Hadoop and AWS
Developing with Hadoop in the AWS cloud

- **Hadoop** is Linux based.
- You can install Linux at home and run these examples.
- We will create a Linux instance using **AWS** and **EC2** to run our code.
• Log in to your **AWS** account.

• Select the **EC2** service.
• Click on **Launch Instance**
• Click on **Quick Launch Wizard**

• Select **Ubuntu Server 14.04 LTS**
• Click on Review and Launch.

### Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more about instance types and how they can meet your computing needs.](#)

Filter by: All instance types ▼ Current generation ▼ Show/Hide Columns

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GB memory, EBS only)

*Note: T2 instances are VPC-only. Your T2 instance will launch into your VPC. [Learn more about T2 and VPC.](#)*

<table>
<thead>
<tr>
<th>Family</th>
<th>Type</th>
<th>vCPUs</th>
<th>Memory (GB)</th>
<th>Instance Storage (GB)</th>
<th>EBS Optimized Available</th>
<th>Network Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>General purpose</td>
<td>t2.micro</td>
<td>1</td>
<td>1</td>
<td>EBS only</td>
<td>-</td>
<td>Low to Moderate</td>
</tr>
<tr>
<td>General purpose</td>
<td>t2.small</td>
<td>1</td>
<td>2</td>
<td>EBS only</td>
<td>-</td>
<td>Low to Moderate</td>
</tr>
<tr>
<td>General purpose</td>
<td>t2.medium</td>
<td>2</td>
<td>4</td>
<td>EBS only</td>
<td>-</td>
<td>Low to Moderate</td>
</tr>
<tr>
<td>General purpose</td>
<td>m3.medium</td>
<td>1</td>
<td>3.75</td>
<td>1 x 32 (SSD)</td>
<td>-</td>
<td>Moderate</td>
</tr>
<tr>
<td>General purpose</td>
<td>m3.large</td>
<td>2</td>
<td>7.5</td>
<td>1 x 48 (SSD)</td>
<td>Yes</td>
<td>High</td>
</tr>
<tr>
<td>General purpose</td>
<td>m3.xlarge</td>
<td>4</td>
<td>10</td>
<td>2 x 48 (SSD)</td>
<td>Yes</td>
<td>Moderate</td>
</tr>
<tr>
<td>General purpose</td>
<td>m3.2xlarge</td>
<td>8</td>
<td>30</td>
<td>2 x 80 (SSD)</td>
<td>Yes</td>
<td>High</td>
</tr>
<tr>
<td>General purpose</td>
<td>c3.large</td>
<td>2</td>
<td>9.75</td>
<td>2 x 16 (SSD)</td>
<td>-</td>
<td>Moderate</td>
</tr>
<tr>
<td>General purpose</td>
<td>c3.xlarge</td>
<td>4</td>
<td>7.5</td>
<td>2 x 48 (SSD)</td>
<td>Yes</td>
<td>Moderate</td>
</tr>
<tr>
<td>General purpose</td>
<td>c3.2xlarge</td>
<td>8</td>
<td>15</td>
<td>2 x 80 (SSD)</td>
<td>Yes</td>
<td>High</td>
</tr>
</tbody>
</table>

[Review and Launch](#)
• Click on **Launch** to start the instance (this can take a few seconds).
• Create a new key pair.
• Give it a name.
• Click **Download Key Pair** and save the file somewhere you can find it easily.

• Click **Launch Instance**.
• Click **View Instance**.

### Launch Status

- Your instance is now launching
- The following instance launch has been initiated: `i-07020c62`

**Get notified of estimated charges**
- Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).

### How to connect to your instance

Your instance is launching, and it may take a few minutes until it is in the **running** state, when it will be ready for you to use. Usage hours on your new instance will start immediately and continue to accrue until you stop or terminate your instance.

Click **View Instances** to monitor your instance's status. Once your instance is in the **running** state, you can connect to it from the instances screen. Find out how to connect to your instance.

