Day 2 Operations

Robert Zahradníček

About me

- 5+ years doing DevOps
- Leaded DevOps teams in different environments
- Cloud-native and automation evangelist

Anyone can deploy an application but what's next?

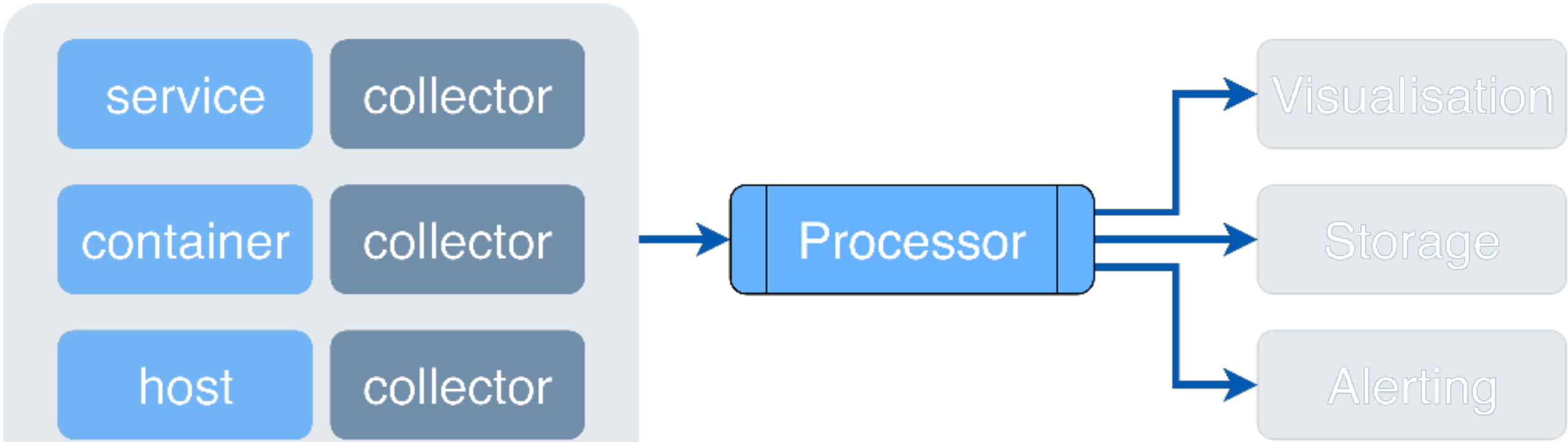
Once “something” goes into production, “day 2 operations” is the remaining time period until this “something” isn't killed or replaced with “something else.”

