

Скачивание необходимых инструментов

```
parallels@debian-gnu-linux-12:~/uber$ curl https://raw.githubusercontent.com/helm/helm/master/scripts/get-helm-3 | bash
% Total    % Received % Xferd  Average Speed   Time   Time     Current
          Dload  Upload Total Spent   Left Speed
100 11903 100 11903    0     0 14028      0 --:--:-- --:--:-- 14020
Downloading https://get.helm.sh/helm-v3.16.4-linux-arm64.tar.gz
Verifying checksum... Done.
Preparing to install helm into /usr/local/bin
[sudo] password for parallels:
helm installed into /usr/local/bin/helm
parallels@debian-gnu-linux-12:~/uber$ helm version
version.BuildInfo{Version:"v3.16.4", GitCommit:"7877b45b63f95635153b29a42c0c2f4273ec45ca", GitTreeState:"clean", GoVersion:"go1.22.7"}
-----
```

```
parallels@debian-gnu-linux-12:~$ curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-arm64
% Total    % Received % Xferd  Average Speed   Time   Time     Current
          Dload  Upload Total Spent   Left Speed
100 99.0M 100 99.0M    0     0 4474k      0 0:00:22 0:00:22 --:--:-- 6713k
parallels@debian-gnu-linux-12:~$ sudo install minikube-linux-arm64 /usr/local/bin/minikube
parallels@debian-gnu-linux-12:~$ minikube version
minikube version: v1.34.0
commit: 210b148df93a80eb872ecbeb7e35281b3c582c61
parallels@debian-gnu-linux-12:~$ minikube start --driver=docker
😊 minikube v1.34.0 on Debian 12.7 (arm64)
✨ Using the docker driver based on user configuration
-----
```

```
parallels@debian-gnu-linux-12:~$ minikube start --driver=docker
😊 minikube v1.34.0 on Debian 12.7 (arm64)
✨ Using the docker driver based on user configuration

❗ The requested memory allocation of 1977MiB does not leave room for system overhead (total system memory: 1977MiB). You may face stability issues.
💡 Suggestion: Start minikube with less memory allocated: 'minikube start --memory=1977mb'

🚀 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...: 91.95 MiB / 307.61 MiB 29.89% 101.52 Ki
  > gcr.io/k8s-minikube/kicbase...: 100.22 MiB / 441.45 MiB 22.70% 101.61 K
  > index.docker.io/kicbase/sta...: 441.45 MiB / 441.45 MiB 100.00% 124.94
❗ minikube was unable to download gcr.io/k8s-minikube/kicbase:v0.0.45, but successfully downloaded docker.io/kicbase/stable:v0.0.45 as a fallback image
🔥 Creating docker container (CPUs=2, Memory=1977MB) ...
  > kubectl.sha256: 64 B / 64 B [=====] 100.00% ? p/s 0s
  > kubelet.sha256: 64 B / 64 B [=====] 100.00% ? p/s 0s
  > kubeadm.sha256: 64 B / 64 B [=====] 100.00% ? p/s 0s
  > kubelet: 71.18 MiB / 71.18 MiB [=====] 100.00% 1.64 MiB p/s 44s
  > kubectl: 52.44 MiB / 52.44 MiB [=====] 100.00% 143.24 KiB p/s 6m15s
  > kubeadm: 54.25 MiB / 54.25 MiB [=====] 100.00% 92.80 KiB p/s 9m59s

  • 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
💡 kubectl not found. If you need it, try: 'minikube kubectl -- get pods -A'
📝 Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
-----
```

Проверка

```
parallels@debian-gnu-linux-12:~$ minikube kubectl -- get nodes
NAME      STATUS    ROLES      AGE      VERSION
minikube  Ready     control-plane  5m9s    v1.31.0
```

