Resources Optimization
in Kubernetes

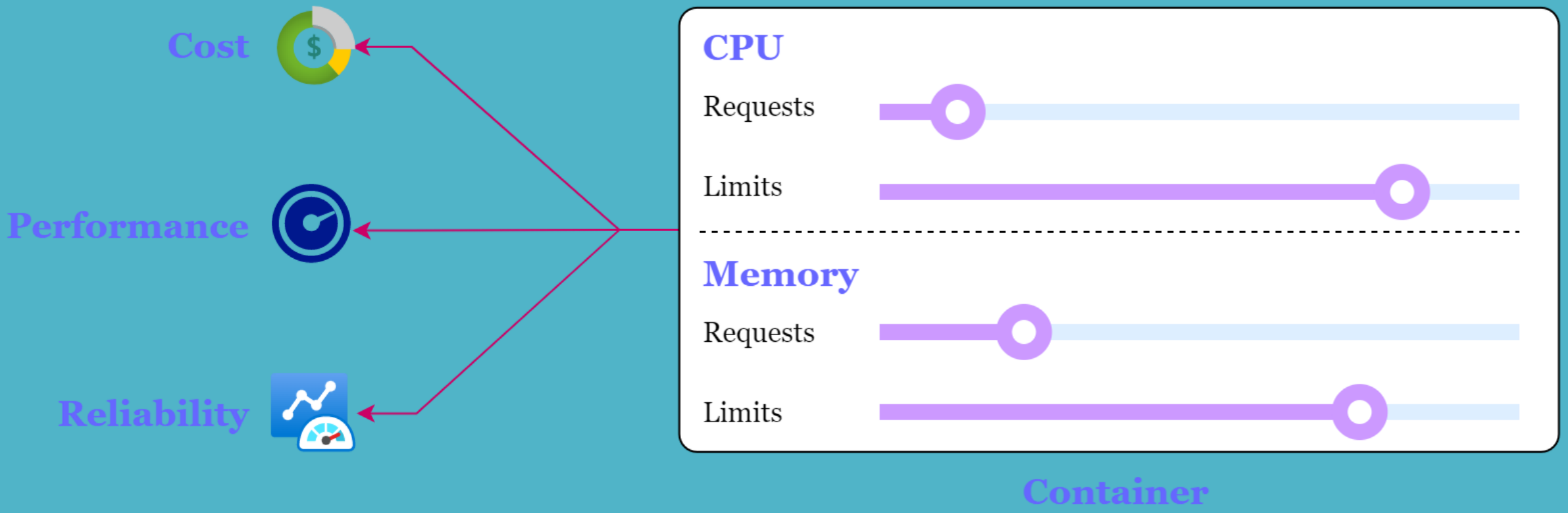
Trịnh Quốc Việt

VietOpenInfra Meetup 33rd – 8/2023

Content

- Requests & Limits
- Monitor resources
- How to optimize your resource
- Tools & demo

