

## Домашнее задание для к уроку 8 - CI/CD

В рамках данного задания нужно продолжить работу с CI приложения из лекции. То есть для начала выполнения этого домашнего задания необходимо проделать то, что показывалось в лекции. Все задание должно выполняться применительно к файлам в директории `practice/8.ci-cd/app`

Переделайте шаг деплоя в CI/CD, который демонстрировался на лекции таким образом, чтобы при каждом прогоне шага `deploy` в кластер применялись манифесты приложения. При этом версия докер образа в деплойменте при апплае должна подменяться на ту, что была собрана в шаге `build`.

Для этого самым очевидным способом было бы воспользоваться утилитой `sed`.

- Измените образ в деплойменте приложения (файл `kube/deployment.yaml`) на плейсхолдер.

Вот это

```
image: nginx:1.12 # это просто плейсхолдер
```

На это

```
image: __IMAGE__
```

- Измените шаг деплоя в `.gitlab-ci.yml`, чтобы изменять `IMAGE` на реальное имя образа и тег

Это

```
- kubectl set image deployment/$CI_PROJECT_NAME *=$CI_REGISTRY_IMAGE:
$CI_COMMIT_REF_SLUG.$CI_PIPELINE_ID --namespace $CI_ENVIRONMENT_NAME
```

На это

```
- sed -i "s,__IMAGE__,${CI_REGISTRY_IMAGE}:${CI_COMMIT_REF_SLUG}.${CI_PIPELINE_ID},g"
kube/deployment.yaml
- kubectl apply -f kube/ --namespace $CI_ENVIRONMENT_NAME
```

Вторую строчку шага деплоя (которая отслеживает статус деплоя) оставьте без изменений.

- Попробуйте закоммитить свои изменения, запустить их в репозиторий (тот же, который вы создавали во время лекции на `Gitlab.com`) и посмотреть на выполнение CI в интерфейсе `Gitlab`.

Так как окружений у нас два (`stage` и `prod`), то помимо образа при апплае из CI нам также было бы хорошо подменять `host` в `ingress.yaml`. Попробуйте реализовать это по аналогии, подставляя в ингресс вместо плейсхолдера значение переменной `$CI_ENVIRONMENT_NAME`

- Так же попробуйте протестировать откат на предыдущую версию, при возникновении ошибки при деплое

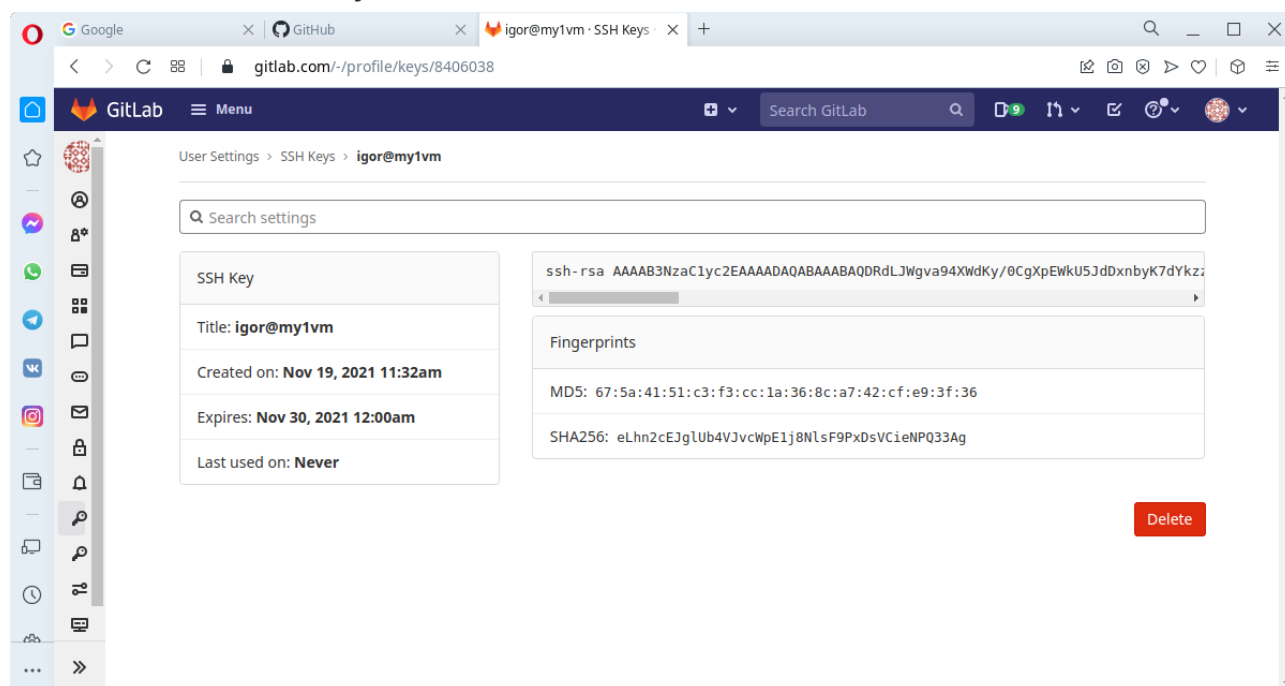
Для этого можно изменить значение переменной DB\_HOST в deployment.yaml на какое нибудь несуществующее. Тогда при старте приложения оно не сможет найти БД и будет постоянно рестартовать. CI должен в течении progressDeadlineSeconds: 300 и после этого запустить процедуру отката. При этом не должно возникать недоступности приложения, так как старая реплика должна продолжать работать, пока новая пытается стартовать.

