

kubernetes_ingress

Question 1

Pour installer kind, il faut d'abord installer docker. Ensuite, il faut installer kind avec la commande suivante :

```
curl -Lo ./kind https://kind.sigs.k8s.io/dl/v0.17.0/kind-linux-amd64
chmod +x ./kind
sudo mv ./kind /usr/local/bin/kind
```

Pour créer un cluster, il faut utiliser la commande suivante :

```
kind create cluster --config kind-config.yaml
```

Question 2

Pour installer Nginx Ingress Controller, il faut utiliser la commande suivante :

```
kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/master/deploy/static/provider/kind/deploy.yaml
```

On doit appliquer la commande suivante :

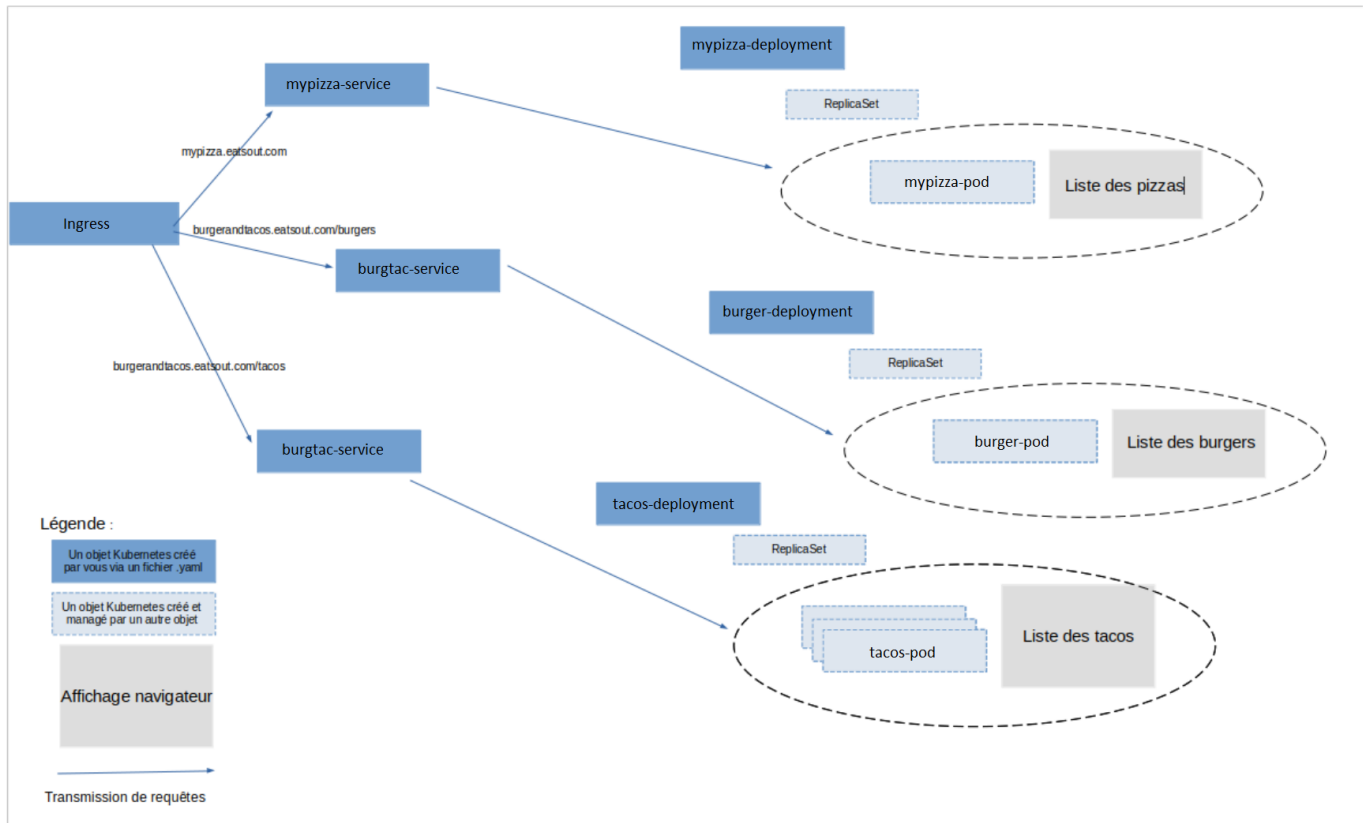
```
kubectl wait --namespace ingress-nginx --for=condition=ready pod --
selector=app.kubernetes.io/component=controller --timeout=90s
```

Pour vérifier que le controller est bien installé, il faut utiliser la commande suivante :

```
kubectl get pods -n ingress-nginx
```

Question 3

Complétion du schéma :



