

September 13th, 2025

# Container Live Migration

## The Missing Piece of the Kubernetes Ecosystem

# Hello



Kunal Das

Developer Advocate APAC, CAST AI

Organizer of  
CNCf Kolkata,  
Cloud Computing Circle,  
Hashicorp User Group Bangalore

7x Azure, 1x Hashicorp Certified, FinOps Certified  
Engineer



**Kubernetes is easy**  
Isn't it?



# PROJECTS AND PRODUCTS

<p><b>Cloud Native Ecosystem</b></p> <p>RELM, K8s, Prometheus, Grafana, Istio, Service Mesh, etc.</p>	<p><b>Cloud Native Ecosystem</b></p> <p>KV, V, etc.</p>	<p><b>Cloud Native Ecosystem</b></p> <p>Flux, Argo, etc.</p>	<p><b>Cloud Native Ecosystem</b></p> <p>etc.</p>
<p><b>Cloud Native Ecosystem</b></p> <p>etc.</p>	<p><b>Cloud Native Ecosystem</b></p> <p>etc.</p>	<p><b>Cloud Native Ecosystem</b></p> <p>etc.</p>	<p><b>Cloud Native Ecosystem</b></p> <p>etc.</p>
<p><b>Cloud Native Ecosystem</b></p> <p>etc.</p>	<p><b>Cloud Native Ecosystem</b></p> <p>etc.</p>	<p><b>Cloud Native Ecosystem</b></p> <p>etc.</p>	<p><b>Cloud Native Ecosystem</b></p> <p>etc.</p>
<p><b>Cloud Native Ecosystem</b></p> <p>etc.</p>	<p><b>Cloud Native Ecosystem</b></p> <p>etc.</p>	<p><b>Cloud Native Ecosystem</b></p> <p>etc.</p>	<p><b>Cloud Native Ecosystem</b></p> <p>etc.</p>
<p><b>Cloud Native Ecosystem</b></p> <p>etc.</p>	<p><b>Cloud Native Ecosystem</b></p> <p>etc.</p>	<p><b>Cloud Native Ecosystem</b></p> <p>etc.</p>	<p><b>Cloud Native Ecosystem</b></p> <p>etc.</p>
<p><b>Cloud Native Ecosystem</b></p> <p>etc.</p>	<p><b>Cloud Native Ecosystem</b></p> <p>etc.</p>	<p><b>Cloud Native Ecosystem</b></p> <p>etc.</p>	<p><b>Cloud Native Ecosystem</b></p> <p>etc.</p>

# How to handle the zoo?

Cli ninja skills



Automation



**Automation** kills my job 🤖

**NO WAY..!**

# Automation is the key ingredient

These culprits exist because managing cloud-native infrastructure **manually** is nearly impossible.

Automation tools:

- Select the type, size, and number of compute instances your K8s workloads actually need
- Bin-pack pods into nodes to maximize resource utilization
- Manage the entire Spot instance lifecycle,
- Set the right requests to fight cloud waste while maintaining performance (rightsize workloads)

**Kubernetes is efficient per default,  
no?**





10

30%

29

50%

11

80%

3

100%

## Key findings

### CPU and memory utilization

10%

CPU utilization across clusters is **worse** this year

AVERAGE CPU UTILIZATION

23%

AVERAGE MEMORY UTILIZATION

Surprisingly, the average CPU utilization across clusters is **worse** this year than last year by a lot more than we expected: now at 10% (23% worse than in 2024), while average memory utilization was marginally better at 23%, indicating no significant year-over-year improvement in resource efficiency across cloud platforms compared to our previous report from 2024.