### Проверка кластера

```
igor@my1vm: ~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
igor@my1vm:~$ 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'.
igor@my1vm:~$
```

Зарегистрировал аккаунт [GitLab](#). Добавил в настройках своего аккаунта на Gitlab.com свой публичный SSH ключ



```
ssh-add id_rsa_gitlab
eval $ssh-agent
eval $ssh-agent -s
eval $(ssh-agent)
```

```
igor@my1vm: ~/.ssh - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
igor@my1vm:~/.ssh$ ssh-add id_rsa_gitlab
Could not open a connection to your authentication agent.
igor@my1vm:~/.ssh$ eval $ssh-agent
-bash: eval: -a: invalid option
eval: usage: eval [arg ...]
igor@my1vm:~/.ssh$ eval $ssh-agent -s
-bash: eval: -a: invalid option
eval: usage: eval [arg ...]
igor@my1vm:~/.ssh$ eval $(ssh-agent)
Agent pid 93449
igor@my1vm:~/.ssh$
```

ssh-add id\_rsa\_gitlab

```
igor@my1vm: ~/.ssh - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
Agent pid 93449
igor@my1vm:~/.ssh$ ssh-add id_rsa_gitlab
Enter passphrase for id_rsa_gitlab:
Identity added: id_rsa_gitlab (id_rsa_gitlab)
igor@my1vm:~/.ssh$
```

cd ~/cicd

```
igor@my1vm: ~/cicd - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
igor@my1vm:~/.ssh$ ssh-add id_rsa_gitlab
Enter passphrase for id_rsa_gitlab:
Identity added: id_rsa_gitlab (id_rsa_gitlab)
igor@my1vm:~/.ssh$ cd ~/cicd
igor@my1vm:~/cicd$
```

git push -u origin master  
git push origin master

```
igor@my1vm: ~/cicd - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
igor@my1vm:~/cicd$ git push -u origin master
error: src refspec master does not match any.
error: failed to push some refs to 'git@gitlab.com:TolstikovIA/geekbrains.git'
igor@my1vm:~/cicd$ git push origin master
error: src refspec master does not match any.
error: failed to push some refs to 'git@gitlab.com:TolstikovIA/geekbrains.git'
igor@my1vm:~/cicd$
```

git status

```
igor@my1vm: ~/cicd - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
igor@my1vm:~/cicd$ git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)
igor@my1vm:~/cicd$
```

echo "test" >> test.txt

git add .

git commit -m "Initial commit"

```
igor@my1vm: ~/cicd - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка

igor@my1vm:~/cicd$ echo "test" >> test.txt
igor@my1vm:~/cicd$ git commit -m "Initial commit"
On branch master

Initial commit

Untracked files:
  test.txt

nothing added to commit but untracked files present
igor@my1vm:~/cicd$ git add .
igor@my1vm:~/cicd$ git commit -m "Initial commit"
[master (root-commit) 1865302] Initial commit
1 file changed, 1 insertion(+)
create mode 100644 test.txt
igor@my1vm:~/cicd$
```

git push -u origin master

```
igor@my1vm: ~/cicd - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка

igor@my1vm:~/cicd$ git push -u origin master
The authenticity of host 'gitlab.com (172.65.251.78)' can't be established.
ECDSA key fingerprint is SHA256:HbW3g8zUjNSksFbqTiUWPwg2Bq1x8xdGUrliXFzSnUw.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'gitlab.com,172.65.251.78' (ECDSA) to the list of known hosts.
Counting objects: 3, done.
Writing objects: 100% (3/3), 221 bytes | 221.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
remote:
remote: To create a merge request for master, visit:
remote:   https://gitlab.com/TolstikovIA/geekbrains/-/merge_requests/new?merge_request%5Bsource_branch%5D=master
remote:
remote: To gitlab.com:TolstikovIA/geekbrains.git
* [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
igor@my1vm:~/cicd$
```

git push -u origin main  
kubectl create ns gitlab

```
igor@my1vm: ~/cicd - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка

igor@my1vm:~/cicd$ git push -u origin main
error: src refspec main does not match any.
error: failed to push some refs to 'git@gitlab.com:TolstikovIA/geekbrains.git'
igor@my1vm:~/cicd$ kubectl create ns gitlab
namespace/gitlab created
igor@my1vm:~/cicd$
```

kubectl get ns

```
igor@my1vm: ~/cicd - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка

igor@my1vm:~/cicd$ kubectl get ns
NAME          STATUS   AGE
default       Active   15h
gitlab        Active   4m33s
kube-node-lease Active   15h
kube-public   Active   15h
kube-system   Active   15h
igor@my1vm:~/cicd$
```

