
Архитектура распределённых приложений

Лабораторная работа №3

Выполнил: студент группы ИП-312

Дорогин Н. С.

Проверил: преподаватель Кафедры вычислительных систем

Челканова Т.В.

Задание:

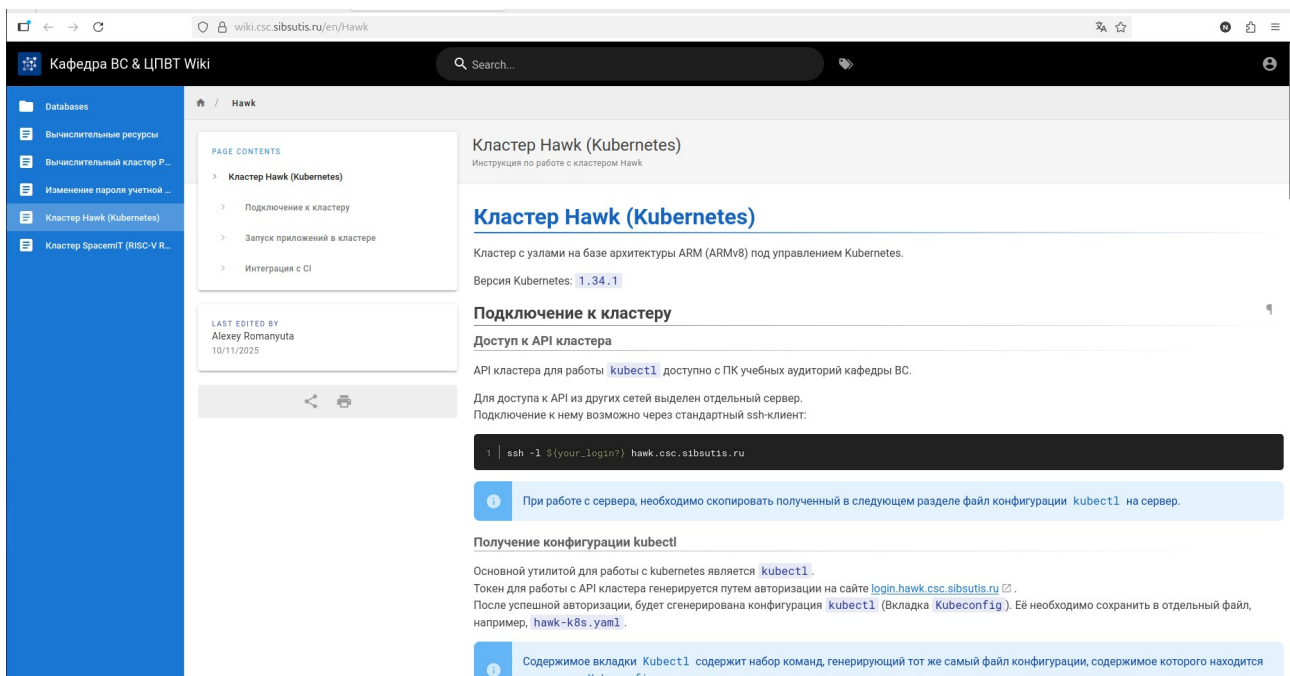
1. Получить доступ к API кластера Hawk, получить конфигурацию kubectl.
 2. Настроить окружение на сервере или ПК.
 3. Доработать CI для загрузки образа приложения в реестр кластера Hawk. Использовать для этого переменные **HAWK_***:
 - **HAWK_REGISTRY** - аналог **CI_REGISTRY**
 - **HAWK_REGISTRY_IMAGE** - аналог **CI_REGISTRY_IMAGE**
 - **HAWK_REGISTRY_USER** - аналог **CI_REGISTRY_USER**
 - **HAWK_REGISTRY_PASSWORD** - аналог **CI_REGISTRY_PASSWORD**
 4. Написать набор манифестов kubernetes для развертывания приложения в кластере.
 5. Продемонстрировать работу приложения.
-

Ход работы:

1, 2) Берём из лекции QR-код открытия вики с информацией о кластере:

Hawk

- hawk.csc.sibsutis.ru
- Кластер (на момент написания) из 7 узлов: 3 мастера, 4 воркера
- Для работы с API выделен Access-сервер
- API закрыт сетью кластера и сетью Кафедры ВС.