[Link](#)

**Let's talk about infrastructure**

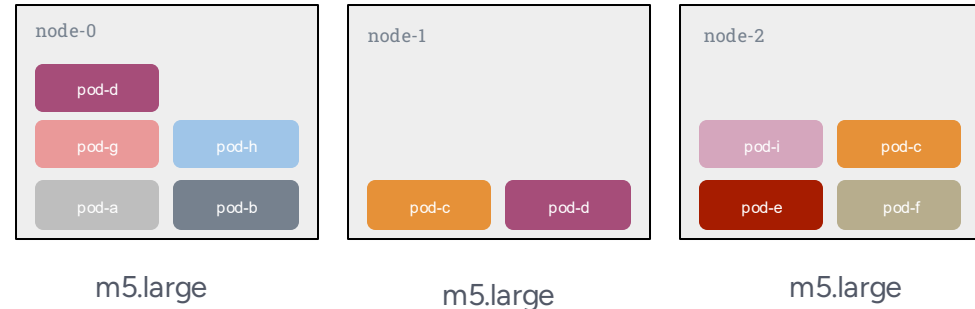
# Autoscaling with efficient compaction

Kubernetes distribute Workloads on nodes

Distribution is adjusted based on need,  
configuration, workload

**Only nodes cost money!** (ok also traffic and  
control plane)

Cost saving is not so complicated!



# Autoscaling with efficient compaction

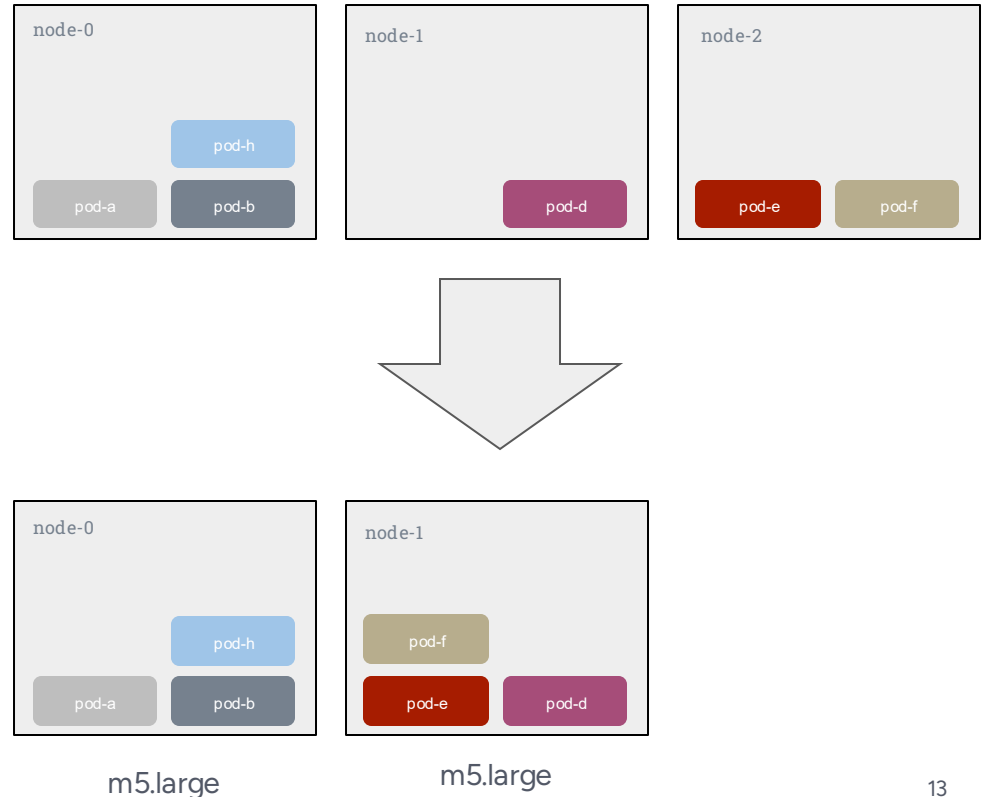
Internal Scheduling and scaling is cost neutral

