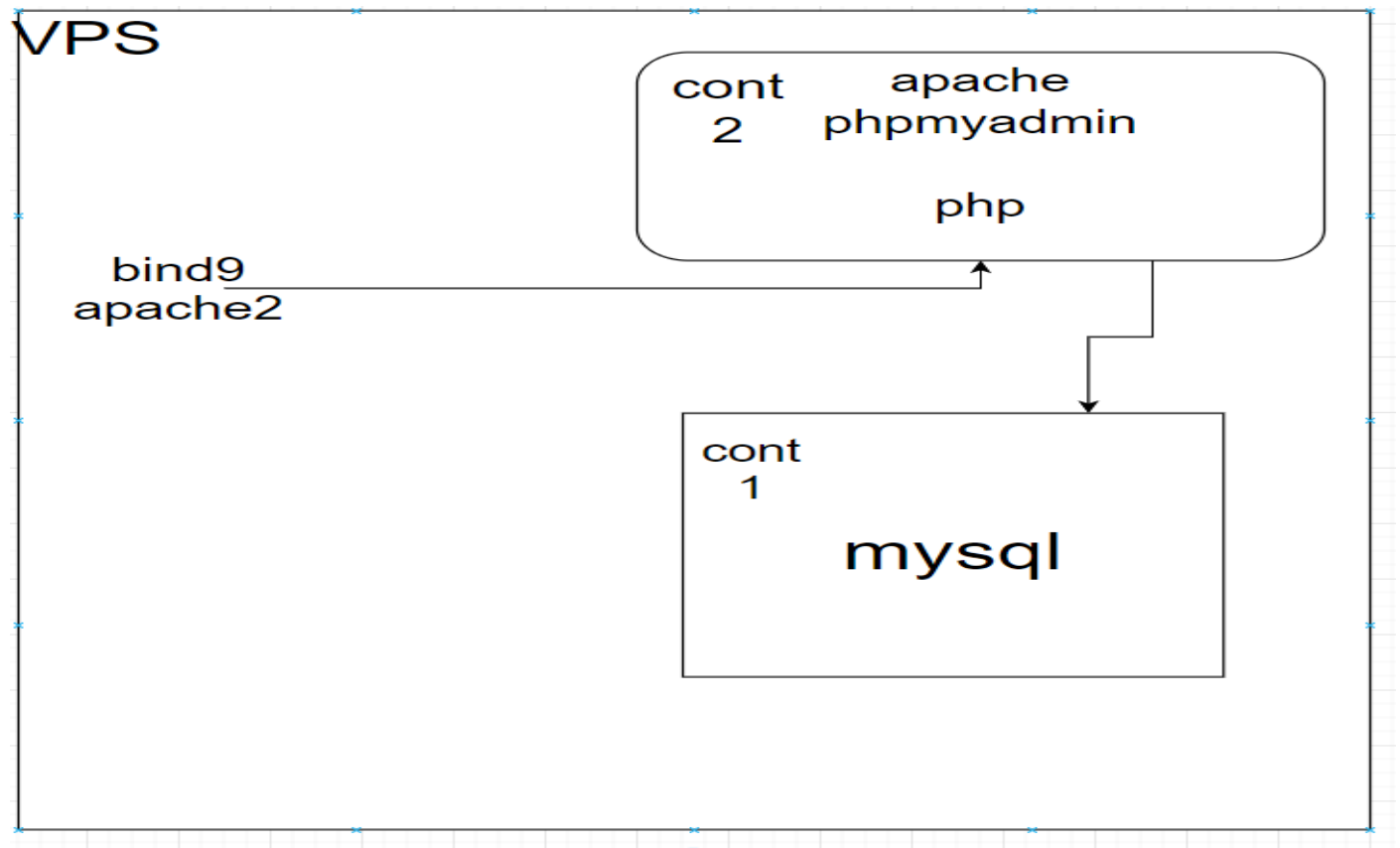


Automatización con Ansible

El proyecto constara de dos contenedores Docker que cumplirán distintas funciones, la finalidad de este proyecto es crear u entorno de producción de una aplicación web.

Diseño general del proyecto:



Podemos separarla en tres ramas globales:

- La VPS actuara como dns (`bind9`) interno y proxy inverso (`Apache`) para mantener nuestro contenedor de publicación seguro.
- El segundo contenedor tendrá dispondrá de `apache`, `php`, `phpmyadmin` y `laravel`
- El tercer contenedor solo ejecutara la base de datos (`mysql`).

