Ansible: Automatización de TI, Comparativa de Herramientas y su Uso con Podman

Introducción a Ansible

Ansible se presenta como una aplicación de software de automatización de TI de código abierto, operada mediante línea de comandos y desarrollada en Python. Su capacidad principal reside en la configuración de sistemas, el despliegue de software y la orquestación de flujos de trabajo complejos para respaldar la implementación de aplicaciones, la realización de actualizaciones de sistemas y una variedad de otros procesos informáticos.[1] Esta herramienta se ha consolidado como una solución estándar dentro del ámbito de la automatización empresarial, ofreciendo funcionalidades para la gestión de la configuración, el aprovisionamiento de infraestructura y el despliegue de aplicaciones en diversos entornos, que abarcan desde la nube híbrida hasta infraestructuras locales e incluso el Internet de las Cosas (IoT).[3] La flexibilidad y potencia de Ansible permiten a los profesionales de TI alcanzar una mayor eficiencia y consistencia en sus operaciones.[3]

Una de las características distintivas de Ansible es su arquitectura sin agentes. A diferencia de otras herramientas de gestión de configuración, Ansible no requiere la instalación de software o agentes en los nodos gestionados. En su lugar, se comunica con estos sistemas de forma temporal a través de conexiones remotas, utilizando SSH para sistemas basados en Unix y Windows Remote Management (WinRM) para sistemas Windows, lo que permite la ejecución de comandos de PowerShell.[4] Esta aproximación reduce la sobrecarga y la complejidad en los sistemas administrados, simplificando la implementación y el mantenimiento de la automatización.

La simplicidad y la facilidad de uso son puntos fuertes de Ansible. Los playbooks, que son la base de la automatización en Ansible, se escriben en YAML, un lenguaje de serialización de datos diseñado para ser legible por humanos.[2] Esta elección de lenguaje facilita la creación y comprensión de los scripts de automatización, incluso para aquellos que no tienen una formación exhaustiva en programación.[2] La comunidad de Ansible, respaldada por Red Hat, contribuye activamente al proyecto, lo que se traduce en una amplia gama de módulos y una documentación exhaustiva.[3]

La arquitectura de Ansible se fundamenta en la interacción entre un nodo de control y los nodos gestionados.[2] El nodo de control es el sistema donde se instala Ansible y desde donde se ejecutan los playbooks y comandos. Los nodos gestionados son los sistemas que se desean automatizar. La comunicación entre el nodo de control y los nodos gestionados se establece a través de la información definida en un archivo de

inventario.[3] Este inventario puede ser estático, con los hosts definidos en un archivo de texto plano (en formatos INI o YAML), o dinámico, obteniendo la información de fuentes externas como proveedores de nube o sistemas de gestión de configuración.[5] Esta flexibilidad en la gestión del inventario permite a Ansible adaptarse a infraestructuras diversas y cambiantes, incluyendo entornos en la nube.[5]

Las funcionalidades principales de Ansible abarcan un amplio espectro de tareas de automatización de TI.[4] Permite el aprovisionamiento de nueva infraestructura, configurando servidores, redes y otros componentes. Facilita la gestión de la configuración de sistemas existentes, asegurando que se mantengan en el estado deseado mediante la instalación de software, la modificación de archivos de configuración y la gestión de servicios.[4] Ansible también se utiliza para el despliegue de aplicaciones, coordinando los pasos necesarios para instalar y configurar software en múltiples sistemas.[1] Además, Ansible es capaz de orquestar flujos de trabajo complejos que involucran múltiples sistemas y tareas, como la implementación de aplicaciones de varios niveles o la realización de actualizaciones coordinadas.[1]

Si bien la consulta se centra en la versión de código abierto de Ansible, es importante mencionar que Red Hat ofrece Red Hat Ansible Automation Platform, una solución comercial que amplía las capacidades de Ansible con características de nivel empresarial.[8] Esta plataforma incluye una interfaz de usuario web (Automation Controller, anteriormente Ansible Tower), gestión centralizada, herramientas de análisis y contenido certificado, lo que facilita la adopción y escalabilidad de la automatización en organizaciones de mayor tamaño.[10]

En el núcleo de la ejecución de tareas en Ansible se encuentran los módulos. Estos son pequeños fragmentos de código reutilizables que Ansible envía y ejecuta en los nodos gestionados para realizar acciones específicas, como instalar paquetes, gestionar archivos o configurar servicios.[2] Una característica importante de muchos módulos de Ansible es su idempotencia. Esto significa que un módulo puede ejecutarse varias veces, pero solo realizará cambios en el sistema si es necesario para alcanzar el estado deseado.[2] Esta propiedad garantiza que la automatización sea segura y predecible, evitando cambios no intencionados en los sistemas.

