ENTERPRISE DEVOPS TECHCON

Disaster Recovery and Kubernetes - What could possibly go wrong ?

Presenter
Eric De Witte

Photo by Jason Blackeye on Unsplash

Eric DE WITTE

bio

VMware Tanzu Emerging Solution Engineer

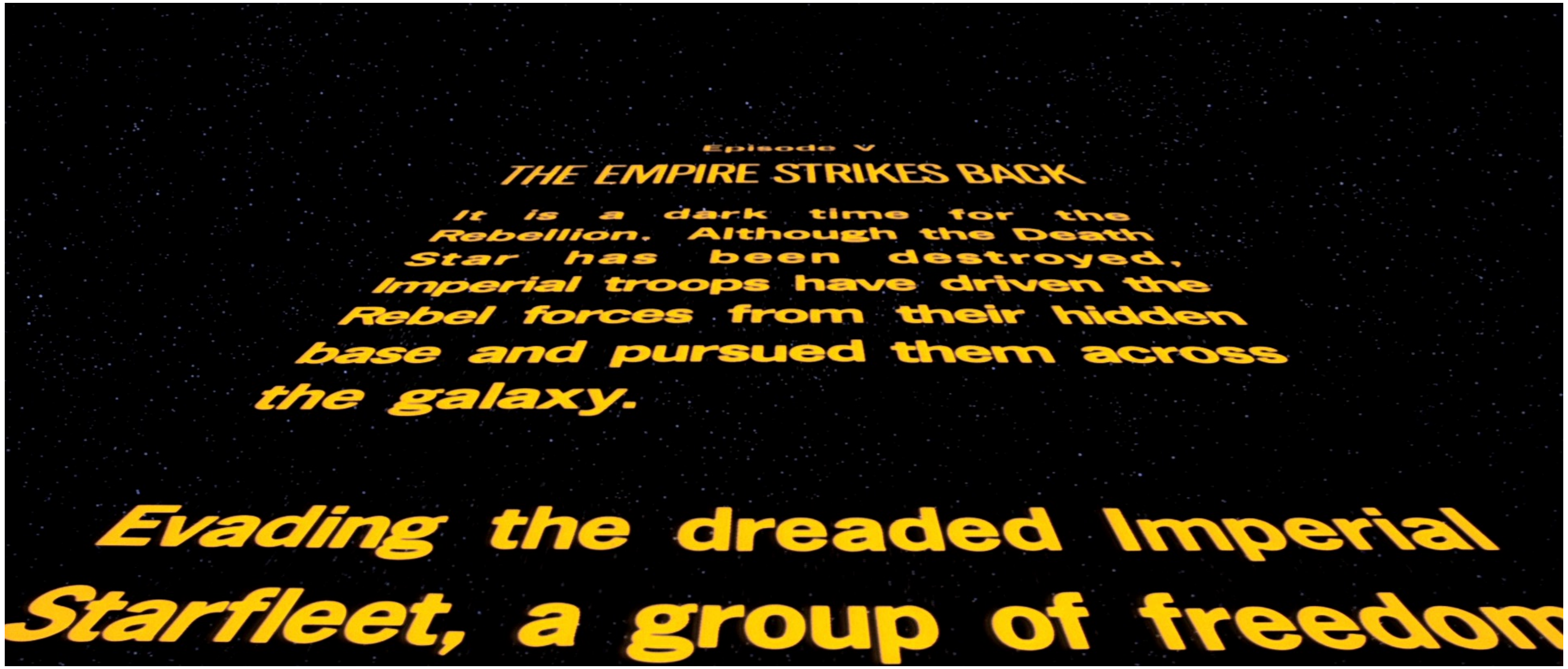
Twitter: @vEDW

Mail: edewitte@vmware.com

Spending too much time in labs ?

Why this talk ?

- It all started a long time ago …
- A customer POC. Datacenters, future edge locations … and a question about Disaster Recovery



Episode V

THE EMPIRE STRIKES BACK

It is a dark time for the Rebellion. Although the Death Star has been destroyed, Imperial troops have driven the Rebel forces from their hidden base and pursued them across the galaxy.

Evading the dreaded Imperial Starfleet, a group of freedom

Risk Assessment

- Datacenter downtime :
  - un-planned
    - Fire
    - Flooding
    - Telco failures
  - Planned
    - Facilities Maintenance
- Other ?

(I’m focusing on multiple failures impacting 1 site)

TELL ME SOMETHING INTERESTING ABOUT YOURSELF.

Traditional vs Cloud Native

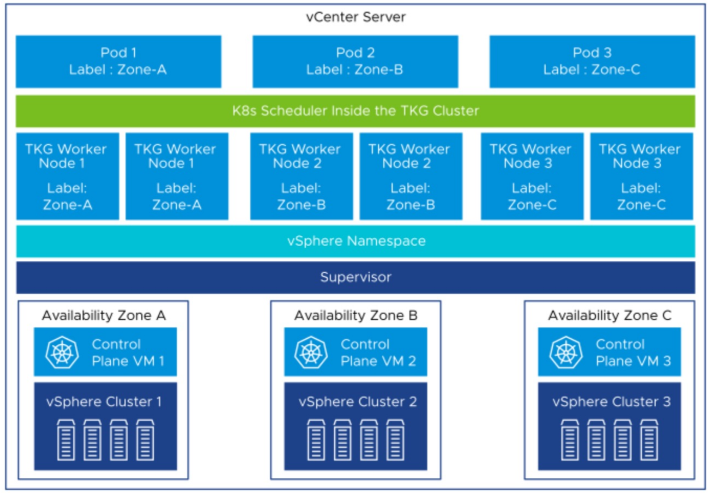
Typical Enterprise customer

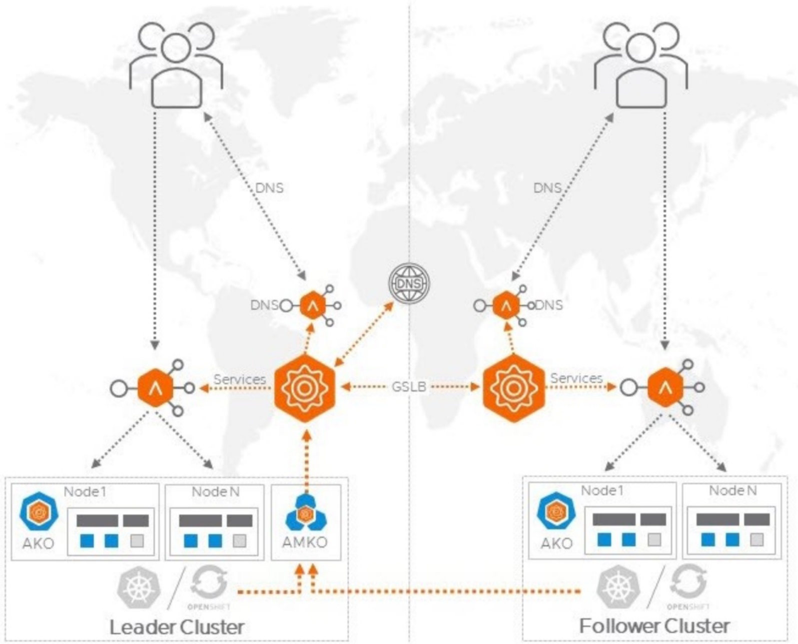
- Mainly Infrastructure based resiliency
- Typically using 1 or 2 Datacenters within a “region”
- Using technologies like:
    - Storage Based replication
    - VMware Site Recovery Manager
    - vSphere Metro Stretched Cluster
- Perform DR testing 1x / year (maybe)

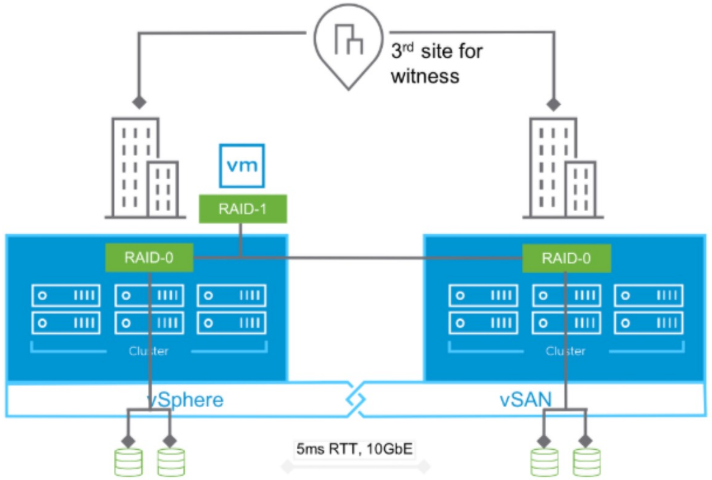
Cloud based unicorns

- Mainly Application based resiliency
- Uses cloud regions backed by availability zones
- Perform resiliency testing using Chaos Engineering on a regular basis

Possible architectures for Kubernetes site resiliency on vSphere







The dreaded stretched-cluster topic

- Running K8S on vSphere stretched cluster is an ANTI-PATTERN.
- You should never need it if your application is a well architected Cloud Native Application (including the data layers)
- unfortunately, not all applications are well architected hence the request from customers to support stretched clusters.

Dual Data Center Architecture - ETCD Mismatch

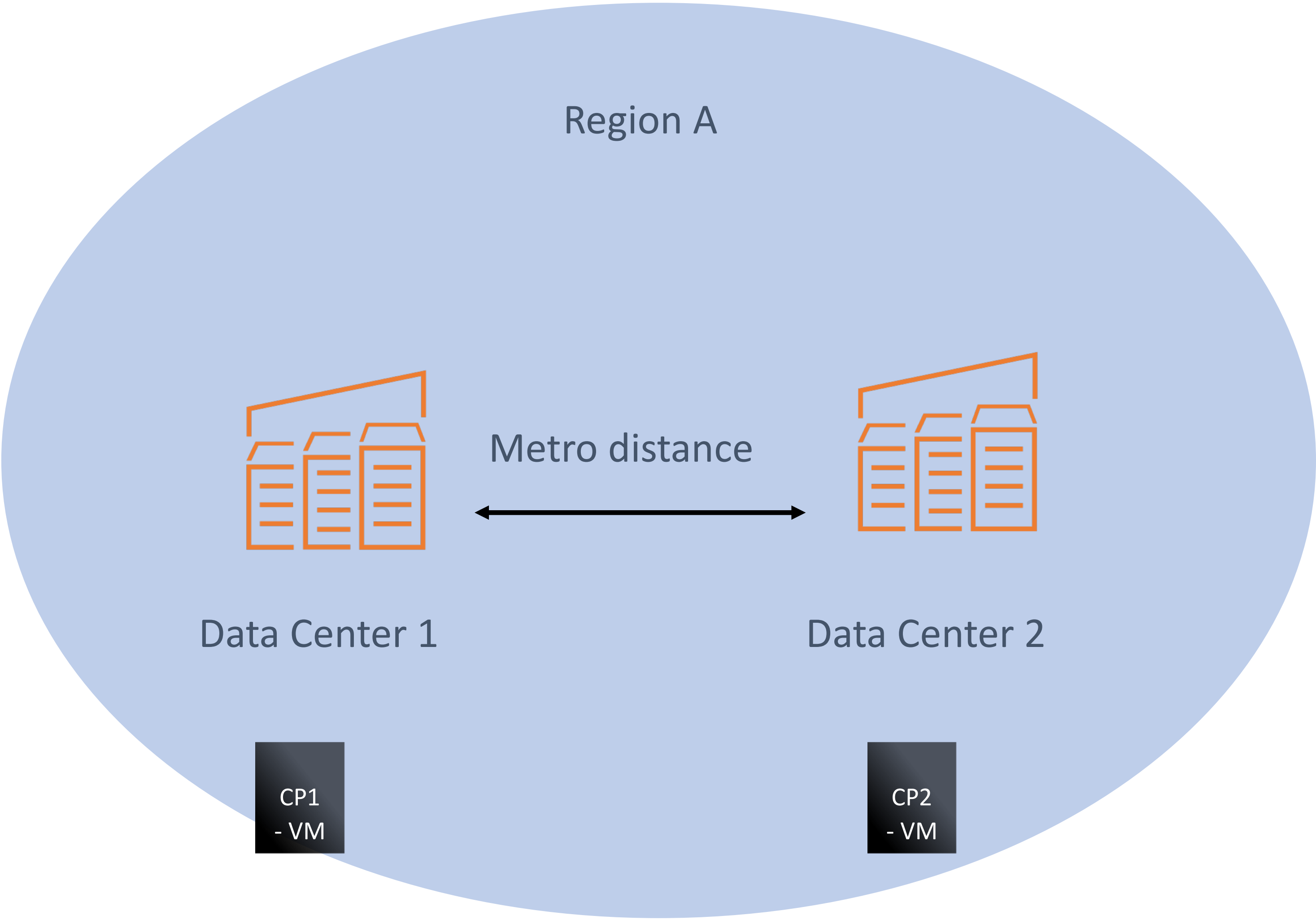
Some facts:

- Kubernetes uses ETCD
- ETCD relies on the RAFT consensus algorithm
- In high availability mode, you need an odd number of etcd nodes **3**, 5, 7…

