Lightweight PaaS for Jenkins CI Environments with Docker

Josef Fuchshuber
QAware GmbH
http://www.qaware.de

June 25, 2014

#jenkinsconf

About me

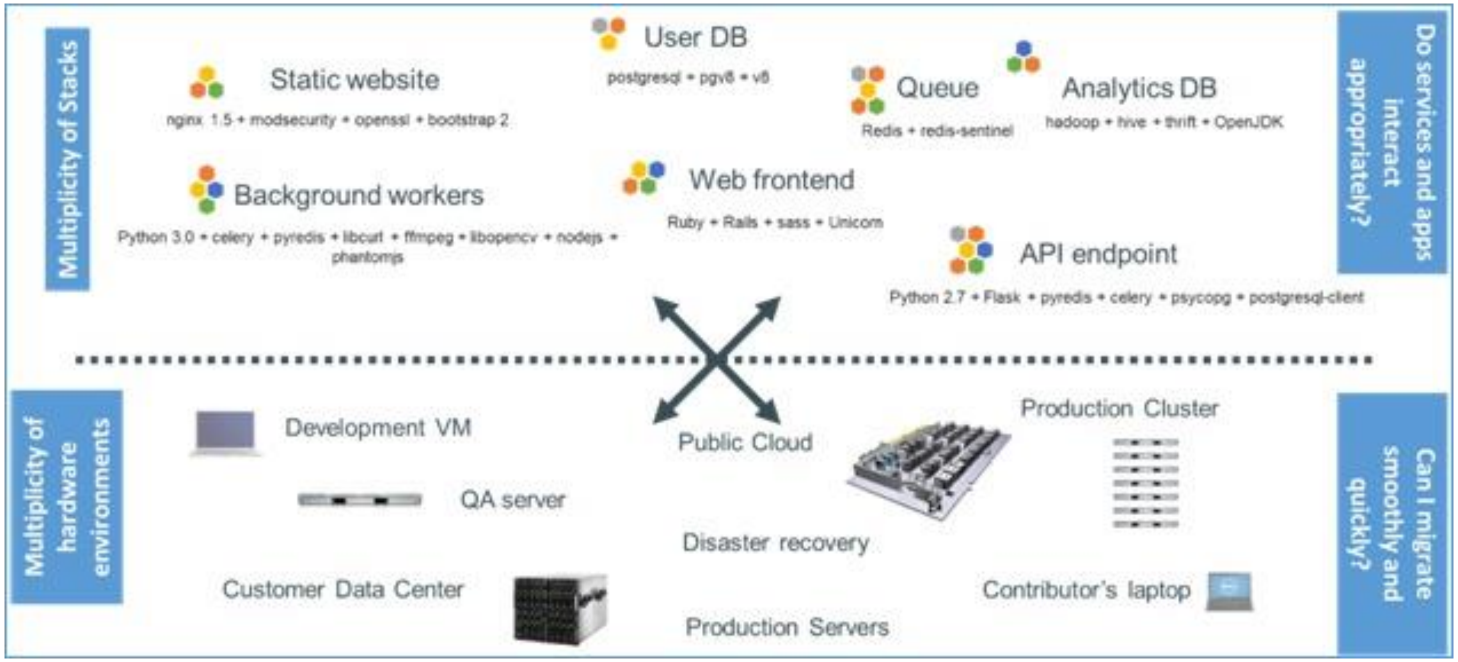
- Job:
  - Senior Software Architect (JEE)
  - Manager of software development tools
- Company: QAware GmbH, Munich
  - Custom software company
  - Customers:
    - Telecommunication industry
    - Automotive industry

Agenda

- Introduction to Docker
- Dynamic Build Slaves with Docker
- Lightweight Paas: Continuous Deployment with Jenkins and Docker

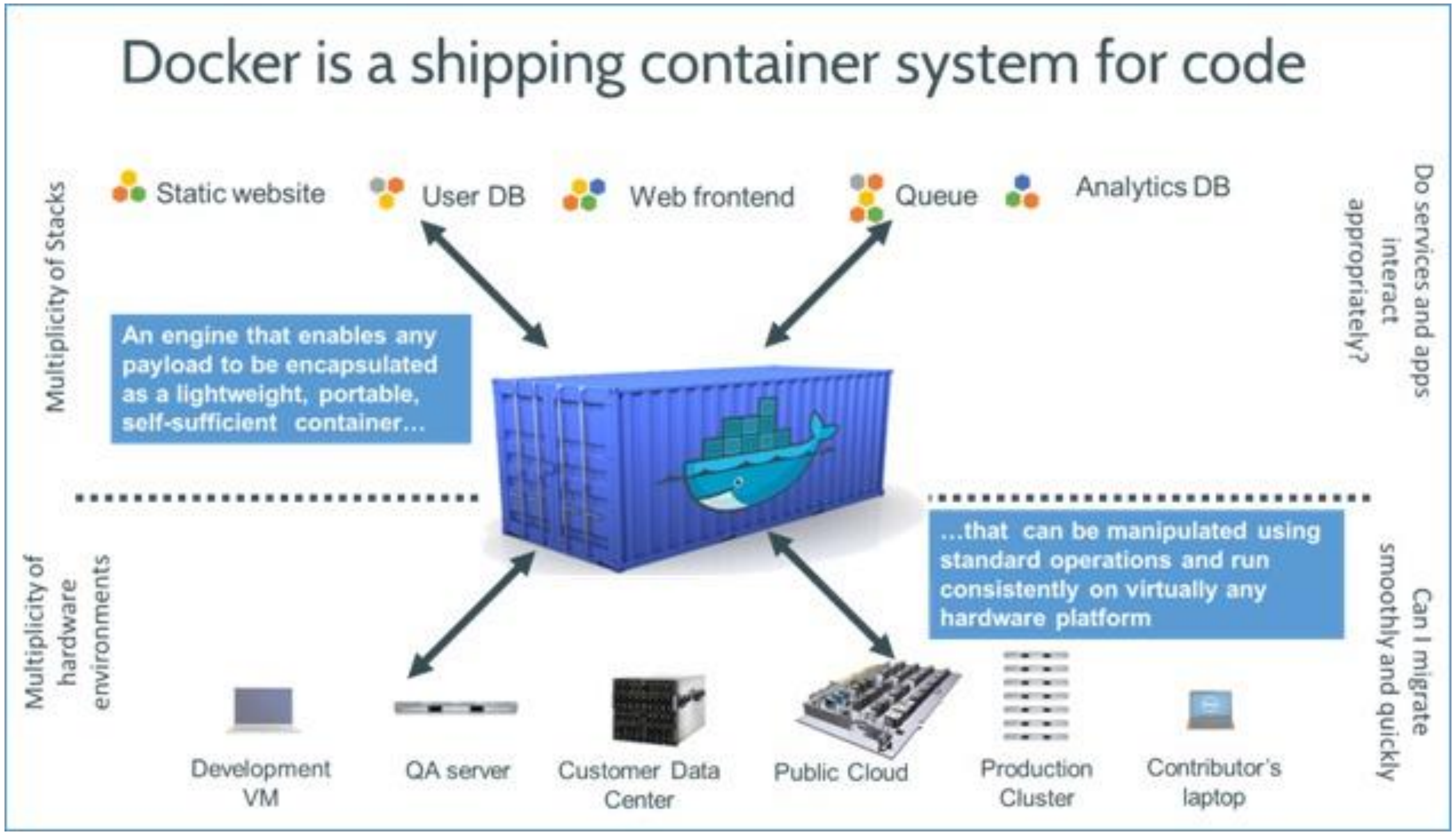
“Containerization is the new virtualization”

