CI Monkey

Will Soula

Drilling Info

http://www.drillinginfo.com

https://github.com/drillinginfo/ci-monkey

October 23, 2014

Issues

- New repos
  - many new repos being created as we move to new platform
  - repeated work configuring jobs
- Regressions
  - broken builds from pull request merges
  - it merges cleanly but does it cause regressions
- Cowboy Coders
  - suddenly job starts failing and configuration is different
  - no way to track configuration changes

CI Monkey Solutions

- New repos - automatically get ci jobs (CI Monkey)
  - enhanced to detect build type
- Regressions - repos automatically get github pull request builds (GHPRB)
  - test the merge
- Cowbody Coders - configuration is in source control (JJB)
  - Allows you to fix messed up config by just deleting the jenkins folder and running the ci-monkey
  - Allows to change config if everything is the same it can replace it in the yaml files

GHPRB Workflow

- Where is it in the SDLC?
  - Starts and ends in the dev part of the SDLC
  - Small little CI loop
- What is the workflow?
  - Create your feature branch and begin work
  - Upon first commit or after tests committed open pull request
  - Continue pushing to your feature branch as normal
  - Profit from CI on a dev branch
- How to keep pull request from being accepted too soon
  - Put pull request in review card
  - Denote it is in progress in the title

How To Test Codebase

- Monolithic
  - All code in one repo
  - Run like normal build
- Modules and sub-modules
  - If tests are in same repo
    - Pass sha and refspec
    - Run down stream tests against them
  - If running downstream tests
    - Publish to different location
    - Pass to repository manager (bower, ivy, maven, etc)

Beyond Tests

- Deployment of documentation
  - Why
    - People without access to github
    - People not tech savvy
  - How
    - Do doc build
    - If it is a pull request figure out the pull request number
      - Change the deploy location
      - Deploy to "pr" folder
    - Deploy as normal
    - If last commit was a pull request, delete the pull request deploy

Beyond Tests

- Customer demos, both internal and external
    - Deploy your latest code continuously so other dev can work on consuming your latest code (dual agile), instead of waiting for one to finish before other starts
    - If using monolithic codebase can deploy what is coming next to a site for consumption and feedback on the beta
- Comment in pull request buid number deployed in, but only if already commented
    - Get pull request number from last git message
    - Use that to comment on pull request using github api

What is Jenkins Job Builder

- Opensource Jenkins job configurer created and used by openstack
- Store job configuration in code base
  - allows to see how job is configured without needing to go to jenkins
  - can have multiple sets of the same job for different branches
- Uses Python under the covers
- Job config is written in yaml

Features of Jenkins Job Builder

- Opensource and easy to add new plugins to
  - I am an active contributor
  - Learn Git, Gerrit and rewriting history
- Macros for common tasks
  - pull in common macro file
  - multiple jobs per file
- Templates for common jobs
  - like macros but for entire jobs
  - multiple jobs per file
- Test yaml created
  - can verify there are no syntax errors
  - aids in adding new plugin

How to use Jenkins Job Builder

Installation:

- install python
- clone from github
- sudo python setup.py install

Configuration (/etc/jenkins_jobs/jenkins_jobs.ini):

```
1 [jenkins]
2 user=USERNAME
3 password=PASSWORD
4 url=JENKINS_URL
5 ignore_cache=IGNORE_CACHE_FLAG
```

How to use JJB

Commands

- jenkins-jobs update jenkins/
- jenkins-jobs test jenkins/ -o output
- update Jenkins Job Builder itself

Job Configuration with JJB

Job

```
- job
    name: test-job
    description: A test job of JJB
    project-type: freestyle
    properties:
      - github:
          url: https://git.drillinginfo.com/DIGlobal/map-widget/
      - inject:
          script-content: |
                      #!/bin/bash -ex
                      {{ rm -rf /jenkins/workspace/$JOB_NAME || true; }}
                      mkdir -p /jenkins/workspace/$JOB_NAME
                      echo default_branch=`curl -s -k -H "Authorization: token [token]" "
          properties-file: "/jenkins/workspace/$JOB_NAME/properties.prop"
    scm:
      - git:
          url: ssh://[github_url]/DIGlobal/map-widget.git
          branches:
            - "$default_branch"
          wipe-workspace: true
          browser: githubweb
          browser-url: https://[github_url]/DIGlobal/map-widget/
          git-tool: System Default Git
    triggers:
      - github
      - pollscm: ''
    builders:
```

```
28       - shell: |
29                 echo Hello World
30     wrappers:
31       - build-name:
32           name: ${ENV,GIT_BRANCH"}-#$BUILD_NUMBER
```

Macro Use with JJB

Macro

```
 1 - builder:
 2     name: builder-macro
 3     builders:
 4       - ant:
 5           targets: "{task}"
 6           properties:
 7              failonerror: false
 8           ant-name: "Ant 1.6"
 9 - job
10     name: test-macroA
11     description: Run the integration tests
12     project-type: freestyle
13     builders:
14       - builder-macro:
15           task: testi
16 - job
17     name: test-macroB
18     description: Run the unit tests
19     project-type: freestyle
20     builders:
21       - builder-macro:
22           task: testu
```

