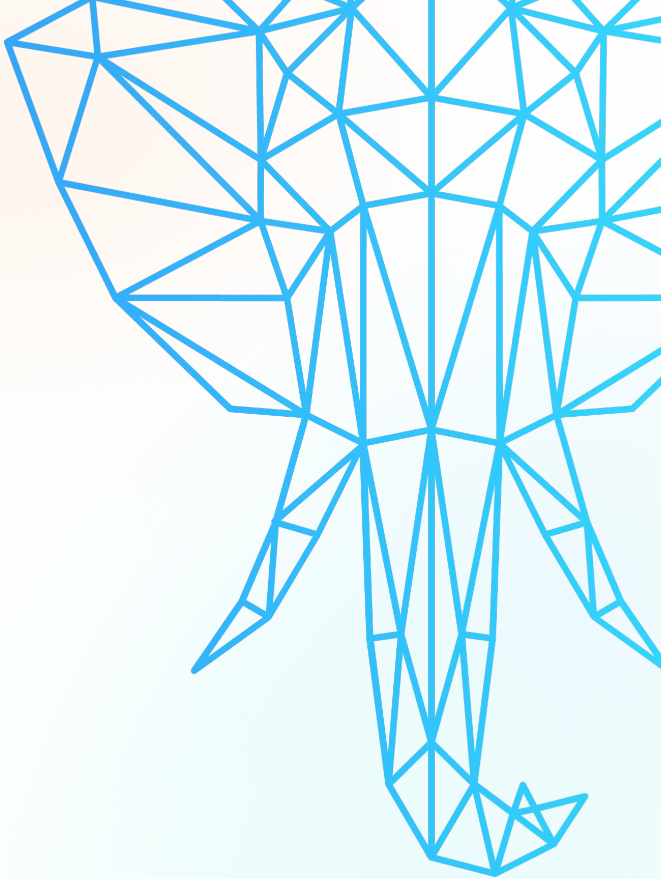
Postgres in the Container Era

Álvaro Hernández
@ahachete



`whoami`

- Founder & CEO, OnGres
- 20+ years Postgres user and DBA
- Mostly doing R&D to create new, innovative software on Postgres
- More than 135 tech talks, most about Postgres
- Founder and President of the NPO *Fundación PostgreSQL*
- AWS Data Hero

Alvaro Hernandez

<aht@ongres.com>

aht.es

This is a journey from

doubting, being scared, and avoiding

to

embracing, loving and needing

containers

But what about Kubernetes???

https://www.rawpixel.com/image/434575/massive-container-ship

This is a journey from

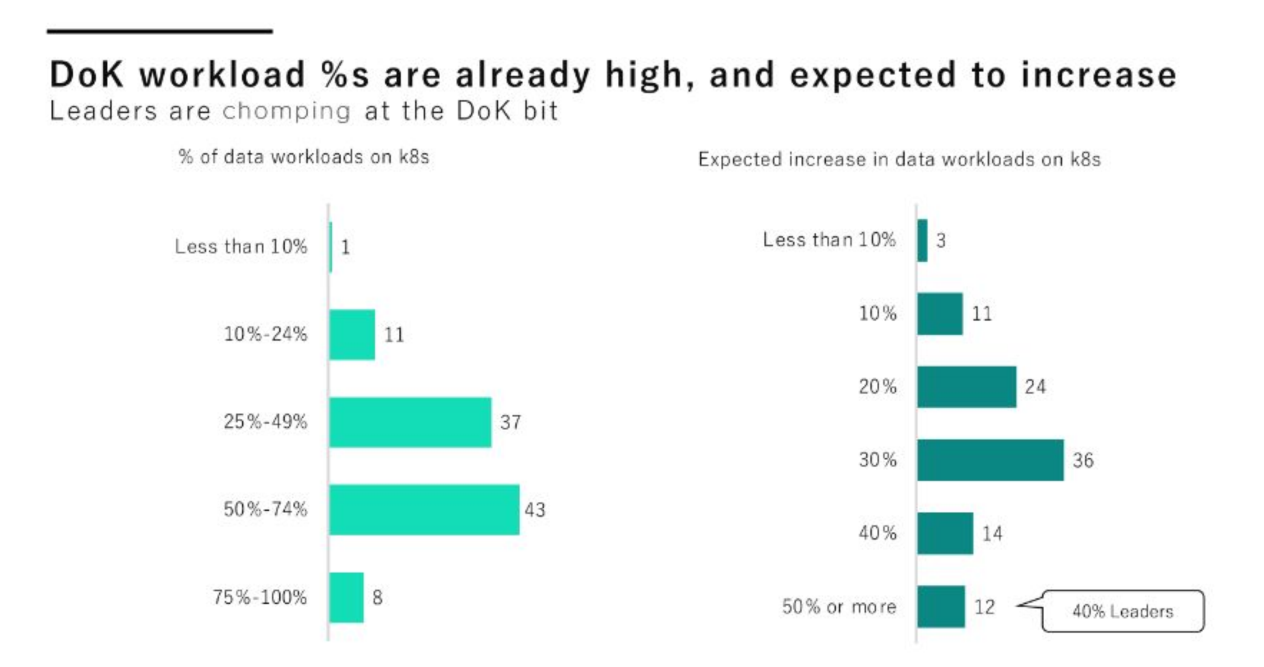
doubting, being scared, and avoiding

to

embracing, loving and needing

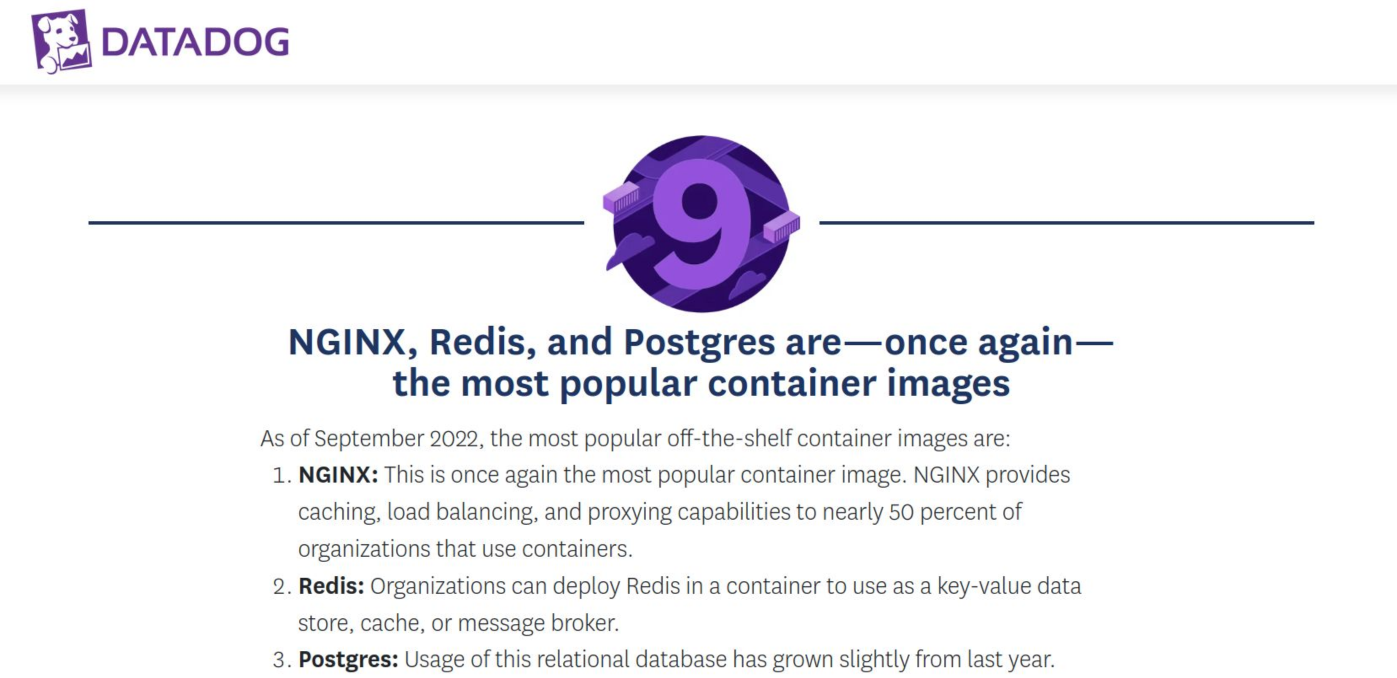
containers and Kubernetes

Containers and Kubernetes are here to stay



https://dok.community/wp-content/uploads/2022/10/DoK_Report_2022.pdf

Containers and Kubernetes are here to stay



DATADOG

9

NGINX, Redis, and Postgres are—once again—the most popular container images

As of September 2022, the most popular off-the-shelf container images are:

1. **NGINX:** This is once again the most popular container image. NGINX provides caching, load balancing, and proxying capabilities to nearly 50 percent of organizations that use containers.
2. **Redis:** Organizations can deploy Redis in a container to use as a key-value data store, cache, or message broker.
3. **Postgres:** Usage of this relational database has grown slightly from last year.

https://www.datadoghq.com/container-report/

So what's a container anyway?

Pack

once

Deploy

anywhere

What is NOT a container

A container is not a “lightweight” VM

What is NOT a container

A container is not a “lightweight” VM

- Well, a container is more lightweight than a VM

What is NOT a container

A container is not a “lightweight” VM

- Well, a container is more lightweight than a VM
- Well, actually a container can be a VM

  (but that’s not what we meant here)

So what's technically a container?

- Namespace isolation

```
$ unshare …
```

So what’s technically a container?

