

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ**

ФАКУЛЬТЕТ ИНФОКОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ

Отчет по лабораторной работе 3

Тема: «Ansible + Caddy»

Выполнili:

Кутуков.Д, Кижваткин Н,

Фоченков С, Бочарников М, Субботин

А

Проверил:

Самохин Н.Ю.

Санкт-Петербург

2025 г.

СОДЕРЖАНИЕ

Введение	3
Часть 1. Установка и настройка Ansible.....	4
1 Установка Ansible и подготовка окружения	4
2 Проверка подключения к управляемому хосту.....	5
3 Использование ad-hoc команд.....	6
Часть 2. Установка Caddy	9
1 Создание роли Ansible.....	9
2 Реализация задач роли.....	9
3 Создание основного playbook'a	9
Часть 3. Домен и настройка Caddyfile	10
1 Регистрация доменного имени.....	10
2 Использование шаблонов и переменных.....	11
3 Автоматическое создание конфигурации и перезагрузка сервиса.....	11
Задания.....	13
1 Переписывание ad-hoc команд в формат playbook	13
2 Расширение конфигурации веб-сервера Caddy	14
Вывод	17

Введение

Цель работы освоить базовые и расширенные возможности системы управления конфигурациями Ansible, а также автоматизировать установку и настройку веб-сервера Caddy с использованием playbook'ов, ролей и шаблонов Jinja2.

Задание

1. Установить и настроить Ansible.
2. Выполнить базовые ad-hoc команды.
3. Создать Ansible Playbook для установки веб-сервера Caddy.
4. Настроить Caddy с использованием шаблона и переменных.
5. Переписать ad-hoc команды работы с файлами в формат playbook.
6. Расширить конфигурацию Caddy (добавить собственную страницу и заголовки).

Описание используемого окружения

Лабораторная работа выполнялась на локальной машине (localhost), которая одновременно выступала в роли управляющего узла Ansible и целевого сервера.

В процессе выполнения использовались следующие компоненты:

- Операционная система семейства MacOS (до части 2, далее использовался сервер по SSH)
- Python версии 3.13.9
- Ansible
- Веб-сервер Caddy

Часть 1. Установка и настройка Ansible

1 Установка Ansible и подготовка окружения

На первом этапе лабораторной работы была выполнена установка менеджера пакетов `pip` для Python, необходимого для дальнейшей установки Ansible. После установки `pip` был установлен сам Ansible, который является основным инструментом для управления конфигурациями и автоматизации. После завершения установки была выполнена проверка корректности установки Ansible.

Далее была создана отдельная рабочая директория для выполнения лабораторной работы. В данной директории были размещены основные конфигурационные файлы Ansible.

Был создан файл `ansible.cfg`, в котором были заданы базовые параметры работы Ansible. В частности, была отключена проверка SSH-ключей хостов, а также указан путь к `inventory`-файлу. Это позволило упростить работу с локальным хостом и избежать лишних предупреждений при запуске команд.

Также была создана директория `inventory`, внутри которой был размещён файл `hosts`. В данном файле была описана группа хостов `my_servers`, в которую был добавлен локальный сервер (`localhost`). Таким образом, Ansible был настроен для работы с локальной системой в качестве управляемого узла.

```
● ○ ● ⌂ .../Documents/temp/lab3ansible
  Downloading resolvelib-1.2.1-py3-none-any.whl.metadata (3.7 kB)
Collecting MarkupSafe≥2.0 (from jinja2≥3.1.0→ansible-core~2.20.1→ansible)
  Downloading markupsafe-3.0.3-cp313-cp313-macosx_11_0_arm64.whl.metadata (2.7 kB)
Collecting cffi≥2.0.0 (from cryptography→ansible-core~2.20.1→ansible)
  Using cached cffi-2.0.0-cp313-cp313-macosx_11_0_arm64.whl.metadata (2.6 kB)
Collecting pycparser (from cffi≥2.0.0→cryptography→ansible-core~2.20.1→ansible)
  Using cached pycparser-2.23-py3-none-any.whl.metadata (993 bytes)
  Downloading ansible-13.1.0-py3-none-any.whl (54.1 MB)
  ━━━━━━━━━━━━━━━━ 54.1/54.1 MB 16.1 MB/s 0:00:03
  Downloading ansible_core-2.20.1-py3-none-any.whl (2.4 MB)
  ━━━━━━━━━━━━━━ 2.4/2.4 MB 10.3 MB/s 0:00:00
  Downloading resolvelib-1.2.1-py3-none-any.whl (18 kB)
  Using cached jinja2-3.1.6-py3-none-any.whl (134 kB)
  Downloading markupsafe-3.0.3-cp313-cp313-macosx_11_0_arm64.whl (12 kB)
  Downloading pyyaml-6.0.3-cp313-cp313-macosx_11_0_arm64.whl (173 kB)
  Downloading cryptography-46.0.3-cp311-abi3-macosx_10_9_universal2.whl (7.2 MB)
  ━━━━━━━━━━━━ 7.2/7.2 MB 10.5 MB/s 0:00:00
  Using cached cffi-2.0.0-cp313-cp313-macosx_11_0_arm64.whl (181 kB)
  Using cached pycparser-2.23-py3-none-any.whl (118 kB)
  Installing collected packages: resolvelib, PyYAML, pycparser, MarkupSafe, jinja2, cffi, cryptography, ansible-core, ansible
  Successfully installed MarkupSafe-3.0.3 PyYAML-6.0.3 ansible-13.1.0 ansible-core-2.20.1 cffi-2.0.0
  cryptography-46.0.3 jinja2-3.1.6 pycparser-2.23 resolvelib-1.2.1

[notice] A new release of pip is available: 25.2 → 25.3
[notice] To update, run: pip install --upgrade pip
> cd Documents
> cd temp
> mkdir lab3ansible
> cd lab3ansible
> nano ansible.cfg
> mkdir inventory
> nano inventory/hosts
~/Documents/temp/lab3ansible 
```