Resource sharing makes Kubernetes efficient

Underutilized nodes are expensive

Make use of the elasticity of the cloud

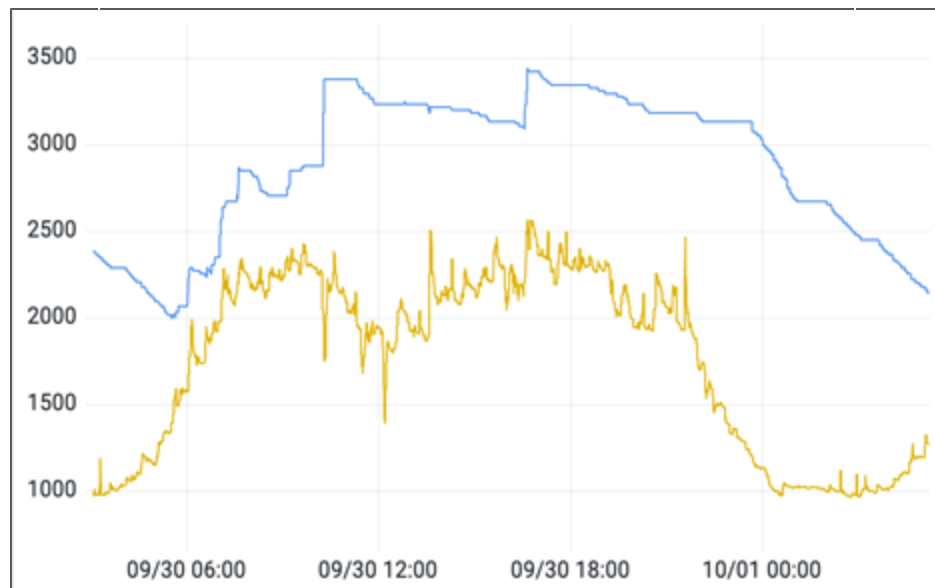
Make sure you have a bin-packing solution



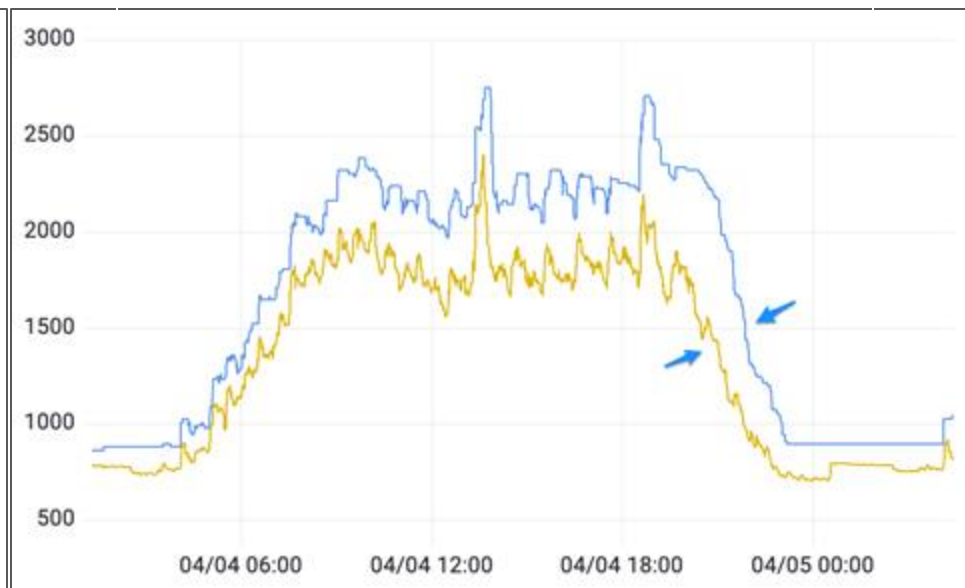
# CAST AI autoscaler

Reducing the gap between provisioned and used resources

## Before optimization



## After optimization

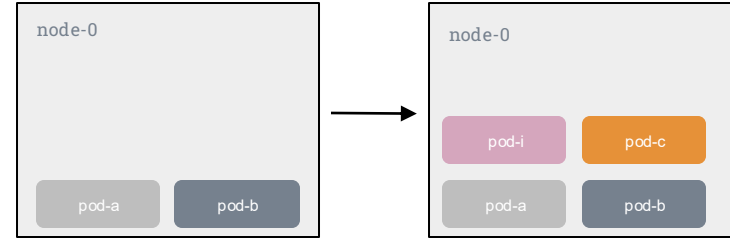


# What about workload?

# HPA and VPA

## Horizontal Pod Autoscaler

Adjust number of pods for load (CPU, requests)

