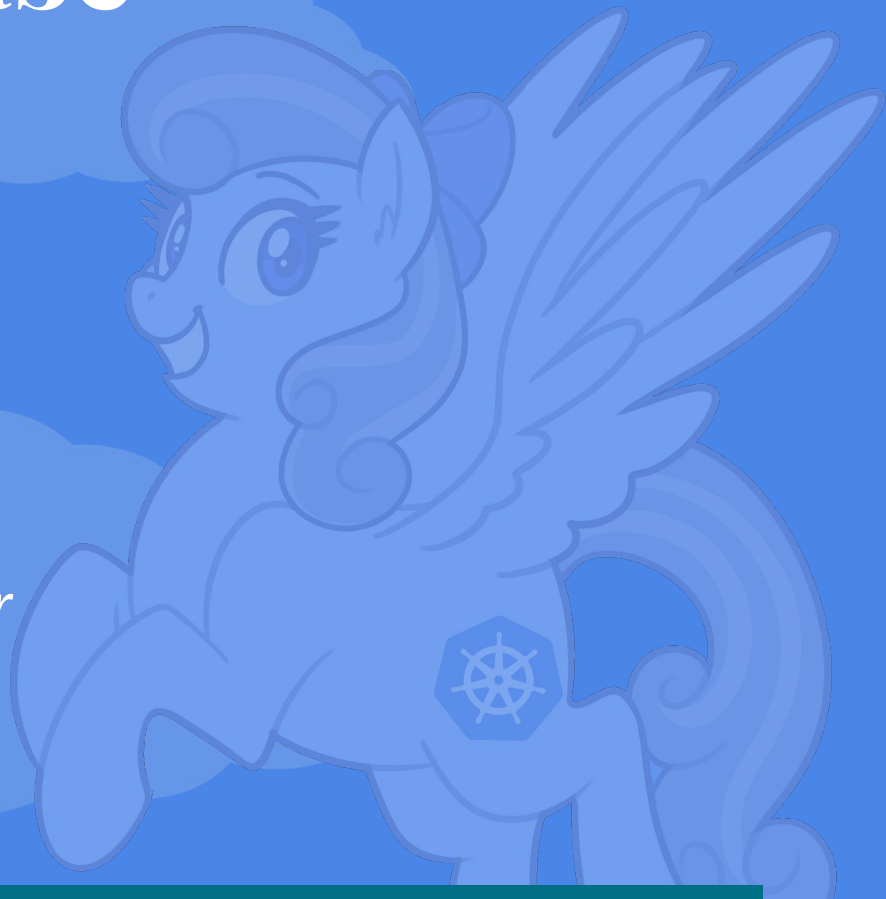
You got Database in my Cloud!

—

Kubernetes Foreign Data Wrapper for Postgres



Who am I?

