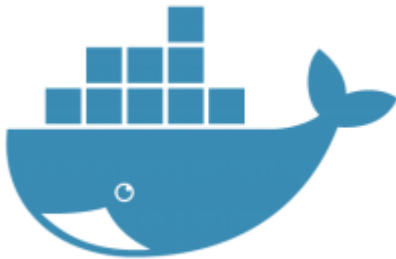




BeeCon 2016



Alfresco Under Docker



[Philippe Dubois]

Mikel Asla

Developer @keensoft



Topics



- Docker overview
- “Dockerized” Alfresco
- Our approach
- Future lines of action



Docker overview



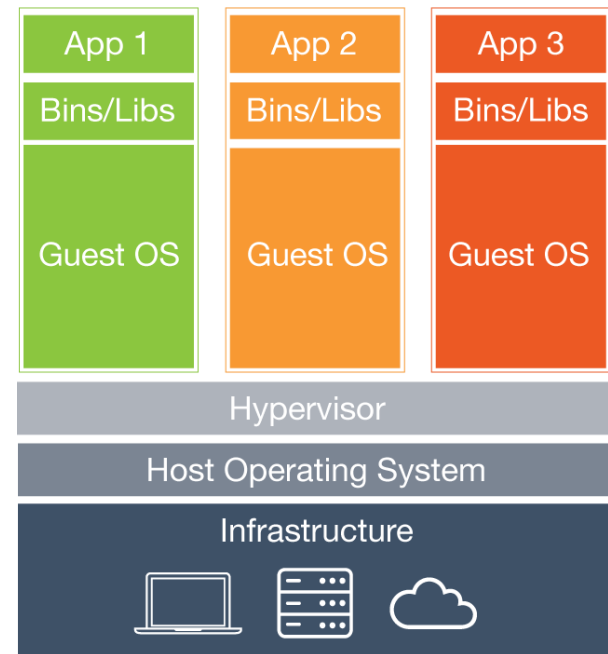
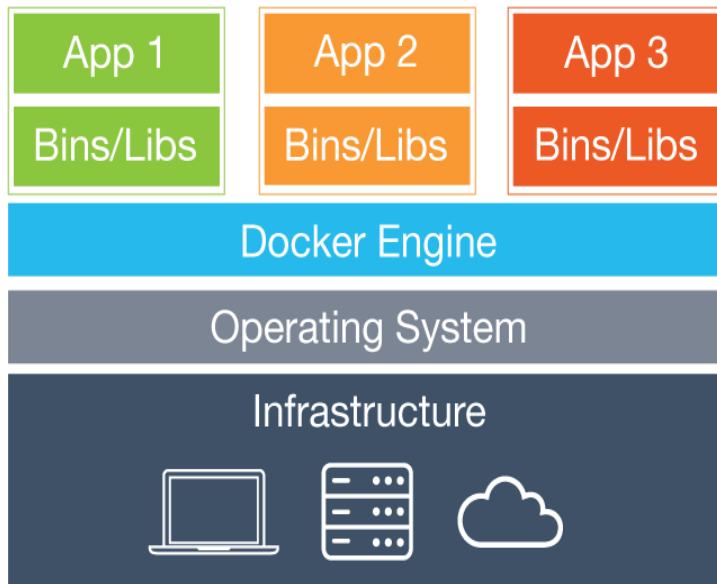
- “Docker allows you to package an application with all of its dependencies into a standardized unit for software development.”
- Declare your app, run and ship anywhere



Docker overview



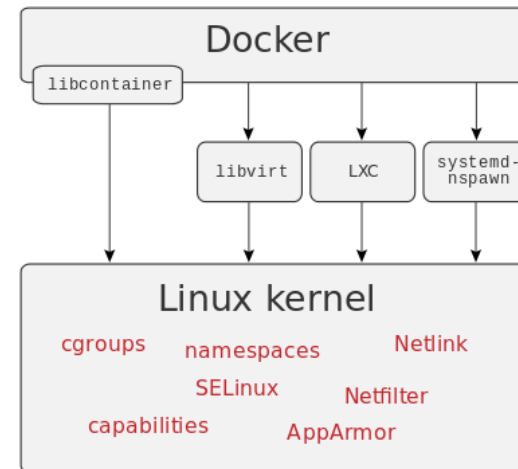
- Containers are not Virtual Machines



Docker overview

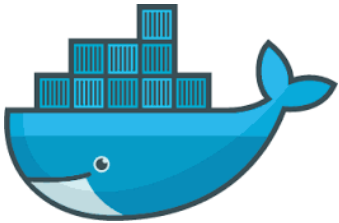


- Docker Host
 - Docker Engine
 - Docker Client
- Build Docker Images
- Run Docker containers based on an Image and any arbitrary commands you'd like on top
- Publish Docker Images (Public or Private)

