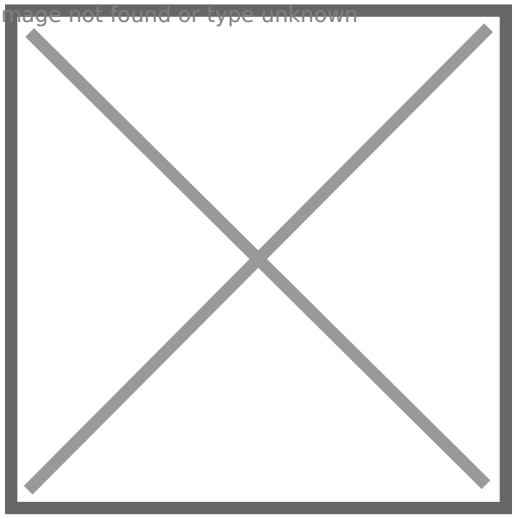
Moving Files & Folders Around The Container

Pterodactyl does not have documentation on moving files and folders around the container you are in as it believes you already have the knowledge as you were the one that installed the panel in the first place, as in, knowing Linux commands. The trouble is, most panel installs are done by system admins or our team directly, so this unfortunately doesn't get informed. In simple terms, we need to learn what **../**, **.**, and what a **directory** means.

A directory is a folder. It's a term used in Linux but also Windows, but the term is used for more computer proficient users. In a **Linux** system, in the **root/beginning/start** location, like the **C:** drive on Windows, there are directories which run the system, such as **/root /var /srv /home** and more. See below for an example.

Image not found or type unknown

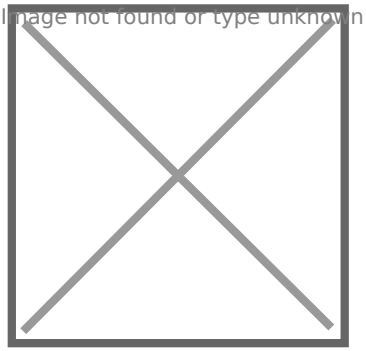


In Pterodactyl, your servers are stored in **/var/lib/pterodactyl/volumes**. You might be thinking "Wait, var is variable files? That doesn't make sense?" Well, it includes directories such as **/var/lib** for MySQL/MariaDB or MongoDB storage and **/var/www** for website files like the Ptero panel. You've got to see it as **These are variable > They change > Hence the name**. You are storing data in them that will change by the user or the program using them.

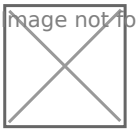
Now, in Pterodactyl, each server is contained as a UUID, which is explained **here**, see below:



Now, in the panel itself, these are mounted as **/home/container**, if you check under the File Manager, you'll see this. You can't go further **"back"** from /home/container such as **/home** or even **/**, so this area is your **root**, your **beginning**, your **start**. When we start moving things soon, you will soon realise we can't go back any further.



The main **relative path** operand we use is **../**. We call this **dot dot slash**, easy to say, simple to do. For example, if I was in **/root** and I wanted to move the file called **steve.txt** to **/srv**, I would do **mv /root/steve.txt /srv**. Now, that might be very confusing at first, but basically the command **mv** means **Move**. We can go and see that file with **cd /srv** which will take us to that directory and it means **Change Directory**. Doing **ls** will list all the items in that directory, showing us the **steve.txt**.





(~ means root in this context don't worry about it, but you can see we moved it in /srv.)

If you were using the other **relative path** operand, **.**, mostly just called **dot**, you are telling it to **move** here/**come** here. So, if I was in **/root** and I wanted to **move /srv/steve.txt** to me, I would do **cd /root**, then **mv /srv/steve.txt .**. That dot means it "put it where I am" essentially.

Now, how do we apply this knowledge to the panel? Well, let's say we wanted to move data from a folder named **Server 1** and put it into the **root** of the container, so we

can start the server with that data.

Now onto actually doing it!

The panel automatically does **mv** when pressing the move button, but where does **../** come into play? Well, when we want to move a file or folder backwards, we do **../** at the front. Each time we do it, it moves back another directory. Watch below:

> *Watch the **New Location:** at the bottom of the box. This will really help nail it first time!*



You'll also notice that to move something back into a folder, we go to the front, and pop in where it needs to go. When knowing that **../** goes **back**, and **typing the name** goes **forward**, you've now mastered moving your server files around!

Revision #2
Created 7 December 2024 20:26:04 by JasmeowTheCat
Updated 8 December 2024 19:43:28 by JasmeowTheCat