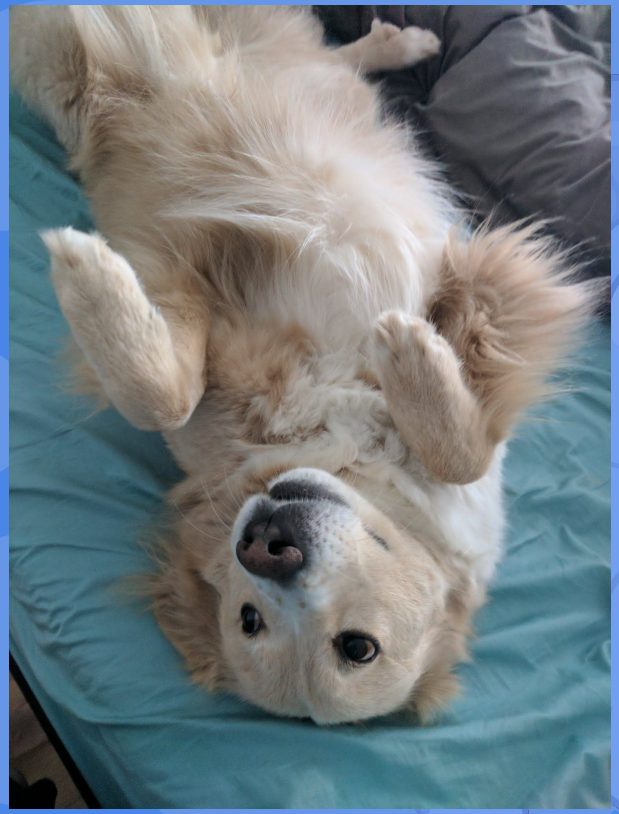
Liz Frost

Twitter: @stillinbeta

slack.k8s.io: @liz

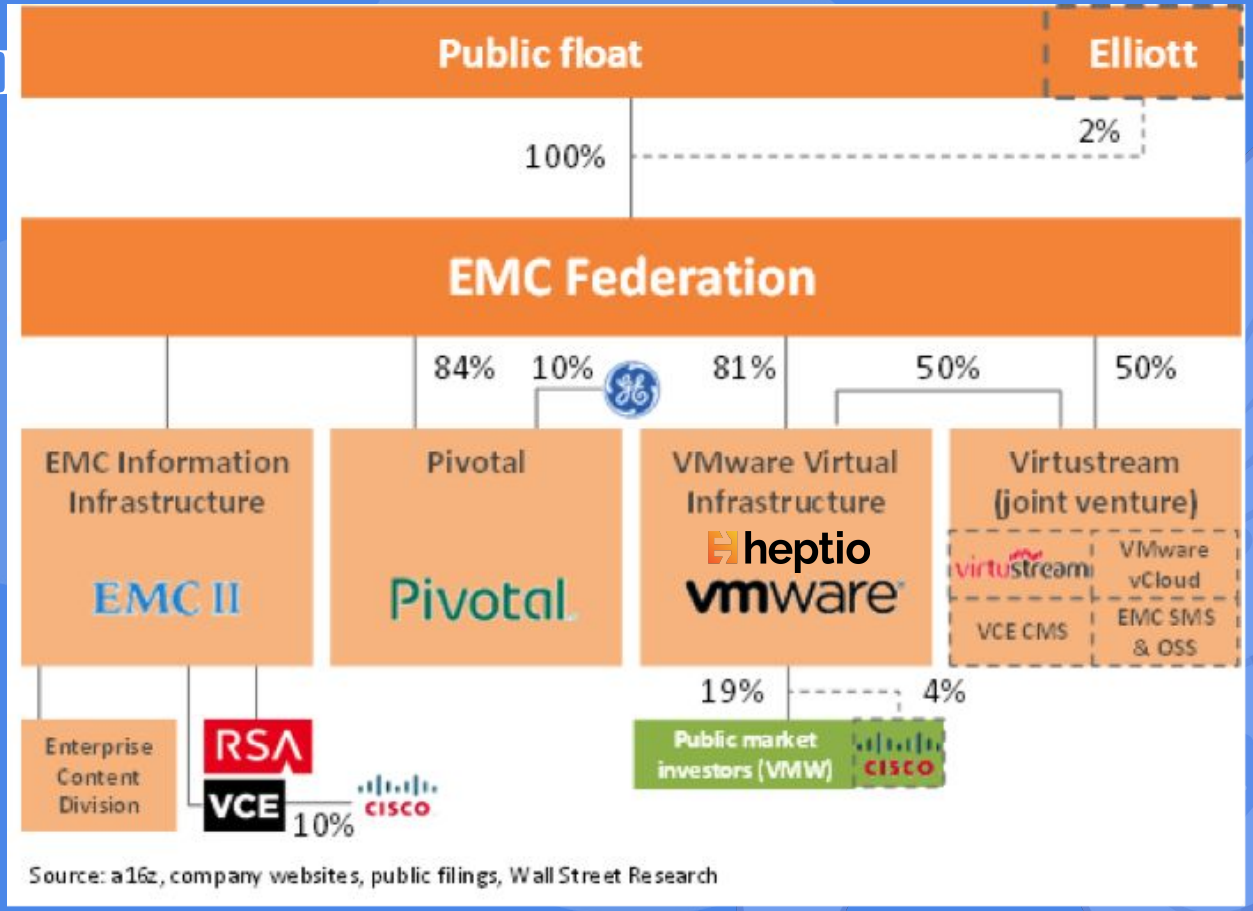
Who am I?

Who am I?

