



**МИНОБРНАУКИ РОССИИ**

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

**«МИРЭА – Российский технологический университет»**

**РТУ МИРЭА**

**Институт кибербезопасности и цифровых технологий**

**Кафедра КБ-4 «Интеллектуальные системы информационной безопасности»**

Дисциплина «Технологии обеспечения информационной безопасности»

**Отчет**

**о проделанной практической работе №5**

Выполнил студент 1 курса

Группы: ББМО-02-24

Дмитриев Д.В.

Проверил

Петров К. Е.

Москва

2024

### ПР3. Сбор логов

1. Создать 2 виртуальные машины на базе ОС Debian 12

<https://www.virtualbox.org/wiki/Downloads>

<https://cdimage.debian.org/debiancd/current/amd64/iso-cd/debian-12.1.0-amd64-netinst.iso>

2. Обеспечить между ними сетевой обмен <https://www.virtualbox.org/manual/ch06.html>

3. Включить на 1й из ВМ передачу логов по протоколу rsyslog на 2ю ВМ

<https://www.tecmint.com/install-rsyslog-centralized-logging-in-centos-ubuntu/>

4. Установить и настроить получение логов на сервер с использованием Loki

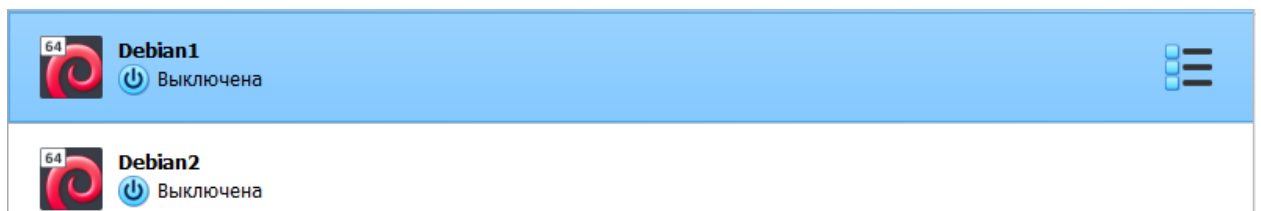
<https://github.com/grafana/loki> [https://docs.google.com/document/d/11tjK\\_lvp1-SVsFZjgOTr1vV3-q6vBAsZYIQ5ZeYBkyM/view](https://docs.google.com/document/d/11tjK_lvp1-SVsFZjgOTr1vV3-q6vBAsZYIQ5ZeYBkyM/view) (источник можно выбрать самостоятельно)

SVsFZjgOTr1vV3- q6vBAsZYIQ5ZeYBkyM/view (источник можно выбрать самостоятельно)

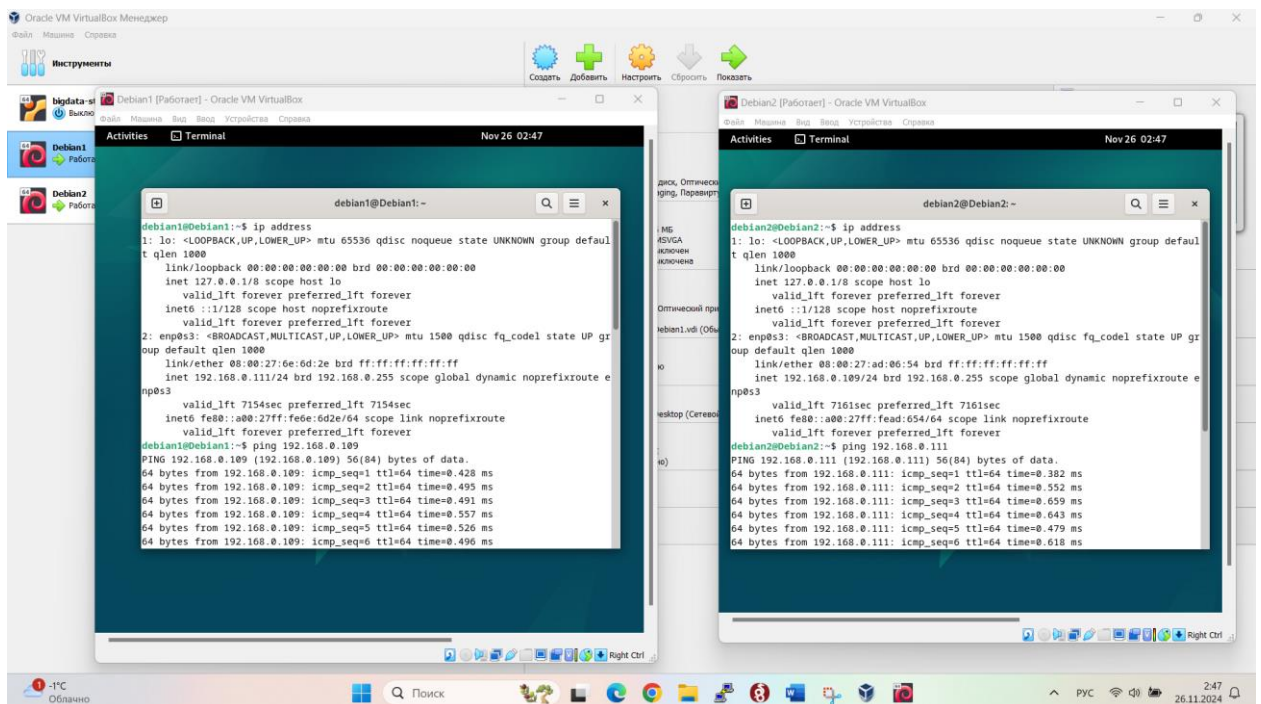
5. Установить и настроить получение логов на сервер с использованием Signoz

<https://signoz.io/> <https://signoz.io/blog/loki-vs-elasticsearch/> (источник можно выбрать самостоятельно)

