

# Big Data / Lab05

TEAM KAPPA

WEIDELE, PRÜLL, TOMONDY, BAUER

## HDInsight cluster

### HDInsight-Cluster erstellen ...

#### Grundlagen

[Speicher](#)[Sicherheit + Netzwerkbetrieb](#)[Konfiguration + Preise](#)[Tags](#)[Bewerten + erstellen](#)

Neu bei HDInsight? Starten Sie mit unseren [Schulungsressourcen](#).

Erstellen Sie einen verwalteten HDInsight-Cluster. Treffen Sie Ihre Auswahl zwischen Spark, Kafka, Hadoop, Storm und vielen anderen. [Weitere Informationen](#)

#### Projektdetails

Wählen Sie das Abonnement aus, um bereitgestellte Ressourcen und Kosten zu verwalten. Verwenden Sie Ressourcengruppen wie z. B. Ordner zum Organisieren und Verwalten all Ihrer Ressourcen.

Abonnement \*

Azure für Bildungseinrichtungen



Ressourcengruppe \*

Meine-Ressourcengruppe



[Neues Element erstellen](#)

#### Clusterdetails

Benennen Sie Ihren Cluster, wählen Sie eine Region sowie einen Clustertyp und eine Clusterversion aus. [Weitere Informationen](#)

Clustername \*

hdikafka



Region \*

Deutschland, Westen-Mitte



Clustertyp \*

**Kafka**

[Änderung](#)



Version \*

Kafka 2.1.1 (HDI 4.0)



#### Clusteranmeldeinformationen

Geben Sie neue Anmeldeinformationen ein, die zur Verwaltung oder für den Zugriff auf den Cluster verwendet werden.

Benutzername für Clusteranmeldung \* ⓘ

admin

Kennwort für Clusteranmeldung \*

.....



Clusteranmeldekennwort bestätigen \*

.....



SSH-Benutzername (Secure Shell) \* ⓘ

sshuser

Verwenden Sie ein  
Clusteranmeldekennwort für SSH.



Wählen oder erstellen Sie Speicherkonten, die für die Protokolle, die Auftragseingabe und die Auftragsausgabe des Clusters verwendet werden. Konfigurieren Sie bei Bedarf den Zugriff des Clusters auf diese Konten.

### Primärer Speicher

Wählen oder erstellen Sie ein Speicherkonto, das als Standardspeicherort für Clusterprotokolle und weitere Ausgabe verwendet wird.

Primärer Speichertyp *	<div>Azure Storage</div>
Auswahlmethode * ⓘ	<input checked="" type="radio"/> Aus Liste auswählen <input type="radio"/> Zugriffsschlüssel verwenden
Primäres Speicherkonto *	<div>(Neu) myhdikafkastorage</div> <div><a href="#">Neu erstellen</a></div>
Container * ⓘ	<div>hdikafka-2021-04-11t14-30-09-573z</div>

### Data Lake Storage Gen1

Geben Sie für den Cluster Details für den Zugriff auf Data Lake Storage Gen1 an. Der Cluster kann auf alle Data Lake Storage Gen1-Konten zugreifen, auf die der ausgewählte Dienstprinzipal zugreifen kann.

Data Lake Storage Gen1-Zugriff [Zugriffseinstellungen konfigurieren](#)

### Zusätzliche Azure Storage-Instanzen

Verknüpfen Sie zusätzliche Azure Storage-Konten mit dem Cluster.

Kontoname

[Azure Storage hinzufügen](#)

### Benutzerdefinierte Ambari-Datenbank

Verwenden Sie eine externe Ambari-Datenbank für mehr Flexibilität und erweiterte Steuerungs- und Anpassungsfunktionen. [Weitere Informationen](#)

SQL-Datenbank für Ambari ⓘ

### Externe Metadatenpeicher

Um Ihre Hive- und Oozie-Metadaten außerhalb dieses Cluster zu speichern, wählen Sie eine SQL-Datenbank aus. [Weitere Informationen](#)

SQL-Datenbank für Hive ⓘ

SQL-Datenbank für Oozie ⓘ

Hiermit werden die Sicherheits- und Netzwerkeinstellungen für Ihren Cluster konfiguriert.

### Enterprise-Sicherheitspaket

Verbinden Sie diesen Cluster mit Active Directory Domain Services (AAD DS), um besser steuern zu können, wer auf den Cluster zugreifen kann. [Weitere Informationen](#)

☐ Enterprise-Sicherheitspaket aktivieren ⓘ (Fügt 0.00885465 EUR pro Kernstunde hinzu.)