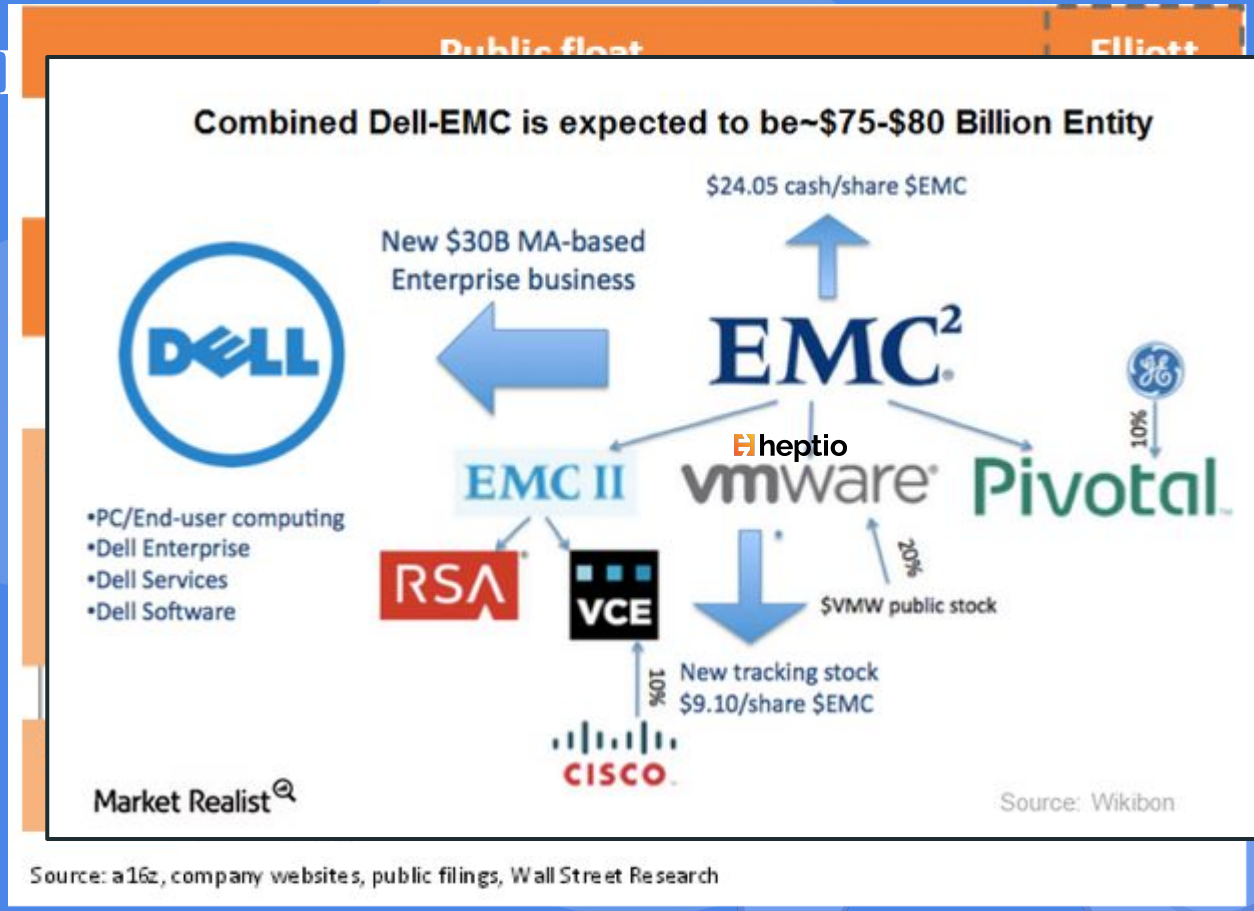


Who am I?

Who a


Public float
Elliott
100%
2%
EMC Federation
84%
10%
81%
50%
50%
EMC Information Infrastructure
EMC II
Pivotal
Pivotal
VMware Virtual Infrastructure
heptio
vmware
Virtustream (joint venture)
virtustream
VMware vCloud
VCE CMS
EMC SMS & OSS
19%
4%
Enterprise Content Division
RSA
VCE
10%
cisco
Public market investors (VMW)
cisco
Source: a16z, company websites, public filings, Wall Street Research

