

Manual for the Replication of MrJinIT.com Server

Jin Cherng Chong (33170193)

*School of Engineering and Information Technology
Murdoch University
12/05/2018*

Table of Contents

1.0	Introduction	1
2.0	Amazon EC2 & DNS	2
2.1	Setting up Amazon EC2	2
2.2	Setting up server DNS	6
3.0	Word Press	3
3.1	Setting up Word Press	3
3.2	WordPress Features	3
4.0	Server Scripting	4
4.1	Setting up GitLab	4
5.0	Gog Git Services	5
5.1	Setting Up and Installing Gog	4
5.2	Creating and installing a repository	4
5.3	Uploading files and folders	4
6.0	Conclusion	6
7.0	References	6

Abstract:

This report provides systematic documentation on how I created and customised my own EC2 server. The report should be considered a resource guide that aims to teach students the basics of setting up an EC2 server, WordPress, and a personal web-based git repository (Gitlab). Any of the information that I used to set

1.0 Introduction

This report provides step by step documentation on how I created and customised my own EC2 server. The report should be considered a resource guide that aims to teach students the basics of setting up an EC2 server, WordPress, and a personal web-based git repository (Gogs). My personal WordPress serves as a resume' outlining my qualities and knowledge while my web-based git repository is intended for the storage of my university assignments and notes. My WordPress website have a button, when clicked, will redirects to my GitLab.

This report assumes that readers will have an Amazon EC2 server with the following requirements-

Minimum:

Server: Amazon EC2

OS: Ubuntu Server 16.04 LTS (HVM), SSD Volume Type

Processor: Intel Xeon at 2.5GHZ with 1 vCPUs

Memory: 1 GiB

Storage: 8 GiB

The report also assumes that the reader has the following software installed on their computer-

Putty

PuttyGen

Google Chrome

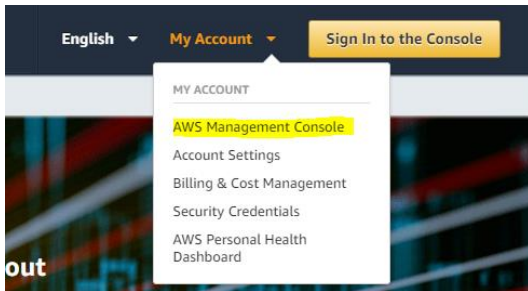
For my personal EC2 server, the amount of storage I have is 20 GiB. But 20 GiB storage is not necessary. The operating system I used in conjunction to the Amazon EC2 operating system is Windows 10 and the domain registrar I refer to in the report is Namesilo.com.

2.0 Amazon EC2 & DNS

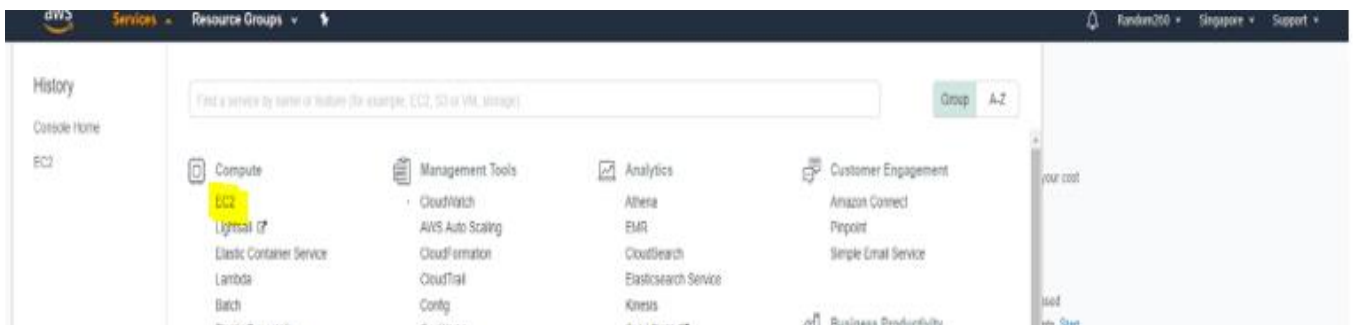
Amazon EC2 is a cloud-based server offered in the Amazon Webserver market place [1]. For small to medium business, it serves as cheaper alternative to buying and maintaining their own server [2]. A company that is purchasing a cloud server only has to really has to consider the renting cost of the server, while a company that wants to own their own server needs to consider additional factors such as the environment where the server is stored and whether or not they need to hire workers to maintain the server. These additional factors will affect the total cost of ownership for a server drastically. A domain name system (DNS) is a system that translates the domain name to the IP address identifying a device [3]. The domain name can be purchased from registrars such as- Namesilo and Crazydomains [4].

2.1 Creating Amazon EC2 Instance

To start, make sure you are logged into your Amazon Webserver Account and are in the AWS Management Console: <https://aws.amazon.com/ec2/>



Once you are in the AWS Management console
Click on the 'Services' tab
Click on the 'EC2' option



You should now see a 'Launch Instance' option.

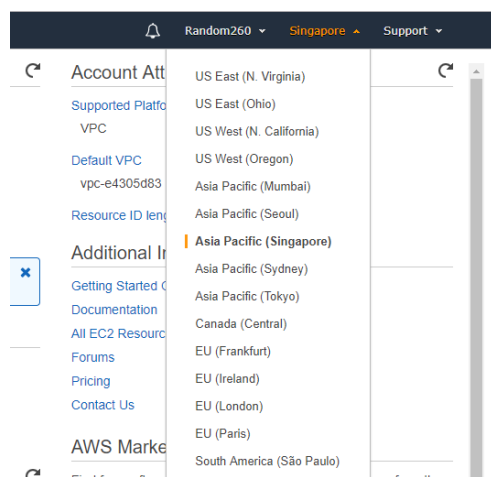
This option allows you to deploy and configure an Amazon EC2 Server.

