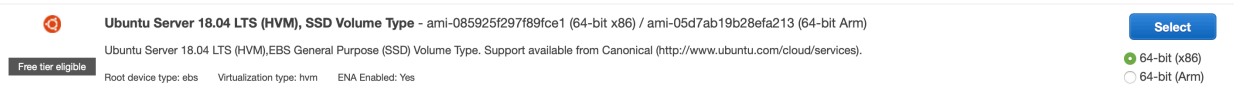
Brian Davis
tug75085
915409443
2 May 2020

Final Exam

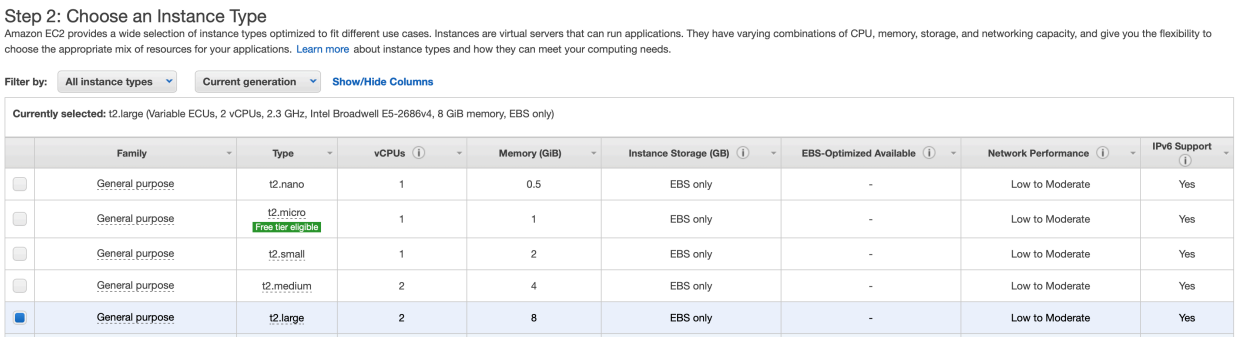
Part 1: MapReduce

EC2 Creation

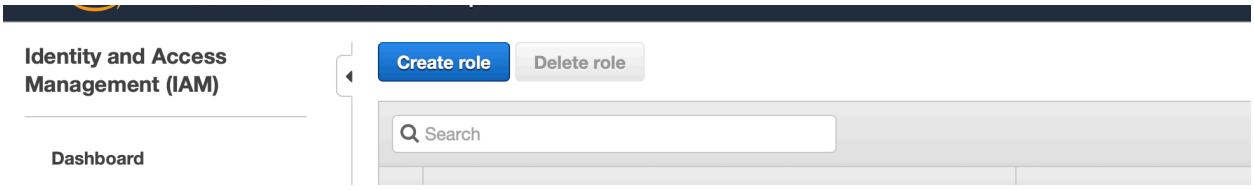
First, I chose the AMI type Ubuntu Server because I was able to find success setting up Hadoop on a local virtual machine running Ubuntu previously. The AMI type is seen below,



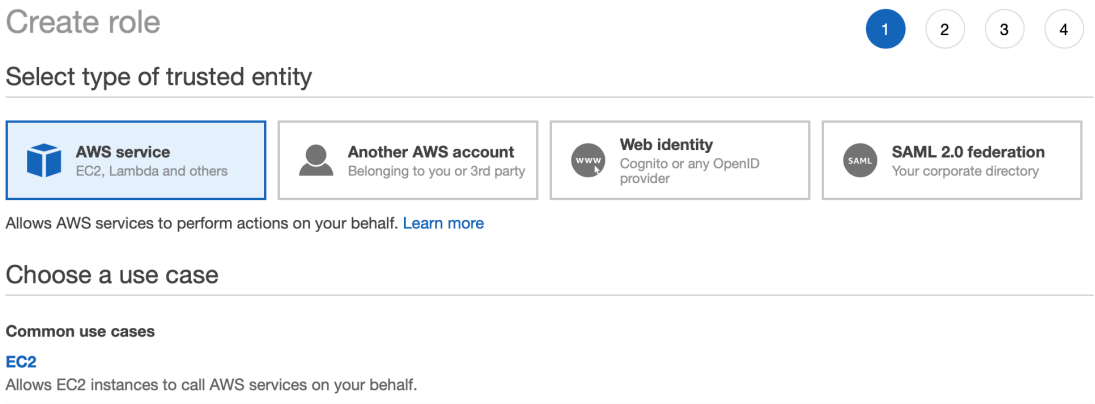
Next, I chose the instance type t2.large, which has 8 GiB of memory, in comparison to 2 GiB on the t2.micro. This is necessary in order to ensure Hadoop and MapReduce can run smoothly, as I had limited memory issues on my local environment previously. The instance type is seen below,



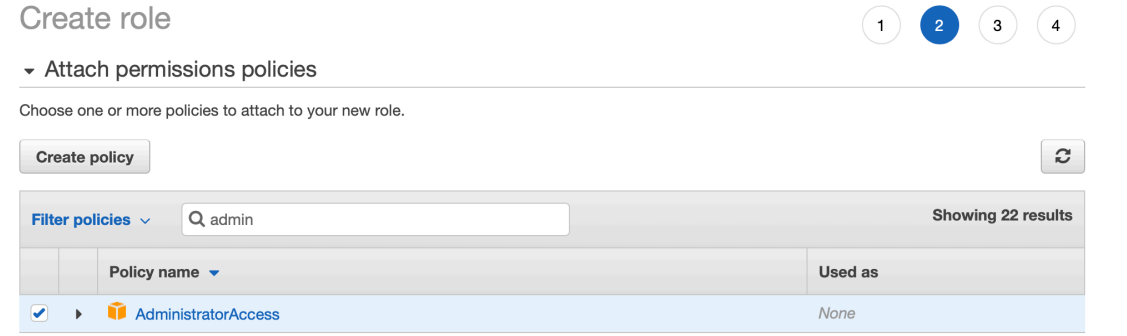
Next I created an IAM Role that allows Administrator Access to all AWS services. First I created the role, as seen below,

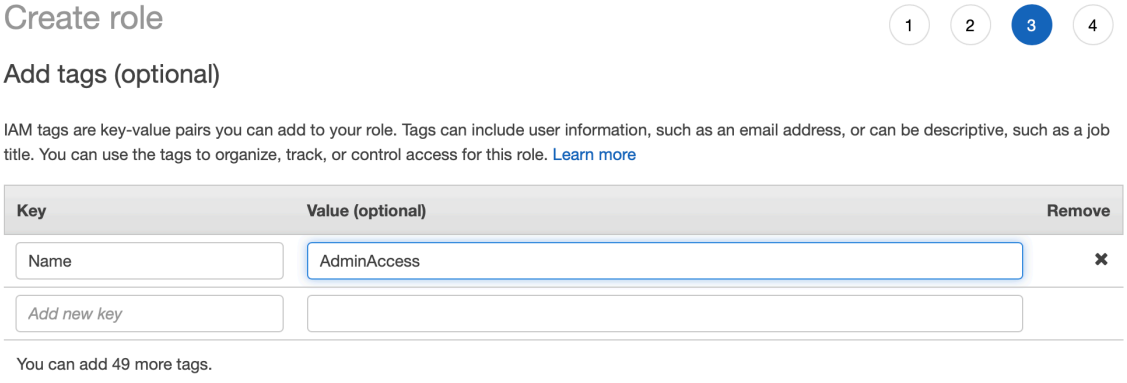


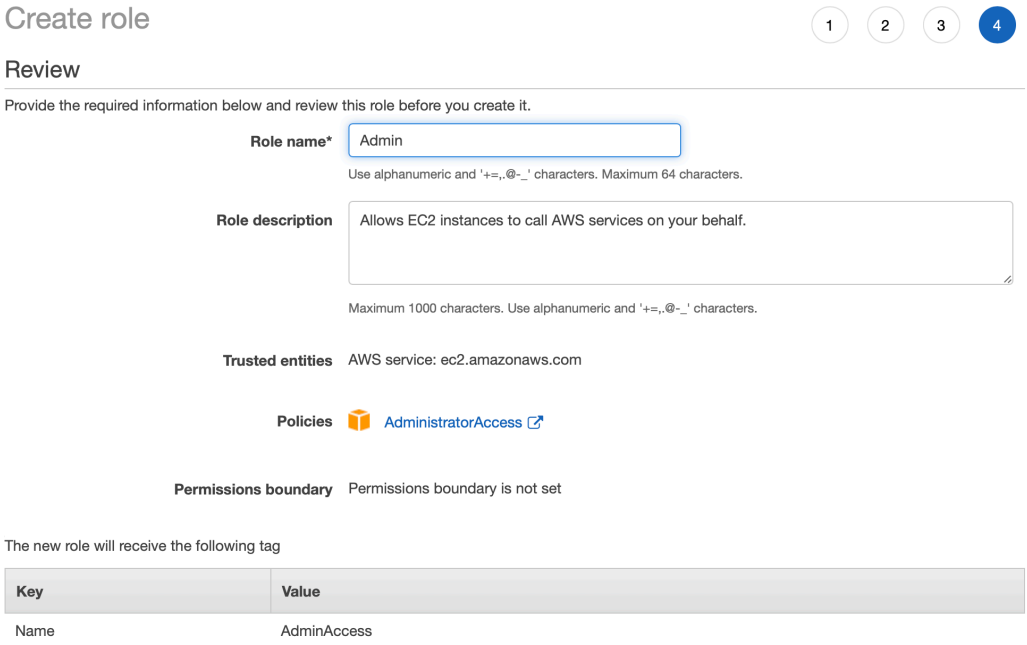
Then I selected AWS the IAM role to correspond with all AWS services.



