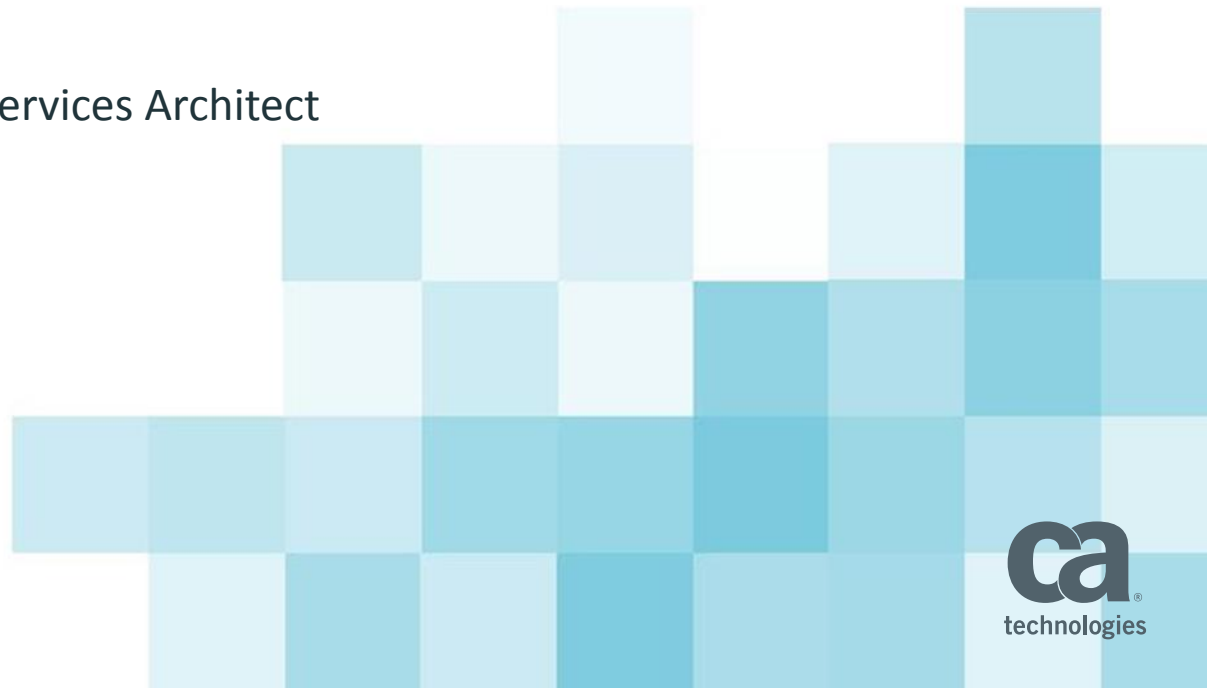


CA Release Automation Community Webcast Series

Microservices, Docker and CA Release Automation

Keith Puzey
Senior Principal Engineering Services Architect

28th September 2016



What We'll Cover in Today's Webcast

- How to leverage containers for microservices.
- In addition, you'll see a demo of how CA Release Automation's integration with Docker enables:
 - Dynamic creation and management of container environments
 - Dynamic provisioning of testing nodes and management
 - Rapid creation of environments for testing and development

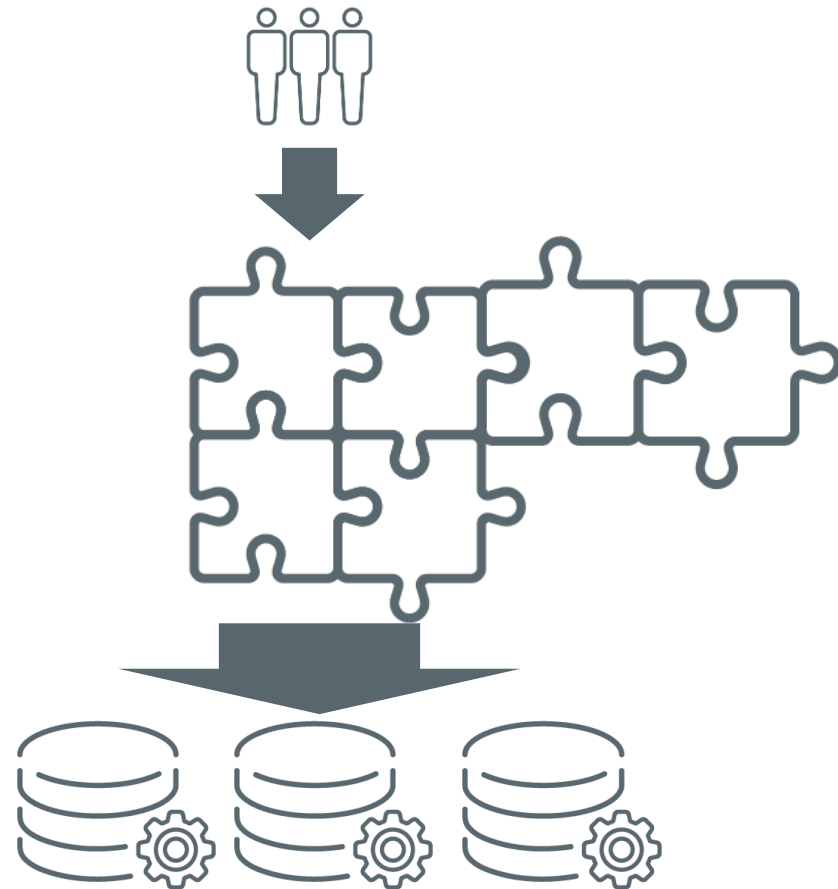
Why is This Important?