14s

2 Проверка подключения к управляемому хосту

После настройки inventory-файла была выполнена проверка связи между Ansible и целевым хостом. Для этого использовались стандартные ad-hoc команды Ansible с модулями ping и setup.

Так как работа выполнялась на локальной машине, было использовано подключение типа local. В результате выполнения команд было подтверждено, что Ansible успешно взаимодействует с целевым хостом и может выполнять на нём задачи.

```
● ○ ● ⚡ .../Documents/temp/lab3ansible
` ansible my_servers -m ping -c local
[WARNING]: Host 'local_server' is using the discovered Python interpreter at '/opt/homebrew/bin/python3.14', but future installation of another Python interpreter could cause a different interpreter to be discovered. See https://docs.ansible.com/ansible-core/2.20/reference_appendices/interpreter_discovery.html for more information.
local_server | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/opt/homebrew/bin/python3.14"
    },
    "changed": false,
    "ping": "pong"
}
` which python
/Users/f0x1d/.local/share/mise/install/python/3.13.9/bin/python
` ansible my_servers -m setup -c local
[WARNING]: Host 'local_server' is using the discovered Python interpreter at '/opt/homebrew/bin/python3.14', but future installation of another Python interpreter could cause a different interpreter to be discovered. See https://docs.ansible.com/ansible-core/2.20/reference_appendices/interpreter_discovery.html for more information.
local_server | SUCCESS => {
    "ansible_facts": {
        "ansible_all_ipv4_addresses": [
            "192.168.1.147",
            "172.31.19.50"
        ],
        "ansible_all_ipv6_addresses": [
            "fe80::1074:f49b:c9c2:2971%en0",
            "fda2:f617:c656:465f:c01:2aef:7780:cf7",
            "fe80::bc6e:46ff:fe43:5980%awd10",
            "fe80::bc6e:46ff:fe43:5980%llw0",
            "fe80::9bd9:5f11:a282:b92%utun0",
            "fe80::8528:ff22:d8c1:4a84%utun1",
            "fe80::7b2b:df5c:df8e:9c1%utun2",
            "fe80::ce81:b1c:bd2c:69e%utun3",
        ]
    }
}
```

3 Использование ad-hoc команд

На следующем этапе были опробованы ad-hoc команды Ansible для выполнения простых операций.

Сначала с помощью модуля shell был создан текстовый файл в домашней директории пользователя с тестовым содержимым. После этого было вручную проверено, что файл был успешно создан и содержит ожидаемый текст.

```
● ○ ● 📂 .../Documents/temp/lab3ansible
❯ rm test.txt
❯ ansible my_servers -c local -m shell -a 'echo testfilecontent > $HOME/Documents/temp/lab3ansible/test.txt'
[WARNING]: Host 'local_server' is using the discovered Python interpreter at '/opt/homebrew/bin/python3.14', but future installation of another Python interpreter could cause a different interpreter to be discovered. See https://docs.ansible.com/ansible-core/2.20/reference_appendices/interpreter_discovery.html for more information.
local_server | CHANGED | rc=0 >>

❯ cat test.txt
testfilecontent
~/Documents/temp/lab3ansible ) |
```