### TLS

Wählen Sie die für Ihren Cluster unterstützte TLS-Mindestversion aus. [Weitere Informationen](#)


TLS-Mindestversion ⓘ

### Netzwerkeinstellungen

Ressourcenanbieterverbindung ⓘ

Hiermit wird dieser Cluster mit einem virtuellen Netzwerk verbunden. [Weitere Informationen](#)

Virtuelles Netzwerk ⓘ

 Die Auswahl eines virtuellen Netzwerks wird dringend empfohlen. Durch das Erstellen eines Kafka-Clusters ohne virtuelles Netzwerk wird der Cluster möglicherweise unbrauchbar, weil nur vom virtuellen Netzwerk aus zugängliche Clients mit den einzelnen Knoten des Clusters kommunizieren können.

### Verschlüsselung während der Übertragung

Konfigurieren Sie die Einstellungen für die Verschlüsselung während der Übertragung. [Weitere Informationen](#)

☐ Verschlüsselung während der Übertragung aktivieren ⓘ

### Verschlüsselung im Ruhezustand

Einstellungen für Datenträgerverschlüsselung konfigurieren [Weitere Informationen](#)

☐ Eigenen Schlüssel aus Key Vault angeben ⓘ

☐ Verschlüsselung auf dem Host für temporären Datenträger aktivieren ⓘ

### Kafka-REST-Proxy

Fügen Sie Ihrem Cluster den Kafka-REST-Proxy hinzu.

☐ Kafka-REST-Proxy aktivieren [Weitere Informationen](#)

### Identität

Wählen Sie eine benutzerseitig zugewiesene Dienstidentität aus, die Ihren Cluster für das Enterprise-Sicherheitspaket oder die Datenträgerverschlüsselung repräsentiert. [Weitere Informationen](#)

Benutzerseitig zugewiesene verwaltete Identität ⓘ

Konfigurieren Sie Clusterleistung und Preise. [Weitere Informationen](#)

### Knotenkonfiguration

Konfigurieren Sie Größe und Leistung Ihres Clusters, und zeigen Sie Informationen zu den geschätzten Kosten an.

Die Kostenschätzung in der Tabelle berücksichtigt keine Abbonementrabatte oder Kosten in Zusammenhang mit Speicher, Netzwerk oder Datenübertragung.

**i** Diese Konfiguration verwendet 32 von 40 verfügbaren Kernen in der Region "Deutschland, Westen-  
Nutzung von Kernen anzeigen

**+ Anwendung hinzufügen**

Knotentyp	Knotengröße	Knotenanzahl	Geschätzte Koste...
Hauptknoten-K...	E4 V3 (4 Kerne, 32 GB RAM), 0.32 EUR/Stunde ▾	2	0.64 EUR
Zookeeper-Knot...	A4 v2 (4 Kerne, 8 GB RAM), 0.21 EUR/Stunde ▾	3	0.64 EUR
Standard Datent...	S30 (1 TB pro Datenträger), 0.05 EUR/Stunde	<input type="text" value="1"/> ✓	0.14 EUR
Worker-Knoten	E4 V3 (4 Kerne, 32 GB RAM), 0.32 EUR/Stunde ▾	<input type="text" value="3"/> ✓	0.96 EUR

**Geschätzte Gesamtkosten/Stunde 2.38 EUR**

### Skriptaktionen

Verwenden Sie Skriptaktionen zum Ausführen benutzerdefinierter PowerShell- oder Bash-Skripts auf Clusterknoten während der Clusterbereitstellung. [Informationen zu Skriptaktionen](#)

**+ Skriptaktion hinzufügen**

# HDInsight-Cluster erstellen ...

✓ Überprüfung erfolgreich.

Grundlagen Speicher Sicherheit + Netzwerkbetrieb Konfiguration + Preise Tags Bewerten + erstellen

Kafka 2.1.1 (HDI 4.0)

**2.38 EUR Geschätzte Gesamtkosten/Stunde**

Diese Schätzung berücksichtigt keine Abbonementrabatte oder Kosten im Zusammenhang mit Speicher, Netzwerk oder Datenübertragung.

