|  |
| --- |
| A black and white emblem with two birds and a scroll  AI-generated content may be incorrect. |
| МИНОБРНАУКИ РОССИИ  Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** |

**РАБОТА ДОПУЩЕНА К ЗАЩИТЕ**

Руководитель

программы \_\_\_\_\_\_\_\_\_\_\_ А.Т. Тарланов

«17» июня 2025 г.

**ИТОГОВАЯ АТТЕСТАЦИОННАЯ РАБОТА**

по дополнительной программы профессиональной переподготовки

«Технологии Девопс»

На тему: **«**Набор тематических заданий по проведению итоговой аттестации DevOps-инженера – вариант 1122**»**

Обучающийся *Павлов Никита Сергеевич*

*Подпись Фамилия, имя, отчество*

|  |  |  |
| --- | --- | --- |
|  |  |  |
| группа | ИКБО-50-23 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Руководитель работы** | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *подпись* |  | А.Т. Тарланов |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Москва 2025 г.

**СОДЕРЖАНИЕ**

[1. ВВЕДЕНИЕ 3](#_Toc199704462)

[2. ОСНОВНАЯ ЧАСТЬ 5](#_Toc199704463)

[2.1 Модуль 6. Администрирование баз данных 5](#_Toc199704464)

[2.2 Модуль 7. Масштабируемость и отказоустойчивость 10](#_Toc199704465)

[2.3 Модуль 8. Мониторинг, логирование и оповещение событий 14](#_Toc199704466)

[2.4 Модуль 9. Виртуализация в DevOps 20](#_Toc199704467)

[2.5 Модуль 12. Конфигурационное управление. Что такое IaC 22](#_Toc199704468)

[2.6 Модуль 13. Системы контроля версий. Распределённая система управления версиями Git 25](#_Toc199704469)

[2.7 Модуль 14. Жизненный цикл ПО 28](#_Toc199704470)

[2.8 Модуль 15. Практические навыки работы с Docker 33](#_Toc199704471)

[3 ЗАКЛЮЧЕНИЕ 35](#_Toc199704472)

[СПИСОК ЛИТЕРАТУРЫ 37](#_Toc199704473)

# 1. ВВЕДЕНИЕ

Современная DevOps-экосистема требует комплексного подхода, объединяющего администрирование инфраструктуры, автоматизацию процессов, контроль версий и мониторинг систем. Рост спроса на специалистов, способных работать с распределёнными системами, конвейерами CI/CD и облачными технологиями, делает освоение ключевых инструментов DevOps критически важным для ИТ-отрасли.

Данная итоговая работа посвящена практическому освоению технологий, формирующих основу DevOps-практик. В рамках выполнения заданий модулей №6, 7, 8, 9, 12, 13, 14 и 15 дополнительной профессиональной программы «Технологии Девопс» были реализованы следующие аспекты:

* *Администрирование PostgreSQL:* от установки СУБД до проектирования структуры данных, создания пользователей, управления базами данных и выполнения SQL-запросов, включая операции SELECT, INSERT и другие.
* *Настройка отказоустойчивого кластера:* работа с инструментами HAProxy и Pacemaker для обеспечения масштабируемости и отказоустойчивости, включая настройку SSH-ключей, синхронизацию времени и управление узлами.
* *Мониторинг и логирование:* развёртывание стека мониторинга на базе Prometheus, Grafana и Node Exporter, настройка дашбордов и сбор метрик для анализа производительности системы.
* *Виртуализация:* установка и настройка инструментов, таких как vim, mc и net-tools, для работы в виртуальных средах.
* *Конфигурационное управление (IaC):* использование Ansible для автоматизации управления инфраструктурой, создание и выполнение playbook'ов.
* *Системы контроля версий:* развёртывание Git-сервера, управление репозиториями, создание веток, слияние изменений и организация совместной работы.
* *Жизненный цикл ПО:* настройка Jenkins для автоматизации CI/CD-процессов, включая установку плагинов, создание пользователей и управление задачами.
* *Работа с Docker:* создание Docker-образов, настройка контейнеров для развёртывания приложений, включая PostgreSQL, и управление их жизненным циклом.

Особое внимание уделено интеграции компонентов в единую экосистему: от контроля версий исходного кода через Git до визуализации метрик работы системы в Grafana. Практические задания охватывают полный цикл DevOps-процессов — настройку окружения, развёртывание сервисов, конфигурацию инструментов мониторинга и автоматизацию сборки.

Цель работы — формирование профессиональных компетенций в области DevOps через практическое взаимодействие с ключевыми технологиями: управление базами данных, обеспечение отказоустойчивости, настройка систем мониторинга, автоматизация жизненного цикла ПО и работа с контейнеризацией. Результаты демонстрируют готовность к реализации реальных задач в области сопровождения и развития ИТ-инфраструктуры.

# 2. ОСНОВНАЯ ЧАСТЬ

## 2.1 Модуль 6. Администрирование баз данных

Начальный этап включал загрузку и установку PostgreSQL с использованием скрипта (Рисунок 1). Это обеспечило корректную инсталляцию СУБД и подготовку окружения для дальнейшей работы.

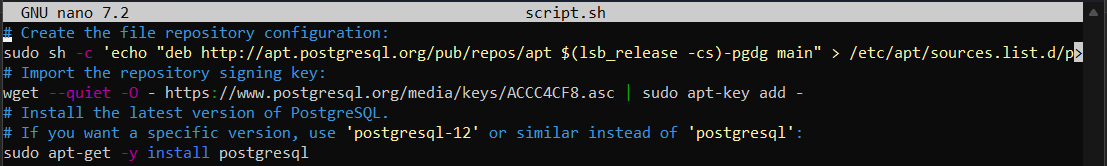


Рисунок 1 – Скрипт установки PostgreSQL

После установки была запущена служба PostgreSQL (Рисунок 2), что подтвердило её работоспособность. Проверка статуса службы показала, что СУБД готова к использованию.

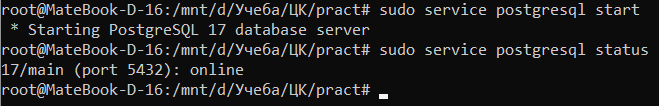


Рисунок 2 – Запуск службы PostgreSQL и проверка работоспособности

Для выполнения административных задач был осуществлён вход в системный аккаунт postgres (Рисунок 3). Это позволило управлять базами данных и пользователями.

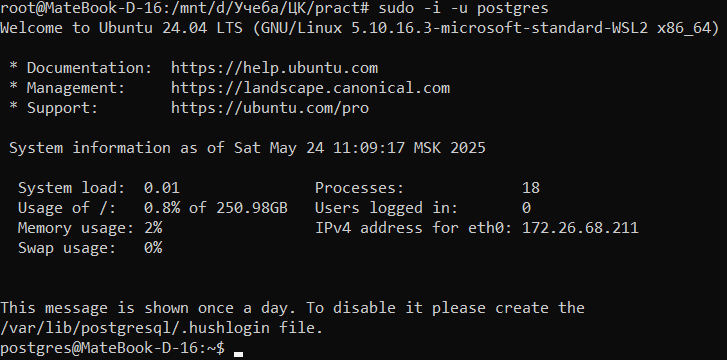


Рисунок 3 – Вход в аккаунт postgres

Работоспособность консоли PostgreSQL была проверена (Рисунок 4), что подтвердило возможность выполнения SQL-запросов и взаимодействия с СУБД.

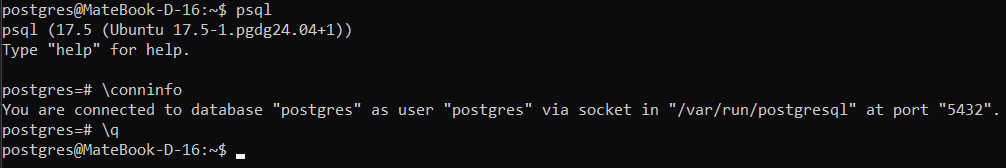


Рисунок 4 – Проверка работоспособности консоли

Был создан пользователь nikita (Рисунок 5) с назначенными правами. Это обеспечило разграничение доступа и безопасность работы с базами данных.

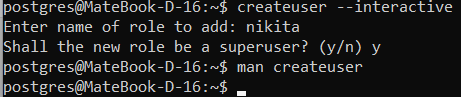


Рисунок 5 – Создание нового пользователя

Под учётной записью nikita была создана новая база данных (Рисунки 6 и 7). Это продемонстрировало навыки управления объектами СУБД.

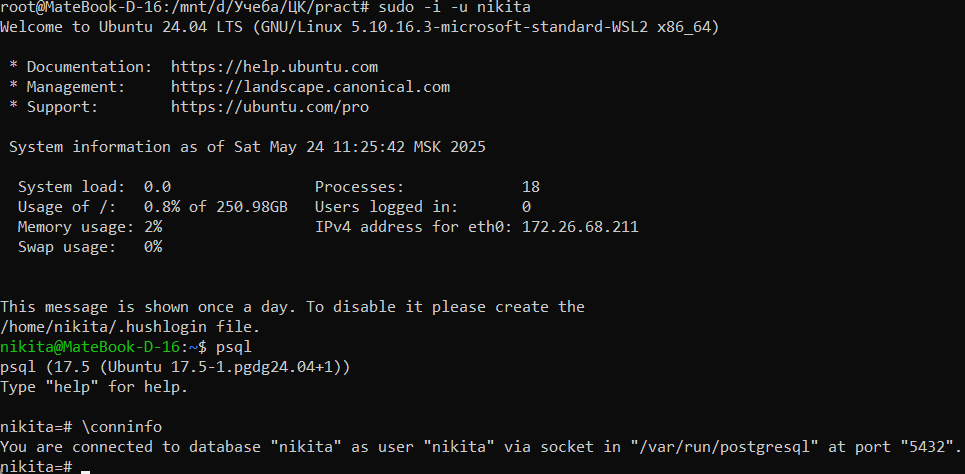


Рисунок 6 – Создание базы данных под учетной записью nikita

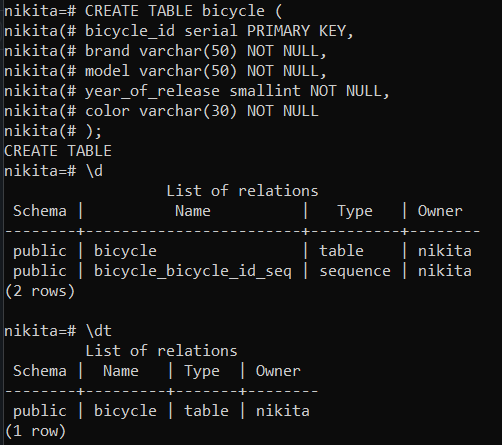


Рисунок 7 – Создание и просмотр базы данных

В созданной базе данных была сформирована таблица, заполнена данными (Рисунок 8) и проверена с помощью оператора SELECT (Рисунки 9 и 11). Также были добавлены новые записи (Рисунок 10), что показало умение выполнять основные операции с данными.

