



# ADMINISTRACIÓN DE INFRAESTRUCTURAS Y SISTEMAS INFORMÁTICOS (AISI)

Grado en Ingeniería Informática

Grado en Ingeniería Informática

Roberto R. Expósito ([roberto.rey.exposito@udc.es](mailto:roberto.rey.exposito@udc.es))

Jorge Veiga ([jorge.veiga@udc.es](mailto:jorge.veiga@udc.es))

# PRÁCTICA 4

## **Ansible**



# Objetivo

3

- El propósito de esta práctica es aprender a utilizar las opciones básicas de **Ansible**, una herramienta IaC de código abierto desarrollada y soportada comercialmente por Red Hat, Inc.
- Ansible permite gestionar configuraciones, aprovisionar y administrar recursos de infraestructura, desplegar aplicaciones y orquestar muchas otras tareas IT de una forma sencilla, flexible y ágil




<https://www.ansible.com>



# Consideraciones iniciales

4

- Clona el [repositorio de la práctica 3](#) para obtener los ficheros necesarios para realizar los ejercicios propuestos
  - **Recuerda:** sin espacios, acentos, eñes o caracteres “raros” en la ruta 
- **Debes usar el Vagrant box** creado en la práctica 1 con Packer junto con los ficheros *Vagrantfile* disponibles para cada ejercicio en el repositorio de esta práctica 3
  - **Excepto en el primer ejercicio**, donde se usará un *box* diferente que ya está configurado en el *Vagrantfile* proporcionado para este ejercicio
  - Para el resto de los ejercicios simplemente modifica el *box* en el *Vagrantfile* correspondiente para usar el tuyo
- Recuerda que la carpeta del proyecto Vagrant (donde reside el fichero *Vagrantfile*) se comparte con la ruta **/vagrant** en las VMs
  - Resulta muy útil para poder editar ficheros desde el *host* y acceder a ellos desde las VMs



# Justificación de la práctica

5

- La realización de esta práctica se justificará de la siguiente forma:
  - Documento en formato PDF que incluya las capturas de pantalla indicadas para demostrar la realización de la **parte principal de cada ejercicio**

- **Debes incluir capturas similares a las mostradas en las transparencias:**



- **11, 14 (EJ1); 19 (EJ2); 24 (EJ3); 32, 33 (EJ4)**

Busca este icono en la parte superior derecha



**IMPORTANTE**



- **ENTREGA** a través de Moodle: **13/04 (15:30)**
- **ES OBLIGATORIO** usar la nomenclatura que se propone para nombrar los recursos y debe apreciarse sin confusión en las capturas aportadas
  - **NO RECORTES** las capturas de pantalla, **debe verse toda la información** que sea relevante para comprobar el trabajo realizado
- **NO** seguir estas normas **IMPLICA UNA CALIFICACIÓN “C”** en esta práctica



# Ejercicio 1

6

- **Despliegue de una aplicación web Node.js+Express+MariaDB en CentOS mediante Vagrant y comandos Ansible *ad-hoc***
  - [Node.js](#) es un entorno de ejecución de JavaScript para el lado servidor diseñado para construir aplicaciones en red escalables
  - [Express](#) es un *framework* flexible escrito en JavaScript que permite desarrollar aplicaciones web Node.js de forma rápida
  - [MariaDB](#) es un sistema de gestión de bases de datos derivado de MySQL
- En el repositorio de la práctica dispones de todos los ficheros necesarios para este ejercicio que se muestran en la figura inferior

```
.
├── ansible.cfg
├── ansible.inventory
├── provisioning
│   ├── ansible-controller.sh
│   ├── servers.sh
│   └── vbox.rb
├── Vagrantfile
├── webapp
│   ├── app.js
│   └── package.json
```



# Ejercicio 1

7

- Ficheros relevantes para el ejercicio:
  - **Vagrantfile**: despliega **tres VMs** y las aprovisiona usando *shell* scripts
    - **ansible**: VM que actúa como nodo controlador Ansible
    - **webapp**: VM donde se ejecuta la aplicación web Node.js
      - El puerto 8080 del *host* está redireccionado al puerto 80 de esta VM
    - **db**: VM donde se ejecuta la base de datos MariaDB
  - **provisioning**: directorio con dos scripts para aprovisionar las VMs
  - **ansible.cfg**: fichero con configuración básica de Ansible
  - **ansible.inventory**: fichero de inventario de Ansible
  - **webapp**: directorio que contiene
    - **app.js**: fichero con el código fuente JavaScript de la aplicación Node.js
    - **package.json**: fichero en formato JSON que define la configuración y los módulos de los que depende la aplicación Node.js
      - Para instalar los módulos Node.js necesarios para la aplicación web usaremos el gestor de paquetes *Node Package Manager* (NPM)



# Ejercicio 1

8

- Guía orientativa:

- **Modifica el Vagrantfile para cambiar el *hostname* de las tres VMs**
- **Debes sustituir "xxx" por tus iniciales correspondientes**
- Modifica el fichero de inventario Ansible en función del *hostname* de tus VMs



Fichero de inventario  
modificado →

```
# Web application server
[webapp]
rre-aisi2223-webapp
# Database server
[db]
rre-aisi2223-db
# Group 'servers' with webapp and db
[servers:children]
webapp
db
```

- En el código fuente de la aplicación Node.js (*webapp/app.js*), **modifica el nombre** de la base de datos, el **host** donde se ejecuta y el **usuario** de conexión (líneas 7-10)
- **Debes sustituir "xxx" por tus iniciales correspondientes**
- **NO modifiques nada más en el código fuente**


