

AUTOMATIZACIÓN DE PROCESOS Y APROVISIONAMIENTO DE SISTEMAS EN UNA INTRANET CON ANSIBLE

Yurany Andrea Castro Gómez
IES SERPIS – VALENCIA
Curso 2022/2023

RESUMEN

El objetivo principal del presente trabajo ha sido simular un entorno virtualizado de un entorno productivo en una Intranet, en la cual habrá una gestión centralizada de aprovisionamiento de recursos en los nodos gestionados para hacer de ellos un conjunto de sistemas homogéneos, y realizar una automatización de procesos y tareas administrativas que son efectuadas de manera repetitiva en las Empresas de TI y que pueden ser comúnmente propensas a errores, como lo son por ejemplo las instalaciones y configuraciones de servicios de red, gestionar proyectos con grandes inventarios, administrar conexiones, desplegar una aplicación o hacer configuraciones de hardware y de software entre otras. Con el uso de Ansible automatizamos distintos tipos de configuraciones iniciales y procesos diarios de forma más rápida, sencilla y eficaz de manera remota mediante el uso del protocolo SSH administrando así los nodos secundarios a través de los módulos que se incluyen en esta herramienta y la cual ha podido demostrar su efectividad a partir del desarrollo de 'Playbooks', llamados así a los manuales de jugadas, los cuales se han utilizado para simplificar las distintas tareas del proyecto. Con la Automatización mediante Ansible, se ha podido conseguir una mayor eficiencia al poder eliminar los procesos manuales, ajustar la infraestructura de la empresa con más facilidad, además de obtener resultados en menor tiempo en los despliegues de aplicaciones, y de a futuro adaptar e implementar las innovaciones necesarias más rápidamente como consecuencia de las buenas prácticas DevOps. En definitiva, con la Automatización desde Ansible se ha podido sacar mayor provecho en los resultados de las tareas y procesos rutinarios realizados al minimizar el tiempo y la complejidad en su ejecución y por consiguiente obtener el resultado deseado.

INDICE

1. JUSTIFICACIÓN.....	4
2. GESTION DEL PROYECTO:.....	5
I. Tablero Kanban del proyecto.....	5
II. Repositorio personal en Github.....	6
3. HERRAMIENTAS UTILIZADAS.....	6
4. DESCRIPCIÓN DEL PROYECTO:.....	7
4.1. Análisis.....	7
¿Qué es Ansible?.....	7
¿Cómo funciona?.....	8
Instalación.....	8
Conceptos claves.....	11
Arquitectura.....	12
4.2. Diseño.....	13
Entorno de trabajo.....	13
Esquema de red: Intranet.....	14
Casos de uso.....	15
4.3. Desarrollo.....	15
4.4. Pruebas.....	16
4.5. Documentación.....	16
5. TRABAJOS FUTUROS.....	16
6. CONCLUSIONES.....	16
7. BIBLIOGRAFÍA Y WEBGRAFÍA.....	16
8. ANEXOS:.....	16
8.1. Manuales de Instalación.....	16
Instalación de Ansible — Documentación de Ansible.....	16
8.2. Partes del código y/o ficheros de configuración importantes.....	16
8.3. Descripción de los contenidos incluidos en soporte digital.....	16

INDICE DE IMAGENES

1. JUSTIFICACIÓN

Actualmente la automatización va obteniendo mayor importancia y un papel crucial en las empresas de TI, esto es debido a que la complejidad de tareas que requieren mantenimiento, monitoreo, actualizaciones, escalado, tareas de administración diaria, cada vez aumenta en entornos empresariales.

Para los administradores de sistemas y demás equipos de informáticos, mantener todo actualizado, realizar tareas repetitivas y complicadas de forma manual resulta una tarea bastante laboriosa y propensa a los errores, ya que se podrían omitir pasos durante un proceso, además se ha de añadir el hecho de que se necesitaría contar con personal capacitado para ocuparse de ello.

La automatización ayuda en gran medida a evitar todos los problemas causados por una administración de sistemas manual, a simplificar la ejecución de tareas repetitivas y complejas, y así gestionar los procesos de forma más rápida y eficiente .

