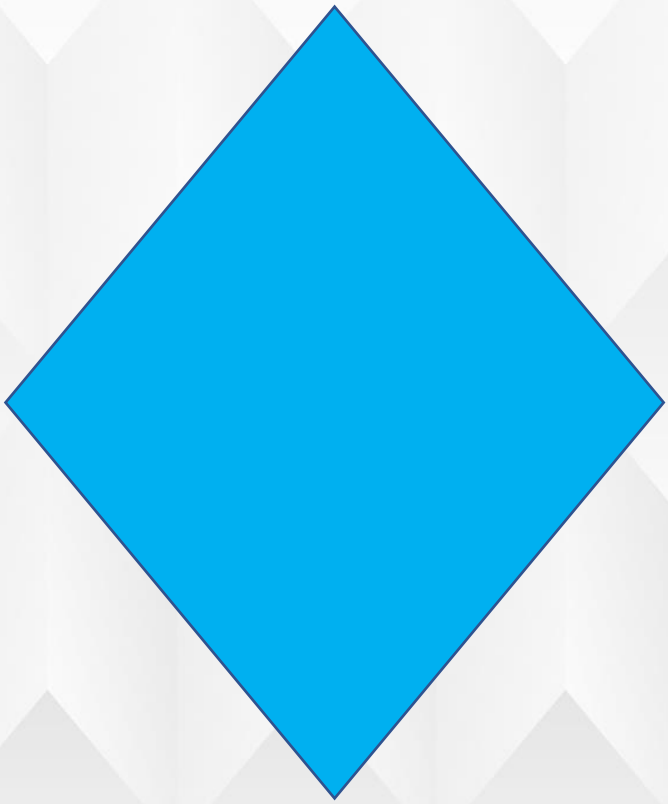


Diamond Supply Co.

Project and Portfolio II

Nathan O'Neal

Full Sail University

Table of Contents

Inventory

EQUIPMENT	OPERATIING SYSTEM	ADDITIONAL INFO	IP ADDRESS
Router/Custom Network	MacOS Big Sur	-	10.10.229.1
Docker	CentOS	Ghost Container	10.10.229.11
NginX Reverse Proxy	CentOS	Reverse Proxy	10.10.229.10
Wordpress	Ubuntu	LAMP Stack running WordPress	10.10.229.12

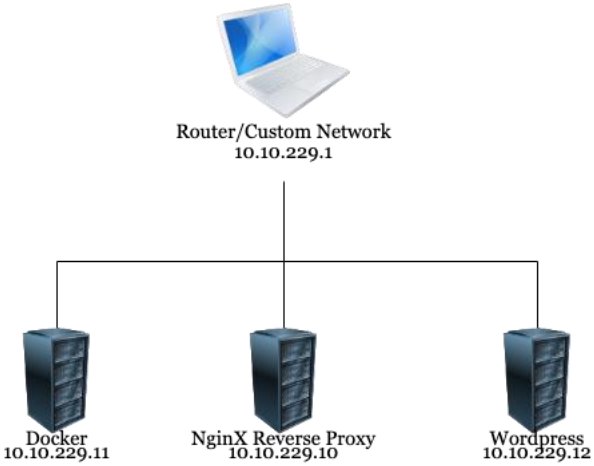
Custom Network

NETWORK NAME	SUBNET IP	SUBNET MASK	DNS	GATEWAY
ITE229	10.10.229.0	255.255.255.0	10.10.229.1	10.10.229.1

IDs and Passwords

ACCOUNT	USER ID	PASSWORD
CentOS Root User	root	Fullsail1!
CentOS Network User	root	Fullsail1!
CentOS Host User	noneal	Fullsail1!
MySQL Network User	root	Fullsail1!
MySQL Host User	noneal	0yi8jd30u1
WordPress Admin	admin	EORiFe30Gu977PlM01

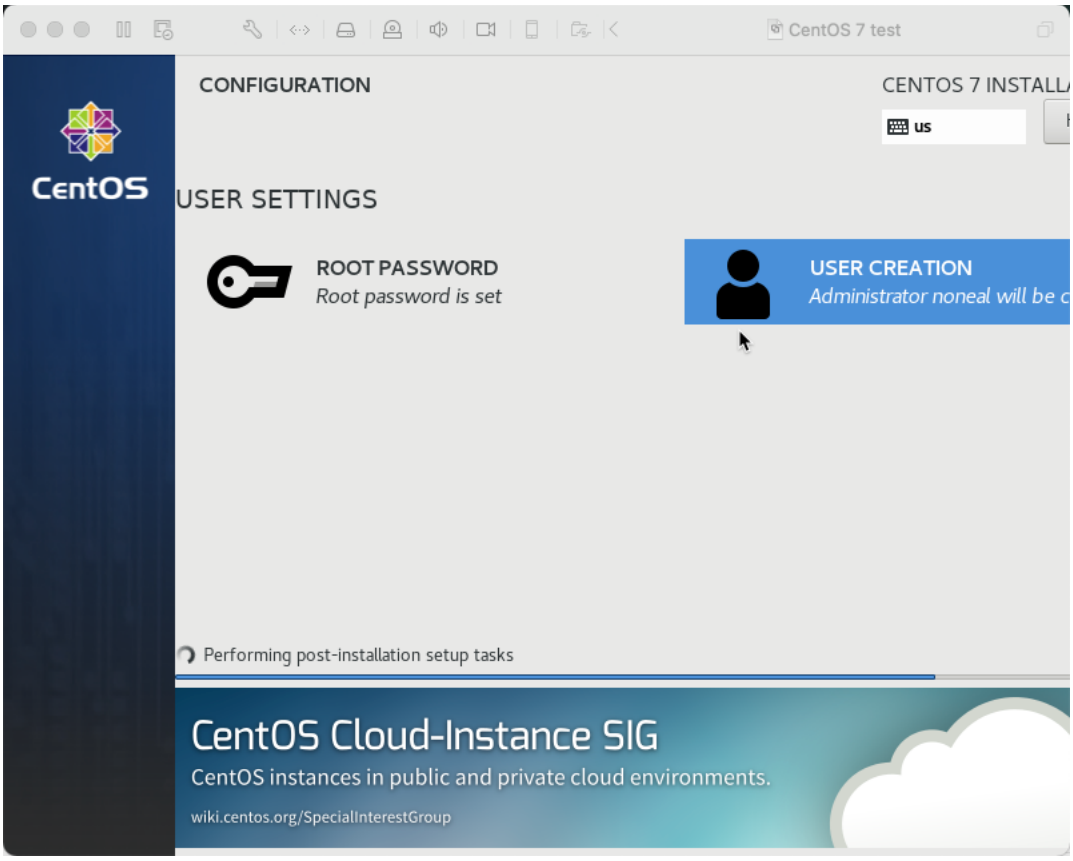
Network Topology Diagram



Node.js Application (Ghost) on Docker

Install base CentOS 7 Virtual Machine "ITE229-docker (11)"

1. Set network custom network (10.10.229.1/24)
2. Set processors to **2** with **1024 MB** of memory
3. Select installation destination to **VMware, VMware Virtual S**
4. Click on **Network & Host Name** and press **Configure**
5. Click on **IPv4 Settings** then click **Add**
6. Use the following settings
    a. Address- **10.10.229.11**
    b. Netmask- **255.255.255.0**
    c. Gateway- **10.10.229.1**
7. Click **Done** and then click **Begin Installation**
8. Click on **Root Password** and set it to **Fullsail1!**



SSH into CentOS VM

1. On your terminal use the command **ssh root@10.10.229.11**
2. Type **yes** to accept the key
3. Type in the root password (**Fullsail1!**)

Update CentOS

1. Use the command **yum update -y**

```
 vim-minimal                x86_64 2:7.4.629-8.el7_9           updates 443 k
 virt-what                  x86_64 1.18-4.el7_9.1              updates  30 k
 wpa_supplicant             x86_64 1:2.6-12.el7_9.2            updates 1.2 M
 zlib                       x86_64 1.2.7-20.el7_9              updates  90 k

Transaction Summary
================================================================================
Install    1 Package
Upgrade  112 Packages

Total download size: 252 M
Downloading packages:
Delta RPMs disabled because /usr/bin/applydeltarpm not installed.
warning: /var/cache/yum/x86_64/7/updates/packages/NetworkManager-libnm-1.18.8-2.
el7_9.x86_64.rpm: Header V3 RSA/SHA256 Signature, key ID f4a80eb5: NOKEY
Public key for NetworkManager-libnm-1.18.8-2.el7_9.x86_64.rpm is not installed
(1/113): NetworkManager-libnm-1.18.8-2.el7_9.x86_64.rpm    | 1.7 MB   00:02
(2/113): bash-4.2.46-35.el7_9.x86_64.rpm                   | 1.0 MB   00:03
(3/113): NetworkManager-wifi-1.18.8-2.el7_9.x86_64.rpm     | 202 kB   00:06
(4/113): bind-export-libs-9.11.4-26.P2.el7_9.9.x86_64.rpm  | 1.1 MB   00:08
(5/113): ca-certificates-2021.2.50-72.el7_9.noarch.rpm     | 379 kB   00:04
(6/113): centos-release-7-9.2009.1.el7.centos.x86_64.rpm   |  27 kB   00:00
(7/113): coreutils-8.22-24.el7_9.2.x86_64.rpm              | 3.3 MB   00:06
(10/113): NetworkManager-t 4% [-                ] 483 kB/s |  11 MB   08:32 ETA
```

Install EPEL Packages

1. Use the command **yum install epel-release -y**

```
Loading mirror speeds from cached hostfile
 * base: mirror.math.princeton.edu
 * extras: mirror.net.cen.ct.gov
 * updates: mirror.atl.genesisadaptive.com
Resolving Dependencies
--> Running transaction check
---> Package epel-release.noarch 0:7-11 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package                Arch             Version          Repository        Size
================================================================================
Installing:
 epel-release           noarch           7-11             extras            15 k

Transaction Summary
================================================================================
Install  1 Package

Total download size: 15 k
Installed size: 24 k
Is this ok [y/d/N]:
```

Install Docker CE

1. Use the command **curl -fsSL https://get.docker.com -o get-docker.sh**
2. Use the command **sudo sh get-docker.sh**

```
adding repo from: https://download.docker.com/linux/centos/docker-ce.repo
grabbing file https://download.docker.com/linux/centos/docker-ce.repo to /etc/yu
m.repos.d/docker-ce.repo
repo saved to /etc/yum.repos.d/docker-ce.repo
+ '[' stable '!=' stable ']'
+ sh -c 'yum makecache'
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
epel/x86_64/metalink                                        |  19 kB     00:00
 * base: mirror.math.princeton.edu
 * epel: download-ib01.fedoraproject.org
 * extras: mirror.net.cen.ct.gov
 * updates: mirror.atl.genesisadaptive.com
base                                                        | 3.6 kB     00:00
docker-ce-stable                                            | 3.5 kB     00:00
extras                                                      | 2.9 kB     00:00
updates                                                     | 2.9 kB     00:00
(1/13): docker-ce-stable/7/x86_64/filelists_db                |  32 kB   00:00
(2/13): docker-ce-stable/7/x86_64/updateinfo                  |   55 B   00:00
(3/13): docker-ce-stable/7/x86_64/primary_db                  |  78 kB   00:00
(4/13): docker-ce-stable/7/x86_64/other_db                    | 124 kB   00:00
(5/13): epel/x86_64/prestodelta                               | 9.1 kB   00:01
(6/13): extras/7/x86_64/filelists_db                          | 277 kB   00:01
```