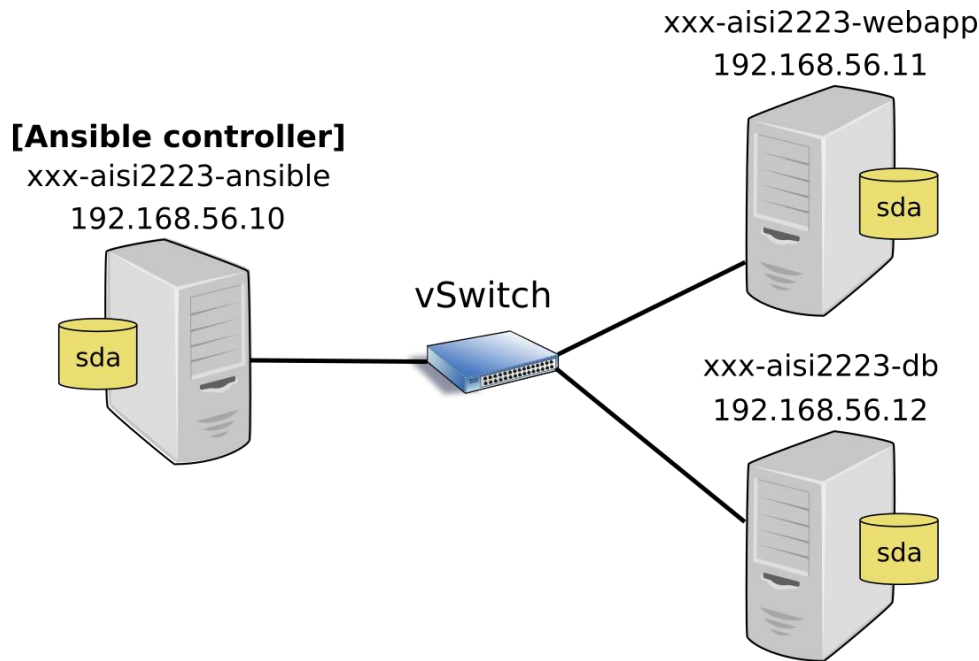




# Ejercicio 1

9

- Guía orientativa (cont.):
  - Despliega las VMs con Vagrant y accede por *ssh* al controlador Ansible
    - *vagrant ssh ansible*
  - Esquema gráfico del despliegue:



Disponemos de conectividad entre las VMs a nivel de IP y *hostname*. La conectividad *ssh* desde el controlador Ansible hacia las VMs webapp y db está configurado para hacerse sin introducir *password* (*ssh passwordless*)



# Ejercicio 1

10

- Guía orientativa (cont.):
  - Desde el controlador Ansible, comprueba que funciona la conectividad *ssh passwordless* con las VMs **webapp** y **db**

```
[vagrant@rre-aisi2223-ansible ~]$ ssh rre-aisi2223-webapp
The authenticity of host 'rre-aisi2223-webapp (192.168.56.11)' can't be established.
RSA key fingerprint is SHA256:gNRD0u/zfwc8j1bRGX70CLXAQPGpD/PVR6i6u/Z5UzI.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'rre-aisi2223-webapp,192.168.56.11' (RSA) to the list of known hosts.
[vagrant@rre-aisi2223-webapp ~]$
[vagrant@rre-aisi2223-webapp ~]$ exit
logout
Connection to rre-aisi2223-webapp closed.
[vagrant@rre-aisi2223-ansible ~]$ ssh rre-aisi2223-db
The authenticity of host 'rre-aisi2223-db (192.168.56.12)' can't be established.
RSA key fingerprint is SHA256:j2nZWTBIKLoiEyRb0U0K7HsBo85wd1UZKBdWk8KqTeY.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'rre-aisi2223-db,192.168.56.12' (RSA) to the list of known hosts.
[vagrant@rre-aisi2223-db ~]$
[vagrant@rre-aisi2223-db ~]$ exit
logout
Connection to rre-aisi2223-db closed.
[vagrant@rre-aisi2223-ansible ~]$
```

- Comprueba la instalación de Ansible ejecutando el comando: *ansible --version*

```
[vagrant@rre-aisi2223-ansible ~]$ ansible --version
ansible [core 2.13.3]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/vagrant/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3.9/site-packages/ansible
  ansible collection location = /home/vagrant/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.9.13 (main, Nov 16 2022, 15:31:39) [GCC 8.5.0 20210514 (Red Hat 8.5.0-15)]
  jinja version = 3.1.2
  libyaml = True
[vagrant@rre-aisi2223-ansible ~]$
```



# Ejercicio 1

11



- Guía orientativa (cont.):
  - Ejecuta el siguiente comando *ad-hoc* Ansible usando el módulo **shell** para comprobar los *hostnames* y la versión del *kernel* de las VMs **webapp** y **db**
    - `ansible servers -m shell -a "hostname && uname -a"`

Hostnames de las  
VMs correctamente  
configurados



```
[vagrant@rre-aisi2223-ansible ~]$ ansible servers -m shell -a "hostname && uname -a"
rre-aisi2223-db | CHANGED | rc=0 >>
rre-aisi2223-db
Linux rre-aisi2223-db 4.18.0-372.26.1.el8_6.x86_64 #1 SMP Tue Sep 13 18:09:48 UTC 2022 x86_64 x86_64
rre-aisi2223-webapp | CHANGED | rc=0 >>
rre-aisi2223-webapp
Linux rre-aisi2223-webapp 4.18.0-372.26.1.el8_6.x86_64 #1 SMP Tue Sep 13 18:09:48 UTC 2022 x86_64 x86_64
[vagrant@rre-aisi2223-ansible ~]$
```



