

edunext.co



www.edunext.co



OPEN/edX[®]

AT HOME

Conference _____ 2023
hosted by MIT





Crunching the K8S Numbers

Jhony Avella
Lawrence McDaniel

Some numbers and lessons we learned operating
OpenedX installations in Kubernetes

About me...

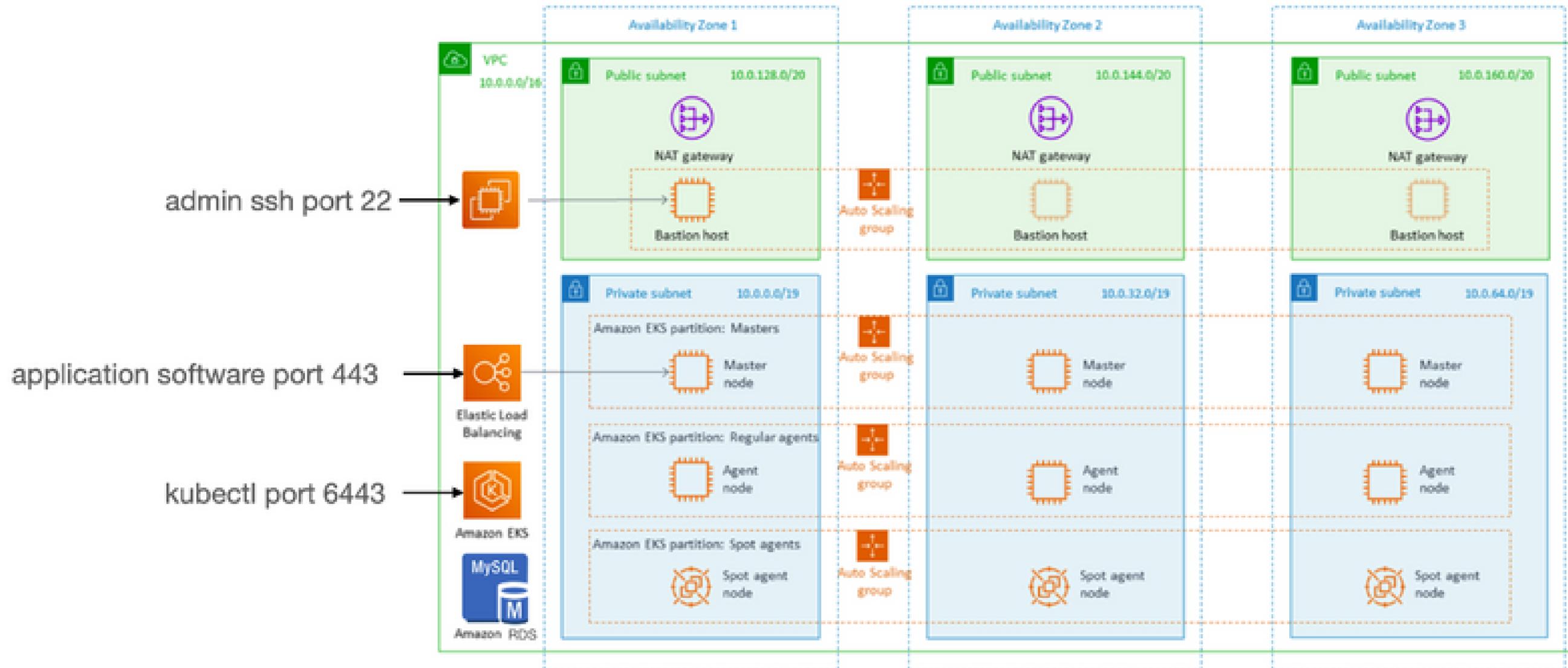
- OpenedX development
- Infrastructure As Code
- Kubernetes for OpenedX



AGENDA

1. Basic Kubernetes Stack.
2. Some numbers operating OpenedX in Kubernetes.
3. Testimony from one operator in the community.
4. Ongoing efforts from the Kubernetes Large Instance working group.
5. Edunext experience.

