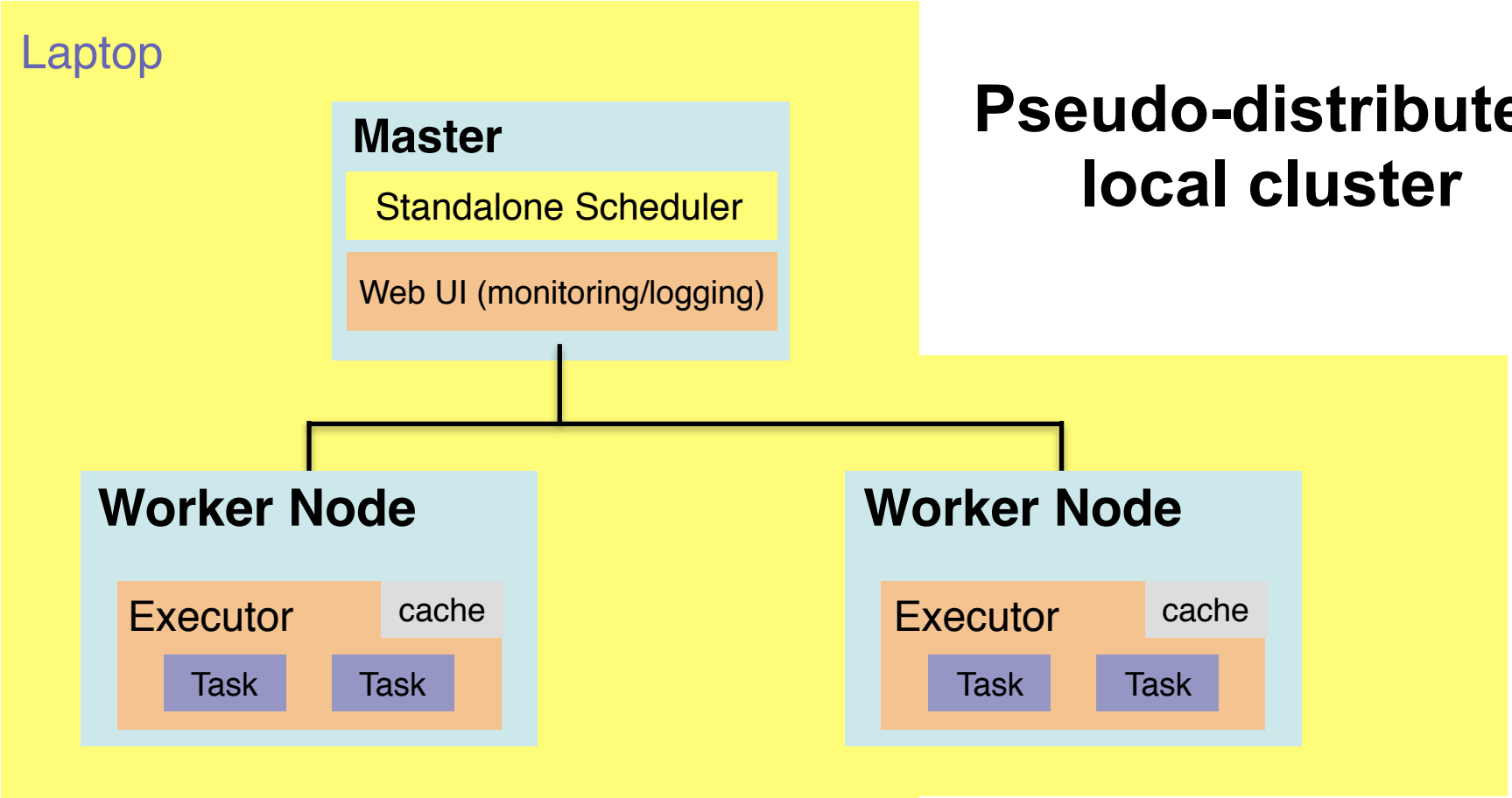
livelessons
video instruction from technology experts

