

Ссылки

Папка с презентациями:

<http://e42.link/k8s2603>

GitLab: <http://gitlab.ci-cd.workshop.express42.io>

Обратная связь по первому дню

<http://e42.link/ci-cd1>

Непрерывная поставка

Непрерывная поставка

Это возможность быстро и безопасно отдать любое изменение пользователям.



Принципы построения Continuous Delivery

- Проверять качество сборки
- Компьютер выполняет повторяющиеся задачи; люди решают проблемы
- Работать маленькими кусочками
- Постоянно улучшать
- Ответственность на всех

Минимальный набор, необходимый для CD

- Непрерывная интеграция
- Инфраструктура как код
- Политики деплоя

Непрерывная поставка с K8S

Настраиваем Gitlab для поставки в kubernetes

Практика 1_cd

Авторизация

Проще всего авторизоваться через уже существующий конфиг.

Откройте в Gitlab настройки вашей группы. Добавьте ENV-переменную

`KUBE_CONFIG` с выводом команды: `cat ~/.kube/config | base64`

user01-reddit > CI / CD Settings

Environment variables ?

Environment variables are applied to or tags. You can use environment var made available to the running applica

- REGISTRY_PASSWORD
- REGISTRY_URL
- REGISTRY_USER
- KUBE_CONFIG

Input variable key

[Save variables](#) [Reveal values](#)

Дополняем gitlab-ci.yml для UI

Добавляем стадию `deploy`. В ней мы установим kubectl и авторизуемся в kubernetes.

```
stages:
  ...
  - deploy

deploy:
  stage: deploy
  script:
    - apk add -U openssl curl tar gzip bash ca-certificates git
    - curl -LO https://storage.googleapis.com/kubernetes-release/release/$(curl -s
https://storage.googleapis.com/kubernetes-
release/release/stable.txt)/bin/linux/amd64/kubectl
    - chmod +x ./kubectl && mv ./kubectl /usr/local/bin/kubectl
    - mkdir -p ~/.kube/ && echo $KUBE_CONFIG | base64 -d > ~/.kube/config
    - kubectl cluster-info
    - kubectl apply -f kubernetes/
```

Проверяем работу

Применяем изменения:

```
git commit -am 'add release and deploy' && git push -u origin master
```

В последней джобе пайплайна мы должны увидеть информацию о кластере, с которым работаем и падение команды `kubectl apply -f kubernetes/`.

```
$ kubectl cluster-info
Kubernetes master is running at https://35.195.168.5
GLBCDefaultBackend is running at https://35.195.168.5/api/v1/namespaces/kube-system/services/default-http-backend:http/proxy
Heapster is running at https://35.195.168.5/api/v1/namespaces/kube-system/services/heapster/proxy
KubeDNS is running at https://35.195.168.5/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy
Metrics-server is running at https://35.195.168.5/api/v1/namespaces/kube-system/services/https:metrics-server:/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
$ kubectl apply -f kubernetes/
error: the path "kubernetes/" does not exist
ERROR: Job failed: exit code 1
```

Повторить

Добавьте стадию `deploy` для `post` и `comment`.

Не забудьте в имени docker контейнера подставить название нужного сервиса

```
stages:
  ...
  - deploy

deploy:
  stage: deploy
  script:
    - apk add -U openssl curl tar gzip bash ca-certificates git
    - curl -LO https://storage.googleapis.com/kubernetes-release/release/$(curl -s
https://storage.googleapis.com/kubernetes-
release/release/stable.txt)/bin/linux/amd64/kubectl
    - chmod +x ./kubectl && mv ./kubectl /usr/local/bin/kubectl
    - mkdir -p ~/.kube/ && echo $KUBE_CONFIG | base64 -d > ~/.kube/config
    - kubectl cluster-info
    - kubectl apply -f kubernetes/
```

Описываем k8s- ресурсы для приложений

Практика cd_2

Пример deployment

Это пример ресурса deployment, который можно взять за основу.

```
apiVersion: apps/v1
kind: Deployment
metadata:      # Описание Deployment, название и лейблы.
  name: nginx
  labels:
    app: nginx
spec:
  replicas: 1
  selector:    # Лейблы pod`ов, за которые отвечает deployment
    matchLabels:
      app: nginx
  template:
    metadata:
      name: nginx
      labels:    # Лейблы, с которыми нужно создавать pod
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx:1.15.8
```

Пример Service

```
kind: Service
apiVersion: v1
metadata:
  name: nginx
spec:
  type: ClusterIP
  selector:
    app: nginx
  ports:
  - protocol: TCP
    port: 80
    targetPort: 80
```

