Amazon Elastic MapReduce (Version 2009-03-31)

Quick Reference Card (page 1)

Create Job Flow

JAR `$ ./elastic-mapreduce --create --alive --input s3n://mybucket/input --output s3n://mybucket/output --log-uri s3n://my-example-bucket/logs`
Stream `$ ./elastic-mapreduce --create --alive --stream --input s3n://mybucket/input --output s3n://mybucket/output --log-uri s3n://my-example-bucket/logs`
Pig `$ ./elastic-mapreduce --create --alive --name "Hive Test" --input s3n://mybucket/input --output s3n://mybucket/output --num-instances 5 --instance-type c1.medium`
Hive `$ ./elastic-mapreduce --create --alive --name "Hive Test" --input s3n://mybucket/input --output s3n://mybucket/output --num-instances 5 --instance-type c1.medium`

Add a Job Flow Step

JAR `$ ./elastic-mapreduce -j j-JobFlowId`
Stream `$ ./elastic-mapreduce -j j-JobFlowId --streaming`

Terminate a Job Flow

$ ./elastic-mapreduce --jobflow j-*JobId* terminate

Get Information About a Job Flow

$ ./elastic-mapreduce --describe –jobflow [*JobFlowId*]

List Job Flows

$ ./elastic-mapreduce --list [–active] [-running] [-terminated]

SSH Into a Master Node

ssh –i [keyfile.pem] hadoop@[EC2_master_node_DNS]

Use Additional Files and Libraries With the Mapper or Reducer

-cache s3n://bucket/path_to_executable#local_path

Adding Files to the Distributed Cache

Single file: --cache s3n://my_bucket/sample_dataset.dat#sample_dataset_cached.dat
Archive file: --cache-archive s3n://my_bucket/sample_dataset.tgz#sample_dataset_cached
Multiple files: --cache s3n://my_bucket/sample_binary.bin#sample_binary_cached.bin
--cache s3n://my_other_bucket/sample_binary1.bin#sample_binary1_cached.bin
--cache-archive s3n://my_bucket/sample_dataset.tgz#sample_dataset_cached

Enable Output Data Compression Using the Console and a Streaming Job Flow

-jobconf mapred.output.compress=true

Hadoop File Locations

Failure logs: /mnt/var/log/hadoop/ on each instance, or daemons/<instance ID>/ on Amazon S3
UI for MapReduce job tracker(s): http://[master_dns_name]:9100/
UI for HDFS name node(s): http://[master_dns_name]:9101/
Temporary files: /mnt/var/lib/hadoop/tmp
Cache: /mnt/var/lib/hadoop/mapred/taskTracker/archive/

Credential File Fields

"access-id": "181sAmPlE18ABA",
"private-key": "ABA/A1SaMpLejla/AS1a",
"key-pair": "my-key",
"key-pair-file": "/home/user/keys/mykey",
"region": "us-east-1",
"log-uri": "s3://mybucket/emr-logs"

Useful Links

Forum: http://developer.amazonwebservices.com/connect/forum.jspa?forumID=52
Resource Center: http://developer.amazonwebservices.com/connect/kbcategory.jspa?categoryID=259
Articles & Tutorials: http://developer.amazonwebservices.com/connect/kbcategory.jspa?categoryID=265
Release Notes: http://developer.amazonwebservices.com/connect/kbcategory.jspa?categoryID=260
Code samples and libraries: http://developer.amazonwebservices.com/connect/kbcategory.jspa?categoryID=262
Developer Tools: http://developer.amazonwebservices.com/connect/kbcategory.jspa?categoryID=266
Technical Documentation: http://developer.amazonwebservices.com/connect/kbcategory.jspa?categoryID=261
WSDL Location: http://elasticmapreduce.amazonaws.com/doc/2009-03-31/ElasticMapReduce.wsdl
CLI Download: http://developer.amazonwebservices.com/connect/entry.jspa?externalID=2264&categoryID=273

Log File Locations

Logs/<jobid>/jobs/
Logs/<jobid>/node/
Logs/<jobid>/steps/
Logs/<jobid>/steps/<stepNumber>/syslog
Logs/<jobid>/steps/<stepNumber>/stdout
Logs/<jobid>/steps/<stepNumber>/controller
Logs/<jobid>/steps/<stepNumber>/stderr
Logs/<jobid>/task-attempts/

Revised: 10/1/09

Amazon Elastic MapReduce (Version 2009-03-31)

