KubeCon CloudNativeCon

Europe 2018

Are You Ready to be Edgy?

Bringing Your Applications to the Edge of the Network

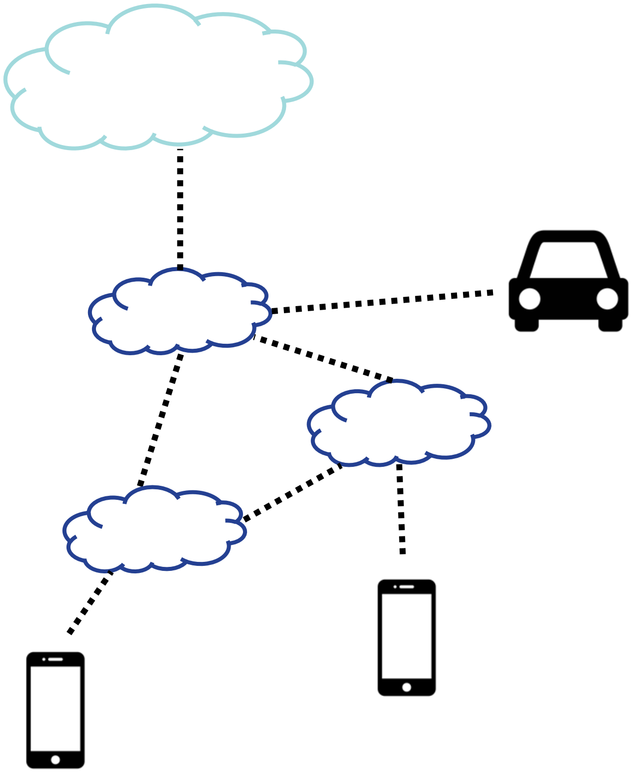
Megan O’Keefe Steve Louie

Hello!

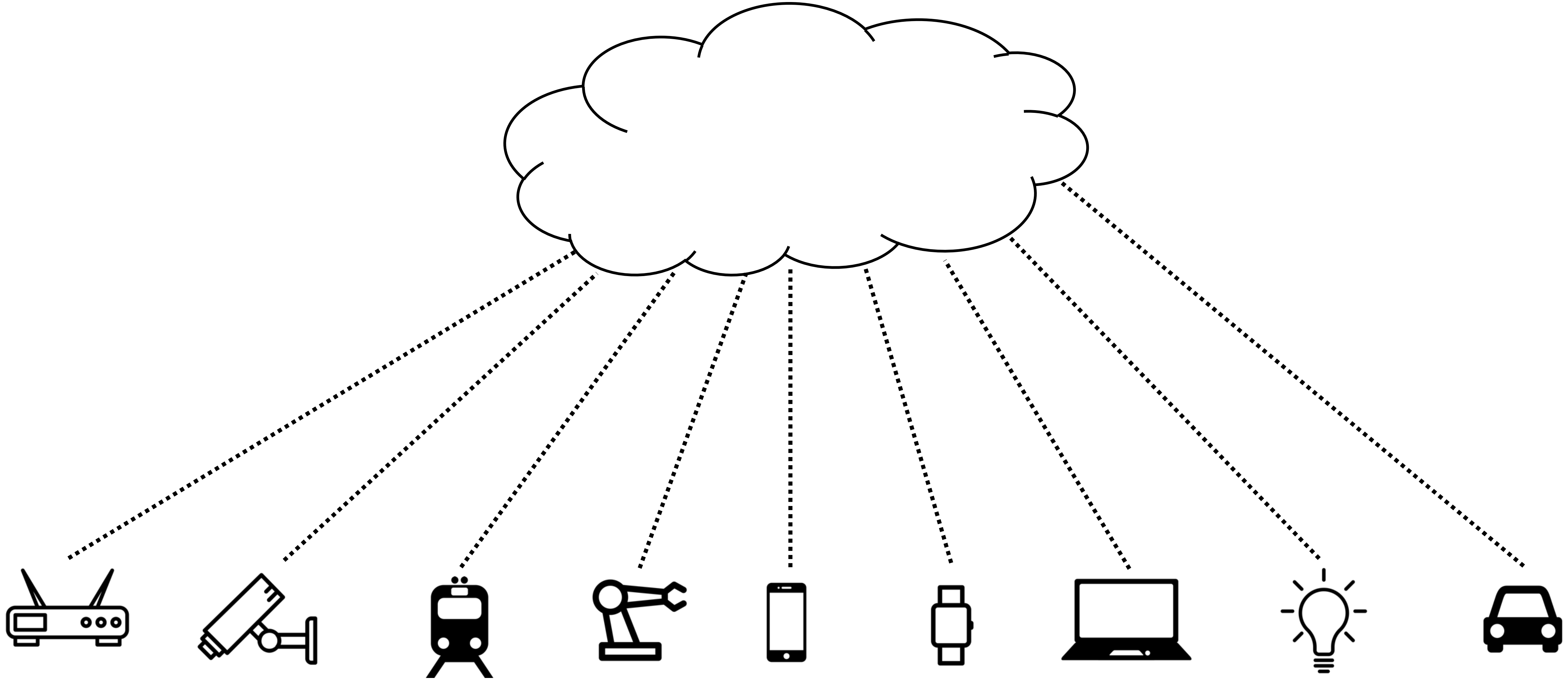
- We are Software Engineers at Cisco
- Based out of Massachusetts, USA
- We work on customer-facing prototypes in Cloud, video, and data center technologies
- Last 18 months = Edge Computing

On Deck

1. Why Edge Computing?
2. What is the Edge?
3. Cool Edge Use Cases
4. Edge-Ready Apps
5. Edge + Kubernetes
6. Demo!



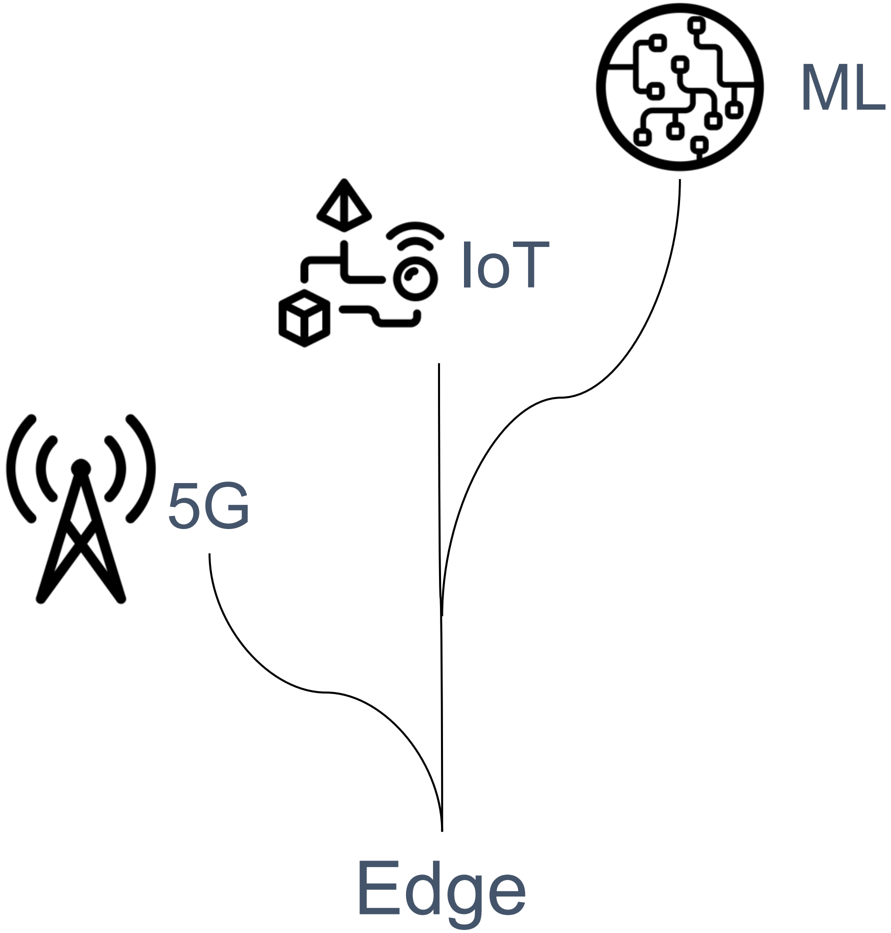
The Bandwidth Problem



The Latency Problem

- Evolution of wireline and **5G mobile** access technologies
- Proliferation of **IoT** sensors, devices, and connected Things
- Need for low-latency and real-time, data-driven decision making (**machine learning)**
- Web / OTT / App companies vying for **subscriber proximity**