The screenshot shows a web browser displaying the 'Hawk' page on the 'Кафедра ВС & ЦПБТ Wiki'. The page title is 'Кластер Hawk (Kubernetes)' with the subtitle 'Инструкция по работе с кластером Hawk'. The main content includes a section 'Кластер Hawk (Kubernetes)' describing it as an ARM-based cluster managed by Kubernetes, version 1.34.1. It details the connection process, API access, and provides a terminal command: `ssh -l $(your_login?) hawk.csc.sibsutis.ru`. It also mentions that a configuration file for kubectl is generated and stored in a 'Kubeconfig' tab. The left sidebar shows the site's navigation menu, and the top bar includes a search function.

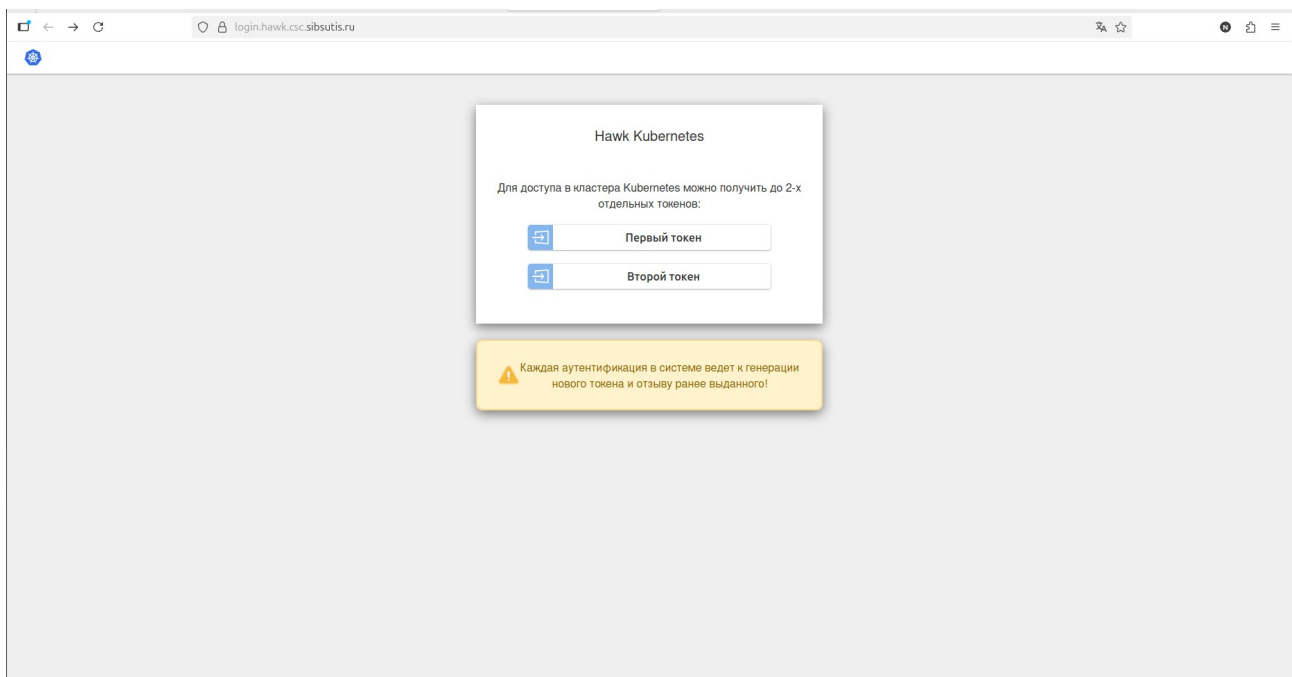
Получаем доступ к API кластера:

