



МИНОБРНАУКИ РОССИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«МИРЭА – Российский технологический университет»  
**РТУ МИРЭА**

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**  
по направлению дополнительной программы профессиональной переподготовки  
«Технологии Девопс»

**Тема работы: «Тема берется из списка тем»**

Обучающийся:

\_\_\_\_\_  
(Подпись)

студент Д.О. Враженко

Руководитель работы

\_\_\_\_\_  
(Подпись)

доцент А.Т. Тарланов

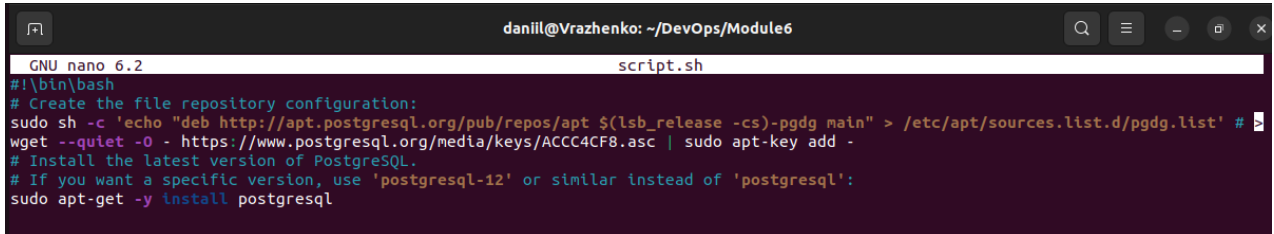
Москва 2025 г.

## Оглавление

Модуль 6. Администрирование баз данных.....	3
Модуль 8. Мониторинг, логирование и оповещение событий.....	7
Модуль 9. Виртуализация в DevOps.....	13
Модуль 14. Жизненный цикл ПО.....	15
Вывод.....	22

## Модуль 6. Администрирование баз данных

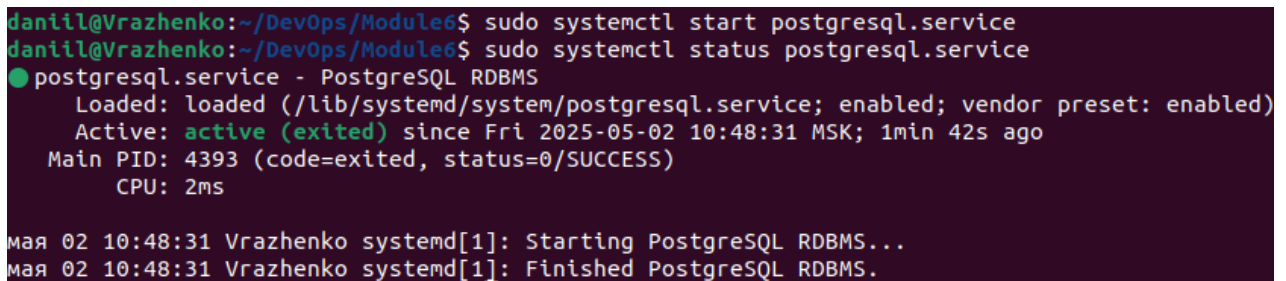
Установим postgresql 12 на ubuntu (рис. 1):



```
danil@Vrazhenko: ~/DevOps/Module6
GNU nano 6.2 script.sh
#!/bin/bash
# Create the file repository configuration:
sudo sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt $(lsb_release -cs)-pgdg main" > /etc/apt/sources.list.d/pgdg.list' #
wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key add -
# Install the latest version of PostgreSQL.
# If you want a specific version, use 'postgresql-12' or similar instead of 'postgresql':
sudo apt-get -y install postgresql
```

Рисунок 1 - Установка PostgreSQL

Проверим работоспособность базы данных (рис. 2):

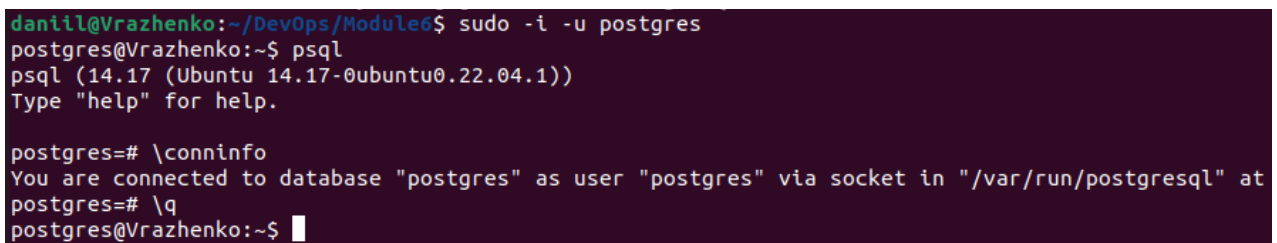


```
danil@Vrazhenko:~/DevOps/Module6$ sudo systemctl start postgresql.service
danil@Vrazhenko:~/DevOps/Module6$ sudo systemctl status postgresql.service
● postgresql.service - PostgreSQL RDBMS
   Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; vendor preset: enabled)
   Active: active (exited) since Fri 2025-05-02 10:48:31 MSK; 1min 42s ago
     Main PID: 4393 (code=exited, status=0/SUCCESS)
        CPU: 2ms

мая 02 10:48:31 Vrazhenko systemd[1]: Starting PostgreSQL RDBMS...
мая 02 10:48:31 Vrazhenko systemd[1]: Finished PostgreSQL RDBMS.
```

Рисунок 2 - Проверка работоспособности

Войдём в potgres (рис. 3):

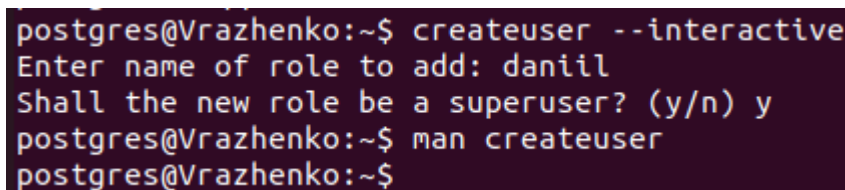


```
danil@Vrazhenko:~/DevOps/Module6$ sudo -i -u postgres
postgres@Vrazhenko:~$ psql
psql (14.17 (Ubuntu 14.17-0ubuntu0.22.04.1))
Type "help" for help.

postgres=# \conninfo
You are connected to database "postgres" as user "postgres" via socket in "/var/run/postgresql" at
postgres=# \q
postgres@Vrazhenko:~$
```

Рисунок 3 - Вход в postgres

Создадим пользователя (рис. 4):



```
postgres@Vrazhenko:~$ createuser --interactive
Enter name of role to add: danil
Shall the new role be a superuser? (y/n) y
postgres@Vrazhenko:~$ man createuser
postgres@Vrazhenko:~$
```

Рисунок 4 - Создание пользователя

Создадим базу данных (рис. 5):

```
danil@Vrazhenko:~/DevOps/Module6$ sudo -u danil psql
psql (14.17 (Ubuntu 14.17-0ubuntu0.22.04.1))
Type "help" for help.

danil=# \conninfo
You are connected to database "danil" as user "danil" via socket in "/var/run/postgresql" at port "5432"
danil=#
```

Рисунок 5 - Создание базы данных

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

```
danil=# CREATE TABLE snowboarder (
equip_id serial PRIMARY KEY,
title varchar (50) NOT NULL,
company varchar (25) NOT NULL,
size varchar (25) check (size in ('XS', 'S', 'M', 'L', 'XL', 'XXL'))
);
CREATE TABLE
danil=# \d
               List of relations
Schema |           Name           |  Type   | Owner
-----+-----+-----+-----
public | snowboarder              | table   | danil
public | snowboarder_equip_id_seq | sequence | danil
(2 rows)

danil=# \dt
               List of relations
Schema |   Name   | Type | Owner
-----+-----+-----+-----
public | snowboarder | table | danil
(1 row)
```

