

# TP KUBERNETES

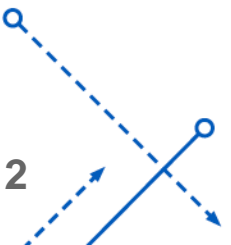


// kodemade



# Sammaire

- **Introduction**
- **Objectif du TP K8S**
- **Installation et configuration de l'environnement de déploiement**
- **Configuration des manifestes K8S**
- **Test et validation du déploiement**
- **Conclusion**



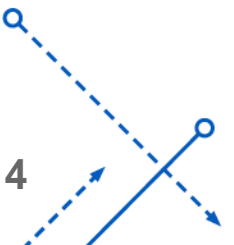
# INTRODUCTION



# Introduction

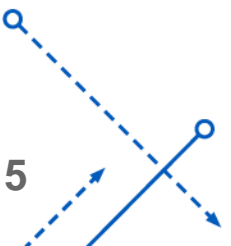
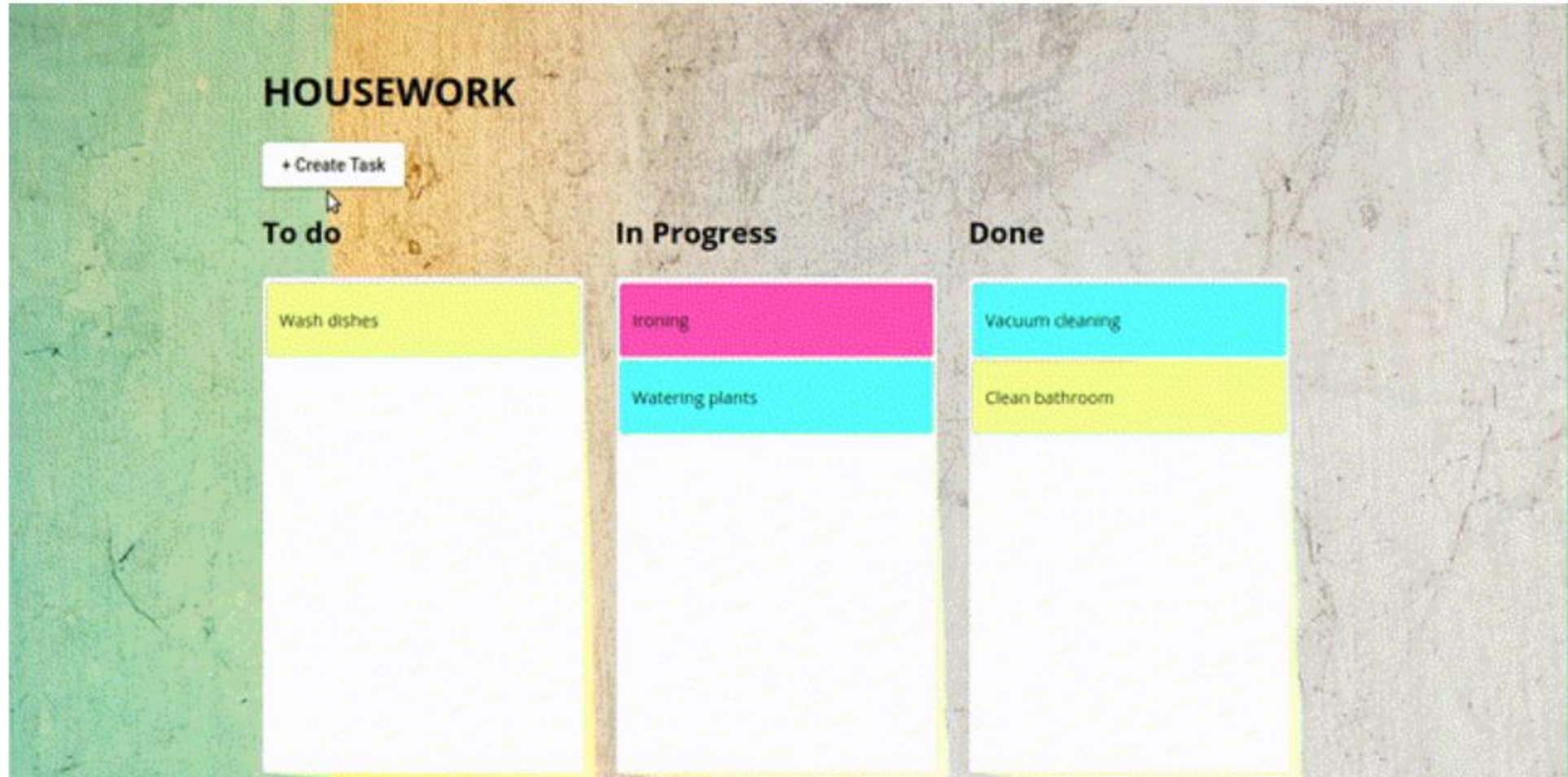
Kubernetes est un système open-source de déploiement, d'orchestration et de gestion de conteneurs qui facilite la mise en place d'environnements de développement et de production à grande échelle. Kanban est une méthode de gestion de projet visuelle qui utilise des tableaux pour représenter l'état de chaque tâche ou activité à effectuer.

Dans ce TP, vous allez apprendre à déployer une application Kanban sur Kubernetes en utilisant les outils et les techniques les plus courantes de la pratique DevOps. Pour cela, vous aurez besoin de connaissances de base sur les conteneurs, Docker, et les commandes de base de Kubernetes.





# Interface de l'app Kanban



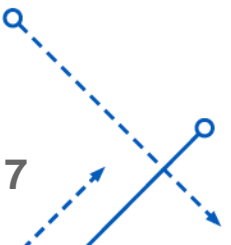
# OBJECTIF DU TP



# Objectif

L'objectif du TP K8S est de :

- Apprendre à installer et configurer Kubernetes sur une infrastructure cloud ou sur votre machine locale.
  - Maîtriser les notions de base sur les conteneurs, Docker et les commandes de base de Kubernetes.
  - Comprendre comment créer des déploiements Kubernetes pour vos applications.
  - Apprendre à configurer des services Kubernetes pour exposer vos applications à l'extérieur du cluster.
- Résultat attendu: Une application Kanban entièrement fonctionnelle déployée sur Kubernetes.



# INSTALLATION ET CONFIGURATION DE L'ENVIRONNEMENT



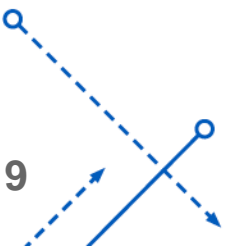


# Installation et configuration de l'environnement

Installer et configurer Kubernetes sur votre machine.

Afin de préparer l'environnement de déploiement vous devez installer Minikube et VirtualBox (si l'ENV sera sur votre machine locale) & Kubectl :

- Installing Docker - <https://docs.docker.com/install/>
- Installing minikube - <https://kubernetes.io/docs/tasks/tools/install-minikube/>
- Installing kubectl - <https://kubernetes.io/docs/tasks/tools/install-kubectl/>



# CONFIGURATION DES MANIFESTES

## K8S



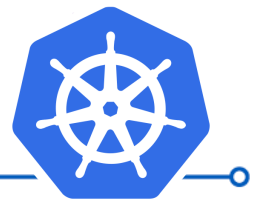
# Configuration des manifestes K8S

Afin de créer les déploiements et les services nécessaires pour réaliser le projet, vous devez créer et déployer les manifestes K8S des déploiements et services qui assurent le respect de l'architecture suivant.

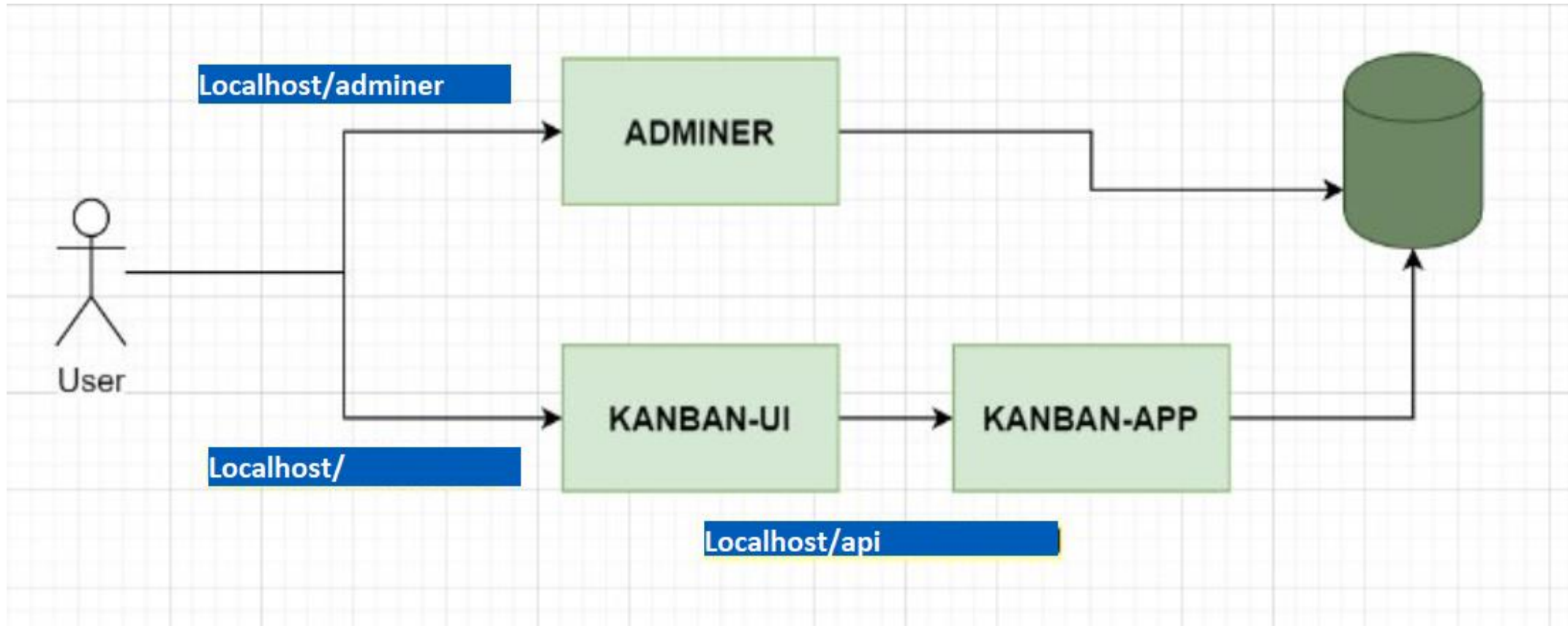
Il contient 3 composants (conteneurs) :

- postgres - base de données
- kanban-app - service backend, desservant les points de terminaison REST pour un frontend
- kanban-ui - service frontal
- Adminer – Pour l'administration de l'application



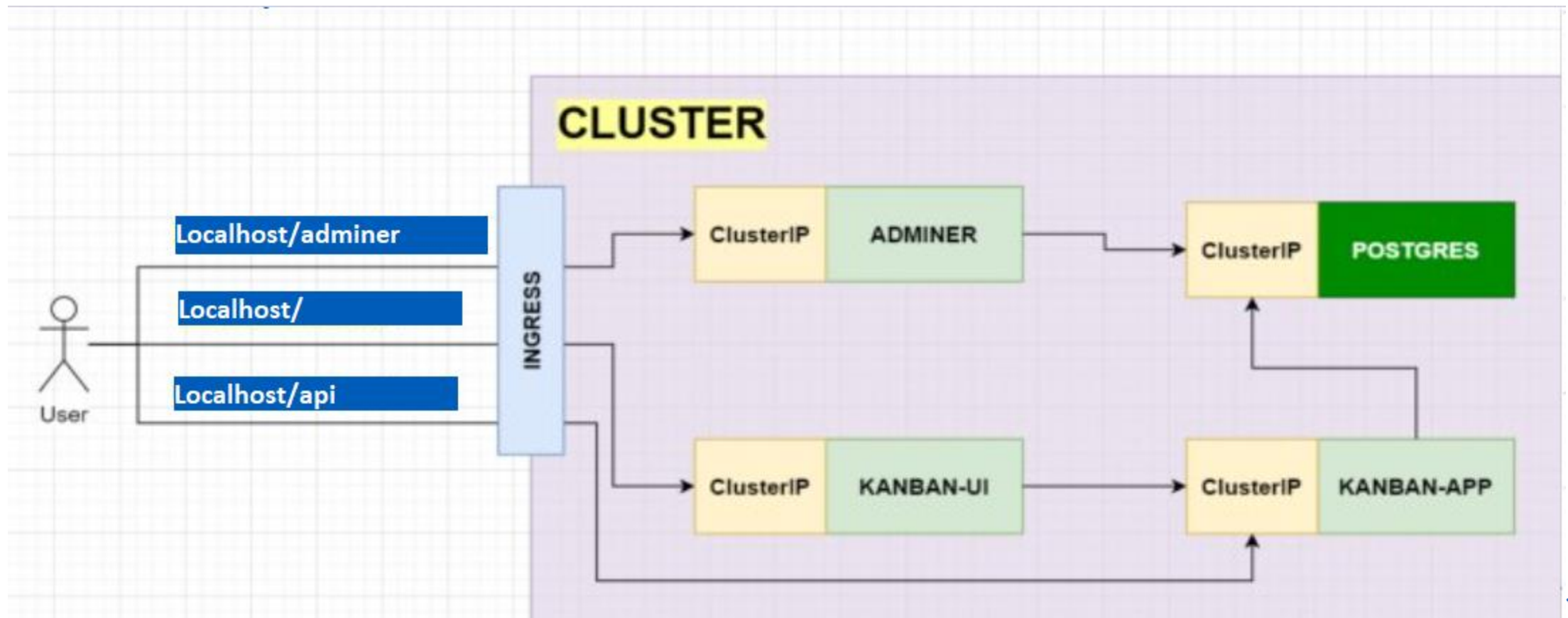


# Configuration des manifestes K8S





# Configuration des manifestes K8S (création des services)





# Configuration des manifestes K8S

Afin de déployer l'application vous devez créer la liste des manifestes suivants :

adminer-deployment.yaml

adminer-svc.yaml

kanban-app-deployment.yaml

kanban-app-svc.yaml

kanban-ui-deployment.yaml

Kanban-ui-svc.yaml

postgres-deployment.yaml

postgres-svc.yaml

ingress.yaml (cours K8S Advanced Ingress)



# Configuration des manifestes K8S

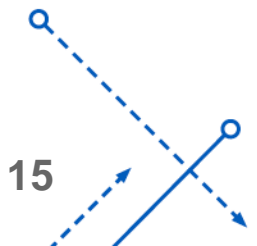
La création et la configuration de Postgres est un cas particulier, une base de données nécessite un persistent volume (PV) qu'est un volume se trouve à l'extérieur du conteneur afin de garder les données en cas de suppression des Pods, ce volume sera disponible dans le cluster ou bien la machine locale, une passerelle (PV Claim), sera le responsable de la communication entre le volume de conteneur à l'aide de volumeMounts et le PV (volume extérieur). Afin de protéger notre base de données, nous devons ajouter un fichier config pour le Postgres qui contient les informations d'accès à notre DB.

Pour cela, vous devez créer :

[postgres-config.yaml](#)

[postgres-pvc.yaml](#)

[postgres-pv.yaml](#)





# Configuration des manifestes K8S

Pour adapter la configuration de l'application à notre base de données Postgres, le fichier ConfigMap doit avoir les informations suivantes :

*data:*

*POSTGRES\_DB: kanban*

*POSTGRES\_USER: kanban*

*POSTGRES\_PASSWORD: kanban*

Le backend (kanban-app) aura besoin également des informations suivantes pour se connecter à la BDD :

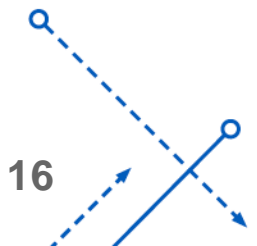
*data:*

*POSTGRES\_DB: kanban*

*POSTGRES\_USER: kanban*

*POSTGRES\_PASSWORD: kanban*

*DB\_SERVER: <lien vers la BDD à définir>*





# TEST ET VALIDATION DU DÉPLOIEMENT



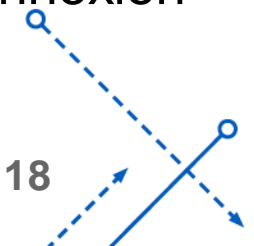
# Test et validation du déploiement

Après avoir fait la configuration de déploiement, vous devez déployer tous les manifestes K8S que vous avez créé avec les images suivantes :

- adminer:4.7.6-standalone pour le pod adminer
- wkrzywiec/kanban-app:k8s pour le pod kanban-app
- wkrzywiec/kanban-ui:k8s pour le pod kanban-ui
- postgres:9.6-alpine pour le pod postgres

Ensuite, vous pouvez tester de se connecter sur les différents chemins notés dans l'INGRESS via un navigateur web et de créer des taches dans Kanban.

Remarque : vous pouvez utiliser les adresses IP dans les DNS en précisant le port de connexion pour accéder aux différents pods.



# CONCLUSION



## Conclusion

Au cours de ce TP, vous allez apprendre comment déployer une application Kanban sur Kubernetes en utilisant les outils et les techniques les plus courantes de la pratique DevOps. En suivant les étapes décrites dans le TP, vous allez pouvoir mettre en place une infrastructure Kubernetes fonctionnelle et déployer avec succès l'application Kanban.

