Лабораторная работа №2

1) Проверяем, что сервер с Ansible подключился к "клиенту"

```
(.venv) semyondanshin@MacBook-Pro-Semen LR2 % ansible my_servers -m ping -c local
[WARNING]: Platform darwin on host local_server is using the discovered Python interpreter at /Users/semyondanshin
core/2.15/reference_appendices/interpreter_discovery.html for more information.
local_server | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/Users/semyondanshin/Programming/startofworms/LR2/.venv/bin/python3.9"
    },
    "changed": false,
    "ping": "pong"
}
```

2) Создаем текстовый файл с производным содержимым, через модуль shell и проверяем содержимое:

```
(.venv) semyondanshin@MacBook-Pro-Semen LR2 % ansible my_servers -c local -m shell -a 'echo skibidi > $HOME/Programming/startofworms/LR2/text.txt'
[WARNING]: Platform darwin on host local_server is using the discovered Python interpreter at /Users/semyondanshin/Programming/startofworms/LR2/.ver
core/2.15/reference_appendices/interpreter_discovery.html for more information.
local_server | CHANGED | rc=0 >>

(.venv) semyondanshin@MacBook-Pro-Semen LR2 % cat ./text.txt
skibidi
```

3) Удаляем файл через модуль file и проверяем результат:

```
(.venv) semyondanshin@MacBook-Pro-Semen LR2 % ansible my_servers -c local -m file -a 'path=$HOME/Programming/startofworms/LR2/text.txt state=absent'
[WARNING]: Platform darwin on host local_server is using the discovered Python interpreter at /Users/semyondanshin/Programming/startofworms/LR2/.venv/
core/2.15/reference_appendices/interpreter_discovery.html for more information.
local_server | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/Users/semyondanshin/Programming/startofworms/LR2/.venv/bin/python3.9"
    },
    "changed": true,
    "path": "/Users/semyondanshin/Programming/startofworms/LR2/text.txt",
    "state": "absent"
}
(.venv) semyondanshin@MacBook-Pro-Semen LR2 % cat ./text.txt
cat: ./text.txt: No such file or directory
```

4) Добавляем роль *caddy_deploy* и прописываем файлы конфигурации
5) Прописываем задачи для установки caddy на сервере и запускаем:

```
(.venv) semyondanshin@MacBook-Pro-Semen LR2 % ANSIBLE_COMMAND_WARNINGS=False ansible-playbook caddy_deploy.yaml --user root

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

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

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

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

6) Расширяем файлы конфигурации, для автоматического выпуска ssl
сертификата, конфигурации сервера и замены стандартной страницы Caddy

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

- name: Replace default web page with custom index.html
  copy:
    src: templates/index.html
    dest: /usr/share/caddy/index.html
    owner: www-data
    group: www-data
    mode: '0644'

- name: Ensure web root permissions
  file:
    path: /usr/share/caddy
    state: directory
    owner: www-data
    group: www-data
    mode: '0755'

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

7) Запускаем playbook на удаленном сервере:

```
(.venv) semyondanshin@MacBook-Pro-Semen LR2 % ANSIBLE_COMMAND_WARNINGS=False ansible-playbook caddy_deploy.yaml --user root

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

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

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

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

TASK [caddy_deploy : Replace default web page with custom index.html] *************************************
changed: [local_server]

TASK [caddy_deploy : Ensure web root permissions] *********************************************************
ok: [local_server]

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

PLAY RECAP ************************************************************************************************
local_server               : ok=10   changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

8) Проверяем результат

Дополнительно:
1. Оформить операции с файлом в отдельный плейбук:

```yaml
- name: Manage test file lifecycle
  hosts: localhost
  tasks:
    - name: Create a test file with initial content
      copy:
        content: "test_file_content"
        dest: "{{ ansible_env.HOME }}/test.txt"

    - name: Verify the file was created (optional)
      stat:
        path: "{{ ansible_env.HOME }}/test.txt"
      register: test_file_status

    - name: Print verification message
      debug:
        msg: "File exists and contains the expected content."
      when: test_file_status.stat.exists

    - name: Update the test file content
      copy:
        content: "updated_test_file_content"
        dest: "{{ ansible_env.HOME }}/test.txt"

    - name: Delete the test file
      file:
        path: "{{ ansible_env.HOME }}/test.txt"
        state: absent

    - name: Verify the file was deleted (optional)
      stat:
        path: "{{ ansible_env.HOME }}/test.txt"
      register: test_file_status_deleted

    - name: Print deletion verification message
      debug:
        msg: "File successfully deleted."
      when: not test_file_status_deleted.stat.exists
```