Installing WordPress under XAMPP (on Mac)

Steps:

○ Download XAMPP installer from: https://sourceforge.net/projects/xampp/

○ On a Mac you may have to consult this: fixing-macos-cannot-verify-app-free-malware

○ Possible: Nginx detour: You may see this if you have already installed NGINX:

○
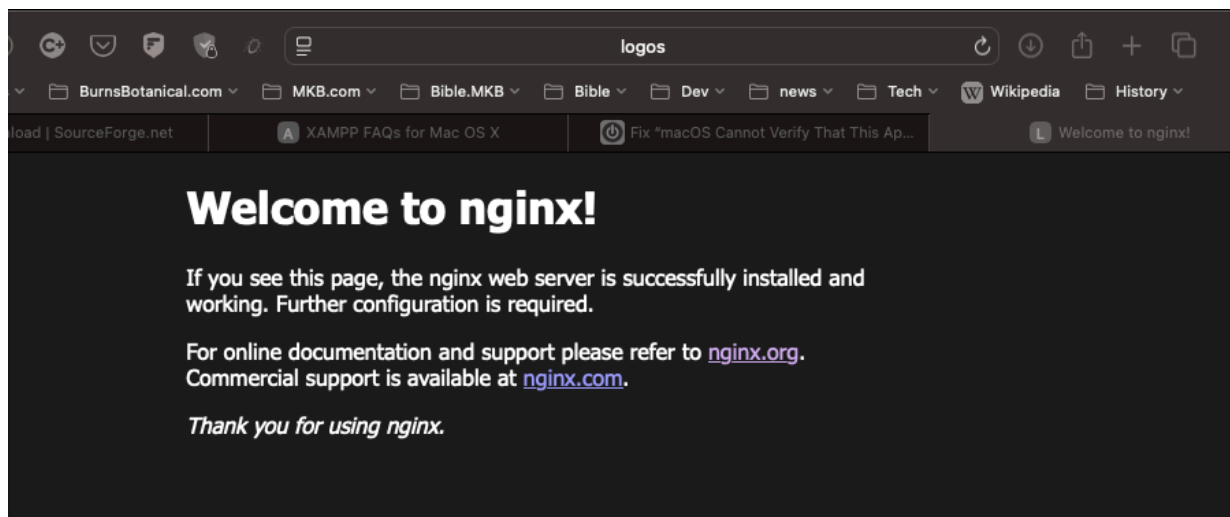
Click on the nginx.org link.   But in my case there was nothing left to configure. However, since I have nginx running already and running Apache, I must stop Nginx with this:

        sudo nginx -s stop

○ In the control panel for XAMPP (which differs for different OSs (Mac, Linux, Windows))

    ○ Start at least MySQL Database and Apache Web Server

○ When those services are up, the main control panel has a button: "Go to Application" which opens a tab on your web browser and opens the dashboard of XAMPP.   It should look like this:

❸ XAMPP Apache + MariaDB + PHP + Perl

Welcome to XAMPP for OS X 8.2.4

You have successfully installed XAMPP on this system! Now you can start using Apache, MariaDB, PHP and other components. You can find more info in the FAQs section or check the HOW-TO Guides for getting started with PHP applications.

XAMPP is meant only for development purposes. It has certain configuration settings that make it easy to develop locally but that are insecure if you want to have your installation accessible to others.

Start the XAMPP Control Panel to check the server status.

Community

XAMPP has been around for more than 10 years – there is a huge community behind it. You can get involved by joining our Forums, liking us on Facebook, or following our exploits on Twitter.

○ You should explore this top level links:
- ○ Apache Friends
- ○ FAQs
- ○ HOW-TO Guides
- ○ PHPInfo
- ○ phpMyAdmin

○ Find where XAMPP files are stored on your filesystem:
- ○ For Mac it should be /Applications/XAMPP
- ○ For Windows:
- ○ For Linux:
- ○ Now explore that file tree.
    - ○ bin
    - ○ cgi-bin
    - ○ etc
    - ○ htdocs
    - ○ logs
    - ○ For Mac: manager-osx.app
    - ○ uninstall

○ xamppfiles (This is where the bulk of the files are)

○

○ Install WordPress inside your XAMPP

　○ https://wordpress.org/

　○ https://wordpress.org/download/

　　○ NOTE: this next link is the howto-install document. Give it a look over, but DON'T start any steps yet. There is a link for 'Before You Install' that you should follow and read first. This is because there are preliminary steps that we want to get captured in a Git repository before doing any modifications. The first paragraph of the 'howto' talks about tools available to do some of the install for you. The purpose of this exercise is to learn about the internals as shipped to we can explore add-ons and modifications we can do to WordPress.

