Distributed CI: Scaling Jenkins on Mesos and Marathon

Roger Ignazio – Puppet Labs, Inc.

MesosCon 2015 – Seattle, WA

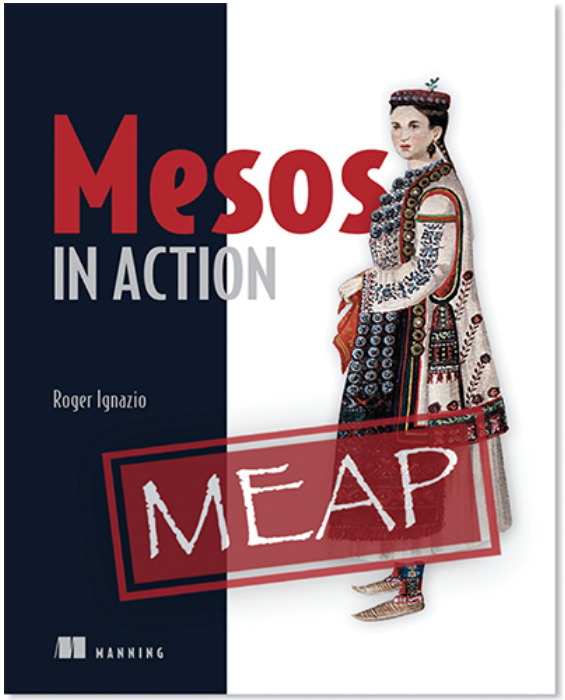
About Me

Roger Ignazio

QE Automation Engineer
Puppet Labs, Inc.

@rogerignazio

Mesos In Action



mesosinaction.com

Code: `cftwmesos`

Agenda

- Puppet Labs' testing environment
- Conventional methods for scaling Jenkins
- Motivations for re-architecting CI
- Breaking up the Jenkins monolith
- Demo, outcomes, and future work

Audience Poll

Intro to Mesos

- General-purpose cluster manager
- Represent many machines as a single entity
- Advertise resources directly to applications

Intro to Marathon

- Mesos framework that provides private PaaS
  - Manages long-running tasks
- Easily scale apps to *N* instances
- Automatically restarts failed app instances

Intro to Jenkins

- Distributed, open source CI tool
- Repeatable build/test of software projects
- Large community, rich plugin ecosystem

Puppet Labs' Testing Environment

Testing at Puppet Labs

- 4k to 5k builds/day across 75 platforms
- 15 Jenkins clusters
  - Loosely based on team, project, function
  - ~ 1,300 executors across ~ 240 build machines
- Tooling: Beaker, vmpooler

Testing at Puppet Labs

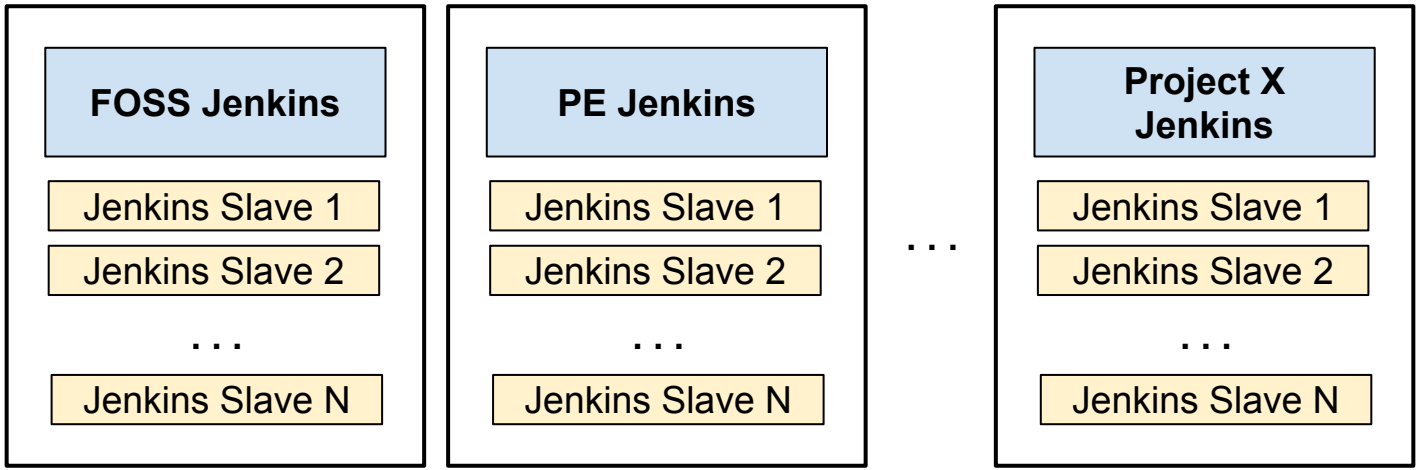
- Configuration management
  - Most job configs and scripts stored in Jenkins
  - Infrastructure managed by Puppet
- Reporting
  - Jenkins UI (x15!)
  - Clockin, Waylon