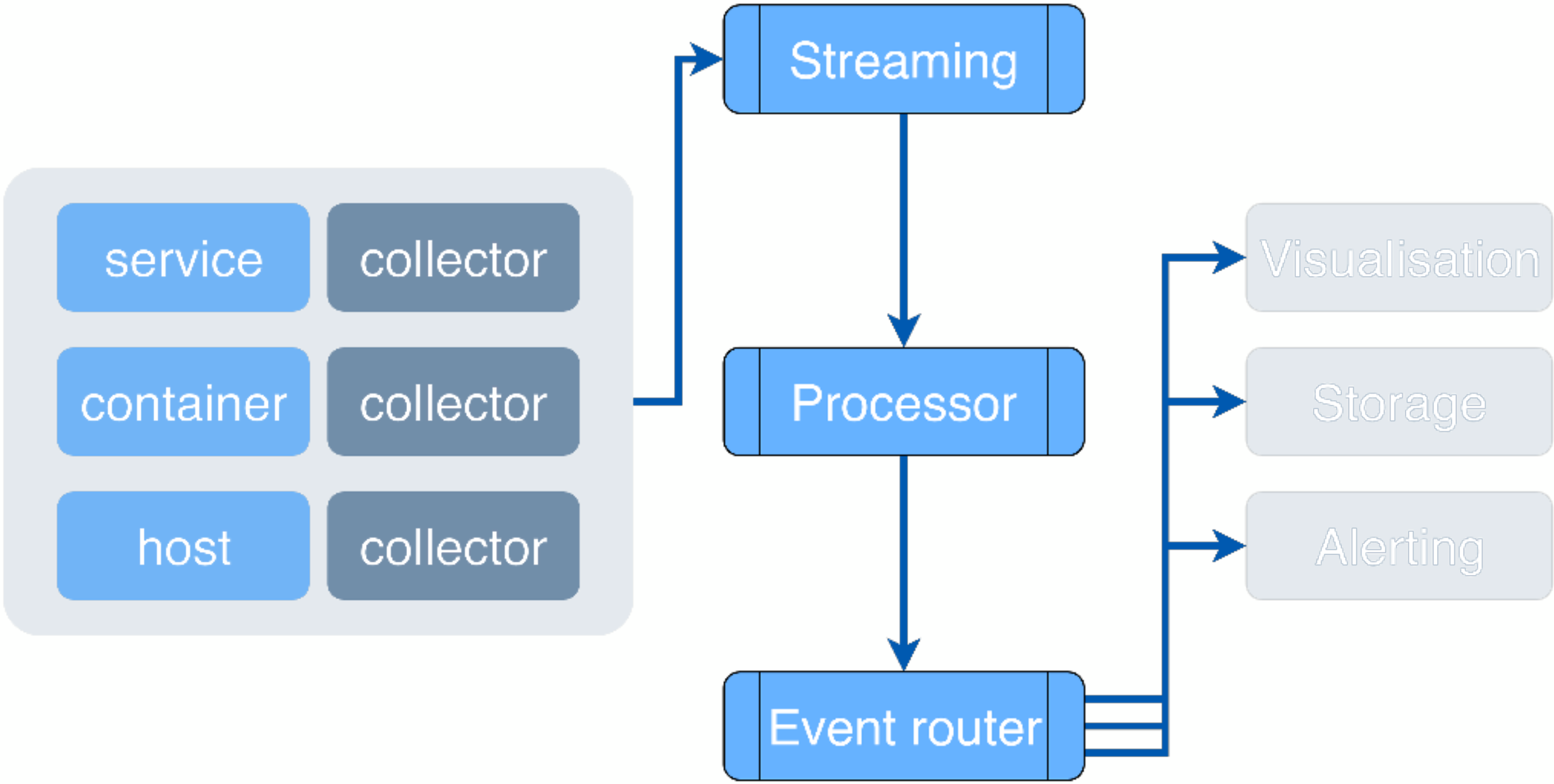
Day 2 Operations

Monitoring	Maintenance	Optimization	Upgrade
Metrics	Proactive/Reactive measures	Billing	Backup/Restore
Logging	Troubleshooting	Service/Node scaling	Upgrade/Patch
Visualisation	Debugging	Auditing/ Compliance	Disaster recovery
Alerting	New versions		
	Configuration management		

Considerations

- Collecting (data types, local-scrapping)
  - Metrics vs. Logs
- Processing (centralised, event-routing)
  - Local processing during collection or tailored data directly from the application
- Alerting (external integrations)
  - Chat tools
- Visualisation (tailored dashboards)
- Storage (data retention)


service
collector
container
collector
host
collector
Processor
Visualisation
Storage
Alerting


service
collector
container
collector
host
collector
Streaming
Processor
Event router
Visualisation
Storage
Alerting

Cloud Native Landscape

v20180425

See the interactive landscape at l.cncf.io

Greyed logos are not open source

App Definition and Development

Database and Data Warehouse

Vitess (CNCF Incubating)

Streaming

NATS (CNCF Incubating)

Source Code Management

Application Definition and Image Build

Continuous Integration / Continuous Delivery (CI/CD)

Orchestration & Management

Scheduling & Orchestration

kubernetes (CNCF Graduated)

Coordination & Service Discovery

CoreDNS (CNCF Incubating)

Service Management

gRPC (CNCF Incubating), envoy (CNCF Incubating), LINKERD (CNCF Incubating), Open Policy Agent (CNCF Sandbox)

Runtime

Cloud-Native Storage

ROOK (CNCF Sandbox)

Container Runtime

containerd (CNCF Incubating), rkt (CNCF Incubating)

Cloud-Native Network

CNI (CNCF Incubating)

Provisioning

Host Management / Tooling

Infrastructure Automation

Container Registries

Secure Images

Notary (CNCF Incubating), TUF (CNCF Incubating)

Key Management

spiffe (CNCF Sandbox), SPIRE (CNCF Sandbox)

Cloud

Public

Private

l.cncf.io

This landscape is intended as a map through the previously uncharted terrain of cloud native technologies. There are many routes to deploying a cloud native application, with CNCF Projects representing a particularly well-traveled path.

