

Платформа для создания архитектурного репозитория ИТ-решений ArchTech

Инструкция администратора

Москва 2026

Содержание

1. Общие положения 3
2. Контакты технического специалиста 3
3. Инсталляция программного обеспечения 3
4. Начало работы 4
5. Установка и настройка Docker и Docker Compose 5
6. Установка и настройка GitLab 5
7. Установка и настройка KeyCloak 7
8. Установка PlantUML 9
9. Установка стартовой страницы ArchTech 10
10. Установка и настройка Портала Архитектурного репозитория ArchTech 11
11. Проверка корректности создания всех контейнеров 14

1. Общие положения

Платформа для создания архитектурного репозитория ИТ-решений ArchTech (далее – **ArchTech**) — это инновационный инструмент для управления цифровой архитектурой, разработанный с учётом современных требований к архитектурной зрелости и стандартизации в крупных организациях. Продукт предназначен для описания, анализа и контроля архитектуры информационных систем, ИТ-сервисов и программных продуктов, а также для повышения качества принимаемых архитектурных решений.

2. Контакты технического специалиста

Технический специалист (кому можно задавать вопросы по программному обеспечению):

1. ФИО: Синчелеев Денис Максимович
2. Телефон: 8-961-258-55-58
3. Email: info@arch-guru.ru

3. Инсталляция программного обеспечения

ArchTech реализована как SPA-приложение (Single Page Application). Для стандартного развертывания используется Docker и Docker Compose. Продукт поддерживает работу в корпоративном режиме, включая ролевую модель и развертывание в Kubernetes.

Front-end: VueJS

Back-end: NodeJS, GitLab, PlantUML, KeyCloak

ОС: Linux (поддерживаются любые дистрибутивы)

На данный момент ArchTech развёрнута на одном выделенном сервере в форме 7 контейнеров Docker:

1. gitlab на 81 порту;
2. keycloak на 8433 порту,
3. plantuml на 8079 порту,
4. startpage на 80 порту,
5. archtech на 8080 порту
6. postgresql на 5432 порту
7. reverse-proxy на 7443 и 8433-портах

Характеристики сервера:

1. Операционная система: Ubuntu 24.04 LTS
2. Тип сервера: Виртуальный
3. Площадка размещения: Облачная инфраструктура Reg.ru
4. Процессор: Xeon E-2176G 3.70 ГГц, 6 ядер / 12 Ht
5. RAM: 32 ГБ DDR4

6. Диски: 2 x 240 ГБ SSD

4. Начало работы

Вся система разворачивается в стандартной конфигурации на одном сервере (на данный момент в облаке reg.ru). Для подключения к этому серверу по SSH нужно:

- Запустить cmd.exe в Windows
- Перейти в директорию с OpenSSH: cd C:\Windows\System32\OpenSSH
- Подключиться к серверу: ssh root@89.108.124.153
- Ввести пароль: Для получения пароля необходимо направить запрос на ящик info@arch-guru.ru

После этого будет доступна консоль сервера с правами root.

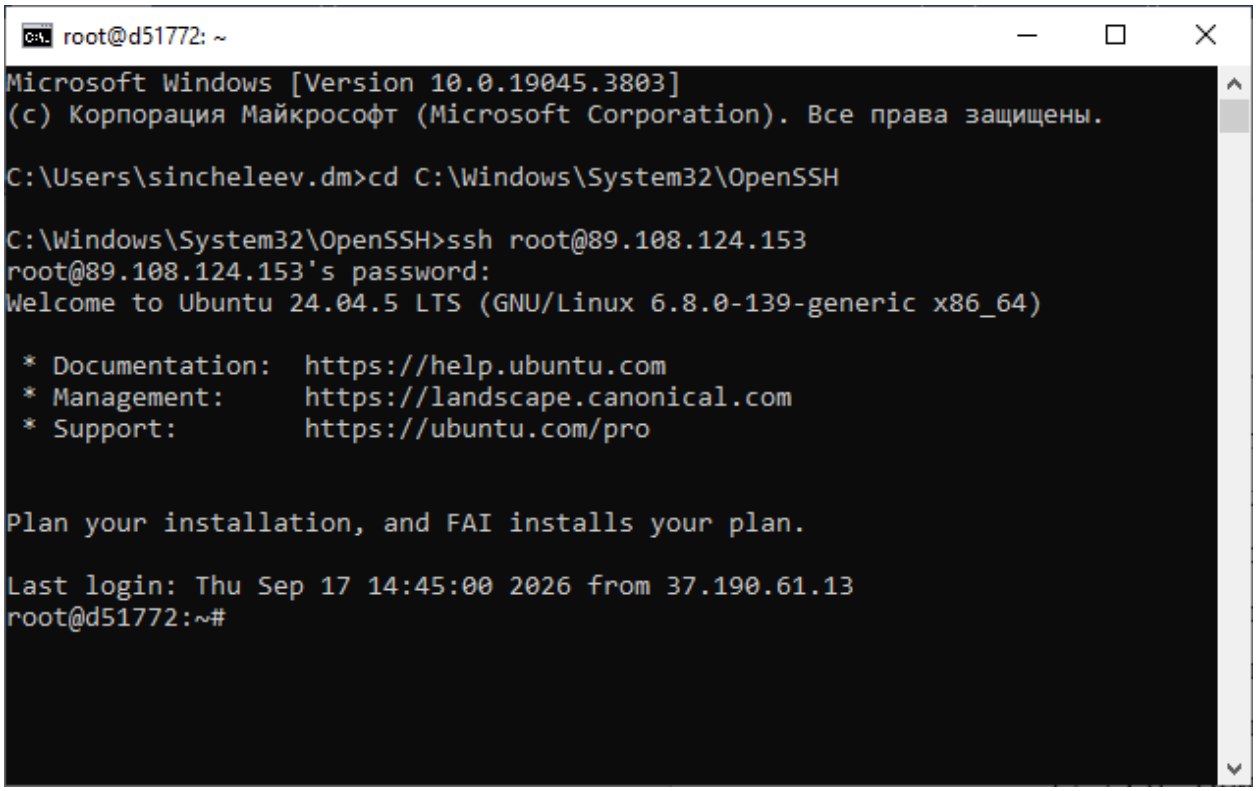


Рисунок 1. Подключение к серверу по SSH

Для возможности копирования данных на сервер необходимо:

