

WordPress GitHub Workflow

Introduction

GitHub Repo Versioning

Configuring a Codeanywhere WordPress site for
storage (versioning) on a GitHub Repository

Goal of this Presentation

- Act as a step by step guide to configuring a WordPress site for storing (saving) on a **PUBLIC** GitHub Repository (REPO)
- Because the GitHub account we are using is **PUBLIC**, we have to make changes to our WordPress site and our WordPress MySQL database to **protect** them from being checked out and used **except** by vetted **developers** and **approved** project team members

High-level Overview

1. Create a new WordPress site
2. Edit the `.htaccess` and `wp-config.php` to fix the 8 MB max file upload issue
3. Add a new Admin user to the site – **this is the WP admin user account you can share with developers**
4. Harden the WordPress Database for **Public** GitHub storage
 - Change the MySQL Database root login from being passwordless to having a **strong password**
5. **Create a new file named `composer.json`**
6. **Run the `composer` command on the SSH Terminal**
7. Override the WordPress MySQL DB options table **home** and **siteurl** settings – so that our WordPress can run on different platforms and environments

High-level Overview

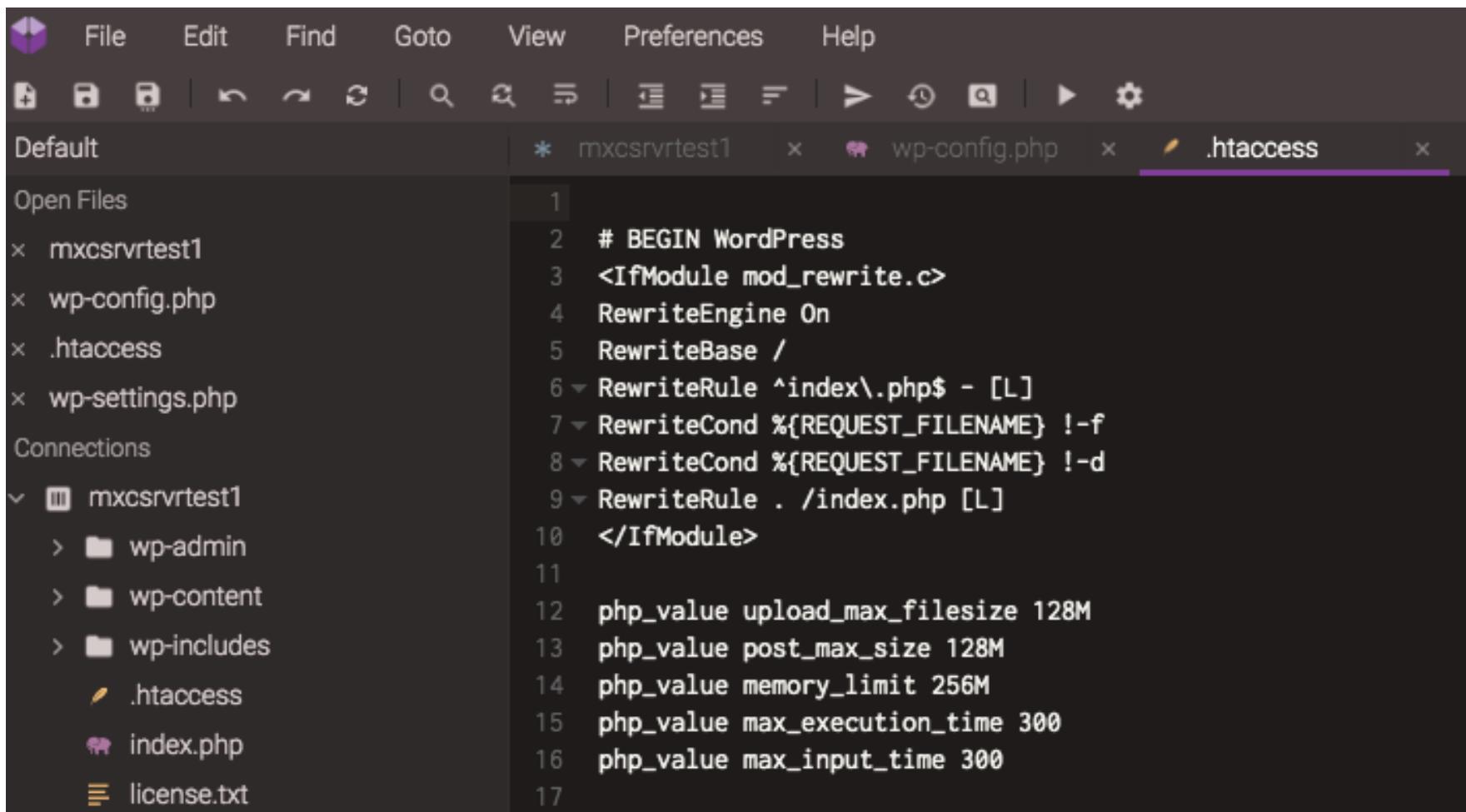
8. Create an encrypted backup of our WordPress Database for storage on PUBLIC GitHub repository
9. Create the GitHub repository and populate it with the WordPress site files and encrypted database backup

(1) Create a WordPress CMS Website

- Create a WordPress CMS Website
 - Create a menu
 - Add some pictures to the media library
 - Create some categories and tags
 - Create a few Pages
 - Create a few Posts, using Categories and Tags

(2) Fix the 8 MB Max File Upload Issue

.htaccess file mods



The screenshot shows a file manager window with the following structure:

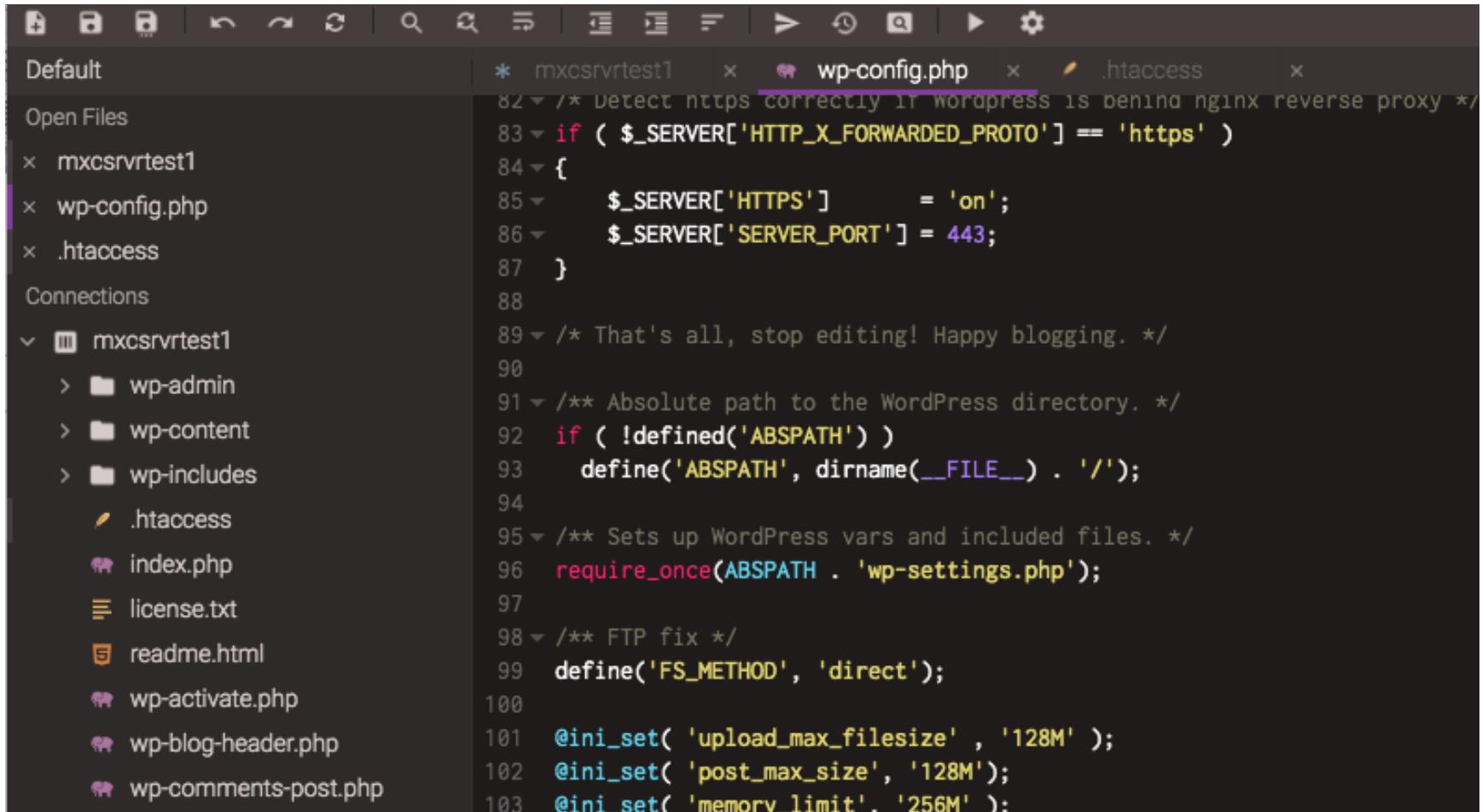
- Default**: mxcsrvrtest1, wp-config.php, .htaccess
- Open Files**: mxcsrvrtest1, wp-config.php, .htaccess, wp-settings.php
- Connections**: mxcsrvrtest1 (expanded): wp-admin, wp-content, wp-includes, .htaccess, index.php, license.txt

The .htaccess file content is as follows:

```
1
2 # BEGIN WordPress
3 <IfModule mod_rewrite.c>
4 RewriteEngine On
5 RewriteBase /
6 RewriteRule ^index\.php$ - [L]
7 RewriteCond %{REQUEST_FILENAME} !-f
8 RewriteCond %{REQUEST_FILENAME} !-d
9 RewriteRule . /index.php [L]
10 </IfModule>
11
12 php_value upload_max_filesize 128M
13 php_value post_max_size 128M
14 php_value memory_limit 256M
15 php_value max_execution_time 300
16 php_value max_input_time 300
17
```

(2) Fix the 8 MB Max File Upload Issue

wp-config.php file mods



The screenshot shows a file manager interface with a sidebar on the left and a code editor on the right. The sidebar lists 'Default', 'Open Files' (wp-config.php), and 'Connections'. Under 'mxcsrvrtest1', there are files: wp-admin, wp-content, wp-includes, .htaccess, index.php, license.txt, readme.html, wp-activate.php, wp-blog-header.php, and wp-comments-post.php. The wp-config.php file is selected in the sidebar and is displayed in the code editor.

```
*  mxcsrvrtest1  x  wp-config.php  x  .htaccess  x
82  /* Detect https correctly if Wordpress is behind nginx reverse proxy */
83  if ( $_SERVER['HTTP_X_FORWARDED_PROTO'] == 'https' )
84  {
85      $_SERVER['HTTPS']      = 'on';
86      $_SERVER['SERVER_PORT'] = 443;
87  }
88
89  /* That's all, stop editing! Happy blogging. */
90
91  /** Absolute path to the WordPress directory. */
92  if ( !defined('ABSPATH') )
93      define('ABSPATH', dirname(__FILE__) . '/');
94
95  /** Sets up WordPress vars and included files. */
96  require_once(ABSPATH . 'wp-settings.php');
97
98  /** FTP fix */
99  define('FS_METHOD', 'direct');
100
101 @ini_set( 'upload_max_filesize' , '128M' );
102 @ini_set( 'post_max_size' , '128M' );
103 @ini_set( 'memory_limit' , '256M' );
```

(3) Add a new Admin user

Username: userIT320

Password: IT320@Winter2020!

The screenshot shows the 'Add New User' page in the WordPress admin interface. The left sidebar is dark with white text, showing the 'Users' menu item as selected. The main area has a light gray background. The 'Add New User' title is at the top. Below it is a sub-instruction: 'Create a brand new user and add them to this site.' The form fields are as follows:

- Username (required)**: userIT320
- Email (required)**: mikehchase@gmail.com (highlighted with a red circle containing the number 1 and a validation message: 'You must use a different email than the original Admin user')
- First Name**: user
- Last Name**: IT320
- Website**: (empty field)
- Password**: Winter202@IT320! (highlighted with a green bar and the word 'Strong')
- Send User Notification**: An unchecked checkbox with the label: 'Send the new user an email about their account.'
- Role**: Administrator

At the bottom left is a blue 'Add New User' button. To the right of the password field are two buttons: 'Hide' and 'Cancel'.

Or use a made up email and do not send user notification

The screenshot shows a user interface for adding a new user. On the left, there's a vertical sidebar with a dark theme featuring navigation items like 'Dashboard', 'Users', 'Groups', 'Inbox', 'Logs', and 'menu'. The main area has a light gray background and is titled 'Add New User'. It contains fields for creating a new user account:

- Username (required)**: A text input field containing 'userIT320'.
- Email (required)**: A text input field containing 'xyz@notvaidemail.com'.
- First Name**: A text input field containing 'User'.
- Last Name**: A text input field containing 'IT320'.
- Website**: An empty text input field.
- Password**: A text input field containing 'IT320@Winter2020!' which is highlighted with a green bar indicating it is a strong password.
- Send User Notification**: A checkbox labeled 'Send the new user an email about their account.' which is unchecked.
- Role**: A dropdown menu currently set to 'Administrator'.

At the bottom left is a blue button labeled 'Add New User'.

New User Added

1 0 + New

Screen Options

Users [Add New](#)

New user created. [Edit user](#)

All (2) | Administrator (2)

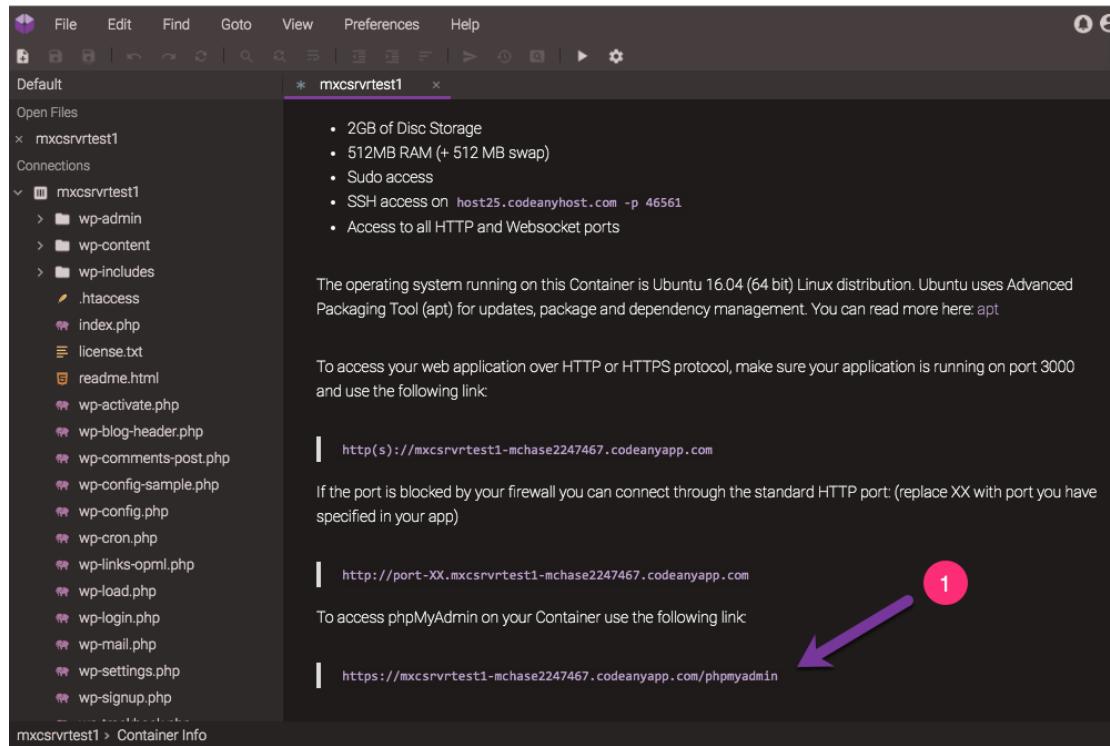
Bulk Actions [Apply](#) Change role to... [Change](#)

<input type="checkbox"/> Username	Name	Email	Role
<input type="checkbox"/>  mxchase	—	mchase2@cdm.depaul.edu	Administrator
<input type="checkbox"/>  userIT320	user IT320	mikehchase@gmail.com	Administrator
<input type="checkbox"/> Username	Name	Email	Role

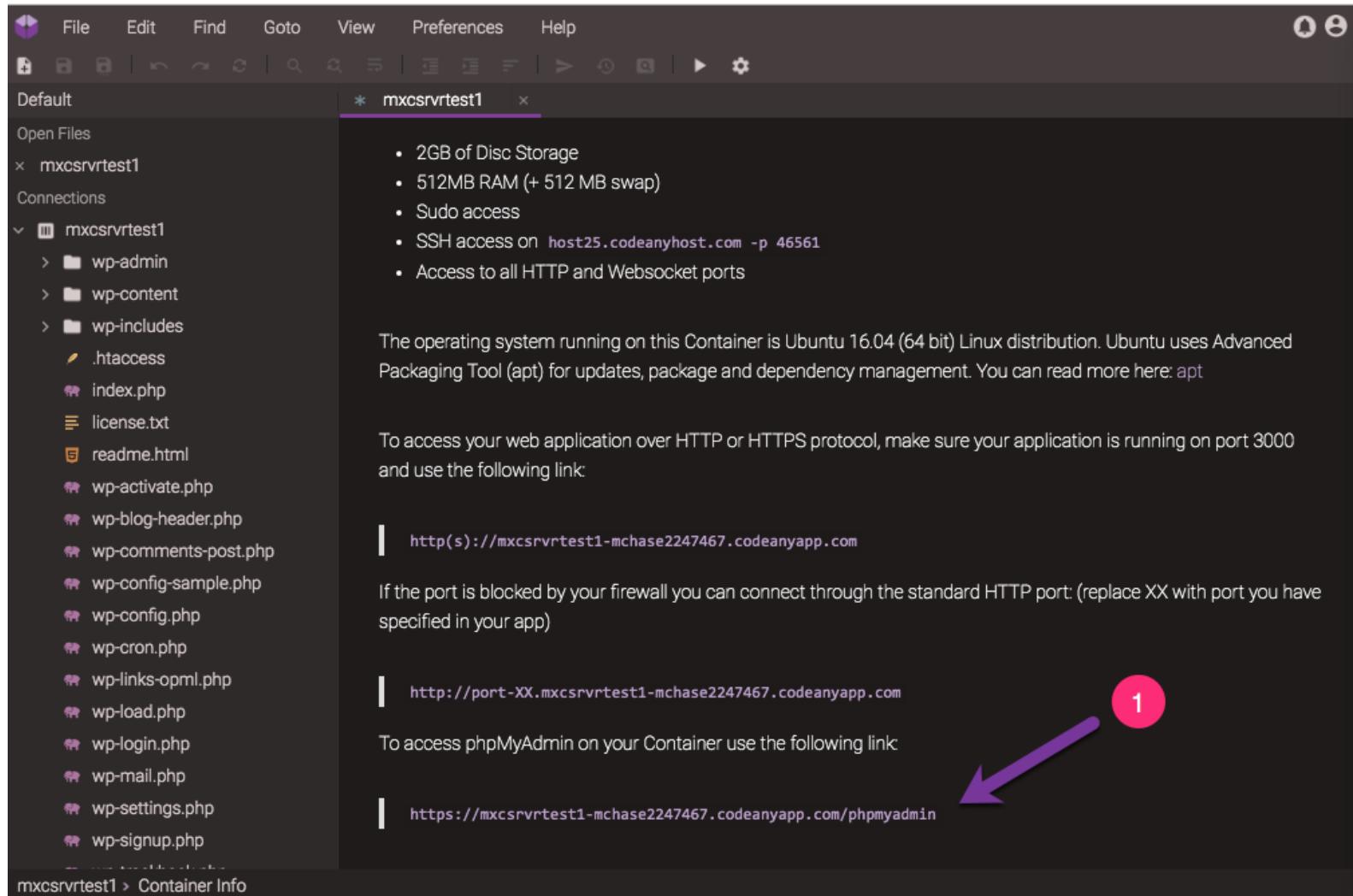
Bulk Actions [Apply](#) Change role to... [Change](#)

