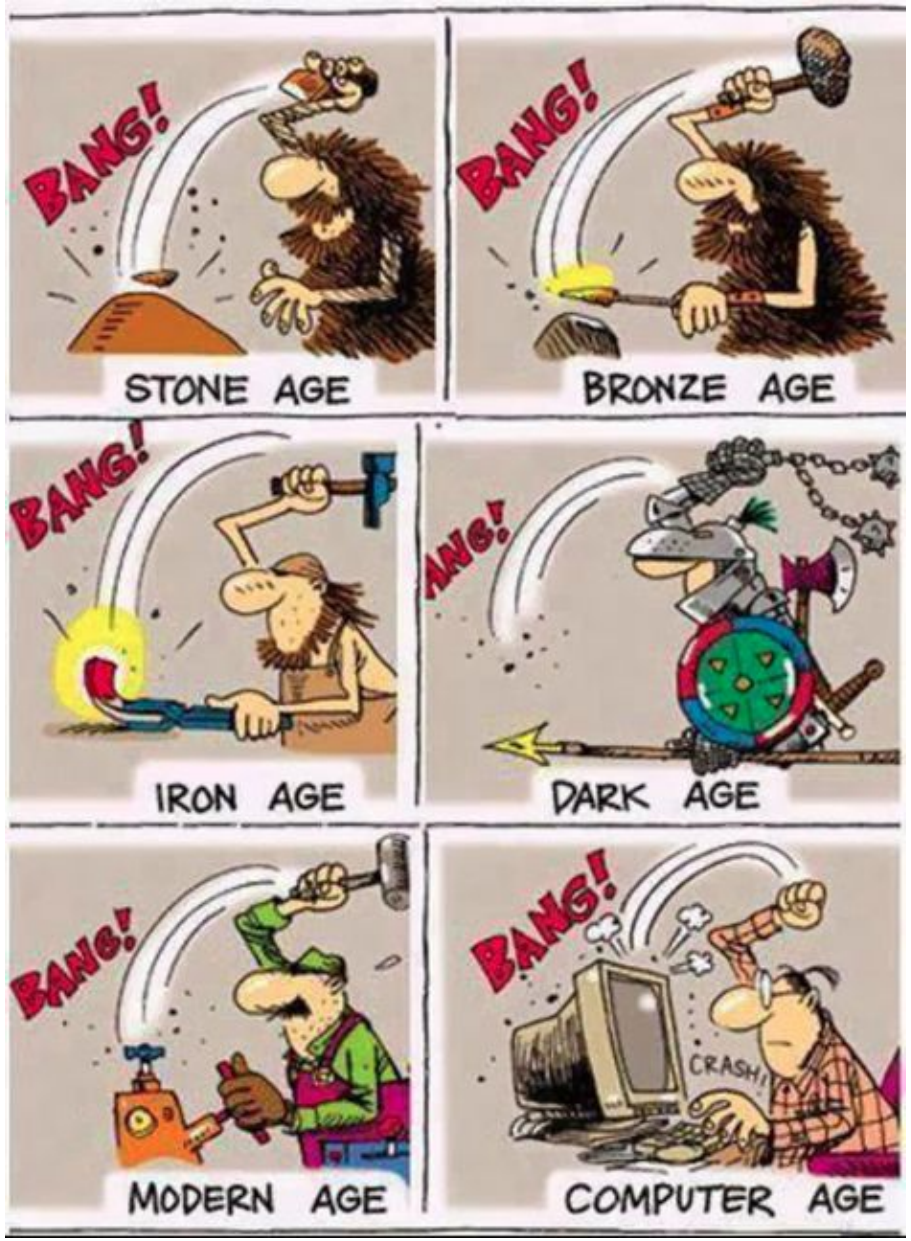
Cloud Native Application Development
The Twelve Factor App Methodology

Robert Baumgartner
Senior Solution Architect

Red Hat

Humans had been problem solvers…

until they've decided to become PROGRAMMERS



The Twelve-Factor App

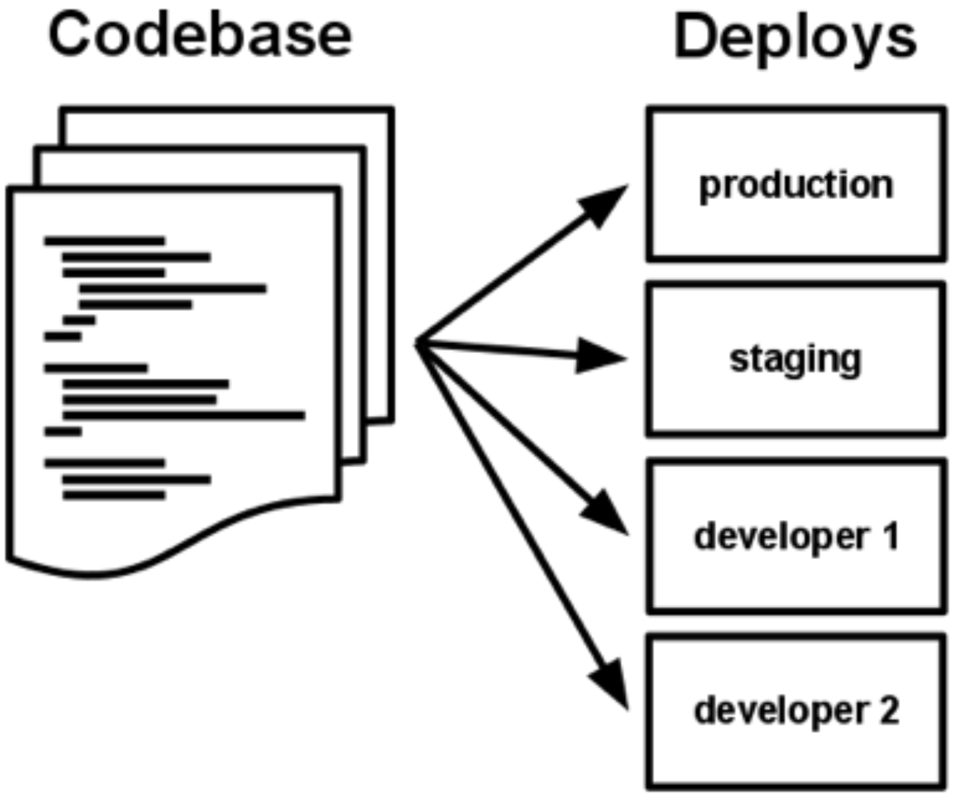
- **A methodology**
- Manifesto
- Best practices
- Principles

