AWS EMR & MRJob

Running MRJob on the EMR Cluster

AWS Services Edit

Amazon Web Services

Compute

- **EC2** — Virtual Servers in the Cloud
- **EC2 Container Service** — Run and Manage Docker Containers
- **Elastic Beanstalk** — Run and Manage Web Apps
- **Lambda** — Run Code in Response to Events

Storage & Content Delivery

- **S3** — Scalable Storage in the Cloud
- **CloudFront** — Global Content Delivery Network
- **Elastic File System** PREVIEW — Fully Managed File System for EC2
- **Glacier** — Archive Storage in the Cloud
- **Import/Export Snowball** — Large Scale Data Transport
- **Storage Gateway** — Hybrid Storage Integration

Database

- **RDS** — Managed Relational Database Service
- **DynamoDB** — Managed NoSQL Database
- **ElastiCache** — In-Memory Cache
- **Redshift** — Fast, Simple, Cost-Effective Data Warehousing
- **DMS** — Managed Database Migration Service

Networking

- **VPC** — Isolated Cloud Resources
- **Direct Connect** — Dedicated Network Connection to AWS
- **Route 53** — Scalable DNS and Domain Name Registration

Developer Tools

- **CodeCommit** — Store Code in Private Git Repositories
- **CodeDeploy** — Automate Code Deployments
- **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
- **Config** — 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

- **Identity & Access Management** — Manage User Access and Encryption Keys
- **Directory Service** — Host and Manage Active Directory
- **Inspector** PREVIEW — Analyze Application Security
- **WAF** — Filter Malicious Web Traffic
- **Certificate Manager** — Provision, Manage, and Deploy SSL/TLS Certificates

Analytics

- **EMR** — Managed Hadoop Framework
- **Data Pipeline** — Orchestration for Data-Driven Workflows
- **Elasticsearch Service** — Run and Scale Elasticsearch Clusters
- **Kinesis** — Work with Real-Time Streaming Data
- **Machine Learning** — Build Smart Applications Quickly and Easily

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
- **Device Farm** — Test Android, FireOS, and iOS Apps on Real Devices in the Cloud
- **Mobile Analytics** — Collect, View and Export App Analytics
- **SNS** — 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
- **SES** — Email Sending and Receiving Service
- **SQS** — Message Queue Service
- **SWF** — 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

AWS | Services | Edit

Amazon Web Services

Compute
- EC2 — Virtual Servers in the Cloud
- EC2 Container Service — Run and Manage Docker Containers
- Elastic Beanstalk — Run and Manage Web Apps
- Lambda — Run Code in Response to Events

Storage & Content Delivery
- S3 — Scalable Storage in the Cloud
- CloudFront — Global Content Delivery Network
- Elastic File System PREVIEW — Fully Managed File System for EC2
- Glacier — Archive Storage in the Cloud
- Import/Export Snowball — Large Scale Data Transport
- Storage Gateway — Hybrid Storage Integration

Database
- RDS — Managed Relational Database Service
- DynamoDB — Managed NoSQL Database
- ElastiCache — In-Memory Cache
- Redshift — Fast, Simple, Cost-Effective Data Warehousing
- DMS — Managed Database Migration Service

Networking
- VPC — Isolated Cloud Resources
- Direct Connect — Dedicated Network Connection to AWS
- Route 53 — Scalable DNS and Domain Name Registration

Developer Tools
- CodeCommit — Store Code in Private Git Repositories
- CodeDeploy — Automate Code Deployments
- 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
- Config — 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
- Identity & Access Management — Manage User Access and Encryption Keys
- Directory Service — Host and Manage Active Directory
- Inspector PREVIEW — Analyze Application Security
- WAF — Filter Malicious Web Traffic
- Certificate Manager — Provision, Manage, and Deploy SSL/TLS Certificates

Analytics
- EMR — Managed Hadoop Framework
- Data Pipeline — Orchestration for Data-Driven Workflows
- Elasticsearch Service — Run and Scale Elasticsearch Clusters
- Kinesis — Work with Real-Time Streaming Data
- Machine Learning — Build Smart Applications Quickly and Easily

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
- Device Farm — Test Android, FireOS, and iOS Apps on Real Devices in the Cloud
- Mobile Analytics — Collect, View and Export App Analytics
- SNS — 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
- SES — Email Sending and Receiving Service
- SQS — Message Queue Service
- SWF — 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