The Challenge


Multiplicity of Stacks
Static website
nginx 1.5 + modsecurity + openssl + bootstrap 2
User DB
postgresql + pgv8 + v8
Queue
Redis + redis-sentinel
Analytics DB
hadoop + hive + thrift + OpenJDK
Background workers
Python 3.0 + celery + pyredis + libcurl + ffmpeg + libopencv + nodejs + phantomjs
Web frontend
Ruby + Rails + sass + Unicorn
API endpoint
Python 2.7 + Flask + pyredis + celery + psycopg + postgresql-client
Do services and apps interact appropriately?
Multiplicity of hardware environments
Development VM
QA server
Public Cloud
Production Cluster
Disaster recovery
Customer Data Center
Production Servers
Contributor's laptop
Can I migrate smoothly and quickly?

source: https://www.docker.io/the_whole_story

The Docker solution


Docker is a shipping container system for code
Multiplicity of Stacks
Static website
User DB
Web frontend
Queue
Analytics DB
Do services and apps interact appropriately?
An engine that enables any payload to be encapsulated as a lightweight, portable, self-sufficient container...
...that can be manipulated using standard operations and run consistently on virtually any hardware platform
Multiplicity of hardware environments
Development VM
QA server
Customer Data Center
Public Cloud
Production Cluster
Contributor's laptop
Can I migrate smoothly and quickly

source: https://www.docker.io/the_whole_story

Quick Facts: docker

- Docker Inc. (formerly dotCloud Inc.)
- Introduced in March 2013
- Current version 0.11 is the release candidate for 1.0
- Huge Github community (12,000 stars and counting)
- Open source: Apache v2 licence

Why should we use Docker?

- It makes it easy to create lightweight, portable, and self-sufficient containers.
- Configure once, run anywhere
- Solving dependency hell
- Huge win for automation and deployments
- Containers are perfect for:
  - Continuous integration & test applications
  - Build services
  - Run services
  - Build your own Platform-as-a-Service (PaaS)

Docker is a client-server application

- Docker client and daemon can run on the same system.
  - You can connect a Docker client with a remote Docker daemon.
  - They communicate via sockets or through a RESTful API.
- Users interact with the client to command the daemon, e.g. to create, run, and stop containers.
- The daemon, receiving those commands, does the job, e.g. run a container, stop a container.

Some Docker vocabulary

- Container
- Image
- Layer
- Dockerfile
- Registry
- Repository
- Tag

Insights: Under the hood

Union file systems (UnionFS)

- Union file systems operate by creating layers.
- They combine layers into a single image
- Supports: AUFS, btrfs, vfs and DeviceMapper

