



# Masterclass

....

*Elastic MapReduce*

Ian Massingham – Technical Evangelist

 @IanMmmm

# Masterclass

A technical deep dive beyond the basics

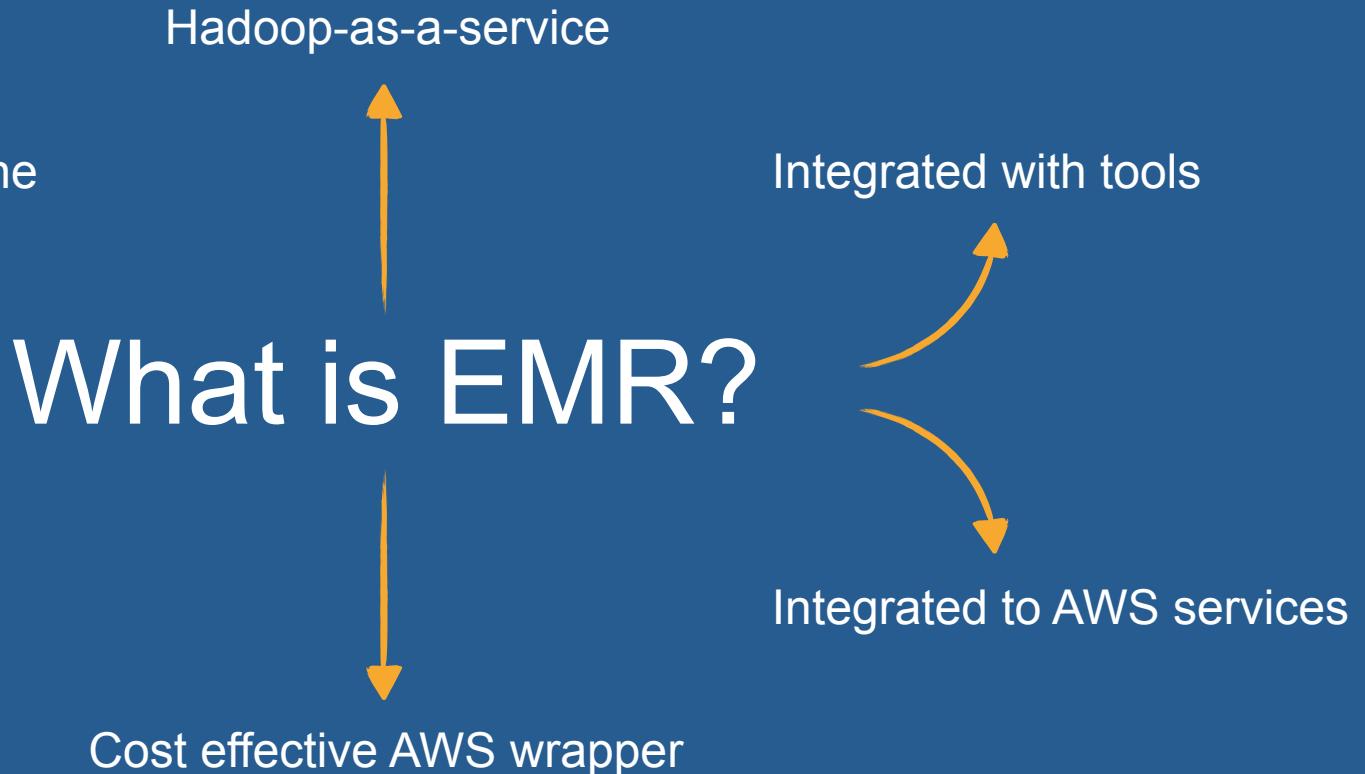
Help educate you on how to get the best from AWS technologies

Show you how things work and how to get things done

Broaden your knowledge in ~45 mins

# Amazon Elastic MapReduce

A key tool in the toolbox to help with ‘Big Data’ challenges  
Makes possible analytics processes previously not feasible  
Cost effective when leveraged with EC2 spot market  
Broad ecosystem of tools to handle specific use cases



A framework ◀

Splits data into pieces ◀

Lets processing occur ◀

Gathers the results ◀

Very large  
click log  
(e.g TBs)

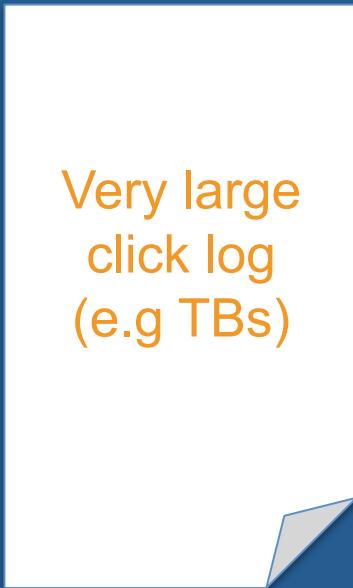
Lots of actions by  
John Smith



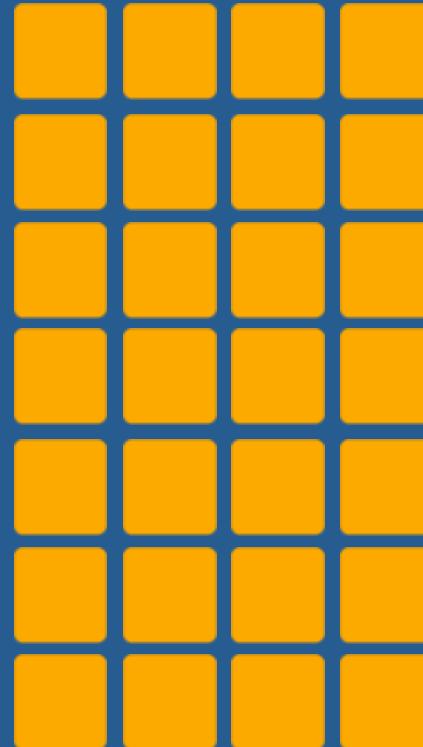
Very large  
click log  
(e.g TBs)



Lots of actions by  
John Smith



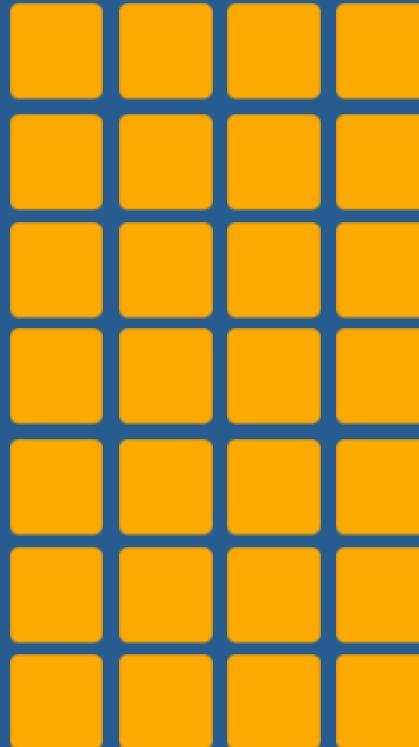
Split the log  
into many  
small pieces



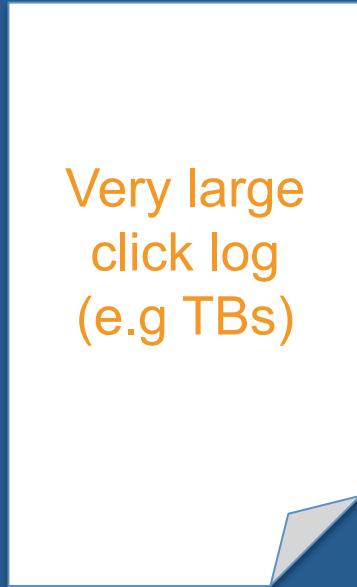


Split the log  
into many  
small pieces

Lots of actions by  
John Smith



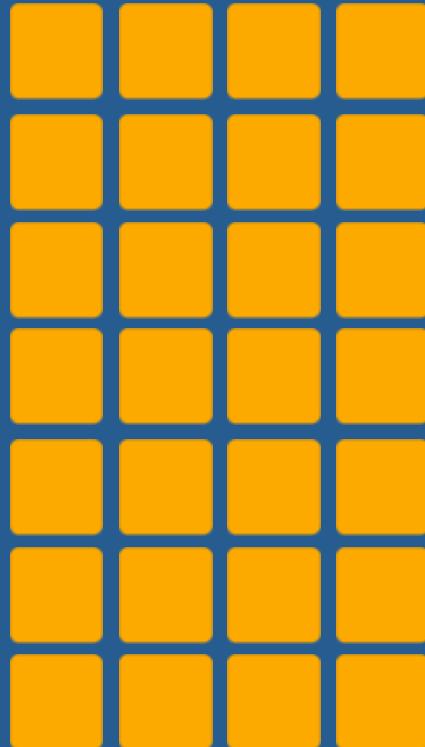
Process in an  
EMR cluster



Lots of actions by  
John Smith



Split the log  
into many  
small pieces



Process in an  
EMR cluster



Aggregate  
the results  
from all the  
nodes

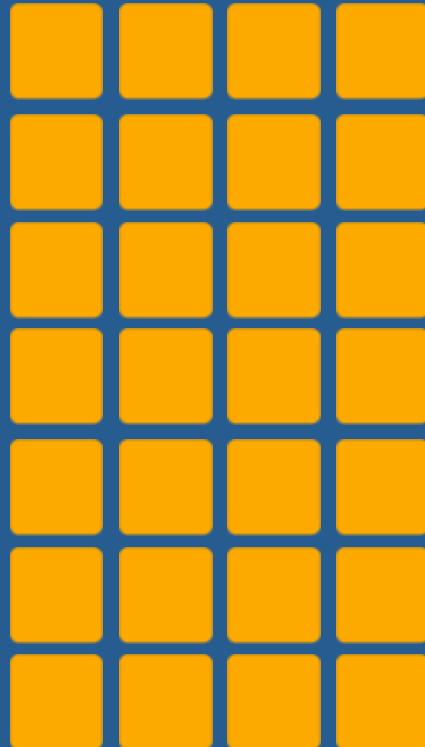
Very large  
click log  
(e.g TBs)



Lots of actions by  
John Smith



Split the log  
into many  
small pieces



Process in an  
EMR cluster



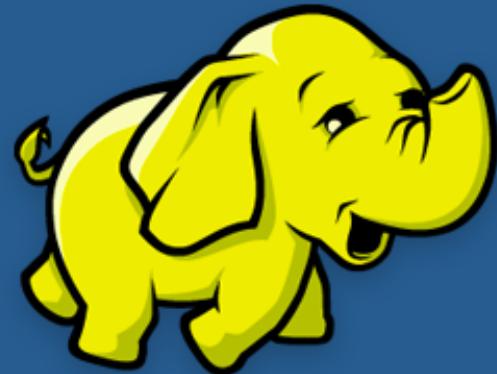
Aggregate  
the results  
from all the  
nodes

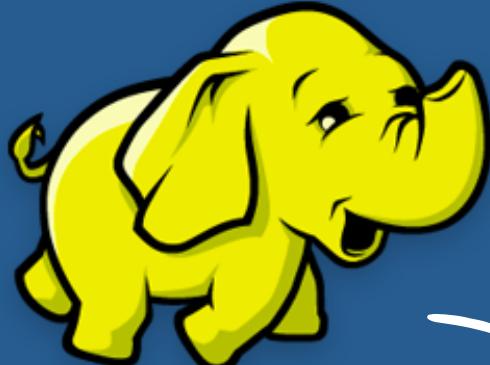
What  
John  
Smith  
did

Very large  
click log  
(e.g TBs)

Insight in a fraction of the time

What  
John  
Smith  
did





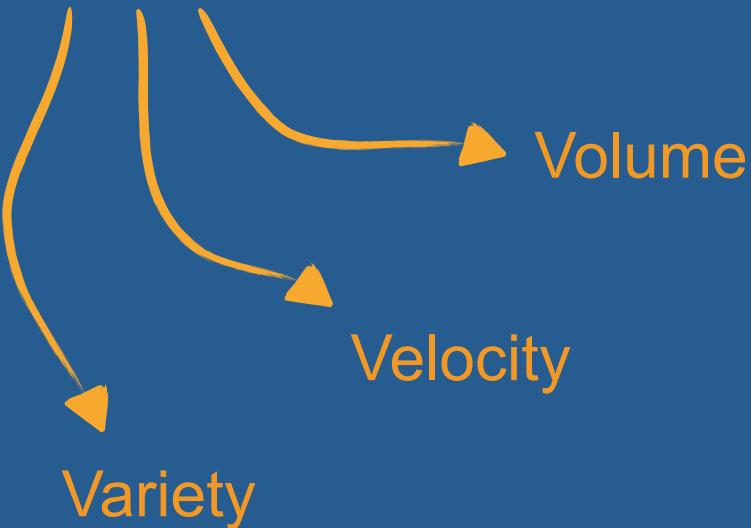
HDFS  
Reliable storage



MapReduce  
Data analysis

# Big Data != Hadoop

# Big Data != Hadoop



# Big Data != Hadoop



When you need to innovate to collect, store, analyze, and manage your data

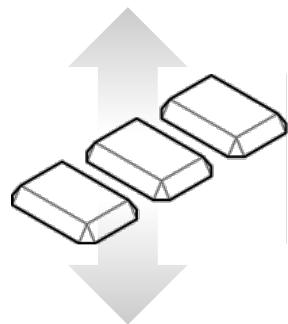
# Storage → Big Data → Compute

*How do you get your data into AWS?*



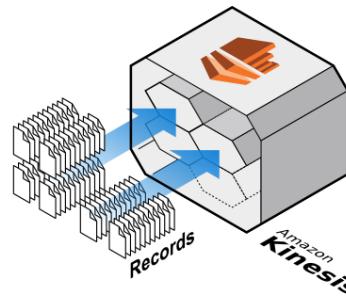
## AWS Direct Connect

Dedicated low latency  
bandwidth



## AWS Import/Export

Physical media shipping



## Amazon Kinesis

Highly scalable stream  
processing for event/record  
buffering

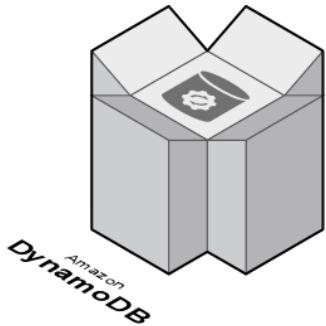


## Amazon Storage Gateway

Sync local storage to the cloud

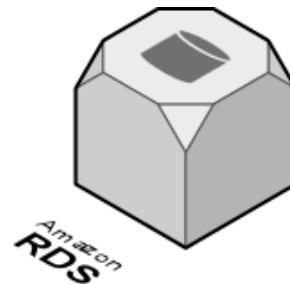
# Storage → Big Data → Compute

*Where do you put your data once it's in AWS?*



## DynamoDB

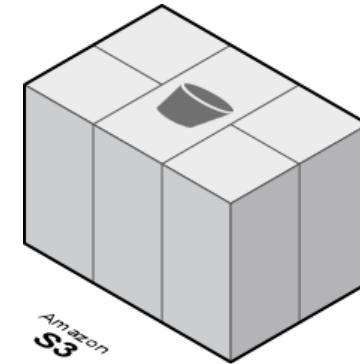
NoSQL, Schemaless,  
Provisioned throughput  
database



## Relational Database

### Service

Fully managed database  
(*PostgreSQL, MySQL,  
Oracle, MSSQL*)



## S3

Object datastore up to 5TB  
per object  
99.999999999% durability

# Elastic MapReduce

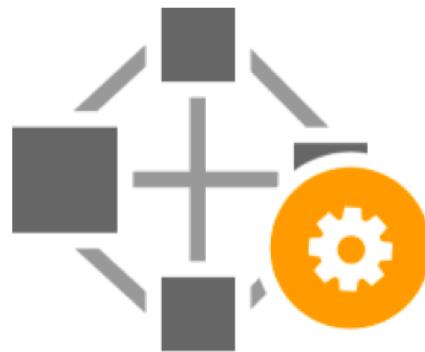
Getting started, tools & operating modes

## Upload



Upload your data and processing application to S3.

## Create



Configure and create your cluster by specifying data inputs, outputs, cluster size, security settings, etc.