CLOUD NATIVE Landscape

CLOUD NATIVE COMPUTING FOUNDATION

Redpoint Amplify

Platforms

Certified Kubernetes - Distribution

Certified Kubernetes - Hosted

Certified Kubernetes - Installer

Non-Certified Kubernetes

PaaS/Container Service

Observability & Analysis

Monitoring

Prometheus (CNCF Incubating)

Logging

fluentd (CNCF Incubating)

Tracing

Jaeger (CNCF Incubating), OPENTRACING (CNCF Incubating)

Serverless

Special

Kubernetes Certified Service Provider

Kubernetes Training Partner

Day 2 Operations

Monitoring	Maintenance	Optimization	Upgrade
Metrics	Proactive/Reactive measures	Billing	Backup/Restore
Logging	Troubleshooting	Service/Node scaling	Upgrade/Patch
Visualisation	Debugging	Auditing/ Compliance	Disaster recovery
Alerting	New versions		
	Configuration management		

Proactive/Reactive measures

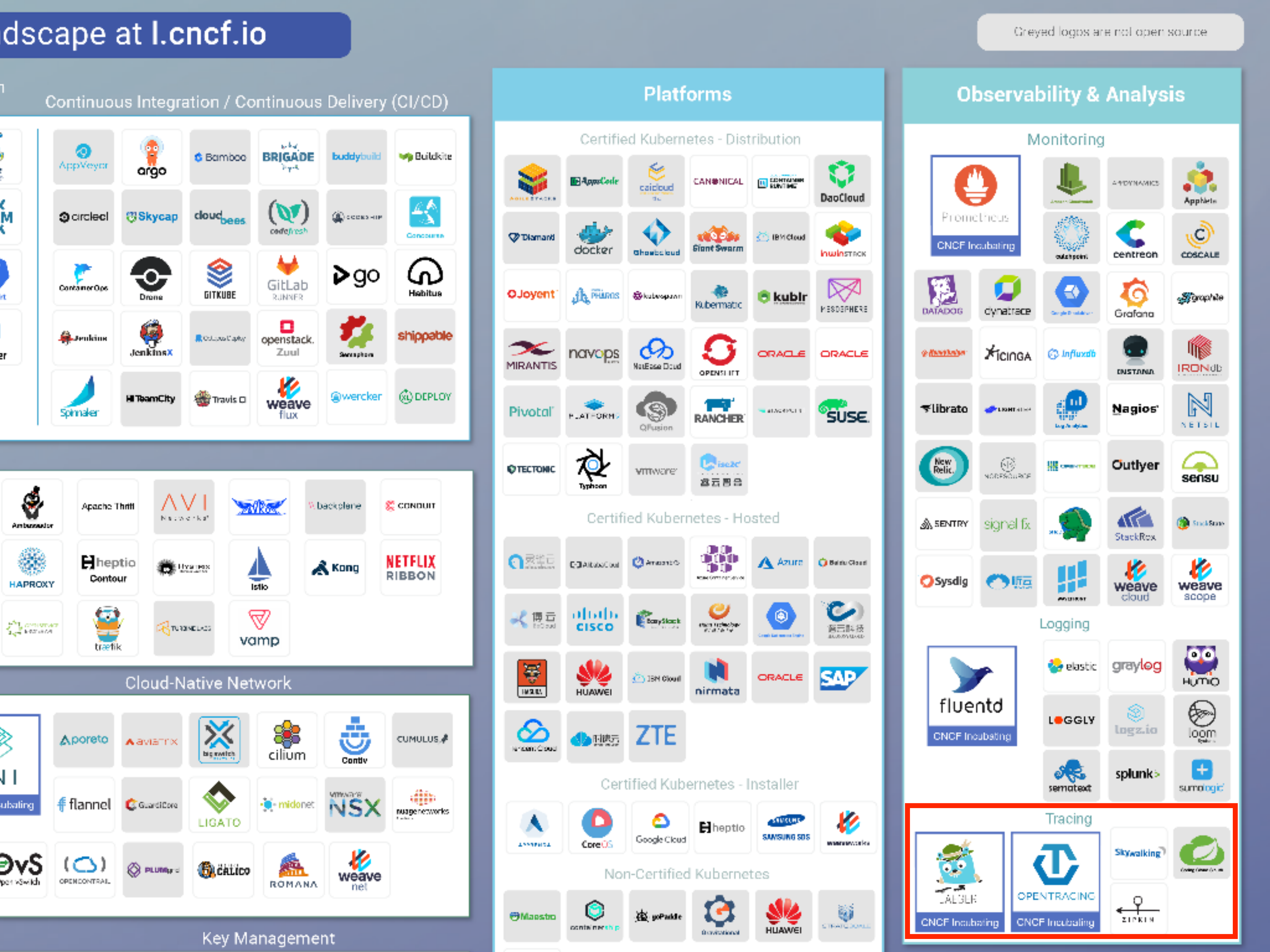
- High availability
- Chaos engineering
    - Proactively break parts of the system to understand how it reacts.
- Health-checks
    - Containers, Services, Hosts
- Recovery
    - How is the continuous operation of the cluster and the services accomplished? What happens when cluster or critical component goes down?

