LITS Docker Proof-of-Concept

Background

Linux containers provide a light-weight, fast form of virtualization. While Linux containers have been around for over a decade, they were not widely deployed because of their complexity. This is changing with Docker. Docker is an open platform that offers a user-friendly way to build and deploy containers across multiple platforms (i.e. laptop, on-premises, and in the cloud). This functionality could be very advantageous for software developers. In this project, LITS will build on-premises and cloud-based Docker platforms to better understand system management, redundancy, networking, and security in a containerized infrastructure.

Once the platforms are built, the project will “Dockerize” two Emory applications to learn how Docker images are built, managed, and used in development.

This proof of concept project is a research and recommends initiative that centers on adoption, usage, and viability of Docker technologies within LITS and feasibility of establishing a Campus wide service.

Business Case and Objectives

The goal of this proof-of-concept is to increase LITS’s understanding of containers experientially by modeling two Emory applications using Docker. In addition to the knowledge gained by staff during the study, the project will also document findings for future use should Emory decide to deploy container technology on-premises or in the cloud.

Technology innovation drivers

- Standardize and reuse across teams
- Remove obstacles to distributed development
- Remove SDLC bottle necks
- Automate assembly process
- Remove manual hand offs
- Identify and resolve qualify issues faster
- Enable agile environment (development and operations)

Please see attached BusinessCase-Mini-DockerStudy.docx for additional details.

Iterations

1. Iteration #1: Demonstrate a simple, realistic Dockerized Web App
2. Iteration #2: Create useful base images, research logging
3. Iteration #3: Create useful base images continued
4. Iteration #4: Docker 1.9 upgrade, Swarm, and Secret Data

Resources

- LITS Guide to Creating Docker Images

Outstanding Operational Issues

Issue Description	Status	Notes
Containers running on dockerdmz1.cc.emory.edu are unable to reach resources in the admin core. Specifically, the WebEase web app running in a container from the emory-webease-webapp-2.0 container cannot send requests to the WebEase ESB service in the QA ESB cluster.	Closed	Firewall rule added 11/18 (REQ38573) Needs to be tested before we can close this issue. Steve W confirmed the web app successfully starts in the container now.
Unable to access the emory-webease-webapp-2.0 container on dockerpoc1.cc.emory.edu. I was able to do this successfully many time over the course of the week of November 9, but at the demo on Friday and now on Saturday I am unable to access the container after startup.	Closed	Steve tested and it is working now.
File system on dockerpoc1.cc.emory.edu reports full from time to time. I suggest we just add a ton of storage to this for now...like a TB or something until we decide how much of the build an package resources we are going to regularly preserve.	Closed	Mike Lewis add a 100G and will add more as needed.
Need to ensure that unpriviliged users can run build & package	Open	Steve Wheat has this one.

Based on our 11/13 discussion we should rename dockerpoc1 to be pocpackage or dockerpackage or something similar.	Closed	Mike Lewis created a CNAME to point dockerpackage at dockerpoc1.
In addition to dockerdmz1 we need a dockeradmin1 in the admin core.	Closed	Mike Lewis has this one.
Unable to access the container running the image emory/webease-webapp-2.0:latest on dockerdmz1.cc.emory.edu/8080/webease. Per the resolved item above, the app does start successfully in the container, but we just can't access the app...may be a firewall issue.	Closed	Access to these containers works if you are using 2FA when logging into the VPN. Steve tested and it is working now.
Log messages sent to Emory's central SYSLOG server by containers using the docker SYSLOG log driver do not contain the tags or container ID	Open	Paul has requested help from the Systems team and it is in their queue. The current theory is the SYSLOG-NG configuration needs to be altered for the hosts running the docker engine.