squash the flakes!

FOSDEM 2024
Daniel Hiller
agenda

- about me
- about flakes
- impact of flakes
- flake process
- tools
- the future
- Q&A
- want to help?
about me

- Software Engineer @ Red Hat OpenShift Virtualization team
- KubeVirt CI, automation in general
about flakes

a question:

who can explain what a flake is?
about flakes

a flake

...
about flakes

A *flake* is a test that, without any code change, will either *fail* or *pass* in successive runs.
about flakes

another question:

who thinks handling flakes is important?
about flakes

another question:

who has had to deal with flakes?
about flakes

another question:

who has to deal with flakes on a regular basis?
about flakes

flakes are caused

either by production code (a bug)

or flaky test code (also a bug, but handled differently)
about flakes

source: https://prow.ci.kubevirt.io/pr-history/?org=kubevirt&repo=kubevirt&pr=9445
impact of flakes

from “a survey of flaky tests”:  
- 97% of flakes were **false alarms**, and
- more than 50% of flakes could not be **reproduced in isolation**

this leads to the conclusion: “ignoring flaky tests is ok”

source: “A survey of flaky tests”
impact of flakes
impact of flakes

in CI automated testing must give a reliable signal of stability

any failed test run signals that the product is unstable

test runs failed due to flakes do not give this reliable signal

they only waste time
impact of flakes

Flaky tests waste everyone’s time - they cause

- longer feedback cycles for developers
- slowdown of merging pull requests - “retest trap”
- reversal of acceleration effects (i.e. batch testing)
impact of flakes

Flaky tests also cause trust issues - they make people

- lose trust in automated testing
- ignore test results
minimizing the impact

tldr; exclude (aka quarantine) a flaky test from test runs as early as possible, but only as long as necessary
minimizing the impact

what do we need?

- ability to move a test from set of stable tests into set of quarantined tests and back
- a report over possible flaky tests
- enough runtime data to triage flakes
  - devs decide whether we quarantine right away or they can fix them in time
minimizing the impact

how can we find flaky tests?

any merged PR had all tests succeeding in the end,

thus any test run with test failures from that PR might contain execution of flaky tests
the flake process

regular meeting
- look at flakes
- decide: fix or quarantine?
- hand to dev
- bring back in

emergency quarantine

source: QUARANTINE.md

FOSDEM 2024
tools

ci honoring QUARANTINE label

- **presubmits skip** quarantined tests
- **periodics execute** quarantined tests to check their stability

---

sources:
- https://github.com/kubevirt/kubevirt/blob/38c01c34acecfaf89078b1bbaba8d9cf3cf0d4d/automation/test.sh#L452
- https://github.com/kubevirt/kubevirt/blob/38c01c34acecfaf89078b1bbaba8d9cf3cf0d4d/hack/functests.sh#L69
- https://github.com/kubevirt/kubevirt/blob/38c01c34acecfaf89078b1bbaba8d9cf3cf0d4d/tests/canary_upgrade_test.go#L177
tools

flake stats report

the high level overview

(source)
tools

flakefinder report

the detail overview

gives an overview of the current flaky tests

<table>
<thead>
<tr>
<th>Serial</th>
<th>sig-compute</th>
<th>rfe_id : 393</th>
<th>crit : high</th>
<th>vendor : <a href="mailto:cnv-qe@redhat.com">cnv-qe@redhat.com</a></th>
<th>level : system</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM Live Migration with a dedicated migration network should migrate over that network</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
tools

ci-health

record metrics over merge-queue-length, time-to-merge, retests-to-merge and merges-per-day

FOSDEM 2024
tools

Overview of Quarantine tests

Total: 1 tests
https://github.com/kubevirt/kubevirt/tree/c35be138d85864b17946ae1fe07f99a75445d501/tests/operator/operator.go

```go
var _ = Describe("[Serial][sig-operator]Operator", Serial, decorators.SigOperator, func() {
  Describe("[rfe_id:2897][crit:medium][vendor:cnv-qe@redhat.com][level:component]Dynamic feature
detection", func() {
    It("[test_id:3153][QUARANTINE] Ensure infra can handle dynamically detecting
DataVolume Support", func() {
```

Last updated: 2023-09-08 10:18:02.458076498 +0000 UTC m=+3.056989100
tools

testgrid

drill down on all jobs for kubevirt/kubevirt that are running inside KubeVirt Prow

FOSDEM 2024
check-tests-for-flakes test lane

tries to catch flakes before they enter the codebase

(source)

a test lane that

- selects the changed e2e test files from the commit set
- runs changed e2e tests five times
- runs in random execution order to catch order-dependent test
tools

ci-search

search for terms in prow
job logs (see openshift
ci-search)
in a nutshell

In regular intervals:

- follow up on previous action items
- derive action items from data available
- hand action items over to dev teams
- revisit and dequarantine quarantined tests
the future - more data, more tooling

gaps we want to close:

- collect more data - run the majority of tests frequently
- close the retest gap - stop retesting after a certain threshold
- get better in detecting new flakes
- long term - automatic quarantine PRs when new flakes have entered the codebase
Q&A

Any questions?

Who else is trying to tackle this problem?

What have you done to solve this problem?
Thank you for your patience!

Feel free to send questions and comments:

mailto: dhiller@redhat.com
k8s slack: @dhiller
mastodon: @dhiller@fosstodon.org
web: www.dhill.de
interested in **kubevirt.io**?
want to help?

- join [#kubevirt-dev](#) Slack channel
- join [kubevirt-dev](#) Google group
- fix flakes on [kubevirt/kubevirt](#)