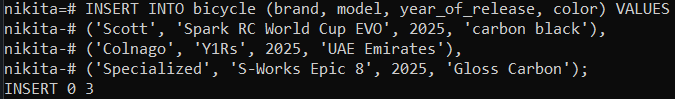


Рисунок 8 – Операция заполнения таблицы данными

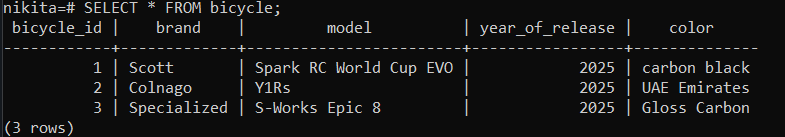


Рисунок 9 – Использование оператора SELECT для вывода записей на экран

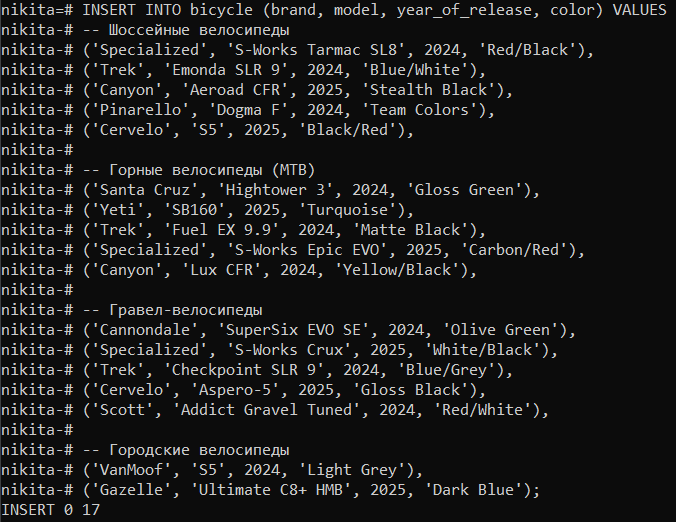


Рисунок 10 – Добавление новых данных в таблицу БД

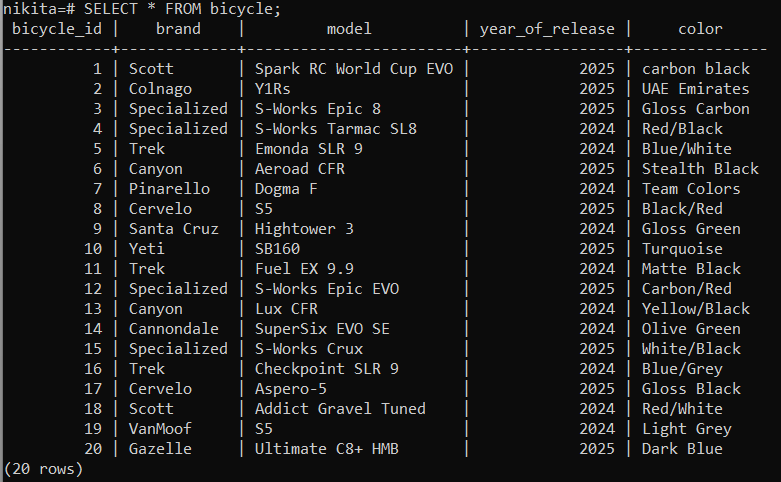


Рисунок 11 – Использование оператора SELECT для вывода всех записей БД

## 2.2 Модуль 7. Масштабируемость и отказоустойчивость

Для корректной работы кластера была выполнена синхронизация времени между узлами с помощью timedatectl (Рисунок 12). Это важно для согласованного взаимодействия серверов в распределённой системе.

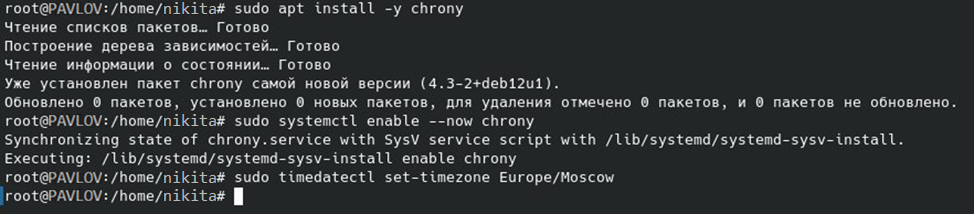


Рисунок 12 – Настройка синхронизации времени

Проверены параметры IP-адресов узлов (Рисунок 13), созданы и распределены SSH-ключи (Рисунки 14–15). Это обеспечило безопасное управление узлами кластера без необходимости ввода пароля.



Рисунок 13 – Параметры ip узлов в сети

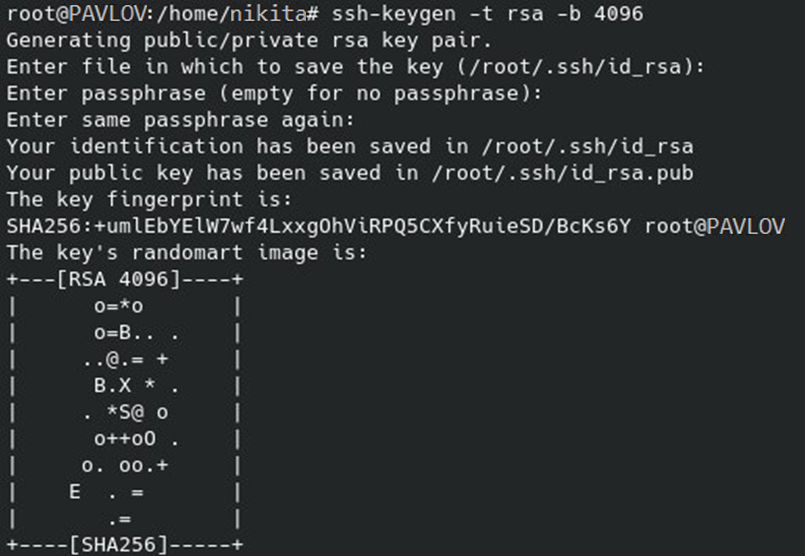


Рисунок 14 – Создание ssh-ключа

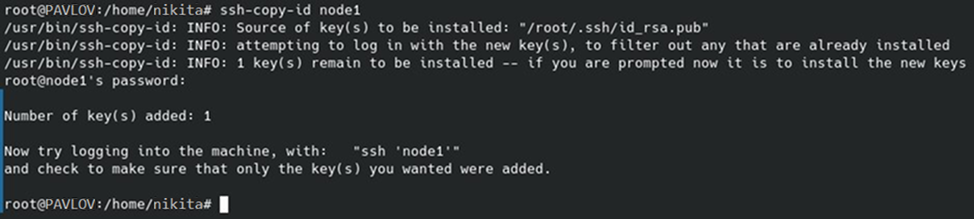


Рисунок 15 – Настройка связи по ssh

Установлен пароль для пользователя hacluster (Рисунок 16) и выполнена аутентификация на узле node1 (Рисунок 17).

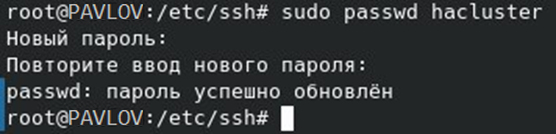


Рисунок 16 – Смена пароля пользователя hacluster

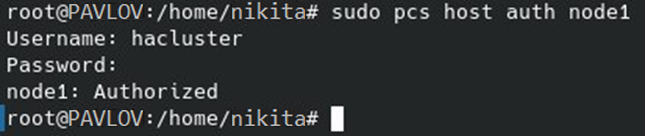


Рисунок 17 – Аутентификация на node1

Создан кластер pgcluster (Рисунок 18), запущен и активирован (Рисунки 19–20). Проверка статуса (Рисунок 21) подтвердила его работоспособность.

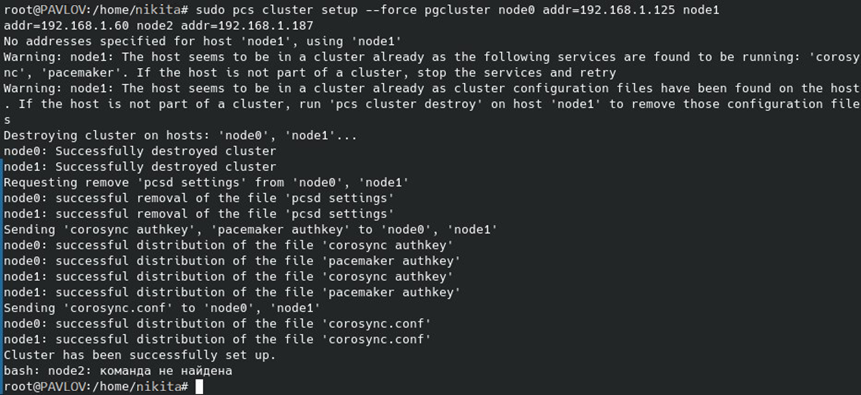


Рисунок 18 – Создание кластера pgcluster

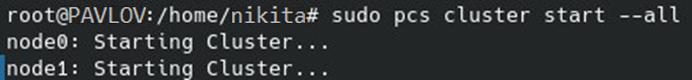


Рисунок 19 – Запуск кластера



Рисунок 20 – Выполнение enable

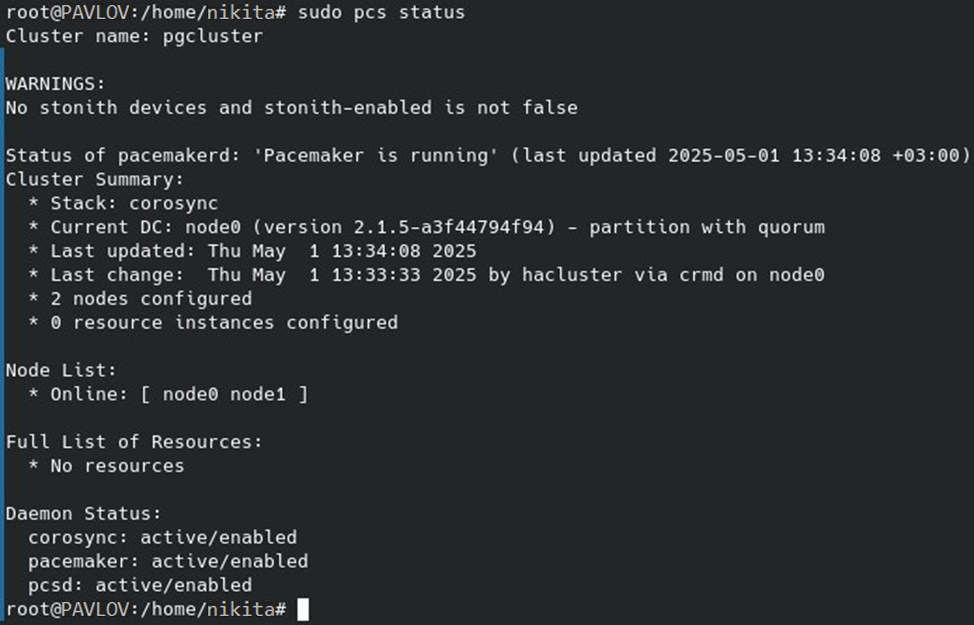


Рисунок 21 – Проверка статуса кластера

Создана кластерная БД (Рисунок 22), запущена (Рисунок 23) и настроена (Рисунок 25). Смена пароля пользователя (Рисунок 24) обеспечила безопасность доступа.