Para automatizar la generación del proyecto completo crearemos un playbook que maneja todos los playbook y roles, llamado `play_global.yml`

play_global.yml U X

proyecto_git > proyectomorg > PAnsible > play_global.yml > 3 > [] roles

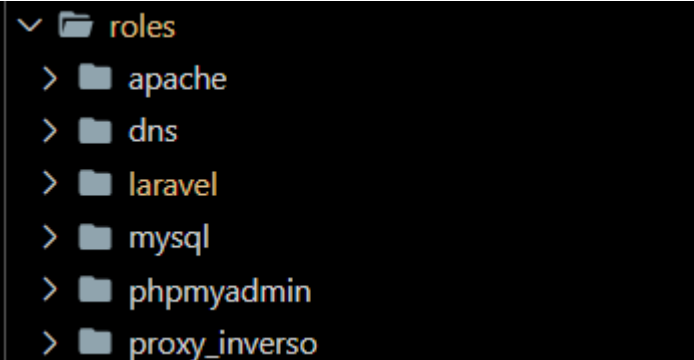
```
1  ---
2  - name: Ejecución de tarea de git
3    import_playbook: play_git.yml
4
5  - name: configurar bind9 como dns
6    hosts: controlador
7    roles:
8      - dns
9
10 - name: configurar apache2 como proxy
11   hosts: controlador
12   become: true
13   roles:
14     - proxy_inverso
15
16 - name: Instalar y configurar apache2.yml
17   hosts: webservers
18   remote_user: tienda juegos
19   become: true
20   roles:
21     - apache
22     - phpmyadmin
23     - laravel
24
25 - name: Bases de datos mysql servers
26   hosts: sgbd
27   become: true
28   roles:
29     - mysql
30
31
32 ...
```

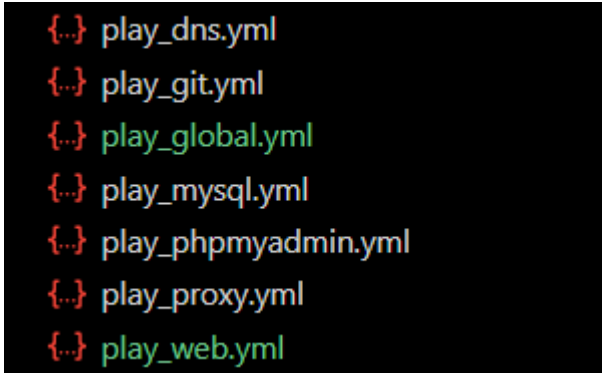
Para dar contexto al proyecto cree un inventario para realizar distintos test

```
{...} play_global.yml U    inventory_test X
proyecto_git > proyectomorg > PAnsible > inventory_test
1  [servidores]
2  localhost ansible_connection=local
3  172.18.20.2 host_name=apache2
4  172.18.20.3 host_name=mysql2
5
6  [controlador]
7  localhost
8
9  [sgbd]
10 172.18.20.3
11
12 [webservers]
13 172.18.20.2
14
15 [tiendajuegos:children]
16 sgbd
17 webserver
```

Definimos cinco grupos donde añadiremos nuestros servidores (sobrenombres) que utilizaremos a la hora de realizar una ejecución a un grupo en concreto.

Para realizar pruebas unitarias cree roles únicos para cada servicio a implementar.

