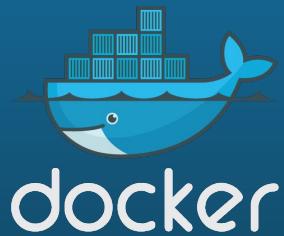


Modernize Traditional Apps Enablement Workshop

Fall 2017



Introductions

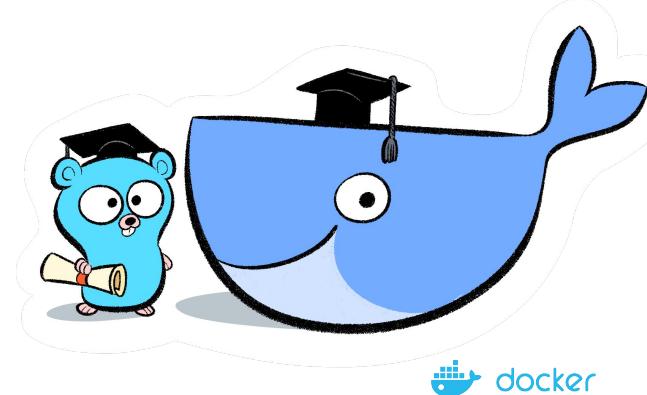
Who We Are:

- Bart Tubalinal - bart.tubalinal@docker.com - Solution Architect
- Mark Church - church@docker.com - Technical Account Manager



General Info

- Slack Channel Setup for the Group:
 - dockr.ly/community
 - Join the “partners” channel
- Lab access and documentations will be provided in the partners-workshop channel
- Required EE Advanced NFR License
- Team Work: Learn and Help Others
- Ask Questions



Day 1 Agenda

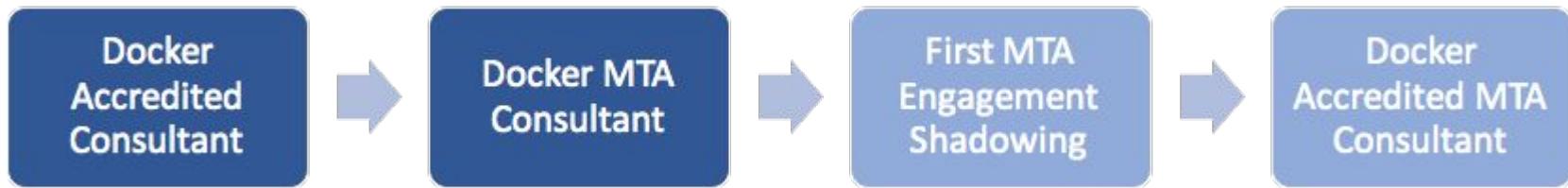
- About this Workshop
- MTA Introduction & Positioning
- MTA POC
 - Foundations
 - Roles & Responsibilities
 - Methodology & Tooling
- Hands-on Lab
- Wrap-Up



About this Workshop



Docker Accredited Consultant + MTA Consultant

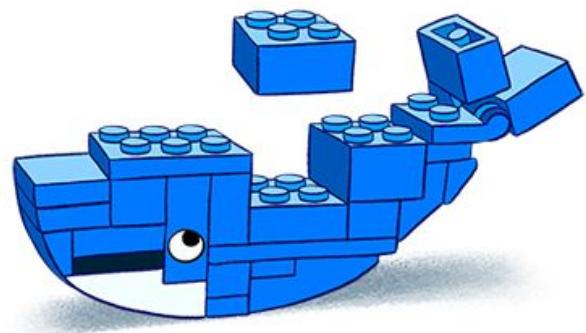


What we expect from you:

- Be interactive and hands-on
- Provide feedback on what you need to be successful
- Keep first engagement in mind - you'll be on the front line

MTA Accreditation Engagement Shadow

- Identify shadowing opportunities by region
- Co-deliver 1+ MTA POC
 - 2-4 Partner Docker Accredited Consultants
 - 1 Docker Solutions Architect (SA)
- DAC + MTA Accreditation on Successful Co



What this workshop will cover

- Understanding MTA
- Technical positioning of MTA POC
- Foundations of an MTA POC - Windows and Linux
- Mechanics and supporting assets for delivering MTA POCs

What this workshop *WON'T* cover

- Application development or framework deep-dives
- Application troubleshooting
- Details of follow-on activities (e.g. production launch, app operationalization, etc.)

MTA Introduction



Why MTA?

New microservices are great idea, but....

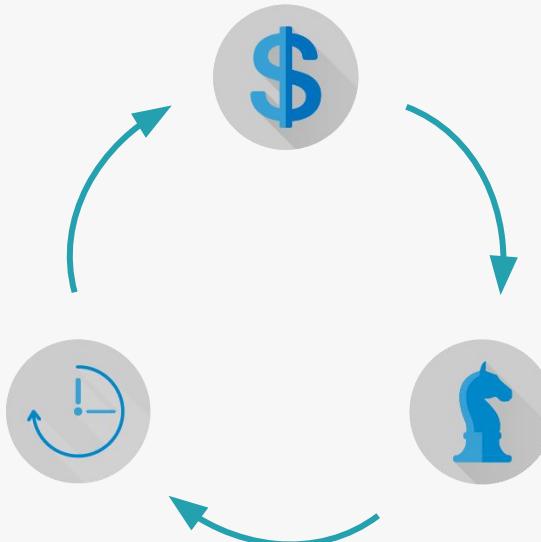
...what about the other 99% of apps already deployed?

"For a business application that is used for 15 years, the cost to go live is, on average, 8% of the lifetime TCO."

- Gartner

80% of IT Budget

Spent on maintenance & upkeep



Stuck Keeping the Lights on

Making it difficult to
keep up with
accelerating
standards

Application Changes

become too complex,
difficult, and/or costly
to implement



What Is A Legacy App Really?

The date in which that code was written isn't the only indicator that you're dealing with a legacy application. There's several other behaviors to keep an eye out for.



Contains a lot of lost knowledge



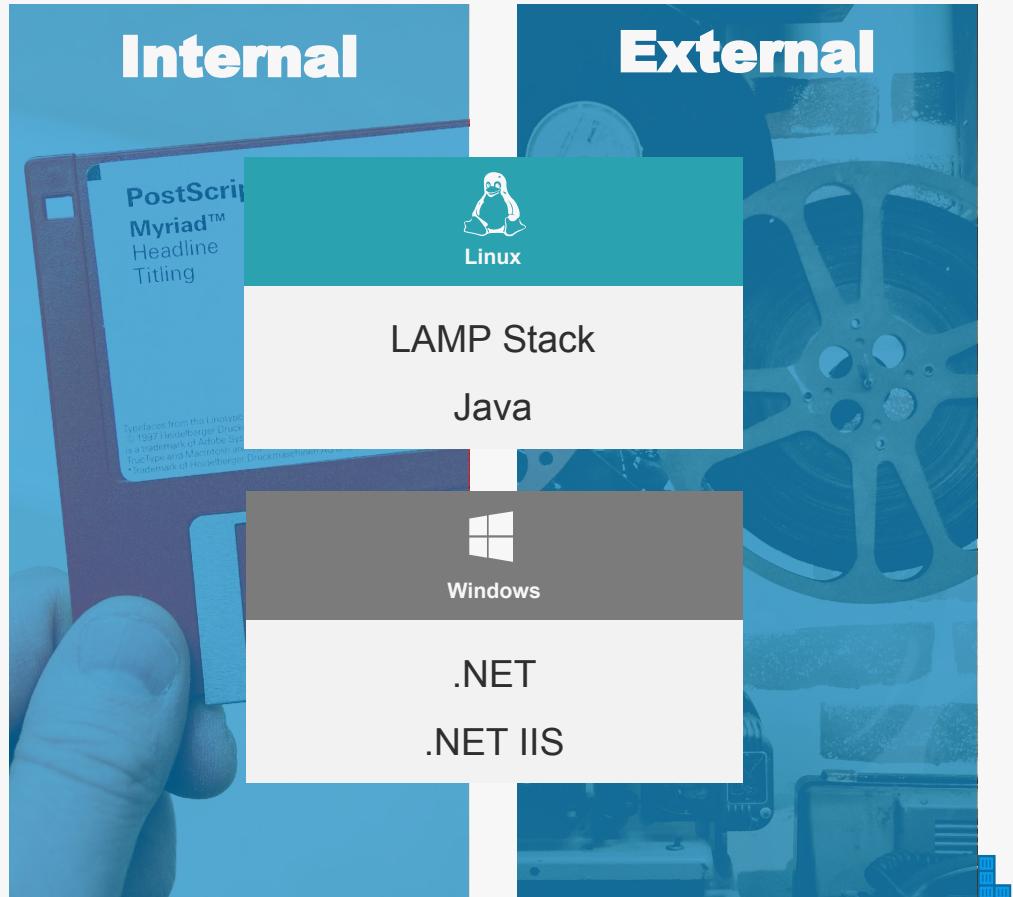
Most updates are band-aid fixes.



Dynamic scaling isn't possible, or takes way too long.



Maintenance windows are a quarterly or bi-annual event. And contain more anxiety than confidence



Understanding “Legacy” or “Traditional” Apps

Definition of “legacy” or “traditional” apps are relative

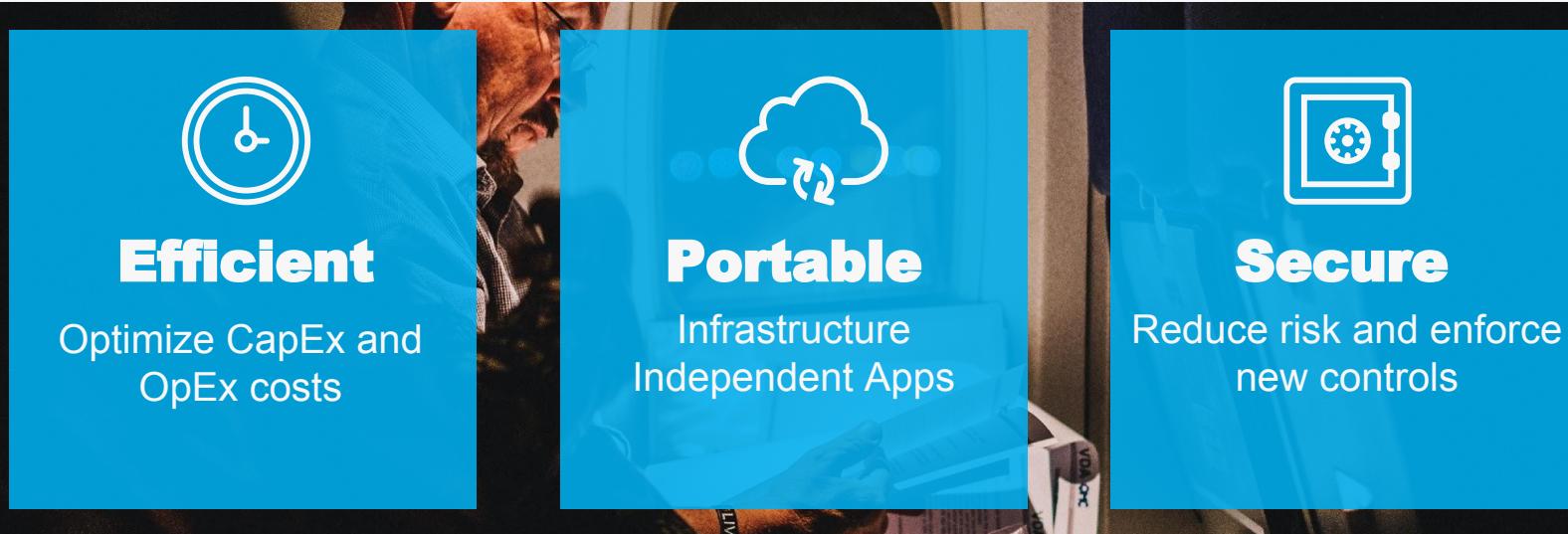
Docker Containers are fantastic tools for modernizing apps but they aren’t magic