(4) Configure MySQL DB to have username and password

1. From the Codeanywhere Info page, access the phpMyAdmin login page
2. Login to phpMyAdmin



(1) Access the phpMyAdmin



The screenshot shows a terminal window with a dark theme. The title bar says "mxcsrvrtest1". The left sidebar lists "Open Files" and "Connections". Under "Connections", "mxcsrvrtest1" is expanded, showing its contents:

- > wp-admin
- > wp-content
- > wp-includes
 - .htaccess
 - index.php
 - license.txt
 - readme.html
 - wp-activate.php
 - wp-blog-header.php
 - wp-comments-post.php
 - wp-config-sample.php
 - wp-config.php
 - wp-cron.php
 - wp-links-opml.php
 - wp-load.php
 - wp-login.php
 - wp-mail.php
 - wp-settings.php
 - wp-signup.php

The main pane displays system specifications and operational details:

- 2GB of Disc Storage
- 512MB RAM (+ 512 MB swap)
- Sudo access
- SSH access on <host25.codeanyhost.com -p 46561>
- Access to all HTTP and Websocket ports

The operating system running on this Container is Ubuntu 16.04 (64 bit) Linux distribution. Ubuntu uses Advanced Packaging Tool (apt) for updates, package and dependency management. You can read more here: [apt](#)

To access your web application over HTTP or HTTPS protocol, make sure your application is running on port 3000 and use the following link:

```
http(s)://mxcsrvrtest1-mchase2247467.codeanyapp.com
```

If the port is blocked by your firewall you can connect through the standard HTTP port: (replace XX with port you have specified in your app)

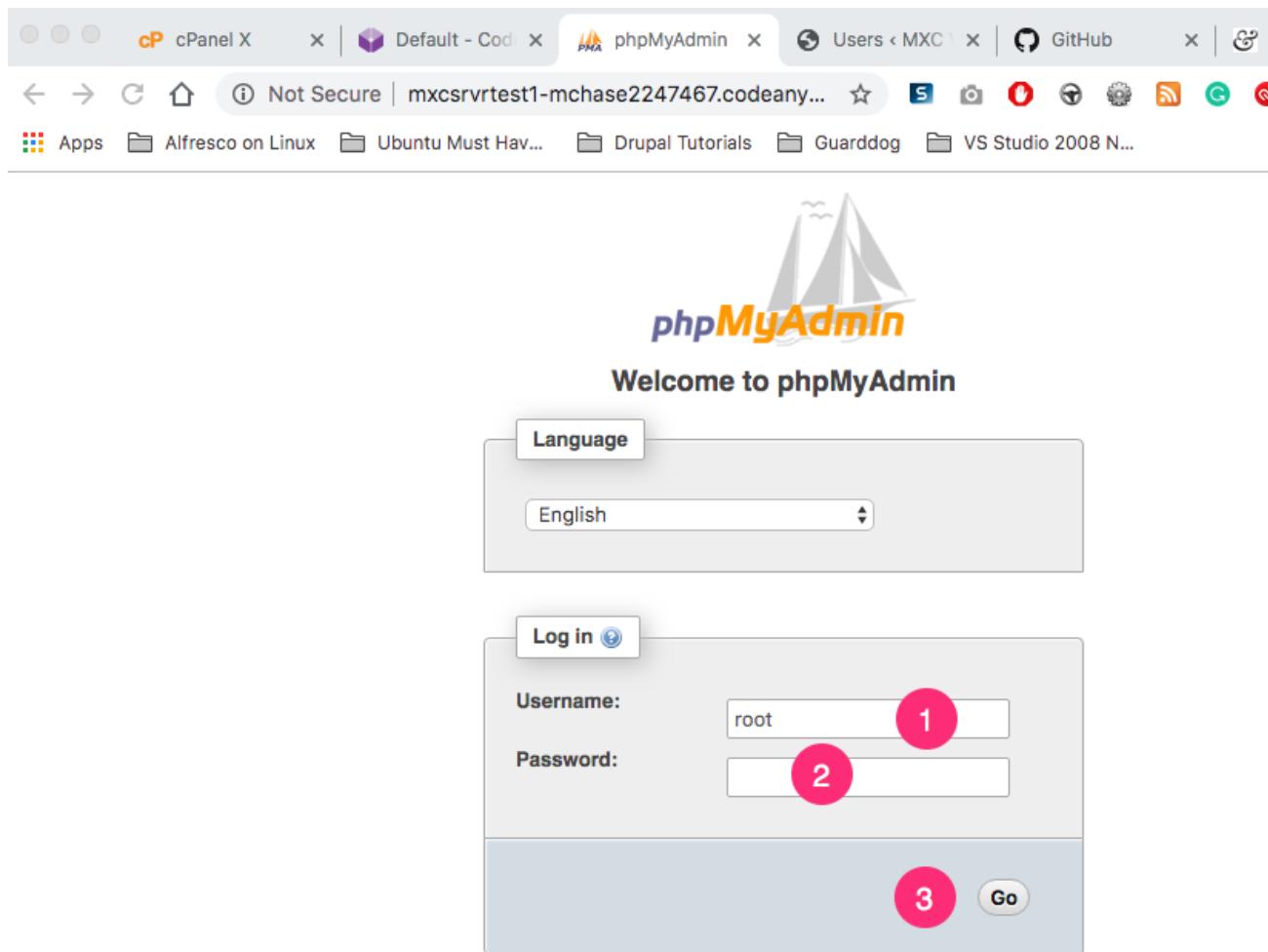
```
http://port-XX.mxcsrvrtest1-mchase2247467.codeanyapp.com
```

To access phpMyAdmin on your Container use the following link:

```
https://mxcsrvrtest1-mchase2247467.codeanyapp.com/phpmyadmin
```

A large purple arrow points from the number "1" in a pink circle to the "phpmyadmin" link.

(2) Login as root, passwordless



(1) Select Home

(2) Select User accounts

The screenshot shows the phpMyAdmin interface for a MySQL database on localhost. A red circle labeled '1' highlights the 'User accounts' tab in the top navigation bar. Another red circle labeled '2' highlights the 'User accounts' link in the main content area, which is titled 'User accounts overview'. The left sidebar lists various databases, with 'wordpress' selected. The main content area displays a table of user accounts:

User name	Host name	Password	Global privileges	Grant	Action
debian-sys-maint	localhost	Yes	ALL PRIVILEGES	Yes	Edit privileges Export
mysql.session	localhost	Yes	SUPER	No	Edit privileges Export
mysql.sys	localhost	Yes	USAGE	No	Edit privileges Export
root	localhost	No	ALL PRIVILEGES	Yes	Edit privileges Export

Below the table, there are buttons for 'Check all' and 'With selected: Export'. At the bottom, there are buttons for 'New' (with 'Add user account') and 'Remove selected user accounts'.

(3) Edit privileges of the root account

The screenshot shows the phpMyAdmin interface for managing MySQL user accounts. The main title is "User accounts overview". The table lists four users:

User name	Host name	Password	Global privileges	Grant	Action
debian-sys-maint	localhost	Yes	ALL PRIVILEGES	Yes	Edit privileges Export
mysql.session	localhost	Yes	SUPER	No	Edit privileges Export
mysql.sys	localhost	Yes	USAGE	No	Edit privileges Export
root	localhost	No	ALL PRIVILEGES	Yes	Edit privileges Export

A purple arrow points to the "Edit privileges" link for the "root" user, which is highlighted with a red circle containing the number 3.

Below the table, there are buttons for "New" (Add user account) and "Remove selected user accounts". A note says: "(Revoke all active privileges from the users and delete them afterwards.)" and includes a checkbox for "Drop the databases that have the same names as the users."

(4) Select Change password

The screenshot shows the MySQL Workbench interface with the following details:

- Server:** localhost
- Tab Bar:** Databases, SQL, Status, User accounts, Export, Import, Settings, More.
- Sub-Tab Bar:** Global, Database, Change password (highlighted), Login Information.
- Title:** Edit privileges: User account 'root'@'localhost'
- Note:** Note: You are attempting to edit privileges of the user with which you are currently logged in.
- Global privileges:** Check all (checkbox checked).
- Data:** SELECT, INSERT (checkboxes checked).
- Structure:** CREATE, ALTER (checkboxes checked).
- Administration:** GRANT, SUPER (checkboxes checked).

A purple arrow points from the text "Select Change password" in the slide title to the "Change password" tab in the interface. A red circle with the number "1" is placed over the note about editing privileges.

Change the Password

Record your new DB Password!

The screenshot shows the 'User accounts' section of the phpMyAdmin interface for the 'root' user at 'localhost'. The 'Change password' tab is selected. A note at the top states: 'Note: You are attempting to edit privileges of the user with which you are currently logged in.' Three numbered circles highlight specific fields: circle 1 points to the 'Password:' input field, circle 2 points to the 'Re-type:' input field, and circle 3 points to the 'Go' button.

Server: localhost

Databases SQL Status User accounts Export Import Settings More

Global Database Change password Login Information

Edit privileges: User account 'root'@'localhost'

Note: You are attempting to edit privileges of the user with which you are currently logged in.

Change password

No Password

Password: 1 Re-type: 2

Password Hashing: Native MySQL authentication

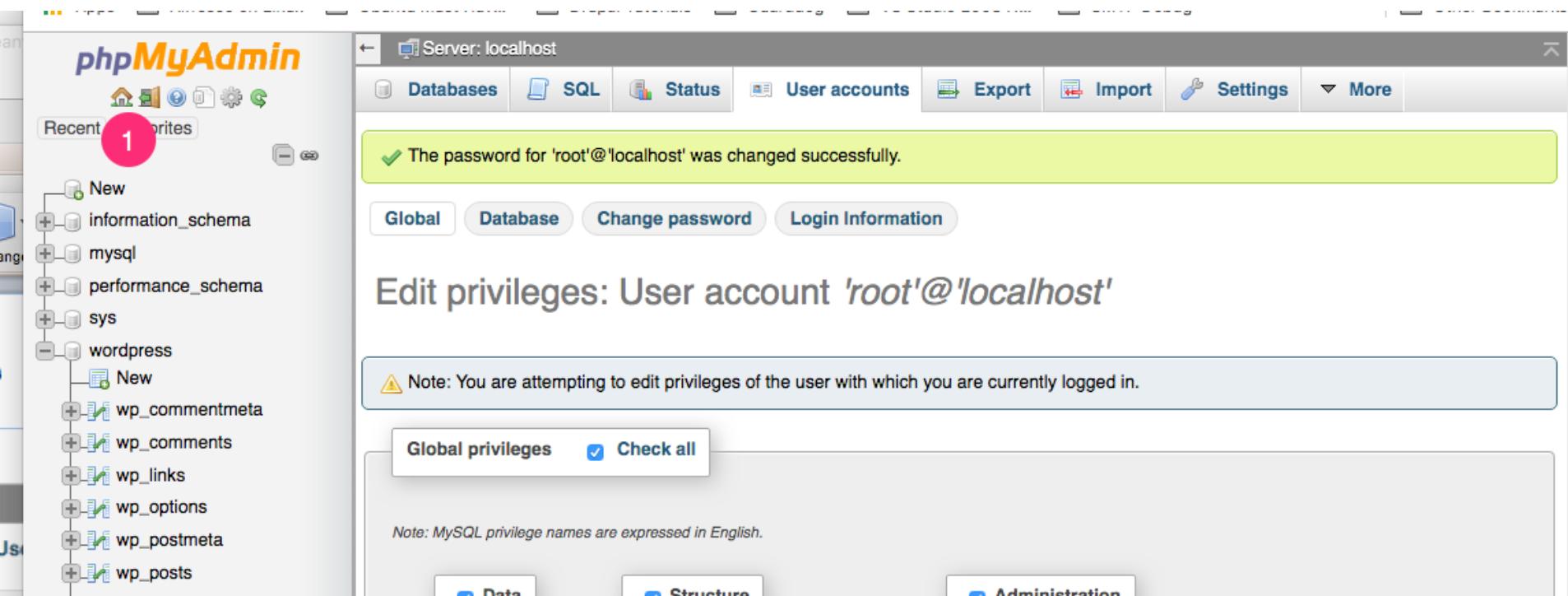
Generate password 3

Go

root password successfully changed

The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with icons for Databases, SQL, Status, User accounts, Export, Import, and Settings. Below the toolbar, a message box displays a green checkmark and the text: "The password for 'root'@'localhost' was changed successfully." Underneath this message, there are four tabs: Global, Database, Change password, and Login Information. The "Change password" tab is currently selected. The main content area has a heading: "Edit privileges: User account 'root'@'localhost'". A note below the heading says: "⚠ Note: You are attempting to edit privileges of the user with which you are currently logged in." There are two buttons: "Global privileges" and "Check all". A note at the bottom states: "Note: MySQL privilege names are expressed in English." At the bottom of the screen, there are three tabs: Data, Structure, and Administration.

Select the Home icon and re-login to phpMyAdmin



The screenshot shows the phpMyAdmin interface for a MySQL server running on localhost. A pink circle highlights the 'Home' icon in the top-left corner of the sidebar. The main content area displays a green success message: 'The password for 'root'@'localhost' was changed successfully.' Below this message, there are tabs for 'Global', 'Database', 'Change password', and 'Login Information'. The 'Change password' tab is active. The title bar indicates the current server is 'localhost'. The sidebar on the left lists databases: 'information_schema', 'mysql', 'performance_schema', 'sys', 'wordpress', and several tables under 'wordpress'. At the bottom of the interface, there are tabs for 'Data', 'Structure', and 'Administration'.

Login with root and new password

The screenshot shows a web browser window with the following details:

- Address Bar:** Secure | mxcsrvrtest1-mchase2247467.codeanyapp.com:300... (highlighted in red)
- Toolbar:** Includes icons for search, refresh, and various browser functions.
- Navigation Bar:** Shows links to "Ubuntu Must Hav...", "Drupal Tutorials", "Guarddog", "VS Studio 2008 N...", and "SMTP Debug".
- Content Area:**
 - phpMyAdmin Logo:** A sailboat icon with the text "php**M**y**A**dmin".
 - Welcome Message:** Welcome to phpMyAdmin
 - Language Selection:** A dropdown menu labeled "Language" set to "English".
 - Login Form:** A large form with:
 - Log in** button with a question mark icon.
 - Username:** Input field containing "root".
 - Password:** Input field containing a series of dots (".....").
 - Go** button at the bottom right of the form.

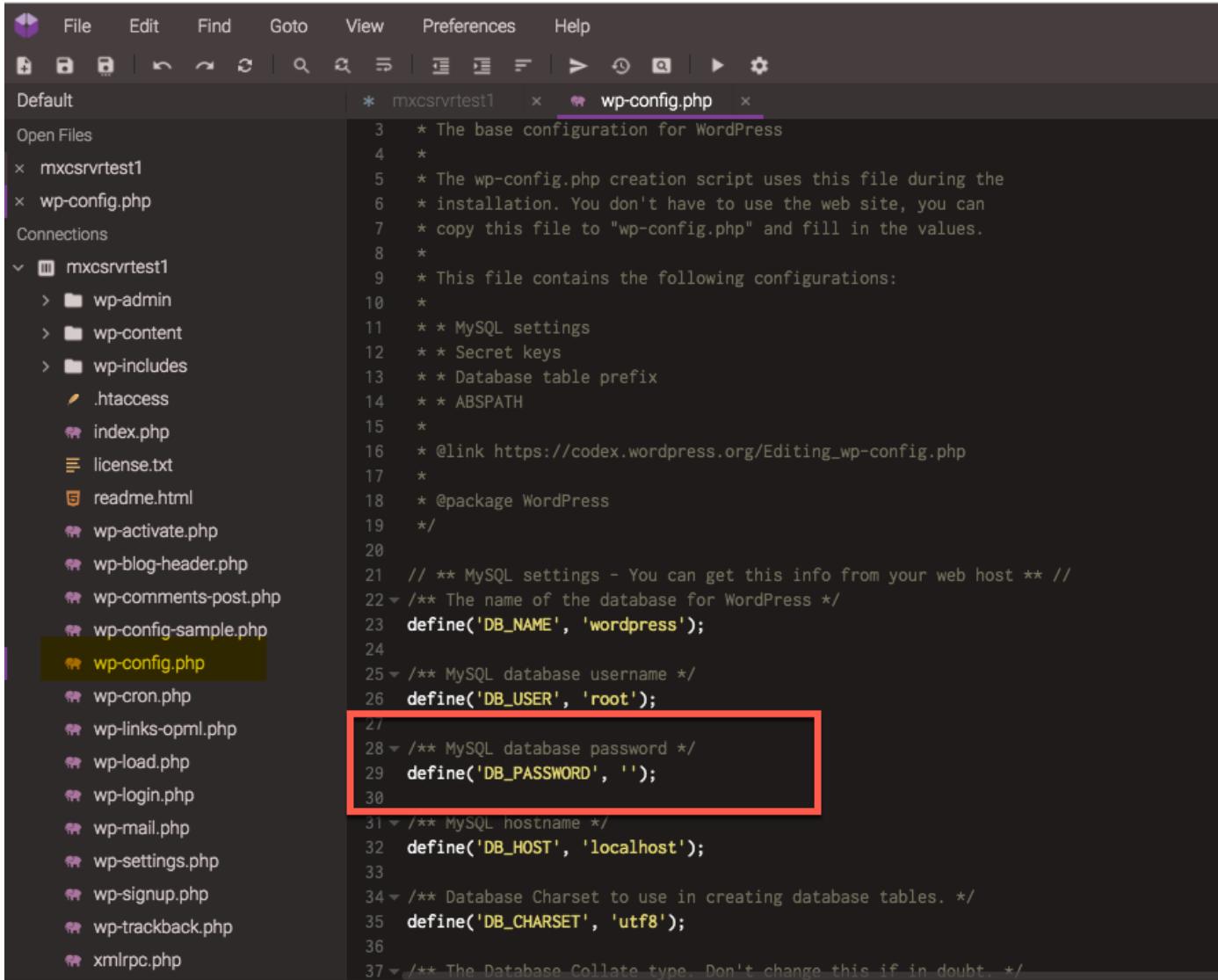
New root password works!