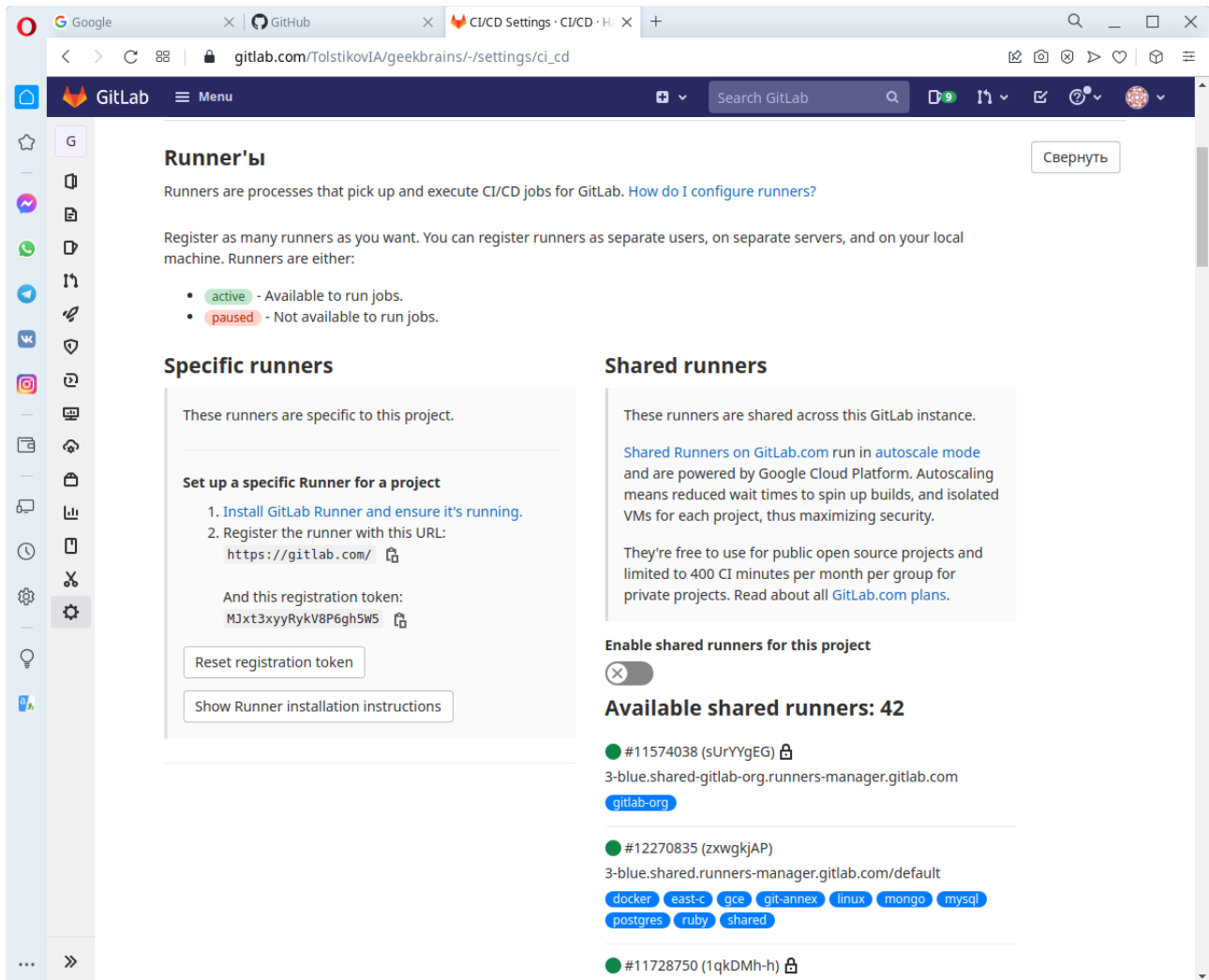
kubectl create ns stage  
kubectl create ns prod

```
igor@my1vm: ~/cicd - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка

igor@my1vm:~/cicd$ kubectl create ns stage
namespace/stage created
igor@my1vm:~/cicd$ kubectl create ns prod
namespace/prod created
igor@my1vm:~/cicd$
```

Shared runners

disable shared runners for this project



Set up a specific Runner for a project

And this registration token:

MJxt3xyyRykv8P6gh5W5

vi gitlab-runner.yaml

git add .

```

igor@my1vm: ~/geekbrains/practice/8.ci-cd/gitlab-runner - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
...
apiVersion: v1
kind: ServiceAccount
metadata:
  name: gitlab-runner
  labels:
    app: gitlab-runner
...
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: gitlab-runner
  labels:
    app: gitlab-runner
rules:
- apiGroups: ["*"]
  resources: ["*"]
  verbs: ["*"]
...
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: gitlab-runner
  labels:
    app: gitlab-runner
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: gitlab-runner
subjects:
- kind: ServiceAccount
  name: gitlab-runner
  namespace: gitlab
...
apiVersion: v1
kind: Secret
metadata:
  name: gitlab-runner
  labels:
    app: gitlab-runner
type: Opaque
stringData:
  runner-registration-token: MJxt3xyyRyKv8P6gh5W5
  runner-token: ""
...
apiVersion: v1
kind: ConfigMap
metadata:
  name: gitlab-runner
  labels:
    app: gitlab-runner
data:
  entrypoint: |
    #!/bin/bash
    set -e
    mkdir -p /home/gitlab-runner/.gitlab-runner/
    cp /scripts/config.toml /home/gitlab-runner/.gitlab-runner/

    # Register the runner
    if [[ -f /secrets/accesskey && -f /secrets/secretkey ]]; then
      export CACHE_S3_ACCESS_KEY=$(cat /secrets/accesskey)
      export CACHE_S3_SECRET_KEY=$(cat /secrets/secretkey)
    fi

    if [[ -f /secrets/gcs-application-credentials-file ]]; then
      export GOOGLE_APPLICATION_CREDENTIALS="/secrets/gcs-application-credentials-file"
    elif [[ -f /secrets/gcs-application-credentials-file ]]; then
      export GOOGLE_APPLICATION_CREDENTIALS="/secrets/gcs-application-credentials-file"
    else
      if [[ -f /secrets/gcs-access-id && -f /secrets/gcs-private-key ]]; then
        export CACHE_GCS_ACCESS_ID=$(cat /secrets/gcs-access-id)
        # echo -e used to make private key multiline (in google json auth key private key is oneline with \n)
        export CACHE_GCS_PRIVATE_KEY=$(echo -e $(cat /secrets/gcs-private-key))
      fi
    fi

    if [[ -f /secrets/runner-registration-token ]]; then
      export REGISTRATION_TOKEN=$(cat /secrets/runner-registration-token)
    fi

    if [[ -f /secrets/runner-token ]]; then
      export CI_SERVER_TOKEN=$(cat /secrets/runner-token)
    fi

    if ! sh /scripts/register-the-runner; then
      exit 1
    fi

    # Start the runner
    exec /entrypoint run --user-gitlab-runner \
      --working-directory=/home/gitlab-runner

  config.toml: |
    concurrent = 3
    check_interval = 30
    log_level = "info"
  configure: |
    set -e
    cp /init-secrets/* /secrets
  register-the-runner: |
    #!/bin/bash
    MAX_REGISTER_ATTEMPTS=30

    for i in $(seq 1 "${MAX_REGISTER_ATTEMPTS}"); do
      echo "Registration attempt ${i} of ${MAX_REGISTER_ATTEMPTS}"
      /entrypoint register \
        --non-interactive

      retval=$?

      if [ ${retval} = 0 ]; then
        break
      elif [ ${i} = ${MAX_REGISTER_ATTEMPTS} ]; then
        exit 1
      fi

      sleep 5
    done

    exit 0
:wd

