

squash the flakes!

stackconf 2024

[Daniel Hiller](#)



agenda

- about me
- about flakes
- impact of flakes
- flake process
- tools
- the future
- Q&A

about me

- Software Engineer @ [Red Hat OpenShift Virtualization](#) team
- [KubeVirt](#) CI, automation in general

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

a *test*

can also **fail** for reasons beyond our control

that is not a flake to us

about flakes

PR History: kubevirt/kubevirt #9445

9d41878

pull-kubevirt-e2e-k8s-1.25-sig-compute-migrations	1637934812398358528	1636633531918585856	1636403749595385856
pull-kubevirt-e2e-k8s-1.25-sig-compute	1637934815221125120	1636403757321293824	
pull-kubevirt-e2e-k8s-1.25-sig-network	1637934813975416852	1636403756985749504	
pull-kubevirt-e2e-k8s-1.25-sig-operator	1637934815393091584	1636403757992382464	
pull-kubevirt-e2e-k8s-1.25-sig-storage	1637934814088663040	1636633532048609280	1636403756704731136
pull-kubevirt-e2e-k8s-1.26-sig-compute	1637934816471027712	1636404222087925760	
pull-kubevirt-e2e-k8s-1.26-sig-network	1637934816085151744	1636403758915129344	

pull-kubevirt-e2e-k8s-1.25-sig-compute-migrations #1636633531918585856

Job History PR History Artifacts

Test started last Friday at 8:40 AM passed after 1h9m49s. ([more info](#))

JUnit

91/1406 Tests Passed!

1315/1406 Tests Skipped.

Build Log

Show all hidden lines [Raw build-log.txt](#)

pull-kubevirt-e2e-k8s-1.25-sig-compute-migrations #1636403749595385856

Job History PR History Artifacts

Test started last Thursday at 7:33 PM failed after 1h18m18s. ([more info](#))