The screenshot shows the phpMyAdmin interface connected to the 'localhost' server and the 'wordpress' database. The left sidebar lists databases: 'information_schema', 'mysql', 'performance_schema', 'sys', and 'wordpress'. The 'wordpress' database is selected, displaying its tables: wp_commentmeta, wp_comments, wp_links, wp_options, wp_postmeta, wp_posts, wp_termmeta, wp_terms, wp_term_relationships, wp_term_taxonomy, wp_usermeta, and wp_users. The 'Structure' tab is active, showing the table names, their row counts (e.g., 153 for wp_options), storage engine (InnoDB), and character set (utf8mb4_unicode_520_ci). A summary at the bottom indicates 12 tables and a total of 284 rows.

Table	Action	Rows	Type	Collation
wp_commentmeta	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_520_ci
wp_comments	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_unicode_520_ci
wp_links	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_520_ci
wp_options	Browse Structure Search Insert Empty Drop	153	InnoDB	utf8mb4_unicode_520_ci
wp_postmeta	Browse Structure Search Insert Empty Drop	45	InnoDB	utf8mb4_unicode_520_ci
wp_posts	Browse Structure Search Insert Empty Drop	31	InnoDB	utf8mb4_unicode_520_ci
wp_termmeta	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_520_ci
wp_terms	Browse Structure Search Insert Empty Drop	6	InnoDB	utf8mb4_unicode_520_ci
wp_term_relationships	Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_unicode_520_ci
wp_term_taxonomy	Browse Structure Search Insert Empty Drop	6	InnoDB	utf8mb4_unicode_520_ci
wp_usermeta	Browse Structure Search Insert Empty Drop	35	InnoDB	utf8mb4_unicode_520_ci
wp_users	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_unicode_520_ci
12 tables	Sum	284	InnoDB	latin1_swedish_ci

Check all With selected:

Edit the WordPress wp-config.php file and add the new phpMyAdmin Database password

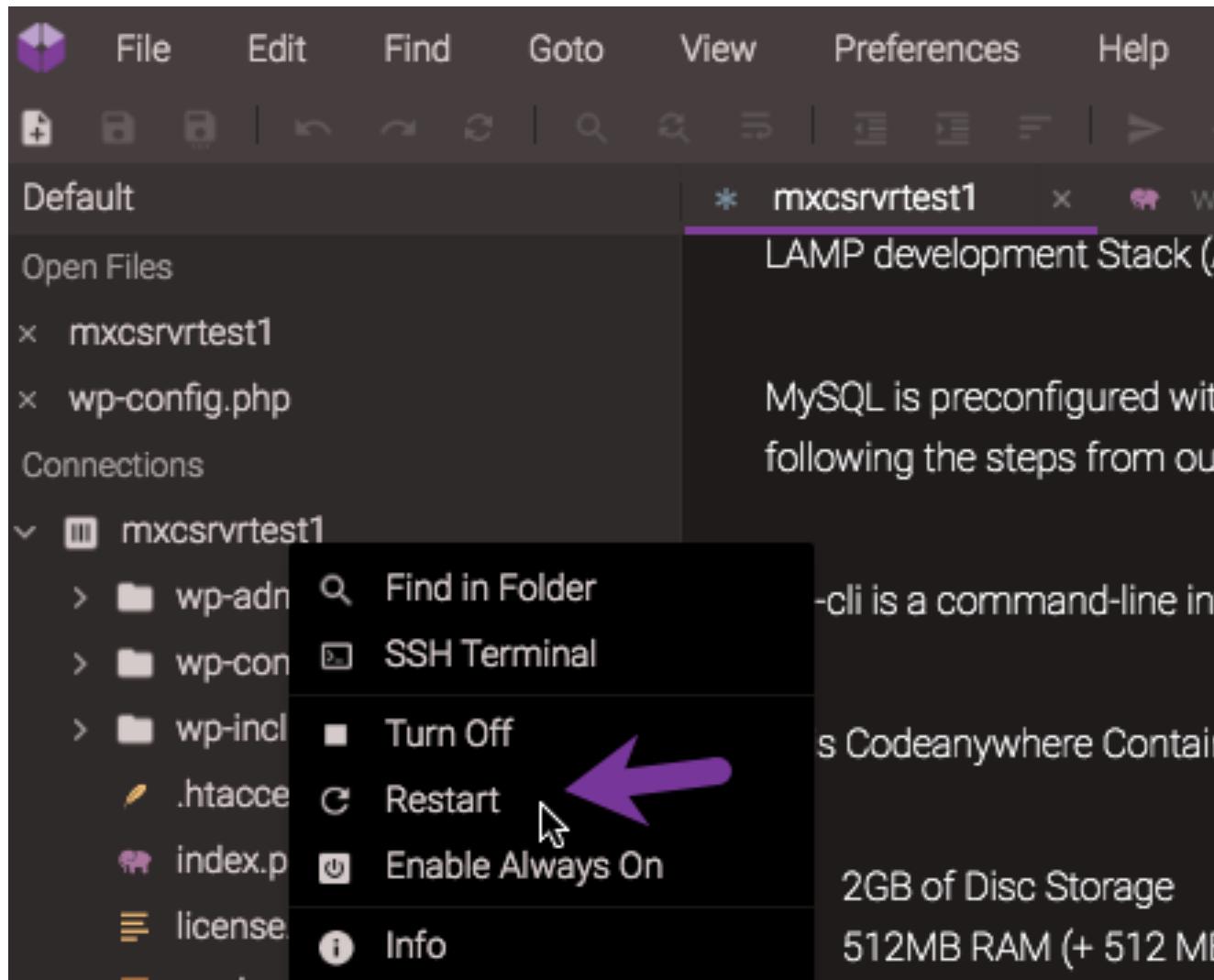


```
File Edit Find Goto View Preferences Help
Default mxcsrvrtest1 wp-config.php
Open Files
x mxcsrvrtest1
x wp-config.php
Connections
v mxcsrvrtest1
> wp-admin
> wp-content
> wp-includes
.htaccess
index.php
license.txt
readme.html
wp-activate.php
wp-blog-header.php
wp-comments-post.php
wp-config-sample.php
wp-config.php
wp-cron.php
wp-links-opml.php
wp-load.php
wp-login.php
wp-mail.php
wp-settings.php
wp-signup.php
wp-trackback.php
xmlrpc.php
3 * The base configuration for WordPress
4 *
5 * The wp-config.php creation script uses this file during the
6 * installation. You don't have to use the web site, you can
7 * copy this file to "wp-config.php" and fill in the values.
8 *
9 * This file contains the following configurations:
10 *
11 * * MySQL settings
12 * * Secret keys
13 * * Database table prefix
14 * * ABSPATH
15 *
16 * @link https://codex.wordpress.org/Editing_wp-config.php
17 *
18 * @package WordPress
19 */
20
21 // ** MySQL settings - You can get this info from your web host ** //
22 /** The name of the database for WordPress */
23 define('DB_NAME', 'wordpress');
24
25 /** MySQL database username */
26 define('DB_USER', 'root');
27
28 /** MySQL database password */
29 define('DB_PASSWORD', '');
30
31 /** MySQL hostname */
32 define('DB_HOST', 'localhost');
33
34 /** Database Charset to use in creating database tables. */
35 define('DB_CHARSET', 'utf8');
36
37 /** The Database Collate type. Don't change this if in doubt. */
```

Add the DB_PASSWORD and save

```
18 // ** MySQL settings - You can get this info from your web host ** //
19 */
20
21 // ** MySQL settings - You can get this info from your web host ** //
22 /** The name of the database for WordPress */
23 define('DB_NAME', 'wordpress');
24
25 /** MySQL database username */
26 define('DB_USER', 'root');
27
28 /** MySQL database password */
29 define('DB_PASSWORD', 'IT320@Winter2020!');
30
31 /** MySQL hostname */
32 define('DB_HOST', 'localhost');
33
34 /** Database Charset to use in creating database tables */
35
```

Restart the Server Connection then test that the WP site still works



WP site still works!

Screenshot of a web browser showing a WordPress site titled "MXC WP Test 1". The page displays a post about a squirrel, featuring a large image of a squirrel at a bird feeder.

The browser tabs show the following sites:

- cPanel X
- Default - Codeanywhere
- MXC WP Test 1 – Just a test
- mxcsvrtest1-mchase2
- wptest768/wp-config.php

The address bar shows the URL: mxcsvrtest1-mchase2247467.codeanyapp.com.

The page content includes:

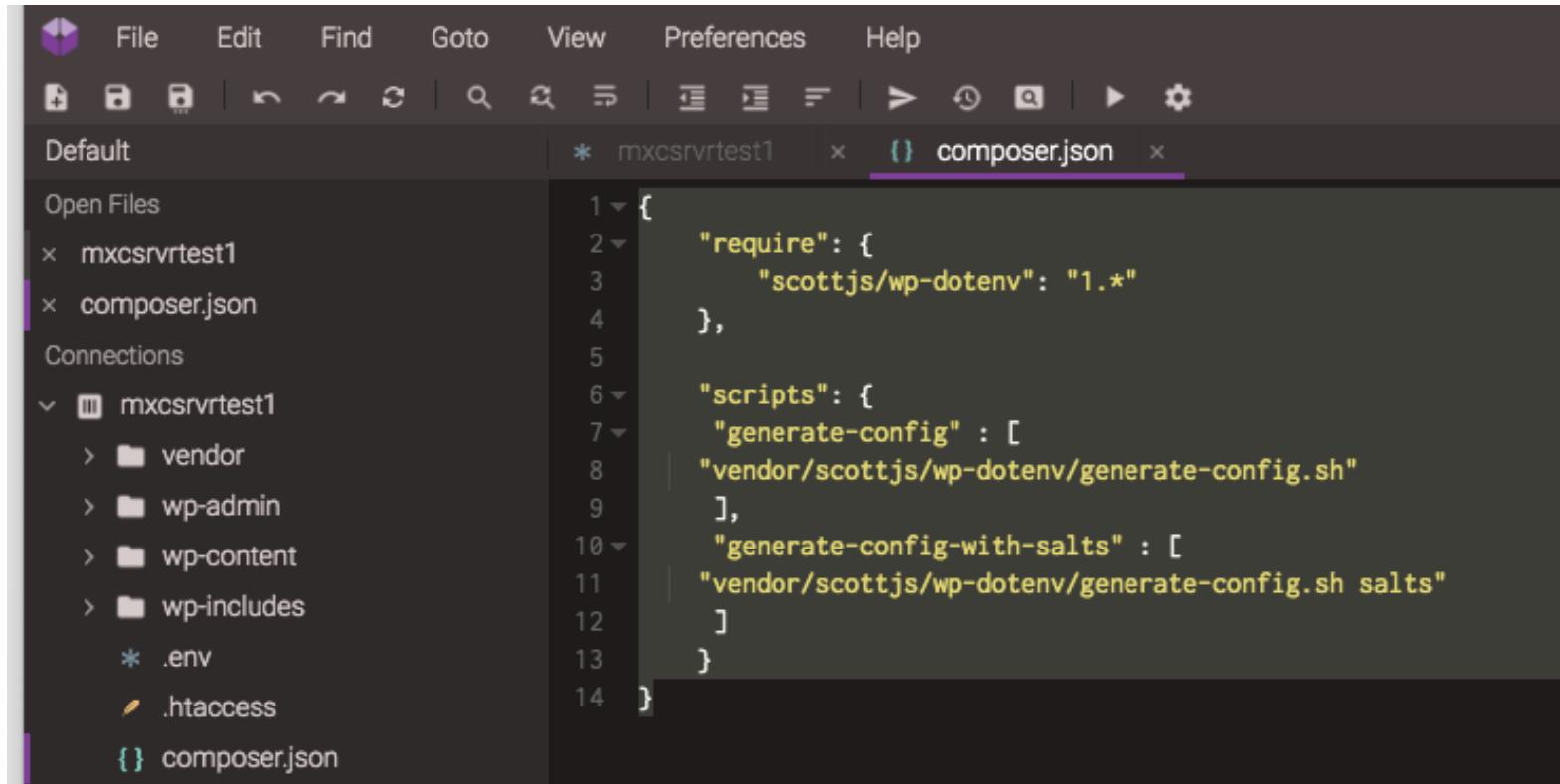
- Header: MXC WP Test 1 Just another WordPress site
- Header: Page Test 1 Page Test 2 Sample Page
- Section: SQUIRREL
- Title: Gray Squirrel
- Post details: By mxchase December 11, 2019 No Comments
- Image: A squirrel standing on a wooden board, looking at a hanging orange corn cob.

Background on DotEnv (.env) file

- The DotEnv is a file named `.env` that allows us to hide the MySQL database name and login credentials from the public
- We install a composer PHP script (`autoload.php`) that makes it possible for the WordPress `wp-config.php` to be able to read and import settings from the `.env` file
- We then create a `.env` file and configure it to contain the database environment and login credentials
- For all this to happen, we have to have `composer PHP package manager` installed on our web host and Codeanywhere connection serves do have composer installed.

(5) Create a composer.json file

- Create a composer.json file
- Edit the file to contain a generic json code block



The screenshot shows a code editor interface with a dark theme. The menu bar includes File, Edit, Find, Goto, View, Preferences, and Help. The toolbar contains various icons for file operations like Open, Save, and Find. The left sidebar displays the project structure under 'Default': 'mxcsrvrtest1' (closed), 'composer.json' (closed), 'Connections' (closed), 'mxcsrvrtest1' (open), containing 'vendor', 'wp-admin', 'wp-content', 'wp-includes', '.env', and '.htaccess'. The right pane shows the contents of the 'composer.json' file:

```
1 {  
2   "require": {  
3     "scottjs/wp-dotenv": "1.*"  
4   },  
5  
6   "scripts": {  
7     "generate-config" : [  
8       "vendor/scottjs/wp-dotenv/generate-config.sh"  
9     ],  
10    "generate-config-with-salts" : [  
11      "vendor/scottjs/wp-dotenv/generate-config.sh salts"  
12    ]  
13  }  
14 }
```

Codeanywhere potential composer issues

- Before you continue – please test your connection container composer
 - Open an SSH Terminal window
 - Execute a composer –V command

```
?> composer -V
```
 - Execute a composer self-update command

```
?> composer self-update
```
 - What we are looking for is whether your WordPress and Composer are installed and configured correctly

Why are we testing and what are we looking for?

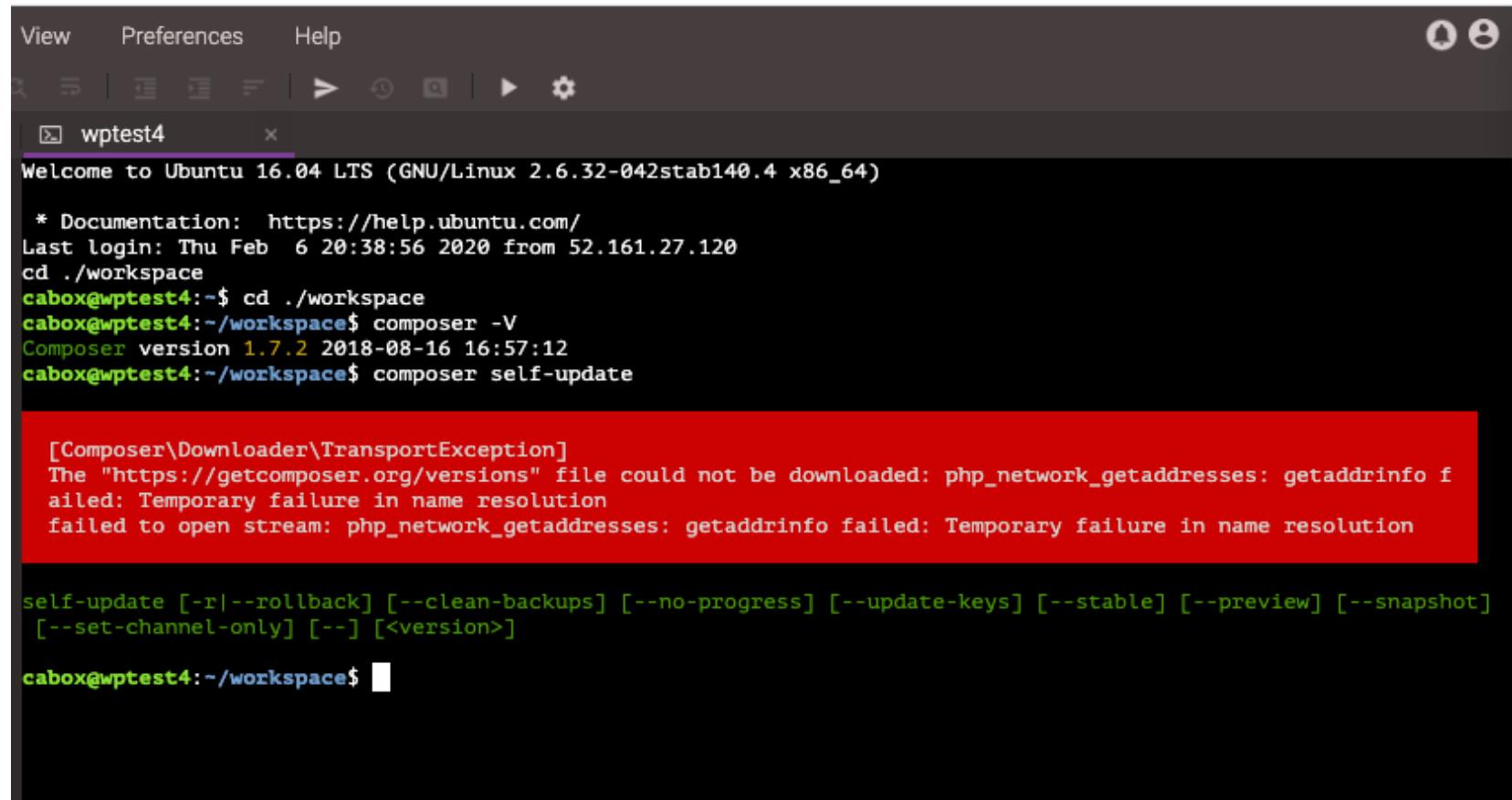
We are using a basic composer command to test that WordPress and Composer are installed and configured correctly.

If they are, the composer self-update command will work successfully and update composer for 1.7.2 to 1.9.3

If they are not then you may see an error.

