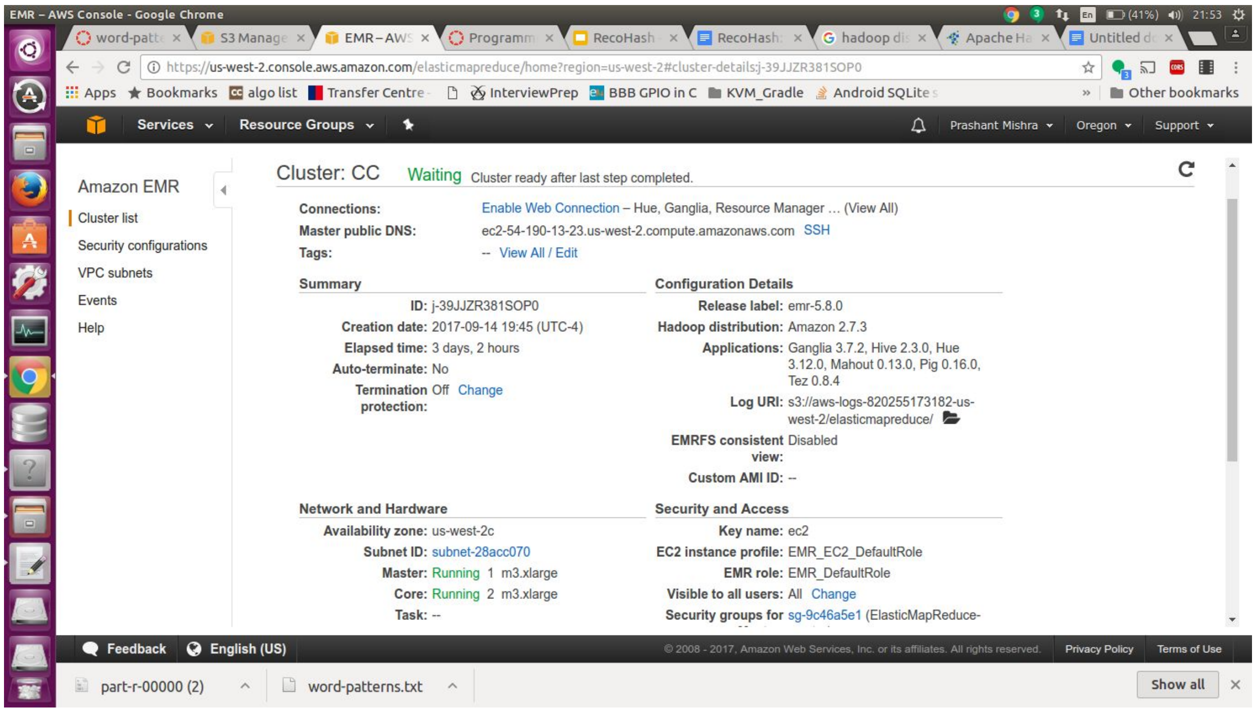
Readme for Assignment 1

Submission Zip Folder Contents

1. 3 JAR files: SingleWordCount.jar, DoubleWordCount.jar, DistributedWordCount.jar
2. Readme
3. 3 Output Folders: SingleWordCountOutput, DoubleWordCountOutput, DistributedWordCount

Cluster Setup

I have used Amazon AWS EMR (Elastic Map-Reduce) to set up a cluster of 3 machines (One master and two slave nodes). Screenshots of the setup below:


EMR – AWS Console - Google Chrome
https://us-west-2.console.aws.amazon.com/elasticmapreduce/home?region=us-west-2#cluster-details:j-39JJZR381SOP0
Services
Resource Groups
Prashant Mishra
Oregon
Support
Amazon EMR
Cluster list
Security configurations
VPC subnets
Events
Help
Feedback
English (US)
© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
part-r-00000 (2)
word-patterns.txt
Show all

Cluster: CC Waiting Cluster ready after last step completed.