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

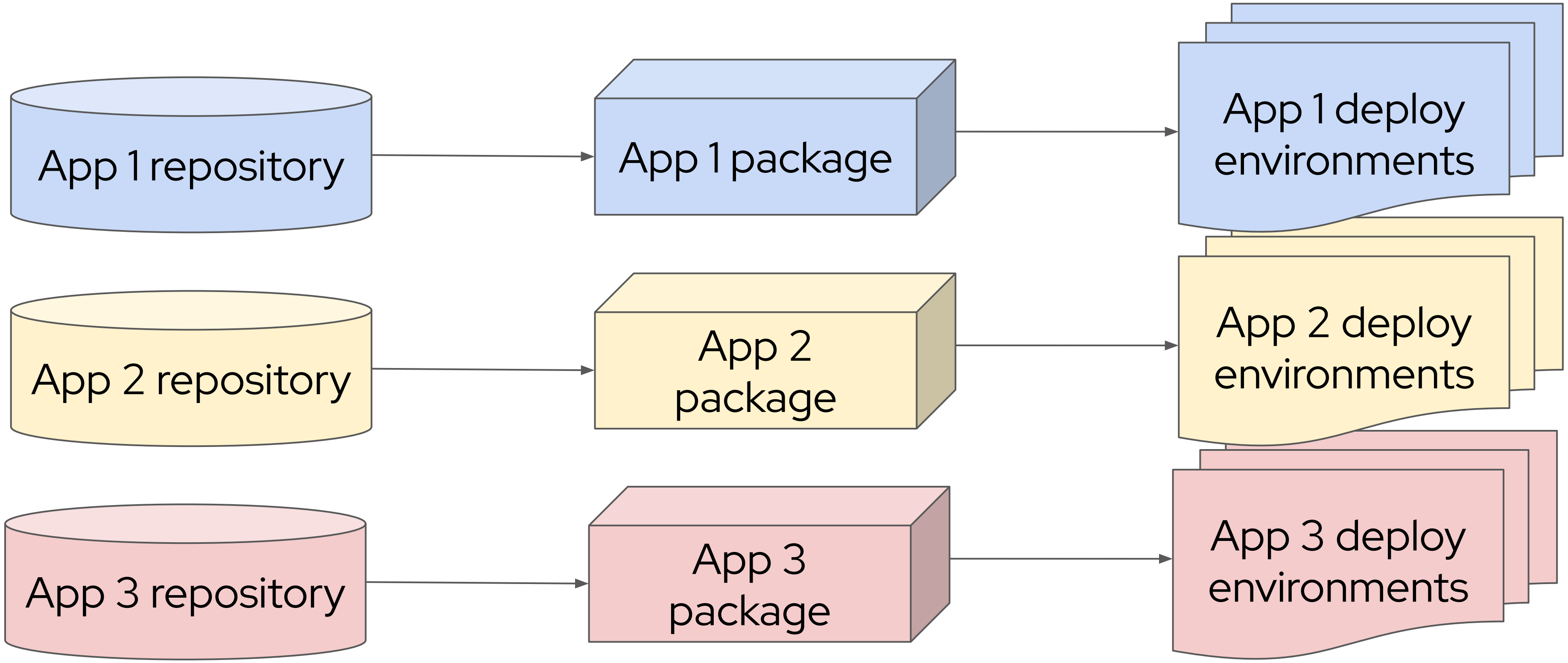
One codebase tracked in revision control, many deploys



One codebase = one app

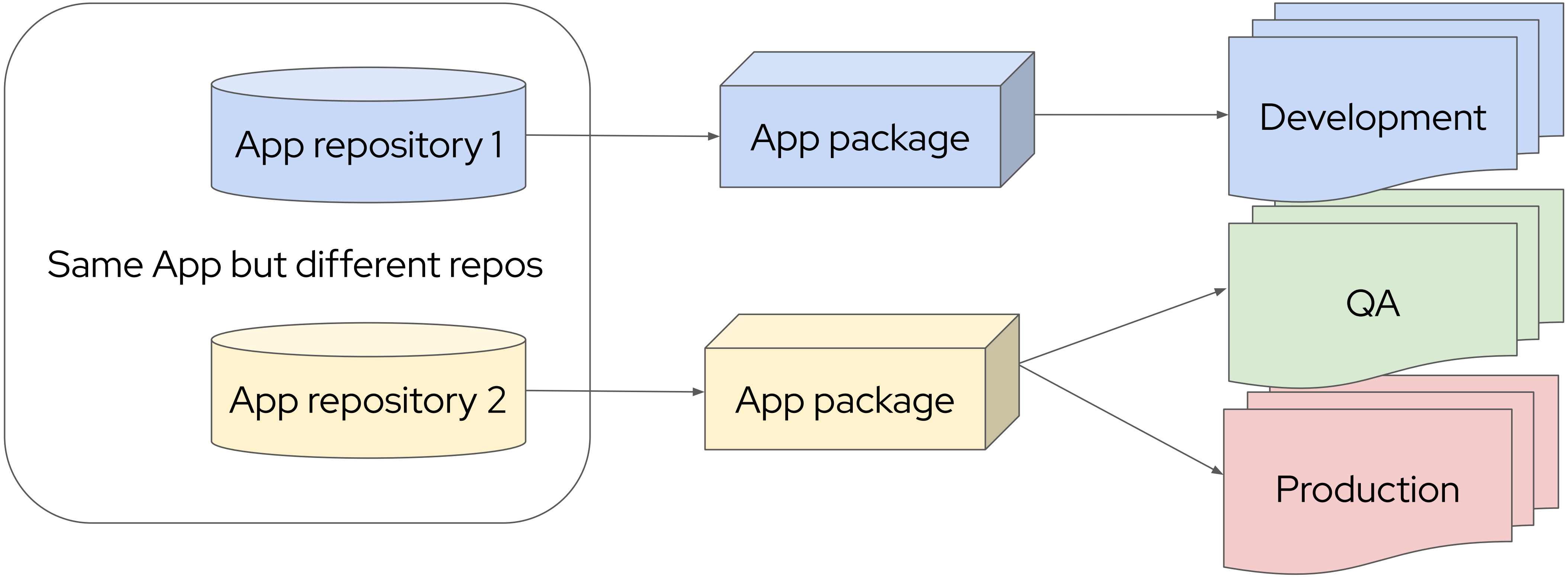
1 **Codebase** (What does it mean?)

Use Version Control - But use it the right way!