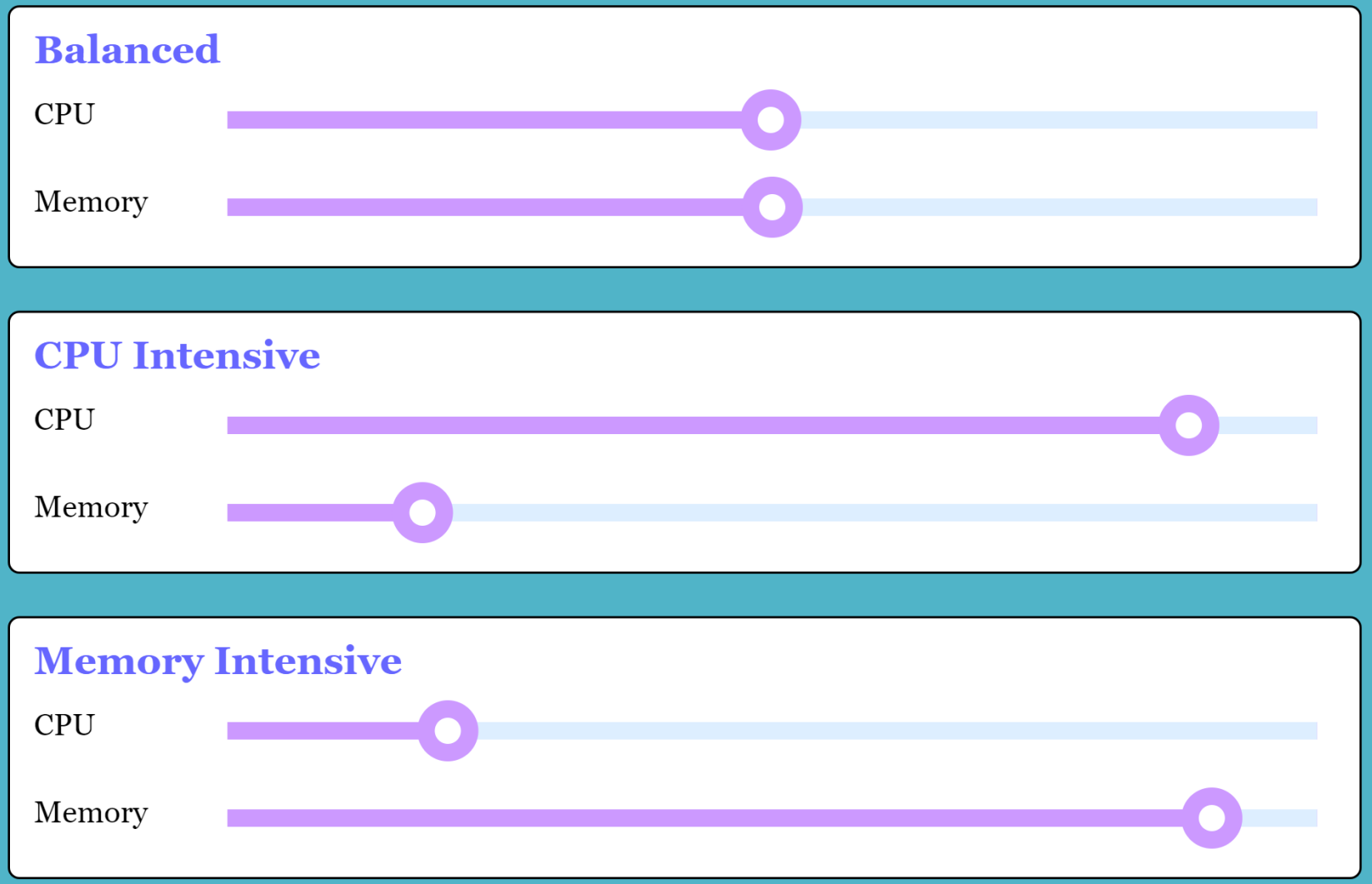
Limits & Requests



Resources Optimization in kubernetes

Limits & Requests

Application types

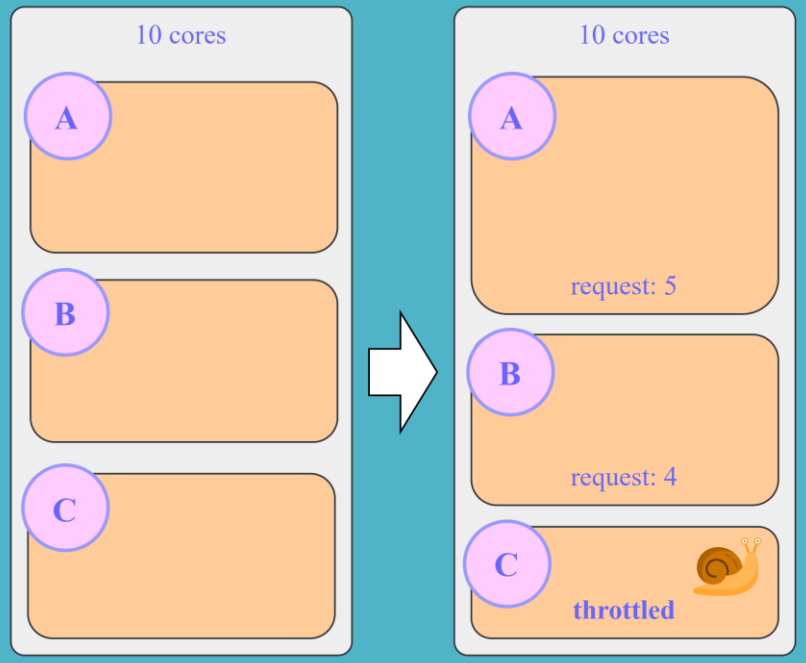


Resources Optimization in kubernetes

Limits & Requests

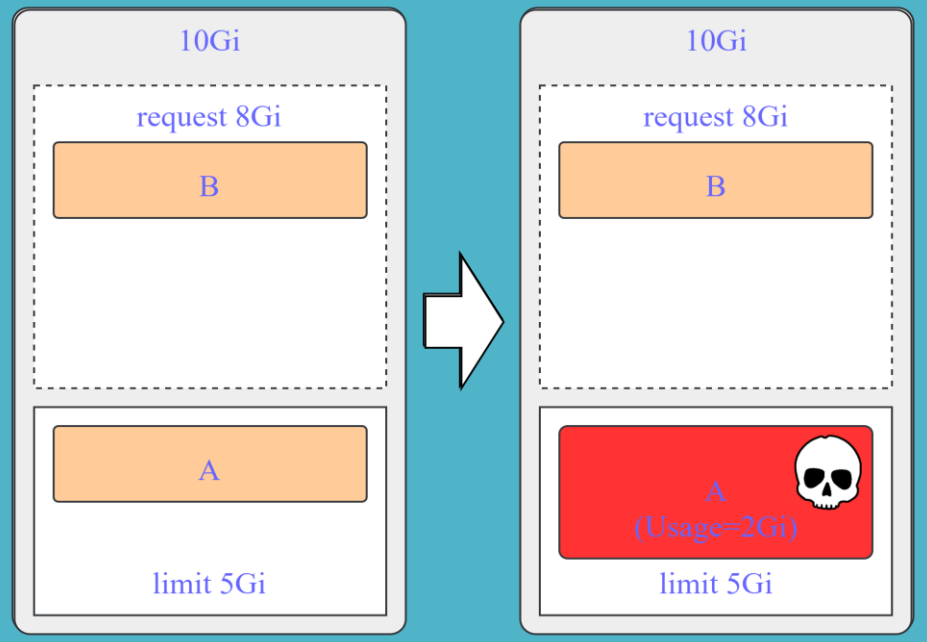
CPU

- computing processing time
- compressible resource
- throttling



Memory

- Measure in Mi, Gi..
- Non-compressible resource
- Out Of Memory killed