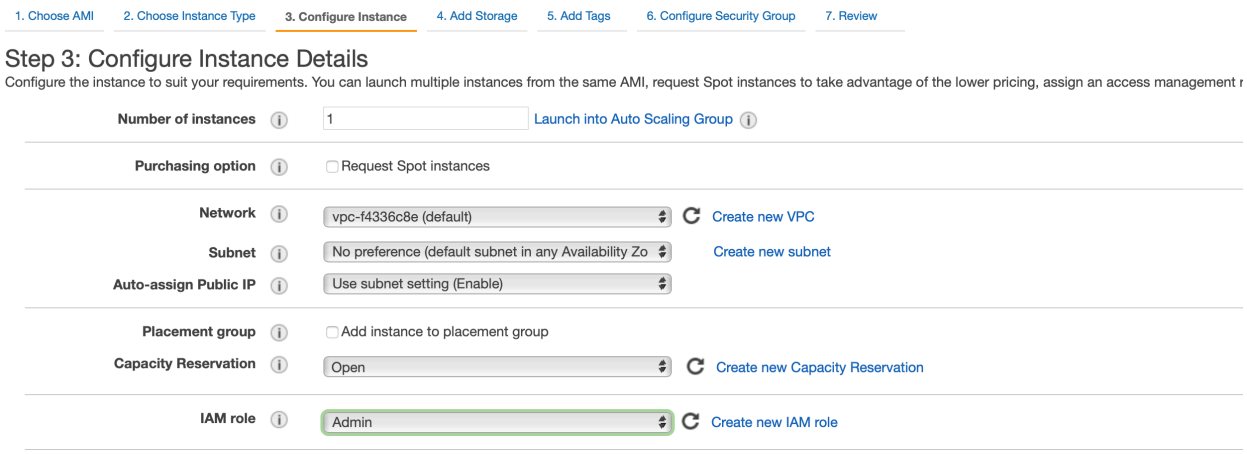
I created and named the role Administrator Access.



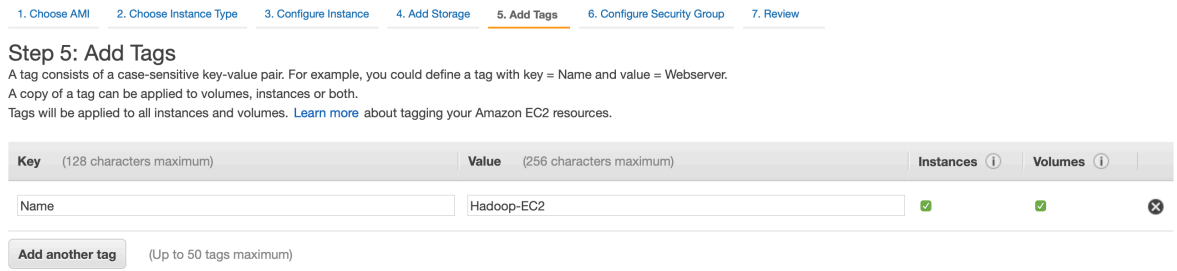




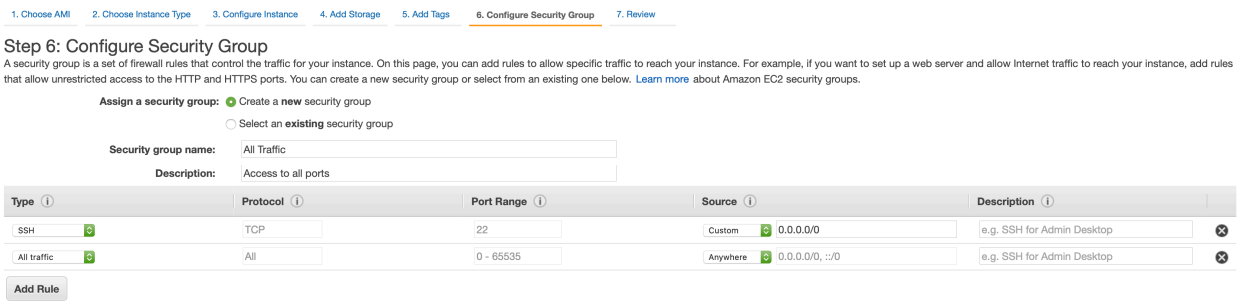
Next I selected the IAM role as a field in the Configure Instance Details of the EC2 Creation options,



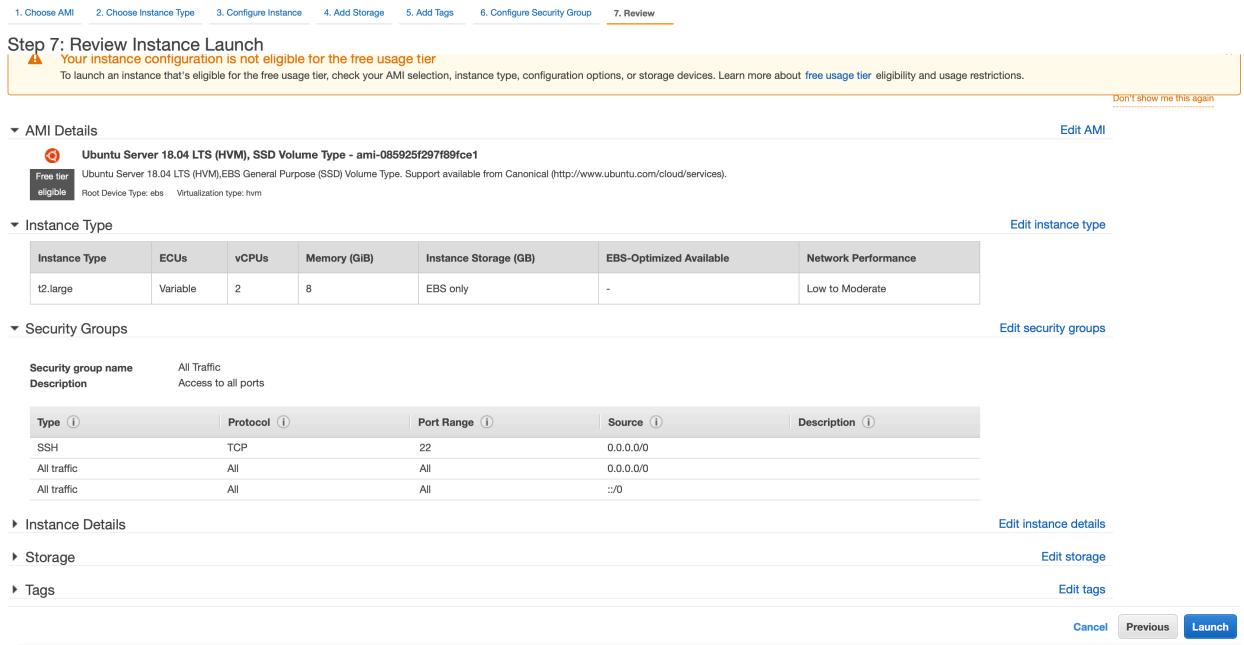
Next I gave the instance an identifier tag,



In the Configure Security Group tab of the EC2 creation, I selected all ICMP and TCP ports so that the Hadoop webserver can work properly with limited flaws, as seen below,



After setting all of these options, I created the EC2!



Setting up Hadoop on Ubuntu EC2 Instance

Now that we have our Ubuntu EC2 running, we can SSH into it,

```
briandavis@Brians-MacBook-Pro Desktop % ssh -i "tu.pem" ubuntu@ec2-3-90-162-201.compute-1.amazonaws.com
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-1065-aws x86_64)

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

  System information as of Thu Apr 30 00:53:37 UTC 2020

  System load:  0.02              Processes:           112
  Usage of /:   13.7% of 7.69GB   Users logged in:     0
  Memory usage: 2%                IP address for eth0: 172.31.24.19
  Swap usage:   0%

0 packages can be updated.
0 updates are security updates.



The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-24-19:~$
```

First, we must install Java and Hadoop. We gather the tar files and untar them as seen below,

```
root@ip-172-31-24-19:/home/ubuntu# ls
hadoop-2.7.3  hadoop-2.7.3.tar.gz  jdk-8u131-linux-x64.tar.gz  jdk1.8.0_131
root@ip-172-31-24-19:/home/ubuntu#
```

First, we have to set environment variables in the ~/.bashrc, as seen below,

```
export HADOOP_HOME=/home/ubuntu/hadoop-2.7.3
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib"
export JAVA_HOME=/home/ubuntu/jdk1.8.0_131
PATH=$PATH:$JAVA_HOME/bin:$HADOOP_HOME/bin:$HADOOP_HOME/sbin
```

Next check that the Java and Hadoop libraries can be found,

```
root@ip-172-31-24-19:/home/ubuntu# java -version
java version "1.8.0_131"
Java(TM) SE Runtime Environment (build 1.8.0_131-b11)
Java HotSpot(TM) 64-Bit Server VM (build 25.131-b11, mixed mode)
root@ip-172-31-24-19:/home/ubuntu# hadoop version
Hadoop 2.7.3
Subversion https://git-wip-us.apache.org/repos/asf/hadoop.git -r baa91f7c6bc9cb92be5982de4719c1c8af91ccff
Compiled by root on 2016-08-18T01:41Z
Compiled with protoc 2.5.0
From source with checksum 2e4ce5f957ea4db193bce3734ff29ff4
This command was run using /home/ubuntu/hadoop-2.7.3/share/hadoop/common/hadoop-common-2.7.3.jar
root@ip-172-31-24-19:/home/ubuntu#
```