Рисунок 6 - Создание и просмотр базы данных

Заполним таблицу данными (рис. 7):

```
danil=# INSERT INTO snowboarder (title, company, size) VALUES ('snowboard', 'burton', 'XL');
INSERT 0 1
danil=# INSERT INTO snowboarder (title, company, size) VALUES ('binding', 'burton', 'XL');
INSERT 0 1
danil=# INSERT INTO snowboarder (title, company, size) VALUES ('boots', 'burton', 'XL');
INSERT 0 1
danil=#
```

Рисунок 7 - Заполнение таблицы данными

Проверим таблицу (рис. 8):

```

daniil=# SELECT * FROM snowboarder;
 equip_id | title      | company | size
-----+-----+-----+-----
          1 | snowboard  | burton  | XL
          2 | binding    | burton  | XL
          3 | boots      | burton  | XL
(3 rows)

```

**Рисунок 8 - Проверка таблицы**

Дополним таблицу новыми данными (рис. 9):

```

daniil=# INSERT INTO snowboarder (title, company, size) VALUES ('helmet', 'burton', 'XL');
INSERT 0 1
daniil=# INSERT INTO snowboarder (title, company, size) VALUES ('goggles', 'burton', 'XL');
INSERT 0 1
daniil=# INSERT INTO snowboarder (title, company, size) VALUES ('gloves', 'burton', 'XXL');
INSERT 0 1
daniil=# INSERT INTO snowboarder (title, company, size) VALUES ('jacket', 'burton', 'XXL');
INSERT 0 1
daniil=# INSERT INTO snowboarder (title, company, size) VALUES ('pants', 'burton', 'XL');
INSERT 0 1
daniil=# INSERT INTO snowboarder (title, company, size) VALUES ('socks', 'burton', 'XL');
INSERT 0 1
daniil=# INSERT INTO snowboarder (title, company, size) VALUES ('snowboard bag', 'burton', 'XL');
INSERT 0 1
daniil=# INSERT INTO snowboarder (title, company, size) VALUES ('snowboard', 'dakron', 'XXL');
INSERT 0 1
daniil=# INSERT INTO snowboarder (title, company, size) VALUES ('snowboard', 'dakron', 'L');
INSERT 0 1
daniil=# INSERT INTO snowboarder (title, company, size) VALUES ('snowboard', 'vorton', 'XS');
INSERT 0 1
daniil=# INSERT INTO snowboarder (title, company, size) VALUES ('snowboard', 'vorton', 'S');
INSERT 0 1
daniil=# INSERT INTO snowboarder (title, company, size) VALUES ('snowboard', 'vorton', 'M');
INSERT 0 1
daniil=# INSERT INTO snowboarder (title, company, size) VALUES ('snowboard', 'vorton', 'L');
INSERT 0 1
daniil=# INSERT INTO snowboarder (title, company, size) VALUES ('snowboard', 'vorton', 'XL');
INSERT 0 1
daniil=# INSERT INTO snowboarder (title, company, size) VALUES ('snowboard', 'vorton', 'XXL');
INSERT 0 1

```

**Рисунок 9 - Дополнение таблицы данными**

Снова проверим таблицу (рис. 10):

```

danil=# SELECT * FROM snowboarder;
 equip_id |      title      | company | size
-----+-----+-----+-----
          1 | snowboard       | burton  | XL
          2 | binding         | burton  | XL
          3 | boots           | burton  | XL
          4 | helmet          | burton  | XL
          5 | goggles         | burton  | XL
          6 | gloves          | burton  | XXL
          7 | jacket          | burton  | XXL
          8 | pants           | burton  | XL
          9 | socks           | burton  | XL
         10 | snowboard bag   | burton  | XL
         11 | snowboard       | dakron  | XXL
         12 | snowboard       | dakron  | L
         13 | snowboard       | vorton  | XS
         14 | snowboard       | vorton  | S
         15 | snowboard       | vorton  | M
         16 | snowboard       | vorton  | L
         17 | snowboard       | vorton  | XL
         18 | snowboard       | vorton  | XXL
(18 rows)

```

Рисунок 10 - Повторная проверка таблицы

## Модуль 8. Мониторинг, логирование и оповещение событий

Загрузим сжатый пакет `node_exporter` (рис. 11):

```
daniil@Vrazhenko:~/DevOps/Module8$ wget https://github.com/prometheus/node_exporter/releases/download/v1.9.1/node_exporter-1.9.1.linux-amd64.tar.gz
--2025-05-10 13:38:33-- https://github.com/prometheus/node_exporter/releases/download/v1.9.1/node_exporter-1.9.1.linux-amd64.tar.gz
Распознаётся github.com (github.com)... 140.82.121.4
Подключение к github.com (github.com)[140.82.121.4]:443... соединение установлено.
HTTP-запрос отправлен. Ожидание ответа... 302 Found
Адрес: https://objects.githubusercontent.com/github-production-release-asset-2e65be/9524057/dc8ec09c-2975-42a2-9591-57dd1ffff7b7?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20250510%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20250510T103836Z&X-Amz-Expires=300&X-Amz-Signature=f834b6d80fd3ba1ad89db49356e1f1319bf9f19aa3dfe29726f0017881d30491&X-Amz-SignedHeaders=host&response-content-disposition=attachment%3B%20filename%3Dnode_exporter-1.9.1.linux-amd64.tar.gz&response-content-type=application%2Foctet-stream [переход]
--2025-05-10 13:38:36-- https://objects.githubusercontent.com/github-production-release-asset-2e65be/9524057/dc8ec09c-2975-42a2-9591-57dd1ffff7b7?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20250510%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20250510T103836Z&X-Amz-Expires=300&X-Amz-Signature=f834b6d80fd3ba1ad89db49356e1f1319bf9f19aa3dfe29726f0017881d30491&X-Amz-SignedHeaders=host&response-content-disposition=attachment%3B%20filename%3Dnode_exporter-1.9.1.linux-amd64.tar.gz&response-content-type=application%2Foctet-stream
Распознаётся objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.110.133, 185.199.111.133, 185.199.108.133, ...
Подключение к objects.githubusercontent.com (objects.githubusercontent.com)[185.199.110.133]:443... соединение установлено.
HTTP-запрос отправлен. Ожидание ответа... 200 OK
Длина: 11582410 (11M) [application/octet-stream]
Сохранение в: 'node_exporter-1.9.1.linux-amd64.tar.gz'

node_exporter-1.9.1.linux-amd64. 100%[=====] 11,04M 168KB/s за 63s
2025-05-10 13:39:43 (178 KB/s) - 'node_exporter-1.9.1.linux-amd64.tar.gz' сохранён [11582410/11582410]
```

Рисунок 11 - Загрузка `node_exporter`

Извлечем двоичный файл `node_exporter` (рис. 12):

```
daniil@Vrazhenko:~/DevOps/Module8$ tar xvfz node_exporter-1.9.1.linux-amd64.tar.gz
node_exporter-1.9.1.linux-amd64/
node_exporter-1.9.1.linux-amd64/LICENSE
node_exporter-1.9.1.linux-amd64/NOTICE
node_exporter-1.9.1.linux-amd64/node_exporter
```

Рисунок 12 - Извлечение двоичного файла `node_exporter`

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

```
daniil@Vrazhenko:~/DevOps/Module8$ cd node_exporter-1.9.1.linux-amd64/
daniil@Vrazhenko:~/DevOps/Module8/node_exporter-1.9.1.linux-amd64$ chmod +x node_exporter
```

