DevOps at inBay Technologies

Part 2 • Continuous Delivery

John Marshall, Director Software Development

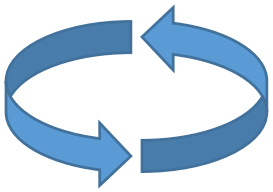
inBay TECHNOLOGIES INC.

Goals

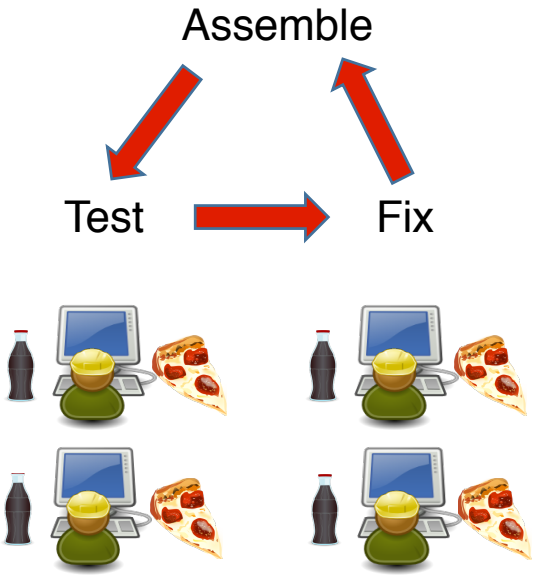
- Predictability
  - Velocity
  - Quality
- Agility
  - Short commitments, change lanes with minimal notice
  - Minimize the investment to validation cycle

