



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: «Системы конфигурационного управления»

Выполнил:

Студент группы ИНБО-04-19

Подольский К.В.

Проверил:

ассистент Борzych Н.Ю.

2021 г

Вариант 12

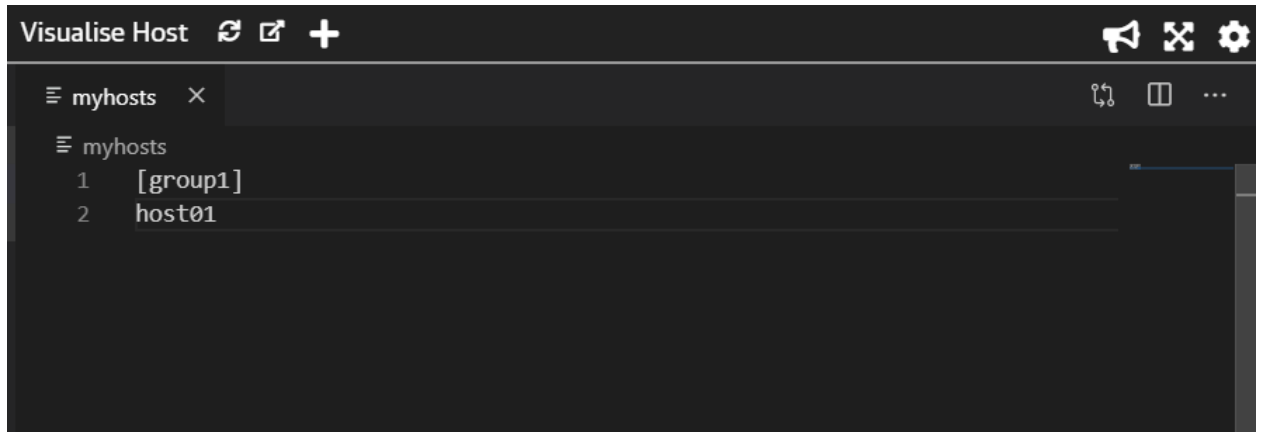


Рис. 1 – Создан файл myhosts

Что такое inventory у ansible? Как будет выглядеть inventory если добавить host02 в группу group1 и группу group2?

Ansible inventory – это файл, содержащий список устройств, групп устройств и переменных (vars).

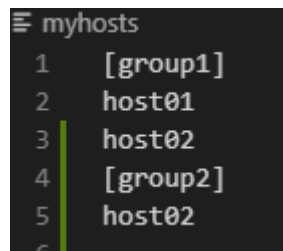


Рис. 2 – inventory файл

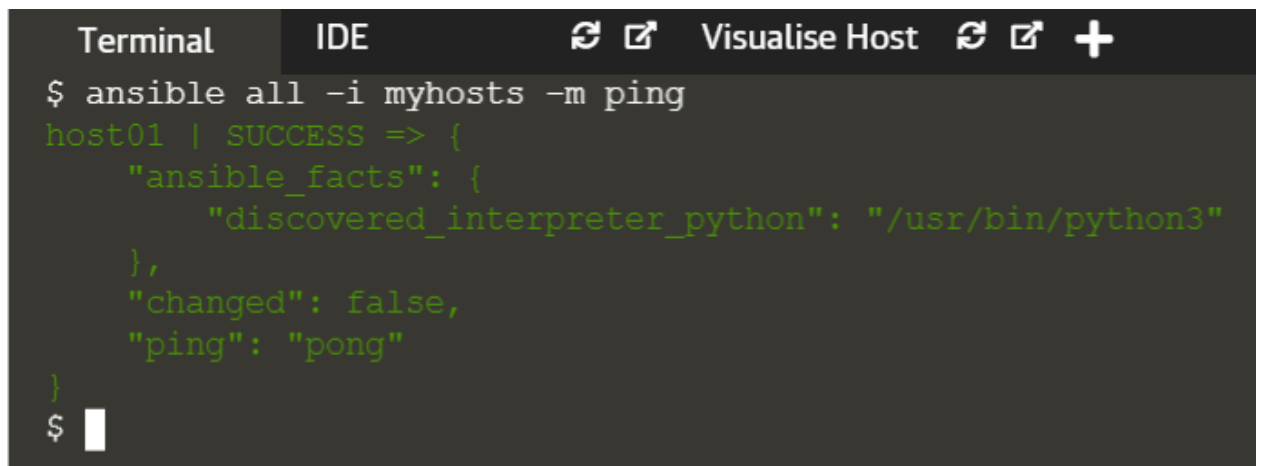


Рис. 3 – Проверка наличия доступа

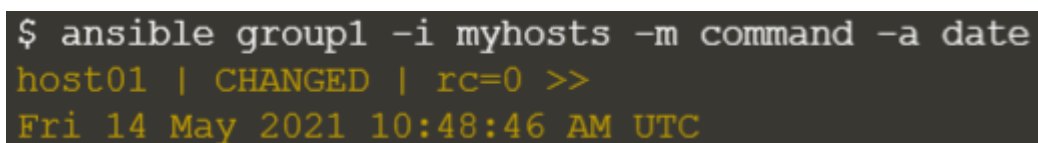


Рис. 4 – Вывод текущей даты

```
! webserver.yml X
! webserver.yml
1 - name: Web server installed
2   hosts: group1
3   become: yes
4   tasks:
5     - name: latest nginx version installed
6       apt:
7         name: nginx-light
8         state: latest
9         update_cache: true
```