## Grundlagen

Abonnement	Azure für Bildungseinrichtungen
Ressourcengruppe	Meine-Ressourcengruppe
Region	Deutschland, Westen-Mitte
Clustername	(neu) hdikafka
Clustertyp	Kafka 2.1.1 (HDI 4.0)
Benutzername für Clusteranmeldung	admin
SSH-Benutzername (Secure Shell)	sshuser
Verwenden Sie ein Clusteranmeldekennwort für SSH.	Aktiviert

## Sicherheit + Netzwerkbetrieb

TLS-Mindestversion	1.2
Ressourcenanbieterverbindung	Inbound
Verschlüsselung im Ruhezustand	Deaktiviert
Verschlüsselung während der Übertragung	Deaktiviert
Verschlüsselung auf dem Host für temporären Datenträger	Deaktiviert
Kafka-REST-Proxy	Deaktiviert

## Speicher

Primärer Speichertyp	Azure Storage
Primäres Speicherkonto	(neu) myhdikafkastorage
Container	hdikafka-2021-04-11t14-30-09-573z
Zusätzliche Azure Storage-Instanzen	Keine
Data Lake Storage Gen1-Zugriff	Deaktiviert

## Clusterkonfiguration

Hauptknoten	2 Knoten, E4 V3 (4 Kerne, 32 GB RAM)
Zookeeper	3 Knoten, A4 v2 (4 Kerne, 8 GB RAM)
Datenträger	3 Knoten, S30
Worker	3 Knoten, E4 V3 (4 Kerne, 32 GB RAM)

Erstellen

« Zurück

Weiter

[Vorlage zur Automatisierung herunterladen](#)

## Install jq

```
sshuser@hn0-hdikaf:~$ sudo apt -y install jq
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  libopts25
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  libonig2
The following NEW packages will be installed:
  jq libonig2
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 231 kB of archives.
After this operation, 797 kB of additional disk space will be used.
Get:1 http://azure.archive.ubuntu.com/ubuntu xenial-updates/universe amd64 libonig2 amd64 5.9.6-1ubuntu0.1 [86.7 kB]
Get:2 http://azure.archive.ubuntu.com/ubuntu xenial-updates/universe amd64 jq amd64 1.5+dfsg-1ubuntu0.1 [144 kB]
Fetched 231 kB in 0s (1,641 kB/s)
Selecting previously unselected package libonig2:amd64.
(Reading database ... 148314 files and directories currently installed.)
Preparing to unpack .../libonig2_5.9.6-1ubuntu0.1_amd64.deb ...
Unpacking libonig2:amd64 (5.9.6-1ubuntu0.1) ...
Selecting previously unselected package jq.
Preparing to unpack .../jq_1.5+dfsg-1ubuntu0.1_amd64.deb ...
Unpacking jq (1.5+dfsg-1ubuntu0.1) ...
Processing triggers for libc-bin (2.23-0ubuntu11.2) ...
Processing triggers for man-db (2.7.5-1) ...

Progress: [ 54%] [#####.....]
```

