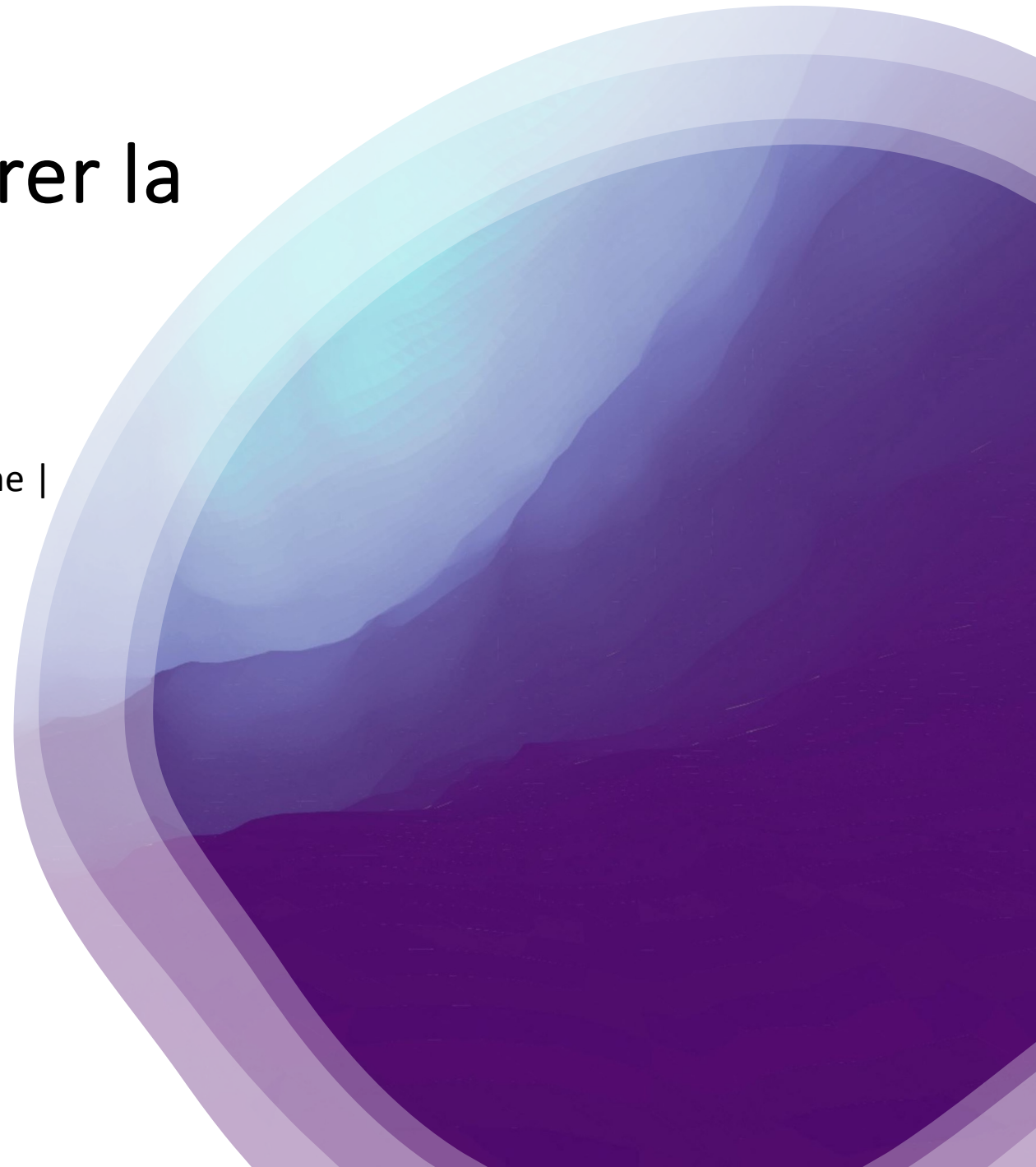


# SAE5.DevCloud.03: Orchestrer la conteneurisation d'une application

PECOUT Lucas | KEELY Joshua | MAUREL Jilian | HACINI Sofiane |  
KERADEC-DUHARD Iwen



# Sommaire

Introduction

I – Contexte / Objectifs

II.1 – Objectif projet

II.2 – Approche DevOps

II – Organisation de notre gestion de projet

II.1 – Méthodes agiles

II.2 - Répartition des tâches

III – Conception

III.1 - Architecture réseau virtualisé

III.2 - Architecture réseau conteneurisé

III.3 - Architecture microservices

IV.4 – Fonctionnalités supplémentaires

IV - Déploiement

IV.1 - Déploiement réseau virtualisé

IV.2 - Développement application et Dockerfile

IV.3 - Déploiement cluster Kubernetes

IV.4 – Mise en place du monitoring

V – Retours d'expérience

V.1 – Problèmes rencontrés

V.1 – Notre ressenti

Conclusion

# Introduction

---

**Le meilleur site E-Design**

**Nouveauté 2023**

En savoir plus

Produits

# 1) Contexte / Objectifs

---



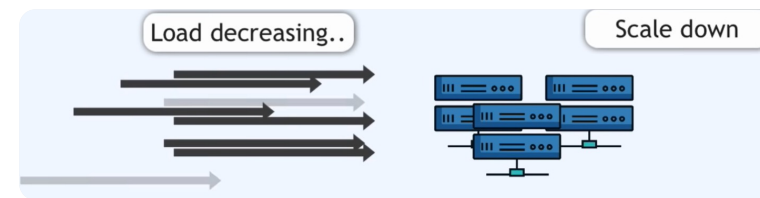
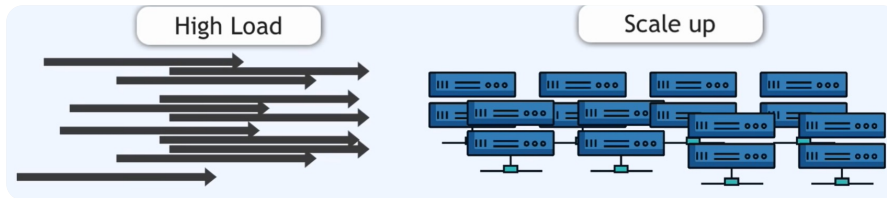


## II.1 – Objectifs projet

- Développer une plateforme e-commerce présentable à un expert DevOps et un client
- Apprendre à utiliser des outils DevOps, à travailler en équipe, apprendre à être proche du client et de ses besoins
- **Développement d'Application** : Créer une application en utilisant une architecture moderne et flexible, basé sur un outil d'orchestration de conteneurs très puissant

### *Quelles fonctionnalités les outils d'orchestrations offrent-ils aux clients?*

- **Adaptabilité/ Flexibilité** : Assurer que l'application puisse s'adapter aux pics d'activité.



