
JavaCro'18

# Docker for Developers

Željko Trogrlić

# First Era


- Go through long documentation, if there is one
- Download library source or JARs from Net
- Beg for help
- Be lucky if it works in a week

# Second Era


- Use build tools: Maven or Gradle
- Get dependencies from repository
- Do the rest manually or get an VM image
  - Database
  - Web server
  - Reverse proxy
  - User management (LDAP, Keycloak)
- Documentation, begging, or both

# Third Era

JavaCro'18

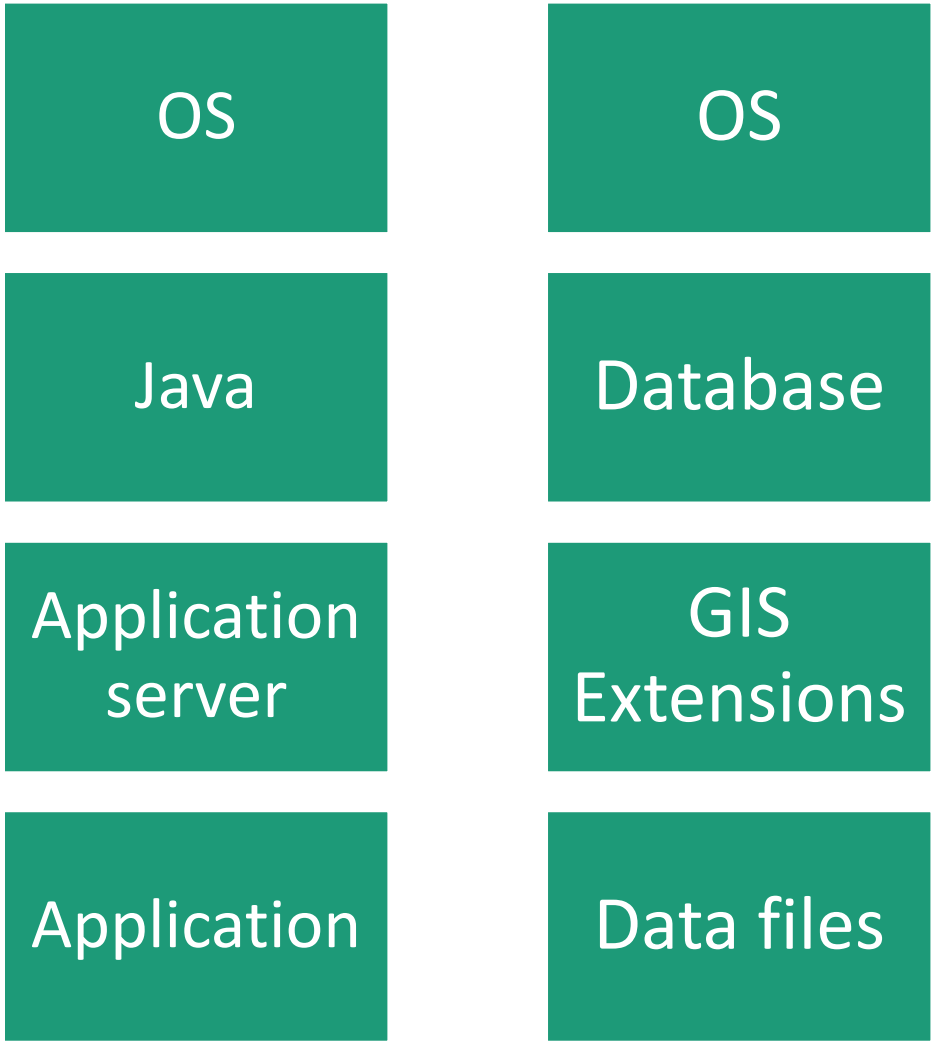
- Can new developer become productive within an hour?

