From Jenkins to BuildBot: just our experience

Timofey Turenko
MariaDB Corporation

Andrey Vasilyev
Maxim Kosterin
Evgeny Vlasov

Open Innovations Association FRUCT
Product to build & test
Product to build & test

- incredible load and weird broken quires
- any combination of router/filters/core settings
- several Maxscales (HA setup)
- different Linux distributions
- different HW/Virtual Machines
- different products (MariaDB/MySQL/Percona server)
- different versions
- different topologies
- different failures simulation
Jenkins and Everything as code

Long long time ago…

- Jenkins Job Builder from OpenStack? YAML, easy to load…
  - no parametrized !include
  - only jobs can be loaded (plugins, views, configs – Jenkins CLI from own scripts?)
  - YAML is not really human-writable

- Pipelines? even less can be code! => JJB again!
Angry Jenkins

- Still no full "Everything is code solution"
- Mess of plugins
- how to implement complex triggering logic? Even more plugins!
  - Matrix jobs, nested jobs, parametrized triggers, …
  - even more plugins
- all – triggers, builders, parameters definitions are in the one place
- bugs and resources consumption
  - one test goes mad – huge log – Jenkins down! (or whole host!)
same job, but different triggers

not a jobs, but step

not jobs, but but schedulers
Job DSL? Other CI?
BuildBot...
The agony of choice

- Everything is code
- Complex triggers:
  - any schedule for any test configuration
  - any host selection algorithm
- Can survive very long runs, huge logs
- Can process huge number of parameters
  - one test goes mad – huge log – Jenkins down! (or whole host!)
BuildBot

- MariaDB server uses BuildBot
- Initially no UI, only code
- Continuing with Jenkins means moving to Job DSL
- Very easy to implement any triggering logic: Python code can generate triggers
The Devil in the details
Nested tasks: parameters transfer

COMMON_PROPERTIES = [  
    "name",
    "repository",
    "branch",
    "target",
    ...

factory.addStep(steps.Trigger(  
    name="Call the 'build' scheduler",
    schedulerNames=['build'],
    copy_properties=COMMON_PROPERTIES,
    ...

from . import properties

BUILD_AND_TEST_PROPERTIES = [  
    properties.build_name(),
    properties.build_target(),
    ...

MANUAL_SCHEDULER = schedulers.ForceScheduler(  
    name="build_and_test",
    label="Build and test",
    builderNames=['build_and_test'],
    codebases=properties.codebaseParameter(),
    properties=BUILD_AND_TEST_PROPERTIES
)
Schedulers debugging

- not possible to see “next scheduled build”
- only way now: dry runs with debug output

```python
BUILD_INTERVAL = 1
launchTime = 18
for branch in constants.NIGHTLY_SCHEDS:
    ...
    nightlyScheduler = schedulers.Nightly(
        name="build_and_performance_test_{0}_nightly".format(branch),
        builderNames=["build_and_performance_test"],
        hour=launchTime % 24, minute=0,
        codebases={"": {
            "branch": branch,
            "repository": constants.MAXSCALE_REPOSITORY
        }},
        properties=nightlyProperties
    )
    SCHEDULERS.append(nightlyScheduler)
```
Task creation steps

- write scheduler
- write builder (collect all steps to builder – build, publish, report, cleanup)
- if needed, describe new parameters
- do not forget to add new scheduler, builder into __init__.py

- scheduler and builder are in one YAML
- builders, publishers, wrappers – can be in some YAML template, hard to forget anything
UI :(  

