GlassFish Build Infrastructure

Arindam Bandyopadhyay

October 25, 2017

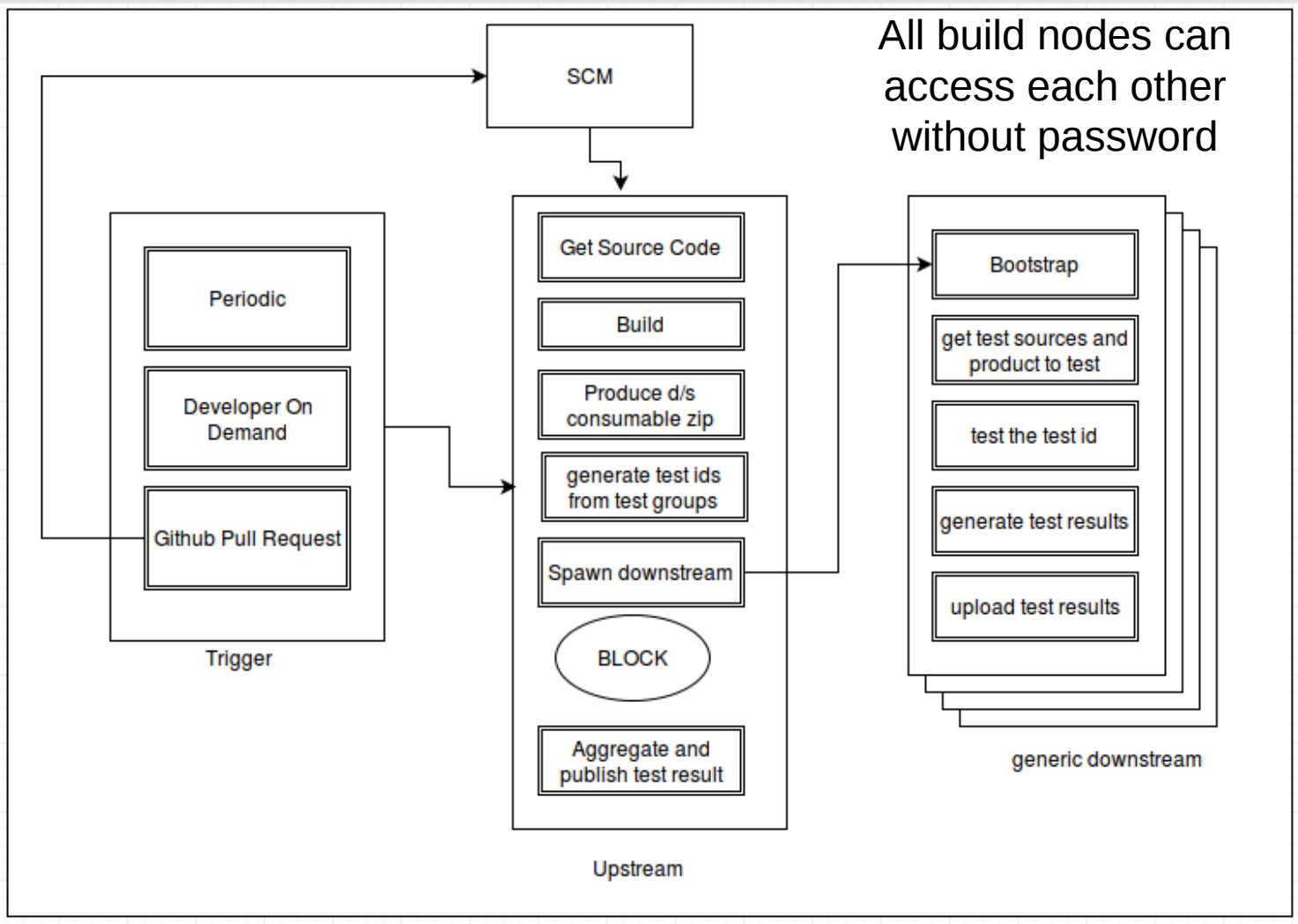
Safe Harbor Statement

The following is intended to outline our general product direction.It is intended for information purposes only, and may not be incorporated into any contract.It is not a commitment to deliver any material, code, or functionality and should not be relied upon in making purchasing decisions.The development, release and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

What is GlassFish CI Pipeline?

