RED HAT
FORUM
Europe, Middle East & Africa

redhat.

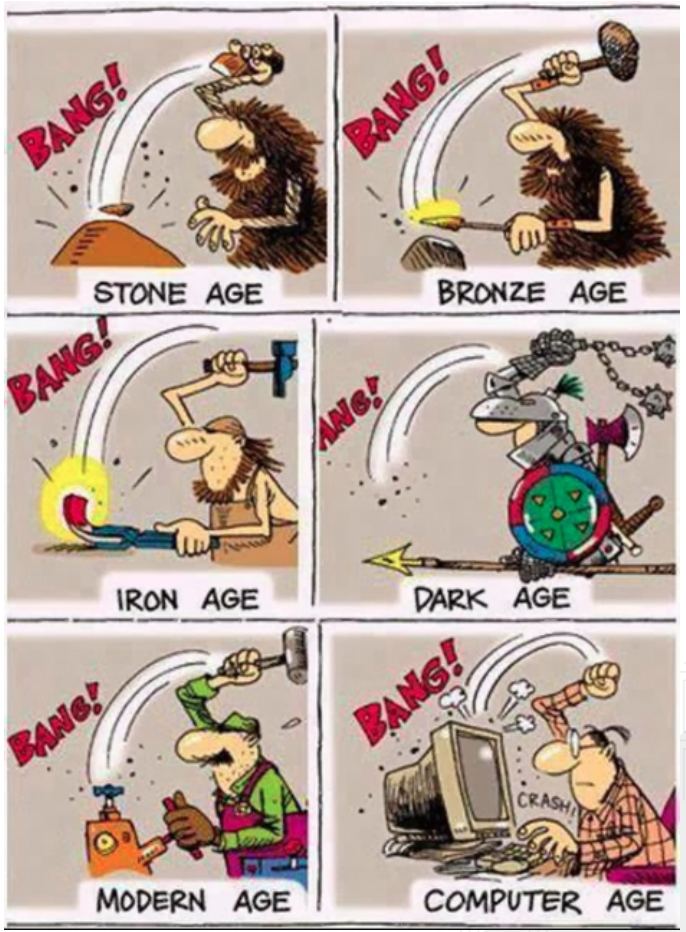
The Twelve Factor App Methodology

Entwicklung mit OpenShift

Robert Baumgartner
Senior Solution Architect
24. Oktober 2017

Humans had been problem solvers...

until they've decided to become PROGRAMMERS



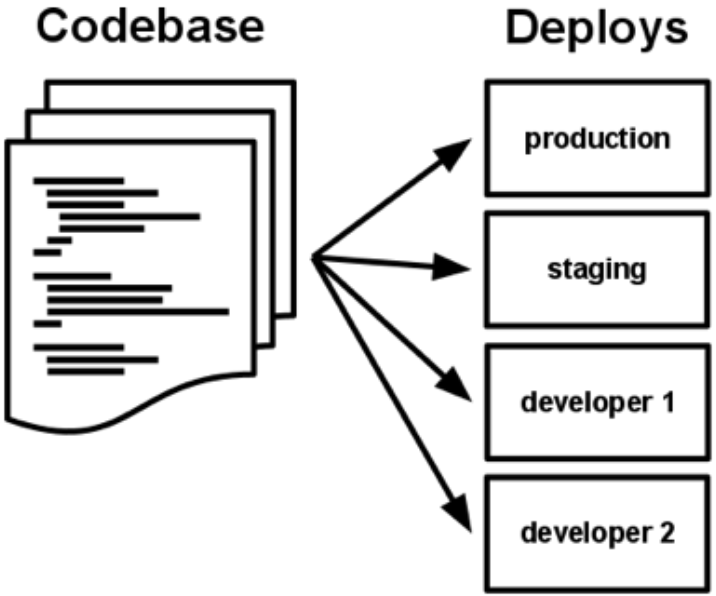
- **A methodology**
- Manifesto
- Best practices
- Principles

Created by heroku https://12factor.net/

1. Codebase
2. Dependencies
3. Config
4. Backing services
5. Build, release, run
6. Processes
7. Port binding
8. Concurrency
9. Disposability
10. Dev/prod parity
11. Logs
12. Admin processes

