Олимпиадное задание по Ansible

Тема: Организация работы с базами данных в различных контурах

Цель задания:

С помощью Ansible необходимо написать роль для бд mysql, соответствующую следующим требованиям:

- возможность переиспользовать роль для всех баз данных, описанных инвентарем;
- возможность использовать разные контуры через переменную env;
- использование тегов, для разных сценариев работ с базой данных.

Участник должен продемонстрировать умение пользоваться **inventory, group_vars, шаблонами Jinja2, ролями, handlers, tags** и принципами идемпотентности Ansible.

Инфраструктура:

В распоряжении участников есть 2 виртуальных сервера:

- Master (Бастион под ansible)
- DB1 (Сервер под базы данных PROD)
- DB2 (Сервер под базы данных STAGE)

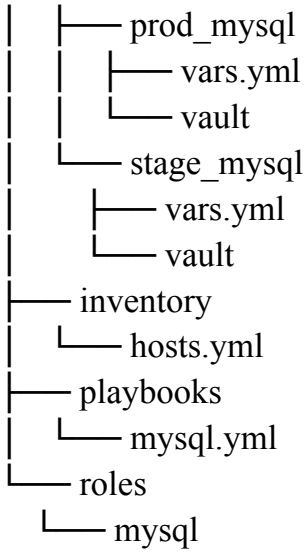
Все хосты доступны по SSH с одинаковыми учётными данными root/toor.

Доступ ansible организуйте по SSH **ключу**! Приложение 1

Требования к структуре проекта:

Проект Ansible должен быть организован по best practices и по окончанию задания быть в git репозитории:

```
.
├── README.md
├── ansible.cfg
├── group_vars
```



Требования к ansible оформлению

1. Inventory

Inventory должен быть в формате **YAML** (hosts). Инвентарь должен хранится по пути inventory/hosts.

Хосты должны быть разделены на группы: prod_mysql, stage_mysql. От этих двух групп должны исходить базы данных с описанием переменных, пример:

```
prod_web_mysql:
   vars:
     vgname: mysql-vg
     service_port: 3306
     create_db_name:
       - prod_web1
       - prod_web2
     db_name: web_database
     mysql:
       wsrep_sync_wait: 1
     mysql_user:
       - user: web_user1
         password: "{{ web_user1_pass }}"
         priv: prod_web1.*:SELECT
         id: 1
       - user: web_user2
         password: "{{ web_user2_pass }}"
         priv: prod_web2.*:SELECT
         id: 2
       - user: prod_user
         password: "{{ prod_user_pass }}"
         id: 3
```

```
hosts:
  prod_db_web:
    ansible_host: 192.168.100.100
```

Создайте все 4 базы данных. В группе prod_mysql: prod_web_mysql, prod_back_mysql. В группе stage_mysql: stage_web_mysql, stage_back_mysql.

1) prod_web_mysql
    a) порт 3306
    b) должны создаться базы web_database, prod_web1, prod_web2
    c) применяемые параметры: wsrep_sync_wait: 1
    d) пользователи:
        i) web_user1
            (1) пароль из переменной web_user1_pass
            (2) id = 1
            (3) Права на SELECT в базе prod_web1
        ii) web_user2
            (1) пароль из переменной web_user2_pass
            (2) id = 2
            (3) Права на SELECT в базе prod_web2
        iii) prod_user2
            (1) пароль из переменной prod_user_pass
            (2) id = 3
            (3) Полные права на все базы
2) prod_back_mysql
    a) порт 3308
    b) должны создаться базы back_database, prod_back1, prod_back2
    c) применяемые параметры: wsrep_sync_wait: 1
    d) пользователи:
        i) back_user1
            (1) пароль из переменной back_user1_pass
            (2) id = 1
            (3) Права на SELECT в базе prod_back1
        ii) back_user2
            (1) пароль из переменной back_user2_pass
            (2) id = 2
            (3) Права на SELECT в базе prod_back2
        iii) prod_user2
            (1) пароль из переменной prod_user_pass
            (2) id = 3