- Jenkins based CI which runs all GlassFish tests inside docker container for every pull request.
- Runs all GlassFish dev tests(10,000+) for each of the Pull Requests before merge.
- Runs the tests quickly (10,000+ tests within 1 hour and 30 minutes).
- Runs Findbug and copyright check tool for every checkin.
- Every commit is fully tested.
- Pushes the binary to public download location every night.
- rq.sh
  - command line utility to trigger all(or some) of the GlassFish test suites from CLI
- GitHub integration
  - Conversation with the bot to request for test run
  - All tests run for the PR in remote queue
  - Updates the PR with the results

Pipeline Design Diagram



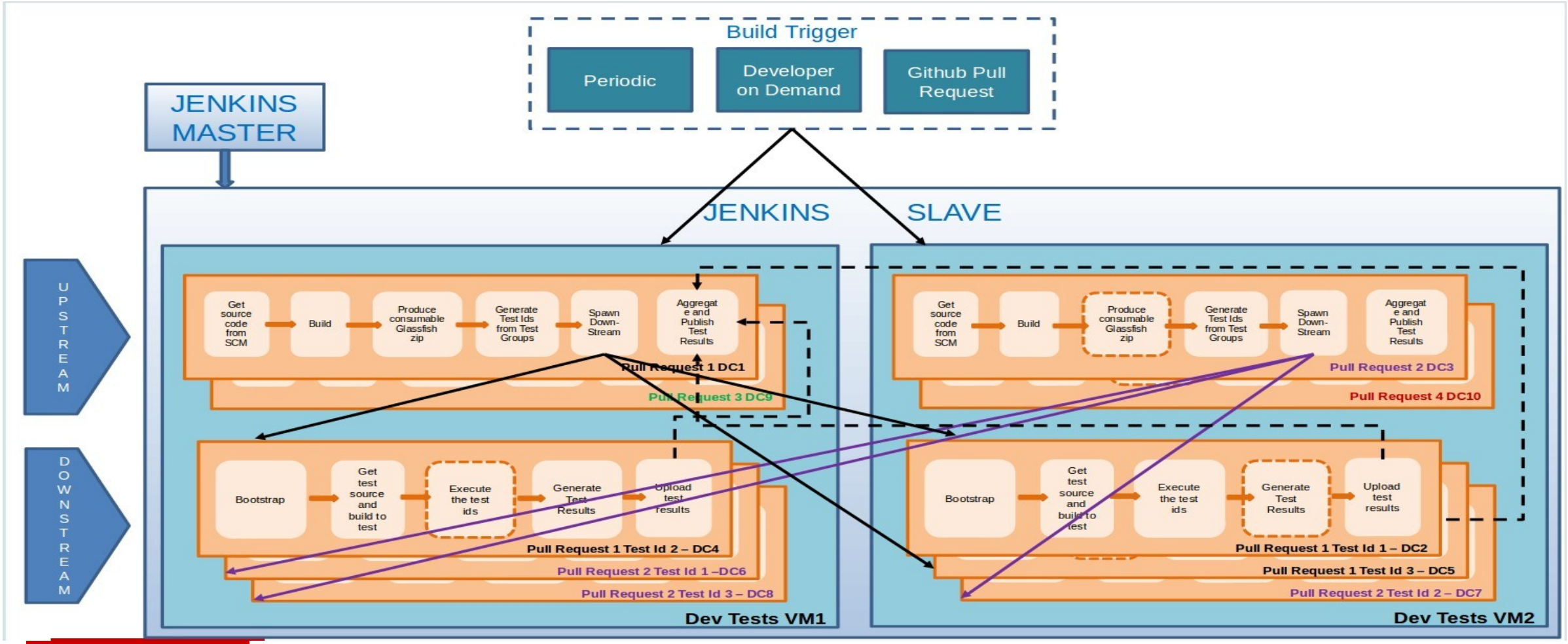
The CI Jobs

- gf-master-continuous-check
  - Continuously verifies the sanity of the pipeline
  - Detects and tracks unstable tests
  - Provides history of builds/tests
- gf-remote-queue
  - Builds Pull request and runs all tests
  - Takes developer branch as param
  - Takes “target” as param
  - Does not “auto-push”
- gf-generic-test-job
  - Runs concurrent generic tests
- Gf-master-github-pr
  - Triggered by GitHub pull requests (creation)
- gf-nightly
  - nightly job takes binary from gf-master-continuous-check and pushes the binary to public download location
- gf-promotion
  - Weekly job that does a maven release and pushes the binary to to public download location.
  - The binary is CTS tested.

