

Desafio # 6

Fecha de entrega: 30/07/2025

Objetivo:

Este desafío tiene como objetivo implementar un playbooks de Ansible y Roles para cada servicio.

Podremos desplegar un nuevo EC2 en AWS Academy de ser necesario para utilizar una familia de Ubuntu, **importante** usar la misma PEM para facilitar el acceso.

Utilizar el README de Sandbox como ayuda para acceder a la instancia de EC2 y configurar el *inventory.ini*

Escenario:

Nuestro equipo de trabajo recientemente inició un nuevo sprint y nos asignaron una tarea muy importante para modularizar nuestro proyecto de *configuration management*, este sprint tiene como fecha de cierre el 06/08/2025 y debemos tener una versión funcional el día 30/07/2025 para la reunión de demos.

Este proyecto utiliza como herramienta de *configuration as a code* Ansible y permite instalar y desplegar nuestro sitio web de Wordpress en un host EC2 que es utilizaremos a demanda en un laboratorio de AWS Academy.

Como estamos trabajando en un entorno de Demo, necesitamos instalar PHP junto con sus componentes y MySQL dentro de una instancia EC2, no es necesario para este proyecto una base de datos RDS.

Requisitos:

1. Utilizar un Bastion de AWS Academy
2. Identificar y crear posibles variables que se puedan reutilizar.
3. Crear un archivo playbook y roles específicos para PHP y MySQL
4. Testar y validar que todo funciona.

Proyecto ejemplo: <https://medium.com/how-to-install-wordpress-on-ubuntu-a-step-by-step-guide>

Entregables:

Los entregables establecidos para este proyecto con:

1. Código fuente del playbook de Ansible publicado en un repositorio de Github.
2. Guía detallada de cómo utilizar el Role publicado en el archivo README.md del repositorio.
3. Evidencia de las pruebas con resultado exitoso.

Evaluacion:

- Entrega en fecha.
- Redactar documentación legible y que sea comprendida por terceros.
- Añade material de soporte adicional.
 - Ejemplo: Diagrama de alto nivel.
- Cumple con las consignas solicitadas.
- Utilizar un archivo *playbook.yaml* y una carpeta role para separar cada servicio
- El entregable es funcional.
 - Ejemplo: repositorio con los archivos necesarios
 - Estructura de archivos (comando *tree* de Linux)