Based on the type of error, this presentation outlines the steps to take to try and resolve them

Transport Exception error – on executing composer self-update or composer install



The screenshot shows a terminal window titled "wptest4" running on an Ubuntu 16.04 LTS system. The user has navigated to their workspace directory and run the command "composer self-update". The output shows the command being run, followed by an error message in a red box indicating a "TransportException" due to a failed network connection. The error message states that the "https://getcomposer.org/versions" file could not be downloaded because "php_network_getaddresses: getaddrinfo failed: Temporary failure in name resolution". Below the error message, the help text for the "self-update" command is displayed, and the command is completed with a status indicator.

```
View Preferences Help
File Edit View Insert Cell Terminal Help
wptest4
Welcome to Ubuntu 16.04 LTS (GNU/Linux 2.6.32-042stab140.4 x86_64)

 * Documentation: https://help.ubuntu.com/
Last login: Thu Feb  6 20:38:56 2020 from 52.161.27.120
cd ./workspace
cabox@wptest4:~$ cd ./workspace
cabox@wptest4:~/workspace$ composer -V
Composer version 1.7.2 2018-08-16 16:57:12
cabox@wptest4:~/workspace$ composer self-update

[Composer\Downloader\TransportException]
The "https://getcomposer.org/versions" file could not be downloaded: php_network_getaddresses: getaddrinfo f
ailed: Temporary failure in name resolution
failed to open stream: php_network_getaddresses: getaddrinfo failed: Temporary failure in name resolution

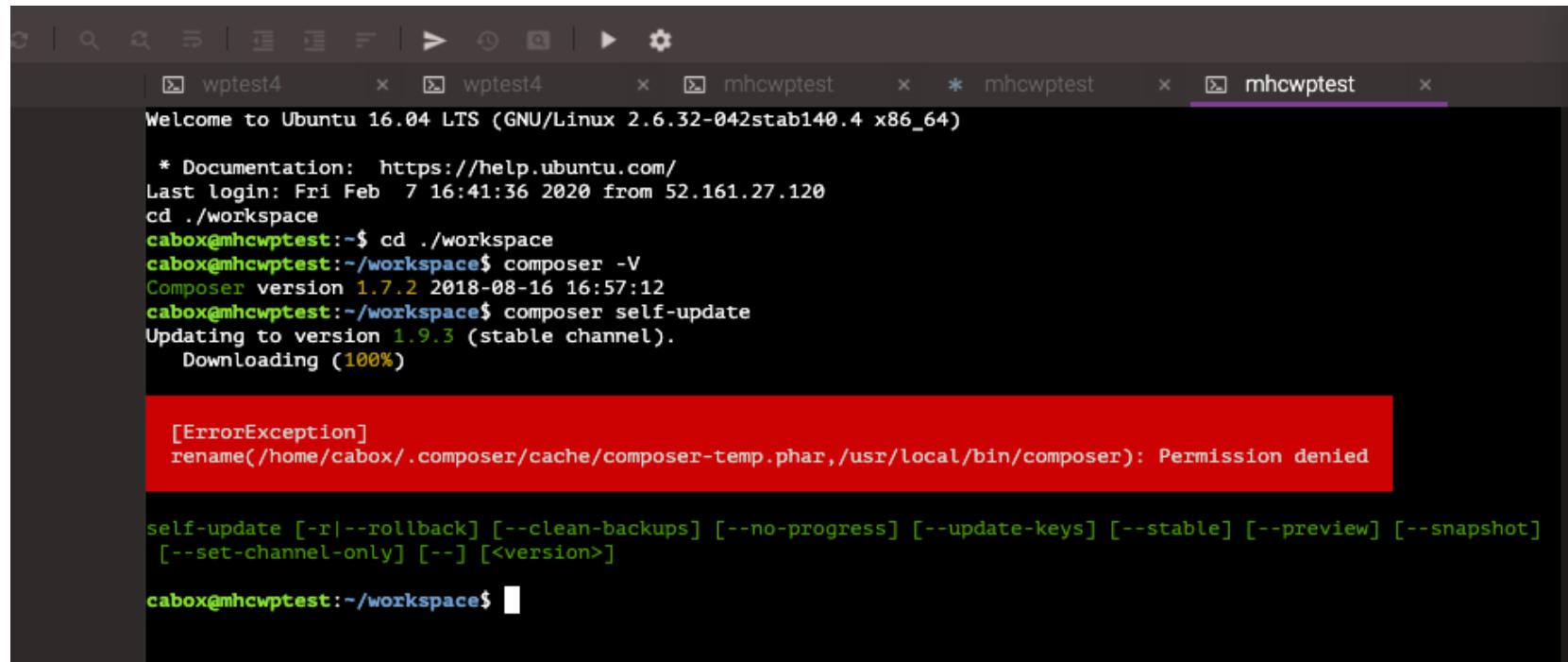
self-update [-r|--rollback] [--clean-backups] [--no-progress] [--update-keys] [--stable] [--preview] [--snapshot]
[--set-channel-only] [--] [<version>]

cabox@wptest4:~/workspace$ █
```

The **Transport Exception** Error Means the WordPress site or the underlying LAMP or Composer is not configured correctly

- There is no fix for this – sometimes the Codeanywhere or the server spin up fails.
- Destroy the connection-container, wait 5 mins and then create a new Codeanywhere Connection – Container for WordPress and build and configure a new WordPress site

After you remake the container and make a new WP site you get a **Permission denied** error while executing composer self-update command



The screenshot shows a terminal window with several tabs open, but the current tab, "mhcwptest", is active. The terminal displays a standard Ubuntu 16.04 LTS welcome message. The user runs "cd ./workspace" and then "composer -V" which shows Composer version 1.7.2. The user then runs "composer self-update". The output shows the update process starting at 1.9.3 and reaching 100% download. A red box highlights the error message: "[ErrorException] rename(/home/cabox/.composer/cache/composer-temp.phar,/usr/local/bin/composer): Permission denied". Below the error, the help text for the "self-update" command is shown, and the command "cabox@mhcwptest:~/workspace\$" is at the bottom.

```
Welcome to Ubuntu 16.04 LTS (GNU/Linux 2.6.32-042stab140.4 x86_64)

 * Documentation: https://help.ubuntu.com/
Last login: Fri Feb  7 16:41:36 2020 from 52.161.27.120
cd ./workspace
cabox@mhcwptest:~$ cd ./workspace
cabox@mhcwptest:~/workspace$ composer -V
Composer version 1.7.2 2018-08-16 16:57:12
cabox@mhcwptest:~/workspace$ composer self-update
Updating to version 1.9.3 (stable channel).
  Downloading (100%)

[ErrorException]
rename(/home/cabox/.composer/cache/composer-temp.phar,/usr/local/bin/composer): Permission denied

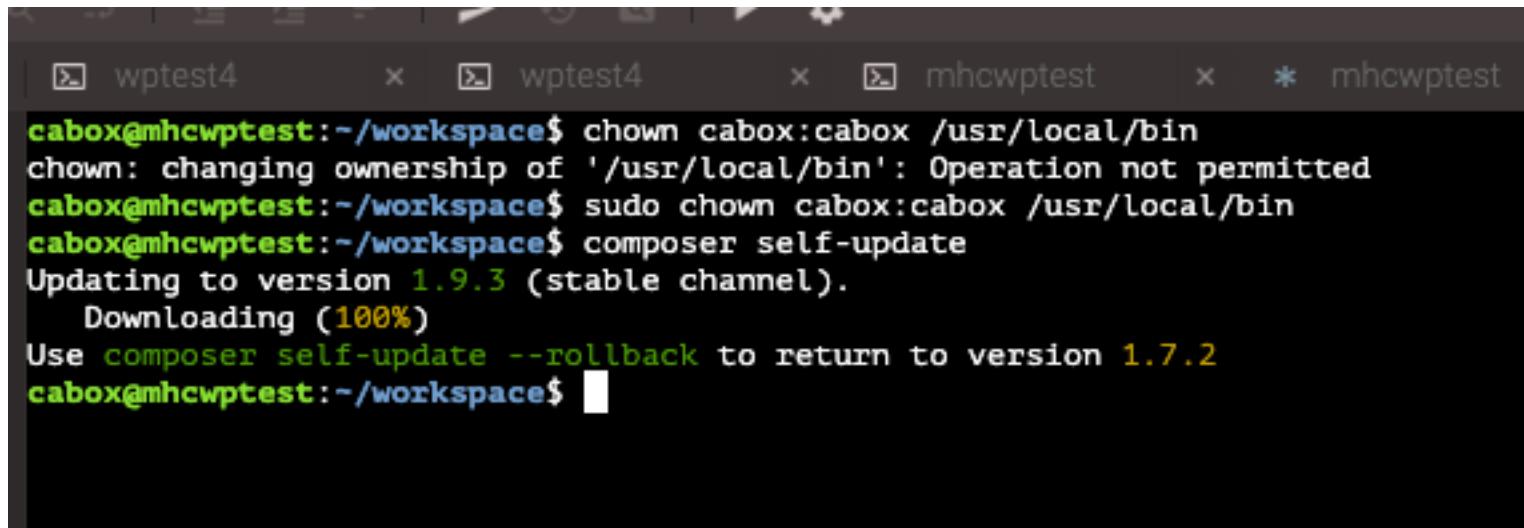
self-update [-r|--rollback] [--clean-backups] [--no-progress] [--update-keys] [--stable] [--preview] [--snapshot]
[--set-channel-only] [--] [<version>]

cabox@mhcwptest:~/workspace$
```

Permission denied errors can be fixed with one simple (to me) command

- To fix, we must adjust the folder ownership of the `/usr/local/bin` folder (its the folder mentioned in the error message)
- To change the folder ownership, we execute a `chown` command as follows:

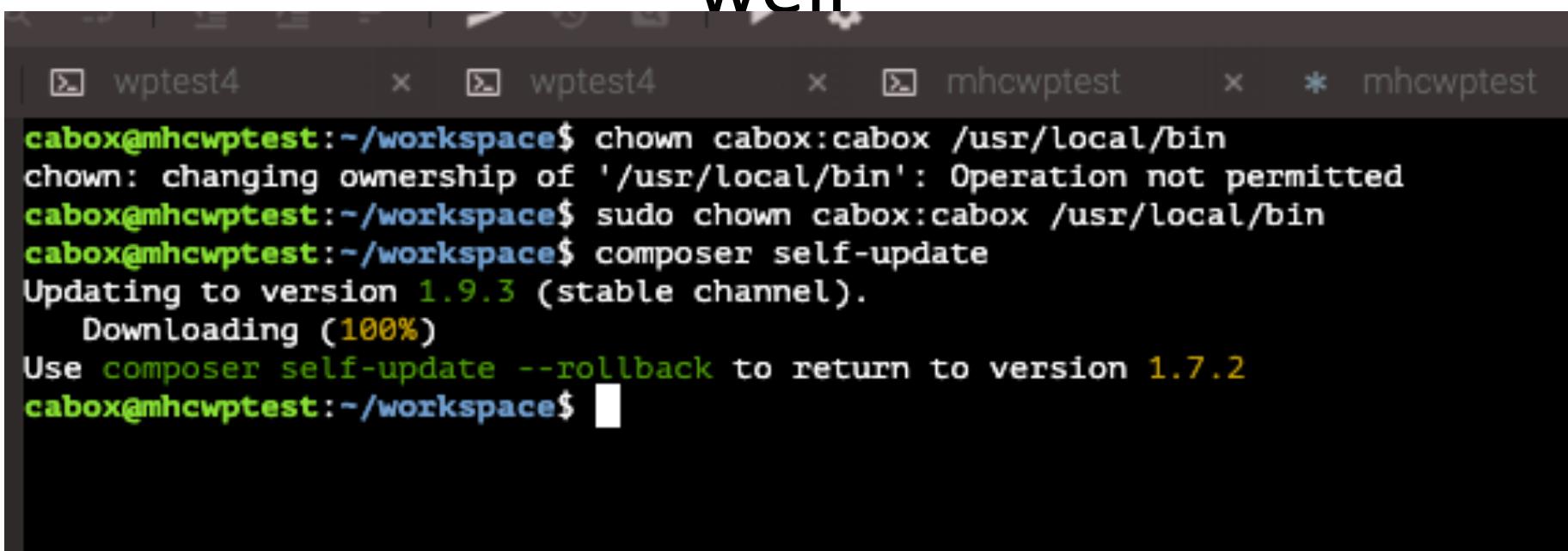
```
cabox@mhcwpptest ~workspace$ sudo chown cabox:cabox /usr/local/bin
```



The screenshot shows a terminal window with four tabs at the top: "wptest4", "wptest4", "mhcwpptest", and "mhcwpptest". The active tab is "mhcwpptest". The terminal output is as follows:

```
cabox@mhcwpptest:~/workspace$ chown cabox:cabox /usr/local/bin
chown: changing ownership of '/usr/local/bin': Operation not permitted
cabox@mhcwpptest:~/workspace$ sudo chown cabox:cabox /usr/local/bin
cabox@mhcwpptest:~/workspace$ composer self-update
Updating to version 1.9.3 (stable channel).
    Downloading (100%)
Use composer self-update --rollback to return to version 1.7.2
cabox@mhcwpptest:~/workspace$ █
```

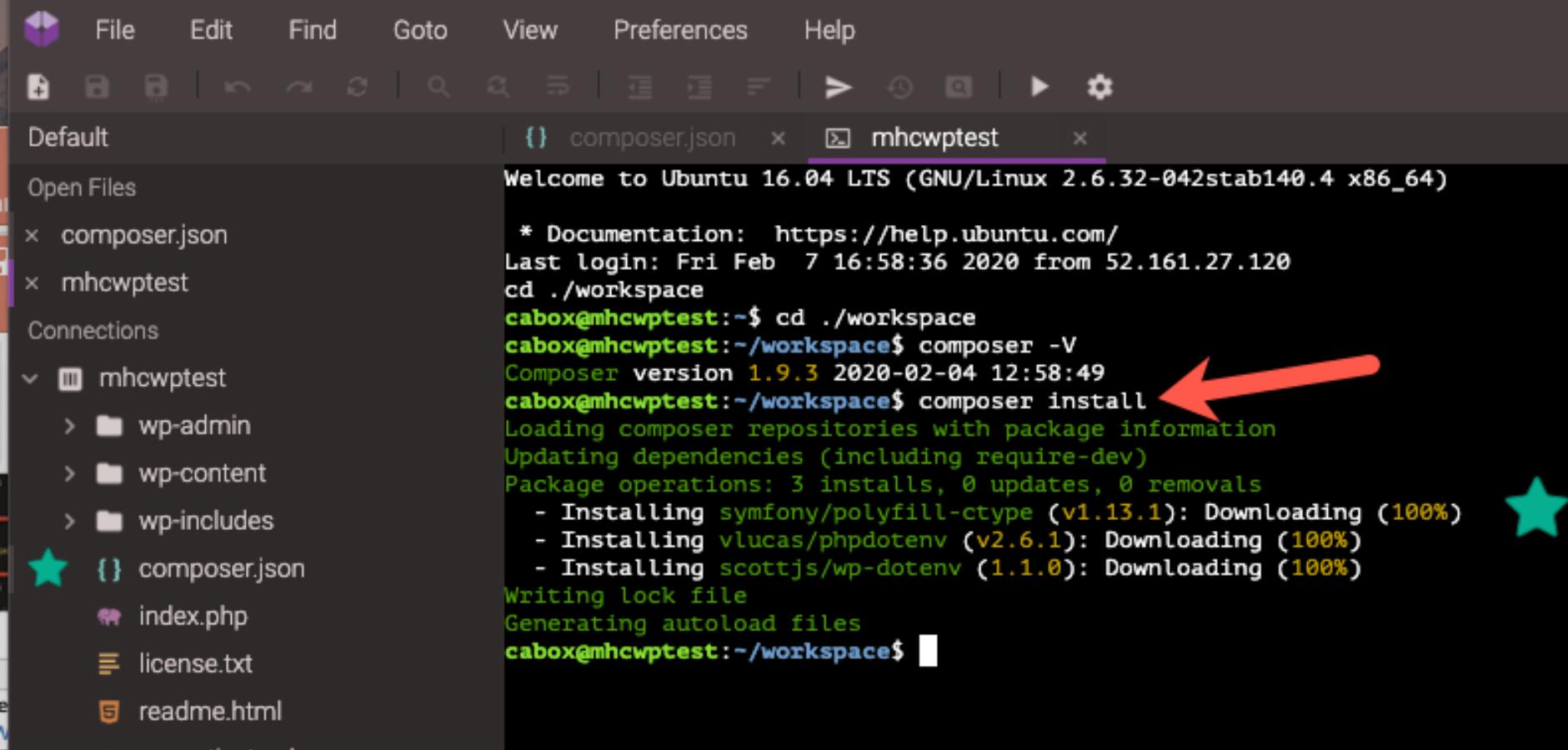
Rerun the composer self-update to test the fix. Composer commands should work – the composer install should work now as well



The screenshot shows a terminal window with four tabs open: 'wptest4' (x), 'wptest4' (x), 'mhcwptest' (x), and 'mhcwptest' (*). The active tab is 'mhcwptest'. The terminal output is as follows:

```
cabox@mhcwptest:~/workspace$ chown cabox:cabox /usr/local/bin
chown: changing ownership of '/usr/local/bin': Operation not permitted
cabox@mhcwptest:~/workspace$ sudo chown cabox:cabox /usr/local/bin
cabox@mhcwptest:~/workspace$ composer self-update
Updating to version 1.9.3 (stable channel).
    Downloading (100%)
Use composer self-update --rollback to return to version 1.7.2
cabox@mhcwptest:~/workspace$ █
```

After executing the chown command on /usr/local/bin – run the composer install

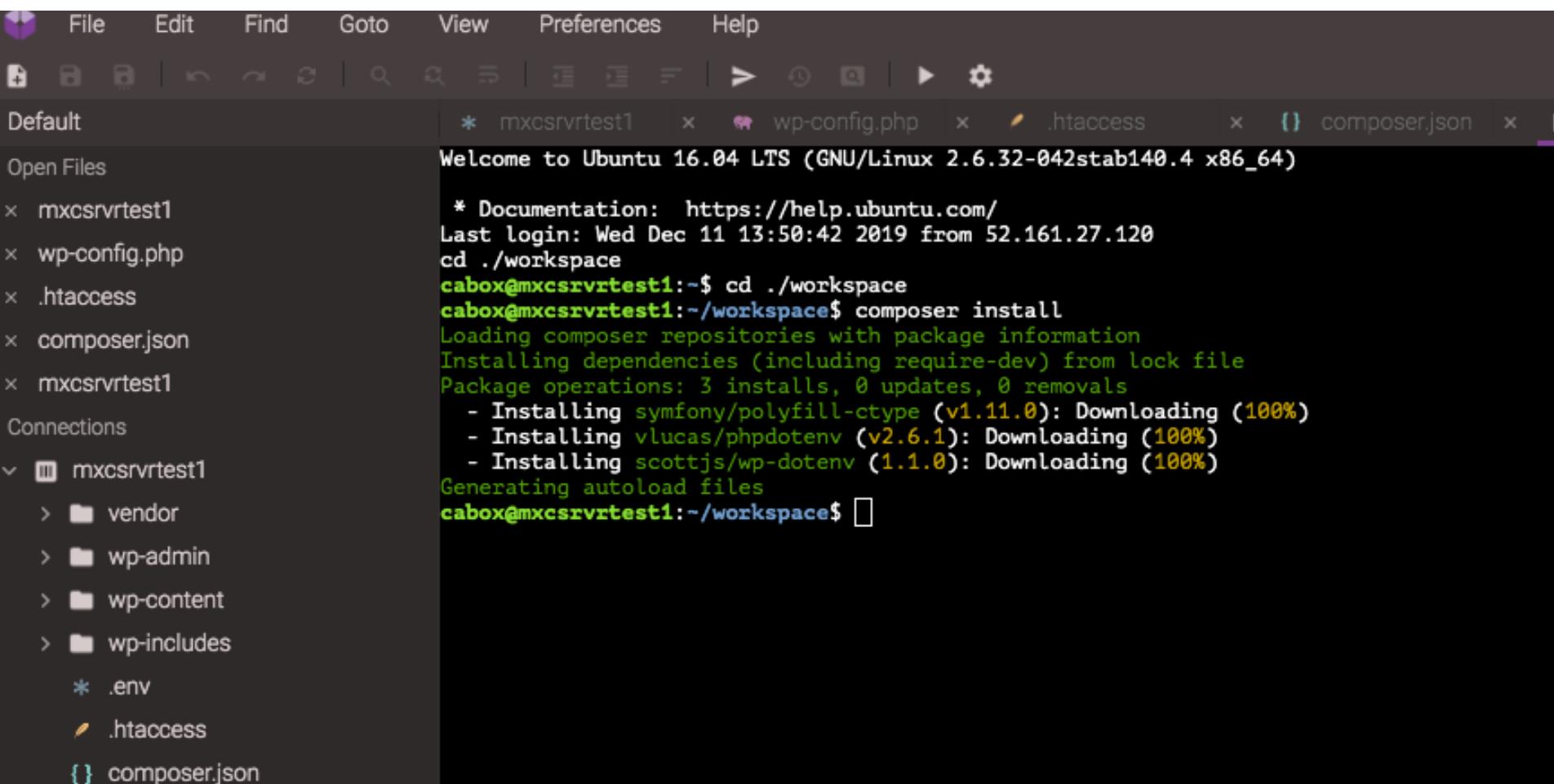


```
Welcome to Ubuntu 16.04 LTS (GNU/Linux 2.6.32-042stab140.4 x86_64)

 * Documentation: https://help.ubuntu.com/
Last login: Fri Feb  7 16:58:36 2020 from 52.161.27.120
cd ./workspace
cabox@mhcwpptest:~/workspace$ cd ./workspace
cabox@mhcwpptest:~/workspace$ composer -V
Composer version 1.9.3 2020-02-04 12:58:49
cabox@mhcwpptest:~/workspace$ composer install
Loading composer repositories with package information
Updating dependencies (including require-dev)
Package operations: 3 installs, 0 updates, 0 removals
- Installing symfony/polyfill-ctype (v1.13.1): Downloading (100%)
- Installing vlucas/phpdotenv (v2.6.1): Downloading (100%)
- Installing scottjs/wp-dotenv (1.1.0): Downloading (100%)
Writing lock file
Generating autoload files
cabox@mhcwpptest:~/workspace$
```