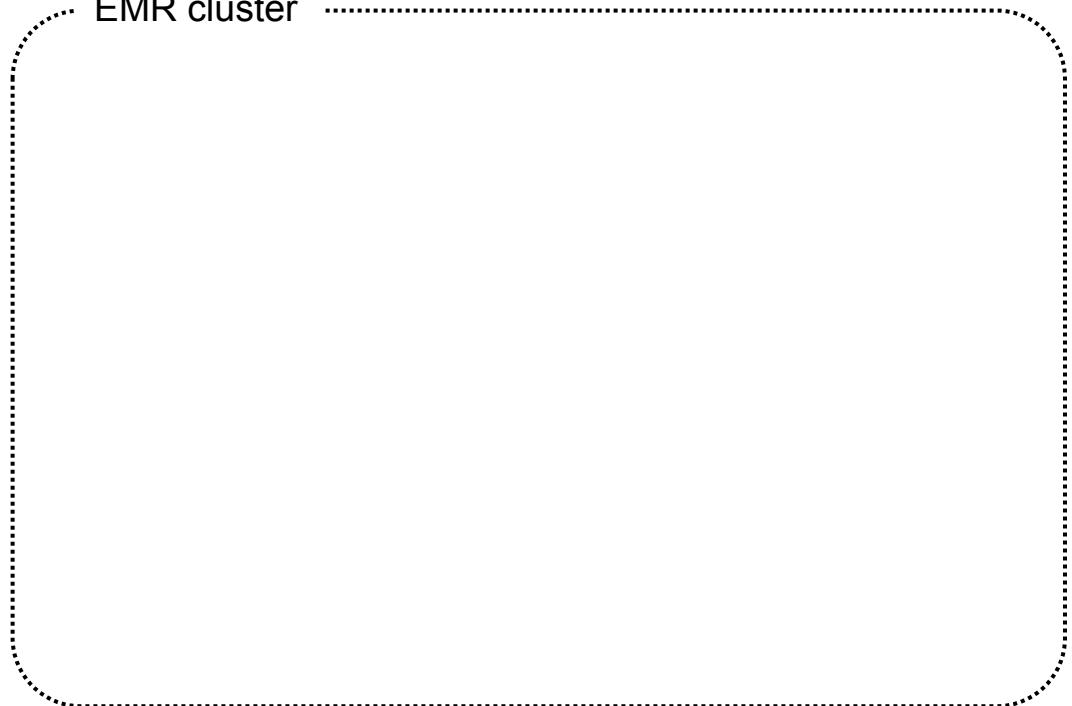
## Monitor



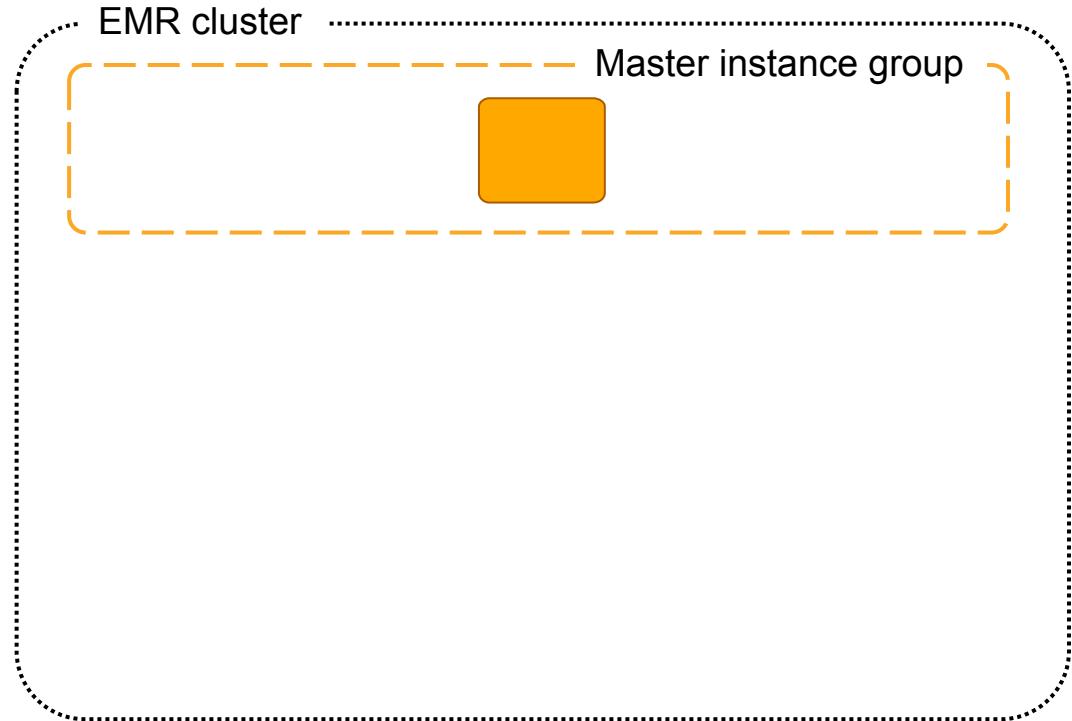
Monitor the health and progress of your cluster. Retrieve the output in S3.

Start an EMR  
cluster using  
console or cli  
tools

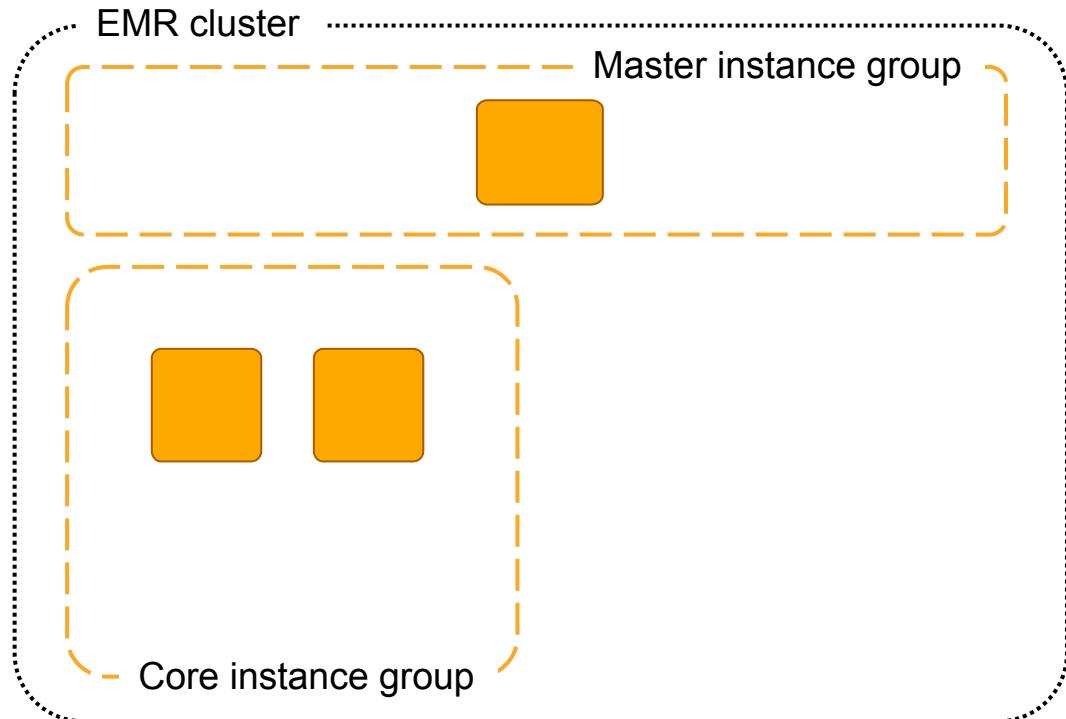
EMR cluster



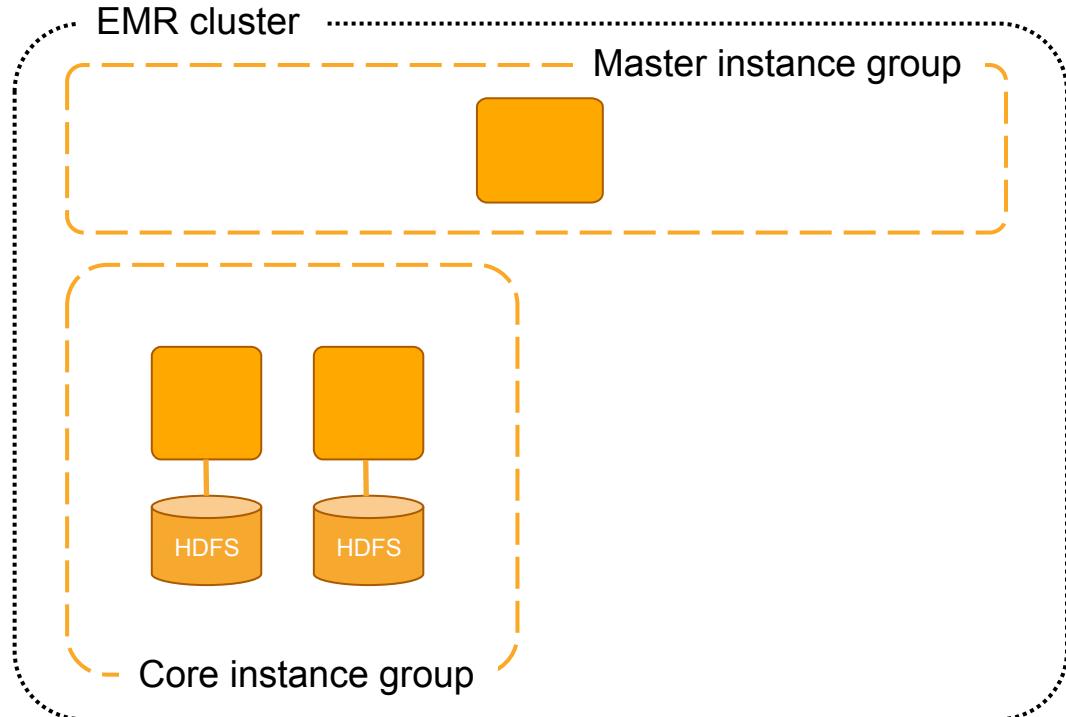
Master instance group created that controls the cluster (runs MySQL)



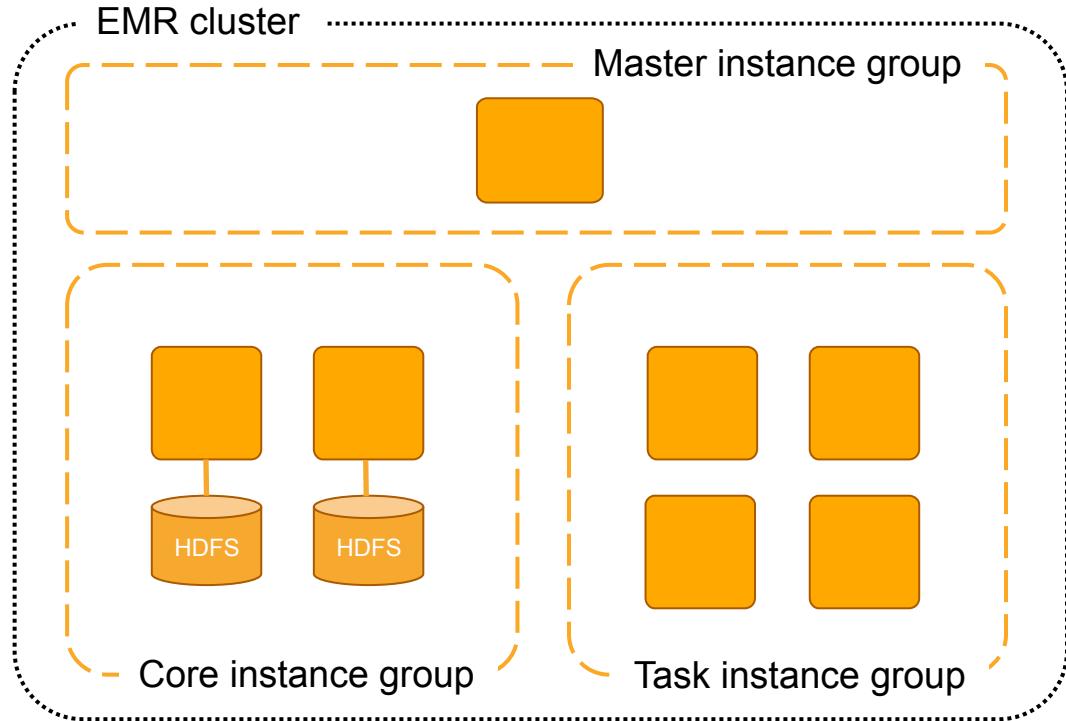
Core instance group created for life of cluster



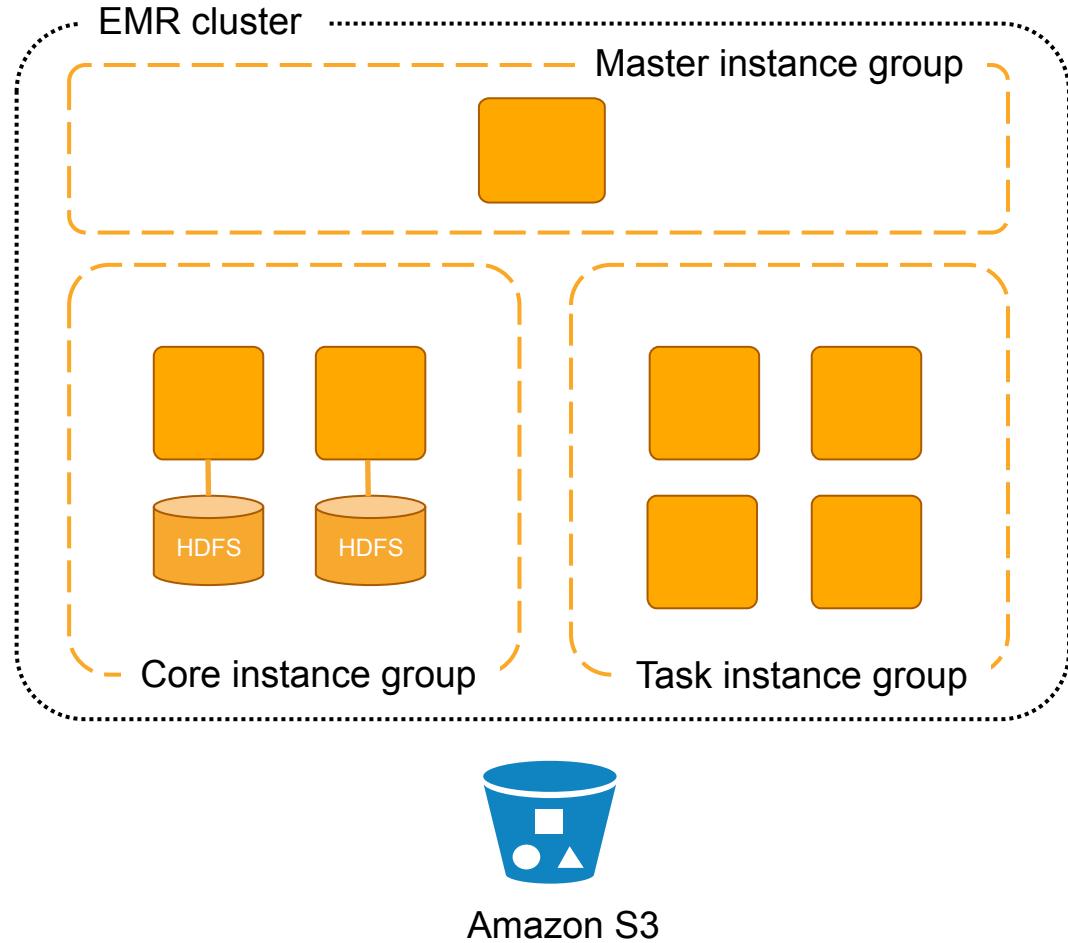
Core instances  
run DataNode  
and TaskTracker  
daemons



