МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение

высшего образования

**«Северный (Арктический) федеральный университет имени М. В. Ломоносова»**

Высшая школа информационных систем и технологий

**Отчет о лабораторном практикуме**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| По модулю | | Облачные технологии | | |
|  | | | | |
|  | | | | |
| На тему | Настройка балансировщика нагрузки NGINX | | | |
|  | | | | |
|  | | | Выполнил обучающийся:  Лушев Никита Сергеевич |
|  | | |  |
|  | | | Направление подготовки / специальность:  09.03.01 Информатика и вычислительная техника |
|  | | |  |
|  | | | Курс: 3 |
|  | | | Группа: 151219 |
|  | | | Руководитель:  А. П. Тарасов |
|  | | |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Отметка о зачете |  |  |  |  |
|  |  | (отметка прописью) |  | (дата) |
| Руководитель |  |  |  | А. П. Тарасов |
|  |  | (подпись руководителя) |  | (инициалы, фамилия) |

Архангельск 2025

Цель работы

Цель работы: Установить операционные системы, добиться их связности, установить на хосты веб-сервер nginx, выполнить настройку балансировщика.

Основные задачи:

* Установить Ubuntu на каждый из трёх узлов;
* Установить на сервера веб-сервер nginx;
* Выполнить настройку балансировщика и проверить его работоспособность.

ОГЛАВЛЕНИЕ

[Цель работы 2](#_Toc1)

[1 РАЗВЕРТЫВАНИЕ СЕРВЕРА PROXMOX VE 5](#_Toc2)

[1.1 Установка alt virtualization 5](#_Toc3)

[1.2 Создание нод Proxmox VE 10](#_Toc4)

[1.3 Разрешение ролей для хранилища 18](#_Toc5)

[1.4 Установка первой ноды кластером 19](#_Toc6)

[2 НАСТРОЙКА ВИРТУАЛЬНОГО КОММУТАТОРА 20](#_Toc7)

[2.1 Создание виртуального коммутатора 20](#_Toc8)

[3 Контрольные вопросы 21](#_Toc9)

[3.1 Что такое Proxmox? 21](#_Toc10)

[3.2 Чем отличается Proxmox от OpenStack? 21](#_Toc11)

[3.3 Какие компоненты входят в Proxmox? 21](#_Toc12)

[4 Вывод 23](#_Toc13)

# **УСТАНОВКА И НАСТРОЙКА СЕРВЕРОВ**

## **Установка ubuntu**

Создадим виртуальную машину с именем «balancer», тип гостевой операционной системы - «Ubuntu». Объём оперативной памяти установим «6144 Мб», а количетсво процессоров поставим на «3» (рисунок 1).

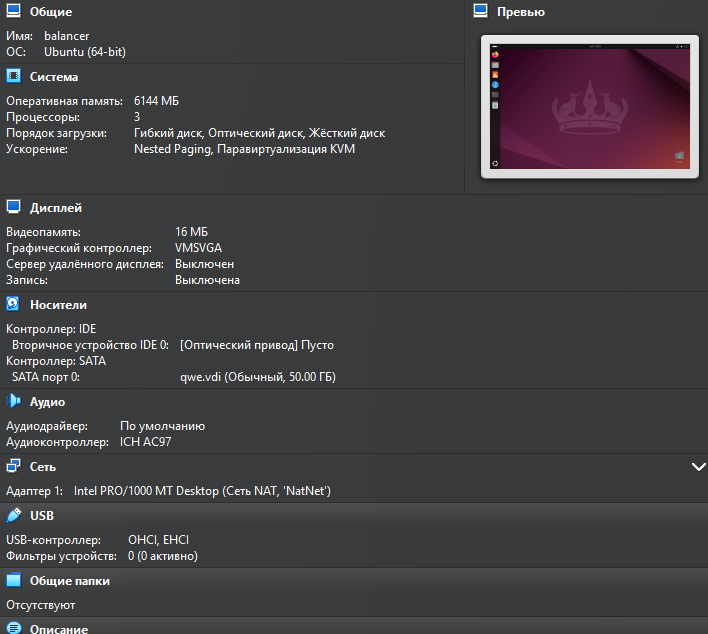


Рисунок 1 – Имя и ОС виртуальной машины

После этого склонируем машину и присвоим клонам названия «web1» и «web2» (рисунок 2).