1. Kubernetes Stack

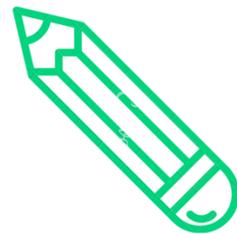


2. Some numbers on OpenedX operated with K8S



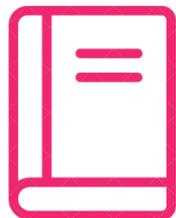
\$US 340-700

Cost of running an OpenedX instance for up to 100-200 concurrent users



+700K

Student records



+2M

Enrollment records

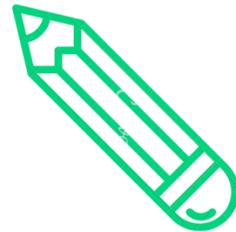


+30K

Courses



2. Some numbers on OpenedX operated with K8S



+ 40

Initiatives running in
Kubernees



20K

Users tested
concurrently in K8S



3. Lawrence testimony

openedx 2023 Kubernetes talk by Lawrence McDaniel

Share

OPEN
edx 2023

Full Stack
with lawrencemcdaniel.com

Watch on YouTube

The image shows a YouTube video player thumbnail. At the top left, there is a small profile picture of Lawrence McDaniel and the text 'openedx 2023 Kubernetes talk by Lawrence McDaniel'. To the right of this is a 'Share' button with a right-pointing arrow icon. Below the header is the 'OPEN edx 2023' logo, where 'OPEN' is in a red box, 'edx' is in a dark blue box, and '2023' is in a light blue box. The main title 'Full Stack' is written in large white font, with a red play button icon over the letter 'S'. Below the title, the word 'with' is written in a yellow, slanted font, followed by the website 'lawrencemcdaniel.com' in white. At the bottom left, there is a 'Watch on YouTube' button with the YouTube logo.

Why Kubernetes

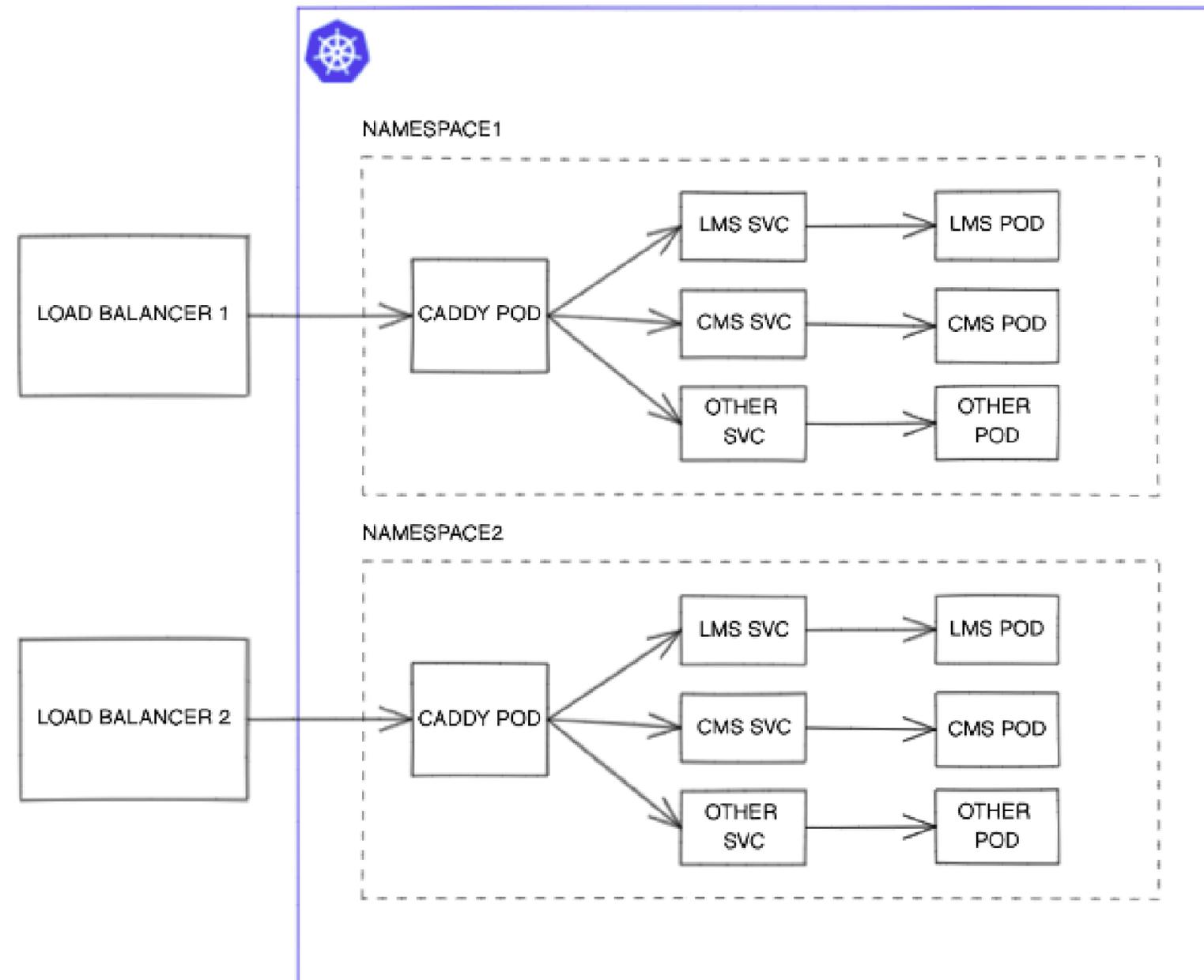
- Scaling
- Resource management
- High availability
- Portability
- Automation
- Security