Quick Reference Card (page 2)

Command Line Commands

--create	Create a new job flow
--name	NAME Name of the job flow
--alive	Create a job flow that stays running even though it has executed all its steps
--JAR JAR	Add a step that executes a jar
--main-class MAIN_CLASS	Specify main class for the JAR
--arg ARG	Specify an argument to a jar step or a streaming step
--jobconf JOB_CONF	Specify jobconf arguments to pass to streaming
--stream	Add a step that performs Hadoop streaming
--step_name STEP_NAME	Add a step to the work flow
--input INPUT	Input to the steps, e.g. s3://mybucket/input
--output OUTPUT	The output to the steps, e.g. s3://mybucket/output
--mapper MAPPER	The mapper program or class
--cache CACHE_FILE	A file to load into the cache, e.g. s3://mybucket/sample.py#sample.py
--reduce REDUCER	The reducer program or class
--list, --describe	List all job flows created in the last 2 days
--active	List running, starting, or shutting down job flows
--state STATE	List job flows in STATE
--all	List all job flows in the last 2 months
--nosteps	Do not list steps when listing jobs
-n, --max-results MAX_RESULTS	Maximum number of results to list
--terminate	Terminate the job flow
--num-instances NUM_INSTANCES	Number of instances in the job flow
--key_pair KEYPAIR	The type of the instances to launch
--log_uri LOG_URI	Location in Amazon S3 to store logs from the job flow, e.g. s3://mybucket/logs
--endpoint ENDPOINT	Specify the web service endpoint to talk to
-j, --jobflow JOB_FLOW_ID	Job flow ID
--instance-type INSTANCE_TYPE	The type of the instances to launch
-c CREDENTIALS_FILE	File containing access_id and secret key
--credentials	
-a, --access_id ACCESS_ID	AWS Access ID
-k, --secret_key SECRET_KEY	AWS Secret Key
-v, --verbose	Turn on verbose logging of program interaction
--debug	Print stack traces when exceptions occur
--version	Print a version string
-h, --help	Show help message

Hive Commands

hive [<-f filename>|<-e query-string>] [-S] [-hiveconf x=y]* [-d Var=Value]*

-e '*query string*'	SQL from command line (interactive)
-f *filename*	SQL from file
-d Var=*Value*	Passes value into Hive script as ${Var}
-S	Silent mode in interactive shell where only data is emitted
-hiveconf *x=y*	Use this to set Hive or Hadoop configuration variables
add FILE *value value*	Adds a file to the list of resources
! *command*	Execute a shell command from Hive shell
dfs *dfs command*	Execute dfs command from Hive shell
list FILE	List all the resources already added
list FILE *value*	Check given resources are already added or not
query string	Execute Hive query and send results to stdout
Quit	Exit interactive shell
set *key=value*	Set configuration variable
Set	List configuration variables overridden by user or Hive
set -v	List all Hadoop and Hive configuration variables

Pig Relational Operators

COGROUP	alias BY field_alias [INNER | OUTER] , alias BY field_alias [INNER | OUTER] [PARALLEL n] ;
CROSS	alias, alias [, alias …] [PARALLEL n];
DISTINCT	alias [PARALLEL n];
DUMP	alias;
FILTER	alias BY expression;
FOREACH	{ gen_blk | nested_gen_blk } [AS schema];
GROUP	alias { [ALL] | [BY {[field_alias [, field_alias]] | * | [expression]] } [PARALLEL n];
JOIN	alias BY field_alias, alias BY field_alias [, alias BY field_alias …] [USING "replicated"] [PARALLEL n];
LIMIT	alias n;
LOAD	'data' [USING function] [AS schema];
ORDER	alias BY { * [ASC|DESC] | field_alias [ASC|DESC] [, field_alias [ASC|DESC] …] } [PARALLEL n];
SAMPLE	alias size;
SPLIT	alias INTO alias IF expression, alias IF expression [, alias IF expression …];
STORE	alias INTO 'directory' [USING function];
STREAM	alias [, alias …] THROUGH {'command' | cmd_alias } [AS schema] ;
UNION	alias, alias [, alias …];

CLI Configuration

```
:endpoint => "https://elasticmapreduce.amazonaws.com",
:ca_file => File.join(File.dirname(__FILE__), "cacert.pem"),
:aws_access_key => [my_access_id],
:aws_secret_key => [my_secret_key],
:signature_algorithm => :V2
```