CP3
- VM

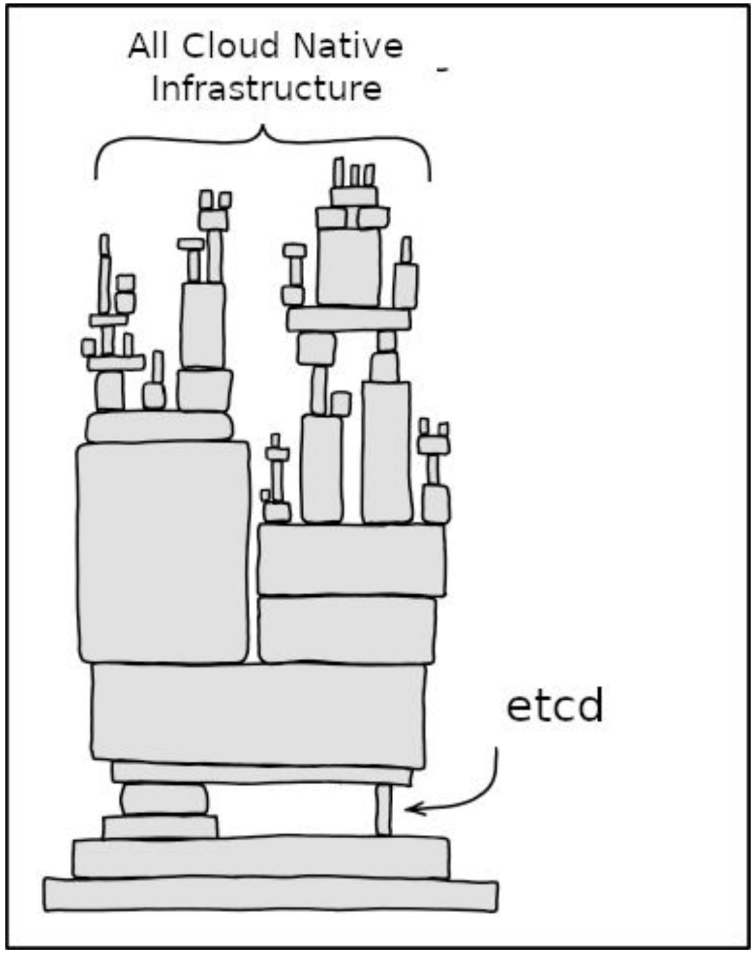
Problem:

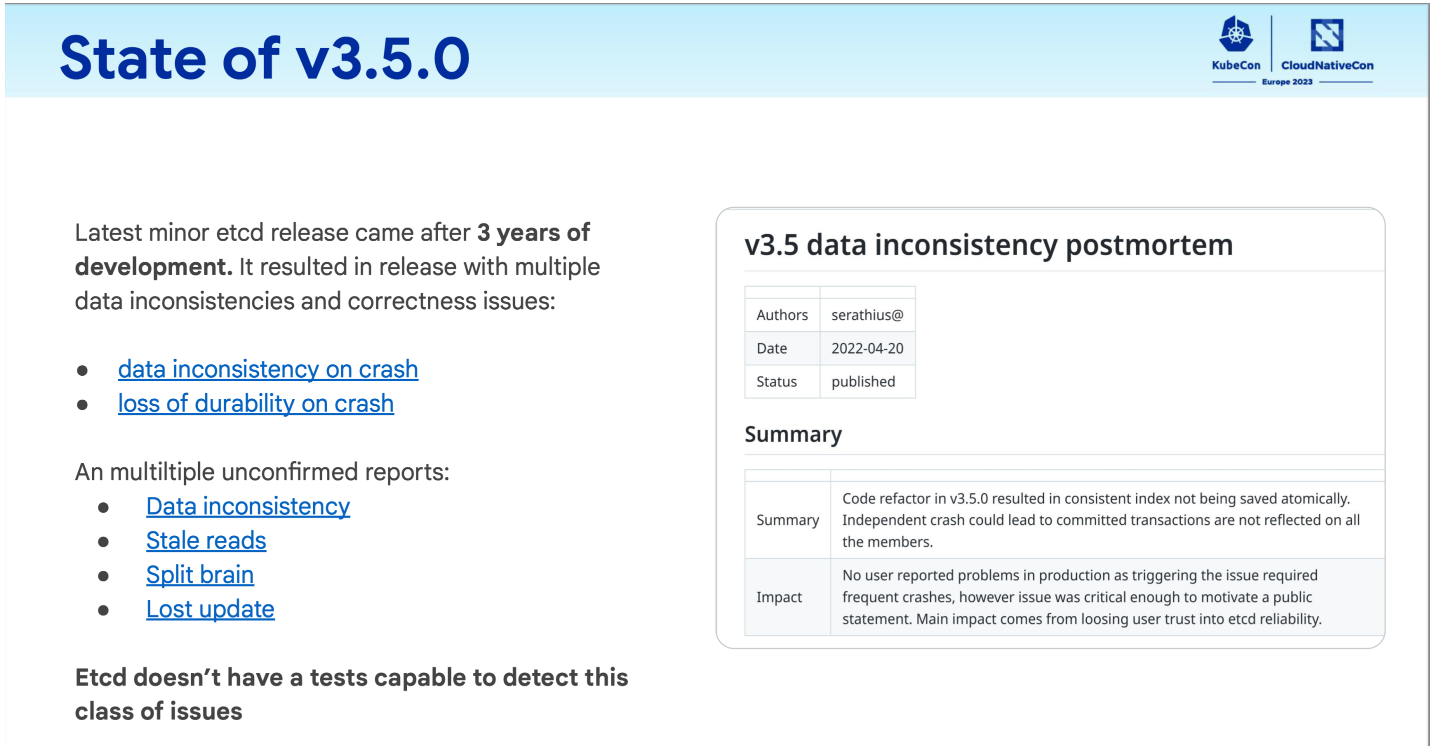
- Where to put the 3rd Control Plane Node??
- How to ensure Control Plane nodes are always distributed across our two data centers?



ETCD – Kubecon EU 2023

On the Hunt for Etcd Data Inconsistencies - *Marek Siarkowicz, Google*





State of v3.5.0

Latest minor etcd release came after **3 years of development.** It resulted in release with multiple data inconsistencies and correctness issues:

- data inconsistency on crash
- loss of durability on crash

An multiltiple unconfirmed reports:

- Data inconsistency
- Stale reads
- Split brain
- Lost update

Etcd doesn't have a tests capable to detect this class of issues

v3.5 data inconsistency postmortem

Authors	serathius@
Date	2022-04-20
Status	published

Summary

Summary	Code refactor in v3.5.0 resulted in consistent index not being saved atomically. Independent crash could lead to committed transactions are not reflected on all the members.
Impact	No user reported problems in production as triggering the issue required frequent crashes, however issue was critical enough to motivate a public statement. Main impact comes from loosing user trust into etcd reliability.

https://youtu.be/IIMs0EjQZHg

Some ETCD issues fixed

- Provide a better liveness probe for when etcd runs as a Kubernetes pod
  - https://github.com/etcd-io/etcd/issues/13340

- Improvements for etcd liveness probes
  - https://github.com/kubernetes/kubeadm/issues/2567

- Add Patches field in InitConfiguration and JoinConfiguration
  - https://github.com/kubernetes-sigs/cluster-api/pull/5897

Current k8S on vSphere VSAN Stretched Cluster support

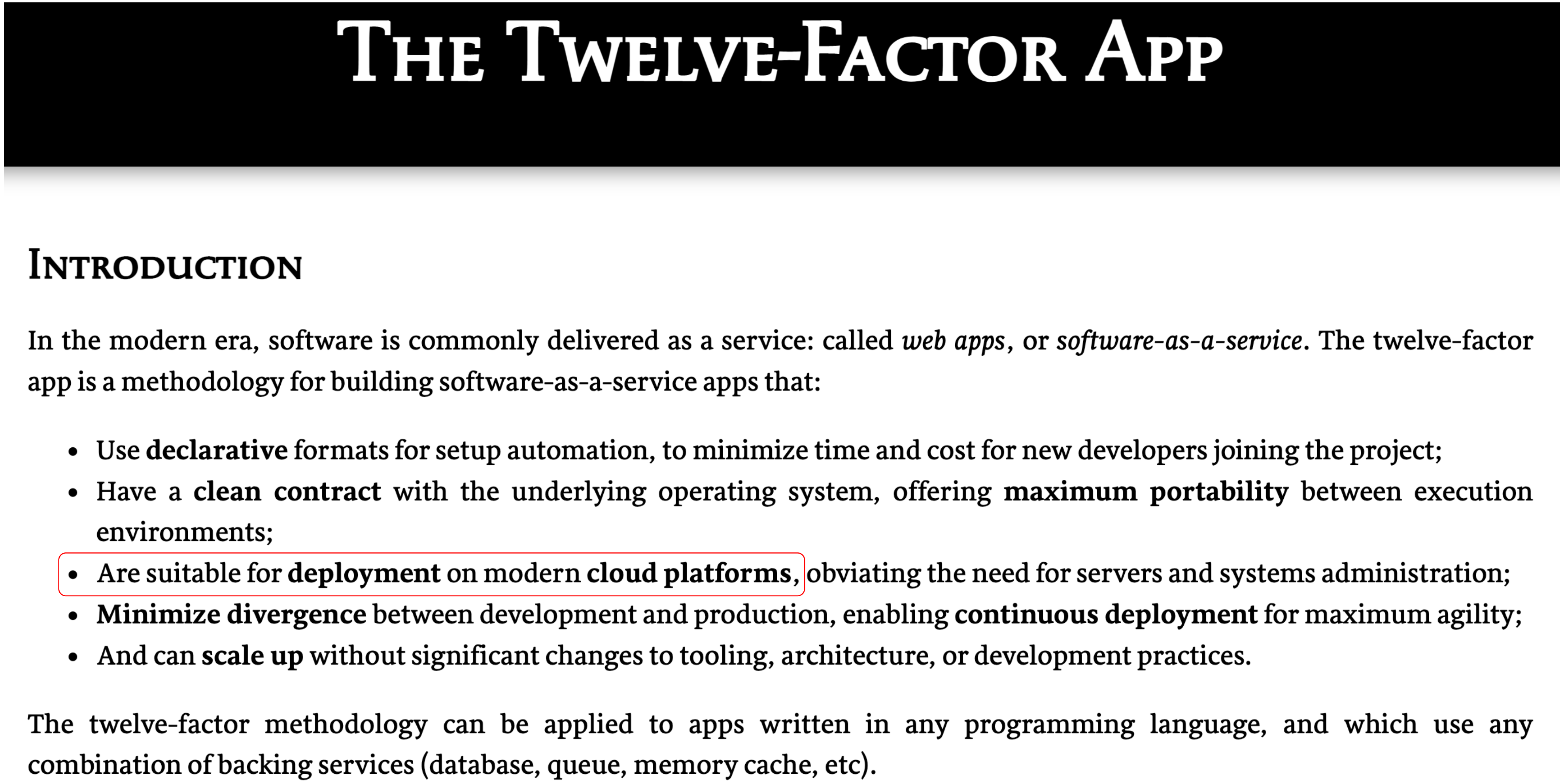
- Tanzu Kubernetes Grid integrated edition:
  - Automation : BOSH
  - Supported when following solutions guide :
  - https://docs.vmware.com/en/VMware-Tanzu-Kubernetes-Grid-Integrated-Edition/1.14/tkgi/GUID-solutions-using-vsan-stretched-clusters.pdf

- vSphere with Tanzu:
  - Automation : K8S Cluster-API
  - Not supported at this time (yet)

- Tanzu Kubernetes Grid
  - Automation : K8S Cluster-API
  - Not Supported at this time (yet)
  - AZ docs https://docs.vmware.com/en/VMware-Tanzu-Kubernetes-Grid/2.4/using-tkg/workload-clusters-multi-az-vsphere.html

12 factor apps

https://12factor.net/



THE TWELVE-FACTOR APP

INTRODUCTION

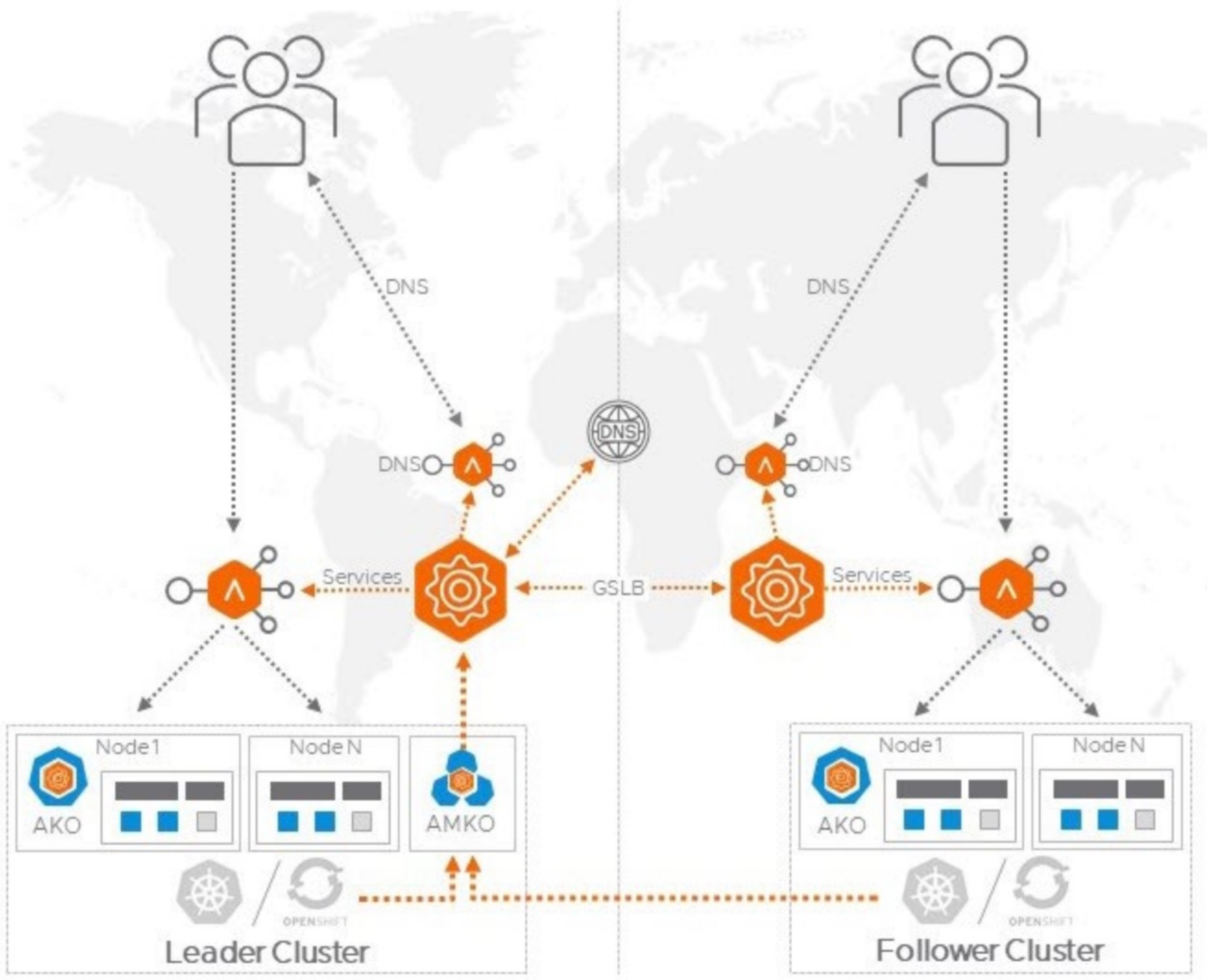
In the modern era, software is commonly delivered as a service: called *web apps*, or *software-as-a-service*. The twelve-factor app is a methodology for building software-as-a-service apps that:

- Use **declarative** formats for setup automation, to minimize time and cost for new developers joining the project;
- Have a **clean contract** with the underlying operating system, offering **maximum portability** between execution environments;
- Are suitable for **deployment** on modern **cloud platforms**, obviating the need for servers and systems administration;
- **Minimize divergence** between development and production, enabling **continuous deployment** for maximum agility;
- And can **scale up** without significant changes to tooling, architecture, or development practices.