(6) After creating the composer.json file, execute (run) the composer install command

Execute the “composer install” command to install the composer phpdotenv and wp-dotenv files. These files are essential for us being able to use the .env in our process



The screenshot shows a terminal window with a dark theme. The title bar includes icons for file operations and a menu bar with File, Edit, Find, Goto, View, Preferences, and Help. The main area displays a terminal session on an Ubuntu 16.04 LTS system. The session starts with a welcome message and then shows the user navigating to their workspace and running the 'composer install' command. The output of the command shows the installation of three packages: symfony/polyfill-ctype, vlucas/phpdotenv, and scottjs/wp-dotenv. The terminal ends with a prompt and a square icon.

```
Welcome to Ubuntu 16.04 LTS (GNU/Linux 2.6.32-042stab140.4 x86_64)

 * Documentation: https://help.ubuntu.com/
Last login: Wed Dec 11 13:50:42 2019 from 52.161.27.120
cd ./workspace
cabox@mxcsvrtest1:~/workspace$ composer install
Loading composer repositories with package information
Installing dependencies (including require-dev) from lock file
Package operations: 3 installs, 0 updates, 0 removals
  - Installing symfony/polyfill-ctype (v1.11.0): Downloading (100%)
  - Installing vlucas/phpdotenv (v2.6.1): Downloading (100%)
  - Installing scottjs/wp-dotenv (1.1.0): Downloading (100%)
Generating autoload files
cabox@mxcsvrtest1:~/workspace$ 
```

(7) Overriding the DB options table `siteurl` and `home` `.env` file + mods to the `wp-config.php` file

- To override the database options table `siteurl` and `home` settings, we will employ a `.env` file to contain WordPress database environmental settings and database login credentials

(7) WordPress Database options table

WordPress database – options table, **siteurl** and **home** values are the URL paths for the original WordPress site. These values prevent our WordPress site from being installed and run on different platforms or environments

The screenshot shows the phpMyAdmin interface for the 'wordpress' database. The left sidebar shows the database structure with 'wp_options' highlighted and selected, indicated by a red box. The main area displays the 'wp_options' table with the following data:

option_id	option_name	option_value	autoload
1	siteurl	https://mxcsvrtest1-mchase2247467.codeanyapp.com	yes
2	home	https://mxcsvrtest1-mchase2247467.codeanyapp.com	yes
3	blogname	MXC WP Test 1	yes
4	blogdescription	Just another WordPress site	yes

(7) BACKGROUND on Configuring WP for Public GitHub Storage

We must re-configure our WordPress site configuration files to ignore the database options table site environment to allow our GitHub Repo versioned site to be downloaded from GitHub and run on different platforms and environments.

Download a copy of the [wp-config.php](#) as a backup

Add the following code after the initial comment block in the [wp-config.php](#)

```
define('WP_HOME', 'http://'. $_SERVER['HTTP_HOST']);  
define('WP_SITEURL', 'http://'. $_SERVER['HTTP_HOST']);
```

(8) Address WordPress Configuration File Hard Coded Settings

The `wp-config.php` file has a block of code where the database name, username, and password as well as the database table prefix are hard coded values

1. We can not put this information on a public repository
2. We will contain our configuration settings in a `.env` file that we will not publish (push) to the GitHub repository.
3. We will publish an example `.env` file, and provide our trusted developers with the correct settings values

Download a copy of the wp-config.php as a backup

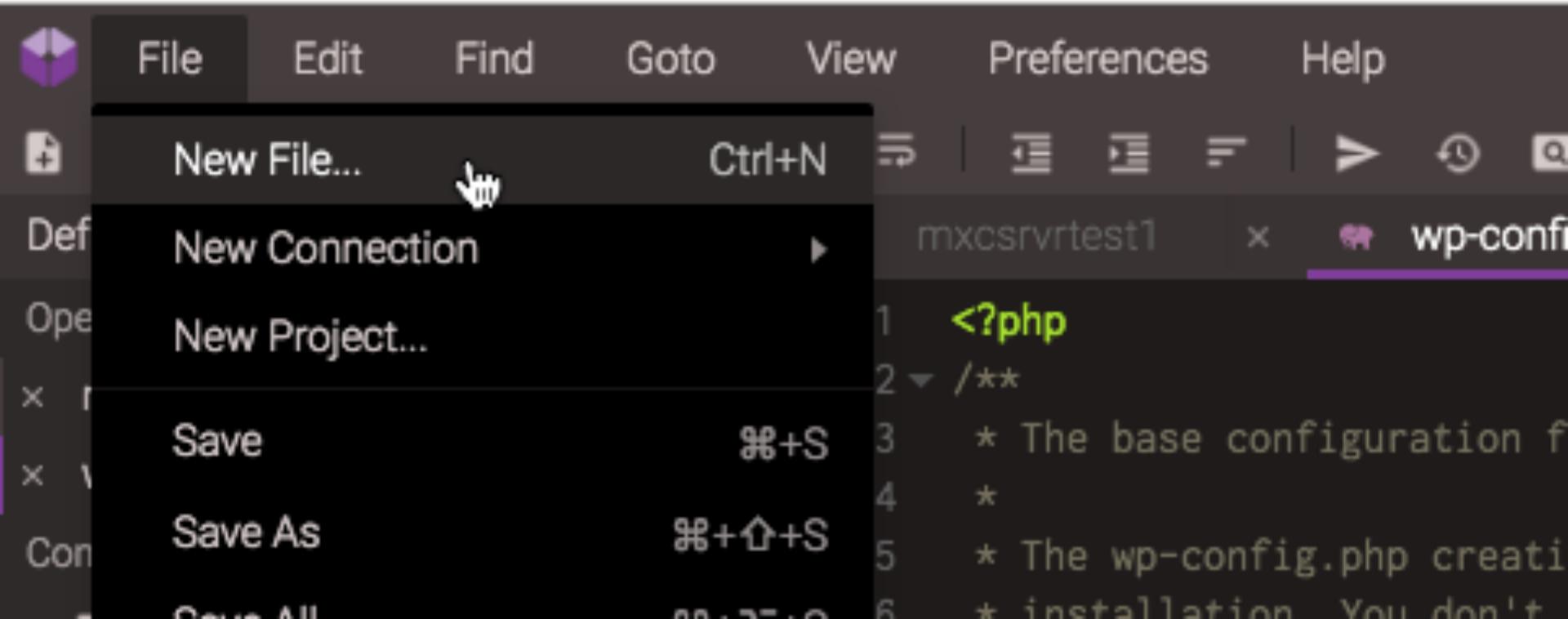
Add this code to the wp-config.php

Informs the WP site to use the [.env](#) file settings

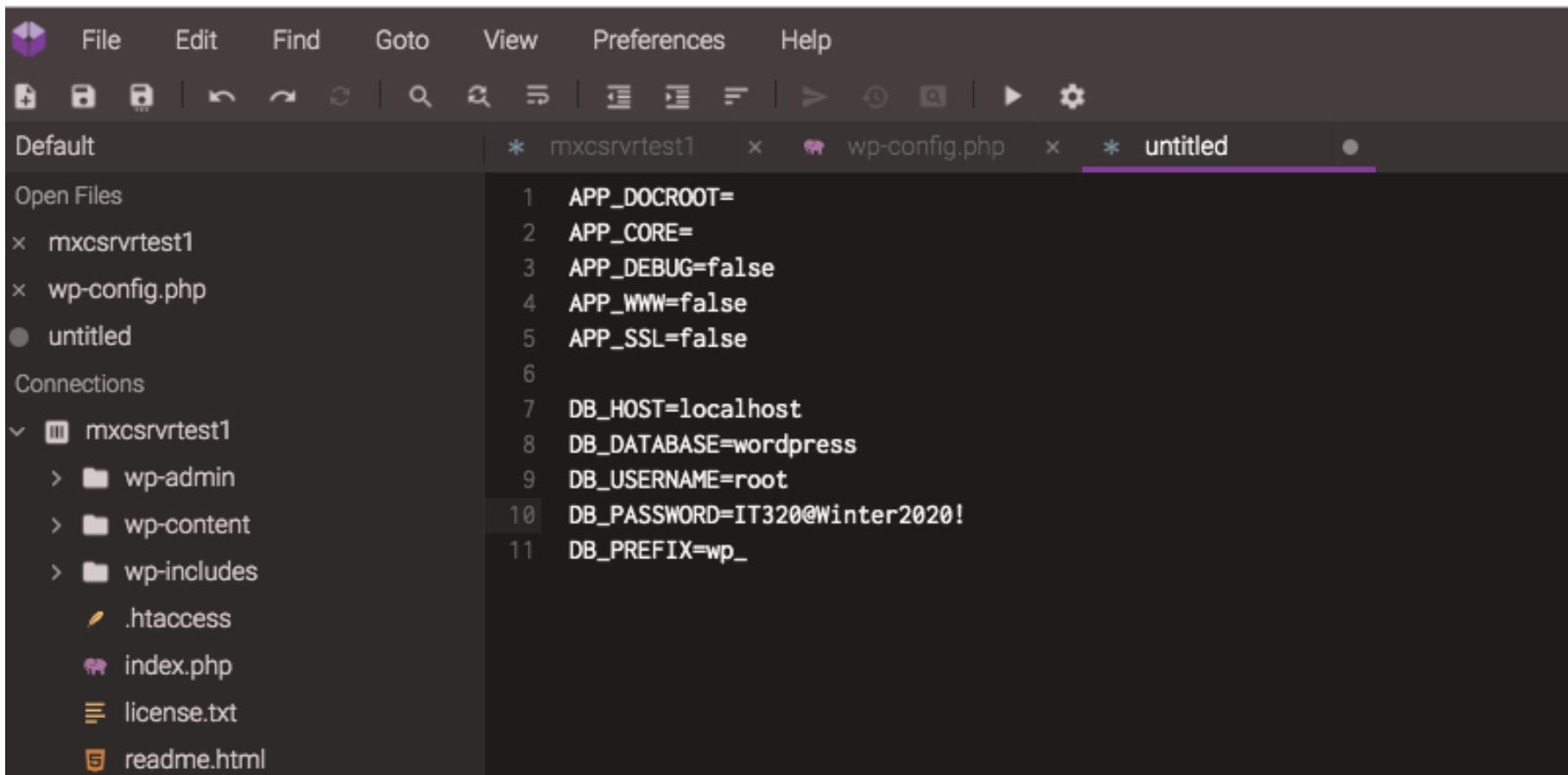
```
/**  
 * Include Dotenv library to pull config options from .env file.  
 */  
  
if(file_exists(__DIR__ . '/vendor/autoload.php')) {  
    require_once __DIR__ . '/vendor/autoload.php';  
    $dotenv = new Dotenv\Dotenv(__DIR__);  
    $dotenv->load();  
}  
if(file_exists(dirname(__DIR__) . '/vendor/autoload.php')) {  
    require_once dirname(__DIR__) . '/vendor/autoload.php';  
    $dotenv = new Dotenv\Dotenv(dirname(__DIR__));  
    $dotenv->load();  
}
```

Create the new .env file

File > New File...



Create a new .env file with correct Database and root login credentials + valid database table prefix



The screenshot shows a code editor interface with a dark theme. The top menu bar includes File, Edit, Find, Goto, View, Preferences, and Help. Below the menu is a toolbar with various icons. The left sidebar displays the project structure under 'Default':

- Open Files:
 - mxcsrvrtest1
 - wp-config.php
 - untitled
- Connections:
 - mxcsrvrtest1
 - wp-admin
 - wp-content
 - wp-includes
 - .htaccess
 - index.php
 - license.txt
 - readme.html

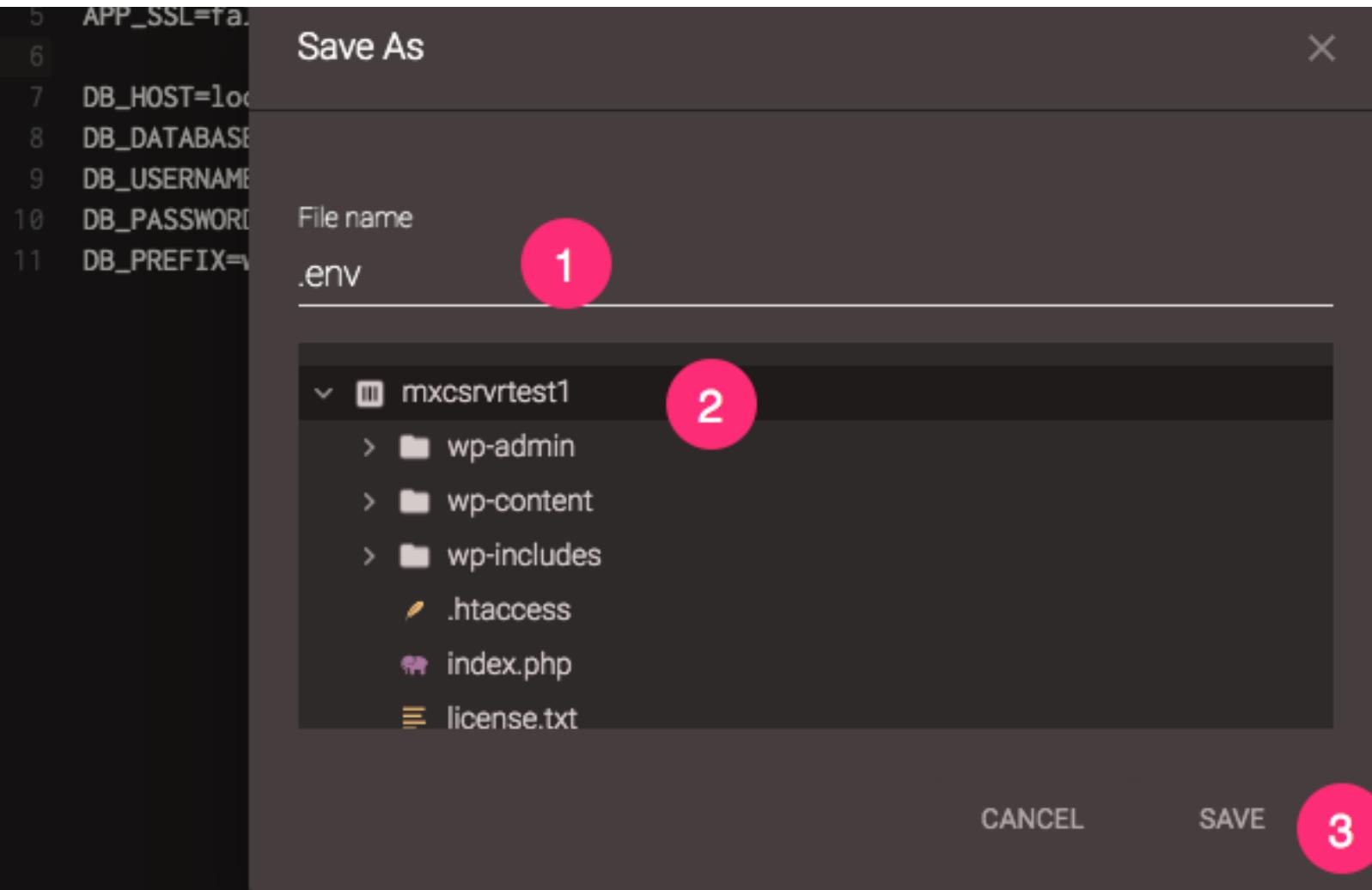
The 'untitled' tab is currently active, showing the following .env file content:

```
APP_DOCROOT=
APP_CORE=
APP_DEBUG=false
APP_WWW=false
APP_SSL=false
DB_HOST=localhost
DB_DATABASE=wordpress
DB_USERNAME=root
DB_PASSWORD=IT320@Winter2020!
DB_PREFIX=wp_
```