The Amazon EC2 will deploy in a default location. For me the default location is in Singapore and it is my optimal location. The optimal server location is often determined by figuring out the purpose of your server. If you want to run a gaming server and the majority of your player base is located in Texas, then your server should be closer Texas. Having a gaming server closer to your player base location means that data will be transferred quicker from the gaming server to the player base computer's and vice versa [5]. The result of quicker transfer of data means lower ping and better performance for the player's [5]. However, the optimal location for this server will be the location closest to you. The reason for this is to minimise the time it will take for us to upload our files to the Gitlab.

Click '(The default location)'

Select your optimal location

Click 'Launch Instance'



Create Instance

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

[Launch Instance](#)

Note: Your instances will launch in the Asia Pacific (Singapore) region

[Service Health](#)

[Scheduled Events](#)

Now, it is time to configure our Amazon EC2. You will be confronted by 7 configurations steps- choose AMI, instance type, storage, tags, security groups, and review.

Refer back to page 2 of this report to figure out what 'Amazon Machine Image' and 'Instance Type' to choose.

Leave 'Configure instance' settings as the default.

Refer back to page 2 of this report to work out amount of 'storage' to add

Leave 'Add Tags' settings as the default

For 'Configuring Security Groups' step

Type ssh-wordpress-and-gitlab in 'Security group name:' textbox.
 Click 'Add Rule'
 Click 'Custom TCP Rule' menu in the 'type' column
 Select 'HTTP'

By selecting HTTP, it allows web request to be accepted by our server. This is necessary for our word press website.

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:

Description:

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
Custom TCP	TCP	(e.g. 49152-6553)	Custom 0.0.0.0:::0	e.g. SSH for Admin Desktop

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.

Click 'Review and launch'

This is how my server is configured it should be very similar to your server you just configured.

Step 7: Review Instance Launch

AMI Details [Edit AMI](#)

Ubuntu Server 16.04 LTS (HVM), SSD Volume Type - ami-52d4802e
 Ubuntu Server 16.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).
 Free tier eligible Root Device Type: ebs Virtualization type: hvm

Instance Type [Edit instance type](#)

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

Security Groups [Edit security groups](#)

Security group name: ssh-wordpress-and-gitlab
 Description: launch-wizard-4 created 2018-05-13T03:23:15.206+08:00

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	0.0.0.0/0	
HTTP	TCP	80	0.0.0.0/0	
HTTP	TCP	80	:::0	

Instance Details [Edit instance details](#)

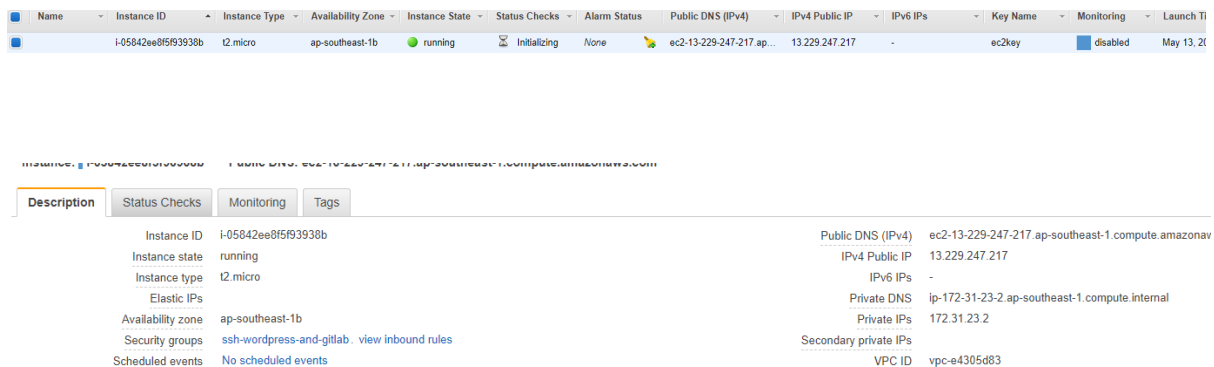
Storage [Edit storage](#)

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encrypted
Root	/dev/sda1	snap-0b495f8e103390e57	20	gp2	100 / 3000	N/A	Yes	Not Encrypted

Tags [Edit tags](#)

Finally:

Click 'Launch' if you are happy.
Select 'Create a new key pair'
Type ec2key or whatever name you want to call key in 'Key pair name' textbox
Click 'Download Key Pair'
Click 'Launch Instances'
Click 'View Instances'



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs	Key Name	Monitoring	Launch Time
	i-05842ee8f5f93938b	t2.micro	ap-southeast-1b	running	Initializing	None	ec2-13-229-247-217.ap...	13.229.247.217	-	ec2key	disabled	May 13, 20...

Description	Status Checks	Monitoring	Tags
Instance ID	i-05842ee8f5f93938b		
Instance state	running		
Instance type	t2.micro		
Elastic IPs			
Availability zone	ap-southeast-1b		
Security groups	ssh-wordpress-and-gitlab. view inbound rules		
Scheduled events	No scheduled events		
Public DNS (IPv4)	ec2-13-229-247-217.ap-southeast-1.compute.amazonaws.com		
IPv4 Public IP	13.229.247.217		
IPv6 IPs	-		
Private DNS	ip-172-31-23-2.ap-southeast-1.compute.internal		
Private IPs	172.31.23.2		
Secondary private IPs			
VPC ID	vpc-e4305d83		

Congratulations!!! Your Amazon EC2 Server should be set up perfectly to begin the installation of WordPress and Gitlab.