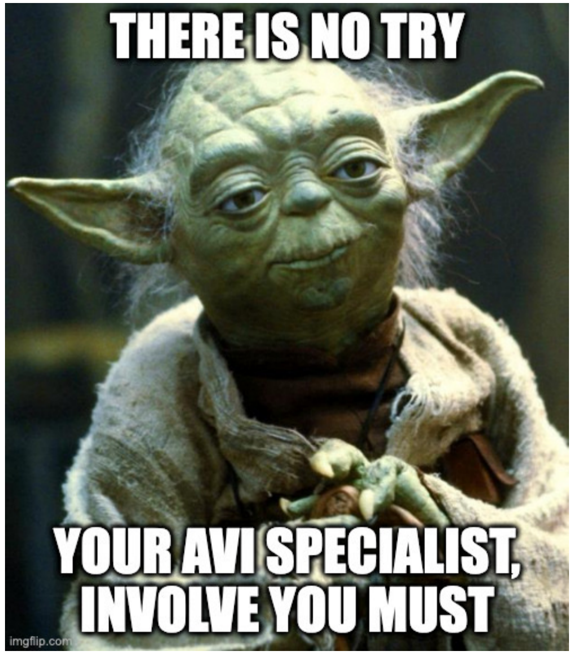
The twelve-factor methodology can be applied to apps written in any programming language, and which use any combination of backing services (database, queue, memory cache, etc).

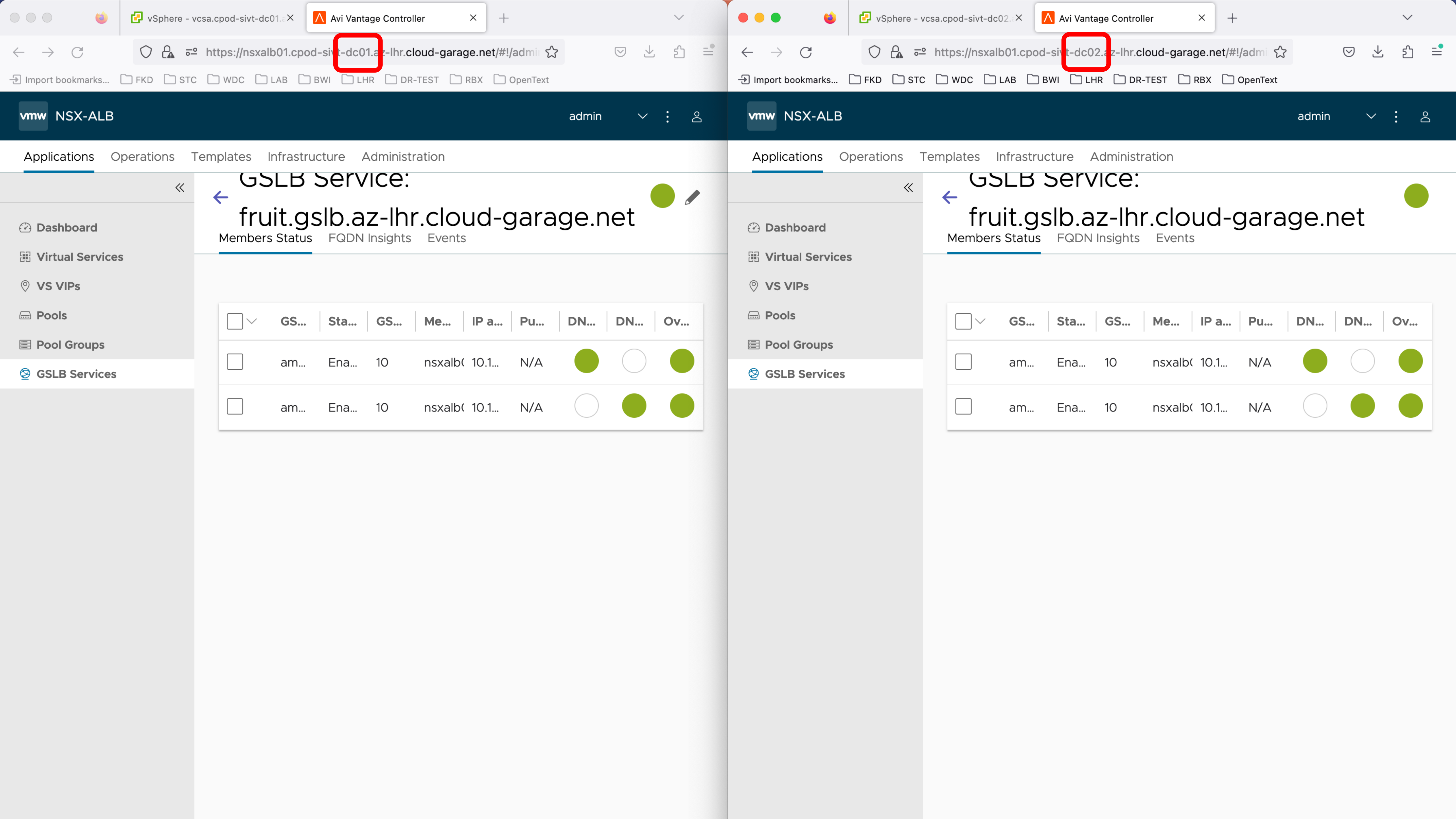
Let’s go the cloud native way : application based resiliency !

https://blog.andreasm.io/2022/10/23/gslb-with-ako-amko-nsx-advanced-loadbalancer



- ○ Setup dual site
- ○ Setup GSLB
- ○ Done (?)





GSLB Service:
fruit.gslb.az-lhr.cloud-garage.net

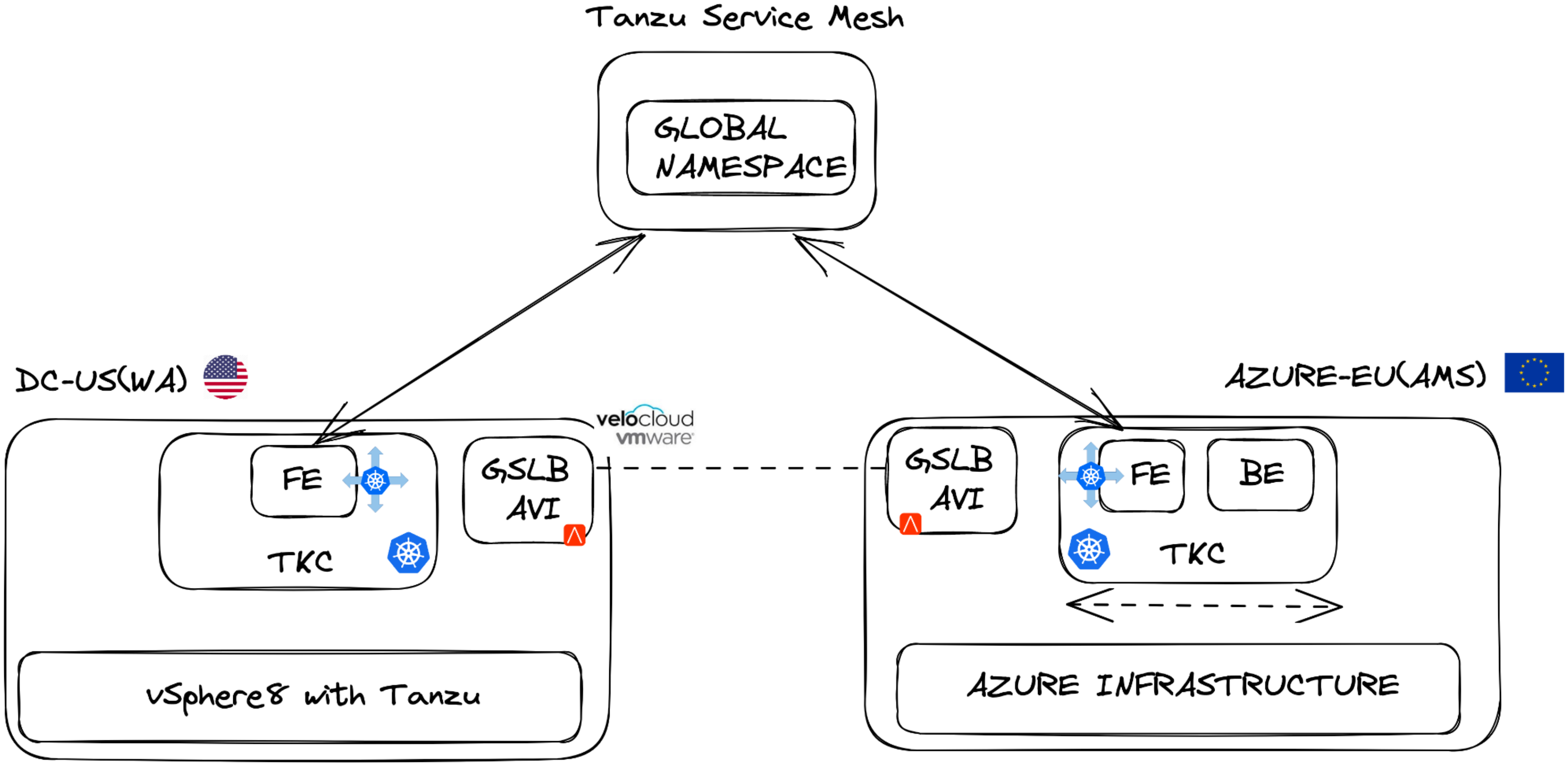
□	GS...	Sta...	GS...	Me...	IP a...	Pu...	DN...	DN...	Ov...
□	am...	Ena...	10	nsxalb(	10.1...	N/A	●	○	●
□	am...	Ena...	10	nsxalb(	10.1...	N/A	○	●	●

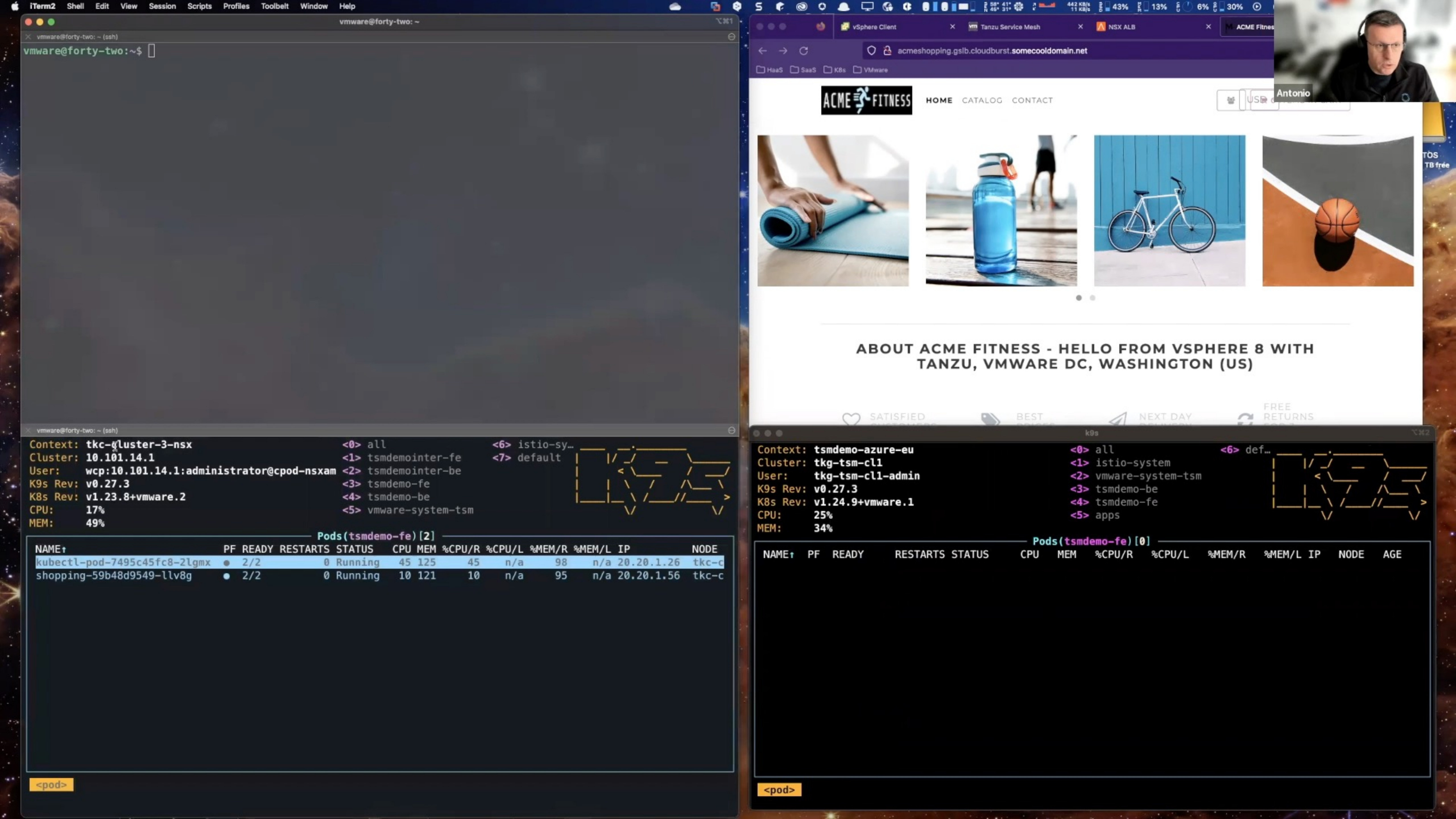
GSLB Service:
fruit.gslb.az-lhr.cloud-garage.net