- Скачать и установить бесплатный FTP-клиент для Linux WinSCP: https://winscp.net/eng/download.php
- Зайти в WinSCP и добавить новое подключение
  - Имя хоста: 89.108.124.153
  - Порт: 22
  - Имя пользователя: root
  - Пароль: Для получения пароля необходимо направить запрос на ящик info@arch-guru.ru
- Нажать Войти

- Сохранить ярлык для подключения на рабочем столе
- После этого можно копировать, удалять и редактировать любые данные на сервере по FTP

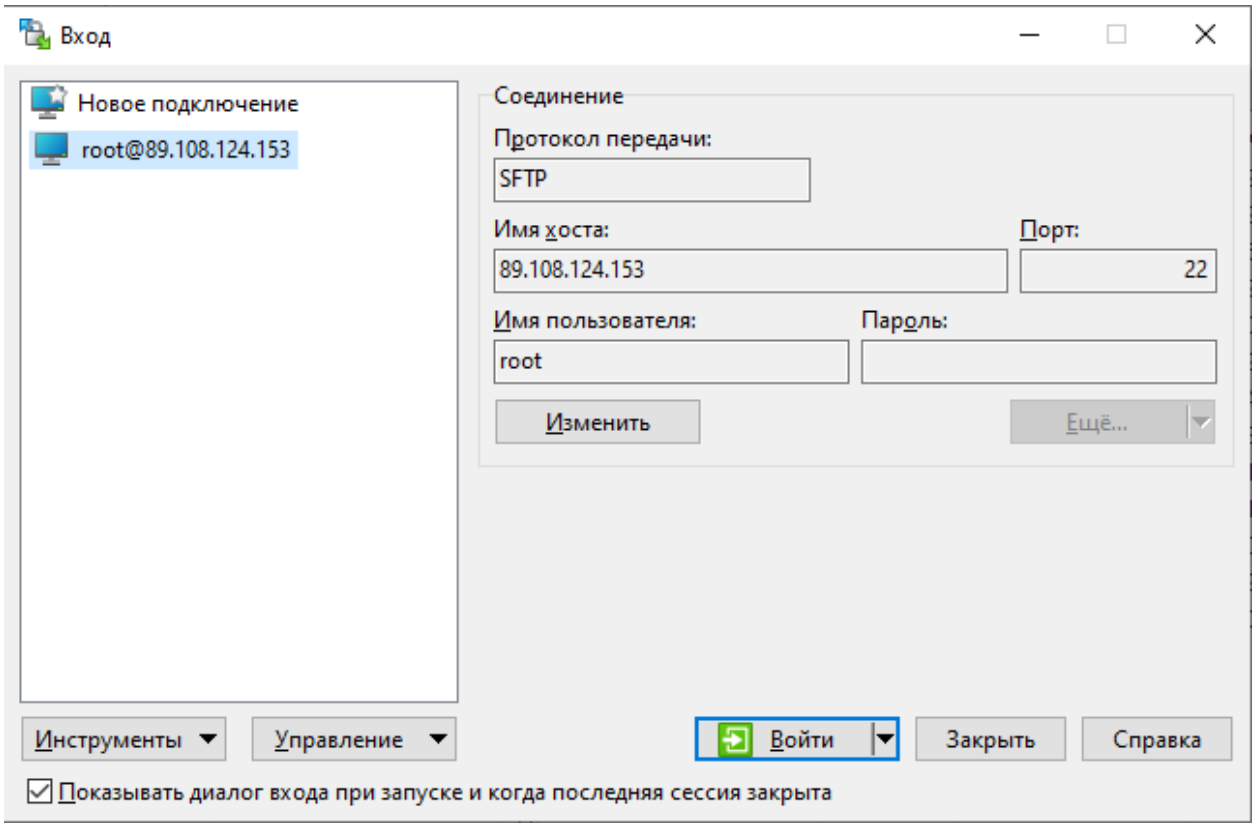


Рисунок 2. Создание нового подключения в WinSCP

5. Установка и настройка Docker и Docker Compose

Для установки Docker и Docker Compose необходимо ввести следующие команды в консоли:

- Обновить индексы пакетов apt: sudo apt update
- Установить Docker: sudo apt install docker-ce –y
- Установить Docker-compose: sudo apt-get install docker-compose
- Проверить установку: docker –version
- Проверить установку: docker-compose --version

6. Установка и настройка GitLab

Для установки GitLab необходимо ввести следующие команды в консоли:

- Перейти в каталог GitLab: cd /home/gitlab
- Запустить docker-compose.yaml: docker-compose up

После установки необходимо создать новый проект в GitLab. Для этого нужно:

- Зайти в административный интерфейс GitLab: http://89.108.124.153:81
- Ввести учётные данные администратора:
  - Логин: root
  - Пароль: Для получения пароля необходимо направить запрос на ящик info@arch-guru.ru

– После установки GitLab пароль от УЗ root автоматически сохраняется хранится в файле /home/gitlab/config/ initial_root_password

– Создать новый проект в разделе меню Projects/New Project/Create new project/ Create blank project http://89.108.124.153:81/projects/new#blank_project

