Setting Up a 2-Node Hadoop Cluster and Kafka for **Distributed Data Collection and Web Log Analysis**

Japhari Mbaru

Table of Contents

Introduction.

This report outlines the steps to set up a 2-node Hadoop cluster and Kafka for collecting distributed data from multiple nodes, such as web logs. Additionally, it provides a guide to writing a MapReduce code for web log analysis and setting up a dashboard for data visualization. This setup ensures efficient data processing and real-time insights into the collected data.

Weblog Analysis.

Weblog analysis is a crucial process for understanding user behaviors and improving web services. It involves examining the log files generated by web servers to gain insights into user interactions with a website [1]. These log files contain valuable information such as user IP addresses, timestamps, requested URLs, HTTP status codes, and the amount of data transferred. By analyzing this data, businesses can optimize their websites, improve user experience, and enhance their marketing strategies.

Technology Stack.

In this project, the following technology stack is utilized:

1. Hadoop.
    - Version - 3.4.0
    - Components - HDFS (Hadoop Distributed File System), YARN (Yet Another Resource Negotiator), MapReduce
    - Purpose - Distributed storage and processing of large datasets across a cluster of machines.
2. Apache Kafka.
    - Version - Latest stable release
    - Components - Kafka brokers, Kafka topics
    - Purpose - Distributed streaming platform for building real-time data pipelines and streaming applications.

3. Apache Zookeeper.
    - Purpose - Centralized service for maintaining configuration information, naming, providing distributed synchronization, and providing group services.
4. Java Development Kit (JDK).
5. Apache Hive.
    - Purpose – To allow querying and managing large datasets residing in distributed storage.

6. Java Development Kit (JDK).
7.
    - Version - 8 or later.
    - Purpose - Required for running Hadoop and Kafka.
8. Ubuntu Linux.
    - Version - 24.04 or later.
    - Purpose - Operating system for virtual machines.

9. VMware Fusion.
    - Purpose - Virtualization software to run multiple virtual machines on Mac OS.
10. SSH.
    - Purpose - Secure Shell (SSH) protocol for secure network services between nodes.
11. Dashboard Tool.
    - Visualization tool.
    - Purpose - For data visualization and dashboard creation to monitor and analyze data.

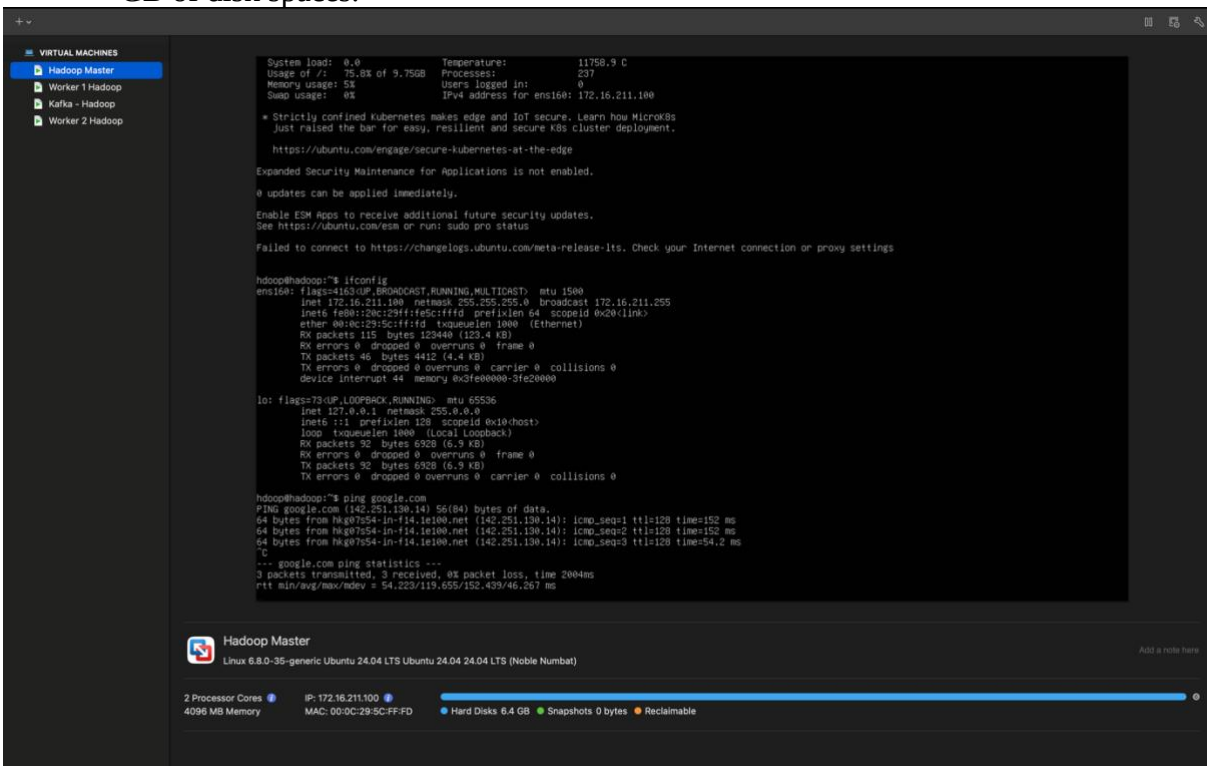
Hadoop Distributed File System.

Distributed file system (DFS) is a transformation of traditional file systems to perform file read, write and execution of petabyte or larger-sized datasets with high-velocity and different structures. In order to process these large amounts of data in an inexpensive and efficient way, Hadoop Distributed File System (HDFS) is used and designed to scale up from a single server to hundreds of servers, with a very high degree of fault tolerance [2]

Prerequisites

Hardware Requirements

- Four Virtual machines with Linux (Ubuntu) installed with at least 4GB RAM and 25 GB of disk spaces.



- Stable network connection between the nodes.

    Configure all the Virtual Machine with static IP addresses for all the servers

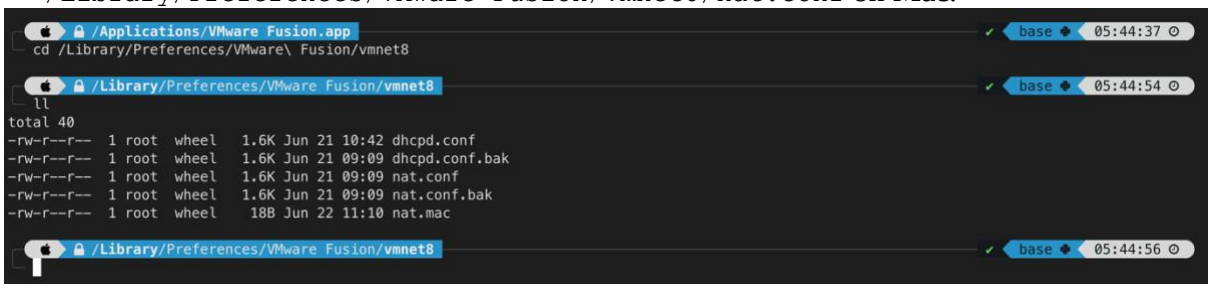
    Mac operating System

    - **Open VMware Fusion** Start VMware Fusion on your Mac.
    - **Navigate to the Virtual Machine Settings**.
    - Select the virtual machine you want to configure.
    - Go to **Virtual Machine** > **Settings** from the menu bar.
    - **Configure the Network Adapter**.

- Click on **Network Adapter** in the settings menu.
  *Ensure the network connection is set to **Share with my Mac (NAT)**. This setting allows your virtual machine to use the Mac's network connection, sharing the same IP address but maintaining unique network ports.*

Advanced NAT Settings.
(To ensure that IP Address will not change on different Networks especially by connecting to different Wi-Fi.

- Open configuration file directly. This file is typically located at `/Library/Preferences/VMware Fusion/vmnet8/nat.conf` on Mac.



- Edit the `dhcpd.conf` on Mac and comment the host vmnet8

```
# See Instructions below if you want to modify it.
#
# We set domain-name-servers to make some DHCP clients happy
# (dhclient as configured in SuSE, TurboLinux, etc.).
# We also supply a domain name to make pump (Red Hat 6.x) happy.
#


###### VMNET DHCP Configuration. Start of "DO NOT MODIFY SECTION" #####
# Modification Instructions: This section of the configuration file contains
# information generated by the configuration program. Do not modify this
# section.
# You are free to modify everything else. Also, this section must start
# on a new line
# This file will get backed up with a different name in the same directory
# if this section is edited and you try to configure DHCP again.

# Written at: 06/21/2024 09:09:55
allow unknown-clients;
default-lease-time 1800;                # default is 30 minutes
max-lease-time 7200;                    # default is 2 hours

subnet 172.16.211.0 netmask 255.255.255.0 {
        range 172.16.211.128 172.16.211.254;
        option broadcast-address 172.16.211.255;
        option domain-name-servers 172.16.211.2;
        option domain-name localdomain;
        default-lease-time 1800;                # default is 30 minutes
        max-lease-time 7200;                    # default is 2 hours
        option netbios-name-servers 172.16.211.2;
        option routers 172.16.211.2;
}
# host vmnet8 {
#       hardware ethernet 00:50:56:C0:00:08;
#       fixed-address 172.16.211.1;
#       option domain-name-servers 0.0.0.0;
#       option domain-name "";
#       option routers 0.0.0.0;
# }
###### VMNET DHCP Configuration. End of "DO NOT MODIFY SECTION" #######
```

- Edit the `nat.conf` on Mac and edit NAT Gateway address and VM net host IP Address which will be used in the Virtual Machines.

```
# VMware NAT configuration file
# Manual editing of this file is not recommended. Using UI is preferred.

[host]

# Use MacOS network virtualization API
useMacosVmnetVirtApi = 1

# NAT gateway address
ip = 172.16.211.2
netmask = 255.255.255.0

# VMnet device if not specified on command line
device = vmnet8

# Allow PORT/EPRT FTP commands (they need incoming TCP stream ...)
activeFTP = 1

# Allows the source to have any OUI.  Turn this on if you change the OUI
# in the MAC address of your virtual machines.
allowAnyOUI = 1

# VMnet host IP address
hostIp = 172.16.211.1
```

- Restart the Network Services and Open VM to configure

```
sudo /Applications/VMware\ Fusion.app/Contents/Library/vmnet-cli --stop
sudo /Applications/VMware\ Fusion.app/Contents/Library/vmnet-cli –start
```

VM Network Configuration

- Open VMware Fusion Start VMware Fusion on your Mac and Start all VM.
- SSH to All the VM and change Net plan and restart all the server

```
hdoop@hadoop:~$  sudo vim /etc/netplan/50-cloud-init.yaml
```

- Change the Configuration to match the Gateway

```
# This file is generated from information provided by the datasource.  Changes
# to it will not persist across an instance reboot.  To disable cloud-init's
# network configuration capabilities, write a file
# /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:
# network: {config: disabled}
network:
    ethernets:
        ens160:
            dhcp4: false
            addresses:  [172.16.211.100/24]
            gateway4: 172.16.211.2
            nameservers:
              addresses: [8.8.8.8,8.8.4.4]
    version: 2
~
~
~
~
~
~
~
~
~
~
~
~
~
```

Figure 1 Hadoop Master Netplan

```
# This file is generated from information provided by the datasource. Changes
# to it will not persist across an instance reboot. To disable cloud-init's
# network configuration capabilities, write a file
# /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:
# network: {config: disabled}
network:
    ethernets:
        ens160:
            dhcp4: false
            addresses: [172.16.211.101/24]
            gateway4: 172.16.211.2
            nameservers:
                addresses: [8.8.8.8, 8.8.4.4]
    version: 2
~
~
~
~
~
~
~
~
~
~
~
~
~
```

Figure 2 Worker 1 Netplan

```
# This file is generated from information provided by the datasource. Changes
# to it will not persist across an instance reboot. To disable cloud-init's
# network configuration capabilities, write a file
# /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:
# network: {config: disabled}
network:
    ethernets:
        ens160:
            dhcp4: false
            addresses: [172.16.211.102/24]
            gateway4: 172.16.211.2
            nameservers:
                addresses: [8.8.8.8, 8.8.4.4]
    version: 2
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
```

Figure 3 Worker 2 Netplan

```
# This file is generated from information provided by the datasource. Changes
# to it will not persist across an instance reboot. To disable cloud-init's
# network configuration capabilities, write a file
# /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:
# network: {config: disabled}
network:
    ethernets:
        ens160:
            dhcp4: false
            addresses: [172.16.211.103/24]
            gateway4: 172.16.211.2
            nameservers:
                addresses: [8.8.8.8, 8.8.4.4]
    version: 2
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
```

Figure 4 Kafka Server Netplan

- Apply the configurations and restart server to use the configured IP Addresses.

```
hdoop@hadoop:~$ sudo netplan apply
```

- Ping google.com and other VMs to test if the configuration is working

```
hdoop@hadoop:~$ ping google.com
PING google.com (142.250.66.142) 56(84) bytes of data.
64 bytes from hkg12s29-in-f14.1e100.net (142.250.66.142): icmp_seq=1 ttl=128 time=41.0 ms
64 bytes from hkg12s29-in-f14.1e100.net (142.250.66.142): icmp_seq=2 ttl=128 time=44.3 ms
64 bytes from hkg12s29-in-f14.1e100.net (142.250.66.142): icmp_seq=3 ttl=128 time=39.3 ms
64 bytes from hkg12s29-in-f14.1e100.net (142.250.66.142): icmp_seq=4 ttl=128 time=44.2 ms
64 bytes from hkg12s29-in-f14.1e100.net (142.250.66.142): icmp_seq=5 ttl=128 time=42.4 ms
64 bytes from hkg12s29-in-f14.1e100.net (142.250.66.142): icmp_seq=6 ttl=128 time=50.4 ms
64 bytes from hkg12s29-in-f14.1e100.net (142.250.66.142): icmp_seq=7 ttl=128 time=47.3 ms
64 bytes from hkg12s29-in-f14.1e100.net (142.250.66.142): icmp_seq=8 ttl=128 time=45.0 ms
64 bytes from hkg12s29-in-f14.1e100.net (142.250.66.142): icmp_seq=9 ttl=128 time=51.7 ms
64 bytes from hkg12s29-in-f14.1e100.net (142.250.66.142): icmp_seq=10 ttl=128 time=44.0 ms
64 bytes from hkg12s29-in-f14.1e100.net (142.250.66.142): icmp_seq=11 ttl=128 time=43.2 ms
64 bytes from hkg12s29-in-f14.1e100.net (142.250.66.142): icmp_seq=12 ttl=128 time=51.6 ms
64 bytes from hkg12s29-in-f14.1e100.net (142.250.66.142): icmp_seq=13 ttl=128 time=53.4 ms
64 bytes from hkg12s29-in-f14.1e100.net (142.250.66.142): icmp_seq=14 ttl=128 time=42.0 ms
64 bytes from hkg12s29-in-f14.1e100.net (142.250.66.142): icmp_seq=15 ttl=128 time=49.7 ms
64 bytes from hkg12s29-in-f14.1e100.net (142.250.66.142): icmp_seq=16 ttl=128 time=41.7 ms
64 bytes from hkg12s29-in-f14.1e100.net (142.250.66.142): icmp_seq=17 ttl=128 time=38.3 ms
64 bytes from hkg12s29-in-f14.1e100.net (142.250.66.142): icmp_seq=18 ttl=128 time=64.6 ms
64 bytes from hkg12s29-in-f14.1e100.net (142.250.66.142): icmp_seq=19 ttl=128 time=72.1 ms
64 bytes from hkg12s29-in-f14.1e100.net (142.250.66.142): icmp_seq=20 ttl=128 time=40.9 ms
64 bytes from hkg12s29-in-f14.1e100.net (142.250.66.142): icmp_seq=21 ttl=128 time=48.3 ms
64 bytes from hkg12s29-in-f14.1e100.net (142.250.66.142): icmp_seq=22 ttl=128 time=46.5 ms
64 bytes from hkg12s29-in-f14.1e100.net (142.250.66.142): icmp_seq=23 ttl=128 time=62.0 ms
64 bytes from hkg12s29-in-f14.1e100.net (142.250.66.142): icmp_seq=24 ttl=128 time=57.1 ms
64 bytes from hkg12s29-in-f14.1e100.net (142.250.66.142): icmp_seq=25 ttl=128 time=40.9 ms
```

Configure Hostname Resolution

Hostname	IP Address
masternode	176.16.211.100
workernode1	172.16.211.101
Workernode2	172.16.211.102

- Edit the Hostname on all Nodes to match their corresponding IP Addresses.

```
hdoop@hadoop:~$ sudo vim /etc/hosts
```

```
127.0.0.1 localhost
#127.0.0.1 hadoop
172.16.211.100 master
172.16.211.101 worker1
172.16.211.102 worker2
172.16.211.103 kafk
# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
~
~
~
~
~
```