### Here are some helpful resources to get you started

- How to connect to your Linux instance
- Learn about AWS Free Usage Tier
- Amazon EC2 User Guide
- Amazon EC2 Discussion Forum

While your instances are launching you can also

- Create status check alarms to be notified when these instances fail status checks. (Additional charges may apply)
- Create and attach additional EBS volumes. (Additional charges may apply)
- Manage security groups

[View Instances]
• Click the instance (it'll have a green light next to it), to display information about it.

Our instance is now running.

This will be important in a minute

• Click on the **Security Groups** link.
• Select the 'quicklaunch-1' group.

• Select the 'Inbound' tab.
Make sure you have this rule. We'll be logging in through **port 22** in a minute.
EC2 Dashboard

Instances
Spot Requests
Reserved Instances

Images
AMIs
Bundle Tasks

Elastic Block Store
Volumes
Snapshots

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

Launch Instance

Connect

Actions

Filter:

All instances

All instance types

Search Instances

1 to 1 of 1 Instances

Name
Instance ID
Instance Type
Availability Zone
Instance State
Status Checks
Alarm Status

i-3193ce49
i1.micro
us-east-1b
running
Initializing
None

Create Status Check Alarm

System Status Checks

These checks monitor the AWS systems required to use this instance and ensure they are functioning properly.

Instance Status Checks

These checks monitor your software and network configuration for this instance.

Additional Resources

© 2008 - 2013, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy  Terms of Use
• Select the Java SSH Client option.
• Enter the path to the key pair file you downloaded, i.e. right-click on the file if you're not sure.
PC 15 version 2.1

APPGATE NETWORK SECURITY AB ("APPGATE")

MINDTERM END-USER LICENSE AGREEMENT
(LIMITED COMMERCIAL USE)

PLEASE REVIEW THE FOLLOWING TERMS AND CONDITIONS PRIOR TO ACCESSING, DOWNLOADING
AND/OR OTHERWISE USING ANY OF THE LICENSED PRODUCTS, AS HEREIN AFTER DEFINED.

THE USE OF THE LICENSED PRODUCTS AS WELL AS ANY UPDATES THEREO TO IS SUBJECT TO THE
TERMS AND CONDITIONS OF THE THIS LICENSE AGREEMENT (THE "AGREEMENT"). BY OPENING THE
RELEVANT SOFTWARE PACKAGE, BY SELECTING THE [AGREE AND/OR ACCEPT] BUTTON,
DOWNLOADING AND/OR OTHERWISE USING THE SOFTWARE OR ANY PORTION THEREOF, LICENSEE
(THE FIRM, COMPANY OR OTHER PERSON HAVING RECEIVED THE LICENSED SOFTWARE PURSUANT TO
AN ORDER ON THE APPGATE WEB SITE OR OTHERWISE) ARE AGREEING TO THE BOUND BY THE TERMS
AND CONDITIONS OF THE AGREEMENT AND ARE ENTERING INTO THE AGREEMENT WITH APPGATE
NETWORK SECURITY AB ("LICENSOR" or "APPGATE").

1. DEFINITIONS

As used in this Agreement, the following terms shall have the following meanings:

Yes  No

MindTerm home directory. 'C:\Users\martyn.DCSHT.000\Application Data\MindTerm' does not exist, create it?

Yes  No

Do you want to add this host to your set of known hosts (check fingerprint)
Setting up Putty for AWS instance connection

- Start PuTTYgen (Start menu, click All Programs > PuTTY > PuTTYgen).
- Click on **Load button**
- Find the folder with your *.pem key in.
- Select **All Files *.*** and click on your AWS .pem key.
• A success message should appear, now we need to save the key in PUTTY's own format.

• Click on **Save private key**.

• Confirm you wish to save **without** a passphrase, and save in the same directory.
Connecting to our instance using PuTTY SSH

- Go to Start > All Programs > PuTTY > PuTTY to load up PUTTY SSH.