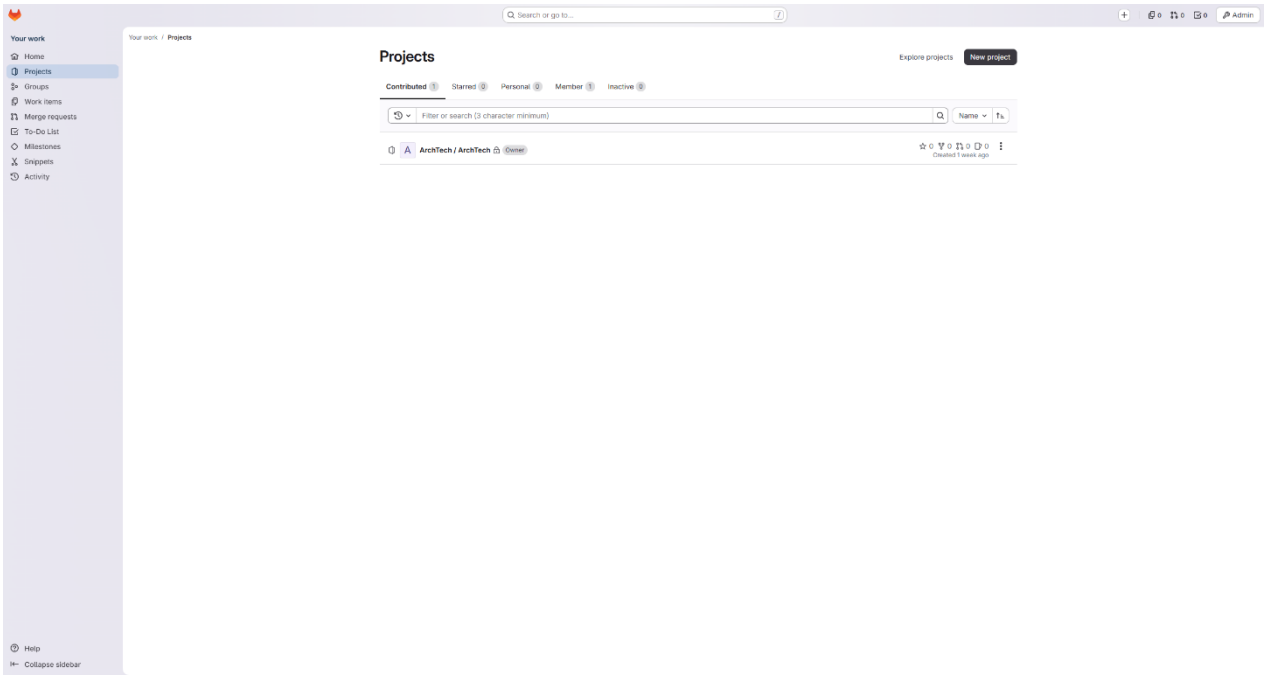


Рисунок 3. Создание нового проекта в GitLab (шаг 1)

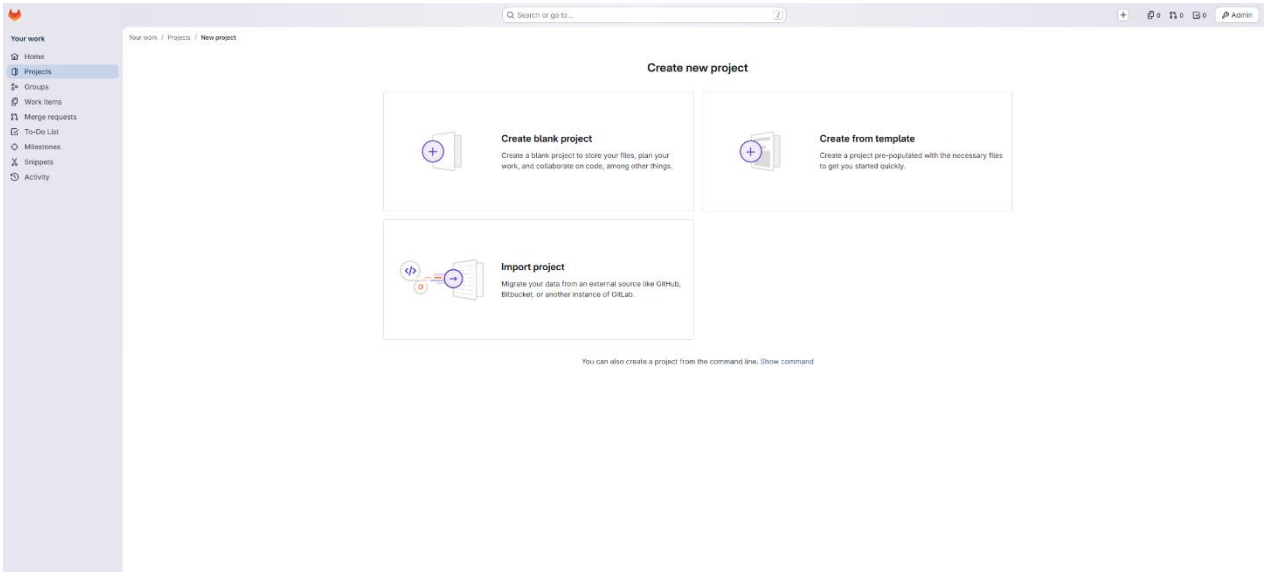


Рисунок 4. Создание нового проекта в GitLab (шаг 2)