IMPORTANT TO ACCESS EC2 SERVER:

Refer to either Windows or Linux link depending on operating system you are using in conjunction to your EC2 server.

Windows: <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/putty.html>

Linux:

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AccessingInstancesLinux.html>

2.2 Setting up server DNS

This section is where you will be setting up your website's DNS service. We will be installing apache2 to help do this. This section is part of the installation of a WordPress website.

Before we begin make sure you are connected an Amazon EC2 server through an ssh client such as Putty. Also, just to be safe update all your packages to their latest versions. The latest packages of software tend to have less vulnerabilities [6]

Type 'sudo apt update' in ssh client. Then type 'y' when prompted
Type 'sudo apt upgrade' in ssh client. Then type 'y' when prompted

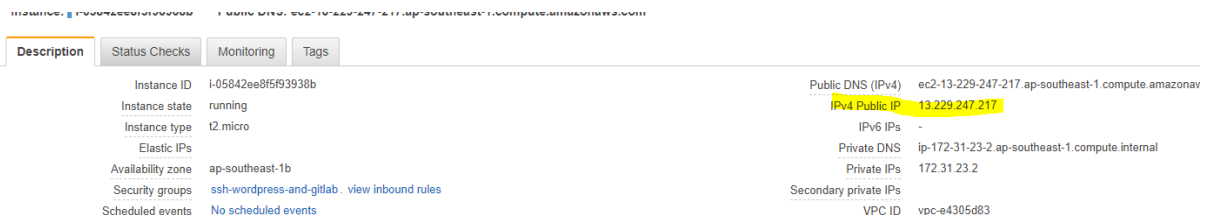
Once upgraded, we need to install apache2.

Type 'sudo apt install apache2' then type 'y' when prompted

To check whether or not the webserver was configured and installed correctly try to go to the website (not from your server web browser). The web addresses for the apache2 web server is the Amazon EC2 'IPv4 Public IP'.

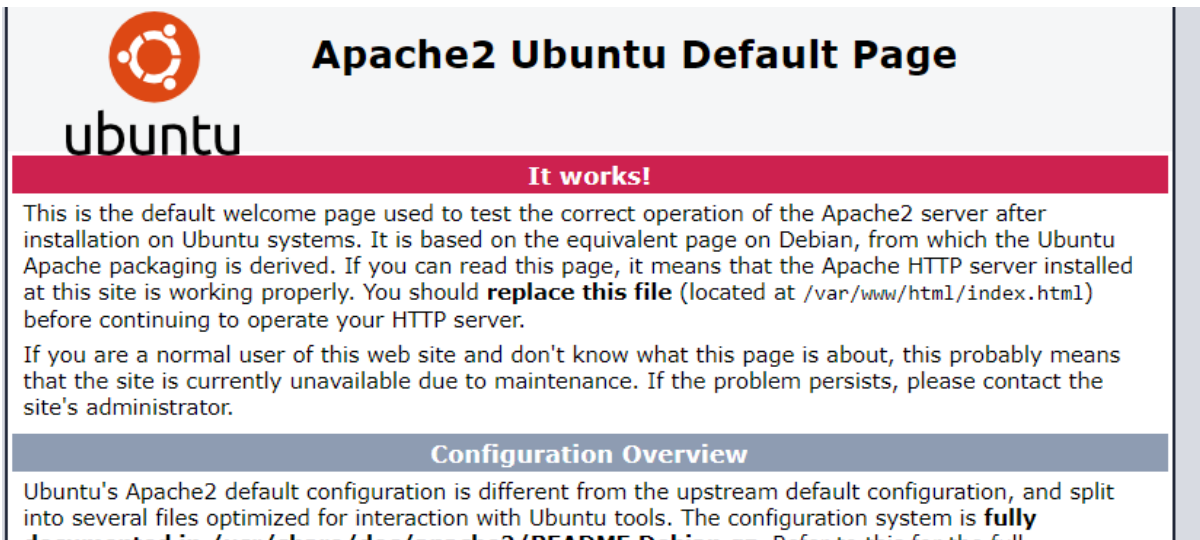
This can be found in the Amazon EC2 dashboard website:

Click 'Instances'
Refer to the IPv4 Public IP at the bottom of the website



Description	Status Checks	Monitoring	Tags
Instance ID	i-05842ee8f5f93938b		
Instance state	running		
Instance type	t2.micro		
Elastic IPs			
Availability zone	ap-southeast-1b		
Security groups	ssh-wordpress-and-gitlab view inbound rules		
Scheduled events	No scheduled events		
Public DNS (IPv4)	ec2-13-229-247-217.ap-southeast-1.compute.amazonaws.com		
IPv4 Public IP	13.229.247.217		
IPv6 IPs	-		
Private DNS	ip-172-31-23-2.ap-southeast-1.compute.internal		
Private IPs	172.31.23.2		
Secondary private IPs			
VPC ID	vpc-e4305d83		

