|  |
| --- |
| 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

«30» мая 2025 г.

**ИТОГОВАЯ АТТЕСТАЦИОННАЯ РАБОТА**

по дополнительной программы профессиональной переподготовки

«Технологии Девопс»

На тему: **«**Набор тематических заданий по проведению итоговой аттестации DevOps-инженера – вариант 287**»**

Обучающийся *Враженко Даниил Олегович*

*Подпись Фамилия, имя, отчество*

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

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

Москва 2025 г.

СОДЕРЖАНИЕ

[Введение 3](#__RefHeading___Toc52862_883944595)

[Модуль 6. Администрирование баз данных 4](#__RefHeading___Toc52864_883944595)

[Модуль 8. Мониторинг, логирование и оповещение событий 8](#__RefHeading___Toc1267_3933951894)

[Модуль 9. Виртуализация в DevOps 14](#__RefHeading___Toc1719_1609335944)

[Модуль 13. Системы контроля версий. Распределённая система управления версиями Git 16](#__RefHeading___Toc4772_492110456)

[Модуль 14. Жизненный цикл ПО 19](#__RefHeading___Toc1723_1609335944)

[Заключение 27](#__RefHeading___Toc52866_883944595)

[Список литературы 29](#__RefHeading___Toc52868_883944595)

# ****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, 8, 9, 13 и 14 дополнительной профессиональной программы «Технологии Девопс» были реализованы следующие аспекты:

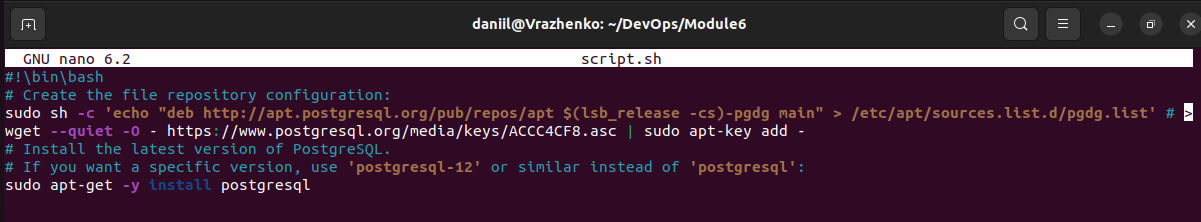
* Администрирование PostgreSQL: от установки СУБД до проектирования структуры данных и управления пользователями;
* Настройка полноценного стека мониторинга на базе Prometheus, Grafana и Node Exporter;
* Работа с системами контроля версий: развёртывание Git-сервера, branching-стратегии и управление репозиториями;
* Автоматизация жизненного цикла ПО через настройку Jenkins-сервера;
* Освоение инструментов виртуализации и управления инфраструктурой.

Особое внимание уделено интеграции компонентов в единую экосистему: от версионного контроля 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.

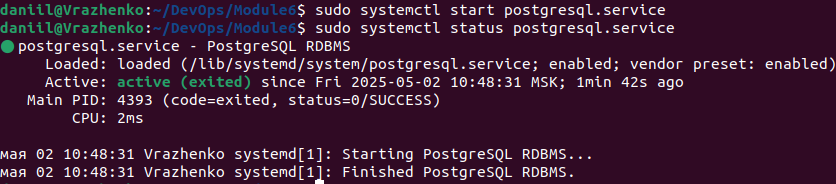
Цель работы — формирование профессиональных компетенций в области DevOps через практическое взаимодействие с ключевыми технологиями: управление базами данных, настройка систем мониторинга, организация CI/CD-процессов и работа с распределёнными системами контроля версий. Результаты демонстрируют готовность к реализации реальных задач в области сопровождения и развития ИТ-инфраструктуры.

# Модуль 6. Администрирование баз данных

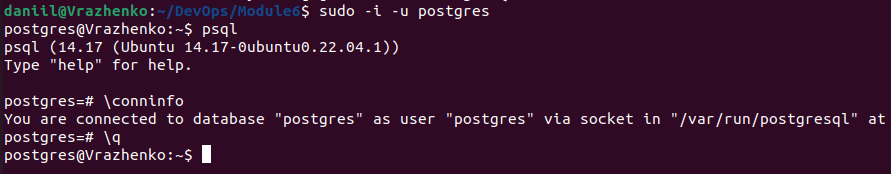
Установим postgresql 12 на ubuntu (рис. 1):

Рисунок 1 - Установка PostgreSQL

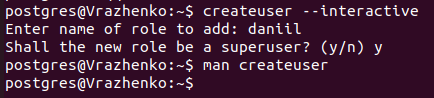
Проверим работоспособность базы данных (рис. 2):

Рисунок 2 - Проверка работоспособности

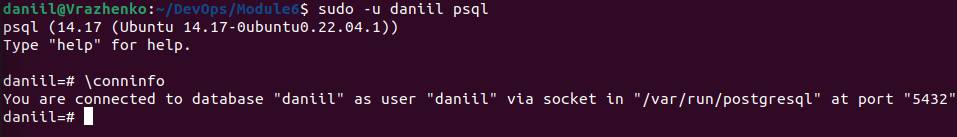
Войдём в potgres (рис. 3):

Рисунок 3 - Вход в postgres

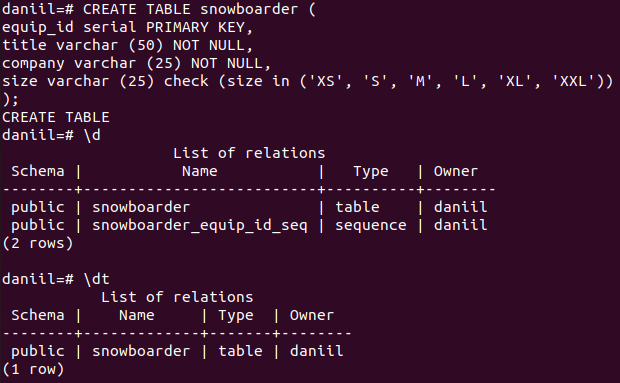
Создадим пользователя (рис. 4):

Рисунок 4 - Создание пользователя

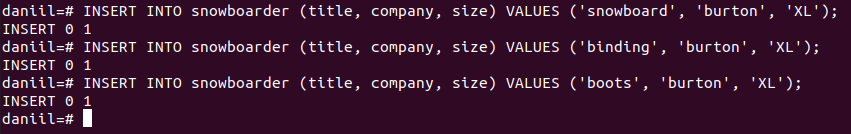
Создадим базу данных (рис. 5):

Рисунок 5 - Создание базы данных

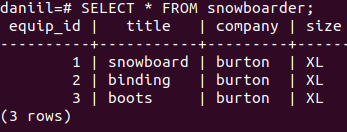
Создадим и просмотрим таблицу в базе данных (рис. 6):

Рисунок 6 - Создание и просмотр базы данных

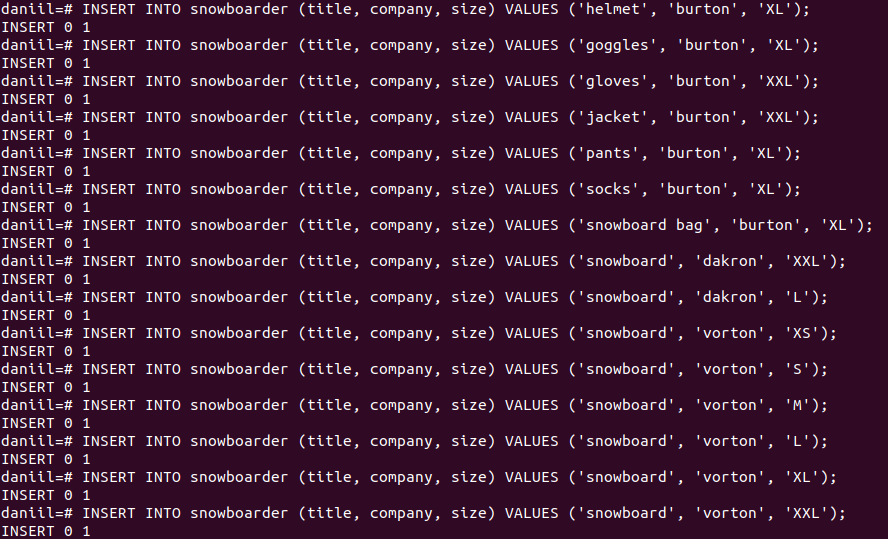
Заполним таблицу данными (рис. 7):

