Avoid Single Point of Failure in Cloud-native Application

Phuong Le - Thang Chung

Hanoi - Dec 2018

VIETNAM WEB SUMMIT

NASH TECH

The Power to Innovate

$ whoami

- Phuong Le
  - Engineering Manager at NashTech
  - Over 15-years' experience in the full life-cycle of the software. I have passion on people development and help them grow in their career path

- Thang Chung
  - Solution Architect at NashTech
  - Over 10 years in software development industry, mainly focus on modern software architectures such as n-tiers, microservices, and cloud-native application.
  - https://github.com/thangchung
  - https://medium.com/@thangchung

- Key Takeaways
  - Cloud Native Fundamentals
  - Single Point of Failure in Cloud-native Apps
- Q&A

VIETNAM WEB SUMMIT

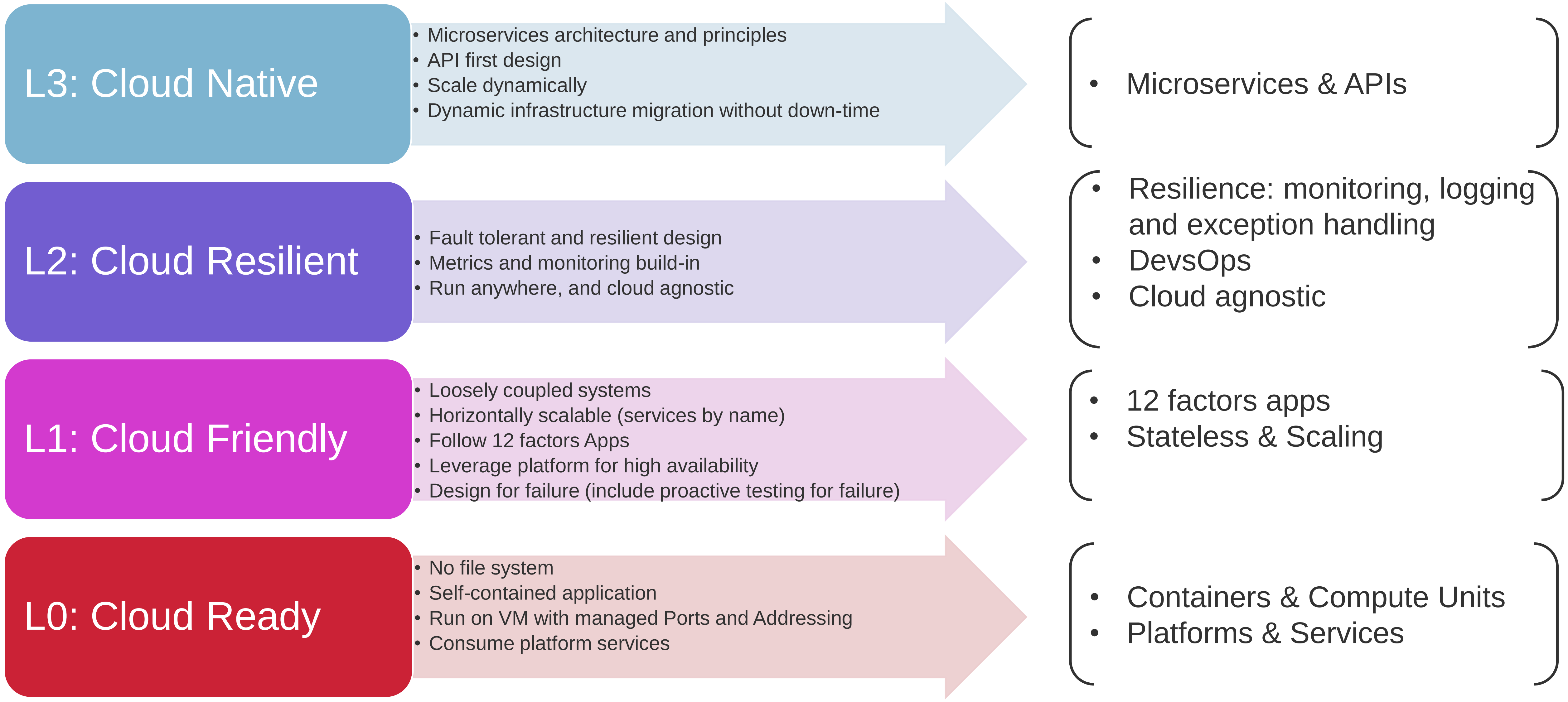
NASH TECH
The Power to Innovate

Cloud Native Fundamentals

“It is the first step that is troublesome” - proverb

Cloud-native Apps Maturity

VIETNAM WEB SUMMIT