Затем созданный файл был удалён с помощью модуля file, что подтвердило возможность управления файловой системой целевого хоста с использованием Ansible.

```
● ○ ● 📂 .../Documents/temp/lab3ansible
❯ rm test.txt
❯ ansible my_servers -c local -m shell -a 'echo testfilecontent > $HOME/Documents/temp/lab3ansible/test.txt'
[WARNING]: Host 'local_server' is using the discovered Python interpreter at '/opt/homebrew/bin/python3.14', but future installation of another Python interpreter could cause a different interpreter to be discovered. See https://docs.ansible.com/ansible-core/2.20/reference_appendices/interpreter_discovery.html for more information.
local_server | CHANGED | rc=0 >>

❯ cat test.txt
testfilecontent
❯ ansible my_servers -c local -m file -a 'path=$HOME/Documents/temp/lab3ansible/test.txt state=absent'
[WARNING]: Host 'local_server' is using the discovered Python interpreter at '/opt/homebrew/bin/python3.14', but future installation of another Python interpreter could cause a different interpreter to be discovered. See https://docs.ansible.com/ansible-core/2.20/reference_appendices/interpreter_discovery.html for more information.
local_server | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/opt/homebrew/bin/python3.14"
    },
    "changed": true,
    "path": "/Users/f0x1d/Documents/temp/lab3ansible/test.txt",
    "state": "absent"
}
❯ ls -a
. .. ansible.cfg inventory
~/Documents/temp/lab3ansible |
```

Данный этап позволил ознакомиться с принципами работы ad-hoc команд и модулями Ansible для выполнения единичных задач.

Часть 2. Установка Caddy

1 Создание роли Ansible

После изучения базовых ad-hoc команд был выполнен переход к более структурированному подходу использованию playbook'ов и ролей Ansible.

Для этого в рабочей директории была создана папка roles, после чего с помощью утилиты ansible-galaxy была инициализирована роль caddy_deploy. В результате была автоматически создана стандартная структура роли, включающая директории для задач, шаблонов, переменных и других компонентов.

2 Реализация задач роли

В файле tasks/main.yml роли caddy_deploy были последовательно описаны шаги установки веб-сервера Caddy.

Сначала были установлены необходимые системные пакеты и зависимости, требуемые для работы с HTTPS-репозиториями. Далее был добавлен GPG-ключ официального репозитория Caddy, после чего был подключен сам репозиторий.

На заключительном этапе данной части был выполнен процесс установки пакета caddy с обновлением кэша пакетов. После выполнения playbook'a веб-сервер Caddy был успешно установлен в системе.

3 Создание основного playbook'a

Для запуска роли был создан основной playbook, в котором была указана группа хостов my_servers, тип подключения local, а также роль caddy_deploy.

Запуск playbook'a показал успешное выполнение всех шагов, что подтвердило корректность настройки роли и работоспособность Ansible playbook'ов.

```

changed: [local_server]

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

PLAY RECAP *****
local_server : ok=6    changed=3    unreachable=0    failed=0    skipped=0    rescued
=0    ignored=0

root@anton:~/ansible# service caddy status
WARNING: terminal is not fully functional
Press RETURN to continue
● caddy.service - Caddy
   Loaded: loaded (/lib/systemd/system/caddy.service; enabled; vendor preset: enabled)
     Active: active (running) since Sun 2025-12-14 18:24:14 UTC; 30s ago
       Docs: https://caddyserver.com/docs/
 Main PID: 1399268 (caddy)
    Tasks: 7 (limit: 2262)
      Memory: 11.0M
        CPU: 94ms
      CGroup: /system.slice/caddy.service
              └─1399268 /usr/bin/caddy run --environ --config /etc/caddy/Caddyfile

Dec 14 18:24:14 anton caddy[1399268]: {"level":"warn","ts":1765707854.6425793,"logger":"http.auto_>
Dec 14 18:24:14 anton caddy[1399268]: {"level":"warn","ts":1765707854.6428156,"logger":"http","msg">
Dec 14 18:24:14 anton caddy[1399268]: {"level":"warn","ts":1765707854.6428294,"logger":"http","msg">
Dec 14 18:24:14 anton caddy[1399268]: {"level":"info","ts":1765707854.6428342,"logger":"http.log",>
Dec 14 18:24:14 anton caddy[1399268]: {"level":"info","ts":1765707854.6430438,"logger":"tls.cache",>
Dec 14 18:24:14 anton caddy[1399268]: {"level":"info","ts":1765707854.6432042,"msg":"autosaved con>
Dec 14 18:24:14 anton systemd[1]: Started Caddy.
Dec 14 18:24:14 anton caddy[1399268]: {"level":"info","ts":1765707854.6456113,"msg":"serving initi>
Dec 14 18:24:14 anton caddy[1399268]: {"level":"info","ts":1765707854.6496282,"logger":"tls","msg">
Dec 14 18:24:14 anton caddy[1399268]: {"level":"info","ts":1765707854.6526196,"logger":"tls","msg">
lines 1-21/21 (END)

