

Big Data Step-by-Step

Boston Predictive Analytics
Big Data Workshop

Microsoft New England Research &
Development Center, Cambridge, MA

Saturday, March 10, 2012

by **Jeffrey Breen**

President and Co-Founder
Atmosphere Research Group
email: jeffrey@atmosgrp.com
Twitter: @JeffreyBreen



<http://atms.gr/bigdata0310>

Big Data Infrastructure

Part 3: Taking it to the cloud... easily... with Whirr

Code & more on github:

<https://github.com/jeffreybreen/tutorial-201203-big-data>

Overview

- Download and install Apache whirr to our local Cloudera VM
- Use whirr to launch a Hadoop cluster on Amazon EC2
- Tell our local Hadoop tools to use the cluster instead of the local installation
- Run some tests
- How to use Hadoop's "distcp" to load data into HDFS from Amazon's S3 storage service
- Extra credit: save money with Amazon's spot instances

Heavy lifting by jclouds and Whirr

jclouds - <http://www.jclouds.org/>

“jclouds is an open source library that helps you get started in the cloud and reuse your java and clojure development skills. Our api allows you freedom to use portable abstractions or cloud-specific features. We test support of 30 cloud providers and cloud software stacks, including Amazon, GoGrid, Ninefold, vCloud, OpenStack, and Azure.”

jclouds™

Apache Whirr - <http://whirr.apache.org/>

“Apache Whirr is a set of libraries for running cloud services.

Whirr provides:

- A cloud-neutral way to run services. You don't have to worry about the idiosyncrasies of each provider.
- A common service API. The details of provisioning are particular to the service.
- Smart defaults for services. You can get a properly configured system running quickly, while still being able to override settings as needed.

You can also use Whirr as a command line tool for deploying clusters.”



Just what we want!



Whirr makes it look easy

- All you need is a simple config file

```
whirr.cluster-name=hadoop-ec2
whirr.instance-templates=1 hadoop-namenode+hadoop-jobtracker,15 hadoop-datanode+hadoop-tasktracker
whirr.provider=aws-ec2
whirr.identity=${env:AWS_ACCESS_KEY_ID}
whirr.credential=${env:AWS_SECRET_ACCESS_KEY}
whirr.hardware-id=m1.large
whirr.location-id=us-east-1
whirr.image-id=us-east-1/ami-49e32320
whirr.java.install-function=install_oab_java
whirr.hadoop.install-function=install_cdh_hadoop
whirr.hadoop.configure-function=configure_cdh_hadoop
```

- And one line to launch your cluster

```
$ ./whirr launch-cluster --config hadoop-ec2.properties
```

```
Bootstrapping cluster
Configuring template
Configuring template
Starting 3 node(s) with roles [hadoop-datanode, hadoop-tasktracker]
Starting 1 node(s) with roles [hadoop-namenode, hadoop-jobtracker]
```

One line?!? That's too easy! What didn't you show us?

- Download and install Whirr ($\geq 0.7.1$!)
- Specify your AWS security credentials
- Create a key pair to access the nodes
- Install R and add-on packages onto each node
- Configure VM to use cluster's Hadoop instance & run a proxy
- Copy data onto the cluster & run a test
- So... let's walk through those steps next...

Download & Install Whirr ($\geq 0.7.1$)

- Find an Apache mirror

<http://www.apache.org/dyn/closer.cgi/whirr/>

- From your VM's shell, download it with `wget`

```
$ wget http://apache.mirrors.pair.com/whirr/stable/whirr-0.7.1.tar.gz
```

- Installing is as simple as expanding the tarball

```
$ tar xf whirr-0.7.1.tar.gz
```

- Modify your path so this new version runs

```
$ export PATH="~/whirr-0.7.1/bin:$PATH"
```

```
$ whirr version
```

```
Apache Whirr 0.7.1
```

Amazon Login Info

- From AWS Management Console, look up your Access Keys
 - “Access Key ID” → `whirr.identity`
 - “Secret Access Key” → `whirr.credential`
- You *could* enter into Whirr’s config file, but please don’t
 - instead, just pick up environment variables in config file:

```
whirr.identity=${env:AWS_ACCESS_KEY_ID}
whirr.credential=${env:AWS_SECRET_ACCESS_KEY}
```