→ Meeting all these needs requires an evolution of network, **compute**, storage, security, **app orchestration**

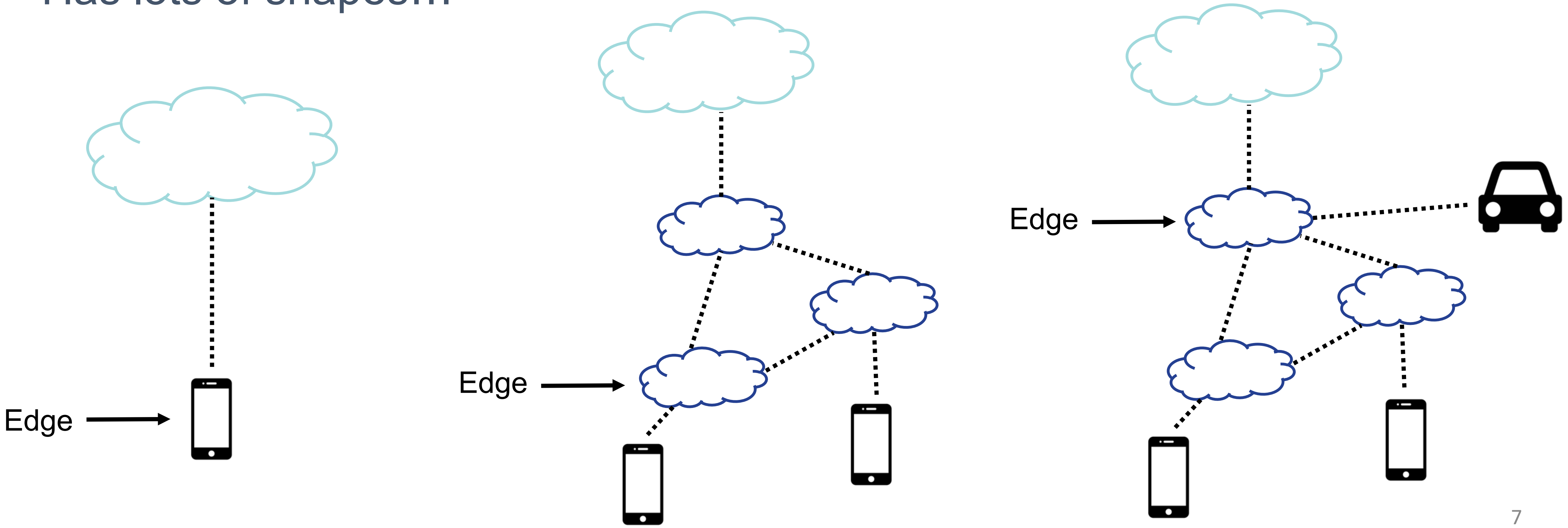


What is Edge Computing?

Edge Computing is the process of moving intensive workloads from the Cloud out to the **Edge** of the network.

Edge Computing…

- Extends the Cloud to the network Edge
- Is not a replacement for the Cloud
- Has lots of shapes…



Benefits of Edge Computing

Low Latency
(VR/AR)

High Bandwidth
(Video Processing)

Temporary/Secure Data
(Facial Recognition)

Edge Computing Use Cases

IoT

- **Healthcare**
- Energy
- Lighting
- Transportation
- Public safety
- Disaster detection
- Industrial robotics
- Agriculture
- Drones
- Wearables