Lesson 4: Spark Internals

4.8 Setting Up Your Own Cluster

Pseudo-distributed local cluster



Pseudo-distributed local cluster

Master

```
${SPARK_HOME}/bin/spark-class org.apache.spark.deploy.master.Master \
-h 127.0.0.1 \
-p 7077 \
--webui-port 8080
```

Workers (x 2)

```
${SPARK_HOME}/bin/spark-class org.apache.spark.deploy.worker.Worker \
-c 1 \
-m 1G \
spark://127.0.0.1:7077
```

Pseudo-distributed local cluster

One Master + two Workers: Run each process in separate terminal window

5. Master (java)

Master (java) | Worker 1 (java) | Worker 2 (java)

```
jonathan$ ${SPARK_HOME}/bin/spark-class org.apache.spark.deploy.master.Master -h 127.0.0.1 -p 7077 --webui-port 8080
15/08/12 12:56:48 INFO master.Master: Registered signal handlers for [TERM, HUP, INT]
15/08/12 12:56:49 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
15/08/12 12:56:49 INFO spark.SecurityManager: Changing view acls to: jonathandinu
15/08/12 12:56:49 INFO spark.SecurityManager: Changing modify acls to: jonathandinu
15/08/12 12:56:49 INFO spark.SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users with view permissions: Set(jonathandinu); users with modify permissions: Set(jonathandinu)
15/08/12 12:56:49 INFO slf4j.Slf4jLogger: Slf4jLogger started
15/08/12 12:56:49 INFO Remoting: Starting remoting
15/08/12 12:56:49 INFO Remoting: Remoting started; listening on addresses :[akka.tcp://sparkMaster@127.0.0.1:7077]
15/08/12 12:56:49 INFO util.Utils: Successfully started service 'sparkMaster' on port 7077.
15/08/12 12:56:49 INFO server.Server: jetty-8.y.z-SNAPSHOT
15/08/12 12:56:49 INFO server.AbstractConnector: Started SelectChannelConnector@localhost:6066
15/08/12 12:56:49 INFO util.Utils: Successfully started service on port 6066.
15/08/12 12:56:49 INFO rest.StandaloneRestServer: Started REST server for submitting applications on port 6066
15/08/12 12:56:49 INFO master.Master: Starting Spark master at spark://127.0.0.1:7077
15/08/12 12:56:49 INFO master.Master: Running Spark version 1.4.1
15/08/12 12:56:49 INFO server.Server: jetty-8.y.z-SNAPSHOT
15/08/12 12:56:49 INFO server.AbstractConnector: Started SelectChannelConnector@0.0.0.0:8080
15/08/12 12:56:49 INFO util.Utils: Successfully started service 'MasterUI' on port 8080.
15/08/12 12:56:49 INFO ui.MasterWebUI: Started MasterWebUI at http://10.3.35.20:8080
15/08/12 12:56:49 INFO master.Master: I have been elected leader! New state: ALIVE
15/08/12 12:56:59 INFO master.Master: Registering worker 10.3.35.20:59807 with 1 cores, 1024.0 MB RAM
15/08/12 12:56:59 INFO master.Master: Registering worker 10.3.35.20:59809 with 1 cores, 1024.0 MB RAM
```

EC2: Setup

1. Create AWS account: https://aws.amazon.com/account/

2. Get Access keys:

AWS | Services | Edit | Jonathan [redacted] | Global | Support

Dashboard

Details

Groups

Users

Roles

Policies

Identity Providers

Account Settings

Credential Report

Encryption Keys

My Account

Billing & Cost Management

Security Credentials

Sign Out

Your Security Credentials

Use this page to manage the credentials for your AWS account. To manage credentials for AWS Identity and Access Management [redacted].