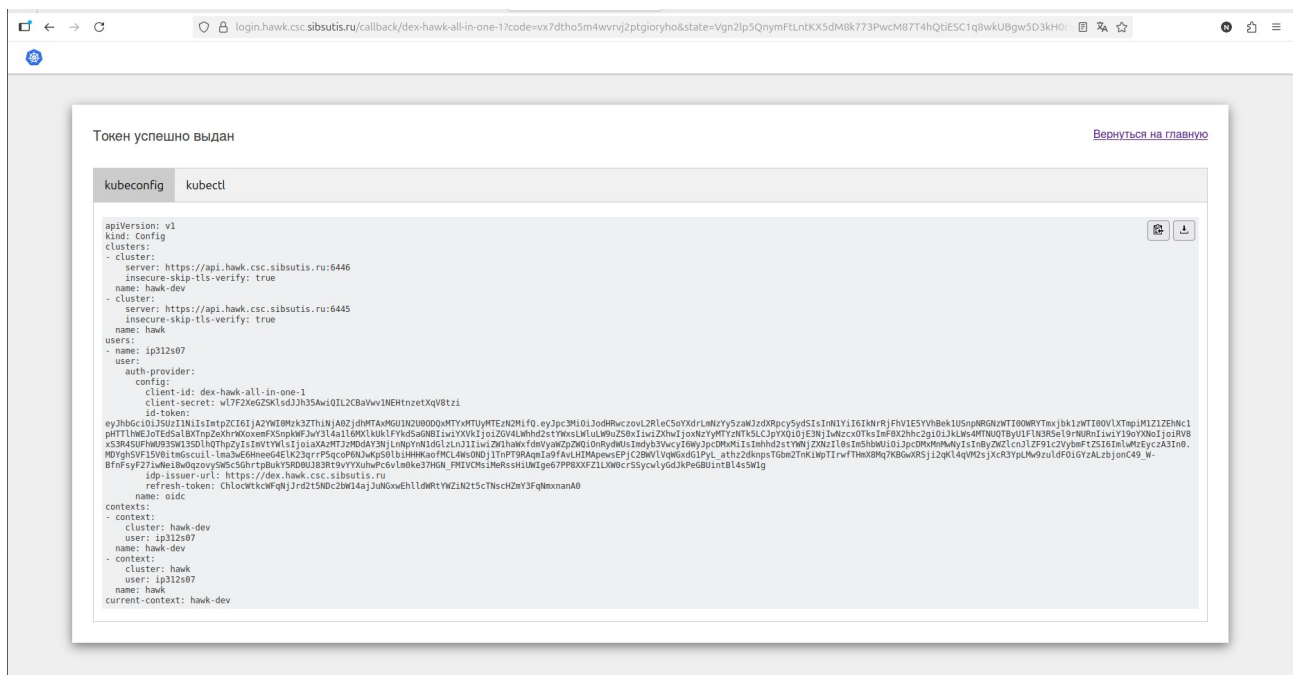
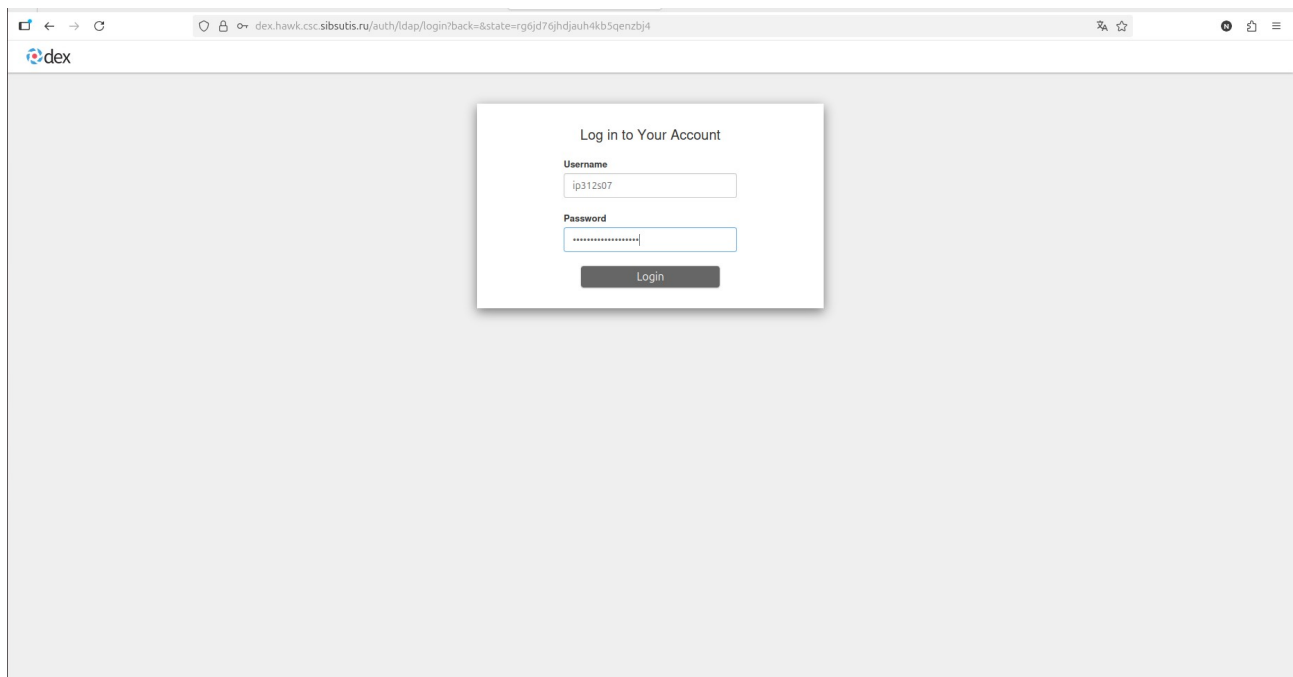
```
gastello123@gastello123:~/Desktop/my-simple-distributed-application$ ssh -l ip312s07 hawk.csc.sibsutis.ru
Authorized uses only. All activity may be monitored and reported.
(ip312s07@hawk.csc.sibsutis.ru) Password:
Creating directory '/home/ipa/ip312s07'.
Users online:

csc      pts/0      Nov  2 20:01 (37.195.133.203)
csc      pts/1      Nov  2 20:01 (37.195.133.203)
romanutaaa pts/2      Nov  2 20:08 (37.195.133.203)
romanutaaa pts/3      Nov  2 20:09 (37.195.133.203)
romanutaaa pts/4      Nov  2 20:11 (91.196.245.200)
romanutaaa pts/5      Nov  2 20:11 (91.196.245.200)
ip312s07@hawk-access:~$
```