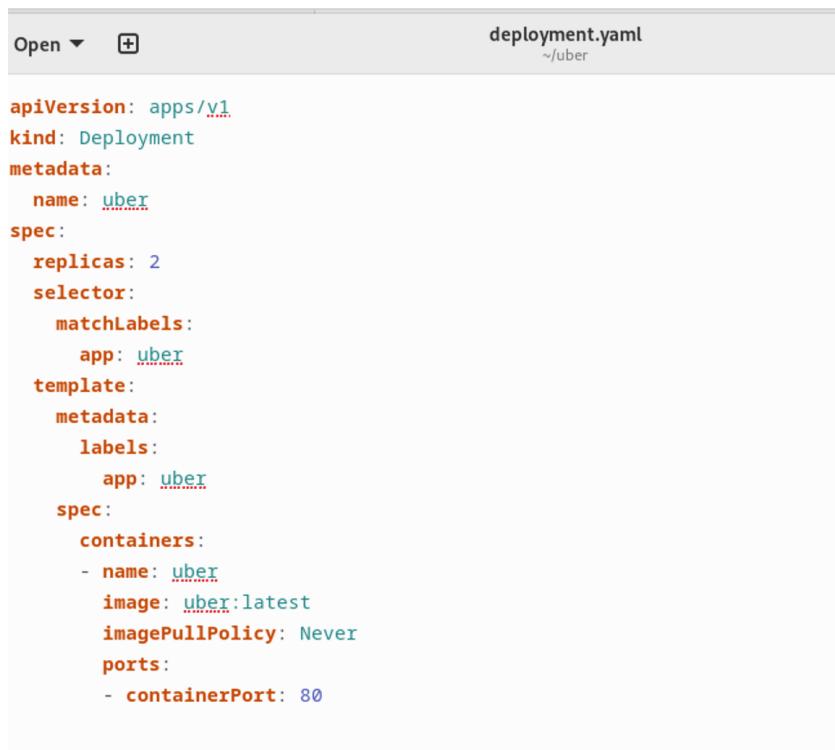
Создали Dockerfile в корне проекта. Далее собираем из него образ

(*docker build -t uber:latest .*)



```
FROM nginx:alpine
COPY src /usr/share/nginx/html
```

Там же создаем Deployment.yaml



```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: uber
spec:
  replicas: 2
  selector:
    matchLabels:
      app: uber
  template:
    metadata:
      labels:
        app: uber
    spec:
      containers:
        - name: uber
          image: uber:latest
          imagePullPolicy: Never
        ports:
          - containerPort: 80
```

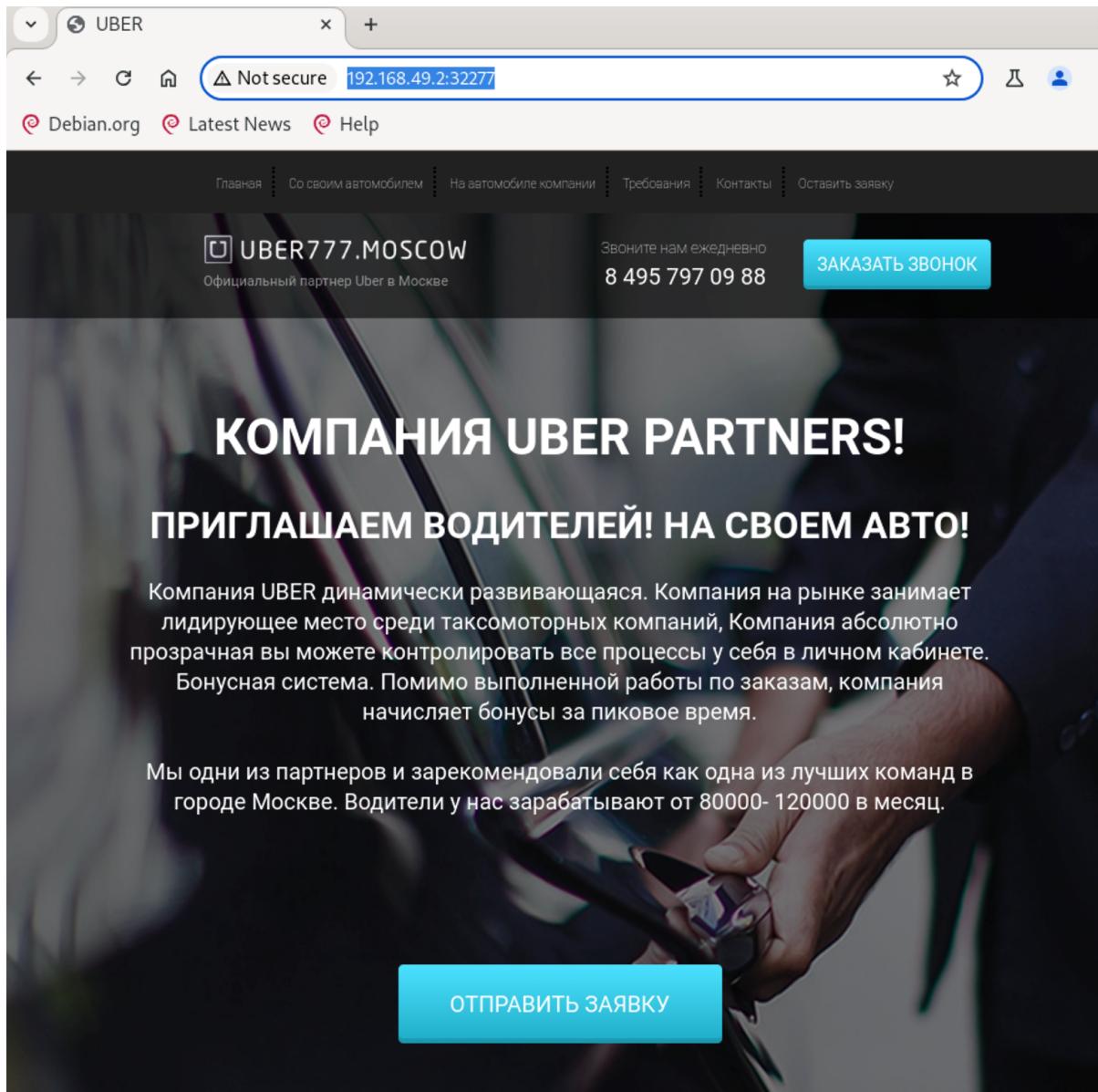
И Service.yaml



```
apiVersion: v1
kind: Service
metadata:
  name: uber-service
spec:
  selector:
    app: uber
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
  type: NodePort
```

```
parallels@debian-gnu-linux-12:~/uber$ minikube kubectl -- apply -f deployment.yaml
minikube kubectl -- apply -f service.yaml
deployment.apps/uber created
service/uber-service created
```

```
parallels@debian-gnu-linux-12:~/uber$ minikube kubectl -- get pods
NAME           READY   STATUS    RESTARTS   AGE
uber-84b5c8fc4b-2xssg  1/1     Running   0          7s
uber-84b5c8fc4b-9rv2k  1/1     Running   0          7s
parallels@debian-gnu-linux-12:~/uber$ minikube kubectl -- get svc
NAME        TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
kubernetes  ClusterIP  10.96.0.1      <none>         443/TCP      52m
uber-service NodePort  10.107.100.88  <none>         80:32277/TCP  14m
parallels@debian-gnu-linux-12:~/uber$
```