Ansible Playbook: El Motor de la Automatización

Los Ansible Playbooks son la piedra angular de la automatización con Ansible. Se definen como archivos en formato YAML que describen el estado deseado de los sistemas y orquestan la ejecución de tareas para alcanzar ese estado.[6] Estos playbooks se utilizan para una amplia gama de propósitos, incluyendo la gestión de la

configuración de sistemas, el despliegue de aplicaciones y la automatización de flujos de trabajo de TI complejos que involucran múltiples hosts.[6] Su diseño permite la definición de secuencias de acciones complejas de una manera estructurada y repetible.[13]

Los playbooks encuentran aplicación en diversos escenarios. Pueden automatizar tareas rutinarias como la actualización de paquetes en servidores, la instalación y configuración de software (como servidores web o bases de datos), la gestión de cuentas de usuario, la configuración de firewalls y la realización de actualizaciones progresivas en entornos de producción sin interrumpir el servicio.[6] La capacidad de definir un conjunto de tareas y ejecutarlas de manera consistente en múltiples sistemas convierte a los playbooks en una herramienta esencial para la gestión eficiente de infraestructuras de TI.[14]

La sintaxis de un playbook se basa en el formato YAML, que se caracteriza por su legibilidad y estructura jerárquica.[6] Un playbook típicamente comienza con la directiva --- y a menudo termina con ....[15] Está compuesto por una o más "plays", donde cada play se dirige a un conjunto específico de hosts definidos en el inventario y contiene una lista de tareas a ejecutar.[13]

Dentro de la estructura de un playbook, se identifican varios componentes clave.[15] La sección hosts especifica los servidores o grupos de servidores del inventario donde se ejecutarán las tareas definidas en el play.[6] La sección tasks enumera las acciones que se llevarán a cabo en los hosts de destino. Cada tarea invoca un módulo de Ansible y puede incluir parámetros específicos para ese módulo.[6] Opcionalmente, se puede incluir una sección vars para definir variables que se pueden utilizar para personalizar el comportamiento del playbook.[15] Los handlers son tareas especiales que solo se ejecutan cuando son notificadas por otras tareas, generalmente después de que una tarea ha realizado un cambio en el sistema (por ejemplo, reiniciar un servicio después de actualizar su configuración).[4] Finalmente, los roles son unidades de contenido de Ansible reutilizables que permiten organizar y compartir código de automatización de manera más eficiente.[3]

La ejecución de un playbook se realiza mediante el comando ansible-playbook, seguido del nombre del archivo del playbook.[3] Este comando interpreta el playbook y ejecuta las tareas definidas en secuencia en los hosts especificados. Ansible proporciona varias opciones para el comando ansible-playbook que permiten controlar su comportamiento. Por ejemplo, la opción --syntax-check permite verificar la sintaxis del playbook sin ejecutarlo, lo que ayuda a identificar errores antes de que afecten a los sistemas.[3] La opción --check o -C realiza una "ejecución en seco" del

playbook, simulando los cambios que se realizarían pero sin aplicarlos realmente, lo que resulta útil para previsualizar el impacto de la automatización.[3] La opción --verbose aumenta la cantidad de información mostrada durante la ejecución, lo que puede ser útil para la depuración.[16]

Existen numerosos ejemplos de playbooks, desde los más básicos hasta los más avanzados.[3] Un ejemplo sencillo podría ser un playbook que simplemente imprime un mensaje en los hosts gestionados.[18] Ejemplos más complejos incluyen playbooks para actualizar todos los paquetes de un sistema, instalar y configurar un servidor web como Apache o Nginx, gestionar cuentas de usuario en múltiples servidores o desplegar aplicaciones completas.[6] Repositorios en línea como ansible/ansible-examples en GitHub [19] ofrecen una amplia colección de playbooks de inicio que demuestran diversas características y casos de uso de Ansible. Estos ejemplos son un recurso valioso para aprender a escribir y utilizar playbooks en diferentes escenarios de automatización.