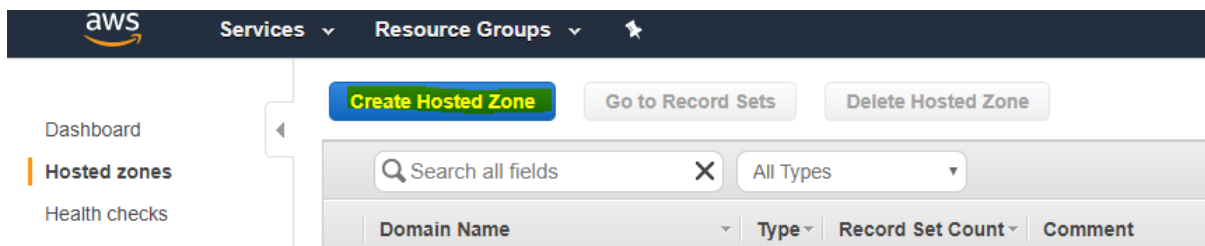
Open your web browser on your main computer and
Copy and paste the 'Ipv4 Public IP' in the web browser's url search box check if you see a website called 'Apache2 Ubuntu Default Page'*



*If the above does not appear then when you tried to go to the apache2 web addresses from a web browser then you most likely made a mistake along the way. The possible reasons for this may include- not adding a HTTP rule (page 4), firewall issues, mistyping, or forgetting to type commands. Try typing in the ssh client 'sudo ufw disable' to turn of firewall. Now, it is time to assign a domain name to the web address. A domain registrar allows for the purchase of a domain name. The one registrar I will be referring to is namesilo.com.

From the Amazon EC2 dashboard

Click 'Services' tab
Click 'Route 53' option found below
Select 'DNS management'
Click 'Create Hosted Zone' button twice



On the right hand side of the website

Enter your domain name in 'Domain Name:' textbox
Click 'Create'

Create Hosted Zone

A hosted zone is a container that holds information about how you want to route traffic for a domain, such as example.com, and its subdomains.

Domain Name: MrJint.com

Comment:

Type: Public Hosted Zone

A public hosted zone determines how traffic is routed on the Internet.

Create

Click on the dot on the left of 'Domain Name'
Select 'Go to Record Sets'
Select on the dot on the left of 'Name'

On the right, you can see the 'value' There should be 4 domain names. What you need to do now is copy these four 'value' and go to your registrar and enter them in the Name Server configuration fields. This process will slightly differ from registrar to registrar. However, I will show you how I did it using through Name silo.

Copy 'value' from record set

Alias: Yes No

TTL (Seconds): 172800

Value: ns-1396.awsdns-46.org.
 ns-1816.awsdns-35.co.uk.
 ns-113.awsdns-14.com.

The domain name of a name server.
 Enter multiple name servers on separate lines.

Go to Registrar (Namesilo.com)
Sign in
Click 'Manage My Domains'
Click on the domain so you get to 'Domain Console'
Select 'Change' on the right of Nameservers heading

Print Certificate:

NameServers ([Change](#))

ns-148.awsdns-18.com
 ns-1159.awsdns-16.org
 ns-1764.awsdns-28.co.uk
 ns-1012.awsdns-62.net

[View/Manage Registered NameServers](#)

*Paste all 4 “value” from record set into the ‘NameServer:’ textbox
Press ‘submit’*

NameServer 1: ns-860.awsdns-43.net *
NameServer 2: ns-1396.awsdns-46.org *
NameServer 3: ns-1816.awsdns-35.co.uk
NameServer 4: ns-1012.awsdns-62.net
NameServer 5: ns-113.awsdns-14.com|
NameServer 6:

SUBMIT

There is an issue with this method is that a restart to the EC2 may cause IP to change thus DNS may not point to the server address. To circumvent this, use an elastic ip address.
So from to EC2 dashboard:

*Click ‘Elastic IPs’ then ‘Allocate new address’
Click ‘Action’ and ‘Associate with instance’*

3.0 Word Press

In this section, I will be explaining how to install, set up word press, and add features. Before we can actually install WordPress there are a number of software we need to install these include: php7, MySQL, and WordPress [7].

3.1 Setting up WordPress

We need to remove the index.html because when we install WordPress we should use word press’s index.html

```
sudo rm /var/www/index.html
```

Then we should install and configure mysql to work with WordPress. The package **mysql** is a database management system, which is needed for WordPress. The database for WordPress will also be created

```
sudo apt-get install mysql-server php7.0-mysql  
Enter password of mysql  
mysql -u root -p  
Type create database [enter database name here];  
Type create user ‘[Enter username]’@’localhost’ Identified by ‘[Enter WordPress password]’;  
Type GRANT ALL PRIVILEGES ON [database name].* TO ‘[Username entered above]’@’localhost’;  
Enter FLUSH PRIVILEGES;  
Type Exit;
```

```
affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database wordpress;
Query OK, 1 row affected (0.00 sec)

mysql> create user 'wordpress'@'localhost' Identified by '' '0.7.6';
Query OK, 0 rows affected (0.00 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql> Exit;
```

The next step is installing **php7.0**

```
sudo apt-get install php7.0 libapache2-mod-php7.0 php7.0-cgi
```

After installing php7.0, we will need to install Wordpress and modify one of the files to link it to the newly created database. The first step is going to the /var/www/html directory. Since it is in the public web directory it can be stored and accessed by anyone with permission.

```
Type cd /var/www/html
Type sudo wget https://wordpress.org/latest.zip
Type sudo apt-get unzip
Type sudo unzip latest.zip
Enter sudo rm latest.zip
Type sudo cp -r wordpress/* /var/www/html
Finally type sudo rm -r wordpress
```

