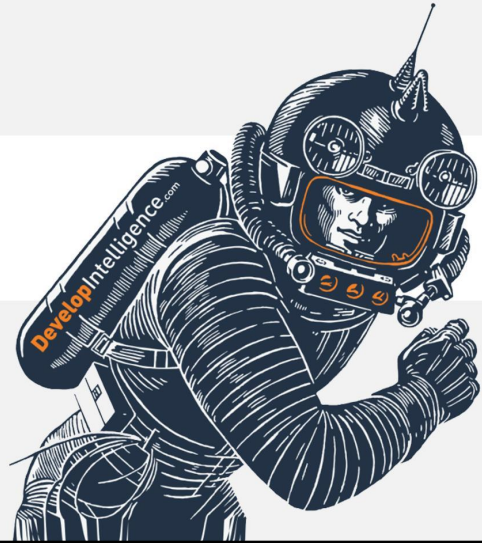


Containerization





Module Outline



- 12 Factor Apps
- Logging
- Configuration
- Other Considerations



12Factor Apps



1. Use version control
2. Explicit Dependencies
3. Separate Configuration
4. Decouple Backing Services
5. Build, release, run
6. Stateless processes
7. Port binding
8. Scale out concurrency
9. Disposability
10. Dev/Prod Parity
11. Logs as event streams
12. Separate admin processes

<http://12factor.net>



No log files! Log to stdout/stderr

External log aggregation to archive and search logs

Containers have limited, ephemeral local storage. So storing logs in log files would consume precious storage space that will likely run out. Also, containers will typically be "scaled out", with multiple instances of the application running across multiple machines. This means that debugging without centralized logging could be very difficult.



Used to specify environmental differences

- Dev DB vs Prod DB connection
- credentials
- etc

Should not live inside container image!

Ideally, expose to the service at runtime.

- Environment variables
- Dynamic config discovery

A container image should be considered the canonical build of an application and its dependencies. We want to deploy the exact same image of an application across our environments (dev, test, prod) in order to ensure our environments are as similar as possible. The only thing that should change for our application across environments are the configuration values exposed to it.



Eliminate app startup order (robustness and loose coupling)

Stateful/stateless (not critical)

Do not assume co-location of services



Containerization