L3: Cloud Native

- Microservices architecture and principles
- API first design
- Scale dynamically
- Dynamic infrastructure migration without down-time

- Microservices & APIs

L2: Cloud Resilient

- Fault tolerant and resilient design
- Metrics and monitoring build-in
- Run anywhere, and cloud agnostic

- Resilience: monitoring, logging and exception handling
- DevsOps
- Cloud agnostic

L1: Cloud Friendly

- Loosely coupled systems
- Horizontally scalable (services by name)
- Follow 12 factors Apps
- Leverage platform for high availability
- Design for failure (include proactive testing for failure)

- 12 factors apps
- Stateless & Scaling

L0: Cloud Ready

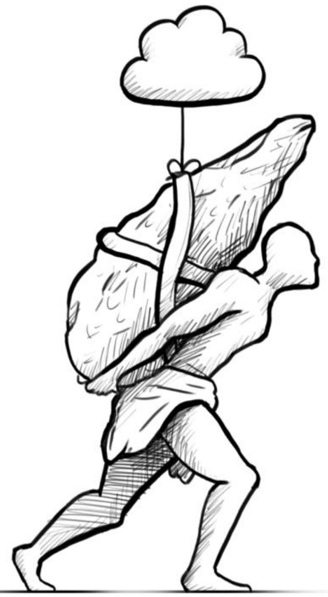
- No file system
- Self-contained application
- Run on VM with managed Ports and Addressing
- Consume platform services

- Containers & Compute Units
- Platforms & Services

Allan Beck & John McTeague (JPMorgan Chase & Co.)

Cloud Native Application Maturity (cont.)

