

Agenda:

- HW6 overview
- HBase setup EC2 AMI instance
- Hbase (and Spark) on EMR
- HBase Shell (CLI)
- Code examples

HBase Basics

Apache HBase Reference Guide:

https://hbase.apache.org/book.html#getting_started

HBase setup on EC2 AMI instance:

Highly recommended to avoid local Windows setup – use EC2 AMI from class

Match the EMR versions of spark (3.1.2) and HBase (2.4.4)

<https://dlcdn.apache.org/spark/spark-3.2.1/spark-3.2.1-bin-hadoop3.2.tgz>

```
tar xzvf spark-3.1.2-bin-hadoop3.2.tgz
```

```
export SPARK_HOME=/home/centos/software/spark-3.1.2-bin-hadoop3.2
```

```
export PATH=$SPARK_HOME/bin:$PATH
```

```
export PYTHONPATH=$SPARK_HOME/python:$SPARK_HOME/python/lib/py4j-0.10.9-src.zip
```

```
export PYSPARK_PYTHON=python3
```

Download Hbase 2.4.4

<https://archive.apache.org/dist/hbase/2.4.4/hbase-2.4.4-bin.tar.gz>

```
tar xzvf hbase-2.4.4-bin.tar.gz
```

```
edit hbase-2.4.4/conf/hbase-env.sh
```

and add the following line (note, the location is based on the class AMI on AWS)

```
export JAVA_HOME=/usr/java/jdk1.8.0_161
```

Next start up HBase:

```
/home/centos/hbase-2.4.4/bin/start-hbase.sh
```

Note: to stop HBase, use this:

```
<hbase_home>/bin/stop-hbase.sh
```

You can now start the shell (CLI):

```
/home/centos/hbase-2.4.4/bin/hbase shell
```

HBase Shell

Use "help" to get list of supported commands.

Use "exit" to quit this interactive shell.

For Reference, please visit: <http://hbase.apache.org/2.0/book.html#shell>

Version 2.4.4, rc49f7f63fca144765bf7c2da41791769286dfccc, Wed Nov 10 09:50:56 UTC 2021

Took 0.0016 seconds

```
hbase:001:0> version
```

2.4.4, rc49f7f63fca144765bf7c2da41791769286dfccc, Wed Nov 10 09:50:56 UTC 2021

Took 0.0003 seconds

Hbase (and Spark) on EMR

This will be slightly different than previous setups. Instead of using the basic setup screen, go into Advanced Options.

The screenshot shows the 'Create Cluster - Quick Options' page in the AWS EMR console. At the top, there is a link 'Go to advanced options' which is highlighted by an orange arrow pointing from a red box labeled 'Click Here'. Below this, the 'General Configuration' section includes a 'Cluster name' input field, a checked 'Logging' checkbox, an 'S3 folder' dropdown set to 's3://aws-logs-869062832896-us-east-2/elasticmap', and 'Launch mode' radio buttons for 'Cluster' (selected) and 'Step execution'. The 'Software configuration' section shows a 'Release' dropdown set to 'emr-5.34.0' and a list of 'Applications' with radio buttons. The first application, 'Core Hadoop: Hadoop 2.10.1, Hive 2.3.8, Hue 4.9.0, Mahout 0.13.0, Pig 0.17.0, and Tez 0.9.2', is selected. Other options include 'HBase: HBase 1.4.13, Hadoop 2.10.1, Hive 2.3.8, Hue 4.9.0, Phoenix 4.14.3, and ZooKeeper 3.4.14', 'Presto: Presto 0.261 with Hadoop 2.10.1 HDFS and Hive 2.3.8 Metastore', and 'Spark: Spark 2.4.8 on Hadoop 2.10.1 YARN and Zeppelin 0.10.0'. The footer contains 'Feedback', 'English (US)', '© 2022, Amazon Web Services, Inc. or its affiliates.', 'Privacy', 'Terms', and 'Cookie preferences'.