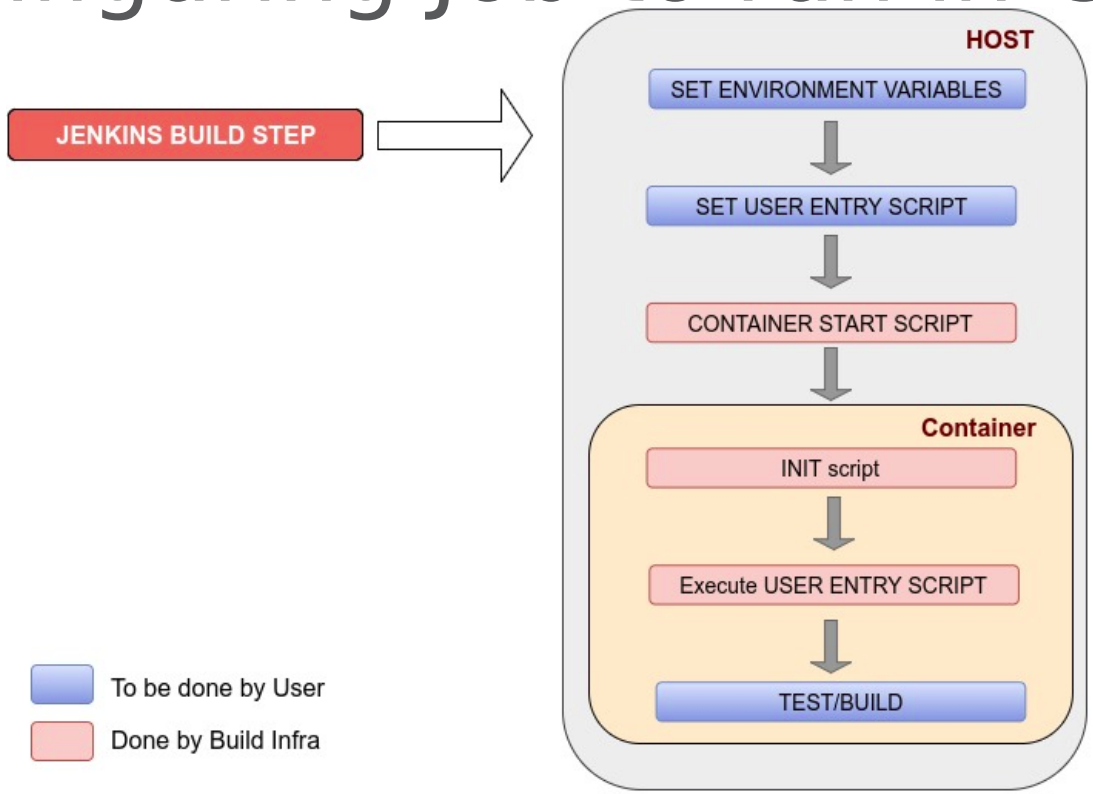
Interpreting Pipeline Status

Status	Meaning
Success	1. The build is successful 2. All the downstream tests ran successfully
Failure	1. Build failure 2. Infrastructure failure
Aborted	1. If the upstream job is 'aborted' by jenkins for some reason
Unstable	1.Unit tests failed 2. Any of the downstream jobs has test failures 3. One or more downstream test jobs 'Failed' due to configuration 4. Any of the downstream job was 'Aborted'. (jenkins queue timeout or test timeout)

The complete picture (pipeline inside docker)



Configuring Job to run in Container



- Set Environment Variables
- Pass the list of environment variables that need to be set inside the container. Jenkins Build and global variables are automatically made available inside the container
- Set Entry Script
- Full path to the user script that will be called by the container. This will be acting as the entry script for the user.
- Script needs to be either in workspace or part of the NFS location
- Entry script should contain the command to run build/tests.

Compute

	#CPU Core = 2 Memory = 8 GB	#CPU Core = 4 Memory = 16 GB	#CPU Core = 8 Memory = 64 GB	Total
RHELS 6.9	-	-	1	01
RHELS 7.0	-	1	-	01
RHELS 7.2	26	19	-	45
Total	**26**	**20**	**1**	**47**

Lines of Code

- All JavaEE repositories selected for EE4J
    - Files – 62478
    - Blank – 1173854
    - Comments – 3364071
    - Code – 4827540

- CTS and TCK
    - Files – 32639
    - Blank – 481928
    - Comments – 664165
    - Code - 3977088


ORACLE