(3) Полные права на все базы

3) stage_web_mysql
    a) порт 3310
    b) должны создаться базы web_database, stage_web1, stage_web2
    c) применяемые параметры: wsrep_sync_wait: 1
    d) пользователи:
        i) web_user1
            (1) пароль из переменной web_user1_pass
            (2) id = 1
            (3) Права на SELECT в базе stage_web1
        ii) web_user2
            (1) пароль из переменной web_user2_pass
            (2) id = 2
            (3) Права на SELECT в базе stagge_web2
        iii) stage_user2
            (1) пароль из переменной stage_user_pass
            (2) id = 3
            (3) Полные права на все базы
4) stage_back_mysql
    a) порт 3309
    b) должны создаться базы back_database, stage_back1, stage_back2
    c) применяемые параметры: wsrep_sync_wait: 1
    d) пользователи:
        i) back_user1
            (1) пароль из переменной back_user1_pass
            (2) id = 1
            (3) Права на SELECT в базе stage_back1
        ii) back_user2
            (1) пароль из переменной back_user2_pass
            (2) id = 2
            (3) Права на SELECT в базе stage_back2
        iii) prod_user2
            (1) пароль из переменной stage_user_pass
            (2) id = 3
            (3) Полные права на все базы

2. Group Vars

group_vars/prod_mysql/: каталог для переменных для prod окружения.

3. Роли

Роли должны храниться в отдельном каталоге roles.

Создайте роль mysql со структурой:

```
├── handlers
│   ├── main..yml
├──tasks
│   ├── main.yml
│   ├── install.yml
│   ├── lvm.yml
│   ├── users.yml
│   ├── любые другие tasks файлы на ваше усмотрение
├──templates
│   ├── my.cnf.j2
│   ├── mysql.service.j2
├──vars
│   ├── main.yml
```

4. Handlers

В роле должны использоваться **handlers** для перезапуска сервисов.

1) рестарт демонов и reload systemd должен происходит исключительно при помощи notify

5. Playbook

Главный плейбук mysql.yml должен запускать роли в зависимости от группы хостов.
Должна быть поддержка **тегов** (users, install).

Хранение плейбуков должно быть организовано в каталоге playbooks

Техническое задание

Файл конфигурации

1) Путь к инвентарю должен быть указан как inventory/hosts
2) Плейбуки должны быть доступны из директории playbooks/
3) Роли должны быть доступны из директории roles/
4) Сбор фактов используйте в режиме smart
5) В качестве интерпретатора укажите /usr/bin/python3
6) Отключите проверку ключей хоста при подключении по SSH

Плейбук

1) Создайте playbook, “mysql.yml”
    a) плейбук должен запускать роль mysql
    b) При запуске, плейбук должен Обязательно принимать extra_vars env=prod/stage
        i) Запускаться одновременно stage и prod окружения не должны
        ii) Без указания лимита на конкретную базу данных, запуск должен осуществляться по принципу env + ’ _mysql’
    c) В pre_tasks организуйте сбор фактов по каждой базе отдельно, в разрезе env + ‘_mysql’

Переменные

1) Организуйте две группы переменных: prod_mysql, stage_mysql

group_vars/prod_mysql/:каталог для переменных для stage окружения.

2) В vars.yml должен быть реализован список pkg, с указанием пакетов для установки mariadb.
    a) mariadb
    b) mariadb_server
    c) mariadb_devel
    d) mariadb_server_utils
    e) версия пакетов 11.4, должна подставляться из переменной-список versions, mariadb: 11.4
3) В файле vault должны хранится переменные
    a) vault_root_mysql_password: "P@ssw0rd"

b) web_user1_pass: "web_user_pass1"
c) web_user2_pass: "web_user_pass2"
d) prod_user_pass: "prod_user_pass"
e) back_user1_pass: "back1"
f) back_user2_pass: "back2"

Файл vault должен быть зашифрован секретной фразой при помощи ansible-vault. Файл должен быть подгружен вместе с ansible в репозиторий git. Путь укажите в ansible.cfg, расшифрование файла должен происходить без указаний дополнительных параметров и паролей.

group_vars/stage_mysql/:каталог для переменных для stage окружения.

4) В vars.yml должен быть реализован список pkg, с указанием пакетов для установки mariadb.
    a) mariadb
    b) mariadb_server
    c) mariadb_devel
    d) mariadb_server_utils
    e) версия пакетов 11.4, должна подставляться из переменной-список version, mariadb: 11.4
5) В файле vault должны хранится переменные
    a) vault_root_mysql_password: "P@ssw0rd"
    b) web_user1_pass: "web_user_pass1"
    c) web_user2_pass: "web_user_pass2"
    d) stage_user_pass: "stage_user_pass"
    e) back_user1_pass: "back1"
    f) back_user2_pass: "back2"

Файл vault должен быть зашифрован секретной фразой при помощи ansible-vault. Файл должен быть подгружен вместе с ansible в репозиторий git. Путь укажите в ansible.cfg, расшифрование файла должен происходить без указаний дополнительных параметров и паролей.

