

Performance Testing for Kafka Topics

❖ Introduction:

Apache Kafka is a powerful event streaming platform used for real-time data processing and communication. To ensure its reliability and effectiveness, rigorous performance testing is essential. Following are the key aspects -

1. Testing Strategies:

- Unit Testing: Validates the behaviour of Kafka Streams applications.
- Integration Testing: Ensures seamless interaction with other components.
- **Load Testing: Evaluates scalability, throughput, and fault resilience.**

2. The Role of Topology Test Driver:

- **Simulates an In-Memory Kafka Cluster:**
 - Eliminates the need for an actual Kafka cluster during testing.
 - Simplifies unit testing by avoiding dedicated Kafka cluster setup.
- **Testing Kafka Streams Topologies:**
 - Processes simulated records through transformations.
 - Verifies expected outcomes.
- **Result Verification:**
 - Ensures correct functionality (e.g., filtering, transformation).
- **Isolated Testing:**
 - Conduct tests without affecting production or integration clusters.
 - Maintains system integrity and stability.

❖ Objective & Purpose:

1. Guarantee Dependability:

- Verify data integrity and system behaviour.
- Ensure Kafka Streams applications function correctly.

2. Evaluate Scalability:

- Measure throughput and latency under varying workloads.
- Optimize Kafka configurations for peak performance.

3. Assess Resilience to Faults:

- Test fault tolerance mechanisms.
- Validate system behaviour during failures.

4. Capacity Planning:

- Understand Kafka's performance on specific hardware.
- Calculate sizing based on message sizes, partitions, and network speeds.

Remember, robust and reliable real-time data processing solutions depend on thorough Kafka performance testing!

❖ **Pre-Requisite Setup:**

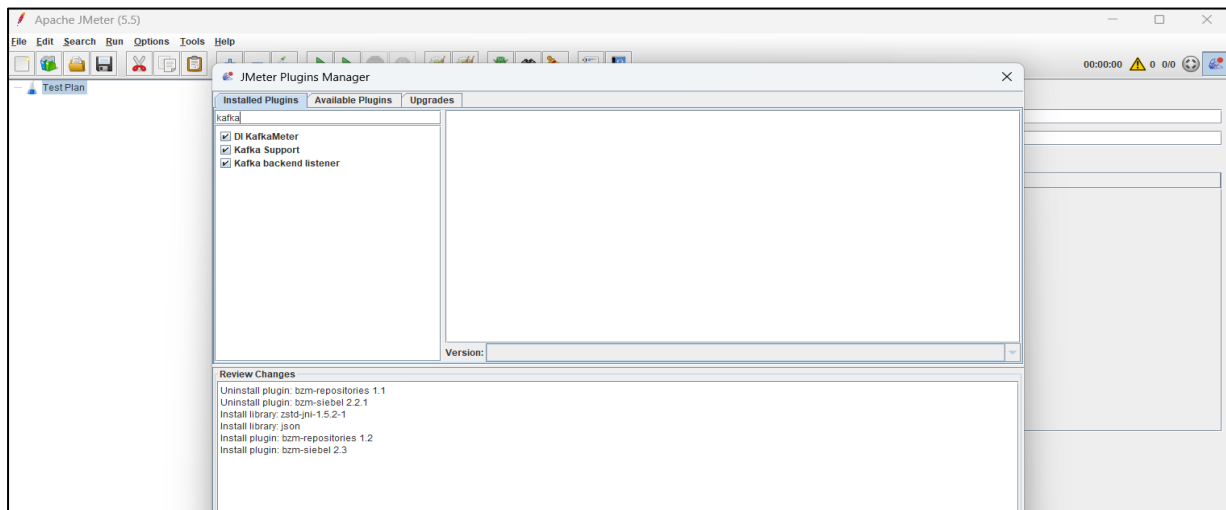
- Kafka Server should be downloaded & Installed in Virtual Machine/Physical Machine/Cloud Platform.
- Following kafka services should be up and running.
 - Zookeeper Server
 - Kafka Server
 - Topic should be created.
 - Producer server
 - Consumer Server
- E2E Kafka servers (message streaming) should be working.

