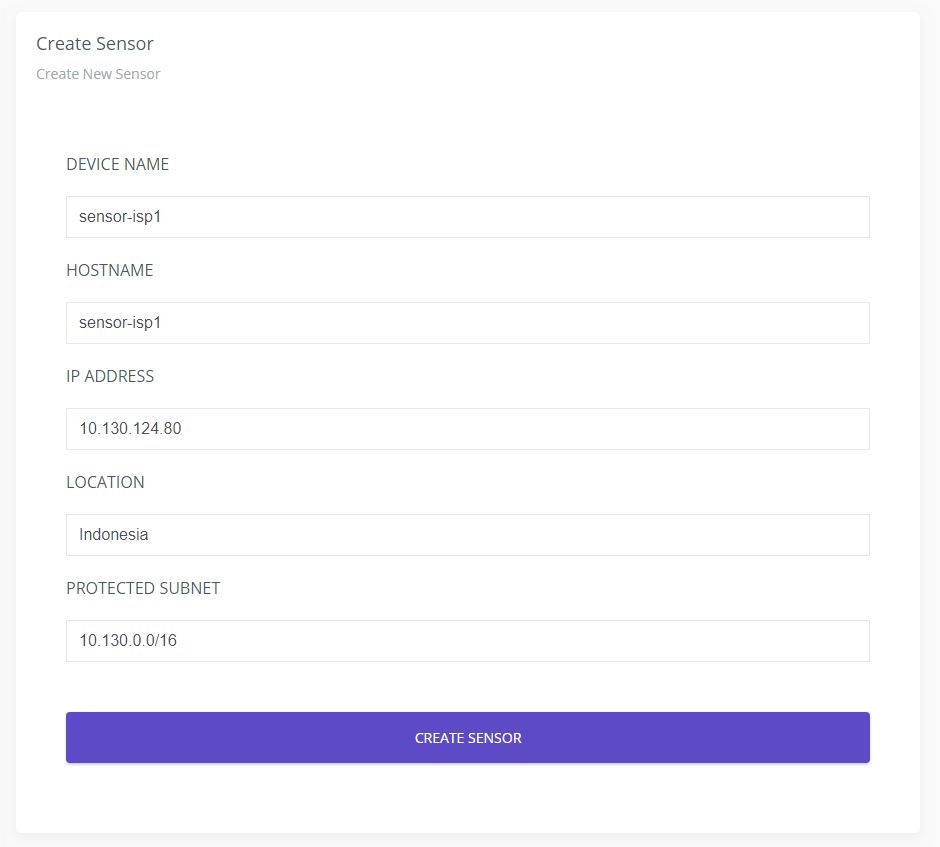
### Membuat Sensor

Pertama buka web kaspa client untuk membuat sensor baru, pertama login dan dan pada halaman awal dashboard, klik tombol Add New Sensor



Gambar 3.4.1 Klik pada tombol Add New Sensor

Selanjutnya masukkan informasi seperti nama sensor, hostname, ip address, location, protected subnet, seperti contoh berikut.

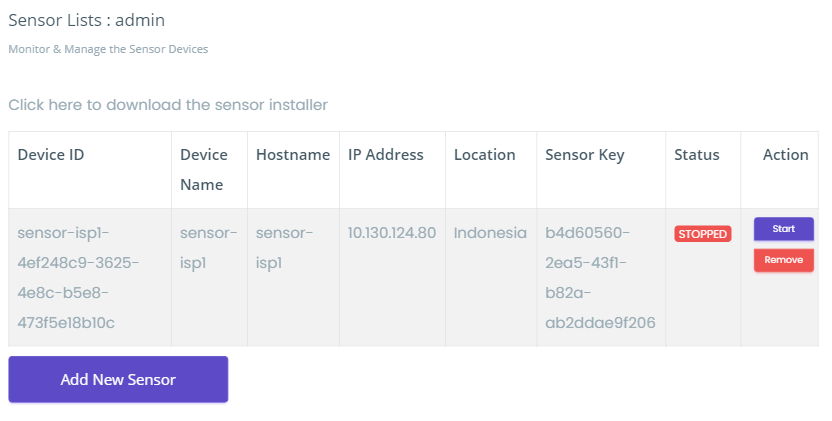


Gambar 3.4.2 Menambahkan Sensor Baru

Keterangan:

* Device Name: nama dari sebuah sensor
* Hostname: hostname dari server yang nantinya akan dipasang sensor
* IP address: IP address dari server yang nantinya akan dipasang sensor
* Location: Lokasi geografis dari server
* Protected Subnet: Subnet yang nantinya akan dilindungi oleh sensor

Setelah berhasil menambahkan sensor, hasilnya akan tampil seperti pada halaman berikut ini



Gambar 3.4.3 Tampilan dari sensor yang telah berhasil diambahkan

Setelah sukses catat atau copy sensor key dan device id

Unduh file bash script installer dan jalankan pada server yang akan dijadikan sensor (yang sudah terinstall docker sebelumnya).

|  |
| --- |
| # wget --content-disposition http://<host>:5000/api/sensors/v1.0/downloadinstaller  # chmod +x ./installer.sh  # ./installer.sh |

Masukkan informasi yang diminta…

Tunggu proses berjalanan sampai selesai. Jika berhasil cek apakah sudah terdapat direktori /etc/snoqtt

Jika sudah maka pindah ke direktori tersebut, build docker image dan jalankan.

|  |
| --- |
| # cd /etc/snoqtt  # ./build\_snoqtt.sh  # ./start\_snoqtt.sh |

Untuk menghentikan sensor tinggal jalankan perintah

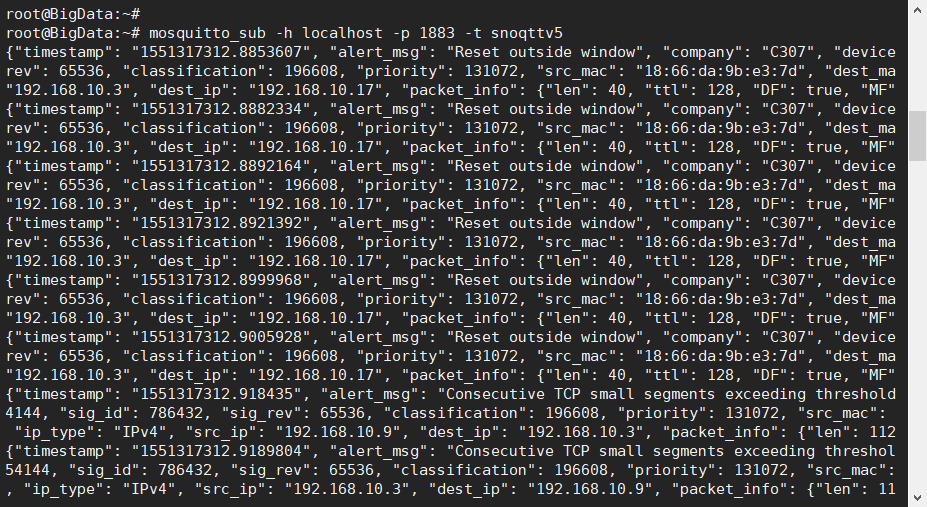
|  |
| --- |
| # ./stop\_snoqtt.sh |

### Cek Apakah Sensor Berhasil di Deploy

Untuk melakukan pengujian apakah sensor berhasil dideploy dengan benar, yang pertama lakukan koneksi ke server master (yang terinstall mqtt dan confluent). Lalu jalankan perintah berikut ini. Gunakan salah satu, atau bergantian jika salah satu berhasil.