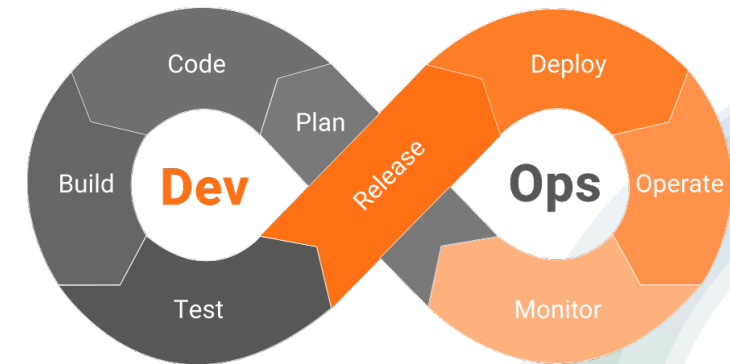
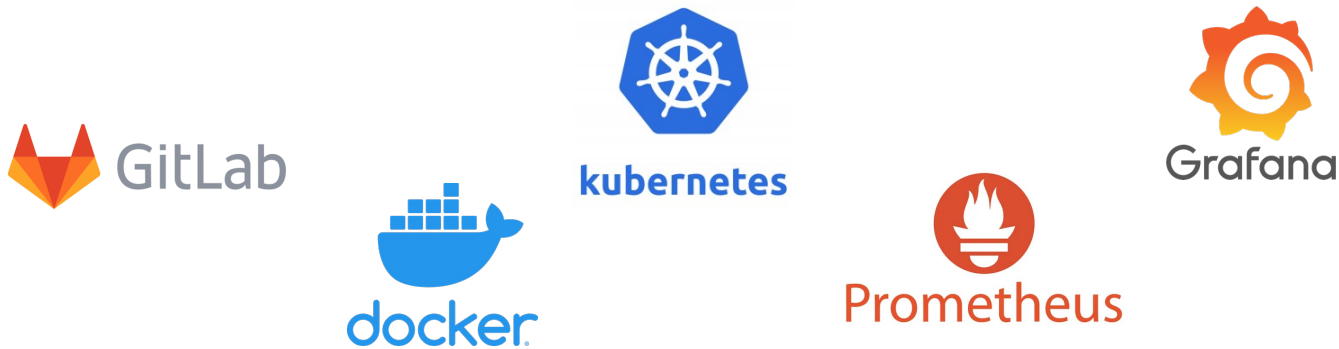
- **Efficacité** : Meilleure gestion des ressources, réduisant les coûts et améliorant les performances.
- **Fiabilité** : Moins de risques de panne, chaque partie de l'application pouvant fonctionner de manière autonome.
  - > Disaster Recovery: L'infrastructure a un mécanisme de backup des données et peut les restaurer rapidement



## II.2 – Approche DevOps

### Que veut dire DevOps?

- Contraction des mots "Développement" et "Opérations"
- **Objectifs**: Accélérer le cycle de développement et de déploiement des logiciels tout en maintenant une haute qualité et une stabilité opérationnelle
- **Métiers**: Admin/Ingénieur/Architecte DevOps, Sécurité DevOps (DevSecOps), Analyste de Données DevOps
- **Outils utilisés**: correspondant aux différentes étapes de déploiement



### Comment est liée cette approche aux autres métiers de l'entreprise?

- Collaboration entre les **équipes de dev** et les **équipes opérationnelles** qui gèrent l'infrastructure et les systèmes  
=> L'approche agile encourage une **culture d'entreprise dynamique**, centrée sur la **collaboration**, l'**adaptabilité** et l'innovation continue, affectant positivement de nombreux aspects de l'entreprise



## II – Organisation de notre gestion de projet

---



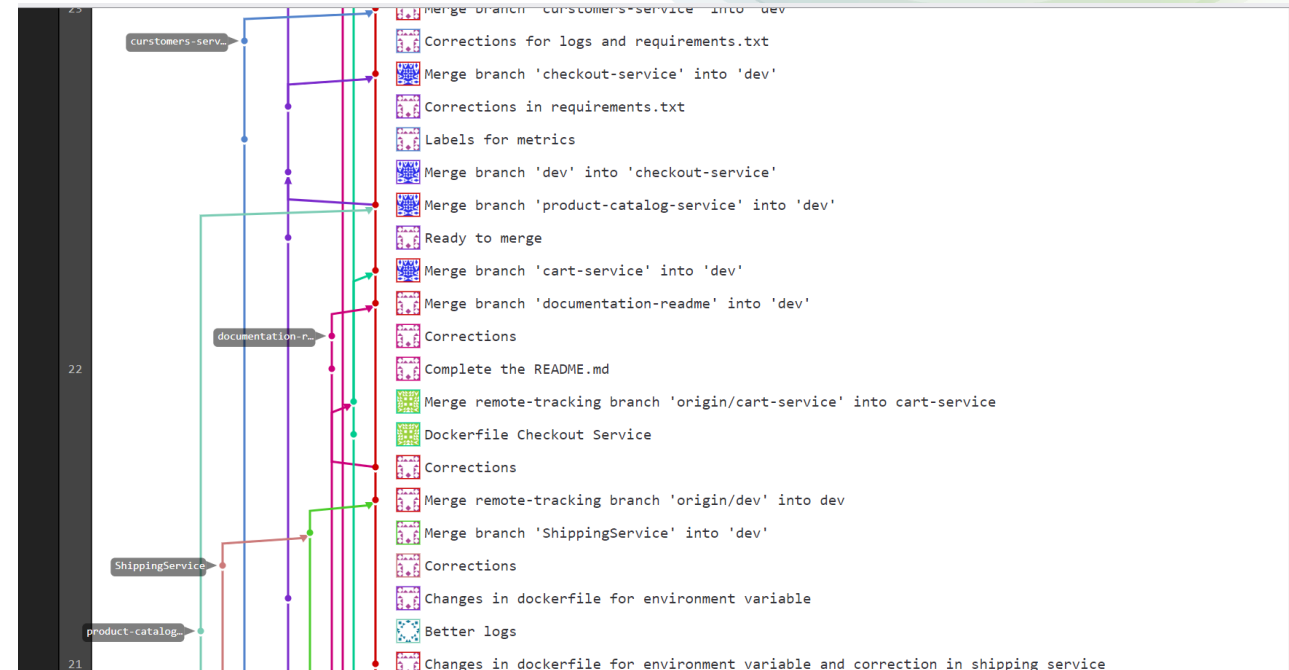
## II.1 – Méthodes agiles

### Notion de **Sprints**

- Période définie pendant laquelle une équipe travaille pour compléter un ensemble spécifique de tâches ou pour livrer une partie fonctionnelle du projet
- 3 sprints de 20h de travail

