Урок 3. Используем Ansible на практике (base)

Установка приложения MySQL + WordPress + FPM + nginx с помощью плейбука Ansible.
* (по желанию, но рекомендую) Доработать проект чтобы получить сразу работающее приложение Wordpress.

Настраиваем каталог проекта. В репозитории git - ветка lesson03, папка Project.
Ранее была выполнена настройка хоста geekbrains (Debian 12) на авторизацию по SSH с помощью сертификата и повышение прав пользователя Server Administrator (sa) с помощью утилиты sudo без пароля.

На ansible хосте:

```
ssh-keygen -t rsa -C "sa@ansible.local"
ssh-copy-id sa@geekbrains.local
```

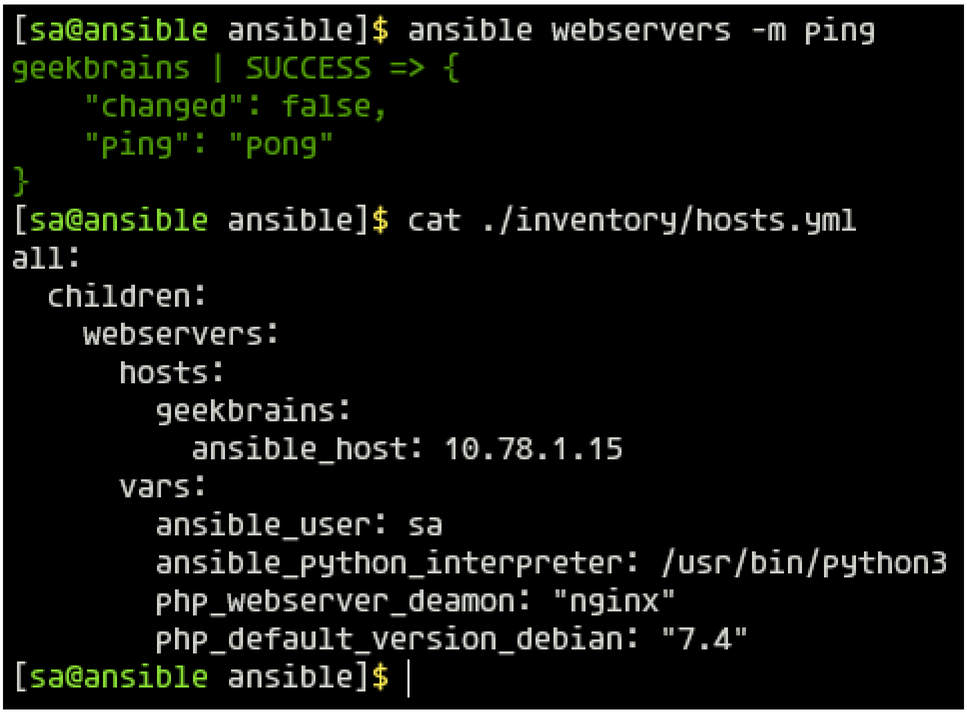
На geekbrains хосте:

```
cat > /etc/sudoers.d/sa_without_pwd << _EOF_
sa ALL=(ALL) NOPASSWD: ALL
_EOF_
```

1. Обогащаем фаил hosts.yml (и прорабатываем информационные сообщения из предыдущего ДЗ - указываем путь к интерпритатору)

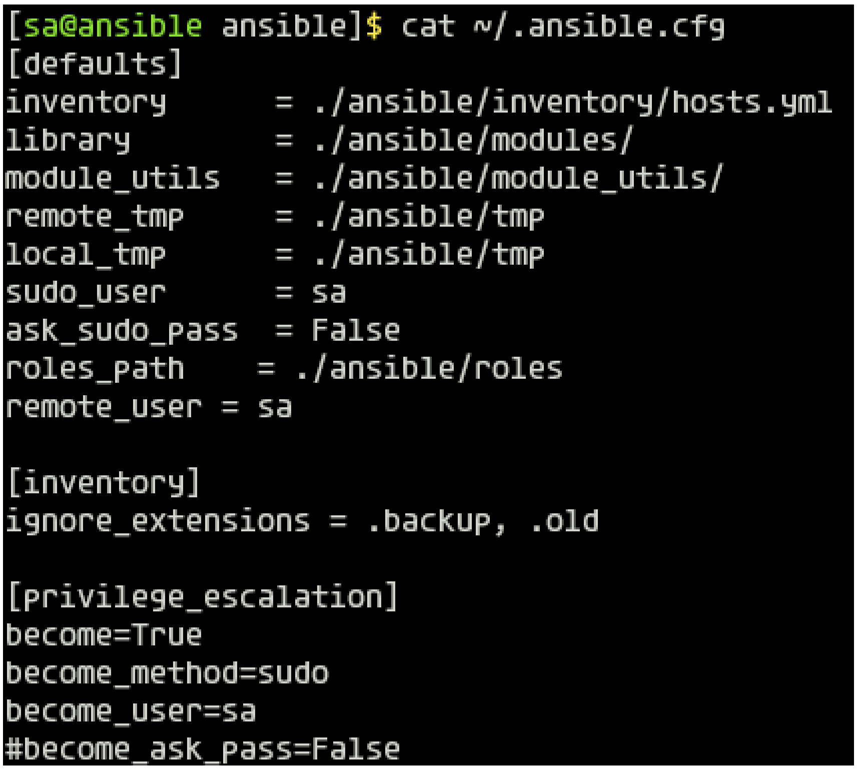
   ```
   all:
     children:
       webservers:
         hosts:
           geekbrains:
             ansible_host: 10.78.1.14
         vars:
           ansible_user: sa
           ansible_python_interpreter: /usr/bin/python
           php_webserver_deamon: "nginx"
   ```