Рисунок 7 - Заполнение таблицы данными

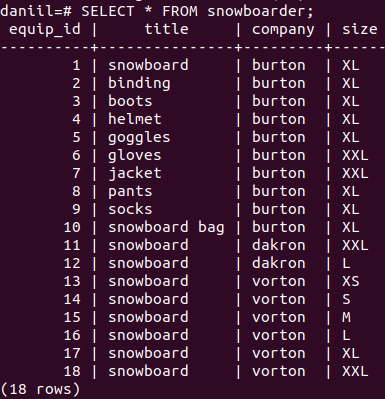
Проверим таблицу (рис. 8):

Рисунок 8 - Проверка таблицы

Дополним таблицу новыми данными (рис. 9):

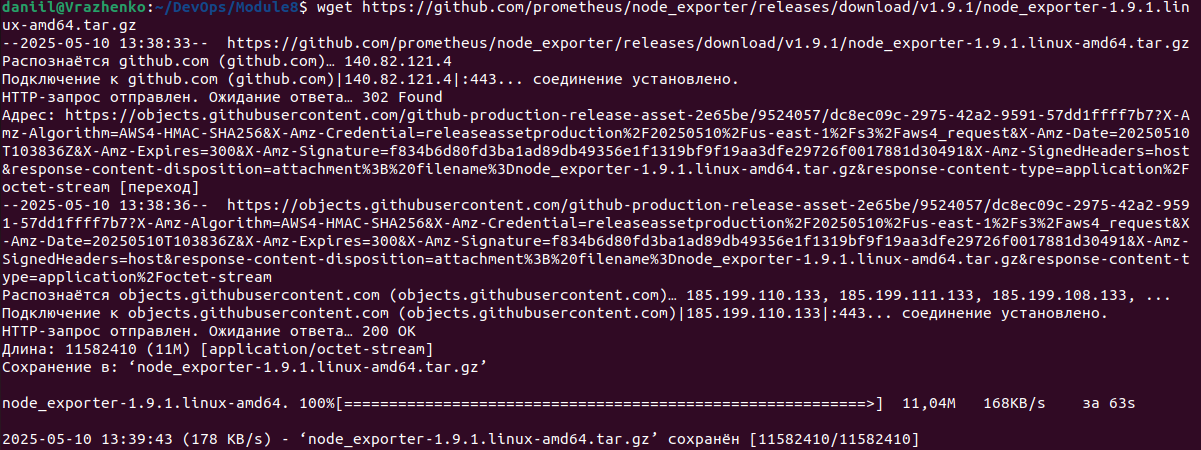
Рисунок 9 - Дополнение таблицы данными

Снова проверим таблицу (рис. 10):

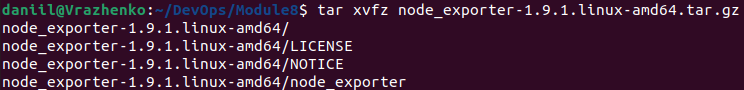
Рисунок 10 - Повторная проверка таблицы

# Модуль 8. Мониторинг, логирование и оповещение событий

Загрузим сжатый пакет node\_exporter (рис. 11):

Рисунок 11 - Загрузка node\_exporter

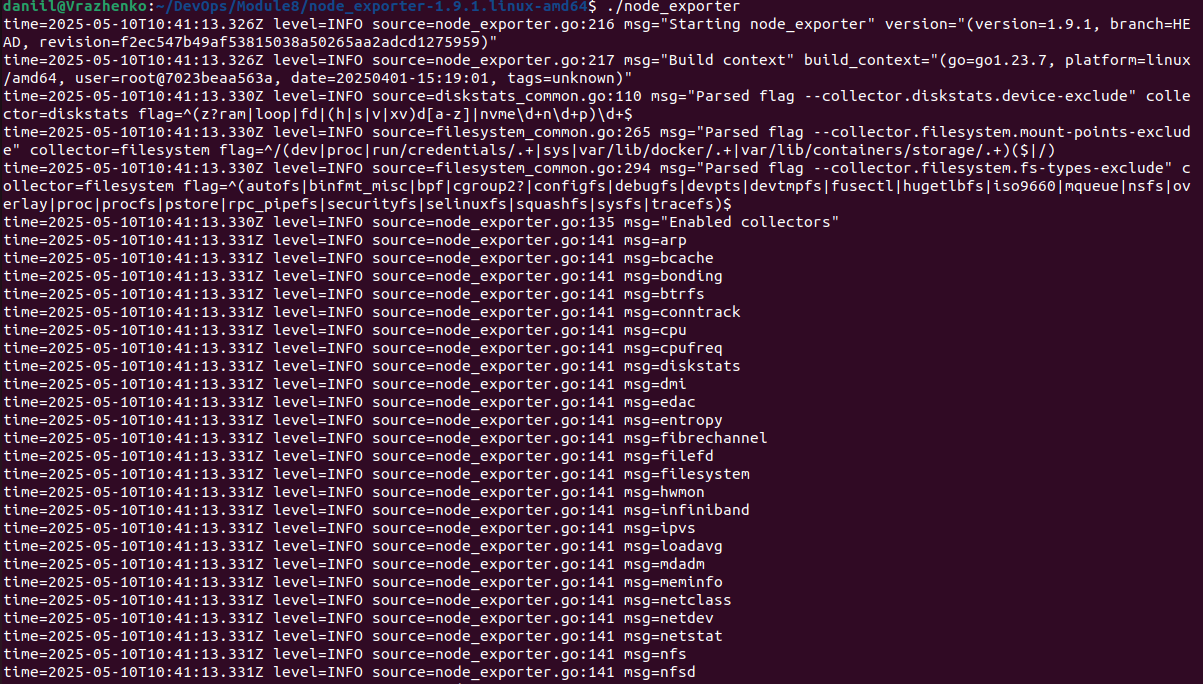
Извлечем двоичный файл node\_exporter (рис. 12):

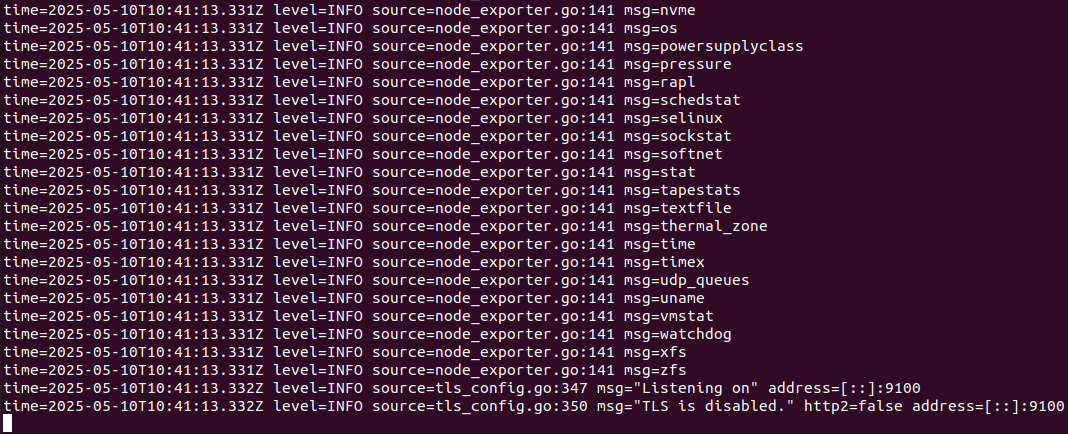
Рисунок 12 - Извлечение двоичного файла node\_exporter

Перейдем в каталог и сделаем двоичный файл исполняемым (рис. 13):

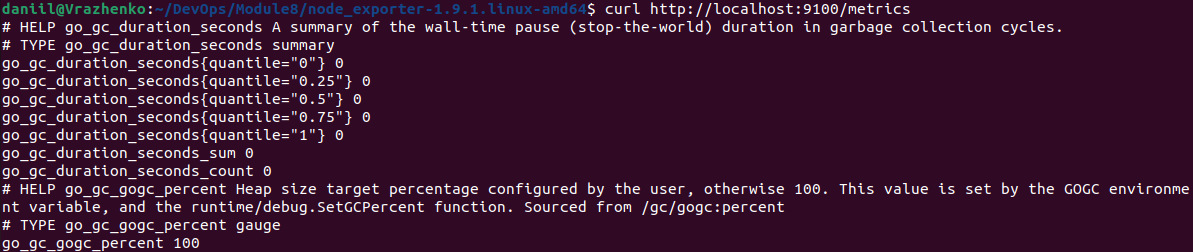
Рисунок 13 - Переход в каталог и изменение прав доступа