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)



```

danil@Vrazhenko:~/DevOps/Module8/node_exporter-1.9.1.linux-amd64$ ./node_exporter
time=2025-05-10T10:41:13.326Z level=INFO source=node_exporter.go:216 msg="Starting node_exporter" version="(version=1.9.1, branch=HEAD, revision=f2ec547b49af53815038a50265aa2adcd1275959)"
time=2025-05-10T10:41:13.326Z level=INFO source=node_exporter.go:217 msg="Build context" build_context="(go=go1.23.7, platform=linux/amd64, user=root@7023beaa563a, date=20250401-15:19:01, tags=unknown)"
time=2025-05-10T10:41:13.330Z level=INFO source=diskstats_common.go:110 msg="Parsed flag --collector.diskstats.device-exclude" collector=diskstats flag=(z7ram|loop|fd|(h|s|v|xv)d[a-z]|nvme\d+n\d+p)\d+$
time=2025-05-10T10:41:13.330Z level=INFO source=filesystem_common.go:265 msg="Parsed flag --collector.filesystem.mount-points-exclude" collector=filesystem flag=(/dev|proc|run|credentials/.+|sys|var/lib/docker/.+|var/lib/containers/storage/.+)(/$|/)|
time=2025-05-10T10:41:13.330Z level=INFO source=filesystem_common.go:294 msg="Parsed flag --collector.filesystem.fs-types-exclude" collector=filesystem flag=(autofs|binfmt_misc|bpf|cgroup2?|configfs|debugfs|devpts|devtmpfs|fusectl|hugetlbfs|iso9660|mqueue|nsfs|ov
erlay|proc|procfs|pstore|rpc_pipefs|securityfs|selinuxfs|squashfs|sysfs|tracefs)$
time=2025-05-10T10:41:13.330Z level=INFO source=node_exporter.go:135 msg="Enabled collectors"
time=2025-05-10T10:41:13.331Z level=INFO source=node_exporter.go:141 msg=arp
time=2025-05-10T10:41:13.331Z level=INFO source=node_exporter.go:141 msg=bcache
time=2025-05-10T10:41:13.331Z level=INFO source=node_exporter.go:141 msg=bonding
time=2025-05-10T10:41:13.331Z level=INFO source=node_exporter.go:141 msg=btrfs
time=2025-05-10T10:41:13.331Z level=INFO source=node_exporter.go:141 msg=conntrack
time=2025-05-10T10:41:13.331Z level=INFO source=node_exporter.go:141 msg=cpu
time=2025-05-10T10:41:13.331Z level=INFO source=node_exporter.go:141 msg=cputime
time=2025-05-10T10:41:13.331Z level=INFO source=node_exporter.go:141 msg=diskstats
time=2025-05-10T10:41:13.331Z level=INFO source=node_exporter.go:141 msg=dmide
time=2025-05-10T10:41:13.331Z level=INFO source=node_exporter.go:141 msg=edac
time=2025-05-10T10:41:13.331Z level=INFO source=node_exporter.go:141 msg=entropy
time=2025-05-10T10:41:13.331Z level=INFO source=node_exporter.go:141 msg=fibrechannel
time=2025-05-10T10:41:13.331Z level=INFO source=node_exporter.go:141 msg=filefd
time=2025-05-10T10:41:13.331Z level=INFO source=node_exporter.go:141 msg=filesystem
time=2025-05-10T10:41:13.331Z level=INFO source=node_exporter.go:141 msg=hwmon
time=2025-05-10T10:41:13.331Z level=INFO source=node_exporter.go:141 msg=infiniband
time=2025-05-10T10:41:13.331Z level=INFO source=node_exporter.go:141 msg=ipvs
time=2025-05-10T10:41:13.331Z level=INFO source=node_exporter.go:141 msg=loadavg
time=2025-05-10T10:41:13.331Z level=INFO source=node_exporter.go:141 msg=mdadm
time=2025-05-10T10:41:13.331Z level=INFO source=node_exporter.go:141 msg=meminfo
time=2025-05-10T10:41:13.331Z level=INFO source=node_exporter.go:141 msg=netclass
time=2025-05-10T10:41:13.331Z level=INFO source=node_exporter.go:141 msg=netdev
time=2025-05-10T10:41:13.331Z level=INFO source=node_exporter.go:141 msg=netstat
time=2025-05-10T10:41:13.331Z level=INFO source=node_exporter.go:141 msg=nfs
time=2025-05-10T10:41:13.331Z level=INFO source=node_exporter.go:141 msg=nfsd

```

Рисунок 14 - Первая часть вывода

```

time=2025-05-10T10:41:13.331Z level=INFO source=node_exporter.go:141 msg=nvme
time=2025-05-10T10:41:13.331Z level=INFO source=node_exporter.go:141 msg=os
time=2025-05-10T10:41:13.331Z level=INFO source=node_exporter.go:141 msg=powersupplyclass
time=2025-05-10T10:41:13.331Z level=INFO source=node_exporter.go:141 msg=pressure
time=2025-05-10T10:41:13.331Z level=INFO source=node_exporter.go:141 msg=rapl
time=2025-05-10T10:41:13.331Z level=INFO source=node_exporter.go:141 msg=schedstat
time=2025-05-10T10:41:13.331Z level=INFO source=node_exporter.go:141 msg=selinux
time=2025-05-10T10:41:13.331Z level=INFO source=node_exporter.go:141 msg=sockstat
time=2025-05-10T10:41:13.331Z level=INFO source=node_exporter.go:141 msg=softnet
time=2025-05-10T10:41:13.331Z level=INFO source=node_exporter.go:141 msg=stat
time=2025-05-10T10:41:13.331Z level=INFO source=node_exporter.go:141 msg=tapestats
time=2025-05-10T10:41:13.331Z level=INFO source=node_exporter.go:141 msg=textfile
time=2025-05-10T10:41:13.331Z level=INFO source=node_exporter.go:141 msg=thermal_zone
time=2025-05-10T10:41:13.331Z level=INFO source=node_exporter.go:141 msg=time
time=2025-05-10T10:41:13.331Z level=INFO source=node_exporter.go:141 msg=timex
time=2025-05-10T10:41:13.331Z level=INFO source=node_exporter.go:141 msg=udp_queues
time=2025-05-10T10:41:13.331Z level=INFO source=node_exporter.go:141 msg=uname
time=2025-05-10T10:41:13.331Z level=INFO source=node_exporter.go:141 msg=vmstat
time=2025-05-10T10:41:13.331Z level=INFO source=node_exporter.go:141 msg=watchdog
time=2025-05-10T10:41:13.331Z level=INFO source=node_exporter.go:141 msg=xfs
time=2025-05-10T10:41:13.331Z level=INFO source=node_exporter.go:141 msg=zfs
time=2025-05-10T10:41:13.332Z level=INFO source=tls_config.go:347 msg="Listening on" address=[::]:9100
time=2025-05-10T10:41:13.332Z level=INFO source=tls_config.go:350 msg="TLS is disabled." http2=false address=[::]:9100

```

Рисунок 15 - Вторая часть вывода

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

```

danil@Vrazhenko:~/DevOps/Module8/node_exporter-1.9.1.linux-amd64$ curl http://localhost:9100/metrics
# HELP go_gc_duration_seconds A summary of the wall-time pause (stop-the-world) duration in garbage collection cycles.
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 0
go_gc_duration_seconds{quantile="0.25"} 0
go_gc_duration_seconds{quantile="0.5"} 0
go_gc_duration_seconds{quantile="0.75"} 0
go_gc_duration_seconds{quantile="1"} 0
go_gc_duration_seconds_sum 0
go_gc_duration_seconds_count 0
# HELP go_gc_gogc_percent Heap size target percentage configured by the user, otherwise 100. This value is set by the GOGC environme
nt variable, and the runtime/debug.SetGCPercent function. Sourced from /gc/gogc:percent
# TYPE go_gc_gogc_percent gauge
go_gc_gogc_percent 100