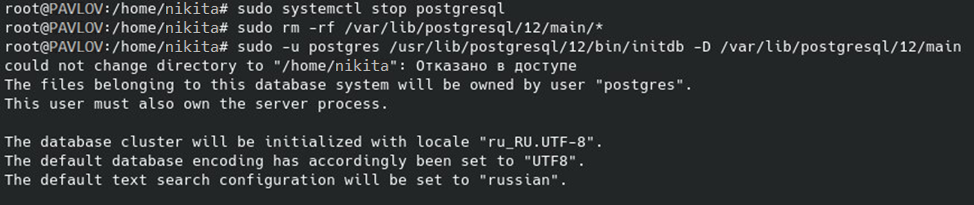


Рисунок 22 – Создание кластера БД

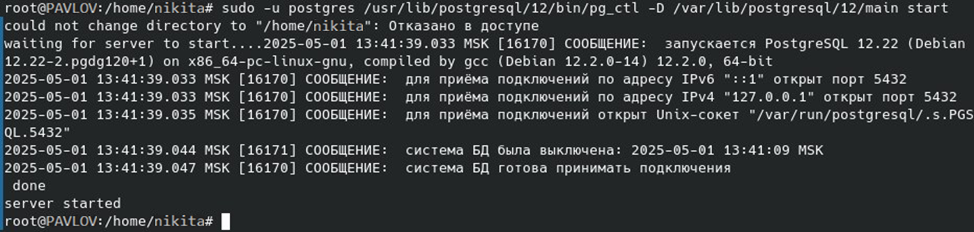


Рисунок 23 – Запуск БД

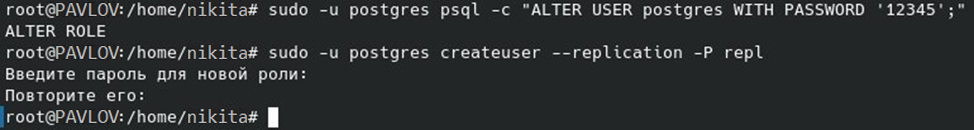


Рисунок 24 – Смена пароля пользователя

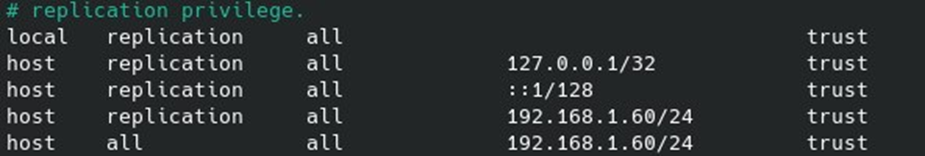


Рисунок 25 – Содержимое файла конфигурации

## 2.3 Модуль 8. Мониторинг, логирование и оповещение событий

Загружен и распакован пакет node\_exporter (Рисунки 26–27) для сбора системных метрик.

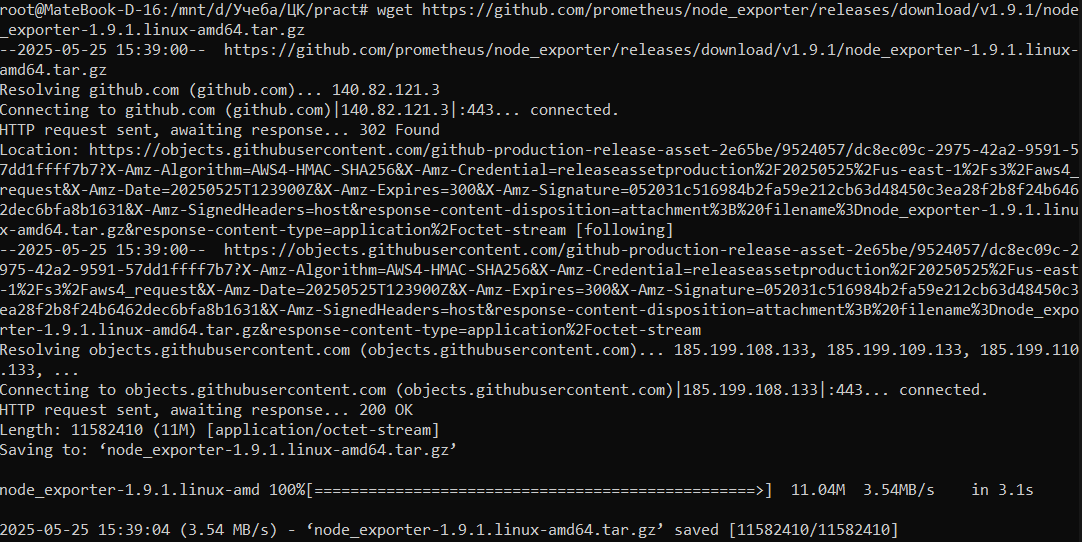


Рисунок 26 – Загрузка пакета node\_exporter

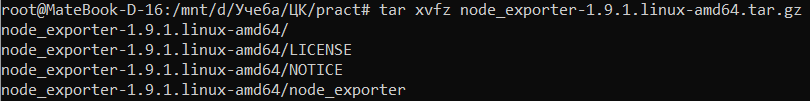


Рисунок 27 – Извлечение файла из архива

Настроены права доступа (Рисунок 28) и запущен сервис (Рисунки 29–30). Проверка через порт 9100 (Рисунок 31) подтвердила сбор данных.



Рисунок 28 – Смена разрешений файла (исполняемый)



Рисунок 29 – Запуск node\_exporter (часть 1)

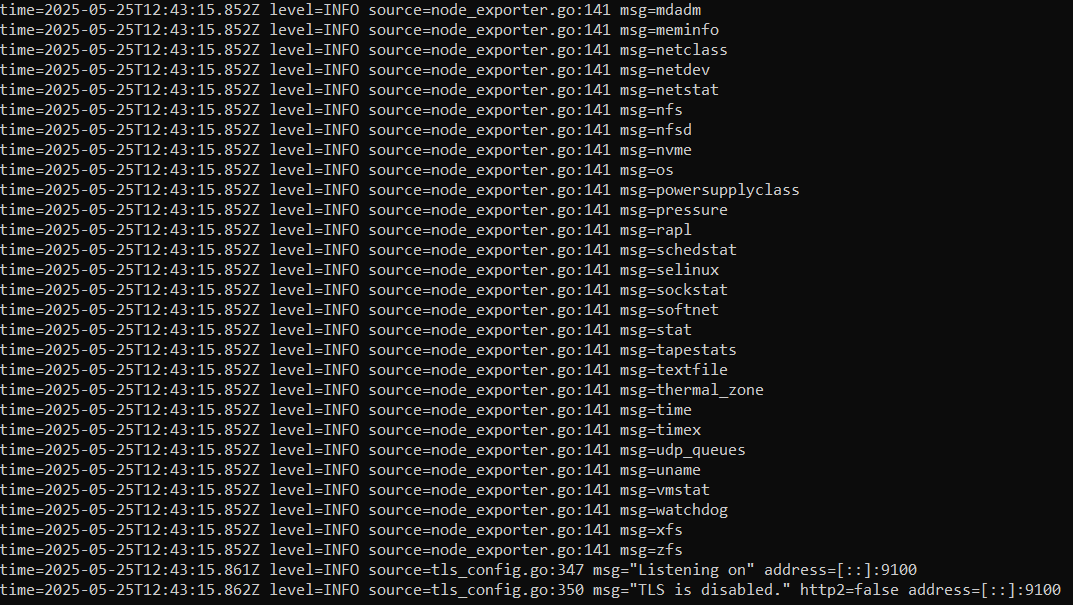


Рисунок 30 – Запуск node\_exporter (часть 2)

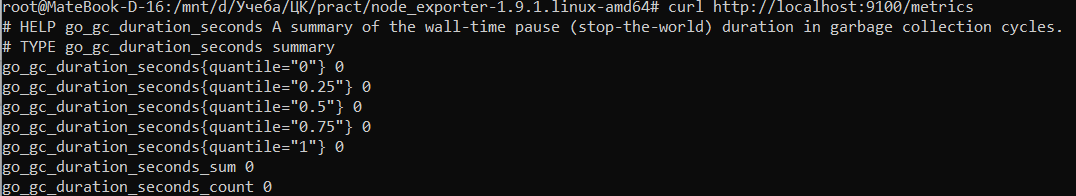


Рисунок 31 – Проверка метрик через порт 9100

Загружен архив Prometheus (Рисунок 32), распакован (Рисунок 33) и настроен конфигурационный файл (Рисунок 34).

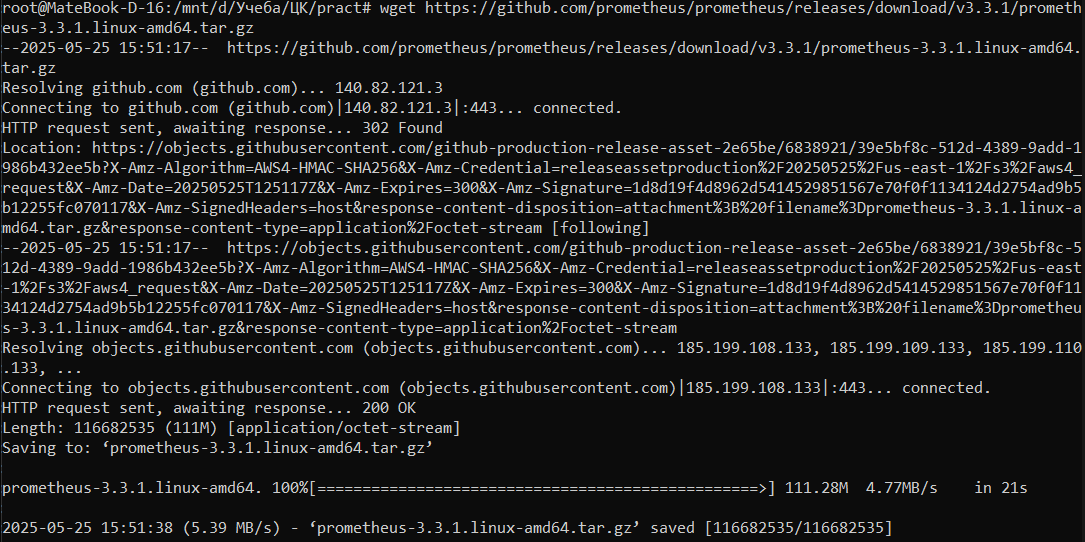


Рисунок 32 – Загрузка архива пакета Prometheus

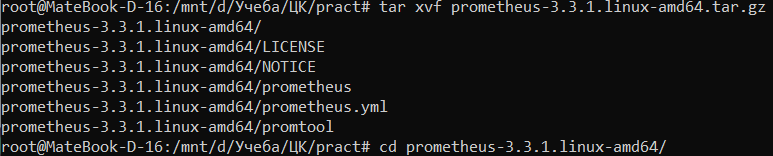


Рисунок 33 – Распаковка архива

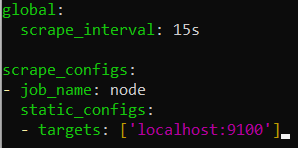


Рисунок 34 – Данные файла конфигурации

Запуск сервиса (Рисунок 35) обеспечил сбор и хранение метрик с Node Exporter.