Ansible Navigator: Una Interfaz Moderna para Ansible

ansible-navigator se presenta como una utilidad de interfaz de línea de comandos (CLI) diseñada para interactuar con Ansible Core y los entornos de ejecución de automatización.[10] Su propósito principal es proporcionar una interfaz basada en texto que facilite la validación de contenido de Ansible y ofrezca retroalimentación directa a los usuarios.[10] La existencia de laboratorios interactivos proporcionados por Red Hat para aprender a utilizar ansible-navigator subraya su creciente importancia dentro del ecosistema de Ansible.[8]

Entre las características principales de ansible-navigator se encuentra su interfaz de usuario basada en texto (TUI), que ofrece una forma más estructurada e interactiva de ejecutar playbooks en comparación con la salida de texto plano del comando ansible-playbook.[10] Esta interfaz permite a los usuarios ejecutar playbooks, visualizar los resultados de las tareas en tiempo real, e incluso depurar la ejecución paso a paso, inspeccionando variables y hechos recopilados de los hosts gestionados. La integración con los entornos de ejecución de automatización es otra característica clave, ya que asegura un entorno consistente y portable para la ejecución de los playbooks.[10]

Los casos de uso específicos de ansible-navigator son diversos. Se considera particularmente útil para desarrolladores que crean y prueban contenido de Ansible, ya que facilita la validación y la obtención de retroalimentación inmediata sobre su código. Para los operadores de TI, ansible-navigator ofrece capacidades de

depuración interactivas que pueden simplificar la identificación y resolución de problemas durante la ejecución de los playbooks. Además, es una herramienta valiosa para aquellos que trabajan con entornos de ejecución de Ansible, ya que proporciona una interfaz unificada para gestionar y ejecutar la automatización dentro de estos entornos consistentes. En general, ansible-navigator está diseñado para aquellos usuarios que prefieren una experiencia de línea de comandos más interactiva y estructurada en comparación con la salida tradicional de ansible-playbook.

Comparativa Detallada: Ansible Playbook vs. Ansible Navigator

Para comprender mejor las diferencias entre ansible-playbook y ansible-navigator, es útil realizar una comparación detallada de sus funcionalidades, casos de uso y las ventajas y desventajas de cada uno.

Característica	**Ansible Playbook**	**Ansible Navigator**
Interfaz	Línea de comandos (CLI)	Interfaz de usuario basada en texto (TUI)
Uso Principal	Ejecución automatizada de playbooks	Desarrollo, prueba y depuración interactiva de playbooks
Método de Ejecución	Comando directo (ansible-playbook)	Comando interactivo (ansible-navigator run)
Interactividad	Limitada a opciones de línea de comandos	Alta interactividad durante la ejecución y depuración
Depuración	Principalmente a través de salida verbose y logs	Depuración paso a paso, inspección de variables y hechos
Formato de Salida	Texto plano, configurable con opciones	Estructurado y visualmente organizado en la TUI
Gestión de Entornos	Requiere configuración explícita del entorno	Integración con entornos de ejecución de Ansible
Público Objetivo	Operadores de TI, sistemas de CI/CD	Desarrolladores de Ansible, operadores para depuración

ansible-playbook se recomienda principalmente para la ejecución automatizada y no interactiva de playbooks, especialmente en flujos de trabajo de integración y entrega continua (CI/CD) o en tareas programadas donde no se requiere intervención manual. Es la herramienta estándar y más ampliamente utilizada para ejecutar la automatización de Ansible. Su naturaleza ligera y su amplia adopción la convierten en una opción robusta para la implementación de automatización a gran escala.

Por otro lado, ansible-navigator resulta más apropiado para las fases de desarrollo, prueba y depuración de playbooks. Su interfaz interactiva facilita la exploración del estado de los sistemas durante la ejecución, la inspección de variables y hechos, y la identificación de problemas de configuración o lógica en los playbooks. También es especialmente útil cuando se trabaja con entornos de ejecución de Ansible, ya que proporciona una manera consistente de ejecutar la automatización dentro de estos contenedores preconfigurados.

Entre las ventajas de ansible-playbook se encuentran su amplia adopción y documentación, su estatus como la forma estándar de ejecutar la automatización de Ansible y su relativa ligereza. Sin embargo, puede resultar menos interactivo para la depuración y su salida, especialmente para playbooks complejos, puede ser verbosa y difícil de analizar. Requiere también un buen conocimiento de las opciones de la línea de comandos para controlar su comportamiento.

ansible-navigator, por su parte, ofrece una interfaz TUI más intuitiva para la ejecución y depuración, lo que puede facilitar el trabajo con playbooks complejos. Su integración con los entornos de ejecución de Ansible asegura una mayor consistencia en la ejecución. No obstante, al ser una herramienta relativamente más nueva, puede tener una curva de aprendizaje ligeramente más pronunciada para aquellos que ya están muy familiarizados con ansible-playbook, y añade una capa adicional de abstracción sobre la ejecución directa de playbooks.