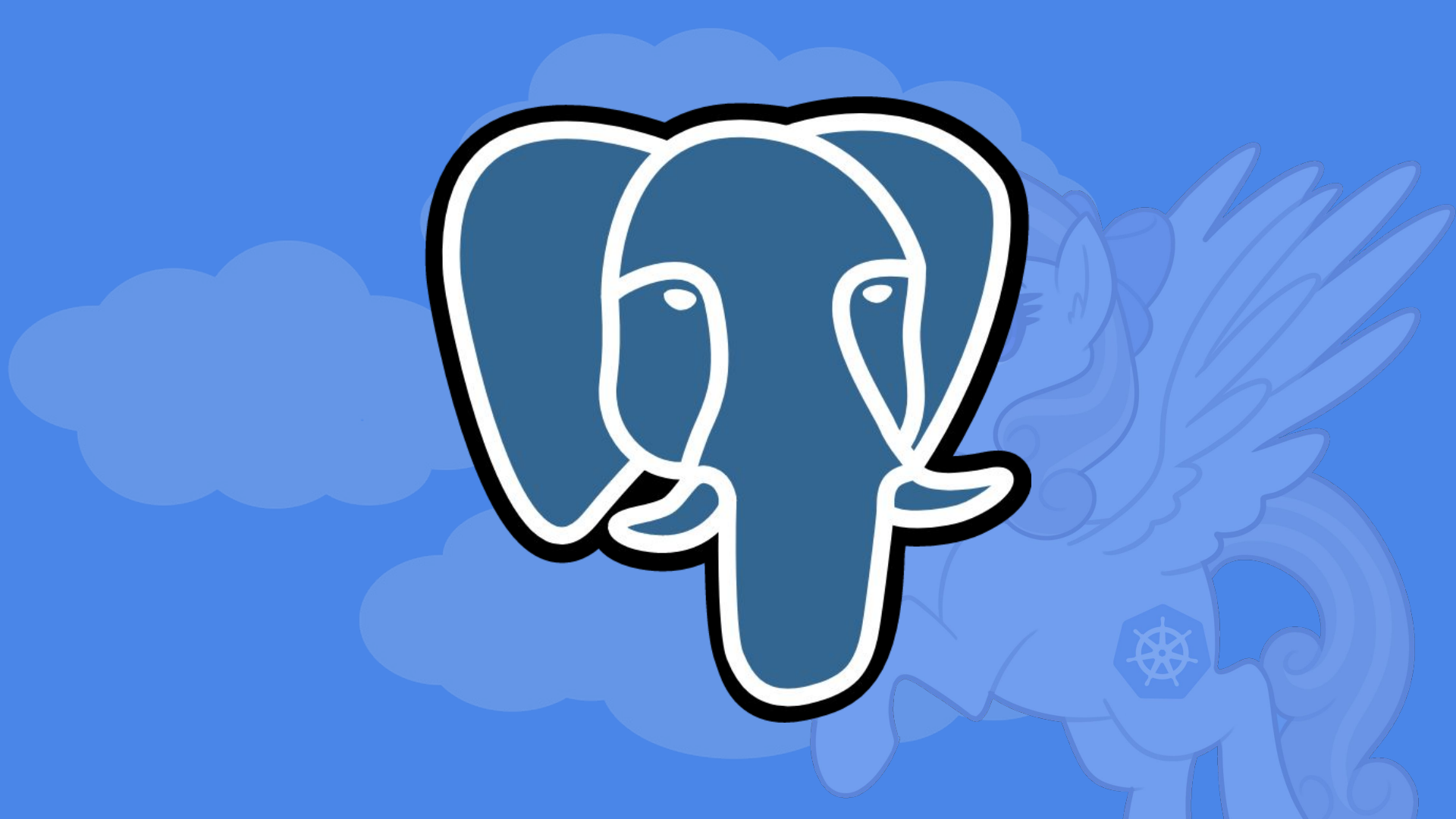
Who ar



Source: a16z, company websites, public filings, Wall Street Research

Who am I?

(Stickers are available)



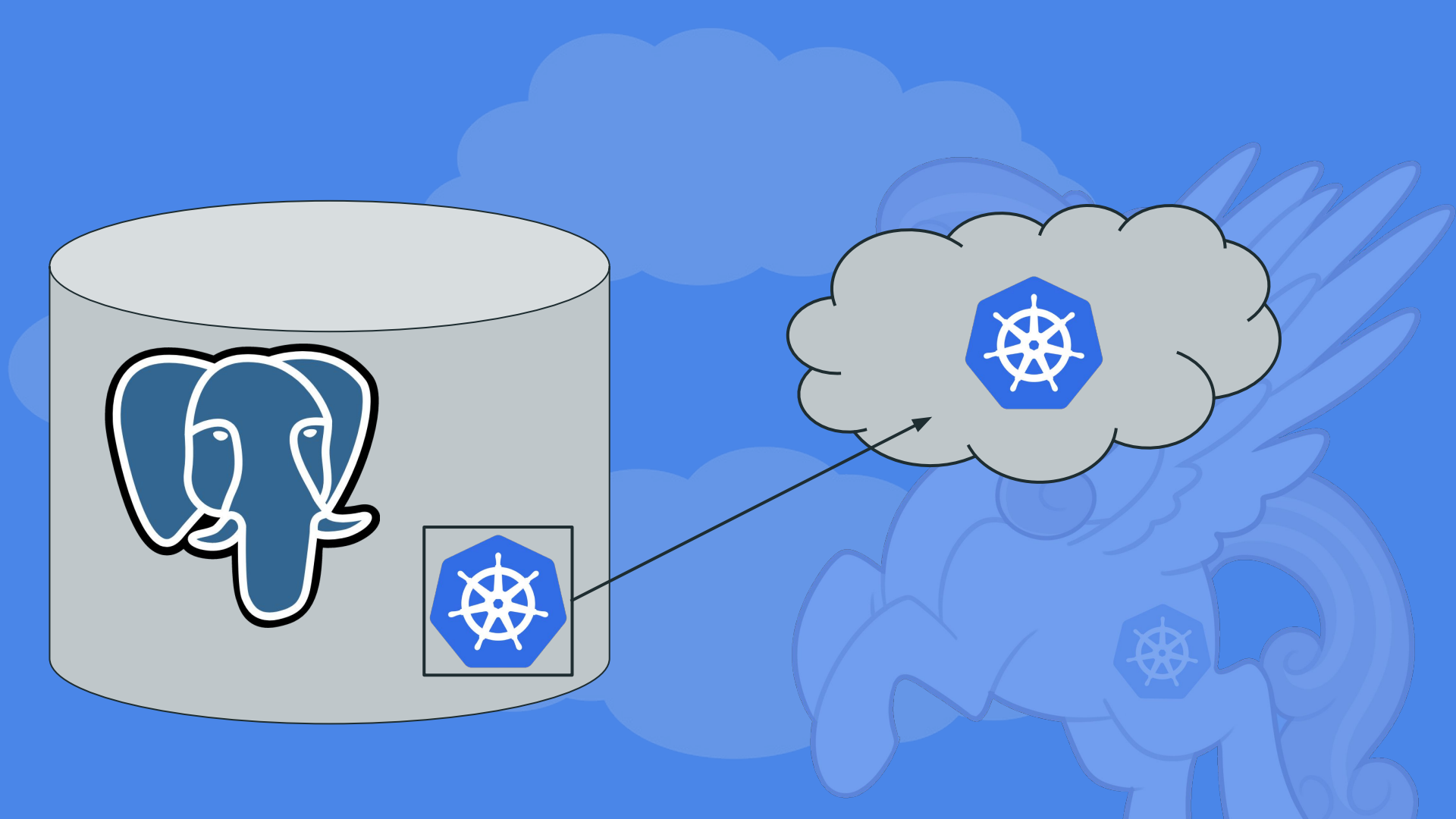
...get it?