Start and Enable Docker

1. Use command **systemctl start docker**
2. Use command **systemctl enable docker**

```
[root@localhost ~]# systemctl start docker
[root@localhost ~]# systemctl enable docker
Created symlink from /etc/systemd/system/multi-user.target.wants/docker.service
to /usr/lib/systemd/system/docker.service.
[root@localhost ~]# systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor prese
t: disabled)
   Active: active (running) since Mon 2022-06-06 18:11:23 EDT; 48s ago
     Docs: https://docs.docker.com
 Main PID: 43221 (dockerd)
   CGroup: /system.slice/docker.service
           └─43221 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/con...
```

Test Docker (hello-world)

1. Use command **docker run hello-world**

```
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

Disable SELinux on CentOS 7 Virtual Machine

1. Use command **nano /etc/selinux/config**
    a. If nano is not installed use the command **yum install nano**
2. Change **SELINUX=enforcing** to **SELINUX=disabled**
3. Exit and save

```
nathanoneal — root@docker:~ — ssh root@10.10.229.11 — 88×34
  GNU nano 2.3.1            File: /etc/selinux/config

# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of three values:
#     targeted - Targeted processes are protected,
#     minimum - Modification of targeted policy. Only selected processes are protected.
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

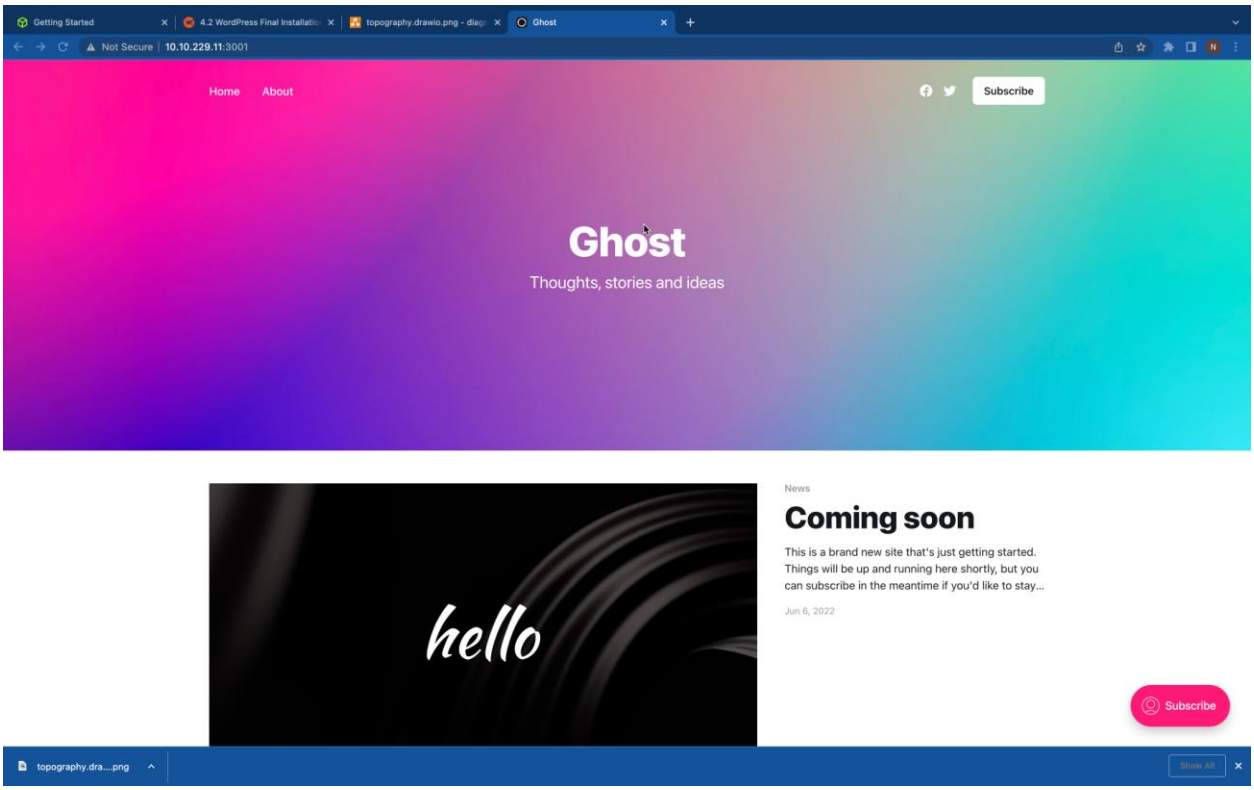
Install Ghost Docker Container

1. Use command **docker run -d –name ghost -p 3001:2368 -e url=http://10.10.229.11:3001 ghost**

```
[root@localhost ~]# docker run -d --name ghost -p 3001:2368 -e url=http://10.10.
229.11:3001 ghost
Unable to find image 'ghost:latest' locally
latest: Pulling from library/ghost
42c077c10790: Downloading  13.06MB/31.38MB
1d1b4cabe4ab: Download complete
6a3018913cd2: Downloading  14.01MB/34.52MB
c4fc3bf11f21: Download complete
9e869d8b07a7: Download complete
a974149cb6d6: Downloading  933.1kB/1.448MB
232908ff12e6: Waiting
16c1304c6525: Waiting
d8cc3f8b58a4: Waiting
```

Test Ghost

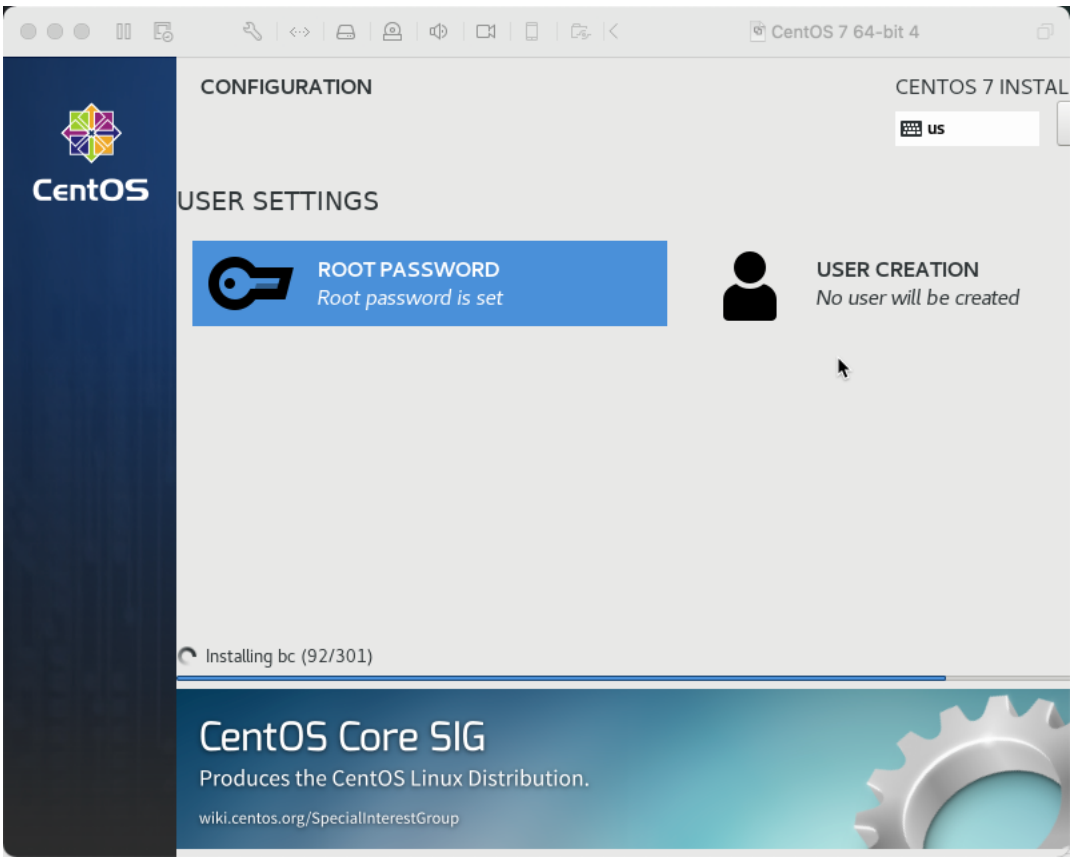
1. Open your browser and go to **http://10.10.229.11:3001**



NginX Reverse Proxy

Install base CentOS 7 Virtual Machine "ITE229-NginX (10)

1. Set network custom network (10.10.229.1/24)
2. Set processors to **1** with **1024 MB** of memory
3. Select installation destination to **VMware, VMware Virtual S**
4. Click on **Network & Host Name** and press **Configure**
5. Click on **IPv4 Settings** then click **Add**
6. Use the following settings
    a. Address- **10.10.229.10**
    b. Netmask- **255.255.255.0**
    c. Gateway- **10.10.229.1**
7. Click **Done** and then click **Begin Installation**
8. Click on **Root Password** and set it to **Fullsail1!**



SSH into CentOS VM

1. On your terminal use the command **ssh root@10.10.229.10**
2. Type **yes** to accept the key

3. Type in the root password (**Fullsail1!**)

```
nathanoneal@Nathans-MacBook-Pro ~ % ssh root@10.10.229.10
root@10.10.229.10's password:
Last login: Mon Jun  6 18:46:09 2022 from gateway
[root@nginx ~]#
```

Update CentOS

1. Use the command **yum update -y**

```
te
---> Package sudo.x86_64 0:1.8.23-10.el7 will be updated
---> Package sudo.x86_64 0:1.8.23-10.el7_9.2 will be an update
---> Package systemd.x86_64 0:219-78.el7 will be updated
---> Package systemd.x86_64 0:219-78.el7_9.5 will be an update
---> Package systemd-libs.x86_64 0:219-78.el7 will be updated
---> Package systemd-libs.x86_64 0:219-78.el7_9.5 will be an update
---> Package systemd-sysv.x86_64 0:219-78.el7 will be updated
---> Package systemd-sysv.x86_64 0:219-78.el7_9.5 will be an update
---> Package tuned.noarch 0:2.11.0-9.el7 will be updated
---> Package tuned.noarch 0:2.11.0-11.el7_9 will be an update
---> Package tzdata.noarch 0:2020a-1.el7 will be updated
---> Package tzdata.noarch 0:2022a-1.el7 will be an update
---> Package util-linux.x86_64 0:2.23.2-65.el7 will be updated
---> Package util-linux.x86_64 0:2.23.2-65.el7_9.1 will be an update
---> Package vim-minimal.x86_64 2:7.4.629-7.el7 will be updated
---> Package vim-minimal.x86_64 2:7.4.629-8.el7_9 will be an update
---> Package virt-what.x86_64 0:1.18-4.el7 will be updated
---> Package virt-what.x86_64 0:1.18-4.el7_9.1 will be an update
---> Package wpa_supplicant.x86_64 1:2.6-12.el7 will be updated
---> Package wpa_supplicant.x86_64 1:2.6-12.el7_9.2 will be an update
---> Package zlib.x86_64 0:1.2.7-18.el7 will be updated
---> Package zlib.x86_64 0:1.2.7-20.el7_9 will be an update
```

