Comparing and contrasting Docker images from Ortus, Adobe, and Lucee

Charlie Arehart
@carehart
charlie@carehart.org

Updated: 9/1/2022
Who is this talk for?

- Speaking to those using/considering using containers, whether for ACF or Lucee
- Many here may know of Ortus Commandbox images for CF and Lucee
  - But both Adobe and Lucee offer their own container images (or “Docker images”)
- We’ll assess differences among them: what they do, how they work/are configured
  - Capabilities vary pretty widely
  - Commandbox images are more capable than the others, but those may still surprise some
- May help you consider alternatives, or at least help you help others do that
Other housekeeping

• Talk also presumes you already understand Docker
  • Including why you might want to use it, when it makes sense
  • You may be just exploring it, using it for CI/CD, running in prod or not
• Even if totally new, should still get considerable value from the talk
  • Indeed, may help get you started far faster than just on your own
• Presentation slides available at carehart.org/presentations
• Thank you to the ITB committee/Ortus for having me back again!
Me.about()

• Independent consultant
• Long-time community contributor
• Not too active in Ortus community, as I don’t do development professionally, so don’t use most of the wonderful tools
• But I do help people on whatever platform they’re on (CF, Lucee, Commandbox, etc.)

• …a couple more housekeeping items before we get rolling…
Focus is NOT on custom-building images

- Speaking here on the pre-built images from these vendors
- Some may prefer to build their own images from scratch -- not covered in this talk
  - See recent blog posts from Mark Drew, markdrew.io/slimmer-lucee-docker-images
  - Github project from Igal Sapir: github.com/isapir/lucee-docker
  - Beware you may well find on Dockerhub images named “coldfusion” or “lucee” that are NOT from vendors but from people publishing their own (perhaps from scratch)
- Choice to use vendor image vs your own is similar to frameworks debate 😊
On Compose, Dockerfiles, Kubernetes

• Finally, demos will be using images via compose
  • Of course can use Dockerfile’s if desired/when needed
  • Apply what I show however you may deploy the images

• Indeed, this is not to deny value of awesome kubernetes
  • Of course can deploy these images via k8s, via any implementation
  • Those familiar with k8s may often convert compose specs to pod(s)
    • Tool to help do this at kompose.io (free and cross-platform)

• …now we can move on to the real focus of the talk …
Finding these Docker images

• All are available via DockerHub
  • hub.docker.com/u/adobecoldfusion
  • hub.docker.com/u/lucee
  • hub.docker.com/r/ortussolutions/commandbox/

• Adobe also makes their images available via Amazon ECR:
  • gallery.ecr.aws/adobe
  • Offered only via jFrog bintray May 2018-Sep 2021…odd choice, past is past

• Let’s take a quick look at those image repositories
About differing images and tags

- Adobe ColdFusion
  - Offers CF, as well as:
    - CF Addons (Solr, PDFg/cfhtmltopdf support)
    - PMT (monitor)
    - API Manager
    - API Manager addons
  - Repository for each of them
  - Uses tags for each update

- Lucee
  - Offers Lucee, one repository
  - Other repos are very old
  - Uses tags for different engine versions, and
    - Snapshots, RC
    - Major JVM versions
    - Major Tomcat versions
    - Lucee Light, Lucee+nginx
    - Optional alpine

- Ortus CommandBox
  - Offers CF and Lucee, one repo
  - Uses tags for the engines and:
    - Major CF versions
    - Lucee versions (including snapshots)
    - Major JVM versions
    - Optional alpine
Docs for each

• Docs include discussion of image tag formats, environment vars, more
• Adobe CF Docker image help
  • helpx.adobe.com/coldfusion/using/docker-images-coldfusion.html
  • And some on the dockerhub page
• Lucee Docker image help
  • Primarily the dockerhub page
  • Also github.com/lucee/lucee-dockerfiles
• Commandbox image help
  • commandbox.ortusbooks.com/deploying-commandbox/docker
  • And some on the dockerhub page
CF licensing

• Elephant in the room…

• Deploying ACF images for production will indeed require licensing
  • Same is true of prod deploying ACF via Commandbox images (which can’t support CF Standard, due to underlying WAR implementation)

• Adobe CF is free for development
  • Many folks use containers entirely for development/exploration, rather than production deployment

• Adobe states container licensing as being that:
  • For CF Enterprise, can deploy 8 containers
  • For CF Standard, each container must be licensed
  • More: coldfusion.adobe.com/2019/03/coldfusion-licensing-docker-containers
Download counts...fwiw

Adobe ColdFusion
- 500k, across Dockerhub, ECR
  - Over past year
  - 3 CF engine "repositories"
  - With a few tags each