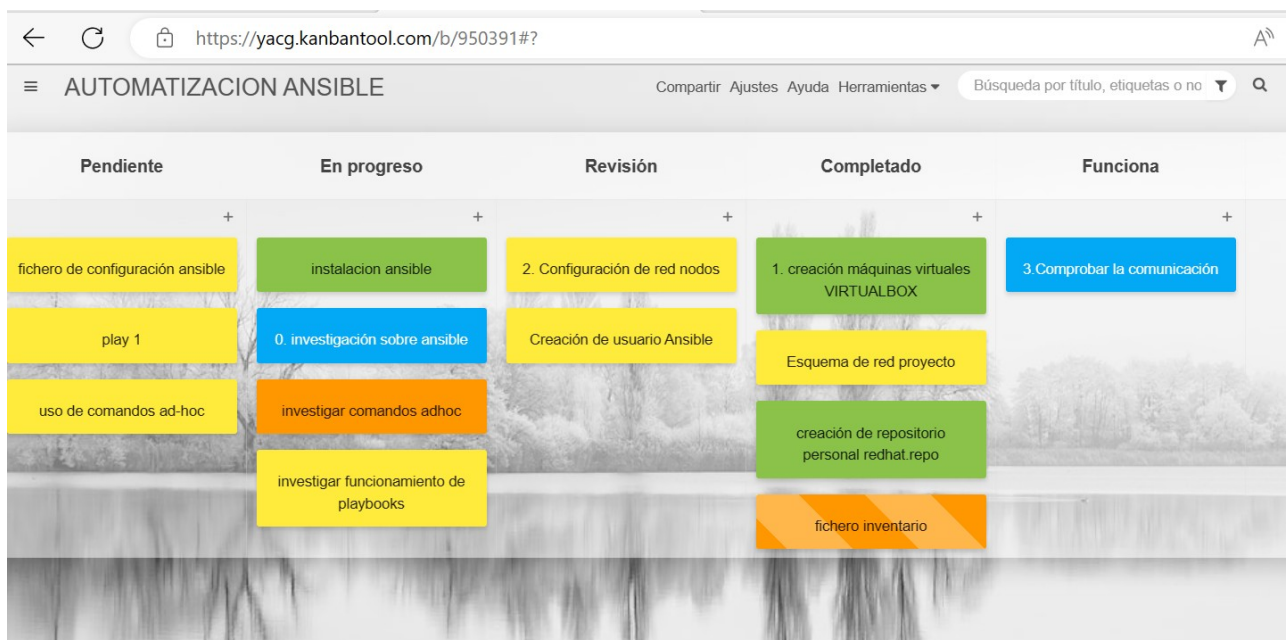
Con el fin de conseguir este objetivo, se ha elegido la herramienta de automatización Ansible, la cual sorprende con grandes utilidades, y permite:

- Implementar rápidamente aplicaciones de varios niveles
- Aprovisionar nuestros servidores/hosts en un entorno productivo virtualizado
- Gestionar configuraciones de forma más sencilla, fiable y segura
- Fomenta el uso de programas opensource y es compatible con distintos sistemas
- El uso de manuales escritos en YAML (playbooks), los cuales pueden ser interpretados por personas que no sean administradores de sistemas o perfiles informáticos.
- Facilita la orquestación, una vez se define la infraestructura mediante playbooks, se puede usar el mismo para implementar la misma orquestación en otros servidores.

2. GESTION DEL PROYECTO:

Para llevar a cabo la elaboración de este proyecto se ha gestionado la planificación de una serie de tareas propuestas mediante el uso de dos herramientas muy útiles en un entorno de trabajo diario, como lo es un tablero Kanban que permite llevar un seguimiento de su estado, se pueden indicar fechas, tiempo dedicado, y registro y descripción de los trabajos realizados o por hacer; y además para tener una organización de los archivos y directorios que se crean en el proceso se ha escogido Github para crear un repositorio donde depositar todos estos archivos y tenerlos accesibles cuando se quieran revisar o añadir más documentación.

