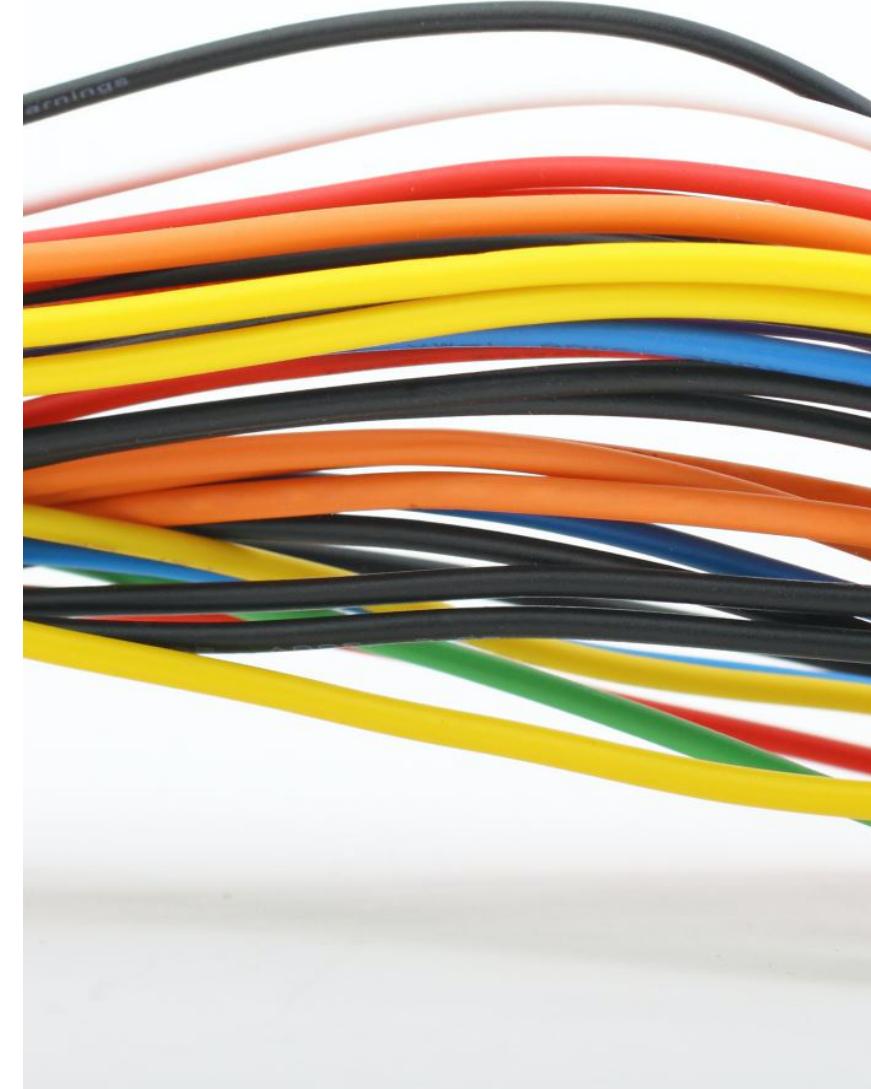


KUBERNETES 101 FOR JAVA DEVELOPERS

Heba Elayoty



WHO IS HEBA!

- A software engineer @Microsoft
- Long-term Java Developer
- Cloud native developer and K8s community member

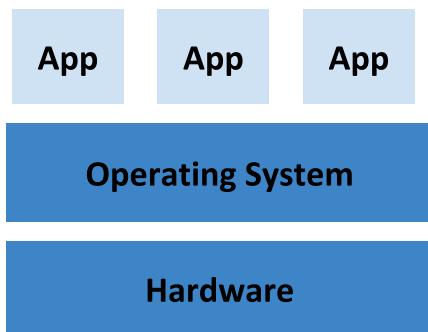


AGENDA

- Why do we need to run Java apps in containers?
- **Containerization:** Introducing Dockerfiles
- **Orchestration:** Introducing Kubernetes
- Connecting a Java (Spring Boot) app to cloud services
- Best practices for Java apps on Kubernetes
- **Cloud Native Fx:** Introducing Quarkus
- Quarkus vs Spring-boot
- Questions