# Ejercicio 1

12

- Usando **comandos ad-hoc Ansible** ejecutados desde el controlador:
  - Instala el paquete **python3-mysqclient** en **webapp** y **db** (módulo [package](#))
  - Instala los paquetes **npm** y **nodejs** solo en **webapp** (módulo **package**)
  - Copia el directorio `/vagrant/webapp` en la ruta `/home/vagrant` de **webapp** (módulo [copy](#))
    - Establece **vagrant** como el propietario de usuario y de grupo
  - Instala las dependencias de la aplicación Node.js en **webapp** (módulo [npm](#))
    - Como parámetro al módulo npm debemos indicar el *path* en **webapp** donde reside el fichero `package.json`, el cual define las dependencias que se instalarán
      - `ansible webapp -b -m npm -a "path=/home/vagrant/webapp"`
  - **Conéctate por ssh a webapp** y comprueba la instalación (ver imagen inferior)

Dependencias de la aplicación Node.js instaladas en el directorio `webapp/node_modules` de la VM **webapp**, junto con el código fuente Javascript (`app.js`)

```
[vagrant@rre-aisi2223-ansible ~]$ ssh rre-aisi2223-webapp
Last login: Wed Jan 25 10:22:28 2023 from 192.168.56.10
[vagrant@rre-aisi2223-webapp ~]$ ls -l webapp/
total 36
-rw-r--r--. 1 vagrant vagrant 959 Jan 25 10:21 app.js
drwxr-xr-x. 68 vagrant vagrant 4096 Jan 25 10:22 node_modules
-rw-r--r--. 1 vagrant vagrant 21647 Jan 25 10:22 package-lock.json
-rw-r--r--. 1 vagrant vagrant 172 Jan 25 10:21 package.json
[vagrant@rre-aisi2223-webapp ~]$ ls webapp/node_modules/
accepts      content-type  ee-first     fresh
array-flatten  cookie        encodeurl    function-bind
bignumber.js  cookie-signature  escape-html  get-intrinsic
body-parser   core-util-is  etag         has
bytes         debug         express      has-symbols
call-bind     depd          finalhandler http-errors
content-disposition  destroy      forwarded    iconv-lite
[vagrant@rre-aisi2223-webapp ~]$
```



# Ejercicio 1

13

- Usando **comandos ad-hoc Ansible** ejecutados desde el controlador (cont.):
  - Instala el paquete **mariadb-server** en **db** (módulo **package**)
  - Arranca el servicio **mariadb** en **db** y activa su inicio automático en el arranque del sistema (módulo [service](#))
  - Crea la base de datos en **db** (módulo [mysql\\_db](#))
    - Como parámetro **db** indica el nombre de la base de datos que figura en el código fuente de la aplicación Node.js (ver fichero `webapp/app.js`)
  - Crea en **db** el usuario de conexión a la base de datos (módulo [mysql\\_user](#))
    - Como parámetro **name** especifica el nombre de usuario apropiado en tu caso
    - Como parámetro **password** usa el que figura en el código fuente de la aplicación
    - Dale a tu usuario todos los privilegios de acceso (parámetro **priv=\*.\*:ALL**)
    - Como parámetro **host** especifica el *hostname* apropiado para tu VM **webapp**
      - Este parámetro *host* especifica la máquina que se conectará a la base de datos
  - Finalmente, ejecuta la aplicación Web Node.js en **webapp** usando el módulo por defecto ([command](#))
    - `ansible webapp -b -a "node /home/vagrant/webapp/app.js" &`



# Ejercicio 1

14



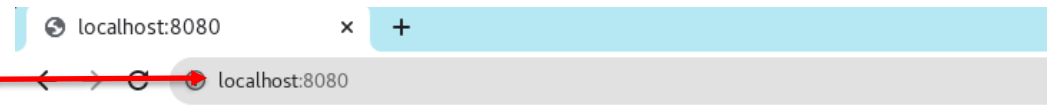
- Accede **desde el controlador Ansible** a la VM que ejecuta la aplicación web Node.js usando el comando **curl**
- Prueba usando tanto la dirección IP como el *hostname* de la VM webapp

```
[vagrant@rre-aisi2223-ansible ~]$ curl 192.168.56.11
<h2><u>GEI AISI 2022/2023: Node.js+Express+MariaDB</u>
<p>URL: http://192.168.56.11/
<p> Wed Jan 25 2023 10:34:28 GMT+0000 (Coordinated Universal Time)
<p>MariaDB connection from user rre-aisi2223: <span style="color: green;">PASSED</span>
[vagrant@rre-aisi2223-ansible ~]$
[vagrant@rre-aisi2223-ansible ~]$ curl rre-aisi2223-webapp
<h2><u>GEI AISI 2022/2023: Node.js+Express+MariaDB</u>
<p>URL: http://rre-aisi2223-webapp/
<p> Wed Jan 25 2023 10:34:45 GMT+0000 (Coordinated Universal Time)
<p>MariaDB connection from user rre-aisi2223: <span style="color: green;">PASSED</span>
[vagrant@rre-aisi2223-ansible ~]$
```

El test de conexión de la aplicación Web a la base de datos funciona

- Accede ahora **desde el navegador de tu host**: <http://localhost:8080>

El acceso por *localhost* funciona porque el puerto 8080 de tu *host* está redireccionado al puerto 80 de la VM webapp (ver *Vagrantfile*)



**GEI AISI 2022/2023: Node.js+Express+MariaDB**

**URL: http://localhost:8080/**

**Wed Jan 25 2023 10:37:35 GMT+0000 (Coordinated Universal Time)**

**MariaDB connection from user rre-aisi2223: PASSED**

Comprueba si funciona el acceso desde el navegador de tu *host* usando el *hostname* y la dirección IP de la VM webapp





## Ejercicio 2

15

- **Despliegue de un servidor web Apache en Ubuntu mediante Vagrant y Ansible Playbooks**
- En el repositorio de la práctica dispones de todos los ficheros necesarios para este ejercicio que se muestran en la figura inferior
  - **Modifica el Vagrantfile para cambiar el box y el hostname de la VM** 
  - Configura en el Vagrantfile un *provisioner* de tipo **ansible\_local**:
    - [https://developer.hashicorp.com/vagrant/docs/provisioning/ansible\\_local](https://developer.hashicorp.com/vagrant/docs/provisioning/ansible_local)
    - Define el playbook a ejecutar como “*provisioning/playbook.yml*”
    - Configura el modo de instalación de Ansible en la VM mediante “pip3”

```
.
├── provisioning
│   ├── handlers
│   │   └── main.yml
│   ├── playbook.yml
│   ├── tasks
│   │   └── main.yml
│   ├── templates
│   │   ├── apache.conf.j2
│   │   └── index.html.j2
│   ├── vars
│   │   └── main.yml
│   └── vbox.rb
└── Vagrantfile
```



# Ejercicio 2

16

- **Ficheros del ejercicio:**
  - **provisioning:** directorio que contiene el Playbook de Ansible y todos los ficheros auxiliares necesarios para aprovisionar la VM
    - **playbook.yml:** Playbook de Ansible que utiliza los siguientes ficheros
      - **vars/main.yml:** define las variables
      - **handlers/main.yml:** define los manejadores
      - **tasks/main.yml:** define las tareas
  - **templates:** directorio que contiene
    - **apache.conf.j2:** plantilla Jinja2 con la configuración del sitio web
    - **index.html.j2:** plantilla Jinja2 con el código HTML de la página web

Contenido del fichero  
playbook.yml →

```
---  
- hosts: all  
  become: yes  
  
  vars_files:  
    - vars/main.yml  
  
  handlers:  
    - import_tasks: handlers/main.yml  
  tasks:  
    - import_tasks: tasks/main.yml
```





# Ejercicio 2

17

- Guía orientativa:
  - En el fichero ***vars/main.yml*** modifica las variables ***http\_user***, ***http\_host*** y ***http\_conf***
    - **Debes poner tu nombre y apellidos y sustituir "xxx" por tus iniciales correspondientes**
  - En el fichero ***handlers/main.yml***
    - Define un manejador para reiniciar el servidor web Apache (servicio ***apache2***)
  - En el fichero ***tasks/main.yml*** define **seis tareas**
    1. Usando el módulo ***package***, instala el servidor web Apache (paquete ***apache2***)
    2. Usando el módulo ***apache2\_module***, activa el módulo de Apache ***rewrite***
    3. Usando el módulo ***file***, elimina el fichero ***/etc/apache2/sites-enabled/{{ default\_site }}***
      - Esa ruta representa el sitio web por defecto que sirve Apache
    4. Usando el módulo ***file***, crea el directorio ***/var/www/{{ http\_host }}*** para el nuevo sitio web
      - Usa permisos "0755" para el nuevo directorio
    5. Usando el módulo ***template***, copia el fichero ***templates/index.html.j2*** en la siguiente ruta: ***/var/www/{{ http\_host }}/index.html***
    6. Usando el módulo ***template***, copia el fichero ***templates/apache.conf.j2*** en la siguiente ruta: ***/etc/apache2/sites-enabled/{{ http\_conf }}*** para habilitar el nuevo sitio web
  - En las tareas **2, 3, 5 y 6** notifica el **handler** para reiniciar el servicio de Apache



## Ejercicio 2

18

- Despliega la VM con Vagrant y fíjate en la fase de aprovisionamiento que ejecuta el Playbook de Ansible

Ejemplo de ejecución del Playbook de Ansible. Recuerda que puedes usar ***vagrant up --provision*** para forzar el aprovisionamiento. Resulta útil durante el desarrollo y prueba de un playbook

```
default: Running ansible-playbook...
→ PLAY [all] *****

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

TASK [Install apache server] *****
changed: [default]

TASK [Enable rewrite Apache module] *****
changed: [default]

TASK [Disable Apache default site] *****
changed: [default]

TASK [Create document root] *****
changed: [default]

TASK [Copy index.html template page] *****
changed: [default]

TASK [Setup Apache virtual host] *****
changed: [default]

RUNNING HANDLER [Restart apache] *****
changed: [default]

PLAY RECAP *****
default                : ok=8    changed=7    unreachable=0    failed=0    skipped=0
```