- and set them for your session session

```
$ export AWS_ACCESS_KEY_ID="your access key id here"
$ export AWS_SECRET_ACCESS_KEY="your secret access key here"
```

- While we’re at it, create a key pair

```
$ ssh-keygen -t rsa -P ""
```



Configuration file highlights

Specify how many nodes of each type

```
whirr.instance-templates=1 hadoop-namenode+hadoop-jobtracker,10 hadoop-datanode+hadoop-tasktracker
```

Select instance size & type (m1.large, c1.xlarge, m2.large, etc., as described at <http://aws.amazon.com/ec2/instance-types/>)

```
whirr.hardware-id=m1.large
```

Use a RightScale-published CentOS image (with transitory “instance” storage)

```
whirr.image-id=us-east-1/ami-49e32320
```

EC2 Amazon Machine Image: ami-49e32320

Description

Tags

AMI ID: ami-49e32320

AMI Name: RightImage_CentOS_5.6_x64_v5.7.14

Description: -

Source: rightscale-us-east/RightImage_CentOS_5.6_x64_v5.7.14.manifest.xml

Owner: 411009282317

Visibility: Public

Product Code:

State: available

Kernel ID: aki-825ea7eb

RAM Disk ID: -

Image Type: machine

Architecture: x86_64

Platform: Cent OS

Root Device Type: instance store

Root Device: /dev/sda1

Image Size: 0 bytes

Block Devices: N/A - Instance Store

Virtualization: paravirtual

Launch the Cluster

Yes, just one line... but then pages of output

```
$ whirr launch-cluster --config hadoop-ec2.properties
```

```
Bootstrapping cluster
```

```
Configuring template
```

```
Configuring template
```

```
Starting 1 node(s) with roles [hadoop-namenode, hadoop-jobtracker]
```

```
Starting 10 node(s) with roles [hadoop-datanode, hadoop-tasktracker]
```

```
[...]
```

```
Running configure phase script on: us-east-1/i-e301ab87
```

```
configure phase script run completed on: us-east-1/i-e301ab87
```

```
[...]
```

```
You can log into instances using the following ssh commands:
```

```
'ssh -i /home/cloudera/.ssh/id_rsa -o "UserKnownHostsFile /dev/null" -o StrictHostKeyChecking=no cloudera@107.22.25.82'
```

```
'ssh -i /home/cloudera/.ssh/id_rsa -o "UserKnownHostsFile /dev/null" -o StrictHostKeyChecking=no cloudera@204.236.222.162'
```

```
'ssh -i /home/cloudera/.ssh/id_rsa -o "UserKnownHostsFile /dev/null" -o StrictHostKeyChecking=no cloudera@23.20.97.157'
```

```
'ssh -i /home/cloudera/.ssh/id_rsa -o "UserKnownHostsFile /dev/null" -o StrictHostKeyChecking=no cloudera@75.101.192.112'
```

```
'ssh -i /home/cloudera/.ssh/id_rsa -o "UserKnownHostsFile /dev/null" -o StrictHostKeyChecking=no cloudera@50.16.43.91'
```

```
'ssh -i /home/cloudera/.ssh/id_rsa -o "UserKnownHostsFile /dev/null" -o StrictHostKeyChecking=no cloudera@107.22.84.246'
```

```
'ssh -i /home/cloudera/.ssh/id_rsa -o "UserKnownHostsFile /dev/null" -o StrictHostKeyChecking=no cloudera@23.20.134.238'
```

```
'ssh -i /home/cloudera/.ssh/id_rsa -o "UserKnownHostsFile /dev/null" -o StrictHostKeyChecking=no cloudera@107.22.61.144'
```

```
'ssh -i /home/cloudera/.ssh/id_rsa -o "UserKnownHostsFile /dev/null" -o StrictHostKeyChecking=no cloudera@23.20.6.74'
```

```
'ssh -i /home/cloudera/.ssh/id_rsa -o "UserKnownHostsFile /dev/null" -o StrictHostKeyChecking=no cloudera@174.129.137.89'
```

```
'ssh -i /home/cloudera/.ssh/id_rsa -o "UserKnownHostsFile /dev/null" -o StrictHostKeyChecking=no cloudera@107.21.77.224'
```

