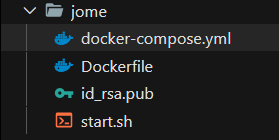
**ANSIBLE - MONSTAFTP + APACHE2 + MYSQL**

La idea de nuestro proyecto consiste en levantar un servicio de FTP conectado a una base de datos y apache2.

La base de datos se llamará ftp y contendrá un usuario, jome.

**Contenedores Docker 🡪 FTP + MYSQL**

****

-Se utilizará un Dockerfile con la instalación mínima para ambos contenedores.

FROM ubuntu

RUN apt update && apt install -yq --no-install-recommends \

    apt-utils \

    wget \

    nano \

    net-tools \

    sudo \

    curl \

    unzip \

    openssh-server \

    openssh-client \

    dos2unix \

    python3\

    software-properties-common

#indicamos el timezone

ARG TZ=Europe/Madrid

ENV TZ=${TZ}

RUN ln -snf /usr/share/zoneinfo/$TZ /etc/localtime && echo $TZ > /etc/timezone

EXPOSE 22

COPY ./id\_rsa.pub /root

COPY ./start.sh /root

RUN dos2unix /root/start.sh

RUN chmod +x /root/start.sh

ENTRYPOINT [ "/root/start.sh" ]

**start.sh**

#!/bin/bash

set -e

newUser(){

    # ---------------- creación de usuario

    echo "MAQ-->usuariod-->${USUARIO}" > /root/datos.txt

    useradd -rm -d /home/"${USUARIO}" -s /bin/bash "${USUARIO}"

    echo "root:${PASSWD}" | chpasswd

    echo "${USUARIO}:${PASSWD}" | chpasswd

}

config\_Sudoers(){

    echo "${USUARIO} ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers

}

config\_ssh(){

    sed -i 's/#PermitRootLogin prohibit-password/PermitRootLogin yes/' /etc/ssh/sshd\_config

    sed -i 's/#Port 22/Port 22/' /etc/ssh/sshd\_config

    if [ ! -d /root/.ssh ]

    then

        mkdir /root/.ssh

        cat /root/id\_rsa.pub >> /root/.ssh/authorized\_keys

    fi

    /etc/init.d/ssh start

}

main(){

    newUser

    config\_Sudoers

    config\_ssh

}

main

tail -f /dev/null

**docker-compose.yml**

version: '3.8'

services:

  ftp:

    container\_name: ftp

    build:

      context: .

      dockerfile: Dockerfile

    environment:

      - USUARIO=jome

      - PASSWD=jome

    ports:

      - "8080:80"

      - "20:20"

      - "21:21"

    networks:

      riot:

          ipv4\_address: 180.170.160.10

  mysql:

    container\_name: mysql

    build:

      context: .

      dockerfile: Dockerfile

    environment:

      - USUARIO=jome

      - PASSWD=jome

    networks:

      riot:

          ipv4\_address: 180.170.160.20

networks:

  riot:

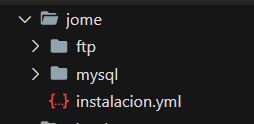
      ipam:

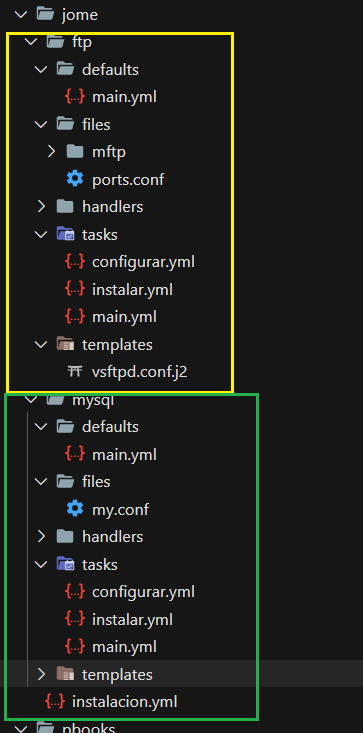
          driver: default

          config:

              - subnet: 180.170.160.0/24

**PLAYBOOK - Estructura**





**defaults 🡪** se indicaran las variables a usar para automatizar tareas.

**main.yml**

**paquetesPhp 🡪** indicaremos los paquetes necesarios para utilizar la interfaz web monstaftp con index.php

ftp\_ip: 180.170.160.1

paquetesPhp:

  - php

  - php-cli

  - php-fpm

  - php-gd

  - php-common

  - php-mysql

  - php-xml

  - php-zip

  - php-curl

  - php-mbstring

  - php-xmlrpc

**tasks**

instalar.yml

- name: Instalar VSFTPD

  become: true

  apt:

    name: vsftpd

    state: latest

- name: Instalar Apache2

  become: true

  apt:

    name: apache2

    state: latest

configurar.yml

---

- name: Configurar VSFTPD

  template: src=vsftpd.conf.j2 dest=/etc/vsftpd.conf

- name: Adding nologin shell

  lineinfile:

    state: present

    dest: /etc/shells

    line: "/usr/sbin/nologin"

    regexp: "^/usr/sbin/nologin"

    insertafter: EOF

- name: Iniciar VSFTPD

  service: name=vsftpd state=started enabled=true

- name: Crear directorio para acceder a Monstaftp

  file:

    path: /home/ftp

    state: directory

    owner: www-data

    group: www-data

    mode: 0777

- name: Copiar monstaftp

  copy:

    src: ./files/mftp

    dest: /var/www/html

- name: Eliminar index.html

  command: rm /var/www/html/index.html

- name: Copiar fichero ports.conf

  copy:

    src: ./files/ports.conf

    dest: /etc/apache2/ports.conf

- name: Instalación PhP

  become: true

  apt:

    name: "{{ paquetesPhp }}"

    state: latest

- name: Reiniciar Apache2

  service: name=apache2 state=restarted enabled=true

main.yml

- import\_tasks: instalar.yml

- import\_tasks: configurar.yml

files

ports.conf 🡪 actualizar puerto 8080 que será utilizado en ftp

# If you just change the port or add more ports here, you will likely also

# have to change the VirtualHost statement in

# /etc/apache2/sites-enabled/000-default.conf

Listen 80

Listen 8080

<IfModule ssl\_module>

        Listen 443

</IfModule>

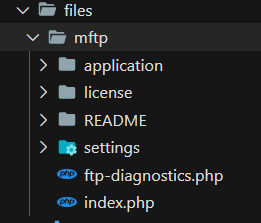
<IfModule mod\_gnutls.c>

        Listen 443

</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet

Interfaz web MONSTAFTP



**MYSQL**

**defaults 🡪** variables a utilizar en mysql.

**main.yml 🡪** usuarios y contraseñas

user\_mysql:

  - root

pass\_mysql: jome

nombre\_bbdd: ftp

nombre\_user: jome

**task**

instalar.yml

- name: Actualizacion de repositorios

  become: true

  apt:

    upgrade: 'yes'

    update\_cache: yes

    cache\_valid\_time: 3600

- name: Instalar PyMySql

  become: true

  apt:

    name: python3-pymysql

    state: latest

- name: Instalar MySQL

  become: true

  apt:

    name: mariadb-server

    state: latest

configurar.yml

- name: Iniciar MySql

  service: name=mysql state=started enabled=true

- name: Archivo my.conf

  copy: src=./files/my.conf dest=/tmp/file

- name: Actualizar contraseña de root

  mysql\_user:

    name: "{{ user\_mysql }}"

    password: "{{ pass\_mysql }}"

    login\_user: "{{ user\_mysql }}"

    login\_password: "{{ pass\_mysql }}"

    check\_implicit\_admin: yes

    priv: "\*.\*:ALL,GRANT"

    login\_unix\_socket: /var/run/mysqld/mysqld.sock

- name: Crear bbdd

  mysql\_db:

    login\_user: "{{ user\_mysql }}"

    login\_password: "{{ pass\_mysql }}"

    name: "{{ nombre\_bbdd }}"

    state: present

- name: Crear usuario bbdd

  mysql\_user:

    login\_user: "{{ user\_mysql }}"

    login\_password: "{{ pass\_mysql }}"

    name: "{{ nombre\_user }}"

    password: "{{ pass\_mysql }}"

    priv: "ftp.\*:ALL,GRANT"

    state: present

main.yml

- import\_tasks: instalar.yml

- import\_tasks: configurar.yml

**Instalación.yml 🡪** ejecución de los roles

- name: Configuración mysql

  hosts: sgbd

  remote\_user: root

  become: true

  roles:

    - { role: mysql }

- name: Configuración VSFTPD

  hosts: webservers

  remote\_user: root

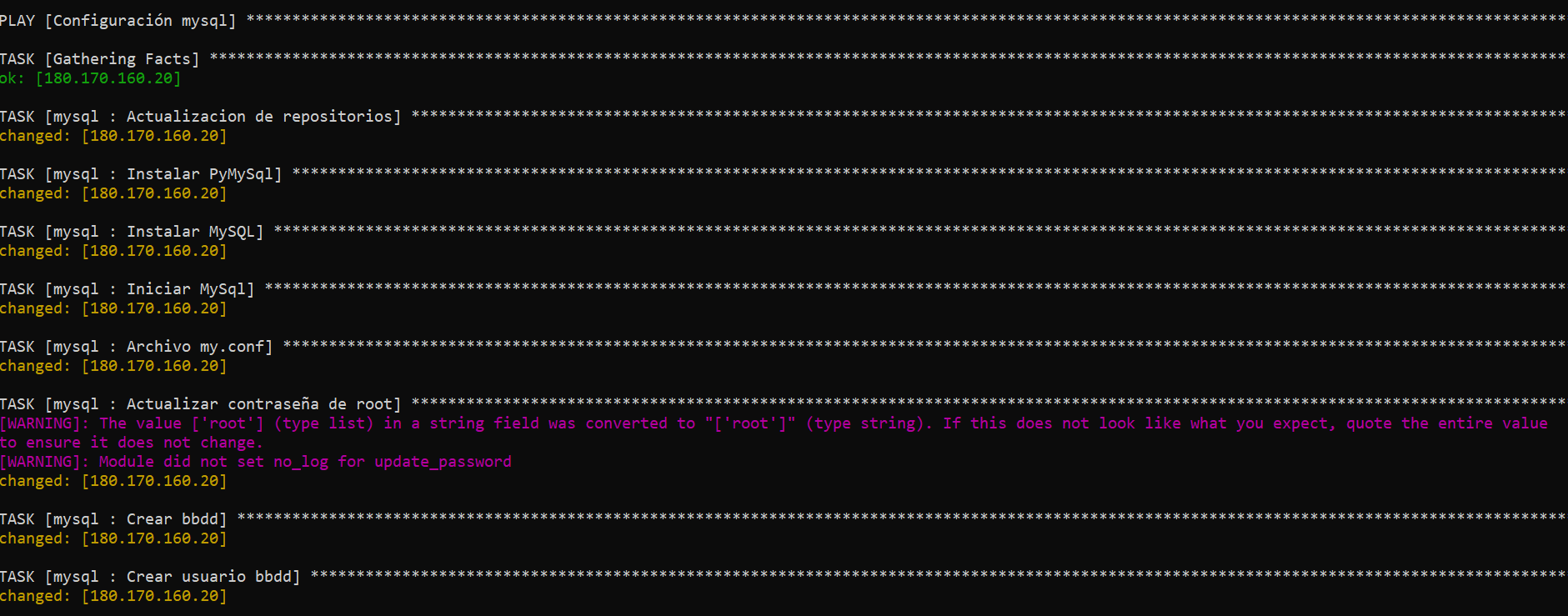
  become: true

  roles:

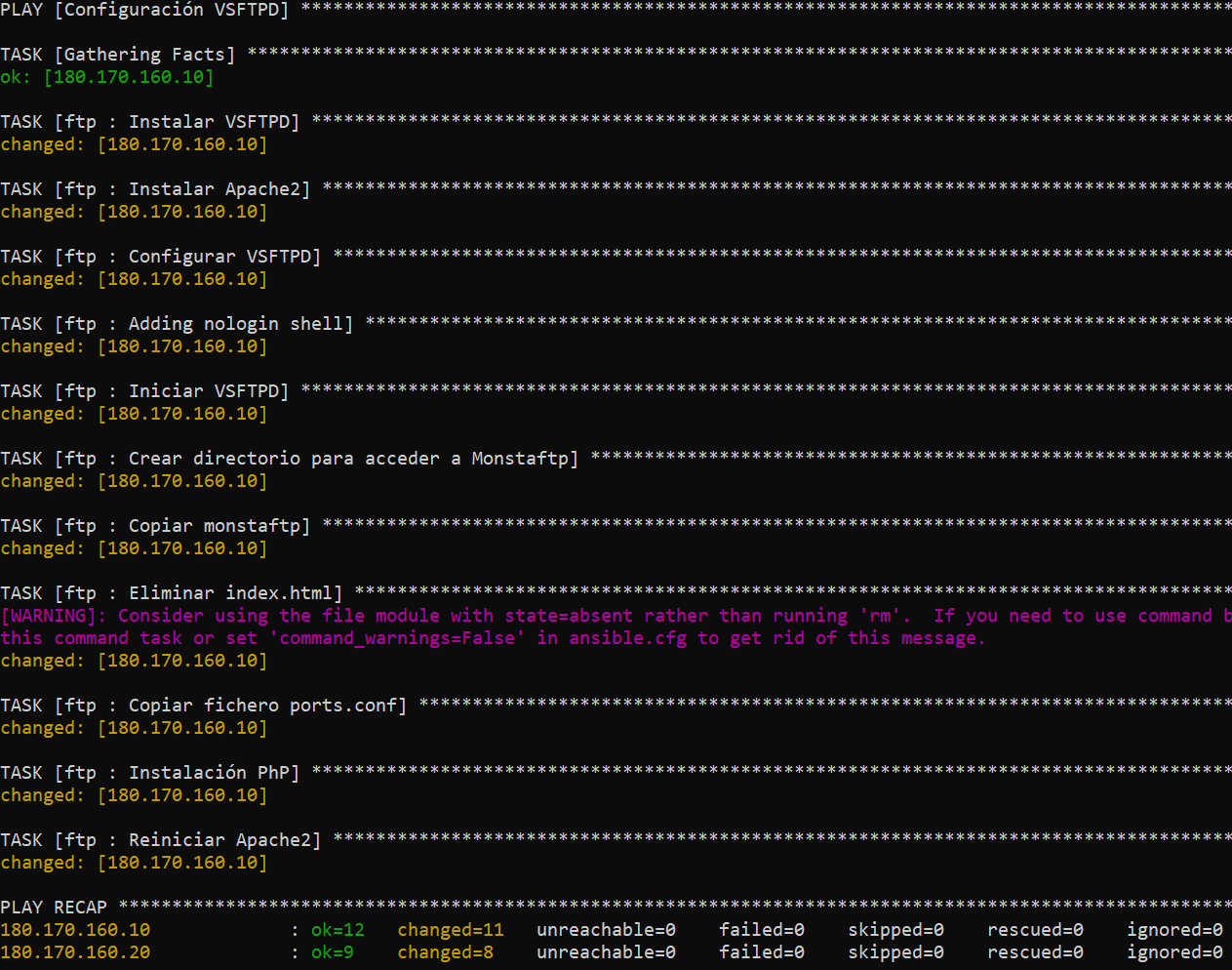
    - { role: ftp }

**Ejecución del playbook**

role: mysql

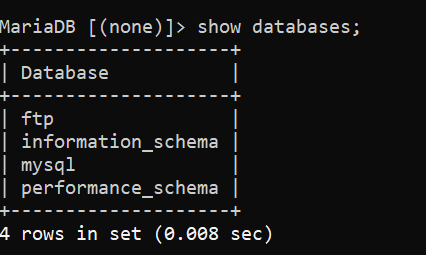


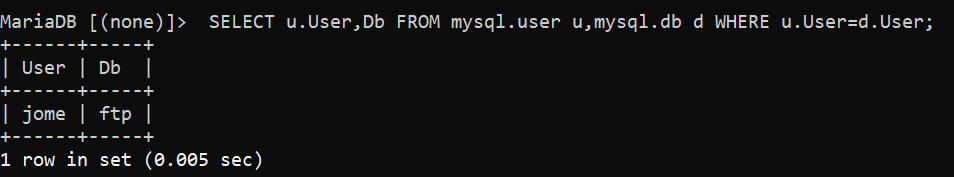
role: ftp



**CONTENEDOR MYSQL**

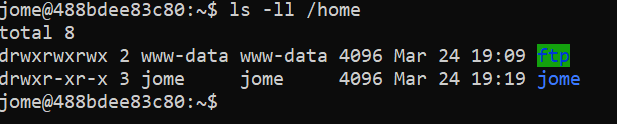
Si accedemos al contenedor ssh [jome@180.170.160.20](mailto:jome@180.170.160.20) comprobaremos que se ha realizado la configuración del rol “mysql”.





**CONTENEDOR FTP**

Comprobaremos que se ha creado el directorio ftp con los permisos asignados en el rol.



**MonstaFTP 🡪** aplicación FTP de código abierto basado en web, puede ser instalado de forma sencilla en cualquier servidor web que tenga soporte para PHP.

