Background

Currently, many organization or companies use kafka message as a tool for the communication between different machine or virtual instance deployed on the cloud. Receiving special Kafka will trigger the data pipeline to process the data properly.

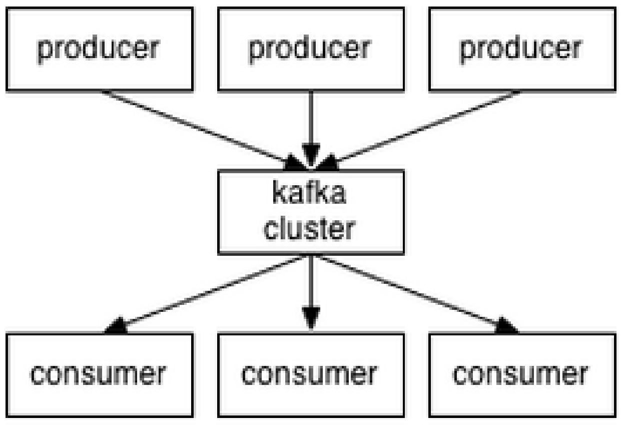
According to the official website of kafka organization, Kafka is a distributed, partitioned, replicated commit log service. It provides the functionality of a messaging system, but with a unique design.

Basic knowledge:

- Kafka maintains feeds of messages in categories called topics.
- We'll call processes that publish messages to a Kafka topic producers.
- We'll call processes that subscribe to topics and process the feed of published messages consumers..
- Kafka is run as a cluster comprised of one or more servers each of which is called a broker.

Architecture

A stream of messages of a particular type is defined as a **topic**. A Message is defined as a payload of bytes and a Topic is a category or feed name to which messages are published.
A **Producer** can be anyone who can publish messages to a Topic.
The published messages are then stored at a set of servers called **Brokers** or **Kafka Cluster**.
A **Consumer** can subscribe to one or more Topics and consume the published Messages by pulling data from the Brokers.



Communication between the clients and the servers is done with a simple, high-performance, language agnostic TCP protocol. We provide a Java client for Kafka, but clients are available in many languages. For here, we use python client.

Environment Setting

I tested the kafka on the **Ubuntu 12.04 LTS.**

1. install zookeeper

java

We need Java.

Install JRE in Ubuntu

$ sudo apt-get install default-jre

Install OpneJDK

$ sudo apt-get install default-jdk

2. zookeeper

$ wget
http://mirrors.ukfast.co.uk/sites/ftp.apache.org/zookeeper/stable/zookeeper-3.4.6.tar.gz

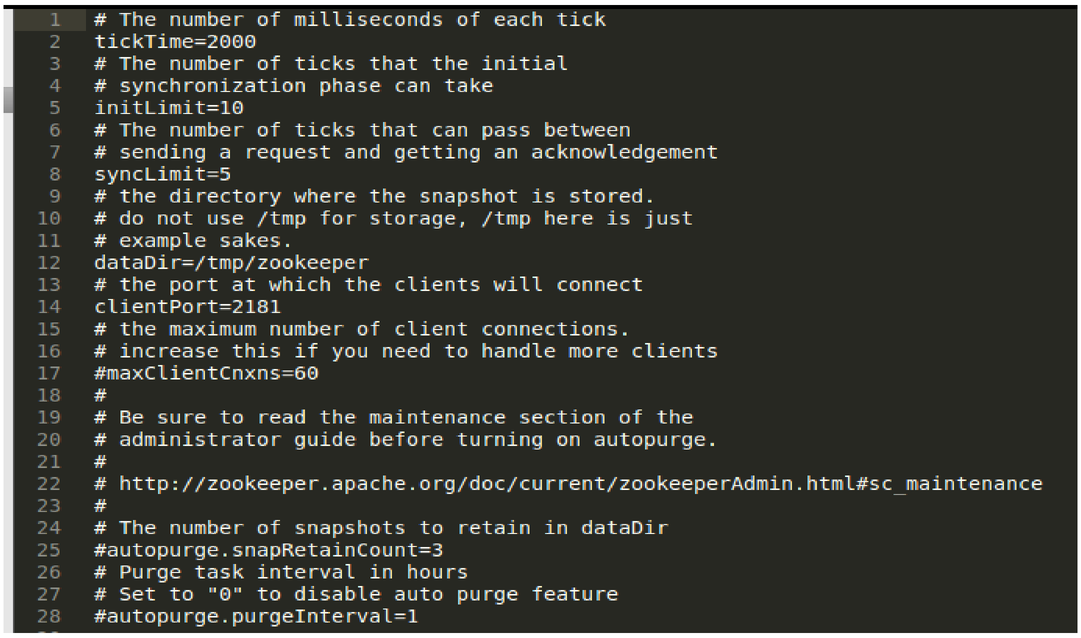
$ tar -xvf zookeeper-3.4.6.tar.gz

Test zookeeper

[1] Configuration for zookeeper, written in zoo.cfg

$ cd zookeeper-3.4.6/

$ cp conf/zoo_sample.cfg conf/zoo.cfg



```
# The number of milliseconds of each tick
tickTime=2000
# The number of ticks that the initial
# synchronization phase can take
initLimit=10
# The number of ticks that can pass between
# sending a request and getting an acknowledgement
syncLimit=5
# the directory where the snapshot is stored.
# do not use /tmp for storage, /tmp here is just
# example sakes.
dataDir=/tmp/zookeeper
# the port at which the clients will connect
clientPort=2181
# the maximum number of client connections.
# increase this if you need to handle more clients
#maxClientCnxns=60
#
# Be sure to read the maintenance section of the
# administrator guide before turning on autopurge.
#
# http://zookeeper.apache.org/doc/current/zookeeperAdmin.html#sc_maintenance
#
# The number of snapshots to retain in dataDir
#autopurge.snapRetainCount=3
# Purge task interval in hours
# Set to "0" to disable auto purge feature
#autopurge.purgeInterval=1
```

As we can see from the configuration file, the default client port is 2181.

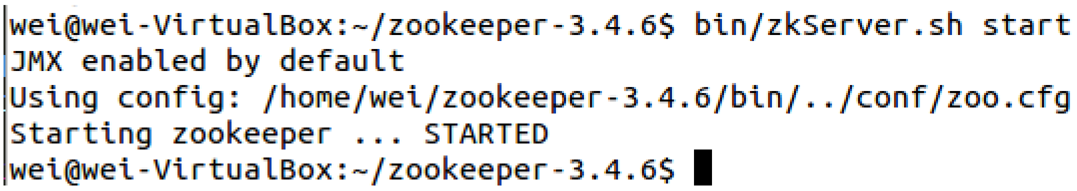
[2] Start zookeeper according to the configuration file

$ bin/zkServer.sh start

JMX enabled by default

Using config: /home/ubuntu/zookeeper-3.4.6/bin/../conf/zoo.cfg

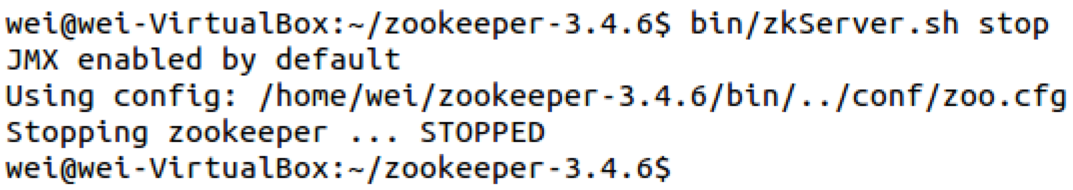
Starting zookeeper ... STARTED



```
wei@wei-VirtualBox:~/zookeeper-3.4.6$ bin/zkServer.sh start
JMX enabled by default
Using config: /home/wei/zookeeper-3.4.6/bin/../conf/zoo.cfg
Starting zookeeper ... STARTED
wei@wei-VirtualBox:~/zookeeper-3.4.6$
```

[3] Stop zookeeper:

$bin/zkServer.sh stop



```
wei@wei-VirtualBox:~/zookeeper-3.4.6$ bin/zkServer.sh stop
JMX enabled by default
Using config: /home/wei/zookeeper-3.4.6/bin/../conf/zoo.cfg
Stopping zookeeper ... STOPPED
wei@wei-VirtualBox:~/zookeeper-3.4.6$
```

As showed above, you can use command line to initial the zookeeper service on local machine, but here we need use python client to implement this service.

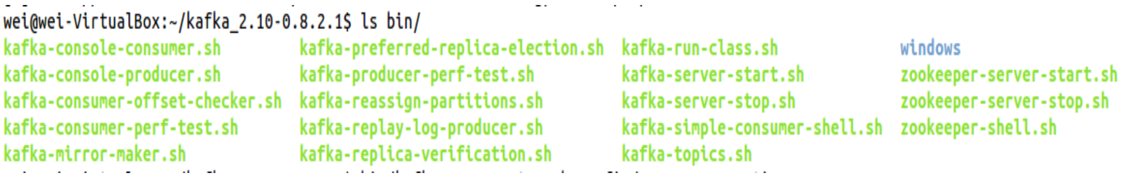
3.kafka python client:

Install python client for kafka:

cd

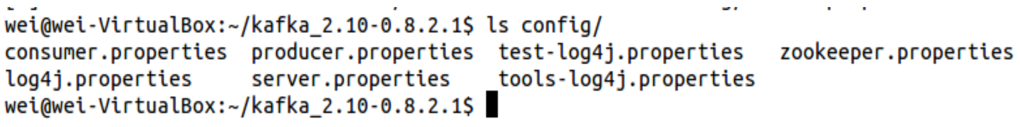
sudo pip install kafka-python

test:All the bin files:



```
wei@wei-VirtualBox:~/kafka_2.10-0.8.2.1$ ls bin/
kafka-console-consumer.sh        kafka-preferred-replica-election.sh  kafka-run-class.sh              windows
kafka-console-producer.sh        kafka-producer-perf-test.sh          kafka-server-start.sh           zookeeper-server-start.sh
kafka-consumer-offset-checker.sh kafka-reassign-partitions.sh         kafka-server-stop.sh            zookeeper-server-stop.sh
kafka-consumer-perf-test.sh      kafka-replay-log-producer.sh         kafka-simple-consumer-shell.sh  zookeeper-shell.sh
kafka-mirror-maker.sh            kafka-replica-verification.sh        kafka-topics.sh
```

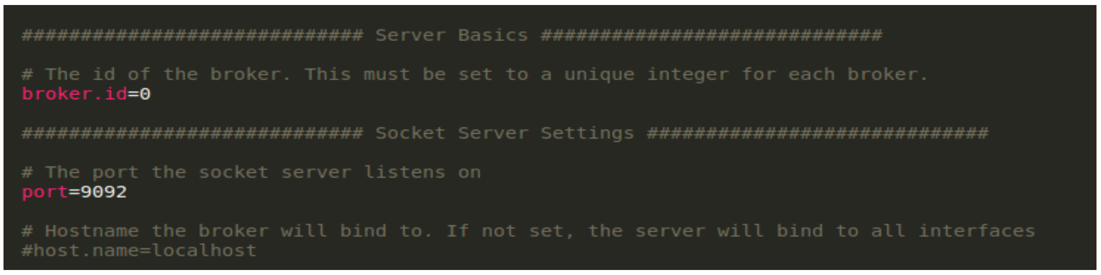
Check configuration file for test:



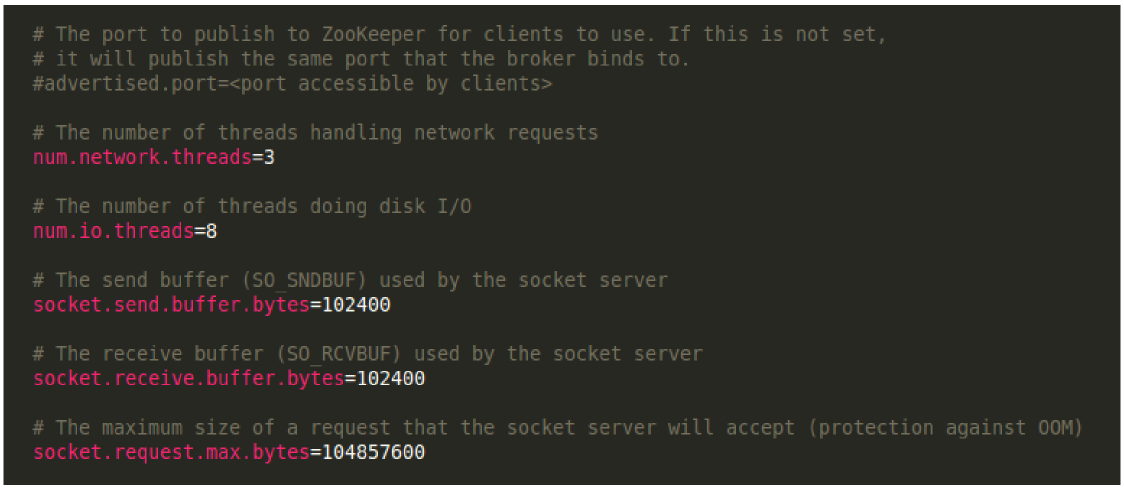
```
wei@wei-VirtualBox:~/kafka_2.10-0.8.2.1$ ls config/
consumer.properties  producer.properties  test-log4j.properties   zookeeper.properties
log4j.properties     server.properties    tools-log4j.properties
wei@wei-VirtualBox:~/kafka_2.10-0.8.2.1$
```

open server.properties to check the default settings

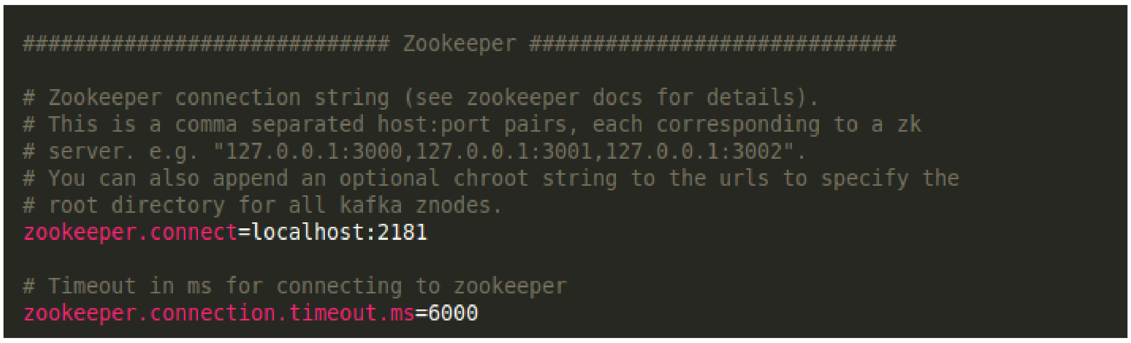
important setting:



```
############################# Server Basics #############################

# The id of the broker. This must be set to a unique integer for each broker.
broker.id=0

############################# Socket Server Settings #############################

# The port the socket server listens on
port=9092

# Hostname the broker will bind to. If not set, the server will bind to all interfaces
#host.name=localhost
```



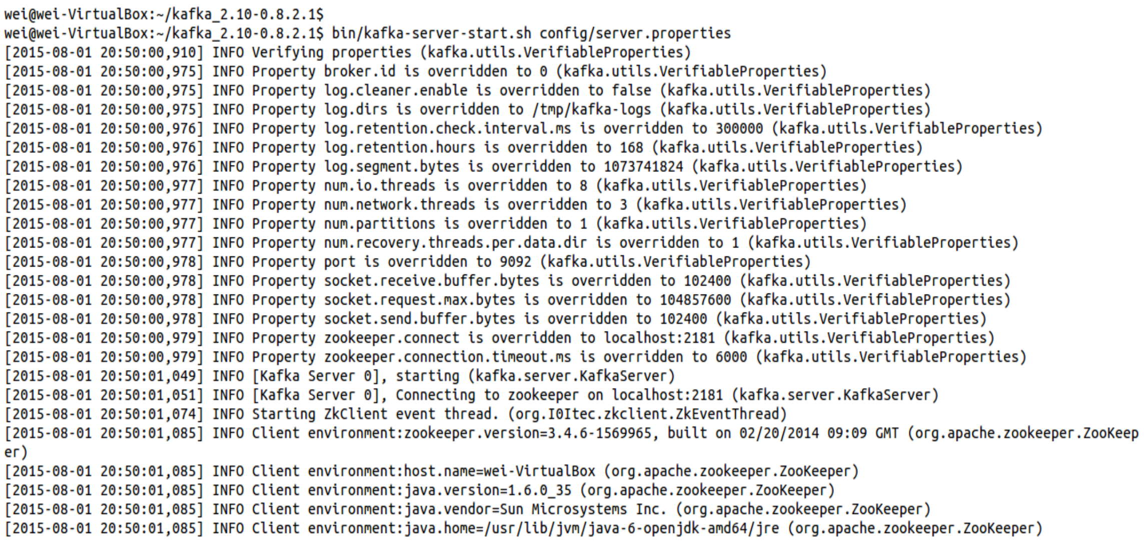
```
# The port to publish to ZooKeeper for clients to use. If this is not set,
# it will publish the same port that the broker binds to.
#advertised.port=<port accessible by clients>

# The number of threads handling network requests
num.network.threads=3

# The number of threads doing disk I/O
num.io.threads=8

# The send buffer (SO_SNDBUF) used by the socket server
socket.send.buffer.bytes=102400

# The receive buffer (SO_RCVBUF) used by the socket server
socket.receive.buffer.bytes=102400

# The maximum size of a request that the socket server will accept (protection against OOM)
socket.request.max.bytes=104857600
```



```
############################# Zookeeper #############################

# Zookeeper connection string (see zookeeper docs for details).
# This is a comma separated host:port pairs, each corresponding to a zk
# server. e.g. "127.0.0.1:3000,127.0.0.1:3001,127.0.0.1:3002".
# You can also append an optional chroot string to the urls to specify the
# root directory for all kafka znodes.
zookeeper.connect=localhost:2181

# Timeout in ms for connecting to zookeeper
zookeeper.connection.timeout.ms=6000
```

Test implementation of kafka service.

$ bin/kafka-server-start.sh config/server.properties



```
wei@wei-VirtualBox:~/kafka_2.10-0.8.2.1$
wei@wei-VirtualBox:~/kafka_2.10-0.8.2.1$ bin/kafka-server-start.sh config/server.properties
[2015-08-01 20:50:00,910] INFO Verifying properties (kafka.utils.VerifiableProperties)
[2015-08-01 20:50:00,975] INFO Property broker.id is overridden to 0 (kafka.utils.VerifiableProperties)
[2015-08-01 20:50:00,975] INFO Property log.cleaner.enable is overridden to false (kafka.utils.VerifiableProperties)
[2015-08-01 20:50:00,975] INFO Property log.dirs is overridden to /tmp/kafka-logs (kafka.utils.VerifiableProperties)
[2015-08-01 20:50:00,976] INFO Property log.retention.check.interval.ms is overridden to 300000 (kafka.utils.VerifiableProperties)
[2015-08-01 20:50:00,976] INFO Property log.retention.hours is overridden to 168 (kafka.utils.VerifiableProperties)
[2015-08-01 20:50:00,976] INFO Property log.segment.bytes is overridden to 1073741824 (kafka.utils.VerifiableProperties)
[2015-08-01 20:50:00,977] INFO Property num.io.threads is overridden to 8 (kafka.utils.VerifiableProperties)
[2015-08-01 20:50:00,977] INFO Property num.network.threads is overridden to 3 (kafka.utils.VerifiableProperties)
[2015-08-01 20:50:00,977] INFO Property num.partitions is overridden to 1 (kafka.utils.VerifiableProperties)
[2015-08-01 20:50:00,977] INFO Property num.recovery.threads.per.data.dir is overridden to 1 (kafka.utils.VerifiableProperties)
[2015-08-01 20:50:00,978] INFO Property port is overridden to 9092 (kafka.utils.VerifiableProperties)
[2015-08-01 20:50:00,978] INFO Property socket.receive.buffer.bytes is overridden to 102400 (kafka.utils.VerifiableProperties)
[2015-08-01 20:50:00,978] INFO Property socket.request.max.bytes is overridden to 104857600 (kafka.utils.VerifiableProperties)
[2015-08-01 20:50:00,978] INFO Property socket.send.buffer.bytes is overridden to 102400 (kafka.utils.VerifiableProperties)
[2015-08-01 20:50:00,979] INFO Property zookeeper.connect is overridden to localhost:2181 (kafka.utils.VerifiableProperties)
[2015-08-01 20:50:00,979] INFO Property zookeeper.connection.timeout.ms is overridden to 6000 (kafka.utils.VerifiableProperties)
[2015-08-01 20:50:01,049] INFO [Kafka Server 0], starting (kafka.server.KafkaServer)
[2015-08-01 20:50:01,051] INFO [Kafka Server 0], Connecting to zookeeper on localhost:2181 (kafka.server.KafkaServer)
[2015-08-01 20:50:01,074] INFO Starting ZkClient event thread. (org.I0Itec.zkclient.ZkEventThread)
[2015-08-01 20:50:01,085] INFO Client environment:zookeeper.version=3.4.6-1569965, built on 02/20/2014 09:09 GMT (org.apache.zookeeper.ZooKeep
er)
[2015-08-01 20:50:01,085] INFO Client environment:host.name=wei-VirtualBox (org.apache.zookeeper.ZooKeeper)
[2015-08-01 20:50:01,085] INFO Client environment:java.version=1.6.0_35 (org.apache.zookeeper.ZooKeeper)
[2015-08-01 20:50:01,085] INFO Client environment:java.vendor=Sun Microsystems Inc. (org.apache.zookeeper.ZooKeeper)
[2015-08-01 20:50:01,085] INFO Client environment:java.home=/usr/lib/jvm/java-6-openjdk-amd64/jre (org.apache.zookeeper.ZooKeeper)
```

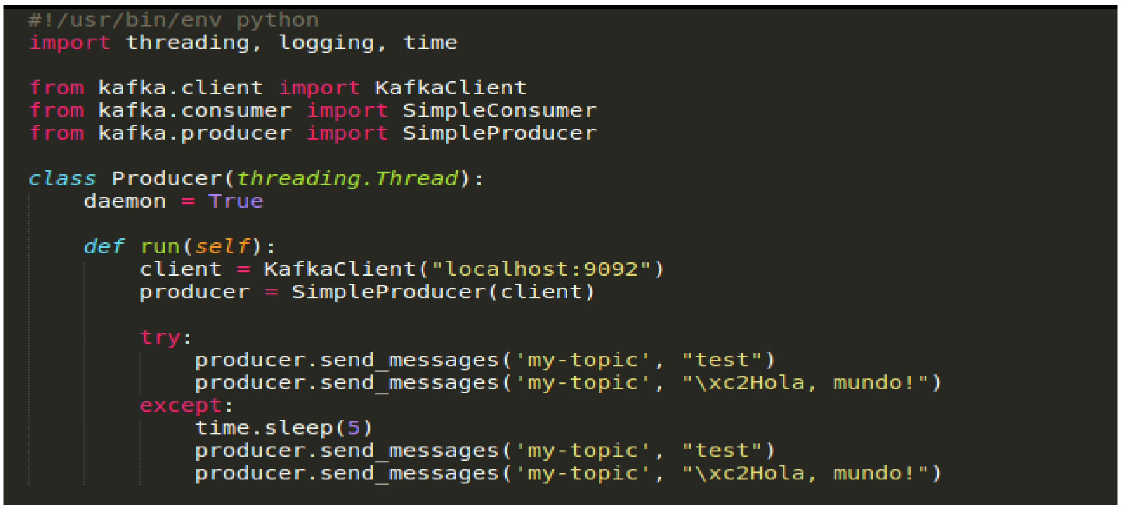
As we can see from the output, the port is verridden to 9092 (kafka.utils.VerifiableProperties), Kafka server 0 connecting to zookeeper on localhost:2181.