Sprint	Période	Sprint Planning	Sprint Review
1	24 oct. au 17 nov.	24 oct. à 8h	17 nov. 13h30
2	20 nov. au 11 déc.	17 nov. 13h30	12 déc. 8h
3	13 déc. au 26 jan.	26 jan. à 13h30	Soutenance

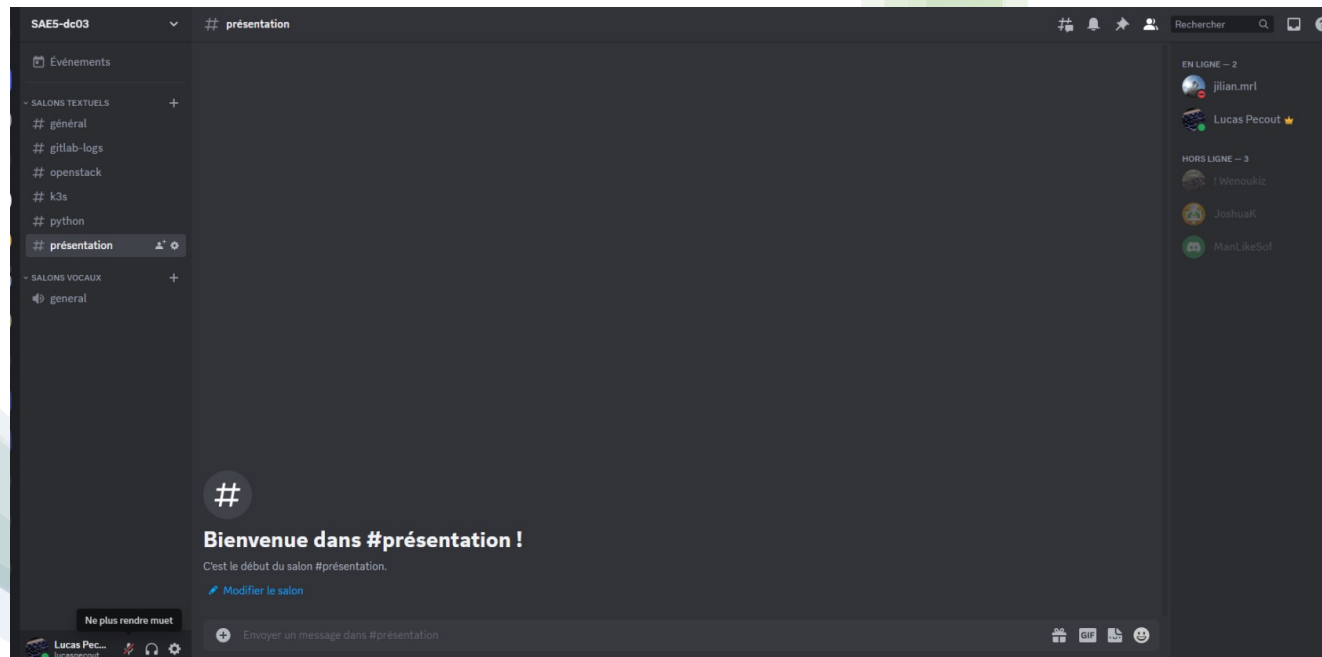
### Contrôle de version avec **Gitlab**





## II.2 - Répartition des tâches/Organisation

- Début de **Sprint 1**: Ajout de chaque tâche sur Gitlab
- Répartition/Attribution des tâches suivant les points fort de chacun
- **Communication interne**: Explications des différentes tâches que nous avons accomplies entre nous
- Réunion en fin de chaque journée (+4h de TP) pour faire le point
- **To Do > In progress > Done**
- Mise en place d'un salon Discord pour pouvoir communiquer en dehors des cours





# III.1 - Architecture réseau virtualisé

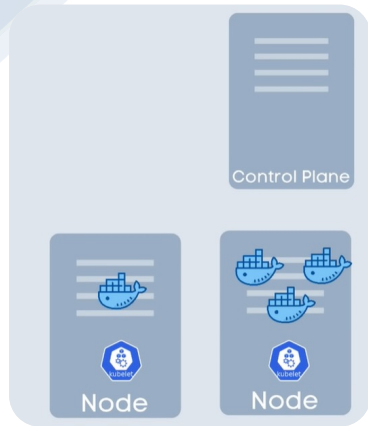
Application déployée sur un cluster **Kubernetes**: **1 Master node** et **2 Worker Node**

Qu'est ce qui tourne sur le Master Node ?

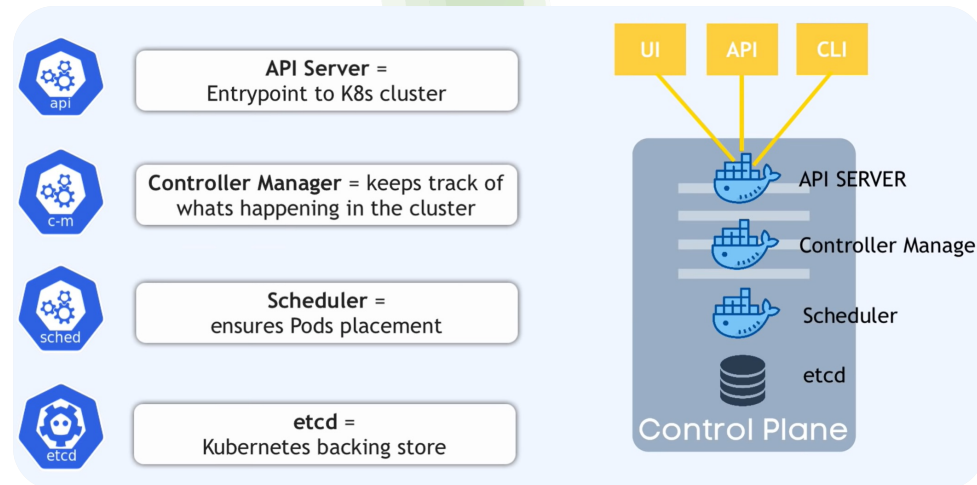
Plusieurs **processes** importants qui sont nécessaires au bon fonctionnement du cluster Kubernetes!