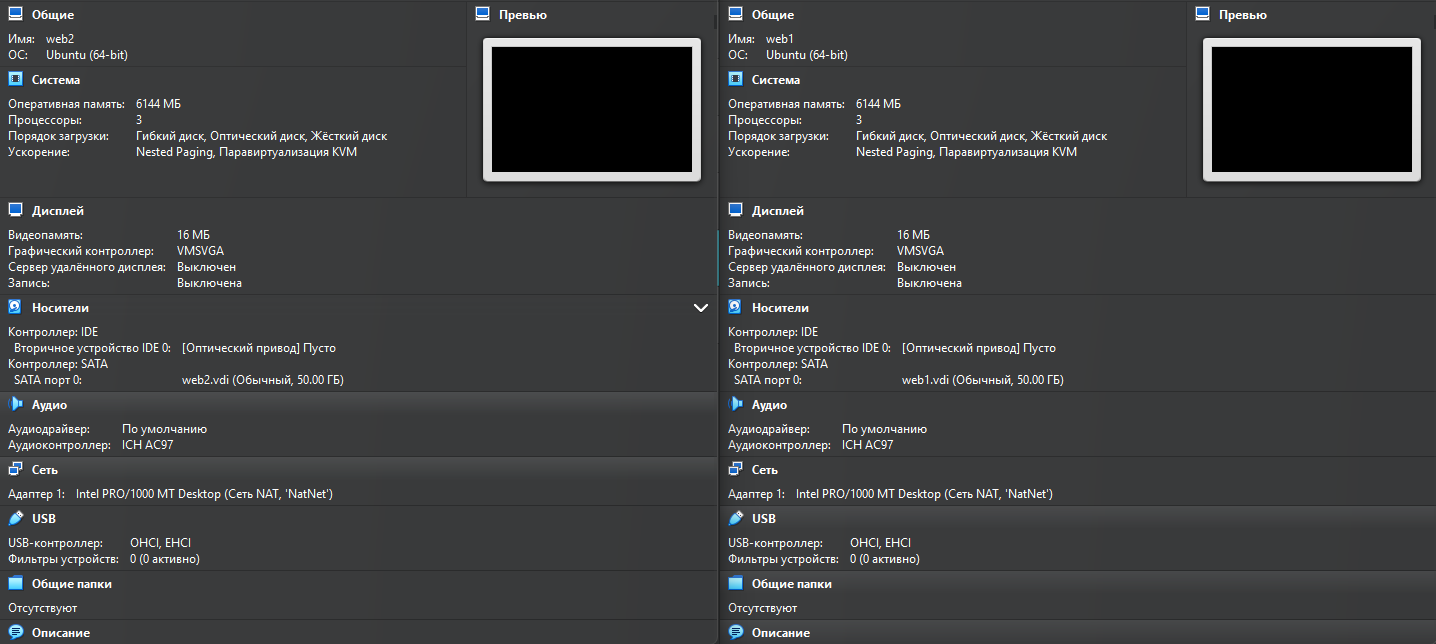


Рисунок 2 – Клонирование машины

Далее из стандартной установки следует упомянуть шаг 9, на котором мы настраиваем пользователя ubuntu (рисунок 3).

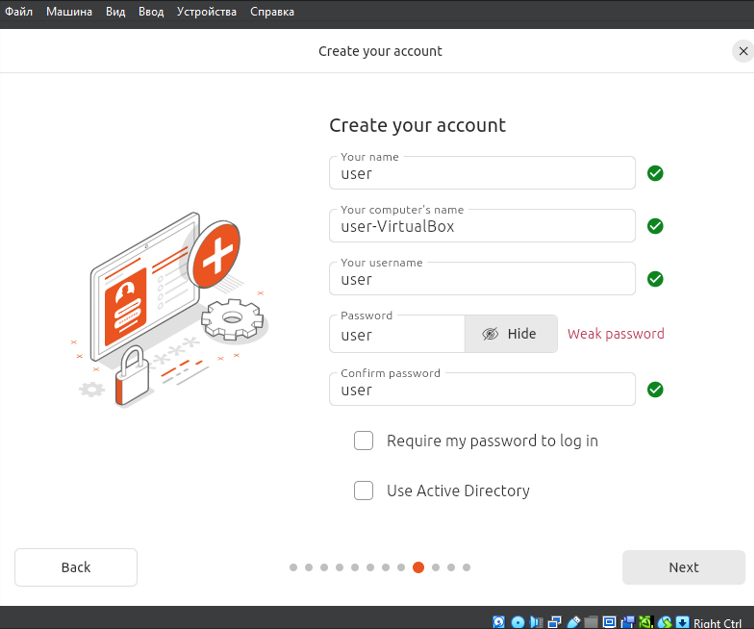


Рисунок 3 – Настройка пользователя

После установки системы на сервера, нужно зайти на каждый сервер, обновить репозитории и установить веб-сервер «nginx» (рисунок 7).