# Ejercicio 2

19

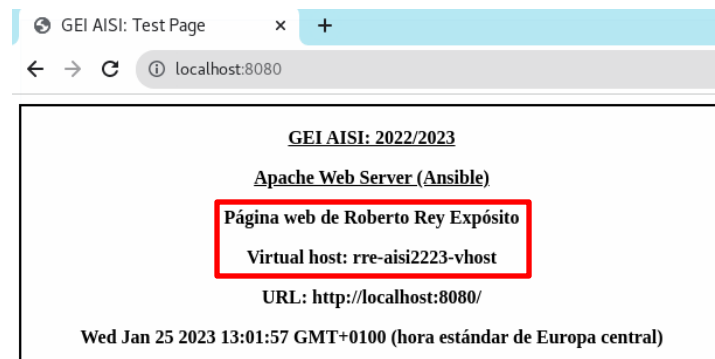


- Conéctate por *ssh* a la VM y accede al servidor web usando *curl*

Hostname correctamente configurado

```
vagrant@rre-aisi2223-ansible:~$ curl localhost
<html>
<head>
  <meta charset= "utf-8">
  <title>GEI AISI: Test Page</title>
  <script type="text/javascript">
    function getURL() {
      document.write("URL: " + window.location.href);
    }
    function getTime() {
      document.getElementById("current_date").innerHTML = Date();
    }
  </script>
</head>
<body>
  <div style="width:600px;height:220px;border:2px solid #000;text-align: center;">
    <strong><br>
    <u>GEI AISI: 2022/2023</u>
    <p><u>Apache Web Server (Ansible)</u></p>
    <p>Página web de Roberto Rey Expósito</p>
    <p>Virtual host: rre-aisi2223-vhost</p>
    <p><script>getURL();</script></p>
    <p><div id="current_date"><script>getTime();</script></p>
  </strong>
</div>
</body>
</html>
vagrant@rre-aisi2223-ansible:~$
```

- Accede desde el navegador de tu *host* al servidor web


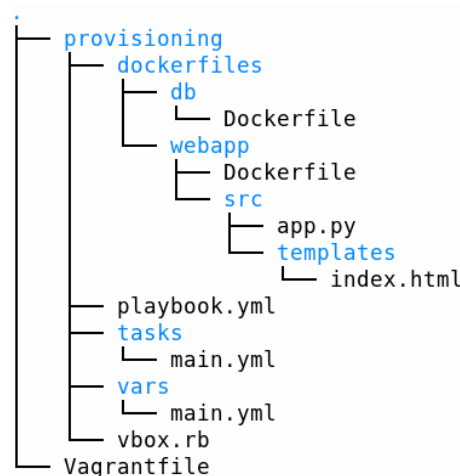




# Ejercicio 3

20

- **Despliegue de una aplicación web Flask+MySQL en Ubuntu mediante Vagrant, Ansible Playbooks y Docker**
- En el repositorio de la práctica dispones de todos los ficheros necesarios para este ejercicio que se muestran en la figura inferior
  - **Modifica el Vagrantfile para cambiar el box y el hostname de la VM** 
  - Configura en el Vagrantfile un *provisioner* de tipo **ansible\_local**:
    - [https://developer.hashicorp.com/vagrant/docs/provisioning/ansible\\_local](https://developer.hashicorp.com/vagrant/docs/provisioning/ansible_local)
    - Define el playbook a ejecutar como “*provisioning/playbook.yml*”
    - Configura el modo de instalación de Ansible en la VM mediante “pip3”



Este *provisioner* se configuró de la misma forma en el Vagrantfile del ejercicio previo, por lo que puedes copiar la parte del código correspondiente en el Vagrantfile de este ejercicio. NO copies el Vagrantfile completo para evitar conflictos de nombrado de las VMs en VirtualBox



# Ejercicio 3

21

- **Ficheros del ejercicio:**
  - **provisioning:** directorio que contiene el Playbook de Ansible y todos los ficheros auxiliares necesarios para aprovisionar la VM
    - **playbook.yml:** Playbook de Ansible que utiliza los siguientes ficheros
      - **vars/main.yml:** define las variables
      - **tasks/main.yml:** define las tareas
  - **dockerfiles:** directorio que contiene los ficheros necesarios para los contenedores
    - **db:** *Dockerfile* para la base de datos MySQL
    - **webapp:** *Dockerfile* para la aplicación web Flask
      - **src/app.py:** código fuente de la aplicación
      - **src/templates/index.html:** plantilla con la página que sirve la aplicación

Contenido del fichero  
playbook.yml →

```
---  
- hosts: all  
  become: yes  
  
vars_files:  
  - vars/main.yml  
  
tasks:  
  - import_tasks: tasks/main.yml
```



# Ejercicio 3

22

- Guía orientativa:
  - En el fichero **vars/main.yml** modifica las variables definidas
  - En el fichero **dockerfiles/webapp/src/app.py** modifica la variable **db\_host** (línea 10)
    - **En ambos casos, debes sustituir "xxx" por tus iniciales y NO modificar nada más**
  - En el fichero **tasks/main.yml** define las siguientes tareas:
    - Usando el módulo **package**, instala "Docker SDK for Python" (paquete **python3-docker**)
      - Es una dependencia de los módulos de Ansible para Docker que vamos a usar
    - Usando el módulo [\*\*docker\\_image\*\*](#), crea las dos imágenes usando el **loop** ya definido
    - Usando el módulo [\*\*docker\\_network\*\*](#), crea una red de tipo **bridge** y nómbrala usando la variable **network** definida en **vars/main.yml**
    - Usando el módulo [\*\*docker\\_container\*\*](#), crea un contenedor para ejecutar MySQL usando su imagen correspondiente y conéctalo a la red creada previamente
      - Nombra el contenedor usando la variable **db\_container** definida en **vars/main.yml**
      - Usa la opción **purge\_networks** para que el contenedor solo se conecte a la red indicada
    - Usando el módulo **docker\_container**, crea un contenedor para ejecutar la aplicación Flask usando su imagen correspondiente y conéctalo a la red creada previamente
      - Nombra el contenedor usando la variable **webapp\_container** definida en **vars/main.yml**
      - Usa la opción **purge\_networks** para que el contenedor solo se conecte a la red indicada
      - Publica el puerto 5000 del contenedor al puerto 80 de la VM



# Ejercicio 3

23

- Despliega la VM con Vagrant y fíjate en la fase de aprovisionamiento que ejecuta el Playbook de Ansible