Where we started

Development Cycle



Non-Stop Release Party



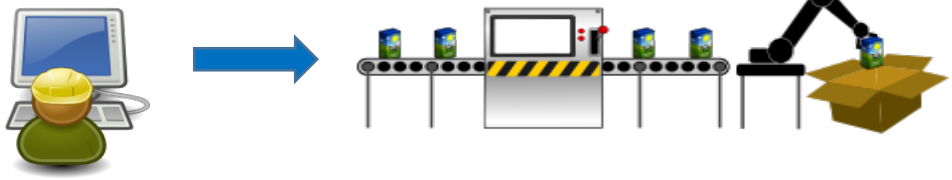
Aftermath

What we want to achieve

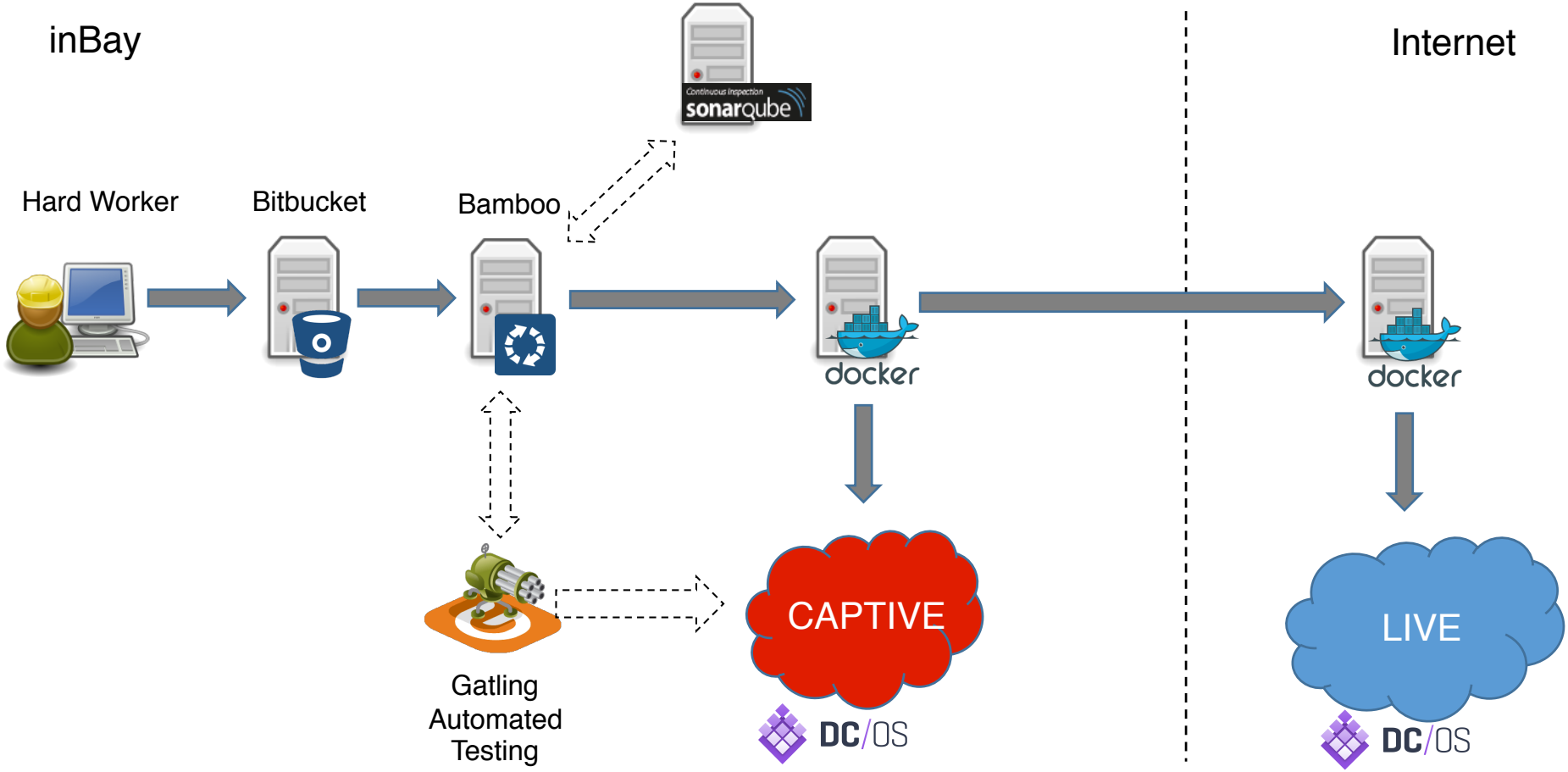
Develop

Build → Test → Deploy

Aftermath



Our Setup



Building…

Working with the CD Pipeline

- Requires a bit of a mind shift
  - Each submit will go live (on its own)
  - Treat the pipeline as a live system
  - Avoid application-test coupling

- Automate back pressure as well as forward flow
  - You need to have safeguards & road blocks
  - We use Sonarqube's quality gate
  - Breaks the build if a gate fails

Automated Versioning

- There are no snapshots
  - Every build is a potential release & must have a unique version
- Traditional versions like 1.1.0 don't really make sense
  - There are no 'releases' just a continual evolution of patches
- We use timestamps for versions
  - Dates are more meaningful than arbitrary version numbers
  - We include build metadata: Bamboo build number and git commit ID

House cleaning

- Docker artifacts accumulate in a hurry
  - On your development machine
  - On the build server
  - In the Docker Registry
  - In the cluster
- Automated cleanup is a necessity
  - We do this using scheduled jobs

Testing…

Predictability/agility only as good as your testing

- Start early (it's hard to catch up)
  - It will become part your velocity & culture
- Avoid accumulating test debt
  - You cannot reliably change uncovered code
  - You will end up writing these tests when you can least afford it
- Peer test reviews as/more important than code reviews
  - You can automate tests
  - But you need people to identify them
  - Best done before you implement

Invest in your test infrastructure

- You will write a lot of tests
  - Try and make the task as efficient and pleasant as possible

- We use Gatling, with it we have…
  - Build libraries of re-usable test steps, scenarios & support classes
  - Implemented environment variable based test configuration
  - Test suites can be run
    - Locally by developers via sbt
    - In the cloud in a Docker container

Mock External Services

- Prefer mocks over actual services, since mocks can
  - Perform message content validation
  - Inject error responses

- Our mock services

  Have an automation API that the automated tests use
  - To validate operations
  - Inject behaviors

Deploying…

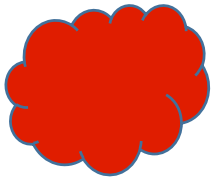
Our Environments

Live

- Production Services
- Behind HAProxy
- Secure Access
- Everything redundant

Testing

- Mock services
- Behind HAProxy
- Secure Access
- Everything redundant



Dev Lab

- Some real services
- Some services off
- Both secure and insecure access
- Non redundant

Local

- Services?
- No HAProxy
- Non-redundant
- Insecure access

Roll Your Own Dockers

- Allows you to have a common image hierarchy
- Common runtime & support environment
- Common place(s) to apply updates & patches
- More efficient Docker storage usage

Be environment independent

- All settings should be configurable by environment variables
  - Especially environment related settings
- Log to stdout
  - Use a Docker log driver to get the logs where you want them

- We can automate pushing the image through the pipe

  Dev -> QA -> Staging -> Production
  - By supplying the relevant env var file at each stage
  - By allowing each environment to supply its own log routing

Configuration via Environment Variables

- We use Typesafe Config for our configuration
- We built a reference.conf preprocessor SBT plugin
  - Takes in src/…/reference.conf
  - Injects env var overrides for all the vars in the file
  - Outpus target/.../reference.conf
- We do this for important vendor files as well

Small company life

- You can’t run a full cluster for every developer
  - Have a minimal deployment available for developer machines
  - Be able to run without things like https/tls
  - Private certificate authorities are a pain

Logging and troubleshooting with micro services

- Store logs in an indexed, searchable storage
  - We use Elasticsearch
  - Use a uniform log record layout
- Use correlators to link service spanning activities together
  - In inter-service messages
  - In log messages

Where did that Redis get to?

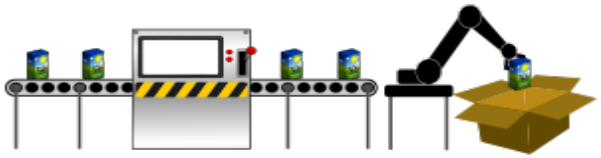
- In DC/OS things move around during deploys/restarts
  - Not everybody likes that, like things involving Redis
  - Your dev/test cluster should have space for things to move

Where we are today

Develop

Build → Test → Deploy

Aftermath



???

< 30 min