Connections: Enable Web Connection – Hue, Ganglia, Resource Manager ... (View All)
Master public DNS: ec2-54-190-13-23.us-west-2.compute.amazonaws.com SSH
Tags: -- View All / Edit

Summary

ID: j-39JJZR381SOP0
Creation date: 2017-09-14 19:45 (UTC-4)
Elapsed time: 3 days, 2 hours
Auto-terminate: No
Termination protection: Off Change

Configuration Details

Release label: emr-5.8.0
Hadoop distribution: Amazon 2.7.3
Applications: Ganglia 3.7.2, Hive 2.3.0, Hue 3.12.0, Mahout 0.13.0, Pig 0.16.0, Tez 0.8.4
Log URI: s3://aws-logs-820255173182-us-west-2/elasticmapreduce/
EMRFS consistent view: Disabled
Custom AMI ID: --

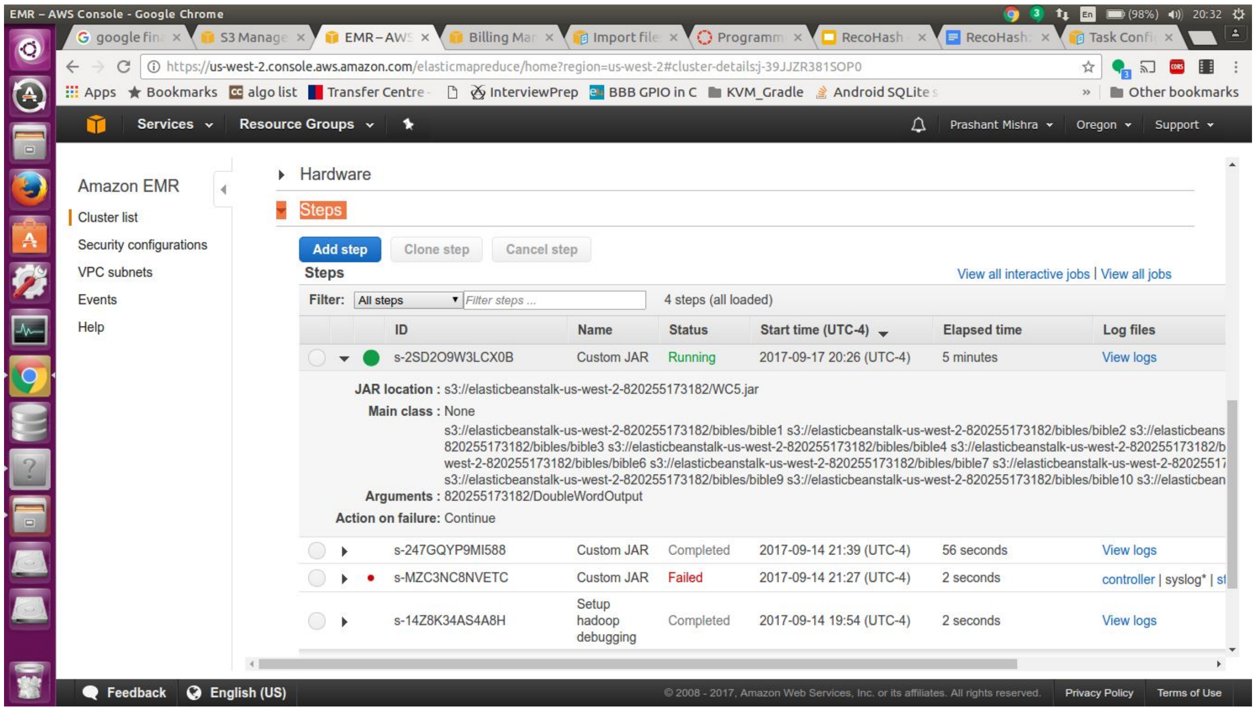
Network and Hardware

Availability zone: us-west-2c
Subnet ID: subnet-28acc070
Master: Running 1 m3.xlarge
Core: Running 2 m3.xlarge
Task: --

Security and Access

Key name: ec2
EC2 instance profile: EMR_EC2_DefaultRole
EMR role: EMR_DefaultRole
Visible to all users: All Change
Security groups for sg-9c46a5e1 (ElasticMapReduce-

Once this cluster is set up, we can add a map-reduce job by adding a new step with a custom jar file that contains all the map-reduce JAVA code. A screenshot of a particular step is show below. All the parameters to the job is stored in AWS S3. As you can see we can specify the location of the custom jar, and location of all the 10 copies of bible and the output folder as arguments to this jar.


