AWS EMR & MRJob

Running MRJob on the EMR Cluster

Amazon Web Services

Compute



Virtual Servers in the Cloud



EC2 Container Service





Run and Manage Web Apps

Storage & Content Delivery



Scalable Storage in the Cloud



Global Content Delivery Network



Glacier Archive Storage in the Cloud

Import/Export Snowball Large Scale Data Transport

Storage Gateway
Hybrid Storage Integration

Database



Managed Relational Database Service



ElastiCache In-Memory Cache

Fast, Simple, Cost-Effective Data Warehousing

Managed Database Migration Service

Networking



Isolated Cloud Resources



Route 53 Scalable DNS and Domain Name Registration

Developer Tools



Store Code in Private Git Repositories



CodePipeline Release Software using Continuous Delivery

Management Tools



CloudWatch

Monitor Resources and Applications

CloudFormation

Create and Manage Resources with Templates

CloudTrail

Track User Activity and API Usage

Track Resource Inventory and Changes

OpsWorks Automate Operations with Chef

Service Catalog Create and Use Standardized Products

Trusted Advisor Optimize Performance and Security

Security & Identity



Directory Service Host and Manage Active Directory

Inspector PREVIEW Analyze Application Security

Filter Malicious Web Traffic

Certificate Manager Provision, Manage, and Deploy SSL/TLS Certificates

Analytics



EMR

Managed Hadoop Framework



Orchestration for Data-Driven Workflows

Elasticsearch Service Run and Scale Elasticsearch Clusters

Work with Real-Time Streaming Data



Internet of Things



AWS IoT

Connect Devices to the Cloud

Game Development



GameLift

Deploy and Scale Session-based Multiplayer Games

Mobile Services



Mobile Hub

Build, Test, and Monitor Mobile Apps



Cognito User Identity and App Data Synchronization









Push Notification Service

Application Services



API Gateway
Build, Deploy and Manage APIs



AppStream