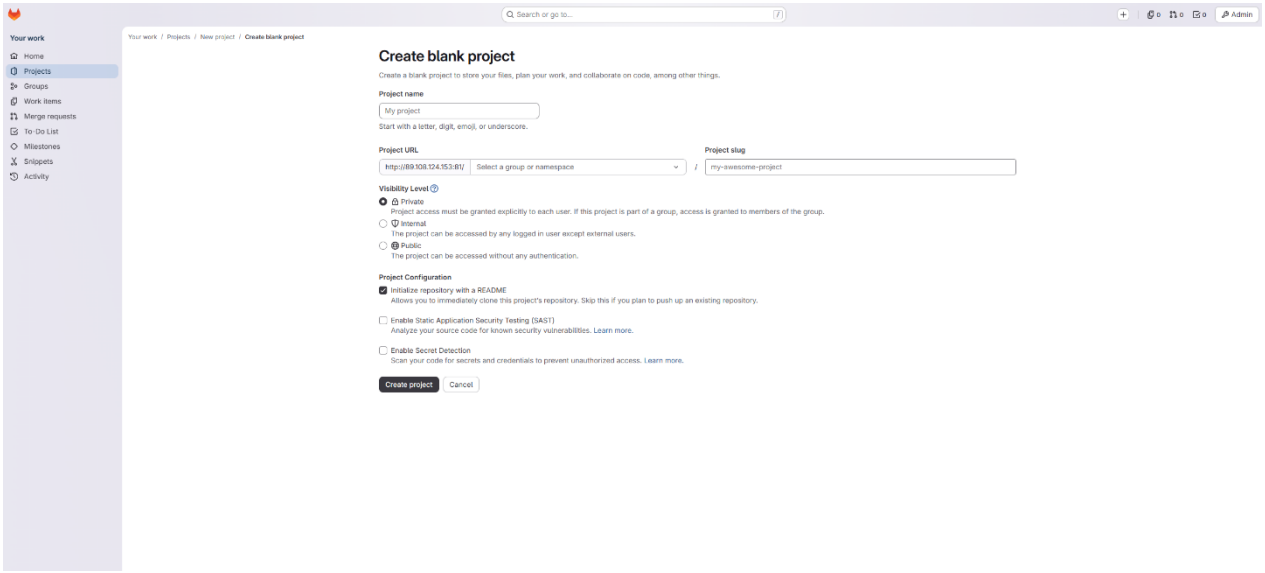


Рисунок 5. Создание нового проекта в GitLab (шаг 3)

Содержимое файла docker-compose.yaml в директории /home/gitlab:

```
version: "3.8"
services:
  gitlab:
    image: 'gitlab/gitlab-ce:latest'
    restart: always
    container_name: gitlab
    user: root
    hostname: 'localhost'  # Укажите ваше доменное имя
    shm_size: 1gb
    environment:
      GITLAB_OMNIBUS_CONFIG: |
        external_url 'https://test.arch-guru.ru:81'  # Укажите ваш URL
        nginx['listen_port'] = 81
        nginx['listen_https'] = false
        gitlab_rails['gitlab_shell_ssh_port'] = 2221
    ports:
      - '81:81'
      - '2221:21'
    volumes:
      - '/home/gitlab/config/:/etc/gitlab'
      - '/home/gitlab/logs/:/var/log/gitlab'
      - '/home/gitlab/data/:/var/opt/gitlab'
```

7. Установка и настройка KeyCloak

Для установки KeyCloak необходимо ввести следующие команды в консоли:

- Перейти в каталог KeyCloak: cd /home/keycloak

- Запустить docker-compose.yaml: docker-compose up

Сразу после установки необходимо отключить SSL в контейнере KeyCloak. Для этого нужно:

- Зайти в контейнер keycloak: docker exec -it keycloak bash
- Зайти в конфигурационный файл keycloak: /opt/keycloak/bin/kcadm.sh config credentials --server http://localhost:8080 --realm master --user admin --password admin
- Отключить SSL в конфигурационном файле keycloak: /opt/keycloak/bin/kcadm.sh update realms/master -s sslRequired=NONE

Далее необходимо создать новый realm в KeyCloak. Для этого нужно:

- Зайти в административный интерфейс KeyCloak: https://test.arch-guru.ru:8443
- Ввести учётные данные администратора:
  - Логин: admin
  - Пароль: Для получения пароля необходимо направить запрос на ящик info@arch-guru.ru
- Создать новый realm с названием archtech в разделе меню Manage realms/Create realm
- При импорте использовать готовый конфигурационный файл: /home/keycloak/realm-export.json

