

Title of the project

Aviation Data Analysis

- Create an account in AWS
- Create an instance.
- Connect to the instance from your local machine using SSH command with the corresponding pem/key file for the instance.
- Install Hadoop in ec2 instance – ubuntu.
- Ubuntu version used 22.04.
- Steps to install Hadoop in ubuntu server.
- Java Installation:
`sudo apt update`
`sudo apt install openjdk-8-jdk -y`
`java -version`
- Hadoop Installation:
- Run the following command to create a new user with the name “hadoop”:
`sudo adduser Hadoop`
Note: click enter with all the default values while adding a user.
- Switch to the newly created hadoop user
`su – Hadoop`
- Now configure password-less SSH access for the newly created hadoop user. Generate an SSH keypair first:
`ssh-keygen -t rsa`
- Copy the generated public key to the authorized key file and set the proper permissions:
`cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys`
`chmod 640 ~/.ssh/authorized_keys`
- Now try to SSH to the localhost.
`ssh localhost`
- Use the following command to download Hadoop 3.3.4:
`wget https://dlcdn.apache.org/hadoop/common/hadoop-3.3.4/hadoop-3.3.4.tar.gz`

- Once you've downloaded the file, you can unzip it to a folder on your hard drive:

```
tar xzf hadoop-3.3.4.tar.gz
```

- Rename the extracted folder to remove version information. This is an optional step, but if you don't want to rename, then adjust the remaining configuration paths.

```
mv hadoop-3.3.4 hadoop
```

- Next, you will need to configure Hadoop and Java Environment Variables on your system.
- Open the ~/.bashrc file in your favorite text editor:

```
nano ~/.bashrc
```

- Append the below lines to the file. You can find the JAVA_HOME location by running `dirname $(dirname $(readlink -f $(which java)))` command on the terminal.

```
export JAVA_HOME= /usr/lib/jvm/java-8-openjdk-amd64/  
export HADOOP_HOME=/home/hadoop/hadoop  
export HADOOP_INSTALL=$HADOOP_HOME  
export HADOOP_MAPRED_HOME=$HADOOP_HOME  
export HADOOP_COMMON_HOME=$HADOOP_HOME  
export HADOOP_HDFS_HOME=$HADOOP_HOME  
export HADOOP_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"
```

- Save the file and close it.
- Load the above configuration in the current environment.

```
source ~/.bashrc
```

- You also need to configure JAVA_HOME in hadoop-env.sh file. Edit the Hadoop environment variable file in the text editor:

```
nano $HADOOP_HOME/etc/hadoop/hadoop-env.sh
```

- Search for the "export JAVA_HOME" and configure it.
- Next is to configure Hadoop configuration files available under etc directory.

- First, you will need to create the namenode and datanode directories inside the Hadoop user home directory. Run the following command to create both directories:

```
mkdir -p ~/hadoopdata/hdfs/{namenode,datanode}
```

- Next, edit the core-site.xml file and update with your system hostname:

```
nano $HADOOP_HOME/etc/hadoop/core-site.xml
```

- Change the following name as per your system hostname:

```
<configuration>

  <property>

    <name>fs.defaultFS</name>

    <value>hdfs://localhost:9000</value>

  </property>

</configuration>
```

- Save and close the file.
- Then, edit the hdfs-site.xml file.

```
nano $HADOOP_HOME/etc/hadoop/hdfs-site.xml
```

- Change the NameNode and DataNode directory paths as shown below:

```
<configuration>

  <property>

    <name>dfs.replication</name>

    <value>1</value>

  </property>

  <property>

    <name>dfs.name.dir</name>

    <value>file:///home/hadoop/hadoopdata

      /hdfs/namenode</value>

  </property>

  <property>
```

```
<name>dfs.data.dir</name>
<value>file:///home/hadoop/hadoopdata
/hdfs/datanode</value>
</property>
</configuration>
```

- Save and close the file.
- Then, edit the mapred-site.xml file.

```
nano $HADOOP_HOME/etc/hadoop/mapred-site.xml
```

- Make the following changes:

```
<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
</configuration>
```

- Save and close the file.
- Then, edit the yarn-site.xml file:

```
nano $HADOOP_HOME/etc/hadoop/yarn-site.xml
```

- Make the following changes:

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

- Save the file and close it.

- Before starting the Hadoop cluster. You will need to format the Namenode as a hadoop user.
- Run the following command to format the Hadoop Namenode.

`hdfs namenode -format`

- Once the namenode directory is successfully formatted with hdfs file system, you will see the message "Storage directory /home/hadoop/hadoopdata/hdfs/namenode has been successfully formatted".
- Then start the Hadoop cluster with the following command.