Рис. 5 – Создан файл webserver.yml

```
Terminal IDE Visualise Host
$ rm /etc/apt/sources.list.d/yarn.list
$ ansible-playbook -i myhosts webserver.yml

PLAY [Web server installed] *****

TASK [Gathering Facts] *****
ok: [host01]

TASK [latest nginx version installed] *****
changed: [host01]

PLAY RECAP *****
host01 : ok=2 changed=1 unreachable=0 failed=0 skipped=0 rescued=0 ignored
=0
$
```

Рис. 6 – Удаление файла yarn.lock и исполнение playbook

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Рис. 7 – Открытие 80 порта

```
$ ansible-playbook -i myhosts webserver.yml

PLAY [Web server installed] *****

TASK [Gathering Facts] *****
ok: [host01]

TASK [latest nginx version installed] *****
ok: [host01]

PLAY RECAP *****
host01 : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Рис. 8 – Повторное исполнение playbook

Что изменилось при повторном запуске? В каком статусе оказались задачи и почему так?

При повторном запуске все статусы задач стали «ok», потому что между исполнениями playbook не было внесено никаких изменений

```
! webserver.yml
1  ---
2  - name: Web server installed
3    hosts: group1
4    become: yes
5    tasks:
6      - name: latest nginx version installed
7        apt:
8          name: nginx-light
9          state: latest
10         update_cache: true
11      - name: nginx enabled and running
12        service:
13          name: nginx
14          enabled: true
15          state: started
```

Рис. 9 – Добавление task'а к playbook

```
$ ansible-playbook -i myhosts webserver.yml

PLAY [Web server installed] *****

TASK [Gathering Facts] *****
ok: [host01]

TASK [latest nginx version installed] *****
ok: [host01]

TASK [nginx enabled and running] *****
ok: [host01]

PLAY RECAP *****
host01 : ok=3    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

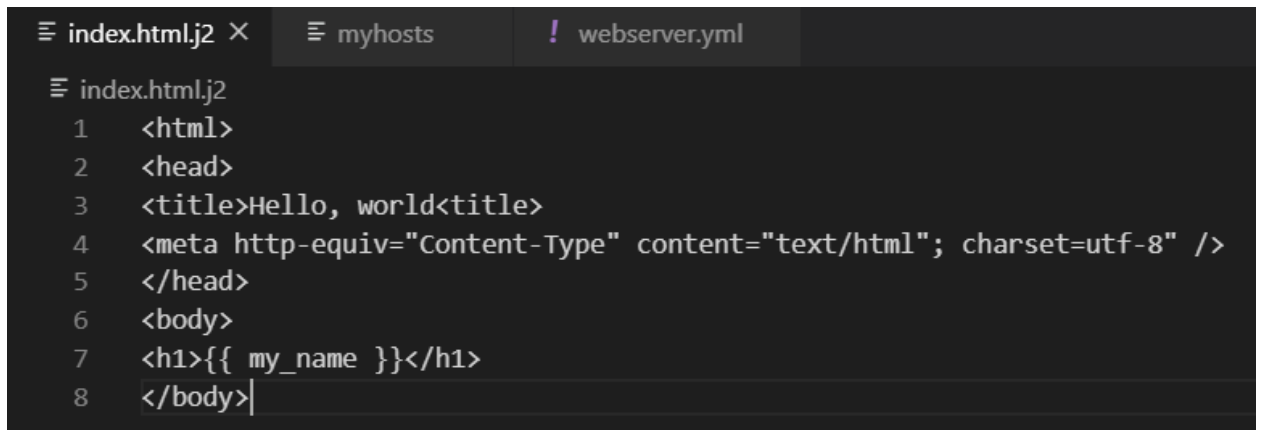
Рис. 10 – Повторное исполнение playbook с новым taskом

Были ли произведены какие-либо изменения в системе при выполнении второго таска?