To learn more about the types of AWS credentials and how they're used, see AWS Security Credentials in AWS General Referenc[redacted]

+ Password

+ Multi-Factor Authentication (MFA)

− Access Keys (Access Key ID and Secret Access Key)

You use access keys to sign programmatic requests to AWS services. To learn how to sign requests using your access keys, see the signing documentation. For your protection, store your access keys securely and do not share them. In addition, AWS recommends that you rotate your access keys every 90 days.

Note: You can have a maximum of two access keys (active or inactive) at a time.

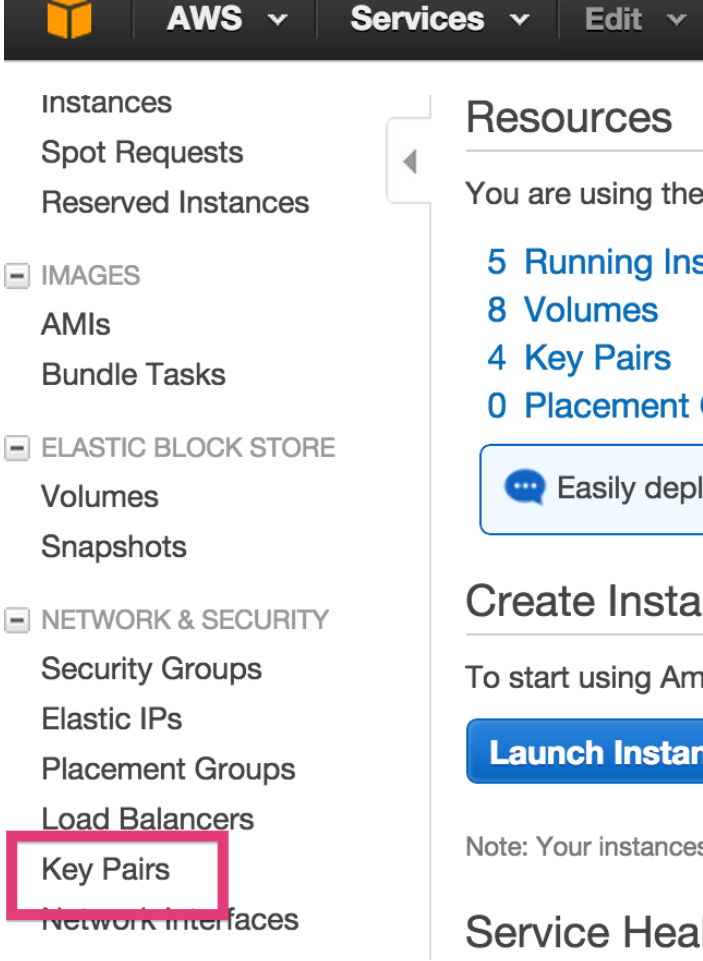
Created	Deleted	Access Key ID	Last Used	Last Used Region	Last Used Service	Status	Actions
Jul 8th 2014		[redacted]	2015-08-04 23:41 PDT	us-east-1	s3	Active	Make Inactive | Delete
Oct 28th 2014		[redacted]	2015-08-12 13:09 PDT	N/A	s3	Active	Make Inactive | Delete

EC2: Setup

1. Include the following lines in your `~/.bash_profile` (or `~/.bashrc`):

```
export AWS_ACCESS_KEY_ID=xxxxxxx
export AWS_SECRET_ACCESS_KEY=xxxxxx
```