(На этом этапе имела место проблема с синхронизацией времени на сервере, хранящем учётные записи из-за которой не проходили правильный логин и пароль — те же, что и от GitLab. Для решения этой проблемы пришлось обратиться к администратору)

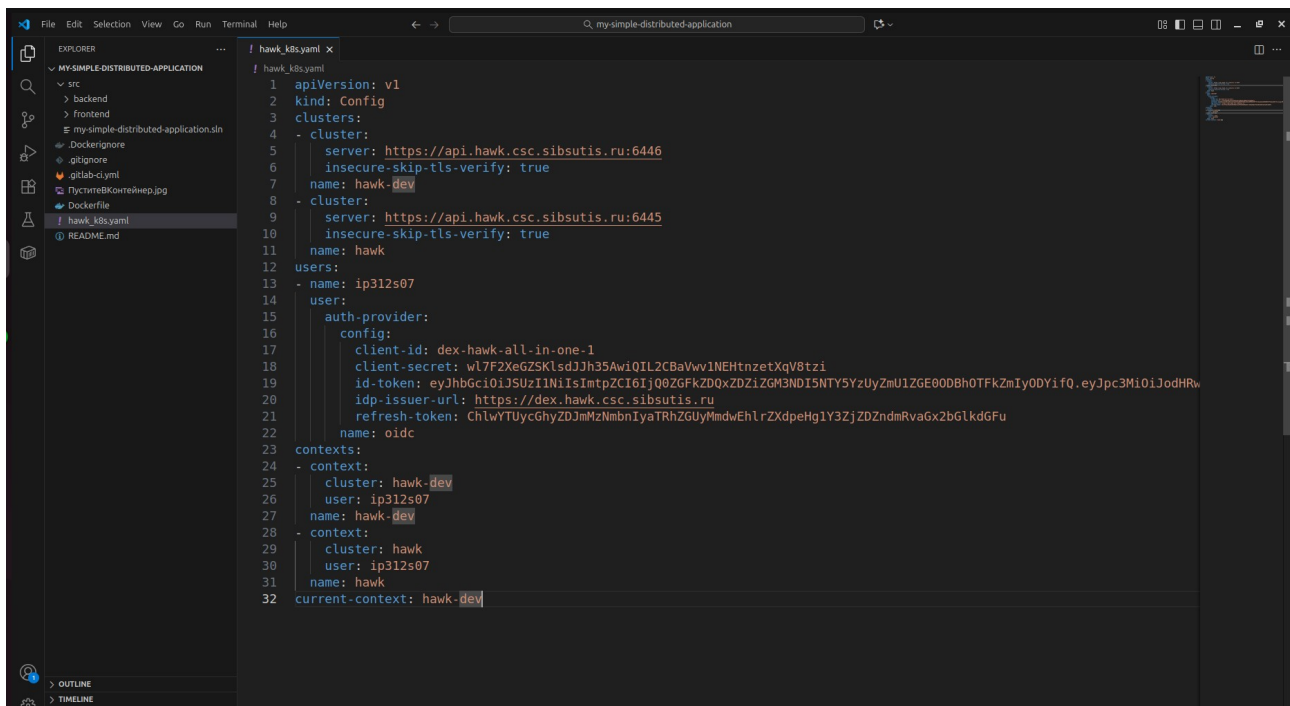
Получаем конфигурацию (я выбираю первый токен):