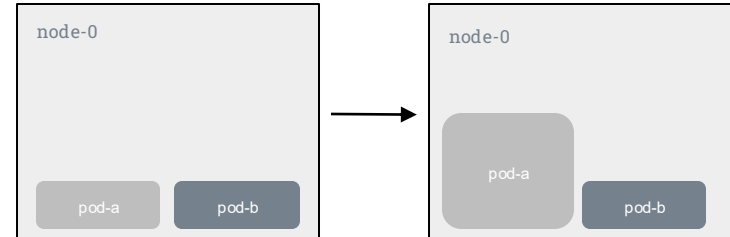


## Vertical Pod Autoscaler

Requirement for efficient nodes scaling

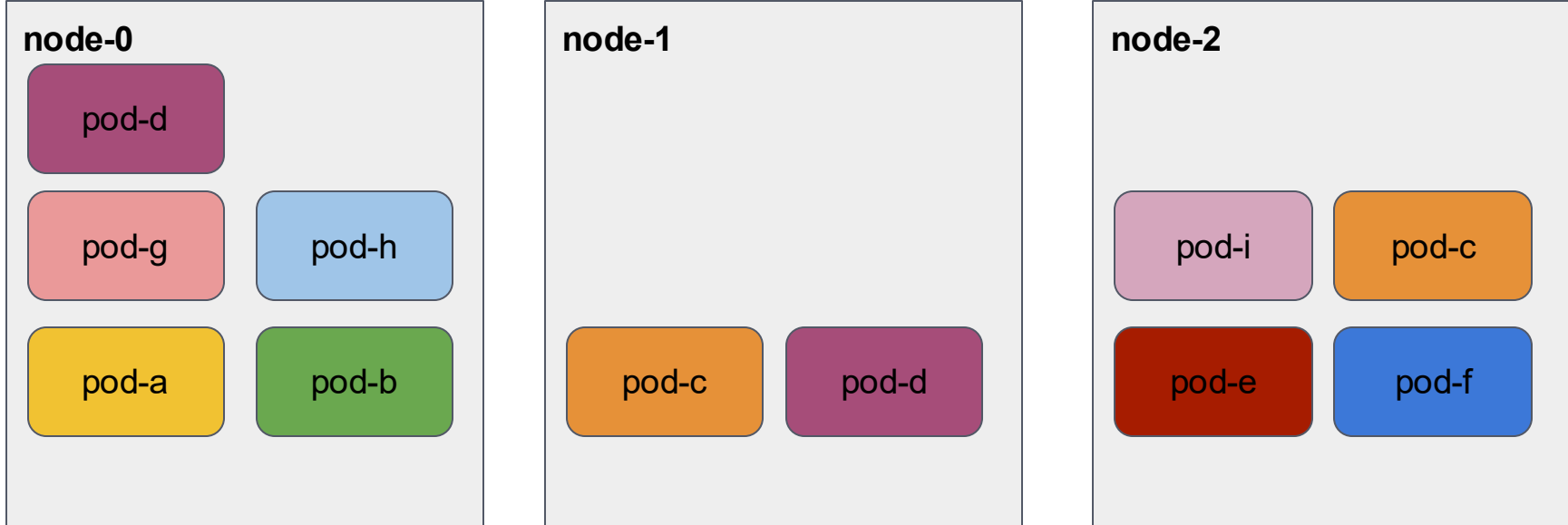
Adjust pod resource request

Dangerous side effects (pod too big to schedule)

