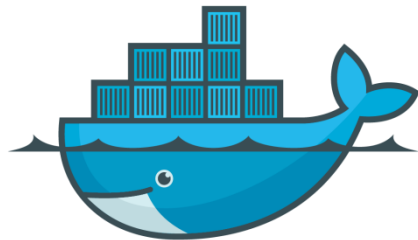




**Jenkins**



**docker**



**Infrastructure for openMDM  
Continuous Integration Environment**





- **ORGA-39 Define infrastructure components for Integration environment**
- ORGA-62 Provide continuous integration for integration environment
- ORGA-66 setup basic infrastructure in eclipse hosted environment
- ORGA-67 setup oracle database for integration environment

# Basic Concept



- Continuous integration (CI) environment on central server
- Continuous build of openMDM components
- Tests in the integration environment with Oracle database & ODS server
- Criteria for good tests
  - well defined environment
  - reproducible
  - independent
  - scalable



# Jenkins

- Jenkins & Hudson
- State of the art open source CI servers
- Run build jobs, tests, deployment
- Hudson: Eclipse project
  - provided on Eclipse foundation's CI servers

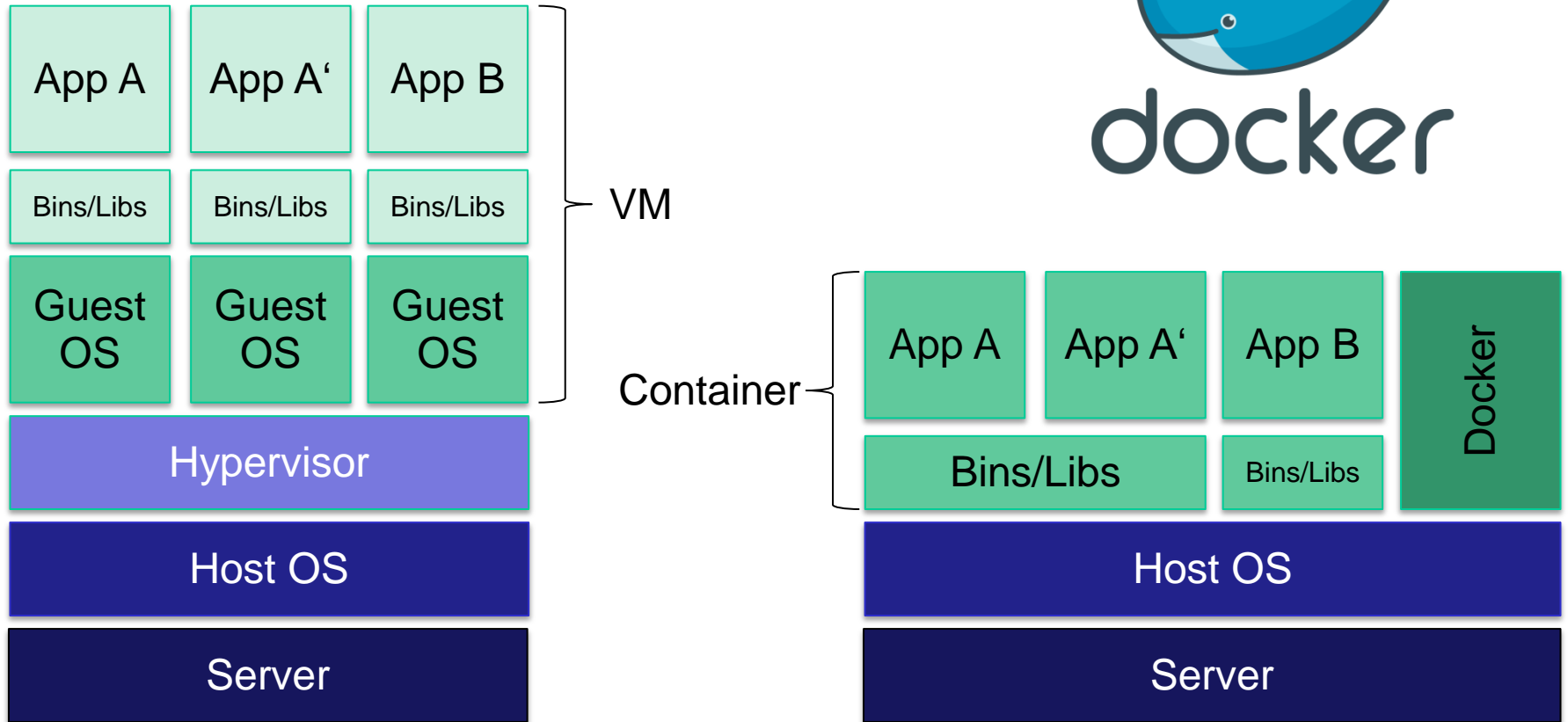
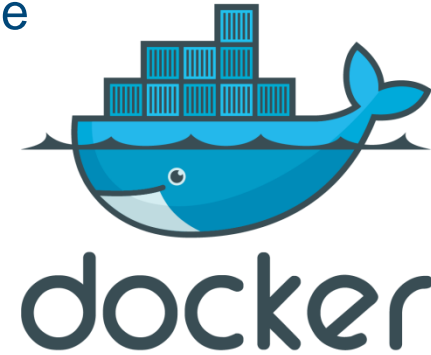
# Hudson



