REDUCING RESISTANCE
Deployment as a Surface

JEFFREY HULTEN
WHITEPAGES, INC.
FIRST, A STORY...
ONCE UPON A TIME

THERE WAS A DEVELOPER, WRITING A WEB APPLICATION
SHE DEPLOYED CODE TO PRODUCTION BY HAND AND EVERY DAY
UNTIL ONE DAY

HER TEAMMATE OVERWROTE HER CHANGES
BECAUSE OF THIS

SHE DECIDE
TO PACKAGE
HER CODE
BUT,

THAT DIDN’T ACCOUNT FOR CONFIGURATION
SO,

SHE DEPLOYED THAT PACKAGE WITH CHEF*  

* or other configuration management tool, our protagonist isn’t picky...
so,

SHE DEPLOYED THAT PACKAGE WITH PUPPET*

* I told you she isn’t picky…
BUT THAT LED TO

DEPLOYMENT TIME

COMPLEXITY
Until finally

She put it in a Docker container*

* or AMI… Like I said, not picky…
AND

SHE APPLIED
THE PRINCIPALS
OF THE TWELVE
FACTOR APP

http://12factor.net/
And ever since that day,

she chose her constraints to free herself of deployment pain.
TWELVE FACTORS?
THE TWELVE FACTORS

I. Codebase
One codebase tracked in revision control, many deploys

II. Dependencies
Explicitly declare and isolate dependencies

III. Config
Store config in the environment

IV. Backing Services
Treat backing services as attached resources

V. Build, release, run
Strictly separate build and run stages

VI. Processes
Execute the app as one or more stateless processes

VII. Port binding
Export services via port binding

VIII. Concurrency
Scale out via the process model

IX. Disposability
Maximize robustness with fast startup and graceful shutdown

X. Dev/prod parity
Keep development, staging, and production as similar as possible

XI. Logs
Treat logs as event streams

XII. Admin processes
Run admin/management tasks as one-off processes
THE ONE FACTOR

KEEP IT SIMPLE AND STRAIGHTFORWARD
“Simple can be harder than complex: You have to work hard to get your thinking clean to make it simple. But it’s worth it in the end because once you get there, you can move mountains.”

— Steve Jobs
A HARD SELL?

“Simplicity is a great virtue but it requires hard work to achieve it and education to appreciate it. And to make matters worse: complexity sells better.”

— Edsger Dijkstra
DEPLOYMENT: AN ANALOGY
YOU WANT TO FLY
BUT

YOU’RE A PIECE OF PAPER

FLICKR.COM/PHOTOS/94246383@N00/8637421219
FLIGHT REQUIRES THAT YOU

INCREASE LIFT & DECREASE DRAG
A FRESH SHEET OF PAPER

DOESN’T HAVE ENOUGH STRUCTURE
A BALL OF PAPER
HAS NO LIFT
BUT FOLD THE PAPER CAREFULLY

AND YOU CAN FLY
SO WHAT CONSTITUTES

DEPLOYMENT SURFACE?
IS IT

LIBRARIES AND FRAMEWORKS?

Flickr.com/photos/spam/5086168739
WHAT ABOUT RUNTIMES?

FLICKR.COM/PHOTOS/ROFANATOR/6454108731
COULD IT BE COMMON SERVICES?

Flickr.com/photos/1stpix_diecast_dioramas/7104489715
HOW ABOUT WIRE PROTOCOLS?

FLICKR.COM/PHOTOS/KTGEEK/6264018764
BUSINESS PRACTICES?

OR

FLICKR.COM/PHOTOS/FREDERICKMBROCKS/6154678068
YES
EVERY SURFACE REQUIRES A CONTROL
LIBRARIES AND FRAMEWORKS

- Does the third party framework solve more problems than it creates?
- Will that still be as true in the last week of the project as it is in the first?
- Is there a simpler way?
EXAMPLE: RUBY API SERVICE
RUNTIMES

• Runtime: Ruby VM, JVM, Erlang VM, Binary Executable

• How many runtimes are needed to serve your customers?

• Are you adding complexity because your runtime doesn’t work the way you need to?
EXAMPLE: RUBY, UNICORN, AND CONNECTION POOLS
RUNTIMES? NOT LANGUAGES?

- Languages are (mostly) a build time concern.
- Runtimes are a deploy time concern.
We program with constructs[...] but we are in a business of artifacts.

—RICH HICKEY
COMMON SERVICES

- How many data stores/in-memory caches/load balancers are you using?
- Does your team have the expertise to support them all?
- Does your team have the time/‘bandwidth’ to learn a new service?
- Nothing is a simple as the packaging says.
DUPLICATE COMMON SERVICES

Postgres & MySQL
Redis & Memcache
Riak & Cassandra
Nginx & Apache HTTPd
Zookeeper & etcd
ActiveMQ & Rabbit MQ
WIRE PROTOCOLS

- Eventually you will need to look at the traffic.
- Can you inspect your services on the wire?
- Does a new wire protocol justify the additional complexity?
DUPLICATE WIRE PROTOCOLS

HTTP & ???
Avro & Thrift & Protobuf
JSON & XML
BUSINESS PRACTICES

- Do your practices lift you up or drag you down?
- Do you release features faster than the code rots?
- Do you expect to find all defects before you go to production?
The art of programming is the art of organizing complexity, of mastering multitude and avoiding its bastard chaos as effectively as possible.

-EDSGER W DIJKSTRA
IN THE BEGINNING...

REMEMBER OUR DEVELOPER?
STRATEGIES OF DEPLOYMENT

- Manual Push
- Package Push
- Configuration Management Tool (Chef/Puppet)
- Docker Container
$ for s in 1 2 3; do \ 
rsync -a ~/src web${s}.example.com:/opt/app; 
ssh web${s}.example.com \ 
sudo service apache2 restart; \ 
done
MANUAL PUSH

$ for s in 1 2 3; do \
    rsync -a ~/src web${s}.example.com:/opt/app; \
    ssh web${s}.example.com \ 
        sudo service apache2 restart; \ 
    done
CREATE PACKAGE

$ fpm -s dir -d deb -n myapp -v 1.2.3 ~/src
Created deb package {:path=>"myapp_1.2.3_amd64.deb"}
INSTALL PACKAGE

$ sudo dpkg -i myapp_1.2.3_amd64.deb
apt_package "myapp" do
  source "/path/to/myapp_1.2.3_amd64.deb"
  action :install
end
FROM ubuntu:trusty
MAINTAINER Debbie Dev <ddev@myappco.com>
ADD ./myapp_1.2.3_amd64.deb /opt/pkgs
RUN dpkg -i /opt/pkgs/myapp_1.2.3_amd64.deb
ENV ENVIRONMENT prod
CMD service start myapp
The first step toward the management of disease was replacement of demon theories and humours theories by the germ theory. That very step, the beginning of hope, in itself dashed all hopes of magical solutions. It told workers that progress would be made stepwise, at great effort, and that a persistent, unremitting care would have to be paid to a discipline of cleanliness. So it is with software engineering today.

–FRED BROOKS
FINDING ME

Jeffrey Hulten

@jhulten

linkedin.com/in/jhulten

jhulten@whitepages.com
Q & A
SESSION FEEDBACK

Please provide feedback on this session!

You can do so in 3 ways:

• Visit this session on the Mobile App. Click Session Feedback.
• Scan the unique QR Code for this session located at the front and back of the room.
• Visit the unique URL for this session located at the front and back of the room.

Thank you for providing your feedback