AWS | Services | Edit

Elastic MapReduce | Cluster List

Create cluster | View details | Clone | Terminate

Filter: All clusters | Filter clusters ... | 33 clusters (all loaded)

Name	ID	Status
W261-HW7-Cluster	j-1RMR7E2ETDKUI	Terminated User request
MrJobGraph70.rcordell.20160310.073911.833087	j-3TTFDOMPHNNX9	Terminated User request
MrJobWikiLinks.rcordell.20160310.022646.063022	j-2DCVDR6DLXDF5	Terminated User request
MrJobGraph70.rcordell.20160310.011948.945503	j-2LUOQKS5TTDRS	Terminated User request
MrJobGraph70.rcordell.20160310.010238.762576	j-4LU2XGYNZ66J	Terminated All steps completed
MrJobGraph70.rcordell.20160310.005030.424530	j-1KSN9L1QDMQLS	Terminated All steps completed
MrJobGraph70.rcordell.20160310.004202.962144	j-3QVV2HZ5RTMAV	Terminated

Manually Create EMR Cluster in the AWS Console

Elastic MapReduce › Create Cluster

Create Cluster - Quick Options

Go to advanced options

Name the cluster - you will use this name in your MrJob

General Configuration

Cluster name HW9Cluster

Logging

S3 folder s3://w261-rlc-hw9/emr/logs

Launch mode Cluster Step execution

Set up a logging location in S3

Software configuration

Vendor Amazon MapR

Latest machine images are best

Release emr-4.4.0

Applications All Applications: Ganglia 3.7.2, Hadoop 2.7.1, Hive 1.0.0, Hue 3.7.1, Mahout 0.11.1, Pig 0.14.0, and Spark 1.6.0

Select the minimum configuration you need

Core Hadoop: Hadoop 2.7.1 with Ganglia 3.7.2, Hive 1.0.0, and Pig 0.14.0

Presto-Sandbox: Presto 0.136 with Hadoop 2.7.1 HDFS and Hive 1.0.0 Metastore

Spark: Spark 1.6.0 on Hadoop 2.7.1 YARN with Ganglia 3.7.2

Select machine type and number of nodes

Hardware configuration

Instance type c1.medium

Number of instances 4 (1 master and 3 core nodes)

Security and access

EC2 key pair AWS_EC2_VAGRANT Learn how to create an EC2 key pair.

Permissions Default Custom

Select the AWS key you want to use

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

EMR role EMR_DefaultRole

EC2 instance profile EMR_EC2_DefaultRole

Go get a cup of tea or coffee

Cancel Create cluster

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

AWS | Services | Edit | Ron Cordell | N. California | Support

Elastic MapReduce | Cluster List | EMR Help

Create cluster | View details | Clone | Terminate

Filter: All clusters | Filter clusters ... | 34 clusters (all loaded)

	Name	ID	Status	Creation time (UTC-7)	Elapsed time	Normalized instance hours
	HW9Cluster	j-16DK319TTQN03	Waiting Cluster ready	2016-03-16 09:35 (UTC-7)	21 minutes	8

Summary

Master public DNS: ec2-54-193-30-41.us-west-1.compute.amazonaws.com

Termination protection: Off Change

Tags: -- View All / Edit

Hardware Resize

Master: Running 1 c1.medium

Core: Running 3 c1.medium

View Cluster Details to see how to SSH to the Master Node

View cluster details | View monitoring details

Steps Add Step | View all interactive jobs

Name	Status	Start time (UTC-7)	Elapsed time
Setup hadoop debugging	Completed	2016-03-16 09:45 (UTC-7)	4 seconds

Bootstrap Actions

Name

No bootstrap actions available

W261-HW7-Cluster	j-1RMR7E2ETDKUI	Terminated User request	2016-03-10 00:15 (UTC-7)	7 minutes	0
MrJobGraph70.rcordell.20160310.073911.833087	j-3TTFDOMPHNNX9	Terminated User request	2016-03-09 23:42 (UTC-7)	38 minutes	7

SSH to the Cluster Master Node

AWS | Services | Edit | Ron Cordell | N. California | Support

Elastic MapReduce | Cluster List > Cluster Details | EMR Help

Add step | Resize | Clone | Terminate | AWS CLI export

Click the SSH link

Cluster: HW9Cluster Waiting Cluster ready after last step completed.