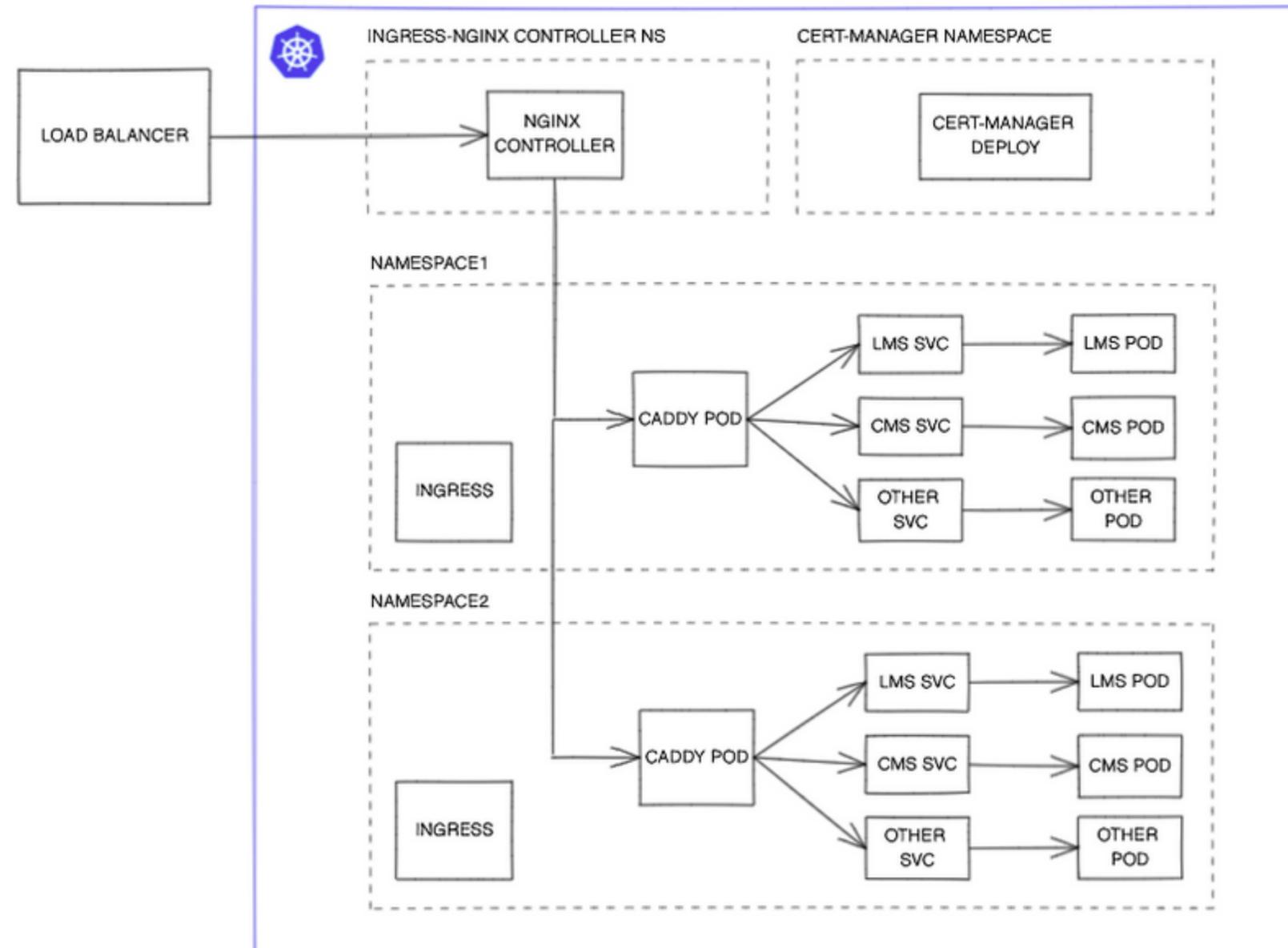
4. Ongoing efforts from the Kubernetes Large Instance working group.

- A central load balancer with automatic SSL certificates
- Automatic scaling of both nodes and pods
- Some level of monitoring tools
- Shared resources (ElasticSearch case)
- Publishing OpenedX K8S Harmony Helmchart

A central load balancer with automatic SSL certificates



A central load balancer with automatic SSL certificates



Automatic scaling of both nodes and pods

1.

POD-AUTOSCALING METHODS

- HPA
- VPA
- KEDA

2.

