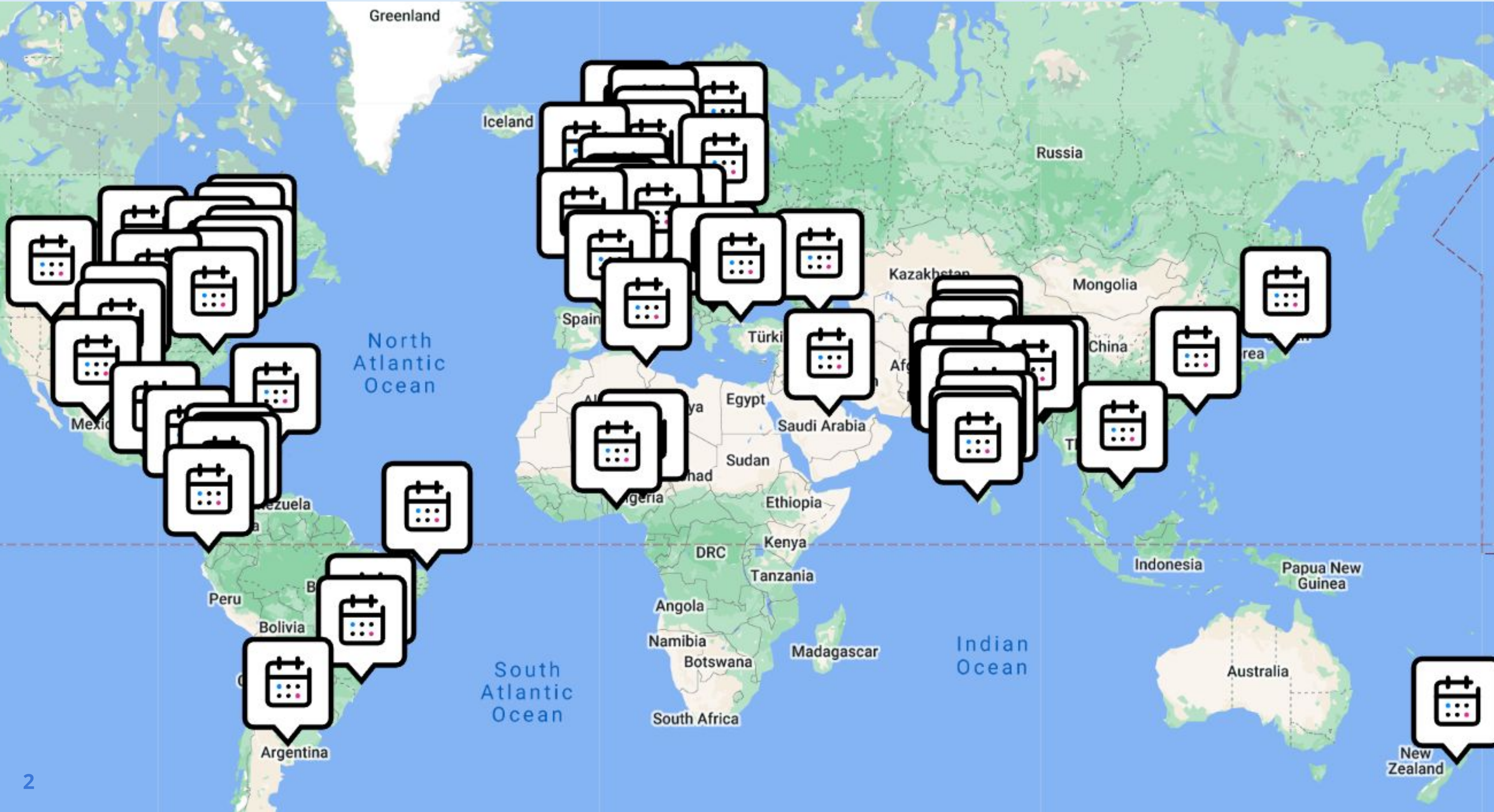
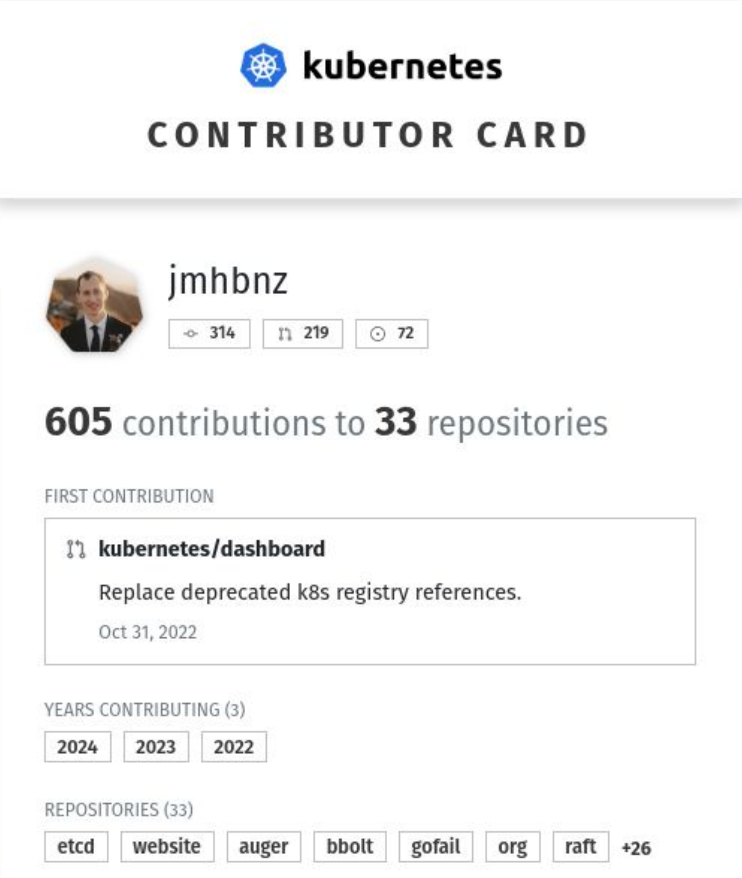
Kubernetes for the next decade

