IFB299 – Release 1
Contribution Report
James Tuerlings n8351473

In order to display the team's website online, a remote server was required. For this project we agreed that the service should be be available to our users at all times and that the service used to host the server should be free of charge. To this end it was decided that Amazon's EC2 (Elastic Compute Cloud) service was appropriate for our needs [1]. EC2 is capable of 'launching' virtual server instances that run system images of its user's choosing. For our purposes, an Ubuntu virtual server was created using the maximum available memory (30 GB), this was done through Amazon's server creation wizard. Ubuntu was chosen over other operating systems due to its ease of use and extensive documentation throughout the web. A minimalistic security scheme was chosen for the server as its contents do not need protection from outside access due to its nature as a prototype. For a full implementation, more advanced security measures could be applied. Protection against termination was added such that the server could not terminate unexpectedly. Once installation was completed, the server instance was added to the instance list as seen in Figure 1.

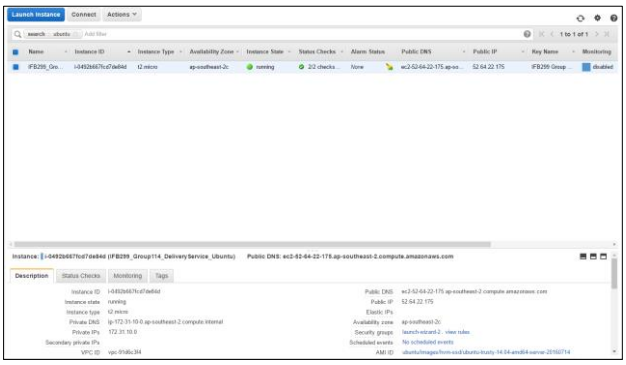

Figure 1: The Group's Server

In order to properly connect to the server, the SSH/Telnet program 'PuTTY' was used [2]. By using the IP address of the remote server, in addition to a security key-pair generated on the EC2 site, the program 'PuTTYgen' [3] was able to generate a secure authentication key to allow 'PuTTY' to remotely connect to the server using SSH. PuTTY's settings can be seen in Figure 2, whilst the securely connected terminal can be seen in Figure 3.

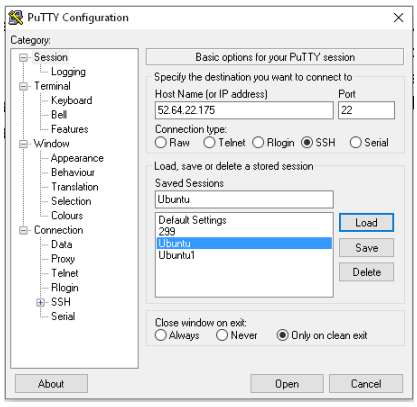

Figure 2: PuTTY Settings

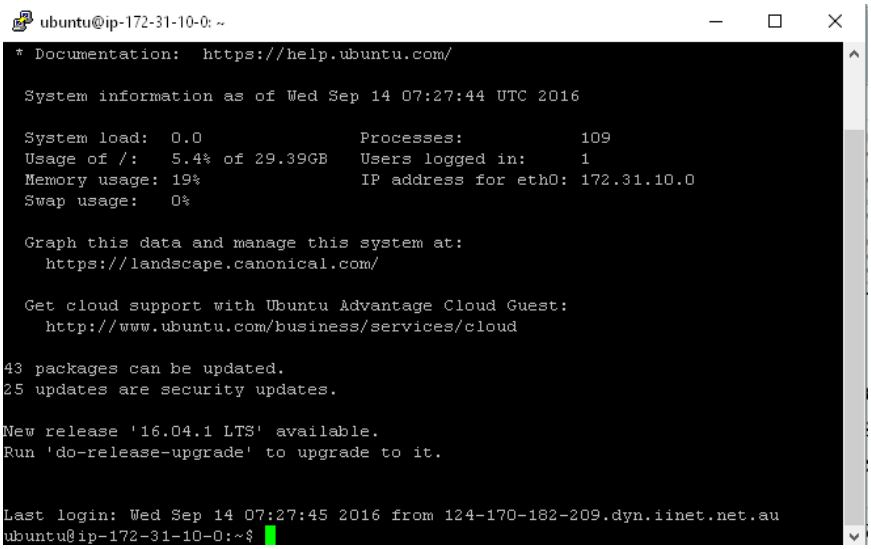

Figure 3: PuTTY - Server connection

Having successfully established a connection to the server, the relevant software packages needed to be installed. For our project a simple LAMP (Linux, Apache, MySQL and PHP) set up was sufficient. Apache2, MySQL and PHP 5 were also installed. This was done following commands provided in an article by Sverdlov [4]. In order to test that the installations had worked correctly, the remote server's IP address was entered into an unrelated web browser; the apache2 information page was displayed as the installation had been successful. Having completed the majority of the configuration process, the final task in enabling online functionality was porting the project from our developer machines to the remote server instance. In order to do this easily and securely, another PuTTY companion program, PSCP was downloaded. PSCP is a secure-copy client which allows simple and relatively secure command-line file transfers. This program was used to copy the zipped web files from the developer machines to the remote server.

Having transferred the necessary files, they were moved into apache's web directory. After this was completed, the IP address of the server (52.64.22.175) could be typed into any browser to access the website as seen in figure 4. This concludes the report detailing James Tuerlings' contribution.

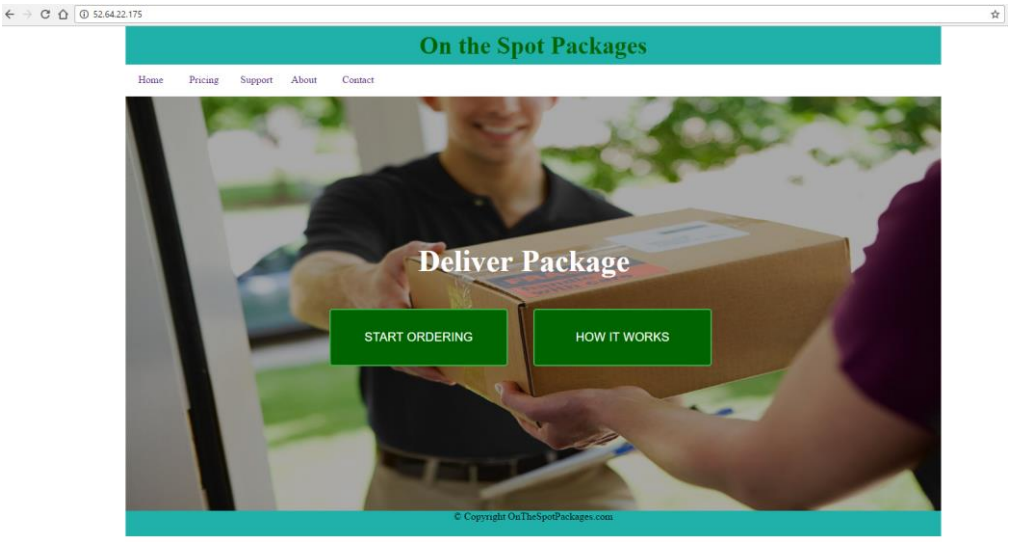

Figure 4: The online website

References

[1] Amazon Web Services. (2016). “Amazon EC2- Virtual Server Hosting”. http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html

[2] Tatham, S. (2016). PuTTY, “SSH and Telnet client” http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

[3] Tatham, S. (2016). PuTTYgen, “PuTTY key generation package” http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

[4] Sverdlov, E. (2012). “How to Install Linux, Apache, MySQL, PHP (LAMP) stack on Ubuntu” https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-on-ubuntu

[5] Tatham, S. (2016). PSCP, “PSCP, command-line secure file copy” http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html