- Test to ping hostname all the nodes

```
hdoop@hadoop:~$ ping worker1
PING worker1 (172.16.211.101) 56(84) bytes of data.
64 bytes from worker1 (172.16.211.101): icmp_seq=1 ttl=64 time=1.66 ms
64 bytes from worker1 (172.16.211.101): icmp_seq=2 ttl=64 time=1.42 ms
64 bytes from worker1 (172.16.211.101): icmp_seq=3 ttl=64 time=1.11 ms
64 bytes from worker1 (172.16.211.101): icmp_seq=4 ttl=64 time=0.820 ms
64 bytes from worker1 (172.16.211.101): icmp_seq=5 ttl=64 time=0.850 ms
64 bytes from worker1 (172.16.211.101): icmp_seq=6 ttl=64 time=0.868 ms
64 bytes from worker1 (172.16.211.101): icmp_seq=7 ttl=64 time=0.718 ms
64 bytes from worker1 (172.16.211.101): icmp_seq=8 ttl=64 time=0.720 ms
64 bytes from worker1 (172.16.211.101): icmp_seq=9 ttl=64 time=1.47 ms
64 bytes from worker1 (172.16.211.101): icmp_seq=10 ttl=64 time=0.739 ms
64 bytes from worker1 (172.16.211.101): icmp_seq=11 ttl=64 time=0.390 ms
64 bytes from worker1 (172.16.211.101): icmp_seq=12 ttl=64 time=0.823 ms
64 bytes from worker1 (172.16.211.101): icmp_seq=13 ttl=64 time=0.901 ms
64 bytes from worker1 (172.16.211.101): icmp_seq=14 ttl=64 time=0.434 ms
64 bytes from worker1 (172.16.211.101): icmp_seq=15 ttl=64 time=0.693 ms
^C
--- worker1 ping statistics ---
15 packets transmitted, 15 received, 0% packet loss, time 14211ms
rtt min/avg/max/mdev = 0.390/0.907/1.657/0.350 ms
hdoop@hadoop:~$ ping worker2
PING worker2 (172.16.211.102) 56(84) bytes of data.
64 bytes from worker2 (172.16.211.102): icmp_seq=1 ttl=64 time=2.92 ms
64 bytes from worker2 (172.16.211.102): icmp_seq=2 ttl=64 time=0.505 ms
64 bytes from worker2 (172.16.211.102): icmp_seq=3 ttl=64 time=0.729 ms
64 bytes from worker2 (172.16.211.102): icmp_seq=4 ttl=64 time=1.05 ms
64 bytes from worker2 (172.16.211.102): icmp_seq=5 ttl=64 time=0.507 ms
64 bytes from worker2 (172.16.211.102): icmp_seq=6 ttl=64 time=0.768 ms
64 bytes from worker2 (172.16.211.102): icmp_seq=7 ttl=64 time=0.609 ms
64 bytes from worker2 (172.16.211.102): icmp_seq=8 ttl=64 time=0.956 ms
64 bytes from worker2 (172.16.211.102): icmp_seq=9 ttl=64 time=0.703 ms
64 bytes from worker2 (172.16.211.102): icmp_seq=10 ttl=64 time=0.521 ms
64 bytes from worker2 (172.16.211.102): icmp_seq=11 ttl=64 time=0.816 ms
64 bytes from worker2 (172.16.211.102): icmp_seq=12 ttl=64 time=0.944 ms
64 bytes from worker2 (172.16.211.102): icmp_seq=13 ttl=64 time=0.648 ms
64 bytes from worker2 (172.16.211.102): icmp_seq=14 ttl=64 time=2.97 ms
64 bytes from worker2 (172.16.211.102): icmp_seq=15 ttl=64 time=0.846 ms
^C
--- worker2 ping statistics ---
15 packets transmitted, 15 received, 0% packet loss, time 14329ms
rtt min/avg/max/mdev = 0.505/1.033/2.974/0.767 ms
hdoop@hadoop:~$
```

Technology Stack:

Software Requirements:

- Java Development Kit (JDK) installed on three machines.

```
hdoop@hadoop:~$ sudo apt install default-jdk default-jre
```

```
hdoop@hadoop:~$ java -version
openjdk version "1.8.0_412"
OpenJDK Runtime Environment (build 1.8.0_412-8u412-ga-1~24.04.2-b08)
OpenJDK 64-Bit Server VM (build 25.412-b08, mixed mode)
hdoop@hadoop:~$
```

- SSH setup on both machines for password-less login.
    1. Create a dedicated user for Hadoop on all nodes

```
hadoop@hadoop:/home/hdoop$ sudo adduser hdoop
```

    2. On the Master Node switch user to `hdoop` and generate SSH key

```
hadoop@hadoop:/home/hdoop$ su hdoop
Password:
hdoop@hadoop:~$ ssh-keygen -t rsa
```

    3. Add the generated public key to the master server `authorized keys`.

```
hdoop@hadoop:~$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
hdoop@hadoop:~$
```

    4. Copy Keys to both workers – `worker1` and `worker2`.

```
hdoop@hadoop:~$ ssh-copy-id hadoop@worker1
```

```
hdoop@hadoop:~$ ssh-copy-id hadoop@worker2
```

Architecture Overview

The architecture consists of a 2-node Hadoop cluster and a Kafka setup to collect distributed data. Data is ingested into Kafka from multiple sources and then processed using Hadoop's MapReduce. A dashboard tool is used to visualize the processed data

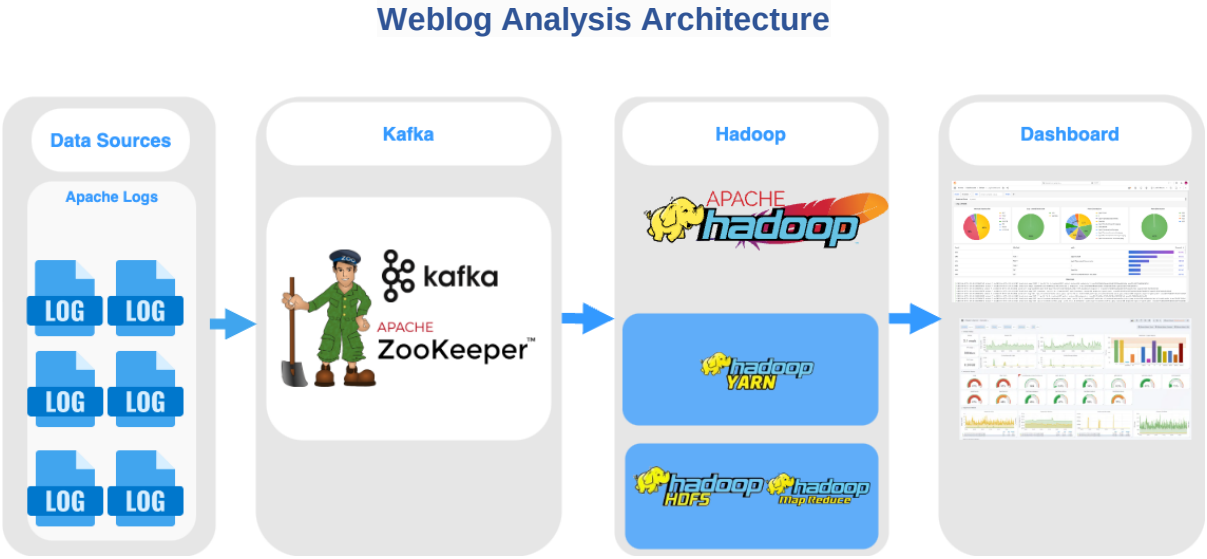


Figure 5 Weblog Analysis Architecture

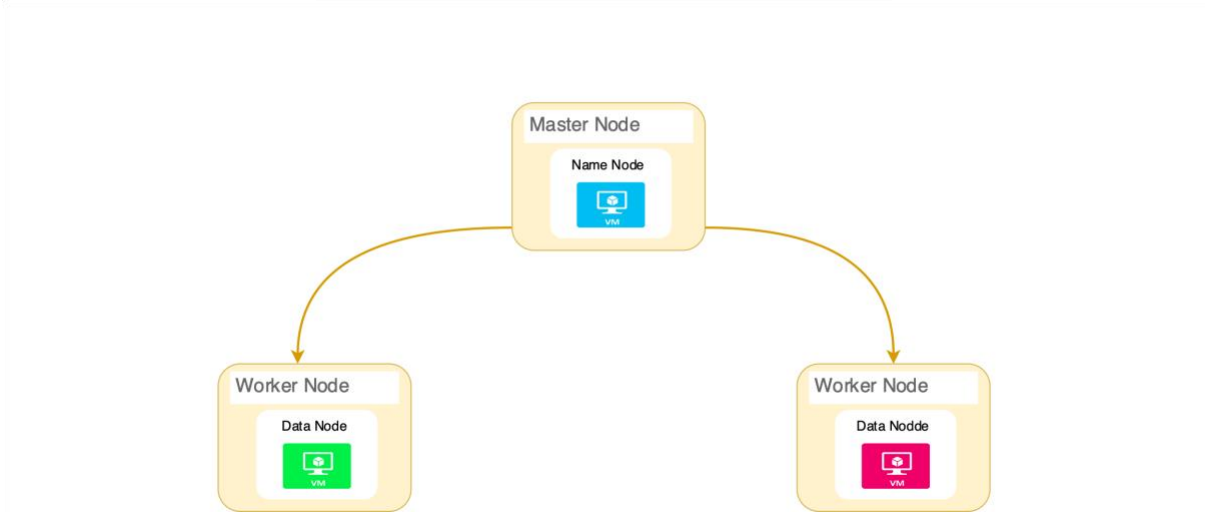


Figure 6 Master Worker Architecture

Setting Up a 2-Node Hadoop Cluster

Installing Hadoop Version 3.4.0 (Master and Workers)

- Download Hadoop from the official website (version 3.4.0).

```
hdoop@hadoop:~$ wget https://www.apache.org/dyn/closer.cgi/hadoop/common/hadoop-3.4.0/hadoop-3.4.0.tar.gz
```

- Extract the Hadoop tar file on both master and worker nodes

```
hdoop@hadoop:~$ tar xzf hadoop-3.4.0.tar.gz
```

- Set Environment Varriables

```
hdoop@hadoop:~$ sudo vim .bashrc
```

```
#Hadoop Related Options
export HADOOP_HOME=/home/hdoop/hadoop-3.4.0
export HADOOP_INSTALL=$HADOOP_HOME
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib/native"

export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-arm64
".bashrc" 129L, 4249B                                                    128,0-1       Bot
```

- Verify Hadoop Installation

```
hdoop@hadoop:~$ hadoop version
Hadoop 3.4.0
Source code repository git@github.com:apache/hadoop.git -r bd8b77f398f626bb7791783192ee7a5dfaeec760
Compiled by root on 2024-03-04T06:35Z
Compiled on platform linux-x86_64
Compiled with protoc 3.21.12
From source with checksum f7fe694a3613358b38812ae9c31114e
This command was run using /home/hdoop/hadoop-3.4.0/share/hadoop/common/hadoop-common-3.4.0.jar
hdoop@hadoop:~$
```

Configuring Hadoop

- Configuring Master Node
- Edit `hadoop-env.sh - Set `JAVA_HOME` variable`

```
hdoop@hadoop:~$ sudo vim $HADOOP_HOME/etc/hadoop/hadoop-env.sh
```

```
###
# Generic settings for HADOOP
###

# Technically, the only required environment variable is JAVA_HOME.
# All others are optional.  However, the defaults are probably not
# preferred.  Many sites configure these options outside of Hadoop,
# such as in /etc/profile.d

# The java implementation to use. By default, this environment
# variable is REQUIRED on ALL platforms except OS X!
 export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-arm64
```

- Configure `core-site.xml` – Set the default filesystem to HDFS and specify the master node.

```
hdoop@hadoop:~$ sudo vim $HADOOP_HOME/etc/hadoop/core-site.xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
<property>
  <name>hadoop.tmp.dir</name>
  <value>/home/hdoop/tmpdata</value>
</property>
<property>
  <name>fs.default.name</name>
  <value>hdfs://master:9000</value>
</property>
</configuration>
~
~
```

- Configure `hdfs-site.xml` – Set the replication factor and specify the namenode and datanode directories.

```
hdoop@hadoop:~$ sudo vim  $HADOOP_HOME/etc/hadoop/hdfs-site.xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
<property>
  <name>dfs.data.dir</name>
  <value>/home/hdoop/dfsdata/namenode</value>
</property>
<property>
  <name>dfs.data.dir</name>
  <value>/home/hdoop/dfsdata/datanode</value>
</property>
<property>
  <name>dfs.replication</name>
  <value>3</value>
</property>
</configuration>
~
~
```

- Edit workers file on the master node and define the data nodes.

```
hdoop@hadoop:~$ sudo vim $HADOOP_HOME/etc/hadoop/workers
```

```
worker1
worker2
~
~
~
~
~
~
~
~
~
```

- Copy all configurations from the master node to both data nodes

```
hdoop@hadoop:~$ scp $HADOOP_HOME/etc/hadoop/* hadoop@worker1:$HADOOP_HOME/etc/hadoop/
```

```
hdoop@hadoop:~$ scp $HADOOP_HOME/etc/hadoop/* hadoop@worker2:$HADOOP_HOME/etc/hadoop/
```

- Configure `mapred-site.xml` – Set the MapReduce framework to YARN (Master Node).

```
hdoop@hadoop:~$ sudo vim $HADOOP_HOME/etc/hadoop/mapred-site.xml
```

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
  <property>
    <name>yarn.app.mapreduce.am.env</name>
    <value>HADOOP_MAPRED_HOME=/home/hdoop/hadoop-3.4.0</value>
  </property>
  <property>
    <name>mapreduce.map.env</name>
    <value>HADOOP_MAPRED_HOME=/home/hdoop/hadoop-3.4.0</value>
  </property>
  <property>
    <name>mapreduce.reduce.env</name>
    <value>HADOOP_MAPRED_HOME=/home/hdoop/hadoop-3.4.0</value>
  </property>
</configuration>
~
~
~
~
~
~
```

- Configure `yarn-site.xml` – Specify the ResourceManager hostname (Master Node).

```
hdoop@hadoop:~$ sudo vim $HADOOP_HOME/etc/hadoop/yarn-site.xml
```

```
<?xml version="1.0"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<configuration>
    <property>
            <name>yarn.acl.enable</name>
            <value>0</value>
    </property>

    <property>
            <name>yarn.resourcemanager.hostname</name>
            <value>master</value>
    </property>

    <property>
            <name>yarn.nodemanager.aux-services</name>
            <value>mapreduce_shuffle</value>
    </property>
</configuration>
~
~
~
~
~
```

Starting the Hadoop Cluster

- Format the HDFS on the master node.
  On the Master format the Hadoop namenode.

```
hdoop@hadoop:~$ hdfs namenode -format
```

- Start DFS Service and Yarn

```
hdoop@hadoop:~$ start-all.sh
```

- Verify status of Hadoop cluster (Master Node)

```
hdoop@hadoop:~$ jps
10977 SecondaryNameNode
11153 ResourceManager
9558 NodeManager
28667 Jps
10750 NameNode
hdoop@hadoop:~$
```

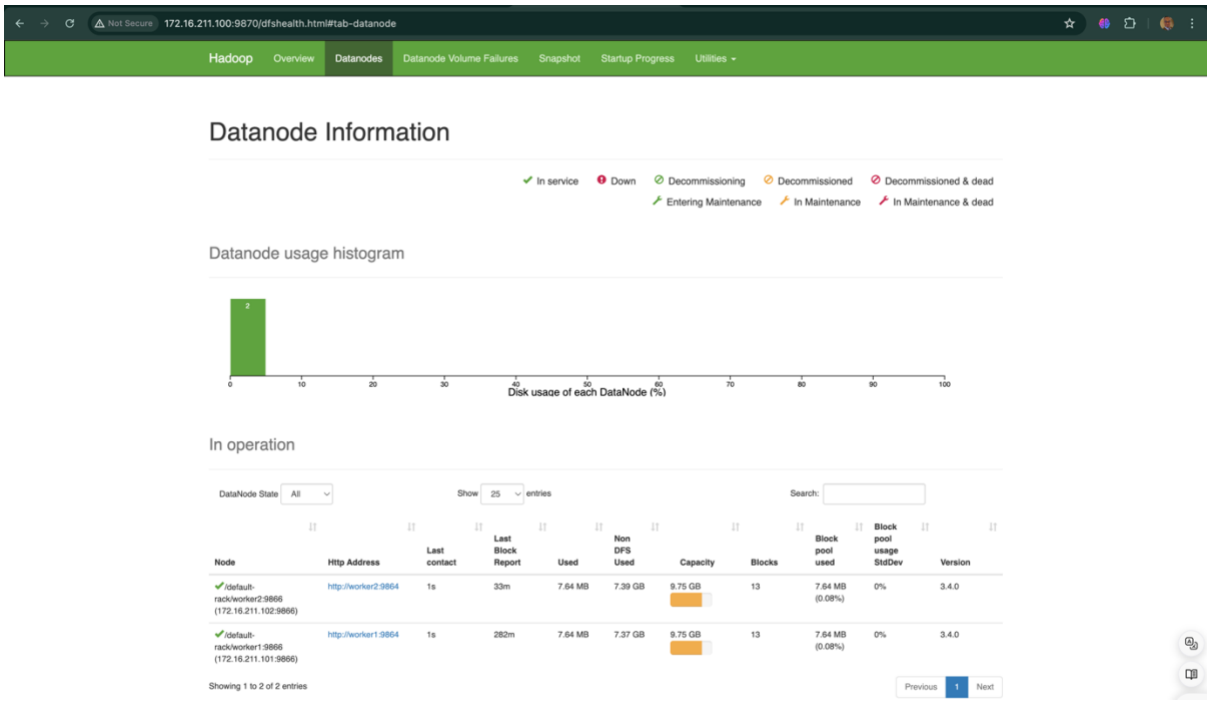
- Verify status of Hadoop (`Worker1`).

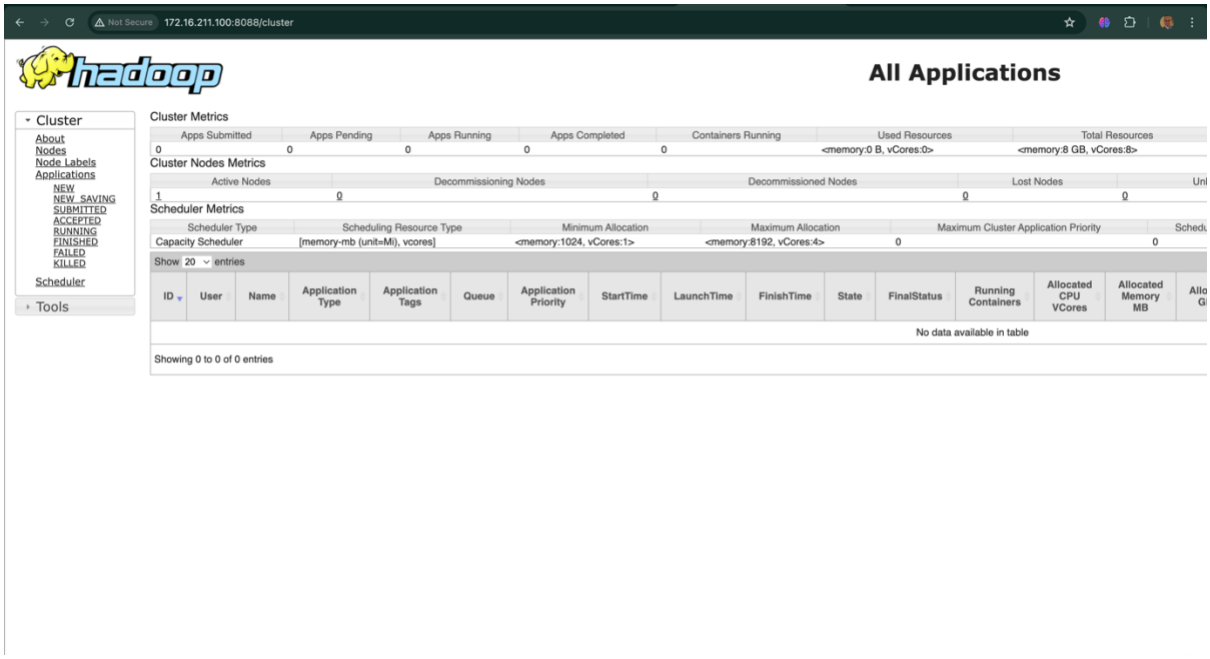
```
hdoop@hadoop:~$ jps
14368 Jps
6193 DataNode
14226 NodeManager
hdoop@hadoop:~$
```

- Verify status of Hadoop (`Worker2`).