Не было

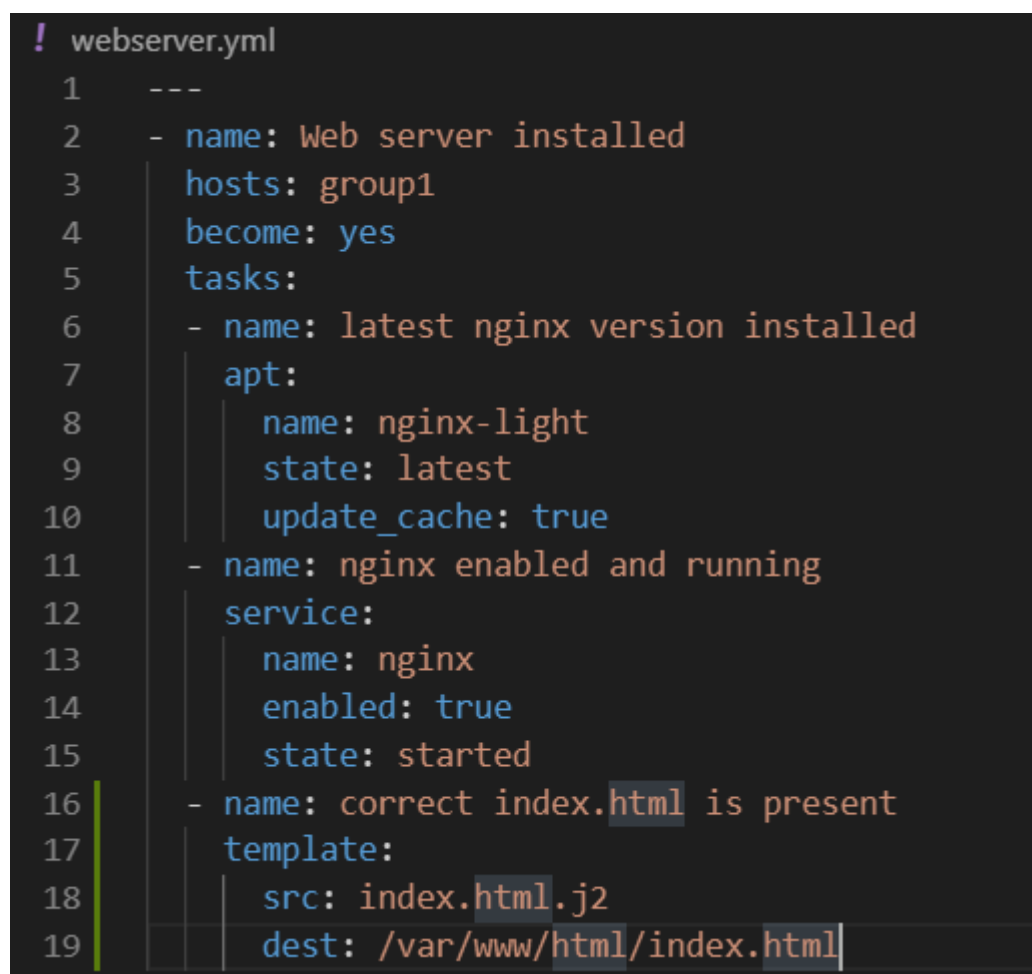
Какие есть состояния у задач после выполнения в ansible?

Ok, change, unreachable, failed, skipped, rescued, ignored

A screenshot of a code editor with a dark theme. At the top, there are three tabs: 'index.html.j2' (active), 'myhosts', and 'webserver.yml'. The 'index.html.j2' tab is selected, and it shows an HTML template. The code is as follows:

```
1 <html>
2 <head>
3 <title>Hello, world</title>
4 <meta http-equiv="Content-Type" content="text/html"; charset=utf-8" />
5 </head>
6 <body>
7 <h1>{{ my_name }}</h1>
8 </body>
```

Рис. 11 – Создание файла index.html.j2

A screenshot of a code editor with a dark theme. The 'webserver.yml' tab is selected, and it shows an Ansible playbook. The code is as follows:

```
! webserver.yml
1 ---
2 - name: Web server installed
3   hosts: group1
4   become: yes
5   tasks:
6     - name: latest nginx version installed
7       apt:
8         name: nginx-light
9         state: latest
10        update_cache: true
11    - name: nginx enabled and running
12      service:
13        name: nginx
14        enabled: true
15        state: started
16    - name: correct index.html is present
17      template:
18        src: index.html.j2
19        dest: /var/www/html/index.html
```

Рис. 12 – Добавление нового таска

```

≡ myhosts
1  [group1]
2  host01
3  [all:vars]
4  my_name=Default student name

```

Рис. 13 – Добавление блока к inventory

```

$ ansible-playbook -i myhosts webserver.yml

PLAY [Web server installed] *****

TASK [Gathering Facts] *****
ok: [host01]

TASK [latest nginx version installed] *****
ok: [host01]

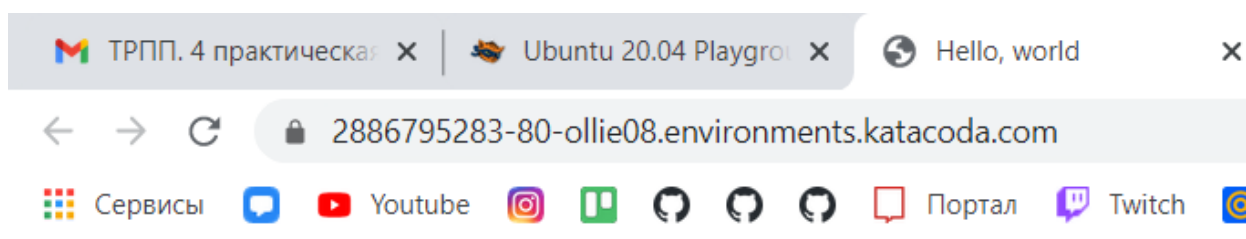
TASK [nginx enabled and running] *****
ok: [host01]

TASK [correct index.html is present] *****
changed: [host01]

PLAY RECAP *****
host01 : ok=4  changed=1  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0