My Instances

Launch Instance Instance Actions Show/Hide Refresh Help

Viewing: All Instances All Instance Types Search 1 to 11 of 11 Instances

	Name	Instance	AMI ID	Root Device	Type	State	Status Checks	Monitoring	Security Groups	Key Pair Name
<input checked="" type="checkbox"/>	hadoop-ec2-b361b7d7	i-b361b7d7	ami-49e32320	instance store	m1.large	running	initializing...	basic	jclouds#hadoop-ec2#us-east-1	jclouds#hadoop-
<input type="checkbox"/>	hadoop-ec2-bf61b7db	i-bf61b7db	ami-49e32320	instance store	m1.large	running	initializing...	basic	jclouds#hadoop-ec2#us-east-1	jclouds#hadoop-
<input type="checkbox"/>	hadoop-ec2-b961b7dd	i-b961b7dd	ami-49e32320	instance store	m1.large	running	initializing...	basic	jclouds#hadoop-ec2#us-east-1	jclouds#hadoop-
<input type="checkbox"/>	hadoop-ec2-bb61b7df	i-bb61b7df	ami-49e32320	instance store	m1.large	running	initializing...	basic	jclouds#hadoop-ec2#us-east-1	jclouds#hadoop-
<input type="checkbox"/>	hadoop-ec2-8561b7e1	i-8561b7e1	ami-49e32320	instance store	m1.large	running	initializing...	basic	jclouds#hadoop-ec2#us-east-1	jclouds#hadoop-
<input type="checkbox"/>	hadoop-ec2-8761b7e3	i-8761b7e3	ami-49e32320	instance store	m1.large	running	initializing...	basic	jclouds#hadoop-ec2#us-east-1	jclouds#hadoop-

EC2 Instance: hadoop-ec2-b361b7d7 (i-b361b7d7) ec2-184-72-94-100.compute-1.amazonaws.com

Description Status Checks Monitoring Tags

AMI: RightImage_CentOS_5.6_x64_v5.7.14 (ami-49e32320) Security Groups: jclouds#hadoop-ec2#us-east-1 State: running Owner: 581302678308 Subnet ID: - Virtualization: paravirtual Reservation: r-c387b6a3 Platform: -	Zone: us-east-1b Type: m1.large Scheduled Events: No scheduled events VPC ID: - Source/Dest. Check: Placement Group: RAM Disk ID: - Key Pair Name: jclouds#hadoop-ec2#us-east-1#95
---	---

Security Groups

Create Security Group Delete Show/Hide Refresh Help

Viewing: EC2 Security Groups Search 1 to 8 of 8 Items

	Name	VPC ID	Description
<input checked="" type="checkbox"/>	jclouds#hadoop-ec2#us-east-1		jclouds#hadoop-ec2#us-east-1

1 Security Group selected

Security Group: jclouds#hadoop-ec2#us-east-1

Details Inbound

Create a new rule: Custom TCP rule

Port range: (e.g., 80 or 49152-65535)

Source: 0.0.0.0/0 (e.g., 192.168.2.0/24, sg-47ad482e, or 1234567890/default)

Add Rule

Apply Rule Changes

ICMP	Port (Service)	Source	Action
	ALL	sg-c38d5aab (jclouds#hadoop-ec2#us-east-1)	Delete
TCP	Port (Service)	Source	Action
	1 - 65535	sg-c38d5aab (jclouds#hadoop-ec2#us-east-1)	Delete
	22 (SSH)	0.0.0.0/0	Delete
	8020	107.21.192.85/32	Delete
	8021	107.21.192.85/32	Delete
	50030	74.104.166.41/32	Delete
	50070	74.104.166.41/32	Delete
UDP			

Install R and Packages

- `install-r+packages.sh` contains code to download and install R, plyr, rmr and their prerequisites

- whirr will run scripts on each node for us

```
$ whirr run-script --script install-r+packages.sh --config hadoop-ec2.properties
```

