The Key Components of Adopting CI The OpenStack Way

Who Am I

- Wajdi Al-Hawari
- Software Developer @ Internap
  - OpenStack Ironic
  - Server/Network Automation

Who Is This Talk For

- Developers
- Users/Operators
- Product Owners et al.

What We Are Covering Today

- History and Evolution
- The Solution
- What's Next?
- Questions?

Our Landscape

- ~30 developers
- Four cross-functional teams
- Test-Driven Process
- Over 100 active repositories

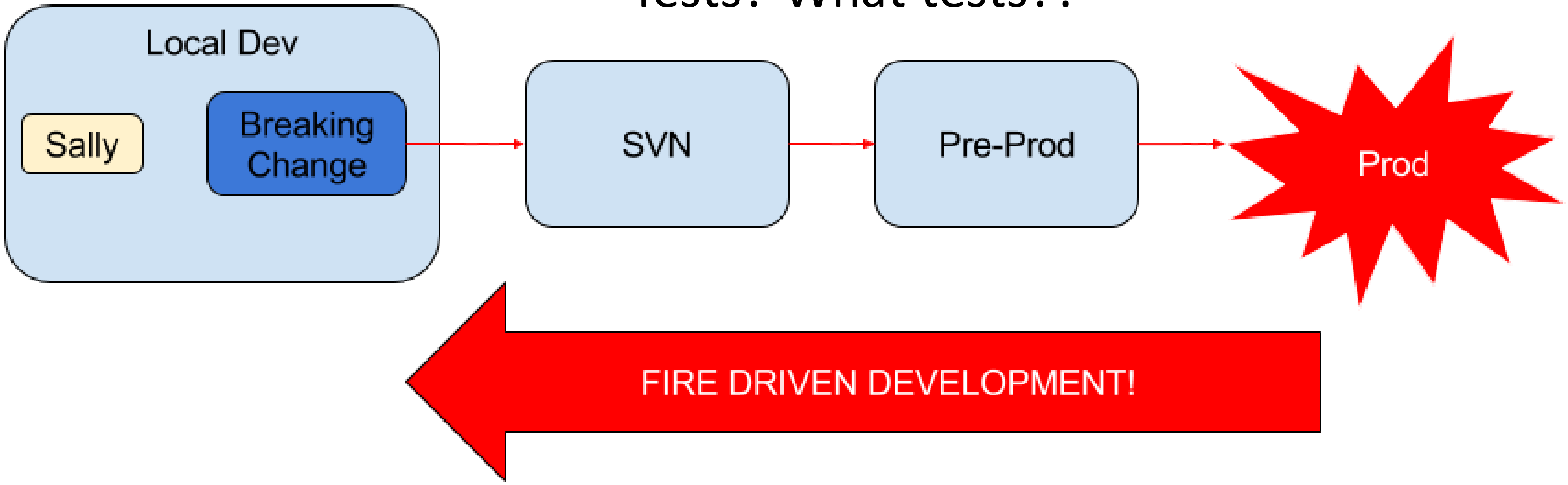
The Goal

- Make it better, stronger, faster

Problem	Solved?
Code Confidence	No
Integration/System Tests	No
Prevent Production Outages	No

A Long Long Time Ago.....Version 1

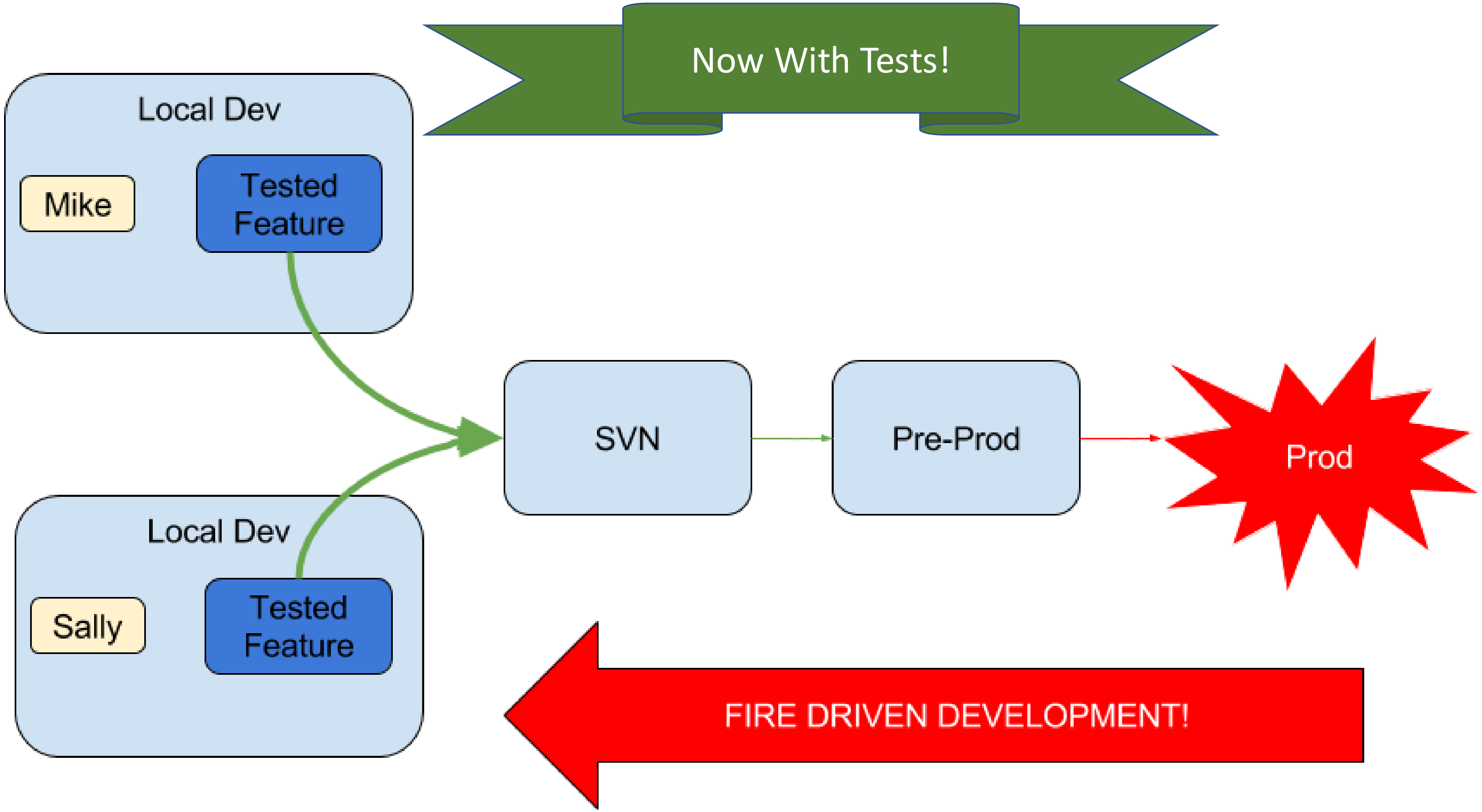
Tests? What tests??



