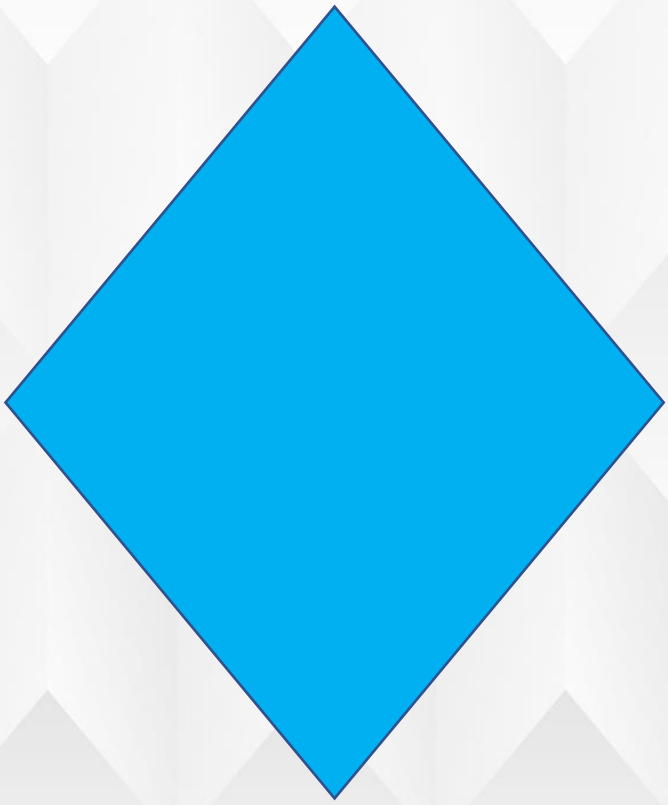


Diamond Supply Co.

By Kyle Ford

Table of Contents

Inventory

EQUIPMENT	OPERATIING SYSTEM	ADDITIONAL INFO	IP ADDRESS
Router/Custom Network			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

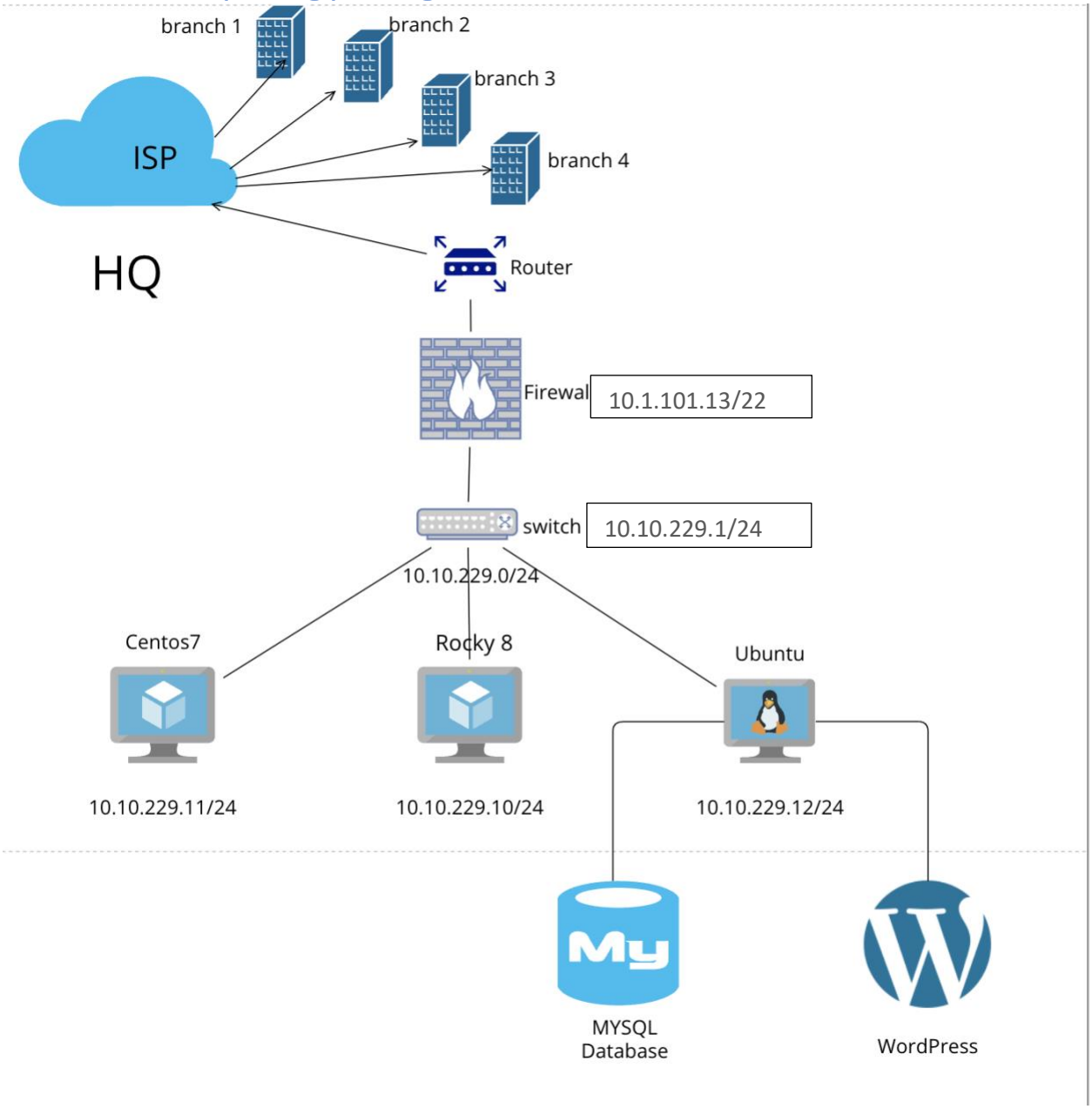
Custom Network

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	[redacted]	[redacted]
CentOS Network User	[redacted]	[redacted]
CentOS Host User	[redacted]	[redacted]
MySQL Network User	[redacted]	[redacted]
MySQL Host User	[redacted]	[redacted]
WordPress Admin	[redacted]	[redacted]

Network Topology Diagram



Node.js Application (Ghost) on Docker

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

1. Begin by downloading the CentOS 7 ISO file from the official website. You can choose between a minimal or DVD ISO depending on your specific needs.
2. Next, launch your preferred virtualization software. For example, if you are utilizing Oracle VirtualBox, click on the "New" button to create a new virtual machine.
3. In the "Create Virtual Machine" wizard, assign a name to your virtual machine and select "Linux" as the type, and "Red Hat (64-bit)" as the version.
4. Allocate the appropriate amount of memory and number of CPUs to the virtual machine, taking into consideration your system resources and the anticipated workload.
5. Create a new virtual hard disk or use an existing one if available. Assign the appropriate amount of disk space to accommodate your requirements.
6. In the "Storage" settings, attach the CentOS 7 ISO file to the virtual CD/DVD drive.
7. Start the virtual machine and proceed with the CentOS 7 installation wizard. Select your preferred language, time zone, and keyboard layout.
8. Select "Installation Destination" and choose the disk or partition where CentOS 7 will be installed. You may opt to utilize automatic partitioning as well.
9. Choose the packages you want to install, keeping in mind that a minimal installation necessitates selecting solely the "Base System" package group.
10. Configure the network settings, including the hostname, IP address, and DNS servers.
11. Establish the root password and create a regular user account.
12. Click on "Begin Installation" and wait for the installation to complete.
13. Once the installation is complete, reboot the virtual machine and log in as the user you created.

There you have it! By following these instructions, you'll now have a base CentOS 7 virtual machine installed and ready to utilize.

SSH into CentOS VM

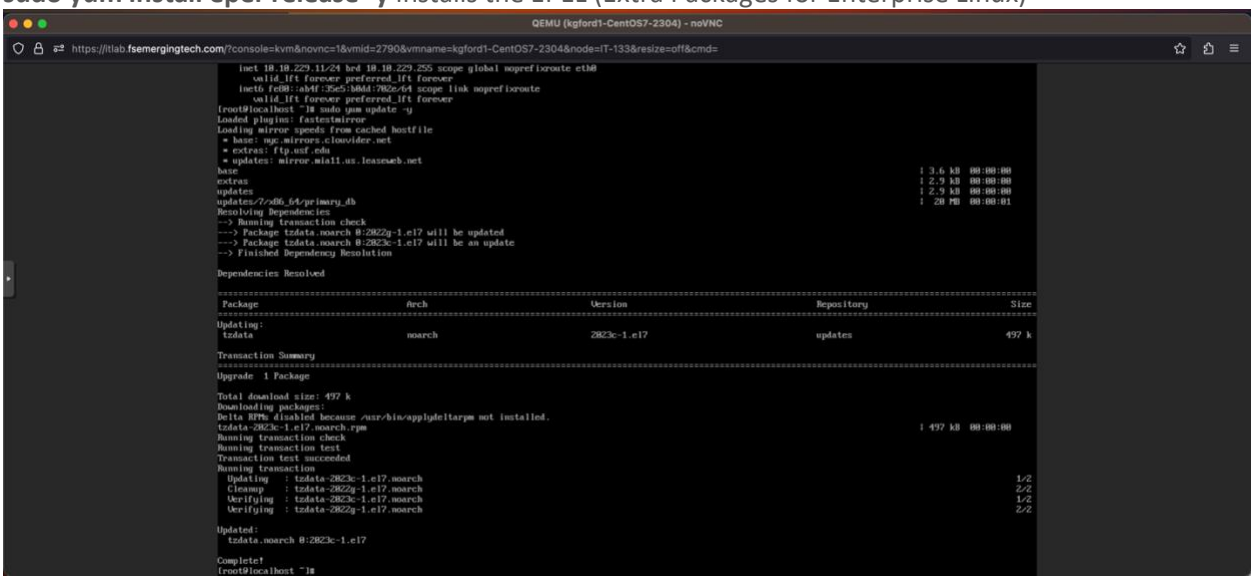
1. Begin by launching your CentOS VM in your preferred virtualization software.
2. Retrieve the IP address assigned to the CentOS VM by running the command **ip addr show** in the terminal within the VM.
3. Ensure that the SSH service is installed and running on the CentOS VM. Verify this by running the command **systemctl status sshd.service**.
4. On your local machine, launch a terminal and input the command **ssh username@ip_address**. Replace "username" with the username of the account on the CentOS VM that you wish to access, and replace "ip_address" with the IP address of the CentOS VM.
5. If you have not previously connected to that IP address, you will receive a prompt to add the IP address to the list of known hosts. Input "yes" to proceed.
6. You will then be prompted to enter the password for the username that you specified earlier. Input the password and press Enter.
7. You should now have successfully logged into the CentOS VM via SSH.

These steps will allow you to manage your CentOS VM using the terminal through SSH.

Update CentOS

- **sudo yum update -y** updates the CentOS 7 system with the latest updates available.
- **sudo yum install nano** installs the Nano text editor which helps edit configuration files.

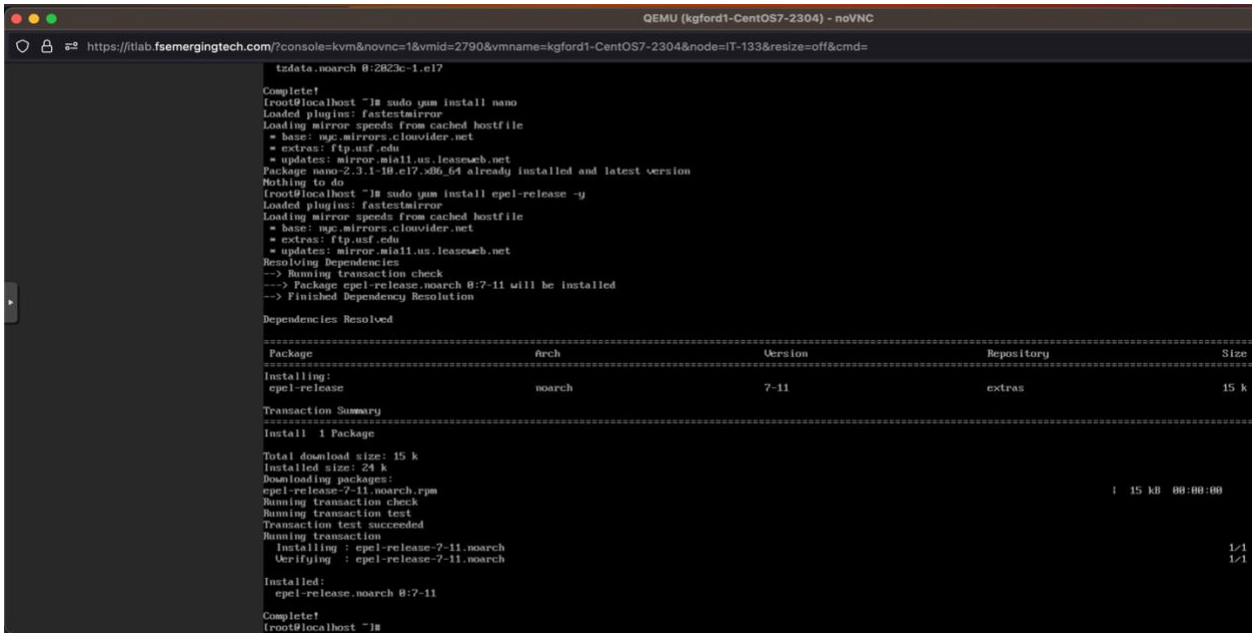
sudo yum install epel-release -y installs the EPEL (Extra Packages for Enterprise Linux)



Install EPEL Packages

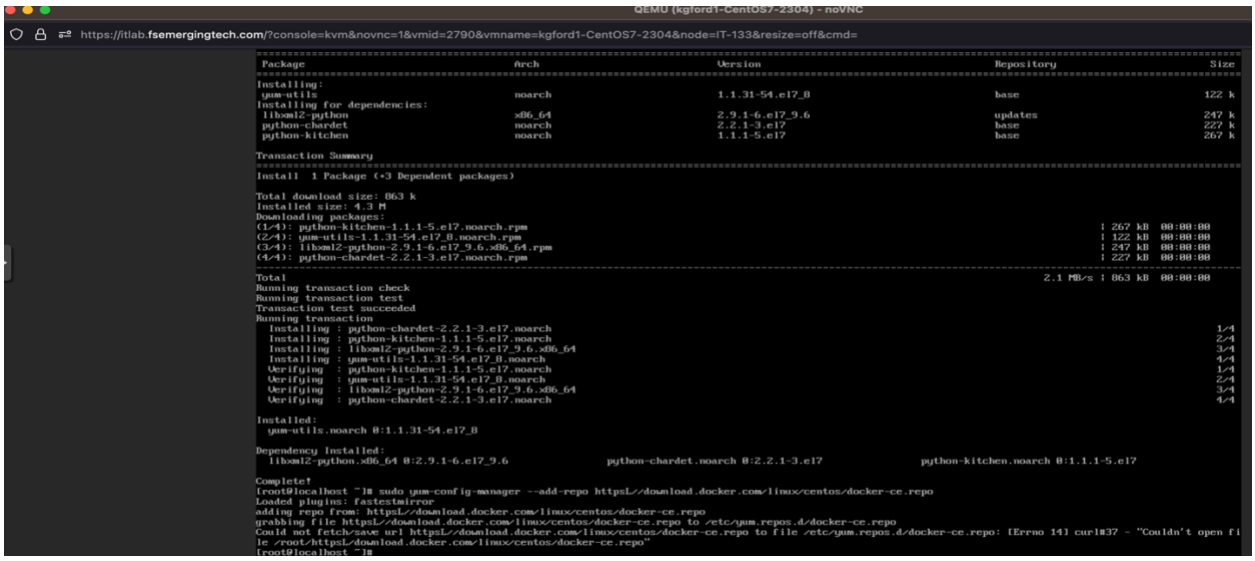
- **sudo yum install epel-release -y** installs the EPEL (Extra Packages for Enterprise Linux) repository which has additional software packages that are not available in the default CentOS repository.

- **yum install -y yum-utils** installs the yum utilities package which provides utilities for managing software packages with yum.