- Switch back to the AWS console, and copy the address of your instance, it'll look something like 54.171.121.255

- This is the address of the instance that we'll be using to connect to.
Scroll down and click on **Auth**
• Now click on **Browse** and navigate to the key you just saved (ends with '.ppk' extension).

• Now click on **Open**.

• Click on **yes** when the security alert appears.
• Type **ubuntu** as the login name and press **Enter** key

• We don't need a password as our key will be sent across to the instance.
Success! We're now logged in to our **Ubuntu** instance
Installing Java:

$ sudo apt-get update

$ sudo apt-get install openjdk-6-jre

Installing Hadoop:

• Get the file from external site:

  $ wget https://archive.apache.org/dist/hadoop/core/hadoop-0.22.0/hadoop-0.22.0.tar.gz

• Unpack it:

  $ tar xzf hadoop-0.22.0.tar.gz

• Copy it to somewhere more sensible like our local user directory.

  $ sudo cp -r hadoop-* /usr/local

Note: You can copy the below and press SHIFT + Ins to paste in to your terminal window.
• Did you get this error?

  sudo: unable to resolve host ip-172-30-0-12

$ sudo nano /etc/hosts

127.0.0.1 localhost
127.0.1.1 ip-172-30-0-12

The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
#

• Save the file (ctrl-x then type y for yes).
• Edit the terminal script
  $ nano ~/.bash

• Add these lines at the bottom:

  export JAVA_HOME=usr/
  export HADOOP_HOME=usr/local/hadoop-0.22.0

• Save the file (ctrl-x and type 'y')

• Add it to the terminal environment
  $ source ~/.bash

• Now when Hadoop needs Java the terminal will point it in the right direction
Let's move in to the main directory of the application

$ cd /usr/local/hadoop-*

Now edit Hadoop's set up script

$ sudo nano conf/hadoop-env.sh

- Save (ctrl-x, then type 'y')

```
export JAVA_HOME=/usr
```

```#!/usr/bin/env bash

# Set Hadoop-specific environment variables here.

# The only required environment variable is JAVA_HOME. All others are
# optional. When running a distributed configuration it is best to
# set JAVA_HOME in this file, so that it is correctly defined on
# remote nodes.

# The java implementation to use. Required.
export JAVA_HOME=/usr

# Extra Java CLASSPATH elements. Optional.
# export HADOOP_CLASSPATH="<extra_entries>:"$HADOOP_CLASSPATH"

# The maximum amount of heap to use, in MB. Default is 1000.
# export HADOOP_HEAPSIZE=2000

# Extra Java runtime options. Empty by default.
# if [ "$HADOOP_OPTS" == "" ]; then export HADOOP_OPTS="-server"; else HADOOP_OPTS="-server"; fi

# Command specific options appended to HADOOP_OPTS when specified
export HADOOP_NAMENODE_OPTS="-Dcom.sun.management.jmxremote $HADOOP_NAMENODE_OPTS"
export HADOOP_SECONDARYNAMENODE_OPTS="-Dcom.sun.management.jmxremote $HADOOP_SECONDARYNAMENODE_OPTS"
export HADOOP_DATANODE_OPTS="-Dcom.sun.management.jmxremote $HADOOP_DATANODE_OPTS"
export HADOOP_BALANCER_OPTS="-Dcom.sun.management.jmxremote $HADOOP_BALANCER_OPTS"
export HADOOP_JOBTRACKER_OPTS="-Dcom.sun.management.jmxremote $HADOOP_JOBTRACKER_OPTS"
```
• Add the configuration file to the terminals scope:

  
  $ source conf/hadoop-env.sh

• Running an example using Single node mode:

  
  • Calculating PI:

  $ sudo bin/hadoop jar hadoop-mapred-examples-*.jar pi 10 10000000
Another example, using some actual data

- Create a directory to put our data in
  
  $ sudo mkdir input

- Copy the very interesting README.txt file to our new input folder

  $ sudo cp README.txt LICENSE.txt input

- Now we count up the total words and what they are
  (Hadoop will create the output folder for us)

  $ sudo bin/hadoop jar hadoop-mapred-examples-*.jar wordcount input output

- Have a look at the final output

  $ nano output/part-r-00000
Shutting down your instance

- Amazon charges by the hour, so make sure you close your instance after each session.
- Select the instance that is running through EC2 option in the AWS console
- Right-click and select Stop to halt the instance, or Terminate to remove and delete everything.
Hadoop in the AWS Cloud

One last example, this time using AWS to create the Hadoop cluster for us.

