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

Институт Информационных технологий (ИТ)

Кафедра Математического обеспечения и стандартизации информационных технологий (МОСИТ)

|  |
| --- |
| **ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 3 (DOCKER)** |
| **по дисциплине** |
| **«Технологии разработки программных приложений»**  **Тема: «Системы контроля версий»** |

|  |  |
| --- | --- |
| Выполнил студент группы ИКБО-08-21 | Юнаков И.П. |
| Принял | Туманова М.Б. |

|  |  |  |
| --- | --- | --- |
| Практическая работа выполнена | «\_\_»\_\_\_\_\_\_\_202\_г. | *(подпись студента)* |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_202\_\_ г. | *(подпись руководителя)* |

Москва 2023

Оглавление

[Часть 1 – Подготовка инфраструктуры 3](#_Toc137049682)

[1. Создание виртуальных машин 3](#_Toc137049683)

[2. Настройка сетевого адаптера виртуальных машин 3](#_Toc137049684)

[3. Создание управляющей машины 4](#_Toc137049685)

[Часть 2 – Настройка Ansible 5](#_Toc137049686)

[Часть 3 – Использование Ansible для конфигурации хостов 6](#_Toc137049687)

[1. Факты в Ansible 6](#_Toc137049688)

[2. Playbook 6](#_Toc137049689)

[3. Более сложный playbook 8](#_Toc137049690)

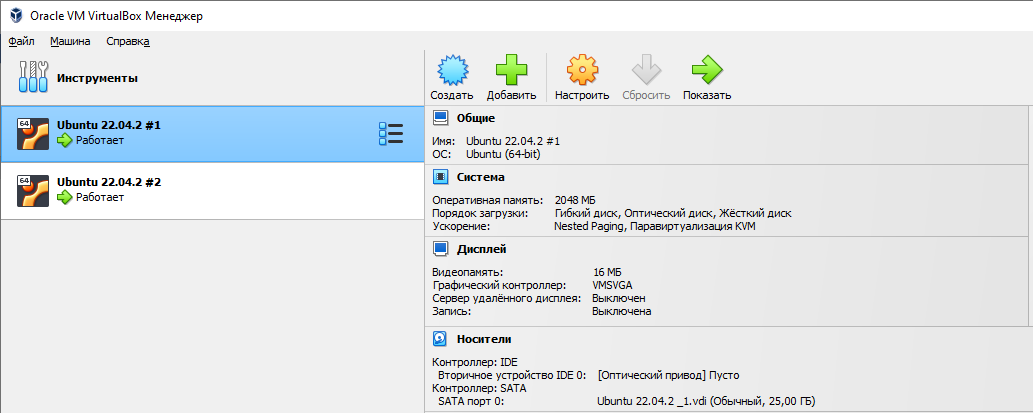
[4. Роли 10](#_Toc137049691)

[5. Индивидуальное задание 11](#_Toc137049692)

# Часть 1 – Подготовка инфраструктуры

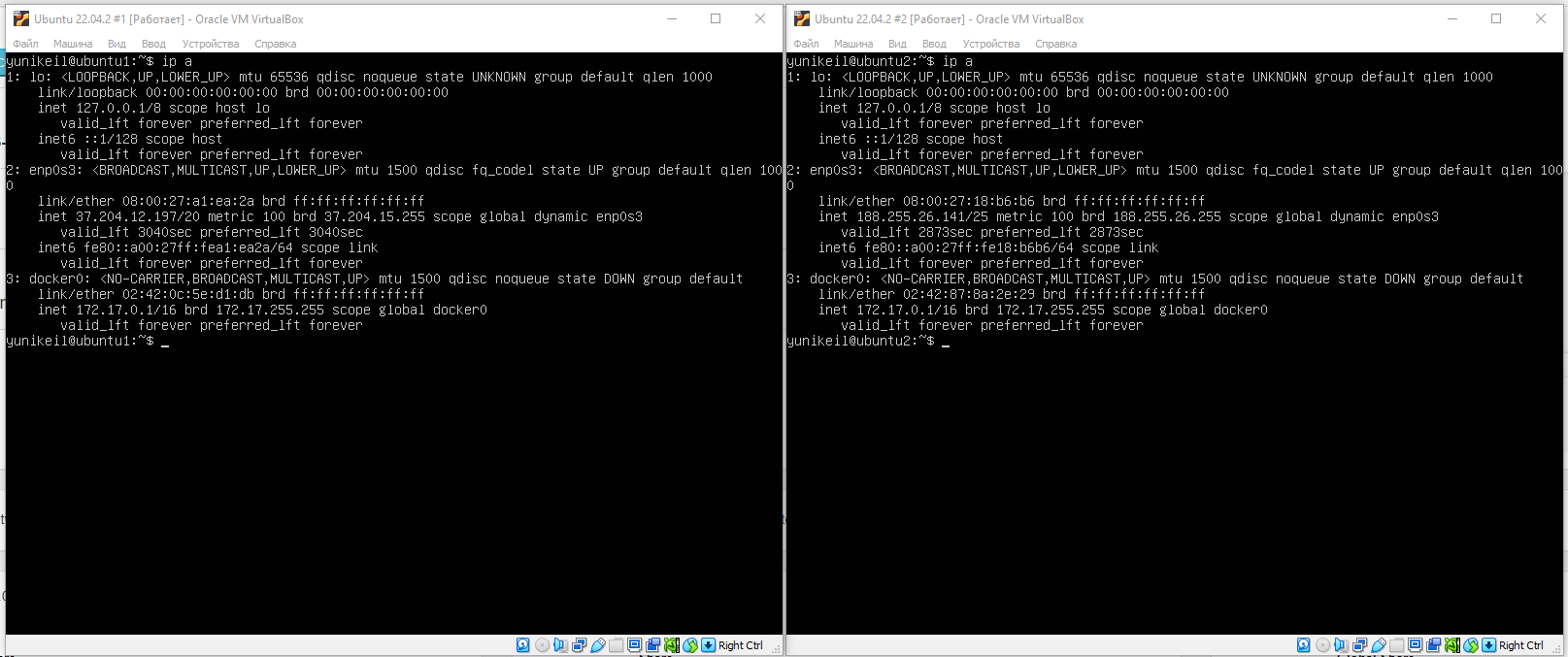
1. Создание виртуальных машин

Для установки виртуальных машин будет использоваться Oracle VM Box, а также Ubuntu server 22.04.

