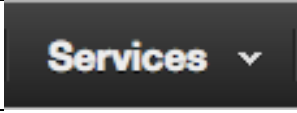
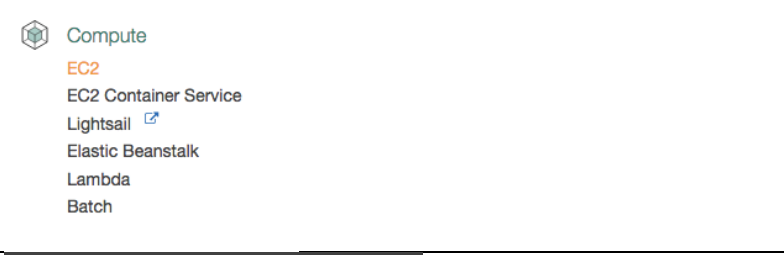
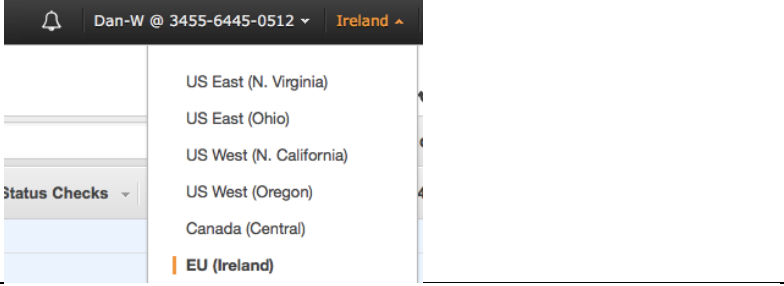
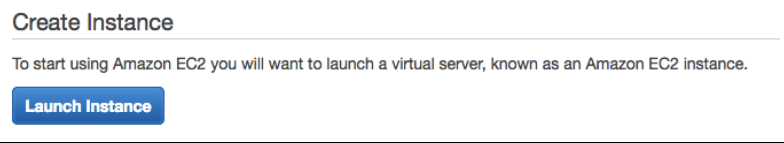
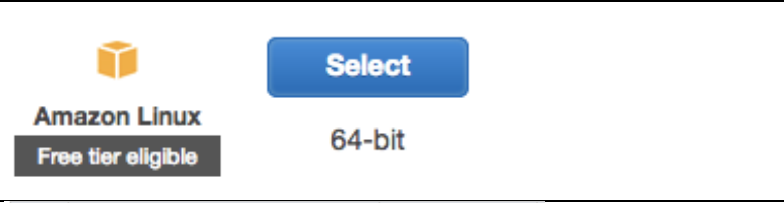
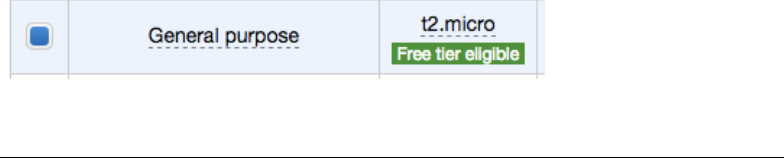
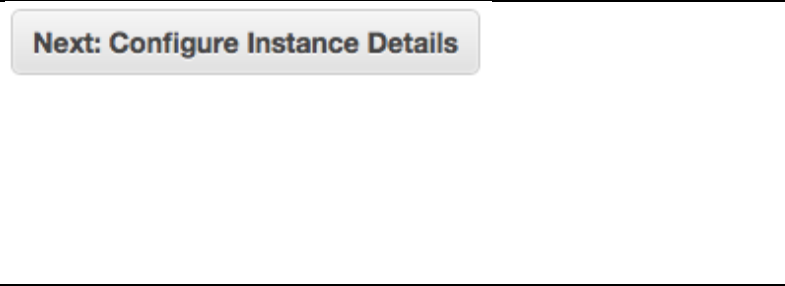
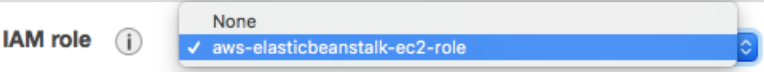