Conventional Methods for Scaling Jenkins

Scaling Jenkins

- Two common deployments
  - Single Jenkins master with many jobs
  - Master per team, project, or function
- Not highly available
- Can’t load balance across masters
- Static partitioning kills overall utilization

Scaling Jenkins



90% Utilized (Normal)

140% Utilized (40 builds in queue)

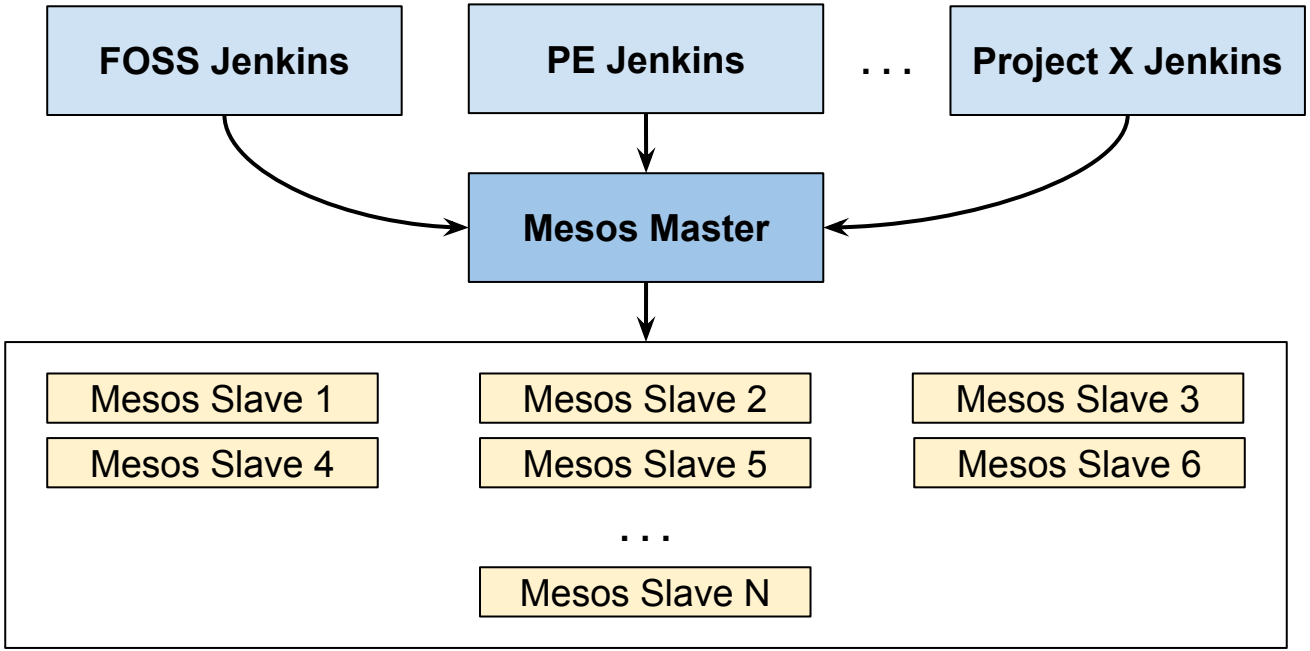
0% Utilized (Idle)

Scaling Jenkins

What can we do about it?

On-demand resources!

