

Предисловие

Репозиторий на GitHub: <https://github.com/pashokitsme/net-tech-ansible>

Этот репозиторий, как и этот отчёт - первый этап braim.org/network-contest

Изначально оформлено в .md и предполагалось к просмотру в README репозитория.

В ходе выполнения использовался пакетный (проектный) менеджер [uv](#). Поэтому для вызова команд используется `uv run <command>`.

В качестве машин использовались три докер-контейнера с базовым образом `alpine` и установленным `python3` & `openssh`. Публичные и приватные ключи намеренно оставлены в `keys`, поскольку необходимы только для доступа к контейнерам, проверка хоста также отключена для удобства (т.к. требует `yes` после каждого ребилда).

Задание

Создание venv и установка ansible

Я обычно не использую ansible, поэтому он не был установлен на моём компьютере. Предпочту использовать venv для этого.

```
repos/it-planet/net-tech-ansible
> uv venv
Using CPython 3.13.1 interpreter at: /opt/homebrew/opt/python@3.13/bin/python3.13
Creating virtual environment at: .venv
Activate with: source .venv/bin/activate

repos/it-planet/net-tech-ansible
> uv run ansible --version
ansible [core 2.18.1]
  config file = None
  configured module search path = ['/Users/pavel.smirnov/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /Users/pavel.smirnov/Source/repos/it-planet/net-tech-ansible/.venv/lib/python3.13/site-packages/ansible
  ansible collection location = /Users/pavel.smirnov/.ansible/collections:/usr/share/ansible/collections
  executable location = /Users/pavel.smirnov/Source/repos/it-planet/net-tech-ansible/.venv/bin/ansible
  python version = 3.13.1 (main, Dec  3 2024, 17:59:52) [Clang 15.0.0 (clang-1500.1.0.2.5)] (/Users/pavel.smirnov/Source/repos/it-planet/net-tech-ansible/.venv/bin/python3)
  jinja version = 3.1.5
  libyaml = True
```

Dockerfile

Для ansible необходимо наличие openssh и python3, но в alpine по умолчанию их нет, поэтому нужен [Dockerfile](#).

```
FROM alpine:latest

ARG PUBKEY_PATH

RUN apk update
RUN apk add --no-cache openssh python3

RUN mkdir -p /var/run/sshd && \
    echo 'root:root' | chpasswd && \
    ssh-keygen -A

RUN sed -i 's/#PermitRootLogin prohibit-password/PermitRootLogin yes/' /etc/ssh/sshd_config
RUN sed -i 's/#PubkeyAuthentication/PubkeyAuthentication/' /etc/ssh/sshd_config

EXPOSE 22

CMD ["/usr/sbin/sshd", "-D"]
```

Создание docker-compose файла и поднятие контейнеров

Полная конфигурация представлена в файле `docker-compose.yml`.

Если кратко: публичные ключи из `keys` монтируются как `/root/.ssh/authorized_keys`, а для сервера `srv01` монтируется директория `data` -> `/etc/ansible/IT-Planet`, для того, чтобы был доступ с хоста.

`services:`

```

srv01:
  build: .
  container_name: srv01
  hostname: srv01
  volumes:
    - ./keys/srv01.pub:/root/.ssh/authorized_keys
    - ./data:/etc/ansible/IT-Planet
  networks:
    - net01
  ports:
    - 6001:22

host01:
  build: .
  container_name: host01
  hostname: host01
  volumes:
    - ./keys/host01.pub:/root/.ssh/authorized_keys
  networks:
    - net01
  ports:
    - 6002:22

host02:
  build: .
  container_name: host02
  hostname: host02
  volumes:
    - ./keys/host02.pub:/root/.ssh/authorized_keys
  networks:
    - net01
  ports:
    - 6003:22

networks:
  net01:
    driver: bridge
    ipam:
      config:
        - subnet: 10.0.2.0/24
          gateway: 10.0.2.1