It is also important to change the permissions of the files and folders.

```
Type sudo chown -r www-data:www-data /var/www/html/
Type sudo chmod -R 755 /var/www/html/
```

```
ubuntu@ip-172-31-23-2:/var/www/html$ ls -la
total 200
drwxr-xr-x  5 www-data www-data 4096 May 18 19:36 .
drwxr-xr-x  3 root      root    4096 May 12 21:37 ..
-rwxr-xr-x  1 www-data www-data  418 May 18 19:35 index.php
-rwxr-xr-x  1 www-data www-data 19935 May 18 19:35 license.txt
-rwxr-xr-x  1 www-data www-data  7415 May 18 19:35 readme.html
-rwxr-xr-x  1 www-data www-data  5458 May 18 19:35 wp-activate.php
drwxr-xr-x  9 www-data www-data 4096 May 18 19:35 wp-admin
-rwxr-xr-x  1 www-data www-data   364 May 18 19:35 wp-blog-header.php
-rwxr-xr-x  1 www-data www-data 1889 May 18 19:35 wp-comments-post.php
-rwxr-xr-x  1 www-data www-data  2853 May 18 19:35 wp-config-sample.php
drwxr-xr-x  4 www-data www-data 4096 May 18 19:35 wp-content
-rwxr-xr-x  1 www-data www-data  3669 May 18 19:35 wp-cron.php
drwxr-xr-x 18 www-data www-data 12288 May 18 19:35 wp-includes
-rwxr-xr-x  1 www-data www-data  2422 May 18 19:35 wp-links-opml.php
-rwxr-xr-x  1 www-data www-data  3306 May 18 19:35 wp-load.php
-rwxr-xr-x  1 www-data www-data 37760 May 18 19:35 wp-login.php
-rwxr-xr-x  1 www-data www-data  8048 May 18 19:35 wp-mail.php
-rwxr-xr-x  1 www-data www-data 16246 May 18 19:35 wp-settings.php
-rwxr-xr-x  1 www-data www-data 30091 May 18 19:35 wp-signup.php
-rwxr-xr-x  1 www-data www-data  4620 May 18 19:35 wp-trackback.php
-rwxr-xr-x  1 www-data www-data  3065 May 18 19:35 xmlrpc.php
ubuntu@ip-172-31-23-2:/var/www/html$
```

The last couple of steps would be modifying the configuration file to link to the database created earlier. We want to rename the file as well since the current name has the word ‘sample’ in it [7].

```
Enter sudo mv wp-config-sample.php wp-config.php
Type sudo nano wp-config.php
Change Db_name, DB_usr and DB_Password with configurations typed earlier
Type sudo service apache2 restart
Type sudo service mysql restart
```

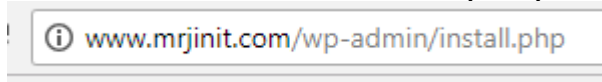
```
/** The name of the database for WordPress */
define('DB_NAME', 'wordpress');

/** MySQL database username */
define('DB_USER', 'wordpress');

/** MySQL database password */
define('DB_PASSWORD', 'helloworld');

/** MySQL hostname */
define('DB_HOST', 'localhost');
```

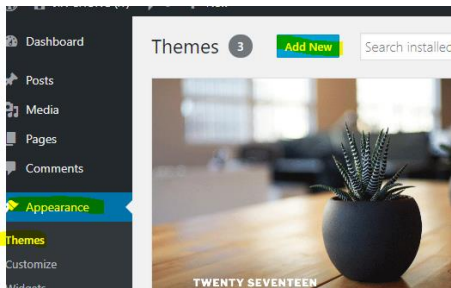
Now check if everything is installed correctly by going to either the server’s domain name or servers IP address. Then follow the prompts to complete installation.



3.2 WordPress Features

One of the important features is learning how to **install a theme**. A theme is a collection of files that are designed to work together in order to produce a graphical user interface for a weblog [8]. WordPress allows the installation of free themes or paid themes at different price points. The parent theme I used for my website is Genesis Framework, while the child theme is beautiful pro theme.

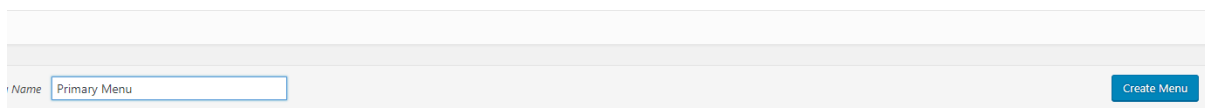
Go to Dashboard
 Hover mouse over menu 'Appearance'
 Click 'Themes'
 Click 'Add new'
 Click 'Upload theme and Choose File'
 Choose and Upload zip file with the theme. (It can be located in any directory on server)
 Click 'Activate'



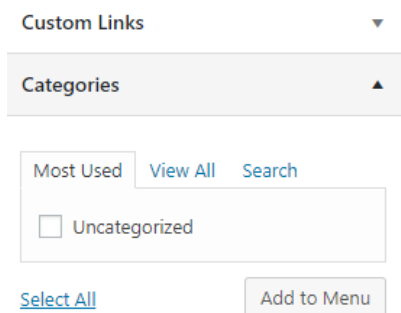
Another Important feature is creating a **category menu**:

A custom-link menu function is when a user clicks on the menu he will automatically go to the url. The url may be a website or it may be the path to download a file. Another menu type offered is categories menu. This menu allows for submenus. Finally, when a post menu is created and clicked it will redirect to the URL of the post created. Below are screenshots are showing how to create a category menu.