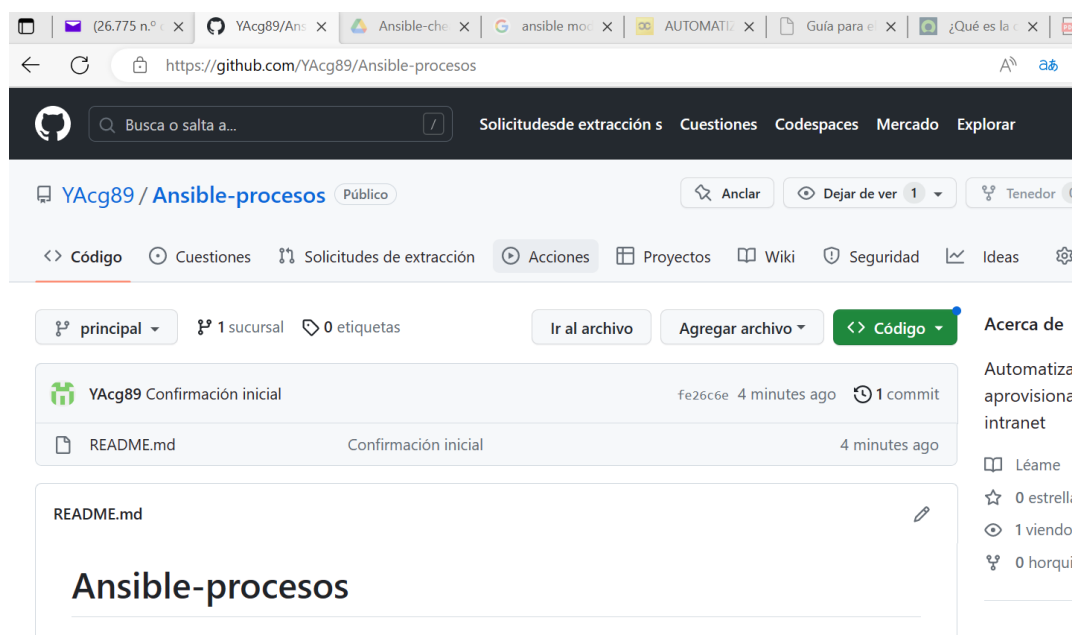
I. Tablero Kanban del proyecto



Nombre del tablero: [AUTOMATIZACION ANSIBLE - Yacg - Kanban Tool](https://yacg.kanbantool.com/b/950391#?)

II. Repositorio personal en Github

Se crea repositorio personal en Github para guardar todas las configuraciones iniciales de Ansible y sus nodos, como también el desarrollo de todos los playbooks elaborados. Este repositorio está disponible para el público en general y además para quien desee a futuro añadir más contenido.



3. HERRAMIENTAS UTILIZADAS

Para la puesta en marcha y finalización del proyecto se han propuesto y definido las siguientes herramientas y tecnologías:

- **Virtualbox:** Software para la creación de máquinas virtuales y simulación del entorno productivo
- **OS RedHat Enterprise Linux:** RHEL release 9.2 (Plow) y RHEL release 9.1 (Plow)
- **Ansible:** para automatizar tareas y gestionar el aprovisionamiento de servidores
- **Github:** Repositorio personal para guardar código
- **YAML:** formato que tienen los playbooks
- **Draw.io:** herramienta para diseñar el esquema de red
- **Phyton :** lenguaje de scripts que usa Ansible en sus módulos

4. DESCRIPCIÓN DEL PROYECTO:

4.1. Análisis

Se ha planteado para este proyecto realizar una simulación en virtualbox de un entorno productivo dentro de una intranet en la que se realizará una serie de tareas y procesos (repetitivos para administradores de sistemas y técnicos devops) que han de ser automatizados con la herramienta de automatización Ansible, la cual se ha elegido para comprobar que en la actualidad cada vez más las empresas optan por emplear este tipo de tecnologías con el objetivo de no solo ahorrar tiempo y agilizar las tareas, sino también a reducir los errores humanos, además de aumentar la productividad y la disponibilidad para la desarrollo de otras tareas más complejas.

Por lo tanto para tener una visión general de como se trabaja con esta tecnología es necesario saber que Ansible solo necesita dos cosas que son:

- Conexión por SSH a los equipos/servidores donde se vayan a ejecutar playbooks o roles, e
- Instalar Python en los equipos/servidores a los que se va a conectar el equipo que contenga Ansible.