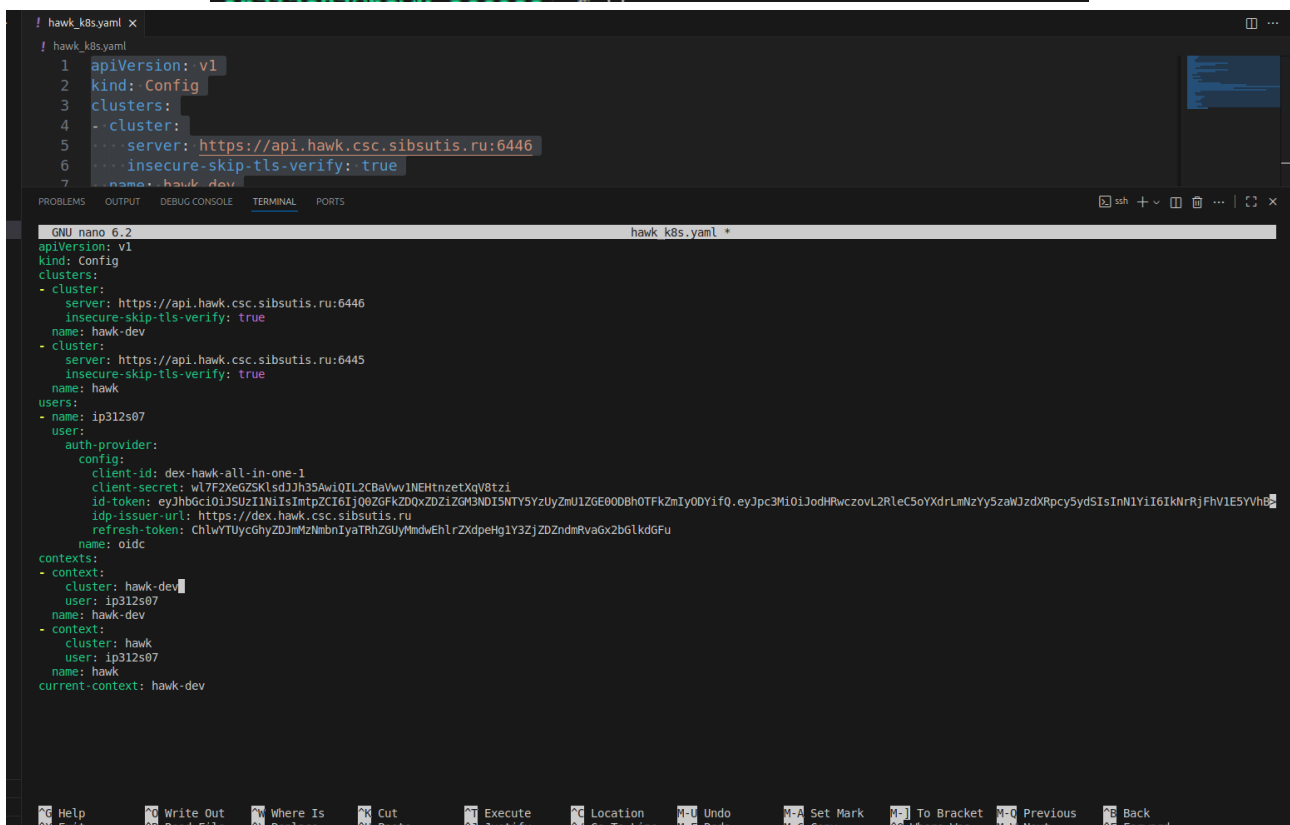
Можем выбрать любой файл из двух предложенных, они оба делают одно и то же, я беру kubeconfig.