　○ https://developer.wordpress.org/advanced-administration/before-install/howto-install/

　○ My install on my laptop is at /Applications/XAMPP/xamppfiles/htdocs/RDwordpress

　○ Before doing the manual install steps, first turn the top level WP directory (RDwordpress) into a git repository and add that to your GitHub account.

　○ When creating the repository GitHub Desktop will give you the option of selecting a .gitignore file. I select the 'wordpress' version. However, once you do that very little of the content will be tracked as evidence of using a 'git status' command. You should edit that .gitignore as suggested in the next step.

　○ I commented out many of the files and directories except leaving /wp-content/themes/twenty*/ and /wp-content/plugins/hello.php, *.log, and .htaccess . This is because while doing research and development I might choose to modify some of the original files and I want to track those changes.

　○ My GitHub repository is https://github.com/MichaelKentBurns/RDwordpress.git so you can consult it and see exactly what I changed during this install and future research and development work.

　○ Before you start

　○ https://developer.wordpress.org/advanced-administration/wordpress/wp-config/

　○ While editing the wpconfig.php file with database name, username and password and when you commit and push that to GitHub, you

should get an email from 'GitGuardian' telling you you just exposed some secret information to the world. Take a look at that and follow some of the links in the email so you understand the issue. Normally, this is a serious matter so don't ignore it.   If someone were to peek into my public repository they would have valuable secrets that would let them viciously attack my new wordpress site.   HOWEVER, in this specific case where I am creating a wordpress site inside XAMPP on my local and private laptop, I'm safe for several reasons. 1) This is a learning exercise and not a production wordpress site there is nothing sensitive in it, and in fact I'm committing everything to GitHub so I can detect any malicious changes. 2) All of this is on my private, password protected, and physically under my control at all times, so nobody can access it at all. (I do not provide any network access to this machine inside the firewall of my private network, and I don't take this laptop to public places.   So learn this lesson, and heed these warnings whenever you encounter them.   If you hope to get work to deal with other people's WordPress sites, you are responsible for their privacy and it is a serious matter that could affect your career.

Once you have edited your wp-config you are ready to activate the self configuration script. As it says in the howto-install doc mentioned above, simply point your browser to your localhost wordpress wp-admin/install.php page, which in my case is https://localhost/RDwordpress/wp-admin/install.php   That page will ask a few questions and then it will quickly configure your wordpress and end up in the wp-admin dashboard.   Two of the questions will be the username for a new admin account in your WP install and the password to go with it.   (I need to go through that again and screen capture the details here).   After it is finished you will be able to login to your new wordpress site admin panel using that username/ password.   The URL will be something like https://localhost/RDwordpress/wp-admin/ . From there you can look down the tool bar on the left for the 'Users' page. There you will find yourself with the role of Administrator.   Keep that username and password private, and don't use a password that you use for anything else.   Logging in as Administrator allows you full control of every detail of your wordpress instance!!

At this point you should be able to use a shell or GitHub Desktop and see that there are 'No local changes'. But, where did all the details like the admin username/pw go? If it's not in any files just setup, then where is it? The answer is that all of that, and a lot more, are stored in the mySQL database which is not itself a file in the WordPress directory.   So, how do you see it? How does that get into your Git

repository? Good Question!

○ Remember the phpMyAdmin that you used to create the database? Go there in a new tab that you open to the XAMPP dashboard page (https://localhost/). Then look at the tabs in the blue banner and find phpMyAdmin. Find the RDwordpress database in the left navigation bar. Open that and see the data tables.

○ When you click on one of the tables you will see the records.   Try RDwp_options.   That table contains the main options that control your site.   Also look at RDwp_users and see your admin user account.

○ Let's dump that data into files in your git repository. Use a shell to create three hidden directories next to the .git directory: mkdir .data .pages .posts

○ Use ls -la to see that directory and you should see things like wp-content, index.php and a bunch of other wp-* files and directories. You should probably make those directories private: chmod 700 .data .pages .posts

○ Back at the table view of phpMyAdmin under the table of rows, you will see Query results operations.   Choose Export. In that form, take all the defaults until you get to Export method: For that choose Custom and that will open up a number of other options, one of which will be 'Output:'   First, just select 'View output as text' and click the Export button at the bottom. This will show you what the dump will look like.

○ After browsing that some, change the output to 'Save output to a file.' You should find that file in the Downloads directory that your browser defaults to.   Repeat this for each table. You will notice that initially some of the tables will not have any rows, and thus the Export won't work.   But if there are rows then you can export. Don't forget to change the Export method: to "Custom".   You should end up with about 9 output files.

○ Copy those from your downloads directory to the .data directory in your wp instance.

○ Now go back to GitHub Desktop and you should see those files show up as changes.   Commit those with the comment 'Initial data dump' and push it.   Then find them in your GitHub repository.   You have now exported your database.

○ Later on we will automate that data dump process.

○

○ You now have a full server stack with a WordPress instance installed on

your development machine.