

Лабораторная №2

Ansible + Caddy

Часть 1 - Установка и настройка Ansible

Устанавливаем пакетный менеджер pip для нашего python: `curl <https://bootstrap.pypa.io/get-pip.py>(http://get-pip.py) -o get-pip.py && python3 get-pip.py`

Устанавливаем, собственно, ansible: `python3 -m pip install ansible`

```
~ python3 -m pip install ansible
Collecting ansible
  Using cached ansible-8.7.0-py3-none-any.whl (48.4 MB)
Requirement already satisfied: ansible-core~2.15.7 in ./venv/lib/python3.9/site-packages (from ansible) (2.15.13)
Requirement already satisfied: cryptography in ./venv/lib/python3.9/site-packages (from ansible-core~2.15.7->ansible) (44.0.0)
Requirement already satisfied: resolvelib<1.1.0,>=0.5.3 in ./venv/lib/python3.9/site-packages (from ansible-core~2.15.7->ansible) (1.0.1)
Requirement already satisfied: jinja2>=3.0.0 in ./venv/lib/python3.9/site-packages (from ansible-core~2.15.7->ansible) (3.1.4)
Requirement already satisfied: packaging in ./venv/lib/python3.9/site-packages (from ansible-core~2.15.7->ansible) (24.2)
Requirement already satisfied: importlib-resources<5.1,>=5.0 in ./venv/lib/python3.9/site-packages (from ansible-core~2.15.7->ansible) (5.0.7)
Requirement already satisfied: PyYAML>=5.1 in ./venv/lib/python3.9/site-packages (from ansible-core~2.15.7->ansible) (6.0.2)
Requirement already satisfied: MarkupSafe>=2.0 in ./venv/lib/python3.9/site-packages (from jinja2>=3.0.0->ansible-core~2.15.7->ansible) (3.0.2)
Requirement already satisfied: cffi>=1.12 in ./venv/lib/python3.9/site-packages (from cryptography->ansible-core~2.15.7->ansible) (1.17.1)
Requirement already satisfied: pycparser in ./venv/lib/python3.9/site-packages (from cffi>=1.12->cryptography->ansible-core~2.15.7->ansible) (2.22)
Installing collected packages: ansible
Successfully installed ansible-8.7.0
```

Выбираем или создаем директорию, где будем работать. Создаем базовый конфиг файл, затем папку inventory и в ней файл с хостами (тренироваться будем на localhost)

```
[defaults]
host_key_checking = false
inventory          = inventory/hosts
```

```
[my_servers]
local_server      ansible_host=localhost
```

Проверяем, что сервер с Ansible подключился к “клиенту” (в нашем случае это одна и та же машина, localhost): `ansible my_servers -m ping -c local` и/или `ansible my_servers -m setup -c local`

```

~ ansible my_servers -m ping -c local
[WARNING]: Platform darwin on host local_server is using the discovered Python interpreter at /opt/homebrew/bin/python3.11, but future
installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-
core/2.15/reference_appendices/interpreter_discovery.html for more information.
local_server | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/opt/homebrew/bin/python3.11"
  },
  "changed": false,
  "ping": "pong"
}
~ ansible my_servers -m setup -c local
[WARNING]: Platform darwin on host local_server is using the discovered Python interpreter at /opt/homebrew/bin/python3.11, but future
installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-
core/2.15/reference_appendices/interpreter_discovery.html for more information.
local_server | SUCCESS => {
  "ansible_facts": {
    "ansible_all_ipv4_addresses": [

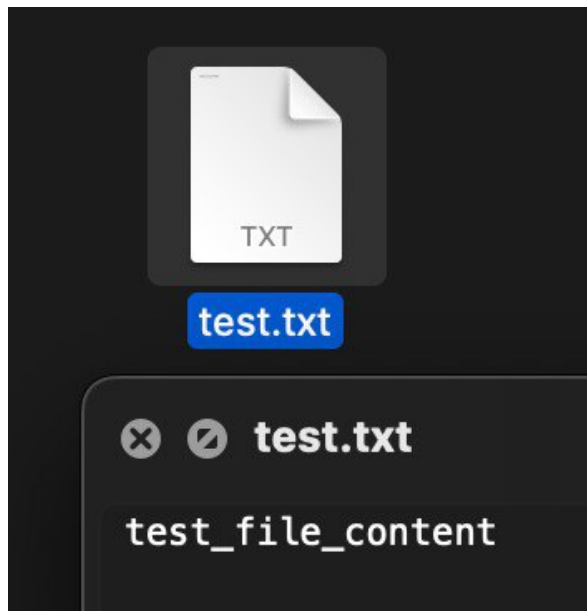
```

```

~ ansible my_servers -c local -m shell -a 'echo test_file_content > $HOME/test.txt'
[WARNING]: Platform darwin on host local_server is using the discovered Python interpreter at /opt/homebrew/bin/python3.11, but future
installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-
core/2.15/reference_appendices/interpreter_discovery.html for more information.
local_server | CHANGED | rc=0 >>

