## Steps based off screenshots (in /snaps folder)

### The words in brackets indicate the snapshot the step is referring to

### Requires the PUTTY application package

# Amazon “Free Tier” Account Creation

1. Create a new Amazon account with a non-associated Gmail address.
2. Sign up for Amazon Web Services.

# Amazon EC2 Instance Creation

1. Click Launch Instance on the top-left (EC2-1).
2. For the purposes of this project, Amazon Linux 64-bit should be selected (indicated as part of the Free-Tier program) (EC2-2).
3. The 1GB t2 micro memory instance type should be selected (EC2-3).
4. Leave Instance Details as default, as any changes here could bring about complications or uncertainty in the future when dealing with network details and such (EC2-4).
5. Change the root storage disk size to 30 gb. This should be more than enough space for the simple needs we require, and is of course also eligible for the Free Tier (EC2-5).
6. At this point you are able to name or “Tag” your instance. There are several other tags available to set here though the only one to be concerned with for our purposes is the Name tag. I have called my instance “vlinux” but any other simple name will also be appropriate (EC2-6).

# SSH KeyPair Generation

1. Choose the “Select an existing Security Group” radio button, and then select the only group there (named default). (EC2-7).
2. Proceed to launch the instance and create a new key pair, at the same time assigning it a name. Don’t forget to click Download Key Pair here! You can’t download it anywhere else or call up this dialog box again. (EC2-8& EC2-9).

# Setting up Putty and Encrypted PPK File

1. Open PuttyGen, load in the private key pem file you just downloaded before. Make sure the file types to look for dropdown is set to All Files. If you have unfortunately forgotten to download this pem file from step 9, you will have to re-create the instance.. (EC2-14).
2. Decide whether you want to use a passphrase or not. For a simple project like this one, I did not think it was necessary. (EC2-15).
3. Generate a ppk PUTTY Private Key file from the pem file, and you will be prompted to choose a location to save this key to, I suggest saving it to the same place as the pem file.

# Stopping and Restarting the EC2 Instance

1. Download and install the AWS Console Mobile app, I have found this is more reliable than desktop in terms of managing the instance state.
2. Log into your AWS account.
3. Enter the EC2 Instances sub-menu.
4. Select the instance(s) you want to stop.
5. Tap the Stop button.
6. Similarly, you can tap the Start button if the instance was stopped, in effect having restarted the instance.

# Connecting to the Instance

1. On the instance list on the EC2 page, select your instance.
2. On the description tag below, copy the public IP number.
3. Open Putty and feel free to save this as a new session; right-click the tray icon and click new session.
4. Paste the ip into the Host Name field, click on Data under the Connection category on the left column and enter ec2-user into the Auto-login username.
5. Click back on the session tab and click save but do not open yet.
6. Right click on the tray icon again and click add key, and select the ppk file you downloaded earlier.
7. Right click on the tray icon once more and click Saved Sessions > (your sessions name). The connection should be established.

# Creating / Removing EBS Volumes

1. Launch this instance, and then click Volumes under the ELASTIC BLOCK STORE category on the left column. Click Create Volume on the top-left corner of this window. (EC2-10).
2. For this volume, choose the same availability zone as the current root drive is allocated to. (EC2-11).
3. Right click this new volume, and click Attach Volume. (EC2-12).
4. Click on the instance field, it should show your instance id as the only item in the list that pops up. Note the Device field value here, this is important. (EC2-13).

# Managing EBS Volumes

1. You will first need to make a new file system with the location of the drive (sdf in my case, also displayed in the details of the drive when clicked on in the AWS window).  
     
   In the terminal, type “mkfs /dev/sdf” where sdf is the device name field from step 12.

Make a new directory which maps to this drive, I called mine “bigdata”.

* mkdir /mnt/bigdata
* mount /dev/sdf /mnt/bigdata

Check disk space using the “df” command.

Verify that the 100gb extra EBS volume has been mounted on the directory you specified above. You will notice this in the same output from the “df” command, as it also displays the size of each drive as well as the directory they are mounted on.(EC2-18).

1. To confirm the drive can be written to, a simple test can be performed. You can change the working directory to the one which the drive is mounted on, list its contents, write the current date to a new file called “x”, output the contents of this file and then list the contents of the folder. The screenshot outlines how to do this. (EC2-19).

# Installing and Configuring Postgres, setting up DB Owner

1. Instructions at <https://www.unixmen.com/postgresql-9-4-released-install-centos-7/> should be followed, and the instance’s “Amazon Linux” is roughly equivalent to the CentOS 6.x 64bit Operating System, hence these sections can be followed when required.
2. Set access permissions (Ask dad about this and setting up the DB Owner)

# Installing Apache and PHP, and Managing the WebServer

1. Ensure PostGres is installed.
2. Install additional packages IN THIS ORDER (Apache service is named Httpd);

Httpd

Php

Php-pgsql

phpPgAdmin

1. Once installed, the httpd and phpPgAdmin services must be started in order to use the management console. Type “service httpd start” to start it. You can verify it is running through observing the output of “service httpd status”.
2. The same commands can be replaced with the phpPgAdmin service. When satisfied, head to “(instance ip)/phpPgAdmin”. You should be presented with the management console.
3. Click on the left “PostgreSQL” and log in with the DB user you set up previously.
4. Click Create database, then expand its Schemas > public > Tables categories. Create the tables here that will be used in the Web App later on.

# Backup/Restore Test Data

1. Click Export on the top-right of the phpPgAdmin management console.
2. Select Structure and data, Format : SQL and the Download option.
3. Click Export, and save the dump file somewhere safe.
4. To restore, simply click Import and locate and open the dump file you saved.
5. Not only will test data be restored, but the whole database including tables, their structure, foreign keys and so on.

# Backup/Restore Test Data

1. Click Export on the top-right of the phpPgAdmin management console.
2. Select Structure and data, Format : SQL and the Download option.
3. Click Export, and save the dump file somewhere safe.
4. To restore, simply click Import and locate and open the dump file you saved.
5. Not only will test data be restored, but the whole database including tables, their structure, foreign keys and so on.

# Writing the Web App

To be continued..