```
hdoop@hadoop:~$ jps
6787 DataNode
14536 NodeManager
14665 Jps
hdoop@hadoop:~$
```

- Access Hadoop

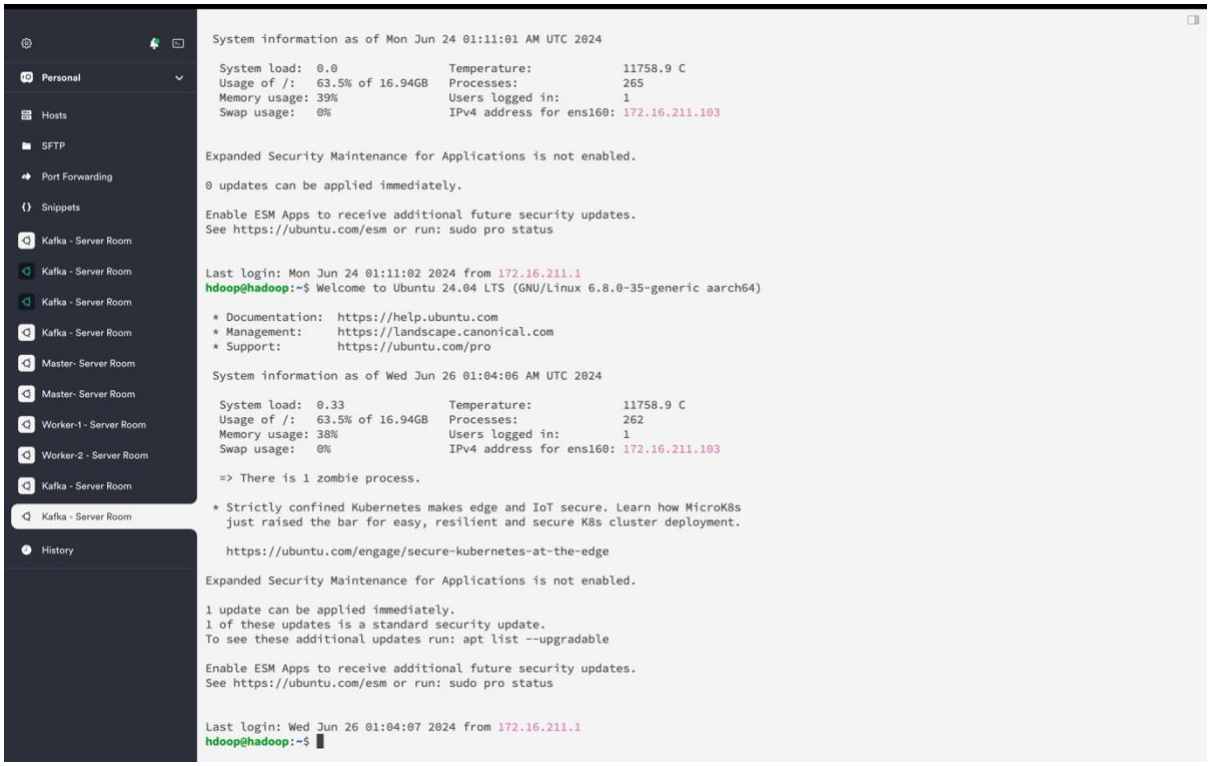




Setting Up Kafka for Distributed Data Collection

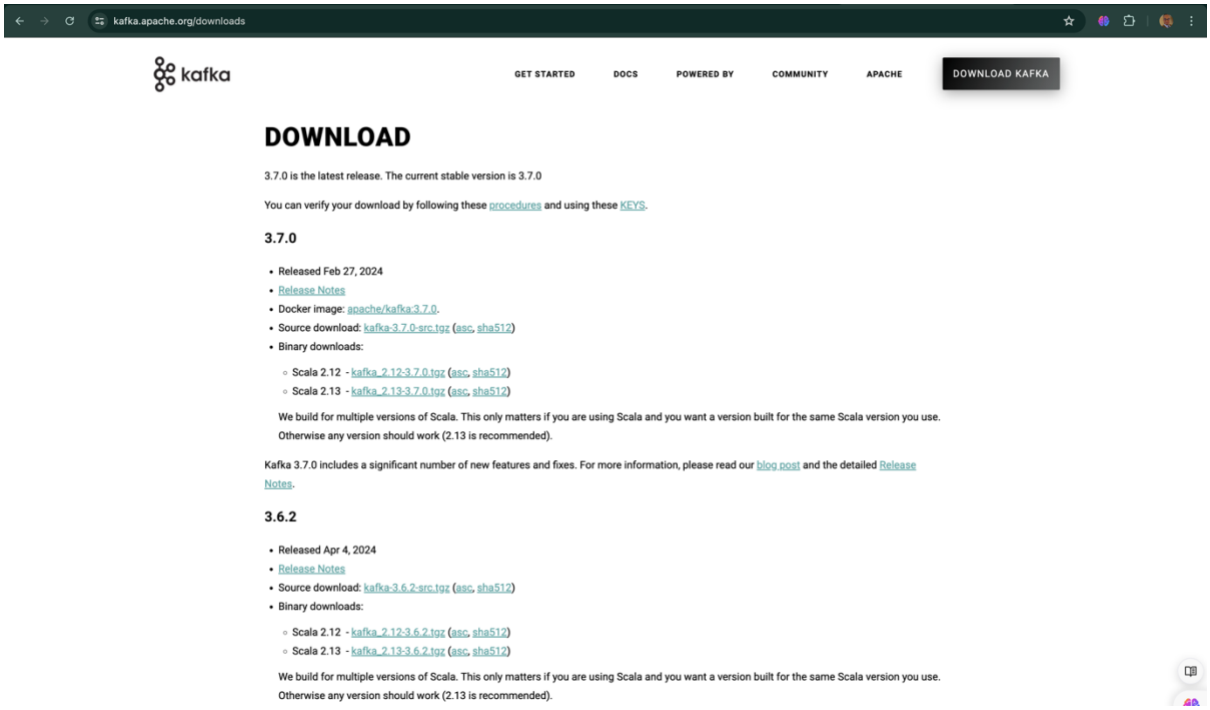
Installing Zookeeper

- SSH to Kafka Server.



Installing Kafka

- Download the latest version of Kafka from the official website.



- Extract Kafka.

```
hdoop@hadoop:~/kafka$ sudo tar -xvf kafka_2.13-3.7.0.tgz
```

Configuring Kafka

- Set Environment Variables - Add Kafka paths to `.bashrc` on both nodes.

```
# Kafka environment variables
export KAFKA_HOME=/home/hdoop/kafka/kafka2
export PATH=$PATH:$KAFKA_HOME/bi
"/home/hdoop/.bashrc" 134L, 4390B                                                    134,33        Bot
```

- Creating a system service file to manage zookeeper "`zookeeper.service`"

```
hdoop@hadoop:~/kafka$ sudo vim /etc/systemd/system/zookeeper.service
```

```
[Unit]
Description=Apache Zookeeper Server
After=network.target

[Service]
Type=simple
ExecStart=/home/hdoop/kafka/kafka2/bin/zookeeper-server-start.sh /home/hdoop/kafka/kafka2/config/zookeeper.properties
ExecStop=/home/hdoop/kafka/kafka2/bin/zookeeper-server-stop.sh

[Install]
WantedBy=multi-user.target
~
~
~
~
~
```

- Reload the system service

```
hdoop@hadoop:~/kafka$ sudo systemctl daemon-reload
```

- Creating a system service file to manage zookeeper "`kafka.service`"

```
hdoop@hadoop:~/kafka$ sudo vim /etc/systemd/system/kafka.service
```

```
[Unit]
Description=Apache Kafka Server
After=zookeeper.service

[Service]
Type=simple
ExecStart=/bin/sh -c '/home/hdoop/kafka/kafka2/bin/kafka-server-start.sh /home/hdoop/kafka/kafka2/config/server.properties > /home/hdoop/kafka
/kafka2/kafka-logs/kafka.log 2>&1'
ExecStop=/home/hdoop/kafka/kafka2/bin/kafka-server-stop.sh
Restart=on-failure
User=hdoop
Group=hdoop

[Install]
WantedBy=multi-user.target
~
~
~
~
```

- Checking Status (Zookeeper and Kafka)

```
[sudo] password for hdoop:
● zookeeper.service - Apache Zookeeper Server
     Loaded: loaded (/etc/systemd/system/zookeeper.service; enabled; preset: enabled)
     Active: active (running) since Fri 2024-06-21 11:31:17 UTC; 4 days ago
   Main PID: 891 (java)
      Tasks: 31 (limit: 4550)
     Memory: 110.8M (peak: 117.3M)
        CPU: 9min 7.617s
     CGroup: /system.slice/zookeeper.service
             └─891 java -Xmx512M -Xms512M -server -XX:+UseG1GC -XX:MaxGCPauseMillis=20 -XX:InitiatingHeapOccupancyPercent=35 -XX:+ExplicitGC>

Jun 21 11:31:17 hadoop zookeeper-server-start.sh[891]: [2024-06-21 11:31:17,943] INFO Snapshot loaded in 9 ms, highest zxid is 0x0, digest is>
Jun 21 11:31:17 hadoop zookeeper-server-start.sh[891]: [2024-06-21 11:31:17,944] INFO Snapshotting: 0x0 to /tmp/zookeeper/version-2/snapshot.>
Jun 21 11:31:17 hadoop zookeeper-server-start.sh[891]: [2024-06-21 11:31:17,945] INFO Snapshot taken in 1 ms (org.apache.zookeeper.server.Zoo>
Jun 21 11:31:17 hadoop zookeeper-server-start.sh[891]: [2024-06-21 11:31:17,952] INFO zookeeper.request_throttler.shutdownTimeout = 10000 ms >
Jun 21 11:31:17 hadoop zookeeper-server-start.sh[891]: [2024-06-21 11:31:17,952] INFO PrepRequestProcessor (sid:0) started, reconfigEnabled=f>
Jun 21 11:31:17 hadoop zookeeper-server-start.sh[891]: [2024-06-21 11:31:17,966] INFO Using checkIntervalMs=60000 maxPerMinute=10000 maxNever>
Jun 21 11:31:17 hadoop zookeeper-server-start.sh[891]: [2024-06-21 11:31:17,967] INFO ZooKeeper audit is disabled. (org.apache.zookeeper.audi>
Jun 21 11:31:18 hadoop zookeeper-server-start.sh[891]: [2024-06-21 11:31:18,188] INFO Creating new log file: log.1 (org.apache.zookeeper.serv>
Jun 21 14:59:51 hadoop zookeeper-server-start.sh[891]: [2024-06-21 14:59:51,829] INFO Unable to read additional data from client, it probably>
Jun 21 15:00:11 hadoop zookeeper-server-start.sh[891]: [2024-06-21 15:00:11,877] INFO Expiring session 0x10000000daf0001, timeout of 18000ms >
lines 1-20/20 (END)
```

```
hdoop@hadoop:~/kafka$ sudo systemctl status kafka.service
● kafka.service - Apache Kafka Server
     Loaded: loaded (/etc/systemd/system/kafka.service; enabled; preset: enabled)
     Active: active (running) since Fri 2024-06-21 15:14:29 UTC; 4 days ago
   Main PID: 8073 (sh)
      Tasks: 75 (limit: 4550)
     Memory: 409.2M (peak: 409.7M)
        CPU: 32min 7.490s
     CGroup: /system.slice/kafka.service
             ├─8073 /bin/sh -c "/home/hdoop/kafka/kafka2/bin/kafka-server-start.sh /home/hdoop/kafka/kafka2/config/server.properties > /home/>
             └─8074 java -Xmx1G -Xms1G -server -XX:+UseG1GC -XX:MaxGCPauseMillis=20 -XX:InitiatingHeapOccupancyPercent=35 -XX:+ExplicitGCInvo>

Jun 21 15:14:29 hadoop systemd[1]: Started kafka.service - Apache Kafka Server.
lines 1-12/12 (END)
```

Kafka Commands

- Creating Kafka Topics

```
hdoop@hadoop:~/kafka$ kafka-topics.sh --create --bootstrap-server localhost:9092 --replication-factor 1 --partitions 1 --topic logs
Created topic logs.
hdoop@hadoop:~/kafka$
```

- List of the Kafka Topics

```
hdoop@hadoop:~/kafka$ kafka-topics.sh --bootstrap-server localhost:9092 --list
__consumer_offsets
access_logs
logs
website_logs
zanbil_logs
hdoop@hadoop:~/kafka$
```

Apache Hive

- Download Apache Hive from apache hive website and `ssh` to Hadoop server and extract hive files.

```
hdoop@hadoop:/$ wget https://dlcdn.apache.org/hive/hive-4.0.0/apache-hive-4.0.0-bin.tar.gz
```

- Move the files in /usr/local folder and configure environment in the `bashrc` script and update the script with the following configurations

```
#Apache Hive Configurations
export HIVE_HOME=/usr/local/hive
export PATH=$PATH:$HIVE_HOME/bin

^G Help        ^O Write Out   ^W Where Is    ^K Cut         ^T Execute     ^C Location    M-U Undo       M-A Set Mark
^X Exit        ^R Read File   ^\ Replace     ^U Paste       ^J Justify     ^/ Go To Line  M-E Redo       M-G Copy
```

```
hdoop@hadoop:/$ source ~/.bashrc
```

- Ensure that dfs and yarn are running and then configure hive to work with Hadoop

Setting Up Hive Configuration Files

- Create or edit the `hive-site.xml` file

```
hdoop@hadoop:~$ cp $HIVE_HOME/conf/hive-default.xml.template $HIVE_HOME/conf/hive-site.xml
hdoop@hadoop:~$
```

- Edit `hive-site.xml`

```
hdoop@hadoop:~$ nano $HIVE_HOME/conf/hive-site.xml
```

```
  GNU nano 7.2                                                  hive-site.xml
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?><!--
   Licensed to the Apache Software Foundation (ASF) under one or more
   contributor license agreements.  See the NOTICE file distributed with
   this work for additional information regarding copyright ownership.
   The ASF licenses this file to You under the Apache License, Version 2.0
   (the "License"); you may not use this file except in compliance with
   the License.  You may obtain a copy of the License at

       http://www.apache.org/licenses/LICENSE-2.0

   Unless required by applicable law or agreed to in writing, software
   distributed under the License is distributed on an "AS IS" BASIS,
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
   See the License for the specific language governing permissions and
   limitations under the License.
--><configuration>
  <!-- WARNING!!! This file is auto generated for documentation purposes ONLY! -->
  <!-- WARNING!!! Any changes you make to this file will be ignored by Hive.   -->
  <!-- WARNING!!! You must make your changes in hive-site.xml instead.         -->
  <!-- Hive Execution Parameters -->
  <property>
    <name>hive.exec.script.wrapper</name>
    <value/>
    <description/>
  </property>
  <property>
    <name>hive.exec.plan</name>
    <value/>
    <description/>
  </property>
  <property>
    <name>hive.exec.stagingdir</name>
    <value>.hive-staging</value>
    <description>Directory name that will be created inside table locations in order to support HDFS encryption. This is replaces ${hi>
  </property>
  <property>
    <name>hive.exec.scratchdir</name>
    <value>/tmp/hive</value>
    <description>HDFS root scratch dir for Hive jobs which gets created with write all (733) permission. For each connecting user, an >
  </property>
                                [ Read 6924 lines ]
^G Help      ^O Write Out  ^W Where Is  ^K Cut       ^T Execute   ^C Location  M-U Undo    M-A Set Mark
^X Exit      ^R Read File  ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line M-E Redo    M-G Copy
```

- Replace all occurrences of `${system:java.io.tmpdir}` to `/tmp/hive`

```
<property>
    <name>hive.exec.local.scratchdir</name>
    <value>/tmp/hive/hdoop</value>
    <description>Local scratch space for Hive jobs</description>
</property>
 <property>
    <name>hive.downloaded.resources.dir</name>
    <value>/tmp/hive_io/${hive.session.id}_resources</value>
    <description>Temporary local directory for added resources in the remote file system.</description>
</property>
<property>
    <name>hive.querylog.location</name>
    <value>/tmp/hive/hdoop</value>
    <description>Location of Hive run time structured log file</description>
</property>
<property>
    <name>hive.server2.logging.operation.log.location</name>
    <value>/tmp/hive/hdoop/operation_logs</value>
    <description>Top level directory where operation logs are stored if logging functionality is
enabled</description>
</property>
```

- Create Hive Ware House Directory

```
hdoop@hadoop:/$ hadoop fs -mkdir -p /user/hive/warehouse
2024-06-27 09:36:16,146 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classe
s where applicable
hdoop@hadoop:/$
```

- Add permission for the directory to be accessed.

