МИНОБРНАУКИ РОССИИ

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

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных технологий

Отчет по практической работе №4

по дисциплине «Технологии разработки программных приложений»

Тема практической работы: «Ansible»

Выполнил:

Студент группы ИКБО-04-20 Хан А.А.

Проверила: Овчинникова М.А.

Москва 2022 г.

СОДЕРЖАНИЕ

Цель работы: Знакомство с программным решением для удаленного управления конфигурациями Ansible.

Индивидуальный вариант: 9.

1. Ansible. Inventory

Создадим файл myhosts, который будет содержать inventory, в котором есть одна группа с именем group1 и хост к которому мы подключаемся. Выполнение задачи представлено на Листинге 1.1.

Листинг 1.1 – Создание файла myhosts

```
root@LAPTOP-UCPRRTEQ:/home/khan# nano myhosts
root@LAPTOP-UCPRRTEQ:/home/khan# cat myhosts
[group1]
localhost ansible_connection=local
ansible_python_interpreter=/usr/bin/python3
```

Что такое inventory у ansible? Это файл в котором записаны все сервера, которыми нужно управлять через Ansible.

Как будет выглядеть inventory если добавить host02 в группу group1 и группу group2? Ответ представлен на Листинге 1.2.

Листинг 1.2 – Inventory

```
[group1]
host01
host02

[group2]
```

2. Ansible. Выполнение команд

С помощью ansible можно выполнять простые действия при помощи одной команды. Например, такое действие как ping — проверить наличие доступа у ansible до указанного хоста.

Для этого, выполним команду ansible all -i myhosts -m ping (Листинг 2.1).

Листинг 2.1 – Выполнение команды ping

```
root@LAPTOP-UCPRRTEQ:/home/khan# ansible all -i myhosts -m ping
localhost | SUCCESS => {
    "changed": false,
    "ping": "pong"
```

```
}
```

Так как все прошло хорошо, то Ansible вывел текст с меткой SUCCESS.

Предыдущее действие выполнялось при помощи модуля (опция -m) ping. При помощи модуля command можно выполнять любую команду командной строки, для этого используется параметр -a (аргумент который передается модулю).

Выполним любую известную команду linux (Листинг 2.2).

Листинг 2.2 – Выполнение команды date

```
root@LAPTOP-UCPRRTEQ:/home/khan# ansible group1 -i myhosts -m command -adate
localhost | CHANGED | rc=0 >>
Wed Apr 13 18:02:10 MSK 2022
```

Если в команде есть пробелы, то ее необходимо заключать в кавычки (Листинг 2.3).

Листинг 2.3 – Выполнение команды, в которой есть пробелы

```
root@LAPTOP-UCPRRTEQ:/home/khan# ansible group1 -i myhosts -m command -a
"ls -la"
localhost | CHANGED | rc=0 >>
total 72
drwxr-xr-x 8 khan     khan     4096 Apr 13 17:58 .
drwxr-xr-x 3 root     root     4096 Mar 25 10:30 ..
drwxr-xr-x 4 khan     khan     4096 Apr 12 20:03 .ansible
lrwxrwxrwx 1 khan     khan       23 Mar 25 10:37 .aws ->
/mnt/c/Users/nasty/.aws
lrwxrwxrwx 1 khan     khan       25 Mar 25 10:37 .azure ->
/mnt/c/Users/nasty/.azure
-rw------- 1 khan     khan     8359 Apr 13 17:44 .bash_history
-rw-r--r-- 1 khan     khan      220 Mar 25 10:30 .bash_logout
-rw-r--r-- 1 khan     khan     3771 Mar 25 10:30 .bashrc
drwxr-xr-x 5 khan     khan     4096 Mar 26 10:42 .docker
drwxr-xr-x 2 khan     khan     4096 Mar 25 10:30 .landscape
drwxr-xr-x 3 khan     khan     4096 Apr  1 15:04 .local
-rw-r--r-- 1 khan     khan        0 Apr 13 11:55 .motd_shown
-rw-r--r-- 1 khan     khan      807 Mar 25 10:30 .profile
-rw-r--r-- 1 khan     khan        0 Mar 26 09:27 .sudo_as_admin_successful
-rw------- 1 khan     khan     7292 Mar 26 14:01 .viminfo
-rw-r--r-- 1 khan     khan      185 Mar 26 14:01 Dockerfile
drwxr-xr-x 2 root     root     4096 Apr  1 15:07 data
drwxr-xr-x 2 khan     khan     4096 Mar 26 09:59 myfiles
-rw-r--r-- 1 root     root       88 Apr 13 17:58 myhosts
```