```

Часть 3. Домен и настройка Caddyfile

1 Регистрация доменного имени

Для проверки работы веб-сервера Caddy с HTTPS был зарегистрирован бесплатный домен на сервисе duckdns.org, который был привязан к IP-адресу сервера. Это позволило использовать полноценное доменное имя и автоматически получить TLS-сертификат от Let's Encrypt.

The screenshot shows the Duck DNS web interface. At the top, there's a yellow rubber duck icon. To its right, the text "Duck DNS" is displayed. Below this, account information is shown: account F0x1d@github, type free, token b7ff3faa-7f6b-43f7-a992-87e485eb7805, token generated 1 minute ago, and created date 14 Dec 2025, 10:26:07. A green banner at the bottom of this section says "success: ip address for verycooldomainswag.duckdns.org updated to 178.72.131.82". The main area is titled "domains 1/5" and contains a table with columns: domain, current ip, ipv6, and changed. One row is visible for the domain "verycooldomainswag" with current ip "178.72.131.82", no ipv6 address listed, and last update "0 seconds ago". There are "update ip" and "delete domain" buttons for this row. At the bottom of the page, a note states: "This site is protected by reCAPTCHA and the Google Privacy Policy and Terms of Service apply."

2 Использование шаблонов и переменных

Для гибкой настройки Caddy был создан шаблон конфигурационного файла Caddyfile с использованием синтаксиса Jinja2. В шаблоне использовались переменные для доменного имени и параметров логирования. Значения переменных были вынесены в отдельный файл vars/main.yml, что позволило легко изменять настройки без правки самого шаблона.

3 Автоматическое создание конфигурации и перезагрузка сервиса

В задачи роли были добавлены шаги по генерации конфигурационного файла Caddy из шаблона и его размещению в системной директории. После обновления конфигурации сервис Caddy был автоматически перезагружен, что обеспечило применение новых настроек без полной остановки сервиса.

```

TASK [Gathering Facts] ****
[WARNING]: Platform linux on host local_server is using the discovered Python interpreter at
/usr/bin/python3.11, but future installation of another Python interpreter could change the
meaning of that path. See https://docs.ansible.com/ansible-
core/2.17/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] ****
ok: [local_server]

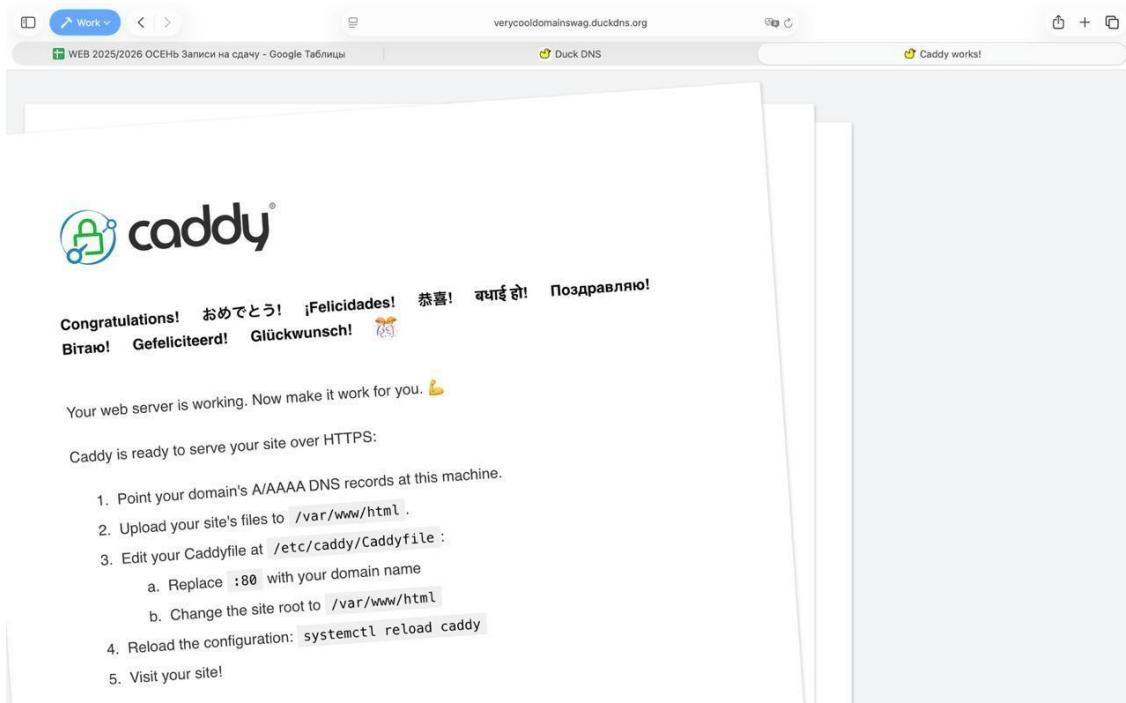
TASK [caddy_deploy : Install Caddy webserver] ****
ok: [local_server]

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    rescued
               ignored=0