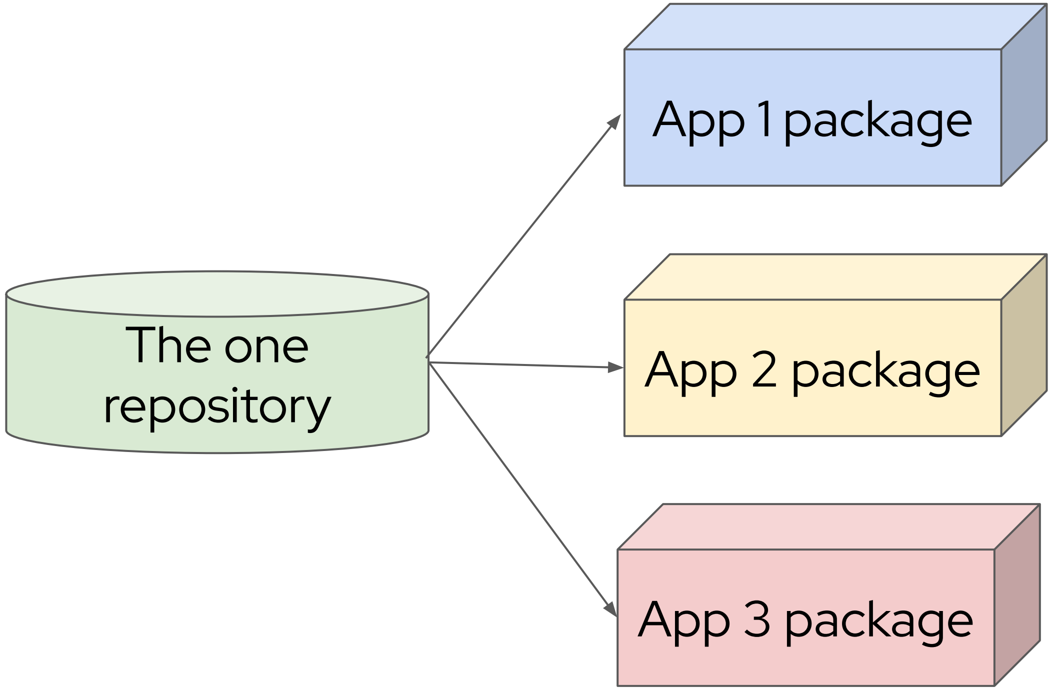
1 **Codebase** (DO NOTs)

DO NOT have different codebases for different deployments



1 **Codebase** (DO NOTs)

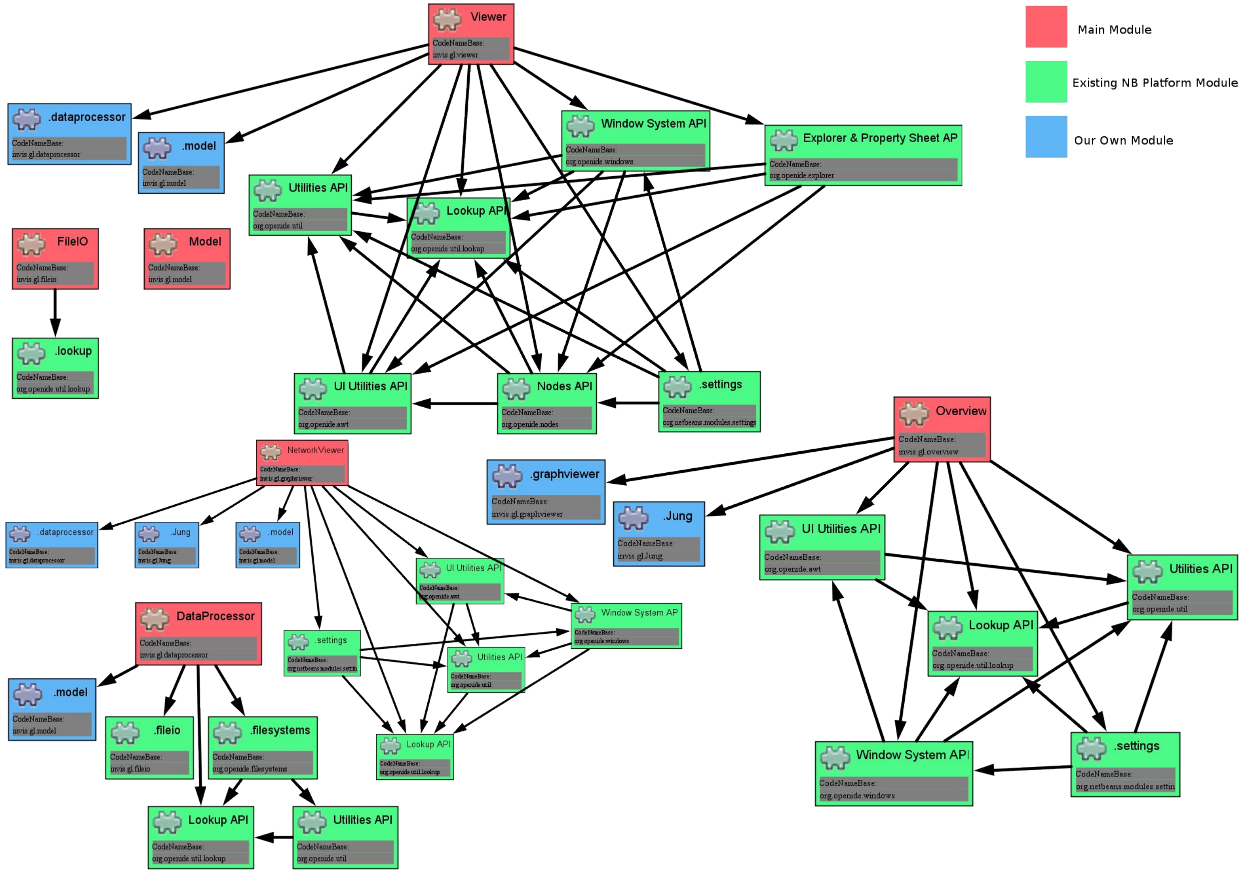
DO NOT have multiple apps and docs in the same repository



DEMO

2 Dependencies

Explicitly declare and isolate dependencies


Viewer
CodeNameBase: invis.gl.viewer
.dataprocessor
CodeNameBase: invis.gl.dataprocessor
.model
CodeNameBase: invis.gl.model
Window System API
CodeNameBase: org.openide.windows
Explorer & Property Sheet AP
CodeNameBase: org.openide.explorer
Utilities API
CodeNameBase: org.openide.util
Lookup API
CodeNameBase: org.openide.util.lookup
FileIO
CodeNameBase: invis.gl.fileio
Model
CodeNameBase: invis.gl.model
.lookup
CodeNameBase: org.openide.util.lookup
UI Utilities API
CodeNameBase: org.openide.awt
Nodes API
CodeNameBase: org.openide.nodes
.settings
CodeNameBase: org.netbeans.modules.settings
Overview
CodeNameBase: invis.gl.overview
NetworkViewer
CodeNameBase: invis.gl.graphviewer
.graphviewer
CodeNameBase: invis.gl.graphviewer
.Jung
CodeNameBase: invis.gl.Jung
.dataprocessor
CodeNameBase: invis.gl.dataprocessor
.Jung
CodeNameBase: invis.gl.Jung
.model
CodeNameBase: invis.gl.model
UI Utilities API
CodeNameBase: org.openide.awt
Window System AP
CodeNameBase: org.openide.windows
.settings
CodeNameBase: org.netbeans.modules.settin
Utilities API
CodeNameBase: org.openide.util
Lookup API
CodeNameBase: org.openide.util.lookup
DataProcessor
CodeNameBase: invis.gl.dataprocessor
.model
CodeNameBase: invis.gl.model
.fileio
CodeNameBase: invis.gl.fileio
.filesystems
CodeNameBase: org.openide.filesystems
Lookup API
CodeNameBase: org.openide.util.lookup
Utilities API
CodeNameBase: org.openide.util
UI Utilities API
CodeNameBase: org.openide.awt
Utilities API
CodeNameBase: org.openide.util
Lookup API
CodeNameBase: org.openide.util.lookup
Window System API
CodeNameBase: org.openide.windows
.settings
CodeNameBase: org.netbeans.modules.settin
Main Module
Existing NB Platform Module
Our Own Module

2 **Dependencies** (What does it mean?)