3. Ansible playbook

Создадим файл webserver.yml (Листинг 3.1).

Листинг 3.1 – Создание файла

```
root@LAPTOP-UCPRRTEQ:/home/khan# nano webserver.yml
root@LAPTOP-UCPRRTEQ:/home/khan# cat webserver.yml
---
- name: Web server installed
  hosts: group1
  become: yes
  tasks:
  - name: latest nginx version installed
    apt:
      name: nginx-light
      state: latest
      update_cache: true
```

Выполним команду ansible-playbook -i myhosts webserver.yml (Листинг 3.2).

Листинг 3.2 – Выполнение указанной команды

```
root@LAPTOP-UCPRRTEQ:/home/khan# ansible-playbook -i myhosts
webserver.yml

PLAY [Web server installed]
*********************************************************************************
TASK [Gathering Facts]
*********************************************************************************
ok: [localhost]
TASK [latest nginx version installed]
*********************************************************************************
ok: [localhost]
PLAY RECAP
*********************************************************************************
localhost                  : ok=2    changed=0    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
```

Она исполняет только что созданный playbook. В самом playbook’е используется модуль apt, который нужен для взаимодействия с менеджером пакетов. После запуска этой команды в системе должен установиться указанный пакет (в нашем случае – nginx-light).

Откроем порт 80 на хосте. Нас приветствует стандартная домашняя страница nginx (Рисунок 3.1).

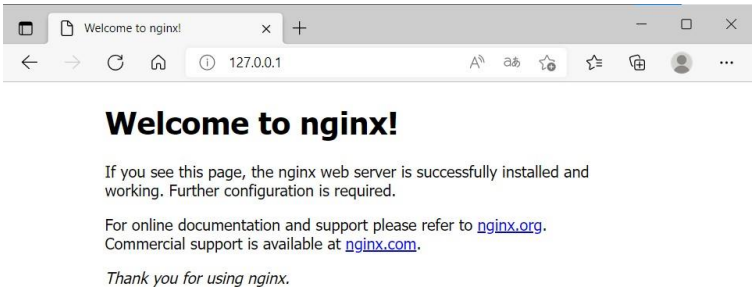

Рисунок 3.1 – Домашняя страница nginx

Исполним playbook еще раз (Листинг 3.3, Рисунок 3.2).

Листинг 3.3 – Повторное исполнение playbook’а

```
root@LAPTOP-UCPRRTEQ:/home/khan# ansible-playbook -i myhosts
webserver.yml

PLAY [Web server installed]
*********************************************************************************
TASK [Gathering Facts]
*********************************************************************************
ok: [localhost]
TASK [latest nginx version installed]
*********************************************************************************
ok: [localhost]
PLAY RECAP
*********************************************************************************
localhost                  : ok=2    changed=0    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
```

Рисунок 3.2 – Повторное открытие хоста

Что изменилось при повторном запуске? Изменений не произошло.

В каком статусе оказались задачи и почему так? Все задачи сохранили статус «ok», как при первом запуске. Так как Ansible сам по себе декларативный, то есть, мы не указываем в плейбуке, что ему нужно сделать, мы указываем, что хотим видеть. В итоге это означает не «установить nginx», а «пакет nginx должен быть в apt в таком-то state-е». По умолчанию state – present. Это значит, что если у нас уже есть какой-то пакет в Ansible, то state у него, соответственно, будет неизменным. Мы используем state – latest, в таком случае, если apt найдет какую-то более свежую версию пакета, то он попробует ее обновить.

