Инструкцию по установке Single архитектуры CS при помощи ansible и openstack CLI

Содержание

1. Архитектура

В данной инструкции рассмотрим скрипты автоматизации деплоя и организации необходимых тестовых контуров в автоматическом режиме программного продукта Р7 Сервер Базовый в архитектуре **Single**.

В архитектуре типа **Single** все модули системы инсталлируются в рамках одного сервера. Данная установка является стандартным вариантом установки Программного комплекса.

Архитектура позволяет без дополнительных затрат на инфраструктуру (сервера, дисковые массивы) эксплуатировать Программный комплекс, применяя минимальный функционал по резервированию данных и системы в целом.

2. Создание ВМ и установка Продукта

Инструмент, который понадобится для автоматизированной установки продукта - это **ansible** и настроенный доступ к **openstack CLI**.

Примечание: Некоторые **tasks** использованы из-за особенностей опыта эксплуатации на нашей архитектуре, скорее всего для Вас потребуется лишь основная функциональная часть кода.

Для данного способа необходимо предварительно создать ключ.

Пример команды:

```
# Добавляем открытый ключ
openstack keypair create <имя ключа> --public-key <путь до открытого
ключа>
```

2.1 Переменные, которые используются для поднятия ВМ

```
---
availability_zone : "ru-3b"
# имя созданного ранее ключа
public_key : ansible
# конфигурация сервера
flavor : PRC50.4-8192
volume_type : fast.ru-3b
# имя диска и ВМ, можно использовать одну переменную
disk_name : kh_single_cs
server_name : kh_single_cs
# размер диска
disk_size : 60
# переменные для подключения к ВМ по ssh
# порт ssh у нас изменён с 22 на 2235
ansible_port : 2235
ansible_user: root
# Имя образа или его id
UUID_image: "AltLinuxServer_10+net+ssh+nx"
```

