WikiJS, Open Source, Powerful, Configurable, Intuitive

https://www.youtube.com/embed/Dd8_plibBYk

WikiJS is a tremendously powerful, extremely full-featured open source, self hostable Wiki system. It has a modern user interface, and tons of settings, configuration, and customization options.

Installing it with Docker-CE and Docker-Compose makes it a breeze to get WikiJS up and running in no time.

Today, we'll go through setting up WikiJS, and the basics of configuration and usage.

What you'll need

- Docker-CE
- Docker-Compose
- (optional) Portainer-CE (GUI for Docker)
- (optional) NGinX Proxy Manager (for full url access and SSL)
- About 15 minutes of your time.

Installing Docker-CE, Docker-Compsoe, Portainer-CE, and NGinX Proxy Manager

In the video, I show you how to create a user with sudo privileges, so you aren't setting up these things as root. It only takes a minute to do, so definitely take that step if you haven't already.

You can easily install Docker-CE, Docker-Compose, Portainer-CE, and NGinX Proxy manager by using this quick install script I created and maintain on Github. Just use the command:

```
wget https://raw.githubusercontent.com/bmcgonag/docker_installs/main/install_docker_nproxyman.sh
```

To download the script to your desired host.

Change the permissios to make the script executable:

`chmod +x ./install_docker_nproxyman.sh`

and then run the script with the command:

`./install_docker_nproxyman.sh`

When run, the script will prompt you to select your host operating system, then will ask you which bits of software you want to install.

Simply enter 'y' for each thing you want to install.

At some point, you'll be asked for your super user (sudo) password as well.

Allow the script to complete installation.

At this point, you might want to log out and back in, as this will allow you to use the `docker` and `docker-compose` commands without the need of "sudo" in front of them.

Installing WikiJS

Now that we have Docker-CE and Docker-Compose installed, we can move forward with getting WikiJS installed and setup. First, let's create a folder structure to help keep our docker applications organized and easy to backup.

Let's create a "docker" folder to be the top level (parent) folder. If you already have an organizational structure for your docker installs, then don't worry about this part.

And we'll move into that parent level folder:

`mkdir docker`

`cd docker`

Now, let's make a directory for our WikiJS install and move into it.

`mkdir wikijs`

`cd wikijs`

Now that we are inside our "wikijs" folder, we'll create a file called "docker-compose.yml"

`nano docker-compose.yml`

We need to copy the following yaml code block, and paste it into the "docker-compose.yml" file we have open for editing. So, highlight the code-block below, and copy it.

```
version: "3"
services:
  wikijs:
    image: lscr.io/linuxserver/wikijs
    container_name: wikijs
    environment:
      - PUID=1000
      - PGID=1000
      - TZ=America/Chicago
    volumes:
      - ./config:/config
      - ./data:/data
    ports:
      - 3000:3000
    restart: unless-stopped
```

Paste the code-block into the open nano editor with a right-click, paste, or with the hotkey combination CTRL + Shift + V.

In the file, you'll want to note a few places, where you may want to make changes.

First, let's look at the port mapping. 3000 is a very common port for lots of applications, so it's generally a good idea to change this to some other port on your host machine. The "host" machine is the physical system you are putting the dockerized application on.   The "container" is the application space where the app runs (separated / sandboxed away from the physical OS). The port and volume mappings are setup as

`<host>:<container>`

So, feel free to change the host side of a volume mapping or port mapping in the docker-compose.yml file.   Just do not change the container side unless you are a developer, and know what you are doing.

For my setup, we used port 9190 instead of port 3000, so my port mapping looked like:

```
    ports:
      - 9190:3000
```

Additionally, if you are using the stacks feature in Portainer-CE to run this, instead of a docker-compose.yml file, change the volume mappings to use the absolute (full) path to the wikijs folder we created, instead of the relative `./config` and `./data` paths we have in the docker-compose.yml file here.

In my case, for a Portainer-CE stack, I would have my volume mappings set like:

```
volumes:
  - /home/brian/docker/wikijs/config:/config
  - /home/brian/docker/wikijs/data:/data
```

Next, you'll want to make sure your PGID and PUID are set correctly. You can check this, by opening another terminal window (or SSHing into your host machine with a second terminal), and entering the command:

`id`

Then comparing the values in the output to the values in the docker-compose.yml file.

Finally, set your timezone correctly for the TZ variable.

Once everything is set in your docker-compose.yml file, save it with CTrl + O, then press enter to confirm, and close the nano editor with CTRL + X.

Now, we are ready to run the docker-compose file and pull down the WikiJS image, and start the container.

Just enter:

`docker-compose up -d`

In your terminal (from inside the ./docker/wikijs directory), and allow the image to download, and start.

When the container is done starting you should see `done` in the terminal. Make sure you don't see any errors in the terminal output. If everything went well, you should now be able to use a web browser to go to your host machine's IP and the port you entered in the docker-compose.yml file, and be presented with the first run wizard of WikiJS.

Check out the video linked at the top of this article to get details on the basic configuration and usage of WikiJS, as well as how to setup your Wiki with a FQDN (Fully Qualified Domain Name) and LetsEntcrypt SSL Certificate.

Support my channel and content

Support my Channel and ongoing efforts through Patreon:

https://www.patreon.com/bePatron?u=234177

Revision #1
Created 12 December 2022 18:00:06 by Brian McGonagill
Updated 12 December 2022 18:01:03 by Brian McGonagill