Lucee
- 1500k + on Dockerhub
  - over 5+ years
  - And 9 "repositories"
  - That main lucee/lucee image has had 7000+ tags
  - As for lucee52-nginx image
    - Last updated 4 years ago
    - Has 1000k+ of those
    - 1500k+ total
  - Beware lucee/lucee5
    - also not updated in 5 yrs

Ortus CommandBox
- 1000k+ on Dockerhub
  - Over 5 years
  - Across both CFML engines
  - Just 1 “repository”, over 600 tags
Another “elephant”: CPU architecture support

### Adobe ColdFusion
- AMD (x86) 64-bit only
  - (for now)

### Lucee
- AMD (x86) 64-bit only
  - (for now)
- See discussion on building these lucee/lucee images as ARM via docker build from dockerfile:
  - [dev.lucee.org/t/lucee-docker-on-apple-silicon/9208/7](dev.lucee.org/t/lucee-docker-on-apple-silicon/9208/7)
  - [dev.lucee.org/t/lucee-docker-on-apple-silicon/9208/13](dev.lucee.org/t/lucee-docker-on-apple-silicon/9208/13)

### Ortus CommandBox
- AMD (x86) 64-bit
- ARM 64-bit
Side-note: all these images are Linux images

• This will be obvious/old info to those using Docker already

• To those new to Docker:
  • This does NOT mean you “can’t run the images in Windows”
  • Docker Desktop for Windows or Linux VMs could run these
    • WSL (Windows Subsystem for Linux) is yet another way to run Linux and Docker

• As for CFML running on Linux…
  • Beware case-sensitivity of file names
  • Otherwise most would never notice CF is running “on Linux”
    • Just as how many don’t notice/realize CF is running on Java
Demo time!

- Let’s run the most basic example of each of the engines
- I’ll show a compose file for each
  - Don’t worry if you’ve never seen a compose file or have never used Docker
  - We’ll come back and elaborate on some things
- Also, I will show them as run via VS Code and its Docker extension
  - But you could run them (and the images) from the command line
  - Or via Portainer or even Docker Desktop (as alternative mgt Uis)
  - Or again via Kubernetes manifests, which can also be managed from all 3
- ...

@carchart
charlie@carchart.org
Still much more to cover

- Image sizes, time awaiting first pull
- Environment variables
- Implementing CFML code
- Admin setting configuration
- Acessing CF or Lucee admin (or disabling)
- Setting CF or Lucee admin password
Beware of first pull download time

- If you were to run these yourself for first time, images must be downloaded
  - Time will depend on image size (more in a moment) and your bandwidth

- Docker (and Compose and Kubernetes) all automatically pull image when used for first time, caching it for later reuse
  - Can also use docker pull (or docker compose pull) to force a pull

- Compose offers commands/args/pull_policy regarding forced pulls
  - baeldung.com/ops/docker-compose-latest-image

- Note: Kubernetes pulls images on each pod deployment if:
  - You don’t specify an imagepullpolicy and
  - You use “latest” tag or no tag at all
  - kubernetes.io/docs/concepts/containers/images/#imagepullpolicy-defaulting
# Image sizes (as shown at Dockerhub)

<table>
<thead>
<tr>
<th>Adobe ColdFusion</th>
<th>Lucee</th>
<th>Ortus CommandBox</th>
</tr>
</thead>
<tbody>
<tr>
<td>For CF2018 and 2016</td>
<td>About 270-600mb, depending on:</td>
<td>For CF2018 and 2016</td>
</tr>
<tr>
<td>About 600mb</td>
<td>Light or not</td>
<td>For CF2021</td>
</tr>
<tr>
<td>For CF2021</td>
<td>Nginx or not</td>
<td>For Lucee</td>
</tr>
<tr>
<td>About 200mb</td>
<td>Alpine or not</td>
<td>About 450-750mb, depending on:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Alpine or not</td>
</tr>
</tbody>
</table>
Environment variables

• Most Docker images offer configurability via “environment vars”
  • Each container is free to implement them however they want
  • Be aware they are case-sensitive (name and expected values)
• In the case of these CF engine images, there’s quite a disparity…
Built-in environment variables in CF images

Adobe ColdFusion

- acceptEULA=YES (required)
- password (Admin)
- serial/previousSerial (if used)
- installModules (list or all)
- importModules (file listing them)
- importCFSettings (json file of settings, from cfsetup)
- importCFSettingsPassphrase (if set in cfsetup)
- setupScript (to name script to run at startup, such as to call adminapi methods)
- And many more (see online help earlier)
Built-in environment variables in CF images

