#!/bin/bash

# Update package lists

apt-get update -y

echo "Removing old Services"

# Remove old services

apt-get remove -y apache2

apt-get remove -y libopendkim\*

apt-get remove -y opendkim

apt-get remove -y postfix

apt-get remove -y php\*

apt-get remove -y postgresql\*

rm -rf /etc/apache2

rm -rf /etc/opendkim\*

systemctl stop sendmail.service

echo "Stopping firewall and selinux"

# Disable SELinux and stop firewall

setenforce 0

sed -i 's/^SELINUX=.\*/SELINUX=disabled/' /etc/selinux/config

ufw disable

echo "Installing Main services..."

# Install main services

apt-get install -y openssh-client

apt-get install -y libc6:i386

apt-get install -y libpam0g:i386

apt-get install -y nano

apt-get install -y rsync

apt-get install -y wget

apt-get install -y xinetd

apt-get install -y gcc

apt-get install -y make

apt-get install -y apache2

apt-get install -y perl

apt-get install -y libapache2-mod-ssl

apt-get install -y zip

apt-get install -y unzip

apt-get install -y cron

apt-get install -y php

apt-get install -y php-pgsql

apt-get install -y php-mysql

apt-get install -y php-common

apt-get install -y php-pdo

apt-get install -y php-opcache

apt-get install -y php-mcrypt

apt-get install -y php-imap

apt-get install -y php-mbstring

apt-get install -y php-soap

apt-get install -y php-xmlrpc

apt-get install -y cron

apt-get install -y php-ssh2

apt-get update -y

echo "Enabling php 7.0"

# Enable PHP 7.0

sudo add-apt-repository ppa:ondrej/php

sudo apt-get update -y

sudo apt-get install -y php7.0

sudo apt-get install -y php7.0-pgsql

# Continue with the rest of the script...

echo "Installing remi and epel repositories..."

# Add remi and epel repositories

apt-get install -y software-properties-common

add-apt-repository ppa:remi/php

apt-get update -y

echo "Installing PHP modules..."

# Install PHP modules

apt-get install -y php-pgsql

apt-get install -y php-mysql

apt-get install -y php-common

apt-get install -y php-pdo

apt-get install -y php-opcache

apt-get install -y php-mcrypt

apt-get install -y php-imap

apt-get install -y php-mbstring

apt-get install -y php-soap

apt-get install -y php-xmlrpc

echo "Updating ca-certificates..."

apt-get update -y

echo "Modifying server info..."

# Modify PHP configuration

sed -i 's/upload\_max\_filesize = 2M/upload\_max\_filesize = 5G/g' /etc/php/7.0/apache2/php.ini

sed -i 's/max\_file\_uploads = 20/max\_file\_uploads = 200/g' /etc/php/7.0/apache2/php.ini

sed -i 's/post\_max\_size = 8M/post\_max\_size = 5G/g' /etc/php/7.0/apache2/php.ini

sed -i 's/memory\_limit = 128M/memory\_limit = -1/g' /etc/php/7.0/apache2/php.ini

sed -i 's/max\_input\_time = 60/max\_input\_time = 3600/g' /etc/php/7.0/apache2/php.ini

sed -i 's/;max\_input\_nesting\_level = 64/max\_input\_nesting\_level = 10000/g' /etc/php/7.0/apache2/php.ini

sed -i 's/; max\_input\_vars = 1000/max\_input\_vars = 100000/g' /etc/php/7.0/apache2/php.ini

sed -i 's/default\_socket\_timeout = 60/default\_socket\_timeout = 360/g' /etc/php/7.0/apache2/php.ini

sed -i 's/max\_execution\_time = 30/max\_execution\_time = 3600/g' /etc/php/7.0/apache2/php.ini

echo "Restarting services..."

# Restart Apache

systemctl restart apache2.service

systemctl enable apache2.service

echo "Installing Composer..."

