OTICS Advanced Analytics Inc.

MAADS-VIPER™ INSTALLATION and QUICK USER GUIDE

Date: December, 2021
MAADS-VIPER
v.5.5.60+

This guide will provide common setup instructions for new users who want to run VIPER in their environment(s). For any questions, users are encouraged to email support@otics.ca **with subject line: "VIPER HELP"** (no quotes).

A. SETUP Instructions: Watch the YouTube video or follow instructions below.

1) Download VIPER and extract its contents

a. You should see six (6) files:

i. This file: MAADS-Viper Installation Guide.pdf
ii. MAADS-Viper-Product Brief.pdf
iii. MAADS-Viper-Tests Results.pdf
iv. VIPER binary (this binary will be for your OS: Windows, Linux, Mac, etc.)
v. VIPER.ENV file (this is a viper configuration file)
vi. Token.Tok

b. Quick Start using REST API:

i. Create Topic:

1. http://localhost:8000/createtopics?topic=VIPERDEMOTOPIC1&maadsalgokey=&replicationfactor=3&numpartitions=3&companyname=OTICS Advanced Analytics Inc.&contactname=Sebastian&contactemail=sebastian.maurice@otics.ca&description=test&location=Toronto&vipertoken=demo&enabletls=1
    a. **Note 1:** &enabletls=1, if set to 1 then Kafka has SSL/TLS enabled, otherwise set to 0 and VIPER will connect with SSL/TLS
    b. **Note 2:** Change host:port to host and port VIPER is running on
    c. **Note 3:** Change replication factor and numpartitions to whatever number you wish
    d. **RETURN VALUE should look something like:** {"Topic":"VIPERDEMOTOPIC1","ProducerId":"ProducerId-oPtvGImUcijAXOdlrrDNYfyakSf5Rp"}

ii. **Produce to Topic (use the ProducerID to produce to topic):**

1. http://localhost:8000/producetotopic?topic=VIPERDEMOTOPIC1&producerid=ProducerId-oPtvGImUcijAXOdlrrDNYfyakSf5Rp&externalprediction=This is my test topic&vipertoken=demo&enabletls=1&delay=5000
    a. **Note 1:** &enabletls=1, if set to 1 then Kafka has SSL/TLS enabled, otherwise set to 0.
    b. **Note 2:** Change host:port to host and port VIPER is running on
    c. **Note 3:** Notice the Producer ID from 1d return value
    d. **Note 4:** Write your message in "externalprediction=This is my test topic"
    e. **Note 5:** &delay=5000, this means VIPER will wait 5000 ms before backing out, if no response from Kafka

iii. Subscribe Consumer to Topic VIPERDEMOTOPIC1:

1. http://localhost:8000/subscribeconsumer?topic=VIPERDEMOTOPIC1&companyname=OTICS test&contactname=sebastian maurice&contactemail=sebastian.maurice@otics.ca&location=Toronot&description=This is a test consumer&vipertoken=Demo
    a. **Note 2:** Change host:port to host and port VIPER is running on
    b. **Note 3:** Change replication factor and numpartitions to whatever number you wish
    c. **RETURN VALUE should look something like:** {"Consumerid":"ConsumerId-D9ib85jS40UeDCCgLRjgkbJamnJPez","Topic":"VIPERDEMOTOPIC1"}

iv. **Consume From Topic:** VIPERDEMOTOPIC1

1. http://localhost:8000/consumefromtopic?topic=VIPERDEMOTOPIC1&consumerid=ConsumerId-D9ib85jS40UeDCCgLRjgkbJamnJPez&companyname=OTICS&vipertoken=demo&enabletls=1&delay=1000&partition=1&offset=0
    a. **Note 1:** &enabletls=1, if set to 1 then Kafka has SSL/TLS enabled, otherwise set to 0.
    b. **Note 2:** Change host:port to host and port VIPER is running on
    c. **Note 3:** &delay=1000, this means VIPER will wait 1000 ms before backing out, if no response from Kafka

d. **Note 4:** Set the partition to the number of your choice, or -1 to let VIPER autodetect
e. **Note 5:** Set the offset to your choice, or -1 to go to the last offset

v. **List Stats in VIPER:**
1. http://localhost:8000/viperstats?vipertoken=demo

c. For actual (non-Demo) use you will need:
i. ADMIN.tok
1. This allows admin users to create topics, activate/deactivate topics, produce to topics
ii. USER.tok
1. This allows users to consume from topic

d. VIPER comes with its own embedded database to store metadata – called "VIPER.db"