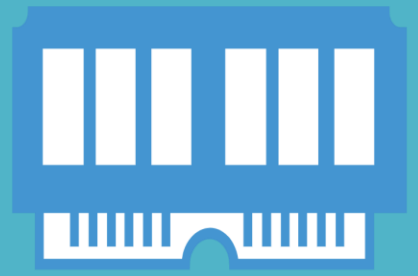


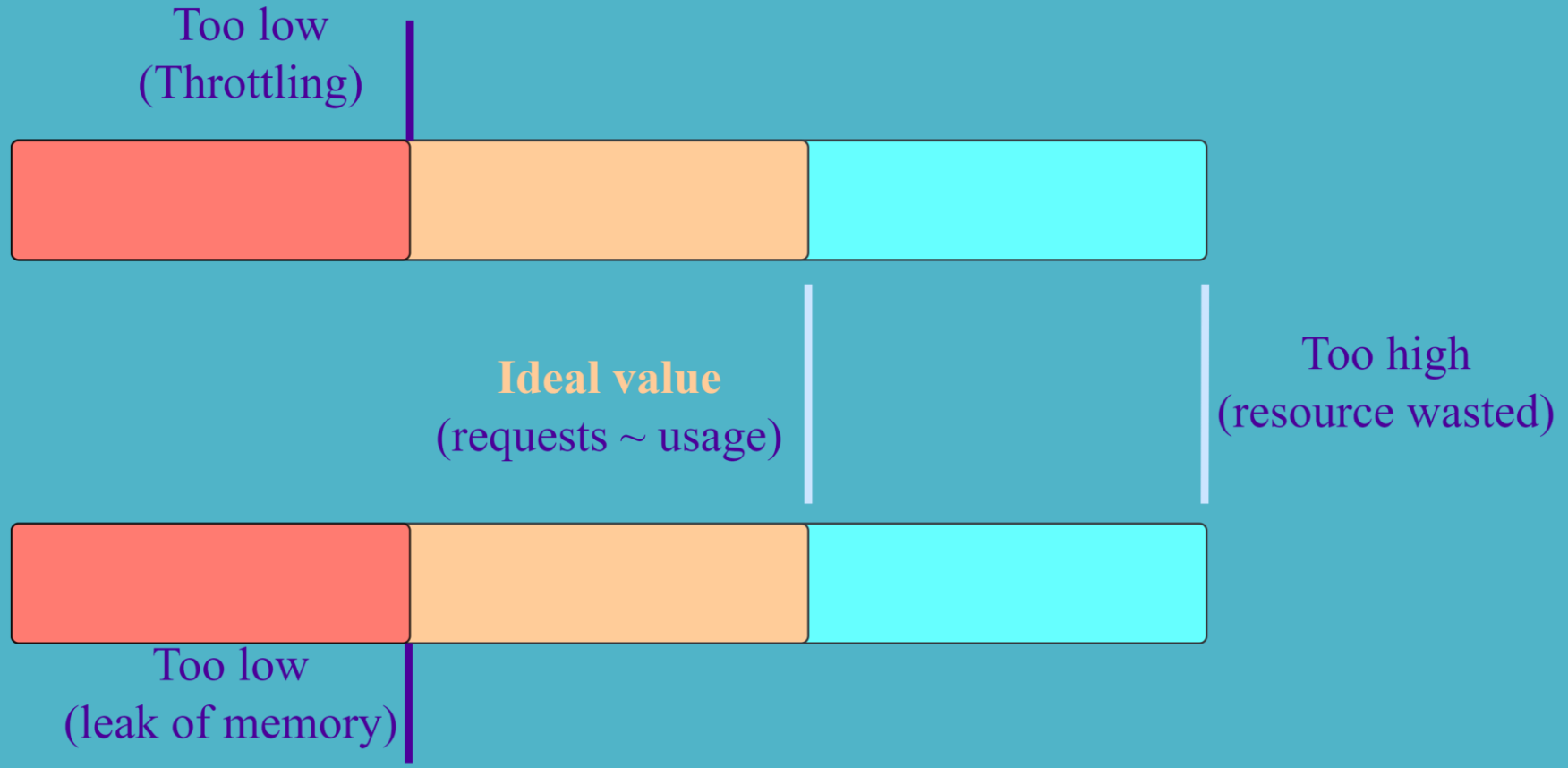
Limits & Requests

Requests

CPU

Memory

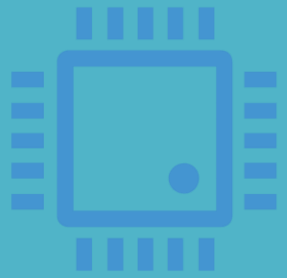




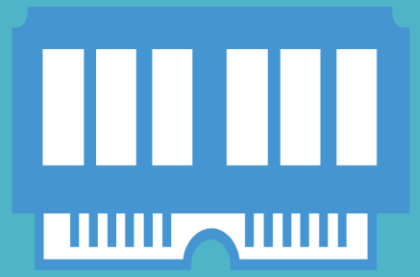
Limits & Requests

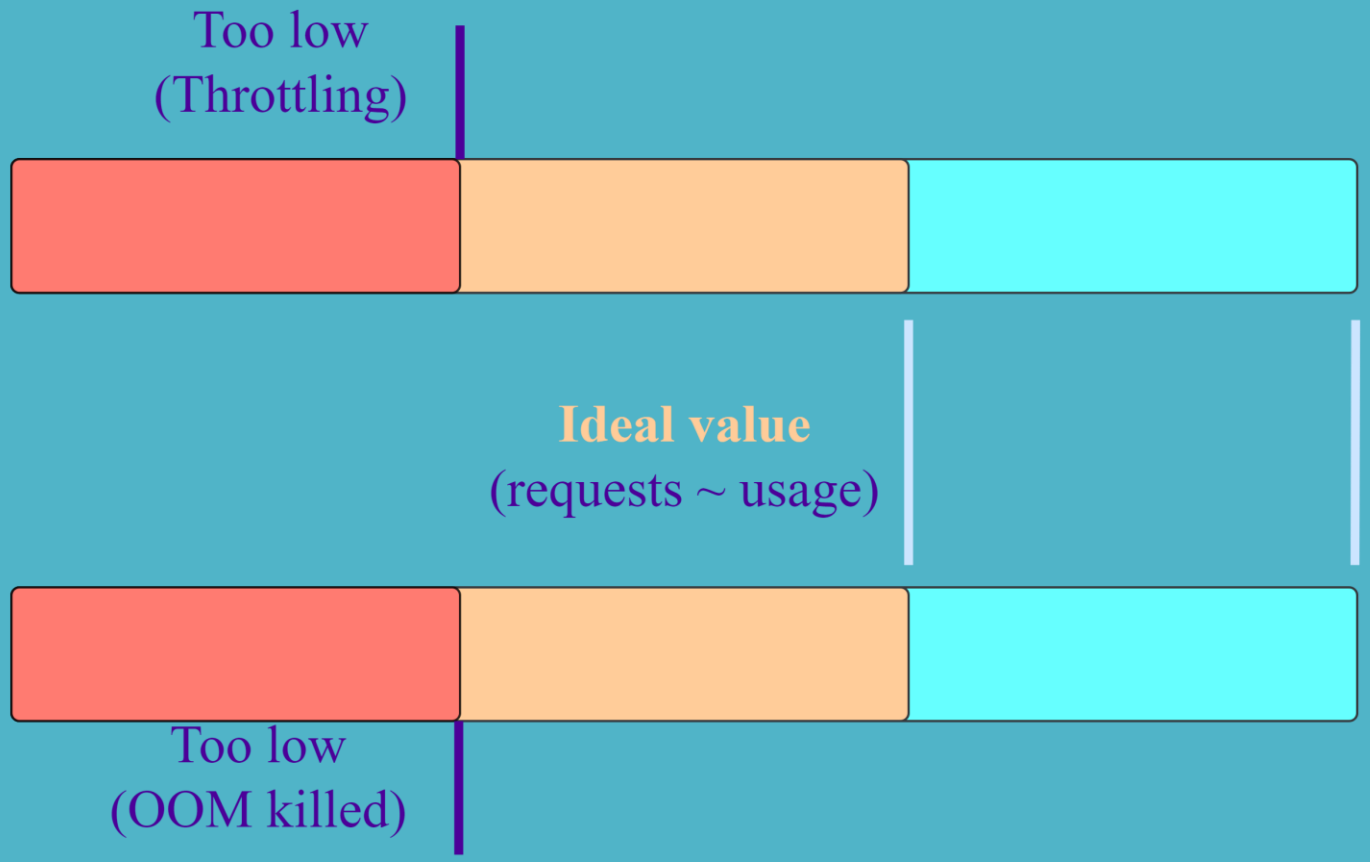
Limits

CPU

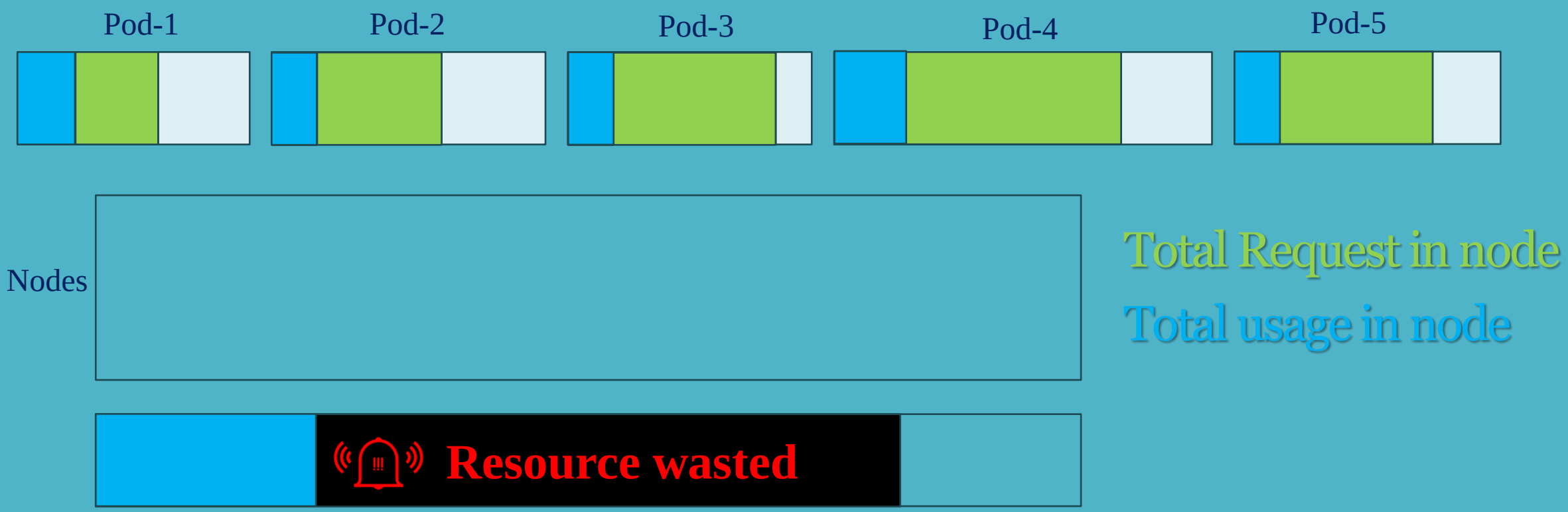


Memory

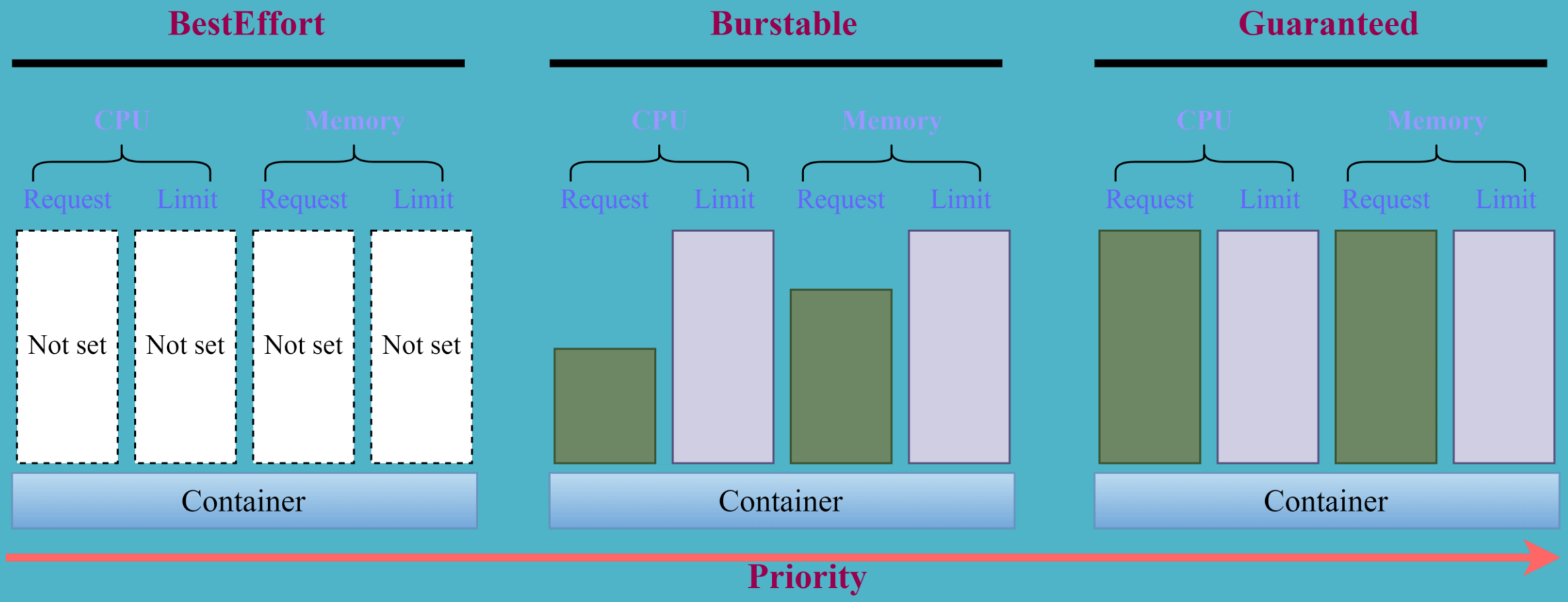




Limits & Requests - Scheduling



Limits & Requests - QoS