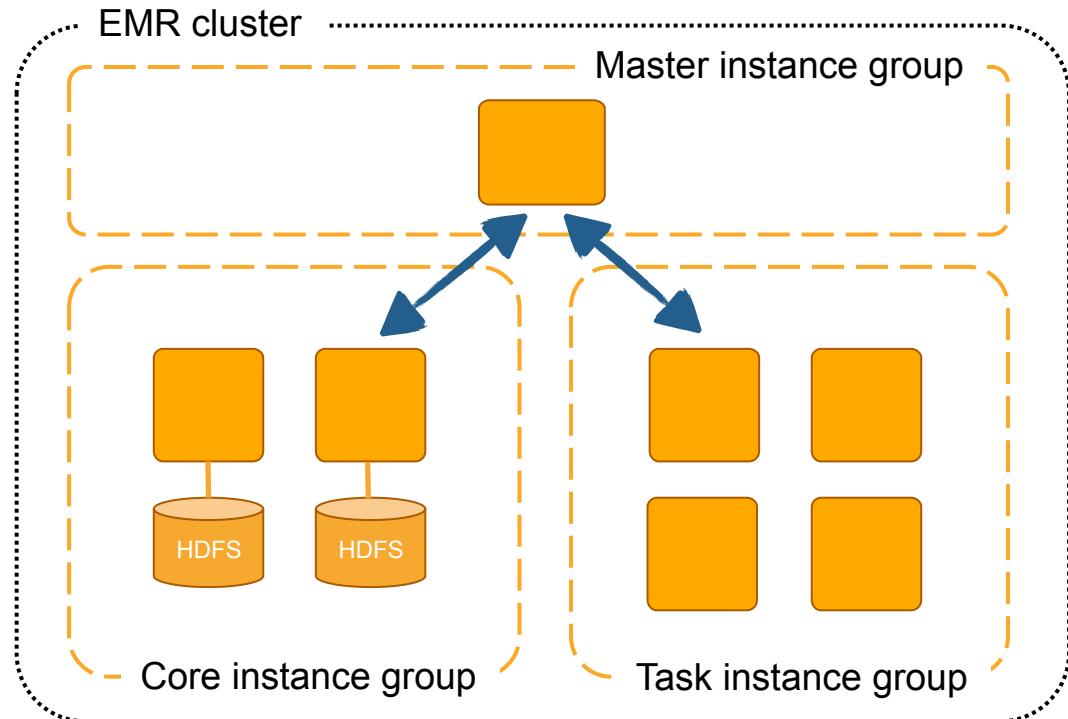
Optional task instances can be added or subtracted to perform work



S3 can be used  
as underlying ‘file  
system’ for input/  
output data

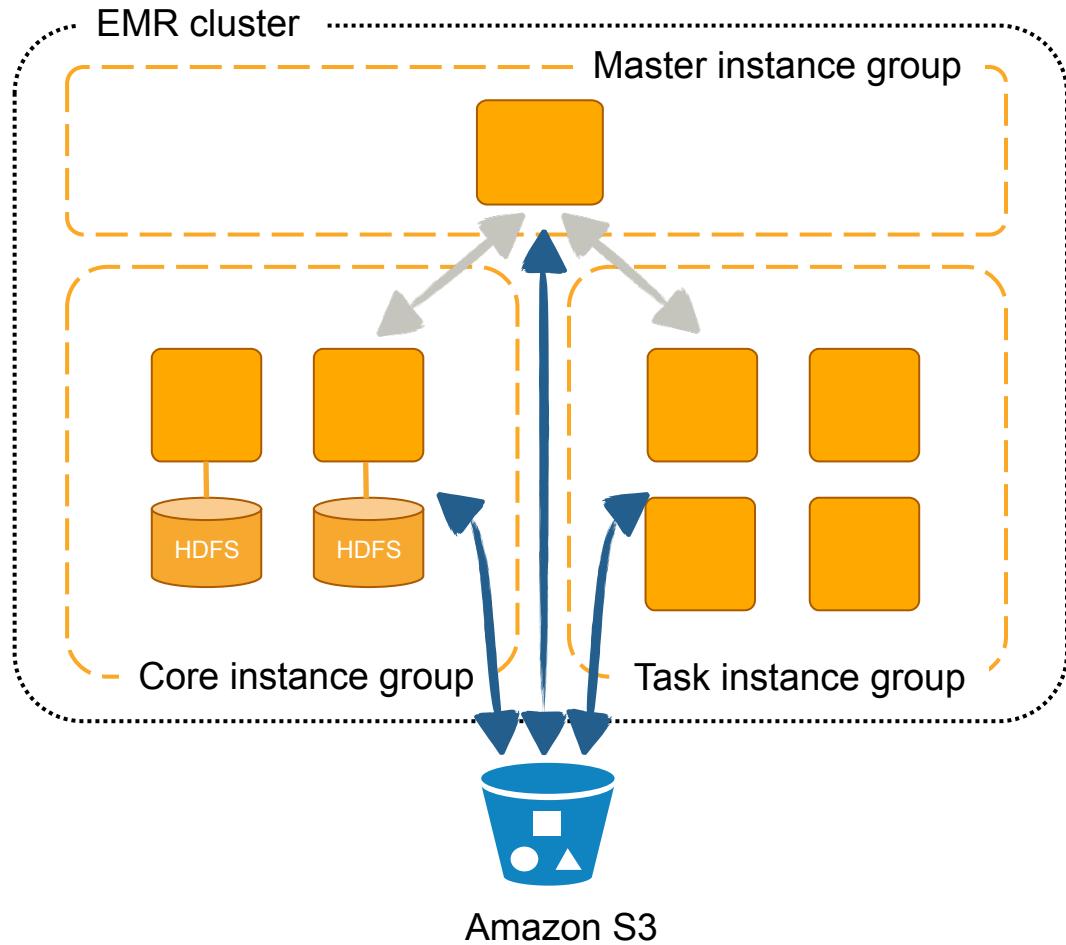


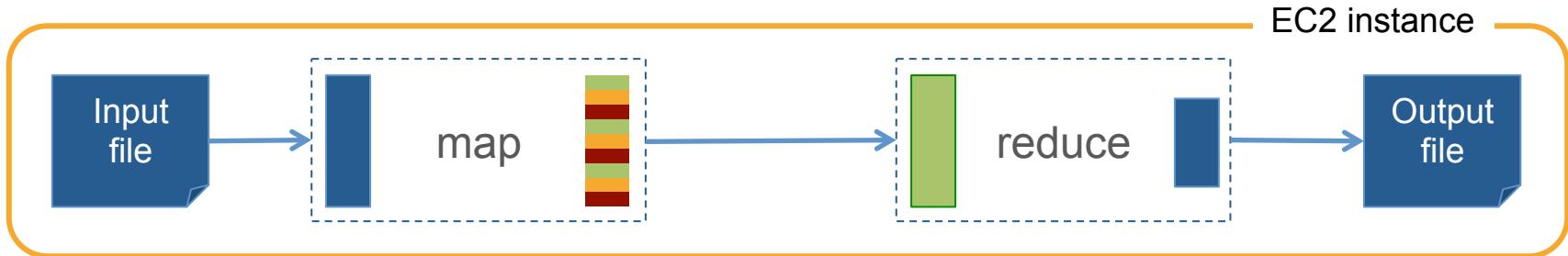
Master node  
coordinates  
distribution of  
work and  
manages cluster  
state

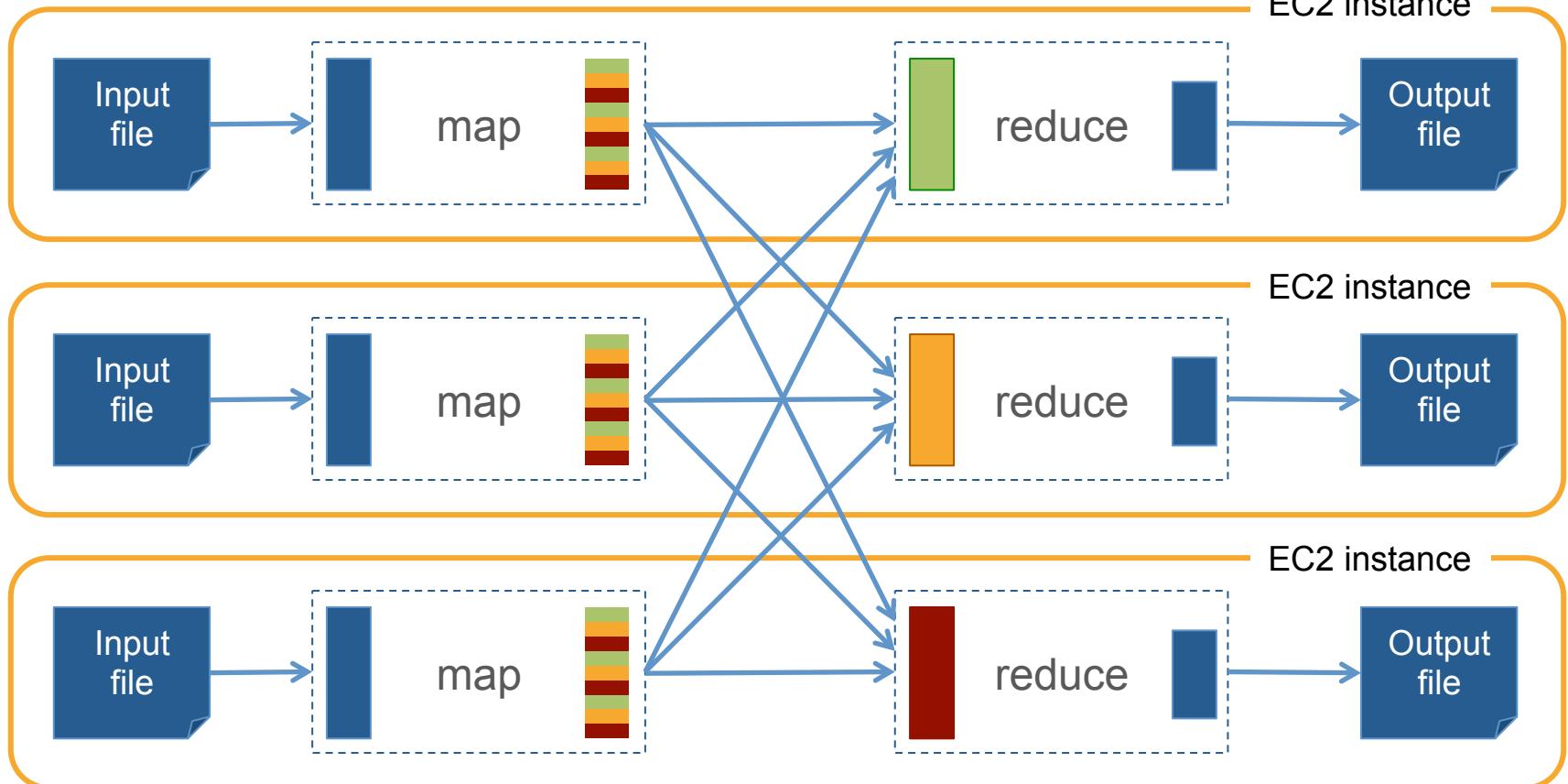


Amazon S3

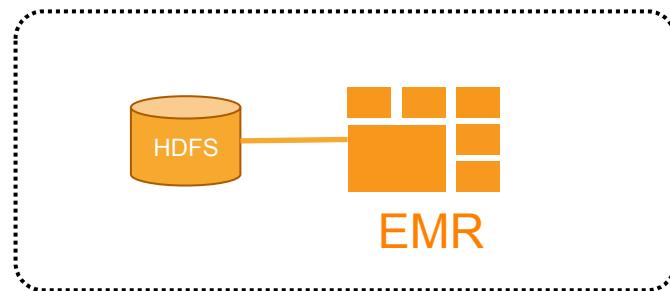
Core and Task  
instances read-  
write to S3



