Appendix A : HOWTOs

Contents

[Amazon “Free Tier” Account Creation 2](#_Toc480316746)

[AWS Console for Mobile 3](#_Toc480316747)

[TortoiseGit for Windows 4](#_Toc480316748)

[OctoDroid 5](#_Toc480316749)

[SSH KeyPair Generation 6](#_Toc480316750)

[Amazon EC2 Instance Creation 7](#_Toc480316751)

[Setting up Putty and Encrypted PPK File 8](#_Toc480316752)

[Stopping and Restarting the EC2 Instance 9](#_Toc480316753)

[Connecting to the Instance 10](#_Toc480316754)

[Creating / Removing EBS Volumes 11](#_Toc480316755)

[Managing EBS Volumes 12](#_Toc480316756)

[Installing and Configuring Postgres, setting up DB Owner 13](#_Toc480316757)

[Installing Apache and PHP, and Managing the WebServer 14](#_Toc480316758)

[Backup/Restore Test Data 15](#_Toc480316759)

[Writing the Web App 16](#_Toc480316760)

# Amazon “Free Tier” Account Creation

**NOTE: You may already have an Amazon “shopping” account, used for purchasing books, DVDs etc. It’s possible to let this account to be the basis for your usage of Amazon Web Services, bearing in mind that any billing for the project will be sent to your personal account. Billing refers to anything beyond Amazon’s “free tier”. If you prefer billing on the project to be kept separate (e.g. Academic / corporate requirements) then best practice is to create a new Amazon account first.**

1. Create a new Amazon account if necessary
2. Tie the Amazon account to the Amazon Web Services Free Tier, by clicking “Create a Free Account” here: <https://aws.amazon.com/start-now/>
3. After following the standard prompts, you will be notified that services will be available soon.
4. Once services are available, make sure you are signed into the Web Console from the “Sign In to the Console” link here: <https://aws.amazon.com/console/>
5. Try to use some services, the easiest test is the “S3” service to share very large files privately or publicly

# AWS Console for Mobile

1. Download the AWS Console Mobile Application which can be found on the Google Play store: <https://play.google.com/store/apps/details?id=com.amazon.aws.console.mobile> (aws-1)
2. Open on your mobile and sign in as usual
3. You should see a similar screen to (aws-2), from which you should click the “EC2 Instances” row
4. From here, you can manage your instance/s through clicking its row, and the appropriate option. Or simply to review its status (aws-3)

# TortoiseGit for Windows

1. Download the application from https://tortoisegit.org/download/ (tor-1) and make sure you choose the right architecture as highlighted in the screenshot
2. Installation should go smoothly, go with all default installation options
3. If you try to run the application this dialog will show (tor-2) making clear that its usage is as a context menu
4. Create a new folder on a drive and then Right click > “Git Create repository here”
5. Ignore the checkbox and click Ok
6. This folder is now your local Git Repository, and from another Right click more options should be available under the TortoiseGit context menu (tor-3)
7. You can optionally Right click > TortoiseGit > Settings and fill in your User Name and Email to save time against filling it in manually each time (tor-4)
8. When you have worked off this folder and are happy with your changes, you can push it to GitHub. Firstly however the changes must be committed, though these two steps can be done at once.
9. Simply Right click > Git Commit -> “master”
10. Type an input message, optionally check to include author date and/or author, check “All” for changes made (I found this was preferable) and then click “Commit & Push”. (tor-5)
11. This new Git repo can be viewed on GitHub.com, after signing in on there also

# OctoDroid

1. Download the OctoDroid for GitHub Mobile Application which can be found on the Google Play store: https://play.google.com/store/apps/details?id=com.gh4a&hl=en\_GB (oct-1)
2. Open on your mobile and sign in as usual
3. Select your repository for viewing / modification
4. From here, you can manage your instance/s through clicking its row, and the appropriate option. Or simply to review its status (oct-2)
5. The commit trail here is very useful as an example to quickly check if your latest commit has gone through (oct-3)

# SSH KeyPair Generation

1. Log into the Amazon Management Console <https://aws.amazon.com/console>
2. Choose EC2 from the high-level list of services under the “Compute” Category (key-1)
3. Click Key Pairs on the left-hand menu
4. Click “Create Key Pair”
5. Choose a simple new Key Pair name, typically named after yourself (key-2)
6. It will automatically download this as a .pem file, used in the later steps

# Amazon EC2 Instance Creation

1. Click Launch Instance on the top-left of the Amazon Management Console (EC2).
2. For the purposes of this project, Amazon Linux 64-bit should be selected (indicated as part of the Free-Tier program) (EC2-1).
3. The 1GB t2 micro memory instance type should be selected (EC2-2).
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 (EC2-3).
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 (EC2-4).
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 (EC2-5).
7. Choose the “Select an existing Security Group” radio button, and then select the only group there (named default). (EC2-6).
8. 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. (EC2-7 & EC2-8).

# Setting up Putty and Encrypted PPK File

1. Download and Install PuTTY from this site: <http://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html> (put-1) using the full msi Windows installer package for your current system architecture
2. Open PuTTYGen and Load in your .pem file downloaded during “SSH KeyPair Generation” by clicking “Load” highlighted in (put-2)
3. Invent a memorable passphrase and enter it, and again to confirm it and finally generate a .ppk PUTTY Private Key file from the pem file by clicking “Generate” (put-3)
4. Click Save private key 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.
5. Add PPK file to the Windows Startup folder by simply dragging it in, the easiest way to access this folder is to paste “%appdata%\Microsoft\Windows\Start Menu\Programs\Startup” into the address bar of Windows Explorer (put-4)
6. Double click this file to test (also executed on startup), enter passphrase
7. PuTTY should now be authenticated with this file, and you can proceed to connect to any EC2 instance which was activated using the SSH keypair corresponding to the .pem file

# Stopping and Restarting the EC2 Instance

1. Optionally download and install the AWS Console Mobile app, I have found this is more responsive than desktop in terms of managing the instance state. Link to app on Google Play Store: <https://play.google.com/store/apps/details?id=com.amazon.aws.console.mobile&hl=en_GB>
2. Log into your AWS account either on Mobile or Desktop into Management Console. Desktop url: https://aws.amazon.com/console
3. Enter the EC2 Instances sub-menu.
4. Select the instance(s) you want to stop.
5. Tap the Stop button, the instance will now perform a Shut Down
6. Similarly, you can tap the Start button if the instance was stopped, in effect having restarted the instance.
7. At any point, call up Instance Settings > Get System Log to verify the state of the instance console

# 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 (con-1).
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. (EC2-9).
2. For this volume, choose the same availability zone as the current root drive is allocated to. (EC2-10).
3. Right click this new volume, and click Attach Volume. (EC2-11).
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. (EC2-12).

# 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.(EC2-16).

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. (EC2-17).

# Installing and Configuring Postgres, setting up DB Owner

1. yum install postgres
2. Set access permissions (Ask dad about this and setting up the DB Owner)

# Installing Apache and PHP, and Managing the WebServer

1. Ensure PostgreSQL is installed, type “yum list postgresql\*” to list available packages, version 9.4 seems safe as 9.5 had some bugs. Type “yum install postgresql94-server” to continue with package installation and follow the prompts (pak-1).
2. Upon trying to start the service (“service postgresql94 start”), the cluster must be initialized as stated. Type the required command (pak-2)
3. Try to start the service again and it should start
4. Install additional packages IN THIS ORDER (Apache service is named Httpd);

Yum install Httpd

Yum install Php

Yum install Php-pgsql

Yum install 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 (type “service phpPgAdmin start”. 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, DROP THE DATABASE FIRST (bak-1) and re-create, select the database from the tree controller on the left, click the SQL tab at the top, then “Choose file” and then locate and open the dump file you saved. (bak-2)
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..