Problem Tracker

Problem	Solved?
Code Confidence	No
Integration/System Tests	No
Prevent Production Outages	No

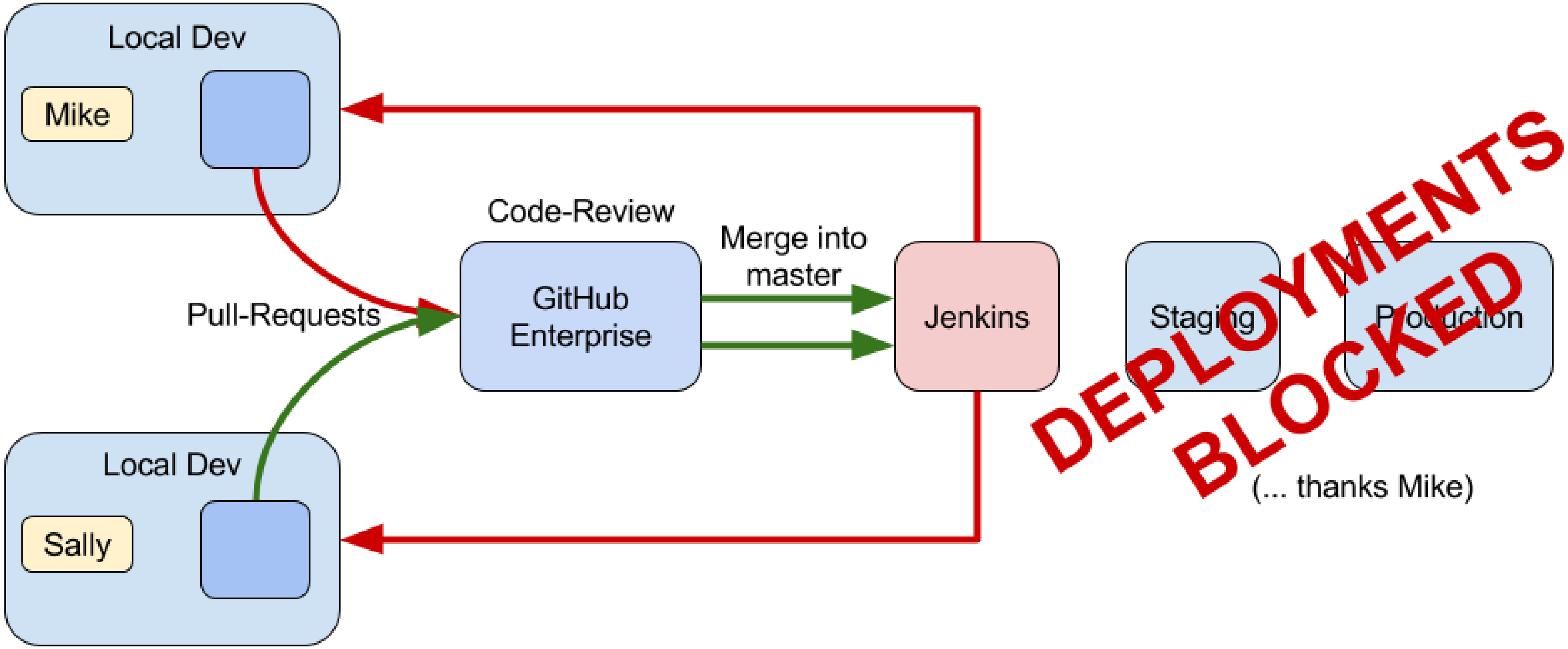
A Long Long Time Ago.....Version 2



Problem Tracker

Problem	Solved?
Code Confidence	Somewhat....
Integration/System Tests	No
Prevent Production Outages	No

A Long Long Time Ago.....Version 3



Problem Tracker

Problem	Solved?
Code Confidence	Almost!
Integration/System Tests	No
Prevent Production Outages	Slow Delivery
Jenkins Needed Love	No

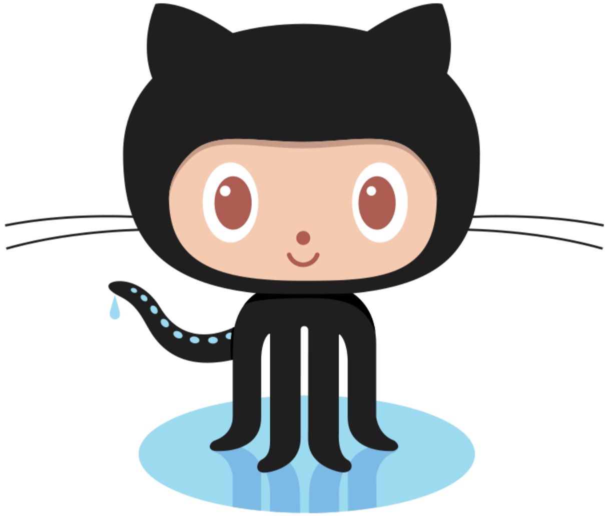
Minor Improvements

- Smarter usage of GitHub
- Jenkins Job Builder

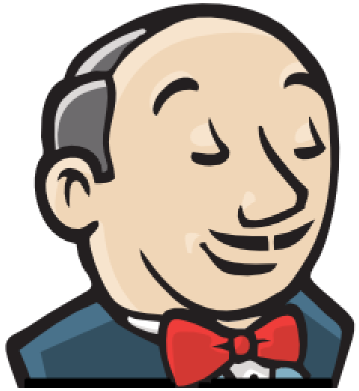
Problem Tracker.....Updated Again

Problem	Solved?
Code Confidence	Almost!
Integration/System Tests	No
Prevent Production Outages	Not Quite
Jenkins Needed Love	Getting There!