Melakukan pengujian dengan menggunakan mosquitto

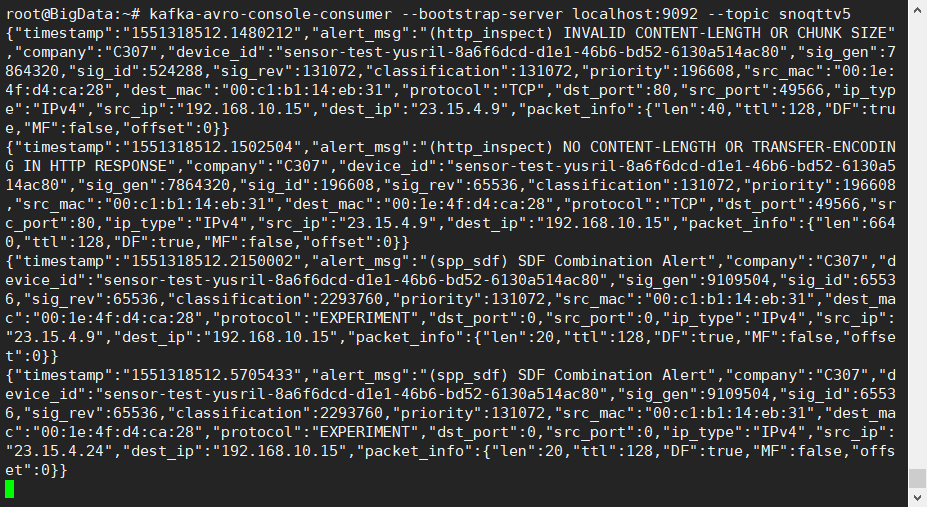
|  |
| --- |
| # mosquitto\_sub -h localhost -p 1883 -t snoqttv5 |



Gambar 3.4.4 Output pesan yag ditangkap oleh mosquitto

Melakukan pengujian dengan menggunakan kafka consumer

|  |
| --- |
| # kafka-avro-console-consumer --bootstrap-server localhost:9092 --topic snoqttv5 |



Gambar 3.4.5 Output pesan yang ditangkap oleh Kafka Consumer

Lalu lakukan serangan pada sensor contohnya menggunakan tool nmap, owasp, dsb. lalu pantau terminal yang menjalankan perintah mosquitto atau kafka diatas apakah terdapat *message* yang masuk. Jika ada maka berhasil.

## Deploy Spark Streaming Job

### Clone Repositori

Pertama lakukan clone repositori kaspa core system

|  |
| --- |
| # git clone https://github.com/dimasmamot/KaspaCoreSystem.git |

Lalu ubah konfigurasi aplikasi menyesuaikan environment yang dibuat. File konfigurasi terletak di KaspaCoreSystem/src/main/resources/application.conf

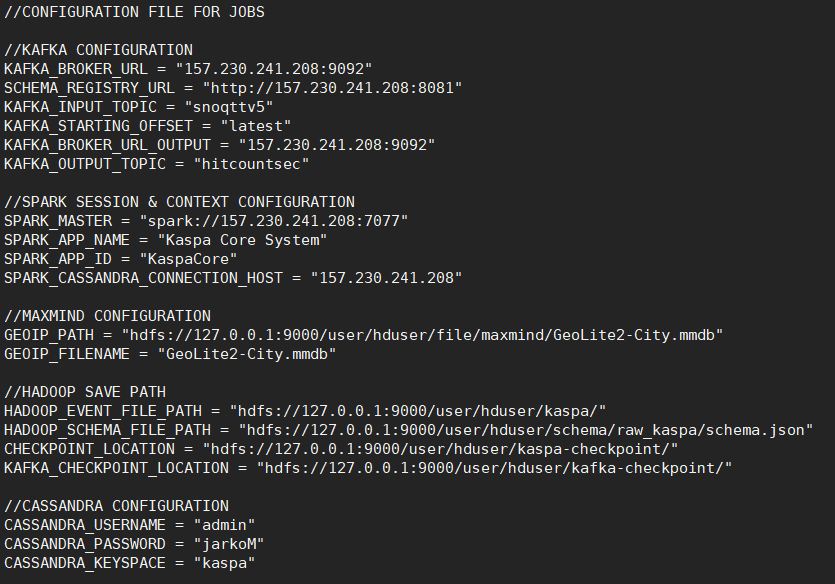
Berikut adalah daftar konfigurasi yang dapat diubah.

