Cgroup v2: Before You Jump In

Tony Gosselin | Mike Tougeron

Adobe Systems

Who we are

Tony Gosselin, Senior Cloud Engineer – Ethos @ Adobe

- Twitter: @sfotony
- Github: sfotony
- gosselin@adobe.com

Mike Tougeron, Lead Cloud Engineer – Ethos @ Adobe

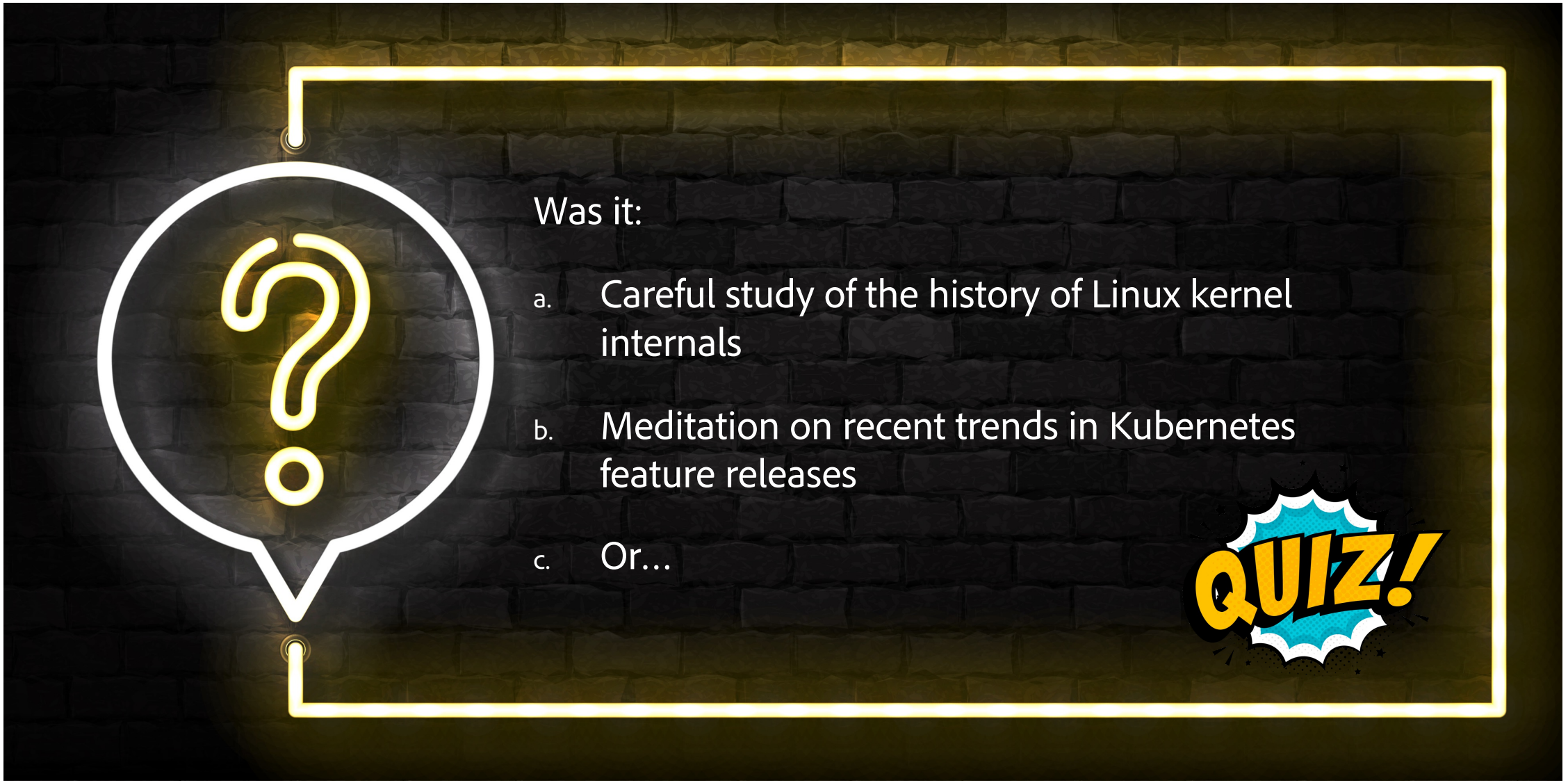
- Twitter: @mtougeron
- Github: mtougeron
- https://grepmymind.com
- mike@tougeron.com or tougeron@adobe.com