Low Latency Application Streaming

CloudSearch

Managed Search Service

Elastic Transcoder Easy-to-Use Scalable Media Transcoding



Email Sending and Receiving Service



Message Queue Service



Workflow Service for Coordinating Application Components

Enterprise Applications



WorkSpaces

Desktops in the Cloud



WorkDocs

Secure Enterprise Storage and Sharing Service



WorkMail

Secure Email and Calendaring Service

Manually Create EMR Cluster in the AWS Console

Work with Real-Time Streaming Data

Build Smart Applications Quickly and Easily

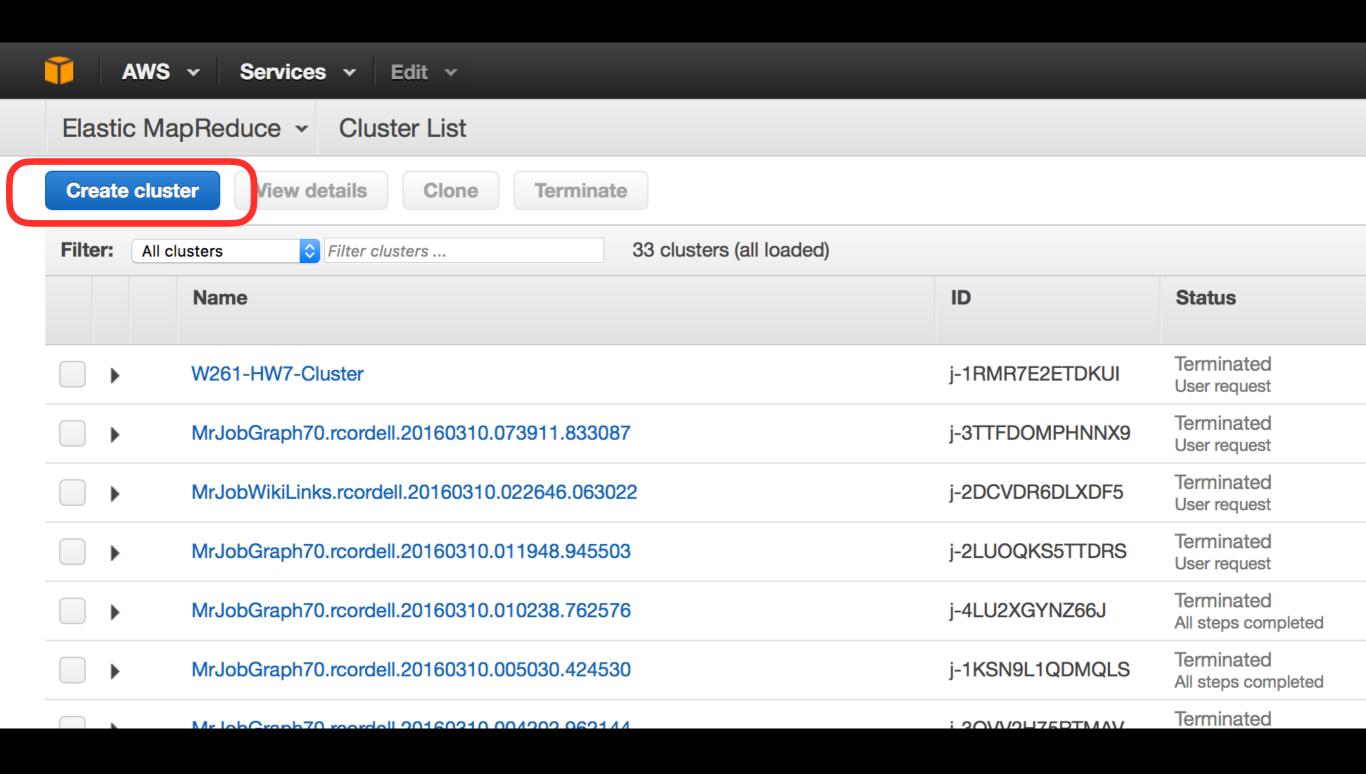
Machine Learning



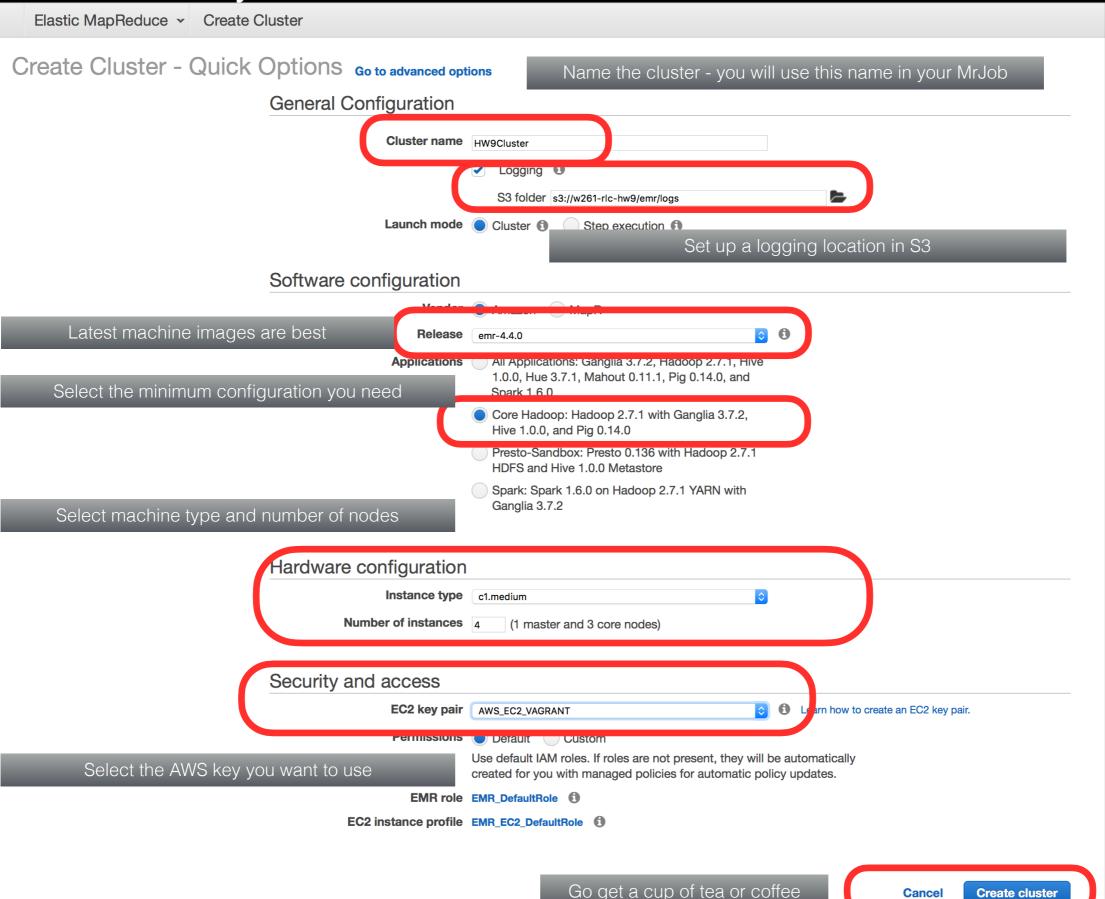
Internet of Things AWS IoT Store Code in Private Git Repositories Connect Devices to the Cloud Game Development GameLift Release Software using Continuous Delivery Deploy and Scale Session-based Multiplayer Games Mobile Services Mobile Hub Monitor Resources and Applications Build, Test, and Monitor Mobile Apps Cognito Create and Manage Resources with Templates User Identity and App Data Synchronization Device Farm Track User Activity and API Usage Test Android, FireOS, and iOS Apps on Real Devices in the Mobile Analytics Track Resource Inventory and Changes Collect, View and Export App Analytics Automate Operations with Chef Push Notification Service Create and Use Standardized Products **Application Services API** Gateway Optimize Performance and Security Build, Deploy and Manage APIs AppStream Low Latency Application Streaming Identity & Access Management CloudSearch Manage User Access and Encryption Keys Managed Search Service **Elastic Transcoder** Host and Manage Active Directory Easy-to-Use Scalable Media Transcoding Email Sending and Receiving Service SQS Message Queue Service ■ Certificate Manager Provision, Manage, and Deploy SSL/TLS Certificates Workflow Service for Coordinating Application Components **Enterprise Applications**



