Sri Indu
College of Engineering & Technology
UGC Autonomous Institution
Recognized under 2(f) & 12(B) of UGC Act 1956,
NAAC, Approved by AICTE &
Permanently Affiliated to JNTUH
Estd.2001

SKILL DEVELOPMENT COURSE

(ETL-KAFKA/TALEND) LAB MANUAL

III Year–I Semester

DEPARTMENT OF ARTIFICIAL INTELLEGENCE
AND DATA SCIENCE

ACADEMIC YEAR 2024-25

SRI INDU COLLEGE OF ENGINEERING & TECHNOLOGY
(An Autonomous Institution under UGC, New Delhi)
Recognized under 2(f) and 12(B) of UGC Act 1956
NBA Accredited, Approved by AICTE and Permanently affiliated to JNTUH
Sheriguda (V), Ibrahimpatnam, R.R.Dist, Hyderabad - 501 510

DEPARTMENT OF

COMPUTER SCIENCE ENGINEERING(ARTIFICIAL INTELLIGENCE AND DATA SCIENCE)

LAB MANUAL

Branch: CSE(AI&CS)	**Class: B.Tech- III Year-I sem**
Subject: Computer Networks Lab	**Code:R22CSD3241**
Academic Year: 2023-24	**Regulation: R22**
Core/Elective/H&S: Core	**Credits: 1.5**

Prepared By
Name: MISS.T.GLORY

Verified By
Head of the Department:

SRI INDU COLLEGE OF ENGINEERING & TECHNOLOGY
B. TECH –COMPUTER SCIENCE AND ENGINEERING(CS)

INSTITUTION VISION

To be a premier Institution in Engineering & Technology and Management with competency, values and social consciousness.

INSTITUTION MISSION

IM1 Provide high quality academic programs, training activities and research facilities.

IM2 Promote Continuous Industry-Institute interaction for employability, Entrepreneurship, leadership and research aptitude among stakeholders.

IM3 Contribute to the economical and technological development of the region, state and nation.

DEPARTMENT VISION

To be a Technologically adaptive centre for computing by grooming the students as top notch professionals.

DEPARTMENT MISSION

The Department has following Missions:

DM1 To offer quality education in computing.

DM2 To provide an environment that enables overall development of all the stakeholders.

DM3 To impart training on emerging technologies like data analytics , artificial intelligence and internet of things.

DM4 To encourage participation of stake holders in research and development.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PEO1: **Higher Studies:** Graduates with an ability to pursue higher studies and get employment in reputed institutions and organizations.

PEO2: **Domain knowledge:** Graduate with an ability to design and develop a product.

PEO3: **Professional Career:** Graduate with an ability to design and develop a product.

PEO4: **Life Long Learning:** Graduate with an ability to learn advanced skills to face professional competence through lifelong learning.

PROGRAMOUTCOMES(POs)

PO	Description
PO 1	**Engineering Knowledge:** To be able to apply knowledge of computing, mathematics, Science and Engineering appropriate to the discipline
PO 2	**Problem Analysis:** To be able identify, formulate & analyze a problem, and ascertain and define the computing requirements appropriate to its solution.
PO 3	**Design & Development Solutions:** To be able to design, implement, and evaluatea computer-based system, process, component, or program to meet desired needs.
PO 4	**Investigation of complex problems:** To be able to identify and analyze user needs And consider them in the selection, creation, evaluation and administration of computer-based systems for providing valid solutions to complex problems.
PO 5	**Modern Tool Usage:** To posses skills for creating and in using contemporary techniques, skills, and tools necessary for computing Practice.
PO 6	**Engineering&Society:**Toapplyconceptualknowledgerelevanttoprofessionalengineeringpracticesinsocietal,health,safety,legalandculturalissuesandtheirconsequences
PO 7	**Environment &Sustainability:** To be able to Analyze the local and global impact of computing on individuals, organizations, and society and work towards sustainable development.
PO 8	**Ethics:** To understand contemporary professional, ethical, legal, security and social issue sand responsibilities.
PO 9	**Individual & Team work:** To Be able to function effectively as an individual andon teams to accomplish a common goal.
PO 10	**Communication:** To communicate precisely and effectively both in oral andwritten form with a range of audiences.
PO 11	**Project management & finance:** To apply engineering and management principles For managing and leading economically feasible projects in multi-disciplinaryenvironments with an effective project plan.
PO 12	**Life Long Learning:** To recognize the need for and an ability to engage in independent& lifelong learning for continuing professional development.
Program Specific Outcomes	
PSO 1	Develop software projects using standard practices and suitable programming environment.
PSO 2	Identify , formulate and solve the real life problems faced in the society, industry and other areas by applying the skills of the programming languages, networks and databases learned.
PSO 3	To apply computer science knowledge in exploring and adopting latest technologies in different co-curricular activities.

COURSE OUTCOMES

C216.1	Implement data link layer farming methods and Analyze error detection and error correction codes.
C216.2	Implement and analyze routing and congestion issues in network design.
C216.3	Implement Encoding and Decoding techniques used in presentation layer.

COsMAPPINGWITHPOs&PSOs

Course Outcome	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO11	PO 12	PSO 1	PSO2	PSO3
C216.1	2	2	3	2	2	-	-	-	-	1	-	-	1	2	1
C216.2	2	1	2	1	2	-	-	-	-	-	2	-	1	1	1
C216.3	1	2	1	2	1	-	-	--	-	-	-	-	2	1	1
C216	**1.6**	**1.6**	**2.0**	**1.6**	**1.6**	**-**	**-**	**-**	**-**	**0.3**	**0.6**	**-**	**1.3**	**1.3**	**1.16**

List of Experiments

SNO	PROGRAMS
1	**Week 1-:**Installing and Running Kafka.
2	**Week 2 -:** Configuring Multi-Broker Kafka
3	**Week 3 - :** Exploring Zookeeper.
4	**Week 4 - :** Overview of kafka consumers
5	**Week 5- :** Using Kafka with Docker..
6	**Week 6- :** Developing Kafka Producer.
7	**Week 7- :** Sending Messages with Callback.
8	**Week 8- :**Kafka monitoring and schema Registry.
9	**Week 9- :** Kafka streams and kafka connectors.
10	**Week 10- :** Integration of kafka with storm.
11	**Week 11- :** Kafka integration with spark and flume.
12	**Week 12- :** Kafka Application as Consumer and Producer.
13	**Week 13- :** Word Count Per Record

INTRODUCTION TO KAFKA COMPONENTS

TOPICS

- A topic is a feed name or category to which records are published.
- Topics in Kafka are always multi-subscriber that is, a topic can have zero, one, or many consumers that subscribe to the data written to it.
- For each topic, the Kafka cluster maintains a partition log.

PARTITIONS

- A topic may have many partitions so that it can handle an arbitrary amount of data.
- If you create multiple partition for a single topic then the data will be stored randomly on any partition. Which means the data will be received in unordered manner by the consumer.

PARTITION OFFSET

- Each partitioned message has a unique sequence ID called an offset.

REPLICATION FACTOR

Replicas are nothing but backups of a partition. If the replication factor of the topic is set to 4, then Kafka will create four identical replicas of each partition and place them in the cluster to make them available for all its operations. Replicas are never used to read or write data. They are used to prevent data loss.

BROKERS

Brokers are simple systems responsible for maintaining published data. Kafka brokers are stateless, so they use Zookeeper for maintaining their cluster state. Each broker may have zero or more partitions per topic.

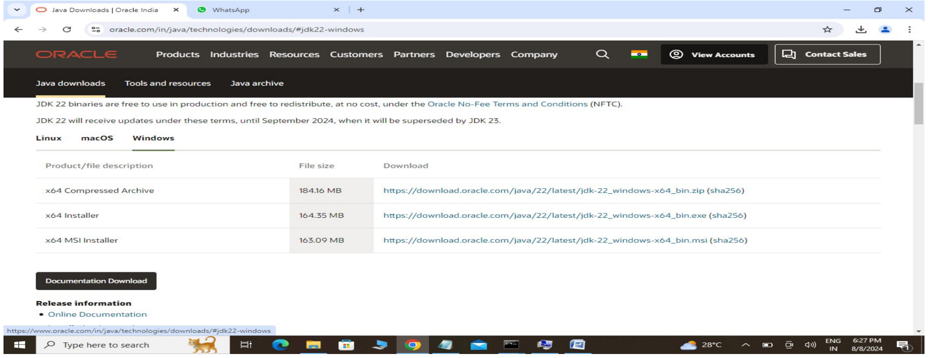
ZOOKEEPER

Zookeeper is used for managing and coordinating Kafka brokers. It is mainly used to notify producers and consumers about the presence or about the failure of any broker in the Kafka system.

EXPERIMENT NO 1

NAME OF THE EXPERIMENT: Installing and Running Kafka.

STEP 1: Check java is installed or not, if not install jdk latest version



JDK 22 binaries are free to use in production and free to redistribute, at no cost, under the Oracle No-Fee Terms and Conditions (NFTC).

JDK 22 will receive updates under these terms, until September 2024, when it will be superseded by JDK 23.

Linux macOS Windows

Product/file description	File size	Download
x64 Compressed Archive	184.16 MB	https://download.oracle.com/java/22/latest/jdk-22_windows-x64_bin.zip (sha256)
x64 Installer	164.35 MB	https://download.oracle.com/java/22/latest/jdk-22_windows-x64_bin.exe (sha256)
x64 MSI Installer	163.09 MB	https://download.oracle.com/java/22/latest/jdk-22_windows-x64_bin.msi (sha256)

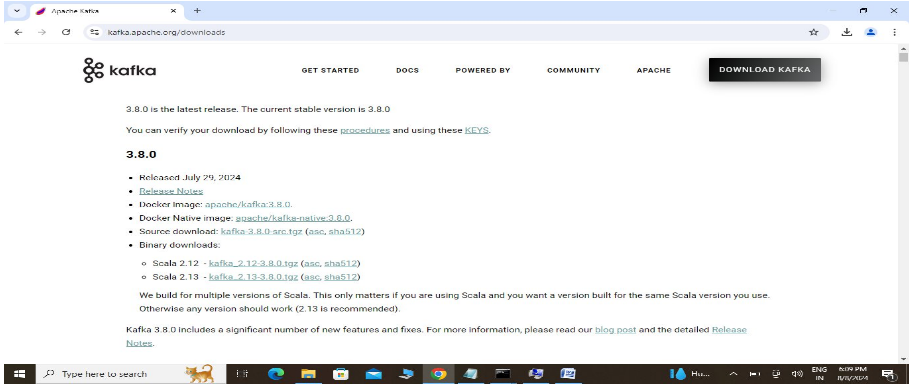
Documentation Download

Release information

- Online Documentation

https://www.oracle.com/in/java/technologies/downloads/#jdk22-windows
Type here to search
28°C
ENG IN
6:27 PM 8/8/2024

STEP 2: Install Apache Kafka



3.8.0 is the latest release. The current stable version is 3.8.0

You can verify your download by following these procedures and using these KEYS.

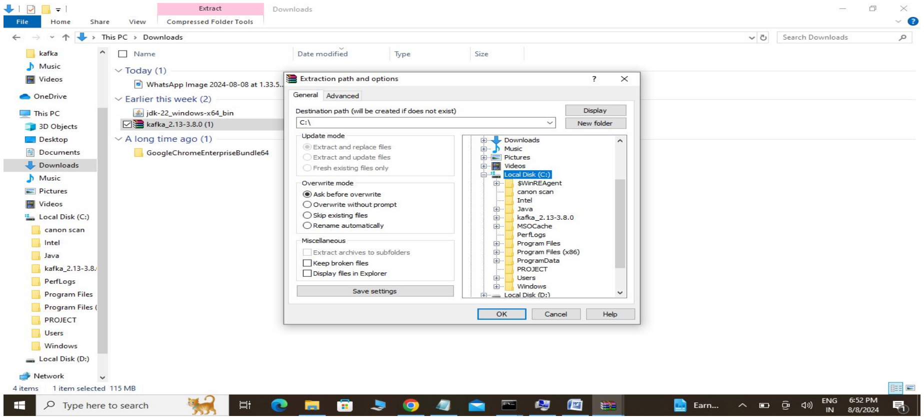
3.8.0

- Released July 29, 2024
- Release Notes
- Docker image: apache/kafka:3.8.0.
- Docker Native image: apache/kafka-native:3.8.0.
- Source download: kafka-3.8.0-src.tgz (asc, sha512)
- Binary downloads:
  - Scala 2.12 - kafka_2.12-3.8.0.tgz (asc, sha512)
  - Scala 2.13 - kafka_2.13-3.8.0.tgz (asc, sha512)

  We build for multiple versions of Scala. This only matters if you are using Scala and you want a version built for the same Scala version you use. Otherwise any version should work (2.13 is recommended).

Kafka 3.8.0 includes a significant number of new features and fixes. For more information, please read our blog post and the detailed Release Notes.

STEP 3: Extract Apache Kafka into C drive



Today (1)
WhatsApp Image 2024-08-08 at 1.33.5...

Earlier this week (2)
jdk-22_windows-x64_bin
kafka_2.13-3.8.0 (1)

A long time ago (1)
GoogleChromeEnterpriseBundle64

Extraction path and options

General | Advanced

Destination path (will be created if does not exist)

C:\

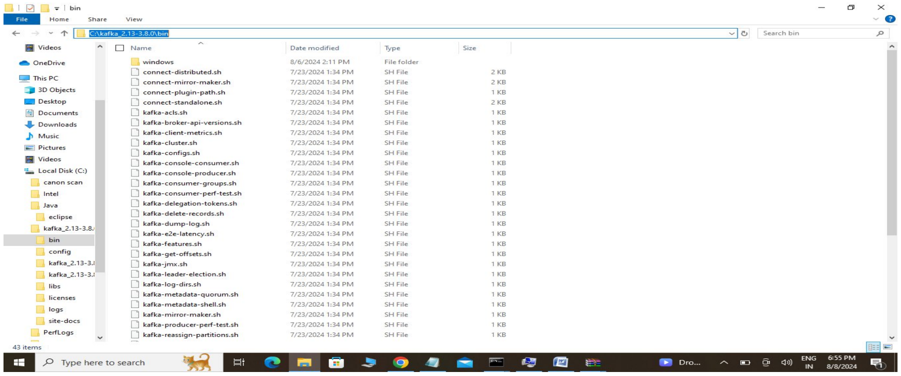
Update mode
- ● Extract and replace files
- ○ Extract and update files
- ○ Fresh existing files only

Overwrite mode
- ● Ask before overwrite
- ○ Overwrite without prompt
- ○ Skip existing files
- ○ Rename automatically

Miscellaneous
- □ Extract archives to subfolders
- □ Keep broken files
- □ Display files in Explorer

STEP 4: PATH SETUP

Copy Path

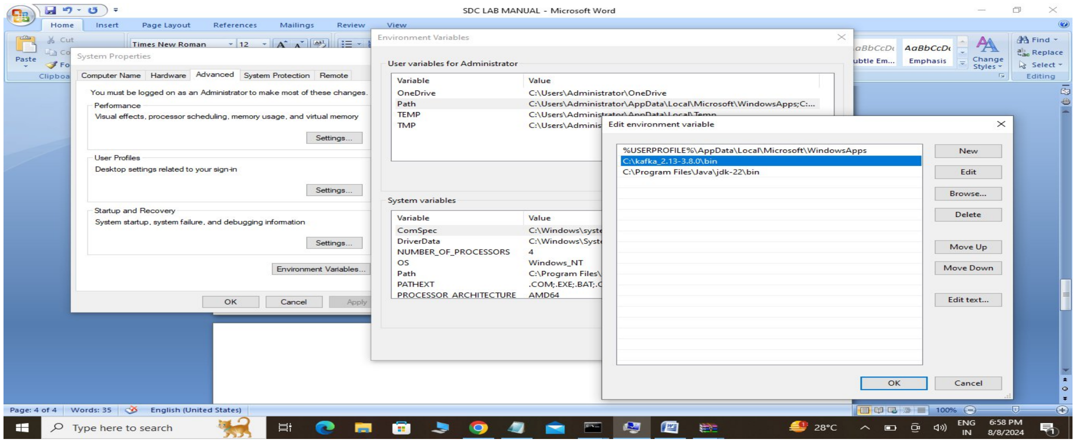


C:\kafka_2.13-3.8.0\bin

Name	Date modified	Type	Size
windows	8/6/2024 2:11 PM	File folder	
connect-distributed.sh	7/23/2024 1:34 PM	SH File	2 KB
connect-mirror-maker.sh	7/23/2024 1:34 PM	SH File	2 KB
connect-plugin-path.sh	7/23/2024 1:34 PM	SH File	1 KB
connect-standalone.sh	7/23/2024 1:34 PM	SH File	2 KB
kafka-acls.sh	7/23/2024 1:34 PM	SH File	1 KB
kafka-broker-api-versions.sh	7/23/2024 1:34 PM	SH File	1 KB
kafka-client-metrics.sh	7/23/2024 1:34 PM	SH File	1 KB
kafka-cluster.sh	7/23/2024 1:34 PM	SH File	1 KB
kafka-configs.sh	7/23/2024 1:34 PM	SH File	1 KB
kafka-console-consumer.sh	7/23/2024 1:34 PM	SH File	1 KB
kafka-console-producer.sh	7/23/2024 1:34 PM	SH File	1 KB
kafka-consumer-groups.sh	7/23/2024 1:34 PM	SH File	1 KB
kafka-consumer-perf-test.sh	7/23/2024 1:34 PM	SH File	1 KB
kafka-delegation-tokens.sh	7/23/2024 1:34 PM	SH File	1 KB
kafka-delete-records.sh	7/23/2024 1:34 PM	SH File	1 KB
kafka-dump-log.sh	7/23/2024 1:34 PM	SH File	1 KB
kafka-e2e-latency.sh	7/23/2024 1:34 PM	SH File	1 KB
kafka-features.sh	7/23/2024 1:34 PM	SH File	1 KB
kafka-get-offsets.sh	7/23/2024 1:34 PM	SH File	1 KB
kafka-jmx.sh	7/23/2024 1:34 PM	SH File	1 KB
kafka-leader-election.sh	7/23/2024 1:34 PM	SH File	1 KB
kafka-log-dirs.sh	7/23/2024 1:34 PM	SH File	1 KB
kafka-metadata-quorum.sh	7/23/2024 1:34 PM	SH File	1 KB
kafka-metadata-shell.sh	7/23/2024 1:34 PM	SH File	1 KB
kafka-mirror-maker.sh	7/23/2024 1:34 PM	SH File	1 KB
kafka-producer-perf-test.sh	7/23/2024 1:34 PM	SH File	1 KB
kafka-reassign-partitions.sh	7/23/2024 1:34 PM	SH File	1 KB

43 items

- Search Edit The System Environment Variable
- Environment Variable->Path-->Edit-->New-->Paste Path-Ok
- Kafka(copy path-->C:\kafka_2.13-3.8.0\bin)



SDC LAB MANUAL - Microsoft Word

System Properties

You must be logged on as an Administrator to make most of these changes.

Performance
Visual effects, processor scheduling, memory usage, and virtual memory

User Profiles
Desktop settings related to your sign-in

Startup and Recovery
System startup, system failure, and debugging information

Environment Variables

User variables for Administrator

Variable	Value
OneDrive	C:\Users\Administrator\OneDrive
Path	C:\Users\Administrator\AppData\Local\Microsoft\WindowsApps;C:...
TEMP	C:\Users\Administrator\AppData\Local\Temp
TMP	C:\Users\Adminis

System variables

Variable	Value
ComSpec	C:\Windows\syste
DriverData	C:\Windows\Syste
NUMBER_OF_PROCESSORS	4
OS	Windows_NT
Path	C:\Program Files\
PATHEXT	.COM;.EXE;.BAT;.C
PROCESSOR_ARCHITECTURE	AMD64

Edit environment variable

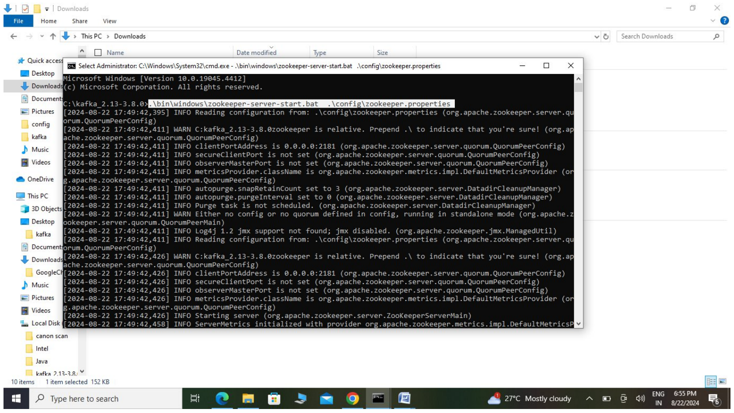
%USERPROFILE%\AppData\Local\Microsoft\WindowsApps
C:\kafka_2.13-3.8.0\bin
C:\Program Files\Java\jdk-22\bin

EXPERIMENT NO 2

NAME OF THE EXPERIMENT: Configuring Multi-Broker Kafka

STEP 1: Open cmd and enter into the path C:\kafka_2.13-3.8.0
PATH: C:\kafka_2.13-3.8.0

TYPE: .\bin\windows\zookeeper-server-start.bat .\config\zookeeper.properties

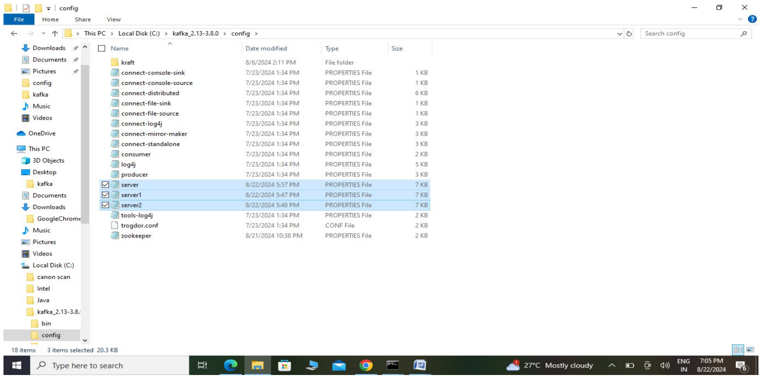


Select Administrator: C:\Windows\System32\cmd.exe - .\bin\windows\zookeeper-server-start.bat .\config\zookeeper.properties

```
Microsoft Windows [Version 10.0.19045.4412]
(c) Microsoft Corporation. All rights reserved.

C:\kafka_2.13-3.8.0>.\bin\windows\zookeeper-server-start.bat  .\config\zookeeper.properties
[2024-08-22 17:49:42,395] INFO Reading configuration from: .\config\zookeeper.properties (org.apache.zookeeper.server.qu
orum.QuorumPeerConfig)
[2024-08-22 17:49:42,411] WARN C:kafka_2.13-3.8.0zookeeper is relative. Prepend .\ to indicate that you're sure! (org.ap
ache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-08-22 17:49:42,411] INFO clientPortAddress is 0.0.0.0:2181 (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-08-22 17:49:42,411] INFO secureClientPort is not set (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-08-22 17:49:42,411] INFO observerMasterPort is not set (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-08-22 17:49:42,411] INFO metricsProvider.className is org.apache.zookeeper.metrics.impl.DefaultMetricsProvider (or
g.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-08-22 17:49:42,411] INFO autopurge.snapRetainCount set to 3 (org.apache.zookeeper.server.DatadirCleanupManager)
[2024-08-22 17:49:42,411] INFO autopurge.purgeInterval set to 0 (org.apache.zookeeper.server.DatadirCleanupManager)
[2024-08-22 17:49:42,411] INFO Purge task is not scheduled. (org.apache.zookeeper.server.DatadirCleanupManager)
[2024-08-22 17:49:42,411] WARN Either no config or no quorum defined in config, running in standalone mode (org.apache.z
ookeeper.server.quorum.QuorumPeerMain)
[2024-08-22 17:49:42,411] INFO Log4j 1.2 jmx support not found; jmx disabled. (org.apache.zookeeper.jmx.ManagedUtil)
[2024-08-22 17:49:42,411] INFO Reading configuration from: .\config\zookeeper.properties (org.apache.zookeeper.server.qu
orum.QuorumPeerConfig)
[2024-08-22 17:49:42,426] WARN C:kafka_2.13-3.8.0zookeeper is relative. Prepend .\ to indicate that you're sure! (org.ap
ache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-08-22 17:49:42,426] INFO clientPortAddress is 0.0.0.0:2181 (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-08-22 17:49:42,426] INFO secureClientPort is not set (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-08-22 17:49:42,426] INFO observerMasterPort is not set (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-08-22 17:49:42,426] INFO metricsProvider.className is org.apache.zookeeper.metrics.impl.DefaultMetricsProvider (or
g.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-08-22 17:49:42,426] INFO Starting server (org.apache.zookeeper.server.ZooKeeperServerMain)
[2024-08-22 17:49:42,458] INFO ServerMetrics initialized with provider org.apache.zookeeper.metrics.impl.DefaultMetricsP
```

STEP 2: Open kafka folder and click on config

STEP 3: Copy the server and paste 2 times and save it with server1 and server2



This PC > Local Disk (C:) > kafka_2.13-3.8.0 > config

	Name	Date modified	Type	Size
	kraft	8/6/2024 2:11 PM	File folder	
	connect-console-sink	7/23/2024 1:34 PM	PROPERTIES File	1 KB
	connect-console-source	7/23/2024 1:34 PM	PROPERTIES File	1 KB
	connect-distributed	7/23/2024 1:34 PM	PROPERTIES File	6 KB
	connect-file-sink	7/23/2024 1:34 PM	PROPERTIES File	1 KB
	connect-file-source	7/23/2024 1:34 PM	PROPERTIES File	1 KB
	connect-log4j	7/23/2024 1:34 PM	PROPERTIES File	3 KB
	connect-mirror-maker	7/23/2024 1:34 PM	PROPERTIES File	3 KB
	connect-standalone	7/23/2024 1:34 PM	PROPERTIES File	3 KB
	consumer	7/23/2024 1:34 PM	PROPERTIES File	2 KB
	log4j	7/23/2024 1:34 PM	PROPERTIES File	5 KB
	producer	7/23/2024 1:34 PM	PROPERTIES File	3 KB
☑	server	8/22/2024 5:57 PM	PROPERTIES File	7 KB
☑	server1	8/22/2024 5:47 PM	PROPERTIES File	7 KB
☑	server2	8/22/2024 5:48 PM	PROPERTIES File	7 KB
	tools-log4j	7/23/2024 1:34 PM	PROPERTIES File	2 KB
	trogdor.conf	7/23/2024 1:34 PM	CONF File	2 KB
	zookeeper	8/21/2024 10:38 PM	PROPERTIES File	2 KB

18 items   3 items selected 20.3 KB

STEP 4 : Now OPEN server1 and change the borker.id , listeners and log.dirs

Examples: - These three steps are there in below images

1. In Server, it contains borker.id=0 then convert 0 into 1 in server1 and 2 in server2

2. In Server contains this : #listeners=PLAINTEXT://:9092 then

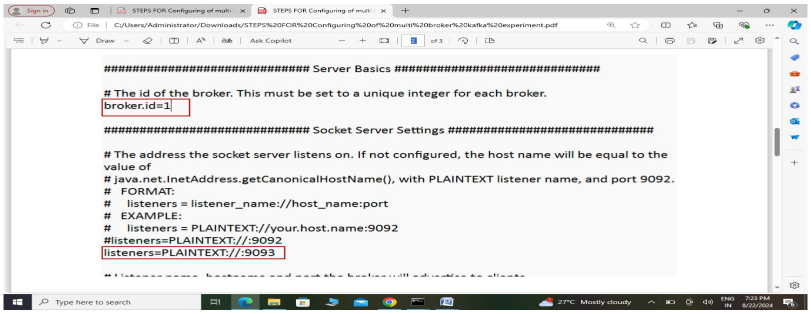
We have to copy it and paste same and change host number