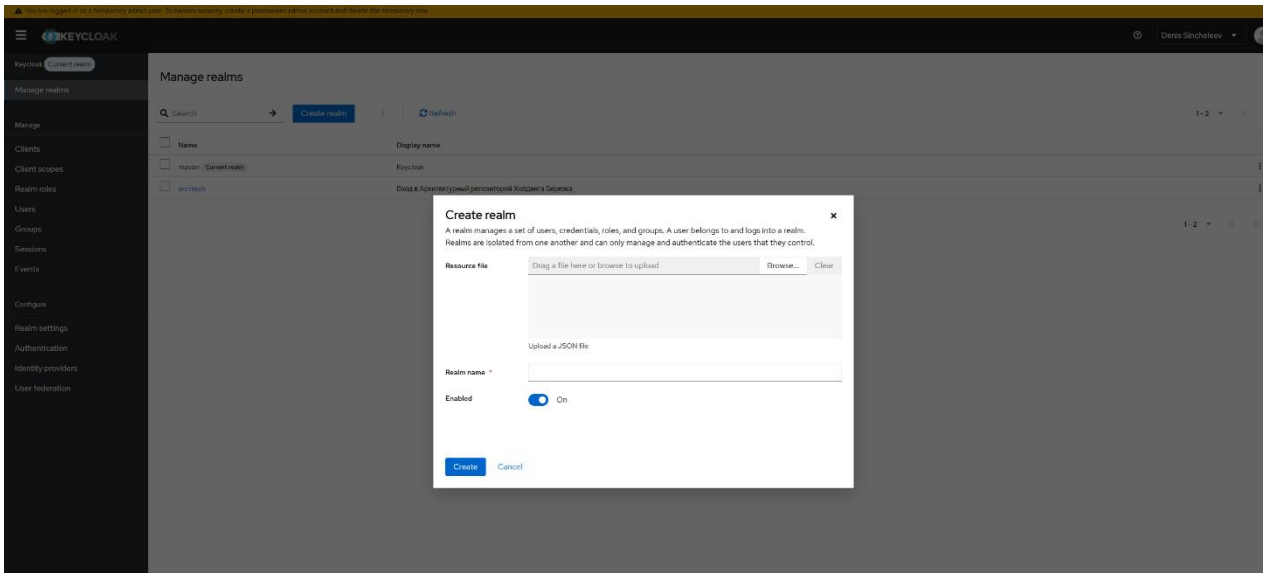


Рисунок 6. Создание нового realm в KeyCloak

Содержимое файла docker-compose.yaml в директории /home/keycloak:

```
version: '3.8'

services:
  keycloak:
    image: quay.io/keycloak/keycloak:26.3.2
    container_name: keycloak
```

```
    volumes:
      - ./certs:/etc/x509/certs
    environment:
      KC_DB: dev-file
      KEYCLOAK_ADMIN: admin
      KEYCLOAK_ADMIN_PASSWORD: admin
      KC_HOSTNAME: test.arch-guru.ru
      KC_HOSTNAME_STRICT: false
      KC_HOSTNAME_STRICT_HTTPS: true
      KC_PROXY: edge
      KC_HTTP_ENABLED: false
      KC_HEALTH_ENABLED: true
      KC_METRICS_ENABLED: true
      KC_LOG_LEVEL: INFO
      KC_TLS_SKIP_VERIFY: true
      KC_HTTPS_CERTIFICATE_FILE: /etc/x509/certs/fullchain.crt
      KC_HTTPS_CERTIFICATE_KEY_FILE: /etc/x509/certs/server.key
      KC_TLS_CIPHER_SUITES:
"TLS_AES_128_GCM_SHA256,TLS_CHACHA20_POLY1305_SHA256,TLS_AES_256_GC
M_SHA384,ECDHE-RSA-AES128-GCM-SHA256,ECDHE-RSA-AES256-GCM-
SHA384,DHE-RSA-AES128-GCM-SHA256,DHE-RSA-AES256-GCM-SHA384"
    command: start
    ports:
      - "8443:8443"
    networks:
      - keycloak-network
    restart: unless-stopped

networks:
  keycloak-network:
    driver: bridge
```

8. Установка PlantUML

Для установки PlantUML необходимо ввести следующие команды в консоли:

- Перейти в каталог PlantUML: cd /home/plantuml
- Запустить docker-compose.yaml: docker-compose up

После установки необходимо проверить корректность установки PlantUML. Для этого нужно:

- Зайти в интерфейс PlantUML: http://89.108.124.153:8079

Содержимое файла docker-compose.yaml в директории /home/plantuml:

```
version: "3"
services:
  plantuml:
    image: seaf/plantuml-server:jetty
    container_name: plantuml
    ports:
```

```
    - "8079:8080"
  restart: unless-stopped
```

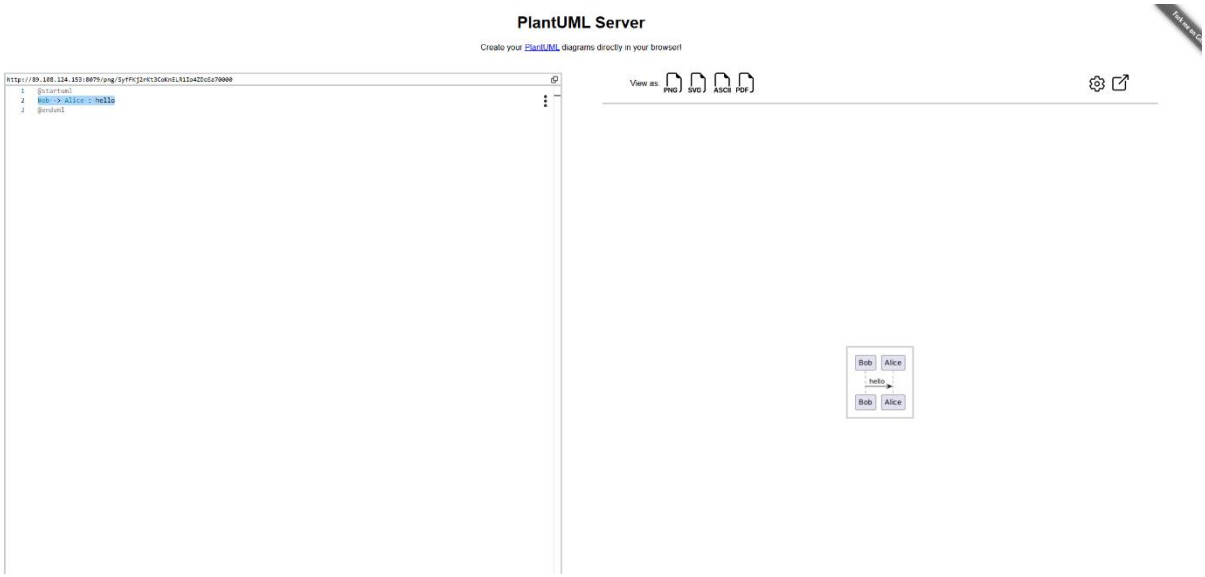


Рисунок 7. Веб-интерфейс PlantUML

9. Установка стартовой страницы ArchTech

Для установки стартовой страницы ArchTech необходимо ввести следующие команды в консоли:

- Перейти в каталог со стартовой страницей ArchTech: cd /home/startpage
- Запустить docker-compose.yaml: docker-compose

После установки необходимо проверить корректность установки стартовой страницы ArchTech. Для этого нужно:

- Зайти в интерфейс установки стартовой страницы ArchTech: https://test.arch-guru.ru