Instalación de Ansible Playbook

Antes de poder ejecutar playbooks, es necesario instalar Ansible en el nodo de control. Existen requisitos previos que deben cumplirse, siendo el principal la presencia de Python y el gestor de paquetes pip en el sistema.[5]

Los procedimientos de instalación varían dependiendo del sistema operativo. En sistemas basados en Linux, como Debian y Ubuntu, la instalación se suele realizar utilizando el gestor de paquetes apt. Los comandos típicos son:

Bash

```
sudo apt update
sudo apt install ansible
```

En distribuciones Red Hat, CentOS o Fedora, se utiliza yum o dnf:

Bash

```
sudo yum install ansible
```

o

Bash

```
sudo dnf install ansible
```

Para la instalación en Windows, el método oficialmente soportado es a través del Windows Subsystem for Linux (WSL).[5] Dentro de un entorno WSL, se pueden seguir los procedimientos de instalación para la distribución Linux elegida. También es posible instalar Ansible directamente en Windows utilizando un entorno Python, aunque esta opción podría tener algunas limitaciones en comparación con la instalación en Linux.

En macOS, Ansible puede instalarse utilizando pip. Se recomienda utilizar un entorno virtual de Python para evitar conflictos con los paquetes del sistema:

Bash

```
python3 -m venv ansible_venv
source ansible_venv/bin/activate
pip install ansible
```

Instalación de Ansible Navigator

Al igual que Ansible Core, la instalación de ansible-navigator requiere Python y pip en el nodo de control. Es posible que existan dependencias adicionales según las funcionalidades específicas que se deseen utilizar, especialmente en relación con los entornos de ejecución.

El procedimiento de instalación estándar para ansible-navigator es a través de pip. Se aconseja instalarlo dentro de un entorno virtual de Python para mantener un entorno de trabajo limpio y aislado:

Bash

```
python3 -m venv ansible_navigator_venv
source ansible_navigator_venv/bin/activate
pip install ansible-navigator
```

Si bien el comando de instalación es generalmente el mismo en diferentes sistemas operativos, es importante consultar la documentación oficial de ansible-navigator para identificar posibles consideraciones específicas o dependencias adicionales requeridas en ciertos sistemas operativos para habilitar todas sus funcionalidades o para la configuración de entornos de ejecución particulares.

Automatización con Ansible y Podman

Podman es un motor de contenedores sin demonio que permite desarrollar, gestionar y ejecutar contenedores OCI en sistemas Linux.[7] Ansible puede utilizarse para automatizar diversas tareas relacionadas con Podman, como la gestión del ciclo de vida de los contenedores y la configuración dentro de ellos.

Los casos de uso de Ansible para la gestión de contenedores Podman son variados. Ansible proporciona módulos específicos para interactuar con Podman, lo que permite a los usuarios crear, iniciar, detener, reiniciar y eliminar contenedores de forma automatizada. También se pueden gestionar imágenes de contenedores

(descargar, construir, eliminar) y volúmenes de almacenamiento persistente asociados a los contenedores.

Además de la gestión del ciclo de vida, Ansible puede automatizar la configuración de aplicaciones y servicios que se ejecutan dentro de los contenedores Podman. Esto se puede lograr de varias maneras. Una opción es ejecutar playbooks en el host que interactúan con los contenedores en ejecución, por ejemplo, utilizando el módulo podman_exec para ejecutar comandos dentro de un contenedor. Otra estrategia consiste en construir imágenes de contenedor personalizadas que ya incluyan la configuración deseada, utilizando Ansible durante el proceso de construcción de la imagen (aunque esto es más común con herramientas como Ansible Builder).

Procedimientos Prácticos: Ansible y Podman en Acción

A continuación, se presentan algunos ejemplos prácticos de cómo se puede utilizar Ansible para automatizar tareas con Podman.

Ejemplo 1: Crear un contenedor Podman y desplegar una aplicación web (Nginx)

Este playbook utiliza el módulo podman_container para crear e iniciar un contenedor Nginx.