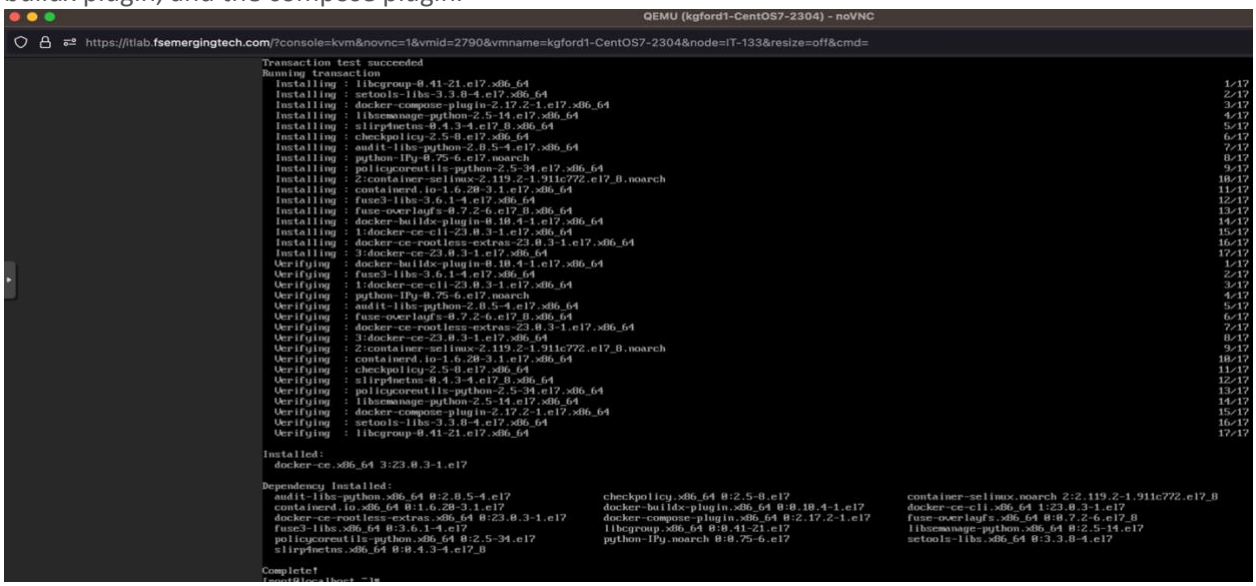


Install Docker CE

- **yum-config-manager --add-repo https://download.docker.com/linux/centos/dockerce.repo** adds the Docker repository to the available list of repositories, allowing the system to download and install the Docker CE engine.

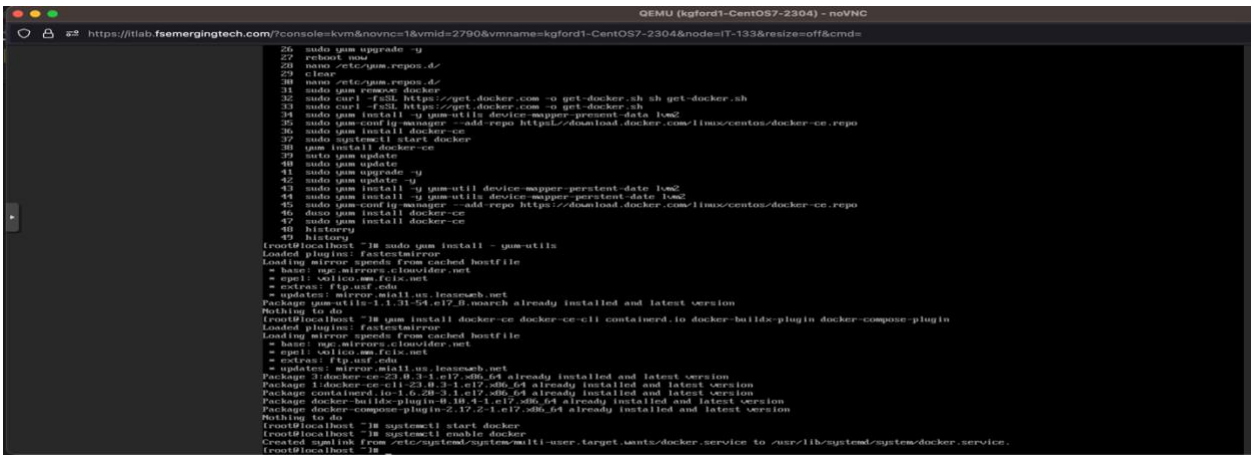


- **yum install docker-ce docker-ce-cli containerd.io docker-buildx-plugin dockercompose-plugin** installs the Docker CE engine and additional features such as the Docker CLI, containerd.io, the buildx plugin, and the compose plugin.



Start and Enable Docker

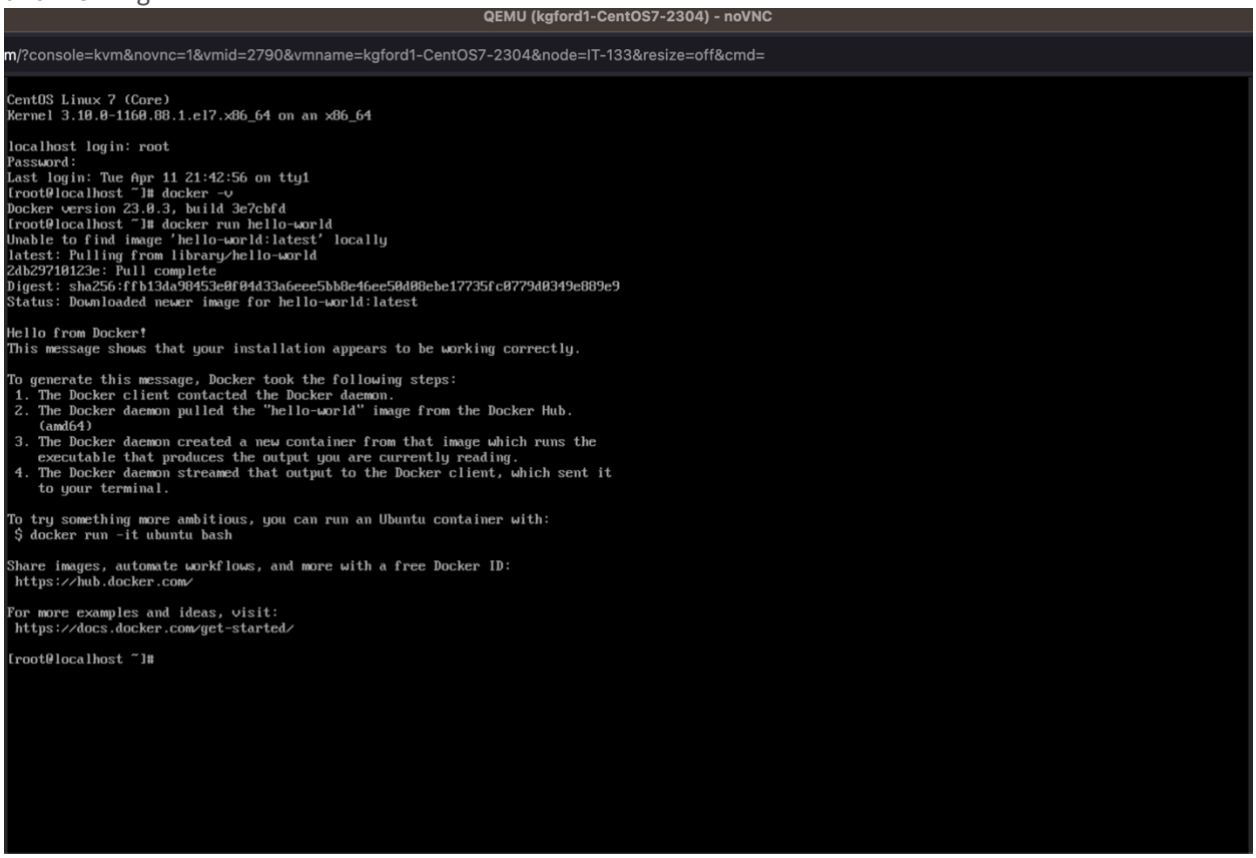
- **systemctl start docker** starts the Docker service.
- **systemctl enable docker** enables the Docker service to start automatically during system boot.



Test Docker (hello-world)

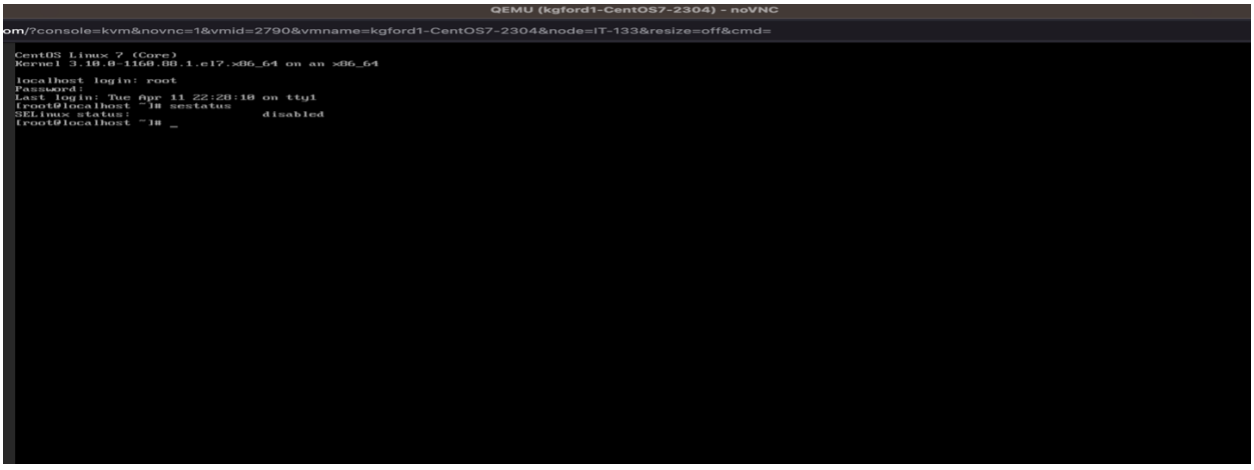
- **docker -v** displays the Docker version currently installed on the system.

- **docker run hello-world** runs the "hello-world" container to ensure Docker is correctly installed and working.



Disable SELinux on CentOS 7 Virtual Machine

- **nano /etc/selinux/config** opens the SELinux configuration file in the Nano text editor.
- Change the text "Enforcing" option to "Disabled" in the configuration file.
- Save the changes and exit the Nano text editor.
- **sestatus** displays the current status of SELinux.

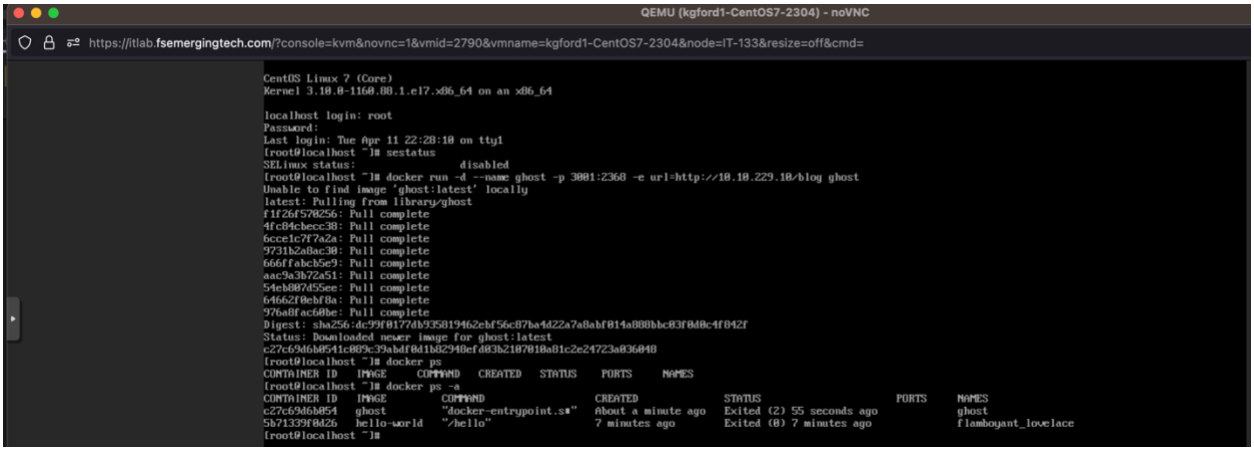


Install Ghost Docker Container

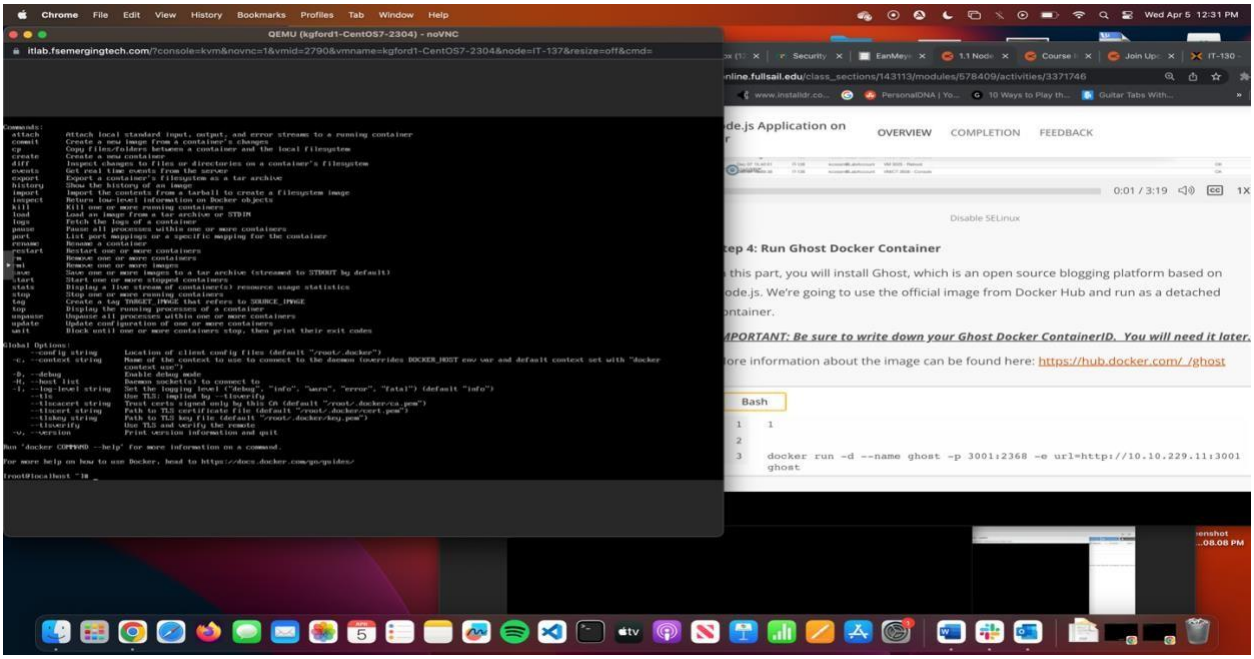
- **docker run -d --name ghost -p 3001:2368 -e url=http://10.10.229.10/blog ghost** runs the Ghost Docker container in detached mode with the name "ghost", mapping port 3001 on the host to port 2368 in the container, and setting the "url" environment variable to "http://10.10.229.10/blog".

- **docker ps -a** lists all the containers that are currently running or have run on the system.

C27c69d6b054



Test Ghost


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Wed Apr 5 12:31 PM
QEMU (kgford1-CentOS7-2304) - noVNC
itlab.fsemergingtech.com/?console=kvm&novnc=1&vmid=2790&vmname=kgford1-CentOS7-2304&node=IT-137&resize=off&cmd=
OVERVIEW
COMPLETION
FEEDBACK
0:01 / 3:19
Disable SELinux
ep 4: Run Ghost Docker Container
this part, you will install Ghost, which is an open source blogging platform based on
ode.js. We're going to use the official image from Docker Hub and run as a detached
ntainer.
MPORTANT: Be sure to write down your Ghost Docker ContainerID. You will need it later.
ore information about the image can be found here: https://hub.docker.com/_/ghost
Bash
1
2
3
docker run -d --name ghost -p 3001:2368 -e url=http://10.10.229.11:3001
ghost

NginX Reverse Proxy

Install base Rocky 8 Virtual Machine “ITE229-NginX (10)

1. Begin by downloading the Rocky Linux 8 ISO image from the official website. Ensure that you download the correct ISO image for your virtualization software.

2. Open your virtualization software and create a new virtual machine. Select "Linux" as the operating system and "Red Hat (64-bit)" as the version.

3. Assign a meaningful name to your virtual machine and allocate the necessary resources, such as memory and CPU cores. Ensure that the resources you allocate are sufficient for your intended use.

4. Create a new virtual hard disk with a size that is appropriate for your use case.

5. In the virtual machine settings, select the Rocky Linux 8 ISO image as the installation media.

6. Start the virtual machine and follow the installation wizard. Ensure that you select the appropriate language, time zone, and keyboard layout during the installation process.

7. Customize the installation settings as necessary, including partitioning, software selection, and network configuration.

8. Once the installation is complete, reboot the virtual machine and log in with the root account.

9. Update the system packages and install any necessary additional software.

10. Finally, ensure that you have taken all necessary security precautions, such as disabling unnecessary services and configuring firewalls.

By following these instructions in a professional manner, you can ensure that your base Rocky Linux 8 virtual machine is installed correctly and is ready for use.

SSH into Rocky 8 VM

1. Make sure that your Rocky 8 Linux virtual machine is running and is connected to the network.

2. Open a terminal or command prompt on your local machine.

3. Type the following command to connect to the virtual machine via SSH:

ssh [username]@[ip-address]

Replace **[username]** with the username that you want to use to log in to the Rocky 8 virtual machine, and replace **[ip-address]** with the IP address of the virtual machine.

4. If this is the first time you are connecting to the virtual machine from your local machine, you will be prompted to confirm the authenticity of the host by checking the fingerprint of the SSH key. Verify the fingerprint matches the one from the virtual machine's key and type "yes" to continue.