`start-all.sh`

- Use the below command to copy the data.

`scp -i "aviationproject.pem" -A AviationData.csv ubuntu@ec2-3-90-199-230.compute-1.amazonaws.com:/home/ubuntu`

```

Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.19.0-1025-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Sun Jun 18 08:40:31 UTC 2023

System load:  0.03173828125   Processes:           267
Usage of /:   65.7% of 7.57GB   Users logged in:     2
Memory usage: 21%             IPv4 address for ens3: 172.31.25.234
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

65 updates can be applied immediately.
46 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Sun Jun 18 08:30:20 2023 from 49.205.211.188
ubuntu@ip-172-31-25-234:~$ sudo cp -a /home/ubuntu/ /home/hadoop/
ubuntu@ip-172-31-25-234:~$ sudo cp -a /home/ubuntu/AviationData.csv /home/hadoop/
ubuntu@ip-172-31-25-234:~$ sudo su hadoop
hadoop@ip-172-31-25-234:/home/ubuntu$ cd ~
hadoop@ip-172-31-25-234:~$ ls
AviationData.csv  hadoop  hadoop-3.3.4.tar.gz  hadoopdata  id_rsa.pub  id_rsa.pub.pub  ubuntu
hadoop@ip-172-31-25-234:~$

```

```
hadoop@ip-172-31-25-234: ~  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
  
Last login: Sun Jun 18 08:30:20 2023 from 49.205.211.188  
ubuntu@ip-172-31-25-234:~$ sudo cp -a /home/ubuntu/ /home/hadoop/  
ubuntu@ip-172-31-25-234:~$ sudo cp -a /home/ubuntu/AviationData.csv /home/hadoop/  
ubuntu@ip-172-31-25-234:~$ sudo su hadoop  
hadoop@ip-172-31-25-234:/home/ubuntu$ cd ~  
hadoop@ip-172-31-25-234:~$ ls  
AviationData.csv  hadoop  hadoop-3.3.4.tar.gz  hadoopdata  id_rsa.pub  id_rsa.pub.pub  ubuntu  
hadoop@ip-172-31-25-234:~$ jps  
9970 Jps  
7763 DataNode  
8410 NodeManager  
7580 NameNode  
7996 SecondaryNameNode  
8253 ResourceManager  
hadoop@ip-172-31-25-234:~$ pwd  
/home/hadoop  
hadoop@ip-172-31-25-234:~$ hdfs dfs -ls /user/hadoop  
ls: '/user/hadoop': No such file or directory  
hadoop@ip-172-31-25-234:~$ hdfs dfs -mkdir /user  
mkdir: '/user': File exists  
hadoop@ip-172-31-25-234:~$ hdfs dfs -ls /user/  
Found 1 items  
drwxr-xr-x - hadoop supergroup 0 2023-06-18 08:04 /user/harika  
hadoop@ip-172-31-25-234:~$ hdfs dfs -ls /user/harika/  
Found 1 items  
drwxr-xr-x - hadoop supergroup 0 2023-06-18 08:04 /user/harika/data  
hadoop@ip-172-31-25-234:~$ hdfs dfs -put /home/hadoop/AviationData.csv /user/harika/data  
hadoop@ip-172-31-25-234:~$ hdfs dfs -ls /user/harika/data  
Found 1 items  
-rw-r--r-- 1 hadoop supergroup 22379062 2023-06-18 08:43 /user/harika/data/AviationData.csv  
hadoop@ip-172-31-25-234:~$
```

- Next install druid in ec2 instance – ubuntu.
- Below are the steps to install the druid in ubuntu:

```
wget https://dlcdn.apache.org/druid/24.0.2/apache-druid-24.0.2-bin.tar.gz  
tar xvfz apache-druid-24.0.2-bin.tar.gz  
cd apache-druid-24.0.2  
export DRUID_HOME=/home/ubuntu/apache-druid-24.0.2  
./bin/start-micro-quickstart
```

ubuntu@ip-172-31-25-234: ~/apache-druid-24.0.2

Memory usage: 2% IPv4 address for ens3: 172.31.25.234
Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.

5 updates can be applied immediately.

6 of these updates are standard security updates.

To see these additional updates run: `apt list --upgradable`

Enable ESM Apps to receive additional future security updates.