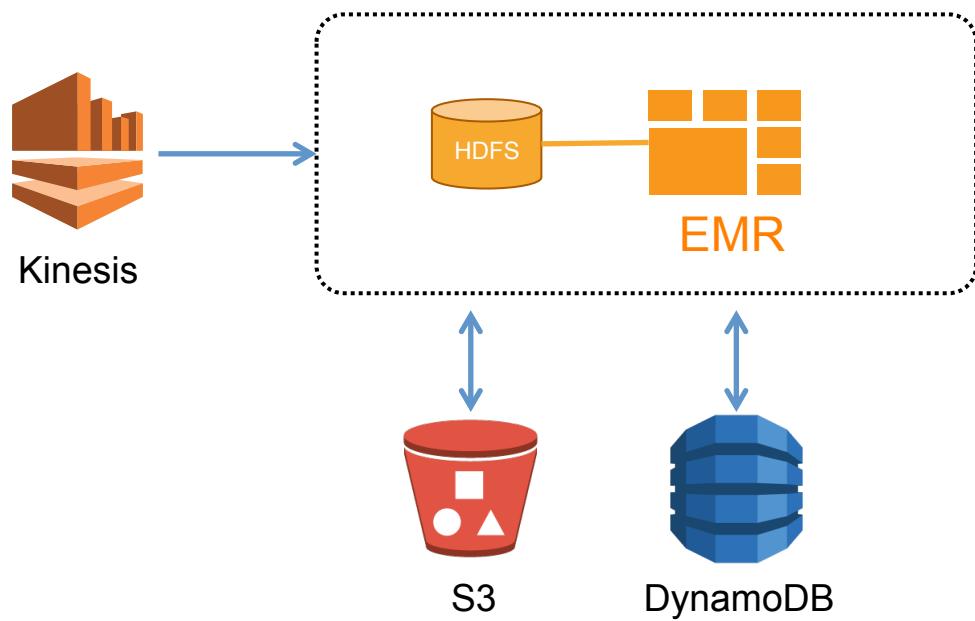




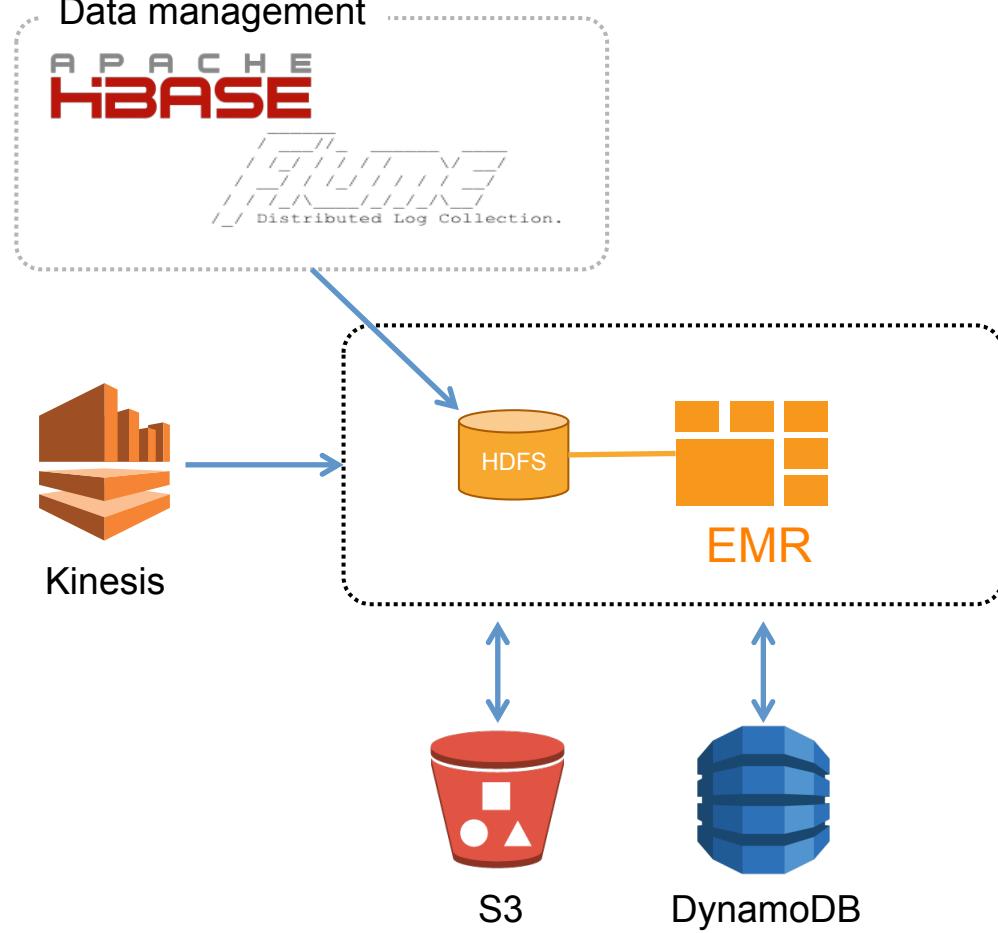
# Integration



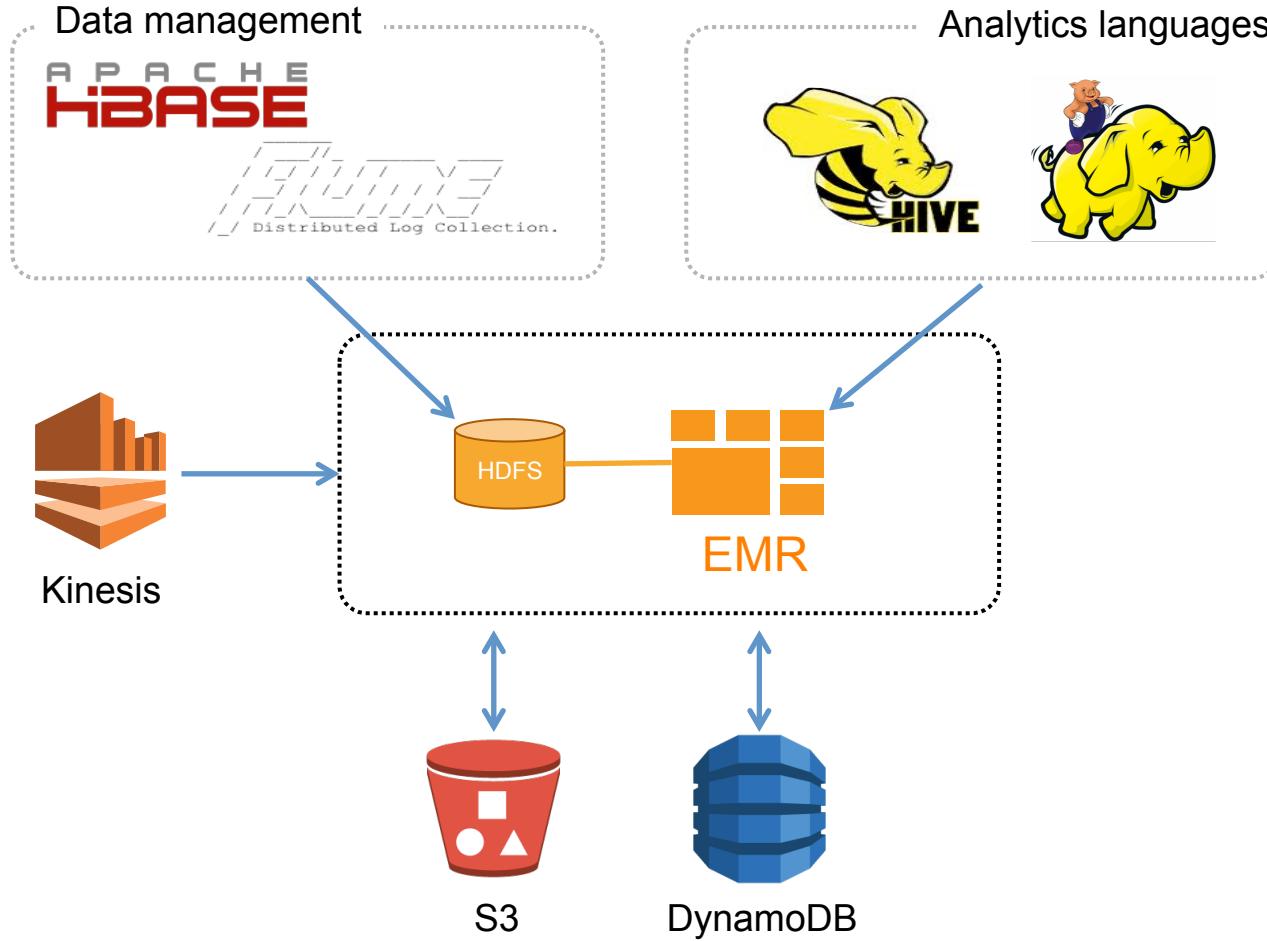
# Integration

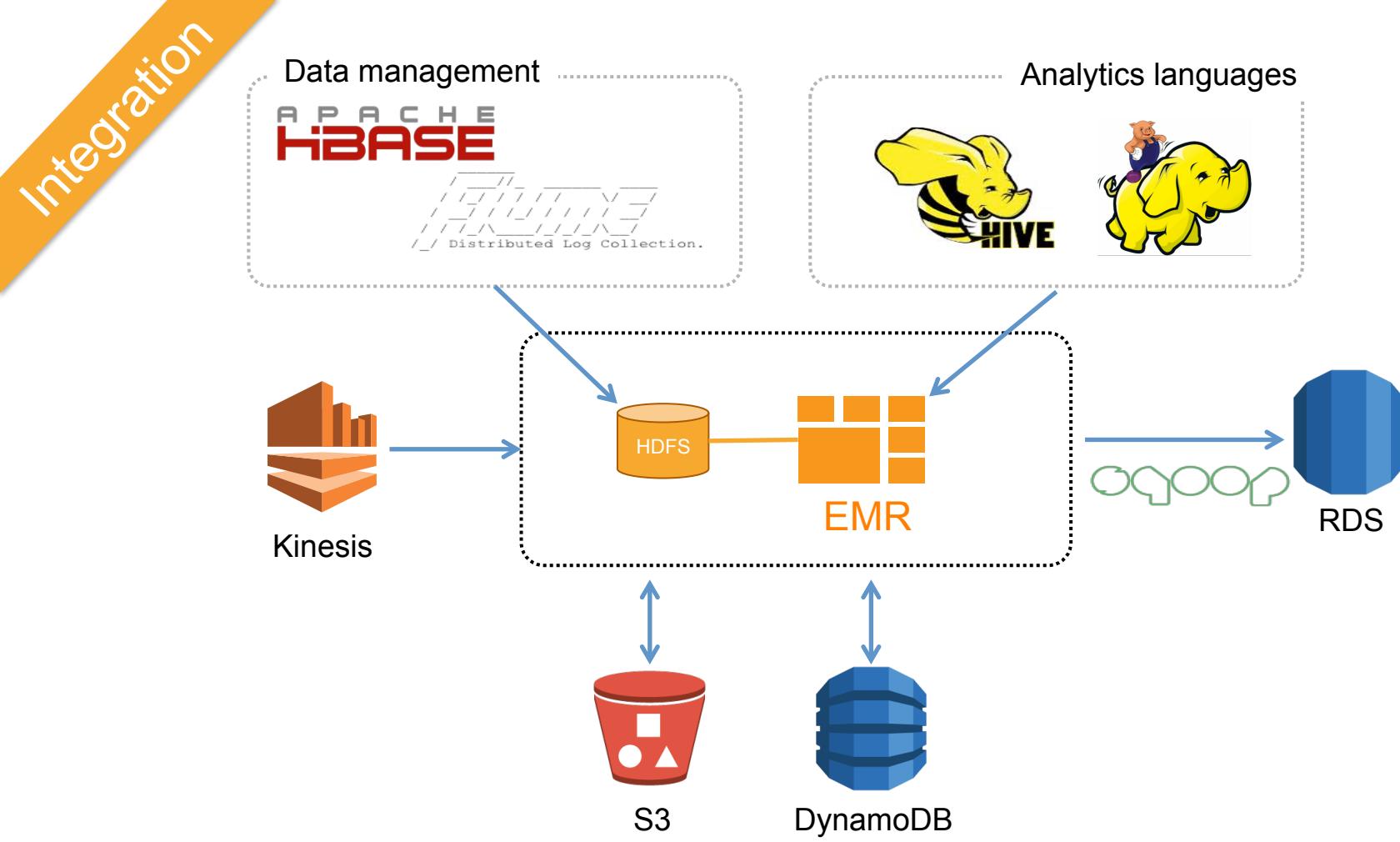


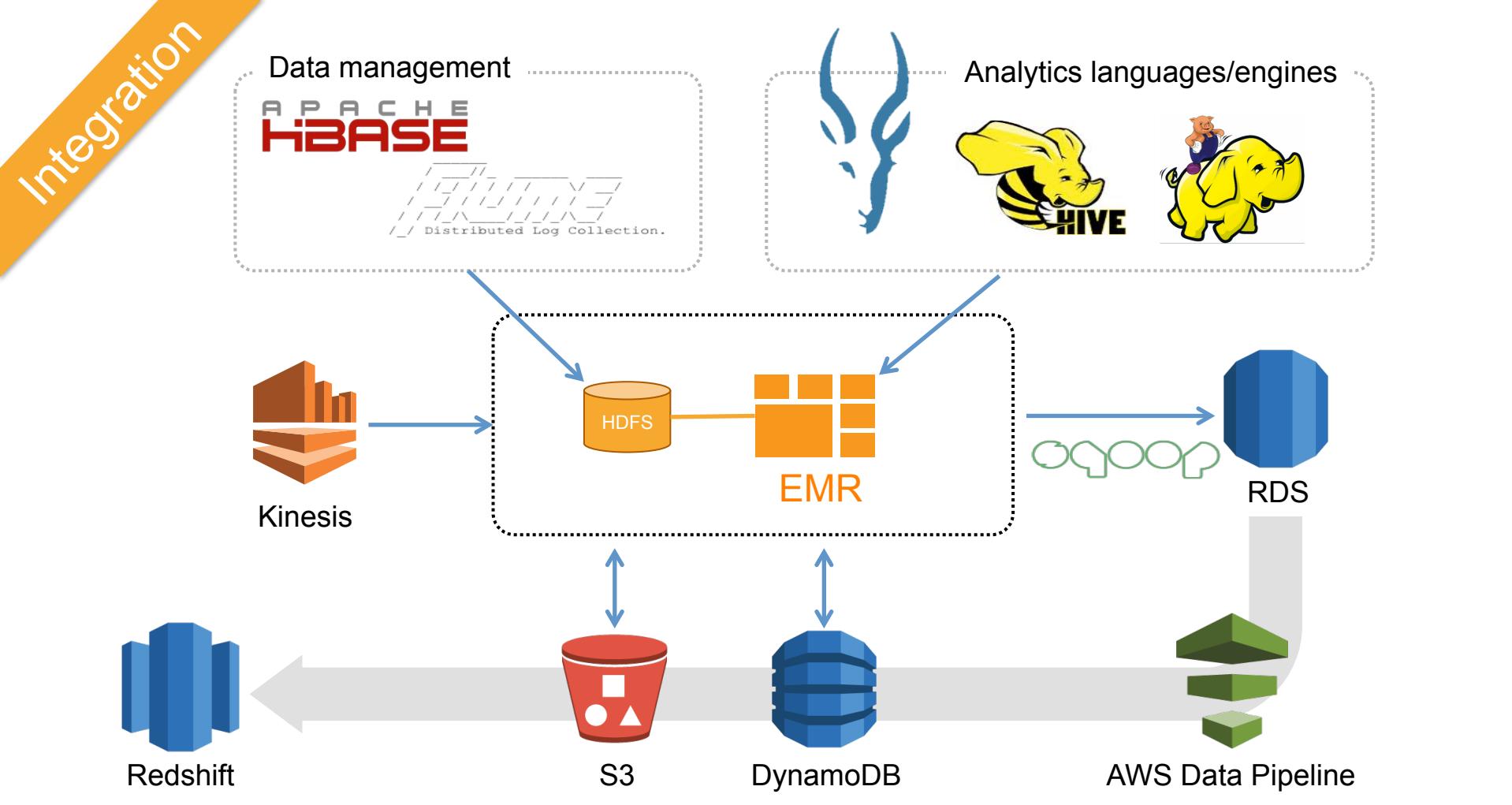
# Integration

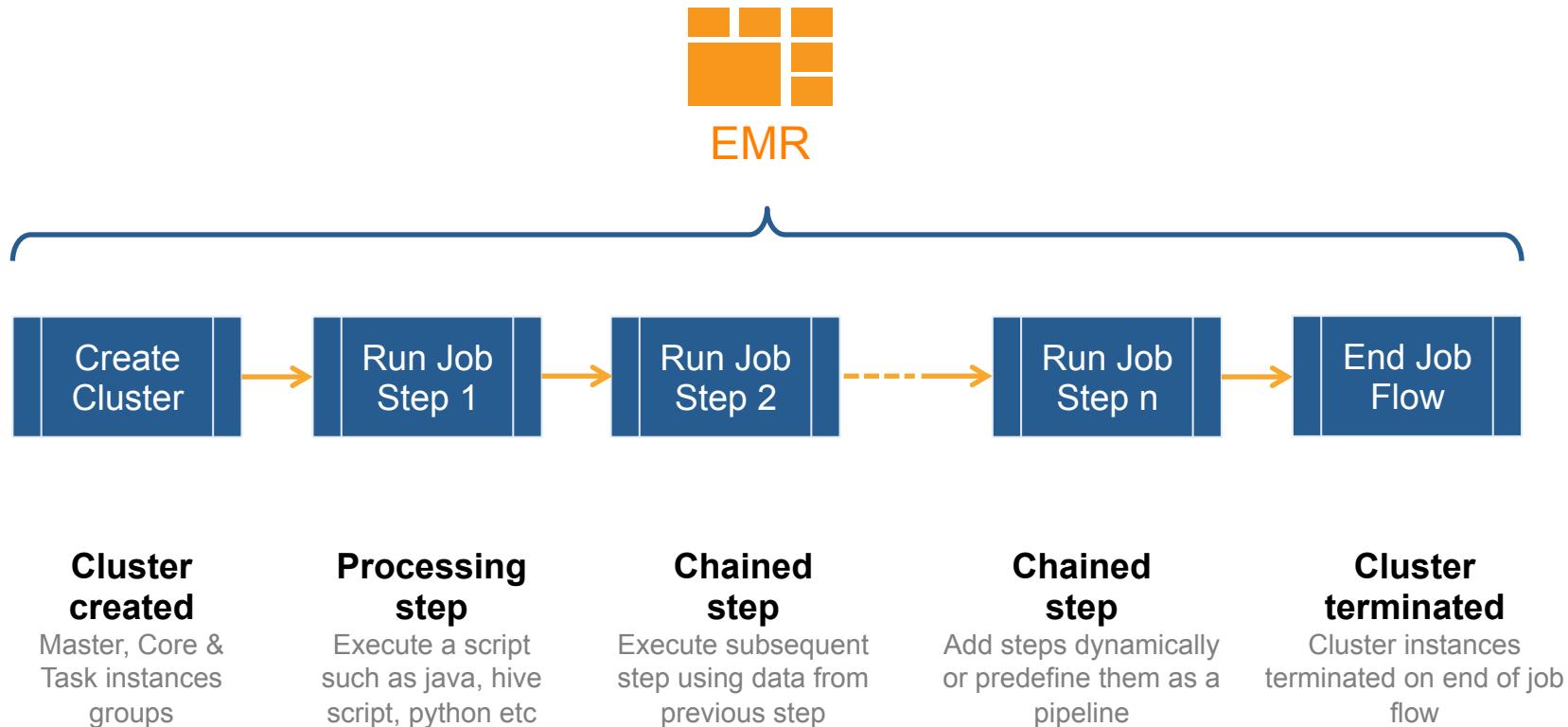


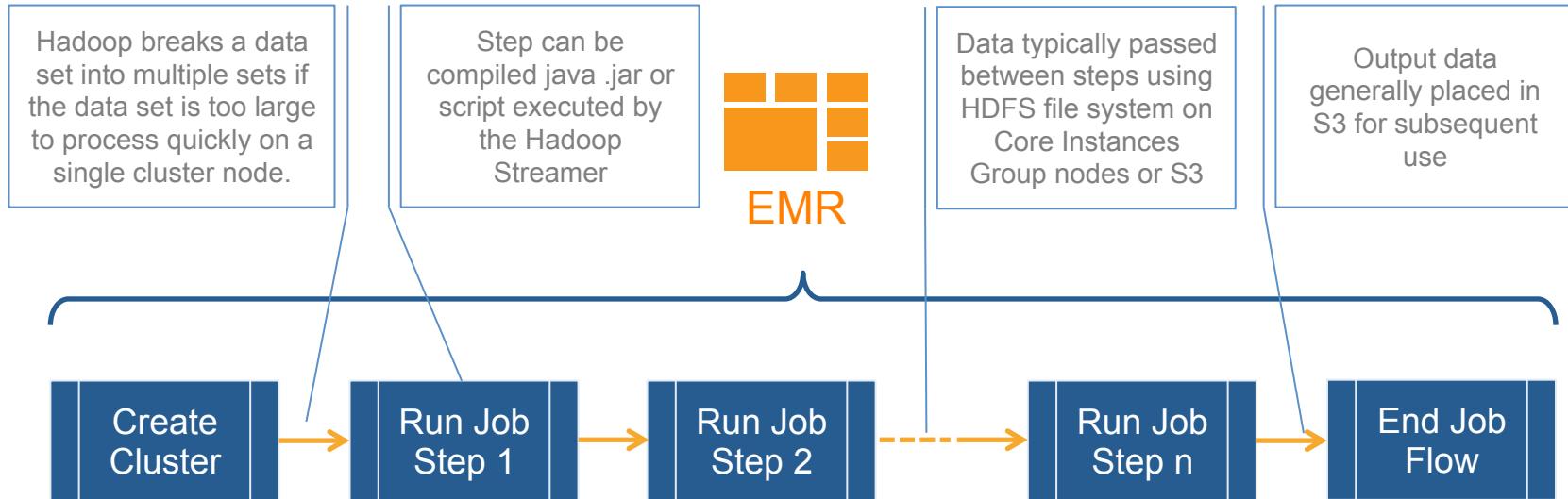
# Integration











### Cluster created

Master, Core & Task instances groups

### Processing step

Execute a script such as java, hive script, python etc

### Chained step

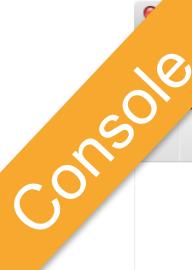
Execute subsequent step using data from previous step