NODE-ATUSCALING METHODS

- Cluster autoscaler
- Karpenter

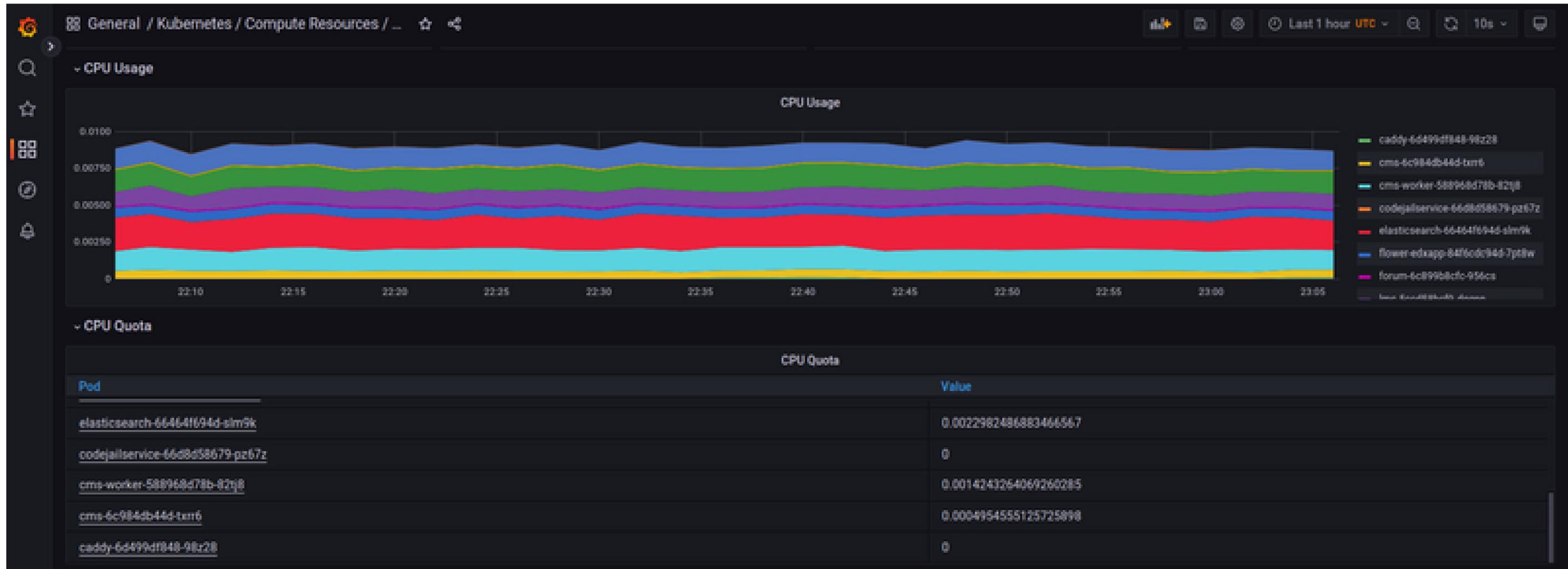
Automatic scaling of both nodes and pods - current approach

- HPA + any node-autoscaling method (pod-autoscaling implemented in <https://github.com/eduNEXT/tutor-contrib-pod-autoscaling/>).
- Automatic scaling of both nodes and pods
- In most cases, LMS, CMS, and workers are more CPU-intensive processes than memory-intensive.
- UWSGI can be tuned to get better performance.

Automatic scaling of both nodes and pods - current approach

```
resources:  
limits:  
  cpu: 1  
  memory: 2Gi  
requests:  
  cpu: 200m  
  memory: 1Gi
```