```



```
igor@my1vm: ~/cicd - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
igor@my1vm:~/cicd$ git add .
igor@my1vm:~/cicd$ vi gitlab-runner.yaml
Error detected while processing /home/igor/.vimrc:
line 1:
E487: Argument must be positive: tabstop=0
Press ENTER or type command to continue
igor@my1vm:~/cicd$
```

kubectl apply --namespace gitlab -f gitlab-runner/gitlab-runner.yaml

```
igor@my1vm: ~/geekbrains/practice/8.ci-cd - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
igor@my1vm:~/geekbrains/practice/8.ci-cd/gitlab-runner$ cd ..
igor@my1vm:~/geekbrains/practice/8.ci-cd$ kubectl apply --namespace gitlab -f gitlab-runner/gitlab-runner.yaml
serviceaccount/gitlab-runner created
role.rbac.authorization.k8s.io/gitlab-runner created
rolebinding.rbac.authorization.k8s.io/gitlab-runner created
secret/gitlab-runner created
configmap/gitlab-runner created
deployment.apps/gitlab-runner created
igor@my1vm:~/geekbrains/practice/8.ci-cd$
```

kubectl apply -f deployment.yaml

kubectl apply -f ingress.yaml

kubectl apply -f service.yaml

```
igor@my1vm: ~/geekbrains/practice/8.ci-cd/app/kube - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
igor@my1vm:~/geekbrains/practice/8.ci-cd/app/kube$ kubectl apply -f deployment.yaml
deployment.apps/geekbrains created
igor@my1vm:~/geekbrains/practice/8.ci-cd/app/kube$ kubectl apply -f ingress.yaml
ingress.networking.k8s.io/geekbrains created
igor@my1vm:~/geekbrains/practice/8.ci-cd/app/kube$ kubectl apply -f service.yaml
service/geekbrains created
igor@my1vm:~/geekbrains/practice/8.ci-cd/app/kube$
```

kubectl apply -f statefulset.yaml

kubectl apply -f service.yaml

kubectl apply -f secret.yaml

```
igor@my1vm: ~/geekbrains/practice/8.ci-cd/app/kube/postgres - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
igor@my1vm:~/geekbrains/practice/8.ci-cd/app/kube/postgres$ kubectl apply -f statefulset.yaml
statefulset.apps/database created
igor@my1vm:~/geekbrains/practice/8.ci-cd/app/kube/postgres$ kubectl apply -f service.yaml
service/database created
igor@my1vm:~/geekbrains/practice/8.ci-cd/app/kube/postgres$ kubectl apply -f secret.yaml
secret/app created
igor@my1vm:~/geekbrains/practice/8.ci-cd/app/kube/postgres$
```

Доступные runner'ы

GitLab CI/CD Settings · CI/CD · H · X

gitlab.com/TolstikovIA/geekbrains/-/settings/ci\_cd

active - Available to run jobs.  
paused - Not available to run jobs.

### Specific runners

These runners are specific to this project.

**Set up a specific Runner for a project**

1. Install GitLab Runner and ensure it's running.
2. Register the runner with this URL:  
<https://gitlab.com/>

And this registration token:  
MJxt3xyyRykv8P6gh5W5

Reset registration token

Show Runner installation instructions

### Shared runners

These runners are shared across this GitLab instance.

[Shared Runners on GitLab.com](#) run in autoscale mode and are powered by Google Cloud Platform. Autoscaling means reduced wait times to spin up builds, and isolated VMs for each project, thus maximizing security.

They're free to use for public open source projects and limited to 400 CI minutes per month per group for private projects. Read about all [GitLab.com plans](#).

**User validation required**

Shared runners cannot be enabled until a valid credit card is on file.

Validate account

### Доступные специфические runner'ы

#12373398 (zNHTu4kf) Remove runner

gitlab-runner-7467648488-lgtzc

### Available shared runners: 42

#11574038 (sUrYYgEG)

3-blue.shared-gitlab-org.runners-manager.gitlab.com

gitlab-org

#11574076 (8zCxmPt)

2-green.shared-gitlab-org.runners-manager.gitlab.com

```
igor@my1vm: ~/cicd - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
igor@my1vm:~/geekbrains/practice/8.ci-cd$ cd ~/cicd
igor@my1vm:~/cicd$ kubectl get ns
NAME          STATUS   AGE
default       Active   18h
gitlab        Active   3h2m
kube-node-lease Active   18h
kube-public   Active   18h
kube-system   Active   18h
prod          Active   167m
stage         Active   167m
igor@my1vm:~/cicd$
```