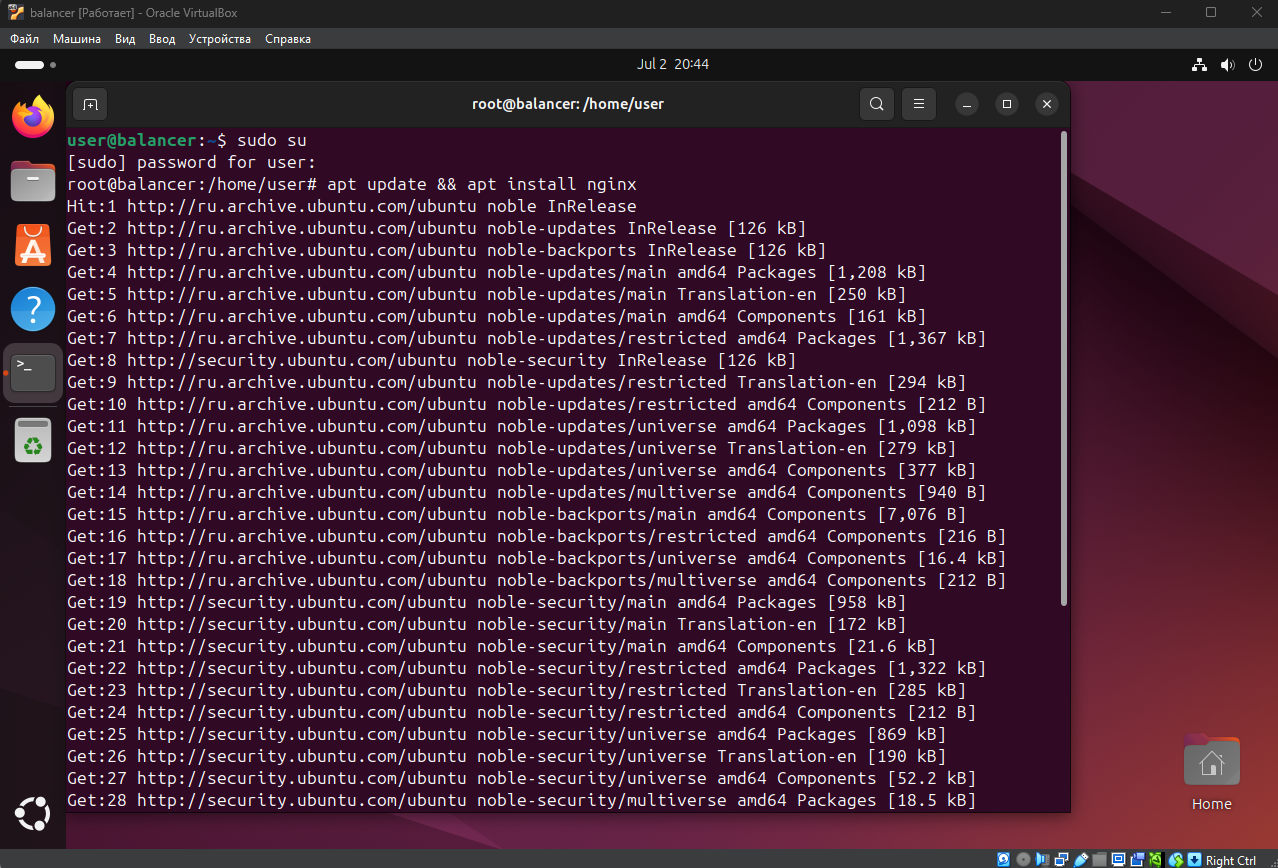


Рисунок 4 – Начальная настройка сервера

Также изменим файлик hosts на каждом из серверов (рисунок 5).

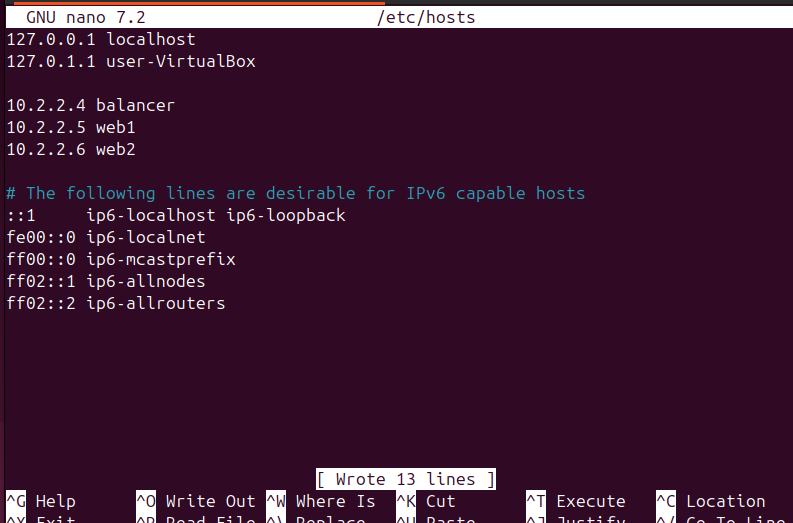


Рисунок 5 – Hosts

Затем перейдём к настройке балансировщика, посредством открытия файла конфигурации по умолчанию, и напишем следующие дополнения (рисунок 6).