Monitoring tools



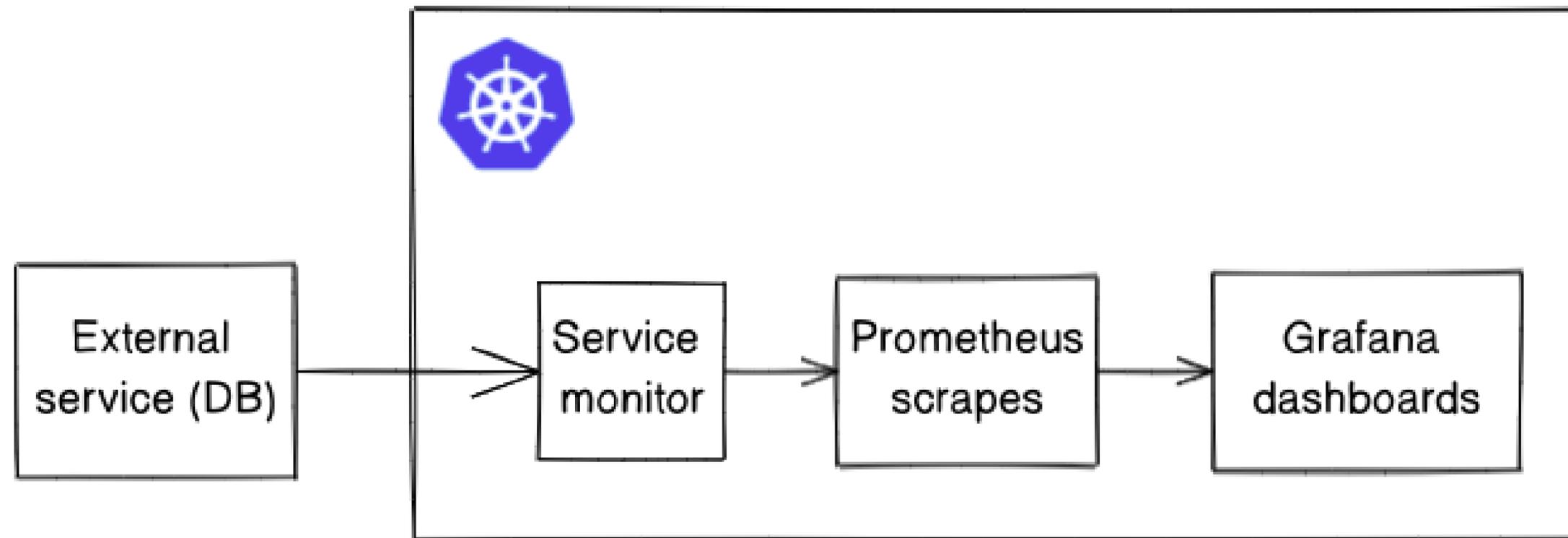
Monitoring tools

Kube-Prometheus Stack

- Prometheus
- Alertmanager
- Grafana
- Exporters:
 - node-exporter
 - kube-state-metrics
- Prometheus Operator

Monitoring in Kubernetes

Kube Prometheus Stack

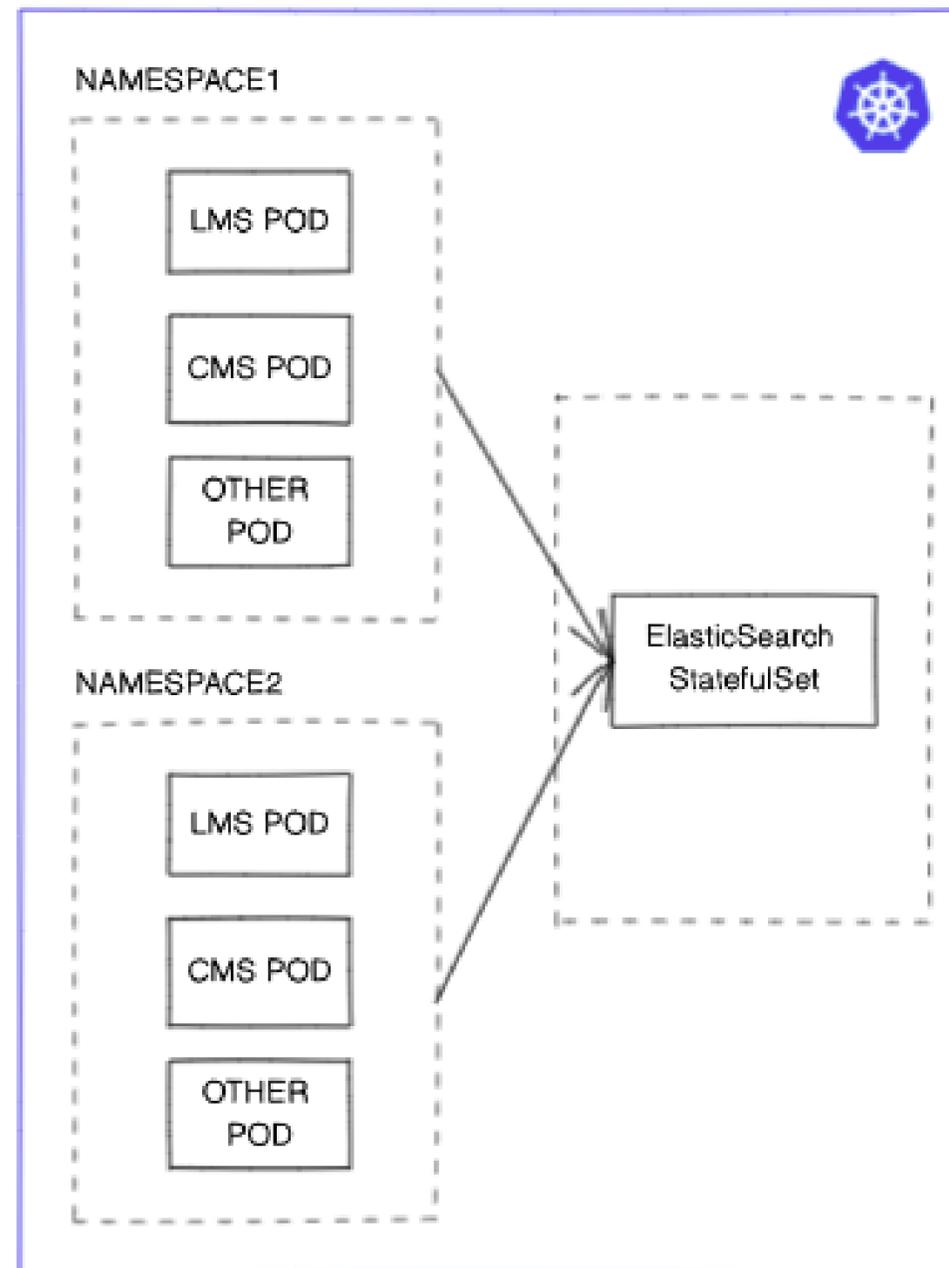


Monitoring in Kubernetes. Challenges



- Unified dashboard for OpenedX monitoring.
- Prometheus at scale (monitoring of multiple installations in a cluster, or multiple clusters with OpenedX installations)