```
hdoop@hadoop:/$ hadoop fs -chmod -R 755 /user/hive/warehouse
2024-06-27 09:38:01,084 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classe
s where applicable
hdoop@hadoop:/$
```

- Create a temporary tmp directory

```
hdoop@hadoop:~/hive/conf$ hdfs dfs -mkdir /user/tmp
```

```
hdoop@hadoop:~/hive/conf$ hdfs dfs -chmod g+w /user/tmp
```

- Initialize the Hive Metastore

```
hdoop@hadoop:~/hive$ bin/schematool -initSchema -dbType derby
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/hdoop/hive/lib/log4j-slf4j-impl-2.17.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/hdoop/hadoop-3.4.0/share/hadoop/common/lib/slf4j-log4j12-1.7.32.jar!/org/slf4j/impl/StaticLogge
rBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Metastore connection URL:        jdbc:derby:;databaseName=metastore_db;create=true
Metastore Connection Driver :    org.apache.derby.jdbc.EmbeddedDriver
Metastore connection User:       APP
Starting metastore schema initialization to 3.1.0
Initialization script hive-schema-3.1.0.derby.sql
```

```
Initialization script completed
hdoop@hadoop:/$
```

- To re-initialize the Hive remember to Delete Metastore Database Directory

- Start Hive

```
hdoop@hadoop:~/hive$ bin/beeline -u jdbc:hive2:// -n scott -p tiger
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/hdoop/hive/lib/log4j-slf4j-impl-2.17.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/hdoop/hadoop-3.4.0/share/hadoop/common/lib/slf4j-log4j12-1.7.32.jar!/org/slf4j/impl/StaticLogge
rBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Connecting to jdbc:hive2://
24/06/28 06:25:02 [main]: WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java clas
ses where applicable
Hive Session ID = 1461e519-e018-4e51-b56d-51f3e8cad3a6
24/06/28 06:25:03 [main]: WARN session.SessionState: METASTORE_FILTER_HOOK will be ignored, since hive.security.authorization.manager i
s set to instance of HiveAuthorizerFactory.
24/06/28 06:25:03 [main]: WARN metastore.ObjectStore: datanucleus.autoStartMechanismMode is set to unsupported value null . Setting it
to value: ignored
24/06/28 06:25:03 [main]: WARN util.DriverDataSource: Registered driver with driverClassName=org.apache.derby.jdbc.EmbeddedDriver was n
ot found, trying direct instantiation.
24/06/28 06:25:03 [main]: WARN util.DriverDataSource: Registered driver with driverClassName=org.apache.derby.jdbc.EmbeddedDriver was n
ot found, trying direct instantiation.
24/06/28 06:25:03 [main]: WARN DataNucleus.MetaData: Metadata has jdbc-type of null yet this is not valid. Ignored
24/06/28 06:25:03 [main]: WARN DataNucleus.MetaData: Metadata has jdbc-type of null yet this is not valid. Ignored
24/06/28 06:25:03 [main]: WARN DataNucleus.MetaData: Metadata has jdbc-type of null yet this is not valid. Ignored
24/06/28 06:25:03 [main]: WARN DataNucleus.MetaData: Metadata has jdbc-type of null yet this is not valid. Ignored
24/06/28 06:25:03 [main]: WARN DataNucleus.MetaData: Metadata has jdbc-type of null yet this is not valid. Ignored
24/06/28 06:25:03 [main]: WARN DataNucleus.MetaData: Metadata has jdbc-type of null yet this is not valid. Ignored
24/06/28 06:25:04 [main]: WARN DataNucleus.MetaData: Metadata has jdbc-type of null yet this is not valid. Ignored
24/06/28 06:25:04 [main]: WARN DataNucleus.MetaData: Metadata has jdbc-type of null yet this is not valid. Ignored
24/06/28 06:25:04 [main]: WARN DataNucleus.MetaData: Metadata has jdbc-type of null yet this is not valid. Ignored
24/06/28 06:25:04 [main]: WARN DataNucleus.MetaData: Metadata has jdbc-type of null yet this is not valid. Ignored
24/06/28 06:25:04 [main]: WARN DataNucleus.MetaData: Metadata has jdbc-type of null yet this is not valid. Ignored
24/06/28 06:25:04 [main]: WARN DataNucleus.MetaData: Metadata has jdbc-type of null yet this is not valid. Ignored
Connected to: Apache Hive (version 3.1.3)
Driver: Hive JDBC (version 3.1.3)
Transaction isolation: TRANSACTION_REPEATABLE_READ
Beeline version 3.1.3 by Apache Hive
0: jdbc:hive2://>
```

Collecting Distributed Data Using Kafka

- Producing Data to Kafka

A Kafka producer sends records to a Kafka topic

```
import os
import signal
import sys

# Initialize the Kafka producer
try:
    producer = KafkaProducer(bootstrap_servers='localhost:9092')
    print("Kafka producer initialized successfully.")
except Exception as e:
    print(f"Error initializing Kafka producer: {e}")
    sys.exit(1)

# Path to the log file
log_file_path = '/home/hdoop/connector/kafka-log-connector/access.log'

# Function to read and send logs
def produce_logs():
    try:
        with open(log_file_path, 'r') as file:
            for line in file:
                producer.send('access_logs', value=line.encode('utf-8'))
                print(f"Produced message: {line.strip()}")  # Debug statement
                time.sleep(0.1)  # simulate some delay
        print("Finished producing logs.")  # Debug statement
    except Exception as e:
        print(f"Error producing logs: {e}")

# Handle signal interruption to ensure clean exit
def signal_handler(sig, frame):
    print("Interrupt received, stopping...")
    producer.close()
    sys.exit(0)

signal.signal(signal.SIGINT, signal_handler)

# Produce logs
produce_logs()

# Close the producer
producer.close()
print("Producer closed.")
```

Consuming Data from Kafka

A Kafka consumer reads records from a Kafka topic.

```
import signal
import sys
import logging

# Configure logging
logging.basicConfig(level=logging.INFO, format='%(asctime)s - %(levelname)s - %(message)s')

# Initialize the Kafka consumer
try:
    consumer = KafkaConsumer(
        'access_logs',
        bootstrap_servers='localhost:9092',
        auto_offset_reset='latest',
        enable_auto_commit=True,
        consumer_timeout_ms=10000  # Set consumer timeout
    )
    logging.info("Kafka consumer initialized successfully.")
except Exception as e:
    logging.error(f"Error initializing Kafka consumer: {e}")
    sys.exit(1)

# Initialize HDFS client (ensure the URL matches your HDFS configuration)
try:
    hdfs_client = InsecureClient('http://172.16.211.100:9870', user='hdoop')
    logging.info("HDFS client initialized successfully.")
except Exception as e:
    logging.error(f"Error initializing HDFS client: {e}")
    sys.exit(1)

# Path to HDFS directory
hdfs_path = '/user/hdoop/anomaly/'

# Ensure the HDFS directory exists
try:
    hdfs_client.makedirs(hdfs_path)
    logging.info(f"HDFS directory {hdfs_path} ensured.")
except Exception as e:
    logging.error(f"Error ensuring HDFS directory: {e}")
    sys.exit(1)

# Handle signal interruption to ensure clean exit
def signal_handler(sig, frame):
```

```
# Path to HDFS directory
hdfs_path = '/user/hdoop/anomaly/'

# Ensure the HDFS directory exists
try:
    hdfs_client.makedirs(hdfs_path)
    logging.info(f"HDFS directory {hdfs_path} ensured.")
except Exception as e:
    logging.error(f"Error ensuring HDFS directory: {e}")
    sys.exit(1)

# Handle signal interruption to ensure clean exit
def signal_handler(sig, frame):
    logging.info("Interrupt received, stopping...")
    consumer.close()
    sys.exit(0)

signal.signal(signal.SIGINT, signal_handler)

# Function to consume logs and write to HDFS
def consume_logs():
    log_file_path = os.path.join(hdfs_path, 'access_logs.txt')
    try:
        with hdfs_client.write(log_file_path, encoding='utf-8', overwrite=True) as writer:
            message_count = 0
            for message in consumer:
                log_message = message.value.decode('utf-8')
                logging.info(f"Consumed message: {log_message}")
                writer.write(log_message + '\n')
                message_count += 1
        logging.info(f"Finished writing to HDFS. Total messages: {message_count}")
    except Exception as e:
        logging.error(f"Error consuming logs or writing to HDFS: {e}")

# Consume logs
consume_logs()

# Close the consumer
consumer.close()
logging.info("Consumer closed.")
```

Running Apache Kafka

- Activate Python environment .

```
hdoop@hadoop:~/connector/kafka-log-connector$ source venv/bin/activate
(venv) hdoop@hadoop:~/connector/kafka-log-connector$
```

- Running producer to send logs to kafka

```
(venv) hdoop@hadoop:~/connector/kafka-log-connector$ python3 producer.py
```

- Output of the process

```
Kafka producer initialized successfully.
Produced message: 172.16.0.1 [12/Jun/2024:00:05:44 -0800] "DELETE /api/messages/receive HTTP/1.1" 403 14865
Produced message: 172.16.0.1 [21/Jun/2024:18:31:44 -0800] "GET /api/notifications HTTP/1.1" 404 10906
Produced message: 172.16.0.5 [13/Jun/2024:08:36:44 -0800] "PUT /api/wishlist/remove HTTP/1.1" 403 8666
Produced message: 10.0.4.1 [20/Jun/2024:17:27:44 -0800] "PUT /api/reviews/delete HTTP/1.1" 401 9963
Produced message: 196.249.101.2 [19/Jun/2024:10:54:44 -0800] "POST /api/transactions/status HTTP/1.1" 201 3447
Produced message: 192.168.2.1 [15/Jun/2024:18:03:44 -0800] "GET /api/disbursement/all-repayments/2021 HTTP/1.1" 500 12186
Produced message: 172.16.3.4 [15/Jun/2024:16:33:44 -0800] "PUT /api/address/add HTTP/1.1" 403 9435
Produced message: 172.16.0.4 [12/Jun/2024:20:02:44 -0800] "PUT /api/repayment/2022/repayment HTTP/1.1" 401 4211
Produced message: 10.0.1.2 [14/Jun/2024:01:21:44 -0800] "PUT /api/products/electronics HTTP/1.1" 404 8324
Produced message: 172.16.3.3 [21/Jun/2024:10:14:44 -0800] "POST /api/notifications HTTP/1.1" 401 12179
Produced message: 196.249.101.4 [15/Jun/2024:17:36:44 -0800] "DELETE /api/billing/invoice HTTP/1.1" 500 9122
Produced message: 196.249.101.5 [12/Jun/2024:07:09:44 -0800] "GET /api/account/settings HTTP/1.1" 200 10689
Produced message: 196.249.100.4 [17/Jun/2024:18:56:44 -0800] "PUT /api/profile/view HTTP/1.1" 201 12346
Produced message: 192.168.1.4 [17/Jun/2024:10:56:44 -0800] "DELETE /api/auth/reset-password HTTP/1.1" 403 14507
Produced message: 10.0.2.1 [12/Jun/2024:11:41:44 -0800] "POST /api/feedback/submit HTTP/1.1" 401 4420
Produced message: 10.0.1.5 [17/Jun/2024:05:37:44 -0800] "PUT /api/products/clothing HTTP/1.1" 401 5765
Produced message: 172.16.2.1 [18/Jun/2024:18:32:44 -0800] "POST /api/search HTTP/1.1" 201 7072
Produced message: 172.16.2.5 [20/Jun/2024:18:47:44 -0800] "POST /api/reviews/delete HTTP/1.1" 500 13450
Produced message: 172.16.0.4 [17/Jun/2024:22:47:44 -0800] "POST /api/billing/invoice HTTP/1.1" 403 10467
Produced message: 10.0.2.1 [21/Jun/2024:07:47:44 -0800] "POST /api/user/login HTTP/1.1" 201 10302
Produced message: 172.16.1.4 [22/Jun/2024:14:01:44 -0800] "POST /api/address/delete HTTP/1.1" 401 8805
Produced message: 196.249.102.65 [12/Jun/2024:02:54:44 -0800] "GET /api/help/contact HTTP/1.1" 401 10886
Produced message: 10.0.4.4 [19/Jun/2024:07:59:44 -0800] "GET /api/notifications HTTP/1.1" 403 2147
Produced message: 196.249.102.4 [17/Jun/2024:12:36:44 -0800] "DELETE /api/reviews/edit HTTP/1.1" 404 10569
Produced message: 196.249.100.5 [22/Jun/2024:06:48:44 -0800] "PUT /api/notifications HTTP/1.1" 404 2840
Produced message: 192.168.2.3 [15/Jun/2024:15:56:44 -0800] "GET /api/address/delete HTTP/1.1" 200 7284
Produced message: 196.249.101.3 [21/Jun/2024:06:17:44 -0800] "GET /api/settings/view HTTP/1.1" 200 14374
Produced message: 196.249.103.5 [15/Jun/2024:07:21:44 -0800] "DELETE /api/auth/forgot-password HTTP/1.1" 401 9015
Produced message: 10.0.3.4 [12/Jun/2024:09:48:44 -0800] "POST /api/repayment/2022/repayment HTTP/1.1" 200 14127
```

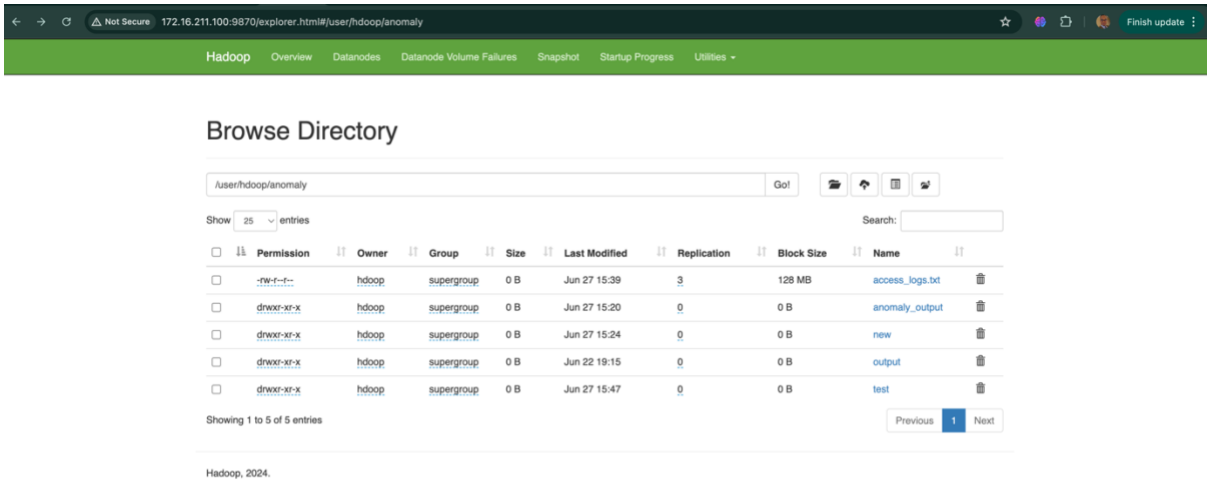
- Running Consumer to send data to HDFS

```
(venv) hdoop@hadoop:~/connector/kafka-log-connector$ python3 consumer.py
```

- Results of the process

```
188.108.242.207 - - [26/Jan/2019:19:43:03 +0330] "GET /image/62170/productModel/150x150 HTTP/1.1" 200 3901 "https://www.zanbil.ir/" "Mozilla/5
.0 (Windows NT 6.1; Win64; x64; rv:64.0) Gecko/20100101 Firefox/64.0" "-"

188.108.242.207 - - [26/Jan/2019:19:43:03 +0330] "GET /image/63227/productModel/150x150 HTTP/1.1" 200 2933 "https://www.zanbil.ir/" "Mozilla/5
.0 (Windows NT 6.1; Win64; x64; rv:64.0) Gecko/20100101 Firefox/64.0" "-"

188.208.61.169 - - [26/Jan/2019:19:43:03 +0330] "GET /image/61627/productModel/150x150 HTTP/1.1" 200 2041 "https://www.zanbil.ir/m/product/315
77/61625/%D9%85%D8%A7%D8%B4%DB%8C%D9%86-%D8%B8%D8%B1%D9%81%D8%B4%D9%88%DB%8C%DB%8C-%D8%A7%DB%8C%D8%B3%D8%AA%D8%A7%D8%AF%D9%87-%D9%85%DB%8C%D8%
AF%DB%8C%D8%A7-%D9%85%D8%AF%D9%84-WQP12-J7617K-W" "Mozilla/5.0 (Android 4.2.1; Mobile; rv:60.0) Gecko/60.0 Firefox/60.0" "-"

188.108.242.207 - - [26/Jan/2019:19:43:03 +0330] "GET /image/55867/productModel/150x150 HTTP/1.1" 200 3414 "https://www.zanbil.ir/" "Mozilla/5
.0 (Windows NT 6.1; Win64; x64; rv:64.0) Gecko/20100101 Firefox/64.0" "-"

188.208.61.169 - - [26/Jan/2019:19:43:03 +0330] "GET /image/31577?name=7617k.1.edited.jpg&wh=max HTTP/1.1" 200 32948 "https://www.zanbil.ir/m/
product/31577/61625/%D9%85%D8%A7%D8%B4%DB%8C%D9%86-%D8%B8%D8%B1%D9%81%D8%B4%D9%88%DB%8C%DB%8C-%D8%A7%DB%8C%D8%B3%D8%AA%D8%A7%D8%AF%D9%87-%D9%8
5%DB%8C%D8%AF%DB%8C%D8%A7-%D9%85%D8%AF%D9%84-WQP12-J7617K-W" "Mozilla/5.0 (Android 4.2.1; Mobile; rv:60.0) Gecko/60.0 Firefox/60.0" "-"

188.108.242.207 - - [26/Jan/2019:19:43:03 +0330] "GET /image/57274/productModel/150x150 HTTP/1.1" 200 5687 "https://www.zanbil.ir/" "Mozilla/5
.0 (Windows NT 6.1; Win64; x64; rv:64.0) Gecko/20100101 Firefox/64.0" "-"

188.108.242.207 - - [26/Jan/2019:19:43:03 +0330] "GET /image/7589/productModel/150x150 HTTP/1.1" 200 2939 "https://www.zanbil.ir/" "Mozilla/5.
0 (Windows NT 6.1; Win64; x64; rv:64.0) Gecko/20100101 Firefox/64.0" "-"

54.36.148.32 - - [26/Jan/2019:19:43:03 +0330] "GET /filter/b88%2Cp28%2C1003%7C350%20%D9%88%D8%A7%D8%AA?o=1003 HTTP/1.1" 302 0 "-" "Mozilla/5.0
 (compatible; AhrefsBot/6.1; +http://ahrefs.com/robot/)" "-"

188.108.242.207 - - [26/Jan/2019:19:43:03 +0330] "GET /image/64844/productModel/150x150 HTTP/1.1" 200 3862 "https://www.zanbil.ir/" "Mozilla/5
.0 (Windows NT 6.1; Win64; x64; rv:64.0) Gecko/20100101 Firefox/64.0" "-"

[2024-06-26 10:37:43,763] ERROR Error processing message, terminating consumer process:  (kafka.tools.ConsoleConsumer$)
org.apache.kafka.common.errors.TimeoutException
Processed a total of 11014 messages
hdoop@hadoop:~/kafka$
```

- HDFS Results



Writing MapReduce Code for Web Log Analysis

- Understanding Web Log Data

Web log data typically includes information such as IP addresses, timestamps, request methods, URLs, response codes, and user agents. Analyzing this data can provide insights into user behavior, traffic patterns, and potential issues.

- Writing the Mapper Class.

Create a Mapper class to parse web log entries. This class will extract relevant fields from each log entry and emit key-value pairs for further processing.

LogMapper.java

The `LogMapper` class processes each line of the web log, extracts the URL, and outputs key-value pairs.