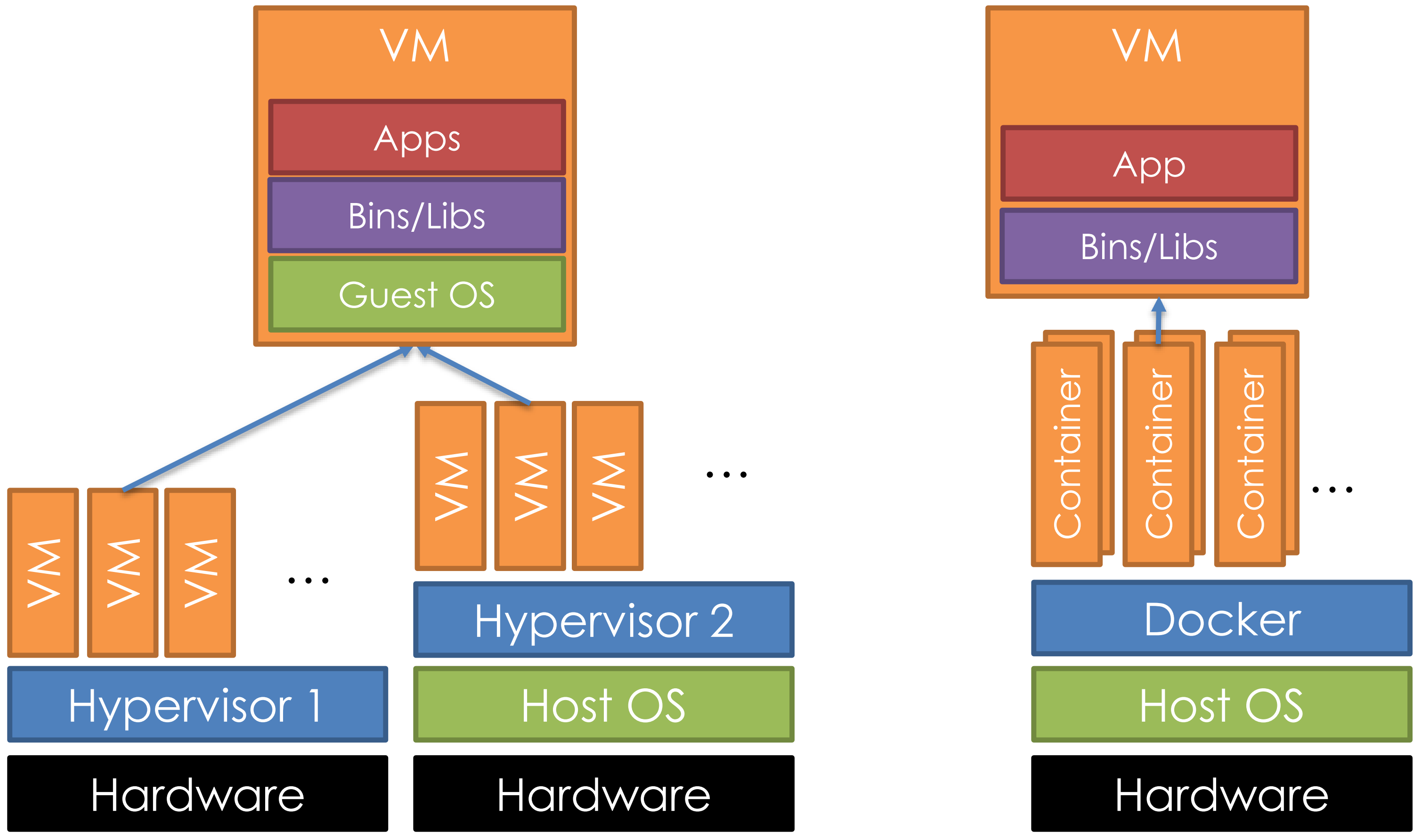
Namespaces & control groups

- Provide isolated workspace for containers
- Controlling resource (CPU, memory, block I/O, network, etc.)