YAML

```yaml
---
- name: Crear y desplegar un contenedor Nginx con Podman
  hosts: localhost
  tasks:
    - name: Asegurar que la imagen de Nginx esté presente
      community.docker.podman_image:
        name: nginx:latest
        pull: true

    - name: Crear y ejecutar el contenedor Nginx
      community.docker.podman_container:
        name: webserver
        image: nginx:latest
        ports:
```

```
      - "80:80"
    state: started
```

Este playbook primero se asegura de que la imagen nginx:latest esté presente en el sistema local. Si no lo está, la descarga del registro de contenedores. Luego, crea y ejecuta un contenedor llamado webserver basado en esta imagen, mapeando el puerto 80 del contenedor al puerto 80 del host.

Ejemplo 2: Configurar un contenedor Podman con Ansible para una base de datos (PostgreSQL)

Este ejemplo muestra cómo iniciar un contenedor PostgreSQL y configurar una variable de entorno para la contraseña inicial.

YAML

```
---
- name: Crear y configurar un contenedor PostgreSQL con Podman
  hosts: localhost
  tasks:
    - name: Asegurar que la imagen de PostgreSQL esté presente
      community.docker.podman_image:
        name: postgres:13
        pull: true

    - name: Crear y ejecutar el contenedor PostgreSQL
      community.docker.podman_container:
        name: database
        image: postgres:13
        env:
          POSTGRES_PASSWORD: "your_secret_password"
        ports:
          - "5432:5432"
        state: started
```

Este playbook se asegura de tener la imagen postgres:13 y luego crea un contenedor llamado database con una contraseña inicial configurada a través de la variable de

entorno POSTGRES_PASSWORD. El puerto 5432 del contenedor se mapea al puerto 5432 del host.

Ejemplo 3: Gestionar el ciclo de vida de múltiples contenedores Podman con un playbook (aplicación web con base de datos)

Este playbook demuestra cómo gestionar múltiples contenedores interconectados.

YAML

```yaml
---
- name: Gestionar aplicación web y base de datos con Podman
  hosts: localhost
  tasks:
    - name: Asegurar que las imágenes estén presentes
      community.docker.podman_image:
        name: postgres:13
        pull: true
      register: postgres_image

    - name: Asegurar que la imagen de Nginx esté presente
      community.docker.podman_image:
        name: nginx:latest
        pull: true
      register: nginx_image

    - name: Crear y ejecutar la red para los contenedores
      community.docker.podman_network:
        name: my_app_network
        state: present

    - name: Crear y ejecutar el contenedor de base de datos
      community.docker.podman_container:
        name: db
        image: postgres:13
        env:
          POSTGRES_PASSWORD: "db_password"
```

```
      networks:
        - name: my_app_network
      ports:
        - "5432:5432"
      state: started

  - name: Crear y ejecutar el contenedor de la aplicación web
    community.docker.podman_container:
      name: web
      image: nginx:latest
      networks:
        - name: my_app_network
      ports:
        - "80:80"
      links:
        - "db:database"
      state: started

  - name: Detener todos los contenedores de la aplicación
    community.docker.podman_container:
      name: "{{ item }}"
      state: stopped
    loop:
      - db
      - web

  - name: Iniciar todos los contenedores de la aplicación
    community.docker.podman_container:
      name: "{{ item }}"
      state: started
    loop:
      - db
      - web

  - name: Eliminar todos los contenedores de la aplicación
    community.docker.podman_container:
      name: "{{ item }}"
      state: absent
    loop:
```

```
      - db
      - web

  - name: Eliminar la red de la aplicación
    community.docker.podman_network:
      name: my_app_network
      state: absent
```

Este playbook gestiona una aplicación web simple con un contenedor de base de datos. Crea una red, inicia los contenedores de la base de datos y la aplicación web, y también incluye tareas para detener, iniciar y eliminar los contenedores, así como para eliminar la red.

Conclusión

En resumen, Ansible se presenta como una herramienta de automatización de TI potente y versátil, capaz de simplificar una amplia gama de tareas, desde la configuración de sistemas hasta el despliegue de aplicaciones y la orquestación de flujos de trabajo complejos. Si bien tanto ansible-playbook como ansible-navigator son herramientas para interactuar con Ansible, difieren en su interfaz y casos de uso principales. ansible-playbook es el motor principal para la ejecución automatizada de playbooks, ideal para entornos de producción y flujos de trabajo de CI/CD. Por otro lado, ansible-navigator ofrece una interfaz más interactiva y estructurada, lo que lo convierte en una herramienta valiosa para el desarrollo, la prueba y la depuración de playbooks, especialmente cuando se utilizan entornos de ejecución de Ansible.