```

Рис. 14 – Исполнение playbook с новым таском



Default student name

Рис. 15 – Открытие 80 порта

Kirill Podolskiy

Рис. 16 – Вывод имени и фамилии

```

$ ansible-playbook -i myhosts -e 'my_name="19I0623"' webserver.yml

PLAY [Web server installed] *****

TASK [Gathering Facts] *****
ok: [host01]

TASK [latest nginx version installed] *****
ok: [host01]

TASK [nginx enabled and running] *****
ok: [host01]

TASK [correct index.html is present] *****
changed: [host01]

PLAY RECAP *****
host01 : ok=4    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

```

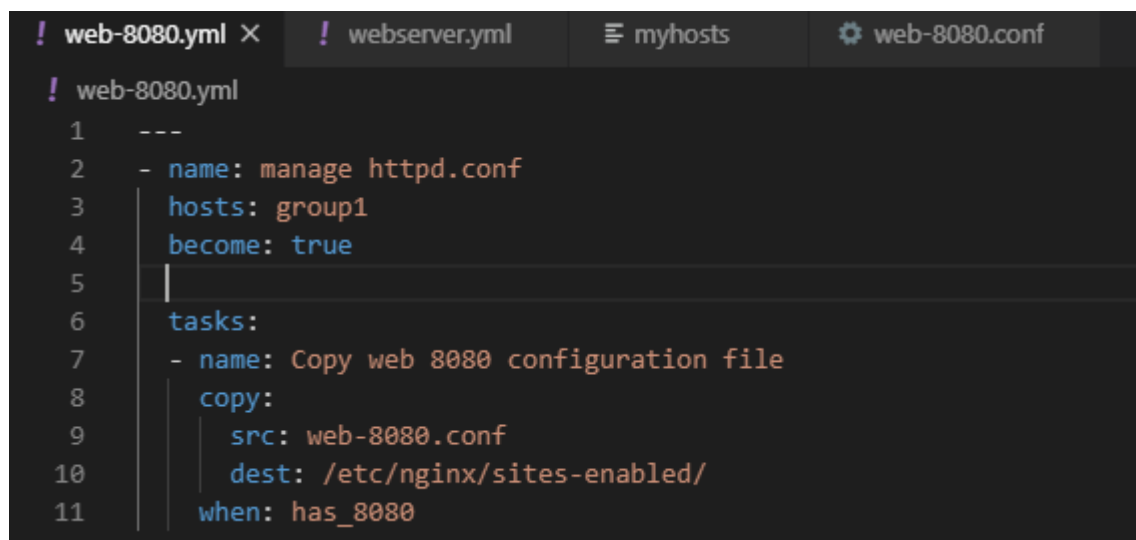
Рис. 17 – Переопределение my_name

19I0623

Рис. 18 – Вывод номера студенческого билета

В каком порядке переопределялись переменные?

**Поле с переменными в Inventory - Поле с переменными в playbook -
Параметры команды**



```

! web-8080.yml x  ! webserver.yml  myhosts  web-8080.conf
! web-8080.yml
1  ---
2  - name: manage httpd.conf
3    hosts: group1
4    become: true
5
6  tasks:
7    - name: Copy web 8080 configuration file
8      copy:
9        src: web-8080.conf
10       dest: /etc/nginx/sites-enabled/
11       when: has_8080

```

Рис. 19 – Создание нового playbook

```

web-8080.conf
1  server {
2      listen 8080;
3
4      location / {
5          return 200 'Some text';
6          add_header Content-Type 'text/plain; charset=utf-8';
7      }
8  }

```

Рис. 20 – Создание файла web-8080.conf

```

1  [group1]
2  host01
3  [all:vars]
4  my_name=Default student name
5  has_8080=True

```

Рис. 21 – Добавление новой переменной в inventory

```

$ ansible-playbook -i myhosts web-8080.yml

PLAY [manage httpd.conf] *****
TASK [Gathering Facts] *****
ok: [host01]

TASK [Copy web 8080 configuration file] *****
changed: [host01]

PLAY RECAP *****
host01 : ok=2  changed=1  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0

$ 