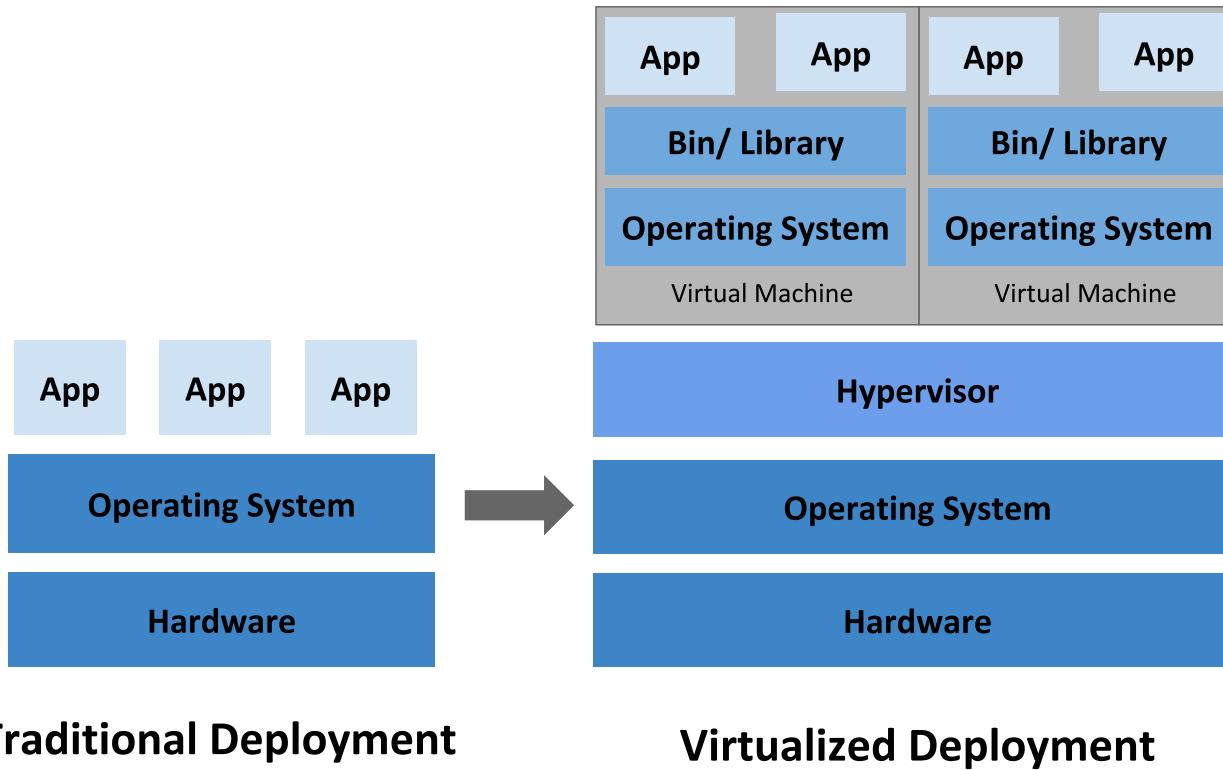


WHY DO WE NEED CONTAINERS

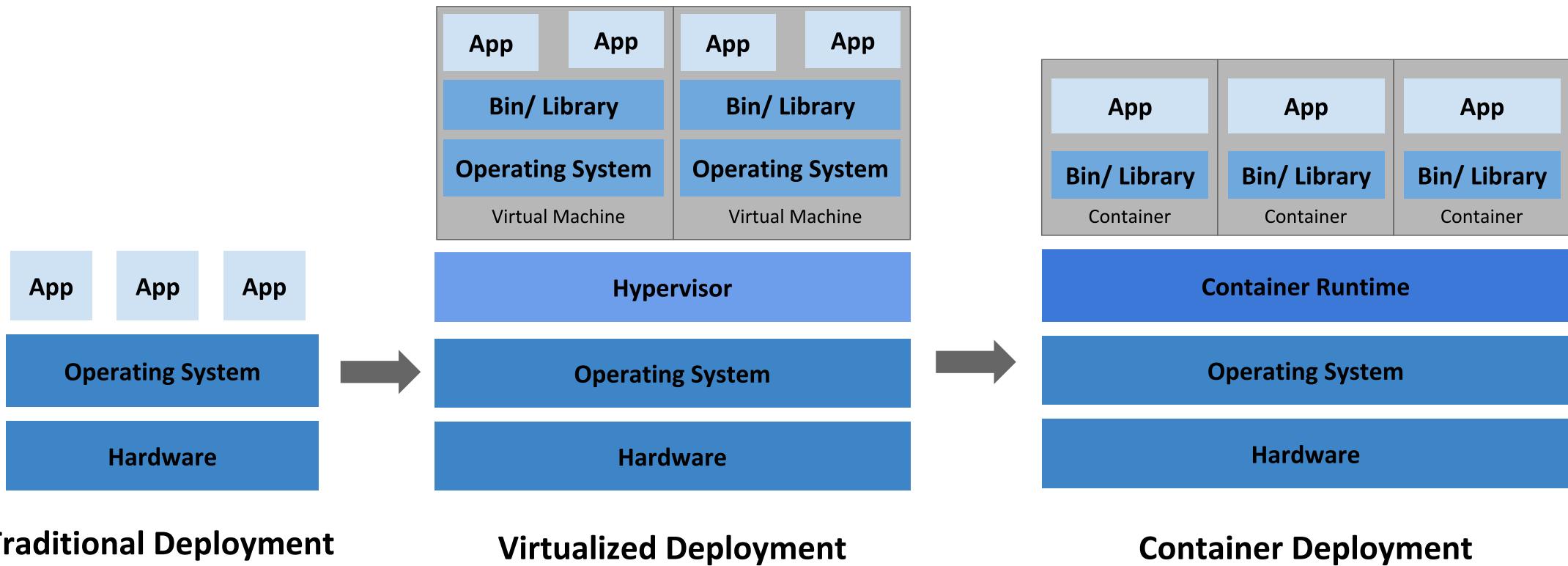


Traditional Deployment

WHY DO WE NEED CONTAINERS



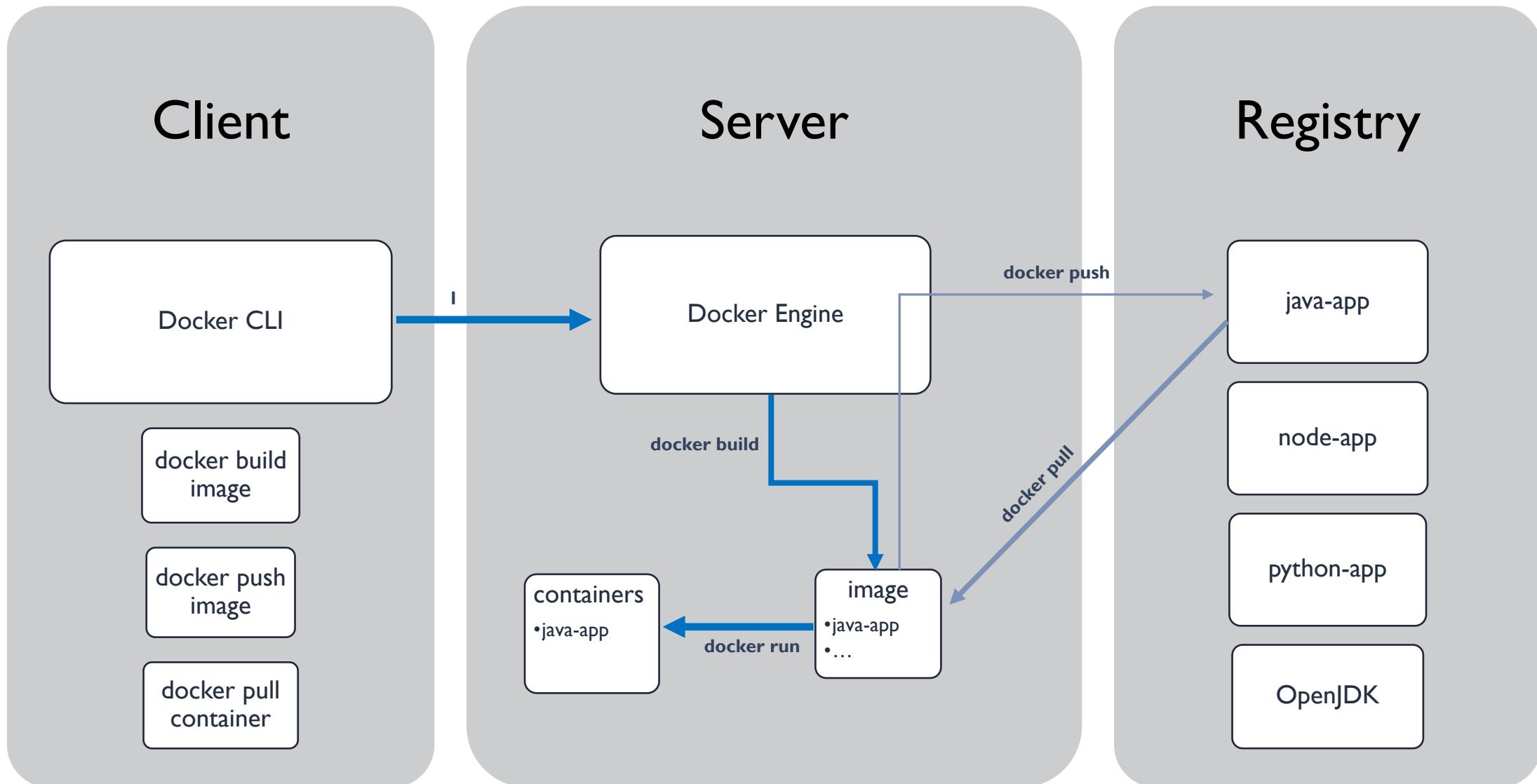