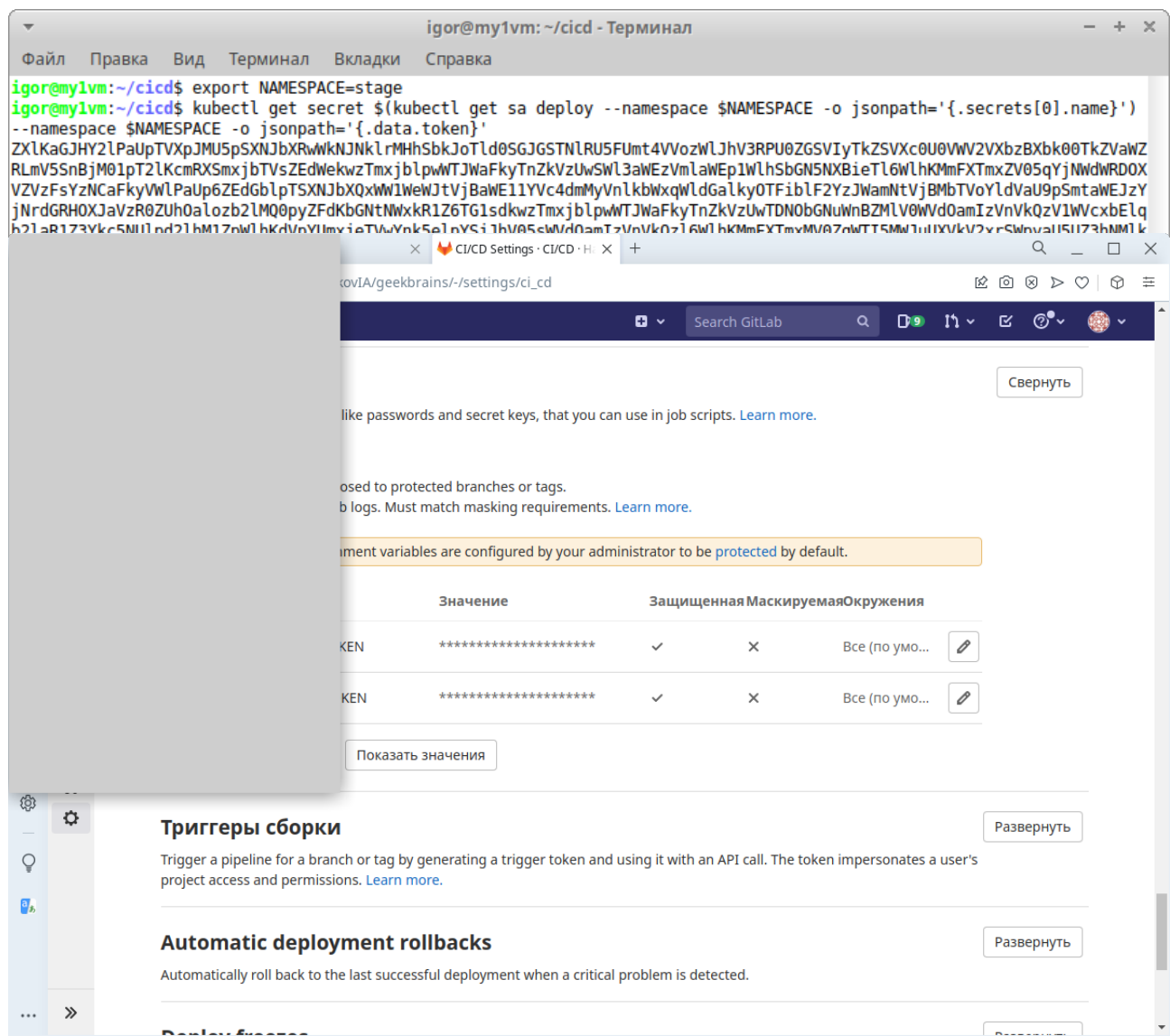
```
kubectl create sa deploy --namespace stage
kubectl create rolebinding deploy --serviceaccount stage:deploy --clusterrole edit --namespace stage
kubectl create sa deploy --namespace prod
kubectl create rolebinding deploy --serviceaccount prod:deploy --clusterrole edit --namespace prod
```

```
igor@my1vm: ~/cicd - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
igor@my1vm:~/cicd$ kubectl create sa deploy --namespace stage
serviceaccount/deploy created
igor@my1vm:~/cicd$ kubectl create rolebinding deploy --serviceaccount stage:deploy --clusterrole edit --namespace stage
rolebinding.rbac.authorization.k8s.io/deploy created
igor@my1vm:~/cicd$ kubectl create sa deploy --namespace prod
serviceaccount/deploy created
igor@my1vm:~/cicd$ kubectl create rolebinding deploy --serviceaccount prod:deploy --clusterrole edit --namespace prod
rolebinding.rbac.authorization.k8s.io/deploy created
igor@my1vm:~/cicd$
```



```
-namespace $NAMESPACE -o jsonpath='{.data.token}'
ZXlKaGJHY2lPaUpTVXpJMU5pS3NjbnkXRwWkNjNklrMHhSbkJoTld0SGJGJSTNlRU5FUmt4VVoZWlJhV3RPU0ZGSVlYTkZSVXc0U0VWV2V
XbzBxbXk00TkZVaWZRLmV3NjBjM0pTl2lkmRXSMxjbZVsZEdWekwzTmxjblpwWTJwFakYfnZkVzUwSWl3aWeZVmlaWepC1WlhSBGn5N
XbieTl6WlhKmmFXTmxZV05qYjNwDWRDOXVZvZfYzNcAkfyWlPaU6ZEdGblpTSXNjBXqXww1WeWJrVjBaWe11YVc4dmMyVnlbk
WxqWldGalkyOTFiblf2YzJwamNtVjBmbTVoYldVaU9pSmtaWEJzYjNrdGRHOXJaVzR0ZUhoalozb2lMQ0pyZFdkBGNtNWxkR1Z6TG1sdkwz
TmxjblpwWTJwFakYfnZkVzUwTDNObGnuWnBZMlVOWVdOamIzVnVkQzYjV1WVcxblq62laR1Z3Ykc5Nulpd2lhM1ZpWlhKdVpYUmxeTV
wYnk5elpYjVjV05swVdOamIzVnVkQz6WlhKmmFXTmxMv0ZqWTl5MwJuUXkVz2XrSWpwaU5UZ3hNMlK1WvdVdIEY3lOQzAwWmpa
a0xSM1S1Zl1l0WkRVE5EQTR0bVE0WVpVNElpd2lMz1PZWpwaVMzbHpkR1Z0T25ObGnuWnBZMlZoWTJ0dMRXNTBpbk4WvdYdkbE9T
amXjR3h2ZVZVnKOS5ReFgV2m8bZbqZIV5bW43ZE5WeThfNndRNlRWtlBONkxYSWMxNEl2OFM2aEd0a1hxS18tZGhCQk5Sc1NNmmV5Tn
JVZFBVaUgwQ2tVeUlxN0tOQzLSU09LTXA0eGdzbmZKAepPvklISW9DYzJ2a1h0T0lZTE95eXcwR2dCWjFGSjNiX20tTkpLTGYxwUN5ZGdhZ
FY3YU01Mmx4NkpTZ2Y3TGpKTXlVSi1NRjV5NjhMNEFoWlZuRmtBUnR1RfFVZ3haN0p6THVVOQmFHXzk4UINlN1lxSGRKb3R6cUx2eXh
WwFYtZmtVLVMzMYRlQy0hXmDydgF4dHdxUEm5aVfXVWdINTfOEJlS18E0Vb0YnFkrjhlT1VyMFZSNzhNdHhCcGjXWnMwVnp6YU
ZiY0f1eVVBajRTYMS53Bjb19MDNepzQndaa1hPaHRrTZWZr18vSHBaODYS1ee=ig
```