- Monolithic Deployment
- Traditional Infrastructure



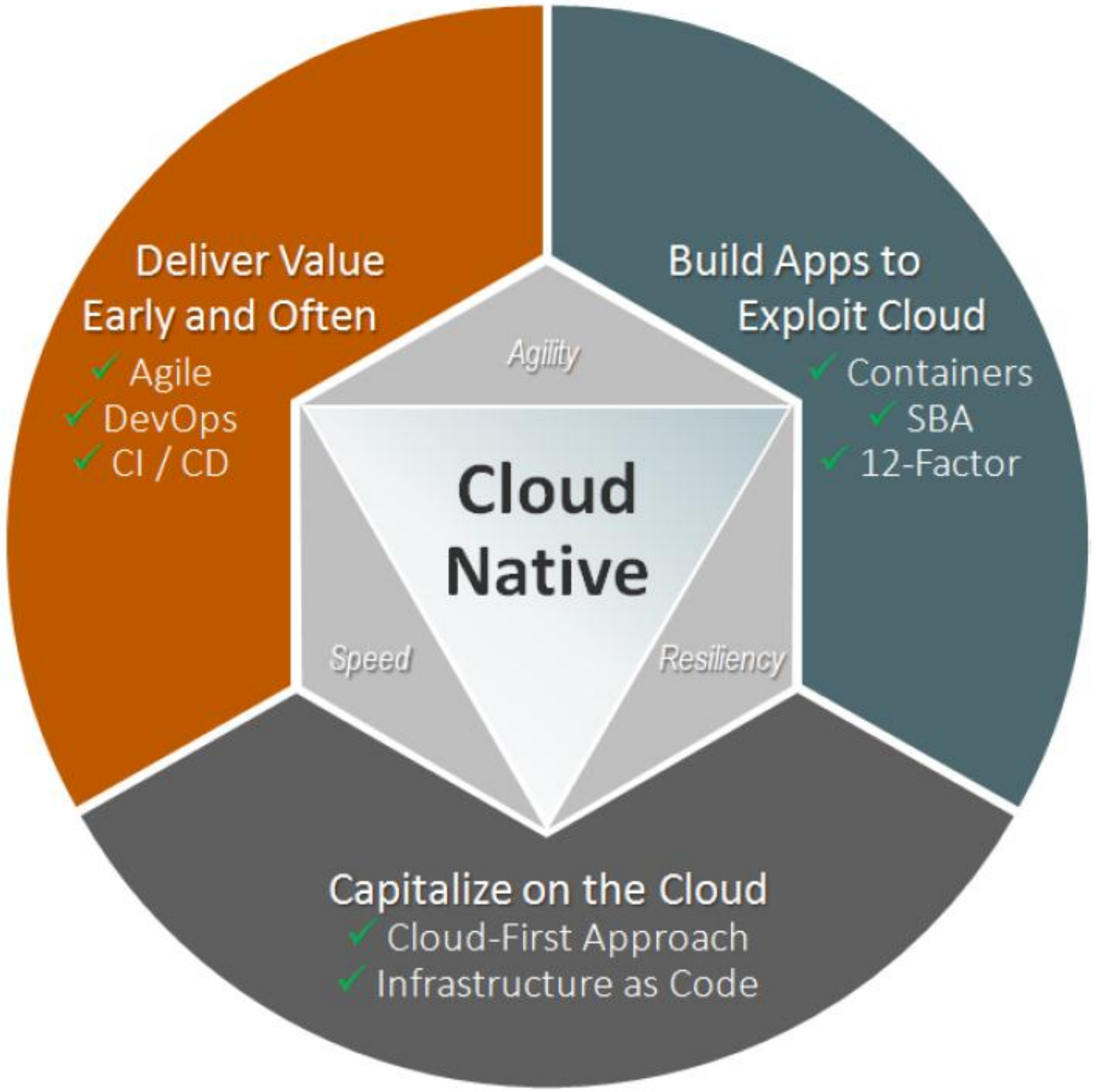
- Containerization
- 12-Factor App Principles

- Microservices
- Cloud-native Apps

Reference at **Steinzeit war gestern! Die vielfältigen Wege der Cloud-nativen Evolution**

Cloud Native Application Development

2017



Reference from Cloud Native Application development - A New Computing Paradigm by Oracle

Design Cloud-native Apps Factors

- **Factor 1**: One Code-base, One Application
- **Factor 2**: API first
- **Factor 3**: Dependency Management
- **Factor 4**: Design, Build, Release and Run
- **Factor 5**: Configuration, Credentials and Code
- **Factor 6**: Logs
- **Factor 7**: Disposability
- **Factor 8**: Backing Services
- **Factor 9**: Environment Parity
- **Factor 10**: Administrative Processes
- **Factor 11**: Port Binding
- **Factor 12**: Stateless Processes
- **Factor 13**: Concurrency
- **Factor 14**: Telemetry
- **Factor 15**: Authentication & Authorization

Single Point of Failure in Cloud-native Apps

“Every man is the architect of his own fortune” - proverb

Kubernetes High-Availability

K8s HA is not just about the stability of Kubernetes itself. It is about setting up Kubernetes, along with supporting components such as etcd, in such a way that there is no single point of failure