Effective troubleshooting

- A high level view to discover where the problem has originated

- Capability of tracing an issue throughout the stack (the idea is to track and identify issues and perform root-cause analysis in distributed environment)

- Effectively communicate the problem



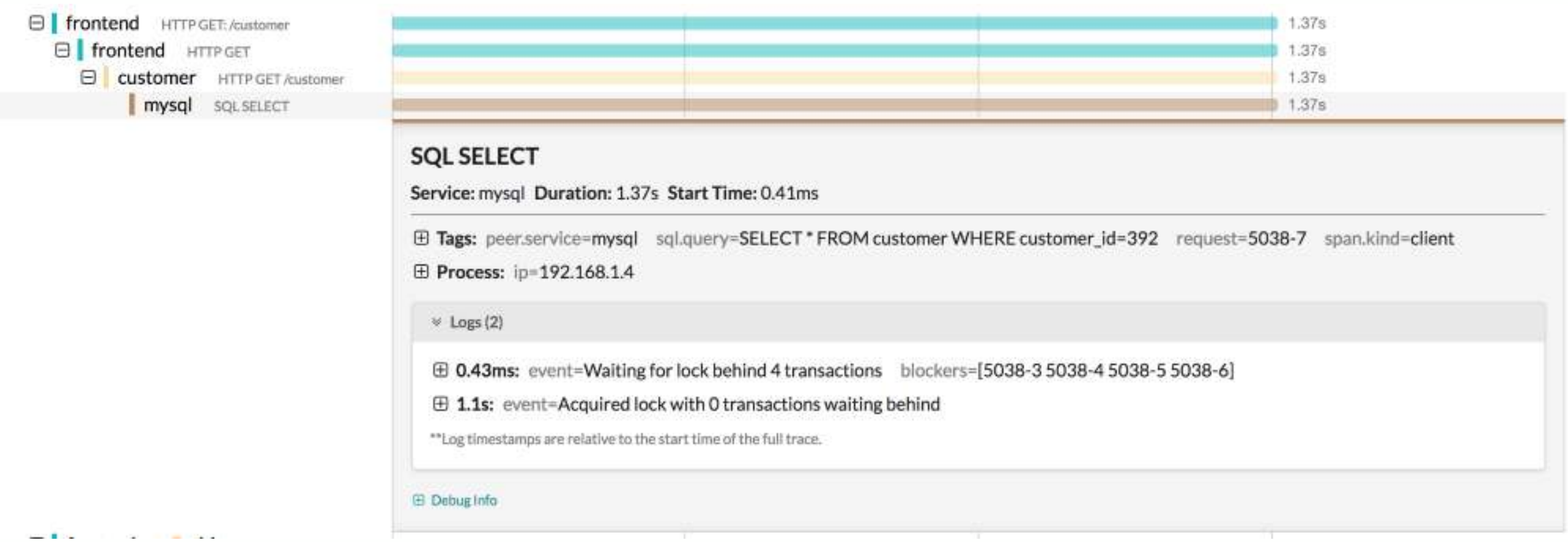
dscape at l.cncf.io
Greyed logos are not open source
Continuous Integration / Continuous Delivery (CI/CD)
Platforms
Certified Kubernetes - Distribution
Certified Kubernetes - Hosted
Certified Kubernetes - Installer
Non-Certified Kubernetes
Observability & Analysis
Monitoring
CNCF Incubating
Logging
CNCF Incubating
Tracing
CNCF Incubating
CNCF Incubating
Cloud-Native Network
Key Management