Job Template Use with JJB

Job Template, like macro but for entire job:

```
 1 - builder:
 2     name: ant-build-macro
 3     builders:
 4       -  ant: "{targets}"
 5 - wrapper:
 6     name: build-name-macro
 7     wrappers:
 8       - build-name:
 9           name: ${ENV,var="GIT_BRANCH"}-#$BUILD_NUMBER
10 - job-template:
11     name: "{name}-build"
12     description: Build the {name} project
13     project-type: freestyle
14     builders:
15       - ant-build-macro:
16           targets: "test"
17       - build-name-macro
18 - job-template:
19     name: "{name}-jjb"
20     description: Build the {name} gradle build
21     project-type: freestyle
22     builders:
23       - ant-build-macro:
24           targets: "jjb"
25       - build-name-macro
26
27 - project:
```

```
28      name: testAnt
29      jobs:
30        - '{name}-build'
31        - '{name}-jjb'
```

End Game

Centralized configuration, simple job file

```
 1 - project:
 2     name: map-widget
 3     repo: map-widget
 4     organization: DIGlobal
 5     type: javascript
 6     build-results-trigger: ""
 7     jobs:
 8       - "{name}-build"
 9       - "{name}-pull-request"
10       - "{name}-jjb"
```

How DI uses Jenkins Job Builder

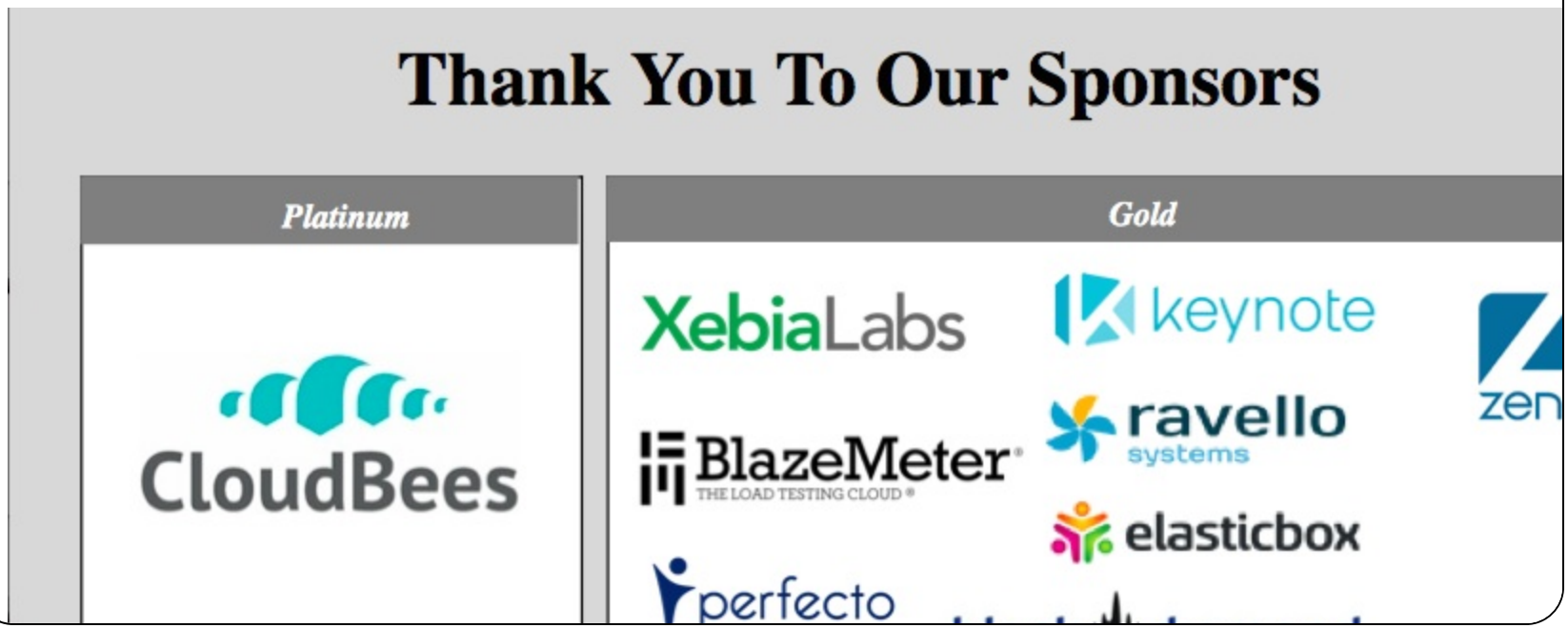
- No one can manually configure a job
  - have to refer new employees to documentation about JJB
  - few people have manual configure ability
- jjb-init job
  - takes branch, repo, and org as parameters
  - runs JJB against jenkins folder
- clean cache before running JJB
- run every weekend in case configuration somehow got changed
  - few trusted people mess up
  - job left broken from jjb-init job

CI Monkey

- Crawls github organizations looking for repos without a jenkins folder
- Lays down a template for the type of repo it is, or a generic one if it cannot determine
- Replaces values in template
- Common tasks
  - clone from git
  - comment on pull requests
  - github url
  - creating github webhook
  - triggering main build from push
  - triggering pull request build from push
  - put name of branch in build name
  - hipchat notifications

CI Monkey

- Lays down three jobs
  - one to build the default branch of a repo
    - Always wipe the workspace on clone
    - Allows to change branch being built by changing the default branch in github
  - one to build pull requests
    - Can run just tests
    - Could publish for further testing of pull request
    - Run JJB if changing jenkins folder
  - one to run JJB
    - pull request runs this too but the change
    - Afterwards, main build runs and verifies config changes


Thank You To Our Sponsors
Platinum
CloudBees
Gold
XebiaLabs
keynote
ravello
systems
BlazeMeter
THE LOAD TESTING CLOUD
elasticbox
perfecto
zen