AST
AGENCY FOR STATE TECHNOLOGY

Docker Presentation for the State of Florida

christine.lovett@docker.com

al.barnes@docker.com

July 26, 2018

Agenda:

- Overview of Docker
- Candidates for Containerization
- Dockerfile Best Practices
- Cluster Architecture and Orchestration
- Building Secure Software Supply Chains
- Live Demo

The Docker Enterprise Container Platform
Enabling the Software Supply Chain

- Diverse Applications
- Disparate Infrastructure
- Lifecycle Management
- Orchestrate Complex Systems
- Secure by Default
- Edge / IoT
- Serverless Anywhere

Docker EE is built to enable these core objectives

Choice

- Hybrid and multi-clouds
- Windows and Linux
- Traditional apps and microservices
- DevOps and existing ops processes

Agility

- Unified operations
- Rapid delivery and response
- Cost efficiency

Security

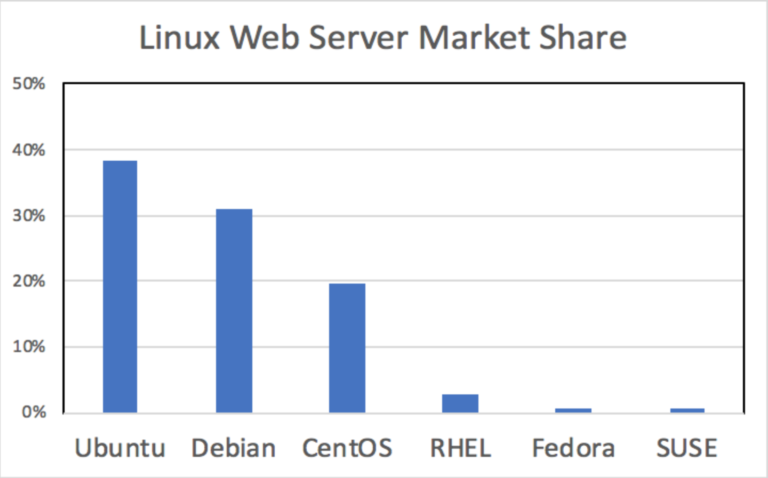
- Safer apps
- Governance
- Chain of custody
- Threat mitigation

CHOICE

Only Container Platform that is Multi-Linux, Multi-OS and Multi-Cloud

81% companies with 1,000+ employees have multiple clouds

Source: Rightscale 2018 State of the Cloud Report

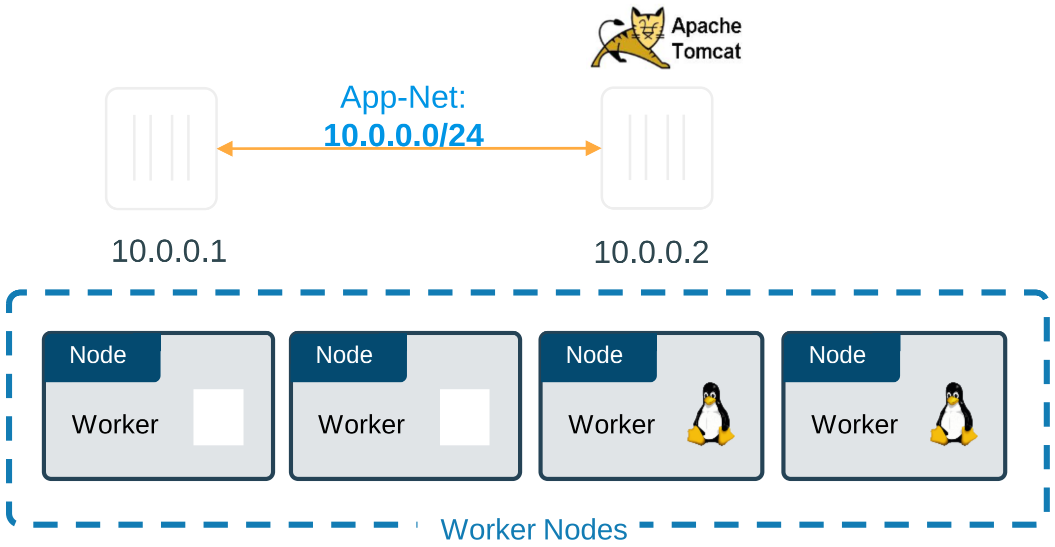


Source: https://w3techs.com/technologies/details/os-linux/all/all

Docker Enterprise Edition is certified to run on CentOS, RHEL, Ubuntu, SUSE, Oracle Linux and Windows Server and can be deployed into all major public clouds while maintaining the same operating experience

CHOICE

Only Container Platform to Deliver First-Class Support and Interoperability across Linux and Windows

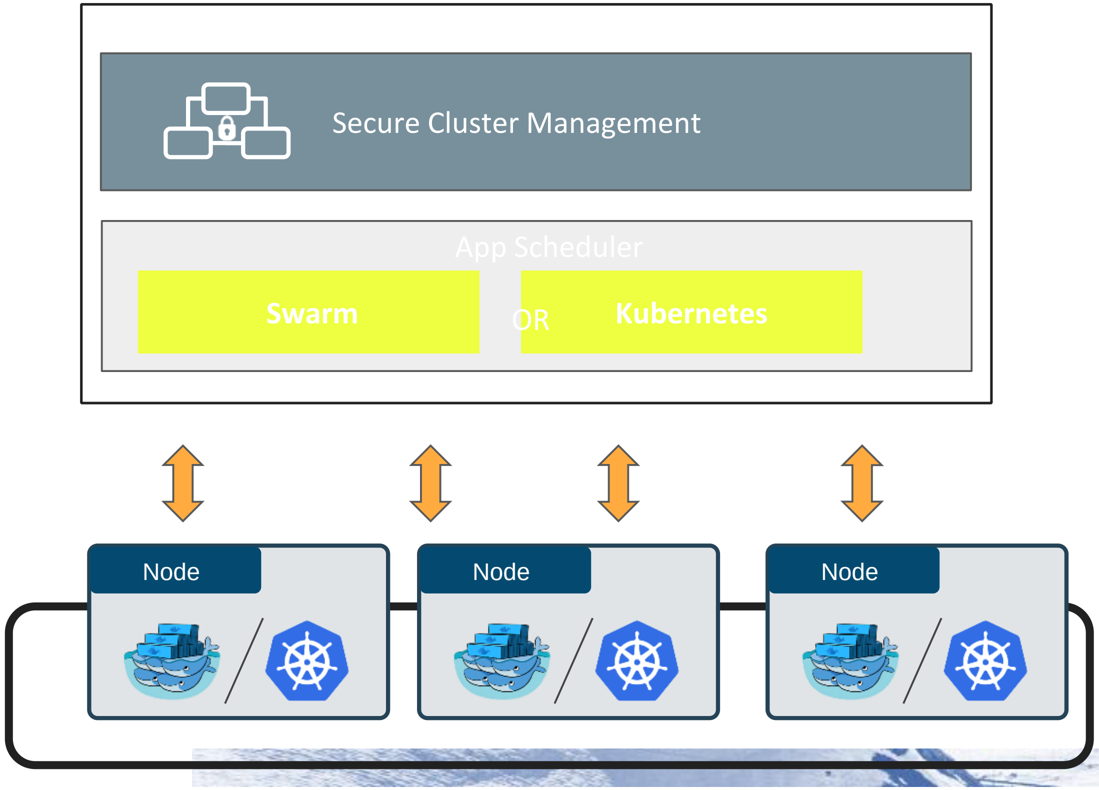


- Leverage best-in-class technologies across Windows and Linux
- Connect Windows and Linux containers in the same cluster through a common overlay network
- Build Compose files for hybrid applications
- Leverage labels and constraints for intelligent placement and scheduling

CHOICE

Choice of Swarm and Kubernetes: Only Solution That Lets You Run Swarm Today, Kubernetes Tomorrow and Vice Versa