## Set up password variable

```
sshuser@hn0-hdikaf:~$ export password='PA55W0RD'
```

## Explore the cluster

```
sshuser@hn0-hdikaf:~$ export clusterName=$(curl -u admin:$password -sS -G "http://headnodehost:8080/api/v1/clusters" |
jq -r '.items[].Clusters.hdikafka')
sshuser@hn0-hdikaf:~$ export KAFKAZKHOSTS=$(curl -sS -u admin:$password -G https://$clusterName.azurehdinsight.net/api
/v1/clusters/$clusterName/services/ZOOKEEPER/components/ZOOKEEPER_SERVER | jq -r '["\(.host_components[]).HostRoles.hos
t_name):2181"] | join(",")' | cut -d', ' -f1,2);
curl: (6) Could not resolve host: null.azurehdinsight.net
sshuser@hn0-hdikaf:~$ export clusterName=$(curl -u admin:$password -sS -G "http://headnodehost:8080/api/v1/clusters" |
jq -r '.items[].Clusters.cluster_name')
sshuser@hn0-hdikaf:~$ export KAFKAZKHOSTS=$(curl -sS -u admin:$password -G https://$clusterName.azurehdinsight.net/api
/v1/clusters/$clusterName/services/ZOOKEEPER/components/ZOOKEEPER_SERVER | jq -r '["\(.host_components[]).HostRoles.hos
t_name):2181"] | join(",")' | cut -d', ' -f1,2);
sshuser@hn0-hdikaf:~$ echo $KAFKAZKHOSTS
zk0-hdikaf.mvvuoqfaoyrejkjdjii4ij2lsg.frax.internal.cloudapp.net:2181,zk1-hdikaf.mvvuoqfaoyrejkjdjii4ij2lsg.frax.inter
nal.cloudapp.net:2181
sshuser@hn0-hdikaf:~$ export KAFKABROKERS=$(curl -sS -u admin:$password -G https://$clusterName.azurehdinsight.net/api
/v1/clusters/$clusterName/services/KAFKA/components/KAFKA_BROKER | jq -r '["\(.host_components[]).HostRoles.host_name):
9092"] | join(",")' | cut -d', ' -f1,2);
sshuser@hn0-hdikaf:~$ echo $KAFKABROKERS
wn0-hdikaf.mvvuoqfaoyrejkjdjii4ij2lsg.frax.internal.cloudapp.net:9092,wn1-hdikaf.mvvuoqfaoyrejkjdjii4ij2lsg.frax.inter
nal.cloudapp.net:9092
sshuser@hn0-hdikaf:~$ /usr/hdp/current/kafka-broker/bin/kafka-topics.sh --create --replication-factor 3 --partitions 8
--topic test --zookeeper $KAFKAZKHOSTS
Created topic "test".
sshuser@hn0-hdikaf:~$ /usr/hdp/current/kafka-broker/bin/kafka-topics.sh --list --zookeeper $KAFKAZKHOSTS
test
sshuser@hn0-hdikaf:~$ /usr/hdp/current/kafka-broker/bin/kafka-topics.sh --delete --topic test --zookeeper $KAFKAZKHOST
S
Topic test is marked for deletion.
Note: This will have no impact if delete.topic.enable is not set to true.
sshuser@hn0-hdikaf:~$ /usr/hdp/current/kafka-broker/bin/kafka-topics.sh --list --zookeeper $KAFKAZKHOSTS
sshuser@hn0-hdikaf:~$ /usr/hdp/current/kafka-broker/bin/kafka-topics.sh --create --replication-factor 3 --partitions 8
--topic test --zookeeper $KAFKAZKHOSTS
Created topic "test".
sshuser@hn0-hdikaf:~$ /usr/hdp/current/kafka-broker/bin/kafka-topics.sh --list --zookeeper $KAFKAZKHOSTS
test
sshuser@hn0-hdikaf:~$ /usr/hdp/current/kafka-broker/bin/kafka-topics.sh
Create, delete, describe, or change a topic.
Option                                Description
-----
--alter                                Alter the number of partitions,
                                        replica assignment, and/or
                                        configuration for the topic.
--config <String: name=value>         A topic configuration override for the
```

## Produce records to topic „test”

```
sshuser@hn0-hdikaf:~$ /usr/hdp/current/kafka-broker/bin/kafka-console-producer.sh --broker-list $KAFKABROKERS --topic test
>first message
>second message to topic test
>third message to topic test
>last message before quit with ctrl+c ;)
^Csshuser@hn0-hdikaf:~$
sshuser@hn0-hdikaf:~$
```

## Consume records from topic „test”

```
sshuser@hn0-hdikaf:~$
sshuser@hn0-hdikaf:~$ /usr/hdp/current/kafka-broker/bin/kafka-console-consumer.sh --bootstrap-server $KAFKABROKERS --topic test --from-beginning
second message to topic test
last message before quit with ctrl+c ;)
third message to topic test
first message

^CProcessed a total of 4 messages
sshuser@hn0-hdikaf:~$
```



## Create topic "bestellungen"

```
sshuser@hn0-hdikaf:~$  
sshuser@hn0-hdikaf:~$ /usr/hdp/current/kafka-broker/bin/kafka-topics.sh --create --replication-factor 3 --partitions 8 --topic bestellungen --zookeeper $KAFKA_ZKHOSTS  
Created topic "bestellungen".  
sshuser@hn0-hdikaf:~$
```