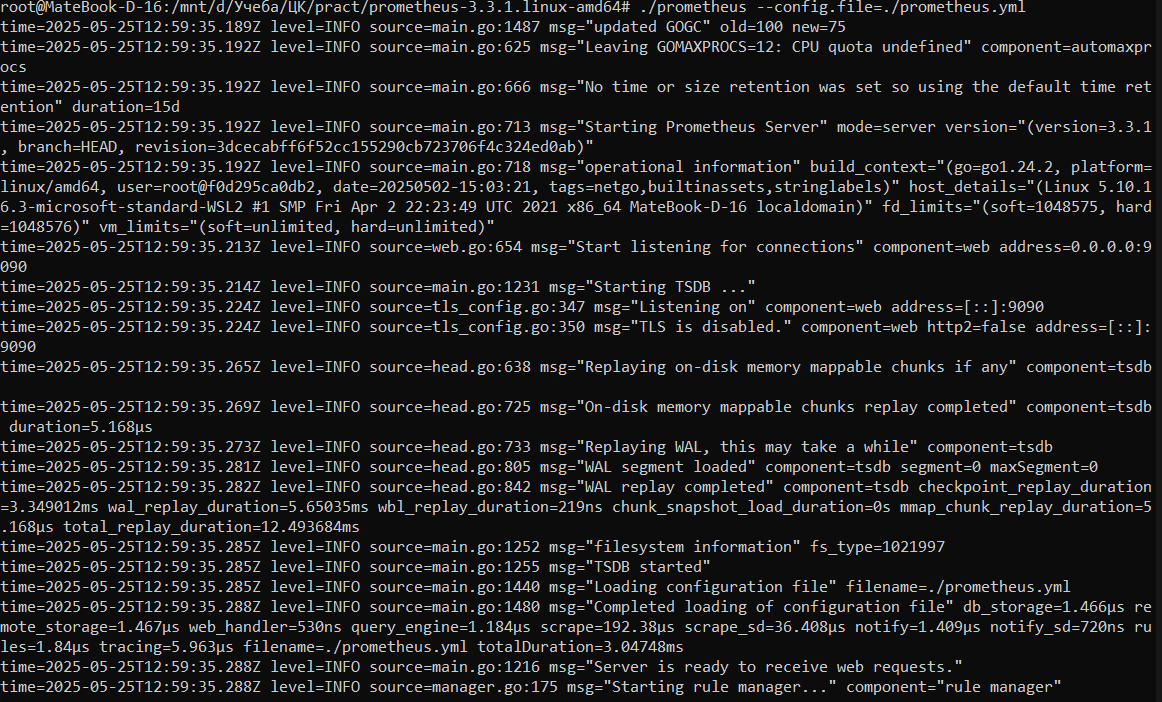


Рисунок 35 – Запуск Prometheus

Установлен пакет Grafana (Рисунки 36–37), сервис запущен и проверен (Рисунок 38).

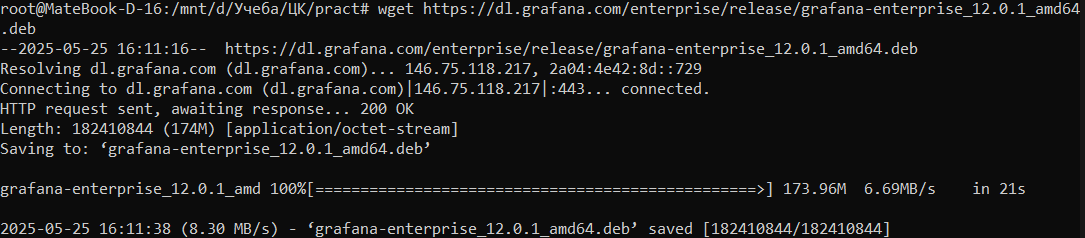


Рисунок 36 – Загрузка пакета Grafana

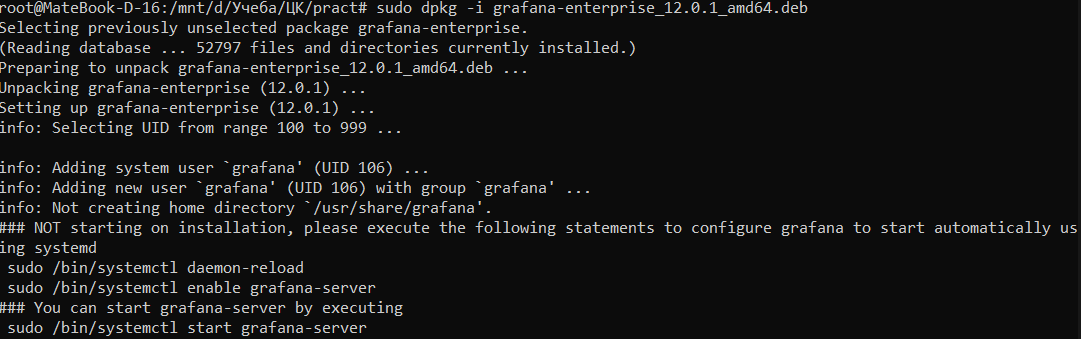


Рисунок 37 – Установка Grafana

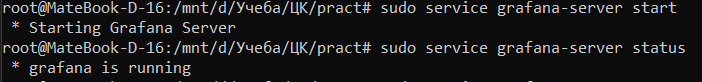


Рисунок 38 – Запуск Grafana и проверка статуса

Настроен источник данных Prometheus (Рисунок 39) и импортирован готовый дашборд (Рисунок 40). Итоговая визуализация метрик (Рисунок 41) показала работоспособность стека мониторинга.

