



ADMINISTRACIÓN DE INFRAESTRUCTURAS Y SISTEMAS INFORMÁTICOS (AISI)

Grado en Ingeniería Informática

Grado en Ingeniería Informática

Roberto R. Expósito (roberto.rey.exposito@udc.es)

PRÁCTICA 3

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



A N S I B L E

<https://github.com/ansible/ansible>



Justificación de la práctica

4

- La **entrega** de la práctica consistirá en un **breve documento** en formato **PDF** que incluya las **todas capturas de pantalla** mostradas en las transparencias:
 - 14, 15, 16 (EJ1); 21, 22 (EJ2); 28, 29 (EJ3); 38, 39 (EJ4)**



Para ayudar a identificarlas, estas transparencias incluyen el icono de un monitor en la parte superior derecha



IMPORTANTE




- En ocasiones, durante la práctica, se pide crear recursos con un nombre que empieza por un **prefijo** que contiene información del estudiante y del curso
- ES OBLIGATORIO** usar la siguiente nomenclatura para nombrar los recursos:
<iniciales del nombre y apellidos><curso>-<nombre del recurso>
 - Ejemplo: El alumno Roberto Rey Expósito, que hace la práctica en el curso 2024/2025, utilizará el siguiente prefijo: **rre2425**
- 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 la práctica



Consideraciones iniciales

5

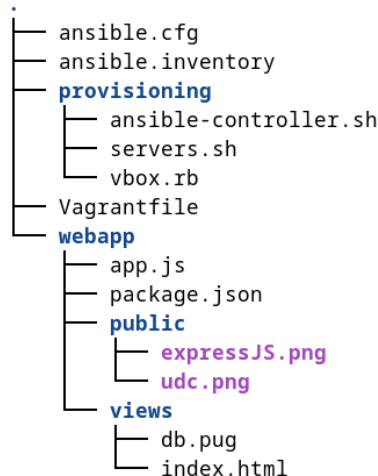
- Clona el [repositorio de la práctica 3](#) para obtener los ficheros necesarios para realizar los ejercicios propuestos
 - **Recuerda:** sin espacios, acentos, eñes o caracteres “raros” en la ruta 
- Cada ejercicio proporciona un fichero *Vagrantfile* que deberás modificar de acuerdo a las instrucciones dadas en el propio ejercicio
 - **A excepción del ejercicio 1, deberás usar el Vagrant box** creado con Packer en la práctica 1
 - En el primer ejercicio se usará un *box* diferente que ya está configurado en el *Vagrantfile* proporcionado
- Recuerda que la carpeta del proyecto Vagrant (donde reside el fichero *Vagrantfile*) se comparte con la ruta **/vagrant** en las VM
 - Resulta muy útil para poder editar ficheros desde el *host* y acceder a ellos desde las VM



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* web ligero y 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





Ejercicio 1

7

- Ficheros relevantes para el ejercicio:
 - **Vagrantfile:** despliega **tres VM** y las aprovisiona usando *shell scripts*
 - **ansible:** VM que actúa como nodo controlador Ansible
 - **web:** 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 que contiene los *scripts* para aprovisionar las VM
 - **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 web Node.js
 - Para instalar los módulos Node.js necesarios para la aplicación web usaremos el gestor de paquetes *Node Package Manager* (NPM)
 - **views:** directorio con las plantillas de las vistas HTML que sirve la aplicación web
 - **public:** directorio que contiene los ficheros estáticos que sirve la aplicación web



Ejercicio 1

8

- **Modifica el Vagrantfile para cambiar el *hostname* de las VM**
 - **Modifica la variable *STUDENT_PREFIX* (línea 7)**
 - Debes sustituir **X** por tu **prefijo**
- **Modifica el inventario de Ansible en función del *hostname* de tus VM**



Fichero de inventario
modificado →

```
# Web application server
[web]
rre2425-web
# Database server
[db]
rre2425-db
# Group 'servers' with web and db
[servers:children]
web
db
```

- **Modifica la plantilla HTML que está en el directorio *webapp/views***
 - Abre el fichero *index.html* con un **editor de texto en tu equipo** para incluir tu nombre y apellidos
 - **Debes modificar únicamente la variable *name* (línea 9)**