5. Enter the password for the username when prompted.

6. You should now be logged into the Rocky 8 virtual machine via SSH and can run commands in the remote shell.

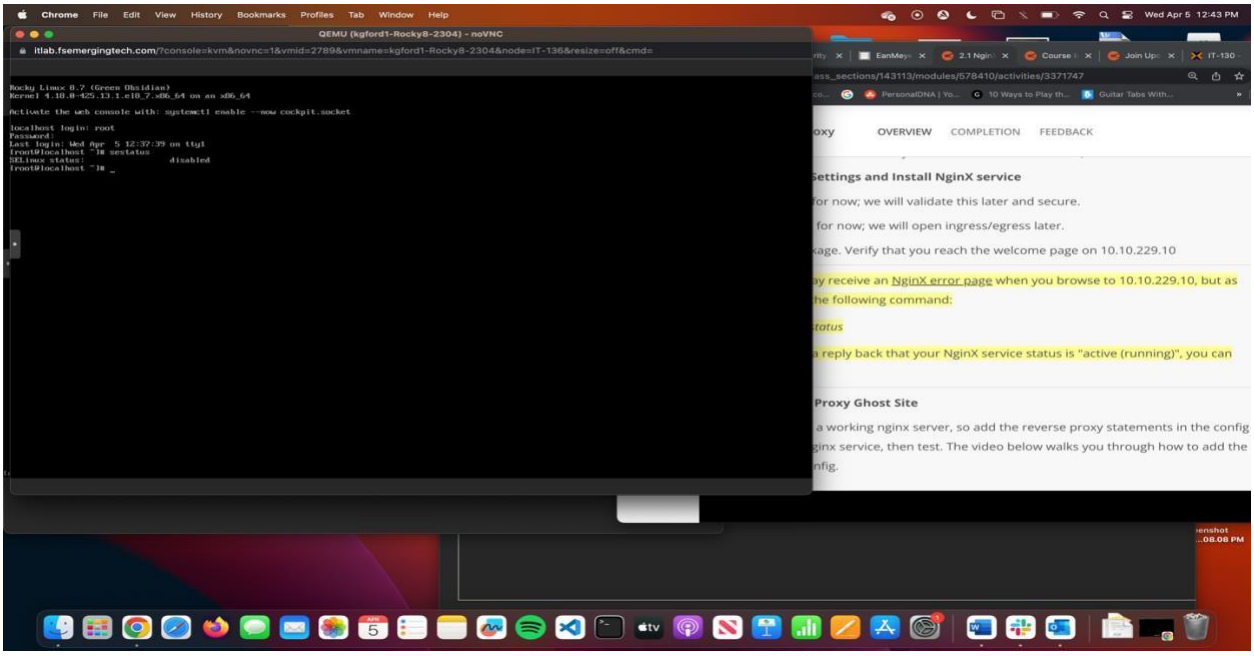
Update Rocky 8

- yum update -y: This command updates all packages on the system to their latest versions, and the -y flag automatically answers "yes" to any prompts or questions that may arise during the update process.
- yum install nano: This command installs the Nano text editor, which is a lightweight and user-friendly editor for editing configuration files on the system.

Disable SELinux

- nano /etc/selinux/config: This command opens the SELinux configuration file using the Nano text editor, which allows you to edit the configuration settings for SELinux.

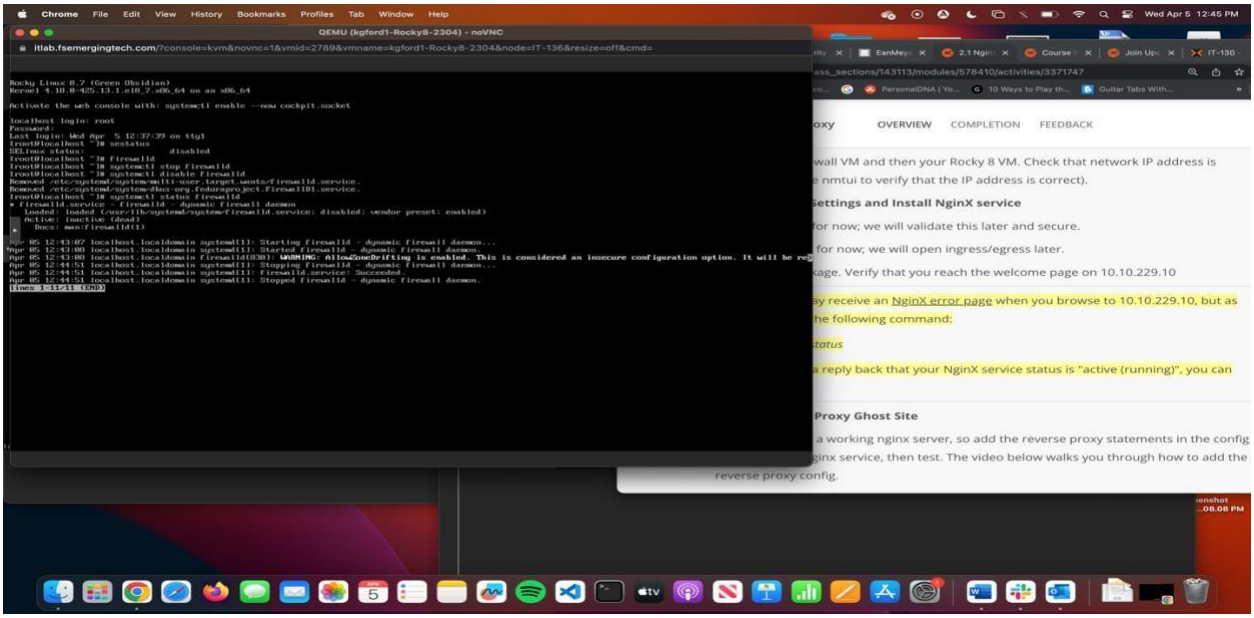
- Within the SELinux configuration file, change the "Enforcing" parameter to "Disabled", save the changes, and reboot the system. The command sestatus is used to verify that the changes took effect and that SELinux is disabled.



Disable Firewall

- systemctl status firewalld: This command checks the status of the firewalld service, which is a firewall management tool for CentOS/RHEL systems.

- systemctl stop firewalld and systemctl disable firewalld: These commands stop and disable the firewalld service, which temporarily disables the firewall on the system.

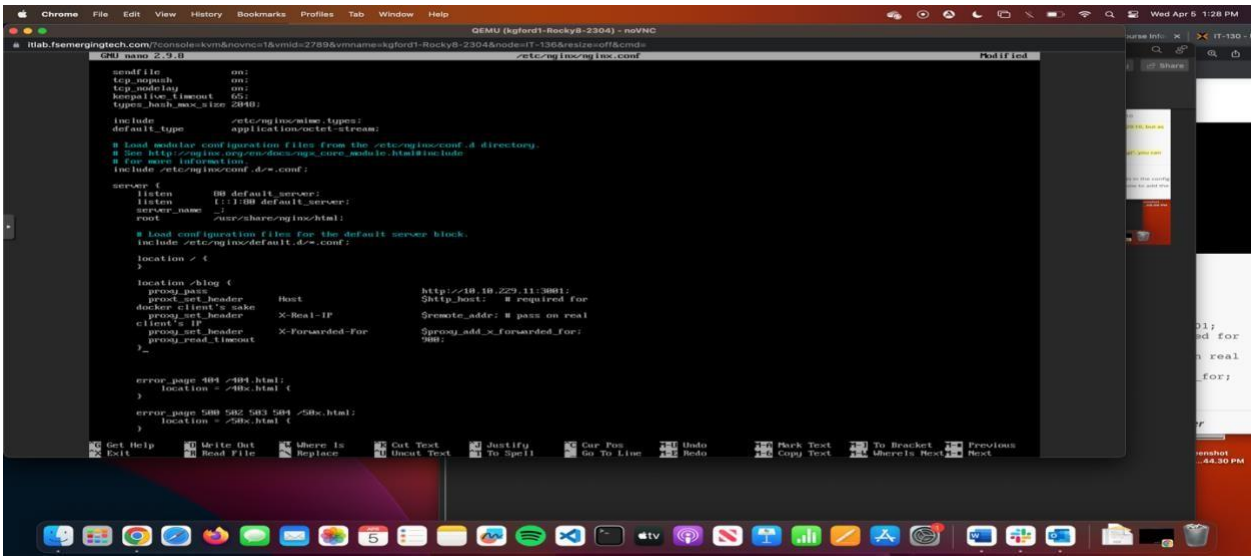


Install EPEL Packages

- yum install epel-release -y: This command installs the Extra Packages for Enterprise Linux (EPEL) repository, which provides additional packages that are not included in the default CentOS/RHEL repositories.

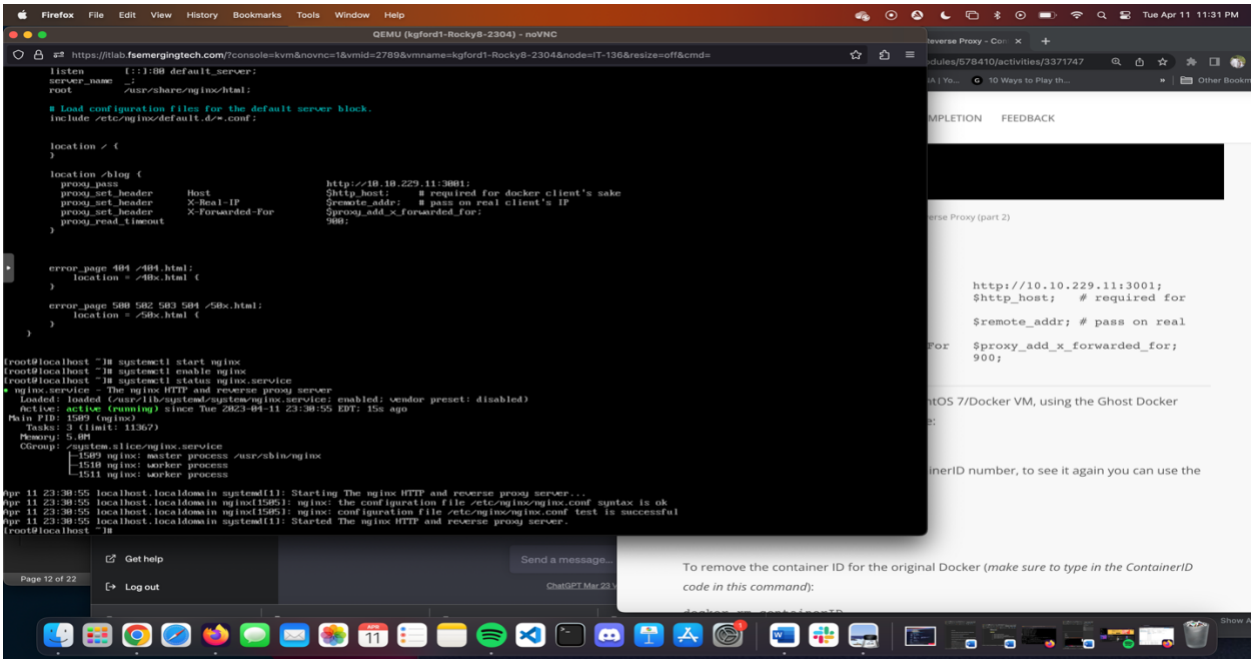
Install NginX

- yum install nginx -y: This command installs the Nginx web server, which is a popular open-source web server used for serving web content and reverse proxying.

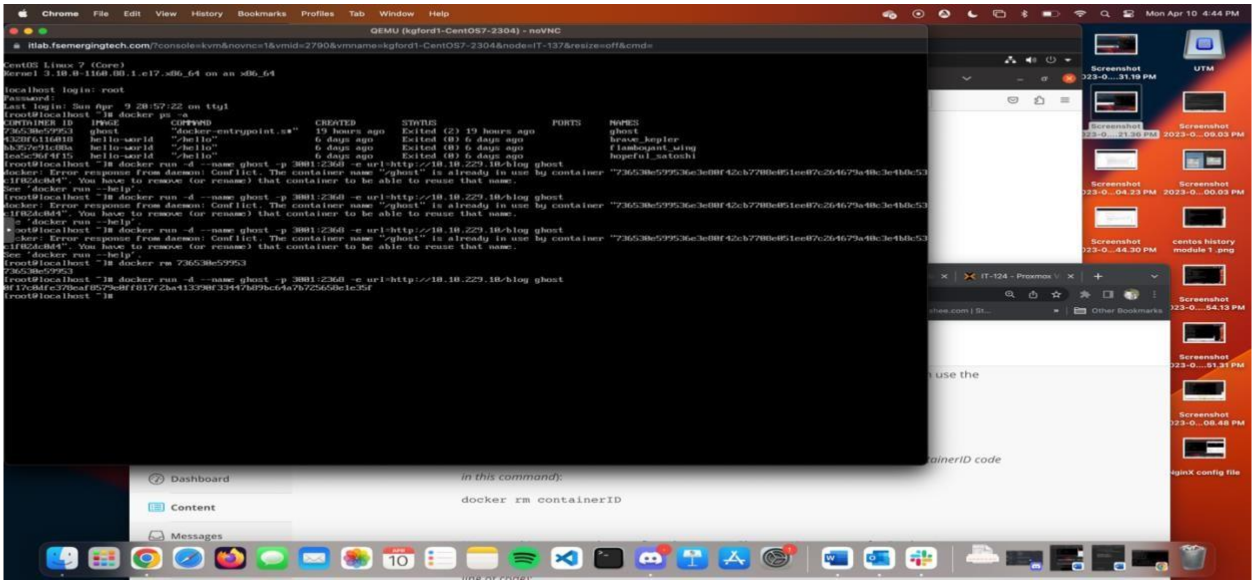


Start and Enable NginX

- systemctl enable nginx and systemctl start nginx: These commands enable and start the Nginx web server service.
- service nginx status: This command checks the status of the Nginx web server service, verifying that it is running.
- nano /etc/nginx/nginx.conf: This command opens the Nginx configuration file in the Nano text editor, allowing you to add or modify configuration settings for the web server.
- Within the Nginx configuration file, add the reverse proxy statements for the Ghost site, save the changes, and reload the Nginx service. The reverse proxy configuration allows Nginx to act as an intermediary between the client and the Ghost application running on a separate server



Reverse Proxy to Ghost Site



Install WordPress on Ubuntu - LAMP Stack

Base Ubuntu 18.04 Install

1. Begin by downloading the ISO file for Ubuntu 18.04 LTS from the official Ubuntu website.
2. Install virtualization software like Oracle VM VirtualBox or VMware Workstation.
3. To create a new virtual machine, launch the virtualization software and press the "New" button.
4. Choose "Linux" as the operating system and "Ubuntu (64-bit)" as the version for the virtual machine.
5. Give the virtual machine at least 2 GB of RAM and create a new virtual hard disk with a minimum size of 20 GB.
6. Choose "Use an existing virtual hard disk file" and then navigate to the Ubuntu 18.04 LTS ISO file.
7. Configure the virtual machine's settings to meet your needs, such as the number of processors and network settings.
8. Launch the virtual machine and choose "Install Ubuntu" from the boot menu.
9. Select your preferred language and press the "Continue" button.
10. Choose your keyboard layout and press the "Continue" button.
11. Select "Minimal installation" and click "Continue" on the "Updates and Other Software" screen.
12. Select "Erase disk and install Ubuntu" and click "Continue" on the "Installation Type" screen.
13. Select the virtual disk on which you want to install Ubuntu and press the "Install Now" button.
14. Click "Continue" on the pop-up window to confirm the changes.
15. Click "Continue" after selecting your time zone.
16. Enter your user information, which should include your name, computer name, username, and password. When finished, press the "Continue" button.
17. Ubuntu will now begin the installation process. This could take some time.

18. Restart the virtual machine once the installation is complete.

Set Static IP

1. Ctrl+Alt+T will launch a terminal in your Ubuntu virtual machine.

2. To edit the network configuration file, enter the command below:
   sudo nano /etc/netplan/01-netcfg.yaml

3. Find the section for your network interface in the file. It will probably have the name "eth0" or "ens33" and should be listed under "network".

4. The section should be changed to include a static IP address.

5. By pressing Ctrl+X, Y, and Enter, you can save and close the document.

6. By entering the following command, you can apply the new network configuration:
   sudo netplan apply

7. By entering the following command, you can check to see if the new IP address has been set:
   Ip address display

8. Under the network interface you changed, you should see the new IP address listed.

SSH into Ubuntu VM

Open the Terminal

- ssh username@server_ip_address:

use your user name and the the remote Server's IP address in the command above
This allows remote access to the Ubuntu machine.