Scaling Jenkins



Motivations for Re-architecting CI

User Stories

As a **`$role`**,

I { want, need } **`$something`**,

so that **`$outcome`**

User Stories

As a Developer, I want tests to be run against pull requests, so that I have confidence in the code about to be merged

User Stories

As a Developer, I don't want to worry about the underlying infrastructure of the CI system

User Stories

As a CI consumer, I want a central location to view all CI activity, so that I don't have to visit multiple URLs

User Stories

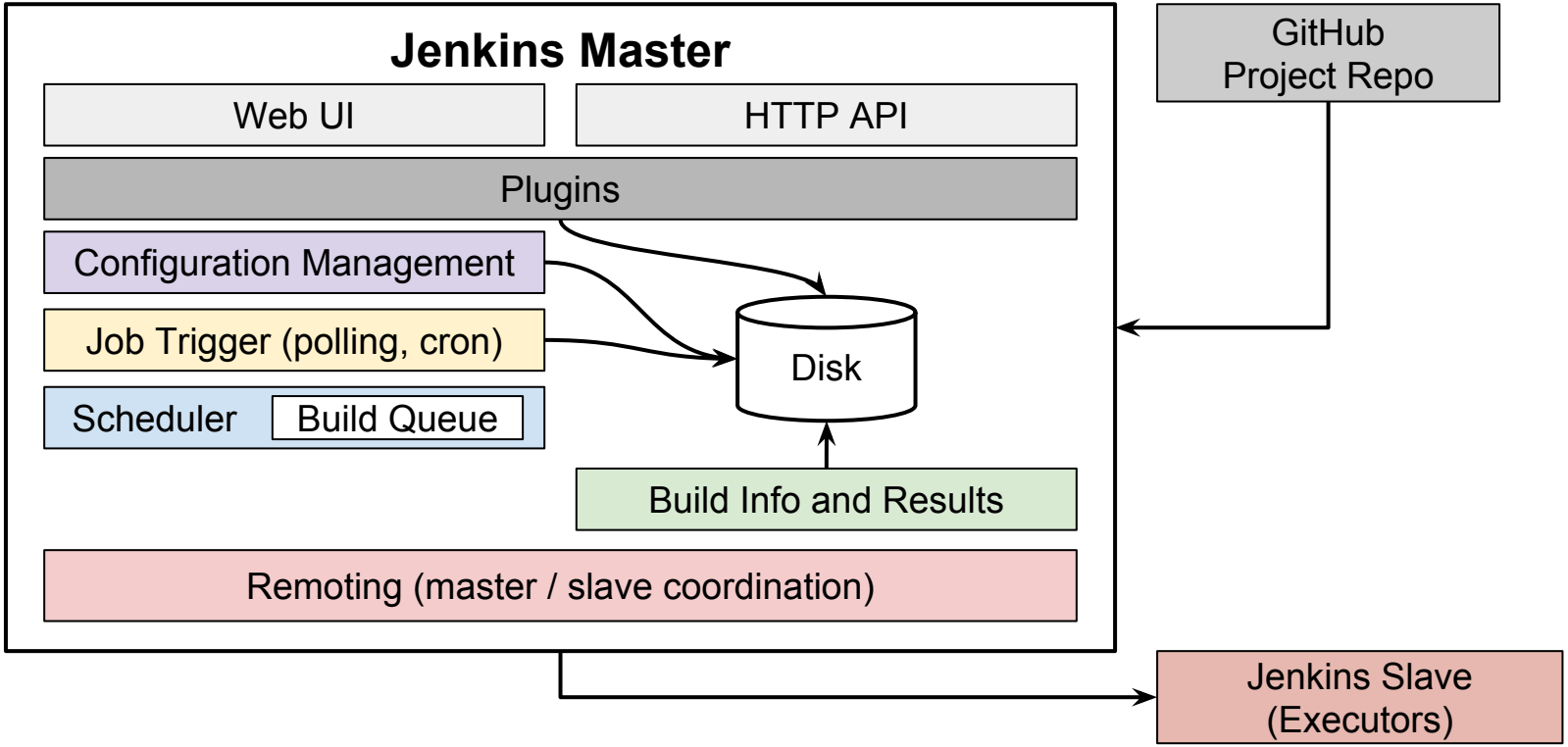
As a QE, I want slaves to be on-demand, so that infrastructure resources are used more efficiently