Рисунок 1 – Созданные виртуальные машины

1. Настройка сетевого адаптера виртуальных машин

Первым шагом требуется настроить виртуальный адаптер в режим работы «сетевой мост». После этого командой ip a или hostname -I нужно найти адрес системы.

Рисунок 2 – Информации о настройках сети в виртуальных машинах

Следующим шагом требуется подключиться по ssh. Также в начале может потребоваться принять ключи, или добавить свои для беспарольного доступа. Так как в windows OpenSSH нет команды ssh-copy-id, приходится использовать её эквивалент.

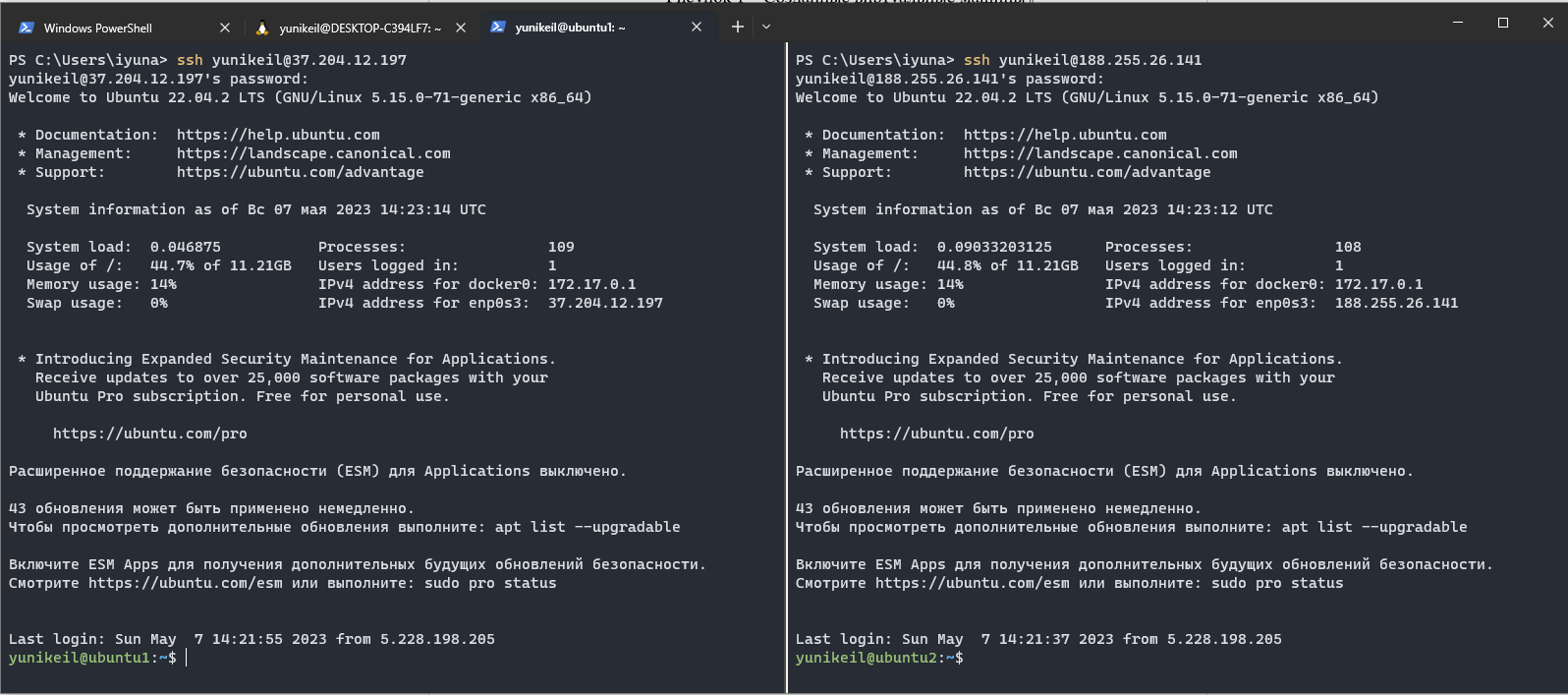
PS C:\Users\iyuna> type $env:USERPROFILE\.ssh\id\_rsa.pub | ssh yunikeil@188.255.26.141 "cat >> .ssh/authorized\_keys"

yunikeil@188.255.26.141's password:

PS C:\Users\iyuna> type $env:USERPROFILE\.ssh\id\_rsa.pub | ssh yunikeil@37.204.12.197 "cat >> .ssh/authorized\_keys"

yunikeil@37.204.12.197's password:

PS C:\Users\iyuna>

Рисунок 3 – ssh подключение к виртуальным машинам

Задавать статический адрес в данный момент не имеет надобности, т.к. времени аренды dhcp будет достаточно.

1. Создание управляющей машины

Для установки Ansible будем использовать wsl2 (Windows Subsystem for Linux). Если требуется установить wsl – просто нужно использовать   
wsl --install. Система автоматически поставит стандартную os.

PS C:\Users\iyuna> wsl ~

yunikeil@DESKTOP-C394LF7:~$ cat /etc/issue

Ubuntu 20.04.5 LTS \n \l

yunikeil@DESKTOP-C394LF7:~$

Установка Ansible:

yunikeil@DESKTOP-C394LF7:~$ sudo apt install ansible

Reading package lists... Done

...