Consumer

- Gaming
- VR/AR
- Secure messaging

Enterprise

- **Machine learning**
- Collaboration
- Virtual desktop
- Firewall

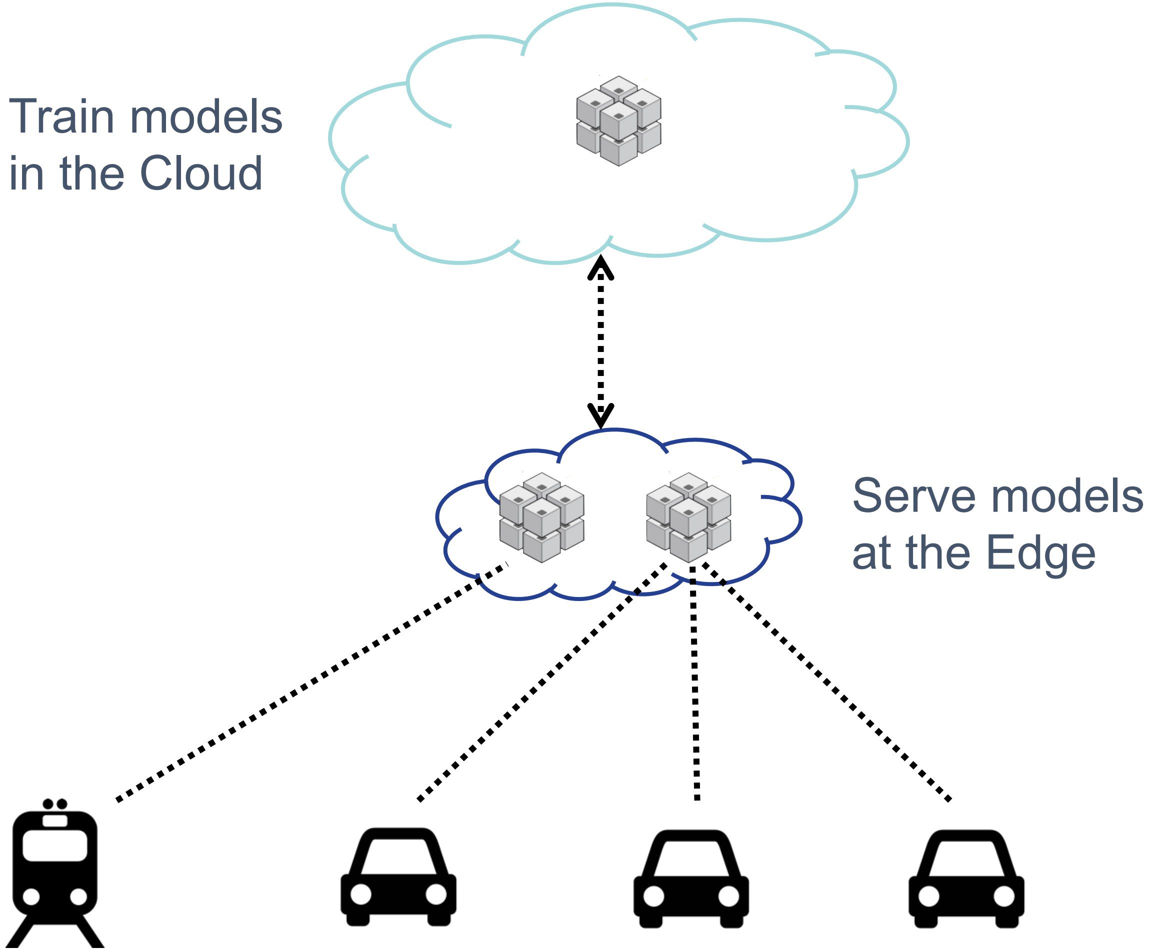
Web / OTT

- Edge as a Service
- CDN
- Video optimization

Service Provider

- C-RAN
- NFV

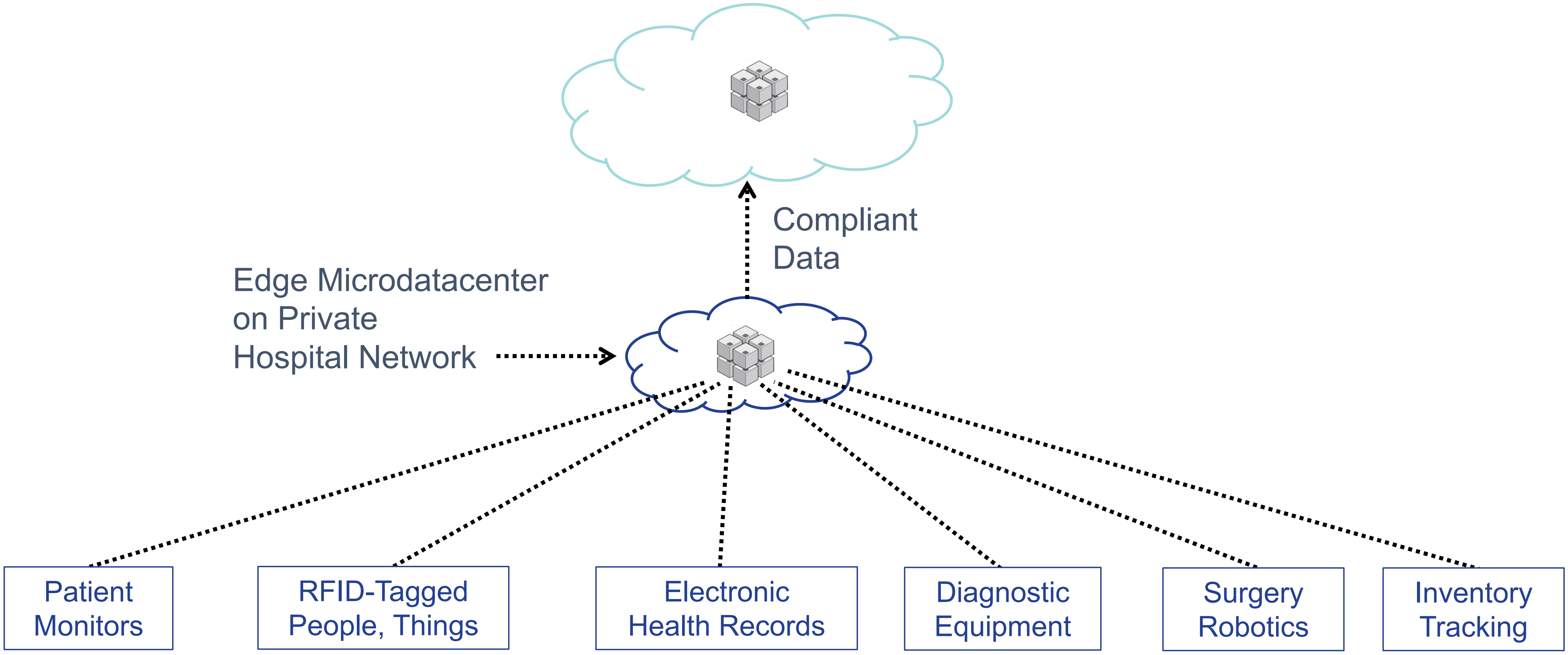
Edge Use Case: Enterprise ML



Train models in the Cloud

Serve models at the Edge

Edge Use Case: Healthcare



Compliant Data

Edge Microdatacenter on Private Hospital Network

Patient Monitors

RFID-Tagged People, Things

Electronic Health Records

Diagnostic Equipment

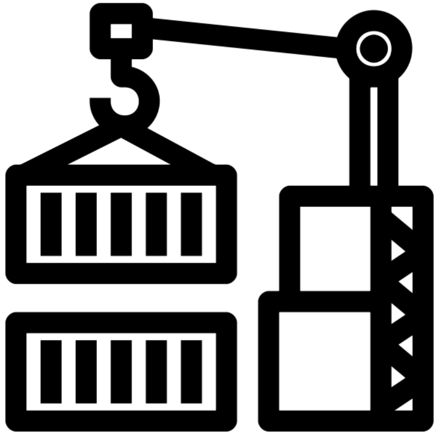
Surgery Robotics

Inventory Tracking

Edge-Ready Applications

Architecture

- API-, Microservices-Driven – Application can be deconstructed to run in Cloud, at the Edge
- Hierarchical Topology – Edge workloads and Cloud workloads have different purposes
- Minimal Size, Maximum Portability – Small, ideally <100MB application images built for maximum portability across environments



Edge-Ready Applications

Scalability

- Edge components must be highly scalable across thousands of clusters
- Resource consumption must scale linearly with additional Edge resources

Fault Tolerance

- Edge microservices can run offline
- Failure resiliency – Graceful handling of network partitioning events and hardware failures

Edge-Ready Applications

Storage

- Cache ephemeral state at the edge as much as possible
- Sync persistent state with the public Cloud

Network

- Have a set of common service hostnames for your apps to enable anycast edge DNS routing

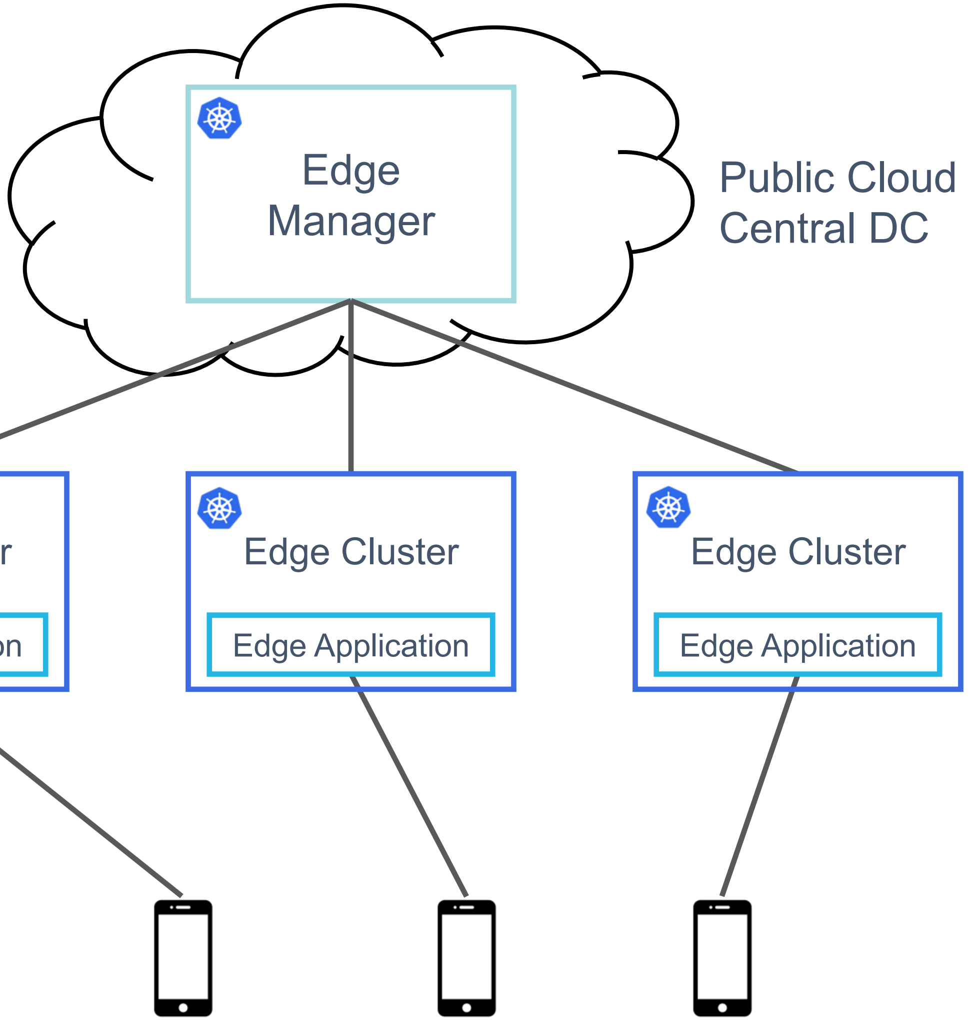
Security

- Multitenancy (multiple users per Edge deployment)
- Private Registry – signed, secure images

Why Kubernetes at the Edge?