Docker EE Orchestration

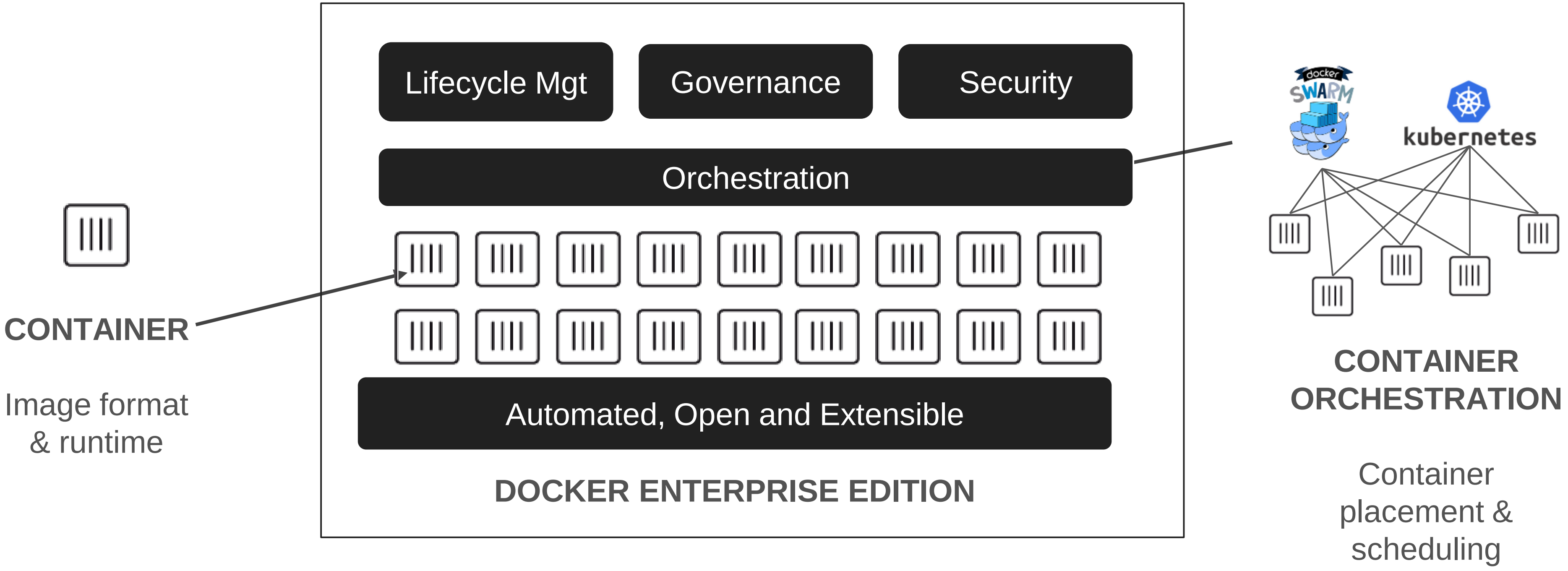


Docker EE Cluster

Docker EE is the only platform that allows you to run both Swarm and Kubernetes in the same cluster:

- Developers do not need to select orchestrators
- Freedom to change orchestrators as needs arise
- EE Manager Nodes are both Swarm and Kubernetes enabled
- Every worker node is both Kubernetes API- and Swarm API-ready

End-to-end container lifecycle



Lifecycle Mgt

Governance

Security

Orchestration

Automated, Open and Extensible

DOCKER ENTERPRISE EDITION

CONTAINER

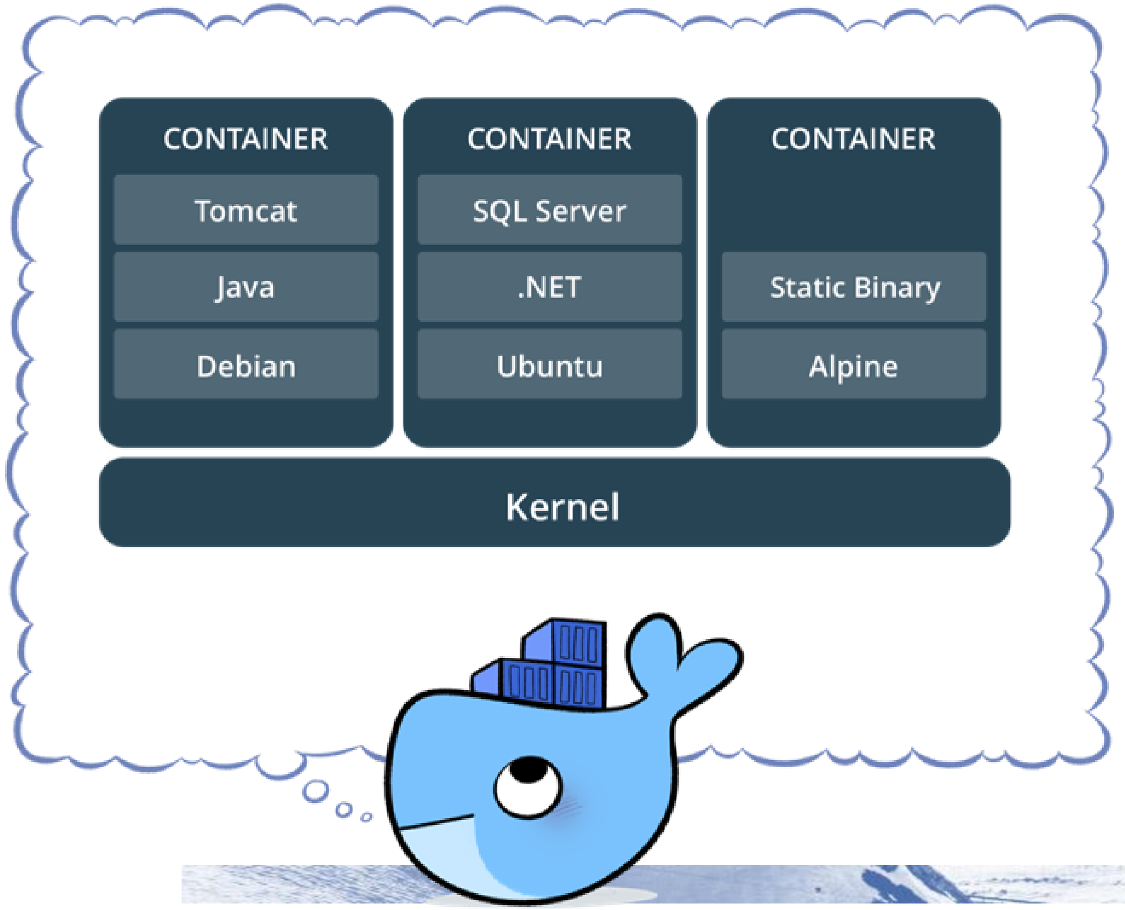
Image format
& runtime

CONTAINER ORCHESTRATION

Container
placement &
scheduling

Organizations also require Lifecycle Management + Governance + Security + Automation + Extensibility

What is a container?



- Standardized packaging for software and dependencies
- Isolate apps from each other
- Share the same OS kernel
- Works with all major Linux distros and Windows Server 2016

How to create a Docker container

Without Docker

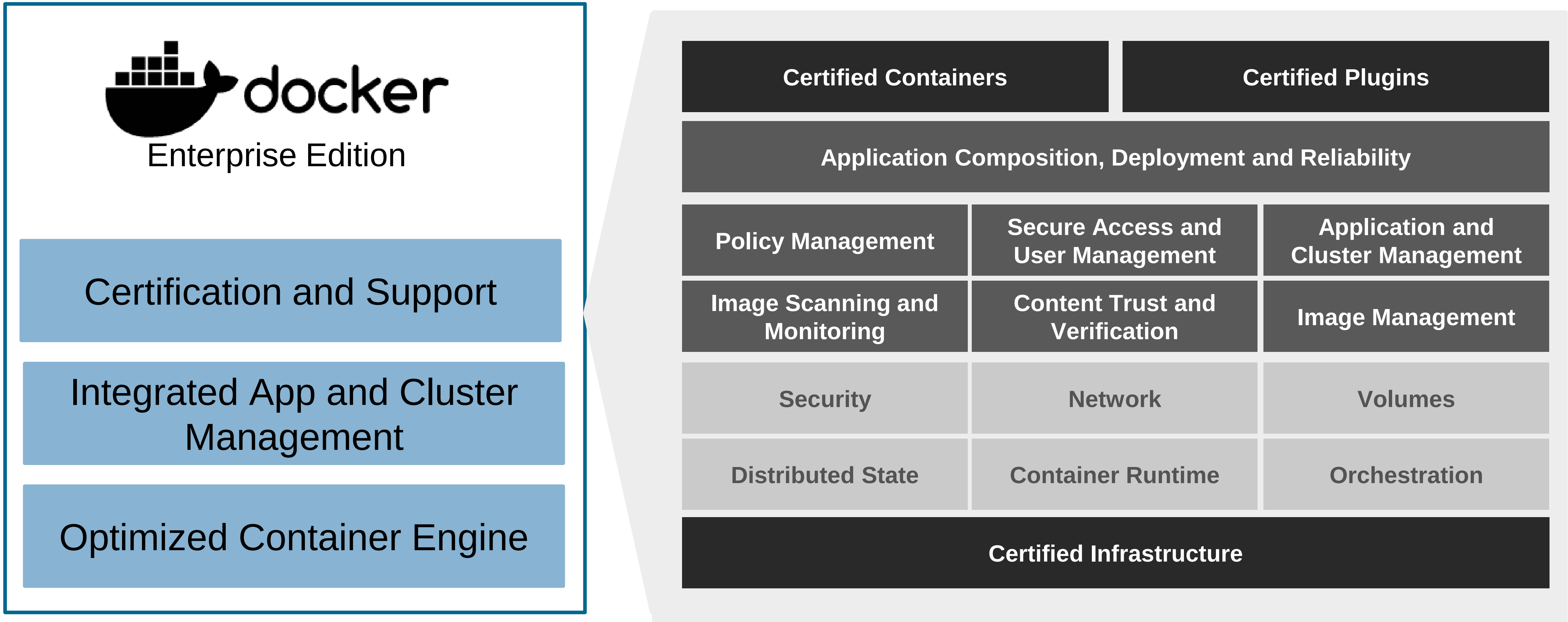
100 Page Binder

With Docker

Single Text File

- Replace the printed (often out of date) runbooks for app deployment and ops documentation
- Define instructions including: ports, volumes, environment variables, health-checks, and more

- Dockerfile contains all commands to assemble a Docker container
- Dockerfile containing all the instructions to deploy your app.
- Enables consistent deployments across multiple environments



docker

Enterprise Edition

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

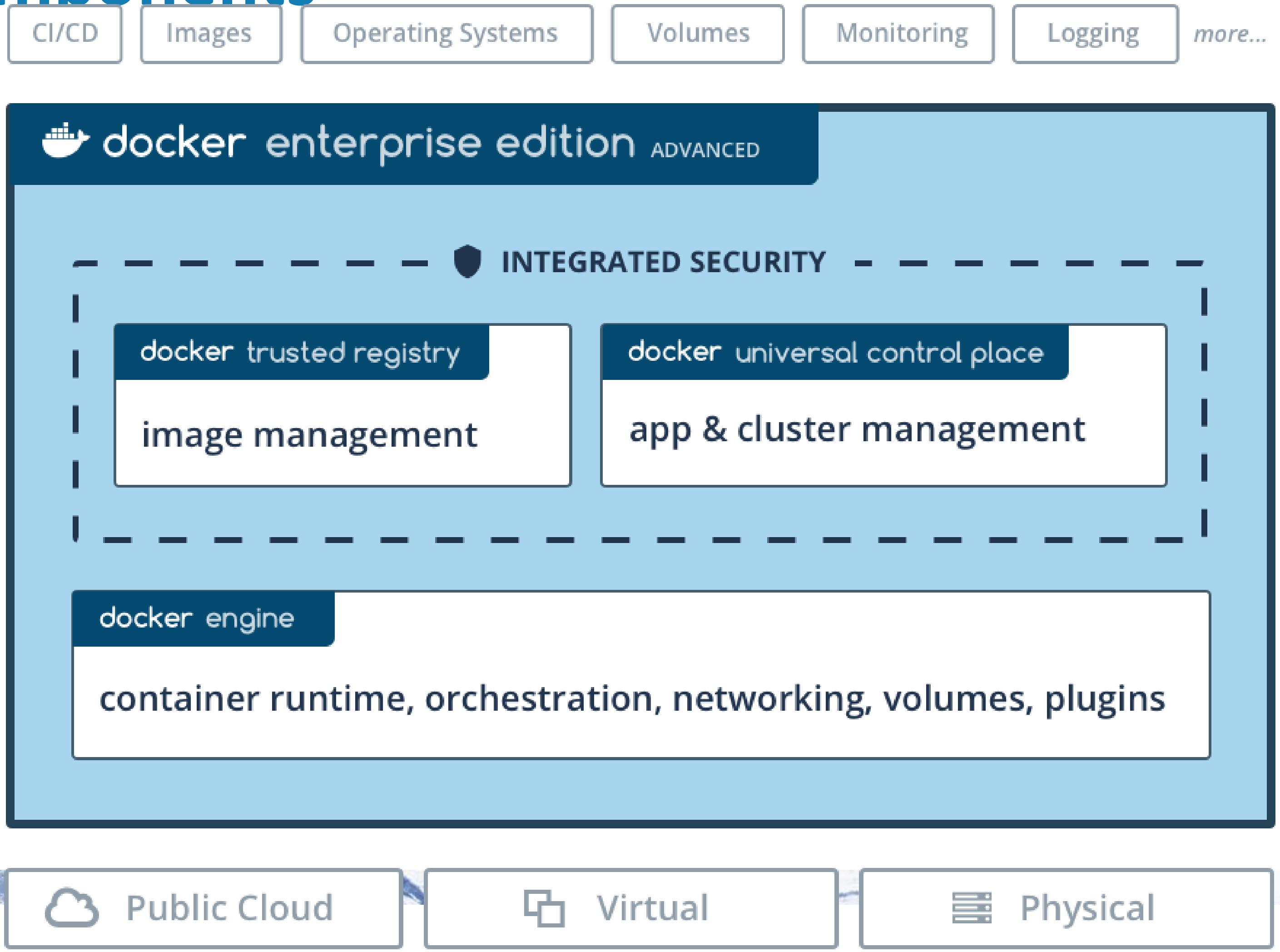
Image Scanning and Monitoring | Content Trust and Verification | Image Management

Security | Network | Volumes

Distributed State | Container Runtime | Orchestration

Certified Infrastructure

Docker EE Components



CI/CD | Images | Operating Systems | Volumes | Monitoring | Logging | *more...*

docker enterprise edition ADVANCED

INTEGRATED SECURITY

docker trusted registry

image management

docker universal control place

app & cluster management

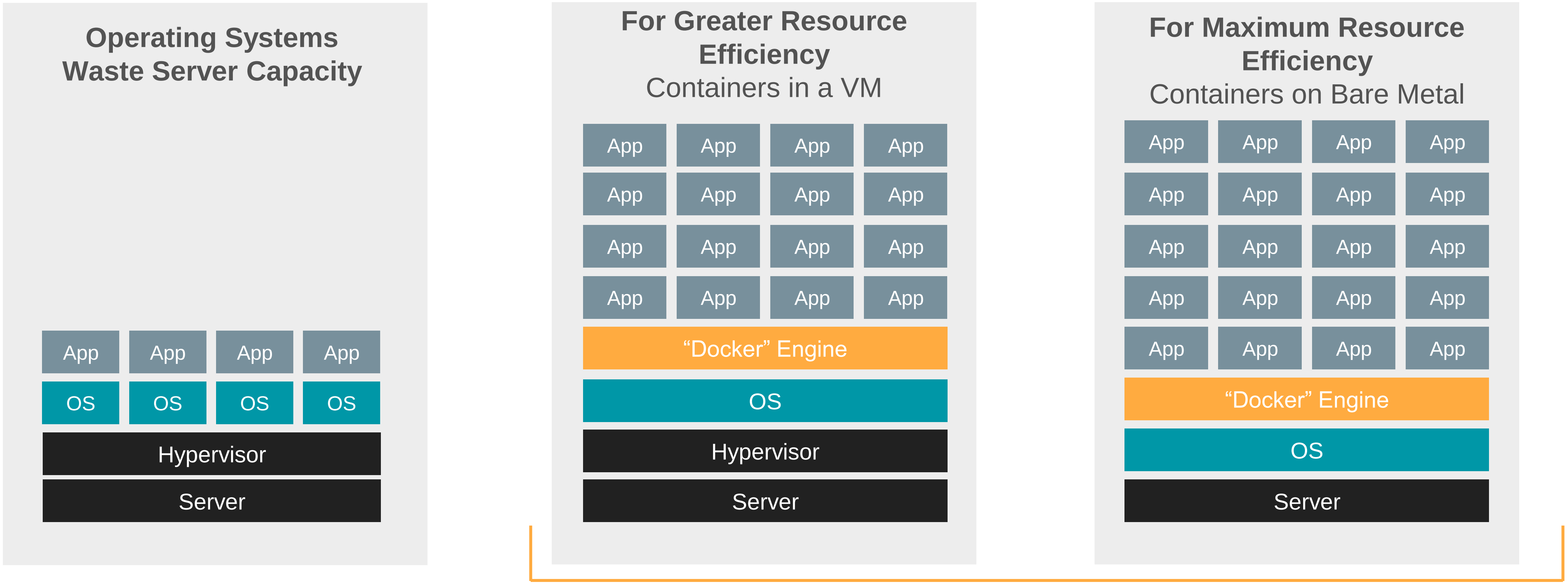
docker engine

container runtime, orchestration, networking, volumes, plugins

Public Cloud | Virtual | Physical

Why containerize applications?

Docker containers



Operating Systems Waste Server Capacity

App App App App

OS OS OS OS

Hypervisor

Server

For Greater Resource Efficiency
Containers in a VM

App App App App

App App App App

App App App App

App App App App

“Docker” Engine

OS

Hypervisor

Server

For Maximum Resource Efficiency
Containers on Bare Metal

App App App App

App App App App

App App App App

App App App App

App App App App

“Docker” Engine

OS

Server

Server Count ↓

VM and OS Licenses ↓

Power, Space, Cooling ↓

Admin ↓

Candidates for containerization

App Selection

General Guidance

Linux	Windows
Java EE	.NET Framework
Components fundamentally compatible with modern OS (e.g. RHEL 7, WS 2016, etc.)	
Server-side only components (no GUI)	
Requires little / no refactoring	

App Selection

Runtime / App Servers

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

App Selection

Architecture

Linux	Windows
2 or 3 tier app	
Up to 5 application components [e.g. service, job, queue, proxy, etc.]	
Components do not require clustering (can run as a single node)	
Databases can be excluded for containerization (i.e. can be migrated / connected to independently)	

Best practices

Building Efficient Images

1) Minimize Number of Layers

Bad!

```
RUN apt-get update
RUN apt-get install -y wget
RUN rm -rf
/var/lib/apt/lists/*
```

Good!

```
RUN apt-get update &&\
   apt-get install -y wget &&\
   rm -rf /var/lib/apt/lists/*
```

Building Efficient Images

2) Use cache-busting: ensuring you never use cache outdated updates

Bad!

```
RUN apt-get update
RUN apt-get install -y wget
RUN rm -rf
/var/lib/apt/lists/*
```

Good!

```
RUN apt-get update &&\
  apt-get install -y wget &&\
  rm -rf /var/lib/apt/lists/*
```

Building Efficient Images

3) Update by version

Good!

```
RUN apt-get update &&\
   apt-get install -y wget &&\
   rm -rf /var/lib/apt/lists/*
```

Better!

```
RUN apt-get update &&\
   apt-get install -y
wget=1.17.1-1ubuntu1.1 &&\
   rm -rf /var/lib/apt/lists/*
```

Building Efficient Images

4) Separate layers that will break the cache to utilize layer caching

BAD!

```
COPY . /usr/src
RUN npm install
```

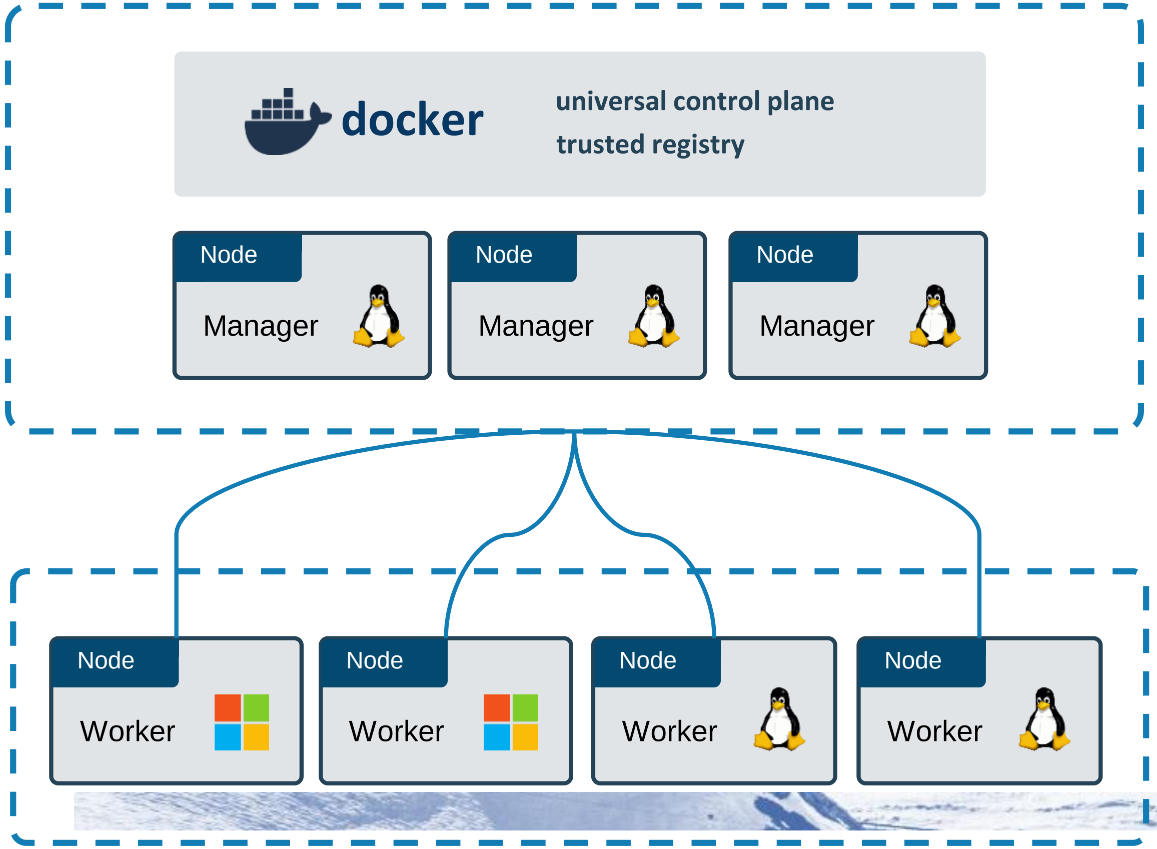
Good!

```
COPY package.json /usr/src/package.json
RUN npm install
COPY . /usr/src
```

Architecture

Mixed Windows and Linux Clusters

UNIFORMLY OPERATE, MANAGE, AND SECURE WINDOWS AND LINUX CONTAINERS



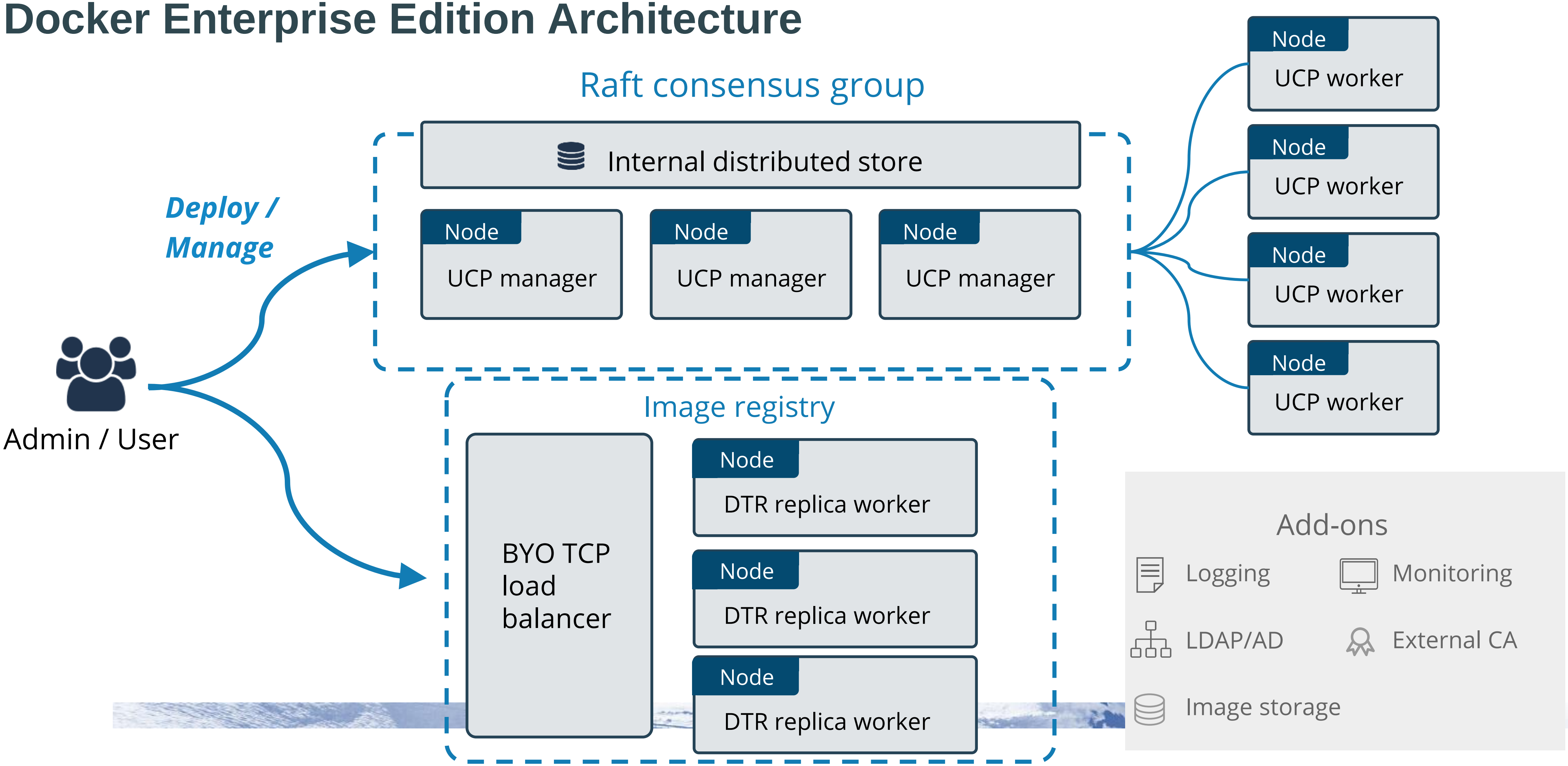
KEY FEATURES

- Extend enterprise security features like image signing, image scanning, and secrets management to both Windows and Linux worker nodes
- Leverage the same LDAP/AD integration and RBAC rules across Windows and Linux nodes
- Visualize all apps in the same environment

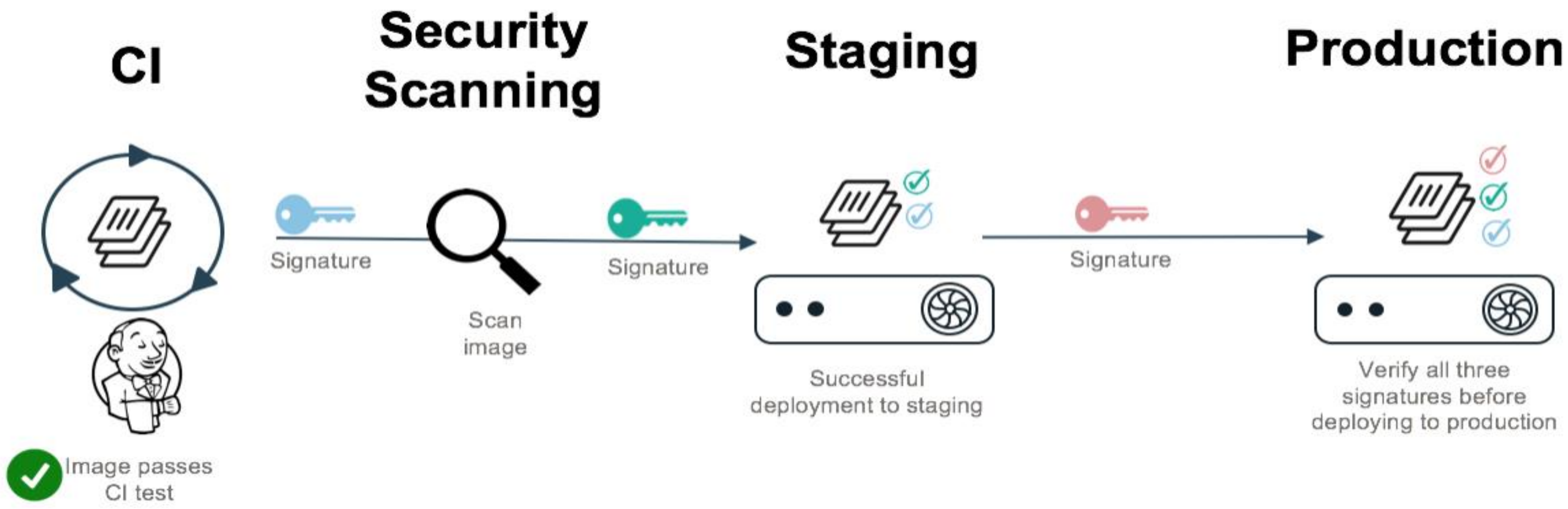
BENEFITS

- Improve resource utilization and incur less management overhead with centralized management across Windows and Linux apps
- Reduce risk with consistent processes and policies across Windows and Linux apps