- Containers are an ideal technology for the creation and deployment of microservices
- Containers are a light-weight runtime environments
 - Contains many of the core components of a virtual machine
 - Isolated services of an operating system
- Containers can be used to package an application and allows it to run on-premises in a private cloud and is portable so can also be run on the public cloud

Microservice Application

What is a MicroService

- A suite of small services each running its own process
- Loosely Coupled
- Bounded context
- Microservices inter connected generally over HTTP API

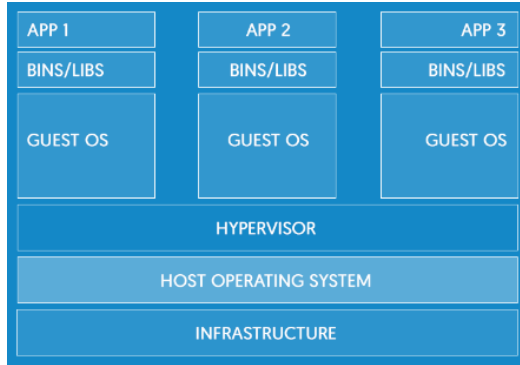


What is docker

- Docker is an open platform for developers and sysadmins to build, ship, and run distributed applications and consists of:
 - Docker Engine, a portable, lightweight runtime and packaging tool.
 - Docker Hub, a cloud service for sharing applications and automating workflows.
 - Docker enables apps to be quickly assembled from components and eliminates the friction between development, QA, and production environments.
- IT can ship faster and run the same app, unchanged, on laptops, data center VMs, and any cloud.

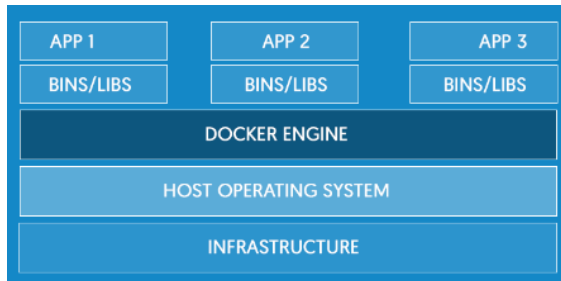


Docker and Virtual Machines



Virtual Machines

Each virtualized application includes not only the application - which may be only 10s of MB - and the necessary binaries and libraries, but also an entire guest operating system - which may weigh 10s of GB.



Docker

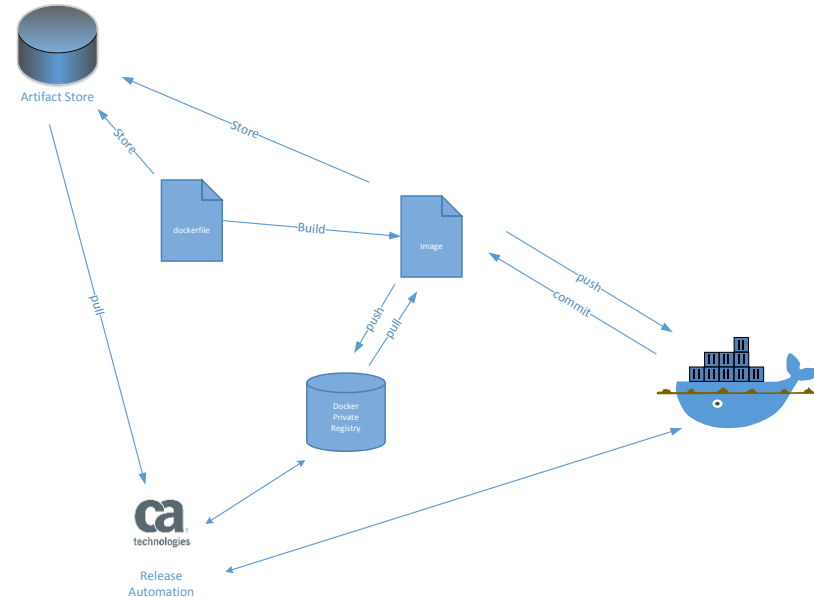
The Docker Engine container comprises just the application and its dependencies. It runs as an isolated process in userspace on the host operating system, sharing the kernel with other containers. Thus, it enjoys the resource isolation and allocation benefits of VMs but is much more portable and efficient.