Making The Decision



- Already invested financially
- We don't want to move away from it

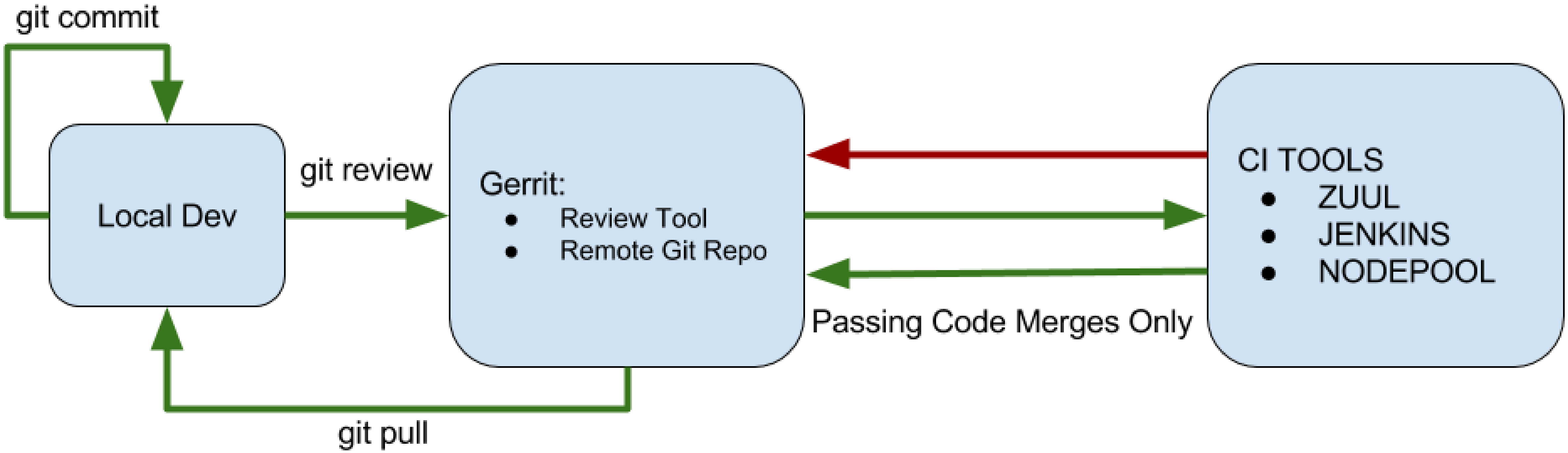


- Does what we want it to do
- Job Builder is great!

We Know What's Out There

Software-Factory

Version 4……The OpenStack Way!



git commit

Local Dev

git review

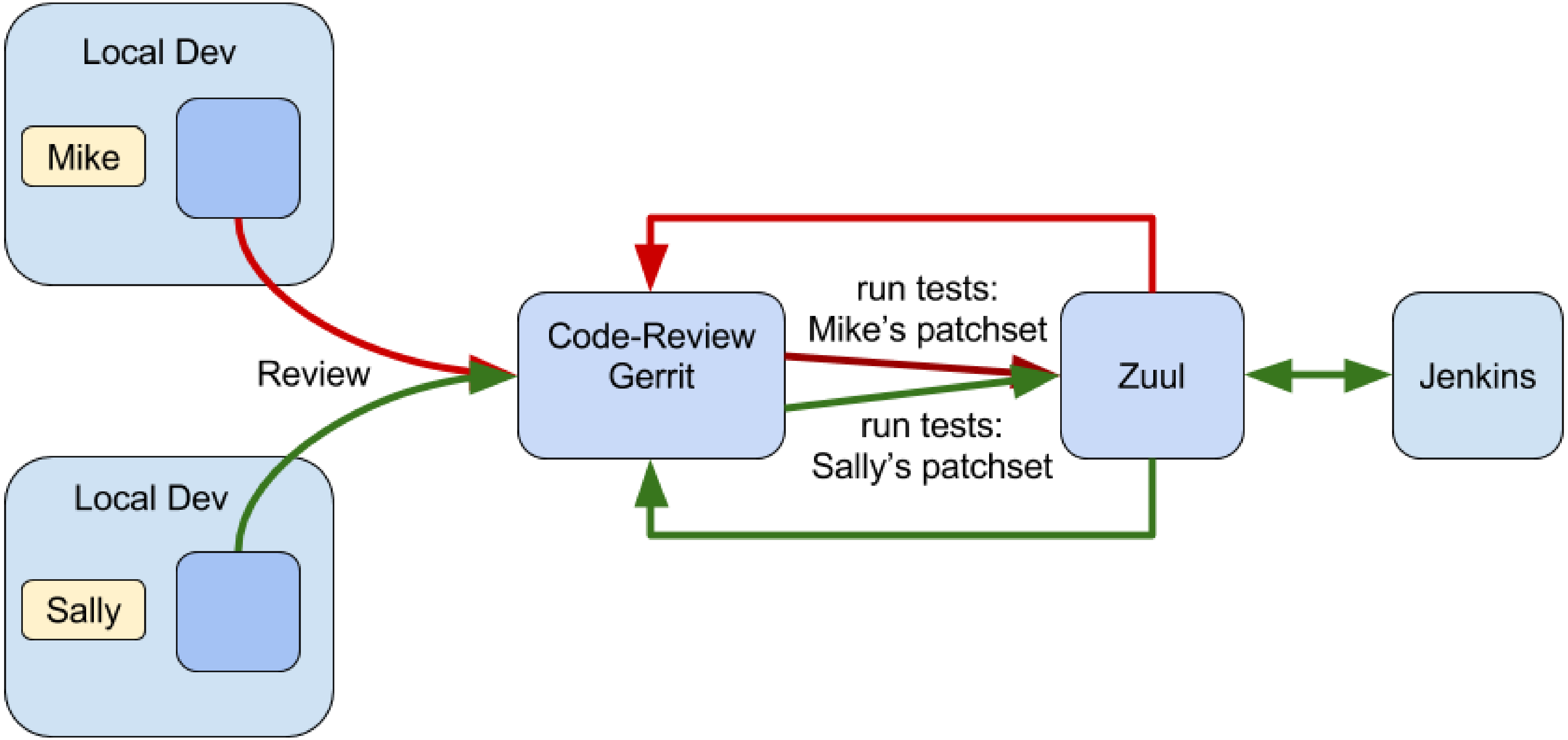
Gerrit:
- Review Tool
- Remote Git Repo

CI TOOLS
- ZUUL
- JENKINS
- NODEPOOL

Passing Code Merges Only

git pull

Version 4.....Mike and Sally!



Problem Tracker – Solved!

Problem	Solved?
Code Confidence	Yes!
Integration/System Tests	Yes!
Prevent Production Outages	Yes!
Jenkins Needed Love	Yes!

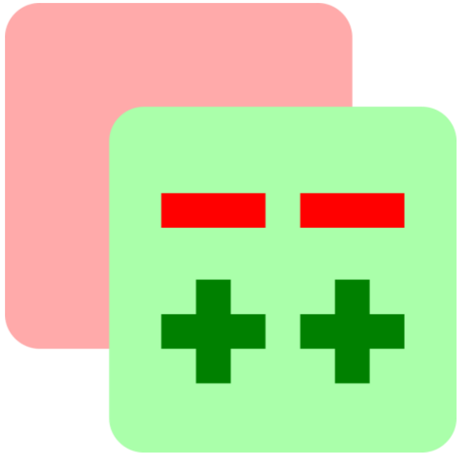
Before OpenStack CI?