Jaeger UI | Lookup by Trace ID... | Dependencies | Search

» frontend: HTTP GET /dispatch

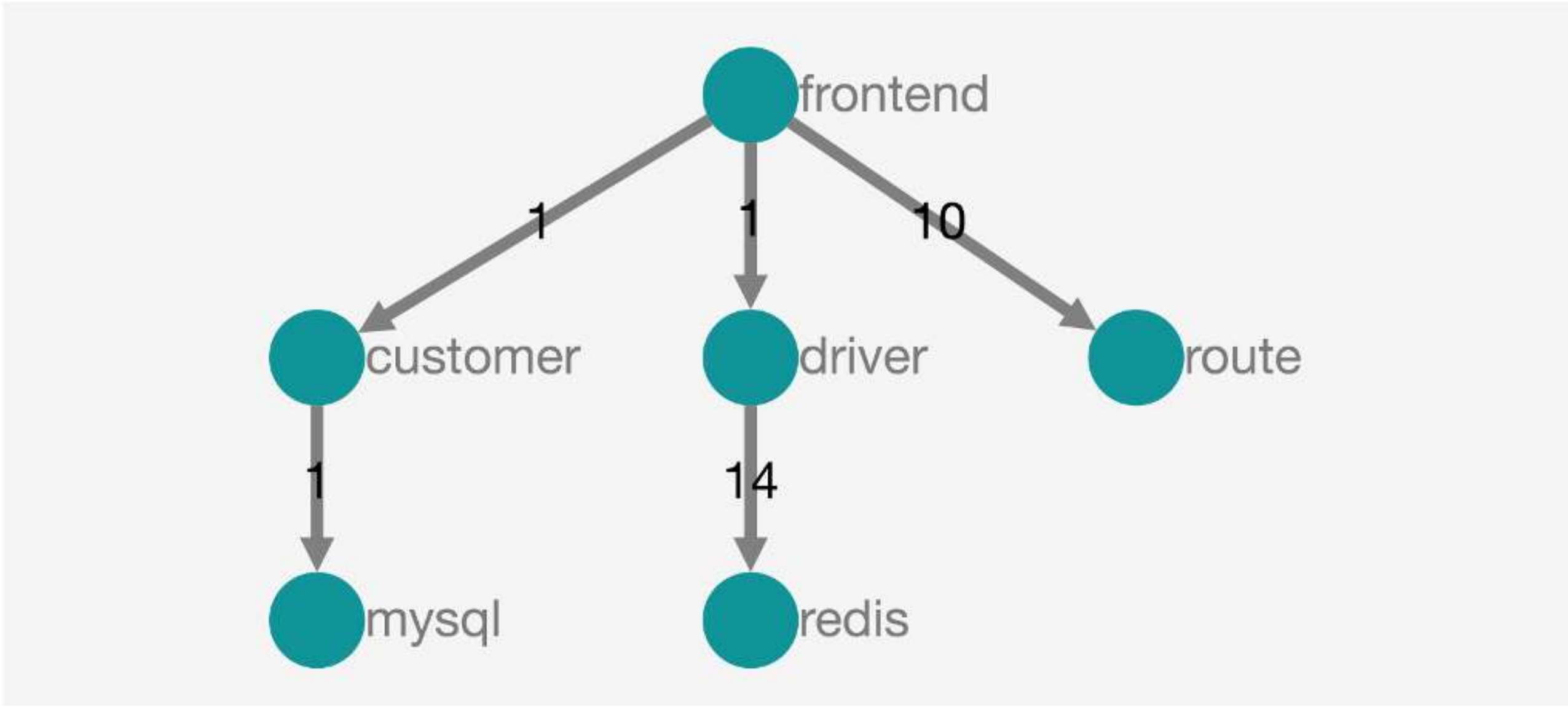
View Options | Search...