First we need a place to put the data after it has been produced...

**Amazon S3** (Simple Storage Service):
An online storage web service providing storage through web services interfaces (REST, SOAP, and BitTorrent)
Setting up the storage

- Select S3 from the console
Welcome to Amazon Simple Storage Service

Amazon S3 is storage for the internet. It is designed to make web-scale computing easier for developers.

Amazon S3 provides a simple web services interface that can be used to store and retrieve any amount of data, at any time, from anywhere on the web. It gives any developer access to the same highly scalable, reliable, secure, fast, inexpensive infrastructure that Amazon uses to run its own global network of web sites. The service aims to maximize benefits of scale and to pass those benefits on to developers.

You can read, write, and delete objects ranging in size from 1 byte to 5 terabytes each. The number of objects you can store is unlimited. Each object is stored in a bucket with a unique key that you assign.

Get started by simply creating a bucket and uploading a test object, for example a photo or .txt file.

Create Bucket

S3 at a glance

Create

Create a bucket in one of several Regions. You can choose a Region to optimize for latency, minimize costs, or address regulatory environments.

Add

Upload objects to your bucket. Amazon S3 durably stores your data in multiple facilities and on multiple devices within each facility.

Manage

Manage your data with Amazon S3’s lifecycle management capabilities, including the ability to automatically archive objects to even lower cost storage options.
Give it a name

(not MyBucket – something unique, also NO CAPITAL LETTERS)

Choose Ireland from the region list

(it's closer, so less latency)
Your new bucket
Running a MapReduce program in AWS

・Select **Elastic MapReduce** in AWS console
Welcome to Amazon Elastic MapReduce

Amazon Elastic MapReduce (Amazon EMR) is a web service that enables businesses, researchers, data analysts, and developers to easily and cost-effectively process vast amounts of data.

You do not appear to have any clusters. Create one now:

How Elastic MapReduce Works

**Upload**
- Upload your data and processing application to S3.
- Learn more

**Create**
- Configure and create your cluster by specifying data inputs, outputs, cluster size, security settings, etc.
- Learn more

**Monitor**
- Monitor the health and progress of your cluster. Retrieve the output in S3.
- Learn more
• Select **Configure sample application**.
• Choose the **Word count** example from the drop down menu.
• Click on the **Output location** folder and select your new **bucket**.

![Configure Sample Application](image)

Change to your bucket name.

s3://<your bucket-name>/logging/

• Click **OK** when done.
Next, specify how many instances you want – just leave it at two for now (the more instances the more £££ it will be to run your job).
Select your keypair

Create a New Job Flow

Here you enter advanced details about your job flow, such as an EC2 key pair, to use VPC, and your job flow debugging options.

Amazon EC2 Key Pair: mykeypair2

Use an existing key pair to SSH into the master node of the Amazon EC2 cluster as the user "hadoop".

Amazon VPC Subnet ID: No preference

To run this job flow in a Virtual Private Cloud (VPC), select a subnet. See Create a VPC.

Configure your logging options. Learn more.

Amazon S3 Log Path: [Your S3 bucket]

Optional: To copy log files from the job flow to Amazon S3, specify an Amazon S3 bucket.

Enable Debugging: Yes

Yes means EMR will store an index of your logs (requires an Amazon S3 Log Path).

Set advanced job flow options.

Keep Alive: Yes

Yes means the job flow will keep running after processing is complete.

Termination Protection: Yes

Yes prevents your nodes from shutting down due to accident or error.