### Chained step

Add steps dynamically or predefine them as a pipeline

### Cluster terminated

Cluster instances terminated on end of job flow



AWS Elastic MapReduce Management Console

MapReduce Manage... +

Services EC2 S3 RDS CloudFront Route 53 Edit

Ian Massingham Ireland Help

Elastic MapReduce Create Cluster EMR Help

## Cluster Configuration

[Configure sample application](#)

Cluster name

Termination protection  Yes

No

Prevents accidental termination of the cluster: to shut down the cluster, you must turn off termination protection. [Learn more](#)

Logging  Enabled

Copy the cluster's log files automatically to S3. [Learn more](#)

Log folder S3 location

/

/

Debugging  Enabled

Index logs to enable console debugging functionality (requires logging). [Learn more](#)

## Tags

**i** Optional: Add up to 10 tags to your EMR cluster. A tag consists of a case-sensitive key-value pair. Tags on EMR clusters are propagated to the underlying EC2 instances. [Learn more](#) about tagging your Amazon EMR clusters.

Key

Value (optional)

## Software Configuration

Hadoop distribution  Amazon

Use Amazon's Hadoop distribution. [Learn more](#)

AMI version

Determines the base configuration of the instances in your cluster, including the Hadoop version. [Learn more](#)

MapR

Use MapR's Hadoop distribution. [Learn more](#)

## Console

## Configuration

Cluster name

Word count

Termination protection

- 
- Yes
- 
- 
- No

Prevents accidental termination of the cluster: to shut down the cluster, you must turn off termination protection. [Learn more](#)

## Configure Sample Application

**i** Select a sample application to auto-populate the Create Cluster page

Select sample application

Word count

Output location

s3://ianmas-aws-emr/wordcount/output/2014-05-06/12



Logging

 Enabled

s3://ianmas-aws-emr/logs/2014-05-06/12-46-04



s3://&lt;bucket-name&gt;/&lt;folder&gt;/

Debugging

 Enabled

Cancel

Ok

## Tags

**i** Optional: Add up to 50 tags to the underlying EC2 instances.

Key

Add a key to create a tag

## Hadoop Streaming

Utility that comes with the Hadoop distribution

Allows you to create and run Map/Reduce jobs with any executable or script as the mapper and/or the reducer

Reads the input from standard input and the reducer outputs data through standard output

By default, each line of input/output represents a record with tab separated key/value

additional software and customize your applications. [Learn more](#)

Bootstrap action type	Name	S3 location	Optional arguments
Add bootstrap action	Select a bootstrap action		
	<a href="#">Configure and add</a>		

## Steps

 A step is a unit of work you submit to the cluster. A step might contain one or more Hadoop jobs, or contain instructions to install or configure an application. You can submit up to 256 steps to a cluster. [Learn more](#)

Name	Action on failure	JAR S3 location	Arguments
Word count	Terminate cluster	/home/hadoop/contrib/streaming/hadoop-streaming.jar	-mapper s3://eu-west-1.elasticmapreduce/samples/wordcount/wordSplitter.py -reducer aggregate -input s3://eu-west-1.elasticmapreduce/samples/wordcount/input -output s3://ianmas-aws-emr/wordcount/output /2014-05-06/12-50-13

Add step	Select a step
	<a href="#">Configure and add</a>

**Auto-terminate**  Yes      Automatically terminate cluster after the last step is completed.

No      Keep cluster running until you terminate it.

[Cancel](#)

[Create cluster](#)

Bootstrap action type	Name	S3 location	Optional arguments
Add bootstrap action	Select a bootstrap action		

**Add Step**

Step type Streaming program

Name\* Word count

Mapper\* **s3://eu-west-1.elasticmapreduce/samples/wordcount/wordSp**

Reducer\* aggregate

Input S3 location\* s3://eu-west-1.elasticmapreduce/samples/wordcount/inp

s3://<bucket-name>/<folder>/

Output S3 location\* s3://ianmas-aws-emr/wordcount/output/2014-05-06/12

s3://<bucket-name>/<folder>/

Arguments

Action on failure Terminate cluster

What to do if the step fails.

**Mapper Script Location (S3)**

S3 location of the map function or the name of the Hadoop streaming command to run.

S3 location of the reduce function or the name of the Hadoop streaming command to run.

**Cancel** **Save**

## Streaming: word count python script

Stdin > Stdout

```
#!/usr/bin/python
import sys
import re

def main(argv):
    pattern = re.compile("[a-zA-Z][a-zA-Z0-9]*")
    for line in sys.stdin:
        for word in pattern.findall(line):
            print "LongValueSum:" + word.lower() + "\t" + "1"

if __name__ == "__main__":
    main(sys.argv)
```

## Streaming: word count python script

Stdin > Stdout

```
#!/usr/bin/python
import sys
import re

def main(argv):
    pattern = re.compile("[a-zA-Z][a-zA-Z0-9]*")
    for line in sys.stdin:
        for word in pattern.findall(line):
            print "LongValueSum:" + word.lower() + "\t" + "1"

if __name__ == "__main__":
    main(sys.argv)
```

Read words from StdIn line by line



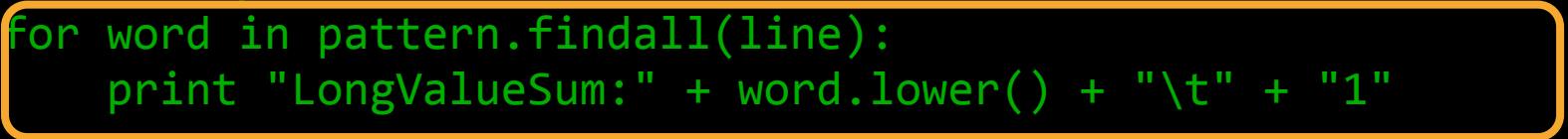
## Streaming: word count python script

Stdin > Stdout

```
#!/usr/bin/python
import sys
import re

def main(argv):
    pattern = re.compile("[a-zA-Z][a-zA-Z0-9]*")
    for line in sys.stdin:
        for word in pattern.findall(line):
            print "LongValueSum:" + word.lower() + "\t" + "1"

if __name__ == "__main__":
    main(sys.argv)
```



Output to StdOut tab delimited record  
e.g. "LongValueSum:abacus\t1"

# Console

Bootstrap action type	Name	S3 location	Optional arguments
Add bootstrap action	Select a bootstrap action		

**Add Step** X

**Step type** Streaming program

**Name\*** Word count

**Mapper\*** s3://eu-west-1.elasticmapreduce/samples/wordcount/wordSp

S3 location of the map function or the name of the Hadoop streaming command to run.

**Reducer\*** aggregate

S3 location of the reduce function or the name of the Hadoop streaming command to run.

**Input S3 location\*** s3://eu-west-1.elasticmapreduce/samples/wordcount/inp 

s3://<bucket-name>/<folder>/

**Output S3 location\*** s3://ianmas-aws-emr/wordcount/output/2014-05-06/12 

s3://<bucket-name>/<folder>/

**Arguments** 

**Action on failure** Terminate cluster What to do if the step fails.

**Cancel** **Save**

**Input source** 

# Console

Bootstrap action type	Name	S3 location	Optional arguments
Add bootstrap action	Select a bootstrap action		

**Add Step** X

**Step type** Streaming program

**Name\*** Word count

**Mapper\*** s3://eu-west-1.elasticmapreduce/samples/wordcount/wordSp

S3 location of the map function or the name of the Hadoop streaming command to run.

**Reducer\*** aggregate

S3 location of the reduce function or the name of the Hadoop streaming command to run.

**Input S3 location\*** s3://eu-west-1.elasticmapreduce/samples/wordcount/inp 

s3://<bucket-name>/<folder>/

**Output S3 location\*** s3://ianmas-aws-emr/wordcount/output/2014-05-06/12 

s3://<bucket-name>/<folder>/

**Arguments** 

**Action on failure** Terminate cluster What to do if the step fails.

**Cancel** **Save**

**Output location** 

# Console

Bootstrap action type	Name	S3 location	Optional arguments
Add bootstrap action	Select a bootstrap action		

**Add Step** X

Step type Streaming program

Name\* Word count

Mapper\* s3://eu-west-1.elasticmapreduce/samples/wordcount/wordSp

Reducer\* aggregate

Input S3 location\* s3://eu-west-1.elasticmapreduce/samples/wordcount/inp 

s3://<bucket-name>/<folder>/

Output S3 location\* s3://ianmas-aws-emr/wordcount/output/2014-05-06/12 

s3://<bucket-name>/<folder>/

Arguments

Action on failure Terminate cluster What to do if the step fails.

Reducer\* aggregate 

Reducer script or function

Cancel Save

# Console

Bootstrap action type	Name	S3 location	Optional arguments
Add bootstrap action	Select a bootstrap action		

Add Step

Step type Streaming program

Name\* Word count

Mapper\* s3://eu-west-1.elasticmapreduce/samples/wordcount/wordSp

Reducer\* aggregate

Input S3 location\* s3://eu-west-1.elasticmapreduce/samples/wordcount/inp

Output S3 location\* s3://ianmas-aws-emr/wordcount/output/2014-05-06/12

Arguments

Action on failure Terminate cluster

What to do if the step fails.

Cancel Save

**Reducer\*** aggregate

S3 location of the reduce function or the name of the Hadoop streaming command to run.

Aggregate: Sort inputs and add up totals  
e.g  
"Abacus 1"  
"Abacus 1"  
becomes  
"Abacus 2"

## Hardware Configuration

**ⓘ** Specify the [networking](#) and [hardware](#) configuration for your cluster. If you need more than 20 EC2 instances, [complete this form](#). Request [Spot instances](#) (unused EC2 capacity) to save money.

**Network** vpc-b7b8b3d5 (172.31.0.0/16) (default)

Use a Virtual Private Cloud (VPC) to process sensitive data or connect to a private network. [Create a VPC](#)

**EC2 Subnet** No preference (random subnet)

[Create a Subnet](#)

EC2 instance type	Count	Request spot	
Master	m1.small	1	<input type="checkbox"/>
Core	m1.small	2	<input type="checkbox"/>
Task	m1.small	0	<input type="checkbox"/>

The Master instance assigns Hadoop tasks to core and task nodes, and monitors their status.

Core instances run Hadoop tasks and store data using the Hadoop Distributed File System (HDFS).

Task instances run Hadoop tasks.

## Security and Access

**EC2 key pair** Proceed without an EC2 key pair

Use an existing key pair to SSH into the master node of the Amazon EC2 cluster as the user "hadoop". [Learn more](#)

**IAM user access**  All other IAM users

Control the visibility of this cluster to other IAM users. [Learn more](#)

No other IAM users

**EC2 role** Proceed without role

Control permissions for applications on the cluster. [Learn more](#)

## Bootstrap Actions

**ⓘ** Bootstrap actions are scripts that are executed during setup before Hadoop starts on every cluster node. You can use them to install additional software and customize your applications. [Learn more](#)

Bootstrap action type	Name	S3 location	Optional arguments
-----------------------	------	-------------	--------------------

**Add bootstrap action** [Select a bootstrap action](#)

# Instance groups

## Master instance group

Manages the job flow: coordinating the distribution of the MapReduce executable and subsets of the raw data, to the core and task instance groups

It also tracks the status of each task performed, and monitors the health of the instance groups.

To monitor the progress of the job flow, you can SSH into the master node as the Hadoop user and either look at the Hadoop log files directly or access the user interface that Hadoop

# Instance groups

## Master instance group

Manages the job flow: coordinating the distribution of the MapReduce executable and subsets of the raw data, to the core and task instance groups

It also tracks the status of each task performed, and monitors the health of the instance groups.

To monitor the progress of the job flow, you can SSH into the master node as the Hadoop user and either look at the Hadoop log files directly or access the user interface that Hadoop

## Core instance group

Contains all of the core nodes of a job flow.

A core node is an EC2 instance that runs Hadoop map and reduce tasks and stores data using the Hadoop Distributed File System (HDFS).

The EC2 instances you assign as core nodes are capacity that must be allotted for the entire job flow run.

Core nodes run both the DataNodes and TaskTracker Hadoop daemons.

# Instance groups

## Master instance group

Manages the job flow: coordinating the distribution of the MapReduce executable and subsets of the raw data, to the core and task instance groups

It also tracks the status of each task performed, and monitors the health of the instance groups.

To monitor the progress of the job flow, you can SSH into the master node as the Hadoop user and either look at the Hadoop log files directly or access the user interface that Hadoop

## Core instance group

Contains all of the core nodes of a job flow.

A core node is an EC2 instance that runs Hadoop map and reduce tasks and stores data using the Hadoop Distributed File System (HDFS).

The EC2 instances you assign as core nodes are capacity that must be allotted for the entire job flow run.

Core nodes run both the DataNodes and TaskTracker Hadoop daemons.

## Task instance group

Contains all of the task nodes in a job flow.

The task instance group is optional.

You can add it when you start the job flow or add a task instance group to a job flow in progress.

You can increase and decrease the number of task nodes. Because they don't store data and can be added and removed from a job flow, you can use task nodes to manage the EC2 instance capacity your job flow uses

## Hardware Configuration

**ⓘ** Specify the [networking](#) and [hardware](#) configuration for your cluster. If you need more than 20 EC2 instances, [complete this form](#). Request [Spot instances](#) (unused EC2 capacity) to save money.

**Network**  Use a Virtual Private Cloud (VPC) to process sensitive data or connect to a private network. [Create a VPC](#)

**EC2 Subnet**  [Create a Subnet](#)

	EC2 instance type	Count	Request spot	
<b>Master</b>	<input type="text" value="m1.small"/>	1	<input type="checkbox"/>	The Master instance assigns Hadoop tasks to core and task nodes, and monitors their status.
<b>Core</b>	<input type="text" value="m1.small"/>	2	<input type="checkbox"/>	Core instances run Hadoop tasks and store data using the Hadoop Distributed File System (HDFS).
<b>Task</b>	<input type="text" value="m1.small"/>	0	<input type="checkbox"/>	Task instances run Hadoop tasks.

## Security and Access

**EC2 key pair**  Use an existing key pair to SSH into the master node of the Amazon EC2 cluster as the user "hadoop". [Learn more](#)

**IAM user access**  All other IAM users  
 No other IAM users Control the visibility of this cluster to other IAM users. [Learn more](#)

**EC2 role**  Control permissions for applications on the cluster. [Learn more](#)

## Bootstrap Actions

**ⓘ** Bootstrap actions are scripts that are executed during setup before Hadoop starts on every cluster node. You can use them to install additional software and customize your applications. [Learn more](#)

Bootstrap action type	Name	S3 location	Optional arguments
<a href="#">Add bootstrap action</a> <input type="text" value="Select a bootstrap action"/>			

## Hardware Configuration

Specify the [networking](#) and [hardware](#) configuration for your cluster. If you need more than 20 EC2 instances, [complete this form](#). Request [Spot instances](#) (unused EC2 capacity) to save money.