Edge Tenets	→ Kubernetes
Vast diversity of Edge use cases	Kubernetes is workload-agnostic (containers, VMs, functions).
Edge has many shapes	Kubernetes supports many nodes and works on a variety of hardware platforms.
Edge is flexible and dynamic	Kubernetes was built to support apps moving around and being spun up and down.
Edge must be repeatable, consistent	Kubernetes provides a consistent platform that has been proven in production at scale.
Edge should be familiar to developers	Kubernetes adoption is huge! Why re-invent the wheel for Edge?

Workflow: Edge Kubernetes

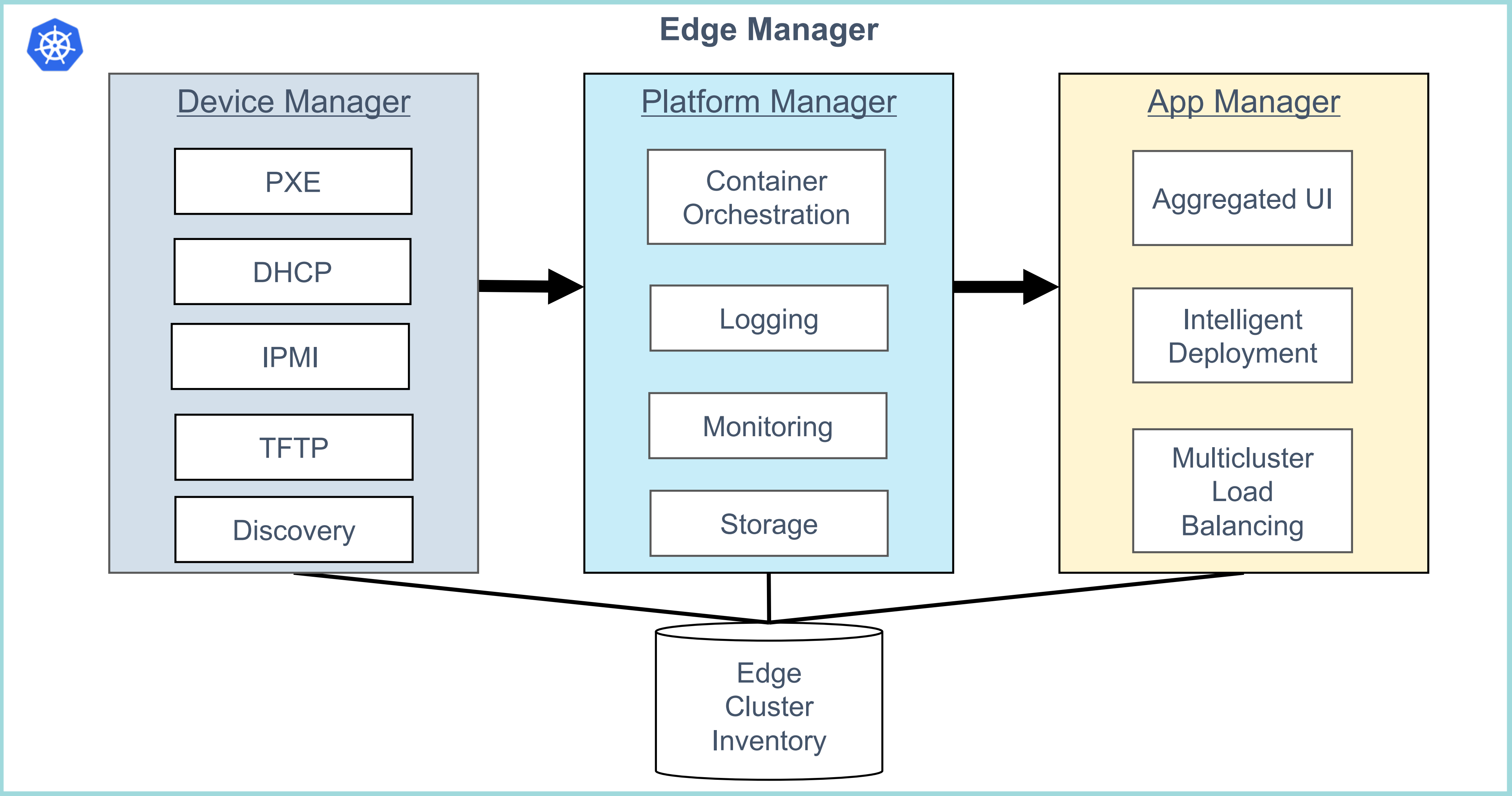