```

Рисунок 16 - Наименьшая часть вывода



Загрузим архив пакета Prometheus (рис. 17):

```
danill@Vrazhenko:~/DevOps/Module8$ wget https://github.com/prometheus/prometheus/releases/download/v3.3.1/prometheus-3.3.1.linux-amd64.tar.gz
--2025-05-10 14:57:13-- https://github.com/prometheus/prometheus/releases/download/v3.3.1/prometheus-3.3.1.linux-amd64.tar.gz
Распознается github.com (github.com)... 140.82.121.3
Подключение к github.com (github.com)[140.82.121.3]:443... соединение установлено.
HTTP-запрос отправлен. Ожидание ответа... 302 Found
Адрес: https://objects.githubusercontent.com/github-production-release-asset-2e65be/6838921/39e5bf8c-512d-4389-9add-1986b432ee5b?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20250510%2Fus-east-1%2Ffs3%2Faws4_request&X-Amz-Date=20250510T115715Z&X-Amz-Expires=300&X-Amz-Signature=97eae5d52a0ee061fb4f207e65d1a8327cf6503c59f37580f05cf4a1294712&X-Amz-SignedHeaders=host&response-content-disposition=attachment%3B%20filename%3Dprometheus-3.3.1.linux-amd64.tar.gz&response-content-type=application%2Foctet-stream [переход]
--2025-05-10 14:57:15-- https://objects.githubusercontent.com/github-production-release-asset-2e65be/6838921/39e5bf8c-512d-4389-9add-1986b432ee5b?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20250510%2Fus-east-1%2Ffs3%2Faws4_request&X-Amz-Date=20250510T115715Z&X-Amz-Expires=300&X-Amz-Signature=97eae5d52a0ee061fb4f207e65d1a8327cf6503c59f37580f05cf4a1294712&X-Amz-SignedHeaders=host&response-content-disposition=attachment%3B%20filename%3Dprometheus-3.3.1.linux-amd64.tar.gz&response-content-type=application%2Foctet-stream
Распознается objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.109.133, 185.199.110.133, 185.199.111.133, ...
Подключение к objects.githubusercontent.com (objects.githubusercontent.com)[185.199.109.133]:443... соединение установлено.
HTTP-запрос отправлен. Ожидание ответа... 200 OK
Длина: 116682535 (111М) [application/octet-stream]
Сохранение в: 'prometheus-3.3.1.linux-amd64.tar.gz'

prometheus-3.3.1.linux-amd64.tar 100%[=====] 111,28M 171KB/s за 10m 53s
2025-05-10 15:08:11 (174 KB/s) - 'prometheus-3.3.1.linux-amd64.tar.gz' сохранён [116682535/116682535]
```

Рисунок 17 - Загрузка пакета Prometheus

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

```
danill@Vrazhenko:~/DevOps/Module8$ tar xvf prometheus-3.3.1.linux-amd64.tar.gz
prometheus-3.3.1.linux-amd64/
prometheus-3.3.1.linux-amd64/LICENSE
prometheus-3.3.1.linux-amd64/NOTICE
prometheus-3.3.1.linux-amd64/prometheus
prometheus-3.3.1.linux-amd64/prometheus.yml
prometheus-3.3.1.linux-amd64/promtool
danill@Vrazhenko:~/DevOps/Module8$ cd prometheus-3.3.1.linux-amd64/
```

Рисунок 18 - Извлечение двоичного файла и переход в каталог

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

```
global:
  scrape_interval: 15s

scrape_configs:
  - job_name: node
    static_configs:
      - targets: ["localhost:9100"]
```

Рисунок 19 - Содержимое prometheus.yml

Запустим Prometheus (рис. 20):

```

danil@Vrazhenko: ~/DevOps/Module8/prometheus-3.3.1.linux-amd64$ ./prometheus --config.file=./prometheus.yml
time=2025-05-10T12:24:32.846Z level=INFO source=main.go:1487 msg="updated GOGC" old=100 new=75
time=2025-05-10T12:24:32.846Z level=INFO source=main.go:625 msg="Leaving GOMAXPROCS=7: CPU quota undefined" component=automaxprocs
time=2025-05-10T12:24:32.847Z level=INFO source=main.go:666 msg="No time or size retention was set so using the default time retention" duration=15d
time=2025-05-10T12:24:32.847Z level=INFO source=main.go:713 msg="Starting Prometheus Server" mode=server version="(version=3.3.1, branch=HEAD, revision=3dcecabff6f52cc155290cb723706f4c324ed0ab)"
time=2025-05-10T12:24:32.847Z level=INFO source=main.go:718 msg="operational information" build_context="(go=go1.24.2, platform=linux/amd64, user=root@f0d295ca0db2, date=20250502-15:03:21, tags=netgo,builtinassets,stringlabels)" host_details="(Linux 6.8.0-58-generic #60-22.04.1-Ubuntu SMP PREEMPT_DYNAMIC Fri Mar 28 16:09:21 UTC 2 x86_64 Vrazhenko (none))" fd_limits="(soft=1048575, hard=1048576)" vm_limits="(soft=unlimited, hard=unlimited)"
time=2025-05-10T12:24:32.848Z level=INFO source=web.go:654 msg="Start listening for connections" component=web address=0.0.0.0:9090
time=2025-05-10T12:24:32.850Z level=INFO source=main.go:1231 msg="Starting TSDB ..."
time=2025-05-10T12:24:32.852Z level=INFO source=tls_config.go:347 msg="Listening on" component=web address=[::]:9090
time=2025-05-10T12:24:32.852Z level=INFO source=tls_config.go:350 msg="TLS is disabled." component=web http2=false address=[::]:9090
time=2025-05-10T12:24:32.856Z level=INFO source=head.go:638 msg="Replaying on-disk memory mappable chunks if any" component=tsdb
time=2025-05-10T12:24:32.856Z level=INFO source=head.go:725 msg="On-disk memory mappable chunks replay completed" component=tsdb duration=5.241µs
time=2025-05-10T12:24:32.857Z level=INFO source=head.go:733 msg="Replaying WAL, this may take a while" component=tsdb
time=2025-05-10T12:24:32.857Z level=INFO source=head.go:805 msg="WAL segment loaded" component=tsdb segment=0 maxSegment=0
time=2025-05-10T12:24:32.857Z level=INFO source=head.go:842 msg="WAL replay completed" component=tsdb checkpoint_replay_duration=18.398µs wal_replay_duration=592.823µs wbl_replay_duration=81ns chunk_snapshot_load_duration=0s mmap_chunk_replay_duration=5.241µs total_replay_duration=705.826µs
time=2025-05-10T12:24:32.859Z level=INFO source=main.go:1252 msg="filesystem information" fs_type=EXT4_SUPER_MAGIC
time=2025-05-10T12:24:32.859Z level=INFO source=main.go:1255 msg="TSDB started"
time=2025-05-10T12:24:32.859Z level=INFO source=main.go:1440 msg="Loading configuration file" filename=./prometheus.yml
time=2025-05-10T12:24:32.860Z level=INFO source=main.go:1480 msg="Completed loading of configuration file" db_storage=2.038µs remote_storage=2.545µs web_handler=629ns query_engine=1.492µs scrape=393.208µs scrape_sd=21.433µs notify=1.672µs notify_sd=416ns rules=10.038µs tracing=6.079µs filename=./prometheus.yml totalDuration=854.464µs
time=2025-05-10T12:24:32.860Z level=INFO source=main.go:1216 msg="Server is ready to receive web requests."
time=2025-05-10T12:24:32.860Z level=INFO source=manager.go:175 msg="Starting rule manager..." component="rule manager"

```

Рисунок 20 - Запуск Prometheus

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

```

danil@Vrazhenko: ~/DevOps/Module8$ sudo dpkg -i grafana-enterprise_12.0.0_amd64.deb
[sudo] пароль для danil:
Выбор ранее не выбранного пакета grafana-enterprise.
(Чтение базы данных ... на данный момент установлено 206402 файла и каталога.)
Подготовка к распаковке grafana-enterprise_12.0.0_amd64.deb ...
Распаковывается grafana-enterprise (12.0.0) ...
Настраивается пакет grafana-enterprise (12.0.0) ...
Добавляется системный пользователь «grafana» (UID 130) ...
Добавляется новый пользователь «grafana» (UID 130) в группу «grafana» ...
Не создается домашний каталог «/usr/share/grafana».
### NOT starting on installation, please execute the following statements to configure grafana to start automatically using systemd
sudo /bin/systemctl daemon-reload
sudo /bin/systemctl enable grafana-server
### You can start grafana-server by executing
sudo /bin/systemctl start grafana-server

```

Рисунок 21 - Распаковка пакета .deb

Запустим grafana-server (рис. 22):

```

danil@Vrazhenko: ~/DevOps/Module8$ sudo systemctl daemon-reload
[sudo] пароль для danil:
danil@Vrazhenko: ~/DevOps/Module8$ sudo systemctl start grafana-server
danil@Vrazhenko: ~/DevOps/Module8$ sudo systemctl status grafana-server
● grafana-server.service - Grafana instance
   Loaded: loaded (/lib/systemd/system/grafana-server.service; disabled; vendor preset: enabled)
   Active: active (running) since Sat 2025-05-10 19:31:38 MSK; 14s ago
     Docs: http://docs.grafana.org
    Main PID: 2185 (grafana)
      Tasks: 22 (limit: 6795)
     Memory: 322.5M
        CPU: 4.397s
    CGroup: /system.slice/grafana-server.service
            └─2185 /usr/share/grafana/bin/grafana server --config=/etc/grafana/grafana.ini --pidfile=/run/grafana/grafana-server.p

```

Рисунок 22 - Запуск grafana-server

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

```

danil@Vrazhenko: ~/DevOps/Module8$ sudo systemctl enable grafana-server.service
Synchronizing state of grafana-server.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable grafana-server
Created symlink /etc/systemd/system/multi-user.target.wants/grafana-server.service → /lib/systemd/system/grafana-server.service.

