

WHITE PAPER

# Los beneficios de un orquestador sin agentes

Pablo Chalimoniuk  
Cloud Architect at Nubiral

  
**nubiral**  
Integrating the Future



**Ansible es un software open source que automatiza el aprovisionamiento de software, el despliegue de aplicaciones y la gestión de configuraciones.**

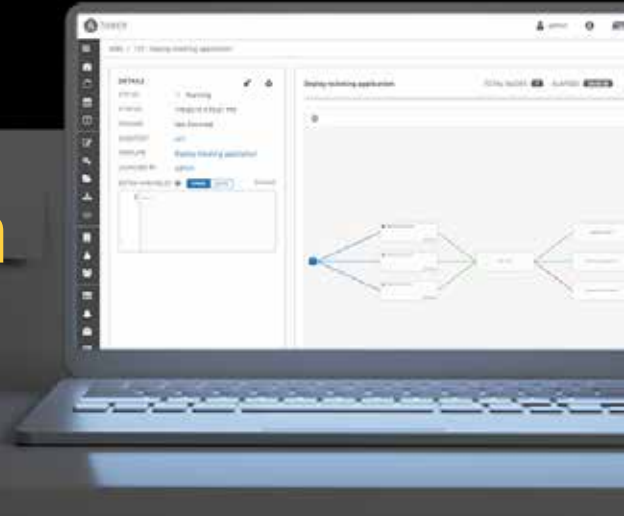
Ansible posibilita que los sysadmins/devops puedan gestionar sus servidores, tanto físicos como virtuales, automatizando tareas de administración mediante el uso de recetas (playbooks) definiendo un estándar dentro de la compañía. Estas recetas contienen todas las tareas que son ejecutadas en los equipos administrados, desde la instalación de paquetes como así también la configuración de las aplicaciones de forma sencilla y paralela, cuando se trata de propagar despliegues en múltiples equipos.

Una de las grandes ventajas de Ansible es que no tiene demandas de instalación de agentes en los equipos administrados. El único requisito para poder utilizar ansible en los servidores remotos es que cuenten con alguna versión de Python instalada, ya que su lenguaje de scripting es YAML (YAML Ain't Markup Language).

Ansible fue creado por Michael DeHaan en 2012, cuya empresa, Ansible Inc, fue vendida a RedHat en 2015. Actualmente recibe contribuciones desde el punto de vista del desarrollo desde más de 4.000 usuarios (y creciendo) alrededor del mundo.



# El núcleo de la cuestión



El método que utiliza **Ansible** para conectarse con los equipos remotos es el protocolo de conexión que ya está configurado (o debería) en todos los equipos. Por ej en los equipos Unix este protocolo es el SSH, y en los servidores Windows se logra mediante Windows Remote Management a través de scripting de Powershell.

Una vez establecida la comunicación con los equipos administrados, pequeños módulos de ansible se transfieren por el mecanismo de transporte establecido, luego son ejecutados remotamente, y finalmente son removidos del destino, todo esto en una única acción en forma de bloque.

Como resultado de esta interacción, estos pequeños módulos retornan una salida en formato JSON (JavaScript Object Notation) que son procesados por el controlador Ansible (el equipo host) y mostrados como salida regular en la consola.

Algo interesante a destacar es que un tráfico ligero de redes tales como estos módulos de

**Ansible** pueden desencadenar una gran cantidad de actividad remota. Estos módulos son gestores de recursos idempotentes, esto significa que el módulo es capaz de determinar qué acciones tomar en determinados escenarios, como por ej, antes de instalar un paquete del sistema operativo puede verificar que ese mismo paquete ya esté instalado y cuál es su versión actual, reduciendo de esta manera la cantidad de ejecuciones innecesarias.