Shared resources (ElasticSearch case)



5. Edunext experience :)

- GitOps
- Kubernetes on bare-metal
- Logging
- Common Issues



Large-scale deployments and GitOps

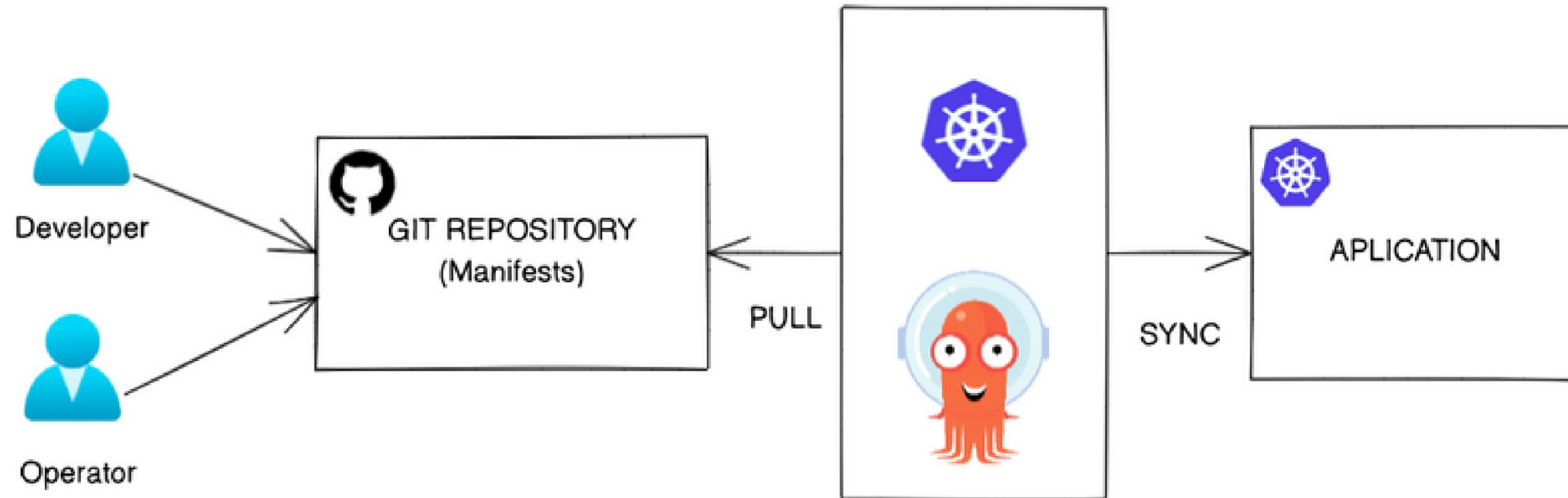


Large-scale deployments and GitOps



- Tutor offers a CLI-Based mechanism to update the cluster components.
- We wanted to keep a set of features from our old operation model:
 - A GUI to manage deployments
 - Authentication system to access the GUI
 - Functionalities like Rollbacks, resource pruning, logging, etc.
- Gitops-based solution.

Large-scale deployments - ArgoCD



Large-scale deployments - ArgoCD



Most common **ArgoCD** resources:

- **Projects:** allows to group resources and create access roles to those
- **Application:** A set of resources defined in a git repository that is synchronized to one or multiple clusters
- **Secrets:** stores credentials to connect to clusters, repositories or to define user Tokens.

Large-scale deployments - ArgoCD

Applications / [toc-prod.boston.jupiter](#)

APPLICATION DETAILS PODS

APP DETAILS

APP DIFF

SYNC

SYNC STATUS

HISTORY AND ROLLBACK

DELETE

REFRESH



Log out

APP HEALTH

Healthy

CURRENT SYNC STATUS

Synced To main (229b6a0)

MORE

Author: Jhony Avella <jhony.avella@edunext.co> -
Comment: chore: using 1 pod but 8 workers

LAST SYNC RESULT

Sync OK To 229b6a0

MORE

Succeeded 8 hours ago (Sun Mar 26 2023 14:04:30 GMT-0500)
Author: Jhony Avella <jhony.avella@edunext.co> -
Comment: chore: using 1 pod but 8 workers