Go to Dashboard
 Hover over 'Appearances'
 Click 'Menu'
 State the 'Menu Name' and click create menu



Click 'Categories' on the left hand side



Check 'Uncategorized'

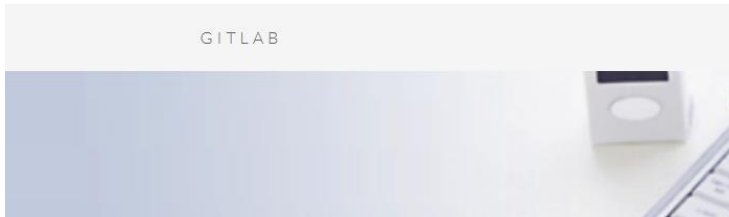
Click 'Add to menu'

Expand 'Uncategorized' and change label to ideal name of menu for it is: GitLab

Check 'Primary Navigation Menu' (allows the menu to appear on home page at top)

Save menu

JIN CHONG (IT)



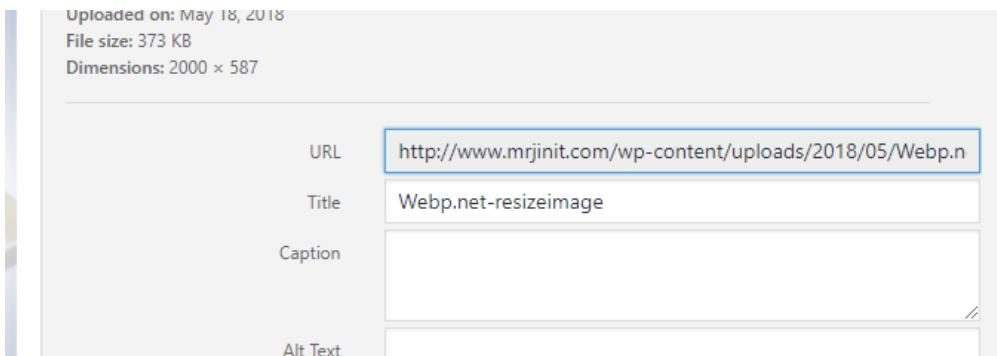
The next feature I have implemented in my website was **creating a menu and linking to an item** on my server. (Custom Links)

Go to Dashboard

Click 'Media' and 'Upload New Media'

Click on Uploaded Media

Copy 'URL'



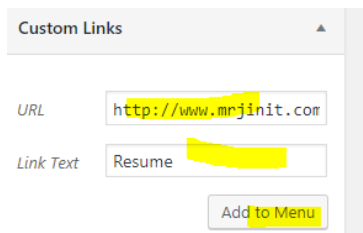
Go to Dashboard

Hover over 'Appearances'

Click 'Menu'

Expand 'Custom Links' and change url to the url copied

Change 'Link text' to name of menu and Add to Menu then Save Menu



The Third feature is **adding widgets**:-

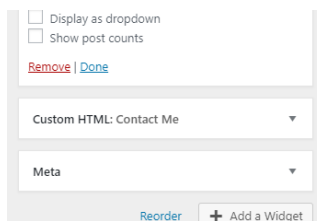
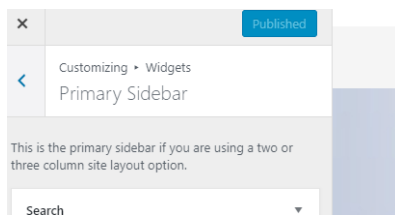
The changing of widgets is quite simple.

Open 'Customisation' tab on the home page

Click 'Widget'

Select 'Primary side bar'

Then you can add or delete widgets on the right hand side



The fourth feature is **changing the background**

Go to Wordpress dashboard
Click 'Media' and then 'Add new' to upload background image
Click on the image to get url
Click on 'Editor' found in 'Appearances' menu
Replace

```
body {  
    background-color: #fff;  
    color: #666;  
    font-family: 'Lato', sans-serif;  
    font-size: 18px;  
    font-weight: 300;  
    line-height: 1.625;  
}
```

With

```
body {  
    background-image: url('http://www.mrjinit.com/wp-  
content/uploads/2018/05/3147.jpg');  
    font-family: 'Lato', sans-serif;  
    font-size: 18px;  
    font-weight: 300;  
    line-height: 1.625;  
}
```

4.0 Server Scripting

```
1 #!/bin/bash
2 #This script does three functions: Print server health + Backup Word Files and Database every day
3
4 #Print server health
5 now=$(date +%d_%m_%y)
6 cd /home/ubuntu/Backup
7 mkdir $now
8 echo "Date of System Health information:" > $now/$now.txt
9 echo "" >> $now/$now.txt
10 date >> $now/$now.txt
11 echo "-----" >> $now/$now.txt
12 echo "Current Operating System Information:" >> $now/$now.txt
13 echo "" >> $now/$now.txt
14 uname -a >> $now/$now.txt
15 echo "-----" >> $now/$now.txt
16 echo "Current Server Uptime:" >> $now/$now.txt
17 echo "" >> $now/$now.txt
18 uptime >> $now/$now.txt
19 echo "-----" >> $now/$now.txt
20 echo "Disk Usage:" >> $now/$now.txt
21 echo "" >> $now/$now.txt
22 df >> $now/$now.txt
23 echo "-----" >> $now/$now.txt
24 echo "Last Logins:" >> $now/$now.txt
25 echo "" >> $now/$now.txt
26 last -a >> $now/$now.txt
27 echo "-----" >> $now/$now.txt
28
29 #Shows Wordpress themes
30 echo "WordPress Themes uploads:" >> /home/ubuntu/Backup/$now/$now.txt
31 echo "" >> /home/ubuntu/Backup/$now/$now.txt
32 cd /
33 cd /var/www/html/wp-content/themes
34 ls >> /home/ubuntu/Backup/$now/$now.txt
35
36
37 #WordPress Database Backup
38 cd
39 cd /home/ubuntu/Backup/$now
40 mysqldump --add-drop-table -h localhost -u wordpress -p[password] wordpress | bzip2 -c > ubuntu.bak.sql.bz2
41
42 #WordPress File Backup
43 cp -r /var/www/html /home/ubuntu/Backup/$now
44
45
```