Lucas Käldström

VIETNAM WEB SUMMIT

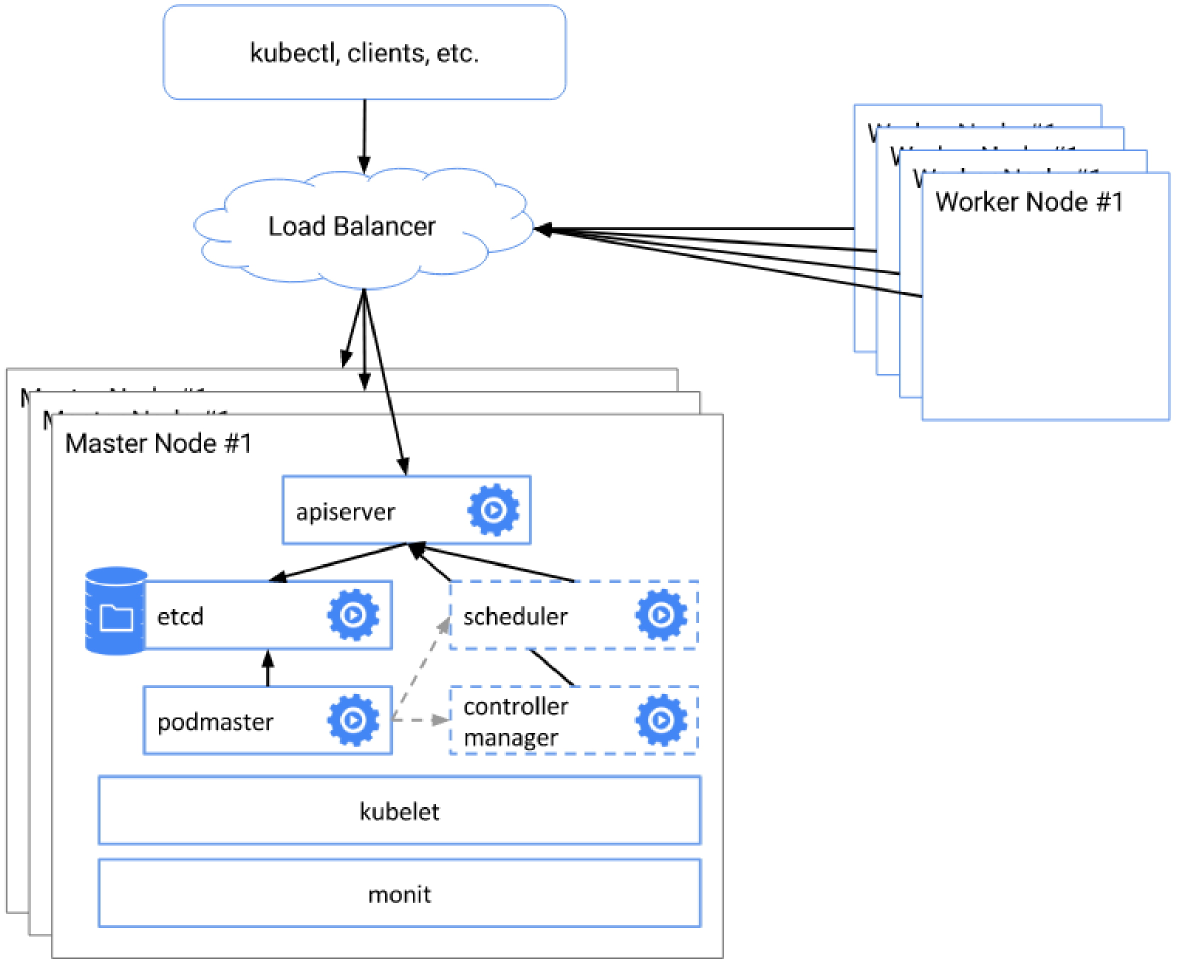
“Multi-master” vs “High-availability”

Factors can lead the single point of failure

- Load balancer to cluster
- Kube-dns
- Etcd
- A single master cluster

Achieve Kubernetes HA

- Setup Etcd cluster
- Setup load balancer
  - Keepalived
  - HA proxy
  - NginX
- Setup 3 master nodes
- Join worker nodes



THANK YOU

www.nashtechglobal.com

Q&A