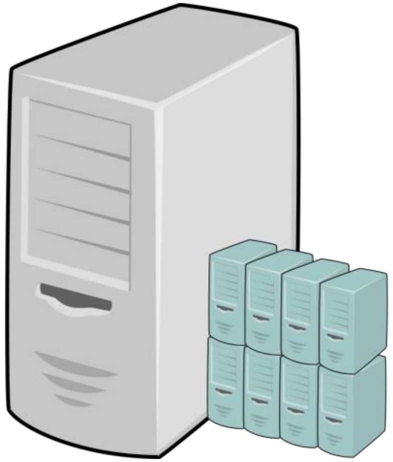
- Merge first, test later
- Different change-sets were not tested against each other
- No one stops you from merging in to master
- Code-Review seemed optional

After OpenStack CI

- Test first, merge later
- Change-sets were tested against each other
- You *could* insta-merge, but Jenkins has final say (Stops Mike from being Mike)
- Code Review is enforced, both by system, and developer adoption

The OpenStack CI

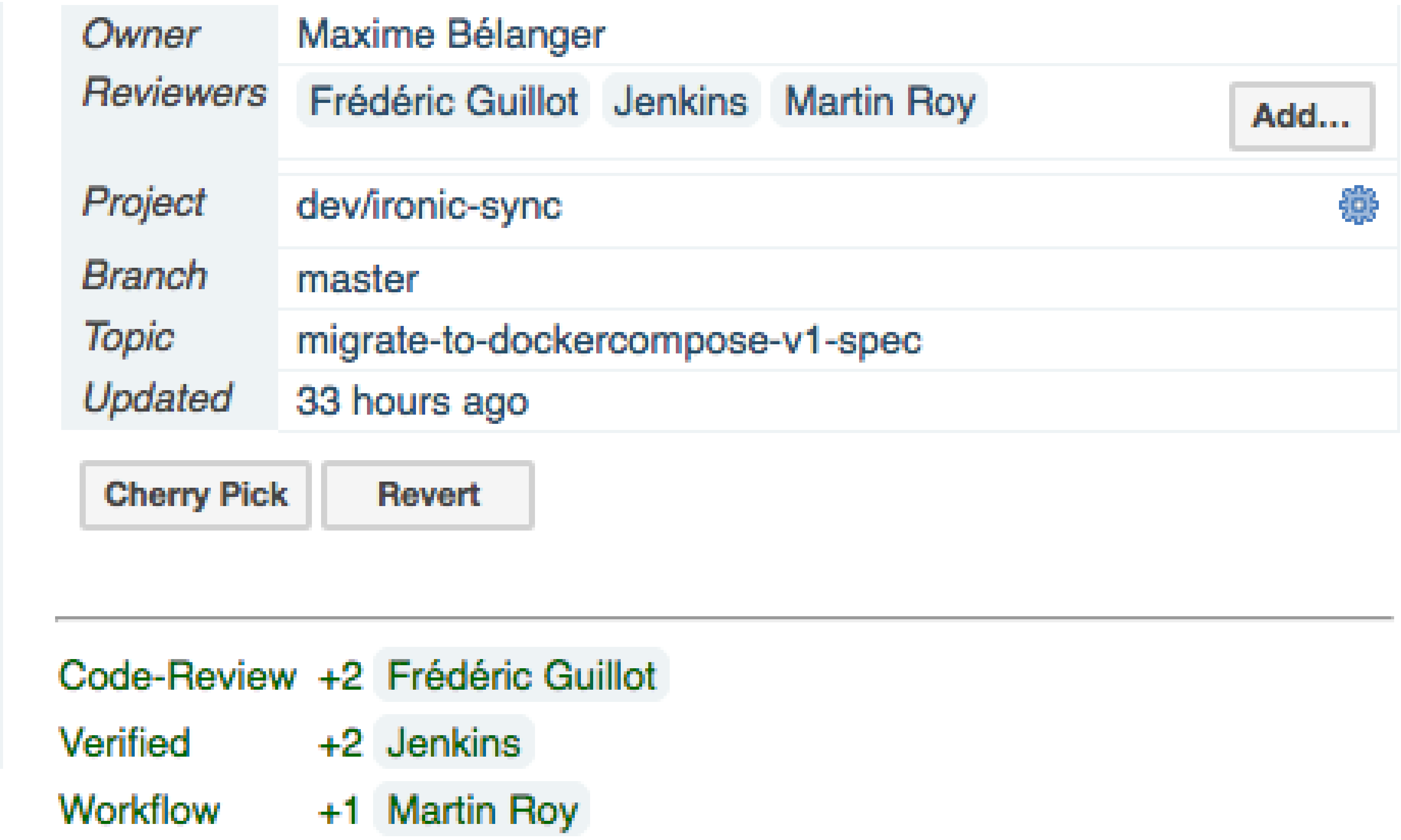




Gerrit

- Collaborative review tool with web interface
- Git plug-in `git review` adds on to standard git commands
- All standard `git` commands still work
- Voting system

Gerrit - Work Flows



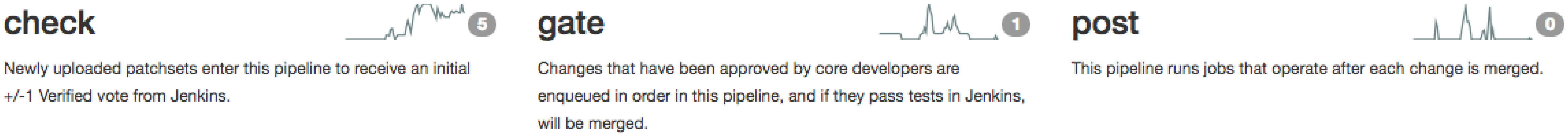
Owner	Maxime Bélanger
Reviewers	Frédéric Guillot Jenkins Martin Roy
Project	dev/ironic-sync
Branch	master
Topic	migrate-to-dockercompose-v1-spec
Updated	33 hours ago

Code-Review	+2	Frédéric Guillot
Verified	+2	Jenkins
Workflow	+1	Martin Roy

Zuul: The Brains

- Developed specifically for Gerrit
- Decides what happens on:
    - successful tests
    - failed tests
    - Jenkins jobs that should be triggered

Zuul – Pipelines



check 5

Newly uploaded patchsets enter this pipeline to receive an initial +/-1 Verified vote from Jenkins.