Motivation

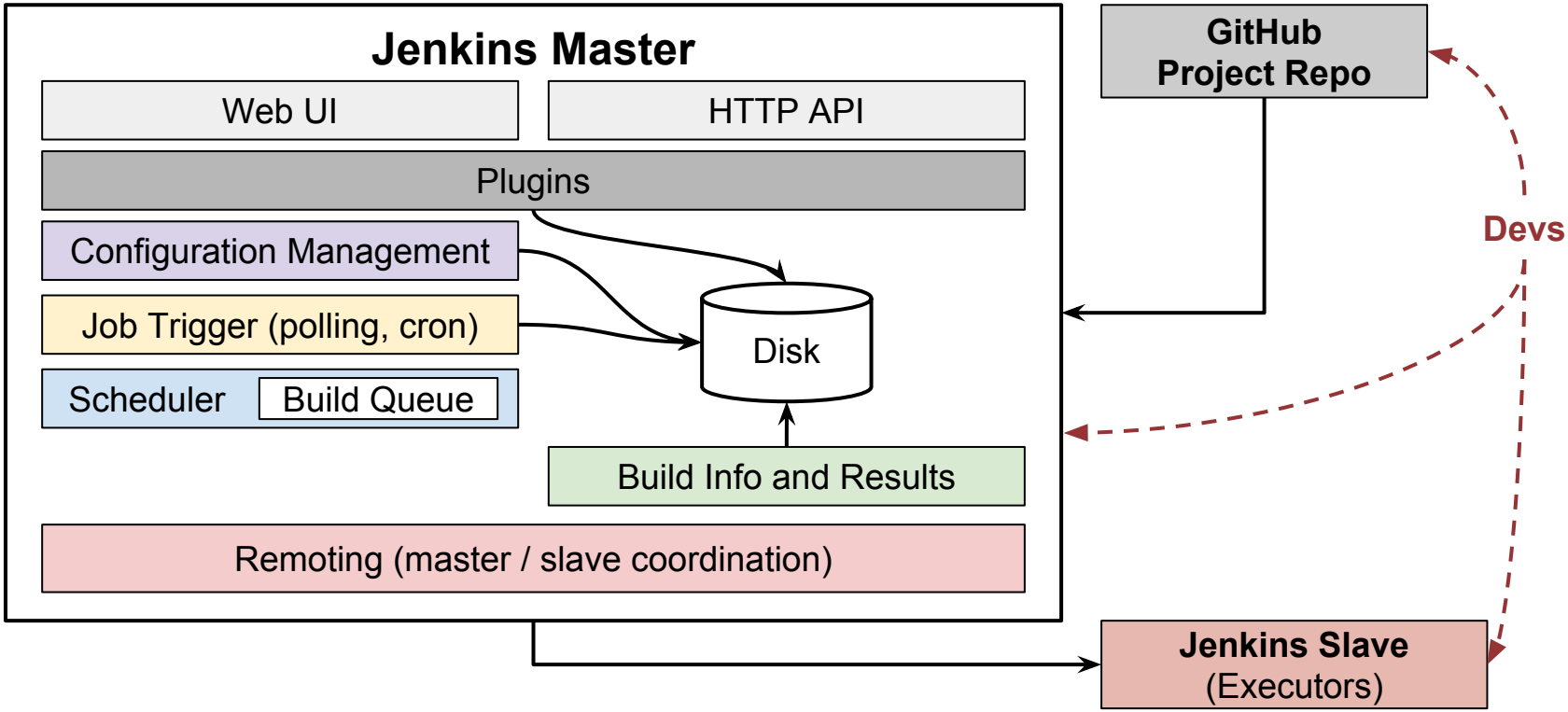
- Reduce friction in dev workflows
- Event (and data)-driven system
- Improve reporting and user experience
- Scale to meet growing demand