Public Cloud /
Central DC

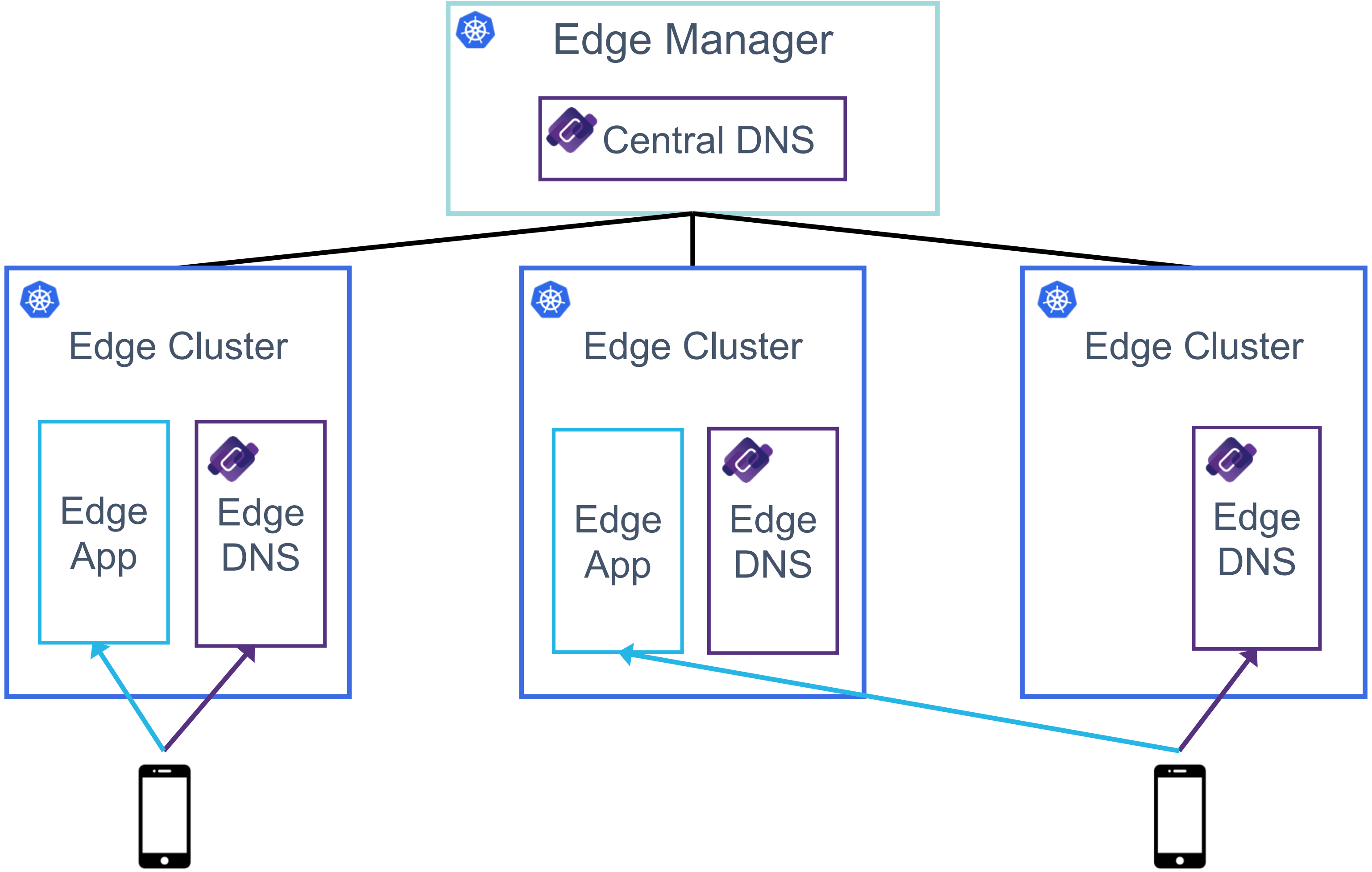
Each Edge Cluster
is autonomous

End-device traffic
routes to the closest
Edge Cluster

Edge Platform Architecture



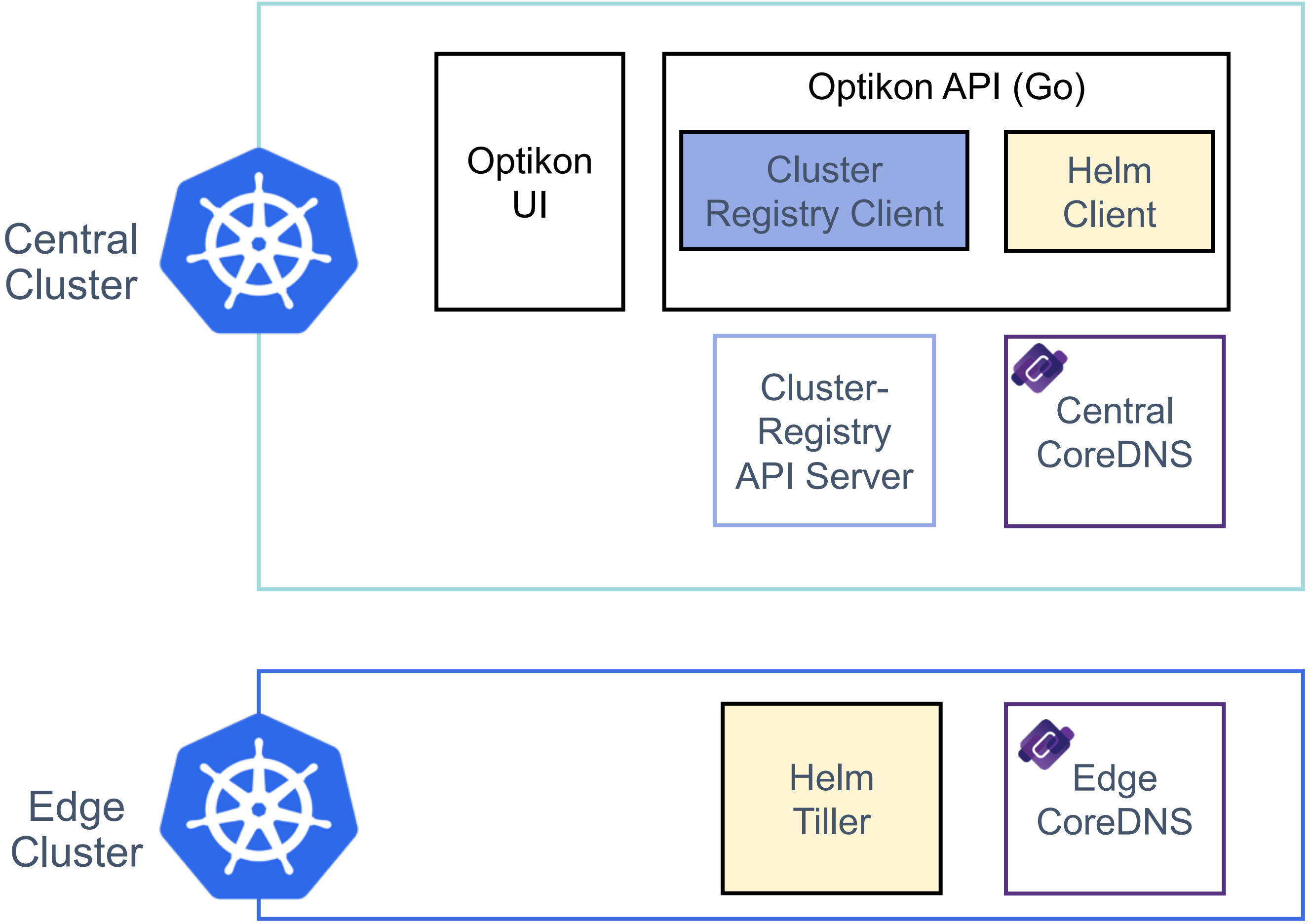
Edge Platform Architecture



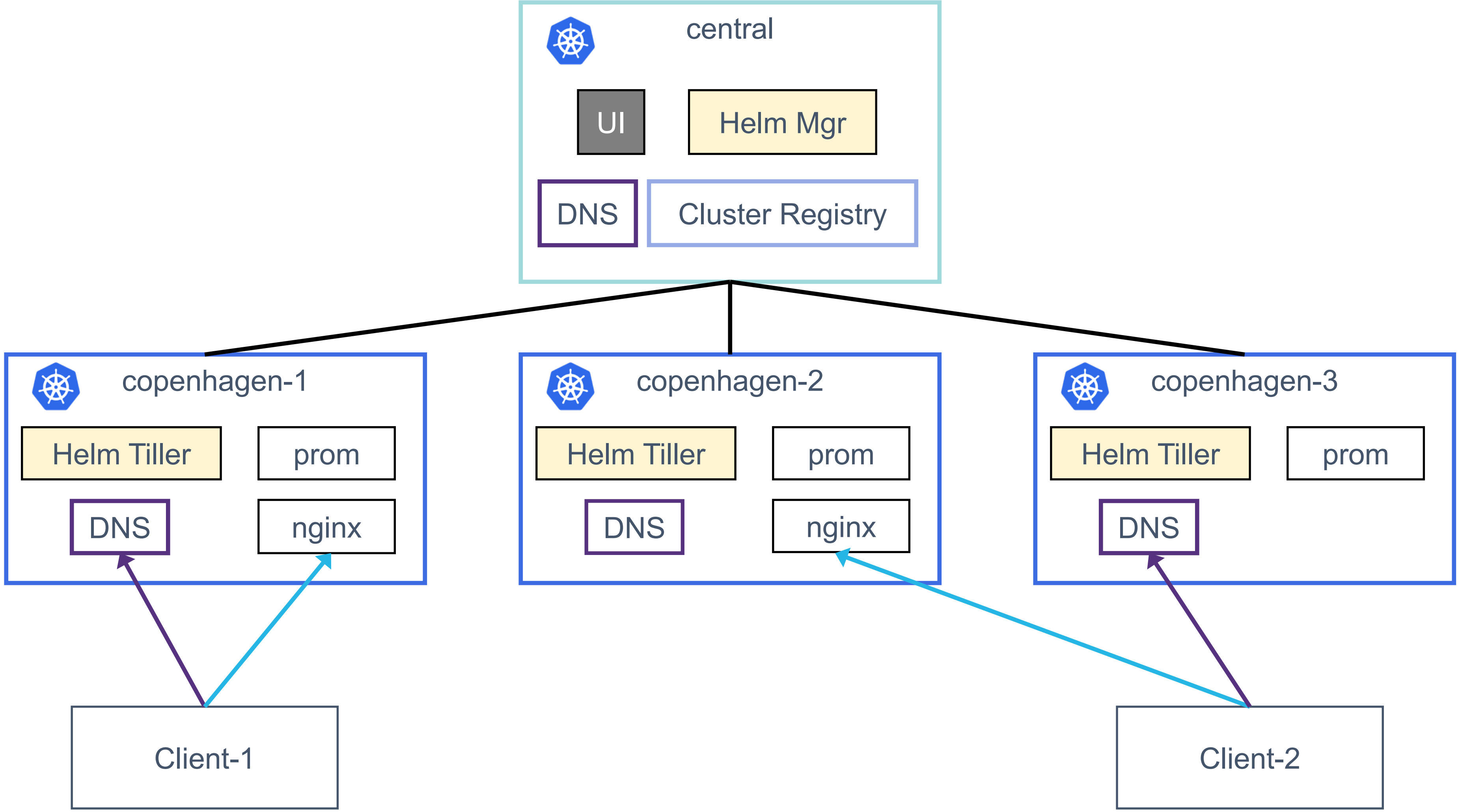
Device Flow

1. Anycast DNS for app destination
2. Device talks to nearest instance

Optikon = Edge App Management



Optikon Demo - Setup



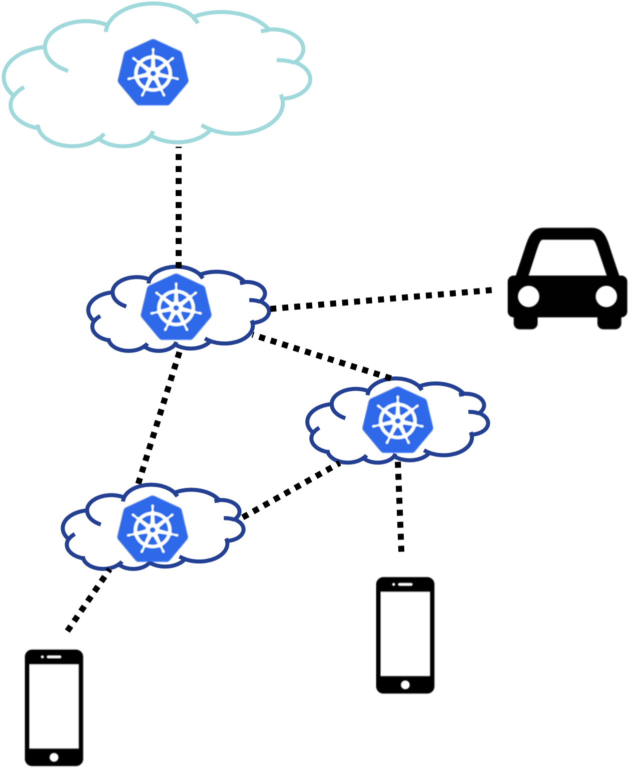
Optikon – What's Next?

- Cross-cluster auto-scaling / "fuzzy" deployments
- Integrate multi-cluster load balancing + traffic policy (Istio)