- And then you get to see **pages** and **pages** of output for **each** and **every** node!

```
** Node us-east-1/i-eb01ab8f: [10.124.18.198, 107.21.77.224]
rightscale-epel | 951 B 00:00
Setting up Install Process
Resolving Dependencies
--> Running transaction check
---> Package R.x86_64 0:2.14.1-1.el5 set to be updated
--> Processing Dependency: libRmath-devel = 2.14.1-1.el5 for package: R
---> Package R-devel.i386 0:2.14.1-1.el5 set to be updated
--> Processing Dependency: R-core = 2.14.1-1.el5 for package: R-devel
[...]
```

- Hopefully it ends with something positive like

```
* DONE (rmr)
Making packages.html ... done
```

install-r+packages.sh

```
sudo yum -y --enablerepo=epel install R R-devel
```

```
sudo R --no-save << EOF
```

```
install.packages(c('RJSONIO', 'itertools', 'digest', 'plyr'), repos="http://cran.revolutionanalytics.com", INSTALL_opts=c('--byte-compile')) )
```

```
EOF
```

```
# install latest version of the rmr package from RHadoop's github repository:
```

```
branch=master
```

```
wget --no-check-certificate https://github.com/RevolutionAnalytics/RHadoop/tarball/$branch -O - | tar zx  
mv RevolutionAnalytics-RHadoop* RHadoop
```

```
sudo R CMD INSTALL --byte-compile RHadoop/rmr/pkg/
```

```
sudo su << EOF1
```

```
cat >> /etc/profile <<EOF
```

```
export HADOOP_HOME=/usr/lib/hadoop
```

```
EOF
```

```
EOF1
```


Switch from local to cluster Hadoop

- CDH uses linux's alternatives facility to specify the location of the current configuration files

```
$ sudo /usr/sbin/alternatives --display hadoop-0.20-conf
hadoop-0.20-conf - status is manual.
  link currently points to /etc/hadoop-0.20/conf.pseudo
  /etc/hadoop-0.20/conf.empty - priority 10
  /etc/hadoop-0.20/conf.pseudo - priority 30
Current `best' version is /etc/hadoop-0.20/conf.pseudo.
```

- Whirr generates the config file we need to create a “conf.ec2” alternative

```
$ sudo mkdir /etc/hadoop-0.20/conf.ec2
$ sudo cp -r /etc/hadoop-0.20/conf.empty /etc/hadoop-0.20/conf.ec2
$ sudo rm -f /etc/hadoop-0.20/conf.ec2/*-site.xml
$ sudo cp ~/.whirr/hadoop-ec2/hadoop-site.xml /etc/hadoop-0.20/conf.ec2/
$ sudo /usr/sbin/alternatives --install /etc/hadoop-0.20/conf hadoop-0.20-conf /etc/hadoop-0.20/conf.ec2 30
$ sudo /usr/sbin/alternatives --set hadoop-0.20-conf /etc/hadoop-0.20/conf.ec2
$ sudo /usr/sbin/alternatives --display hadoop-0.20-conf
hadoop-0.20-conf - status is manual.
  link currently points to /etc/hadoop-0.20/conf.ec2
  /etc/hadoop-0.20/conf.empty - priority 10
  /etc/hadoop-0.20/conf.pseudo - priority 30
  /etc/hadoop-0.20/conf.ec2 - priority 30
Current `best' version is /etc/hadoop-0.20/conf.pseudo.
```

Fire up a proxy connection

- Whirr generates a proxy to connect your VM to the cluster

```
$ ~/.whirr/hadoop-ec2/hadoop-proxy.sh
```

```
Running proxy to Hadoop cluster at ec2-107-21-77-224.compute-1.amazonaws.com.  
Use Ctrl-c to quit.
```

```
Warning: Permanently added 'ec2-107-21-77-224.compute-1.amazonaws.com,  
107.21.77.224' (RSA) to the list of known hosts.
```

- Any hadoop commands executed on your VM should go to the cluster instead

```
$ hadoop dfsadmin -report
```

```
Configured Capacity: 4427851038720 (4.03 TB)
```

```
Present Capacity: 4144534683648 (3.77 TB)
```

```
DFS Remaining: 4139510718464 (3.76 TB)
```

```
DFS Used: 5023965184 (4.68 GB)
```

```
DFS Used%: 0.12%
```

```
Under replicated blocks: 0
```

```
Blocks with corrupt replicas: 0
```

```
Missing blocks: 0
```

```
[...]
```