Создаем файл .gitlab-ci.yml в корне проекта

```
⚠️ .gitlab-ci.yml ×  
Users > valeriali > VSProjects > uber > ⚡ .gitlab-ci.yml  
1 stages:  
2   - build  
3   - deploy_staging  
4   - deploy_production  
5  
6 variables:  
7   DOCKER_IMAGE: registry.gitlab.com/lera8lee/uber:$CI_COMMIT_SHORT_SHA  
8  
9 build:  
10  stage: build  
11    script:  
12      - docker build -t $DOCKER_IMAGE .  
13      - echo $CI_REGISTRY_PASSWORD | docker login -u $CI_REGISTRY_USER --password-stdin $CI_REGISTRY  
14      - docker push $DOCKER_IMAGE  
15  
16 deploy_staging:  
17  stage: deploy_staging  
18    script:  
19      - kubectl apply -f deployment.yaml  
20      - kubectl apply -f service.yaml  
21    environment:  
22      name: staging  
23      url: http://192.168.49.2:32277  
24    only:  
25      - main  
26  
27 deploy_production:  
28  stage: deploy_production  
29    script:  
30      - kubectl apply -f deployment.yaml  
31      - kubectl apply -f service.yaml  
32    environment:  
33      name: production  
34      url: http://192.168.49.2:32277  
35    only:  
36      - main  
37
```

Создание кластера в проекте

lera8lee / uber / Kubernetes

Did you know?
Every new Google Cloud Platform (GCP) account receives \$300 in credit upon [sign up](#). In partnership with Google, GitLab is able to offer an additional \$200 for both new and existing GCP accounts to get started with GitLab's Google Kubernetes Engine Integration.

Apply for credit

Agent

Connect a cluster

Tell us what you think
We would love to learn more about your experience with the GitLab Agent.
Give feedback

my-agent successfully created
Optional, for additional configuration settings, a [configuration file](#) can be created in the repository. You can do so within the default branch by creating the file at: .gitlab/agents/my-agent/config.yaml

Name	Connection status	Last contact	Version	Agent ID	Configuration
my-agent	Never connected	Never	2118994	Default configuration	⋮

Подключились к нашему кластеру через helm (нужная команда была указана после создания кластера)

```
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "gitlab" chart repository
Update Complete. *Happy Helming!*
Release "my-agent" does not exist. Installing it now.
NAME: my-agent
LAST DEPLOYED: Sat Dec 21 17:03:58 2024
NAMESPACE: gitlab-agent-my-agent
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
Thank you for installing gitlab-agent.

Your release is named my-agent.

## Changelog

### 1.17.0

- The default replica count has been increased from `1` to `2` to allow a zero-downtime upgrade experience.
  You may use `--set replicas=1` to restore the old default behavior.
parallels@debian-gnu-linux-12:~/uber$
```

После статус поменялся на “Connected”

Name	Connection status	Last contact	Version	Agent ID	Configuration	⋮
my-agent	Connected	just now	17.7.0	2118994	Default configuration ⓘ	