2.2 Основные tasks по созданию ВМ

```
---
# Создание диска для ВМ с образом
- name: create disk
  os_volume:
    state: present
    bootable: yes
    availability_zone: "{{ availability_zone }}"
    size: "{{ disk_size }}"
    image: "{{ UUID_image }}"
    volume_type: "{{ volume_type }}"
    display_name: "{{ disk_name }}"
  register: volume_info

- name: volume_info
  debug:
    msg: "{{ volume_info }}"

# Создание ВМ на основе созданного диска
- name: create instance
  os_server:
    name: "{{ server_name }}"
    boot_volume: "{{ volume_info.volume.name }}"
    key_name: "{{ public_key }}"
    timeout: 200
    availability_zone: "{{ availability_zone }}"
    auto_ip: no
    network: nat
    flavor: "{{ flavor }}"
  register: vm_info

- name: vm_info
  debug:
    msg: "{{ vm_info }}"
# добавление хоста в память inventory на время выполнения playbook
- name: add host cs-01
  ansible.builtin.add_host:
    name: cs-01
    groups: cs
    ansible_host: "{{ vm_info.server.private_v4 }}"
```

```
# ожидается статус Active сервера
- name: check status server {{ vm_info.server.id }}
  ansible.builtin.command: openstack server show
"{{ vm_info.server.id }}" -f json
  register: active
  until: "'ACTIVE' in active.stdout"
  retries: 100
  delay: 15

# Удаление подписи ssh, не обязательно, но т.к. создаётся постоянно много
# ВМ, то может возникнуть проблема с подключением
- name: update key
  ansible.builtin.command: ssh-keygen -R
["{{ vm_info.server.private_v4 }}"]:2235

# Ожидание доступности по ssh порту
- name: waiting when 2235 port is ready
  wait_for:
    port: "{{ ansible_port }}"
    host: "{{ hostvars['cs-01'].ansible_host }}"
    search_regex: OpenSSH
    connect_timeout: 600
    delay: 10

# Добавление второго сетевого интерфейса
- name: add network
  ansible.builtin.command: openstack server add network
"{{ vm_info.server.id }}" 6f25d493-483e-4157-8cad-5a7e338666fa

# Ожидание 10с до продолжения
- name: sleep 10s
  pause:
    seconds: 10

# Перезагрузка сервера средствми openstack CLI
- name: reboot server openstack
  ansible.builtin.command: openstack server reboot
"{{ vm_info.server.id }}"
  register: reboot
  until: reboot.rc == 0
  retries: 10
  delay: 15
```

```
- name: check status server {{ vm_info.server.id }}
  ansible.builtin.command: openstack server show
"{{ vm_info.server.id }}" -f json
  register: active
  until: "'ACTIVE' in active.stdout"
  retries: 100
  delay: 15

- name: sleep 10s
  pause:
    seconds: 10

- name: waiting when 2235 port is ready
  wait_for:
    port: "{{ ansible_port }}"
    host: "{{ hostvars['cs-01'].ansible_host }}"
    search_regex: OpenSSH
    connect_timeout: 600
    delay: 10

# сбор информации по серверу для дальнейшего использования
- name: info for server
  ansible.builtin.setup:
  delegate_to: cs-01
  delegate_facts: True
  register: ipv4_172

- name: ipv4_172
  debug:
    msg: "{{ ipv4_172 }}"

# обновления файла с открытыми ключами,
# если нужен доступ будет кому-то ещё
- name: add file /root/.ssh/authorized_keys
  ansible.builtin.copy:
    src: authorized_keys
    dest: /root/.ssh/authorized_keys
    owner: root
    group: root
    mode: '0600'
  become: true
  delegate_to: cs-01
```

```
- name: waiting when 2235 port is ready
  wait_for:
    port: "{{ ansible_port }}"
    host: "{{ hostvars['cs-01'].ansible_host }}"
    search_regex: OpenSSH
    connect_timeout: 600
    delay: 10

- name: sleep 30s
  pause:
    seconds: 30

- name: reboot server openstack
  ansible.builtin.command: openstack server reboot
"{{ vm_info.server.id }}"
  register: reboot
  until: reboot.rc == 0
  retries: 10
  delay: 15

- name: check status server {{ vm_info.server.id }}
  ansible.builtin.command: openstack server show
"{{ vm_info.server.id }}" -f json
  register: active
  until: "'ACTIVE' in active.stdout"
  retries: 20
  delay: 15

- name: sleep 10s
  pause:
    seconds: 10

- name: waiting when 2235 port is ready
  wait_for:
    port: "{{ ansible_port }}"
    host: "{{ hostvars['cs-01'].ansible_host }}"
    search_regex: OpenSSH
    delay: 10
    connect_timeout: 600
```

```
# расширение диска на ВМ, если этого не произошло автоматически.
# в данном примере у нас не расширяется на 2 ОС, поэтому чётко прописано
# условие для выполнения
- name: resize /dev/sda 2
  block:

  - name: growpart
    ansible.builtin.command: growpart /dev/sda 2

  - name: resize2fs
    ansible.builtin.command: resize2fs /dev/sda2

  delegate_to: cs-01
  when: ipv4_172.ansible_facts.ansible_distribution_major_version == '22'

- name: resize /dev/sda 1
  block:

  - name: growpart
    ansible.builtin.command: growpart /dev/sda 1

  - name: resize2fs
    ansible.builtin.command: resize2fs /dev/sda1

  delegate_to: cs-01
  when: ipv4_172.ansible_facts.ansible_distribution == 'CentOS'

# изменение hostname
- name: change hostname
  ansible.builtin.command: hostnamectl set-hostname "{{ server_name }}"
  delegate_to: cs-01

- name: restart systemd-hostnamed
  ansible.builtin.service:
    name: systemd-hostnamed
    state: restarted
  delegate_to: cs-01

# изменение временной зоны
- name: set timezone to Europe/Moscow
  timezone: name=Europe/Moscow
  delegate_to: cs-01
```

```
- name: reboot server openstack
  ansible.builtin.command: openstack server reboot
"{{ vm_info.server.id }}"
  register: reboot
  until: reboot.rc == 0
  retries: 10
  delay: 15

- name: check status server {{ vm_info.server.id }}
  ansible.builtin.command: openstack server show
"{{ vm_info.server.id }}" -f json
  register: active
  until: "'ACTIVE' in active.stdout"
  retries: 100
  delay: 15

- name: sleep 10s
  pause:
    seconds: 10

- name: waiting when 2235 port is ready
  wait_for:
    port: "{{ ansible_port }}"
    host: "{{ hostvars['cs-01'].ansible_host }}"
    search_regex: OpenSSH
    connect_timeout: 600
    delay: 10
# Вывод информации для подключения
- name: ipv4_172_eth1
  debug:
    msg: "{{ ipv4_172.ansible_facts.ansible_eth1.ipv4.address }}"
  when: ipv4_172.ansible_facts.ansible_eth1.ipv4.address is defined

- name: ipv4_172_ens8
  debug:
    msg: "{{ ipv4_172.ansible_facts.ansible_ens8.ipv4.address }}"
  when: ipv4_172.ansible_facts.ansible_ens8.ipv4.address is defined

- name: ipv4_172_ens7
  debug:
    msg: "{{ ipv4_172.ansible_facts.ansible_ens7.ipv4.address }}"
  when: ipv4_172.ansible_facts.ansible_ens7.ipv4.address is defined

- name: ipv4_172_ens4
  debug:
    msg: "{{ ipv4_172.ansible_facts.ansible_ens4.ipv4.address }}"
  when: ipv4_172.ansible_facts.ansible_ens4.ipv4.address is defined
```

```
- name: ipv4_192
  debug:
    msg: "{{ vm_info.server.private_v4 }}"
  when: vm_info.server.private_v4 is defined

- name: name server
  debug:
    msg: "{{ server_name }}"
  when: server_name is defined
```