Span Name	Timeline	193.71ms	387.42ms	581.14ms	774.85ms
frontend HTTP GET /dispatch					
frontend HTTP GET: /customer	307.98ms				
frontend HTTP GET	307.91ms				
customer HTTP GET /customer	305.66ms				
mysql SQL SELECT	305.47ms				
frontend Driver::findNearest	256.79ms				
driver Driver::findNearest	256.37ms				
redis FindDriverIDs	21.51ms				
redis GetDriver	27.81ms				
redis GetDriver	9.66ms				
redis GetDriver	11.6ms				
redis GetDriver	11.71ms				
redis GetDriver	8.04ms				
redis GetDriver	41.64ms				
redis GetDriver	10.67ms				
redis GetDriver	12.76ms				
redis GetDriver	14.81ms				
redis GetDriver	7.66ms				
redis GetDriver	29.31ms				
redis GetDriver	10.74ms				
redis GetDriver	16.11ms				
frontend HTTP GET: /route	64.24ms				
frontend → route HTTP GET /route	64.19ms				
frontend HTTP GET: /route	69.13ms				
frontend → route HTTP GET /route	69.08ms				
frontend HTTP GET: /route	45.62ms				
frontend → route HTTP GET /route	45.56ms				
frontend HTTP GET: /route	54.55ms				
frontend → route HTTP GET /route	54.52ms				


frontend HTTP GET: /customer
frontend HTTP GET
customer HTTP GET /customer
mysql SQL SELECT
1.37s
1.37s
1.37s
1.37s
SQL SELECT
Service: mysql Duration: 1.37s Start Time: 0.41ms
Tags: peer.service=mysql sql.query=SELECT * FROM customer WHERE customer_id=392 request=5038-7 span.kind=client
Process: ip=192.168.1.4
Logs (2)
0.43ms: event=Waiting for lock behind 4 transactions blockers=[5038-3 5038-4 5038-5 5038-6]
1.1s: event=Acquired lock with 0 transactions waiting behind
**Log timestamps are relative to the start time of the full trace.
Debug Info

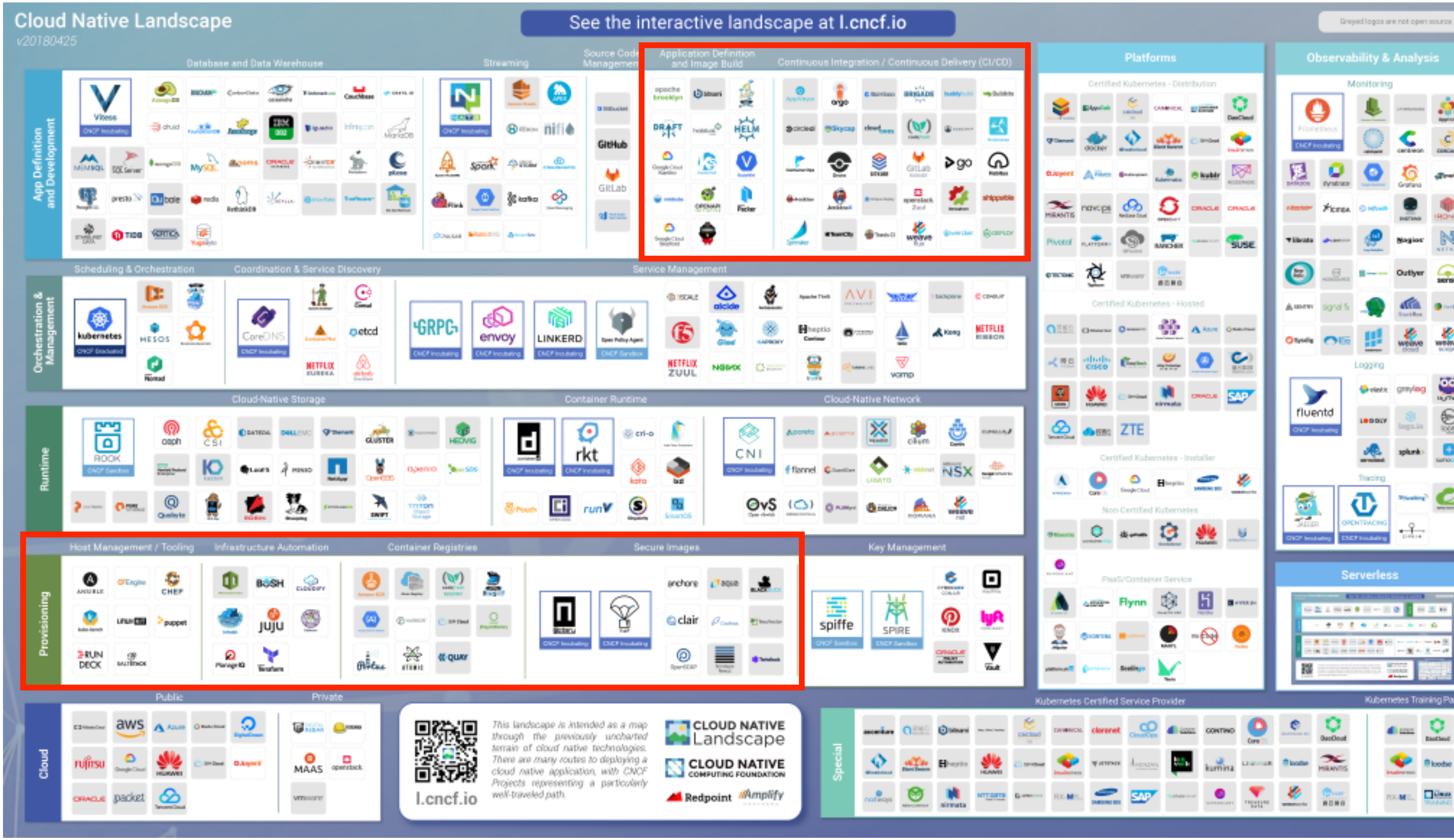
Jaeger UI | Lookup by Trace ID... | Dependencies | Search