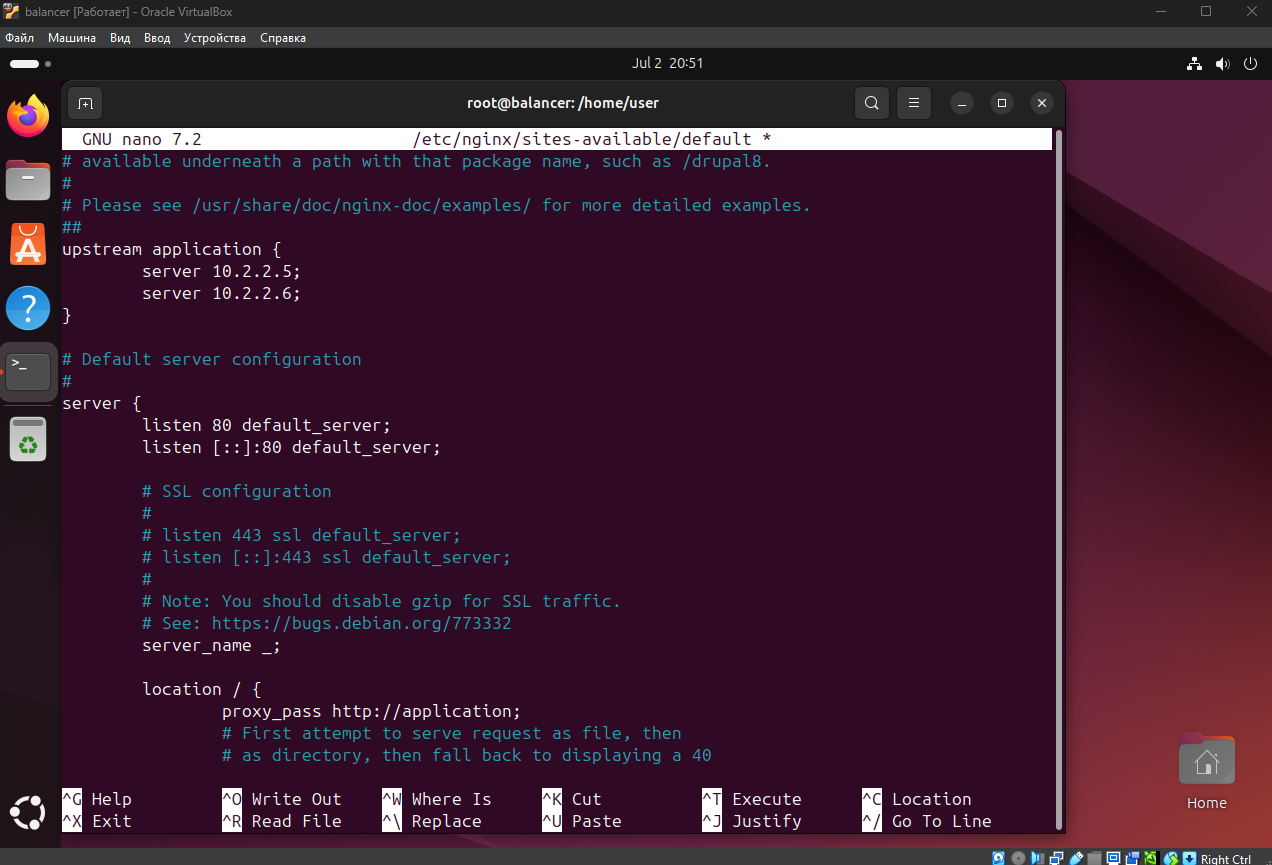


Рисунок 6 – Настройка балансировщика

По завершении его настройки на метод «Round Robin» мы можем видеть результат, который показывает, что первый запрос отправляет нас на первый сервер, второй запрос – на второй сервер и так по кругу. Этот алгоритм имеет ограничение в виде того, что экземпляры бэкенда будут простаивать просто потому, что ждут своей очереди (рисунок 7).

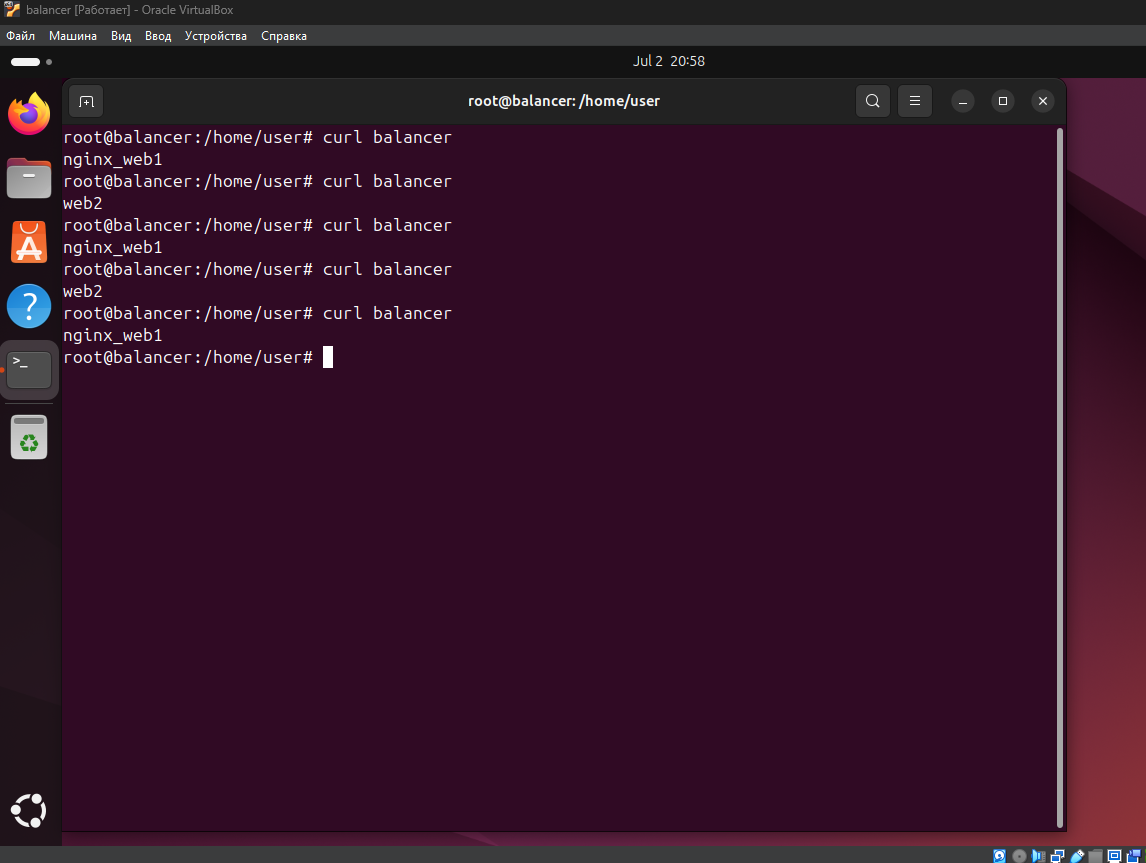


Рисунок 7 – Метод «Round Robin»

Далее изменим настройки конфигурационного файла, чтобы испытать метод «Weighted Round Robin», то же самое что предыдущий, но с указанием веса, который определяет как много трафика распределяется на конкретный экземпляр приложения. (рисунок 8).