Processing triggers for man-db (2.9.1-1) ...

yunikeil@DESKTOP-C394LF7:~$ ansible --version

ansible 2.9.6

config file = /etc/ansible/ansible.cfg

configured module search path = ['/home/yunikeil/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']

ansible python module location = /usr/lib/python3/dist-packages/ansible

executable location = /usr/bin/ansible

python version = 3.8.10 (default, Mar 13 2023, 10:26:41) [GCC 9.4.0]

yunikeil@DESKTOP-C394LF7:~$

Вся настройка ansible может производиться через файл конфигурации (/etc/ansible/ansible.cfg).

# Часть 2 – Настройка Ansible

Данный код показывает содержание файла-инвентаря Ansible, который находится в директории Ansible и содержит информацию о хостах, которые используются при работе с Ansible. В первой секции [webservers] перечислены хосты, которые используются в Ansible. Во второй секции [webservers: vars] указан интерпретатор Python, который используется при выполнении задач на этих хостах.

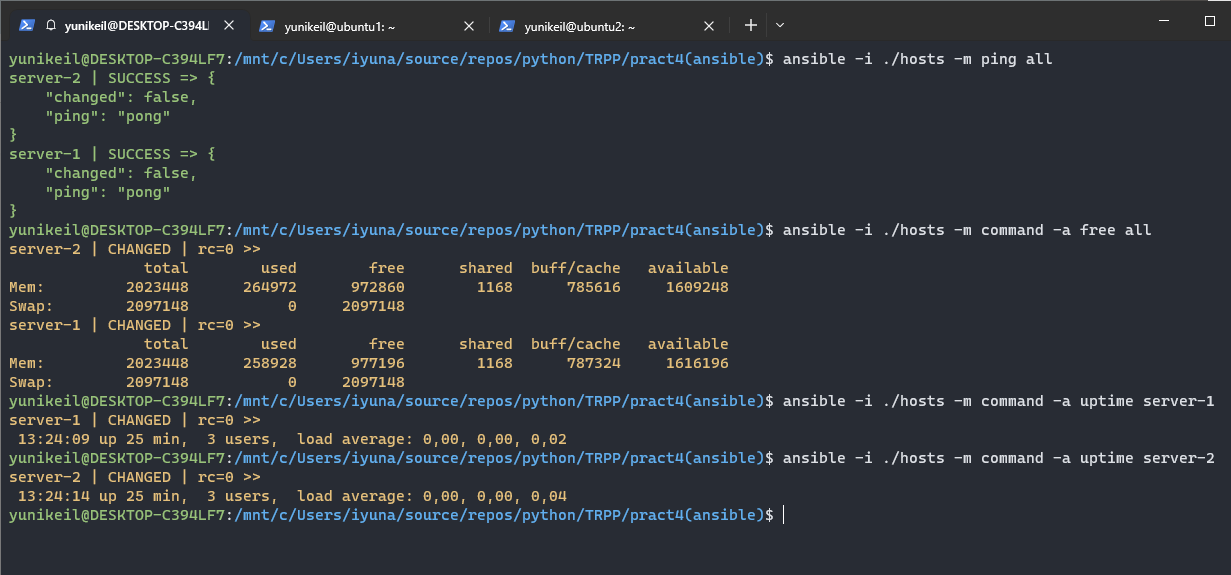
[webservers]

server-1 ansible\_host=5.228.149.236

server-2 ansible\_host=46.242.33.10

[webservers:vars]

ansible\_python\_interpreter=/usr/bin/python3

Рисунок 4 – Выполнение команд из инструкции

# Часть 3 – Использование Ansible для конфигурации хостов

1. Факты в Ansible

Факты Ansible — это данные, относящиеся к удаленным системам, включая операционные системы, IP-адреса, присоединенные файловые системы и многое другое. Факты могут использоваться для того, чтобы использовать поведение или состояние одной системы в качестве конфигурации на других системах. Ansible хранит факты в формате JSON, элементы группируются по нодам. Факты могут быть получены с помощью модуля настройки, который используется для сбора этих фактов. Факты могут также быть зарегистрированы как переменные в плейбуке, что позволяет получить доступ к определенным фактам и использовать их в задачах. В переменной регистрации хранится словарь, содержащий множество других ключей помимо ключа stdout, такие как rc (код возврата), start (время выполнения команды), end (время завершения команды), stderr (ошибки) и т.д. Собрать факты можно, используя команду ansible {server\_name} -i ./hosts -m command -m setup.