# Technical Solution – Virtualization

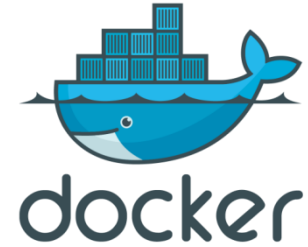
## Docker

- State of the art, lightweight, open source virtualization
- Independent containers run on a single OS instance
- Containers are isolated, but share OS, bins, libs



# Technical Solution – Virtualization

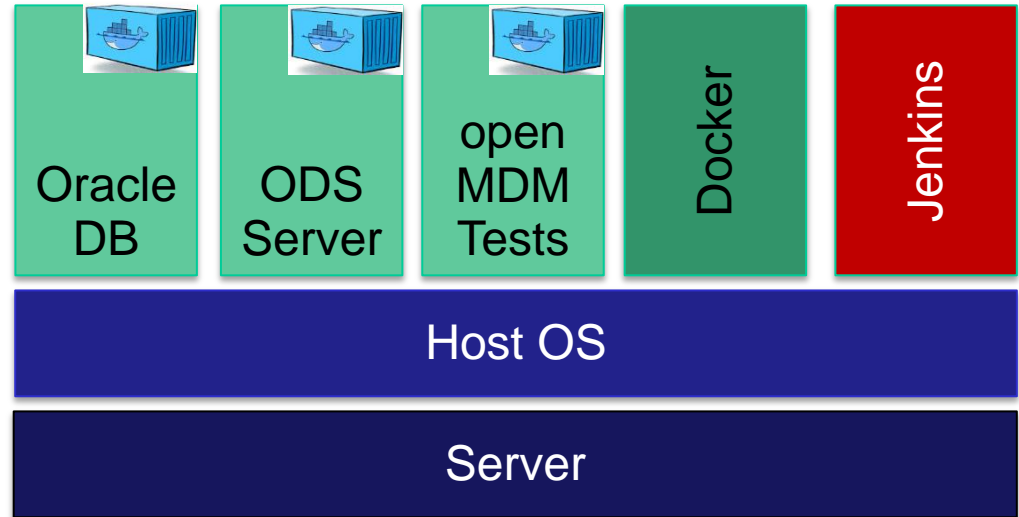
## Docker



- Containers are build of images
- Images contain everything needed for a single application
- Images can be predefined and shared on Docker Hub
  
- Multiple instances of the same container may be started in parallel
- Container instances may communicate with each other via network
- State of containers is deleted after shutdown
  - resources from within the container may be preserved
- Every instance of a container starts with exactly the same environment

# Implementation

- Docker images for
  - Oracle DB
  - ODS Server (currently Peak ODS Server)
- Jenkins CI Server
  - runs CI Builds
  - creates openMDM Tests container
  - starts DB, ODS server, and test containers
  - evaluates and publishes results of test runs
  - may run multiple tests in parallel



# Variations of Implementation



Jenkins may start Docker containers for

- each **test**

PRO: no DB cleanup and setup after each test (fast)

CONTRA: start up time for containers (slow)

- each **build**

PRO: start up time for containers only once (fast)

CONTRA: cleanup and setup of DB for each test (slow),  
well defined environment and reproducibility may be influenced



# Integration in Development Environment



- Jenkins/Hudson checks out code from Git repository
  - Checks regularly for changes
  - On demand
- Eclipse can be connected to Jenkins/Hudson
  - Developer gets informed about failure/success of builds
  - Push up message/mail

# Next Steps

- HighQSoft Avalon ODS server
- Setup of ODS server with MDM scheme
- Setup on Eclipse cloud
- Setup of first CI builds
- Implementing tests

## Obstacle

- Eclipse cloud's OS currently too old for Docker
  - OS get probably updated
  - Jenkins/Hudson supports slave CI servers that may run Docker