Добавим task к playbook'у и исполним его (Листинги 3.4-3.5).

Листинг 3.4 – Добавление task'а

```
 - name: nginx enabled and running
   service:
     name: nginx
     enabled: true
     state: started
```

Листинг 3.5 – Повторное исполнение playbook'а

```
root@LAPTOP-UCPRRTEQ:/home/khan# ansible-playbook -i myhosts
webserver.yml

PLAY [Web server installed]
*********************************************************************************
TASK [Gathering Facts]
*********************************************************************************
ok: [localhost]
TASK [latest nginx version installed]
*********************************************************************************
ok: [localhost]
TASK [nginx enabled and running]
*********************************************************************************
ok: [localhost]
PLAY RECAP
*********************************************************************************
localhost                  : ok=3    changed=0    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
```

Модуль service взаимодействует с системными сервисами. Этот task удостоверяется, что сервис nginx запущен и включен по умолчанию.

Были ли произведены какие-либо изменения в системе при выполнении второго таска? По логу видно, что добавилась обработка созданной задачи (результат: успешное выполнение).

Какие есть состояния у задач после выполнения в ansible?

- ok – задача выполнена
- changed – задача что-то изменила
- unreachable – задача не достигла удаленной системы
- failed – задача провалилась при настройке
- skipped – пропущенная задача
- rescued – восстановленная задача
- ignored – проигнорированная задача.

4. Переменные и шаблоны

Создадим файл index.html.j2 (Листинг 4.1).

Листинг 4.1 – Создание файла

```
root@LAPTOP-UCPRRTEQ:/home/khan# nano index.html.j2
root@LAPTOP-UCPRRTEQ:/home/khan# cat index.html.j2
<html>
<head>
<title>Hello, world</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

```
</head>
<body>
<h1>{{ my_name }}</h1>
</body>
```

Добавим task к playbook’у (Листинг 4.2).

Листинг 4.2 – Добавление task’а

```
  - name: correct index.html is present
    template:
      src: index.html.j2
      dest: /var/www/html/index.html
```

Добавим блок к inventory (Листинг 4.3).

Листинг 4.3 – Изменение inventory

```
[all:vars]
my_name=Default student name
```

Исполним playbook и откроем порт 80, увидим текст «Default student name» (Листинг 4.4, Рисунок 4.1).

Листинг 4.4 – Исполнение playbook’а

```
root@LAPTOP-UCPRRTEQ:/home/khan# ansible-playbook -i myhosts
webserver.yml

PLAY [Web server installed]
*********************************************************************************
TASK [Gathering Facts]
*********************************************************************************
ok: [localhost]
TASK [latest nginx version installed]
*********************************************************************************
ok: [localhost]
TASK [nginx enabled and running]
*********************************************************************************
ok: [localhost]
TASK [correct index.html is present]
*********************************************************************************
changed: [localhost]
PLAY RECAP
*********************************************************************************
localhost                  : ok=4    changed=1    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
```

Default student name

Рисунок 4.1 – Экранная форма содержимого страницы localhost

Добавим блок с переменными к playbook’у (удобнее размещать его выше секции tasks), указав свое имя и фамилию в качестве значения (Листинг 4.5).

Листинг 4.5 – Добавление блока с переменными

```
vars:
  my_name: Khan Anastasiia
```

Исполним playbook и обновим страницу по порту 80, мы должны увидеть добавленные фамилию и имя (Рисунок 4.2).

Рисунок 4.2 – Экранная форма содержимого страницы localhost

Теперь переопределим переменную my_name из командной строки, это делается при помощи параметра –e, укажем в качестве значения номер студенческого билета (Листинг 4.6, Рисунок 4.3).

Листинг 4.6 – Переопределение переменной из командной строки

```
root@LAPTOP-UCPRRTEQ:/home/khan# ansible-playbook -i myhosts -e
'my_name="20И1675"' webserver.yml

PLAY [Web server installed]
*********************************************************************************
TASK [Gathering Facts]
*********************************************************************************
ok: [localhost]
TASK [latest nginx version installed]
*********************************************************************************
ok: [localhost]
TASK [nginx enabled and running]
*********************************************************************************
ok: [localhost]
TASK [correct index.html is present]
*********************************************************************************
changed: [localhost]
PLAY RECAP
*********************************************************************************
localhost                  : ok=4    changed=1    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
```

Рисунок 1.4.3 – Экранная форма содержимого страницы localhost

В каком порядке переопределялись переменные? Переменная my_name была определена в html-файле, затем определена в inventory, переопределена в playbook и снова переопределена через командную строку.

5. Conditionals и handlers

Создадим второй playbook, в нем мы создадим конфигурацию веб-сервера для отображения некоторого текста по порту 8080. Причем, сделаем так, что ее можно включать или выключать.

Создадим playbook, называющийся web-8080.yml (Листинг 5.1).

Листинг 5.1 – Создание playbook'а

```
root@LAPTOP-UCPRRTEQ:/home/khan# nano web-8080.yml
root@LAPTOP-UCPRRTEQ:/home/khan# cat web-8080.yml
---
- name: manage httpd.conf
  hosts: group1
  become: true

  tasks:
  - name: Copy web 8080 configuration file
    copy:
      src: web-8080.conf
      dest: /etc/nginx/sites-enabled/
    when: has_8080
```

Добавим файл web-8080.conf (Листинг 5.2).

Листинг 5.2 – Добавление файла web-8080.conf

```
root@LAPTOP-UCPRRTEQ:/home/khan# nano web-8080.conf
root@LAPTOP-UCPRRTEQ:/home/khan# cat web-8080.conf
server {
  listen 8080;
  location / {
    return 200 'Some text';
    add_header Content-Type 'text/plain; charset=utf-8';
  }
}
```

Добавим переменную has_8080 со значением True в секцию переменных в inventory (Листинг 5.3).

Листинг 5.3 – Добавление переменной в инвентарь

```
[all:vars]
my_name=Default student name
has_8080=True
```

Исполним playbook (Листинг 5.4).

Листинг 5.4 – Исполнение playbook'а

```
root@LAPTOP-UCPRRTEQ:/home/khan# ansible-playbook -i myhosts web-8080.yml

PLAY [manage httpd.conf]
*********************************************************************
TASK [Gathering Facts]
```

```
*****************************************************************************
ok: [localhost]
TASK [Copy web 8080 configuration file]
*****************************************************************************
changed: [localhost]
PLAY RECAP
*****************************************************************************
localhost                  : ok=2    changed=1    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
```

Какой статус имеет задача «Copy web 8080 configuration file»? changed.

Поменяем значение переменной на True и исполним playbook. Откроем порт 8080 на хосте в браузере (Рисунок 5.1).

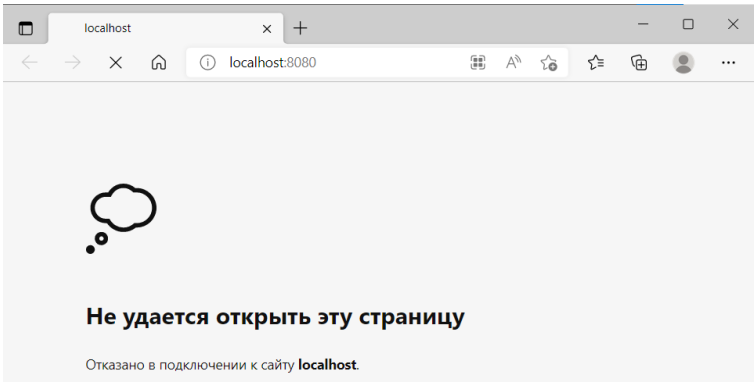

Рисунок 5.1 – Открытие в браузере

Была ли скопирована новая конфигурация? Нет.

Работает ли сервер на порту 8080? Нет.

Для того, чтобы наши изменения применились необходимо сообщить сервису nginx об изменениях. Это делается при помощи команды `systemctl reload nginx`. Но мы сделаем все в Ansible при помощи handler'а – в случае если изменяется конфигурация, будет посылаться сигнал reload. Для этого обновим содержимое playbook'а (Листинг 5.5).

Листинг 5.5 – Изменение playbook'а

```
    notify:
      - reload_nginx
  handlers:
  - name: reload_nginx
    service:
      name: nginx
      state: reloaded
```

Исполним playbook (Листинг 5.6).

Листинг 5.6 – Исполнение playbook'а

```
root@LAPTOP-UCPRRTEQ:/home/khan# ansible-playbook -i myhosts web-8080.yml
```

```
PLAY [manage httpd.conf]
*********************************************************************************
TASK [Gathering Facts]
*********************************************************************************
ok: [localhost]
TASK [Copy web 8080 configuration file]
*********************************************************************************
ok: [localhost]
PLAY RECAP
*********************************************************************************
localhost                  : ok=2    changed=0    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
```

Какой элемент добавился в выводе? Задача «Copy web 8080 configuration file» получила статус «ok».

Был ли перезапущен сервис nginx? Нет.

Для того, чтобы наш handler исполнился, начнем все сначала – удалим конфигурацию, которая была скопирована (Листинг 5.7).

Листинг 5.7 – Удаление конфигурации

```
root@LAPTOP-UCPRRTEQ:/home/khan# rm -r /etc/nginx/sites-enabled/web-
8080.conf
```

Исполним playbook (Листинг 5.8, Рисунок 5.2).

Листинг 5.8 – Исполнение playbook'а

```
root@LAPTOP-UCPRRTEQ:/home/khan# ansible-playbook -i myhosts web-8080.yml

PLAY [manage httpd.conf]
*********************************************************************************
TASK [Gathering Facts]
*********************************************************************************
ok: [localhost]
TASK [Copy web 8080 configuration file]
*********************************************************************************
changed: [localhost]
RUNNING HANDLER [reload_nginx]
*********************************************************************************
changed: [localhost]
PLAY RECAP
*********************************************************************************
localhost                  : ok=3    changed=2    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
```

Рисунок 5.2 – Экранная форма содержимого страницы localhost

Что поменялось в выводе по сравнению с первым запуском? Исполнена задача по копированию конфигурации.

Работает ли сервер теперь? Да.

Заменим «Some text» в конфигурации веб-сервера на свое имя и фамилию и исполним playbook (Листинги 5.9-5.10, Рисунок 5.3).

Листинг 5.9 – Замена текста в конфигурации

```
root@LAPTOP-UCPRRTEQ:/home/khan# nano web-8080.conf
root@LAPTOP-UCPRRTEQ:/home/khan# cat web-8080.conf
server {
  listen 8080;
  location / {
    return 200 'Khan Anastasiia';
    add_header Content-Type 'text/plain; charset=utf-8';
  }
}
```

Листинг 5.10 – Исполнение playbook'а

```
root@LAPTOP-UCPRRTEQ:/home/khan# ansible-playbook -i myhosts web-8080.yml

PLAY [manage httpd.conf]
**********************************************************************
TASK [Gathering Facts]
**********************************************************************
ok: [localhost]
TASK [Copy web 8080 configuration file]
**********************************************************************
changed: [localhost]
RUNNING HANDLER [reload_nginx]
**********************************************************************
changed: [localhost]
PLAY RECAP
**********************************************************************
localhost                  : ok=3    changed=2    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
```

Рисунок 5.3 – Экранная форма содержимого страницы localhost

Был ли сервер перезапущен? Да.

Что такое handler в ansible? В каких случаях он исполняется? Обработчик – это список задач, на которые ссылается имя, они ничем не отличаются от общих задач. Обработчики уведомляются уведомителем (notify). Если не уведомлено, обработчики не будут выполнены. Независимо от того, сколько уведомителей уведомят, обработчики будут выполняться только один раз после выполнения всех задач.

6. Ansible & docker

Создадим файл docker.yml (Листинг 6.1).

Листинг 6.1 – Создание файла docker.yml

```
root@LAPTOP-UCPRRTEQ:/home/khan# nano docker.yml
root@LAPTOP-UCPRRTEQ:/home/khan# cat docker.yml
---
- name: manage docker
  hosts: group1
  become: true

  vars:
    drupal_port: 8090

  tasks:
  - name: Install docker and some dependencies
    apt:
     name: python3-pip, docker-ce
     state: present

  - name: Start docker service
    service:
      name: docker
      state: started

  - name: Install docker python module
    pip:
      name: docker

  - name: Create Drupal container
    docker_container:
      name: drupal
      image: drupal
      state: started
      ports:
        - "{{ drupal_port }}:80"
      volumes:
        - drupal_modules:/var/www/html/modules
        - drupal_profiles:/var/www/html/profiles
        - drupal_themes:/var/www/html/themes
        - drupal_sites:/var/www/html/sites
```

Данный playbook убеждается, что docker установлен и запущен, а также установлены модули для работы с ним. Затем, создает и запускает контейнер из образа drupal.

Исполним playbook и откроем порт 8090 (Листинг 6.2, Рисунок 6.1).

Листинг 6.2 – Исполнение playbook'а

```
root@LAPTOP-UCPRRTEQ:/home/khan# ansible-playbook -i myhosts docker.yml

PLAY [manage docker]
*********************************************************************
TASK [Gathering Facts]
*********************************************************************
ok: [localhost]
TASK [Install docker and some dependencies]
*********************************************************************
changed: [localhost]
```

```
TASK [Start docker service]
*********************************************************************************
changed: [localhost]
TASK [Install docker python module]
*********************************************************************************
changed: [localhost]
TASK [Create Drupal container]
*********************************************************************************
changed: [localhost]
PLAY RECAP
*********************************************************************************
localhost                  : ok=5    changed=4    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
```

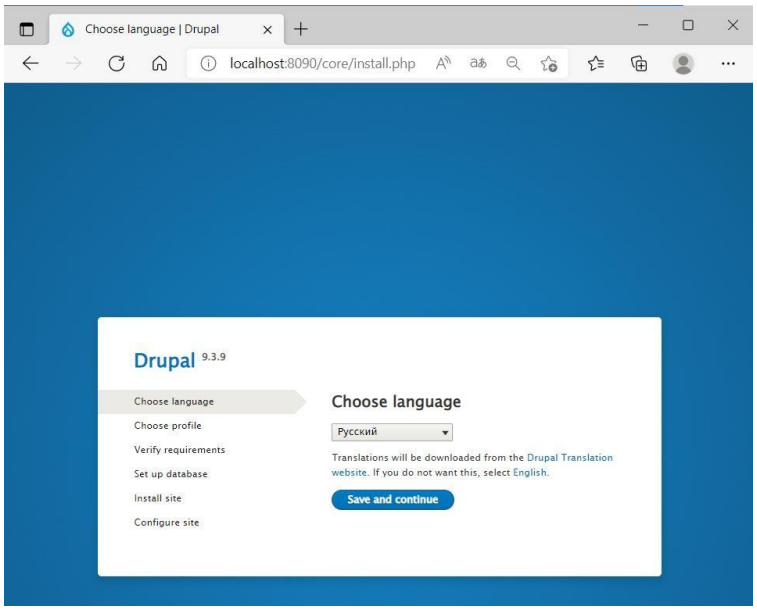

Рисунок 6.1 – Экранная форма содержимого страницы localhost

Затем исполним playbook второй раз (Листинг 6.3, Рисунок 6.2).

Листинг 6.3 – Повторное исполнение playbook'а

```
root@LAPTOP-UCPRRTEQ:/home/khan# ansible-playbook -i myhosts docker.yml

PLAY [manage docker]
*********************************************************************************
TASK [Gathering Facts]
*********************************************************************************
ok: [localhost]
TASK [Install docker and some dependencies]
*********************************************************************************
ok: [localhost]
TASK [Start docker service]
*********************************************************************************
ok: [localhost]
TASK [Install docker python module]
*********************************************************************************
ok: [localhost]
TASK [Create Drupal container]
*********************************************************************************
ok: [localhost]
PLAY RECAP
*********************************************************************************
```

```
localhost                  : ok=5    changed=0    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
```

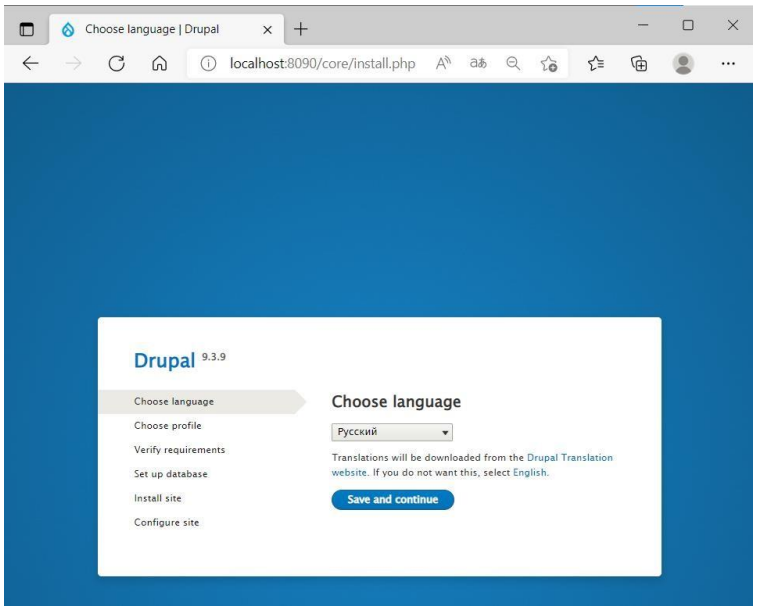

Рисунок 6.2 – Экранная форма содержимого страницы localhost

Был ли пересоздан контейнер? Нет, контейнер не был пересоздан.

Поменяем порт на 8000 + 28, исполним playbook и откроем этот порт (Листинг 6.4, Рисунок 6.3).

Листинг 6.3 – Изменение порта и исполнение playbook'а

```
root@LAPTOP-UCPRRTEQ:/home/khan# nano docker.yml
root@LAPTOP-UCPRRTEQ:/home/khan# cat docker.yml
---
- name: manage docker
  hosts: group1
  become: true

  vars:
    drupal_port: 8028

  tasks:
  - name: Install docker and some dependencies
    apt:
     name: python3-pip, docker-ce
     state: present

  - name: Start docker service
    service:
      name: docker
      state: started

  - name: Install docker python module
    pip:
      name: docker

  - name: Create Drupal container
    docker_container:
      name: drupal
```

```
      image: drupal
      state: started
      ports:
        - "{{ drupal_port }}:80"
      volumes:
        - drupal_modules:/var/www/html/modules
        - drupal_profiles:/var/www/html/profiles
        - drupal_themes:/var/www/html/themes
        - drupal_sites:/var/www/html/sites