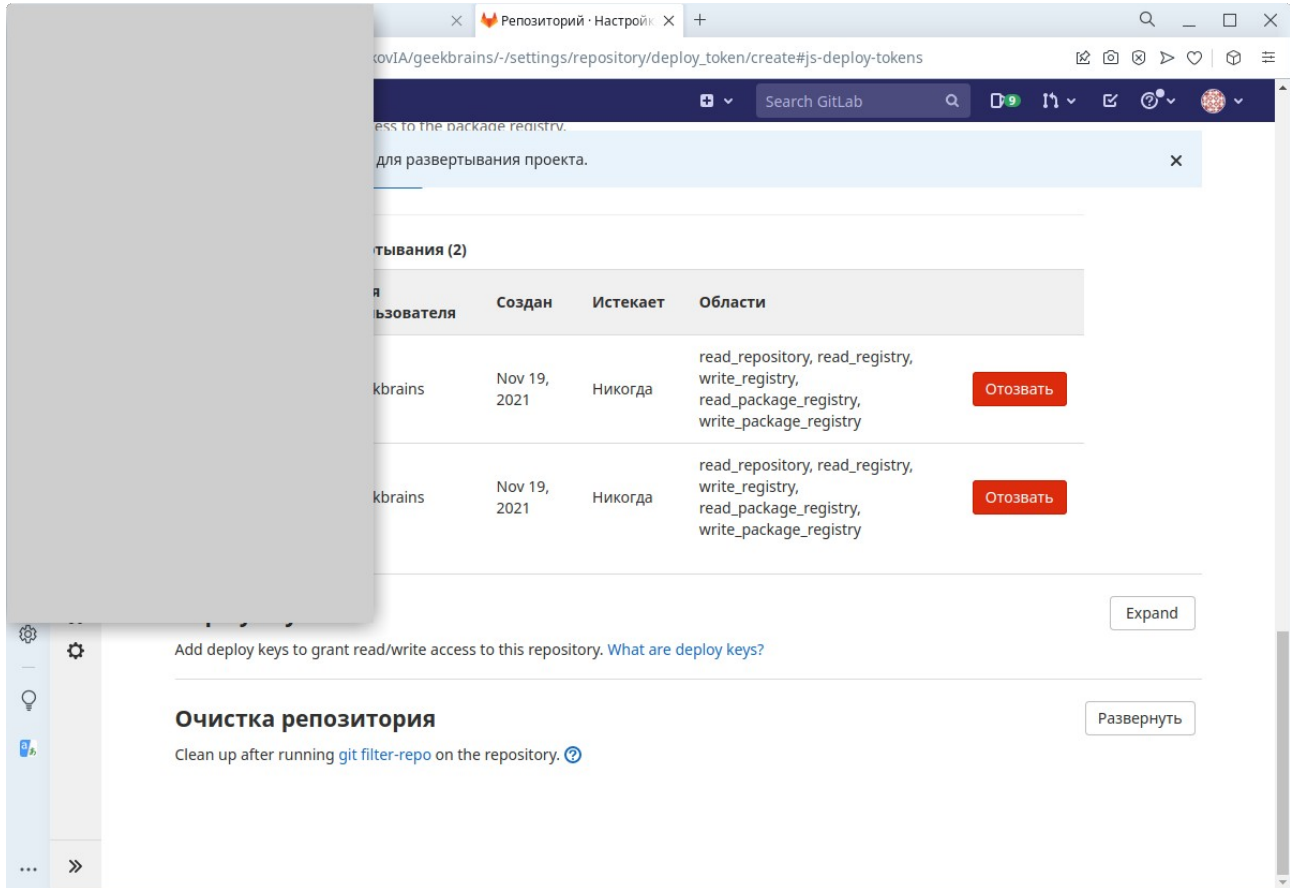
ZXlKaGJHY2lPaUpTVXpJMU5pSXNjbXRwWkNjNklrMHhSbkJoTld0SGJSTNlRU5FUmt4VVozWlJhV3RPU0ZGSVlyTkZSVXc0U0VWV2VXbzBXbk00TkZVaWZRLmV5SNBjM01pT2lKcmRXSxmjbTVsZEdWekwzTmxjblpwWTJWafKyTnZkVzUwSWl3aWEzVmlaWEp1WlhSbGN5NXBieTl6WlhKmmfXTmxZV05qYjNwdWRDOXVZVzFsYzNCaFkyVWlPaUp3Y205a0lpd2lhM1ZpWlhkdVpYUmxjeTVwYnk5elpYSljhV05sWVdOamIzVnVqKzI6WldOeVpYUxVibUZ0WlNjNklrUmXjR3h2ZVMxMGIydGxiaTB5WkhOM2JTSXNjBxQxJW1WeWJtVjBaWE11YVvc4dmMvYnIkbGxqWldGalkyOTfIblF2YzJWeWrtbGma05ZfoFkYJODmRXNTBmMbTVoYldVaU9pSmtaWEJzJYjNraUxDSnJkV0psY201bGRHVmFbWbx2TDNOBwNuWnBZMlOTfWJODmRXNTBmMaUz52aefkYVXRZV05qYjNwdWRDNTp1FpT2lJmI6QTRzakprTlMweU1tWlXIMVfJtTWJ1dFIUUXdZeTfRtURSa1pqa3lZbUU1T1dZaUxDSnpkV0lpT2lKcmVYTjBaVzA2YzJWeWrtbGpaV0ZqWTi5MWJuUTZjSEp2WkRwa1pYQnNiM2tpZlEuRmpoa2FkSWhGOFFKS1Vfa0hxZXpHZ0VwTkQ1WldUcHataTRVY0xISG1UVFd1emprZVR2d0lKzk5LMjR3cmqtQVdaWFFSblJlRkNkKckNTdmkxZEtbC0Y4OTRxbTgxNEVzZm5vWmlvdFNZU1hVM3fwcVZub21PWENfa0dnZINOZWozZUZUBWdlvQU9NRmdPZTN1OXNOVThtemV6Uk1KQUZuR2JMHNdjRtR2RkCFZwbNzQZ0NXRFdna2hHTmNjYORHeTcyOHdDNnYwdVU3c0dEbzUyTlV1aVhErmlUdGpVm2ctN0dpbXBXMtZOcj1YzVWSFF5tk1PR21DaW5VRzJORHwREjC3Q4dE3e1cVlaRm5tbG9wczdFWi1BTEU5R2VscVwVbkNjLUaNa2FqcDFXSkd2YnhBa2ludTlwb1dKU0UzOqLzb2VZG2TmkvcVJ2bDJUigor@mv14m



