

UNIVERSIDAD INTERNACIONAL DE LA RIOJA (UNIR)

La Universidad en Internet

*Herramientas de Automatización de Despliegues
(MEXDEVOPS)*

**Maestría en Desarrollo y Operaciones de Software
Febrero 2025**

Actividad 3: Desarrollo práctico

Luis Enrique Méndez Cantero

Índice

Objetivos	3
Descripción de la actividad.....	3
Estructura del proyecto.....	3
Desarrollo	4
Configuración del Vagrantfile	4
Playbook de Ansible.....	5
Tareas del Rol	5
Variables del partido (id del primer jugador)	6
Ejecución	6
Resultado	6
Conclusiones.....	7
Referencias	7

Objetivos

Esta actividad práctica se centra en desarrollar un sistema de generación automática y aleatoria de partidos de tenis utilizando Vagrant y Ansible. Este ejercicio tiene como finalidad afianzar los conocimientos sobre la automatización de despliegues y la gestión de configuraciones, así como el uso de variables en diferentes ubicaciones dentro de Ansible.

- Desarrollar un trabajo práctico con Vagrant y Ansible.
- Aprender a desarrollar playbooks y roles en Ansible.
- Afianzar el aprendizaje de diferentes ubicaciones de las variables en Ansible.

Descripción de la actividad

La Universidad Internacional de La Rioja (UNIR) organizará un partido de tenis, para lo cual es necesario implementar un sistema que genere automáticamente los datos de dos tenistas. Al ejecutar el comando `vagrant up`, se debe crear el archivo `partido.txt` en la ubicación `/vagrant/partido.txt`, mostrando en pantalla la información de los dos jugadores y la fecha del partido.

Estructura del proyecto

Directorio/Archivo	Descripción
<code>/tennis_match/</code>	Carpeta raíz del proyecto
— <code>Vagrantfile</code>	Archivo de configuración para Vagrant
— <code>ansible/</code>	Carpeta con los scripts de Ansible
— <code>playbook.yml</code>	Playbook principal de Ansible
— <code>roles/</code>	Roles de Ansible
— <code>match_generator/</code>	Rol para la generación de partidos

Directorio/Archivo	Descripción
└─ tasks/	Definición de tareas del rol
└─ main.yml	Tareas principales del rol
└─ vars/	Variables utilizadas en el rol
└─ partido_vars.yml	Variables del partido
└─ output/	Carpeta de salida de datos
└─ partido.txt	Archivo generado automáticamente

Desarrollo

Configuración del Vagrantfile

Se utiliza Ubuntu 18.04 como la máquina virtual base, con Ansible instalado dentro de la VM para gestionar la automatización, también se sincronizarán los folders para mapear folders de la computadora host (Windows 11) con la máquina que va a ser creada en Virtual Box.

```
Vagrant.configure("2") do |config|
  config.vm.box = "ubuntu/bionic64"

  # Instalar Ansible dentro de la máquina virtual
  config.vm.provision "shell", inline: <<-SHELL
    sudo apt-get update
    sudo apt-get install -y ansible python3-pip
  SHELL

  # Asegurar que ansible y output están sincronizados
  config.vm.synced_folder "./ansible", "/home/vagrant/ansible"
  config.vm.synced_folder "./output", "/vagrant"

  # Correr el playbook desde la máquina virtual
  config.vm.provision "shell", inline: <<-SHELL
```

```
    ansible-playbook /home/vagrant/ansible/playbook.yml
  SHELL
end
```

Playbook de Ansible

Definirá la ejecución del rol para generar la información del partido.

```
---
- name: Generar informacion de partido de Tennis
  hosts: localhost
  connection: local
  become: yes
  vars:
    first_name: "Luis"
    last_name: "Mendez"
    city: "Puebla"
  roles:
    - match_generator
```

Tareas del Rol

El rol maneja la obtención de datos estáticos definidos en nuestro repositorio y también dinámicos obtenidos desde una API web que genera información random y está a nuestra disposición en formato JSON.

```
---
- name: Leer ID de partido_vars.yml
  include_vars:
    file: "../vars/partido_vars.yml"

- name: Obtener informacion para el segundo jugador
  uri:
    url: "https://random-data-api.com/api/users/random_user?size=1"
    return_content: yes
  register: json_response

- name: Generar partido.txt
  copy:
    dest: "/vagrant/partido.txt"
    content: |
      ** INFORMACION DEL PARTIDO UNIR **
```

```
First Name: {{ first_name }}.
Last Name: {{ last_name }}.
City: {{ city }}.
id: {{ id }}.

First Name: {{ (json_response.json[0].first_name) }}.
Last Name: {{ (json_response.json[0].last_name) }}.
City: {{ (json_response.json[0].address.city) }}.
id: {{ (json_response.json[0].id) }}.

Fecha: {{ ansible_date_time.iso8601 }}.
```

Variables del partido (id del primer jugador)

Como parte del ejercicio crearemos un archivo yml con el id del primer jugador (el cual es estático) y que es usado por el rol.

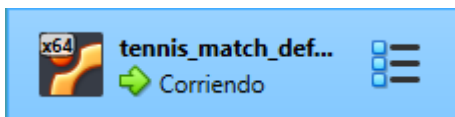
```
---
id: 1234
```

Ejecución

Para ver los resultados necesitamos correr el siguiente comando:

vagrant up

Si todo funciona correctamente la máquina será creada y se podrá ver en la aplicación de Virtual Box así:



Resultado

Tras la ejecución de vagrant up, se genera el archivo partido.txt con el siguiente contenido:

** INFORMACION DEL PARTIDO UNIR **

First Name: Luis.

Last Name: Mendez.

City: Puebla.

id: 1234.

First Name: Bret.

Last Name: Friesen.

City: West Mathildaside.

id: 4443.

Fecha: 2025-02-07T16:56:49Z.

Conclusiones

Esta actividad permitió comprender de forma práctica el uso de Vagrant para la gestión de máquinas virtuales y Ansible para la automatización de configuraciones. Además, se reforzó el manejo de variables en diferentes contextos y la integración de APIs externas para obtener datos dinámicos.

Referencias

1. Vagrant by HashiCorp. (2023). *Vagrant Documentation*. Recuperado de <https://www.vagrantup.com/docs>
2. Ansible by Red Hat. (2023). *Ansible Documentation*. Recuperado de <https://docs.ansible.com/>
3. Random Data API. (2023). *User API Documentation*. Recuperado de <https://random-data-api.com/documentation>