Копируем конфигурацию и вставляем её в yaml файл:



Теперь, возвращаясь к командной строке для взаимодействия с сервером, создаём уже на сервере файл конфигурации и вставляем туда то же самое:

```
ip312s07@hawk-access:~$ nano hawk_k8s.yaml
```



Теперь мы можем пользоваться утилитой kubectl:

```
ip312s07@hawk-access:~$ ls
hawk k8s.yaml
ip312s07@hawk-access:~$ kubectl --kubeconfig hawk_k8s.yaml -n ip312 get pods
NAME                                READY   STATUS              RESTARTS   AGE
ip312s29-app-6b77759745-gh67z      0/1     ImagePullBackOff    0           2d1h
ip312s29-app-8644c796f4-q8gsj      0/1     ImagePullBackOff    0           2d1h
ip312s07@hawk-access:~$
```

Сохраняем конфиг, чтобы каждый раз не прописывать его в команде:

```
ip312s07@hawk-access:~$ export KUBECONFIG=hawk_k8s.yaml
ip312s07@hawk-access:~$ kunectl -n ip312 get pods
-bash: kunectl: command not found
ip312s07@hawk-access:~$ kubectl -n ip312 get pods
NAME                                READY   STATUS              RESTARTS   AGE
ip312s29-app-6b77759745-gh67z      0/1     ImagePullBackOff    0           2d1h
ip312s29-app-8644c796f4-q8gsj      0/1     ImagePullBackOff    0           2d1h
ip312s07@hawk-access:~$
```


3) Доработанный CI для загрузки образов в реестр кластера Hawk:

```
gitlab-ci.yml
1  stages:
2    - build
3
4
5  ci-job:
6
7    stage: build
8
9    image: docker:28.1
10
11    services:
12      - docker:28.1-dind
13
14
15    before_script:
16      - echo "$CI_REGISTRY_PASSWORD" | docker login -u "$CI_REGISTRY_USER" --password-stdin "$CI_REGISTRY"
17      - docker context create ci-context
18      - docker context use ci-context
19      - docker buildx create --name multiarch-builder --driver docker-container --use ci-context
20
21    script:
22      - echo "Building..."
23      - |
24          if [ -n "$CI_COMMIT_TAG" ]; then
25            IMAGE_TAG=$CI_COMMIT_TAG
26          else
27            IMAGE_TAG=${CI_COMMIT_SHORT_SHA:0:8}
28          fi
29      - echo "READING THE IMAGE TAG"
30
31      - docker buildx build --platform linux/amd64,linux/arm64 -t ${CI_REGISTRY_IMAGE}:${IMAGE_TAG} --push .
32      - echo "${CI_REGISTRY_IMAGE}:${IMAGE_TAG} WAS SUCCESSFULLY BUILT AND PUSHED IN GITLAB"
33
34
35    rules:
36      - if: $CI_COMMIT_TAG
37        when: always
38      - when: manual
39
```

```

40 hawk-job:
41   stage: build
42
43   image: docker:28.1
44
45   services:
46   - docker:28.1-dind
47
48   before_script:
49   - echo "$HAWK_REGISTRY_PASSWORD" | docker login -u "$HAWK_REGISTRY_USER" --password-stdin "$HAWK_REGISTRY"
50   - docker context create hawk-context
51   - docker context use hawk-context
52   - docker buildx create --name multiarch-builder --driver docker-container --use hawk-context
53   script:
54   - |
55     if [ -n "$CI_COMMIT_TAG" ]; then
56       IMAGE_TAG=$CI_COMMIT_TAG
57     else
58       IMAGE_TAG=${CI_COMMIT_SHORT_SHA:0:8}
59     fi
60     echo "READING THE IMAGE TAG"
61
62   - docker buildx build --platform linux/amd64,linux/arm64 -t ${HAWK_REGISTRY_IMAGE}:${IMAGE_TAG} --push .
63   - echo "${HAWK_REGISTRY_IMAGE}:${IMAGE_TAG} WAS SUCCESFULLY PUSHED TO HAWK"
64
65   rules:
66   - if: $CI_COMMIT_TAG
67     when: always
68   - when: manual
69