## Deploy tokens

geekbrains stage  
nhYE8mUscBo2vNbc1qzL

geekbrains prod  
G2md2EAgR6EPsGGxkM5\_



```
kubectl create secret docker-registry gitlab-registry --docker-server=registry.gitlab.com --docker-username=<USERNAME> --docker-password=<PASSWORD> --docker-email=admin@admin.admin --namespace stage
```

```
kubectl create secret docker-registry gitlab-registry --docker-server=registry.gitlab.com --docker-username=<USERNAME> --docker-password=<PASSWORD> --docker-email=admin@admin.admin --namespace prod
```

```
kubectl create secret docker-registry gitlab-registry --docker-server=registry.gitlab.com --docker-username=geekbrains --docker-password=nhYE8mUscBo2vNbc1qzL --docker-email=admin@admin.admin --namespace stage
```

```
kubectl create secret docker-registry gitlab-registry --docker-server=registry.gitlab.com --docker-username=geekbrains --docker-password=G2md2EAgR6EPsGGxkM5_ --docker-email=admin@admin.admin --namespace prod
```

```
igor@my1vm: ~/cicd - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
igor@my1vm:~/cicd$ kubectl create secret docker-registry gitlab-registry --docker-server=registry.gitlab.com --docker-username=geekbrains --docker-password=nHYE8mUscBo2vNbc1qzL --docker-email=admin@admin.admin --namespace stage
secret/gitlab-registry created
igor@my1vm:~/cicd$ kubectl create secret docker-registry gitlab-registry --docker-server=registry.gitlab.com --docker-username=geekbrains --docker-password=G2md2EAgR6EPsGGxkM5_ --docker-email=admin@admin.admin --namespace prod
secret/gitlab-registry created
igor@my1vm:~/cicd$
```

kubectl patch serviceaccount default -p '{"imagePullSecrets": [{"name": "gitlab-registry"}]}' -n stage

kubectl patch serviceaccount default -p '{"imagePullSecrets": [{"name": "gitlab-registry"}]}' -n prod

```
igor@my1vm: ~/cicd - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
igor@my1vm:~/cicd$ kubectl patch serviceaccount default -p '{"imagePullSecrets": [{"name": "gitlab-registry"}]}' -n stage
serviceaccount/default patched
igor@my1vm:~/cicd$ kubectl patch serviceaccount default -p '{"imagePullSecrets": [{"name": "gitlab-registry"}]}' -n prod
serviceaccount/default patched
igor@my1vm:~/cicd$
```

kubectl apply --namespace stage -f app/kube/postgres/

kubectl apply --namespace prod -f app/kube/postgres/

```
igor@my1vm: ~/geekbrains/practice/8.ci-cd - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
igor@my1vm:~/geekbrains/practice/8.ci-cd$ kubectl apply --namespace stage -f app/kube/postgres/
secret/app created
service/database created
statefulset.apps/database created
igor@my1vm:~/geekbrains/practice/8.ci-cd$ kubectl apply --namespace prod -f app/kube/postgres/
secret/app created
service/database created
statefulset.apps/database created
igor@my1vm:~/geekbrains/practice/8.ci-cd$
```

vi app/kube/ingress.yaml

```
igor@my1vm: ~/geekbrains/practice/8.ci-cd - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: geekbrains
  annotations:
    kubernetes.io/ingress.class: nginx-external
spec:
  rules:
  - host: postgres.stage.info
    http:
      paths:
      - path: /users
        pathType: Prefix
        backend:
          service:
            name: prometheus
            port:
              number: 8080
:wq
```