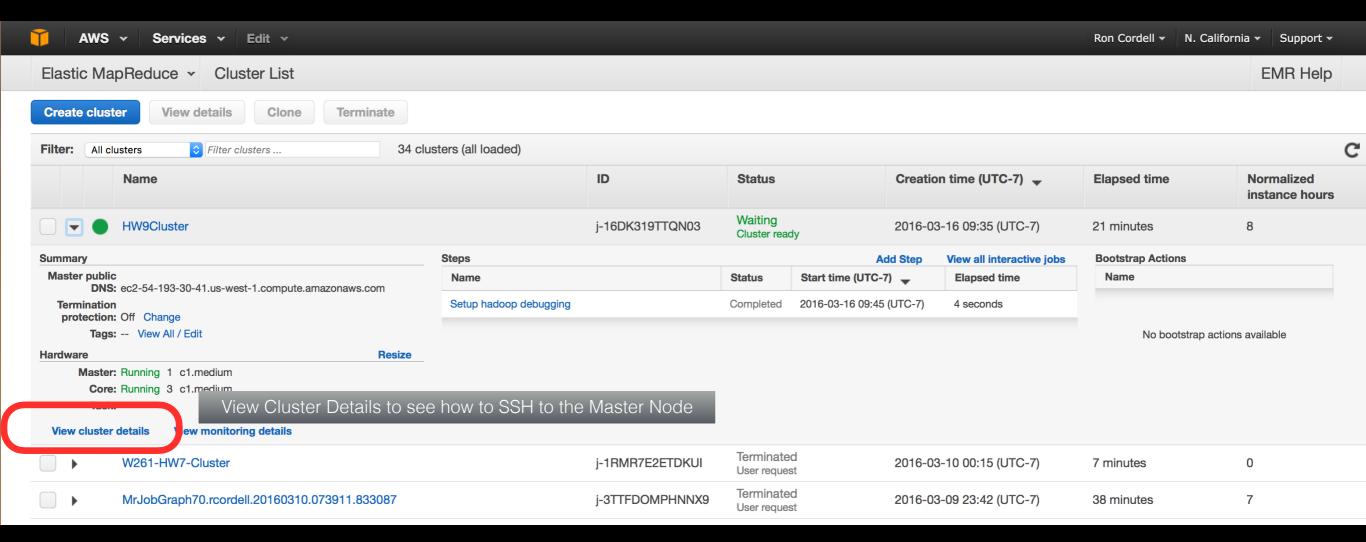
Manually Create EMR Cluster in the AWS Console