Ejemplo de ejecución del Playbook de Ansible. Recuerda que puedes usar **vagrant up --provision** para forzar el aprovisionamiento. Resulta útil durante el desarrollo y prueba de un playbook

```
==> default: Running provisioner: ansible_local...
      default: Installing Ansible...
      default: Running ansible-playbook...

PLAY [all] *****

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

TASK [Install Docker SDK for Python] *****
changed: [default]

TASK [Build Docker images from Dockerfiles] *****
changed: [default] => (item={'name': 'rre-aisi2223-webapp-img', 'directory': 'webapp'})
changed: [default] => (item={'name': 'rre-aisi2223-db-img', 'directory': 'db'})

TASK [Create network] *****
changed: [default]

TASK [Run MySQL container] *****
changed: [default]

TASK [Run Flask container] *****
changed: [default]

PLAY RECAP *****
default                : ok=6    changed=5    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```



# Ejercicio 3

24



- Conéctate por *ssh* a la VM y obtén información sobre:
  - Contenedores (comprueba su estado y sus nombres), imágenes y redes

```
vagrant@rre-aisi2223-ansible:~$ docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                                NAMES
318fc44ed592   rre-aisi2223-webapp-img             "flask run"             5 minutes ago Up 5 minutes   0.0.0.0:80->5000/tcp               rre-aisi2223-webapp
cddb374dadee   rre-aisi2223-db-img                 "docker-entrypoint.s..." 5 minutes ago Up 5 minutes   3306/tcp, 33060/tcp               rre-aisi2223-db

vagrant@rre-aisi2223-ansible:~$ docker image ls
REPOSITORY          TAG         IMAGE ID      CREATED        SIZE
rre-aisi2223-db-img  latest     6f69ce64f7ee  5 minutes ago  452MB
rre-aisi2223-webapp-img latest     5fade881fdda  6 minutes ago  482MB
mysql               5.7       e982339a20a5  8 days ago    452MB
ubuntu              focal     d5447fc01ae6  6 weeks ago    72.8MB

vagrant@rre-aisi2223-ansible:~$ docker network ls
NETWORK ID      NAME                DRIVER  SCOPE
d89577d1d34f    bridge             bridge  local
26665acf4d55    host               host    local
086e0e004b3e    none              null    local
991aede1d909    rre-aisi2223-network bridge  local

vagrant@rre-aisi2223-ansible:~$
```

- Accede desde el navegador de tu **host** a la aplicación web para comprobar el correcto funcionamiento y el acceso a la base de datos







# Ejercicio 3

25


- Realiza múltiples accesos desde el navegador de tu **host** (F5) y revisa desde la VM los logs del contenedor que ejecuta la aplicación web

```
vagrant@rre-aisi2223-ansible:~$ docker logs rre-aisi2223-webapp
* Serving Flask app 'app.py'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://172.18.0.3:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 277-321-249
10.0.2.2 - - [26/Jan/2023 09:57:17] "GET / HTTP/1.1" 200 -
10.0.2.2 - - [26/Jan/2023 09:57:17] "GET /favicon.ico HTTP/1.1" 404 -
10.0.2.2 - - [26/Jan/2023 09:57:40] "GET / HTTP/1.1" 200 -
10.0.2.2 - - [26/Jan/2023 09:57:41] "GET / HTTP/1.1" 200 -
10.0.2.2 - - [26/Jan/2023 09:57:43] "GET / HTTP/1.1" 200 -
10.0.2.2 - - [26/Jan/2023 09:57:43] "GET / HTTP/1.1" 200 -
vagrant@rre-aisi2223-ansible:~$
```



# Ejercicio 4

26

- **Despliegue de WordPress en un entorno LAMP (Linux+Apache+MySQL+PHP) basado en Ubuntu mediante Vagrant y Ansible Roles**
  - [WordPress](#) es un sistema de gestión de contenidos de código abierto desarrollado en PHP enfocado a la creación y gestión de páginas web
    - Orientado a entornos que ejecuten MySQL y Apache, pero también soporta otros entornos (p.e. PostgreSQL y Nginx)
- En el repositorio de la práctica dispones de todos los ficheros necesarios para este ejercicio que se muestran en la figura inferior
  - **Modifica el Vagrantfile para cambiar el box y el hostname de la VM** 

```
.
├── provisioning
│   ├── playbook.yml
│   ├── requirements.yml
│   ├── tasks
│   │   └── main.yml
│   ├── templates
│   │   └── config-default.php.j2
│   └── vars
│       └── main.yml
├── vbox.rb
└── Vagrantfile
```



# Ejercicio 4

27

- **Ficheros del ejercicio:**

- **provisioning:** directorio que contiene el Playbook de Ansible y todos los ficheros auxiliares necesarios para aprovisionar la VM

- **playbook.yml:** Playbook de Ansible que utiliza los siguientes ficheros

- **vars/main.yml:** define las variables
- **tasks/main.yml:** define las tareas

Contenido del fichero **playbook.yml** →

```
---
- hosts: all
  become: true

  vars_files:
    - vars/main.yml

  roles:
    - geerlingguy.firewall
    - geerlingguy.mysql
    - geerlingguy.apache
    - geerlingguy.php-mysql
    - geerlingguy.php