```

Рисунок 23 - Настройка grafana-server

Настроим источники данных (рис. 24):

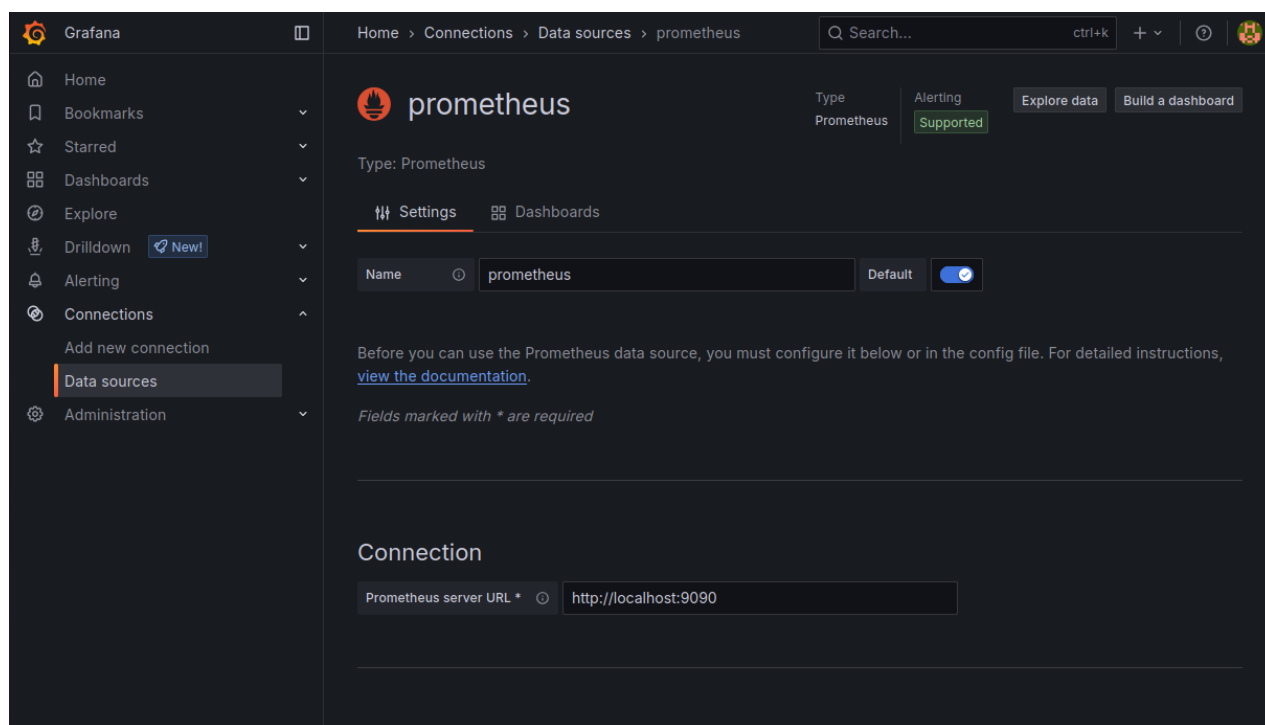


Рисунок 24 - Настройки источников данных

Используем готовый дашборд (рис. 25):

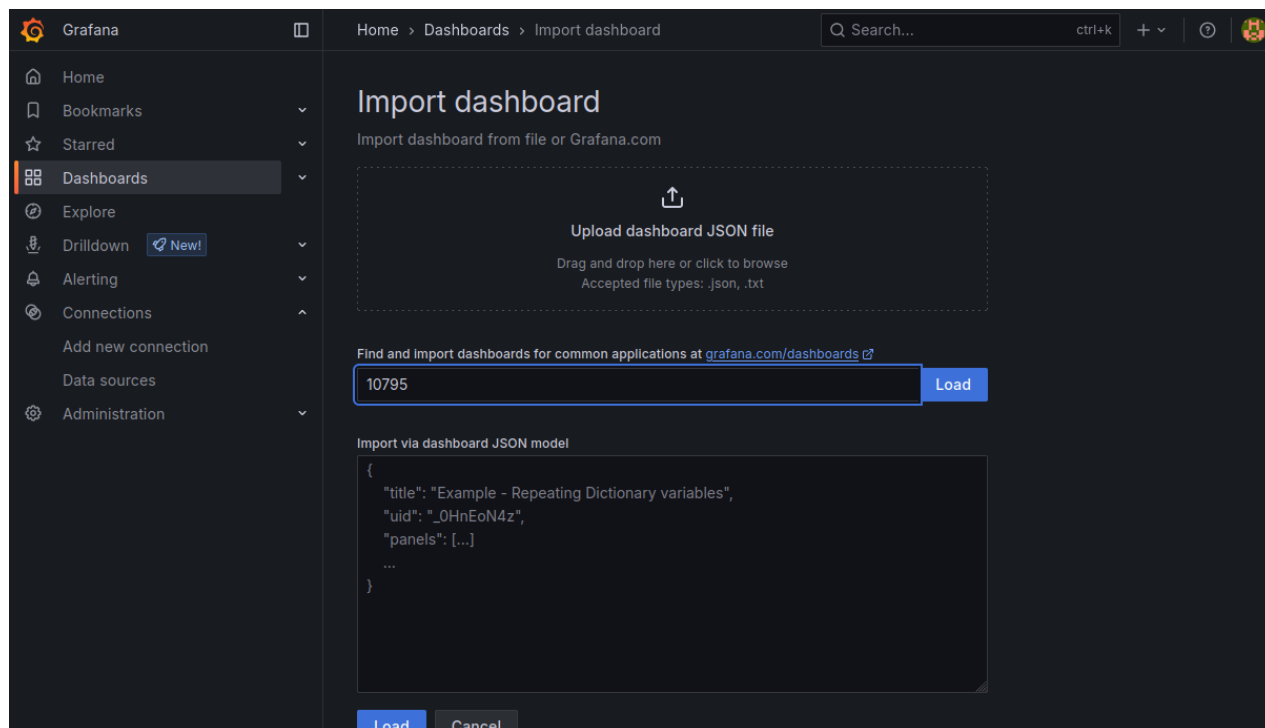


Рисунок 25 - Импорт готового дашборда

Откроем установленный дашборд (рис. 26):