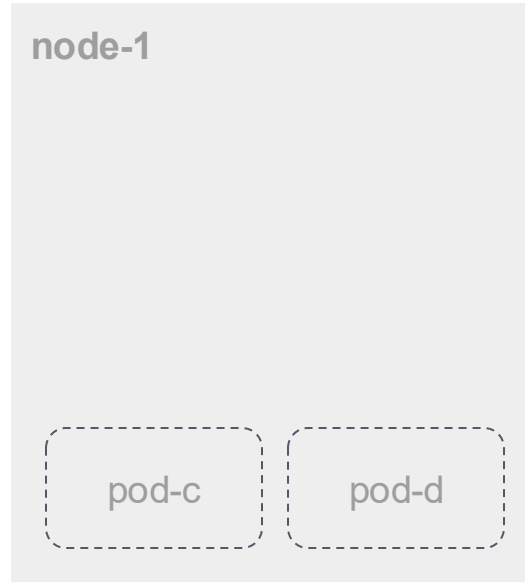
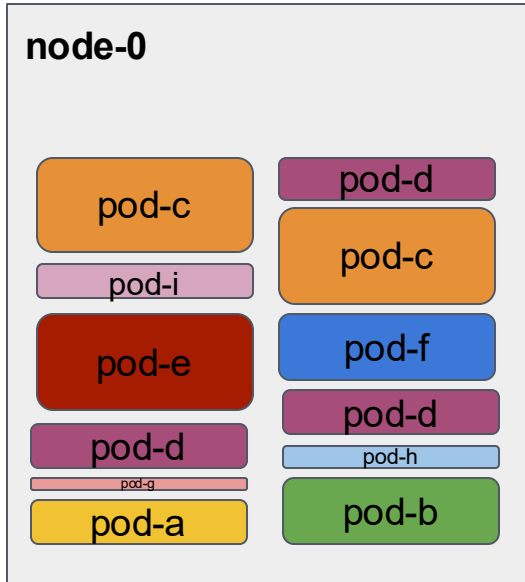




# Combining all of it - Before



# Combining all of it - After



**So..... everything is awesome?**

It's always the same....



# What does “Container Live Migration” means?

The ability to move containers (pods) in the **SAME** cluster  
between nodes without interruption and restarts.



How does it work?