Definitely not in
Kansas anymore

Test Hadoop with a small job

Download my fork of Jonathan Seidman's sample R code from github

```
$ mkdir hadoop-r
$ cd hadoop-r
$ git init
$ git pull git://github.com/jeffreybreen/hadoop-R.git
```

Grab first 1,000 lines from ASA's 2004 airline data

```
$ curl http://stat-computing.org/dataexpo/2009/2004.csv.bz2 | bzip2 \
  | head -1000 > 2004-1000.csv
```

Make some directories in HDFS and load the data file

```
$ hadoop fs -mkdir /user/cloudera
$ hadoop fs -mkdir asa-airline
$ hadoop fs -mkdir asa-airline/data
$ hadoop fs -mkdir asa-airline/out
$ hadoop fs -put 2004-1000.csv asa-airline/data/
```

Run Jonathan's sample streaming job

```
$ cd airline/src/deptdelay_by_month/R/streaming
$ hadoop jar /usr/lib/hadoop/contrib/streaming/hadoop-streaming-*.jar \
  -input asa-airline/data -output asa-airline/out/dept-delay-month \
  -mapper map.R -reducer reduce.R -file map.R -file reduce.R
[...]
$ hadoop fs -cat asa-airline/out/dept-delay-month/part-00000
2004      1      973      UA      11.55293
```


distcp: using Hadoop to load its own data

```
$ hadoop distcp -D fs.s3n.awsAccessKeyId=$AWS_ACCESS_KEY_ID \  
  -D fs.s3n.awsSecretAccessKey=$AWS_SECRET_ACCESS_KEY \  
  s3n://asa-airline/data asa-airline
```

```
12/03/08 21:42:21 INFO tools.DistCp: srcPaths=[s3n://asa-airline/data]  
12/03/08 21:42:21 INFO tools.DistCp: destPath=asa-airline  
12/03/08 21:42:27 INFO tools.DistCp: sourcePathsCount=23  
12/03/08 21:42:27 INFO tools.DistCp: filesToCopyCount=22  
12/03/08 21:42:27 INFO tools.DistCp: bytesToCopyCount=1.5g  
12/03/08 21:42:31 INFO mapred.JobClient: Running job: job_201203082122_0002  
12/03/08 21:42:32 INFO mapred.JobClient: map 0% reduce 0%  
12/03/08 21:42:41 INFO mapred.JobClient: map 14% reduce 0%  
12/03/08 21:42:45 INFO mapred.JobClient: map 46% reduce 0%  
12/03/08 21:42:46 INFO mapred.JobClient: map 61% reduce 0%  
12/03/08 21:42:47 INFO mapred.JobClient: map 63% reduce 0%  
12/03/08 21:42:48 INFO mapred.JobClient: map 70% reduce 0%  
12/03/08 21:42:50 INFO mapred.JobClient: map 72% reduce 0%  
12/03/08 21:42:51 INFO mapred.JobClient: map 80% reduce 0%  
12/03/08 21:42:53 INFO mapred.JobClient: map 83% reduce 0%  
12/03/08 21:42:54 INFO mapred.JobClient: map 89% reduce 0%  
12/03/08 21:42:56 INFO mapred.JobClient: map 92% reduce 0%  
12/03/08 21:42:58 INFO mapred.JobClient: map 99% reduce 0%  
12/03/08 21:43:04 INFO mapred.JobClient: map 100% reduce 0%  
12/03/08 21:43:05 INFO mapred.JobClient: Job complete: job_201203082122_0002  
[...]
```

Are you sure you want to shut down?