➔ listeners=PLAINTEXT://:9093 for server1 and

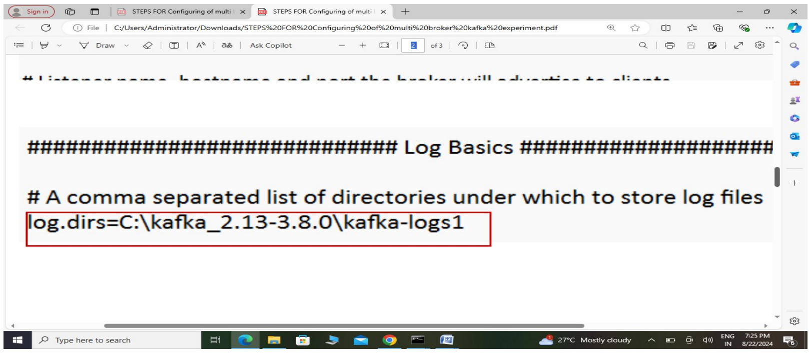
➔ listeners=PLAINTEXT://:9094 for server2

3. In server, it contains log.dirs

We have change, log into log1 and log2 in server1 and server2



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

# The id of the broker. This must be set to a unique integer for each broker.
broker.id=1

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

# The address the socket server listens on. If not configured, the host name will be equal to the
value of
# java.net.InetAddress.getCanonicalHostName(), with PLAINTEXT listener name, and port 9092.
#   FORMAT:
#     listeners = listener_name://host_name:port
#   EXAMPLE:
#     listeners = PLAINTEXT://your.host.name:9092
#listeners=PLAINTEXT://:9092
listeners=PLAINTEXT://:9093
```

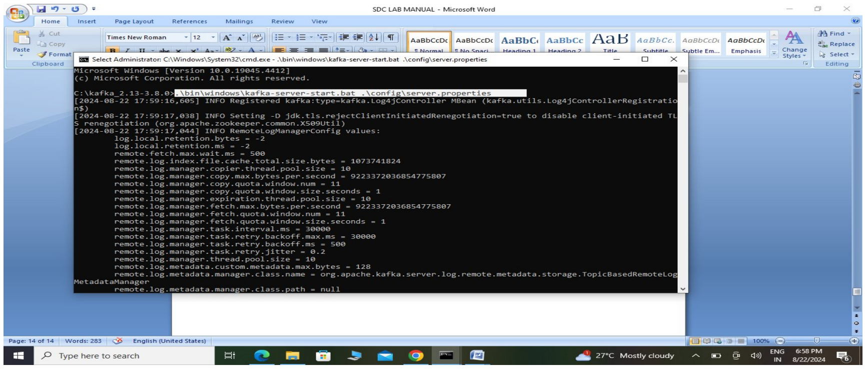


```
############################# Log Basics #############################

# A comma separated list of directories under which to store log files
log.dirs=C:\kafka_2.13-3.8.0\kafka-logs1
```

STEP 5: Open A NEW CMD and enter into the path C:\kafka_2.13-3.8.0

PATH: C:\kafka_2.13-3.8.0

TYPE: .\bin\windows\kafka-server-start.bat .\config\server.properties

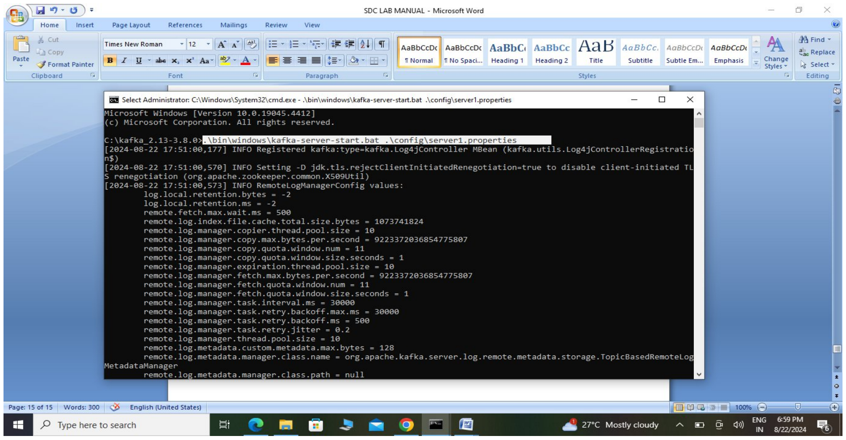


```
Microsoft Windows [Version 10.0.19045.4412]
(c) Microsoft Corporation. All rights reserved.

C:\kafka_2.13-3.8.0>.\bin\windows\kafka-server-start.bat .\config\server.properties
[2024-08-22 17:59:16,605] INFO Registered kafka:type=kafka.Log4jController MBean (kafka.utils.Log4jControllerRegistratio
n$)
[2024-08-22 17:59:17,038] INFO Setting -D jdk.tls.rejectClientInitiatedRenegotiation=true to disable client-initiated TL
S renegotiation (org.apache.zookeeper.common.X509Util)
[2024-08-22 17:59:17,044] INFO RemoteLogManagerConfig values:
        log.local.retention.bytes = -2
        log.local.retention.ms = -2
        remote.fetch.max.wait.ms = 500
        remote.log.index.file.cache.total.size.bytes = 1073741824
        remote.log.manager.copier.thread.pool.size = 10
        remote.log.manager.copy.max.bytes.per.second = 9223372036854775807
        remote.log.manager.copy.quota.window.num = 11
        remote.log.manager.copy.quota.window.size.seconds = 1
        remote.log.manager.expiration.thread.pool.size = 10
        remote.log.manager.fetch.max.bytes.per.second = 9223372036854775807
        remote.log.manager.fetch.quota.window.num = 11
        remote.log.manager.fetch.quota.window.size.seconds = 1
        remote.log.manager.task.interval.ms = 30000
        remote.log.manager.task.retry.backoff.max.ms = 30000
        remote.log.manager.task.retry.backoff.ms = 500
        remote.log.manager.task.retry.jitter = 0.2
        remote.log.manager.thread.pool.size = 10
        remote.log.metadata.custom.metadata.max.bytes = 128
        remote.log.metadata.manager.class.name = org.apache.kafka.server.log.remote.metadata.storage.TopicBasedRemoteLog
MetadataManager
        remote.log.metadata.manager.class.path = null
```

STEP 6: Open A NEW CMD and enter into the path C:\kafka_2.13-3.8.0

PATH: C:\kafka_2.13-3.8.0

TYPE: .\bin\windows\kafka-server-start.bat .\config\server1.properties



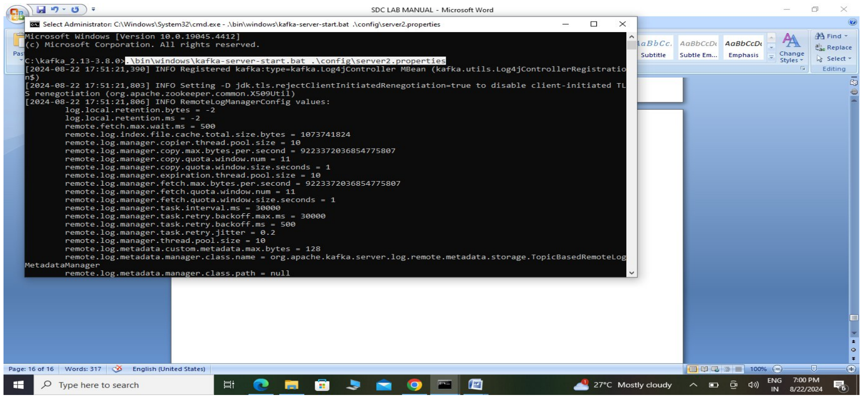
Select Administrator: C:\Windows\System32\cmd.exe - .\bin\windows\kafka-server-start.bat .\config\server1.properties

```
Microsoft Windows [Version 10.0.19045.4412]
(c) Microsoft Corporation. All rights reserved.

C:\kafka_2.13-3.8.0>.\bin\windows\kafka-server-start.bat .\config\server1.properties
[2024-08-22 17:51:00,177] INFO Registered kafka:type=kafka.Log4jController MBean (kafka.utils.Log4jControllerRegistratio
n$)
[2024-08-22 17:51:00,570] INFO Setting -D jdk.tls.rejectClientInitiatedRenegotiation=true to disable client-initiated TL
S renegotiation (org.apache.zookeeper.common.X509Util)
[2024-08-22 17:51:00,573] INFO RemoteLogManagerConfig values:
        log.local.retention.bytes = -2
        log.local.retention.ms = -2
        remote.fetch.max.wait.ms = 500
        remote.log.index.file.cache.total.size.bytes = 1073741824
        remote.log.manager.copier.thread.pool.size = 10
        remote.log.manager.copy.max.bytes.per.second = 9223372036854775807
        remote.log.manager.copy.quota.window.num = 11
        remote.log.manager.copy.quota.window.size.seconds = 1
        remote.log.manager.expiration.thread.pool.size = 10
        remote.log.manager.fetch.max.bytes.per.second = 9223372036854775807
        remote.log.manager.fetch.quota.window.num = 11
        remote.log.manager.fetch.quota.window.size.seconds = 1
        remote.log.manager.task.interval.ms = 30000
        remote.log.manager.task.retry.backoff.max.ms = 30000
        remote.log.manager.task.retry.backoff.ms = 500
        remote.log.manager.task.retry.jitter = 0.2
        remote.log.manager.thread.pool.size = 10
        remote.log.metadata.custom.metadata.max.bytes = 128
        remote.log.metadata.manager.class.name = org.apache.kafka.server.log.remote.metadata.storage.TopicBasedRemoteLog
MetadataManager
        remote.log.metadata.manager.class.path = null
```

STEP 7: Open A NEW CMD and enter into the path C:\kafka_2.13-3.8.0

PATH: C:\kafka_2.13-3.8.0

TYPE: .\bin\windows\kafka-server-start.bat .\config\server2.properties



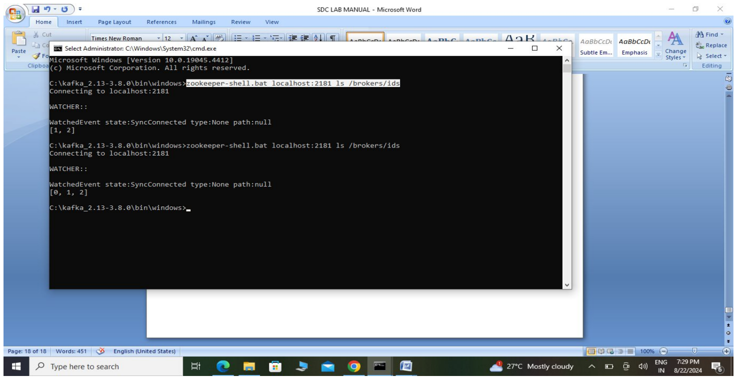
Select Administrator: C:\Windows\System32\cmd.exe - .\bin\windows\kafka-server-start.bat .\config\server2.properties

```
Microsoft Windows [Version 10.0.19045.4412]
(c) Microsoft Corporation. All rights reserved.

C:\kafka_2.13-3.8.0>.\bin\windows\kafka-server-start.bat .\config\server2.properties
[2024-08-22 17:51:21,390] INFO Registered kafka:type=kafka.Log4jController MBean (kafka.utils.Log4jControllerRegistratio
n$)
[2024-08-22 17:51:21,803] INFO Setting -D jdk.tls.rejectClientInitiatedRenegotiation=true to disable client-initiated TL
S renegotiation (org.apache.zookeeper.common.X509Util)
[2024-08-22 17:51:21,806] INFO RemoteLogManagerConfig values:
        log.local.retention.bytes = -2
        log.local.retention.ms = -2
        remote.fetch.max.wait.ms = 500
        remote.log.index.file.cache.total.size.bytes = 1073741824
        remote.log.manager.copier.thread.pool.size = 10
        remote.log.manager.copy.max.bytes.per.second = 9223372036854775807
        remote.log.manager.copy.quota.window.num = 11
        remote.log.manager.copy.quota.window.size.seconds = 1
        remote.log.manager.expiration.thread.pool.size = 10
        remote.log.manager.fetch.max.bytes.per.second = 9223372036854775807
        remote.log.manager.fetch.quota.window.num = 11
        remote.log.manager.fetch.quota.window.size.seconds = 1
        remote.log.manager.task.interval.ms = 30000
        remote.log.manager.task.retry.backoff.max.ms = 30000
        remote.log.manager.task.retry.backoff.ms = 500
        remote.log.manager.task.retry.jitter = 0.2
        remote.log.manager.thread.pool.size = 10
        remote.log.metadata.custom.metadata.max.bytes = 128
        remote.log.metadata.manager.class.name = org.apache.kafka.server.log.remote.metadata.storage.TopicBasedRemoteLog
MetadataManager
        remote.log.metadata.manager.class.path = null
```

STEP 8 : Check the multi brokers ids with given link which can connecting to localhost:2181

PATH: C:\kafka_2.13-3.8.0\bin\windows

TYPE: .\bin\windows\zookeeper-shell.bat localhost:2181 ls /brokers/ids



Select Administrator: C:\Windows\System32\cmd.exe

```
Microsoft Windows [Version 10.0.19045.4412]
(c) Microsoft Corporation. All rights reserved.

C:\kafka_2.13-3.8.0\bin\windows>zookeeper-shell.bat localhost:2181 ls /brokers/ids
Connecting to localhost:2181

WATCHER::

WatchedEvent state:SyncConnected type:None path:null
[1, 2]

C:\kafka_2.13-3.8.0\bin\windows>zookeeper-shell.bat localhost:2181 ls /brokers/ids
Connecting to localhost:2181

WATCHER::

WatchedEvent state:SyncConnected type:None path:null
[0, 1, 2]

C:\kafka_2.13-3.8.0\bin\windows>_
```

- Ⓟ It Contains 3 Brokers [0,1,2]

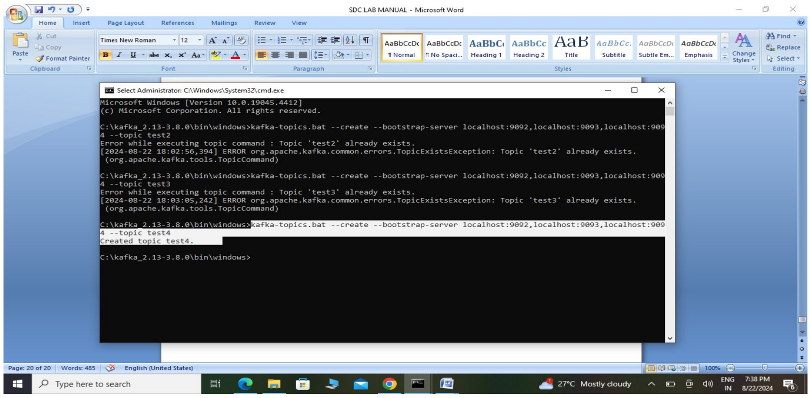
STEP 9 : Checking of multi broker server

STEP 10: open cmd create topic test2

PATH: C:\kafka_2.13-3.8.0\bin\windows

TYPE:

kafka-topics.bat --create --bootstrap-server localhost:9092,localhost:9093,localhost:9094 --topic test2



SDC LAB MANUAL - Microsoft Word
Select Administrator: C:\Windows\System32\cmd.exe

```
Microsoft Windows [Version 10.0.19045.4412]
(c) Microsoft Corporation. All rights reserved.

C:\kafka_2.13-3.8.0\bin\windows>kafka-topics.bat --create --bootstrap-server localhost:9092,localhost:9093,localhost:909
4 --topic test2
Error while executing topic command : Topic 'test2' already exists.
[2024-08-22 18:02:56,394] ERROR org.apache.kafka.common.errors.TopicExistsException: Topic 'test2' already exists.
 (org.apache.kafka.tools.TopicCommand)

C:\kafka_2.13-3.8.0\bin\windows>kafka-topics.bat --create --bootstrap-server localhost:9092,localhost:9093,localhost:909
4 --topic test3
Error while executing topic command : Topic 'test3' already exists.
[2024-08-22 18:03:05,242] ERROR org.apache.kafka.common.errors.TopicExistsException: Topic 'test3' already exists.
 (org.apache.kafka.tools.TopicCommand)

C:\kafka_2.13-3.8.0\bin\windows>kafka-topics.bat --create --bootstrap-server localhost:9092,localhost:9093,localhost:909
4 --topic test4
Created topic test4.

C:\kafka_2.13-3.8.0\bin\windows>
```

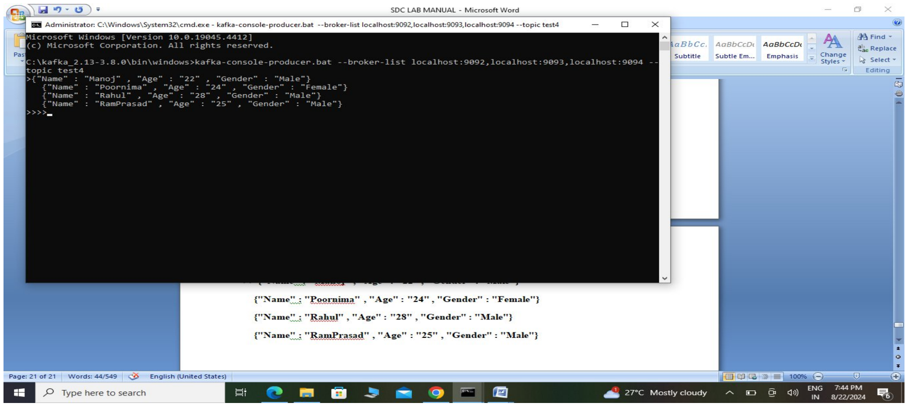
STEP 11 : Open producer on another cmd

PATH: C:\kafka_2.13-3.8.0\bin\windows

TYPE:

kafka-console-producer.bat --broker-list localhost:9092,localhost:9093,localhost:9094 --topic test2

```
>> {"Name" : "Manoj" , "Age" : "22" , "Gender" : "Male"}
   {"Name" : "Poornima" , "Age" : "24" , "Gender" : "Female"}
   {"Name" : "Rahul" , "Age" : "28" , "Gender" : "Male"}
   {"Name" : "RamPrasad" , "Age" : "25" , "Gender" : "Male"}
```



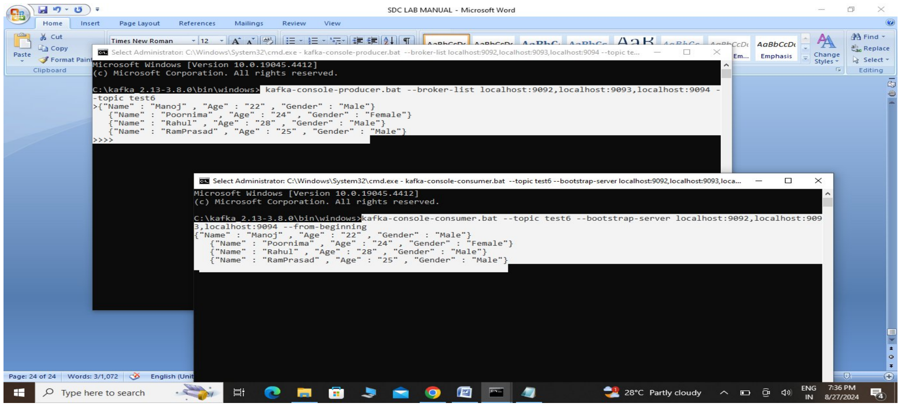
SDC LAB MANUAL - Microsoft Word
Administrator: C:\Windows\System32\cmd.exe - kafka-console-producer.bat --broker-list localhost:9092,localhost:9093,localhost:9094 --topic test4

```
Microsoft Windows [Version 10.0.19045.4412]
(c) Microsoft Corporation. All rights reserved.

C:\kafka_2.13-3.8.0\bin\windows>kafka-console-producer.bat --broker-list localhost:9092,localhost:9093,localhost:9094 --
topic test4
>{"Name" : "Manoj" , "Age" : "22" , "Gender" : "Male"}
  {"Name" : "Poornima" , "Age" : "24" , "Gender" : "Female"}
  {"Name" : "Rahul" , "Age" : "28" , "Gender" : "Male"}
  {"Name" : "RamPrasad" , "Age" : "25" , "Gender" : "Male"}
>>>>
```

STEP 12: Open consumer on another cmd

PATH: C:\kafka_2.13-3.8.0\bin\windows

TYPE: kafka-console-consumer.bat --topic test2 --bootstrap-server localhost:9092,localhost:9093,localhost:9094 --from-beginning



Select Administrator: C:\Windows\System32\cmd.exe - kafka-console-producer.bat --broker-list localhost:9092,localhost:9093,localhost:9094 --topic te...

```
Microsoft Windows [Version 10.0.19045.4412]
(c) Microsoft Corporation. All rights reserved.

C:\kafka_2.13-3.8.0\bin\windows> kafka-console-producer.bat --broker-list localhost:9092,localhost:9093,localhost:9094 -
-topic test6
>{"Name" : "Manoj" , "Age" : "22" , "Gender" : "Male"}
   {"Name" : "Poornima" , "Age" : "24" , "Gender" : "Female"}
   {"Name" : "Rahul" , "Age" : "28" , "Gender" : "Male"}
   {"Name" : "RamPrasad" , "Age" : "25" , "Gender" : "Male"}
>>>>
```

Select Administrator: C:\Windows\System32\cmd.exe - kafka-console-consumer.bat --topic test6 --bootstrap-server localhost:9092,localhost:9093,loca...

```
Microsoft Windows [Version 10.0.19045.4412]
(c) Microsoft Corporation. All rights reserved.

C:\kafka_2.13-3.8.0\bin\windows>kafka-console-consumer.bat --topic test6 --bootstrap-server localhost:9092,localhost:909
3,localhost:9094 --from-beginning
{"Name" : "Manoj" , "Age" : "22" , "Gender" : "Male"}
   {"Name" : "Poornima" , "Age" : "24" , "Gender" : "Female"}
   {"Name" : "Rahul" , "Age" : "28" , "Gender" : "Male"}
   {"Name" : "RamPrasad" , "Age" : "25" , "Gender" : "Male"}
```

EXPERIMENT NO: 3

NAME OF THE EXPERIMENT: Exploring Zookeeper.

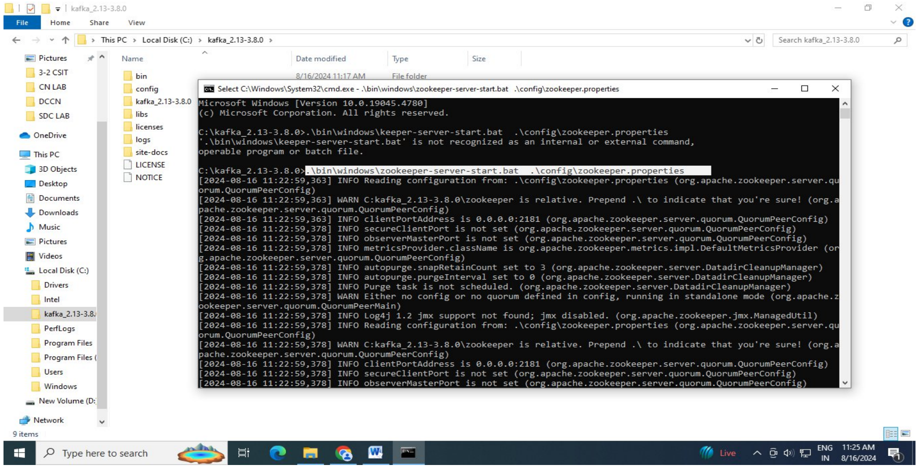
STEP 1:enter into c driveà kafka_2.13-3.8.0àC:\kafka_2.13-3.8.0\configàopen server

log.dirs=C:\kafka_2.13-3.8.0\kafka-logs (replace temp with C:\kafka_2.13-3.8.0) and SAVE the file

STEP 2: Enter into c driveà kafka_2.13-3.8.0àC:\kafka_2.13-3.8.0\configàopen zookeeper

dataDir=C:\kafka_2.13-3.8.0\zookeeper (replace temp with C:\kafka_2.13-3.8.0) and SAVE the file

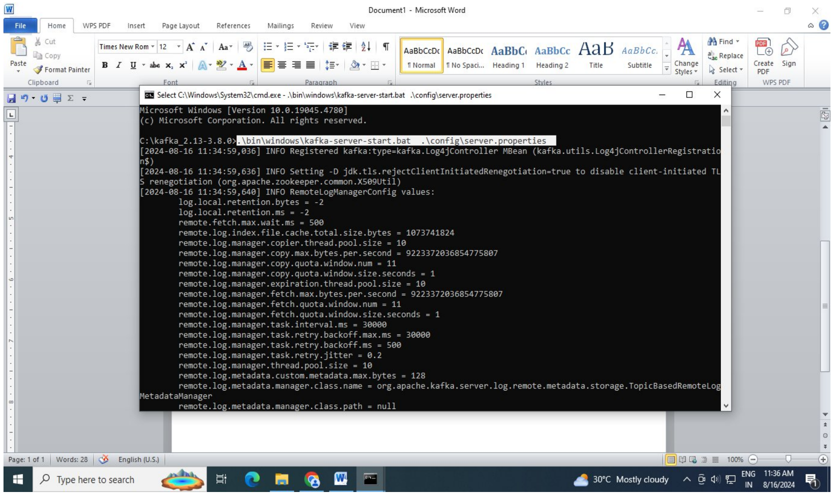
STEP 3: Open cmd and enter into the path C:\kafka_2.13-3.8.0
PATH: C:\kafka_2.13-3.8.0
TYPE:.\bin\windows\zookeeper-server-start.bat .\config\zookeeper.properties



Select C:\Windows\System32\cmd.exe - .\bin\windows\zookeeper-server-start.bat .\config\zookeeper.properties

```
Microsoft Windows [Version 10.0.19045.4780]
(c) Microsoft Corporation. All rights reserved.

C:\kafka_2.13-3.8.0>.\bin\windows\keeper-server-start.bat  .\config\zookeeper.properties
'.\bin\windows\keeper-server-start.bat' is not recognized as an internal or external command,
operable program or batch file.

C:\kafka_2.13-3.8.0>.\bin\windows\zookeeper-server-start.bat  .\config\zookeeper.properties
[2024-08-16 11:22:59,363] INFO Reading configuration from: .\config\zookeeper.properties (org.apache.zookeeper.server.qu
orum.QuorumPeerConfig)
[2024-08-16 11:22:59,363] WARN C:kafka_2.13-3.8.0\zookeeper is relative. Prepend .\ to indicate that you're sure! (org.a
pache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-08-16 11:22:59,363] INFO clientPortAddress is 0.0.0.0:2181 (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-08-16 11:22:59,378] INFO secureClientPort is not set (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-08-16 11:22:59,378] INFO observerMasterPort is not set (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-08-16 11:22:59,378] INFO metricsProvider.className is org.apache.zookeeper.metrics.impl.DefaultMetricsProvider (or
g.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-08-16 11:22:59,378] INFO autopurge.snapRetainCount set to 3 (org.apache.zookeeper.server.DatadirCleanupManager)
[2024-08-16 11:22:59,378] INFO autopurge.purgeInterval set to 0 (org.apache.zookeeper.server.DatadirCleanupManager)
[2024-08-16 11:22:59,378] INFO Purge task is not scheduled. (org.apache.zookeeper.server.DatadirCleanupManager)
[2024-08-16 11:22:59,378] WARN Either no config or no quorum defined in config, running in standalone mode (org.apache.z
ookeeper.server.quorum.QuorumPeerMain)
[2024-08-16 11:22:59,378] INFO Log4j 1.2 jmx support not found; jmx disabled. (org.apache.zookeeper.jmx.ManagedUtil)
[2024-08-16 11:22:59,378] INFO Reading configuration from: .\config\zookeeper.properties (org.apache.zookeeper.server.qu
orum.QuorumPeerConfig)
[2024-08-16 11:22:59,378] WARN C:kafka_2.13-3.8.0\zookeeper is relative. Prepend .\ to indicate that you're sure! (org.a
pache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-08-16 11:22:59,378] INFO clientPortAddress is 0.0.0.0:2181 (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-08-16 11:22:59,378] INFO secureClientPort is not set (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-08-16 11:22:59,378] INFO observerMasterPort is not set (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
```

NOTE: DON'T CLOSE CMD
STEP 4: Open A NEW CMD and enter into the path C:\kafka_2.13-3.8.0
PATH: C:\kafka_2.13-3.8.0
TYPE: .\bin\windows\kafka-server-start.bat .\config\server.properties



Select C:\Windows\System32\cmd.exe - .\bin\windows\kafka-server-start.bat .\config\server.properties

```
Microsoft Windows [Version 10.0.19045.4780]
(c) Microsoft Corporation. All rights reserved.

