

## Instalacja pierwszej aplikacji i poruszanie się po klastrze

### 1. Deploy kontenera nginx na klastrze kubernetes:

```
kubectl create deployment nginx --image=nginx
```

### 2. Utworzenie serwisu dla kontenera nginx w celu udostępnienia go w sieci:

```
kubectl create service loadbalancer nginx --tcp=80:80
```

### 3. Wyświetlanie, logowanie oraz znajdowanie zasobów:

```
kubectl get nodes - wyświetla wszystkie nody (workery)
```

```
kubectl get namespaces - wyświetla wszystkie dostępne namespaces
```

```
kubectl get pod --all-namespaces - wyświetla wszystkie pody ze wszystkich namespace'ów
```

```
kubectl get pod - wyświetla wszystkie pody z aktualnego namespace'a
```

```
kubectl get pod -o wide - wyświetla wszystkie pody z aktualnego namespace'a z dodatkowymi informacjami, jak IP wew., node na którym jest uruchomiony pod oraz nominated node - co oznacza priorytet noda. Wskazuje on na znaczenie poda w stosunku do innych podów
```

```
kubectl get pod -n <nazwa_namespace> - wyświetla listę podów z danego namespace
```

```
kubectl get services - wyświetla listę serwisów w danym namespace
```

```
kubectl get deployment - wyświetla listę deploymentów
```

```
kubectl get ingresses.extensions --all-namespaces - wyświetla listę ingresów (nazwę usług wraz z nazwą hosta)
```

```
kubectl get pvc - wyświetla listę persistent volume claim
```

```
kubectl describe pod <nazwa_poda> - wyświetla informacje o podzie
```

```
kubectl logs <nazwa_poda> - wyświetla logi kontenera, jeśli w podzie znajduje się więcej niż jeden kontener podajemy dodatkowo nazwę kontenera, czyli: kubectl logs <nazwa_poda> <nazwa_kontenera>
```

```
kubectl config view - wyświetla ustawienia kubeconfig (w tym miejscu możemy podejrzeć, w którym namespace aktualnie się znajdujemy)
```

### 4. Zmiana, edycja i tworzenie obiektów

```
kubectl config set-context $(kubectl config current-context) --namespace=<nazwa_namespace> - zmiana namespace
```

```
kubectl create namespace <nazwa_nowego_namespace> - stworzenie nowego namespace'a
```

```
kubectl create -f manifest.yaml - tworzymy nowy obiekt (może to być deployment, serwis, pod itp)
```

```
kubectl create -f ./manifest1.yaml -f ./manifest2.yaml - tworzymy obiekty z wielu źródeł
```

`kubectl create -f ./dir` - tworzymy obiekty ze wszystkich plików manifesty znajdujących się w danym katalogu

`kubectl apply -f manifest.yaml` - zatwierdzamy zmiany, które wprowadziliśmy do pliku manifestu

`kubectl delete pod <nazwa_poda>` - usuwanie poda

`kubectl delete svc <nazwa_serwisu>` - usunięcie serwisu

`kubectl delete deployment <nazwa_deploymentu>` - usunięcie deploymentu

`kubectl delete pvc <nazwa_volumenu>` - usunięcie persistent volume