Select Hadoop, Spark, and HBase

Create Cluster - Advanced Options

[Go to quick options](#)

Step 1: Software and Steps
Step 2: Hardware
Step 3: General Cluster Settings
Step 4: Security

Software Configuration

Release emr-6.5.0

<input checked="" type="checkbox"/> Hadoop 3.2.1	<input type="checkbox"/> Zeppelin 0.10.0	<input type="checkbox"/> Livy 0.7.1
<input type="checkbox"/> JupyterHub 1.4.1	<input type="checkbox"/> Tez 0.9.2	<input type="checkbox"/> Flink 1.14.0
<input type="checkbox"/> Ganglia 3.7.2	<input checked="" type="checkbox"/> HBase 2.4.4	<input type="checkbox"/> Pig 0.17.0
<input type="checkbox"/> Hive 3.1.2	<input type="checkbox"/> Presto 0.261	<input type="checkbox"/> ZooKeeper 3.5.7
<input type="checkbox"/> JupyterEnterpriseGateway 2.1.0	<input type="checkbox"/> MXNet 1.8.0	<input type="checkbox"/> Sqoop 1.4.7
<input type="checkbox"/> Hue 4.9.0	<input type="checkbox"/> Phoenix 5.1.2	<input type="checkbox"/> Trino 360
<input type="checkbox"/> Oozie 5.2.1	<input checked="" type="checkbox"/> Spark 3.1.2	<input type="checkbox"/> HCatalog 3.1.2
<input type="checkbox"/> TensorFlow 2.4.1		

Multiple master nodes (optional)
☐ Use multiple master nodes to improve cluster availability. [Learn more](#)

AWS Glue Data Catalog settings (optional)

[Feedback](#) [English \(US\)](#) © 2022, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Click **Next**

On the next screen, you can adjust the Cluster Nodes to Spot instances instead of On-demand to help save some charges.

aws

Services

Greg Forrest

Ohio

Support

Cluster Nodes and Instances

Choose the instance type, number of instances, and a purchasing option. [Learn more about instance purchasing options](#)

Console options for automatic scaling have changed. [Learn more](#)

Node type	Instance type	Instance count	Purchasing option
Master Master - 1	m5.xlarge 4 vCore, 16 GiB memory, EBS only storage EBS Storage: 32 GiB Add configuration settings	1 Instances	<input type="radio"/> On-demand <input checked="" type="radio"/> Spot Use on-demand as max price
Core Core - 2	m5.xlarge 4 vCore, 16 GiB memory, EBS only storage EBS Storage: 32 GiB Add configuration settings	2 Instances	<input type="radio"/> On-demand <input checked="" type="radio"/> Spot Use on-demand as max price
Task Task - 3	m5.xlarge 4 vCore, 16 GiB memory, EBS only storage EBS Storage: 32 GiB Add configuration settings	0 Instances	<input type="radio"/> On-demand <input checked="" type="radio"/> Spot Use on-demand as max price

Feedback

English (US)

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Privacy Policy

Terms of Use

Click **Next**

Give your cluster a name and **deselect Termination protection**

aws Services ▾

Greg Forrest ▾ Ohio ▾ Support ▾

Create Cluster - Advanced Options [Go to quick options](#)

Step 1: Software and Steps
Step 2: Hardware
Step 3: General Cluster Settings
Step 4: Security

General Options

Cluster name

☒ Logging ⓘ

S3 folder

☐ Log encryption ⓘ

☒ Debugging ⓘ

☐ Termination protection ⓘ

Tags ⓘ

Key	Value (optional)
<input type="text" value="Add a key to create a tag"/>	<input type="text"/>

Feedback English (US) ▾

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of](#)

Click **Next**

My Drive - Google

Inbox (10) - Gmail

Piazza - Ask, Answer

CS: E-88 (1541)

EMR - AWS Console

HBase shell console

PySpark - Environment

us-east-2.console.aws.amazon.com/elasticmapreduce/home?region=us-east-2#create-cluster

AppsLogin - Oracle Enterprise...OEM 13cEnterprise eTIME®prod Cloudera Managerdev Cloudera ManagerGranite Intranetreset eis passwordOther bookmarks

awsServicesGreg ForrestOhioSupport

Step 1: Software and Steps

Step 2: Hardware

Step 3: General Cluster Settings

Step 4: Security

Security Options

EC2 key pairgreg-e88

☒ Cluster visible to all IAM users in account

Permissions

☒ Default☐ Custom

Use default IAM roles. If roles are not present, they will be automatically created for you with managed policies for automatic policy updates.

EMR roleEMR_DefaultRole

EC2 instance profileEMR_EC2_DefaultRole

Auto Scaling roleEMR_AutoScaling_DefaultRole

Security Configuration

EC2 security groups

Cancel

Previous

Create cluster

Basic CLI Commands

Here is a nice summary:

<https://learnhbase.net/2013/03/02/hbase-shell-commands/>

Create and review table

create '<table_name>', {NAME => '<column_family1>'}, {Name => '<column_family2>'}

or simply:

create '<table_name>', '<column_family1>' [, '<column_family2>', ... '<column_familyN>']

Example:

```
hbase(main):002:0> create 'CallLog', 'callDetails', 'callMetrics'
```

```
0 row(s) in 1.5540 seconds
```

```
=> Hbase::Table - CallLog
```

describe '<table_name>'

```
hbase(main):004:0> describe 'CallLog'
```

```
Table CallLog is ENABLED
```

```
CallLog
```

```
COLUMN FAMILIES DESCRIPTION
```

```
{NAME => 'callDetails', BLOOMFILTER => 'ROW', VERSIONS => '1', IN_MEMORY => 'false', KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREV
```

```
ER', COMPRESSION => 'NONE', MIN_VERSIONS => '0', BLOCKCACHE => 'true',
```

```
BLOCKSIZE
```

```
=> '65536', REPLICATION_SCOPE => '0'}
```

```
{NAME => 'callMetrics', BLOOMFILTER => 'ROW', VERSIONS => '1', IN_MEMORY => 'false', KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREV
```

```
ER', COMPRESSION => 'NONE', MIN_VERSIONS => '0', BLOCKCACHE => 'true',
```

```
BLOCKSIZE
```

```
=> '65536', REPLICATION_SCOPE => '0'}
```

```
2 row(s) in 0.0480 seconds
```

Note the 2 rows – one for each column family in the table.

list

```
hbase(main):005:0> list
```

```
TABLE
```

```
CallLog
```

```
1 row(s) in 0.0190 seconds
```


=> ["CallLog"]

exists '<table_name>'

hbase(main):006:0> exists 'CallLog'

Table CallLog does exist

0 row(s) in 0.0160 seconds

Put and get data

put '<table_name>', '<row_key>', '<column_family>:<column>', '<value>'

hbase(main):008:0> put 'CallLog','20220303080000','callDetails:from','6171234567'

Took 0.1645 seconds

get '<table_name>', '<row_key>'

hbase:006:0> get 'CallLog', '20220303080000'

COLUMN CELL

callDetails:from timestamp=2022-03-02T20:58:55.549, value=6171234567

1 row(s)

Took 0.0713 seconds

scan '<table_name>'

hbase:007:0> scan 'CallLog'

ROW COLUMN+CELL

*20220303080000 column=callDetails:from, timestamp=2022-03-02T20:58:55.549
, value=6171234567*

1 row(s)

Took 0.0528 seconds

Versions:

alter '<table_name>', {NAME => '<column_family>', VERSIONS => <N> }

alter 'CallLog', { NAME => 'callDetails', VERSIONS => 3 }

Demo:

put 'CallLog', '20220303080000', 'callDetails:StillThere','Yes1'

put 'CallLog', '20220303080000', 'callDetails:StillThere','Yes2'

put 'CallLog', '20220303080000', 'callDetails:StillThere','Yes3'

put 'CallLog', '20220303080000', 'callDetails:StillThere','Yes4'

put 'CallLog', '20220303080000', 'callDetails:StillThere','No'

get 'CallLog', '20220303080000', {COLUMN => 'callDetails:StillThere', VERSIONS => 2}