Kernel compatibility is a foundational “litmus test” for Docker Container Compatibility



Figure 1. Introducing **Punch Card to Docker**...just kidding

Docker EE Gives Legacy Applications Modern Capabilities Without Any Recoding or Refactoring of the App



Size of Infrastructure

50%
Reduction

Deployment Speed

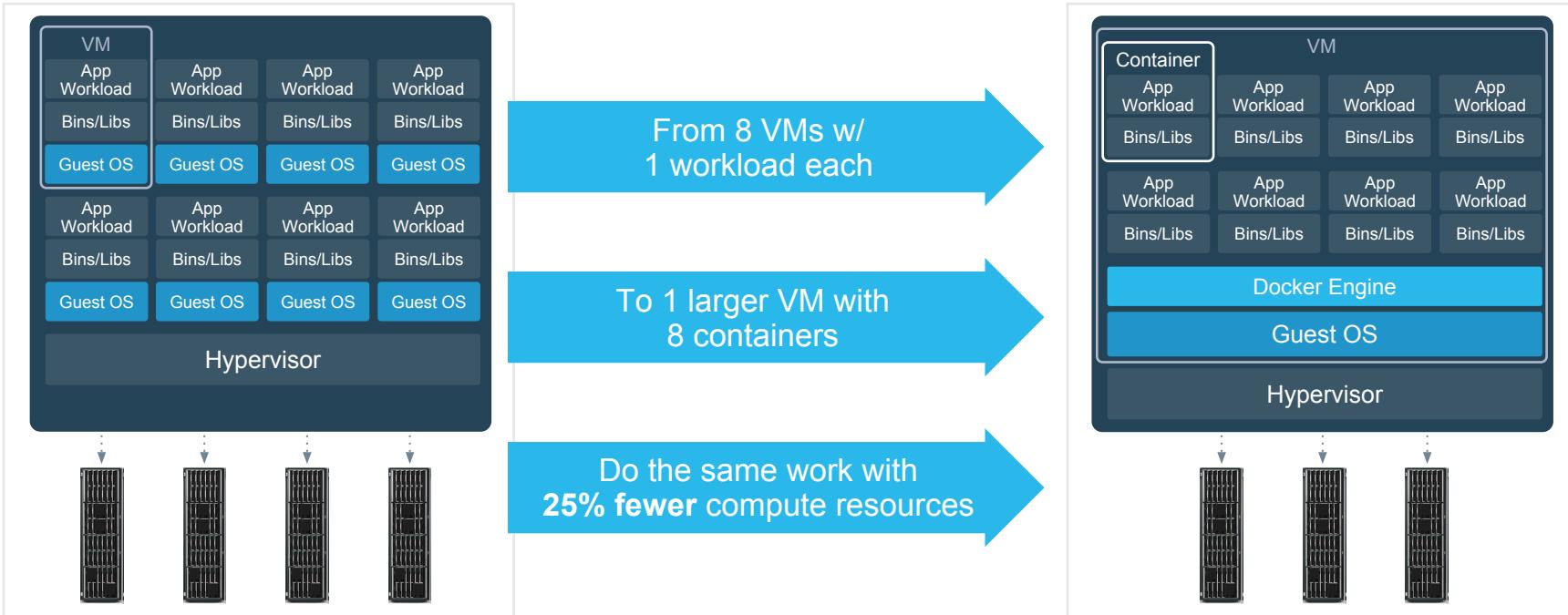
90%
Faster

MTTR for Patching

90%
Faster



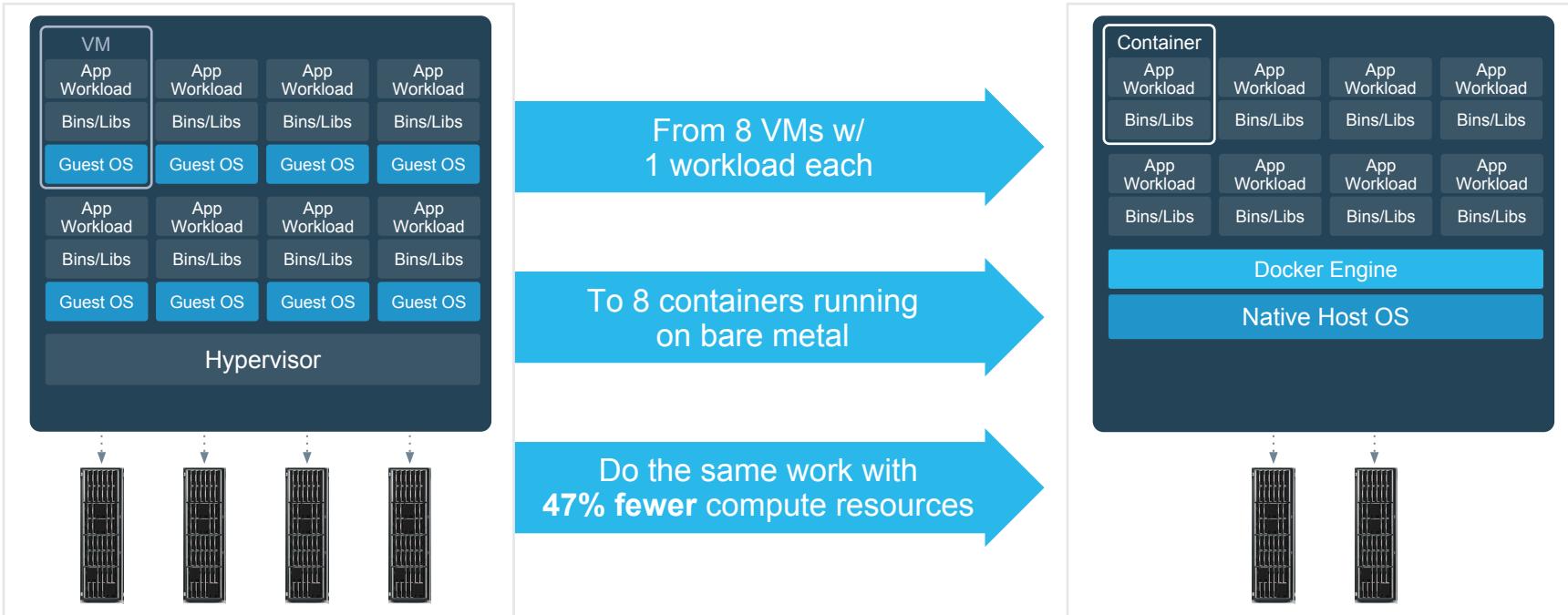
Infrastructure Optimization: VMs + Containers



ADDITIONAL SAVINGS:

30-35% less storage, 7% less RAM, Fewer OS's / Hypervisors to manage, Reduced data center footprint

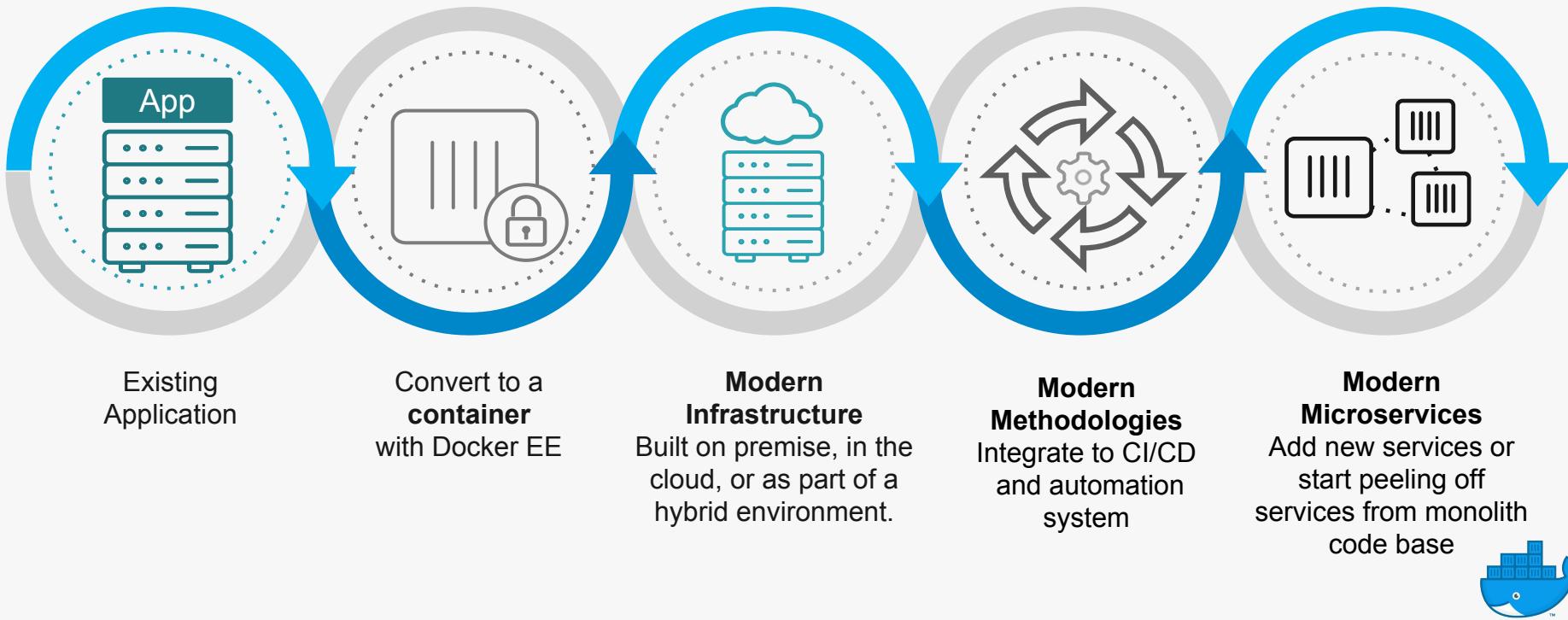
Infrastructure Optimization: Bare Metal



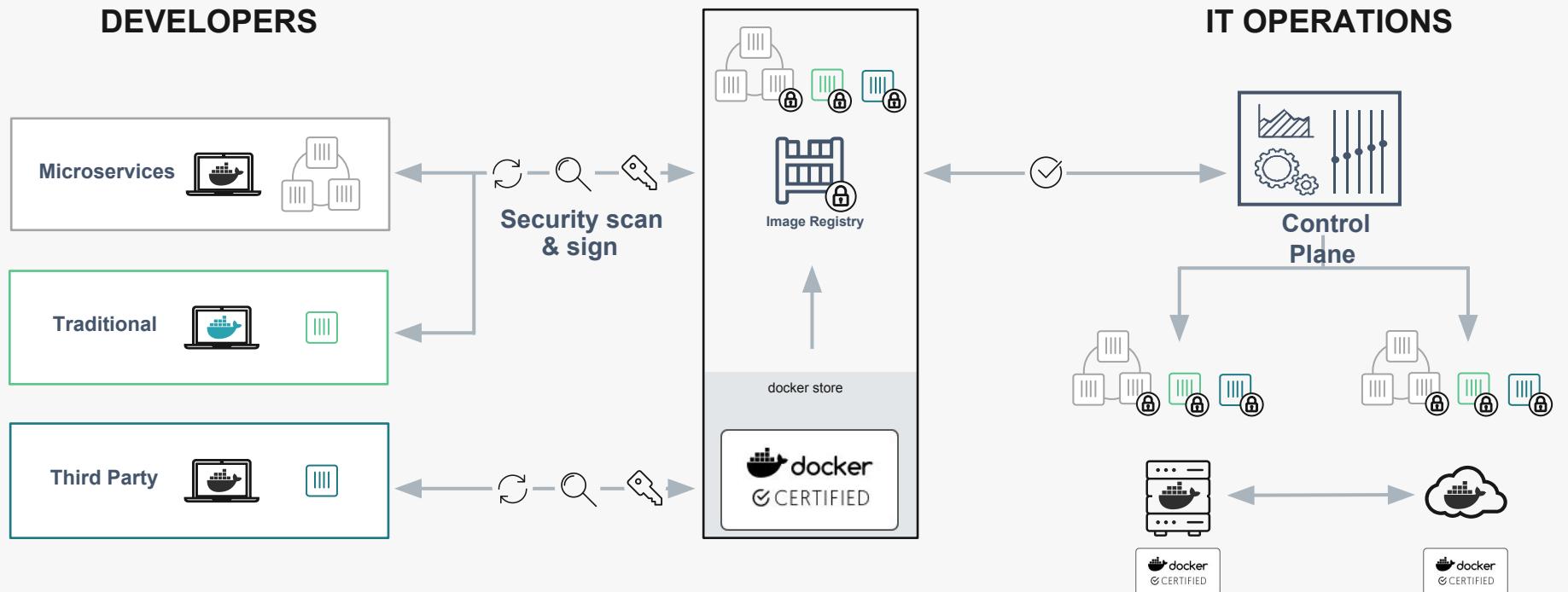
ADDITIONAL SAVINGS:

30% less storage, 7% less RAM, Fewer OS's / Hypervisors to manage, Reduced data center footprint

Methodology: Docker EE Modernizes Apps and Infrastructure

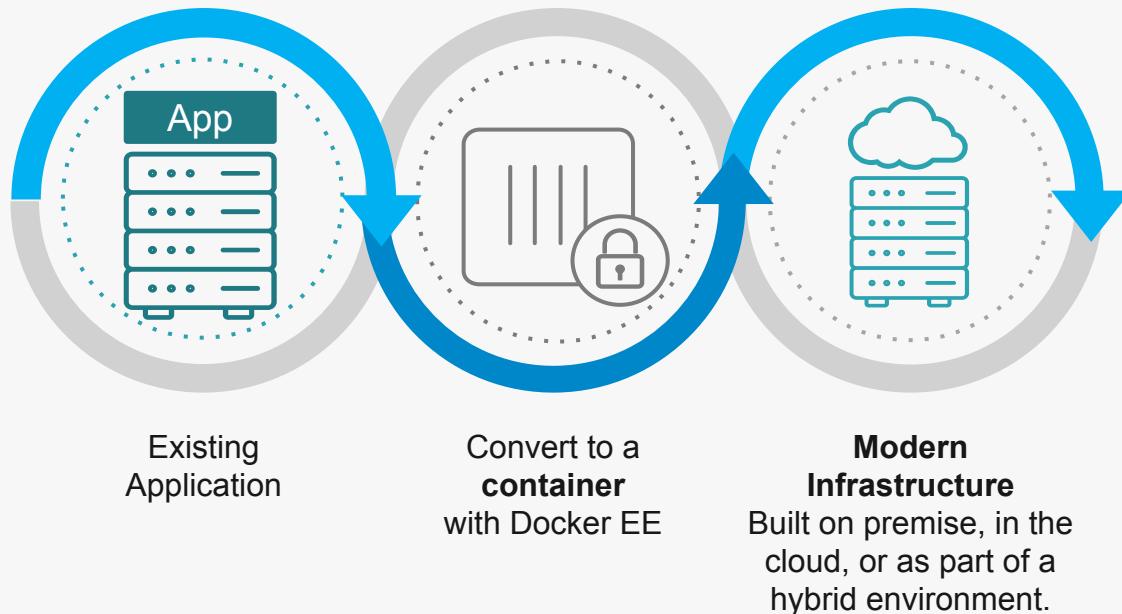


Docker EE CaaS for MTA And Beyond



Get Started Today

The quickest way to cut into that 80%



What's Included

- 1 week onsite support /3 weeks remote
- Deploy Docker EE to cloud or on prem infrastructure
- Containerize one application
- End-to-end app deploy using Docker EE
- App operations using Docker EE

In the first week

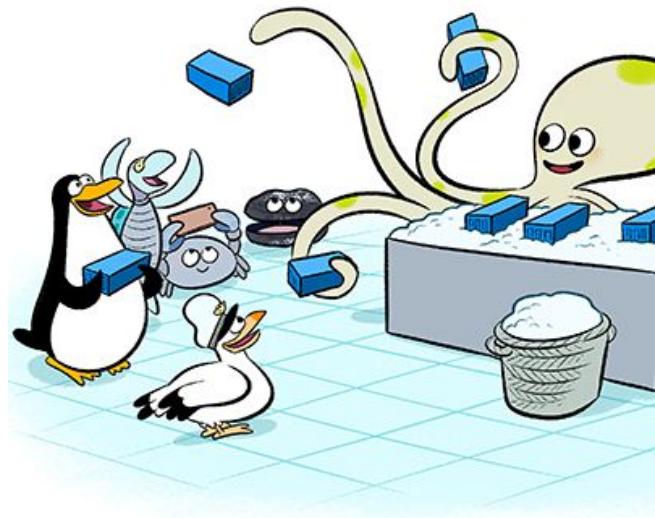
- Containerize App Components
- Compose App Components
- Deploy App Stack to Docker EE
- Docker EE platform and tools showcase
- Performance testing and tuning



What does an MTA POC include?

30-day POC

- Remote POC Prep
 - Infra Setup
 - App Assessment / Validation
- 5 days Onsite Working Session
 - Hands on keyboard work
- 3 Weeks Remote Support
 - Responsive support + platform inquiries



MTA Offerings

In just 30 days we can transform your Windows or Linux applications to run as a container making it more efficient, more portable, and more secure, all without touching a line of code.



Microsoft



Cloud



Cisco Advisory Services

Datacenter

Hewlett Packard
Enterprise

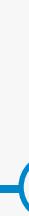
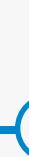
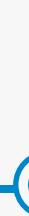
HPE
POINTNEXT

Datacenter

More Info: Docker.com/MTA



Where do we go from here?



We get together to further understand your business objectives & expectations

Identify an application you want to modernize as well as scope out resource plans

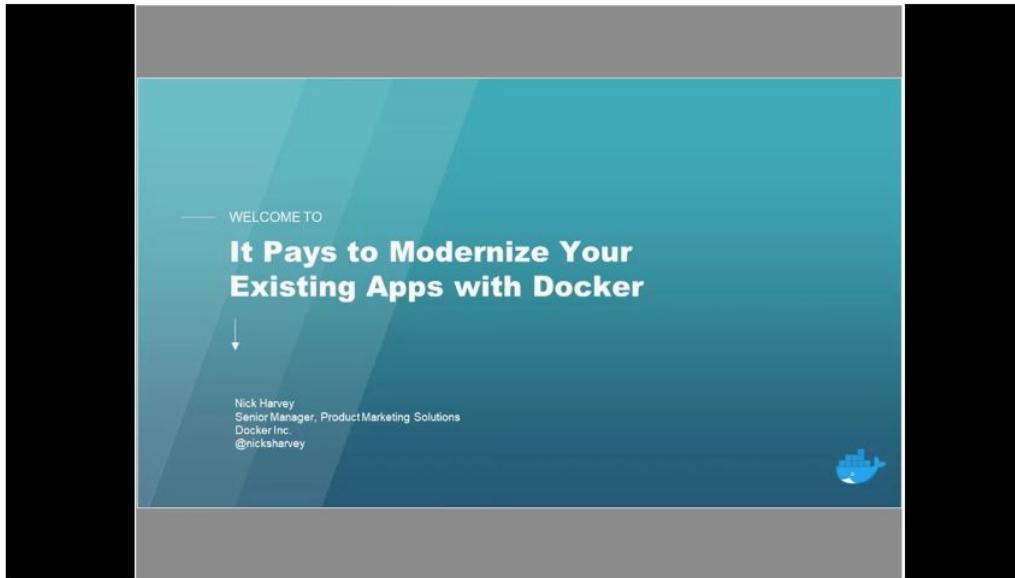
Pick a Start Date

More Info: **Docker.com/MTA**



Want more MTA Positioning?

Check out most recent [MTA Webinar on YouTube](#)



Modernize Traditional Apps (MTA) with Docker Enterprise Edition

864 views

1 like 5 dislike 2 share ...

MTA POC to Expand



docker

A wide-angle photograph of a two-lane asphalt road. The road is marked with a solid yellow center line and white dashed lines on the edges. It stretches from the foreground into the distance, where it disappears into a horizon line. The sky above is filled with dramatic, layered clouds. The left side of the sky is illuminated with warm orange and yellow tones, while the right side is cooler with blues and greys. The overall scene conveys a sense of vastness and journey.

MTA is the first stop on a long road...

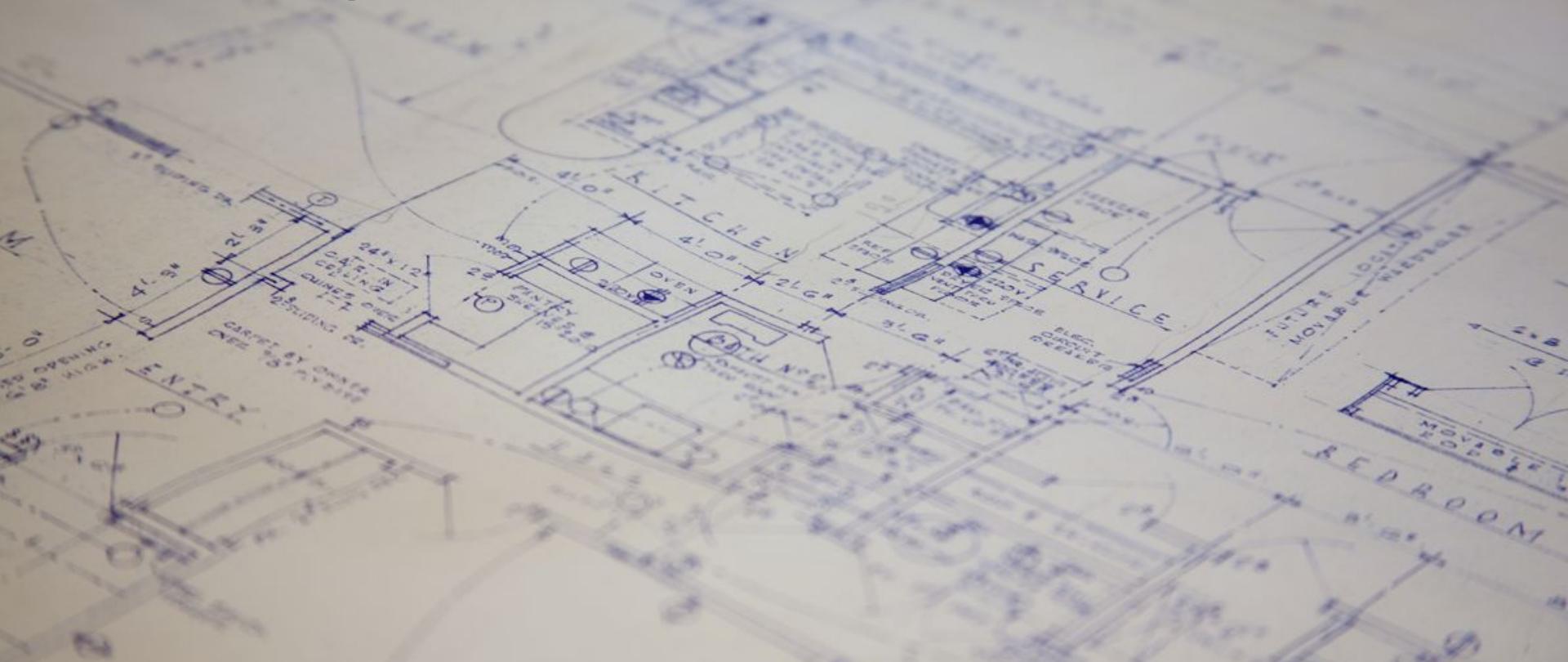
Beyond the POC

- MTA POC
 - 30-day POC
- **Expand Opportunities**
 - Operationalizing App
 - Integration Services
 - Production Launch
 - Operationalizing Additional Apps