James Blair


Greenland
Iceland
Russia
Kazakhstan
Mongolia
China
Korea
Spain
Türkiye
Egypt
Saudi Arabia
Sudan
Chad
Nigeria
Ethiopia
Kenya
DRC
Tanzania
Angola
Namibia
Botswana
Madagascar
South Africa
North Atlantic Ocean
South Atlantic Ocean
Indian Ocean
Mexico
Venezuela
Peru
Bolivia
Argentina
Indonesia
Papua New Guinea
Australia
New Zealand



kubernetes

CONTRIBUTOR CARD

jmhbnz

314 | 219 | 72

605 contributions to **33** repositories

FIRST CONTRIBUTION

kubernetes/dashboard

Replace deprecated k8s registry references.

Oct 31, 2022

YEARS CONTRIBUTING (3)

2024 | 2023 | 2022

REPOSITORIES (33)

etcd | website | auger | bbolt | gofail | org | raft | +26

contribcard.clotributor.dev

Who am I?

- Open source enthusiast & contributor
- Maintainer & sig co-chair @ etcd-io
- Specialist Architect @ Red Hat

Contact

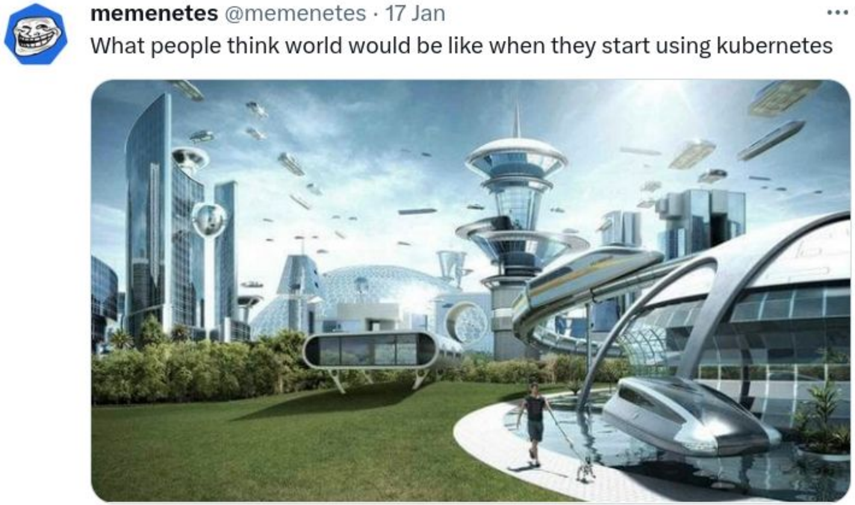
james.blair@redhat.com

Github

github.com/jmhbnz

Are we there yet?

- Kubernetes is 10 years old now
- Are we done? Can we pack up and go home?



memenetes @memenetes · 17 Jan

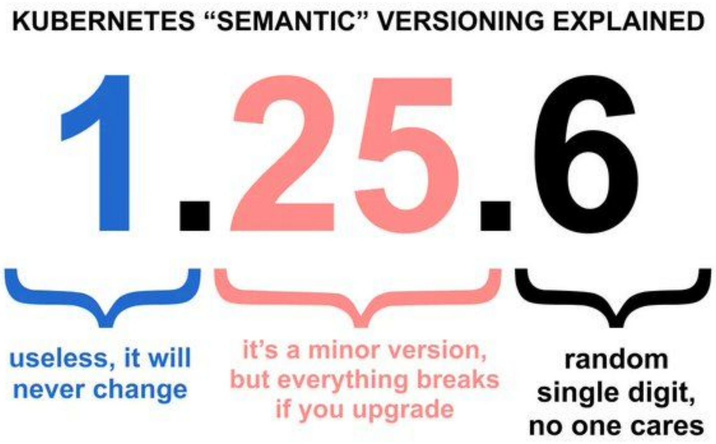
What people think world would be like when they start using kubernetes

Predictions

- Some of us will spend the next 2-5 years migrating vm fleets into k8s
- Running clusters with bare metal compute nodes will be standard practice
- A majority of clusters will have compute accelerators available
- The current monopoly of gpu based accelerators for k8s will be disrupted
- Orgs running k8s will all write their own custom org specific k8s operators and crd’s