|  |  |
| --- | --- |
| Nama Konfigurasi | Deskripsi |
| KAFKA\_BROKER\_URL | Url kafka broker, server yang terinstall kafka. Contoh: “10.252.108.22:9092” |
| SCHEMA\_REGISTRY\_URL | Url tempat disimpannya schema registry, server yang terinstall kafka. Contoh: “http://10.252.108.22:8081” |
| KAFKA\_BROKER\_URL\_OUTPUT | URL Output kafka broker, server yang terinstall kafka. Contoh: “10.252.108.22:9092” |
| SPARK\_MASTER | Url master dari spark, server yang terinstall spark. Contoh: "spark://10.252.108.22:7077" |
| SPARK\_CASSANDRA\_CONNECTION\_HOST | Ip dari server yang terinstall Cassandra. Contoh: "10.252.108.98" |
| GEOIP\_PATH | Lokasi disimpannya file GeoIP. Contoh: "hdfs://10.252.108.22:9000/user/hduser /file/maxmind/GeoLite2-City.mmdb" |
| GEOIP\_FILENAME | Nama dari file GeoIP |
| HADOOP\_EVENT\_FILE\_PATH | Tempat disimpannya event yang ditangkap sensor |
| HADOOP\_SCHEMA\_FILE\_PATH | Tempat disimpannya schema spark |
| CHECKPOINT\_LOCATION | Tempat disimpannya checkpopint streaming dari aplikasi |
| KAFKA\_CHECKPOINT\_LOCATION | Tempat disimpannya checkpopint kafka |
| CASSANDRA\_USERNAME | Username dari cassandra yang telah dibuat sebelumnya |
| CASSANDRA\_PASSWORD | Password dari cassandra yang telah dibuat sebelumnya |

Semua hal yang berhubungan dengan hdfs, gunakan IP yang digunakan pada konfigurasi hdfs pada file core-site.xml. jika menggunakan localhost, set menjadi localhost / 127.0.0.1, jika menggunakan nama domain / ip lain, gunakanlah IP tersebut.

Dan untuk semua yang berhubungan dengan spark, gunakan IP / domain yang digunakan untuk menjalankan spark master seperti 157.230.241.208

Contoh konfigurasi:



Gambar 3.5.1 Contoh konfigurasi job spark

### Upload Schema

Upload schema yang berada di KaspaCoreSystem/src/main/resources/schema.json menggunakan perintah berikut.

|  |
| --- |
| $ cd KaspaCoreSystem  $ hdfs dfs -put src/main/resources/schema.json hdfs://localhost:9000/user/hduser/schema/raw\_kaspa |

### Upload Database Maxmind

Pertama unduh terlebih dahulu database yang versi lite (gratis) jika anda punya yang berbayar silahkan menggunakan yang berbayar.

|  |
| --- |
| $ wget https://geolite.maxmind.com/download/geoip/database/GeoLite2-City.tar.gz  $ tar xzf GeoLite2-City.tar.gz  $ cd GeoLite2-City\*  $ hdfs dfs -put GeoLite2-City.mmdb hdfs://localhost:9000/user/hduser/file/maxmind/GeoLite2-City.mmdb |

### Compile

Pada tahap ini akan dilakukan compile pada KaspaCoreSystem

|  |
| --- |
| $ cd KaspaCoreSystem  $ sbt assembly |

Jika sudah selesai maka upload jar ke HDFS.