# Install Composer

curl -sS <https://getcomposer.org/installer> | php

mv composer.phar /usr/local/bin/composer

echo "Installing PostgreSQL..."

# Install PostgreSQL

apt-get install -y postgresql

echo "Configuring HTTP app..."

# Edit the Apache configuration

sed -i 's/#NameVirtualHost \\*:80/NameVirtualHost \*:80/g' /etc/apache2/conf-available/httpd.conf

# Edit the gm.conf

cat <<EOL > /etc/apache2/sites-available/gm.conf

<VirtualHost \*:80>

ServerName 176.126.87.210

DocumentRoot '/usr/gm/public/'

<Directory /usr/gm/public/>

AllowOverride all

Options Indexes FollowSymLinks

Order Deny,Allow

Require all granted

</Directory>

</VirtualHost>

EOL

# Enable the gm.conf site

a2ensite gm.conf

# Change ownership of directories

chown -R www-data:www-data /usr/gm

chown -R www-data:www-data /usr/gm/storage/logs

chown -R www-data:www-data /usr/gm/storage/\*

chown -R www-data:www-data /usr/gm/public/\*

# Restart Apache

systemctl restart apache2.service

# Install additional packages

apt-get install -y firewalld

systemctl start firewalld

systemctl enable firewalld

firewall-cmd --zone=public --permanent --add-service=http

firewall-cmd --zone=public --permanent --add-port=5432/tcp

firewall-cmd --reload

# Additional configurations...

echo "Installing phpPgAdmin..."

# Install phpPgAdmin

apt-get install -y phpmyadmin

# Configure phpPgAdmin

cat <<EOL > /etc/apache2/conf-available/phppgadmin.conf

Alias /phppgadmin /usr/share/phppgadmin

<Directory /usr/share/phppgadmin>

AllowOverride all

Require all granted

</Directory>

EOL

# Enable phpPgAdmin configuration

a2enconf phppgadmin

# Restart Apache

systemctl restart apache2.service

echo "Configuring PostgreSQL..."

# Continue with your PostgreSQL configuration steps...

# ...

# Download and unzip iresponse.zip

cd /usr/

wget <https://chantorhost.live/iresponse.zip>

unzip iresponse.zip

mv iresponse gm

# Copy PostgreSQL configuration files

rm -rf /var/lib/postgresql/13/main/pg\_hba.conf

cp /usr/gm/install/pg\_hba\_trust.conf /var/lib/postgresql/13/main/pg\_hba.conf

rm -rf /var/lib/postgresql/13/main/postgresql.conf

cp /usr/gm/install/postgresql.conf /var/lib/postgresql/13/main/postgresql.conf

# Restart PostgreSQL

systemctl restart postgresql-13.service

systemctl enable postgresql-13.service

# Additional configurations...

echo "Installing Java..."

# Download and install Java

cd /opt/

wget --no-cookies --no-check-certificate --header "Cookie: gpw\_e24=http%3A%2F%2Fwww.oracle.com%2F; oraclelicense=accept-securebackup-cookie" "<https://github.com/frekele/oracle-java/releases/download/8u212-b10/jdk-8u212-linux-x64.tar.gz>"

tar -xvf jdk-8u212-linux-x64.tar.gz

cd /opt/jdk1.8.0\_212/

# Configure alternatives

update-alternatives --install /usr/bin/java java /opt/jdk1.8.0\_212/bin/java 2

update-alternatives --config java

# Set environment variables

export JAVA\_HOME=/opt/jdk1.8.0\_212

export JRE\_HOME=/opt/jdk1.8.0\_212/jre

export PATH=$PATH:/opt/jdk1.8.0\_212/bin:/opt/jdk1.8.0\_212/jre/bin

export PATH=$JAVA\_HOME/bin:$PATH

# Configure Java permissions

echo 'apache ALL = NOPASSWD: /opt/jdk1.8.0\_212/bin/java' | sudo EDITOR='tee -a' visudo

# Restart Apache

systemctl restart apache2.service

# Continue with the rest of your script...

echo "Configuring HTTP app..."