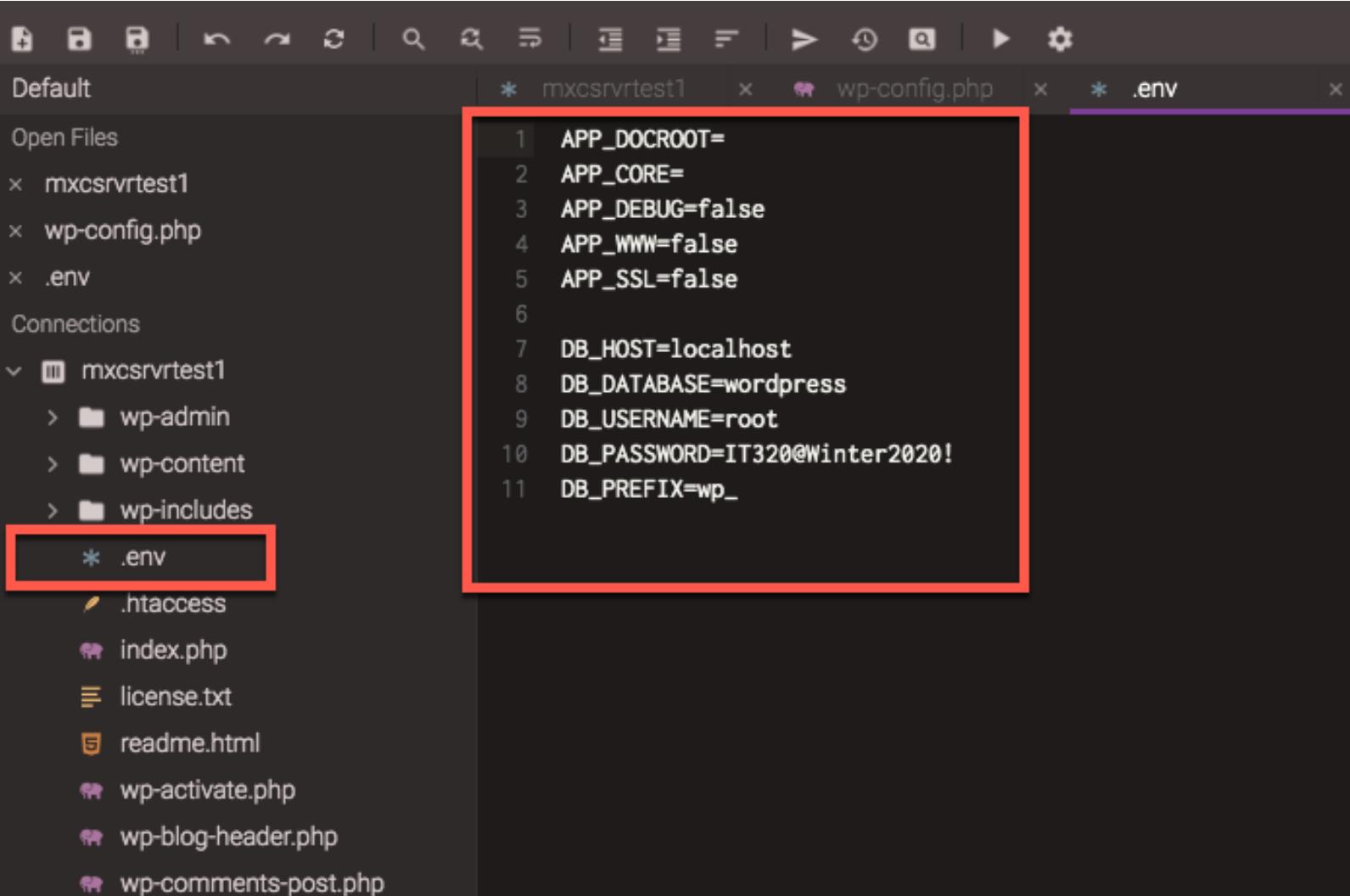
Note about copying .env files

- On Mac(s) you can not save a .env file, as files that start with a period are reserved for system files.
- To get around this, we usually name these files as dot-env and then we can copy them to Linux platforms and rename them to .env
- Make sure that if you copied the .env file from another platform, you check that its file name is .env.
- And if its not, rename it on Codeanywhere

- (1) Enter File name of .env
- (2) Select the connection folder
- (3) SAVE



.env file has been created



The screenshot shows a code editor interface with a dark theme. On the left, the sidebar lists several files: 'Default', 'Open Files' (including 'mxcsrvrtest1', 'wp-config.php', and '.env'), 'Connections' (with 'mxcsrvrtest1' expanded to show 'wp-admin', 'wp-content', and 'wp-includes'), and a list of PHP files ('.htaccess', 'index.php', 'license.txt', 'readme.html', 'wp-activate.php', 'wp-blog-header.php', and 'wp-comments-post.php'). The '.env' file is selected and highlighted with a red border. The main workspace shows the contents of the '.env' file:

```
1 APP_DOCROOT=
2 APP_CORE=
3 APP_DEBUG=false
4 APP_WWW=false
5 APP_SSL=false
6
7 DB_HOST=localhost
8 DB_DATABASE=wordpress
9 DB_USERNAME=root
10 DB_PASSWORD=IT320@Winter2020!
11 DB_PREFIX=wp_
```

(1) wp-config.php modifications

Original wp-config.php

```
mxcsrvrtest1  x  wp-config.php  x

<?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://codex.wordpress.org/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');
```

Modified to use .env file

```
1  <?php
2  /**
3   * The base configuration for WordPress
4   *
5   * The wp-config.php creation script uses this file during the
6   * installation. You don't have to use the web site, you can
7   * copy this file to "wp-config.php" and fill in the values.
8   *
9   * This file contains the following configurations:
10  *
11  * ** MySQL settings
12  * ** Secret keys
13  * ** Database table prefix
14  * ** ABSPATH
15  *
16  * @link https://codex.wordpress.org/Editing_wp-config.php
17  *
18  * @package WordPress
19  */
20
21  define('WP_HOME', 'http://' . $_SERVER['HTTP_HOST']);
22  define('WP_SITEURL', 'http://' . $_SERVER['HTTP_HOST']);
23  /**
24   * Include Dotenv library to pull config options from .env file.
25  */
26
27  if(file_exists(__DIR__ . '/vendor/autoload.php')) {
28      require_once __DIR__ . '/vendor/autoload.php';
29      $dotenv = new Dotenv\Dotenv(__DIR__);
30      $dotenv->load();
31  }
32  if(file_exists(dirname(__DIR__) . '/vendor/autoload.php')) {
33      require_once dirname(__DIR__) . '/vendor/autoload.php';
34      $dotenv = new Dotenv\Dotenv(dirname(__DIR__));
35      $dotenv->load();
36  }
37
```

(2) wp-config.php modifications

Original wp-config.php

```
38 // ** MySQL settings - You can get this info from your web host **
39 /** The name of the database for WordPress */
40 define('DB_NAME', 'wordpress');
41
42 /** MySQL database username */
43 define('DB_USER', 'root');
44
45 /** MySQL database password */
46 define('DB_PASSWORD', 'IT320@Winter2020!');
47
48 /** MySQL hostname */
49 define('DB_HOST', 'localhost');
50
51 /** Database Charset to use in creating database tables. */
52 define('DB_CHARSET', 'utf8');
53
54 /** The Database Collate type. Don't change this if in doubt. */
55 define('DB_COLLATE', '');
56
```

Add .env getenv commands for DATABASE and Host

```
37
38 // ** MySQL settings - You can get this info from your web host **
39 /** The name of the database for WordPress */
40 # define('DB_NAME', 'wordpress');
41 define('DB_NAME', getenv('DB_DATABASE'));
42 /** MySQL database username */
43 # define('DB_USER', 'root');
44 define('DB_USER', getenv('DB_USERNAME'));
45 /** MySQL database password */
46 #define('DB_PASSWORD', 'IT320@Winter2020!');
47 define('DB_PASSWORD', getenv('DB_PASSWORD'));
48 /** MySQL hostname */
49 #define('DB_HOST', 'localhost');
50 define('DB_HOST', getenv('DB_HOST'));
51 /** Database Charset to use in creating database tables. */
52 define('DB_CHARSET', 'utf8');
53
54 /** The Database Collate type. Don't change this if in doubt. */
55 define('DB_COLLATE', '');
56
```

(3) wp-config.php modifications

Original wp-config.php

```
38 // ** MySQL settings - You can get this info from your web host **
39 define('DB_NAME', 'wordpress');
40
41
42 define('DB_USER', 'root');
43
44
45 define('DB_PASSWORD', 'IT320@Winter2020!');
46
47
48 define('DB_HOST', 'localhost');
49
50
51 define('DB_CHARSET', 'utf8');
52
53
54 define('DB_COLLATE', ''');
55
56
```

Modified to use .env file

Remove original hard coded values

```
37
38 // ** MySQL settings - You can get this info from your web host **
39 define('DB_NAME', getenv('DB_DATABASE'));
40
41
42 define('DB_USER', getenv('DB_USERNAME'));
43
44
45 define('DB_PASSWORD', getenv('DB_PASSWORD'));
46
47
48 define('DB_HOST', getenv('DB_HOST'));
49
50
51 define('DB_CHARSET', 'utf8');
52
53
54 define('DB_COLLATE', ''');
55
56
```

Config ENV Settings – Before I removed the old hard coded settings values

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

/** MySQL database username */
#define('DB_USER', 'root');
define('DB_USER', getenv('DB_USERNAME'));

/** MySQL database password */
#define('DB_PASSWORD', 'Winter2020');
define('DB_PASSWORD', getenv('DB_PASSWORD'));

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

Config ENV Settings – only DotEnv Values

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

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

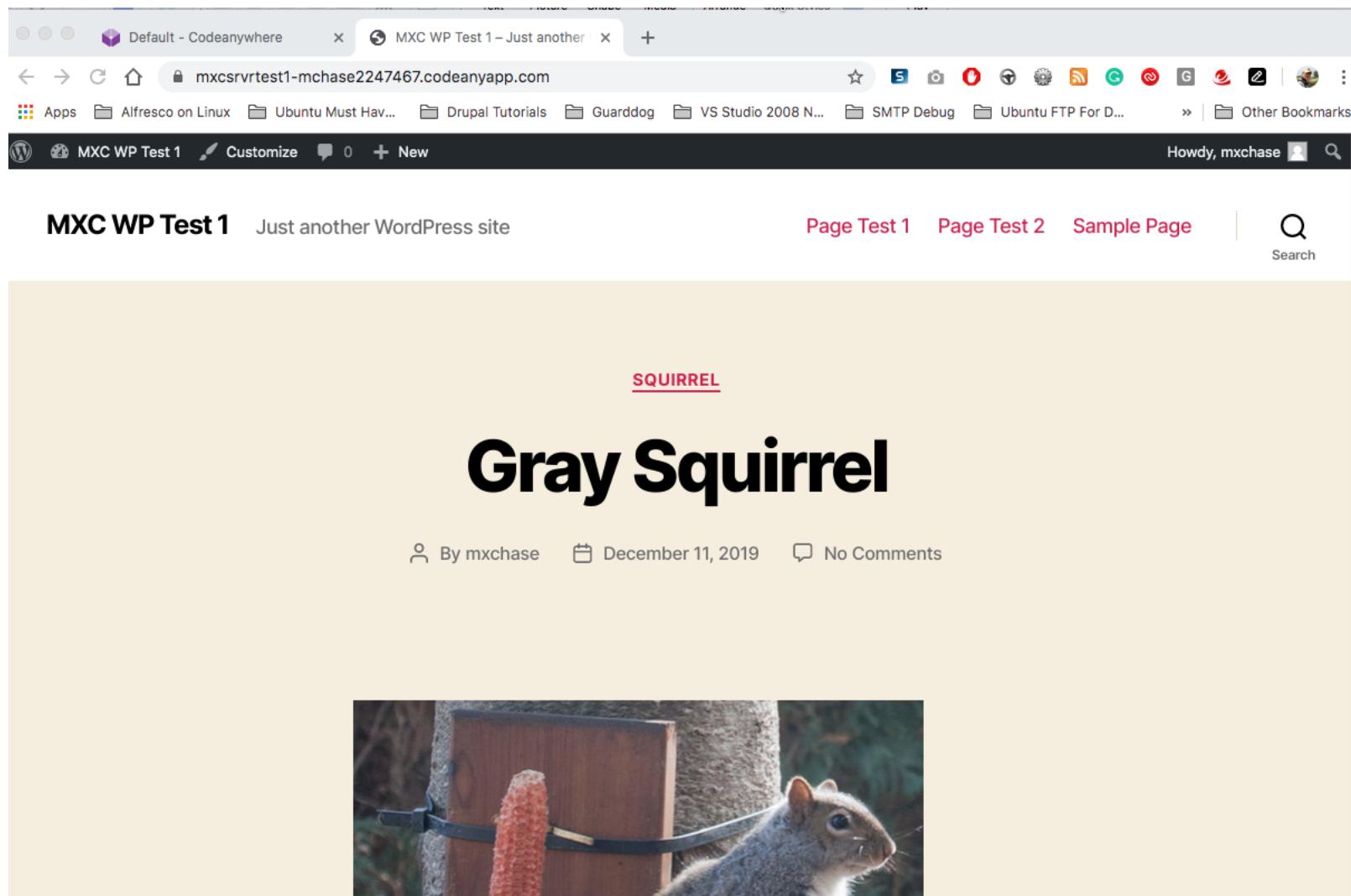
/** MySQL database password */
define('DB_PASSWORD', getenv('DB_PASSWORD'));

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

Save wp-config.php file changes, retest WP site

- Save `wp-config.php` file changes
- restart the Codeanywhere connection
- Test that the WordPress site still works from the info page link

WordPress site works, using the .env settings



A screenshot of a web browser window displaying a WordPress test site. The browser's address bar shows the URL `mxcsvrtest1-mchase2247467.codeanyapp.com`. The page title is "MXC WP Test 1 – Just another WordPress site". The main content features a large heading "Gray Squirrel" with the word "SQUIRREL" in red, underlined text above it. Below the heading is a small bio and a photo of a squirrel. The top navigation bar includes links for "Page Test 1", "Page Test 2", and "Sample Page". The bottom of the page shows the author information "By mxchase" and the date "December 11, 2019".

Default - Codeanywhere

MXC WP Test 1 – Just another WordPress site

SQUIRREL

Gray Squirrel

By mxchase December 11, 2019 No Comments



(8) Create an encrypted version of the MySQL DB backup file

- We have to encrypt (**harden**) our MySQL Database for storage on a PUBLIC GitHub repository.
- We can not prevent anyone from downloading a copy of our repository, but we can prevent unauthorized access to the WordPress content by encrypting the database backup and storing the encrypted version and not the database backup file on the GitHub repository
- We will then provide the encryption key to trusted developers who will use it to decrypt the database and import it into their local version of the WordPress site.

9.1 Backup the MySQL database

- Open an SSH Terminal
- SQL Command to backup (dump) a database
mysql dump -u [user] -p [db_name] > [output-db-filename.sql]
- Configure and Execute (run) the SQL command:
mysql dump -u root -p wordpress > my-wp-backup.sql

```
cabox@mxcsvrtest1:~/workspace$ mysqldump -u root -p wordpress > my-wp-backup.sql
Enter password:
cabox@mxcsvrtest1:~/workspace$ ls 0l
ls: cannot access '0l': No such file or directory
cabox@mxcsvrtest1:~/workspace$ ls -l
total 860
-rw-rw-r-- 1 cabox    www-data    268 Dec 11 14:00 composer.json
-rw-r--r-- 1 cabox    www-data   5429 Dec 11 14:01 composer.lock
-rw-rw-r-- 1 cabox    www-data    420 Dec  6 16:51 index.php
-rw-rw-r-- 1 cabox    www-data  19935 Dec  6 16:51 license.txt
-rw-rw-r-- 1 cabox    www-data 645085 Dec 11 15:40 my-wp-backup.sql
-rw-rw-r-- 1 cabox    www-data   7368 Dec  6 16:51 readme.html
drwxrwsr-x 6 cabox    www-data  4096 Dec 11 14:05 vendor
-rw-rw-r-- 1 cabox    www-data   6939 Dec  6 16:51 wp-activate.php
```



Backup the DB – command line

```
* Documentation: https://help.ubuntu.com/
Last login: Sat Dec 14 12:39:18 2019 from 52.161.27.120
cd ./workspace
cabox@mxcsrvrtest1:~$ cd ./workspace
cabox@mxcsrvrtest1:~/workspace$ mysqldump -u root -p wordpress > my-wp-backup.sql
Enter password:
cabox@mxcsrvrtest1:~/workspace$ ll
total 956
drwxrwsr-x  6 cabox    www-data   4096 Dec 14 13:05 .
drwxr-xr-x 12 cabox    cabox     4096 Dec 14 12:54 ../
-rw-r--r--  1 cabox    www-data   268 Dec 14 12:51 composer.json
-rw-r--r--  1 cabox    www-data  5429 Dec 14 12:07 composer.lock
-rw-r--r--  1 cabox    www-data  161 Dec 14 12:08 .env
-rw-r--r--  1 cabox    www-data  157 Dec 14 12:07 .env-sample
-rw-r--r--  1 cabox    www-data   83 Dec 14 12:07 gitignore
-rw-r--r--  1 www-data www-data  235 Dec 14 12:12 .htaccess
-rw-rw-r--  1 cabox    www-data  420 Dec 14 12:12 index.php
-rw-rw-r--  1 cabox    www-data 19935 Dec 14 12:12 license.txt
-rw-rw-r--  1 cabox    www-data 719671 Dec 14 13:05 my-wp-backup.sql
-rw-rw-r--  1 cabox    www-data  7368 Dec 14 12:12 readme.html
drwxrwsr-x  6 cabox    www-data   4096 Dec 14 12:51 vendor/
-rw-rw-r--  1 cabox    www-data  6939 Dec 14 12:12 wp-activate.php
drwxrwsr-x  9 cabox    www-data   4096 Dec 14 12:12 wp-admin/
-rw-rw-r--  1 cabox    www-data  369 Dec 14 12:12 wp-blog-header.php
-rw-rw-r--  1 cabox    www-data 2283 Dec 14 12:12 wp-comments-post.php
-rw-r----- 1 cabox    www-data 3697 Dec 14 12:54 wp-config.php
-rw-rw-r--  1 www-data www-data 2898 Dec 14 12:12 wp-config-sample.php
drwxrwsr-x  7 cabox    www-data   4096 Dec 14 12:28 wp-content/
-rw-rw-r--  1 cabox    www-data  3955 Dec 14 12:12 wp-cron.php
drwxrwsr-x 20 cabox    www-data 12288 Dec 14 12:12 wp-includes/
-rw-rw-r--  1 cabox    www-data 2504 Dec 14 12:12 wp-links-opml.php
-rw-rw-r--  1 cabox    www-data 3326 Dec 14 12:12 wp-load.php
-rw-rw-r--  1 cabox    www-data 47597 Dec 14 12:12 wp-login.php
-rw-rw-r--  1 cabox    www-data 8483 Dec 14 12:12 wp-mail.php
-rw-rw-r--  1 cabox    www-data 19120 Dec 14 12:12 wp-settings.php
-rw-rw-r--  1 cabox    www-data 31112 Dec 14 12:12 wp-signup.php
```

9.2 Encrypt the db backup file with GNU Privacy Guard

- What is GNU Privacy Guard?
 - GNU Privacy Guard (GnuPG), also known as GPG, is a tool for secure communication that was created by Werner Koch as Free Software under the GNU Project. ... GPG uses a combination of symmetric-key cryptography and public-key cryptography
 - Its also no-cost and included on our Codeanywhere WordPress connection server.
- Encrypt with database backup file using gpg command:
`gpg -c <<db_backup.sql>>`
- Verify that the encrypted file exists and has size

Run the gpg command, enter passphrase and repeat
gpg -c my-wp-backup.sql

```
View Preferences Help
| < > | < > | < > | < > | < > | < > | < >
* mxcsrvrtest1 x mxcsrvrtest1 x
cabox@mxcsrvrtest1:~/workspace$ gpg -c my-wp-backup.sql
```

```
View Preferences Help
| < > | < > | < > | < > | < > | < > | < >
* mxcsrvrtest1 x mxcsrvrtest1 x
cabox@mxcsrvrtest1:~/workspace$ gpg -c my-wp-backup.sql
gpg: gpg-agent is not available in this session
Enter passphrase: 
```

passphrase:
Winter2020

```
View Preferences Help
| < > | < > | < > | < > | < > | < > | < >
* mxcsrvrtest1 x mxcsrvrtest1 x
cabox@mxcsrvrtest1:~/workspace$ gpg -c my-wp-backup.sql
gpg: gpg-agent is not available in this session
Repeat passphrase: 
```

sql.gpg file verified

```
cabox@mxcsvrtest1:~/workspace$ gpg -c my-wp-backup.sql
gpg: gpg-agent is not available in this session
cabox@mxcsvrtest1:~/workspace$ ls -l
total 1040
-rw-rw-r-- 1 cabox www-data    268 Dec 11 14:00 composer.json
-rw-r--r-- 1 cabox www-data   5429 Dec 11 14:01 composer.lock
-rw-rw-r-- 1 cabox www-data    420 Dec  6 16:51 index.php
-rw-rw-r-- 1 cabox www-data  19935 Dec  6 16:51 license.txt
-rw-rw-r-- 1 cabox www-data 645085 Dec 11 15:40 my-wp-backup.sql
-rw-rw-r-- 1 cabox www-data 182137 Dec 11 15:49 my-wp-backup.sql.gpg ←
-rw-rw-r-- 1 cabox www-data   7368 Dec  6 16:51 readme.html
drwxrwsr-x 6 cabox www-data   4096 Dec 11 14:05 vendor
-rw-rw-r-- 1 cabox www-data   6939 Dec  6 16:51 wp-activate.php
drwxrwsr-x 9 cabox www-data   4096 Dec  6 16:51 wp-admin
-rw-rw-r-- 1 cabox www-data    369 Dec  6 16:51 wp-blog-header.php
-rw-rw-r-- 1 cabox www-data   2283 Dec  6 16:51 wp-comments-post.php
-rw-r----- 1 cabox www-data  3783 Dec 11 14:02 wp-config.php
```