1 - Codebase

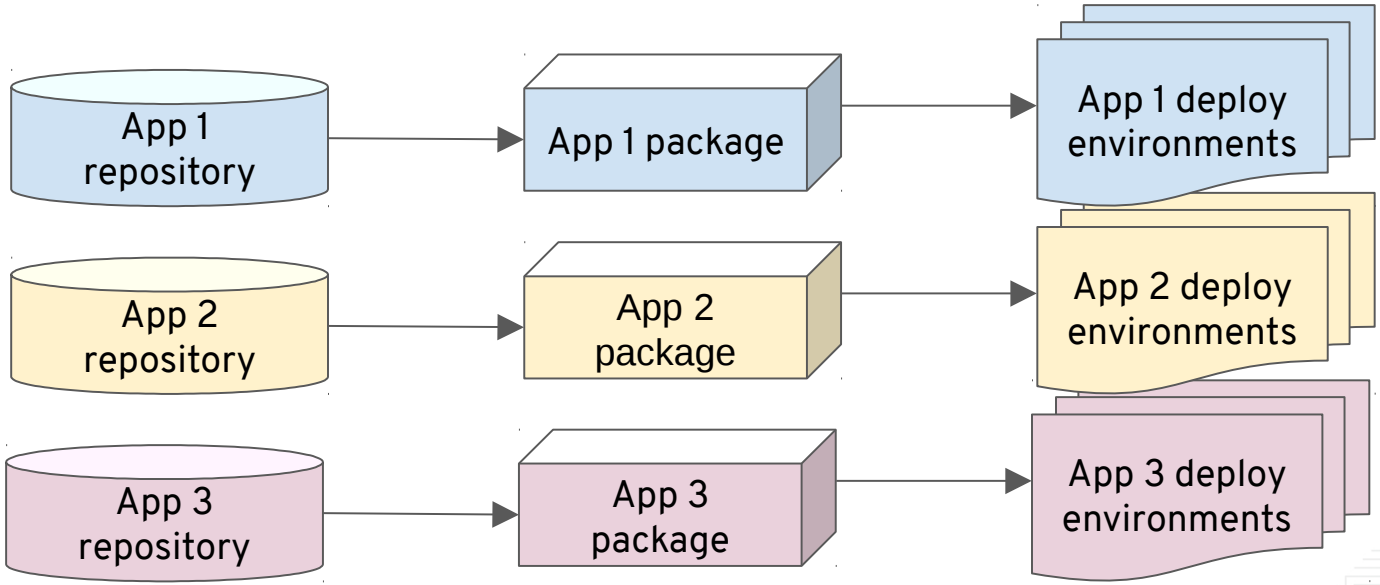
One codebase tracked in revision control, many deploys



One codebase = one app

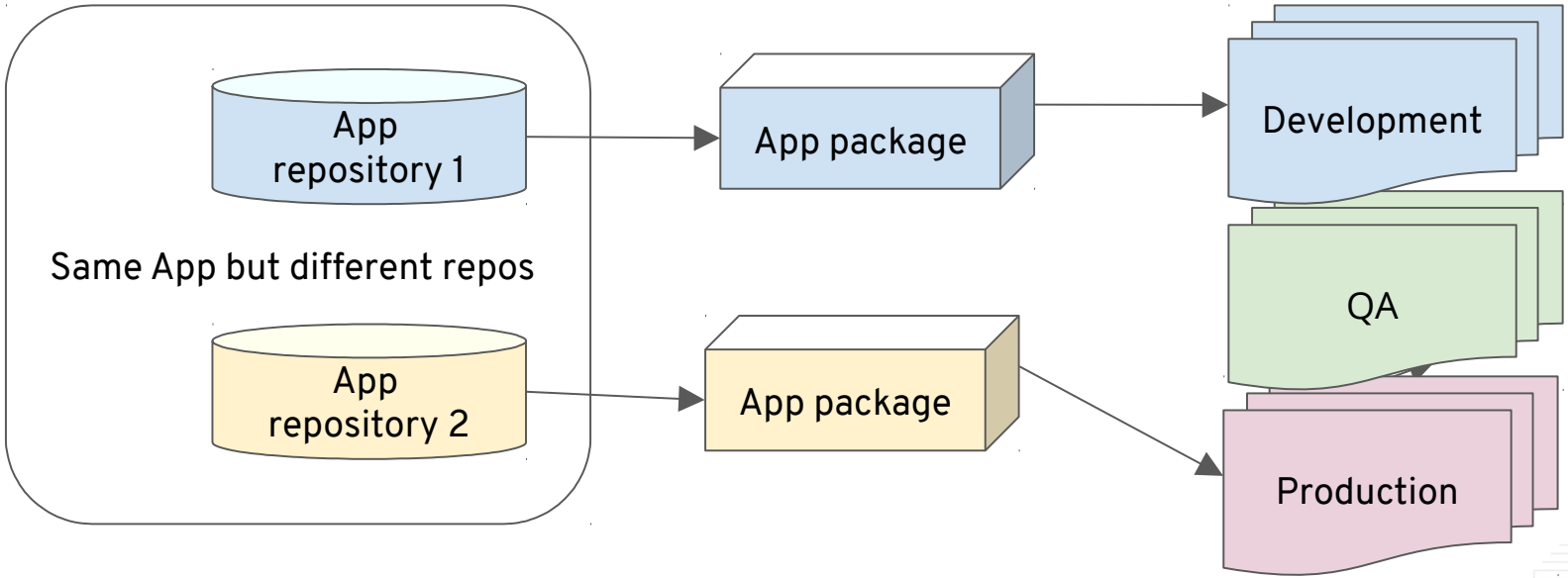
1 - Codebase (What does it mean?)