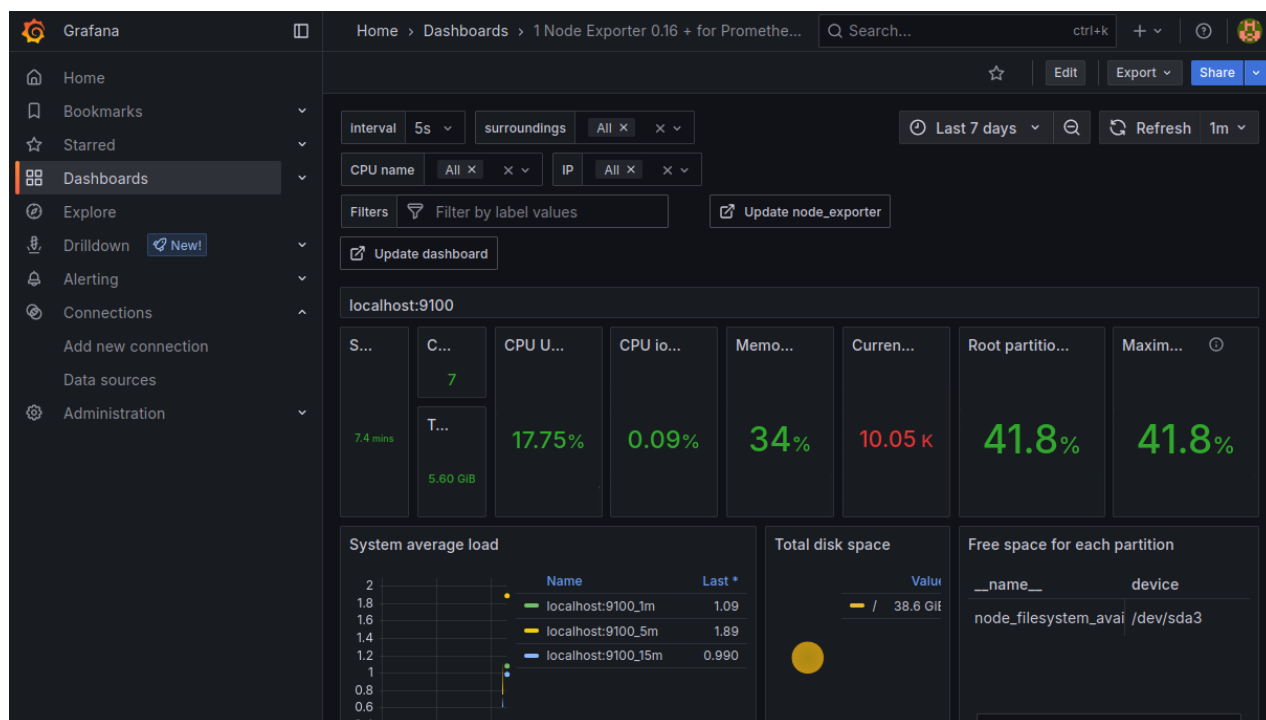


Рисунок 26 - Установленный дашборд

## Модуль 9. Виртуализация в DevOps

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

```
daniil@Vrazhenko:~/DevOps/Module9$ sudo apt-get update
Сущ:1 http://ru.archive.ubuntu.com/ubuntu jammy InRelease
Сущ:2 http://ru.archive.ubuntu.com/ubuntu jammy-updates InRelease
Сущ:3 http://apt.postgresql.org/pub/repos/apt jammy-pgdg InRelease
Сущ:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Сущ:5 http://ru.archive.ubuntu.com/ubuntu jammy-backports InRelease
Чтение списков пакетов... Готово
```

Рисунок 27 - Обновление базы данных пакетов

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

```
daniil@Vrazhenko:~/DevOps/Module9$ sudo apt -y install vim
Чтение списков пакетов... Готово
Построение дерева зависимостей... Готово
Чтение информации о состоянии... Готово
Будут установлены следующие дополнительные пакеты:
  vim-common vim-runtime vim-tiny
Предлагаемые пакеты:
```

Рисунок 28 - Установка vim

```
daniil@Vrazhenko:~/DevOps/Module9$ vim --version
VIM - Vi IMproved 8.2 (2019 Dec 12, собрано Apr 02 2025 12:39:01)
Запатки: 1-16, 647, 17-579, 1969, 580-647, 678, 648-1848, 4975,
8-1859, 1873, 1860-1969, 1992, 1970-1992, 2010, 1993-2068, 2106,
, 4901, 4919, 213, 1840, 1846-1847, 2110-2112, 2121
С изменениями, внесёнными team+vim@tracker.debian.org
Скомпилировано: team+vim@tracker.debian.org
```

Рисунок 29 - Версия vim

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

```
daniil@Vrazhenko:~/DevOps/Module9$ sudo apt -y install mc
Чтение списков пакетов... Готово
Построение дерева зависимостей... Готово
Чтение информации о состоянии... Готово
Будут установлены следующие дополнительные пакеты:
  libssh2-1 mc-data
Предлагаемые пакеты:
```

Рисунок 30 - Установка mc

```
daniil@Vrazhenko:~/DevOps/Module9$ mc --version
GNU Midnight Commander, версия 4.8.27
Скомпилирован с библиотекой GLib версии 2.68.4
С библиотекой S-Lang 2.3.2 и с базой данных terminfo
Со встроенным редактором и поддержкой Aspell
```

Рисунок 31 - Версия mc

Установка net-tools (рис. 32):

```
daniil@Vrazhenko:~$ sudo apt -y install net-tools
[sudo] пароль для daniil:
Чтение списков пакетов... Готово
Построение дерева зависимостей... Готово
Чтение информации о состоянии... Готово
Следующие НОВЫЕ пакеты будут установлены:
 net-tools
Обновлено 0 пакетов, установлено 1 новых пакетов,
```

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):

```
daniil@Vrazhenko:~$ ifconfig --version
net-tools 2.10-alpha
```

Рисунок 33 - Версия net-tools



# Модуль 14. Жизненный цикл ПО

Установим Jenkins с помощью официальной инструкции (рис. 34-35):

```
danil@Vrazhenko:~/DevOps/Module14$ sudo wget -O /etc/apt/keyrings/jenkins-keyring.asc \
https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
echo "deb [signed-by=/etc/apt/keyrings/jenkins-keyring.asc]" \
https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
/etc/apt/sources.list.d/jenkins.list > /dev/null
sudo apt-get update
sudo apt-get install jenkins
--2025-05-11 11:36:06-- https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
Распознается pkg.jenkins.io (pkg.jenkins.io) 146.75.118.133, 2a04:4e42:8d::645
Подключение к pkg.jenkins.io (pkg.jenkins.io)[146.75.118.133]:443... соединение установлено.
HTTP-запрос отправлен. Ожидание ответа... 200 OK
Длина: 3175 (3,1K) [application/pgp-keys]
Сохранение в: '/etc/apt/keyrings/jenkins-keyring.asc'

/etc/apt/keyrings/jenkins-keyrin 100%[=====>] 3,10K --KB/s за 0s
2025-05-11 11:36:08 (25,0 MB/s) - '/etc/apt/keyrings/jenkins-keyring.asc' сохранён [3175/3175]

Сущ:1 http://ru.archive.ubuntu.com/ubuntu jammy InRelease
Сущ:2 http://security.ubuntu.com/ubuntu jammy-security InRelease
Сущ:3 http://apt.postgresql.org/pub/repos/apt jammy-pgdg InRelease
Сущ:4 http://ru.archive.ubuntu.com/ubuntu jammy-updates InRelease
Игн:5 https://pkg.jenkins.io/debian-stable binary/ InRelease
Сущ:6 http://ru.archive.ubuntu.com/ubuntu jammy-backports InRelease
Пол:7 https://pkg.jenkins.io/debian-stable binary/ Release [2 044 B]
Пол:8 https://pkg.jenkins.io/debian-stable binary/ Release.gpg [833 B]
Пол:9 https://pkg.jenkins.io/debian-stable binary/ Packages [29,0 kB]
Получено 31,8 kB за 5с (6 793 B/s)
Чтение списков пакетов... Готово
W: http://apt.postgresql.org/pub/repos/apt/dists/jammy-pgdg/InRelease: Ключ хранится в унаследованной связке ключей trusted.gpg (/etc/apt/trusted.gpg), подробности см. в разделе DEPRECATION в apt-key(8).
N: Пропускается получение настроенного файла «main/binary-i386/Packages», так как репозиторий «http://apt.postgresql.org/pub/repos/a
pt jammy-pgdg InRelease» не поддерживает архитектуру «i386»
Чтение списков пакетов... Готово
Построение дерева зависимостей... Готово
```

Рисунок 34 - Первая часть установки

```
Чтение информации о состоянии... Готово
Следующие НОВЫЕ пакеты будут установлены:
jenkins
Обновлено 0 пакетов, установлено 1 новых пакетов, для удаления отмечено 0 пакетов, и 250 пакетов не обновлено.
Необходимо скачать 92,2 МБ архивов.
После данной операции объём занятого дискового пространства возрастёт на 94,4 МБ.
Пол:1 https://pkg.jenkins.io/debian-stable binary/ jenkins 2.504.1 [92,2 МБ]
Получено 92,2 МБ за 6мин 15с (246 kB/s)
Выбор ранее не выбранного пакета jenkins.
(Чтение базы данных ... на данный момент установлено 221004 файла и каталога.)
Подготовка к распаковке .../jenkins_2.504.1_all.deb ...
Распаковывается jenkins (2.504.1) ...
Настраивается пакет jenkins (2.504.1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/jenkins.service → /lib/systemd/system/jenkins.service.
```