Container formats

- Wraps these all together:
  - Libcontainer
  - LXC
  - …

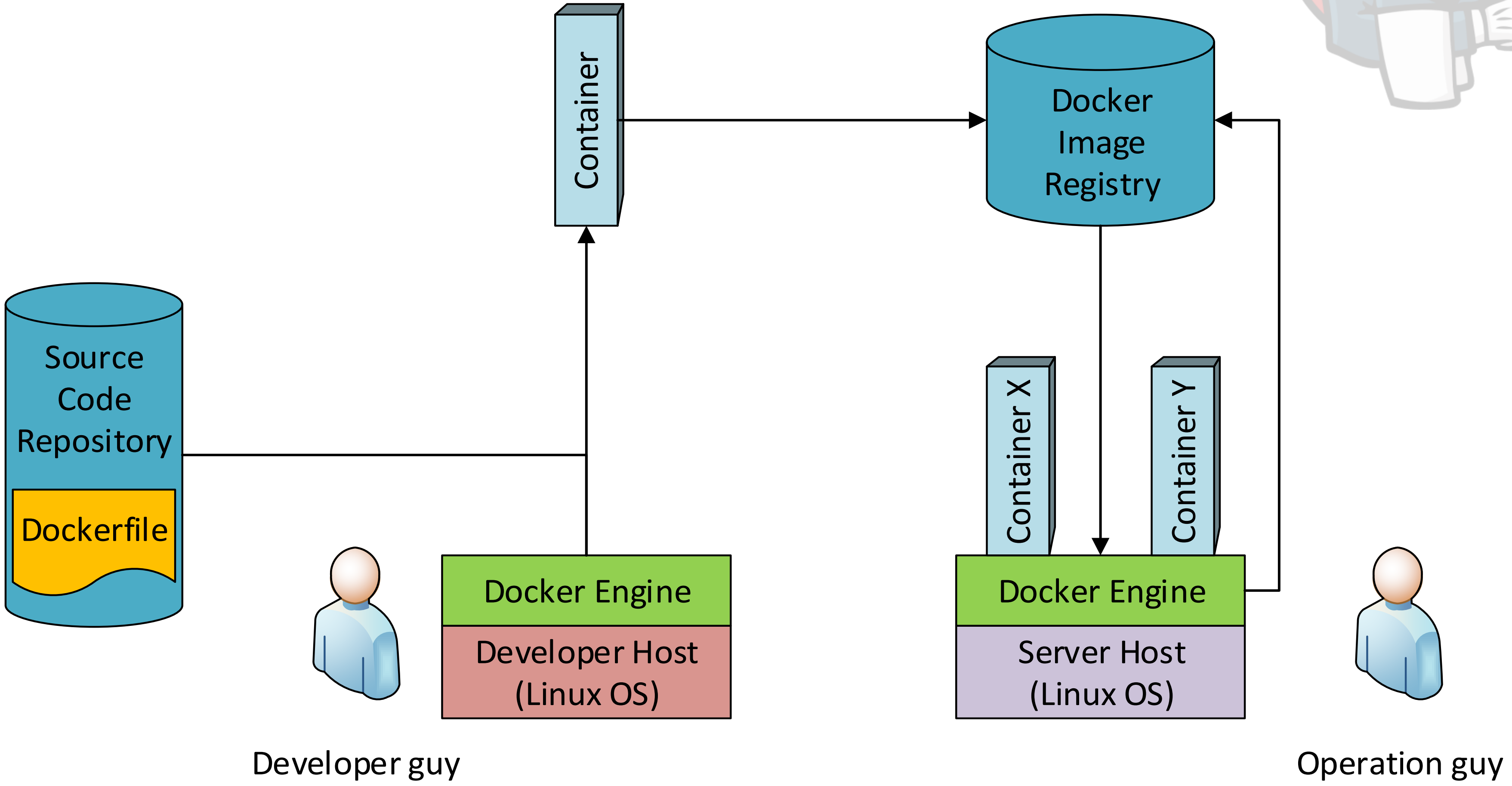
VMs vs. Containers



Type 1 / Type 2 Virtualization

Containerization

The Docker workflow

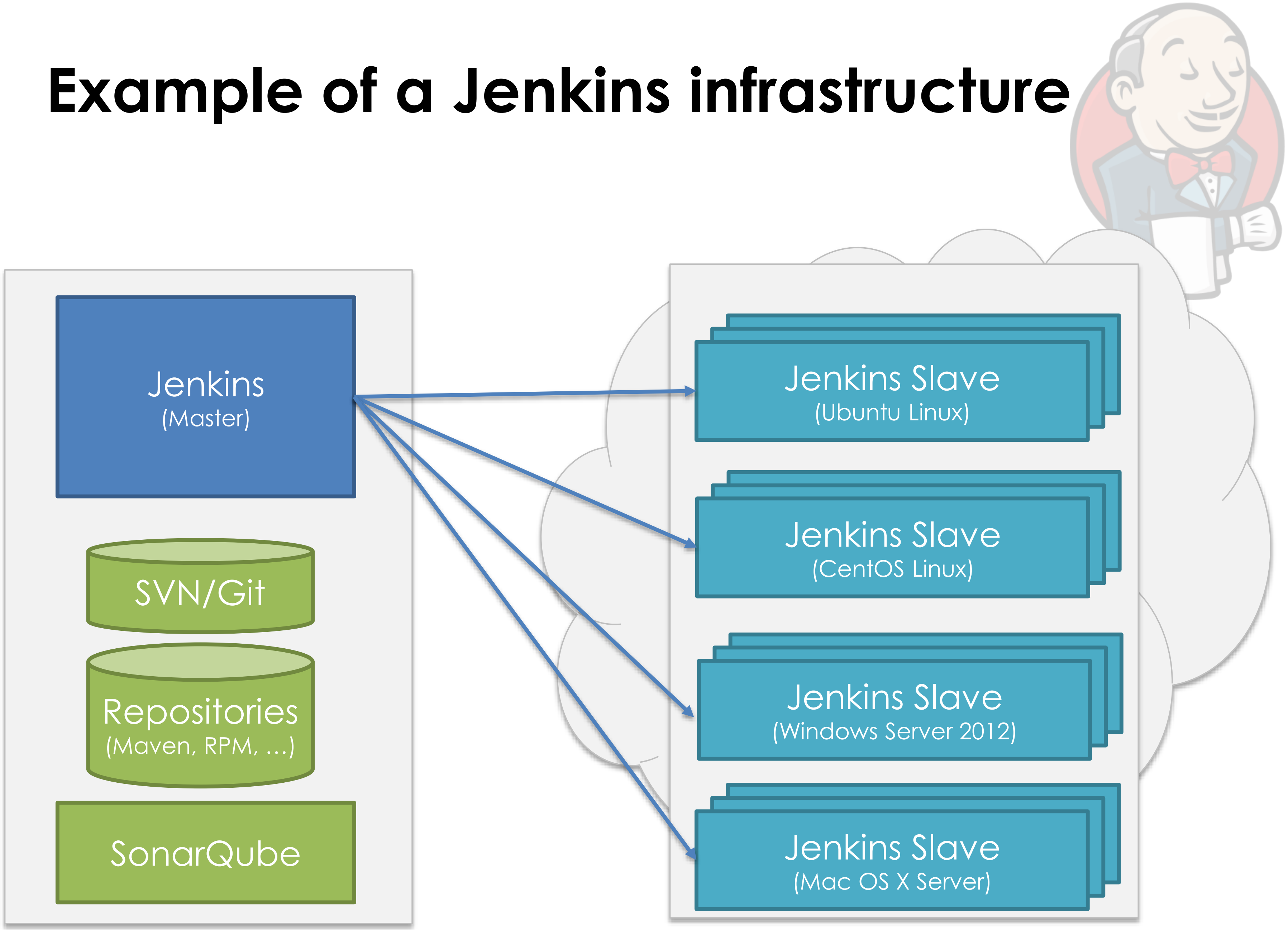


DEMO

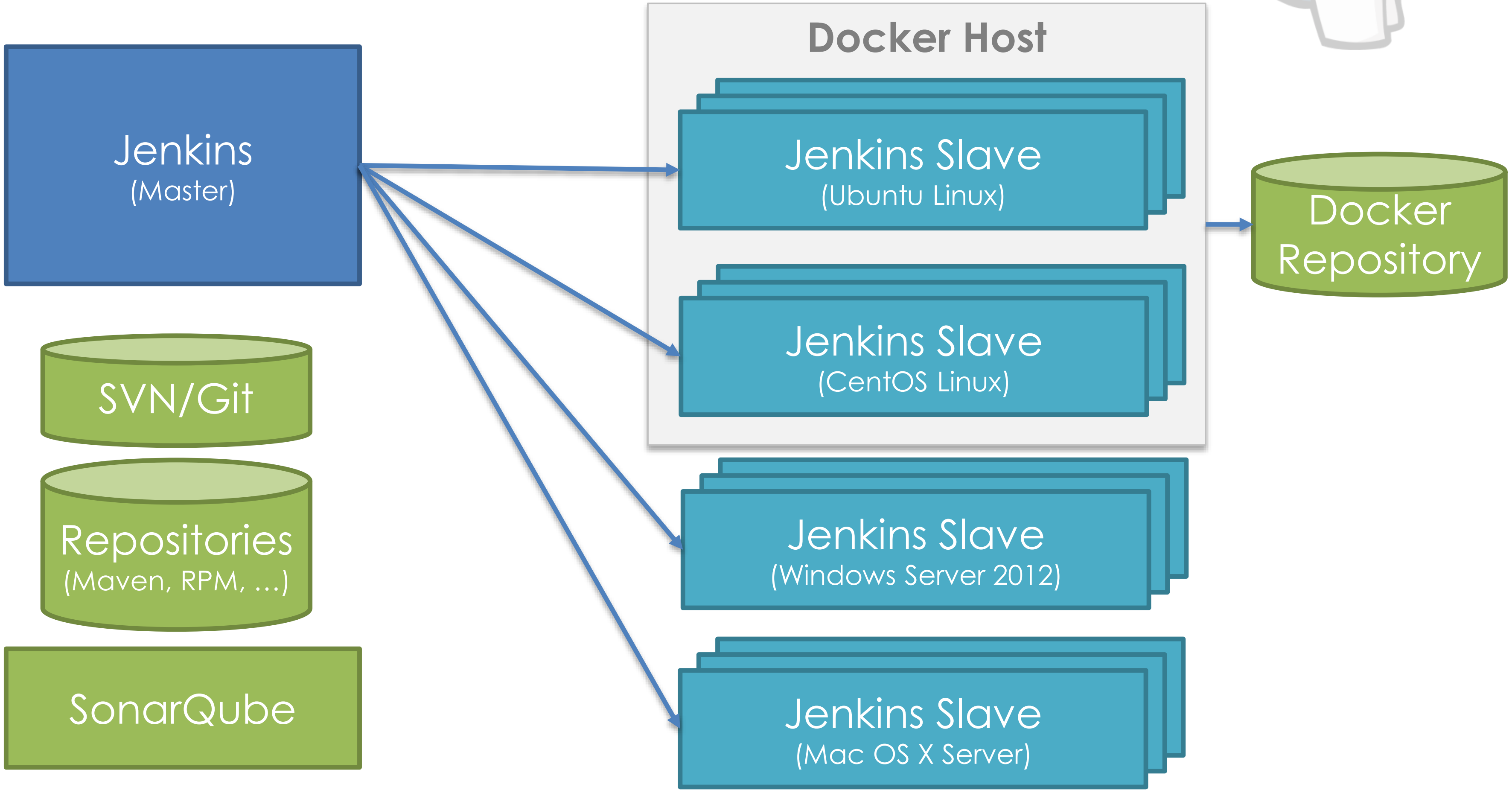
Agenda

- Introduction to Docker
- **Dynamic Build Slaves with Docker**
- Lightweight Paas: Continuous Deployment with Jenkins and Docker

Example of a Jenkins infrastructure



What Docker can change



The idea: The Docker build workflow

1. Developers look for base images in public/private Docker repositories
2. Optional: Customize it and push it back to the Docker repository
3. Jenkins administrator defines images as Jenkins slave
4. Define Jenkins job(s) for this Docker slave.
5. Run Jenkins job:
   1. Jenkins starts defined image as Docker container
   2. Jenkins runs the job inside the container
   3. Jenkins stops the container (tag it)

Quick Facts: Docker Plugin

- Links:
  - https://wiki.jenkins-ci.org/display/JENKINS/Docker+Plugin
  - https://github.com/jenkinsci/docker-plugin
- Features:
  - Dynamic provisiioning of a Jenkins slave on a Docker host
  - Run a single build job
  - Tear-down the slave
  - Commit the container

Requirements for the container

- Connectable: SSH server
- Accessible: User (e.g. „jenkins“)
- Runnable: Java JDK

- Documentation: https://wiki.jenkins-ci.org/display/JENKINS/Docker+Plugin
- Ready-made jenkins slave: „evarga/jenkins-slave“

Why use dynamic slaves with Docker

- Fast startup
- Every job runs in its own clear container
- Job-parallelization is no problem
- Lazy resource binding and no long-running processes
- Devops: Separation of concerns
  – Developer: Worries about the container's inside
  – Ops: Worries about the container's outside