Network [vpc-b7b8b3d5 \(172.31.0.0/16\) \(default\)](#) Use a Virtual Private Cloud (VPC) to process sensitive data or connect to a private network. [Create a VPC](#)

EC2 Subnet [No preference \(random subnet\)](#) [Create a Subnet](#)

EC2 instance type	Count	Request spot	
Master	<a href="#">m1.small</a>	1	<input type="checkbox"/>
Core	<a href="#">m1.small</a>	2	<input type="checkbox"/>
Task	<a href="#">m1.small</a>	0	<input type="checkbox"/>

The Master instance assigns Hadoop tasks to core and task nodes, and monitors their status.  
Core instances run Hadoop tasks and store data using the Hadoop Distributed File System (HDFS).  
Task instances run Hadoop tasks.

## Security and Access

EC2 key pair [Proceed without an EC2 key pair](#) Use an existing key pair to access the Amazon EC2 cluster. [more](#)

IAM user access  All other IAM users  No other IAM users Control the visibility of this users. [Learn more](#)

EC2 role [Proceed without role](#) Control permissions for applications on the cluster. [Learn more](#)

SSH onto master node with this keypair

## Bootstrap Actions

Bootstrap actions are scripts that are executed during setup before Hadoop starts on every cluster node. You can use them to install additional software and customize your applications. [Learn more](#)

Bootstrap action type	Name	S3 location	Optional arguments
-----------------------	------	-------------	--------------------

Add bootstrap action [Select a bootstrap action](#)

MapReduce Manage...

## Bootstrap Actions

**i** Bootstrap actions are scripts that are executed during setup before Hadoop starts on every cluster node. You can use them to install additional software and customize your applications. [Learn more](#)

Bootstrap action type	Name	S3 location	Optional arguments
Add bootstrap action	Select a bootstrap action		
	<a href="#">Configure and add</a>		

## Steps

**i** A step is a unit of work you submit to the cluster. A step might contain one or more Hadoop jobs, or contain instructions to install or configure an application. You can submit up to 256 steps to a cluster. [Learn more](#)

Name	Action on failure	JAR S3 location	Arguments
Word count	Terminate cluster	/home/hadoop/contrib/streaming/hadoop-streaming.jar	-mapper s3://eu-west-1.elasticmapreduce/samples/wordcount/wordSplitter.py -reducer aggregate -input s3://eu-west-1.elasticmapreduce/samples/wordcount/input -output s3://ianmas-aws-emr/wordcount/output /2014-05-06/12-50-13

Keep cluster running so you can continue to work on it



Add step	Select a step
	<a href="#">Configure and add</a>
Auto-terminate	<input checked="" type="radio"/> Yes
	<input type="radio"/> No
	Automatically terminate cluster after the last step is completed.
	Keep cluster running until you terminate it.

[Cancel](#)[Create cluster](#)

# S3 output

...  
aakar 3  
abdal 3  
abdelaziz 18  
abolish 3  
aboriginal 12  
abraham 6  
abseron 9  
abstentions 15  
accession 73  
accord 90  
achabar 3  
achievements 3  
acic4  
acknowledged 6  
acquis 9  
acquisitions 3  
...

```
./elastic-mapreduce --create --stream
--mapper
    s3://elasticmapreduce/samples/wordcount/wordSplitter.py
--input
    s3://elasticmapreduce/samples/wordcount/input
--output
    s3n://myawsbucket/output
--reducer
    aggregate
```

```
./elastic-mapreduce --create --stream
--mapper
    s3://elasticmapreduce/samples/wordcount/wordSplitter.py
--input
    s3://elasticmapreduce/samples/wordcount/input
--output
    s3n://myawsbucket/output
--reducer
    aggregate
```

```
./elastic-mapreduce --create --stream  
--mapper  
    s3://elasticmapreduce/samples/wordcount/wordSplitter.py  
--input  
    s3://elasticmapreduce/samples/wordcount/input  
--output  
    s3n://myawsbucket/output  
--reducer  
    aggregate
```

```
./elastic-mapreduce --create --stream  
--mapper  
    s3://elasticmapreduce/samples/wordcount/wordSplitter.py  
--input  
    s3://elasticmapreduce/samples/wordcount/input  
--output  
    s3n://myawsbucket/output  
--reducer  
    aggregate
```

```
./elastic-mapreduce --create --stream
--mapper
    s3://elasticmapreduce/samples/wordcount/wordSplitter.py
--input
    s3://elasticmapreduce/samples/wordcount/input
--output
    s3n://myawsbucket/output
--reducer
    aggregate
```

# Defaults

launches a job flow to run on a single-node cluster  
using an Amazon EC2 m1.small instance

when your steps are running correctly on a small set  
of sample data, you can launch job flows to run on  
multiple nodes

You can specify the number of nodes and the type of  
instance to run with the `--num-instances` and `--  
instance-type` parameters

# Starting a flow

```
elastic-mapreduce  
--create  
--key-pair micro  
--region eu-west-1  
--name MyJobFlow  
--num-instances 5  
--instance-type m2.4xlarge  
--alive  
--log-uri s3n://mybucket/EMR/log
```

# Starting a flow

```
elastic-mapreduce  
  --create  
  --key-pair micro  
  --region eu-west-1  
  --name MyJobFlow  
  --num-instances 5  
  --instance-type m2.4xlarge  
  --alive  
  --log-uri s3n://mybucket/EMR/log
```

Instance type/count



# Starting a flow

```
elastic-mapreduce  
--create  
--key-pair micro  
--region eu-west-1  
--name MyJobFlow  
--num-instances 5  
--instance-type m2.4xlarge  
--alive  
--log-uri s3n://mybucket/EMR/log
```

Don't terminate cluster –  
keep it running so steps  
can be added



# Starting a flow

```
elastic-mapreduce  
--create  
--key-pair micro  
--region eu-west-1  
--name MyJobFlow  
--num-instances 5  
--instance-type m2.4xlarge  
--alive  
--pig-interactive --pig-versions latest  
--hive-interactive --hive-versions latest  
--hbase  
--log-uri s3n://mybucket/EMR/log
```

Adding Hive, Pig and Hbase to the job flow



# Starting a flow

```
elastic-mapreduce  
--create  
--bootstrap-action s3://elasticmapreduce/bootstrap-  
actions/configure-hadoop --args  
"-s,dfs.block.size=1048576"  
--key-pair micro  
--region eu-west-1  
--name MyJobFlow  
--num-instances 5  
--instance-type m2.4xlarge  
--alive  
--pig-interactive --pig-versions latest  
--hive-interactive --hive-versions latest  
--hbase  
--log-uri s3n://mybucket/EMR/log
```

Using a bootstrapping  
action to configure the  
cluster

You can specify up to 16 bootstrap actions per job flow by providing multiple *bootstrap-action* parameters

### s3://elasticmapreduce/bootstrap-actions/ **configure-daemons**

Configure JVM properties

```
--bootstrap-action s3://elasticmapreduce/bootstrap-  
actions/configure-daemons --args --namenode-heap-  
size=2048,--namenode-opts=-XX:GCTimeRatio=19
```

### s3://elasticmapreduce/bootstrap-actions/ **configure-hadoop**

Configure cluster wide hadoop settings

```
./elastic-mapreduce -create --bootstrap-action s3://  
elasticmapreduce/bootstrap-actions/configure-hadoop --args  
"--site-config-file,s3://myawsbucket/config.xml,-  
s.mapred.tasktracker.map.tasks.maximum=2"
```

### s3://elasticmapreduce/bootstrap-actions/ configurations/latest/**memory-intensive**

Configure cluster wide memory settings

```
./elastic-mapreduce -create --bootstrap-action s3://  
elasticmapreduce/bootstrap-actions/configurations/latest/  
memory-intensive
```

You can specify up to 16 bootstrap actions per job flow by providing multiple *bootstrap-action* parameters

### s3://elasticmapreduce/bootstrap-actions/run-if

Conditionally run a command

```
./elastic-mapreduce --create --alive \
--bootstrap-action s3://elasticmapreduce/bootstrap-
actions/run-if \
--args "instance.isMaster=true,echo running on
master node"
```

### Shutdown actions

Run scripts on shutdown

A bootstrap action script can create one or more shutdown actions by writing scripts to the /mnt/var/lib/instance-controller/public/shutdown-actions/ directory. Each script must run and complete within 60 seconds.

### Custom actions

Run a custom script

```
--bootstrap-action "s3://elasticmapreduce/bootstrap-
actions/download.sh"
```



Bootstrapping

## Bootstrap Actions

Bootstrap actions are scripts that are executed during setup before Hadoop starts on every cluster node. You can use them to install additional software and customize your applications. [Learn more](#)

Bootstrap action type	Name	S3 location	Optional arguments
Add bootstrap action	Select a bootstrap action		
	<a href="#">Configure and add</a>		

## Steps

A step is a unit of work you submit to the cluster. A step might contain one or more Hadoop jobs, or contain instructions to install or configure an application. You can submit up to 256 steps to a cluster. [Learn more](#)

Name	Action on failure	JAR S3 location	Arguments
Word count	Terminate cluster	/home/hadoop/contrib/streaming/hadoop-streaming.jar	-mapper s3://eu-west-1.elasticmapreduce/samples/wordcount/wordSplitter.py -reducer aggregate -input s3://eu-west-1.elasticmapreduce/samples/wordcount/input -output s3://ianmas-aws-emr/wordcount/output/2014-05-06/12-50-13

Add step	Select a step
	<a href="#">Configure and add</a>

Auto-terminate  Yes

Automatically terminate cluster after the last step is completed.

No

Keep cluster running until you terminate it.

[Cancel](#)

[Create cluster](#)

# Bootstrapping

## Bootstrap Actions

**i** Bootstrap actions are scripts that are executed during setup before Hadoop starts on every cluster node. You can use them to install additional software and customize your applications. [Learn more](#)

Bootstrap action type	Name	S3 location	Optional arguments
Add bootstrap action	Configure Hadoop		
	<a href="#">Configure and add</a>		

## Steps

**i** A step is a unit of work you can add to a cluster to configure an application. You can add multiple steps to a cluster.

Name

Word count

**Add Bootstrap Action**

**Bootstrap action type** Configure Hadoop

**Name**

**S3 location** s3://eu-west-1.elasticmapreduce/bootstrap-actions/configure-hadoop

**Optional arguments**  
`--site-key-value  
io.file.buffer.size=65536`

[Cancel](#) [Add](#)

# Integrated tools

## Selecting the right one

## Hadoop streaming

Unstructured data with wide variation  
of formats, high flexibility on language

Broad language support (Python,  
Perl, Bash...)

Mapper, reducer, input and output all  
supplied by the jobflow

Scale from 1 to unlimited number of  
processes

## Hadoop streaming

Unstructured data with wide variation of formats, high flexibility on language

Broad language support (Python, Perl, Bash...)

Mapper, reducer, input and output all supplied by the jobflow

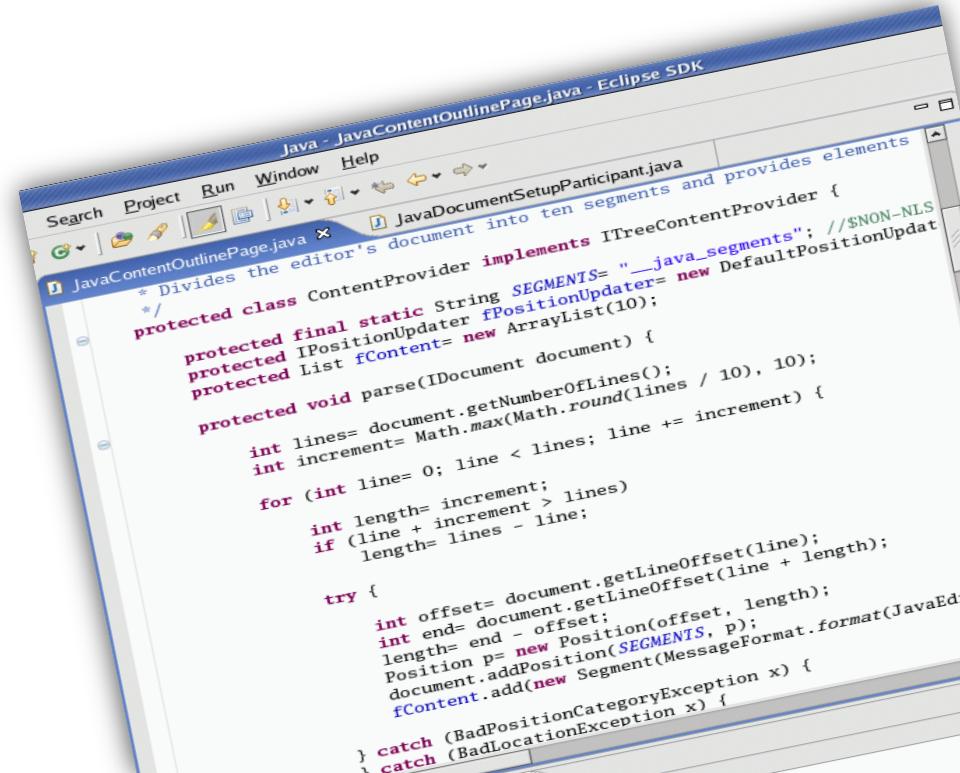
Scale from 1 to unlimited number of processes

Doing analytics in Eclipse is wrong

2-stage dataflow is rigid

Joins are lengthy and error-prone

Prototyping/exploration requires compile & deploy



The screenshot shows the Eclipse IDE interface with a Java editor window titled "Java - JavaContentOutlinePage.java - Eclipse SDK". The code in the editor is for a class named ContentProvider, which implements ITreeContentProvider. The code is annotated with Javadoc comments explaining its purpose: "Divides the editor's document into ten segments and provides elements". The code itself handles parsing an IDocument object into segments, calculating offsets, and adding segments to a PositionUpdater. It includes try-catch blocks for handling BadPositionCategoryException and BadLocationException.

```
protected class ContentProvider implements ITreeContentProvider {
    /**
     * Divides the editor's document into ten segments and provides elements
     */
    protected final static String SEGMENTS = "__java_segments";
    protected IPositionUpdater fPositionUpdater= new DefaultPositionUpdater();
    protected List<IContent> fContent= new ArrayList(10);

    protected void parse(IDocument document) {
        int lines= document.getNumberOfLines();
        int increment= Math.max(Math.round(lines / 10), 10);
        for (int line= 0; line < lines; line += increment) {
            int length= increment;
            if (line + increment > lines)
                length= lines - line;
            try {
                int offset= document.getLineOffset(line);
                int end= document.getLineOffset(line + length);
                length= end - offset;
                Position p= new Position(offset, length);
                document.addPosition(SEGMENTS, p);
                fContent.add(new Segment(MessageFormat.format(JavaEd
```

## Hadoop streaming

Unstructured data with wide variation of formats, high flexibility on language

Broad language support (Python, Perl, Bash...)

Mapper, reducer, input and output all supplied by the jobflow

Scale from 1 to unlimited number of processes

## Hive

Structured data which is lower value or where higher latency is tolerable

## Pig

Semi-structured data which can be mined using set operations

## Hbase

Near real time K/V store for structured data

# Hive integration

Schema on read



## SQL Interface for working with data

Simple way to use Hadoop

Create Table statement references data location on S3

Language called HiveQL, similar to SQL

An example of a query could be:

```
SELECT COUNT(1) FROM sometable;
```

Requires to setup a mapping to the input data

Uses SerDes to make it possible to query different input formats

Powerful data types (Array & Map..)