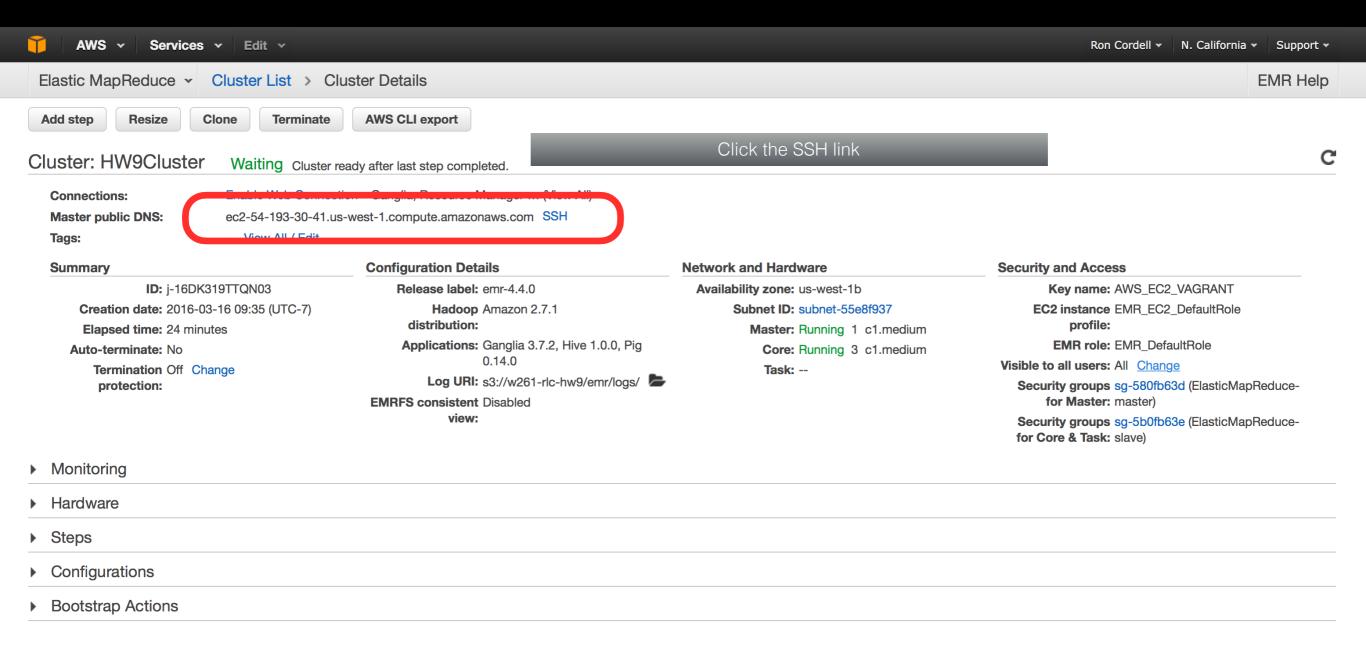
Manually Create EMR Cluster in the AWS Console



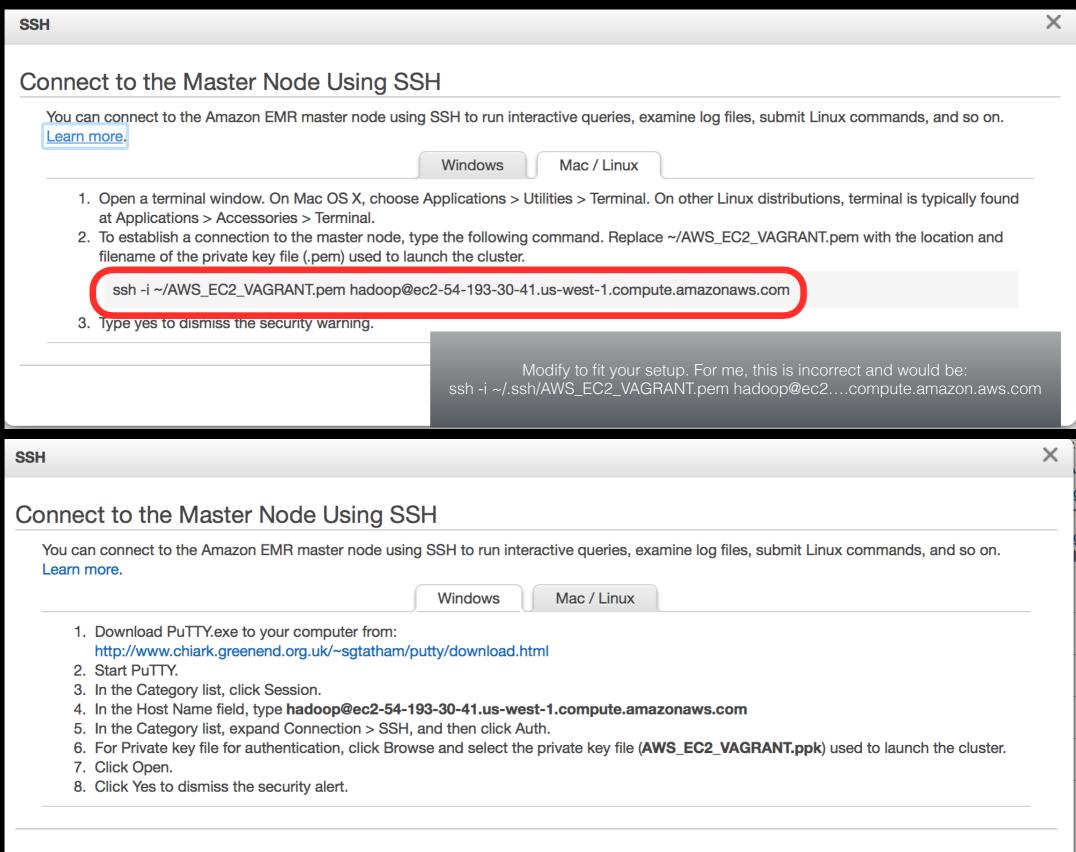
Cluster Ready - Expanded View of the Cluster in the Cluster List



