

```
#!/bin/sh
#puppetServerInstall.sh

# 1. Installs a puppet server installation on ubuntu OS. There is an optional check
#    for K8s that is
#    lab specific to Simplilearn.
# 2. Creates a wordpress.conf file for initializing Apache web server.
# 3. Creates wp-config.php file. Note that this script will ask create a wordpress u
ser and password.

#NOTE: This script is not secure, don't leave this script on your server! Use at you
r own risk. :D
# Be aware that most of this script runs as a normal user.
# There will be instructions to switch to root when certifications are being set up
using
# puppetserver command.

echo "Do you want to set up host and client entries in /etc/hosts?"
read yesNo

if [ $yesNo = 'y' -o $yesNo = 'Y' ]
then
    echo "Enter ip address of client1: "
    read ipAddr

    echo "Setting up hosts..."
    sudo hostnamectl set-hostname puppetserver.ec2.internal
    sudo -- sh -c "echo '$(hostname -i | cut -d' ' -f1) puppetserver.ec2.internal puppe
t' >> /etc/hosts"
    sudo -- sh -c "echo '$ipAddr client1.ec2.internal' >> /etc/hosts"
fi

echo "Do you want to download and install puppetserver?"
read yesNo

if [ $yesNo = 'y' -o $yesNo = 'Y' ]
then
    echo "Puppetserver install..."
    curl -O https://apt.puppetlabs.com/puppet7-release-xenial.deb
    sudo dpkg -i puppet7-release-xenial.deb
    sudo apt-get update

    sudo apt-get -y install puppetserver
fi

echo "Is this K8s and need to clean up the lab?"
read yesNo

if [ $yesNo = 'y' -o $yesNo = 'Y' ]
then
    #--
    #FOR k8s issues
    #---
    sudo apt-get update --fix-missing
    sudo apt-get remove puppet-agent puppet-master puppet puppetserver puppet-agent
    sudo apt --fix-broken install
    sudo apt clean
    sudo rm -rf /tmp/*
    sudo apt install puppetserver
fi

echo "Do you want to set up java vm 512m line (Simplilearn AMI labs)?"
read yesNo

if [ $yesNo = 'y' -o $yesNo = 'Y' ]
then
    echo "Java vm memory adjustment..."
    ##### JAVA 512M BEGIN #####
    sed -E "s/(JAVA_ARGS=\"-Xms)(\\w+)( -Xmx)(\\w+)/JAVA_ARGS=\"-Xms512m -Xmx512m/g" /et
c/default/puppetserver > tempPuppetServer.txt
    sudo mv tempPuppetServer.txt /etc/default/puppetserver
    # Note: you can try using the automated sed method but double check the file for c
orrectness!

```

```

# -OR- manually edit:
#     /etc/default/puppetserver
# change the JAVA_ARGS variable to:
# change JAVA_ARGS="-Xms512m -Xmx512m
##### JAVA 512M END #####
fi

echo "Do you want to (re)start puppetserver?"
read yesNo

if [ $yesNo = 'y' -o $yesNo = 'Y' ]
then
    echo "Starting puppetserver..."
    sudo systemctl restart puppetserver
fi

echo "Do you want to wipe out previous certs and recreate? (y/n)"
read certFlg

if [ $certFlg = 'y' -o $certFlg = 'Y' ]
then
    echo "setting up puppetserver certs..."

    sudo rm /etc/puppetlabs/puppetserver/ca -rvf
    sudo rm /etc/puppetlabs/puppet/ssl -rvf

    set certMessageShow = 'y'
else
    set certMessageShow = 'n'
    echo "SKIPPING cert setup..."
fi

echo "Create wordpress.conf and wp-config.php files?"
read createWordConfigFile

if [ $createWordConfigFile = 'y' -o $createWordConfigFile = 'Y' ]
then

    echo "What is the password for the wordpress DB user?"
    read wordpress_password

    echo "Creating wordpress config.file on local user dir: wordpress_configfiles..."
    mkdir wordpress_configfiles

    cat <<EOT >> wordpress_configfiles/wordpress.conf
<VirtualHost *:80>
    DocumentRoot /srv/www/wordpress
    <Directory /srv/www/wordpress>
        Options FollowSymLinks
        AllowOverride Limit Options FileInfo
        DirectoryIndex index.php
        Require all granted
    </Directory>
    <Directory /srv/www/wordpress/wp-content>
        Options FollowSymLinks
        Require all granted
    </Directory>
</VirtualHost>
EOT

    cat <<EOT >> wordpress_configfiles/wp-config.php
<?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the installation.
 * You don't have to use the web site, you can copy this file to "wp-config.php"
 * and fill in the values.
 *
 * This file contains the following configurations:
 *
 * * MySQL settings
 * * Secret keys