Запустим node\_exporter и увидим вывод, указывающий на то, что Node Exporter теперь запущен на порту 9100 (рис. 14-15)

Рисунок 14 - Первая часть вывода

Рисунок 15 - Вторая часть вывода

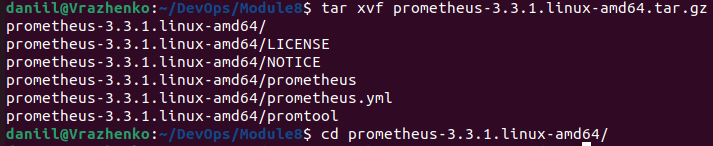
Проверим, что метрики экспортируются через порт 9100 (рис. 16):

Рисунок 16 - Наименьшая часть вывода

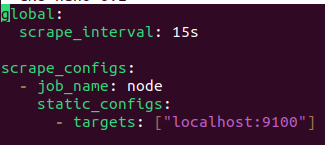
Загрузим архив пакета Prometheus (рис. 17):

Рисунок 17 - Загрузка пакета Prometheus

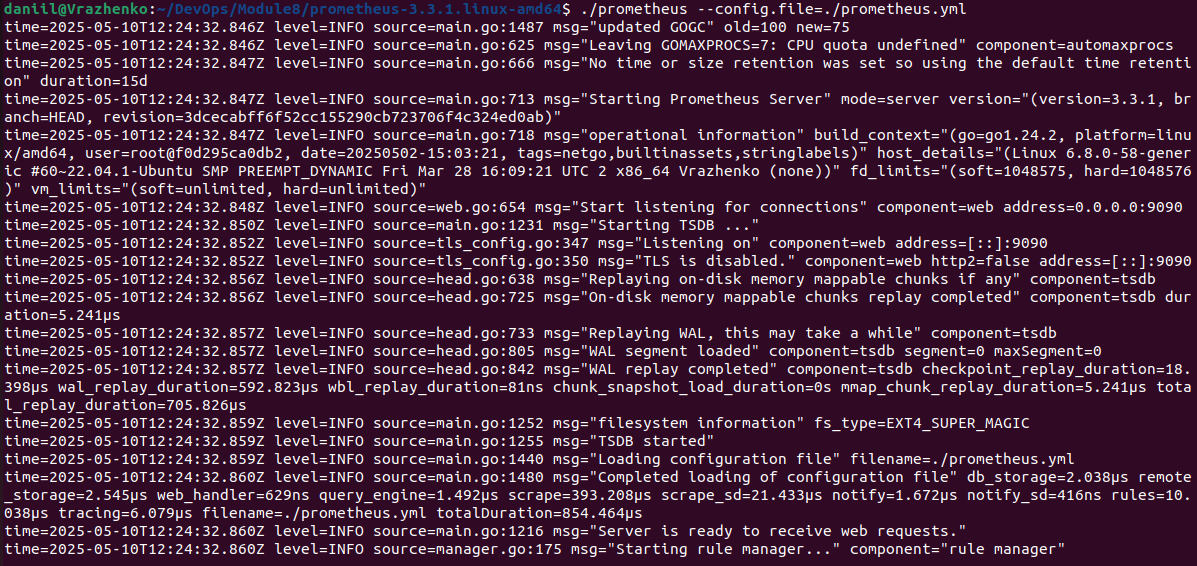
Извелечем двоичный файл и перейдем в каталог (рис. 18):

Рисунок 18 - Извлечение двоичного файла и переход в каталог

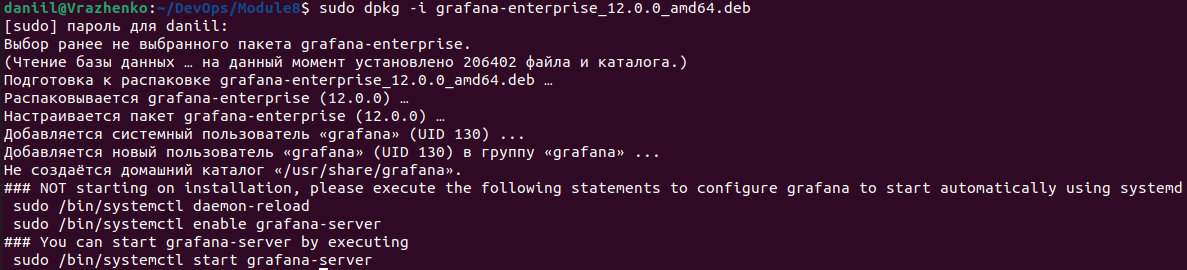
Настроим Prometheus с помощью изменения содержимого файла prometheus.yml (рис. 19):

Рисунок 19 - Содержимое prometheus.yml

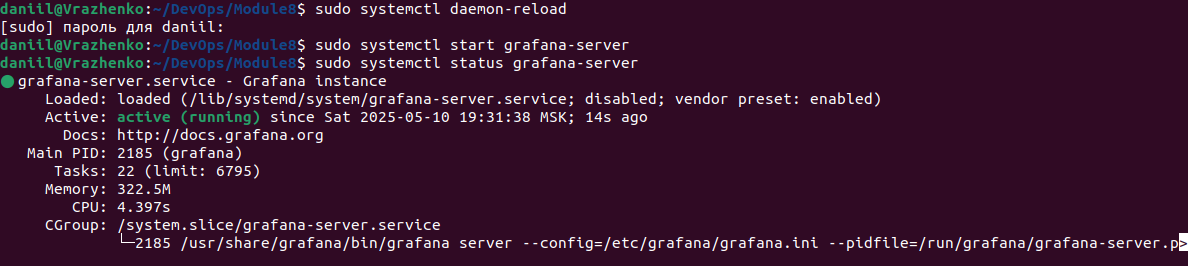
Запустим Prometheus (рис. 20):

Рисунок 20 - Запуск Prometheus

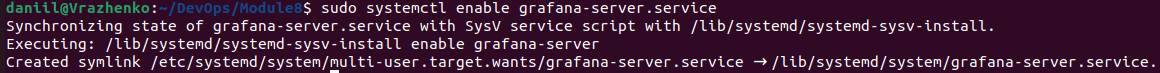
Выполним установку с помощью пакета .deb (рис. 21):

Рисунок 21 - Распаковка пакета .deb

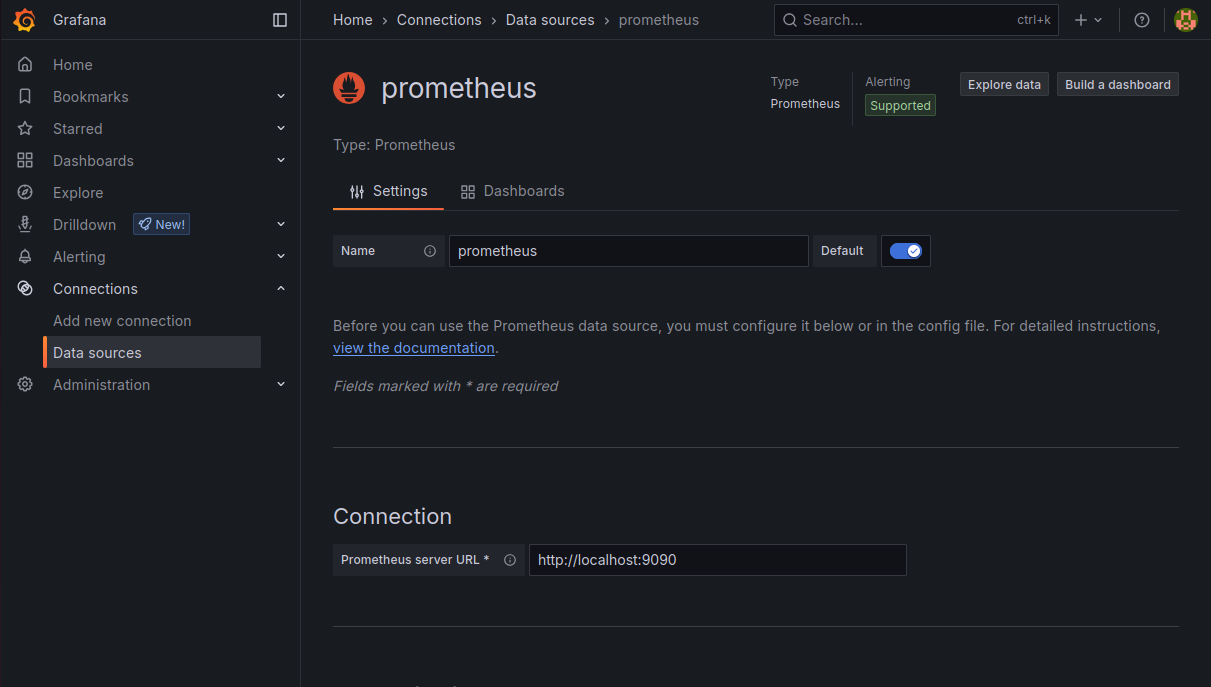
Запустим grafana-server (рис. 22):

Рисунок 22 - Запуск grafana-server

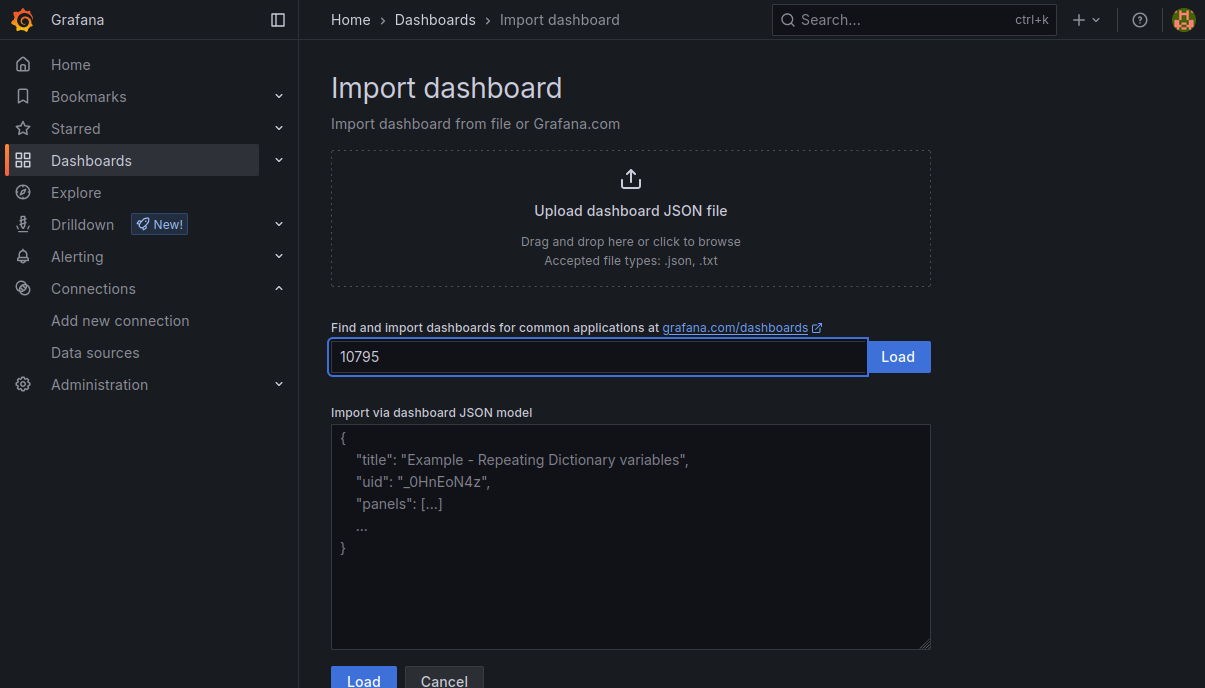
Настроим grafana-server для запуска при загрузке (рис. 23):

Рисунок 23 - Настройка grafana-server

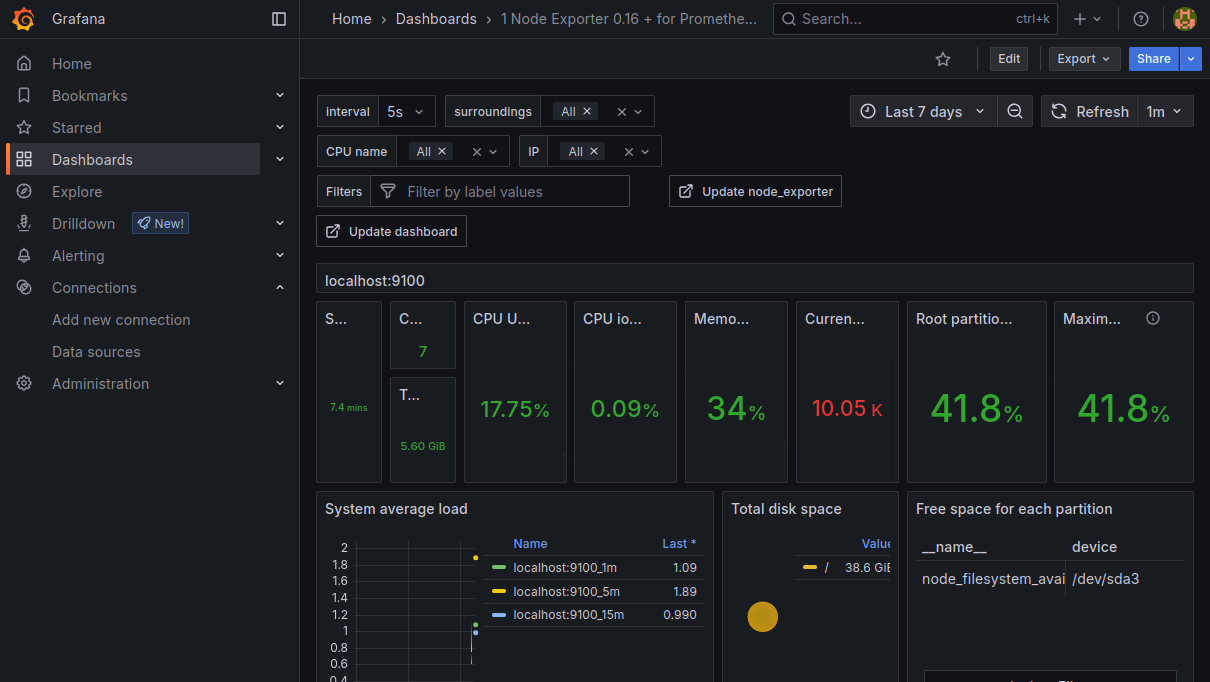
Настроим источники данных (рис. 24):

Рисунок 24 - Найстройки источников данных

Используем готовый дашборд (рис. 25):

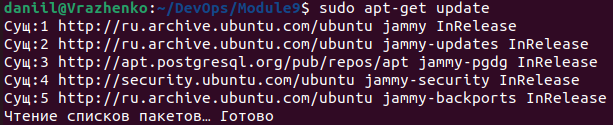
Рисунок 25 - Импорт готового дашборда

Откроем установленный дашборд (рис. 26):

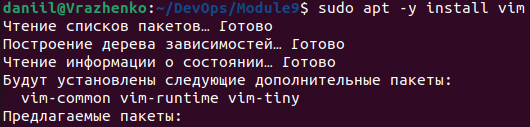
Рисунок 26 - Установленный дашборд

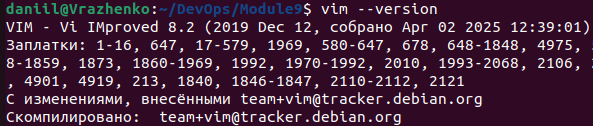
# Модуль 9. Виртуализация в DevOps

Обновим базу данных пакетов с помощью apt-get (рис. 27):

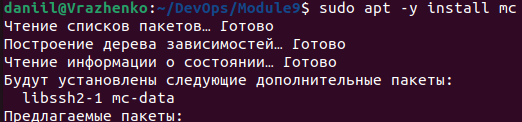
Рисунок 27 - Обновление базы данных пакетов

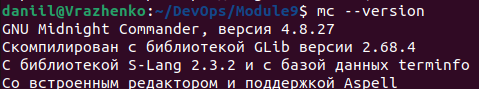
Установка vim и проверка версии (рис. 28-29):

Рисунок 28 - Установка vim

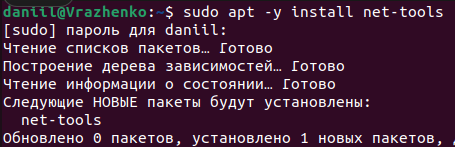
Рисунок 29 - Версия vim

Установка mc и проверка версии (рис. 30-31):

Рисунок 30 - Установка mc

Рисунок 31 - Версия mc

Установка net-tools (рис. 32):

