

Deploy

1. Приведите файл `.gitlab-ci.yml` в скопированном fork'e хрaste к виду

Адрес Вашего первого мастера можно посмотреть при помощи команды

```
kubectl cluster-info
```

выполненной на этом первом мастере

```
variables:
```

```
  K8S_API_URL: https://172.<xx>.<yyy>.2:6443
```

```
stages:
```

- build
- test
- cleanup
- push
- deploy

```
build:
```

```
  stage: build
```

```
  script:
```

- docker build -t

```
$CI_REGISTRY/$CI_PROJECT_NAMESPACE/$CI_PROJECT_NAME:$CI_COMMIT_REF_SLUG.$CI_PIPELINE_ID .
```

```
test:
```

```
  stage: test
```

```
  image:
```

```
    name: docker/compose:1.23.2
```

```
    entrypoint: [""]
```

```
  script:
```

- docker-compose
- p "\$CI_PROJECT_NAME" "\$CI_PIPELINE_ID"
- up

```
  --abort-on-container-exit
```

```
  --exit-code-from app
```

```
  --quiet-pull
```

```
cleanup:
```

```
  stage: cleanup
```

```
  image:
```

```
    name: docker/compose:1.23.2
```

```
    entrypoint: [""]
```

```

script:
  - docker-compose -p "$CI_PROJECT_NAME"_"$CI_PIPELINE_ID" down
when: always

push:
  stage: push
  before_script:
    - docker login -u gitlab-ci-token -p $CI_BUILD_TOKEN $CI_REGISTRY
  script:
    - docker push
$CI_REGISTRY/$CI_PROJECT_NAMESPACE/$CI_PROJECT_NAME:$CI_COMMIT_REF_SLUG.$CI_PIPELINE_ID
  only:
    - master

deploy:
  stage: deploy
  image: centosadmin/kubernetes-helm:3.1.2
  environment:
    name: production
  script:
    - kubectl config set-cluster k8s --insecure-skip-tls-verify=true
--server=$K8S_API_URL
    - kubectl config set-credentials ci --token=$K8S_CI_TOKEN
    - kubectl config set-context ci --cluster=k8s --user=ci
    - kubectl config use-context ci
    - helm upgrade --install $CI_PROJECT_PATH_SLUG .helm
      --set image=$CI_REGISTRY/$CI_PROJECT_NAMESPACE/$CI_PROJECT_NAME
      --set imageTag=$CI_COMMIT_REF_SLUG.$CI_PIPELINE_ID
      --wait
      --timeout 300s
      --atomic
      --debug
      --namespace $CI_PROJECT_PATH_SLUG-$CI_ENVIRONMENT_NAME
  only:
    - master

```

2. Откройте файл `.helm/values.yaml` и в нем отредактируйте

```
ingress:
  host: xpaste.s<номер своего логина>.edu.slurm.io
```

заменив `<>` на свой номер логина

3. Добавьте изменения в git, сделайте `commit` и `push`'те в gitlab

```
git add .
```

```
git commit -m "add deploy stage"
```

```
git push
```

4. Откройте в браузере Ваш **fork** проекта, перейдите на вкладку **CI/CD -> Pipelines** в меню справа и дождитесь завершения всех шагов.

5. **Откройте** приложение в браузере по адресу **xpaste.s<номер своего логина>.edu.slurm.io** и убедитесь, что оно работает.

Открывайте приложение в режиме "инкогнито", т.к. в обычном режиме браузер блокирует доступ по http. В Google Chrome к примеру, можно запустить режим с помощью комбинации "Ctrl+Shift+N".