# Edit Apache configuration

sed -i 's/#NameVirtualHost \*:80/NameVirtualHost \*:80/g' /etc/apache2/conf-available/httpd.conf

# Edit the gm.conf

cat <<EOL > /etc/apache2/sites-available/gm.conf

<VirtualHost \*:80>

ServerName 176.126.87.210

DocumentRoot '/usr/gm/public/'

<Directory /usr/gm/public/>

AllowOverride all

Options Indexes FollowSymLinks

Order Deny,Allow

Require all granted

</Directory>

</VirtualHost>

EOL

# Enable the gm.conf site

a2ensite gm.conf

# Change ownership of directories

chown -R www-data:www-data /usr/gm

chown -R www-data:www-data /usr/gm/storage/logs

chown -R www-data:www-data /usr/gm/storage/\*

chown -R www-data:www-data /usr/gm/public/\*

# Restart Apache

systemctl restart apache2.service

# Install additional packages

apt-get install -y firewalld

systemctl start firewalld

systemctl enable firewalld

firewall-cmd --zone=public --permanent --add-service=http

firewall-cmd --zone=public --permanent --add-port=5432/tcp

firewall-cmd --reload

# Configure additional settings...

echo "Installing phpPgAdmin..."

# Install phpPgAdmin

apt-get install -y phppgadmin

# Configure phpPgAdmin

cat <<EOL > /etc/apache2/conf-available/phppgadmin.conf

Alias /phppgadmin /usr/share/phppgadmin

<Directory /usr/share/phppgadmin>

AllowOverride all

Require all granted

</Directory>

EOL

# Enable phpPgAdmin configuration

a2enconf phppgadmin

# Restart Apache

systemctl restart apache2.service

# Configure PostgreSQL...

# Continue with your PostgreSQL configuration steps...

# ...

# Download and unzip iresponse.zip

cd /usr/

wget <https://chantorhost.live/iresponse.zip>

unzip iresponse.zip

mv iresponse gm

# Copy PostgreSQL configuration files

rm -rf /var/lib/postgresql/13/main/pg\_hba.conf

cp /usr/gm/install/pg\_hba\_trust.conf /var/lib/postgresql/13/main/pg\_hba.conf

rm -rf /var/lib/postgresql/13/main/postgresql.conf

cp /usr/gm/install/postgresql.conf /var/lib/postgresql/13/main/postgresql.conf

# Restart PostgreSQL

systemctl restart postgresql-13.service

systemctl enable postgresql-13.service

# Configure Java...

# Continue with your Java configuration steps...

# ...

# Restart Apache

systemctl restart apache2.service

# Continue with the rest of your script...

echo "Uploading iresponse.jar to /usr/"

cd /usr/

wget <https://chantorhost.live/iresponse.jar>

# Create a backup of existing iresponse.jar (if exists)

mv /usr/gm/app/api/lib/iresponse.jar /usr/gm/app/api/lib/iresponse.jar\_backup

# Move the new iresponse.jar

mv iresponse.jar /usr/gm/app/api/lib/

# Change ownership

chown -R www-data:www-data /usr/gm/app/api/lib/

# Restart Apache

systemctl restart apache2.service

echo "Disabling SELinux..."

sudo setenforce 0

sudo sed -i 's/^SELINUX=.\*/SELINUX=disabled/' /etc/selinux/config

# Check SELinux status

sestatus

echo "Configuring cron jobs..."

# Edit the crontab

crontab -e

# Add your desired cron jobs

# Example:

# 0 \* \* \* \* java -Dfile.encoding=UTF8 -jar /usr/gm/app/api/iresponse\_services.jar ...

# Save and exit the crontab editor

echo "All configurations completed successfully!"