InterConnect 2016
The Premier Cloud & Mobile Conference

Creating 12-Factor Applications with WAS Liberty on Bluemix

A Practical Guide

Erin Schnabel
schnabel@us.ibm.com
@ebullientworks

February 21 – 25
MGM Grand & Mandalay Bay
Las Vegas, Nevada

What are Twelve Factor applications?

- “a methodology for building software-as-a-service applications”
  - Created by the developers at Heroku
- The Twelve Factors can be applied to applications
  - In any programming language
  - With any backing services (or cloud provider..)
- http://12factor.net/

Why should you care?

- The 12 Factor app is a *methodology* for building applications that:
  - Use declarative formats for setup automation
  - Have a clean contract with the underlying OS
  - Are suitable for cloud deployment
  - Has minimum divergence between development and test environments
  - Can scale up without significant work

- Ultimately, it is **portable**, perfect for running **in the cloud** and can be maintained in a **continuous delivery pipeline**

  ➔ **MICROSERVICES**

12-Factors + Liberty = ♥

- Twelve-Factor Application:
  - https://github.com/WASdev/sample.microservices.12factorapp
  - http://wasdev.net/docs/creating-a-12-factor-application-with-was-liberty
- Simple application built to demonstrate all of the factors
- Application sample is:
  - A war with code, and
  - A packaged liberty server with confg

Game ON!

- Microservices-based application
- Enables developers to experiment with microservices concepts

- Core set of services written in Java, JavaScript

- Java services built upon the 12-factor sample as a base

- http://game-on.org
- http://wasdev.net

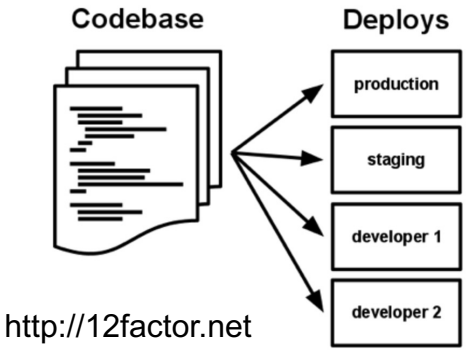
The 12 Factors

Factor 1 - Codebase

- "One codebase tracked in revision control, many deploys"-12factor.net
- Key points
  - One-to-one correlation between codebase and the app
    - 'If there are multiple codebases, it's not an app – it's a distributed system"
  - Codebase is the same across environments: dev, staging, production

- Options
  - Git, Subversion, Mercurial, …

- Does this really mean one repository per service?



http://12factor.net

Factor 2 - Dependencies

- “Explicitly declare and isolate dependencies”-12factor.net

- Key points:
  - App declares all dependencies, completely and exactly
  - App does not rely on “pre-requisite” system-wide packages
  - Use dependency isolation tool, e.g. gradlew

- Implementation:
  - Docker vs. CF
  - gradlew for ensuring the correct version of gradle

Factor 3 - Config

- “Store config in the environment” -12factor.net

- Key points:
  - Config includes anything that can vary between deploys
  - Does not include internal application config – e.g. features in Liberty

- How we implemented this:
  - Environment variables for Containers
  - For Bluemix: environment variables via a yaml file or VCAP_SERVICES

Factor 4 - Backing Services

- “Treat backing services as attached resources”-12factor.net
- Key points:
  - Backing services → Datastore, Watson, ….
  - Resources can be attached and detached at will

- How we implemented this:
  - Location of the database stored in environment variables
    - VCAP_SERVICES or Bluemix environment yaml file
    - Fetched from etcd as config store

Factor 5 – Build, release, run

- “Strictly separate build and run stages”-12factor.net

- Key points:
  - Strict separation between the three stages: Build, Release, Run
  - e.g. no code changes at runtime

- How we implemented this:
  - Build pipeline! Git commit to master → build → live!
  - Local: gradle/maven + docker-compose

Factor 6 - Processes

- “Execute the app as one or more stateless processes”-12factor.net

- Key points:
  - 12-Factor processes are stateless and share-nothing
  - Never assume that anything cached will be available on a future request

- Options
  - Use a datastore or shared cache to store data that needs persisting
  - Provides advantages in terms of scaling

Factor 7 – Port Binding

- “Export services via port binding”-12factor.net

- Key points:
  - App is completely self-contained
  - “…The web app exports HTTP as a service by binding to a port,…” ***
  - Host and port should be provided by the environment

- Options
  - Liberty server variables: Bluemix routing vs. local environment
  - Liberty server package is an ideal self-contained unit.