- LUCEE_JAVA_OPTS
Built-in environment variables in CF images

- `APP_DIR`
- `USER`
- `cfconfig_[engine setting]`
- `BOX_SERVER_APP_SERVERHOMEDIRECTORY`
- `BOX_SERVER_CFCONFIGFILE`
- `BOX_SERVER_APP_CFENGINE`
- `BOX_SERVER_PROFILE`
- `BOX_SERVER_WEB_REWRITES_ENABLE`
- `BOX_INSTALL / box_install`
- `CFPM_INSTALL` and `CFPM_UNINSTALL`
- Also supports `<<SECRET:*>>` for secret values, or `_FILE` suffix
Implementing or pointing to CFML code

• Each CFML engine image offers a default location (within the container as deployed)

• You can copy code into an image using Dockerfiles (or docker cp)

• You also can use bind mounts or volumes, whether pointed to in:
  • Compose file
  • Dockerfile
  • Command line (Docker Run)

• This talk can’t elaborate on those options

• ...

@carehart
charlie@carehart.org
Default app folder

Adobe ColdFusion

- /app
- Files typical of cf wwwroot also there, including folders:
  - CFIDE
  - cf_scripts
  - restplay

Lucee

- /var/www
- If nothing mounted there, 4 default files are there:
  - index.cfm
  - Application.cfc
  - debug.cfm
  - favicon.ico

Ortus CommandBox

- /app
- If nothing mounted there, 4 default files are there:
  - index.cfm
  - commandBoxLogo300.png
  - 403.html
Admin setting configuration

• Naturally, your code may not work if you haven’t configured things like datasources, mappings, etc.
• All 3 vendor images offer different ways to facilitate that…
Adobe ColdFusion

- In CF2021:
  - `importCFSettings` env var
- In CF2021, 2018, 2016:
  - Car file mounted into `/data` folder in container
  - Or AdminAPI, callable in cfml named in `setupScript` env var

Lucee

- Can copy lucee config files into:
  - `/opt/lucee/web/
  - `/opt/lucee/server/lucee-server/context`

Ortus CommandBox

- `cfconfig import`, via env vars:
  - `BOX_SERVER_CFCON`
  - `FIGFILE`
  - `cfconfig_[engine setting]`
Default port for web apps (and admin, if any)

<table>
<thead>
<tr>
<th>Adobe ColdFusion</th>
<th>Lucee</th>
<th>Ortus CommandBox</th>
</tr>
</thead>
<tbody>
<tr>
<td>8500</td>
<td>8888</td>
<td>8080 and 8443</td>
</tr>
</tbody>
</table>

charlie@carehart.org
@carehart
Acessing CF or Lucee admin (if enabled)

Adobe ColdFusion

- /CFIDE/administrator/index.cfm

Lucee

- /lucee/admin/server.cfm
- /lucee/admin/web.cfm

Ortus CommandBox

- See links on left
Setting CF or Lucee admin password

Adobe ColdFusion

- password env var

Lucee

- Place a password.txt file in container at /opt/lucee/server/lucee-server/context
- Can do with:
  - Dockerfile COPY
  - Compose bind mount
  - docker cp

Ortus CommandBox

- adminPassword property in cfconfig
Many more things we could discuss

- Enabling/disabling admin access
- Limiting admin access by IP
- Enabling/disabling browsing of directories
- Modifying JVM args within the containers
- Enabling FR within the containers
- Benchmarking performance differences
- Showing integration with web servers in front of CFML engine
- Showing integration with database servers (running in other containers or not)
- And still more
- Look for a part 2 or blog posts to come
Bonus tip: searching/listing the many tags for Lucee and CommandBox

• Dockerhub ui offers search (filter)
  • Useful if looking FOR something
  • But can’t seem to negate, not useful for finding anything BUT something

• While there is a docker search, it does not support searching tags

• There is an http api, which can list all tags
  • Can format with jq (if installed) and limit results with grep or awk, for instance

• For instance, how to filter out the 6,800+ lucee/lucee image tags that say SNAPSHOT, -RC, -beta or -BETA?
  • wget -q https://registry.hub.docker.com/v1/repositories/lucee/lucee/tags -O - | jq -r '.[].name' | grep -Ev 'SNAPSHOT|-RC|-beta|-BETA'
Some closing thoughts

• Getting my compose files
  • `github.com/carehart/awesome-cf-compose`
  • See also `github.com/docker/awesome-compose`, which inspired it
    • Briefly: why I haven’t yet put my contributions there instead

• Where can you get help on these things
  • No particular place focuses on any of the engine docker images
  • Normal places for support for specific engines, or wider cf community
    • See `cf411.com/cfcommhelp`
  • And of course I can help directly, free to a point, or via consulting
Summary

- We’ve seen there are indeed options for running CFML images
  - Those from each vendor have their pros and cons
  - Folks can decide if one or another suits them better
- The images from each will surely evolve over time
- Again, see links I’ve shared for the images, help in using them, my compose files, and getting help beyond this session
- With that, enjoy the rest of this wonderful conference!