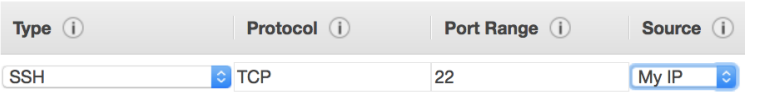


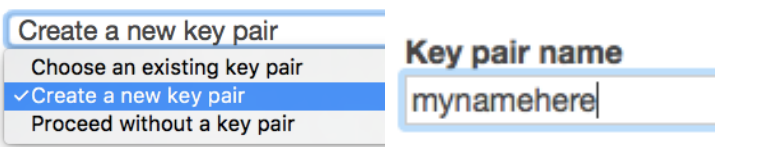

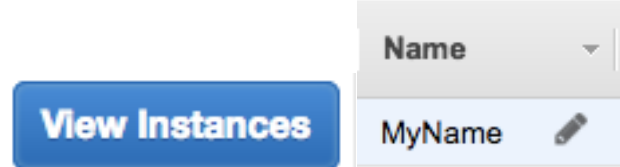


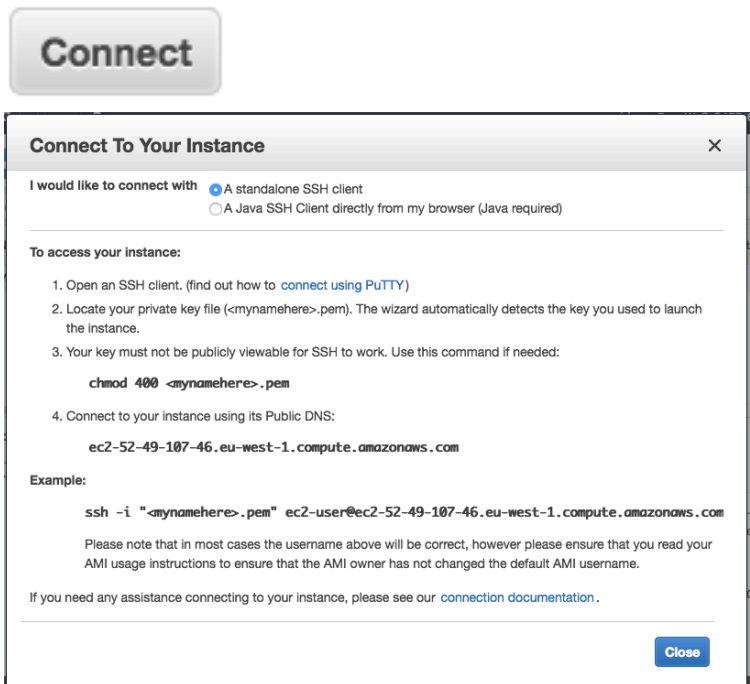
Training Materials

An Amazon Web Services account is required for this workshop. Students can register for an account with \$40 of free credit at <https://aws.amazon.com/education/awseducate/>.

Launching a Virtual Server using EC2

<p>Navigate to the Amazon Management Console</p>	<p>https://aws.amazon.com/</p>
<p>Go to the Services Drop-Down</p>	
<p>Select EC2</p>	
<p>Make sure you are in the EU-West region (Ireland) using the drop-down in the top right corner of the screen.</p>	
<p>Under Create Instance select Launch Instance</p>	
<p>Create Amazon Linux AMI (Free Tier Eligible)</p>	
<p>Select General Purpose t2.micro</p>	
<p>Configure instance details</p>	

<p>Select the aws-elasticbeanstalk-ec2-role.</p> <p>Roles encapsulate permissions (the ability to access, view or modify parameters) within AWS. They can be applied to EC2 instances, AWS users or groups of users.</p>	
<p>Add storage – defaults are fine.</p>	
<p>Add Tags Add a name tag with your name, for identification of your instance.</p>	 <p>Choose the Add tag button or click to add a Name tag.</p>
<p>Move on to security groups</p>	
<p>Edit Security groups - Allow SSH, Source: My IP</p>	
<p>Review and Launch</p>	
<p>Launch</p>	
<p>Create a new Key Pair, give it your name, press Download and save to your machine (mynamehere.pem)</p>	
<p>Save this keypair somewhere easy to get to. We suggest a new folder in your home directory called "Key"</p>	<p>Mac/Linux</p> <pre>\$ mkdir Key \$ mv Downloads/mynamehere.pem Key/mynamehere.pem</pre>
<p>Launch your instance</p>	
<p>Click View Instances.</p> <p>Lots of instances will be launching at the same time – use the name tags to find yours.</p>	