COLUMN CELL

callDetails:StillThe timestamp=2022-03-02T21:04:39.118, value=No

re

*callDetails:StillThe timestamp=2022-03-02T21:04:39.072, value=Yes4
re
1 row(s)
Took 0.0237 seconds*

Dropping Tables

1st disable, then drop

disable '<table_name>'
drop '<table_name>'

create namespace '<name_space>'

Create a namespace

create_namespace 'lab'
create_namespace, alter_namespace, describe_namespace, drop_namespace,
list_namespace, list_namespace_tables

Scripting HBase CLI Commands

You can also package commands into a file and run them as a script:

`./bin/hbase shell <file_containing_commands>`

lab_setup.txt:

create_namespace 'lab'

create 'lab:date_hour', 'url'

describe 'lab:date_hour'

exit

`CMD> ./bin/hbase shell ./lab_setup.txt`

0 row(s) in 1.2090 seconds

0 row(s) in 1.3890 seconds

Table lab:date_hour is ENABLED

lab:date_hour

COLUMN FAMILIES DESCRIPTION

```
{NAME => 'url', BLOOMFILTER => 'ROW', VERSIONS => '1', IN_MEMORY => 'false', KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER', COMPRESSION => 'NONE', MIN_VERSIONS => '0', BLOCKCACHE => 'true', BLOCKSIZE => '65536', REPLICATION_SCOPE => '0'}
```

1 row(s) in 0.0410 seconds

Code Options

Python

Happybase

Happybase python library to connect to HBase via the Thrift API

<https://happybase.readthedocs.io/en/latest/>

To install, ensure python3-devel is installed and then use pip3 install:

```
sudo yum install python3-devel  
sudo pip3 install happybase
```

For local testing, you need to start the thrift service:

```
<HBASE_HOME>/bin/hbase-daemon.sh start thrift
```

The thrift service should already be running on the EMR cluster.

```
spark-submit --packages org.apache.spark:spark-avro_2.12:3.1.2 --master yarn  
HoursCounterSparkHbaseHappy.py file-input-avro
```

Hortonworks Spark-Hbase Connector (SHC)

<https://github.com/hortonworks-spark/shc>

Supports Spark accessing HBase using Spark SQL dataframes

```
spark-submit --packages org.apache.spark:spark-avro_2.12:3.1.2,com.hortonworks:shc-  
core:1.1.1-2.1-s_2.11 --repositories http://repo.hortonworks.com/content/groups/public/ --files  
/etc/hbase/conf/hbase-site.xml HoursCounterSparkHbaseSHC.py
```

Java

```
// https://mvnrepository.com/artifact/org.apache.spark/spark-sql  
compile group: 'org.apache.hbase', name: 'hbase', version: '2.4.4'
```

```
// https://mvnrepository.com/artifact/org.apache.spark/spark-sql  
compile group: 'org.apache.hbase', name: 'hbase-client', version: '2.4.4'
```

To add hbase libraries into the spark classpath - update `spark.driver.extraClassPath` and `spark.executor.extraClassPath` in `spark-defaults.conf`

```
sudo vim /etc/spark/conf/spark-defaults.conf
```

```
:/usr/lib/hbase/*:/usr/lib/hadoop/hadoop-aws.jar:/usr/lib/hbase/lib/htrace-core-3.1.0-incubating.jar  
:/usr/lib/hbase/lib/metrics-core-2.2.0.jar
```

```
spark-submit --class package cscie88.spring2022.week6.HoursCounterSparkJobAvroHBase  
lab6-0.0.1-SNAPSHOT.jar input
```