GROUP BY: **NODE**

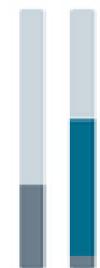
UNSCHEDULABLE



ip-10-0-1-117.ec2.internal

node

Kernel Version: 5.15.0-1031-aws
OS/Arch: linux/amd64



CPU MEM



PODS



ip-10-0-2-97.ec2.internal

node

Kernel Version: 5.15.0-1031-aws
OS/Arch: linux/amd64



CPU MEM



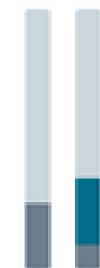
PODS



ip-10-0-3-244.ec2.internal

node

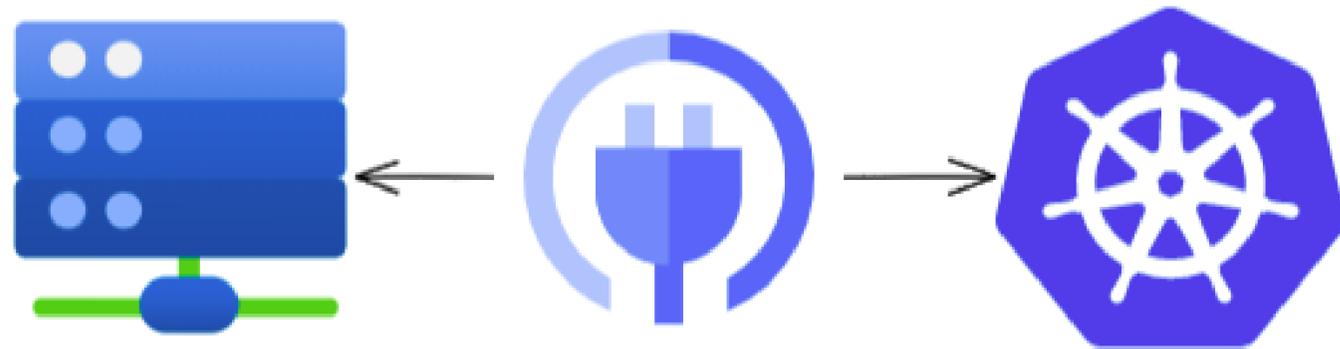
Kernel Version: 5.15.0-1031-aws
OS/Arch: linux/amd64



