TIER Docker Container Specification

Statement

All Docker containers created or maintained by TIER are built to the specifications described in this document. We have tried to limit the requirements here to the minimum possible needed for compatibility while providing developers with as much flexibility as possible.

Background

In order to facilitate support, TIER has made decisions throughout the course of the project to standardize on certain sub-components and more recently to ensure that TIER containers are compatible with our default orchestration strategy of using Docker Swarm mode via Compose.

Specification

1. Base Linux Image
    a. All TIER developed containers must be based on the current Centos image. As of March, 2018, this is Centos 7.
        i. Source: standard maintained Centos 7 docker image
        ii. (Under Discussion) potential use of Centos 7 image from Dockerhub that includes what is needed to use systemd as init (instead of supervisord). We may enable this option if obtaining/implementing the logging changes we need to supervisord are hard - - https://hub.docker.com/r/centos/systemd/
        iii. When build pipelines are published for production, they must include a *yum update* step.
    b. TIER Ancillary Containers may be based on other Linux distributions
        i. An ancillary container is a non-TIER Component container (e.g., a database, message queue manager, etc.) that is used as part of a docker-compose to enable the TIER Component to function.
        ii. TIER Ancillary Containers may be based on other non-Centos Linux distributions as long as the container in question is published by the organization maintaining the product and and TIER-related changes are marginal.
2. Servlet Engine
    a. Tomcat will be used whenever a servlet engine is needed.
3. Java Distribution
    a. TIER will use Azul's curated Zulu Java in it's releases
    b. The ability to assist end users with the download and installation of Oracle's Java distribution will be preserved. Under these circumstances, users must accept the Oracle license and be prompted to do so via Internet2-approved language.
4. Database
    a. If a relational database is provided within a container, MARIADB will be used.
    b. In general, database support is normally handled externally by the user or via a TIER-maintained MARIADB container.
5. Multi-Process Container
    a. Supervisord will be used whenever a container needs more than one process.
6. TIER Beacon
    a. Run the TIER Beacon code on a regular interval as specified in the documentation. Unless the component has its own scheduling mechanism for running external code, this requirement will usually result in the need to support cron and run supervisord in the container.
7. Container Configuration
    a. Standard Data
        i. Containers may receive configuration data via the environment as described below for Secret Data (7.b)
        ii. Configuration data may be mounted into the container from external storage
        iii. Configuration data may be "burned" into the container while it is being built.
        iv. There are many trade-offs between ii and iii, some environments will choose to enable the end user to build their containers using either method.
    b. Secret Data
        i. The preferred mechanism to support data that must be protected (e.g., passwords, keys, etc.) is Swarm-mode Docker Secrets.
        ii. Docker secrets are read-only to the application.
        iii. Secret Processing - Docker Secrets are processed using one of the two mechanisms described below
            1. Secrets/Pointers-to-Secrets are passed in the environment using the syntax described below. Either a single value may be supplied or, with the _FILE suffix, a file pointing to a docker swarm secret location
                a. COMPONENT_DATABASE_PASSWORD=foobar
                b. COMPONENT_DATABASE_PASSWORD_FILE=/run/secrets/my_password_file
                c. Container startup scripts
                    i. Start-up scripts process the environment and do whatever setup is needed to make secrets usable in the application.
                    ii. If the environment contains both a _FILE and name-only variables, the _FILE form is to be used.
                d. Documentation/comments for each attribute is required.
            2. A naming convention is developed for all application files that will exist in /run/secrets. Scripting within the container appropriately processes these files, linking them to the application components as appropriate. Documentation /comments re: the naming convention and files is required.
8. Container Orchestration
    a. Containers designed for compatibility/ease of use with Docker SWARM mode using Docker Stack Deploy and Compose files.
    b. Work to not preclude the use of other orchestration frameworks.
    c. Secrets are automatically mounted in /run/secrets by docker stack deploy using a compose file.
9. Logging
    a. All logs from all elements within a container are written to stdout
    b. Goal: easily parsable records; future work is likely to include json formatted logs
    c. Lines (records) within each log file start with the following format
        a. Component Name (e.g., Shibboleth IdP, Grouper Loader, etc.)
        b. Native logfile name (e.g., Catalina.out, shibd.log, etc.)
        c. Environment (e.g., Prod, Dev, Test)
        d. A user supplied token via the environment
        v. The text of the logfile line, without modification.

    d. Records within a line are separated by the semi-colon character. Semicolons are not permitted in the first four fields and must be removed if present.
    e. Spaces also need to be removed from the (c) Environment and (d) User Supplied fields of each record. *If anyone remembers why we need ro remove these spaces, please comment here.*
    f. Example Records

        i. supervisord;console;testing;Build:1.2.3;2018-04-02 18:27:30,778 CRIT Set uid to user 0
        ii. tomcat;catalina.out;testing;Build:1.2.3;2018-04-02 18:27:32,915 [main] INFO org.apache.coyote.http11.Http11NioProtocol-Initializing ProtocolHandler ["https-jsse-nio-443"]
    g. Timestamps in logs must default to UTC. Documentation should exist to assist users with changing this default to a local timezone. The default of UTC instead of EST or PST seems logical given that many future campus deployments will include components deployed in multiple timezones for redundancy.
    h. Logfile Format Known Issues
        i. Supervisord - we are presently unable to change the supervisord logfile format, a requested change is in progress.
        ii. Grouper and Shibboleth IdP - Tomcat log format configuration work still needed.

10. Other Specifications
    a. ...
    b. ...
    c. ...