```



Задания

1 Переписывание ad-hoc команд в формат playbook

В рамках дополнительного задания был создан отдельный playbook, который полностью повторяет логику ad-hoc команд, но в структурированном виде.

```
---
  name: Files actions
  hosts: my_servers
  connection: local
  gather_facts: true

  vars:
    test_file: "{{ ansible_env.HOME }}/test.txt"
    content_step_1: "test_file_content"
    content_step_3: "updated_content_before_delete"

  tasks:
    - name: 1 - create file with initial content
      ansible.builtin.copy:
        dest: "{{ test_file }}"
        content: "{{ content_step_1 }}"
        mode: "0644"
    - name: 2 - verify file exists
      ansible.builtin.stat:
        path: "{{ test_file }}"
      register: st
    - name: 3 - change file content before deletion
      ansible.builtin.copy:
        dest: "{{ test_file }}"
        content: "{{ content_step_3 }}"
        mode: "0644"
      when: st.stat.exists
    - name: 4 - delete file
      ansible.builtin.file:
        path: "{{ test_file }}"
        state: absent
---

file_playbook.yml
1.1
All
```

Playbook последовательно:

1. Создаёт текстовый файл
2. Изменяет его содержимое
3. Удаляет файл

Это позволило закрепить навыки написания простых playbook'ов и продемонстрировать преимущества декларативного подхода Ansible.

2 Расширение конфигурации веб-сервера Caddy

Конфигурация веб-сервера Caddy была расширена путём добавления пользовательских HTTP-заголовков. Изменения были внесены в шаблон конфигурационного файла Caddyfile, который используется Ansible для автоматической генерации конфигурации сервера.

```
  {{ domain_name }} {
    root * {{ site_root }}
    file_server

    header {
      X-Lab "lab03"
      X-Content-Type-Options "nosniff"
      Referrer-Policy "no-referrer"
      Permissions-Policy "geolocation=()"
    }

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

  {{ log_level }} {{ log_file }} {{ log_level }} {{ log_file }}
```

1,1

All

В роль Ansible были добавлены новые шаги, отвечающие за создание пользовательского HTML-файла index.html. Данный файл используется в качестве стартовой страницы веб-сервера вместо стандартной страницы Caddy.

```
state: present

- name: ensure site root exists
  ansible.builtin.file:
    path: "{{ site_root }}"
    state: directory
    mode: "0755"
  become: true

- name: put custom index.html (Hello world)
  ansible.builtin.copy:
    dest: "{{ site_root }}/index.html"
    content: |
      <!doctype html>
      <html lang="en">
      <head>
        <meta charset="utf-8" />
        <title>Hello</title>
      </head>
      <body>
        <h1>Hello world</h1>
      </body>
    </html>
    mode: "0644"
  become: true

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

- name: Reload with new config
roles/caddy_deploy/tasks/main.yml
```

31,1

98%

После применения обновлённой конфигурации при обращении к серверу отображается пользовательская страница. В HTTP-ответе сервера присутствуют добавленные пользовательские заголовки, что подтверждает корректность выполненных изменений.



Hello world

Вывод

В ходе лабораторной работы был установлен и настроен Ansible, подготовлено рабочее окружение и выполнена настройка inventory. Были изучены и применены ad-hoc команды для управления целевым хостом, а также освоен формат playbook'ов и ролей Ansible.

С использованием Ansible был автоматизирован процесс установки и настройки веб-сервера Caddy. Конфигурация сервера была вынесена в шаблон с использованием переменных, что обеспечило гибкость и удобство управления настройками. Дополнительно конфигурация веб-сервера была расширена, добавлены пользовательские HTTP-заголовки и развернута собственная стартовая HTML-страница.

В результате выполнения работы были получены практические навыки автоматизации администрирования сервисов и управления конфигурациями с использованием Ansible.