```
import ...

1 usage
public class LogMapper extends Mapper<LongWritable, Text, Text, IntWritable> {
    3 usages
    private static final IntWritable one = new IntWritable( value: 1);
    6 usages
    private Text outputKey = new Text();

    1 usage
    private static final Pattern logPattern = Pattern.compile(
            regex: "^\\S+ \\S+ \\S+ \\[(\\d{2})/(\\w{3})/(\\d{4}):(\\d{2}):(\\d{2}):(\\d{2}) \\+\\d{4}] \"\\S+ (\\S+) \\S+\" (\\d{3}

    public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException {
        String line = value.toString();
        Matcher matcher = logPattern.matcher(line);
        if (matcher.find()) {
            // Extract hour for traffic analysis
            String hour = matcher.group(4);
            outputKey.set("Hour_" + hour);
            context.write(outputKey, one);

            // Extract URL for most visited URLs analysis
            String url = matcher.group(7);
            outputKey.set("URL_" + url);
            context.write(outputKey, one);

            // Extract status code for HTTP status code distribution analysis
            String statusCode = matcher.group(8);
            outputKey.set("Status_" + statusCode);
            context.write(outputKey, one);
        }
    }
}
```

- Writing the Reducer Class

Create a Reducer class to count URL hits. This class will sum up the counts for each URL emitted by the Mapper.

LogReducer.java

The `LogReducer` class counts the occurrences of each URL.

```java
import ...

2 usages
public class LogReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
    public void reduce(Text key, Iterable<IntWritable> values, Context context) throws IOException, InterruptedException {
        int sum = 0;
        for (IntWritable val : values) {
            sum += val.get();
        }
        context.write(key, new IntWritable(sum));
    }
}
```

WebLogAnalysis.java

The `WebLogAnalysis` class configures and runs the Hadoop job.

```java
import ...

public class WebLogAnalysis {
    public static void main(String[] args) throws Exception {
        Configuration conf = new Configuration();
        Job job = Job.getInstance(conf, jobName: "web log analysis");
        job.setJarByClass(WebLogAnalysis.class);
        job.setMapperClass(LogMapper.class);
        job.setCombinerClass(LogReducer.class);
        job.setReducerClass(LogReducer.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);
        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));
        System.exit(job.waitForCompletion( verbose: true) ? 0 : 1);
    }
}
```

Running the MapReduce Job

- Create a JAR file and import to the server to run mapReduce job.

```
hdoop@hadoop:~$ hadoop jar /home/hdoop/Weblog-1.0-SNAPSHOT.jar  org.swahili.WebLogAnalysis /user/hdoop/anomaly/access_logs.txt /user/hdoop/ano
maly
```

```
        File Input Format Counters
                Bytes Read=892217
2024-06-27 12:24:11,769 INFO mapred.LocalJobRunner: Finishing task: attempt_local53144900_0001_m_000000_0
2024-06-27 12:24:11,770 INFO mapred.LocalJobRunner: map task executor complete.
2024-06-27 12:24:11,774 INFO mapred.LocalJobRunner: Starting task: attempt_local53144900_0001_r_000000_0
2024-06-27 12:24:11,774 INFO mapred.LocalJobRunner: Waiting for reduce tasks
2024-06-27 12:24:11,779 INFO output.PathOutputCommitterFactory: No output committer factory defined, defaulting to FileOutputCommitterFactory
2024-06-27 12:24:11,781 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2024-06-27 12:24:11,781 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ign
ore cleanup failures: false
2024-06-27 12:24:11,781 INFO mapred.Task:  Using ResourceCalculatorProcessTree : [ ]
2024-06-27 12:24:11,783 INFO mapred.ReduceTask: Using ShuffleConsumerPlugin: org.apache.hadoop.mapreduce.task.reduce.Shuffle@523e4349
2024-06-27 12:24:11,786 WARN impl.MetricsSystemImpl: JobTracker metrics system already initialized!
2024-06-27 12:24:11,816 INFO reduce.MergeManagerImpl: MergerManager: memoryLimit=637114752, maxSingleShuffleLimit=159278688, mergeThreshold=42
0495744, ioSortFactor=10, memToMemMergeOutputsThreshold=10
2024-06-27 12:24:11,821 INFO reduce.EventFetcher: attempt_local53144900_0001_r_000000_0 Thread started: EventFetcher for fetching Map Completi
on Events
2024-06-27 12:24:11,844 INFO reduce.LocalFetcher: localfetcher#1 about to shuffle output of map attempt_local53144900_0001_m_000000_0 decomp:
29067 len: 29071 to MEMORY
2024-06-27 12:24:11,845 INFO reduce.InMemoryMapOutput: Read 29067 bytes from map-output for attempt_local53144900_0001_m_000000_0
2024-06-27 12:24:11,847 INFO reduce.MergeManagerImpl: closeInMemoryFile -> map-output of size: 29067, inMemoryMapOutputs.size() -> 1, commitMe
mory -> 0, usedMemory ->29067
2024-06-27 12:24:11,847 INFO reduce.EventFetcher: EventFetcher is interrupted.. Returning
2024-06-27 12:24:11,848 INFO mapred.LocalJobRunner: 1 / 1 copied.
2024-06-27 12:24:11,848 INFO reduce.MergeManagerImpl: finalMerge called with 1 in-memory map-outputs and 0 on-disk map-outputs
2024-06-27 12:24:11,859 INFO mapred.Merger: Merging 1 sorted segments
2024-06-27 12:24:11,859 INFO mapred.Merger: Down to the last merge-pass, with 1 segments left of total size: 29052 bytes
2024-06-27 12:24:11,863 INFO reduce.MergeManagerImpl: Merged 1 segments, 29067 bytes to disk to satisfy reduce memory limit
2024-06-27 12:24:11,863 INFO reduce.MergeManagerImpl: Merging 1 files, 29071 bytes from disk
2024-06-27 12:24:11,864 INFO reduce.MergeManagerImpl: Merging 0 segments, 0 bytes from memory into reduce
2024-06-27 12:24:11,864 INFO mapred.Merger: Merging 1 sorted segments
2024-06-27 12:24:11,864 INFO mapred.Merger: Down to the last merge-pass, with 1 segments left of total size: 29052 bytes
2024-06-27 12:24:11,864 INFO mapred.LocalJobRunner: 1 / 1 copied.
2024-06-27 12:24:11,904 INFO Configuration.deprecation: mapred.skip.on is deprecated. Instead, use mapreduce.job.skiprecords
2024-06-27 12:24:12,386 INFO mapreduce.Job: Job job_local53144900_0001 running in uber mode : false
2024-06-27 12:24:12,387 INFO mapreduce.Job:  map 100% reduce 0%
2024-06-27 12:24:12,551 INFO mapred.Task: Task:attempt_local53144900_0001_r_000000_0 is done. And is in the process of committing
2024-06-27 12:24:12,557 INFO mapred.LocalJobRunner: 1 / 1 copied.
2024-06-27 12:24:12,558 INFO mapred.Task: Task attempt_local53144900_0001_r_000000_0 is allowed to commit now
2024-06-27 12:24:12,582 INFO output.FileOutputCommitter: Saved output of task 'attempt_local53144900_0001_r_000000_0' to hdfs://master:9000/us
er/hdoop/anomaly/new
2024-06-27 12:24:12,582 INFO mapred.LocalJobRunner: reduce > reduce
2024-06-27 12:24:12,583 INFO mapred.Task: Task 'attempt_local53144900_0001_r_000000_0' done.
2024-06-27 12:24:12,584 INFO mapred.Task: Final Counters for attempt_local53144900_0001_r_000000_0: Counters: 30
        File System Counters
                FILE: Number of bytes read=63566
                FILE: Number of bytes written=771723
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
```

```
        File System Counters
                FILE: Number of bytes read=63566
                FILE: Number of bytes written=771723
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=892217
                HDFS: Number of bytes written=23457
                HDFS: Number of read operations=10
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=3
                HDFS: Number of bytes read erasure-coded=0
        Map-Reduce Framework
                Combine input records=0
                Combine output records=0
                Reduce input groups=1493
                Reduce shuffle bytes=29071
                Reduce input records=1493
                Reduce output records=1493
                Spilled Records=1493
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=12
                Total committed heap usage (bytes)=298319872
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Output Format Counters
                Bytes Written=23457
2024-06-27 12:24:12,584 INFO mapred.LocalJobRunner: Finishing task: attempt_local53144900_0001_r_000000_0
2024-06-27 12:24:12,584 INFO mapred.LocalJobRunner: reduce task executor complete.
2024-06-27 12:24:13,389 INFO mapreduce.Job:  map 100% reduce 100%
2024-06-27 12:24:13,390 INFO mapreduce.Job: Job job_local53144900_0001 completed successfully
2024-06-27 12:24:13,407 INFO mapreduce.Job: Counters: 36
        File System Counters
                FILE: Number of bytes read=68958
                FILE: Number of bytes written=1514375
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=1784434
                HDFS: Number of bytes written=23457
                HDFS: Number of read operations=15
```

```
        File Output Format Counters
                Bytes Written=23457
2024-06-27 12:24:12,584 INFO mapred.LocalJobRunner: Finishing task: attempt_local53144900_0001_r_000000_0
2024-06-27 12:24:12,584 INFO mapred.LocalJobRunner: reduce task executor complete.
2024-06-27 12:24:13,389 INFO mapreduce.Job:  map 100% reduce 100%
2024-06-27 12:24:13,390 INFO mapreduce.Job: Job job_local53144900_0001 completed successfully
2024-06-27 12:24:13,407 INFO mapreduce.Job: Counters: 36
        File System Counters
                FILE: Number of bytes read=68958
                FILE: Number of bytes written=1514375
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=1784434
                HDFS: Number of bytes written=23457
                HDFS: Number of read operations=15
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=4
                HDFS: Number of bytes read erasure-coded=0
        Map-Reduce Framework
                Map input records=19860
                Map output records=29790
                Map output bytes=624550
                Map output materialized bytes=29071
                Input split bytes=118
                Combine input records=29790
                Combine output records=1493
                Reduce input groups=1493
                Reduce shuffle bytes=29071
                Reduce input records=1493
                Reduce output records=1493
                Spilled Records=2986
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=12
                Total committed heap usage (bytes)=545783808
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=892217
        File Output Format Counters
                Bytes Written=23457
```

- View the Output

```
hdoop@hadoop:~$ hdfs dfs -cat /user/hdoop/anomaly/output/part-r-00000
```

- Results of the Mapreduce

```
196.249.102.65  GET     /api/products/clothing  403     6049
PUT     /api/quiz/integration/nida      500     13423
GET     /api/disbursement/all-repayments/2022   403     5669
DELETE  /api/order/checkout     201     14724
POST    /api/products/clothing  404     13559
DELETE  /api/quiz/integration/nida      403     2646
GET     /api/cart/add   403     13624
DELETE  /api/cart/add   500     2095
DELETE  /api/quiz/integration/nida      403     3938
POST    /api/disbursement/all-repayments/2021   200     11926
GET     /api/quiz/integration/nida      201     6261
PUT     /api/cart/add   403     9721
PUT     /api/user/logout        401     14143
POST    /api/repayment/2022/repayment   500     4884
POST    /api/applicant/area/search-administrative-area-level    201     3903
PUT     /api/user/login 500     12859
GET     /api/products/electronics       401     10439
DELETE  /api/repayment/2022/repayment   404     8248
GET     /api/disbursement/all-repayments/2021   401     5425
GET     /api/disbursement/all-repayments/2022   404     3991
DELETE  /api/user/login 201     2593
GET     /api/order/checkout     500     10891
POST    /api/user/login 401     5926
GET     /api/disbursement/all-repayments/2021   500     6031
GET     /api/order/checkout     403     14328
DELETE  /api/cmg/saveCmgProfileAttachment       200     5774
POST    /api/applicant/area/search-administrative-area-level    403     9277
DELETE  /api/disbursement/all-repayments/2021   404     13284
PUT     /api/cmg/saveCmgProfileAttachment       401     7067
POST    /api/quiz/integration/nida      401     11821
POST    /api/notifications      500     3700
GET     /api/user/logout        403     6098
PUT     /api/repayment/2022/repayment   403     6884
POST    /api/repayment/2022/repayment   500     7897
PUT     /api/order/checkout     500     4151
PUT     /api/cart/add   500     5142
PUT     /api/user/register      201     10641
DELETE  /api/products/books     200     10284
PUT     /api/cart/add   403     3027
POST    /api/quiz/integration/nida      403     3525
POST    /api/cmg/saveCmgProfileAttachment       200     8763
GET     /api/repayment/2021/repayment   201     11260
DELETE  /api/cmg/saveCmgProfileAttachment       500     4523
POST    /api/repayment/2022/repayment   200     6370

GET     /api/settings/update    403     6737
GET     /api/profile/view       201     2492
POST    /api/auth/forgot-password       401     6343
GET     /api/products/electronics       200     7207
POST    /api/reviews/add        401     11607
DELETE  /api/favorites/remove   201     5263
DELETE  /api/cmg/saveCmgProfileAttachment       200     9435
DELETE  /api/reviews/delete     500     2106
GET     /api/billing/invoice    403     9156
DELETE  /api/feedback/submit    500     14199
GET     /api/repayment/2022/repayment   201     6518
PUT     /api/favorites/add      403     4702
GET     /api/address/add        403     9997
PUT     /api/account/settings   404     5324
GET     /api/reviews/add        404     3718
GET     /api/wishlist/remove    200     12307
GET     /api/repayment/2021/repayment   200     10058
PUT     /api/profile/view       404     12666
GET     /api/notifications      500     6385
DELETE  /api/search     401     5465
DELETE  /api/auth/verify-email  201     5589
DELETE  /api/products/electronics       201     8651
POST    /api/disbursement/all-repayments/2022   200     5310
GET     /api/order/checkout     201     9109
POST    /api/reviews/add        404     12859
PUT     /api/wishlist/add       404     10407
POST    /api/user/logout        404     5038
POST    /api/reviews/add        403     10670
POST    /api/favorites/add      201     11231
DELETE  /api/auth/verify-email  401     13542
GET     /api/feedback/view      404     6564
GET     /api/help/faq   201     14035
POST    /api/account/settings   401     14435
PUT     /api/user/login 500     12371
POST    /api/address/add        200     7310
GET     /api/reviews/delete     201     3202
POST    /api/products/clothing  403     6048
POST    /api/cmg/cmgProfileAttachment   201     6670
DELETE  /api/user/register      201     14719
DELETE  /api/favorites/add      403     6753
GET     /api/wishlist/add       500     9932
PUT     /api/feedback/view      403     12931
PUT     /api/transactions/status        404     3512
DELETE  /api/favorites/remove   500     9032
PUT     /api/account/overview   200     3384
POST    /api/disbursement/all-repayments/2021   404     14631
POST    /api/address/update     401     2790
```

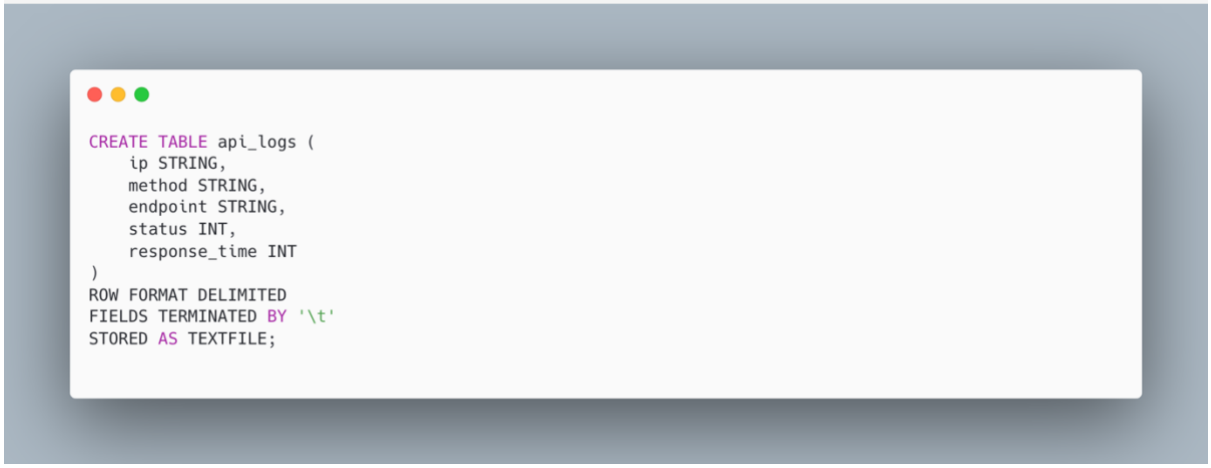
Connecting Data to Hive

- Creating Hive Table
- Starting Hive shell