```

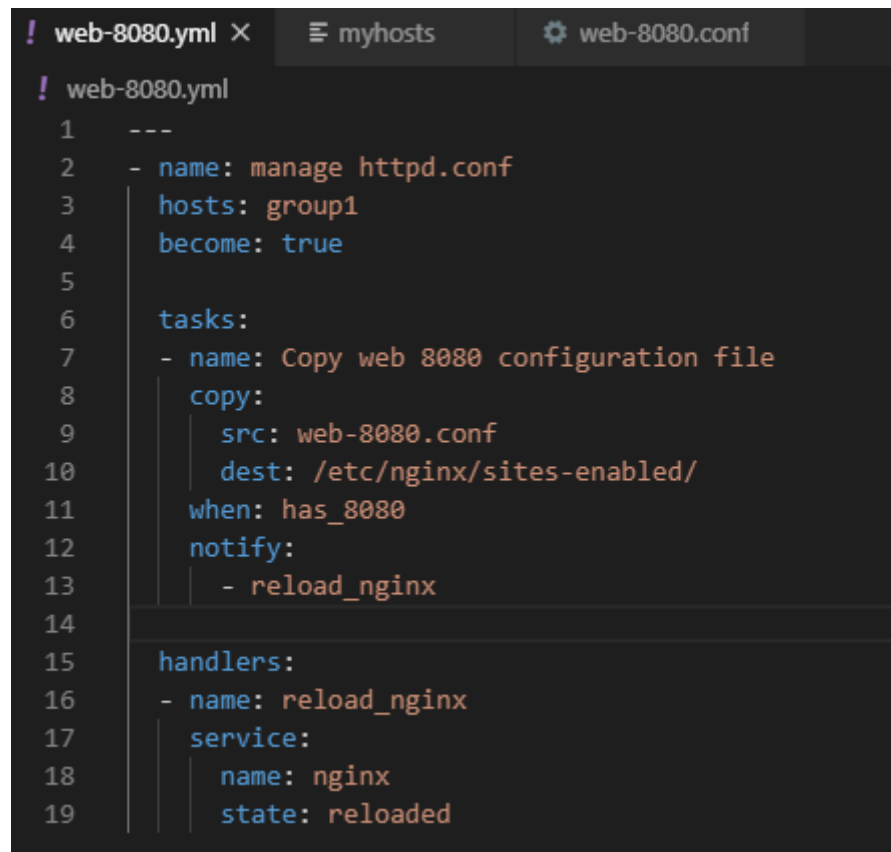
Рис. 22 – Исполнение playbook

Какой статус имеет задача “Copy web 8080 configuration file”?

Changed

Была ли скопирована новая конфигурация? Работает ли сервер на порту 8080?

Была скопирована, сервер не работает

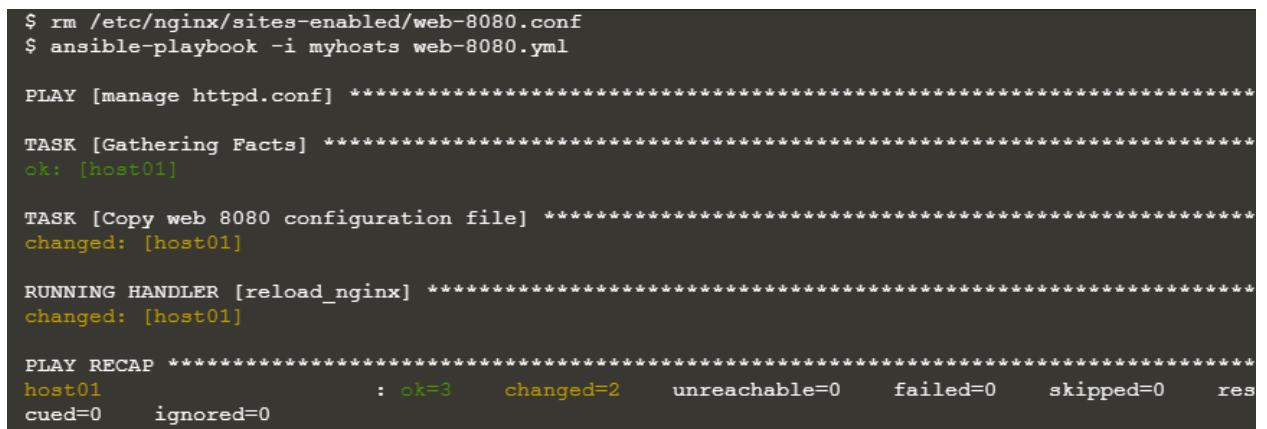


```
! web-8080.yml X myhosts web-8080.conf
! web-8080.yml
1 ---
2 - name: manage httpd.conf
3   hosts: group1
4   become: true
5
6   tasks:
7     - name: Copy web 8080 configuration file
8       copy:
9         src: web-8080.conf
10        dest: /etc/nginx/sites-enabled/
11        when: has_8080
12        notify:
13          - reload_nginx
14
15      handlers:
16        - name: reload_nginx
17          service:
18            name: nginx
19            state: reloaded
```

Рис. 23 – Изменение playbook

Какой элемент добавился в выводе? Был ли перезапущен сервис nginx?

Элементы не добавились. Сервер не перезапустился



```
$ rm /etc/nginx/sites-enabled/web-8080.conf
$ ansible-playbook -i myhosts web-8080.yml

PLAY [manage httpd.conf] *****

TASK [Gathering Facts] *****
ok: [host01]

TASK [Copy web 8080 configuration file] *****
changed: [host01]

RUNNING HANDLER [reload_nginx] *****
changed: [host01]

PLAY RECAP *****
host01 : ok=3 changed=2 unreachable=0 failed=0 skipped=0 res
cued=0 ignored=0
```