Рисунок 35 - Вторая часть установки

Настроим Jenkins как демон (рис. 36):



```

daniil@Vrazhenko:~/DevOps/Module14$ systemctl cat jenkins
# /lib/systemd/system/jenkins.service
#
# This file is managed by systemd(1). Do NOT edit this file manually!
# To override these settings, run:
#
#     systemctl edit jenkins
#
# For more information about drop-in files, see:
#
#     https://www.freedesktop.org/software/systemd/man/systemd.unit.html
#

[Unit]
Description=Jenkins Continuous Integration Server
Requires=network.target

```

Рисунок 36 - Просмотр подробностей

Создадим пользователя для запуска службы (рис. 37):

```

daniil@Vrazhenko:~/DevOps/Module14$ id jenkins
uid=131(jenkins) gid=139(jenkins) группы=139(jenkins)

```

Рисунок 37 - id пользователя

Установим Java (рис. 38-39):

```

daniil@Vrazhenko:~/DevOps/Module14$ sudo apt update
Сущ:1 http://ru.archive.ubuntu.com/ubuntu jammy InRelease
Сущ:2 http://apt.postgresql.org/pub/repos/apt jammy-pgdg InRelease
Сущ:3 http://ru.archive.ubuntu.com/ubuntu jammy-updates InRelease
Сущ:4 http://ru.archive.ubuntu.com/ubuntu jammy-backports InRelease
Сущ:5 http://security.ubuntu.com/ubuntu jammy-security InRelease
Игн:6 https://pkg.jenkins.io/debian-stable binary/ InRelease
Сущ:7 https://pkg.jenkins.io/debian-stable binary/ Release
Чтение списков пакетов... Готово
Построение дерева зависимостей... Готово
Чтение информации о состоянии... Готово
Может быть обновлено 250 пакетов. Запустите «apt list --upgradable» для
W: http://apt.postgresql.org/pub/repos/apt/dists/jammy-pgdg/InRelease:
с/apt/trusted.gpg), подробности см. в разделе DEPRECATION в apt-key(8)
N: Пропускается получение настроенного файла «main/binary-i386/Package
pt jammy-pgdg InRelease» не поддерживает архитектуру «i386»
daniil@Vrazhenko:~/DevOps/Module14$ sudo apt install openjdk-11-jre
Чтение списков пакетов... Готово
Построение дерева зависимостей... Готово
Чтение информации о состоянии... Готово
Будут установлены следующие дополнительные пакеты:

```

Рисунок 38 - Первая часть установки

```

0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...

done.
done.
danil@Vrazhenko:~/DevOps/Module14$ java --version
openjdk 11.0.27 2025-04-15
OpenJDK Runtime Environment (build 11.0.27+6-post-Ubuntu-0ubuntu122.04)
OpenJDK 64-Bit Server VM (build 11.0.27+6-post-Ubuntu-0ubuntu122.04, mixed mode, sharing)

```

Рисунок 39 - Вторая часть установки

Запустим Jenkins (рис. 40):

```

danil@Vrazhenko:~/DevOps/Module14$ sudo systemctl enable jenkins
Synchronizing state of jenkins.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable jenkins
danil@Vrazhenko:~/DevOps/Module14$ sudo systemctl start jenkins
danil@Vrazhenko:~/DevOps/Module14$ sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
   Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2025-05-11 17:45:04 MSK; 2min 6s ago
     Main PID: 7517 (java)
       Tasks: 50 (limit: 6795)
      Memory: 736.1M
         CPU: 28.321s
        CGroup: /system.slice/jenkins.service
                └─7517 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --http=

```

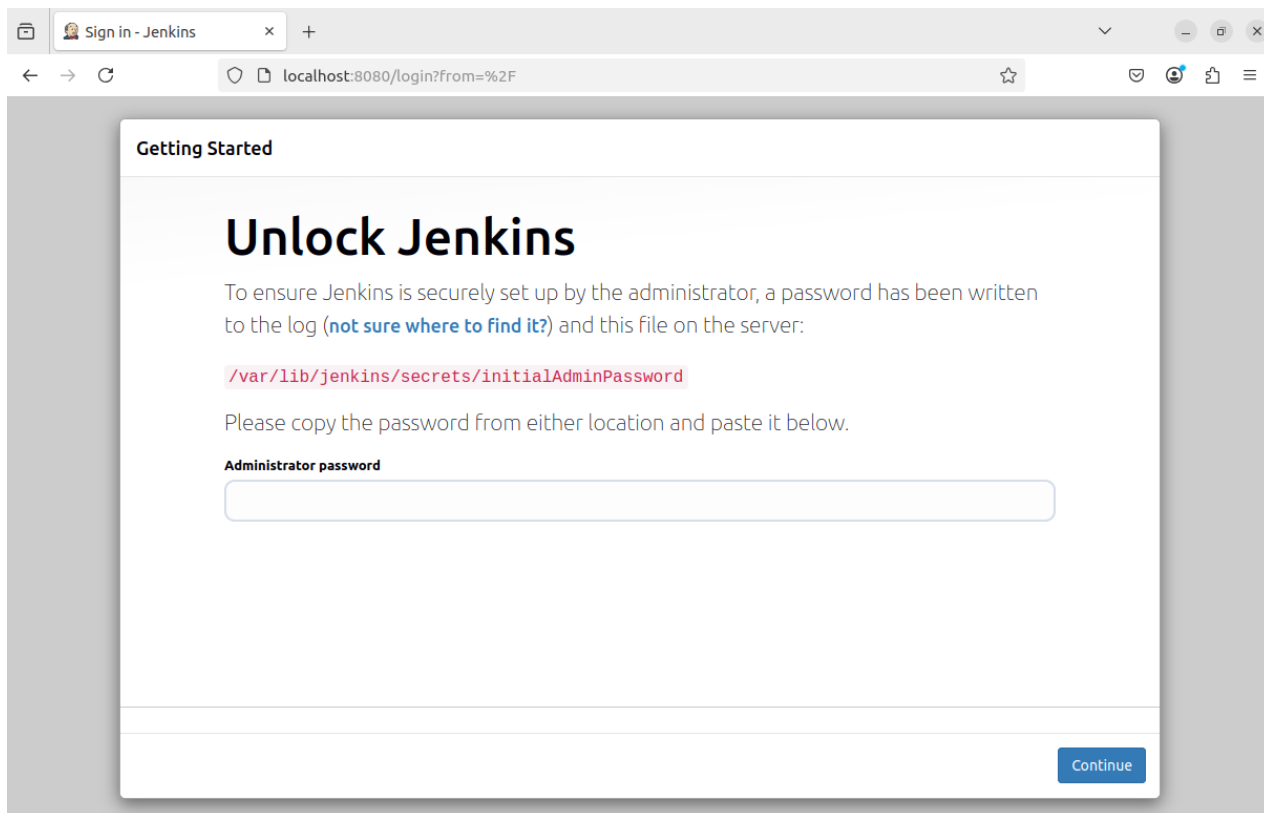
```

мая 11 17:44:41 Vrazhenko jenkins[7517]: 6bca3b5711234a6b9d1def5d06084e5d
мая 11 17:44:41 Vrazhenko jenkins[7517]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
мая 11 17:44:41 Vrazhenko jenkins[7517]: *****
мая 11 17:44:41 Vrazhenko jenkins[7517]: *****
мая 11 17:44:41 Vrazhenko jenkins[7517]: *****
мая 11 17:45:04 Vrazhenko jenkins[7517]: 2025-05-11 14:45:04.462+0000 [id=39] INFO jenkins.InitReactorRunner$1#onAttac
мая 11 17:45:04 Vrazhenko jenkins[7517]: 2025-05-11 14:45:04.496+0000 [id=24] INFO hudson.lifecycle.Lifecycle#onReady
мая 11 17:45:04 Vrazhenko systemd[1]: Started Jenkins Continuous Integration Server.
мая 11 17:45:13 Vrazhenko jenkins[7517]: 2025-05-11 14:45:13.368+0000 [id=60] INFO h.m.DownloadService$Downloadable#l
мая 11 17:45:13 Vrazhenko jenkins[7517]: 2025-05-11 14:45:13.369+0000 [id=60] INFO hudson.util.Retrier#start: Perform
lines 1-20/20 (END)