□	GS...	Sta...	GS...	Me...	IP a...	Pu...	DN...	DN...	Ov...
□	am...	Ena...	10	nsxalb(	10.1...	N/A	●	○	●
□	am...	Ena...	10	nsxalb(	10.1...	N/A	○	●	●

Let’s go the cloud native way : application based resiliency with TSM

https://apps-cloudmgmt.techzone.vmware.com/blog/cloud-bursting-tanzu-service-mesh





iTerm2 Shell Edit View Session Scripts Profiles Toolbelt Window Help

```
vmware@forty-two:~$
```

acmeshopping.gslb.cloudburst.somecooldomain.net

ABOUT ACME FITNESS - HELLO FROM VSPHERE 8 WITH TANZU, VMWARE DC, WASHINGTON (US)

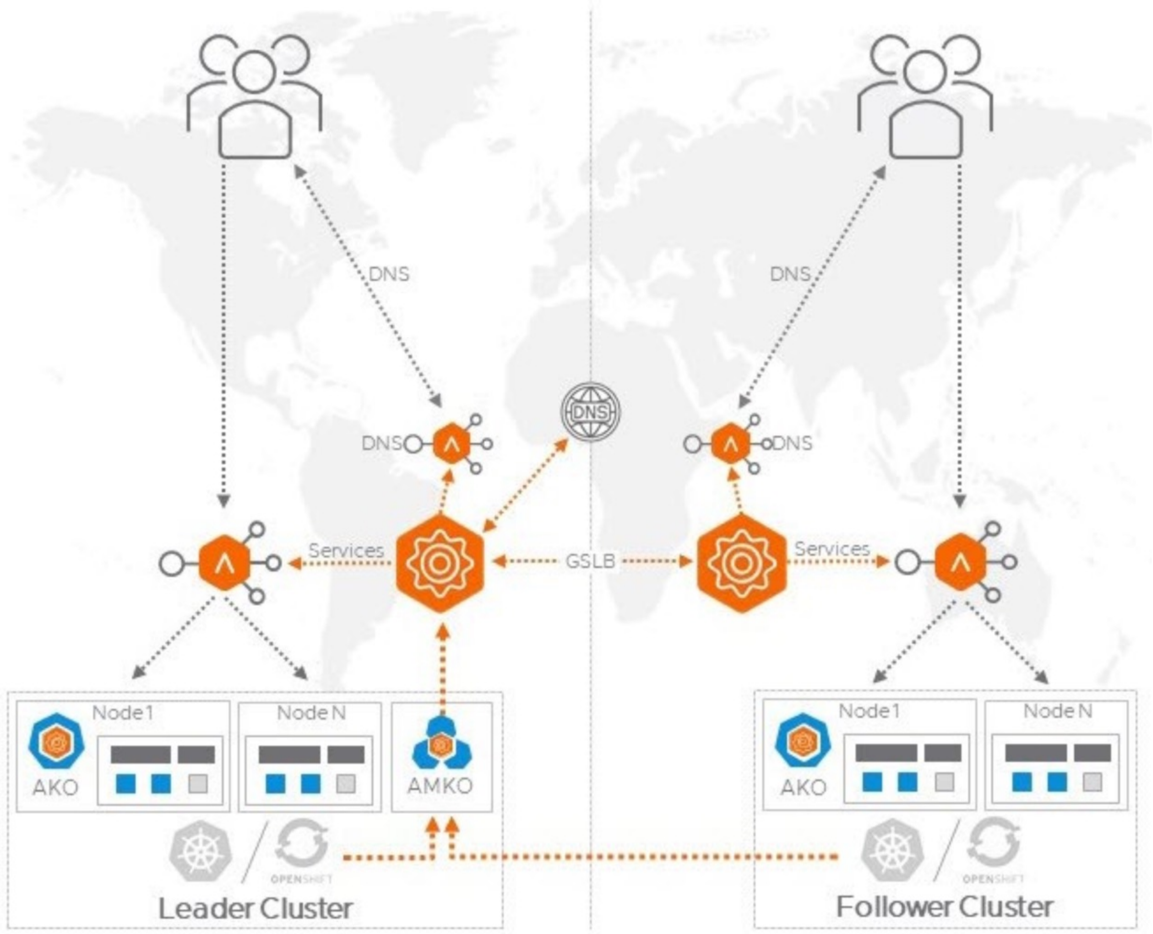
vmware@forty-two: ~ (ssh)

```
Context: tkc-cluster-3-nsx                                <0> all                   <6> istio-sy…
Cluster: 10.101.14.1                                      <1> tsmdemointer-fe       <7> default
User:    wcp:10.101.14.1:administrator@cpod-nsxam <2> tsmdemointer-be
K9s Rev: v0.27.3                                          <3> tsmdemo-fe
K8s Rev: v1.23.8+vmware.2                                 <4> tsmdemo-be
CPU:     17%                                              <5> vmware-system-tsm
MEM:     49%
                                       Pods(tsmdemo-fe)[2]
NAME↑                           PF READY RESTARTS STATUS   CPU MEM %CPU/R %CPU/L %MEM/R %MEM/L IP          NODE
kubectl-pod-7495c45fc8-2lgmx    ●  2/2          0 Running   45 125     45    n/a     98    n/a 20.20.1.26  tkc-c
shopping-59b48d9549-llv8g       ●  2/2          0 Running   10 121     10    n/a     95    n/a 20.20.1.56  tkc-c

<pod>
```

k9s

```
Context: tsmdemo-azure-eu                      <0> all                 <6> def…
Cluster: tkg-tsm-cl1                           <1> istio-system
User:    tkg-tsm-cl1-admin                     <2> vmware-system-tsm
K9s Rev: v0.27.3                               <3> tsmdemo-be
K8s Rev: v1.24.9+vmware.1                      <4> tsmdemo-fe
CPU:     25%                                   <5> apps
MEM:     34%
                                   Pods(tsmdemo-fe)[0]
NAME↑  PF  READY    RESTARTS STATUS    CPU   MEM  %CPU/R  %CPU/L  %MEM/R  %MEM/L IP  NODE  AGE

<pod>
```

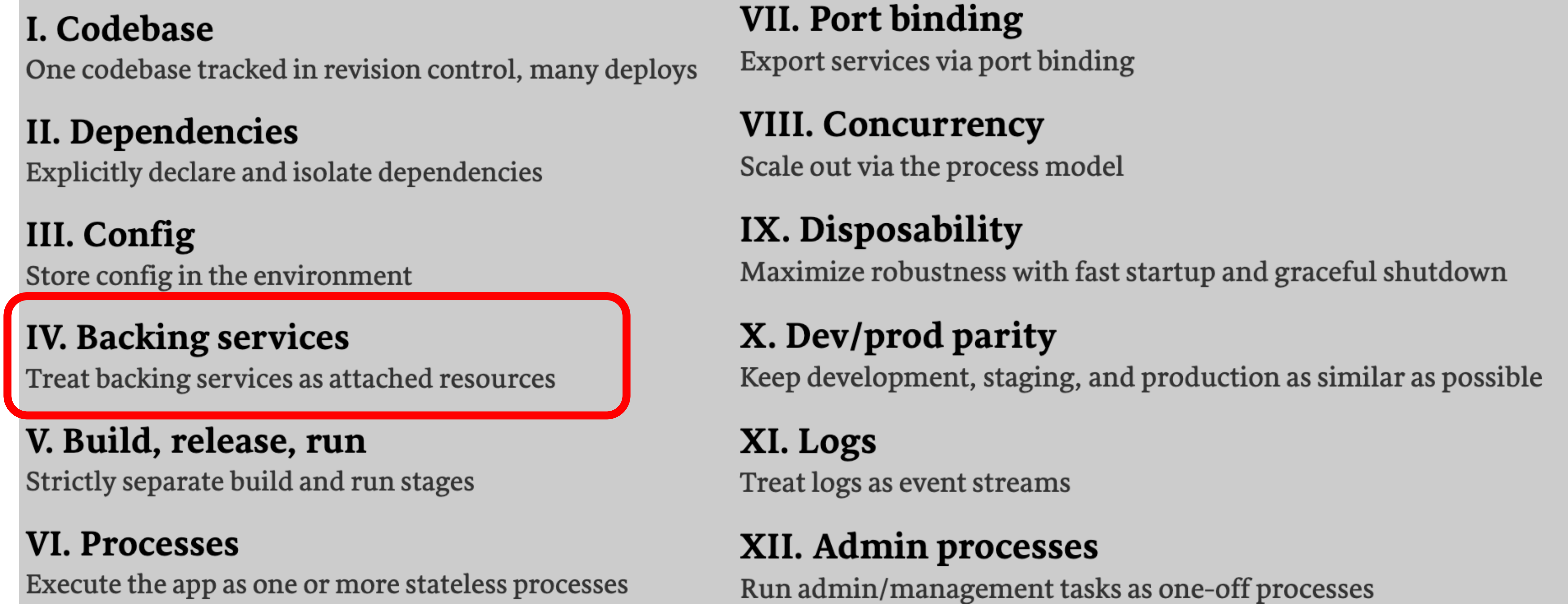
Let's go the cloud native way : application based resiliency !



Managing DR of stateful services is super challenging.

12 factor apps

https://12factor.net/



I. Codebase
One codebase tracked in revision control, many deploys

II. Dependencies
Explicitly declare and isolate dependencies

III. Config
Store config in the environment

IV. Backing services
Treat backing services as attached resources

V. Build, release, run
Strictly separate build and run stages

VI. Processes
Execute the app as one or more stateless processes

VII. Port binding
Export services via port binding

VIII. Concurrency
Scale out via the process model

IX. Disposability
Maximize robustness with fast startup and graceful shutdown

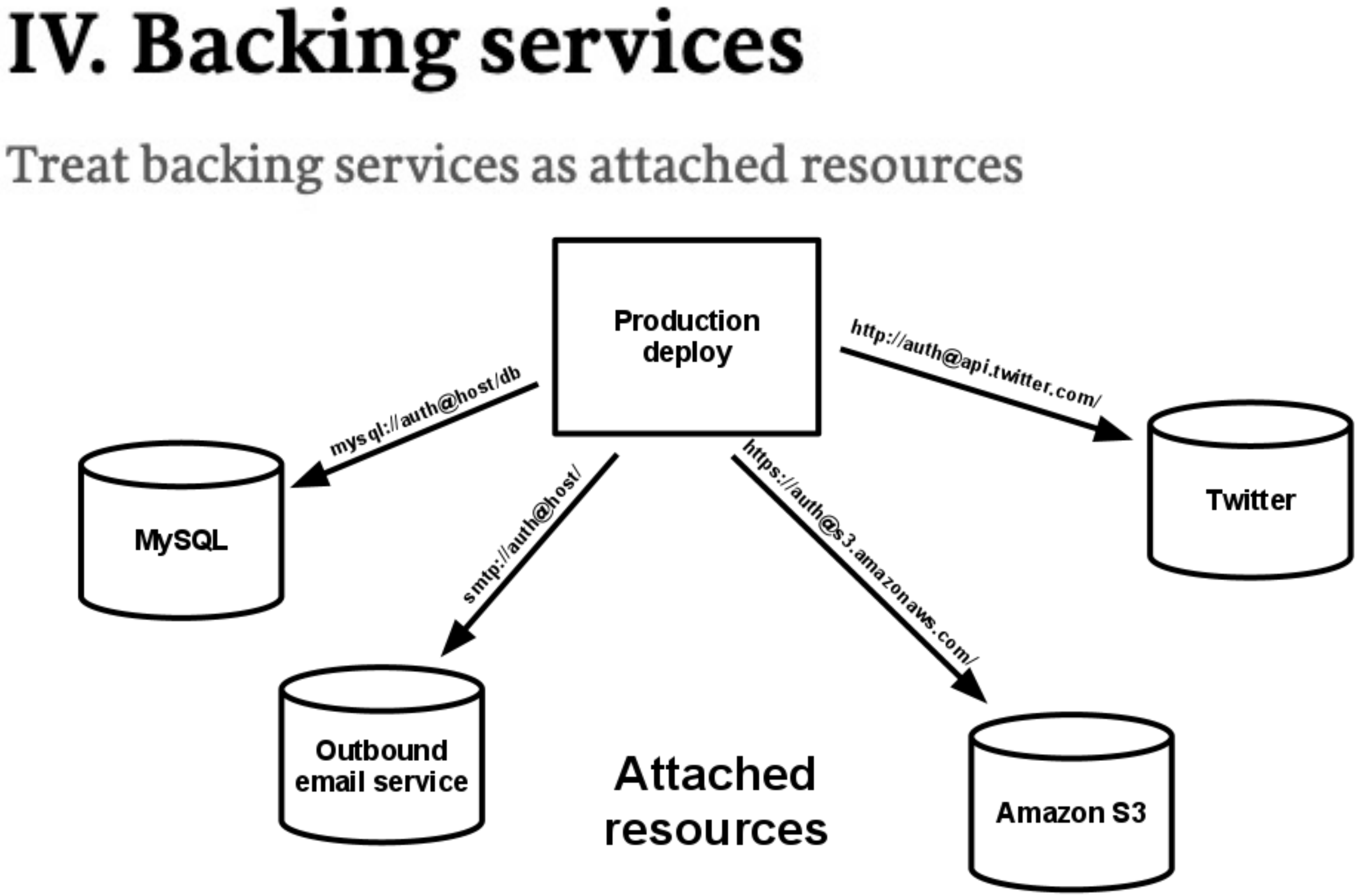
X. Dev/prod parity
Keep development, staging, and production as similar as possible

