CONTAINERIZING BANNER WITH DOCKER

MARIANNE GILLFILLAN

SIG

Presenter

Marianne Gillfillan

Sr. DBA/Certified Cloud Architect

gillfillan@sigcorp.com

- 30+ years of Higher Ed and IT Systems
- 20+ years of Banner DBA
- Expertise: Ellucian Banner products, systems administration and support, Oracle Cloud migrations, AWS support, devops, networking, and disaster recovery

SIG

SIG Introduction

Industry Commitment
SIG was established in 1987 serving higher education with IT initiatives that enhance services for students, faculty, staff and alumni.

Ellucian® Partner
SIG is one of the largest privately held higher education consulting firms in the U.S., and we have been a long-standing partner with Ellucian for many years.

Consulting Continuity
Powered by more than 100 professionals, with an average tenure of 9.5 years, SIG can provide consulting continuity to keep your goals on task.

Agility to Respond
SIG provides a full lifecycle of services from strategy through managed services, and our ability to move quickly is what we do best. We can respond to uncertainty with flexibility as new opportunities arise.

More than
100
People

More than
35
Years Serving
Higher Education

Agenda

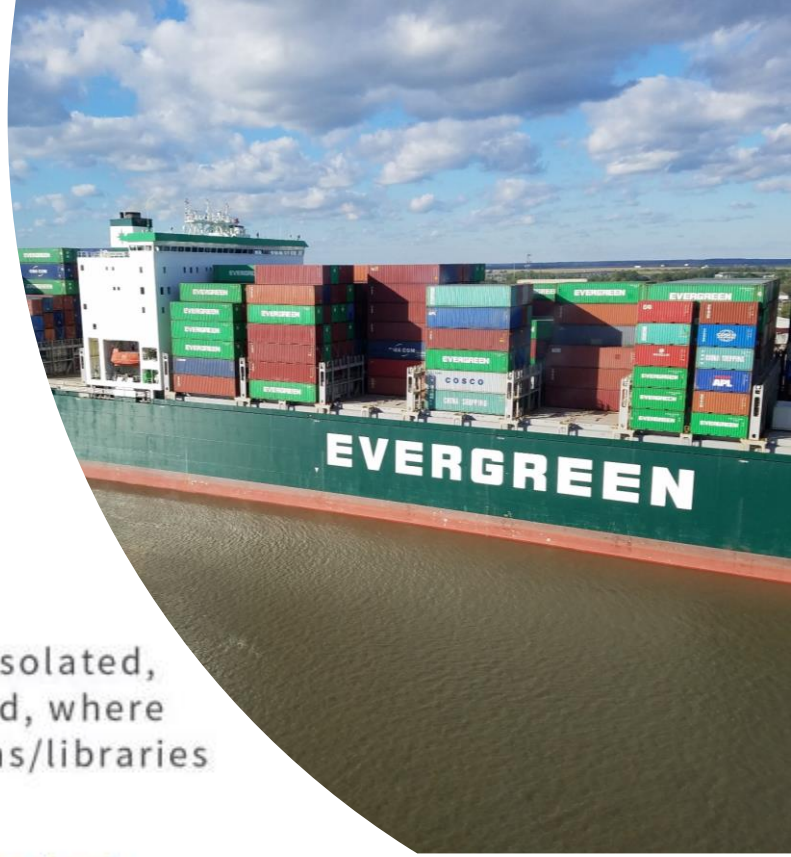
- Audience - Beginners
- Learn a bit about containers
- Verify Workstation preparation
- Create some images
- Lunch
- Finish creating images
- Create Banner application
- Pat yourself on the back and plan your next vacation