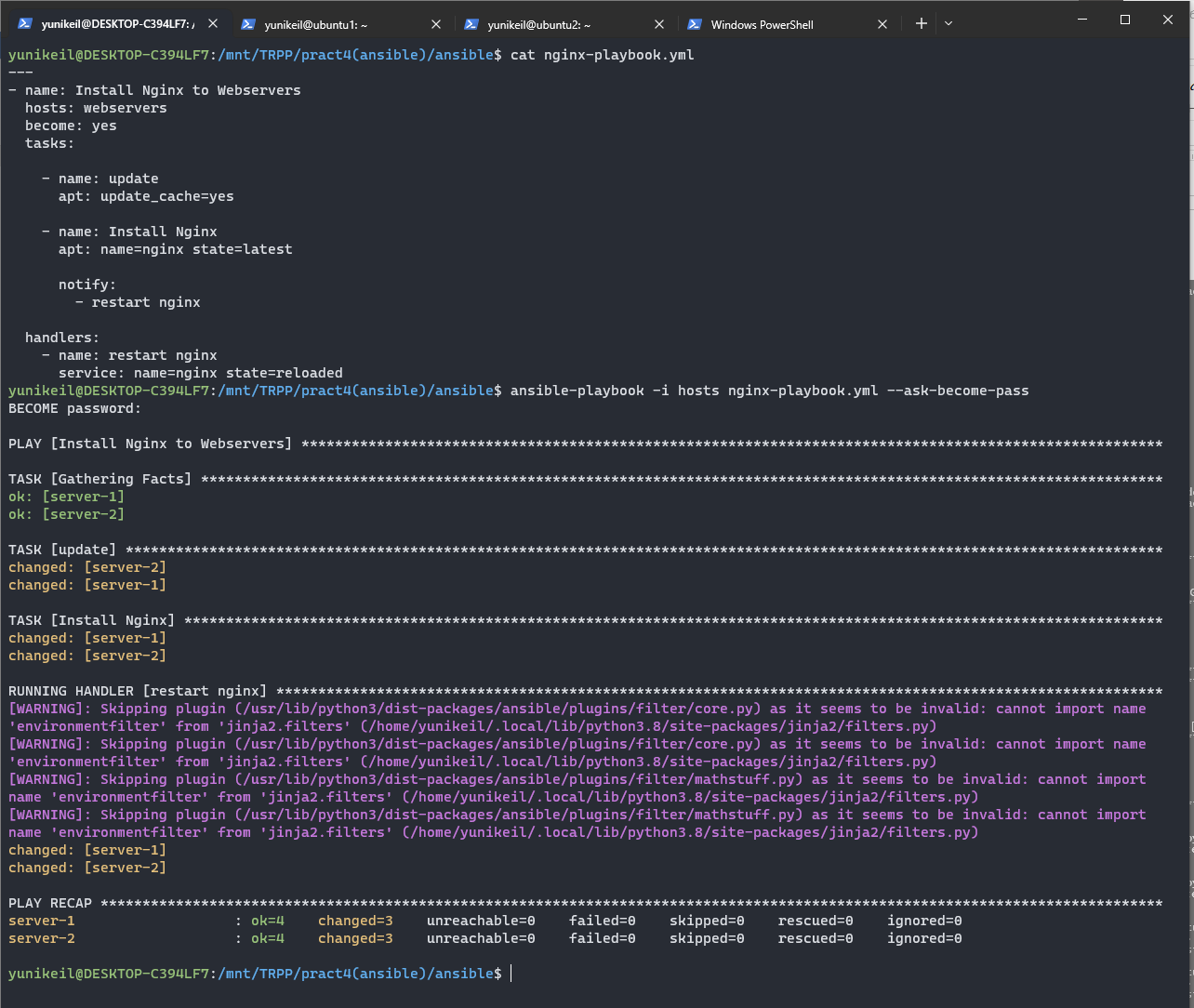
1. Playbook

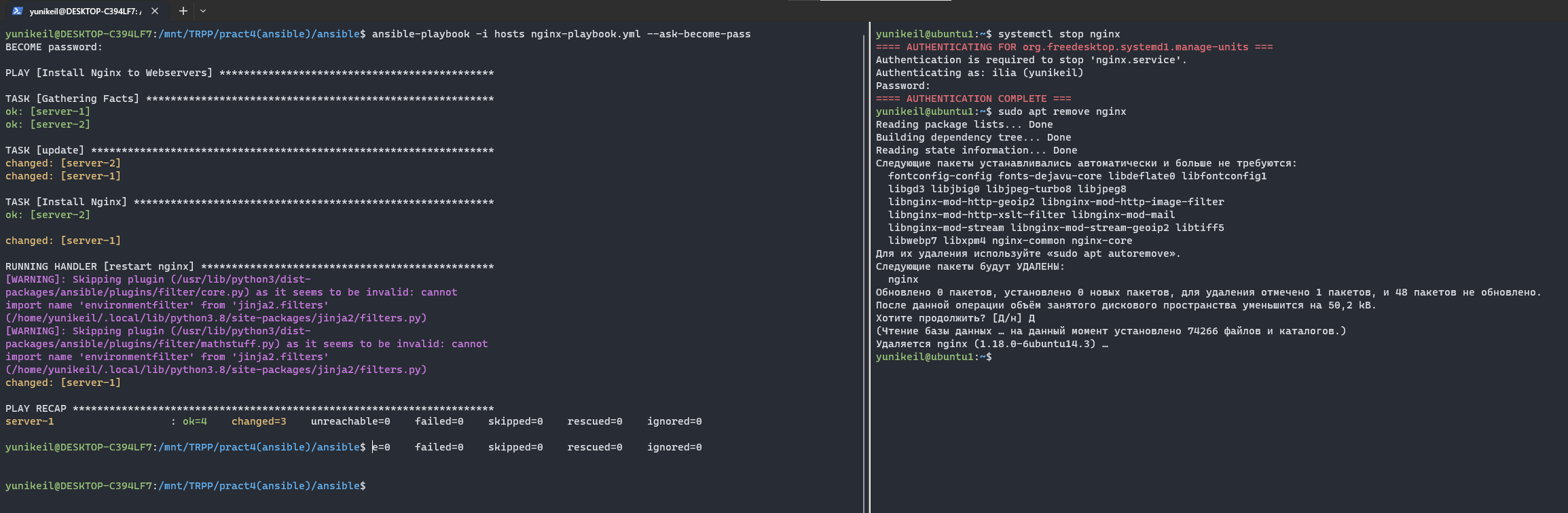
Ansible playbook — это конфигурационный сценарий на языке YAML, который впоследствии будет выполняться на управляемых хостах.

Ниже представлен Ansible playbook, который устанавливает Nginx на управляемых хостах. Этот playbook содержит следующие шаги:

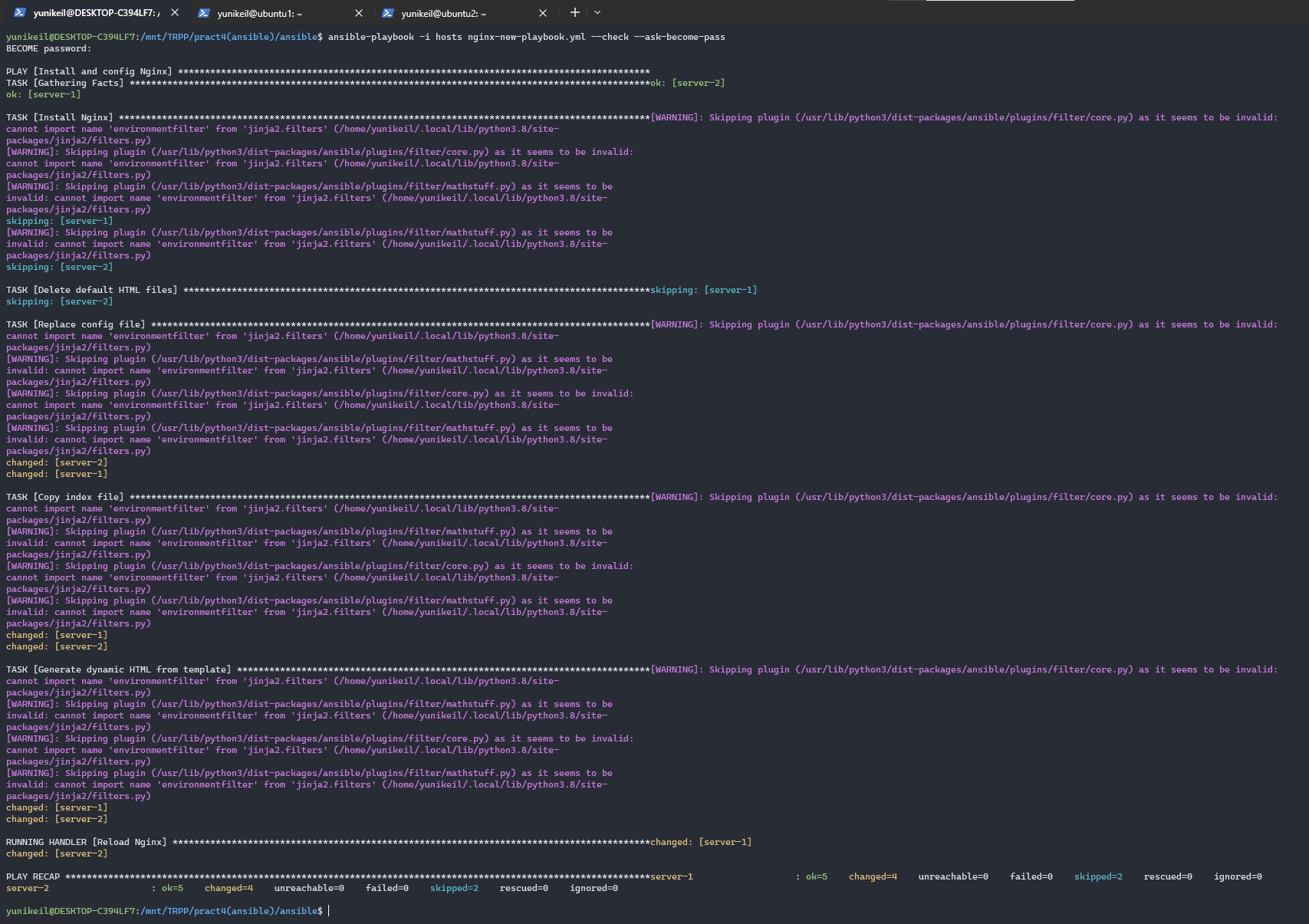
* Он обновляет список пакетов на управляемых хостах
* Он устанавливает Nginx на управляемых хостах
* Он выполняет перезапуск Nginx

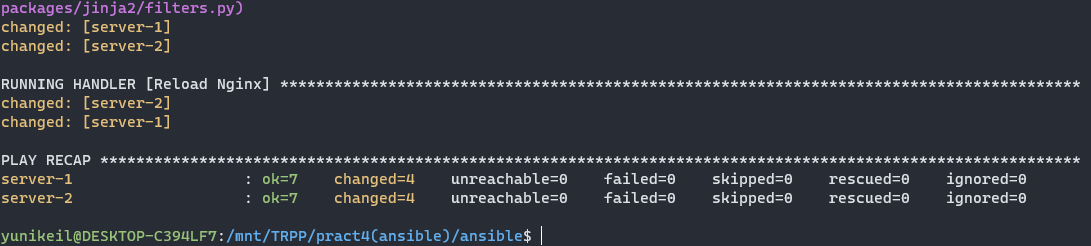
Эти шаги выполняются на хостах, определенных в секции hosts. В данном случае управляемыми хостами являются те, которые определены в группе webservers. Данный playbook также использует обработчик handler, который выполняет перезапуск Nginx после его установки.

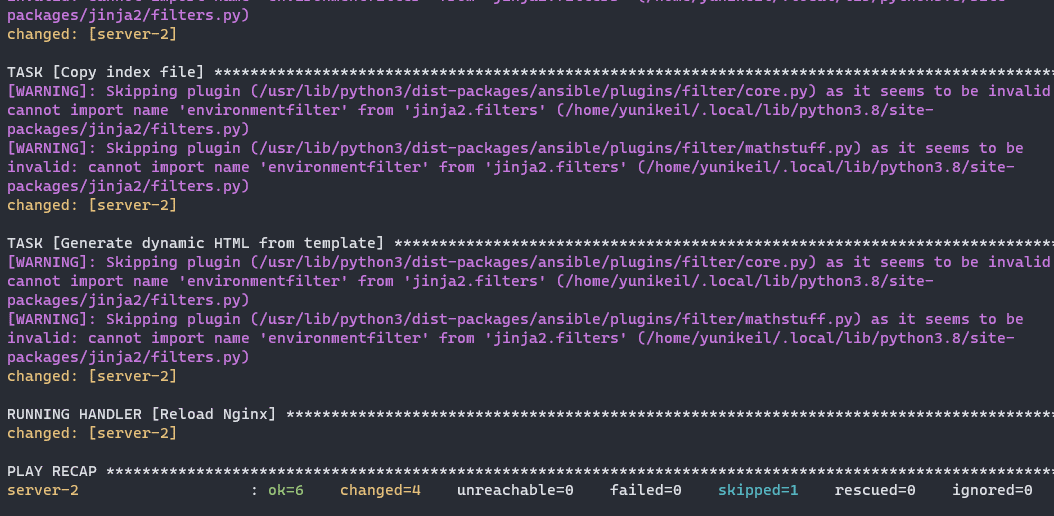
Рисунок 5 – Ansible playbook, который устанавливает Nginx на управляемых хостах

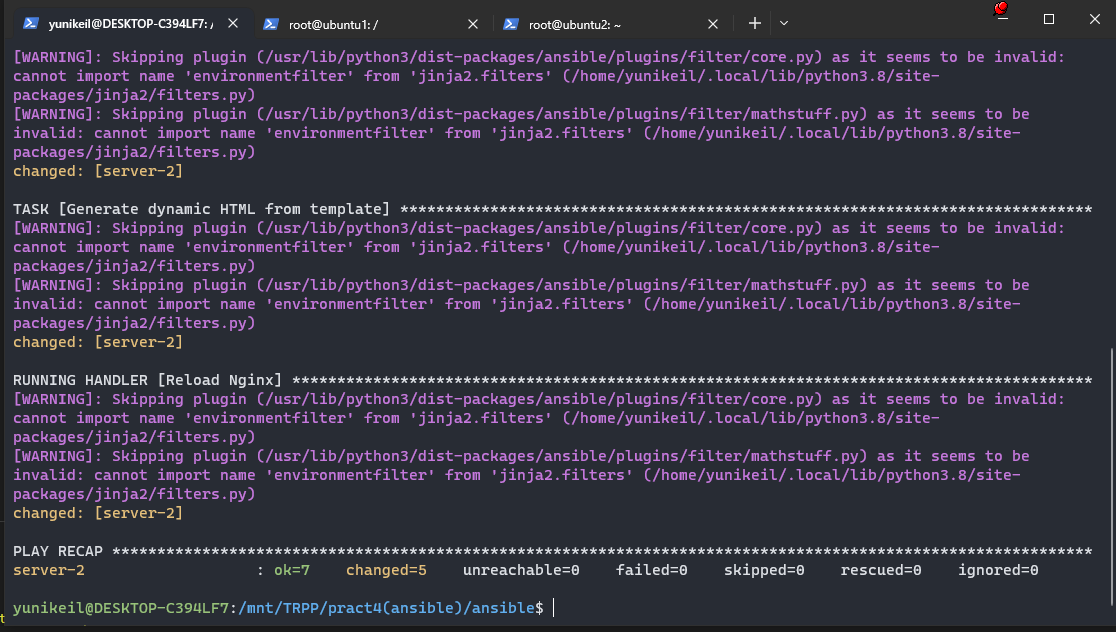
 Рисунок 6 – Удаление nginx с первого сервера(справа) и его переустановка(слева)

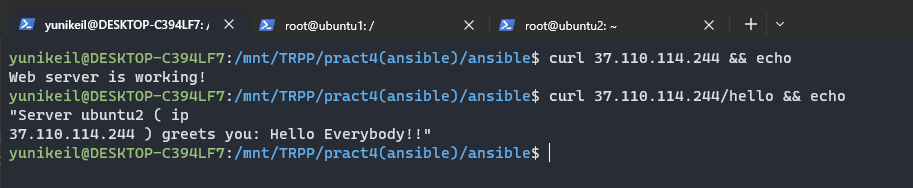
1. Более сложный playbook

Рисунок 7 – Результат выполнения команды из инструкции.