Пример Ingress

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: nginx
  labels:
    app: nginx
spec:
  rules:
  - http:
    paths:
    - path: /*          # Какие запросы перенаправлять
      backend:
        serviceName: nginx # На какой `service` перенаправлять
        servicePort: 80   # На какой порт сервиса перенаправлять
```

Полезные команды

Если не заработало это может помочь вам локализовать проблему

```
kubectl get pods/svc # Посмотреть поды/сервисы и краткую информацию о них:
```

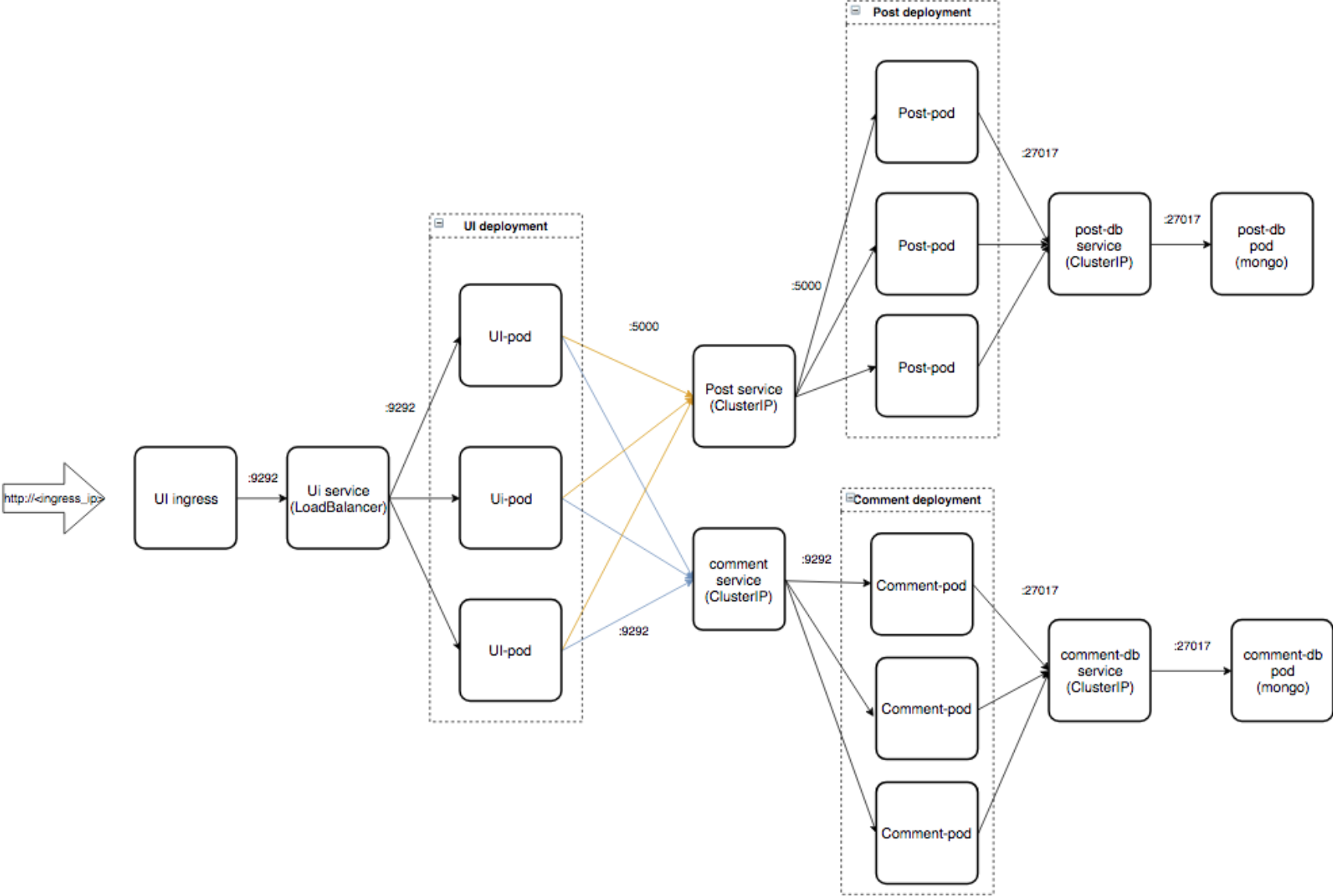
```
kubectl describe pod/svc <pod/service name> # Более подробная информация про конкретный под или сервис
```

Полезные команды

Еще все сервисы содержат healthcheck, их значения можно получить:

```
kubectl exec -it <ui_pod_name> curl http://localhost:9292/healthcheck  
kubectl exec -it <post_pod_name> curl http://localhost:5000/healthcheck  
kubectl exec -it <comment_pod_name> curl http://localhost:9292/healthcheck
```

Чего мы хотим?



deployment для ui

Создаем папку `~/project/ui/kubernetes`. В ней будем описывать манифесты для запуска ui-приложения в k8s.

Опишите `deployment.yaml` для `ui`.

- Название `ui`
- Должны быть указаны лейблы `app: reddit` и `component: ui`
- Количество подов: 3
- Образ: `gcr.io/cd-k8s-236617/user-<your number>-ui:v0.0.1`

deployment для ui

Указываем api, kind, имя и ставим метки:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: ui
  labels:
    app: reddit
    component: ui
...
```

Указываем количество реплик и какие поды запускать:

```
...
spec:
  replicas: 3
  selector:
    matchLabels:
      app: reddit
      component: ui
...
```

deployment для ui

Описываем запускаемый контейнер

```
...
template:
  metadata:
    name: ui
  labels:
    app: reddit
    component: ui
  spec:
    containers:
    - name: ui
      image: gcr.io/cd-k8s-236617/user-<your number>-ui:v0.0.1
```

Примените изменения

```
kubectl apply -f kubernetes/
```

Убедитесь, что приложение запустилось командой `kubectl get pods`

Команда для проверки, что приложение работает:

```
kubectl exec -it <имя пода> curl http://localhost:9292
```

Пример вывода:

```
<!DOCTYPE html>
<html lang='en'>
<head>
<meta charset='utf-8'>
<meta content='IE=Edge,chrome=1' http-equiv='X-UA-Compatible'>
<meta content='width=device-width, initial-scale=1.0' name='viewport'>
...
```

service для ui

В `~/project/ui/kubernetes/service.yaml` описываем service для ui.

- Название `ui`
- лейблы `app: reddit`, `component: ui`
- port: `9292`, targetPort: `9292`
- тип: `NodePort`

service для ui

Указываем api, kind, имя и ставим метки:

```
apiVersion: v1
kind: Service
metadata:
  name: ui
  labels:
    app: reddit
    component: ui
...
```

Укажем в качестве типа NodePort и что данный сервис применим к app: reddit

```
...
spec:
  type: NodePort
  selector:
    app: reddit
    component: ui
...
```

service для ui

И пропишем необходимые порты

```
...  
ports:  
- protocol: TCP  
  port: 9292  
  targetPort: 9292
```

Примените изменения

```
kubectl apply -f kubernetes/service.yaml
```

Проверяем, что сервис работает.

```
kubectl get svc ui # убеждаемся что ресурс создан  
  
# выбираем под, через который проверим работу сервиса  
kubectl get pods  
  
# делаем запрос через сервис, на выходе должен быть html  
kubectl exec -it <pod name> curl http://ui:9292
```

ingress для ui

В качестве ingress`а будем использовать nginx-ingress

Установим его с помощью **Helm** (о нем подробнее расскажем в следующей лекции 😊)

В **Helm 3** отсутствуют предустановленные репозитории helm чартов, поэтому добавим **stable** репозиторий:

```
$ helm repo add stable https://charts.helm.sh/stable
$ helm repo update
"stable" has been added to your repositories
```

Теперь можем выполнить поиск уже существующего чарта **nginx-ingress**:

```
$ helm search repo nginx
NAME                CHART VERSION  APP VERSION  DESCRIPTION
stable/nginx-ingress  1.33.0         0.30.0      An nginx Ingress controller that uses
ConfigMap...
```

ingress для ui

```
kubectl create ns nginx-ingress #создадим namespace для nginx-ingress
```

```
helm upgrade --install nginx-ingress stable/nginx-ingress --version=1.41.3 --  
namespace=nginx-ingress
```

Проверка что **helm chart** установлен:

```
$ helm ls -n nginx-ingress
```

NAME	NAMESPACE	REVISION	UPDATED	STATUS	CHART
nginx-ingress	nginx-ingress	1	2020-03-02 ...	deployed	nginx-ingress-1.41.3

ingress для ui

В `~/project/ui/kubernetes/ingress.yaml` описываем ingress для ui.

- Название: `ui`
- Лейблы: `app: reddit` и `component: ui`
- Имя бекенда: `ui`
- Порт на бекенде: `9292`

ingress для ui

Указываем api, kind, имя и ставим метки:

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: ui
  labels:
    app: reddit
    component: ui
  annotations:
    kubernetes.io/ingress.class: nginx
...
```

ingress для ui

И прописываем правила. Доступ по http, по всем маршрутам, backend port 9292

```
spec:
  rules:
  - host: ui.< EXTERNAL-IP >.nip.io
    http:
      paths:
      - backend:
          serviceName: ui
          servicePort: 9292
```

EXTERNAL-IP берем отсюда:

```
$ kubectl get svc -n nginx-ingress
NAME                                TYPE           CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE
nginx-ingress-controller           LoadBalancer  10.0.104.13  104.155.22.61  ...        18h
...
```

Примените изменения

```
kubectl apply -f kubernetes/ingress.yaml
```

Убедитесь, что `ingress` создан командой `kubectl get ingress ui`. Ingress долго настраивается, дождитесь, пока в поле address не появится ip. Откройте `ui.EXTERNAL_IP.nip.io` в браузере, должна быть видна страница.

Microservices Reddit in ui-857448d9d8-mcn4t container

Can't show blog posts, some problems with the post service. Refresh?

Menu

[All posts](#)

[New post](#)

Post

Опишите `deployment` и `service` для `post`

- описывайте конфигурацию в `~/project/post/kubernetes`
- лейблы `app: reddit`, `component: post`
- используйте образ `gcr.io/cd-k8s-236617/user-<your number>-post:v0.0.1`
- тип сервиса: `ClusterIP`
- `post` слушает `5000` порт, это необходимо отразить в `service`
- количество подов `3`

Различия с ui

Манифесты для `ui` и `post` выглядят почти одинаково. Различаются они следующими параметрами:

- В labels component сменить с `ui` на `post`
- В deployment в image заменить образ на `gcr.io/cd-k8s-236617/user-
<your number>-post:v0.0.1`
- В service тип сменить с `NodePort` на `ClusterIP`
- В service заменить порт `9292` на `5000`

Post ingress

- Нужен ли ingress для post и comment?
 - Если нужен, то зачем и каким он должен быть?
 - Если не нужен, то почему?
- Почему тип сервиса для post - ClusterIP, а не NodePort или LoadBalancer?

База данных для post

Создайте `deployment` и `service` базы данных для post:

- deployment опишите в `~/project/post/kubernetes/db-deployment.yaml`
- service опишите в `~/project/post/kubernetes/db-service.yaml`
- название deployment и service: `post-db`
- лейблы `app: reddit`, `component: post-db`
- используйте образ `mongo:3.2`
- количество подов `1`
- mongodb слушает порт `27017`
- тип сервиса `ClusterIP`

Так выглядит успех

Если все получилось и ресурсы для post были описаны корректно, то с главной страницы должно пропасть сообщение с ошибкой.

Microservices Reddit in ui-8cb65dbff-l5hbx container

Menu

[All posts](#)

[New post](#)

Comment

Опишите `deployment` и `service` для `comment`

- сохраните конфигурацию в `~/project/comment/kubernetes`
- лейблы `app: reddit`, `component: comment`
- используйте образ `gcr.io/cd-k8s-236617/user-<your number>-comment:v0.0.1`
- Comment слушает `9292` порт, это необходимо отразить в `service`
- задайте верный тип для `service`

Comment-db

Создайте `deployment` и `service` базы данных для comment:

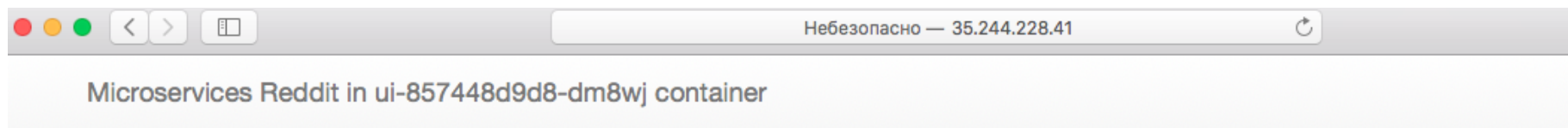
- deployment опишите в `~/project/comment/kubernetes/db-deployment.yaml`
- service опишите в `~/project/comment/kubernetes/db-service.yaml`
- название deployment и service: `comment-db`
- лейблы `app: reddit`, `component: comment-db`
- используйте образ `mongo:3.2`
- количество подов `1`
- mongodb слушает порт `27017`
- тип сервиса `ClusterIP`

Проверяем:

```
kubectl get ingress
```

NAME	HOSTS	ADDRESS	PORTS	AGE
ui	ui.< EXTERNAL-IP >.nip.io	35.244.228.41	80	3h

```
http://ui.< EXTERNAL-IP >.nip.io
```



Menu

[All posts](#)

[New post](#)

Пост и комментарий должны создаваться

Microservices Reddit in ui-8cb65dbff-l5hbx container



0



Title!

03-05-2019
18:48

Menu

[All posts](#)

[New post](#)

[Go to the link](#)

Post my comment

Запустим изменения

Запустите изменения внесенные в репозитории `ui`, `post` и `comment`

В репозитории `ui`:

```
git add kubernetes/  
git commit -m "ui kubernetes manifests" && git push
```

 Аналогично для репозиториев `post` и `comment`