  tasks:
    - import_tasks: tasks/main.yml
```

- **templates:** directorio que contiene
  - **config-default-php.j2:** plantilla Jinja2 con la configuración de la base de datos necesaria para WordPress
- **requirements.yml:** fichero que **deberás completar** para definir los roles de Ansible que usaremos en el despliegue (ver Playbook)
  - Todos los roles están disponibles en [Ansible Galaxy](#)



# Ejercicio 4

28

- Guía orientativa:
  - Completa el fichero **requirements.yml** para definir los roles de Ansible necesarios para el despliegue del entorno LAMP
    - Especifica de forma explícita la versión de cada rol (usa la última disponible en Galaxy)
      - Ver transparencia 68 del tutorial de Ansible
    - <https://galaxy.ansible.com/geerlingguy/apache>
    - <https://galaxy.ansible.com/geerlingguy/mysql>
    - <https://galaxy.ansible.com/geerlingguy/php>
    - <https://galaxy.ansible.com/geerlingguy/php-mysql>
    - <https://galaxy.ansible.com/geerlingguy/firewall>
  - Configura en el *Vagrantfile* un *provisioner* de tipo **ansible\_local**:
    - Configura el modo de instalación y el *playbook* a ejecutar igual que en ejercicios previos
    - Configura dicho aprovisionamiento con Ansible para que realice la descarga automática de los roles desde Galaxy y se instalen en la VM
      - Debes indicar la ruta al fichero que declara los roles (**requirements.yml**)
      - Debes instalar los roles en la ruta **/home/vagrant/roles** de la VM
      - Para ello, consulta las opciones **galaxy\_role\_file** y **galaxy\_roles\_path**
        - [https://developer.hashicorp.com/vagrant/docs/provisioning/ansible\\_common](https://developer.hashicorp.com/vagrant/docs/provisioning/ansible_common)



# Ejercicio 4

29

- Despliega la VM con Vagrant y fíjate que se descargan los roles durante la fase de aprovisionamiento antes de ejecutar el Playbook

Descarga e  
instalación  
de los roles

```
==> default: Running provisioner: ansible_local...
      default: Installing Ansible...
      default: Running ansible-galaxy...
Starting galaxy role install process
- downloading role 'firewall', owned by geerlingguy
- downloading role from https://github.com/geerlingguy/ansible-role-firewall/archive/2.5.1.tar.gz
- extracting geerlingguy.firewall to /home/vagrant/roles/geerlingguy.firewall
- geerlingguy.firewall (2.5.1) was installed successfully
- downloading role 'mysql', owned by geerlingguy
- downloading role from https://github.com/geerlingguy/ansible-role-mysql/archive/4.3.0.tar.gz
- extracting geerlingguy.mysql to /home/vagrant/roles/geerlingguy.mysql
- geerlingguy.mysql (4.3.0) was installed successfully
- downloading role 'apache', owned by geerlingguy
- downloading role from https://github.com/geerlingguy/ansible-role-apache/archive/3.3.0.tar.gz
- extracting geerlingguy.apache to /home/vagrant/roles/geerlingguy.apache
- geerlingguy.apache (3.3.0) was installed successfully
- downloading role 'php-mysql', owned by geerlingguy
- downloading role from https://github.com/geerlingguy/ansible-role-php-mysql/archive/2.1.0.tar.gz
- extracting geerlingguy.php-mysql to /home/vagrant/roles/geerlingguy.php-mysql
- geerlingguy.php-mysql (2.1.0) was installed successfully
- dependency geerlingguy.php already pending installation.
- downloading role 'php', owned by geerlingguy
- downloading role from https://github.com/geerlingguy/ansible-role-php/archive/4.8.0.tar.gz
- extracting geerlingguy.php to /home/vagrant/roles/geerlingguy.php
- geerlingguy.php (4.8.0) was installed successfully
      default: Running ansible-playbook...
```

Ejecución del  
Playbook

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

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

TASK [geerlingguy.firewall : Ensure iptables is present.] *****
ok: [default]
```

```
RUNNING HANDLER [geerlingguy.php : restart php-fpm] *****
skipping: [default]
```

Fin de la ejecución  
del Playbook

```
PLAY RECAP *****
default                : ok=97   changed=32   unreachable=0   failed=0   skipped=43
```



# Ejercicio 4

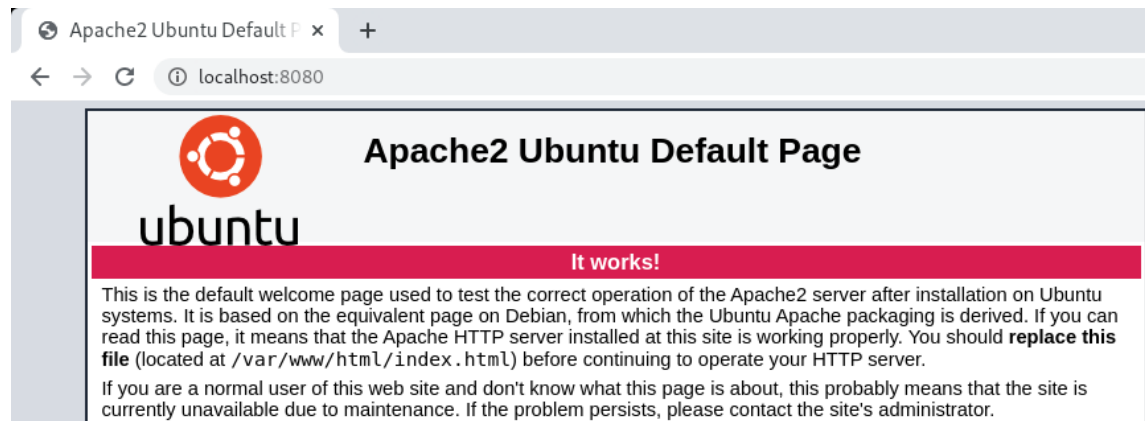
30

- Conéctate por `ssh` a la VM y comprueba que la instalación de los roles se ha realizado en la ruta indicada previamente (`/home/vagrant/roles`)

```
vagrant@rre-aisi2223-ansible:~$ pwd
/home/vagrant
vagrant@rre-aisi2223-ansible:~$ ls -la roles/
total 28
drwxrwxr-x 7 vagrant vagrant 4096 Jan 26 16:18 .
drwxr-xr-x 6 vagrant vagrant 4096 Jan 26 16:18 ..
drwxrwxr-x 10 vagrant vagrant 4096 Jan 26 16:18 geerlingguy.apache
drwxrwxr-x 9 vagrant vagrant 4096 Jan 26 16:18 geerlingguy.firewall
drwxrwxr-x 10 vagrant vagrant 4096 Jan 26 16:18 geerlingguy.mysql
drwxrwxr-x 10 vagrant vagrant 4096 Jan 26 16:18 geerlingguy.php
drwxrwxr-x 8 vagrant vagrant 4096 Jan 26 16:18 geerlingguy.php-mysql
vagrant@rre-aisi2223-ansible:~$
```