MTA POC Foundations



POC Scope and Schedule



POC Scope

In-Scope	Out-of-Scope
<ul style="list-style-type: none">• Containerize and compose single app• Containerize compatible application server(s) where applicable• Install/configure Docker EE infrastructure• Demonstrate end-to-end deployment of app using Docker EE• Perform basic testing (baseline vs. containerized)	<ul style="list-style-type: none">• Integration with existing Docker infrastructure• App refactoring• Database management or deployment• Data migration• Integration with build solutions• App server migration/upgrades/etc.• High availability infrastructure configuration

POC Schedule

<u>Prep</u>	<u>Week 1</u>	<u>Week 2</u>	<u>Week 3</u>	<u>Week 4</u>
Prep Remote Infra Setup + App Validation <i>10-20 hours</i>				
	<u>3-5 Days Onsite</u> Onsite Working Sessions <i>30-40 hours</i>			
		<u>Remote / Self-Guided</u> Remote Support as Needed <i>10 hours</i>		

The background of the image is a blurred, motion-filled scene of an airport terminal. People are walking on a moving walkway, and the ceiling is filled with bright, colorful lights from various signs and advertisements. The overall effect is one of speed and movement.

POC Roles and Responsibilities

Partner Roles

Solution Architect:

Responsible for:

- Managing POC
planning, preparation
and *onsite execution*

Critical skillsets:

- Docker App Lifecycle
- Enterprise Integration
- Docker Networking

Application Engineer:

Responsible for:

- Application
examination,
containerization,
configuration and
deployment

Critical skillsets:

- Deep App, App Framework, Middleware expertise
- App Troubleshooting

Infra Engineer:

Responsible for:

- Deploying POC infrastructure

Critical skillsets:

- Docker EE Installation / Config
- App Dependent Infra
- Docker Networking

Customer Roles

Business Owner:

Responsible for:

- Driving entire MTA POC process on customer side

Critical skillsets:

- Technical
- Experienced in current process and pain points

Application SME:

Responsible for:

- Demonstrating initial “clean” app deployment
- Supporting app containerization

Critical skillsets:

- Application technical expert (i.e developer or architect)

Ops SME:

Responsible for:

- Supporting app containerization
- Providing intelligence around app portfolio and current state

Critical skillsets:

- Operations tech expert
- Experienced in deploying / troubleshooting app deployments

MTA POC Demo



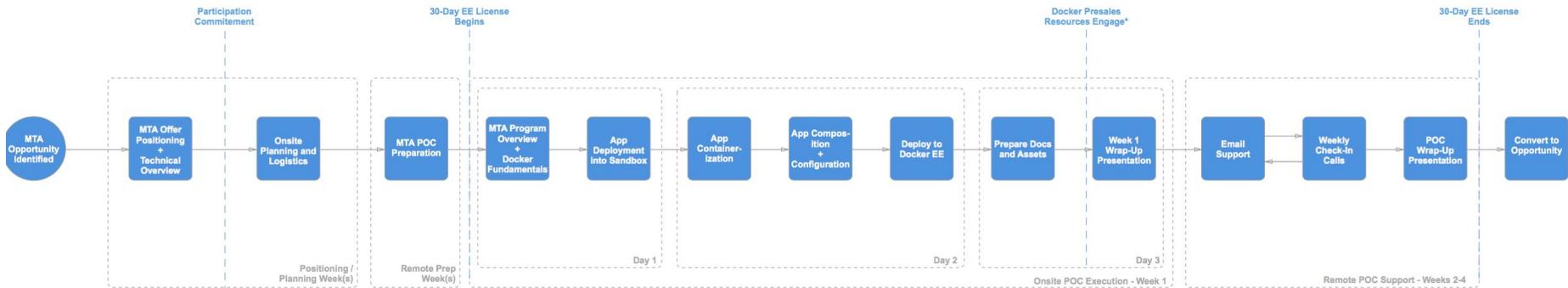
MTA POC Methodology



MTA POC Methodology Overview

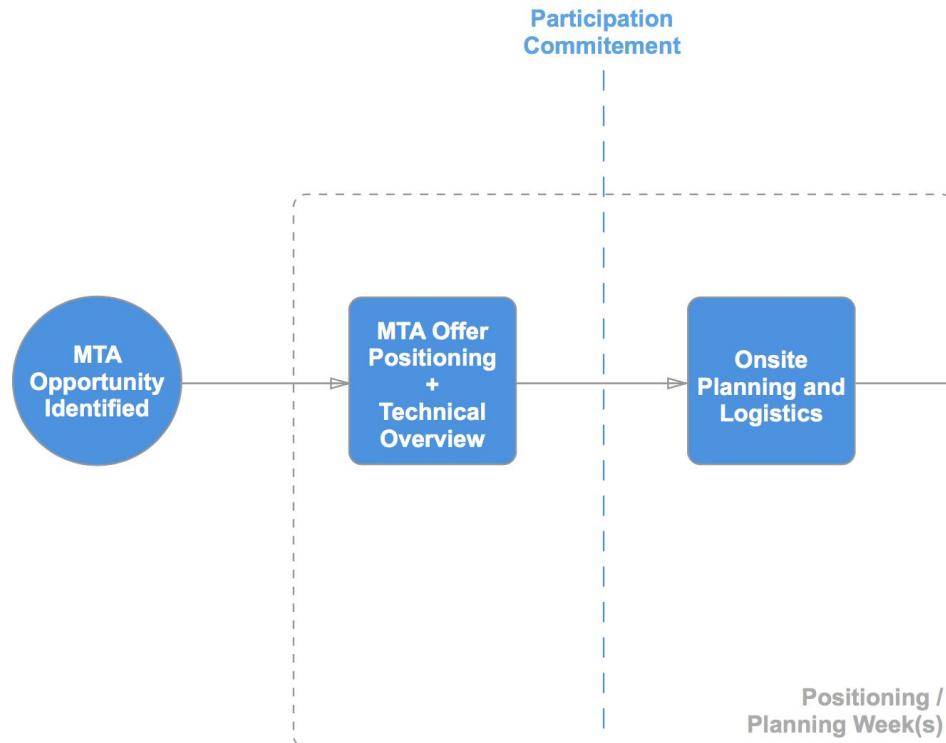
4 Stages

- Positioning / Planning
- Preparation
- Onsite Execution
- Remote Follow-Up / Support



MTA POC Methodology

Positioning / Planning Phase



MTA Positioning + Technical Overview

- Support use case and program positioning
- Review technical details of program

Resources

MTA Positioning Deck

MTA Solution Brief

MTA Technical Overview

MTA Infrastructure Diagram

Roles & Responsibilities

Partner Solution Architect:

- Supporting the technical portions of the positioning including infrastructure and application selection
- Running through technical deck.

Onsite Planning and Logistics

- Cover planning and logistical details of POC leading up to the onsite including app selection and infra setup

Resources

MTA POC Planning

POC Schedule

App Assessment Guide - .NET

App Assessment Guide - Java

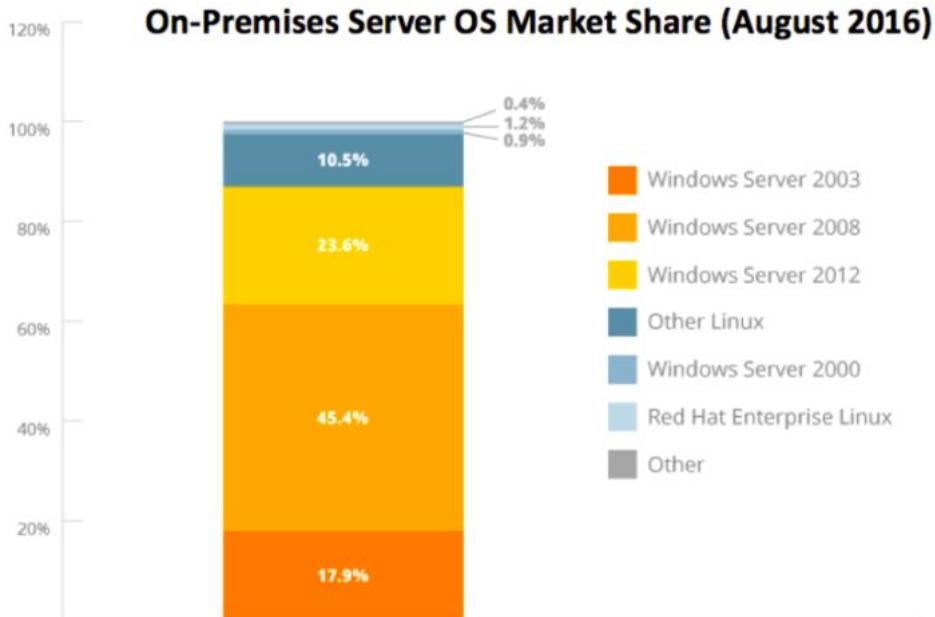
Roles & Responsibilities

Partner Solution Architect:

- Running through planning deck
- Coordinating consensus of success criteria and scope
- Documenting functional requirements for target application (e.g. what constitutes a functioning application)
- Addressing questions about overall process
- Addressing questions about infrastructure setup
- Addressing questions about app selection

POC App Selection

OS & Application Landscape



Vast majority of traditional apps running on **legacy OSes**

~80% .NET Framework or Java

App Selection / Prioritization

POC Objectives for Selecting Apps



- Select for success - can be containerized in 3 days of effort or less
- Representative of customer applications (app servers, frameworks, OS, etc.)
- App includes engineering level (developer, architect, etc.) expertise in customer team
- Limited dependencies > limited unknowns and teams need to be involved

App Selection / Prioritization

General Guidance

Linux	Windows
Java EE	.NET Framework
Alignment with existing initiatives and/or has some business value	
Components fundamentally compatible with modern OS (e.g. RHEL 7, WS 2016, etc.)	
Server-side only components (no Desktop GUI)	
Requires little / no refactoring	
Manageable dependencies / able to run in as close to isolated environment as possible	

App Selection / Prioritization

Runtime / App Servers

Linux	Windows
Java EE	.NET Framework 2.0+
	ASP.NET, Windows Services
Tomcat, WebLogic, WebSphere, Jetty, Glassfish (*)	IIS 6, 7 or 8
Custom applications - Commercial off the Shelf (COTS) not preferred	

*Must be compatible with modern OS in current version - no app migration



App Selection / Prioritization

Architecture

Linux	Windows
2 or 3 tier app	
Up to 5 applications components*	
Components do not require clustering (i.e. can be run as a single node)	
Databases can be excluded for containerization (i.e. can be migrated / connected to independently)	

*Independent runtime components (e.g. service, job, queue, proxy, etc.)

App Selection / Prioritization

Database Backend

Linux	Windows
Oracle, MySQL, Postgres, DB2	SQL Server
Basic Authentication	SQL Auth, Integrated Auth*
Decoupled from app (i.e. can be deployed independent from container)	

*Requires domain configuration w/ Active Directory gMSA

App Selection / Prioritization

User Authentication

Linux	Windows
None, Basic, Federated	None, Basic, Federated, Windows*
	No domain level user impersonation

*Requires domain configuration w/ Active Directory gMSA

App Selection / Prioritization

Data Persistence

Linux	Windows
Session state persistence managed external to the app is ideal*	
Runtime data written to external database or network share is ideal	
Data written to disk in known location	

*If single node, in-memory session state only - sticky load balancer is required

App Selection / Prioritization

Misc.

Linux	Windows
Service dependency endpoints should be domain name resolvable and configurable (i.e. no hardcoded IPs or domain names for services that need to be containerized)	
Application startup time should be < ~1 minute	
All components should be installable through unattended installer or from raw binaries, files and configurations	

QUIZ - Question #1

Positioning / Planning

Is the following a qualified MTA POC app? Why?



SharePoint Web Tier

ASP.NET App Tier

MSMQ

SQL Server Backend

QUIZ - Question #2

Positioning / Planning

When selecting an app, what other resources (besides the app artifacts themselves) need to be made available?



QUIZ - Question #3

Positioning / Planning

What's an example of an app implementation detail that will break Docker Service Discovery?



QUIZ - Question #5

Positioning / Planning

What's the minimum .NET Framework version supported for Windows MTA POCs?

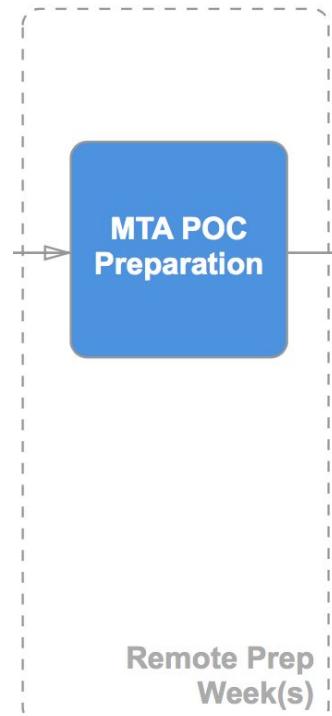


BREAK



MTA POC Methodology

Preparation Phase



MTA POC Preparation

- POC infrastructure installation and setup
- Final application selection and technical drill-down

Resources

Infrastructure Readiness Checklist

App Readiness Checklist

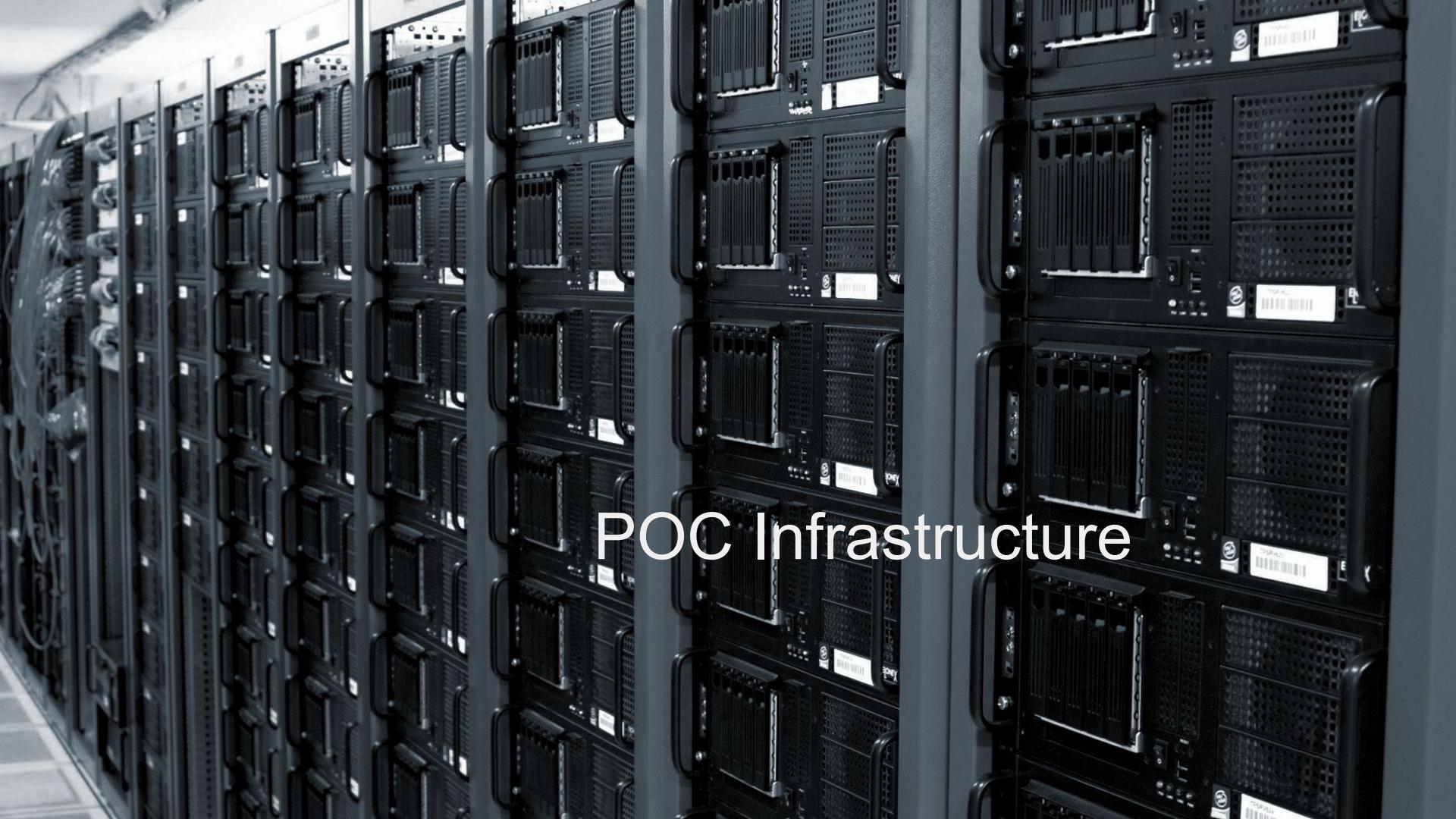
Roles & Responsibilities

Partner Solution Architect:

- Confirm infrastructure is setup to meeting POC requirements
- Review details of selected application(s) to check for technical fit
- Confirm application deployment artifacts and SME will be available for onsite week

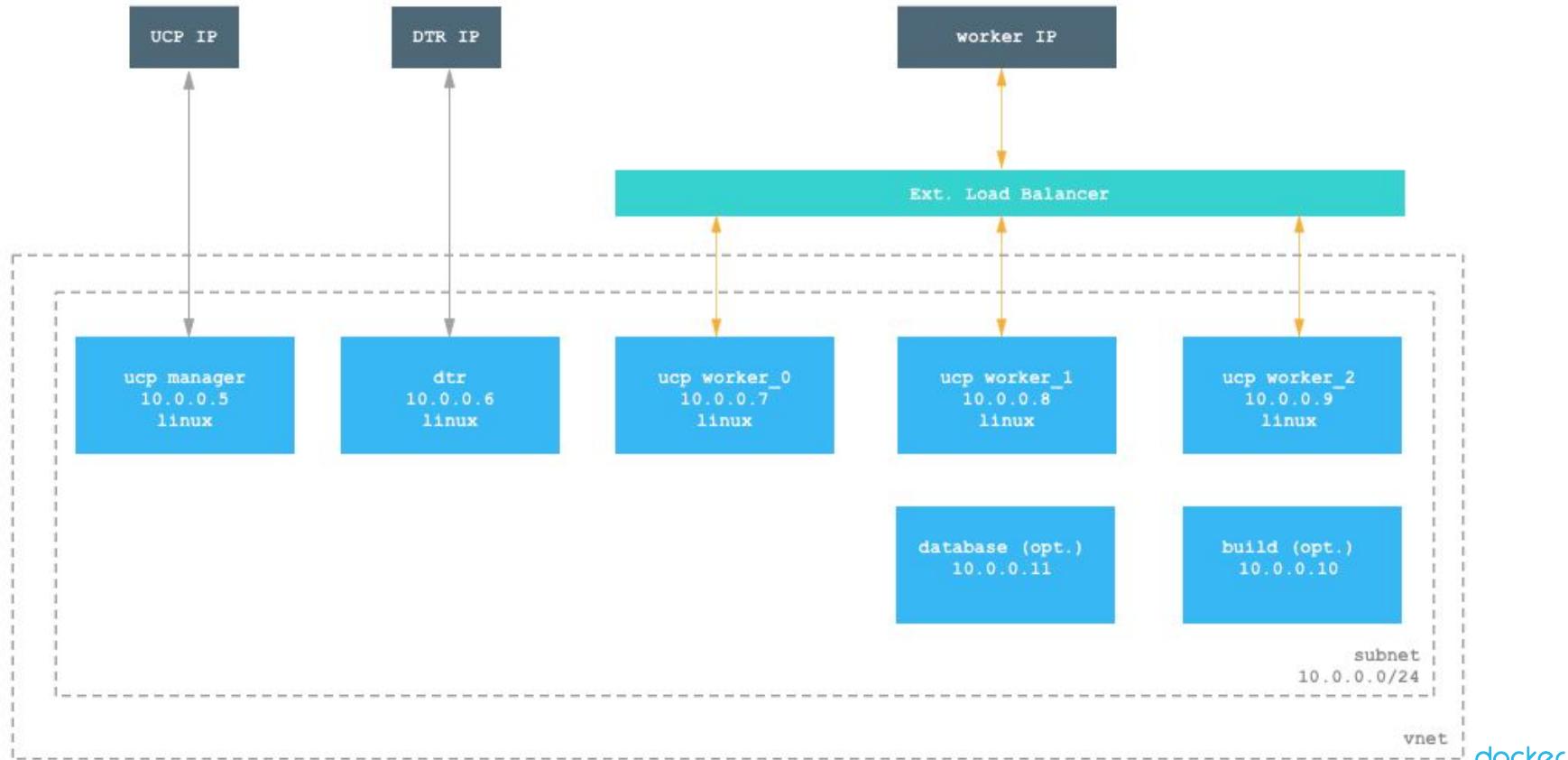
Infra Engineer:

- Deliver infrastructure diagram, requirements and technical details
- Ensure infrastructure is setup to meet POC requirements

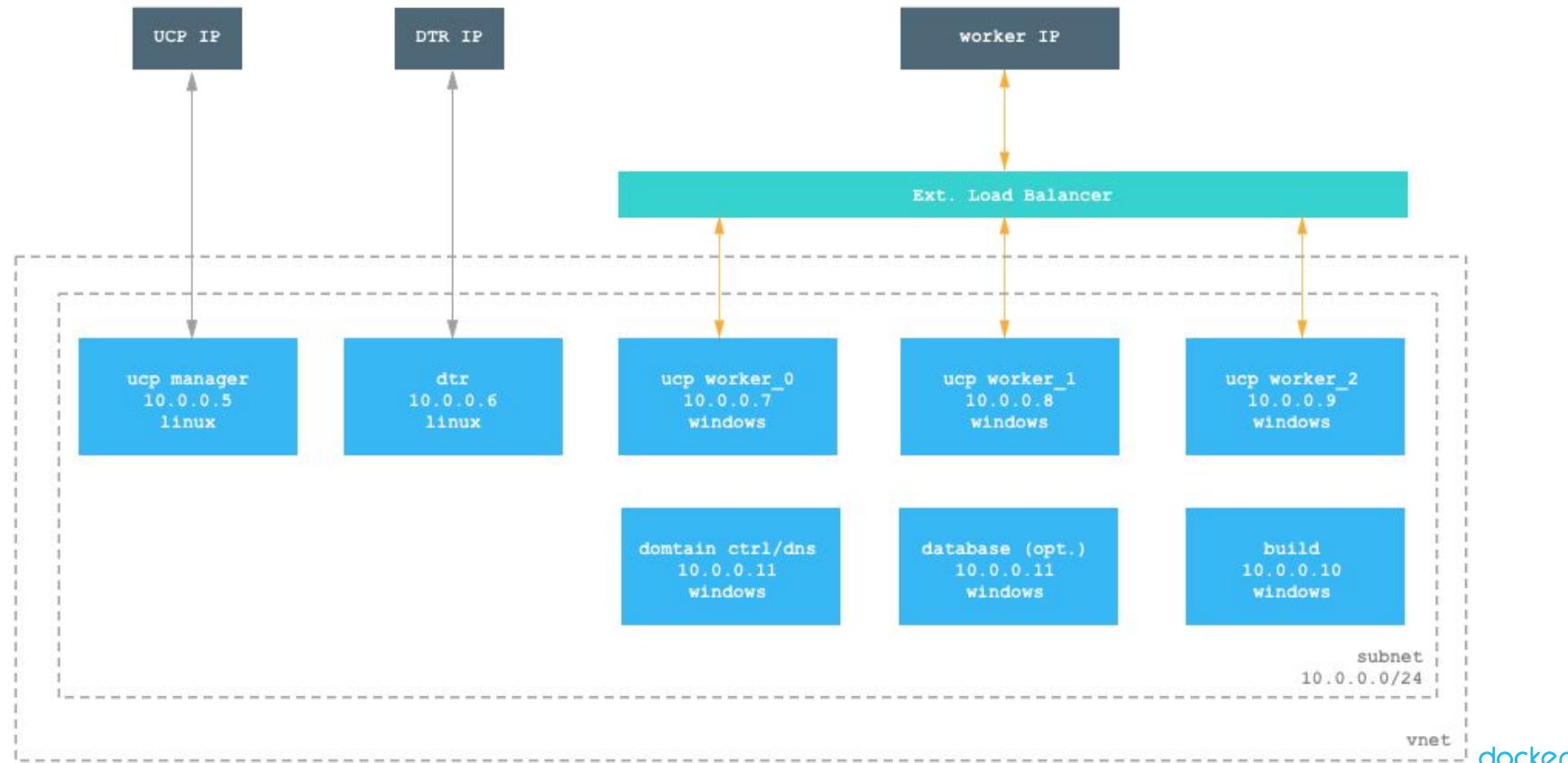
A black and white photograph showing a dense row of server racks in a data center. The racks are filled with various components like drives and network cards. The perspective is from a low angle, looking up at the top of the racks.

POC Infrastructure

POC Infrastructure - Linux



POC Infrastructure - Windows



POC Minimum Infrastructure Checklist

- Servers
 - 1 Linux* node - UCP Manager
 - 1 Linux* node - DTR (single replica)
 - 3 Linux* nodes or 3 Windows Server 2016 VMs - UCP Workers
 - 1 Windows Domain Controller node (Windows Apps Only)
 - 1 Linux or Windows Database node
 - 1 Linux* or Windows Build node
- Networking
 - 1 vNet and subnet
 - 1 load balancer - UCP Workers as backend pool
- Storage
 - 40 GB per VM w/ 80 GB on DTR VM

*Compatible Docker EE Distribution

QUIZ - Question #1

Preparation Phase

Why does MTA POC infrastructure leverage isolated infrastructure? Provide one example.



QUIZ - Question #2

Preparation Phase

What infrastructure is required for Windows that is optional for Linux apps in an MTA POC?



QUIZ - Question #3

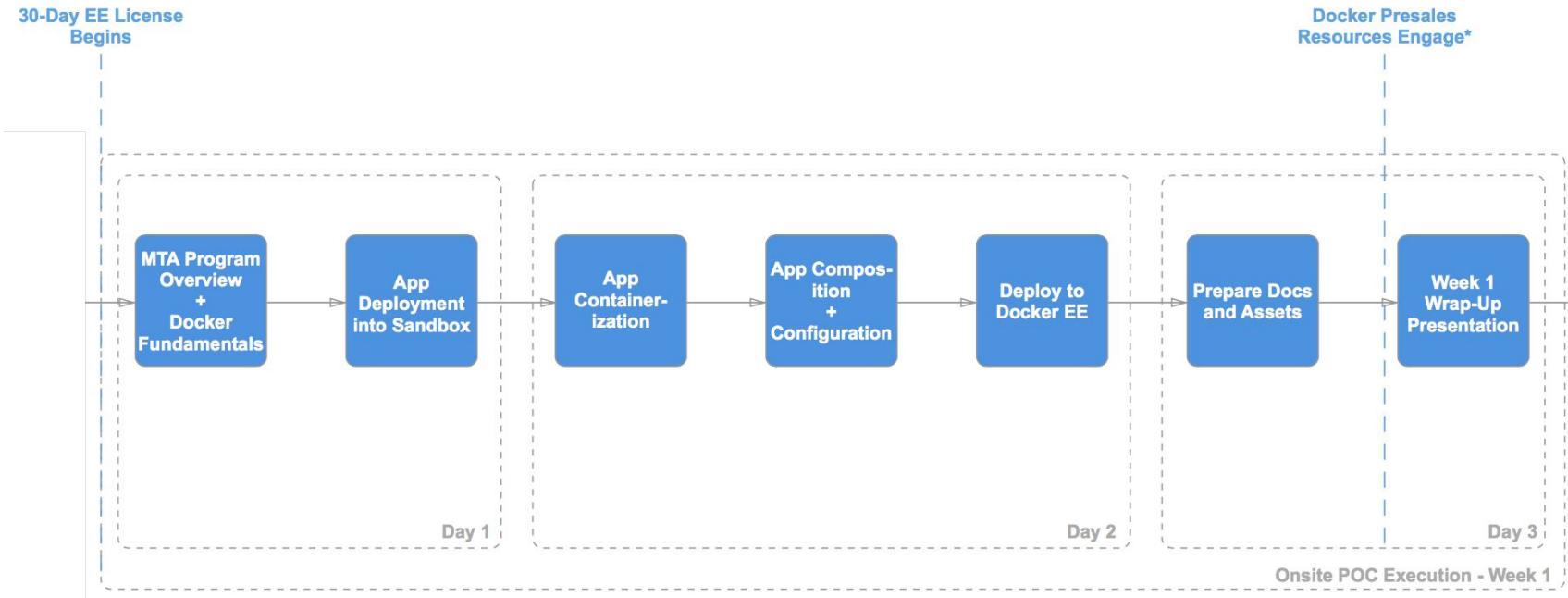
Preparation Phase

What infrastructure is optional for Windows that is not relevant for Linux apps in an MTA POC?



MTA POC Methodology

On-Site Execution Phase



Program Overview + Docker Fundamentals

- Set program / logistical context for the week
- Provide foundational knowledge of Docker + Docker EE

Resources

MTA Program Overview

MTA Docker Fundamentals

Roles & Responsibilities

Partner Solution Architect:

- Review MTA Program Overview Presentation
- Review Docker Fundamentals*

Docker Sales Engineer (optional):

- Review Docker fundamentals

*if Docker Sales Engineer not available



App Deployment into Sandbox

- Deploy app to “build” machine
- Determine deployment baselines and functional success level of app

Resources

Customer Evidence Workbook

Roles & Responsibilities

Customer App SME:

- Deploying target application to “build” server, explaining the artifacts and steps required to successfully deploy the application
- Completing application deployment, ensuring it’s functioning to agreed upon functional level

Partner Solution Architect:

- Coordinating deploying process with customers
- Collecting Application information and filling out Customer Evidence Worksheet

Partner Application Engineer:

- Recording application deployment steps

App Containerization

- Create Docker using Image2Docker or manually
- Successfully run containerized app on “build” server

Resources

Image2Docker - Windows

Image2Docker - Linux

Dockerfile Best Practices

Docker Credential Specs - Windows

Roles & Responsibilities

Partner Solution Architect:

- Describe process of containerization, educating customer on tooling (e.g. Image2Docker) and general Dockerfile best practices

Partner Application Engineer:

- Deploy baseline file structure to location on build server
- Running Image2Docker against build server and creating baseline Dockerfile
- Modifying Dockerfile to capture any missing configurations
- Modifying App configurations (as necessary) to connect to dependencies (e.g. databases, services, etc.)

App Containerization (Continued)

Roles & Responsibilities

Customer Application SME:

- Troubleshooting issues that may arise during the containerization process regarding the app
- Assisting Application Engineer with understanding functions of application relevant to containerization process

App Composition & Configuration

- Build and test application stack w/ docker-compose
- Externalize configuration

Resources

Docker Compose Guide

Variable Config Injection

Roles & Responsibilities

Partner Solution Architect:

- Educating customer on best-practices for externalizing configurations - using environment variables and secrets at runtime

Partner Application Engineer:

- Creating/modifying docker-compose.yml and docker-compose.prod.yml file to match target application stack
- Externalizing environment specific application variables into environment variables

Deploy to Docker EE

- Deploy app as Docker Service / Docker Stack to Docker EE
- Validate app functionality

Resources

Docker Compose Guide

Variable Config Injection

Roles & Responsibilities

Partner Solution Architect:

- Educating customer on Docker EE deployment process using their app

Partner Application Engineer:

- Deploying app as a service and/or stack to Docker EE
- Validating application functions based on baseline functional requirements

Prepare Customer Docs and Assets

- Prepare documentation package
- Prepare week 1 wrap-up presentation

Resources

- Week 1 Wrap-Up Presentation
- Customer Docs Package

Roles & Responsibilities

Partner Solution Architect:

- Preparing onsite wrap-up presentation
- Drafting customer wrap-up email to customer, including application and infrastructure information

Partner Application Engineer:

- Preparing application deployment document, samples and scripts
- Preparing application information (stats, screenshots, etc.) for wrap-up presentation
- Drafting application overview email template - providing to Solution Architect
- Handing-off all remaining application information to Solution Architect

Prepare Customer Docs and Assets (Cont.)

Roles & Responsibilities

Partner Infra Engineer:

- Preparing infrastructure diagrams and documentation
- Drafting environment access / infrastructure overview email template - providing to Solution Architect
- Handing-off all remaining infrastructure information to Solution Architect

Week 1 Wrap-Up Presentation

- Deliver week 1 wrap-up presentation

Resources

Week 1 Wrap-Up Presentation

Roles & Responsibilities

Partner Solution Architect:

- Co-presenting wrap-up presentation
- Filling out Customer Evidence worksheet
- Sending customer wrap-up email to customer
- Hand-Off ALL POC information to Docker Sales Engineer

Docker Sales Engineer:

- Co-presenting wrap-up presentation
- Schedule weekly check-in calls for weeks 2-4 + final POC wrap-up call
- Establish support process

QUIZ - Question #1

On-Site Execution Phase

What are the minimum customer roles required (i.e. in the room) for application containerization?



QUIZ - Question #2

On-Site Execution Phase

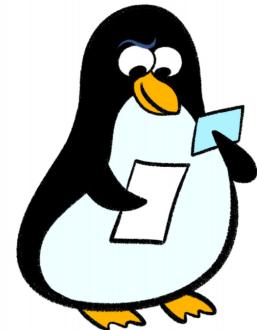
During app containerization, what are the three most common **Docker commands**?