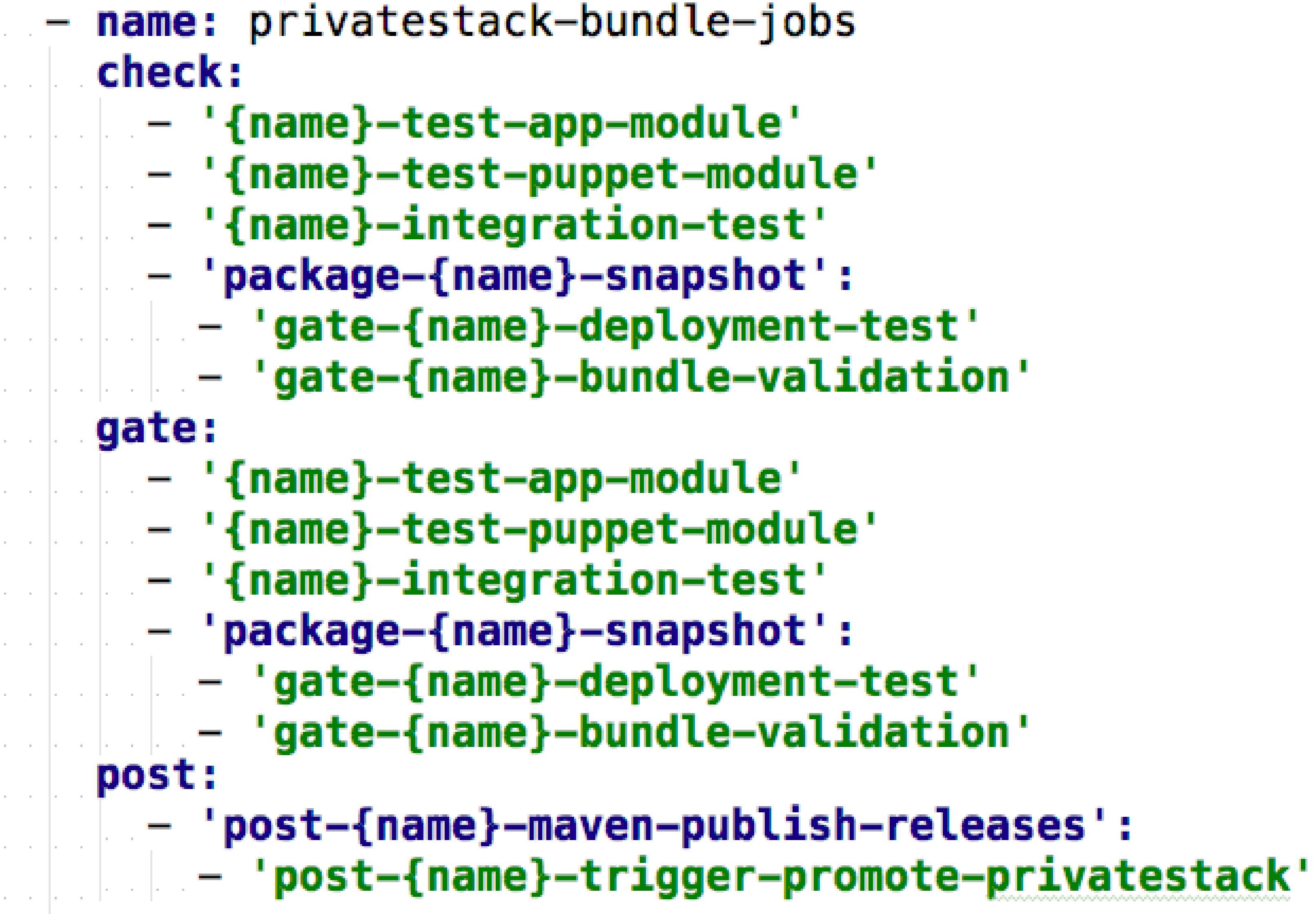
gate 1

Changes that have been approved by core developers are enqueued in order in this pipeline, and if they pass tests in Jenkins, will be merged.

post 0

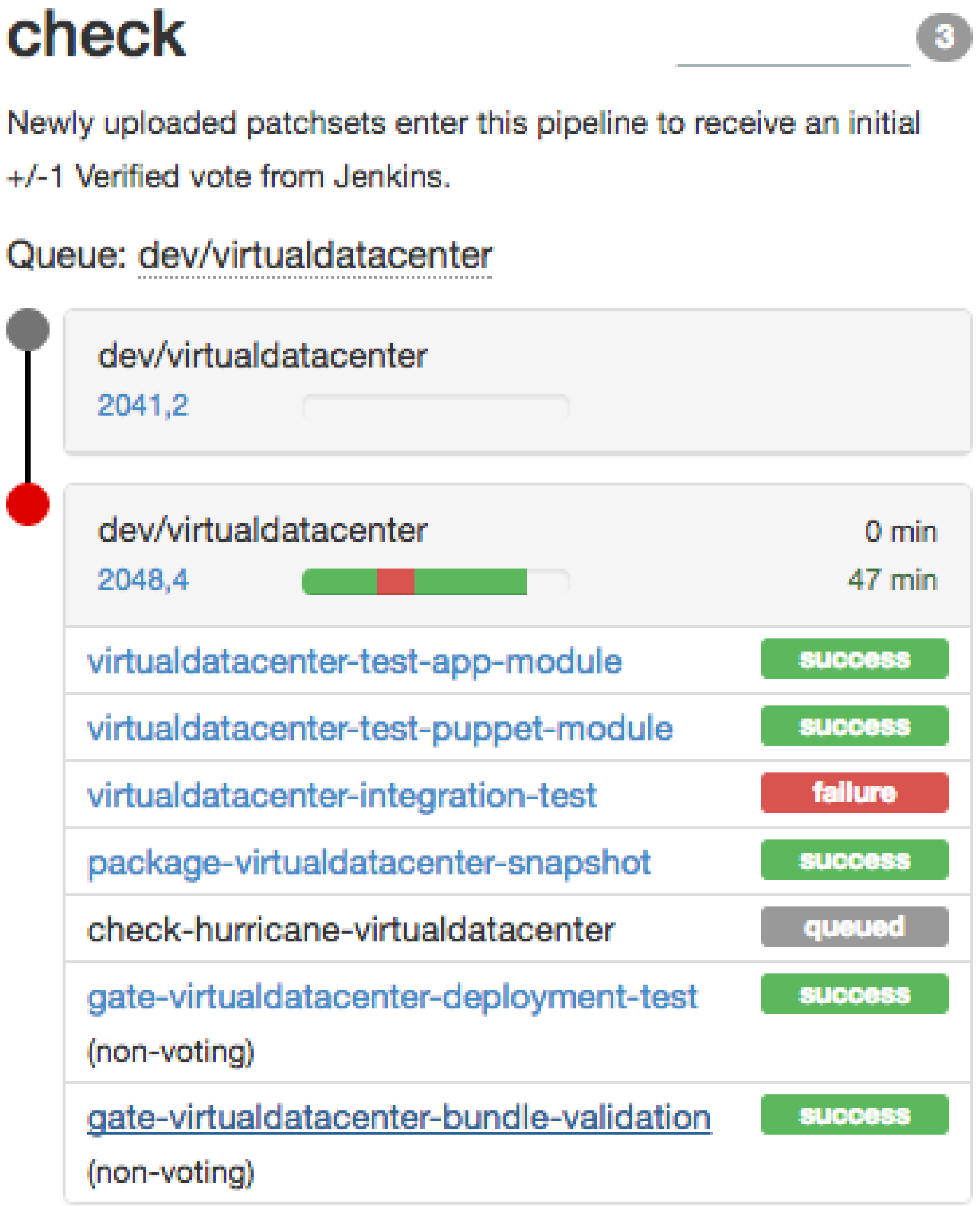
This pipeline runs jobs that operate after each change is merged.