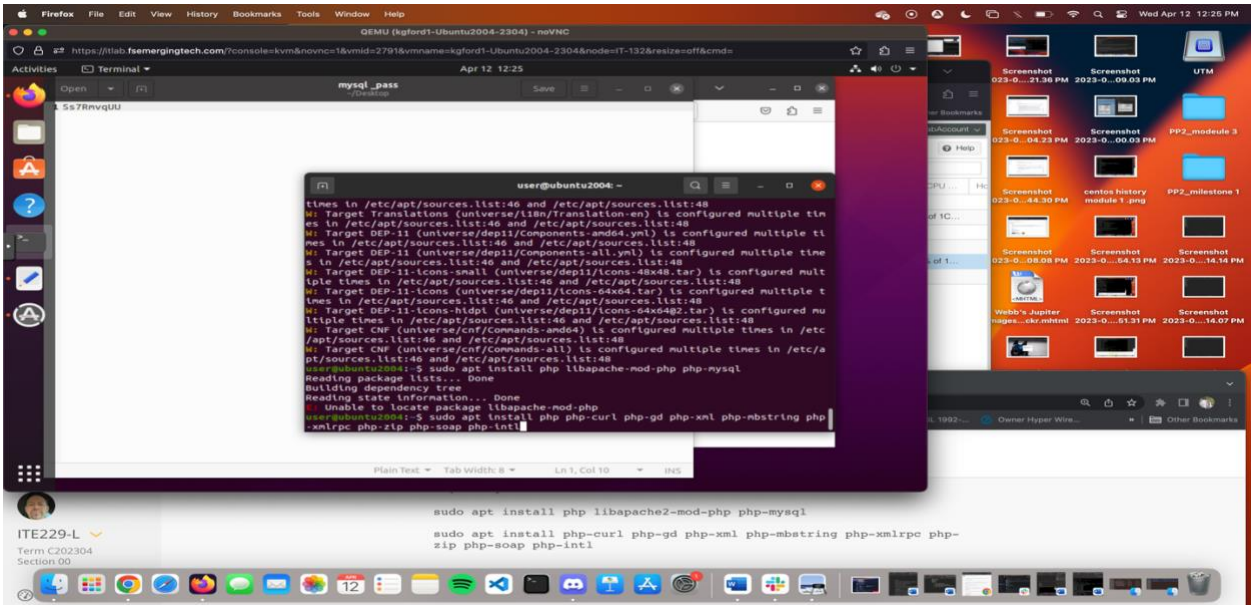
Update Ubuntu

- sudo apt-get update
- sudo apt upgrade

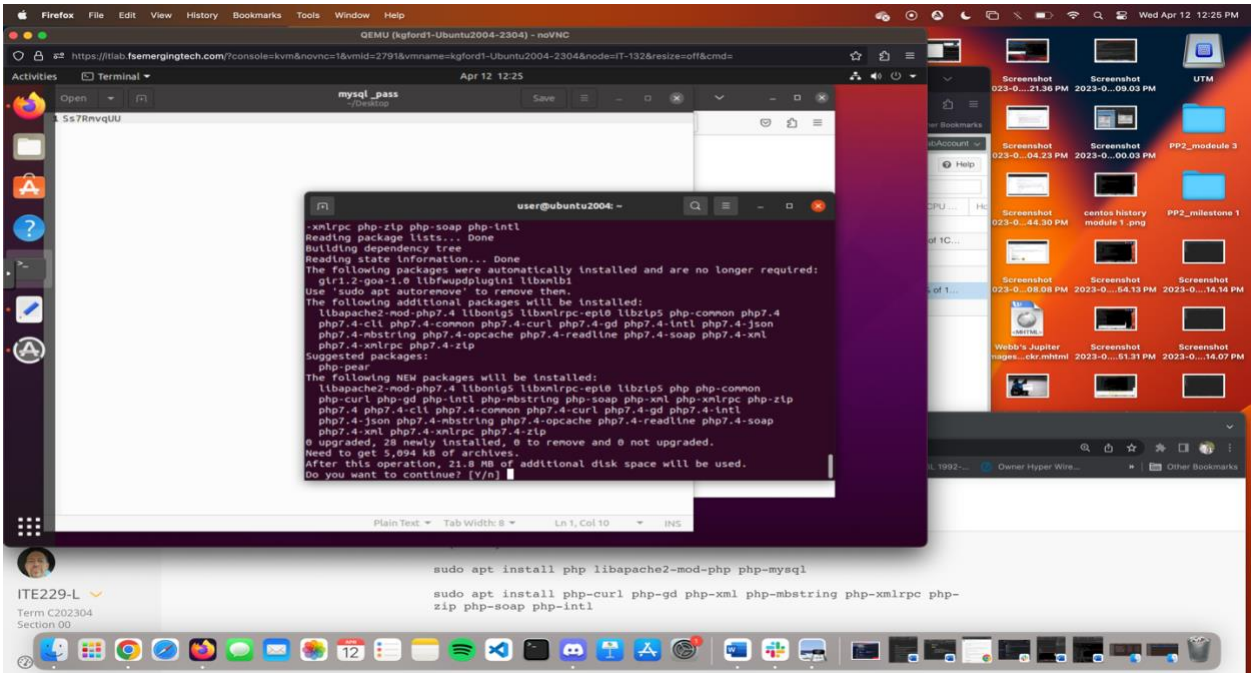
The first command lists upgrades for the installed packages, and the second command updates those upgrades to the most recent versions.

Install and Configure Apache

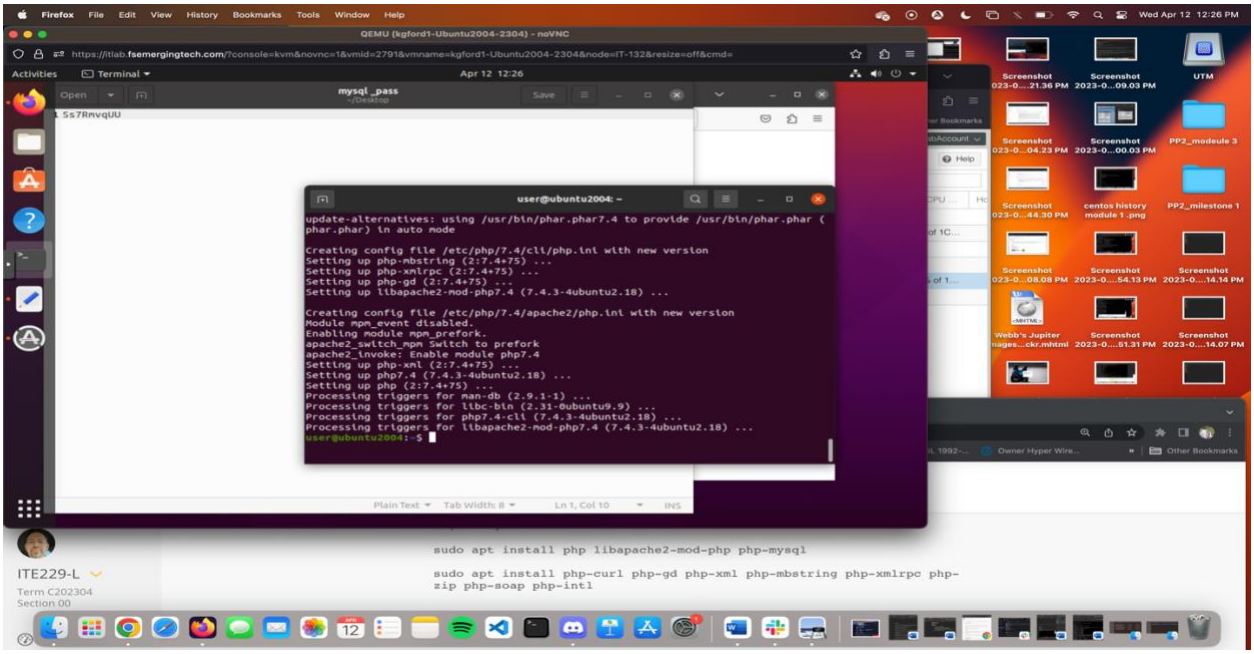
- sudo apt install apache2: Command to install packages for Apache web server

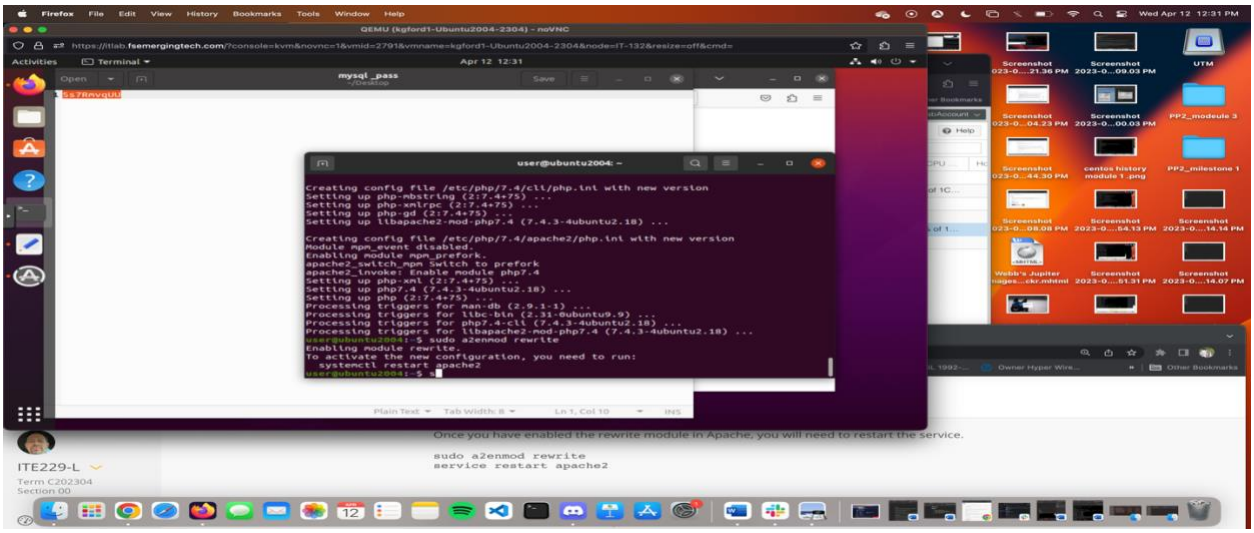


Say yes to install packages



Installing packages



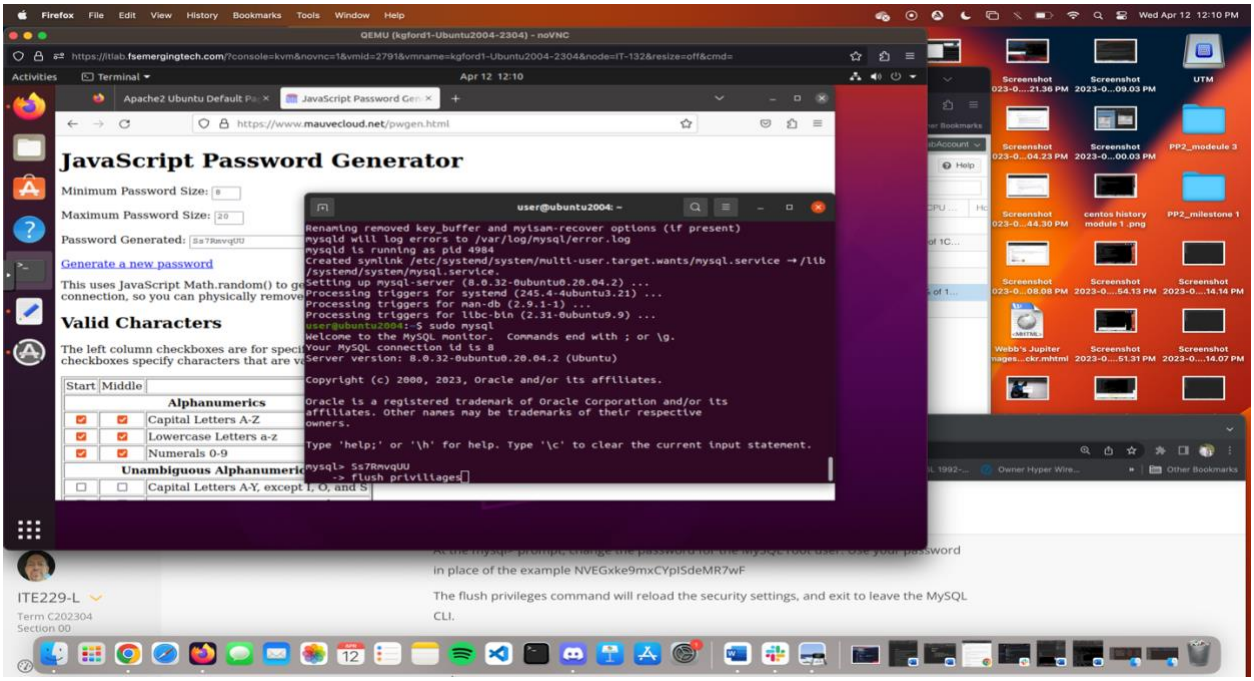


Install and Configure MySQL

- sudo apt install mysql-server:

This command installs the MySQL package

Create a complex password for MySQL root user:



- sudo mysql: This command opens the MySQL command-line interface

- ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'new_password';

FLUSH PRIVILEGES;

Exit; :

Replace **new_password** with the password you created in the previous step. The first command changes the password for the MySQL root user, and the second command reloads the security settings. The **exit** command exits the MySQL CLI.

- sudo nano /etc/apt/sources.list:

This command opens the **sources.list** file in the Nano text editor. Add the keyword **universe** to the end of each line that doesn't already have it. Then save and exit.

- sudo apt update:

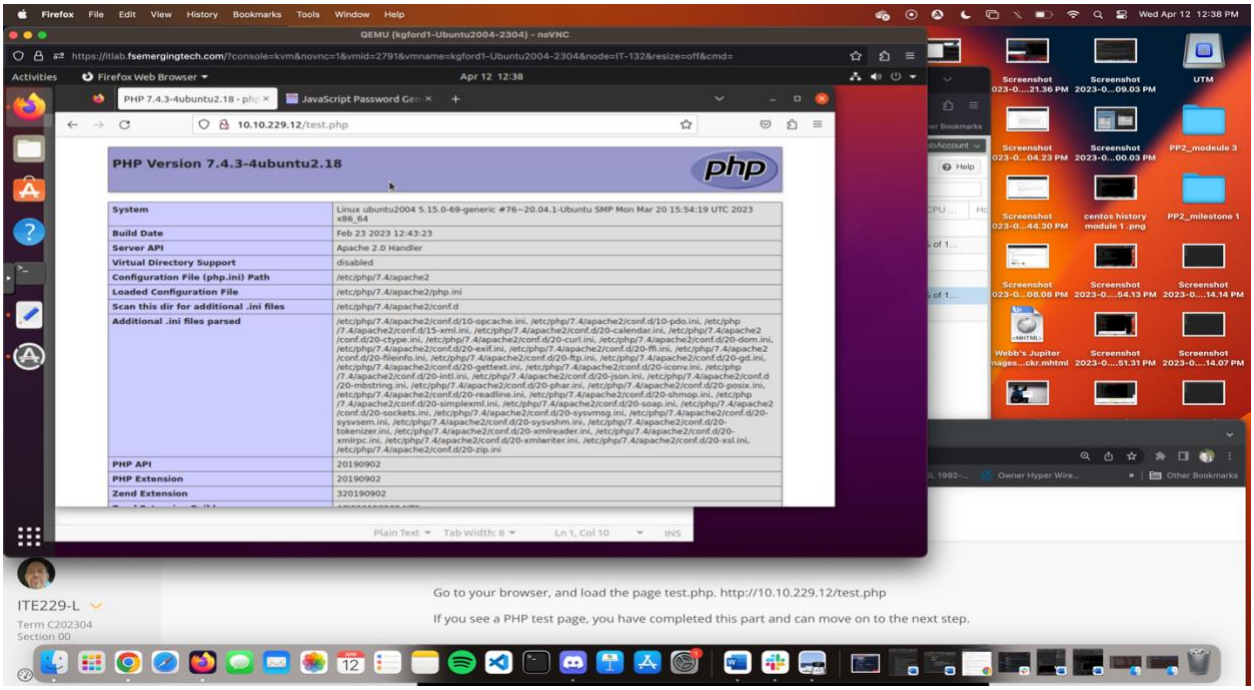
This command updates the package lists to include the newly added repository.

