



CI Monkey

Will Soula

Drilling Info

<http://www.drillinginfo.com>



Will Soula

Drilling Info

<http://www.drillinginfo.com>

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 build 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
- Test yaml created
 - can verify there are no syntax errors
 - aids in adding new plugin
- Templates for common jobs
 - like macros but for entire jobs
 - multiple jobs per file

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
```

Job Configuration with JJB

Job

```
1 - job
2   name: test-job
3   description: A test job of JJB
4   project-type: freestyle
5   properties:
6     - github:
7       url: https://git.drillinginfo.com/DIGlobal/map-widget/
8   scm:
9     - git:
10      url: ssh://git@git.drillinginfo.com/DIGlobal/map-widget.git
11      branches:
12        - "HEAD"
13      wipe-workspace: true
14      browser: githubweb
15      browser-url: https://git.drillinginfo.com/DIGlobal/map-widget/
16      git-tool: System Default Git
17  triggers:
18    - github
19    - pollscm: ''
20  builders:
21    - shell: |
22      echo Hello World
23  wrappers:
24    - inject:
25      script-content: echo REAL_BRANCH=`git rev-parse HEAD | git branch --contains |
26    - build-name:
27      name: ${PROFFILE,file="branch.prop",property="REAL_BRANCH"}-#${BUILD_NUMBER}
```

Macro Use with JJB

Macro

```
1 - builder:
2   name: builder-macro
3   builders:
4     - shell: |
5         echo "You passed {task}"
6 - job
7   name: test-macroA
8   description: A test of echoing foo
9   project-type: freestyle
10  builders:
11    - builder-macro:
12      task: foo
13 - job
14   name: test-macroB
15   description: A test of echoing bar
16   project-type: freestyle
17   builders:
18     - builder-macro:
19       task: bar
```

Job Template Use with JJB

Job Template, like macro but for entire job:

```
1 - builder:
2   name: ant-build
3   builders:
4     ant: "{targets}"
5 - job-template:
6   name: "{name}-build"
7   description: Build the {name} project
8   project-type: freestyle
9   builders:
10    - ant-build:
11      targets: "test"
12 - job-template:
13   name: "{name}-jjob"
14   description: Build the {name} gradle build
15   project-type: freestyle
16   builders:
17    - ant-build:
18      targets: "jjob"
19
20 - project:
21   name: testAnt
22   jobs:
23     - '{name}-build'
24     - '{name}-jjob'
```

How to use JJB

Commands

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

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 on poll
 - put name of branch in build name

CI Monkey

- Lays down three jobs
 - one to build the HEAD of a repo
 - HEAD on a freshly checked out repo is the default branch
 - Always wipe the workspace on clone
 - Allows to change branch being built by changing the default branch in github and wiping the job workspace
 - 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

Demo