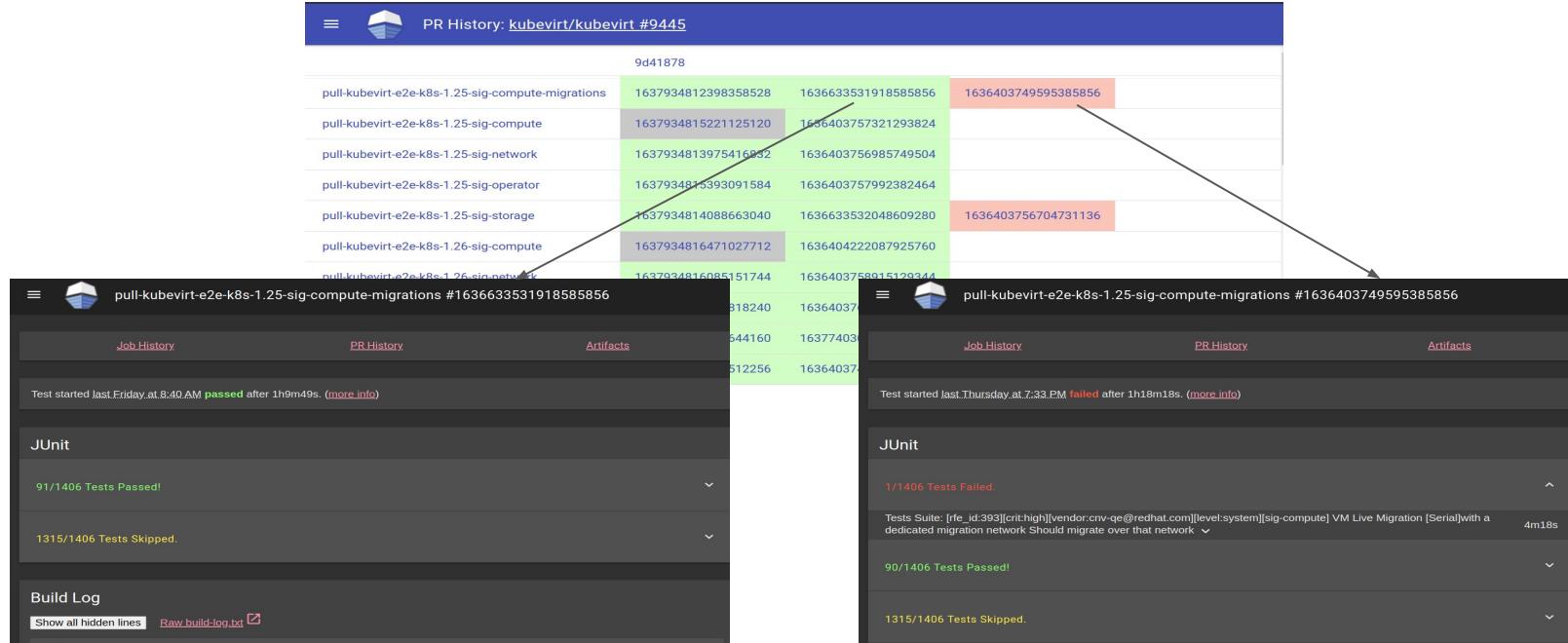
JUnit

1/1406 Tests Failed.

Tests Suite: [rfe_id:393][crit:high][vendor:cnv-qe@redhat.com][level:system][sig:compute] VM Live Migration [Serial]with a dedicated migration network Should migrate over that network

90/1406 Tests Passed!

1315/1406 Tests Skipped.



source: <https://prow.ci.kubevirt.io/pr-history/?org=kubevirt&repo=kubevirt&pr=9445>

about flakes

is it important?

about flakes

does it occur regularly?

about flakes

how often do you have to deal with it?

about flakes

"... test flakiness was a **frequently encountered problem**, with

- **20%** of respondents claiming to experience it **monthly**,
- **24%** encountering it on a **weekly** basis and
- **15%** dealing with it **daily**"

source: "[A survey of flaky tests](#)"

about flakes

"... In terms of severity, of the **91% of developers** who claimed to deal with flaky tests at least a few times a year,

- **56%** described them as a **moderate** problem and
- **23%** thought that they were a **serious** problem. ..."

source: "[A survey of flaky tests](#)"

about flakes

flakes are caused

either by **production code**

or by **test code**

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

conclusion: "ignoring flaky tests is ok"

*code under test actually is not broken, but it works as expected

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 **cause trust issues** - they make people

- **lose** trust in automated testing
- **ignore** test results

minimizing the impact

def: **quarantine**¹

to exclude a flaky test from test runs **as early as possible**, but **only as long as necessary**