Рисунок 8 – Пробный прогон плейбука

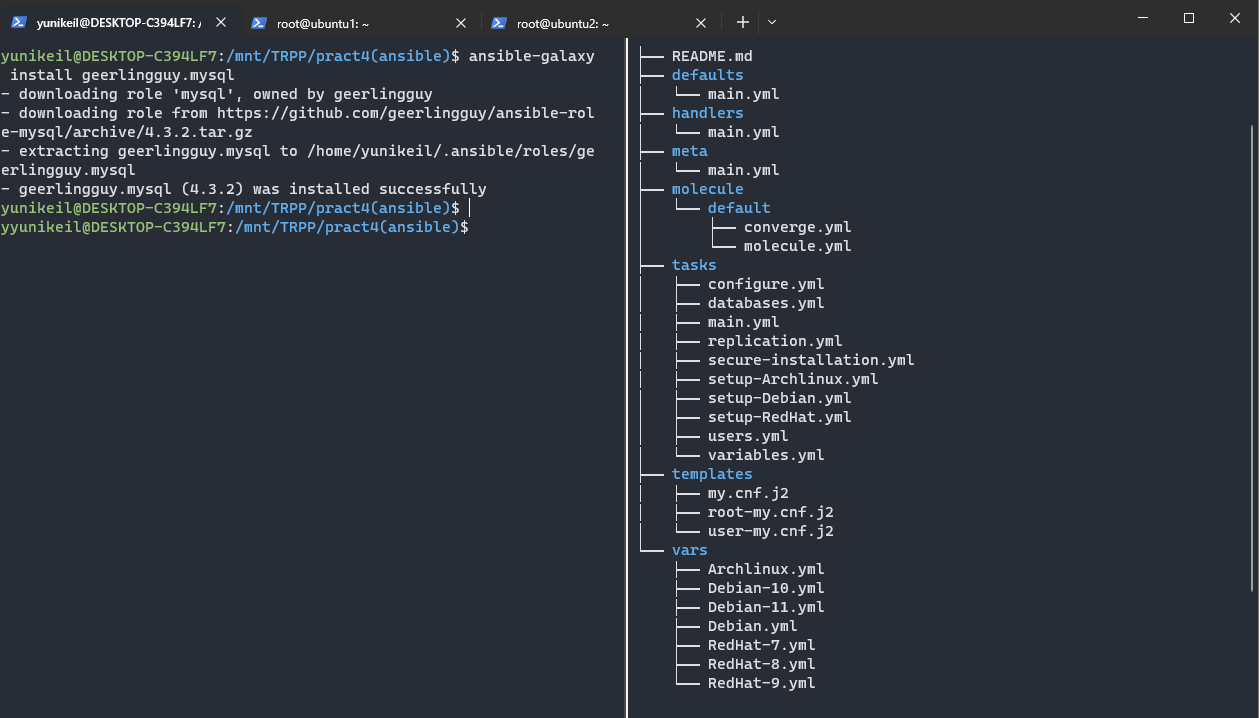
Рисунок 9 – выполнкнный плейбу

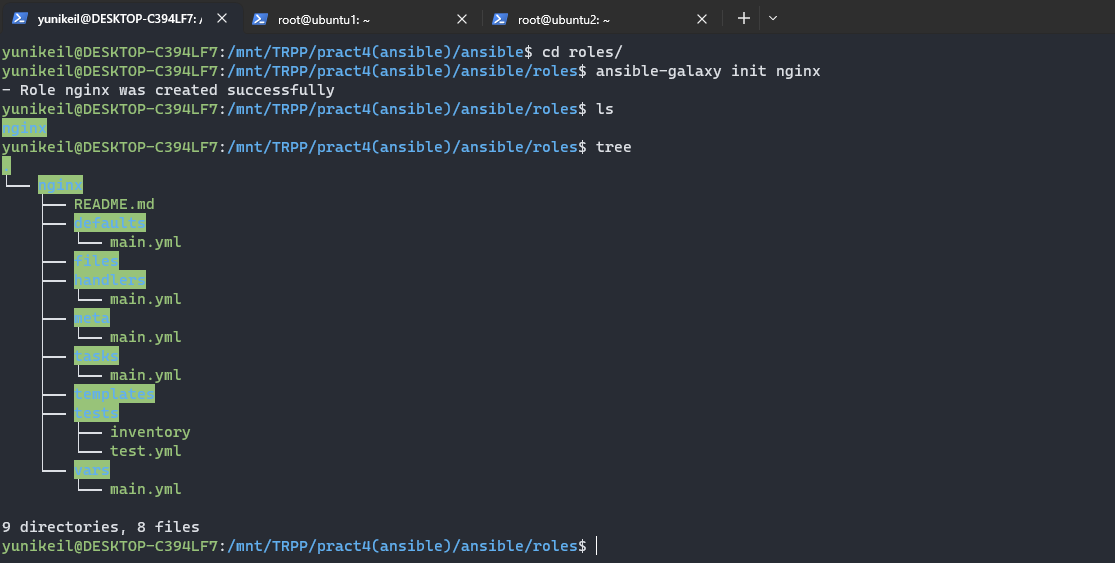
рисунок 10 – заменён стандартный файл веб сервера

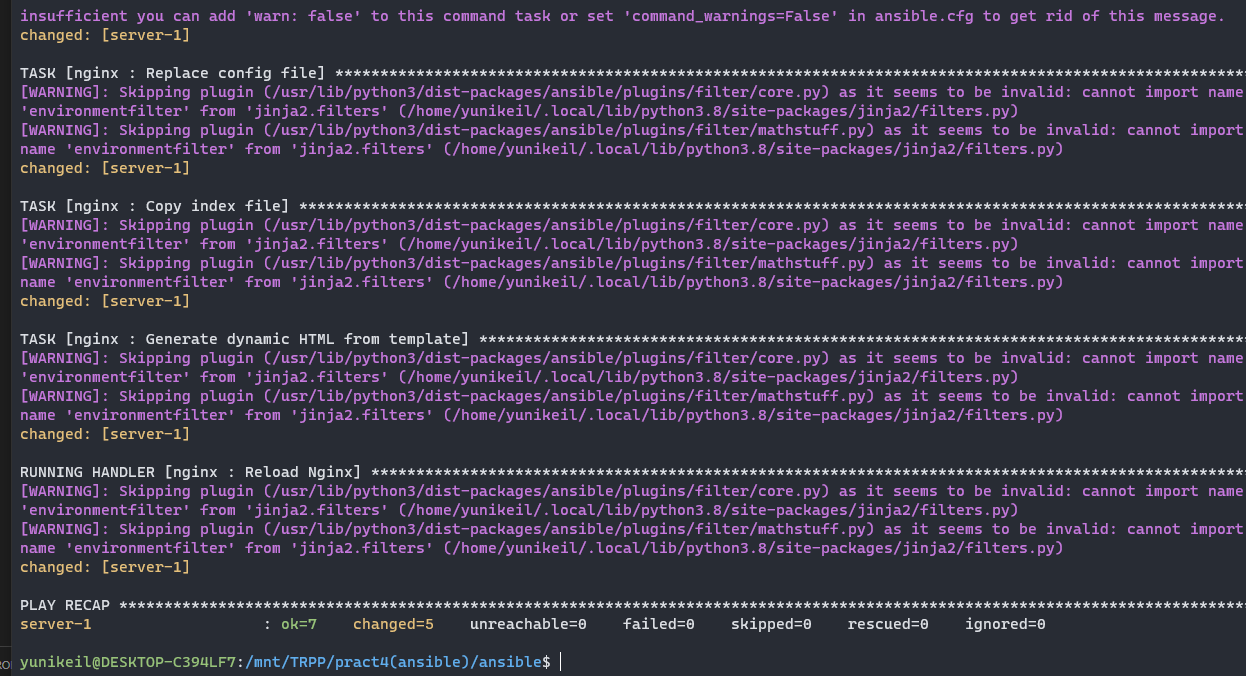
рисунок 11 замена стартовых страниц nginx

1. Роли

Роль в Ansible — набор задач или обработчик переменных, файлов и других артефактов, которые распространяются и подключаются как единое целое к плейбуку. Обычно она отвечает за высокоуровневые задачи: установку баз данных, веб-серверов и др.