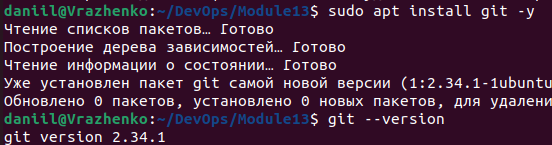
Рисунок 32 - Установка net-tools

К сожалению, проверить версию net-tools напрямую (как, у vim или mc) не получится, так как net-tools — это набор отдельных утилит, а не единая программа. Однако можно использовать флаг --version с некоторыми инструментами из этого пакета, например, ifconfig (рис. 33):

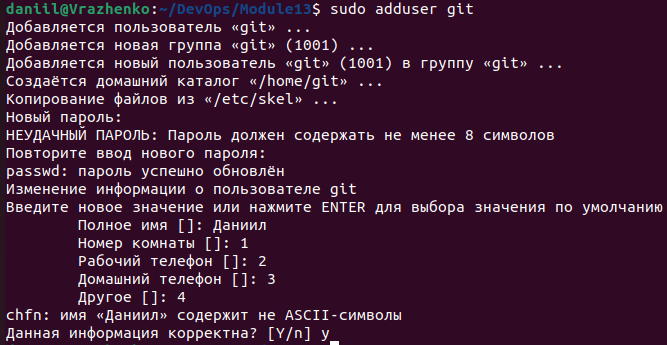
Рисунок 33 - Версия net-tools

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

Установим Git на сервер (рис. 34):

Рисунок 34 - Установка Git

Создадим пользователя git (рис. 35):

Рисунок 35 - Создание пользователя git

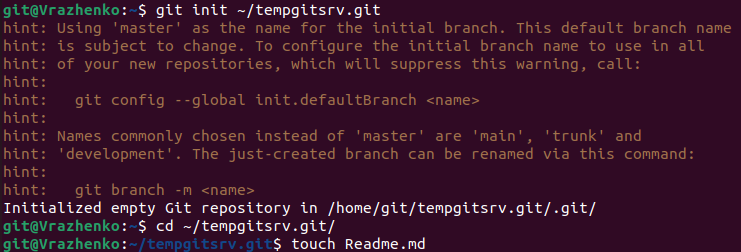
Логинимся под пользователя git (рис. 36):

Рисунок 36 - Логин под пользователя git

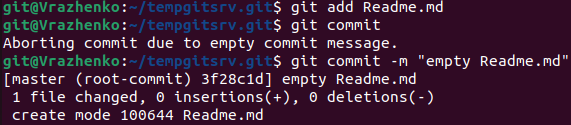
Предварительно настроим Git (рис. 37):

Рисунок 37 - Предварительные настройки Git

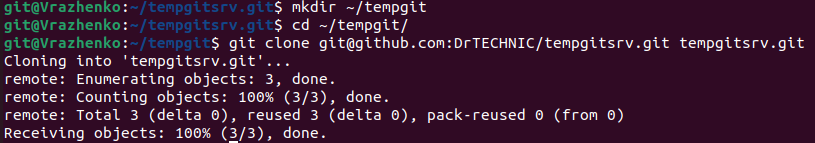
Инициализируем новый репозиторий и добавим в него пустой файл (рис. 38):

Рисунок 38 - Инициализация репозитория и добавление файла

Закоммитим изменение (рис. 39):

Рисунок 39 - Коммит изменения

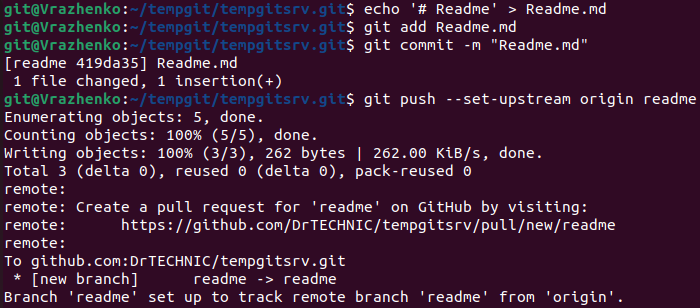
Склонируем Git локально (рис. 40):

Рисунок 40 - Клонирование Git локально

Создадим новую ветвь (рис. 41):

Рисунок 41 - Создание новой ветви

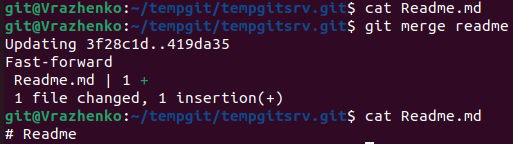
Создадим новый файл и отправим изменения в удаленный репозиторий (рис. 42):

Рисунок 42 - Отправвление изменений в репозиторий

Выведим список всех веток на сервере (рис. 43):

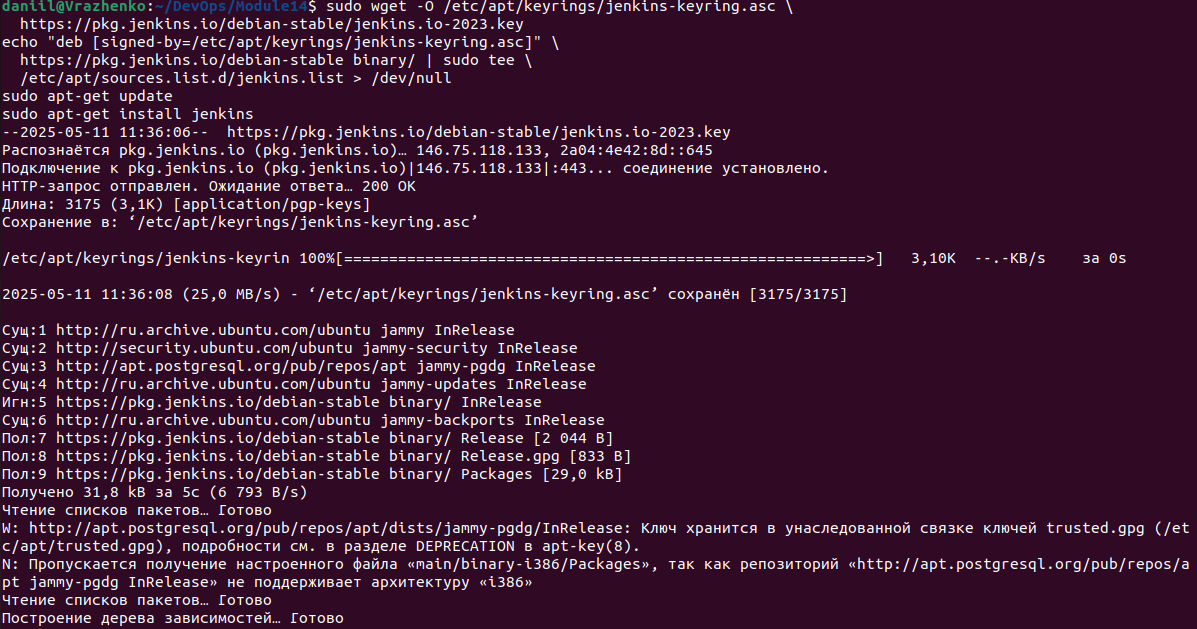
Рисунок 43 - Список всех веток

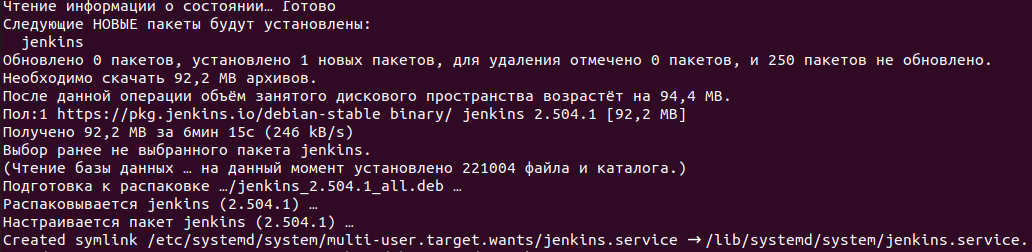
Сольем ветки (рис. 44):

Рисунок 44 - Слияние веток

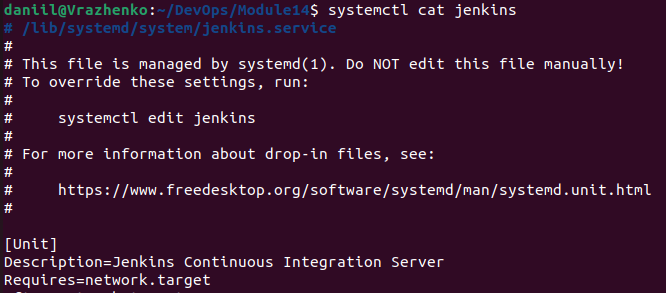
# Модуль 14. Жизненный цикл ПО

Установим Jenkins с помощью официальной инструкции (рис. 45-46):

