

## PARTIE 2

```
kubectl apply -n database -f mysql-secret.yml  
kubectl apply -n database -f mysql-configmap.yml  
kubectl apply -n database -f mysql-statefulset.yaml  
kubectl apply -n database -f mysql-service.yml
```

```
C:\Users\Neiky\Documents\BUT3\Kubernetes\tp1\database>kubectl exec -i mysql-statefulset-0 -n database -- printenv | findstr MYSQL_  
MYSQL_USER=neiky  
MYSQL_PASSWORD=mysql_password  
MYSQL_ROOT_PASSWORD=root_password  
MYSQL_DATABASE=wordpress
```

perso (quand on change les secrets) **pas prendre en compte**

```
kubectl -n database delete statefulset mysql-statefulset  
kubectl -n database delete pvc mysql-persistent-storage-mysql-statefulset-0  
kubectl -n database apply -f mysql-statefulset.yml
```

```
kubectl apply -f wordpress-pv.yml  
kubectl apply -f wordpress-pvc.yml
```

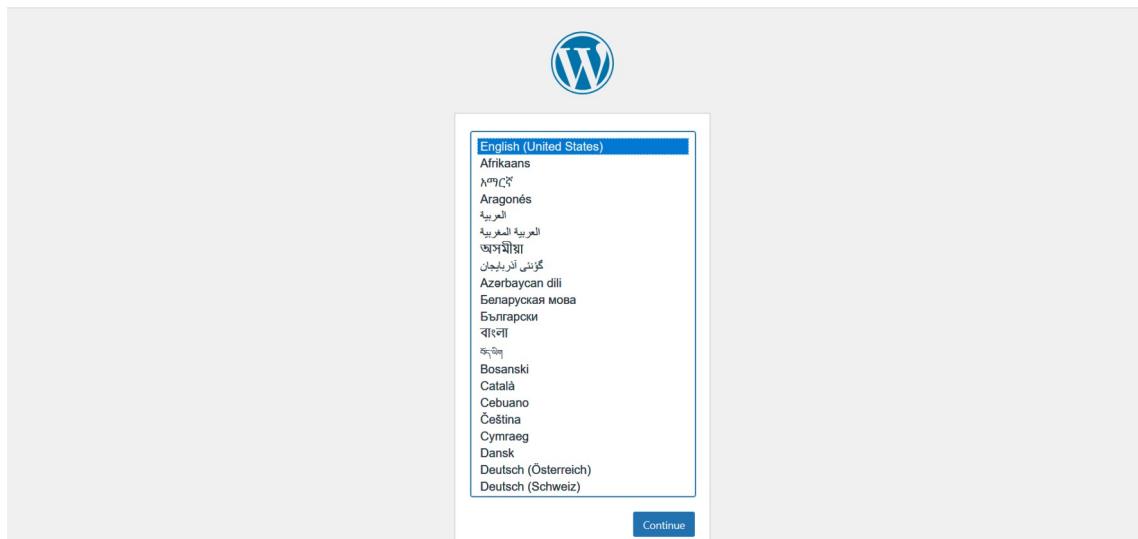
```
kubectl apply -f mysql-secret.yml -n middle  
kubectl apply -f mysql-configmap.yml -n middle
```

```
kubectl apply -n middle -f wordpress-deployment.yml  
kubectl apply -n middle -f wordpress-service.yml
```

```
C:\Users\Neiky\Documents\BUT3\Kubernetes\tp1\database>kubectl exec -it wordpress-59d6cbfd48-7gf78 -n middle -- printenv  
| findstr WORDPRESS_  
WORDPRESS_DB_USER=neiky  
WORDPRESS_DB_NAME=wordpress  
WORDPRESS_DB_HOST=mysql-service.middle.svc.cluster.local  
WORDPRESS_DB_PASSWORD=password
```

```
kubectl apply -n front -f TP/front/nginx-cm.yml
```

```
kubectl port-forward svc/wordpress-service 8080:80 -n middle
```



## VERIFICATION

1) Les ressources sont déployées dans le bon namespace ?

```
kubectl get all -n database  
kubectl get all -n middle  
kubectl get all -n front
```

```
kubectl get configmap,secret,pvc,statefulset -n database  
kubectl get configmap,secret,pvc,deployment -n middle
```

2) Pourquoi déployer MySQL ConfigMap et MySQL dans les namespaces middle et database ?

- Isolation et séparation des responsabilités : le namespace database regroupe tout ce qui est lié à la BDD (StatefulSet, PV, Service) ; middle contient l'application qui consomme la BDD (WordPress).
- Les ConfigMap et Secrets sont namespaced : une ConfigMap/Secret ne peut pas être référencée directement par un Pod d'un autre namespace. Si WordPress (dans middle) doit utiliser les mêmes secrets/configs que MySQL (dans database), il faut soit dupliquer ces ConfigMap/Secret dans middle, soit exposer les informations d'une autre façon.
- Sécurité / RBAC / quotas : en scindant par namespace, on peut appliquer des politiques différentes.

3) Les pods sont-ils correctement démarrés ? Quelle commande utiliser ?

```
kubectl get pods -n database
```

```
kubectl get pods -n middle
```

```
kubectl get pods -n front
```

4) Lorsqu'un pod est en erreur, comment accéder aux events du namespace et aux logs du pod ? Commandes à utiliser.

```
kubectl get events -n <namespace>
```

```
kubectl logs <pod-name> -n <namespace>
```