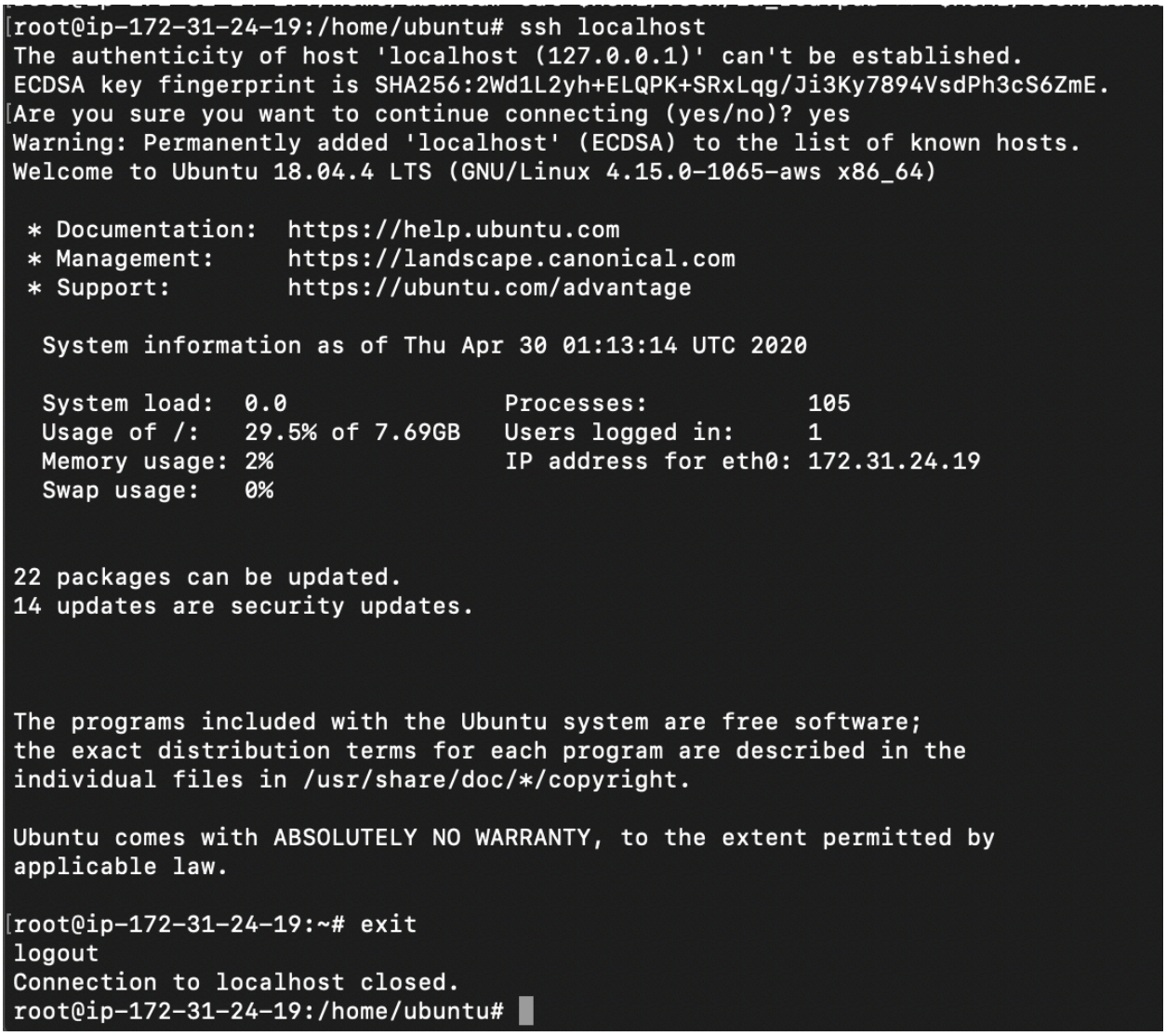
Next we activate the environment,

```
root@ip-172-31-24-19:/home/ubuntu# source ~/.bashrc
```

Then we ensure that ssh permissions for localhost are authorized,

```
root@ip-172-31-24-19:/home/ubuntu# ssh-keygen -t rsa -P ''
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:cIqX31MAuA0GIwXcSrHgfFH0bpDzlZYwRT5S9fpCnzQ root@ip-172-31-24-19
The key's randomart image is:
+---[RSA 2048]----+
|..+**+ ++..      |
|o.ooo.* =.o .    |
| +.o =.*.B.  .   |
|  o  .*== ...    |
|     . ++S  o.E  |
|      ... ...+ o |
|         . o. +  |
|            ..   |
|                 |
+----[SHA256]-----+
root@ip-172-31-24-19:/home/ubuntu# cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys
```

Example of ssh-ing into localhost and subsequently exiting,



Next we must make the following modifications to the Hadoop XML and sh files in order to successfully execute tasks on the Hadoop file system.

Core-site.xml

```
<configuration>
<property>
<name>hadoop.tmp.dir</name>
<value>/home/ubuntu/hadooptmp/hadoop-${user.name}</value>
<description>A base for other temporary directories.</description>
</property>
<property>
<name>fs.default.name</name>
<value>hdfs://localhost:9000</value>
</property>
</configuration>
```

Hadoop-env.sh

```
# The java implementation to use.
export JAVA_HOME=/home/ubuntu/jdk1.8.0_131
```

Mapred-site.xml

```
<configuration>
<property>
<name>mapred.job.tracker</name>
<value>localhost:9001</value>
</property>
</configuration>
```

Yarn-site.xml

```
<!-- Site specific YARN configuration properties -->
<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>
</configuration>
```

Hdfs-site.xml

```
<configuration>
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
<property><name>dfs.name.dir</name>
<value>file:///home/ubuntu/hadoopdata/hdfs/namenode</value>
</property>
<property>
<name>dfs.data.dir</name>
<value>file:///home/ubuntu/hadoopdata/hdfs/datanode</value>
</property>
</configuration>
```

After editing all of the files (as seen finished below), we can now move on to working on Hadoop,

```
root@ip-172-31-24-19:/home/ubuntu/hadoop-2.7.3/etc/hadoop# vi core-site.xml
root@ip-172-31-24-19:/home/ubuntu/hadoop-2.7.3/etc/hadoop# readlink -f `which java`
/home/ubuntu/jdk1.8.0_131/bin/java
root@ip-172-31-24-19:/home/ubuntu/hadoop-2.7.3/etc/hadoop# vi hadoop-env.sh
root@ip-172-31-24-19:/home/ubuntu/hadoop-2.7.3/etc/hadoop# vi hadoop-env.sh
root@ip-172-31-24-19:/home/ubuntu/hadoop-2.7.3/etc/hadoop# vi hadoop-env.sh
root@ip-172-31-24-19:/home/ubuntu/hadoop-2.7.3/etc/hadoop# cp mapred-site.xml.template mapred-site.xml
root@ip-172-31-24-19:/home/ubuntu/hadoop-2.7.3/etc/hadoop# vi mapred-site.xml
root@ip-172-31-24-19:/home/ubuntu/hadoop-2.7.3/etc/hadoop# pwd
/home/ubuntu/hadoop-2.7.3/etc/hadoop
root@ip-172-31-24-19:/home/ubuntu/hadoop-2.7.3/etc/hadoop# vi yarn-site.xml
root@ip-172-31-24-19:/home/ubuntu/hadoop-2.7.3/etc/hadoop# vi hdfs-site.xml
root@ip-172-31-24-19:/home/ubuntu/hadoop-2.7.3/etc/hadoop#
```

First, we format the Hadoop filesystem,

```
root@ip-172-31-24-19:/home/ubuntu# hdfs namenode -format
20/04/30 01:39:27 INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = ip-172-31-24-19.ec2.internal/172.31.24.19
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 2.7.3
STARTUP_MSG:   classpath = /home/ubuntu/hadoop-2.7.3/etc/hadoop:/home/ub
f4j-api-1.7.10.jar:/home/ubuntu/hadoop-2.7.3/share/hadoop/common/lib/com
3/share/hadoop/common/lib/jsch-0.1.42.jar:/home/ubuntu/hadoop-2.7.3/shar
```

Starting the NameNode, DataNode, SecondaryNameNode,

```
root@ip-172-31-24-19:/home/ubuntu/hadoop-2.7.3/sbin# ./start-dfs.sh
20/04/30 01:40:57 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Starting namenodes on [localhost]
localhost: starting namenode, logging to /home/ubuntu/hadoop-2.7.3/logs/hadoop-root-namenode-ip-172-31-24-19.out
localhost: starting datanode, logging to /home/ubuntu/hadoop-2.7.3/logs/hadoop-root-datanode-ip-172-31-24-19.out
Starting secondary namenodes [0.0.0.0]
The authenticity of host '0.0.0.0 (0.0.0.0)' can't be established.
ECDSA key fingerprint is SHA256:2Wd1L2yh+ELQPK+SRxLqg/Ji3Ky7894VsdPh3cS6ZmE.
Are you sure you want to continue connecting (yes/no)? yes
0.0.0.0: Warning: Permanently added '0.0.0.0' (ECDSA) to the list of known hosts.
0.0.0.0: starting secondarynamenode, logging to /home/ubuntu/hadoop-2.7.3/logs/hadoop-root-secondarynamenode-ip-172-31-24-19.out
20/04/30 01:41:16 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
root@ip-172-31-24-19:/home/ubuntu/hadoop-2.7.3/sbin#
```

Starting the ResourceManager and NodeManager,

```
root@ip-172-31-24-19:/home/ubuntu/hadoop-2.7.3/sbin# ./start-yarn.sh
starting yarn daemons
starting resourcemanager, logging to /home/ubuntu/hadoop-2.7.3/logs/yarn-root-resourcemanager-ip-172-31-24-19.out
localhost: starting nodemanager, logging to /home/ubuntu/hadoop-2.7.3/logs/yarn-root-nodemanager-ip-172-31-24-19.out
root@ip-172-31-24-19:/home/ubuntu/hadoop-2.7.3/sbin#
```

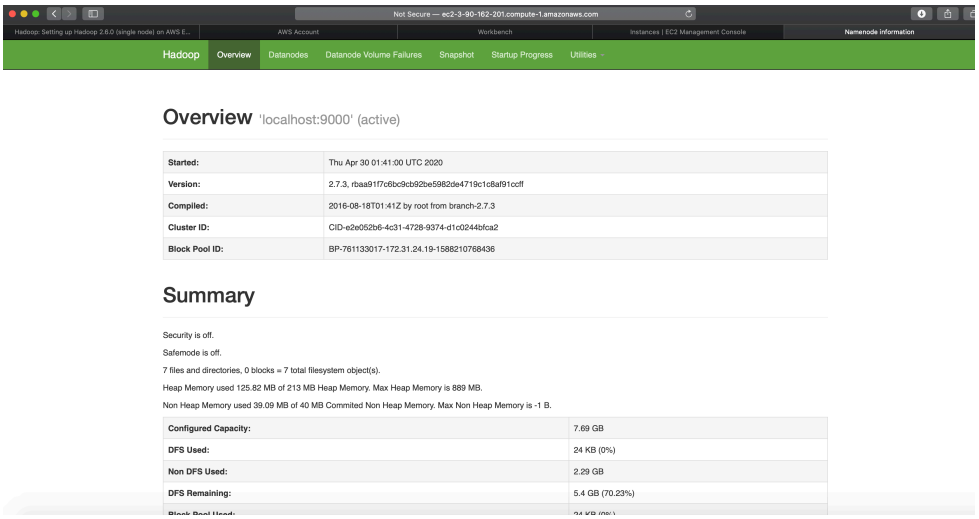
Starting the Mr-JobHistoryServer,

```
root@ip-172-31-24-19:/home/ubuntu/hadoop-2.7.3/sbin# mr-jobhistory-daemon.sh start historyserver
starting historyserver, logging to /home/ubuntu/hadoop-2.7.3/logs/mapred-root-historyserver-ip-172-31-24-19.out
```

Checking to see if all of the resources are activated,

```
root@ip-172-31-24-19:/home/ubuntu/hadoop-2.7.3/sbin# jps
2260 ResourceManager
1892 DataNode
2743 JobHistoryServer
2556 NodeManager
2109 SecondaryNameNode
2781 Jps
1711 NameNode
root@ip-172-31-24-19:/home/ubuntu/hadoop-2.7.3/sbin#
```

Viewing the website that displays the Hadoop file system



Making the input and output directories on the Hadoop file system,

```
root@ip-172-31-24-19:/home/ubuntu# hdfs dfs -mkdir /input
20/04/30 02:22:05 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
root@ip-172-31-24-19:/home/ubuntu# hdfs dfs -mkdir /output
20/04/30 02:22:11 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
```

Copying the *Pride and Prejudice* text file to the input directory on the Hadoop file system,

```
root@ip-172-31-24-19:/home/ubuntu# hdfs dfs -copyFromLocal PrideAndPrejudice.txt /input
20/04/30 02:31:55 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
```

MapReduce Modifications

The WordCount.java class supplied by Apache Hadoop was modified to meet the needs of a bigram counter, namely, BigramCount.java. The Mapper of the BigramCount.java script below stores the previous and current String tokens in private globals, which is important since Hadoop is a distributed filesystem. As the list of input tokens are looped through, the previous and current String tokens are concatenated to form bigrams, which are in turn collected and mapped to the reducers.

```
public static class Map extends MapReduceBase implements Mapper<LongWritable, Text, Text, IntWritable> {
  private final static IntWritable one = new IntWritable(1);
  private Text bigram = new Text();
  // To construct the bigrams, we store previous and current words
  private String prevToken = null;
  private String currToken = null;

  public void map(LongWritable key, Text value, OutputCollector<Text, IntWritable> output, Reporter reporter) throws IOException {
    String line = value.toString();
    StringTokenizer tokenizer = new StringTokenizer(line);
    // Loop through all of the words
    while (tokenizer.hasMoreTokens()) {
      // Get the current word
      currToken = tokenizer.nextToken();
      // Only construct bigrams if the current word has a previous word
      if (prevToken != null) {
        // Combine the previous word with the current word to construct a bigram
        bigram.set(prevToken + " " + currToken);
        output.collect(bigram, one);
      }
      // Update the previous word and restart loop
      prevToken = currToken;
    }
  }
}
```

The Reducer supplied by Apache Hadoop did not need to be modified for this assignment,

```
public static class Reduce extends MapReduceBase implements Reducer<Text, IntWritable, Text, IntWritable> {
  public void reduce(Text key, Iterator<IntWritable> values, OutputCollector<Text, IntWritable> output, Reporter reporter) throws IOException {
    int sum = 0;
    while (values.hasNext()) {
      sum += values.next().get();
    }
    output.collect(key, new IntWritable(sum));
  }
}
```

In addition, I set the number of reducers to 5 programmatically for our bigram counter, as seen below,

```
  public static void main(String[] args) throws Exception {
    JobConf conf = new JobConf(WordCount.class);
    conf.setJobName("bigramcount");

    conf.setOutputKeyClass(Text.class);
    conf.setOutputValueClass(IntWritable.class);

    conf.setMapperClass(Map.class);
    conf.setCombinerClass(Reduce.class);
    conf.setReducerClass(Reduce.class);

    conf.setInputFormat(TextInputFormat.class);
    conf.setOutputFormat(TextOutputFormat.class);

    // Set the number of reducer tasks to 5
    conf.setNumReduceTasks(5);

    FileInputFormat.setInputPaths(conf, new Path(args[0]));
    FileOutputFormat.setOutputPath(conf, new Path(args[1]));

    JobClient.runJob(conf);
  }
}
```

The main portion of this program is counting the bigrams from the *Pride and Prejudice* text file. In order to do this, we need to first compile out MapReduce program so that it can be run on Hadoop. Below, we can see the compilation of the BigramCount.java MapReduce program into class files in the BigramCountClasses directory,

```
root@ip-172-31-24-19:/home/ubuntu/BigramCount# javac -classpath /home/ubuntu/hadoop-2.7.3/etc/hadoop:/home/ubuntu/hadoop-2.7.3/share/hadoop/common/lib/*:/home/ubuntu/h
adoop-2.7.3/share/hadoop/common/*:/home/ubuntu/hadoop-2.7.3/share/hadoop/hdfs:/home/ubuntu/hadoop-2.7.3/share/hadoop/hdfs/lib/*:/home/ubuntu/hadoop-2.7.3/share/hadoop/
hdfs/*:/home/ubuntu/hadoop-2.7.3/share/hadoop/yarn/lib/*:/home/ubuntu/hadoop-2.7.3/share/hadoop/yarn/*:/home/ubuntu/hadoop-2.7.3/share/hadoop/mapreduce/lib/*:/home/ubu
ntu/hadoop-2.7.3/share/hadoop/mapreduce/*:/home/ubuntu/hadoop-2.7.3/contrib/capacity-scheduler/*.jar -d BigramCountClasses/ BigramCount.java
```

Then we compile the BigramCount classes into BigramCount.jar, all seen below

```
root@ip-172-31-24-19:/home/ubuntu/BigramCount# jar -cvf BigramCount.jar -C BigramCountClasses/ .
added manifest
adding: BigramCount$Map.class(in = 2103) (out= 898)(deflated 57%)
adding: BigramCount.class(in = 1598) (out= 785)(deflated 50%)
adding: BigramCount$Reduce.class(in = 1597) (out= 643)(deflated 59%)
```

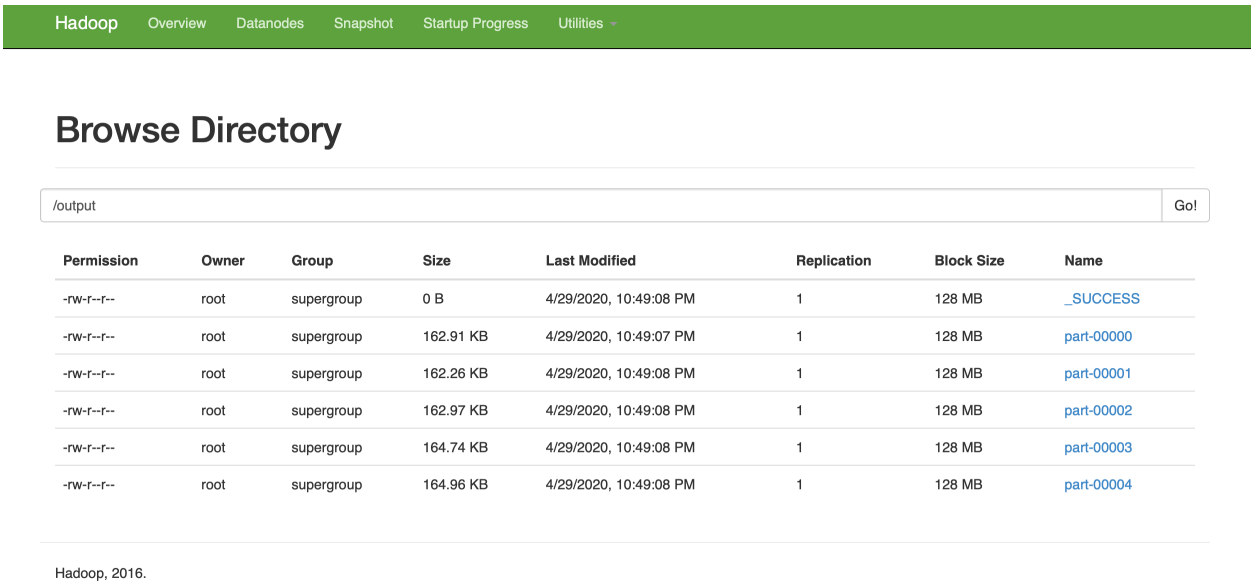
Next, we run the above generated BigramCount.jar on the Hadoop file system, as seen below,

```
root@ip-172-31-24-19:/home/ubuntu/BigramCount# hdfs dfs -rmdir /output
20/04/30 02:48:59 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
root@ip-172-31-24-19:/home/ubuntu/BigramCount# hadoop jar BigramCount.jar BigramCount /input /output
20/04/30 02:49:04 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
20/04/30 02:49:05 INFO Configuration.deprecation: session.id is deprecated. Instead, use dfs.metrics.session-id
20/04/30 02:49:05 INFO jvm.JvmMetrics: Initializing JVM Metrics with processName=JobTracker, sessionId=
20/04/30 02:49:05 INFO jvm.JvmMetrics: Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already initialized
20/04/30 02:49:05 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with
ToolRunner to remedy this.
20/04/30 02:49:05 INFO mapred.FileInputFormat: Total input paths to process : 2
20/04/30 02:49:05 INFO mapreduce.JobSubmitter: number of splits:2
20/04/30 02:49:05 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local760142010_0001
20/04/30 02:49:05 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
20/04/30 02:49:05 INFO mapreduce.Job: Running job: job_local760142010_0001
20/04/30 02:49:05 INFO mapred.LocalJobRunner: OutputCommitter set in config null
20/04/30 02:49:05 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapred.FileOutputCommitter
20/04/30 02:49:05 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 1
20/04/30 02:49:06 INFO mapred.LocalJobRunner: Waiting for map tasks
20/04/30 02:49:06 INFO mapred.LocalJobRunner: Starting task: attempt_local760142010_0001_m_000000_0
20/04/30 02:49:06 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 1
20/04/30 02:49:06 INFO mapred.Task:  Using ResourceCalculatorProcessTree : [ ]
20/04/30 02:49:06 INFO mapred.MapTask: Processing split: hdfs://localhost:9000/input/Hamlet.txt:0+725060
20/04/30 02:49:06 INFO mapred.MapTask: numReduceTasks: 5
20/04/30 02:49:06 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857584)
20/04/30 02:49:06 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
20/04/30 02:49:06 INFO mapred.MapTask: soft limit at 83886080
20/04/30 02:49:06 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857600
20/04/30 02:49:06 INFO mapred.MapTask: kvstart = 26214396; length = 6553600
20/04/30 02:49:06 INFO mapred.MapTask: Map output collector class = org.apache.hadoop.mapred.MapTask$MapOutputBuffer
```