Install and configure PHP

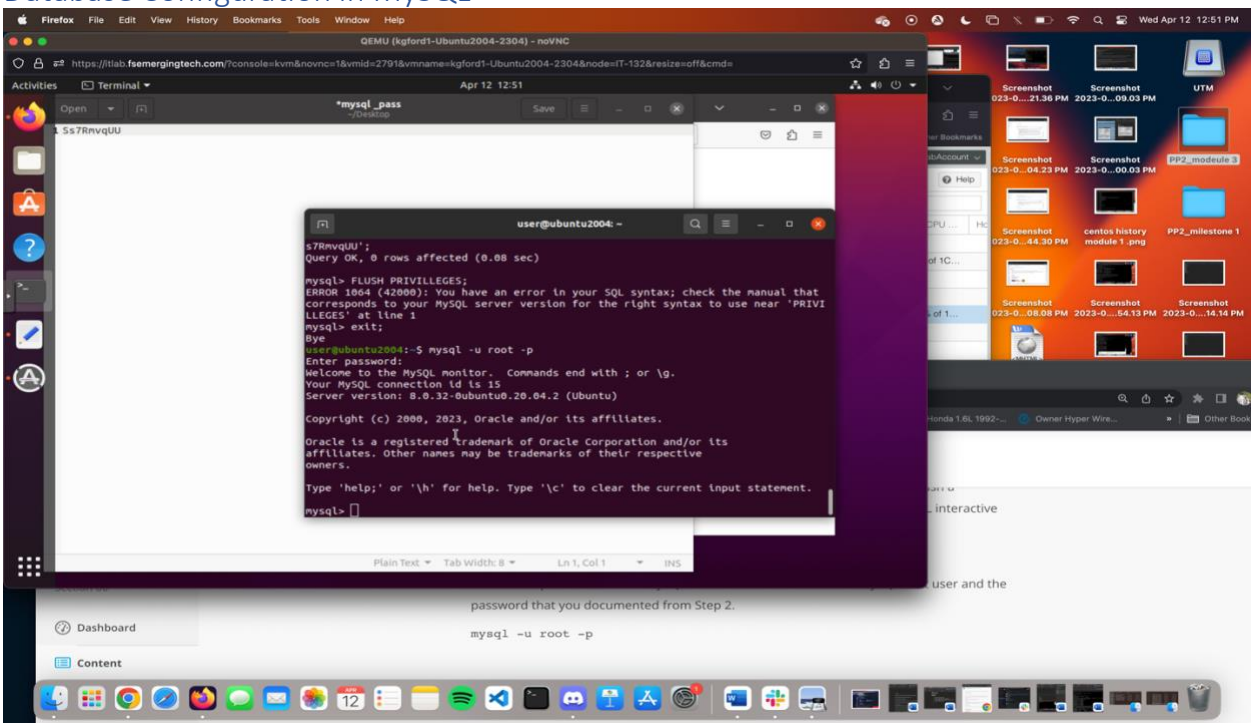
- sudo apt install php libapache2-mod-php php-mysql
- sudo apt install php-curl php-gd php-xml php-mbstring:

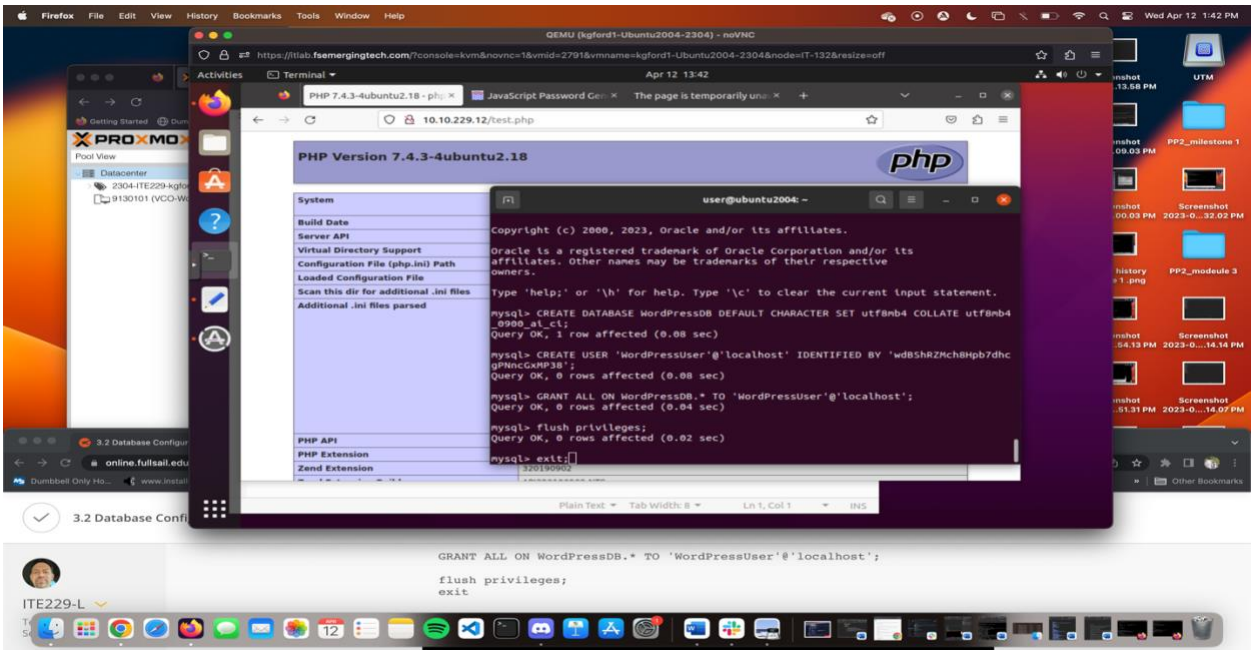
installs PHP and the required extensions for WordPress to work.

Test PHP



Database Configuration in MySQL





Install WordPress

WordPress Install using git client, from GitHub:

Empty the /var/www/html directory, then clone the git repository into /var/www/html

- sudo chown $USER:$USER /var/www/html/*: removes all files and folders inside the /var/www/html directory.
- sudo rm /var/www/html/*

Verify that your html directory is empty:

sudo ls /var/www/html/

Install git:

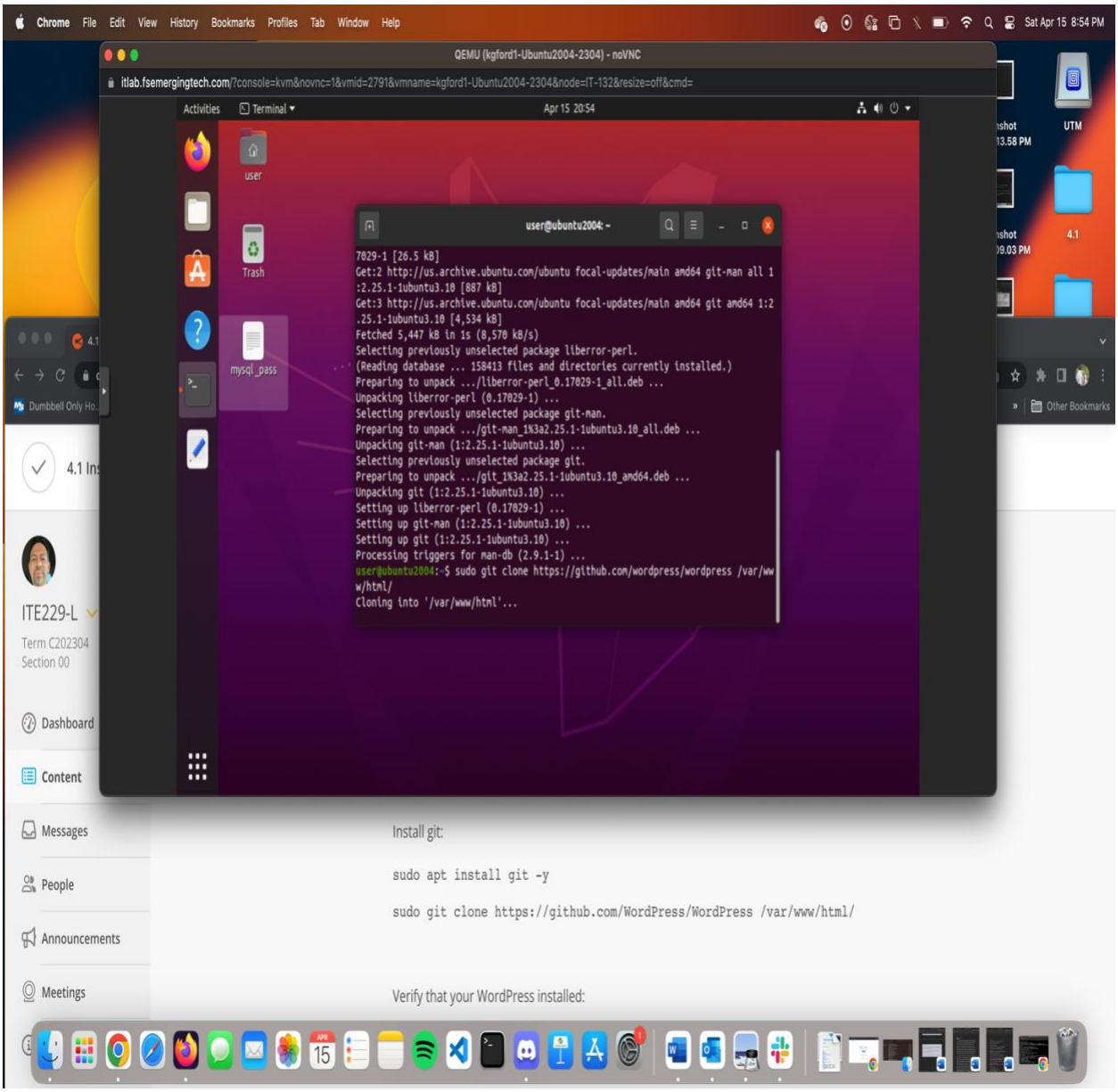
- sudo apt install git -y

Clone WordPress

- sudo git clone https://github.com/WordPress/WordPress /var/www/html/: clones the WordPress repository from GitHub into the /var/www/html directory.

Verify that your WordPress installed:

- sudo ls /var/www/html: verifies that WordPress is installed in the /var/www/html directory



Edit Ownership

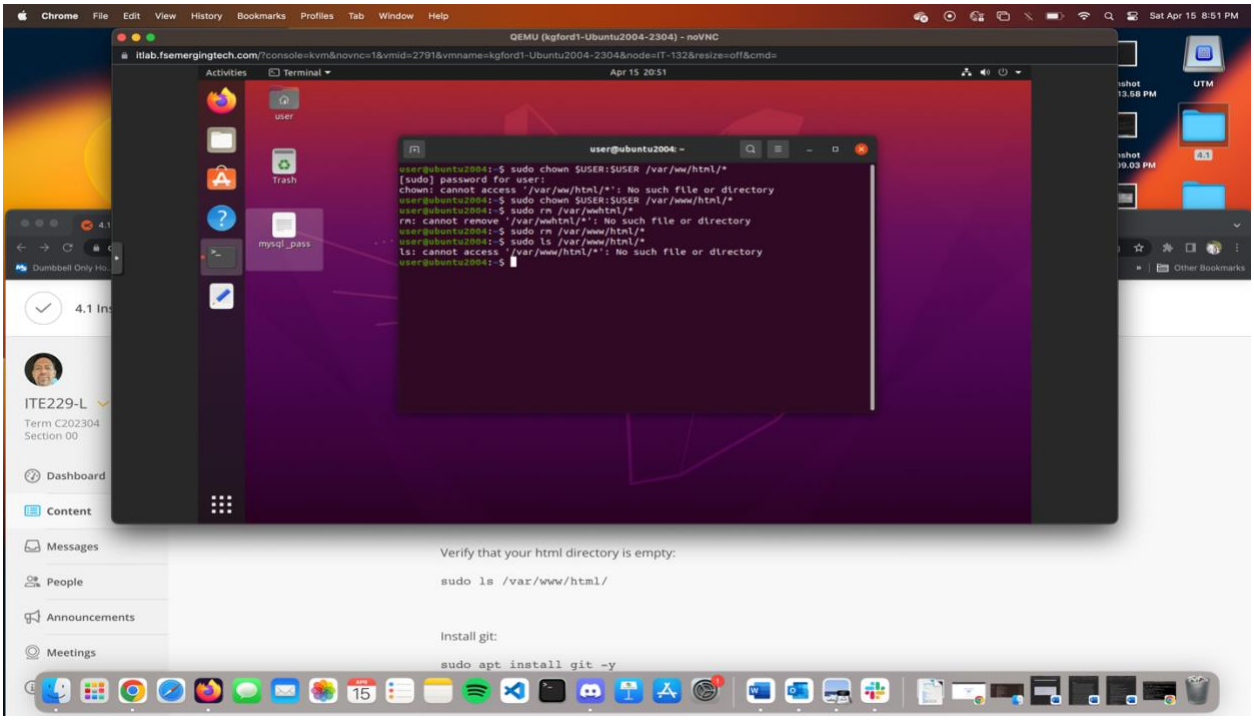
Verify permissions:

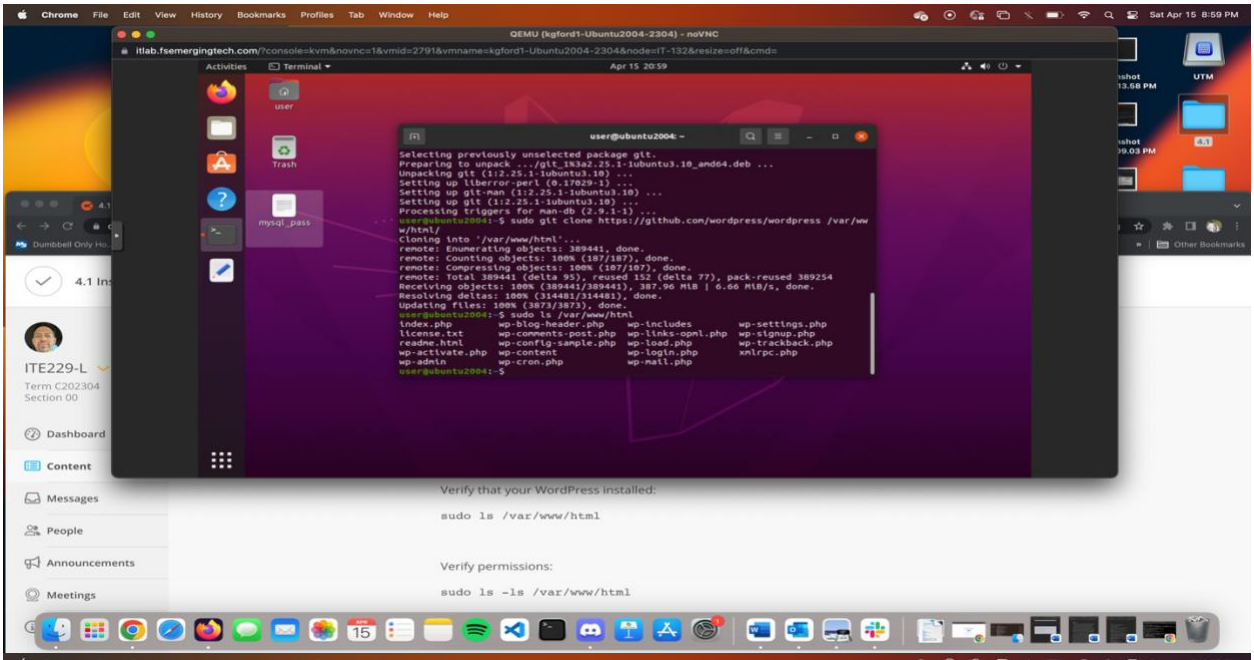
- sudo ls -ls /var/www/html: lists the contents of the /var/www/html directory with permissions to verify that the ownership is set correctly.

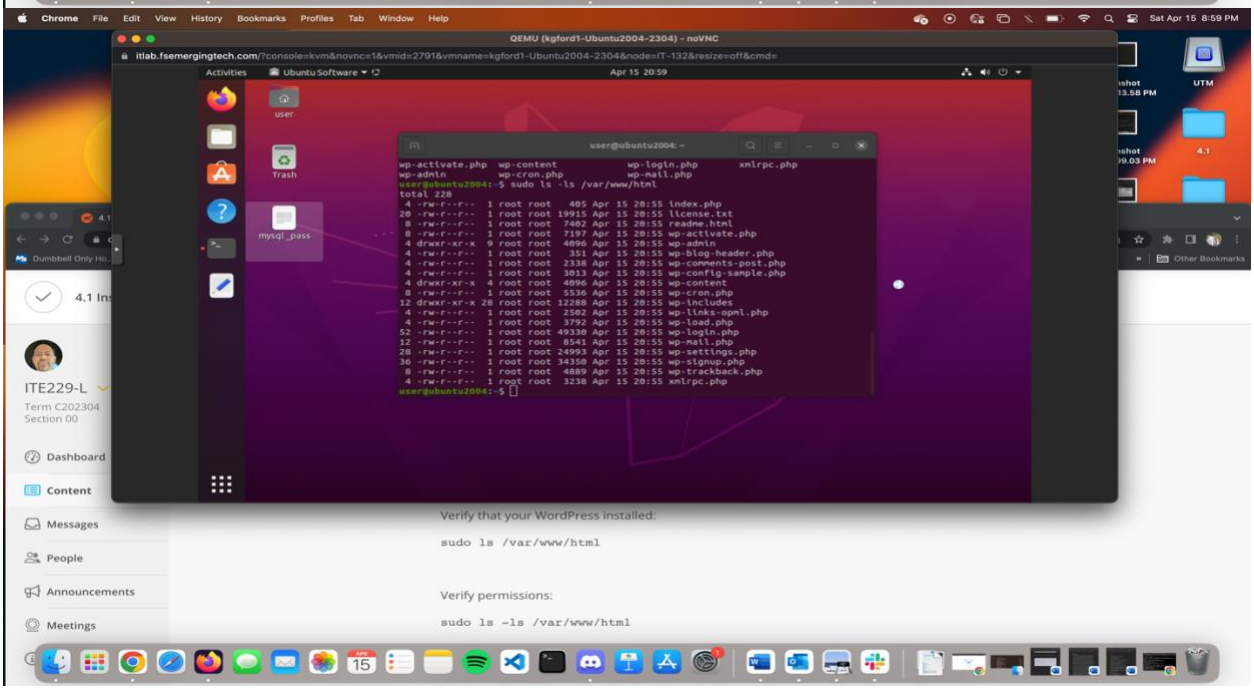
Set the ownership of the /var/www/html contents to the Apache user, and set ownership on the html directory itself.

- sudo chown -R www-data:www-data /var/www/html/*: sets the ownership of all files and folders inside the /var/www/html directory to the Apache user.