Kubernetes @ Adobe: Ethos

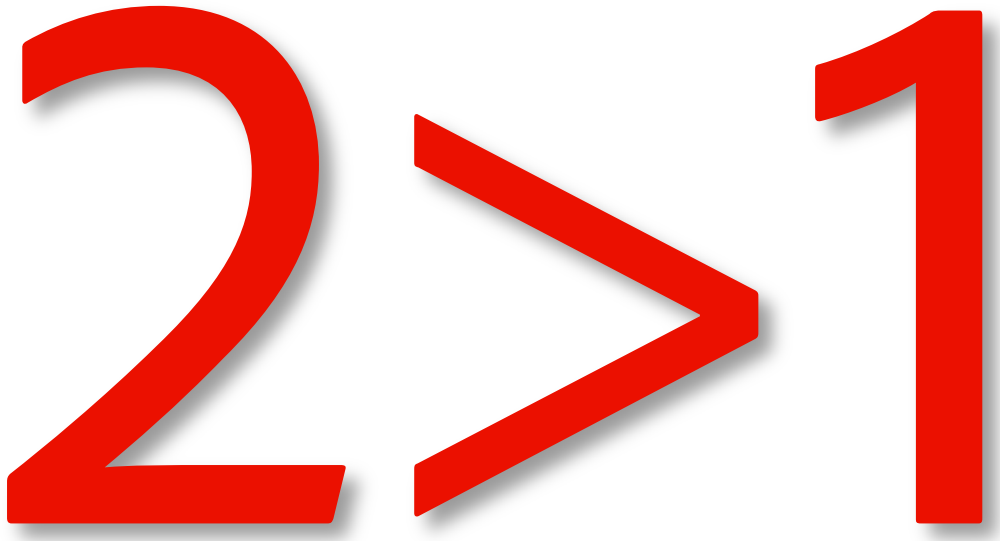
- How Adobe creates, deploys, tests, and runs containerized workloads
- CaaS, PaaS, CI/CD, cluster onboarding, ingress, security controls
- 230 clusters on AWS (EC2 and EKS), Azure, and VMware
- Running > 2,300,000 pods
- Over 18k compute nodes with 1.8 PB memory & 500k cpu
- We make the container mistakes once for everyone, so other teams can focus on writing and deploying great code
- **We have nothing to do with how often Acrobat updates**

Adobe Ethos

How this talk came about…



The reality is that we knew that…



…and we don't often give a lot of thought to the layers under Kubernetes

How we came across cgroup v2 in Kubernetes

- Upgrade hosts to Kinvolk Flatcar 2969.0.0, defaulting to cgroup v2
  - Kubelet fails to start
- Configure kubelet (Kubernetes 1.20) and docker to use cgroup v2
  - Cluster boots! But….
  - Metrics-server is missing key metrics
- Run a newer cAdvisor (0.43) as a DaemonSet
  - Other metrics have now changed, our telemetry team gathers their pitchforks and torches
- Finally, we talk with Sig Node…

What are cgroups?

- Linux kernel functionality – short for “control groups” – January 2008
- Accessible via virtual filesystem (like everything in Linux)
- Affects resource limits, prioritization, accounting and control
- Without cgroups, there are no containers or Kubernetes
- How does Kubernetes make use of cgroups?
  - Container resource enforcement
  - Container resource metrics

cgroup v2 - what changed

- Introduced in March 2015, declared “non-experimental” in kernel 4.5
- Single, strict hierarchy