What is a Foreign Data Wrapper?

- ODBC
- SQLite

- Cassandra
- Redis

- Hue Bulbs?
- FDWs??
- Kubernetes!



github.com/dennwc/go-fdw

README.md

Go FDW for PostgreSQL

An experimental Go project template for building PostgreSQL Foreign Data Wrappers (FDW).

Tested with PostgreSQL v9.6 and Go 1.8.1 on Ubuntu x64.

Supports:

- Table scan
- EXPLAIN
- Table options

Contributions are welcome!

Getting started

Module entry point is defined in `fdw.go` file (see `SetTable`). This file contains a basic working example, so give it a try. Later you will need to rewrite it to suit your needs.

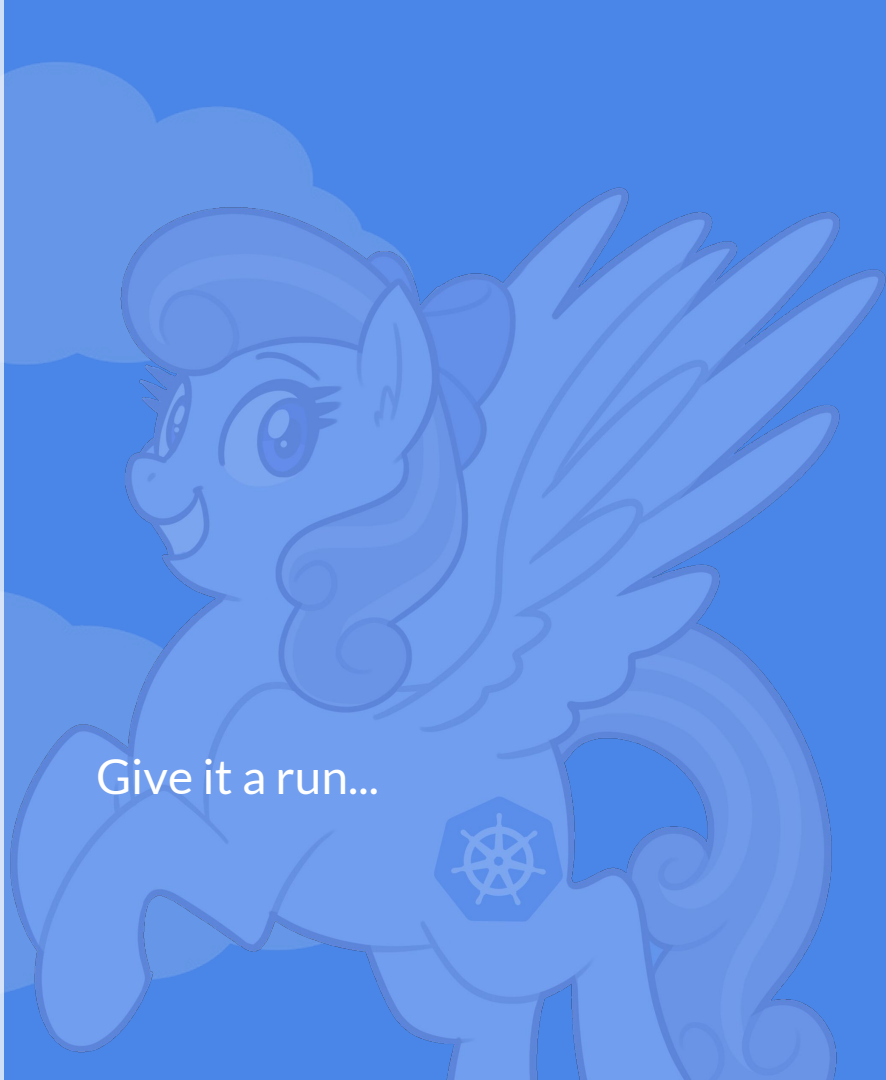
Success! Someone did the hard part for us!

```
CREATE EXTENSION IF NOT EXISTS k8s_fdw;

CREATE SERVER IF NOT EXISTS kind
  FOREIGN DATA WRAPPER k8s_fdw
  OPTIONS (kubeconfig '/kubeconfig');

CREATE FOREIGN TABLE IF NOT EXISTS pods (
  name      text OPTIONS (alias 'metadata.name')
, namespace text OPTIONS (alias 'metadata.namespace')
)
  SERVER kind
  OPTIONS (
    namespace 'kube-system'
  , apiVersion 'v1'
  , kind 'Pod'
  );
```

