|  |
| --- |
|  |
| 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 № 4** |
| **по дисциплине**  «ТЕХНОЛОГИЯ РАЗРАБОТКИ ПРОГРАММНЫХ ПРИЛОЖЕНИЙ» |
| **Тема: «Системы конфигурационного управления»** |

|  |  |
| --- | --- |
| Выполнил студент группы ИКБО-32-21 | Быченков А.К. |
| Принял | Миронов А.И. |

|  |  |  |
| --- | --- | --- |
| Практическая работа выполнена | «\_\_»\_\_\_\_\_\_\_2023 г. | *(подпись студента)* |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_2023 г. | *(подпись руководителя)* |

Москва 2023

**Цель работы:** получить навыки настройки вычислительной инфраструктуры при помощи системы конфигурационного управления Ansible.

**Задание**

Написать роль для запуска сервера nginx, написать playbook для применения роли, провести тестовый запуск playbook’а, в случае успешного прохождения теста, применить playbook к серверам.

Необходимо добавить переменную, содержащую ФИО, номер группы и номер варианта. Данная переменная должна выводиться в шаблонный файл nginx.

Установка пакета выполняется при помощи модуля APT, используемого для установки nginx в базовой роли. Добавьте в playbook task по установке пакета согласно варианту.

Вариант 4: пакет patch.

**Ход работы**

Были созданы 2 виртуальные машины Debian в VirtualBox при помощи инструмента Vagrant. Для каждой машины был назначен ip-адрес, настроено подключение по SSH по паролю. Конфигурация Vagrant:

Листинг 1 – Vagrantfile

Vagrant.configure("2") do |config|

config.vm.define "node1" do |node|

node.vm.box = "debian/bullseye64"

node.vm.network "public\_network", ip: "192.168.0.5", bridge: "Realtek RTL8822CE 802.11ac PCIe Adapter"

end

config.vm.define "node2" do |node|

node.vm.box = "debian/bullseye64"

node.vm.network "public\_network", ip: "192.168.0.6", bridge: "Realtek RTL8822CE 802.11ac PCIe Adapter"

end

config.vm.provision "shell", inline: <<-SHELL

echo -e "vagrant\nvagrant" | passwd root

echo "PermitRootLogin yes" >> /etc/ssh/sshd\_config

sed -in 's/PasswordAuthentication no/PasswordAuthentication yes/g' /etc/ssh/sshd\_config

service ssh restart

SHELL

end

Настроим inventory-файл, прописав в нем ip-адреса запущенных виртуальных машин:

Листинг 2 – hosts

[nodes]

node1 ansible\_host=192.168.0.5 ansible\_user=root ansible\_password=vagrant

node2 ansible\_host=192.168.0.6 ansible\_user=root ansible\_password=vagrant

Проверим подключение ansible к виртуальным машинам, используя модуль ping:



Рисунок 1. Проверка подключения к виртуальным машинам

Командой ansible-galaxy init nginx инициализируем новую роль nginx. Будет создан каталог nginx. В файл vars/main.yml добавим переменные, содержащие ФИО, номер группы и номер варианта:

Листинг 3 – vars/main.yml

fio: Быченков Александр Константинович

group\_number: ИКБО-32-21

variant\_number: 4

html\_dir: /usr/share/nginx/html

Напишем шаблонный файл конфига nginx templates/nginx.conf.j2:

Листинг 4 – vars/main.yml

user www-data;

worker\_processes auto;

pid /run/nginx.pid;

events {

worker\_connections 1024;

}

http {

include /etc/nginx/mime.types;

default\_type application/octet-stream;

sendfile on;

keepalive\_timeout 65;

server {

listen 80 default\_server;

listen [::]:80 default\_server;

server\_name \_;

root /var/www/html;

index index.html;

location / {

root /usr/share/nginx/html;

try\_files $uri $uri/ $uri.html =404;

}

}

}

Напишем шаблонный файл templates/index.html.j2 для вывода заданных переменных:

Листинг 4 – templates/index.html.j2

<html lang="ru">

<head>

<meta charset="UTF-8" />

<meta http-equiv="X-UA-Compatible" content="IE=edge" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>Nginx Template</title>

</head>

<body>

<div>

<p>ФИО:</p>

<p>{{fio}}</p>

<p>Группа:</p>

<p>{{group\_number}}</p>

<p>Номер варианта:</p>

<p>{{variant\_number}}</p>

</div>

</body>

<style>

div {

display: grid;

grid-template-columns: repeat(2, 1fr);

width: 300px;

height: 200px;

padding-inline: 1.5em;

align-items: center;

justify-content: center;

font-size: 1.5em;

font-family: Arial, Helvetica, sans-serif;

}

</style>

</html>

В файл handlers/main.yml напишем обработчики, которые будут вызываться после выполнения задач:

Листинг 5 – handlers/main.yml

- name: Start Nginx

systemd:

name: nginx

state: started

- name: Reload Nginx

systemd:

name: nginx

state: reloaded

Напишем задачи в файл tasks/main.yml. Необходимо установить nginx, настроить его, добавить шаблонный файл:

Листинг 6 – tasks/main.yml

- name: Install nginx

apt:

name: nginx

state: present

update-cache: true

when: ansible\_os\_family == "Debian"

notify: Nginx Systemd

- name: Replace config file

vars:

nginx\_user: user

worker\_processes: 2

worker\_connections: 256

template:

src: templates/nginx.conf.j2

dest: /etc/nginx/nginx.conf

mode: 0644

register: result

failed\_when: result.failed == true

notify: Reload Nginx

- name: Generate dynamic HTML from template

template: src=templates/index.html.j2 dest={{ html\_dir }} owner=root mode=0644

notify: Reload Nginx

Роль nginx настроена. Теперь запустим эту роль в playbook.yml, а также установим пакет nginx:

Листинг 7 – playbook.yml

- name: Install and config Nginx via Role

hosts: all

become: true

roles:

- nginx

tasks:

- name: Install patch

apt:

name: patch

state: present

update\_cache: true

Проведем тестовый запуск playbook командой ansible-playbook -i hosts playbook.yml –check:

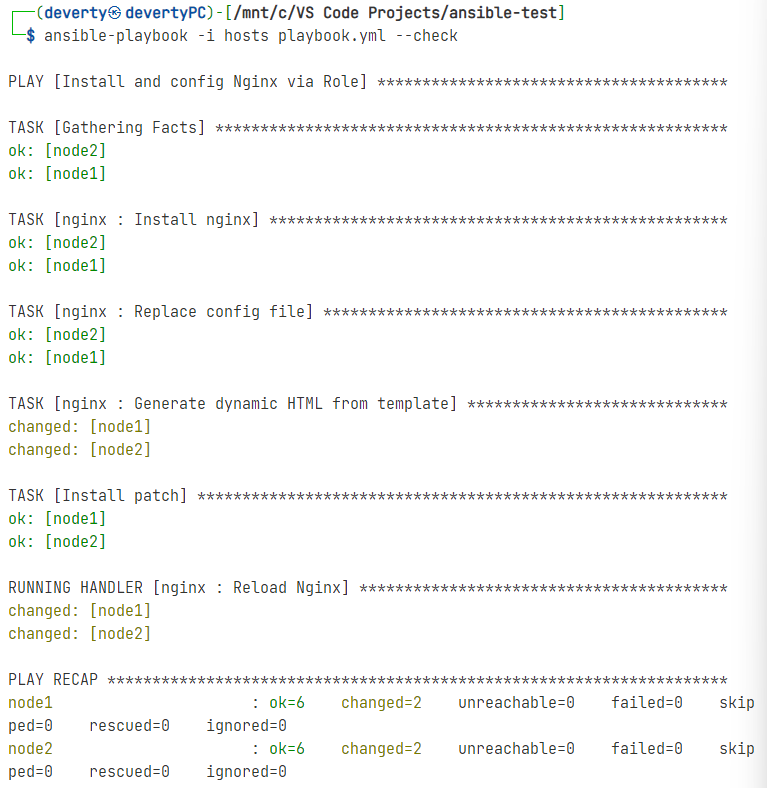


Рисунок 2. Тестовый запуск playbook.yml

Все задачи были успешно протестированы, теперь применим playbook к виртуальным машинам:

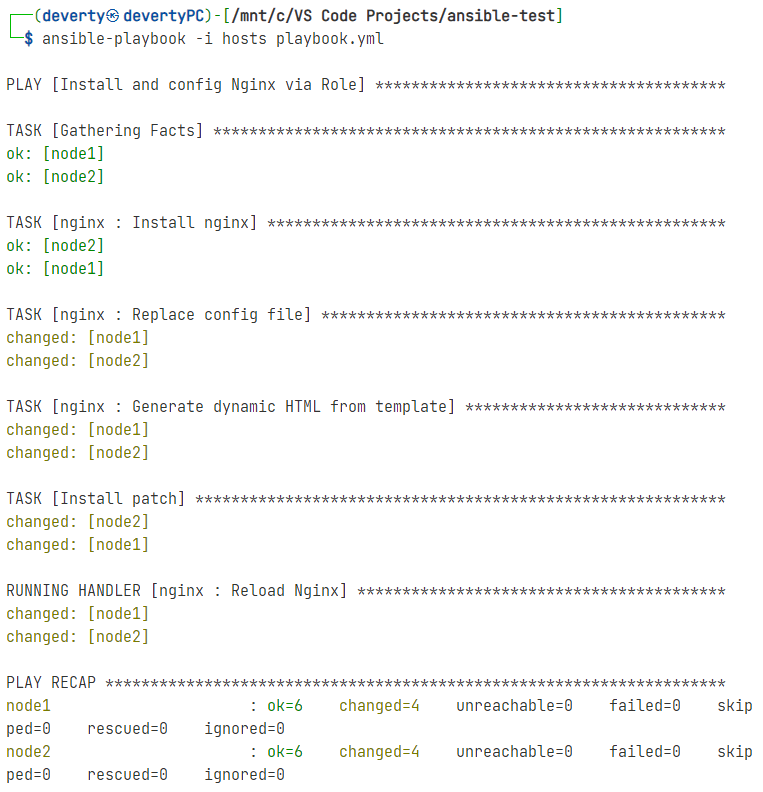


Рисунок 3. Запуск playbook.yml

Проверим работу nginx, обратившись по ip-адресу одной из виртуальных машин:

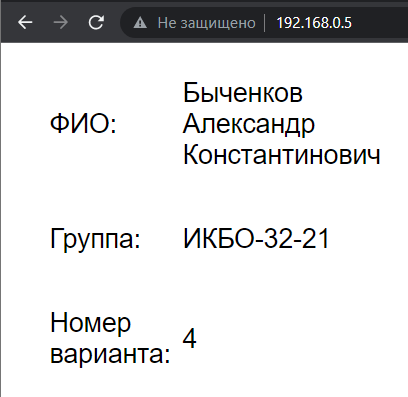


Рисунок 4. Ответ веб-сервера nginx

**Вывод**

В ходе выполнения данной практической работы мною были получены навыки настройки вычислительной инфраструктуры при помощи системы конфигурационного управления Ansible.