```

Commit 7b839210 

Try to change dotnet image 2

Pipeline #48155  Passed for lab3



build 

Related jobs

-  ci-job
-  hawk-job

По сути он делает всё то же самое, что и CI из второй лабораторной работы, только два раза: для GitLab и для Hawk.

4) Манифесты для развертывания приложения в кластере

my_StatefulSet.yaml

```
GNU nano 6.2 my_StatefulSet.yaml
apiVersion: apps/v1
kind: StatefulSet
metadata:
  labels:
    app: ip312s07-app
    owner: ip312s07
  name: ip312s07-app
  namespace: ip312
spec:
  replicas: 1
  revisionHistoryLimit: 10
  selector:
    matchLabels:
      app: ip312s07-app
  serviceName: ip312s07-service
  template:
    metadata:
      labels:
        app: ip312s07-app
        owner: ip312s07
    spec:
      containers:
        - name: ip312s07-app
          image: cr.hawk.csc.sibsis.ru/hawk-dev/ip312s07-my-simple-distributed-application:adb493a0
          command: ["dotnet", "/app/Backend.dll"]
          imagePullPolicy: IfNotPresent
          ports:
            - name: http
              containerPort: 5000
          securityContext:
            capabilities:
              drop: ["ALL"]
            drop: ["ALL"]
            allowPrivilegeEscalation: false
      resources:
        requests:
          cpu: 32m
          memory: 16Mi
        limits:
          cpu: 64m
          memory: 32Mi
      readinessProbe:
        httpGet:
          path: /
          port: 5000
        initialDelaySeconds: 15
        periodSeconds: 10
        failureThreshold: 3
      livenessProbe:
        httpGet:
          path: /
          port: 5000
        initialDelaySeconds: 30
        periodSeconds: 10
        failureThreshold: 3
      updateStrategy:
        type: RollingUpdate
```


my_Service.yaml

```
GNU nano 6.2
apiVersion: v1
kind: Service
metadata:
  labels:
    app: ip312s07-app
    owner: ip312s07
  name: ip312s07-service
  namespace: ip312
spec:
  ports:
    - name: http
      port: 7777
      protocol: TCP
      targetPort: 5000
  selector:
    app: ip312s07-app
  type: ClusterIP
```

