ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ
УНИВЕРСИТЕТ им. А. И. ГЕРЦЕНА»

Выпускная квалификационная работа

Автоматизация процесса развёртывания приложений в кластере Kubernetes на основе применения методик CI/CD

Работу выполнил:
Родионов Родион Николаевич
Руководитель выпускной квалификационной работы:
Старший преподаватель Аксютин Павел Александрович

Санкт-Петербург 2025

Актуальность исследования связана с возрастающей потребностью в автоматизации развертывания приложений в Kubernetes через CI/CD, что снижает риски человеческого фактора и обеспечивает воспроизводимость процессов за счёт подхода "Infrastructure as Code".

Цель данной выпускной квалификационной работы состоит в создании автоматизированной системы тестирования, сборки и развертывания на основе Kubernetes и методологий CI/CD

Для реализации цели были сформированы следующие **задачи**:
1. Описать методологию DevOps и практики CI/CD.
2. Выполнить обзор платформы Kubernetes.
3. Показать основные принципы развертывания приложений в Kubernetes.
4. Спроектировать и настроить виртуальный сервер и инфраструктуру.
5. Реализовать способ развертывания приложения в Kubernetes.
6. Разработать механизм автоматизации развертывания приложения и реализовать механизм на конкретном примере.
7. Протестировать корректность работы автоматизированного развертывания.

Основные методологии и инструменты, используемые в работе

Понятие DevOps методологии

DevOps — это методология в сфере IT, направленная на объединение процессов разработки и эксплуатации программного продукта или услуги.

CI (Continuous Integration) — непрерывная интеграция, при которой изменения кода автоматически интегрируются в один программный проект.

CD (Continuous Delivery) — непрерывная поставка — практика автоматизации процесса развертывания приложений.

GitOps — это современная методология управления инфраструктурой и приложениями, которая использует Git-репозиторий в качестве единого источника истины для всех конфигураций и изменений.

Обзор устройства платформы Kubernetes

Платформа Kubernetes представляет собой цифровую среду обеспечивающую оркестрирование контейнеризованными рабочими задачами и сервисами.

Под — минимальная неделимая единица развертывания, «обертка» для одного или нескольких контейнеров.

Деплоймент предоставляет декларативные обновления для Подов.

Особенности развертывания приложений в Kubernetes

Развертывание приложения в Kubernetes можно реализовать с помощью инструмента командной строки kubectl и YAML или JSON файлами, содержащими декларативную конфигурацию приложения — манифестами.

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-nginx
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.14.2
        ports:
        - containerPort: 80
```

Пример YAML манифеста ресурса deployment

Helm — менеджер пакетов для Kubernetes. Подобно тому, как пакетный менеджер операционной системы упрощает установку программного обеспечения, Helm упрощает развертывание приложений и ресурсов в кластерах Kubernetes, позволяя реализовать эту задачу одной командой.

```
$ helm install happy-panda bitnami/wordpress
```

Пример команды Helm

Helmfile — это декларативный инструмент для развертывания Helm-чартов. Он расширяет возможности Helm, предоставляя продвинутые функции оркестрации через YAML-конфигурационный файл (как правило helmfile.yaml)

```yaml
repositories:
 - name: prometheus-community
   url: https://prometheus-community.github.io/helm-charts

releases:
- name: prom-norbac-ubuntu
  namespace: prometheus
  chart: prometheus-community/prometheus
  set:
  - name: rbac.create
    value: false
```

Пример файла helmfile.yaml

Проектирование и настройка виртуального сервера и инфраструктуры

В связи с ограниченными ресурсами было принято решение развернуть K3S кластер (K3S — это дистрибутив Kubernetes) на один узел. Для хранения данных использовалась СУБД PostgreSQL, для управления базами данных — DB Operator. Чтобы декларативно прописать развертывание инфраструктуры использовался Helmfile, для Vaultwarden был сделан Helm chart и настроена автоматическая сборка контейнера с помощью Github Actions, а для развертывания настроен Argo CD.

На первом этапе реализации процесса автоматизации была создана и настроена виртуальная машина (ВМ). В качестве операционной системы принято решение использовать Alt-p11-jeos.

Сама виртуальная машина была развернута в Proxmox Virtual Environment — системе виртуализации с открытым исходным кодом.

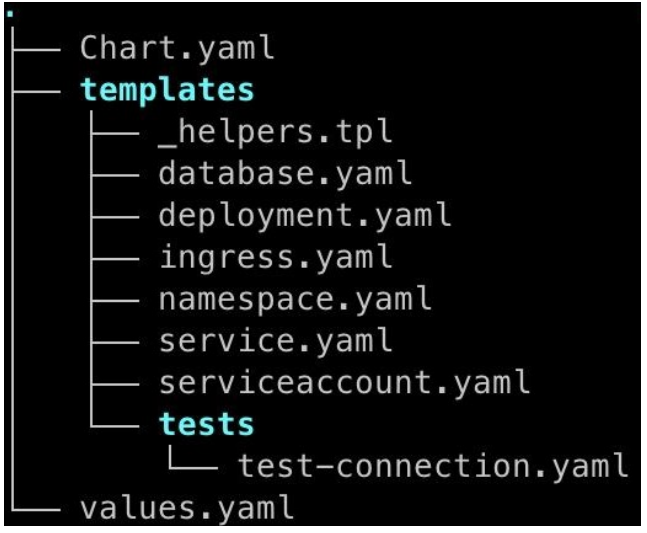
Структура файлов в директории Helmfile

Для большей стабильности системы конфигурации приложений были прописаны декларативно с использованием Helmfile. В файле helmfile.yaml находится перечень релизов. В файлах values.yaml прописана конфигурация приложений. В файлах secrets.yaml зашифрована чувствительная часть конфигурации.

Создание Helm чарта для Vaultwarden

Helm чарт (Chart) — это формат упаковки приложений для Kubernetes, который содержит все необходимые ресурсы и настройки для их развертывания.

Helm использует шаблоны — файлы написанные на языке программирования Go. Шаблоны позволяют динамически создавать ресурсы Kubernetes с помощью подстановки значений из файла values.yaml или аргументов командной строки.



Структура чарта