Рисунок 45 - Первая часть установки

Рисунок 46 - Вторая часть установки

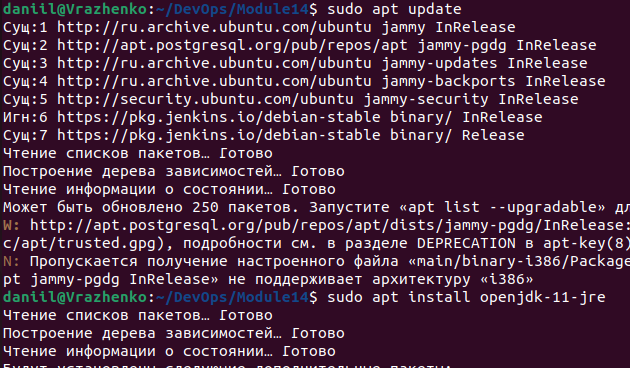
Настроим Jenkins как демон (рис. 47):

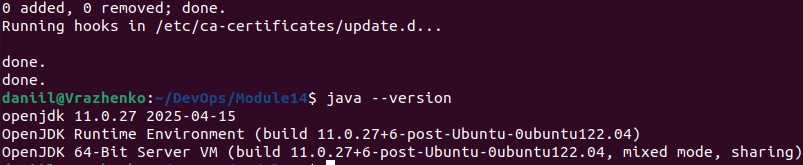
Рисунок 47 - Просмотр подробностей

Создадим пользователя для запуска службы (рис. 48):

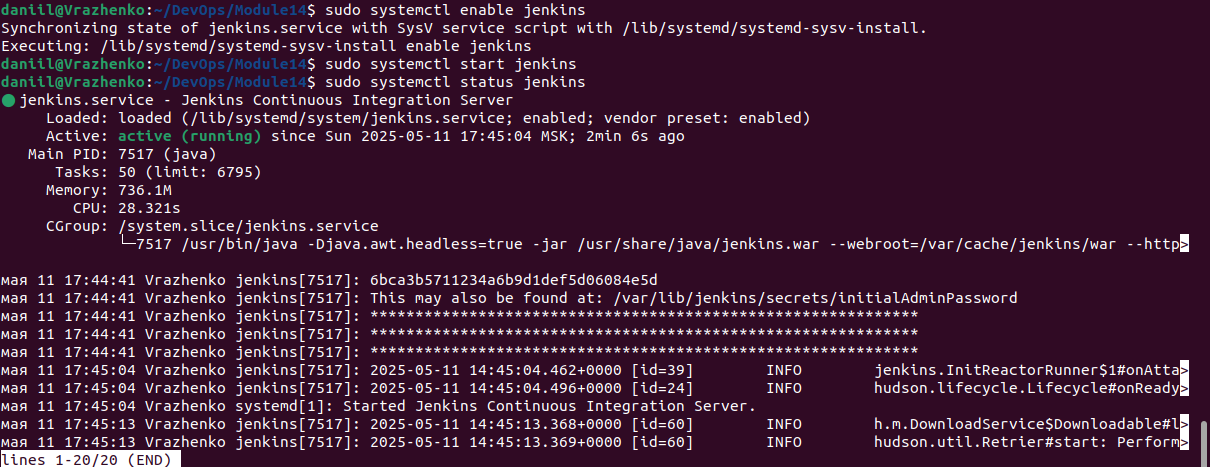
Рисунок 48 - id пользователя

Установим Java (рис. 49-50):

Рисунок 49 - Первая часть установки

Рисунок 50 - Вторая часть установки

Запустим Jenkins (рис. 51):

Рисунок 51 - Запуск Jenkins и его статус

Разблокируем Jenkins (рис. 52):

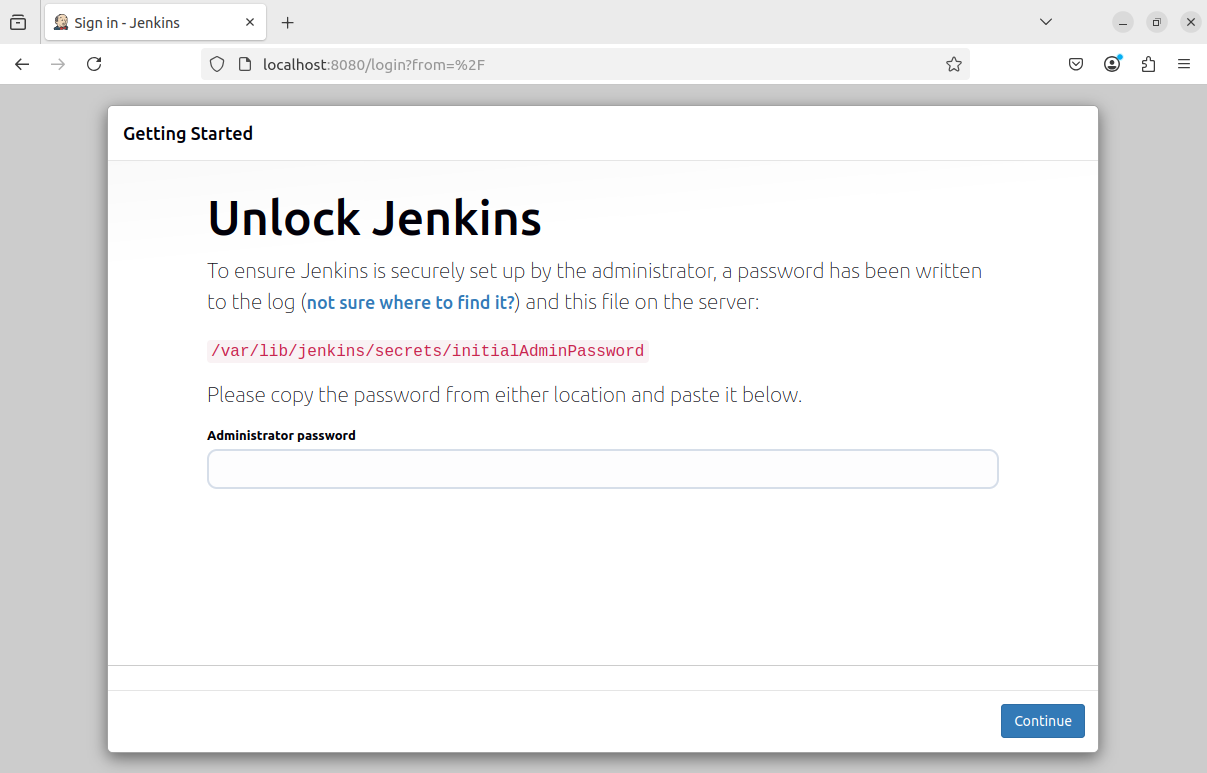


Рисунок 52 - Разблокировка Jenkins

Получим пароль для разблокировки Jenkins (рис. 53):