kubectl apply --namespace stage -f app/kube

```
igor@my1vm: ~/geekbrains/practice/8.ci-cd - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
igor@my1vm:~/geekbrains/practice/8.ci-cd$ kubectl apply --namespace stage -f app/kube
deployment.apps/geekbrains created
ingress.networking.k8s.io/geekbrains created
service/geekbrains created
igor@my1vm:~/geekbrains/practice/8.ci-cd$
```

vi app/kube/ingress.yaml

```
igor@my1vm: ~/geekbrains/practice/8.ci-cd - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: geekbrains
  annotations:
    kubernetes.io/ingress.class: nginx-external
spec:
  rules:
  - host: postgres.prod.info
    http:
      paths:
      - path: /users
        pathType: Prefix
        backend:
          service:
            name: prometheus
            port:
              number: 8080
~
-- INSERT --                                     9,26      All
```

kubectl apply --namespace prod -f app/kube

```
igor@my1vm: ~/geekbrains/practice/8.ci-cd - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
igor@my1vm:~/geekbrains/practice/8.ci-cd$ kubectl apply --namespace prod -f app/kube
deployment.apps/geekbrains created
ingress.networking.k8s.io/geekbrains created
service/geekbrains created
igor@my1vm:~/geekbrains/practice/8.ci-cd$
```

Проверяем работу приложения

kubectl get svc -A

```
igor@my1vm: ~/geekbrains/practice/8.ci-cd - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
igor@my1vm:~/geekbrains/practice/8.ci-cd$ kubectl get svc -A
NAMESPACE   NAME           TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
default     kubernetes     ClusterIP   10.96.0.1    <none>        443/TCP          20h
kube-system kube-dns        ClusterIP   10.96.0.10   <none>        53/UDP,53/TCP,9153/TCP  20h
prod        database        ClusterIP   10.108.184.143 <none>        5432/TCP          113m
prod        geekbrains      ClusterIP   10.106.110.116 <none>        8000/TCP          88m
stage       database        ClusterIP   10.100.246.195 <none>        5432/TCP          113m
stage       geekbrains      ClusterIP   10.101.30.120 <none>        8000/TCP          91m
igor@my1vm:~/geekbrains/practice/8.ci-cd$
```

kubectl get svc -A

kubectl get pod

```
igor@my1vm: ~/geekbrains/practice/8.ci-cd/app/kube/postgres - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка

igor@my1vm:~/geekbrains/practice/8.ci-cd/app/kube/postgres$ kubectl get svc -A
NAMESPACE      NAME          TYPE        CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
default        database      ClusterIP    10.108.221.177   <none>           5432/TCP          21m
default        geekbrains    ClusterIP    10.98.189.227    <none>           8000/TCP          23m
default        kubernetes    ClusterIP    10.96.0.1        <none>           443/TCP           22h
kube-system     kube-dns       ClusterIP    10.96.0.10       <none>           53/UDP,53/TCP,9153/TCP  22h
prod            database      ClusterIP    10.108.184.143   <none>           5432/TCP          3h14m
prod            geekbrains    ClusterIP    10.106.110.116   <none>           8000/TCP          169m
stage           database      ClusterIP    10.100.246.195    <none>           5432/TCP          3h15m
stage           geekbrains    ClusterIP    10.101.30.120     <none>           8000/TCP          172m

igor@my1vm:~/geekbrains/practice/8.ci-cd/app/kube/postgres$ kubectl get pod
NAME                READY   STATUS    RESTARTS   AGE
database-0          0/1     Pending   0           22m
geekbrains-55df48c88d-c5lfs  1/1     Running   0           25m
geekbrains-55df48c88d-zl42h  1/1     Running   0           25m
igor@my1vm:~/geekbrains/practice/8.ci-cd/app/kube/postgres$
```

kubectl get pods --all-namespaces

```
igor@my1vm: ~/geekbrains/practice/8.ci-cd/app/kube - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка

igor@my1vm:~/geekbrains/practice/8.ci-cd/app/kube$ kubectl get pods --all-namespaces
NAMESPACE      NAME                                                    READY   STATUS    RESTARTS   AGE
default        database-0                                              0/1     Pending   0           16m
default        geekbrains-55df48c88d-c5lfs                            1/1     Running   0           68m
default        geekbrains-55df48c88d-zl42h                            1/1     Running   0           68m
gitlab          gitlab-runner-7467648488-lgtzc                       1/1     Running   0           4h46m
kube-system     coredns-78fcd69978-s4whs                             1/1     Running   1 (22h ago)  22h
kube-system     etcd-minikube                                          1/1     Running   1 (22h ago)  22h
kube-system     kube-apiserver-minikube                               1/1     Running   1 (22h ago)  22h
kube-system     kube-controller-manager-minikube                     1/1     Running   1 (22h ago)  22h
kube-system     kube-proxy-s7cmd                                       1/1     Running   1 (22h ago)  22h
kube-system     kube-scheduler-minikube                               1/1     Running   1 (22h ago)  22h
kube-system     storage-provisioner                                    1/1     Running   3           22h
prod            database-0                                              0/1     Pending   0           3h57m
prod            geekbrains-55df48c88d-cst9d                            1/1     Running   0           3h32m
prod            geekbrains-55df48c88d-gj8sz                            1/1     Running   0           3h32m
stage           database-0                                              0/1     Pending   0           3h57m
stage           geekbrains-55df48c88d-9hznh                            1/1     Running   0           3h35m
stage           geekbrains-55df48c88d-ghnfl                            1/1     Running   0           3h35m
igor@my1vm:~/geekbrains/practice/8.ci-cd/app/kube$
```

kubectl top pod