¹: Martin Fowler - Eradicating Non-Determinism in Tests

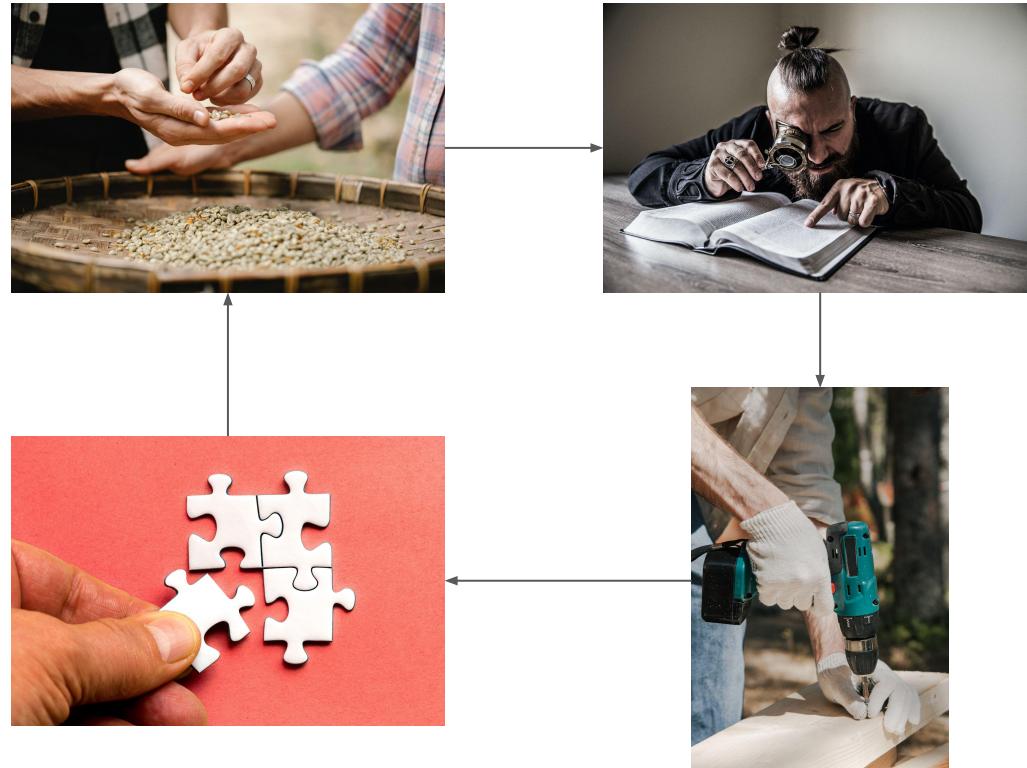
the flake process

regular meeting

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

emergency quarantine

source: [QUARANTINE.md](#)



minimizing the impact

how to find flaky tests?

any merged PR had all tests succeeding in the end,

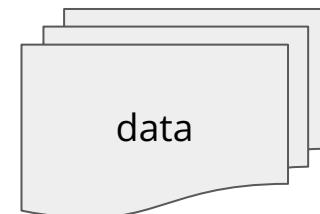
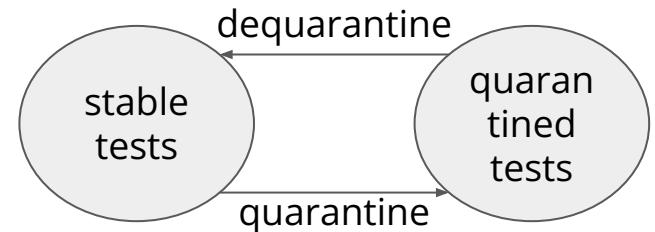
thus any test run with test failures from that PR *might* contain execution of flaky tests

PR History: kubevirt/kubevirt #10634			
12756fa			
pull/kubevirt-spidocs	1719919846822514688	171976231076777312	171724110644602880
pull/kubevirt-build-arm64	17199198464546638720	1719762310169366528	1717241105101050912
pull/kubevirt-build	171991984630615296	171976231009063360	1717241105109487616
pull/kubevirt-check-tests-for-flakes	1717241104786526208		
pull/kubevirt-check-unsigned-tests	1719919849574151424	171976231746863488	1717241109979074560
pull/kubevirt-client-python	1719919846923177984	1719762310019483648	171724110749646592
pull/kubevirt-code-lint	1717241113405820928		
pull/kubevirt-e2e-arm64	1717241112550182912		
pull/kubevirt-e2e-k8s-1.26-sig-compute	1719919853160108032	1719762316933499136	1717241115981123584
pull/kubevirt-e2e-k8s-1.26-sig-network	1719919851482386432	1719762315433218048	171724111477372864
pull/kubevirt-e2e-k8s-1.26-sig-operator	17199198539989468832	17197944228061952	1717096478920101888 1719762317782028288
pull/kubevirt-e2e-k8s-1.28-sig-storage	1719919852317057928	1719762316516889000	1717241115045793792
pull/kubevirt-e2e-k8s-1.27-ipvs-sig-network	1719919849825636352	171976231587724288	171724110838906880
pull/kubevirt-e2e-k8s-1.27-sig-compute	171991985599651840	1719762320920222080	1717241119256479008
pull/kubevirt-e2e-k8s-1.27-sig-network	1719919854688161280	1719762318611694784	171724111753978584
pull/kubevirt-e2e-k8s-1.27-sig-operator	1719919856494579712	1719762321137471488	171724112054456064
pull/kubevirt-e2e-k8s-1.27-sig-storage	171990038025181312	1719815842004209664	1717096231945555584 1717241118493511688
pull/kubevirt-e2e-k8s-1.28-sig-compute-migrations	1720001509401300992	171997601457190028	17199198508081274368 1719762314430777932
pull/kubevirt-e2e-k8s-1.28-sig-compute	171998795857610752	171991985904032240	171976232366636800 1717241122742341632
pull/kubevirt-e2e-k8s-1.28-sig-network	1719919857366994944	1719762321959555072	1717241120922013968
pull/kubevirt-e2e-k8s-1.28-sig-operator	1719919860001017856	171976232448720384	1717241123501310656
pull/kubevirt-e2e-k8s-1.28-sig-storage	171991985821050048	1719762322840358912	1717241121941229568
pull/kubevirt-e2e-kind-1.27-sinov	1719919845945905152	1719762309670244552	1717241104096171136
pull/kubevirt-e2e-kind-1.27-gpu	171991984582659944	171977979453419520	1717241104013896448
pull/kubevirt-e2e-windows2016	1719919845677469696	1719772088580686448	171724110409645936
pull/kubevirt-fossa	171991984646353920	1719762310567825408	171724110577381952