```

Создаем текстовый файл с произвольным содержимым, через модуль shell:
`ansible my_servers -c local -m shell -a 'echo test_file_content > $HOME/test.txt'`



Проверяем, что по нужному пути создан нужный файл с нужным именем и содержимым

```

---
- name: Manage test file
  hosts: my_servers
  connection: local
  tasks:
    - name: Create a text file with initial content
      shell: echo test_file_content > $HOME/test.txt

    - name: Change the content of the file
      shell: echo new_file_content > $HOME/test.txt

    - name: Remove the text file
      file:
        path: $HOME/test.txt
        state: absent

```

```

~ ansible-playbook playbook.yml

PLAY [Manage test file] *****

TASK [Gathering Facts] *****
[WARNING]: Platform darwin on host local_server is using the discovered Python interpreter at /opt/homebrew/bin/python3.11, but future
installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-
core/2.15/reference_appendices/interpreter_discovery.html for more information.
ok: [local_server]

TASK [Create a text file with initial content] *****
changed: [local_server]

TASK [Change the content of the file] *****
changed: [local_server]

TASK [Remove the text file] *****
changed: [local_server]

PLAY RECAP *****
local_server      : ok=4   changed=3   unreachable=0   failed=0   skipped=0   rescued=0   ignored=0

```

Удаляем файл через модуль file: `ansible my_servers -c local -m file -a 'path=$HOME/test.txt state=absent'`

```

~ ansible my_servers -c local -m file -a 'path=$HOME/test.txt state=absent'
[WARNING]: Platform darwin on host local_server is using the discovered Python interpreter at /opt/homebrew/bin/python3.11, but future
installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-
core/2.15/reference_appendices/interpreter_discovery.html for more information.
local_server | CHANGED => {
  "ansible_facts": {
    "discovered_interpreter_python": "/opt/homebrew/bin/python3.11"
  },
  "changed": true,
}

```

Часть 2 - Установка Caddy

В части 1 были рассмотрены базовые ad-hoc команды для Ansible, пора переходить к более сложным конструкциям - [Ansible Playbooks](#). Устанавливать будем вебсервер [Caddy](#). Для начала создадим в рабочей директории папку **roles** и в ней инициализируем исходное конфигурационное “дерево”: **ansible-galaxy init caddy_deploy**

```
Processing triggers for man-db (2.10.2-1) ...
root@hakuna-matata:~/ansible_project/roles# tree
.
├── caddy_deploy
│   ├── README.md
│   ├── defaults
│   │   └── main.yml
│   ├── files
│   ├── handlers
│   │   └── main.yml
│   ├── meta
│   │   └── main.yml
│   ├── tasks
│   │   └── main.yml
│   ├── templates
│   ├── tests
│   │   ├── inventory
│   │   └── test.yml
│   └── vars
│       └── main.yml
└── 9 directories, 8 files
root@hakuna-matata:~/ansible_project/roles#
```

Наполняем файл `roles/caddy_deploy/tasks/main.yml`. Здесь мы описываем непосредственно шаги, которые будут выполняться в нашем плейбуке (по сути, несколько команд `ansible -m *****` подряд)

И в рабочей директории создаём файл конфигурации самого плейбука, где указываем нужные нам хосты и роли соответственно