DEMO

Agenda

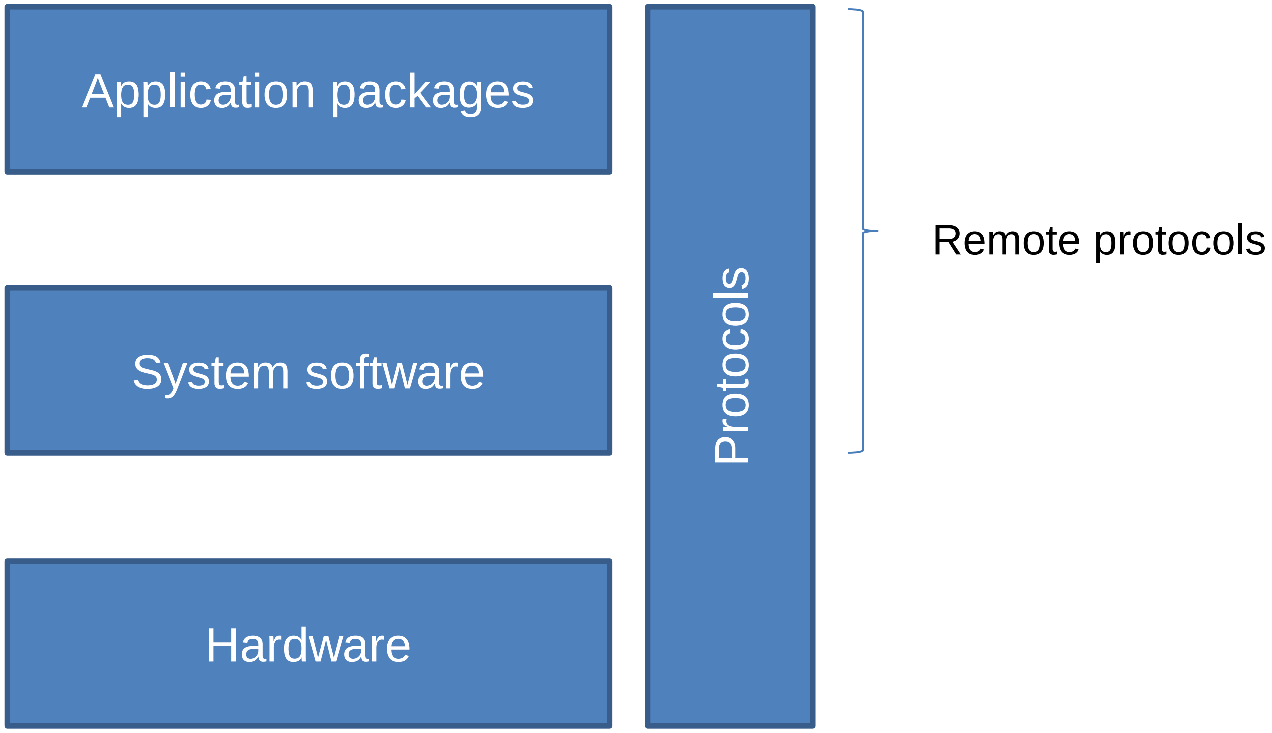
- Introduction to Docker
- Dynamic Build Slaves with Docker
- **Lightweight Paas: Continuous Deployment with Jenkins and Docker**

Definition of QAware‘s „TI architecture“

- JAR/WAR/EAR
- (Cron) jobs
- …

- OS
- VM (Java, .NET, …)
- Server (db, web, …)
- Libraries
- …

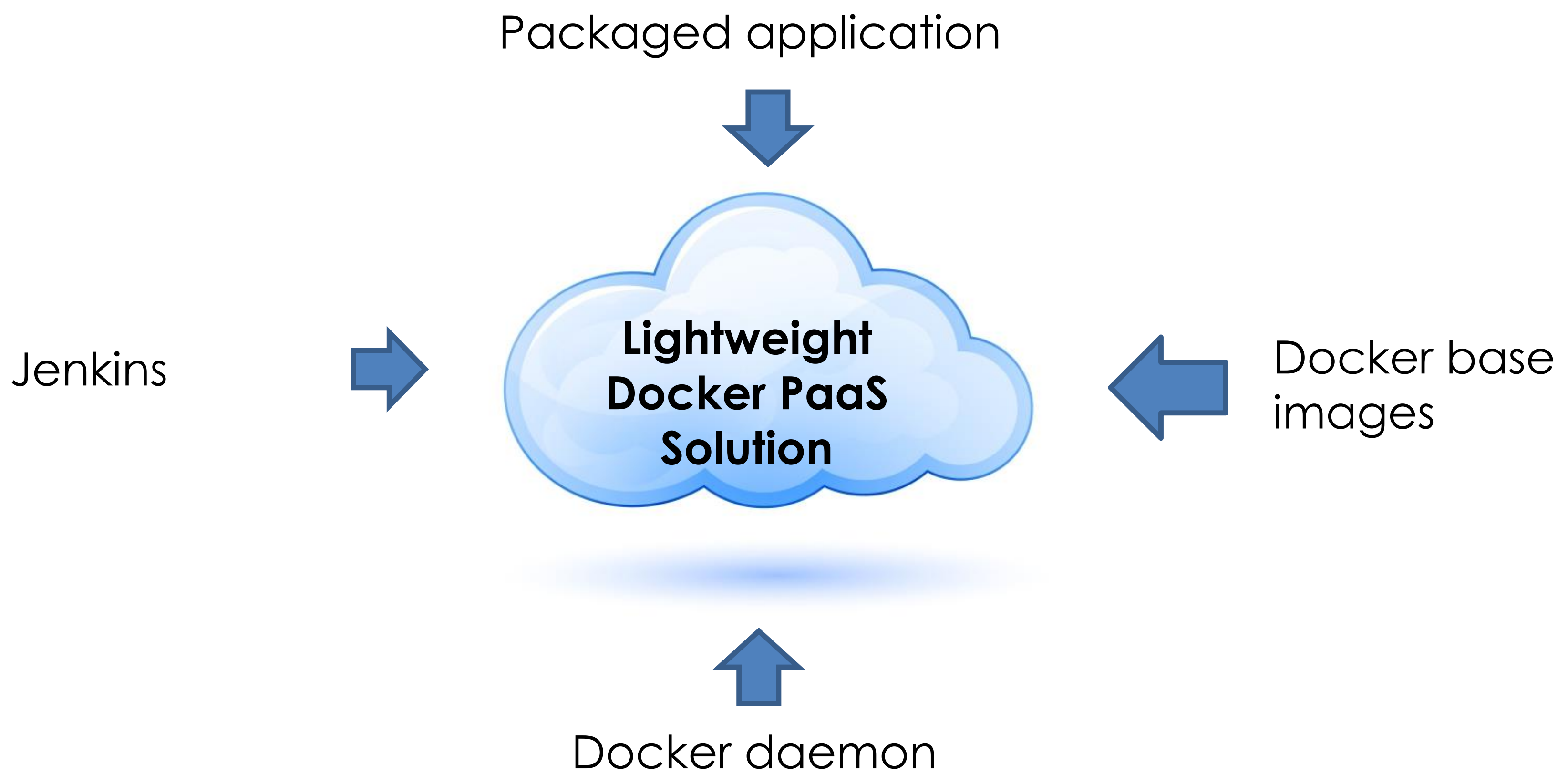
- Server
- Memory
- Network equipment
- …



Why to use an own lightweight PaaS solution in your CI/CD process?