Force Directed Graph | **DAG**



Configuration management

- Container image changes vs. applying changes to EC2 etc. (Ansible, Puppet, Chef, Terraform, Cloudformation…)
- How to install version X?
- Immutable infrastructure



Cloud Native Landscape
v20180425
See the interactive landscape at l.cncf.io
Greyed logos are not open source
App Definition and Development
Database and Data Warehouse
Streaming
Source Code Management
Application Definition and Image Build
Continuous Integration / Continuous Delivery (CI/CD)
Platforms
Certified Kubernetes - Distribution
Certified Kubernetes - Hosted
Certified Kubernetes - Installer
Non-Certified Kubernetes
PaaS/Container Service
Observability & Analysis
Monitoring
Logging
Tracing
Serverless
Orchestration & Management
Scheduling & Orchestration
Coordination & Service Discovery
Service Management
Runtime
Cloud-Native Storage
Container Runtime
Cloud-Native Network
Provisioning
Host Management / Tooling
Infrastructure Automation
Container Registries
Secure Images
Key Management
Cloud
Public
Private
Special
Kubernetes Certified Service Provider
Kubernetes Training Partner
Vitess
kubernetes
CoreDNS
GRPC
envoy
LINKERD
ROOK
rkt
CNI
fluentd
JAEGER
OPENTRACING
spiffe
SPIRE
aws
MAAS
packet
This landscape is intended as a map through the previously uncharted terrain of cloud native technologies. There are many routes to deploying a cloud native application, with CNCF Projects representing a particularly well-traveled path.
l.cncf.io
CLOUD NATIVE Landscape
CLOUD NATIVE COMPUTING FOUNDATION
Redpoint
Amplify

Day 2 Operations

Monitoring	Maintenance	Optimization	Upgrade
Metrics	Proactive/Reactive measures	Billing	Backup/Restore
Logging	Troubleshooting	Service/Node scaling	Upgrade/Patch
Visualisation	Debugging	Auditing/ Compliance	Disaster recovery
Alerting	New versions		
	Configuration management		

Billing

- Are your resources being used effectively?
- Disposal of old instances in the cloud
- Down-scaling and up-scaling, spot instances

Scaling

- When to scale what (service-level vs. node)?

- Does your scaling policy fit your overall design?

Auditing and compliance

- Everything is running as it should and there is no drift in configuration?
- Who accessed that in what way and when and how?
- Who get’s to install which service in what way?
- What services can talk to each other and in what way?
- Is everything getting where it needs to be? Does some traffic need priority?

Day 2 Operations

Monitoring	Maintenance	Optimization	Upgrade
Metrics	Proactive/Reactive measures	Billing	Backup/Restore
Logging	Troubleshooting	Service/Node scaling	Upgrade/Patch
Visualisation	Debugging	Auditing/ Compliance	Disaster recovery
Alerting	New versions		
	Configuration management		

Key take-aways

- Properly managing cloud-native systems isn’t straightforward!
- It can get out of hand pretty quickly
- Sooner or later it’s a complex system with multiple cooperating parts and different owners across the whole stack
- Unify as much as you can

“What is your preferred interface towards this kind of system?”

- https://github.com/cncf/landscape
- http://opentracing.io
- https://github.com/jaegertracing/jaeger
- https://medium.com/opentracing/take-opentracing-for-a-hotrod-ride-f6e3141f7941