<insert OpenLens screenshot>



Les services tournent majoritairement sur les **workers Nodes**



- Notre cluster Kubernetes est installé sur une infrastructure **OpenStack**



UGA-Net: 152.77.180.0/24

## Pare-feu

## Redirection de ports:

WAN	TCP	*	*	Ce Pare-feu	3389 (MS RDP)	192.168.115.34	3389 (MS RDP)	xrdp
WAN	TCP	*	*	Ce Pare-feu	22 (SSH)	192.168.115.10	22 (SSH)	ssh
WAN	TCP	*	*	Ce Pare-feu	8080	192.168.115.10	8080	traefik
WAN	TCP	*	*	Ce Pare-feu	80 (HTTP)	192.168.115.10	80 (HTTP)	
WAN	TCP	*	*	Ce Pare-feu	8081	192.168.115.10	5432	Postgres
WAN	TCP	*	*	Ce Pare-feu	8082	192.168.115.10	6443	K8s - Lens

## Règles LAN

IPv4 *	LAN net	*	*	*	*	*	*	Default allow LAN to any rule
IPv6 *	LAN net	*	*	*	*	*	*	Default allow LAN IPv6 to any rule
IPv4 ICMP	LAN net	*	*	Ce Pare-feu	*	*	*	ICMP LAN

## Règles WAN

IPv4 TCP	*	*	Ce Pare-feu	443 (HTTPS)	*	*	*	
IPv4 TCP	*	*	192.168.115.34	3389 (MS RDP)	*	*	*	xrdp
IPv4 TCP	*	*	192.168.115.10	22 (SSH)	*	*	*	ssh
IPv4 TCP	*	*	192.168.115.10	80 (HTTP)	*	*	*	
IPv4 TCP	*	*	192.168.115.10	5432	*	*	*	Postgres
IPv4 TCP	*	*	192.168.115.10	6443	*	*	*	K8s - Lens

## GW UGA

IP address: 152.77.180.1

## WAN interface (vtnet0)

IP address: 152.77.180.51

MAC address: fa:16:3e:f5:8b:d7

## LAN interface (vtnet1)

IP address: 192.168.115.254

MAC address: fa:16:3e:19:97:9b

OPNSense

IP address: 192.168.115.12

OS: CentosStream8

Flavor: m3.small  
RAM: 1GBRôles: - DHCP  
- Pare-feu  
- Routeur

DNS server: 152.77.1.22

LAN network: 192.168.115.0/24

Master-Node

IP address: 192.168.115.10  
OS: CentosStream8Flavor: m3.small  
RAM: 4GB

Worker-Node1

IP address: 192.168.115.11  
interface: ens7  
OS: CentosStream8Flavor: m4.tiny  
RAM: 2GB

Worker-Node2

IP address: 192.168.115.12  
interface: ens7  
OS: CentosStream8Flavor: m4.tiny  
RAM: 2GB

UBUNTU

IP address: 192.168.115.34  
OS: UbuntuFlavor: m43.large  
RAM: 2GB

Roles: OPNSense web GUI

Kube-files

IP address: 192.168.115.29  
OS: TrueNASFlavor: m3.small  
RAM: 1GBDescription: Utilisé comme NAS  
virtuel



## III.2 - Architecture conteneurisée

- Chaque service déployé dans un/plusieurs **pods**

-  **Namespace** rapidtradeco

-  **Ingress/Gateway**: redirection de routes

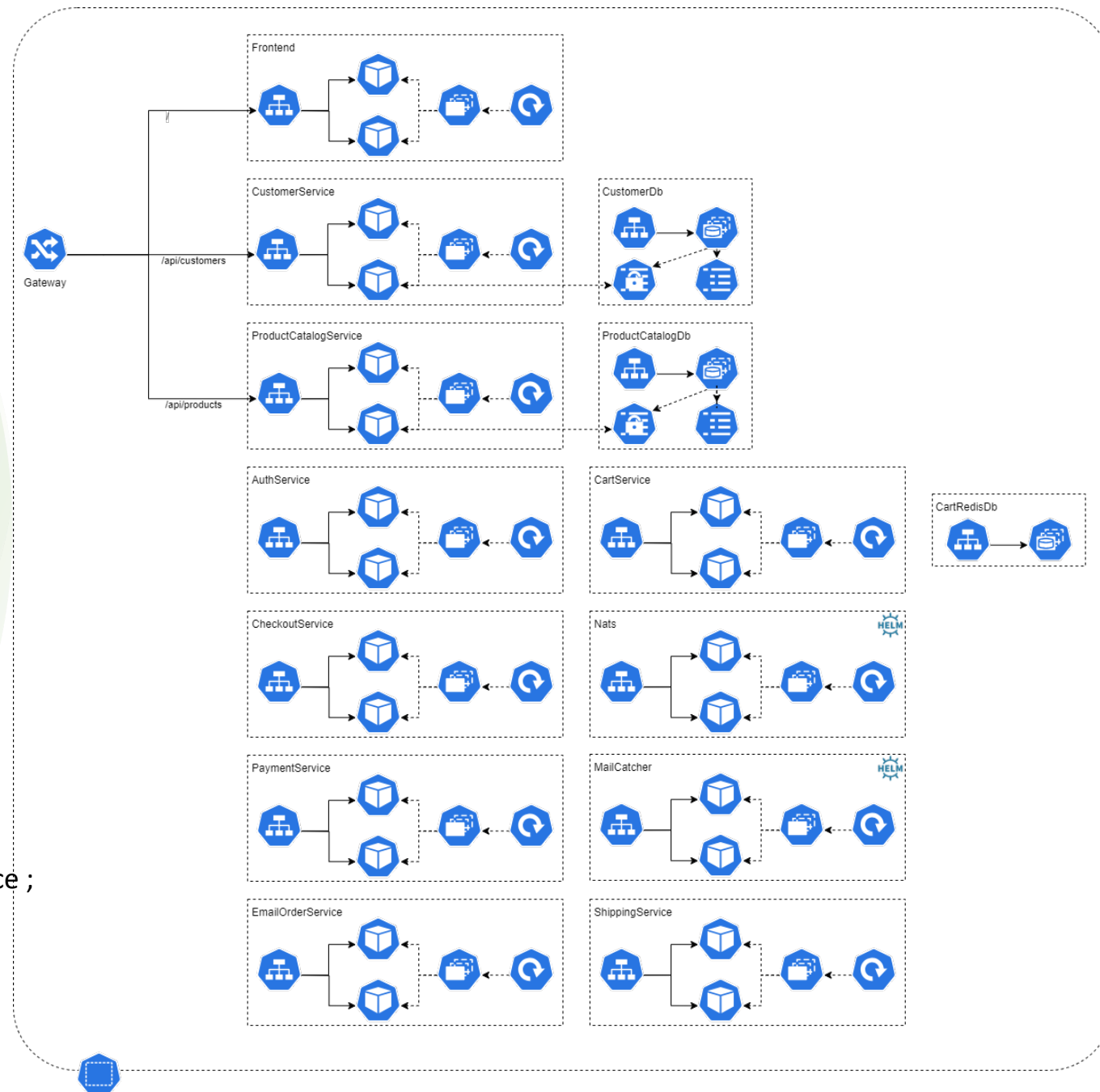
```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: frontend-ingress
  namespace: rapidtradeco
spec:
  rules:
  - http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: frontend-service
            port:
              name: front
```