Breaking Up The Monolith

The Jenkins Monolith



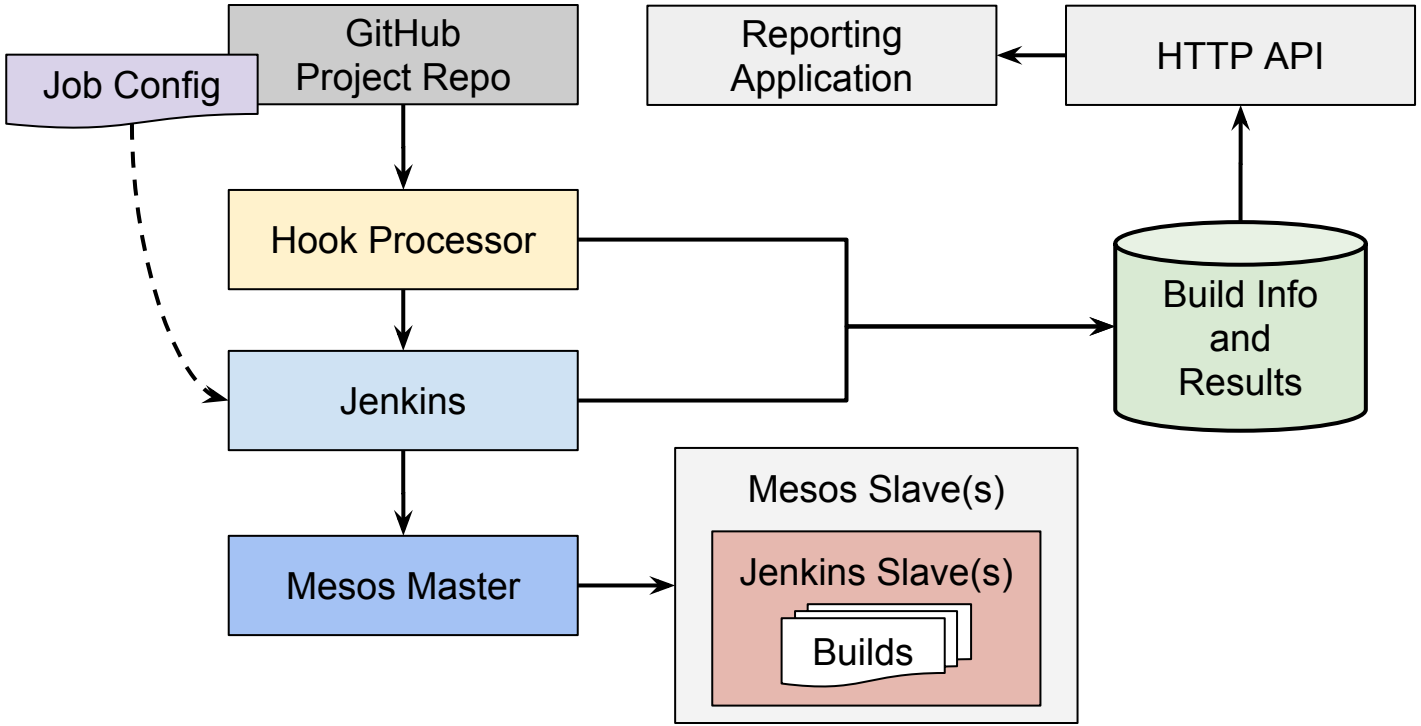
The Jenkins Monolith



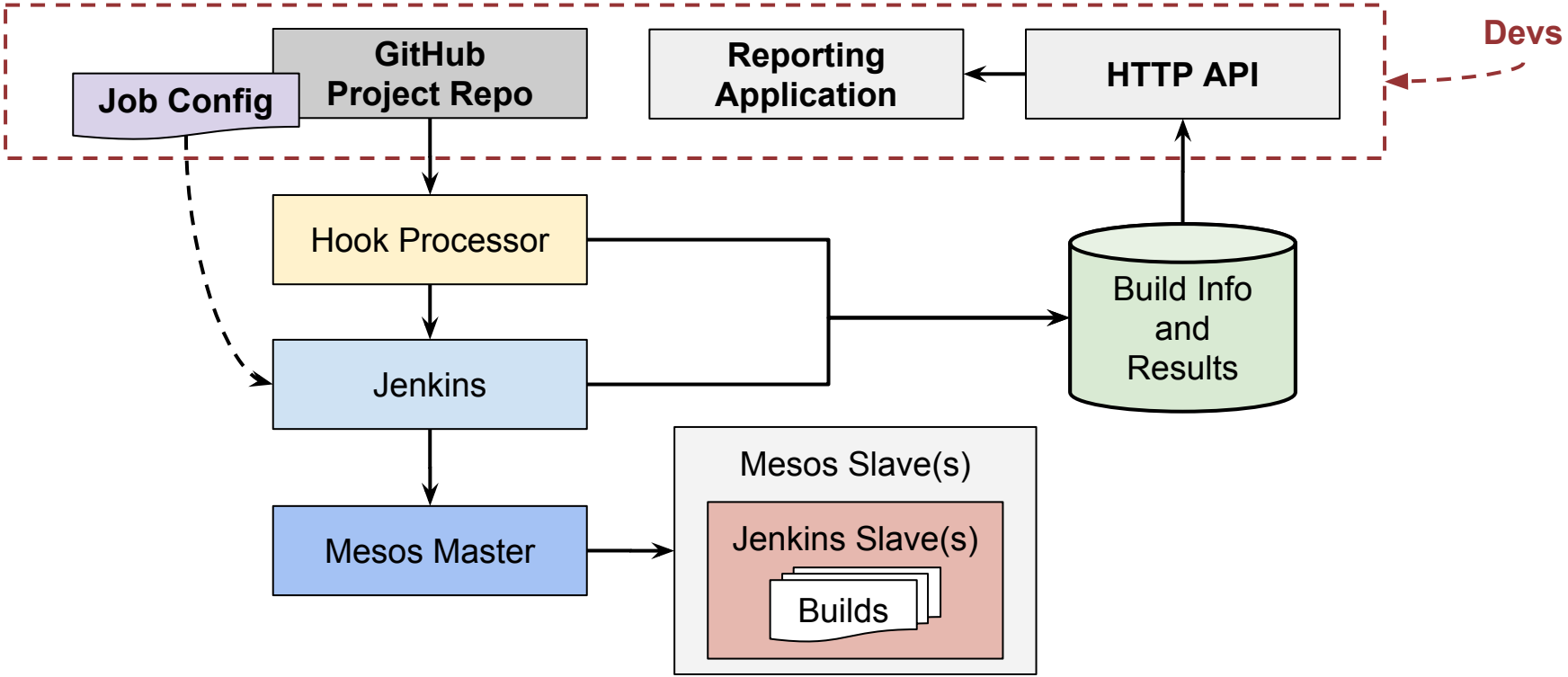
Breaking Up The Monolith