- sudo chown www-data:www-data /var/www/html: sets the ownership of the /var/www/html directory to the Apache user




QEMU (kgford1-Ubuntu2004-2304) - noVNC
itlab.fsemergingtech.com/?console=kvm&novnc=1&vmid=2791&vmname=kgford1-Ubuntu2004-2304&node=IT-132&resize=off&cmd=
user@ubuntu2004: ~
Selecting previously unselected package git.
Preparing to unpack .../git_1%3a2.25.1-1ubuntu3.10_amd64.deb ...
Unpacking git (1:2.25.1-1ubuntu3.10) ...
Setting up liberror-perl (0.17029-1) ...
Setting up git-man (1:2.25.1-1ubuntu3.10) ...
Setting up git (1:2.25.1-1ubuntu3.10) ...
Processing triggers for man-db (2.9.1-1) ...
user@ubuntu2004:~$ sudo git clone https://github.com/wordpress/wordpress /var/www/html/
Cloning into '/var/www/html'...
remote: Enumerating objects: 389441, done.
remote: Counting objects: 100% (187/187), done.
remote: Compressing objects: 100% (107/107), done.
remote: Total 389441 (delta 95), reused 152 (delta 77), pack-reused 389254
Receiving objects: 100% (389441/389441), 387.96 MiB | 6.66 MiB/s, done.
Resolving deltas: 100% (314481/314481), done.
Updating files: 100% (3873/3873), done.
user@ubuntu2004:~$ sudo ls /var/www/html
index.php wp-blog-header.php wp-includes wp-settings.php
license.txt wp-comments-post.php wp-links-opml.php wp-signup.php
readme.html wp-config-sample.php wp-load.php wp-trackback.php
wp-activate.php wp-content wp-login.php xmlrpc.php
wp-admin wp-cron.php wp-mail.php
user@ubuntu2004:~$
ITE229-L
Term C202304
Section 00
Dashboard
Content
Messages
People
Announcements
Meetings
Verify that your WordPress installed:
sudo ls /var/www/html
Verify permissions:
sudo ls -ls /var/www/html


QEMU (kgford1-Ubuntu2004-2304) - noVNC
itlab.fsemergingtech.com/?console=kvm&novnc=1&vmid=2791&vmname=kgford1-Ubuntu2004-2304&node=IT-132&resize=off&cmd=
user@ubuntu2004: ~
wp-activate.php wp-content wp-login.php xmlrpc.php
wp-admin wp-cron.php wp-mail.php
user@ubuntu2004:~$ sudo ls -ls /var/www/html
total 228
4 -rw-r--r-- 1 root root 405 Apr 15 20:55 index.php
20 -rw-r--r-- 1 root root 19915 Apr 15 20:55 license.txt
8 -rw-r--r-- 1 root root 7402 Apr 15 20:55 readme.html
8 -rw-r--r-- 1 root root 7197 Apr 15 20:55 wp-activate.php
4 drwxr-xr-x 9 root root 4096 Apr 15 20:55 wp-admin
4 -rw-r--r-- 1 root root 351 Apr 15 20:55 wp-blog-header.php
4 -rw-r--r-- 1 root root 2338 Apr 15 20:55 wp-comments-post.php
4 -rw-r--r-- 1 root root 3013 Apr 15 20:55 wp-config-sample.php
4 drwxr-xr-x 4 root root 4096 Apr 15 20:55 wp-content
8 -rw-r--r-- 1 root root 5536 Apr 15 20:55 wp-cron.php
12 drwxr-xr-x 28 root root 12288 Apr 15 20:55 wp-includes
4 -rw-r--r-- 1 root root 2502 Apr 15 20:55 wp-links-opml.php
4 -rw-r--r-- 1 root root 3792 Apr 15 20:55 wp-load.php
52 -rw-r--r-- 1 root root 49330 Apr 15 20:55 wp-login.php
12 -rw-r--r-- 1 root root 8541 Apr 15 20:55 wp-mail.php
28 -rw-r--r-- 1 root root 24993 Apr 15 20:55 wp-settings.php
36 -rw-r--r-- 1 root root 34350 Apr 15 20:55 wp-signup.php
8 -rw-r--r-- 1 root root 4889 Apr 15 20:55 wp-trackback.php
4 -rw-r--r-- 1 root root 3238 Apr 15 20:55 xmlrpc.php
user@ubuntu2004:~$
ITE229-L
Term C202304
Section 00
Dashboard
Content
Messages
People
Announcements
Meetings
Verify that your WordPress installed:
sudo ls /var/www/html
Verify permissions:
sudo ls -ls /var/www/html

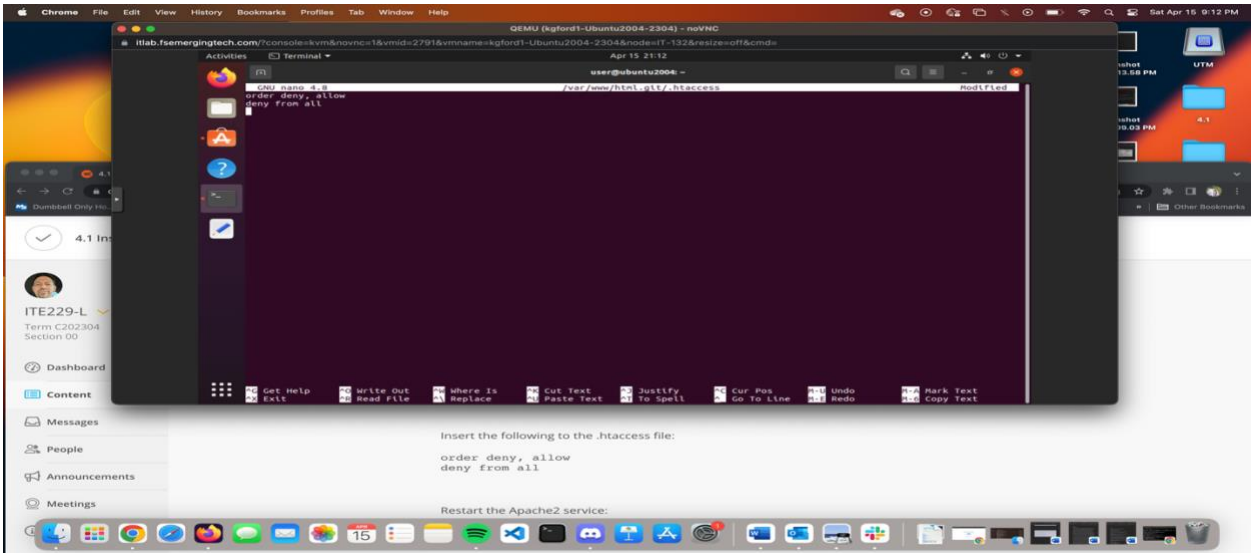
Edit .htaccess

- sudo nano /var/www/html/.git/.htaccess: creates a .htaccess file under the .git folder

Insert the following lines to the .htaccess file to secure the .git directory from unauthorized access:

order deny,

allow deny from all

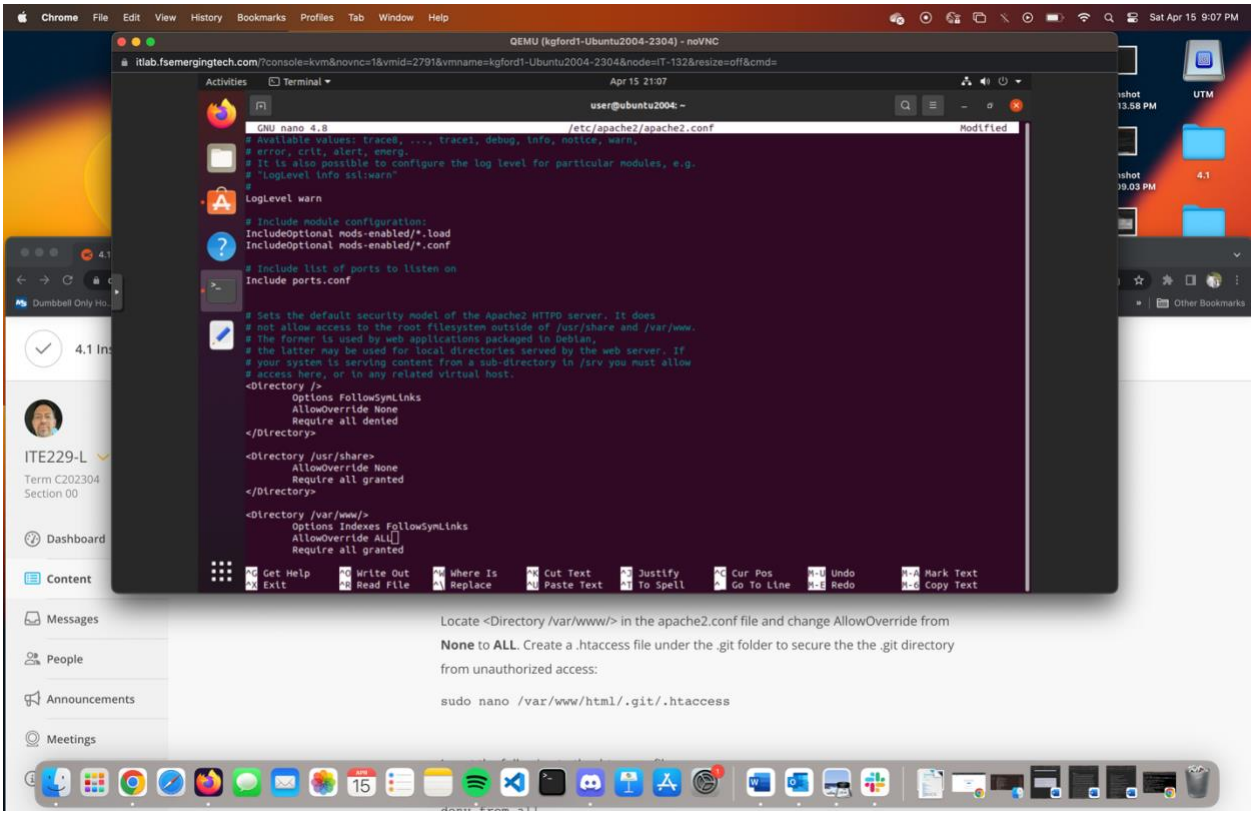


WordPress Configuration

Enable overrides for the /var/www path. Open the /etc/apache2/apache2.conf file.

- sudo nano /etc/apache2/apache2.conf: opens the global Apache configuration file in the Nano text editor

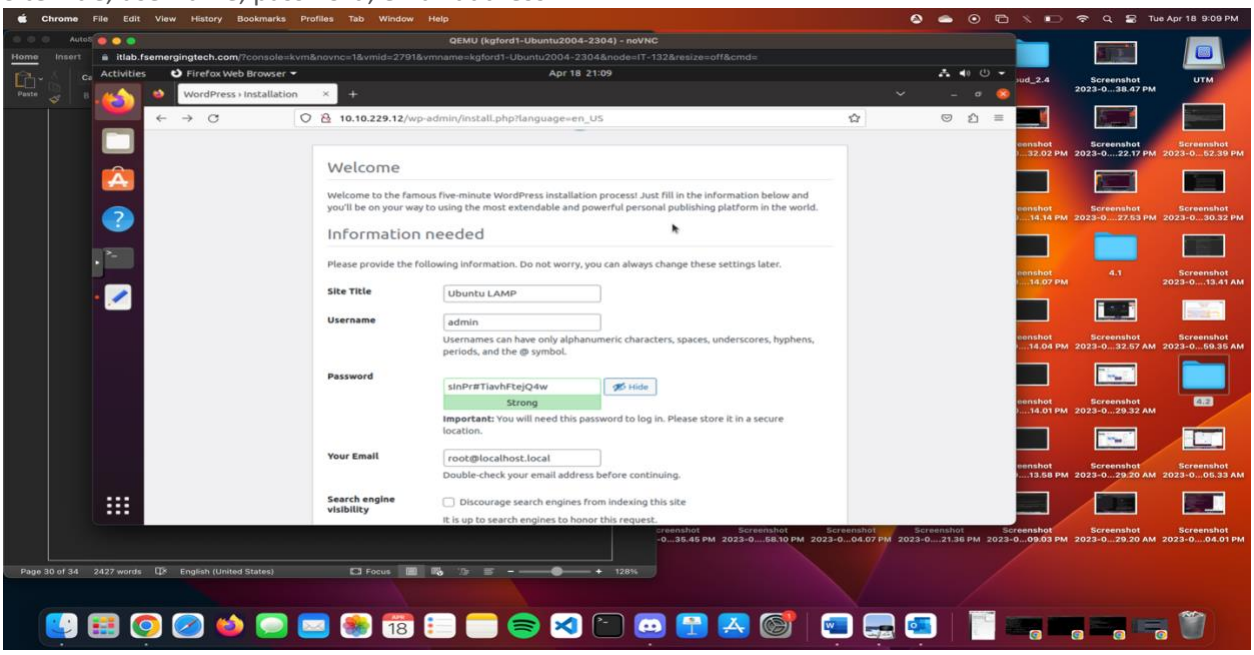
Change AllowOverride from None to ALL for "Directory /var/www" in the apache2.conf file. This allows overrides for the /var/www path.



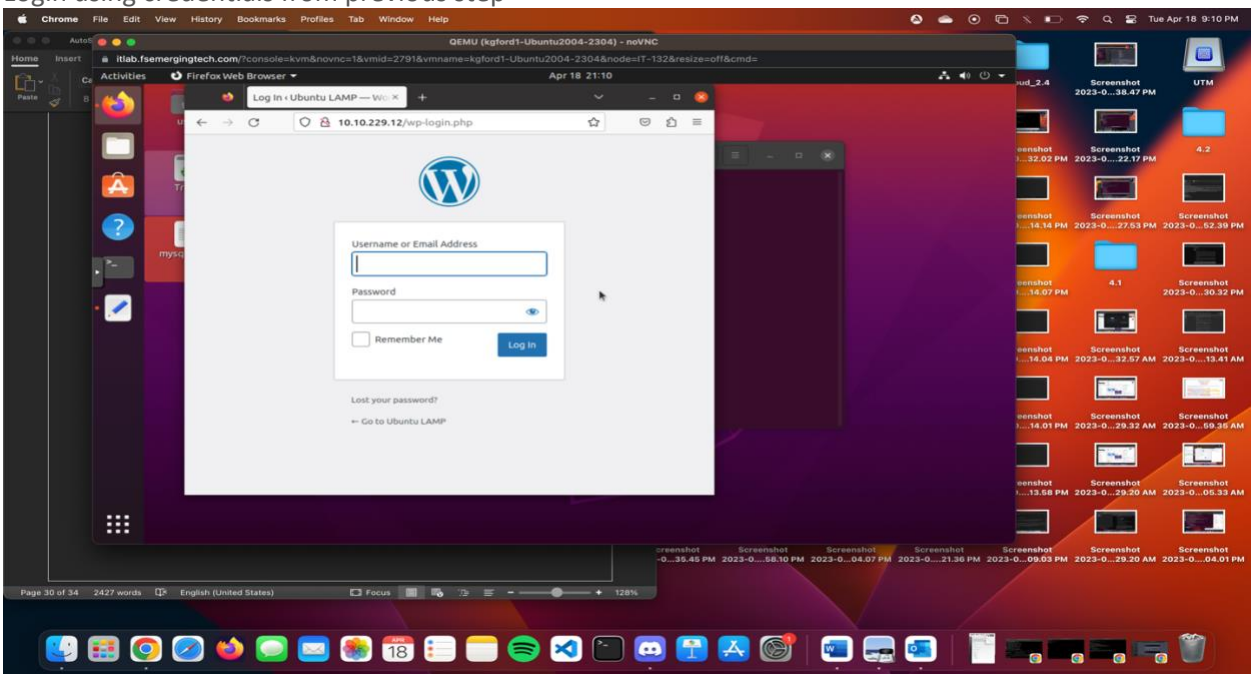
- **sudo service apache2 restart: restarts the Apache2 service to implement the configuration changes**

WordPress Configuration Process

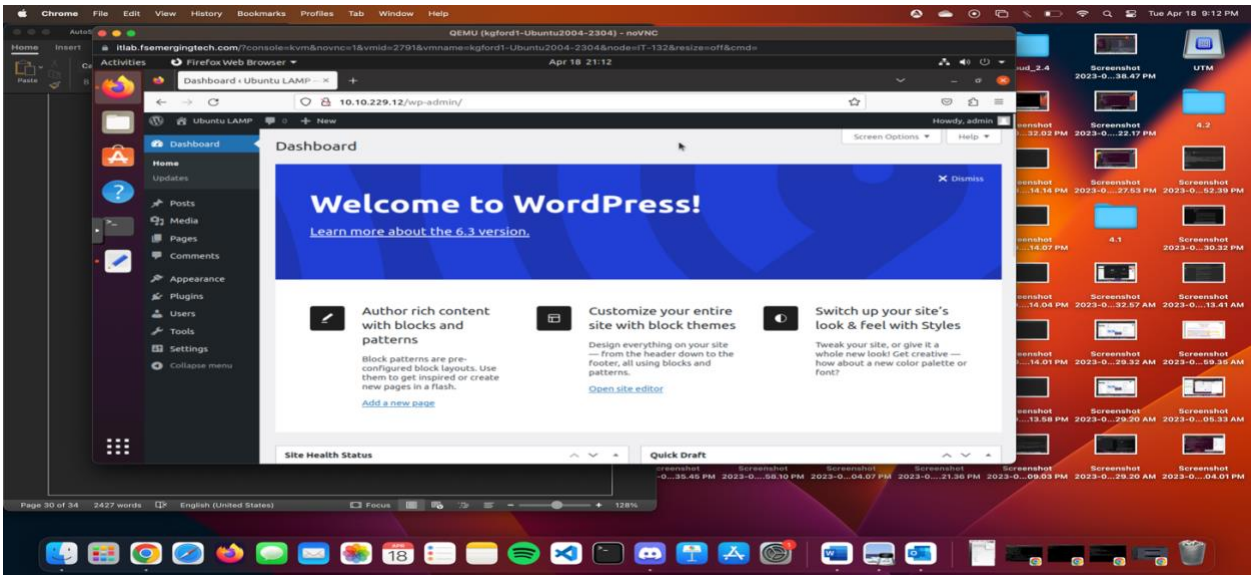
Input:
Site Title, username, password, email address