Proof of encryption

.sql backup file

```
CREATE TABLE `wp_users` (
  `ID` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
  `user_login` varchar(60) COLLATE utf8mb4_unicode_520_ci NOT NULL DEFAULT '',
  `user_pass` varchar(255) COLLATE utf8mb4_unicode_520_ci NOT NULL DEFAULT '',
  `user_nicename` varchar(50) COLLATE utf8mb4_unicode_520_ci NOT NULL DEFAULT '',
  `user_email` varchar(100) COLLATE utf8mb4_unicode_520_ci NOT NULL DEFAULT '',
  `user_url` varchar(100) COLLATE utf8mb4_unicode_520_ci NOT NULL DEFAULT '',
  `user_registered` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',
  `user_activation_key` varchar(255) COLLATE utf8mb4_unicode_520_ci NOT NULL DEFAULT '',
  `user_status` int(11) NOT NULL DEFAULT '0',
  `display_name` varchar(250) COLLATE utf8mb4_unicode_520_ci NOT NULL DEFAULT '',
  PRIMARY KEY (`ID`),
  KEY `user_login_key` (`user_login`),
  KEY `user_nicename` (`user_nicename`),
  KEY `user_email` (`user_email`)
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_520_ci;
/*!14001 SET character_set_client = @saved_cs_client */;

-- 
-- Dumping data for table `wp_users`
-- 

LOCK TABLES `wp_users` WRITE;
/*!14000 ALTER TABLE `wp_users` DISABLE KEYS */;
INSERT INTO `wp_users` VALUES (1,'mxchase','$P$BiP89cISxYSPZWQS4sSWJv8pnMz49K1','mxchase:43:08','','0','mxchase'),(2,'userIT320','$P$8Dd2zZPRIuWIR./IS5jDt0HKRNOxD61','userit320','54','','0','user IT320');
/*!14000 ALTER TABLE `wp_users` ENABLE KEYS */;
UNLOCK TABLES;
/*!14013 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!140101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!140014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!140014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!140101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!140101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!140101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!140111 SET SQL_NOTES=@OLD_SQL_NOTES */;

-- Dump completed on 2019-12-11 15:40:38
```

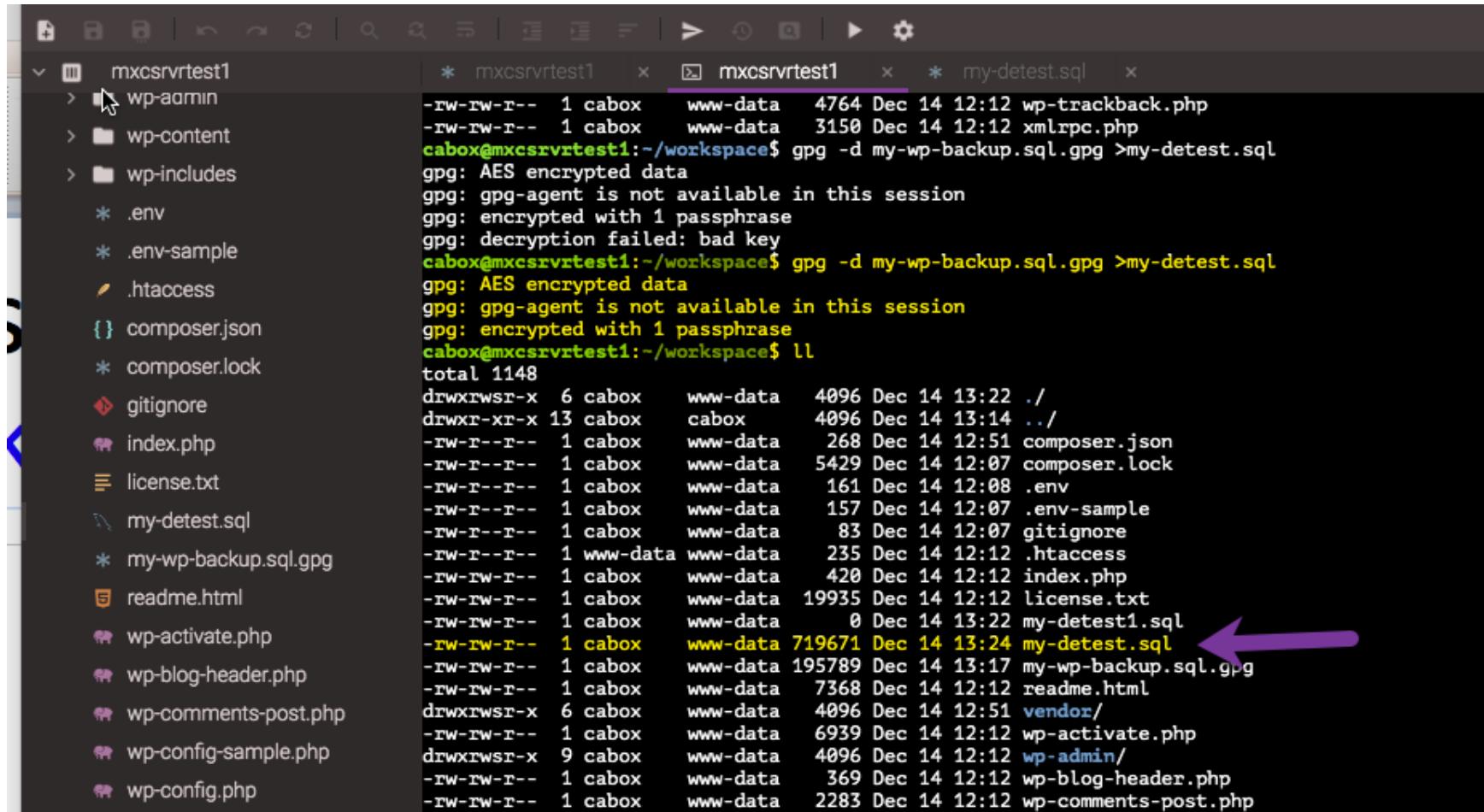
Encrypted file

Extra Step – Verify that you can decrypt the encrypted file

- I know it sounds crazy, but you must test that you can decrypt the encrypted DB backup file
- If you skip this step, and run into issues trying to decrypt it, you will never create a WordPress site.
- My Preferred Method:
 - Download the backup file to your desktop for safe keeping
 - Decrypt the encrypted backup to a new file name, like [wp-detest.sql](#)
 - Verify that the decrypted file has an appropriate file size, and cat it out (display its contents) to verify it is a valid copy the actual SQL Backup file.
- **REMEMBER to delete the decryption test file before creating the GitHub REPO or it will be pushed to GitHub**

Test decrypt

gpg -d my-wp-backup.sql.gpg >my-detest.sql

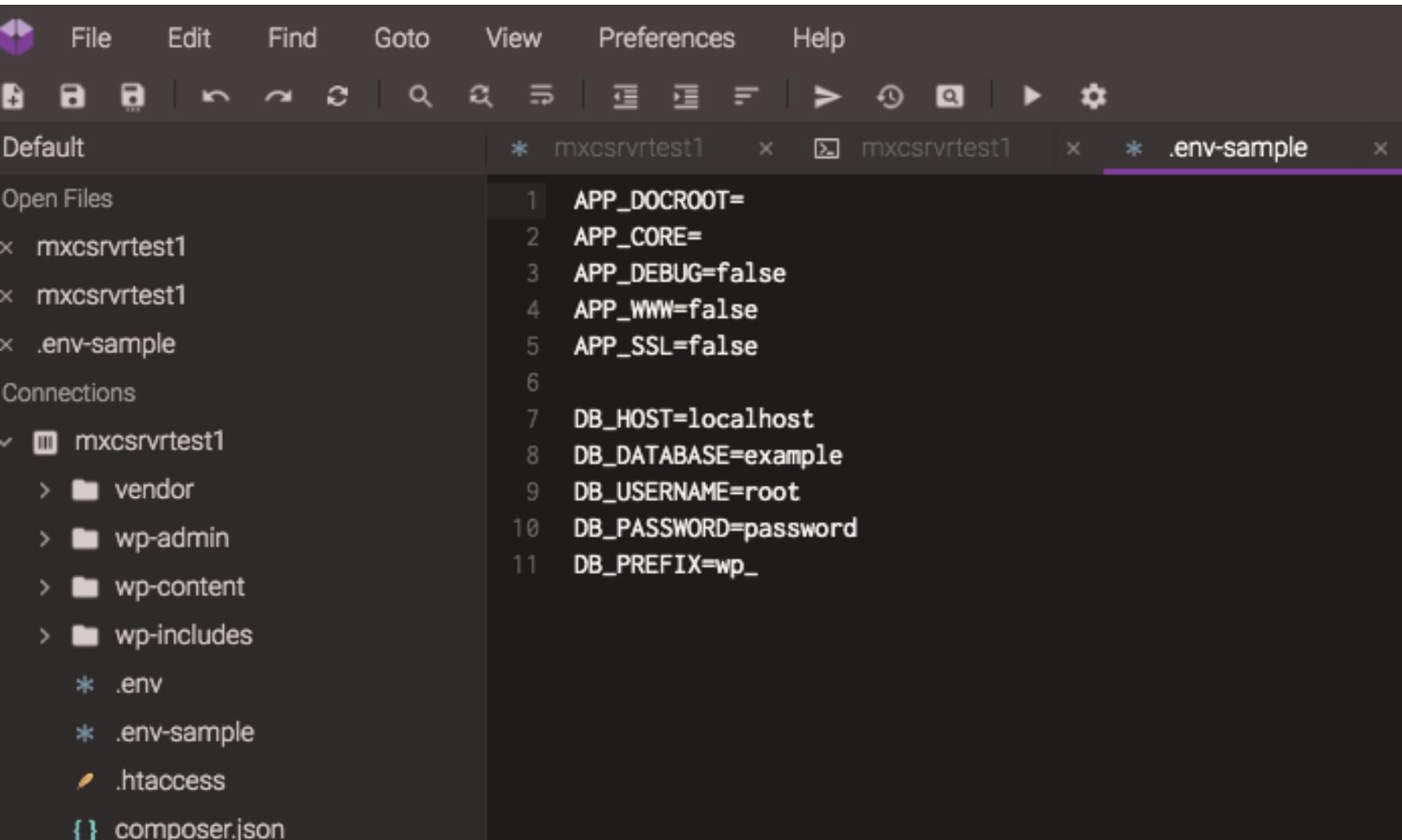


```
* mxcsrvrtest1 x mxcsrvrtest1 x * my-detest.sql x
-rw-rw-r-- 1 cabox www-data 4764 Dec 14 12:12 wp-trackback.php
-rw-rw-r-- 1 cabox www-data 3150 Dec 14 12:12 xmlrpc.php
cabox@mxcsrvrtest1:~/workspace$ gpg -d my-wp-backup.sql.gpg >my-detest.sql
gpg: AES encrypted data
gpg: gpg-agent is not available in this session
gpg: encrypted with 1 passphrase
gpg: decryption failed: bad key
cabox@mxcsrvrtest1:~/workspace$ gpg -d my-wp-backup.sql.gpg >my-detest.sql
gpg: AES encrypted data
gpg: gpg-agent is not available in this session
gpg: encrypted with 1 passphrase
cabox@mxcsrvrtest1:~/workspace$ ll
total 1148
drwxrwsr-x 6 cabox www-data 4096 Dec 14 13:22 .
drwxr-xr-x 13 cabox cabox 4096 Dec 14 13:14 ../
-rw-r--r-- 1 cabox www-data 268 Dec 14 12:51 composer.json
-rw-r--r-- 1 cabox www-data 5429 Dec 14 12:07 composer.lock
-rw-r--r-- 1 cabox www-data 161 Dec 14 12:08 .env
-rw-r--r-- 1 cabox www-data 157 Dec 14 12:07 .env-sample
-rw-r--r-- 1 cabox www-data 83 Dec 14 12:07 gitignore
-rw-r--r-- 1 www-data www-data 235 Dec 14 12:12 .htaccess
-rw-rw-r-- 1 cabox www-data 420 Dec 14 12:12 index.php
-rw-rw-r-- 1 cabox www-data 19935 Dec 14 12:12 license.txt
-rw-rw-r-- 1 cabox www-data 0 Dec 14 13:22 my-detest1.sql
-rw-rw-r-- 1 cabox www-data 719671 Dec 14 13:24 my-detest.sql ←
-rw-rw-r-- 1 cabox www-data 195789 Dec 14 13:17 my-wp-backup.sql.gpg
-rw-rw-r-- 1 cabox www-data 7368 Dec 14 12:12 readme.html
drwxrwsr-x 6 cabox www-data 4096 Dec 14 12:51 vendor/
-rw-rw-r-- 1 cabox www-data 6939 Dec 14 12:12 wp-activate.php
drwxrwsr-x 9 cabox www-data 4096 Dec 14 12:12 wp-admin/
-rw-rw-r-- 1 cabox www-data 369 Dec 14 12:12 wp-blog-header.php
-rw-rw-r-- 1 cabox www-data 2283 Dec 14 12:12 wp-comments-post.php
```

(9) Build New Repo on GitHub

1. Create an `.env-sample` file with purposely incorrect database credentials
2. Create a `.gitignore` file – so that we do not push the `.env` file or the `.sql` backup file
- 3. Login to your GitHub Account**
4. Create a new repo in GitHub -
5. Follow the GitHub instructions to populate the repository (repo)

.env-sample – we put this on the repo and not the .env file



The image shows a screenshot of a dark-themed code editor interface. At the top, there is a menu bar with options: File, Edit, Find, Goto, View, Preferences, and Help. Below the menu is a toolbar with various icons for file operations like Open, Save, and Find. The left sidebar displays the project structure under 'Default':

- Open Files:
 - mxcsrvrtest1
 - mxcsrvrtest1
 - .env-sample
- Connections:
 - mxcsrvrtest1
 - vendor
 - wp-admin
 - wp-content
 - wp-includes

In the main workspace, three tabs are visible:

- * mxcsrvrtest1
- * mxcsrvrtest1
- * .env-sample (highlighted with a purple underline)

The content of the .env-sample tab is as follows:

```
1 APP_DOCROOT=
2 APP_CORE=
3 APP_DEBUG=false
4 APP_WWW=false
5 APP_SSL=false
6
7 DB_HOST=localhost
8 DB_DATABASE=example
9 DB_USERNAME=root
10 DB_PASSWORD=password
11 DB_PREFIX=wp_
```

.gitignore file

```
#Ignore the local .env file
```

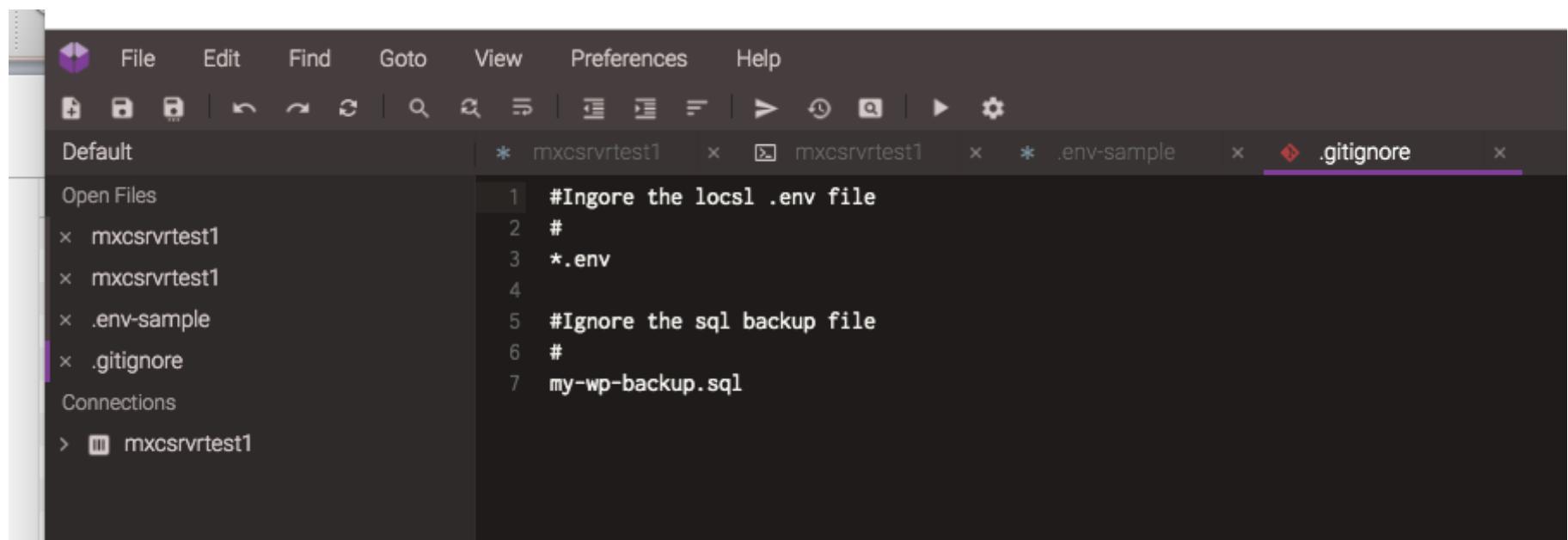
```
#
```

```
*.env
```

```
#Ignore the sql backup file
```

```
#
```

```
my-wp-backup.sql
```