```

Использование: `docker compose up -d --build`.

```

repos/it-planet/net-tech-ansible
> docker compose up -d
[+] Running 0/0
  ⚡ Service host02 Building
  ⚡ Service srv01 Building
  ⚡ Service host01 Building
[+] Building 0.1s (18/18) FINISHED
=> [host02 internal] load build definition from Dockerfile
=> => transferring dockerfile: 360B
=> [host01 internal] load build definition from Dockerfile
=> => transferring dockerfile: 360B
=> [srv01 internal] load build definition from Dockerfile
=> => transferring dockerfile: 360B
=> [srv01 internal] load metadata for docker.io/library/alpine:latest
=> [host02 internal] load .dockerrcignore
=> => transferring context: 2B
=> [host01 internal] load .dockerrcignore
=> => transferring context: 2B
=> [srv01 internal] load .dockerrcignore
=> => transferring context: 2B
=> [srv01 1/5] FROM docker.io/library/alpine:latest
=> CACHED [host01 5/5] RUN sed -i '/#PermitRootLogin prohibit-password/PermitRootLogin yes/' /etc/ssh/sshd_config
=> CACHED [host01 2/5] RUN apk update
=> CACHED [host01 3/5] RUN apk add --no-cache openssh
=> CACHED [host01 4/5] RUN mkdir -p /var/run/sshd && echo 'root:root' | chpasswd && ssh-keygen -A
=> [host01] exporting to image
=> => exporting layers
=> => writing image sha256:711377ced06e92179c53b6fcdf1dacdecf06ff98bf35eea019d5c11eadc69223
=> => naming to docker.io/library/net-tech-ansible-host02
=> [srv01] exporting to image
=> => exporting layers
=> => writing image sha256:89b853a6bc51bdf9300812bc3945314a5985663bdf3902ed6d906d74f1157ea8
=> => naming to docker.io/library/net-tech-ansible-srv01
=> [host02] exporting to image
=> => exporting layers
=> => writing image sha256:ddaa6ebcc6e340e97debf6dcbe13bc2e9457863af7d65c76657ddef038da4930
[+] Running 7/70 docker.io/library/net-tech-ansible-host02
✓ Service host02 Built
✓ Service srv01 Built
✓ Service host01 Built
✓ Network net-tech-ansible_net01 Created
✓ Container host01 Started
✓ Container srv01 Started
✓ Container host02 Started

```

Создание ansible inventory

Полная конфигурация представлена в файле `inventory.yml`.

Выключена проверка хоста, почему - описано выше. Порты для ssh указаны такие, как в `docker-compose.yml`, как и ключи.

```

Ansible Inventory - Ansible inventory files (inventory.json)
1   all:
2     vars:
3       ansible_python_interpreter: /usr/bin/python3.12
4       ansible_user: root
5       ansible_host_key_checking: false
6
7     children:
8       servers:
9         hosts:
10      |   srv01:
11      |     ansible_host: 127.0.0.1
12      |     ansible_port: 6001
13      |     ansible_ssh_private_key_file: ./keys/srv01
14
15     clients:
16       vars:
17         ansible_host: 127.0.0.1
18
19       hosts:
20         host01:
21           ansible_port: 6002
22           ansible_ssh_private_key_file: ./keys/host01
23
24         host02:
25           ansible_port: 6003
26           ansible_ssh_private_key_file: ./keys/host02
27

```

Тестовый пинг

Использование: `[uv run ansible -i inventory.yml -m ping all]`.

```

net-tech-ansible master*--? 0x7F
[> uv run ansible all -m ping -i inventory.yml
srv01 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
host01 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
host02 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}

```

Создание playbook

Конфигурация также представлена в файле `playbook.yml`.

```

- name: Gather client infos
  hosts: clients
  vars:
    output_dir: '/etc/ansible/IT-Planet/'
    output_file: '{{ [output_dir, inventory_hostname] | path_join }}'
    output_host: srv01

  tasks:
    - name: Gather local ip address
      ansible.builtin.shell: hostname -i
      register: ip_local

    - name: Gather hostname