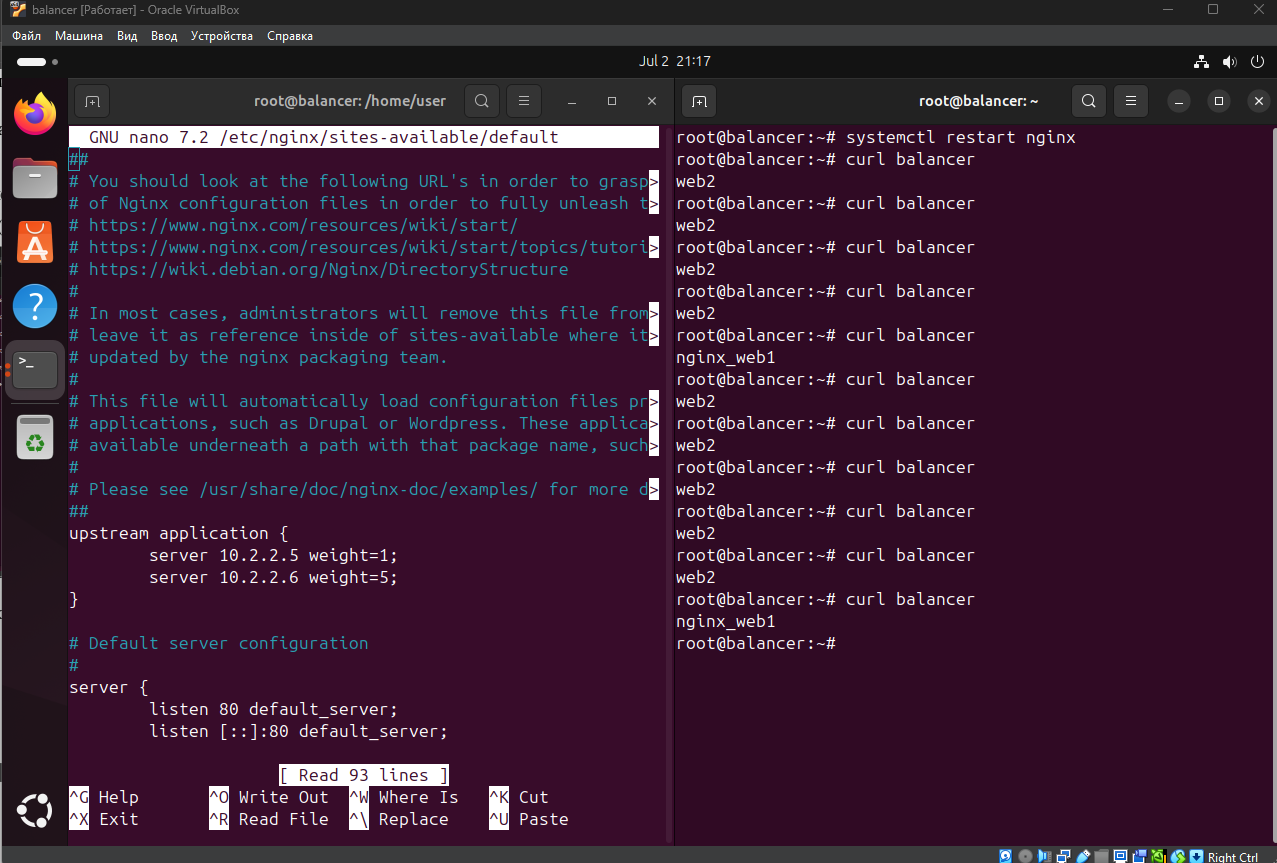


Рисунок 8 – Метод «Weighted Round Robin»

Далее изменим настройки конфигурационного файла, чтобы испытать метод «Least Connection». Этот метод обеспечивает равномерное распределение нагрузки между экземплярами приложения, как раз основываясь на количестве соединений с сервером. (рисунок 9).

