0FS (Zero Footprint Software)

The Bonsai Framework term **0FS (Zero FootPrint Software)** is in reference to an approach to selecting and packaging products in a way that is portable and environment agnostic. I also find this approach increases the compatibility of moving your system to Cloud based architecture services and microservices.

> A 0FS (Zero Footprint Software) should be self-contained at two levels.
>
> **Level 1** - download the file containing runtime, runtime engine, container, configs. Be able to untar and then start the software. Then as the system grows, easily move the 0FS to a new physical/virtual server by with tar, copy and untar. The only caveat is to use consistent user iDs and GUID's across systems.
>
> The Zero Footprint Software should look exactly the same when doing a binary comparison between environments. Even environment property files should be replicated to be exactly the same.
>
> Alternatively environmental differences can be loaded with the appropriate property file based on a singular system variable.
>
> Transient information should be packaged outside of the package using a consistent naming convention.
>
> Everything required for the software (like a Bonsai Tree should) should be included as one package. With this philosophy, for example, a container like Tomcat should be packaged with it's own instance of Java. There should be no direct dependency on the operating system libraries.
>
> Where possible avoid unnecessary environmental differences. If viable, use self-renewing certificates.
>
> **Level 2** - also package the operating system. This only became possible with Container technology.
>
> Further, out of the box, a 0FS is per-configured with,
>
> - Log Management (logs should rollover and compress)
> - Secured following best practices
> - Optimized trimming unnecessary fat for production use
> - Documented
> - Handle moving between environments (Dev, IST, QAT, Production ect...) without making changes during deployment.
>
> **Level 3** - cloud compatible.
>
> With this model the level 2 package can register itself to route appropriately between promotion environments (IST, QAT and Production). Stateful data like logging is externalized. There are no unique identifiers so the package may be easily replicated.
>
> **Level 4** - apply this technique to your Development Environment,
>
> Even make your tools like Eclipse, Source Control Client, etc... Zero Footprint so Developers and Operators can quickly get up to speed.
>
> **Level 5** - CICD
>
> Level 3 packages always built from recipes. Any change, even a simple config change happens in the receipe where upon a new Level 3 instance is rebuilt from scratch using the recipe.

This Bonsai Framework focuses on applications. Consider layering on software packing using Ubuntu Snaps or Flatpak and further up the chain with Docker (application isolation) and LXC (container isolation).