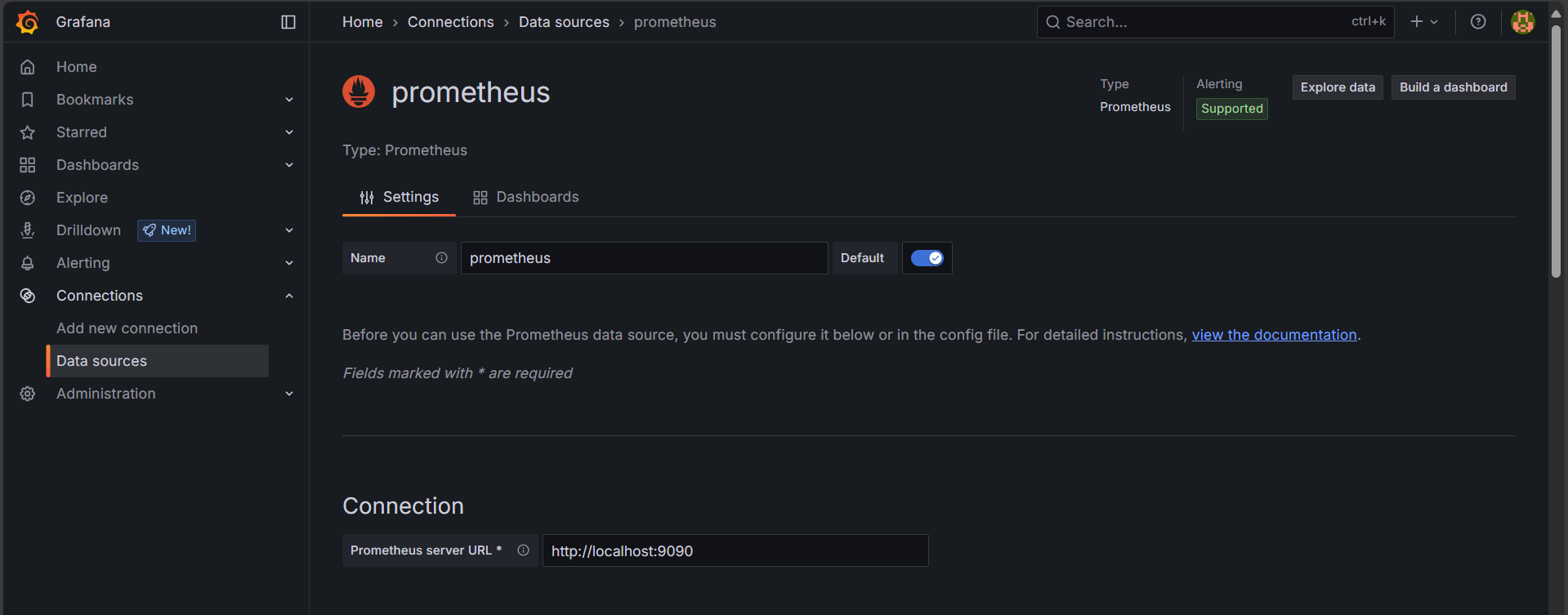


Рисунок 39 – Настройка источника данных

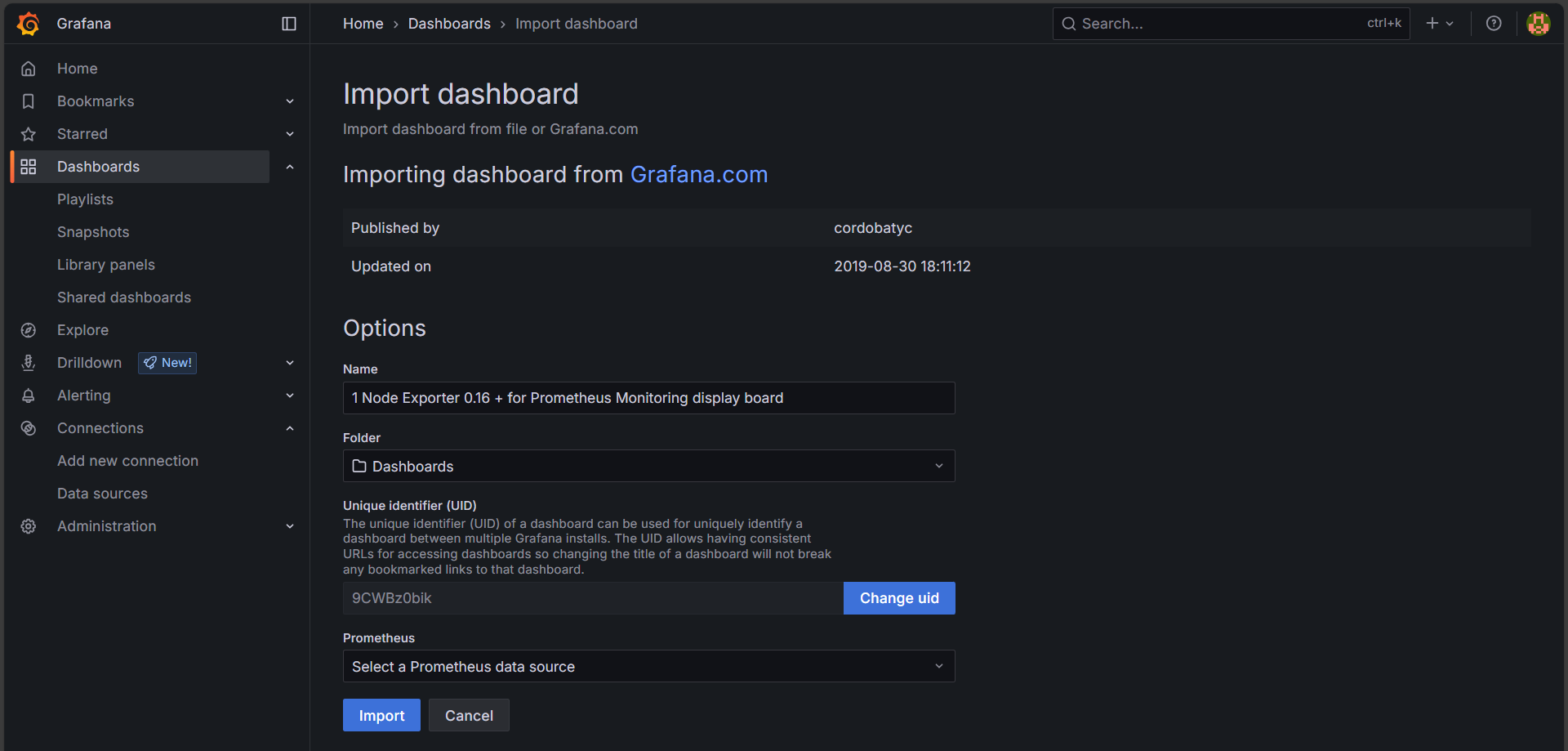


Рисунок 40 – Импорт готового дашборда

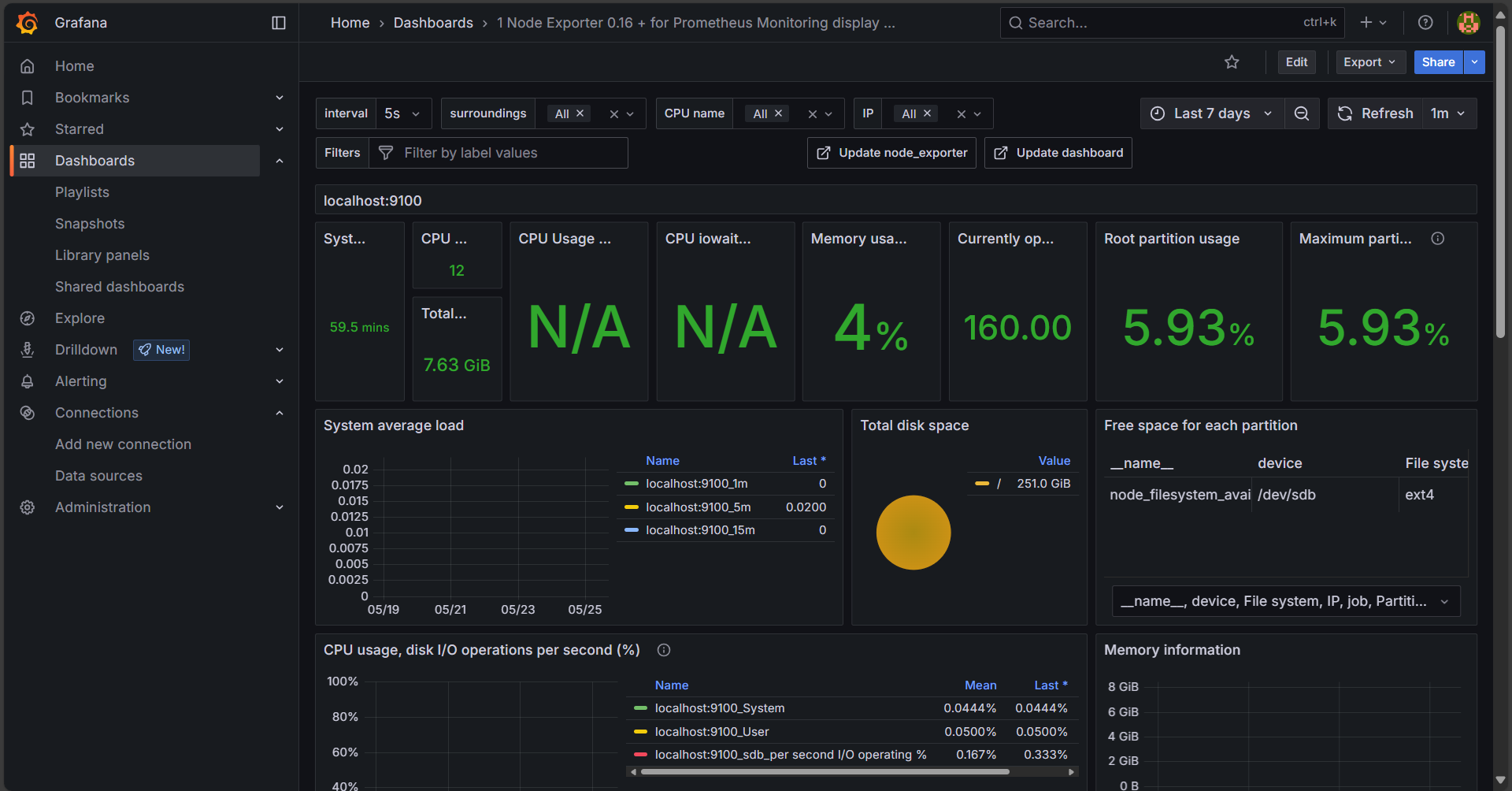


Рисунок 41 – Итоговые данные на дашборде

## 2.4 Модуль 9. Виртуализация в DevOps

На предварительно развернутой ВМ с Ubuntu 24.04 выполнено обновление пакетов системы (Рисунок 42) для обеспечения актуальности всех компонентов перед установкой дополнительных инструментов.

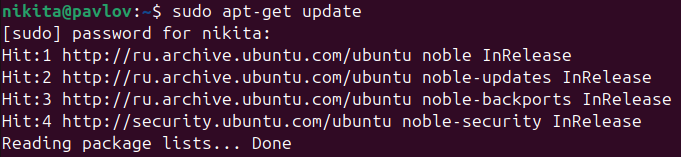


Рисунок 42 – Обновление пакета установки

Установлен текстовый редактор vim (Рисунок 43) с проверкой его версии (Рисунок 44), что обеспечило удобное редактирование конфигурационных файлов в терминале.

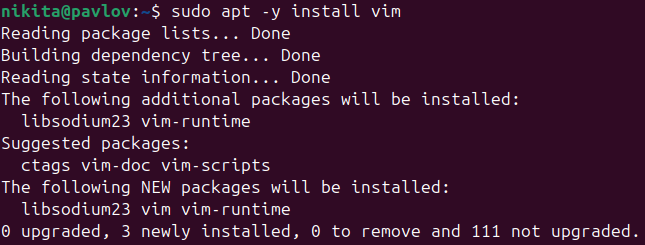


Рисунок 43 – Установка vim

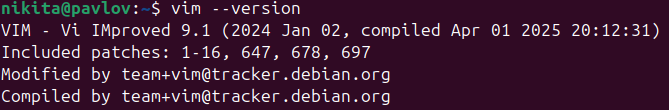


Рисунок 44 – Проверка версии vim

Инсталлирован файловый менеджер mc (Midnight Commander) (Рисунок 45) с подтверждением версии (Рисунок 46), значительно упрощающий навигацию по файловой системе.

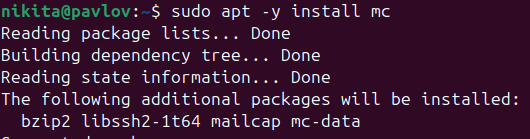


Рисунок 45 – Установка mc

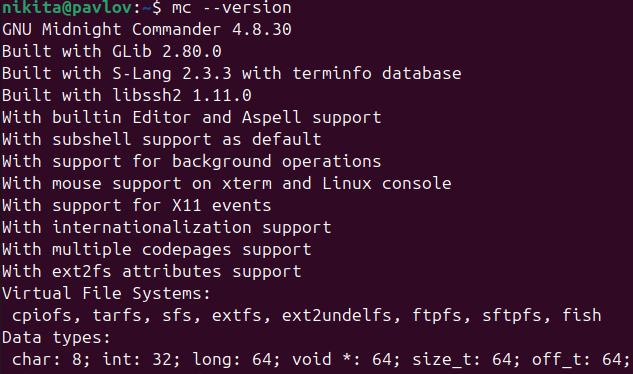


Рисунок 46 – Проверка версии mc

Добавлен пакет net-tools (Рисунок 47), включающий утилиту ifconfig (Рисунок 48), что позволило проводить базовую диагностику сетевых подключений.

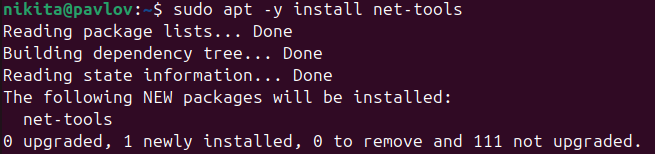


Рисунок 47 – Установка net-tools



Рисунок 48 – Проверка версии ifconfig

## 2.5 Модуль 12. Конфигурационное управление. Что такое IaC

Установлена и проверена версия Ansible (Рисунок 49), что подтвердило готовность системы к автоматизации управления конфигурациями.

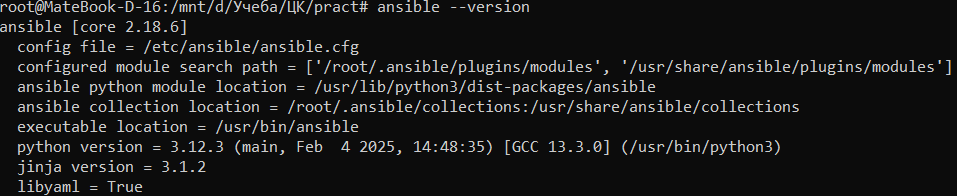


Рисунок 49 – Установленная версия ansible

Проведена настройка базового конфигурационного файла Ansible (Рисунок 50), включая определение рабочих параметров и переменных среды.