SSH to the Cluster Master Node



SSH instructions - click the SSH link



We're In!

```
(W261env)rcordell@Rons-iMac-Retina:~/Documents/MIDS/W261/week09/HW9$ ssh -i ~/.ssh/AWS_EC2_VAGRANT.pem hadoop@ec2-54-193-30-41.us-west-1.compute.amazonaws.com
The authenticity of host 'ec2-54-193-30-41.us-west-1.compute.amazonaws.com (54.193.30.41)' can't be established.
ECDSA key fingerprint is SHA256:6sEckR8N5Wz0x9AlTgT/co2KVcPmKCsqbjqTYPctnVM.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'ec2-54-193-30-41.us-west-1.compute.amazonaws.com,54.193.30.41' (ECDSA) to the list of known hosts.
Last login: Wed Mar 16 16:47:37 2016
```

```
__| __| )
_| ( / Amazon Linux AMI
___|\__|
```

https://aws.amazon.com/amazon-linux-ami/2015.09-release-notes/5 package(s) needed for security, out of 15 available Run "sudo yum update" to apply all updates.

```
EEEEEEEEEEEEEEEEE MMMMMMM
                                 M::::::: M R:::::::::::R
EE:::::EEEEEEEEE:::E M:::::::M
                               M:::::::M R:::::RRRRRR:::::R
 E::::E
            EEEEE M:::::::M
                              M::::::: M RR::::R
                                                    R::::R
                 M::::::M:::M
                             M:::M:::::M
                                                    R::::R
 E:::::EEEEEEEEE M:::::M M:::M M:::M M::::M
                                          R:::RRRRRR::::R
                                          R:::::::RR
 E::::::E
                 M:::::M M:::M:::M M:::::M
 E::::EEEEEEEEE
                M:::::M
                         M:::::M M:::::M
                                          R:::RRRRRR::::R
 E::::E
                 M:::::M
                          M:::M
                                 M:::::M
                                          R:::R
                                                    R::::R
 E::::E
            EEEEE M:::::M
                           MMM
                                  M:::::M
                                          R:::R
                                                    R::::R
EE:::::EEEEEEEEE::::E M:::::M
                                  M:::::M
                                          R:::R
                                                    R::::R
E:::::: M:::::M
                                  M:::::M RR::::R
                                                    R::::R
EEEEEEEEEEEEEEEEE MMMMMM
                                  MMMMMM RRRRRR
                                                    RRRRRR
```

