

Системы управления конфигурациями

Урок 7. Практика на проекте

Создать роль Ansible установки серверов приложения и развернуть Wordpress на целевые ВМ.

Часть 1. Разворачивание веб-сервера.

Устанавливаем зависимости для вэб сервера, роль для деплоя Wordpress у нас уже есть, модифицируем ее позже:

```
ansible-galaxy install geerlingguy.nginx geerlingguy.php
```

Инициализируем роль:

```
ansible-galaxy init deploy_webserver
```

Создадим задачу на создание self-signed сертификата (т.к. для создания проекта этого будет достаточно, в прод среде можно использовать PKI компании или воспользоваться Let's Encrypt PKI для выпуска) **/home/sa/ansible/roles/deploy_webserver/tasks:**

```
---

- name: Create self-signed certificate, if configured.
  command: >
    openssl req -x509 -nodes -subj '/CN={{ vm_name }}' -days 365
    -newkey rsa:4096 -sha256 -keyout "/etc/ssl/private/{{ vm_name }}.key" -out
    "/etc/ssl/certs/{{ vm_name }}.crt"
    creates="/etc/ssl/certs/{{ vm_name }}.crt"
```

Создадим файл с переменными и конфигом nginx

/home/sa/ansible/roles/deploy_webserver/vars/main.yml:

```
---

# vars file for deploy_webserver

server_name: "{{ vm_name }}"
php_default_version_debian: "7.4"
php_version: "7.4"
php_webserver_daemon: "nginx"
php_packages_state: "latest"
php_enable_php_fpm: true
php_executable: "php"
php_memory_limit: "128M"
php_max_execution_time: "90"
php_upload_max_filesize: "64M"
php_date_timezone: "Europe/Moscow"

nginx_vhosts:
- listen: "80"
  server_name: "{{ server_name }} www.{{ server_name }}"
  return: "301 https://{{ server_name }}$request_uri"
```

```

filename: "{{ server_name }}.80.conf"

- listen: "443 ssl http2"
  server_name: "{{ server_name }}"
  root: "/var/www/html"
  index: "index.php index.html index.htm"
  error_page: "500 502 503 504 /50x.html"
  access_log: "/var/log/nginx/{{ server_name }}.access.log main"
  error_log: ""
  state: "present"
  template: "{{ nginx_vhost_template }}"
  filename: "{{ server_name }}.conf"
  extra_parameters: |
    location ~ \.php$ {
        fastcgi_split_path_info ^(.+\.php)(/.+)$;
        fastcgi_pass 127.0.0.1:9000;
        fastcgi_index index.php;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        include fastcgi_params;
    }
    ssl_certificate "/etc/ssl/certs/{{ vm_name }}.crt";
    ssl_certificate_key "/etc/ssl/private/{{ vm_name }}.key";
    ssl_protocols TLSv1.1 TLSv1.2;
    ssl_ciphers HIGH:!aNULL:!MD5;

php_packages:
- php{{php_default_version_debian}}
- php{{php_default_version_debian}}-fpm
- php{{php_default_version_debian}}-cli
- php{{php_default_version_debian}}-common
- php{{php_default_version_debian}}-gd
- php{{php_default_version_debian}}-mbstring
- php{{php_default_version_debian}}-pdo
- php{{php_default_version_debian}}-xml
- php{{php_default_version_debian}}-curl
- php{{php_default_version_debian}}-gd
- php{{php_default_version_debian}}-mbstring
- php{{php_default_version_debian}}-xml
- php{{php_default_version_debian}}-xmlrpc
- php{{php_default_version_debian}}-soap
- php{{php_default_version_debian}}-intl
- php{{php_default_version_debian}}-zip
- php{{php_default_version_debian}}-mysql

```

Создадим роль `/home/sa/ansible/roles/deploy_webserver/deploy_webserver.yml`:

```

---
- hosts: app01
  become: true

  vars_files:
    - vars/main.yml

  roles:
    - role: deploy_webserver
      vars:

```

```

        vm_name: "app01.local"
    - role: geerlingguy.nginx
      vars:
        vm_name: "app01.local"
    - geerlingguy.php

- hosts: app02
  become: true

vars_files:
  - vars/main.yml
roles:
  - role: deploy_webserver
    vars:
      vm_name: "app02.local"
  - role: geerlingguy.nginx
    vars:
      vm_name: "app02.local"
  - geerlingguy.php

```

Запускаем деплой:

```
cd ~/ansible && ansible-playbook roles/deploy_webserver/deploy_webserver.yml
```

```

[sa@ansible ansible]$ cd ~/ansible && ansible-playbook roles/deploy_webserver/deploy_webserver.yml
PLAY [app01] *****
TASK [Gathering Facts] *****
ok: [app01]
TASK [deploy_webserver : Create self-signed certificate, if configured.] *****
ok: [app01]

```

Задача по настройке веб-сервера отработала, замечаний нет:

```

PLAY RECAP *****
app01      : ok=47  changed=7    unreachable=0    failed=0    skipped=19    rescued=0    ignored=0
app02      : ok=48  changed=13   unreachable=0    failed=0    skipped=19    rescued=0    ignored=0