Next step is build a server and sending message through it according to the architecture of kafka service.

Work Example

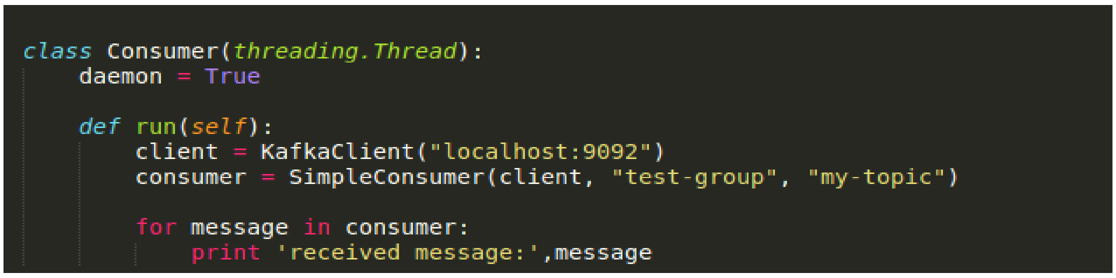
Example code to test the kakfa client.

Producer:



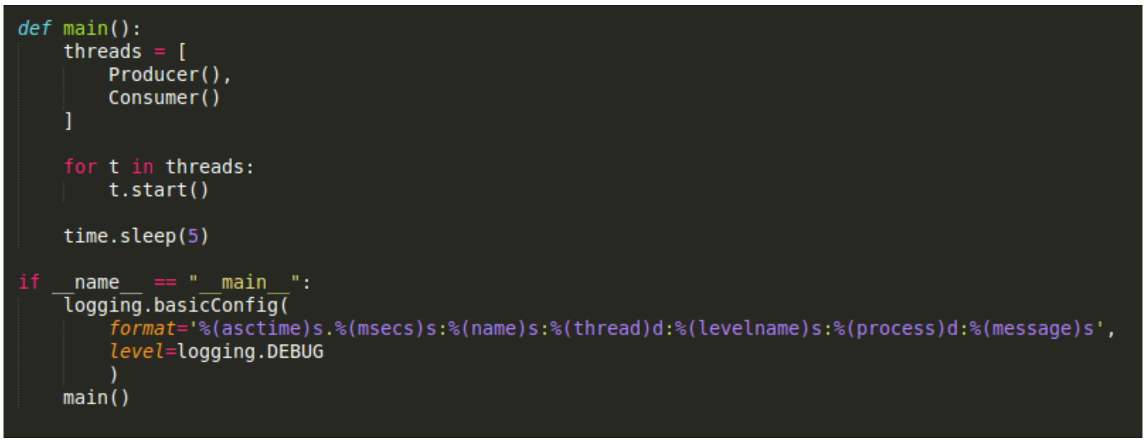
```python
#!/usr/bin/env python
import threading, logging, time

from kafka.client import KafkaClient
from kafka.consumer import SimpleConsumer
from kafka.producer import SimpleProducer

class Producer(threading.Thread):
    daemon = True

    def run(self):
        client = KafkaClient("localhost:9092")
        producer = SimpleProducer(client)

        try:
            producer.send_messages('my-topic', "test")
            producer.send_messages('my-topic', "\xc2Hola, mundo!")
        except:
            time.sleep(5)
            producer.send_messages('my-topic', "test")
            producer.send_messages('my-topic', "\xc2Hola, mundo!")
```

Consumer:



```python
class Consumer(threading.Thread):
    daemon = True

    def run(self):
        client = KafkaClient("localhost:9092")
        consumer = SimpleConsumer(client, "test-group", "my-topic")

        for message in consumer:
            print 'received message:',message
```

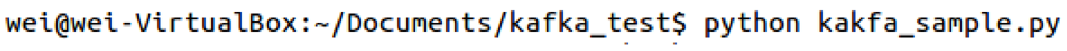
main program:



```python
def main():
    threads = [
        Producer(),
        Consumer()
    ]

    for t in threads:
        t.start()

    time.sleep(5)

if __name__ == "__main__":
    logging.basicConfig(
        format='%(asctime)s.%(msecs)s:%(name)s:%(thread)d:%(levelname)s:%(process)d:%(message)s',
        level=logging.DEBUG
        )
    main()
```

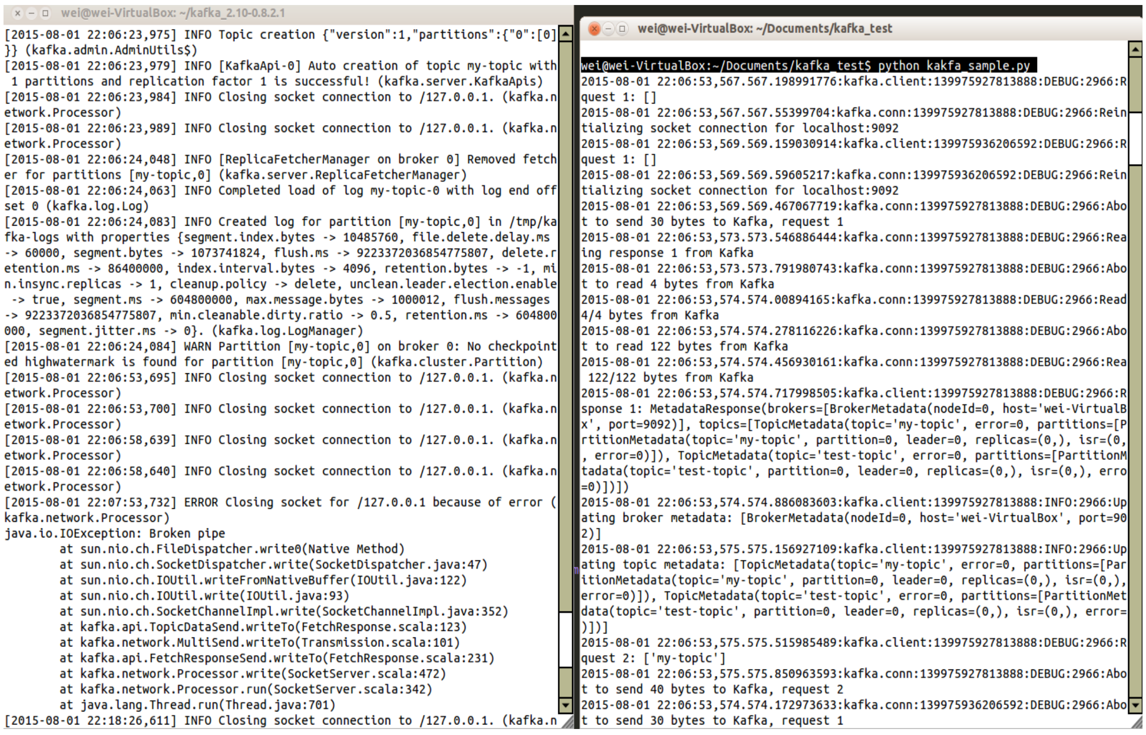
result:

run the program:



```
wei@wei-VirtualBox:~/Documents/kafka_test$ python kakfa_sample.py
```

left part is the output of kafka server, right part is the kafka client which runs the example program.

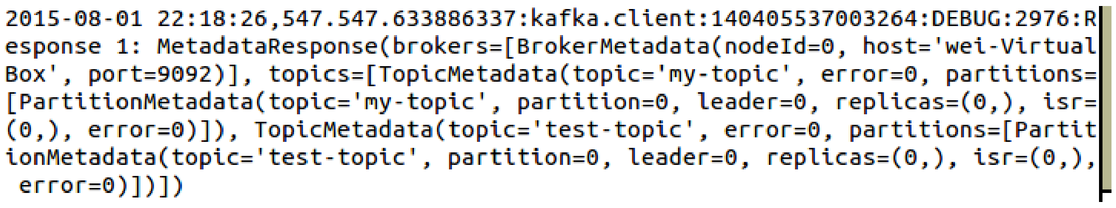


```
[2015-08-01 22:06:23,975] INFO Topic creation {"version":1,"partitions":{"0":[0]
}} (kafka.admin.AdminUtils$)
[2015-08-01 22:06:23,979] INFO [KafkaApi-0] Auto creation of topic my-topic with
 1 partitions and replication factor 1 is successful! (kafka.server.KafkaApis)
[2015-08-01 22:06:23,984] INFO Closing socket connection to /127.0.0.1. (kafka.n
etwork.Processor)
[2015-08-01 22:06:23,989] INFO Closing socket connection to /127.0.0.1. (kafka.n
etwork.Processor)
[2015-08-01 22:06:24,048] INFO [ReplicaFetcherManager on broker 0] Removed fetch
er for partitions [my-topic,0] (kafka.server.ReplicaFetcherManager)
[2015-08-01 22:06:24,063] INFO Completed load of log my-topic-0 with log end off
set 0 (kafka.log.Log)
[2015-08-01 22:06:24,083] INFO Created log for partition [my-topic,0] in /tmp/ka
fka-logs with properties {segment.index.bytes -> 10485760, file.delete.delay.ms
-> 60000, segment.bytes -> 1073741824, flush.ms -> 9223372036854775807, delete.r
etention.ms -> 86400000, index.interval.bytes -> 4096, retention.bytes -> -1, mi
n.insync.replicas -> 1, cleanup.policy -> delete, unclean.leader.election.enable
 -> true, segment.ms -> 604800000, max.message.bytes -> 1000012, flush.messages
-> 9223372036854775807, min.cleanable.dirty.ratio -> 0.5, retention.ms -> 604800
000, segment.jitter.ms -> 0}. (kafka.log.LogManager)
[2015-08-01 22:06:24,084] WARN Partition [my-topic,0] on broker 0: No checkpoint
ed highwatermark is found for partition [my-topic,0] (kafka.cluster.Partition)
[2015-08-01 22:06:53,695] INFO Closing socket connection to /127.0.0.1. (kafka.n
etwork.Processor)
[2015-08-01 22:06:53,700] INFO Closing socket connection to /127.0.0.1. (kafka.n
etwork.Processor)
[2015-08-01 22:06:58,639] INFO Closing socket connection to /127.0.0.1. (kafka.n
etwork.Processor)
[2015-08-01 22:06:58,640] INFO Closing socket connection to /127.0.0.1. (kafka.n
etwork.Processor)
[2015-08-01 22:07:53,732] ERROR Closing socket for /127.0.0.1 because of error (
kafka.network.Processor)
java.io.IOException: Broken pipe
        at sun.nio.ch.FileDispatcher.write0(Native Method)
        at sun.nio.ch.SocketDispatcher.write(SocketDispatcher.java:47)
        at sun.nio.ch.IOUtil.writeFromNativeBuffer(IOUtil.java:122)
        at sun.nio.ch.IOUtil.write(IOUtil.java:93)
        at sun.nio.ch.SocketChannelImpl.write(SocketChannelImpl.java:352)
        at kafka.api.TopicDataSend.writeTo(FetchResponse.scala:123)
        at kafka.network.MultiSend.writeTo(Transmission.scala:101)
        at kafka.api.FetchResponseSend.writeTo(FetchResponse.scala:231)
        at kafka.network.Processor.write(SocketServer.scala:472)
        at kafka.network.Processor.run(SocketServer.scala:342)
        at java.lang.Thread.run(Thread.java:701)
[2015-08-01 22:18:26,611] INFO Closing socket connection to /127.0.0.1. (kafka.n
```

```
wei@wei-VirtualBox:~/Documents/kafka_test$ python kakfa_sample.py
2015-08-01 22:06:53,567.567.198991776:kafka.client:139975927813888:DEBUG:2966:R
quest 1: []
2015-08-01 22:06:53,567.567.55399704:kafka.conn:139975927813888:DEBUG:2966:Rein
tializing socket connection for localhost:9092
2015-08-01 22:06:53,569.569.159030914:kafka.client:139975936206592:DEBUG:2966:R
quest 1: []
2015-08-01 22:06:53,569.569.59605217:kafka.conn:139975936206592:DEBUG:2966:Rein
tializing socket connection for localhost:9092
2015-08-01 22:06:53,569.569.467067719:kafka.conn:139975927813888:DEBUG:2966:Abo
t to send 30 bytes to Kafka, request 1
2015-08-01 22:06:53,573.573.546886444:kafka.conn:139975927813888:DEBUG:2966:Rea
ing response 1 from Kafka
2015-08-01 22:06:53,573.573.791980743:kafka.conn:139975927813888:DEBUG:2966:Abo
t to read 4 bytes from Kafka
2015-08-01 22:06:53,574.574.00894165:kafka.conn:139975927813888:DEBUG:2966:Read
4/4 bytes from Kafka
2015-08-01 22:06:53,574.574.278116226:kafka.conn:139975927813888:DEBUG:2966:Abo
t to read 122 bytes from Kafka
2015-08-01 22:06:53,574.574.456930161:kafka.conn:139975927813888:DEBUG:2966:Rea
 122/122 bytes from Kafka
2015-08-01 22:06:53,574.574.717998505:kafka.client:139975927813888:DEBUG:2966:R
sponse 1: MetadataResponse(brokers=[BrokerMetadata(nodeId=0, host='wei-VirtualB
x', port=9092)], topics=[TopicMetadata(topic='my-topic', error=0, partitions=[P
rtitionMetadata(topic='my-topic', partition=0, leader=0, replicas=(0,), isr=(0,
, error=0)]), TopicMetadata(topic='test-topic', error=0, partitions=[PartitionM
tadata(topic='test-topic', partition=0, leader=0, replicas=(0,), isr=(0,), erro
=0)])])
2015-08-01 22:06:53,574.574.886083603:kafka.client:139975927813888:INFO:2966:Up
ating broker metadata: [BrokerMetadata(nodeId=0, host='wei-VirtualBox', port=90
2)]
2015-08-01 22:06:53,575.575.156927109:kafka.client:139975927813888:INFO:2966:Up
ating topic metadata: [TopicMetadata(topic='my-topic', error=0, partitions=[Par
itionMetadata(topic='my-topic', partition=0, leader=0, replicas=(0,), isr=(0,),
error=0)]), TopicMetadata(topic='test-topic', error=0, partitions=[PartitionMet
data(topic='test-topic', partition=0, leader=0, replicas=(0,), isr=(0,), error=
)])]
2015-08-01 22:06:53,575.575.515985489:kafka.client:139975927813888:DEBUG:2966:R
quest 2: ['my-topic']
2015-08-01 22:06:53,575.575.850963593:kafka.conn:139975927813888:DEBUG:2966:Abo
t to send 40 bytes to Kafka, request 2
2015-08-01 22:06:53,574.574.172973633:kafka.conn:139975936206592:DEBUG:2966:Abo
t to send 30 bytes to Kafka, request 1
```

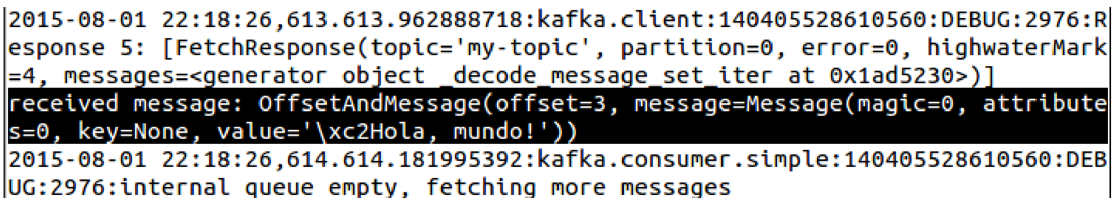
key output of program:

produce message:



```
2015-08-01 22:18:26,547.547.633886337:kafka.client:140405537003264:DEBUG:2976:R
esponse 1: MetadataResponse(brokers=[BrokerMetadata(nodeId=0, host='wei-Virtual
Box', port=9092)], topics=[TopicMetadata(topic='my-topic', error=0, partitions=
[PartitionMetadata(topic='my-topic', partition=0, leader=0, replicas=(0,), isr=
(0,), error=0)]), TopicMetadata(topic='test-topic', error=0, partitions=[Partit
ionMetadata(topic='test-topic', partition=0, leader=0, replicas=(0,), isr=(0,),
 error=0)])])
```

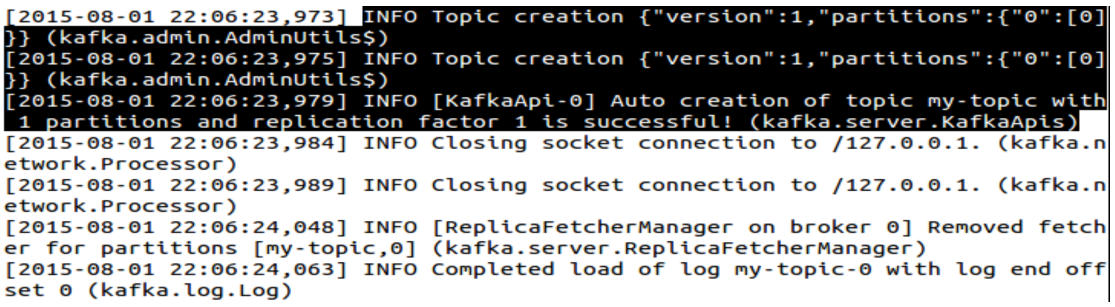
consume message:



```
2015-08-01 22:18:26,613.613.962888718:kafka.client:140405528610560:DEBUG:2976:R
esponse 5: [FetchResponse(topic='my-topic', partition=0, error=0, highwaterMark
=4, messages=<generator object _decode_message_set_iter at 0x1ad5230>)]
received message: OffsetAndMessage(offset=3, message=Message(magic=0, attribute
s=0, key=None, value='\xc2Hola, mundo!'))
2015-08-01 22:18:26,614.614.181995392:kafka.consumer.simple:140405528610560:DEB
UG:2976:internal queue empty, fetching more messages
```

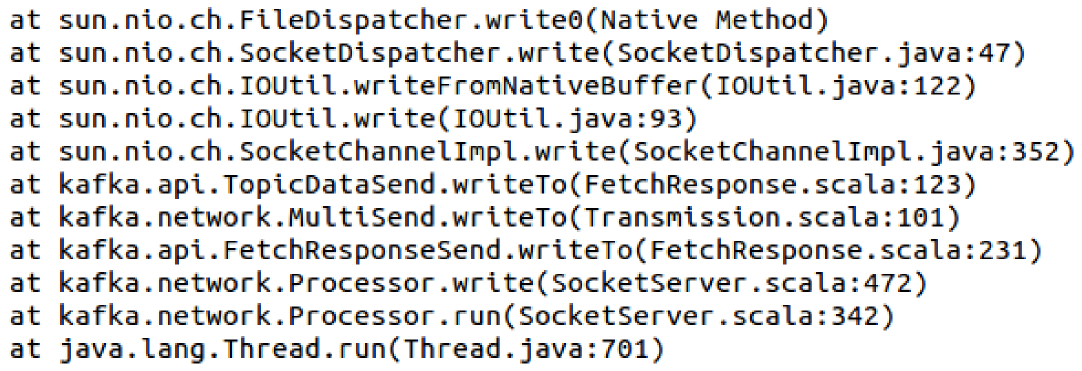
key output of kafka server:

create new topic for 'my-topic' and set log file:



```
[2015-08-01 22:06:23,973] INFO Topic creation {"version":1,"partitions":{"0":[0]
}} (kafka.admin.AdminUtils$)
[2015-08-01 22:06:23,975] INFO Topic creation {"version":1,"partitions":{"0":[0]
}} (kafka.admin.AdminUtils$)
[2015-08-01 22:06:23,979] INFO [KafkaApi-0] Auto creation of topic my-topic with
 1 partitions and replication factor 1 is successful! (kafka.server.KafkaApis)
[2015-08-01 22:06:23,984] INFO Closing socket connection to /127.0.0.1. (kafka.n
etwork.Processor)
[2015-08-01 22:06:23,989] INFO Closing socket connection to /127.0.0.1. (kafka.n
etwork.Processor)
[2015-08-01 22:06:24,048] INFO [ReplicaFetcherManager on broker 0] Removed fetch
er for partitions [my-topic,0] (kafka.server.ReplicaFetcherManager)
[2015-08-01 22:06:24,063] INFO Completed load of log my-topic-0 with log end off
set 0 (kafka.log.Log)
```

Passing messages (through java code):



```
 at sun.nio.ch.FileDispatcher.write0(Native Method)
 at sun.nio.ch.SocketDispatcher.write(SocketDispatcher.java:47)
 at sun.nio.ch.IOUtil.writeFromNativeBuffer(IOUtil.java:122)
 at sun.nio.ch.IOUtil.write(IOUtil.java:93)
 at sun.nio.ch.SocketChannelImpl.write(SocketChannelImpl.java:352)
 at kafka.api.TopicDataSend.writeTo(FetchResponse.scala:123)
 at kafka.network.MultiSend.writeTo(Transmission.scala:101)
 at kafka.api.FetchResponseSend.writeTo(FetchResponse.scala:231)
 at kafka.network.Processor.write(SocketServer.scala:472)
 at kafka.network.Processor.run(SocketServer.scala:342)
 at java.lang.Thread.run(Thread.java:701)
```

Code Assessment

Download the source code from github:
https://github.com/mumrah/kafka-python/releases/tag/v0.9.4

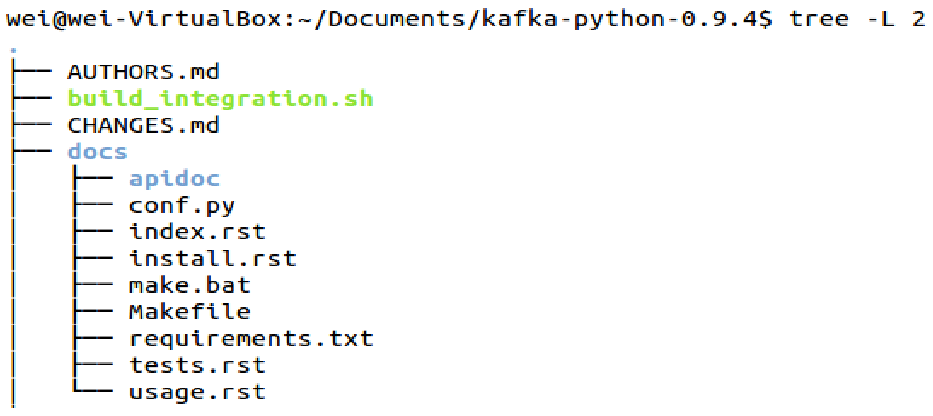
use pylint to do a simple code assessment for main functions.

[1]. Install pylint by pip install pylint.

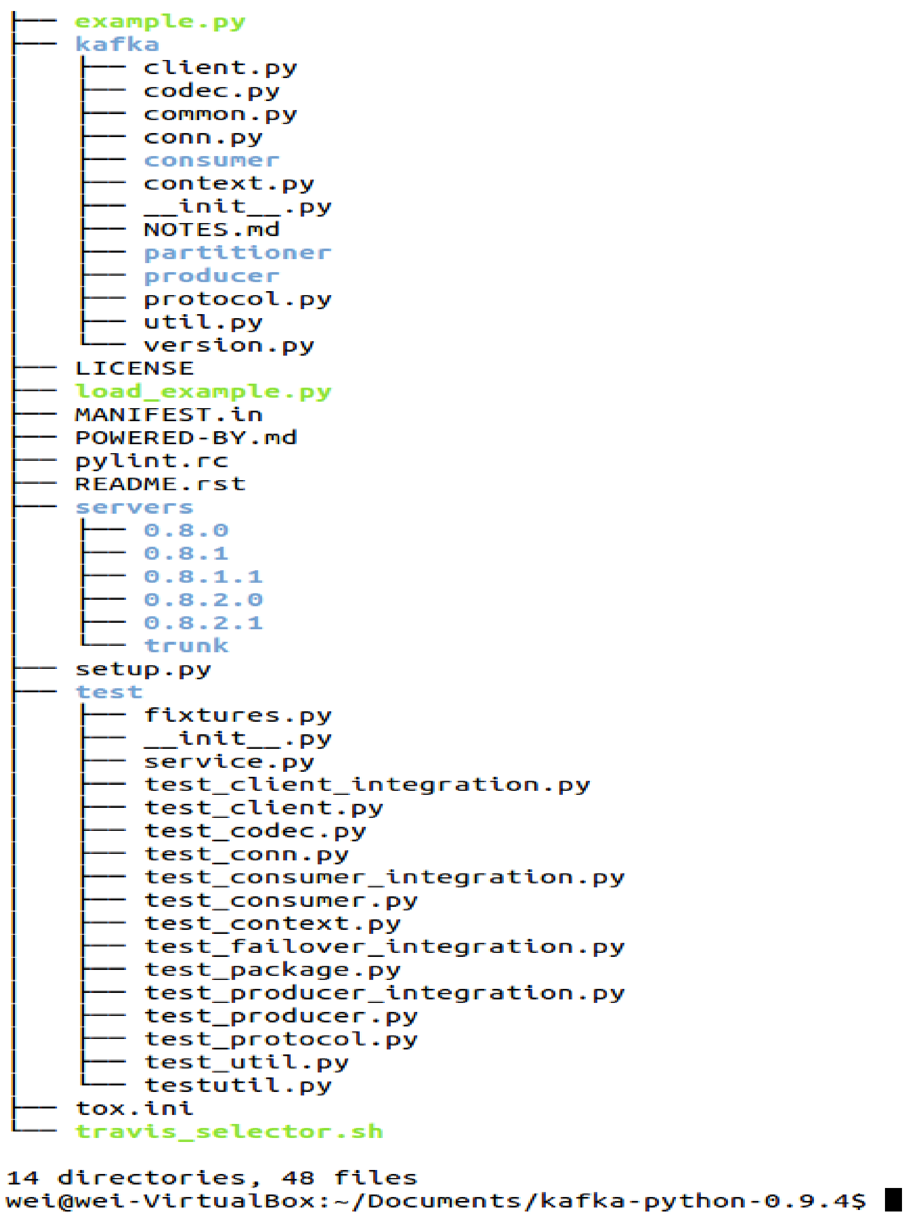
[2]. To grade your source code, run pylint --rcfile [filepath of .pylintrc] [package directory path or a single file]
For example, pylint --rcfile .pylintrc lib/pkg

[3]. To grade your test code, run pylint --disable=F0401,E0611 --rcfile .pylintrc test

The structure of this package shows below:

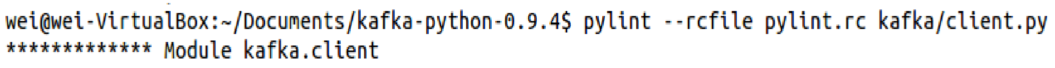


```
wei@wei-VirtualBox:~/Documents/kafka-python-0.9.4$ tree -L 2
.
├── AUTHORS.md
├── build_integration.sh
├── CHANGES.md
├── docs
│   ├── apidoc
│   ├── conf.py
│   ├── index.rst
│   ├── install.rst
│   ├── make.bat
│   ├── Makefile
│   ├── requirements.txt
│   ├── tests.rst
│   └── usage.rst
│
```



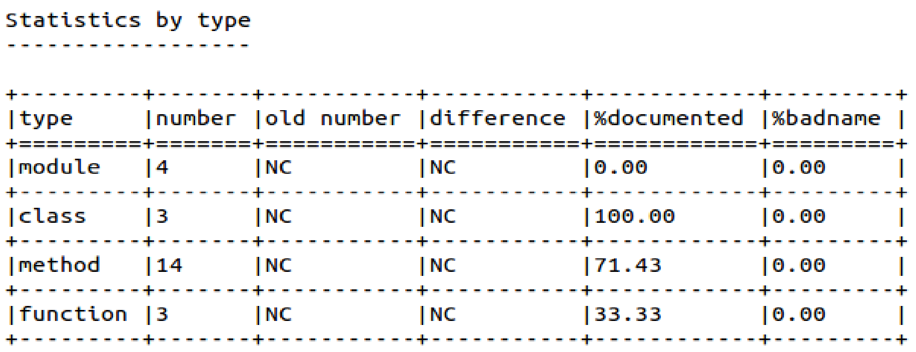
```
├── example.py
├── kafka
│   ├── client.py
│   ├── codec.py
│   ├── common.py
│   ├── conn.py
│   ├── consumer
│   ├── context.py
│   ├── __init__.py
│   ├── NOTES.md
│   ├── partitioner
│   ├── producer
│   ├── protocol.py
│   ├── util.py
│   └── version.py
├── LICENSE
├── load_example.py
├── MANIFEST.in
├── POWERED-BY.md
├── pylint.rc
├── README.rst
├── servers
│   ├── 0.8.0
│   ├── 0.8.1
│   ├── 0.8.1.1
│   ├── 0.8.2.0
│   ├── 0.8.2.1
│   └── trunk
├── setup.py
├── test
│   ├── fixtures.py
│   ├── __init__.py
│   ├── service.py
│   ├── test_client_integration.py
│   ├── test_client.py
│   ├── test_codec.py
│   ├── test_conn.py
│   ├── test_consumer_integration.py
│   ├── test_consumer.py
│   ├── test_context.py
│   ├── test_failover_integration.py
│   ├── test_package.py
│   ├── test_producer_integration.py
│   ├── test_producer.py
│   ├── test_protocol.py
│   ├── test_util.py
│   └── testutil.py
├── tox.ini
└── travis_selector.sh

14 directories, 48 files
wei@wei-VirtualBox:~/Documents/kafka-python-0.9.4$ █
```

code assessment for client.py:



```
wei@wei-VirtualBox:~/Documents/kafka-python-0.9.4$ pylint --rcfile pylint.rc kafka/client.py
************* Module kafka.client
```

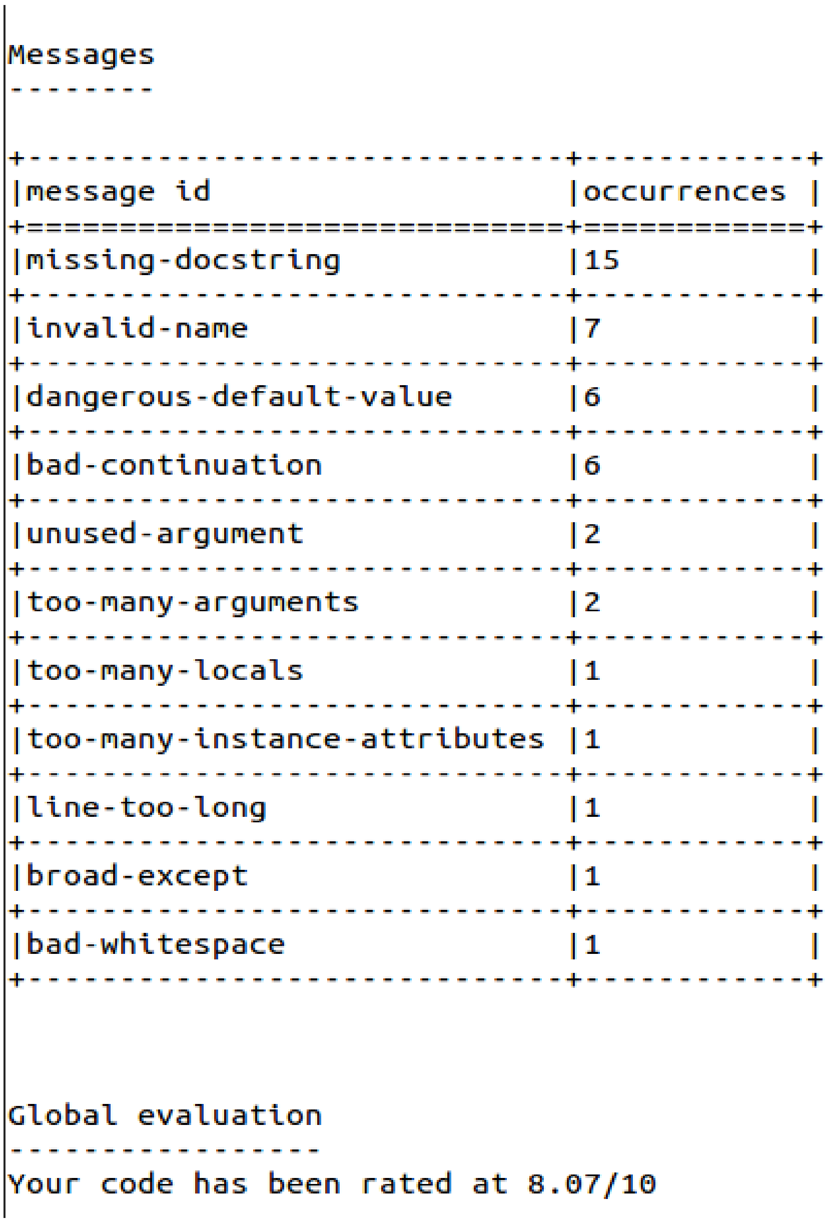
result:



```
Statistics by type
------------------

+---------+-------+-----------+-----------+------------+---------+
|type     |number |old number |difference |%documented |%badname |
+=========+=======+===========+===========+============+=========+
|module   |4      |NC         |NC         |0.00        |0.00     |
+---------+-------+-----------+-----------+------------+---------+
|class    |3      |NC         |NC         |100.00      |0.00     |
+---------+-------+-----------+-----------+------------+---------+
|method   |14     |NC         |NC         |71.43       |0.00     |
+---------+-------+-----------+-----------+------------+---------+
|function |3      |NC         |NC         |33.33       |0.00     |
+---------+-------+-----------+-----------+------------+---------+
Raw metrics
-----------

+----------+-------+------+---------+-----------+
|type      |number |%     |previous |difference |
+==========+=======+======+=========+===========+
|code      |272    |59.52 |NC       |NC         |
+----------+-------+------+---------+-----------+
|docstring |113    |24.73 |NC       |NC         |
+----------+-------+------+---------+-----------+
|comment   |49     |10.72 |NC       |NC         |
+----------+-------+------+---------+-----------+
|empty     |23     |5.03  |NC       |NC         |
+----------+-------+------+---------+-----------+
```

```
Duplication
-----------

+-------------------------+------+---------+-----------+
|                         |now   |previous |difference |
+=========================+======+=========+===========+
|nb duplicated lines      |0     |NC       |NC         |
+-------------------------+------+---------+-----------+
|percent duplicated lines |0.000 |NC       |NC         |
+-------------------------+------+---------+-----------+
```