2.3 Инсталляцию CS на примере ОС Alt Linux

```
---
# Обновление репозитория и пакетов до актуальной версии
- name: update repo
  ansible.builtin.shell:
    cmd: apt-get update

- name: update all packages to their latest version
  ansible.builtin.shell:
    cmd: apt-get dist-upgrade -y

# Скачивание скрипта установки CS
- name: download script
  get_url:
    url: "{{ url_cs_alt }}"
    dest: /tmp/install-ALTLinux.sh
    mode: '0644'
    validate_certs: no
# Удаление пакетов согласно инструкции на сайте
- name: remove packages "Altlinux"
  ansible.builtin.package:
    name: "{{ packagesAlt }}"
    state: absent

# Удаление данных БД MySQL
- name: recursively remove directory
  ansible.builtin.file:
    path: /var/lib/mysql/db
    state: absent
```

```
# Остановка сервисов, которые могут занимать 80, 443 порты
- name: stop service httpd2
  ansible.builtin.service:
    name: httpd2
    state: stopped
    enabled: yes
  ignore_errors: yes

- name: stop service ahttpd
  ansible.builtin.service:
    name: ahttpd
    state: stopped
    enabled: yes
  ignore_errors: yes

# Установка Сервера Совместной работы
- name: install cs
  ansible.builtin.shell:
    cmd: bash /tmp/install-ALTLinux.sh
  register: runscriptalt

- name: debug runscriptalt
  debug:
    msg: "{{ runscriptalt }}"

# Запуск сервисов для определённой версии Alt Linux
- name: start service
  ansible.builtin.shell:
    cmd: systemctl start monoserve monoserveApiSystem r7-officeTelegram
r7-officeRadicale r7-officeSocketIO r7-officeThumb r7-officeUrlShortener
r7-officeBackup r7-officeFeed r7-officeIndex r7-officeIndex r7-
officeNotify r7-officeMailAggregator r7-officeMailWatchdog r7-
officeMailCleaner r7-officeRadicale r7-officeStorageMigrate r7-
officeStorageEncryption
  when: ansible_facts['distribution_release'] == "Altostratus" and
        ansible_facts['distribution_version'] == "9.1"

- name: enable service
  ansible.builtin.shell:
    cmd: systemctl enable monoserve monoserveApiSystem r7-officeTelegram
r7-officeRadicale r7-officeSocketIO r7-officeThumb r7-officeUrlShortener
r7-officeBackup r7-officeFeed r7-officeIndex r7-officeIndex r7-
officeNotify r7-officeMailAggregator r7-officeMailWatchdog r7-
officeMailCleaner r7-officeRadicale r7-officeStorageMigrate r7-
officeStorageEncryption
  when: ansible_facts['distribution_release'] == "Altostratus" and
        ansible_facts['distribution_version'] == "9.1"
```

Запуск **playbook** , который вызовет эти роли происходит командой:

```
ansible-playbook playbooks/vm_and_cs.yml
```

Сам **playbook** выглядит следующим образом:

```
---
# Запуск на локальном хосте создания ВМ
- hosts: localhost
  gather_facts: false
  roles:
    - create_instans

# Запуск роли на удалённом хосте,
# который добавлен в память инвентори в прошлой роли
- hosts: cs
  gather_facts: true
  become: true
  roles:
    - install_cs
  vars_files:
    - ../vars/var_r7.yml
```

Как вы заметили, у нас указан файл с переменными. В нём указаны ссылки на дистрибутивы. И, чтобы не изменять их в каждой роли или распределять по группам, т.к. он взят с общего пула **roles** и **playbooks**, то было решено вынести данные переменные, для удобства, в отдельный файл:

```
---
# cs
url_cs_redhat: "https://download.r7-office.ru/repo/install-RedHat.sh"
url_cs_deb: "https://download.r7-office.ru/repo/install-AstraLinux.sh"
url_cs_alt: "https://download.r7-office.ru/repo/install-ALTLinux.sh"
# ds
url_ds_redhat: "https://download.r7-office.ru/centos/r7-office-
documentserver-ee.x86_64.rpm"
#url_ds_deb: ""
url_ds_alt: "https://download.r7-office.ru/altlinux/r7-office-
documentserver-ee.x86_64.rpm"
# dsctp
url_dsctp_deb: "https://download.r7-office.ru/ubuntu/r7-office.deb"
url_dsctp_rpm: "https://download.r7-office.ru/centos/r7-office.rpm"
url_dsctp_alt_rpm: "https://download.r7-office.ru/altlinux/r7-office.rpm"
url_dsctp_exe: "https://download.r7-office.ru/windows/r7office_x64.exe"
url_dsctp_exe_32: "https://download.r7-
office.ru/windows/r7office_x86.exe"
url_dsctp_msi: "https://download.r7-office.ru/windows/r7office-7.0.1.msi"
url_dsctp_msi_32: "https://download.r7-office.ru/windows/r7office-
7.0.1.msi"
```

После того, как выполните запуск **playbook** и все шаги пройдут успешно, необходимо будет зарегистрироваться на портале и проверить его работу.