- Work with SIG Multi-cluster to help define best practices, use cases for additional tools
- Scaling Optikon to lots of Edge Kubernetes clusters

github.com/optikon

Key Takeaways

1. High bandwidth + low latency use cases → Edge
2. Use cases include IoT, caching, machine learning
3. Edge applications = flexible, scalable, secure Cloud-native applications
4. Our approach to Edge orchestration: 1 central Kubernetes cluster + 1 cluster per Edge site
5. CNCF projects = awesome building blocks to build new stuff! 🎉



Thank you!

megan037@gmail.com

stephen.r.louie@gmail.com

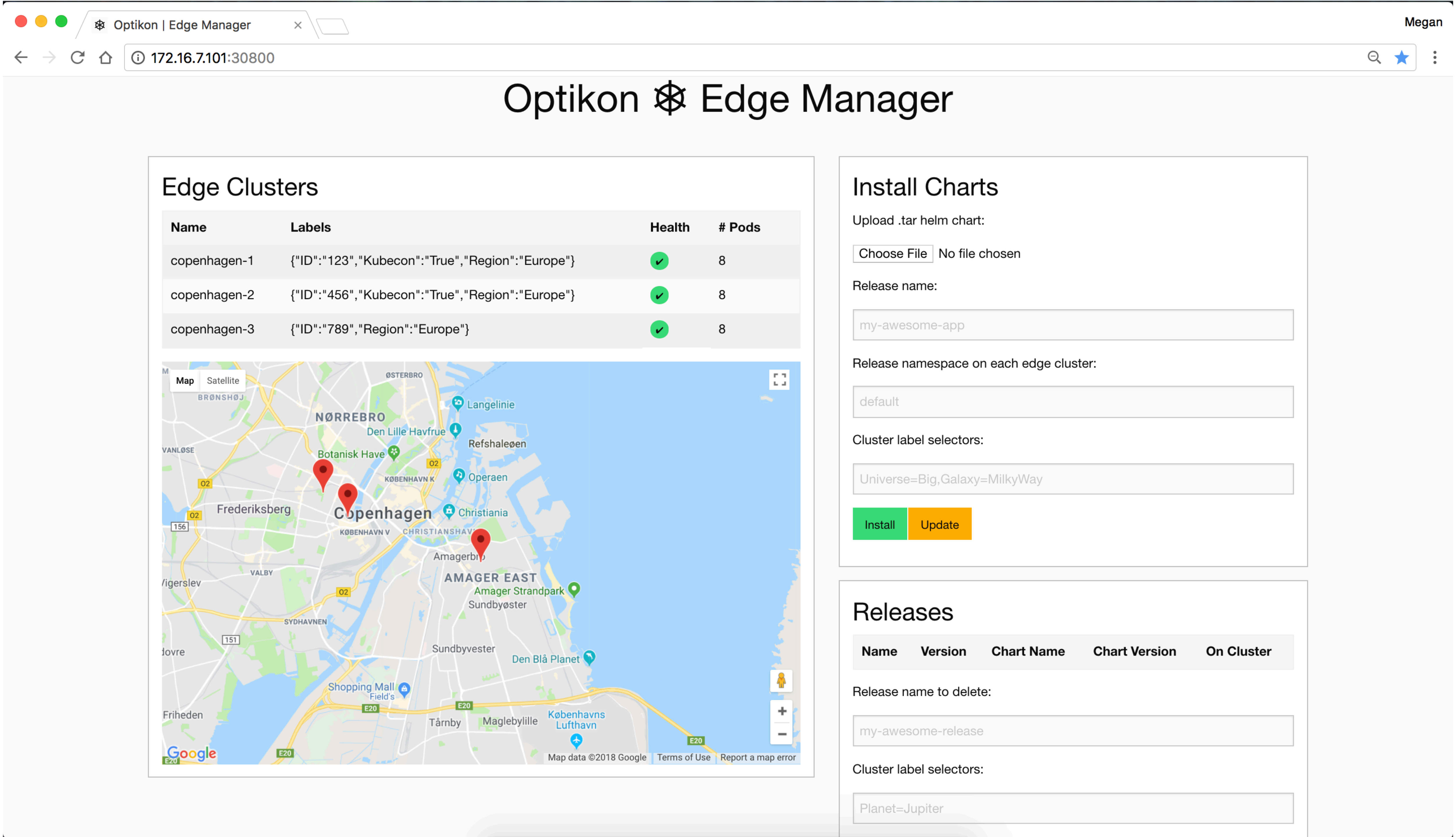
Interested in Edge?