Docker Enterprise Edition Architecture

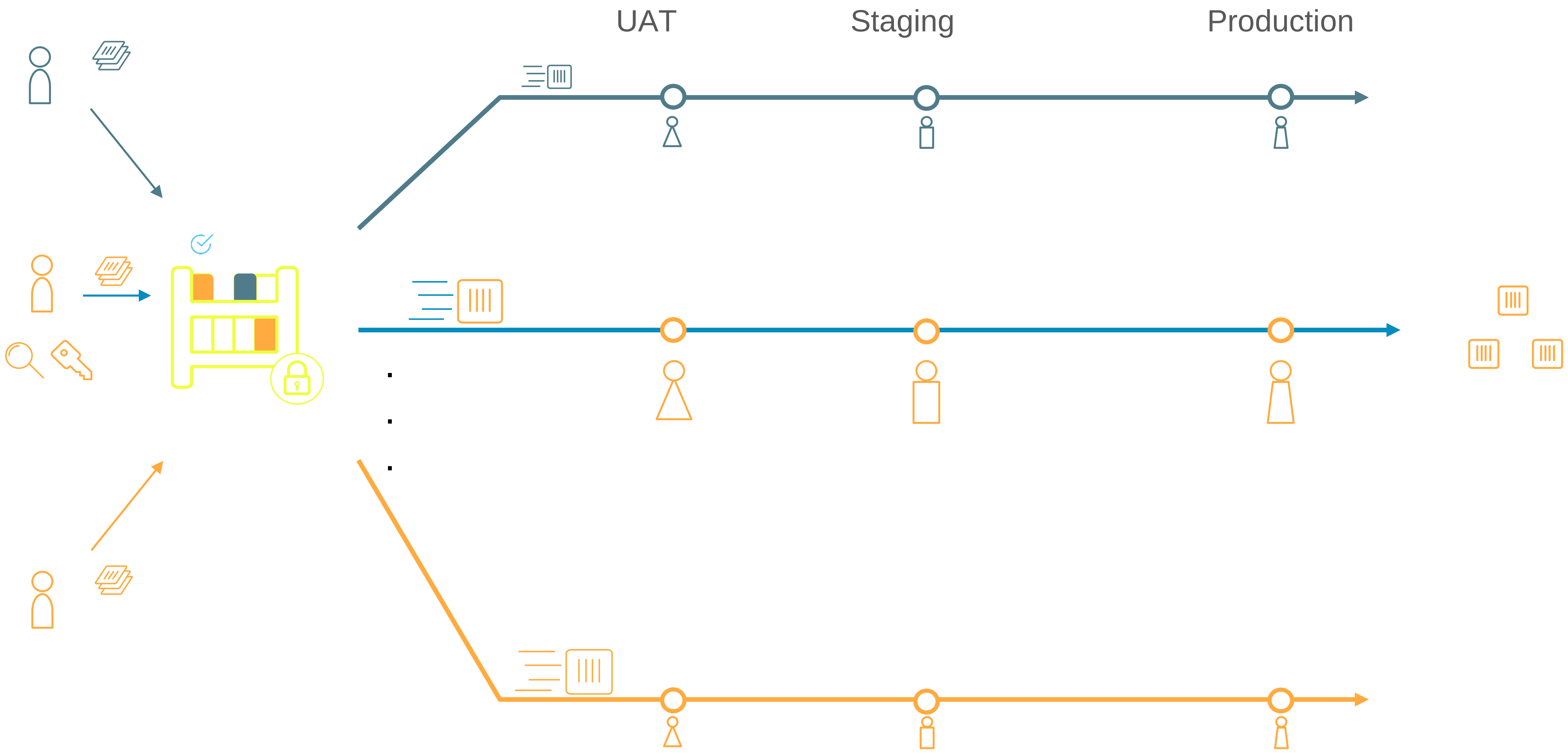


Leverage a secure and automated software supply chain

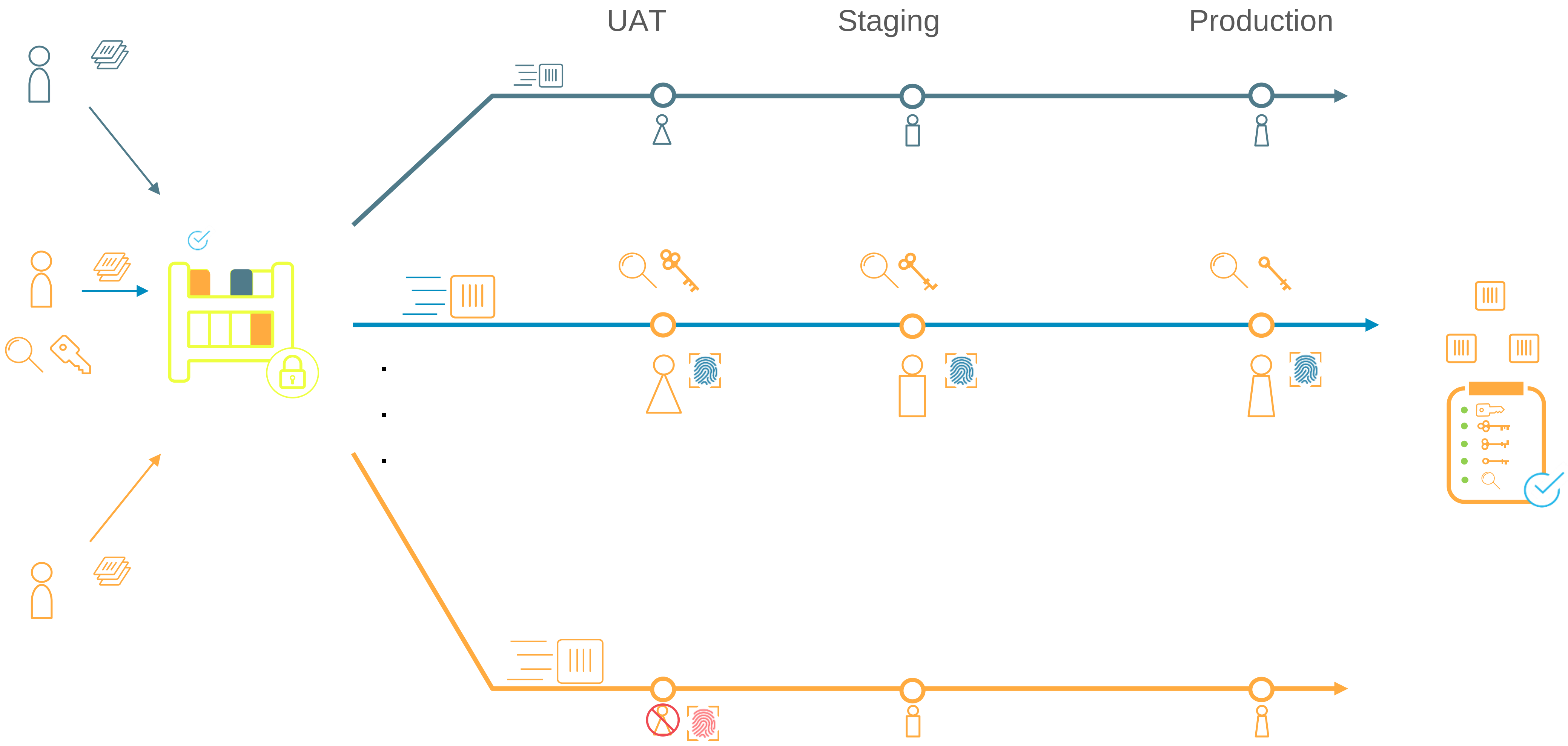


- Establish chain of trust with apps as they move across environments
- Digitally sign containers and only run verified containers
- Freshness guarantee ensures no tampering and latest container is running
- Automate workflow with immutable repos and automated image promotion