```
hdoop@hadoop:~/hive$ hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/hdoop/hive/lib/log4j-slf4j-impl-2.17.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/hdoop/hadoop-3.4.0/share/hadoop/common/lib/slf4j-log4j12-1.7.32.jar!/org/slf4j/impl/StaticLogge
rBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Hive Session ID = e84791e0-2919-4484-bb4e-3cdb17872b61

Logging initialized using configuration in jar:file:/home/hdoop/hive/lib/hive-common-3.1.3.jar!/hive-log4j2.properties Async: true
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. s
park, tez) or using Hive 1.X releases.
Hive Session ID = 2055ecf0-822c-41d2-8f6f-73071321715e
hive>
```

- Create the table.

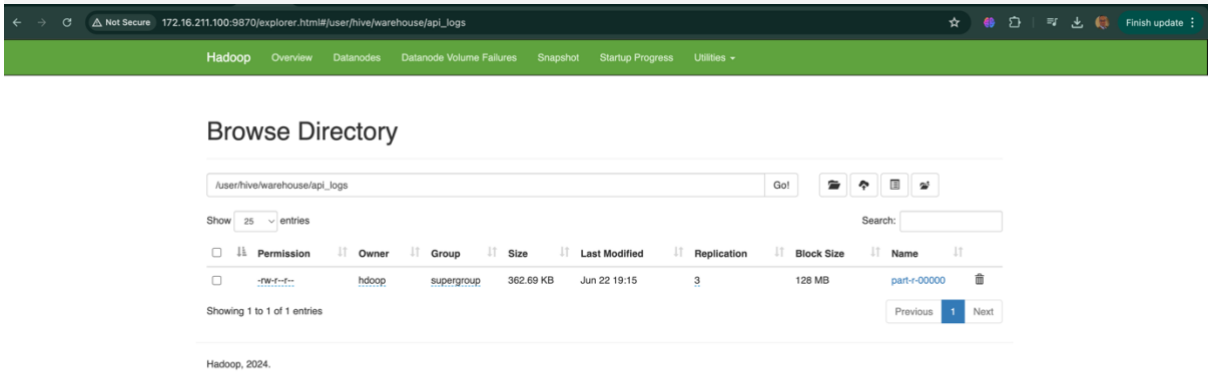
```
hive> CREATE TABLE api_logs (
    >     ip STRING,
    >     method STRING,
    >     endpoint STRING,
    >     status INT,
    >     response_time INT
    > )
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY '\t'
    > STORED AS TEXTFILE;
OK
Time taken: 0.921 seconds
hive>
```



- Load the Data.

```
hive> LOAD DATA INPATH '/user/hdoop/anomaly/output/part-r-00000' INTO TABLE api_logs;
Loading data to table default.api_logs
OK
Time taken: 0.959 seconds
```

```
hive> LOAD DATA INPATH '/user/hdoop/anomaly/output/part-r-00000' INTO TABLE api_logs;
Loading data to table default.api_logs
OK
Time taken: 0.959 seconds
hive>
```



- Count the Number of Requests by Method.

```
SELECT method, COUNT(*) as request_count
FROM api_logs
GROUP BY method;
```

```
hive> SELECT method, COUNT(*) as request_count
    > FROM api_logs
    > GROUP BY method;
Query ID = hdoop_20240628070815_12f2717f-6910-43df-ad6b-4504d53c67f0
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Job running in-process (local Hadoop)
2024-06-28 07:08:18,798 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_local281544927_0001
MapReduce Jobs Launched:
Stage-Stage-1:  HDFS Read: 742806 HDFS Write: 0 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
```

```
/api/billing/transactions       209
/api/cart/add   180
/api/cmg/cmgProfileAttachment   186
/api/cmg/saveCmgProfileAttachment       179
/api/disbursement/all-repayments/2021   215
/api/disbursement/all-repayments/2022   168
/api/favorites/add      174
/api/favorites/remove   186
/api/feedback/submit    188
/api/feedback/view      160
/api/help/contact       177
/api/help/faq   190
/api/help/tickets       203
/api/messages/receive   202
/api/messages/send      177
/api/notifications      212
/api/order/checkout     190
/api/products/books     179
/api/products/clothing  203
/api/products/electronics       220
/api/profile/update     195
/api/profile/view       226
/api/quiz/integration/nida      203
/api/repayment/2021/repayment   181
/api/repayment/2022/repayment   194
/api/reviews/add        164
/api/reviews/delete     187
/api/reviews/edit       187
/api/search     210
/api/settings/update    182
/api/settings/view      196
/api/transactions/history       163
/api/transactions/status        205
/api/user/login 209
/api/user/logout        195
/api/user/register      198
/api/wishlist/add       202
/api/wishlist/remove    207
DELETE  16
GET     24
POST    27
PUT     17
Time taken: 3.784 seconds, Fetched: 56 row(s)
hive>
```

Setting Up a Dashboard for Data Visualization

- Installing and Configuring a Dashboard Tool

Apache Superset

Is an open-source data visualization tool designed to make data exploration and visualization for data analyst and scientist

- Install Apache Superset by using Docker

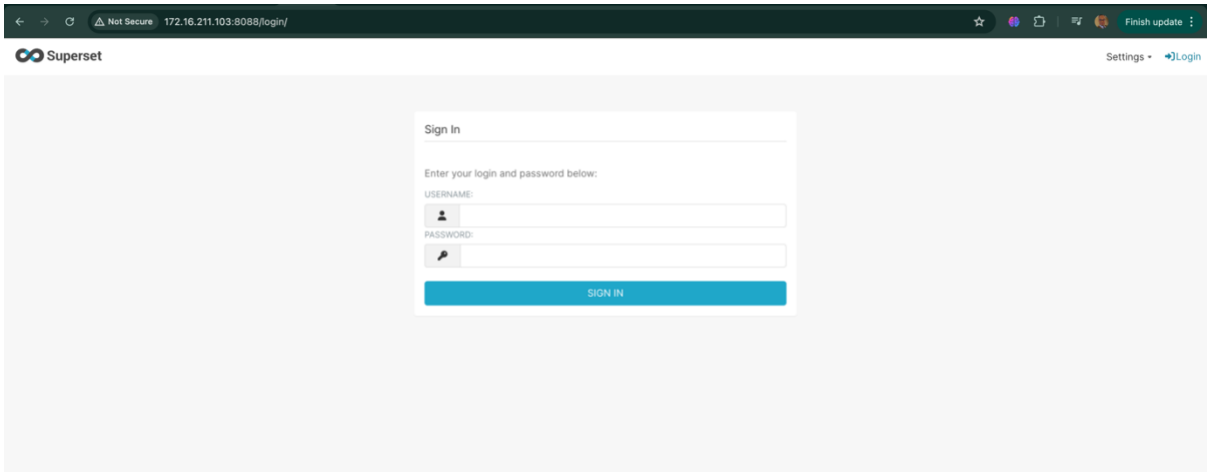
- Start docker

```
hdoop@hadoop:~$ cd ~/superset/superset
hdoop@hadoop:~/superset/superset$ docker-compose up -d
[+] Running 38/38
 ✔ db 14 layers [##############]      0B/0B      Pulled                                                    128.1s
   ✔ 559a76444520 Pull complete                                                                            85.2s
   ✔ 059b5c43db41 Pull complete                                                                            85.3s
   ✔ 7ac4aa6c99e9 Pull complete                                                                            85.4s
   ✔ f695f6dafe22 Pull complete                                                                            85.4s
   ✔ 3531e6d72caa Pull complete                                                                            85.7s
   ✔ 8460f5c0f010 Pull complete                                                                            85.9s
   ✔ c419f8ddd2fb Pull complete                                                                            85.9s
   ✔ e54f6c55c74f Pull complete                                                                            87.3s
   ✔ 414f2433a842 Pull complete                                                                           122.5s
   ✔ 685dc4594efc Pull complete                                                                           122.5s
   ✔ f8589575005d Pull complete                                                                           122.5s
   ✔ c34aeee549ee Pull complete                                                                           122.5s
   ✔ 95867d3b307d Pull complete                                                                           122.6s
   ✔ fb5080f3ceef Pull complete                                                                           122.6s
 ✔ superset 14 layers [##############]      0B/0B      Pulled                                                 99.6s
   ✔ 22d97f6a5d13 Pull complete                                                                            10.0s
   ✔ b41a1d042542 Pull complete                                                                            10.2s
   ✔ eee500b73766 Pull complete                                                                            10.7s
   ✔ 456660446522 Pull complete                                                                            10.7s
   ✔ bde57e7cd6a8 Pull complete                                                                            11.0s
   ✔ 1aa1008a895f Pull complete                                                                            11.0s
   ✔ ee246f9d15bc Pull complete                                                                            82.8s
   ✔ 47b5c113d190 Pull complete                                                                            82.8s
   ✔ 2e3dced614cf Pull complete                                                                            82.8s
   ✔ 0f88779ba806 Pull complete                                                                            93.2s
   ✔ 78cd5b17b843 Pull complete                                                                            93.9s
   ✔ 21a2db0ac509 Pull complete                                                                            94.1s
   ✔ f70e67a57df7 Pull complete                                                                            94.1s
   ✔ af474dd739d6 Pull complete                                                                            94.2s
 ✔ redis 7 layers [########]      0B/0B      Pulled                                                        118.3s
   ✔ 533ebe80b764 Pull complete                                                                            92.5s
   ✔ 7bb4ee5b1ade Pull complete                                                                            93.4s
   ✔ c71da199a45b Pull complete                                                                            96.3s
   ✔ 25c04f2d495a Pull complete                                                                           108.2s
   ✔ f5750872ffb4 Pull complete                                                                           108.2s
   ✔ 4f4fb700ef54 Pull complete                                                                           108.2s
   ✔ fe3c2d43235d Pull complete                                                                           112.3s
[+] Running 5/5
 ✔ Network superset_default   Created                                                                       0.1s
 ✔ Volume "superset_db_home"  Created                                                                       0.0s
 ✔ Container superset_redis   Started                                                                       5.4s
 ✔ Container superset_db      Started                                                                       5.4s
 ✔ Container superset_app     Started                                                                       0.6s
hdoop@hadoop:~/superset/superset$
```

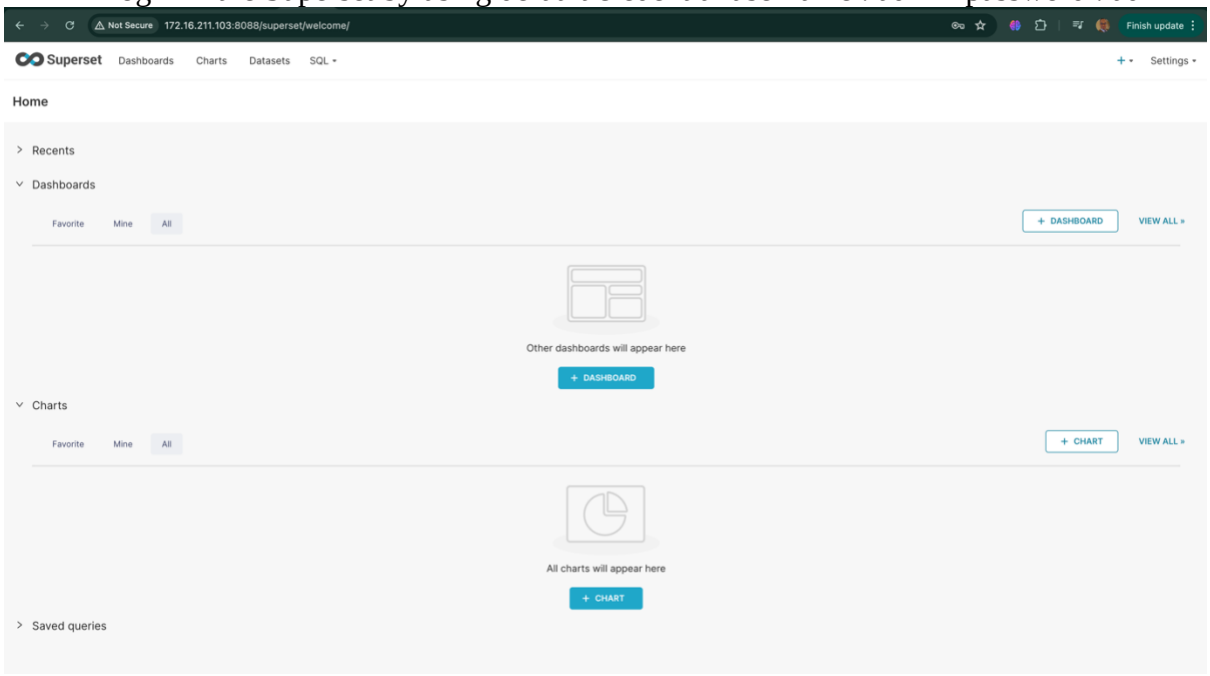
- Accessing the contained to create user for the Apache Superset