## Python script for sending data to Kafka cluster

```
send_to_kafka.py > ...  
1  from kafka import KafkaProducer  
2  import time  
3  import os  
4  # Replace the `ip_address` entries with the IP address of your worker nodes  
5  # NOTE: you don't need the full list of worker nodes, just one or two.  
6  producer = KafkaProducer(  
7      bootstrap_servers=['wn0-hdikaf.mdyd0ctxohzupps0jl0ruvnnxe.frax.internal.cloudapp.net:9092'])  
8  file = "lab05data.txt"  
9  print("Working on file: " + file)  
10 file = open(os.path.join(file))  
11 while True:  
12     line = file.readline().rstrip()  
13     if not line:  
14         break  
15     else:  
16         print(line)  
17         producer.send('bestellungen', line.encode())  
18         time.sleep(0.001)  
19 file.close()  
20 print("end")  
21
```

```
sshuser@hn0-hdikaf:~/python_scripts$ sshuser@hn0-hdikaf:~/python_scripts$ python3 send_to_kafka.py  
Working on file: lab05data.txt  
1277134,1,2003-12-02  
2435457,2,2005-06-01  
2338545,3,2001-02-17  
2218269,1,2002-12-27  
441153,4,2002-10-11  
1921624,2,2005-08-31  
2096652,3,2004-05-31  
818736,2,2004-02-17  
284560,3,2003-07-27  
1211224,5,2004-05-08  
1984086,1,2004-09-16  
1389539,3,2005-06-07  
454575,2,2004-10-23  
149028,3,2003-06-19  
1105843,2,2005-04-07  
353813,2,2004-08-05  
903779,3,2001-05-21  
639194,2,2003-10-19  
308031,4,2000-09-07  
1794933,1,2003-02-14  
625085,3,2004-05-21  
893673,2,2000-12-04  
1704675,2,2001-04-10  
2116614,3,2002-07-17
```

## Python script for consuming data from Kafka cluster

```
recv_from_kafka.py > ...
1  from kafka import KafkaConsumer
2  # Replace the `ip_address` entries with the IP address of your worker nodes
3  # Again, you only need one or two, not the full list.
4  # Note: auto_offset_reset='earliest' resets the starting offset to the beginning
5  #       of the topic
6  consumer = KafkaConsumer(
7      | bootstrap_servers=['wn0-hdikaf.mdyd0ctxohzupps0jl0ruvnxn.frax.internal.cloudapp.net:9092'], auto_offset_reset='earliest')
8  consumer.subscribe(['bestellungen'])
9  while True:
10     for msg in consumer:
11         print(msg)
12     print("end")
13
```

```
ConsumerRecord(topic='bestellungen', partition=6, offset=127, timestamp=1618668373966, timestamp_type=0, key=None, value=b'403118,3,2005-11-04', headers=[], checksum=None, serialized_key_size=-1, serialized_value_size=19, serialized_header_size=-1)
ConsumerRecord(topic='bestellungen', partition=5, offset=133, timestamp=1618668373969, timestamp_type=0, key=None, value=b'1290587,4,2002-03-20', headers=[], checksum=None, serialized_key_size=-1, serialized_value_size=20, serialized_header_size=-1)
ConsumerRecord(topic='bestellungen', partition=0, offset=147, timestamp=1618668373973, timestamp_type=0, key=None, value=b'1988192,1,2004-06-07', headers=[], checksum=None, serialized_key_size=-1, serialized_value_size=20, serialized_header_size=-1)
ConsumerRecord(topic='bestellungen', partition=4, offset=149, timestamp=1618668373976, timestamp_type=0, key=None, value=b'865516,1,2005-07-09', headers=[], checksum=None, serialized_key_size=-1, serialized_value_size=19, serialized_header_size=-1)
ConsumerRecord(topic='bestellungen', partition=0, offset=148, timestamp=1618668373975, timestamp_type=0, key=None, value=b'819317,3,2005-04-04', headers=[], checksum=None, serialized_key_size=-1, serialized_value_size=19, serialized_header_size=-1)
ConsumerRecord(topic='bestellungen', partition=4, offset=150, timestamp=1618668373983, timestamp_type=0, key=None, value=b'326668,2,2001-01-13', headers=[], checksum=None, serialized_key_size=-1, serialized_value_size=19, serialized_header_size=-1)
ConsumerRecord(topic='bestellungen', partition=3, offset=154, timestamp=1618668373984, timestamp_type=0, key=None, value=b'418066,2,2005-05-09', headers=[], checksum=None, serialized_key_size=-1, serialized_value_size=19, serialized_header_size=-1)
ConsumerRecord(topic='bestellungen', partition=6, offset=128, timestamp=1618668373985, timestamp_type=0, key=None, value=b'1100008,1,2004-04-03', headers=[], checksum=None, serialized_key_size=-1, serialized_value_size=20, serialized_header_size=-1)
ConsumerRecord(topic='bestellungen', partition=6, offset=129, timestamp=1618668373988, timestamp_type=0, key=None, value=b'2374451,2,2005-05-20', headers=[], checksum=None, serialized_key_size=-1, serialized_value_size=20, serialized_header_size=-1)
ConsumerRecord(topic='bestellungen', partition=1, offset=140, timestamp=1618668373987, timestamp_type=0, key=None, value=b'138549,3,2005-06-27', headers=[], checksum=None, serialized_key_size=-1, serialized_value_size=19, serialized_header_size=-1)
ConsumerRecord(topic='bestellungen', partition=5, offset=134, timestamp=1618668373997, timestamp_type=0, key=None, value=b'2118903,1,2003-07-12', headers=[], checksum=None, serialized_key_size=-1, serialized_value_size=20, serialized_header_size=-1)
ConsumerRecord(topic='bestellungen', partition=6, offset=130, timestamp=1618668373995, timestamp_type=0, key=None, value=b'1414512,3,2002-12-19', headers=[], checksum=None, serialized_key_size=-1, serialized_value_size=20, serialized_header_size=-1)
ConsumerRecord(topic='bestellungen', partition=3, offset=155, timestamp=1618668373996, timestamp_type=0, key=None, value=b'2025567,2,2002-07-20', headers=[], checksum=None, serialized_key_size=-1, serialized_value_size=20, serialized_header_size=-1)
ConsumerRecord(topic='bestellungen', partition=4, offset=151, timestamp=1618668374004, timestamp_type=0, key=None, value=b'47412820,5,2002-04-07', headers=[], checksum=None, serialized_key_size=-1, serialized_value_size=20, serialized_header_size=-1)
```

