

2

ЛР 2. Ansible + Caddy

▼ Задача

На целевом сервере установить Ansible и вебсервер Caddy



Работу можно выполнять локально, на выданной ВМ или одновременно на обоих (см. пункт 2б раздела Задание)

▼ Гайд

▼ Часть 1. Установка и настройка Ansible

- Устанавливаем пакетный менеджер pip для нашего python: `curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py && python3 get-pip.py`
- Устанавливаем, собственно, ansible: `python3 -m pip install ansible`
- Выбираем или создаем директорию, где будем работать. Создаем базовый конфиг файл, затем папку inventory и в ней файл с хостами (тренироваться будем на localhost)

▼ ansible.cfg

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

▼ 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`

```
root@tutor:~/ansible# ansible my_servers -m ping -c local
[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.
local_server | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.10"
    },
    "changed": false,
    "ping": "pong"
}
root@tutor:~/ansible# ansible my_servers -m setup -c local
[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.
local_server | SUCCESS => {
    "ansible_facts": {
        "ansible_all_ipv4_addresses": [
            "192.168.0.145"
        ],
        "ansible_all_ipv6_addresses": [
            "fe80::f816:3eff:fe4e:ce84"
        ],
        "ansible_apparmor": {
            "status": "enabled"
        },
        "ansible_architecture": "x86_64",
        "ansible_bios_date": "04/01/2014",
        "ansible_bios_vendor": "SeaBIOS",
        "ansible_bios_version": "1.15.0-1",
        "ansible_board_asset_tag": "NA",
        "ansible_board_name": "NA",
        "ansible_board_serial": "NA",
        "ansible_board_vendor": "NA",
        "ansible_board_version": "NA",
        "ansible_chassis_asset_tag": "NA",
        "ansible_chassis_serial": "NA",
        "ansible_chassis_vendor": "QEMU",
        "ansible_chassis_version": "pc-i440fx-6.2",
    }
}
```



Ключ `-c local` нужен только для локальных соединений по локалхосту, без него соединение по умолчанию будет пытаться установиться через ssh

 Если бы в конфиге в разделе `my_servers` было бы перечислено несколько серверов (построчно), команды выполнялись бы на всех одновременно. А если вместо `my_servers` было бы `all`, то вообще на всех хостах из файла

5. Пробуем выполнить команду посложнее на нашем клиенте

- Создаем текстовый файл с производным содержимым, через модуль shell: `ansible my_servers -c local -m shell -a 'echo test_file_content > $HOME/test.txt'`
- Проверяем, что по нужному пути создался нужный файл с нужным именем и содержимым
- Удаляем файл через модуль file: `ansible my_servers -c local -m file -a 'path=$HOME/test.txt state=absent'`

```
root@tutor:~/ansible# ansible my_servers -c local -m shell -a 'echo test_file_content > $HOME/test.txt'
[WARNING]: Platform linux on host local_server is using the discovered Python interpreter at /usr/bin/python3.10, but future installation of another Python
interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more
information.
local_server | CHANGED | rc=0 >>
root@tutor:~/ansible# ls -la ..
total 2260
drwx----- 7 root root 4096 Oct  6 12:53 .
drwxr-xr-x 20 root root 4096 Oct  6 11:31 ..
drwxr-xr-x 4 root root 4096 Oct  6 12:33 .ansible
-rw----- 1 root root 7 Oct  6 11:41 .bash_history
-rw-r--r-- 1 root root 3106 Oct 15 2021 .bashrc
drwx----- 3 root root 4096 Oct  6 12:17 .cache
drwxr-xr-x 5 root root 4096 Oct  6 12:25 .local
-rw-r--r-- 1 root root 161 Jul  9 2019 .profile
-rw----- 1 root root 0 Oct  6 12:10 .python_history
drwx----- 2 root root 4096 Oct  6 12:33 .ssh
drwxr-xr-x 3 root root 4096 Oct  6 12:32 ansible
-rw-r--r-- 1 root root 2266947 Oct  6 12:17 get-pip.py
-rw-r--r-- 1 root root 18 Oct  6 12:53 test.txt
root@tutor:~/ansible# cat ./test.txt
test_file_content
root@tutor:~/ansible# ansible my_servers -c local -m file -a 'path=$HOME/test.txt state=absent'
[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.
local_server | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.10"
    },
    "changed": true,
    "path": "/root/test.txt",
    "state": "absent"
}
root@tutor:~/ansible# cat ./test.txt
cat: ./test.txt: No such file or directory
root@tutor:~/ansible#
```