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

npm

Maven™

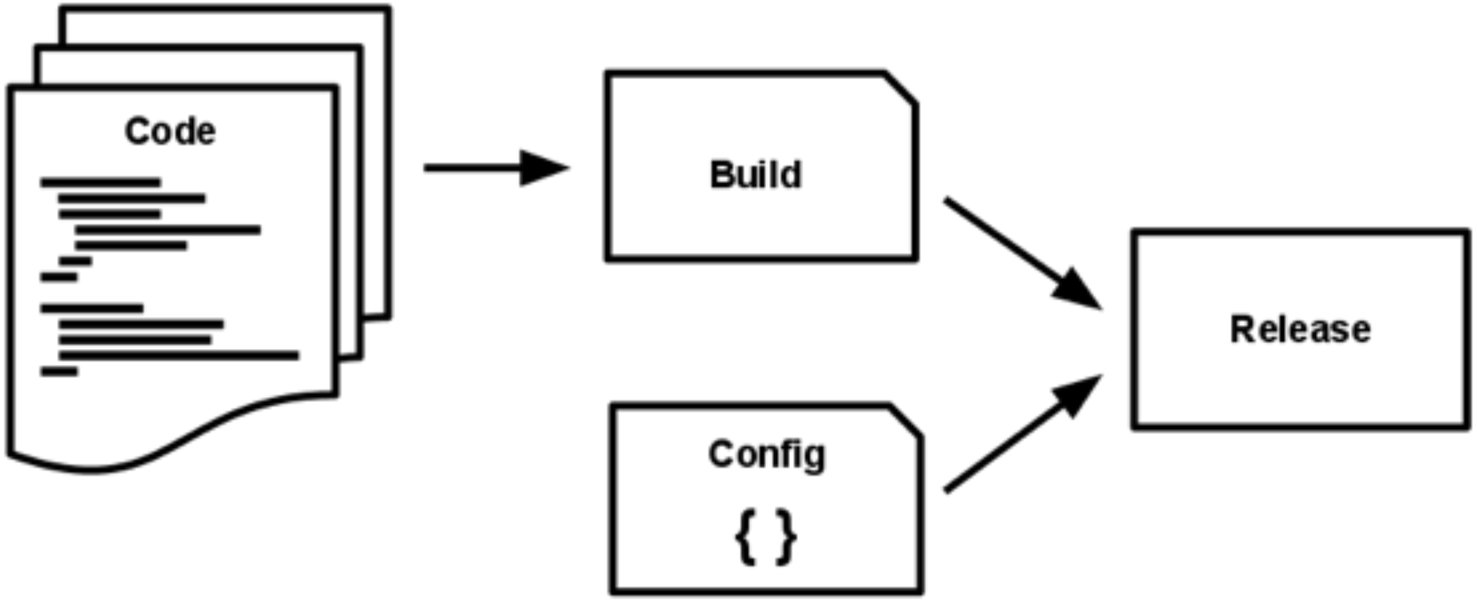
CPAN

gems
Ruby

DEMO

5 Build, release, run

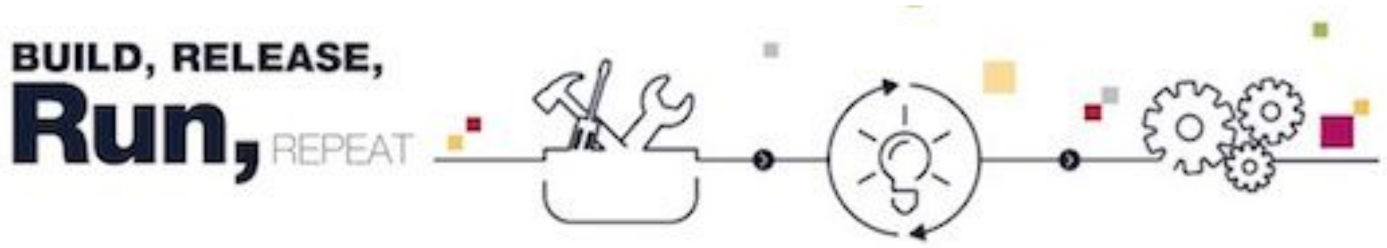
Strictly separate build and run stages



Red Hat

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

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



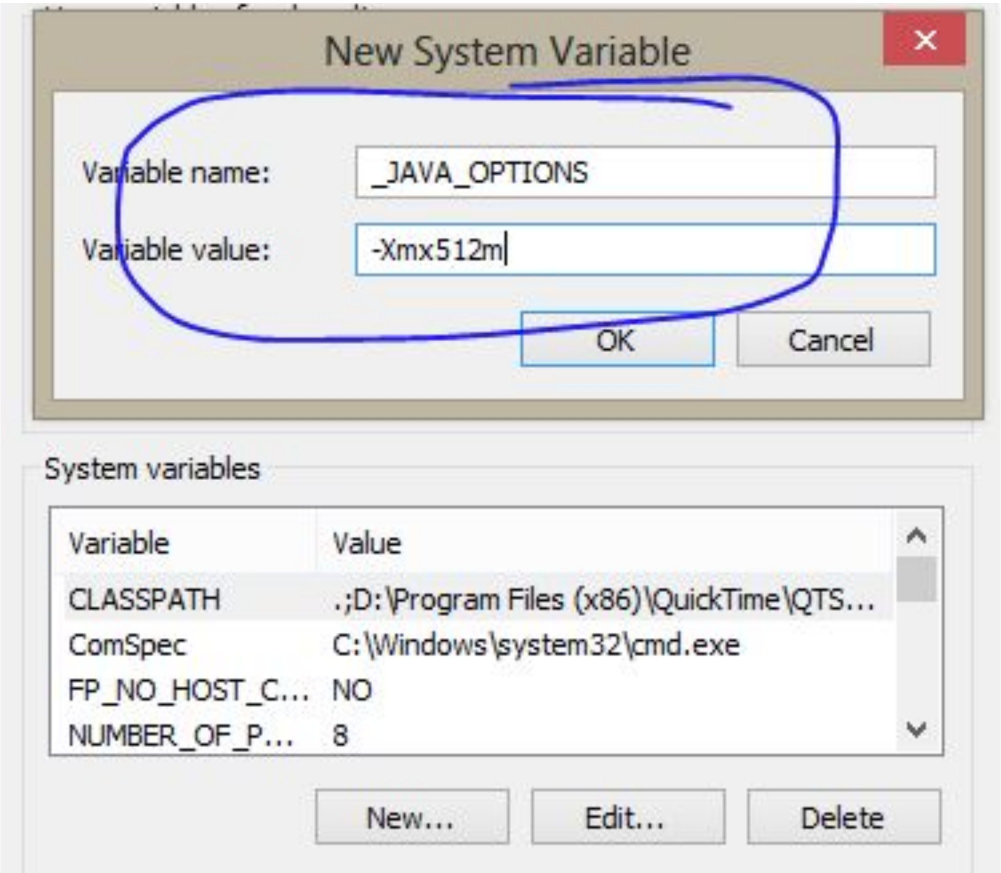
- Every release should always have a unique release ID
- Releases should allow rollbacks

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



New System Variable

Variable name: _JAVA_OPTIONS

Variable value: -Xmx512m

System variables

Variable	Value
CLASSPATH	.;D:\Program Files (x86)\QuickTime\QTS...
ComSpec	C:\Windows\system32\cmd.exe
FP_NO_HOST_C...	NO
NUMBER_OF_P...	8

3 **Config** (What does it mean?)