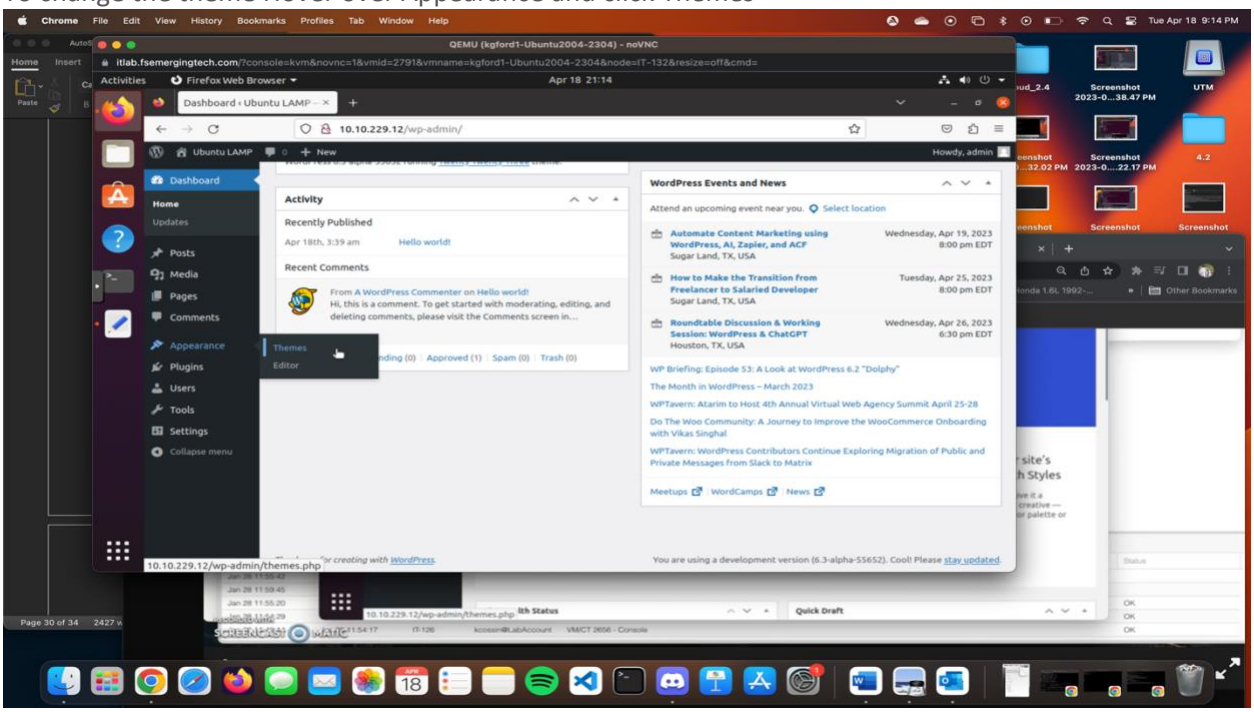
Login using credentials from previous step



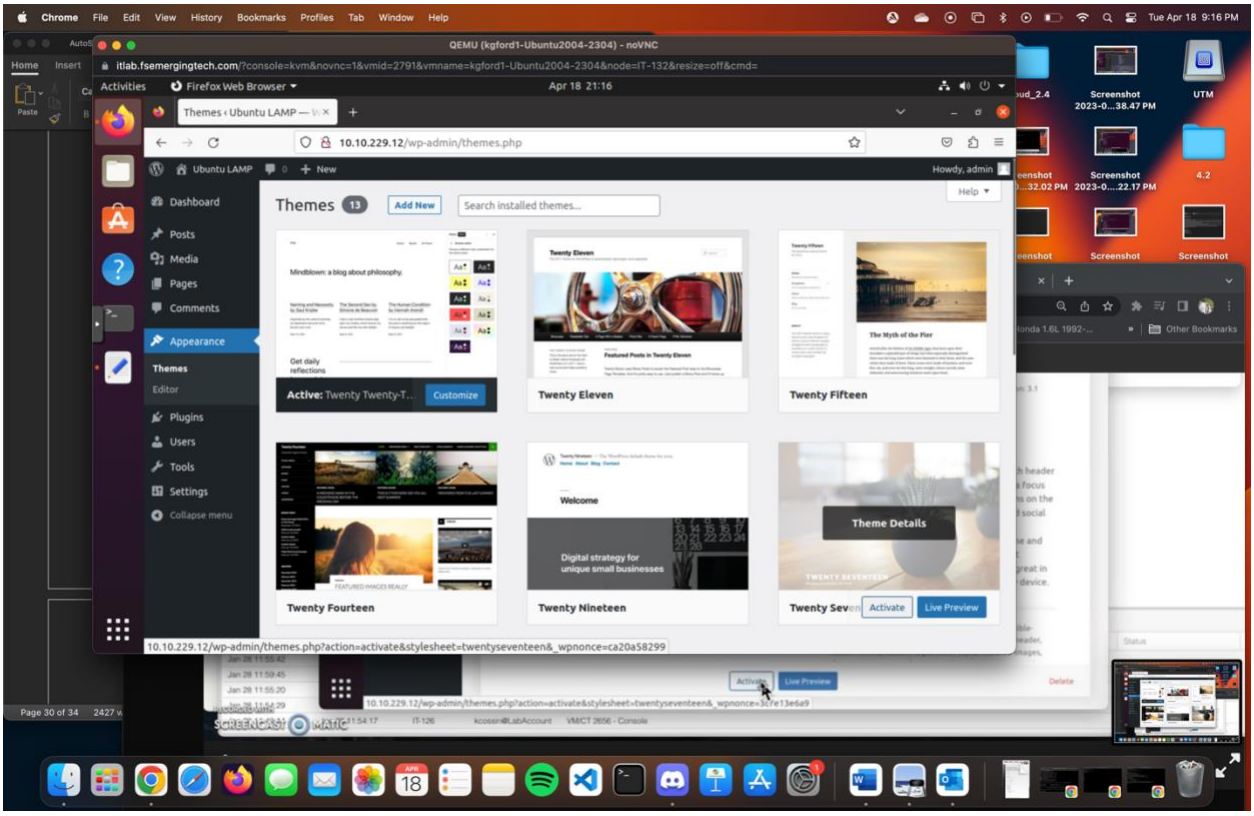
Welcome to WordPress



To change the theme Hover over Appearance and click Themes

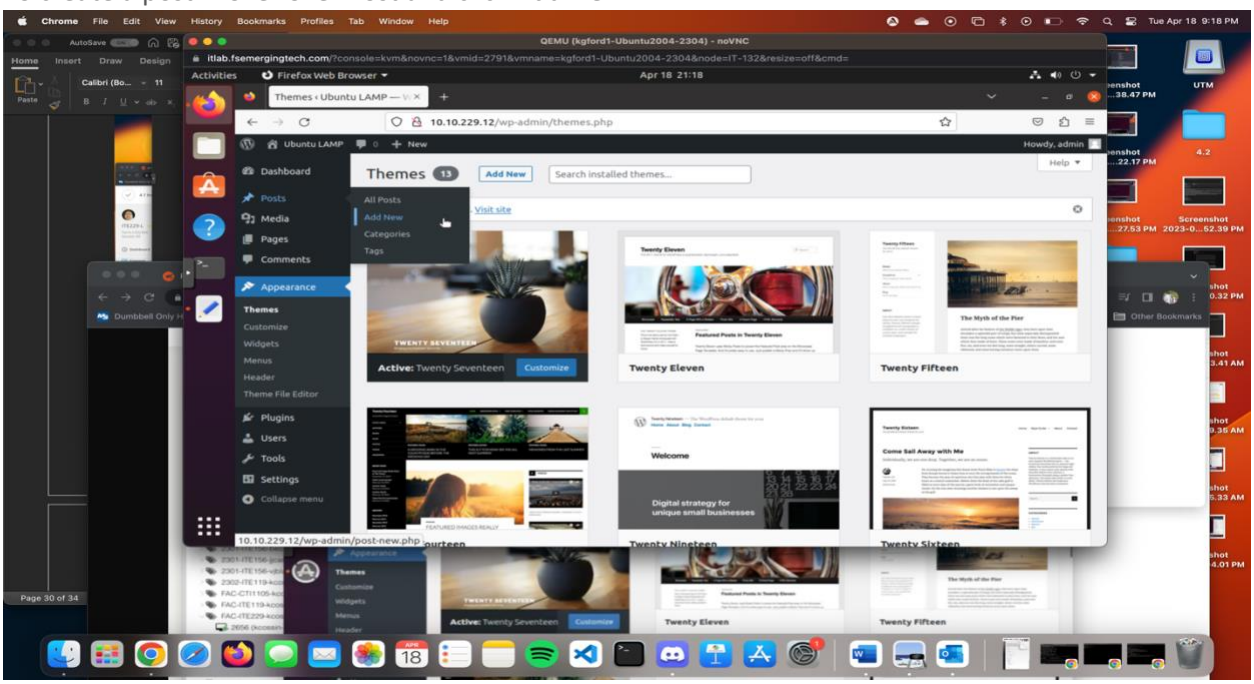


To Apply a theme, click Activate

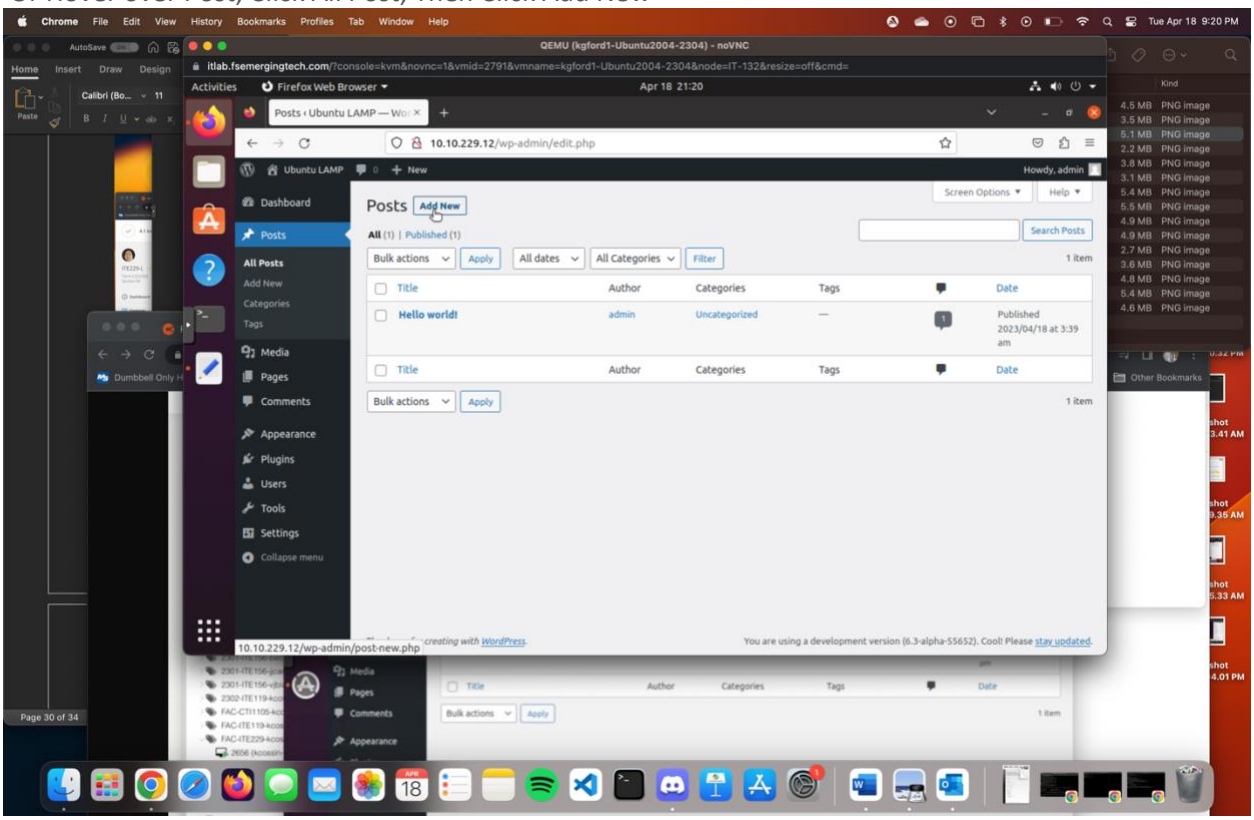


Test WordPress

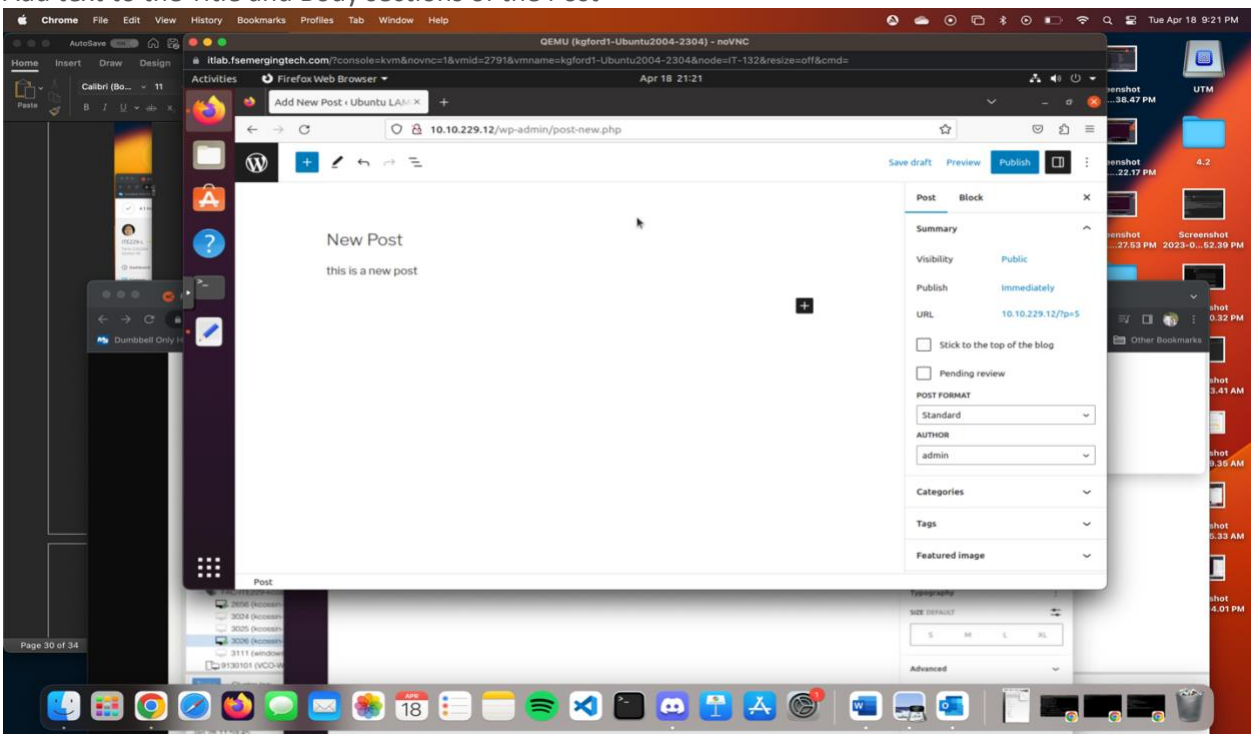
To create a post: Hover over Post and click Add New