Проверяем:



```
[sa@ansible ansible]$ ansible webservers -m ping
geekbrains | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
[sa@ansible ansible]$ cat ./inventory/hosts.yml
all:
  children:
    webservers:
      hosts:
        geekbrains:
          ansible_host: 10.78.1.15
      vars:
        ansible_user: sa
        ansible_python_interpreter: /usr/bin/python3
        php_webserver_deamon: "nginx"
        php_default_version_debian: "7.4"
[sa@ansible ansible]$ 
```

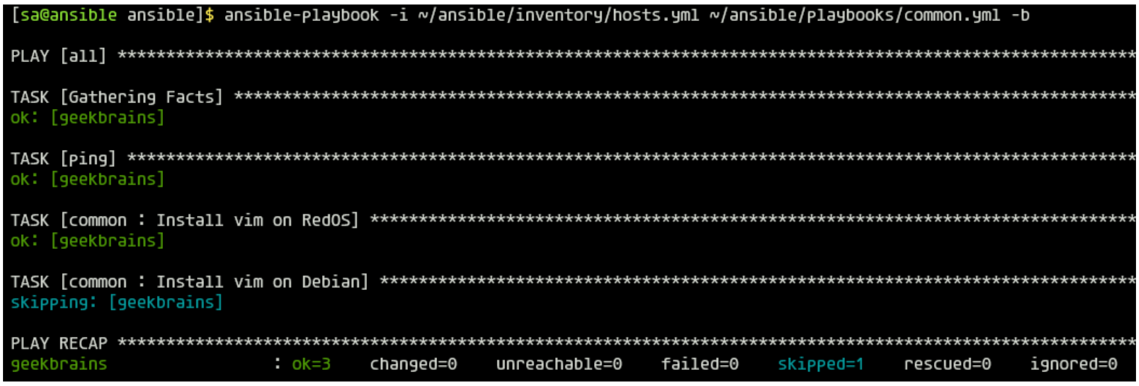
2. Правим фаил-конфигурацию проекта:



```
[sa@ansible ansible]$ cat ~/.ansible.cfg
[defaults]
inventory       = ./ansible/inventory/hosts.yml
library         = ./ansible/modules/
module_utils    = ./ansible/module_utils/
remote_tmp      = ./ansible/tmp
local_tmp       = ./ansible/tmp
sudo_user       = sa
ask_sudo_pass   = False
roles_path     = ./ansible/roles
remote_user = sa

[inventory]
ignore_extensions = .backup, .old

[privilege_escalation]
become=True
become_method=sudo
become_user=sa
#become_ask_pass=False
```

Остальные параметры оставляем по-умолчанию.

Проверяем корректность внесенных параметров ролью из лекции:



```
[sa@ansible ansible]$ ansible-playbook -i ~/ansible/inventory/hosts.yml ~/ansible/playbooks/common.yml -b

PLAY [all] ************************************************************************************************************

TASK [Gathering Facts] ************************************************************************************************
ok: [geekbrains]

TASK [ping] ***********************************************************************************************************
ok: [geekbrains]

TASK [common : Install vim on RedOS] **********************************************************************************
ok: [geekbrains]

TASK [common : Install vim on Debian] *********************************************************************************
skipping: [geekbrains]

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

3. Инициализируем роли из Ansible Galaxy:

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

4. Создаем роль Wordpress.

   ~/ansible/roles/wordpress/wordpress.yml:

```
---
- hosts: all

  vars_files:
    - vars/wordpress.yml
  roles:
    - geerlingguy.nginx
    - geerlingguy.php
    - geerlingguy.mysql
    - wordpress
```

   ~/ansible/roles/wordpress/tasks/main.yml:

```
---
# tasks file for wordpress

# WordPress Configuration
- name: create /var/www/{{ server_name }} directory for unarchiving
  file:
    path: /var/www/{{ server_name }}
    state: directory

- name: Download and unpack latest WordPress
  unarchive:
    src: https://wordpress.org/latest.tar.gz
    dest: "/var/www/{{ server_name }}"
    remote_src: yes
    creates: "/var/www/{{ server_name }}/wordpress"

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

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

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

- name: Set up wp-config
  template:
    src: "wp-config.php.j2"
    dest: "/var/www/{{ server_name }}/wordpress/wp-config.php"
```

~/ansible/roles/wordpress/vars/wordpress.yml:

```
---
server_name: "geekbrains.local" # тут ваш домен
mysql_db: "wordpress"
mysql_user: "wordpress"
mysql_password: "WordpreSS0!"