# Hive

	<b>SQL</b>	<b>HiveQL</b>
<b>Updates</b>	UPDATE, INSERT, DELETE	INSERT, OVERWRITE TABLE
<b>Transactions</b>	Supported	Not supported
<b>Indexes</b>	Supported	Not supported
<b>Latency</b>	Sub-second	Minutes
<b>Functions</b>	Hundreds	Dozens
<b>Multi-table inserts</b>	Not supported	Supported
<b>Create table as select</b>	Not valid SQL-92	Supported

JDBC Listener started by default on EMR

Port 10003 for Hive 0.8.1, Ports 9100-9101 for Hadoop

Download `hive-jdbc-0.8.1.jar`

Access by:

*Open port on security group `ElasticMapReduce-master`  
SSH Tunnel*

```
./elastic-mapreduce -create  
--name "Hive job flow"  
--hive-script  
--args s3://myawsbucket/myquery.q  
--args -d,INPUT=s3://myawsbucket/input,-  
d,OUTPUT=s3://myawsbucket/output
```

HiveQL to execute

```
./elastic-mapreduce  
--create  
--alive  
--name "Hive job flow"  
--num-instances 5 --instance-type m1.Large \  
--hive-interactive
```



Interactive hive session

```
CREATE EXTERNAL TABLE impressions (
    requestBeginTime string,
    adId string,
    impressionId string,
    referrer string,
    userAgent string,
    userCookie string,
    ip string
)
PARTITIONED BY (dt string)
ROW FORMAT
    serde 'com.amazon.elasticmapreduce.JsonSerde'
        with serdeproperties ( 'paths'='requestBeginTime,
adId, impressionId, referrer, userAgent,          userCookie,
ip' )
LOCATION 's3://mybucketsource/tables/impressions' ;
```

```
CREATE EXTERNAL TABLE impressions (
    requestBeginTime string,
    adId string,
    impressionId string,
    referrer string,
    userAgent string,
    userCookie string,
    ip string
)
PARTITIONED BY (dt string)
ROW FORMAT
    serde 'com.amazon.elasticmapreduce.JsonSerde'
    with serdeproperties ( 'paths'='requestBeginTime,
adId, impressionId, referrer, userAgent,          userCookie,
ip' )
LOCATION 's3://mybucketsource/tables/impressions' ;
```

Table structure  
to create  
(happens fast  
as just  
mapping to  
source)

```
CREATE EXTERNAL TABLE impressions (
    requestBeginTime string,
    adId string,
    impressionId string,
    referrer string,
    userAgent string,
    userCookie string,
    ip string
)
PARTITIONED BY (dt string)
ROW FORMAT
    serde 'com.amazon.elasticmapreduce.JsonSerde'
        with serdeproperties ( 'paths'='requestBeg: Source data in
adId, impressionId, referrer, userAgent,
ip' )
LOCATION 's3://mybucketsource/tables/impressions' ;
```

```
hive> select * from impressions limit 5;
```



Selecting from source  
data directly via Hadoop

	Streaming	Hive	Pig	DynamoDB	Redshift
<b>Unstructured Data</b>	✓			✓	
<b>Structured Data</b>		✓	✓	✓	✓
<b>Language Support</b>	Any*	HQL	Pig Latin	Client	SQL
<b>SQL</b>		✓			✓
<b>Volume</b>	Unlimited	Unlimited	Unlimited	Relatively Low	1.6 PB
<b>Latency</b>	Medium	Medium	Medium	Ultra Low	Low

# Cost considerations

Making the most of EC2 resources

**1 instance for 100 hours**

**=**

**100 instances for 1 hour**

**Small instance = \$8**

1 instance for 1,000  
hours

=

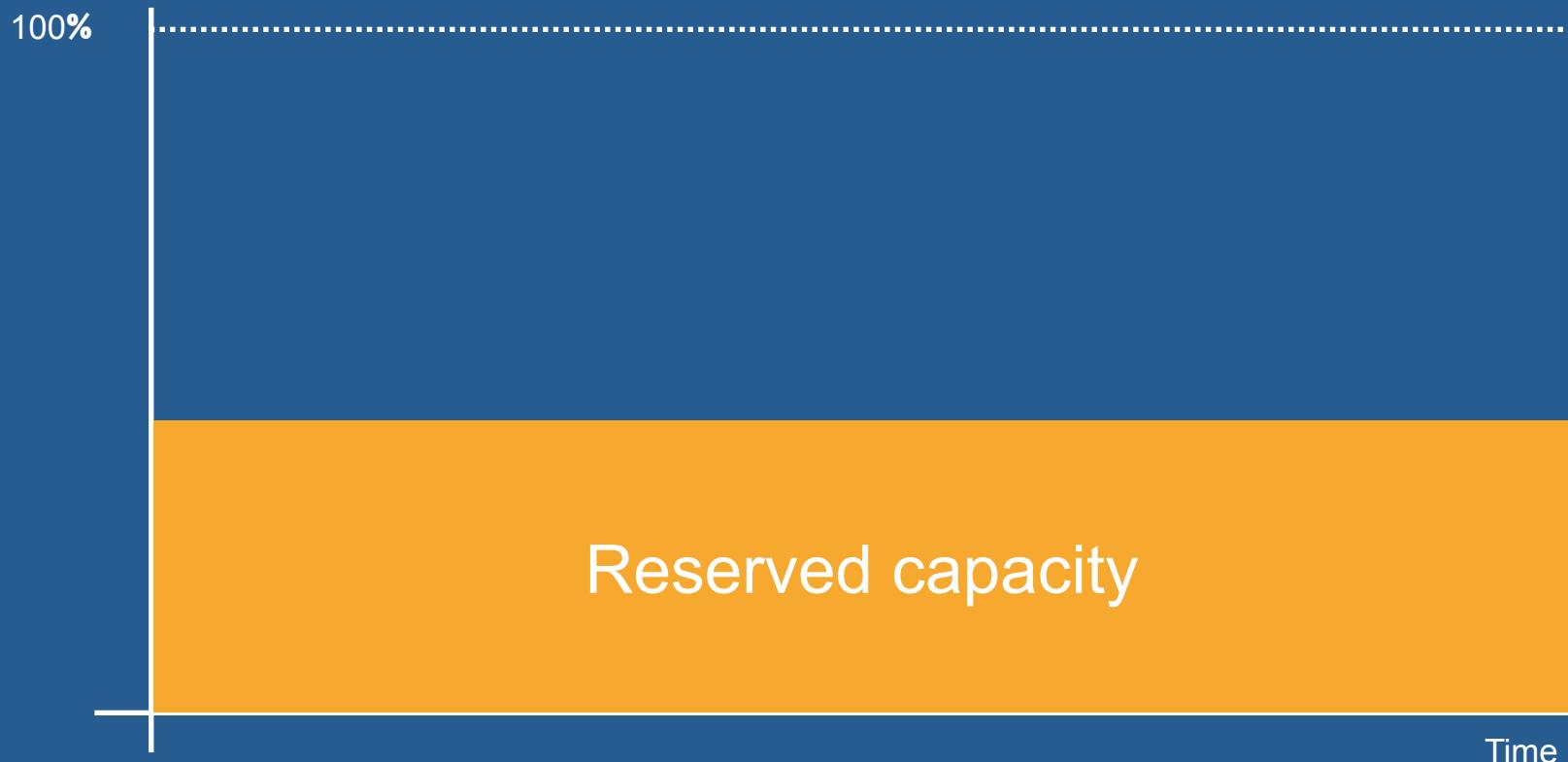
1,000 instances for 1  
hour

**Small instance = \$80**

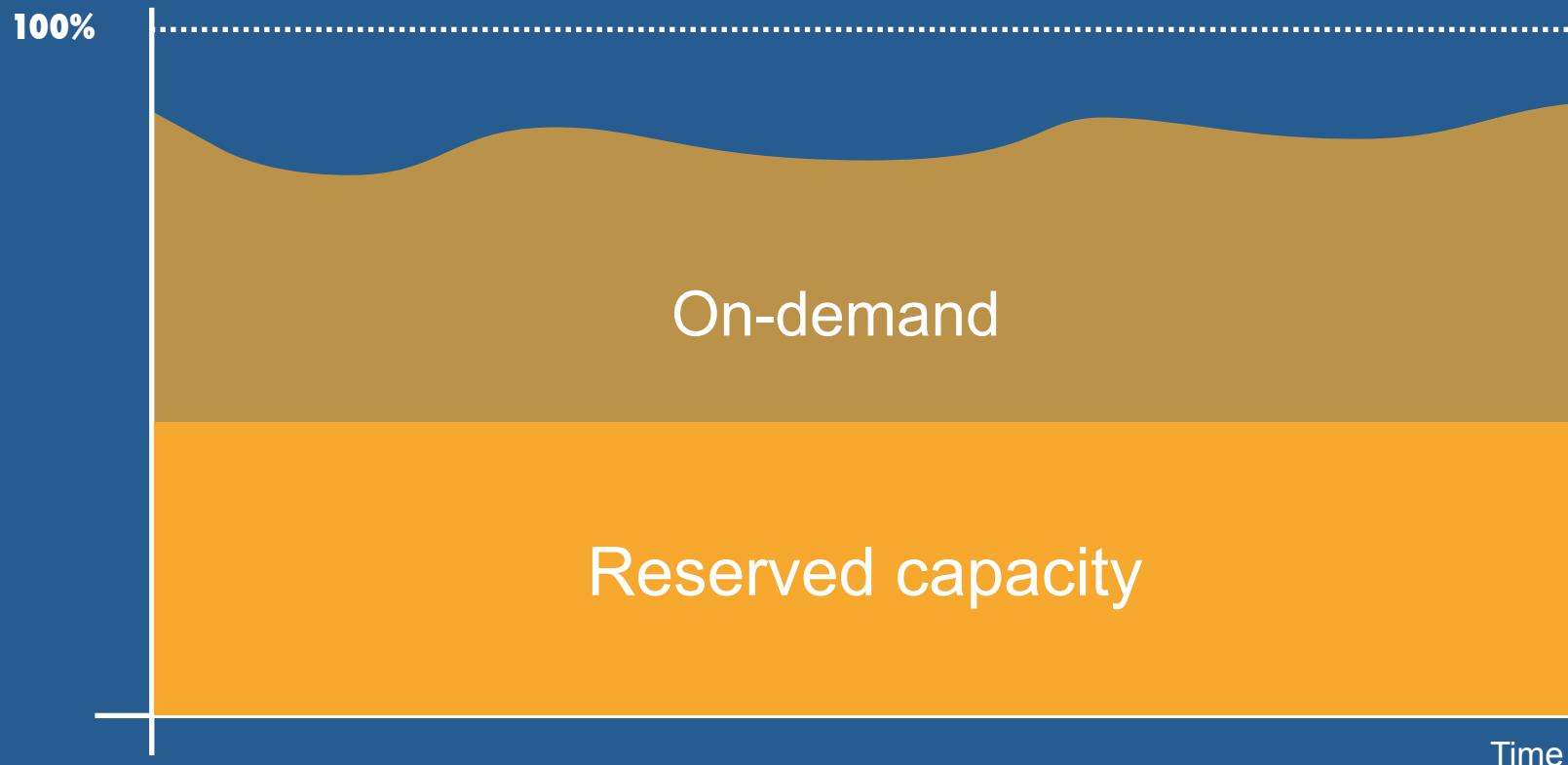
# Achieving economies of scale



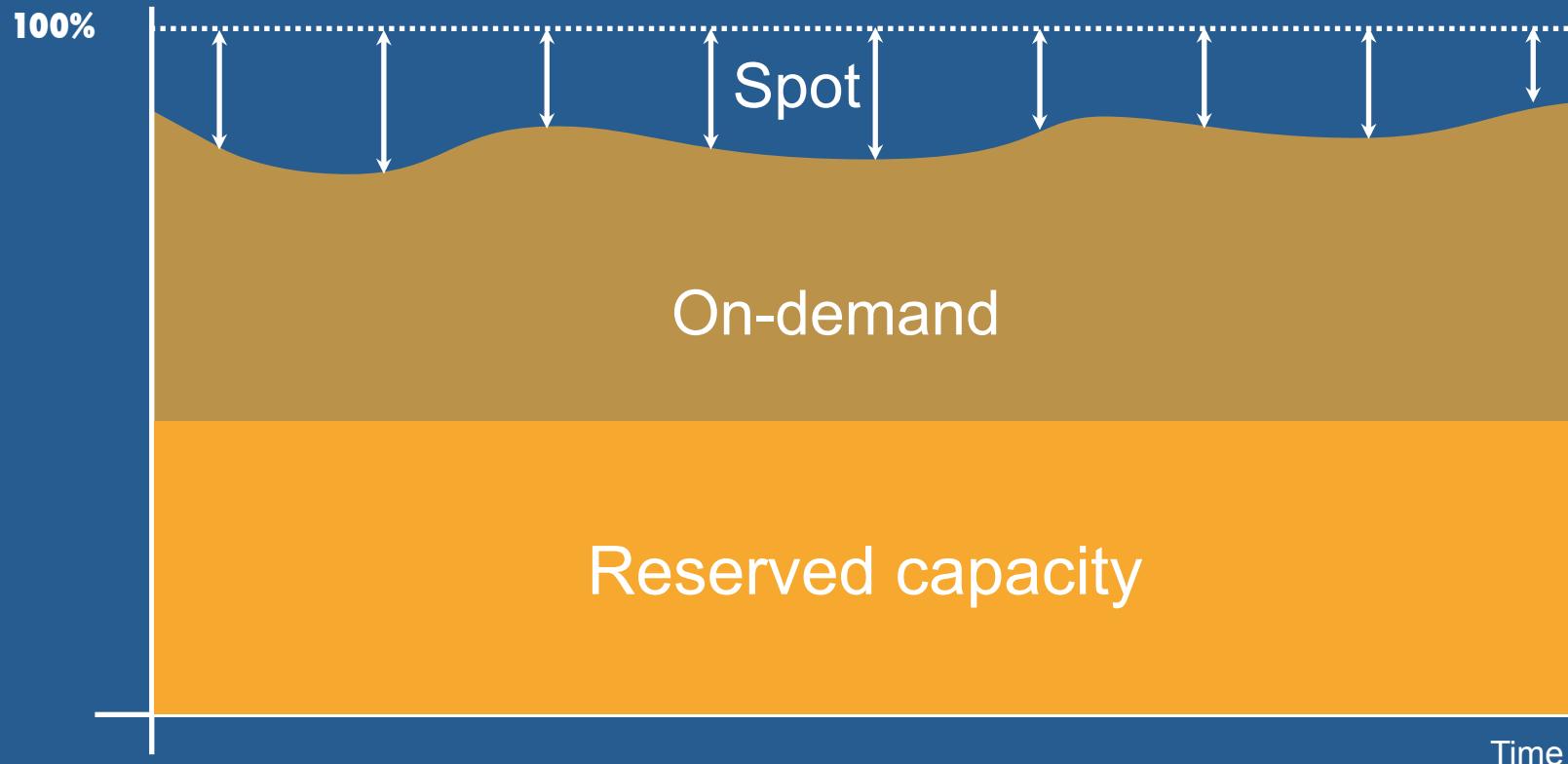
# Achieving economies of scale



# Achieving economies of scale



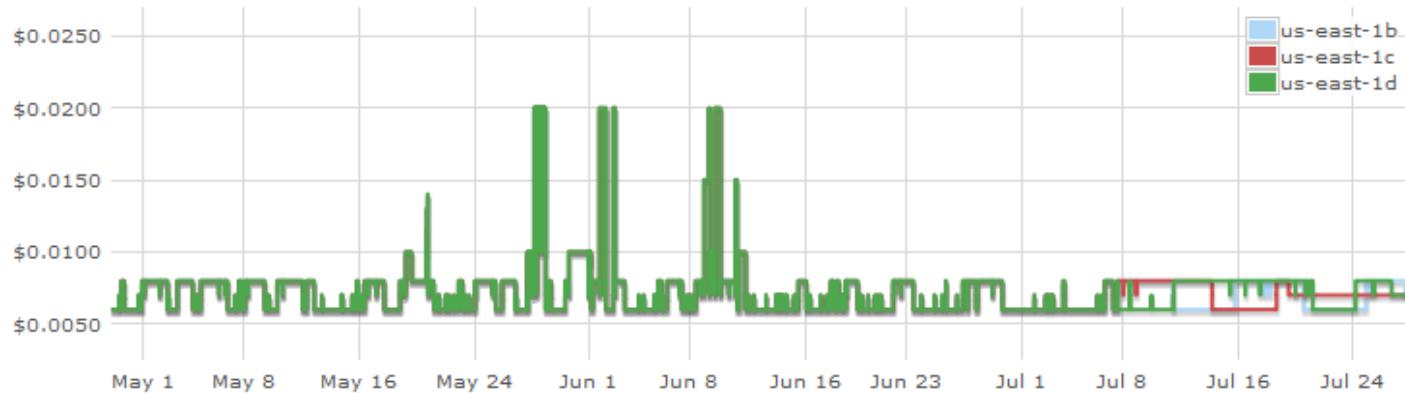
# Achieving economies of scale



## Spot Instance Pricing History

Cancel 

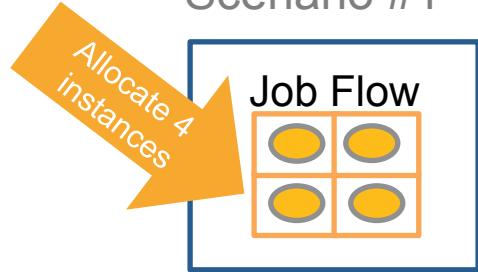
Product: Linux/UNIX ▾   Instance Type: t1.micro ▾   Date Range: 3 months ▾   Zone: All zones ▾