XI. Logs
Treat logs as event streams

XII. Admin processes
Run admin/management tasks as one-off processes

12 factor apps

https://12factor.net/

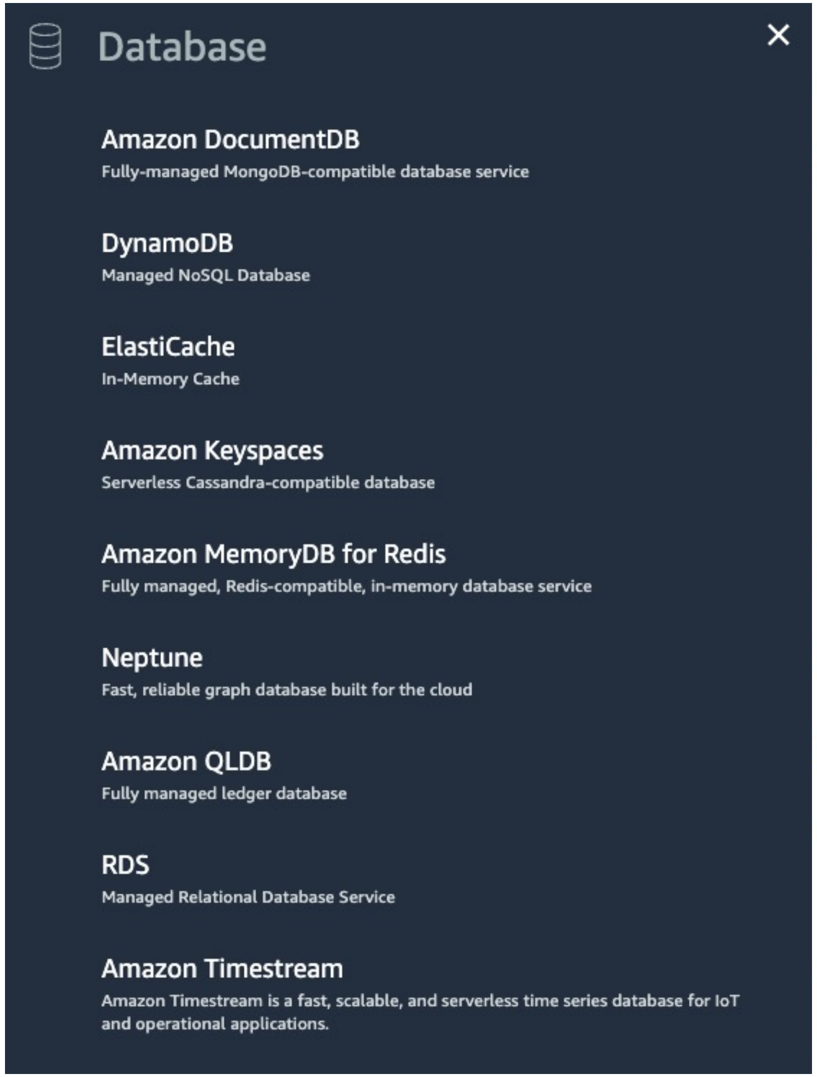


IV. Backing services

Treat backing services as attached resources

- That is cheating !
- It makes resiliency of stateful apps someone else's problem
- "Easy" on cloud thanks to many services available
- What about on-prem ?
  - What DBaas is there ?
  - Do they offer Sites resilience capabilities ?

Cloud DBaas example



Database

Amazon DocumentDB
Fully-managed MongoDB-compatible database service

DynamoDB
Managed NoSQL Database

ElastiCache
In-Memory Cache

Amazon Keyspaces
Serverless Cassandra-compatible database

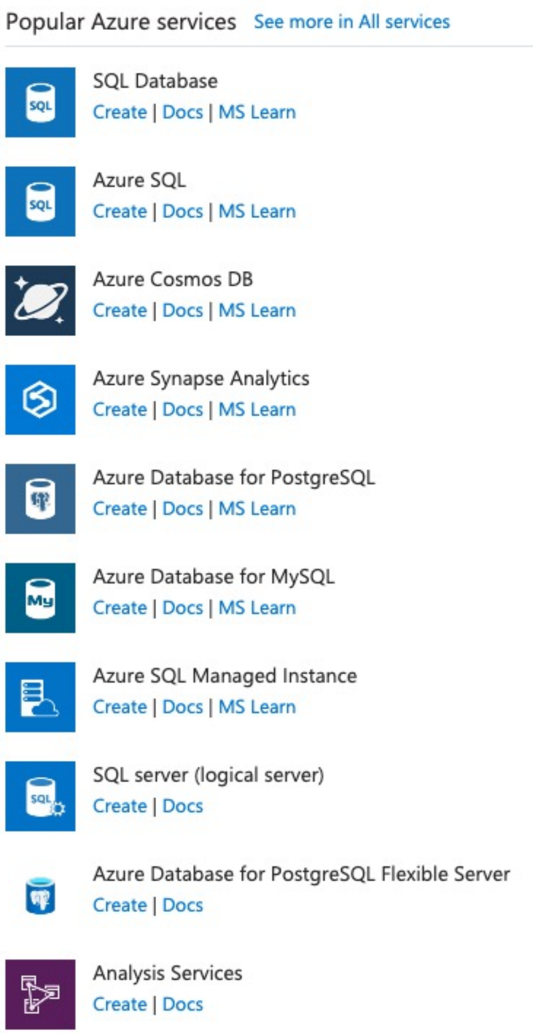
Amazon MemoryDB for Redis
Fully managed, Redis-compatible, in-memory database service

Neptune
Fast, reliable graph database built for the cloud

Amazon QLDB
Fully managed ledger database

RDS
Managed Relational Database Service

Amazon Timestream
Amazon Timestream is a fast, scalable, and serverless time series database for IoT and operational applications.



Popular Azure services See more in All services

SQL Database
Create | Docs | MS Learn

Azure SQL
Create | Docs | MS Learn

Azure Cosmos DB
Create | Docs | MS Learn

Azure Synapse Analytics
Create | Docs | MS Learn

Azure Database for PostgreSQL
Create | Docs | MS Learn

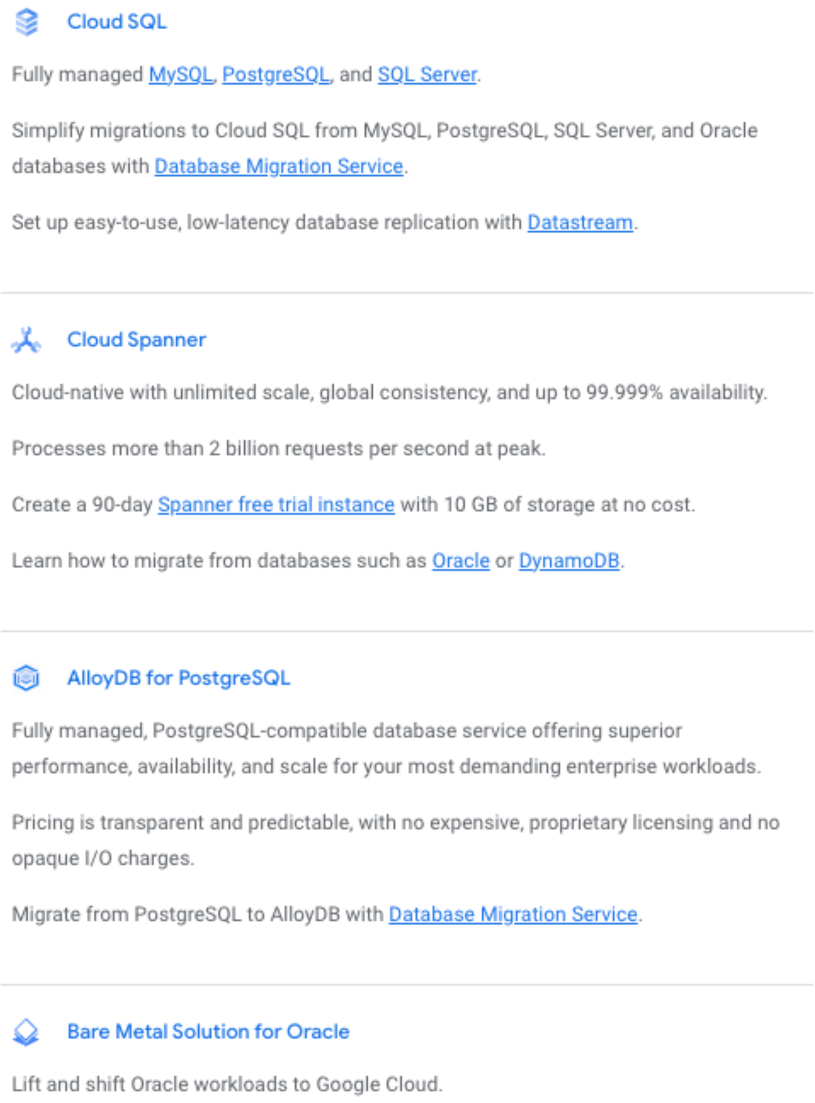
Azure Database for MySQL
Create | Docs | MS Learn

Azure SQL Managed Instance
Create | Docs | MS Learn

SQL server (logical server)
Create | Docs

Azure Database for PostgreSQL Flexible Server
Create | Docs

Analysis Services
Create | Docs



Cloud SQL

Fully managed MySQL, PostgreSQL, and SQL Server.

Simplify migrations to Cloud SQL from MySQL, PostgreSQL, SQL Server, and Oracle databases with Database Migration Service.

Set up easy-to-use, low-latency database replication with Datastream.

Cloud Spanner

Cloud-native with unlimited scale, global consistency, and up to 99.999% availability.

Processes more than 2 billion requests per second at peak.

Create a 90-day Spanner free trial instance with 10 GB of storage at no cost.

Learn how to migrate from databases such as Oracle or DynamoDB.

AlloyDB for PostgreSQL

Fully managed, PostgreSQL-compatible database service offering superior performance, availability, and scale for your most demanding enterprise workloads.

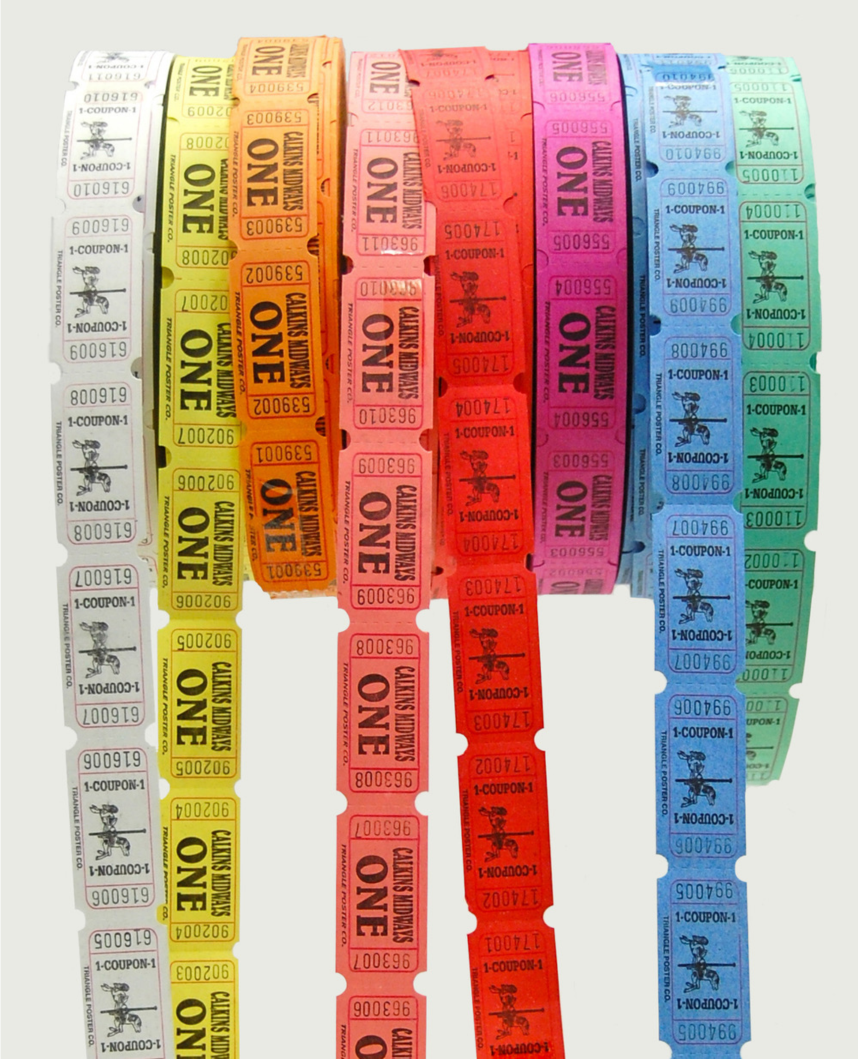
Pricing is transparent and predictable, with no expensive, proprietary licensing and no opaque I/O charges.

Migrate from PostgreSQL to AlloyDB with Database Migration Service.