2) Store all of the above files in the same directory you use to run VIPER
a. VIPER will automatically create necessary directories in that folder
b. **Note:** For Linux users File/Folder permissions may need to be adjusted for VIPER 0644 is usually fine

3) Start VIPER
a. By default, VIPER listens on "Localhost" port=8000
b. You can easily adjust this to whatever host/port you want by typing: [Viper Executable] [host] [port]

4) On Startup VIPER will check for:
a. Valid Tokens
b. VIPER.ENV file

VIPER.ENV Configurations

1) With SSL/TLS enabled

a. If you have enabled SSL/TLS on Kafka brokers then you need to specify additional fields in the configuration file – for example purposes .PEM files are added to the configuration keys, but you can specify folder/file names as you wish:
   i. SSL_CLIENT_CERT_FILE=<client.cer.pem>
   ii. SSL_CLIENT_KEY_FILE=<client.key.pem>
   iii. SSL_SERVER_CERT_FILE=<server.cer.pem>

   Note: First time the plain text values need to be entered, on start VIPER will hide these values. You can update them with plain text again if you change the .pem files then simply restart VIPER to hide the updated values again.

b. **Note:** You need to convert the certificates to a series of PEM files. Here are most common steps to do this:
   i. First extract the Certificate Authority (CA) using ***keytools and openssl:***
      1. `$ keytool -importkeystore -srckeystore kafka.server.truststore.jks -destkeystore server.p12 -deststoretype PKCS12`
         `$ openssl pkcs12 -in server.p12 -nokeys -out server.cer.pem`
      2. **Output:** `server.cer.pem`

   ii. Next, convert the client keystore:
      1. `$ keytool -importkeystore -srckeystore kafka.server.keystore.jks -destkeystore client.p12 -deststoretype PKCS12`
         `$ openssl pkcs12 -in client.p12 -nokeys -out client.cer.pem`
         `$ openssl pkcs12 -in client.p12 -nodes -nocerts -out client.key.pem`
      2. **Output:** `client.cer.pem`
      3. **Output:** `client.key.pem`

c. Add path/file names to the configurations:
   i. SSL_CLIENT_CERT_FILE=client.cer.pem
   ii. SSL_CLIENT_KEY_FILE=client.key.pem
   iii. SSL_SERVER_CERT_FILE=server.cer.pem

2) No SSL/TLS Security:

a. Assuming you have Kafka/Zookeeper running on a broker, simply fill in the following information in the configuration file:
   i. KAFKA_ADVERTISED_HOST_NAME=kafka
   ii. KAFKA_ZOOKEEPER_CONNECT=zookeeper:2181
   iii. KAFKA_CONNECT_BOOTSTRAP_SERVERS=localhost:9092, localhost:9093
   iv. KAFKA_ROOT=/var/kafka

b. With HPDE:
   i. HPDE_SERVER=<HPDE host>
   ii. HPDE_PORT=<HPDE listening port>

c. With Confluent Cloud Access:
   i. CLOUD_USERNAME=<cloud key>
   ii. CLOUD_PASSWORD=<cloud secret>

   Note: First time the plain text values need to be entered, on start VIPER will hide these values. You can update them with plain text again if you change the key/secret then simply restart VIPER to hide the updated values again.

Table 1: Configuration Parameter Details

Configuration Parameter	Description
KAFKA_ADVERTISED_HOST_NAME	Advertised host name in Kafka server properties
KAFKA_ZOOKEEPER_CONNECT	Zookeeper host name and port
KAFKA_CONNECT_BOOTSTRAP_SERVERS=	Kafka bootstrap servers – separate multiple servers by comma
MAADS_ALGORITHM_SERVER	MAADS algorithm server host URL. This will require MAADS software installed in the host. This is needed to generate predictions from algorithms generated by MAADS.