```

Рисунок 40 - Запуск Jenkins и его статус

Разблокируем Jenkins (рис. 41):



**Рисунок 41 - Разблокировка Jenkins**

Получим пароль для разблокировки Jenkins (рис. 42):

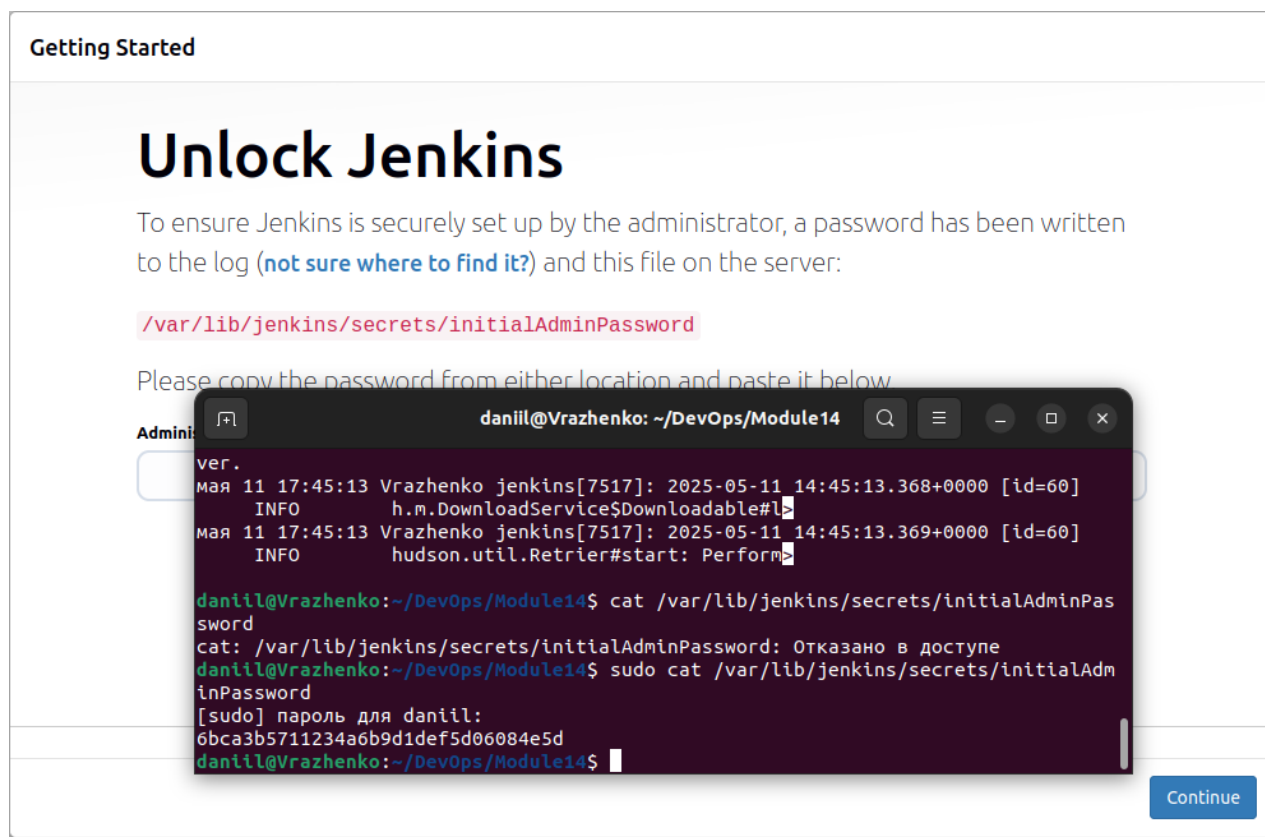


Рисунок 42 - Пароль для Jenkins

Установим плагины для Jenkins (рис. 43-44):

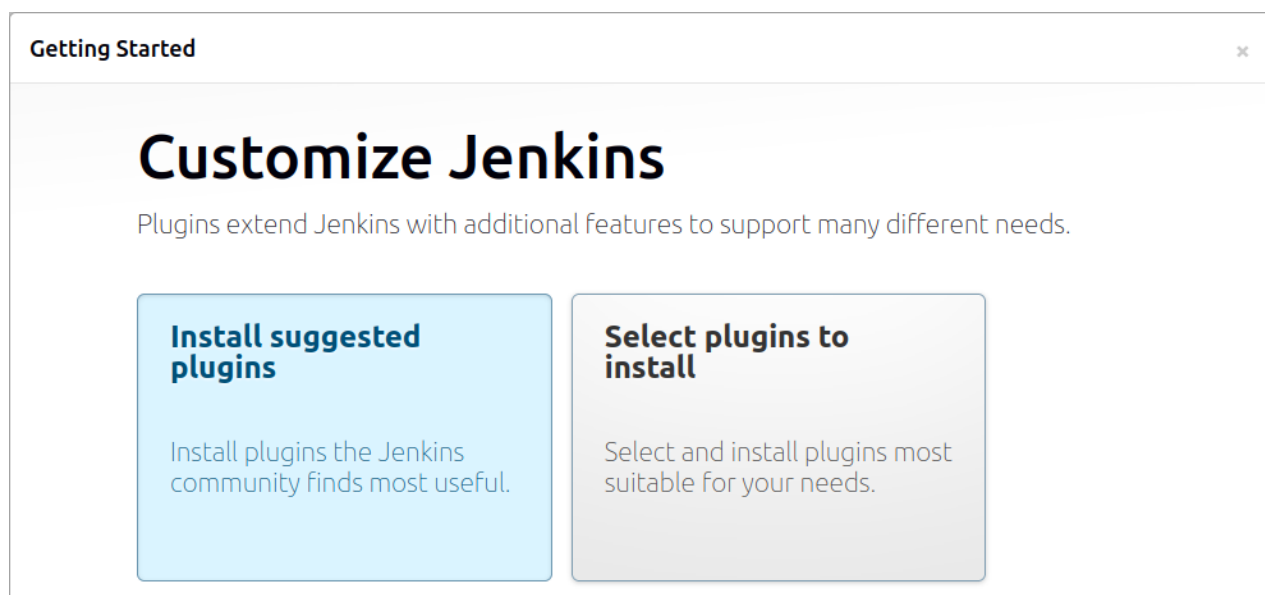


Рисунок 43 - Плагины для Jenkins

# Getting Started

✓ Folders	✓ OWASP Markup Formatter	✓ Build Timeout	⌚ Credentials Binding	** Ionicons API
⌚ Timestamper	⌚ Workspace Cleanup	⌚ Ant	⌚ Gradle	Folders
⌚ Pipeline	⌚ GitHub Branch Source	⌚ Pipeline: GitHub Groovy Libraries	⌚ Pipeline Graph View	OWASP Markup Formatter
⌚ Git	⌚ SSH Build Agents	⌚ Matrix Authorization Strategy	⌚ PAM Authentication	** ASM API
⌚ LDAP	⌚ Email Extension	⌚ Mailer	⌚ Dark Theme	** JSON Path API
				** Struts
				** Pipeline: Step API
				** Token Macro
				Build Timeout
				** bouncycastle API
				** Credentials
				** - required dependency

Рисунок 44 - Установка плагинов Jenkins

Создадим первого пользователя-администратора (рис. 45):

Getting Started

Create First Admin User

Имя пользователя

Daniil

Пароль

....

Повторите пароль

....

Ф.И.О.

Враженко Даниил Олегович

Jenkins 2.504.1

Skip and continue as admin

Save and Continue

Рисунок 45 - Создание первого пользователя-администратора

Видим, что Jenkins готов к использованию (рис. 46):

## Getting Started

# Jenkins is ready!

Your Jenkins setup is complete.

[Start using Jenkins](#)

**Рисунок 46 - Jenkins готов**

## Вывод

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 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.