

# GraphLab Create CommonCrawl Benchmark Instructions

## Part 1: Creating an AWS EC2 Instance

### Sign In to the AWS Console

Sign In to the AWS console: <https://console.aws.amazon.com/console/home>

You will get to the AWS Sign In screen:



---

Account:

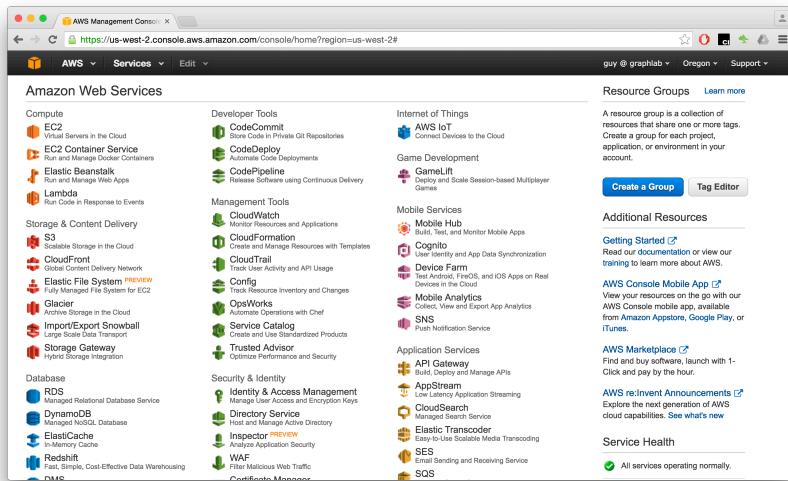
User Name:

Password:

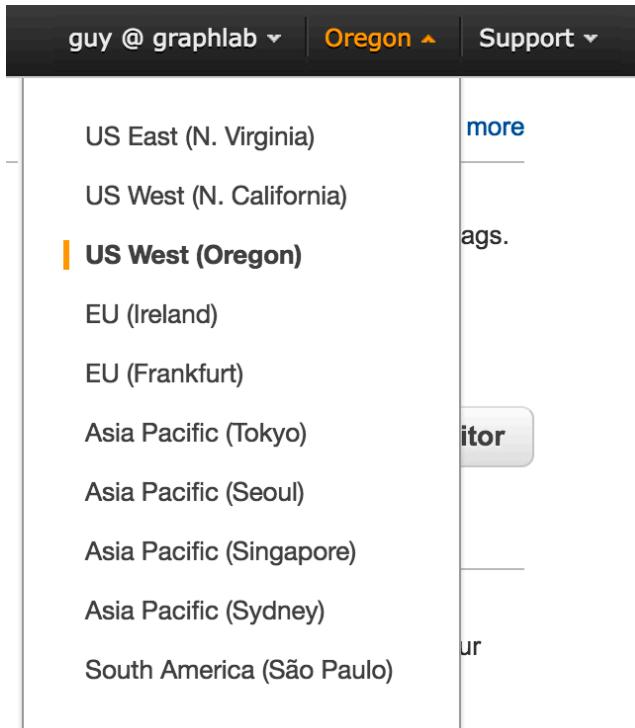
MFA users, enter your code on the next screen.

[Sign-in using root account credentials](#)

Enter your credentials and Sign In. The next screen shows the big list of AWS services:



Note the top-right corner: between your account (*guy @ graphlab* in this example) and the **Support** dropdown, you should see your AWS region. In this example, the region is **Oregon**. If your region is different, click on the regions dropdown and select **US West (Oregon)**. This is the region where we stored the data for this benchmark. If you'll use the same region as we did, then pulling the data from S3 will be much, much faster.



Now that your region is properly set, click on the EC2 icon (first item in first column, top-left icon.)

# Amazon Web Services

## Compute



**EC2**

Virtual Servers in the Cloud

You will be presented with the EC2 Management Console.

The screenshot shows the AWS EC2 Management Console interface. The left sidebar has categories like EC2 Dashboard, Instances, Images, Elastic Block Store, Network & Security, and more. The main content area displays EC2 resources: 30 Running Instances, 5 Elastic IPs, 0 Dedicated Hosts, 241 Snapshots, 52 Volumes, 33 Load Balancers, 58 Key Pairs, 202 Security Groups, and 5 Placement Groups. A callout box highlights the 'Launch Instance' button under the 'Create Instance' section. The right sidebar includes sections for Account Attributes (Supported Platforms, VPC, Default VPC), Resource ID length management, Additional Information (Getting Started Guide, Documentation, All EC2 Resources, Forums, Pricing, Contact Us), and AWS Marketplace (Tableau Server, Rating).