C:\kafka_2.13-3.8.0>.\bin\windows\kafka-server-start.bat  .\config\server.properties
[2024-08-16 11:34:59,036] INFO Registered kafka:type=kafka.Log4jController MBean (kafka.utils.Log4jControllerRegistratio
n$)
[2024-08-16 11:34:59,636] INFO Setting -D jdk.tls.rejectClientInitiatedRenegotiation=true to disable client-initiated TL
S renegotiation (org.apache.zookeeper.common.X509Util)
[2024-08-16 11:34:59,640] INFO RemoteLogManagerConfig values:
        log.local.retention.bytes = -2
        log.local.retention.ms = -2
        remote.fetch.max.wait.ms = 500
        remote.log.index.file.cache.total.size.bytes = 1073741824
        remote.log.manager.copier.thread.pool.size = 10
        remote.log.manager.copy.max.bytes.per.second = 9223372036854775807
        remote.log.manager.copy.quota.window.num = 11
        remote.log.manager.copy.quota.window.size.seconds = 1
        remote.log.manager.expiration.thread.pool.size = 10
        remote.log.manager.fetch.max.bytes.per.second = 9223372036854775807
        remote.log.manager.fetch.quota.window.num = 11
        remote.log.manager.fetch.quota.window.size.seconds = 1
        remote.log.manager.task.interval.ms = 30000
        remote.log.manager.task.retry.backoff.max.ms = 30000
        remote.log.manager.task.retry.backoff.ms = 500
        remote.log.manager.task.retry.jitter = 0.2
        remote.log.manager.thread.pool.size = 10
        remote.log.metadata.custom.metadata.max.bytes = 128
        remote.log.metadata.manager.class.name = org.apache.kafka.server.log.remote.metadata.storage.TopicBasedRemoteLog
MetadataManager
        remote.log.metadata.manager.class.path = null
```

EXPERIMENT NO 4

NAME OF THE EXPERIMENT: Overview of kafka consumers

Create a Kafka Topic:

1. Open a new command prompt in the location C:\kafka\bin\windows.
2. Run the following command:

Output: The topic test created successfully.

Creating Kafka Producer:

1. Open a new command prompt in the location C:\kafka\bin\windows
2. Run the following command:

```
C:\kafka\bin\windows>kafka-console-producer.bat --broker-list
localhost:9092 --topic test
```

The Kafka Producer is executing sending the messages to consumer in a separate window.

Creating Kafka Consumer:

1. Open a new command prompt in the location C:\kafka\bin\windows.
2. Run the following command:

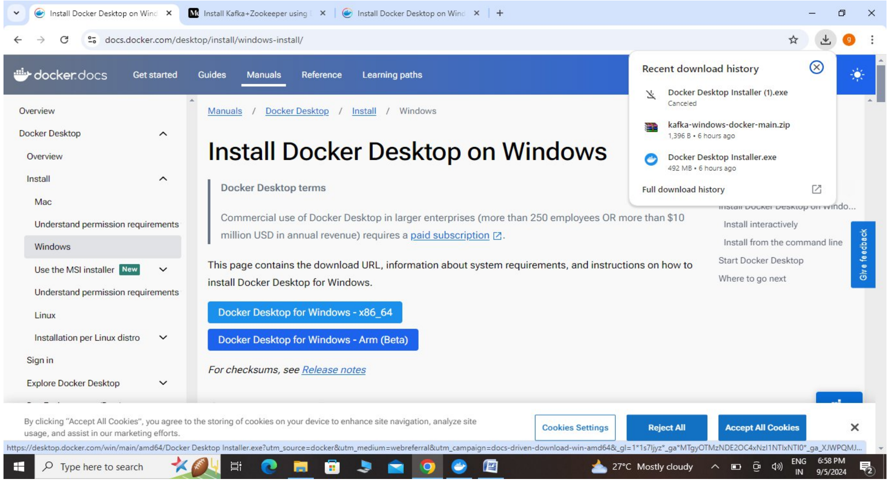
```
kafka-console-producer.bat --broker-list localhost:9092 --topic test
C:\kafka\bin\windows>kafka-topics.bat --create –bootstrap-server localhost:9092 --topic test
```

EXPERIMENT NO: 5

NAME OF THE EXPERIMENT: Using Kafka with Docker.

STEP 1: INSTALL DOCKER

- Download latest docker from https://docs.docker.com/desktop/install/windows-install/
- Follow normal installation steps to install docker desktop



docs.docker.com/desktop/install/windows-install/

Manuals / Docker Desktop / Install / Windows

Install Docker Desktop on Windows

Docker Desktop terms

Commercial use of Docker Desktop in larger enterprises (more than 250 employees OR more than $10 million USD in annual revenue) requires a paid subscription.

This page contains the download URL, information about system requirements, and instructions on how to install Docker Desktop for Windows.

Docker Desktop for Windows - x86_64

Docker Desktop for Windows - Arm (Beta)

For checksums, see Release notes

Recent download history

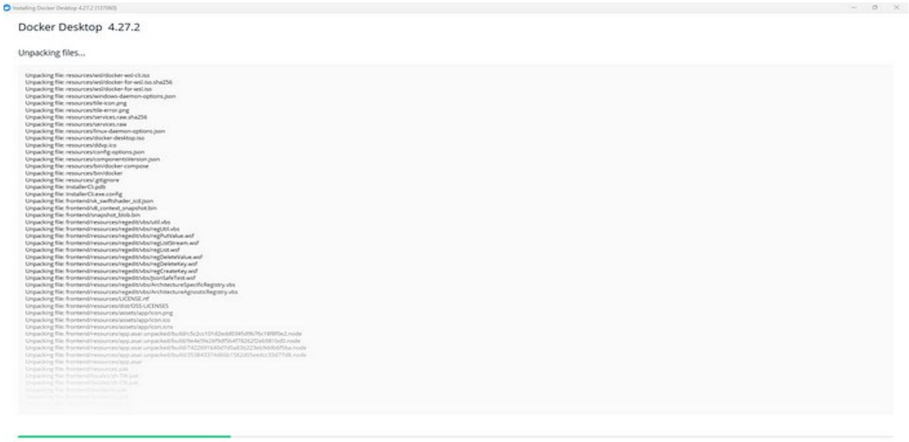
Docker Desktop Installer (1).exe
Canceled

kafka-windows-docker-main.zip
1,396 B • 6 hours ago

Docker Desktop Installer.exe
492 MB • 6 hours ago

Full download history

By clicking "Accept All Cookies", you agree to the storing of cookies on your device to enhance site navigation, analyze site usage, and assist in our marketing efforts.

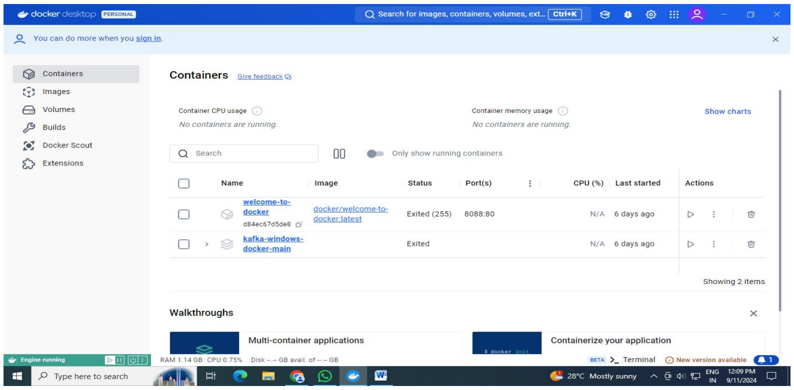


Installing Docker Desktop 4.27.2 (137060)

Docker Desktop 4.27.2

Unpacking files...

```
Unpacking file: resources/wsl/docker-wsl-cli.iso
Unpacking file: resources/wsl/docker-for-wsl.iso.sha256
Unpacking file: resources/wsl/docker-for-wsl.iso
Unpacking file: resources/windows-daemon-options.json
Unpacking file: resources/tile-icon.png
Unpacking file: resources/tile-error.png
Unpacking file: resources/services.raw.sha256
Unpacking file: resources/services.raw
Unpacking file: resources/linux-daemon-options.json
Unpacking file: resources/docker-desktop.iso
Unpacking file: resources/ddvp.ico
Unpacking file: resources/config-options.json
Unpacking file: resources/componentsVersion.json
Unpacking file: resources/bin/docker-compose
Unpacking file: resources/bin/docker
Unpacking file: resources/.gitignore
Unpacking file: InstallerCli.pdb
Unpacking file: InstallerCli.exe.config
Unpacking file: frontend/vk_swiftshader_icd.json
Unpacking file: frontend/v8_context_snapshot.bin
Unpacking file: frontend/snapshot_blob.bin
Unpacking file: frontend/resources/regedit/vbs/util.vbs
Unpacking file: frontend/resources/regedit/vbs/regUtil.vbs
Unpacking file: frontend/resources/regedit/vbs/regPutValue.wsf
Unpacking file: frontend/resources/regedit/vbs/regListStream.wsf
Unpacking file: frontend/resources/regedit/vbs/regList.wsf
Unpacking file: frontend/resources/regedit/vbs/regDeleteValue.wsf
Unpacking file: frontend/resources/regedit/vbs/regDeleteKey.wsf
Unpacking file: frontend/resources/regedit/vbs/regCreateKey.wsf
Unpacking file: frontend/resources/regedit/vbs/JsonSafeTest.wsf
Unpacking file: frontend/resources/regedit/vbs/ArchitectureSpecificRegistry.vbs
Unpacking file: frontend/resources/regedit/vbs/ArchitectureAgnosticRegistry.vbs
Unpacking file: frontend/resources/LICENSE.rtf
Unpacking file: frontend/resources/dist/OSS-LICENSES
Unpacking file: frontend/resources/assets/app/icon.png
Unpacking file: frontend/resources/assets/app/icon.ico
Unpacking file: frontend/resources/assets/app/icon.icns
Unpacking file: frontend/resources/app.asar.unpacked/build/[illegible].node
Unpacking file: frontend/resources/app.asar.unpacked/build/[illegible].node
Unpacking file: frontend/resources/app.asar.unpacked/build/[illegible].node
Unpacking file: frontend/resources/app.asar.unpacked/build/[illegible].node
Unpacking file: frontend/resources/app.asar
Unpacking file: frontend/resources.pak
Unpacking file: frontend/locales/zh-TW.pak
Unpacking file: frontend/locales/zh-CN.pak
[illegible]
```



You can do more when you sign in.

Containers

Container CPU usage
No containers are running.

Container memory usage
No containers are running.

	Name	Image	Status	Port(s)	CPU (%)	Last started	Actions
□	welcome-to-docker d84ec67d5de8	docker/welcome-to-docker:latest	Exited (255)	8088:80	N/A	6 days ago	
□	kafka-windows-docker-main		Exited		N/A	6 days ago	

Showing 2 items

STEP 2: Create docker-compose.yml file

First Create a folder called kafka-windows-docker-main

Inside this folder create a .yml file called docker-compose.yml

docker-compose.yml

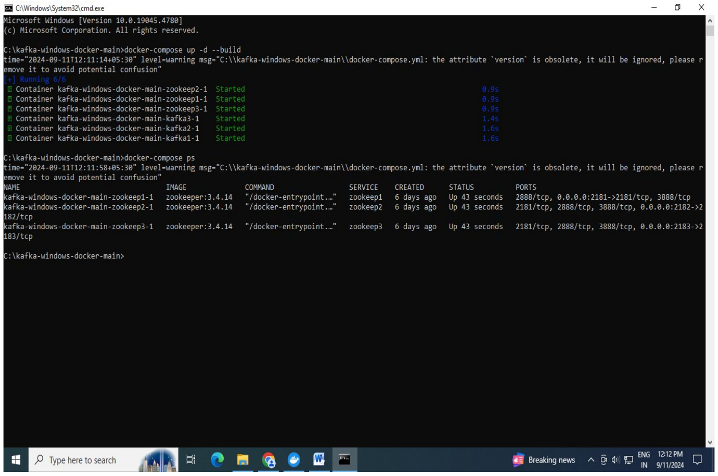
```
version: '2.1'

services:
zookeep1:
image: zookeeper:3.4.14
hostname: zookeep1
ports:
    - "2181:2181"
environment:
ZOO_MY_ID: 1
ZOO_PORT: 2181
ZOO_SERVERS: server.1=zookeep1:2888:3888 server.2=zookeep2:2888:3888 server.3=zookeep3:2888:3888
volumes:
    - ./kafka-windows-docker-data/zookeep1/data:/data
    - ./kafka-windows-docker-data/zookeep1/datalog:/datalog

zookeep2:
image: zookeeper:3.4.14
hostname: zookeep2
ports:
    - "2182:2182"
environment:
ZOO_MY_ID: 2
ZOO_PORT: 2182
ZOO_SERVERS: server.1=zookeep1:2888:3888 server.2=zookeep2:2888:3888 server.3=zookeep3:2888:3888
volumes:
    - ./kafka-windows-docker-data/zookeep2/data:/data
    - ./kafka-windows-docker-data/zookeep2/datalog:/datalog
```

```
zookeep3:
image: zookeeper:3.4.14
hostname: zookeep3
ports:
    - "2183:2183"
environment:
ZOO_MY_ID: 3
ZOO_PORT: 2183
ZOO_SERVERS: server.1=zookeep1:2888:3888 server.2=zookeep2:2888:3888

server.3=zookeep3:2888:3888
volumes:
    - ./kafka-windows-docker-data/zookeep3/data:/data
    - ./kafka-windows-docker-data/zookeep3/datalog:/datalog


kafka1:
image: confluentinc/cp-kafka:7.5.0
hostname: kafka1
ports:
    - "9092:9092"
environment:
KAFKA_ADVERTISED_LISTENERS:
LISTENER_DOCKER_INTERNAL://kafka1:19092,LISTENER_DOCKER_EXTERNAL://${DOCKER_HOST_IP:-
127.0.0.1}:9092
KAFKA_LISTENER_SECURITY_PROTOCOL_MAP:
LISTENER_DOCKER_INTERNAL:PLAINTEXT,LISTENER_DOCKER_EXTERNAL:PLAINTEXT
KAFKA_INTER_BROKER_LISTENER_NAME: LISTENER_DOCKER_INTERNAL
KAFKA_ZOOKEEPER_CONNECT: "zookeep1:2181,zookeep2:2182,zookeep3:2183"
KAFKA_BROKER_ID: 1
KAFKA_LOG4J_LOGGERS:
"kafka.controller=INFO,kafka.producer.async.DefaultEventHandler=INFO,state.change.logger=INFO"
volumes:
    - ./kafka-windows-docker-data/kafka1/data:/var/lib/kafka/data
depends_on:
    - zookeep1
    - zookeep2
    - zookeep3

kafka2:
image: confluentinc/cp-kafka:7.5.0
hostname: kafka2
ports:
    - "9093:9093"
environment:
```

```
KAFKA_ADVERTISED_LISTENERS:
LISTENER_DOCKER_INTERNAL://kafka2:19093,LISTENER_DOCKER_EXTERNAL://${DOCKER_HOST_IP:-
127.0.0.1}:9093
KAFKA_LISTENER_SECURITY_PROTOCOL_MAP:
LISTENER_DOCKER_INTERNAL:PLAINTEXT,LISTENER_DOCKER_EXTERNAL:PLAINTEXT
KAFKA_INTER_BROKER_LISTENER_NAME: LISTENER_DOCKER_INTERNAL
KAFKA_ZOOKEEPER_CONNECT: "zookeep1:2181,zookeep2:2182,zookeep3:2183"
KAFKA_BROKER_ID: 2
KAFKA_LOG4J_LOGGERS:
"kafka.controller=INFO,kafka.producer.async.DefaultEventHandler=INFO,state.change.logger=INFO"
volumes:
    - ./kafka-windows-docker-data/kafka2/data:/var/lib/kafka/data
depends_on:
    - zookeep1
    - zookeep2
    - zookeep3

kafka3:
image: confluentinc/cp-kafka:7.5.0

hostname: kafka3
ports:
    - "9094:9094"
environment:
KAFKA_ADVERTISED_LISTENERS:
LISTENER_DOCKER_INTERNAL://kafka3:19094,LISTENER_DOCKER_EXTERNAL://${DOCKER_HOST_IP:-
127.0.0.1}:9094
KAFKA_LISTENER_SECURITY_PROTOCOL_MAP:
LISTENER_DOCKER_INTERNAL:PLAINTEXT,LISTENER_DOCKER_EXTERNAL:PLAINTEXT
KAFKA_INTER_BROKER_LISTENER_NAME: LISTENER_DOCKER_INTERNAL
KAFKA_ZOOKEEPER_CONNECT: "zookeep1:2181,zookeep2:2182,zookeep3:2183"
KAFKA_BROKER_ID: 3
KAFKA_LOG4J_LOGGERS:
"kafka.controller=INFO,kafka.producer.async.DefaultEventHandler=INFO,state.change.logger=INFO"
volumes:
    - ./kafka-windows-docker-data/kafka3/data:/var/lib/kafka/data
depends_on:
    - zookeep1
    - zookeep2
    - zookeep3
```

STEP 3: Install kafka and zookeeper as per the configuration in docker file

docker-compose up -d — build



```
Microsoft Windows [Version 10.0.19045.4780]
(c) Microsoft Corporation. All rights reserved.

C:\kafka-windows-docker-main>docker-compose up -d --build
time="2024-09-11T12:11:14+05:30" level=warning msg="C:\\kafka-windows-docker-main\\docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please r
emove it to avoid potential confusion"
[+] Running 6/6
 Container kafka-windows-docker-main-zookeep2-1  Started                                                 0.9s
 Container kafka-windows-docker-main-zookeep1-1  Started                                                 0.9s
 Container kafka-windows-docker-main-zookeep3-1  Started                                                 0.9s
 Container kafka-windows-docker-main-kafka3-1    Started                                                 1.4s
 Container kafka-windows-docker-main-kafka2-1    Started                                                 1.6s
 Container kafka-windows-docker-main-kafka1-1    Started                                                 1.6s

C:\kafka-windows-docker-main>docker-compose ps
time="2024-09-11T12:11:58+05:30" level=warning msg="C:\\kafka-windows-docker-main\\docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please r
emove it to avoid potential confusion"
NAME                                 IMAGE              COMMAND                   SERVICE    CREATED      STATUS          PORTS
kafka-windows-docker-main-zookeep1-1   zookeeper:3.4.14   "/docker-entrypoint.…"   zookeep1   6 days ago   Up 43 seconds   2888/tcp, 0.0.0.0:2181->2181/tcp, 3888/tcp
kafka-windows-docker-main-zookeep2-1   zookeeper:3.4.14   "/docker-entrypoint.…"   zookeep2   6 days ago   Up 43 seconds   2181/tcp, 2888/tcp, 3888/tcp, 0.0.0.0:2182->2
182/tcp
kafka-windows-docker-main-zookeep3-1   zookeeper:3.4.14   "/docker-entrypoint.…"   zookeep3   6 days ago   Up 43 seconds   2181/tcp, 2888/tcp, 3888/tcp, 0.0.0.0:2183->2
183/tcp

C:\kafka-windows-docker-main>
```

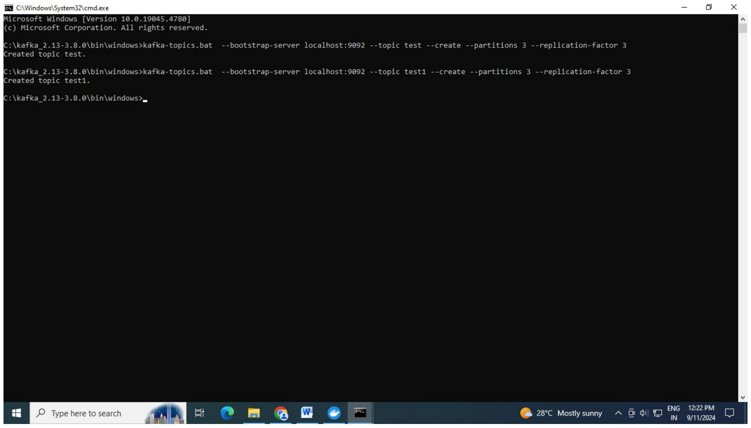
STEP 4: Creating Topics

Create new topic with topic Name “test” with 3 partitions and 3 replication-factor

C:\kafka_2.13-3.8.0\bin\windows>kafka-topics.bat --bootstrap-server localhost:9092 --topic test --create --partitions 3 --replication-factor 3

Create new topic with topicName “test1” with 3 partitions and 3 replication-factor

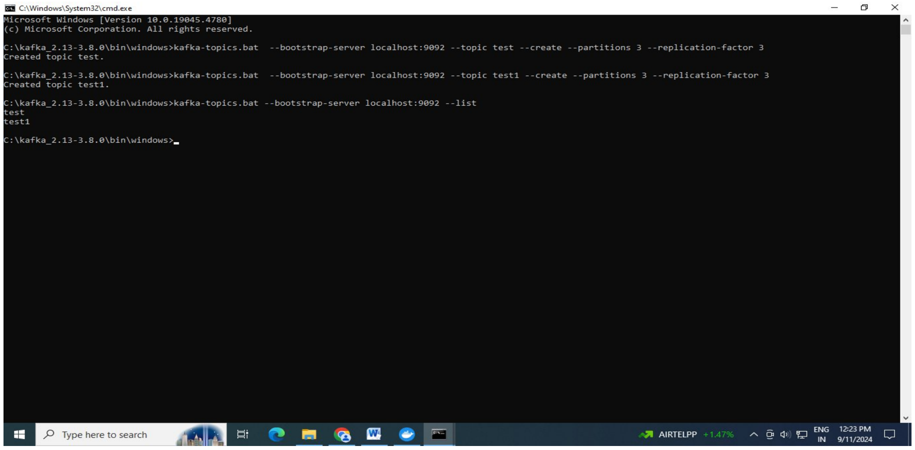
C:\kafka_2.13-3.8.0\bin\windows>kafka-topics.bat --bootstrap-server localhost:9092 --topic test1 --create --partitions 3 --replication-factor 3



```
Microsoft Windows [Version 10.0.19045.4780]
(c) Microsoft Corporation. All rights reserved.

C:\kafka_2.13-3.8.0\bin\windows>kafka-topics.bat  --bootstrap-server localhost:9092 --topic test --create --partitions 3 --replication-factor 3
Created topic test.

C:\kafka_2.13-3.8.0\bin\windows>kafka-topics.bat  --bootstrap-server localhost:9092 --topic test1 --create --partitions 3 --replication-factor 3
Created topic test1.

C:\kafka_2.13-3.8.0\bin\windows>
```

STEP 5 :To List down the topics

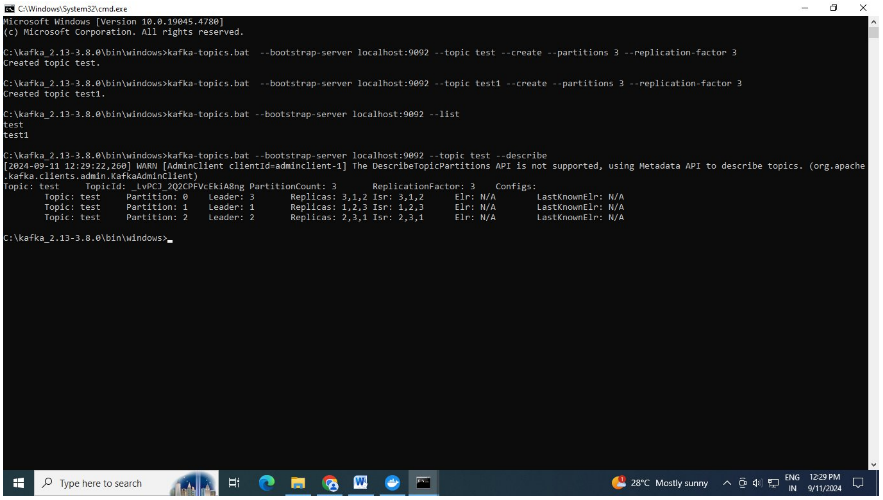
C:\kafka_2.13-3.8.0\bin\windows>kafka-topics.bat --bootstrap-server   localhost:9092 –list



```
Microsoft Windows [Version 10.0.19045.4780]
(c) Microsoft Corporation. All rights reserved.

C:\kafka_2.13-3.8.0\bin\windows>kafka-topics.bat  --bootstrap-server localhost:9092 --topic test --create --partitions 3 --replication-factor 3
Created topic test.

C:\kafka_2.13-3.8.0\bin\windows>kafka-topics.bat  --bootstrap-server localhost:9092 --topic test1 --create --partitions 3 --replication-factor 3
Created topic test1.

C:\kafka_2.13-3.8.0\bin\windows>kafka-topics.bat --bootstrap-server localhost:9092 --list
test
test1

C:\kafka_2.13-3.8.0\bin\windows>
```

STEP 6: Describing a specific topic

kafka-topics.bat --bootstrap-server localhost:9092 --topic test –describe

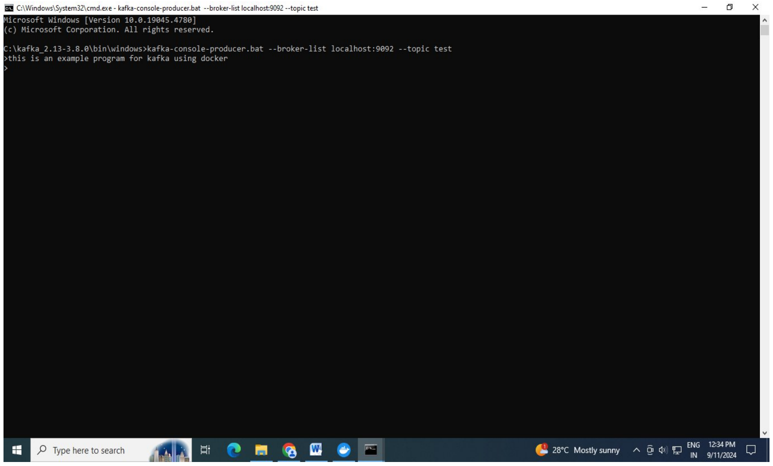


```
Microsoft Windows [Version 10.0.19045.4780]
(c) Microsoft Corporation. All rights reserved.

C:\kafka_2.13-3.8.0\bin\windows>kafka-topics.bat  --bootstrap-server localhost:9092 --topic test --create --partitions 3 --replication-factor 3
Created topic test.

C:\kafka_2.13-3.8.0\bin\windows>kafka-topics.bat  --bootstrap-server localhost:9092 --topic test1 --create --partitions 3 --replication-factor 3
Created topic test1.