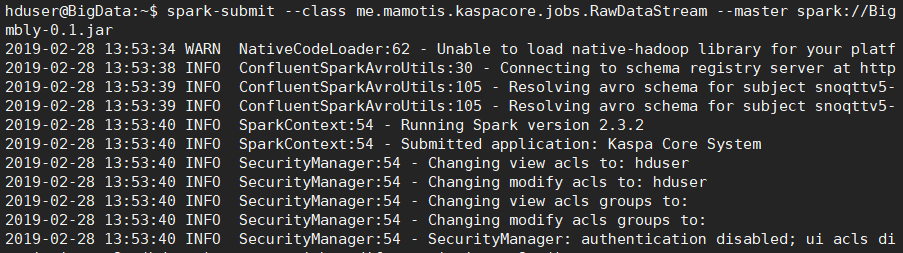
|  |
| --- |
| $ hdfs dfs -put target/scala-2.11/KaspaCore-assembly-0.1.jar hdfs://localhost:9000/user/hduser/job/KaspaCore-assembly-0.1.jar |

### Menjalankan Job Spark Streaming

Untuk menjalankan job pastikan Spark cluster telah berjalan, setelah itu jalankan perintah berikut.

|  |
| --- |
| $ spark-submit --class me.mamotis.kaspacore.jobs.RawDataStream --master spark://IP\_SERVER\_SPARK:7077 --executor-memory 5G --total-executor-cores 8 hdfs://IP\_SERVER\_HDFS:9000/user/hduser/job/KaspaCore-assembly-0.1.jar & |

Sesuaikan dengan IP dari server, untuk –executor-memory sesuaikan dengan memori RAM yang digunakan tiap workernya, sedangkan –total-executor-cores juga sesuaikan total core CPU yang akan digunakan.

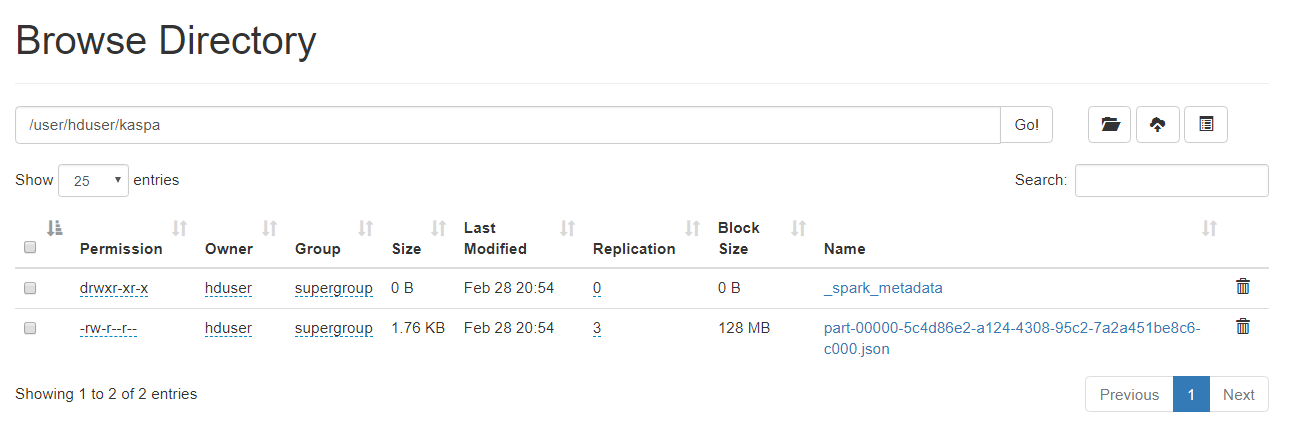


Gambar 3.5.2 Menjalankan job spark

## Mengecek Data

### Pada HDFS

Cek pada web ui hadoop dan buka direktori /user/hduser/kaspa



Gambar 3.6.1 Data yang telah masuk ke HDFS