Why Is X Not Installed On The System - How Docker Images Work

When connecting to your systems via SSH, you may notice that certain applications don't exist, are not installed or not working as they should be. This is due to how docker works and it's image system.

If you are a client who has Pterodactyl installed, servers are managed in containers, which are managed by the software called "Docker". Each server is contained in a container, simply referred to as a "house." Those little houses live together to run anything you wish to in their own little "building". They can't talk to other "houses" (*Unless internal, 172.18.0.1 or your own subnet)*, they can't interact with them, they can't touch them, they are **separate**. They can run a magnitude of software such as Java, NodeJS, Python, SteamCMD, the list goes on. Now, these are managed with **docker images**. Images are released by many users online to help streamline the process of running a certain application. For example, let's take a java one from Pterodactyl's repository on GHCR:

- ghcr.io/pterodactyl/yolks:java_17

What we can see here is that it's using the website **ghcr.io**, which is where people can share and publish docker images. Next up is **/pterodactyl**, which is the author of the images, then **java_17**, meaning that Java 17 will be installed into that **container** when deployed. Deploying means that you've gone to your panel, created a server, selected Java 17 as the image, and the "house" gets "built." Using this analogy should hopefully now make sense. **Nothing** is installed on the **root** level, **doesn't touch** the system whatsoever and doesn't interfere with **other** houses.

This means that when you run **java --version**, **node -v** or other such command, it won't show anything up unless we installed that application at some point for testing or development. Any questions, ask the team.

Revision #2
Created 29 February 2024 20:11:06 by JasmeowTheCat
Updated 14 January 2025 06:31:18 by JasmeowTheCat