minimizing the impact

what do we need?

- easily move a test between the **set of stable** tests and the **set of quarantined** tests
- 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



tools

quarantining

tools

quarantine mechanics:

ci honoring **QUARANTINE*** label

- pre-merge tests **skip** quarantined tests
- periodics **execute** quarantined tests to check their stability

* we use the Ginkgo label - text label is required for backwards compatibility

```
# If KUBEVIRT_QUARANTINE is not set, do not run quarantined tests. When it is
# set the whole suite (quarantined and stable) will be run.
if [ -z "$KUBEVIRT_QUARANTINE" ]; then
    if [ -n "$KUBEVIRT_E2E_SKIP" ]; then
        export KUBEVIRT_E2E_SKIP="${KUBEVIRT_E2E_SKIP}|$KUBEVIRT_QUARANTINE"
    else
        export KUBEVIRT_E2E_SKIP="$KUBEVIRT_QUARANTINE"
    fi
fi
```

```
176
177 | It("[QUARANTINE]should successfully upgrade virt-handler", func() {
178     var expectedEventsLock sync.Mutex
179     expectedEvents := []string{
180         "maxUnavailable=1",
181         "maxUnavailable=10%",
182         "virt-handler=ready",
183         "maxUnavailable=1",
184     }
185     ds, err := virtCli.AppsV1().DaemonSets(flags.KubeVirtInstall
186     if err != nil {  
...  
}
```

sources:

- <https://github.com/kubevirt/kubevirt/blob/38c01c34acecfafc89078b1bbaba8d9cf3cf0d4d/automation/test.sh#L452>
- <https://github.com/kubevirt/kubevirt/blob/38c01c34acecfafc89078b1bbaba8d9cf3cf0d4d/hack/functests.sh#L69>
- https://github.com/kubevirt/kubevirt/blob/38c01c34acecfafc89078b1bbaba8d9cf3cf0d4d/tests/canary_upgrade_test.go#L177



tools

[quarantine overview](#)

[\(source\)](#)

where?

since when?

Overview of Quarantine tests

Total: 1 tests

<https://github.com/kubevirt/kubevirt/tree/c35be138d85864b17946ae1fe07f99a75445d501/tests/operator/operator.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