```

Сделаем пару файлов, чтобы потом проверить работу PHP и впоследствии посмотреть как работает HA-Proxy.

```

cat > /var/www/html/serverinfo.php << _EOF_
_EOF_
<?php
echo "<H1>Server name: ";
echo $_SERVER['SERVER_NAME'];
echo "</H1><br>";
echo "<br>";
phpinfo(INFO_GENERAL);
?>
_EOF_

```

Результат:

←

🔒

↻

🔒 app01.local

PHP 7.4.3-4ubuntu2.18 - phpinfo

Server name: app01.local

PHP Version 7.4.3-4ubuntu2.18

System	Linux app01 5.4.0-153-
Build Date	Feb 23 2023 12:43:23
Server API	FPM/FastCGI

←

🔒

↻

🔒 app02.local

PHP 7.4.3-4ubuntu2.18 - phpinfo

Server name: app02.local

PHP Version 7.4.3-4ubuntu2.18

System	Linux app02 5.4.0-51-g
Build Date	Feb 23 2023 12:43:23
Server API	FPM/FastCGI

Добавим эту роль в деплой-плейбук проекта `/home/sa/ansible/playbooks/main.yml`:

```
---
- name: deploy_vm_ovirt
  import_playbook: /home/sa/ansible/roles/deploy_vm_ovirt/deploy_vm_ovirt.yml

- name: deploy_mysql
  import_playbook: /home/sa/ansible/roles/deploy_mysql/deploy_mysql.yml

- name: deploy_webserver
  import_playbook: /home/sa/ansible/roles/deploy_webserver/deploy_webserver.yml
```

Часть 2. Разворачивание CMS - Wordpress

Модифицируем нашу роль по разливки Wordpress, созданную ранее.

`/home/sa/ansible/roles/wordpress/tasks/main.yml`:

```
---
# tasks file for wordpress

# WordPress Configuration

- name: Download and unpack latest wordpress
  unarchive:
    src: https://wordpress.org/latest.tar.gz
    dest: "/var/www/html"
    remote_src: yes
    creates: "/var/www/html"
```

```

- name: Set ownership
  file:
    path: "/var/www/html"
    state: directory
    recurse: yes
    owner: www-data
    group: www-data

- name: Set permissions for directories
  shell: "/usr/bin/find /var/www/html -type d -exec chmod 750 {} \\";"

- name: Set permissions for files
  shell: "/usr/bin/find /var/www/html -type f -exec chmod 640 {} \\";"

- name: Set up wp-config
  template:
    src: "wp-config.php.j2"
    dest: "/var/www/html/wp-config.php"

```

/home/sa/ansible/roles/wordpress/templates/wp-config.php.j2:

```

<?php
/** The name of the database for WordPress */
define( 'DB_NAME', '{{ mysql_db }}' );
/** MySQL database username */
define( 'DB_USER', '{{ mysql_user }}' );
/** MySQL database password */
define( 'DB_PASSWORD', '{{ mysql_password }}' );
/** MySQL hostname */
define( 'DB_HOST', '{{ mysql_server }}' );
/** Database Charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );
/** The Database Collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );
/** Filesystem access */
define( 'FS_METHOD', 'direct' );

define( 'AUTH_KEY', '{{ lookup('password', '/dev/null chars=ascii_letters
length=64') }}' );
define( 'SECURE_AUTH_KEY', '{{ lookup('password', '/dev/null chars=ascii_letters
length=64') }}' );
define( 'LOGGED_IN_KEY', '{{ lookup('password', '/dev/null chars=ascii_letters
length=64') }}' );
define( 'NONCE_KEY', '{{ lookup('password', '/dev/null chars=ascii_letters
length=64') }}' );
define( 'AUTH_SALT', '{{ lookup('password', '/dev/null chars=ascii_letters
length=64') }}' );
define( 'SECURE_AUTH_SALT', '{{ lookup('password', '/dev/null chars=ascii_letters
length=64') }}' );
define( 'LOGGED_IN_SALT', '{{ lookup('password', '/dev/null chars=ascii_letters
length=64') }}' );
define( 'NONCE_SALT', '{{ lookup('password', '/dev/null chars=ascii_letters
length=64') }}' );

```

```

$table_prefix = 'wp_';

define( 'WP_DEBUG', false );

if ( ! defined( 'ABSPATH' ) ) {
define( 'ABSPATH', dirname( __FILE__ ) . '/' );
}

/** Sets up WordPress vars and included files. */
require_once( ABSPATH . 'wp-settings.php' );

```

/home/sa/ansible/roles/wordpress/vars/wordpress_vault.yml

```

---
mysql_password: "geek_pwd"

```

/home/sa/ansible/roles/wordpress/vars/wordpress.yml

```

---
mysql_db: "wp_geek"
mysql_user: "wp_geek_user"
mysql_server: "mysqlb.local"