WHY DO WE NEED CONTAINERS

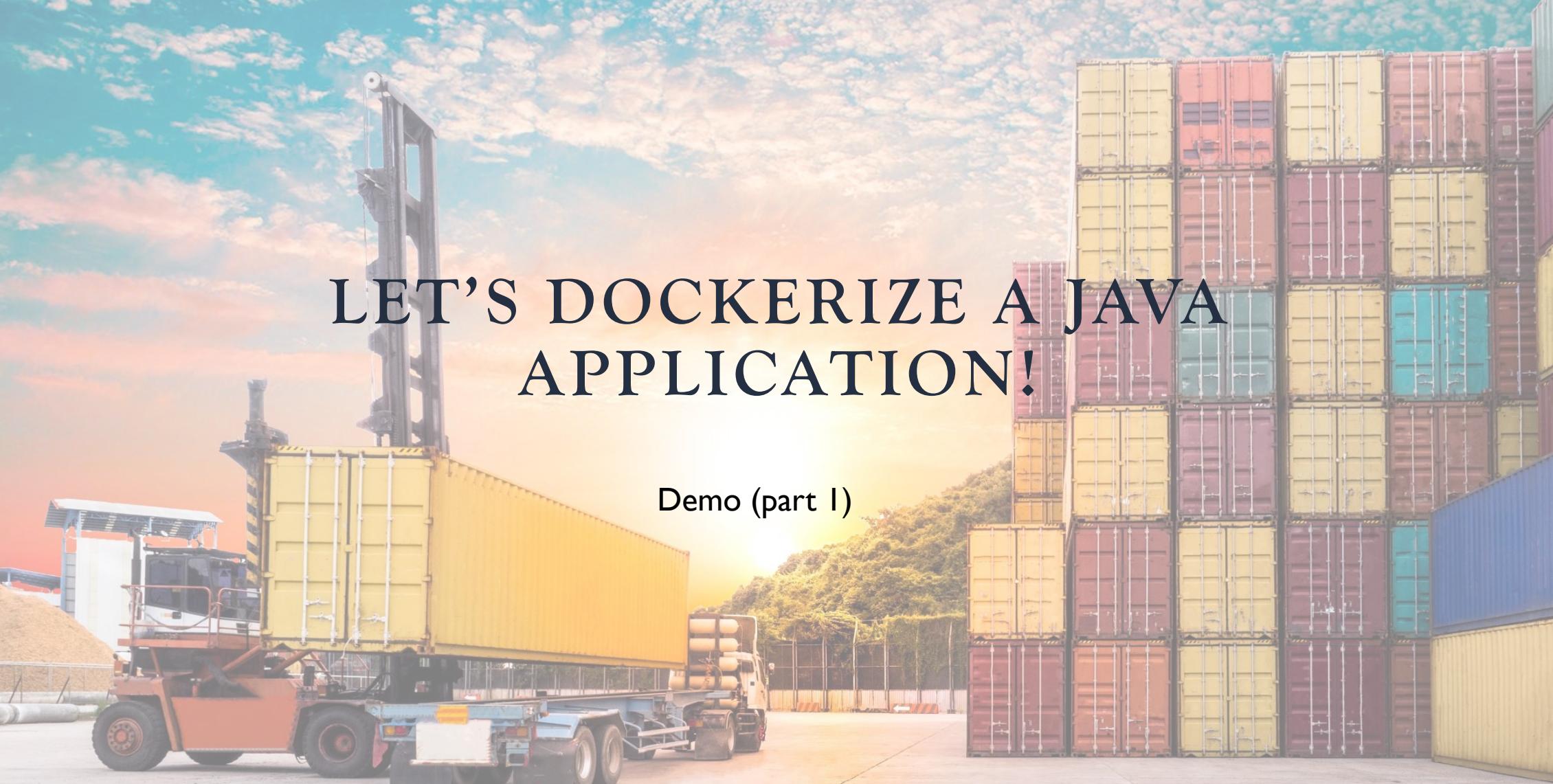


CONTAINERS VS DOCKER

- Container technology is not new
- Open Container Initiative (OCI)
- Docker is one of many container tools:
 - LXC
 - rkt
 - Podman (container engine)
 - Containerd (daemon)
 - runC (container runtime)
 - HyperV containers

DOCKER WORKFLOW





LET'S DOCKERIZE A JAVA APPLICATION!