## Create the Instance

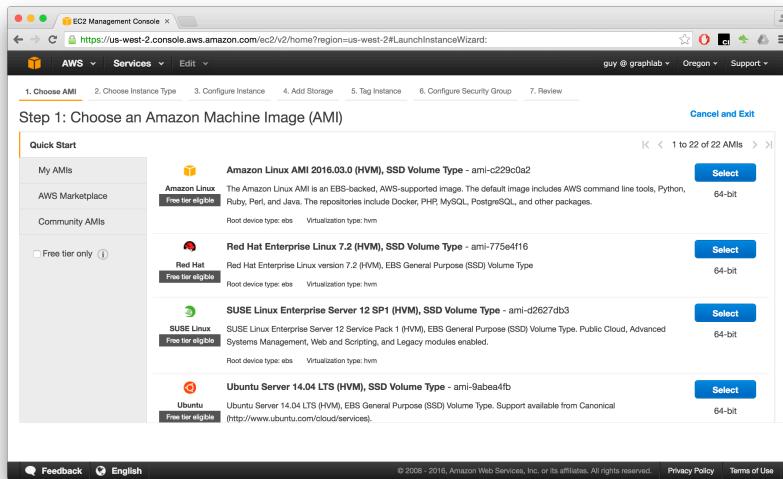
On the left sidebar, under the **INSTANCES** category, choose the **Instances** option.

**Launch Instance**

Click on the **Launch Instnace** button.

### Step 1: Choose AMI

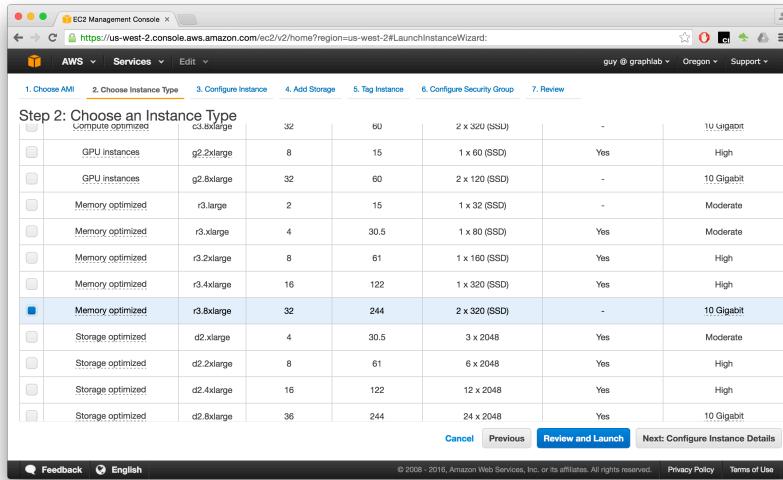
You will now follow a set of steps necessary for launching an instance. This is the screen of **Step 1: Choose an Amazon Machine Image (AMI)**.



Scroll the list of AMIs down to **Ubuntu Server** (the current version is **Ubuntu Server 14.04 LTS (HVM), SSD Volume Type - ami-9abea4fb**). Click on the **Select** button.

## Step 2: Choose Instance Type

Scroll down the list of instance types and choose **r3.8xlarge**. That's a strong machine, with 32 cores, 244 Gigabytes of RAM, and 2 SSD drives, each sized 320 GBs.



After you chose the type, in the breadcrumb list of steps, skip to **4. Add Storage**.

## Step 4: Add Storage (yes, we skipped Step 3!)

**Step 4: Add Storage**  
Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Delete on Termination	Encrypted
Root	/dev/sda1	snap-306df873	16	General Purpose SSD (GP2)	24 / 3000	<input checked="" type="checkbox"/>	Not Encrypted

Add New Volumes

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

## Step 5: Tag Instance

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Configure Security Group 7. Review

### Step 5: Tag Instance

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. [Learn more](#) about tagging your Amazon EC2 resources.

Key (127 characters maximum)	Value (255 characters maximum)
name	gic_memonicrawl_benchmark

Create Tag (Up to 10 tags maximum)

## Step 6: Configure Security Group

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Configure Security Group 7. Review

### Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group:  Create a new security group  Select an existing security group

Security group name: launch-wizard-134  
Description: launch-wizard-134 created 2016-03-31T12:08:00.183+03:00

Type	Protocol	Port Range	Source
SSH	TCP	22	Anywhere 0.0.0.0/0
Custom TCP Rule	TCP	8888	Anywhere 0.0.0.0/0

