# Write to HBase with Pyspark

## Step 1: Log in to Worker Nodes and Install the Library

1. Log in to Worker Node 1:  
docker-compose exec worker1 bash

2. Install the Library using pip3:  
pip3 install happybase

3. Exit Worker Node 1:  
exit

4. Log in to Worker Node 2:  
docker-compose exec worker2 bash

5. Install the Library using pip3:  
pip3 install happybase

6. Exit Worker Node 2:  
exit

## Step 2: Log in to the Master Node and Install the Library

1. Log in to the Master Node:  
docker-compose exec master bash

2. Install the Library using pip3:  
pip3 install happybase

# 2. Create the HBase Table

Open the HBase shell:  
  
hbase shell  
  
Create a table in HBase named my\_table with a column family cf:  
  
create 'my\_table', 'cf'

Exit the hbase shell

exit

# 3. Start the HBase Thrift Server

Start the HBase Thrift server to allow Python to connect via happybase. This command will be run in the master docker container.  
  
hbase thrift start &  
  
The & runs the Thrift server in the background.

# 4. Write Data to HBase Using PySpark

Open a Pyspark session and use the following code to write data to HBase:

import happybase  
from pyspark.sql import SparkSession  
  
# Initialize SparkSession  
spark = SparkSession.builder \  
 .appName('WriteToHBaseWithHappybase') \  
 .getOrCreate()  
  
# Example data (row\_key, column\_family:column, value)  
data = [('row1', 'cf:col1', 'value1'),  
 ('row2', 'cf:col1', 'value2')]  
  
# Function to write data to HBase inside each partition  
def write\_to\_hbase\_partition(partition):  
 connection = happybase.Connection('master')  
 connection.open()  
 table = connection.table('my\_table') # Update table name  
 for row in partition:  
 row\_key, column, value = row  
 table.put(row\_key, {column: value})  
 connection.close()  
  
# Parallelize data and apply the function with foreachPartition  
rdd = spark.sparkContext.parallelize(data)  
rdd.foreachPartition(write\_to\_hbase\_partition)  
  
# Stop the Spark session  
spark.stop()

# 5. Verify Data in HBase

After the script runs, you can verify the data written to HBase by using the HBase shell:  
  
hbase shell  
  
scan 'my\_table'