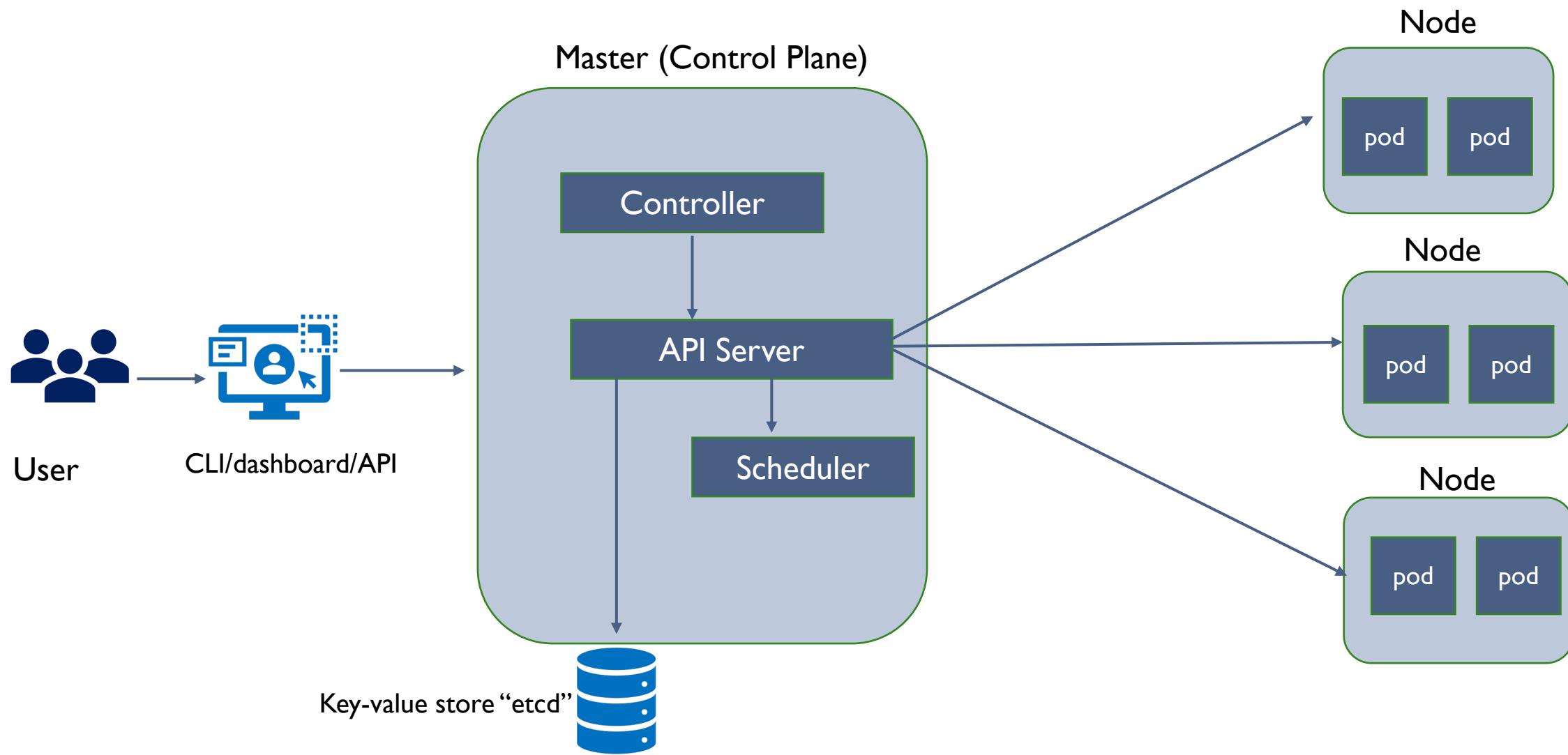
Demo (part I)

KUBERNETES 101

- Originates from Greek, meaning helmsman or pilot
- An open-source project since 2014
- What is Kubernetes?
- Why do we need Kubernetes?



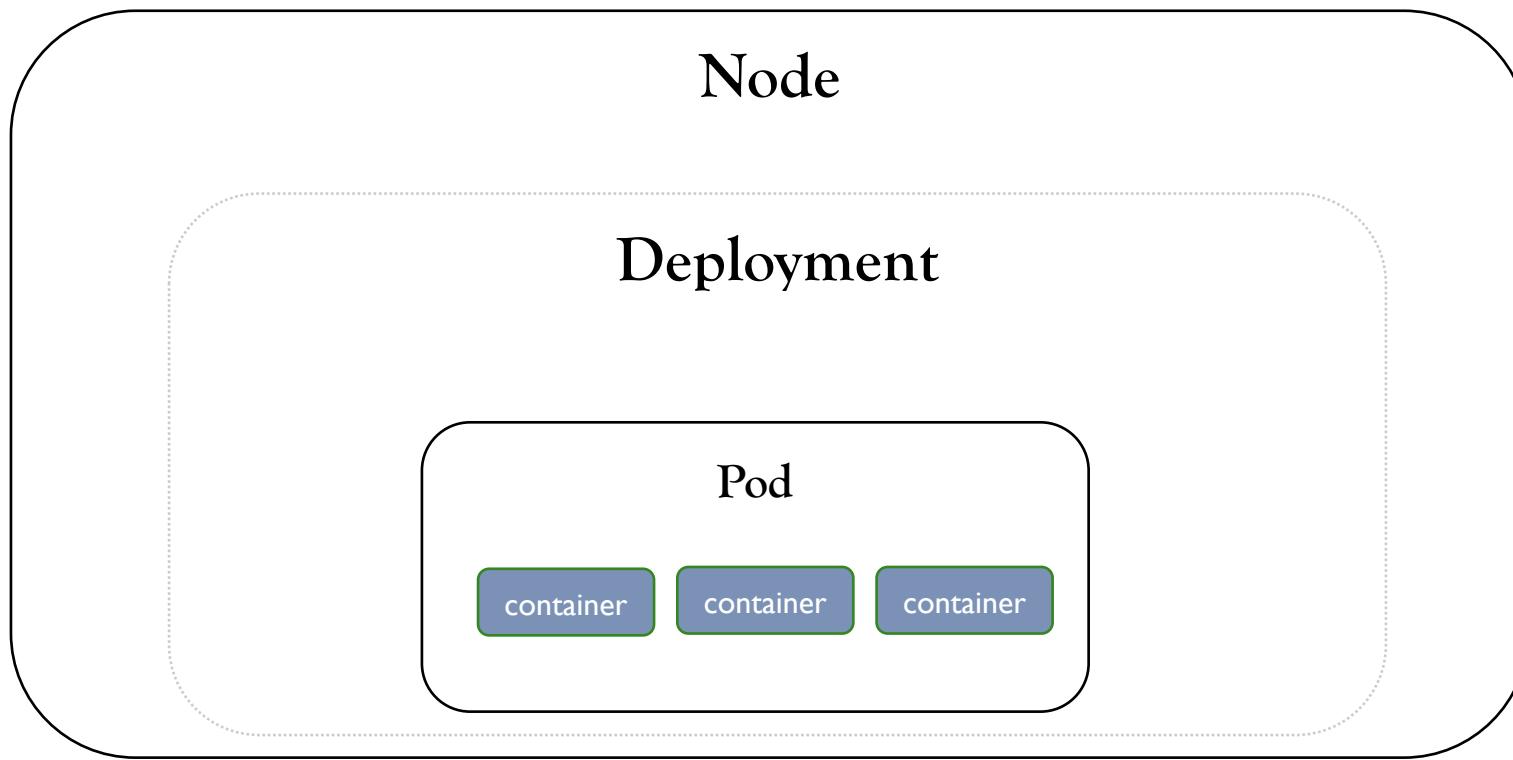
KUBERNETES ARCHITECTURE



CONTROL PLANE/MASTER COMPONENTS

- Controller
 - Loop that watches the state of your cluster and makes changes as needed, always working to maintain your desired state.
- API server
 - Exposes the Kubernetes API. It is the front-end for the Kubernetes control plane.
- Scheduler
 - Watches for newly created Pods with no assigned node and selects a node for them to run on.
- Key-Value store (etcd)
 - Consistent and highly-available key value store used as Kubernetes' backing store for all cluster data.