Use Version Control - But use it the right way!



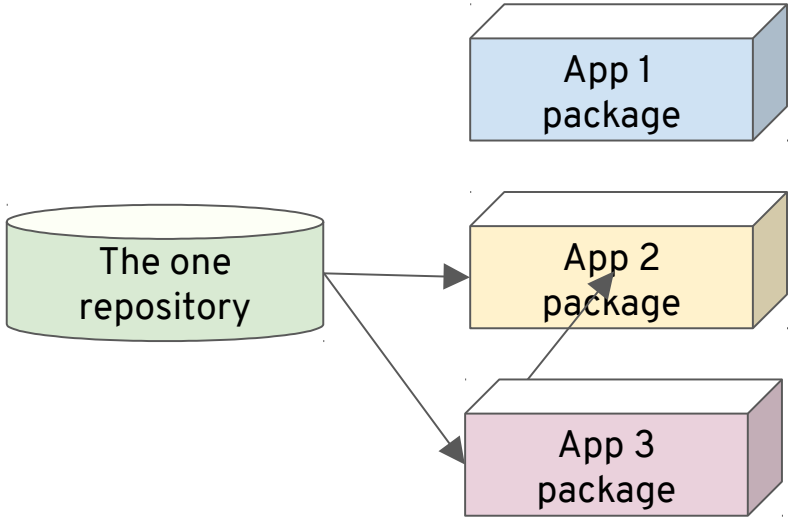
1 - Codebase (DO NOTs)

DO NOT have different codebases for different deployments



1 - Codebase (DO NOTs)

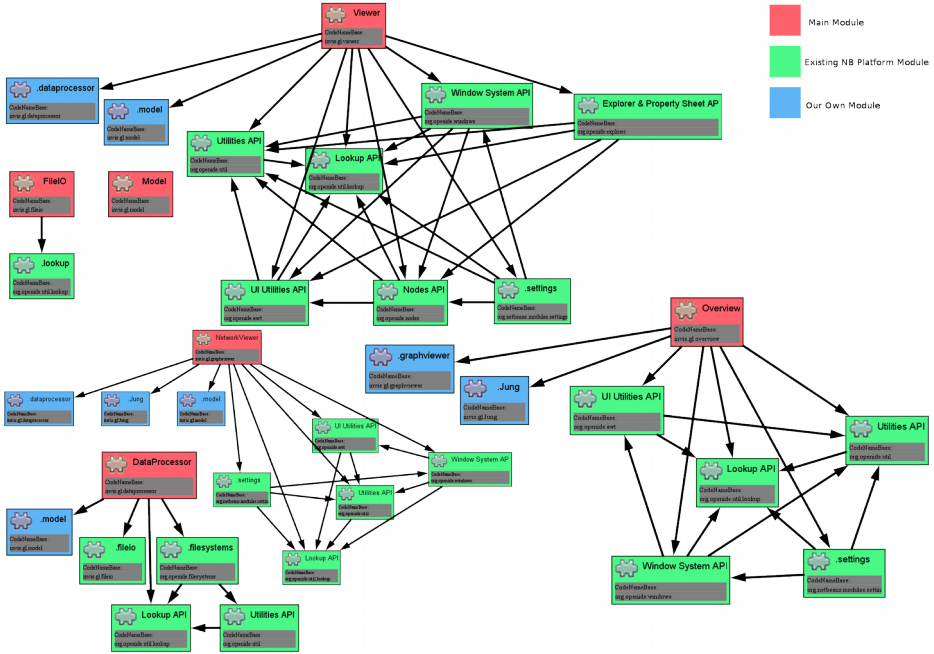
DO NOT have multiple apps and docs in the same repository