1. Создаем 2 виртуальные машины на базе ОС Debian 12

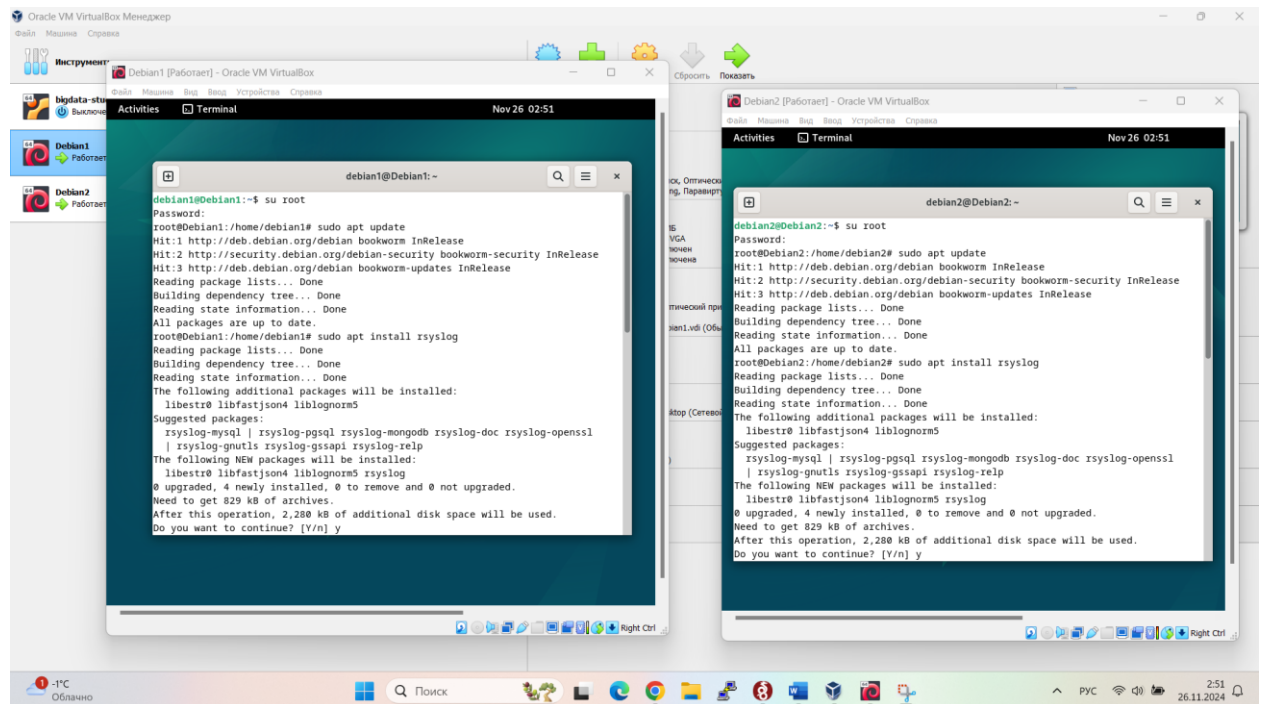


2. Обеспечим между ними сетевой обмен

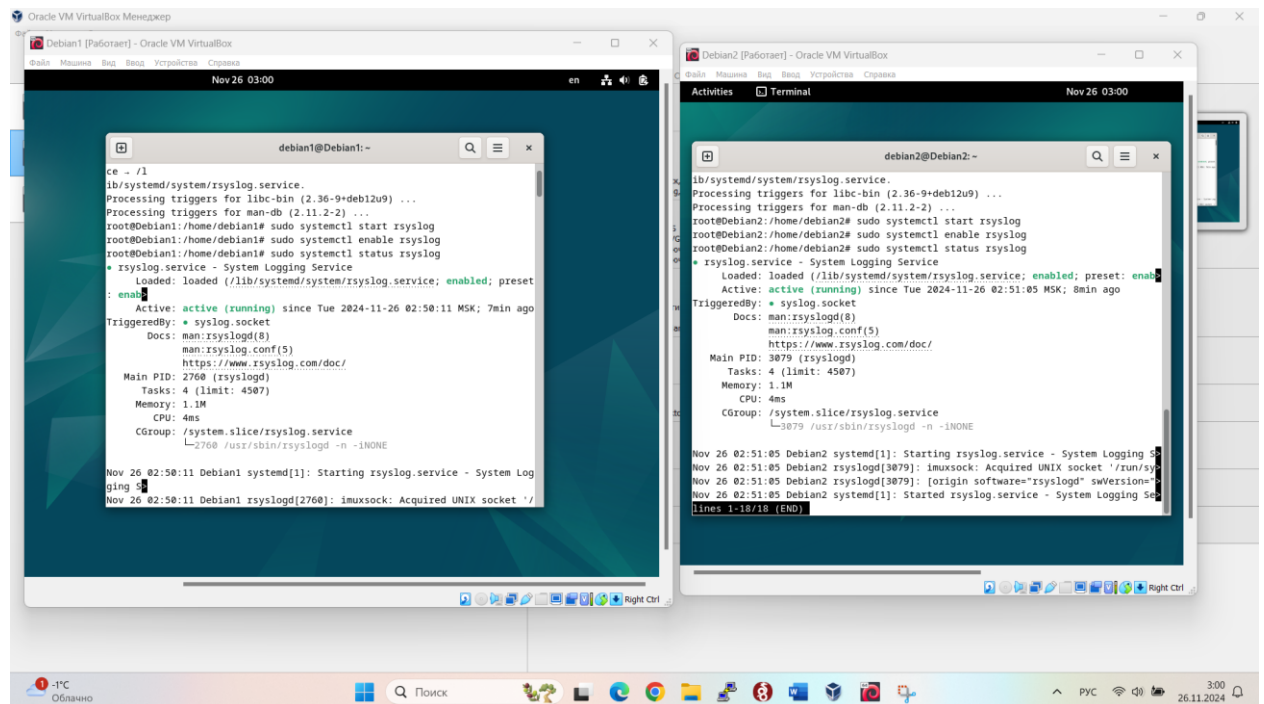


### 3. Включить на 1й ВМ передачу логов по протоколу rsyslog на 2ю ВМ

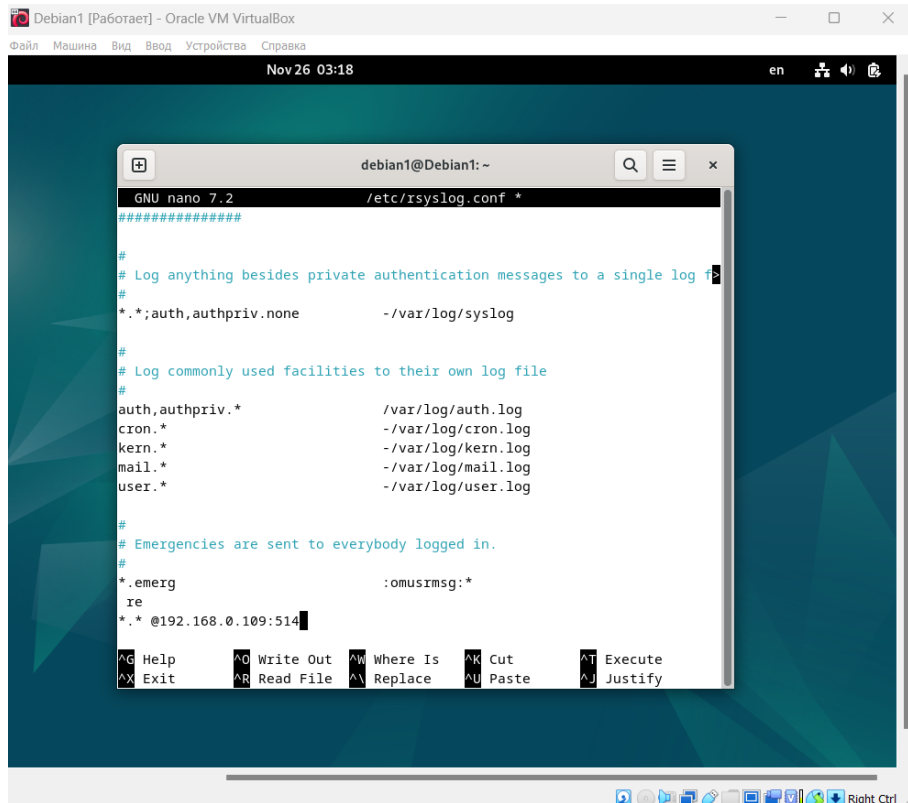
#### 3.1 Установим rsyslog



#### 3.2 Проверим работоспособность rsyslog

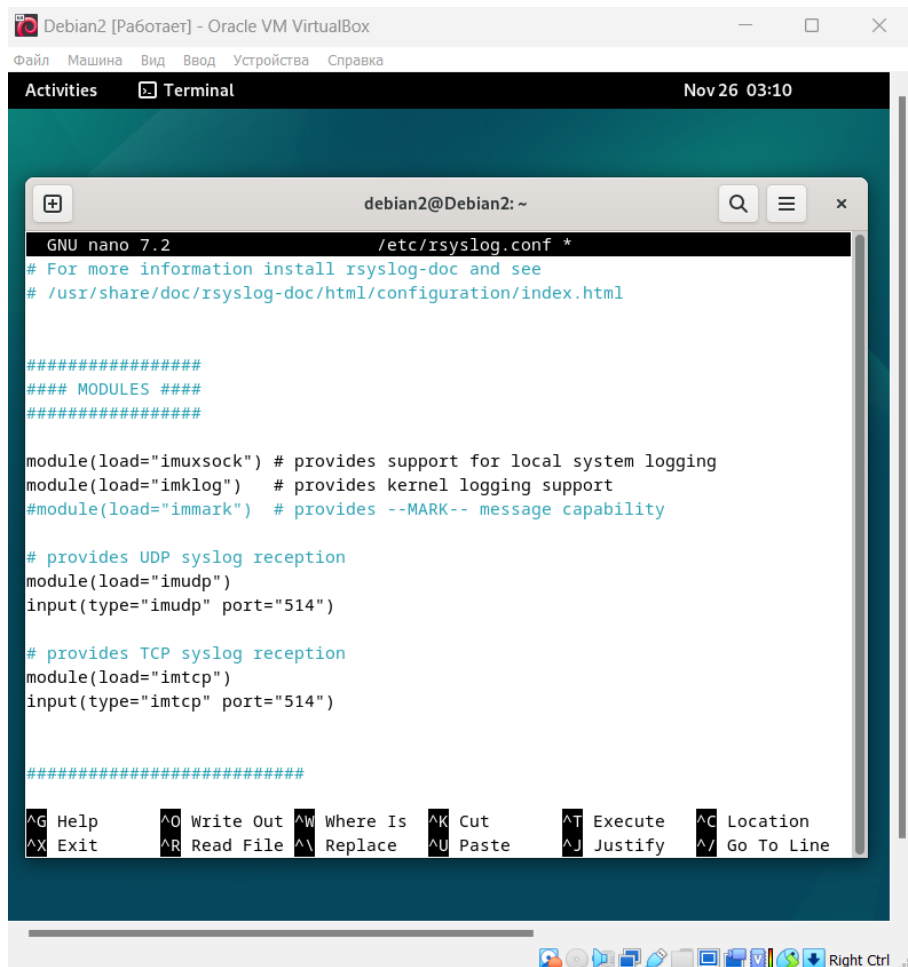


### 3.3 Отредактируем конфигурационный файл на 1й и 2й VM



```
Debian1 [Работает] - Oracle VM VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка
Nov 26 03:18  en  [Speaker]  [Refresh]

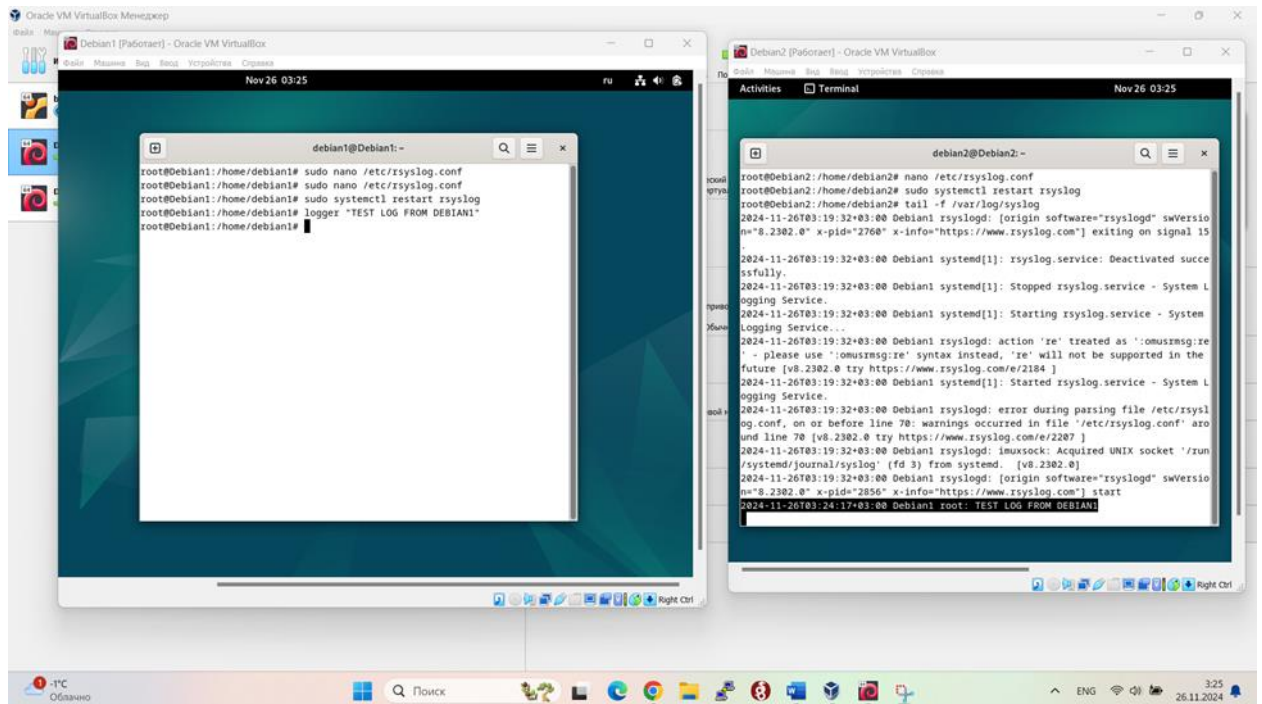
debian1@Debian1: ~
GNU nano 7.2 /etc/rsyslog.conf *
#####
#
# Log anything besides private authentication messages to a single log
#
*.auth,authpriv.none -/var/log/syslog
#
# Log commonly used facilities to their own log file
#
auth,authpriv.* /var/log/auth.log
cron.* /var/log/cron.log
kern.* /var/log/kern.log
mail.* /var/log/mail.log