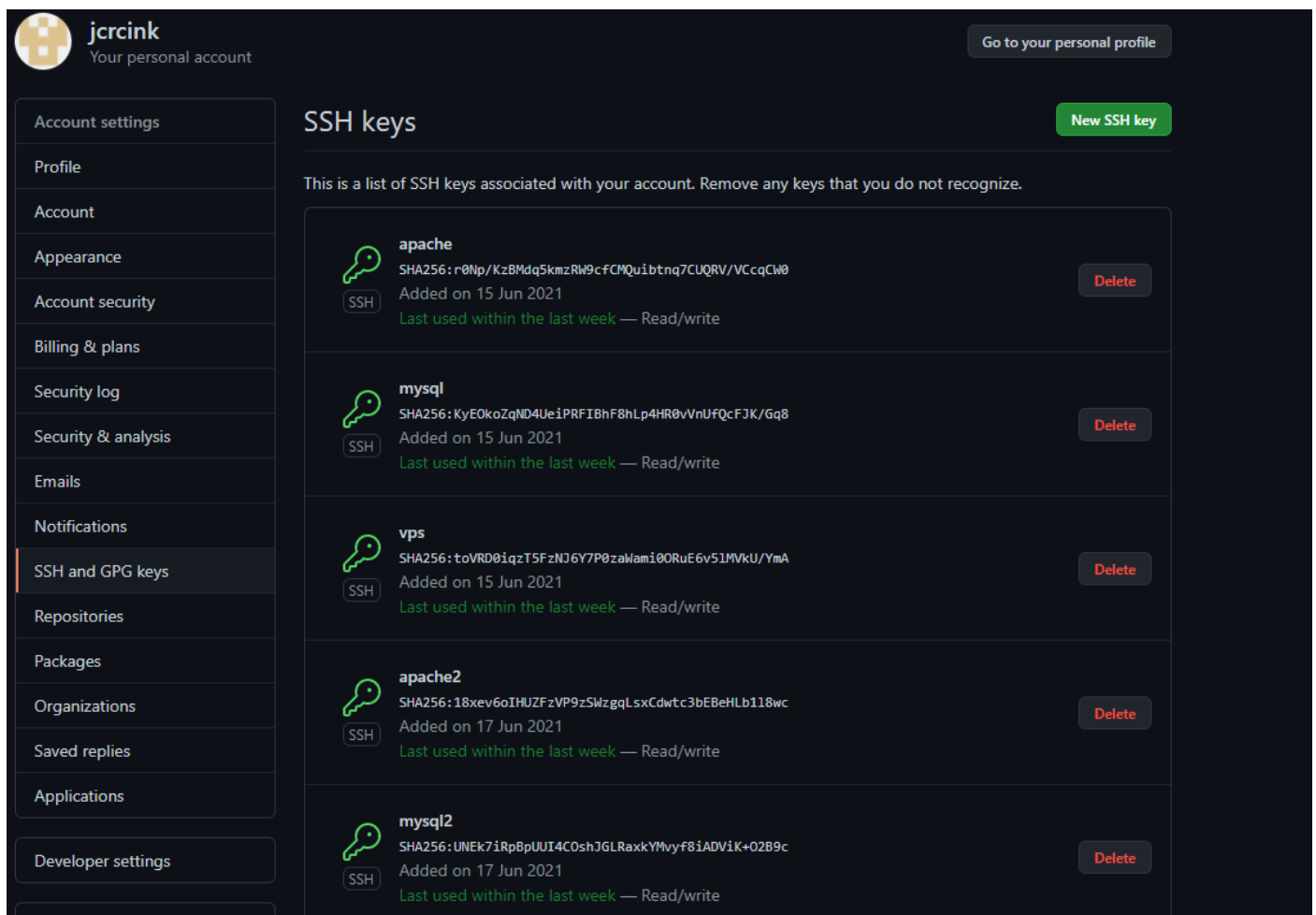
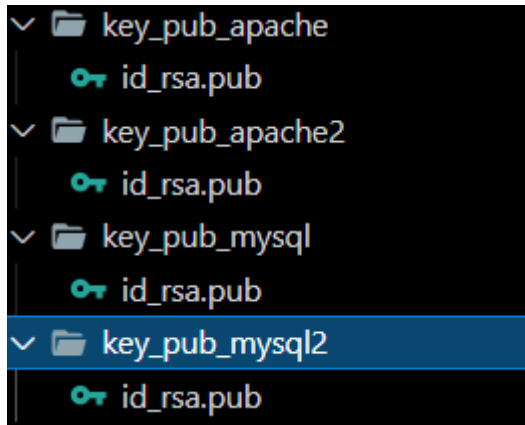




Como nuestro entorno de trabajo va a estar unido a github decidimos crear llaves rsa para otorgar mayor privacidad y seguridad a nuestras conexiones con los contenedores.

Para realizar esta acción creamos el playbook `play_git.yml` donde utilizamos el modulo de ansible llamado `users` que nos creara las llaves publicas/privada de nuestro usuario, en este caso `tiendajuegos`.