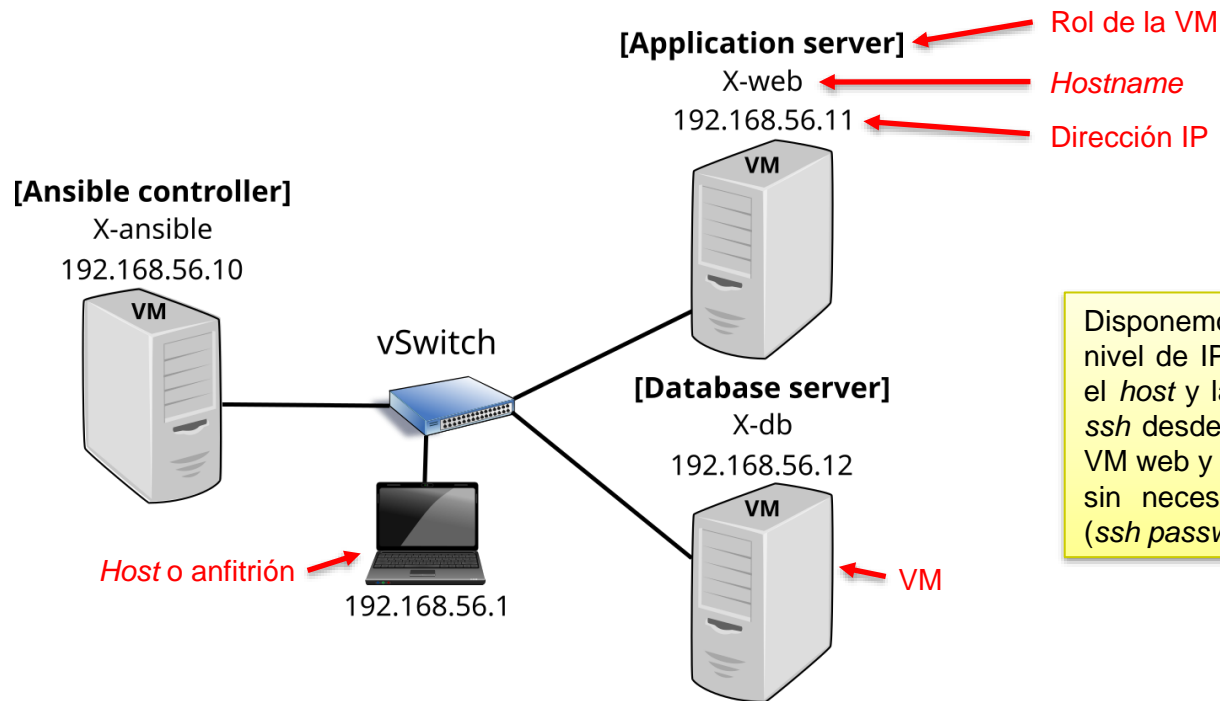




Ejercicio 1

9

- Despliega las VM con Vagrant y accede por *ssh* al controlador Ansible
 - *vagrant ssh ansible*
- Esquema gráfico del despliegue:



Disponemos de conectividad entre las VM a nivel de IP y de *hostname*, así como entre el *host* y las VM. Además, La conectividad *ssh* desde el controlador Ansible hacia las VM web y db está configurada para hacerse sin necesidad de introducir el *password* (*ssh passwordless*)



Ejercicio 1

10

- Desde el controlador Ansible, comprueba que funciona la conectividad *ssh passwordless* con las VM **web** y **db**

```
[vagrant@rre2425-ansible ~]$ ssh rre2425-web
Last login: Fri Feb  7 11:33:55 2025 from 192.168.56.10
[vagrant@rre2425-web ~]$ exit
logout
Connection to rre2425-web closed.
[vagrant@rre2425-ansible ~]$ ssh rre2425-db
Last login: Fri Feb  7 11:33:47 2025 from 192.168.56.10
[vagrant@rre2425-db ~]$ exit
logout
Connection to rre2425-db closed.
[vagrant@rre2425-ansible ~]$
```

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

```
[vagrant@rre2425-ansible ~]$ ansible --version
ansible [core 2.16.3]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/vagrant/.ansible/plugins/modules',
  ansible python module location = /usr/lib/python3.12/site-packages/ansible
  ansible collection location = /home/vagrant/.ansible/collections:/usr/share
  executable location = /usr/bin/ansible
  python version = 3.12.8 (main, Dec 12 2024, 16:30:29) [GCC 8.5.0 20210514]
  jinja version = 3.1.2
  libyaml = True
[vagrant@rre2425-ansible ~]$
```



Ejercicio 1

11

- Ejecuta el siguiente comando *ad-hoc* de Ansible usando el módulo **shell** para comprobar los *hostnames* y la versión del *kernel* de las VM **web** y **db**
 - *ansible servers -m shell -a "hostname && uname -a"*

Hostnames de las
VM correctamente
configurados →

```
[vagrant@rre2425-ansible ~]$ ansible servers -m shell -a "hostname && uname -a"
rre2425-web | CHANGED | rc=0 >>
rre2425-web
Linux rre2425-web 4.18.0-513.9.1.el8_9.x86_64 #1 SMP Wed Nov 29 18:55:19 UTC 2023 x86_64
rre2425-db | CHANGED | rc=0 >>
rre2425-db
Linux rre2425-db 4.18.0-513.9.1.el8_9.x86_64 #1 SMP Wed Nov 29 18:55:19 UTC 2023 x86_64
[vagrant@rre2425-ansible ~]$
```



Ejercicio 1

12

- Usando **comandos ad-hoc Ansible** ejecutados desde el controlador:
 - Instala el paquete **python3-mysqclient** en **web** y **db** (módulo [package](#))
 - Instala los paquetes **npm** y **nodejs** solo en **web** (módulo **package**)
 - Copia el directorio `/vagrant/webapp` del controlador en la ruta `/home/vagrant` de **web** (módulo [copy](#))
 - Establece **vagrant** como el propietario de usuario y de grupo
 - Instala las dependencias de la aplicación Node.js en **web** (módulo [npm](#))
 - Como parámetro **path** del módulo debemos indicar la ruta en la VM **web** donde reside el fichero `package.json` (`/home/vagrant/webapp`)
 - Conéctate por `ssh` a `web` y comprueba la instalación