See <https://ubuntu.com/esm> or run: `sudo pro status`

Last login: Sun Jun 18 07:24:25 2023 from 49.205.211.188

ubuntu@ip-172-31-25-234:~\$ java -version

openjdk version "1.8.0_362"

OpenJDK Runtime Environment (build 1.8.0_362-8u372-ga~us1-0ubuntu1~22.04-b09)

OpenJDK 64-Bit Server VM (build 25.362-b09, mixed mode)

ubuntu@ip-172-31-25-234:~\$ wget https://dlcdn.apache.org/druid/24.0.2/apache-druid-24.0.2-bin.tar.gz

--2023-06-18 07:28:44-- https://dlcdn.apache.org/druid/24.0.2/apache-druid-24.0.2-bin.tar.gz

Resolving dlcdn.apache.org (dlcdn.apache.org)... 151.101.2.132, 2a04:4e42::644

Connecting to dlcdn.apache.org (dlcdn.apache.org)|151.101.2.132|:443... connected.

HTTP request sent, awaiting response... 200 OK

Length: 397459991 (379M) [application/x-gzip]

Saving to: 'apache-druid-24.0.2-bin.tar.gz'

apache-druid-24.0.2-bin.tar.gz 100%[=====>] 379.05M 359MB/s in 1.1s

2023-06-18 07:28:45 (359 MB/s) - 'apache-druid-24.0.2-bin.tar.gz' saved [397459991/397459991]

ubuntu@ip-172-31-25-234: ~/apache-druid-24.0.2

apache-druid-24.0.2-bin.tar.gz 100%[=====>] 379.05M 359MB/s in 1.1s

2023-06-18 07:28:45 (359 MB/s) - 'apache-druid-24.0.2-bin.tar.gz' saved [397459991/397459991]

ubuntu@ip-172-31-25-234:~\$ tar xvfz apache-druid-24.0.2-bin.tar.gz

apache-druid-24.0.2/LICENSE

apache-druid-24.0.2/NOTICE

apache-druid-24.0.2/README

apache-druid-24.0.2/extensions/druid-pac4j/byte-buddy-1.12.7.jar

apache-druid-24.0.2/extensions/druid-pac4j/slf4j-api-1.7.36.jar

apache-druid-24.0.2/extensions/druid-pac4j/objenesis-3.2.jar

apache-druid-24.0.2/extensions/druid-pac4j/pac4j-core-3.8.3.jar

apache-druid-24.0.2/extensions/druid-pac4j/activation-1.1.1.jar

apache-druid-24.0.2/extensions/druid-pac4j/pac4j-oidc-3.8.3.jar

apache-druid-24.0.2/extensions/druid-pac4j/lang-tag-1.7.jar

apache-druid-24.0.2/extensions/druid-pac4j/mockito-core-4.3.1.jar

apache-druid-24.0.2/extensions/druid-pac4j/jcip-annotations-1.0-1.jar

apache-druid-24.0.2/extensions/druid-pac4j/nimbus-jose-jwt-7.9.jar

apache-druid-24.0.2/extensions/druid-pac4j/byte-buddy-agent-1.12.7.jar

apache-druid-24.0.2/extensions/druid-pac4j/json-smart-2.3.jar

apache-druid-24.0.2/extensions/druid-pac4j/asm-9.3.jar

apache-druid-24.0.2/extensions/druid-pac4j/druid-pac4j-24.0.2.jar

apache-druid-24.0.2/extensions/druid-pac4j/accessors-smart-1.2.jar

apache-druid-24.0.2/extensions/druid-pac4j/oauth2-oidc-sdk-6.5.jar

apache-druid-24.0.2/extensions/druid-pac4j/javax.mail-1.6.1.jar

apache-druid-24.0.2/extensions/druid-aws-rds-extensions/aws-java-sdk-core-1.12.264.jar

apache-druid-24.0.2/extensions/druid-aws-rds-extensions/commons-logging-1.1.1.jar

apache-druid-24.0.2/extensions/druid-aws-rds-extensions/httpcore-4.4.11.jar

apache-druid-24.0.2/extensions/druid-aws-rds-extensions/jackson-dataformat-cbor-2.10.5.jar