La capacidad de Ansible para automatizar tareas dentro de entornos de contenedores gestionados por Podman amplía aún más su utilidad. A través de módulos específicos, Ansible puede gestionar el ciclo de vida de los contenedores, configurar aplicaciones dentro de ellos y orquestar implementaciones complejas que involucran múltiples contenedores. Los ejemplos prácticos presentados ilustran cómo Ansible puede simplificar la creación, configuración y gestión de contenedores Podman para diversas aplicaciones y servicios.

Para aquellos que deseen profundizar en el aprendizaje de Ansible, se recomienda consultar la documentación oficial del proyecto Ansible, los recursos proporcionados por Red Hat Ansible Automation Platform, los foros de la comunidad y Ansible Galaxy, donde se pueden encontrar una gran cantidad de roles y colecciones predefinidas para diversas tareas de automatización.

Obras citadas

1. www.redhat.com, fecha de acceso: abril 1, 2025, https://www.redhat.com/en/ansible-collaborative/how-ansible-works#:~:text=Ansible%20is%20an%20open%20source,%2C%20system%20updates%2C%20and%20more.
2. How Ansible Works - Red Hat, fecha de acceso: abril 1, 2025, https://www.redhat.com/en/ansible-collaborative/how-ansible-works
3. Ansible Tutorial for Beginners: Ultimate Playbook & Examples - Spacelift, fecha de acceso: abril 1, 2025, https://spacelift.io/blog/ansible-tutorial
4. What is Ansible? - Medium, fecha de acceso: abril 1, 2025, https://medium.com/@techlatest.net/what-is-ansible-4b0c5afadc7d
5. Ansible (software) - Wikipedia, fecha de acceso: abril 1, 2025, https://en.wikipedia.org/wiki/Ansible_(software)
6. Day 4: Understanding Ansible Playbooks — The Basics | by Vinoth Subbiah | Medium, fecha de acceso: abril 1, 2025, https://medium.com/@vinoji2005/day-4-understanding-ansible-playbooks-the-basics-0414176c72e4
7. Basic Concepts — Ansible Community Documentation, fecha de acceso: abril 1, 2025, https://docs.ansible.com/ansible/latest/network/getting_started/basic_concepts.html
8. Ansible Collaborative - Red Hat, fecha de acceso: abril 1, 2025, https://www.redhat.com/en/ansible-collaborative
9. Ansible architecture — Ansible Community Documentation, fecha de acceso: abril 1, 2025, https://docs.ansible.com/ansible/latest/dev_guide/overview_architecture.html
10. Ansible Automation Platform | Features and Benefits - Red Hat, fecha de acceso: abril 1, 2025, https://www.redhat.com/en/technologies/management/ansible/features
11. Red Hat Ansible Automation Platform - Cisco Partner Solutions & Services, fecha de acceso: abril 1, 2025, https://developer.cisco.com/ecosystem/cpp/solutions/161969/
12. Red Hat Ansible Automation Platform, fecha de acceso: abril 1, 2025, https://www.redhat.com/en/technologies/management/ansible
13. Ansible Playbooks: Complete Guide with Examples - Spacelift, fecha de acceso: abril 1, 2025, https://spacelift.io/blog/ansible-playbooks
14. What is an Ansible Playbook? - Red Hat, fecha de acceso: abril 1, 2025, https://www.redhat.com/en/topics/automation/what-is-an-ansible-playbook
15. What is an Ansible Playbook and How to Write One on Your Own?, fecha de acceso: abril 1, 2025, https://www.simplilearn.com/what-is-ansible-playbook-article
16. Intro to Playbooks — Ansible Documentation - Read the Docs, fecha de acceso: abril 1, 2025, https://cn-ansibledoc.readthedocs.io/zh_CN/latest/user_guide/playbooks_intro.ht

ml
17. Ansible playbooks — Ansible Community Documentation, fecha de acceso: abril 1, 2025, https://docs.ansible.com/ansible/latest/playbook_guide/playbooks_intro.html
18. 4 | From Beginner to Pro: Ansible Playbooks Explained - YouTube, fecha de acceso: abril 1, 2025, https://www.youtube.com/watch?v=7aRrHyQVD8o
19. ansible/ansible-examples: A few starter examples of ansible playbooks, to show features and how they work together. See http://galaxy.ansible.com for example roles from the Ansible community for deploying many popular applications. - GitHub, fecha de acceso: abril 1, 2025, https://github.com/ansible/ansible-examples