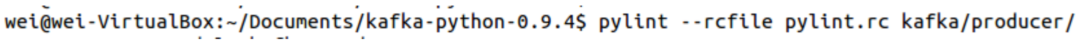
```
Messages by category
--------------------

+-----------+-------+---------+-----------+
|type       |number |previous |difference |
+===========+=======+=========+===========+
|convention |30     |NC       |NC         |
+-----------+-------+---------+-----------+
|refactor   |4      |NC       |NC         |
+-----------+-------+---------+-----------+
|warning    |9      |NC       |NC         |
+-----------+-------+---------+-----------+
|error      |0      |NC       |NC         |
+-----------+-------+---------+-----------+
```



```
Messages
--------

+------------------------------+------------+
|message id                    |occurrences |
+==============================+============+
|missing-docstring             |15          |
+------------------------------+------------+
|invalid-name                  |7           |
+------------------------------+------------+
|dangerous-default-value       |6           |
+------------------------------+------------+
|bad-continuation              |6           |
+------------------------------+------------+
|unused-argument               |2           |
+------------------------------+------------+
|too-many-arguments            |2           |
+------------------------------+------------+
|too-many-locals               |1           |
+------------------------------+------------+
|too-many-instance-attributes  |1           |
+------------------------------+------------+
|line-too-long                 |1           |
+------------------------------+------------+
|broad-except                  |1           |
+------------------------------+------------+
|bad-whitespace                |1           |
+------------------------------+------------+



Global evaluation
-----------------
Your code has been rated at 8.07/10
```

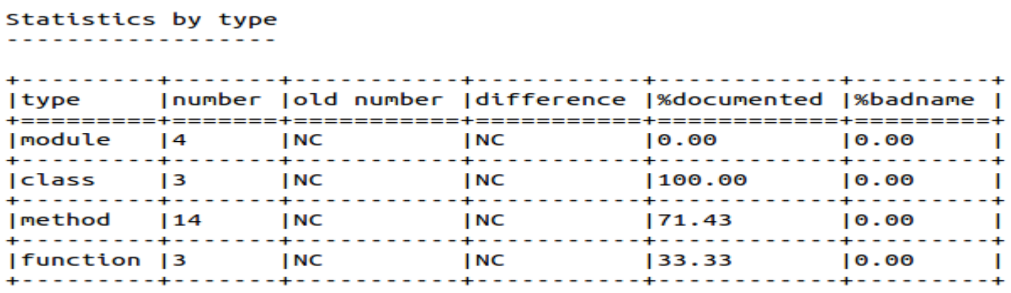
The most serious problem is missing documents 15 times, which may make the code hard to read and understand.

code assessment for producer:



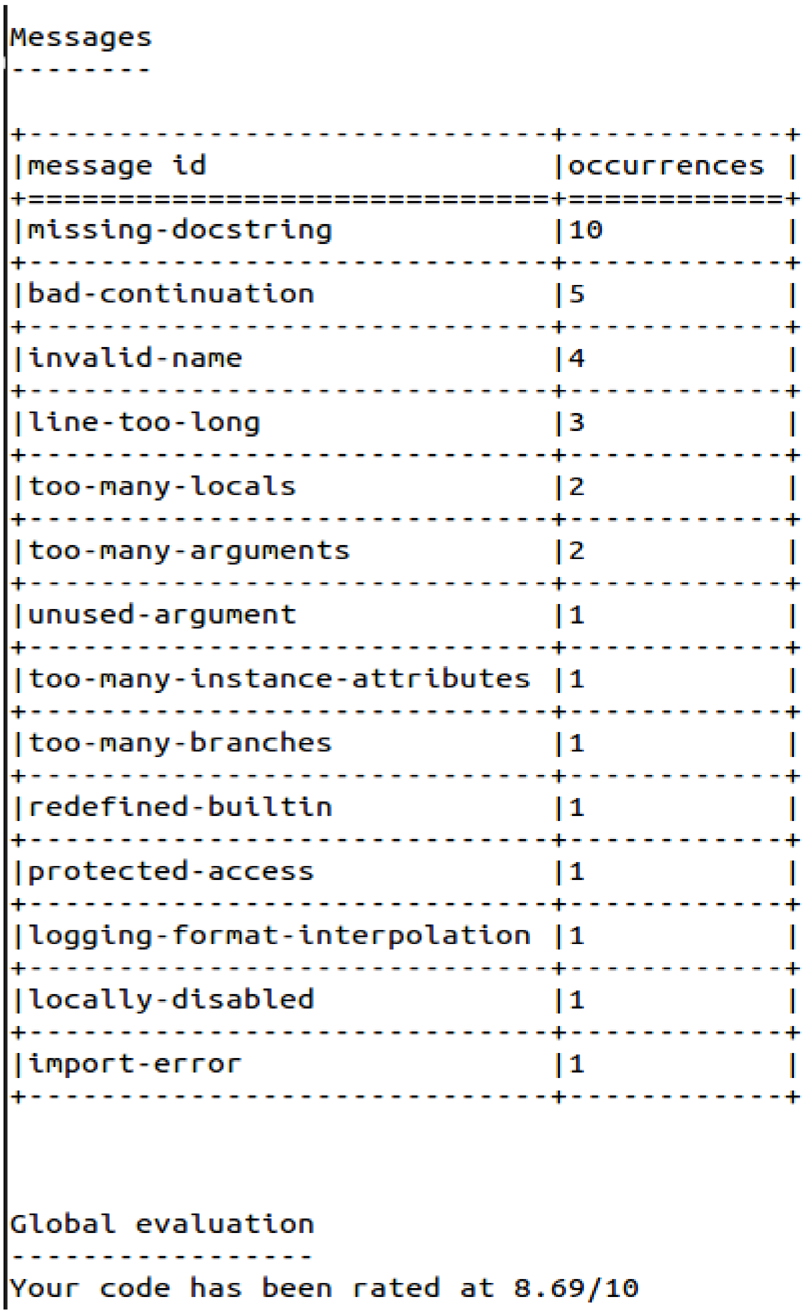
```
wei@wei-VirtualBox:~/Documents/kafka-python-0.9.4$ pylint --rcfile pylint.rc kafka/producer/
```

result:



```
Statistics by type
------------------

+---------+-------+-----------+-----------+------------+---------+
|type     |number |old number |difference |%documented |%badname |
+=========+=======+===========+===========+============+=========+
|module   |4      |NC         |NC         |0.00        |0.00     |
+---------+-------+-----------+-----------+------------+---------+
|class    |3      |NC         |NC         |100.00      |0.00     |
+---------+-------+-----------+-----------+------------+---------+
|method   |14     |NC         |NC         |71.43       |0.00     |
+---------+-------+-----------+-----------+------------+---------+
|function |3      |NC         |NC         |33.33       |0.00     |
+---------+-------+-----------+-----------+------------+---------+
Raw metrics
-----------

+----------+-------+------+---------+-----------+
|type      |number |%     |previous |difference |
+==========+=======+======+=========+===========+
|code      |308    |64.30 |NC       |NC         |
+----------+-------+------+---------+-----------+
|docstring |124    |25.89 |NC       |NC         |
+----------+-------+------+---------+-----------+
|comment   |24     |5.01  |NC       |NC         |
+----------+-------+------+---------+-----------+
|empty     |23     |4.80  |NC       |NC         |
+----------+-------+------+---------+-----------+



Duplication
-----------

+-------------------------+------+---------+-----------+
|                         |now   |previous |difference |
+=========================+======+=========+===========+
|nb duplicated lines      |0     |NC       |NC         |
+-------------------------+------+---------+-----------+
|percent duplicated lines |0.000 |NC       |NC         |
+-------------------------+------+---------+-----------+



Messages by category
--------------------

+-----------+-------+---------+-----------+
|type       |number |previous |difference |
+===========+=======+=========+===========+
|convention |22     |NC       |NC         |
+-----------+-------+---------+-----------+
|refactor   |6      |NC       |NC         |
+-----------+-------+---------+-----------+
|warning    |4      |NC       |NC         |
+-----------+-------+---------+-----------+
|error      |0      |NC       |NC         |
+-----------+-------+---------+-----------+
```

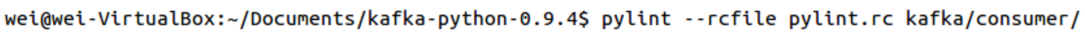
```
+----------------------+------+--------+---------+-----------+
|module                |error |warning |refactor |convention |
+======================+======+========+=========+===========+
|kafka.producer.base   |0.00  |75.00   |100.00   |54.55      |
+----------------------+------+--------+---------+-----------+
|kafka.producer.simple |0.00  |25.00   |0.00     |13.64      |
+----------------------+------+--------+---------+-----------+
|kafka.producer.keyed  |0.00  |0.00    |0.00     |27.27      |
+----------------------+------+--------+---------+-----------+
|kafka.producer        |0.00  |0.00    |0.00     |4.55       |
+----------------------+------+--------+---------+-----------+
```



```
Messages
--------

+------------------------------+------------+
|message id                    |occurrences |
+==============================+============+
|missing-docstring             |10          |
+------------------------------+------------+
|bad-continuation              |5           |
+------------------------------+------------+
|invalid-name                  |4           |
+------------------------------+------------+
|line-too-long                 |3           |
+------------------------------+------------+
|too-many-locals               |2           |
+------------------------------+------------+
|too-many-arguments            |2           |
+------------------------------+------------+
|unused-argument               |1           |
+------------------------------+------------+
|too-many-instance-attributes  |1           |
+------------------------------+------------+
|too-many-branches             |1           |
+------------------------------+------------+
|redefined-builtin             |1           |
+------------------------------+------------+
|protected-access              |1           |
+------------------------------+------------+
|logging-format-interpolation  |1           |
+------------------------------+------------+
|locally-disabled              |1           |
+------------------------------+------------+
|import-error                  |1           |
+------------------------------+------------+



Global evaluation
-----------------
Your code has been rated at 8.69/10
```

