

# Домашнее задание №5

## 1. Установка kubectl и minikube

```
Project ▾
  ▾ devops C:\Users\Aleksy\Pych
Terminal kharlanov x + ▾
kharlanov@dev:~$ curl -LO https://dl.k8s.io/release/`curl -LS https://dl.k8s.io/release/stable.txt`/bin/linux/amd64/kubectl
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 138 100 138 0 0 400 0 --:--:-- --:--:-- --:--:-- 401
100 7 100 7 0 0 9 0 --:--:-- --:--:-- --:--:-- 41
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 138 100 138 0 0 470 0 --:--:-- --:--:-- --:--:-- 470
100 53.7M 100 53.7M 0 0 9567k 0 0:00:05 0:00:05 --:--:-- 10.4M
kharlanov@dev:~$ chmod +x ./kubectl
kharlanov@dev:~$ sudo mv ./kubectl /usr/local/bin/kubectl
kharlanov@dev:~$ kubectl version --client
Client Version: v1.31.3
Kustomize Version: v5.4.2
kharlanov@dev:~$ curl -Lo minikube https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64 \
&& chmod +x minikube
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 99.0M 100 99.0M 0 0 10.0M 0 0:00:09 0:00:09 --:--:-- 10.8M
kharlanov@dev:~$ sudo mkdir -p /usr/local/bin/
kharlanov@dev:~$ sudo install minikube /usr/local/bin/
```

## 2. Запуск локального кластера minikube

```
kharlanov@dev:~$ sudo minikube start --force --driver=docker
😄 minikube v1.34.0 on Ubuntu 22.04 (vbox/amd64)
! minikube skips various validations when --force is supplied; this may lead to unexpected behavior
🌟 Using the docker driver based on user configuration
🚨 The "docker" driver should not be used with root privileges. If you wish to continue as root, use --force.
💡 If you are running minikube within a VM, consider using --driver=none:
📄 https://minikube.sigs.k8s.io/docs/reference/drivers/none/
🚀 Using Docker driver with root privileges
👍 Starting "minikube" primary control-plane node in "minikube" cluster
📦 Pulling base image v0.0.45 ...
📦 Downloading Kubernetes v1.31.0 preload ...
> preloaded-images-k8s-v18-v1...: 326.69 MiB / 326.69 MiB 100.00% 4.47 Mi
> gcr.io/k8s-minikube/kicbase...: 487.90 MiB / 487.90 MiB 100.00% 5.81 Mi
🔥 Creating docker container (CPUs=2, Memory=2200MB) ...
🔧 Preparing Kubernetes v1.31.0 on Docker 27.2.0 ...
  ▪ Generating certificates and keys ...
  ▪ Booting up control plane ...
  ▪ Configuring RBAC rules ...
🔗 Configuring bridge CNI (Container Networking Interface) ...
🔍 Verifying Kubernetes components...
  ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟 Enabled addons: storage-provisioner, default-storageclass
🎉 Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
```

## 3. Проверка работы kubectl и minikube

```
Project v
devops C:\Users\Aleksey\Pyth
Terminal kharlanov x + v
kharlanov@dev:~$ sudo minikube status
[sudo] password for kharlanov:
minikube
...
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured

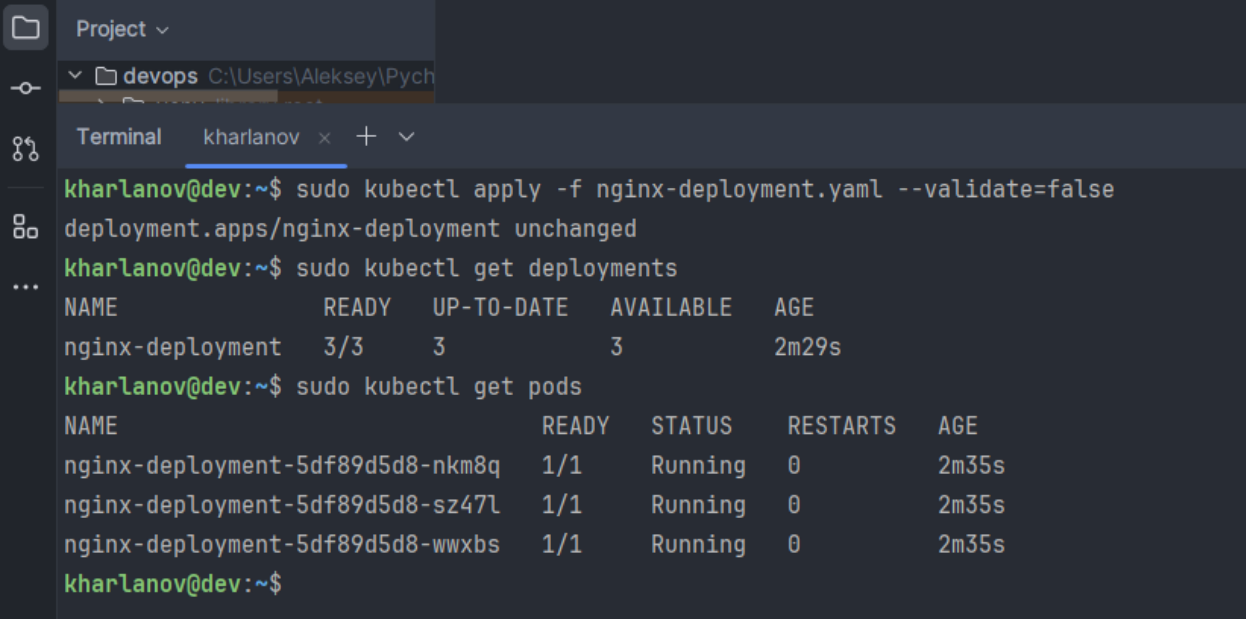
kharlanov@dev:~$ sudo kubectl cluster-info
Kubernetes control plane is running at https://192.168.49.2:8443
CoreDNS is running at https://192.168.49.2:8443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
kharlanov@dev:~$
```

#### 4. Конфигурация deployment nginx

```
Project v
devops C:\Users\Aleksey\Pyth
Terminal kharlanov x + v
GNU nano 6.2
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:latest
        ports:
        - containerPort: 8081
```

## 5. Проверка deployment и pods



The screenshot shows a terminal window with the following commands and output:

```
kharlanov@dev:~$ sudo kubectl apply -f nginx-deployment.yaml --validate=false
deployment.apps/nginx-deployment unchanged
kharlanov@dev:~$ sudo kubectl get deployments
```

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
nginx-deployment	3/3	3	3	2m29s

```
kharlanov@dev:~$ sudo kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
nginx-deployment-5df89d5d8-nkm8q	1/1	Running	0	2m35s
nginx-deployment-5df89d5d8-sz47l	1/1	Running	0	2m35s
nginx-deployment-5df89d5d8-wwxbs	1/1	Running	0	2m35s

```
kharlanov@dev:~$
```