Single, strict hierarchy

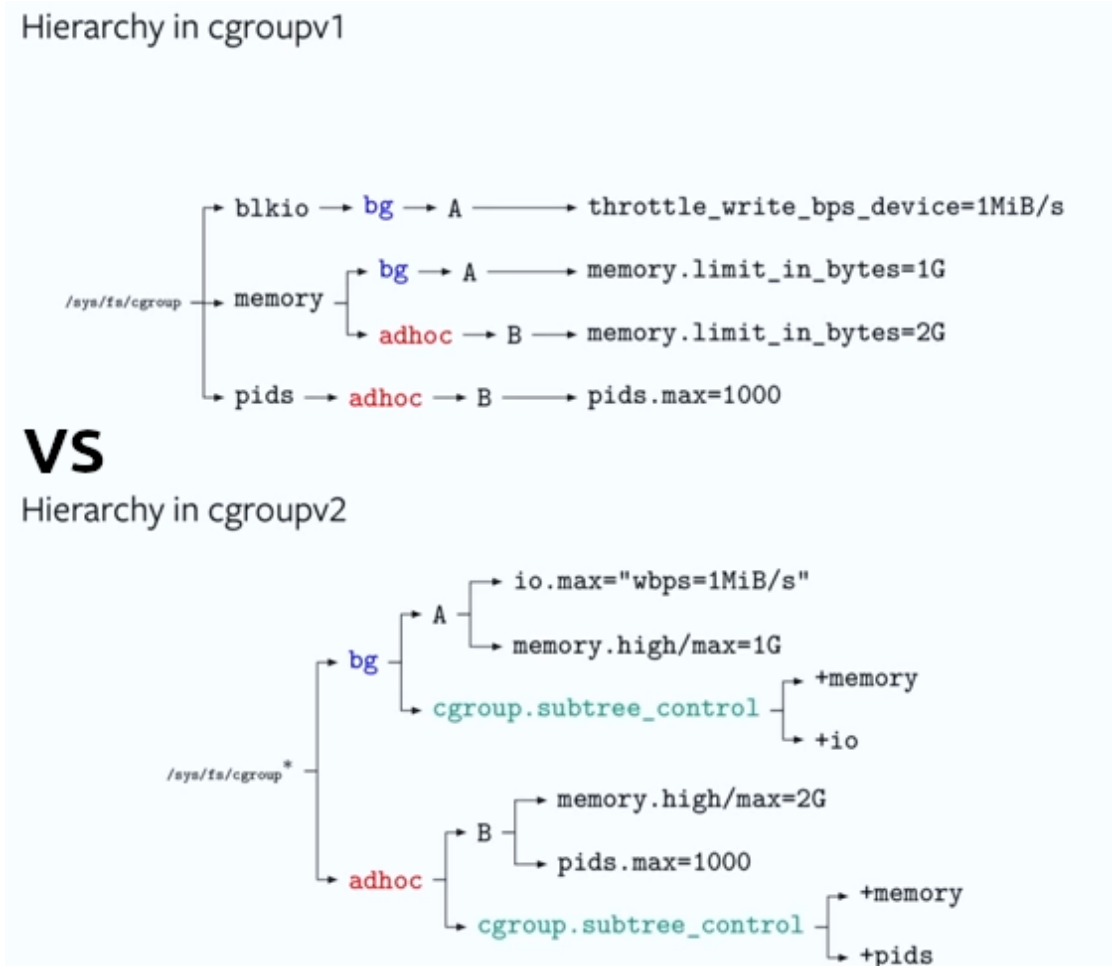


Diagram Source: Pavel Trukhanov, https://medium.com/some-tldrs/tldr-understanding-the-new-control-groups-api-by-rami-rosen-980df476f633

cgroup v2 - what changed

- Introduced in March 2015, declared “non-experimental” in kernel 4.5
- Single, strict hierarchy
  - Safer sub-tree delegation to containers
  - Makes coordinated resource management possible
  - Threads of the same process can now only be assigned to one cgroup
- eBPF-oriented device control (not static files)
- Pressure Stall Information (PSI)
- Resource limitation support for rootless containers

Kubernetes support timeline for cgroup v2

- Late 2019 – Most container runtimes add support
- 1.19 - Initial Kubernetes support for cgroup v2 nodes
- 1.22 – Alpha feature – memory quality of service
- Nov 2021 – cAdvisor 0.43* with full cgroup v2 support released
  - *cAdvisor 0.44 – use this one, at least
- June 2022 – 1.25 – cgroup v2 support graduates to stable

But what does all this mean?