- Unlike the EBS-backed instance we created in Part 2, when the nodes are gone, they're gone—including their data—so you need to copy your results out of the cluster's HDFS before you throw the switch
- You could use `hadoop fs -get` to copy to your local file system

```
$ hadoop fs -get asa-airline/out/dept-delay-month .
```

```
$ ls -lh dept-delay-month
```

```
total 1.0K
```

```
drwxr-xr-x 1 1120 games 102 Mar  8 23:06 _logs
```

```
-rw-r--r-- 1 1120 games  33 Mar  8 23:06 part-00000
```

```
-rw-r--r-- 1 1120 games   0 Mar  8 23:06 _SUCCESS
```

```
$ cat dept-delay-month/part-00000
```

```
2004          1          973          UA          11.55293
```

- Or you could have your programming language of choice save the results locally for you

```
save( dept.delay.month.df, file='out/dept.delay.month.RData' )
```

Say goodnight, Gracie

- control-c to close the proxy connection

```
$ ~/.whirr/hadoop-ec2/hadoop-proxy.sh
```

```
Running proxy to Hadoop cluster at ec2-107-21-77-224.compute-1.amazonaws.com. Use Ctrl-c to quit.
```

```
Warning: Permanently added 'ec2-107-21-77-224.compute-1.amazonaws.com,107.21.77.224' (RSA) to the list of known hosts.
```

```
^C
```

```
Killed by signal 2.
```

- Shut down the cluster

```
$ whirr destroy-cluster --config hadoop-ec2.properties
```

```
Starting to run scripts on cluster for phase destroyinstances: us-east-1/i-c901abad, us-east-1/i-ad01abc9, us-east-1/i-f901ab9d, us-east-1/i-e301ab87, us-east-1/i-d901abbd, us-east-1/i-c301aba7, us-east-1/i-dd01abb9, us-east-1/i-d101abb5, us-east-1/i-f101ab95, us-east-1/i-d501abb1
```

```
Running destroy phase script on: us-east-1/i-c901abad
```

```
[...]
```

```
Finished running destroy phase scripts on all cluster instances
```

```
Destroying hadoop-ec2 cluster
```

```
Cluster hadoop-ec2 destroyed
```

- Switch back to your local Hadoop

```
$ sudo /usr/sbin/alternatives --set hadoop-0.20-conf /etc/hadoop-0.20/conf.pseudo
```

Extra Credit: Use Spot Instances

Through the “`whirr.aws-ec2-spot-price`” parameter, Whirr even lets you bid for excess capacity

<http://aws.amazon.com/ec2/spot-instances/>

<http://aws.amazon.com/pricing/ec2/>

Region: US East (Virginia)	
Linux/UNIX Usage	
Standard On-Demand Instances	
Small (Default)	\$0.080 per Hour
Medium	\$0.160 per Hour
Large	\$0.320 per Hour
Extra Large	\$0.640 per Hour
Micro On-Demand Instances	
Micro	\$0.020 per Hour
HI-Memory On-Demand Instances	
Extra Large	\$0.450 per Hour
Double Extra Large	\$0.900 per Hour
Quadruple Extra Large	\$1.800 per Hour
HI-CPU On-Demand Instances	
Medium	\$0.165 per Hour
Extra Large	\$0.660 per Hour
Cluster Compute Instances	
Quadruple Extra Large	\$1.300 per Hour
Eight Extra Large	\$2.400 per Hour
Cluster GPU Instances	
Quadruple Extra Large	\$2.100 per Hour

Region: US East (Virginia)	
Linux/UNIX Usage	
Standard Spot Instances	
Small (Default)	\$0.027 per Hour
Medium	\$0.038 per Hour
Large	\$0.108 per Hour
Extra Large	\$0.216 per Hour
Micro Spot Instances	
Micro	\$0.006 per Hour
High-Memory Spot Instances	
Extra Large	\$0.153 per Hour
Double Extra Large	\$0.42 per Hour
Quadruple Extra Large	\$0.756 per Hour
High-CPU Spot Instances	
Medium	\$0.054 per Hour
Extra Large	\$0.216 per Hour
Cluster Compute Instances	
Quadruple Extra Large	\$0.45 per Hour
Eight Extra Large	\$0.54 per Hour
Cluster GPU Instances	
Quadruple Extra Large	\$0.665 per Hour