Come to the Edge Computing + Kubernetes BOF at **14:45** today!

Room B4-M2+4

Questions?

Demo Screenshots



Optikon ⎈ Edge Manager

Edge Clusters

Name	Labels	Health	# Pods
copenhagen-1	{"ID":"123","Kubecon":"True","Region":"Europe"}	✓	8
copenhagen-2	{"ID":"456","Kubecon":"True","Region":"Europe"}	✓	8
copenhagen-3	{"ID":"789","Region":"Europe"}	✓	8

Install Charts

Upload .tar helm chart:

Choose File No file chosen

Release name:

my-awesome-app

Release namespace on each edge cluster:

default

Cluster label selectors:

Universe=Big,Galaxy=MilkyWay

Install Update

Releases

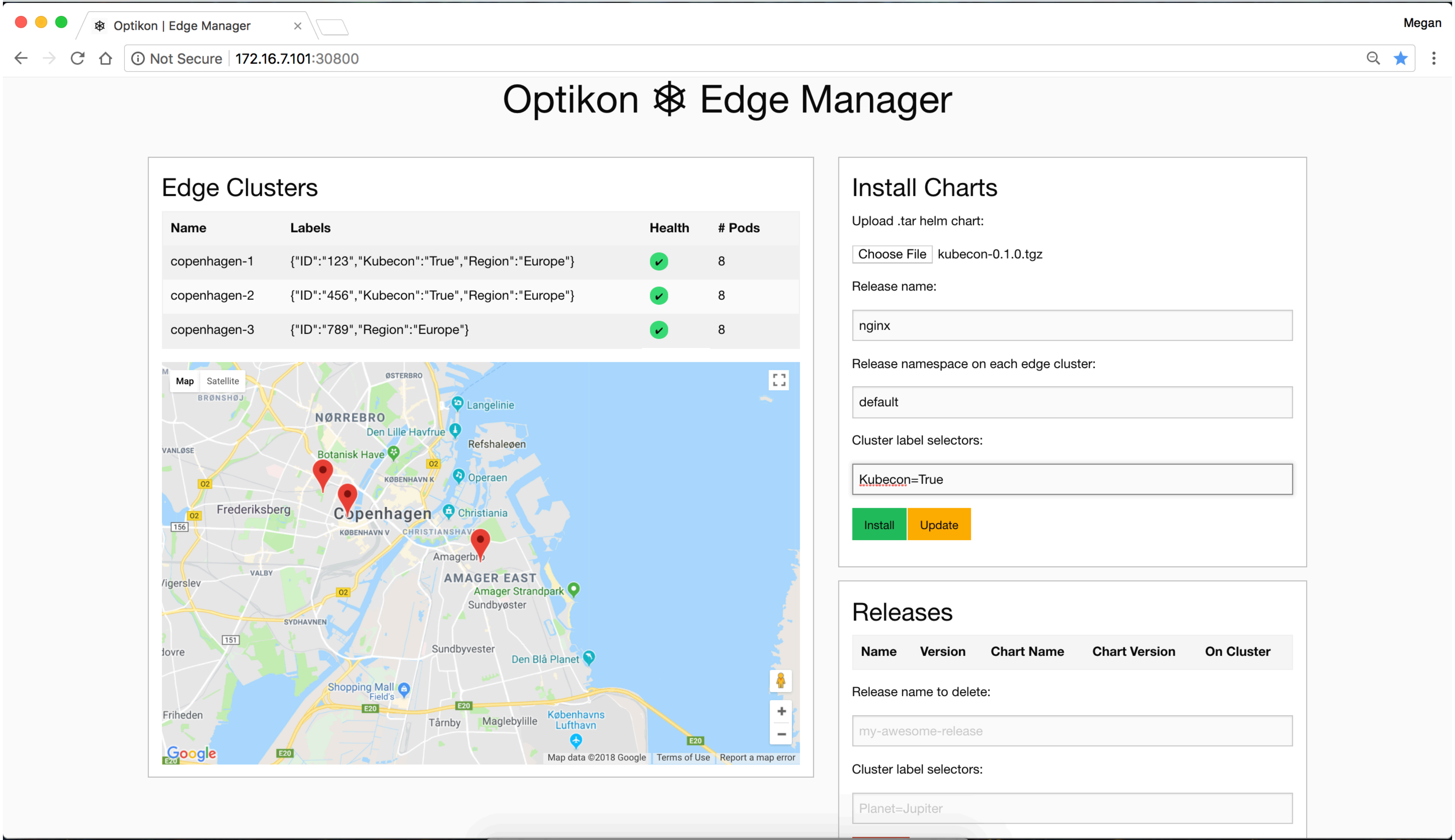
Name	Version	Chart Name	Chart Version	On Cluster

Release name to delete:

my-awesome-release

Cluster label selectors:

Planet=Jupiter



Optikon ⎈ Edge Manager

Edge Clusters

Name	Labels	Health	# Pods
copenhagen-1	{"ID":"123","Kubecon":"True","Region":"Europe"}	✓	8
copenhagen-2	{"ID":"456","Kubecon":"True","Region":"Europe"}	✓	8
copenhagen-3	{"ID":"789","Region":"Europe"}	✓	8

Install Charts

Upload .tar helm chart:

Choose File kubecon-0.1.0.tgz

Release name:

nginx

Release namespace on each edge cluster:

default

Cluster label selectors:

Kubecon=True

Install Update

Releases

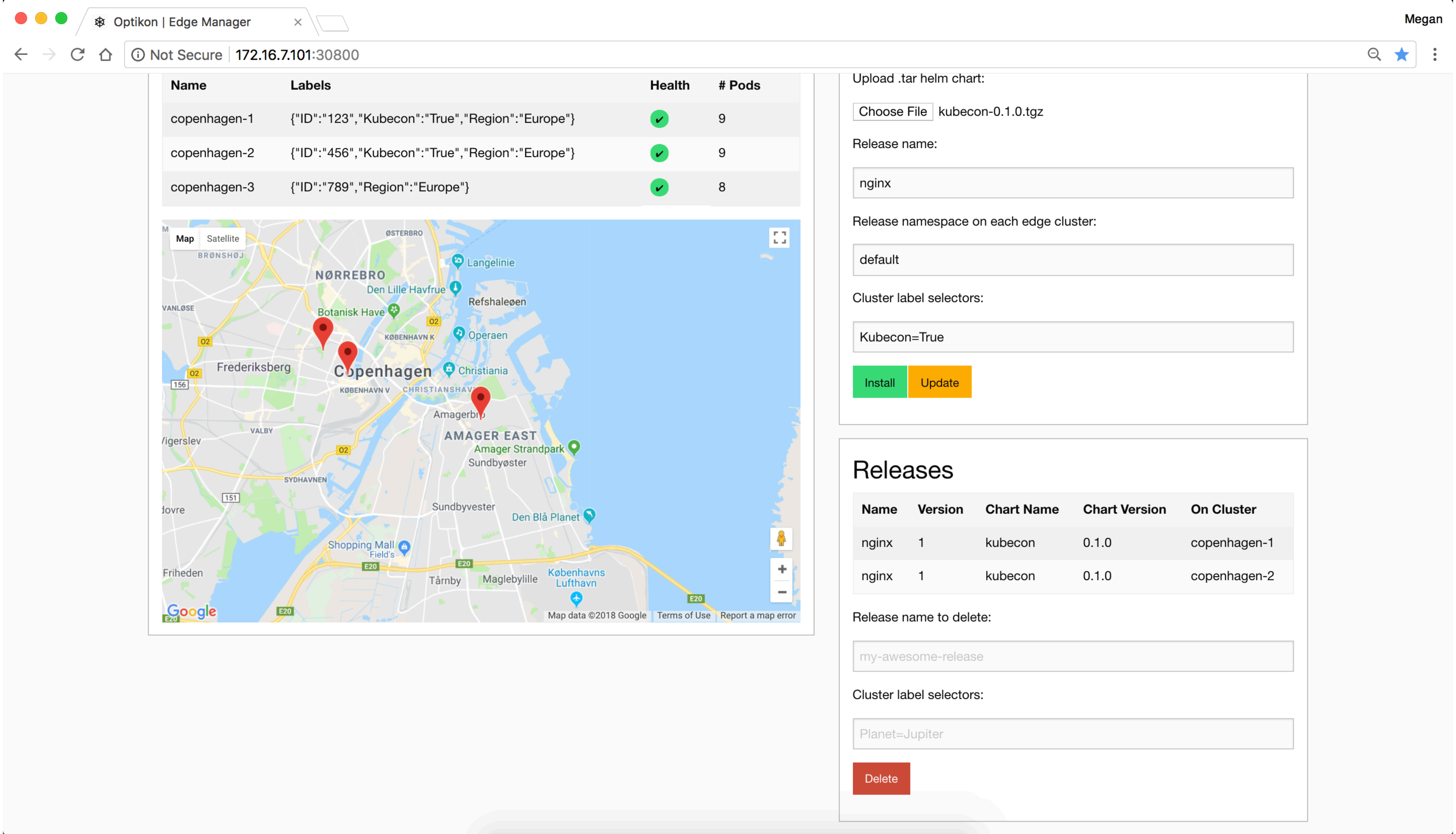
Name	Version	Chart Name	Chart Version	On Cluster

Release name to delete:

my-awesome-release

Cluster label selectors:

Planet=Jupiter



Name	Labels	Health	# Pods
copenhagen-1	{"ID":"123","Kubecon":"True","Region":"Europe"}	✓	9
copenhagen-2	{"ID":"456","Kubecon":"True","Region":"Europe"}	✓	9
copenhagen-3	{"ID":"789","Region":"Europe"}	✓	8

Upload .tar helm chart:

Choose File kubecon-0.1.0.tgz

Release name:

nginx

Release namespace on each edge cluster:

default

Cluster label selectors:

Kubecon=True

Install Update

Releases

Name	Version	Chart Name	Chart Version	On Cluster
nginx	1	kubecon	0.1.0	copenhagen-1
nginx	1	kubecon	0.1.0	copenhagen-2

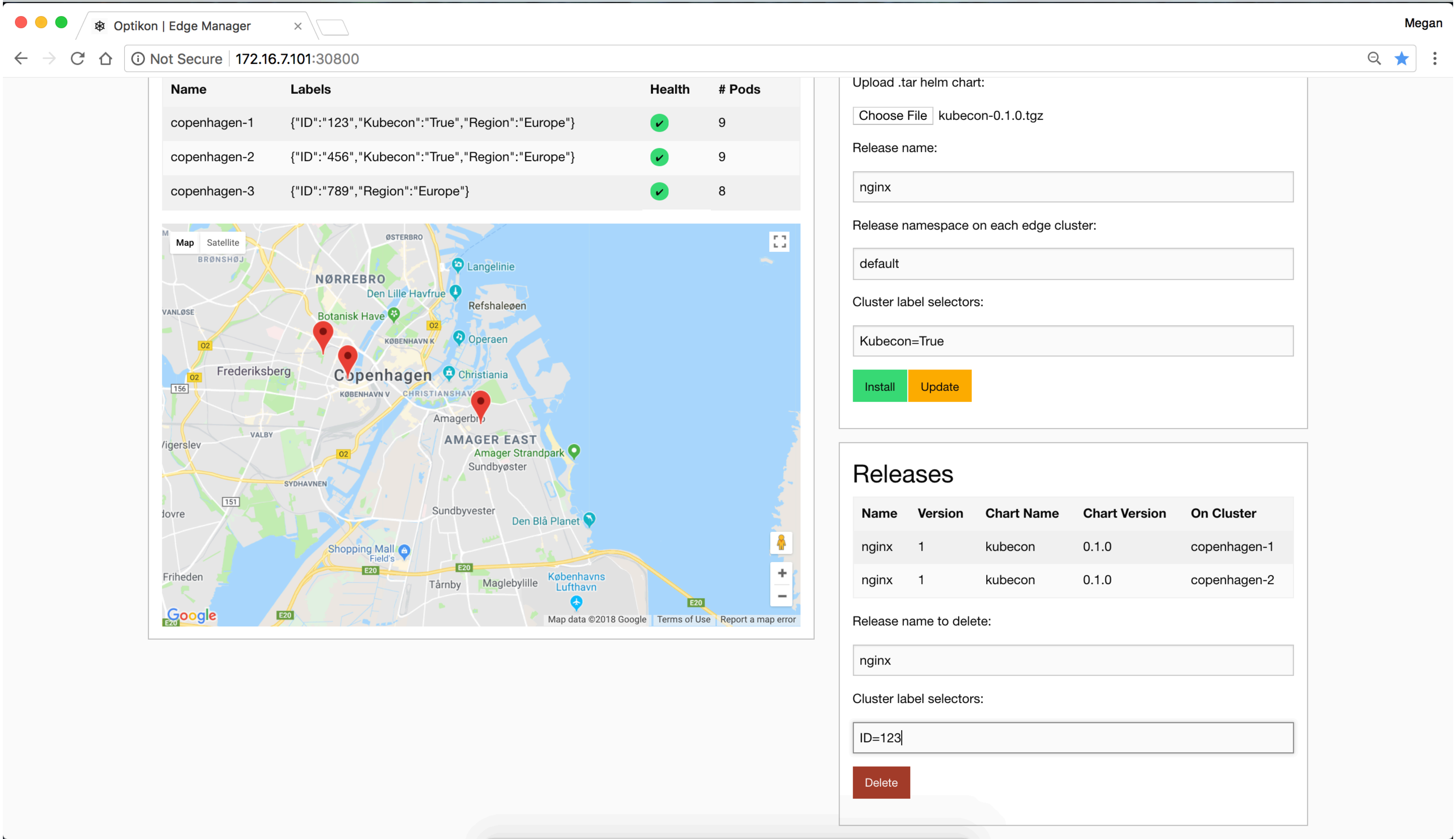
Release name to delete:

my-awesome-release

Cluster label selectors:

Planet=Jupiter

Delete



Name	Labels	Health	# Pods
copenhagen-1	{"ID":"123","Kubecon":"True","Region":"Europe"}	✓	9
copenhagen-2	{"ID":"456","Kubecon":"True","Region":"Europe"}	✓	9
copenhagen-3	{"ID":"789","Region":"Europe"}	✓	8

Upload .tar helm chart:

Choose File kubecon-0.1.0.tgz

Release name:

nginx

Release namespace on each edge cluster:

default

Cluster label selectors:

Kubecon=True

Install Update

Releases

Name	Version	Chart Name	Chart Version	On Cluster
nginx	1	kubecon	0.1.0	copenhagen-1
nginx	1	kubecon	0.1.0	copenhagen-2

Release name to delete:

nginx

Cluster label selectors:

ID=123

Delete