```
hdoop@hadoop:~/superset/superset$ docker exec -it superset_app /bin/bash
superset@118d4d0add63:/app$ superset fab create-admin --username admin --firstname Superset --lastname Admin --email admin@superset.com --pass
word admin
superset db upgrade
superset load_examples
superset init
exit
Loaded your LOCAL configuration at [superset_config.py]
logging was configured successfully
2024-06-28 08:17:36,086:INFO:superset.utils.logging_configurator:logging was configured successfully
2024-06-28 08:17:36,090:INFO:root:Configured event logger of type <class 'superset.utils.log.DBEventLogger'>
/usr/local/lib/python3.10/site-packages/flask_limiter/extension.py:293: UserWarning: Using the in-memory storage for tracking rate limits as n
o storage was explicitly specified. This is not recommended for production use. See: https://flask-limiter.readthedocs.io#configuring-a-storag
e-backend for documentation about configuring the storage backend.
  warnings.warn(
Recognized Database Authentications.
Admin User admin created.
Loaded your LOCAL configuration at [superset_config.py]
logging was configured successfully
2024-06-28 08:17:38,652:INFO:superset.utils.logging_configurator:logging was configured successfully
2024-06-28 08:17:38,654:INFO:root:Configured event logger of type <class 'superset.utils.log.DBEventLogger'>
/usr/local/lib/python3.10/site-packages/flask_limiter/extension.py:293: UserWarning: Using the in-memory storage for tracking rate limits as n
o storage was explicitly specified. This is not recommended for production use. See: https://flask-limiter.readthedocs.io#configuring-a-storag
e-backend for documentation about configuring the storage backend.
  warnings.warn(
```

- Accessing the Superset in the browser through port 8080



- Login in the Superset by using default credential username : admin password : admin



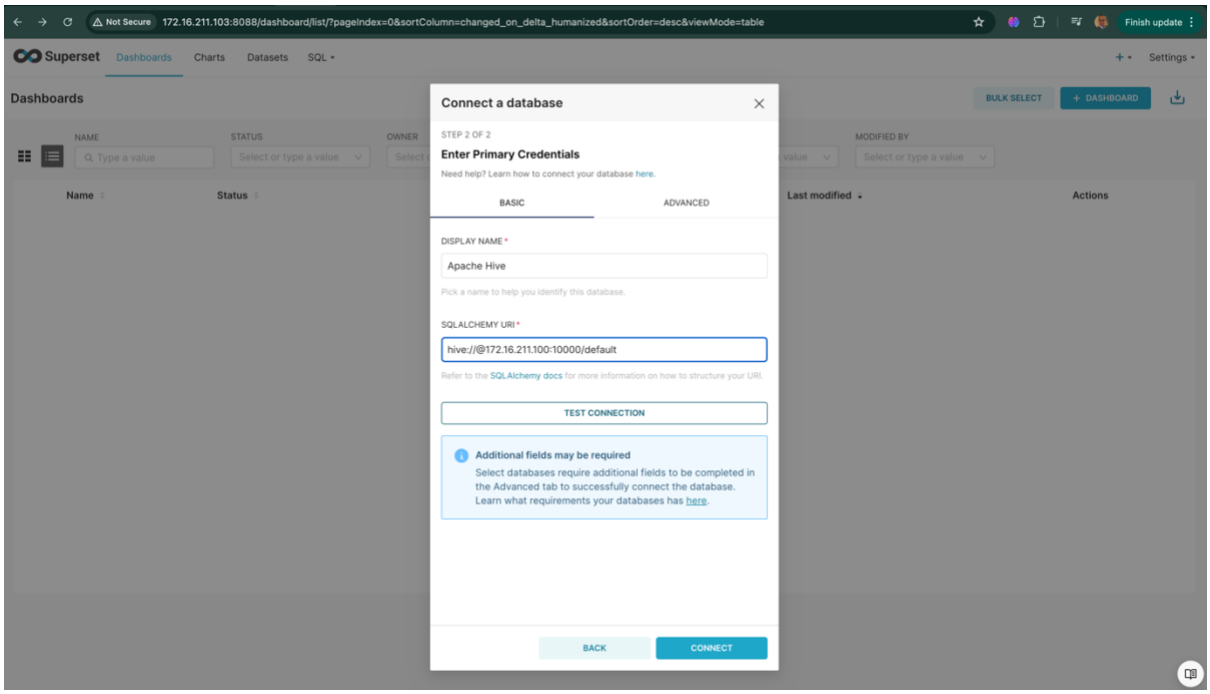
- Integrating Superset with Hadoop

To integrate Hive and Hadoop modify “core-site.xml” to allow user impersonation.

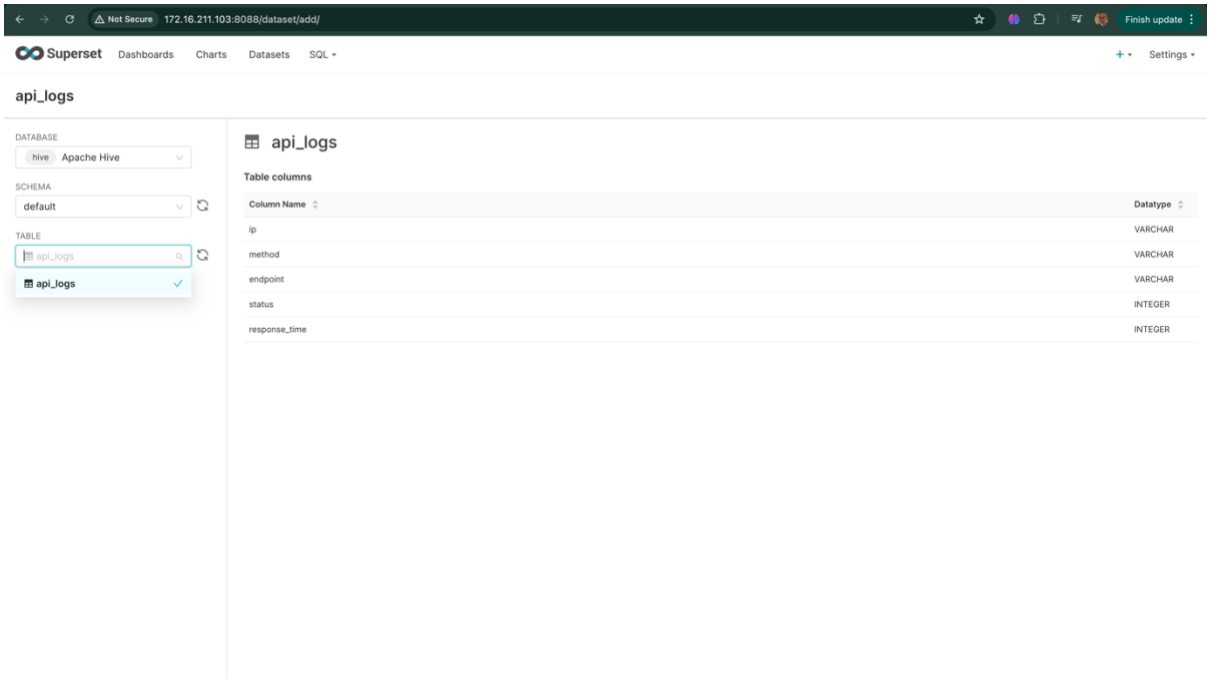
```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
<property>
  <name>hadoop.tmp.dir</name>
  <value>/home/hdoop/tmpdata</value>
</property>
<property>
  <name>fs.default.name</name>
  <value>hdfs://master:9000</value>
</property>
<property>
    <name>hadoop.proxyuser.hdoop.groups</name>
    <value>*</value>
</property>
<property>
    <name>hadoop.proxyuser.hdoop.hosts</name>
    <value>*</value>
</property>
</configuration>
~
~
~
~
~
~
~
~
```

Connecting Hive Database in the Hive.
Login in the Apache Superset.

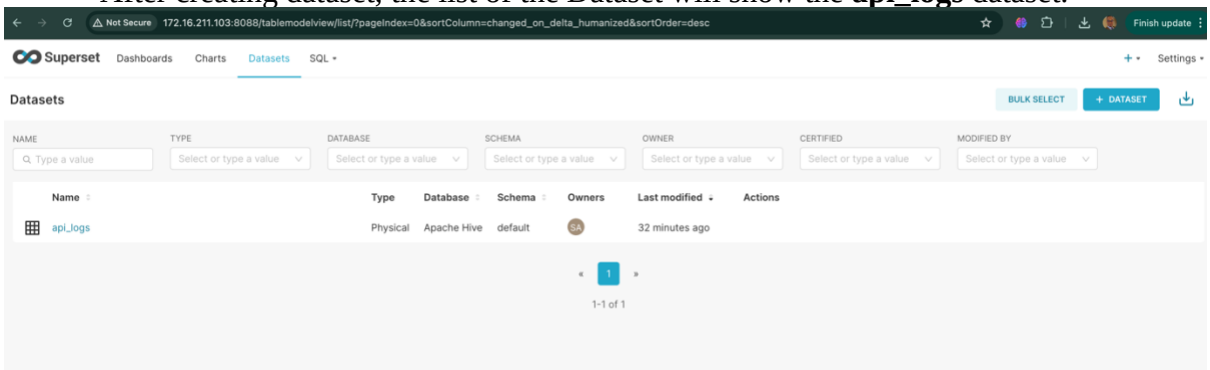
- Go to `Sources` -> `Databases` and click on `+ Database`.
- Select Apache Hive and set url to `hive://@172.16.211.100:10000/default`



- Click Connect and if its successfully, go to the Datasets select database schema and table to create a dataset.



- After creating dataset, the list of the Dataset will show the **api_logs** dataset.



Dashboard

The Log Analysis Dashboard is designed to provide a comprehensive overview of web server log data, enabling the identification and visualization of anomalies and normal request patterns. By analysing key metrics such as HTTP status codes, methods, URLs, IP addresses, and response sizes, the dashboard aims to enhance the understanding of web traffic behaviour and potential issues. This document summarizes the various graphs presented in the dashboard, each tailored to highlight different aspects of the log data.

Key Features of the Dashboard

1. **Anomalies Detection**.
   - ⇒ The dashboard identifies and visualizes anomalies in the log data, which are characterized by unusual HTTP status codes, large response sizes, or high-frequency requests.
   - ⇒ Anomalies are critical as they can indicate potential security threats, server issues, or unusual user behavior.

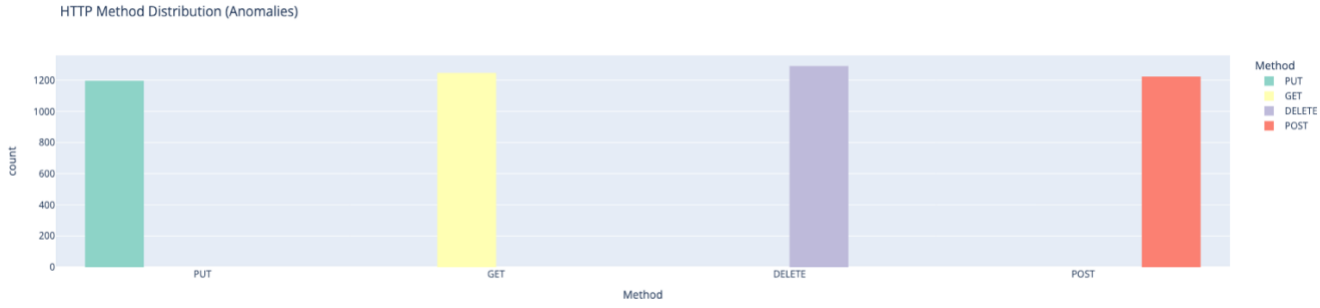
2. **Normal Requests Analysis**:
    ⇒ The dashboard also provides insights into the normal request patterns, helping to establish baselines for regular web traffic.
    ⇒ This analysis is essential for understanding typical user interactions and server performance under normal conditions.
3. **Time Series Analysis**:
    ⇒ A time series graph is included to show the trend of log entries over time, aiding in the detection of patterns or anomalies that evolve.

Summary of the Graphs

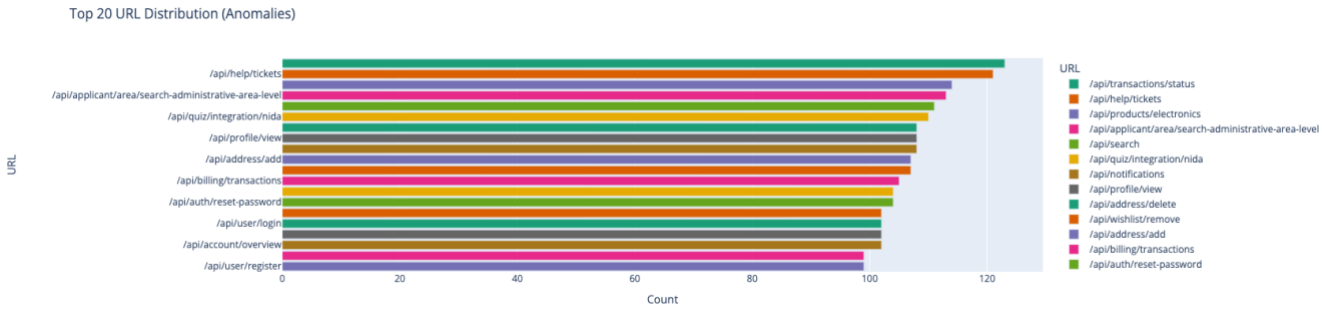
Each graph in the dashboard serves a specific purpose in analyzing the log data:

1. **Anomalies Section**:
    ⇒ **Status Code Distribution (Anomalies)**. A histogram showing the distribution of HTTP status codes among anomalies, helping to identify common error types.

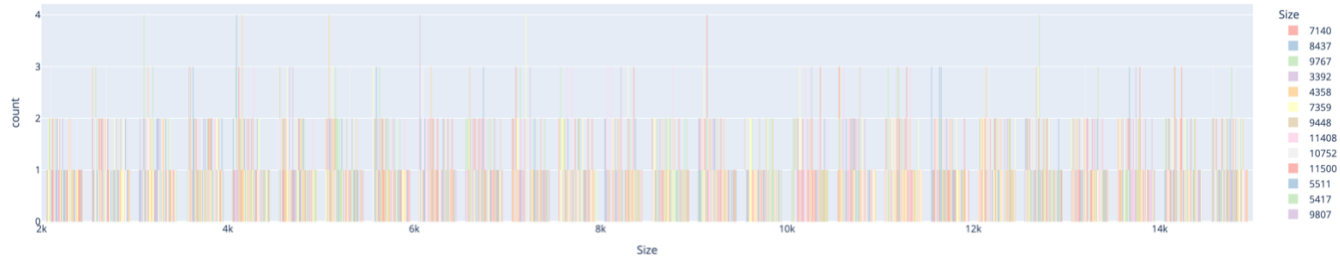
    ⇒ **HTTP Method Distribution (Anomalies)**. A histogram displaying the distribution of HTTP methods for the anomalies, indicating which methods are prone to issues.



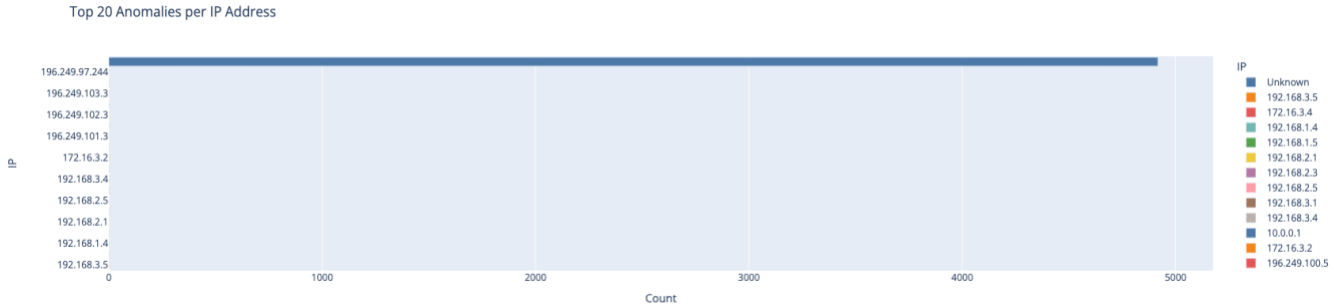
    ⇒ **Top 20 URL Distribution (Anomalies)**. A bar chart depicting the most frequently accessed URLs among anomalies, highlighting problematic endpoints.



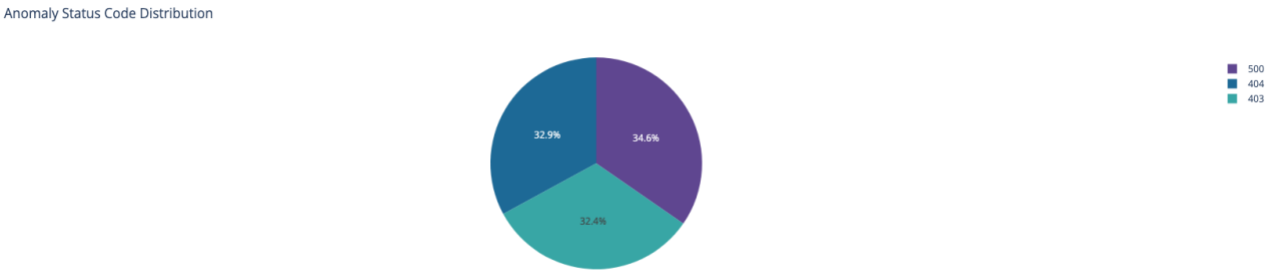