```

/home/sa/ansible/roles/wordpress/wordpress.yml:

```

---
- hosts: app01, app02
  become: true

  vars_files:
    - vars/wordpress.yml
    - vars/wordpress_vault.yml
  roles:
    - role: wordpress

```

Запускаем деплой:

```

cd ~/ansible && ansible-playbook roles/wordpress/wordpress.yml --vault-password-
file ./vault.pass

```

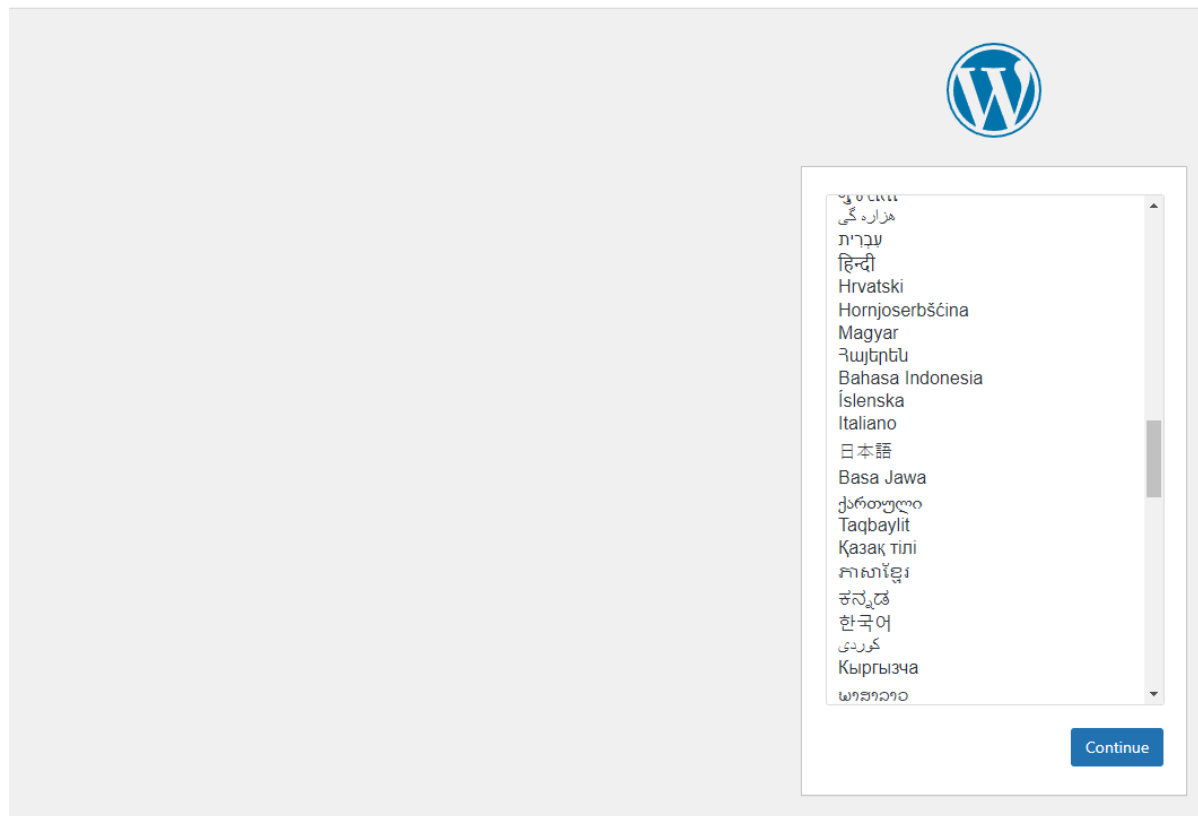
Task выполнен успешно:

```

PLAY RECAP *****
app01                : ok=6    changed=5    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
app02                : ok=6    changed=5    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

```

Видим мастер на одной из нод:



После установки пароля проверяем первую ноду - полет нормальный:

app.local

Mindblown: a blog about philosophy.

Привет, мир!

Добро пожаловать в WordPress. Это ваша первая запись. Отредактируйте или удалите ее, затем начинайте создавать!

29.06.2023

Проверяем вторую ноду - полет нормальный

app.local

Mindblown: a blog about philosophy.

Привет, мир!

Добро пожаловать в WordPress. Это ваша первая запись. Отредактируйте или удалите ее, затем начинайте создавать!

29.06.2023

Добавляем эту роль в наш проект:

Добавим эту роль в деплой-плейбук проекта **/home/sa/ansible/playbooks/main.yml**:

```
---
- name: deploy_vm_ovirt
  import_playbook: /home/sa/ansible/roles/deploy_vm_ovirt/deploy_vm_ovirt.yml

- name: deploy_mysql
  import_playbook: /home/sa/ansible/roles/deploy_mysql/deploy_mysql.yml

- name: deploy_webserver
  import_playbook: /home/sa/ansible/roles/deploy_webserver/deploy_webserver.yml

- name: deploy_wordpress
  import_playbook: /home/sa/ansible/roles/wordpress/wordpress.yml
```