root@LAPTOP-UCPRRTEQ:/home/khan# ansible-playbook -i myhosts docker.ymlPLAY

[manage docker]
*********************************************************************************
TASK [Gathering Facts]
*********************************************************************************
ok: [localhost]
TASK [Install docker and some dependencies]
*********************************************************************************
ok: [localhost]
TASK [Start docker service]
*********************************************************************************
ok: [localhost]
TASK [Install docker python module]
*********************************************************************************
ok: [localhost]
TASK [Create Drupal container]
*********************************************************************************
changed: [localhost]
PLAY RECAP
*********************************************************************************
localhost                  : ok=5    changed=1    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
```

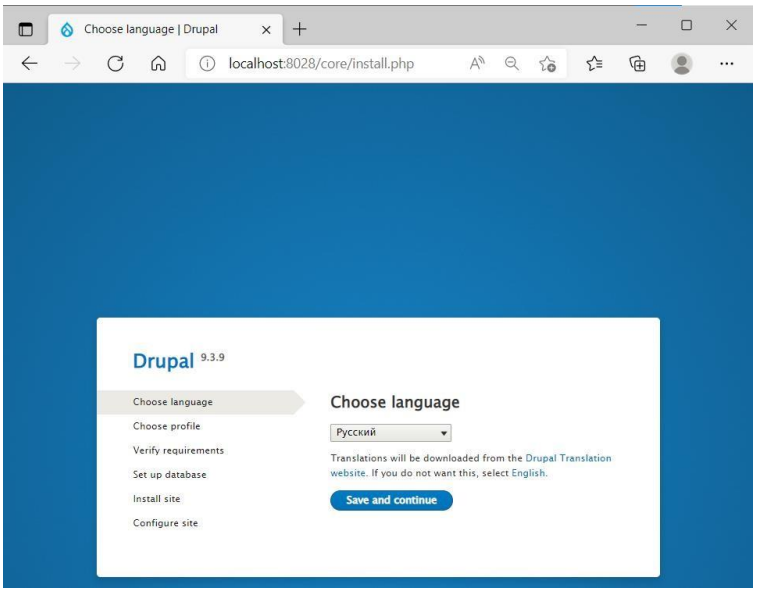

Рисунок 6.3 – Экранная форма содержимого страницы localhost

Был ли пересоздан контейнер теперь? Да, контейнер был пересоздан.

7. Индивидуальное задание

Написать playbook, который устанавливает пакет, согласно варианту.

Пакет: 9. mysql-client

Создадим файл my_playbook.yml (Листинг 7.1).

Листинг 7.1 – Создание playbook'а

```
root@LAPTOP-UCPRRTEQ:/home/khan# cat my_playbook.yml
---
- name: install package mysql-client
  hosts: group1
  become: yes

  tasks:
  - name: Install package mysql-client
    apt:
     name: mysql-client
     state: latest
     update_cache: true
```

Исполним playbook (Листинг 7.2).

Листинг 7.1 – Исполнение playbook'а

```
root@LAPTOP-UCPRRTEQ:/home/khan# ansible-playbook -i myhosts
my_playbook.yml

PLAY [install package mysql-client]
*********************************************************************************
TASK [Gathering Facts]
*********************************************************************************
ok: [localhost]
TASK [Install package mysql-client]
*********************************************************************************
changed: [localhost]
PLAY RECAP
*********************************************************************************
localhost                  : ok=2    changed=1    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
```

По логу видно, что пакет успешно установлен.

Вывод

В ходе выполнения практической работы были изучены следующие аспекты Ansible: Inventory, выполнение команд, файлы playbook, переменные и шаблоны, conditionals и handlers, взаимодействие Ansible и Docker.

Список информационных источников

1. Методические указания для выполнения практической работы №4 «Ansible» М. А. Овчинникова, МИРЭА – Российский Технологический Университет, 2022.

2. Русские блоги – Текст // Электронный. – URL: https://russianblogs.com/article/2554211218/ (дата обращения 13.04.2022).

3. INFOIT.COM.UA – Текст // Электронный. – URL: https://infoit.com.ua/linux/kak-ustanovit-ansible-v-ubuntu-20-04-lts/ (дата обращения 13.04.2022).

4. Хабр – Текст // Электронный. – URL: https://habr.com/ru/company/southbridge/blog/569172/ (дата обращения 13.04.2022).