DEMO

2 - Dependencies

Explicitly declare and isolate dependencies

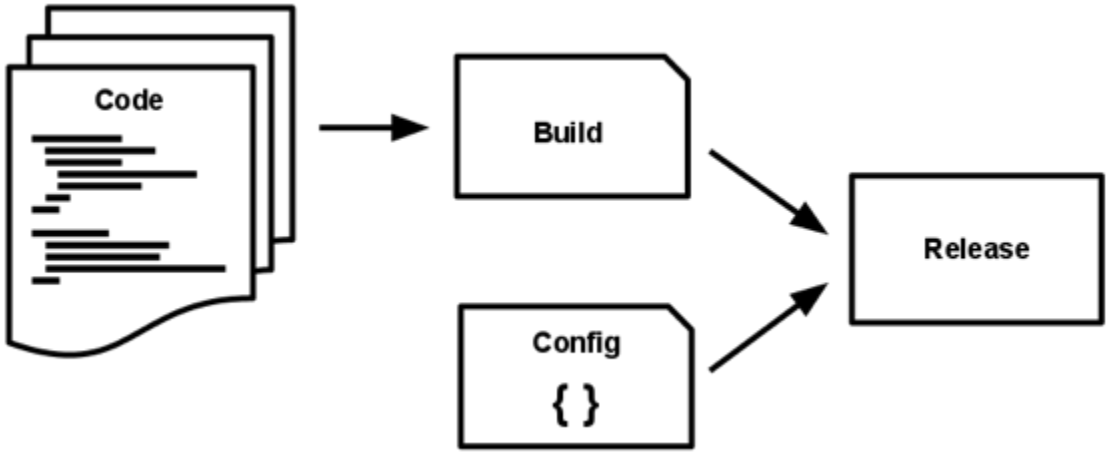


2 - Dependencies (What does it mean?)

- Use a package manager to avoid dependency hell.
- Don't commit dependencies in the codebase repository.

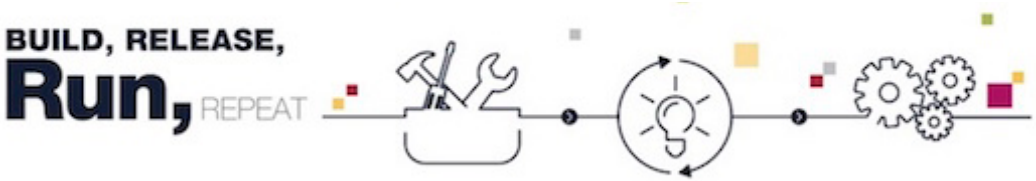
5 - Build, release, run

Strictly separate build and run stages



5 - Build, release, run (What does it mean?)

Use strict separation between the build, release, and run stages.



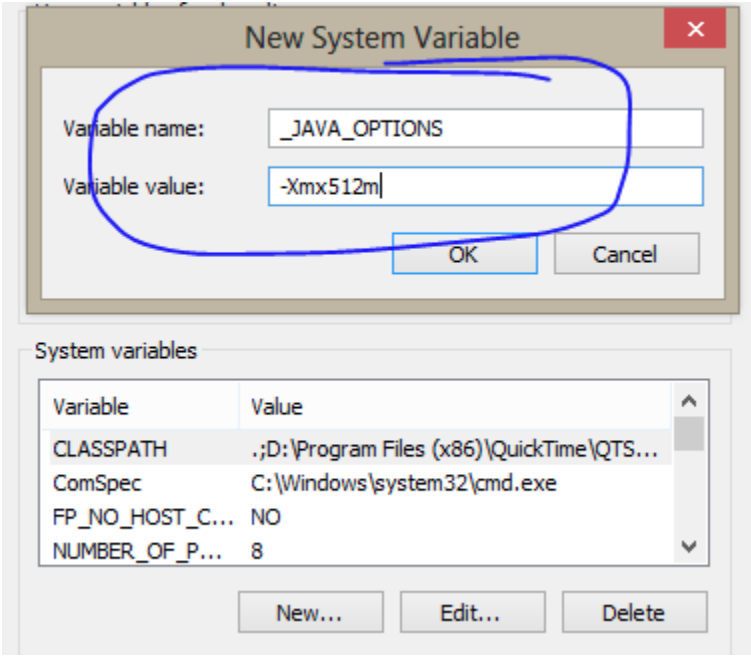
- Every release should always have a unique release ID
- Releases should allow rollbacks