The above is my server scripting code, which I configured to run once every day. This script has three main objectives. The three are- printing different types of server information and storing it in a text file, backing up the MySQL database in a zip file, and backing up all the WordPress files. All the three files: system information text file, MySQL zip and WordPress files are stored in a directory with the name of the directory being the date. This script is useful for me since I plan on migrating my WordPress website to a more cost efficient virtual private server in the future. I also use this script a lot while I am modifying my website. All it takes is a simple 'jinscript' command and everything is backed up, sort of, like how with Microsoft Word you would press ctrl + s to save every so often.

The commands I used for the script are commonly recognised commands. The echo command will output a string. For my script, the string would be written in a text file, with the date as the

txt file's name. The additional commands implemented are also added to text file created earlier.

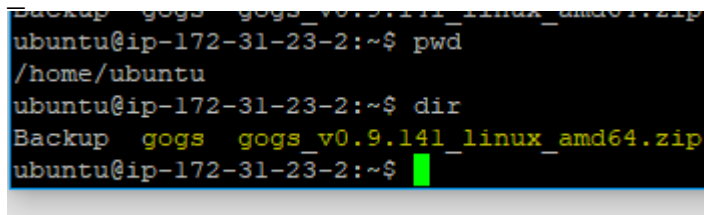
5.0 Gog Git Service

For my server, I decided to go with a lightweight alternative to Gitlab/Github. The Amazon Ec2 instance does not support Gitlab since the minimum specifications needed is 2GB of ram while our instance is 512mb[x]. Please understand that the git repository you are about to install is intended for you only to use. This manual will not support additional user's registrations on your hosted Git service. However, it is possible to run this Git service for many other users.

5.1 Setting Up and Installing Gog

Firstly, download the Gog repository from <https://dl.gogs.io/> [9]. Also create database for Gog

```
Type cd /home/ubuntu
Type wget https://dl.gogs.io/gogs_v0.9.141_linux_amd64.zip
Type unzip gogs_v0.9.141_linux_amd64.zip
Type mysql -u root -p
Type CREATE DATABASE gogs CHARACTER SET utf8mb4 COLLATE
utf8mb4_general_ci;
Type \q
```

A terminal window screenshot showing the command 'dir' being executed. The output lists a directory named 'Backup' containing files 'gogs' and 'gogs_v0.9.141_linux_amd64.zip'. The prompt is 'ubuntu@ip-172-31-23-2:~\$'.

Next step we need to open an inbound port in order for Gogs to work. The port we will be opening is port 3000.

```
Go to Amazon EC2 dashboard
Click 'Security Groups' option on the left
Click on the group name you are using
Click 'Inbound' bottom located at bottom.
Click 'Edit'
Click 'Add Rule'
Type '3000' in port range
Click 'Custom' in source column and change to 'Anywhere' then click save
```

Custom TCP F ▾	TCP	443	Custom ▾	::/0	e.g. SSH for Admin Desktop	✕
Custom TCP F ▾	TCP	3000	Anywhere ▾	0.0.0.0/0, ::/0	Gogs	✕

Add Rule

NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.

Cancel **Save**

Then we will need to run Gogs and set it up

Type `cd /home/ubuntu/gogs`
 Type `./gogs web`
 Go to Firefox and enter IP_ address:3000
 Fill in the details

Please note below are the properties I modified for my server. You should also modify it to suit you.

Password *

Application Name *

Put your organization name here huge and loud!

(Application name is your name of git hub)

Domain *

This affects SSH clone URL

(Should be your Amazon EC2 IP address.)

Application URL *

(Should be your IP address and port)

Then the last step is to create your account where you will upload content and *Click* 'Install'

▼ Admin Account Settings

not have to create an admin account right now, user whoever ID=1 will gain admin access autor

Username

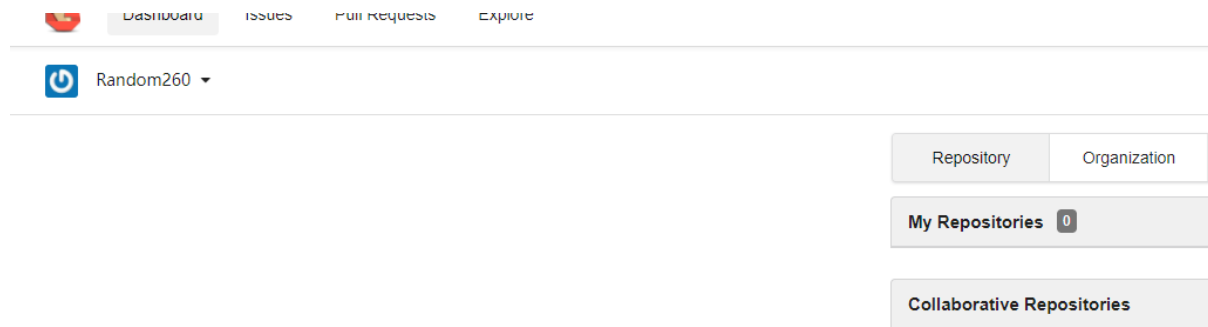
Password

Confirm Password

Admin Email

Install Gogs

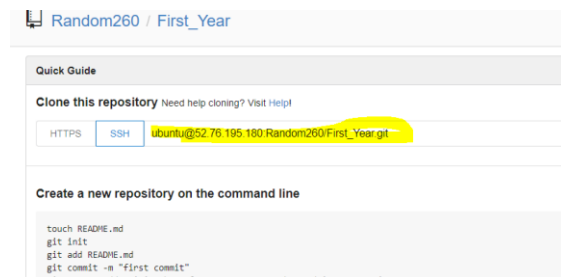
Well done you've just installed your own Gog! It should look like this



5.2 Creating Repository

The process to upload and create repositories is quite simple it will be done through ssh.