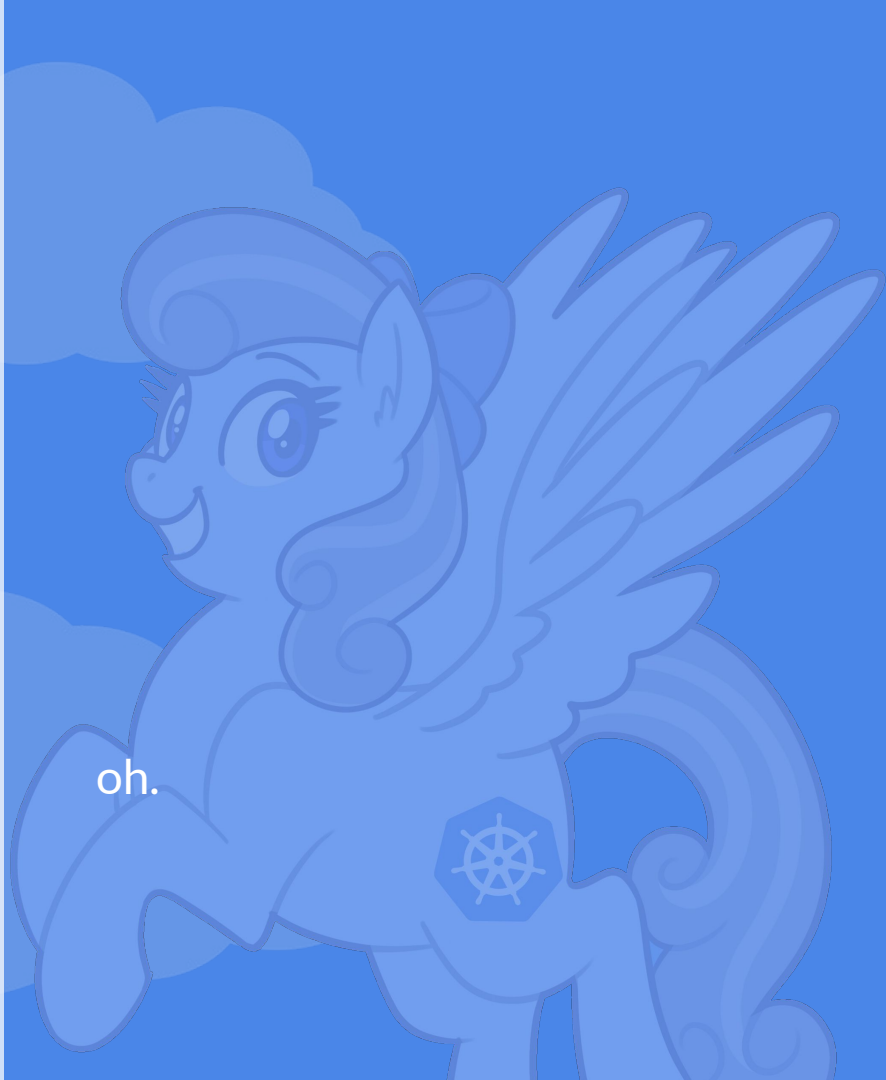
```
CREATE EXTENSION IF NOT EXISTS k8s_fdw;

CREATE SERVER IF NOT EXISTS kind
  FOREIGN DATA WRAPPER k8s_fdw
  OPTIONS (kubeconfig '/kubeconfig');

CREATE FOREIGN TABLE IF NOT EXISTS pods (
  name      text OPTIONS (alias 'metadata.name')
, namespace text OPTIONS (alias 'metadata.namespace')
)
  SERVER kind
  OPTIONS (
    namespace 'kube-system'
  , apiVersion 'v1'
  , kind 'Pod'
  );
```

```
CREATE EXTENSION IF NOT EXISTS k8s_fdw;

CREATE SERVER IF NOT EXISTS kind
  FOREIGN DATA WRAPPER k8s_fdw
  OPTIONS (kubeconfig '/kubeconfig');

CREATE FOREIGN TABLE IF NOT EXISTS pods (
  name      text OPTIONS (alias 'metadata.name')
, namespace text OPTIONS (alias 'metadata.namespace')
)
  SERVER kind
  OPTIONS (
    namespace 'kube-system'
  , apiVersion 'v1'
  , kind 'Pod'
  );
```

```
$ psql --user postgres < test.sql
CREATE EXTENSION
CREATE SERVER
CREATE FOREIGN TABLE
```