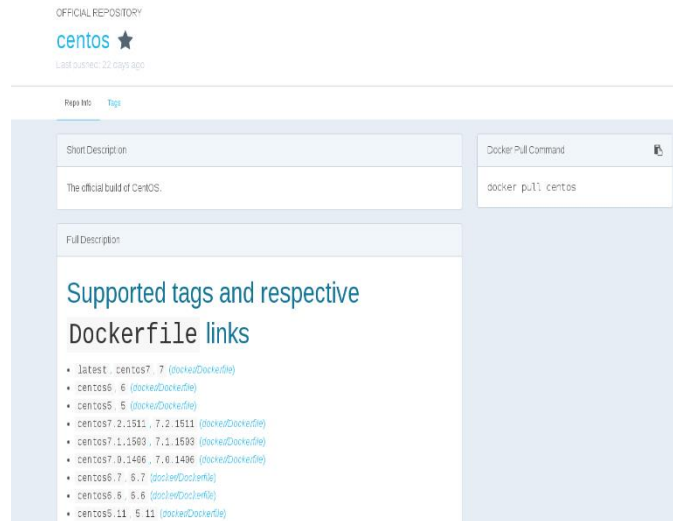


Docker overview

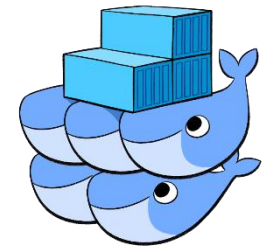
Docker Host



Docker Hub



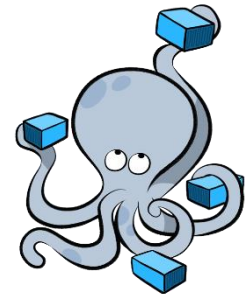
Docker Swarm



Docker Machine



Docker Compose

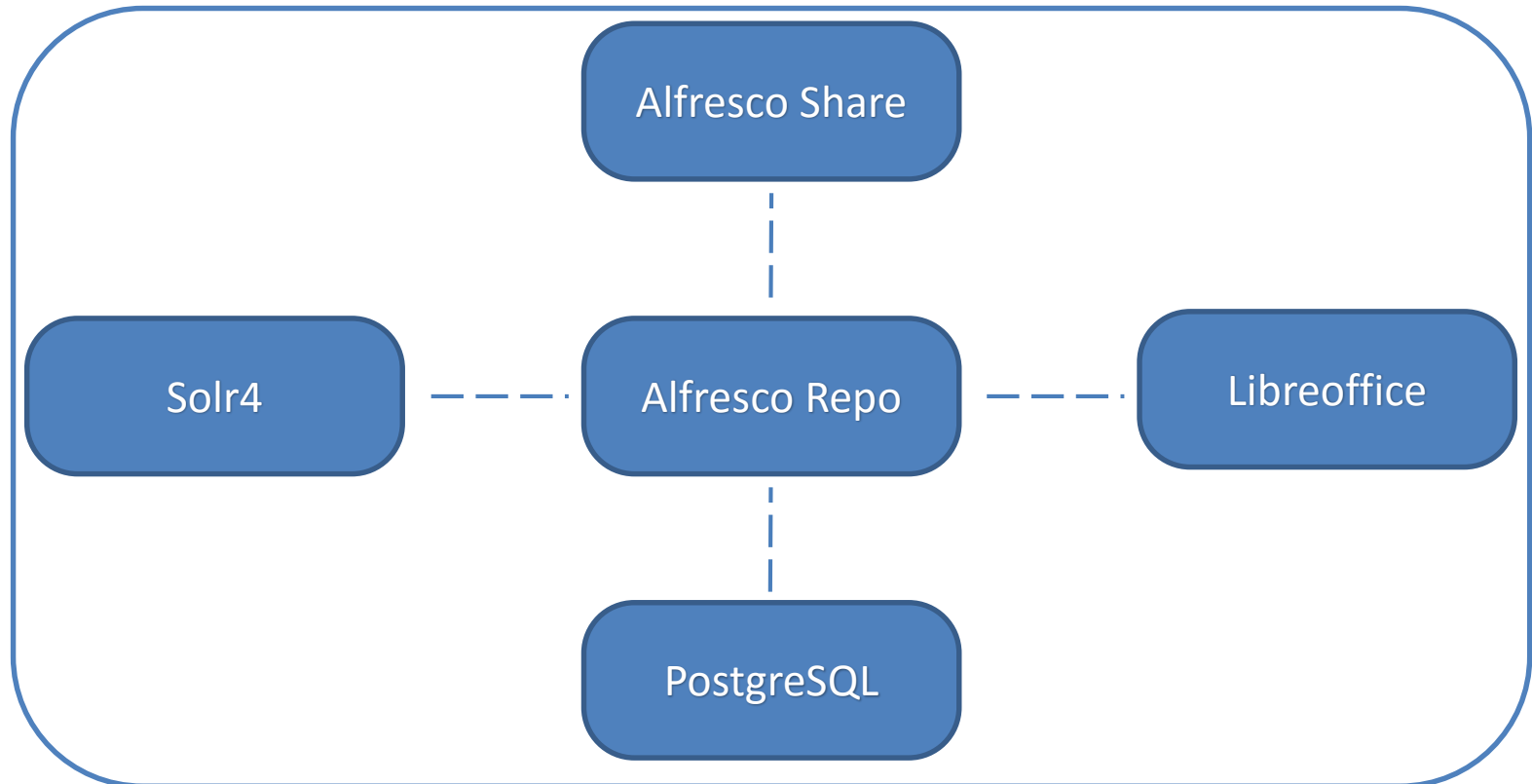


Dockerfile



BeeCon 2016

Dockerized Alfresco



Docker Host



BeeCon 2016

Dockerized Alfresco



Order of The Bee distro is experimenting with Docker

<https://github.com/marsbard/docker-alfresco>

Martin Cosgrave has built an specific tool to orchestrate alfresco components in a very flexible way.

<https://github.com/marsbard/grua>



BeeCon 2016

Dockerized Alfresco



Alfresco is testing Docker

- https://github.com/Alfresco/alf_docker