Bare Metal Solution for Oracle

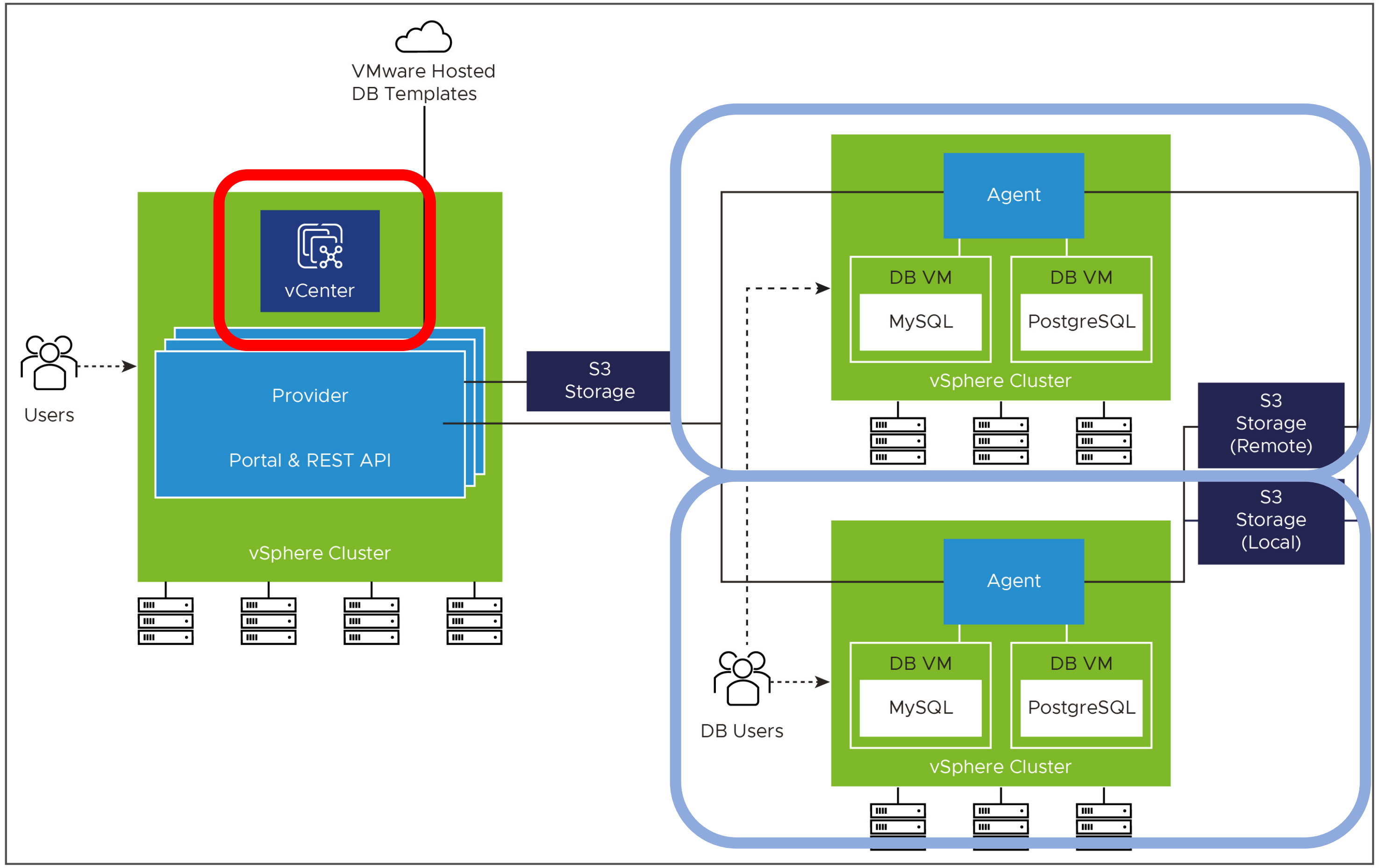
Lift and shift Oracle workloads to Google Cloud.

Typical On-Prem DBaas



DBaas – VM based - VMware Data Services Manager

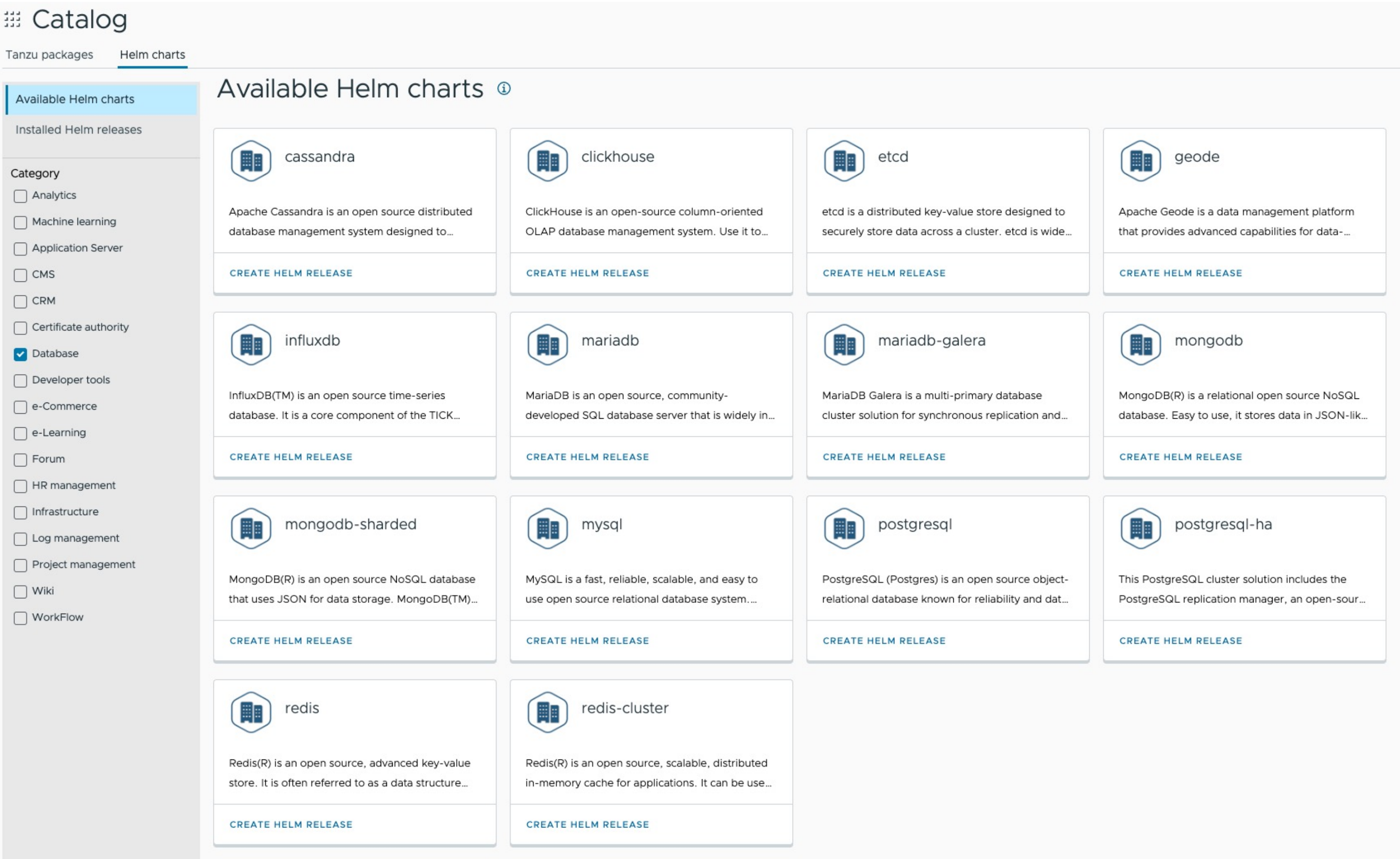
https://www.vmware.com/products/data-services-manager.html



- Can handle cross-cluster replication setup
- All clusters must be managed by same vCenter
- Data services offered
  - MySQL (v8.0.23 – v8.0.32)
  - PostgreSQL (11.19 – 15.2)
  - MS SQL 2019 (std, dev, ent editions)
  - https://docs.vmware.com/en/VMware-Data-Services-Manager/1.5/data-services-manager/GUID-release_notes.html

https://cormachogan.com/dsm/

DBaas – k8S based – Bitnami / VMware Application Catalogue



Catalog

Tanzu packages | Helm charts

Available Helm charts

Installed Helm releases

Category

- ☐ Analytics
- ☐ Machine learning
- ☐ Application Server
- ☐ CMS
- ☐ CRM
- ☐ Certificate authority
- ☑ Database
- ☐ Developer tools
- ☐ e-Commerce
- ☐ e-Learning
- ☐ Forum
- ☐ HR management
- ☐ Infrastructure
- ☐ Log management
- ☐ Project management
- ☐ Wiki
- ☐ WorkFlow

Available Helm charts ⓘ

cassandra
Apache Cassandra is an open source distributed database management system designed to…
CREATE HELM RELEASE

clickhouse
ClickHouse is an open-source column-oriented OLAP database management system. Use it to…
CREATE HELM RELEASE

etcd
etcd is a distributed key-value store designed to securely store data across a cluster. etcd is wide…
CREATE HELM RELEASE

geode
Apache Geode is a data management platform that provides advanced capabilities for data-…
CREATE HELM RELEASE

influxdb
InfluxDB(TM) is an open source time-series database. It is a core component of the TICK…
CREATE HELM RELEASE

mariadb
MariaDB is an open source, community-developed SQL database server that is widely in…
CREATE HELM RELEASE

mariadb-galera
MariaDB Galera is a multi-primary database cluster solution for synchronous replication and…
CREATE HELM RELEASE

mongodb
MongoDB(R) is a relational open source NoSQL database. Easy to use, it stores data in JSON-lik…
CREATE HELM RELEASE

mongodb-sharded
MongoDB(R) is an open source NoSQL database that uses JSON for data storage. MongoDB(TM)…
CREATE HELM RELEASE

mysql
MySQL is a fast, reliable, scalable, and easy to use open source relational database system.…
CREATE HELM RELEASE

postgresql
PostgreSQL (Postgres) is an open source object-relational database known for reliability and dat…
CREATE HELM RELEASE

postgresql-ha
This PostgreSQL cluster solution includes the PostgreSQL replication manager, an open-sour…
CREATE HELM RELEASE

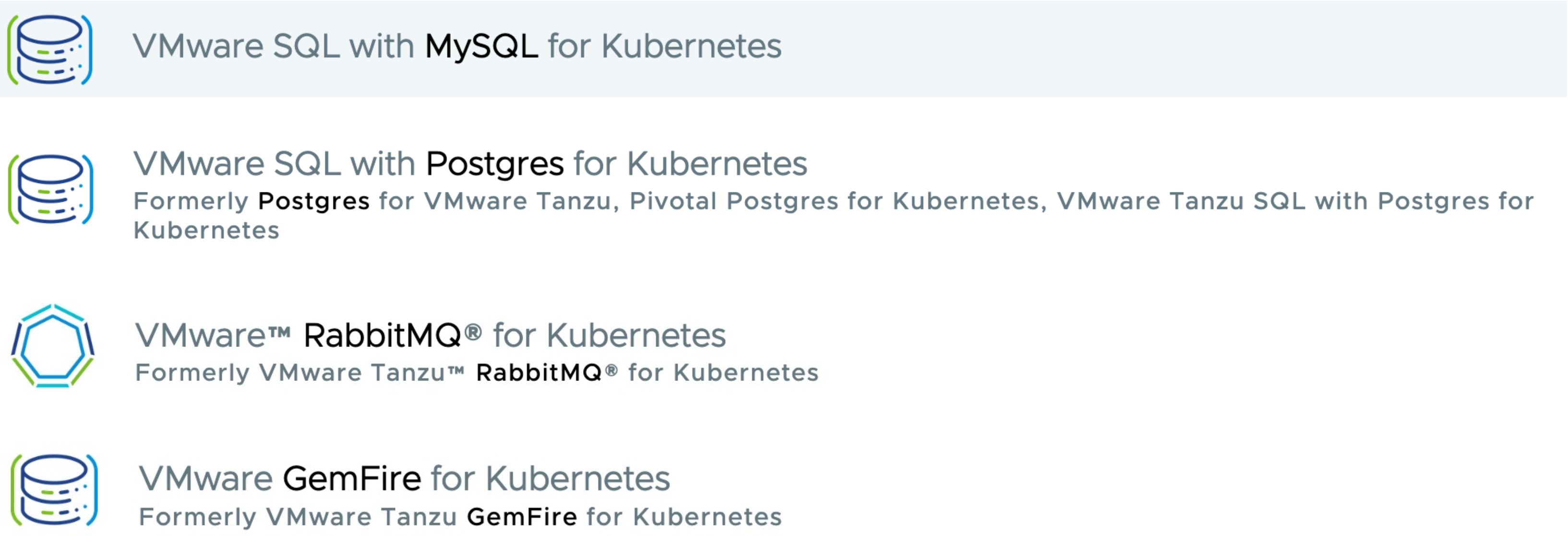
redis
Redis(R) is an open source, advanced key-value store. It is often referred to as a data structure…
CREATE HELM RELEASE

redis-cluster
Redis(R) is an open source, scalable, distributed in-memory cache for applications. It can be use…
CREATE HELM RELEASE

- More DB options
- Based on helm charts
- Helm chart does not span beyond the cluster

DBaas – K8s based - VMware Data operators

https://network.pivotal.io/



VMware SQL with **MySQL** for Kubernetes

VMware SQL with **Postgres** for Kubernetes
Formerly **Postgres** for VMware Tanzu, Pivotal Postgres for Kubernetes, VMware Tanzu SQL with Postgres for Kubernetes

VMware™ **RabbitMQ®** for Kubernetes
Formerly VMware Tanzu™ **RabbitMQ®** for Kubernetes

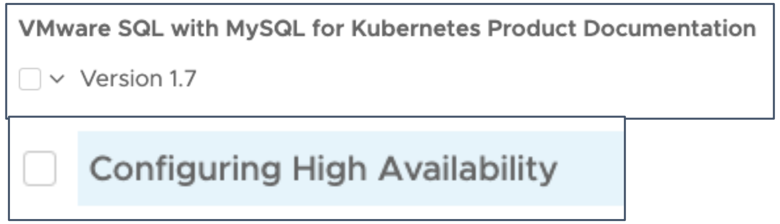
VMware **GemFire** for Kubernetes
Formerly VMware Tanzu **GemFire** for Kubernetes

- K8s based operators

- Data services offered
  - MySQL (v8.0.28 – v8.0.32)
  - PostgreSQL (11.21 – 15.4)
  - RabbitMQ (3.12.4)
  - GemFire (9.15-10.0)