[hadoop@ip-172-31-22-83 ~]\$ [

Install Python Packages (MrJob, etc)

```
A HEW VELSION OF PLP IS AVAILABLE FOR ADMITTORA, THIPLIER WITH THOUGH.
[hadoop@ip-172-31-22-83 ~]$ sudo pip install mrjob
You are using pip version 6.1.1. however version 8.1.0 is available
You should consider upgrading via the 'pip install --upgrade pip' command.
Collecting mrjob
 Downloading mrjob-0.4.6-py2-none-any.whl (244kB)
    100% I
                                          II 245kB 655kB/s
Requirement already satisfied (use --upgrade to upgrade): PyYAML>=3.08 in /usr/local/lib64/python2.7/site-packages (from mrjob)
Requirement already satisfied (use --upgrade to upgrade): boto>=2.6.0 in /usr/lib/python2.7/dist-packages (from mrjob)
Requirement already satisfied (use --upgrade to upgrade): simplejson>=2.0.9 in /usr/local/lib64/python2.7/site-packages (from mrjob)
Collecting filechunkio (from mrjob)
 Downloading filechunkio-1.6.tar.gz
Installing collected packages: filechunkio, mrjob
 Running setup.py install for filechunkio
Successfully installed filechunkio-1.6 mrjob-0.4.6
[hadoop@ip-172-31-22-83 ~]$
```

This is with the latest AMI version - it already has Python 2.7.10 and pip; earlier version may not. You can install whatever you need using sudo.

NOTE: you are logged in as the hadoop user so HDFS files will be under hdfs://users/hadoop

Put your code on the Master

SCP

scp -i ~/.ssh/your_aws_key.pem project/mrjob.py hadoop@ec2-52-53-232-230.us-west-1.compute.amazonaws.com:~/.

Copy-Paste

- ssh to the master,
- open an editor,
- copy your code from your local editor
- paste you code in the remote editor

git

- sudo yum install git
- git clone <my repository>
- git pull <my remote repository>

ssh to the master and submit your job using hadoop runner

python word_count.py -r hadoop --hadoop-home /usr/lib/hadoop enronemail_1h.txt

```
no configs found; falling back on auto-configuration
no configs found; falling back on auto-configuration
creating tmp directory /tmp/word_count.hadoop.20160318.050129.032471
writing wrapper script to /tmp/word_count.hadoop.20160318.050129.032471/setup-wrapper.sh
Using Hadoop version 2.7.1
Copying local files into hdfs://user/hadoop/tmp/mrjob/word_count.hadoop.20160318.050129.032471/files/
PLEASE NOTE: Starting in mrjob v0.5.0, protocols will be strict by default. It's recommended you run your job with --strict-protocols or set
up mrjob.conf as described at https://pythonhosted.org/mrjob/whats-new.html#ready-for-strict-protocols
HADOOP: packageJobJar: [] [/usr/lib/hadoop/hadoop-streaming-2.7.1-amzn-1.jar] /tmp/streamjob2109238968681413244.jar tmpDir=null
HADOOP: Connecting to ResourceManager at ip-172-31-1-250.us-west-1.compute.internal/172.31.1.250:8032
HADOOP: Connecting to ResourceManager at ip-172-31-1-250.us-west-1.compute.internal/172.31.1.250:8032
HADOOP: MetricsConfigRecord disabledInCluster: false instanceEngineCycleSec: 60 clusterEngineCycleSec: 60 disableClusterEngine: false
maxMemoryMb: 3072 maxInstanceCount: 500 lastModified: 1458275059806
HADOOP: Created MetricsSaver j-4YUGOAB3MAN3:i-6330f9d6:RunJar:19930 period:60 /mnt/var/em/raw/i-6330f9d6_20160318_RunJar_19930_raw.bin
HADOOP: Loaded native gpl library
HADOOP: Successfully loaded & initialized native-lzo library [hadoop-lzo rev 72c57a1c06c471da40827b432ecff0de6a5c6dcc]
HADOOP: Total input paths to process: 1
HADOOP: number of splits:4
HADOOP: Submitting tokens for job: job_1458275041561_0001
HADOOP: Submitted application application_1458275041561_0001
HADOOP: The url to track the job: http://ip-172-31-1-250.us-west-1.compute.internal:20888/proxy/application_1458275041561_0001/
HAD00P: Running job: job_1458275041561_0001
HADOOP: Job job_1458275041561_0001 running in uber mode : false
HADOOP: map 0% reduce 0%
HADOOP: map 25% reduce 0%
HADOOP:
        map 83% reduce 0%
        map 100% reduce 0%
HADOOP:
HADOOP:
        map 100% reduce 33%
HADOOP:
        map 100% reduce 67%
HADOOP:
        map 100% reduce 100%
HADOOP: Job job_1458275041561_0001 completed successfully
HADOOP: Counters: 51
HADOOP:
           File System Counters
```