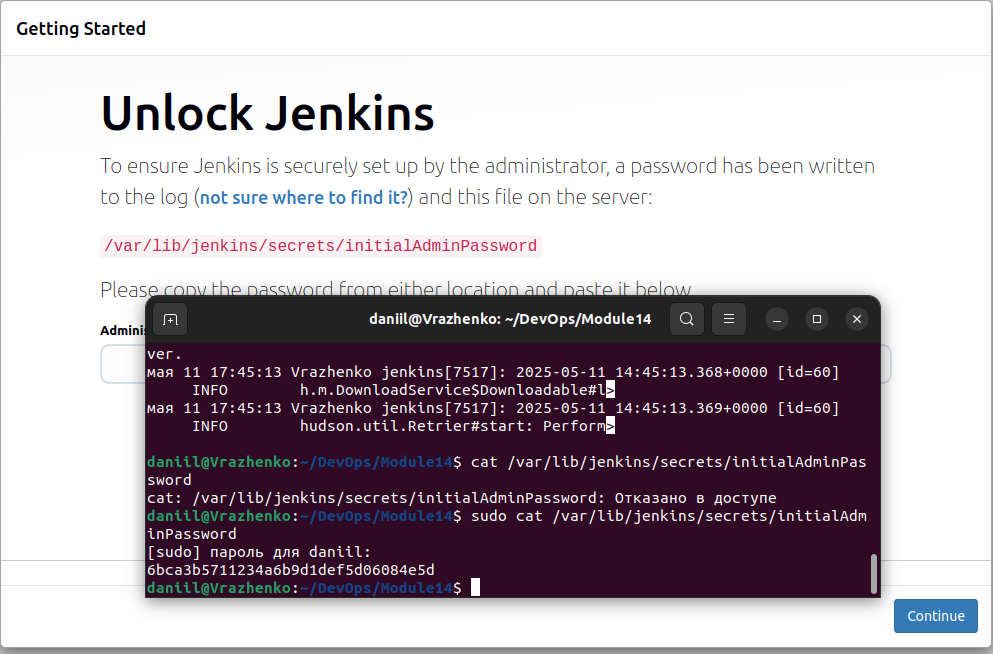
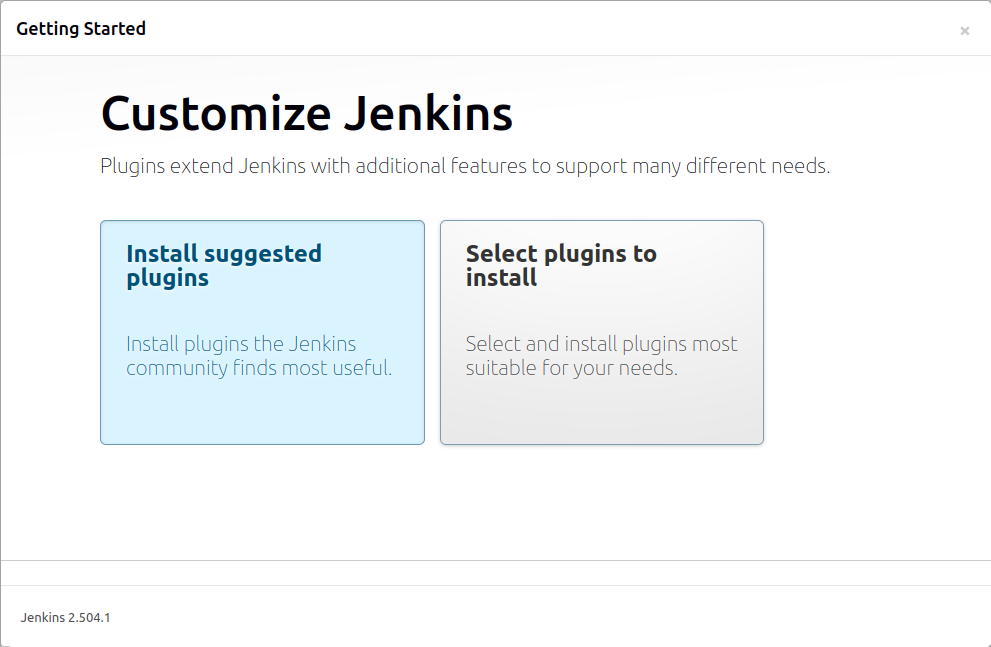
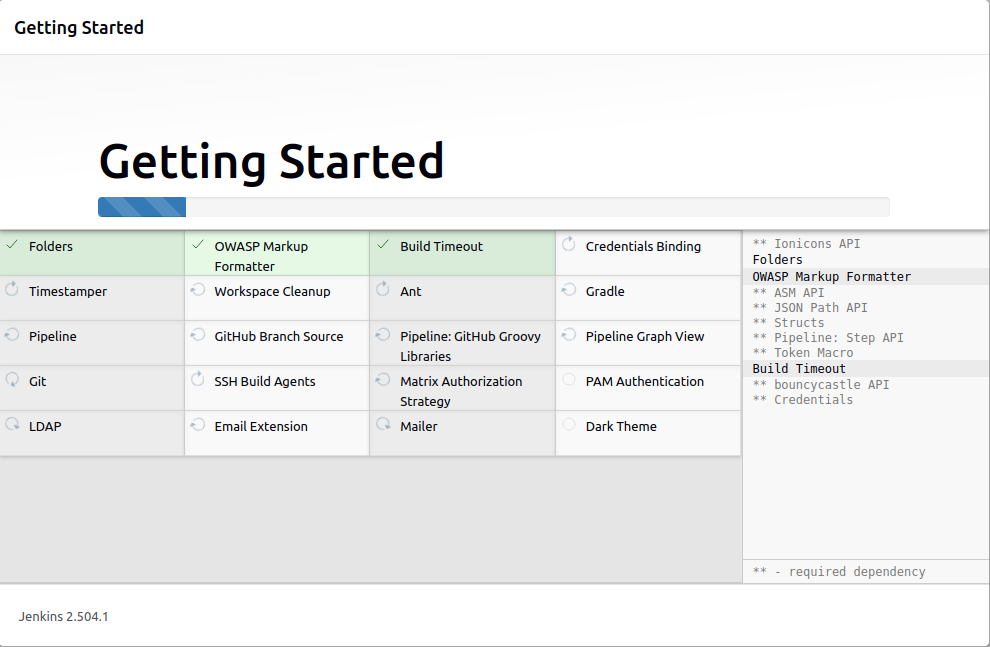


Рисунок 53 - Пароль для Jenkins

Установим плагины для Jenkins (рис. 54-55):

Рисунок 54 - Плагины для Jenkins

Рисунок 55 - Установка плагинов Jenkins

Создадим первого пользователя-администратора (рис. 56):

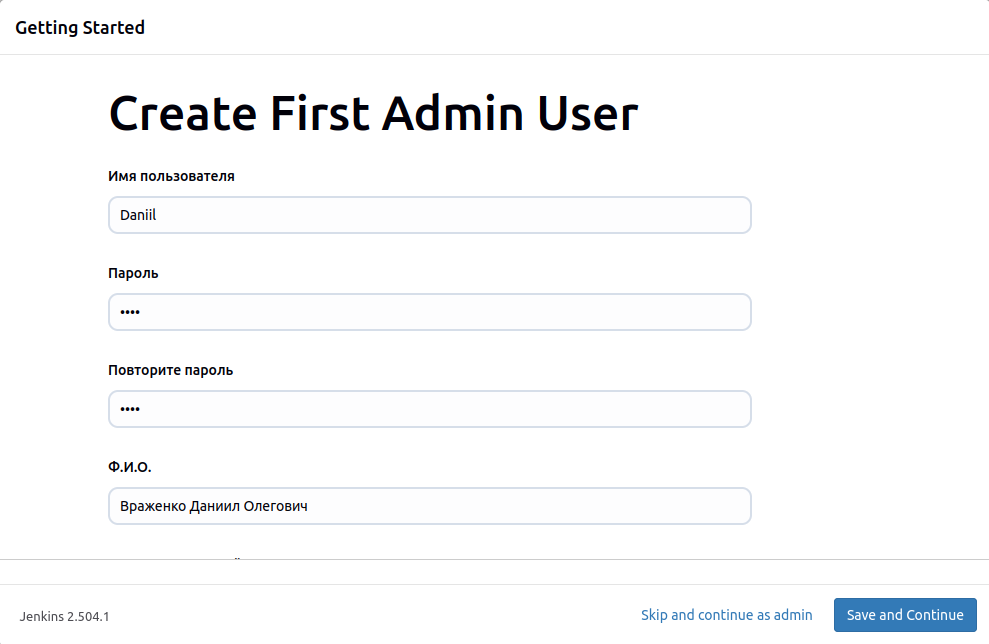
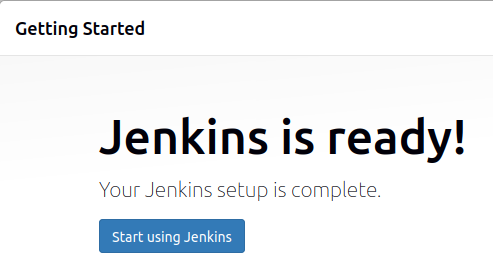
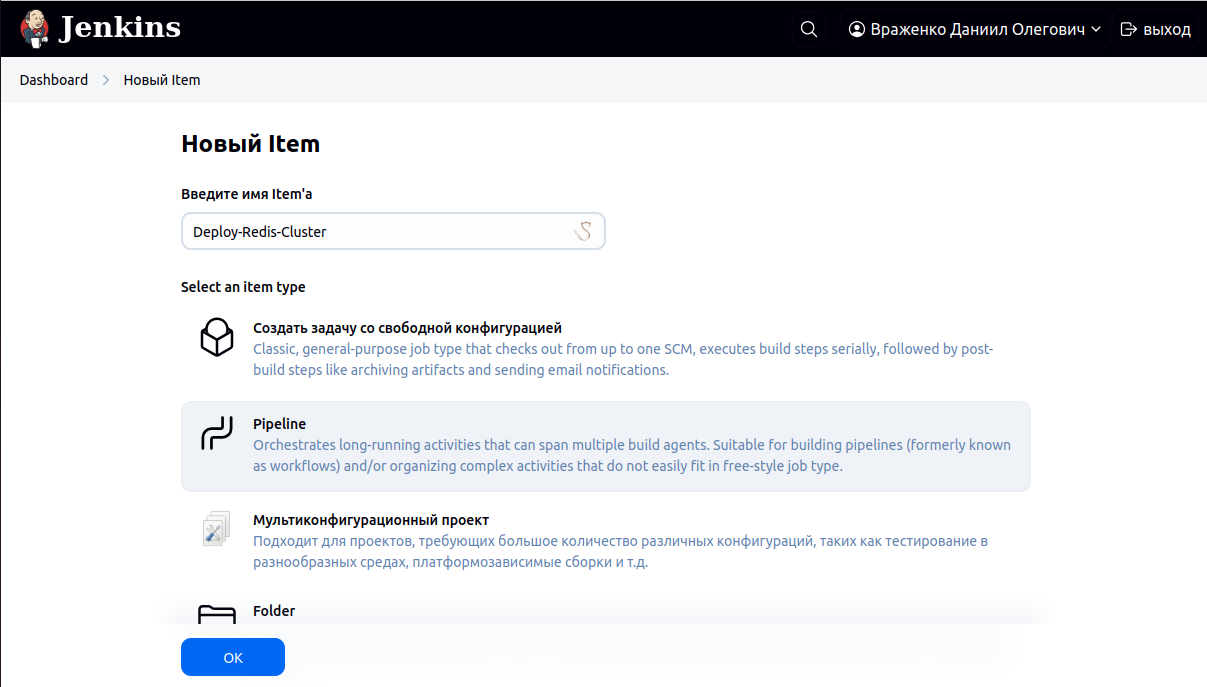