Note:-

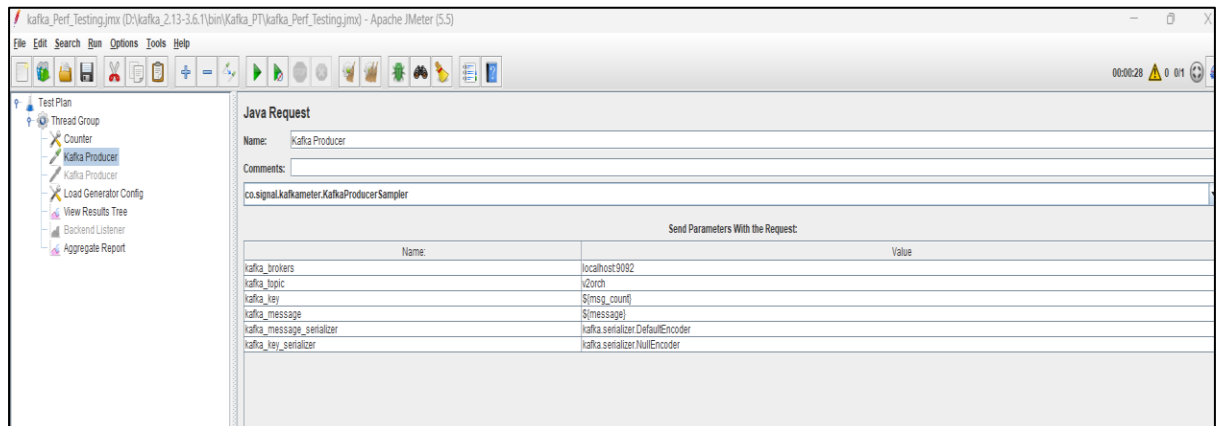
If setup is not completed/ in progress/ facing some issue while configuration. Please refer the installation Setup steps for Kafka Server. Details are shared below, which includes following pointers - “Download, Installation & how to start Kafka servers”.

❖ **Performance Test:**

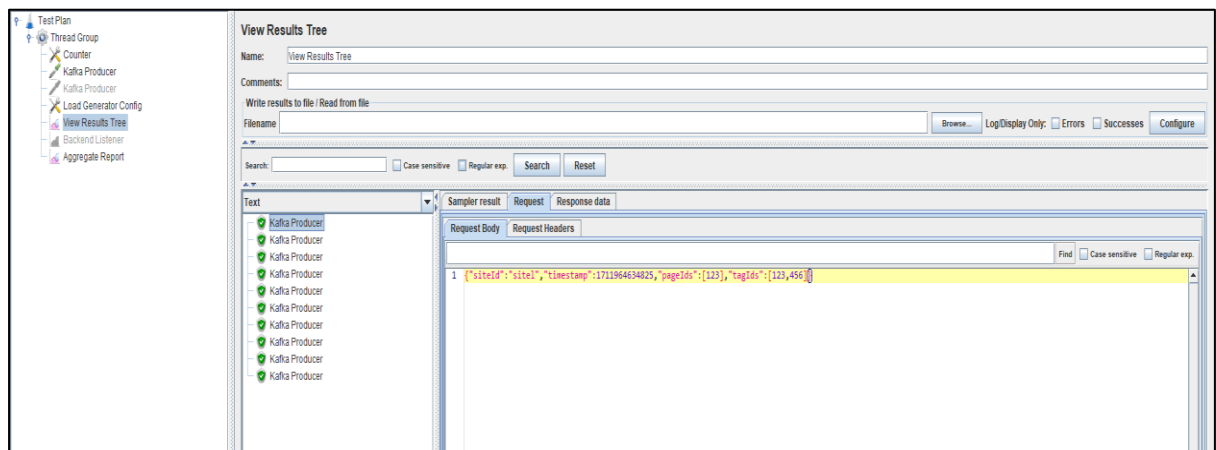
- **To validate/test Kafka Server using JMeter (Performance Tool)**
 - Download & install the JMeter tool.
 - Install JMeter Plugin Manager and integrate with JMeter tool.
 - Open the JMeter GUI.
 - Install the following plugins related to Kafka in JMeter
 1. DI Kafka Meter
 2. Kafka Support
 3. Kafka Backend Listener



- Create the script in JMeter as shown below -



- Performed sanity test to make sure messages were successfully published.