Examples of Containerization adoption

- **Continuous Delivery**
 - Dynamically provision environments for testing in development and testing environments
 - Use containers to dynamically create testing nodes as part of your Continuous Delivery infrastructure
- **Application infrastructure**
 - Containerize infrastructure components to be used in Development and QA environments
- **Application Components**
 - Containerization of micro-services within your applications

Storing Docker Artifacts

- Docker has its own public / private repositories
- Docker artifacts can also be stored in several ways using Repositories:
 - Dockerfile
 - stored as artifacts
 - Images / containers created on demand
 - Docker Images
 - Stored as artifacts
 - Containers created on demand



Igniting Innovation through an Integrated CD Ecosystem

R.A.P.I.D.

Continuous Delivery

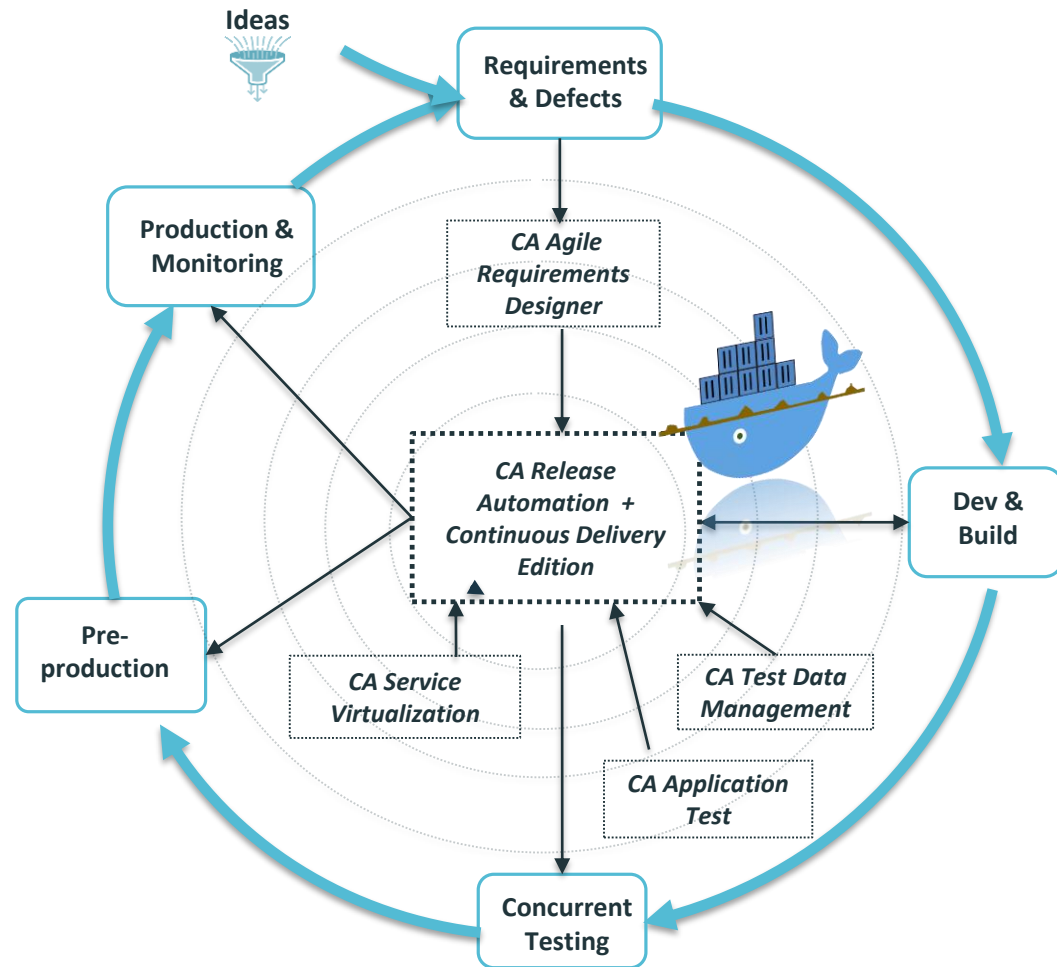
Release Planning & Orchestration

Agile Testing

Provision Data on Demand

Integrated Ecosystem & Feedback

Deploy Simulated Environments



CA Release Automation Action Pack for Docker

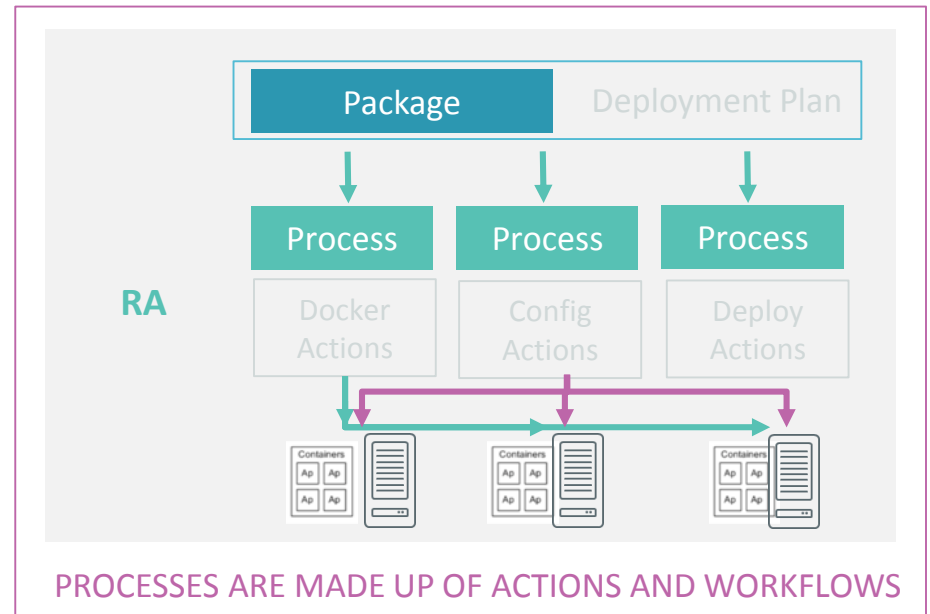


WHAT IT DOES

- Enables users to create workflow processes to provision and configure Docker containers and images on multiple environments directly within a deployment workflow