Factor 8 - Concurrency

- “Scale out via the process model”-12factor.net

- Key points:
  - Recommends splitting processes based on the type of work
    - Request-driven (HTTP) vs. long running / background tasks
  - Scale by making more processes

- Our implementation:
  - Use Bluemix for scaling – via dashboard or Autoscaling Service

Factor 9 - Disposability

- “Maximize robustness with fast startup and graceful shutdown”-12factor.net

- Key points:
  - The 12-factor app's processes are disposable
  - Strive to minimize startup time
  - Robust against 'sudden death'

- Liberty provides fast startup and graceful shutdown
  - Use normal EE container lifecycle

Factor 10 – Dev/Prod parity

- “Keep development, staging, and production as similar as possible” - 12factor.net
- Key points:
  - Use the same (or very similar) services for dev and production
  - Dev/prod parity especially important for backing services
- What we did :
  - Docker Containers (docker-compose) for required services (locally)
  - Run in WDT locally vs. CF runtimes on Bluemix for production

Factor 11 - Logs

- “Treat logs as event streams” -12factor.net
- Key points:
  - A 12-factor app never concerns itself with routing or storage of its output stream
  - Process streams are captured by the execution environment
- Strict adherence:
  - set `com.ibm.ws.logging.trace.file.name=stdout` in bootstrap.properties
- More common
  - Use <toolOfChoice> to read/forward log files Liberty creates

Factor 12 – Admin processes

- “Run admin/management tasks as one-off processes”-12factor.net

- Key points:
  - Keep admin task code with application code
  - Run admin tasks in an identical environment to the app
    - Run against a “Release” / Same config
  - Same dependency isolation: gradlew, bundle exec, python virtualenv

In Conclusion…

In Conclusion…

- Most factors have a variety of implementation options
- Keeping the factors in mind gives us an application that:
  - Can be managed in a continuous delivery pipeline
  - Is a good candidate for microservices architecture
- There may be situations where you don't want to follow the rules

InterConnect 2016
The Premier Cloud & Mobile Conference

Questions?

February 21 – 25
MGM Grand & Mandalay Bay
Las Vegas, Nevada

Notices and Disclaimers

U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.

Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IN NO EVENT SHALL IBM BE LIABLE FOR ANY DAMAGE ARISING FROM THE USE OF THIS INFORMATION, INCLUDING BUT NOT LIMITED TO, LOSS OF DATA, BUSINESS INTERRUPTION, LOSS OF PROFIT OR LOSS OF OPPORTUNITY. IBM products and services are warranted according to the terms and conditions of the agreements under which they are provided.

Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.

Performance data contained herein was generally obtained in a controlled, isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.

Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.

It is the customer's responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer is in compliance with any law

Notices and Disclaimers Con't.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. IBM EXPRESSLY DISCLAIMS ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The provision of the information contained h erein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

IBM, the IBM logo, ibm.com, Aspera®, Bluemix, Blueworks Live, CICS, Clearcase, Cognos®, DOORS®, Emptoris®, Enterprise Document Management System™, FASP®, FileNet®, Global Business Services ®, Global Technology Services ®, IBM ExperienceOne™, IBM SmartCloud®, IBM Social Business®, Information on Demand, ILOG, Maximo®, MQIntegrator®, MQSeries®, Netcool®, OMEGAMON, OpenPower, PureAnalytics™, PureApplication®, pureCluster™, PureCoverage®, PureData®, PureExperience®, PureFlex®, pureQuery®, pureScale®, PureSystems®, QRadar®, Rational®, Rhapsody®, Smarter Commerce®, SoDA, SPSS, Sterling Commerce®, StoredIQ, Tealeaf®, Tivoli®, Trusteer®, Unica®, urban{code}®, Watson, WebSphere®, Worklight®, X-Force® and System z® Z/OS, are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: www.ibm.com/legal/copytrade.shtml.

Thank You

InterConnect 2016
The Premier Cloud & Mobile Conference

Your Feedback is Important!

Access the InterConnect 2016 Conference Attendee Portal to complete your session surveys from your smartphone,

laptop or conference kiosk.

February 21 – 25
MGM Grand & Mandalay Bay
Las Vegas, Nevada