C:\kafka_2.13-3.8.0\bin\windows>kafka-topics.bat --bootstrap-server localhost:9092 --list
test
test1

C:\kafka_2.13-3.8.0\bin\windows>kafka-topics.bat --bootstrap-server localhost:9092 --topic test --describe
[2024-09-11 12:29:22,260] WARN [AdminClient clientId=adminclient-1] The DescribeTopicPartitions API is not supported, using Metadata API to describe topics. (org.apache
.kafka.clients.admin.KafkaAdminClient)
Topic: test     TopicId: _LvPCJ_2Q2CPFVcEkiA8ng PartitionCount: 3       ReplicationFactor: 3    Configs:
        Topic: test     Partition: 0    Leader: 3       Replicas: 3,1,2 Isr: 3,1,2      Elr: N/A        LastKnownElr: N/A
        Topic: test     Partition: 1    Leader: 1       Replicas: 1,2,3 Isr: 1,2,3      Elr: N/A        LastKnownElr: N/A
        Topic: test     Partition: 2    Leader: 2       Replicas: 2,3,1 Isr: 2,3,1      Elr: N/A        LastKnownElr: N/A

C:\kafka_2.13-3.8.0\bin\windows>
```

STEP 7: Produce and consume messages

Produce few messages

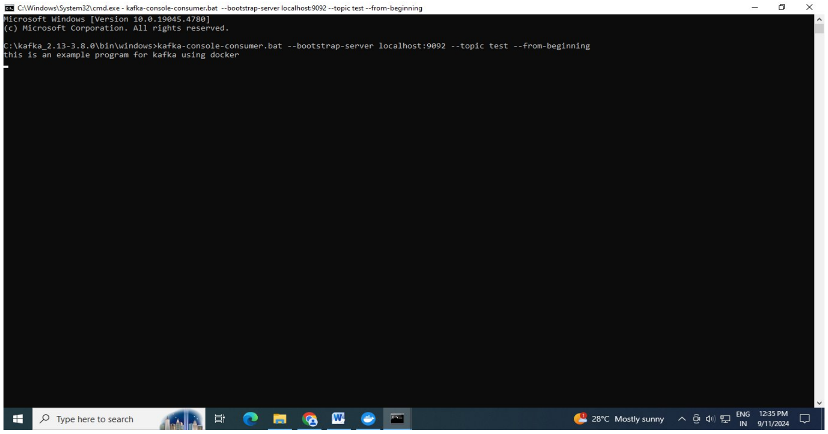
C:\kafka_2.13-3.8.0\bin\windows>kafka-console-producer.bat --broker-list localhost:9092 --topic test


C:\Windows\System32\cmd.exe - kafka-console-producer.bat --broker-list localhost:9092 --topic test
Type here to search
28°C Mostly sunny
ENG
IN
12:34 PM
9/11/2024

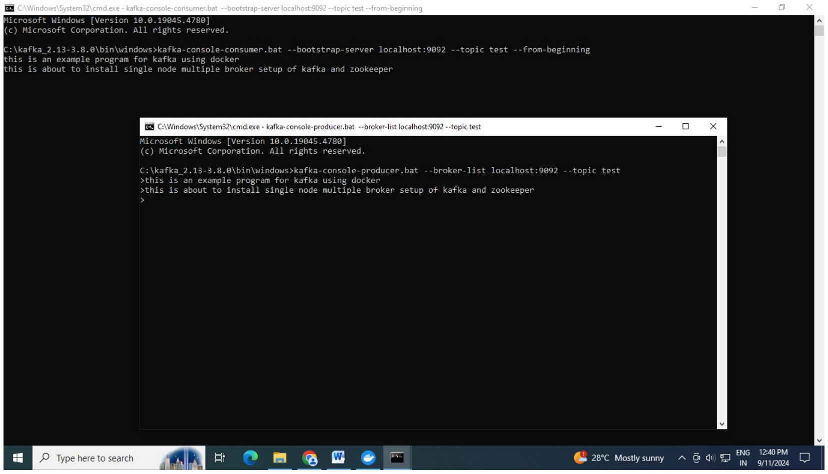
```
Microsoft Windows [Version 10.0.19045.4780]
(c) Microsoft Corporation. All rights reserved.

C:\kafka_2.13-3.8.0\bin\windows>kafka-console-producer.bat --broker-list localhost:9092 --topic test
>this is an example program for kafka using docker
>
```

STEP 8: Consume messages from beginning

```
C:\kafka_2.13-3.8.0\bin\windows>kafka-console-consumer.bat --bootstrap-server localhost:9092 --topic test --from-beginning
```


C:\Windows\System32\cmd.exe - kafka-console-consumer.bat --bootstrap-server localhost:9092 --topic test --from-beginning
Type here to search
28°C Mostly sunny
ENG IN
12:35 PM 9/11/2024

```
Microsoft Windows [Version 10.0.19045.4780]
(c) Microsoft Corporation. All rights reserved.

C:\kafka_2.13-3.8.0\bin\windows>kafka-console-consumer.bat --bootstrap-server localhost:9092 --topic test --from-beginning
this is an example program for kafka using docker
```


C:\Windows\System32\cmd.exe - kafka-console-consumer.bat --bootstrap-server localhost:9092 --topic test --from-beginning
C:\Windows\System32\cmd.exe - kafka-console-producer.bat --broker-list localhost:9092 --topic test
Type here to search
28°C Mostly sunny
ENG IN
12:40 PM 9/11/2024

```
Microsoft Windows [Version 10.0.19045.4780]
(c) Microsoft Corporation. All rights reserved.

C:\kafka_2.13-3.8.0\bin\windows>kafka-console-consumer.bat --bootstrap-server localhost:9092 --topic test --from-beginning
this is an example program for kafka using docker
this is about to install single node multiple broker setup of kafka and zookeeper
```

```
Microsoft Windows [Version 10.0.19045.4780]
(c) Microsoft Corporation. All rights reserved.

C:\kafka_2.13-3.8.0\bin\windows>kafka-console-producer.bat --broker-list localhost:9092 --topic test
>this is an example program for kafka using docker
>this is about to install single node multiple broker setup of kafka and zookeeper
>
```

EXPERIMENT NO: 6

NAME OF THE EXPERIMENT: Developing Kafka Producer.

STEP 1:enter into c drive□ kafka_2.13-3.8.0□C:\kafka_2.13-3.8.0\config□open server

log.dirs=C:\kafka_2.13-3.8.0\kafka-logs (replace temp with C:\kafka_2.13-3.8.0) and SAVE the file

STEP 2: Enter into c drive□

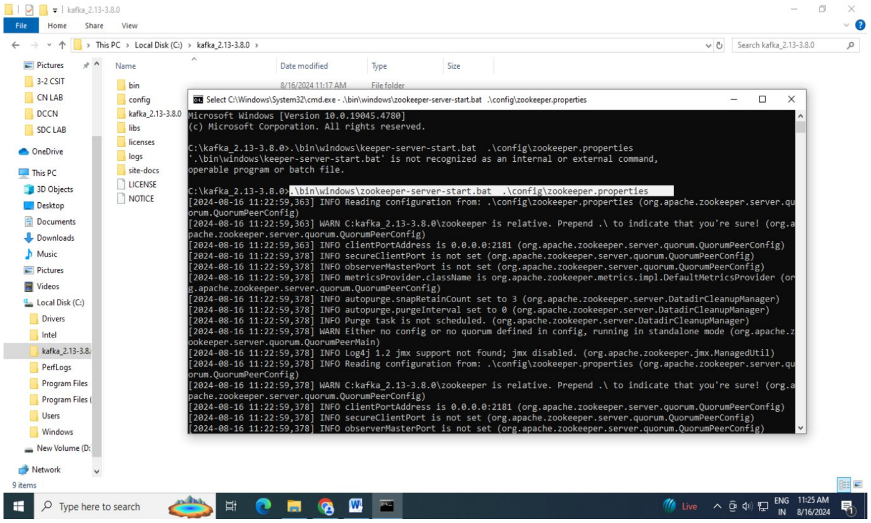
kafka_2.13-3.8.0□C:\kafka_2.13-3.8.0\config□open zookeeper

dataDir=C:\kafka_2.13-3.8.0\zookeeper (replace temp with C:\kafka_2.13-3.8.0) and SAVE the file

STEP 3: Open cmd and enter into the path C:\kafka_2.13-3.8.0

PATH: C:\kafka_2.13-3.8.0

TYPE:.\bin\windows\zookeeper-server-start.bat .\config\zookeeper.properties



Select C:\Windows\System32\cmd.exe - .\bin\windows\zookeeper-server-start.bat .\config\zookeeper.properties

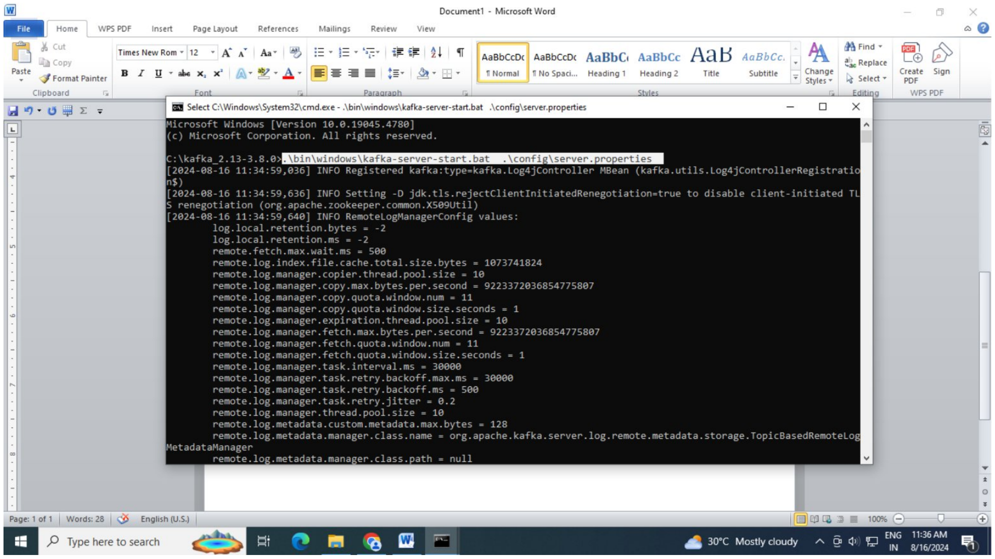
```
Microsoft Windows [Version 10.0.19045.4780]
(c) Microsoft Corporation. All rights reserved.

C:\kafka_2.13-3.8.0>.\bin\windows\keeper-server-start.bat  .\config\zookeeper.properties
'.\bin\windows\keeper-server-start.bat' is not recognized as an internal or external command,
operable program or batch file.

C:\kafka_2.13-3.8.0>.\bin\windows\zookeeper-server-start.bat  .\config\zookeeper.properties
[2024-08-16 11:22:59,363] INFO Reading configuration from: .\config\zookeeper.properties (org.apache.zookeeper.server.qu
orum.QuorumPeerConfig)
[2024-08-16 11:22:59,363] WARN C:kafka_2.13-3.8.0\zookeeper is relative. Prepend .\ to indicate that you're sure! (org.a
pache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-08-16 11:22:59,363] INFO clientPortAddress is 0.0.0.0:2181 (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-08-16 11:22:59,378] INFO secureClientPort is not set (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-08-16 11:22:59,378] INFO observerMasterPort is not set (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-08-16 11:22:59,378] INFO metricsProvider.className is org.apache.zookeeper.metrics.impl.DefaultMetricsProvider (or
g.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-08-16 11:22:59,378] INFO autopurge.snapRetainCount set to 3 (org.apache.zookeeper.server.DatadirCleanupManager)
[2024-08-16 11:22:59,378] INFO autopurge.purgeInterval set to 0 (org.apache.zookeeper.server.DatadirCleanupManager)
[2024-08-16 11:22:59,378] INFO Purge task is not scheduled. (org.apache.zookeeper.server.DatadirCleanupManager)
[2024-08-16 11:22:59,378] WARN Either no config or no quorum defined in config, running in standalone mode (org.apache.z
ookeeper.server.quorum.QuorumPeerMain)
[2024-08-16 11:22:59,378] INFO Log4j 1.2 jmx support not found; jmx disabled. (org.apache.zookeeper.jmx.ManagedUtil)
[2024-08-16 11:22:59,378] INFO Reading configuration from: .\config\zookeeper.properties (org.apache.zookeeper.server.qu
orum.QuorumPeerConfig)
[2024-08-16 11:22:59,378] WARN C:kafka_2.13-3.8.0\zookeeper is relative. Prepend .\ to indicate that you're sure! (org.a
pache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-08-16 11:22:59,378] INFO clientPortAddress is 0.0.0.0:2181 (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-08-16 11:22:59,378] INFO secureClientPort is not set (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-08-16 11:22:59,378] INFO observerMasterPort is not set (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
```

NOTE:DON’T CLOSE CMD

STEP 4: Open A NEW CMD and enter into the path C:\kafka_2.13-3.8.0

PATH: C:\kafka_2.13-3.8.0

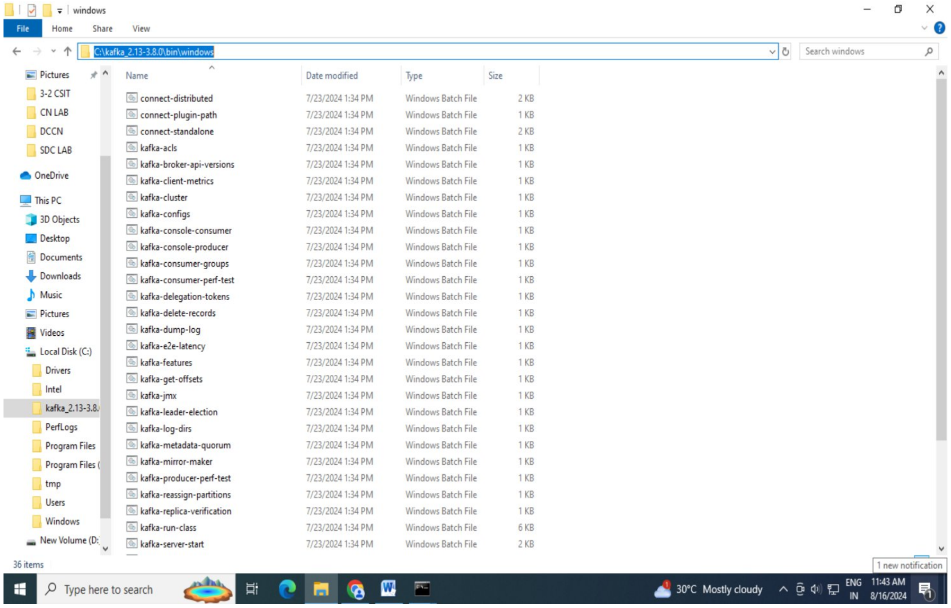
TYPE: .\bin\windows\kafka-server-start.bat .\config\server.properties



Select C:\Windows\System32\cmd.exe - .\bin\windows\kafka-server-start.bat .\config\server.properties

```
Microsoft Windows [Version 10.0.19045.4780]
(c) Microsoft Corporation. All rights reserved.

C:\kafka_2.13-3.8.0>.\bin\windows\kafka-server-start.bat  .\config\server.properties
[2024-08-16 11:34:59,036] INFO Registered kafka:type=kafka.Log4jController MBean (kafka.utils.Log4jControllerRegistratio
n$)
[2024-08-16 11:34:59,636] INFO Setting -D jdk.tls.rejectClientInitiatedRenegotiation=true to disable client-initiated TL
S renegotiation (org.apache.zookeeper.common.X509Util)
[2024-08-16 11:34:59,640] INFO RemoteLogManagerConfig values:
        log.local.retention.bytes = -2
        log.local.retention.ms = -2
        remote.fetch.max.wait.ms = 500
        remote.log.index.file.cache.total.size.bytes = 1073741824
        remote.log.manager.copier.thread.pool.size = 10
        remote.log.manager.copy.max.bytes.per.second = 9223372036854775807
        remote.log.manager.copy.quota.window.num = 11
        remote.log.manager.copy.quota.window.size.seconds = 1
        remote.log.manager.expiration.thread.pool.size = 10
        remote.log.manager.fetch.max.bytes.per.second = 9223372036854775807
        remote.log.manager.fetch.quota.window.num = 11
        remote.log.manager.fetch.quota.window.size.seconds = 1
        remote.log.manager.task.interval.ms = 30000
        remote.log.manager.task.retry.backoff.max.ms = 30000
        remote.log.manager.task.retry.backoff.ms = 500
        remote.log.manager.task.retry.jitter = 0.2
        remote.log.manager.thread.pool.size = 10
        remote.log.metadata.custom.metadata.max.bytes = 128
        remote.log.metadata.manager.class.name = org.apache.kafka.server.log.remote.metadata.storage.TopicBasedRemoteLog
MetadataManager
        remote.log.metadata.manager.class.path = null
```

STEP 5: Open A NEW CMD and enter into the path C:\kafka_2.13-3.8.0
PATH: C:\kafka_2.13-3.8.0
This step is to create a topic by using the below command

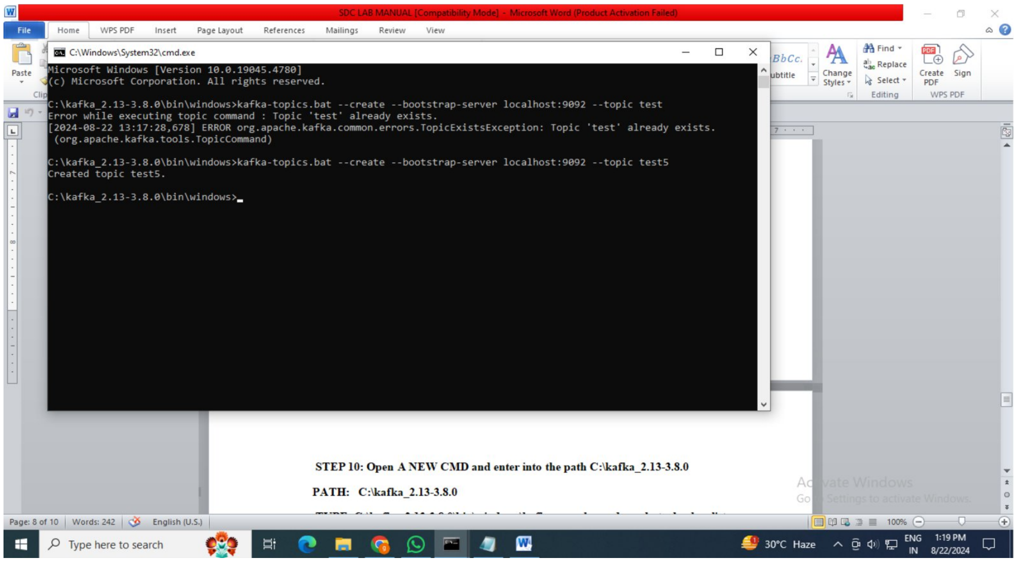


C:\kafka_2.13-3.8.0\bin\windows

Name	Date modified	Type	Size
connect-distributed	7/23/2024 1:34 PM	Windows Batch File	2 KB
connect-plugin-path	7/23/2024 1:34 PM	Windows Batch File	1 KB
connect-standalone	7/23/2024 1:34 PM	Windows Batch File	2 KB
kafka-acls	7/23/2024 1:34 PM	Windows Batch File	1 KB
kafka-broker-api-versions	7/23/2024 1:34 PM	Windows Batch File	1 KB
kafka-client-metrics	7/23/2024 1:34 PM	Windows Batch File	1 KB
kafka-cluster	7/23/2024 1:34 PM	Windows Batch File	1 KB
kafka-configs	7/23/2024 1:34 PM	Windows Batch File	1 KB
kafka-console-consumer	7/23/2024 1:34 PM	Windows Batch File	1 KB
kafka-console-producer	7/23/2024 1:34 PM	Windows Batch File	1 KB
kafka-consumer-groups	7/23/2024 1:34 PM	Windows Batch File	1 KB
kafka-consumer-perf-test	7/23/2024 1:34 PM	Windows Batch File	1 KB
kafka-delegation-tokens	7/23/2024 1:34 PM	Windows Batch File	1 KB
kafka-delete-records	7/23/2024 1:34 PM	Windows Batch File	1 KB
kafka-dump-log	7/23/2024 1:34 PM	Windows Batch File	1 KB
kafka-e2e-latency	7/23/2024 1:34 PM	Windows Batch File	1 KB
kafka-features	7/23/2024 1:34 PM	Windows Batch File	1 KB
kafka-get-offsets	7/23/2024 1:34 PM	Windows Batch File	1 KB
kafka-jmx	7/23/2024 1:34 PM	Windows Batch File	1 KB
kafka-leader-election	7/23/2024 1:34 PM	Windows Batch File	1 KB
kafka-log-dirs	7/23/2024 1:34 PM	Windows Batch File	1 KB
kafka-metadata-quorum	7/23/2024 1:34 PM	Windows Batch File	1 KB
kafka-mirror-maker	7/23/2024 1:34 PM	Windows Batch File	1 KB
kafka-producer-perf-test	7/23/2024 1:34 PM	Windows Batch File	1 KB
kafka-reassign-partitions	7/23/2024 1:34 PM	Windows Batch File	1 KB
kafka-replica-verification	7/23/2024 1:34 PM	Windows Batch File	1 KB
kafka-run-class	7/23/2024 1:34 PM	Windows Batch File	6 KB
kafka-server-start	7/23/2024 1:34 PM	Windows Batch File	2 KB

36 items

TYPE: C:\kafka_2.13-3.8.0\bin\windows\kafka-topics.bat –create --bootstrap-server localhost:9092--topic test



SDC LAB MANUAL [Compatibility Mode] - Microsoft Word (Product Activation Failed)
C:\Windows\System32\cmd.exe

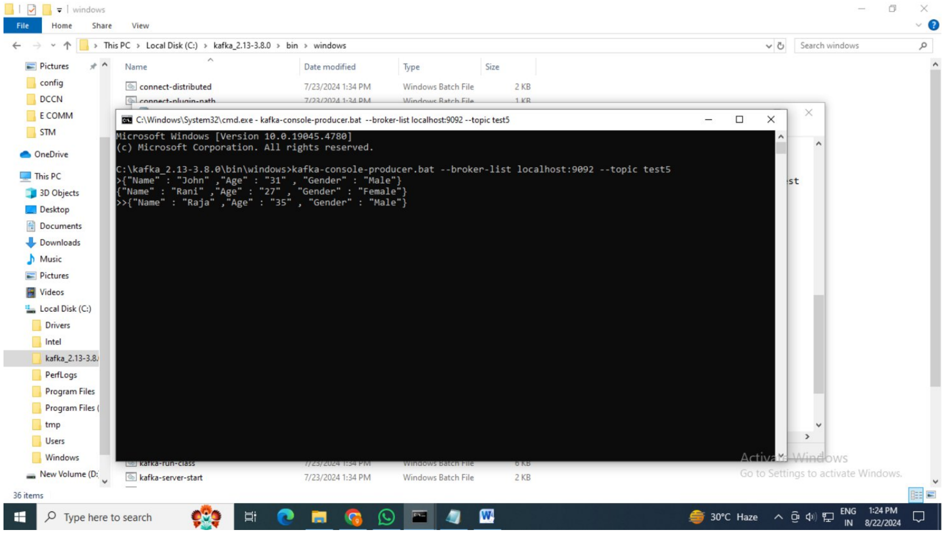
```
Microsoft Windows [Version 10.0.19045.4780]
(c) Microsoft Corporation. All rights reserved.

C:\kafka_2.13-3.8.0\bin\windows>kafka-topics.bat --create --bootstrap-server localhost:9092 --topic test
Error while executing topic command : Topic 'test' already exists.
[2024-08-22 13:17:28,678] ERROR org.apache.kafka.common.errors.TopicExistsException: Topic 'test' already exists.
 (org.apache.kafka.tools.TopicCommand)

C:\kafka_2.13-3.8.0\bin\windows>kafka-topics.bat --create --bootstrap-server localhost:9092 --topic test5
Created topic test5.

C:\kafka_2.13-3.8.0\bin\windows>
```

STEP 10: Open A NEW CMD and enter into the path C:\kafka_2.13-3.8.0

PATH: C:\kafka_2.13-3.8.0

STEP 6: Open A NEW CMD and enter into the path
C:\kafka_2.13-3.8.0
PATH: C:\kafka_2.13-3.8.0
TYPE: C:\kafka_2.13-3.8.0\bin\windows\kafka-console-producer.bat
--broker-list localhost:9092 --topic test
TYPE: (PRODUCER TAB)

{“Name” : “John” ,”Age” : “31” , “Gender” : “Male”}
{“Name” : “Raju” ,”Age” : “41” , “Gender” : “Male”}
{“Name” : “Rani” ,”Age” : “19” , “Gender” : “Female”}



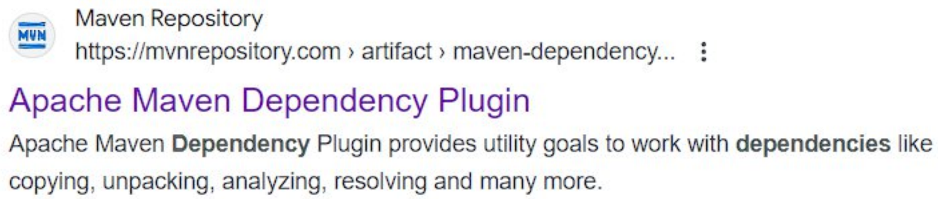
This PC › Local Disk (C:) › kafka_2.13-3.8.0 › bin › windows
C:\Windows\System32\cmd.exe - kafka-console-producer.bat --broker-list localhost:9092 --topic test5

```
Microsoft Windows [Version 10.0.19045.4780]
(c) Microsoft Corporation. All rights reserved.