metrics

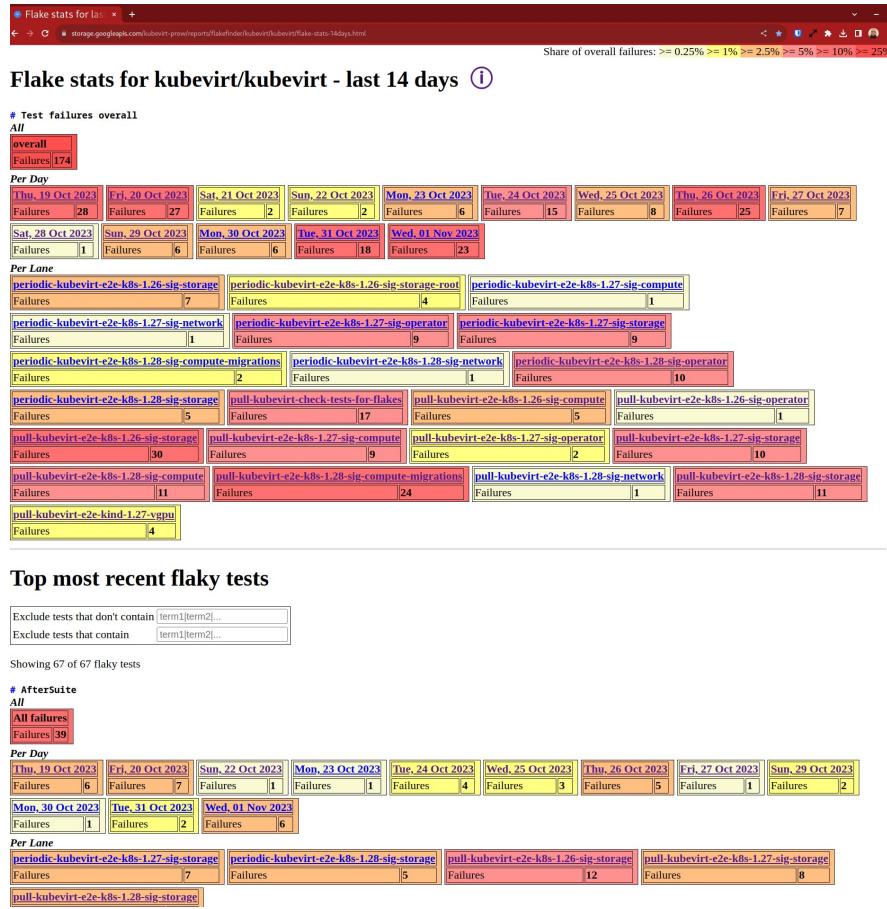


tools

flake stats report

why: detect failure hot spots in one view

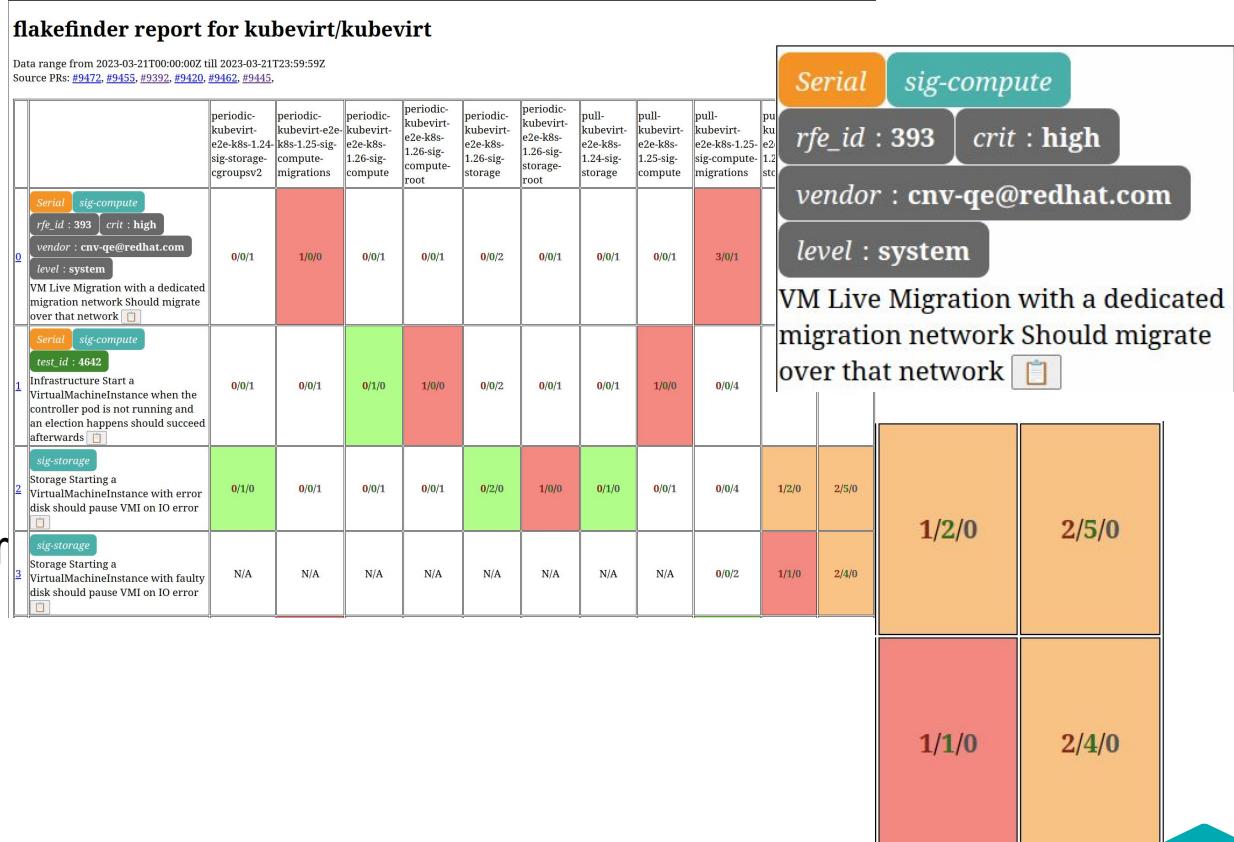
([source](#))



tools

flakefinder report

why: see detailed view for
a certain day

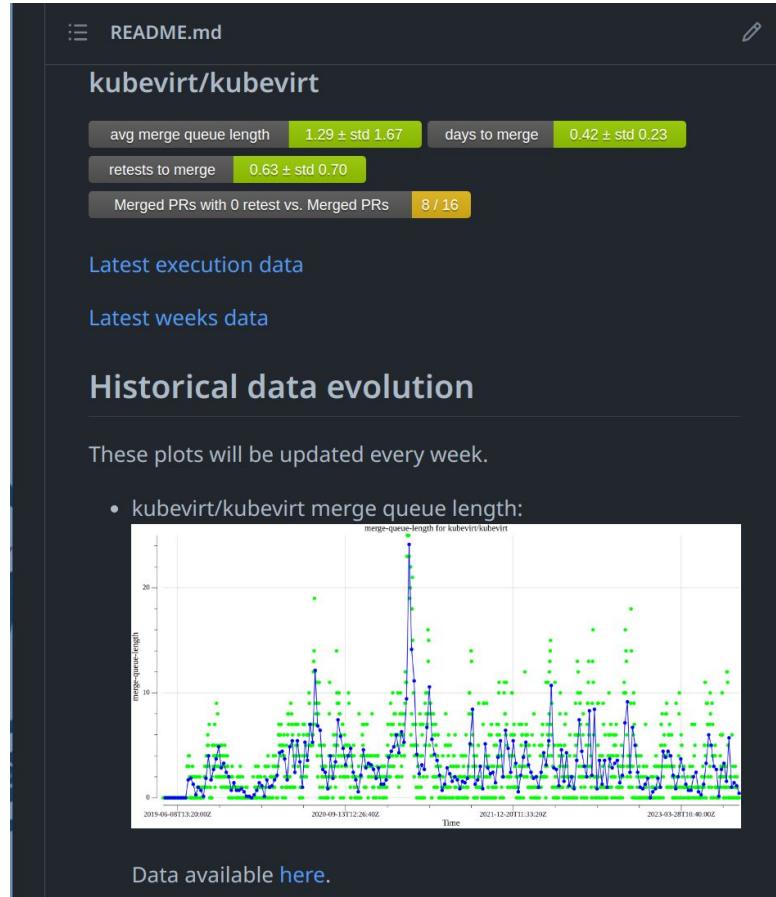


tools

[ci-health](#)

why: show overall CI stability metrics by tracking

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



Data available [here](#).

tools

analysis

tools

ci-search

why: estimate impact as basis for quarantine decision

see openshift ci-search

test_id:1464

2d 1 line bug+ Search

Job: Focus job or bug names by regex ... Skip job or bug names by regex ... 5 20971520 job Wrap lines

Found in 0.14% of runs (1.25% of failures) across 4272 total runs and 272 jobs (11.24% failed) in 39ms - [clear search](#) | [chart view](#) - source code located [on github](#)

pull-kubevirt-e2e-k8s-1.26-sig-compute (all) - 90 runs, 9% failed, 25% of failures match = 2% impact

#1638343071466786816 junit 11 hours ago

```
# [rfe_id:588][crit:medium][vendor:cnv-qe@redhat.com][level:component][sig-compute]ContainerDisk [rfe_id:273][crit:med tests/container_disk_test.go:129]
```

#1638123835775520768 junit 24 hours ago

```
# [rfe_id:588][crit:medium][vendor:cnv-qe@redhat.com][level:component][sig-compute]ContainerDisk [rfe_id:273][crit:med tests/container_disk_test.go:129]
```

pull-kubevirt-e2e-k8s-1.25-sig-compute (all) - 95 runs, 14% failed, 15% of failures match = 2% impact

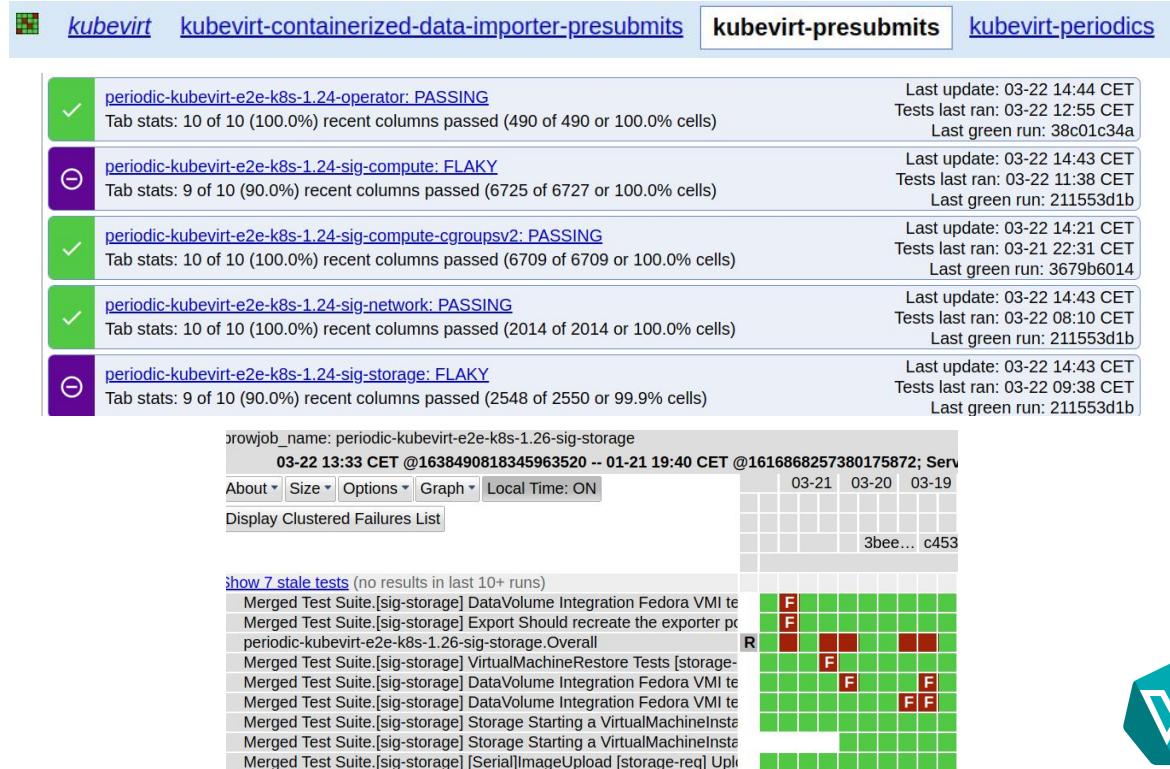
#1638343068434305024 junit 12 hours ago

```
# [rfe_id:588][crit:medium][vendor:cnv-qe@redhat.com][level:component][sig-compute]ContainerDisk [rfe_id:273][crit:med tests/container_disk_test.go:129]
```

tools

testgrid

why: second way to determine instabilities, drill down on all jobs for kubevirt/kubevirt