Whirr bids, waits, and launches

My Spot Instance Requests						
Request Spot Instances Cancel Pricing History						
Viewing: Open <input type="text" value="Search"/>						
	Request ID	Max Price	AMI ID	Instance	Type	Status
<input type="checkbox"/>	sir-02fe5c11	\$0.109	ami-49e32320		m1.large	open
<input type="checkbox"/>	sir-0aa24a11	\$0.109	ami-49e32320		m1.large	open
<input type="checkbox"/>	sir-55255611	\$0.109	ami-49e32320		m1.large	open
<input type="checkbox"/>	sir-5888b611	\$0.109				
<input type="checkbox"/>	sir-5dce9611	\$0.109				
<input type="checkbox"/>	sir-6650bc11	\$0.109				
<input type="checkbox"/>	sir-c11c2e11	\$0.109				
<input type="checkbox"/>	sir-d3eec011	\$0.109				
<input type="checkbox"/>	sir-ed560c11	\$0.109				
<input type="checkbox"/>	sir-fbb76411	\$0.109				
<input type="checkbox"/>	sir-ee1e5e12	\$0.109				

My Spot Instance Requests						
Request Spot Instances Cancel Pricing History						
Viewing: Active <input type="text" value="Search"/>						
	Request ID	Max Price	AMI ID	Instance	Type	Status
<input type="checkbox"/>	sir-02fe5c11	\$0.109	ami-49e32320	i-d101abb5	m1.large	active
<input type="checkbox"/>	sir-0aa24a11	\$0.109	ami-49e32320	i-dd01abb9	m1.large	active
<input type="checkbox"/>	sir-55255611	\$0.109	ami-49e32320	i-d501abb1	m1.large	active
<input type="checkbox"/>	sir-5888b611	\$0.109	ami-49e32320			
<input type="checkbox"/>	sir-5dce9611	\$0.109	ami-49e32320			
<input type="checkbox"/>	sir-6650bc11	\$0.109	ami-49e32320			
<input type="checkbox"/>	sir-c11c2e11	\$0.109	ami-49e32320			
<input type="checkbox"/>	sir-d3eec011	\$0.109	ami-49e32320			
<input type="checkbox"/>	sir-ed560c11	\$0.109	ami-49e32320			
<input type="checkbox"/>	sir-fbb76411	\$0.109	ami-49e32320			

My Instances						
Launch Instance Instance Actions						
Viewing: All Instances <input type="text" value="All Instance Types"/> <input type="text" value="Search"/>						
	Name	Instance	AMI ID	Root Device	Type	State
<input type="checkbox"/>	rstudio	i-4f59882b	ami-7385461a	ebs	t1.micro	stopped
<input checked="" type="checkbox"/>	empty	i-e301ab87	ami-49e32320	instance store	m1.large	running
<input type="checkbox"/>	empty	i-eb01ab8f	ami-49e32320	instance store	m1.large	running
<input type="checkbox"/>	empty	i-f101ab95	ami-49e32320	instance store	m1.large	running
<input type="checkbox"/>	empty	i-f901ab9d	ami-49e32320	instance store	m1.large	running
<input type="checkbox"/>	empty	i-c301aba7	ami-49e32320	instance store	m1.large	running
<input type="checkbox"/>	empty	i-c901abad	ami-49e32320	instance store	m1.large	running
<input type="checkbox"/>	empty	i-d501abb1	ami-49e32320	instance store	m1.large	running