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




```
[vagrant@rre2425-web ~]$ ls -l webapp/
total 52
-rw-r--r--. 1 vagrant vagrant 1407 Feb  5 16:08 app.js
drwxr-xr-x. 114 vagrant vagrant 4096 Feb  5 16:15 node_modules
-rw-r--r--. 1 vagrant vagrant 38438 Feb  5 16:15 package-lock.json
-rw-r--r--. 1 vagrant vagrant 189 Feb  5 16:14 package.json
drwxr-xr-x. 2 vagrant vagrant 38 Feb  5 16:08 views

[vagrant@rre2425-web ~]$ ls webapp/node_modules/
@babel      debug      gopd       mime
accepts     depd       has-symbols  mime-db
acorn       destroy    has-tostringtag  mime-types
acorn-flatten  doctypes   hasown        ms
asap        dunder-proto  http-errors    mysql
assert-never ee-first    iconv-lite     negotiator
babel-walk  encodeurl   inherits       object-assign
bignumber.js es-define-property  ipaddr.js     object-inspect
body-parser es-errors   is-core-module on-finished
bytes       es-object-atoms is-expression  parseurl
call-bind-apply-helpers escape-html  is-promise    path-parse
call-bound  etag        is-regex       path-to-regexp
character-parser  express    isarray        process-nextick-args
constantinople  finalhandler  js-stringify   promise
content-disposition  forwarded   jstransformer  proxy-addr
content-type    fresh       math-intrinsics  pug
cookie          function-bind media-typer      pug-attrs
cookie-signature get-intrinsic merge-descriptors  pug-code-gen
core-util-is    get-proto   methods         pug-error
```



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 una base de datos en **db** (módulo [mysql_db](#))
 - Puedes nombrar tu base de datos como quieras (parámetro **name** o **db** del módulo)
 - Crea en **db** el usuario de conexión a tu base de datos (módulo [mysql_user](#))
 - **Debes crear un usuario usando tu prefijo** (parámetro **name** del módulo) 
 - Establece una contraseña a tu usuario (parámetro **password**)
 - Dale todos los privilegios de acceso (parámetro **priv=db.*:ALL**) a tu base de datos
 - Debes sustituir **db** por el nombre que hayas usado al crear tu base de datos
 - Como parámetro **host** especifica el **hostname** apropiado para tu VM **web**
 - En este parámetro del módulo se especifica la máquina desde la que tu usuario se conectará a la base de datos, no la máquina que ejecuta la base de datos
 - Ejecuta la aplicación Node.js en **web** usando el módulo [shell](#)
 - `ansible web -b -m shell -a "node /home/vagrant/webapp/app.js" &`



Ejercicio 1

14

- Desde el controlador Ansible, accede a la VM que ejecuta la aplicación

```
[vagrant@rre2425-ansible ~]$ curl http://rre2425-web
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>GEI AISI</title>
  <script type="text/javascript">
    function getName() {
      var name = "Roberto Rey Expósito";
      var course = "2024/2025";
      document.getElementById("myName").innerHTML = name;
      document.getElementById("userId").value = name;
      document.getElementById("currentCourse").innerHTML = course;
      document.getElementById("courseId").value = course;
    }
    function getURL() {
      document.write(window.location.href);
    }
    function getTIME() {
      document.getElementById("current_date").innerHTML = Date();
    }
  </script>
</head>
<body onload="getName()">
  <div style="width:600px;height:650px;border:2px solid #000;text-align: center;">
    <strong>
      <p></p>
      <h3><u>GEI AISI: <span id="currentCourse"></u></h3>
      <p>
      <p><u>Node.js+Express+MariaDB WebApp (Ansible)</u></p>
      <p>Página web de <span id="myName"></span></p>
      <p><script>getURL();</script></p>
      <p><div id="current_date"><script>getTIME();</script></p>
```



Ejercicio 1

15



- Accede ahora desde el navegador de tu *host*: <http://X-web>

The screenshot shows a web browser window with the title 'GEI AISI'. The address bar shows 'rre2425-web'. The page content includes the logo of 'UNIVERSIDADE DA CORUÑA', the text 'GEI AISI: 2024/2025', and a logo for 'JS Express'. Below this, it says 'Node.js+Express+MariaDB WebApp (Ansible)'. A red box highlights the text 'Página web de Roberto Rey Expósito' and the URL 'http://rre2425-web/'. Below the red box, it shows the date and time 'Wed Feb 05 2025 20:06:50 GMT+0100 (hora estándar de Europa central)'. The main section is titled 'Database connection' and contains four input fields: 'User name:', 'Password:', 'DB hostname:', and 'DB name:'. A 'Connect' button is at the bottom.

Comprueba si funciona el acceso desde el navegador de tu *host* con estas URL:

<http://localhost>
<http://localhost:8080>
<http://192.168.56.10>
<http://192.168.56.11>



Prueba la conexión a tu base de datos usando tu usuario y *password*



Ejercicio 1



GEI AISI

rre2425-web/db

 UNIVERSIDADE DA CORUÑA

GEI AISI: 2024/2025



Node.js+Express+MariaDB WebApp (Ansible)

Página web de Roberto Rey Expósito

<http://rre2425-web/db>

Fri Feb 07 2025 12:51:52 GMT+0100 (hora estándar de Europa central)

MariaDB URI: `mysql://rre2425@rre2425-db:3306/fake_db`

MariaDB connection status from **user rre2425: OK**




¿Cómo interpretas el URI de conexión a la base de datos?

Si te no funciona la conexión, obtendrás un mensaje de error que te ayudará a conocer el motivo



Ejercicio 2

17

- **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 usar tu box y cambiar el hostname de la VM** 
 - Debes sustituir **X** por tu **prefijo**
 - Configura en el Vagrantfile un *provisioner* de tipo **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”

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



Ejercicio 2

18

- **Ficheros relevantes 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

Contenido del fichero
playbook.yml



```
---
- hosts: all
  become: yes

  vars_files:
    - vars/main.yml

  handlers:
    - ansible.builtin.import_tasks: handlers/main.yml
  tasks:
    - ansible.builtin.import_tasks: tasks/main.yml
```

- **files:** directorio que contiene los ficheros estáticos que sirve el sitio web
- **templates:** directorio que contiene
 - **apache.conf.j2:** plantilla Jinja con la configuración del sitio web
 - **index.html.j2:** plantilla Jinja con el código HTML de la vista



Ejercicio 2

19

- En el fichero ***vars/main.yml*** modifica las variables ***http_user***, ***http_host*** y ***http_conf***
 - Debes poner tu nombre y apellidos y sustituir **X** por tu **prefijo**
- En el fichero ***handlers/main.yml***
 - Usando el módulo **service**, define un manejador para reiniciar Apache (servicio **apache2**)
- En el fichero ***tasks/main.yml*** completa la definición de las **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 de destino: ***/var/www/{{ http_host }}/index.html***
 6. Usando el módulo **template**, copia el fichero ***templates/apache.conf.j2*** en la siguiente ruta de destino ***/etc/apache2/sites-enabled/{{ http_conf }}*** para habilitar el nuevo sitio web
 7. Usando el módulo **copy**, copia los ficheros estáticos existentes en el directorio ***files*** usando el **loop** predefinido y la siguiente ruta de destino: ***/var/www/{{ http_host }}/img/***
- En las tareas **2, 3, 5 y 6** notifica el **handler** para reiniciar el servicio de Apache



Ejercicio 2

20

- 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*** o ***vagrant provision*** para forzar el aprovisionamiento. Resulta útil durante el desarrollo y prueba de un playbook

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

Instalación de Ansible en la VM y ejecución del Playbook

```
PLAY [all] *****  
  
TASK [Gathering Facts] *****  
ok: [default]  
  
TASK [Install Apache web 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]  
  
TASK [Copy static files] *****  
changed: [default] => (item=apache.png)  
changed: [default] => (item=udc.png)  
  
RUNNING HANDLER [Restart apache] *****  
changed: [default]  
  
PLAY RECAP *****  
default : ok=9    changed=8    unreachable=0    failed=0    skipped=0
```



Ejercicio 2

21



- Conéctate por *ssh* a la VM
 - Comprueba los ficheros del sitio web: `ls -lR /var/www/X-vhost`
 - Accede al servidor web usando *curl*

Hostname correctamente
configurado →

```
vagrant@rre2425-ansible:~$ ls -lR /var/www/rre2425-vhost/
/var/www/rre2425-vhost/:
total 8
drwxr-xr-x 2 root root 4096 Feb  6 15:57 img
-rw-r--r-- 1 root root 869 Feb  6 15:57 index.html

/var/www/rre2425-vhost/img:
total 48
-rwxr-xr-x 1 root root 20903 Feb  6 15:57 apache.png
-rwxr-xr-x 1 root root 24371 Feb  6 15:57 udc.png
vagrant@rre2425-ansible:~$
vagrant@rre2425-ansible:~$ curl http://localhost
<html>
<head>
  <meta charset="UTF-8">
  <title>GEI AISI</title>
  <script type="text/javascript">
    function getURL() {
      document.write(window.location.href);
    }
    function getTime() {
      document.getElementById("current_date").innerHTML = Date();
    }
  </script>
</head>
<body>
  <div style="width:600px;height:350px;border:2px solid #000;text-align: center;">
    <strong>
      <p></p>
      <h3><u>GEI AISI: 2024/2025</u></h3>
      <p>
        <p><u>Apache Web Server (Ansible)</u></p>
        <p>Página web de Roberto Rey Expósito</p>
        <p>Virtual host: rre2425-vhost</p>
        <p><script>getURL();</script></p>
        <p><div id="current_date"><script>getTime();</script></p>
      </strong>
    </div>
  </body>
</html>
vagrant@rre2425-ansible:~$
```

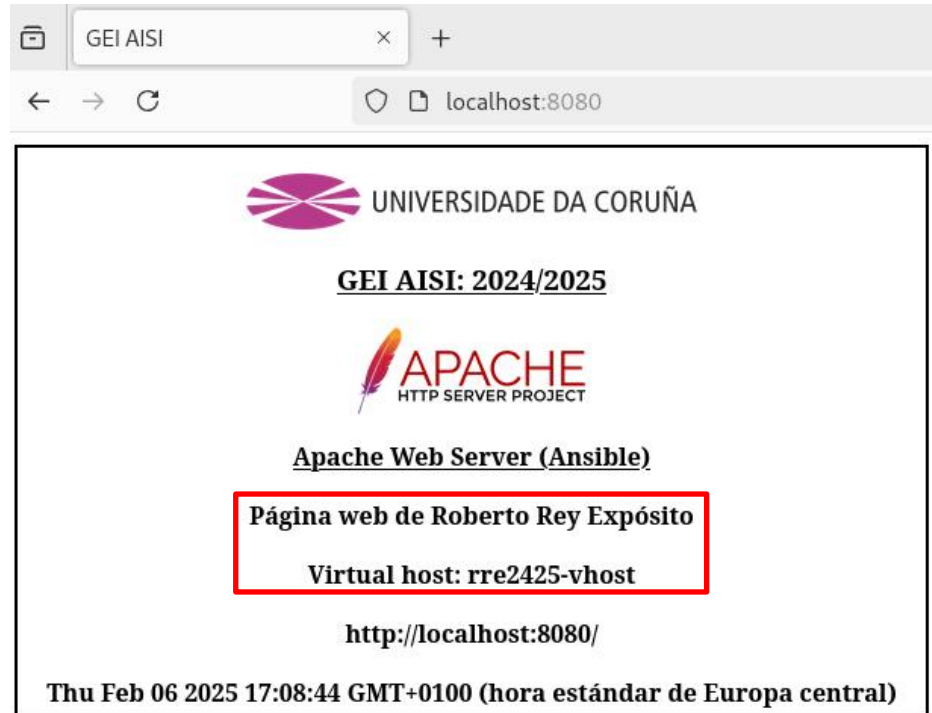


Ejercicio 2

22




- Accede desde el navegador de tu **host**


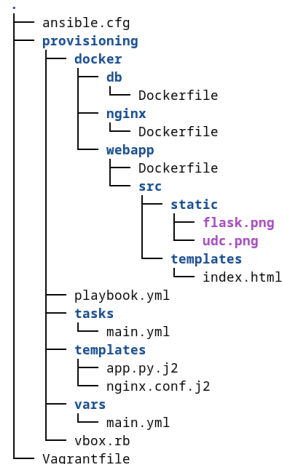




Ejercicio 3

23

- **Despliegue de una aplicación web Nginx+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 usar tu box y cambiar el hostname de la VM** 
 - Debes sustituir **X** por tu **prefijo**
 - Configura en el Vagrantfile un *provisioner* de tipo **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”



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



Ejercicio 3

24

- Ficheros relevantes 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: yes  
  
  vars_files:  
    - vars/main.yml  
  
  tasks:  
    - ansible.builtin.import_tasks: tasks/main.yml
```

- **templates:** directorio que contiene
 - **app.py.j2:** plantilla Jinja con el código Python de la aplicación web Flask
 - **nginx.conf.j2:** plantilla Jinja con la configuración de Nginx como **proxy inverso**
- **docker:** ficheros necesarios para crear las imágenes y contenedores Docker
 - **db:** directorio que contiene el *Dockerfile* para la imagen de la base de datos MySQL
 - **nginx:** directorio que contiene el *Dockerfile* para la imagen de Nginx
 - **webapp:** directorio que contiene el *Dockerfile* para la imagen de la aplicación web
 - **src/templates/index.html:** plantilla Jinja con la vista web que sirve la aplicación
 - **src/static:** directorio que contiene los ficheros estáticos que sirve la aplicación



Ejercicio 3

25

- En el fichero **vars/main.yml** modifica las variables **student_name** y **student_prefix**
 - Debes poner tu nombre y apellidos y sustituir **X** por tu **prefijo**
- En el fichero **tasks/main.yml** completa la definición de las **tareas**:
 - Usando el módulo **package**, instala la biblioteca "*Docker SDK for Python*" (paquete **python3-docker**)
 - Es una dependencia de los módulos de Ansible para Docker que usaremos después
 - Usando el módulo **copy**, copia el directorio **docker** en la ruta de destino: **/home/vagrant/**
 - Usando el módulo **template**, copia el fichero **templates/app.py.j2** en la siguiente ruta de destino: **/home/vagrant/docker/webapp/src/app.py**
 - Usando el módulo **template**, copia el fichero **templates/nginx.conf.j2** en la siguiente ruta de destino: **/home/vagrant/docker/nginx/nginx.conf**
 - Usando el módulo **docker image**, crea las imágenes Docker usando el **loop** predefinido
 - Usando el módulo **docker network**, crea una red de tipo **bridge** y nómbrala (parámetro **name**) usando la variable **container_network** definida en **vars/main.yml**



Ejercicio 3

26

- En el fichero **tasks/main.yml** completa la definición de las **tareas** (cont.):
 - Usando el módulo [docker_container](#), crea tres contenedores Docker para ejecutar la base de datos MySQL, la aplicación web Flask y Nginx como proxy inverso
 - Nombra tus contenedores (parámetro **name**) usando las variables definidas en *vars/main.yml*
 - Asegúrate que los contenedores se conecten **únicamente** a la red creada previamente (ver opción **networks_cli_compatible** del módulo)
 - Para el contenedor Nginx, publica el puerto 80 del contenedor en el puerto 80 de la VM



Ejercicio 3

27

- 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** o **vagrant provision** para forzar el aprovisionamiento. Resulta útil durante el desarrollo y prueba de un playbook

```
==> default: Running provisioner: bootstrap (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 [Copy docker directory] *****
changed: [default]

TASK [Copy app.py.j2 template] *****
changed: [default]

TASK [Copy nginx.conf template] *****
changed: [default]