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/{{ server_name}}/wordpress"
    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 unix:/var/run/php/php-fpm.sock;
          fastcgi_index index.php;
          fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
          include fastcgi_params;
      }
      ssl_certificate /etc/letsencrypt/live/{{ server_name }}/fullchain.pem;
      ssl_certificate_key /etc/letsencrypt/live/{{ server_name
}}/privkey.pem;
      ssl_protocols TLSv1.1 TLSv1.2;
      ssl_ciphers HIGH:!aNULL:!MD5;


php_memory_limit: "128M"
php_max_execution_time: "90"
php_upload_max_filesize: "256M"
php_webserver_daemon: "nginx"
php_packages:
  - php
  - php-cli
  - php-common
  - php-gd
  - php-mbstring
  - php-pdo
  - php-xml
  - php-curl
  - php-gd
  - php-mbstring
  - php-xml
  - php-xmlrpc
  - php-soap
  - php-intl
  - php-zip
  - php-mysql

mysql_databases:
  - name: "{{ mysql_user }}"
    host: "%"
    password: "{{ mysql_password }}"
    priv: "wordpress.*:ALL"
```

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

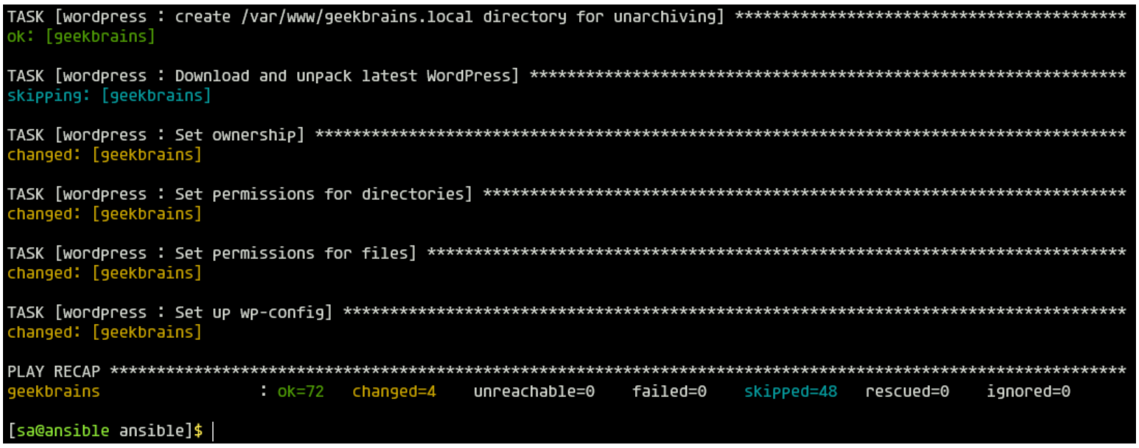
```
<?php
/** The name of the database for WordPress */
define( 'DB_NAME', '{{ mysql_db }}' );
/** MySQL database username */
define( 'DB_USER', '{{ mysql_user }}' );
/** MySQL database password */
define( 'DB_PASSWORD', '{{ mysql_password }}' );
/** MySQL hostname */
define( 'DB_HOST', 'localhost' );
/** 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' );
```

5. Запускаем деплой:

```
cd ~/ansible && ansible-playbook -i inventory/hosts.yml
roles/wordpress/wordpress.yml -b
```

Результат



```
TASK [wordpress : create /var/www/geekbrains.local directory for unarchiving] *******************************************
ok: [geekbrains]

TASK [wordpress : Download and unpack latest WordPress] *****************************************************************
skipping: [geekbrains]

TASK [wordpress : Set ownership] ****************************************************************************************
changed: [geekbrains]

TASK [wordpress : Set permissions for directories] **********************************************************************
changed: [geekbrains]

TASK [wordpress : Set permissions for files] ****************************************************************************
changed: [geekbrains]

TASK [wordpress : Set up wp-config] *************************************************************************************
changed: [geekbrains]

PLAY RECAP **************************************************************************************************************
geekbrains                 : ok=72   changed=4    unreachable=0    failed=0    skipped=48   rescued=0    ignored=0

[sa@ansible ansible]$ |
```