DBaas – K8s based - VMware Data operators

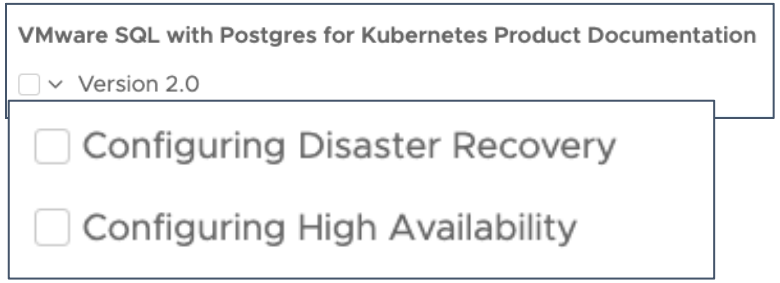
HA/DR/replication features



VMware SQL with MySQL for Kubernetes Product Documentation

□ ⌄ Version 1.7

□ Configuring High Availability

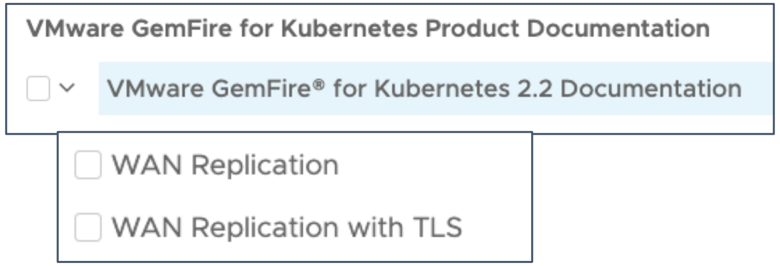


VMware SQL with Postgres for Kubernetes Product Documentation

□ ⌄ Version 2.0

□ Configuring Disaster Recovery

□ Configuring High Availability

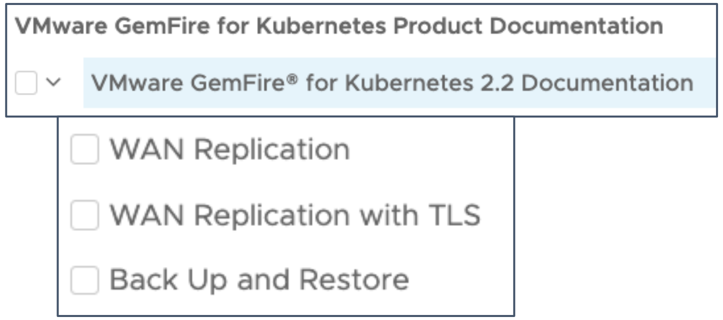


VMware GemFire for Kubernetes Product Documentation

□ ⌄ VMware GemFire® for Kubernetes 2.2 Documentation

□ WAN Replication

□ WAN Replication with TLS



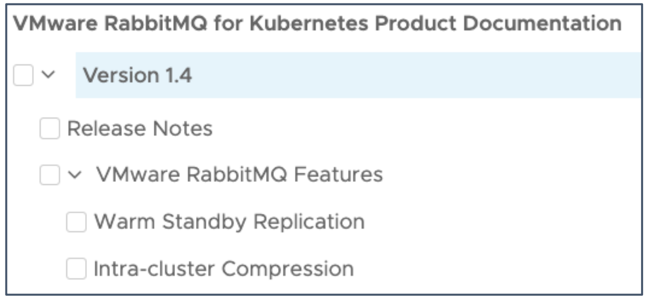
VMware GemFire for Kubernetes Product Documentation

□ ⌄ VMware GemFire® for Kubernetes 2.2 Documentation

□ WAN Replication

□ WAN Replication with TLS

□ Back Up and Restore



VMware RabbitMQ for Kubernetes Product Documentation

□ ⌄ Version 1.4

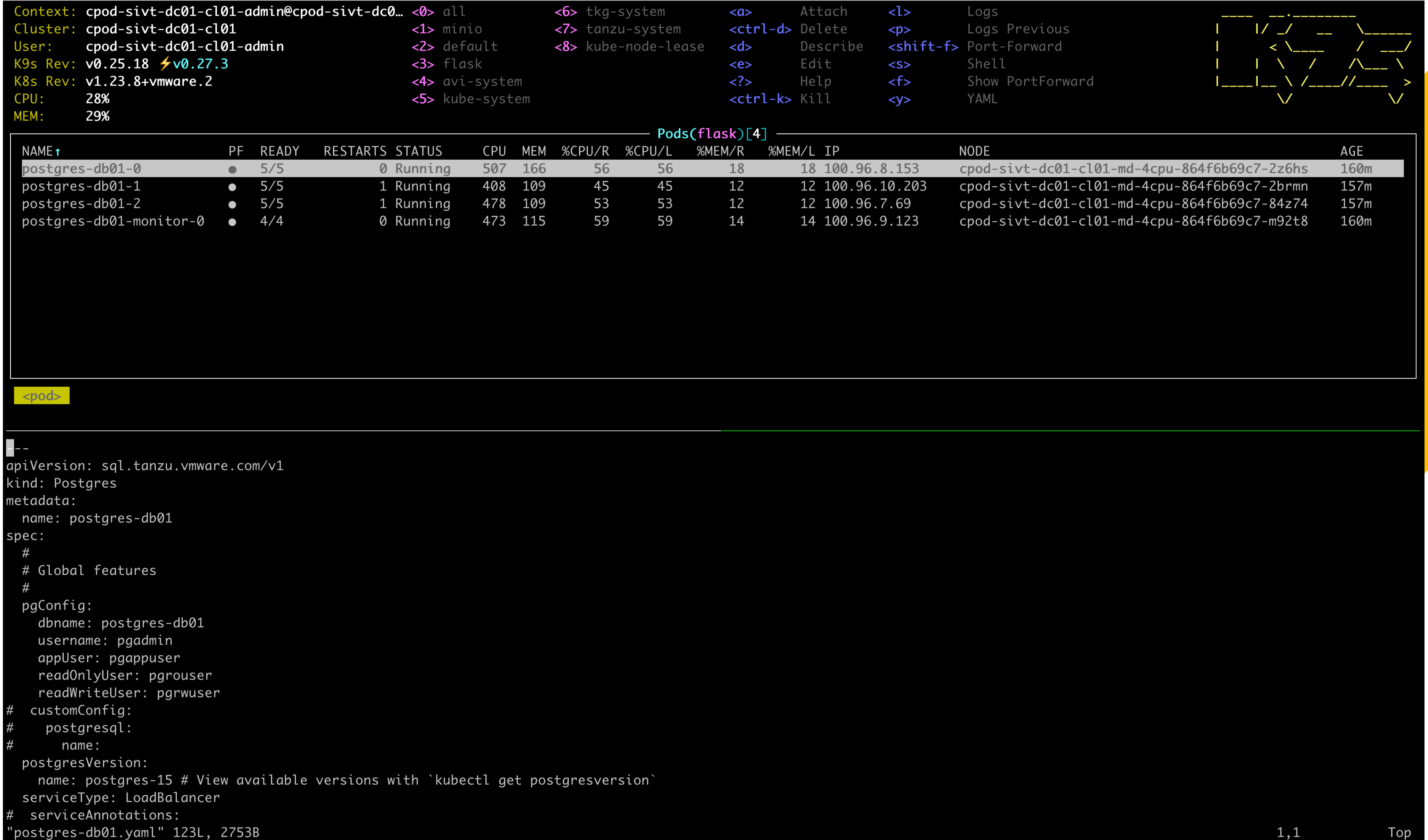
□ Release Notes

□ ⌄ VMware RabbitMQ Features

□ Warm Standby Replication

□ Intra-cluster Compression

- Different HA/DR capability per operator
- Operator HA/DR capability limited WITHIN the k8s cluster
- Object storage required in some cases

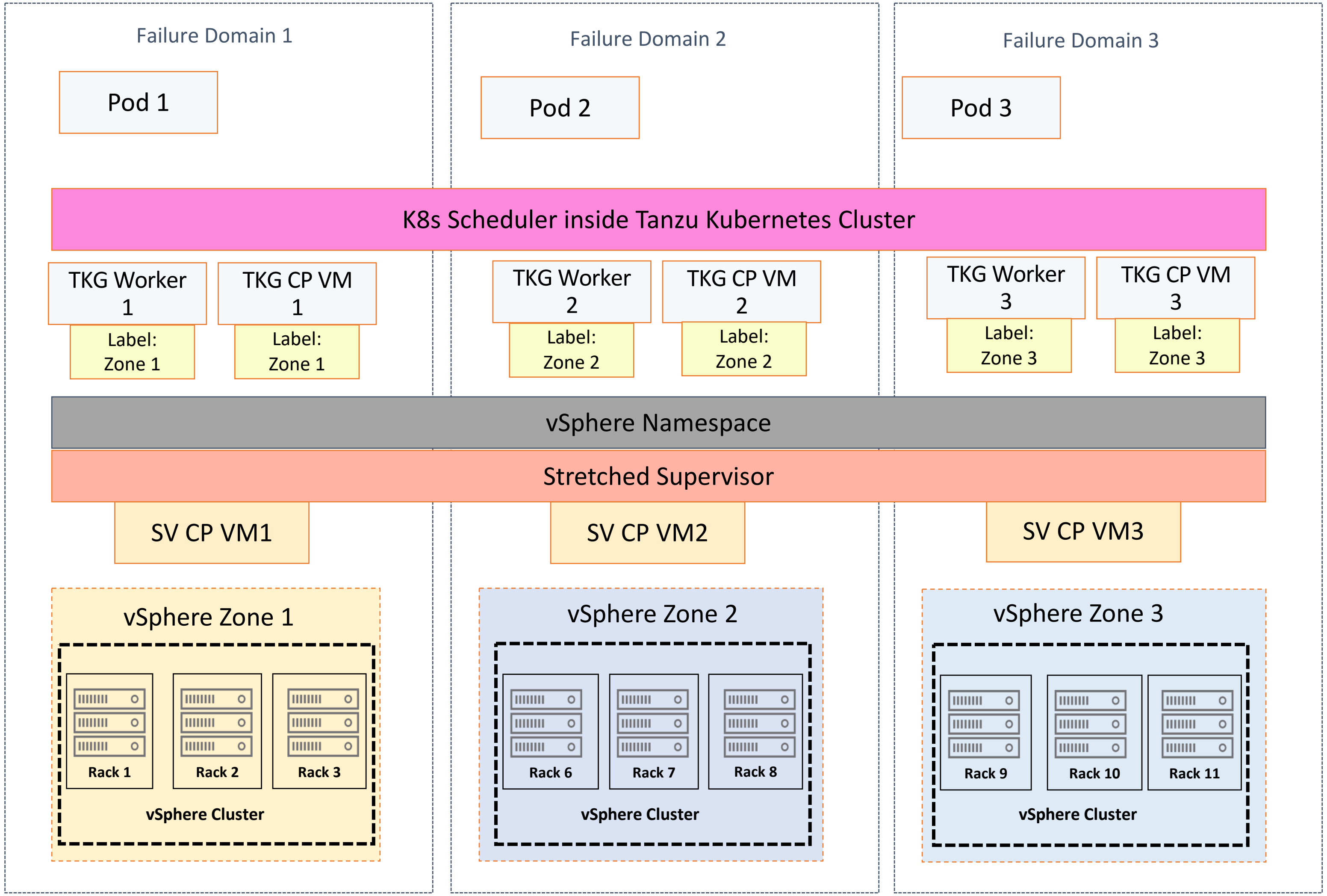


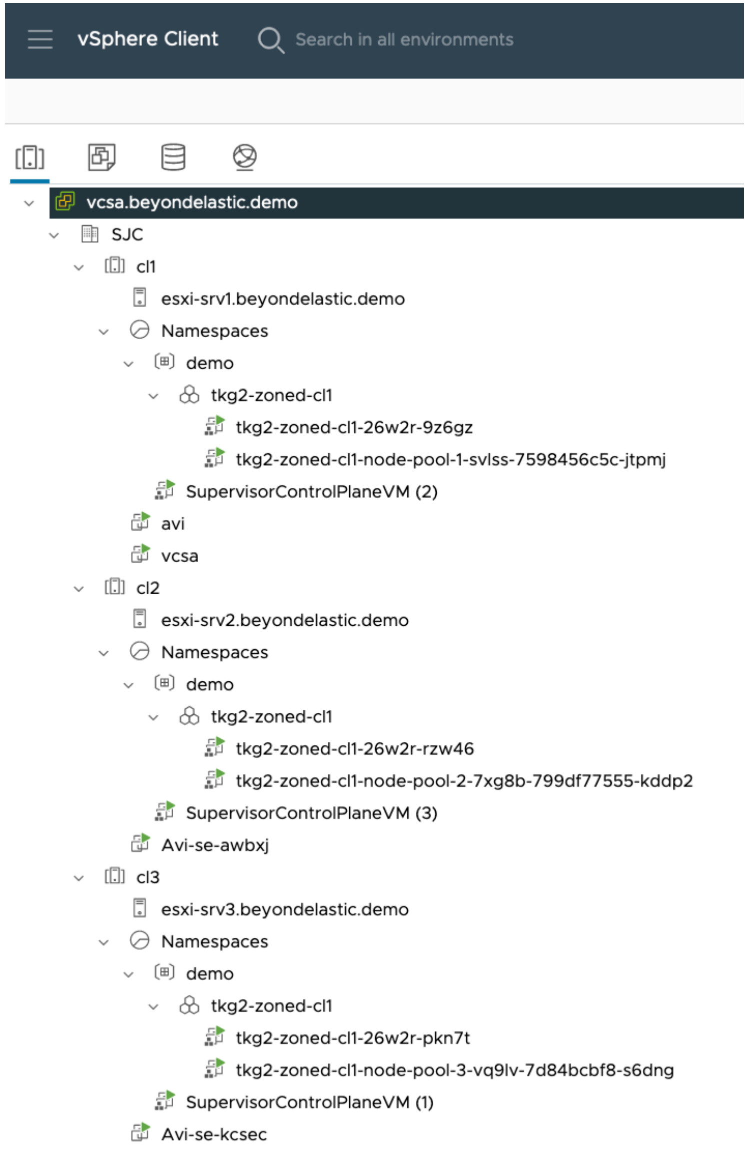
```
Context:  cpod-sivt-dc01-cl01-admin@cpod-sivt-dc0…  <0> all          <6> tkg-system        <a>       Attach     <l>         Logs                   ____  __.________
Cluster:  cpod-sivt-dc01-cl01                        <1> minio        <7> tanzu-system      <ctrl-d>  Delete     <p>         Logs Previous         |    |/ _/   __   \______
User:     cpod-sivt-dc01-cl01-admin                  <2> default      <8> kube-node-lease   <d>       Describe   <shift-f>   Port-Forward          |      < \____    /  ___/
K9s Rev:  v0.25.18 ⚡v0.27.3                          <3> flask                              <e>       Edit       <s>         Shell                 |    |  \   /    /\___ \
K8s Rev:  v1.23.8+vmware.2                           <4> avi-system                         <?>       Help       <f>         Show PortForward      |____|__ \ /____//____  >
CPU:      28%                                        <5> kube-system                        <ctrl-k>  Kill       <y>         YAML                          \/            \/
MEM:      29%
┌──────────────────────────────────────────────────────────────────── Pods(flask)[4] ─────────────────────────────────────────────────────────────────────┐
│ NAME↑                      PF  READY  RESTARTS STATUS     CPU  MEM  %CPU/R  %CPU/L  %MEM/R  %MEM/L IP             NODE                                         AGE  │
│ postgres-db01-0            ●   5/5           0 Running    507  166      56      56      18      18 100.96.8.153   cpod-sivt-dc01-cl01-md-4cpu-864f6b69c7-2z6hs 160m │
│ postgres-db01-1            ●   5/5           1 Running    408  109      45      45      12      12 100.96.10.203  cpod-sivt-dc01-cl01-md-4cpu-864f6b69c7-2brmn 157m │
│ postgres-db01-2            ●   5/5           1 Running    478  109      53      53      12      12 100.96.7.69    cpod-sivt-dc01-cl01-md-4cpu-864f6b69c7-84z74 157m │
│ postgres-db01-monitor-0    ●   4/4           0 Running    473  115      59      59      14      14 100.96.9.123   cpod-sivt-dc01-cl01-md-4cpu-864f6b69c7-m92t8 160m │
└─────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────┘
 <pod>
```

```
---
apiVersion: sql.tanzu.vmware.com/v1
kind: Postgres
metadata:
  name: postgres-db01
spec:
  #
  # Global features
  #
  pgConfig:
    dbname: postgres-db01
    username: pgadmin
    appUser: pgappuser
    readOnlyUser: pgrouser
    readWriteUser: pgrwuser
#  customConfig:
#    postgresql:
#      name:
  postgresVersion:
    name: postgres-15 # View available versions with `kubectl get postgresversion`
  serviceType: LoadBalancer
#  serviceAnnotations:
"postgres-db01.yaml" 123L, 2753B                                                                                                                1,1           Top
```