user.* /var/log/user.log
#
# Emergencies are sent to everybody logged in.
#
*.emerg :omusmsg:*
re
*.* @192.168.0.109:514
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify
```



```
Debian2 [Работает] - Oracle VM VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка
Nov 26 03:10

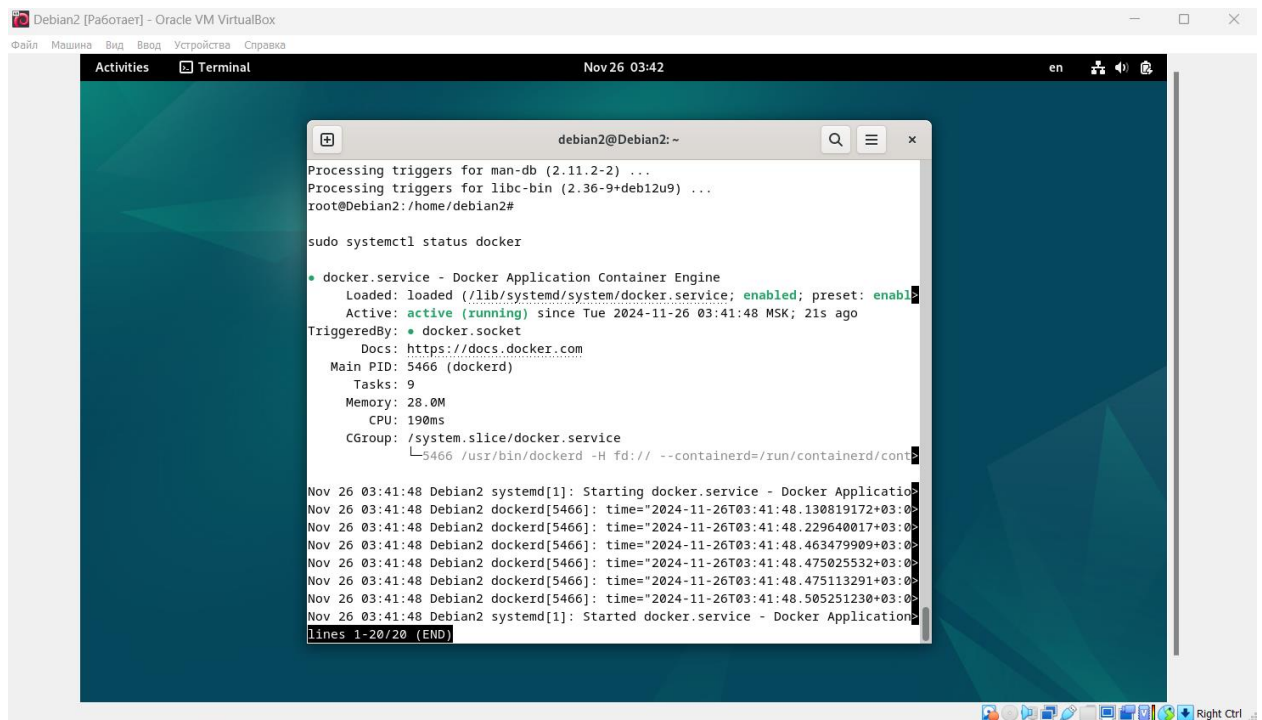
Activities  Terminal
debian2@Debian2: ~
GNU nano 7.2 /etc/rsyslog.conf *
# For more information install rsyslog-doc and see
# /usr/share/doc/rsyslog-doc/html/configuration/index.html
#####
### MODULES ###
#####
module(load="imuxsock") # provides support for local system logging
module(load="imklog") # provides kernel logging support
#module(load="immark") # provides --MARK-- message capability
# provides UDP syslog reception
module(load="imudp")
input(type="imudp" port="514")
# provides TCP syslog reception
module(load="imtcp")
input(type="imtcp" port="514")
#####
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^_ Go To Line
```