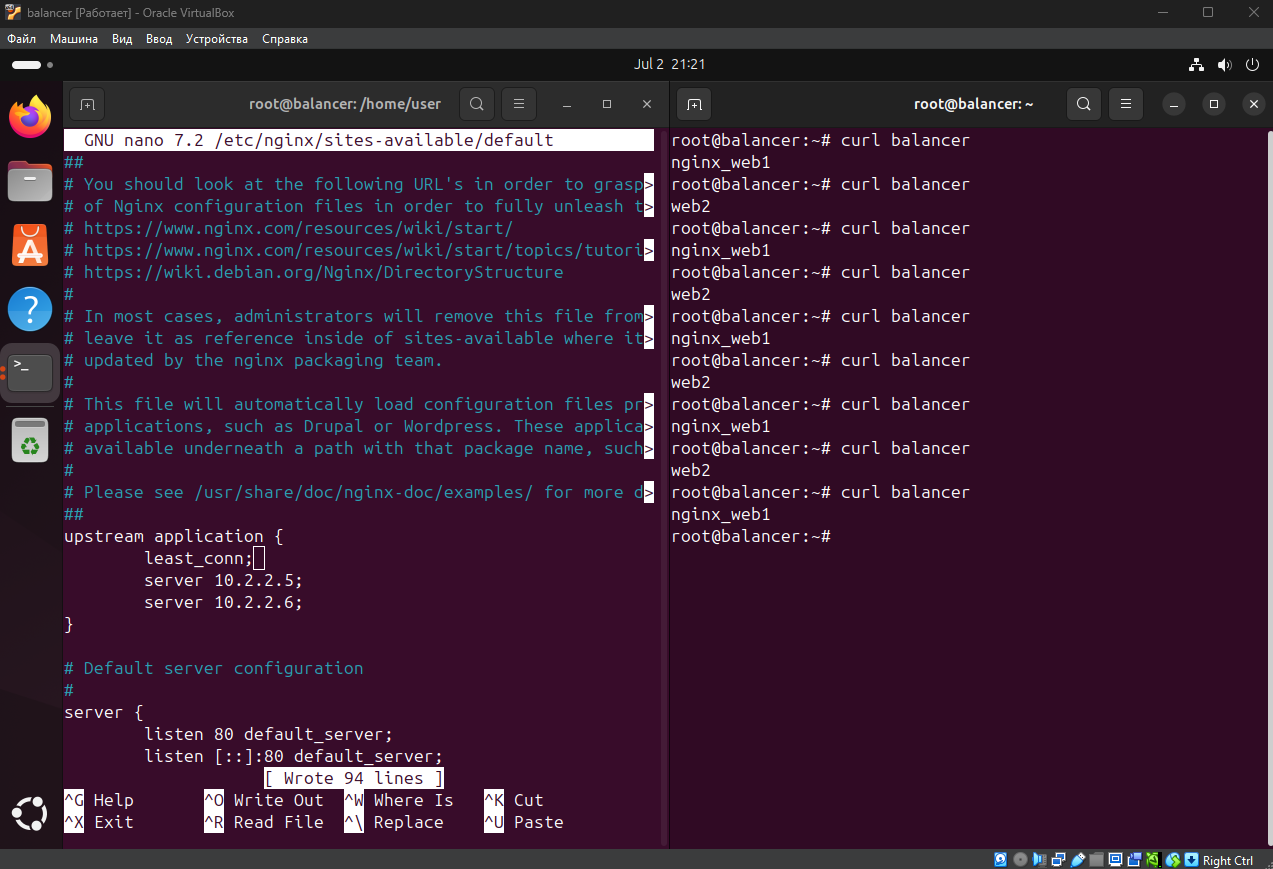


Рисунок 9 – Метод «Least Connection»

Кроме того этот алгоритм может использоваться вместе с весами для адресов, по аналогии с «Round Robin». В этом случае веса будут обозначать количество соединений с этим адресом по отношению к другим весам (рисунок 10).