Cada clave publica se recuperará en el nodo controlador y posteriormente se añadirá a la configuración de github.



```
17
18 # #Generar clave ssh de usuario tiendajuegos
19 - name: Crear SSH Key para diploy github
20   user:
21     name: tiendajuegos
22     generate_ssh_key: yes
23     ssh_key_bits: 4096
24     ssh_key_type: rsa
25     ssh_key_file: .ssh/id_rsa
26     ssh_key_comment: jcrcink@gmail.com
27     state: present
28
```

```
##Uso del Agente SSH para gurdar las claves
- name: Iniciar ssh-agent
  become_user: tiendajuegos
  # shell: eval $(ssh-agent)
  shell: eval $(ssh-agent -s) && ssh-add ~/.ssh/id_rsa

## Copiar llave publica al nodo controlador
- name: copiar llave publica de git
  fetch:
    src: /home/tiendajuegos/.ssh/id_rsa.pub
    dest: "/home/johanvps/proyecto_git/proyectomorg/PAnsible/key_pub_{{ item }}/"
    flat: yes
  loop:
    - apache2
    - mysql2
```

Con estos pasos nuestros contenedores están conectados a github.

Nuestra VPS se encarga de dar nombres DNS a los contenedores con el rol siguiente:

main.yml x

vector_git > proyectomorg > PAnsible > roles > dns > tasks > {--} main.yml > ...

```
2  - name: Instalar bind9
3    apt:
4      name: "{{ item }}"
5      state: latest
6    loop: "{{ bind9_packages }}"
7    notify: restart_bind9
8
9  - name: hostname fact
10   set_fact:
11     ansible_fqdn: dns.{{ domain }}
12
13  - name: copiar named.conf.local file
14    template:
15      src: named.conf.local.j2
16      dest: /etc/bind/named.conf.local
17      owner: root
18      group: bind
19      mode: 0640
20    notify: restart_bind9
21
22  - name: crear directorio de zonas
23    file:
24      path: /etc/bind/zones
25      state: directory
26      owner: root
27      group: bind
28      mode: 0750
29
30  - name: Copy forward file
31    template:
32      src: db.{{ domain }}.j2
33      dest: "/etc/bind/zones/db.{{ domain }}"
34      owner: root
35      group: bind
36      mode: 0640
37    loop: "{{ records.items() | list }}"
38    notify: restart_bind9
39
40  - name: Copy reverse file
41    template:
42      src: db.reverse.j2
43      dest: "/etc/bind/zones/db.{{ rev_domain }}"
44      owner: root
45      group: bind
46      mode: 0640
47    loop: "{{ records.items() | list }}"
48    notify: restart_bind9
49
```

Y las siguientes variables podremos automatizar con mayor facilidad el servicio.

```
---
bind9_packages:
  - bind9
  - bind9utils
  - bind9-doc

domain: project20.local
rev_domain: 20.18.172
host: 172.18.20

records:
  dns:
    forward: 127.0.0.1
    type: A
    typeR: PTR
    last: 1
    rev: "dns.{{domain}}."
  apache:
    forward: "{{ host }}.2.{{ domain }}"
    type: A
    typeR: PTR
    last: 2
    rev: "apache.{{domain}}."
  mysql:
    forward: "{{ host }}.3.{{ domain }}"
    type: A
    typeR: PTR
    last: 3
    rev: "mysql.{{domain}}."
```

También haremos uso de templates para la creación de zonas DNS.

```
projecto_git > projectomorg > PAnsible > roles > dns > templates > dbproject20.localj2
1 ;$TTL 604800 ; 1 week
2 ;$ORIGIN {{ domain }}.
3 @ IN SOA {{ ansible_fqdn }}. root.{{ domain }}. (
4 18 ; serial
5 604800 ; refresh (1 week)
6 86400 ; retry (1 day)
7 2419200 ; expire (4 weeks)
8 604800 ; minimum (1 week)
9 )
10 @ IN NS {{ ansible_fqdn }}.
11 {% for item, value in records.items() %}
12 {{ item }} IN {{ value.type }} {{ value.forward }}
13 {% endfor %}

projecto_git > projectomorg > PAnsible > roles > dns > templates > dbreversej2
1 ;$TTL 604800 ; 1 week
2 ;$ORIGIN {{ rev_domain }}.
3 @ IN SOA {{ ansible_fqdn }}. root.{{ domain }}. (
4 18 ; serial
5 604800 ; refresh (1 week)
6 86400 ; retry (1 day)
7 2419200 ; expire (4 weeks)
8 604800 ; minimum (1 week)
9 )
10 IN NS {{ ansible_fqdn }}.
11 {% for item, value in records.items() %}
12 {{ value.last }} IN {{ value.typeR }} {{ value.rev }}
13 {% endfor %}

projecto_git > projectomorg > PAnsible > roles > dns > templates > named.conf.localj2
1 zone "{{ domain }}" {
2 type master;
3 file "/etc/bind/zones/db.{{ domain }}";
4 };
5
6 zone "{{ rev_domain }}.in-addr.arpa" {
7 type master;
8 file "/etc/bind/zones/db.{{ rev_domain }}";
9 };
10
```