vSphere with Tanzu – Multi AZ

View from 30.000 ft

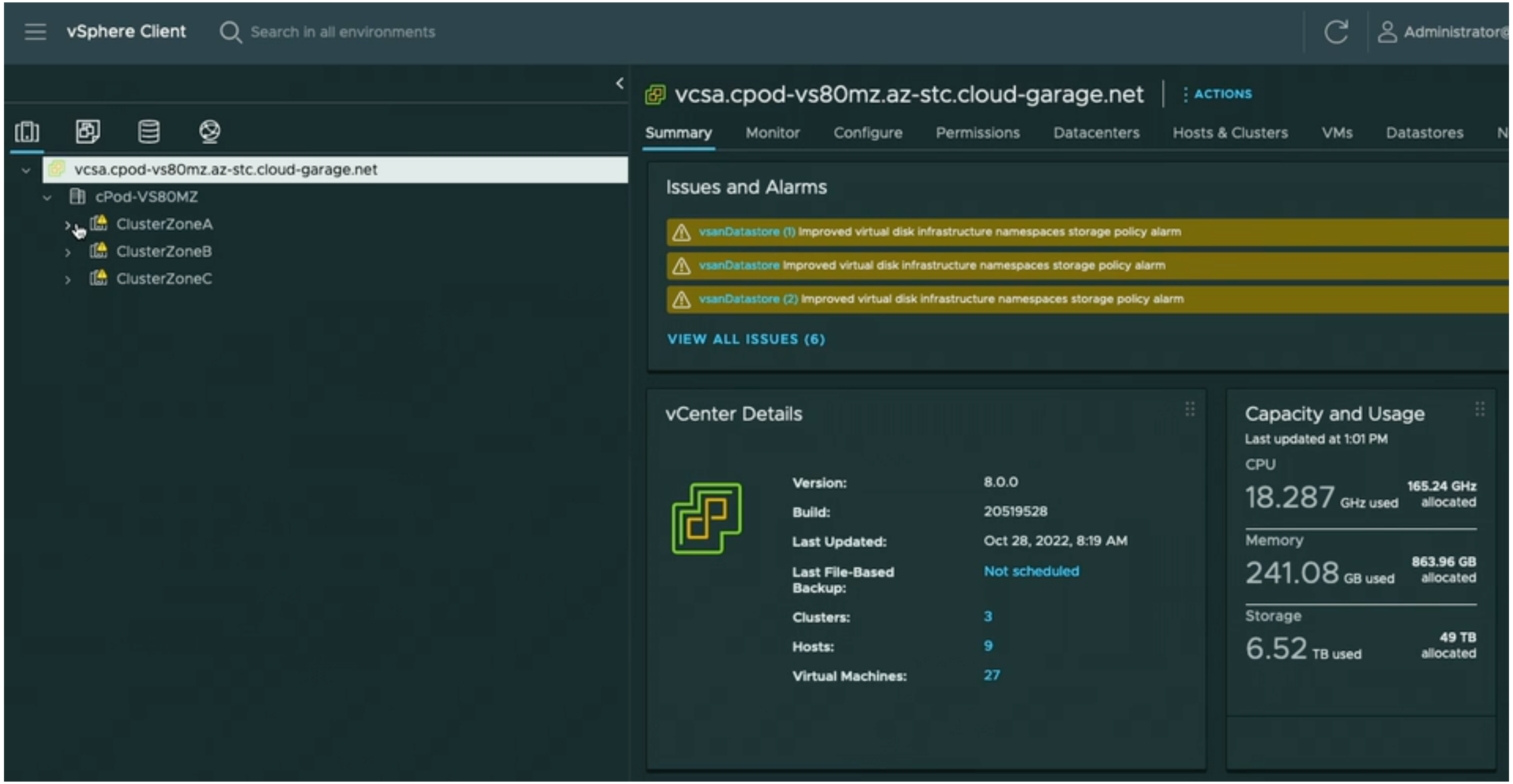




- vcsa.beyondelastic.demo
  - SJC
    - cl1
      - esxi-srv1.beyondelastic.demo
      - Namespaces
        - demo
          - tkg2-zoned-cl1
            - tkg2-zoned-cl1-26w2r-9z6gz
            - tkg2-zoned-cl1-node-pool-1-svlss-7598456c5c-jtpmj
        - SupervisorControlPlaneVM (2)
      - avi
      - vcsa
    - cl2
      - esxi-srv2.beyondelastic.demo
      - Namespaces
        - demo
          - tkg2-zoned-cl1
            - tkg2-zoned-cl1-26w2r-rzw46
            - tkg2-zoned-cl1-node-pool-2-7xg8b-799df77555-kddp2
        - SupervisorControlPlaneVM (3)
      - Avi-se-awbxj
    - cl3
      - esxi-srv3.beyondelastic.demo
      - Namespaces
        - demo
          - tkg2-zoned-cl1
            - tkg2-zoned-cl1-26w2r-pkn7t
            - tkg2-zoned-cl1-node-pool-3-vq9lv-7d84bcbf8-s6dng
        - SupervisorControlPlaneVM (1)
      - Avi-se-kcsec

Demo VMworld 2022 : TKO and TKG new features

https://www.vmware.com/explore/video-library/video-landing.html?sessionid=1655951651150001wqGI&videoId=6315208341112



vcsa.cpod-vs80mz.az-stc.cloud-garage.net

Issues and Alarms

- ⚠ vsanDatastore (1) Improved virtual disk infrastructure namespaces storage policy alarm
- ⚠ vsanDatastore Improved virtual disk infrastructure namespaces storage policy alarm
- ⚠ vsanDatastore (2) Improved virtual disk infrastructure namespaces storage policy alarm

VIEW ALL ISSUES (6)

vCenter Details

Version:	8.0.0
Build:	20519528
Last Updated:	Oct 28, 2022, 8:19 AM
Last File-Based Backup:	Not scheduled
Clusters:	3
Hosts:	9
Virtual Machines:	27

Capacity and Usage

Last updated at 1:01 PM

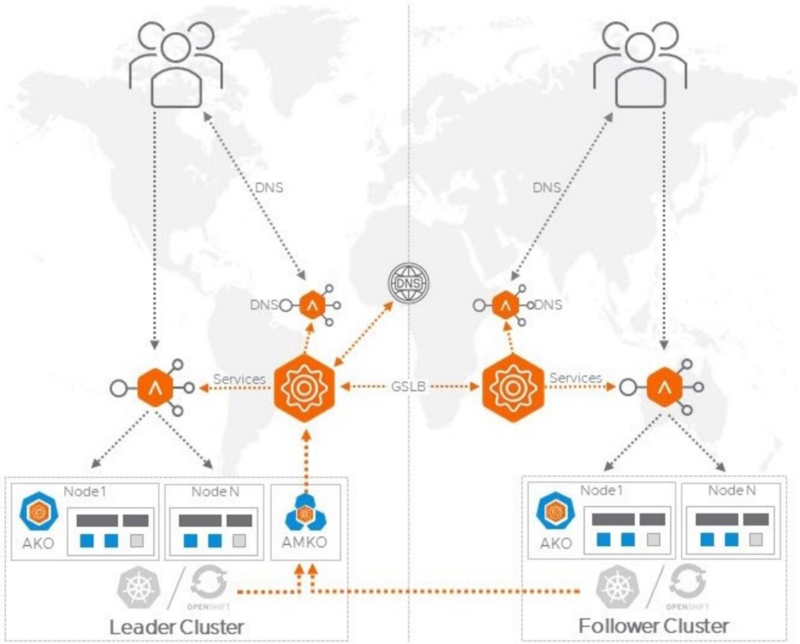
CPU
18.287 GHz used — 165.24 GHz allocated

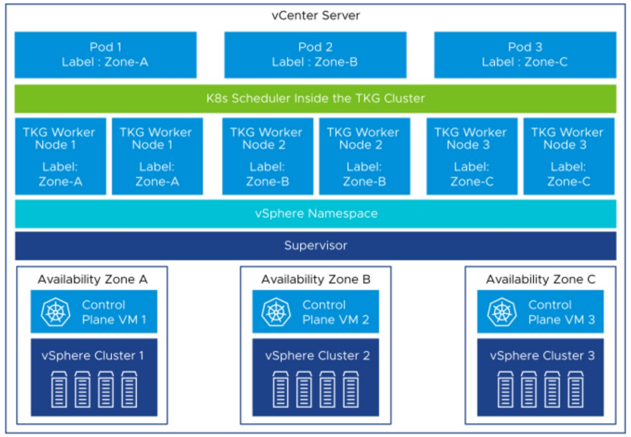
Memory
241.08 GB used — 863.96 GB allocated

Storage
6.52 TB used — 49 TB allocated

Multi-AZ demo part : starting at 31:17

Combining architectures for the most demanding customers





vCenter Server

Pod 1
Label : Zone-A

Pod 2
Label : Zone-B

Pod 3
Label : Zone-C

K8s Scheduler Inside the TKG Cluster

TKG Worker Node 1
Label: Zone-A

TKG Worker Node 1
Label: Zone-A

TKG Worker Node 2
Label: Zone-B

TKG Worker Node 2
Label: Zone-B

TKG Worker Node 3
Label: Zone-C

TKG Worker Node 3
Label: Zone-C

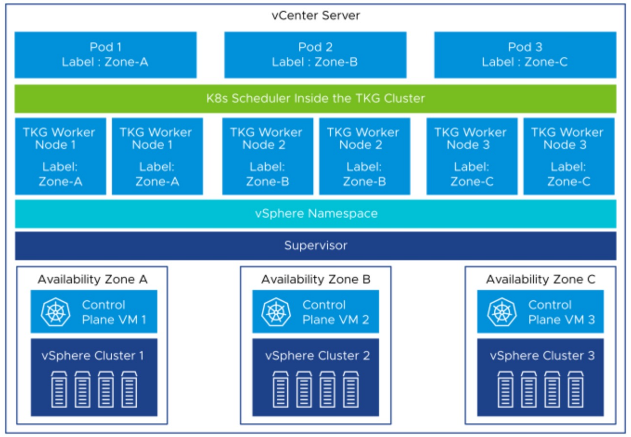
vSphere Namespace

Supervisor

Availability Zone A
Control Plane VM 1
vSphere Cluster 1

Availability Zone B
Control Plane VM 2
vSphere Cluster 2

Availability Zone C
Control Plane VM 3
vSphere Cluster 3



vCenter Server

Pod 1
Label : Zone-A

Pod 2
Label : Zone-B

Pod 3
Label : Zone-C

K8s Scheduler Inside the TKG Cluster

TKG Worker Node 1
Label: Zone-A

TKG Worker Node 1
Label: Zone-A

TKG Worker Node 2
Label: Zone-B

TKG Worker Node 2
Label: Zone-B

TKG Worker Node 3
Label: Zone-C

TKG Worker Node 3
Label: Zone-C

vSphere Namespace

Supervisor

Availability Zone A
Control Plane VM 1
vSphere Cluster 1

Availability Zone B
Control Plane VM 2
vSphere Cluster 2

Availability Zone C
Control Plane VM 3
vSphere Cluster 3

Retro

- Stick to stateless on k8s as much as possible
  - Learn to walk before running
  - Statefull apps on VMs (customers know how to do that)
  - Portfolio approach to statefull : Aria automation for DBaas ?
- Application based resiliency requires to know the application
  - Rewriting it can take time and be challenging
  - If you build a platform without knowing the application needs, you’re in for a bumpy ride for the platform success => find the APP !
- Not ALL solutions will come from VMware
  - Ecosystem
  - 3rd party operators for specific needs (redis, cassandra,…)
  - 3rd party to provide some form of infrastructure solution (portworx, …)

Retro - 2

- What about CI/CD ?
  - Where do you run you CD pipeline ?
  - How do you protect the CD platform ?
  - How do you manage pushing to multiple k8s clusters consistently ?

- Don’t forget to backup your k8S clusters (if using stateful services)
  - Cf Project Velero : https://velero.io/

ENTERPRISE
DEVOPS
TECHCON