Stage	Who?	What?	Why?
Build	CI	WAR / JAR / etc	Avoid "It works in my machine"
Release	CD	Container image	Deployments / Updates and Rollbacks
Run	Platform	Container instance	Speed, Management, Orchestration

DEMO

3 - Config

Store config in the environment



3 - Config (What does it mean?)

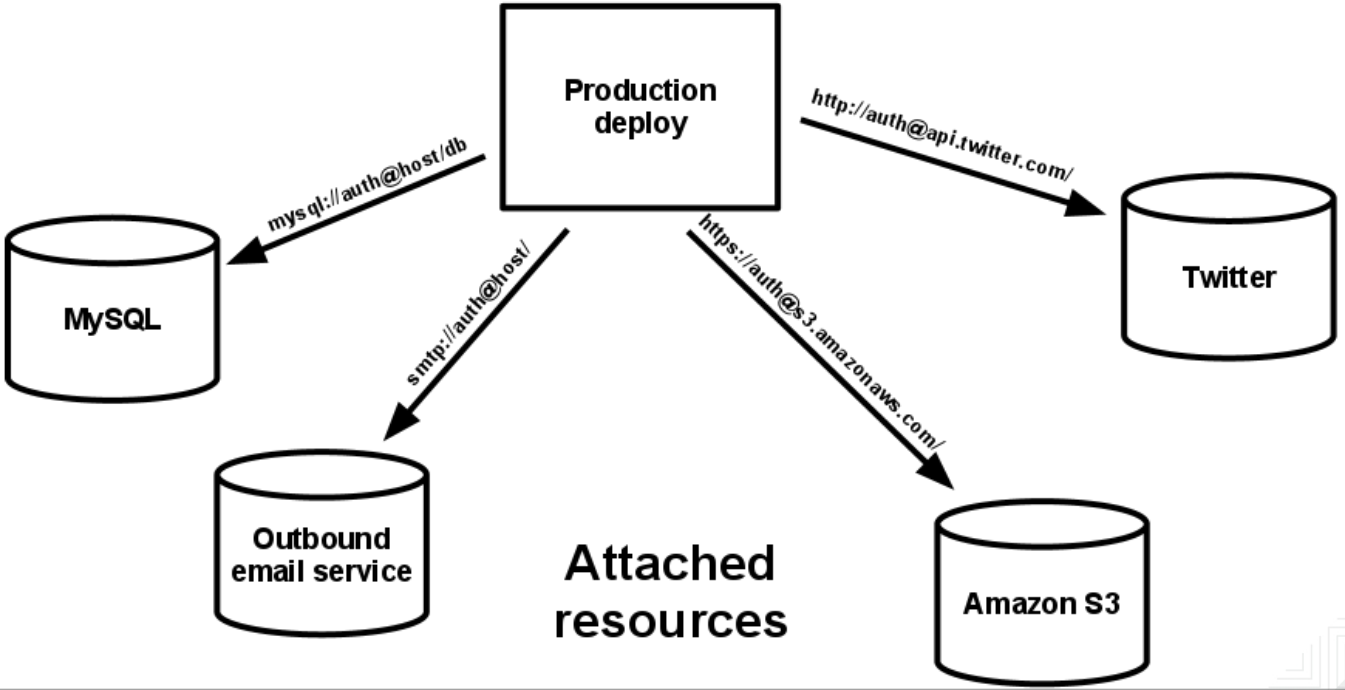
If you have to repackage your application, you're doing it wrong!

Prefer to store the config in Environment Variables

DEMO

4 - Backing Services

Treat backing services as attached resources



6 - Processes (What does it mean?)