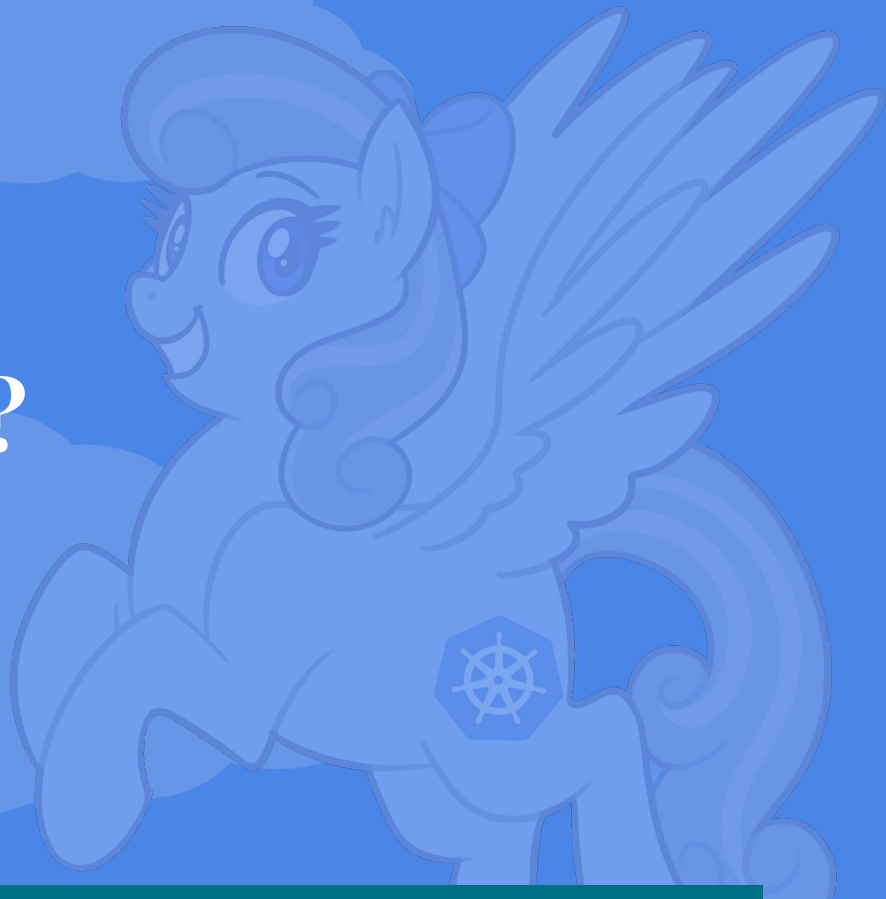


```
$ psql --user postgres < test.sql
CREATE EXTENSION
CREATE SERVER
CREATE FOREIGN TABLE
ERROR:  couldn't get kubeconfig: couldn't
get resource mapper: could not get api
group resources: Get
https://ec2-54-215-202-190.compute-1.amazo
naws.com:6443/api?timeout=32s: net/http:
invalid header field value "postgres:
postgres postgres [local]
SELECT\x00\x00\x00\x00\x00\x00\x00\x00\x00
\x00/v0.0.0 (linux/amd64)
kubernetes/$Format" for key User-Agent
```



```
$ psql --user postgres < test.sql
CREATE EXTENSION
CREATE SERVER
CREATE FOREIGN TABLE
              name                |  namespace
----------------------------------+-------------
 coredns-6f685fffbf-7xrtp         | kube-system
 coredns-6f685fffbf-nzfn7         | kube-system
 etcd-master                      | kube-system
 kube-apiserver-master            | kube-system
 kube-controller-manager-master   | kube-system
 kube-proxy-lk2mq                 | kube-system
 kube-scheduler-master            | kube-system
```

What about more complicated objects?



How about JSONPATH?

```
ALTER FOREIGN TABLE pods
  ADD COLUMN container text
    OPTIONS (alias '{.spec.containers[0].image}')
  ;
```

```
SELECT * from PODS;
```

name	namespace	container
coredns-6f685fffbf-7xrtp	kube-system	k8s.gcr.io/coredns:1.2.6
coredns-6f685fffbf-nzfn7	kube-system	k8s.gcr.io/coredns:1.2.6
etcd-master	kube-system	k8s.gcr.io/etcd:3.2.24
kube-apiserver-master	kube-system	k8s.gcr.io/kube-apiserver:v1.12.1
kube-controller-manager-master	kube-system	k8s.gcr.io/kube-controller-manager:v1.12.1
kube-proxy-lk2mq	kube-system	k8s.gcr.io/kube-proxy:v1.12.1
kube-scheduler-master	kube-system	k8s.gcr.io/kube-scheduler:v1.12.1