- Job configurations
- Build trigger
- Build history

Breaking Up The Monolith



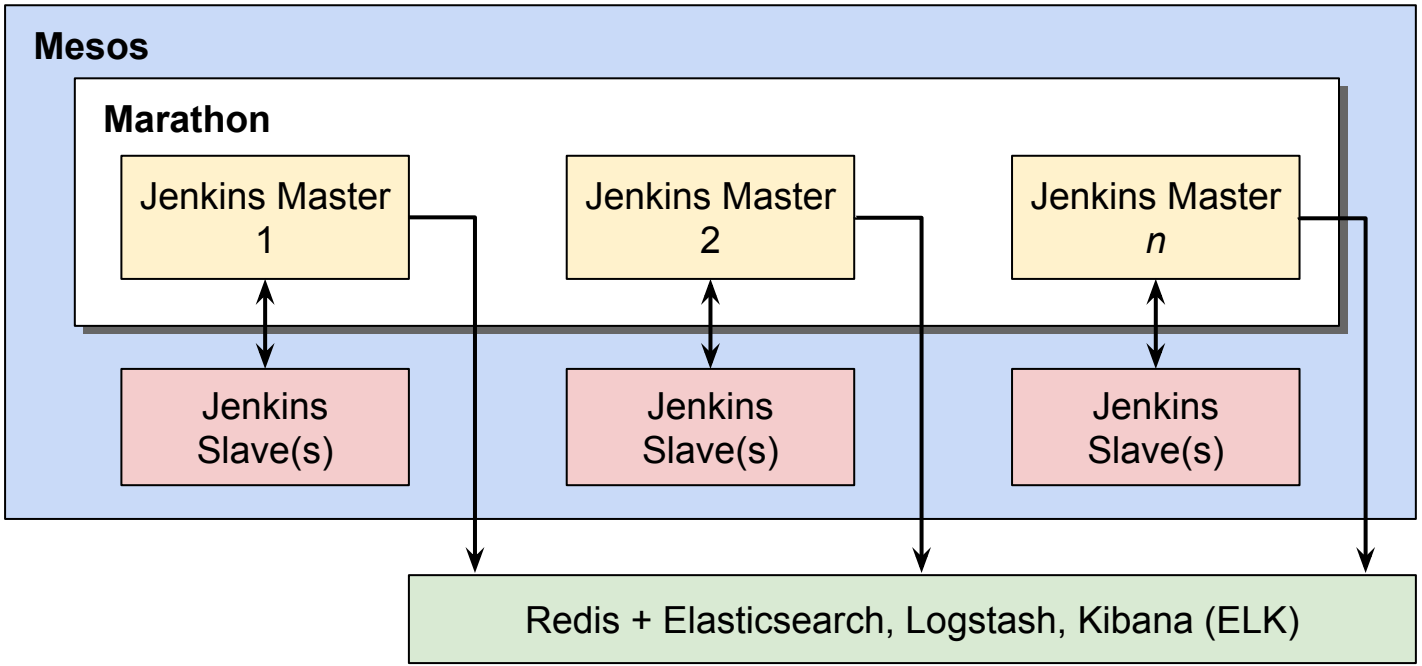
Breaking Up The Monolith



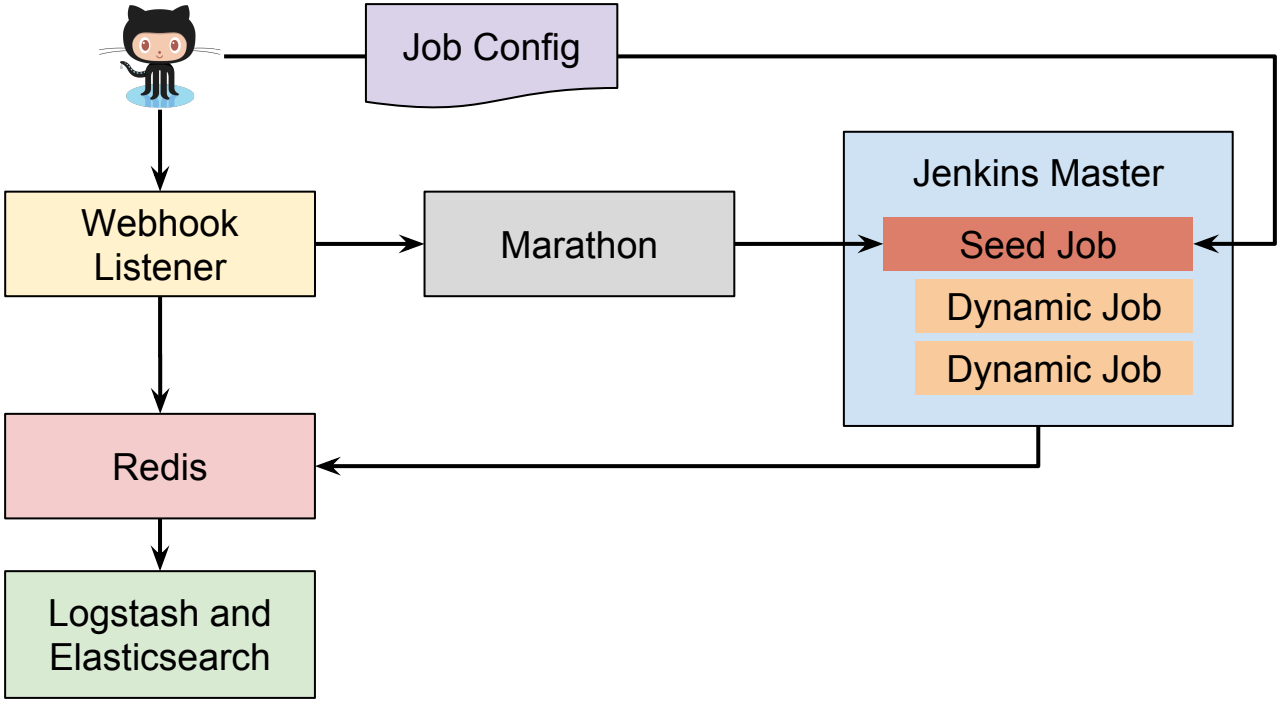
Jenkins on Marathon

- Marathon as a private PaaS
- Scale Jenkins masters horizontally
- Deploy updates, config changes, plugins
- Continuous deployment of CI *?!*