Рис. 24 – Удаление конфигурации и исполнение playbook

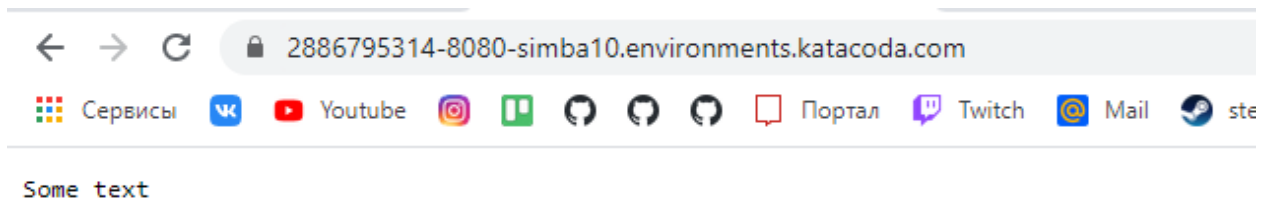


Рис. 25 – Запуск сервера

Что поменялось в выводе по сравнению с первым запуском? Работает ли сервер теперь?

Сервер заработал и вывел текст, который записан в конфигураторе

```
web-8080.conf
1  server {
2      listen 8080;
3
4      location / {
5          return 200 'Kirill Podolskiy';
6          add_header Content-Type 'text/plain; charset=utf-8';
7      }
8  }
```

Рис. 26 – Изменение файла конфигурации

Был ли сервер перезапущен?

Сервер перезапустился

Что такое handler в ansible? В каких случаях он выполняется?

handlers – это действия, которые будут выполняться после завершения задачи. Они запускаются на выполнения только 1 раз и после завершения всего сценария. Ключевое слово notify позволяет запускать handler’ы после успешного выполнения задачи.

```
! docker.yml X
! docker.yml
1 ---
2 - name: manage docker
3   hosts: group1
4   become: true
5
6   vars:
7     drupal_port: 8090
8
9   tasks:
10  - name: Install docker and some dependencies
11    apt:
12      name: python3-pip, docker-ce
13      state: present
14
15  - name: Start docker service
16    service:
17      name: docker
18      state: started
19
20  - name: Install docker python module
21    pip:
22      name: docker
23
24  - name: Create Drupal container
25    docker_container:
26      name: drupal
27      image: drupal
28      state: started
29      ports:
30        - "{{ drupal_port }}:80"
31      volumes:
32        - drupal_modules:/var/www/html/modules
33        - drupal_profiles:/var/www/html/profiles
34        - drupal_themes:/var/www/html/themes
35        - drupal_sites:/var/www/html/sites
```

Рис. 27 – Создание файла docker.yml

```

$ ansible-playbook -i myhosts docker.yml

PLAY [manage docker] *****

TASK [Gathering Facts] *****
ok: [host01]

TASK [Install docker and some dependencies] *****
ok: [host01]

TASK [Start docker service] *****
ok: [host01]

TASK [Install docker python module] *****
changed: [host01]

TASK [Create Drupal container] *****
changed: [host01]

PLAY RECAP *****
host01 : ok=5    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