También definimos un handler para realizar el restart del servicio bind9

```
main.yml x
proyecto_git > proyectomorg > PAnsible > roles > proxy_inverso >
1 ---
2 - name: apache restart
3   service:
4     name: apache2
5     state: restarted
6 ...
```

Proxy inverso

```
main.yml x
proyecto_git > proyectomorg > PAnsible > roles > proxy_inverso > tasks > {..} main.yml > {} 1 > notify
2 ##Desabilitamos los virtualhost de apache2
3 - name: Deshabilitar virtual host por defecto
4   file:
5     path: /etc/apache2/sites-enabled/000-default.conf
6     state: absent
7   notify: apache restart
8
9 #Activación de módulos para proxy
10 - name: Activar módulos para proxy
11   apache2_module:
12     name: "{{ item }}"
13     state: present
14   loop: "{{ apache_mods_enabled }}"
15   notify: apache restart
16
17 - name: Añadir configuración apache vhost
18   template:
19     src: "{{ apache_vhosts_template }}"
20     dest: "/etc/apache2/sites-available/{{ apache_vhosts_filename }}"
21     owner: root
22     group: root
23     mode: 0644
24   notify: apache restart
25   when: apache_create_vhosts
26
27 #Tarea que sustituye a a2ensite
28 - name: Añadir symlink en sites-enabled
29   file:
30     src: "/etc/apache2/sites-available/{{ apache_vhosts_filename }}"
31     dest: "/etc/apache2/sites-enabled/{{ apache_vhosts_filename }}"
32     state: link
33   notify: apache restart
34   when: apache_create_vhosts
35
36 ##Configurar /etc/apache2/apache2.conf
37 ##Permitir metodo POST
38 - name: Insert SetEnvIf Authorization apache2.conf
39   blockinfile:
40     name: /etc/apache2/apache2.conf
41     block: |
42       SetEnvIf Authorization "(.*)" HTTP_AUTHORIZATION=$1
43     marker: "# {mark} ANSIBLE MANAGED BLOCK manager and managed nodes"
44     backup: yes
45     state: present
46   notify: apache restart
47
```


Creamos templates para el virtualhost del proxy

```
TF vhosts.conf.j2 X
proyecto_git > proyectomorg > PAnsible > roles > proxy_inverso > templates > TF vhosts.conf.j2
1  {% for vhost in apache_vhosts %}
2  <VirtualHost {{ apache_listen_ip }}:{{ apache_listen_port }}>
3      SetEnvIf Authorization "(.*)" HTTP_AUTHORIZATION=$1
4      ServerName {{ vhost.servername }}
5      ServerAlias www.{{ vhost.servername }}
6  {% if ip_server is defined %}
7      ProxyPass / http://{{ ip_server }}/
8      ProxyPassReverse / http://{{ ip_server }}/
9  {% endif %}
10     ProxyPreserveHost on
11 </VirtualHost>
12
13 {% endfor %}
```

Variables:

```
{-} main.yml X
proyecto_git > proyectomorg > PAnsible > roles > proxy_inverso > defaults > {-} main.yml > ...
1  ---
2  apache_mods_enabled:
3      - alias
4      - rewrite
5      - proxy
6      - proxy_html
7      - proxy_http
8
9  # Variables de configuración apache2
10 apache_listen_ip: "*"
11 apache_listen_port: 80
12 apache_create_vhosts: true
13 apache_vhosts_filename: "vhosts.conf"
14 apache_vhosts_template: "vhosts.conf.j2"
15
16
17 apache_vhosts:
18     - servername: "jcproject.es"
19 ip_server: "apache"
20
21 #apache_allow_override: "All"
22 #apache_options: "-Indexes +FollowSymLinks"
```