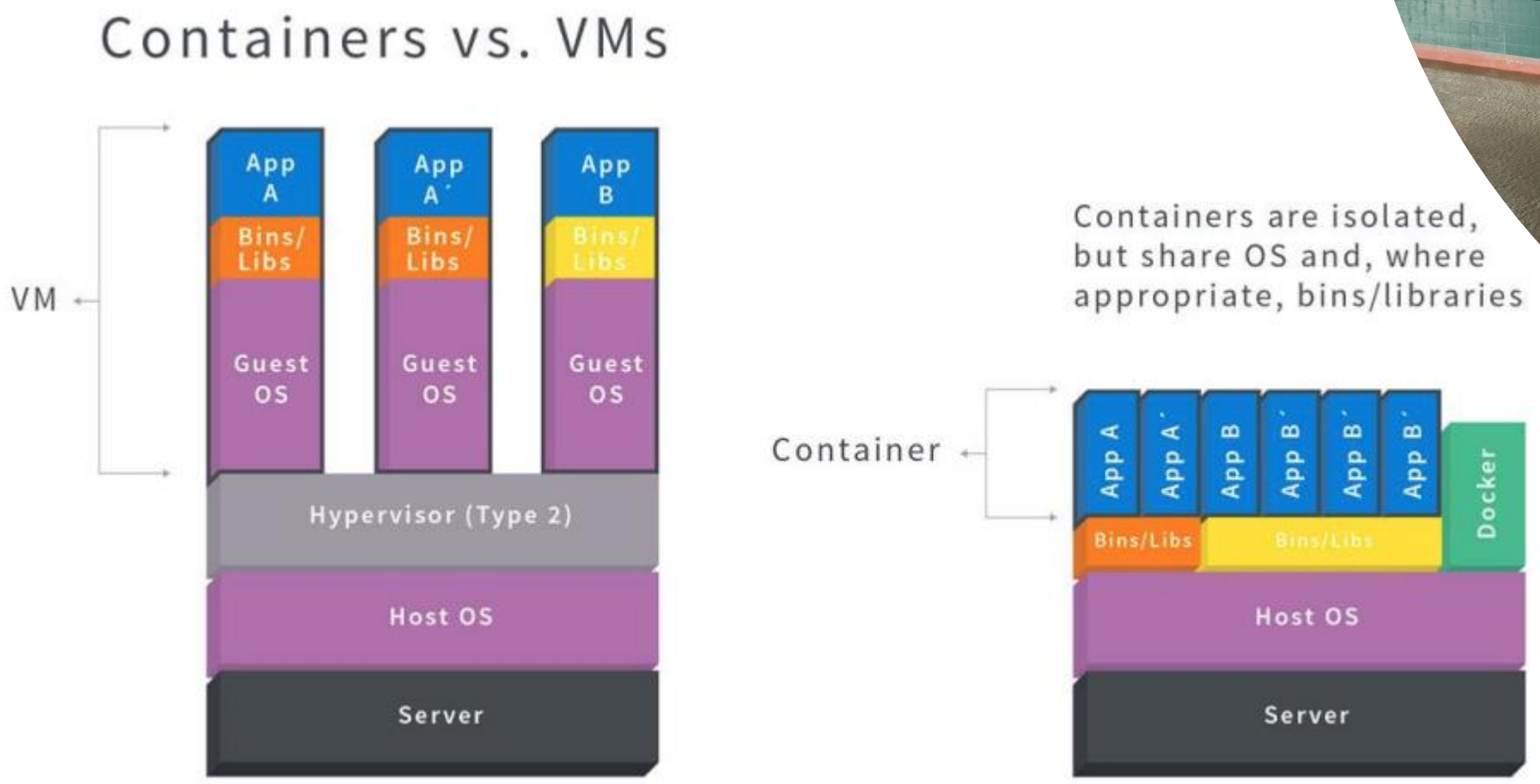
Terminology

- Image – an ordered collection of changes
- Container – a runtime instance of an image
- Layer – a modification to the image
- Dockerfile – a text file with commands to build image
- Registry Service – hosted service containing repos
  - Docker Hub, OCI Registry, AWS ECR
- Repository – collection of images with identifying tags
  - Local or remote

https://docs.docker.com/glossary/

What is a Container?





What is an Image?

- Read-only template
- Set of instructions
  - Creates layers
    - Layers can be shared between images – i.e. cached layers
- Used to build the container
- Immutable – can’t be changed once created
  - Destroy or create a new version (i.e. a tag)
  - Can have multiple tags
- Images themselves do not run
- Images can be built from other images

Basic Commands

- docker ps
- docker stats
- docker exec
- docker build
- docker run
- docker pull/push
- docker rm
- docker rmi
- docker images

Why use containers?

- Why NOT?
- Self-contained
- Replicate them easily
- Improve operational efficiency
- Better resource management (CPU, memory, etc)
- Codable, auditable environment
- Doesn’t matter the size of the institution

If It Isn't Codified, It Doesn't Exist

SIG

Codifying the Infrastructure

- Avoids configuration drift / undocumented changes
  - Self documenting
  - Mitigates issues with configuration differences
    - Application Health (performance differences, inconsistent behaviors, etc.)
    - Security
    - Regulatory Compliance
- Improves ability to make changes quickly
  - Centralized configurations and classifications (modify once in definition)
  - Reusable definitions (build common configurations)
- Repeatable
  - Disaster Recovery (rebuild fast)
  - Scalability (apply reusable definitions to multiple hosts)
- Auditable (change management & tracking, who-what-when)

How are containers codified?

- Dockerfile
  - Instructions to build the container
  - List of commands that typically run at the command line
- Common commands
  - ADD
  - COPY
  - ENV
  - EXPOSE
  - FROM

```
FROM tomcat:8.5-jdk8-openjdk

ENV APP_LOGS=/app_logs

RUN rm -Rf $CATALINA_HOME/webapps.dist

EXPOSE 8080
```

Adding Additional Scripting

- If there's a need to do more advanced scripting than what can be done in a Dockerfile, write it.
  - Bash
  - Ansible
  - Python
  - Whatever

In the Dockerfile

```
CMD /run.sh
```

In the run.sh

```
#
# Update server.xml
#
  if [[ -f "/tmp/resource_${varprefix}.sed" ]]; then
     rm /tmp/resource_${varprefix}.sed
  fi
  cat <<EOF > /tmp/resource_${varprefix}.sed
s|jdbcDatasource|${!jndivar}|
s|jdbcUrl|${!urlvar}|
s|datasourceUser|${!uservar}|
s|datasourcePswd|${!pswdvar}|
EOF
```

Let’s Get Started!

SIG

Workshop Pre-Reqs

- Tools that will be used during the workshop
  - Windows Subsystem for Linux 2 (WSL2)
    - Ubuntu Linux
  - Docker Desktop for Windows
    - https://docs.docker.com/desktop/install/windows-install/
    - Mac users should be able to just install Docker Desktop
  - Visual Studio Code (VS Code)
    - https://code.visualstudio.com/download
  - Docker Hub
    - https://hub.docker.com/
  - Github
    - https://github.com/
- Ready?

Download Workshop files

- Open VS Code
  - Start a Terminal session
  - Create a directory for this workshop somewhere on your machine and then go to it (i.e. mkdir, cd)
- Clone repository
  - https://github.com/gillfimj/coHEsion2022
  - git clone git@github.com:gillfimj/coHEsion2022.git
- What's in the repo?
  - Markdown files
  - PDFs
  - Possibly a few extras at the end of the workshop
    - git pull


QUESTIONS

Contact

Marianne Gillfillan

Sr. DBA/Certified Cloud Architect

gillfillan@sigcorp.com

/strata-information-group

@SIGCorpLIVE

Sigcorp.com


THANK
YOU