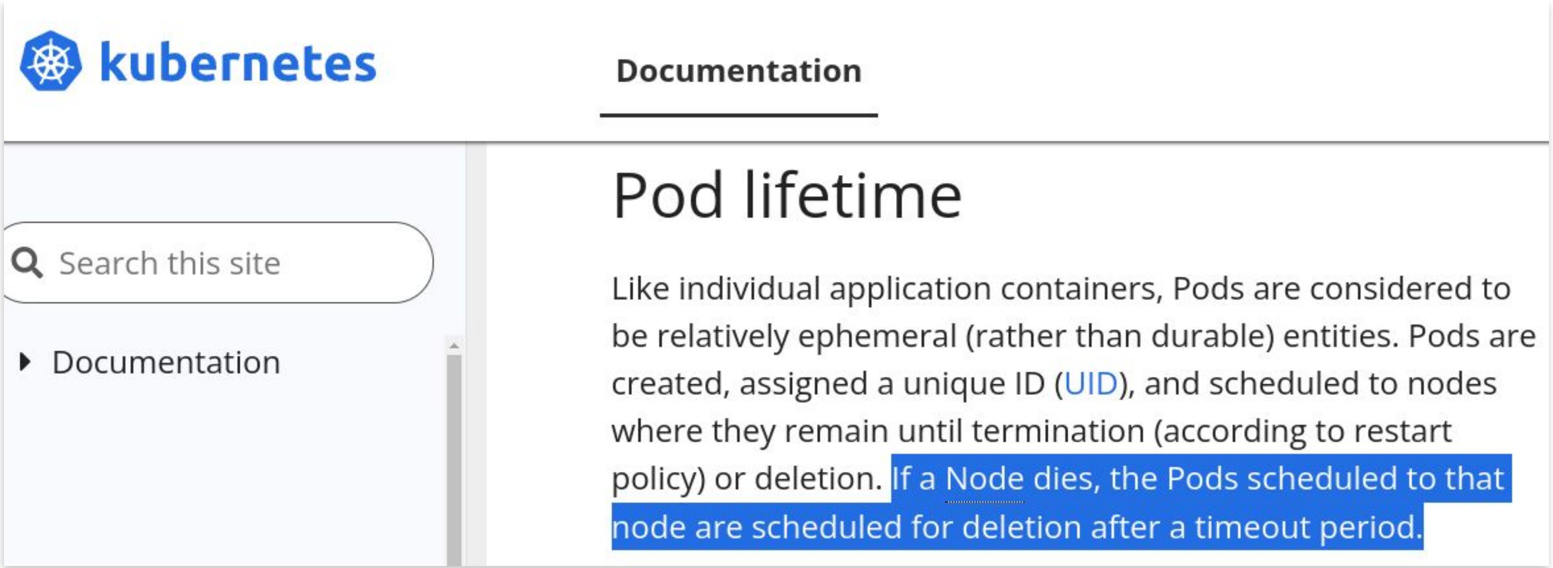
Let’s think about 2.x.x

- We are all so busy on the k8s dancefloor, have we checked the view from the balcony?
- Is what we have now in k8s what we actually need for the next decade?


KUBERNETES “SEMANTIC” VERSIONING EXPLAINED
1.25.6
useless, it will never change
it’s a minor version, but everything breaks if you upgrade
random single digit, no one cares

Exploring one example

- What happens to our pods, when the node they run on dies?