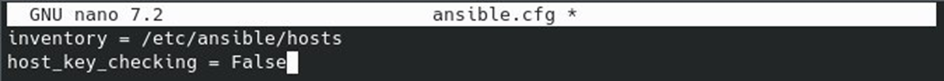


Рисунок 50 – Содержимое конфигурационного файла ansible

Выполнена верификация доступности управляемых узлов (Рисунок 51) с использованием инвентарного файла. Подтверждена корректность сетевых подключений и аутентификации между control-нодой и целевыми серверами



Рисунок 51 – Проверка доступности узлов

Создан комплексный playbook (Рисунки 52-53), содержащий:

* Настройку базовых системных параметров
* Управление пакетами и сервисами
* Конфигурацию сетевых компонентов
* Развертывание тестового окружения



Рисунок 52 – Содержимое playbook’а (часть 1)



Рисунок 53 – Содержимое playbook’а (часть 2)

Playbook успешно применен к целевым узлам (Рисунок 54).



Рисунок 54 – Результаты запуска playbook’а

## 2.6 Модуль 13. Системы контроля версий. Распределённая система управления версиями Git

Установлен и настроен Git (Рисунок 55), включая базовую конфигурацию пользователя (Рисунок 57). Создан системный пользователь git (Рисунок 56) для безопасного управления репозиториями.



Рисунок 55 – Установка Git

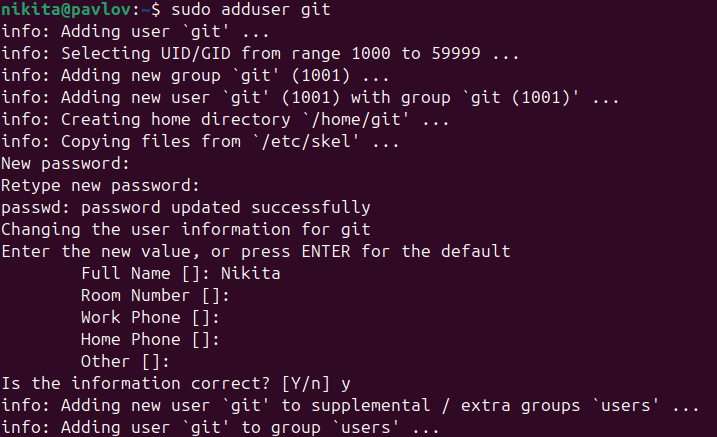


Рисунок 56 – Создание пользователя Git

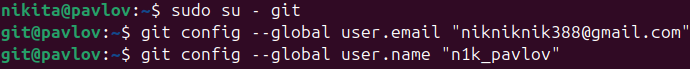


Рисунок 57 – Предварительные настройки Git

Инициализирован новый репозиторий (Рисунок 58).

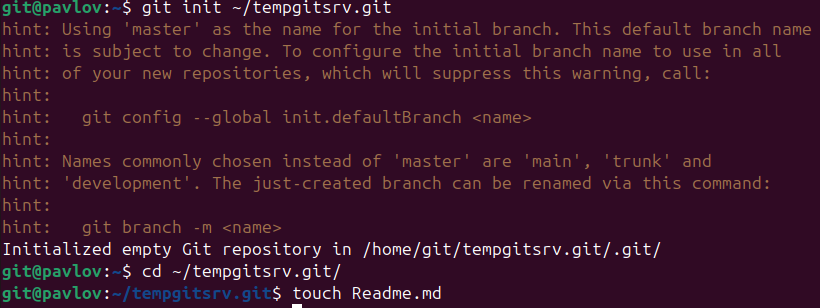


Рисунок 58 – Инициализация репозитория и добавление файла

Отработана процедура добавления файлов в индекс и создания коммитов (Рисунок 59).

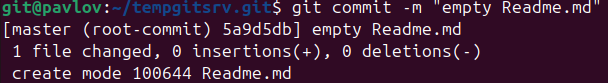


Рисунок 59 – Коммит изменения

Выполнено клонирование репозитория (Рисунок 60), демонстрирующее типовой workflow разработчика.

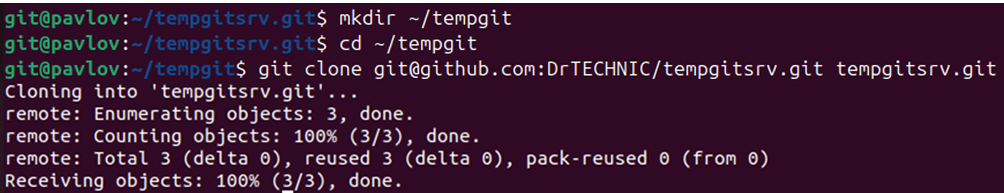


Рисунок 60 – Клонирование Git локально

Создана и переключена на новую ветку (Рисунок 61). Выполнена отправка изменений в удаленный репозиторий (Рисунок 62). Отработаны операции слияния веток (Рисунок 64). Продемонстрировано отображение всех веток репозитория (Рисунок 63).

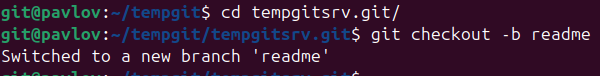


Рисунок 61 – Создание новой ветки

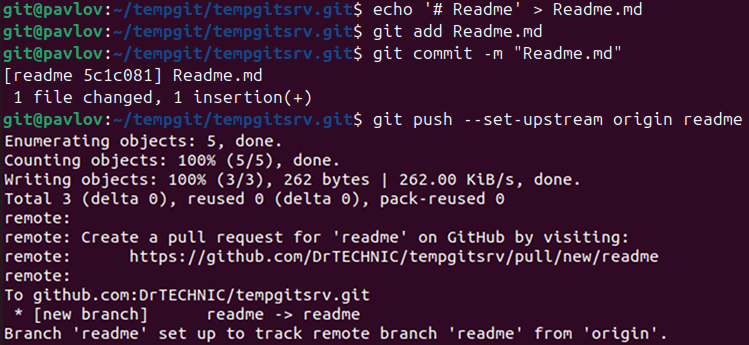


Рисунок 62 – Отправка изменений в репозиторий



Рисунок 63 – Список всех веток

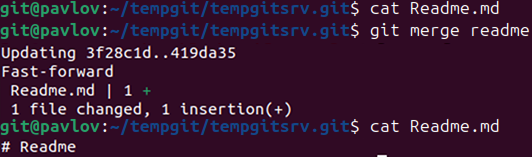


Рисунок 64 – Слияние веток

## 2.7 Модуль 14. Жизненный цикл ПО

Успешно выполнена установка Jenkins (Рисунки 65-66), включая:

* Настройку репозиториев пакетов
* Установку необходимых зависимостей
* Запуск сервиса



Рисунок 65 – Установка Jenkins (часть 1)

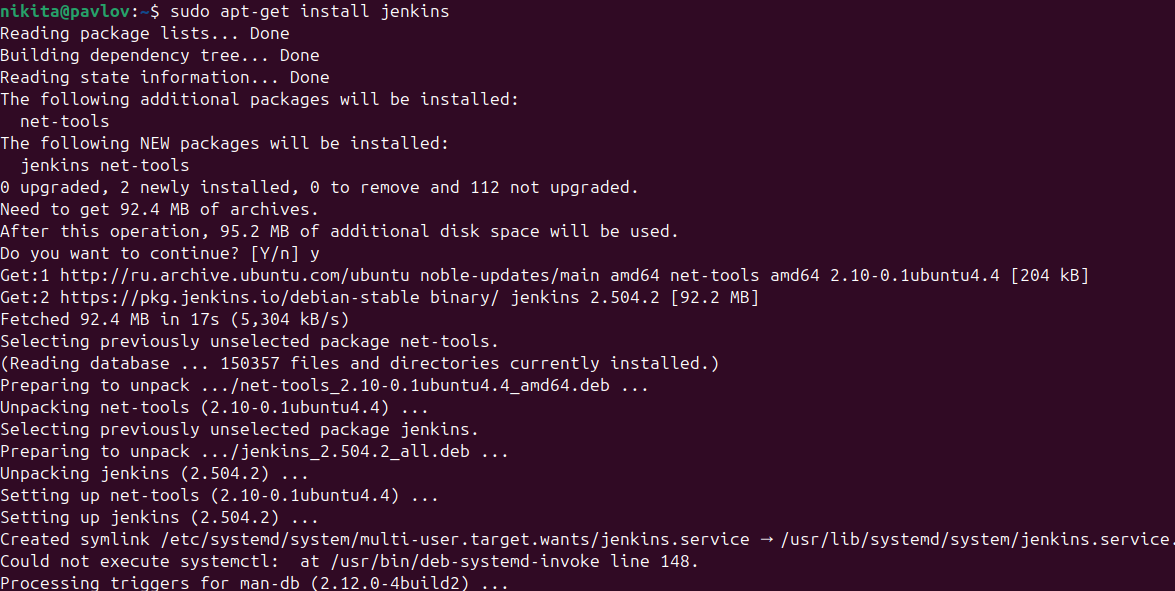


Рисунок 66 – Установка Jenkins (часть 2)

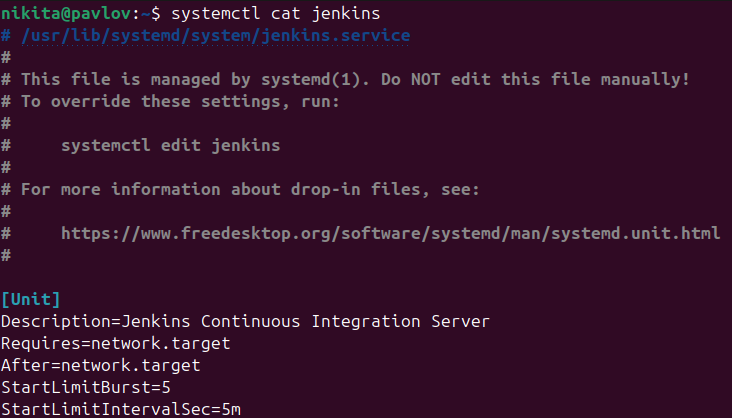


Рисунок 67 – Просмотр подробностей

Создан административный пользователь (Рисунки 68, 75) для безопасного управления сервером.



Рисунок 68 – Пользователь для Jenkins

Установлена Java (Рисунок 69) - обязательное требование для работы Jenkins. Проверена версия Java (Рисунок 70), что подтвердило корректность установки.

