

---

# Install Kafka and fetch RSVPs

## STEPS

### Install Kafka

1. Find latest version confluent - 5.4.1
2. Find scala version corresponding to it - 2.12
3. `wget http://packages.confluent.io/archive/5.4/confluent-5.4.1-2.12.tar.gz -O /opt/confluent.tar.gz`
4. `tar -xzf confluent.tar.gz`
5. `ln -s confluent-5.4.1 confluent`
6. `cd /opt/confluent`
7. `apt update`
8. `apt install openjdk-11-jdk`
9. Add environment vars

```
echo "CONFLUENT_HOME=/opt/confluent
PATH=$PATH:$CONFLUENT_HOME/bin" >> ~/.bashrc
```

10. Reload .bashrc

```
~/.bashrc
```

11. `confluent local start`

### Stream Meetup RSVPs and events to Kafka topics

This command reads from <http://stream.meetup.com/2/rsvps>, it filters only Yes RSVPs from meetups that have tags mentioned in topics.json. It selects only the following columns `rsvp_id`, `member.member_id` as `member_id`, `mtime`, `response`, `guests`, `event.event_id` as `event_id`, `group.group_urlname` as `group_urlname`, `group.group_topics` as `group_topics`. It saves this data in the `rsvps_filtered_stream` Kafka topic.

---

Then, for each RSVP, if the event\_id does NOT exist in the events.idx file, it reads all event data from [https://api.meetup.com/\[meetup\\_name\]/events/\[event\\_id\]](https://api.meetup.com/[meetup_name]/events/[event_id]) and saves the data in the events Kafka topic

```
unbuffer curl -s "http://stream.meetup.com/2/rsvps" \
  | jq --unbuffered -c --argfile topics topics.json 'select(.response == "yes" and
any(.group.group_topics[]; . as $topic | any($topics[]; . == $topic.urlkey))) |
{rsvp_id, member_id: .member.member_id, mtime, response, guests, event_id:
.event.event_id, group_urlname: .group.group_urlname, group_topics:
.group.group_topics}' \
  | tee >(kafka-console-producer --broker-list localhost:9092 --topic
rsvps_filtered_stream --sync > /dev/null 2>&1) \
  | while read -r rsvp; do echo $rsvp | jq --unbuffered -r
'"\"(.group_urlname)\\n\"(.event_id)'" \
  | xargs -n2 -L2 -d'\n' bash -c 'if ! grep -qFx $1 ./events.idx; then echo $1 >>
events.idx && kafka-console-producer --broker-list localhost:9092 --topic events
--sync <<(curl -s "https://api.meetup.com/$0/events/$1") > /dev/null 2>&1; fi'; done
```

Content of ~/topics.json

Filters applied (tags) - we only monitor meetups data for these tags.

```
[
  "big-data",
  "cloud-computing",
  "opensource",
  "newtech",
  "hadoop",
  "apache-kafka",
  "machine-learning",
  "data-analytics",
  "apache-spark",
  "data-visualization",
  "big-data-analytics",
  "data-mining",
  "computer-programming"
]
```