Go to your dashboard
Click '+' button
Fill out the 'repository name' then create

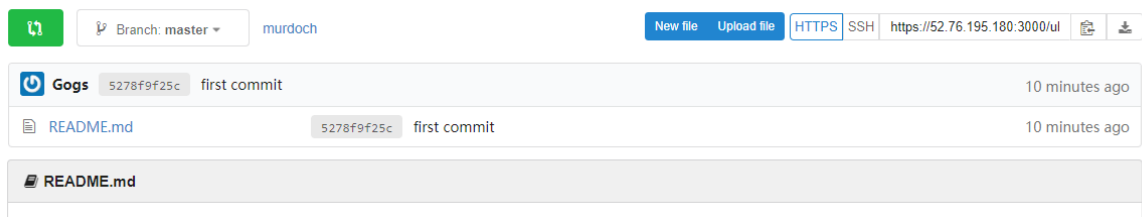


```

Go ssh and log in
Type mkdir [foldername]
Cd [foldername path]
Type git init
Type git clone [local path of repository created above]
Type touch README.md
Type git add README.md
Type git commit -m "commit message"
Type git remote add origin [local path of initial created repository]
Type git push origin master

```

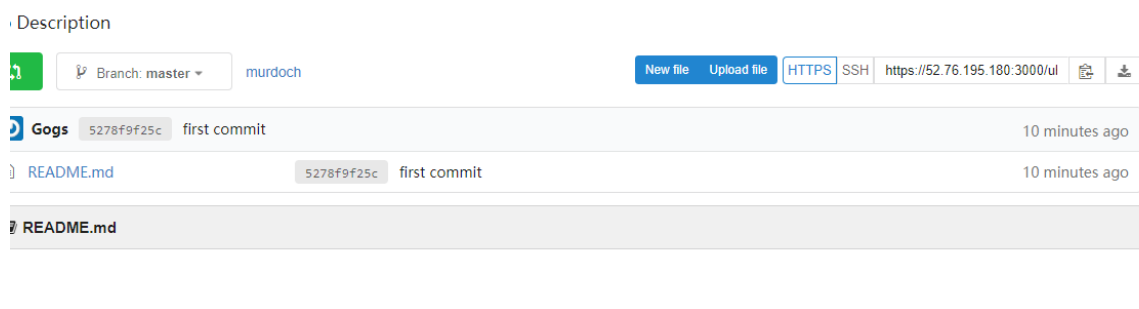
Upon completion of those steps, you can finally see a readme file in repository



5.3 Uploading Files and folders:

There are two different ways of uploading files and folders on your personal git repository. The first method is doing it through the graphical user interface of your git repository website and the second is through the ssh client. I will briefly go over the two methods.

Graphical user interface method:



So to create a file click → 'New file' while
upload file click → 'Upload file'

SSH method:

There are three essential commands used all the time to upload fields and folder [10]. This Manual assume you know how to upload files to your server. The general rule is in order to

upload a file to Git repository you must add it → commit it → push it. In addition, it is important to know that to upload folders you need to have at least one file in it and the steps for uploading files and folders are the same.

Go to ssh and be in the directory you created before you cloned

Type git add [filename]

Type git commit -m “[commit message in the quotation marks]”

Type git push origin master

(The above steps apply to folders as well. The difference is a file must be in the folder and we do not add the folder we add the file in it so essentially the same steps are applied.)

6.0 Conclusion

I hope you found this manual useful. This report aimed to assist you in your understanding of git repositories and WordPress by providing you information on how I created my git repository and WordPress website.

7.0 References

- [1] <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html>
- [2] <https://www.salesforce.com/uk/blog/2015/11/why-move-to-the-cloud-10-benefits-of-cloud-computing.html>
- [3] <https://venturebeat.com/2014/12/31/league-of-legends-servers-are-not-over-capacity-so-heres-why-your-ping-stinks/>
- [4] <https://www.namesilo.com/>
- [5] <https://techwombat.com/how-to-install-wordpress-lamp-on-ubuntu-16-04-vps/>
- [6] <http://www.saxonsgroup.com.au/blog/tech/5-reasons-important-update-software-regularly/>
- [7] <https://www.digitalocean.com/community/tutorials/how-to-install-wordpress-with-lamp-on-ubuntu-16-04>
- [8] <https://codex.wordpress.org/Themes>
- [9] <https://kenfavors.com/code/how-to-install-gogs-on-ubuntu-16-04/>
- [x] <https://docs.gitlab.com/ee/install/requirements.html>
- [10] <http://rogerdudler.github.io/git-guide/>

- Fix up structure
- Add video explaining script
- Gitlab???
-

LINK:

<https://techwombat.com/how-to-install-wordpress-lamp-on-ubuntu-16-04-vps/>
<https://www.atlantic.net/managed-hosting/how-to-install-wordpress-ubuntu-16-04/>
<https://www.tecmint.com/install-wordpress-on-ubuntu-16-04-with-lamp/>

PHP

<https://www.rosehosting.com/blog/install-phpmyadmin-on-ubuntu-16-04/>

1)