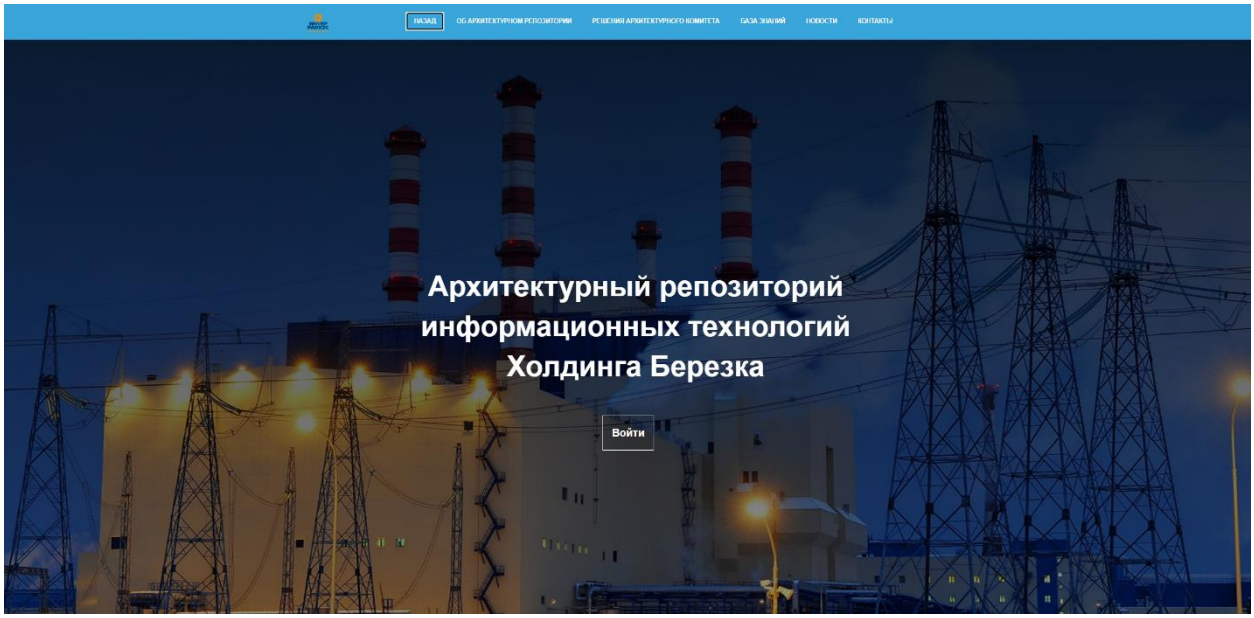


Рисунок 7. Веб-интерфейс стартовой страницы Архитектурного репозитория

Содержимое файла docker-compose.yaml в директории /home/startpage:

```
version: '3.8'
services:
  web:
    build:
      context: .
      dockerfile: Dockerfile
    container_name: startpage
    image: startpage
    ports:
      - "80:80"             # Обычный HTTP-порт
      - "443:443"           # Безопасный HTTPS-порт
    restart: unless-stopped
    environment:
      - NGINX_SSL_CERTIFICATE=/etc/nginx/certs/fullchain.crt
      - NGINX_SSL_CERTIFICATE_KEY=/etc/nginx/certs/server.key
    volumes:
      - ./certs:/etc/nginx/certs # Монтируем директорию с сертификатами в контейнер
```

10. Установка и настройка Портала Архитектурного репозитория ArchTech

Для установки Портала Архитектурного репозитория ArchTech необходимо ввести следующие команды в консоли:

- Перейти в каталог с Порталом Архитектурного репозитория ArchTech: cd /home/archtech
- Запустить docker-compose.yaml со сборкой приложения: docker-compose up --build

После установки необходимо проверить корректность установки Портала Архитектурного репозитория ArchTech. Для этого нужно:

- Зайти в интерфейс Портала Архитектурного репозитория ArchTech: https://test.arch-guru.ru:7443
- Нажать на кнопку Войти в вернем левом углу экрана
- Авторизоваться в KeyCloak с помощью следующих учётных данных:
  - Логин: idm83
  - Пароль: Для получения пароля необходимо направить запрос на ящик info@arch-guru.ru