Jenkins on Marathon



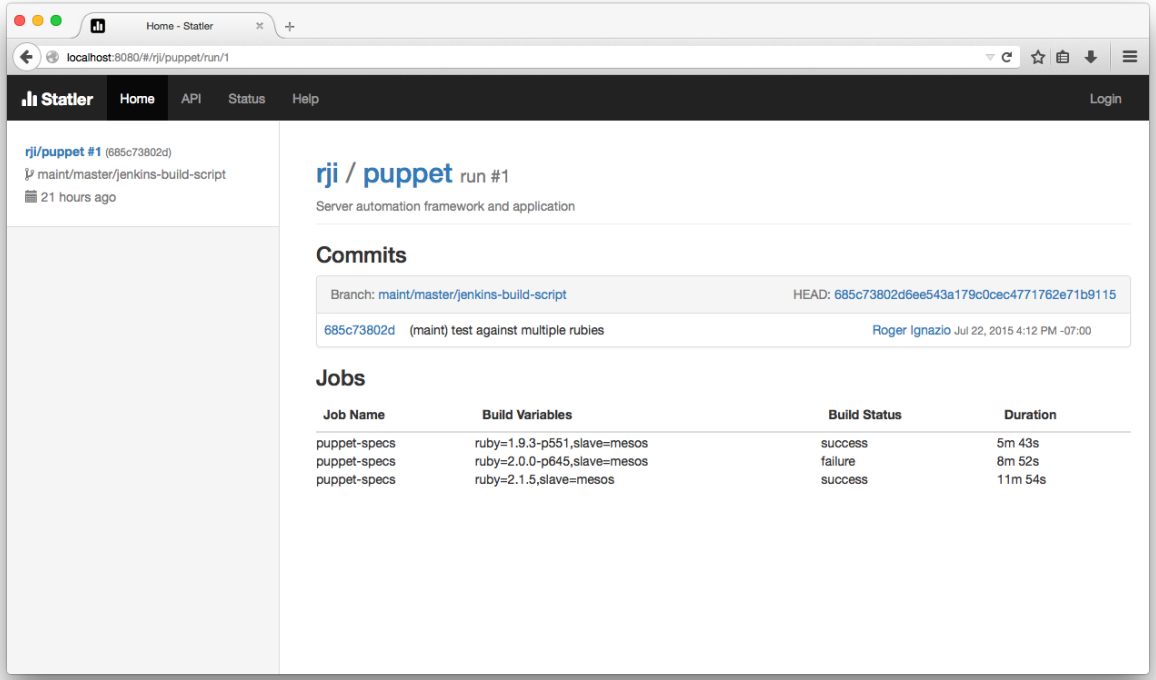
GitHub Webhook Processor



Reporting

- Unique IDs for each event (in Redis)
- Webhooks and build data (in Elasticsearch)
- Query and visualize system activity (in Kibana)
- Build our own reporting app?

Reporting


Home - Statler
localhost:8080/#/rji/puppet/run/1
Statler
Home
API
Status
Help
Login
rji/puppet #1 (685c73802d)
maint/master/jenkins-build-script
21 hours ago
rji / puppet run #1
Server automation framework and application
Commits
Branch: maint/master/jenkins-build-script
HEAD: 685c73802d6ee543a179c0cec4771762e71b9115
685c73802d (maint) test against multiple rubies
Roger Ignazio Jul 22, 2015 4:12 PM -07:00
Jobs
Job Name
Build Variables
Build Status
Duration
puppet-specs
ruby=1.9.3-p551,slave=mesos
success
5m 43s
puppet-specs
ruby=2.0.0-p645,slave=mesos
failure
8m 52s
puppet-specs
ruby=2.1.5,slave=mesos
success
11m 54s

Demo

rji/mesoscon-2015-demo

Outcomes and Future Work

Outcomes

- Single Git-based workflow
- Standardized, stateless Jenkins masters
- Jenkins slaves provisioned on-demand
- Bonus: private PaaS (Marathon)

Future Work

- Adoption
- Single reporting dashboard (with API)
- Intelligent job queueing and throttling
- Job DSL plugin abstraction and templates

Q & A

@rogerignazio