## Node A

### POD

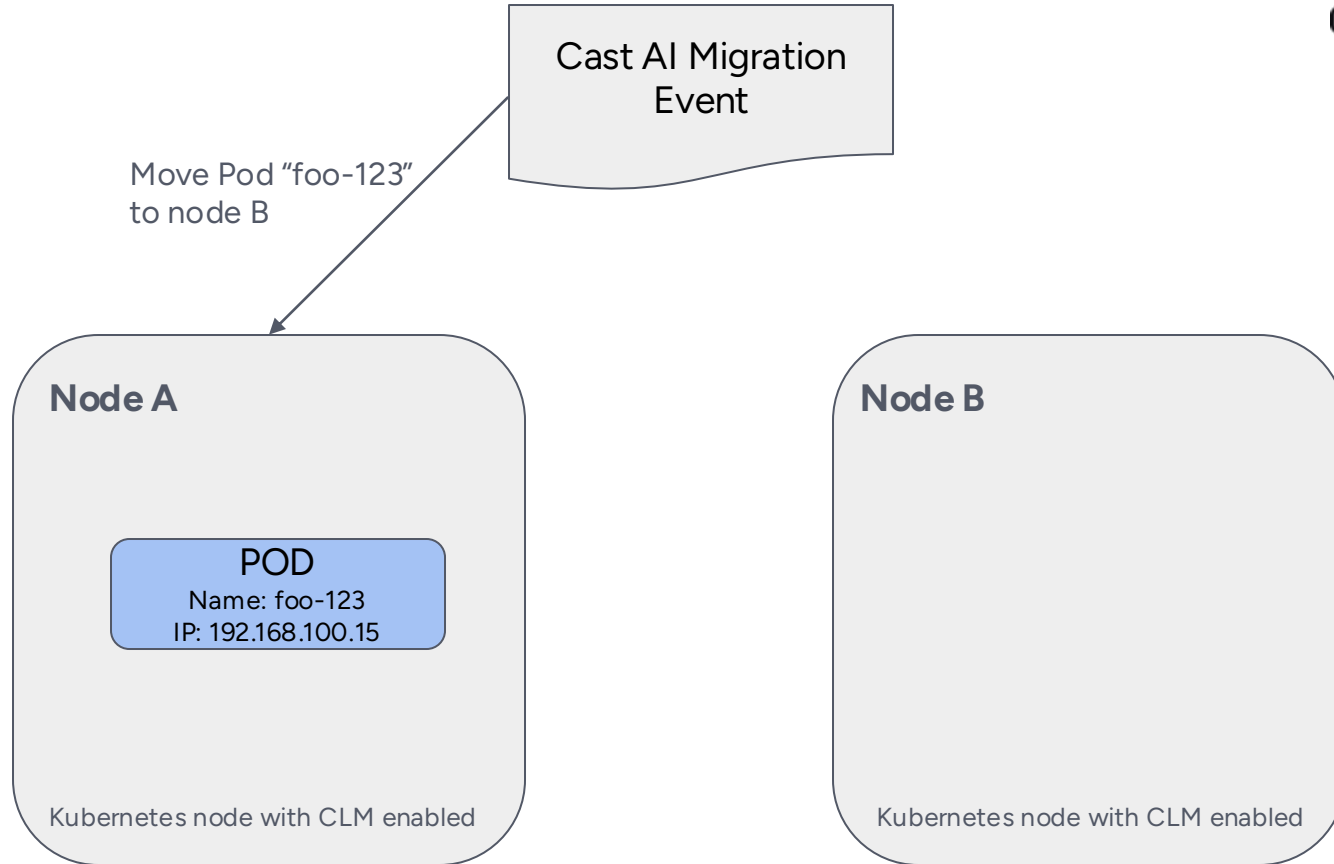
Name: foo-123  
IP: 192.168.100.15

Kubernetes node with CLM enabled

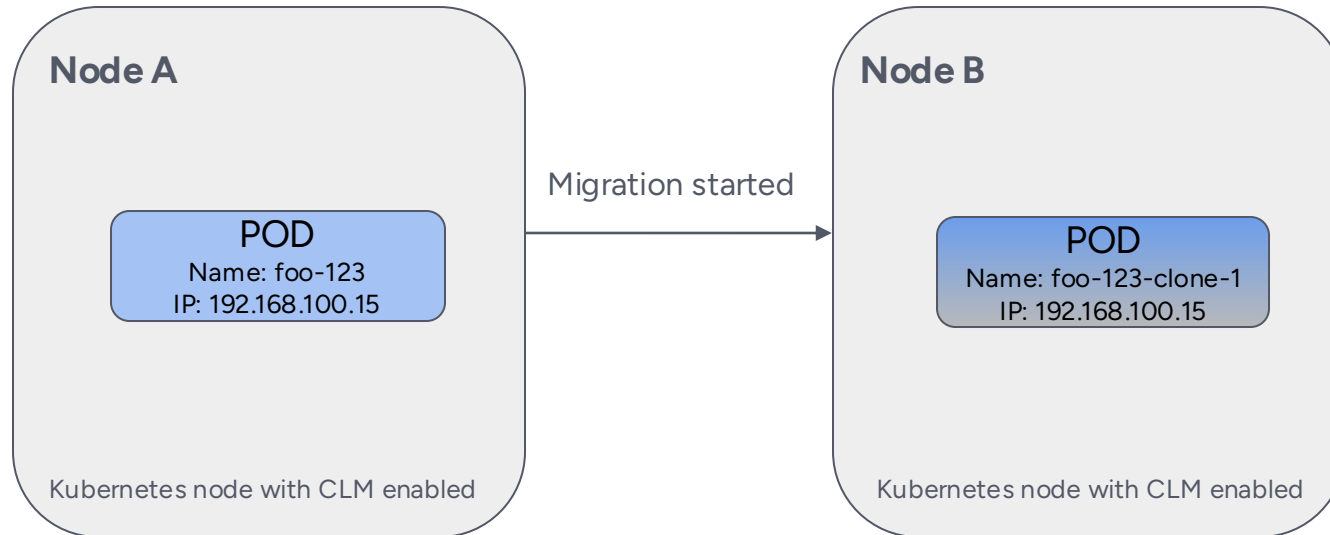
## Node B

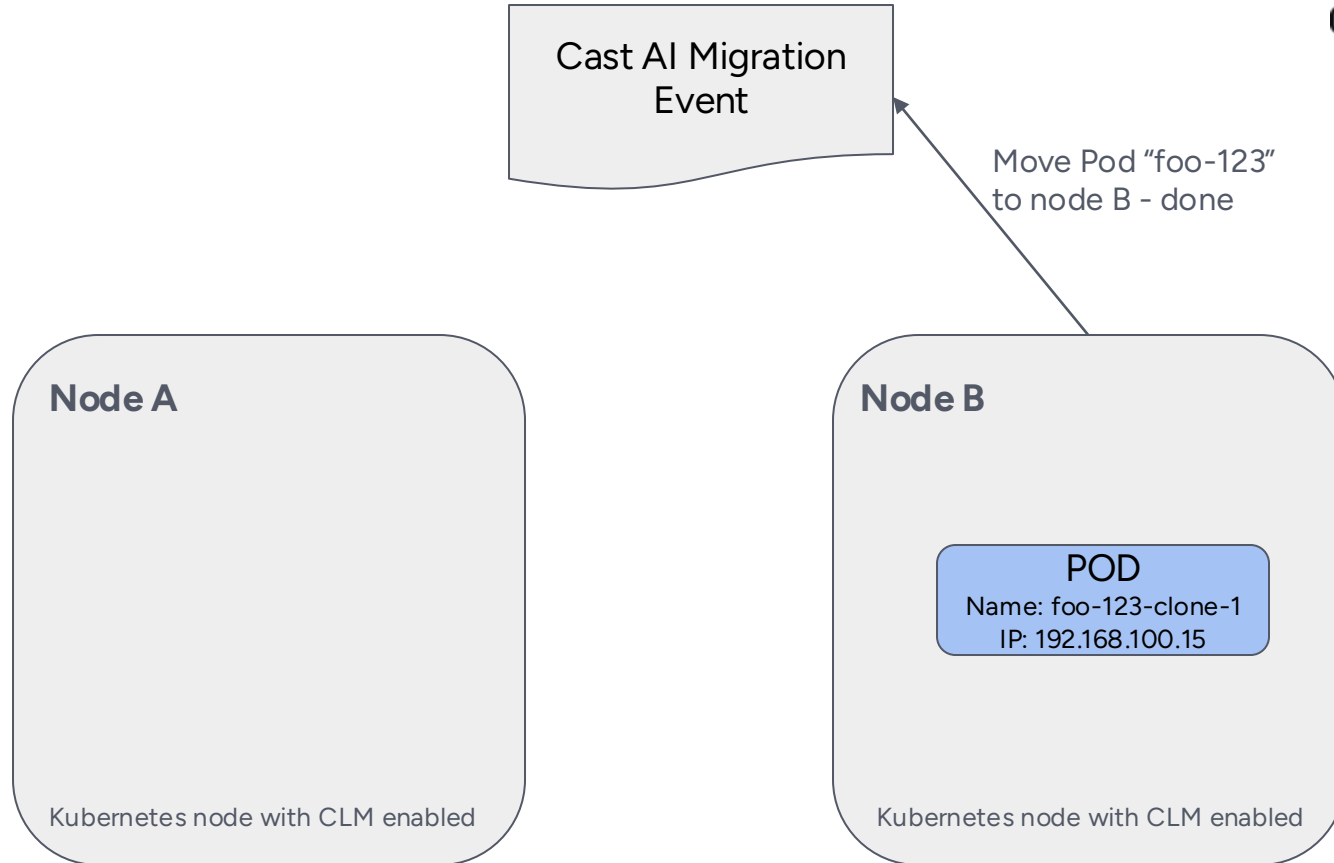
Kubernetes node with CLM enabled