- Navigate to consumer server to validate message got successfully delivered.

```
PS D:\kafka_2.13-3.6.1> .\bin\windows\kafka-console-consumer.bat --bootstrap-server localhost:9092 --topic v2orch
Hi
Testing message
message sent from publisher
{"siteId":"site1","timestamp":1711964553132,"pageIds":[],"tagIds":[]}
{"siteId":"site1","timestamp":1711964634825,"pageIds":[123],"tagIds":[123,456]}
{"siteId":"site1","timestamp":1711964634841,"pageIds":[],"tagIds":[]}
{"siteId":"site2","timestamp":1711964634855,"pageIds":[],"tagIds":[]}
{"siteId":"site1","timestamp":1711964634861,"pageIds":[],"tagIds":[]}
{"siteId":"site1","timestamp":1711964634865,"pageIds":[123],"tagIds":[123,456]}
{"siteId":"site2","timestamp":1711964635335,"pageIds":[123],"tagIds":[123,234]}
{"siteId":"site2","timestamp":1711964635335,"pageIds":[],"tagIds":[]}
{"siteId":"site1","timestamp":1711964635335,"pageIds":[123],"tagIds":[123,456]}
{"siteId":"site2","timestamp":1711964635353,"pageIds":[123],"tagIds":[123,234]}
{"siteId":"site1","timestamp":1711964635355,"pageIds":[123],"tagIds":[123,456]}
```

❖ Installation Setup For Kafka Server:

1. Downloaded the Kafka setup files from below link -

<https://kafka.apache.org/downloads>

2. Extracted the zip file into certain folder.

❖ Start the following servers as instructed below:

➤ **Started Zookeeper Server:**

- **Purpose** => Zookeeper is used to track the status of nodes in the kafka cluster and maintain a list of Kafka topic and messages.
- **Command** => `.\bin\windows\zookeeper-server-start.bat .\config\zookeeper.properties`

```
[2024-04-01 13:38:06,423] INFO clientPortAddress is 9.0.0.0:2181 (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-04-01 13:38:06,427] INFO secureClientPort is not set (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-04-01 13:38:06,423] INFO observerMasterPort is not set (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-04-01 13:38:06,423] INFO metricsProvider.className is org.apache.zookeeper.metrics.impl.DefaultMetricsProvider (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-04-01 13:38:06,423] INFO Starting server (org.apache.zookeeper.server.ZooKeeperServerMain)
[2024-04-01 13:38:06,447] INFO ServerMetrics initialized with provider org.apache.zookeeper.metrics.impl.DefaultMetricsProvider@99f6bfdd (org.apache.zookeeper.server.ServerMetricsImpl)
[2024-04-01 13:38:06,457] INFO ACL digest algorithm is: SHA1 (org.apache.zookeeper.server.auth.DigestAuthenticationProvider)
[2024-04-01 13:38:06,457] INFO zookeeper.DigestAuthenticationProvider.enabled = true (org.apache.zookeeper.server.auth.DigestAuthenticationProvider)
[2024-04-01 13:38:06,462] INFO zookeeper.snapshot.trust.empty : false (org.apache.zookeeper.server.persistence.FileTxnSnapshot)
[2024-04-01 13:38:06,480] INFO (org.apache.zookeeper.server.ZooKeeperServer)
[2024-04-01 13:38:06,480] INFO (org.apache.zookeeper.server.ZooKeeperServer)
[2024-04-01 13:38:06,480] INFO (org.apache.zookeeper.server.ZooKeeperServer)
[2024-04-01 13:38:06,480] INFO (org.apache.zookeeper.server.ZooKeeperServer)
[2024-04-01 13:38:06,480] INFO (org.apache.zookeeper.server.ZooKeeperServer)
[2024-04-01 13:38:06,480] INFO (org.apache.zookeeper.server.ZooKeeperServer)
[2024-04-01 13:38:06,480] INFO (org.apache.zookeeper.server.ZooKeeperServer)
[2024-04-01 13:38:06,480] INFO (org.apache.zookeeper.server.ZooKeeperServer)
[2024-04-01 13:38:06,480] INFO (org.apache.zookeeper.server.ZooKeeperServer)
[2024-04-01 13:38:06,763] INFO Server environment:zookeeper.version=3.8.6-Gadgd36d7c70bcfded52d54ebefa395809baab56, built on 2023-10-05 18:34 UTC (org.apache.zookeeper.server.ZooKeeperServer)
[2024-04-01 13:38:06,763] INFO Server environment:host.name=10.26.146.214 (org.apache.zookeeper.server.ZooKeeperServer)
[2024-04-01 13:38:06,763] INFO Server environment:jvm.vendor=Oracle Corporation (org.apache.zookeeper.server.ZooKeeperServer)
[2024-04-01 13:38:06,763] INFO Server environment:java.home=C:\Program Files\Java\jre-1.8 (org.apache.zookeeper.server.ZooKeeperServer)
[2024-04-01 13:38:06,763] INFO Server environment:java.class.path=D:\Kafka_2.13-3.6.1\libs\activation-1.1.1.jar;D:\Kafka_2.13-3.6.1\libs\aopalliance-repackaged-2.6.1.jar;D:\Kafka_2.13-3.6.1\libs\argparse4j-0.7.0.jar;D:\Kafka_2.13-3.6.1\libs\audience-annotations-0.12.0.jar;D:\Kafka_2.13-3.6.1\libs\caffeine-2.9.3-jar-with-dependencies.jar;D:\Kafka_2.13-3.6.1\libs\checker-qual-3.19.0.jar;D:\Kafka_2.13-3.6.1\libs commons-beanutils-1.9.4.jar;D:\Kafka_2.13-3.6.1\libscommons-cli-1.4.jar;D:\Kafka_2.13-3.6.1\libs commons-codec-1.15.jar;D:\Kafka_2.13-3.6.1\libs commons-digester-2.1.jar;D:\Kafka_2.13-3.6.1\libs commons-io-2.11.0.jar;D:\Kafka_2.13-3.6.1\libs commons-lang3-3.8.1.jar;D:\Kafka_2.13-3.6.1\libs commons-logging-1.2.jar;D:\Kafka_2.13-3.6.1\libs commons-validator-1.7.jar;D:\Kafka_2.13-3.6.1\libs connect-api-3.6.1.jar;D:\Kafka_2.13-3.6.1\libsconnect-basic-auth-extension-3.6.1.jar;D:\Kafka_2.13-3.6.1\libsconnect-info-3.6.1.jar;D:\Kafka_2.13-3.6.1\libs
```

➤ **Started Kafka Server:**

- **Purpose** => Kafka server used to build real-time streaming data pipelines and applications that adapt to the data streams.
- **Command** => `.\bin\windows\kafka-server-start.bat .\config\server.properties`

[illegible]

➤ Created topic:

- Topic Name – v2orch
- **Command** => .\bin\windows\kafka-topics.bat --create --topic v2orch --bootstrap-server localhost:9092 --partitions 1 --replication-factor 1

➤ **Started Producer Server:**

- **Purpose** => Producer is a client application that publishes events to a kafka cluster.
- **Command** => `.bin\windows\kafka-console-producer.bat --broker-list localhost:9092 --topic v2orch`

```
PS D:\kafka_2.13-3.6.1> .\bin\windows\kafka-console-producer.bat --broker-list localhost:9092 --topic v2orch
>Hi
>Testing message
>message sent from publisher
>
```

- **Kafka Broker:**

- **Purpose** => While the producer shall be pushing the message into the kafka cluster, its kafka broker that helps to transfer the message from the producer to a consumer.

- **Started Consumer Server:**

- **Purpose** => Consumer is a client application that subscribe events
- **Command** => `.\bin\windows\kafka-console-consumer.bat --bootstrap-server localhost:9092 --topic v2orch`

```
PS D:\kafka_2.13-3.6.1> .\bin\windows\kafka-console-consumer.bat --bootstrap-server localhost:9092 --topic v2orch
Hi
Testing message
message sent from publisher
```

➤ **To validate E2E setup:**

- Send out any message from the **producer server** as shown in snapshot. Same message should be displayed in **Consumer server** (refer the above snapshot for more details).
- Once message is displayed/received in consumer than E2E setup has been successfully completed.