Servicio de mysql

```
---
- name: Install mysql packages
  apt:
    name: "{{ item }}"
    state: latest
    update_cache: yes
    loop: "{{ mysql_packages }}"
    notify: restart_mysql

- name: Copiar template my.cnf
  template:
    src: my.cnf.j2
    dest: "{{ mysql_conf_dir }}/my.cnf"
    notify: restart_mysql

- name: Start the mysql services
  service:
    name: "{{ mysql_service }}"
    state: started
    enabled: yes

#Permite mayor compatibilidad con phpmyadmin
- name: Cambiar plugin de mysql
  shell: mysql -u root -e 'UPDATE mysql.user SET plugin="mysql_native_password" WHERE user="root" AND host="localhost"'

- name: aplicar FLUSH
  shell: mysql -u root -e 'FLUSH PRIVILEGES'

- name: update mysql root password for all root accounts
  mysql_user:
    name: 'root'
    login_user: 'root'
    login_password: ''
    login_host: 'localhost'
    #host_all: yes # "{{ item }}"
    password: '{{ mysql_root_db_pass }}'
    state: present
```

Instalamos el paquete de mysql y realizamos un update del sistema y notificamos al handler de reiniciar el servicio.

Dato importante es modificar el plugin de mysql_native_password para activar mayor compatibilidad con phpmyadmin.

```
#Crear directorio de bbdd
- name: Directorio bbdd
  file:
    path: /home/tiendajuegos/bbdd
    state: directory
    mode: '0755'
    owner: tiendajuegos
    group: tiendajuegos

# Descargar brach db donde esta el .sql del proyecto
- name: recuperrar de rama de git
  remote_user: tiendajuegos
  git:
    repo: 'git@github.com:nicolas1099/proyectomorg.git'
    dest: /home/tiendajuegos/bbdd
    key_file: /home/tiendajuegos/.ssh/id_rsa
    #accept_hostkey: yes
    version: db

#Importar .sql a BBDD
- name: Importar .sql
  mysql_db:
    name: "{{ item.name }}"
    state: import
    target: /home/tiendajuegos/bbdd/proyecto.sql
  loop: "{{ mysql_db }}"
```

Hacemos un gitpull de nuestro github de la rama db donde se almacena el .sql que importaremos a la BBDD tiendajuegos.

Creamos distintas variables

```
main.yml M my.cnf2 X
proyecto_git > proyectomorg > PAnsible > roles > mysql > templates > my.cnf2
1 [[client]]
2 user=root
3 password={{ mysql_root_db_pass }}
```

```
my.cnf2 X
proyecto_git > proyectomorg > PAnsible > roles > mysql > templates > my.cnf2
16 # * Basic Settings
17 #
18 user = mysql
19 pid-file = /var/run/mysqld/mysqld.pid
20 socket = /var/run/mysqld/mysqld.sock
21 port = {{ mysql_port }}
22 datadir = /var/lib/mysql
23 default-authentication-plugin=mysql_native_password
24
25 # If MySQL is running as a replication slave, this should be
26 # changed. Ref https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.ht
27 # tmpdir = /tmp
28 #
29 # Instead of skip-networking the default is now to listen only on
30 # localhost which is more compatible and is not less secure.
31 bind-address = {{ mysql_bind_address }}
32 #mysqlx-bind-address = 127.0.0.1
33 #
```

Y la definimos en el directorio default

```
main.yml X
proyecto_git > proyectomorg > PAnsible > roles > mysql > defaults > {
1 ---
2 mysql_packages:
3   - python3-apt
4   - apt
5   #- python3-selinux
6   - mysql-server
7   - python3-pymysql
8   #- phpmyadmin
9
10  mysql_service: mysql
11  | | | | | #mysqld
12
13  mysql_conf_dir: "/etc/mysql"
14
15  mysql_port: 3306
16  mysql_bind_address: "0.0.0.0"
17  mysql_root_db_pass: root
18
19  mysql_db:
20  | - name: tiendajuegos
21
22  mysql_users:
23  | | name: tiendajuegos
24  | | pass: tiendajuegos
25  | | priv: " *.*:ALL"
26  | | state: present
27
```

En el repositorio se encuentran los demás roles:

<https://github.com/nicolas1099/proyectomorg/tree/johan-ansible>