EMR – AWS Console - Google Chrome
https://us-west-2.console.aws.amazon.com/elasticmapreduce/home?region=us-west-2#cluster-details:j-39JJZR381SOP0
Services
Resource Groups
Prashant Mishra
Oregon
Support
Amazon EMR
Cluster list
Security configurations
VPC subnets
Events
Help
Hardware
Steps
Add step
Clone step
Cancel step
View all interactive jobs | View all jobs
Feedback
English (US)
© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

Steps

Filter: All steps | Filter steps ... | 4 steps (all loaded)

ID	Name	Status	Start time (UTC-4)	Elapsed time	Log files
s-2SD2O9W3LCX0B	Custom JAR	Running	2017-09-17 20:26 (UTC-4)	5 minutes	View logs
s-247GQYP9MI588	Custom JAR	Completed	2017-09-14 21:39 (UTC-4)	56 seconds	View logs
s-MZC3NC8NVETC	Custom JAR	Failed	2017-09-14 21:27 (UTC-4)	2 seconds	controller | syslog* | st
s-14Z8K34AS4A8H	Setup hadoop debugging	Completed	2017-09-14 19:54 (UTC-4)	2 seconds	View logs

JAR location : s3://elasticbeanstalk-us-west-2-820255173182/WC5.jar
Main class : None
Arguments : s3://elasticbeanstalk-us-west-2-820255173182/bibles/bible1 s3://elasticbeanstalk-us-west-2-820255173182/bibles/bible2 s3://elasticbeans
820255173182/bibles/bible3 s3://elasticbeanstalk-us-west-2-820255173182/bibles/bible4 s3://elasticbeanstalk-us-west-2-820255173182/b
west-2-820255173182/bibles/bible6 s3://elasticbeanstalk-us-west-2-820255173182/bibles/bible7 s3://elasticbeanstalk-us-west-2-8202551
s3://elasticbeanstalk-us-west-2-820255173182/bibles/bible9 s3://elasticbeanstalk-us-west-2-820255173182/bibles/bible10 s3://elasticbean
820255173182/DoubleWordOutput
Action on failure: Continue

JAR Files

For single word count, I got the code from https://github.com/uzresk/aws-emr-examples/tree/master/src/main/java/aws/emr/wordcount. For the other parts I have made slight modifications to the Mapper class to work for the new cases.

Here in the code we have 3 classes, WordCount.java, WordCountMapper.java and WordCountReducer.java

WordCount.java

The main method of this class, initiates a new hadoop job and sets the input and output expected file types and their path from arguments.
Specifically in distributed cache example, this method also sets up the code to add the cache file to the distributed cache, and because of this, each data node can access this file as if from its local file system and processing becomes faster.

WordCountMapper.java

The mapper class has the method map which is common in all three cases. Given a line of text as the input argument, it splits the line into a list of words, and for each of the words issues a map with key as the word and its frequency as 1.

For double word count, I used the key as word[currentIndex] + "," + word[currentIndex + 1] to get pairs of words as key. The rest of the code remains the same.

For distributed cache example, the mapper class has two additional methods, setup and readFile. Setup is an overidden method that always runs before the map method. It gets all the files present in the distributed cache, and then calls readFile on each of those files. ReadFile method basically parses the file line by line and adds all distinct words in a Hash called importantWords. Now the map method has access to this Hash and it issues a map command only for those words that are found in importantWords.

Output Generation
Each of the nodes in the hadoop cluster in AWS EMR has some cores and for fast processing, each cores writes to its own output file. This option can be modified to write all output to a single file but it’s slower, so my output is contained in multiple output part files. The output files themselves are sorted by key.

The location of the output files for this assignment can be found in the 3 different folders: SingleWordCountOutput, DoubleWordOutput and DistributedWordCountOutput. Each of these folders contains 7 part files which contain the frequency for different distinct words.