We were able to map and reduce the program!

```
20/04/30 02:49:08 INFO mapreduce.Job:  map 100% reduce 100%
20/04/30 02:49:08 INFO mapreduce.Job: Job job_local760142010_0001 completed successfully
20/04/30 02:49:09 INFO mapreduce.Job: Counters: 35
        File System Counters
                FILE: Number of bytes read=23909934
                FILE: Number of bytes written=33516712
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=9425780
                HDFS: Number of bytes written=2505656
                HDFS: Number of read operations=92
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=37
        Map-Reduce Framework
                Map input records=27426
                Map output records=227600
                Map output bytes=3218658
                Map output materialized bytes=2160336
                Input split bytes=191
                Combine input records=227600
                Combine output records=122888
                Reduce input groups=61444
                Reduce shuffle bytes=2160336
                Reduce input records=122888
                Reduce output records=61444
                Spilled Records=245776
                Shuffled Maps =10
                Failed Shuffles=0
                Merged Map outputs=10
                GC time elapsed (ms)=9
                Total committed heap usage (bytes)=2687500288
        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=1450120
        File Output Format Counters
                Bytes Written=837465
```

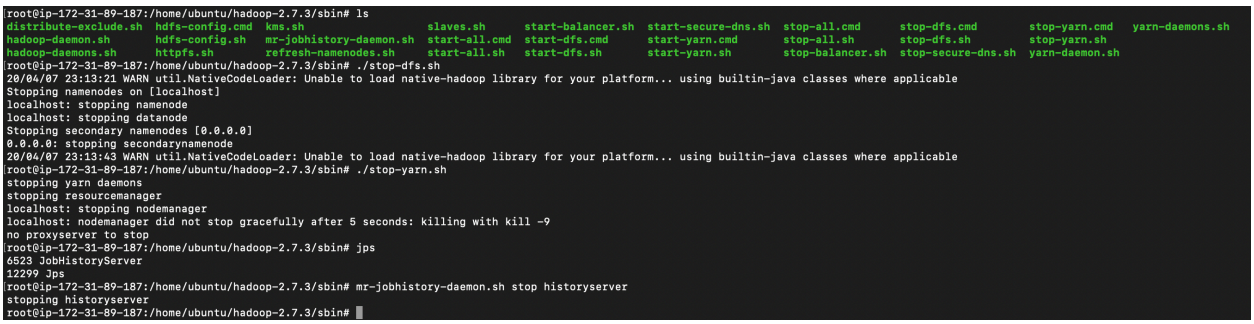
We can see the 5 output files in the Hadoop file system on the website, as seen below,



Next we get the output from the 5 reducers and store it locally, as seen below,

Then we use bash commands to concatenate all of the files into a single data stream, sort that steam alphabetically based on bigram text, and dump it into a single text file, as seen below,

Finally, we are finished with the Hadoop file system and can stop all processes,



Obtaining the Deliverables

For simplicity, all of the deliverables are contained in a single Python script called BigramDeliverables.py. The individual outputs will be elaborated on the following pages, but this script can be seen in actions below,

```
root@ip-172-31-24-19:/home/ubuntu/BigramCount# python3 BigramDeliverables.py
Number of Unique Bigrams: 68010

Examples of Unique Bigrams:

        1)         "'Tis an     1
        2)        "'Tis too     1
        3)    "A gamester!"     1


10 Most Frequent Bigrams and their Frequencies:

        1)           of the   483
        2)            to be   421
        3)           in the   385
        4)           to the   261
        5)           of her   246
        6)             I am   233
        7)           of his   225
        8)         had been   194
        9)        could not   167
       10)          that he   165


Cumulative Frequency of 10 Most Frequent Bigrams: 2780

Cumulative Frequency of All Bigrams: 124974

Faction of 10 Most Frequent Bigrams to All Bigrams: 2780/124974 = 0.02224463 = 2.224463%

Number of Singly Occurring Bigrams: 54161

Examples of Singly Occurring Bigrams:

        1)         "'Tis an
        2)        "'Tis too
        3)    "A gamester!"


5 Most Frequent Words Following "Light" and their Frequencies

        1)        light and     1
        2)         light as     1
        3)       light does     1
        4) light importance     1
        5)         light in     1
        6)         light it     1
        7)      light might     1


root@ip-172-31-24-19:/home/ubuntu/BigramCount#
```

The first part of the BigramDeliverables.py script reads the data from the single concatenated output file into a dictionary with the keys as bigrams and the values as frequencies. This dictionary, bigram_counts, is used for all of the following deliverables.

```
from collections import Counter # For counting highest frequencies
bigram_counts = dict()
output = 'output.txt'
unique_bigrams = 0

# Read in the config file
with open(output) as file:
        # Split based on newlines
        output_lines = file.read().splitlines()
        for line in output_lines:
                # Parse the lines of the output by splitting by whitespace (spaces and tabs)
                line = line.split()
                # Set the bigram key to the first two words
                key = line[0] + ' ' + line[1]
                # Set the biram frequency value to the third word
                value = int(line[2])
                bigram_counts[key] = value
                unique_bigrams += 1
```

1. There are 68,010 unique bigrams. This can be verified since when we read in the file, all of the reducer outputs were already merged (since we only used a single mapper), therefore the number of lines in the file is equivalent to the number of unique bigrams. This counter is seen in the screenshot above, and can be verified in linux below,

```
[root@ip-172-31-24-19:/home/ubuntu/BigramCount# cat output.txt | wc -l
68010
```

a. 3 examples of unique bigrams are easily obtained below, (first three bigrams from the file)

```
Number of Unique Bigrams: 68010

Examples of Unique Bigrams:

        1)          "'Tis an     1
        2)         "'Tis too     1
        3)    "A gamester!"      1
```

```
# 1
print('Number of Unique Bigrams: {}\n'.format(unique_bigrams))

print('Examples of Unique Bigrams:\n')
list_of_bigrams = list(bigram_counts.items())
for i, (key, value) in enumerate(list_of_bigrams[:3], start=1):
    print('\t{:>2}) {:>16}  {:>4}'.format(i, key, value))
print('\n')
```

2. The top ten most frequent bigrams and their counts can be seen below,

```
10 Most Frequent Bigrams and their Frequencies:

        1)           of the   483
        2)            to be   421
        3)           in the   385
        4)           to the   261
        5)           of her   246
        6)             I am   233
        7)           of his   225
        8)         had been   194
        9)        could not   167
       10)          that he   165
```

The Counter module is used to easily find the most frequent counts in this collection of bigrams and their frequencies. The bigram_counts dictionary is passed to the Counter constructor, and subsequently the most_common() method is called with parameter 10 to find the 10 bigrams with the highest frequencies.

```
# 2
print('10 Most Frequent Bigrams and their Frequencies:\n')
# The Counter module provides the most_common() method to guage highest frequencies
bigram_counter = Counter(bigram_counts)
frequent_bigrams = bigram_counter.most_common(10)
for i, (key, value) in enumerate(frequent_bigrams, start=1):
    print('\t{:>2}) {:>16}  {:>4}'.format(i, key, value))
print('\n')
```

3. The bigrams with the 10 highest frequencies have a cumulative frequency of 2780. The cumulative frequency of all bigrams is 124974. Therefore, the ratio of the 10 most frequent bigrams to all bigrams is 2780/124,974 or equivalently 2.22 percent.

```
Cumulative Frequency of 10 Most Frequent Bigrams: 2780

Cumulative Frequency of All Bigrams: 124974

Faction of 10 Most Frequent Bigrams to All Bigrams: 2780/124974 = 0.02224463 = 2.224463%
```

```
# 3
number_of_most_frequent_bigrams = 0
for (key, value) in frequent_bigrams:
    number_of_most_frequent_bigrams += value
print('Cumulative Frequency of 10 Most Frequent Bigrams: {}\n'.format(number_of_most_frequent_bigrams))

number_of_bigrams = sum(bigram_counter.values())
print('Cumulative Frequency of All Bigrams: {}\n'.format(number_of_bigrams))

fraction = number_of_most_frequent_bigrams/number_of_bigrams
print('Faction of 10 Most Frequent Bigrams to All Bigrams: {}/{} = {:.8f} = {:.6f}%\n'.format(number_of_most_frequent_bigrams, number_of_bigrams, fraction, fraction*100))
```

4. There are 54,161 bigrams that only appear once. This can be verified in the screenshot below. This number was obtained by looping through all bigrams in the bigram_counts dictionary and incrementing a counter for each bigram that had a frequency of 1.

a. The first 3 bigrams in the file suffice for this answer too, since they each only occur once.

```
Number of Singly Occurring Bigrams: 54161

Examples of Singly Occurring Bigrams:

        1)          "'Tis an
        2)         "'Tis too
        3)    "A gamester!"
```

```
# 4
number_of_singly_occurring_bigrams = 0
singly_occurring_bigrams = list()
for (key, value) in list_of_bigrams:
    if value == 1:
        number_of_singly_occurring_bigrams += 1
        if len(singly_occurring_bigrams) < 3:
            singly_occurring_bigrams.append(key)
print('Number of Singly Occurring Bigrams: {}\n'.format(number_of_singly_occurring_bigrams))

print('Examples of Singly Occurring Bigrams:\n')
for i, key in enumerate(singly_occurring_bigrams, start=1):
    print('\t{:>2}) {:>16}'.format(i, key))
print('\n')
```