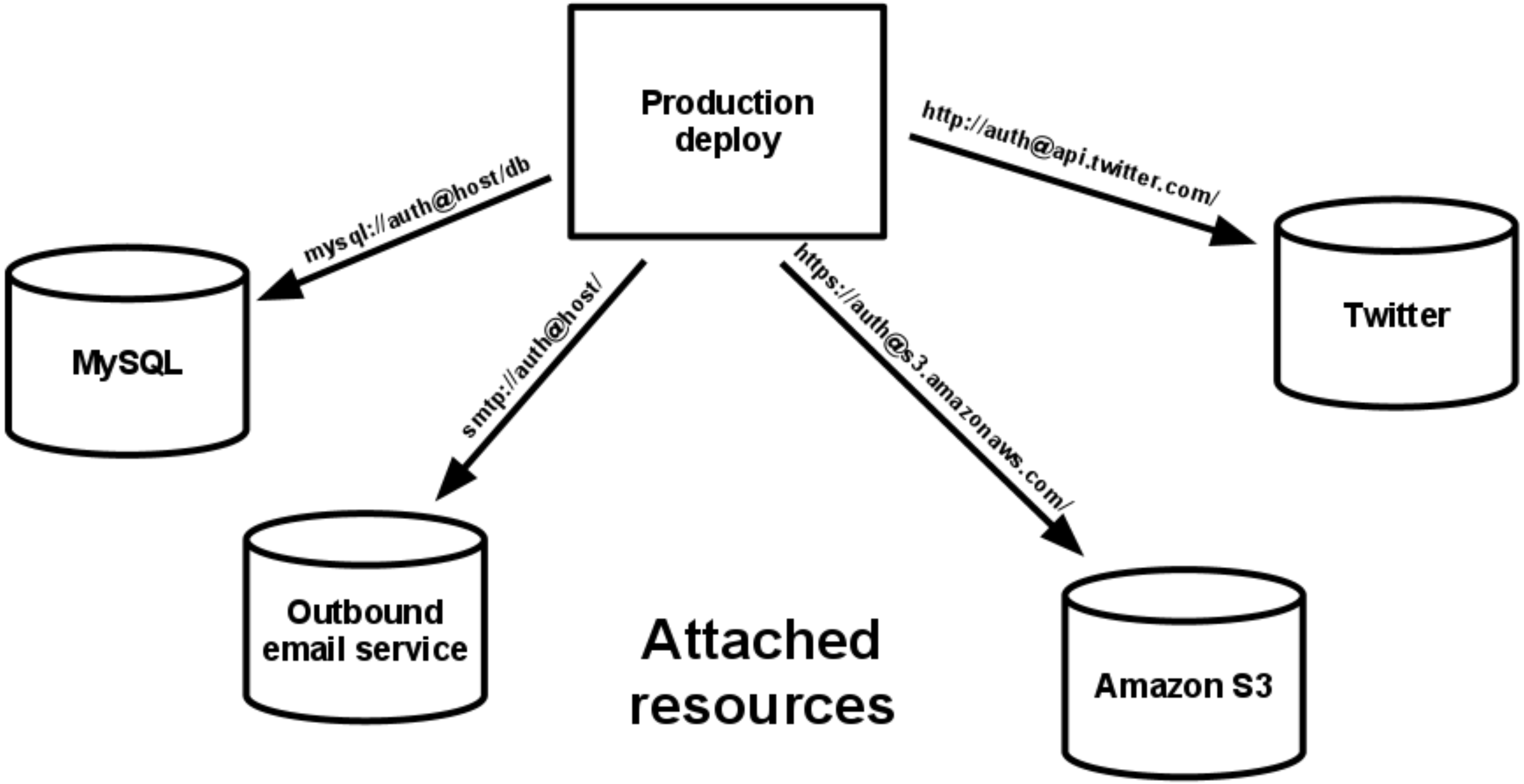
If you have to repackage your application, you're doing it wrong!

Prefer to store the config in Environment Variables

DEMO

4 Backing Services

Treat backing services as attached resources



DEMO

6 **Processes** (What does it mean?)

Execute the app as one or more stateless processes

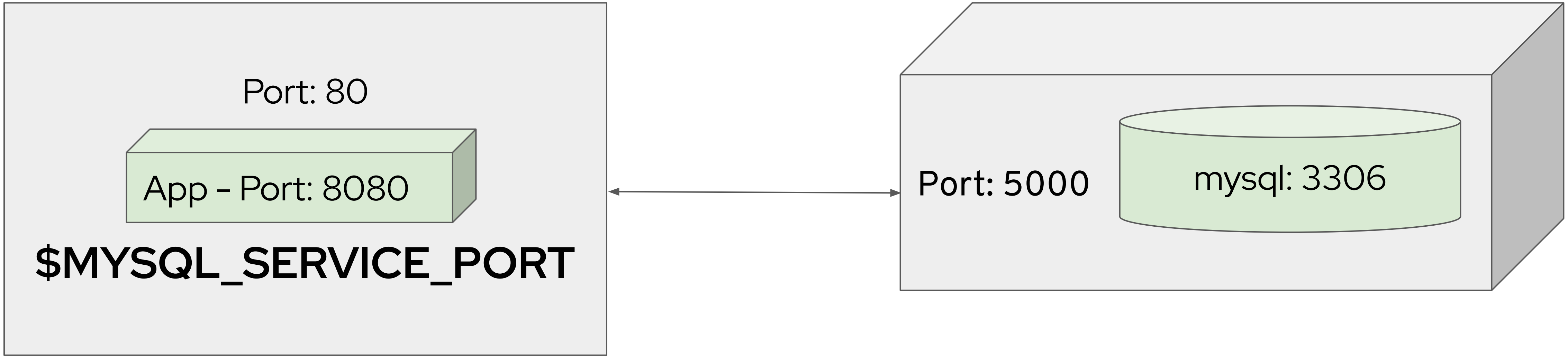
API
FIRST

Twelve-factor processes are stateless and share-nothing

7 **Port Binding** (What does it mean?)