Add Rule

**Warning**  
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Source

Anywhere 0.0.0.0/0

My IP 79.183.60.180/32

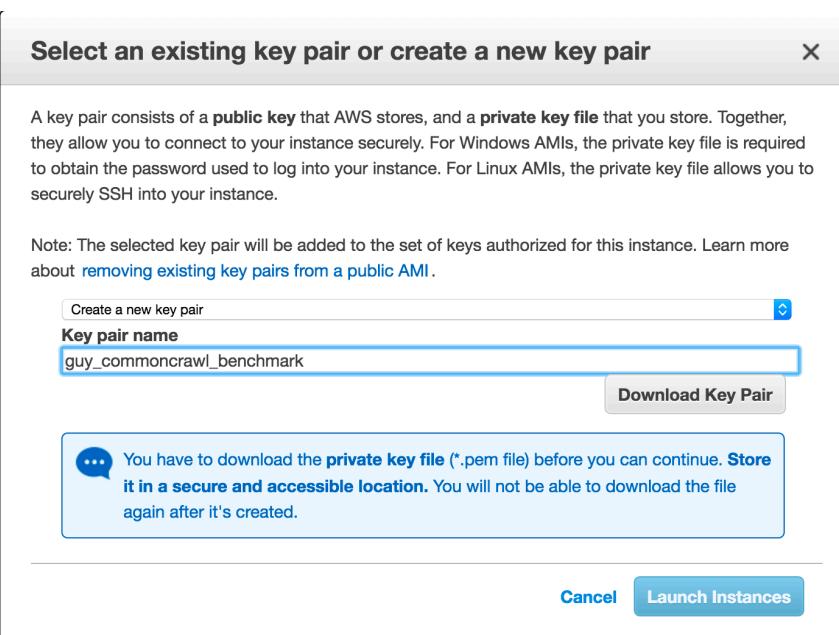
**Review and Launch**

Button

## Launch Your Instance!

You can review your instance's configuration. When finished, click on the **Launch** button to finally launch your instance!

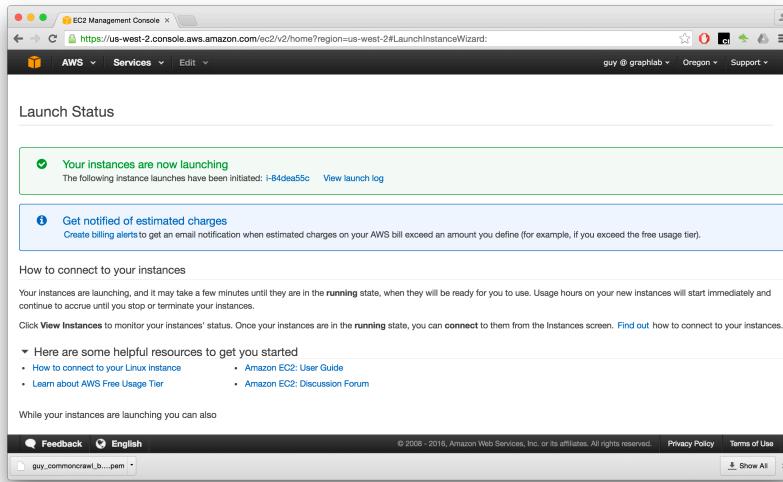
**Launch**



But how will you connect to your instance? Upon Launch, AWS immediately asks you to provide an SSH key pair. If you don't have such a key pair, you can choose the **Create a new key pair** option. Give it a **Key pair name**, then click on the **Download Key Pair** button.

After the key pair we downloaded, click on the **Launch Instances** button.

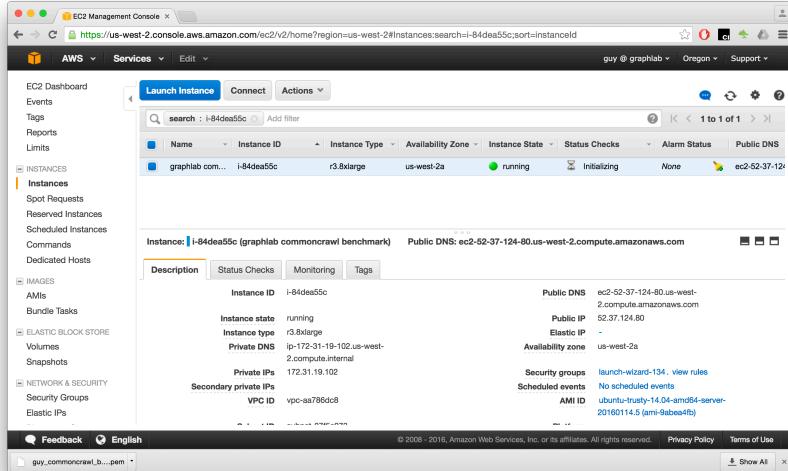
You will get the following status screen:



In my case, it says: The following instance launches have been initiated: i-84dea55c Click on the instance name (**i-84dea55c** in this example) to get to your instance's

status screen.