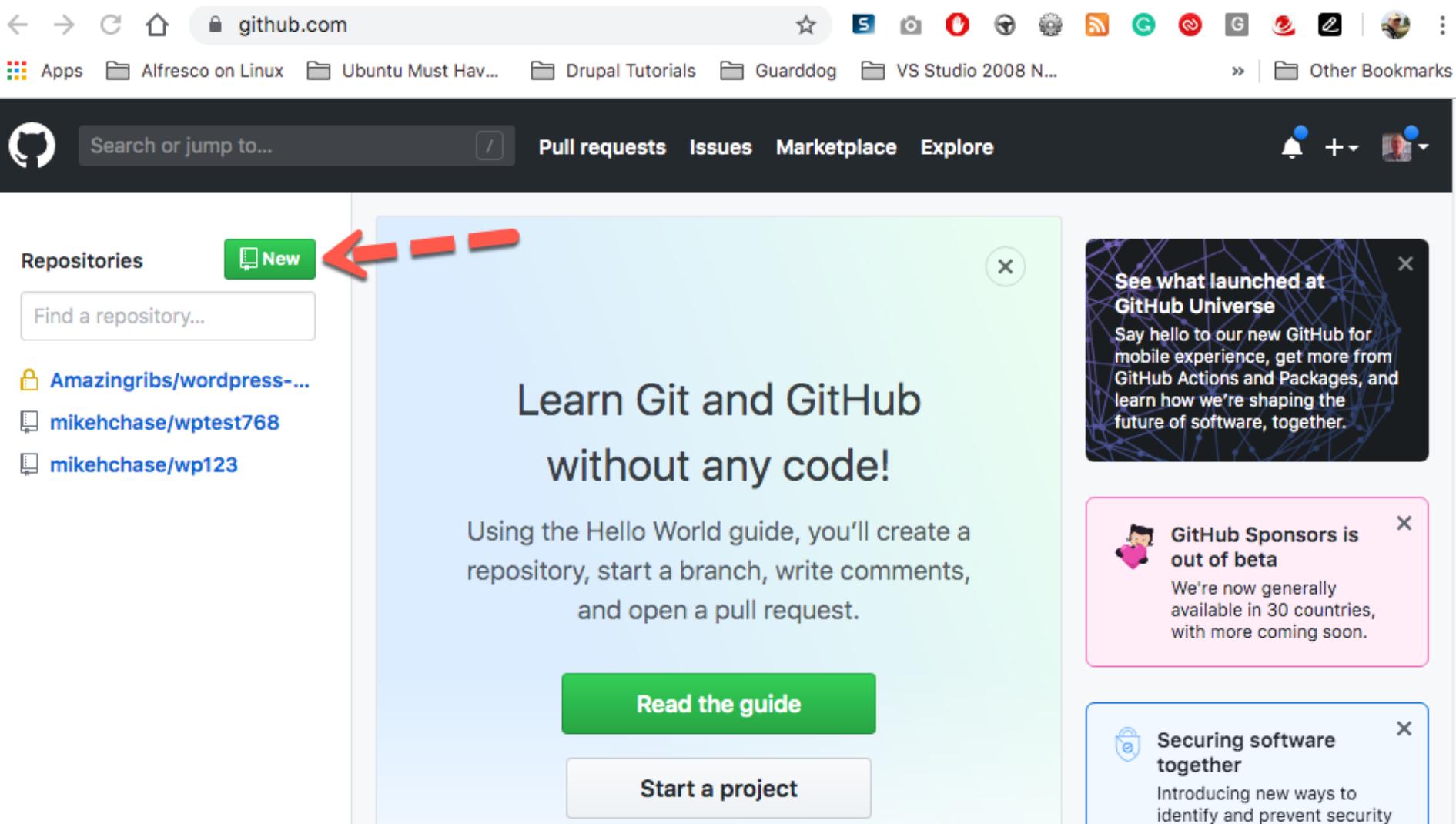
A screenshot of a code editor interface. The menu bar includes File, Edit, Find, Goto, View, Preferences, and Help. The toolbar below the menu has various icons for file operations like Open, Save, Find, and Copy. The left sidebar shows 'Default' and 'Open Files'. Under 'Open Files', there are four tabs: 'mxcsrvrtest1', 'mxcsrvrtest1', '.env-sample', and '.gitignore'. The '.gitignore' tab is currently active and highlighted with a purple bar at the bottom. The main editor area displays the following text:

```
1 #Ignore the local .env file
2 #
3 *.env
4
5 #Ignore the sql backup file
6 #
7 my-wp-backup.sql
```

Login to GitHub

The screenshot shows the GitHub homepage. The top navigation bar includes links for Apps, Alfresco on Linux, Ubuntu Must Hav..., Drupal Tutorials, Guarddog, VS Studio 2008 N..., and Other Bookmarks. The main content area features a large green banner with the text "Learn Git and GitHub without any code!" and two calls-to-action: "Read the guide" and "Start a project". To the right, there are promotional cards for GitHub Universe, GitHub Sponsors, and GitHub's security features.

Create a new Repo



A screenshot of the GitHub homepage. At the top, there's a navigation bar with links like 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. Below the navigation is a search bar and a 'New' button, which is highlighted with a large red arrow pointing to it from the left. To the right of the 'New' button is a green box containing text about learning Git and GitHub without code. Further down, there are two more boxes: one for 'GitHub Sponsors' and another for 'Securing software together'.

github.com

Apps Alfresco on Linux Ubuntu Must Hav... Drupal Tutorials Guarddog VS Studio 2008 N... Other Bookmarks

Search or jump to... / Pull requests Issues Marketplace Explore

Repositories New

Find a repository...

Amazingribs/wordpress-... mikehchase/wptest768 mikehchase/wp123

Learn Git and GitHub without any code!

Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.

Read the guide

Start a project

See what launched at GitHub Universe

Say hello to our new GitHub for mobile experience, get more from GitHub Actions and Packages, and learn how we're shaping the future of software, together.

GitHub Sponsors is out of beta

We're now generally available in 30 countries, with more coming soon.

Securing software together

Introducing new ways to identify and prevent security

Create the Repo

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Owner mikehchase  / it320winter2020  

Great repository names are short and memorable. Need inspiration? How about [ideal-pancake](#)?

Description (optional) 
 IT320 Winter 2020

  **Public**
Anyone can see this repository. You choose who can commit.

 **Private**
You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

Initialize this repository with a README 
This will let you immediately clone the repository to your computer.





GitHub Repo commands to run skip the readme command

There is a missing command to add

The screenshot shows a GitHub repository page for 'mikehchase / it320winter2020'. The page includes a navigation bar with links for Code, Issues (0), Pull requests (0), Actions, Projects (0), Wiki, Security, Insights, and Settings. Below the navigation bar, there's a section titled 'Quick setup — if you've done this kind of thing before' with instructions to 'Set up in Desktop' or 'HTTPS' or 'SSH' with the URL <https://github.com/mikehchase/it320winter2020.git>. A note says to 'Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.' At the bottom, there's a section titled '...or create a new repository on the command line' with the following steps:

```
echo "# it320winter2020" >> README.md
git init 1
git add README.md
git commit -m "first commit" 2
git remote add origin https://github.com/mikehchase/it320winter2020.git 3
git push -u origin master 4
```

GitHub Commands to initialize the Repo, add all the files, then commit and push

```
echo "# it320winter2020" >> README.md
```

```
git init
```

```
git add README.md
```

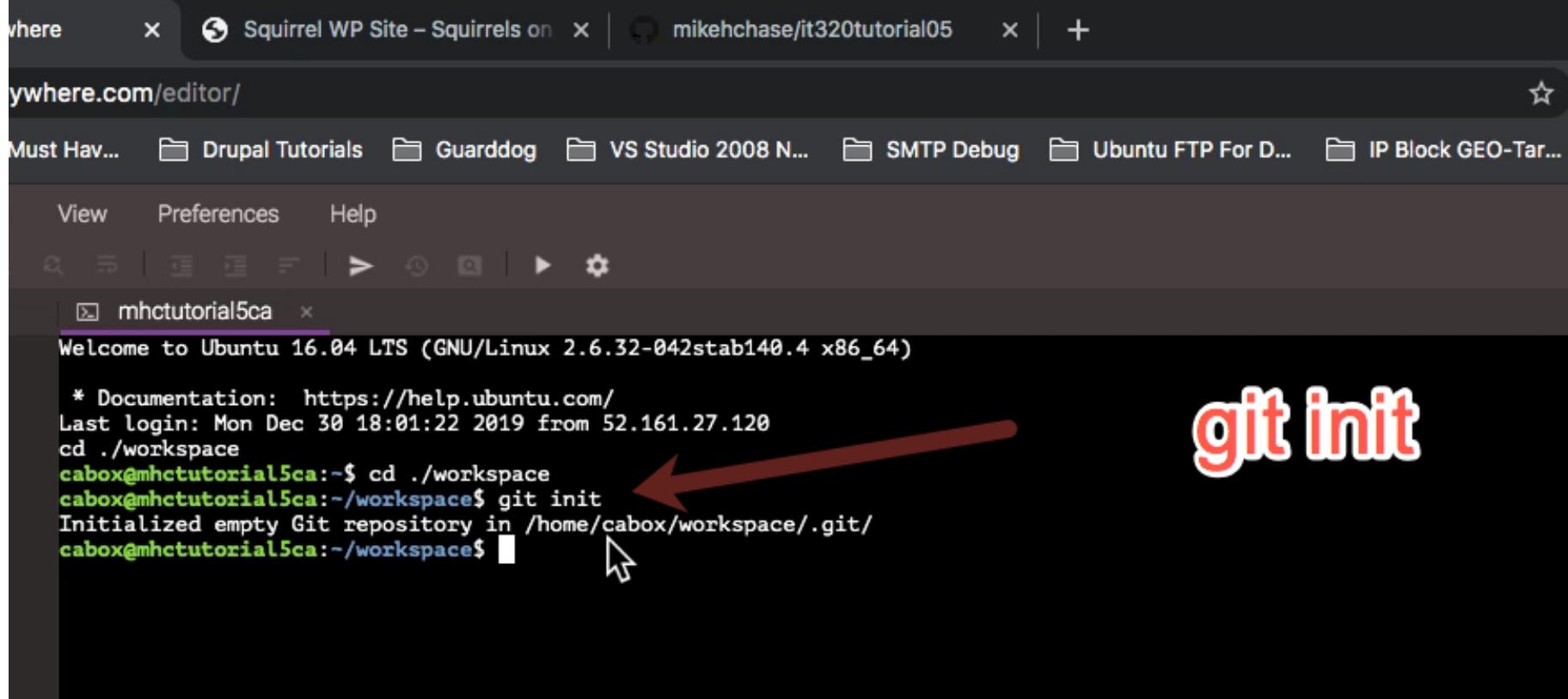
```
git add . ← This is the missing command – Its required
```

```
git commit -m "first commit"
```

```
git remote add origin https://github.com/mikehchase/it320winter2020.git
```

```
git push -u origin master
```

(1) git init command



A screenshot of a terminal window titled "mhctutorial5ca". The window shows a Linux command-line interface. A red arrow points from the word "git init" in a large white font on the right towards the command in the terminal. The terminal output includes:

```
Welcome to Ubuntu 16.04 LTS (GNU/Linux 2.6.32-042stab140.4 x86_64)

 * Documentation: https://help.ubuntu.com/
Last login: Mon Dec 30 18:01:22 2019 from 52.161.27.120
cd ./workspace
cabox@mhctutorial5ca:~/workspace$ git init
Initialized empty Git repository in /home/cabox/workspace/.git/
cabox@mhctutorial5ca:~/workspace$
```

(2) git add .

(3) git commit -m "first commit"

The screenshot shows a terminal window titled 'mhctutorial5ca' running on an Ubuntu 16.04 LTS system. The terminal output is as follows:

```
Welcome to Ubuntu 16.04 LTS (GNU/Linux 2.6.32-042stab140.4 x86_64)

 * Documentation: https://help.ubuntu.com/
Last login: Mon Dec 30 18:01:22 2019 from 52.161.27.120
cd ./workspace
cabox@mhtutorial5ca:~/workspace$ cd ./workspace
cabox@mhtutorial5ca:~/workspace$ git init
Initialized empty Git repository in /home/cabox/workspace/.git/
cabox@mhtutorial5ca:~/workspace$ git add .
cabox@mhtutorial5ca:~/workspace$ git commit -m "first commit"
```

The tail end of the commit command

- all files (except ignore, getting uploaded)

```
create mode 100644 wp-includes/theme-compat/footer.php
create mode 100644 wp-includes/theme-compat/header-embed.php
create mode 100644 wp-includes/theme-compat/header.php
create mode 100644 wp-includes/theme-compat/sidebar.php
create mode 100644 wp-includes/theme.php
create mode 100644 wp-includes/update.php
create mode 100644 wp-includes/user.php
create mode 100644 wp-includes/vars.php
create mode 100644 wp-includes/version.php
create mode 100644 wp-includes/widgets.php
create mode 100644 wp-includes/widgets/class-wp-nav-menu-widget.php
create mode 100644 wp-includes/widgets/class-wp-widget-archives.php
create mode 100644 wp-includes/widgets/class-wp-widget-calendar.php
create mode 100644 wp-includes/widgets/class-wp-widget-categories.php
create mode 100644 wp-includes/widgets/class-wp-widget-custom-html.php
create mode 100644 wp-includes/widgets/class-wp-widget-links.php
create mode 100644 wp-includes/widgets/class-wp-widget-media-audio.php
create mode 100644 wp-includes/widgets/class-wp-widget-media-gallery.php
create mode 100644 wp-includes/widgets/class-wp-widget-media-image.php
create mode 100644 wp-includes/widgets/class-wp-widget-media-video.php
create mode 100644 wp-includes/widgets/class-wp-widget-media.php
create mode 100644 wp-includes/widgets/class-wp-widget-meta.php
create mode 100644 wp-includes/widgets/class-wp-widget-pages.php
create mode 100644 wp-includes/widgets/class-wp-widget-recent-comments.php
create mode 100644 wp-includes/widgets/class-wp-widget-recent-posts.php
create mode 100644 wp-includes/widgets/class-wp-widget-rss.php
create mode 100644 wp-includes/widgets/class-wp-widget-search.php
create mode 100644 wp-includes/widgets/class-wp-widget-tag-cloud.php
create mode 100644 wp-includes/widgets/class-wp-widget-text.php
create mode 100644 wp-includes/wlwmanifest.xml
create mode 100644 wp-includes/wp-db.php
create mode 100644 wp-includes/wp-diff.php
create mode 100644 wp-links-opml.php
```

(4) git remote add origin master...

(5) git push -u origin-master

```
create mode 100644 wp-mail.php
create mode 100644 wp-settings.php
create mode 100644 wp-signup.php
create mode 100644 wp-trackback.php
create mode 100644 xmlrpc.php
cabox@mhtutorial5ca:~/workspace$ git remote add origin https://github.com/mikehchase/it320tutorial05.git
cabox@mhtutorial5ca:~/workspace$ git push -u origin master
Username for 'https://github.com': [REDACTED]
```

.sql
.sql.gpg
onsole

....sql wp_person.sql

To complete the GitHub Commands

- You will need to know your GitHub username and password

```
create mode 100644 wp-links-opml.php
create mode 100644 wp-load.php
create mode 100644 wp-login.php
create mode 100644 wp-mail.php
create mode 100644 wp-settings.php
create mode 100644 wp-signup.php
create mode 100644 wp-trackback.php
create mode 100644 xmlrpc.php
cabox@mhtutorial5ca:~/workspace$ git remote add origin https://github.com/mikehchase/it320tutorial05.git
cabox@mhtutorial5ca:~/workspace$ git push -u origin master
Username for 'https://github.com': mikehchase
Password for 'https://mikehchase@github.com':
Enumerating objects: 2636, done.
Counting objects: 100% (2636/2636), done.
Delta compression using up to 2 threads
Compressing objects: 100% (2580/2580), done.
Writing objects: 65% (1714/2636), 9.77 MiB | 5.69 MiB/s
```

(6) GitHub username and password

```
create mode 100644 wp-links-opml.php
create mode 100644 wp-load.php
create mode 100644 wp-login.php
create mode 100644 wp-mail.php
create mode 100644 wp-settings.php
create mode 100644 wp-signup.php
create mode 100644 wp-trackback.php
create mode 100644 xmlrpc.php
cabox@mhtutorial5ca:~/workspace$ git remote add origin https://github.com/mikehchase/it320tutorial05.git
cabox@mhtutorial5ca:~/workspace$ git push -u origin master
Username for 'https://github.com': mikehchase
Password for 'https://mikehchase@github.com':
Enumerating objects: 2636, done.
Counting objects: 100% (2636/2636), done.
Delta compression using up to 2 threads
Compressing objects: 100% (2580/2580), done.
Writing objects: 65% (1714/2636), 9.77 MiB | 5.69 MiB/s
```

GitHub Repo - notice .env-sample

The screenshot shows a GitHub repository page for 'mikehchase/it320tutorial05'. The repository name is displayed at the top left, along with a star icon, a fork icon, and a 'Unwatch' button. Below the header, there are tabs for 'Code', 'Issues 0', 'Pull requests 0', 'Actions', 'Projects 0', 'Wiki', 'Security', 'Insights', and 'Settings'. The 'Code' tab is selected. The main content area displays the commit history for the 'master' branch. The commits are listed from newest to oldest, with each commit showing the author, file changes, message, and timestamp. The latest commit is by 'mchase2' and is a 'first commit' made 1 minute ago. Other files listed include 'vendor', 'wp-admin', 'wp-content', 'wp-includes', '.env-sample', '.gitignore', and '.htaccess'. At the bottom of the commit list, there are buttons for 'Create new file', 'Upload files', 'Find file', and a green 'Clone or download' button.

File	Message	Time
mchase2 first commit	first commit	1 minute ago
vendor	first commit	1 minute ago
wp-admin	first commit	1 minute ago
wp-content	first commit	1 minute ago
wp-includes	first commit	1 minute ago
.env-sample	first commit	1 minute ago
.gitignore	first commit	1 minute ago
.htaccess	first commit	1 minute ago

Encrypted SQL Backup file (.sql.gpg) is in the Repo, but not the .sql backup file

license.txt	first commit	6 minutes ago
my-wp-backup.sql.gpg	first commit	6 minutes ago
readme.html	first commit	6 minutes ago
wp-activate.php	first commit	6 minutes ago
wp-blog-header.php	first commit	6 minutes ago
wp-comments-post.php	first commit	6 minutes ago
wp-config-sample.php	first commit	6 minutes ago
wp-config.php	first commit	6 minutes ago
wp-cron.php	first commit	6 minutes ago
wp-links-opml.php	first commit	6 minutes ago
wp-load.php	first commit	6 minutes ago
wp-login.php	first commit	6 minutes ago
wp-mail.php	first commit	6 minutes ago
wp-settings.php	first commit	6 minutes ago
wp-signup.php	first commit	6 minutes ago
wp-trackback.php	first commit	6 minutes ago
xmlrpc.php	first commit	6 minutes ago

Now that the Repo exists

- We can create a new WordPress site on Codeanywhere, or any web host that supports building from GitHub and test it out
- SiteGround Student version does not support this, but that's OK, we have Codeanywhere, and we can use it and it will serve us just as well.
- The next Presentation will cover building a WordPress site from GitHub

The new Repository on GitHub

A screenshot of a GitHub profile page for user `mikehchase`. The page includes a profile picture of Michael H Chase, a link to his [Edit profile](#), and a bio stating he is a Chicago Illinois native, Instructor at DePaul University, CMS and Framework developer. The main section shows an [Overview](#) tab selected, along with links for [Repositories](#) (4), [Projects](#) (0), [Packages](#) (0), [Stars](#) (0), [Followers](#) (16), and [Following](#) (2). Below this, there's a section for [Popular repositories](#) featuring `wp123` (Codeanywhere WordPress 123, PHP), `it320winter2020` (IT320 Tutorial 05, PHP), and `wptest768` (wordpress test site, PHP). The repository `it320tutorial05` is highlighted with a red border. At the bottom, there's a [Contribution settings](#) dropdown and a heatmap showing contributions per month and day.

Popular repositories

- `wp123`
Codeanywhere WordPress 123
PHP
- `it320winter2020`
IT320 Tutorial 05
PHP
- `wptest768`
wordpress test site
PHP

it320tutorial05

IT320 Tutorial 05

PHP

12 contributions in the last year

Contribution settings ▾

Less More