Export services via port binding

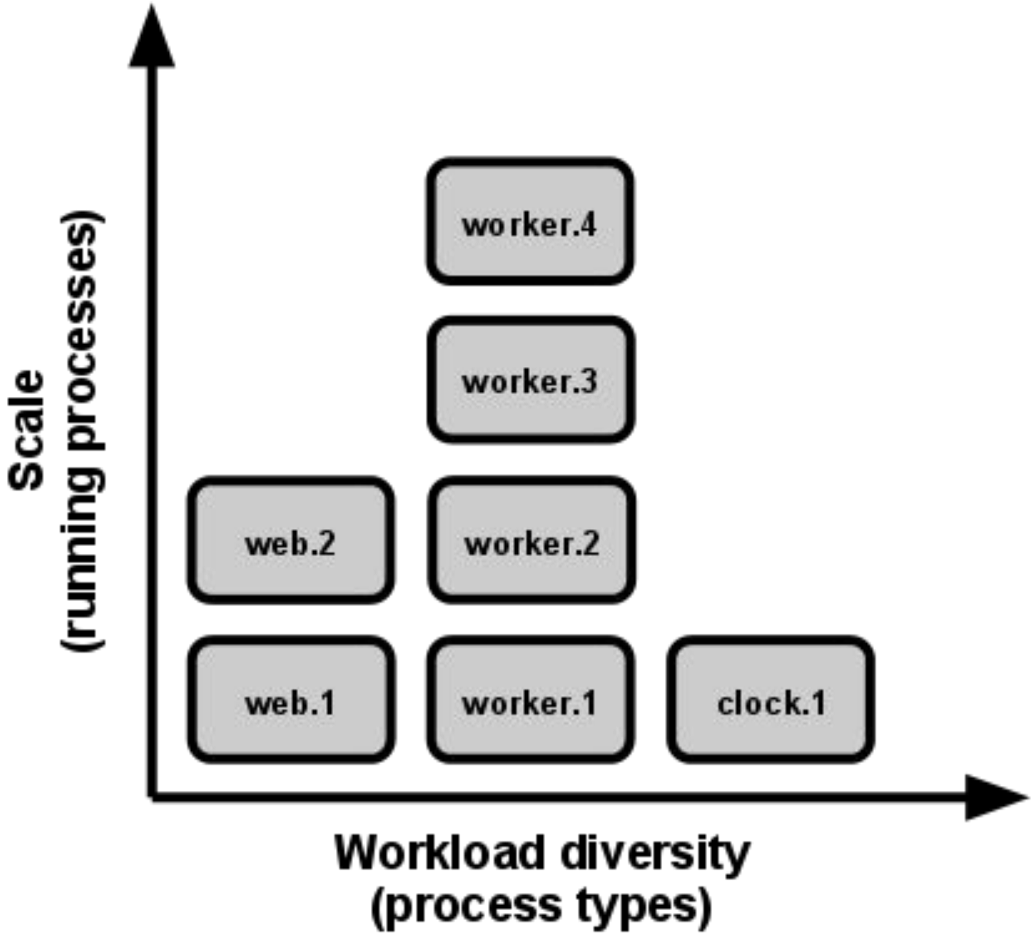
The twelve-factor app is completely self-contained



DEMO

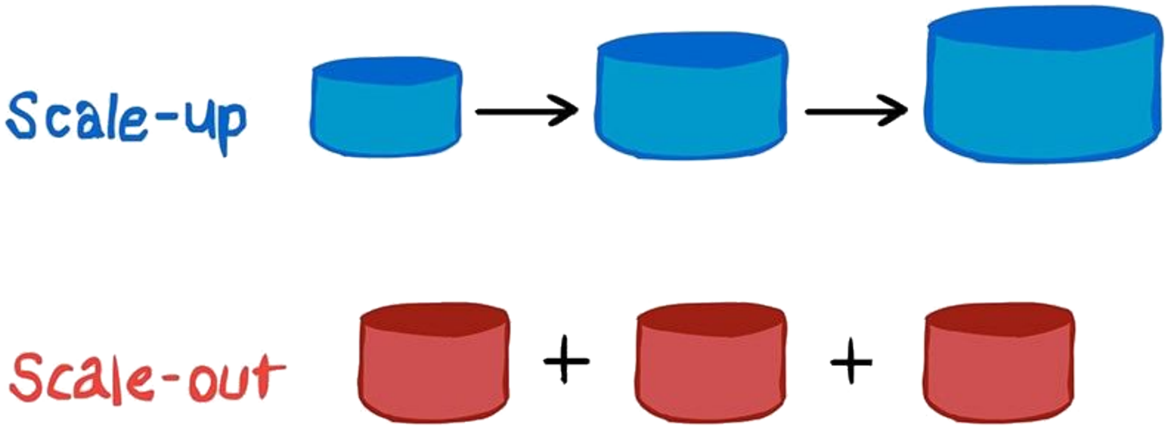
8 Concurrency

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**!

DEMO

10 Dev/prod parity

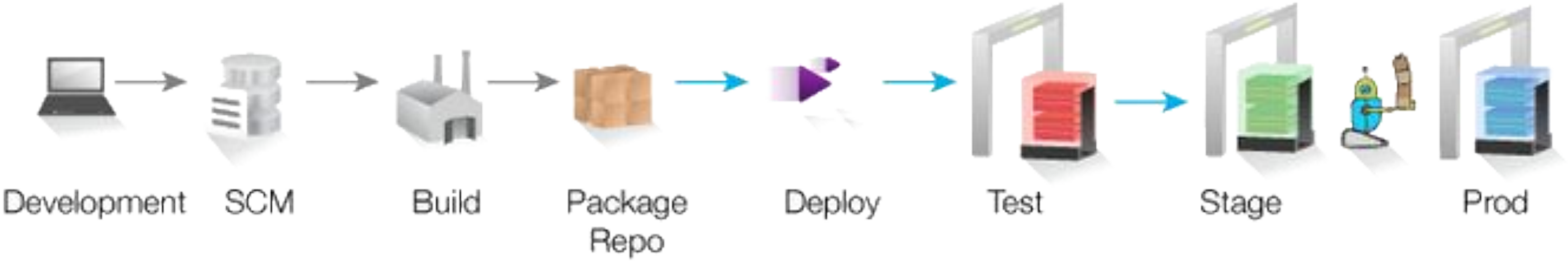
Keep development, staging, and production as similar as possible