- <https://www.alfresco.com/blogs/developer/2016/03/23/testing-alfresco-one-with-docker>



BeeCon 2016

Dockerized Alfresco



Some popular Docker Hub repositories

- <https://hub.docker.com/r/pdubois/docker-alfresco>
- <https://hub.docker.com/r/gui81/alfresco/>



BeeCon 2016

Our approach



- Docker Images as “client stack templates”
- Development day to day continues the same
- Glue scripting
- Periodically, Test committed code in a “client” container
- Run “client” environments quickly to resolve issues



Our approach



- First step is to create basic templates based on Alfresco Supported Stack and your clients requirements



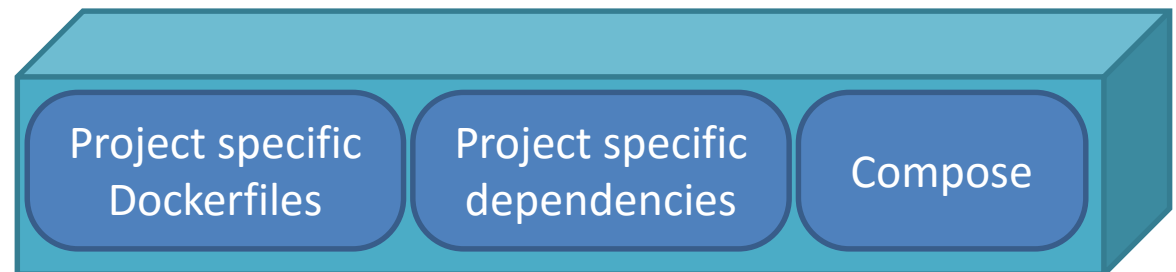
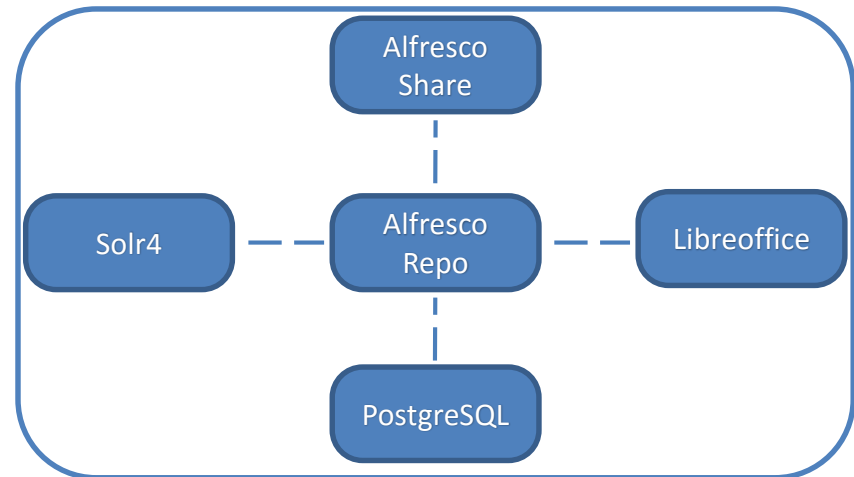
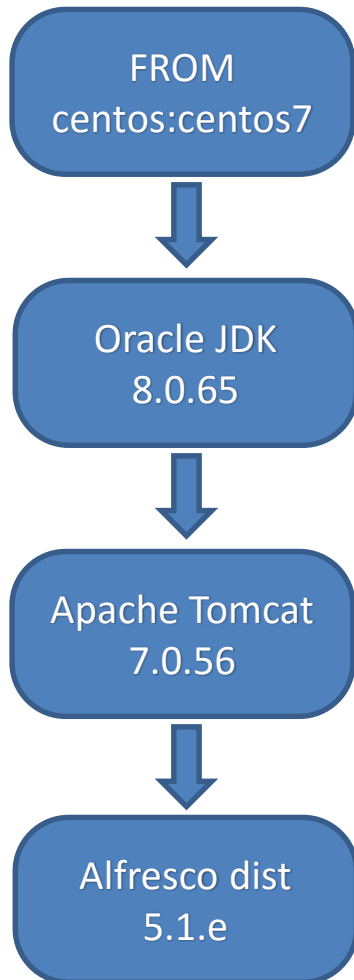
Our approach