KUBERNETES CONCEPTS



DEPLOY A JAVA APPLICATION ON LOCAL K8S & AKS

Demo (part 2)

JAVA AND LINUX CONTAINERS

Challenges

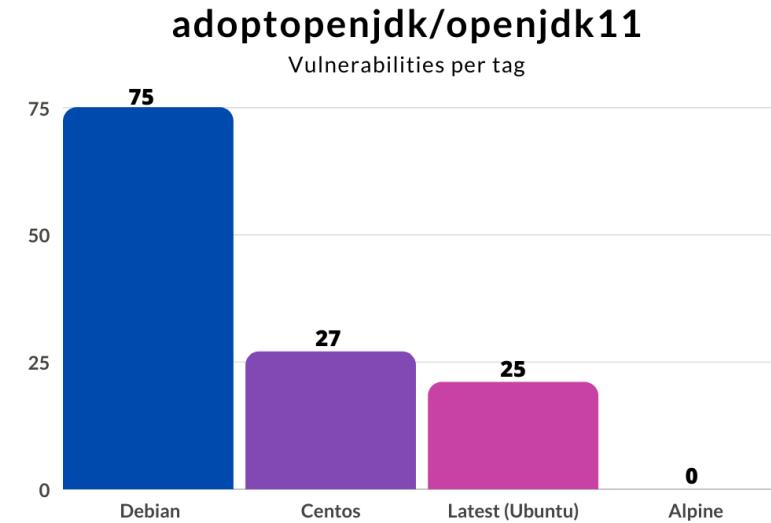
- Container Memory Limits
- Setting Available CPUs

Solution...

- ✓ Use Java 10+
- ✓ If you're not able to upgrade your Java version set your own limits using `-Xmx`.
- ✓ For Java 8 and Java 9, update to the latest version and use:
`-XX:+UnlockExperimentalVMOptions -XX:+UseCGroupMemoryLimitForHeap`

BEST PRACTICES

- Choose the right Docker base image for your Java application
- Scan your Docker images frequently
 - Clair
 - Anchore
 - Aqua Security
 - Docker Scan
- Use JRE images, not JDK



BEST PRACTICES

- Don't run your Docker container as root
- Set CPU requests & limits
- Set JVM Heap (-Xms & -Xmx)
- Set memory requests
- Health Check /Implement readiness and liveness probes
- Monitoring /Logging
- Consider using a cloud native Java framework (i.e., Quarkus, Micronaut)