### 3.4 Проверка получения логов

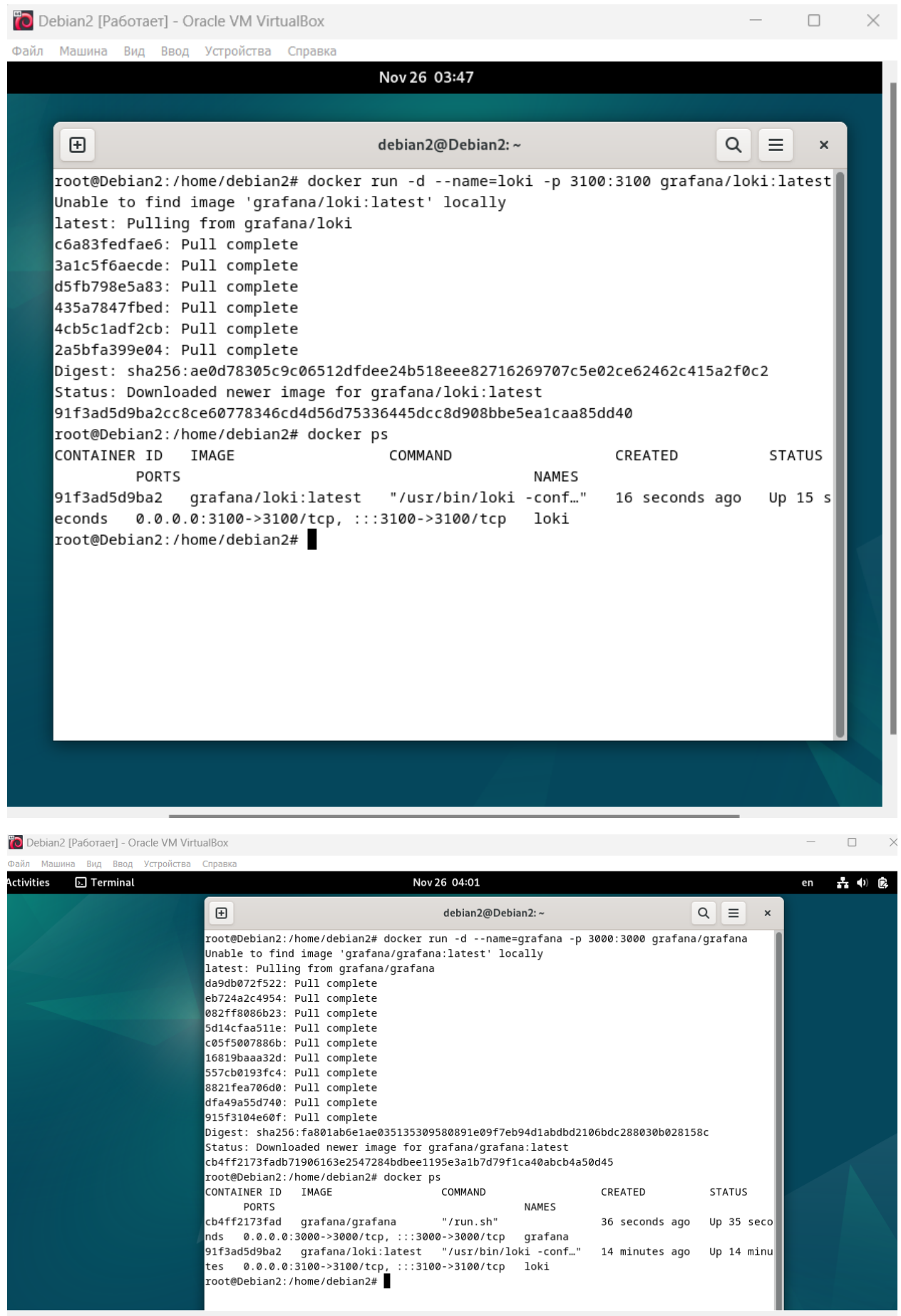


## 4. . Установить и настроить получение логов на сервер с использованием Loki

### 4.1 Установим docker и проверим его работоспособность

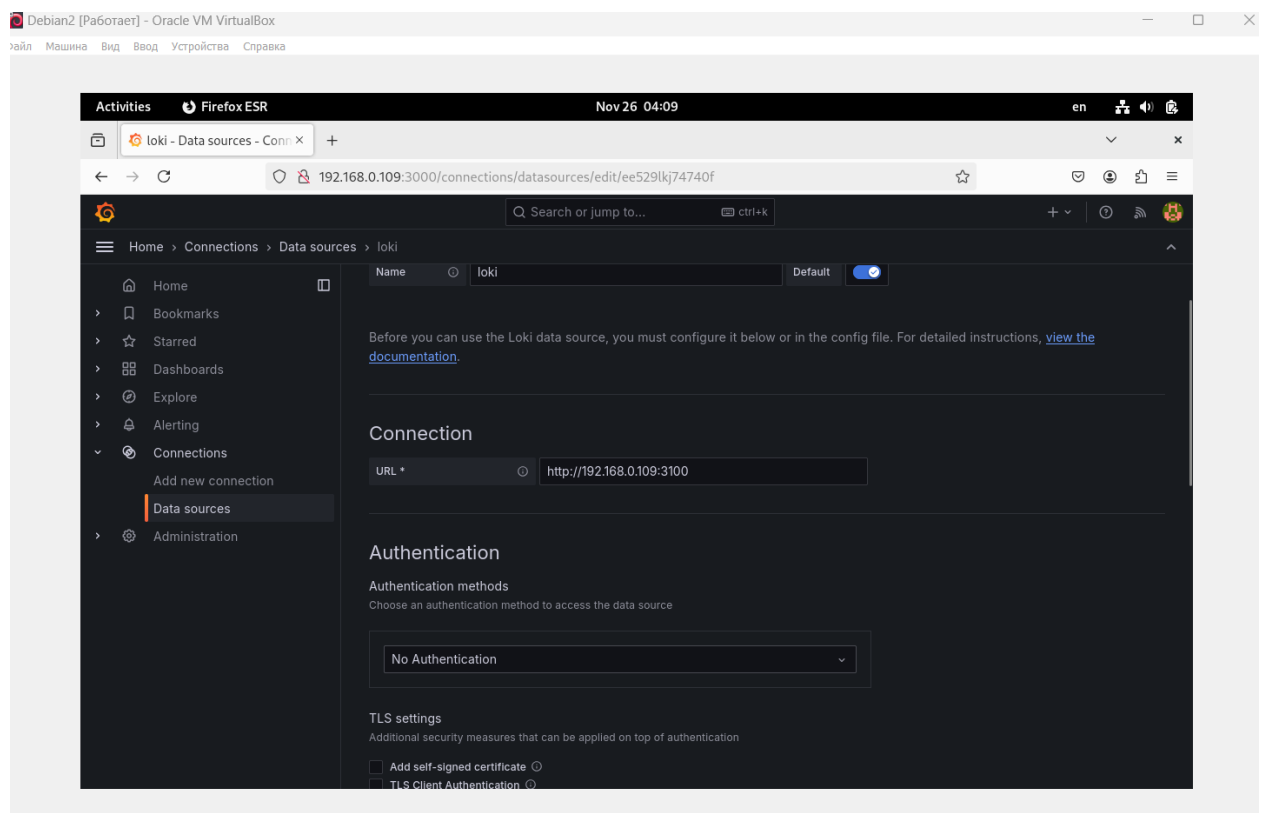
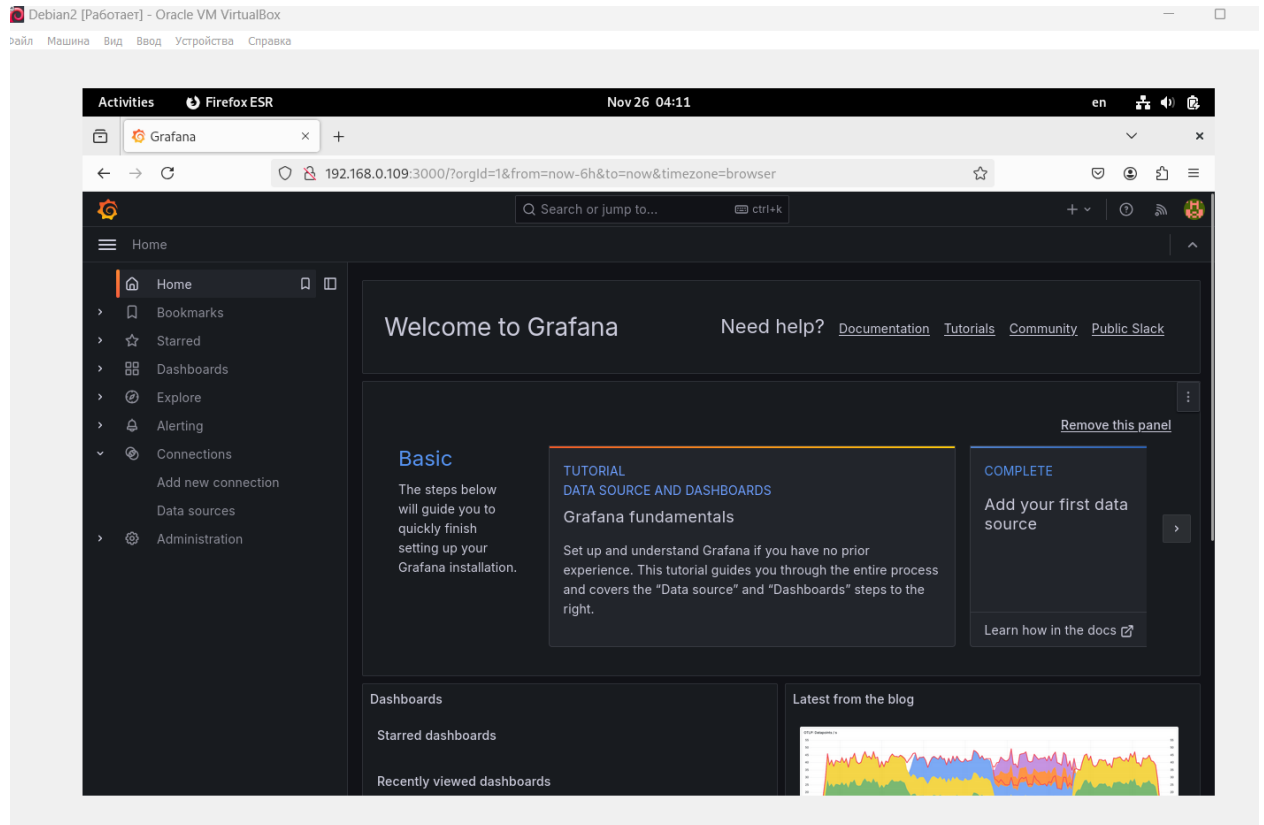