5. The five most frequent words following the word "light" did not make much sense in this context, because 7 words tied with once occurrence. Therefore, I listed all 7 of the bigrams that contained "light" followed by a word

```
5 Most Frequent Words Following "Light" and their Frequencies

         1)        light and     1
         2)         light as     1
         3)       light does     1
         4) light importance     1
         5)         light in     1
         6)         light it     1
         7)      light might     1
```

a. First, we create a dictionary light_bigrams to hold the bigrams containing "light" as the first word. Next, we convert the dictionary of all bigrams to a list of the contents of that dictionary. This will allow us to loop through the items easily. For each bigram in the list, split it using the split() function (and whitespace as a delimiter), and check if the first element of the resulting list is the word "light." If it is, then add this bigram and frequency as a key and value pair, respectively, to our light_bigrams dictionary. If it is not, then continue. Just as before, we invoke the Counter module to count the frequencies of the bigrams in the light_bigrams dictionary. The Counter module is used to easily find the most frequent counts in this collection of bigrams and their frequencies. The light_bigrams dictionary is passed to the Counter constructor, and subsequently the most_common() method is called to find the bigrams with the highest frequencies. It is observed there are 7 words with the frequency of 1, so we print them all.

```
# 5
print('5 Most Frequent Words Following \"Light\" and their Frequencies\n')
# Loop through bigrams to find those that start with "light"
light_bigrams = dict()
for (key, value) in list_of_bigrams:
    words = key.split()
    if words[0] == 'light':
        light_bigrams[key] = value
# The Counter module provides the most_common() method to gauge highest frequencies
light_bigram_counter = Counter(light_bigrams)
frequent_light_bigrams = light_bigram_counter.most_common() # Note that all 7 words
for i, (key, value) in enumerate(frequent_light_bigrams, start=1):
    print('\t{:>2}) {:>16}  {:>4}'.format(i, key, value))
print('\n')
```

6. This assignment was completed on the cloud using an Ubuntu AWS EC2 Instance, as documented above.

Part 2: Pandas

For simplicity, all of the deliverables were again are contained in a single Python script, this time called explore_data_using_pandas.py. The individual outputs will be elaborated on the following pages, but this script can be seen in actions below,

```
[ec2-user@ip-172-31-36-38 PandasDataAnalysis]$ python3 explore_data_using_pandas.py
```

1. Description of the data from my report:

For my final project, I created 6 web scrapers that collect information pertaining to government COVID-19-related emergency orders and stores them in CSV files. This collection of web scrapers uses the BeautifulSoup 4 and requests libraries in order to collect the text from each state in the New England region of the United States. After collecting the text, there are generally 3 elements of information that are scraped. These elements are:

1. Date, or the day that the measure was passed by the government.
2. Description, or the supplied summary of the measure.
3. PDF Link, or the source to the primary text the order.

These elements of information were available and scraped for all 6 of the states in the New England region. The csv files contain these attributes in their columns and the emergency orders in their rows, sorted from most recent to least recent date. Sometimes, more information was available. For example, Vermont and Massachusetts offered Press Release article links and Guidance links. I included these in their respective csv files.

In order to ensure consistency between files, the dates and descriptions had to be pre-processed. The dates were written in a variety of forms in the webpages. Using the datetime library, all of these dates were rewritten into mm/dd/yyyy format.

The descriptions generally needed more intricate processes to extract only the supplied text summaries for the government orders. A common theme across all of the government websites was inconsistent formatting, across both html tabs and the text within.

The inconsistent html tabs made me innovate solutions that rely less on positioning of tabs in the contents, and cater more to the relative structure of the contents. This approach is perhaps more robust to inconsistencies, but could still break if contents are rearranged.

The text was cleaned and pre-processed on a case-by-case basis. For example, the regex library was used for scraping Connecticut's website to eliminate excess spaces in the text. Many of the websites also needed their descriptions to be split to only include the description-associated text. For example, I was able to split off any text not related to the emergency order by looking for rare characters such as colons or parentheses, and breaking the text there. This will generally work, unless the website places one of these characters in the body of a summary.

I envision that this data will be able to be used to create useful features for deep learning algorithms, if enough data is scraped in consistent formats from all of the states.

2. The Pandas package is imported in the screenshot below,

```
# 2
import pandas as pd
```

3. I scraped 6 state's government measures from their websites, so I loaded all 6 state's government measures into separate DataFrames using the read_csv() function. The output from head() is also displayed below,

```
# 3
# Read each state's government measures data into separate DataFrames
ct_df = pd.read_csv('Outputs/ct.csv', encoding='utf8')
ma_df = pd.read_csv('Outputs/ma.csv', encoding='utf8')
vt_df = pd.read_csv('Outputs/vt.csv', encoding='utf8')
ri_df = pd.read_csv('Outputs/ri.csv', encoding='utf8')
nh_df = pd.read_csv('Outputs/nh.csv', encoding='utf8')
me_df = pd.read_csv('Outputs/me.csv', encoding='utf8')
```

In the head() outputs below, we can see that most of the websites scraped had dates, descriptions, and PDF links. Massachusetts and Vermont also offered press release links and guidance links.

```
[ec2-user@ip-172-31-36-38 PandasDataAnalysis]$ python3 explore_data_using_pandas.py
DataFrame for Connecticut
        Date                                        Description                                      Order_PDF_Link
0  04/24/2020  Additional flexibility for Medicaid-enrolled p...  https://portal.ct.gov/-/media/Office-of-the-Go...
1  04/23/2020  Mandatory reporting by managed residential com...  https://portal.ct.gov/-/media/Office-of-the-Go...
2  04/22/2020  Additions to the definition of telehealth prov...  https://portal.ct.gov/-/media/Office-of-the-Go...
3  04/21/2020  Applicability of Executive Order No. 7S, Secti...  https://portal.ct.gov/-/media/Office-of-the-Go...
4  04/17/2020  Cloth face coverings or higher level of protec...  https://portal.ct.gov/-/media/Office-of-the-Go...
DataFrame for Massachusetts
        Date                                        Description                                      Order_PDF_Link Press_Release_Link                                      Guidance_Link
0  04/20/2020   On April 20, the Department of Public Health ...  https://www.mass.gov/doc/csc-data-reporting-or...              NaN  https://www.mass.gov/doc/statewide-advisory-co...
1  04/16/2020   On April 16, Governor Baker issued an emergen...   https://www.mass.gov/doc/april-16-2020-eec-order              NaN                                                NaN
2  04/14/2020   On April 14, the Department of Public Health ...                                                NaN              NaN  https://www.mass.gov/doc/non-healthcare-essent...
3  04/09/2020   On April 9, Governor Baker issued an emergenc...  https://www.mass.gov/doc/april-9-2020-foreign-...              NaN                                                NaN
4  04/09/2020   On April 9, Governor Baker issued an emergenc...  https://www.mass.gov/doc/april-9-2020-nursing-...              NaN  https://www.mass.gov/doc/nursing-student-guida...
DataFrame for Vermont
        Date                                        Description                                      Order_PDF_Link                                Press_Release_Link Guidance_Link
0  04/24/2020  On April 24, outlined some additional openings...  https://governor.vermont.gov/sites/scott/files...  https://governor.vermont.gov/press-release/new...           NaN
1  04/17/2020  On April 17, outlined an approach for the phas...  https://governor.vermont.gov/sites/scott/files...  https://governor.vermont.gov/press-release/gov...           NaN
2  04/10/2020  On April 10, extended Vermont's State of Emerg...  https://governor.vermont.gov/sites/scott/files...  https://governor.vermont.gov/press-release/gov...           NaN
3  04/07/2020  On April 7, requested federal disaster funds t...                                                NaN  https://governor.vermont.gov/press-release/gov...           NaN
4  03/30/2020  On March 30, Governor Scott ordered residents ...  https://governor.vermont.gov/sites/scott/files...          https://governor.vermont.gov/addendum7pr           NaN
DataFrame for Rhode Island
      Date                                        Description                                      Order_PDF_Link
0  4/17/20                          Extension of Executive Orders   https://governor.ri.gov/documents/orders/Execu...
1  4/17/20   Further Preparations for a Predominantly Mail...  https://governor.ri.gov/documents/orders/Execu...
2  4/15/20   Support for Young Adults Aging Out of State C...  https://governor.ri.gov/documents/orders/Execu...
3  4/15/20       Public Meetings and Public Records Requests   https://governor.ri.gov/documents/orders/Execu...
4  4/14/20                 Requiring Cloth Face Masks at Work   https://governor.ri.gov/documents/orders/Execu...
DataFrame for New Hampshire
        Date                                        Description                                      Order_PDF_Link
0  04/2020  nsuring Worker's Compensation coverage of New ...  https://www.governor.nh.gov/news-media/emergen...
1  04/2020  n order temporarily waiving the 28-day separat...  https://www.governor.nh.gov/news-media/emergen...
2  04/2020   Further temporary requirements regarding heal...  https://www.governor.nh.gov/news-media/emergen...
3  04/2020   Activation of the New Hampshire Crisis Standa...  https://www.governor.nh.gov/news-media/emergen...
4  04/2020   Extension of Emergency Orders #1 and #19 (tem...  https://www.governor.nh.gov/news-media/emergen...
DataFrame for Maine
        Date                                        Description                                      Order_PDF_Link
0  4/25/2020  An Order to Expedite the Delivery of Certain U...  https://www.maine.gov/governor/mills/sites/mai...
1  4/25/2020      An Order Extending a Certain Reporting Period  https://www.maine.gov/governor/mills/sites/mai...
2  4/24/2020  An Order Expediting Extraordinary Emergency Fi...  https://www.maine.gov/governor/mills/sites/mai...
3  4/21/2020  An Order Regarding a Covid-19 Related Loan Gua...  https://www.maine.gov/governor/mills/sites/mai...
4  4/21/2020  An Order Regarding Certain Laws Enforced by th...  https://www.maine.gov/governor/mills/sites/mai...
```