2. Download EC2 `keypair`:



EC2: Launch

1. Launch EC2 cluster with script in `$SPARK_HOME/ec2`:

```
./spark-ec2 -k keyname -i ~/.ssh/keyname.pem
  --copy-aws-credentials
  --instance-type=m1.large
  -m m1.large
  -s 19 launch spark
```

```
Shutting down GANGLIA gmond:                               [FAILED]
Starting GANGLIA gmond:                                    [  OK  ]
Connection to ec2-             .compute-1.amazonaws.com closed.
Shutting down GANGLIA gmetad:                              [FAILED]
Starting GANGLIA gmetad:                                   [  OK  ]
Stopping httpd:                                            [FAILED]
Starting httpd: httpd: Syntax error on line 154 of /etc/httpd/conf/httpd.conf: Cannot load /etc/httpd/modules/mod_authz_core.so into server:
/etc/httpd/modules/mod_authz_core.so: cannot open shared object file: No such file or directory
                                                           [FAILED]
Connection to ec2            .compute-1.amazonaws.com closed.
Spark standalone cluster started at http://ec2            .compute-1.amazonaws.com:8080
Ganglia started at http://ec2            .compute-1.amazonaws.com:5080/ganglia
Done!
```

Launch Instance | Connect | Actions

Filter by tags and attributes or search by keyword

1 to 20 of 20

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS
spark-master…	i-8579db2c	m1.large	us-east-1b	running	2/2 checks …	None	ec2-[redacted].comp…
spark-slave-i…	i-6979dbc0	m1.large	us-east-1b	running	2/2 checks …	None	ec2-[redacted].co…
spark-slave-i…	i-6b79dbc2	m1.large	us-east-1b	running	2/2 checks …	None	ec2[redacted]co…
spark-slave-i…	i-6a79dbc3	m1.large	us-east-1b	running	2/2 checks …	None	ec2[redacted].co…
spark-slave-i…	i-6d79dbc4	m1.large	us-east-1b	running	2/2 checks …	None	ec2[redacted].co…
spark-slave-i…	i-6c79dbc5	m1.large	us-east-1b	running	2/2 checks …	None	ec2[redacted].co…
spark-slave-i…	i-6f79dbc6	m1.large	us-east-1b	running	2/2 checks …	None	ec2[redacted]co…
spark-slave-i…	i-6e79dbc7	m1.large	us-east-1b	running	2/2 checks …	None	ec2-[redacted].co…

Instance: i-8579db2c (spark-master-i-8579db2c) **Public DNS:** ec2-[redacted].compute-1.amazonaws.com

Description | Status Checks | Monitoring | Tags

Instance ID i-8579db2c

Instance state running

Public DNS ec2[redacted].compute-1.amazonaws.com

Public IP [redacted]

EC2: Scripts

Login to the Master

```
./spark-ec2 -k keyname -i ~/.ssh/keyname.pem login spark
```

Terminate Cluster

```
./spark-ec2 destroy spark
```

Stop cluster

```
./spark-ec2 stop spark
```

Restart cluster (after stopped)

```
./spark-ec2 -k keyname -i ~/.ssh/keyname.pem start spark
```

Setup IPython/Jupyter

Login to the Master

```
./spark-ec2 -k keyname -i ~/.ssh/keyname.pem login spark
```

Installed needed packages (on master)

```
# Install all the necessary packages on Master
yum install -y tmux
yum install -y pssh
yum install -y python27 python27-devel
yum install -y freetype-devel libpng-devel
wget https://bitbucket.org/pypa/setuptools/raw/bootstrap/ez_setup.py -O - | python27
```

```
easy_install-2.7 pip
easy_install py4j
pip2.7 install ipython==2.0.0
pip2.7 install pyzmq==14.6.0
pip2.7 install jinja2==2.7.3
pip2.7 install tornado==4.2
pip2.7 install numpy
pip2.7 install matplotlib
pip2.7 install nltk
```

Setup IPython/Jupyter

Login to the Master

```
./spark-ec2 -k keyname -i ~/.ssh/keyname.pem login spark
```

Installed needed packages (on workers)

```
# Install all the necessary packages on Workers

pssh -h /root/spark-ec2/slaves yum install -y python27 python27-devel
pssh -h /root/spark-ec2/slaves "wget https://bitbucket.org/pypa/setuptools/raw/bootstrap/ez_setup.py -O - | python27"
pssh -h /root/spark-ec2/slaves easy_install-2.7 pip
pssh -t 10000 -h /root/spark-ec2/slaves pip2.7 install numpy
pssh -h /root/spark-ec2/slaves pip2.7 install nltk
```