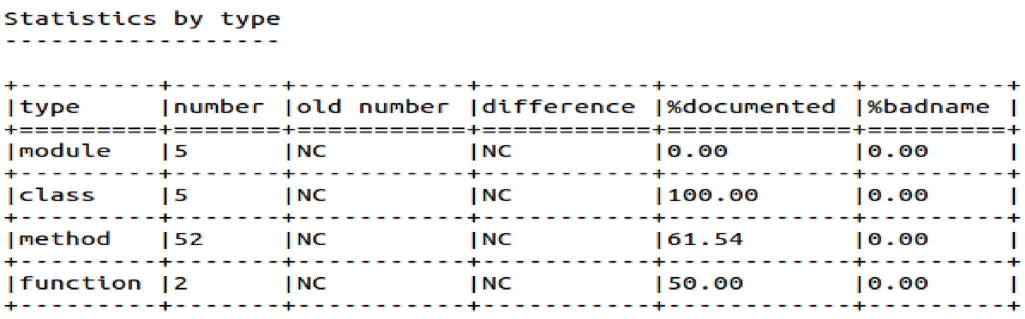
Overall, the code grades is good enough, still missing some documents for functions.

code assessment for consumer:

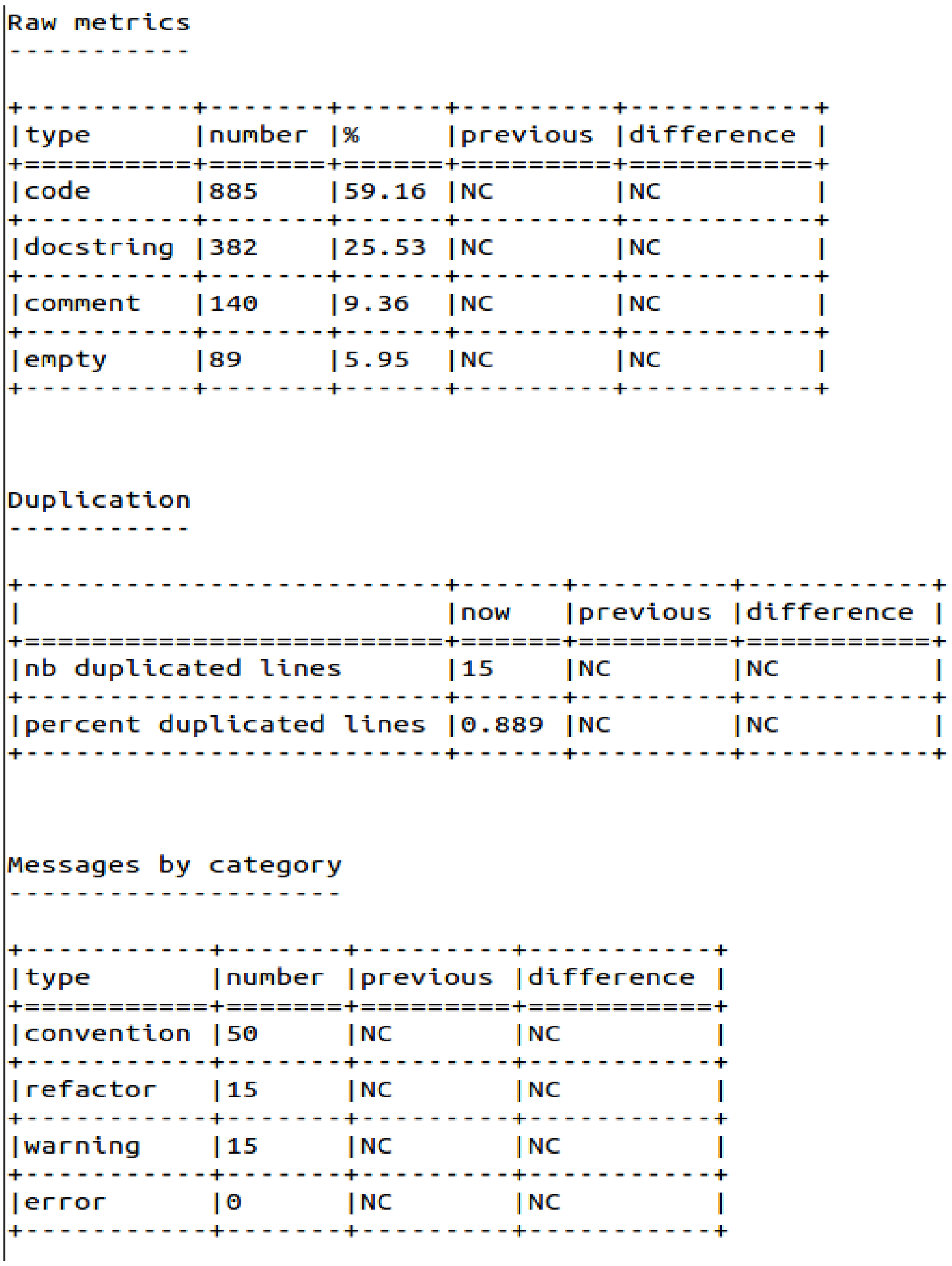


```
wei@wei-VirtualBox:~/Documents/kafka-python-0.9.4$ pylint --rcfile pylint.rc kafka/consumer/
```

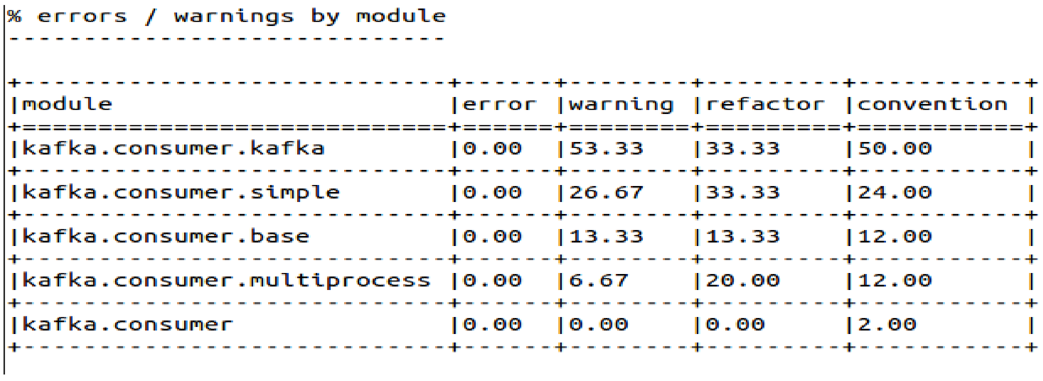
result:



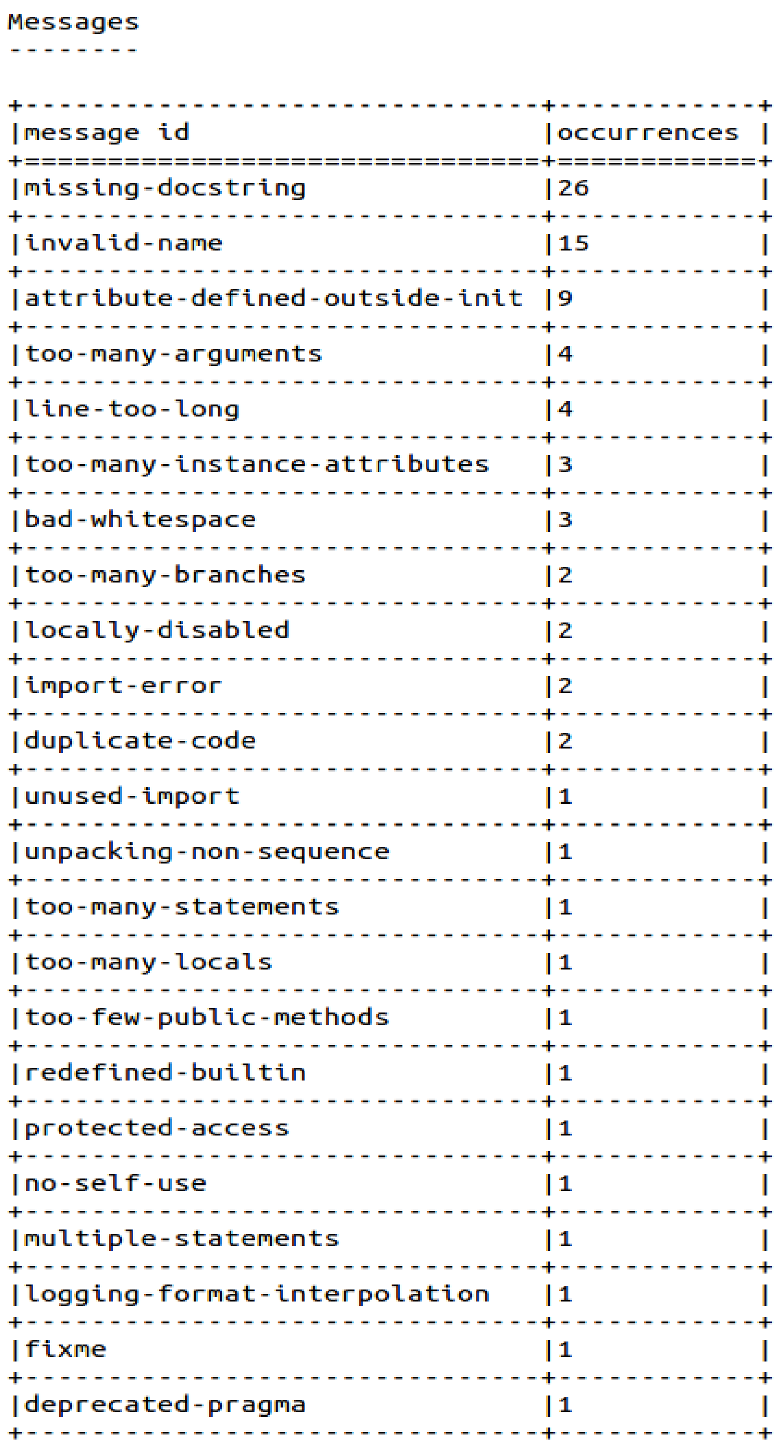
```
Statistics by type
------------------

+---------+-------+-----------+-----------+------------+---------+
|type     |number |old number |difference |%documented |%badname |
+=========+=======+===========+===========+============+=========+
|module   |5      |NC         |NC         |0.00        |0.00     |
+---------+-------+-----------+-----------+------------+---------+
|class    |5      |NC         |NC         |100.00      |0.00     |
+---------+-------+-----------+-----------+------------+---------+
|method   |52     |NC         |NC         |61.54       |0.00     |
+---------+-------+-----------+-----------+------------+---------+
|function |2      |NC         |NC         |50.00       |0.00     |
+---------+-------+-----------+-----------+------------+---------+
```



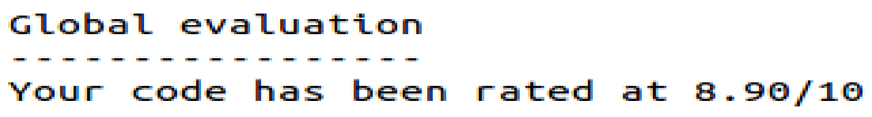
```
Raw metrics
-----------

+----------+-------+------+---------+-----------+
|type      |number |%     |previous |difference |
+==========+=======+======+=========+===========+
|code      |885    |59.16 |NC       |NC         |
+----------+-------+------+---------+-----------+
|docstring |382    |25.53 |NC       |NC         |
+----------+-------+------+---------+-----------+
|comment   |140    |9.36  |NC       |NC         |
+----------+-------+------+---------+-----------+
|empty     |89     |5.95  |NC       |NC         |
+----------+-------+------+---------+-----------+



Duplication
-----------

+-------------------------+------+---------+-----------+
|                         |now   |previous |difference |
+=========================+======+=========+===========+
|nb duplicated lines      |15    |NC       |NC         |
+-------------------------+------+---------+-----------+
|percent duplicated lines |0.889 |NC       |NC         |
+-------------------------+------+---------+-----------+



Messages by category
--------------------

+-----------+-------+---------+-----------+
|type       |number |previous |difference |
+===========+=======+=========+===========+
|convention |50     |NC       |NC         |
+-----------+-------+---------+-----------+
|refactor   |15     |NC       |NC         |
+-----------+-------+---------+-----------+
|warning    |15     |NC       |NC         |
+-----------+-------+---------+-----------+
|error      |0      |NC       |NC         |
+-----------+-------+---------+-----------+
```



```
% errors / warnings by module
-----------------------------

+-----------------------------+------+--------+---------+-----------+
|module                       |error |warning |refactor |convention |
+=============================+======+========+=========+===========+
|kafka.consumer.kafka         |0.00  |53.33   |33.33    |50.00      |
+-----------------------------+------+--------+---------+-----------+
|kafka.consumer.simple        |0.00  |26.67   |33.33    |24.00      |
+-----------------------------+------+--------+---------+-----------+
|kafka.consumer.base          |0.00  |13.33   |13.33    |12.00      |
+-----------------------------+------+--------+---------+-----------+
|kafka.consumer.multiprocess  |0.00  |6.67    |20.00    |12.00      |
+-----------------------------+------+--------+---------+-----------+
|kafka.consumer               |0.00  |0.00    |0.00     |2.00       |
+-----------------------------+------+--------+---------+-----------+
```



```
Messages
--------

+--------------------------------+------------+
|message id                      |occurrences |
+================================+============+
|missing-docstring               |26          |
+--------------------------------+------------+
|invalid-name                    |15          |
+--------------------------------+------------+
|attribute-defined-outside-init  |9           |
+--------------------------------+------------+
|too-many-arguments              |4           |
+--------------------------------+------------+
|line-too-long                   |4           |
+--------------------------------+------------+
|too-many-instance-attributes    |3           |
+--------------------------------+------------+
|bad-whitespace                  |3           |
+--------------------------------+------------+
|too-many-branches               |2           |
+--------------------------------+------------+
|locally-disabled                |2           |
+--------------------------------+------------+
|import-error                    |2           |
+--------------------------------+------------+
|duplicate-code                  |2           |
+--------------------------------+------------+
|unused-import                   |1           |
+--------------------------------+------------+
|unpacking-non-sequence          |1           |
+--------------------------------+------------+
|too-many-statements             |1           |
+--------------------------------+------------+
|too-many-locals                 |1           |
+--------------------------------+------------+
|too-few-public-methods          |1           |
+--------------------------------+------------+
|redefined-builtin               |1           |
+--------------------------------+------------+
|protected-access                |1           |
+--------------------------------+------------+
|no-self-use                     |1           |
+--------------------------------+------------+
|multiple-statements             |1           |
+--------------------------------+------------+
|logging-format-interpolation    |1           |
+--------------------------------+------------+
|fixme                           |1           |
+--------------------------------+------------+
|deprecated-pragma               |1           |
+--------------------------------+------------+
```

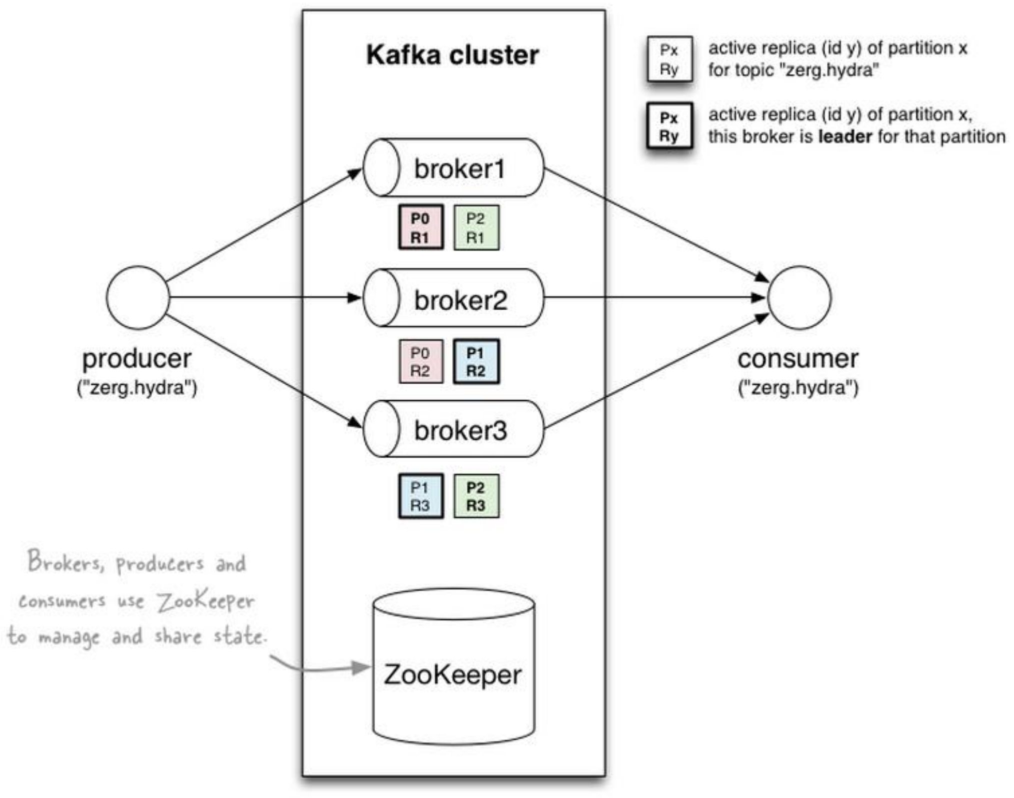


```
Global evaluation
-----------------
Your code has been rated at 8.90/10
```

Overall the code quality of consumer is good.

Existing Vulnerability

The structure of kakfa service:
Kafka a distributed message publishing/subscribing system of one or more brokers, each one with a set of zero or more partitions for each existing topic. Kafka persists periodically messages to disk, so in case of failure last ones might get loss. This speeds up the publishing operation, as publishers do not need to wait until that data gets written to disk.


Kafka cluster
Px Ry active replica (id y) of partition x for topic "zerg.hydra"
Px Ry active replica (id y) of partition x, this broker is leader for that partition
broker1
P0 R1
P2 R1
broker2
P0 R2
P1 R2
broker3
P1 R3
P2 R3
producer ("zerg.hydra")
consumer ("zerg.hydra")
Brokers, producers and consumers use ZooKeeper to manage and share state.
ZooKeeper

After analyze the structure, we can find some serious security related problems.
[1]. Any computer or host in the network can join th kafka broker cluster by starting the process of broker, so it can receive the message sent by producer, and also edit the message then send it to consumer.

[2]. Any computer or host in the network can start a malicious producer or consumer and connect to broker, sending illegal messages or pull privacy message data.

[3]. Broker does not support connection to the zookeeper cluster which enables

Kerberos certification, it does not set the permissions or right level for the data. Any user can access the zookeeper cluster freely and edit or delete the data.

[4]. The topic in the kafka does not support setting access control lists, like white list or black list, also, any consumer or producer connected to the kafka cluster can read or send any message of any topic.

Potentially Risky

With the increasing popularity of kafka, especially in some fields have higher level of data privacy. the problems above is like a time bomb, once the network been hacked or malicious internal user exists, all the private data can be stolen easily, without having to break the Broker server.

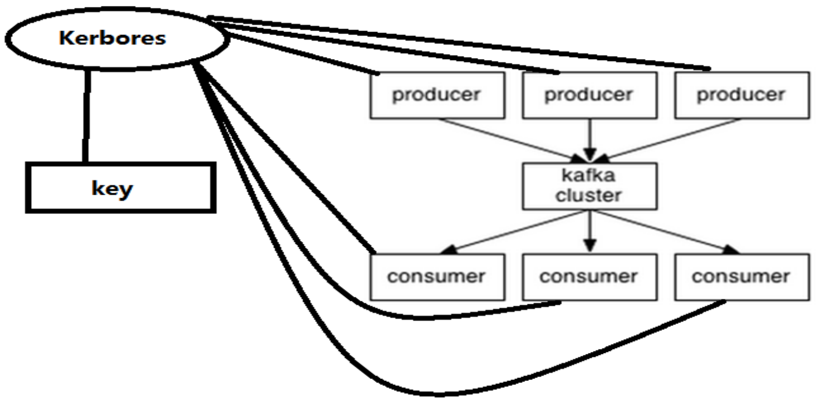
Recommendations

There are two aspects where we can improve kafka's security.
[1]. Authentication
Design and implement two mechanism of authentication based on Kerberos and IP. The former one is a strong authentication, the latter one works under reliable network environment, and easier to deploy compared to former one.

[2]. Authorization
Design and implement a topic level permission model. Divide the operations into four parts:

- READ: pull data from the topic
- WRITE: send data to topic
- CREATE: create topic
- DELETE: delete topic

Design:
When the broker starts, it needs to use the configuration file and key file to pass the certification of KDC (the server of Kerberos), it can join the cluster if the key file matched, otherwise report an error and exit.



Producer or customer mush follow the steps bellow to set a secure connection to the broker. (take producer as an example)

1. Producer asks for the KDC authentication identity, and obtained by TGT (Ticket to Get Tickets) if successfully passed, otherwise report an error and exit.

2. Producer use TGT to send request for the KDC Kafka services, KDC verifies the TGT and return the Session Key and Service Ticket to the Producer .

3.Producer Session Key and Service Ticket are used to establish a connection with the Broker, Broker uses its own key to decrypt Service Ticket to allow producer to communicate with Session Key, then use Session Key authenticate the connection producer established, otherwise refuse the connection.

Conclusion

Kafka service is a popular method to build message pipe between hosts in the network, it is easy to deploy and use, support multiple programming languages and has a good performance.

However, it does not cover many security related problem. Mainly, we can add authentication and authorization to protect the private data when using kafka service to pass information.

Reference

[1]. https://github.com/mumrah/kafka-python
[2]. https://kafka.apache.org/documentation.html
[3]. https://github.com/mumrah/kafka-python/releases/tag/v0.9.4
[4]. https://gist.github.com/devsprint/8663561
[5]. https://en.wikipedia.org/wiki/Kerberos_(protocol)