Monitor resources

What we need to monitor

- *CPU throttling*
- *OOM Kill*
- *Wasted resources*
- *Actual usages*

Monitor resources

Wasted resources

- *cluster/nodes level*
- *Namespace level*
- *Application level*
- *Container level*

Monitor resources

Wasted resources - node level

Cluster	Name	Role	Instance Type	Version	CC	MC	CPU	Memory (GiB)
cluster	staging 01	worker	rke2	v1.24.10+rke2r1	30	47.1 GiB	8.7 19.3 28.5	32.1 28.1 44.1
cluster	staging- 02	worker	rke2	v1.27.1+rke2r1	30	47.1 GiB	4.8 19.6 28.5	21.5 23.0 44.1
cluster	staging 03	worker	rke2	v1.24.10+rke2r1	30	47.1 GiB	11.7 18.4 28.5	29.8 29.3 44.1
cluster	staging- 04	worker	rke2	v1.24.10+rke2r1	30	47.1 GiB	9.8 21.5 28.5	26.9 32.6 44.1
cluster	staging 05	worker	rke2	v1.24.10+rke2r1	30	78.6 GiB	7.6 18.3 28.5	30.3 17.5 75.6
cluster	staging 01	worker	rke2	v1.24.10+rke2r1	12	23.5 GiB	5.0 2.6 10.5	16.1 2.3 20.5