Secure Software Supply Chain



Secure Software Supply Chain



Docker EE Deployment Models - Site Redundancy

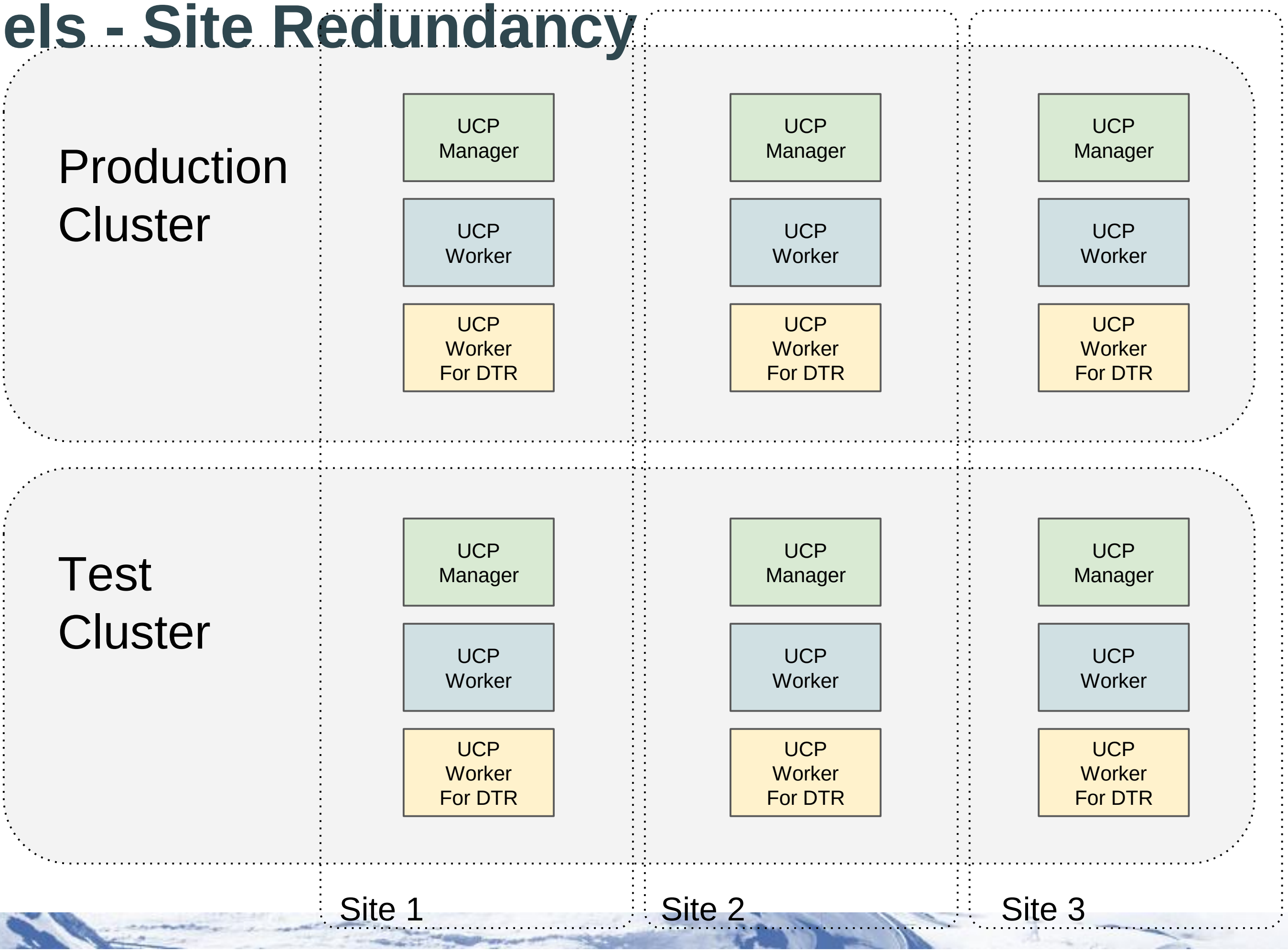
2 Clusters, 3 Sites

- We try and achieve redundancy by spreading a cluster across sites.

- With 3 Sites (And the addition of a 3rd worker node) we are able balance all of the components correctly.

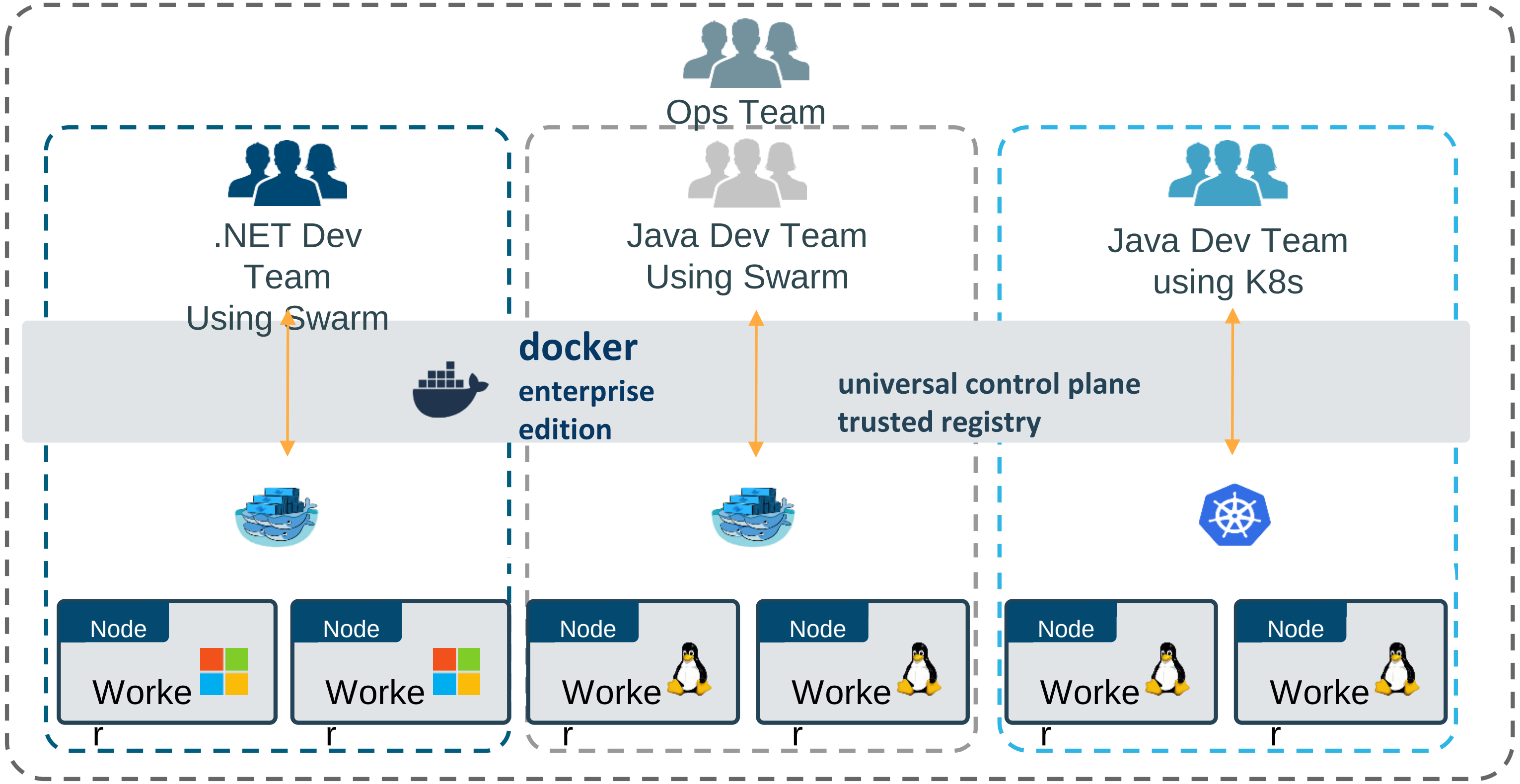
- If site 1 goes down. You lose ⅓ of your managers, therefore Quorum remains.

- 3 Sites does provide you with Site Redundancy!



	Site 1	Site 2	Site 3
Production Cluster	UCP Manager UCP Worker UCP Worker For DTR	UCP Manager UCP Worker UCP Worker For DTR	UCP Manager UCP Worker UCP Worker For DTR
Test Cluster	UCP Manager UCP Worker UCP Worker For DTR	UCP Manager UCP Worker UCP Worker For DTR	UCP Manager UCP Worker UCP Worker For DTR