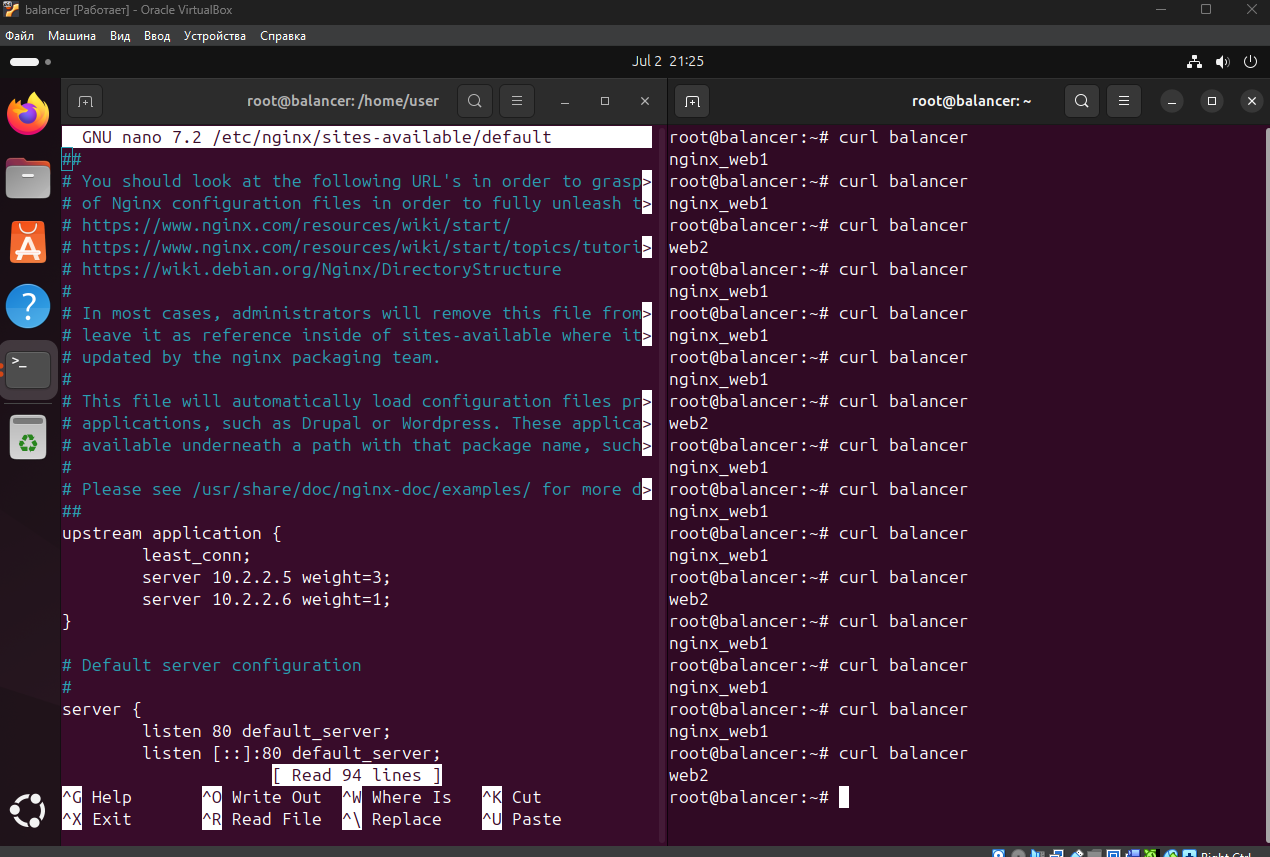


Рисунок 10 – Метод «Weighted Least Connection»

IP Hash будет замыкающим методом, он работает на основе IP-адреса клиента. Он гарантирует, что все запросы с одного адреса будут доставлены на один и тот же экземпляр приложения. IP Hash работает следующим образом: высчитывает хэш у адреса клиента и адреса сервера, и использует этот результат как уникальный ключ при балансировке (рисунок 11).

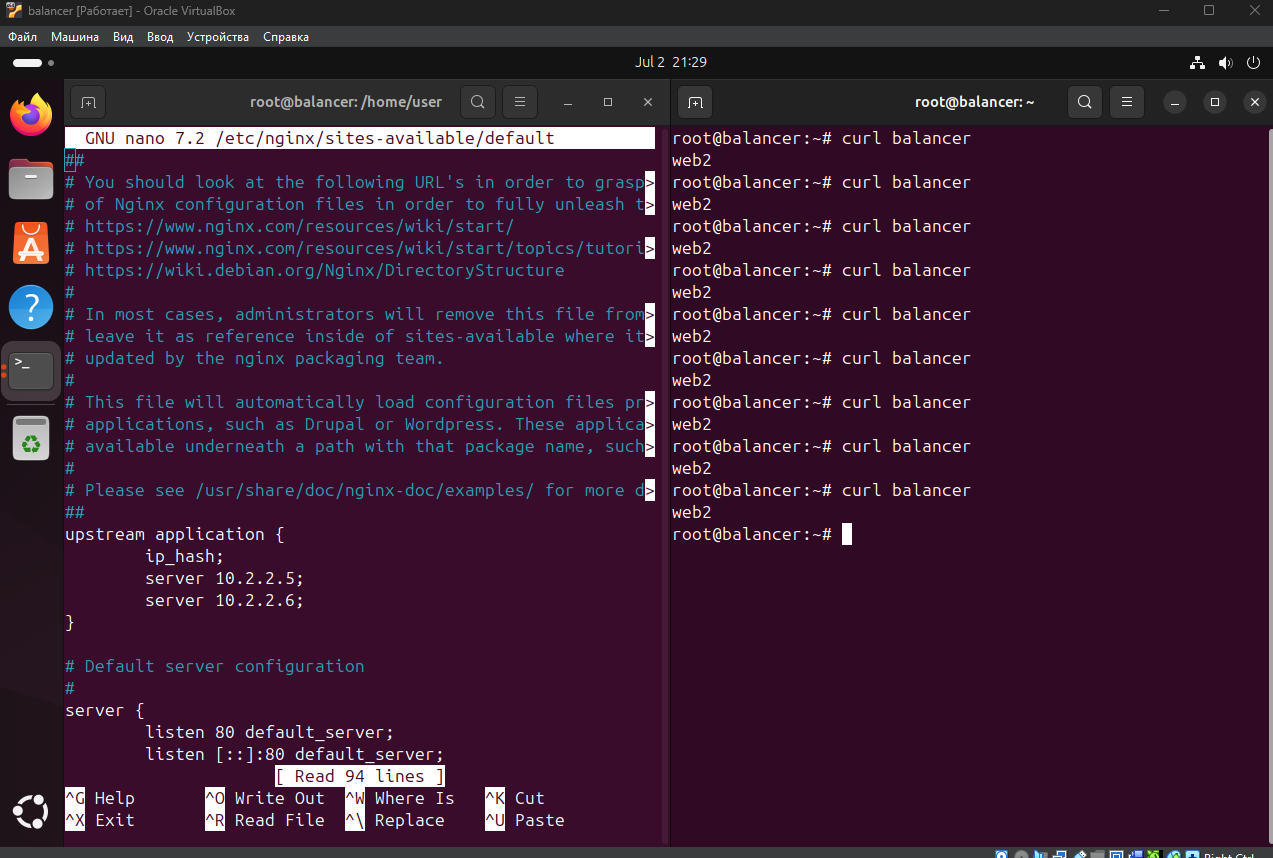


Рисунок 11 – Метод «IP Hash»

Помимо методов балансировки нужно не забывать обнаруживать неполадки с серверам, и, в случае чего, прекращать направлять трафик на «упавшие» экземпляры приложения. Я изменю конфигурацию и намеренно «положу» первый сервер после трёх запросов, и мы увидим что nginx перенаправляет трафик только на второй сервер (рисунок 12).