Connections: Enable Web Connection | Ganglia, Resource Manager UI (View All)

Master public DNS: ec2-54-193-30-41.us-west-1.compute.amazonaws.com SSH

Tags: View All / Edit

Summary

ID: j-16DK319TTQN03
Creation date: 2016-03-16 09:35 (UTC-7)
Elapsed time: 24 minutes
Auto-terminate: No
Termination protection: Off Change

Configuration Details

Release label: emr-4.4.0
Hadoop distribution: Amazon 2.7.1
Applications: Ganglia 3.7.2, Hive 1.0.0, Pig 0.14.0
Log URI: s3://w261-rlc-hw9/emr/logs/
EMRFS consistent view: Disabled

Network and Hardware

Availability zone: us-west-1b
Subnet ID: subnet-55e8f937
Master: Running 1 c1.medium
Core: Running 3 c1.medium
Task: --

Security and Access

Key name: AWS_EC2_VAGRANT
EC2 instance profile: EMR_EC2_DefaultRole
EMR role: EMR_DefaultRole
Visible to all users: All Change
Security groups for Master: sg-580fb63d (ElasticMapReduce-master)
Security groups for Core & Task: sg-5b0fb63e (ElasticMapReduce-slave)

- Monitoring
- Hardware
- Steps
- Configurations
- Bootstrap Actions

SSH instructions - click the SSH link

SSH

Connect to the Master Node Using SSH

You can connect to the Amazon EMR master node using SSH to run interactive queries, examine log files, submit Linux commands, and so on. Learn more.

Windows | Mac / Linux

1. Open a terminal window. On Mac OS X, choose Applications > Utilities > Terminal. On other Linux distributions, terminal is typically found at Applications > Accessories > Terminal.
2. To establish a connection to the master node, type the following command. Replace ~/AWS_EC2_VAGRANT.pem with the location and filename of the private key file (.pem) used to launch the cluster.

```
ssh -i ~/AWS_EC2_VAGRANT.pem hadoop@ec2-54-193-30-41.us-west-1.compute.amazonaws.com
```

3. Type yes to dismiss the security warning.

Modify to fit your setup. For me, this is incorrect and would be:
ssh -i ~/.ssh/AWS_EC2_VAGRANT.pem hadoop@ec2….compute.amazon.aws.com

SSH

Connect to the Master Node Using SSH

You can connect to the Amazon EMR master node using SSH to run interactive queries, examine log files, submit Linux commands, and so on. Learn more.

Windows | Mac / Linux

1. Download PuTTY.exe to your computer from: http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html
2. Start PuTTY.
3. In the Category list, click Session.
4. In the Host Name field, type **hadoop@ec2-54-193-30-41.us-west-1.compute.amazonaws.com**
5. In the Category list, expand Connection > SSH, and then click Auth.
6. For Private key file for authentication, click Browse and select the private key file (**AWS_EC2_VAGRANT.ppk**) used to launch the cluster.
7. Click Open.
8. Click Yes to dismiss the security alert.

Close

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 SHA256:6sEckR8N5WzOx9AlTgT/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.

EEEEEEEEEEEEEEEEEEEE MMMMMMMM           MMMMMMMM RRRRRRRRRRRRRRR
E::::::::::::::::::E M::::::M         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
  E::::E             M::::::M::::M   M::::M::::::M   R:::R      R::::R
  E::::::EEEEEEEEEE   M::::::M M::::M M::::M M::::::M   R:::RRRRRR:::::R
  E::::::::::::::::E   M::::::M  M::::M::::M  M::::::M   R::::::::::::RR
  E::::::EEEEEEEEEE   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::::::EEEEEEEE::::E M::::::M             M::::::M   R:::R      R::::R
E::::::::::::::::::E M::::::M             M::::::M RR::::R      R::::R
EEEEEEEEEEEEEEEEEEEE MMMMMMM             MMMMMMM RRRRRRR      RRRRRR

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

Install Python Packages (MrJob, etc)

```
[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% |████████████████████████████████| 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/
HADOOP: 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%
HADOOP:  map 100% reduce 0%
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
```