- **Routes:**

/ correspond à l'application Web de la boutique

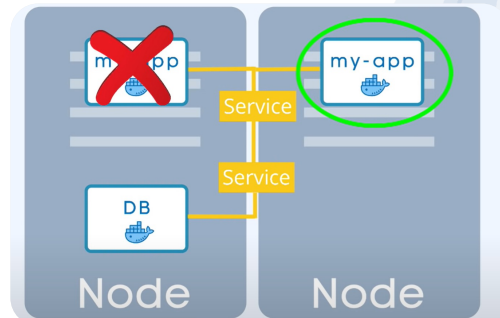
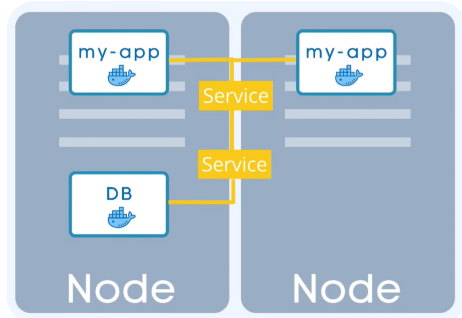
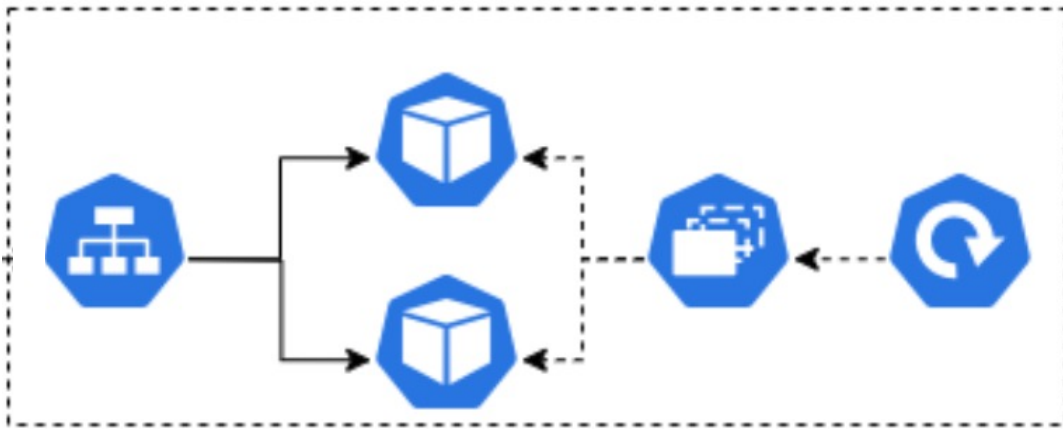
/apis/customers/ redirige vers l'API REST du service CustomersServices

/apis/products/ redirige vers l'API REST du service ProductCatalogService ;



## III.2 - Architecture conteneurisée

**Zoom sur...** Un service K8S



*Pod*

Couche au-dessus d'un conteneur. Un pod peut contenir 1/plusieurs conteneurs (ex:Docker)



*Service*

Adresse IP statique permanente, cycle de vie détaché de celui du Pod



*Limits*

Configurations définissant les ressources maximales (**comme la CPU et la mémoire**) qu'un conteneur peut utiliser

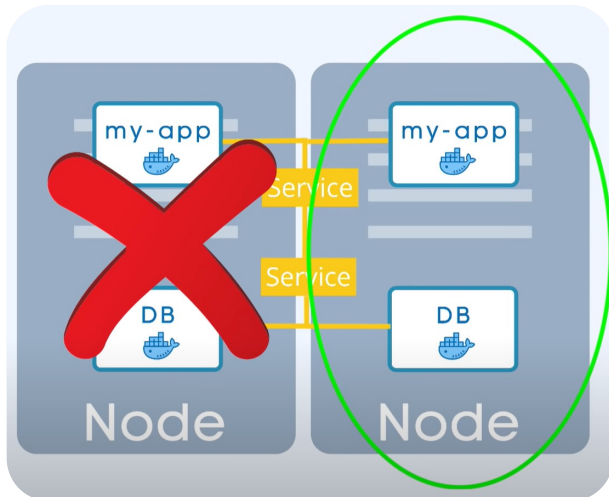
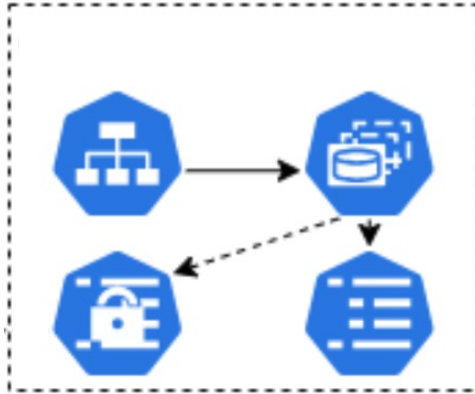


*Deployment*

**Blueprint** permettant de décrire comment les instances de vos applications (pods) doivent être créées et gérées.

## III.2 - Architecture conteneurisée

**Zoom sur...** Une base de données



### Fiabilité de K8S

- 2 replicas de l'application
- 2 replicas de la base de données

**Load Balanced** par le composant Service



### *Service*

Adresse IP statique permanente, cycle de vie détaché de celui du Pod



### *ConfigMap*

**Configuration externe** à l'application  
Données non sécurisées (DB\_URL ...)



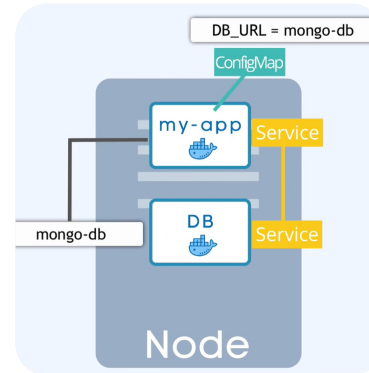
### *Secret*

**Configuration externe** à l'application  
Données sécurisées (credentials...)  
Chiffrés en base64



### *StatefulSets*

Blueprint pour les applications **STATEFUL**  
(stockage persistant, maintient des données)