Define Secure Application Zones to Enforce IT Governance



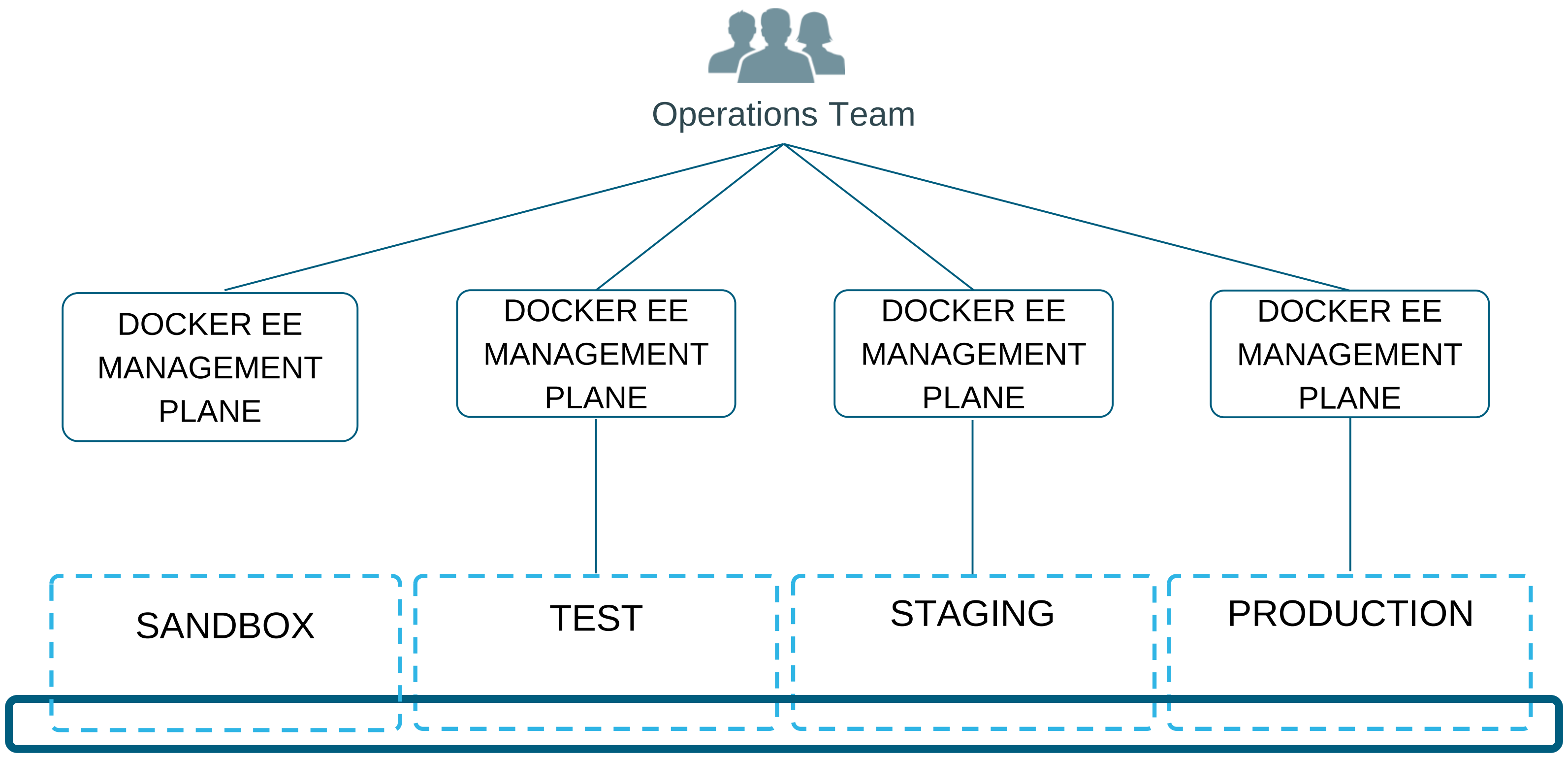
FEATURE / CAPABILITY

- Integrate with LDAP/AD and create granular and flexible access controls
- Combine Namespace isolation with node-based isolation for increased separation

KEY BENEFITS

- Easily define resource-based permissions to different teams and expose only the allotted resources to each team
- Re-allocate resources as needed

Why not use separate clusters instead of application zones?


Operations Team
DOCKER EE MANAGEMENT PLANE
DOCKER EE MANAGEMENT PLANE
DOCKER EE MANAGEMENT PLANE
DOCKER EE MANAGEMENT PLANE
SANDBOX
TEST
STAGING
PRODUCTION

Single cluster, multiple divided zones

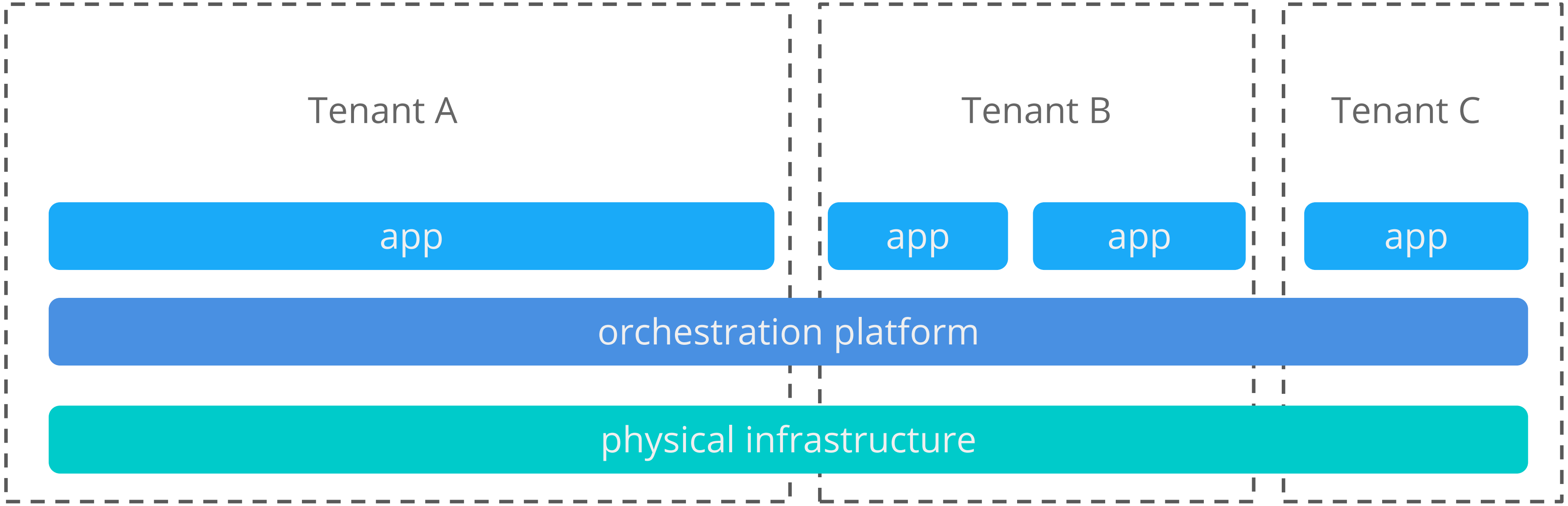
Cluster sprawl

- Redundant management
  - Higher cost (more nodes)
  - Disconnected policies & controls
  - Higher chance of errors

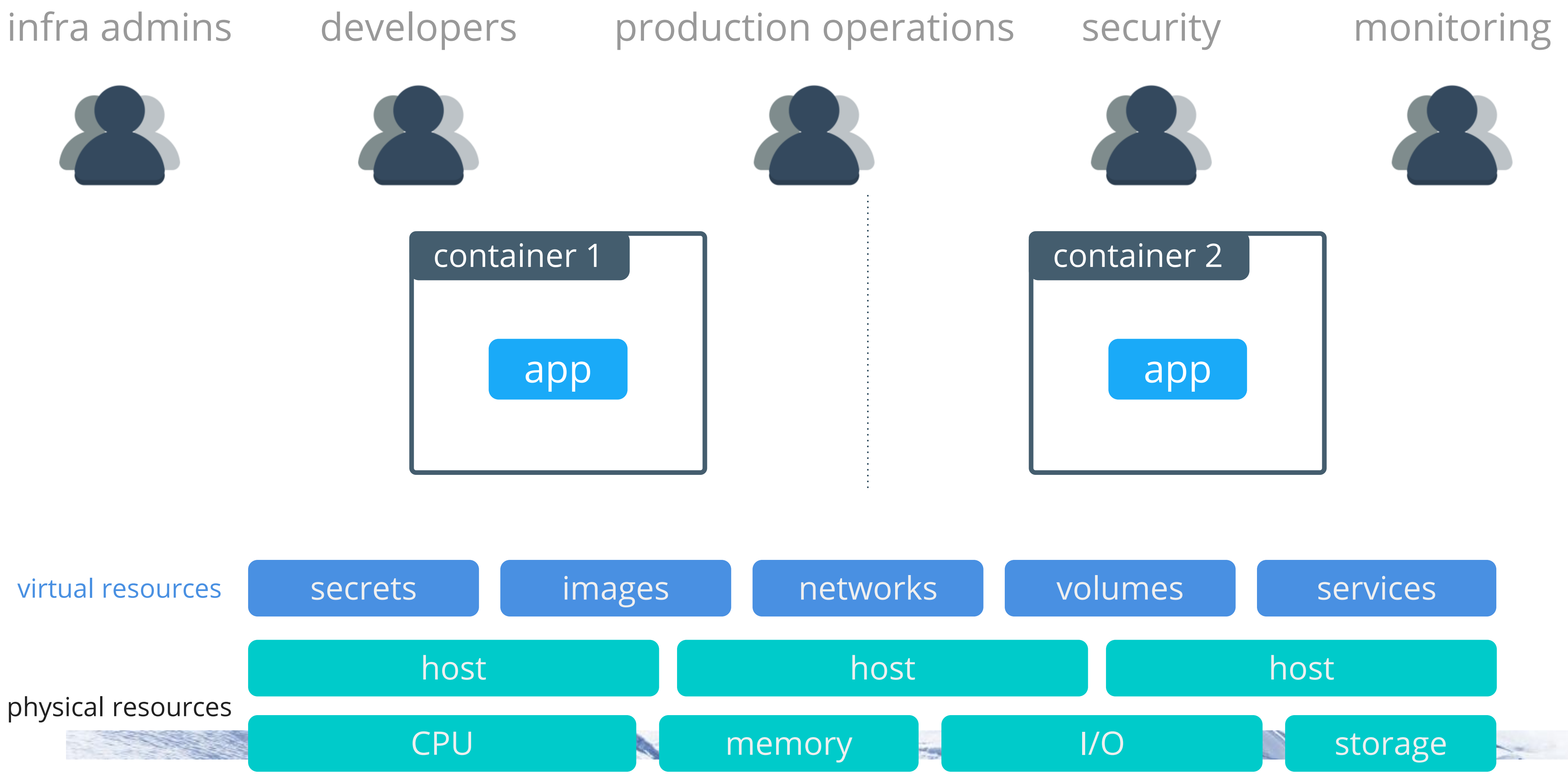
Define Secure Application Zones to Enforce IT Governance