apache-druid-24.0.2/extensions/druid-aws-rds-extensions/druid-aws-rds-extensions-24.0.2.jar

```

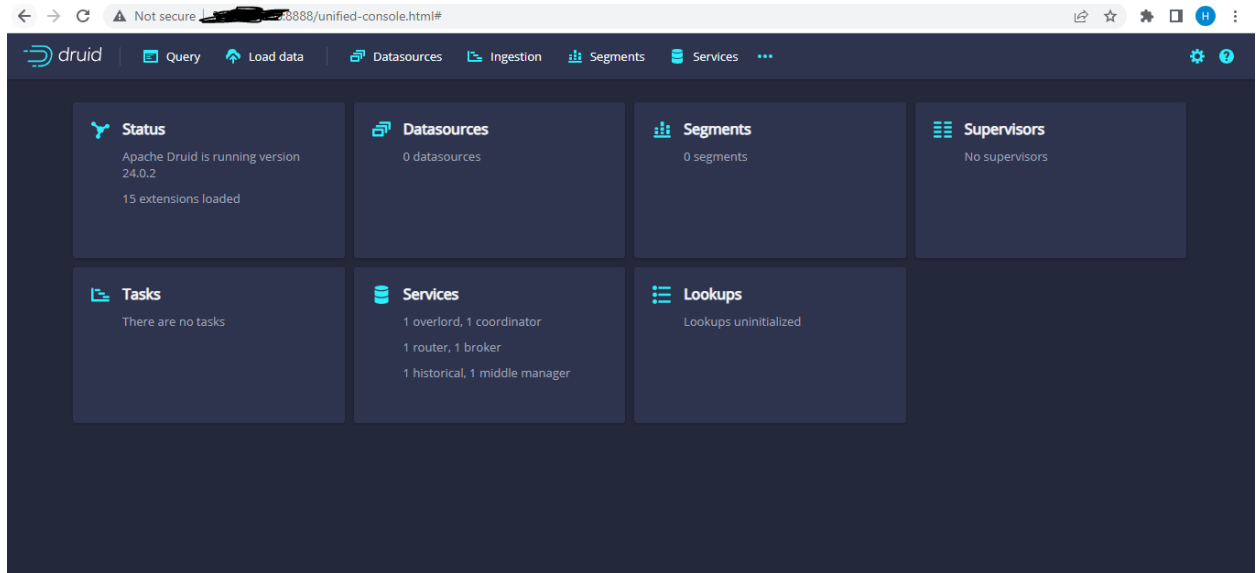
ubuntu@ip-172-31-25-234: ~/apache-druid-24.0.2
/usr/bin/java
ubuntu@ip-172-31-25-234:~/apache-druid-24.0.2$ java -version
openjdk version "1.8.0_362"
OpenJDK Runtime Environment (build 1.8.0_362-8u372-ga~us1-0ubuntu1~22.04-b09)
OpenJDK 64-Bit Server VM (build 25.362-b09, mixed mode)
ubuntu@ip-172-31-25-234:~/apache-druid-24.0.2$ vi ~/.bashrc
ubuntu@ip-172-31-25-234:~/apache-druid-24.0.2$ ubuntu@ip-172-31-25-234:~/apache-druid-24.0.2$
ubuntu@ip-172-31-25-234:~/apache-druid-24.0.2$ ./bin/start-micro-quickstart
[Sun Jun 18 07:37:42 2023] Starting Apache Druid.
[Sun Jun 18 07:37:42 2023] Open http://localhost:8888/ or http://ip-172-31-25-234:8888/ in your browser to access the web console.
[Sun Jun 18 07:37:42 2023] Or, if you have enabled TLS, use https on port 9088.
[Sun Jun 18 07:37:42 2023] Starting services with log directory [/home/ubuntu/apache-druid-24.0.2/log].
[Sun Jun 18 07:37:42 2023] Running command[zookeeper]: bin/run-zk conf
[Sun Jun 18 07:37:42 2023] Running command[coordinator-overlord]: bin/run-druid coordinator-overlord conf/druid/single-server/micro-quickstart
[Sun Jun 18 07:37:42 2023] Running command[broker]: bin/run-druid broker conf/druid/single-server/micro-quickstart
[Sun Jun 18 07:37:42 2023] Running command[router]: bin/run-druid router conf/druid/single-server/micro-quickstart
[Sun Jun 18 07:37:42 2023] Running command[historical]: bin/run-druid historical conf/druid/single-server/micro-quickstart
[Sun Jun 18 07:37:42 2023] Running command[middleManager]: bin/run-druid middleManager conf/druid/single-server/micro-quickstart
client_loop: send disconnect: Connection reset

C:\Users\DELL\Downloads>ssh -i "aviationproject.pem" ubuntu@ec2-3-90-199-230.compute-1.amazonaws.com
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.19.0-1025-aws x86_64)

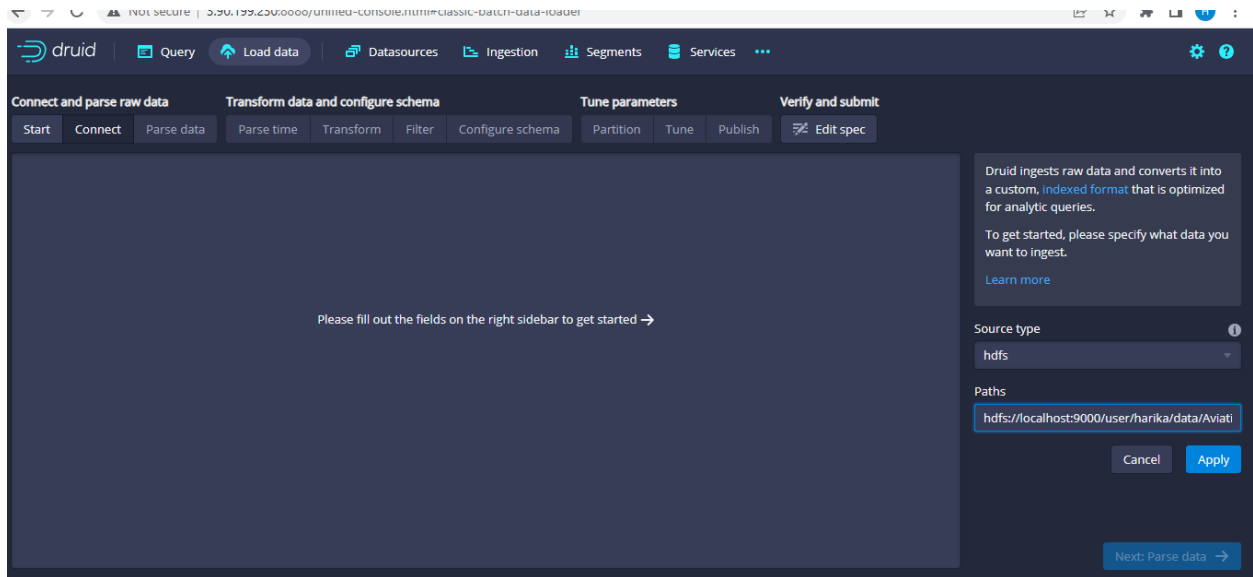
* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

```

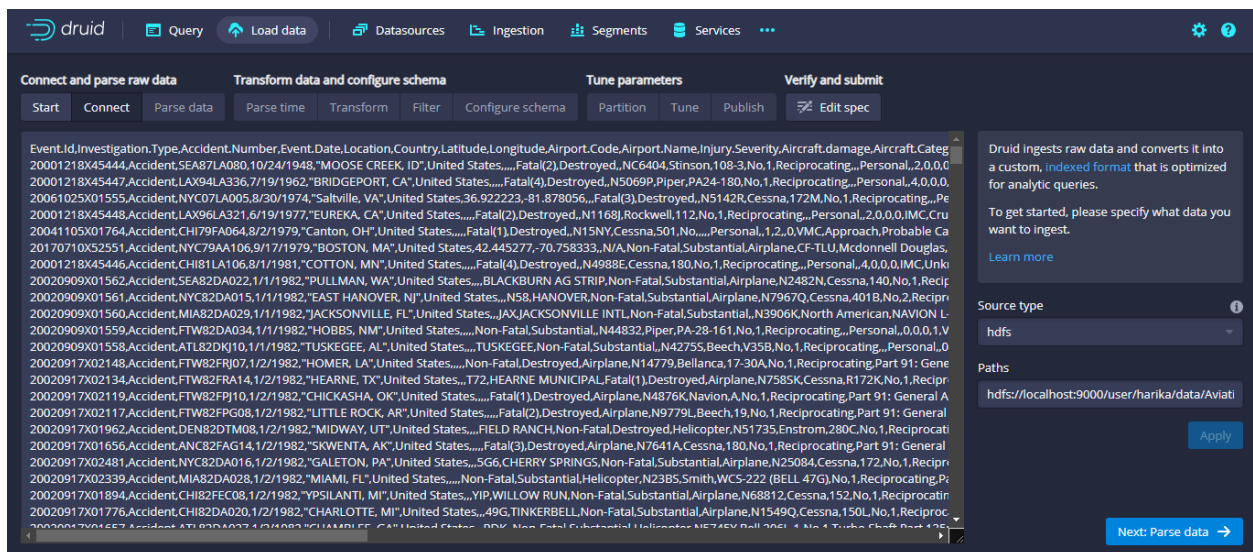
- Open the druid UI



- Load the data into druid from hdfs.



- Click on apply.



- Click on Parse data.

Druid requires flat data (non-nested, non-hierarchical). Each row should represent a discrete event. Ensure that your data appears correctly in a row/column orientation. [Learn more](#)

Event.Id	Investigation.Type	Accident.Number	Event.Date	Location	Country	Latitude	Location
20001218X45444	Accident	SEA87LA080	10/24/1948	MOOSE CREEK, ID	United States	null	nt
20001218X45447	Accident	LAX94LA336	7/19/1962	BRIDGEPORT, CA	United States	null	nt
20061025X01555	Accident	NYC07LA005	8/30/1974	Saltville, VA	United States	36.922223	-8
20001218X45448	Accident	LAX96LA321	6/19/1977	EUREKA, CA	United States	null	nt
20041105X01764	Accident	CHI79FA064	8/2/1979	Canton, OH	United States	null	nt
20170710X52551	Accident	NYC79AA106	9/17/1979	BOSTON, MA	United States	42.445277	-7
20001218X45446	Accident	CHI81LA106	8/1/1981	COTTON, MN	United States	null	nt
20020909X01562	Accident	SEA82DA022	1/1/1982	PULLMAN, WA	United States	null	nt

Showing 1-50 of 500

Next: Parse time →

- The data can be transformed as per the requirement.

Druid partitions data based on the primary time column of your data. This column is stored internally in Druid as `_time`. Configure how to define the time column for this data. If your data does not have a time column, you can select `None` to use a placeholder value. If the time information is spread across multiple columns you can combine them into one by selecting `expression` and defining a transform expression. [Learn more](#)

_time Column: Event.Date	Event.Id	Investigation.Type	Accident.Number	Event.Date M/d/yyyy	Location	Country
1970-01-01T00:00:00.000Z	20001218X45444	Accident	SEA87LA080	10/24/1948	MOOSE CREEK, ID	United States
1970-01-01T00:00:00.000Z	20001218X45447	Accident	LAX94LA336	7/19/1962	BRIDGEPORT, CA	United States
1970-01-01T00:00:00.000Z	20061025X01555	Accident	NYC07LA005	8/30/1974	Saltville, VA	United States
1970-01-01T00:00:00.000Z	20001218X45448	Accident	LAX96LA321	6/19/1977	EUREKA, CA	United States
1979-02-08T00:00:00.000Z	20041105X01764	Accident	CHI79FA064	8/2/1979	Canton, OH	United States
1970-01-01T00:00:00.000Z	20170710X52551	Accident	NYC79AA106	9/17/1979	BOSTON, MA	United States
1981-01-08T00:00:00.000Z	20001218X45446	Accident	CHI81LA106	8/1/1981	COTTON, MN	United States
1982-01-01T00:00:00.000Z	20020909X01562	Accident	SEA82DA022	1/1/1982	PULLMAN, WA	United States
1982-01-01T00:00:00.000Z	20020909X01561	Accident	NYC82DA015	1/1/1982	EAST HANOVER, NJ	United States
1982-01-01T00:00:00.000Z	20020909X01560	Accident	MIA82DA029	1/1/1982	JACKSONVILLE, FL	United States

Parse timestamp from: None Column Expression

Column: Event.Date

Next: Transform →

- Next below are the steps to be followed to install HIVE in ubuntu.
- Access your Ubuntu command line and download the compressed Hive files using and the wget command followed by the download path:
[wget https://downloads.apache.org/hive/hive-3.1.2/apache-hive-3.1.2-bin.tar.gz](https://downloads.apache.org/hive/hive-3.1.2/apache-hive-3.1.2-bin.tar.gz)
- Once the download process is complete, untar the compressed Hive package:
[tar xzf apache-hive-3.1.2-bin.tar.gz](#)

- The Hive binary files are now located in the apache-hive-3.1.2-bin directory.
- Configure Hive Environment Variables (bashrc)
- The \$HIVE_HOME environment variable needs to direct the client shell to the apache-hive-3.1.2-bin directory. Edit the .bashrc shell configuration file using a text editor of your choice (we will be using nano):

```
sudo nano .bashrc
```

- Append the following Hive environment variables to the .bashrc file:


```
export HIVE_HOME="home/hadoop/apache-hive-3.1.2-bin"
export PATH=$PATH:$HIVE_HOME/bin
```
- The Hadoop environment variables are located within the same file.
- Save and exit the .bashrc file once you add the Hive variables. Apply the changes to the current environment with the following command:

```
source ~/.bashrc
```

- Edit hive-config.sh file
- Apache Hive needs to be able to interact with the Hadoop Distributed File System. Access the hive-config.sh file using the previously created \$HIVE_HOME variable:

```
sudo nano $HIVE_HOME/bin/hive-config.sh
```

- Add the HADOOP_HOME variable and the full path to your Hadoop directory:


```
export HADOOP_HOME=/home/hadoop/hadoop-3.2.1
```
- Save the edits and exit the hive-config.sh file.
- Create Hive Directories in HDFS
- Create two separate directories to store data in the HDFS layer:
- The temporary, tmp directory is going to store the intermediate results of Hive processes.
- The warehouse directory is going to store the Hive related tables.
- Create tmp Directory.
- Create a tmp directory within the HDFS storage layer. This directory is going to store the intermediary data Hive sends to the HDFS:

```
hdfs dfs -mkdir /tmp
```

- Add write and execute permissions to tmp group members:

```
hdfs dfs -chmod g+w /tmp
```

- Check if the permissions were added correctly:

```
hdfs dfs -ls /
```

- The output confirms that users now have write and execute permissions.
- Create warehouse Directory
- Create the warehouse directory within the /user/hive/ parent directory:

```
hdfs dfs -mkdir -p /user/hive/warehouse
```

- Add write and execute permissions to warehouse group members:

```
hdfs dfs -chmod g+w /user/hive/warehouse
```

- Check if the permissions were added correctly:

```
hdfs dfs -ls /user/hive
```

- The output confirms that users now have write and execute permissions.
- Configure hive-site.xml File (Optional)
- Use the following command to locate the correct file:

```
cd $HIVE_HOME/conf
```

- List the files contained in the folder using the ls command.
- Create file hive.site.xml in conf folder
- Access the hive-site.xml file using the nano text editor:

```
sudo nano hive-site.xml
```

- Copy below lines of code in hive-site.xml file and save the file

```
<?xml version="1.0"?>
```

```
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
```

```
<configuration><property> <name>javax.jdo.option.ConnectionURL</name>
```

```
<value>jdbc:derby://localhost:1527/metastore_db;create=true</value>
```

```
<description>JDBC connect string for a JDBC metastore</description>
```

```
</property><property>
```

```
<name>javax.jdo.option.ConnectionDriverName</name>
```

```
<value>org.apache.derby.jdbc.ClientDriver</value>
```

```
<description>Driver class name for a JDBC metastore</description>
```

```
</property>
```

```
<property>
```

```
<name>hive.server2.enable.impersonation</name>
```

```
<description>Enable user impersonation for HiveServer2</description>
```

```
<value>>true</value>
```

```
</property>
```

```
<property>
```

```
<name>hive.server2.authentication</name>
```

```
<value>NONE</value>
```

```
<description> Client authentication types. NONE: no authentication check LDAP: LDAP/AD  
based authentication KERBEROS: Kerberos/GSSAPI authentication CUSTOM: Custom
```

```

authentication provider (Use with property hive.server2.custom.authentication.class)
</description>

</property>

<property>

<name>datanucleus.autoCreateTables</name>

<value>True</value>

</property>

</configuration>

```

- Using Hive in a stand-alone mode rather than in a real-life Apache Hadoop cluster is a safe option for newcomers. You can configure the system to use your local storage rather than the HDFS layer by setting the `hive.metastore.warehouse.dir` parameter value to the location of your Hive warehouse directory.
- Next Initiate Derby Database.
- Apache Hive uses the Derby database to store metadata. Initiate the Derby database, from the Hive bin directory using the schema tool command:
`$HIVE_HOME/bin/schematool -dbType derby -initSchema`
- The process can take a few moments to complete.
- Derby is the default metadata store for Hive. If you plan to use a different database solution, such as MySQL or PostgreSQL, you can specify a database type in the `hive-site.xml` file.
- How to Fix guava Incompatibility Error in Hive.
- Locate the guava jar file in the Hive lib directory:
`ls $HIVE_HOME/lib`
- Locate the guava jar file in the Hadoop lib directory as well:
`ls $HADOOP_HOME/share/hadoop/hdfs/lib`
- The two listed versions are not compatible and are causing the error. Remove the existing guava file from the Hive lib directory:
`rm $HIVE_HOME/lib/guava-19.0.jar`
- Copy the guava file from the Hadoop lib directory to the Hive lib directory:
`cp $HADOOP_HOME/share/hadoop/hdfs/lib/guava-27.0-jre.jar $HIVE_HOME/lib/`
- Use the schematool command once again to initiate the Derby database:
`$HIVE_HOME/bin/schematool -dbType derby -initSchema`
- Launch Hive Client Shell on Ubuntu
- Start the Hive command-line interface using the following commands:
`cd $HIVE_HOME/bin`

hive

- You are now able to issue SQL-like commands and directly interact with HDFS.

```
Initialization script completed
schemaTool completed
hadoop@ip-172-31-25-234:~/apache-hive-3.1.2-bin/conf$ cd $HIVE_HOME/bin
hadoop@ip-172-31-25-234:~/apache-hive-3.1.2-bin/bin$ hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/hadoop/apache-hive-3.1.2-bin/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/hadoop/hadoop/share/hadoop/common/lib/slf4j-reload4j-1.7.36.jar!/org/slf4j/impl/StaticLoggerBinder.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 = cfb86b41-e42f-4565-bde5-9bc05a690f85

Logging initialized using configuration in jar:file:/home/hadoop/apache-hive-3.1.2-bin/lib/hive-common-3.1.2.jar!/hive-log4j2.properties Async: true
Hive Session ID = 22b7b697-5ddf-4935-83c8-8e68211d0571
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. spark, tez) or
using Hive 1.X releases.
hive>
```

- Create database and create a table.

```
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/C:/Users/mokalidi/Downloads/apache-hive-2.1.0-bin/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/C:/Users/mokalidi/Downloads/hadoop-2.9.2/share/hadoop/common/lib/slf4j-log4j12.jar!/org/slf4j/impl/StaticLoggerBinder.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]
ERROR StatusLogger No log4j2 configuration file found. Using default configuration: logging only errors to the console
Connecting to jdbc:hive2://
Connected to: Apache Hive (version 2.1.0)
Driver: Hive JDBC (version 2.1.0)
Transaction isolation: TRANSACTION_REPEATABLE_READ
Beeline version 2.1.0 by Apache Hive
hive> create database mydatabase;
OK
No rows affected (1.343 seconds)
hive> show databases;
OK
default
mydatabase
2 rows selected (0.31 seconds)
hive>
```

```

hive> CREATE TABLE AVIATION(Event_Date VARCHAR(255),
> Location VARCHAR(255),
> Injury_Severity VARCHAR(255),
> Total_Fatal_Injuries VARCHAR(255),
> Total_Serious_Injuries VARCHAR(255),
> Total_Minor_Injuries VARCHAR(255),
> Total_Uninjured VARCHAR(255),
> Weather_Condition VARCHAR(255),
> Event_Id VARCHAR(255),
> Investigation_Type VARCHAR(255),
> Accident_Number VARCHAR(255),
> Country VARCHAR(255),
> Latitude VARCHAR(255),
> Longitude VARCHAR(255),
> Airport_Code VARCHAR(255),
> Airport_Name VARCHAR(255),
> Aircraft_damage VARCHAR(255),
> Aircraft_Category VARCHAR(255),
> Registration_Number VARCHAR(255),
> Make VARCHAR(255),
> Model VARCHAR(255),
> Amateur_Built VARCHAR(255),
> Number_of_Engines VARCHAR(255),
> Engine_Type VARCHAR(255),
> FAR_Description VARCHAR(255),
> Schedule VARCHAR(255),
> Purpose_of_flight VARCHAR(255),
> Air_carrier VARCHAR(255),
> Broad_phase_of_flight VARCHAR(255),
> Report_Status VARCHAR(255)
> );

```

```

OK
Time taken: 0.757 seconds
hive>

```

- Load data from HDFS to hive.

```

hive> LOAD DATA INPATH 'hdfs://localhost:9000/user/harika/data/AviationDataNew.csv' OVERWRITE INTO TABLE accidentData;
Loading data to table mydatabase.accidentdata
OK
Time taken: 0.65 seconds
hive>

```

- Perform query operations using the sql commands in hive and save the transformed data into csv and copy it to local/hdfs/s3.
- Use the below path to access hive data:
<hdfs://localhost:9000/user/hive/warehouse/mydatabase.db/accidentdata>

- **Note:** Mydatabase.db is the database name and accidentdata is the table name.
- Load the data into AWS Quick Sight for further analysis.

QuickSight

AviationDataNew

PUBLISH & VISUALIZE CANCEL

Fields All fields included

Focus All fields

Select All | None

Country

Accident_Number

Location

Aircraft_damage

Aircraft_Category

Weather_Condition

Publication_Date

Excluded fields No fields excluded

Query mode

SPICE

20GB of remaining

Data

AviationDataNew.csv

Dataset

Event_Id	Investigati...	Accident_...	Event_Date	Location	Country	Airport_Na...	Injury_Sev...	Aircraft_da...	Aircraft
20020917X...	Accident	FTW82DA079	1982-03-01...	NOACK, TX	United States		Non-Fatal	Substantial	Airpl
20020917X...	Accident	NYC82FGT08	1982-03-02...	SCOTLAND, PA	United States	ROCKTOP	Fatal(2)	Destroyed	Airpl
20020917X...	Accident	MKC82FCG17	1982-03-02...	TOPEKA, KS	United States	MESA VERDE	Fatal(1)	Destroyed	Airpl
20020917X...	Accident	NYC82DFJ11	1982-03-02...	LITTLE VALL...	United States		Non-Fatal	Substantial	Helic

- Perform the analysis on top of the table as per the requirement.