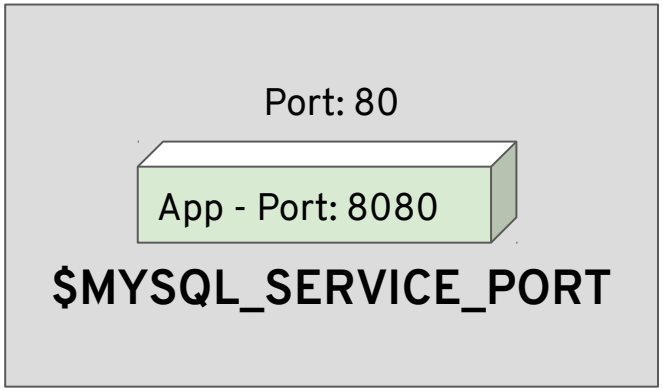
Execute the app as one or more stateless processes

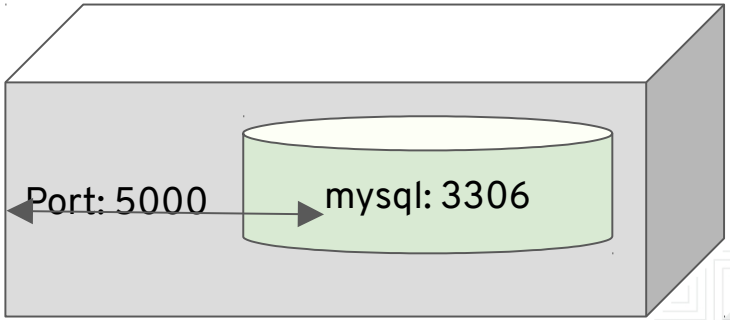
Twelve-factor processes are stateless and share-nothing

7 - Port Binding (What does it mean?)

Export services via port binding

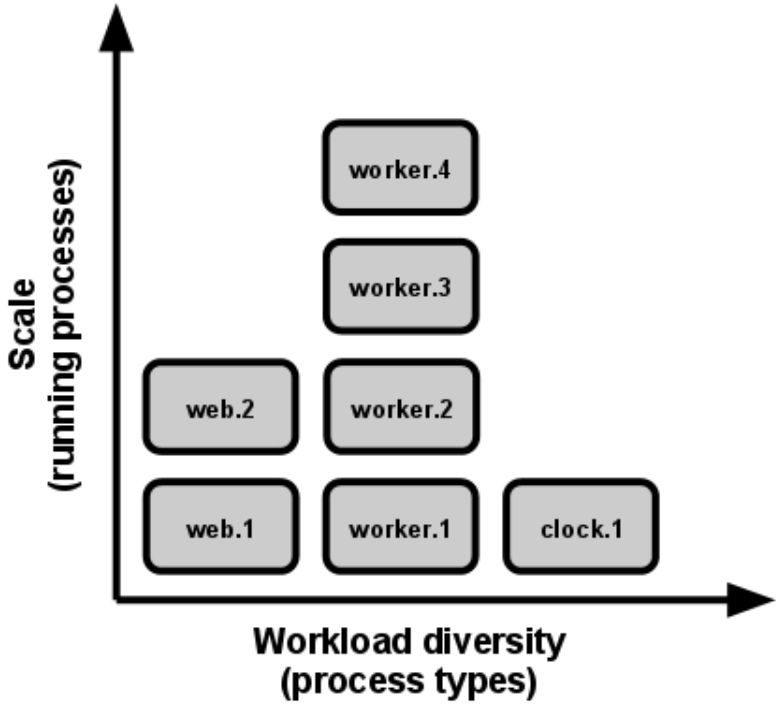
The twelve-factor app is completely self-contained





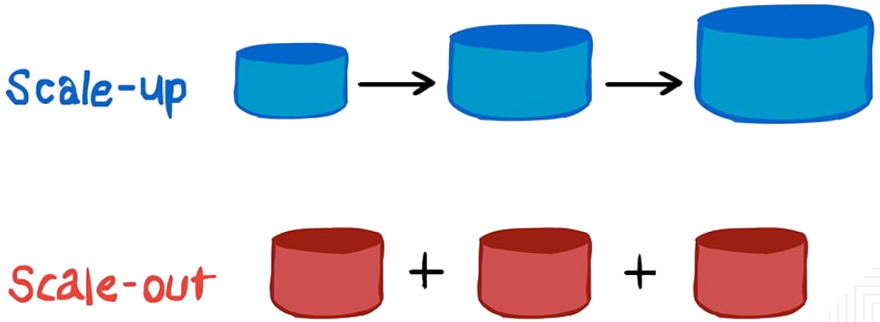
8 - Concurrency

Scale out via the process model



8 - Concurrency

- You can scale up and out
- Scale processes types
- Workload diversity
- It "advocates" for Microservices



DEMO

9 - Disposability

Maximize robustness with fast startup and graceful shutdown

- Processes can be **started or stopped at a moment's notice**
- Processes should **minimize startup time**
- Processes **shutdown gracefully when they receive a SIGTERM**
- Processes **should also be robust against sudden death**

- You cannot **scale, deploy, release, recover fast** if you cannot **start fast**!
- You **cannot start** if you did not **shutdown gracefully**!