## 4.2 Запускаем контейнеры





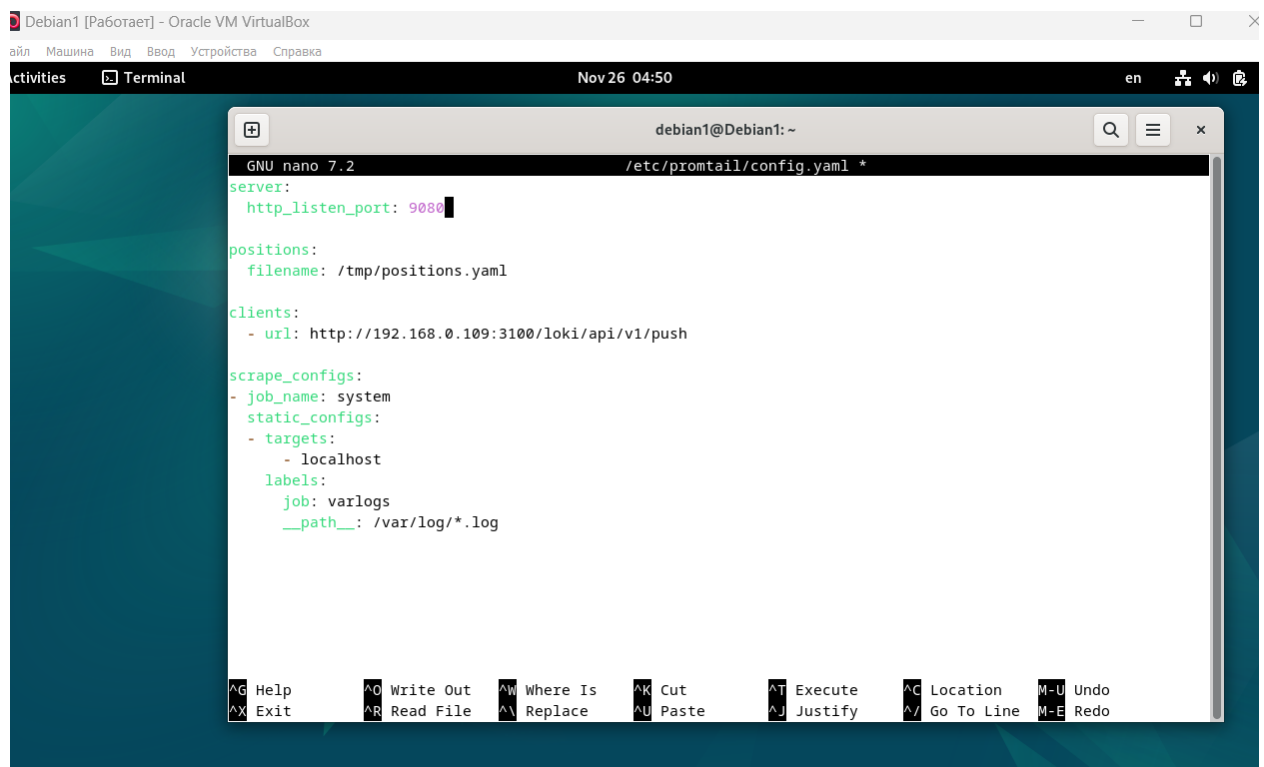
### 4.3 Заходим на web-интерфейс Grafana и добавляем в Grafana связку с Loki:



✓ Data source successfully connected.

Next, you can start to visualize data by [building a dashboard](#), or by querying data in the [Explore view](#).