# Docker Containers vs Virtual Machines

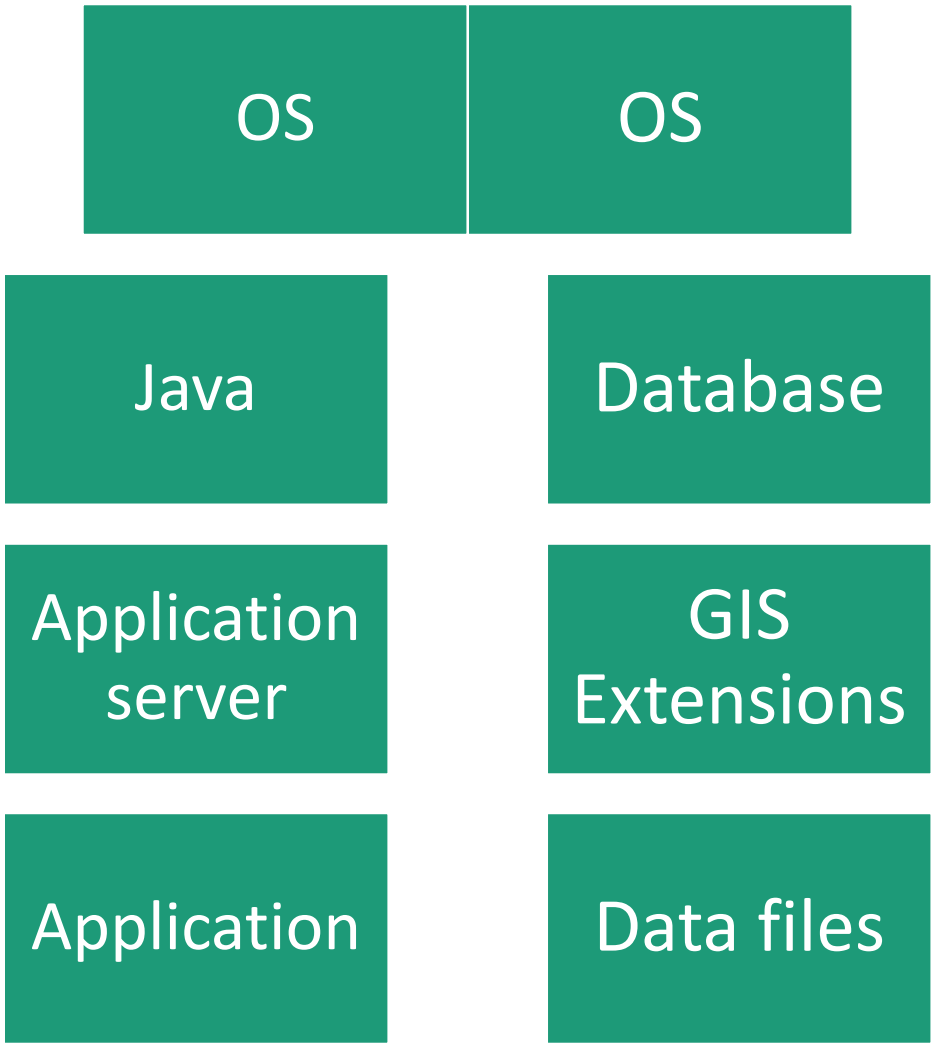


- Shared Kernel
  - Kernel with all modules is already „on”
- Layered file system
- Shared images
- Bonus: huge selection of ready-made containers

# Layered File System



# Layered File System



# Levels of Dockerization


- Docker
  - static images
- Docker Compose
  - multi-container environment description
- Docker Swarm or Kubernetes
  - Managed swarms of containers

# How to Get Docker


- Linux: whatever you usually do
- Windows
  - Pre-10: Docker Toolbox (uses VirtualBox)
  - 10: Docker
    - Hyper-V
    - Linux or Windows containers

# Docker & Spring Boot Example


- Application container
  - OS (Alpine Linux)
  - Java
  - Spring application
- Database container
  - OS
  - PostgreSQL
  - Get it from Docker Hub

# Plain Vanilla Configuration


- Bake files into image (docker)
  - `Dockerfile` per image
- Orchestrate images (docker-compose)
  - `docker-compose.yml`
  - Additional compose files

# Off We Go!

JavaCro'18

- gradlew build
- *docker-compose build*
- docker-compose up –d
- docker-compose ps
- docker-compose logs
- docker-compose down

# Development Optimizations


- Expose more ports for debugging
- Update your app quickly

# How to Update Your App


- Rebuild container every time
- Kill container and work as usual
  - Expose ports from other containers
- Take base image as-is and overlay output file
  - image: openjdk:8-jre-alpine
  - volumes:
    - - .\server\build\libs:/usr/src/sausage
    - - .\server\src\main\resources:/usr/src/sausage/config

# Issues: Handling Many Files


- Docker architecture
  - Client (CLI)
  - Server: build and run
- Docker client copies *all* files to server
  - Slooow! Imagine Node.js folder.
- Use `docker.ignore`

```
*
!build/libs
!build/resources/main
```

# Issues: Unnamed Volumes Issue


- If files are not present in instance:
  - Put your files in „Users” directory
    - Only that directory is visible from VM
  - Otherwise, add directory share to VM

# Deploy Faster

JavaCro'18

- JVM hot swapping
- Spring-Loaded or JRebel
- spring-boot-devtools
  - Two class loaders
    - libraries
    - dev output

# Production

JavaCro'18

- Bake files into image
- Expose configuration as environment variables
- Store images into repository
- Devops take it from there

**Always do final testing with production scripts!**

```
docker-compose
    -f docker-compose.yml
    -f docker-compose.prod.yml
    up
```

# Third Era


- Run a script or two
- Be productive within an hour

**Build tool and Docker recreate complete development environment**

# Get Example

JavaCro'18

- https://github.com/zeljkot/presentations/tree/master/docker-intro

JavaCro'18

# Thank you!

# Alpine Linux

JavaCro'18

- Size: 5 MB (vs 55 for Debian base)
- Base
  - musl libc (hello 13k vs glibc 662k)
  - BusyBox
- ash: Almquist shell from BusyBox
- No Bash and other tools, but could be added