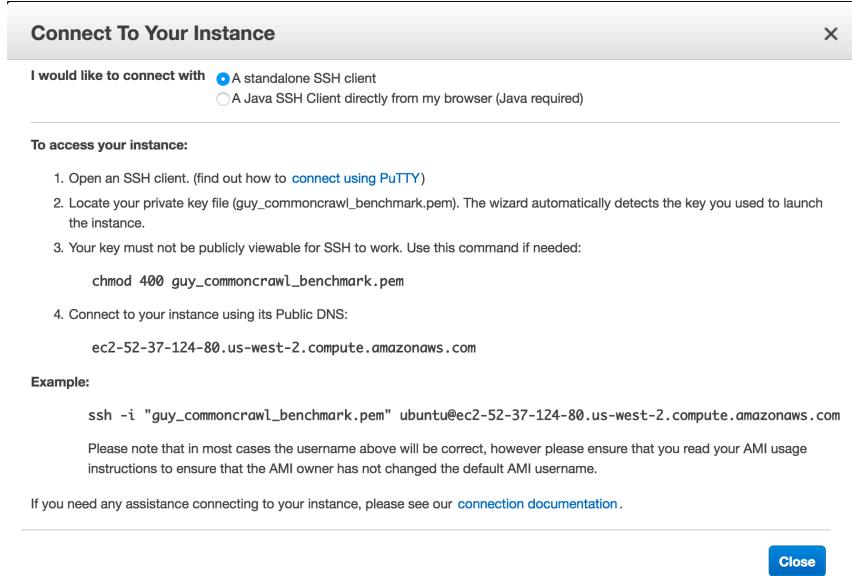
From OS X or Linux, simply run the following command in your terminal:



Click on the **Connect** button.

**Connect**

You will see AWS' instructions for connecting to your instance.



I would like to connect with  A standalone SSH client  
 A Java SSH Client directly from my browser (Java required)

To access your instance:

1. Open an SSH client. (find out how to [connect using PuTTY](#))
2. Locate your private key file (`guy_commoncrawl_benchmark.pem`). The wizard automatically detects the key you used to launch the instance.
3. Your key must not be publicly viewable for SSH to work. Use this command if needed:  
`chmod 400 guy_commoncrawl_benchmark.pem`
4. Connect to your instance using its Public DNS:  
`ec2-52-37-124-80.us-west-2.compute.amazonaws.com`

Example:

```
ssh -i "guy_commoncrawl_benchmark.pem" ubuntu@ec2-52-37-124-80.us-west-2.compute.amazonaws.com
```

Please note that in most cases the username above will be correct, however please ensure that you read your AMI usage instructions to ensure that the AMI owner has not changed the default AMI username.

If you need any assistance connecting to your instance, please see our [connection documentation](#).

**Close**

Follow the instructions, and note the public DNS assigned to your instance (in the example above, it is `ec2-52-37-124-80.us-west-2.compute.amazonaws.com`).

If you are in OS X or Linux, you can connect to your instance from the shell:

```
ssh -i "/PATH/TO/DOWNLOADED/KEY_FILE.pem" ubuntu@<your-ec2-public-dns>.compute.amazonaws.com  
# In my case, the command is:  
ssh -i "/Users/dato/Downloads/guy_commoncrawl_benchmark.pem" ubuntu@ec2-52-37-124-80.us-west-2.compute.amazonaws.com
```

You can also follow AWS' online instructions at:

<https://docs.aws.amazon.com/console/ec2/instances/connect/docs>

They also have [instructions written specifically for PuTTY](#), a famous Windows SSH client.

## Part 2: Installing Requirements on your Instance

Now that you are logged into your instance via ssh, run each of the following commands.

Note: pasting all the commands will only run the first one. Please run each command separately.

The instructions for securing the notebook server are taken from [Jupyter's website](#).

```

# Install Python, VirtualEnv
sudo apt-get update
sudo apt-get install -y build-essential python-setuptools zlib1g-dev
sudo easy_install pip
sudo pip install virtualenv

# Create a VirtualEnv for GraphLab Create
virtualenv graphlab_venv
cd graphlab_venv
source bin/activate
pip install graphlab-create
cd ~

# Install Jupyter (IPython-Notebook)
sudo apt-get install -y python-dev
pip install jupyter

# Password protect Jupyter
jupyter notebook --generate-config
python -c "from notebook.auth import passwd; password = passwd(); open('/home/ubuntu/.jupyter/jupyter_notebook_config.py', 'a').write('c.NotebookApp.password = u'%s'' % (password))"

# Download the Benchmark's IPython Notebook
wget <public address of the notebook>

# Run the notebook
nohup jupyter notebook --no-browser --ip="*" & > pid

```

The Jupyter server is now running. You should browse to <http://<your instance address>:8000>. This is what you're supposed to see in your browser:

 [jupyter](#)

Password:	<input type="text"/>	<input type="button" value="Log in"/>
-----------	----------------------	---------------------------------------

Use the password you entered in order to log into the notebook server.

The rest of the benchmark can be executed via your browser.