INTRODUCING QUARKUS

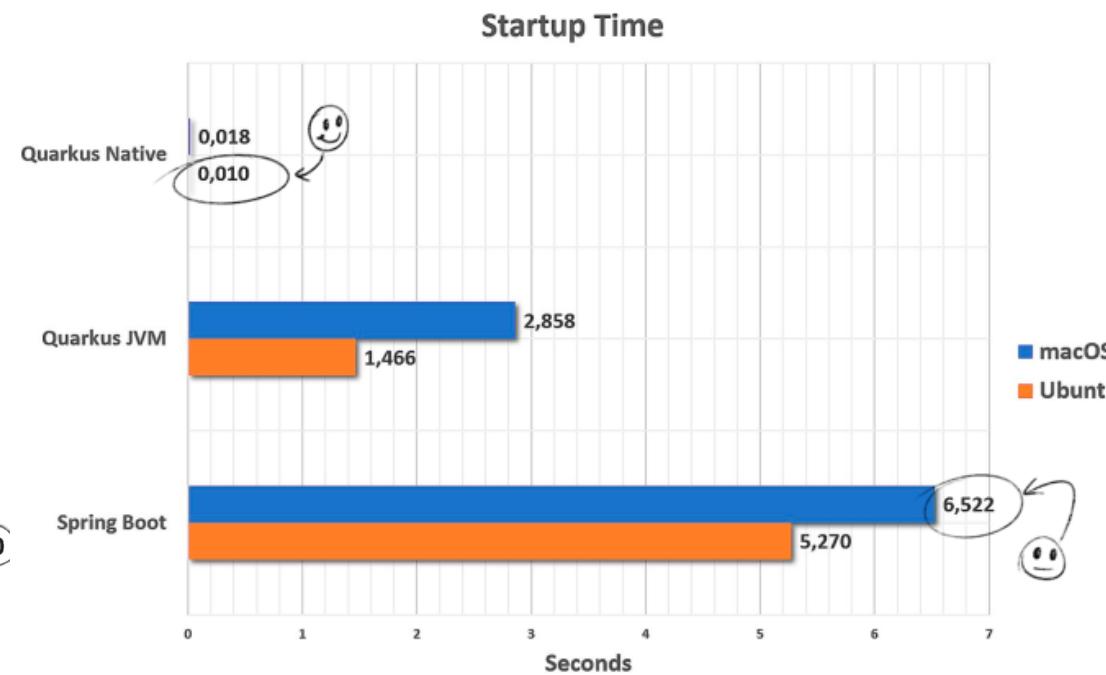
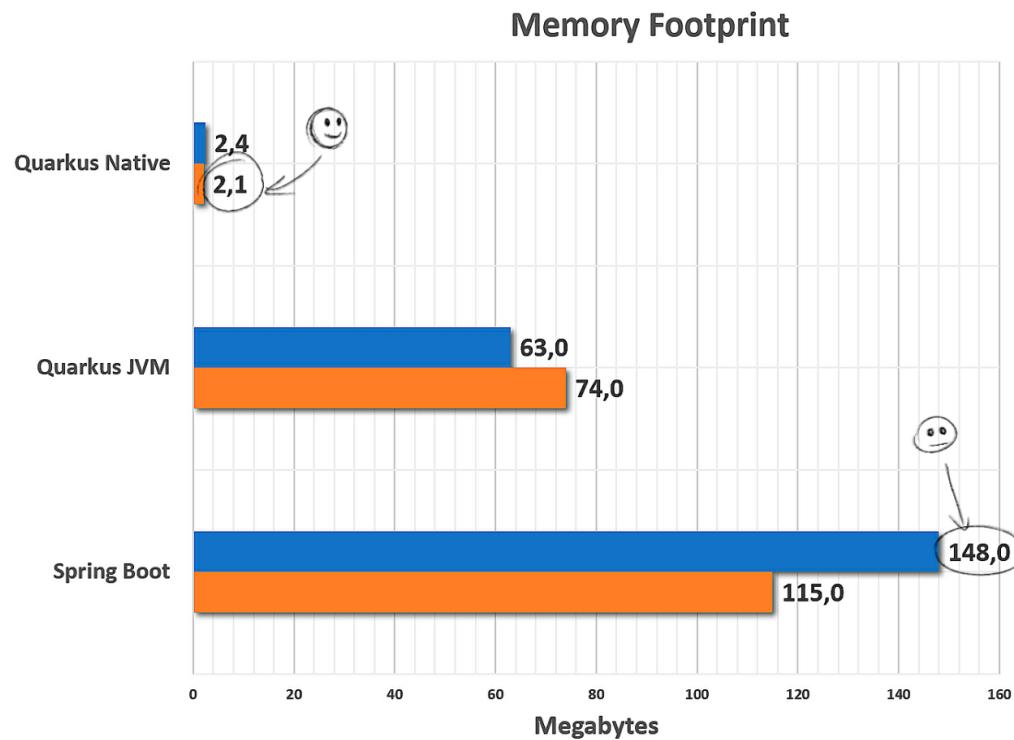


- A full-stack, Kubernetes-native Java framework made for Java Virtual Machines and native compilation
- Optimizes Java specifically for containers
- Tailored for [GraalVM](#)
- Live development mode



*RSS: the Resident Set Size and is used to show how much **memory** is allocated to that process and is in RAM. It does not include **memory** that is swapped out

QUARKUS VS SPRING (PERFORMANCE)



QUARKUS VS SPRING (FEATURES)

Feature	Quarkus	Spring
Build time class initialization	yes	no
Kubernetes resources generation	yes	no
GraalVM native images support	yes	limited
Dependency injection & components management	CDI, Spring DI extension	Spring Core
Reactive / non-blocking web stack	Vert.x	Reactor Netty
Simplified data access	Panache, Spring Data JPA extension	Spring Data: JPA, JDBC, MongoDB, LDAP, KeyValue
Application monitoring	MicroProfile Health, MicroProfile Metrics	Spring Boot Actuator
Resilience & Fault tolerance	MicroProfile Fault Tolerance	Netflix Hystrix
Online project starter	https://code.quarkus.io/	https://start.spring.io/

RESOURCES



LEARN MORE

- [Kubernetes Basics from Branden Burns](#)
- [Improved Docker Container Integration with Java 10](#)
- Octant: <https://octant.dev/>
- Skaffold: <https://github.com/GoogleContainerTools/skaffold>
- Docker networking: <https://docs.docker.com/network/>

FURTHER READINGS*

- [Docker for Java Developers](#)
- [Kubernetes for Java Developers](#)
- Introducing Istio Service Mesh for Microservices <http://bit.ly/istio-book>
- Microservices for Java Developers <http://bit.ly/javamsabook>

* All these resources are free

THANK YOU!



[helayoty/Kubernetes-101-for-Java-developers](#)



[@helayoty](#)



[hebaelayoty](#)

INTRODUCTION TO HELM CHARTS

- Package manager for Kubernetes applications
- Helm terminology:
 - Repository
 - Helm Chart
- Helm 2 vs Helm 3