4. The data types for each column in each DataFrame can be seen below. All of the data is Strings, which are represented as objects by Pandas. Most of the websites scraped had dates, descriptions, and PDF links. Massachusetts and Vermont also offered press release links and guidance links.

```
Datatypes for Connecticut
Date              object
Description       object
Order_PDF_Link    object
dtype: object
Datatypes for Massachusetts
Date                  object
Description           object
Order_PDF_Link        object
Press_Release_Link    object
Guidance_Link         object
dtype: object
Datatypes for Vermont
Date                  object
Description           object
Order_PDF_Link        object
Press_Release_Link    object
Guidance_Link         object
dtype: object
Datatypes for Rhode Island
Date              object
Description       object
Order_PDF_Link    object
dtype: object
Datatypes for New Hampshire
Date              object
Description       object
Order_PDF_Link    object
dtype: object
Datatypes for Maine
Date              object
Description       object
Order_PDF_Link    object
dtype: object
```

```python
# 4
# Check the datatype of each column
print('Datatypes for Connecticut')
datatypes_ct = ct_df.dtypes
print(datatypes_ct)
print('Datatypes for Massachusetts')
datatypes_ma = ma_df.dtypes
print(datatypes_ma)
print('Datatypes for Vermont')
datatypes_vt = vt_df.dtypes
print(datatypes_vt)
print('Datatypes for Rhode Island')
datatypes_ri = ri_df.dtypes
print(datatypes_ri)
print('Datatypes for New Hampshire')
datatypes_nh = nh_df.dtypes
print(datatypes_nh)
print('Datatypes for Maine')
datatypes_me = me_df.dtypes
print(datatypes_me)
print('\n')
```

5. Sorting by the column 'Date', which consists of the dates that the government measures were passed. The top 15 entries are indicated below in descending order.

```
Sorting by Date
         Date                                        Description                                        Order_PDF_Link
0   04/24/2020  Additional flexibility for Medicaid-enrolled p...  https://portal.ct.gov/-/media/Office-of-the-Go...
1   04/23/2020  Mandatory reporting by managed residential com...  https://portal.ct.gov/-/media/Office-of-the-Go...
2   04/22/2020  Additions to the definition of telehealth prov...  https://portal.ct.gov/-/media/Office-of-the-Go...
3   04/21/2020  Applicability of Executive Order No. 7S, Secti...  https://portal.ct.gov/-/media/Office-of-the-Go...
4   04/17/2020  Cloth face coverings or higher level of protec...  https://portal.ct.gov/-/media/Office-of-the-Go...
5   04/15/2020  Approval of temporary additional nursing home ...  https://portal.ct.gov/-/media/Office-of-the-Go...
6   04/14/2020  Modification of state contracting statutes to ...  https://portal.ct.gov/-/media/Office-of-the-Go...
7   04/11/2020       Implementation of a nursing home surge plan.  https://portal.ct.gov/-/media/Office-of-the-Go...
8   04/10/2020  Protections for residential renters impacted b...  https://portal.ct.gov/-/media/Office-of-the-Go...
9   04/09/2020  Suspension and modification of tax deadlines a...  https://portal.ct.gov/-/media/Office-of-the-Go...
10  04/07/2020  Safe workplaces in essential businesses. Tempo...  https://portal.ct.gov/-/media/Office-of-the-Go...
11  04/05/2020  Protection from civil liability for actions or...  https://portal.ct.gov/-/media/Office-of-the-Go...
12  04/02/2020  Prohibition on non-essential lodging. Further ...  https://portal.ct.gov/-/media/Office-of-the-Go...
13  04/01/2020  Safe stores mandatory statewide rules. 60-day ...  https://portal.ct.gov/-/media/Office-of-the-Go...
14  03/31/2020  Continuation of funding for boards of educatio...  https://portal.ct.gov/-/media/Office-of-the-Go...
```

The sort_values() function with the 'Date' column passed as parameter will sort based dates. The ascending parameter is set to False so that the dates are given in descending order. The head() function with a parameter of 15 gives the first 15 entries in the resulting DataFrame,

```
# 5
# Sorting by date and show the top 15 entries in descending order
print('Sorting by Date')
print(ct_df.sort_values('Date', ascending=False).head(15))
print('\n')
```

6. Horizontal and Vertical Filtering

a. Horizontal filtering by selecting the 'Date' and 'Description' columns, which consist of the dates that the government measures were passed and the summaries of those government measures, respectively, and leaving out the column 'Order_PDF_Link', which consists of the links to the source text of the government measure. The top 15 entries are indicated below in descending order.

```
# 6a
# Remove the PDF Order Link column from the DataFrame
print('Horizontal Slicing')
filter_columns = ['Date', 'Description']
vertically_filtered_ct_df = ct_df[filter_columns]
print(vertically_filtered_ct_df.head(15))
print('\n')
```

The orders are displayed in descending order since their dates are from most recent to least recent. Only the "Date" and "Description" columns are seen, confirming our horizontal filter.

```
Horizontal Slicing
         Date                                        Description
0   04/24/2020  Additional flexibility for Medicaid-enrolled p...
1   04/23/2020  Mandatory reporting by managed residential com...
2   04/22/2020  Additions to the definition of telehealth prov...
3   04/21/2020  Applicability of Executive Order No. 7S, Secti...
4   04/17/2020  Cloth face coverings or higher level of protec...
5   04/15/2020  Approval of temporary additional nursing home ...
6   04/14/2020  Modification of state contracting statutes to ...
7   04/11/2020       Implementation of a nursing home surge plan.
8   04/10/2020  Protections for residential renters impacted b...
9   04/09/2020  Suspension and modification of tax deadlines a...
10  04/07/2020  Safe workplaces in essential businesses. Tempo...
11  04/05/2020  Protection from civil liability for actions or...
12  04/02/2020  Prohibition on non-essential lodging. Further ...
13  04/01/2020  Safe stores mandatory statewide rules. 60-day ...
14  03/31/2020  Continuation of funding for boards of educatio...
```

b. Vertical filtering by selecting the rows with dates that are before "03/31/2020". The top 15 entries are indicated below in descending order.

```
# 6b
# Remove any data entries with dates after March 31st, 2020 from the DataFrame
print('Vertical Slicing')
filter_date = '03/31/2020'
horizontally_filtered_ct_df = ct_df[ct_df['Date'] < filter_date]
print(horizontally_filtered_ct_df.head(15))
print('\n')
```

The orders are displayed in descending order since their dates are from most recent to least recent. Only dates before "03/31/2020" are seen, confirming our vertical filter.

```
Vertical Slicing
         Date                                        Description                                        Order_PDF_Link
15  03/30/2020  Requirement of limited group sizes in childcar...  https://portal.ct.gov/-/media/Office-of-the-Go...
16  03/28/2020  Authorization to provide for non-congregant ho...  https://portal.ct.gov/-/media/Office-of-the-Go...
17  03/27/2020  Suspension of license renewals and inspections...  https://portal.ct.gov/-/media/Office-of-the-Go...
18  03/26/2020  Further reduction of social and recreational g...  https://portal.ct.gov/-/media/Office-of-the-Go...
19  03/25/2020  Tolling of time periods for DOT final determin...  https://portal.ct.gov/-/media/Office-of-the-Go...
20  03/24/2020  Extension of class cancellations at all public...  https://portal.ct.gov/-/media/Office-of-the-Go...
21  03/23/2020  Suspension of non-critical probate court opera...  https://portal.ct.gov/-/media/Office-of-the-Go...
22  03/22/2020  Clarification of "Stay Safe, Stay Home" Execut...  https://portal.ct.gov/-/media/Office-of-the-Go...
23  03/21/2020  Modifications to DSS benefits . Flexibility re...  https://portal.ct.gov/-/media/Office-of-the-Go...
24  03/20/2020  "Stay Safe, Stay Home" restrictions on all wor...  https://portal.ct.gov/-/media/Office-of-the-Go...
25  03/19/2020  Postponement of presidential primary to June 2...  https://portal.ct.gov/-/media/Office-of-the-Go...
26  03/18/2020  Closure of large, indoor shopping malls. Closu...  https://portal.ct.gov/-/media/Office-of-the-Go...
27  03/17/2020  Further modification of 180-day school year re...  https://portal.ct.gov/-/media/Office-of-the-Go...
28  03/16/2020  Further reduction of social and recreational g...  https://portal.ct.gov/-/media/Office-of-the-Go...
29  03/15/2020  Cancellation of classes at all public schools....  https://portal.ct.gov/-/media/Office-of-the-Go...
```

7. Here, we merged 2 DataFrames in 2 different ways. First, we merged Connecticut's and Vermont's DataFrames using an "Outer" merge, then we merged them using a "Left" merge. I thought that these would be interesting DataFrames to merge because Vermont is one of the files that had extra columns of data. Let's examine the merges individually below,

```
# 7
# Merge 2 DataFrames
print('Outer Merge')
outer_join = pd.merge(ct_df, vt_df, how='outer')
print(outer_join)

print('Left Merge')
left_join = pd.merge(ct_df, vt_df, how='left')
print(left_join)
```