После запускаем плейбук: `ansible-playbook caddy_deploy.yml` и проверяем, успешно ли все шаги выполнены

```
root@hakuna-matata:~/ansible_project# ansible-playbook caddy_deploy.yml

PLAY [Install and configure Caddy webserver] *****
TASK [Gathering Facts] *****[WARNING
]: Platform linux on host local_server is using the discovered Python interpreter at
/usr/bin/python3.10, but future installation of another Python interpreter could change the meaning
of that path. See https://docs.ansible.com/ansible-
core/2.17/reference_appendices/interpreter_discovery.html for more information.
ok: [local_server]

TASK [caddy_deploy : Install prerequisites] *****ok: [loc
al_server]

TASK [caddy_deploy : Add key for Caddy repo] *****ok: [loc
al_server]

TASK [caddy_deploy : add Caddy repo] *****ok: [loc
al_server]

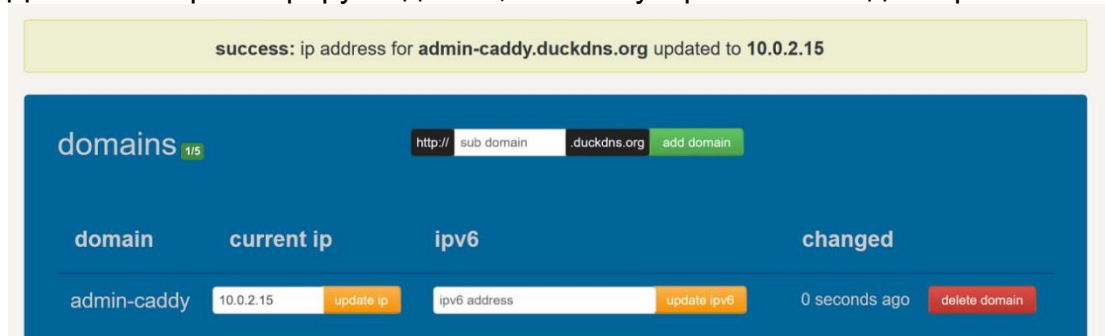
TASK [caddy_deploy : add Caddy src repo] *****ok: [loc
al_server]

TASK [caddy_deploy : Install Caddy webserver] *****changed:
[local_server]

PLAY RECAP *****local_se
rver      : ok=6   changed=1   unreachable=0   failed=0   skipped=0   rescued=0   ignored=0
```

Часть 3 - Домен и настройка Caddyfile

Для начала регистрируем домен, используя ip сетевого адаптера



Создадим простой шаблон:

```
Caddyfile.j2 X
roles > caddy_deploy > templates > Caddyfile.j2
1  {
2      auto_https off
3  }
4
5  :80 {
6      root * /usr/share/caddy
7      file_server
8
9      log {
10         output file {{ log.file }}
11         format json
12         level {{ log.level }}
13     }
14
15 }
```


И определяем переменные

```
Caddyfile.j2  ! main.yml X
roles > caddy_deploy > vars > ! main.yml
1  ---
2  # vars file for caddy_deploy
3
4  domain_name: admin-caddy.duckdns.org
5
6  log: # Можно поиграться со значениями
7      file: /var/log/caddy_access.log
8      level: "INFO"
```

После добавим два новых таска, что на основе вышесозданных файлов создадут конфигурацию caddy и перезапустят сервис