Disable SELinux

1. Use command **nano /etc/selinux/config**
    a. If nano is not installed use the command **yum install nano**
2. Change **SELINUX=enforcing** to **SELINUX=disabled**
3. Exit and save

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of three values:
#     targeted - Targeted processes are protected,
#     minimum - Modification of targeted policy. Only selected processes are protected.
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

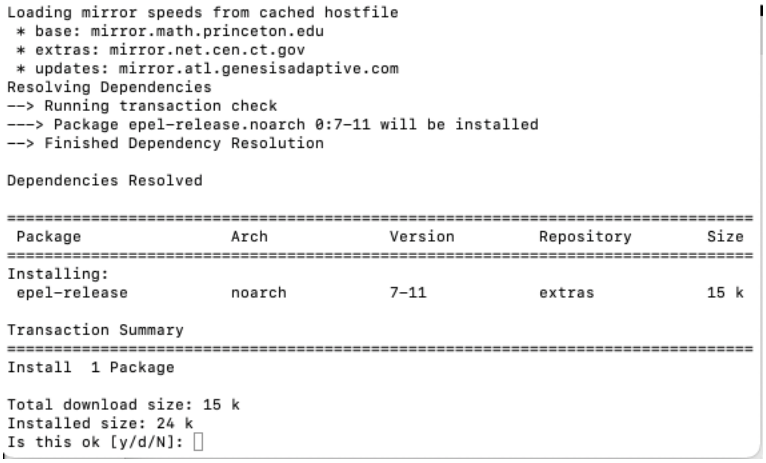
Disable Firewall

1. Use the command **systemctl stop firewalld**
2. Use the command **systemctl disable firewalld**

```
[root@nginx ~]# sudo systemctl stop firewalld
[root@nginx ~]# sudo systemctl disable firewalld
Removed symlink /etc/systemd/system/multi-user.target.wants/firewalld.service.
Removed symlink /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service.
[root@nginx ~]#
```

Install EPEL Packages

1. Use the command **yum install epel-release -y**



```
Loading mirror speeds from cached hostfile
 * base: mirror.math.princeton.edu
 * extras: mirror.net.cen.ct.gov
 * updates: mirror.atl.genesisadaptive.com
Resolving Dependencies
--> Running transaction check
---> Package epel-release.noarch 0:7-11 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package                Arch             Version          Repository        Size
================================================================================
Installing:
 epel-release           noarch           7-11             extras            15 k

Transaction Summary
================================================================================
Install  1 Package

Total download size: 15 k
Installed size: 24 k
Is this ok [y/d/N]:
```

Install NginX

1. Use the command **nano /etc/yum.repos.d/nginx.repo**
    a. Use command **yum install nano** if you do not have it
2. Type

```
[nginx]
name=nginx  repo
baseurl=https://nginx.org/packages/centos/$releasever/$basearch/
gpgcheck=0
enabled=1
```

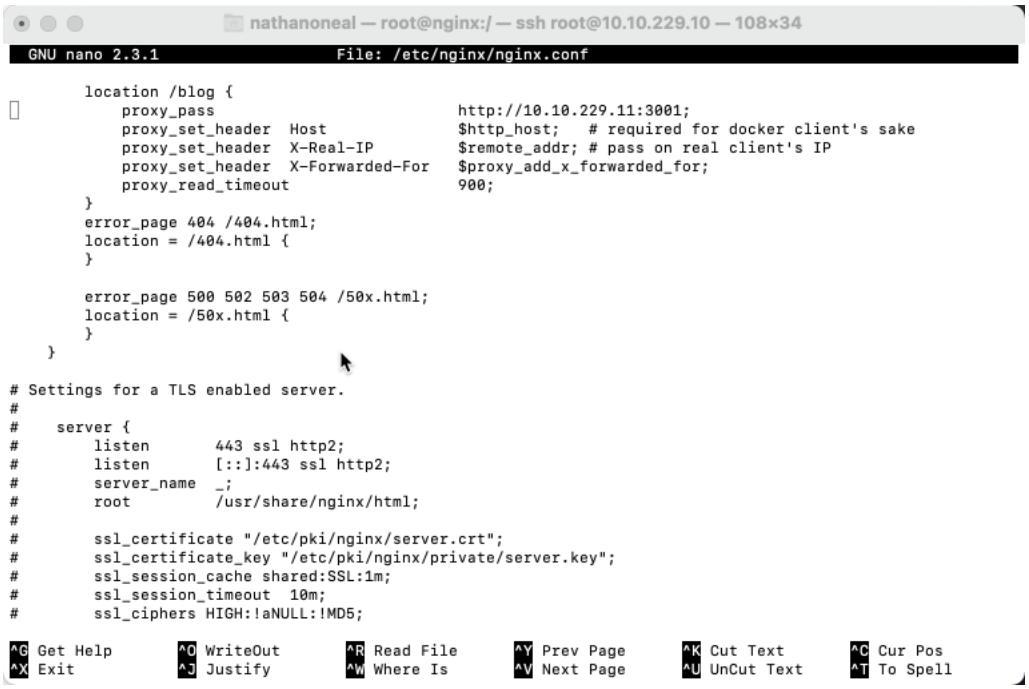
3. Use command **yum install nginx**

```
[root@nginx ~]# nano /etc/yum.repos.d/nginx.repo
[root@nginx ~]# sudo yum install nginx
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: mirror.cs.vt.edu
 * epel: packages.oit.ncsu.edu
 * extras: repos.hou.layerhost.com
 * updates: mirror.us-midwest-1.nexcess.net
nginx                                                                    | 2.9 kB  00:00:00
nginx/x86_64/primary_db                                                  | 235 kB  00:00:02
Resolving Dependencies
--> Running transaction check
---> Package nginx.x86_64 1:1.21.6-1.el7.ngx will be installed
--> Processing Dependency: libpcre2-8.so.0()(64bit) for package: 1:nginx-1.21.6-1.el7.ngx.x86_64
--> Running transaction check
---> Package pcre2.x86_64 0:10.23-2.el7 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================================================
 Package               Arch                  Version                          Repository              Size
================================================================================================================
Installing:
 nginx                 x86_64                1:1.21.6-1.el7.ngx               nginx                  796 k
Installing for dependencies:
 pcre2                 x86_64                10.23-2.el7                      base                   201 k

Transaction Summary
================================================================================================================
Install  1 Package (+1 Dependent package)

Total download size: 998 k
Installed size: 3.3 M
Is this ok [y/d/N]:
```

Start and Enable NginX

1. Use the command **/etc/nginx/nginx.conf**
2. Type

```
location /blog {
  proxy_pass                          http://10.10.229.11:3001;
  proxy_set_header  Host              $http_host;   # required
for docker client's sake
  proxy_set_header  X-Real-IP         $remote_addr; # pass on
real client's IP
  proxy_set_header  X-Forwarded-For
$proxy_add_x_forwarded_for;
  proxy_read_timeout                  900;
}
```



3. Use the command **systemctl start nginx**
4. Use the command **systemctl enable nginx**

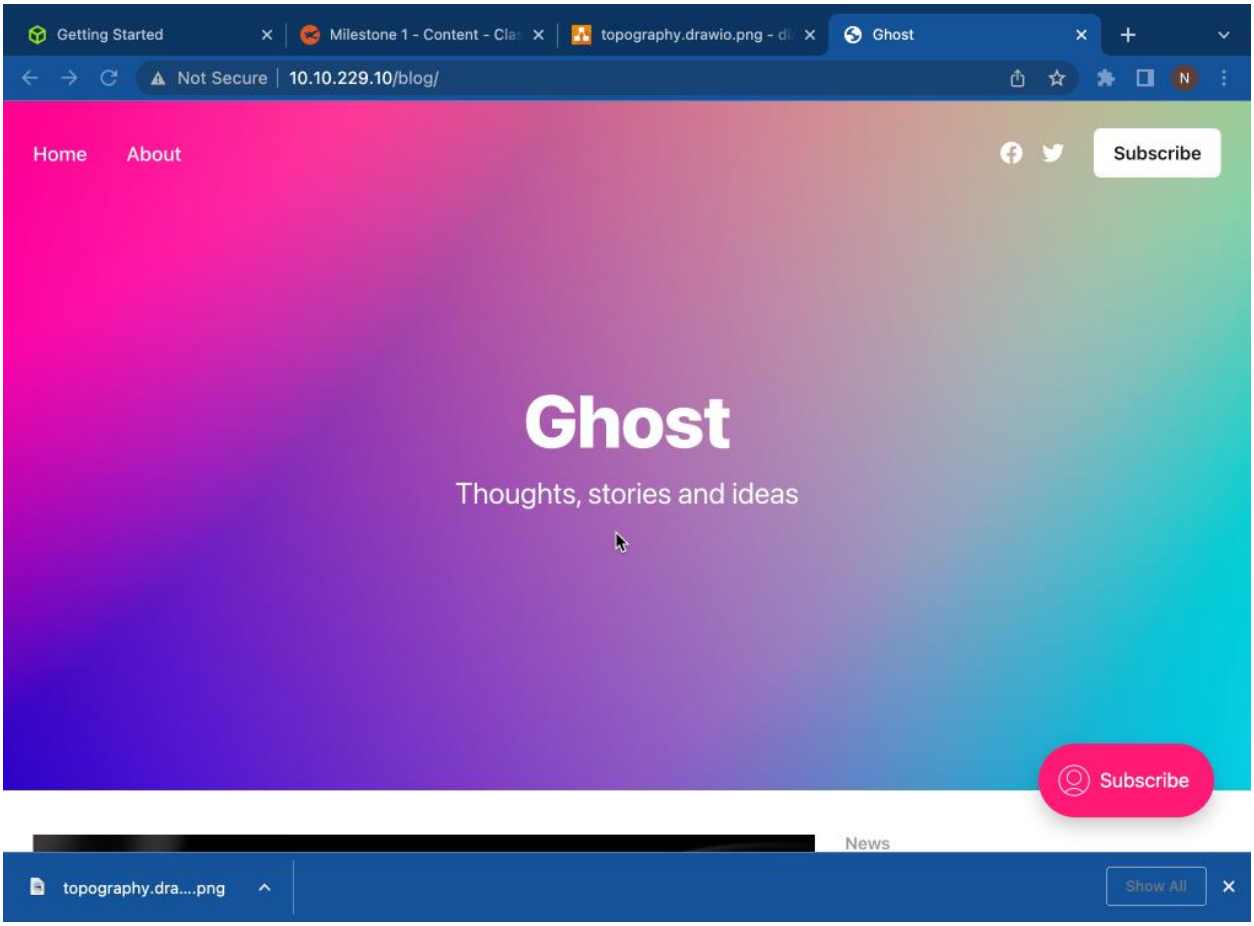
```
[root@nginx ~]# systemctl start nginx
[root@nginx ~]# systemctl enable nginx
Created symlink from /etc/systemd/system/multi-user.target.wants/nginx.service to /usr/lib/systemd/system/nginx.service.
[root@nginx ~]#
```

Reverse Proxy to Ghost Site

1. On the **docker VM** use the command **docker rm *containerID***
2. Use the command **docker run -d --name ghost -p 3001:2368 -e url=http://10.10.229.10/blog ghost**

```
[root@docker ~]# docker rm 8ee5d78fdd2f
8ee5d78fdd2f
[root@docker ~]# docker run -d --name ghost -p 3001:2368 -e url=http://10.10.229.10/blog ghost
0a5dbfe12ba024ebb97a417e2ad7850280697ea4593161613f8e7b860ca24787
[root@docker ~]#
```

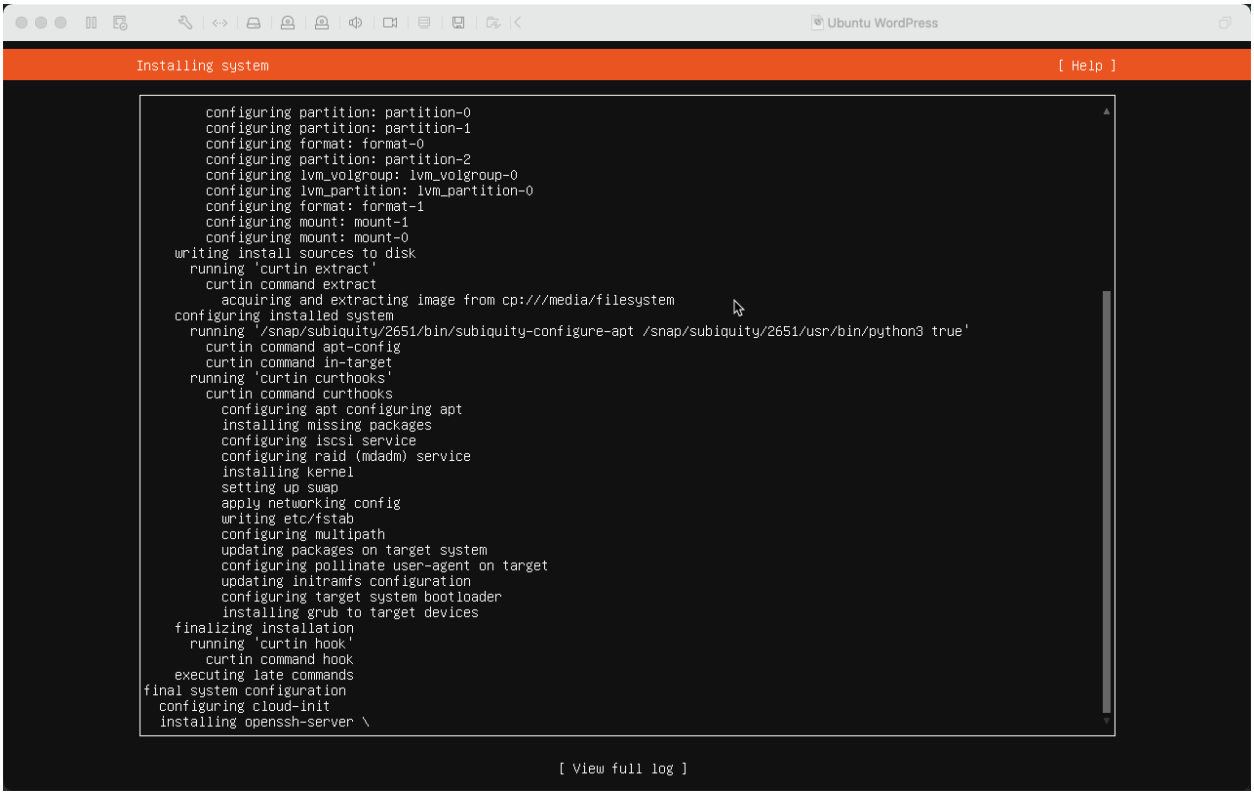
3. Go to **http://10.10.229.10/blog**


Getting Started
Milestone 1 - Content - Cla
topography.drawio.png - di
Ghost
Not Secure
10.10.229.10/blog/
Home
About
Subscribe
Ghost
Thoughts, stories and ideas
Subscribe
News
topography.dra....png
Show All

Install WordPress on Ubuntu - LAMP Stack

Base Ubuntu 18.04 Install

1. Set network custom network (10.10.229.1/24)
2. Set processors to **2** with **1024 MB** of memory
3. To start, choose your preferred language
4. You can press **Continue without updating the installer**
5. Select your keyboard layout



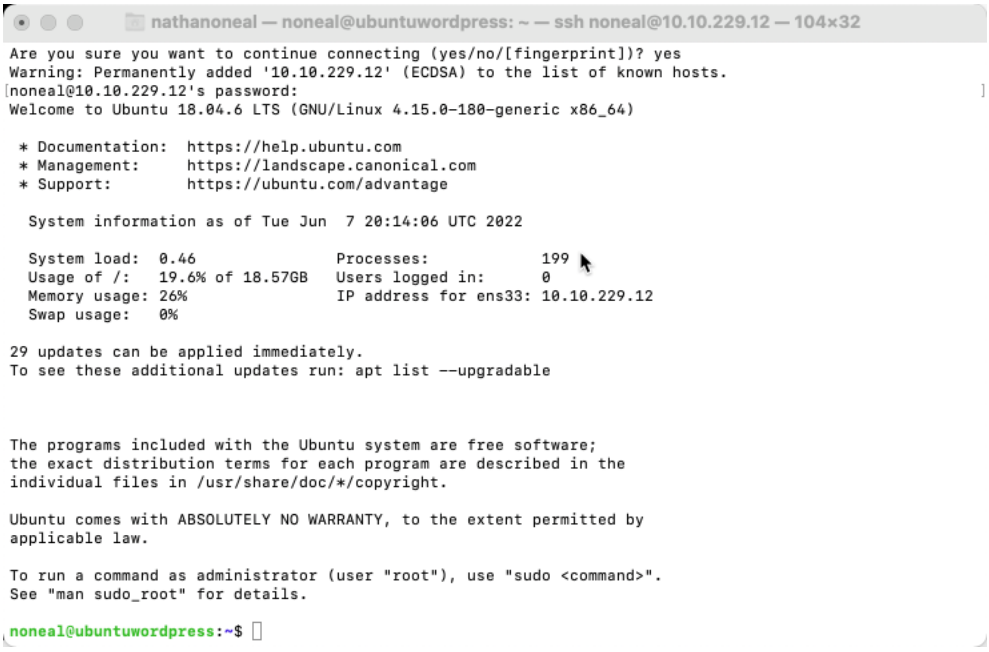
Set Static IP

1. To set up the network configuration
    a. Subnet- 10.10.229.0/24
    b. 10.10.229.12
    c. 10.10.229.2
    d. 10.10.229.2
2. Select **Save** and then click **Done**
3. Click **Done** on proxy address screen
4. Click **Done** on the next screen
5. Make sure **Use an entire disk** option is selected on filesystem page
6. Select **local disk** option on next page
7. On the next page select **Done** then click **continue**

8. Fill in the fields and click **done**
9. Click on **Install OpenSSH** then select **done**
10. Select **done** on the next screen
11. Once the VM reboots then you can sign in through SSH

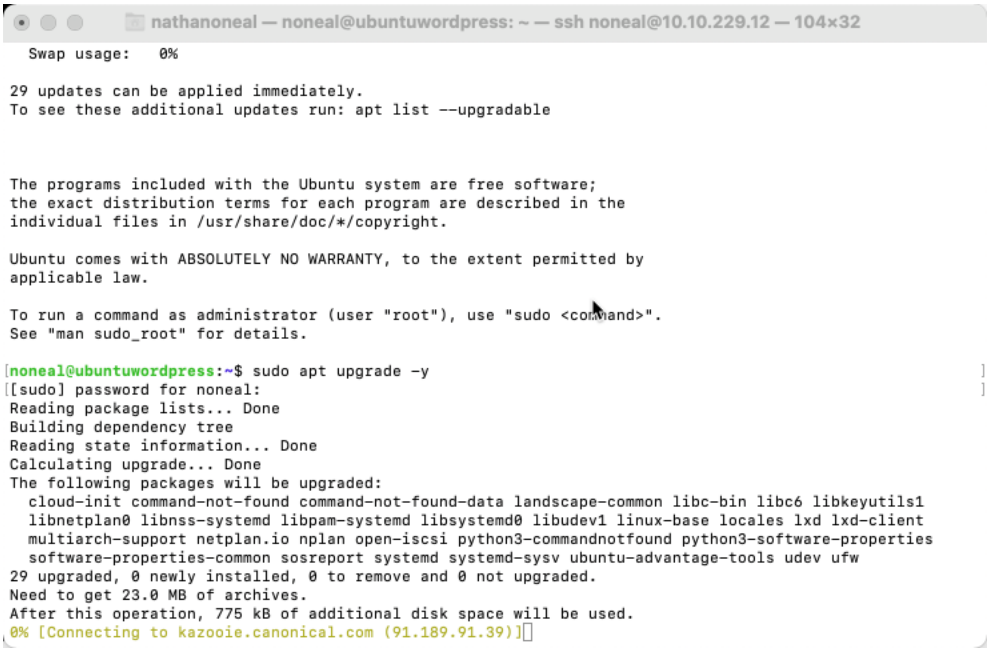
SSH into Ubuntu VM

1. Open the terminal and use the command **ssh *(user)*@10.10.229.12**
2. Type **Yes** when asked if you want to continue connecting
3. Type in your password



Update Ubuntu

1. Use the command **sudo apt upgrade -y**



Install and Configure Apache

1. Use the command **sudo apt install apache2 -y**
2. Use the Command **sudo ufw allow in "Apache Full"**
3. You can got to http://10.10.229.12 to check if the web server installed correctly

```
noneal@ubuntuwordpress:~$ sudo su
root@ubuntuwordpress:/home/noneal# apt install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap
  liblua5.2-0 ssl-cert
Suggested packages:
  www-browser apache2-doc apache2-suexec-pristine | apache2-suexec-custom openssl-blacklist
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3
  libaprutil1-ldap liblua5.2-0 ssl-cert
0 upgraded, 10 newly installed, 0 to remove and 0 not upgraded.
Need to get 1,730 kB of archives.
After this operation, 6,997 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Install and Configure MySQL

1. Use the command **sudo apt install mysql-server -y**
2. Use the command **sudo mysql**
3. At mysql> use **ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY '*(yourpassword*';**
4. Use the command **FLUSH PRIVILEGES;** then use **exit**
5.