QUIZ - Question #3

On-Site Execution Phase

What artifacts need to be created during **application composition and configuration?**



QUIZ - Question #4

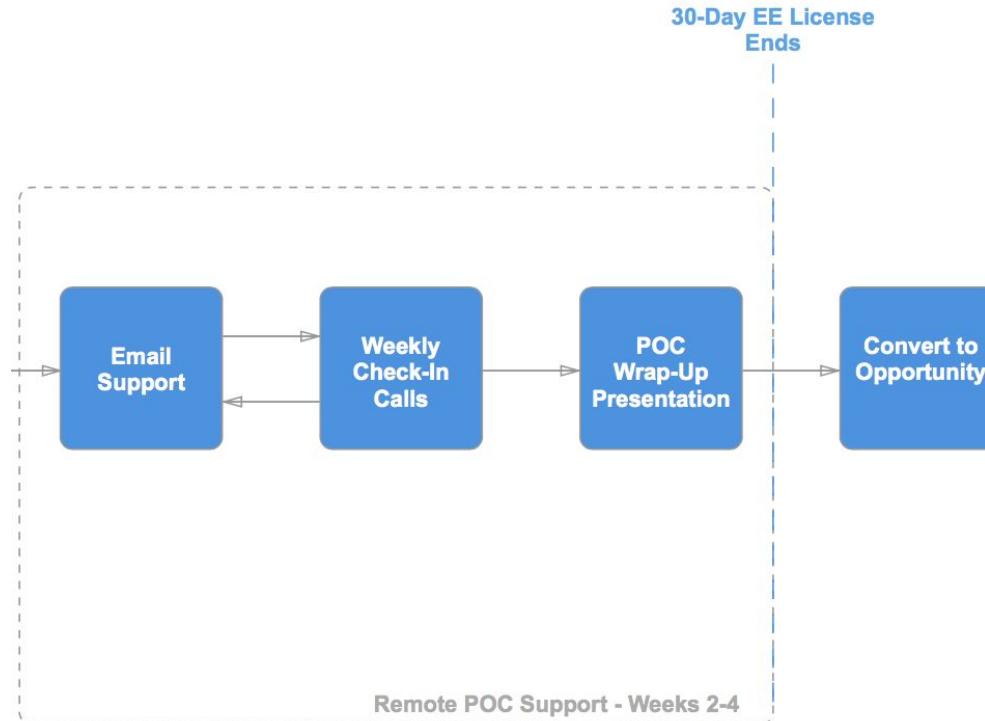
On-Site Execution Phase

If the target application relies on in-memory session state, what networking configuration is required?



MTA POC Methodology

Remote Follow-Up / Support Phase



Remote Email Support

- Triage / answer questions or issues encountered during remote evaluation weeks

Resources

Docker Knowledge Base

Roles & Responsibilities

Docker Sales Engineer:

- Answer and/or triage any issues that the customer encounters. Triage resources include Docker Support and Solution Architect assigned to POC

Weekly Support Check-In

- Triage / answer questions or issues encountered during remote evaluation weeks

Resources

Docker Knowledge Base

Roles & Responsibilities

Docker Sales Engineer:

- Coordinate and lead call
- Answer and/or triage any issues that the customer encounters. Triage resources include Docker Support and Solution Architect assigned to POC

POC Wrap-Up Presentation

- Triage / answer questions or issues encountered during remote evaluation weeks

Resources

POC Wrap Up Presentation

Roles & Responsibilities

Docker Sales Engineer:

- Coordinate and lead call
- Answer and/or triage any issues that the customer encounters. Triage resources include Docker Support and Solution Architect assigned to POC

QUIZ - Question #1

Remote Follow-Up / Support Phase

Who is responsible for leading the POC Wrap-Up presentation?



MTA Technical Considerations



Networking

A majority of traditional apps manage state in memory (e.g. session state, view state, etc.)

- HRM supports sticky sessions but extra considerations need to be made concerning service rescheduling and rolling updates
- A possible workaround is to pin the service to nodes, effectively not allowing the replicas reconcile state
 - Allows for sticky sessions to work but you lose some Swarm mode benefits

Required

- External load balancer listening to containers using host mode

Storage

- If persistent storage is required there's a couple of options:
- NFS share on a NAS device
 - Works well for non-intensive I/O
 - Is supported by the local driver, will be available across worker nodes
- 3rd party volume driver
 - With the proper hardware/driver, has the ability to re-mount volumes and effectively follow the containers around the cluster to various worker nodes

QUIZ - Question #1

If the target application relies on in-memory session state, what networking configuration is required?



BREAK



Hands-on Lab



MTA Labs - Let's Get Hands-On

Labs

- **Required:** MTA In Practice
- **Optional:** Modernize .NET Apps for Ops

```
docker run -d -p 8080
```

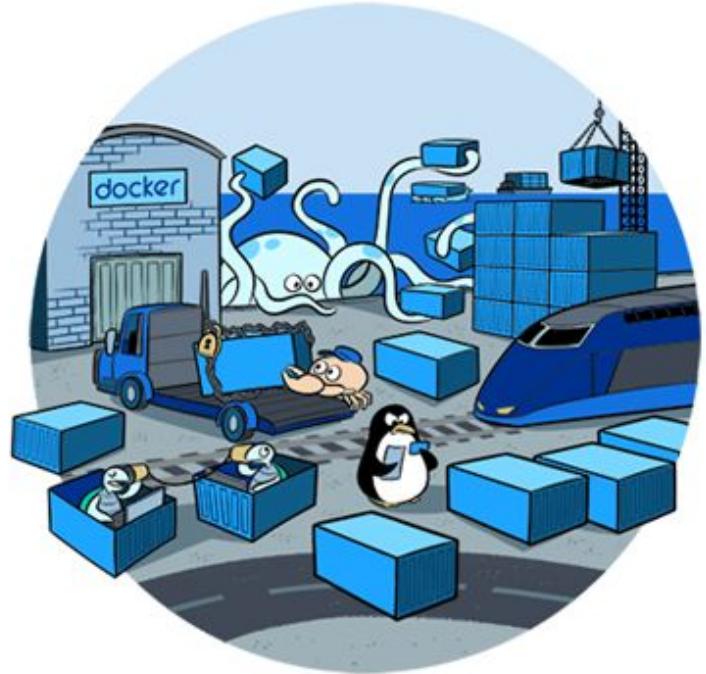


Call To Action

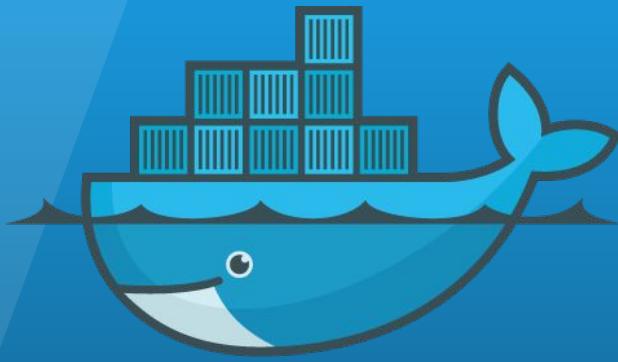


MTA Next Steps

- Prepare for MTA POC engagement shadowing
 - *We'll be scheduling customers in each region soon!*
- Do an MTA POC dry-run
 - Find an app and get started



Thank You!



docker

Appendix

Extra Slides



docker

Breaking down the deployment savings

App deployments before and after Docker

Before : Traditional App Deployment : Manual, Risky, Slow

Take Offline

- Long running processes with several manual steps
- Scheduled out of hours
- Disruption to users

Deploy

- Lengthy Install Guide(50 pages, 100 man hours to write) usually word document and mostly inaccurate
- Bloated App binaries
- Bloated App files

Smoke Test

- Bloated test documents
- Requires prior knowledge of the app
- Manual tests requires Dev and Ops

Acceptance Test

- Manual bloated regression pack, takes multi hours

Go/No-Go

- Low confidence rate
- Rollback is repeat of the entire process

~100 man hours

After : Modern App Deployment : Automated, Proven, Fast

Take Offline

- Need not be scheduled out of hours
- No disruption to users

Deploy

- ONE single command
- ONE light Docker image
- Built in health checks

Acceptance test

- Automated Regression Pack
- Rapid addition of new features

Go/No-Go

- High confidence rate
- Fast rollback repeatable

~<24 man hours

Northern Trust

An MTA Case Study

Northern Trust - An MTA Case Study

- Company Overview
- Environment overview
- Stakeholders
- POC Engagement
 - Preparation
 - Selecting App
 - 5 days onsite
 - Remote support
- Conversion
 - Business value supported by POC metrics
- Services Engagement
 - Build process integration
 - Pilot to production
 - Modernize to micro-services

Company Overview / Environment



NORTHERN
TRUST

Overview

Founded in 1889, Northern Trust is a global leader in asset servicing, asset management, and banking for personal and institutional clients.

Environment

Primarily on-prem infrastructure today. Focused on data center optimization and expanding into public cloud infrastructure. Current deployment model is manual and app specific. Primary apps are .NET, Tomcat and WebLogic

Stakeholders

Enterprise middleware team including operators and developers.

POC Engagement

Objective

Align with cloud initiative to showcase Docker EE expansion into Azure - containerizing existing on-prem app and deploying to cloud

Remote Preparation

Coordination with business stakeholders and technical team on app selection. Selected 3 apps (1 primary, 2 backup). Coordinated with infra team on Azure subscription and template.

Onsite Working Session

Installed infra with Docker EE Adv. Containerized and deployed apps

Remote Support

Resolved open issues and measured process and infra optimization against baseline metrics



Services Engagement

Conversion

Metrics support Northern Trust going all in on Docker EE

Uncovering Service Opportunities

- Establishing operational model for containerizing 400+ apps
- Modernizing dev/deployment toolchain with Docker
- Data center modernization with heterogeneous workloads
- Hybrid cloud implementation strategy

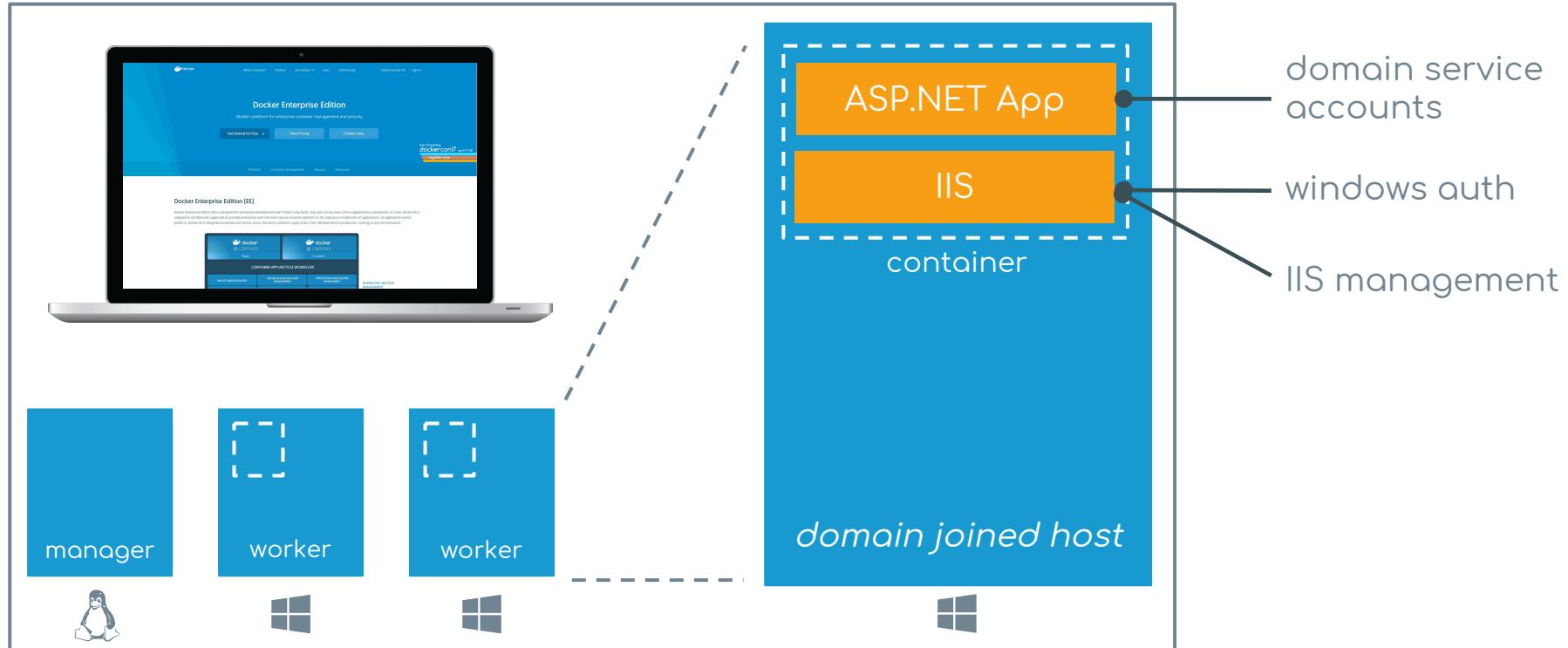
MTA Projects

Uncovering Opportunity

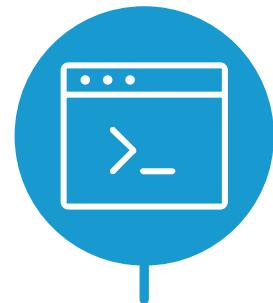
Faster Deployments with Less Infrastructure