Zuul – Jobs Per Pipeline



```
- name: privatestack-bundle-jobs
  check:
    - '{name}-test-app-module'
    - '{name}-test-puppet-module'
    - '{name}-integration-test'
    - 'package-{name}-snapshot':
      - 'gate-{name}-deployment-test'
      - 'gate-{name}-bundle-validation'
  gate:
    - '{name}-test-app-module'
    - '{name}-test-puppet-module'
    - '{name}-integration-test'
    - 'package-{name}-snapshot':
      - 'gate-{name}-deployment-test'
      - 'gate-{name}-bundle-validation'
  post:
    - 'post-{name}-maven-publish-releases':
      - 'post-{name}-trigger-promote-privatestack'
```

Zuul – Pipeline Job & Status



check

3

Newly uploaded patchsets enter this pipeline to receive an initial +/-1 Verified vote from Jenkins.

Queue: dev/virtualdatacenter

dev/virtualdatacenter
2041,2

dev/virtualdatacenter 0 min
2048,4 47 min

virtualdatacenter-test-app-module	success
virtualdatacenter-test-puppet-module	success
virtualdatacenter-integration-test	failure
package-virtualdatacenter-snapshot	success
check-hurricane-virtualdatacenter	queued
gate-virtualdatacenter-deployment-test (non-voting)	success
gate-virtualdatacenter-bundle-validation (non-voting)	success

Jenkins....It's Great

- Does a good job at running what we tell it to run
- Reports everything we need
- Supports complex pipelines

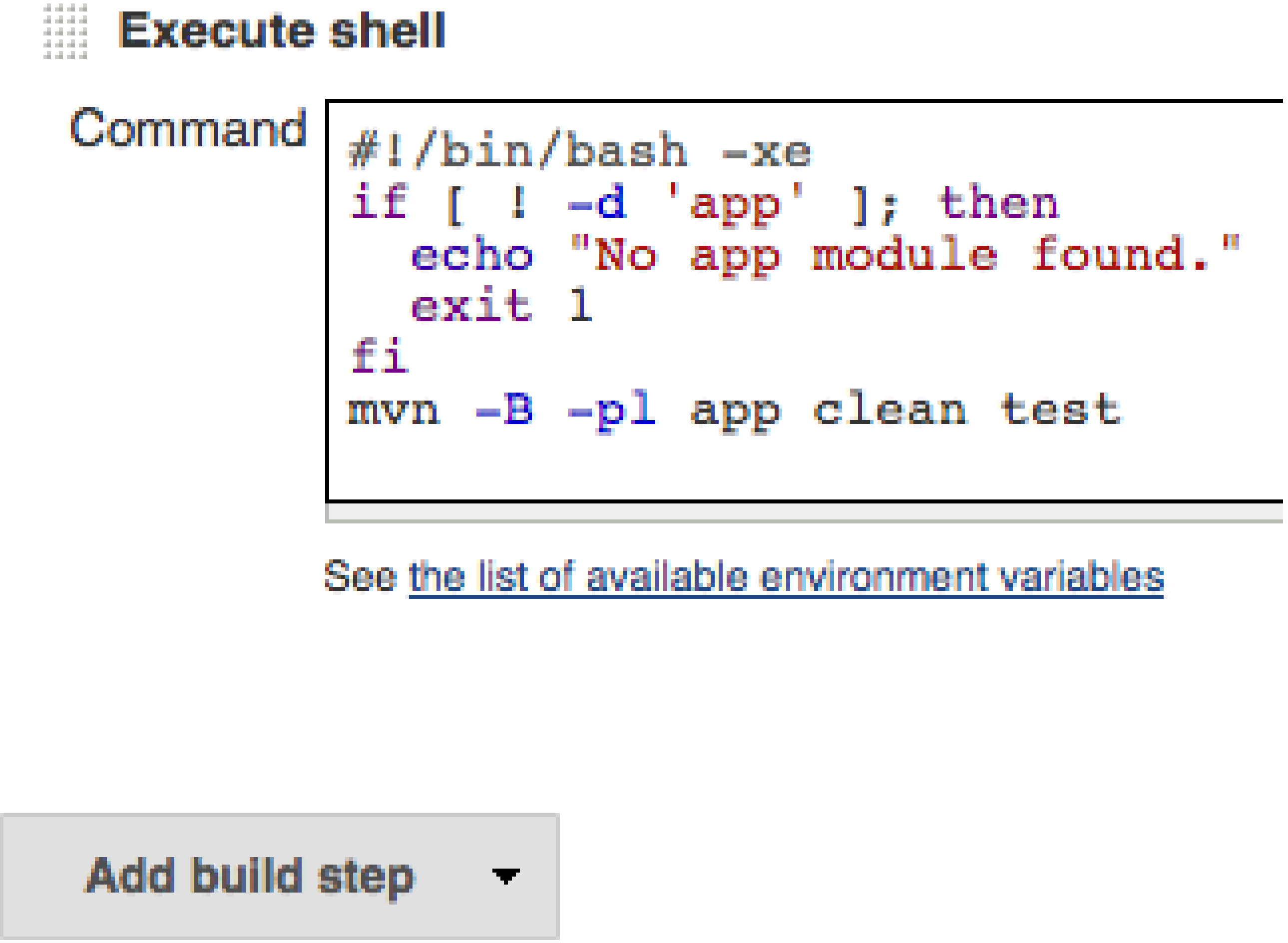
Jenkins….It's Not So Great

- Manual UI entry
- New jobs were created by manually duplicating old jobs
- Human copy/paste errors were high
- Hard & time consuming to create complex jobs

Jenkins + Jenkins Job Builder == BFF

- No more duplicating jobs by hand to create new ones
- No more human copy/paste errors
- Fully automated job creation
- Pipeline complexity moved to Zuul
  - Templating made it even easier
- YAML templates saved in version control