▼ 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`

```
root@tutor:~/ansible# mkdir roles && cd roles
root@tutor:~/ansible/roles# ansible-galaxy init caddy_deploy
- Role caddy_deploy was created successfully
root@tutor:~/ansible/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@tutor:~/ansible/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)

```
---
# tasks file for caddy_deploy

- name: Install prerequisites
  apt:
    pkg:
      - debian-keyring
      - debian-archive-keyring
      - apt-transport-https
      - curl

- name: Add key for Caddy repo
  apt_key:
    url: https://dl.cloudsmith.io/public/caddy/stable/gpg.key
    state: present
    keyring: /usr/share/keyrings/caddy-stable-archive-keyring.gpg
```

```

- name: add Caddy repo
apt_repository:
  repo: "deb [signed-by=/usr/share/keyrings/caddy-stable-archive-keyring.gpg] https://dl.cloudsmith.io/public/caddy/stable/deb/debian any-version main"
  state: present
  filename: caddy-stable

- name: add Caddy src repo
apt_repository:
  repo: "deb-src [signed-by=/usr/share/keyrings/caddy-stable-archive-keyring.gpg] https://dl.cloudsmith.io/public/caddy/stable/deb/debian any-version main"
  state: present
  filename: caddy-stable

- name: Install Caddy webserver
apt:
  name: caddy
  update_cache: yes
  state: present

```

3. Наконец, в рабочей директории создаем собственно файл конфигурации самого плейбука, где указываем нужные нам хосты и роли

```

▼ caddy_deploy.yml

---
- name: Install and configure Caddy webserver # Любое описание
hosts: my_servers # хосты из файла inventory/hosts, где будем выполнять наш плейбук
connection: local # аналог -c local, но для плейбуков

roles:
  - caddy_deploy # собственно, роль для выполнения

```

 Файл конфигурации нужно положить и запускать из рабочей директории, на одном уровне с `ansible.cfg`, `inventory` и `roles`!

4. Запускаем плейбук: `ansible-playbook caddy_deploy.yml` и проверяем, успешно ли все шаги выполнились

```

root@tutor:~/ansible# 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: [local_server]

TASK [caddy_deploy : Add key for Caddy repo] ****
changed: [local_server]

TASK [caddy_deploy : add Caddy repo] ****
changed: [local_server]

TASK [caddy_deploy : add Caddy src repo] ****
changed: [local_server]

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

PLAY RECAP ****
local_server      : ok=6    changed=4    unreachable=0   failed=0    skipped=0   rescued=0   ignored=0

root@tutor:~/ansible# service caddy status
● caddy.service - Caddy
  Loaded: loaded (/lib/systemd/system/caddy.service; enabled; vendor preset: enabled)
  Active: active (running) since Sun 2024-10-06 20:54:41 UTC; 16s ago
    Docs: https://caddyserver.com/docs/
    Main PID: 14477 (caddy)
       Tasks: 7 (limit: 2268)
      Memory: 10.5M
         CPU: 82ms
        CGroup: /system.slice/caddy.service
                  └─14477 /usr/bin/caddy run --environ --config /etc/caddy/Caddyfile