ONPREM	Set to 1, if running VIPER on-premise, or 0 if using Cloud
VIPERDEBUG	Set to 1, if you want additional screen logging, or 0. Set to 2, if you want additional screen ***and*** disk logging. Logs will be written to **./viperlogs/viperlogs.txt** This is helpful if you want to see details when building TML solutions. However, for production deployments, VIPERDEBUG should be set to 1 for optimized performance.
WRITETOVIPERDB	Set to 1, if you want to write Egress and Ingress bytes. Set to 0 if you do not want to write to viper.db. By setting to 0 this will speed up VIPER, but you will not get Egress and Ingress details in AIMS.
COMPRESSIONTYPE	You can force the producer to compress data. You can set this to: NONE, SNAPPY, GZIP, LZ4, default is NONE.
USEHTTP	Set to 1 if using HTTP to connect to VIPER. If SSL_CLIENT_CERT_FILE and SSL_CLIENT_KEY_FILE are specified then VIPER will automatically accept HTTPS connections. However, if USEHTTP=1, then regardless of certificates, HTTP will be used.
LOGSTREAMTOPIC	Enter the name of the topic that you want to write logs to. If this field is non-empty VIPER/HPDE/VIPERVIZ will all write logging information to this stream.
LOGSENDTOEMAILS	Viper will send log emails to these addresses: separate multiple addresses by comma.
LOGSENDTOEMAILSSUBJECT	You can add a custom subject for the email.
LOGSENDTOEMAILFOOTER	Specify additional text to be included in the footer of your email.
KUBERNETES	If deploying to Kubernetes, set to 1 and VIPER will dynamically get IP address of Pod, and free port.
MAXVIPERVIZROLLBACKOFFSET	Sets the maximum rollback offset in VIPERVIZ. This prevents memory heap issues.
MAXVIPERVIZCONNECTIONS	Total number of simultaneous connections to Viperviz. For example, MAXVIPERVIZCONNECTIONS=5
SASLMECHANISM	Choose SASL mechanism. You can specify: PLAIN, SCRAM256, SCRAM512
LOGSTREAMTOPICPARTITIONS	Enter number of partitions for LOGSTREAMTOPIC, i.e. 3
LOGSTREAMTOPICREPLICATION FACTOR	Enter replication factor for LOGSTREAMTOPIC, i.e. 3
MAADS_ALGORITHM_SERVER_P ORT	MAADS algorithm server host PORT. This will require MAADS software installed in the host. This is needed to generate predictions from algorithms generated by MAADS.
MAXTRAININGROWS	Maximum number of rows for training dataset. Higher number will consumer more memory resources.
MAXOPENREQUESTS	How many outstanding requests a connection is allowed to have before sending on it blocks (default 5).
MAXPREDICTIONROWS	Maximum prediction batch size.
MINFORECASTACCURACY	Minimum forecast accuracy of trained TML model. Choose a number between 0-100, default is 0. A model is selected if it is greater than this value.
MAXPREPROCESSMESSAGES	Number of message for preprocessing. Defaults to 2000. Higher number will consume more energy.
BATCHTHREADS	This is used in batch functions like "viperpreprocessbatch" and indicates how many topicids to preprocess concurrently. For example, if BATCHTHREADS=5, and you are preprocessing 10 topicids in batch, then 5 will be preprocessed concurrently at a time.
MAXPERCMESSAGES	Maximum messages when using Topicid to rollback stream. This is useful when even 1% rollbackback could result in millions of message if your total messages are in the billions. Setting MAXPERCMESSAGES=1000 for example, ensures message are 1000 messages from the last message.
MAXCONSUMEMESSAGES	The amount of message you want Viper to consume. Note consuming a large amount will impact memory and network.
MAADS_ALGORITHM_SERVER_ MICROSERVICE	MAADS algorithm server microservice. This will require MAADS software installed in the host. If you use a reverse proxy to access the MAADS software then specify the name here.
MAADS_ALGORITHM_SERVER1	Additional MAADS algorithm server. You can list up to 10,000 MAADS algorithm servers. Just increment the "SERVER#", where #=1,…,10000
MAADS_ALGORITHM_SERVER1_ PORT	Additional MAADS algorithm server port.

MAADS_ALGORITHM_SERVER1_ MICROSERVICE	Additional MAADS algorithm server microservice.
KAFKA_ROOT	Kafka root folder
HPDE_IP	HPDE (Hyper-Predictions for Edge Devices) is another product required for **Real-Time Machine Learning.** Specify the host where it is installed.
HPDE_PORT	HPDE listening port. Specify port. If you specifying port range use "startport:endport", where start port and end port are numbers
VIPER_IP	Specify IP for Viper, use * or leave empty for Viper to choose.
VIPER_PORT	Specify port. If you specifying port range use "startport:endport", where start port and end port are numbers
VIPERVIZ_IP	Specify IP for Viperviz, use * or leave empty for Viper to choose.
VIPERVIZ_PORT	Specify port. If you specifying port range use "startport:endport", where start port and end port are numbers
SSL_CLIENT_CERT_FILE	SSL certificate file needed if Kafka is SSL/TLS enabled
SSL_CLIENT_KEY_FILE	SSL certificate key store file needed if Kafka is SSL/TLS enabled
SSL_SERVER_CERT_FILE	SSL certificate server key file needed if Kafka is SSL/TLS enabled
CLOUD_USERNAME	SASL_PLAIN username to connect to Confluent Cloud
CLOUD_PASSWORD=	SASL_PLAIN password to connect to Confluent Cloud
MAILSERVER	SMTP mailserver host name for sending emails. This is needed if using **AiMS Dashboard** to monitor algorithms in Kafka.
MAILPORT	SMTP mailserver port for sending emails. This is needed if using **AiMS Dashboard** to monitor algorithms in Kafka.
FROMADDR	From address to put in the emails. This is needed if using **AiMS Dashboard** to monitor algorithms in Kafka.
SMTP_USERNAME	SMTP username. This is needed if using **AiMS Dashboard** to monitor algorithms in Kafka.
SMTP_PASSWORD	SMTP password. This is needed if using **AiMS Dashboard** to monitor algorithms in Kafka and alerts are turned on.
SMTP_SSLTLS	Mailserver SSL/TLS enabled: true of false. This is needed if using **AiMS Dashboard** to monitor algorithms in Kafka and alerts are turned on.
SERVICE_USERNAME	If using ServiceNow, specify the ServiceNoew web page login username. This is needed if using **AiMS Dashboard** to monitor algorithms in Kafka and alerts are turned on.
SERVICE_PASSWORD	If using ServiceNow, specify the ServiceNoew web page login password. This is needed if using **AiMS Dashboard** to monitor algorithms in Kafka and alerts are turned on.
SERVICE_ASSIGNEE	If using ServiceNow, specify the ServiceNow the name to assign the ServiceNow ticket to. This is needed if using **AiMS Dashboard** and Alerts are turned on.
SERVICE_FORM_FIELDS	{"key1":"Assignedto","key2":"LastReadofTopic","key3":"Consumerid", "key4":"Brokerhost","key5":"Brokerport","key6":"Companyname", "key7":"Contactemail","key8":"Contactname","key9":"Description", "key10":"Location","key11":"Topic","key12":"Priority","key13":"Produc erid","key14":"LastWritetoTopic"} Users should replace the "Key" values with the names of the fields in the ServiceNow Form. VIPER will update the key values when submitting the incident to ServiceNow. This is needed if using **AiMS Dashboard** and Alerts are turned on.
SERVICE_CONTENTTYPE=applic ation/json	ServiceNow webpage content type. This can be changed but **application/json** should be fine. This is needed if using **AiMS Dashboard** and Alerts are turned on.
POLLING_ALERTS	Polling for alerts in minutes. This is needed if using **AiMS Dashboard** and Alerts are turned on. VIPER will poll for alerts and wait in minutes for the next poll.
COMPANYNAME	Specify company name. This is used when sending emails from AiMS dashboard.
MYSQLDRIVERNAME	Enter MySQL driver name i.e. mysql
MYSQLDB	Enter MySQL DB name
MYSQLUSER	Enter MySQL username
MYSQLPASS	Enter MySQL password
MYSQLHOSTNAME	Enter MySQL hostname – ***If using MYSQL DOCKER set this to: host.docker.internal:3306***
MYSQLMAXLIFETIMEMINUTES	Enter max lifetime in minutes

MYSQLMAXCONN	Enter maximum connections
MYSQLMAXIDLE	Enter number of idle connections
MYSQL_ROOT_PASSWORD	MYSQL DOCKER Container: Set the Root password for MySQL
MYSQL_ROOT_HOST	MYSQL DOCKER Container: Set the Root host for MySQL ie. You can use % to accept connections from any host.
MYSQL_DATABASE	MYSQL DOCKER Container: Set the database name i.e. tmlids – ***This should match MYSQLDB***
MYSQL_USER	MYSQL DOCKER Container: Set the username name i.e. tmluser, avoid "root" - ***This should match MYSQLUSER***
MYSQL_PASSWORD	MYSQL DOCKER Container: Set the password - ***This should match MYSQLPASS***
MAXURLQUERYSTRINGBYTES	This is the size of the URL query string in bytes, if using viperhpdepredictprocess

3) AiMS (Algorithms and Insights Management System) Integration

To manage algorithms and insights across brokers at scale, OTICS provides an **out of the box dashboard** to allow VIPER admins to track every algorithm that producers are producing to, and consumers are consuming from. AiMS can even integrate with ServiceNow and your mail system to send alerts when:

i. No one is consuming from a topic and send you a list of consumers via email
ii. **Auto deactivate** consumers if they have not consumed for x number of days
iii. **Auto deactivate** consumers if they have not consumed for x number of days and auto create a ticket in ServiceNow
iv. Auto deactivate topics/algorithms when no is consuming from them

Having the ability to automatically activate/deactivate topics/algorithms and consumers can result in a reduction in cloud spend by **better controlling algorithms and their use of compute and storage and lead to better use of business resources.**

With VIPER, Apache Kafka and ServiceNow businesses get a truly enterprise system that can scale to unlimited amount of data and algorithms, while ensuring service management levels are maintained and cloud compute and storage costs are tightly controlled.

To integrate AiMS with your **Mail server** you need to specify in VIPER.ENV:

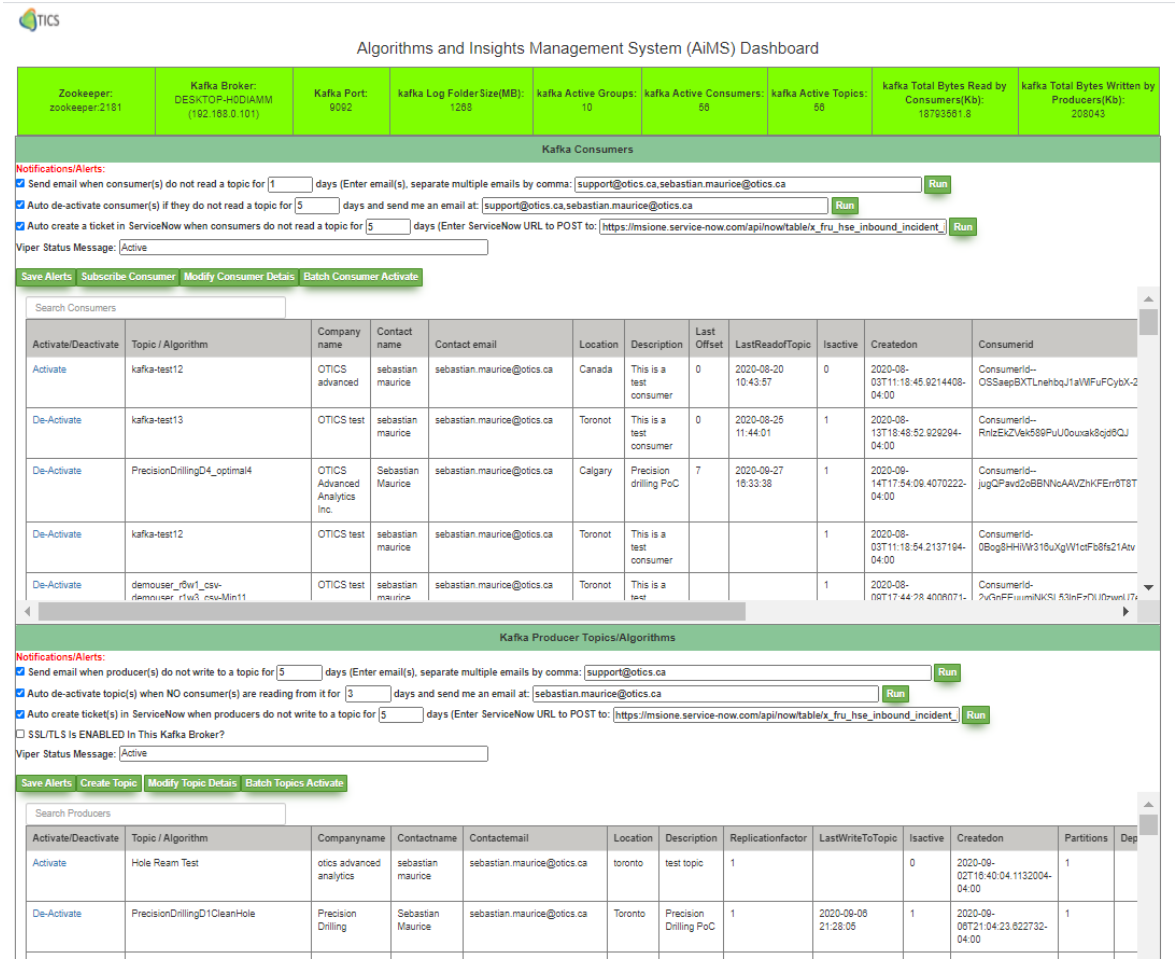
```
MAILSERVER=<mail server>
MAILPORT=<port
FROMADDR=<from address>
SMTP_USERNAME=<smtp username>
SMTP_PASSWORD=<smtp password>
SMTP_SSLTLS=<true/false>
```

To integrate AiMS with **ServiceNow** enter the following in VIPER.ENV:

```
SERVICE_USERNAME=<ServiceNow website username>
SERVICE_PASSWORD=<ServiceNow website password>
SERVICE_ASSIGNEE=<account name to assign ticket to>
SERVICE_FORM_FIELDS={"key1":"Assignedto","key2":"LastReadofTopic","key3":"Consumerid","key4":"Brokerhost","key5":"Brokerport","key6":"Companyname","key7":"Contactemail","key8":"Contactname","key9":"Description","key10":"Location","key11":"Topic","key12":"Priority","key13":"Producerid","key14":"LastWritetoTopic"}
SERVICE_CONTENTTYPE=application/json
```

The Key values are the names of the ServiceNow form fields. You can specify any name for key values, when VIPER creates a ticket it will automatically assign values to these keys and POST the form to the ServiceNow url that you specify in the AiMS dashboard. The value of SERVICE_ASSIGNEE will be the value for Key1.

Figure 1: AiMS Dashboard



You can start VIPER to poll for Alerts from AiMS by entering in VIPER.ENV:

POLLING_ALERTS=<number of minutes to wait to poll for alerts>

4) **You are done! Start VIPER.**

B. Additional Documentation for Accessing VIPER Functionality

1) VIPER is accessed by two methods:
   a. MAADSTML python library: https://pypi.org/project/maadstml/
      i. Scroll down to: **MAADS-VIPER Connector to Manage Apache KAFKA:**

   b. REST API:
      i. When starting VIPER type "Help" to see all the REST endpoints
      ii. The endpoints can be called from ANY programming language.

2) Users can send an email to support@otics.ca for additional help with any of the functions – add **"VIPER HELP" to the subject line** (no quotes)**.**

3) OTICS provides up to **2 hours free virtual training** on an as-needed basis for clients or groups of clients.

OTICS Advanced Analytics Inc.

For On-Premise TML Kafka Deployments:

Below are suggested configurations – some fields may differ or may not apply

Server environment:zookeeper.version=3.6.1--104dcb3e3fb464b30c5186d229e00af9f332524b, built on 04/21/2020 15:01 GMT

Server environment:java.version=1.8.0_144

Server.properties

allow.everyone.if.no.acl.found=true
auto.create.topics.enable=false
broker.id=0
listeners=PLAINTEXT://127.0.0.1:9092
advertised.listeners=PLAINTEXT://127.0.0.1:9092

# Maps listener names to security protocols, the default is for them to be the same. See the config documentation for more details
listener.security.protocol.map=PLAINTEXT:PLAINTEXT,SSL:SSL,SASL_PLAINTEXT:SASL_PLAINTEXT,SASL_SSL:SASL_SSL
# The number of threads that the server uses for receiving requests from the network and sending responses to the network
num.network.threads=3
# The number of threads that the server uses for processing requests, which may include disk I/O
num.io.threads=8
# The send buffer (SO_SNDBUF) used by the socket server
socket.send.buffer.bytes=902400
# The receive buffer (SO_RCVBUF) used by the socket server
socket.receive.buffer.bytes=902400
# The maximum size of a request that the socket server will accept (protection against OOM)
socket.request.max.bytes=969295616
zookeeper.connect=localhost:2181
num.partitions=1
num.recovery.threads.per.data.dir=1
log.flush.interval.messages=30000000
log.flush.interval.ms=1800000
log.retention.minutes=30
log.segment.bytes=1073741824
log.retention.check.interval.ms=300000
delete.topic.enable=true
offsets.topic.replication.factor=1
transaction.state.log.replication.factor=1
transaction.state.log.min.isr=1

zookeeper.properties:

```
# contributor license agreements.  See the NOTICE file distributed with
# this work for additional information regarding copyright ownership.
# The ASF licenses this file to You under the Apache License, Version 2.0
# (the "License"); you may not use this file except in compliance with
# the License.  You may obtain a copy of the License at
#
#    http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
# the directory where the snapshot is stored.
dataDir=/tmp/zookeeper
# the port at which the clients will connect
clientPort=2181
# disable the per-ip limit on the number of connections since this is a non-production config
maxClientCnxns=0
# Disable the adminserver by default to avoid port conflicts.
# Set the port to something non-conflicting if choosing to enable this
#admin.enableServer=true
#admin.serverPort=8080
#authProvider.1=org.apache.zookeeper.server.auth.SASLAuthenticationProvider
requireClientAuthScheme=plain
jaasLoginRenew=3600000
```

producer.properties:

```
bootstrap.servers=localhost:9092
security.protocol=SASL_PLAINTEXT
sasl.mechanism=PLAIN
zookeeper.connect=localhost:2181
```

consumer.properties:

```
security.protocol=SASL_PLAINTEXT
sasl.mechanism=PLAIN
zookeeper.connect=localhost:2181
zookeeper.connection.timeout.ms=6000
group.id=test-consumer-group
```

Add to Java.Env in zookeeper/conf: ***(Note: you may need to create this file using your text editor.)***

```
SERVER_JVMFLAGS=-Djava.security.auth.login.config=C:/CORE_FILES/zookeeper/kafka/config/zookeeper_jaas.conf

CLIENT_JVMFLAGS=-Djava.security.auth.login.config=C:/CORE_FILES/zookeeper/kafka/config/ kafka_server_jaas.conf
```

zookeeper_jaas.conf: ***(Note: you may need to create this file using your text editor.)***

```
Server {
 org.apache.kafka.common.security.plain.PlainLoginModule required
  username="tmladmin"
  password="tmluser!?123"
  user_tmladmin="tmluser!?123"
  user_tmluser="tmluser!?123";
};

QuorumServer {
     org.apache.kafka.common.security.plain.PlainLoginModule required
     username="tmladmin"
     password="tmluser!?123";
};

QuorumLearner {
     org.apache.kafka.common.security.plain.PlainLoginModule required
     username="tmladmin"
     password="tmluser!?123";
};
```

kafka_server_jaas.conf: ***(Note: you may need to create this file using your text editor.)***

```
KafkaServer {
  org.apache.kafka.common.security.plain.PlainLoginModule required
   username="tmladmin"
   password="tmluser!?123"
   user_tmladmin="tmluser!?123";
};

Client {
  org.apache.kafka.common.security.plain.PlainLoginModule required
  username="tmladmin"
  password="tmluser!?123";
};
```

Terminal 1 (start Zookeeper server)

From kafka root directory

Linux:

```
$ export KAFKA_OPTS="-Djava.security.auth.login.config=/home/usename/Documents/kafka_2.11-0.10.1.0/config/zookeeper_jaas.conf"
$ bin/zookeeper-server-start.sh config/zookeeper.properties
```

Windows:

```
SET KAFKA_OPTS=-Djava.security.auth.login.config=C:\CORE_FILES\zookeeper\kafka\config\zookeeper_jaas.conf
```

Start Zookeeper:

```
zookeeper-server-start.bat C:/CORE_FILES/zookeeper/kafka/config/zookeeper.properties
```

Terminal 2 (start Kafka server)

From kafka root directory

Linux:
$ export KAFKA_OPTS="-Djava.security.auth.login.config=/home/usename/Documents/kafka_2.11-0.10.1.0/config/kafka_server_jaas.conf"
$ bin/kafka-server-start.sh config/server.properties

Windows:

SET KAFKA_OPTS=-Djava.security.auth.login.config=C:\CORE_FILES\zookeeper\kafka\config\kafka_server_jaas.conf
Start Kafka Server:
kafka-server-start.bat C:/CORE_FILES/zookeeper/kafka/config/server.properties
kafka_client_jaas.conf
KafkaClient {
 org.apache.kafka.common.security.plain.PlainLoginModule required
  username="tmladmin"
  password=" tmluser!?123";
};

KAFKA CLIENT:
kafka_client_jaas.conf
KafkaClient {
org.apache.kafka.common.security.plain.PlainLoginModule required
  username="tmladmin"
  password=" tmluser!?123";
};

Terminal 3 (start Kafka consumer)
On a client terminal, export client jaas conf file and start consumer:
$ export KAFKA_OPTS="-Djava.security.auth.login.config=/home/username/Documents/kafka_2.11-0.10.1.0/kafka_client_jaas.conf"

Create a Topic:
$ bin/kafka-topics.sh --create --partitions 1 --replication-factor 1 --topic quickstart-events --bootstrap-server localhost:9092

Terminal 4 (start Kafka producer)
If you also want to produce, do this on another terminal window:
$ export KAFKA_OPTS="-Djava.security.auth.login.config=/home/username/Documents/kafka_2.11-0.10.1.0/kafka_client_jaas.conf"

Produce to the Topic:
$ ./bin/kafka-console-producer.sh --broker-list localhost:9092 --topic quickstart-events --producer.config=config/producer.properties

Consume from the Topic:
$ ./bin/kafka-console-producer.sh --topic quickstart-events --from-beginning --bootstrap-server localhost:9092

Note: If Kafka broker complains about clusterID then delete: meta.properties in kafka/kafka-logs and restart broker.

OTICS Advanced Analytics Inc.

TML On-Prem Kafka Running on Linux (Ubuntu): Shell Script

This script below is an example you will need to modify the file paths according to your setup but core components that are needed to run TML technologies with Kafka are listed

```
#!/bin/bash

gnome-terminal -- bash -c "apt-get -y update; apt install default-jdk;sleep 5;cp -r /isodevice/zookeeper /home;sleep 5;cp -r
/isodevice/viper /home;cp -r /isodevice/pythonfiles /home;sleep 10;cp -r /isodevice/hpde /home;chmod -R 777 /home/viper;chmod
-R 777 /home/hpde;chmod -R 777 /home/pythonfiles;chmod -R 777 /home/zookeeper;apt install python3.8;apt-get install
python3-setuptools;python3 -m easy_install install pip; pip install maadstml;pip install joblib;cd /home;cd
zookeeper/kafka/bin;sleep 5;export KAFKA_OPTS=-
Djava.security.auth.login.config=/home/zookeeper/kafka/config/zookeeper_jaas.conf;sleep 2;kill -9 `sudo lsof -t -
i:2181`;./zookeeper-server-start.sh ../config/zookeeper.properties; exec bash"

if [[ $(java -version 2>&1 | grep "OpenJDK Runtime") ]]; then sleep 30; else sleep 120;
fi

gnome-terminal -- bash -c "cd /home;cd zookeeper/kafka/bin;export KAFKA_OPTS=-
Djava.security.auth.login.config=/home/zookeeper/kafka/config/kafka_server_jaas.conf;sleep 2;kill -9 `sudo lsof -t -
i:9092`;./kafka-server-start.sh ../config/server.properties; exec bash"

sleep 10

gnome-terminal -- bash -c "kill -9 `sudo lsof -t -i:8000`;cd /home;cd viper;./viper-linux-amd64 127.0.0.1 8000;exec bash"

sleep 10

gnome-terminal -- bash -c "kill -9 `sudo lsof -t -i:8001`;cd /home;cd hpde;./hpde-linux-amd64 127.0.0.1 8001;exec bash"

gnome-terminal -- bash -c "cd /home;exec bash"
```