## Cast AI Migration Event





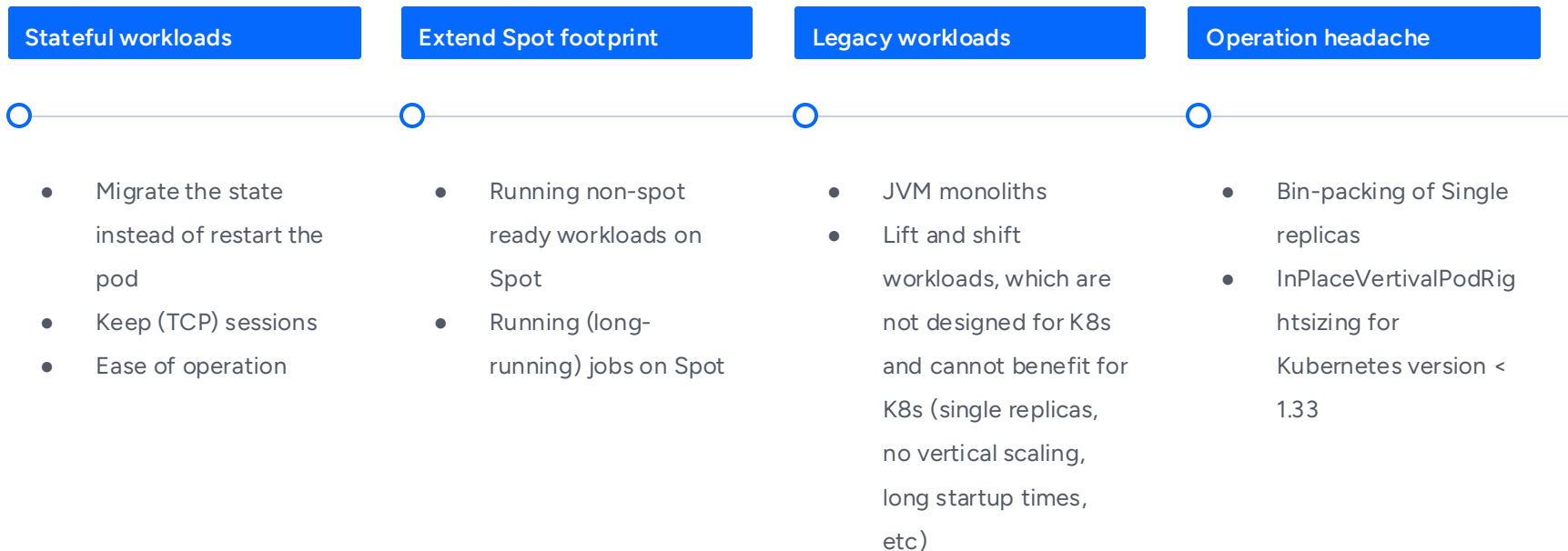
# No interruption, no downtime, no restart



But wait.... Why do i need this in  
Kubernetes environments?



# Some of the use cases



# When is it GA?

AWS EKS

NOW 🙌



GCP GKE

Q4 / 2025 😊



Azure AKS

2026 😞



**Let's live migrate a Minecraft server during  
we're playing**



Feeling lucky?

# Enter our raffle



 **CAST AI**

# THANK YOU



PING ME ANYTIME TO  
CHAT ABOUT  
ANYTHING