- Namespace isolation

```
$ unshare …
```

- Container image / chroot

So what's technically a container?

- Namespace isolation

```
$ unshare …
```

- Container image / chroot
- cgroups

Yeah, but what about I/O?

Yeah, but what about I/O?

https://giphy.com/gifs/KubbAndCo-simple-just-do-it-quite-really-W5Ub2lhJPWlL4iXnNL/

Just.

Use.

External.

Storage

The problems that containers can solve for Postgres

Extensions, extensions, extensions

https://www.youtube.com/watch?v=EMldOiiG1Ko

Option 1: the fatty container

- Size
- Security
- Restarts (downtime)

Option 2: dynamically inject into container

- Security
- Startup time
- Approach followed in StackGres

See https://aht.es/#talks-postgres_extensions_in_kubernetes

Option 3: dynamically generate container images

Option 3: dynamically generate container images

Formula to compute all possible number of images,

considering *n* extensions to choose from

Option 3: dynamically generate container images

Formula to compute all possible number of images, considering *n* extensions to choose from

$$\sum_{r=1}^{r=n} \frac{n!}{r!\,(n-r)!}$$

Option 3: dynamically generate container images

Formula to compute all possible number of images,

considering *n* extensions to choose from

$$\sum_{r=1}^{r=n} \frac{n!}{r!\,(n-r)!}$$

is bigger than

$$\frac{n!}{(n/2)!(n/2)!}$$

Option 3: dynamically generate container images

$$\frac{n!}{(n/2)!(n/2)!}$$

n=10

→ 252

n=200

→ 9.05485146561032811654E+58

Option 3: dynamically generate container images

DEMO

Option 3: dynamically generate container images

- acl
- hypopg
- mv_stats
- orafce
- pg_jobmon
- pg_track_settings
- pg_uuidv7
- vector

Can containers solve the Postgres collation issues?

- OS upgrades are painful and vendor dependent

Can containers solve the Postgres collation issues?

- OS upgrades are painful and vendor dependent
- Containers isolate (and stabilize!) OS image

Can containers solve the Postgres collation issues?

- OS upgrades are painful and vendor dependent
- Containers isolate (and stabilize!) OS image
- Create containers with stable collation (libc or better ICU) and deploy on any OS

Single image, multiple OS

- Containers are (mostly) Linux-only

Single image, multiple OS

- Containers are (mostly) Linux-only
- But Windows (and also Mac) can run (more or less transparently) Linux containers

Single image, multiple OS

- Containers are (mostly) Linux-only
- But Windows (and also Mac) can run (more or less transparently) Linux containers
- Pack once, deploy on any OS

Single image, multiple OS: embed container runtime

DEMO

Kubernetes…

Kubernetes is containers next level

- It’s not “that complicated”

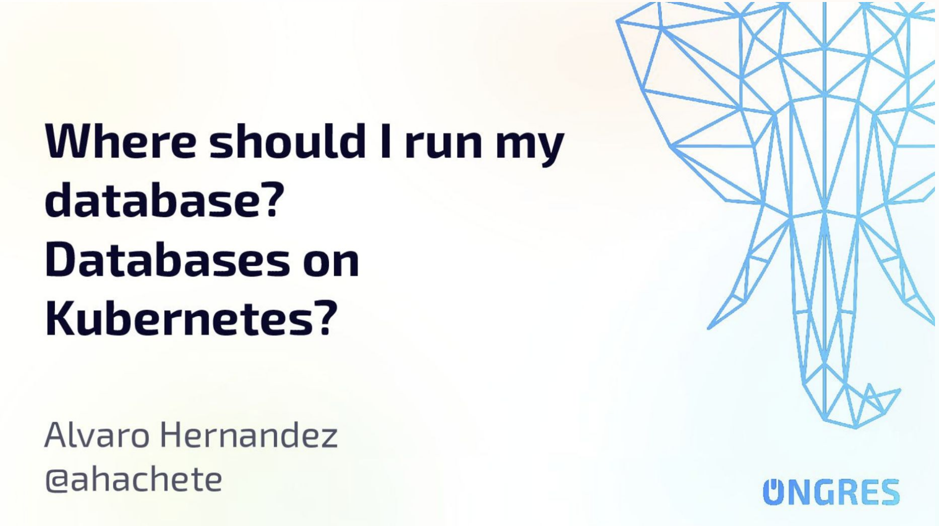
Kubernetes is containers next level

- It’s not “that complicated”
- Allows automation of infrastructure to unknown levels

Kubernetes is containers next level

- It’s not “that complicated”
- Allows automation of infrastructure to unknown levels
- Operators can make extremely hard things extremely easy

Learn more about Postgres on Kubernetes



Where should I run my database?
Databases on Kubernetes?

Alvaro Hernandez
@ahachete

https://speakerdeck.com/ongres/where-should-i-run-my-database

Run Postgres in Containers