Jenkins – Via UI



Execute shell

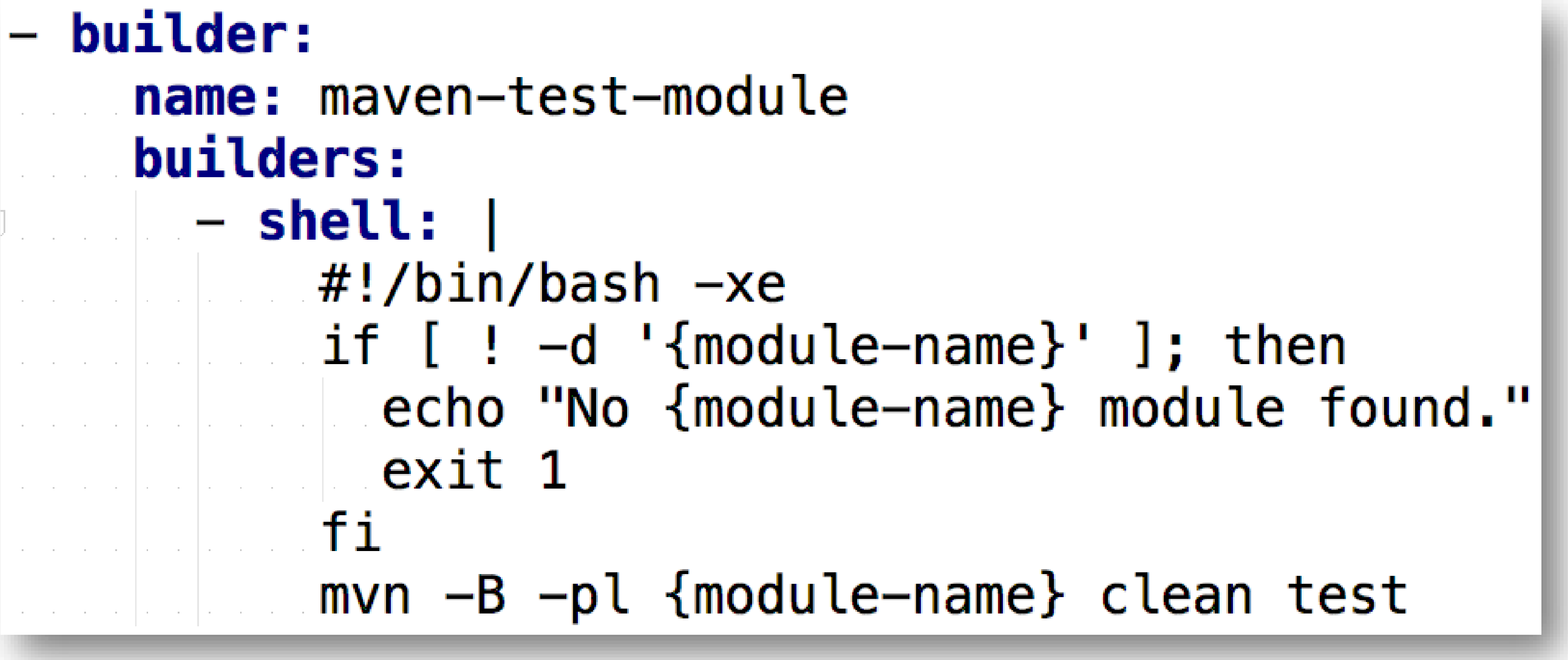
Command

```
#!/bin/bash -xe
if [ ! -d 'app' ]; then
  echo "No app module found."
  exit 1
fi
mvn -B -pl app clean test
```

See the list of available environment variables

Jenkins Job Builder

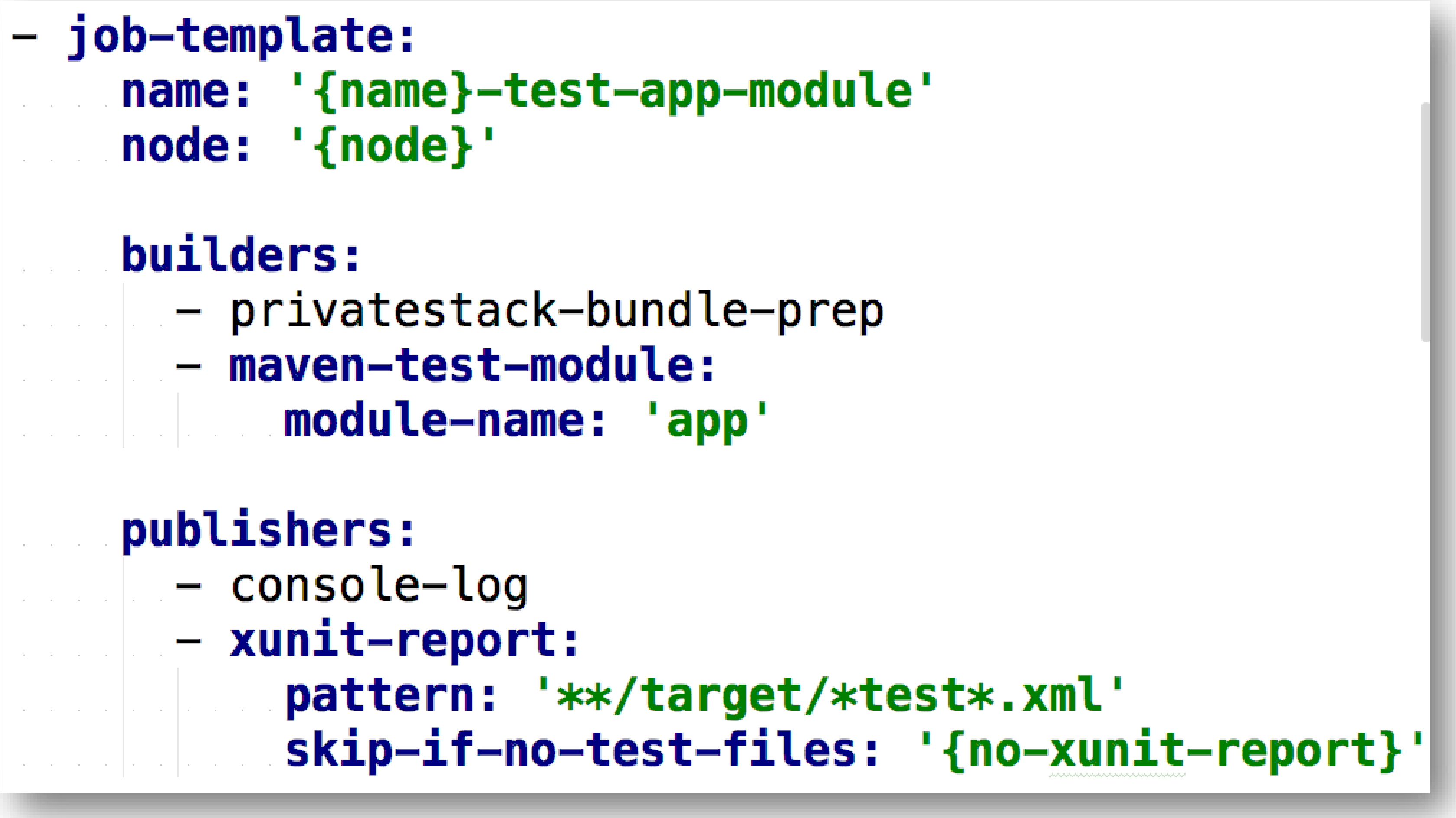
- Define a step (macro) in the job you want to create



```
- builder:
    name: maven-test-module
    builders:
      - shell: |
          #!/bin/bash -xe
          if [ ! -d '{module-name}' ]; then
            echo "No {module-name} module found."
            exit 1
          fi
          mvn -B -pl {module-name} clean test
```