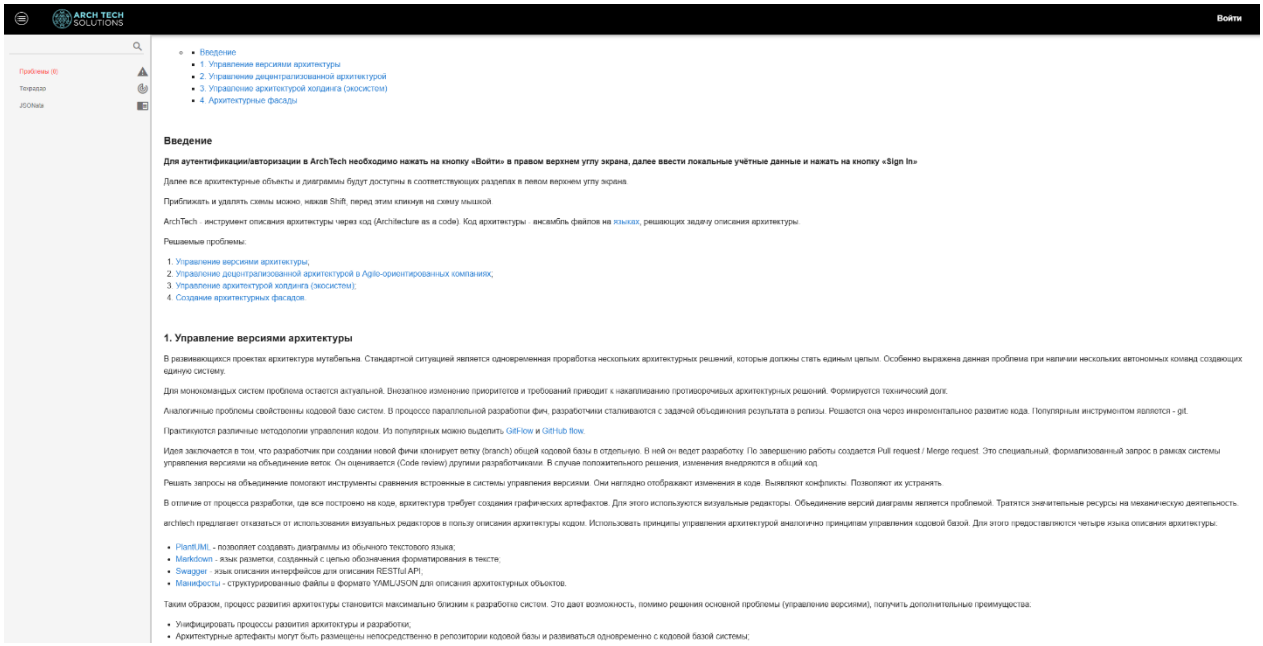


Рисунок 8. Общий вид Портала Архитектурного репозитория до авторизации

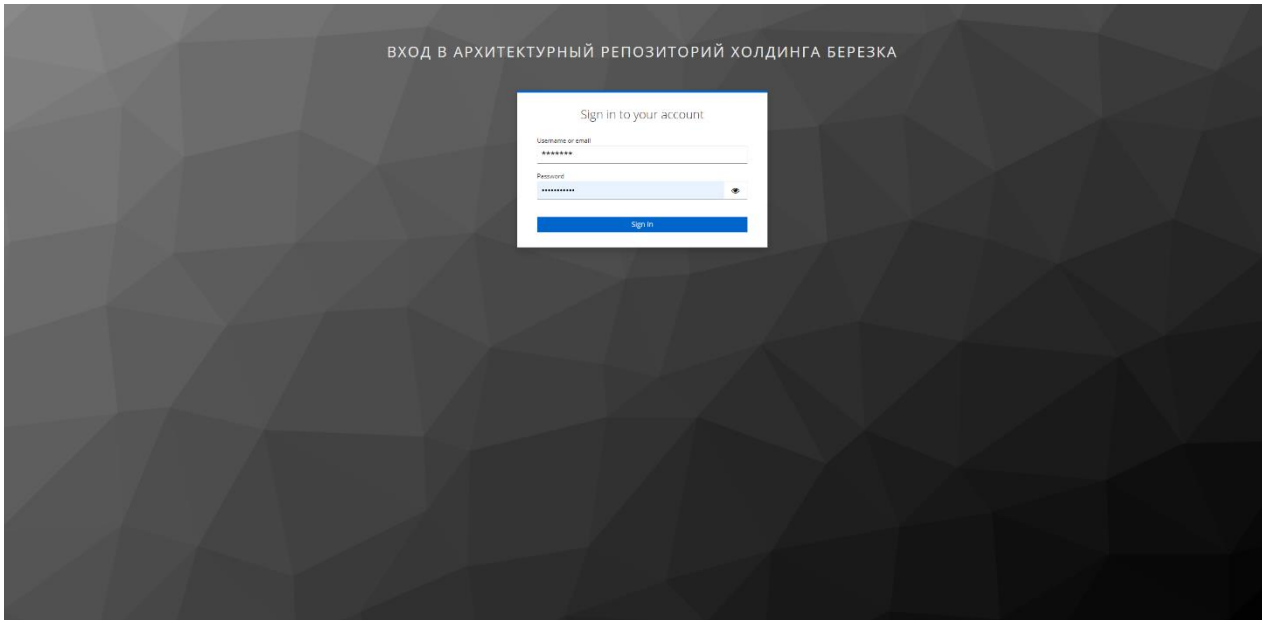


Рисунок 9. Окно аутентификации и авторизации в Портале Архитектурного репозитория

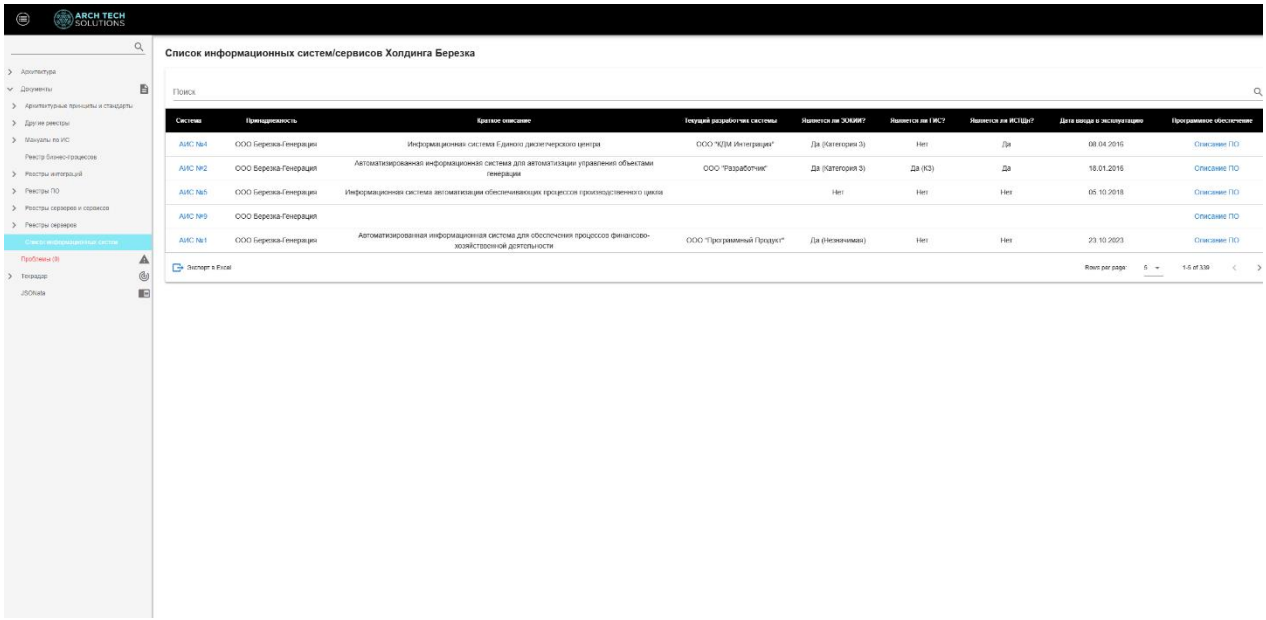


Рисунок 10. Общий вид Портала Архитектурного репозитория после авторизации

Содержимое файла docker-compose.yaml в директории /home/archtech:

```
version: "3.8"

services:
  postgres:
    image: postgres:15
    container_name: postgres
    environment:
      POSTGRES_PASSWORD: admin
      POSTGRES_USER: admin
      POSTGRES_DB: archtech
      PGDATA: /var/lib/postgresql /data/pgdata
    ports:
      - "5432:5432"
    volumes:
      - ./pgdata:/var/lib/postgresql/data/pgdata
    restart: unless-stopped
  nginx:
    container_name: archtech
    image: archtech
    build:
      context: .
      dockerfile: Dockerfile
      target: builder
    env_file:
      - .env
    ports:
      - "8080:8080"
    restart: unless-stopped
  reverse-proxy:
```

```
    build:
      context: .
      dockerfile: Dockerfile
    container_name: reverse-proxy
    image: nginx
    ports:
      - "7443:7443"          # Безопасный HTTPS-порт
      - "8433:8433"          # Безопасный HTTPS-порт
    environment:
      - NGINX_SSL_CERTIFICATE=/etc/nginx/certs/fullchain.crt
      - NGINX_SSL_CERTIFICATE_KEY=/etc/nginx/certs/server.key
    volumes:
      - ./certs:/etc/nginx/certs # Монтируем директорию с сертификатами в контейнер
    restart: unless-stopped
volumes:
  pgdata:
    driver: local
```