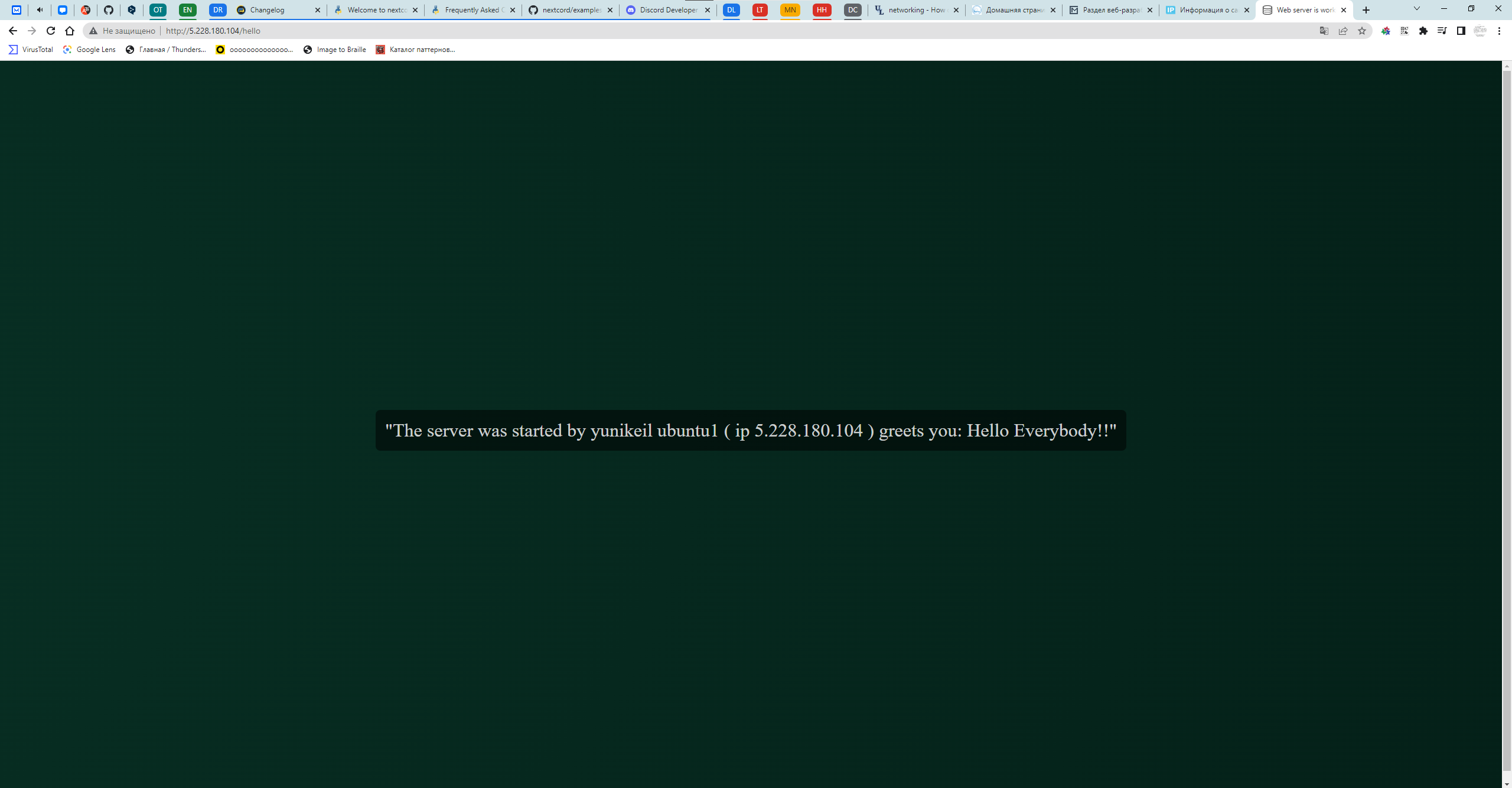
Рисунок 12 установка и просмотр содержимого роли

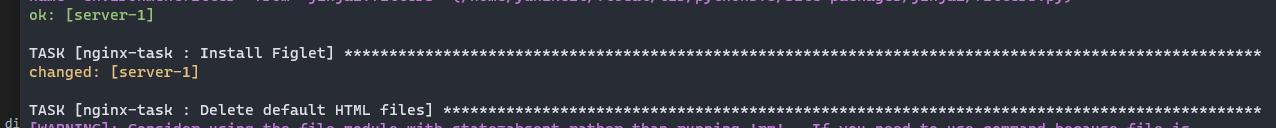
Рисунок 13 – инициализация пользовательской роли

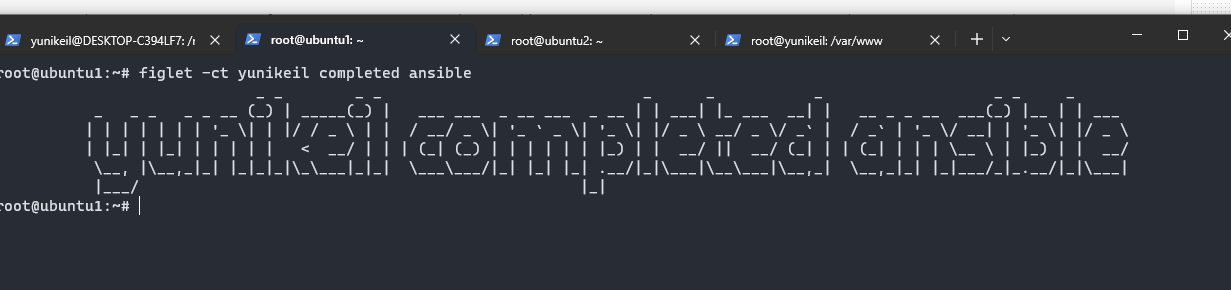
Рисунок 14 – установка роли на первый сервер

1. Индивидуальное задание

FIGlet — компьютерная программа, генерирующая текстовый баннер с буквами, составленными из букв.

Рисунок 15 веб сервер с именем

Рисунок 16 установка пакета фиглет

Рисунок 17 проверка работы пакеты фиглет

1. Вывод о проделанной работе

В ходе практической работы я научился простейшим действиям с инструментом для автоматизации IT процессов, который позволяет управлять конфигурацией и развертыванием приложений на удаленных серверах. С помощью Ansible можно создавать и управлять инфраструктурой, настраивать сервера, устанавливать и обновлять ПО, а также автоматизировать другие задачи. Ansible использует простой и понятный язык описания конфигураций, что делает его доступным для использования разработчиками и системными администраторами.