How to containerize at speed and scale with Docker EE, moving your existing applications to the Cloud

Patrick van der Bleek
Sr. Solutions Engineer NEMEA
Common Challenges Of A Legacy App

- Original app authors are no longer around
- No idea what the app is made of
- When was it last updated?
- Don’t change it! Don’t break it

Diagram:
- **Internal**
  - Linux
  - LAMP Stack
  - Java
- **External**
  - Windows
  - .NET
  - .NET IIS
Modernize Traditional Apps with Docker Enterprise Edition to get portability, security and efficiency of apps without changing the code.

You have to cut into the 80% To Fuel The Innovation
The bimodal IT myth
Docker Enterprise Edition

The only Containers-as-a-Service platform for IT that manages and secures diverse applications across disparate infrastructure, both on-premises and in the cloud.
Docker EE Gives Legacy Applications Modern Capabilities without any recoding or refactoring of the app.

**Efficient**
Optimize CapEx and OpEx costs

**Portable**
Infrastructure Independent Apps

**Secure**
Reduce risk and enforce new controls

Size of Infrastructure
- 50% Reduction

Deployment Speed
- up to 90% Faster

MTTR for Patching
- up to 90% Faster
Docker EE saves time and money

Reduce Total IT Costs by 50%
- Consolidate infrastructure
- Reduce software costs
- Gain operational efficiency

Efficient
Optimize CapEx and OpEx costs
Eliminate the outdated app runbook for a simple Dockerfile

Before

- VMs contain a full OS instance within each VM
- Containers share the kernel of a single OS instance on the physical or virtual server
- Average infrastructure consolidation is 50%

After

- VMs contain a full OS instance within each VM
- Containers share the kernel of a single OS instance on the physical or virtual server
- Average infrastructure consolidation is 50%
Streamline configuration management

**Before**

100 Page Binder

- Replace the printed (often out of date) runbooks for app deployment and ops documentation
- Dockerfile contains all commands to assemble a Docker container
- Define instructions including: ports, volumes, environment variables, healthchecks and more

**After**

Single Text File

- Dockerfile containing all the instructions to deploy your app.
- Enables consistent deployments across multiple environments, and eliminates the problem of “snowflake infrastructure”
Eliminate the outdated app runbook for a simple Dockerfile

Simplify app configuration management

- define app configs in Dockerfile (single container) or Compose file (multi-container)

Eliminate configuration drift

- No more patching in place, deploy new
- New deployment = new container image and tag in registry
- `docker diff` command shows exactly what’s changed in the container compared to the dockerfile
Improve asset management

- Centrally manage all container images in a private registry
- Keep a record of all versions (tags) of images available for
Improve app operations: deployments, rollback with built in app reliability

- Copy and paste or single command to deploy apps and define state
- Rolling updates reduce the risk of new deployments
- Easy roll back to previous known container
- Built in health checks continually monitor containers
- Automatic rescheduling of containers in the event of a failure
Docker EE ensures hybrid cloud portability

Deploy any app anywhere

- Applications can move across multiple infrastructures
- Infrastructure agnostic properties
Containers abstract applications from infrastructure

- Eliminates the “works on my machine” problem
- Containers packages code and dependencies together into an isolated process
- Containers standardize any workload: legacy, microservices, ISV apps (Windows and Linux)
- App configurations “travel” with the app, are not built to the infrastructure
- Easy app composition of simple to complex apps with security, networks, storage, env variables, ports
Container architecture provides infrastructure agnostic packaging and tooling

Container architecture allows applications to be packaged in containers, which can run on any host operating system regardless of the underlying infrastructure. This is achieved through the use of container orchestration platforms like Docker EE, which manages the deployment and scaling of containers across various hosts.

Disparate IT Infrastructure

- **App A**
  - Bins/Lib
- **App B**
  - Bins/Lib
- **App C**
  - Bins/Lib
- **App D**
  - Bins/Lib
- **App E**
  - Bins/Lib

**Docker EE**

**Host OS**

- Linux
- Windows
- Mainframe
- AWS
- Azure
- Other Public Clouds

Disparate IT Infrastructure

The graphic illustrates how containerization allows applications to be deployed on various hosts, including Linux, Windows, Mainframe, AWS, Azure, and other public cloud environments, ensuring flexibility and compatibility across different infrastructures.
Get infrastructure flexibility and portability for legacy apps

Developer can work in whatever environment they're used to.

Application gets moved into Test/QE environment.

Application can then be promoted to production on any public, private, or hybrid infrastructure.
Docker EE enhances application security

Reduce risk profile
- More secure environment
- Reduce surface area
- Vulnerability management
“Gartner asserts that applications deployed in containers are more secure than applications deployed on the bare OS.”

http://blogs.gartner.com/joerg-fritsch/can-you-operationalize-docker-containers/
Reduce the attack surface area of legacy apps

1. Out of the box default settings and profiles
2. Granular controls to customize settings

- Reduce risk associated with older code and components
- Default out of the box settings provide greater security
- Configurable settings allow admins to further isolate the app
- Eliminate all unnecessary syscalls, process, and access to host resources
Run apps on the most secure environment

- The most secure container runtime and orchestration architecture
- Secure by default with out of the box configurations
- Cryptographic node identity
- Automatic mutual TLS across all nodes within the Docker cluster
- Transparent and automatic cert rotation
- External CA integration
- Optionally encrypt container to container traffic
Make apps safer with vulnerability scanning and monitoring

- Security scanning performs binary level scanning of application
- Detailed BOM provides security profile of application packages
- Make informed decisions before deployment
- BOM is maintained and continuously monitored against leading CVE databases
Leverage a secure and automated software supply chain

- Establish chain of trust with apps as they move across environments
- Digitally sign containers and only run verified containers
- Freshness guarantee ensures no tampering and latest container is running
- Automate workflow with immutable repos and automated image promotion
Granular access control for users, apps and nodes

- Restrict access to apps and resources
- Leverage predefined or custom roles available to manage access and permissions
- Create logical or physical isolation between apps and teams
Recap: Docker Enterprise Edition Capabilities

Certification and Support

Integrated App and Cluster Management

Optimized Container Engine

Certified Containers

Certified Plugins

Application Composition, Deployment and Reliability

Policy Management

Secure Access and User Management

Application and Cluster Management

Image Scanning and Monitoring

Content Trust and Verification

Image Management

Security

Network

Volumes

Distributed State

Container Runtime

Orchestration

Certified Infrastructure
Getting started

https://github.com/docker/communitytools-image2docker-win

https://github.com/docker/communitytools-image2docker-linux