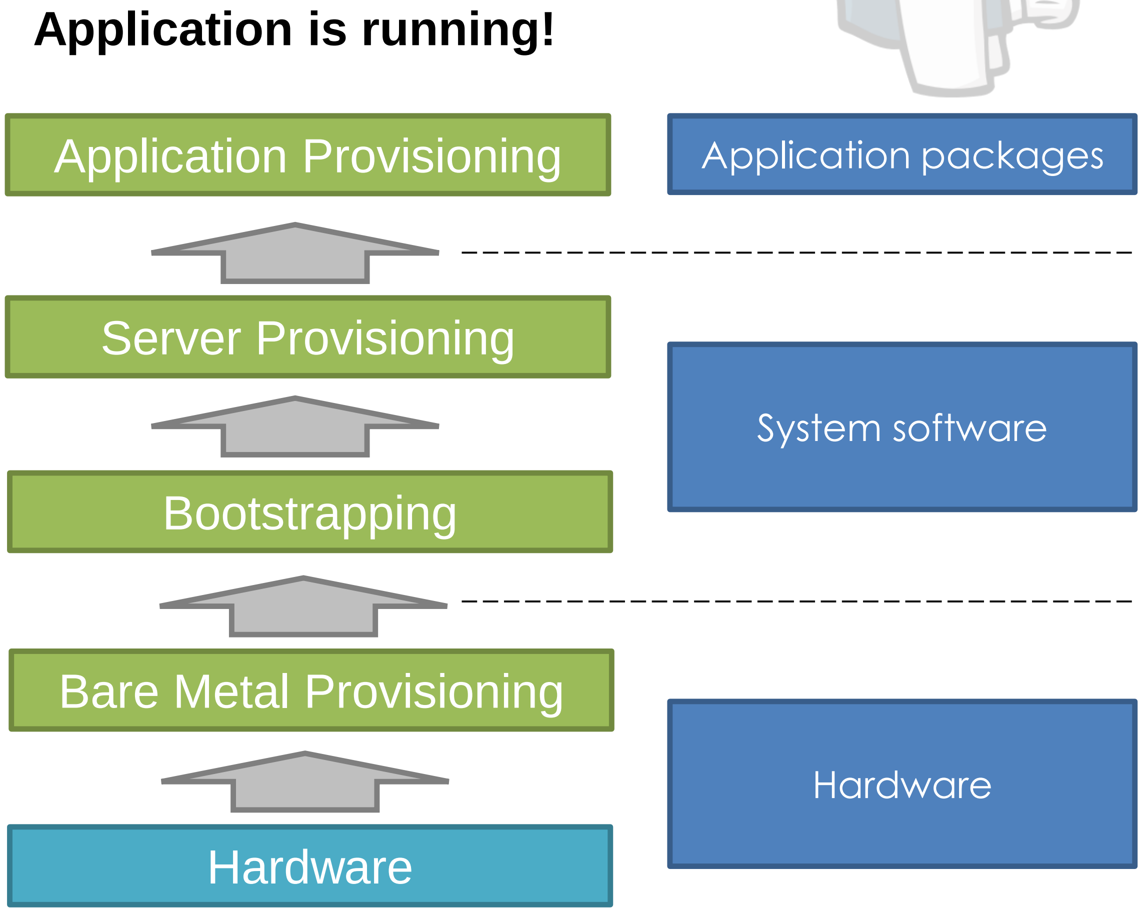
- You can‘t use a commercial PaaS.
- You have a complex TI architecture.
- Integrate early and often! We need an scalable automated build and deployment process.

Lightweight PaaS components

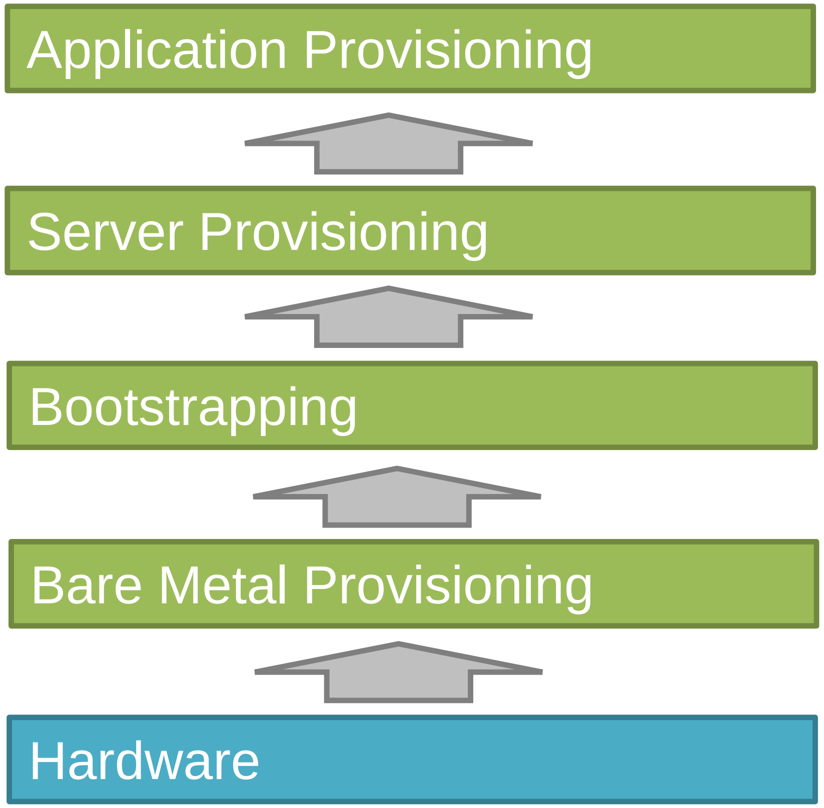


What is Provisioning?