- Important hierarchy changes allow for:
  - PSI – Pressure Stall Information per container
  - Memory QoS <-- Huge for those of us who significantly over-subscribe compute nodes!
  - Container aware OOM killer
  - Rootless containers
- All new resource management features for kubelet going forward will be cgroup v2 only
- cgroup v1 – is stable and will be supported for a while, but removal from systemd is set for EOY 2023

"With great power comes many updates"

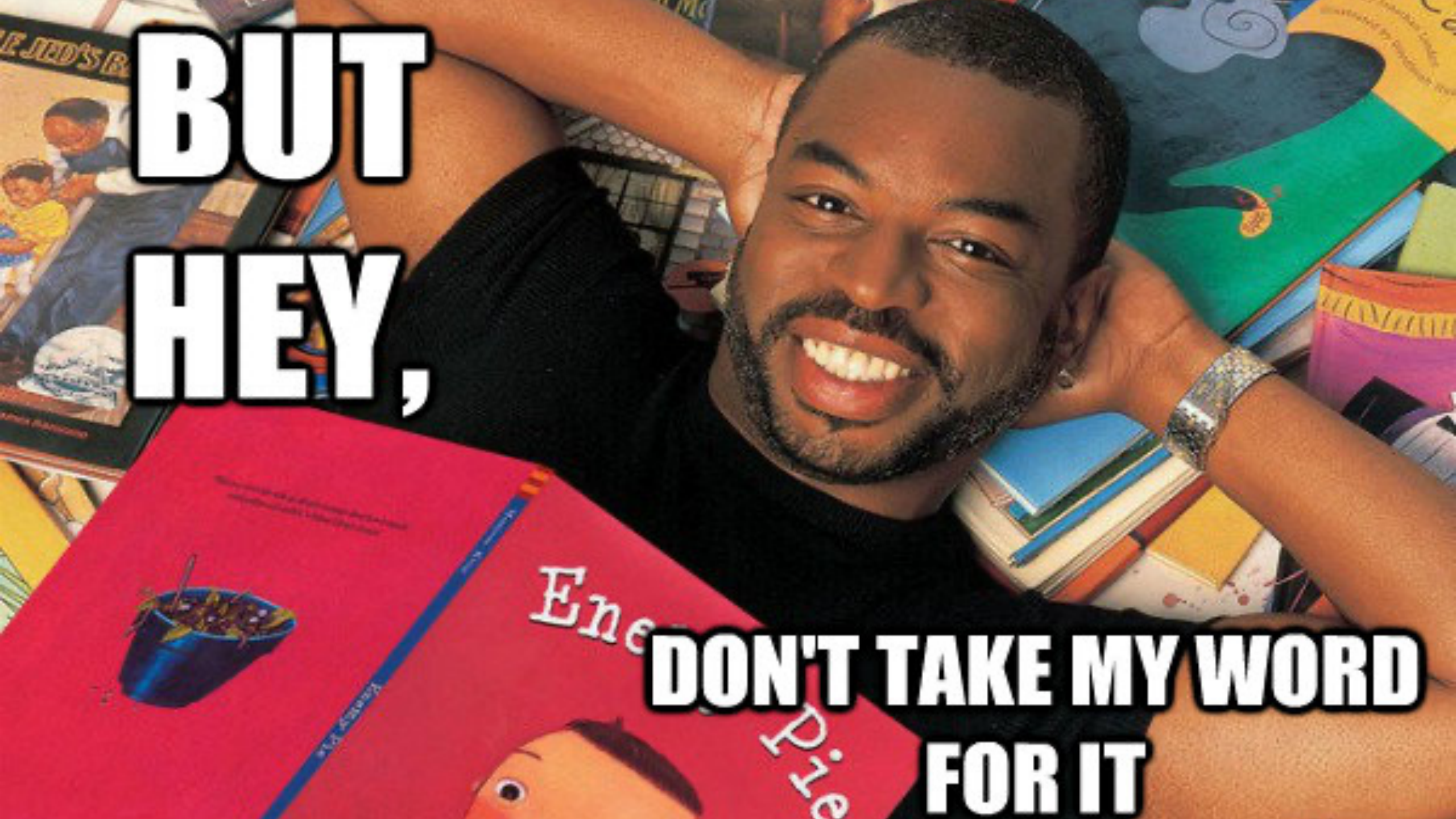
Thankfully not too many this time around...

What cluster operators need to do to prepare

- Make sure you're running an OS version that supports cgroup v2 (duh!)
  - Linux kernel 5.8+
  - stat -fc %T /sys/fs/cgroup/
    # returns either cgroup2fs (v2) or tmpfs (v1)
- Kubernetes >= 1.25 is preferred
- Make sure that kubelet & container runtime are both configured to use systemd
  - kubelet will auto-detect which cgroup version is running on the system <-- means you can test per node if you want
- cAdvisor >= 0.44 (if running as a DaemonSet)

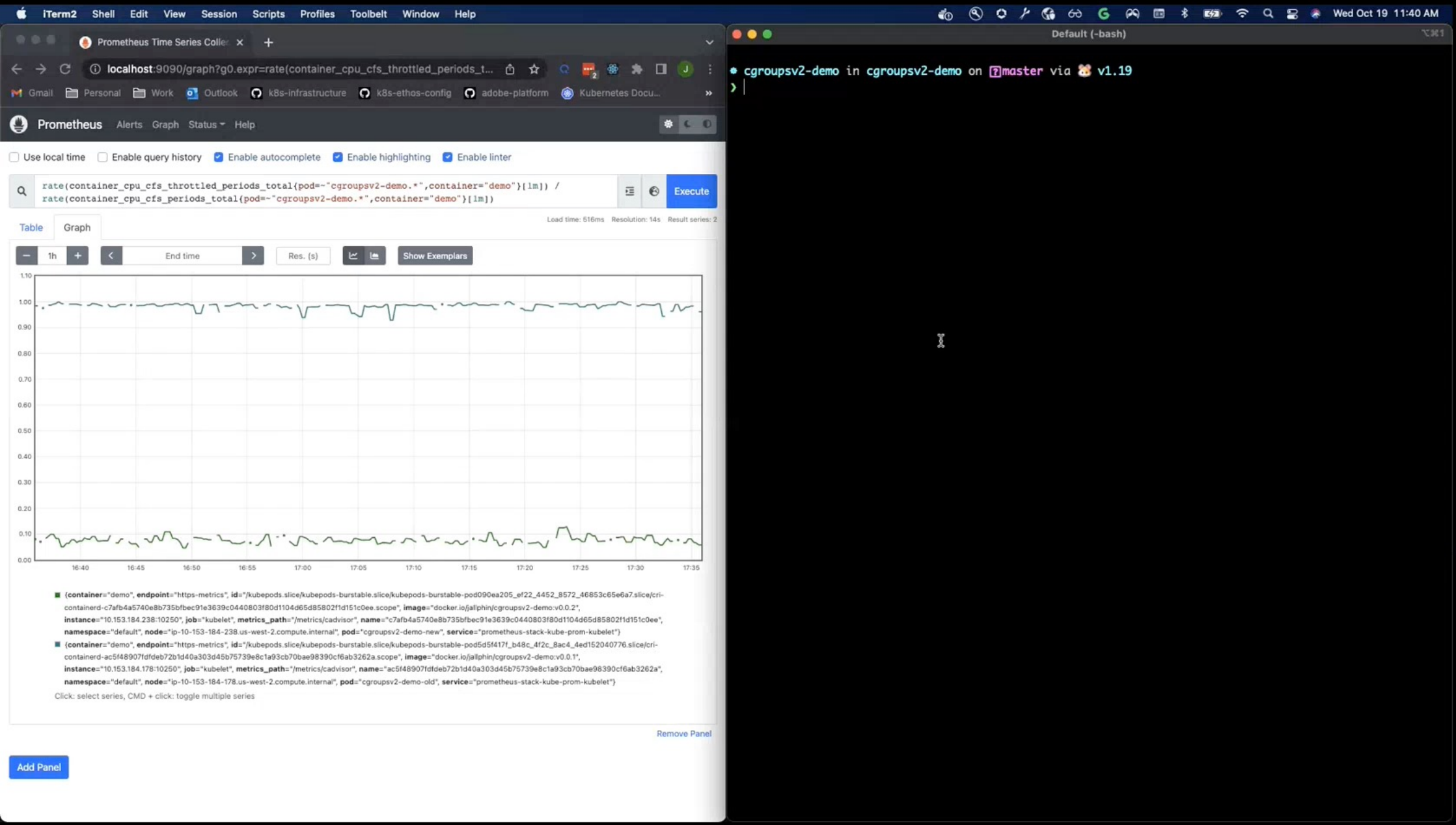
What developers need to do to prepare

- Make sure your language of choice supports cgroup v2 for multi-process apps
- JDK >= 11.0.16 or JDK >= 15
- Go uber-go/automaxproc >= v1.5.0
- Python 3.12 -- maybe? Git issue has been open forever about having *any* native cgroup support
- Testing, testing, and maybe some more testing


