TIER Shibboleth IdP - Docker Reference Implementation

Draft

TIER Reference Implementations are designed to enable rapid evaluation of a component or set of interconnected components and provide a starting point for a full campus deployment. Reference Implementations are developed for Docker Swarm but dependencies are minimized within the containers in order to facilitate the use of other container orchestration mechanisms.

Introduction

Like the other TIER distributions for large-scale production environments, the TIER Shibboleth deployment is targeted for use with Docker Swarm. TIER will link to references on how to set up the Docker environment for this solution but will not provide pre-built virtual machines. Schools needing a vm-based solution should focus on the TIER Shibboleth Appliance instead. The use of Docker Swarm provides a mechanism to manage secrets, handle internal routing of requests, container orchestration, and facilitates hybrid campus-cloud deployments. Container orchestration frameworks other than Docker Swarm (e.g., Amazon ECS) may be evaluated in the future. The TIER Shibboleth Docker distribution itself is designed to support a variety of usage scenarios and has capabilities beyond those used here.

Requirements and Assumptions

1. The Shibboleth IdP must operate in high-availability mode, supporting multiple containers running on diverse hardware.
2. Local, cloud, and hybrid local/cloud deployments should be possible with the deployment.
3. As with the other TIER Shibboleth releases, Shibboleth is delivered to scale horizontally. No database or provisions for cross-node state are made.
4. Load Balancing
    a. External load balancing configuration is out of scope, but a high-level discussion and/or pointers to what a campus will need to do (e.g., sticky sessions) is in-scope.
    b. Note that Shibboleth requests can be sent to any node of a Swarm and the Swarm will direct the requests to an appropriate container.
5. Shibboleth keying material and other commonly changed configuration data are stored as Docker Swarm Secrets and made available to the Shibboleth containers as needed. The Swarm encrypts this data both in transit and at rest.
6. Assumption: school will provide docker host(s) configured for swarm mode.
7. Logging
    a. All logs will be sent to stdout using the TIER container standard format.

Components

1. TIER Shibboleth IdP Docker Container.
    a. Shibboleth IdP software
    b. Installation, after license approval, of the needed pre-requisite software of the Oracle Java 8 JRE and the Tomcat 8.0 application server.
    c. Used here in TIER Docker Shibboleth hybrid mode.
2. Docker Registry
    a. Docker does not automatically provide a container registry for the Swarm environment. TIER includes Docker's registry container for this function.
3. Swarm Mode Secrets
    a. Tooling/documentation on what to place in and how to update the Docker Swarm secrets.
4. Tooling
    a. A Configuration Builder tool that accepts user input and builds a full Shibboleth initial configuration based on TIER default settings.
    b. Scripting / documentation for existing configuration migration into the container environment, separating out secrets, etc.
5. Operational Documentation
    a. High level, focused on summarized and bootstrap information as opposed to a tutorial.
    b. Health status

Synopsis

The configuration present at the start of this larger-scale deployment methodology is that which is generated by the Shibboleth IdP installer. This IdP Installer configuration is burned into the container, including newly-generated certificates, private keys, and other associated material. TIER's configuration mechanism enables you to overlay these default secrets. The Configuration Builder assists with greenfield deployments while other scripts and documentation assist with the migration of existing campus configurations into the Docker Swarm environment.

Upgrades to new minor releases of the TIER Shibboleth Docker container are implemented using the same tooling described above. The campus is effectively maintaining a copy of its Shibboleth configuration tree and this configuration, coupled with the provided scripts and documentation, is used with updated versions of the container. Changes to the Shibboleth application that introduce new or significantly changed configuration files will be dealt with as needed.

Future Work

1. Performance Testing (perhaps some implementation guidance)