cluster	staging- 02	worker	rke2	v1.24.10+rke2r1	12	23.5 GiB	3.8 2.6 10.5	14.6 2.3 20.5
cluster	staging 03	worker	rke2	v1.24.10+rke2r1	12	23.5 GiB	3.8 2.5 10.5	13.7 2.2 20.5

Monitor resources

Wasted resources - namespace level

All Namespaces

ID	A?	Cluster	P	CR	MR	CPU	Memory (MiB)	Cost	Slack Cost
-staging	✓	cluster	44	16.9	31.6 GiB	1.3 16.9	7,185 32,400	3.86	2.97
-staging	✓	cluster	58	14.8	35.4 GiB	0.55 14.8	17,474 36,300	4.16	2.25
longhorn-system	✓	cluster	72	13.15	0 Bytes	27.95 13.15	19,707 0	2.27	0.00
istio-system	✓	cluster	21	12.52	7.4 GiB	0.29 12.52	3,864 7,564	2.16	1.56
-staging	✓	cluster	18	10.9	16.2 GiB	0.29 10.9	3,831 16,640	2.01	1.54
kube-system	✓	cluster	44	6.975	7.3 GiB	5.2 6.98	33,128 7,480	1.40	0.47
-staging	✓	cluster	4	3.58	7.6 GiB	0.05 3.58	1,191 7,826	0.81	0.33
argocd	✓	cluster	10	3.4	3.4 GiB	4.06 3.4	2,812 3,512	0.59	0.27