**SIN DEMANDAS DE  
INSTALACIÓN DE AGENTES EN  
LOS EQUIPOS**

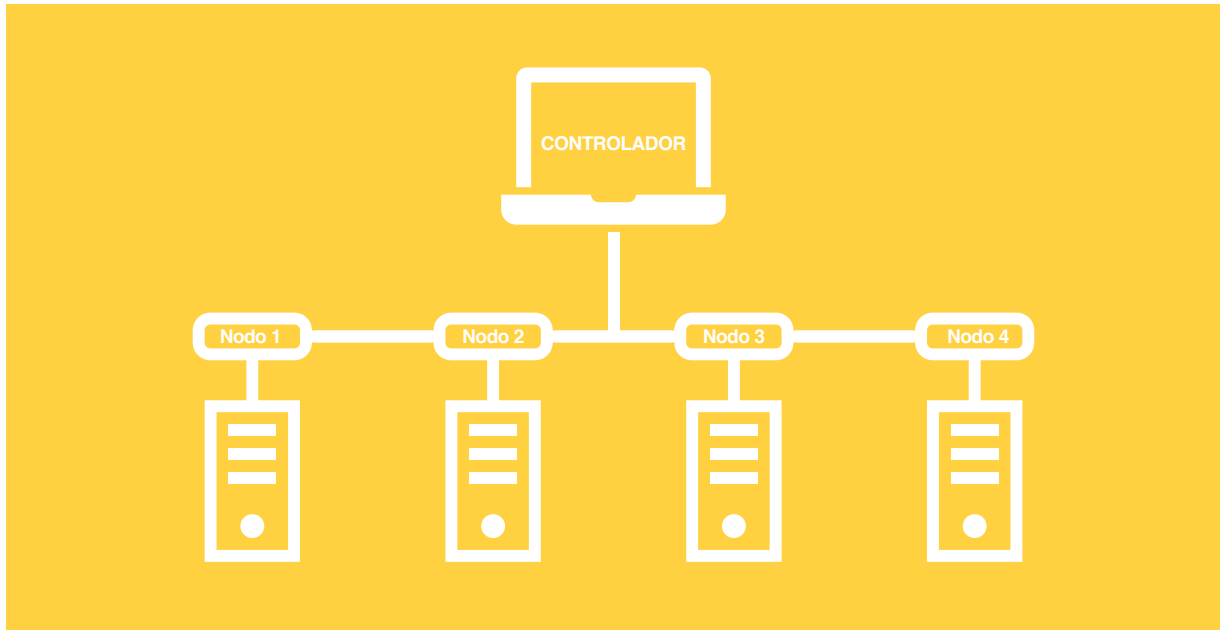


## Arquitectura

En Ansible existen 2 tipos de servidores:

**Controlador:** También llamado host, es el equipo que opera de orquestador.

**Nodo:** Es el nodo remoto que es operado por el controlador.



## Seguridad de red

Los únicos servicios que son necesarios en ejecución en los equipos remotos para posibilitar la interconexión de Ansible son SSH (Unix) y WinRM (Windows). Consecuentemente, Ansible tiene un muy bajo nivel de ataque ya que ambos servicios son periódicamente revisados en niveles críticos a nivel mundial porque determinan en gran parte la seguridad del sistema.

## Nivel de acceso no-root con sudo

Los playbooks de Ansible pueden crearse para hacer login con un determinado usuario y luego pasar a otro usuario (incluyendo root, con o sin requerimiento de password) utilizando los mecanismos de escalamiento de privilegios, o bien directamente como root (si esto está permitido en la configuración de SSH).

Aún utilizando sudo, la transferencia de archivos no está limitada a los permisos del usuario ya que Ansible es lo suficientemente inteligente para transferir el contenido vía SFTP y luego mover el archivo con las credenciales necesarias utilizando sudo. Además evita la transferencia innecesaria cuando los archivos origen/destino coinciden en sus checksums.





## El amigo del firewall

Otros orquestadores necesitan tener conexiones abiertas entre el host y los clientes, lo que implica que deban abrirse puertos en los firewalls que los protegen. Ansible con su modelo empujar-a-demanda no tiene este requisito, ya que opera sin agentes sobre SSH/WinRM, donde todas las conexiones se establecen sobre un servicio de acceso remoto que ya está configurado para ese fin.

**SIN INSTALACIÓN NI  
CONFIGURACIÓN DE AGENTES  
EN EL DESTINO.**

## Sin agentes a la vista

Debido a su naturaleza, cuando Ansible no está gestionando nodos, no hay ningún proceso de Ansible corriendo en esos nodos a la espera de recibir tráfico con órdenes para ejecutar. Esto significa que la performance en los nodos administrados no se ve en absoluto afectada en cuanto a CPU o memoria.

## Encendiendo motores

El tiempo que necesita Ansible para poder comenzar a gestionar equipos remotos se reduce a cero ya que no necesita ni instalación ni configuración de agentes en el destino. En entornos inmensamente grandes, con miles de equipos co-existiendo y donde es necesario una simple actualización de paquetes, esta tarea en apariencia tediosa se transforma en algo simple con Ansible.