```

```

* * Database table prefix
* * ABSPATH
*
.* @link https://wordpress.org/support/article/editing-wp-config-php/
.*
.* @package WordPress
.*

// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'wordpress' );

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

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

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

/** Database charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );

/** The database collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );

/**#@+
 * Authentication unique keys and salts.
 *
 * Change these to different unique phrases! You can generate these using
 * the {@link https://api.wordpress.org/secret-key/1.1/salt/ WordPress.org secret-key
 * service}.
 *
 * You can change these at any point in time to invalidate all existing cookies.
 * This will force all users to have to log in again.
 *
 * @since 2.6.0
 */
/* update these values from this web site:
   https://api.wordpress.org/secret-key/1.1/salt/
 */
define( 'AUTH_KEY',          'put your unique phrase here' );
define( 'SECURE_AUTH_KEY',   'put your unique phrase here' );
define( 'LOGGED_IN_KEY',     'put your unique phrase here' );
define( 'NONCE_KEY',         'put your unique phrase here' );
define( 'AUTH_SALT',         'put your unique phrase here' );
define( 'SECURE_AUTH_SALT',   'put your unique phrase here' );
define( 'LOGGED_IN_SALT',    'put your unique phrase here' );
define( 'NONCE_SALT',        'put your unique phrase here' );

/**#@-*/

/**
 * WordPress database table prefix.
 *
 * You can have multiple installations in one database if you give each
 * a unique prefix. Only numbers, letters, and underscores please!
 */
$table_prefix = 'wp_';

/**
 * For developers: WordPress debugging mode.
 *
 * Change this to true to enable the display of notices during development.
 * It is strongly recommended that plugin and theme developers use WP_DEBUG
 * in their development environments.
 *
 * For information on other constants that can be used for debugging,
 * visit the documentation.
 *
 * @link https://wordpress.org/support/article/debugging-in-wordpress/

```

```

*/
define( 'WP_DEBUG', false );

/* Add any custom values between this line and the "stop editing" line. */

/* That's all, stop editing! Happy publishing. */

/** Absolute path to the WordPress directory. */
if ( ! defined( 'ABSPATH' ) ) {
    define( 'ABSPATH', __DIR__ . '/' );
}

/** Sets up WordPress vars and included files. */
require_once ABSPATH . 'wp-settings.php';

EOT

fi

echo "Create MYSQL file?"
read createMySQLFile

if [ $createMySQLFile = 'y' -o $createMySQLFile = 'Y' ]
then

    if [ -n "$wordpress_password" ]
    then
        echo "What is the password for the wordpress DB user?"
        read wordpress_password
    fi

    cat <<EOT >> wordpress_configfiles/wordpressMySQL.sql
CREATE DATABASE wordpress;
CREATE USER wordpress@localhost IDENTIFIED BY '$wordpress_password';
GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP,ALTER
    ON wordpress.*
    TO wordpress@localhost;

FLUSH PRIVILEGES;
quit
EOT

fi

if [ $certFlg = 'y' -o $certFlg = 'Y' ]
then
    echo "====="
    echo "The rest of the steps *you* must do as root as follows:"
    echo "====="
    echo "sudo -i"
    echo "/opt/puppetlabs/server/apps/puppetserver/bin/puppetserver ca setup"

    echo "/opt/puppetlabs/bin/puppetserver ca list --all\n"
    #echo "/opt/puppetlabs/server/apps/puppetserver/bin/puppetserver ca list --all\n"

    echo "Give the certs a moment to sync then run above to check out the certs as *ro
ot*"
fi

echo "Done."
exit

```