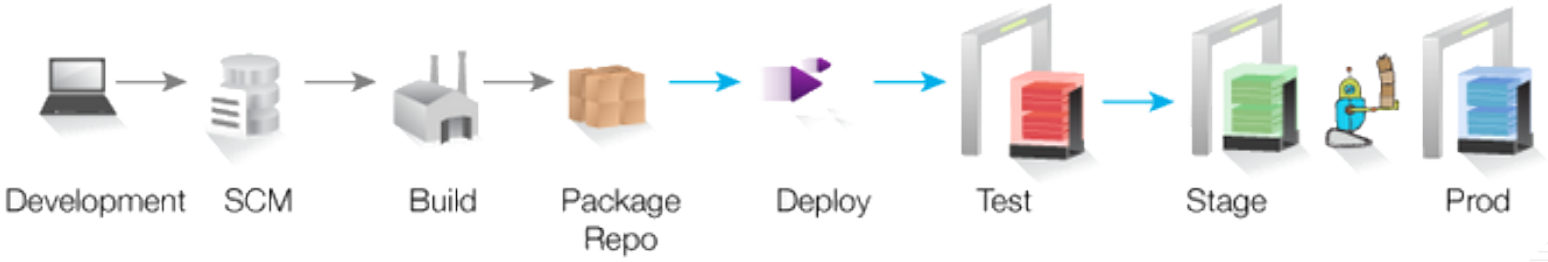
10 - Dev/prod parity

Keep development, staging, and production as similar as possible

Migrating manually directly to staging / production - **not** a great idea.

10 - Dev/prod parity

The twelve-factor app is designed for **continuous deployment** by keeping the gap between development and production small