Visible To All IAM Users: Yes

Yes means the job flow will be visible to all IAM users under your account.

Continue
• Scroll to the bottom of the page.

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

<table>
<thead>
<tr>
<th>Bootstrap action type</th>
<th>Name</th>
<th>S3 location</th>
<th>Optional arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add bootstrap action</td>
<td>Select a bootstrap action</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Configure and add</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Action on failure</th>
<th>JAR location</th>
<th>Arguments</th>
</tr>
</thead>
</table>

Add step

Select a step

Configure and add

Auto-terminate

Yes

Automatically terminate cluster after the last step is completed.

Keep cluster running until you terminate it.

No

EC2 key pair has been selected, SSH access will not be enabled for this cluster. Learn how to create an EC2 Key Pair.

Cancel Create cluster
Setting up your own job (for coursework)

This is the place to configure your Hadoop job by uploading your code and data to your S3 bucket.
Input data:
eu-west-1.elasticmapreduce/samples/wordcount/input

Output data:
This is going to be stored on our S3 bucket...
s3n://lazyeels/wordcount/output/2013-11-01

Todays date
• Click on **Create cluster**.

**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

<table>
<thead>
<tr>
<th>Bootstrap action type</th>
<th>Name</th>
<th>S3 location</th>
<th>Optional arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add bootstrap action</td>
<td>Select a bootstrap action</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**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

<table>
<thead>
<tr>
<th>Name</th>
<th>Action on failure</th>
<th>JAR location</th>
<th>Arguments</th>
</tr>
</thead>
</table>

Add step

<table>
<thead>
<tr>
<th>Auto-terminate</th>
<th>Yes</th>
<th>Automatically terminate cluster after the last step is completed. Keep cluster running until you terminate it.</th>
</tr>
</thead>
</table>

*No EC2 key pair has been selected, SSH access will not be enabled for this cluster. Learn how to create an EC2 Key Pair*
Your MapReduce job is now running.
• Go to your **S3** bucket via the **AWS** console.

• The results have been written to the output folder in parts in HDFS format.
You can delete the results by right-clicking on the folder and selecting delete.

Amazon charges for storage so this is worth doing if you no longer need it.

In addition, Hadoop will fail if it finds a folder with the same name when it writes the output.

**Note:** The S3 bucket is where you would upload your .jar or .py files representing your code, as well as any data. It is worth creating a separate folder for each of your runs.

Click on the upload button to upload them from your local machine.
Some tips:

**Hadoop** is not designed to run on Windows. Consider using **Cygwin** or **Virtualbox** ([https://www.virtualbox.org](https://www.virtualbox.org)), or installing **Linux Mint** ([http://www.linuxmint.com/](http://www.linuxmint.com/)) alongside your Windows install (at home).

Stick to earlier versions of Hadoop such as **0.22.0** (they keep moving things around, especially the class files that you'll need to compile your code to **.jar**)

Most books and tutorials are based on earlier versions of Hadoop.

**Single-node** mode is fine for testing your map-reduce code before deploying it.

There are example programs in the folder at:

**Hadoop-0.22.0/mapreduce/src/examples/org/apachehadoop/examples/**
Get in the habit of stopping your instances when you're finished!

**Hadoop in Action** is your friend! Consider getting a copy:

**Chapter 2**
Shows you how to set everything up from scratch.

**Chapter 3**
Provides some good templates to base your code on.

**Chapter 4**
 Discusses issues you may encounter with the different API versions

**Chapter 9**
Tells you how to launch your MapReduce programs from the command line and AWS console, as well as using S3 buckets for data storage and how to access it.
Some useful links

Installing and usage:

Running a job using the AWS Jobflow (Elastic Map Reduce):
http://cloud.dzone.com/articles/how-run-elastic-mapreduce-job

Theory:

Accessing AWS and Hadoop through the terminal (for Linux users):
http://rodrigodsousa.blogspot.co.uk/2012/03/hadoop-amazon-ec2-updated-tutorial.html