```

```

ansible.builtin.shell: hostname
register: hostname

- name: Gather os info
  ansible.builtin.shell: cat /proc/version
  register: os_info

- name: Gather disk usage
  ansible.builtin.shell: df -H /
  register: disk_usage

- name: Gather memory usage
  ansible.builtin.shell: free -m
  register: memory_usage

- name: Ensure directory exists
  ansible.builtin.file:
    path: '{{ output_dir }}'
    state: directory
  delegate_to: '{{ output_host }}'

- name: Save client infos
  ansible.builtin.copy:
    content: |
      Hostname: {{ hostname.stdout }}
      IP Address: {{ ip_local.stdout }}
      OS: {{ os_info.stdout }}

      Memory Usage:
      {{ memory_usage.stdout }}

      Disk Usage:
      {{ disk_usage.stdout }}
    dest: '{{ output_file }}'
  delegate_to: '{{ output_host }}'

```

Выполнение playbook

Использование: `[uv run ansible-playbook -i inventory.yml playbook.yml]`.

```

repos/it-planet/net-tech-ansible 3s
> uv run ansible-playbook -i inventory.yml playbook.yml

PLAY [Gather client infos] ****
TASK [Gathering Facts] ****
ok: [host02]
ok: [host01]

TASK [Gather local ip address] ****
changed: [host02]
changed: [host01]

TASK [Gather hostname] ****
changed: [host01]
changed: [host02]

TASK [Gather os info] ****
changed: [host01]
changed: [host02]

TASK [Gather disk usage] ****
changed: [host01]
changed: [host02]

TASK [Gather memory usage] ****
changed: [host01]
changed: [host02]

TASK [Ensure directory exists] ****
ok: [host01 -> srv01(127.0.0.1)]
ok: [host02 -> srv01(127.0.0.1)]

TASK [Save client infos] ****
changed: [host02 -> srv01(127.0.0.1)]
changed: [host01 -> srv01(127.0.0.1)]

PLAY RECAP ****
host01           : ok=8    changed=6   unreachable=0    failed=0    skipped=0   rescued=0    ignored=0
host02           : ok=8    changed=6   unreachable=0    failed=0    skipped=0   rescued=0    ignored=0

```

Результат

Директория `[data]` примонтирована к контейнеру `[srv01]` к директории `/etc/ansible/IT-Planet`.

```
repos/it-planet/net-tech-ansible 2s
[] cat data/host@1
Hostname: host01
IP Address: 10.0.2.4
OS: Linux version 6.10.14-linuxkit (root@buildkitsandbox) (gcc (Alpine 13.2.1_git20240309) 13.2.1 20240309, GNU ld (GNU Binutils) 2.42) #1 SMP Fri N
ov 29 17:22:03 UTC 2024

Memory Usage:
total        used         free        shared      buff/cache   available
Mem:       3919          417        3167            3          336       3361
Swap:      1024           0        1024

Disk Usage:
Filesystem      Size  Used Available Use% Mounted on
overlay        31.3G  6.2G    23.5G  21% /
```

```
repos/it-planet/net-tech-ansible 2s
[] cat data/host@2
Hostname: host02
IP Address: 10.0.2.3
OS: Linux version 6.10.14-linuxkit (root@buildkitsandbox) (gcc (Alpine 13.2.1_git20240309) 13.2.1 20240309, GNU ld (GNU Binutils) 2.42) #1 SMP Fri N
ov 29 17:22:03 UTC 2024

Memory Usage:
total        used         free        shared      buff/cache   available
Mem:       3919          418        3165            3          336       3360
Swap:      1024           0        1024

Disk Usage:
Filesystem      Size  Used Available Use% Mounted on
overlay        31.3G  6.2G    23.5G  21% /
```