```

Рис. 28 – Исполнение playbook

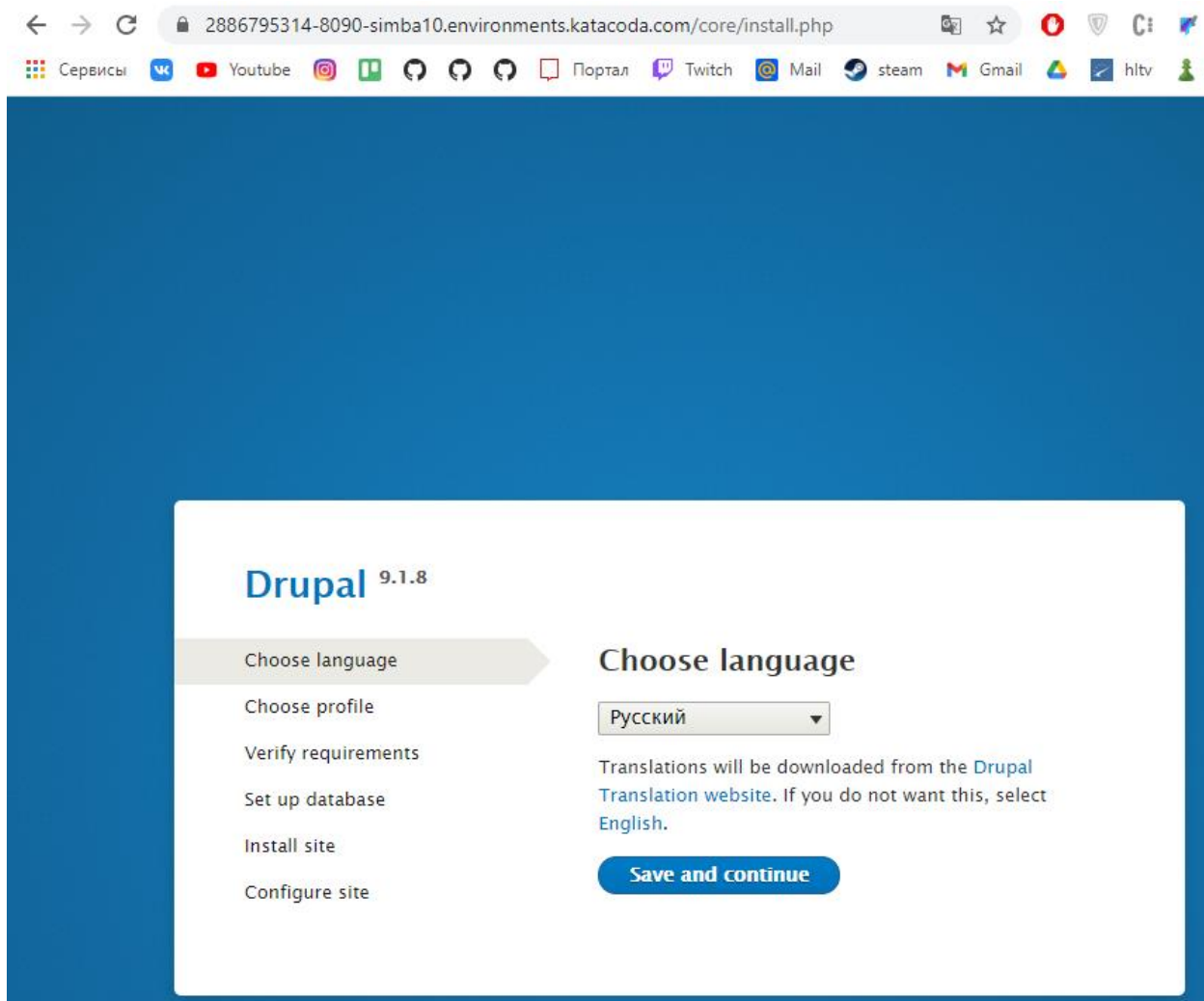


Рис. 29 – Открытый порт 8090

```
$ ansible-playbook -i myhosts docker.yml

PLAY [manage docker] *****

TASK [Gathering Facts] *****
ok: [host01]

TASK [Install docker and some dependencies] *****
ok: [host01]

TASK [Start docker service] *****
ok: [host01]

TASK [Install docker python module] *****
ok: [host01]

TASK [Create Drupal container] *****
ok: [host01]

PLAY RECAP *****
host01 : ok=5    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Рис. 30 - Повторное исполнение playbook

Был ли пересоздан контейнер?

Контейнер не пересоздался

```

docker.yml
1  ---
2  - name: manage docker
3    hosts: group1
4    become: true
5
6    vars:
7      drupal_port: 8012
8
9    tasks:
10   - name: Install docker and some dependencies
11     apt:
12       name: python3-pip, docker-ce
13       state: present
14
15   - name: Start docker service
16     service:
17       name: docker
18       state: started
19
20   - name: Install docker python module
21     pip:
22       name: docker
23
24   - name: Create Drupal container
25     docker_container:
26       name: drupal
27       image: drupal
28       state: started
29       ports:
30         - "{{ drupal_port }}:80"
31       volumes:
32         - drupal_modules:/var/www/html/modules
33         - drupal_profiles:/var/www/html/profiles
34         - drupal_themes:/var/www/html/themes
35         - drupal_sites:/var/www/html/sites

```

Рис. 31 – Изменение порта в playbook

```

$ ansible-playbook -i myhosts docker.yml

PLAY [manage docker] *****

TASK [Gathering Facts] *****
ok: [host01]

TASK [Install docker and some dependencies] *****
ok: [host01]

TASK [Start docker service] *****
ok: [host01]

TASK [Install docker python module] *****
ok: [host01]

TASK [Create Drupal container] *****
changed: [host01]