WORKS ON MY MACHINE!
BAD IDEA
Staging / Production Environment

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

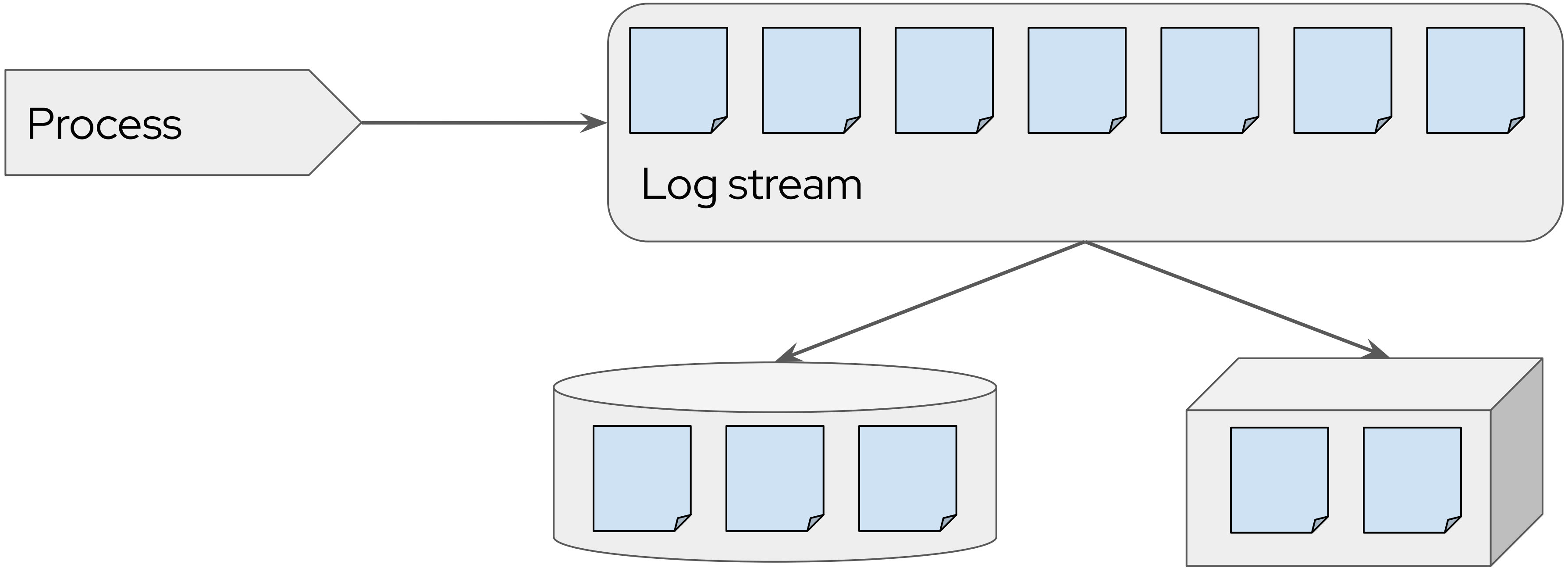


DEMO

11 Logging

Treat logs as event streams

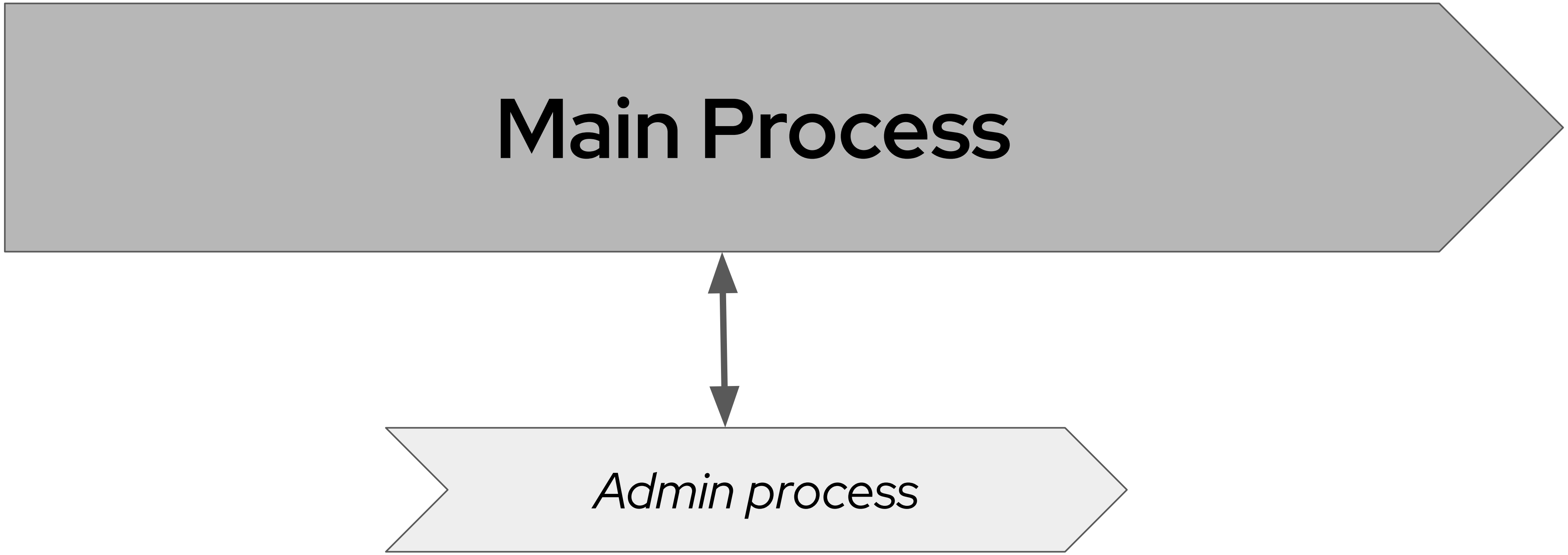
A twelve-factor app **never** concerns itself with **routing** or **storage** of its **output stream**



DEMO

12 Admin processes

Run admin/management tasks as one-off processes



DEMO

Summary

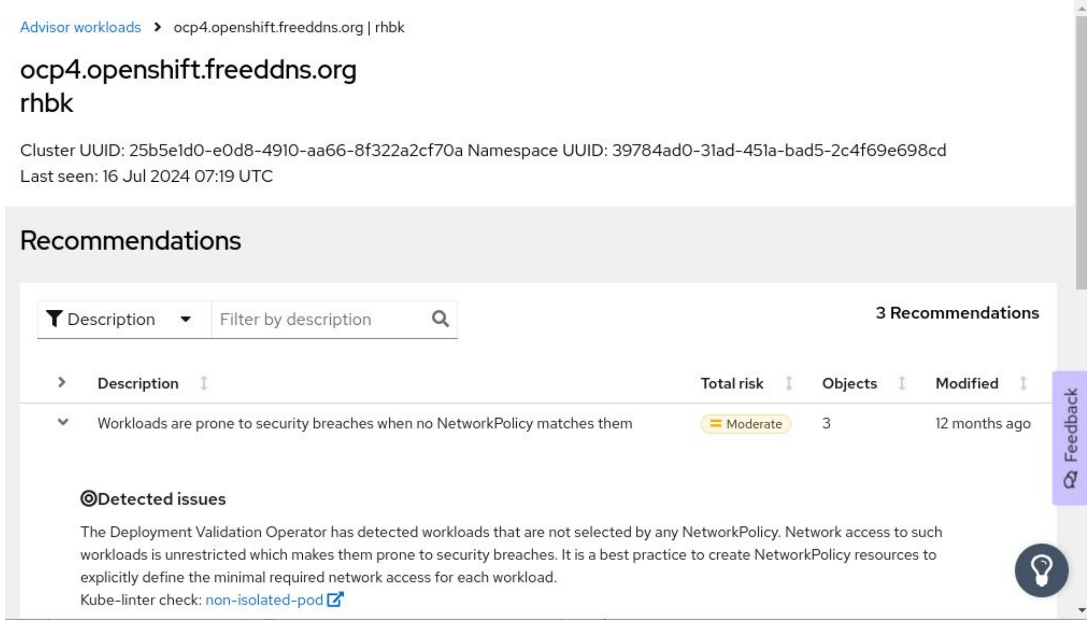
- Methodology: Technology and language agnostic
- OpenShift: Technology and language agnostic
- But satisfied by
  - Containers,
  - Microservices,
  - and CI/CD Pipelines
- Focused on DevOps

