

[Link to 2022 ACAMP Wiki](#)

Advance CAMP Thu. Dec. 8, 2022

Room - 5

Session Title: IAM Testbed: What should it look like? / Containerized offline dev environments

CONVENER: Erik Scott / Bruce Timberlake

MAIN SCRIBE(S): Bruce Timberlake

ADDITIONAL CONTRIBUTORS:

of ATTENDEES: 18

DISCUSSION:

David Hutchins (dockerdude@virginia.edu), Kellen Murphy (wfx6yz@virginia.edu) @ UVA:

docker-swarm is the underlying technology (not just docker-compose)

- provide secrets
- overlay networking

stack:

- docker (same for dev -> prod except for environment variables etc)
- local machine for developers are the same as the prod service
- some work needed to mock the dev data to make it similar enough to production
- nothing sensitive in the environment itself at any time

Vagrant and virt box: totally local, no chance of messing up another developer. Vagrant sets up the VMs, installs docker swarm. Can set up components to point out over network to production or semi-production servers. Especially DB (?)

Vagrant used to ensure that the virtualbox images are the same, vs locally installed versions of the same apps since version differences could creep in over time and not keep each instance actually identical. Very repeatable, and identical.

Docker images stored in a private GitHub repository (used to be a private registry running on a server). GitHub repository is slow, so maybe consider a standalone registry or Docker Hub, etc.

Virtualbox directory sharing is very slow; using Unison instead.

docker stack ls

docker service ls

Traefik - ingress proxy to route all incoming traffic. Provides forwarding oauth with keycloak.

Service label aware, and indicate in configs which services should talk to other ones

Keycloak - OIDC auth/IdP (in Prod, this would be the Shib IdP e.g.) RedHat product.

Mailcatcher - simple SMTP interceptor

Kibana - Elasticsearch

Graylog - Log aggregation

Portainer.io - swarm management. Monitor container performance, etc. Provides a web-based console into the containers.

pgAdmin - web admin UI for PostgreSQL

T.I.C.K - Telegraf, InfluxDB, Chronograf, Kapacitor - performance metrics into time-series database

phpLdapAdmin -

Q: Why Docker Swarm vs Kubernetes?

A: Complexity, mainly. Institutional choice.

Docker swarm has built-in service discovery

dockerdude@virginia.edu - David :-)

Erik Q&A time:

UVA solution = "Small university in a box"

Federated credential management (not just SAML extended attributes). Being pushed from Europe. Will need a lot of development and testing. What to think about when building the testbed? What needs to be addressed?

Will need to work with commercial vendors and do some "bake off" decisions/competitions.
Federal employees vs university employees.
Would be able to avoid FedRAMP that way.

Can you add more SPs to the solution? Just need containers for the applications.

Why are people here in this session today?
comparing this meeting to InterOP from 1990s
No widely-agreed upon testbed these days

Is there a container suite that can be handed to a vendor and ask about Jenkins tests to qualify to bid?

Vagrant example with complete ITAP stack

With swarm, you could simulate a larger environment. Use Selenium/JMeter to simulate
Grouper Training Environment is a one-container all-in-one environment now (could be split into separate containers to "do it right")

Config and secrets are a complicating factor when setting up containers

Test data: all Postgres in the UVA demo

Scripts to export/import database after using the app, so that subsequent sessions can keep state

Grouper Training Environment is programatically built, so it's difficult to keep it up-to-date.
Generating synthetic data is difficult (Faker library)

ARTIFACTS / LINKS

Contact dockerdude@virginia.edu for access:

<https://github.com/UVA-IAM>

- compose-library
- iam-config (Vagrant configuration)
 - major branch for common configurations (all environments)
 - dev, test, qa, prod have their own separate branches and are used in CI/CD workflow configurations
- iam-vagrant (Vagrant and Virtualbox to provision the environment - Ubuntu VMs)
 - ansible subdir contains playbooks invoked by the Virtualbox/Vagrant build process. Starts the docker stack.

(Kellen will work to get some of this public soon!!!)