

МИНЦИФРЫ  
«СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ТЕЛЕКОММУНИКАЦИЙ И ИНФОРМАТИКИ»

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3  
«Базовая часть ansible и управления конфигурацией»  
по дисциплине «Архитектура вычислительных систем»

Выполнил: студент гр. ИП-212

Жеребцов Дмитрий Евгеньевич

Новосибирск 2024

# Постановка задачи

## Задание.

Создать пару виртуальных машин/контейнеров в которых установлено:

- sshd и настроен доступ по паролю/ключу
- python3
- python3-pip
- Инициализировать среду для ansible на “управляющей” системе
  - Создать структуру каталогов {inventory,playbooks,roles}
  - Создать ansible.cfg для работы с созданными каталогами (Указать расположение и назначение директорий, например roles\_path = roles)
  - Сформировать inventory для созданных виртуальных машин/контейнеров
  - Сконфигурировать параметры подключения к системам - имя пользователя ansible\_user, путь до приватного ключа ansible\_ssh\_private\_key\_file
  - Проверить доступность хостов - `ansible -m ping all` или отдельного хоста `ansible -m ping \${host1?}`
- Чтобы посмотреть структуру информации, собираемой ansible о хостах при подключении (gather facts), можно использовать команду `ansible -m setup $ {host?}`, где host - имя хоста в inventory. Результатом выполнения команды является json с информацией о системе. Цель - получить отдельные поля этого json и сформировать свой файл
- Для выполнения ансблом команд docker, пользователь должен быть в группе docker, или иметь доступ к sudo без пароля (Для повышения привелегий в ansible используется `become: true`)
- DoD на 3
  - ansible.cfg
  - директория inventory, содержащая информацию о подключении к хостам
  - Продемонстрировать выполнение ad-hoc команды `ansible -m ping $ {host1?},${host2?}`, где hostN - имя хоста в inventory
  - Продемонстрировать выполнение ad-hoc команды `ansible -m shell -a 'hostname`, где hostN - имя хоста в inventory. Вместо hostname допускается выполнить другую команду
- DoD на 4
  - DoD на 3
  - Написать playbook, который сформирует структурированный файл (yaml, json), содержащий информацию о системе (Для примера можно использовать список параметров из задания 1)
  - Использовать модуль `ansible.builtin.template` (Отдельно создать .j2 шаблон)
  - Написанный playbook разместить в директории playbooks. Шаблон разместить в playbooks/templates. Шаблоны пишутся под jinja2
  - Продемонстрировать результат выполнения playbook на хостах из inventory. Результат (файл с информацией о системе) может размещаться как на управляющей системе, так и на удаленной
- DoD на 5
  - DoD на 4
  - Директория roles, содержащая код отдельных ролей ansible
  - Создать роль для выполнения сценария - например, генерация docker-compose файла и запуск контейнеров из него через модуль `docker_compose_v2` (Создать стандартные файлы роли можно командой `ansible-galaxy role init ${rolename?}`, где rolename - имя роли)
  - Вызвать из playbook созданную роль, продемонстрировать результат

# Выполнение работы

```
FROM ubuntu:latest

RUN apt update && apt install -y openssh-server python3 python3-pip sudo

RUN mkdir /var/run/sshd

RUN echo 'root:1234' | chpasswd

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

EXPOSE 22

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

/nix/store/mrw5mhwvgr465h1xf2pjiyabvlgkx-python3.12-paramiko-3.4.0/lib/python3.12/site-packages/paramiko/pkey.py:180: CryptographyDeprecationWarning: TripleDES has been moved to cryptography.hazmat.decrepit.cipheralgorithms.TripleDES
  "cipher": algorithms.TripleDES,
/nix/store/mrw5mhwvgr465h1xf2pjiyabvlgkx-python3.12-paramiko-3.4.0/lib/python3.12/site-packages/paramiko/transport.py:259: CryptographyDeprecationWarning: TripleDES has been moved to cryptography.hazmat.decrepit.cipheralgorithms.TripleDES,
  "class": algorithms.TripleDES,
[WARNING]: Platform linux on host container1 is using the discovered Python interpreter at /usr/bin/python3.12, 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.
container1 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3.12"
  },
  "changed": false,
  "ping": "pong"
}
[WARNING]: Platform linux on host container2 is using the discovered Python interpreter at /usr/bin/python3.12, 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.
container2 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3.12"
  },
  "changed": false,
  "ping": "pong"
}

* ansible-playbook -i inventory/hosts.ini playbooks/system_info.yml
/nix/store/mrw5mhwvgr465h1xf2pjiyabvlgkx-python3.12-paramiko-3.4.0/lib/python3.12/site-packages/paramiko/pkey.py:180: CryptographyDeprecationWarning: TripleDES has been moved to cryptography.hazmat.decrepit.cipheralgorithms.TripleDES,
  "cipher": algorithms.TripleDES,
/nix/store/mrw5mhwvgr465h1xf2pjiyabvlgkx-python3.12-paramiko-3.4.0/lib/python3.12/site-packages/paramiko/transport.py:259: CryptographyDeprecationWarning: TripleDES has been moved to cryptography.hazmat.decrepit.cipheralgorithms.TripleDES,
  "class": algorithms.TripleDES,

PLAY [Playbook yaml] *****

TASK [Gathering Facts] *****
[WARNING]: Platform linux on host container2 is using the discovered Python interpreter at /usr/bin/python3.12, 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: [container2]
[WARNING]: Platform linux on host container1 is using the discovered Python interpreter at /usr/bin/python3.12, 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: [container1]

TASK [JSON generator] *****
ok: [container1]
ok: [container2]

TASK [Create system info JSON] *****
changed: [container1]
changed: [container2]

PLAY RECAP *****
container1      : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
container2      : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```