#### 4.4 Для передачи логов устанавливаем утилиту Promtail и редактируем его конфигурационный файл



#### 4.5 Запускаем утилиту Promtail

```
root@Debian1:/home/debian1# touch /var/log/test.log
root@Debian1:/home/debian1# echo "THIS IS A TEST12 FOR DEBIAN1" | sudo tee -a /var/log/test.log
THIS IS A TEST12 FOR DEBIAN1
root@Debian1:/home/debian1# cat /var/log/test.log
This is a test_2 log entry
THIS IS A TEST FOR DEBIAN1
THIS IS A TEST123 FOR DEBIAN1
THIS IS A TEST12 FOR DEBIAN1
root@Debian1:/home/debian1# promtail -config.file=/etc/promtail/config.yaml
level=info ts=2024-11-26T02:36:38.732232129Z caller=promtail.go:133 msg="Reloading configuration file" md5sum=c8adb5ca4a231b0bd4f2093e13075c0d
level=info ts=2024-11-26T02:36:38.732895757Z caller=server.go:322 http[::]:9080 grpc[::]:9095 msg="server listening on addresses"
level=info ts=2024-11-26T02:36:38.732999804Z caller=main.go:174 msg="Starting Promtail" version="(version=2.9.4, branch=HEAD, revision=f599ebc535)"
level=warn ts=2024-11-26T02:36:38.733040271Z caller=promtail.go:263 msg="enable watchConfig"
level=info ts=2024-11-26T02:36:43.734095498Z caller=filetargetmanager.go:361 msg="Adding target" key="/var/log/test.log:{job=\"varlogs\"}"
level=info ts=2024-11-26T02:36:43.734155014Z caller=filetarget.go:313 msg="watching new directory" directory=/var/log
ts=2024-11-26T02:36:43.734251646Z caller=log.go:168 level=info msg="Seeked /var/log/test.log - &{Offset:84 Whence:0}"
level=info ts=2024-11-26T02:36:43.734272654Z caller=tailer.go:145 component=tailer msg="tail routine: started" path=/var/log/test.log
```



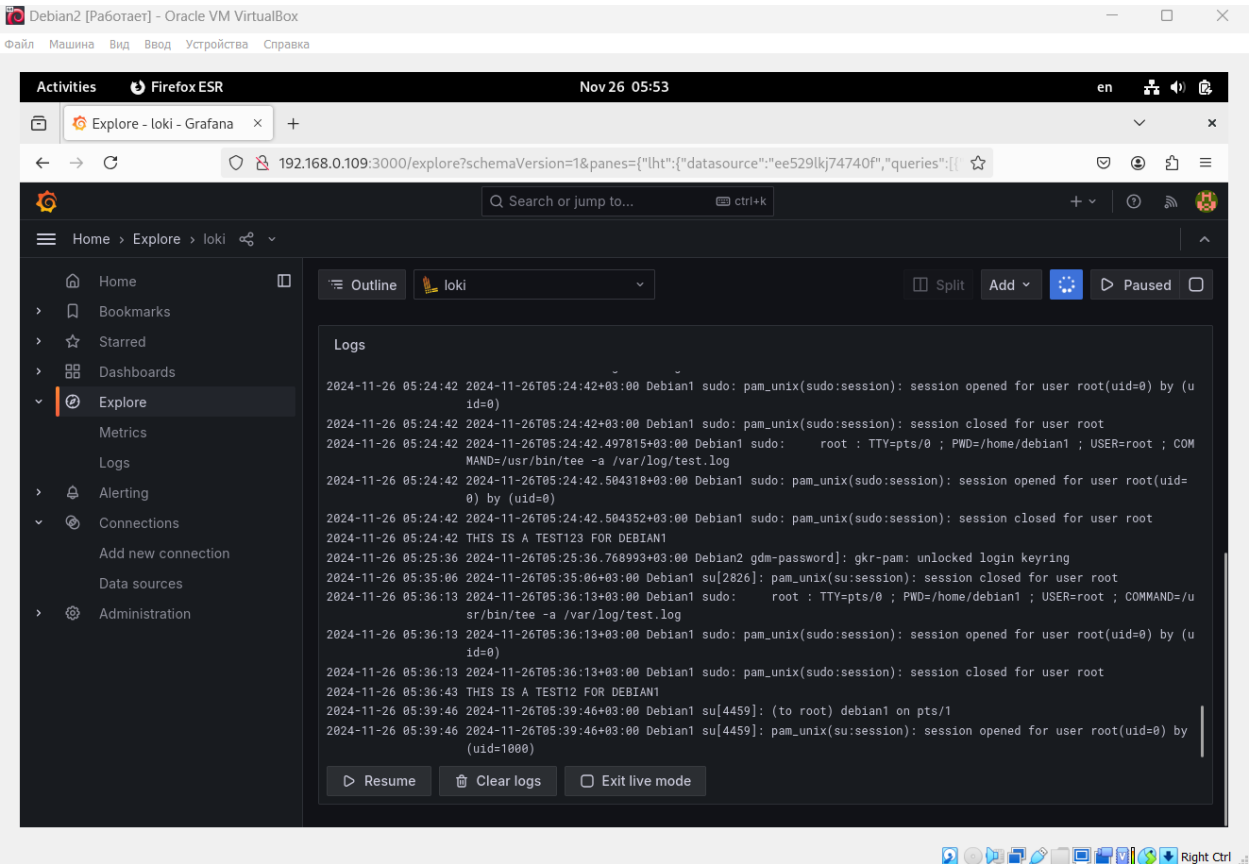
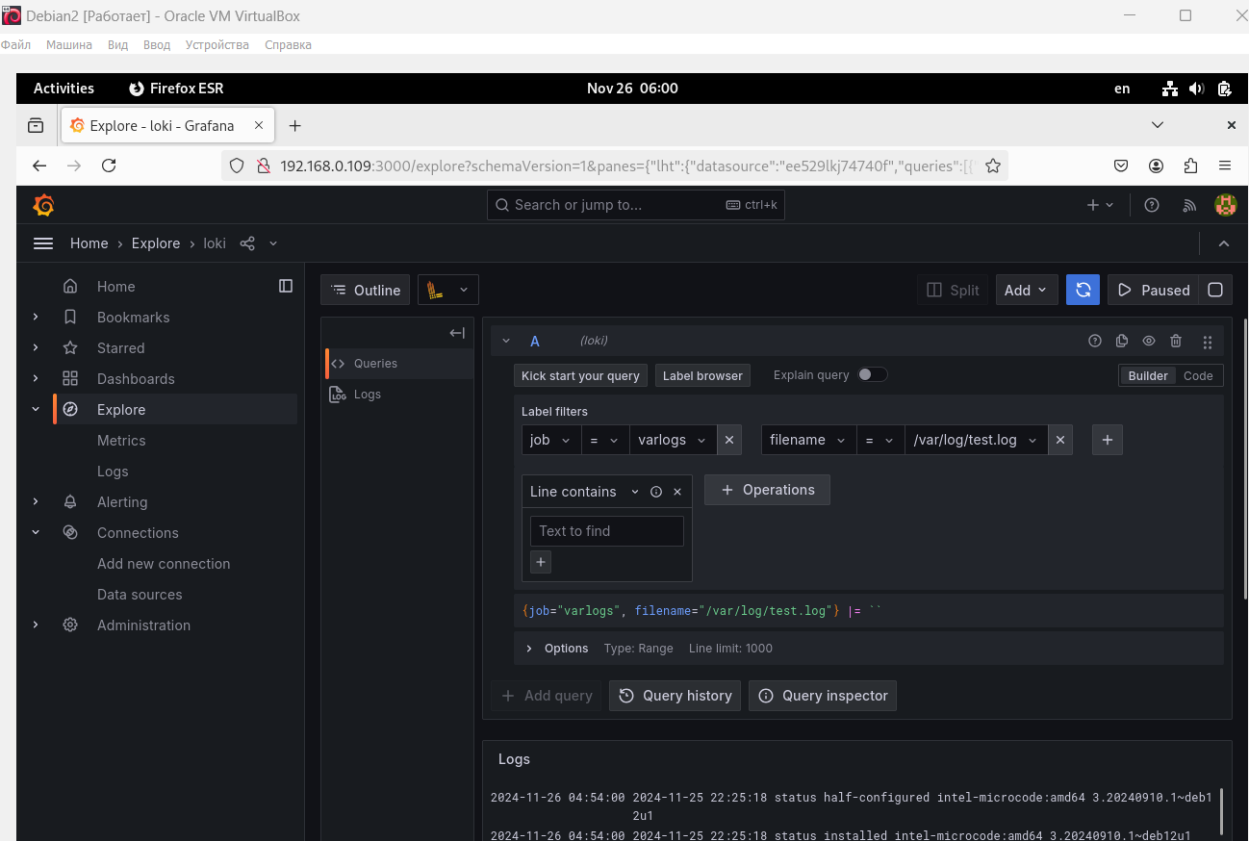
4.6 В файле /tmp/positions.yaml указаны позиции файлообменников для логов, которые будут отправляться с 1й ВМ на 2ю ВМ .

```
debian1@Debian1: ~  
root@Debian1:/home/debian1# cat /tmp/positions.yaml  
positions:  
  /var/log/alternatives.log: "47822"  
  /var/log/auth.log: "4470"  
  /var/log/boot.log: "55335"  
  /var/log/cron.log: "356"  
  /var/log/dpkg.log: "828324"  
  /var/log/fontconfig.log: "5463"  
  /var/log/kern.log: "61196"  
  /var/log/test.log: "27"  
  /var/log/user.log: "633"  
  /var/log/vboxpostinstall.log: "20423"  
root@Debian1:/home/debian1#
```