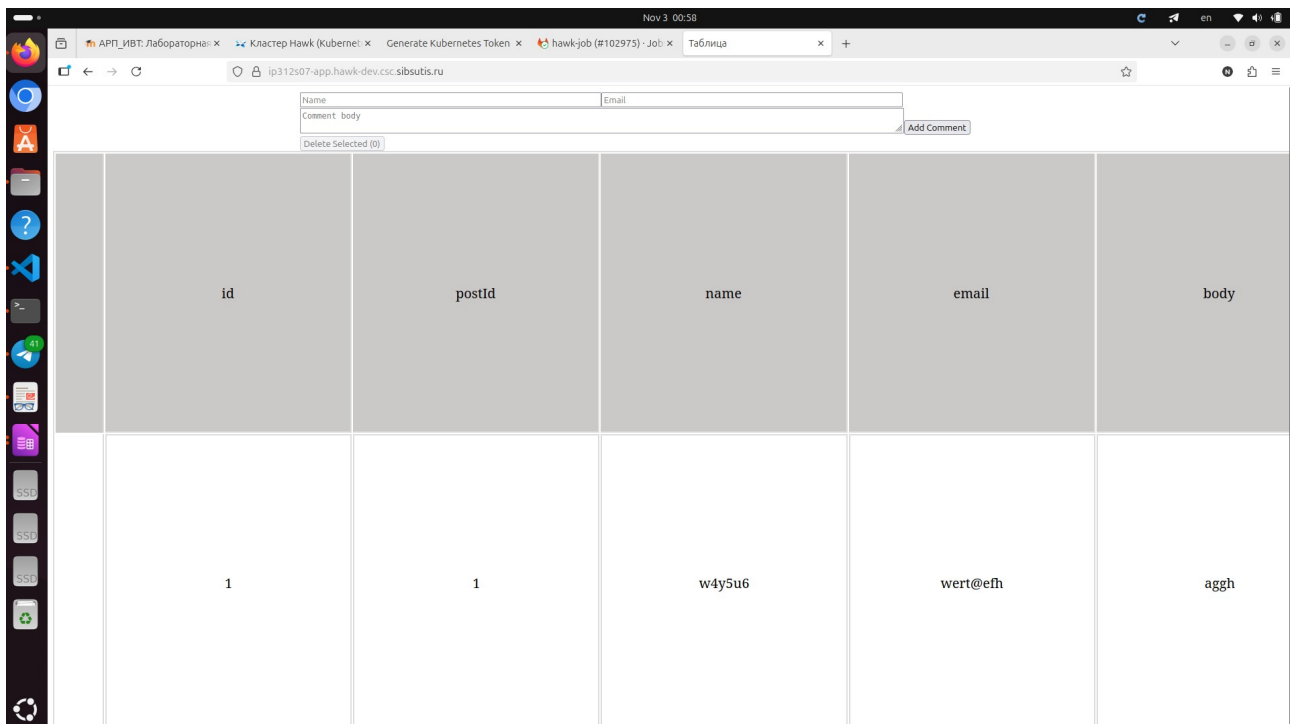
my_Ingress.yaml

```
GNU nano 6.2
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: ip312s07-ingress
  namespace: ip312
  labels:
    owner: ip312s07
spec:
  ingressClassName: traefik-dev
  rules:
  - host: ip312s07-app.hawk-dev.csc.sibsutis.ru
    http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: ip312s07-service
            port:
              number: 7777
```

5) Демонстрация работы приложения

```
ip312s07@hawk-access:~$ kubectl --kubeconfig hawk_k8s.yaml apply -f my_StatefulSet.yaml
statefulset.apps/ip312s07-app created
ip312s07@hawk-access:~$ kubectl --kubeconfig hawk_k8s.yaml apply -f my_Service.yaml
service/ip312s07-service unchanged
ip312s07@hawk-access:~$ kubectl --kubeconfig hawk_k8s.yaml apply -f my_Ingress.yaml
ingress.networking.k8s.io/ip312s07-ingress unchanged
```

```
ip312s07@hawk-access:~$ kubectl --kubeconfig hawk_k8s.yaml -n ip312 get pods
NAME                                READY   STATUS    RESTARTS   AGE
ip312s07-app-0                     0/1     Running   0           39s
ip312s29-app-6b77759745-gh67z     0/1     ImagePullBackOff  0           2d5h
ip312s29-app-8644c796f4-q8gsj     0/1     ImagePullBackOff  0           2d5h
ip312s07@hawk-access:~$ kubectl --kubeconfig hawk_k8s.yaml -n ip312 get pods
NAME                                READY   STATUS    RESTARTS   AGE
ip312s07-app-0                     1/1     Running   0           2m40s
ip312s29-app-6b77759745-gh67z     0/1     ImagePullBackOff  0           2d5h
ip312s29-app-8644c796f4-q8gsj     0/1     ImagePullBackOff  0           2d5h
```



id	postId	name	email	body
1	1	w4y5u6	wert@efh	aggh