BUT
HEY,
DON'T TAKE MY WORD
FOR IT

Quick demo of doing it wrong vs doing it right

Jantzen Allphin, Cloud Engineer, Ethos @ Adobe


iTerm2 Shell Edit View Session Scripts Profiles Toolbelt Window Help
Wed Oct 19 11:40 AM
Prometheus Time Series Colle
localhost:9090/graph?g0.expr=rate(container_cpu_cfs_throttled_periods_t...
Gmail Personal Work Outlook k8s-infrastructure k8s-ethos-config adobe-platform Kubernetes Docu...
Prometheus Alerts Graph Status Help
Use local time Enable query history Enable autocomplete Enable highlighting Enable linter
rate(container_cpu_cfs_throttled_periods_total{pod=~"cgroupsv2-demo.*",container="demo"}[1m]) /
rate(container_cpu_cfs_periods_total{pod=~"cgroupsv2-demo.*",container="demo"}[1m])
Execute
Load time: 516ms Resolution: 14s Result series: 2
Table Graph
1h End time Res. (s) Show Exemplars
1.10 1.00 0.90 0.80 0.70 0.60 0.50 0.40 0.30 0.20 0.10 0.00
16:40 16:45 16:50 16:55 17:00 17:05 17:10 17:15 17:20 17:25 17:30 17:35
{container="demo", endpoint="https-metrics", id="/kubepods.slice/kubepods-burstable.slice/kubepods-burstable-pod090ea205_ef22_4452_8572_46853c65e6a7.slice/cri-containerd-c7afb4a5740e8b735bfbec91e3639c0440803f80d1104d65d85802f1d151c0ee.scope", image="docker.io/jallphin/cgroupsv2-demo:v0.0.2", instance="10.153.184.238:10250", job="kubelet", metrics_path="/metrics/cadvisor", name="c7afb4a5740e8b735bfbec91e3639c0440803f80d1104d65d85802f1d151c0ee", namespace="default", node="ip-10-153-184-238.us-west-2.compute.internal", pod="cgroupsv2-demo-new", service="prometheus-stack-kube-prom-kubelet"}
{container="demo", endpoint="https-metrics", id="/kubepods.slice/kubepods-burstable.slice/kubepods-burstable-pod5d5f417f_b48c_4f2c_8ac4_4ed152040776.slice/cri-containerd-ac5f48907fdfdeb72b1d40a303d45b75739e8c1a93cb70bae98390cf6ab3262a.scope", image="docker.io/jallphin/cgroupsv2-demo:v0.0.1", instance="10.153.184.178:10250", job="kubelet", metrics_path="/metrics/cadvisor", name="ac5f48907fdfdeb72b1d40a303d45b75739e8c1a93cb70bae98390cf6ab3262a", namespace="default", node="ip-10-153-184-178.us-west-2.compute.internal", pod="cgroupsv2-demo-old", service="prometheus-stack-kube-prom-kubelet"}
Click: select series, CMD + click: toggle multiple series
Remove Panel
Add Panel
Default (-bash)
cgroupsv2-demo in cgroupsv2-demo on master via v1.19

Talks and resources to check out

- See more talks this week!

  - *Cgroupv2 Is Coming Soon To a Cluster Near You*
    - David Porter, Google & Mrunal Patel, RedHat
    - https://sched.co/182JZ on Friday @ 2:00pm

  - *Kubernetes SIG Node Intro And Deep Dive*
    - Sergey Kanzhelev & Dawn Chen, Google; Derek Carr, Red Hat
    - https://sched.co/182Pi on Friday @ 4:00pm

- Release Blog
  - *Kubernetes 1.25: cgroup v2 graduates to GA*
  - https://kubernetes.io/blog/2022/08/31/cgroupv2-ga-1-25/

About the artist

Dan Zucco

London-based 3D art and motion director Dan Zucco creates repeating 2D patterns and brings them to life as 3D animated loops. Inspired by architecture, music, modern art, and generative design, he often starts in Adobe Illustrator and builds his animations using Adobe After Effects and Cinema 4D. Zucco's objective for this piece was to create a geometric design that felt like it could have an infinite number of arrangements.

Made with

Adobe Illustrator Adobe After Effects