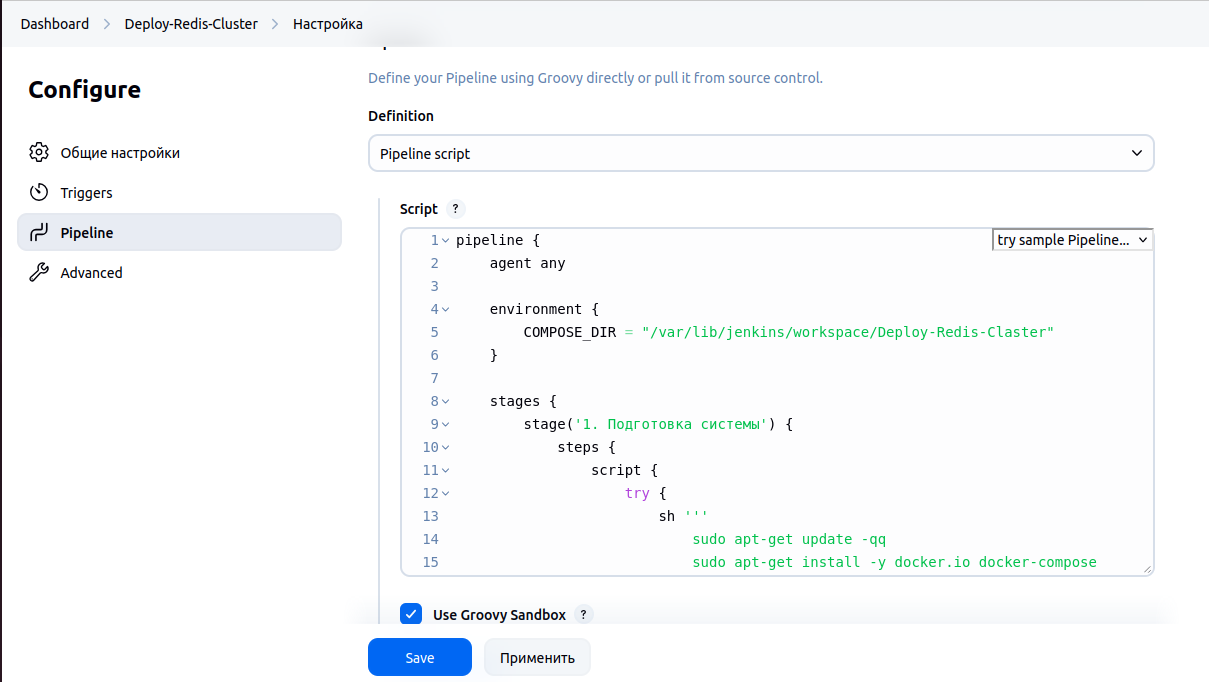
Рисунок 56 - Создание первого пользователя-администратора

Видим, что Jenkins готов к использованию (рис. 57):

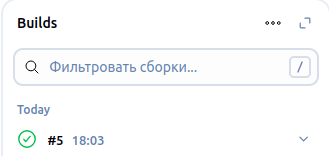
Рисунок 57 - Jenkins готов

Создадим Jenkins Pipeline и настроим его (рис. 58-59):

Рисунок 58 - Создание Pipeline

Рисунок 59 - Небольшая часть Pipeline скрипта

После нескольких попыток и изменений скрипта получаем успешный билд (рис. 60):

Рисунок 60 - Успешный билд

# ****Заключение****

Мы проделали работу по следующим модулям:

**Модуль 6. Администрирование баз данных:**

* Установили PostgreSQL 12 на Ubuntu.
* Проверили работоспособность базы данных.
* Подключились к PostgreSQL через командную строку.
* Создали пользователя и базу данных.
* Сформировали таблицу, заполнили её данными и проверили корректность внесённых записей.
* Дополнили таблицу новыми данными и выполнили повторную проверку.

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

* Загрузили и распаковали Node Exporter, сделали его исполняемым и запустили на порту 9100.
* Убедились, что метрики успешно экспортируются.
* Установили Prometheus, настроили его конфигурационный файл prometheus.yml и запустили сервер.
* Развернули Grafana с помощью пакета .deb, настроили автозапуск сервера.
* Добавили Prometheus в качестве источника данных в Grafana.
* Импортировали готовый дашборд и проверили отображение метрик.

**Модуль 9. Виртуализация в DevOps:**

* Обновили базу пакетов Ubuntu.
* Установили и проверили версии утилит:
  + Vim
  + Midnight Commander (mc)
  + Net-tools

**Модуль 13. Системы контроля версий. Git:**

* Установили Git на сервер.
* Создали и переключились на пользователя git.
* Настроили глобальные параметры Git (имя пользователя и email).
* Инициализировали новый репозиторий и добавили в него файл.
* Создали первый коммит.
* Склонировали репозиторий на локальную машину.
* Создали новую ветку, добавили файл и отправили изменения в удалённый репозиторий.
* Просмотрели список всех веток на сервере.
* Выполнили слияние веток.

**Модуль 14. Жизненный цикл ПО:**

* Установили Jenkins, следуя официальной инструкции.
* Настроили Jenkins как демона, создали системного пользователя для его работы.
* Установили Java.
* Запустили Jenkins.
* Установили рекомендуемые плагины.
* Создали административного пользователя и завершили первоначальную настройку.

# ****Список литературы****

1. Сельский, М. В. *PostgreSQL. Основы администрирования.* — СПб.: Питер, 2021. — 320 с.
2. Хайт, Дж. *DevOps: практика и теория.* — М.: Эксмо, 2022. — 288 с.
3. Неофитов, А. В. Apache Kafka: потоковая обработка данных. — М.: ДМК Пресс, 2023. — 256 с.
4. Лукьянова, Е. С. *Kubernetes: оркестрация контейнеров.* — СПб.: БХВ, 2022. — 384 с.
5. Петров, И. К. *Микросервисная архитектура: принципы и паттерны.* — М.: Инфра-Инженерия, 2020. — 412 с.
6. Смирнов, П. А. *Анализ больших данных в сельском хозяйстве.* — М.: Агропромиздат, 2023. — 198 с.
7. Документация PostgreSQL. — <https://www.postgresql.org/docs/>
8. Васильев, Д. Р. *Цифровая трансформация госсектора.* — М.: Альпина Паблишер, 2021. — 304 с.
9. Громов, А. Ю. *Интеграция ведомственных информационных систем.* — М.: ИТ-Академия, 2022. — 176 с.
10. Козлов, В. И. *Tarantool: высоконагруженные базы данных.* — СПб.: Питер, 2023. — 224 с.