⇒ **Response Size Distribution (Anomalies)**. A histogram illustrating the distribution of response sizes for anomalies, which can reveal unusual data transfers.



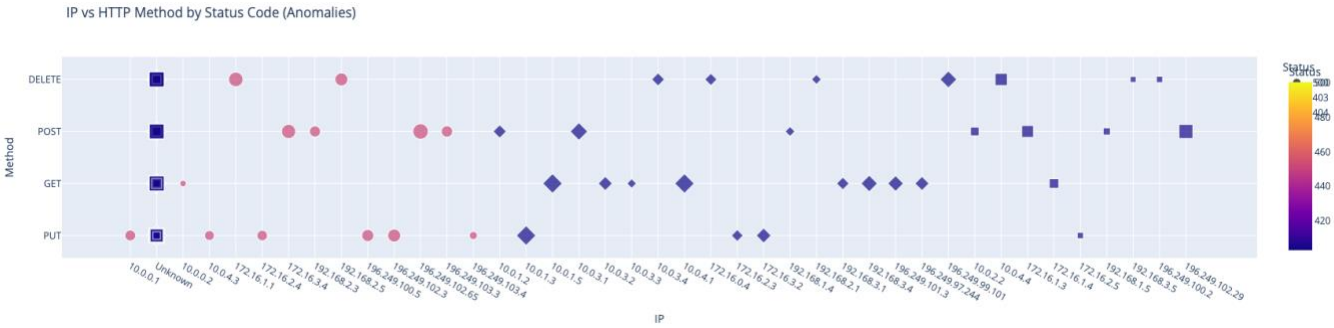
⇒ **Top 20 Anomalies per IP Address**. A bar chart showing the IP addresses most frequently associated with anomalies, potentially indicating sources of attacks or issues.



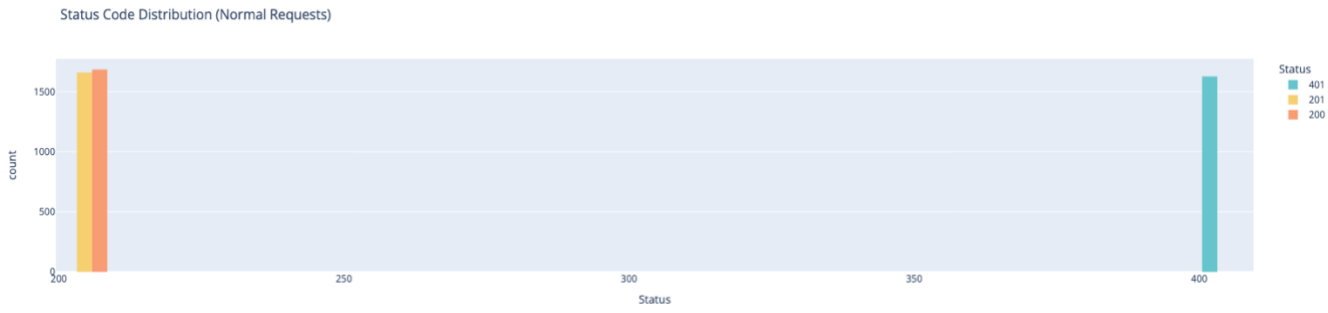
⇒ **Anomaly Status Code Distribution (Pie Chart)**. A pie chart summarizing the distribution of HTTP status codes among anomalies, providing a quick overview of error types.



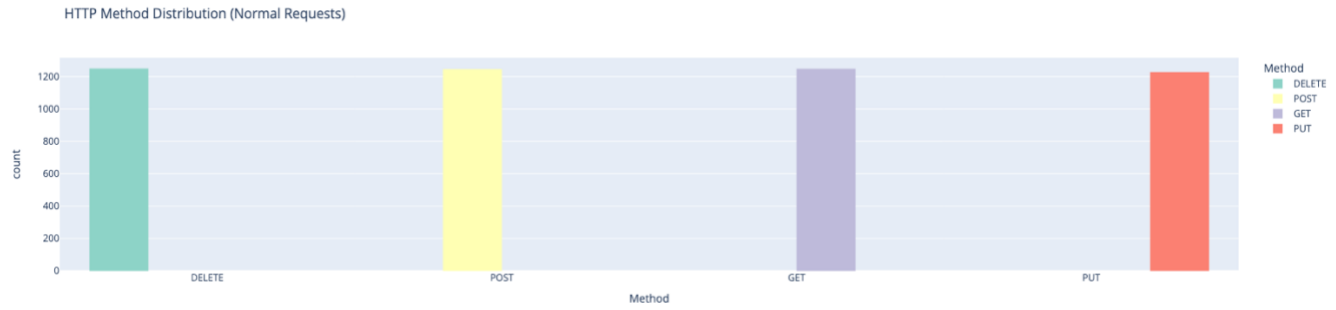
⇒ **IP vs HTTP Method by Status Code (Anomalies)**. A scatter plot showing the relationship between IP addresses and HTTP methods for anomalies, offering insights into the nature of issues based on IP and method combinations.



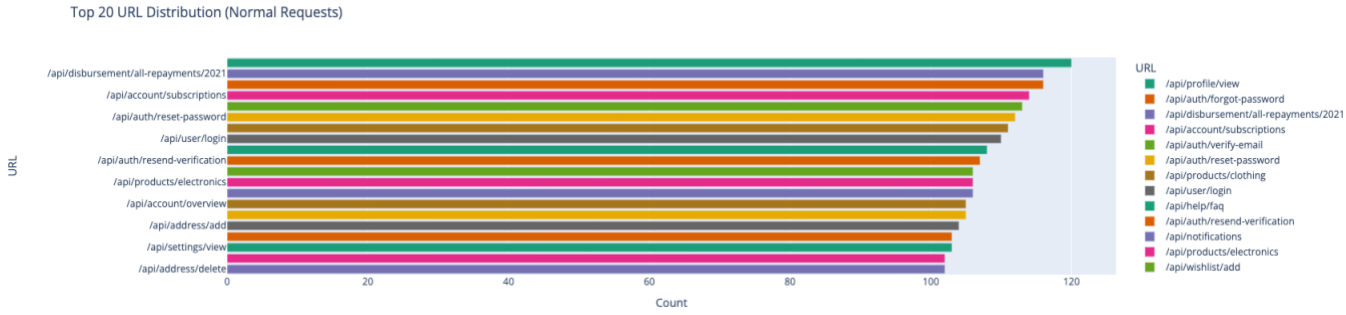
2. **Normal Requests Section**:
   ⇒ **Status Code Distribution (Normal Requests)**: A histogram showing the distribution of HTTP status codes among normal requests, helping to understand the typical success and error rates.



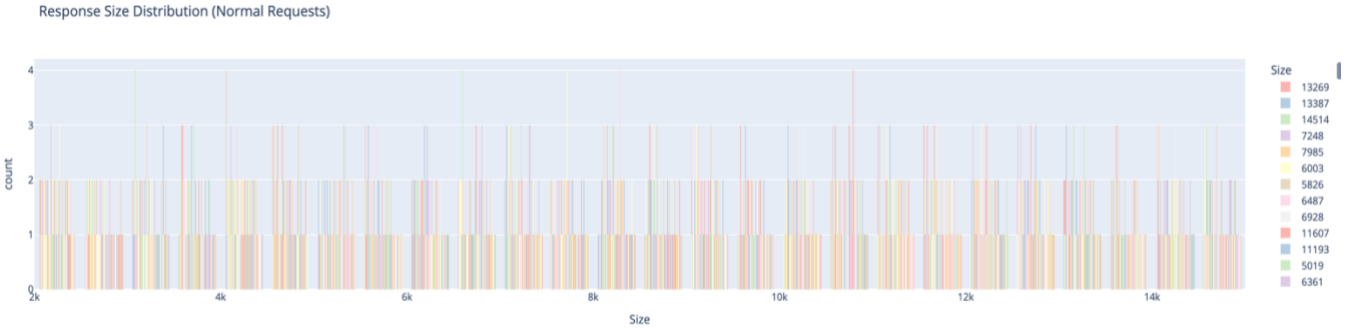
⇒ **HTTP Method Distribution (Normal Requests)**: A histogram displaying the distribution of HTTP methods for normal requests, indicating common user actions.



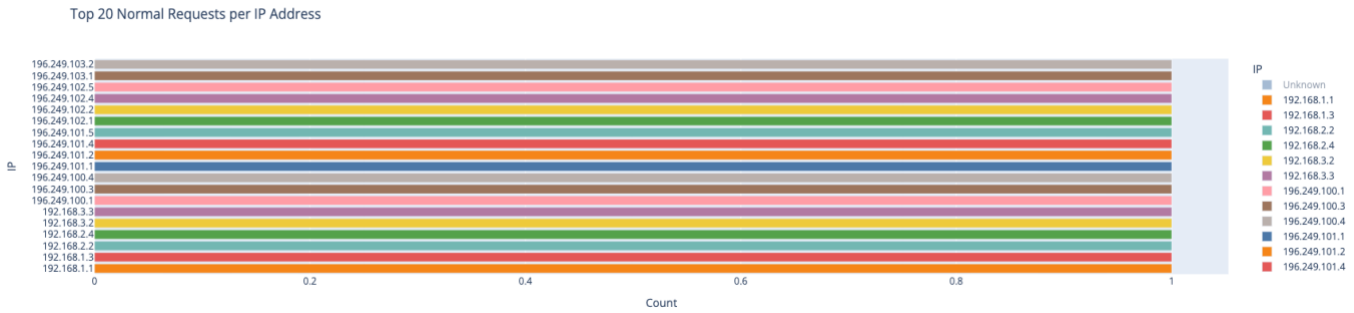
⇒ **Top 20 URL Distribution (Normal Requests)**: A bar chart depicting the most frequently accessed URLs among normal requests, highlighting popular endpoints.



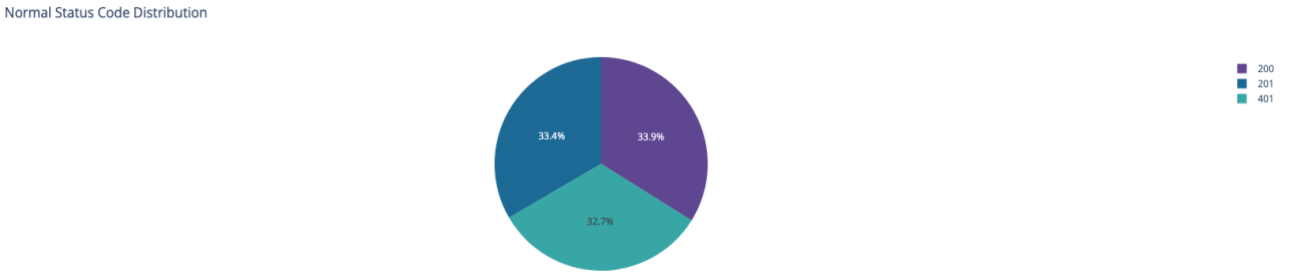
⇒ **Response Size Distribution (Normal Requests)**: A histogram illustrating the distribution of response sizes for normal requests, showing the typical data transfer sizes.



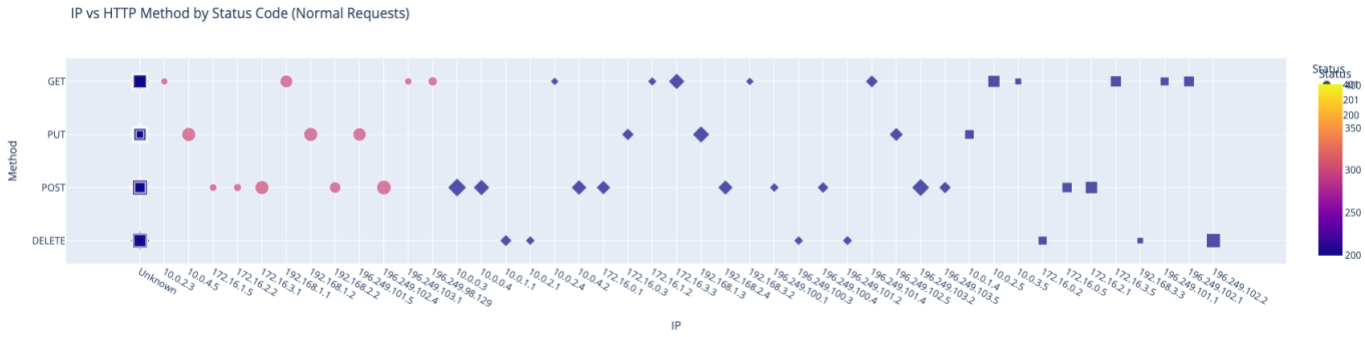
⇒ **Top 20 Normal Requests per IP Address**: A bar chart showing the IP addresses most frequently associated with normal requests, indicating active users or clients.



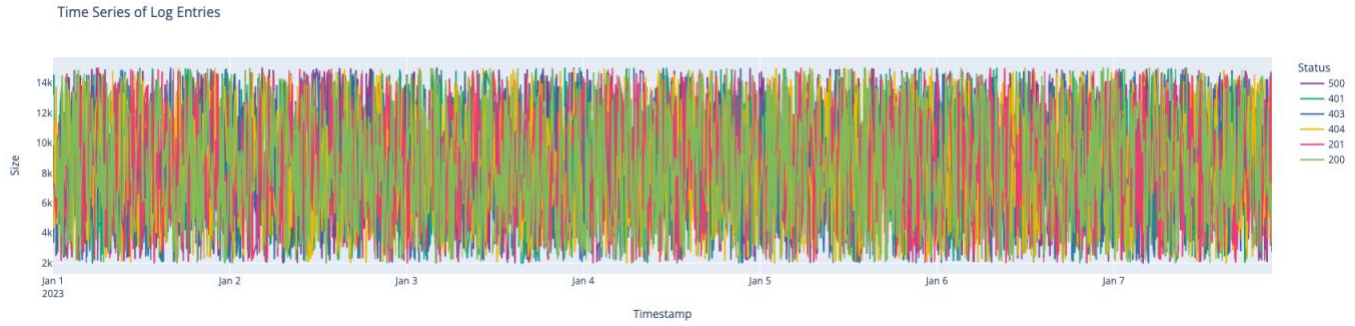
⇒ **Normal Status Code Distribution (Pie Chart)**: A pie chart summarizing the distribution of HTTP status codes among normal requests, providing an overview of request outcomes.

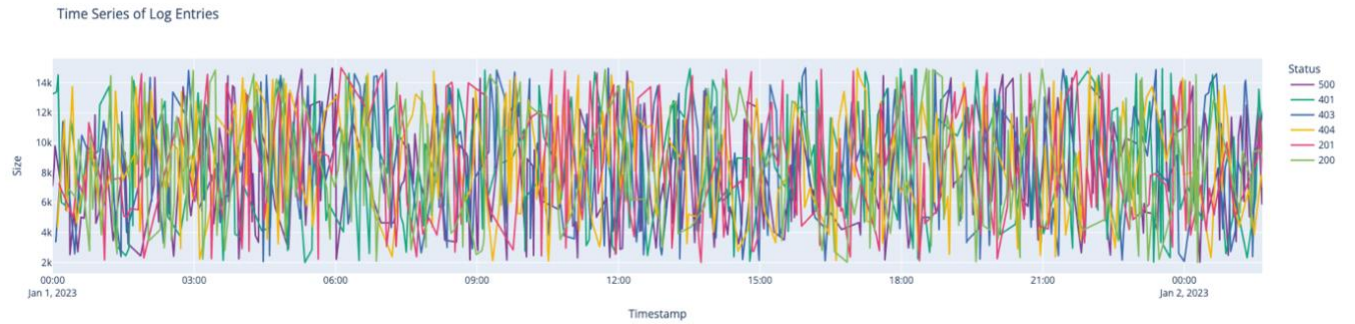


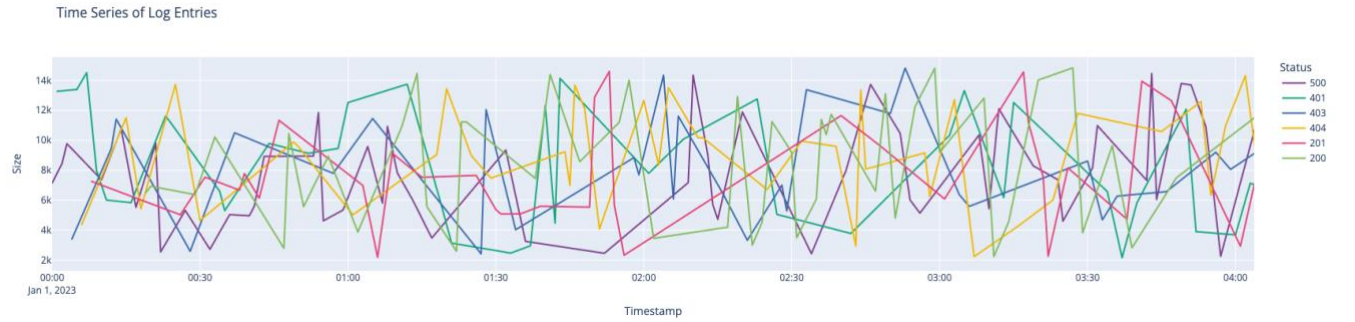
⇒ **IP vs HTTP Method by Status Code (Normal Requests)**: A scatter plot showing the relationship between IP addresses and HTTP methods for normal requests, offering insights into typical interactions based on IP and method combinations.

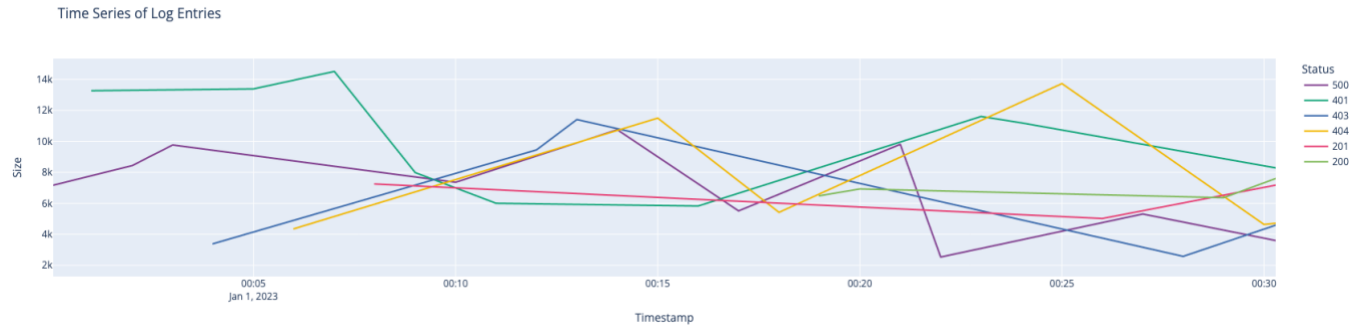


3. **Time Series Analysis**:
   ⇒ **Time Series of Log Entries**: A line graph showing the size of log entries over time, helping to detect trends or periodic patterns in web traffic.


Time Series of Log Entries
Size
14k
12k
10k
8k
6k
4k
2k
Jan 1
2023
Jan 2
Jan 3
Jan 4
Jan 5
Jan 6
Jan 7
Timestamp
Status
500
401
403
404
201
200


Time Series of Log Entries
Size
14k
12k
10k
8k
6k
4k
2k
00:00
Jan 1, 2023
03:00
06:00
09:00
12:00
15:00
18:00
21:00
00:00
Jan 2, 2023
Timestamp
Status
500
401
403
404
201
200


Time Series of Log Entries
Size
14k
12k
10k
8k
6k
4k
2k
00:00
Jan 1, 2023
00:30
01:00
01:30
02:00
02:30
03:00
03:30
04:00
Timestamp
Status
500
401
403
404
201
200


Time Series of Log Entries
Size
14k
12k
10k
8k
6k
4k
2k
00:05
Jan 1, 2023
00:10
00:15
00:20
00:25
00:30
Timestamp
Status
500
401
403
404
201
200

Conclusion

In conclusion, this report provided a comprehensive guide on setting up a 2-node Hadoop cluster and configuring Kafka to collect distributed data from multiple nodes, such as web logs. By implementing MapReduce for web log analysis and setting up a data visualization dashboard, the system ensures efficient data processing and real-time insights.

Key Achievements

1. Hadoop Cluster Setup.
    ⇒ Successfully established a 2-node Hadoop cluster, enabling scalable and distributed data storage and processing.
2. Kafka Configuration.
    ⇒ Configured Kafka for real-time data collection from multiple nodes, facilitating efficient ingestion and handling of web logs.
3. MapReduce Implementation.
    ⇒ Developed and deployed MapReduce code for analyzing web log data, allowing for detailed and distributed data processing.
4. Data Visualization Dashboard.
    ⇒ Created a comprehensive dashboard for visualizing web log analysis results, providing valuable insights into web traffic and potential anomalies.

Benefits

- *Enhanced Data Processing*. The combination of Hadoop and Kafka enables the handling of large-scale, distributed data efficiently.
- *Real-Time Insights*. The dashboard offers real-time visibility into the collected data, highlighting anomalies and normal patterns for proactive decision-making.
- *Scalability*. The 2-node cluster setup ensures that the system can scale as data volume grows, maintaining performance and reliability.

References

[1] S. Bhuvaneswari and T. Anand, "A Comparative Study of Different Log Analyzer Tools to Analyze User Behaviors," *nternational Journal on Recent and Innovation Trends in Computing and Communication*, vol. 3, pp. 2997-3002, 2015.

[2] M. S. Hossain, K. B. Pratik and A. Rahman, "Develop a Model to Secure and Optimize Distributed File Systems for ISP Log Management," *Journal of Financial Services Marketing*, pp. 1-6, 2023.