- Accede desde el navegador de tu **host** al servidor web Apache

Debes ver la página por defecto ya que no hemos modificado la configuración de Apache





# Ejercicio 4

31

- Accede ahora al sitio web de WordPress para proceder a su configuración e instalación inicial: <http://localhost:8080/wordpress>



**Debes usar el siguiente formato para Site Title y Username: xxx-aisi2223-wordpress**

**Pon cualquier password que recuerdes**

**Puedes usar cualquier email ficticio siempre que cumpla el formato esperado**

WordPress › Installation x +

← → ↻ localhost:8080/wordpress/wp-admin/install.php

Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Don't worry, you can always change these settings later.

Site Title rre-aisi2223-wordpress

Username rre-aisi2223-wordpress

Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

Password ..... Show

Strong

Important: You will need this password to log in. Please store it in a secure location.

Your Email email@gmail.com

Double-check your email address before continuing.

Search Engine Visibility ☐ Discourage search engines from indexing this site

It is up to search engines to honor this request.

Install WordPress

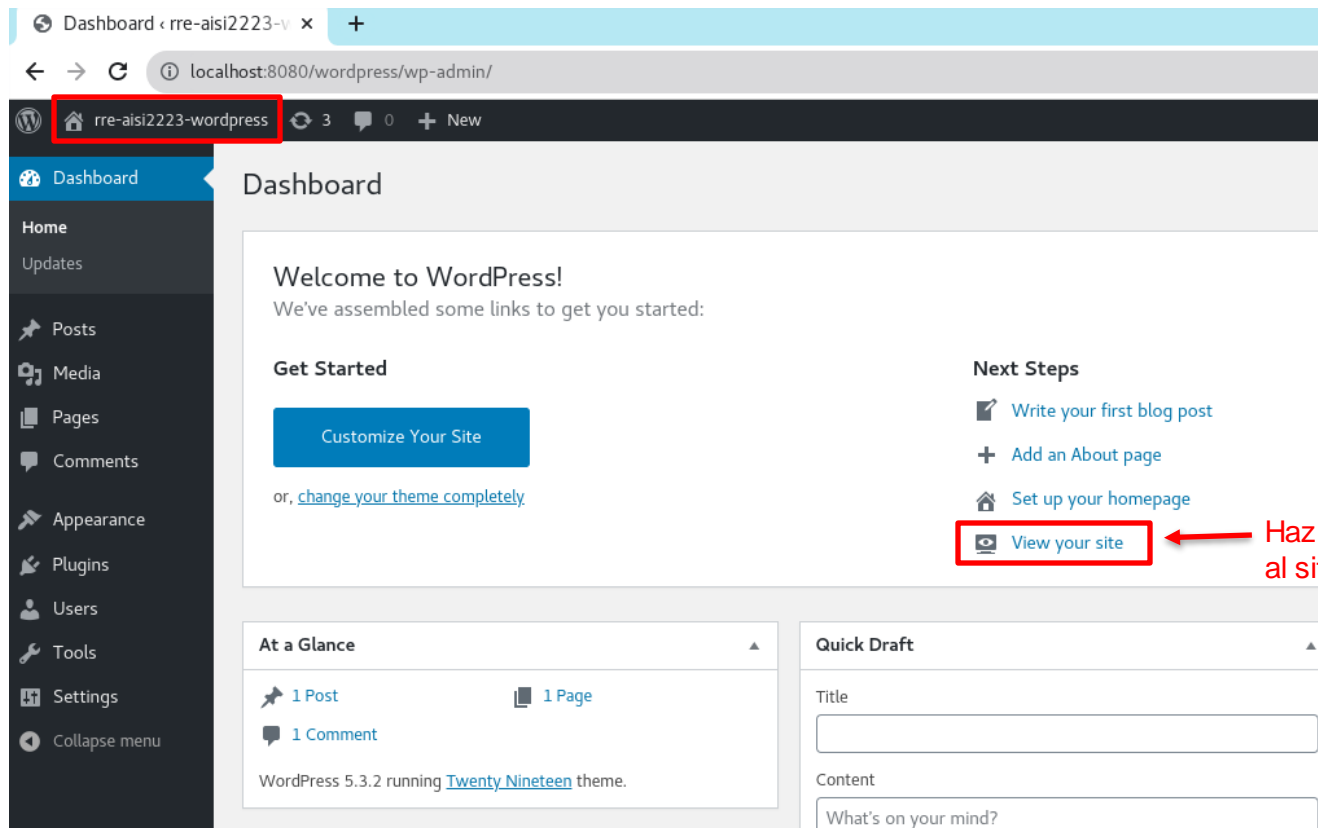


# Ejercicio 4

32



- Tras completar el proceso de instalación, logueate con tu usuario para acceder al *dashboard* que permite administrar el sitio web



Haz *click* aquí para acceder al sitio web por defecto





# Ejercicio 4

33



- Aspecto por defecto del sitio web