Benefits of Secure Application Zones:

- Higher resource utilization
- Specialization of skill sets for discrete layers of IT stack
- Separation of responsibilities for more resilient and manageable architectures



Cluster Resources



infra admins

developers

production operations

security

monitoring

container 1

app

container 2

app

virtual resources

secrets

images

networks

volumes

services

physical resources

host

host

host

CPU

memory

I/O

storage

Docker Certified Infrastructure

Docker Certified Infrastructure:

Reliably Deploy and Manage Docker Enterprise Edition On Your Infrastructure

Reference Architecture

Detailed design and deployment considerations

Best practices developed in real world deployments

Automation & Tooling

Templates and scripts to automate deployment

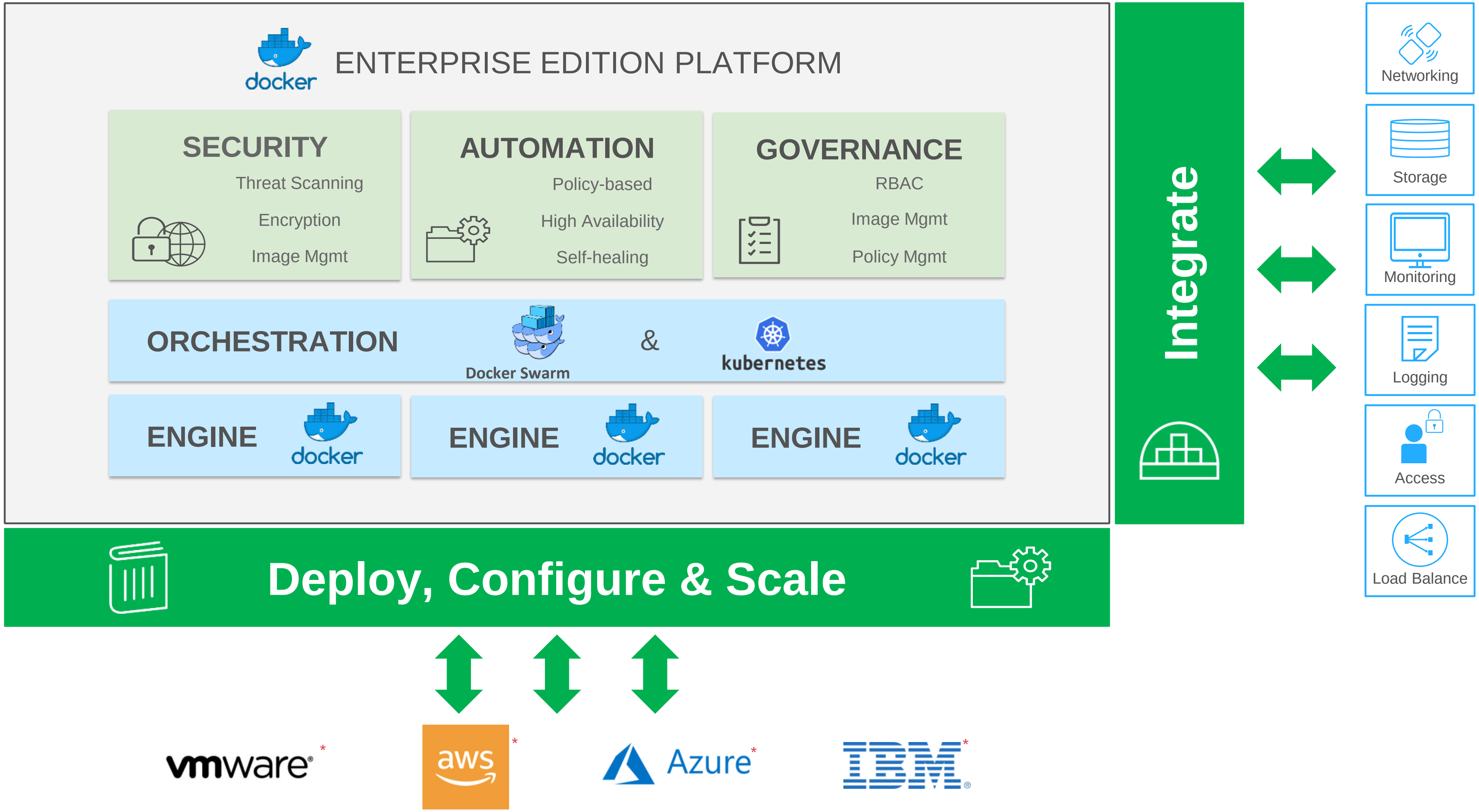
Spend more time on your apps

Ecosystem Integration

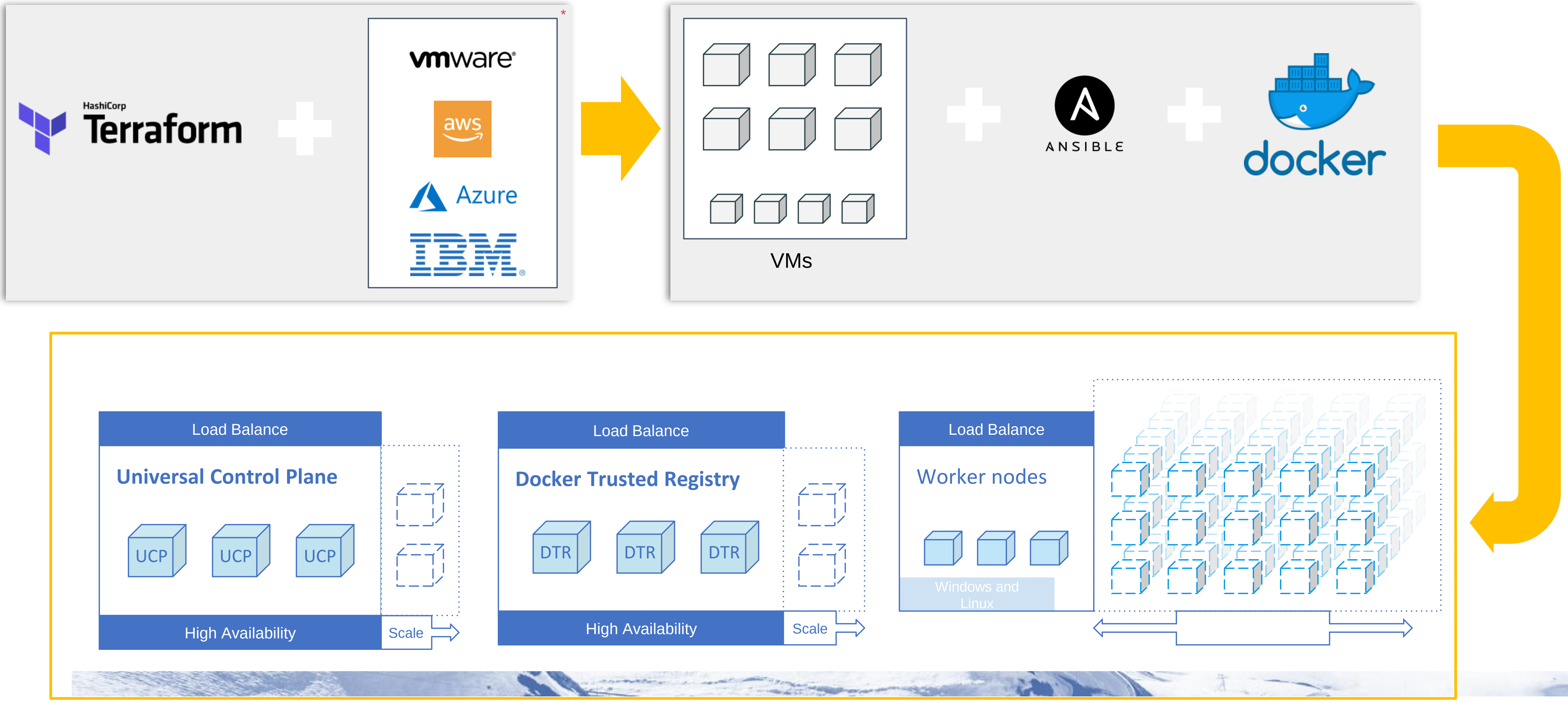
Open, extensible work with popular 3rd party tools

Integrate with what you have

What Does *Docker Certified Infrastructure* Do?



Docker Certified Infrastructure Technical Details



Live Demo

Questions?

Christine Lovett |
Solutions Engineer
christine.lovett@docker.com

Al Barnes |
Account Executive
al.barnes@docker.com