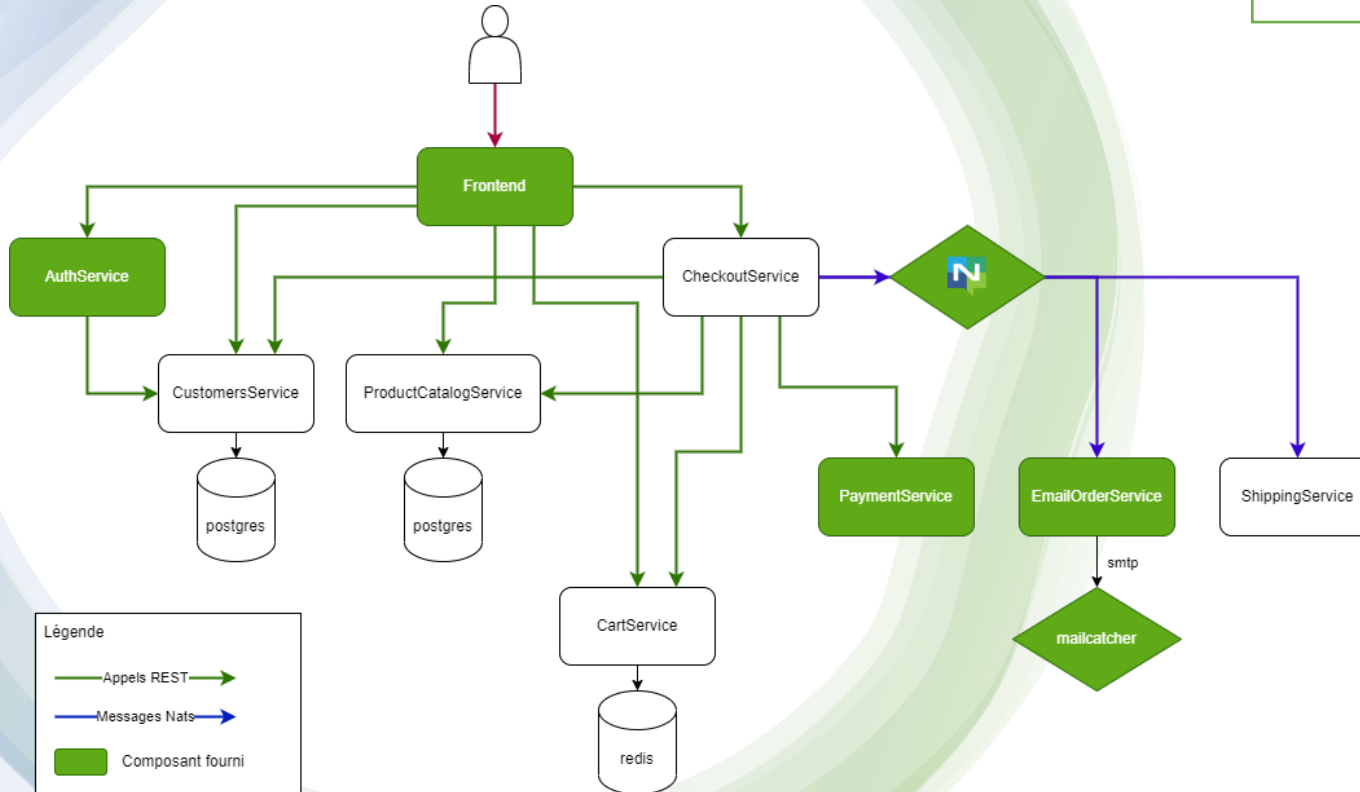


# III.3 - Architecture microservices

- Architecture dite « microservices »
- Services en vert ont été créés par une boîte de prestataires externes

- Tests d'intégration des services en fin de tâches de développement des API

*Comment?*



```
networks:
  default:
    driver: bridge

services:
  frontend:
    image: iutlr-registry.univ-grenoble-alpes.fr/sae5dc03/frontend:latest
    ports:
      - 80:80
    environment:
      AUTHSERVICE_URL: http://auth-service:5000
      AUTHSERVICE_API_KEY: 1234
      CARTSERVICE_URL: http://cart-service:5000
      CARTSERVICE_API_KEY: 1234
      CHECKOUTSERVICE_URL: http://checkout-service:5000
      CHECKOUTSERVICE_API_KEY: 1234
      CUSTOMERSSERVICE_URL: http://customers-service:5000
      CUSTOMERSSERVICE_API_KEY: 1234
      PRODUCTCATALOGSERVICE_URL: http://productcatalog-service:5000
      PRODUCTCATALOGSERVICE_API_KEY: 1234
      LOG_LEVEL: DEBUG
    depends_on:
      - productcatalog-service
      - auth-service
      - customers-service
      - cart-service
      - checkout-service

  adminer:
    image: adminer
    restart: always
    ports:
      - 8080:8080
    depends_on:
      - postgres-product
      - postgres-customer

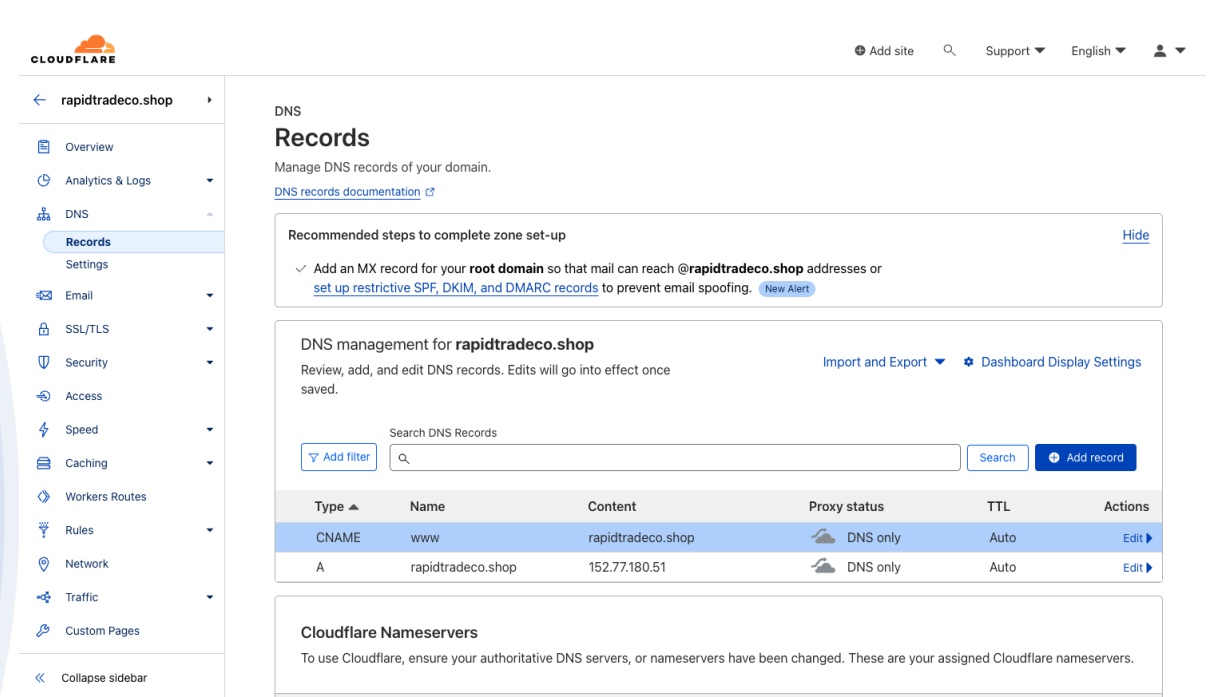
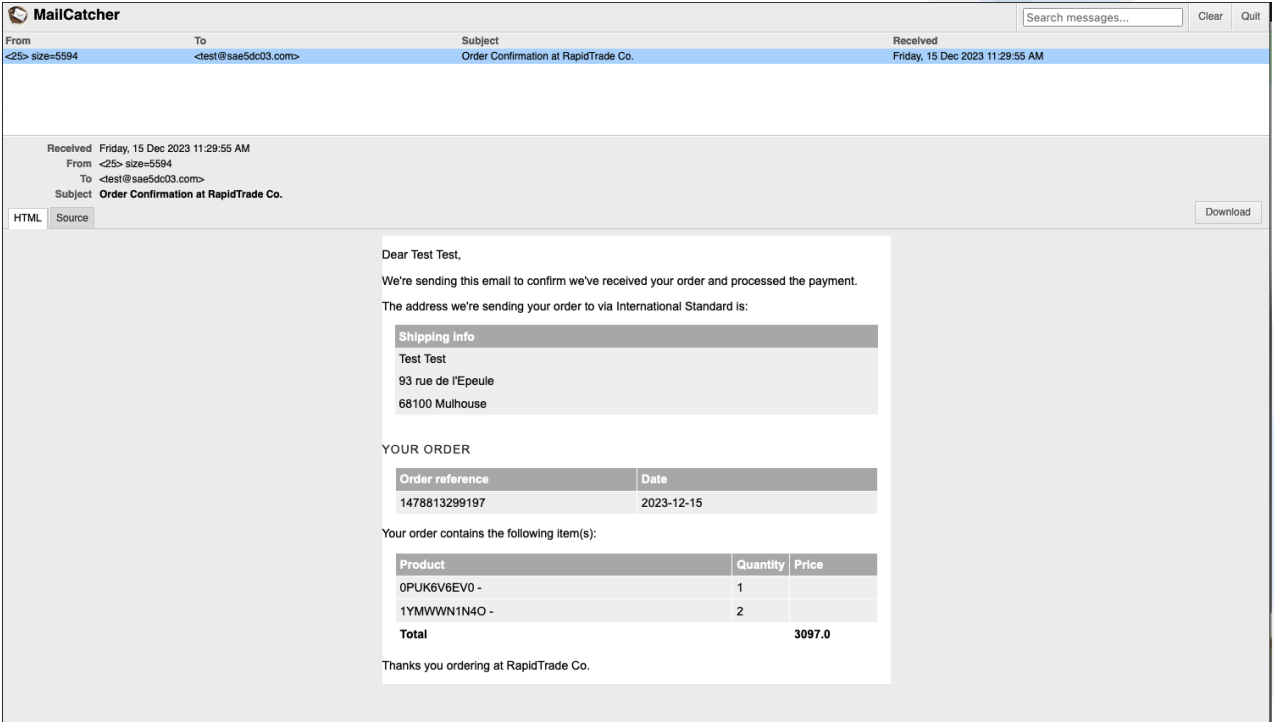
  postgres-customer:
    image: postgres
    environment:
      POSTGRES_PASSWORD: toto123
      POSTGRES_USER: db-customer
      POSTGRES_DB: db-customer
    volumes:
      - ./bdd/customer:/docker-entrypoint-initdb.d

  customers-service:
    restart: always
    image: gricad-registry.univ-grenoble-alpes.fr/iut_rt/but3/sae5dc03-contapp/sae5dc03-a/customer:0.8-amd64
    environment:
      API_KEY: 1234
      POSTGRES_HOST: postgres-customer
      POSTGRES_PORT: 5432
      POSTGRES_DB: db-customer
      POSTGRES_USER: db-customer
      POSTGRES_PASSWORD: toto123
      LOG_LEVEL: DEBUG
    depends_on:
      - postgres-customer
    networks:
      - default
```

**docker-compose.yaml**



# IV.4 – Fonctionnalités supplémentaires



## IV.4 – Fonctionnalités supplémentaires

```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app: nfs-client-provisioner
    name: nfs-client-provisioner
spec:
  template:
    spec:
      containers:
        - name: nfs-client-provisioner
          env:
            - name: NFS_SERVER
              value: 192.168.115.29
            - name: NFS_PATH
              value: /mnt/pool1/kube-files
      volumes:
        - name: nfs-client-root
          nfs:
            server: 192.168.115.29
            path: /mnt/pool1/kube-files
```

## IV.1 - Déploiement


---




## IV.1 - Déploiement réseau virtualisé




## IV.2 – Développement applications et Dockerfile

... sae5dc03-a/checkout 


3 tags

... sae5dc03-a/cartservice 


5 tags

... sae5dc03-a/shipping 

8 tags

... sae5dc03-a/productcat 

7 tags

... sae5dc03-a/customer 

9 tags

```
(venv) PS C:\Users\RT\PycharmProjects\sae5dc03-A> docker build -t gricad-registry.univ-grenoble-alpes.fr/iut_rt/  
but3/sae5dc03-contapp/sae5dc03-a/shipping:0.5-amd64 .
```