- Next step is to define project level image configurations (Auth, Subsystems, ...)
- The simplest is to follow the client installation manual you did and just reproduce it
- You may need other components that cannot be addressed with Docker (Active Directory, ...)



Our approach



Our approach



You can find here our work in this idea

- <https://hub.docker.com/r/keensoft/alfresco-docker-template/>
- <https://github.com/keensoft>



Our approach



Sample containers for testing components

- <https://github.com/angelborroy-ks/docker-openldap>
- <https://hub.docker.com/r/keensoft/libreoffice/>
- <https://hub.docker.com/r/alexeiled/docker-oracle-xe-11g/>
- https://hub.docker.com/_/postgres/



Our approach



Benefits at this point

- Code is tested in client stack based containers
- Non technical members just love it
- Support team can resolve issues even without being part of the development process
- Engage time to project is very low.
- Standard deploy using `apply_amps.sh`



Our approach



How we create basis for the Community

- <https://hub.docker.com/r/mikelasla/alfresco-standalone/>
- <https://github.com/enzor/alfresco>
- <https://github.com/wabson/alfresco-docker-cors>

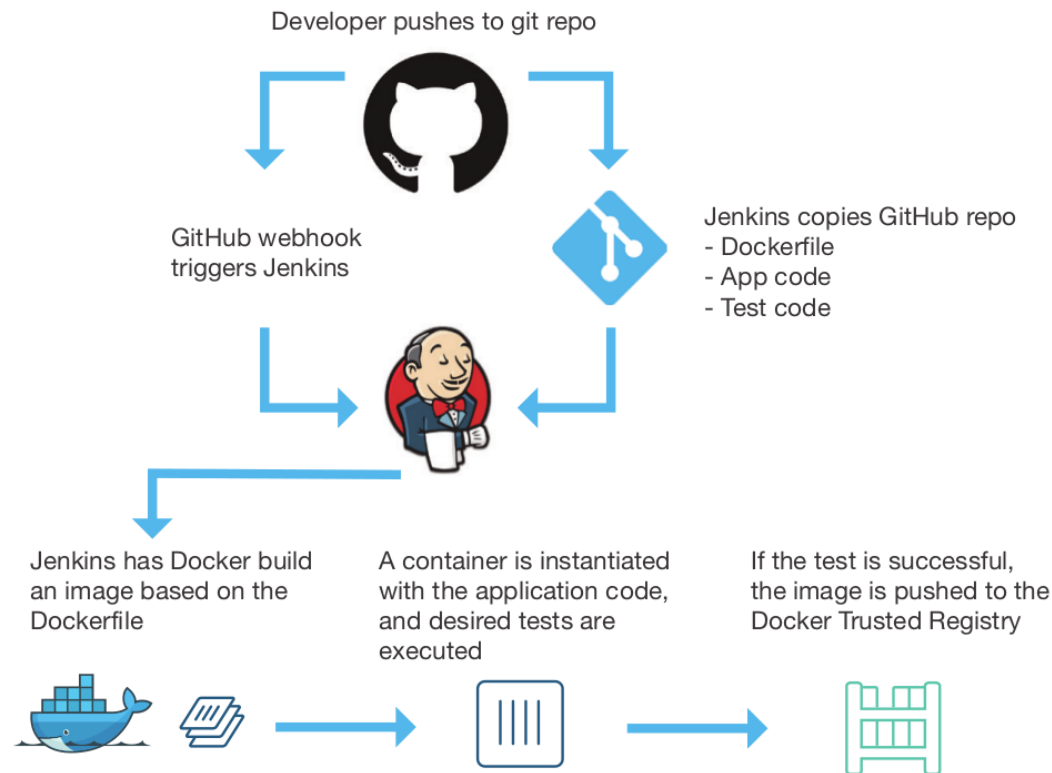


BeeCon 2016

Future lines of action



- Testing
- CI / CD



Shoot!



BeeCon 2016