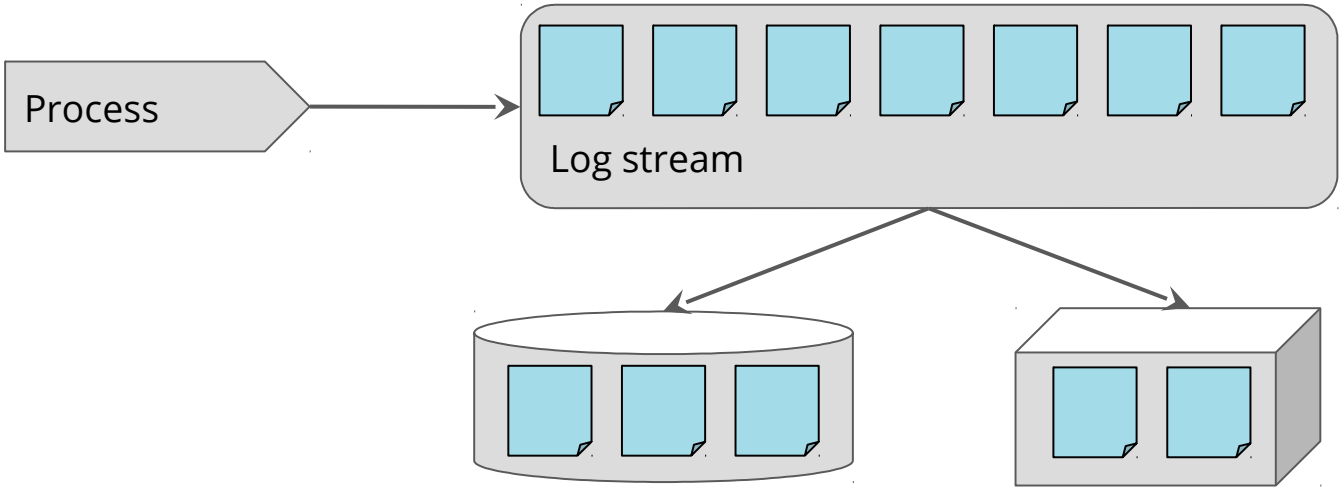


DEMO

11 - Logging

Treat logs as event streams

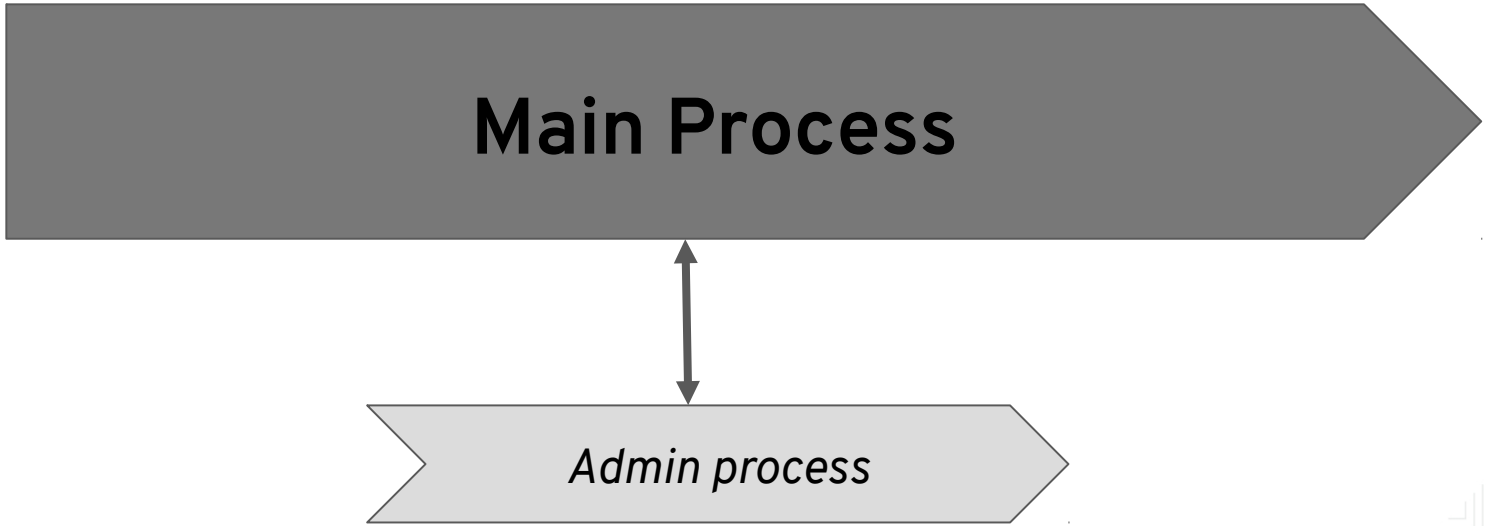
A twelve-factor app **never** concerns itself with **routing** or **storage** of its **output stream**



DEMO

12 - Admin processes

Run admin/management tasks as one-off processes



DEMO

Summary

- Methodology: Technology and language agnostic
- OpenShift: Technology and language agnostic
- But satisfied by
  - Containers,
  - Microservices,
  - and CI/CD Pipelines
- Focused on DevOps

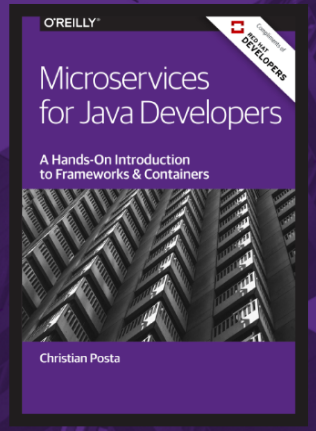
- More info: https://12factor.net/

LOG IN REGISTER ENTER YOUR SEARCH TERM

RED HAT DEVELOPERS

http://developers.redhat.com

TOPICS TECHNOLOGIES COMMUNITY HELP DOWNLOADS



MICROSERVICES FOR JAVA DEVELOPERS:

A hands-on introduction to frameworks and containers.

DOWNLOAD NOW

READ MORE ON MICROSERVICES

- Tear Down Data Silos with Microservices
- Different types of Microservices?
- Scalable Microservices through messaging

Join Red Hat Developers and try it now

Start using .NET on Linux today

Didn't make it to DevNation? Watch Sessions OnDemand

Join Red Hat Developers and try OpenJDK

Getting started, collections, indexes, and dangers. Download now.

RED HAT FORUM

Europe, Middle East & Africa