Legacy Patterns in Modern Infrastructure



Hybrid Cloud Scalability



Declarative
Docker Services



Health
Checks



Complete
Service
Isolation

Heterogenous Cluster / App Stack

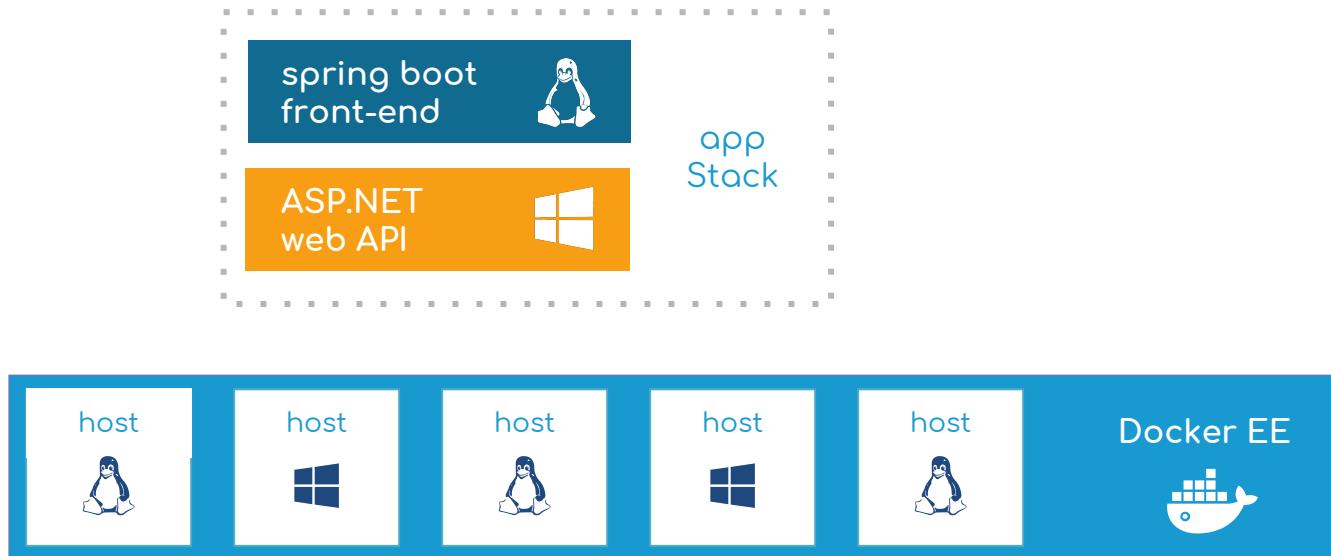


Image2Docker



Image2Docker

Discovery.
Extraction.
Provisioning.
Generation.

PHP

ASP .NET

My SQL

.NET

Apache

IIS

Ubuntu

Windows



Image2Docker - Linux

```
make prepare
```

```
make build
```

```
make builtin-prep
```

```
sudo bin/v2c-darwin64 build -n img.vmdk
```

Image2Docker - Windows

```
Install-Module Image2Docker
```

```
Import-Module Image2Docker
```

```
ConvertTo-Dockerfile `  
-ImagePath c:\iis.vhd `  
-OutputPath c:\i2d2\iis `  
-Artifact IIS
```



MTA Expand



MTA Expand Readiness Checklist

Team Introductions

- MTA Team
- Client business sponsor
- Client subject matter experts
 - Application
 - DevOps
 - Infrastructure

Schedule and location

- Start / End dates & times
- Workspace logistics

Application Ready for Production

- 1 or 2 apps POC containerized
- Existing dev systems
- Existing CI/CD

Infrastructure Readiness

- Docker EE prod infrastructure
- MTA team access to infrastructure

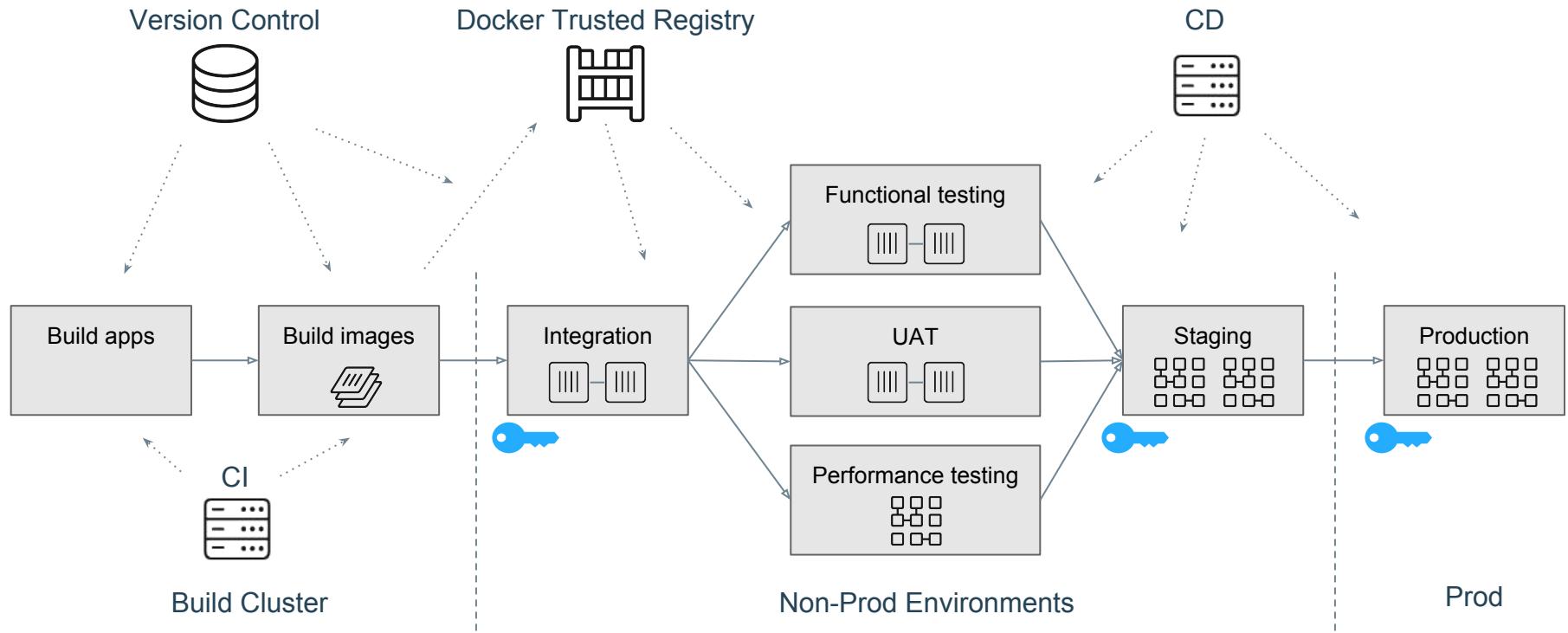
Operationalization in Docker EE

- Automated
 - Continuous Integration / Continuous Deployment
- Secure
 - Hardened application containers
 - Multi-tenant isolation (Docker EE RBAC)
 - Promotion policies (Image signing and scanning)
- Reproducible
 - Single source of truth version control (source code and infrastructure)
- Scalable
 - Enterprise-wide base images
 - Onboarding tools and resources
- Updateable
 - Any app version deployed to any environment (dev, test, staging, prod) fast

Operationalize One Application

Activity	Description	Client Participant/Shadow
Kickoff Review	Set expectations for scope, resources and schedule with team	Bus. Sponsor + SMEs
App Architecture Review	<ul style="list-style-type: none">Understand various app and infra characteristics, validate suitability for industrialization. Decide which Docker EE features to adopt (HRM, Secrets, networks, volumes, etc.)Understand production deployment pattern (big bang, blue/green, canary)	DevOps & App SME
Operationalize App Components	<ul style="list-style-type: none">Operationalize Docker image for each major app component (manual and/or conversion tools)Create Docker Compose files for multiple envs (dev, test, staging, prod)Version control specific configurations for multiple envsValidate application stack runs as expected for at least two envs	App SME
Operationalize App Pipeline	<ul style="list-style-type: none">Create app build process in CI/CDAutomate app deployment for multiple envs in CI/CDAutomate app testing in CI/CDAutomate image promotion in DTR and CI/CD	App SME
App Validation and Testing	<ul style="list-style-type: none">Validate end-to-end deployment processRun client functional test plan	DevOps & App SME
Checkpoints / Sign-off	Meetings to report status, issues, success and next steps	Bus. Owner 

DevOps Pipeline



Docker Universal Control Plane



Operationalize DevOps Pipeline

Activity	Description	Client Participant/Shadow
Kickoff Review	Set expectations for scope, resources and schedule with team	Business Sponsor + SMEs
DevOps Pipeline Review	<ul style="list-style-type: none">Understand various systems and CI/CD characteristics, validate suitability for integration (Version Control, Package Repository, Repository Manager)Enterprise-wide storage, networking, HA requirements review	DevOps & App SME
Operationalize Pipeline Components	<ul style="list-style-type: none">Create isolated build clusterCreate enterprise-wide base images (app stacks, CI, tools)Create enterprise-wide RBAC policies for Docker EECreate enterprise-wide security policies (Content Trust, Scanning)	DevOps SME
Operationalize App Pipeline	<ul style="list-style-type: none">Integrate industrialized app to existing CI/CD systemValidate application stack automated for at least two envs	DevOps SME
Pipeline Validation and Testing	<ul style="list-style-type: none">Validate end-to-end deployment processRun rollback and disaster recovery plan	DevOps & App SME
Checkpoints / Sign-off	Meetings to report status, issues, success and next steps	Business Owner

Review Onboarding Resources

Activity	Description	Client Participant/Shadow
Kickoff Review	Set expectations for scope, resources and schedule with team	Business Sponsor + SMEs
Existing Resources Review	Understand various resource characteristics	Business Sponsor + SMEs
Resources Best Practices	<ul style="list-style-type: none">Internal Marcom for Docker EE service rolloutOnboarding content (howtos, best practices, sample applications)Onboarding tools (wiki, forums, hotline, chat, mailing lists, ticketing, etc)<u>Goto Dev Readiness checklist</u><u>Goto Prod Readiness checklist</u>Internal Docker expertise groupTraining (Internal and Docker Inc.)Support (Internal and Docker Inc.)	Business Sponsor + SMEs
Checkpoints / Sign-off	Meetings to report status, issues, success and next steps	Business Owner