TASK [Build Docker images from Dockerfiles] *****
changed: [default] => (item={'name': 'rre2425-web_img', 'directory': 'webapp'})
changed: [default] => (item={'name': 'rre2425-db_img', 'directory': 'db'})
changed: [default] => (item={'name': 'rre2425-nginx_img', 'directory': 'nginx'})

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

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

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

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

PLAY RECAP *****
default                : ok=10   changed=9   unreachable=0   failed=0   skipped=0
```



Ejercicio 3

28



- Conéctate por `ssh` a la VM y obtén información sobre:
 - Contenedores (comprueba su estado y sus nombres), imágenes y redes
 - Comprueba a qué redes están conectados los contenedores

```
vagrant@rre2425-ansible:~$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
3d2d2bffe311	rre2425-nginx_img	"/docker-entrypoint..."	24 seconds ago	Up 24 seconds	0.0.0.0:80->80/tcp	rre2425-nginx
d3c045e455d9	rre2425-web_img	"flask run"	25 seconds ago	Up 25 seconds	5000/tcp	rre2425-webapp
e33c94efa207	rre2425-db_img	"docker-entrypoint.s..."	26 seconds ago	Up 25 seconds	3306/tcp, 33060/tcp	rre2425-db

```
vagrant@rre2425-ansible:~$
```

```
vagrant@rre2425-ansible:~$ docker image ls
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
rre2425-nginx_img	latest	7e01ea02515c	32 seconds ago	192MB
rre2425-db_img	latest	ee7033977def	39 seconds ago	797MB
rre2425-web_img	latest	c549550ff631	52 seconds ago	608MB
nginx	1.27.4	97662d24417b	4 days ago	192MB
ubuntu	jammy	a24be041d957	2 weeks ago	77.9MB
mysql	9.2.0	3e34946bc4c4	2 weeks ago	797MB

```
vagrant@rre2425-ansible:~$
```

```
vagrant@rre2425-ansible:~$ docker network ls
```

NETWORK ID	NAME	DRIVER	SCOPE
ca0251fee21d	bridge	bridge	local
941b952803d8	host	host	local
3c66a128cc88	none	null	local
5ed122f00555	rre2425-network	bridge	local

```
vagrant@rre2425-ansible:~$
```

```
vagrant@rre2425-ansible:~$ docker inspect --format "{{.NetworkSettings.Networks}}" rre2425-db
```

```
map[rre2425-network:0xc0000e8e00]
```

```
vagrant@rre2425-ansible:~$ docker inspect --format "{{.NetworkSettings.Networks}}" rre2425-webapp
```

```
map[rre2425-network:0xc0000e8e00]
```

```
vagrant@rre2425-ansible:~$ docker inspect --format "{{.NetworkSettings.Networks}}" rre2425-nginx
```

```
map[rre2425-network:0xc0000e8e00]
```

```
vagrant@rre2425-ansible:~$
```

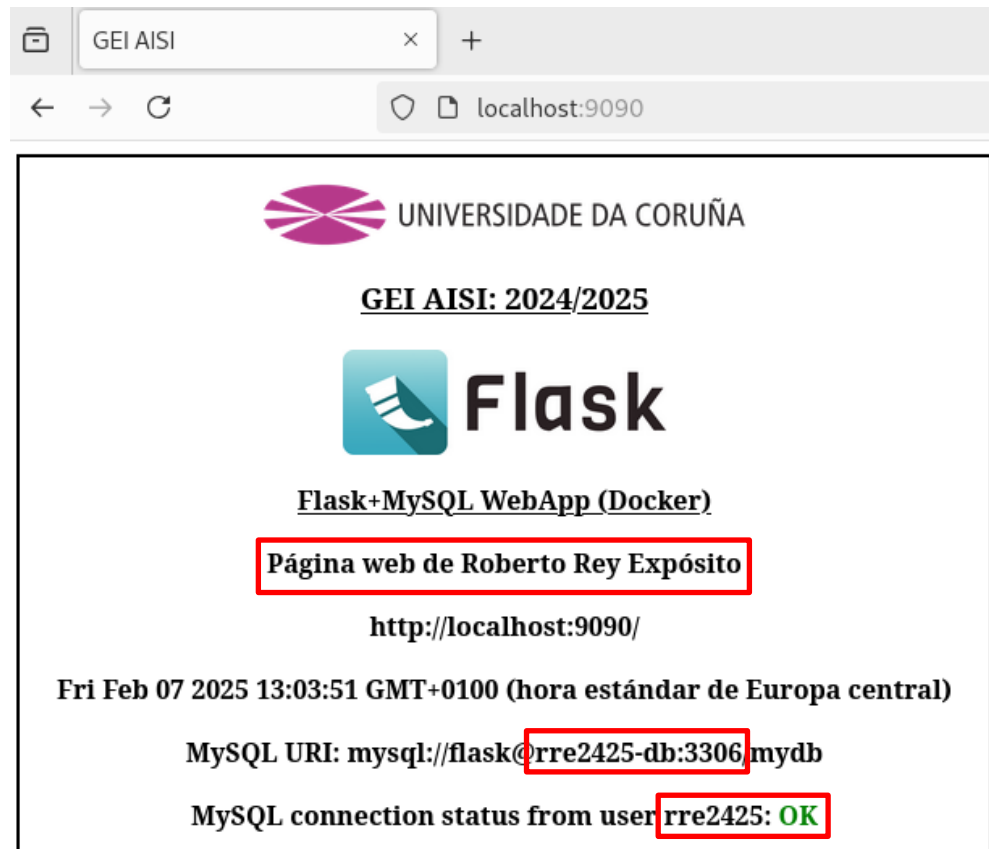


Ejercicio 3

29



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





Ejercicio 3

30

- 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@rre2425-ansible:~$ docker logs rre2425-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: 263-183-265
172.18.0.4 - - [10/Feb/2025 09:44:42] "GET / HTTP/1.0" 200 -
172.18.0.4 - - [10/Feb/2025 09:44:42] "GET /static/udc.png HTTP/1.0" 304 -
172.18.0.4 - - [10/Feb/2025 09:44:42] "GET /static/flask.png HTTP/1.0" 304 -
172.18.0.4 - - [10/Feb/2025 09:45:19] "GET / HTTP/1.0" 200 -
172.18.0.4 - - [10/Feb/2025 09:45:19] "GET /static/udc.png HTTP/1.0" 304 -
172.18.0.4 - - [10/Feb/2025 09:45:19] "GET /static/flask.png HTTP/1.0" 304 -
172.18.0.4 - - [10/Feb/2025 09:45:21] "GET / HTTP/1.0" 200 -
172.18.0.4 - - [10/Feb/2025 09:45:21] "GET /static/udc.png HTTP/1.0" 304 -
172.18.0.4 - - [10/Feb/2025 09:45:21] "GET /static/flask.png HTTP/1.0" 304 -
vagrant@rre2425-ansible:~$
```