PLAY RECAP *****
host01 : ok=5 changed=1 unreachable=0 failed=0 skipped=0 res
cued=0 ignored=0

```

Рис. 32 – Повторное исполнение playbook

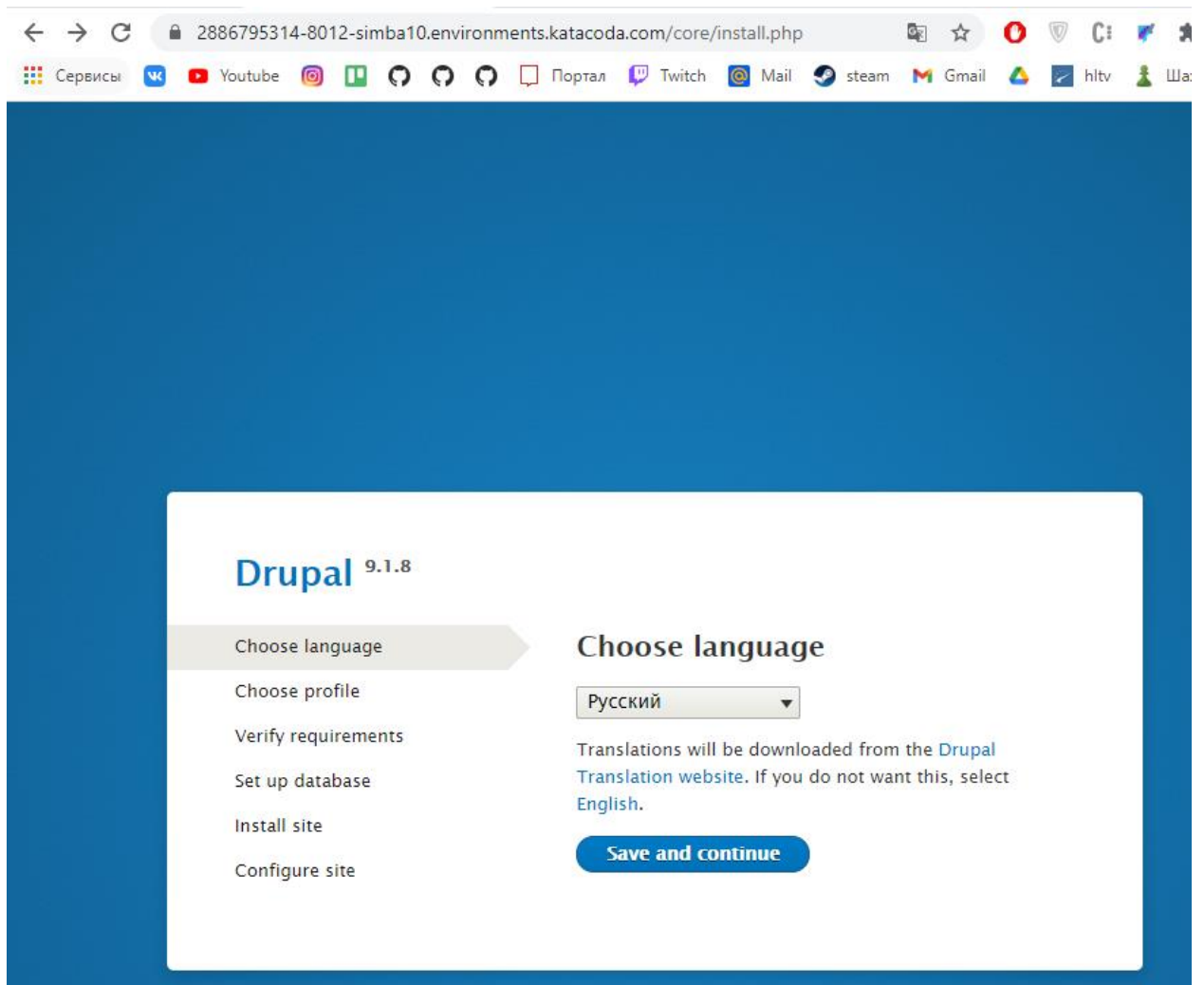


Рис. 33 – Сервер на новом порте

Был ли пересоздан контейнер теперь?

Контейнер пересоздался

```

! my-own-playbook.yml ×
! my-own-playbook.yml
1  ---
2  - name: Web server installed
3    hosts: group1
4    become: yes
5    vars:
6      my_name: Kirill Podolskiy
7    tasks:
8      - name: latest wget version installed
9        apt:
10         name: wget
11         state: latest
12         update_cache: true

```

Рис. 34 – Создание собственного playbook

```

$ ansible-playbook -i myhosts my-own-playbook.yml

PLAY [Web server installed] *****

TASK [Gathering Facts] *****
ok: [host01]

TASK [latest wget version installed] *****
ok: [host01]

PLAY RECAP *****
host01 : ok=2  changed=0  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0

```

Рис. 35 – Исполнение playbook


```

$ wget -V
GNU Wget 1.20.3 built on linux-gnu.

-cares +digest -gpgme +https +ipv6 +iri +large-file -metalink +nls
+ntlm +opie +psl +ssl/openssl

Wgetrc:
    /etc/wgetrc (system)
Locale:
    /usr/share/locale
Compile:
    gcc -DHAVE_CONFIG_H -DSYSTEM_WGETRC="/etc/wgetrc"
    -DLOCALEDIR="/usr/share/locale" -I. -I../src -I../lib
    -I../lib -Wdate-time -D_FORTIFY_SOURCE=2 -DHAVE_LIBSSL -DNDEBUG
    -g -O2 -fdebug-prefix-map=/build/wget-OYIf9/wget-1.20.3=.
    -fstack-protector-strong -Wformat -Werror=format-security
    -DNO_SSLv2 -D_FILE_OFFSET_BITS=64 -g -Wall
Link:
    gcc -DHAVE_LIBSSL -DNDEBUG -g -O2
    -fdebug-prefix-map=/build/wget-OYIf9/wget-1.20.3=.
    -fstack-protector-strong -Wformat -Werror=format-security
    -DNO_SSLv2 -D_FILE_OFFSET_BITS=64 -g -Wall -Wl,-Bsymbolic-functions
    -Wl,-z,relro -Wl,-z,now -lpcrc2-8 -luuid -lidn2 -lssl -lcrypto -lz
    -lpsl ftp-opie.o openssl.o http-ntlm.o ../lib/libgnu.a

Copyright (C) 2015 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later
<http://www.gnu.org/licenses/gpl.html>.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Originally written by Hrvoje Niksic <hniksic@xemacs.org>.
Please send bug reports and questions to <bug-wget@gnu.org>.
$ █

```

Рис. 36 – Демонстрация установленного пакета