```
env:
{{- range $key, $value := .Values.env }}
  - name: {{ $key  }}
    value: {{ quote $value }}
{{- end }}
```

Отрывок из шаблона deployment.yaml

```
env:
  DOMAIN: https://example.net
  SIGNUPS_ALLOWED: 'true'
  SIGNUPS_VERIFY: 'false'
  WEB_VAULT_ENABLED: 'true'
  ADMIN_TOKEN: qwerty123
```

Отрывок из файла values.yaml

```
- env:
  - name: ADMIN_TOKEN
    value: qwerty123
  - name: DOMAIN
    value: https://example.net
  - name: SIGNUPS_ALLOWED
    value: "true"
  - name: SIGNUPS_VERIFY
    value: "false"
  - name: WEB_VAULT_ENABLED
    value: "true"
  - name: DATABASE_URL
```

Отрывок манифеста

Проектирование конвейера CI/CD

Диаграмма деятельности

Для проектирования программы были сделаны UML диаграммы

UML, или Unified Modeling Language, — это унифицированный язык моделирования.

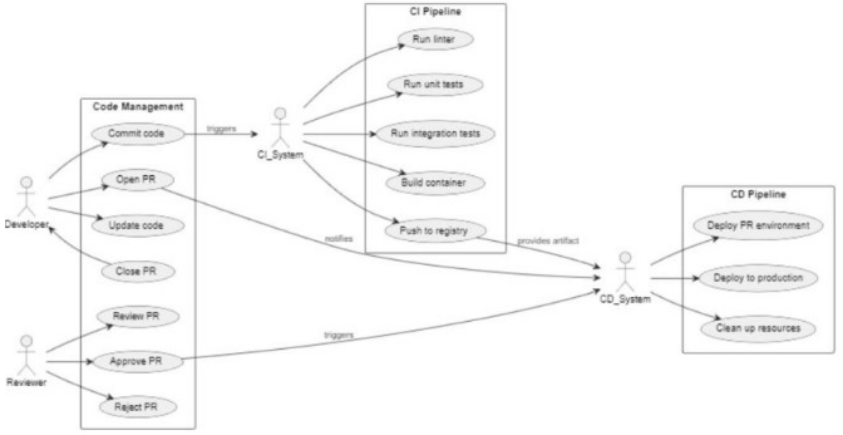


Диаграмма вариантов использования

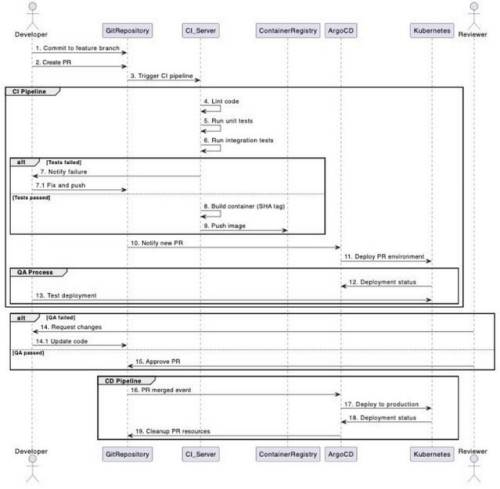


Диаграмма последовательности

Настройка конвейера CI/CD

GitHub Actions - это проприетарная платформа непрерывной интеграции и поставки, которая имеет встроенную систему плагинов Actions (от англ. действия), которые могут выполнять разные функции и быть написаны на разных языках программирования.

