**Harjoitus 7 Datan tuominen Kafkalta Sparkille**

Big Data -ympäristöt

Tapani Alastalo

M1475

**Sisältö**

[1 Millaista kuvitteellista dataa skripti generoi? 2](#_Toc52899741)

[2 Mitä argumentteja skriptille on mahdollista antaa käynnistyksen yhteydessä? 2](#_Toc52899742)

[3 Miten skriptin ajokomentoa tulee muuttaa jos päivämäärämerkinnäksi halutaan 4.4.1994? 2](#_Toc52899743)

[4 Datan käsittelyväli on 5 sekuntia 2](#_Toc52899744)

[5 Tulosteen muoto on alla esitetyn muotoinen 3](#_Toc52899745)

[6 Lisää sensoridatalle suodatus siten, että kaikki ne anturitiedot, joissa lämpötila-arvo on arvovälin 10-25 ulkopuolella, suodatetaan pois tuloksista 4](#_Toc52899746)

# Millaista kuvitteellista dataa skripti generoi?

Sensoridataa sisältäen aikaleiman, paikkatiedon, lämpötilan ja kosteustiedon.

# Mitä argumentteja skriptille on mahdollista antaa käynnistyksen yhteydessä?

Broker ip:t ja porttitieto pilkulla eroteltuna (--kafka-brokers, -k).

Topic (--kafka-topic, -t), oletuksena sensor\_topic.

Viestitiheys (--rate / -r), eli Kuinka monta viestiä sekunnissa. Oletus 10.

Aloituspäivämäärä (--start-date / -s) muodossa YYY-mm-dd

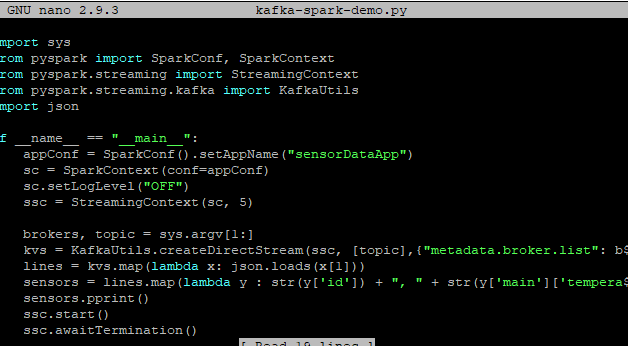
Sensorilukumäärä (--num-sensors, -n). Oletus 10.

# Miten skriptin ajokomentoa tulee muuttaa jos päivämäärämerkinnäksi halutaan 4.4.1994?

python3 sensor-data.py -k koneen-ip-osoite:9092,koneen-ip-osoite:9093,koneen-ip-osoite:9094 -t sensordata -r 1 -n 5 -s 1994-04-04

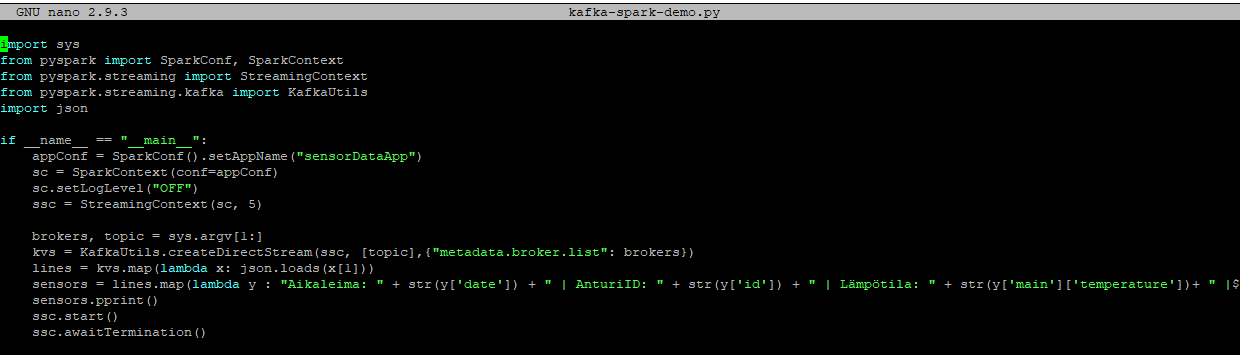
# Datan käsittelyväli on 5 sekuntia

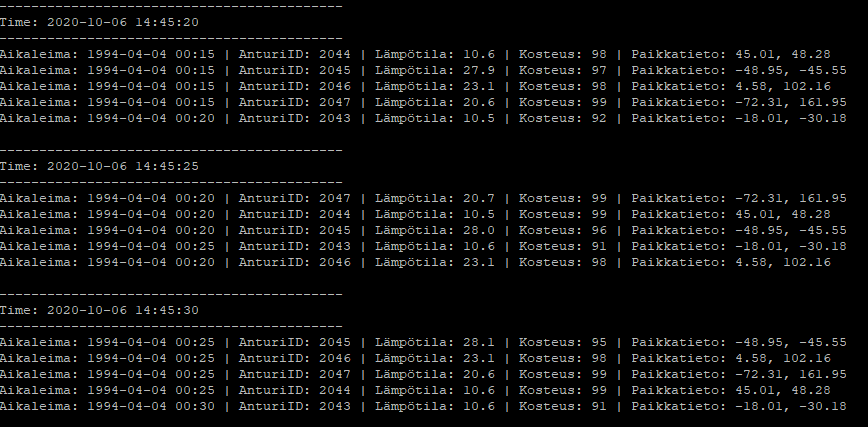
StreamingContext toinen parametri on vaihdettava 10 -> 5.



# Tulosteen muoto on alla esitetyn muotoinen

sensors = lines.map(lambda y : "Aikaleima: " + str(y['date']) + " | AnturiID: " + str(y['id']) + " | Lämpötila: " + str(y['main']['temperature'])+ " | Kosteus: " + str(y['main']['humidity'])+ " | Paikkatieto: " + str(y['coord']['lat'])+ ", " + str(y['coord']['lon']))





# Lisää sensoridatalle suodatus siten, että kaikki ne anturitiedot, joissa lämpötila-arvo on arvovälin 10-25 ulkopuolella, suodatetaan pois tuloksista

lines = kvs.map(lambda x: json.loads(x[1]))

filtered\_lines = lines.filter(lambda y : (y['main']['temperature'] >= 10 and y['main']['temperature'] <= 25))

sensors = filtered\_lines.map(lambda y : str(y['id']) + ", " + str(y['main']['temperature']))

sensors.foreachRDD(lambda x: x.foreach(lambda y: print(y)))