- Application deployment and server configuration.
- Prepare server and runtime environments; OS configuration
- Prepare OS
- Prepare virtualization layer / cloud infrastructure
- Prepare hardware



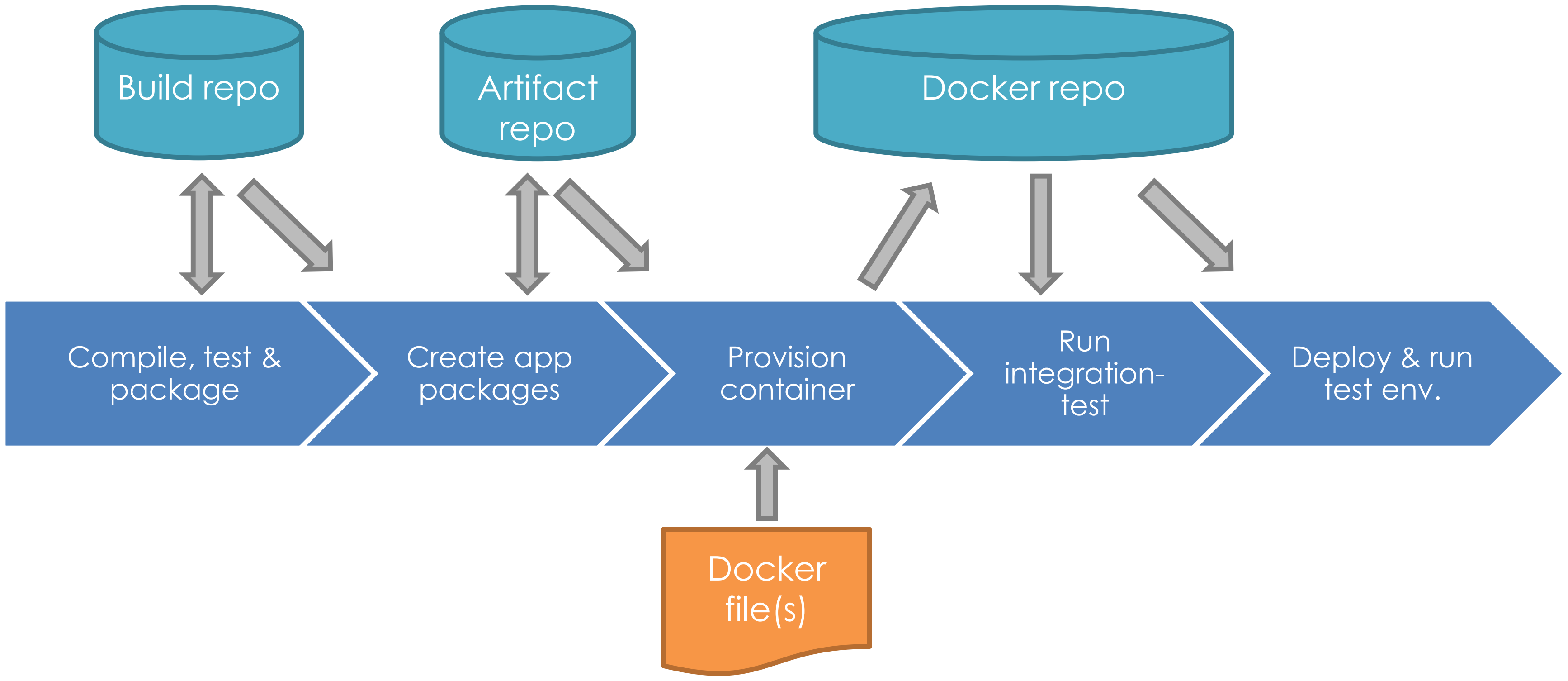
Docker makes „Infrastructure as Code“ easy



- Load and start images
- Prepare images with Dockerfiles

- Base images
- Docker runtime

The Jenkins build-pipeline for Docker deployments



Quick Facts: Docker build publish Plugin

- Links:
  - https://wiki.jenkins-ci.org/display/JENKINS/Docker+build+publish+Plugin
  - https://github.com/jenkinsci/docker-build-publish-plugin
- Features:
  - Only a Dockerfile needed to build your project
  - Tag the image build
  - Publish to private or public Docker repository

Quick Facts: Docker build step plugin

- Links:
  - https://wiki.jenkins-ci.org/display/JENKINS/Docker+build+step+plugin
  - https://github.com/jenkinsci/docker-build-step-plugin
- Features:
  - Execute Docker commands into you job as a build step
  - Export build variables
    - *DOCKER_CONTAINER_IDS* – The IDs of created/started containers
    - *DOCKER_IP_$HOSTNAME* – The IP of running container with hostname $HOSTNAME

DEMO

Thank You To Our Sponsors

Platinum

CloudBees

Gold

zend The PHP Company

MidVision™ Release the innovation

codecentric

Silver

XebiaLabs the deployment automation company

SOASTA Test Faster. Release Sooner.

Corporate

cloudera®

Community

Java User Group Berlin Brandenburg

Lightweight Java User Group Munich