- not possible to change parameters for rebuild  
- schedulers view: only names, deleted items still in the list  
- not the best log viewer
<table>
<thead>
<tr>
<th>Enabled</th>
<th>Scheduler Name</th>
<th>Master</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔️</td>
<td>2.2</td>
<td>maxscale-ci.mariadb.com/home/vagrant/maxscale-buildbot/master</td>
</tr>
<tr>
<td>✔️</td>
<td>2.3</td>
<td>maxscale-ci.mariadb.com/home/vagrant/maxscale-buildbot/master</td>
</tr>
<tr>
<td>✔️</td>
<td>2.4</td>
<td>maxscale-ci.mariadb.com/home/vagrant/maxscale-buildbot/master</td>
</tr>
<tr>
<td>✔️</td>
<td>build</td>
<td>maxscale-ci.mariadb.com/home/vagrant/maxscale-buildbot/master</td>
</tr>
<tr>
<td>✔️</td>
<td>build_all</td>
<td>maxscale-ci.mariadb.com/home/vagrant/maxscale-buildbot/master</td>
</tr>
<tr>
<td>✔️</td>
<td>build_all_triggerable</td>
<td>maxscale-ci.mariadb.com/home/vagrant/maxscale-buildbot/master</td>
</tr>
<tr>
<td>✔️</td>
<td>build_and_performance_test</td>
<td>maxscale-ci.mariadb.com/home/vagrant/maxscale-buildbot/master</td>
</tr>
<tr>
<td>✔️</td>
<td>build_and_performance_test_2.2_nightly</td>
<td>maxscale-ci.mariadb.com/home/vagrant/maxscale-buildbot/master</td>
</tr>
<tr>
<td>✔️</td>
<td>build_and_performance_test_2.3_nightly</td>
<td>maxscale-ci.mariadb.com/home/vagrant/maxscale-buildbot/master</td>
</tr>
<tr>
<td>✔️</td>
<td>build_and_performance_test_2.4_nightly</td>
<td>maxscale-ci.mariadb.com/home/vagrant/maxscale-buildbot/master</td>
</tr>
<tr>
<td>✔️</td>
<td>build_and_performance_test_develop_nightly</td>
<td>maxscale-ci.mariadb.com/home/vagrant/maxscale-buildbot/master</td>
</tr>
<tr>
<td>✔️</td>
<td>build_and_performance_test_on_push</td>
<td>maxscale-ci.mariadb.com/home/vagrant/maxscale-buildbot/master</td>
</tr>
<tr>
<td>✔️</td>
<td>build_and_test</td>
<td>maxscale-ci.mariadb.com/home/vagrant/maxscale-buildbot/master</td>
</tr>
<tr>
<td>✔️</td>
<td>build_and_test_2.2_nightly</td>
<td>maxscale-ci.mariadb.com/home/vagrant/maxscale-buildbot/master</td>
</tr>
<tr>
<td>✔️</td>
<td>build_and_test_2.3_nightly</td>
<td>maxscale-ci.mariadb.com/home/vagrant/maxscale-buildbot/master</td>
</tr>
<tr>
<td>✔️</td>
<td>build_and_test_2.4_0_nightly</td>
<td>maxscale-ci.mariadb.com/home/vagrant/maxscale-buildbot/master</td>
</tr>
<tr>
<td>✔️</td>
<td>build_and_test_2.4_nightly</td>
<td>maxscale-ci.mariadb.com/home/vagrant/maxscale-buildbot/master</td>
</tr>
<tr>
<td>✔️</td>
<td>build_and_test_develop_nightly</td>
<td>maxscale-ci.mariadb.com/home/vagrant/maxscale-buildbot/master</td>
</tr>
<tr>
<td>✔️</td>
<td>build_and_test_distro_2.3_suse_15_libvirt_weekly</td>
<td>maxscale-ci.mariadb.com/home/vagrant/maxscale-buildbot/master</td>
</tr>
<tr>
<td>✔️</td>
<td>build_and_test_distro_2.3_ubuntu_bionic_libvirt_weekly</td>
<td>maxscale-ci.mariadb.com/home/vagrant/maxscale-buildbot/master</td>
</tr>
<tr>
<td>✔️</td>
<td>build_and_test_snapshot_force</td>
<td>maxscale-ci.mariadb.com/home/vagrant/maxscale-buildbot/master</td>
</tr>
</tbody>
</table>
Time flys....

2018

- May 04: Decision, initial commit
- May 10: Server up
- May 31: Tasks transferred from Jenkins
- June 14: Last commit to Jenkins config
- July 14: All schedulers created
- August 06: Jenkins down

September

- Bug fixing, a couple of broken Maxscale releases

October

- "scripts spaghetti" converted into Python code (logs parsing, results publishing)
Boring numbers

- 76 Python files (4.9 KLOC)
- 2 Bash files (36 LOC)
- 540 commits from Aug, 2018

- 207 YAML files (2.2 KLOC)
- 16 Bash files (520 LOC)
- 9 Ruby files (1.9 KLOC)
- 593 commits (Apr 2016 – Apr 2018) – only YAMLs
Win?

- Stability!!
- everything in one language
- no pile of plugins
- less host resources consumption

Next?

- schedulers viewer (plugin?)
- automated parameters transfer for nested jobs
THANK YOU!

https://github.com/mariadb-corporation/maxscale-buildbot
https://github.com/mariadb-corporation/maxscale-jenkins-jobs
https://github.com/OSLL/jjg - JJB YAMLs loader
https://github.com/mariadb-corporation/MaxScale