C:\kafka_2.13-3.8.0\bin\windows>kafka-console-producer.bat --broker-list localhost:9092 --topic test5
>{"Name" : "John" ,"Age" : "31" , "Gender" : "Male"}
{"Name" : "Rani" ,"Age" : "27" , "Gender" : "Female"}
>>{"Name" : "Raja" ,"Age" : "35" , "Gender" : "Male"}
```

EXPERIMENT NO: 7

NAME OF THE EXPERIMENT: Sending Messages with Callback.

STEP 1: Open Google chrome and type “dependency download” and open the below link

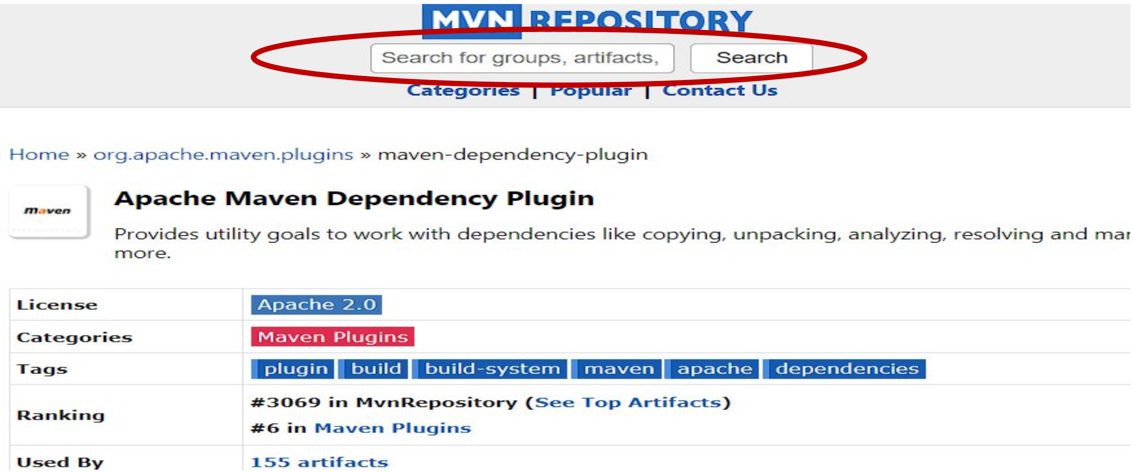


Maven Repository
https://mvnrepository.com › artifact › maven-dependency...

Apache Maven Dependency Plugin

Apache Maven **Dependency** Plugin provides utility goals to work with **dependencies** like copying, unpacking, analyzing, resolving and many more.

STEP 2: Search in the search bar as “kafka-client” then click on “org.apache.kafka>>kafka-client “ in Apache-kafka



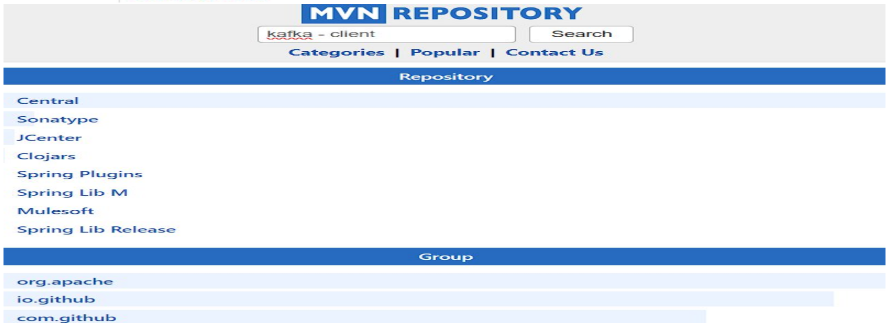
MVN REPOSITORY
Search for groups, artifacts,
Search
Categories | Popular | Contact Us

Home » org.apache.maven.plugins » maven-dependency-plugin

Apache Maven Dependency Plugin

Provides utility goals to work with dependencies like copying, unpacking, analyzing, resolving and mar more.

License	Apache 2.0
Categories	Maven Plugins
Tags	plugin build build-system maven apache dependencies
Ranking	#3069 in MvnRepository (See Top Artifacts) #6 in Maven Plugins
Used By	155 artifacts

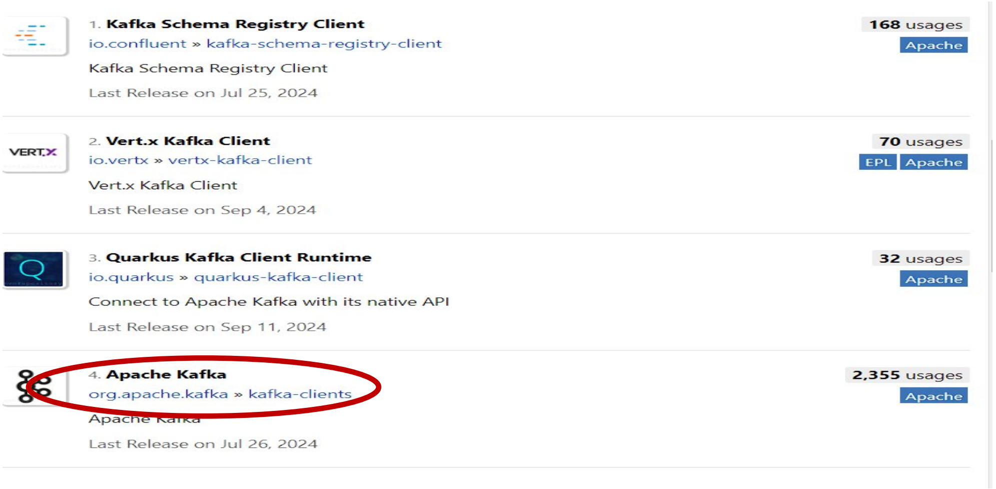


Repository

Central
Sonatype
JCenter
Clojars
Spring Plugins
Spring Lib M
Mulesoft
Spring Lib Release

Group

org.apache
io.github
com.github



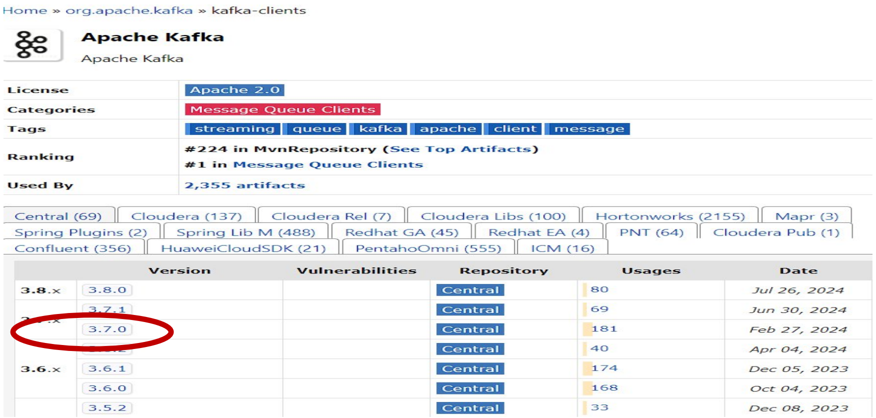
1. **Kafka Schema Registry Client** — 168 usages
io.confluent » kafka-schema-registry-client — Apache
Kafka Schema Registry Client
Last Release on Jul 25, 2024

2. **Vert.x Kafka Client** — 70 usages
io.vertx » vertx-kafka-client — EPL Apache
Vert.x Kafka Client
Last Release on Sep 4, 2024

3. **Quarkus Kafka Client Runtime** — 32 usages
io.quarkus » quarkus-kafka-client — Apache
Connect to Apache Kafka with its native API
Last Release on Sep 11, 2024

4. **Apache Kafka** — 2,355 usages
org.apache.kafka » kafka-clients — Apache
Apache Kafka
Last Release on Jul 26, 2024

- Select 3.7.0 version



Home » org.apache.kafka » kafka-clients

Apache Kafka

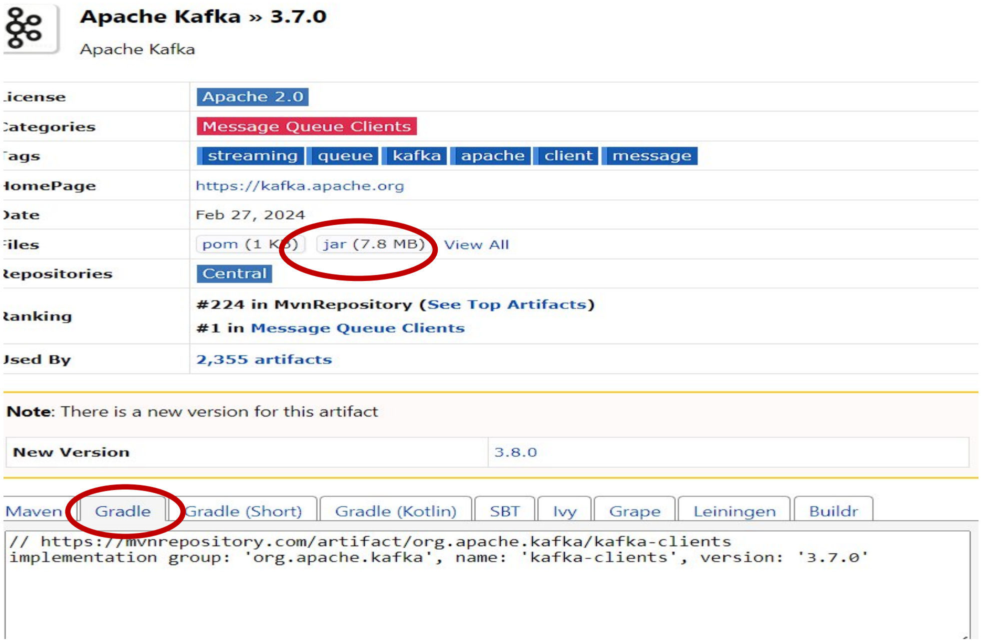
Apache Kafka

License	Apache 2.0
Categories	Message Queue Clients
Tags	streaming queue kafka apache client message
Ranking	**#224 in MvnRepository (See Top Artifacts)** **#1 in Message Queue Clients**
Used By	**2,355 artifacts**

Central (69) | Cloudera (137) | Cloudera Rel (7) | Cloudera Libs (100) | Hortonworks (2155) | Mapr (3) | Spring Plugins (2) | Spring Lib M (488) | Redhat GA (45) | Redhat EA (4) | PNT (64) | Cloudera Pub (1) | Confluent (356) | HuaweiCloudSDK (21) | PentahoOmni (555) | ICM (16)

	Version	Vulnerabilities	Repository	Usages	Date
3.8.x	3.8.0		Central	80	*Jul 26, 2024*
3.7.x	3.7.1		Central	69	*Jun 30, 2024*
	3.7.0		Central	181	*Feb 27, 2024*
3.6.x	3.6.2		Central	40	*Apr 04, 2024*
	3.6.1		Central	174	*Dec 05, 2023*
	3.6.0		Central	168	*Oct 04, 2023*
	3.5.2		Central	33	*Dec 08, 2023*

STEP 3: Select Gradle and click on jar(7.8 MB), then kafka-client dependencies are going to download



Apache Kafka » 3.7.0

Apache Kafka

icense	Apache 2.0
ategories	Message Queue Clients
ags	streaming queue kafka apache client message
lomePage	https://kafka.apache.org
)ate	Feb 27, 2024
iles	pom (1 KB) jar (7.8 MB) View All
tepositories	Central
tanking	**#224 in MvnRepository (See Top Artifacts)** **#1 in Message Queue Clients**
Jsed By	**2,355 artifacts**

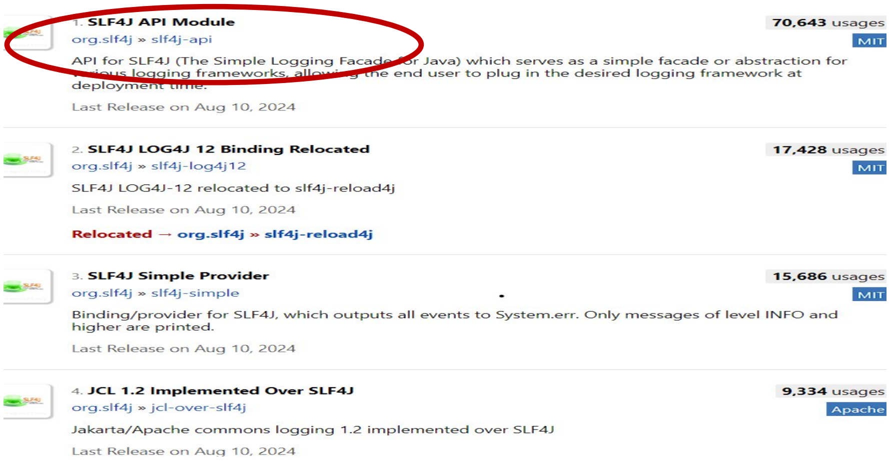
Note: There is a new version for this artifact

New Version	3.8.0

Maven | Gradle | Gradle (Short) | Gradle (Kotlin) | SBT | Ivy | Grape | Leiningen | Buildr

```
// https://mvnrepository.com/artifact/org.apache.kafka/kafka-clients
implementation group: 'org.apache.kafka', name: 'kafka-clients', version: '3.7.0'
```

STEP 4: :Again search in the search bar as " org.slf4j " then click on "org.slf4j >> slf4j-api" in the SLF4J API Module



1. **SLF4J API Module** — **70,643** usages — MIT

org.slf4j » slf4j-api

API for SLF4J (The Simple Logging Facade for Java) which serves as a simple facade or abstraction for various logging frameworks, allowing the end user to plug in the desired logging framework at deployment time.

Last Release on Aug 10, 2024

2. **SLF4J LOG4J 12 Binding Relocated** — **17,428** usages — MIT

org.slf4j » slf4j-log4j12

SLF4J LOG4J-12 relocated to slf4j-reload4j

Last Release on Aug 10, 2024

Relocated → org.slf4j » slf4j-reload4j

3. **SLF4J Simple Provider** — **15,686** usages — MIT

org.slf4j » slf4j-simple

Binding/provider for SLF4J, which outputs all events to System.err. Only messages of level INFO and higher are printed.

Last Release on Aug 10, 2024

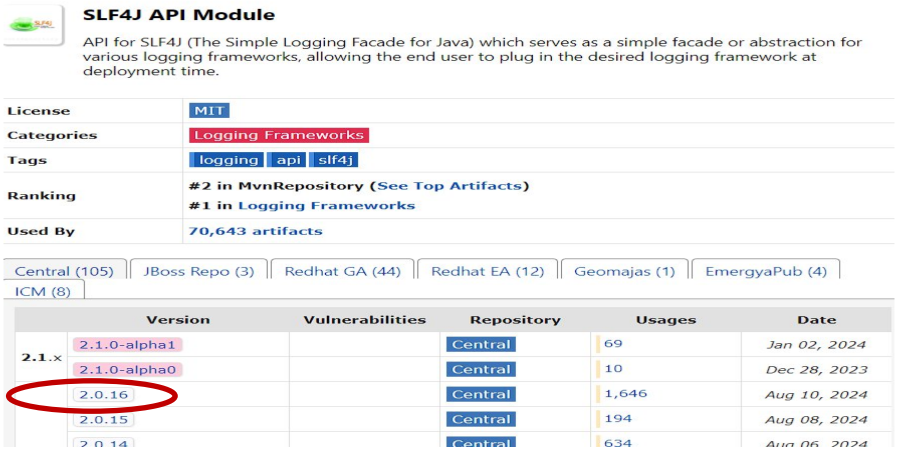
4. **JCL 1.2 Implemented Over SLF4J** — **9,334** usages — Apache

org.slf4j » jcl-over-slf4j

Jakarta/Apache commons logging 1.2 implemented over SLF4J

Last Release on Aug 10, 2024

❖ Select 2.0.16 version



SLF4J API Module

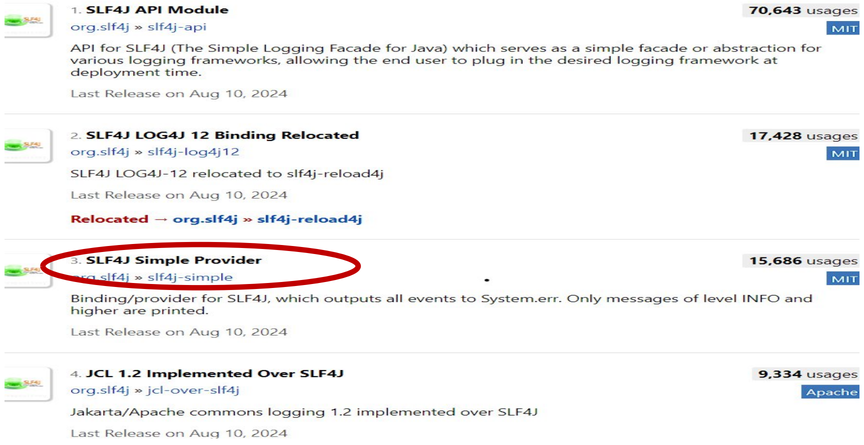
API for SLF4J (The Simple Logging Facade for Java) which serves as a simple facade or abstraction for various logging frameworks, allowing the end user to plug in the desired logging framework at deployment time.

License	MIT
Categories	Logging Frameworks
Tags	logging api slf4j
Ranking	#2 in MvnRepository (See Top Artifacts) #1 in Logging Frameworks
Used By	70,643 artifacts

Central (105) | JBoss Repo (3) | Redhat GA (44) | Redhat EA (12) | Geomajas (1) | EmergyaPub (4) | ICM (8)

	Version	Vulnerabilities	Repository	Usages	Date
2.1.x	2.1.0-alpha1		Central	69	Jan 02, 2024
	2.1.0-alpha0		Central	10	Dec 28, 2023
	2.0.16		Central	1,646	Aug 10, 2024
	2.0.15		Central	194	Aug 08, 2024
	2.0.14		Central	634	Aug 06, 2024

STEP 5: Select Gradle and click on jar, as like that search again as
“ org.slf4j ” then click on “org.slf4j >> slf4j-simple” in the SLF4J Simple Provider



1. **SLF4J API Module** — **70,643** usages
org.slf4j » slf4j-api — MIT
API for SLF4J (The Simple Logging Facade for Java) which serves as a simple facade or abstraction for various logging frameworks, allowing the end user to plug in the desired logging framework at deployment time.
Last Release on Aug 10, 2024

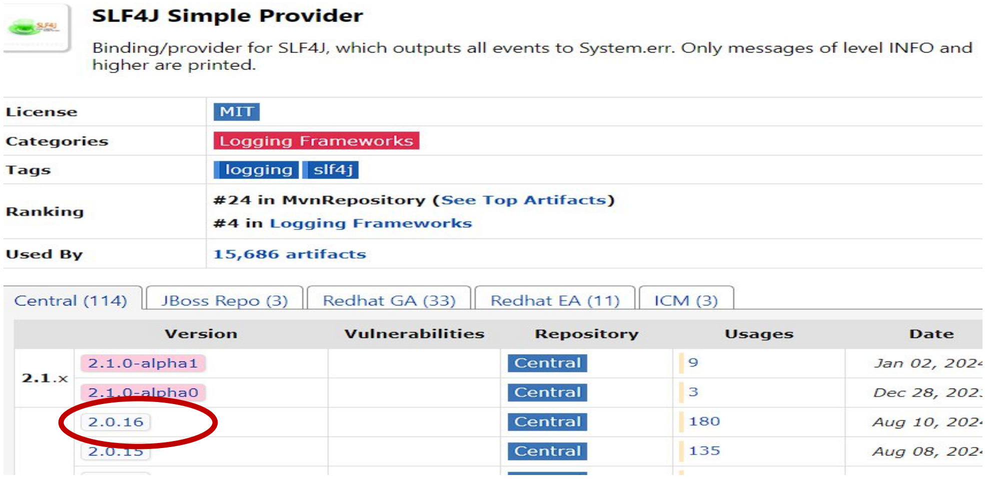
2. **SLF4J LOG4J 12 Binding Relocated** — **17,428** usages
org.slf4j » slf4j-log4j12 — MIT
SLF4J LOG4J-12 relocated to slf4j-reload4j
Last Release on Aug 10, 2024
Relocated → org.slf4j » slf4j-reload4j

3. **SLF4J Simple Provider** — **15,686** usages
org.slf4j » slf4j-simple — MIT
Binding/provider for SLF4J, which outputs all events to System.err. Only messages of level INFO and higher are printed.
Last Release on Aug 10, 2024

4. **JCL 1.2 Implemented Over SLF4J** — **9,334** usages
org.slf4j » jcl-over-slf4j — Apache
Jakarta/Apache commons logging 1.2 implemented over SLF4J
Last Release on Aug 10, 2024

❖ Select 2.0.16 version

STEP 6: Select Gradle and click on jar



SLF4J Simple Provider

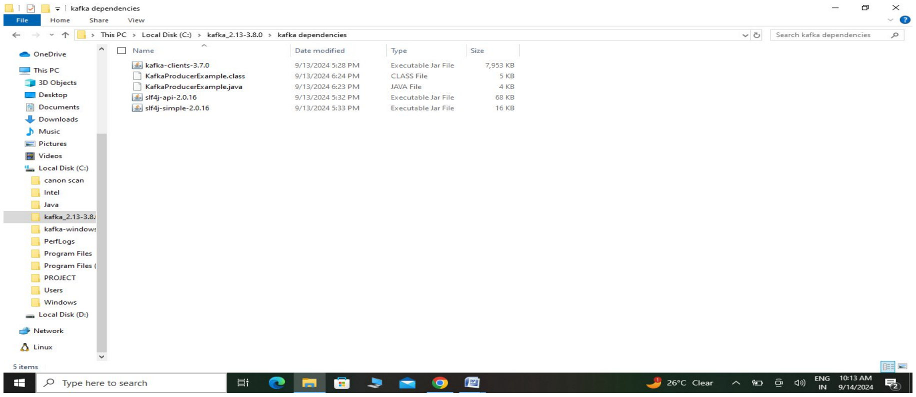
Binding/provider for SLF4J, which outputs all events to System.err. Only messages of level INFO and higher are printed.

License	MIT
Categories	Logging Frameworks
Tags	logging slf4j
Ranking	#24 in MvnRepository (See Top Artifacts) #4 in Logging Frameworks
Used By	15,686 artifacts

Central (114) | JBoss Repo (3) | Redhat GA (33) | Redhat EA (11) | ICM (3)

	Version	Vulnerabilities	Repository	Usages	Date
2.1.x	2.1.0-alpha1		Central	9	Jan 02, 202
	2.1.0-alpha0		Central	3	Dec 28, 202
	2.0.16		Central	180	Aug 10, 202
	2.0.15		Central	135	Aug 08, 202

STEP 7: Copy all the three dependencies from the downloads then create a new folder inside the kafka folder named as ” kafka dependencies”



Name	Date modified	Type	Size
kafka-clients-3.7.0	9/13/2024 5:28 PM	Executable Jar File	7,953 KB
KafkaProducerExample.class	9/13/2024 6:24 PM	CLASS File	5 KB
KafkaProducerExample.java	9/13/2024 6:23 PM	JAVA File	4 KB
slf4j-api-2.0.16	9/13/2024 5:32 PM	Executable Jar File	68 KB
slf4j-simple-2.0.16	9/13/2024 5:33 PM	Executable Jar File	16 KB

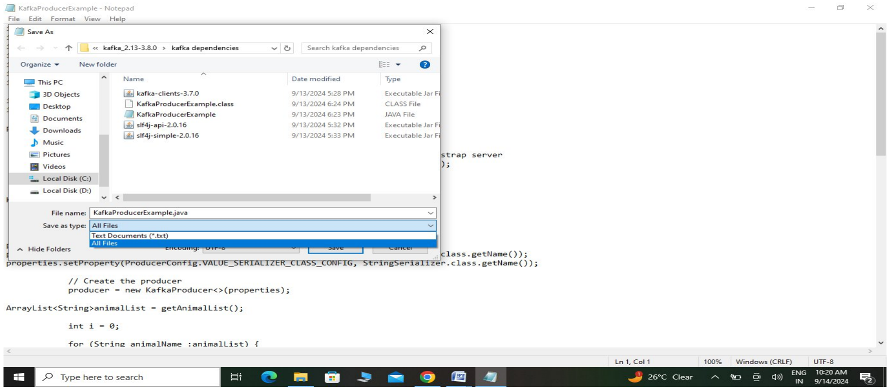
STEP 8:Write down the below code in the notepad and save it in the
“kafka dependencies folder”

KafkaProducerExample.java

```
import org.apache.kafka.clients.producer.KafkaProducer;
import org.apache.kafka.clients.producer.ProducerConfig;
import org.apache.kafka.clients.producer.ProducerRecord;
import org.apache.kafka.clients.producer.RecordMetadata;
import org.apache.kafka.common.serialization.StringSerializer;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

import java.util.ArrayList;
import java.util.Properties;

public class KafkaProducerExample {

    private final static String TOPIC_NAME = "AnimalTopic";
    private final static String bootstrapServers = "localhost:9092"; // Example bootstrap server
    public static Logger logger = LoggerFactory.getLogger(KafkaProducerExample.class);

    public static void main(String[] args) {

KafkaProducer<String, String> producer = null;
        try {

            // Create Producer properties
            Properties properties = new Properties();
properties.setProperty(ProducerConfig.BOOTSTRAP_SERVERS_CONFIG, bootstrapServers);
properties.setProperty(ProducerConfig.KEY_SERIALIZER_CLASS_CONFIG, StringSerializer.class.getName());
properties.setProperty(ProducerConfig.VALUE_SERIALIZER_CLASS_CONFIG, StringSerializer.class.getName());

            // Create the producer
            producer = new KafkaProducer<>(properties);

ArrayList<String>animalList = getAnimalList();
```

```
            int i = 0;

            for (String animalName :animalList) {

                ++i;
                String key = "Id_" + i;
                String value = animalName;  // Use animalName directly

                // Create a producer record
ProducerRecord<String, String>producerRecord = new ProducerRecord<>(TOPIC_NAME, key, value);
logger.info("key = " + key + ", value = " + value);

                // Send data - asynchronous
producer.send(producerRecord, (RecordMetadata recordMetadata, Exception e) -> {
                    if (e == null) {
logger.info("Successfully received the details as: \n"
                                + "Topic = " + recordMetadata.topic() + "\n"
                                + "Partition = " + recordMetadata.partition() + "\n"
                                + "Offset = " + recordMetadata.offset() + "\n"
                                + "Timestamp = " + recordMetadata.timestamp());
                    } else {
logger.error("Can't produce, getting error ", e);
                    }
                }).get(); // Make it synchronous for demo purposes

System.out.println("Successfully sent the Animal name = '" + value + "' to the Topic");

Thread.sleep(4000); // Simulating a delay for demo purposes
            }

        } catch (Exception exe) {
exe.printStackTrace();
        } finally {
            if (producer != null) {
producer.flush();
producer.close();
            }
        }
    }

    private static ArrayList<String>getAnimalList() {

ArrayList<String>animalList = new ArrayList<>();
animalList.add("DOG");
animalList.add("LION");
animalList.add("TIGER");
animalList.add("SNAKE");
animalList.add("CAT");

        return animalList;
    }
```

}



Name	Date modified	Type
kafka-clients-3.7.0	9/13/2024 5:28 PM	Executable Jar Fi
KafkaProducerExample.class	9/13/2024 6:24 PM	CLASS File
KafkaProducerExample	9/13/2024 6:23 PM	JAVA File
slf4j-api-2.0.16	9/13/2024 5:32 PM	Executable Jar Fi
slf4j-simple-2.0.16	9/13/2024 5:33 PM	Executable Jar Fi

File name: KafkaProducerExample.java

Save as type: All Files

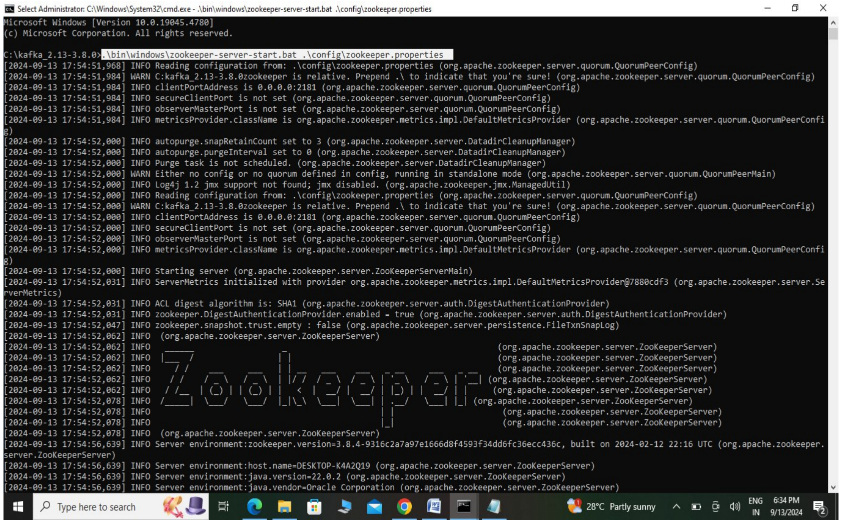
Text Documents (*.txt)
All Files

```
strap server
);

class.getName());
properties.setProperty(ProducerConfig.VALUE_SERIALIZER_CLASS_CONFIG, StringSerializer.class.getName());

            // Create the producer
            producer = new KafkaProducer<>(properties);

ArrayList<String>animalList = getAnimalList();

            int i = 0;

            for (String animalName :animalList) {
```

STEP 9: Run Zookeeper and server in kafka by entering into the below command with two different cmd's (path--C:\kafka_2.13-3.8.0)

Command:

```
.\bin\windows\zookeeper-server-start.bat .\config\zookeeper.properties
```

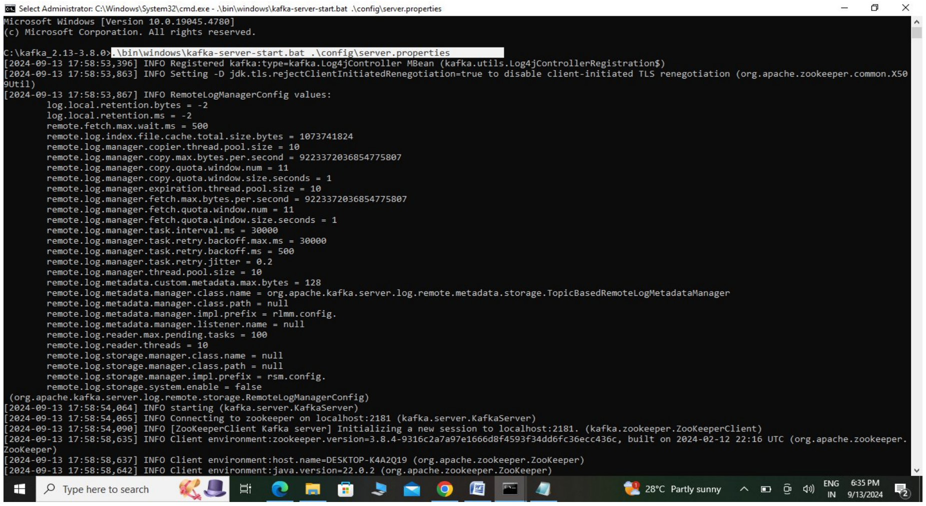


Select Administrator: C:\Windows\System32\cmd.exe - .\bin\windows\zookeeper-server-start.bat .\config\zookeeper.properties
Type here to search
28°C Partly sunny
ENG IN
6:34 PM 9/13/2024

```
Microsoft Windows [Version 10.0.19045.4780]
(c) Microsoft Corporation. All rights reserved.

