

# Actividad: Gestión de inventarios y archivos de configuración

## Escenario del Proyecto "TechCorp"

Sais los administradores de sistemas de "TechCorp". La empresa tiene una infraestructura mixta con entornos de **Desarrollo** y **Producción** conviviendo en la misma red. Se os ha pedido reestructurar la configuración de Ansible para cumplir con los siguientes requisitos complejos:

1. **Node1** será el servidor de Base de Datos (debe administrarse como `root`).
2. **Node2** será un servidor Web de "Desarrollo".
3. **Node3** será un servidor Web de "Producción".
4. No se permiten contraseñas en el archivo de inventario principal (seguridad por oscuridad básica).
5. Las variables deben estar separadas del inventario (uso de `group_vars` y `host_vars`).

## Fase 1: Inventario YAML Estructurado (25 min)

Crea un archivo llamado `hosts.yml`. Debes definir los hosts `node1`, `node2` y `node3` pero asignándoles **alias funcionales** y organizándose en una matriz lógica:

- **Grupos Funcionales:**
  - `db_servers`: Contiene a `node1`.
  - `web_servers`: Contiene a `node2` y `node3`.
- **Grupos de Entorno:**
  - `dev`: Contiene a `node2`.
  - `prod`: Contiene a `node1` y `node3`.
- **Grupo Geográfico:**
  - `madrid`: Contiene a todos los servidores (usando el grupo especial `all`).
- El `node1` debe definirse en el inventario con un alias, por ejemplo `db_master`, de forma que `ansible_host` apunte a `node1`.

## Fase 2: Desacople de Variables

Una buena práctica es no ensuciar el archivo de inventario con variables. Vamos a usar la estructura de directorios nativa de Ansible.

1. **Variables Globales (`group_vars/all.yml`):**

- Define aquí el usuario de conexión genérico (`ansible`) y la contraseña (`password`).
  - Define una variable `empresa`: TechCorp.
2. **Variables Específicas de Web (`group_vars/web_servers.yml`):**
    - Define `http_port: 80`.
    - Define `service_name: apache2`.
  3. **Excepción de Seguridad (`host_vars/db_master.yml`):**
    - Aquí sobreescribiremos la conexión global. Configura este archivo para que la conexión a este host específico se haga con el usuario `root`
    - Define `db_port: 5432`.

## Fase 3: Configuración y Tuning

Configura el archivo `ansible.cfg` que creaste al principio para orquestar todo esto.

**Tarea:** Edita `ansible.cfg` con las siguientes directivas:

1. Apuntar al inventario `hosts.yml` por defecto.
2. Desactivar `host_key_checking`.
3. Cambiar el formato de salida a `yaml` para leer mejor los resultados.
4. Habilitar el log de las operaciones en `/var/log/ansible.log`

## Fase 4: Patrones de Selección y Verificación (20 min)

Ahora probaremos la potencia de los "Patterns" de Ansible para atacar intersecciones de grupos.

**Ejecuta y documenta los siguientes comandos:**

1. **Prueba de variables:** Usa el módulo `debug` para ver qué puerto usaría cada servidor y muestra un mensaje con nombre del servicio y el puerto, para los servidores web:  
  
*(Verifica que Node1 NO responde a esto y que Node2/3 muestran los datos correctos).*
2. **La intersección (AND):** Queremos hacer ping **solo** a los servidores Web que además son de Producción. *Pista: La sintaxis es `grupo1:&grupo2`.*  
 Bash  
*(Debería responder SOLO el node3).*
3. **La exclusión (NOT):** Queremos listar las variables de todos los servidores que **NO** son de base de datos.

4. **Verificación de privilegios:** Comprueba que `node1` está conectando como root y los demás como ansible.

## 7. Entregable Final

El alumno debe subir un archivo comprimido con:

1. La estructura de directorios (`group_vars`, `host_vars`, `ansible.cfg`, `hosts.yml`).
2. Un archivo de texto con la demostración que todo lo pedido funciona correctamente