



SUMMIT
BERLIN

Sponsored by...  CLOUDREACH

Containers in the Cloud: The **good**, the **bad**, and the **ugly**

Originally presented by Christian Nilsson at the AWS Summit in Berlin on February 27, 2019



“

A successful Cloud adoption is not just a technical endeavour, but one of significant organisational & cultural transformation. We specialise helping **large & complex** organisations succeed in that journey.

”

VOLKSWAGEN FINANCIAL SERVICES
THE KEY TO MOBILITY

STRÖER | digital media

Discovery
COMMUNICATIONS

Christian Nilsson - Head of Delivery, Cloud Adoption

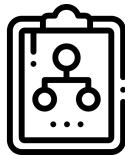
Applications

Data

Operating Systems
CI/CD Pipelines
Security
Governance
Automation
Networking
Patching
Cloud Provider



Drives Innovation and Revenue. Usually the competitive advantage.



Consumes time and resources. Contains well defined solutions but they are not market differentiators

WHY CONTAINERIZE - The Good

Simplified Dev & Deployment

Reduces “Works on my machine!” mentality

Couples application logic and its dependencies

One pipeline for all test

Unified tooling for logging and monitoring

Less responsibilities for developers

Simplified Security

Tracking of vulnerabilities across the business

Reduced packages and dependencies in production

Stronger isolation

Zero trust networking

Cost Efficiencies

Increased uptime

Less infrastructure to run the same number of applications

Reduced IT footprint

Culture

Increased Feature Cadence

Room for innovation & Curiosity

Increased Ownership

Operations Change

Customer Case Study

- Global Professional Services firm
- Launches new Applications weekly
- High degree of regulatory compliance
- Massive Scale

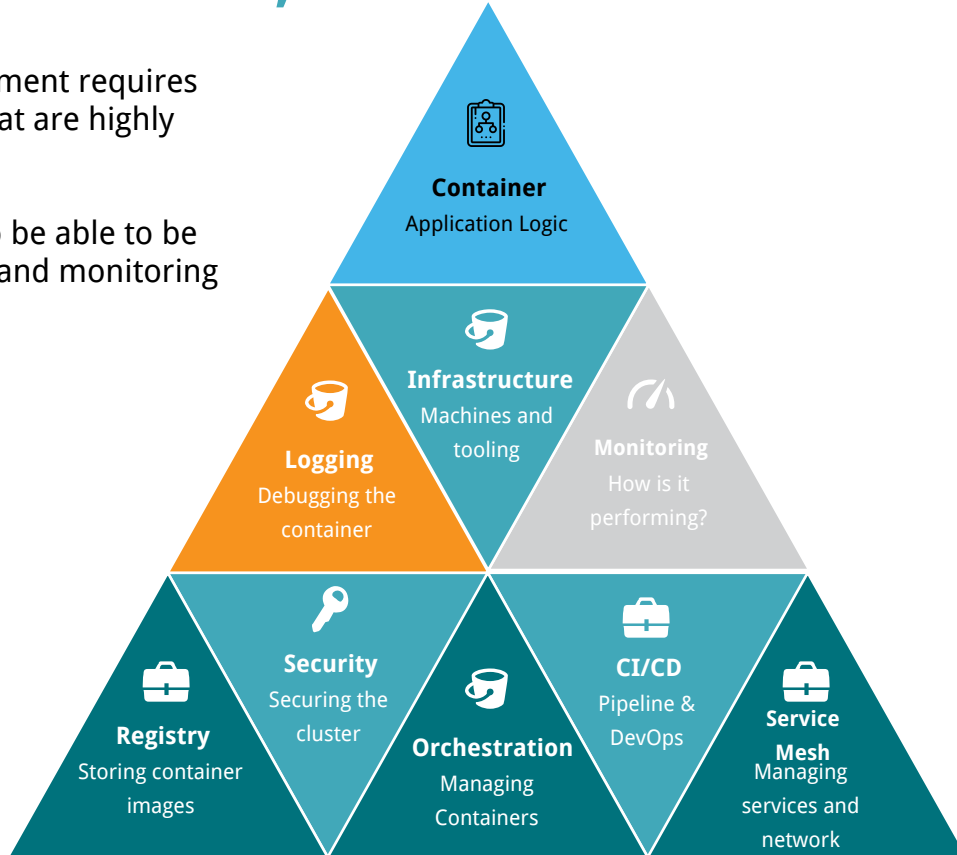
Wanted to Solve:

- Too many standards in applications, vendors and technologies
- Reduced TTM by reducing the infrastructural footprint in each new deployment
- Lower cost due to the same
- Ensure security by network isolation in multi-tenant environments
- No Managed Services (avoid lock-in)
- Multi-region workloads

Building Fault Tolerant, and Scalable Platforms

A successful container deployment requires several supporting services that are highly available.

Ephemeral containers need to be able to be pulled at any time, while logs and monitoring data is stored for analysis.

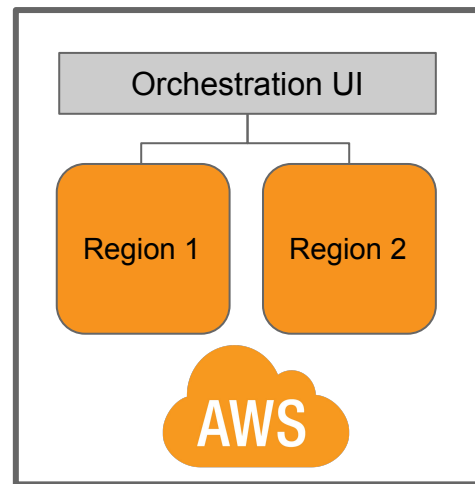


The solution



A custom container orchestration and networking service

■ Infrastructure	AWS
■ Infrastructure Build	KOPS/Terraform
■ Repository Registry	Code: Github Ent. Artefacts: Quay
■ Monitoring	Kubernetes Dashboard / InfluxDB + Grafana
■ Security Scanning	Nessus & Clair
■ Logging	ELK
■ Container Orchestration	Kubernetes & Canal + Calico SDN
■ CI/CD & DevOps Tooling	CircleCI (PR) & Jenkins
■ Service Mesh	Istio



Rolling your own Container Platform

The Good

- Completely abstracted from underlying technology or geography
- Everything is software defined and can be versioned
- Native support for CI/CD pipelines with testing
- Met all requirements of customers
- Changed the culture and way they deploy software

The Bad

- Complex environment to operate
- High run cost
- Fragile architecture
- Don't containerize COTS software

The Ugly

- Relies on many unsupported and obscure software libraries
- A lot of hacks and tweaks to glue it all together
- Development time of nine months for one CSP

The solution



A custom container orchestration and networking service

■ Infrastructure	AWS	AWS
■ Infrastructure Build	KOPS/Terraform	CloudFormation
■ Repository Registry	Code: Github Ent. Artefacts: Quay	Code: CodeCommit Artefacts: ECS
■ Monitoring	Kubernetes Dashboard / InfluxDB + Grafana	DataDog
■ Security Scanning	Nessus & Clair	TwistLock & Aqua
■ Logging	ELK	CloudTrail
■ Container Orchestration	Kubernetes & Canal + Calico SDN	EKS + Calico SDN
■ CI/CD & DevOps Tooling	CircleCI (PR) & Jenkins	CodeBuild + CodeDeploy
■ Service Mesh	Istio	App Mesh