click Connect and a window should open	
--	--

Instructions for Mac / Linux

open terminal navigate to your key folder	<code>\$ cd Key</code>
make your private key private by modifying its permissions.	<code>\$ chmod 400 Name.pem</code>
to avoid having to type the long name over and over, copy it from the connection window set a variable in bash.	<p>Example:</p> <pre>ssh -i "mynamehere.pem" ec2-user@ec2-52-209-161-135.eu-west-1.compute.amazonaws.com</pre> <pre>\$ instance=ec2-user@ec2-ip-ip- ip.ip.location.compute.amazonaws.com</pre> <p>refer to this variable later using <code>\$instance</code></p>
Connect to your instance and type yes to allow the connection.	<pre>\$ ssh -i "Name.pem" \$instance Are you sure you want to continue connecting (yes/no)? yes</pre>
You should now be on the machine.	<code>[ec2-user@ip-ip-ip-ip ~]\$</code>
Terminal should show:	

27th April 2017

Transferring files from an Amazon S3 Bucket to your instance

<p>Create a text file in your home directory and use a new terminal window to transfer it to your instance using scp (Mac)</p>	<pre>\$ scp -i Key/myname.pem newFile.txt \$instance:~</pre>
<p>As an alternative, we can transfer files from an Amazon S3 bucket to the instance using Amazon's s3 copy command.</p> <p>Our bucket is called "ngcm1"</p>	<pre>[ec2-user@ip-ip-ip-ip ~]\$ aws s3 ls s3://ngcm1 [ec2-user@ip-ip-ip-ip ~]\$ aws s3 cp s3://ngcm1/testFile.txt testFile.txt</pre>
<p>in your terminal window for the instance, check that it's been transferred.</p>	<pre>[ec2-user@ip-ip-ip-ip ~]\$ ls</pre>
<p>Code can be run on the instance in the same way</p> <p>Transfer "sim.py" to the instance from the S3 bucket and run it</p>	<pre>[ec2-user@ip-ip-ip-ip ~]\$ python sim.py</pre>
<p>Use scp to transfer the output of the simulation out.csv onto your local machine.</p>	<pre>\$ scp -i Key/myname.pem \$instance:~/out.csv out.csv</pre>

Windows Instructions (with PuTTY)

Use the following instructions from Amazon to convert your .pem key file into a .ppk file, then connect to your running instance over SSH. Move the .ppk file to the new directory called “Key” you created earlier.

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/putty.html?icmpid=docs_ec2_console

Transferring files from an Amazon S3 Bucket to your instance

<p>Use notepad to create an example text file. From the command prompt, transfer it to your instance using <code>pscp</code> (installed with PuTTY)</p>	<pre>C:\Users\Name\Key> pscp -i name.ppk newFile.txt ec2-user@ec2.....amazonaws.com:newFile.txt</pre>
<p>As an alternative, we can transfer files from an Amazon S3 bucket to the instance using Amazon’s <code>s3</code> copy command.</p> <p>Our bucket is called “ngcm1”</p>	<p>From the instance:</p> <pre>[ec2-user@ip-ip-ip-ip ~]\$ aws s3 ls s3://ngcm1</pre> <pre>[ec2-user@ip-ip-ip-ip ~]\$ aws s3 cp s3://ngcm1/testFile.txt testFile.txt</pre>
<p>in your terminal window for the instance, check that it's been transferred.</p>	<pre>[ec2-user@ip-ip-ip-ip ~]\$ ls</pre>
<p>Code can be run on the instance in the same way</p> <p>Transfer “sim.py” to the instance from the S3 bucket and run it</p>	<pre>[ec2-user@ip-ip-ip-ip ~]\$ python sim.py</pre>
<p>Use <code>pscp</code> to transfer the output of the simulation <code>out.csv</code> onto your local machine using the command prompt.</p>	<p>Windows (PuTTY)</p> <pre>> pscp -i <path to Key> ec2-user@ec2.....amazonaws.com:out.csv out.csv</pre>

27th April 2017

A very simple web hosting example with EC2...

We are going to repurpose our running instances to host a static webpage. The free tier allows 24/7 running of a single t2.micro amazon instance for 12 months (750 hours / month free).

Step by step guidance is not provided for this section. We are here to help, the detailed instructions above should give you a good start, and AWS has many rollover information points and comprehensive documentation.

- In the management console, set up your instance to allow access from internet traffic over port 8000.
- Transfer the entire contents of the Web subdirectory on the ngcm1 S3 bucket onto the instance. (hint: aws s3 help)
- navigate to this folder on the instance and launch a web server
`$ python -m SimpleHTTPServer 8000 &`
- In a web browser, navigate to the instance's public URL, and specify port 8000 to view your website.