6)

Роль

1) Используйте главное правило ansible - идемпотентность!
2) Реализуйте хранение файлов каждый базы данных в отдельном логическом lvm диске
    a) используйте диск /dev/sdb

   b) назовите группу томов “mysql-vg”
   c) логический диск должен создавать по имени базы данных, {{ db_name }}
      i) размер логического диска 10G
      ii) файловая система xfs
   d) логический диск должен монтироваться по пути /var/lib/mysql_{{ db_name }}
      i) владельцем каталога должен быть пользователь mysql
      ii) используйте маску прав доступа 755
3) Установите только необходимый набор пакетов mariadb
   a) mariadb
   b) mariadb_server
   c) mariadb_devel
   d) mariadb_server_utils
   e) версия пакетов 11.4, должна подставляться из переменной-список version, mariadb: 11.4
   f) при запуске плейбука с тегом –install, должны устанавливаться пакеты, указанные из переменной pkg и ничего более
4) Настройте демон mariadb для каждой базы данных в контуре
   a) создайте пользователя mysql
      i) пользователь не должен иметь пароля
      ii) пользователь должен не иметь возможности войти в систему
      iii) введите его в группу mysql
   b) Используйте пути конфигураций
      i) /etc/my.cnf.d/mariadb-server-{{ db_name }}.cnf
      ii) /usr/lib/systemd/system/mariadb_{{ db_name }}.service
      iii) /var/lib/mysql_{{ db_name }}/mysqld.pid
      iv) /var/lib/mysql_{{ db_name }}/mysqld.sock
      v) /var/lib/mysql_{{ db_name }}/data
      vi) /var/lib/mysql_{{ db_name }}/error.log
   c) Примените параметр из файла hosts: wsrep_sync_wait: 1
   d) Используйте в конфигурации mariadb параметр skip-name-resolve
5) Пользователь root в mariadb должен иметь доступ к сокеты базы данных, без пароля
   a) остальные пользователи должны иметь доступ только по паролю
   b) вход по сети пользователю root, должен быть недоступен
6) Создайте базы данных в mariadb из переменных db_name и create_db_name
7) Создайте пользователей в mariadb из переменной mysql_user
   a) привилегии должны применяться из ключа priv

b) при запуске плейбука с –tags users, должны создаваться только пользователи и ничего больше

Проверка

Проверка будет осуществляться на чистом, новом стенде. Спулив репозиторий из git и запустив плейбук mysql.yml. Плейбук может быть запущен в любом формате:

- ansible-playbook -e env=prod playbook/mysql.yml -D
- ansible-playbook -e env=stage playbook/mysql.yml -D -l stage_web_mysql
- ansible-playbook -e env=stage playbook/mysql.yml -D -l stage_web_mysql –tags users

В случае запуска тегов, подразумевается, запуск уже после деплоя всей базы данных.

При повторном запуске, если при написании придерживались принципам идемпотентности, не должно быть changed тасок.

Как завершить задание и запушить репозиторий?

Переходите на https://gitverse.ru
Регистрируетесь/Авторизируетесь
Создаете репозиторий с названием apt-ansible25
Скидываете свой репозиторий в ЛС главному эксперту(пояснения будут в день олимпиады)

При проставлении баллов, будет учитываться логическая связь, например: нельзя получить баллы за пользователя и группу mysql, не установив пакеты mariadb.

Приложение 1

-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAAAMwAAAAtzc2
gtZW
QyNTUxOQAAACCLY9DqR7WnicSPEt+4A/gzVBcEtwzD4hrmP2h+sMYz1wAAAJAXjpGMF
46R
jAAAAAtzc2gtZWQyNTUxOQAAACCLY9DqR7WnicSPEt+4A/gzVBcEtwzD4hrmP2h+sMYz1
w
AAAEDAafejOZ0NyMrS2JFlF6aHX/h0eAXEtnblr4DgdzRN4Itj0OpHtaeJxI8S37gD+DNU
FwS3DMPiGuY/aH6wxjPXAAAACWlseWFAaWx5YQECAwQ=
-----END OPENSSH PRIVATE KEY-----