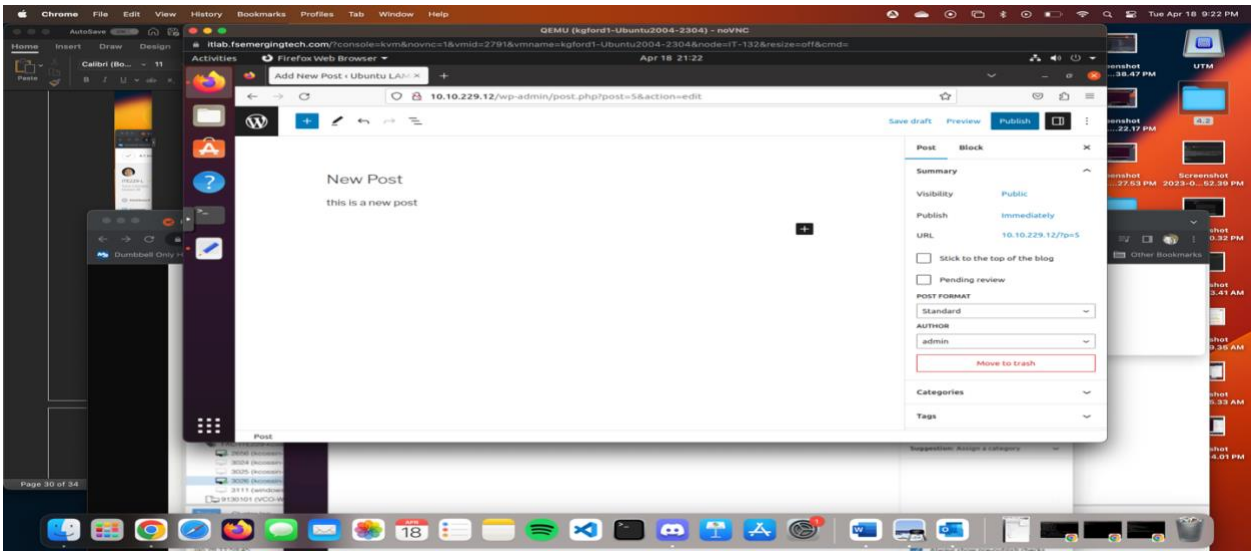
Or Hover over Post, Click All Post, Then Click Add New



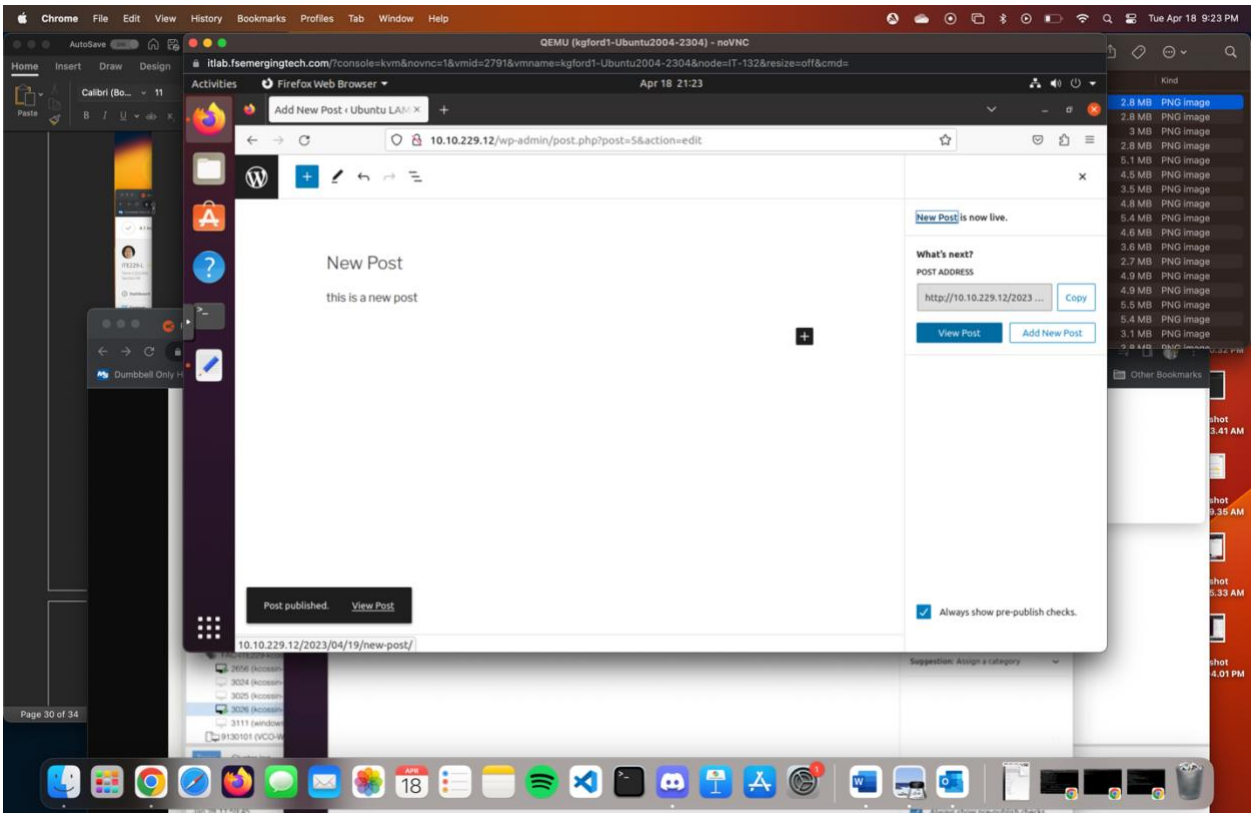
Add text to the Title and Body sections of the Post



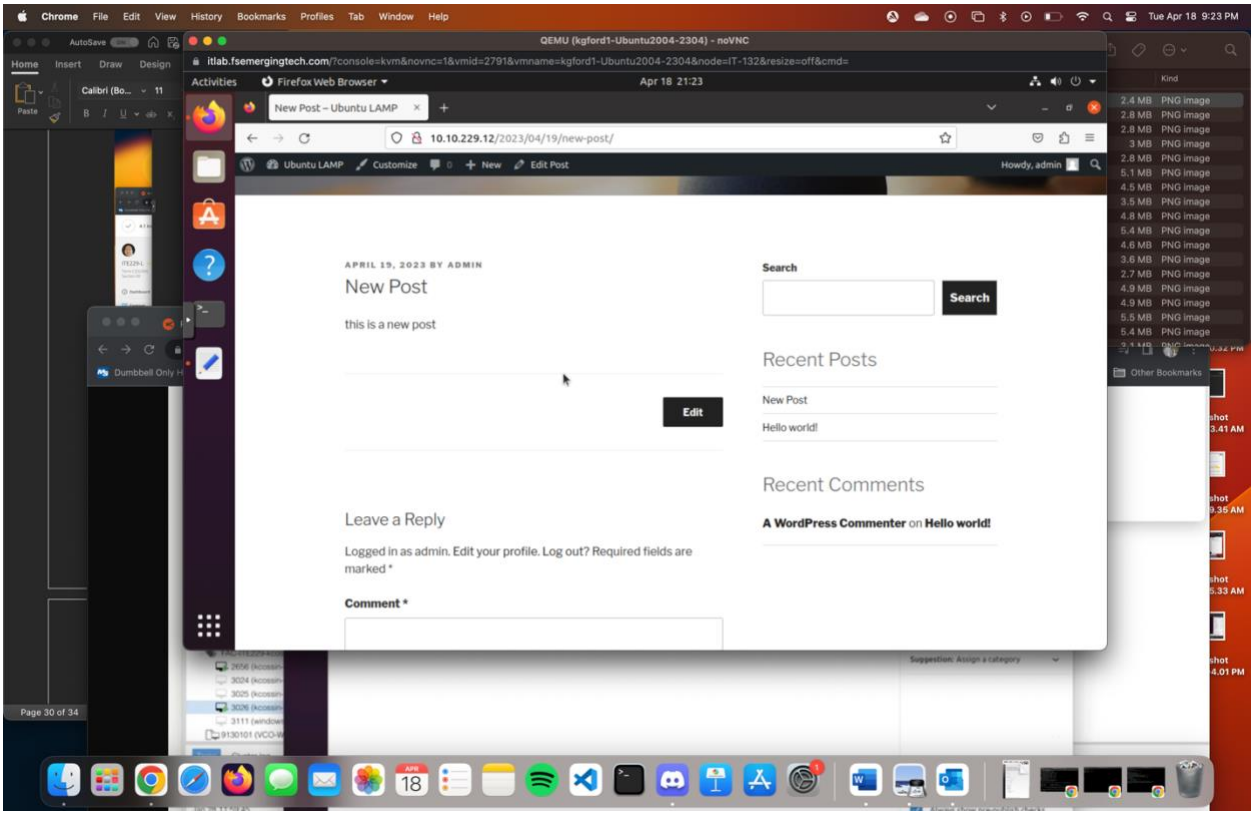
Then Click Publish



Click View Post



Next visit the URL for the WordPress site using a web browser



WordPress Security Settings and Configurations

Security Summary

The security of the WordPress installation has been significantly improved. The following changes have been made:

- File permissions have been updated to restrict access to only the owner and group.
- The WP-Config.php file has been moved to a directory outside the root directory.
- The Shield firewall has been installed and configured.

These changes have made it more difficult for unauthorized users to access sensitive files and directories. They have also helped to protect the WordPress installation from known vulnerabilities and make it more difficult for attackers to gain unauthorized access.

Defense-in-depth

Defense-in-Depth is the idea of using multiple layers of security to shield a system from different types of attacks. I used Defense-in-Depth in this exercise to harden the WordPress installation by putting various security controls in place. Changing file permissions, securing the WP-Config.php file, and installing and configuring the Shield firewall are a few examples of the multiple layers of security measures I used. Each of these steps adds another layer of defense against unauthorized entry and potential attacks.

I have significantly improved the security of the WordPress installation by putting in multiple layers of security measures, making it much more difficult for potential attackers to access the system. The Defense-in-Depth strategy helps to ensure that there are additional layers of security in place to deter a successful attack even if one security measure is circumvented or fails.

Before/After Configuration

steps taken for each of the vulnerabilities checked and the changes made to secure them:

1) Directory and file Permissions:

Before: Accessed the server and ran the command ls -la to view the permissions of the WordPress files and directories.

2) Change File Permissions:

Before: Accessed the server using SSH and ran the command ls -la to view the permissions of a WordPress file.

After: After: Changed permissions of directories to 750 and files to 640 using the command find /path/to/wordpress -type d -exec chmod 750 {} \; and find /path/to/wordpress -type f -exec chmod 640 {} \;.

3) Securing WP-Config.php:

Before: Accessed the server using SSH and located the WP-Config.php file in the WordPress root directory.

After: Moved the WP-Config.php file to a directory outside the root directory and changed its permissions to 400 using the command mv /path/to/wordpress/wp-config.php /path/to/non-public/directory/ && chmod 400 /path/to/non-public/directory/wp-config.php.

4) Firewalls select Shield:

Before: Accessed the WordPress admin dashboard and installed the Shield Security plugin.

After: Configured the Shield firewall by enabling all recommended settings and adding custom rules for specific IPs and user agents.

For each of the changes made, I documented the before and after.

Testing and Validation Process

1. For the file permissions and WP-Config.php changes, I accessed the server using and ran the command **ls -la** to view the permissions of the WordPress files and directories, and the WP-Config.php file. I verified that the permissions were changed to the desired values, restricting access to only the owner and group.

   Next I race the following commands:

   cd /var/www/html

   cd wp-admin

   cd wp-content

   cd wp-includes

   and permission was denied to all

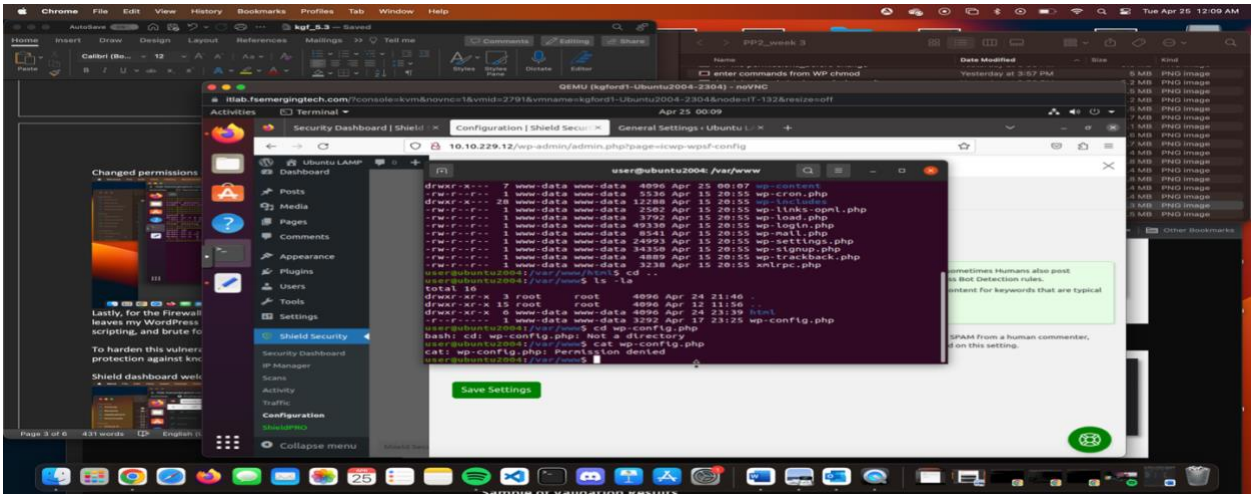
   next entered:

   cd /var/www/

   then

   cat wp-config.php

   premission was denied



2. For the Shield firewall, I logged into the WordPress admin dashboard for the Shield firewall and verified that all suggested settings were active and that the custom rules had been added

properly. After that, I ran a vulnerability scan on the WordPress installation with a vulnerability scanner tool to make sure the system was secure against known threats.

WordPress Security Conclusion

In this exercise, I secured a WordPress installation using the Defense-in-Depth strategy. To defend a system from attack, this method employs multiple layers of security controls. By employing this strategy, I was able to find vulnerabilities in the WordPress installation and fix them, strengthening its security.

By taking similar actions, the Defense-in-Depth strategy can be applied to other systems. Determine any potential system weaknesses first. Next, determine how much risk these vulnerabilities pose. Third, take the necessary security precautions to lessen the risk of an attack.

By taking these precautions, businesses can lower their risk of data breaches, safeguard sensitive data, and preserve the privacy, integrity, and accessibility of their systems and data.

To put the Defense-in-Depth strategy into practice, follow these specific steps:

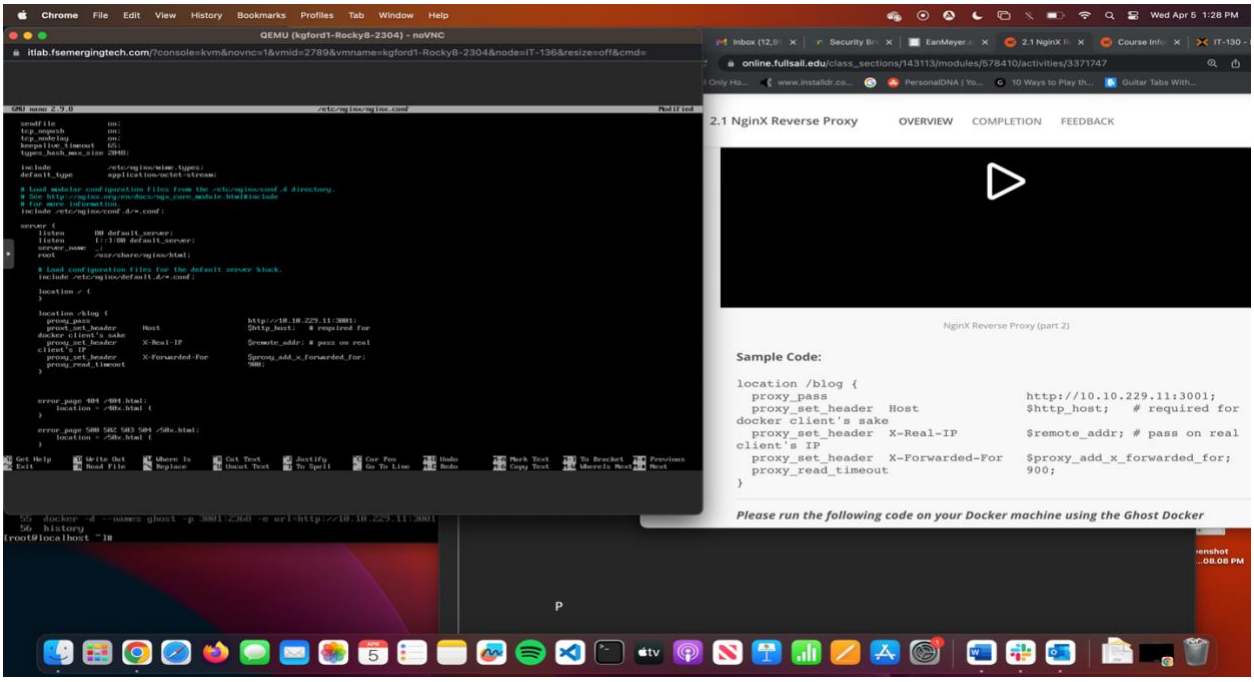
- Conduct a security audit to find any possible system vulnerabilities.
- Vulnerabilities should be prioritized according to their seriousness and likelihood of being exploited.
- Put security measures in place: Security measures should be implemented to lessen the risk of an attack.
- Monitor the system: The system needs to be monitored for signs of an attack.

Full Report Conclusion

This project covered a variety of topics, including setting up a custom network, installing Ghost on Docker, and setting up WordPress on Ubuntu. I discovered that Nginx is used for reverse proxying and serving web content. The article offered detailed instructions for setting up Nginx, starting it, and adding reverse proxy logs for the Ghost site. Additionally, the project covered the installation and testing of Ghost on Docker as well as the LAMP stack installation of WordPress on Ubuntu. The project was informative overall and gave a good overview of each of the subjects covered.

Appendix A

NginX Config File



Appendix B

NginX Access Log File