The "Outer" merge completely combines the rows of each DataFrame. Since both DataFrames had the "Date", "Description", and "PDF_Order_Link" columns in common, the data stacks up very neatly for the most part. However, Vermont introduces two new attributes in this merged DataFrame, namely "Press_Release_Link" and "Guidance_Link". Since Connecticut did not have data for any of these fields, the DataFrame marks these valued as "NaN." On another note, Connecticut did not always provide these fields, so sometimes it has "NaNs" in thse fields too.

```
Outer Merge
         Date                                        Description  ...                                Press_Release_Link                                Guidance_Link
0    04/24/2020  Additional flexibility for Medicaid-enrolled p...  ...                                               NaN                                          NaN
1    04/23/2020  Mandatory reporting by managed residential com...  ...                                               NaN                                          NaN
2    04/22/2020  Additions to the definition of telehealth prov...  ...                                               NaN                                          NaN
3    04/21/2020  Applicability of Executive Order No. 7S, Secti...  ...                                               NaN                                          NaN
4    04/17/2020  Cloth face coverings or higher level of protec...  ...                                               NaN                                          NaN
5    04/15/2020  Approval of temporary additional nursing home ...  ...                                               NaN                                          NaN
6    04/14/2020  Modification of state contracting statutes to ...  ...                                               NaN                                          NaN
7    04/11/2020       Implementation of a nursing home surge plan.  ...                                               NaN                                          NaN
8    04/10/2020  Protections for residential renters impacted b...  ...                                               NaN                                          NaN
9    04/09/2020  Suspension and modification of tax deadlines a...  ...                                               NaN                                          NaN
10   04/07/2020  Safe workplaces in essential businesses. Tempo...  ...                                               NaN                                          NaN
11   04/05/2020  Protection from civil liability for actions or...  ...                                               NaN                                          NaN
12   04/02/2020  Prohibition on non-essential lodging. Further ...  ...                                               NaN                                          NaN
13   04/01/2020  Safe stores mandatory statewide rules. 60-day ...  ...                                               NaN                                          NaN
14   03/31/2020  Continuation of funding for boards of educatio...  ...                                               NaN                                          NaN
15   03/30/2020  Requirement of limited group sizes in childcar...  ...                                               NaN                                          NaN
16   03/28/2020  Authorization to provide for non-congregant ho...  ...                                               NaN                                          NaN
17   03/27/2020  Suspension of license renewals and inspections...  ...                                               NaN                                          NaN
18   03/26/2020  Further reduction of social and recreational g...  ...                                               NaN                                          NaN
19   03/25/2020  Tolling of time periods for DOT final determin...  ...                                               NaN                                          NaN
20   03/24/2020  Extension of class cancellations at all public...  ...                                               NaN                                          NaN
21   03/23/2020  Suspension of non-critical probate court opera...  ...                                               NaN                                          NaN
22   03/22/2020  Clarification of "Stay Safe, Stay Home" Execut...  ...                                               NaN                                          NaN
23   03/21/2020  Modifications to DSS benefits . Flexibility re...  ...                                               NaN                                          NaN
24   03/20/2020  "Stay Safe, Stay Home" restrictions on all wor...  ...                                               NaN                                          NaN
25   03/19/2020  Postponement of presidential primary to June 2...  ...                                               NaN                                          NaN
26   03/18/2020  Closure of large, indoor shopping malls. Closu...  ...                                               NaN                                          NaN
27   03/17/2020  Further modification of 180-day school year re...  ...                                               NaN                                          NaN
28   03/16/2020  Further reduction of social and recreational g...  ...                                               NaN                                          NaN
29   03/15/2020  Cancellation of classes at all public schools....  ...                                               NaN                                          NaN
30   03/14/2020  Suspension of in-person open meeting requireme...  ...                                               NaN                                          NaN
31   03/13/2020  Grants DPH commissioner authority to restrict ...  ...                                               NaN                                          NaN
32   03/12/2020  Prohibition of social and recreational gatheri...  ...                                               NaN                                          NaN
33   04/24/2020  On April 24, outlined some additional openings...  ...  https://governor.vermont.gov/press-release/new...                                          NaN
34   04/17/2020  On April 17, outlined an approach for the phas...  ...  https://governor.vermont.gov/press-release/gov...                                          NaN
35   04/10/2020  On April 10, extended Vermont's State of Emerg...  ...  https://governor.vermont.gov/press-release/gov...                                          NaN
36   04/07/2020  On April 7, requested federal disaster funds t...  ...  https://governor.vermont.gov/press-release/gov...                                          NaN
37   03/30/2020  On March 30, Governor Scott ordered residents ...  ...            https://governor.vermont.gov/addendum7pr                                          NaN
38   03/26/2020  On March 26, Governor Scott directed directed ...  ...  https://governor.vermont.gov/press-release/gov...  https://education.vermont.gov/news/covid-19-gu...
39   03/24/2020  On March 24, Governor Scott issued a "Stay Hom...  ...  https://governor.vermont.gov/press-release/gov...                                          NaN
40   03/23/2020  On March 23, Governor Scott ordered telecommut...  ...  https://governor.vermont.gov/press-release/gov...                                          NaN
41   03/21/2020  On March 21, Governor Scott ordered the closur...  ...  https://governor.vermont.gov/press-release/gov...                                          NaN
42   03/21/2020  On March 21, Governor Scott further restricted...  ...  https://governor.vermont.gov/press-release/gov...                                          NaN
43   03/20/2020  On March 20, Governor Scott ordered suspension...  ...                                               NaN                                          NaN
44   03/18/2020  On March 18, Governor Scott suspended in-perso...  ...  https://dmv.vermont.gov/blog/covid-19-continui...                                          NaN
45   03/17/2020  On March 17, Governor Scott directed childcare...  ...  https://governor.vermont.gov/press-release/gov...  https://education.vermont.gov/documents/guidan...
46   03/16/2020  On March 16, Governor Scott ordered the closur...  ...  https://governor.vermont.gov/press-release/gov...                                          NaN
47   03/15/2020  On March 15, Governor Scott directed the dismi...  ...  https://governor.vermont.gov/press-release/gov...  https://education.vermont.gov/news/covid-19-gu...
48   03/13/2020  On March 13, Governor Scott restricted visitor...  ...                                               NaN  https://dail.vermont.gov/novel-coronavirus-inf...
```

The "Left" merge completely overlays the data in the left parmater into the right parameter. Since both DataFrames had the "Date", "Description", and "PDF_Order_Link" columns in common, but Connecticut has every date that Vermont has, and more, Connecticut ends up overwriting all of Vermont's data in this merged DataFrame. However, Vermont introduces two new attributes in this merged DataFrame, namely "Press_Release_Link" and "Guidance_Link". Since Connecticut did not have data for any of these fields, the DataFrame marks these valued as "NaN."

Left Merge

	Date	Description	Order_PDF_Link	Press_Release_Link	Guidance_Link
0	04/24/2020	Additional flexibility for Medicaid-enrolled p...	https://portal.ct.gov/-/media/Office-of-the-Go...	NaN	NaN
1	04/23/2020	Mandatory reporting by managed residential com...	https://portal.ct.gov/-/media/Office-of-the-Go...	NaN	NaN
2	04/22/2020	Additions to the definition of telehealth prov...	https://portal.ct.gov/-/media/Office-of-the-Go...	NaN	NaN
3	04/21/2020	Applicability of Executive Order No. 7S, Secti...	https://portal.ct.gov/-/media/Office-of-the-Go...	NaN	NaN
4	04/17/2020	Cloth face coverings or higher level of protec...	https://portal.ct.gov/-/media/Office-of-the-Go...	NaN	NaN
5	04/15/2020	Approval of temporary additional nursing home ...	https://portal.ct.gov/-/media/Office-of-the-Go...	NaN	NaN
6	04/14/2020	Modification of state contracting statutes to ...	https://portal.ct.gov/-/media/Office-of-the-Go...	NaN	NaN
7	04/11/2020	Implementation of a nursing home surge plan.	https://portal.ct.gov/-/media/Office-of-the-Go...	NaN	NaN
8	04/10/2020	Protections for residential renters impacted b...	https://portal.ct.gov/-/media/Office-of-the-Go...	NaN	NaN
9	04/09/2020	Suspension and modification of tax deadlines a...	https://portal.ct.gov/-/media/Office-of-the-Go...	NaN	NaN
10	04/07/2020	Safe workplaces in essential businesses. Tempo...	https://portal.ct.gov/-/media/Office-of-the-Go...	NaN	NaN
11	04/05/2020	Protection from civil liability for actions or...	https://portal.ct.gov/-/media/Office-of-the-Go...	NaN	NaN
12	04/02/2020	Prohibition on non-essential lodging. Further ...	https://portal.ct.gov/-/media/Office-of-the-Go...	NaN	NaN
13	04/01/2020	Safe stores mandatory statewide rules. 60-day ...	https://portal.ct.gov/-/media/Office-of-the-Go...	NaN	NaN
14	03/31/2020	Continuation of funding for boards of educatio...	https://portal.ct.gov/-/media/Office-of-the-Go...	NaN	NaN
15	03/30/2020	Requirement of limited group sizes in childcar...	https://portal.ct.gov/-/media/Office-of-the-Go...	NaN	NaN
16	03/28/2020	Authorization to provide for non-congregant ho...	https://portal.ct.gov/-/media/Office-of-the-Go...	NaN	NaN
17	03/27/2020	Suspension of license renewals and inspections...	https://portal.ct.gov/-/media/Office-of-the-Go...	NaN	NaN
18	03/26/2020	Further reduction of social and recreational g...	https://portal.ct.gov/-/media/Office-of-the-Go...	NaN	NaN
19	03/25/2020	Tolling of time periods for DOT final determin...	https://portal.ct.gov/-/media/Office-of-the-Go...	NaN	NaN
20	03/24/2020	Extension of class cancellations at all public...	https://portal.ct.gov/-/media/Office-of-the-Go...	NaN	NaN
21	03/23/2020	Suspension of non-critical probate court opera...	https://portal.ct.gov/-/media/Office-of-the-Go...	NaN	NaN
22	03/22/2020	Clarification of "Stay Safe, Stay Home" Execut...	https://portal.ct.gov/-/media/Office-of-the-Go...	NaN	NaN
23	03/21/2020	Modifications to DSS benefits . Flexibility re...	https://portal.ct.gov/-/media/Office-of-the-Go...	NaN	NaN
24	03/20/2020	"Stay Safe, Stay Home" restrictions on all wor...	https://portal.ct.gov/-/media/Office-of-the-Go...	NaN	NaN
25	03/19/2020	Postponement of presidential primary to June 2...	https://portal.ct.gov/-/media/Office-of-the-Go...	NaN	NaN
26	03/18/2020	Closure of large, indoor shopping malls. Closu...	https://portal.ct.gov/-/media/Office-of-the-Go...	NaN	NaN
27	03/17/2020	Further modification of 180-day school year re...	https://portal.ct.gov/-/media/Office-of-the-Go...	NaN	NaN
28	03/16/2020	Further reduction of social and recreational g...	https://portal.ct.gov/-/media/Office-of-the-Go...	NaN	NaN
29	03/15/2020	Cancellation of classes at all public schools....	https://portal.ct.gov/-/media/Office-of-the-Go...	NaN	NaN
30	03/14/2020	Suspension of in-person open meeting requireme...	https://portal.ct.gov/-/media/Office-of-the-Go...	NaN	NaN
31	03/13/2020	Grants DPH commissioner authority to restrict ...	https://portal.ct.gov/-/media/Office-of-the-Go...	NaN	NaN
32	03/12/2020	Prohibition of social and recreational gatheri...	https://portal.ct.gov/-/media/Office-of-the-Go...	NaN	NaN

8. Finally, we export the merged DataFrames to JSON files.

```
# 8
# Save the outer and left merges to JSON
outer_join.to_json('ct-vt-outer.json', orient='columns')
left_join.to_json('ct-vt-left.json', orient='columns')
```