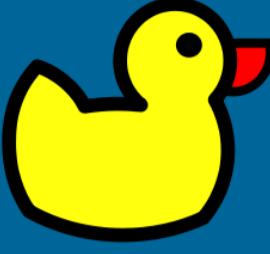
```

▼ Часть 3. Домен и настройка Caddyfile

1. Регистрируем себе бесплатный домен на выданный ранее ip-адрес, например на сервисе duckdns.org

▼ Пример зарегистрированного домена lidd.duckdns.org для VM с адресом 31.129.35.15

Duck DNS spec about why install faqs logout logged in with llidd@github |||



Duck DNS

account llidd@github
 type free
 token 079c28bb-1936-487f-961a-365a0d521070
 token generated 2 minutes ago
 created date 6 Oct 2024, 12:03:33

domain	current ip	ipv6	changed
llidd	31.129.35.15 <button>update ip</button>	<input type="text"/> ipv6 address <button>update ipv6</button>	1 minute ago <button>delete domain</button>

This site is protected by reCAPTCHA and the Google [Privacy Policy](#) and [Terms of Service](#) apply.

[Donate](#) [Bitcoin](#) 16gHnv3NTjpF5ZavMi9QYBFxUkNchdicUS [Patreon](#) [Follow](#)

[Terms of Use](#) [Privacy Statement](#)

2. Попробуем использовать доп. возможности плейбуков - создадим шаблон (Jinja2) и переменные (в формате {{ var }})

▼ roles/caddy_deploy/templates/Caddyfile.j2

```
{{ domain_name }} {
    root * /usr/share/caddy
    file_server

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

▼ roles/caddy_deploy/vars/main.yml

```
---
# vars file for caddy_deploy

domain_name: прописываем.свой.домен

log: # Можно поиграться со значениями
  file: /var/log/caddy_access.log
  level: "INFO"
```

3. Добавляем в наш плейбук (в tasks) новые шаги, отвечающие за создание конфигурационного файла из шаблона и последующую перезагрузку сервиса:

```
- ...
- ...
- name: Create config file
  template:
    src: templates/Caddyfile.j2 # Откуда берем
    dest: /etc/caddy/Caddyfile # Куда кладем

- name: Reload with new config
  service:
    name: caddy
    state: reloaded
```

4. Снова запускаем плейбук, после чего вводим в браузере имя своего домена и убеждаемся, что тестовая страничка Caddy автоматически поднялась на подписанным сертификате с https

▼ Пример для все того же llidd.duckdns.org:

```

root@tutor:~/ansible# 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: [local_server]

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

TASK [caddy_deploy : add Caddy repo] ****
ok: [local_server]

TASK [caddy_deploy : add Caddy src repo] ****
changed: [local_server]

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

TASK [caddy_deploy : Create config file] ****
changed: [local_server]

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

PLAY RECAP ****
local_server          : ok=8    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

root@tutor:~/ansible# cat /etc/caddy/Caddyfile
lidd.duckdns.org {
    root * /usr/share/caddy
    file_server

    log {
        output file /var/log/caddy_access.log
        format json
        level INFO
    }
}
root@tutor:~/ansible#

```

The screenshot shows a browser window with two tabs. The active tab displays the Caddy web server's configuration page, which includes a logo, a message in multiple languages (Congratulations! おめでとう! Felicidades! 恭喜! बधाइ हो! Поздравляю! Вітаю!), and instructions for setting up HTTPS. The second tab shows a certificate viewer for the domain lidd.duckdns.org, displaying details such as the subject (lidd.duckdns.org), issuer (Let's Encrypt), and certificate expiration date (Sunday, January 4, 2025).

▼ Задание

1. Переписать пример с созданием и удалением файла из шага 5 Части 1 с ad-hoc команд на плейбук формат, а так же добавить четвертый шаг - перед удалением поменять содержимое файла на любое другое.

Весь плейбук разрешается разместить в одном yml файле, без разделения на tasks/vars/etc как в Части 2

2. а) "Расширить" конфиг вебсервера Caddy любым функционалом по желанию: например, добавить проксирование, или какие-нибудь заголовки ([header](#)). Вместо дефолтной страницы Caddy подставить свою, хотя бы index.html с Hello world внутри. Добавить это в качестве дополнительного шага в tasks



Возможно, потребуется правка исходного шаблона [caddyfile.j2](#), если новая index страница будет лежать по иному пути

ЛИБО

- б) сделать всю работу на удаленном сервере, а не на одном и том же localhost

▼ Отчетность

Отчет с ходом работы (аналогичный тексту текущей лабораторной) и скриншотами. Можно в документе (Word, PDF, etc) или в git репозитории