Question 4

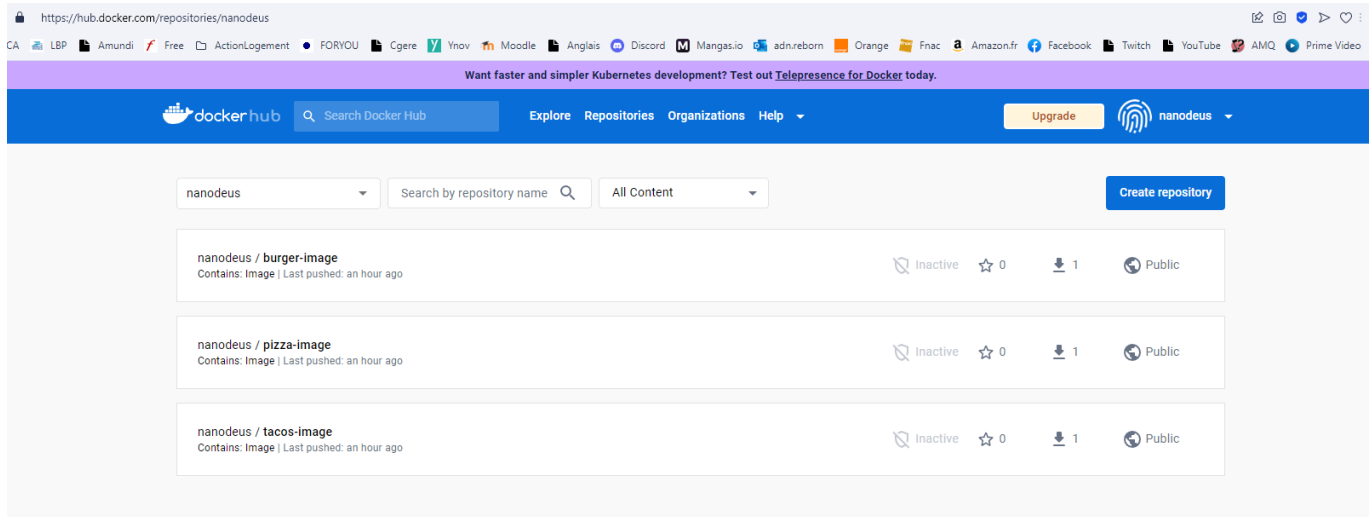
Builder et publier sur le DockerHub :

Nous avons mis en place 3 DockerFiles permettant de créer les 3 images correspondant.

Pour les créer et les publier, on utilise les commandes suivantes :

```
docker build -t nanodeus/tacos-image tacos/
docker build -t nanodeus/burger-image burger/
docker build -t nanodeus/pizza-image pizza/
docker push nanodeus/tacos-image
docker push nanodeus/burger-image
docker push nanodeus/pizza-image
```

On peut retrouver nos images sur le DockerHub : <https://hub.docker.com/u/nanodeus>



Question 5

Pour l'exécution des déploiements et des services, on créait les fichiers yaml correspondants.

Ensuite, on utilise les commandes suivantes :

```

● nano@NanoDeus:~/docker_cours/kubernetes_ingress$ kubectl apply -f burger/burger-deployment.yaml
deployment.apps/burger-deployment created
service/burgers-service created
● nano@NanoDeus:~/docker_cours/kubernetes_ingress$ kubectl apply -f pizza/pizza-deployment.yaml
deployment.apps/pizza-deployment created
service/pizzas-service created
● nano@NanoDeus:~/docker_cours/kubernetes_ingress$ kubectl apply -f tacos/tacos-deployment.yaml
deployment.apps/tacos-deployment created
service/tacos-service created
○ nano@NanoDeus:~/docker_cours/kubernetes_ingress$ █

```

On peut ensuite créer l'ingress qui va nous permettre d'accéder aux services.

On applique la commande suivante :

```

● nano@NanoDeus:~/docker_cours/kubernetes_ingress$ kubectl apply -f ingress/ingress.yaml
ingress.networking.k8s.io/my-ingress created
○ nano@NanoDeus:~/docker_cours/kubernetes_ingress$ █

```

Pour notre cas, il nous faut configurer le fichier hosts de notre machine pour pouvoir accéder aux services.

On ajoute la ligne suivante :

```
127.0.0.1      burgerandtacos.eatsout.com mypizza.eatsout.com
```

On peut ensuite accéder aux services en utilisant curl :

```
● nano@Nano:~/git/docker/Kubernetes_Ingress$ curl burgerandtacos.eatsout.com/burgers
<!DOCTYPE html>
<html>

<head>
  <title>Magasin Burgers</title>
</head>

<body>
  <h1>Magasin Burgers</h1>
</body>

</html>
● nano@Nano:~/git/docker/Kubernetes_Ingress$ curl burgerandtacos.eatsout.com/tacos
<!DOCTYPE html>
<html>

<head>
  <title>Magasin Tacos</title>
</head>

<body>
  <h1>Magasin Tacos</h1>
</body>

</html>
● nano@Nano:~/git/docker/Kubernetes_Ingress$ curl mypizza.eatsout.com
<!DOCTYPE html>
<html>

<head>
  <title>Magasin Pizzas</title>
</head>

<body>
  <h1>Magasin Pizzas</h1>
</body>

○ nano@Nano:~/git/docker/Kubernetes_Ingress$
```

Question 6

Afin de pouvoir gérer une charge plus importante, on peut utiliser la propriété replicas qui permet de créer plusieurs pods pour un même service.

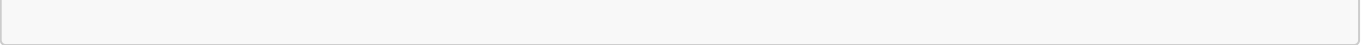
```
replicas: 3
```

Pour vérifier que les pods sont bien créés, on utilise la commande suivante :

```
kubectl get pods
```

On peut aussi utiliser un service de type LoadBalancer.

```
type: LoadBalancer
```



Kubernetes créera automatiquement un équilibreur de charge pour votre Service