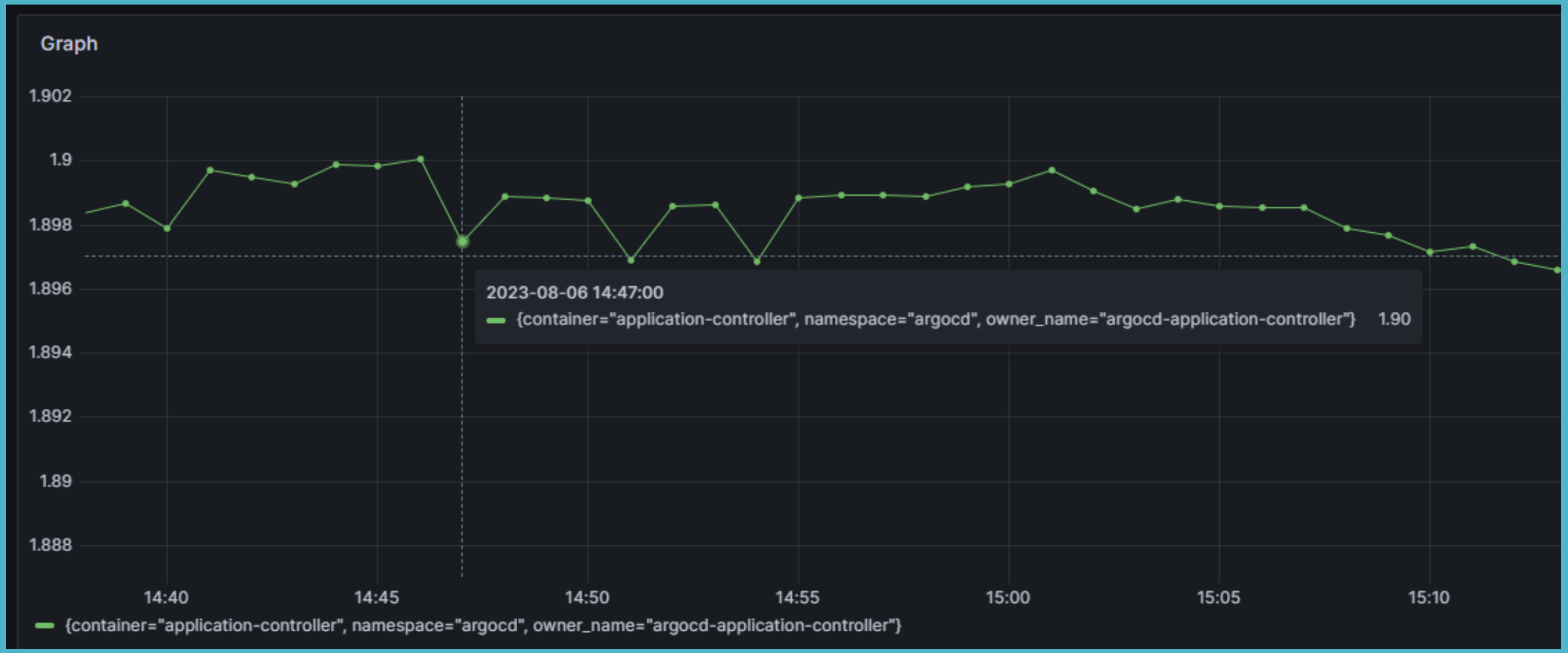
Monitor resources

Wasted resources - Application level

ID	Team	A?	C	P	CR	MR	CPU	Memory (MiB)
			1	117	20.656	8.0 GiB	25.05 20.66	37,750 8,220
istiod			1	10	10.0	4.9 GiB	0.1 10.0	1,701 5,000
cache-service			1	2	4.0	3.9 GiB	0.01 4.0	430 4,000
autoservice			1	4	4.0	3.9 GiB	0.09 4.0	953 4,000
lis-service			1	1	3.0	2.9 GiB	0.0 3.0	80 3,000
nifi			1	1	2.53	6.5 GiB	0.01 2.53	87 6,674
up-service			1	1	2.5	3.4 GiB	0.01 2.5	126 3,500
userservice			1	4	2.4	3.1 GiB	1.28 2.4	1,871 3,200
proj-service			1	1	2.1	1.5 GiB	0.01 2.1	203 1,500

Monitor resources

Container average CPU usages



How to optimize your resource

Always set requests & limits

Set the right requests & limits

Understand well your app

Monitor resource periodically

Tools & demo

Metrics server

kube-capacity

kube-resource-report

prometheus/grafana

Q & A

Ask me any questions

Thank you!