## IV.3 – Déploiement cluster Kubernetes

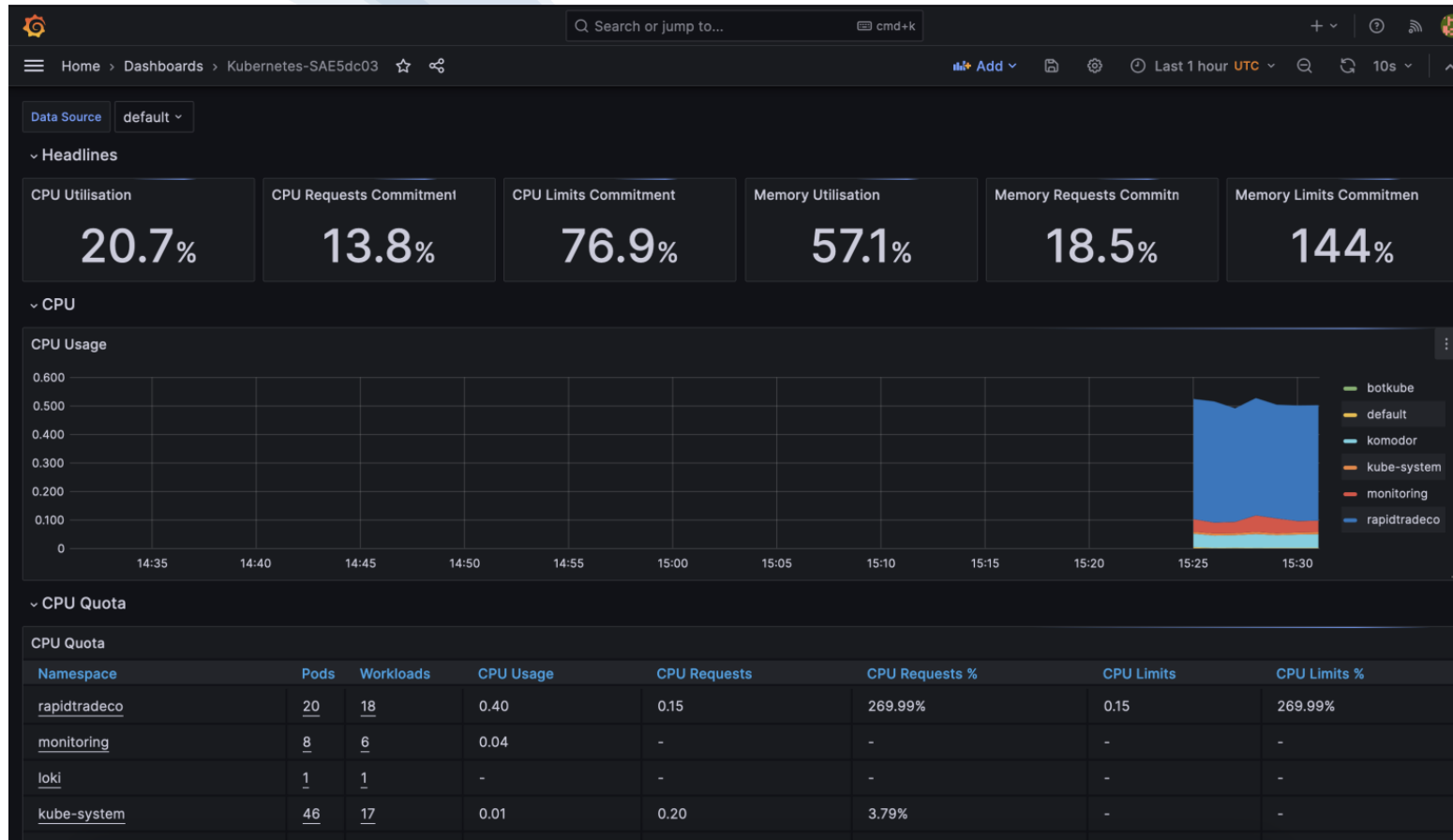
```
admin@worker-node1:~$ curl -sL https://get.k3s.io | K3S_TOKEN=K106deb568d65972c70dc8cbad20ffa  
a4671c8cf544b4be9965e00b201c4a36e12d::1q79yh.237onl2ilxryrk5p K3S_URL=https://192.168.115.10:64  
43 sh -
```

Nous espérons que vous avez reçu de votre administrateur système local les consignes traditionnelles. Généralement, elles se concentrent sur ces trois éléments :

- #1) Respectez la vie privée des autres.
- #2) Réfléchissez avant d'utiliser le clavier.
- #3) De grands pouvoirs confèrent de grandes responsabilités.

```
[sudo] Mot de passe de admin :  
[INFO] Finding release for channel stable  
[INFO] Using v1.27.7+k3s2 as release  
[INFO] Downloading hash https://github.com/k3s-io/k3s/releases/download/v1.27.7+k3s2/sha256sum-amd64.txt  
[INFO] Downloading binary https://github.com/k3s-io/k3s/releases/download/v1.27.7+k3s2/k3s  
[INFO] Verifying binary download  
[INFO] Installing k3s to /usr/local/bin/k3s  
[INFO] Finding available k3s-selinux versions
```

## IV.4 – Mise en place du monitoring



# Conclusion

# Le meilleur site E-Design

## Nouveauté 2023

[En savoir plus](#)

[Produits](#)



## Connexion

Courriel

Mot de passe

Se connecter

Retour au site

[Retour au site](#)

# Le meilleur site E-Design

Pour les personnes connectées: -30% avec le code **ViveLeDesign**  
Nouveauté 2023

[En savoir plus](#)

[Produits](#)



## Tous nos produits

### Canapé Moderne en Cuir



899€

Canapé élégant en cuir véritable avec des coutures raffinées

CANAPÉS, SALON

[Ajouter au panier](#)

### Canapé d'Angle Confortable



1099€

Canapé d'angle spacieux avec des coussins moelleux pour un confort ultime

CANAPÉS, SALON

[Ajouter au panier](#)

### Table à Manger en Chêne Massif



### Fauteuil de Lecture Confortable





## Panier d'achat

Images	Produit	Prix	Quantité	Total
	OPUK6V6EV0 - Canapé Moderne en Cuir	899€	1	899€
Sous-total				899€
Frais de livraison				50€
Total				949€

[Passer la commande](#)

[Supprimer votre panier](#)

[Continuer vos achats](#)



## Valider votre commande

Nom complet

Jean Rault

Adresse de livraison

86 quai Saint-Nicolas

Auxerre

89000

Carte bancaire

Jean Rault

Numéro de la carte

CCV

Année d'expiration



Mois d'expiration



Envoyer

Retour au site

## Valider votre commande

Nom complet

Jean Rault

Adresse de livraison

86 quai Saint-Nicolas

Auxerre

89000

Carte bancaire

Jean Rault

4556426261628918

229

2028



2 - février



Envoyer

Retour au site