CPU MEM



PODS



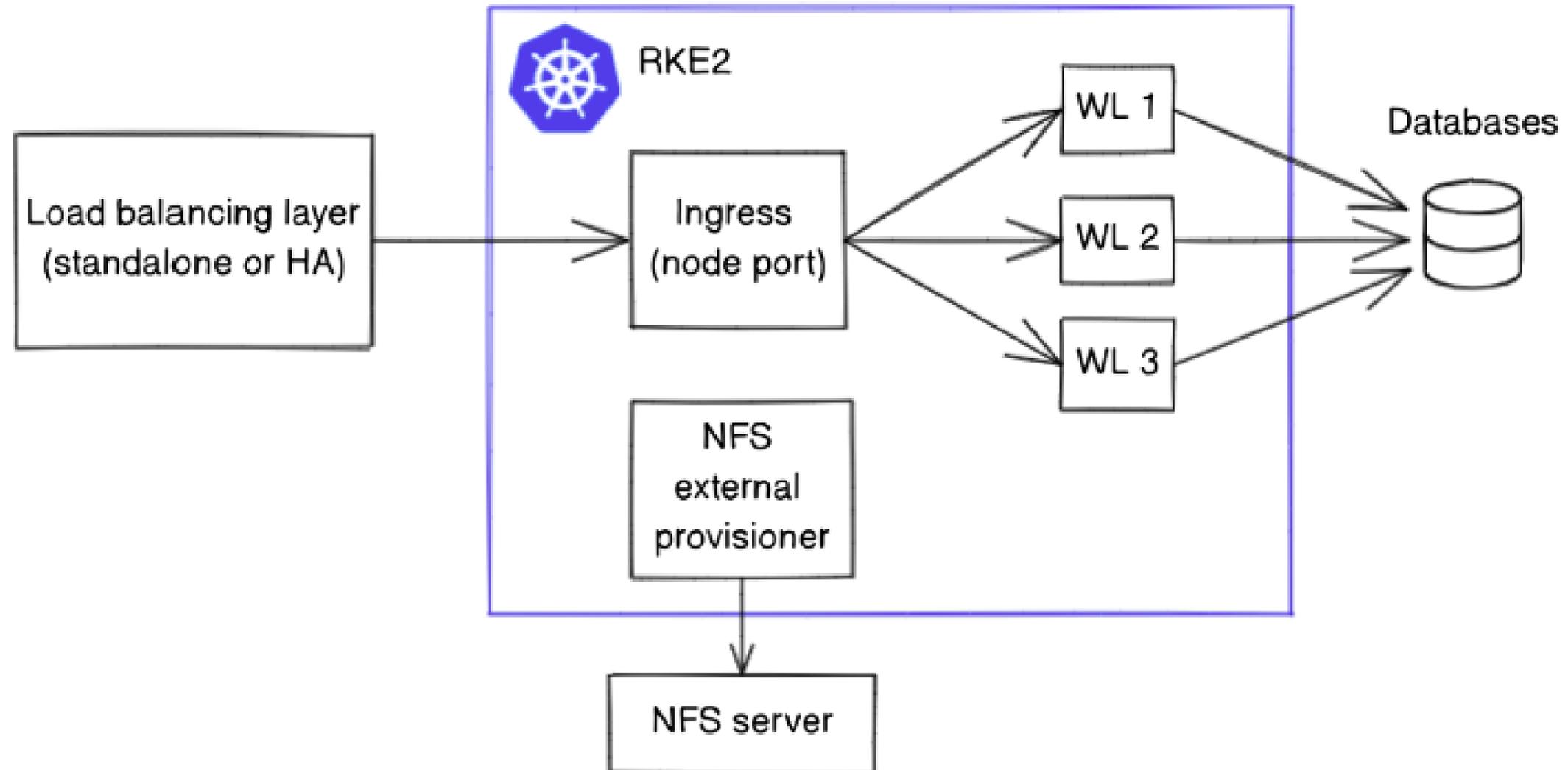
**Kubernetes in
bare-metal.**



Kubernetes in bare-metal. Challenges

- Load balancing layer
- Picking a tool to install a Kubernetes distribution
- Control plane and ETCD roles
- Storage controller.

Kubernetes in bare-metal



Kubernetes in bare-metal

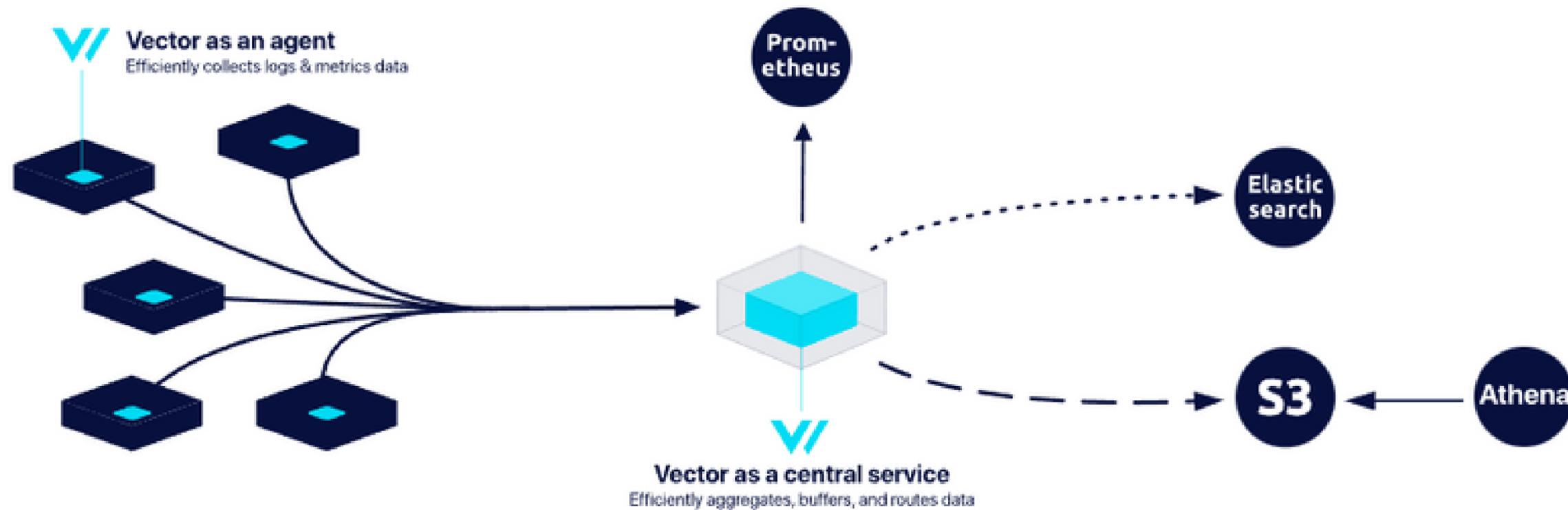
- **RKE** as a Kubernetes provider:
 - Minimum 2 control-plane nodes m6 large
 - Minimum 3 ETCD nodes m6 large to tolerate the failure of one of the nodes.
 - Minimum 3 nodes m6.large (24 GB - 6 CPU)
- Additional load balancer layer (SSL termination inside or outside the cluster). Internal K8S service running as node port.
- An NFS server is required to connect to the cluster
NFS controller



Logging in Kubernetes



Logging in Kubernetes Vector



Logging in Kubernetes

- It's possible to activate control-plane logging in cloud providers, however, in our experience, it leads to important extra-costs
- Cloud providers generally execute log deletion in the K8S clusters. This is not the case for bare-metal servers.
- When using Cloudwatch to store logs, set data retention to save money.
- Logs grouping in a multitenancy cluster is still a challenge.

Common issues operating in Kubernetes.

- Too many 502 errors
 - Pod initialization
 - Pod graceful termination
 - CONN_MAX_AGE configuration
- The IP of the client is not kept
 - Proxy protocol
 - Usage of a L7 reverse proxy
- Async tasks are not completed
 - terminationGracePeriodSeconds setting

Links:

- <https://github.com/openedx/openedx-k8s-harmony>

Thank you!