tools

pre merge detection

tools

check-tests-for-flakes test lane

why: catch flakes before entering main

([source](#))

```
ginko_params="$ginko_params -no-color -succinct --label-filter=!QUARANTINE -randomize-all"
for test_file in $(echo "${NEW_TESTS}" | tr '\n' '\n'); do
    ginko_params+=" -focus-file=${test_file}"
done
|
echo "Test lane: ${TEST_LANE}, preparing cluster up"

if [[ ! "$ginko_params" =~ -dry-run ]]; then
    make cluster-up
    make cluster-sync
else
    # Ginkgo only performs -dryRun in serial mode.
    export KUBEVIRT_E2E_PARALLEL="false"
    NUM_TESTS=1
fi

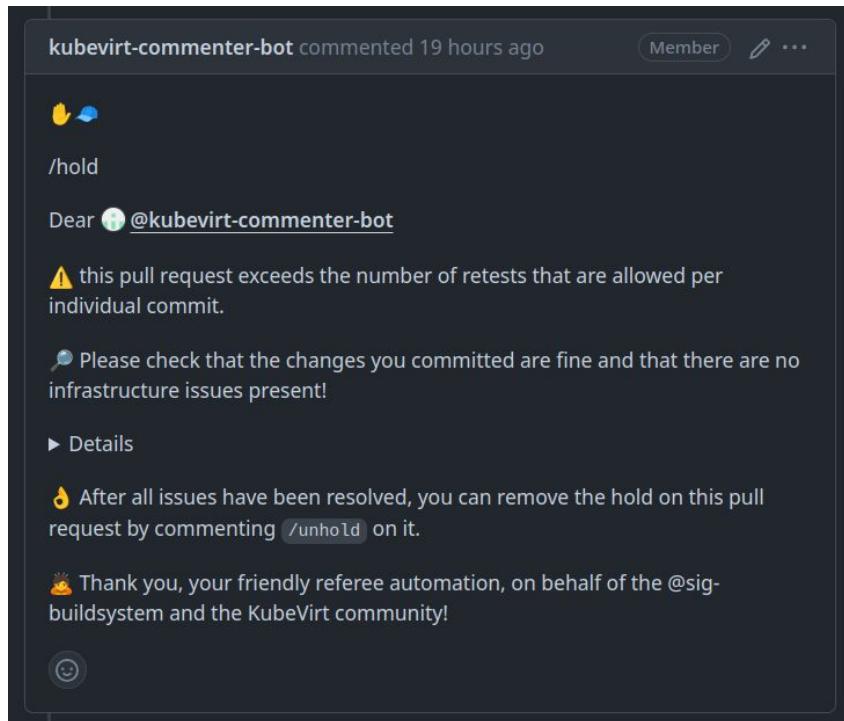
for i in $(seq 1 "$NUM_TESTS"); do
    echo "Test lane: ${TEST_LANE}, run: $i"
    if ! FUNC_TEST_ARGS="$ginko_params" make functest; then
        echo "Test lane: ${TEST_LANE}, run: $i, tests failed!"
        exit 1
    fi
done
```

tools

referee bot

why: stop excessive retesting on PRs without changes

([source](#))



tools

retest metrics dashboard

why:

- show overall CI health via number of retests on PRs
- show PRs exceeding retest count where authors might need support



in a nutshell

In regular intervals:

- follow up on previous action items
- look at data and derive action items
- hand action items over to dev teams
- revisit and dequarantine quarantined tests

main sources of flakiness

- test order dependencies
- concurrency
- data races
- differing execution platforms

key takeaways

- identify outside dependencies you have
- stabilize the testing environment
 - make it resilient against outside dependency failures
 - cache what you can
- use versioning for testing environments

the future - more data, more tooling

gaps we want to close:

- collect more data - run the majority of tests frequently
- steadily improve in detecting new flakes
- use other methods to detect flaky tests, i.e. static code analysis
- long term - automatic quarantine PRs when new flakes have entered the codebase



Q&A

Any questions?

Any suggestions for improvement?

Who else is trying to tackle this problem?

What have you done to solve this?

Thank you for attending!

Further questions?

Feel free to send questions and comments to:

mailto: dhiller@redhat.com

k8s slack: kubernetes.slack.com/
[@dhiller](https://kubernetes.slack.com/messages/@dhiller)

mastodon: [@dhiller@fosstodon.org](https://fosstodon.org/@dhiller)

web: www.dhiller.de

kubevirt.io

KubeVirt welcomes all kinds of contributions!

- **Weekly community meeting every Wed 3PM CET**
- Links:
 - [KubeVirt website](#)
 - [KubeVirt user guide](#)
 - [KubeVirt Contribution Guide](#)
 - [GitHub](#)
 - Kubernetes Slack channels
 - [#virtualization](#)
 - [#kubevirt-dev](#)