Hey, big spender

10+1 m1.large nodes for 3 hours = \$3.56

Timestamp	Type	Operation	Instance ID	Spot Request ID	Max Price	Market Price	Charge
13:28 GMT	SpotUsage:m1.large	RunInstances:S0007	i-55bd0b31	sir-78552a11	0.109	0.108	0.108
13:28 GMT	SpotUsage:m1.large	RunInstances:S0007	i-49bd0b2d	sir-7f8ea011	0.109	0.108	0.108
13:29 GMT	SpotUsage:m1.large	RunInstances:S0007	i-31bd0b55	sir-cf2f9c11	0.109	0.108	0.108
13:29 GMT	SpotUsage:m1.large	RunInstances:S0007	i-47bd0b23	sir-75795e11	0.109	0.108	0.108
13:29 GMT	SpotUsage:m1.large	RunInstances:S0007	i-43bd0b27	sir-f2a61611	0.109	0.108	0.108
13:29 GMT	SpotUsage:m1.large	RunInstances:S0007	i-27bd0b43	sir-4e3a4411	0.109	0.108	0.108
13:29 GMT	SpotUsage:m1.large	RunInstances:S0007	i-41bd0b25	sir-dd1aa612	0.109	0.108	0.108
13:29 GMT	SpotUsage:m1.large	RunInstances:S0007	i-51bd0b35	sir-b20e8c11	0.109	0.108	0.108
13:29 GMT	SpotUsage:m1.large	RunInstances:S0007	i-23bd0b47	sir-17865a11	0.109	0.108	0.108
13:29 GMT	SpotUsage:m1.large	RunInstances:S0007	i-25bd0b41	sir-d6c96c11	0.109	0.108	0.108
13:29 GMT	SpotUsage:m1.large	RunInstances:S0007	i-3fbd0b5b	sir-63b76011	0.109	0.108	0.108
14:28 GMT	SpotUsage:m1.large	RunInstances:S0007	i-55bd0b31	sir-78552a11	0.109	0.108	0.108
14:28 GMT	SpotUsage:m1.large	RunInstances:S0007	i-49bd0b2d	sir-7f8ea011	0.109	0.108	0.108
14:29 GMT	SpotUsage:m1.large	RunInstances:S0007	i-31bd0b55	sir-cf2f9c11	0.109	0.108	0.108
14:29 GMT	SpotUsage:m1.large	RunInstances:S0007	i-47bd0b23	sir-75795e11	0.109	0.108	0.108
14:29 GMT	SpotUsage:m1.large	RunInstances:S0007	i-43bd0b27	sir-f2a61611	0.109	0.108	0.108
14:29 GMT	SpotUsage:m1.large	RunInstances:S0007	i-27bd0b43	sir-4e3a4411	0.109	0.108	0.108
14:29 GMT	SpotUsage:m1.large	RunInstances:S0007	i-41bd0b25	sir-dd1aa612	0.109	0.108	0.108
14:29 GMT	SpotUsage:m1.large	RunInstances:S0007	i-51bd0b35	sir-b20e8c11	0.109	0.108	0.108
14:29 GMT	SpotUsage:m1.large	RunInstances:S0007	i-23bd0b47	sir-17865a11	0.109	0.108	0.108
14:29 GMT	SpotUsage:m1.large	RunInstances:S0007	i-25bd0b41	sir-d6c96c11	0.109	0.108	0.108
14:29 GMT	SpotUsage:m1.large	RunInstances:S0007	i-3fbd0b5b	sir-63b76011	0.109	0.108	0.108
15:28 GMT	SpotUsage:m1.large	RunInstances:S0007	i-55bd0b31	sir-78552a11	0.109	0.108	0.108
15:28 GMT	SpotUsage:m1.large	RunInstances:S0007	i-49bd0b2d	sir-7f8ea011	0.109	0.108	0.108
15:29 GMT	SpotUsage:m1.large	RunInstances:S0007	i-31bd0b55	sir-cf2f9c11	0.109	0.108	0.108
15:29 GMT	SpotUsage:m1.large	RunInstances:S0007	i-47bd0b23	sir-75795e11	0.109	0.108	0.108
15:29 GMT	SpotUsage:m1.large	RunInstances:S0007	i-43bd0b27	sir-f2a61611	0.109	0.108	0.108
15:29 GMT	SpotUsage:m1.large	RunInstances:S0007	i-27bd0b43	sir-4e3a4411	0.109	0.108	0.108
15:29 GMT	SpotUsage:m1.large	RunInstances:S0007	i-41bd0b25	sir-dd1aa612	0.109	0.108	0.108
15:29 GMT	SpotUsage:m1.large	RunInstances:S0007	i-51bd0b35	sir-b20e8c11	0.109	0.108	0.108
15:29 GMT	SpotUsage:m1.large	RunInstances:S0007	i-23bd0b47	sir-17865a11	0.109	0.108	0.108
15:29 GMT	SpotUsage:m1.large	RunInstances:S0007	i-25bd0b41	sir-d6c96c11	0.109	0.108	0.108
15:29 GMT	SpotUsage:m1.large	RunInstances:S0007	i-3fbd0b5b	sir-63b76011	0.109	0.108	0.108
							\$3.56

Obligatory iPhone p0rn



`</infrastructure>`