```yaml
name: Build and Push Docker Image
on:
  pull_request:
jobs:
  docker:
    runs-on: ubuntu-latest
    steps:
      -
        name: Checkout code
        uses: actions/checkout@v4
      -
        name: Set up QEMU
        uses: docker/setup-qemu-action@v3
      -
        name: Set up Docker Buildx
        uses: docker/setup-buildx-action@v3
      -
        name: Build and push
        uses: macbre/push-to-ghcr@master
        with:
          image_name: ${{ github.repository }}
          github_token: ${{ secrets.TOKEN }}
          context: .
          image_tag: ${{ github.sha }}
```

Файл с описанием работы CI

За начало работы CD конвейера в Argo CD отвечают генераторы. Представленный генератор работает при создании PR в указанном репозитории.

```yaml
generators:
- pullRequest:
    github:
      owner: jack-lull
      repo: vaultwarden
      tokenRef:
        secretName: github-token
        key: token
```

Генератор

ApplicationSets представляют собой способ декларативного описания динамически развертываемого приложения.

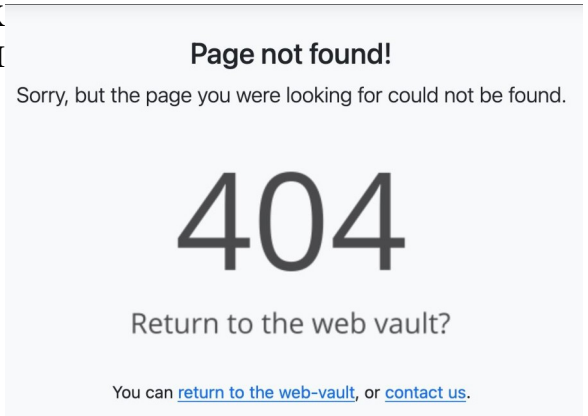
```yaml
template:
  metadata:
    name: 'vaultwarden-pr-{{ .number }}'
    labels:
      project: 'vaultwarden-pr'
    annotations: {}
  spec:
    project: 'default'
    source:
      repoURL: https://github.com/jack-lull/vaultwarden
      targetRevision: '{{ .head_sha }}'
      path: chart
    destination:
      server: https://kubernetes.default.svc
      namespace: 'app-prewiev-{{ .number }}'
    syncPolicy:
      automated:
        prune: false
        selfHeal: false
      syncOptions:
      - CreateNamespace=true
```

Шаблон ApplicationSet

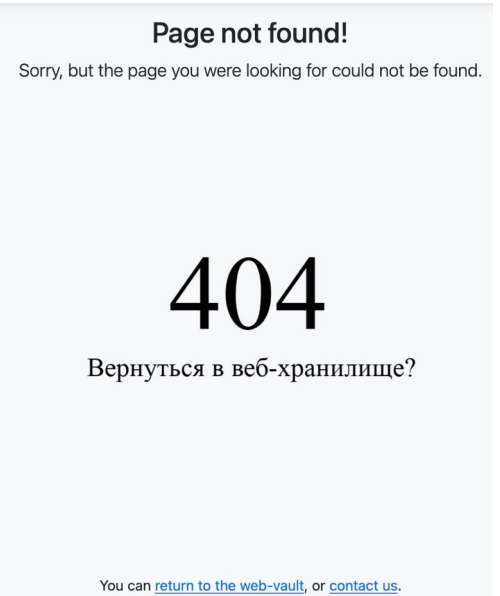
Тестирование работы конвейера CI/CD

Текущее тестирование проводилось с устранением возникающих ошибок на всех этапах работы по автоматизации развертывания приложения.

Для проведения финального тестирования конвейера, файл, содержащий изображение 404 «Return to the web vault?», был изменен на файл с новым изображением.



Страница до изменений



Страница после изменений

Выводы по работе

1. Дано понятие методологии DevOps и практики автоматизации CI/CD.

2. Рассмотрены принципы работы, основы и ресурсы платформы Kubernetes.

3. Представлены особенности развертывания приложений в Kubernetes, как путем нативных манифестов, так и с использованием инструмента шаблонизации Helm.

4. Спроектирован и настроен виртуальный сервер на базе операционной системы Alt-p11-jeos. В кластере K3S организована необходимая для автоматического развертывания приложения инфраструктура: СУБД PostgreSQL, DB Operator, Argo CD.

5. Реализован Helm чарт и все необходимые для приложения шаблоны манифестов.

6. Спроектирован механизм работы конвейера CI/CD с использованием унифицированного языка моделирования UML.

7. Реализована автоматическая сборка и отправка образа контейнера с приложением с использованием платформы GitHub Actions.

8. Настроена конфигурация декларативного инструмента непрерывной доставки Argo CD для динамического развертывания тестовой и производственной версии приложения в кластер.

9. Корректность работы конвейера протестирована путем обновления приложения.

В результате успешно достигнута поставленная цель - создание автоматизированной системы тестирования, сборки и развертывания на основе Kubernetes и методологий CI/CD.

Спасибо за внимание!