Determina a quién pertenecen las IPs resaltadas (en tu caso podrían ser distintas)

- Desde la **VM**, comprueba con **curl** si funciona el acceso con estas URL:

- curl <http://localhost>
- curl <http://localhost:5000>
- curl 172.18.0.3
- curl 172.18.0.3:5000
- curl 172.18.0.4



Utiliza las IPs correspondientes en tu caso



Ejercicio 4

31

- **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 Linux que ejecuten MySQL y Apache, aunque también es posible usar otras alternativas como 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 usar tu *box* y cambiar el *hostname* de la VM** 
 - Debes sustituir **X** por tu **prefijo**

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



Ejercicio 4

32

- **Ficheros relevantes del ejercicio:**

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

- **playbook.yml:** Playbook que define los **roles** de Ansible y usa los 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:
    - ansible.builtin.import_tasks: tasks/main.yml
```

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



Ejercicio 4

33

- Completa el fichero **requirements.yml** para definir los roles de Ansible necesarios para el despliegue del entorno LAMP
 - Especifica de **forma explícita la versión** del rol (usa la última disponible en Galaxy)
 - Ver transparencia 67 del tutorial de Ansible
 - <https://galaxy.ansible.com/ui/standalone/roles/geerlingguy/apache>
 - <https://galaxy.ansible.com/ui/standalone/roles/geerlingguy/mysql>
 - <https://galaxy.ansible.com/ui/standalone/roles/geerlingguy/php>
 - <https://galaxy.ansible.com/ui/standalone/roles/geerlingguy/php-mysql>
 - <https://galaxy.ansible.com/ui/standalone/roles/geerlingguy/firewall>
- Configura en el *Vagrantfile* un *provisioner* de tipo **ansible_local**:
 - Configura el modo de instalación y el *playbook* a ejecutar como en ejercicios previos
 - Adicionalmente, configura la descarga de los roles desde Ansible Galaxy y su instalación automática en la VM
 - Debes indicar la ruta al fichero que define los roles (**requirements.yml**)
 - Debes instalar los roles en la ruta **/home/vagrant/roles** de la VM
 - Consulta las opciones **galaxy_role_file** y **galaxy_roles_path** del *provisioner*
 - https://developer.hashicorp.com/vagrant/docs/provisioning/ansible_common



Ejercicio 4

34

- En el fichero **tasks/main.yml** completa la definición de las **tareas**:
 1. Usando el módulo **package**, instala WordPress (paquete **wordpress**)
 - Todas las dependencias de WordPress, así como su configuración, están resueltas gracias al uso de los roles de Ansible que se ejecutan previamente en el Playbook
 2. Usando el módulo **template**, copia el fichero **templates/config-default.php.j2** en la siguiente ruta de destino: **/etc/wordpress/config-default.php**
 3. Usando el módulo **mysql_db**, crea la base de datos MySQL para WordPress
 - Nombra la base de datos usando la variable **db_name** definida en **vars/main.yml**
 4. Usando el módulo **mysql_user**, crea un usuario para la base de datos
 - Nombra el usuario usando la variable **db_user** definida en **vars/main.yml**
 - Como contraseña del usuario usa la variable **db_user_passwd**
 - Dale al usuario todos los privilegios de acceso (parámetro **priv**) en la base de datos creada previamente (ver ejercicio 1)
 - Como parámetro **host** especifica **localhost** dado que en este caso la base de datos se ejecuta en la misma VM desde la que se accede a la misma



Ejercicio 4

35

- 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: bootstrap (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/2.6.0.tar.gz
- extracting geerlingguy.firewall to /home/vagrant/roles/geerlingguy.firewall
- geerlingguy.firewall (2.6.0) was installed successfully
- downloading role 'mysql', owned by geerlingguy
- downloading role from https://github.com/geerlingguy/ansible-role-mysql/archive/4.3.4.tar.gz
- extracting geerlingguy.mysql to /home/vagrant/roles/geerlingguy.mysql
- geerlingguy.mysql (4.3.4) was installed successfully
- downloading role 'apache', owned by geerlingguy
- downloading role from https://github.com/geerlingguy/ansible-role-apache/archive/4.0.0.tar.gz
- extracting geerlingguy.apache to /home/vagrant/roles/geerlingguy.apache
- geerlingguy.apache (4.0.0) 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/6.0.0.tar.gz
- extracting geerlingguy.php to /home/vagrant/roles/geerlingguy.php
- geerlingguy.php (6.0.0) was installed successfully
default: Running ansible-playbook...
```