## Bonus Point

Die Retention-Time wurde temporär vom Default-Value (168 Stunden) auf 1000 ms reduziert

- ➔ alle alten Messages sind weg
- ➔ danach Retention-Time wieder erhöht

```
sshuser@hnn0-hdikaf:~$ /usr/hdp/current/kafka-broker/bin/kafka-topics.sh --zookeeper zk0-hdikaf.mdyd0ctxohzupps0jl0ruvnnxe.frax.internal.cloudapp.net:2181 --alter --topic bestellungen --config retention.ms=1000
WARNING: Altering topic configuration from this script has been deprecated and may be removed in future releases.
        Going forward, please use kafka-configs.sh for this functionality
Updated config for topic "bestellungen".
sshuser@hnn0-hdikaf:~$ python3 recv_from_kafka.py
python3: can't open file 'recv_from_kafka.py': [Errno 2] No such file or directory
sshuser@hnn0-hdikaf:~$ cd python_scripts/
sshuser@hnn0-hdikaf:~/python_scripts$ python3 recv_from_kafka.py
^CTraceback (most recent call last):
  File "recv_from_kafka.py", line 10, in <module>
    for msg in consumer:
  File "/home/sshuser/.local/lib/python3.5/site-packages/kafka/consumer/group.py", line 1193, in __next__
    return self.next_v2()
  File "/home/sshuser/.local/lib/python3.5/site-packages/kafka/consumer/group.py", line 1201, in next_v2
    return next(self._iterator)
  File "/home/sshuser/.local/lib/python3.5/site-packages/kafka/consumer/group.py", line 1116, in _message_generator_v2
    record_map = self.poll(timeout_ms=timeout_ms, update_offsets=False)
  File "/home/sshuser/.local/lib/python3.5/site-packages/kafka/consumer/group.py", line 655, in poll
    records = self._poll_once(remaining, max_records, update_offsets=update_offsets)
  File "/home/sshuser/.local/lib/python3.5/site-packages/kafka/consumer/group.py", line 702, in _poll_once
    self._client.poll(timeout_ms=timeout_ms)
  File "/home/sshuser/.local/lib/python3.5/site-packages/kafka/client_async.py", line 602, in poll
    self._poll(timeout / 1000)
  File "/home/sshuser/.local/lib/python3.5/site-packages/kafka/client_async.py", line 634, in _poll
    ready = self._selector.select(timeout)
  File "/usr/lib/python3.5/selectors.py", line 441, in select
    fd_event_list = self._epoll.poll(timeout, max_ev)
KeyboardInterrupt
sshuser@hnn0-hdikaf:~/python_scripts$ /usr/hdp/current/kafka-broker/bin/kafka-topics.sh --zookeeper zk0-hdikaf.mdyd0ctxohzupps0jl0ruvnnxe.frax.int
ernal.cloudapp.net:2181 --alter --topic bestellungen --config retention.ms=604800000
WARNING: Altering topic configuration from this script has been deprecated and may be removed in future releases.
        Going forward, please use kafka-configs.sh for this functionality
Updated config for topic "bestellungen".
sshuser@hnn0-hdikaf:~/python_scripts$
```