4.7 На ВМ2 проверяем готовность Loki

```
debian2@Debian2: ~  
root@Debian2:/home/debian2# curl http://localhost:3100/ready  
Ingester not ready: waiting for 15s after being ready  
root@Debian2:/home/debian2# curl http://localhost:3100/ready  
ready  
root@Debian2:/home/debian2#
```

## 4.8 Просматриваем логи от 1й VM



5. Установить и настроить получение логов на сервер с использованием Signoz

5.1 Устанавливаем docker-compose и signoz на VM2 после чего запускаем контейнеры командой **docker-compose up -d**

```
[+] Running 59/11ayers [██████████] 123.4MB/268.8MB Pulling
✓ otel-collector-migrator-sync 2 layers [██] 0B/0B Pulled 11.8s
✓ otel-collector-migrator-async Pulled 11.8s
✓ hotrod 1 layers [██] 0B/0B Pulled 84.4s
✓ load-hotrod 8 layers [██████████] 0B/0B Pulled 39.4s
✓ zookeeper-1 1 layers [██] 0B/0B Pulled 90.4s
✓ clickhouse 7 layers [██████████] 0B/0B Pulled 62.7s
✓ alertmanager 6 layers [██████████] 0B/0B Pulled 76.5s
✓ frontend 12 layers [██████████] 0B/0B Pulled 11.6s
✓ otel-collector 3 layers [██] 0B/0B Pulled 92.4s
✓ logspout 3 layers [██] 0B/0B Pulled 54.3s
✓ query-service 5 layers [██████] 0B/0B Pulled 61.2s

[+] Running 12/12 65.7s
✓ Network clickhouse-setup_default Cre... 0.1s
✓ Container load-hotrod Started 0.7s
✓ Container signoz-zookeeper-1 Started 0.7s
✓ Container hotrod Started 0.7s
✓ Container signoz-clickhouse Healthy 65.7s
✓ Container otel-migrator-sync Exited 65.7s
✓ Container signoz-query-service Healt... 95.6s
✓ Container otel-migrator-async Starte... 65.0s
✓ Container signoz-otel-collector Star... 96.0s
✓ Container signoz-alertmanager Starte... 96.0s
✓ Container signoz-frontend Started 96.4s
✓ Container signoz-logspout Started 96.3s

root@Debian2: /usr/signoz/deploy/docker/clickhouse-setup# █

root@Debian2: /usr/signoz/deploy/docker/clickhouse-setup# docker ps
CONTAINER ID   IMAGE                                     COMMAND                                     CREATED        STATUS        PORTS
0335c2ce8f52   gliderlabs/logspout:v3.2.14             "/bin/logspout syslo..." 3 minutes ago  Up 2 minutes  80/tcp
out
7a71f45df924   signoz/frontend:0.59.0                  "nginx -g 'daemon of..." 3 minutes ago  Up 2 minutes  80/tcp, 0.0.0
.0:3301->3301/tcp, :::3301->3301/tcp
end
b5546b3d9a7c   signoz/signoz-otel-collector:0.111.9    "/signoz-collector -..." 3 minutes ago  Up 2 minutes  0.0.0.0:4317-
4318->4317-4318/tcp, :::4317-4318->4317-4318/tcp
collector
22457791b5cf   signoz/alertmanager:0.23.7              "/bin/alertmanager -..." 3 minutes ago  Up 2 minutes  9093/tcp
manager
f415cdcf553    signoz/query-service:0.59.0              "./query-service -co..." 3 minutes ago  Up 2 minutes  (healthy) 8080/tcp
-service
aedb9775efea   clickhouse/clickhouse-server:24.1.2-alpine "/entrypoint.sh"          3 minutes ago  Up 3 minutes  (healthy) 0.0.0.0:8123-
>8123/tcp, :::8123->8123/tcp, 0.0.0.0:9000->9000/tcp, :::9000->9000/tcp, 0.0.0.0:9181->9181/tcp, :::9181->9181/tcp, 9009/tcp
house
cdfb6f02bcf9   bitnami/zookeeper:3.7.1                 "/opt/bitnami/script..." 3 minutes ago  Up 3 minutes  0.0.0.0:2181-
>2181/tcp, :::2181->2181/tcp, 0.0.0.0:2888->2888/tcp, :::2888->2888/tcp, 0.0.0.0:3888->3888/tcp, :::3888->3888/tcp, 8080/tcp
eeper-1
fa323c972ec3   signoz/locust:1.2.3                      "/docker-entrypoint..." 3 minutes ago  Up 3 minutes  5557-5558/tcp
, 8089/tcp
d131721a7f2a   jaegertracing/example-hotrod:1.30        "/go/bin/hotrod-linu..." 3 minutes ago  Up 3 minutes  8080-8083/tcp
hotrod

root@Debian2: /usr/signoz/deploy/docker/clickhouse-setup# █
```

1. signoz-zookeeper:

Служит для координации и управления распределенными компонентами (используется для Kafka).

2. signoz-alertmanager:

Управляет уведомлениями и алертами (например, когда происходят сбои).

3. signoz-otel-collector:

Средство для сбора метрик, логов и трасс; перенаправляет данные в основной хранилище.

4. signoz-logspout:

Собирает контейнерные логи с Docker и направляет их в Signoz.

5. signoz-query-service:

Выполняет запросы к базе (ClickHouse), отвечает за доставку данных в интерфейс.

6. signoz-clickhouse:

Хранилище данных (метрики, трассировки и логи сохраняются здесь).

7. signoz-frontend:

Интерфейс для пользователей — отображает метрики, трассы и логи.

8. load-hotrod (Locust):

Генератор нагрузки для тестирования (используется для демонстраций).

9. hotrod:

Пример микросервиса, чтобы демонстрировать работу трассировки в Signoz.

5.2 Устанавливаем утилиту **OpenTelemetry**. Она предоставляет инструменты и стандарты для сбора, обработки и экспорта данных о телеметрии (метрики, трассировки и логи) в системы мониторинга и анализа.

