

Core Packaging Assumptions Refresh

TIER Objective

While the TIER project has committed to initial software distributions in the form of independent Docker and Virtual Machine images, TIER is fundamentally focused on the automated build/test pipeline and tooling that will enable the creation of multiple output formats over time. The selection of tooling for both current efficiency and longer-term staying power is important. See <https://spaces.at.internet2.edu/display/TPWG/Core+Packaging+Methodology> for thinking on TIER's build/test needs. The exact tooling discussed at the above URL above does not have to be a part of the final solution as long as their functional equivalent is provided and reasons for different tooling explained.

Expected TIER Process

The following notes outline the anticipated TIER process

1. **Packaged Distributions**
The initial packaged distributions include Docker images and pre-built Virtual Machines to run the containers for Shibboleth, COmanage, and Grouper. Shibboleth includes both the IdP and the Linux SP components (note that SP functionality is already required for COmanage and Grouper).
2. **Build Automation**
The build automation tooling and processes must be capable of multiple output formats and include provisions for automated testing. Containers should not be monolithic and should leverage the fact that TIER has worked to limit the number of different subcomponents such as types of databases, servlet engines, and Java distributions.
3. **TIER Software Component Source**
The TIER component teams will continue to build and package their software using their existing processes and tooling. Their current output will be the input to the packaging process described in this document. TIER Packaging may request changes to the component distributions, for example completely updated versions of Grouper instead of installer live patching, but the basic component structure will remain unchanged.
4. **Configuration**
The active configuration of the components should be largely independent of physical packaging. Containers should mount any needed configuration data via environment variables set by the users. Likewise, Virtual Machine images should provide for both easy mounting of configuration data and local storage of configuration, based on user preference. No changes to the Docker or Virtual Machine images should be needed to make any component configuration change.
5. **Logs**

TIER components often generate flat-file logs by default.¹ The packaging solution will provide a trivial mechanism to mount the log volumes into the containers. The Virtual Machine images will provide, at the user's discretion, functionality for either local or externally mounted storage.
6. **High Availability**
Many of the TIER components, especially Shibboleth, are typically operated in a high availability mode behind load balancing equipment. The packaging solutions must be designed to be compatible with high availability environments. TIER understands that high availability also imposes requirements on component features, databases, installation, and configuration and will work with the packaging firm on solutions.
7. **Monitoring**
The environments need simple processes to start, stop, and monitor the health of running components.

Notes

1) One mode of instrumentation will be via the ELK stack (Elasticsearch, Logstash, Kibana). The typical data source for ELK are log files. Can we agree on a strategy that supports this?