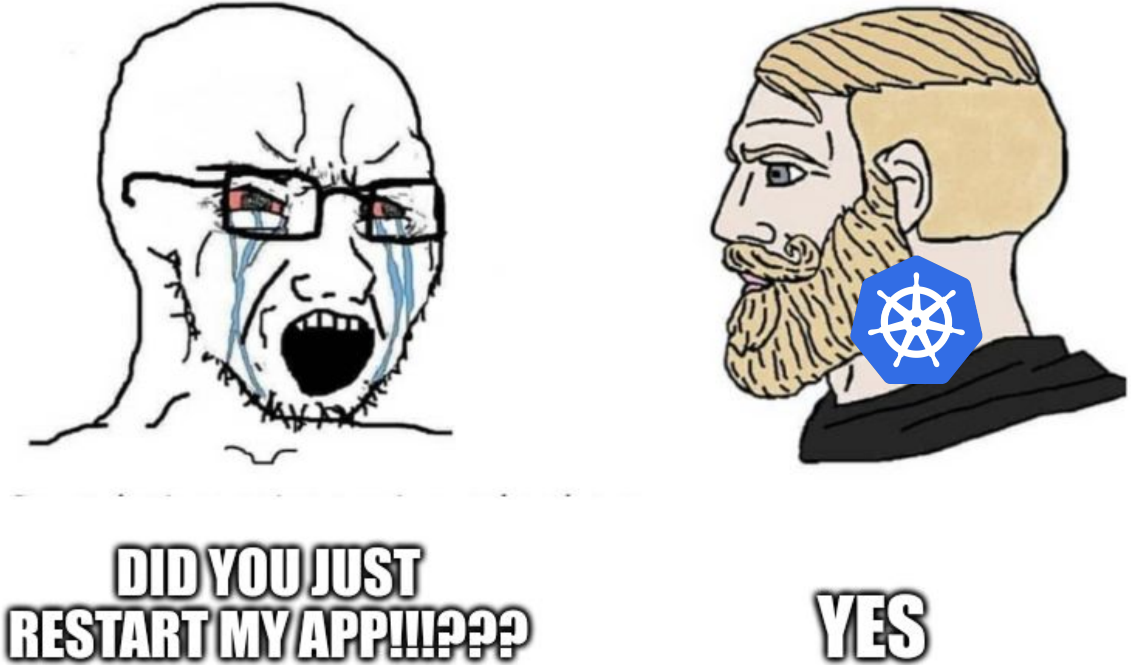


Pod lifetime

Like individual application containers, Pods are considered to be relatively ephemeral (rather than durable) entities. Pods are created, assigned a unique ID (UID), and scheduled to nodes where they remain until termination (according to restart policy) or deletion. If a Node dies, the Pods scheduled to that node are scheduled for deletion after a timeout period.

kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle/#pod-lifetime

What’s the issue?