- Docker actions include operations such as create, export or inspect a container, create and inspect images, and search and obtain information about a containers or images

JOINT VALUE PROPOSITION

- Simplifies and speeds the release process by providing a seamless, integrated workflow to build and deploy application components to Docker Containers
- Helps reduce errors by providing valuable information about Docker environments prior to application deployments

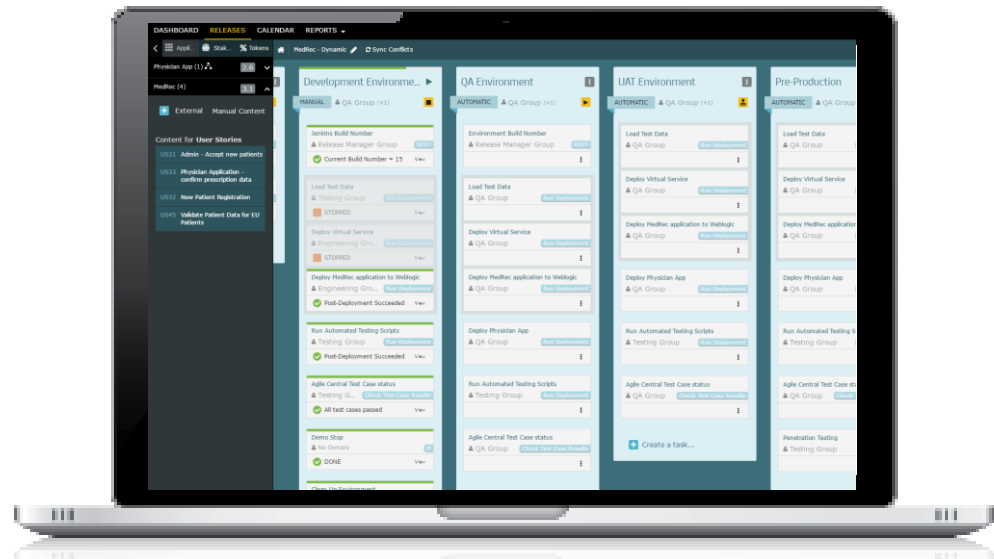


For more information see:

- [CA Release Automation Integration Hub](#)
- [Docker Action Pack video demonstration](#)
- [Docker Action Pack documentation](#)

Demonstrations

- Dynamic creation and management of container environments
- Dynamic provisioning of testing nodes and management
- Rapid creation of environments for testing and development



Session Summary

- How to leverage containers for microservices.
- Dynamic creation and management of container environments
- Dynamic provisioning of testing nodes and management
- Rapid creation of environments for testing and development

Questions?

Comments?



In Closing

Questions after the webcast?

- Post them on the community site
- Contact me directly at keith.Puzey@ca.com

Upcoming RA Community webcasts

- October, 5, 10 am EST, Creating a RA Shared Component



Keith Puzey

Senior Principal Engineering Services Architect

Keith.Puzey@ca.com



in