C:\kafka_2.13-3.8.0>.\bin\windows\zookeeper-server-start.bat .\config\zookeeper.properties
[2024-09-13 17:54:51,968] INFO Reading configuration from: .\config\zookeeper.properties (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-09-13 17:54:51,984] WARN C:kafka_2.13-3.8.0zookeeper is relative. Prepend .\ to indicate that you're sure! (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-09-13 17:54:51,984] INFO clientPortAddress is 0.0.0.0:2181 (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-09-13 17:54:51,984] INFO secureClientPort is not set (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-09-13 17:54:51,984] INFO observerMasterPort is not set (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-09-13 17:54:51,984] INFO metricsProvider.className is org.apache.zookeeper.metrics.impl.DefaultMetricsProvider (org.apache.zookeeper.server.quorum.QuorumPeerConfi
g)
[2024-09-13 17:54:52,000] INFO autopurge.snapRetainCount set to 3 (org.apache.zookeeper.server.DatadirCleanupManager)
[2024-09-13 17:54:52,000] INFO autopurge.purgeInterval set to 0 (org.apache.zookeeper.server.DatadirCleanupManager)
[2024-09-13 17:54:52,000] INFO Purge task is not scheduled. (org.apache.zookeeper.server.DatadirCleanupManager)
[2024-09-13 17:54:52,000] WARN Either no config or no quorum defined in config, running in standalone mode (org.apache.zookeeper.server.quorum.QuorumPeerMain)
[2024-09-13 17:54:52,000] INFO Log4j 1.2 jmx support not found; jmx disabled. (org.apache.zookeeper.jmx.ManagedUtil)
[2024-09-13 17:54:52,000] INFO Reading configuration from: .\config\zookeeper.properties (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-09-13 17:54:52,000] WARN C:kafka_2.13-3.8.0zookeeper is relative. Prepend .\ to indicate that you're sure! (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-09-13 17:54:52,000] INFO clientPortAddress is 0.0.0.0:2181 (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-09-13 17:54:52,000] INFO secureClientPort is not set (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-09-13 17:54:52,000] INFO observerMasterPort is not set (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-09-13 17:54:52,000] INFO metricsProvider.className is org.apache.zookeeper.metrics.impl.DefaultMetricsProvider (org.apache.zookeeper.server.quorum.QuorumPeerConfi
g)
[2024-09-13 17:54:52,000] INFO Starting server (org.apache.zookeeper.server.ZooKeeperServerMain)
[2024-09-13 17:54:52,031] INFO ServerMetrics initialized with provider org.apache.zookeeper.metrics.impl.DefaultMetricsProvider@7880cdf3 (org.apache.zookeeper.server.Se
rverMetrics)
[2024-09-13 17:54:52,031] INFO ACL digest algorithm is: SHA1 (org.apache.zookeeper.server.auth.DigestAuthenticationProvider)
[2024-09-13 17:54:52,031] INFO zookeeper.DigestAuthenticationProvider.enabled = true (org.apache.zookeeper.server.auth.DigestAuthenticationProvider)
[2024-09-13 17:54:52,047] INFO zookeeper.snapshot.trust.empty : false (org.apache.zookeeper.server.persistence.FileTxnSnapLog)
[2024-09-13 17:54:52,062] INFO  (org.apache.zookeeper.server.ZooKeeperServer)
[2024-09-13 17:54:52,062] INFO   ______                  _                                          (org.apache.zookeeper.server.ZooKeeperServer)
[2024-09-13 17:54:52,062] INFO  |___  /                 | |                                         (org.apache.zookeeper.server.ZooKeeperServer)
[2024-09-13 17:54:52,062] INFO     / /    ___     ___   | | __   ___    ___   _ __     ___   _ __   (org.apache.zookeeper.server.ZooKeeperServer)
[2024-09-13 17:54:52,062] INFO    / /    / _ \   / _ \  | |/ /  / _ \  / _ \ | '_ \   / _ \ | '__| (org.apache.zookeeper.server.ZooKeeperServer)
[2024-09-13 17:54:52,062] INFO   / /__  | (_) | | (_) | |   <  |  __/ |  __/ | |_) | |  __/ | |     (org.apache.zookeeper.server.ZooKeeperServer)
[2024-09-13 17:54:52,078] INFO  /_____|  \___/   \___/  |_|\_\  \___|  \___| | .__/   \___| |_| (org.apache.zookeeper.server.ZooKeeperServer)
[2024-09-13 17:54:52,078] INFO                                               | |                      (org.apache.zookeeper.server.ZooKeeperServer)
[2024-09-13 17:54:52,078] INFO                                               |_|                      (org.apache.zookeeper.server.ZooKeeperServer)
[2024-09-13 17:54:52,078] INFO  (org.apache.zookeeper.server.ZooKeeperServer)
[2024-09-13 17:54:56,639] INFO Server environment:zookeeper.version=3.8.4-9316c2a7a97e1666d8f4593f34dd6fc36ecc436c, built on 2024-02-12 22:16 UTC (org.apache.zookeeper.
server.ZooKeeperServer)
[2024-09-13 17:54:56,639] INFO Server environment:host.name=DESKTOP-K4A2Q19 (org.apache.zookeeper.server.ZooKeeperServer)
[2024-09-13 17:54:56,639] INFO Server environment:java.version=22.0.2 (org.apache.zookeeper.server.ZooKeeperServer)
[2024-09-13 17:54:56,639] INFO Server environment:java.vendor=Oracle Corporation (org.apache.zookeeper.server.ZooKeeperServer)
```

Command:

```
.\bin\windows\kafka-server-start.bat .\config\server.properties
```



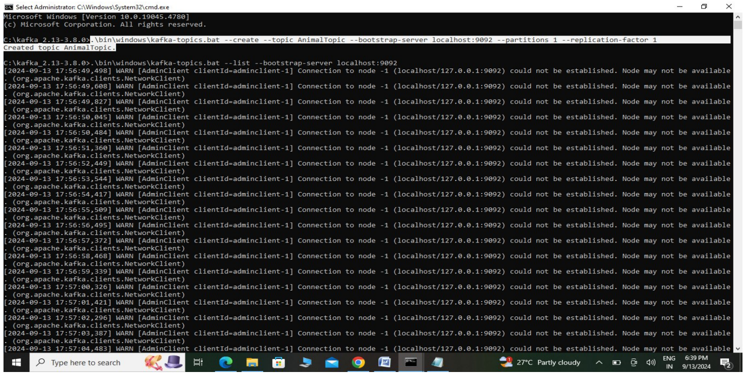
Select Administrator: C:\Windows\System32\cmd.exe - .\bin\windows\kafka-server-start.bat .\config\server.properties
Type here to search
28°C Partly sunny
ENG IN
6:35 PM 9/13/2024

```
Microsoft Windows [Version 10.0.19045.4780]
(c) Microsoft Corporation. All rights reserved.

C:\kafka_2.13-3.8.0>.\bin\windows\kafka-server-start.bat .\config\server.properties
[2024-09-13 17:58:53,396] INFO Registered kafka:type=kafka.Log4jController MBean (kafka.utils.Log4jControllerRegistration$)
[2024-09-13 17:58:53,863] INFO Setting -D jdk.tls.rejectClientInitiatedRenegotiation=true to disable client-initiated TLS renegotiation (org.apache.zookeeper.common.X50
9Util)
[2024-09-13 17:58:53,867] INFO RemoteLogManagerConfig values:
        log.local.retention.bytes = -2
        log.local.retention.ms = -2
        remote.fetch.max.wait.ms = 500
        remote.log.index.file.cache.total.size.bytes = 1073741824
        remote.log.manager.copier.thread.pool.size = 10
        remote.log.manager.copy.max.bytes.per.second = 9223372036854775807
        remote.log.manager.copy.quota.window.num = 11
        remote.log.manager.copy.quota.window.size.seconds = 1
        remote.log.manager.expiration.thread.pool.size = 10
        remote.log.manager.fetch.max.bytes.per.second = 9223372036854775807
        remote.log.manager.fetch.quota.window.num = 11
        remote.log.manager.fetch.quota.window.size.seconds = 1
        remote.log.manager.task.interval.ms = 30000
        remote.log.manager.task.retry.backoff.max.ms = 30000
        remote.log.manager.task.retry.backoff.ms = 500
        remote.log.manager.task.retry.jitter = 0.2
        remote.log.manager.thread.pool.size = 10
        remote.log.metadata.custom.metadata.max.bytes = 128
        remote.log.metadata.manager.class.name = org.apache.kafka.server.log.remote.metadata.storage.TopicBasedRemoteLogMetadataManager
        remote.log.metadata.manager.class.path = null
        remote.log.metadata.manager.impl.prefix = rlmm.config.
        remote.log.metadata.manager.listener.name = null
        remote.log.reader.max.pending.tasks = 100
        remote.log.reader.threads = 10
        remote.log.storage.manager.class.name = null
        remote.log.storage.manager.class.path = null
        remote.log.storage.manager.impl.prefix = rsm.config.
        remote.log.storage.system.enable = false
 (org.apache.kafka.server.log.remote.storage.RemoteLogManagerConfig)
[2024-09-13 17:58:54,064] INFO starting (kafka.server.KafkaServer)
[2024-09-13 17:58:54,065] INFO Connecting to zookeeper on localhost:2181 (kafka.server.KafkaServer)
[2024-09-13 17:58:54,090] INFO [ZooKeeperClient Kafka server] Initializing a new session to localhost:2181. (kafka.zookeeper.ZooKeeperClient)
[2024-09-13 17:58:58,635] INFO Client environment:zookeeper.version=3.8.4-9316c2a7a97e1666d8f4593f34dd6fc36ecc436c, built on 2024-02-12 22:16 UTC (org.apache.zookeeper.
ZooKeeper)
[2024-09-13 17:58:58,637] INFO Client environment:host.name=DESKTOP-K4A2Q19 (org.apache.zookeeper.ZooKeeper)
[2024-09-13 17:58:58,642] INFO Client environment:java.version=22.0.2 (org.apache.zookeeper.ZooKeeper)
```

STEP 10: Creating a topic called AnimalTopic

Command:

```
.\bin\windows\kafka-topics.bat --create --topic AnimalTopic --bootstrap-server localhost:9092 --partitions 1 --replication-factor 1
```

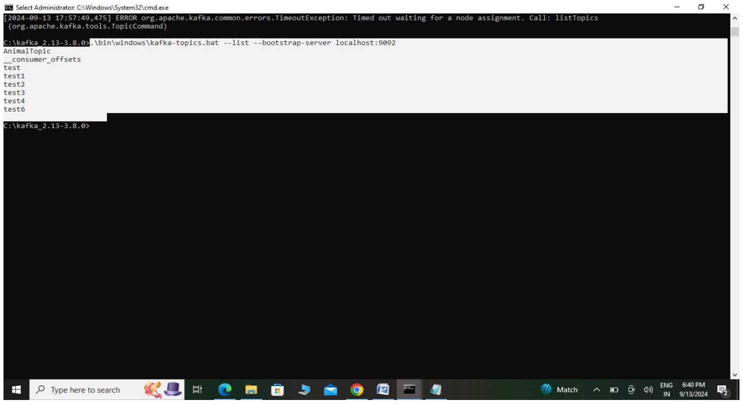


```
Microsoft Windows [Version 10.0.19045.4780]
(c) Microsoft Corporation. All rights reserved.

C:\kafka_2.13-3.8.0>.\bin\windows\kafka-topics.bat --create --topic AnimalTopic --bootstrap-server localhost:9092 --partitions 1 --replication-factor 1
Created topic AnimalTopic.

C:\kafka_2.13-3.8.0>.\bin\windows\kafka-topics.bat --list --bootstrap-server localhost:9092
[2024-09-13 17:56:49,498] WARN [AdminClient clientId=adminclient-1] Connection to node -1 (localhost/127.0.0.1:9092) could not be established. Node may not be available
. (org.apache.kafka.clients.NetworkClient)
[2024-09-13 17:56:49,608] WARN [AdminClient clientId=adminclient-1] Connection to node -1 (localhost/127.0.0.1:9092) could not be established. Node may not be available
. (org.apache.kafka.clients.NetworkClient)
[2024-09-13 17:56:49,827] WARN [AdminClient clientId=adminclient-1] Connection to node -1 (localhost/127.0.0.1:9092) could not be established. Node may not be available
. (org.apache.kafka.clients.NetworkClient)
[2024-09-13 17:56:50,045] WARN [AdminClient clientId=adminclient-1] Connection to node -1 (localhost/127.0.0.1:9092) could not be established. Node may not be available
. (org.apache.kafka.clients.NetworkClient)
[2024-09-13 17:56:50,484] WARN [AdminClient clientId=adminclient-1] Connection to node -1 (localhost/127.0.0.1:9092) could not be established. Node may not be available
. (org.apache.kafka.clients.NetworkClient)
[2024-09-13 17:56:51,360] WARN [AdminClient clientId=adminclient-1] Connection to node -1 (localhost/127.0.0.1:9092) could not be established. Node may not be available
. (org.apache.kafka.clients.NetworkClient)
[2024-09-13 17:56:52,449] WARN [AdminClient clientId=adminclient-1] Connection to node -1 (localhost/127.0.0.1:9092) could not be established. Node may not be available
. (org.apache.kafka.clients.NetworkClient)
[2024-09-13 17:56:53,544] WARN [AdminClient clientId=adminclient-1] Connection to node -1 (localhost/127.0.0.1:9092) could not be established. Node may not be available
. (org.apache.kafka.clients.NetworkClient)
[2024-09-13 17:56:54,417] WARN [AdminClient clientId=adminclient-1] Connection to node -1 (localhost/127.0.0.1:9092) could not be established. Node may not be available
. (org.apache.kafka.clients.NetworkClient)
[2024-09-13 17:56:55,509] WARN [AdminClient clientId=adminclient-1] Connection to node -1 (localhost/127.0.0.1:9092) could not be established. Node may not be available
. (org.apache.kafka.clients.NetworkClient)
[2024-09-13 17:56:56,495] WARN [AdminClient clientId=adminclient-1] Connection to node -1 (localhost/127.0.0.1:9092) could not be established. Node may not be available
. (org.apache.kafka.clients.NetworkClient)
[2024-09-13 17:56:57,372] WARN [AdminClient clientId=adminclient-1] Connection to node -1 (localhost/127.0.0.1:9092) could not be established. Node may not be available
. (org.apache.kafka.clients.NetworkClient)
[2024-09-13 17:56:58,468] WARN [AdminClient clientId=adminclient-1] Connection to node -1 (localhost/127.0.0.1:9092) could not be established. Node may not be available
. (org.apache.kafka.clients.NetworkClient)
[2024-09-13 17:56:59,339] WARN [AdminClient clientId=adminclient-1] Connection to node -1 (localhost/127.0.0.1:9092) could not be established. Node may not be available
. (org.apache.kafka.clients.NetworkClient)
[2024-09-13 17:57:00,326] WARN [AdminClient clientId=adminclient-1] Connection to node -1 (localhost/127.0.0.1:9092) could not be established. Node may not be available
. (org.apache.kafka.clients.NetworkClient)
[2024-09-13 17:57:01,421] WARN [AdminClient clientId=adminclient-1] Connection to node -1 (localhost/127.0.0.1:9092) could not be established. Node may not be available
. (org.apache.kafka.clients.NetworkClient)
[2024-09-13 17:57:02,296] WARN [AdminClient clientId=adminclient-1] Connection to node -1 (localhost/127.0.0.1:9092) could not be established. Node may not be available
. (org.apache.kafka.clients.NetworkClient)
[2024-09-13 17:57:03,387] WARN [AdminClient clientId=adminclient-1] Connection to node -1 (localhost/127.0.0.1:9092) could not be established. Node may not be available
. (org.apache.kafka.clients.NetworkClient)
[2024-09-13 17:57:04,483] WARN [AdminClient clientId=adminclient-1] Connection to node -1 (localhost/127.0.0.1:9092) could not be established. Node may not be available
```

STEP 11: To Listdown all the topics

Command:

```
.\bin\windows\kafka-topics.bat --list --bootstrap-server localhost:9092
```



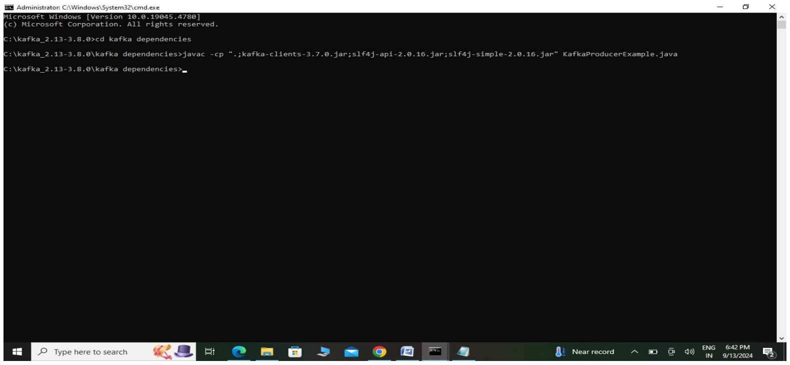
```
[2024-09-13 17:57:49,475] ERROR org.apache.kafka.common.errors.TimeoutException: Timed out waiting for a node assignment. Call: listTopics
 (org.apache.kafka.tools.TopicCommand)

C:\kafka_2.13-3.8.0>.\bin\windows\kafka-topics.bat --list --bootstrap-server localhost:9092
AnimalTopic
__consumer_offsets
test
test1
test2
test3
test4
test6

C:\kafka_2.13-3.8.0>
```

STEP 12: Enter into the folder "kafka dependencies" by using the command

"cd kafka dependencies"

Compilation of the Code:

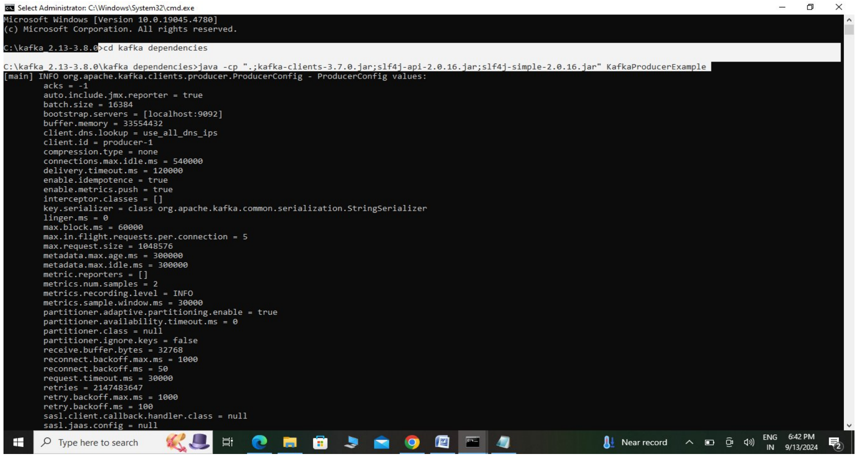
```
javac -cp ".;kafka-clients-3.7.0.jar;slf4j-api-2.0.16.jar;slf4j-simple-2.0.16.jar" KafkaProducerExample.java
```



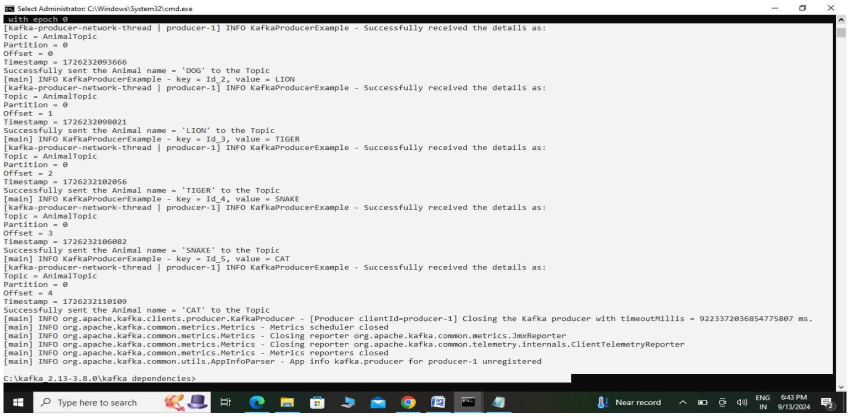
```
Microsoft Windows [Version 10.0.19045.4780]
(c) Microsoft Corporation. All rights reserved.

C:\kafka_2.13-3.8.0>cd kafka dependencies

C:\kafka_2.13-3.8.0\kafka dependencies>javac -cp ".;kafka-clients-3.7.0.jar;slf4j-api-2.0.16.jar;slf4j-simple-2.0.16.jar" KafkaProducerExample.java

C:\kafka_2.13-3.8.0\kafka dependencies>
```

STEP 13:Run the Code using the below command

java -cp ".;kafka-clients-3.7.0.jar;slf4j-api-2.0.16.jar;slf4j-simple-2.0.16.jar" KafkaProducerExample



```
Microsoft Windows [Version 10.0.19045.4780]
(c) Microsoft Corporation. All rights reserved.

C:\kafka_2.13-3.8.0>cd kafka dependencies

C:\kafka_2.13-3.8.0\kafka dependencies>java -cp ".;kafka-clients-3.7.0.jar;slf4j-api-2.0.16.jar;slf4j-simple-2.0.16.jar" KafkaProducerExample
[main] INFO org.apache.kafka.clients.producer.ProducerConfig - ProducerConfig values:
        acks = -1
        auto.include.jmx.reporter = true
        batch.size = 16384
        bootstrap.servers = [localhost:9092]
        buffer.memory = 33554432
        client.dns.lookup = use_all_dns_ips
        client.id = producer-1
        compression.type = none
        connections.max.idle.ms = 540000
        delivery.timeout.ms = 120000
        enable.idempotence = true
        enable.metrics.push = true
        interceptor.classes = []
        key.serializer = class org.apache.kafka.common.serialization.StringSerializer
        linger.ms = 0
        max.block.ms = 60000
        max.in.flight.requests.per.connection = 5
        max.request.size = 1048576
        metadata.max.age.ms = 300000
        metadata.max.idle.ms = 300000
        metric.reporters = []
        metrics.num.samples = 2
        metrics.recording.level = INFO
        metrics.sample.window.ms = 30000
        partitioner.adaptive.partitioning.enable = true
        partitioner.availability.timeout.ms = 0
        partitioner.class = null
        partitioner.ignore.keys = false
        receive.buffer.bytes = 32768
        reconnect.backoff.max.ms = 1000
        reconnect.backoff.ms = 50
        request.timeout.ms = 30000
        retries = 2147483647
        retry.backoff.max.ms = 1000
        retry.backoff.ms = 100
        sasl.client.callback.handler.class = null
        sasl.jaas.config = null
```



```
 with epoch 0
[kafka-producer-network-thread | producer-1] INFO KafkaProducerExample - Successfully received the details as:
Topic = AnimalTopic
Partition = 0
Offset = 0
Timestamp = 1726232093666
Successfully sent the Animal name = 'DOG' to the Topic
[main] INFO KafkaProducerExample - key = Id_2, value = LION
[kafka-producer-network-thread | producer-1] INFO KafkaProducerExample - Successfully received the details as:
Topic = AnimalTopic
Partition = 0
Offset = 1
Timestamp = 1726232098021
Successfully sent the Animal name = 'LION' to the Topic
[main] INFO KafkaProducerExample - key = Id_3, value = TIGER
[kafka-producer-network-thread | producer-1] INFO KafkaProducerExample - Successfully received the details as:
Topic = AnimalTopic
Partition = 0
Offset = 2
Timestamp = 1726232102056
Successfully sent the Animal name = 'TIGER' to the Topic
[main] INFO KafkaProducerExample - key = Id_4, value = SNAKE
[kafka-producer-network-thread | producer-1] INFO KafkaProducerExample - Successfully received the details as:
Topic = AnimalTopic
Partition = 0
Offset = 3
Timestamp = 1726232106082
Successfully sent the Animal name = 'SNAKE' to the Topic
[main] INFO KafkaProducerExample - key = Id_5, value = CAT
[kafka-producer-network-thread | producer-1] INFO KafkaProducerExample - Successfully received the details as:
Topic = AnimalTopic
Partition = 0
Offset = 4
Timestamp = 1726232110109
Successfully sent the Animal name = 'CAT' to the Topic
[main] INFO org.apache.kafka.clients.producer.KafkaProducer - [Producer clientId=producer-1] Closing the Kafka producer with timeoutMillis = 9223372036854775807 ms.
[main] INFO org.apache.kafka.common.metrics.Metrics - Metrics scheduler closed
[main] INFO org.apache.kafka.common.metrics.Metrics - Closing reporter org.apache.kafka.common.metrics.JmxReporter
[main] INFO org.apache.kafka.common.metrics.Metrics - Closing reporter org.apache.kafka.common.telemetry.internals.ClientTelemetryReporter
[main] INFO org.apache.kafka.common.metrics.Metrics - Metrics reporters closed
[main] INFO org.apache.kafka.common.utils.AppInfoParser - App info kafka.producer for producer-1 unregistered

C:\kafka_2.13-3.8.0\kafka dependencies>
```

EXPERIMENT NO: 8

NAME OF THE EXPERIMENT: Kafka monitoring and schema Registry.

Objective:

1. Set up and configure Kafka monitoring to track Kafka metrics.
2. Set up and use Schema Registry to manage message schemas in Kafka.

Prerequisites

- Apache Kafka and Apache Zookeeper installed and running.
- Confluent Schema Registry installed (comes with Confluent Platform) or downloaded separately.
- Prometheus for monitoring metrics.
- Grafana for data visualization.
- Java Development Kit (JDK 8 or above).
- Kafka client libraries and Confluent libraries (for Schema Registry).

Start Kafka and Zookeeper:

Start Zookeeper:

```
bin/zookeeper-server-start.sh config/zookeeper.properties
```

Start Kafka:

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

Create a Kafka Topic for testing schemas (e.g., schema-topic):

```
bin/kafka-topics.sh --create --topic schema-topic --bootstrap-server localhost:9092 --partitions 1 --replication-factor 1
```

Configure Kafka Monitoring with Prometheus and Grafana

Install and Configure JMX Exporter for Kafka:

1. Download the JMX Exporter .jar file from Prometheus's website.
2. Place the .jar file in your Kafka directory and create a configuration file kafka-jmx-exporter.yml:

```
lowercaseOutputName: true
rules:
  - pattern: "kafka.server<type=(.+), name=(.+)>"
  name: "kafka_server_$1_$2"
   labels:
  instance: "kafka"
 help: "Kafka server metrics"
```

Edit the Kafka Server Configuration (config/server.properties) to enable JMX Exporter by adding the following line:

```
KAFKA_OPTS="-javaagent:/path/to/jmx_prometheus_javaagent.jar=8080:/path/to/kafka-jmx-exporter.yml"
```

Restart Kafka for the JMX exporter changes to take effect.

Set Up Prometheus to Scrape Kafka Metrics:

Edit the prometheus.yml configuration file to add Kafka as a target:

```
scrape_configs:
  - job_name: 'kafka'
 static_configs:
- targets: ['localhost:8080']
```

Start Prometheus:

```
./prometheus --config.file=prometheus.yml
```

Set Up Grafana and Add Prometheus as a Data Source

1. Download and install Grafana if not already installed.
2. Open Grafana at http://localhost:3000 and log in.
3. Add Prometheus as a Data Source.
4. Import or create Kafka dashboards to monitor Kafka metrics (e.g., throughput, consumer lag, etc.).

Set Up Schema Registry

Start Schema Registry: In the Confluent Platform, Schema Registry can be started using the command below. If using a standalone Schema Registry, adjust the paths accordingly

```
bin/schema-registry-start config/schema-registry.properties
```

Ensure Schema Registry is accessible on http://localhost:8081.

Check Schema Registry:

Verify the Schema Registry is running by visiting http://localhost:8081/subjects in a browser or with a curl command:

```
curl -X GET http://localhost:8081/subjects
```

Define and Register a Schema

Create a Schema (e.g., a JSON file named transaction-schema.avsc):

```
{

"type": "record",

   "name": "Transaction",

"namespace": "com.example",

"fields": [

   {"name": "id", "type": "int"},
        {"name": "amount", "type": "double"},

        {"name": "user_id", "type": "int"},

        {"name": "timestamp", "type": "string"}

    ]

}
```

Register the Schema with Schema Registry:

Use the curl command to register the schema:

```
curl -X POST -H "Content-Type: application/vnd.schemaregistry.v1+json" \

--data '{"schema": "{\"type\":\"record\",\"name\":\"Transaction\",\"fields\":[{\"name\":\"id\",\"type\":\"int\"},{\"name\":\"amount\",\"type\":\"double\"},{\"name\":\"user_id\",\"type\":\"int\"},{\"name\":\"timestamp\",\"type\":\"string\"}]}"}' \

http://localhost:8081/subjects/schema-topic-value/versions
```

Verify the Schema Registration:

Check if the schema was successfully registered:

```
curl -X GET http://localhost:8081/subjects/schema-topic-value/versions
```

Produce and Consume Messages with Schema Validation:
Write a Producer to Send Messages Using the Schema
Create a new Java file, SchemaProducer.java, with the following code:

```
import org.apache.kafka.clients.producer.KafkaProducer;
import org.apache.kafka.clients.producer.ProducerRecord;
import org.apache.kafka.clients.producer.RecordMetadata;
import io.confluent.kafka.serializers.KafkaAvroSerializer;
import org.apache.avro.generic.GenericData;
import org.apache.avro.generic.GenericRecord;

import java.util.Properties;

public class SchemaProducer {
    public static void main(String[] args) {
        Properties props = new Properties();
        props.put("bootstrap.servers", "localhost:9092");
        props.put("key.serializer", KafkaAvroSerializer.class.getName());
        props.put("value.serializer", KafkaAvroSerializer.class.getName());
        props.put("schema.registry.url", "http://localhost:8081");
        KafkaProducer<String, GenericRecord> producer = new KafkaProducer<>(props);

        String topic = "schema-topic";
        GenericRecord transaction = new GenericData.Record(schema);

        transaction.put("id", 1);
        transaction.put("amount", 150.0);
        transaction.put("user_id", 1001);
        transaction.put("timestamp", "2024-11-01T10:00:00Z");

        ProducerRecord<String, GenericRecord> record = new ProducerRecord<>(topic, transaction);

        producer.send(record, (RecordMetadata metadata, Exception exception) -> {
            if (exception == null) {
                System.out.println("Message sent successfully to " + metadata.topic());
            } else {
                exception.printStackTrace();
            }
        });

        producer.close();
    }
}
```

Run the Producer: This will produce messages using the schema registered in Schema Registry.

Verify Messages Using Kafka Consumer

Create a Kafka consumer to read from the topic:

```
bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic schema-topic --from-beginning
```

Monitor Messages in Grafana: Check your Kafka Grafana dashboard for the latest metrics, including throughput and consumer lag.

EXPERIMENT NO: 9

NAME OF THE EXPERIMENT: Kafka streams and kafka connectors.

Kafka Streams application that reads messages from a Kafka topic, transforms the data (in this case, converting the message to uppercase), and writes the transformed data to another Kafka topic.

Prerequisites:

- Kafka cluster is running.
- Kafka topics input-topic and output-topic are created.
- Dependencies for Kafka Streams (e.g., org.apache.kafka:kafka-streams in your pom.xml for Maven).

Kafka Streams Example:

```
import org.apache.kafka.common.serialization.Serdes;
import org.apache.kafka.streams.KafkaStreams;
import org.apache.kafka.streams.StreamsConfig;
import org.apache.kafka.streams.kstream.KStream;
import org.apache.kafka.streams.kstream.Consumed;
import org.apache.kafka.streams.kstream.Produced;
import java.util.Properties;
public class KafkaStreamsExample {

    public static void main(String[] args) {

        // Set up Kafka Streams configuration
        Properties props = new Properties();
        props.put(StreamsConfig.APPLICATION_ID_CONFIG, "uppercase-stream-app");
        props.put(StreamsConfig.BOOTSTRAP_SERVERS_CONFIG, "localhost:9092");
        props.put(StreamsConfig.DEFAULT_KEY_SERDE_CLASS_CONFIG, Serdes.String().getClass().getName());
        props.put(StreamsConfig.DEFAULT_VALUE_SERDE_CLASS_CONFIG,
Serdes.String().getClass().getName());
        // Define the stream processing logic
        KStream<String, String> inputStream = new KafkaStreams(props)
                .stream("input-topic", Consumed.with(Serdes.String(), Serdes.String()))
                .mapValues(value -> value.toUpperCase());  // Transform value to uppercase
        // Write the processed stream to the output topic
        inputStream.to("output-topic", Produced.with(Serdes.String(), Serdes.String()));
        // Start the Kafka Streams application
        KafkaStreams streams = new KafkaStreams(inputStream, props);
        streams.start();

        // Shutdown hook to close Kafka Streams
        Runtime.getRuntime().addShutdownHook(new Thread(streams::close));
    }
}
```

How to Run:

1. Start your Kafka cluster.
2. Create Kafka topics input-topic and output-topic.
3. Run the Kafka Streams application.
4. Send messages to input-topic (e.g., using Kafka Console Producer).
5. Check output-topic for uppercase transformed messages.

Kafka Connect JDBC Source Connector:

Kafka Connect uses connectors to integrate with external systems. In this example, we'll set up a JDBC Source Connector to read data from a relational database (e.g., MySQL) and stream it into a Kafka topic.

Prerequisites:

- A running Kafka cluster.
- A relational database (e.g., MySQL) with a sample table.
- Kafka Connect with the JDBC source connector configured.

Steps:

1. Set up the JDBC source connector in Kafka Connect (via REST API or properties files).
2. Configure the JDBC Source Connector to pull data from the database.

JDBC Source Connector Configuration (via REST API):

First, let's assume you have a database test_db and a table users with the following structure:

```
CREATE TABLE users (
id INT PRIMARY KEY,
name VARCHAR(100),
email VARCHAR(100)
);
```

Now, configure the JDBC Source Connector to pull data from this users table.

```
{
  "name": "jdbc-source-connector",
  "config": {
    "connector.class": "io.confluent.connect.jdbc.JdbcSourceConnector",
    "tasks.max": "1",
    "connection.url": "jdbc:mysql://localhost:3306/test_db",
    "connection.user": "root",
    "connection.password": "password",
    "topic.prefix": "users_",
    "table.whitelist": "users",
    "mode": "bulk",
    "poll.interval.ms": "1000"
  }
}
```

How to Run:

1. Start Kafka Connect in distributed mode.
2. Post the connector configuration via REST to the Kafka Connect cluster.

```
curl -X POST -H "Content-Type: application/json" --data @jdbc-source-
config.json http://localhost:8083/connectors
```

3 Kafka Connect will start reading the users table and stream the data to Kafka topics (users_users in this case).

4. Use a Kafka consumer (e.g., kafka-console-consumer) to consume the data from the generated topic.

```
   kafka-console-consumer --bootstrap-server localhost:9092 --topic users_users      --
from-beginning
```

Kafka Connect JDBC Sink Connector:

In this example, we'll configure a JDBC Sink Connector to write data from Kafka topics to a relational database (e.g., MySQL).

JDBC Sink Connector Configuration (via REST API):

Here's how you configure a JDBC Sink Connector to write data from Kafka to the users table in a MySQL database.

```
{
    "name": "jdbc-sink-connector",
    "config": {
        "connector.class": "io.confluent.connect.jdbc.JdbcSinkConnector",
        "tasks.max": "1",
        "connection.url": "jdbc:mysql://localhost:3306/test_db",
        "connection.user": "root",
        "connection.password": "password",
        "topics": "users-topic",
        "insert.mode": "insert",
        "auto.create": "true",
        "auto.evolve": "true",
        "pk.mode": "none",
        "batch.size": "1000"
    }
}
```

1. Start Kafka Connect in distributed mode.
2. Post the connector configuration via REST to the Kafka Connect cluster.

```
curl -X POST -H "Content-Type: application/json" --data @jdbc-sink-
config.json http://localhost:8083/connectors
```

3. Kafka Connect will start consuming records from the users-topic Kafka topic and insert them into the users table in the MySQL database.

Comparison of Kafka Streams and Kafka Connect:

Feature	Kafka Streams	Kafka Connect
Purpose	Real-time stream processing in the Kafka ecosystem.	Integration of external systems (data sources and sinks) with Kafka.
Usage	Java client library for building stream processing applications.	Framework for connecting Kafka with external systems using connectors.
Data Flow	Processes data *within* Kafka topics (consuming and producing).	Moves data *in and out* of Kafka topics to/from external systems.
Operations	Complex stream processing (filtering, aggregating, joining).	Simple data transfer between Kafka and external systems.
State	Supports both stateless and stateful processing.	Stateless, focuses on data movement between systems.
Deployment	Embedded in applications as a library.	Can run in standalone or distributed mode.
Examples of Use Cases	Real-time analytics, ETL jobs, event-driven processing.	Data ingestion, data export, CDC, file system integration.

EXPERIMENT NO: 10

NAME OF THE EXPERIMENT: Integration of kafka with storm.

Step-by-Step Example of Kafka + Storm Integration in IntelliJ IDEA

Step 1: Set Up IntelliJ IDEA Project

1. Create a New Project in IntelliJ IDEA:
   - Open IntelliJ IDEA.
   - Click on File → New → Project.
   - Select Java as the project type.
   - Click Next and follow the steps to create the project.
2. Set Up the Project SDK:
   - Make sure the JDK is correctly configured for your project (for example, JDK 8 or JDK 11).
   - Click Next and then Finish to create the project.

Step 2: Download Kafka and Storm JAR Files

1. Download Apache Kafka JAR:
   - Go to Kafka Maven Repository and download the appropriate Kafka JAR (e.g., kafka-clients-3.1.0.jar).
2. Download Apache Storm JAR:
   - Go to Storm Maven Repository and download the Storm Core JAR (e.g., storm-core-2.3.0.jar).
3. Download Kafka Spout JAR:
   - Go to Storm Kafka Spout Repository and download the Storm Kafka Spout JAR (e.g., storm-kafka-client-2.3.0.jar).
4. Download Other Dependencies:
   - You will also need other dependencies like slf4j (Simple Logging Facade for Java) and log4j, which Kafka and Storm depend on.

Step 3: Add JAR Files to IntelliJ IDEA Project

1. Create a lib Directory:
   - Inside your project, create a folder named lib (e.g., ProjectName/lib).
   - Move all the JAR files you downloaded into this lib directory.
2. Add JARs to Classpath:
   - Right-click on your project in IntelliJ and select Open Module Settings (or press F4).
   - Go to Modules → Dependencies tab.
   - Click the + button and choose JARs or directories.
   - Select the lib folder you created and click OK.
3. Check the Module SDK:
   - Make sure the module SDK is set correctly (e.g., Java 11 or Java 8).

Step 4: Write the Kafka + Storm Integration Code

Now, let's create a simple Kafka-Spout and Storm-Bolt example.

Directory Structure:

Css:

```
src
 └── main
     └── java
         └── com
             └── example
                 └── KafkaStormExample.java
 lib
 └── kafka-clients-3.1.0.jar
 └── storm-core-2.3.0.jar
 └── storm-kafka-client-2.3.0.jar
 └── slf4j-api.jar
 └── log4j.jar
```

KafkaStormExample.java

Java:

```
package com.example;

import org.apache.kafka.clients.consumer.ConsumerConfig;
import org.apache.kafka.common.serialization.StringDeserializer;
import org.apache.storm.kafka.spout.KafkaSpout;
import org.apache.storm.kafka.spout.KafkaSpoutConfig;
import org.apache.storm.kafka.bolt.KafkaBolt;
import org.apache.storm.kafka.bolt.selector.DefaultTopicSelector;
import org.apache.storm.topology.TopologyBuilder;
import org.apache.storm.topology.BasicOutputCollector;
import org.apache.storm.topology.OutputFieldsDeclarer;
import org.apache.storm.topology.base.BaseBasicBolt;
import org.apache.storm.tuple.Tuple;
import org.apache.storm.tuple.Values;
import org.apache.storm.LocalCluster;
import org.apache.storm.Config;

import java.util.HashMap;
import java.util.Map;

public class KafkaStormExample {

    // Bolt to process the message from Kafka
    public static class ProcessMessageBolt extends BaseBasicBolt {
        @Override
        public void execute(Tuple tuple, BasicOutputCollector collector) {
            // Extract message from the tuple
            String message = tuple.getStringByField("message");

            // Process the message (convert it to uppercase)
            String processedMessage = message.toUpperCase();

            // Emit the processed message to the next bolt (Kafka Producer)
            collector.emit(new Values(processedMessage));
        }

        @Override
        public void declareOutputFields(OutputFieldsDeclarer declarer) {
            declarer.declare(new org.apache.storm.tuple.Fields("processed-message"));
        }
    }

    public static void main(String[] args) throws Exception {
        // Kafka configuration
        Map<String, Object> kafkaProps = new HashMap<>();
        kafkaProps.put(ConsumerConfig.BOOTSTRAP_SERVERS_CONFIG, "localhost:9092");
        kafkaProps.put(ConsumerConfig.GROUP_ID_CONFIG, "storm-consumer-group");
        kafkaProps.put(ConsumerConfig.KEY_DESERIALIZER_CLASS_CONFIG,
StringDeserializer.class.getName());
        kafkaProps.put(ConsumerConfig.VALUE_DESERIALIZER_CLASS_CONFIG,
StringDeserializer.class.getName());

        // Kafka Spout Configuration
        KafkaSpoutConfig<String, String> kafkaSpoutConfig = KafkaSpoutConfig
                .builder("localhost:9092", "input-topic") // Kafka broker and topic name
                .setProp(kafkaProps)
                .setStartOffset(KafkaSpoutConfig.StartOffset.LATEST) // Start reading from the latest message
                .build();

        // Topology builder
        TopologyBuilder builder = new TopologyBuilder();

        // Set up the Kafka Spout
        builder.setSpout("kafka-spout", new KafkaSpout<>(kafkaSpoutConfig), 1);

        // Set up the ProcessMessageBolt to process messages from Kafka
        builder.setBolt("process-bolt", new ProcessMessageBolt(), 1).shuffleGrouping("kafka-spout");

        // Set up KafkaProducerBolt to send processed messages to Kafka
        builder.setBolt("kafka-producer-bolt", new KafkaBolt<String, String>()
                        .withProducerProperties(kafkaProps)
                        .withTopicSelector(new DefaultTopicSelector("output-topic")), 1)
                .shuffleGrouping("process-bolt");

        // Configure the Storm topology
        Config config = new Config();
        config.setDebug(true);

        // Submit the topology to the Storm cluster (LocalCluster for local testing)
        LocalCluster cluster = new LocalCluster();
        cluster.submitTopology("kafka-storm-topology", config, builder.createTopology());

        // Wait for a while to process messages
        Thread.sleep(10000);

        // Shutdown the cluster after processing
        cluster.shutdown();
    }
}
```

Explanation of the Code:

1. KafkaSpout: Reads messages from Kafka topic input-topic and emits them as tuples.
2. ProcessMessageBolt: Converts each message to uppercase (as a simple transformation) and emits the processed message.
3. KafkaBolt: Sends the processed message to a Kafka topic output-topic.

Step 5: Set Up Kafka Broker and Topics

1. Start Kafka Broker:
   - Download and start a Kafka broker if you don't have one running already.
   - Follow the Kafka quick start guide to set up and start Kafka locally.
2. Create Kafka Topics:
   - Create two Kafka topics using the Kafka command line tools:

     bash
     Copy code

     ```
     kafka-topics --create --topic input-topic --bootstrap-server localhost:9092 --partitions 1 --replication-factor 1
     kafka-topics --create --topic output-topic --bootstrap-server localhost:9092 --partitions 1 --replication-factor 1
     ```

3. Produce Messages to input-topic:
   - Use the Kafka producer tool to send test messages to the input-topic:

```
kafka-console-producer --broker-list localhost:9092 --topic input-topic
```

Type a few test messages, like:

```
hello
world
storm
```

Step 6: Run the Project in IntelliJ IDEA

1. Run the KafkaStormExample.java:
   - In IntelliJ IDEA, click on the green Run button or right-click on the KafkaStormExample.java class and select Run 'KafkaStormExample'.
   - The topology will start and begin reading from the input-topic, processing the messages, and writing the processed messages to output-topic.
2. Consume from the output-topic:
   - Use the Kafka consumer tool to check the processed messages:

```
kafka-console-consumer --bootstrap-server localhost:9092 --topic output-topic --from-beginning
```

   - You should see the messages like:

```
HELLO
WORLD
STORM
```

EXPERIMENT NO 11

NAME OF THE EXPERIMENT: Kafka integration with spark and flume.

Integrating Kafka, Spark, and Flume:

Objective:

To set up an end-to-end data streaming pipeline that uses Flume to ingest data, Kafka as a message broker, and Spark for real-time processing.

Prerequisites

1. Apache Kafka and Apache Zookeeper installed.
2. Apache Flume and Apache Spark installed.
3. Java Development Kit (JDK 8 or above).
4. Scala installed for the Spark job.
5. IntelliJ IDEA or any IDE with Scala support.

Setup

1. Kafka will serve as the message broker.
2. Flume will capture and send data (e.g., log entries) to Kafka.
3. Spark will consume messages from Kafka and process them.

Set Up Kafka :

Start Zookeeper: `bin/zookeeper-server-start.sh config/zookeeper.properties`

Start Kafka: `bin/kafka-server-start.sh config/server.properties`

Create Kafka Topic (e.g., flume-kafka-topic):

```
bin/kafka-topics.sh --create --topic flume-kafka-topic --bootstrap-server localhost:9092 --partitions 1 --replication-factor 1
```

Configure Flume to Send Data to Kafka:

- Create a configuration file for Flume named flume-kafka.conf.
- Copy the following configuration into flume-kafka.conf:

```
 # Define the agent, source, sink, and channel
agent.sources = source1
agent.channels = channel1
agent.sinks = kafka-sink

# Configure the source to monitor a log file
agent.sources.source1.type = exec
agent.sources.source1.command = tail -F /path/to/your/logfile.log
agent.sources.source1.channels = channel1

# Configure the channel as a memory channel
    agent.channels.channel1.type = memory
    agent.channels.channel1.capacity = 1000
```

```
        agent.channels.channel1.transactionCapacity = 100

# Configure Kafka as the sink
agent.sinks.kafka-sink.type = org.apache.flume.sink.kafka.KafkaSink
agent.sinks.kafka-sink.kafka.bootstrap.servers = localhost:9092
agent.sinks.kafka-sink.kafka.topic = flume-kafka-topic
agent.sinks.kafka-sink.channel = channel1
agent.sinks.kafka-sink.kafka.producer.acks = 1
```

Run Flume with the configuration file:

```
bin/flume-ng agent --conf conf --conf-file flume-kafka.conf --name agent -Dflume.root.logger=INFO,console
```

Create a Spark Application to Consume Data from Kafka:
Add Dependencies for Kafka and Spark Streaming (if using Scala):

- In build.sbt for SBT:

```
libraryDependencies += "org.apache.spark" %% "spark-streaming" % "3.0.1"
libraryDependencies += "org.apache.spark" %% "spark-streaming-kafka-0-10" % "3.0.1"
```

In pom.xml for Maven:

```
<!-- Spark and Kafka dependencies for Maven -->
<dependency>
  <groupId>org.apache.spark</groupId>
  <artifactId>spark-streaming_2.12</artifactId>
  <version>3.0.1</version>
</dependency>
<dependency>
  <groupId>org.apache.spark</groupId>
  <artifactId>spark-streaming-kafka-0-10_2.12</artifactId>
  <version>3.0.1</version>
</dependency>
```

Write the Spark Streaming Application:
In a new Scala file named KafkaSparkConsumer.scala, add the following code

```
import org.apache.spark.SparkConf
import org.apache.spark.streaming.{Seconds, StreamingContext}
import org.apache.spark.streaming.kafka010._
import org.apache.kafka.common.serialization.StringDeserializer

object KafkaSparkConsumer {
 def main(args: Array[String]): Unit = {
  // Spark configuration
  val sparkConf = new SparkConf().setAppName("KafkaSparkConsumer").setMaster("local[*]")
  val ssc = new StreamingContext(sparkConf, Seconds(5))
// Kafka parameters
  val kafkaParams = Map[String, Object](
    "bootstrap.servers" -> "localhost:9092",
    "key.deserializer" -> classOf[StringDeserializer],
    "value.deserializer" -> classOf[StringDeserializer],
    "group.id" -> "spark-consumer-group",
    "auto.offset.reset" -> "latest",
    "enable.auto.commit" -> (false: java.lang.Boolean)
  )

  // Define the topic
  val topics = Array("flume-kafka-topic")

  // Create a direct stream
  val stream = KafkaUtils.createDirectStream[String, String](
    ssc,
    LocationStrategies.PreferConsistent,
    ConsumerStrategies.Subscribe[String, String](topics, kafkaParams)
  )

  // Process each message
  stream.map(record => record.value).foreachRDD { rdd =>
    rdd.foreach { message =>
      println(s"Received message: $message")
    }
  }

  // Start streaming
  ssc.start()
  ssc.awaitTermination()
 }
}
```

Run the Spark Application:
Compile and run your KafkaSparkConsumer application.

Verify the Data Flow and Output
In a new terminal, add data to the file that Flume is monitoring:

```
echo "Log Entry 1" >> /path/to/your/logfile.log
echo "Log Entry 2" >> /path/to/your/logfile.log
```

Expected Output in Spark Console:

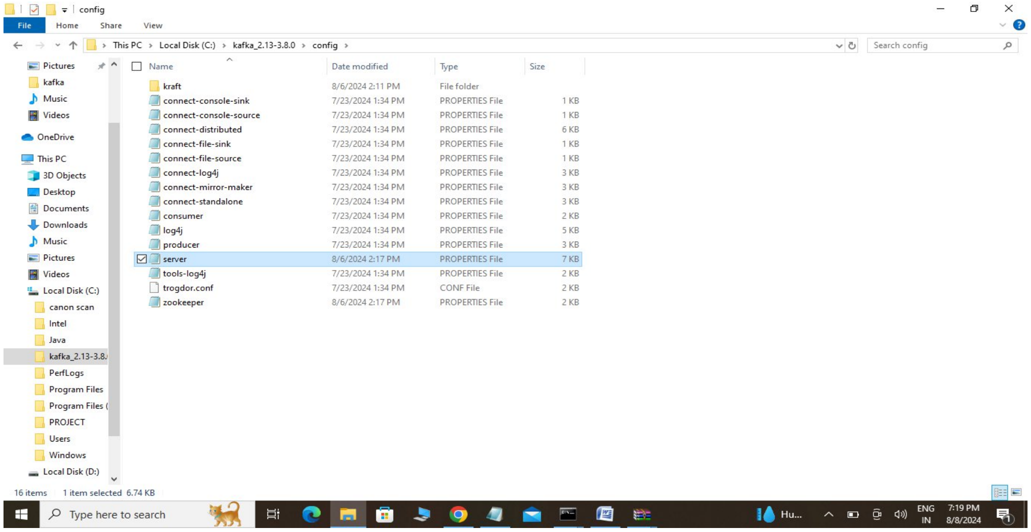
After a few seconds, you should see output in the Spark Streaming application console that looks like this:

```
Received message: Log Entry 1

Received message: Log Entry 2
```

EXPERIMENT NO 12

NAME OF THE EXPERIMENT: Kafka Application as Consumer and Producer.

STEP 1:enter into c drive→ kafka_2.13-3.8.0→C:\kafka_2.13-3.8.0\config→open server



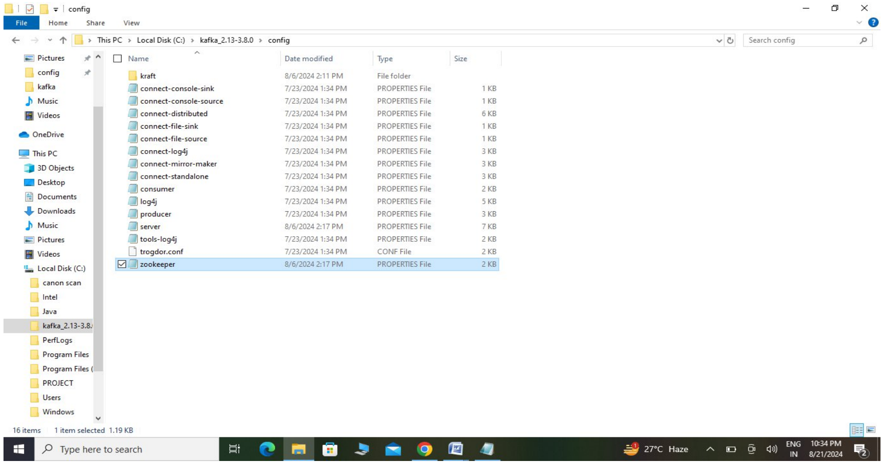
This PC › Local Disk (C:) › kafka_2.13-3.8.0 › config ›

	Name	Date modified	Type	Size
	kraft	8/6/2024 2:11 PM	File folder	
	connect-console-sink	7/23/2024 1:34 PM	PROPERTIES File	1 KB
	connect-console-source	7/23/2024 1:34 PM	PROPERTIES File	1 KB
	connect-distributed	7/23/2024 1:34 PM	PROPERTIES File	6 KB
	connect-file-sink	7/23/2024 1:34 PM	PROPERTIES File	1 KB
	connect-file-source	7/23/2024 1:34 PM	PROPERTIES File	1 KB
	connect-log4j	7/23/2024 1:34 PM	PROPERTIES File	3 KB
	connect-mirror-maker	7/23/2024 1:34 PM	PROPERTIES File	3 KB
	connect-standalone	7/23/2024 1:34 PM	PROPERTIES File	3 KB
	consumer	7/23/2024 1:34 PM	PROPERTIES File	2 KB
	log4j	7/23/2024 1:34 PM	PROPERTIES File	5 KB
	producer	7/23/2024 1:34 PM	PROPERTIES File	3 KB
☑	server	8/6/2024 2:17 PM	PROPERTIES File	7 KB
	tools-log4j	7/23/2024 1:34 PM	PROPERTIES File	2 KB
	trogdor.conf	7/23/2024 1:34 PM	CONF File	2 KB
	zookeeper	8/6/2024 2:17 PM	PROPERTIES File	2 KB

16 items 1 item selected 6.74 KB

log.dirs=C:\kafka_2.13-3.8.0\kafka-logs (replace temp with C:\kafka_2.13-3.8.0) and SAVE the file

STEP 2: Enter into c drive→ kafka_2.13-3.8.0→C:\kafka_2.13-3.8.0\config→open zookeeper



This PC › Local Disk (C:) › kafka_2.13-3.8.0 › config

	Name	Date modified	Type	Size
	kraft	8/6/2024 2:11 PM	File folder	
	connect-console-sink	7/23/2024 1:34 PM	PROPERTIES File	1 KB
	connect-console-source	7/23/2024 1:34 PM	PROPERTIES File	1 KB
	connect-distributed	7/23/2024 1:34 PM	PROPERTIES File	6 KB
	connect-file-sink	7/23/2024 1:34 PM	PROPERTIES File	1 KB
	connect-file-source	7/23/2024 1:34 PM	PROPERTIES File	1 KB
	connect-log4j	7/23/2024 1:34 PM	PROPERTIES File	3 KB
	connect-mirror-maker	7/23/2024 1:34 PM	PROPERTIES File	3 KB
	connect-standalone	7/23/2024 1:34 PM	PROPERTIES File	3 KB
	consumer	7/23/2024 1:34 PM	PROPERTIES File	2 KB
	log4j	7/23/2024 1:34 PM	PROPERTIES File	5 KB
	producer	7/23/2024 1:34 PM	PROPERTIES File	3 KB
	server	8/6/2024 2:17 PM	PROPERTIES File	7 KB
	tools-log4j	7/23/2024 1:34 PM	PROPERTIES File	2 KB
	trogdor.conf	7/23/2024 1:34 PM	CONF File	2 KB
☑	zookeeper	8/6/2024 2:17 PM	PROPERTIES File	2 KB

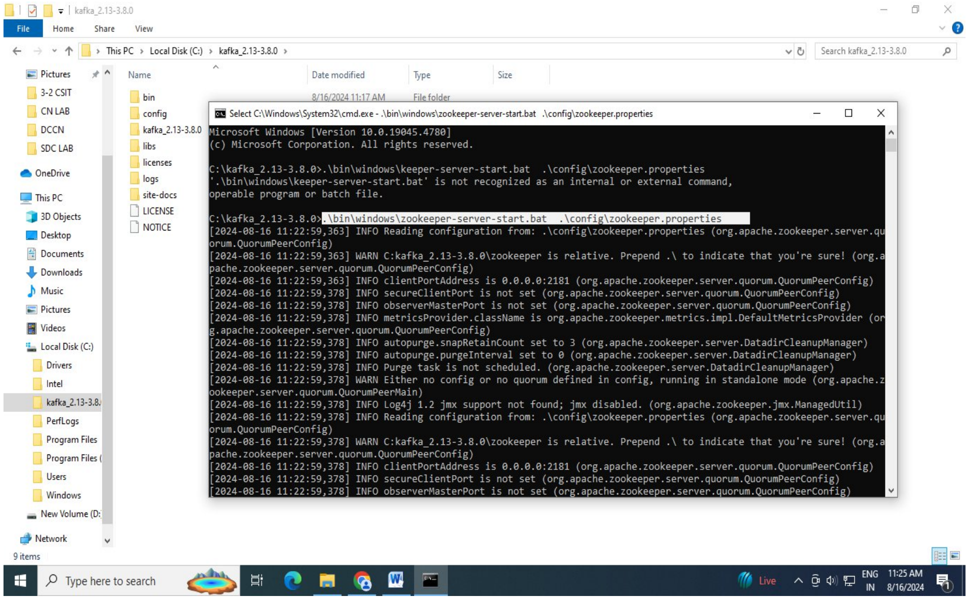
16 items 1 item selected 1.19 KB

dataDir=C:\kafka_2.13-3.8.0\zookeeper (replace temp with C:\kafka_2.13-3.8.0) and SAVE the file

STEP 3: Open cmd and enter into the path C:\kafka_2.13-3.8.0

PATH: C:\kafka_2.13-3.8.0

TYPE:.\bin\windows\zookeeper-server-start.bat .\config\zookeeper.properties



Select C:\Windows\System32\cmd.exe - .\bin\windows\zookeeper-server-start.bat .\config\zookeeper.properties

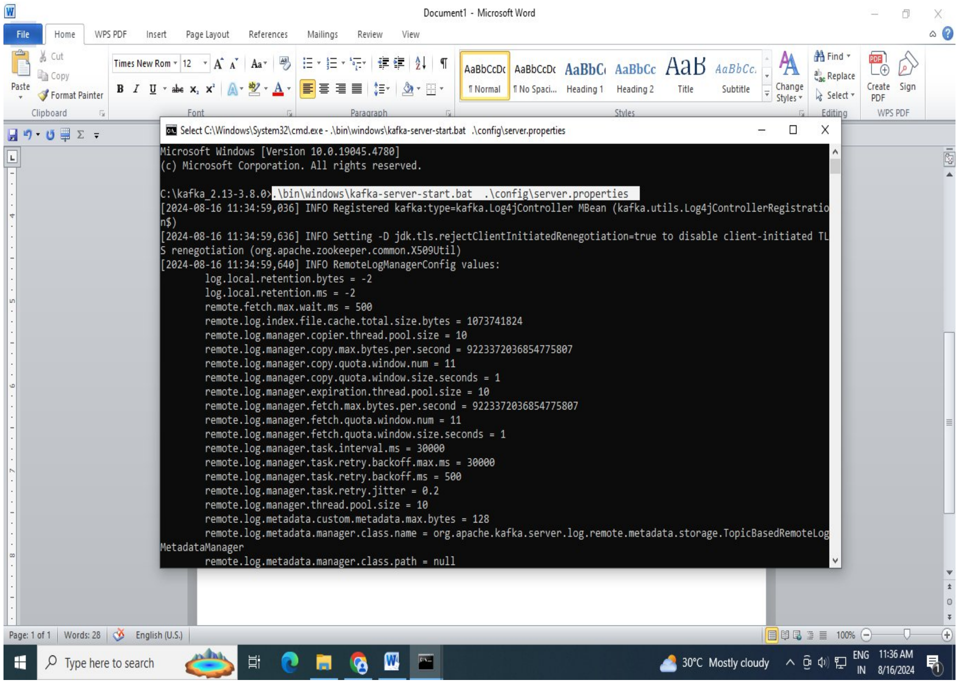
```
Microsoft Windows [Version 10.0.19045.4780]
(c) Microsoft Corporation. All rights reserved.

C:\kafka_2.13-3.8.0>.\bin\windows\keeper-server-start.bat  .\config\zookeeper.properties
'.\bin\windows\keeper-server-start.bat' is not recognized as an internal or external command,
operable program or batch file.

C:\kafka_2.13-3.8.0>.\bin\windows\zookeeper-server-start.bat  .\config\zookeeper.properties
[2024-08-16 11:22:59,363] INFO Reading configuration from: .\config\zookeeper.properties (org.apache.zookeeper.server.qu
orum.QuorumPeerConfig)
[2024-08-16 11:22:59,363] WARN C:kafka_2.13-3.8.0\zookeeper is relative. Prepend .\ to indicate that you're sure! (org.a
pache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-08-16 11:22:59,363] INFO clientPortAddress is 0.0.0.0:2181 (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-08-16 11:22:59,378] INFO secureClientPort is not set (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-08-16 11:22:59,378] INFO observerMasterPort is not set (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-08-16 11:22:59,378] INFO metricsProvider.className is org.apache.zookeeper.metrics.impl.DefaultMetricsProvider (or
g.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-08-16 11:22:59,378] INFO autopurge.snapRetainCount set to 3 (org.apache.zookeeper.server.DatadirCleanupManager)
[2024-08-16 11:22:59,378] INFO autopurge.purgeInterval set to 0 (org.apache.zookeeper.server.DatadirCleanupManager)
[2024-08-16 11:22:59,378] INFO Purge task is not scheduled. (org.apache.zookeeper.server.DatadirCleanupManager)
[2024-08-16 11:22:59,378] WARN Either no config or no quorum defined in config, running in standalone mode (org.apache.z
ookeeper.server.quorum.QuorumPeerMain)
[2024-08-16 11:22:59,378] INFO Log4j 1.2 jmx support not found; jmx disabled. (org.apache.zookeeper.jmx.ManagedUtil)
[2024-08-16 11:22:59,378] INFO Reading configuration from: .\config\zookeeper.properties (org.apache.zookeeper.server.qu
orum.QuorumPeerConfig)
[2024-08-16 11:22:59,378] WARN C:kafka_2.13-3.8.0\zookeeper is relative. Prepend .\ to indicate that you're sure! (org.a
pache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-08-16 11:22:59,378] INFO clientPortAddress is 0.0.0.0:2181 (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-08-16 11:22:59,378] INFO secureClientPort is not set (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-08-16 11:22:59,378] INFO observerMasterPort is not set (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
```

NOTE: DON'T CLOSE CMD

STEP 4: Open A NEW CMD and enter into the path C:\kafka_2.13-3.8.0

PATH: C:\kafka_2.13-3.8.0

TYPE: .\bin\windows\kafka-server-start.bat .\config\server.properties



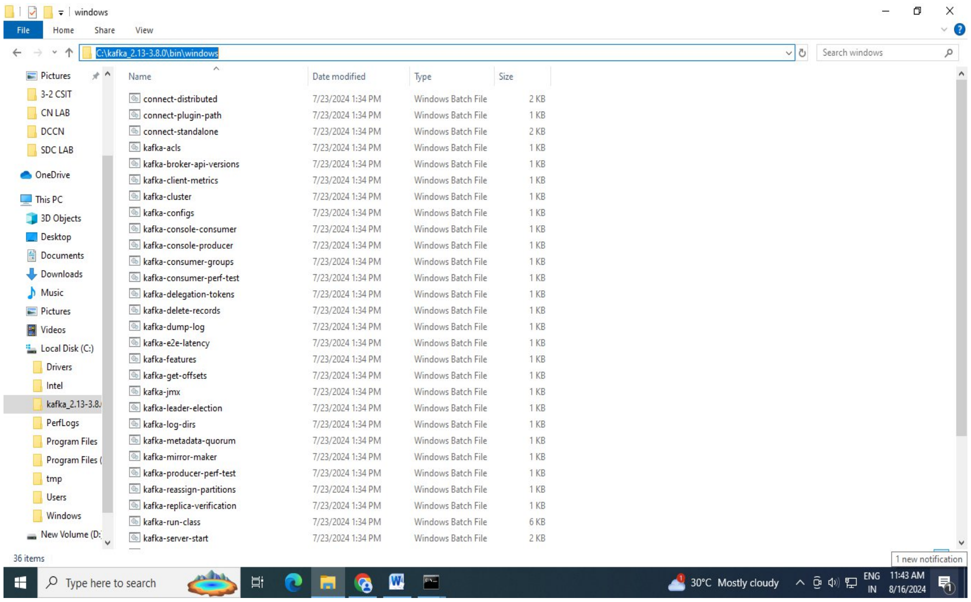
Select C:\Windows\System32\cmd.exe - .\bin\windows\kafka-server-start.bat .\config\server.properties

```
Microsoft Windows [Version 10.0.19045.4780]
(c) Microsoft Corporation. All rights reserved.

C:\kafka_2.13-3.8.0>.\bin\windows\kafka-server-start.bat  .\config\server.properties
[2024-08-16 11:34:59,036] INFO Registered kafka:type=kafka.Log4jController MBean (kafka.utils.Log4jControllerRegistratio
n$)
[2024-08-16 11:34:59,636] INFO Setting -D jdk.tls.rejectClientInitiatedRenegotiation=true to disable client-initiated TL
S renegotiation (org.apache.zookeeper.common.X509Util)
[2024-08-16 11:34:59,640] INFO RemoteLogManagerConfig values:
        log.local.retention.bytes = -2
        log.local.retention.ms = -2
        remote.fetch.max.wait.ms = 500
        remote.log.index.file.cache.total.size.bytes = 1073741824
        remote.log.manager.copier.thread.pool.size = 10
        remote.log.manager.copy.max.bytes.per.second = 9223372036854775807
        remote.log.manager.copy.quota.window.num = 11
        remote.log.manager.copy.quota.window.size.seconds = 1
        remote.log.manager.expiration.thread.pool.size = 10
        remote.log.manager.fetch.max.bytes.per.second = 9223372036854775807
        remote.log.manager.fetch.quota.window.num = 11
        remote.log.manager.fetch.quota.window.size.seconds = 1
        remote.log.manager.task.interval.ms = 30000
        remote.log.manager.task.retry.backoff.max.ms = 30000
        remote.log.manager.task.retry.backoff.ms = 500
        remote.log.manager.task.retry.jitter = 0.2
        remote.log.manager.thread.pool.size = 10
        remote.log.metadata.custom.metadata.max.bytes = 128
        remote.log.metadata.manager.class.name = org.apache.kafka.server.log.remote.metadata.storage.TopicBasedRemoteLog
MetadataManager
        remote.log.metadata.manager.class.path = null
```

STEP 5: Open A NEW CMD and enter into the path C:\kafka_2.13-3.8.0

PATH: C:\kafka_2.13-3.8.0

This step is to create a topic by using the below command

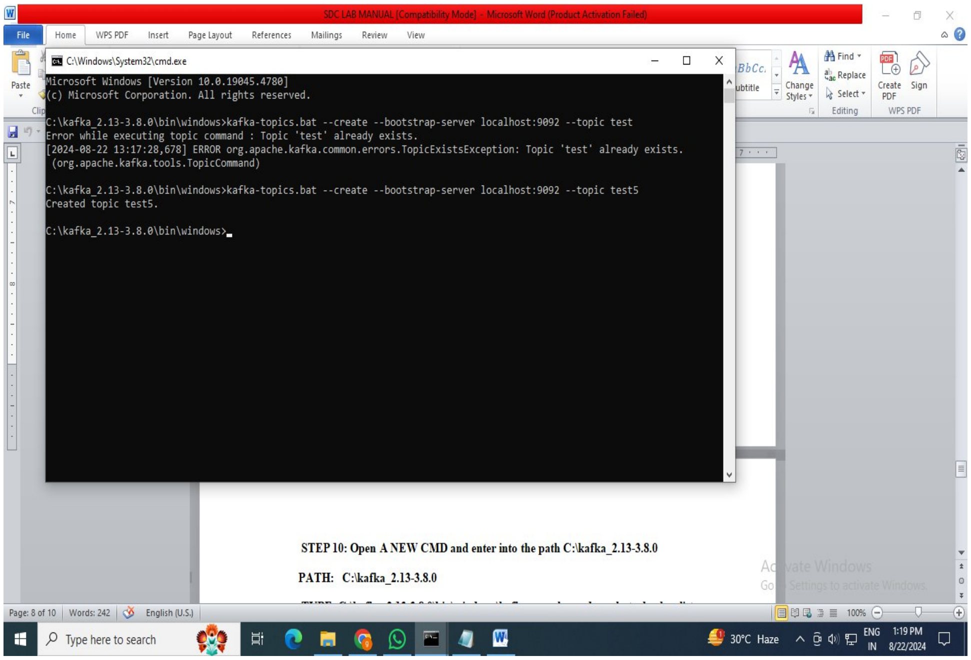


C:\kafka_2.13-3.8.0\bin\windows

Name	Date modified	Type	Size
connect-distributed	7/23/2024 1:34 PM	Windows Batch File	2 KB
connect-plugin-path	7/23/2024 1:34 PM	Windows Batch File	1 KB
connect-standalone	7/23/2024 1:34 PM	Windows Batch File	2 KB
kafka-acls	7/23/2024 1:34 PM	Windows Batch File	1 KB
kafka-broker-api-versions	7/23/2024 1:34 PM	Windows Batch File	1 KB
kafka-client-metrics	7/23/2024 1:34 PM	Windows Batch File	1 KB
kafka-cluster	7/23/2024 1:34 PM	Windows Batch File	1 KB
kafka-configs	7/23/2024 1:34 PM	Windows Batch File	1 KB
kafka-console-consumer	7/23/2024 1:34 PM	Windows Batch File	1 KB
kafka-console-producer	7/23/2024 1:34 PM	Windows Batch File	1 KB
kafka-consumer-groups	7/23/2024 1:34 PM	Windows Batch File	1 KB
kafka-consumer-perf-test	7/23/2024 1:34 PM	Windows Batch File	1 KB
kafka-delegation-tokens	7/23/2024 1:34 PM	Windows Batch File	1 KB
kafka-delete-records	7/23/2024 1:34 PM	Windows Batch File	1 KB
kafka-dump-log	7/23/2024 1:34 PM	Windows Batch File	1 KB
kafka-e2e-latency	7/23/2024 1:34 PM	Windows Batch File	1 KB
kafka-features	7/23/2024 1:34 PM	Windows Batch File	1 KB
kafka-get-offsets	7/23/2024 1:34 PM	Windows Batch File	1 KB
kafka-jmx	7/23/2024 1:34 PM	Windows Batch File	1 KB
kafka-leader-election	7/23/2024 1:34 PM	Windows Batch File	1 KB
kafka-log-dirs	7/23/2024 1:34 PM	Windows Batch File	1 KB
kafka-metadata-quorum	7/23/2024 1:34 PM	Windows Batch File	1 KB
kafka-mirror-maker	7/23/2024 1:34 PM	Windows Batch File	1 KB
kafka-producer-perf-test	7/23/2024 1:34 PM	Windows Batch File	1 KB
kafka-reassign-partitions	7/23/2024 1:34 PM	Windows Batch File	1 KB
kafka-replica-verification	7/23/2024 1:34 PM	Windows Batch File	1 KB
kafka-run-class	7/23/2024 1:34 PM	Windows Batch File	6 KB
kafka-server-start	7/23/2024 1:34 PM	Windows Batch File	2 KB

36 items

TYPE: C:\kafka_2.13-3.8.0\bin\windows\kafka-topics.bat --create --bootstrap-server localhost:9092--topic test



```
Microsoft Windows [Version 10.0.19045.4780]
(c) Microsoft Corporation. All rights reserved.

C:\kafka_2.13-3.8.0\bin\windows>kafka-topics.bat --create --bootstrap-server localhost:9092 --topic test
Error while executing topic command : Topic 'test' already exists.
[2024-08-22 13:17:28,678] ERROR org.apache.kafka.common.errors.TopicExistsException: Topic 'test' already exists.
 (org.apache.kafka.tools.TopicCommand)

C:\kafka_2.13-3.8.0\bin\windows>kafka-topics.bat --create --bootstrap-server localhost:9092 --topic test5
Created topic test5.

C:\kafka_2.13-3.8.0\bin\windows>
```

STEP 10: Open A NEW CMD and enter into the path C:\kafka_2.13-3.8.0

PATH: C:\kafka_2.13-3.8.0

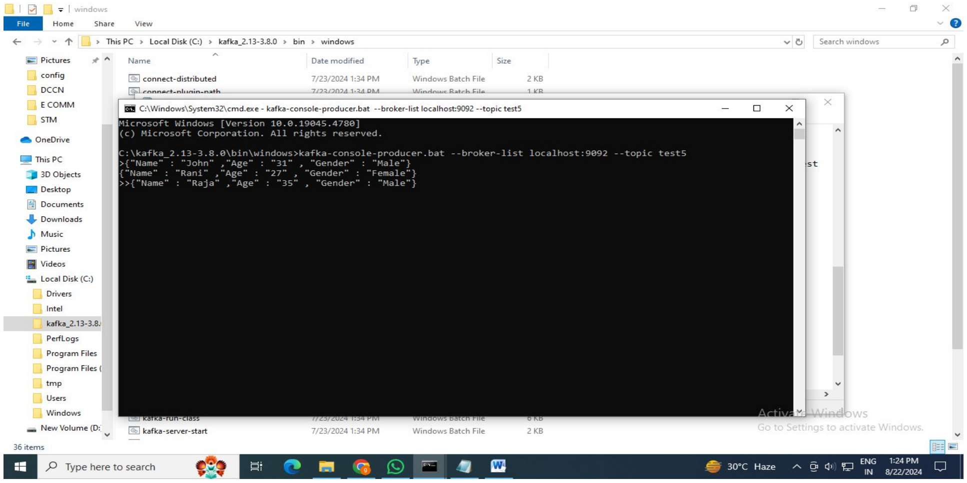
STEP 6: Open A NEW CMD and enter into the path C:\kafka_2.13-3.8.0

PATH: C:\kafka_2.13-3.8.0

TYPE: C:\kafka_2.13-3.8.0\bin\windows\kafka-console-producer.bat --broker-list localhost:9092 --topic test

TYPE: (PRODUCER TAB)

{“Name” : “John” ,”Age” : “31” , “Gender” : “Male”}
{“Name” : “Raju” ,”Age” : “41” , “Gender” : “Male”}
{“Name” : “Rani” ,”Age” : “19” , “Gender” : “Female”}



C:\Windows\System32\cmd.exe - kafka-console-producer.bat --broker-list localhost:9092 --topic test5

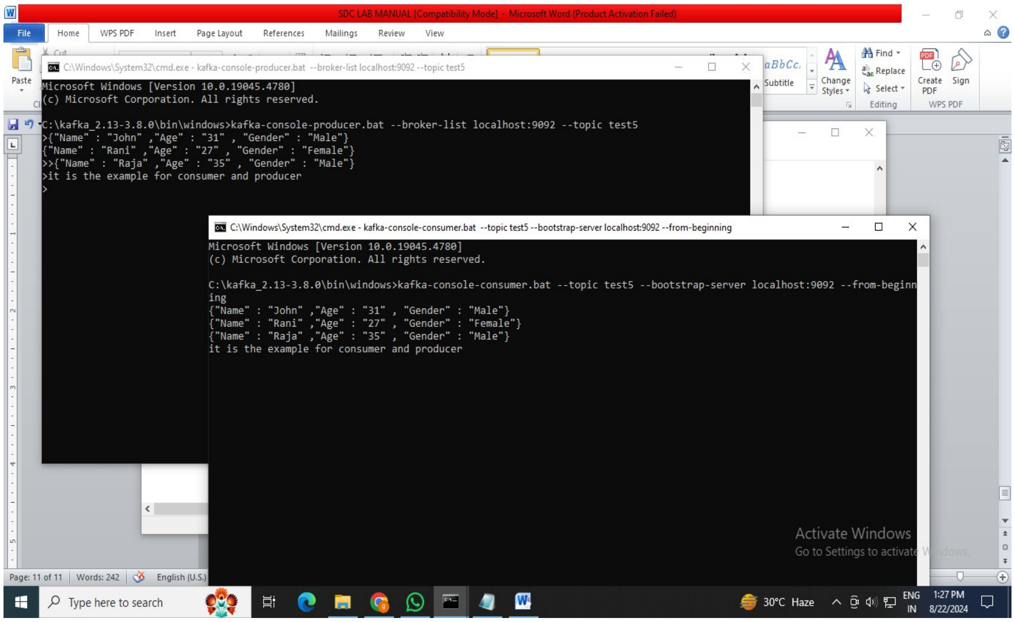
```
Microsoft Windows [Version 10.0.19045.4780]
(c) Microsoft Corporation. All rights reserved.

C:\kafka_2.13-3.8.0\bin\windows>kafka-console-producer.bat --broker-list localhost:9092 --topic test5
>{"Name" : "John" ,"Age" : "31" , "Gender" : "Male"}
{"Name" : "Rani" ,"Age" : "27" , "Gender" : "Female"}
>>{"Name" : "Raja" ,"Age" : "35" , "Gender" : "Male"}
```

STEP 7: Open A NEW CMD and enter into the path C:\kafka_2.13-3.8.0

PATH: C:\kafka_2.13-3.8.0

(CONSUMER TAB)

TYPE: C:\kafka_2.13-3.8.0\bin\windows\kafka-console-consumer.bat --topic test --bootstrap-server localhost:9092 --from-beginning



C:\Windows\System32\cmd.exe - kafka-console-producer.bat --broker-list localhost:9092 --topic test5

```
Microsoft Windows [Version 10.0.19045.4780]
(c) Microsoft Corporation. All rights reserved.

C:\kafka_2.13-3.8.0\bin\windows>kafka-console-producer.bat --broker-list localhost:9092 --topic test5
>{"Name" : "John" ,"Age" : "31" , "Gender" : "Male"}
{"Name" : "Rani" ,"Age" : "27" , "Gender" : "Female"}
>>{"Name" : "Raja" ,"Age" : "35" , "Gender" : "Male"}
>it is the example for consumer and producer
>
```

C:\Windows\System32\cmd.exe - kafka-console-consumer.bat --topic test5 --bootstrap-server localhost:9092 --from-beginning

```
Microsoft Windows [Version 10.0.19045.4780]
(c) Microsoft Corporation. All rights reserved.

C:\kafka_2.13-3.8.0\bin\windows>kafka-console-consumer.bat --topic test5 --bootstrap-server localhost:9092 --from-beginn
ing
{"Name" : "John" ,"Age" : "31" , "Gender" : "Male"}
{"Name" : "Rani" ,"Age" : "27" , "Gender" : "Female"}
{"Name" : "Raja" ,"Age" : "35" , "Gender" : "Male"}
it is the example for consumer and producer
```

EXPERIMENT NO. 13

NAME OF THE EXPERIMENT: Word Count per Record

word count per record in a Kafka stream or a Kafka topic, you would typically perform the following steps:
1. Consume the Kafka records
You first need to consume the records (messages) from your Kafka topic. This is usually done by a consumer application written in a language like Java, Python, or Scala.
2. Parse the message
Kafka messages are typically serialized, so you'll need to deserialize the record into a usable format, like a string or JSON, depending on how the messages are structured.
3. Count the words
Once you have the message in a string format, you can split the string into words and then count them. This would depend on the language you're using, but the general idea is to split the string by spaces (or other delimiters) and count the resulting elements.
4. Process each record
If you're processing a stream, you'll want to calculate the word count for each record as it comes in, rather than processing a bulk batch.

Install Required Libraries
First, make sure you have the confluent-kafka-python library installed. You can install it using pip:

```
pip install confluent-kafka
```

Python Code Example:

```
from confluent_kafka import Consumer, KafkaException, KafkaError

# Configure the Kafka Consumer
conf = {
    'bootstrap.servers': 'localhost:9092',  # Kafka broker
    'group.id': 'word-count-group',         # Consumer group ID
    'auto.offset.reset': 'earliest',        # Start reading from the beginning
}

# Create the consumer instance
consumer = Consumer(conf)

# Kafka topic to consume messages from
topic = 'your_topic_name'  # Replace with your actual topic name
consumer.subscribe([topic])

# Function to calculate the word count of a message
def word_count(text):
    # Split the message by whitespace and count the number of words
    return len(text.split())

# Consume messages and calculate word count
```

```
try:
    while True:
        # Poll for messages (timeout in seconds)
        msg = consumer.poll(timeout=1.0)

        if msg is None:
            continue  # No new message, continue polling

        if msg.error():
            if msg.error().code() == KafkaError._PARTITION_EOF:
                print(f"End of partition reached {msg.partition}")
            else:
                raise KafkaException(msg.error())
        else:
            # Deserialize the message value (assuming it's a UTF-8 string)
            message_value = msg.value().decode('utf-8')

            # Calculate word count
            count = word_count(message_value)
            print(f"Message: {message_value}\nWord Count: {count}\n")

except KeyboardInterrupt:
    print("Consumer interrupted.")
finally:
    # Close the consumer to commit offsets and clean up
    consumer.close()
```

Example Output:

```
Message: This is a test message.
Word Count: 5

Message: Another example with more words.
Word Count: 6
```