- More info: https://12factor.net/
- Demo app: https://github.com/rbaumgar/12factor-app

Red Hat

Deployment Validation Operator

- Not setting resource requests
- Not adding application health checks
- Overprivileged applications
- Lack of monitoring/logging
- Using default namespaces for all objects
- Missing/Wrong security configurations
- Missing or improper pod disruption budget definitions



Advisor workloads › ocp4.openshift.freeddns.org | rhbk

ocp4.openshift.freeddns.org
rhbk

Cluster UUID: 25b5e1d0-e0d8-4910-aa66-8f322a2cf70a Namespace UUID: 39784ad0-31ad-451a-bad5-2c4f69e698cd
Last seen: 16 Jul 2024 07:19 UTC

Recommendations

Description ▾ | Filter by description 🔍 — 3 Recommendations

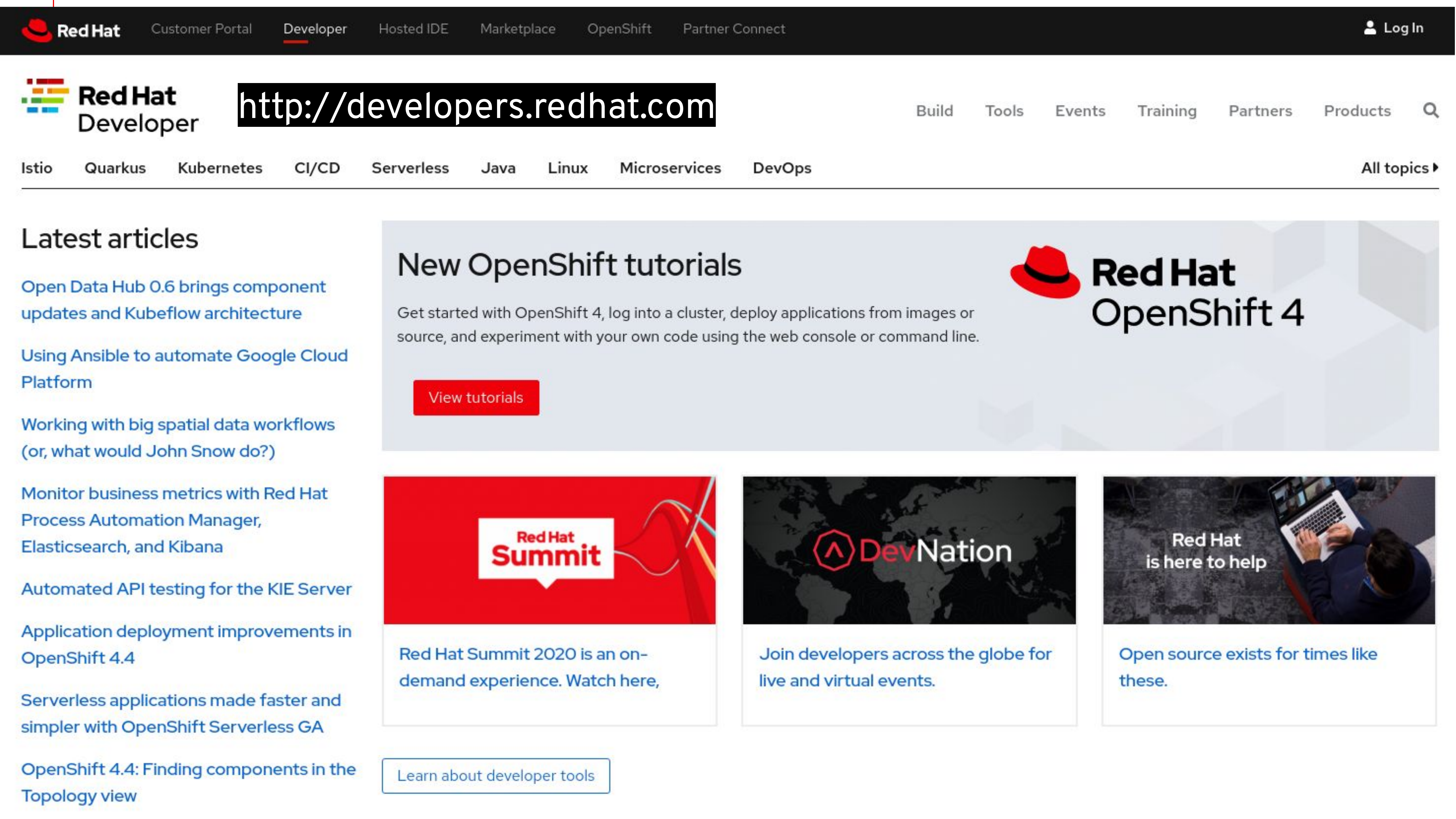
Description	Total risk	Objects	Modified
Workloads are prone to security breaches when no NetworkPolicy matches them	Moderate	3	12 months ago

Detected issues

The Deployment Validation Operator has detected workloads that are not selected by any NetworkPolicy. Network access to such workloads is unrestricted which makes them prone to security breaches. It is a best practice to create NetworkPolicy resources to explicitly define the minimal required network access for each workload.
Kube-linter check: non-isolated-pod

https://console.redhat.com/openshift/insights/advisor/workloads/

https://www.redhat.com/en/blog/boost-your-cluster-operations-with-deployment-validation-and-insights-advisor-for-workloads http://developers.redhat.com



Latest articles

Open Data Hub 0.6 brings component updates and Kubeflow architecture

Using Ansible to automate Google Cloud Platform

Working with big spatial data workflows (or, what would John Snow do?)

Monitor business metrics with Red Hat Process Automation Manager, Elasticsearch, and Kibana

Automated API testing for the KIE Server

Application deployment improvements in OpenShift 4.4

Serverless applications made faster and simpler with OpenShift Serverless GA

OpenShift 4.4: Finding components in the Topology view

New OpenShift tutorials

Get started with OpenShift 4, log into a cluster, deploy applications from images or source, and experiment with your own code using the web console or command line.

Red Hat Summit 2020 is an on-demand experience. Watch here,

Join developers across the globe for live and virtual events.

Open source exists for times like these.

Thank you

in linkedin.com/company/Red-Hat

f facebook.com/RedHatinc

You Tube youtube.com/user/RedHatVideos

twitter.com/RedHat

Red Hat