Также необходимо проверить доступность сервисов по HTTPS:

Сервис PlantUML: https://test.arch-guru.ru:8433

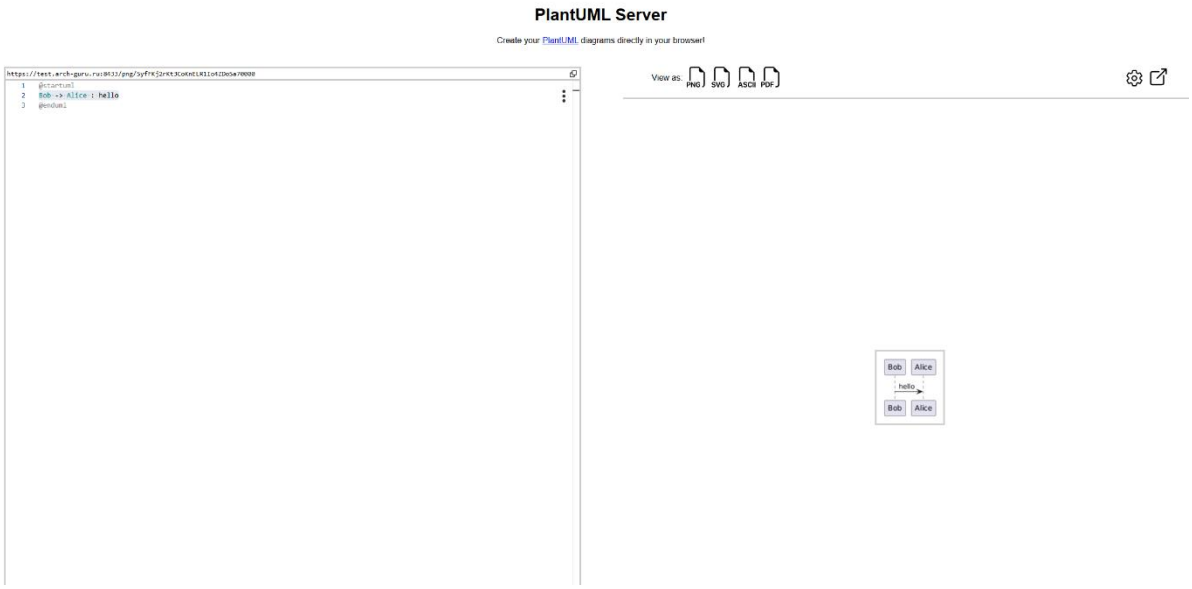


Рисунок 11. Веб-интерфейс PlantUML с доступом по HTTPS

11. Проверка корректности создания всех контейнеров

Для этого необходимо открыть консоль сервера и ввести следующие команды:

- Список всех запущенных контейнеров: docker ps –a
- Список всех образов контейнеров: docker images

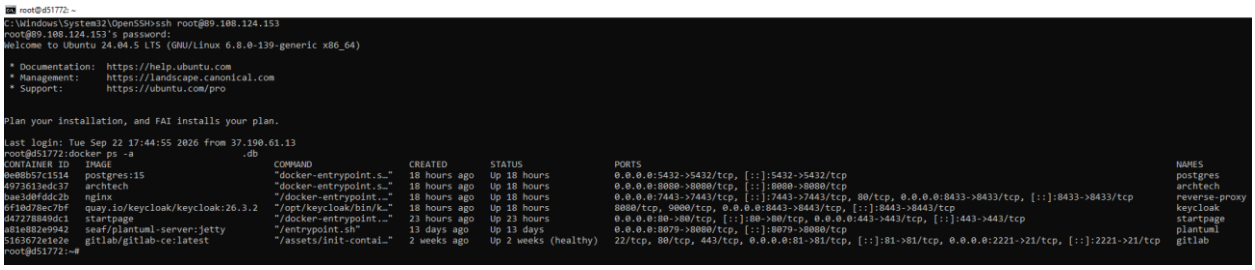


Рисунок 12. Визуализация всех запущенных контейнеров на сервере

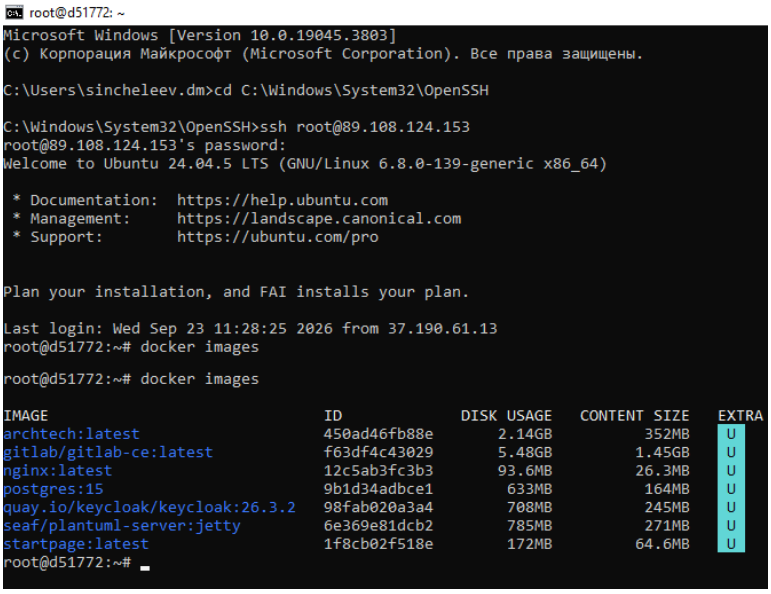


Рисунок 13. Визуализация всех образов контейнеров на сервере