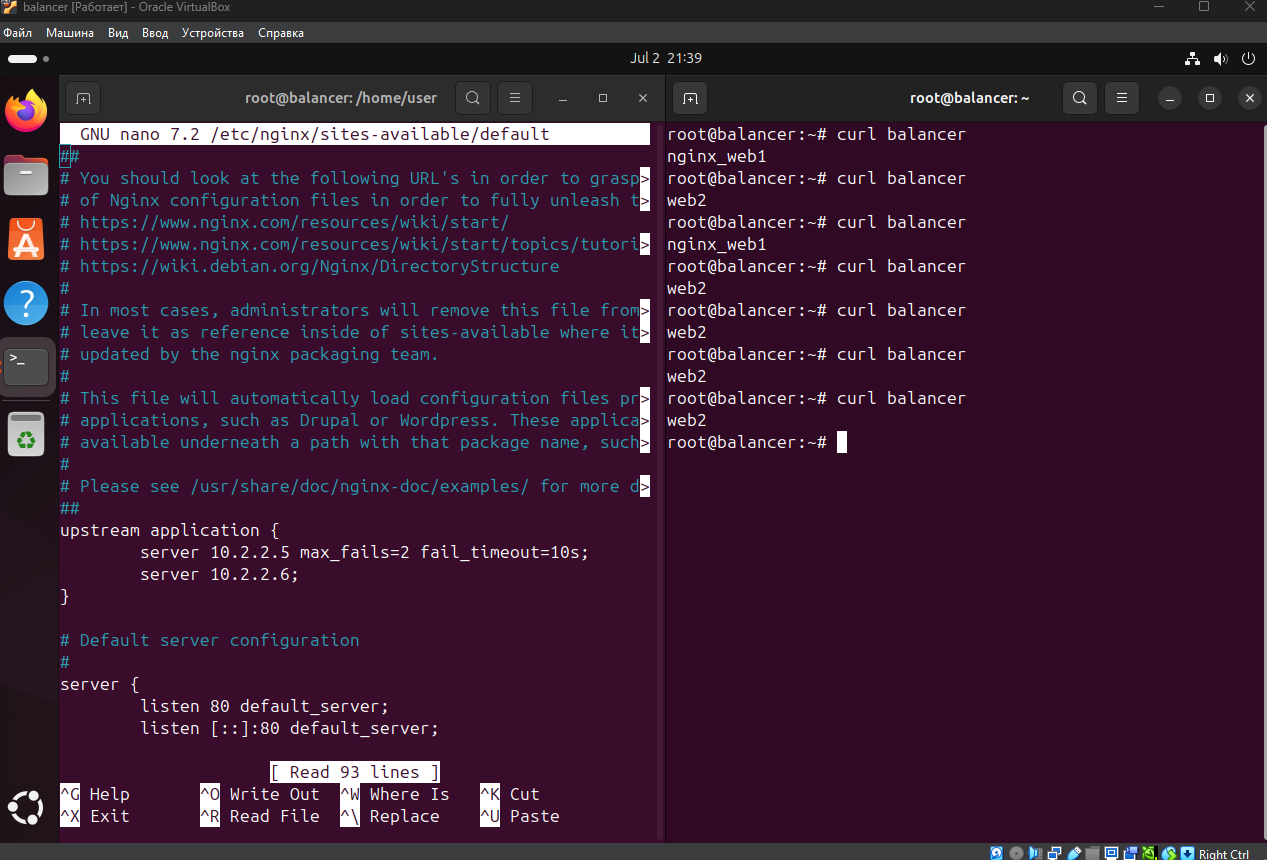


Рисунок 12 – Обработка ошибок

# **Контрольные вопросы**

## **Что такое Nginx?**

Nginx - это высокопроизводительный веб-сервер и обратный прокси-сервер с открытым исходным кодом. Он широко используется для обслуживания статических файлов, организации обратного проксирования, балансировки нагрузки и обеспечения безопасности. Благодаря своей событийной архитектуре, Nginx способен эффективно обрабатывать большое количество одновременных соединений при низком потреблении ресурсов.

## **Что такое Apache2?**

Apache2 - это также веб-сервер с открытым исходным кодом. Он используется для обслуживания веб-контента и приложений. Apache поддерживает модульную архитектуру, что позволяет добавлять различные функции, такие как обработка PHP, шифрование SSL, переадресации и др. В отличие от Nginx, Apache работает по процессной модели, где каждому соединению соответствует отдельный процесс или поток.

## **Какие компоненты входят в Proxmox?**

Балансировщик нагрузки - это компонент сетевой инфраструктуры, предназначенный для распределения входящего трафика между несколькими серверами. Его цель - обеспечить отказоустойчивость, повысить производительность и равномерную загрузку серверов. Балансировщик может использовать различные алгоритмы распределения трафика, такие как Round Robin, Least Connection, Weighted методы и IP Hash.

# **Вывод**

В процессе выполнения задач были приобретены практические навыки по установке и настройке балансировщика.