```
root@ubuntuwordpress:/home/noneal# sudo apt install mysql-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libaio1 libcgi-fast-perl libcgi-pm-perl libencode-locale-perl libevent-core-2.1-6 libfcgi-perl
  libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl libhttp-date-perl libhttp-message-perl
  libio-html-perl liblwp-mediatypes-perl libtimedate-perl liburi-perl mysql-client-5.7
  mysql-client-core-5.7 mysql-common mysql-server-5.7 mysql-server-core-5.7
Suggested packages:
  libdata-dump-perl libipc-sharedcache-perl libwww-perl mailx tinyca
The following NEW packages will be installed:
  libaio1 libcgi-fast-perl libcgi-pm-perl libencode-locale-perl libevent-core-2.1-6 libfcgi-perl
  libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl libhttp-date-perl libhttp-message-perl
  libio-html-perl liblwp-mediatypes-perl libtimedate-perl liburi-perl mysql-client-5.7
  mysql-client-core-5.7 mysql-common mysql-server mysql-server-5.7 mysql-server-core-5.7
0 upgraded, 21 newly installed, 0 to remove and 0 not upgraded.
Need to get 19.6 MB of archives.
After this operation, 156 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

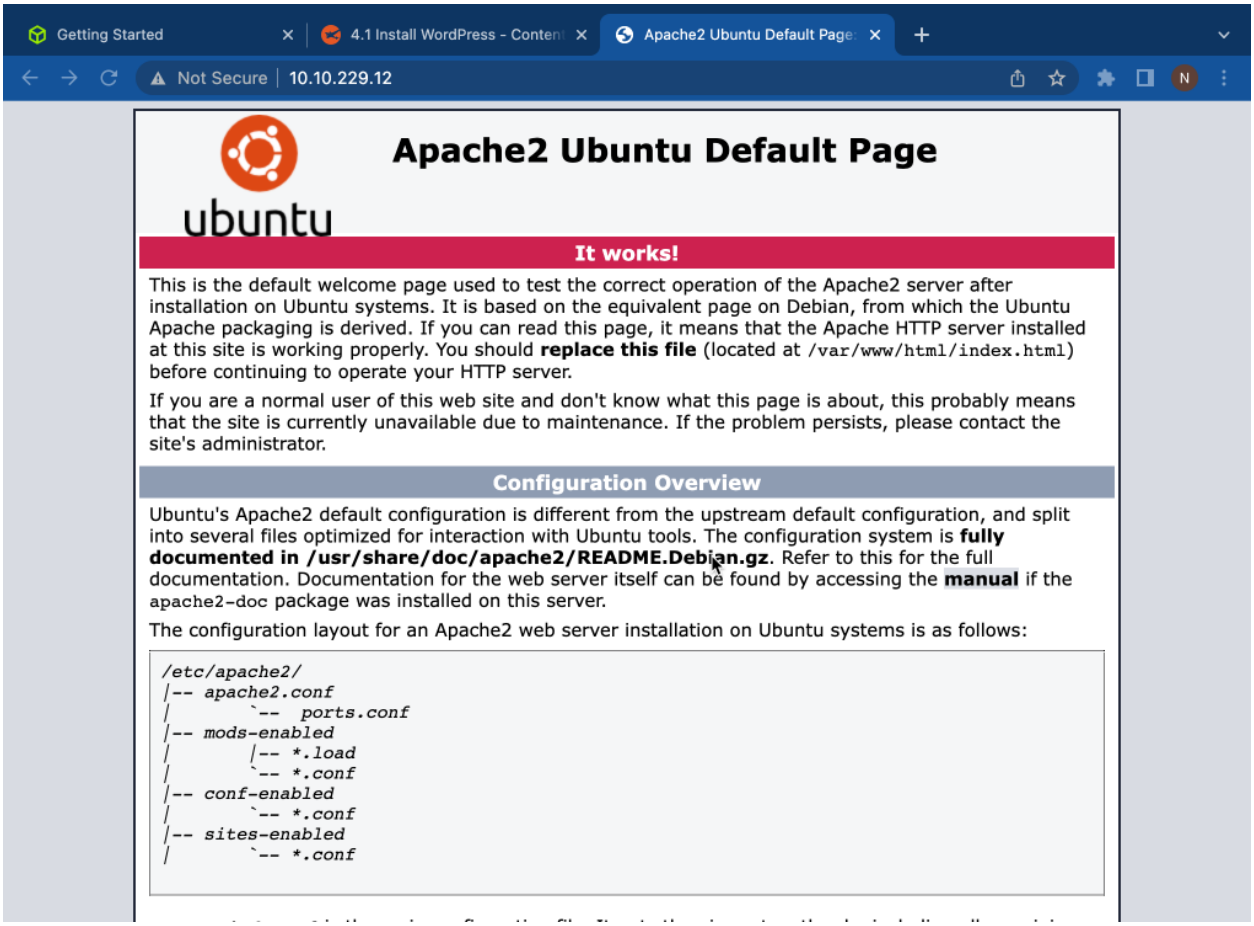
Install and configure PHP

1. Use the command **sudo nano /etc/apt/sources.list** and add **universe** to the end of all lines
2. Save and exit and use the command **sudo apt update -y**
3. Use the command **sudo apt install php libapache2-mod-php php-mysql**
4. Use the command **sudo apt install php-curl php-gd php-xml php-mbstring php-xmlrpc php-zip php-soap php-intl**
5. Use the command **sudo a2enmod rewrite**
6. Use the command **sudo apache2 restart**

```
root@ubuntuwordpress:/home/noneal# sudo apt install php libapache2-mod-php php-mysql -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libapache2-mod-php7.2 libsodium23 php-common php7.2 php7.2-cli php7.2-common php7.2-json
  php7.2-mysql php7.2-opcache php7.2-readline
Suggested packages:
  php-pear
The following NEW packages will be installed:
  libapache2-mod-php libapache2-mod-php7.2 libsodium23 php php-common php-mysql php7.2 php7.2-cli
  php7.2-common php7.2-json php7.2-mysql php7.2-opcache php7.2-readline
0 upgraded, 13 newly installed, 0 to remove and 0 not upgraded.
Need to get 4,137 kB of archives.
After this operation, 18.0 MB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu bionic/main amd64 php-common all 1:60ubuntu1 [12.1 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu bionic-updates/main amd64 php7.2-common amd64 7.2.24-0ubuntu0.
18.04.11 [890 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu bionic-updates/main amd64 php7.2-json amd64 7.2.24-0ubuntu0.18
.04.11 [18.9 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu bionic-updates/main amd64 php7.2-opcache amd64 7.2.24-0ubuntu0
.18.04.11 [165 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu bionic-updates/main amd64 php7.2-readline amd64 7.2.24-0ubuntu
0.18.04.11 [12.2 kB]
Get:6 http://us.archive.ubuntu.com/ubuntu bionic/main amd64 libsodium23 amd64 1.0.16-2 [143 kB]
Get:7 http://us.archive.ubuntu.com/ubuntu bionic-updates/main amd64 php7.2-cli amd64 7.2.24-0ubuntu0.18.
04.11 [1,411 kB]
```

Test PHP

1. To test that PHP installed correctly use the command **sudo nano /var/www/html/test.php** and insert **<?php phpinfo(); ?>**
2. save and exit then go to http://10.10.229.12/test.php
3. if you see the PHP test page you can continue



Database Configuration in MySQL

1. Use the command **mysql -u root -p**
2. Use the command **CREATE DATABASE WordPressDB DEFAULT CHARACTER SET utf8 COLLATE utf8_unicode_ci;**
3. To add a user use the command **GRANT ALL ON WordPressDB.* TO '*User*'@'localhost' IDENTIFIED BY '*password*';**
4. Then use the command **flush privileges;** then use **exit**

Install WordPress

Clone WordPress

1. Use the command **sudo rm /var/www/html/***
2. Use the command **sudo git clone https://github.com/WordPress/WordPress /var/www/html/**
3. Use the command **sudo chown -R www-data:www-data /var/www/html/***
4. Use the command **sudo chown www-data:www-data /var/www/html**

```
root@wordpress:/home/noneal# sudo git clone https://github.com/WordPress/WordPre
ss /var/www/html/
Cloning into '/var/www/html'...
remote: Enumerating objects: 372559, done.
remote: Counting objects: 100% (253/253), done.
remote: Compressing objects: 100% (145/145), done.
remote: Total 372559 (delta 144), reused 186 (delta 108), pack-reused 372306
Receiving objects: 100% (372559/372559), 343.31 MiB | 6.72 MiB/s, done.
Resolving deltas: 100% (301067/301067), done.
```

Edit Ownership

1. Use the command **sudo chown -R www-data:www-data /var/www/html/***
2. Use the command **sudo chown www-data:www-data /var/www/html**
3. Use the command **sudo nano /etcapache2/apache2.conf**
4. Go to the section **<Directory /var/www/>** and change **AllowOverride** to **ALL**

```
noneal@wordpress:~$ sudo chown -R www-data:www-data /var/www/html/*
[sudo] password for noneal:
noneal@wordpress:~$ sudo chown www-data:www-data /var/www/html
```

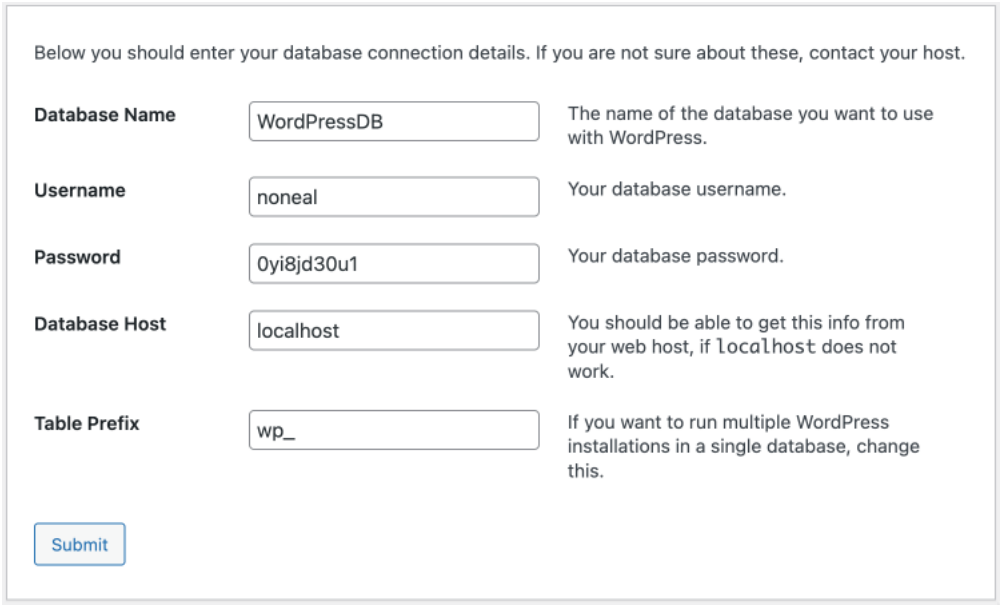
Edit .htaccess

1. Use the command **sudo nano /var/www/html/.git/htaccess**
2. Insert
    a. **Order deny, allow**
    deny from all
3. Use the command **sudo sytemctl restart apache2**

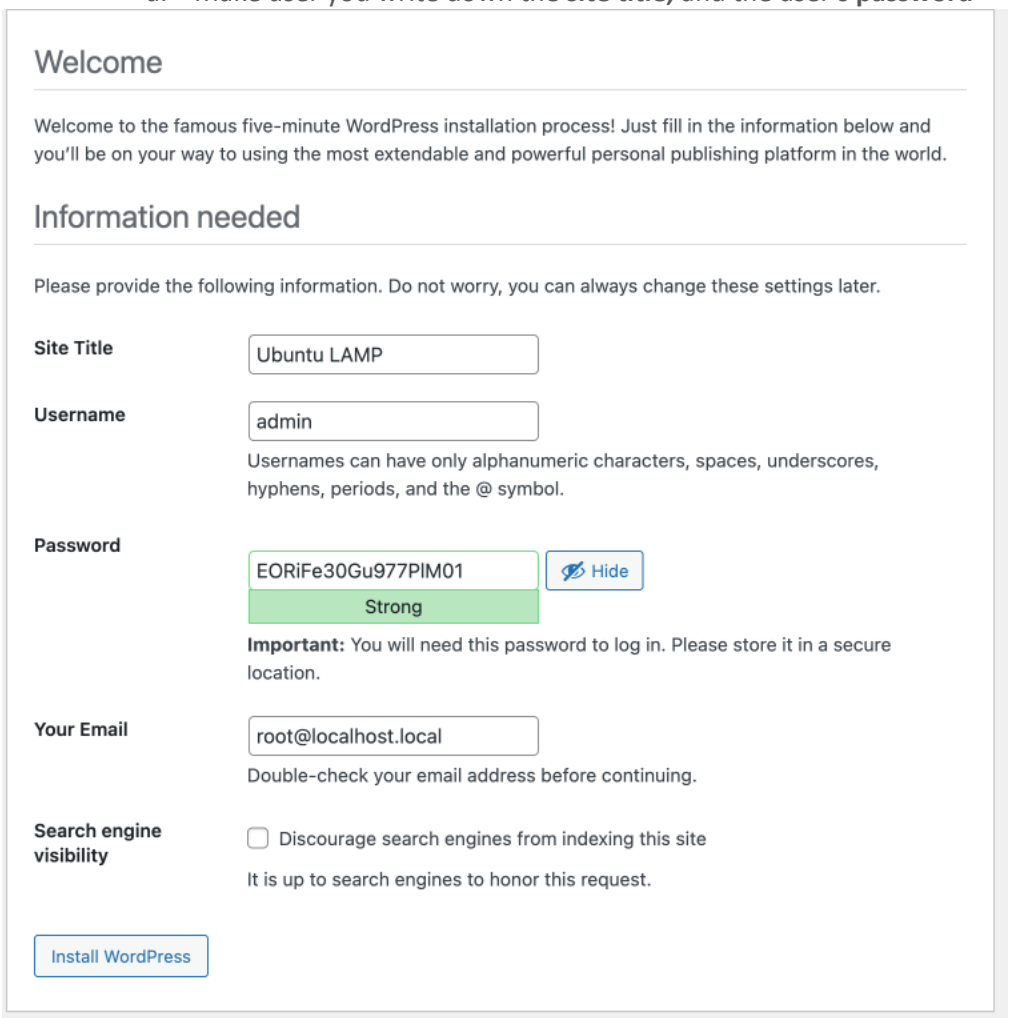
WordPress Configuration

WordPress Configuration Process

1. Go to http://10.10.229.12 where you should see the Wordpress install
2. Use the following:
    a. Database Name -**WordPressDB**
    b. Username – your database username
    c. Password – your database password
    d. Database Host -**localhost**



3. Fill in the prompts and click **Install WordPress**
    a. Make user you write down the **site title,** and the user's **password**



Test WordPress

WordPress Security Settings and Configurations

Security Summary

For hardening WordPress I will be going over file permissions, changing file permissions, securing wp-config.php file, and installing and configuiring a firewall (In this case I will be using Shield). For file permission I simply reviewed the files that were installed with WordPress. When changing file permissions I set the permissions where only the owner can eidt the file or directory while the owner can only read and execute or simply read. For securing wp-config.php I moved the file to one directory level above the html directory where the WordPress files and directories reside. The other thing I did was deny anyone to find the file when using something like FTP. Lastly when setting up a firewall I block different things such as SQL or PHP injections as well as making users have a vaild email addresses and strong passwords.

Defense-in-depth

Defense-in-depth itself means that when approching cybersecurity you want many defense mechanisms that are layed overtop one anouther. This way if one defense fails there is anouther one to back it up. No one can be protected from a single layer of defense and by adding more layers you can help close other security gaps. For a small example, later I am going to show you how to edit the .htaccess file so that the wp-config.php file is harder to find if you do not have access to the main server. If this fails I also moved the to a higher directory so an attacker will not be able to simply be able to use a command like *cd /var/www/html* then use *ls* to find the file. Of course moving it to anouther directory is not going to stop an attacker for long so I changed the file permissions so even if they find the file unless they have the correct permissions they will not be able to edit the file. These layers reduce the overall likelyhood that an attacker can do damage to our server.

File Permissions

Vulnerability

File permissions specify who is allowed to read, write, or execute different files or directories. Improperly configured permissions can allow people who should not have access to a certain file the ability to read or even edit that file.

Configuration

To change the permissions of a certain file you would use the command *sudo cmd 'xxx' /filepath/* xxx refers to the permissions that you want to be set. As an example, you could use *sudo cmd 644 /home/noneal/testfile.txt*

Validation

Steps for validation will be shown in the "Changing File Permission" section.

Changing File Permission

Vulnerability

When I first set up the WordPress server the */var/www/html* directory was set to give the Apache user permissions. This needed to be changed to give additional permissions as well as restrict some. What I

want to do is change the permissions so that for files the owner can read, write, and execute while the group and others can only read and execute. For directories I want to make it where the owner can read and write while the group and others can only read. This change will only allow the owner of the file to edit it which would make it harder for an attacker to possibly deny access to the server.

Configuration

To change the permissions all the directories and files that are in the *html* directory I entered the following commands:

For directories- ***find /var/www/html/* -type d -exec chmod 750 {} \;***

For files- ***find /var/www/html/* -type f -exec chmod 644 {} \;***

```
noneal@wordpress:/var/www/html$ sudo find /var/www/html/* -type d -exec chmod 755 {} \;
noneal@wordpress:/var/www/html$ sudo find /var/www/html/* -type f -exec chmod 644 {} \;
noneal@wordpress:/var/www/html$ ls -la
total 228
drwxr-xr-x  6 www-data www-data  4096 Jun 20 22:30 .
drwxr-xr-x  3 root     root      4096 Jun 20 22:26 ..
drwxr-xr-x  8 root     root      4096 Jun 10 21:41 .git
-rw-r--r--  1 www-data www-data   586 Jun 19 15:27 .htaccess
-rw-r--r--  1 www-data www-data   405 Jun 10 21:32 index.php
-rw-r--r--  1 www-data www-data 19915 Jun 10 21:32 license.txt
-rw-r--r--  1 www-data www-data  7401 Jun 10 21:32 readme.html
-rw-r--r--  1 www-data www-data  7165 Jun 10 21:32 wp-activate.php
drwxr-xr-x  9 www-data www-data  4096 Jun 10 21:32 wp-admin
-rw-r--r--  1 www-data www-data   351 Jun 10 21:32 wp-blog-header.php
-rw-r--r--  1 www-data www-data  2338 Jun 10 21:32 wp-comments-post.php
-rw-r--r--  1 www-data www-data  3001 Jun 10 21:32 wp-config-sample.php
drwxr-xr-x  7 www-data www-data  4096 Jun 21 00:05 wp-content
-rw-r--r--  1 www-data www-data  3919 Jun 10 21:32 wp-cron.php
drwxr-xr-x 26 www-data www-data 12288 Jun 10 21:32 wp-includes
-rw-r--r--  1 www-data www-data  2494 Jun 10 21:32 wp-links-opml.php
-rw-r--r--  1 www-data www-data  3973 Jun 10 21:32 wp-load.php
-rw-r--r--  1 www-data www-data 48499 Jun 10 21:32 wp-login.php
-rw-r--r--  1 www-data www-data  8577 Jun 10 21:32 wp-mail.php
-rw-r--r--  1 www-data www-data 23706 Jun 10 21:32 wp-settings.php
-rw-r--r--  1 www-data www-data 32051 Jun 10 21:32 wp-signup.php
-rw-r--r--  1 www-data www-data  4748 Jun 10 21:32 wp-trackback.php
-rw-r--r--  1 www-data www-data  3236 Jun 10 21:32 xmlrpc.php
```

Validation

After configuring the permissions the way I wanted, I simply tried to use *cd* to change the directory to see if my *noneal* account had permission to read the directories that it should not.

```
noneal@wordpress:/var/www/html$ cd wp-admin
-bash: cd: wp-admin: Permission denied
noneal@wordpress:/var/www/html$ cd wp-content
-bash: cd: wp-content: Permission denied
noneal@wordpress:/var/www/html$ cd wp-includes
```

This shows that the *noneal* account was unable to do so.

Securing wp-config.php

Vulnerability

The wp-config.php file holds important information such as the username and password for the database. If accessed by someone with malicious intent, they gain information that could deny WordPress's access to the database. This would make the site itself useless.

Configuration

To secure the wp-config.php file the first thing I wanted to do is move it one directory level up. To do this I used the command:

sudo mv /var/www/html/wp-config.php /var/www/wp-config.php

```
noneal@wordpress:/var/www$ sudo mv /var/www/html/wp-config.php /var/www/wp-config.php
noneal@wordpress:/var/www$ ls
html  wp-config.php
```

The second thing I wanted to do was change the permission of the file so that only the owner could read it. While I was in the same directory as wp-config.php I used the command:

sudo chmod 400 /var/www/wp-config.php

```
noneal@wordpress:/var/www$ sudo chmod 400 /var/www/wp-config.php
noneal@wordpress:/var/www$ ls -l
total 8
drwxr-xr-x 6 www-data www-data 4096 Jun 21 01:14 html
-r-------- 1 www-data www-data 3283 Jun 10 21:56 wp-config.php
```

The last thing I wanted to do was edit the .htaccess file to change the accessibility of the wp-config.php file. To do this I used ***sudo nano .htaccess*** while in the same directory as the .htaccess file. After I was in the file, I added the following lines to the bottom of the text:

<files wp-config.php>

order allow,deny

deny from all

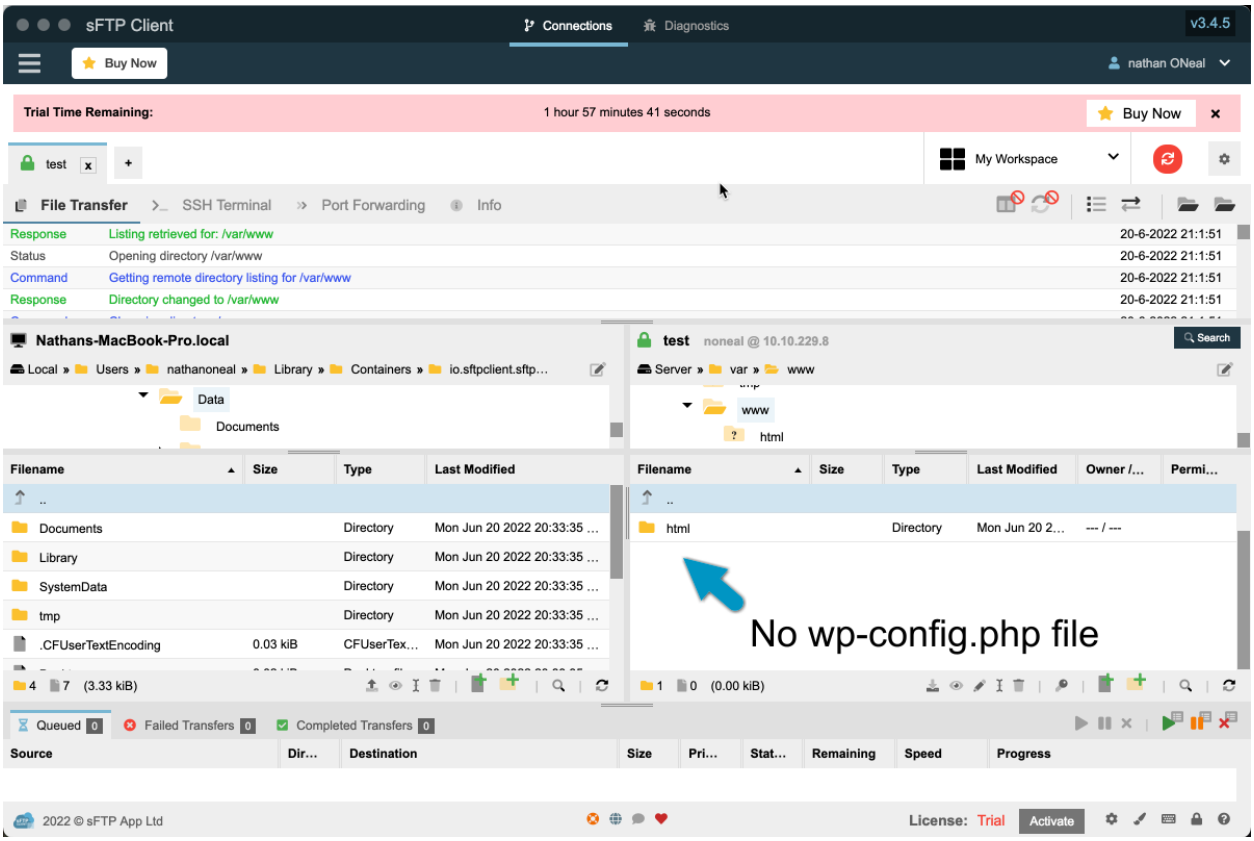
</files>

```
# BEGIN WordPress
# The directives (lines) between "BEGIN WordPress" and "END WordPress" are
# dynamically generated, and should only be modified via WordPress filters.
# Any changes to the directives between these markers will be overwritten.
<IfModule mod_rewrite.c>
RewriteEngine On
RewriteRule .* - [E=HTTP_AUTHORIZATION:%{HTTP:Authorization}]
RewriteBase /
RewriteRule ^index\.php$ - [L]
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule . /index.php [L]
</IfModule>

# END WordPress
```

```
# BEGIN WordPress
# The directives (lines) between "BEGIN WordPress" and "END WordPress" are
# dynamically generated, and should only be modified via WordPress filters.
# Any changes to the directives between these markers will be overwritten.
<IfModule mod_rewrite.c>
RewriteEngine On
RewriteRule .* - [E=HTTP_AUTHORIZATION:%{HTTP:Authorization}]
RewriteBase /
RewriteRule ^index\.php$ - [L]
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule . /index.php [L]
</IfModule>

<files wp-config.php>
order allow,deny
deny from all
</files>
# END WordPress
```

Validation

To validate that I was able to secure the wp-config.php file I downloaded an SFTP application from the MacBook app store. After this I connected to the WordPress server and navigated to the www directory. There was no file in the directory just the html directory.



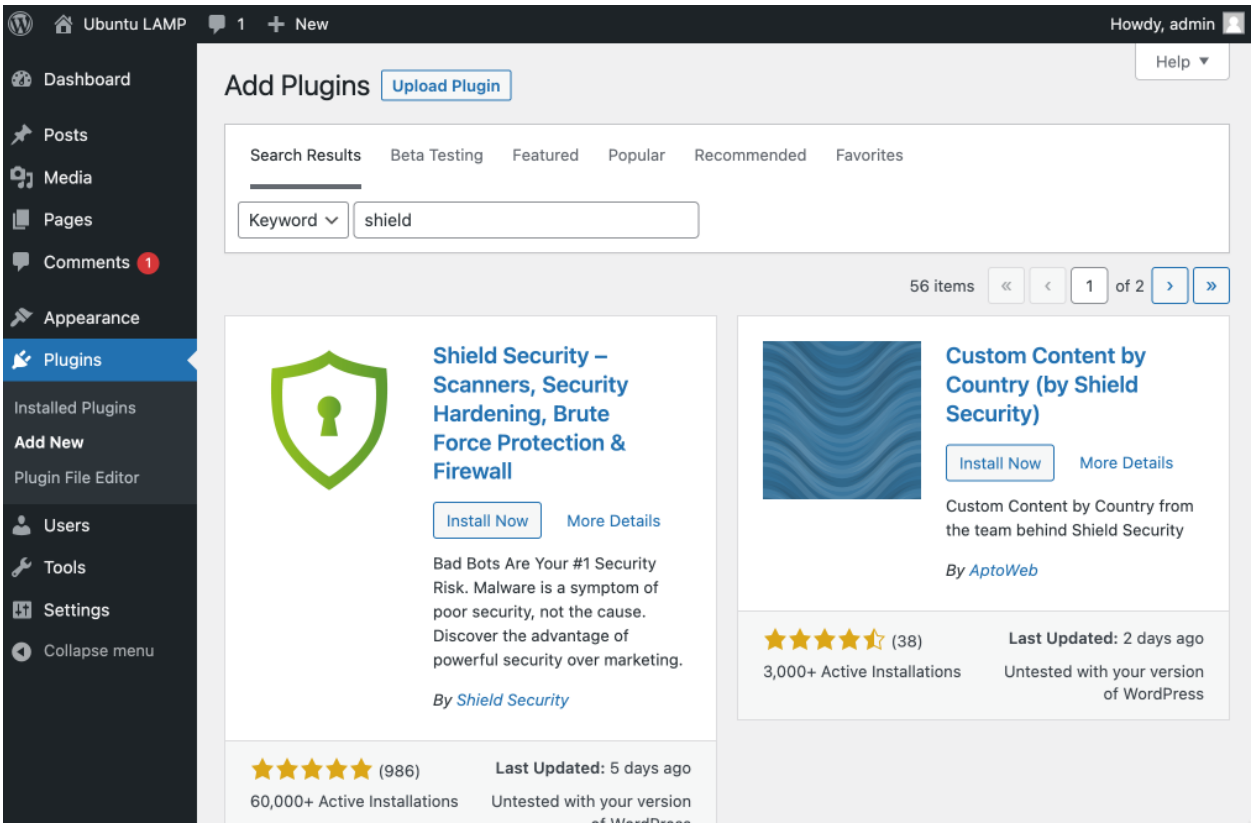
Setting up a Firewall

Vulnerability

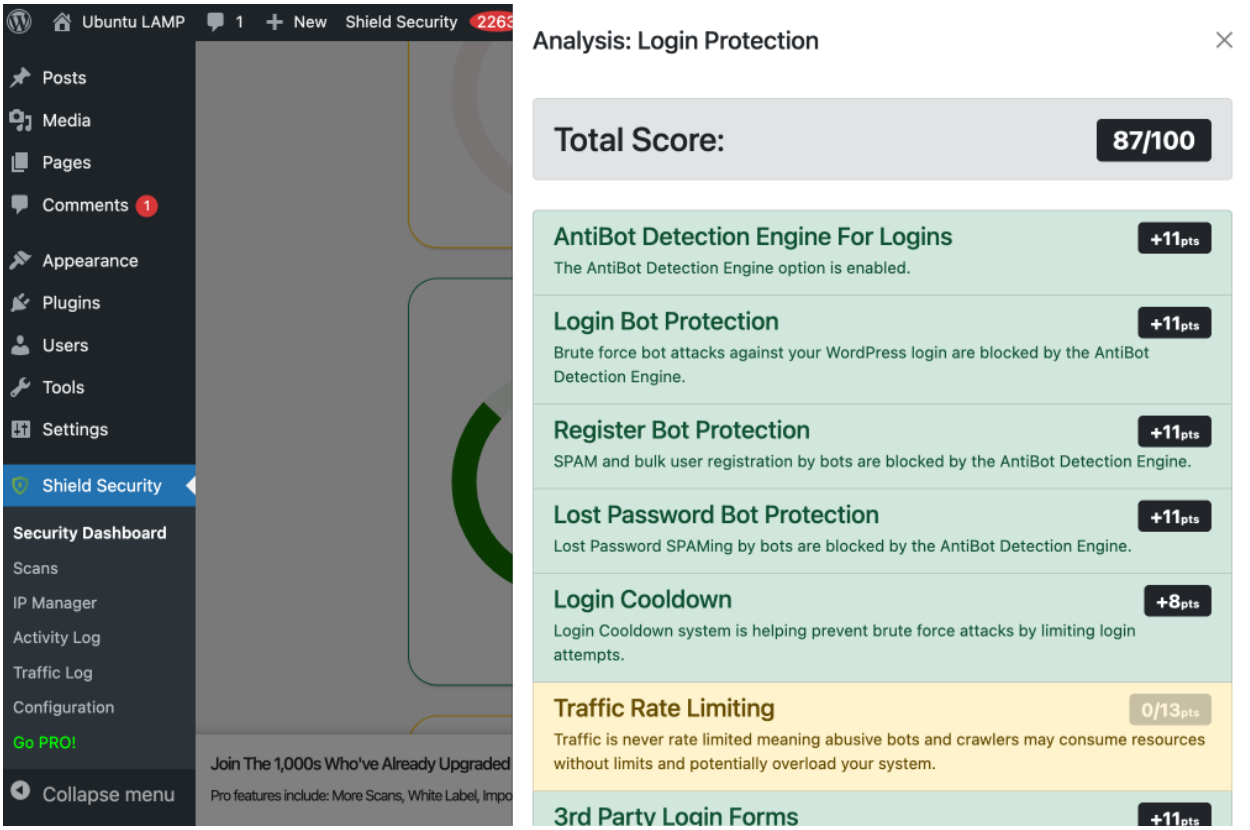
Having a firewall can protect from any malicious or unnecessary network traffic. This can be either protecting against Denial-of-Service Attacks (DDoS), brute forcing, or code injection. The vanilla install of WordPress does not have anything to help with this which is why I had to install a plugin. The plugin I used was the Shield Security plugin. This plugin provides vulnerability scans, traffic logs, and settings for user registration to name a few things. If there is anything suspicious Shield will log the activity for me to look at later and will give me the ability to black or whitelist the IP if need be.

Configuration

Configuration was a simple process. All I had to do was go to my Ubuntu LAMP site and login as the admin user. After this I went to the plugin tab and searched Shield. Once it popped up, I clicked install.

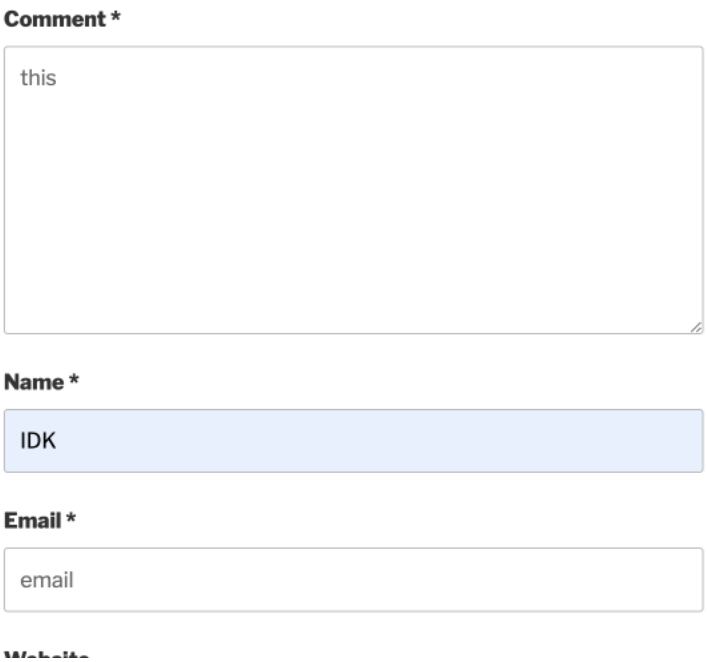


After the install I checked the security dashboard for the security overview which showed how the secure the site was by giving a grade rating. I went through each category and added the setting that I would provide security without harming the site.

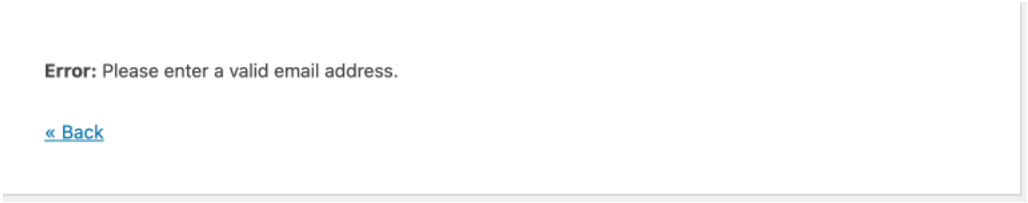


Validation

One of the settings I wanted to test was the email validation. To test this email validation I went to the "Leave a Reply" section on one of the posts and try post a comment with the email "email".



When pressing post I was given the message "Error: Please enter a valid email address."



WordPress Security Conclusion

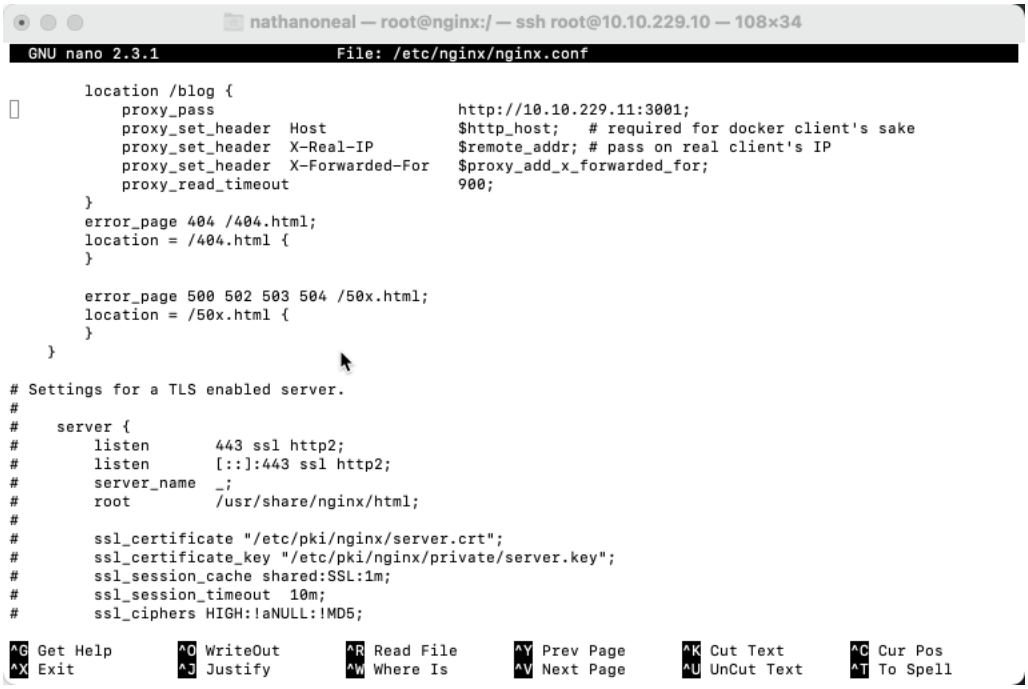
This taught me the importance and some of the techniques of hardening your systems against attack. To do this I made sure that the files for WordPress was secure by editing the file and directory permissions so that it is much harder to be able to edit or even read important files. When it came to even more important files like wp-config.php I moved the file to another directory, changed the permissons even more so that the owner was the only one who could read the file, and edit the .htaccess file so that the accessability is harder without being on the main server. Lastly I installed a firewall so that I could be able to log network and login activity, set user registration requirement, and prevent attacks from bots.

Full Report Conclusion

Throughout this project I was able to successfully set up three servers that were running different programs. The first one was a CentOS server that was running Docker Ghost with the IP 10.10.229.11. This was done simply by setting up CentOS, installing EPEL packages, and then installing Docker CE. After this I had to set up Docker making sure it ran on startup then I installed and setup Ghost. The next server what was set up was another CentOS server that was running Nginx with the IP 10.10.229.10. To do this I set up CentOS, installed EPEL packages, and then installed Nginx. After this I had to set up Nginx making sure it runs on startup and making sure traffic that is goes to the Docker Ghost machine gets pushed through the proxy server. Lastly, I had to make a WordPress server using Ubuntu server with the IP address 10.10.229.12. To do this I had to set up Ubuntu, create and set up a LAMP stack, then install and set up WordPress. This led into setting up security measures to help protect my WordPress site. To do this I had to edit privileges to all the WordPress files, edit the .htcaccess file so that the wp-config.php file was hidden by people who are not on the main server, and add and set up a firewall to WordPress site.

Appendix A

NginX Config File



Appendix B

NginX Access Log File

```
[root@nginx nginx]# tail access.log
10.10.229.11 - - [06/Jun/2022:19:02:59 -0400] "GET /blog/favicon.ico HTTP/1.1" 200 15406 "-" "Mozilla/5.0 Safari/537.36" "-"
10.10.229.1 - - [06/Jun/2022:19:02:59 -0400] "GET /blog/ HTTP/1.1" 200 3401 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/102.0.5005.61 Safari/537.36" "-"
10.10.229.1 - - [06/Jun/2022:19:02:59 -0400] "GET /blog/assets/built/casper.js?v=a44b848c49 HTTP/1.1" 200 1230 "http://10.10.229.10/blog/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/102.0.5005.61 Safari/537.36" "-"
10.10.229.1 - - [06/Jun/2022:19:02:59 -0400] "GET /blog/assets/built/screen.css?v=a44b848c49 HTTP/1.1" 200 7252 "http://10.10.229.10/blog/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/102.0.5005.61 Safari/537.36" "-"
10.10.229.1 - - [06/Jun/2022:19:02:59 -0400] "GET /blog/public/cards.min.css?v=a44b848c49 HTTP/1.1" 200 4735 "http://10.10.229.10/blog/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/102.0.5005.61 Safari/537.36" "-"
10.10.229.1 - - [06/Jun/2022:19:02:59 -0400] "GET /blog/public/cards.min.js?v=a44b848c49 HTTP/1.1" 200 1759 "http://10.10.229.10/blog/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/102.0.5005.61 Safari/537.36" "-"
10.10.229.1 - - [06/Jun/2022:19:03:00 -0400] "GET /blog/members/api/member/ HTTP/1.1" 204 0 "http://10.10.229.10/blog/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/102.0.5005.61 Safari/537.36" "-"
10.10.229.1 - - [06/Jun/2022:19:03:01 -0400] "GET /blog/ghost/api/content/settings/?key=4ba491547586973aa7424a85ca&limit=all HTTP/1.1" 200 1014 "http://10.10.229.10/blog/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/102.0.5005.61 Safari/537.36" "-"
10.10.229.1 - - [06/Jun/2022:19:03:01 -0400] "GET /blog/ghost/api/content/tiers/?key=4ba491547586973aa7424a85ca&limit=all&include=monthly_price,yearly_price,benefits HTTP/1.1" 200 622 "http://10.10.229.10/blog/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/102.0.5005.61 Safari/537.36" "-"
10.10.229.1 - - [06/Jun/2022:19:03:01 -0400] "GET /blog/ghost/api/content/newsletters/?key=4ba491547586973aa7424a85ca&limit=all HTTP/1.1" 200 408 "http://10.10.229.10/blog/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/102.0.5005.61 Safari/537.36" "-"
[root@nginx nginx]#
```