[Close](#)

# EMR with spot instances

## Scenario #1



Duration:

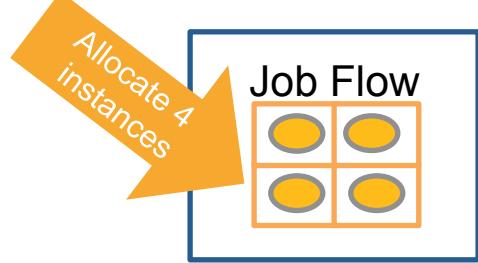
14 Hours

### #1: Cost without Spot

$$4 \text{ instances} * 14 \text{ hrs} * \$0.50 = \\ \$28$$

# EMR with spot instances

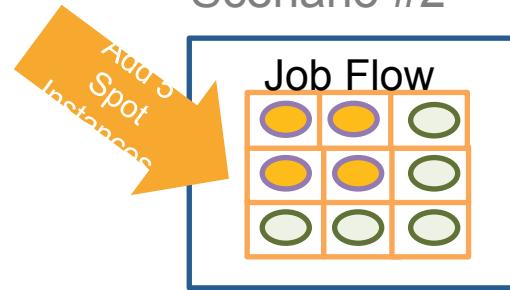
Scenario #1



Duration:

14 Hours

Scenario #2



Duration:

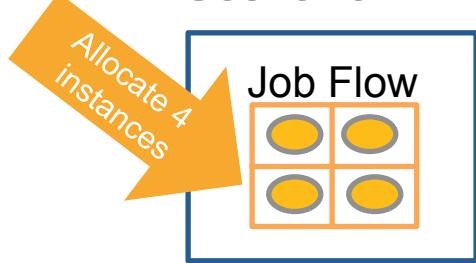
7 Hours

## #1: Cost without Spot

$$4 \text{ instances} * 14 \text{ hrs} * \$0.50 = \\ \$28$$

# EMR with spot instances

Scenario #1



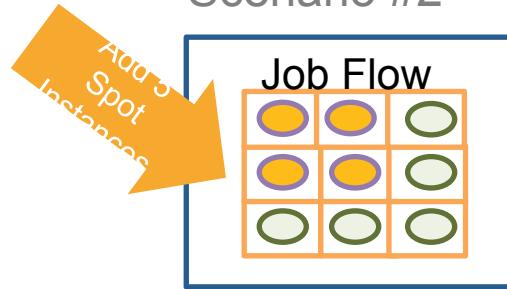
Duration:

14 Hours

## #1: Cost without Spot

$$4 \text{ instances} * 14 \text{ hrs} * \$0.50 = \\ \$28$$

Scenario #2



Duration:

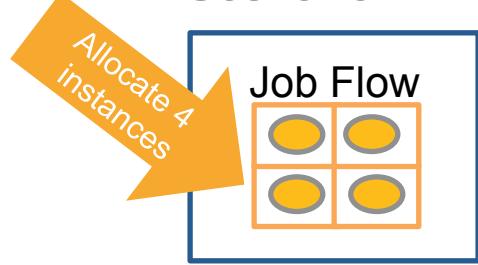
7 Hours

## #2: Cost with Spot

$$4 \text{ instances} * 7 \text{ hrs} * \$0.50 = \$14 + \\ 5 \text{ instances} * 7 \text{ hrs} * \$0.25 = \$8.75 \\ \text{Total} = \$22.75$$

# EMR with spot instances

Scenario #1



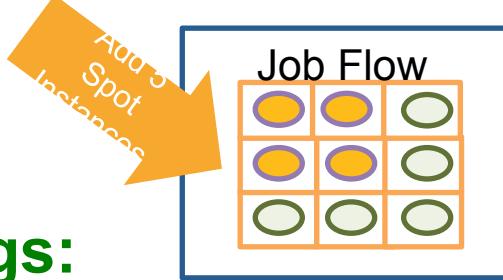
Duration:

14 Hours

## #1: Cost without Spot

4 instances \*14 hrs \* \$0.50 =  
\$28

Scenario #2



Duration:

7 Hours

**Time Savings:**

50%

**Cost Savings:**

~22%

## #2: Cost with Spot

4 instances \*7 hrs \* \$0.50 = \$14 +  
5 instances \* 7 hrs \* \$0.25 = \$8.75  
Total = \$22.75

# Some other things...

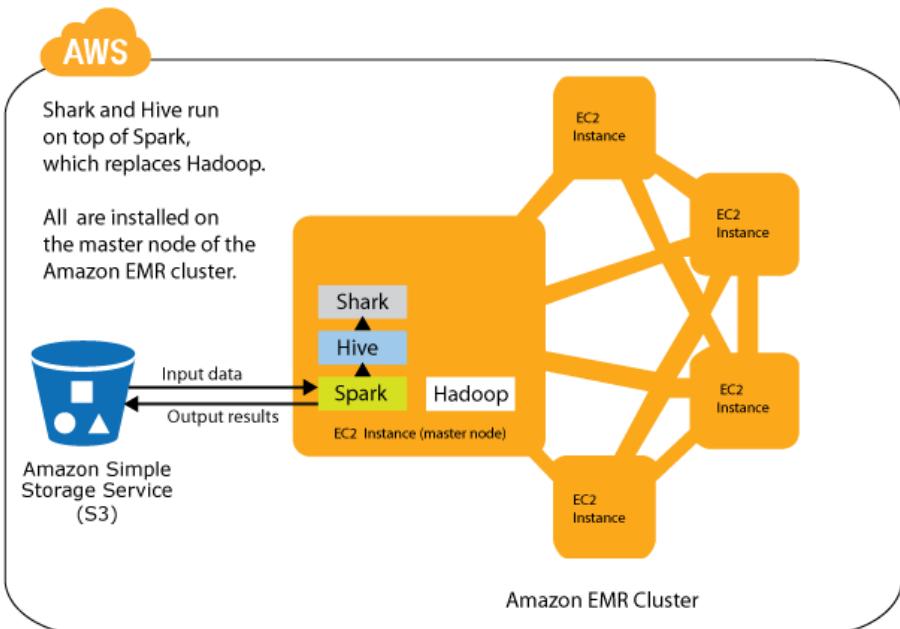
Hints and tips

## Managing big data

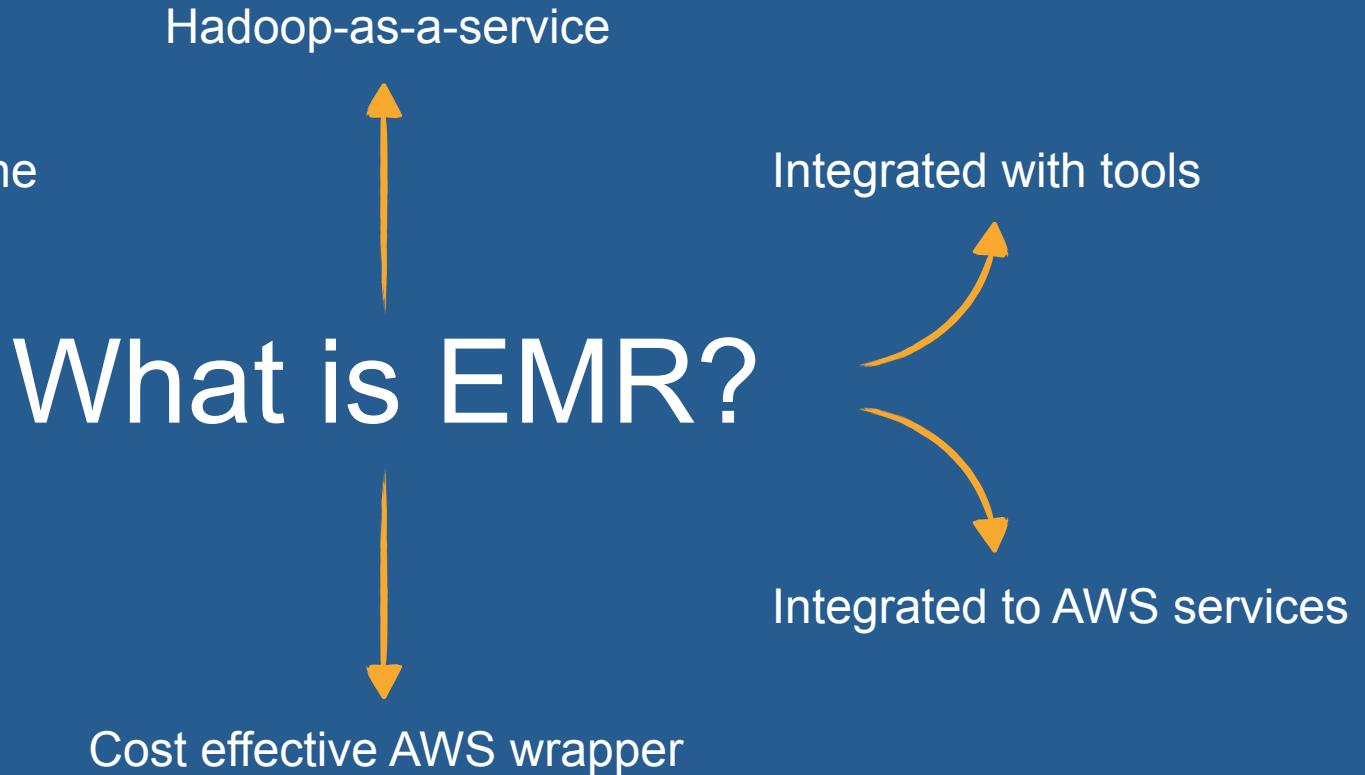
- ▶ Dynamic Creation of Clusters for Processing Jobs
- ▶ Bulk of Data Stored in S3
- ▶ Use Reduced Redundancy Storage for Lower Value / Re-creatable Data, and Lower Cost by 30%
- ▶ DynamoDB for High Performance Data Access
- ▶ Migrate data to Glacier for Archive and Periodic Access
- ▶ VPC & IAM for Network Security

Shark can return results up to 30 times faster than the same queries run on Hive.

```
elastic-mapreduce --create --alive --name "Spark/Shark Cluster"  
--bootstrap-action s3://elasticmapreduce/samples/spark/install-spark-shark.sh  
--bootstrap-name "install Mesos/Spark/Shark"  
--hive-interactive --hive-versions 0.7.1.3  
--ami-version 2.0  
--instance-type m1.xlarge --instance-count 3  
--key-pair ec2-keypair
```



# Summary



# Try the tutorial:

Contextual Advertising using Apache Hive and Amazon EMR

[aws.amazon.com/articles/2855](http://aws.amazon.com/articles/2855)

# Find out more:

[aws.amazon.com/big-data](http://aws.amazon.com/big-data)

<http://aws.amazon.com/elasticmapreduce/getting-started/>

Follow us for more  
events & webinars



Ian Massingham – Technical Evangelist

 @IanMmmm



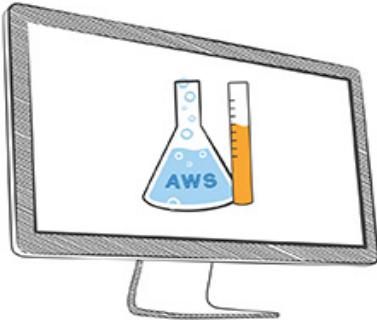
@AWS\_UKI for local AWS events & news



@AWScloud for Global AWS News and Announcements

# AWS Training & Certification

## Self-Paced Labs



Try products, gain new skills, and get hands-on practice working with AWS technologies

[aws.amazon.com/training/  
self-paced-labs](https://aws.amazon.com/training/self-paced-labs)

## Training



Skill up and gain confidence to design, develop, deploy and manage your applications on AWS

[aws.amazon.com/training](https://aws.amazon.com/training/)

## Certification



Demonstrate your skills, knowledge, and expertise with the AWS platform

[aws.amazon.com/certification](https://aws.amazon.com/certification/)

# We typically see customers start by trying our services

All Products

Compute & Networking

Storage

Database

Application Services

Development & Management

AWS Marketplace Software

FAQ »

Find answers to common questions about the AWS Free Tier.



## Amazon EC2 »

Web service that provides resizable compute capacity in the cloud.



## Amazon S3 »

Highly-scalable, reliable, and low-latency data storage.



## Amazon RDS »

Managed MySQL, Oracle and SQL Server databases.



## Amazon CloudWatch »

Monitoring for AWS cloud resources and applications.



## AWS Data Pipeline »

Orchestration for data-driven workflows.



## Amazon DynamoDB »

Fully managed NoSQL database service with seamless scalability.



## Amazon EBS »

Highly available, highly reliable, predictable storage volumes.



## Amazon ELB »

Web service that provides scalability and high availability.



## Amazon ElastiCache »

Managed scale-out caching.



## Amazon SNS »

Web service to set up, operate, and send notifications from the cloud.



## Amazon Elastic Transcoder »

Convert your media files easily, at low cost and at scale.



## Amazon SWF »

Workflow service for building scalable, resilient applications.



## AWS Marketplace »

Partner software pre-configured to run on AWS.

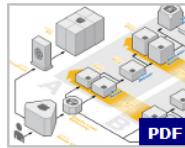


Get started now at : [aws.amazon.com/getting-started](https://aws.amazon.com/getting-started)

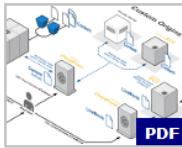
# Design your application for the AWS Cloud

## AWS Reference Architectures

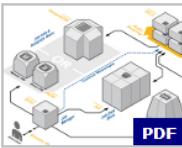
The flexibility of AWS allows you to design your application architectures the way you like. AWS Reference Architecture Datasheets provide you with the architectural guidance you need in order to build an application that takes full advantage of the AWS cloud. Each datasheet includes a visual representation of the architecture and basic description of how each service is used.



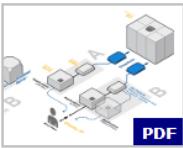
**Web Application Hosting**  
Build highly-scalable and reliable web or mobile-web applications ([PDF](#))



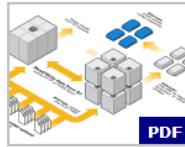
**Content and Media Serving**  
Build highly reliable systems that serve massive amounts of content and media ([PDF](#))



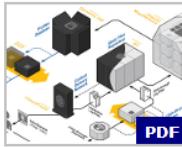
**Batch Processing**  
Build auto-scalable batch processing systems like video processing pipelines ([PDF](#))



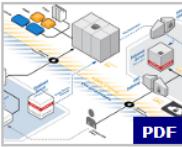
**Fault tolerance and High Availability**  
Build systems that quickly failover to new instances in an event of failure ([PDF](#))



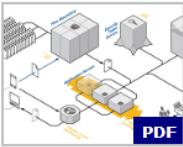
**Large Scale Processing and Huge Data sets**  
Build high-performance computing systems that involve Big Data ([PDF](#))



**Ad Serving**  
Build highly-scalable online ad serving solutions ([PDF](#))



**Disaster Recovery for Local Applications**  
Build cost-effective Disaster Recovery solutions for on-premises applications ([PDF](#))



**File Synchronization**  
Build simple file synchronization service ([PDF](#))

More details on the AWS Architecture Center at : [aws.amazon.com/architecture](http://aws.amazon.com/architecture)