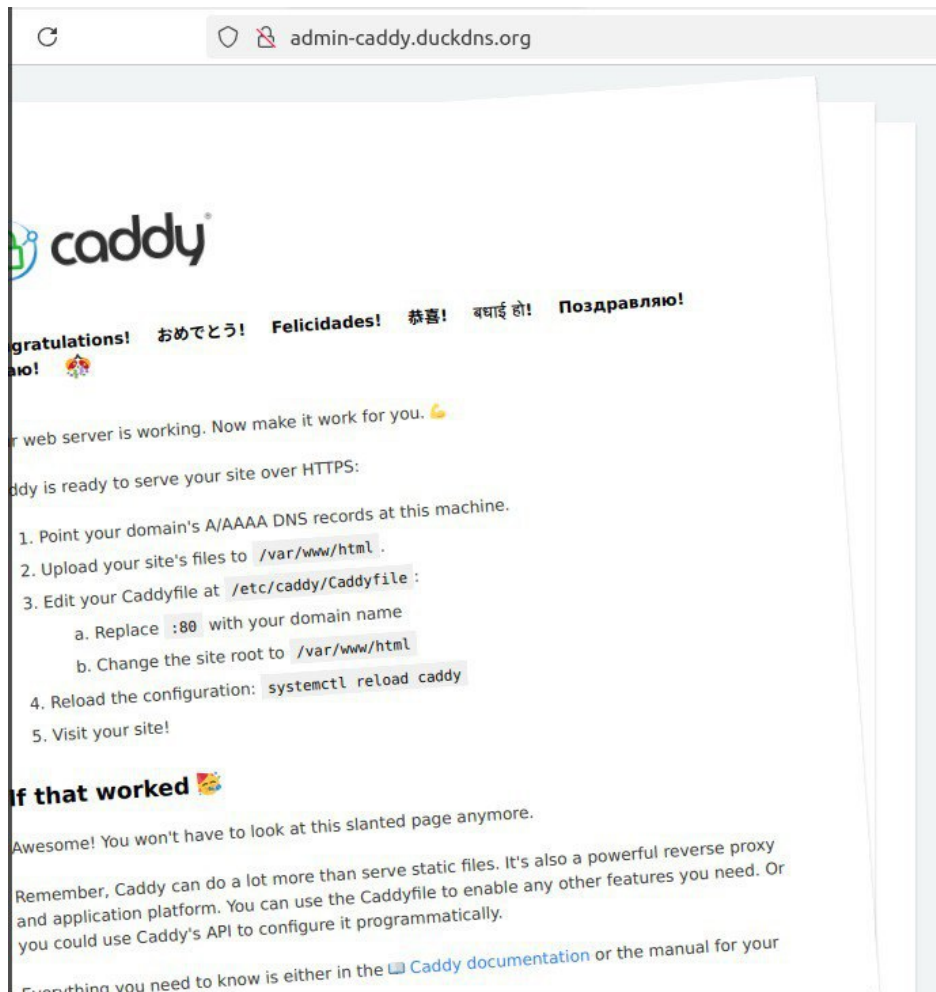
```
34
35  - name: Create config file
36    template:
37      src: templates/Caddyfile.j2
38      dest: /etc/caddy/Caddyfile
39
40  - name: Reload with new config
41    service:
42      name: caddy
43      state: reloaded
```

Проверим работоспособность домена:

```
TASK [caddy_deploy : Create config file] *****
ok: [local_server]

TASK [caddy_deploy : Reload with new config] *****
changed: [local_server]

PLAY RECAP *****
local_server      : ok=8    changed=1    unreachable=0    failed=0    skipped=0    rescue=0
d=0              ignored=0
```



Задание 1

```
root@Inko-Verikuse:/mnt/c/users/svgarik/Documents/Admin2# ansible-playbook caddy_deploy.yml

PLAY [Install and configure Caddy webserver] *****

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

TASK [caddy_deploy : Create a text file with initial content] *****
changed: [local_server]

TASK [caddy_deploy : Read file content] *****
changed: [local_server]

TASK [caddy_deploy : Show file content] *****
ok: [local_server] => {
  "msg": "test_file_content"
}

TASK [caddy_deploy : Change the content of the file] *****
changed: [local_server]

TASK [caddy_deploy : Read file content] *****
changed: [local_server]

TASK [caddy_deploy : Show file content] *****
ok: [local_server] => {
  "msg": "new_file_content"
}

TASK [caddy_deploy : Remove the text file] *****
changed: [local_server]

PLAY RECAP *****
local_server      : ok=8   changed=5   unreachable=0   failed=0   skipped=0   rescued=0   ignored=0
```

```

- name: Create a text file with initial content
  shell: echo test_file_content > $HOME/test.txt

- name: Read file content
  shell: cat $HOME/test.txt
  register: content
  tags: read_file

- name: Show file content
  debug:
    msg: "{{ content.stdout }}"
  tags: read_file

- name: Change the content of the file
  shell: echo new_file_content > $HOME/test.txt

- name: Read file content
  shell: cat $HOME/test.txt
  register: content

- name: Show file content
  debug:
    msg: "{{ content.stdout }}"

- name: Remove the text file
  file:
    path: $HOME/test.txt
    state: absent

```

Задание 2

Был изменён файл конфигурации таким образом, чтобы добавить заголовок:

```

{
    auto_https off
}

:80 {
    root * /usr/share/caddy
    file_server

    log {
        output file {{ log.file }}
        format json
        level {{ log.level }}
    }

    header {
        X-Custom-Header "Dicsworld"
    }
}

```

А также добавлен таск для смены главной страницы


```
- name: Create custom index.html
copy:
  content: |
    <html>
    <head>
      <title>robot</title>
    </head>
    <body>
      <h1>Hello, Samuel Vimes</h1>
    </body>
    </html>
  dest: /usr/share/caddy/index.html
```



Ответы на вопросы: