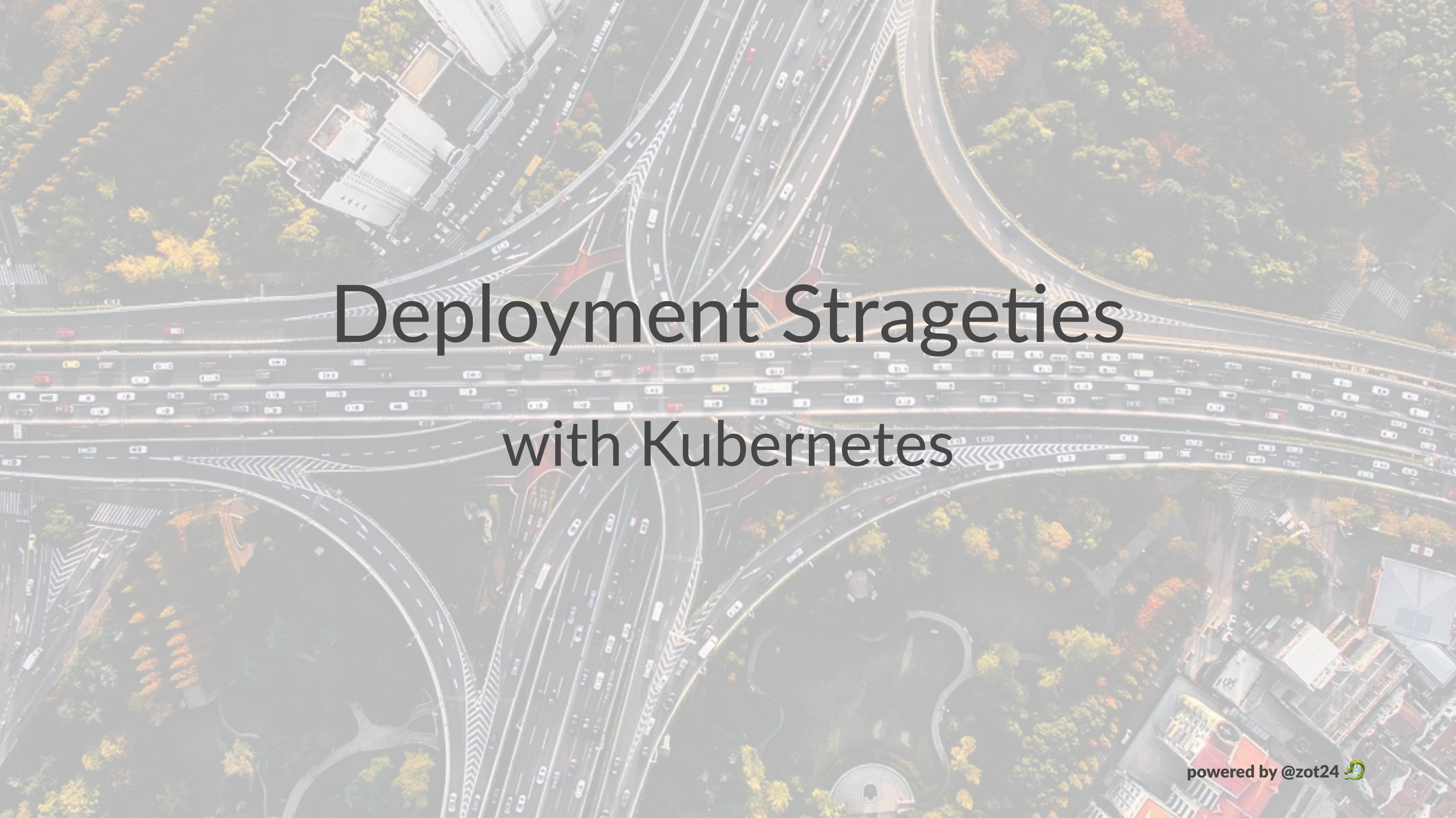


Deployment Strageties

with Kubernetes

powered by @zot24

What will you learn today? 

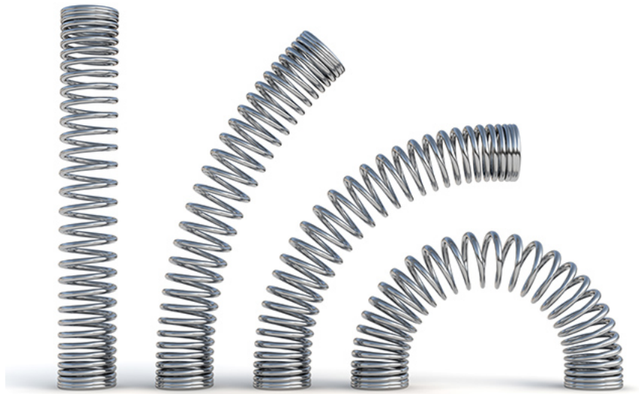
You'll be able to understand the **different choices** we have when it comes to **operate** our applications *deployments*.

Summary 🗓️

- 🏷️ Key concepts
  - 🤾 **Resiliency & Reliability**
  - 📡 Liveness & Readiness
  - 🚀 Deploy != Release
- 👾 Deployment strategies
- 🔑 Take Away

Resiliency 

The ability of an app to recover from certain types of failure and yet remain functional from the customer perspective.



What does it means that an application is **reliabe**?

It *operates perfectly* all the time 👌

but … how?

Resiliency

Summary 🗓️

- 🏷️ Key concepts
  - 🤾 Resiliency & Reliability
  - 📡 **Liveness & Readiness**
  - 🚀 Deploy != Release
- 👾 Deployment strategies
- 🔑 Take Away

What is *liveness* & *readiness* about?

Exposure 🌞 and probe 🔬

Be able to know what the state of an *application* is, so that the ecosystem where **it lives** and which **manages** it can be able to do it's job better.

Example[1] 📡

```
// Our app is not happy if we've got more than 100 goroutines running.
health.AddLivenessCheck("goroutine-threshold",
    healthcheck.GoroutineCountCheck(100))

// Our app is not ready if we can't resolve our upstream dependency in DNS.
health.AddReadinessCheck("upstream-dep-dns",
    healthcheck.DNSResolveCheck("upstream.example.com", 50*time.Millisecond))

// Our app is not ready if we can't connect to our database (`var db *sql.DB`) in <1s.
health.AddReadinessCheck("database",
    healthcheck.DatabasePingCheck(db, 1*time.Second))
```

---

[1] https://github.com/heptiolabs/healthcheck

Summary 🗓️

- 🏷️ Key concepts
  - 🤾 Resiliency & Reliability
  - 📡 Liveness & Readiness
  - 🚀 **Deploy != Release**
- 👾 Deployment strategies
- 🔑 Take Away

Deploy != Release 🚀

Deployment is what happens when you install some version of your software into a particular environment

Release is when you make a system or some part of it (for example, a feature) available to users

Summary 🗓️

- 🏷️ Key concepts
  - 🤾 Resiliency & Reliability
  - 📡 Liveness & Readiness
  - 🚀 Deploy != Release
- 👾 **Deployment strategies**
- 🔑 Take Away

Deployment strategies 

- Recreate
- Ramped
- Blue/Green
- Canary
- A/B testing
- Shadow

Summary 🗓️

- 🏷️ Key concepts
  - 🤾 Resiliency & Reliability
  - 📡 Liveness & Readiness
  - 🚀 Deploy != Release
- 👾 Deployment strategies
- 🔑 **Take Away**

All these *strategies*, will help us to ensure *reliability* in our systems...

When *deploying* first and *releasing* later, we can verify our application from a perspective that otherwise we wouldn't be able to...

Links 🔗

- **Traffic Shadowing and Dark Launching** by *Daniel Byrant*
  - http://bit.ly/2UOATS0
- **Four Priciples of Low-Risk Software Releases** by *Jez Humble*
  - http://bit.ly/2W9wdX6
- **Testing in Production, the safe way** by *Cindy Sridharan*
  - http://bit.ly/2HJJYrJ

Questions?

We can follow up on the subject any Thursday on the *Kubernetes Working Session*

Thank you 👏

🎙️🦜 https://github.com/zot24/talks