Jenkins Job Builder

- Template the job itself



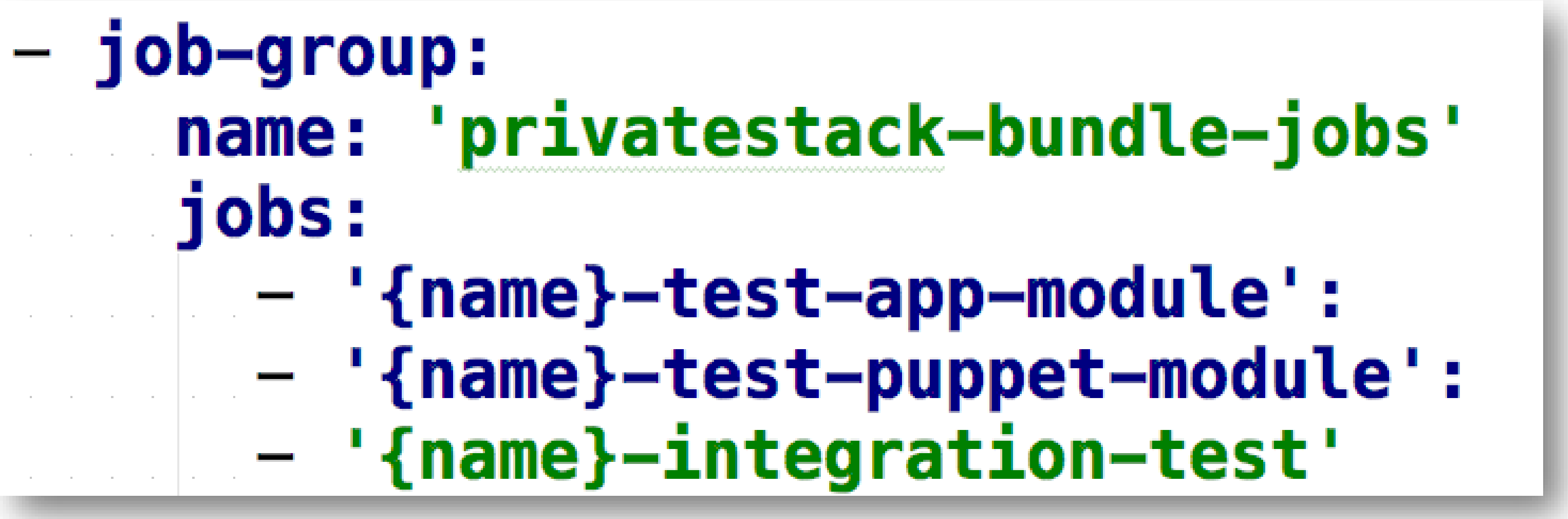
```
- job-template:
    name: '{name}-test-app-module'
    node: '{node}'

    builders:
      - privatestack-bundle-prep
      - maven-test-module:
          module-name: 'app'

    publishers:
      - console-log
      - xunit-report:
          pattern: '**/target/*test*.xml'
          skip-if-no-test-files: '{no-xunit-report}'
```

Jenkins Job Builder

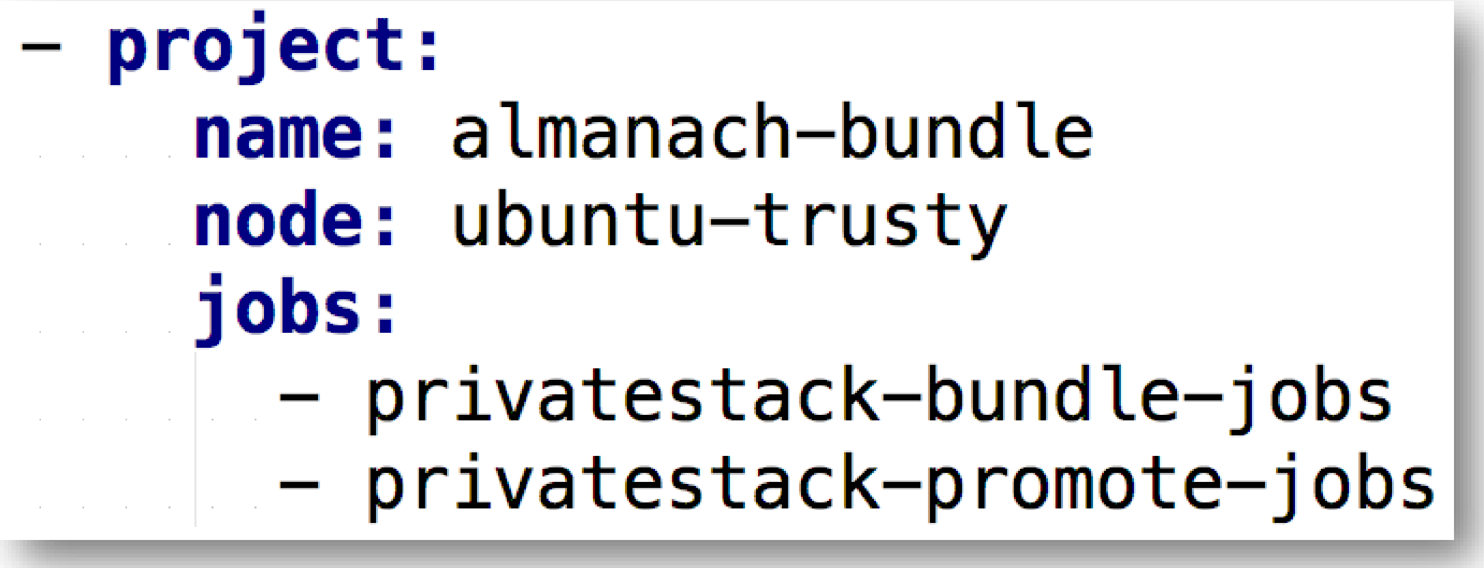
- Group your jobs in a bundle



```
- job-group:
    name: 'privatestack-bundle-jobs'
    jobs:
      - '{name}-test-app-module':
      - '{name}-test-puppet-module':
      - '{name}-integration-test'
```

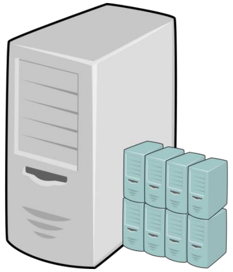
Jenkins Job Builder

- Templates are only defined once and re-used!



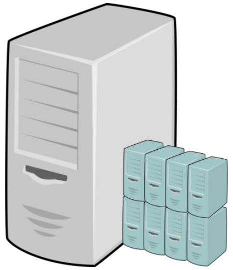
```
- project:
    name: almanach-bundle
    node: ubuntu-trusty
    jobs:
      - privatestack-bundle-jobs
      - privatestack-promote-jobs
```

Before Nodepool



- Limited static test machines
- Too much waiting for builds to run
- Leads to devs "cheating" to the top of the test queue
  - Everybody is sad and waiting.... :-(

Nodepool – Ephemeral Test Machines



- On demand ephemeral builds for testing
- As many as we need when we need them
- No more waiting for a test machine
- No more taking up unnecessary resources when not in use
- Command-line tool for easy maintenance

Wrapping It All Up With Ansible

- Automation tool to provision the CI Stack
  - Dev
  - Production
- Includes tests to validate CI functionality
  - Eat our own dog food by running CI in CI!

Staging & Production Delivery?

What's Next?

- Continue Optimizations
- Bug fixing due diligence
- Speed up certain longer running tests
- More parallelization

Questions?