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

Inicio de la ejecución
del Playbook y sus
roles

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

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

```
TASK [geerlingguy.firewall : Flush iptables the first time playbook runs.] *****
changed: [default]
```

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

Fin de la ejecución
del Playbook

```
PLAY RECAP *****
default                : ok=96   changed=32   unreachable=0   failed=0   skipped=46
```



Ejercicio 4

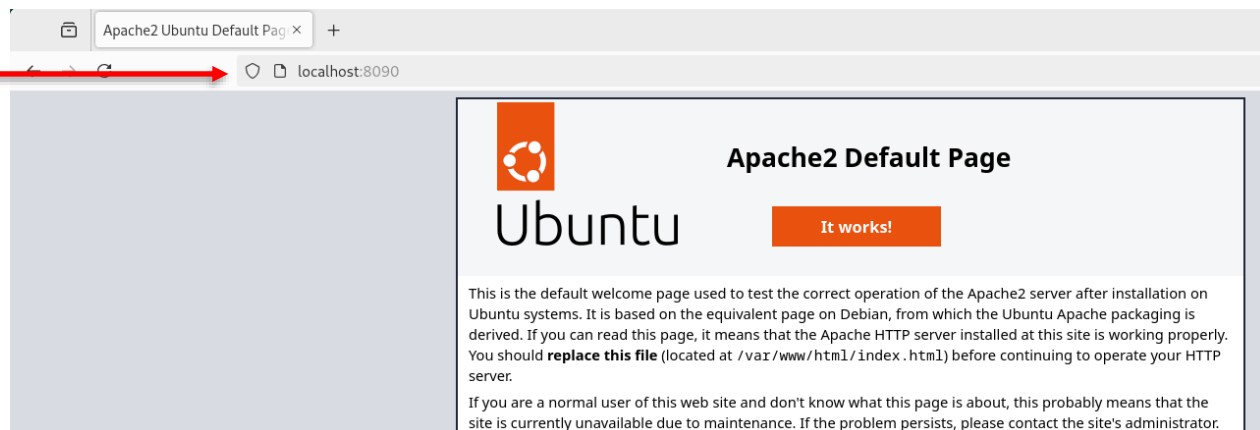
36

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

```
vagrant@rre2425-ansible:~$ pwd
/home/vagrant
vagrant@rre2425-ansible:~$ ls -la roles/
total 28
drwxrwxr-x 7 vagrant vagrant 4096 Feb  7 10:54 .
drwxr-x--- 6 vagrant vagrant 4096 Feb  7 10:54 ..
drwxrwxr-x 10 vagrant vagrant 4096 Feb  7 10:54 geerlingguy.apache
drwxrwxr-x 9 vagrant vagrant 4096 Feb  7 10:54 geerlingguy.firewall
drwxrwxr-x 10 vagrant vagrant 4096 Feb  7 10:54 geerlingguy.mysql
drwxrwxr-x 10 vagrant vagrant 4096 Feb  7 10:54 geerlingguy.php
drwxrwxr-x 8 vagrant vagrant 4096 Feb  7 10:54 geerlingguy.php-mysql
vagrant@rre2425-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

37

- Accede ahora al sitio web de WordPress para proceder a su configuración e instalación inicial: <http://localhost:8090/wordpress>

The screenshot shows the WordPress installation page at `localhost:8090/wordpress/wp-admin/install.php`. The page has a light gray background with the WordPress logo at the top right. A yellow warning triangle with a black exclamation mark is on the left. Red text annotations with arrows point to specific fields: 'Usa tu prefijo X en: Site Title: X-wordpress Username: X' points to the 'Site Title' and 'Username' fields; 'Pon cualquier password que recuerdes' points to the 'Password' field; 'Puedes usar cualquier email ficticio siempre que cumpla el formato esperado' points to the 'Your Email' field. The form fields are: 'Site Title' (value: rre2425-wordpress), 'Username' (value: rre2425), 'Password' (masked with dots, strength indicator 'Strong', and a 'Show' button), 'Your Email' (value: aisi@udc.es), and 'Search engine visibility' (checked checkbox). An 'Install WordPress' button is at the bottom.

WordPress > Installation

localhost:8090/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: rre2425-wordpress

Username: rre2425

Use any alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

Password: [masked] Strong [Show]

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

Your Email: aisi@udc.es

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

Usa tu prefijo X en:
Site Title: X-wordpress
Username: X

Pon cualquier password que recuerdes

Puedes usar cualquier email ficticio siempre que cumpla el formato esperado



Ejercicio 4

38



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



Ejercicio 4

39



- Aspecto por defecto del sitio web