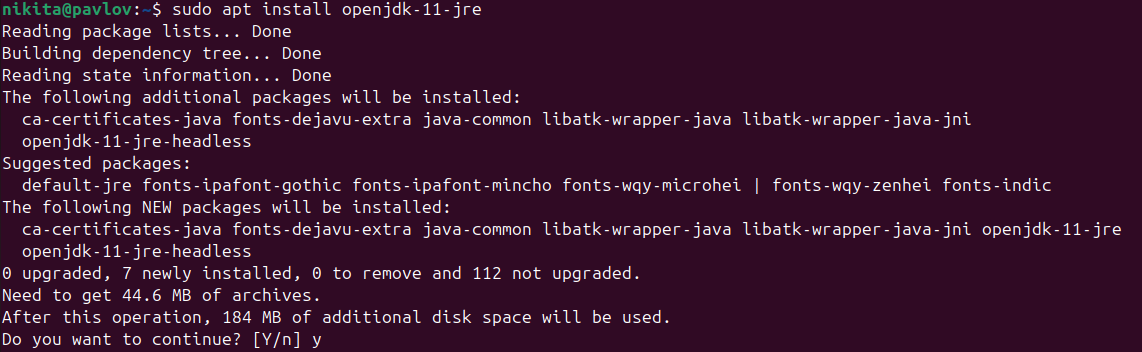


Рисунок 69 – Установка Java

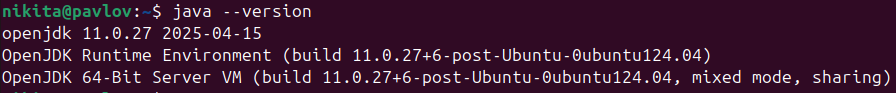


Рисунок 70 – Проверка версии Java

Сервер Jenkins успешно запущен (Рисунок 71).

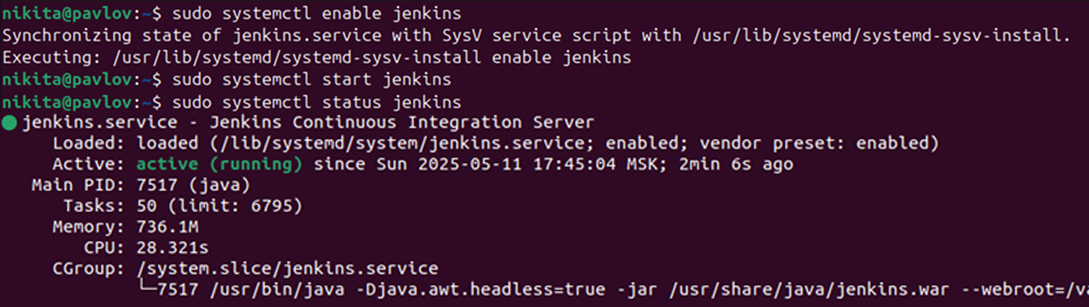


Рисунок 71 – Запуск Jenkins

Проведена первоначальная конфигурация (Рисунок 72).

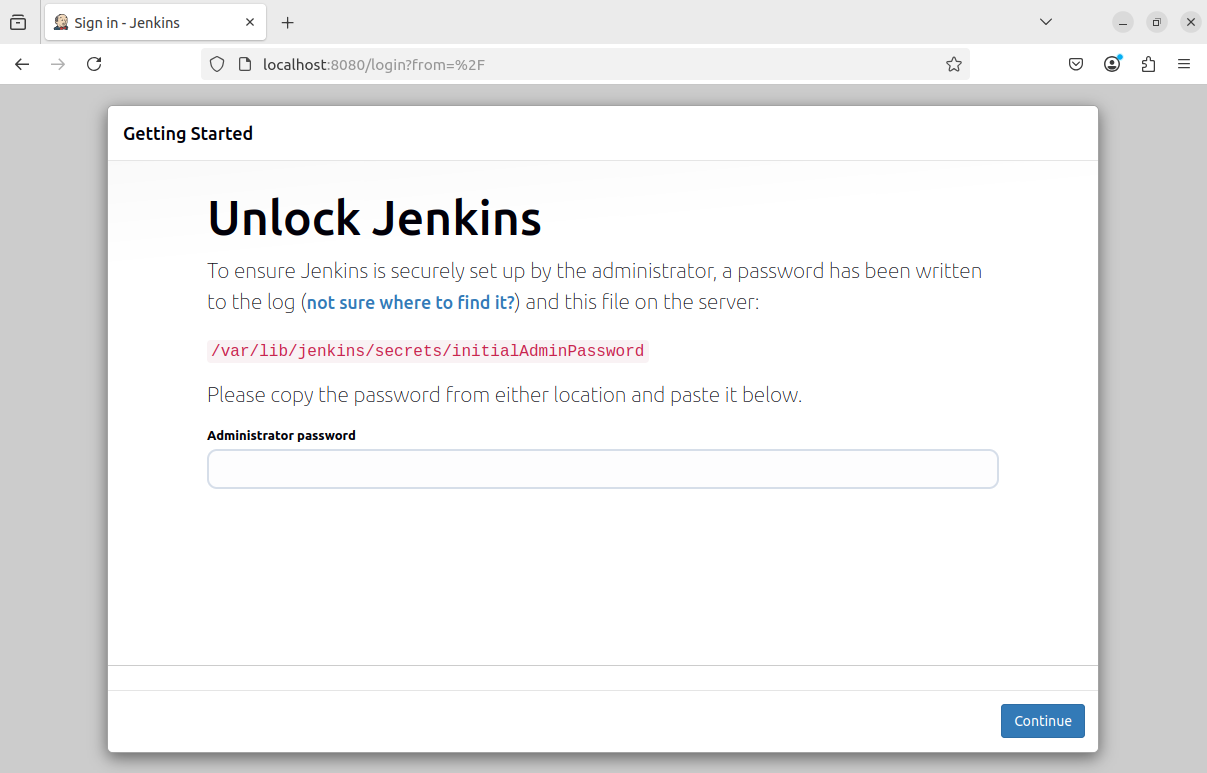


Рисунок 72 – Страница разблокировки Jenkins

Установлен базовый набор плагинов (Рисунки 73-74).

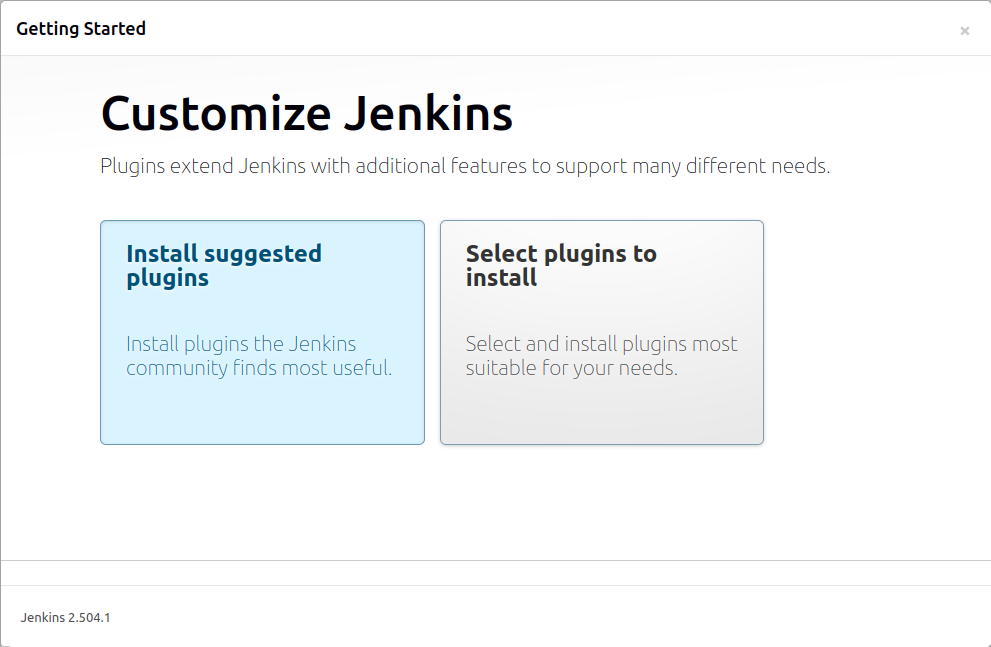


Рисунок 73 – Плагины для Jenkins

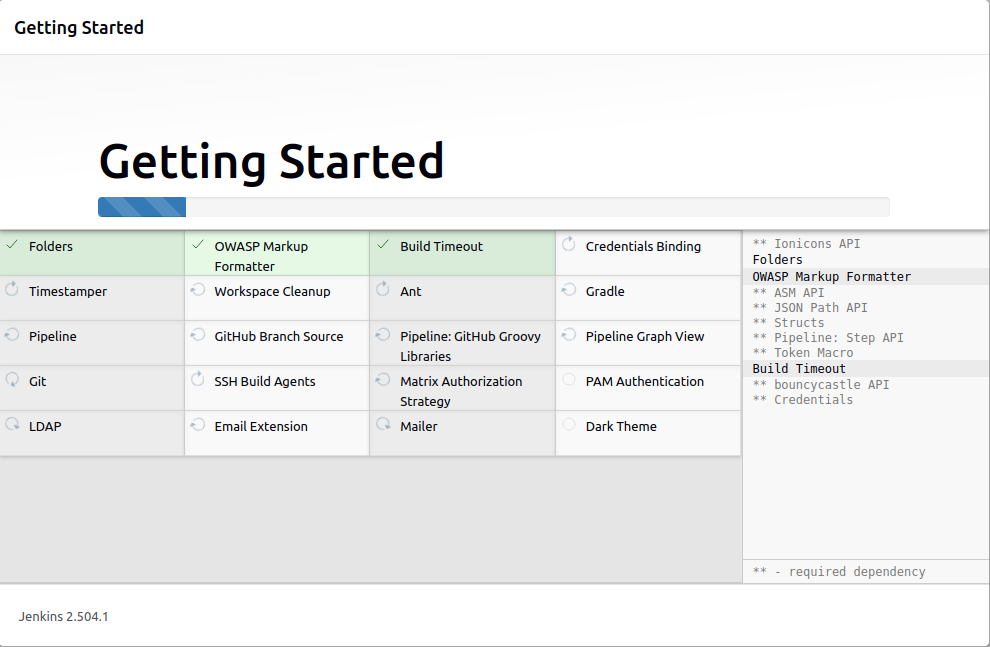


Рисунок 74 – Установка плагинов Jenkins

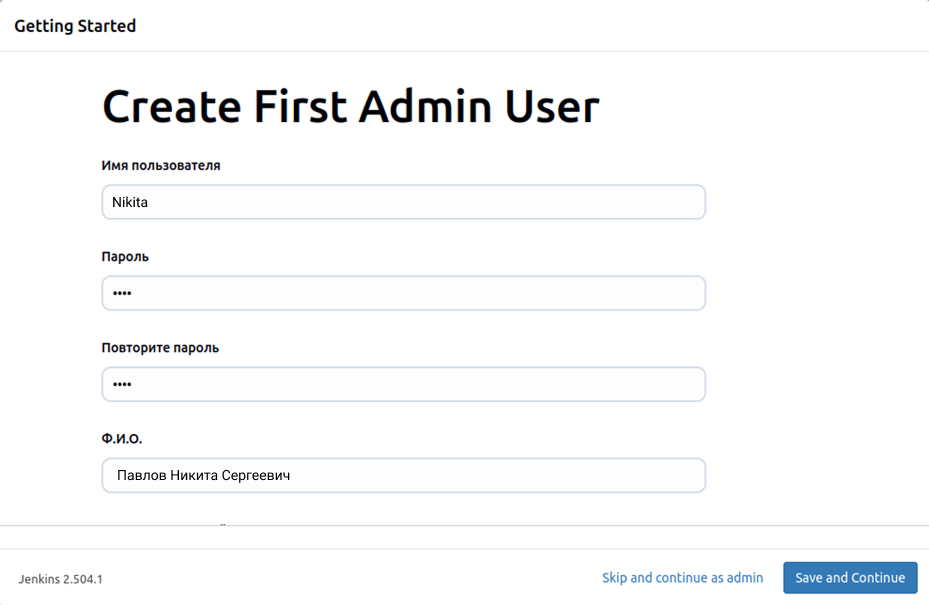


Рисунок 75 – Создание первого пользователя администратора

## 2.8 Модуль 15. Практические навыки работы с Docker

Разработан Dockerfile (Рисунок 76), содержащий:

* Базовый образ на основе ubuntu
* Установку PostgreSQL и зависимостей
* Настройку конфигурации СУБД
* Определение точек монтирования и портов

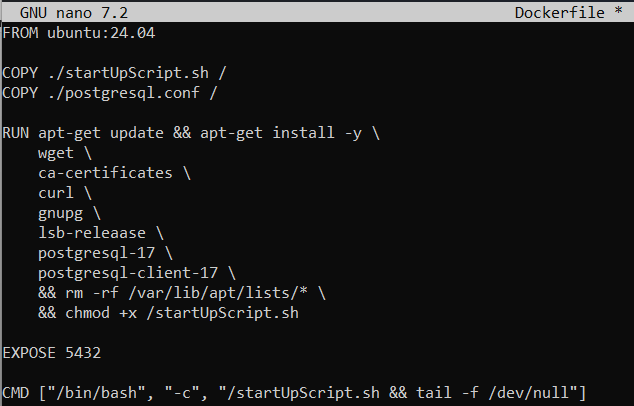


Рисунок 76 – Содержимое Dockerfile

Подготовлен конфигурационный файл PostgreSQL (Рисунок 77) для корректной работы в контейнере.

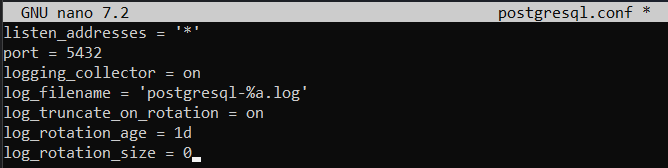


Рисунок 77 – Конфигурация PostgreSQL

Создан стартовый скрипт (Рисунок 78) для автоматической инициализации БД при запуске.

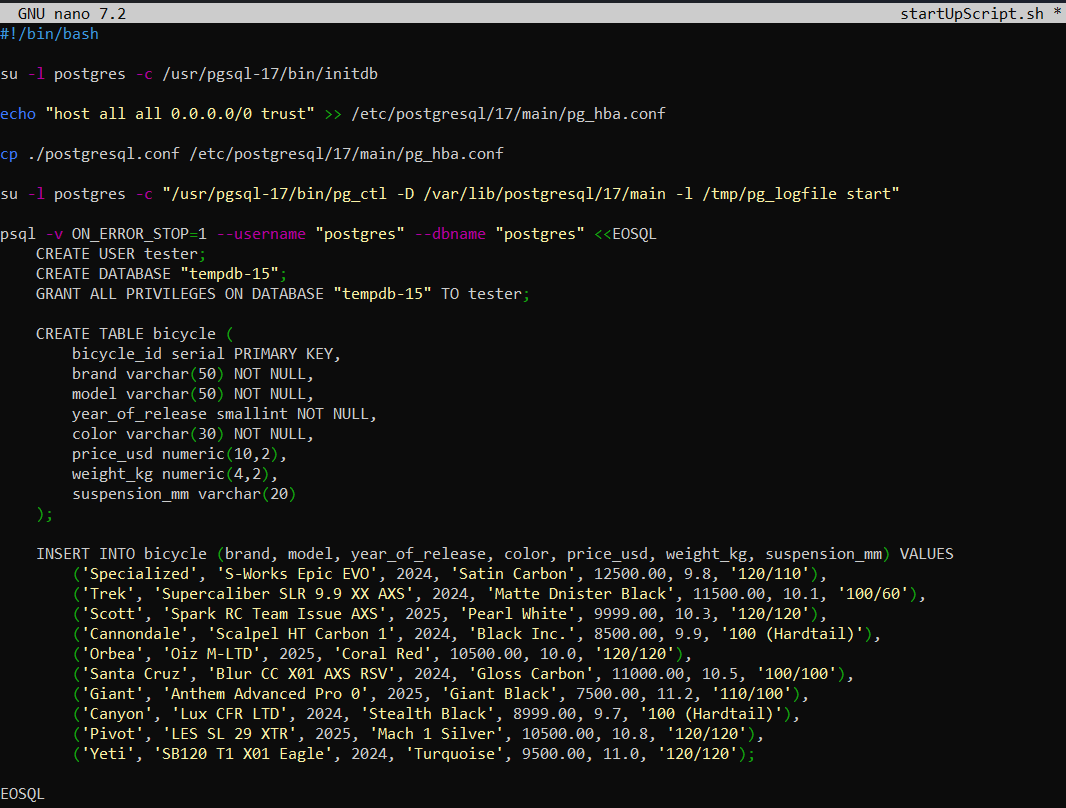


Рисунок 78 – Стартовый скрипт для запуска

Выполнена сборка Docker-образа (Рисунок 79). Запущен контейнер (Рисунок 80) с пробросом портов и подключением volumes.

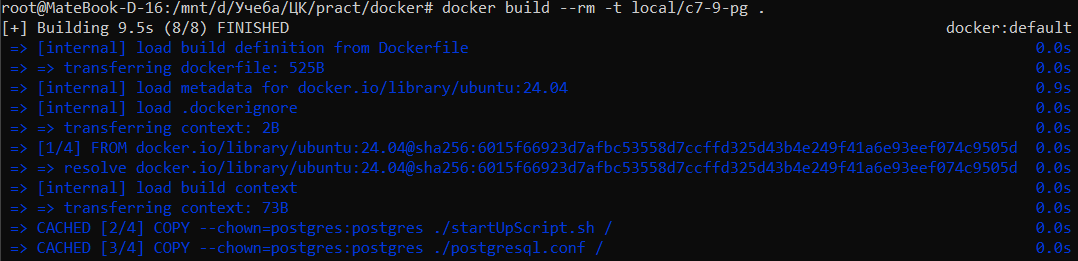


Рисунок 79 – Сборка Docker-образа



Рисунок 80 – Запуск контейнера

# 3 ЗАКЛЮЧЕНИЕ

В ходе выполнения итоговой аттестационной работы по программе «Технологии Девопс» были успешно выполнены следующие практические задания:

1. **Модуль 6. Администрирование баз данных**

* Установка и настройка PostgreSQL.
* Создание пользователей и управление правами доступа.
* Создание базы данных и таблиц.
* Выполнение SQL-запросов (SELECT, INSERT, UPDATE).

1. **Модуль 7. Масштабируемость и отказоустойчивость**

* Настройка синхронизации времени между узлами.
* Генерация SSH-ключей и настройка безопасного доступа.
* Развертывание кластера PostgreSQL с использованием Pacemaker и Corosync.
* Проверка отказоустойчивости кластера.

1. **Модуль 8. Мониторинг, логирование и оповещение событий**

* Установка и настройка Node Exporter для сбора метрик.
* Развертывание Prometheus и настройка сбора данных.
* Установка Grafana, подключение источников данных и импорт дашбордов.
* Визуализация метрик производительности системы.

1. **Модуль 9. Виртуализация в DevOps**

* Развертывание ВМ на базе Ubuntu 24.04
* Установка и настройка дополнительных утилит (vim, mc, net-tools).
* Проверка работоспособности сетевых интерфейсов.

1. **Модуль 12. Конфигурационное управление (IaC)**

* Установка Ansible и настройка инвентарных файлов.
* Создание и выполнение playbook'ов для автоматизации задач.
* Проверка доступности узлов и управление конфигурациями.

1. **Модуль 13. Системы контроля версий (Git)**

* Установка и настройка Git.
* Создание локального репозитория, добавление файлов и коммиты.
* Работа с ветками: создание, слияние, отправка изменений.
* Организация распределённой работы с Git.

1. **Модуль 14. Жизненный цикл ПО**

* Установка и настройка Jenkins.
* Настройка плагинов и создание администратора.
* Интеграция с инструментами CI/CD.

1. **Модуль 15. Практические навыки работы с Docker**

* Создание Dockerfile для развертывания PostgreSQL.
* Сборка Docker-образа и запуск контейнера.
* Настройка конфигурации и проверка работоспособности.

В результате выполнения всех модулей были освоены ключевые технологии DevOps, включая управление базами данных, обеспечение отказоустойчивости, мониторинг, автоматизацию развертывания и работу с контейнеризацией. Практические задания продемонстрировали готовность к решению реальных задач в области DevOps, включая настройку CI/CD, управление инфраструктурой и обеспечение стабильности ИТ-систем.

# СПИСОК ЛИТЕРАТУРЫ

1. Официальная документация PostgreSQL – [Электронный ресурс]. – URL: <https://www.postgresql.org/docs> (дата обращения: 01.05.2025).
2. Pacemaker и Corosync: High Availability Cluster Management – [Электронный ресурс]. – URL: <https://clusterlabs.org> (дата обращения: 08.05.2025).
3. Prometheus: документация – [Электронный ресурс]. – URL: <https://prometheus.io/docs> (дата обращения: 12.05.2025).
4. Grafana Labs: руководство пользователя – [Электронный ресурс]. – URL: <https://grafana.com/docs> (дата обращения: 12.05.2025).
5. Ansible: официальная документация – [Электронный ресурс]. – URL: <https://docs.ansible.com> (дата обращения: 15.05.2025).
6. Git: документация – [Электронный ресурс]. – URL: <https://git-scm.com/doc> (дата обращения: 17.05.2025).
7. Jenkins: User Handbook – [Электронный ресурс]. – URL: <https://www.jenkins.io/doc> (дата обращения: 25.05.2025).
8. Docker: официальная документация – [Электронный ресурс]. – URL: <https://docs.docker.com> (дата обращения: 28.05.2025).
9. DevOps: принципы, практики и инструменты – Ким Д., Хамбл Д., Дебуа П., Уиллис Д. – М.: Альпина Паблишер, 2024. – 480 с.
10. Автоматизация DevOps с помощью Ansible – Лорх Д. – СПб.: Питер, 2023. – 320 с.
11. Kubernetes и Docker: оркестрация контейнеров – Хайнеман Р. – М.: ДМК Пресс, 2025. – 412 с.
12. ГОСТ Р 7.0.100-2018. Библиографическая запись. Библиографическое описание. Общие требования и правила составления – М.: Стандартинформ, 2018. – 124 с.