What’s the issue really?

- Not every application will be refactored to be truly cloud native. However Kubernetes is now our “Operating System” and we want it to run **everything**
- The underlying host failure challenge is not new, it’s just really hard to solve
- Virtualisation hypervisors have a decent solution to this with live migration
- **Kubernetes was originally designed for stateless workloads and the impact of those design choices is now being felt**

What can we do about it?

- Better application level clustering and handling for complex stateful workloads
- Wrap complex stateful applications in KubeVirt VM’s for KVM live migration
- Extend Kubernetes to support live migration for standard containers?

Live migrating processes 🤯

- Containers are just fancy linux processes, so can we freeze and restore a process?
- Yes! Enter **CRIU** aka **C**heckpoint and **R**estore **i**n **U**serspace.

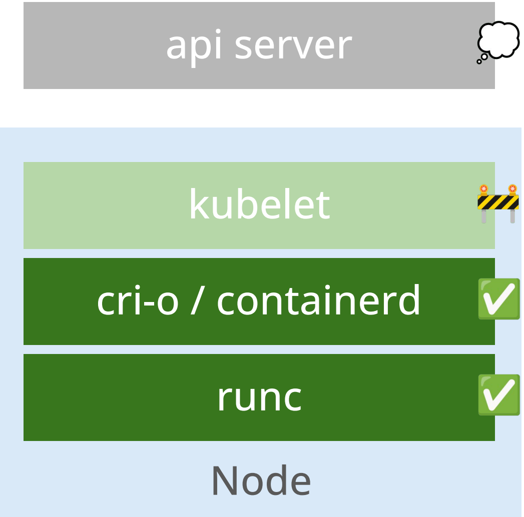
  github.com/checkpoint-restore/criu

  > *“Using this tool, you can freeze a running application (or part of it) and checkpoint it to a hard drive as a collection of files. You can then use the files to restore and run the application from the point it was frozen at.”*

Demo - CRIU

Live migrating containers 🤯

- Not yet. Support in k8s is currently focused on forensic analysis only.



api server 💭

kubelet 🚧

cri-o / containerd ✅

runc ✅

Node

No api server functionality for checkpoint/restore yet, follow KEP #2008

Alpha support for checkpoint, no restore yet

Supports checkpoint & restore

Supports checkpoint & restore

Experimental workarounds

- Can we hack around current limitations and make it work anyway?
- Yes! Enter **CRIK**

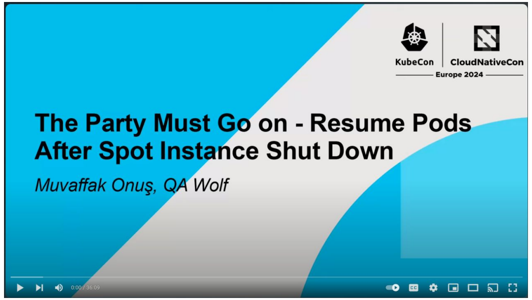
  github.com/qawolf/crik

> *“crik is a project that aims to provide checkpoint and restore functionality for Kubernetes pods mainly targeted for node shutdown and restart scenarios. It is a command wrapper that, under the hood, utilizes criu to checkpoint and restore process trees in a Pod.”*

Demo - CRIK

Last words

- We need to revisit assumptions made about how k8s should behave
- Kubernetes is not done, we still have hard problems to solve
- Working together our awesome cloud native community will solve them and we would love your help and input
- If you’re interested in CRIU and CRIK please review this talk from KubeCon EU 24 which covers the subject in much greater detail than I could in this short talk: youtu.be/c2MbSM9-7Xs.


KubeCon | CloudNativeCon
Europe 2024

The Party Must Go on - Resume Pods After Spot Instance Shut Down

Muvaffak Onuş, QA Wolf

Thank you!