How about a map?

```
ALTER FOREIGN TABLE pods
  ADD COLUMN labels jsonb
      OPTIONS (alias 'metadata.labels')
  ;
```

```
SELECT name, labels FROM pods;

              name               |                                            labels
---------------------------------+--------------------------------------------------------------------------------------------------------
 coredns-6f685fffbf-7xrtp        | {"k8s-app": "kube-dns", "pod-template-hash": "6f685fffbf"}
 coredns-6f685fffbf-nzfn7        | {"k8s-app": "kube-dns", "pod-template-hash": "6f685fffbf"}
 etcd-master                     | {"tier": "control-plane", "component": "etcd"}
 kube-apiserver-master           | {"tier": "control-plane", "component": "kube-apiserver"}
 kube-controller-manager-master | {"tier": "control-plane", "component": "kube-controller-manager"}
 kube-proxy-lk2mq                | {"k8s-app": "kube-proxy", "pod-template-generation": "1", "controller-revision-hash": "6cbfff58bb"}
 kube-scheduler-master           | {"tier": "control-plane", "component": "kube-scheduler"}
(7 rows)
```

```
SELECT name, container, labels->'component' AS component FROM pods;
```

```
             name              |                  container                  |         component
-------------------------------+---------------------------------------------+---------------------------
 coredns-6f685fffbf-7xrtp      | k8s.gcr.io/coredns:1.2.6                    |
 coredns-6f685fffbf-nzfn7      | k8s.gcr.io/coredns:1.2.6                    |
 etcd-master                   | k8s.gcr.io/etcd:3.2.24                      | "etcd"
 kube-apiserver-master         | k8s.gcr.io/kube-apiserver:v1.12.1           | "kube-apiserver"
 kube-controller-manager-master | k8s.gcr.io/kube-controller-manager:v1.12.1 | "kube-controller-manager"
 kube-proxy-lk2mq              | k8s.gcr.io/kube-proxy:v1.12.1               |
 kube-scheduler-master         | k8s.gcr.io/kube-scheduler:v1.12.1           | "kube-scheduler"
```

And for my final trick:

```
CREATE FOREIGN TABLE IF NOT EXISTS replica_sets (
  name      text OPTIONS (alias 'metadata.name')
, replicas  bigint OPTIONS (alias 'status.replicas')
, available bigint OPTIONS (alias 'status.availableReplicas')
)
  SERVER kind
  OPTIONS (
    namespace 'kube-system'
  , apiVersion 'apps/v1'
  , kind 'ReplicaSet'
  );

CREATE FOREIGN TABLE IF NOT EXISTS deployments (
  name      text OPTIONS (alias 'metadata.name')
, replicas  bigint OPTIONS (alias 'status.replicas')
, available bigint OPTIONS (alias 'status.availableReplicas')
)
  SERVER kind
  OPTIONS (
    namespace 'kube-system'
  , apiVersion 'apps/v1'
  , kind 'Deployment'
  );
```

```
CREATE FOREIGN TABLE IF NOT EXISTS replica_sets (
  name      text OPTIONS (alias 'metadata.name')
, replicas  bigint OPTIONS (alias 'status.replicas')
, available bigint OPTIONS (alias 'status.availableReplicas')
)
  SERVER kind
  OPTIONS (
    namespace 'kube-system'
  , apiVersion 'apps/v1'
  , kind 'ReplicaSet'
  );

CREATE FOREIGN TABLE IF NOT EXISTS deployments (
  name      text OPTIONS (alias 'metadata.name')
, replicas  bigint OPTIONS (alias 'status.replicas')
, available bigint OPTIONS (alias 'status.availableReplicas')
)
  SERVER kind
  OPTIONS (
    namespace 'kube-system'
  , apiVersion 'apps/v1'
  , kind 'Deployment'
  );
```

```
SELECT "deployments"."name" AS deployment_name
     , "replica_sets"."name" as replica_name
     , "pods"."name" AS pod_name
  FROM deployments
  JOIN replica_sets on "replica_sets"."name" LIKE "deployments"."name" || '-%'
  JOIN pods on "pods"."name" LIKE "replica_sets"."name" || '-%';
```

```
 deployment_name |    replica_name    |         pod_name
-----------------+--------------------+--------------------------
 coredns         | coredns-6f685fffbf | coredns-6f685fffbf-7xrtp
 coredns         | coredns-6f685fffbf | coredns-6f685fffbf-nzfn7
```

What *doesn't* work?

- Column types mostly ignored

What *doesn't* work?

- Column types mostly ignored
- If it's a not a integel, string, or map…

  ...just kind of give up

What *doesn't* work?

- Column types mostly ignored
- If it's a not a number, string, or map...

  ...just kind of give up
- The codebase not being a not of

  spaghetti

Questions? Concerns?

Come find me!
I'm the one with pink hair!

Twitter: @stillinbeta

slack.k8s.io: @liz

Github: github.com/liztio/k8s-fdw

Docker: liztio/k8s_fdw:master