Allow inbound requests to enable IPython/Jupyter notebook (WARNING: this will create a security risk however)

AMIs
Bundle Tasks

ELASTIC BLOCK STORE
Volumes
Snapshots

NETWORK & SECURITY
Security Groups
Elastic IPs
Placement Groups
Load Balancers
Key Pairs
Network Interfaces

Name	Group ID	Group Name	VPC ID	Description
	[redacted]	[redacted]	[redacted]	Spark EC2 group
				Spark EC2 group
				Spark EC2 group
				Spark EC2 group

Edit

Type	Protocol	Port Range	Source
Custom TCP Rule	TCP	8888	0.0.0.0/0
All ICMP	All	N/A	[redacted]

IPython/Jupyter Profile

Login to the Master

```
./spark-ec2 -k keyname -i ~/.ssh/keyname.pem login spark
```

Set notebook password

```
ipython profile create default

python -c "from IPython.lib import passwd; print passwd()" \
        > /root/.ipython/profile_default/nbpasswd.txt

cat /root/.ipython/profile_default/nbpasswd.txt
# sha1:128de302ca73:6b9a8bd5bhjde33d48cd65ad9cafb0770c13c9df
```

Configure IPython/Jupyter Settings

/root/.ipython/profile_default/ipython_notebook_config.py:

```
# Configuration file for ipython-notebook.
c = get_config()

# Notebook config
c.NotebookApp.ip = '*'
c.NotebookApp.open_browser = False
# It is a good idea to put it on a known, fixed port
c.NotebookApp.port = 8888

PWDFILE="/root/.ipython/profile_default/nbpasswd.txt"
c.NotebookApp.password = open(PWDFILE).read().strip()
```

Configure IPython/Jupyter Settings

/root/.ipython/profile_default/startup/pyspark.py:

```
# Configure the necessary Spark environment
import os
os.environ['SPARK_HOME'] = '/root/spark/'

# And Python path
import sys
sys.path.insert(0, '/root/spark/python')

# Detect the PySpark URL
CLUSTER_URL = open('/root/spark-ec2/cluster-url').read().strip()
```

Configure IPython/Jupyter Settings

Add the following to `/root/spark/conf/spark-env.sh`**:**

```
export PYSPARK_PYTHON=python2.7
```

Sync across workers:

```
~/spark-ec2/copy-dir /root/spark/conf
```

Make sure master's `env` **is correct**

```
source /root/spark/conf/spark-env.sh
```

IPython/Jupyter initialization (on master)

Start remote window manager (`screen` or `tmux`):

```
screen
```

Start notebook server:

```
ipython notebook
```

Detach from session:

```
Ctrl-a d
```

IPython/Jupyter login

On your laptop:

```
http://[YOUR MASTER IP/DNS HERE]:8888
```

IP[y]: Notebook

Password: Log in

IPython/Jupyter login

Test that it all works:

IP[y]: Notebook spark-notebook Last Checkpoint: Aug 12 17:10 (unsaved changes) Logout

File Edit View Insert Cell Kernel Help

Code Cell Toolbar: None

```
In [1]: print CLUSTER_URL

In [2]: import pyspark as ps

In [3]: sc = ps.SparkContext( CLUSTER_URL, 'pyspark')

In [4]: sc.parallelize([1,2,3]).count()
Out[4]: 3
```

EC2: Data

HDFS (ephemeral)

`/root/ephemeral-hdfs`

HDFS (persistent)

`/root/persistent-hdfs`

Amazon S3

`s3n://bucket_name`

Review

- Local mode or cluster mode each have their benefits
- Spark can be run on a variety of cluster managers
- Amazon EC2 enables elastic scaling and ease of development
- By leveraging IPython/Jupyter you can get the performance of a cluster with the ease of interactive development

Next Up: Spark Performance Tuning