## 5.3 Редактируем конфигурационный файл



```
Debian1 [Работает] - Oracle VM VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка
Activities  Terminal  Nov 26 15:17  en  [icons]
debian1@Debian1: ~
GNU nano 7.2  /etc/otel-collector-config.yaml *
receivers:
  otlp:
    protocols:
      grpc:
        #endpoint: 0.0.0.0:4317
      http:
        #endpoint: 0.0.0.0:4318

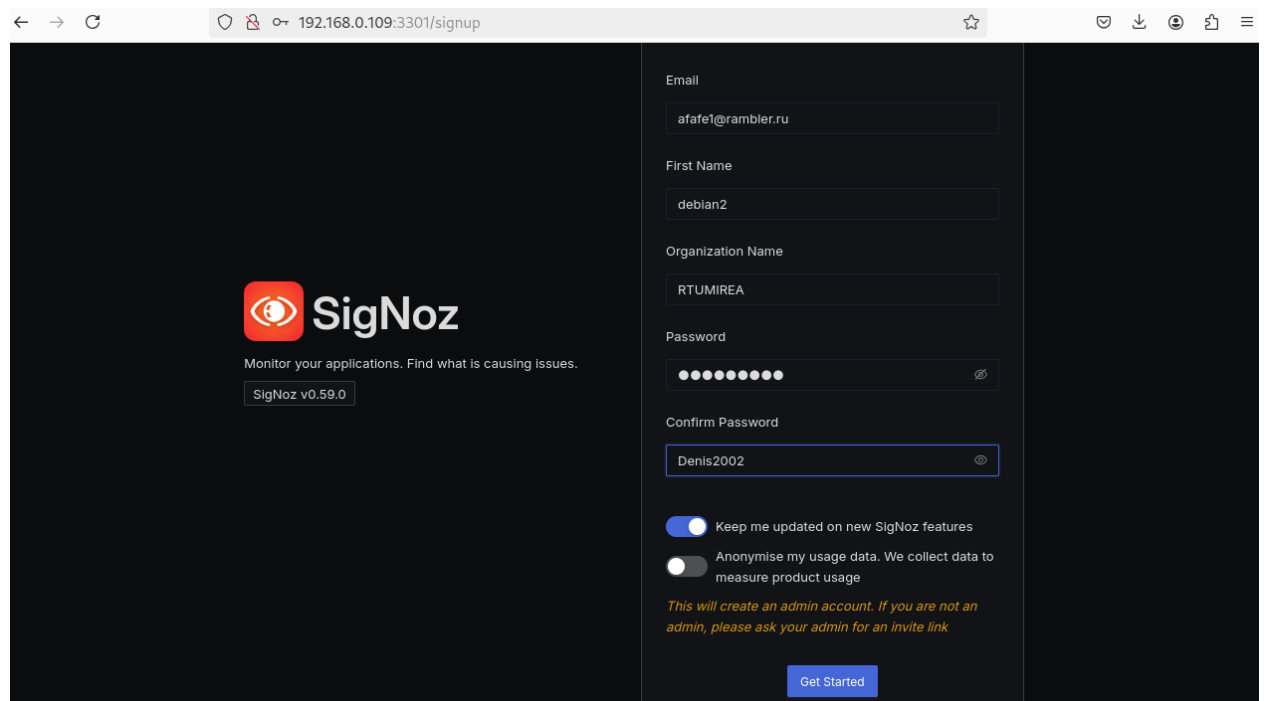
#processors:
# batch:

exporters:
  #logging:
  # logLevel: debug
  otlp:
    endpoint: 192.168.0.109:4317
    tls:
      insecure: true


service:
  pipelines:
    traces:
      receivers: [otlp]
      processors: []
      exporters: [otlp]
    #metrics:
    # receivers: [otlp]
    # processors: [batch]
    # exporters: [logging, otlp]
  logs:
```

## 5.4 запускаем утилиту командой `otelcol-contrib --config=/etc/otel-collector-config.yaml`

## 5.5 Заходим в web-интерфейс Signoz и регистрируемся



← → ↺ 192.168.0.109:3301/signup ☆ [icons]

**Signoz**  
Monitor your applications. Find what is causing issues.  
Signoz v0.59.0

Email

First Name

Organization Name

Password

Confirm Password

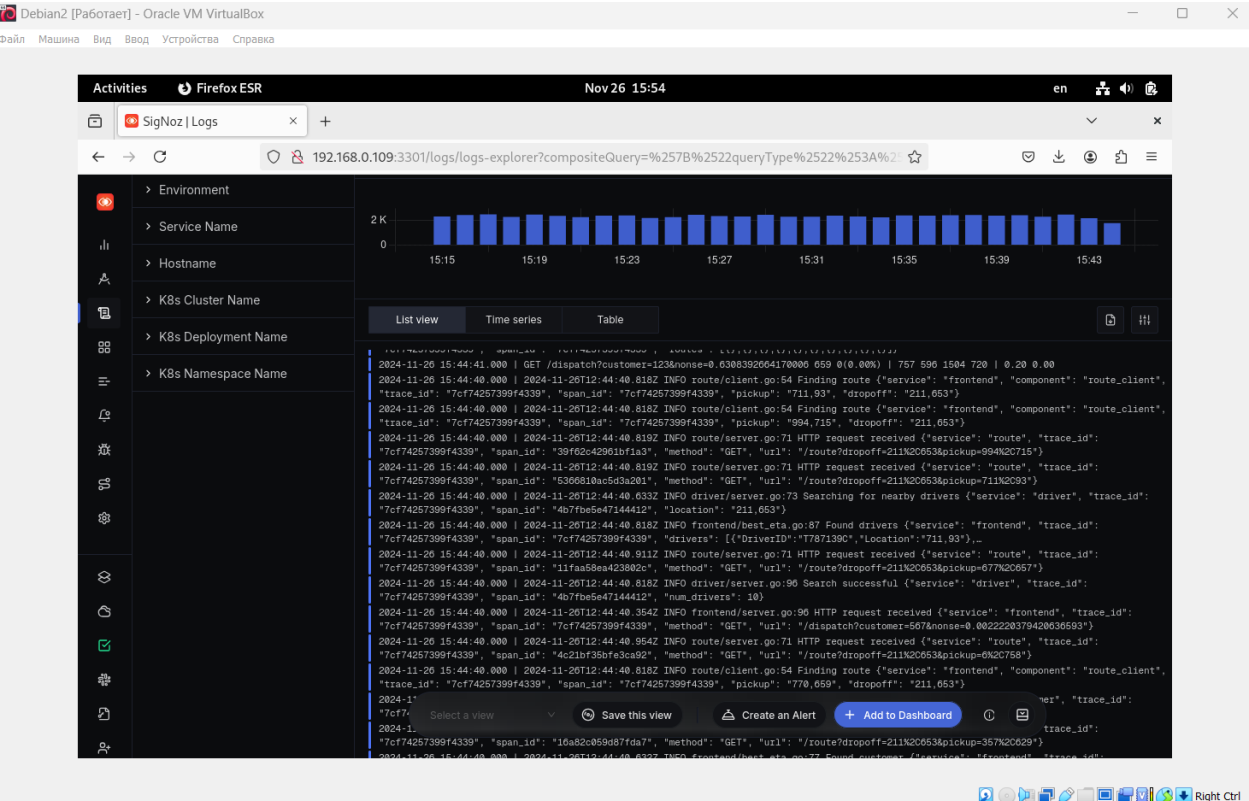
☒ Keep me updated on new Signoz features

☐ Anonymise my usage data. We collect data to measure product usage

*This will create an admin account. If you are not an admin, please ask your admin for an invite link*

[Get Started](#)

## 5.6 Проверка работоспособности



Вывод:

**Loki** - это система для сбора, хранения и обработки логов, разработанная компанией **Grafana**. Она предназначена для работы с большими объемами логов, обеспечивая их эффективное хранение, поиск и визуализацию. **Promtail** - это утилита сбора логов, которая используется в связке с **Loki** для получения, обработки и отправки логов в систему **Loki**.

**Signoz** - система мониторинга и анализа производительности, которая предоставляет инструменты для сбора, хранения и визуализации телеметрических данных (включая метрики, логи и трассировки) для приложений (**SigNoz** собирает данные через **OpenTelemetry**)