Para conocer toda la utilidad que posee Ansible, se han analizado aspectos que son importantes para entender como funciona.

¿Qué es Ansible?

Según la página oficial de redhat, Ansible es un motor open source que automatiza los procesos para preparar las infraestructuras, implementar las aplicaciones, gestionar configuraciones de servidores, de dispositivos de red y/o servicios en la nube a la vez, de una forma más rápida y sencilla.

- ✓ Está escrito en Python
- ✓ Es un software libre
- ✓ Está provisto de una amplia colección de módulos
- ✓ Utiliza playbooks y comandos ad-hoc para la ejecución de tareas o conjunto de tareas
- ✓ Utiliza el formato YAML en el diseño de playbooks, de fácil lectura y declarativo

- ✓ No necesita agentes en los nodos gestionados, por lo tanto no se necesita instalar ningún software solo Python

Está disponible en Fedora, RedHat, Centos, pero además es compatible con diversas distribuciones Linux, MacOS, y con Windows solo para versiones que puedan gestionar subsistemas linux.

¿Cómo funciona?

Ansible funciona realizando conexiones a los nodos/servidores administrados mediante **SSH** enviando así los módulos que lo integran, estos módulos son definidos como pequeños programas que hacen una tarea específica dentro de un playbook.

Esto se traduce a ejecutar comandos específicos o a la creación de scripts para ejecutar tareas, por eso Ansible utiliza “**playbooks**” o también llamados manual de jugadas, en los cuales se define una tarea específica que se ejecuta una sola vez (play) o un conjunto de tareas a realizar por los nodos seleccionados.

Instalación

En este apartado se describe como es la instalación de Ansible en sistemas RHEL 9. Como se ha dicho anteriormente, Ansible no necesita agentes (**agentless**) para administrar máquinas ya que lo hace a través del protocolo SSH, y por eso no existen demonios para ejecutar.

Sólo es necesario instalar Ansible en el nodo principal, que en nuestro caso es la “WORKSTATION”, y a partir de aquí se puede administrar tantas máquinas remotas como se desee. Además hay que añadir que cuando Ansible ejecuta tareas de automatización no deja en estas software instalado o ejecutándose en segundo plano.

Requisitos para el uso de Ansible:

- En el nodo principal: Tener instalado Python2 (versión 2.7) o Python3 (versiones 3.5 o superiores). *A excepción de sistemas windows, que no puede instalarlo en nodo principal; y conexión SSH.
- En los nodos gestionados: Tener instalado Python2 (versión 2.7) o Python3 (versiones 3.5 o superiores).

Ahora, lo primero que se debe tener en cuenta en este momento es conocer la fuente de donde se obtendrá la herramienta de Automatización Ansible, y esa fuente es un repositorio EPEL que se puede encontrar en comunidades de software libre o en las páginas de distribuciones Linux.

También se puede crear un repositorio y añadirlo al **fichero /etc/yum.repos.d/** que es donde se quedan almacenados todos los **archivos .rpms**. Para el proyecto se ha creado un nodo "WORKSTATION", y se ha hecho de la siguiente forma:

1) Creación del repositorio:

- Registro de la cuenta de RedHat para poder disponer de los repositorios para RHEL 9

```
[root@workstation ~]# subscription-manager register
Registrándose a: subscription.rhsm.redhat.com:443/subscription
Nombre de usuario: yurany.castro89
Contraseña:
The system has been registered with ID: e8dc79dd-6155-4f39-8a5b-95e227558cba
The registered system name is: workstation
```

- Habilitar la suscripción de RedHat para disponer de los repositorios autorizados:
[Cómo habilitar la suscripción RHEL en RHEL 8 - LinuxParty](#)

- Lista de repositorios una vez hecha la suscripción

```
[root@workstation ~]# yum repolist
Error al cargar el complemento "config_manager": '*prog'
Actualización de repositorios de Subscription Management.
id del repositorio                nombre del repositorio
EPEL                             Repo redhat EPEL
rhel-9-for-x86_64-appstream-rpms  Red Hat Enterprise Linux 9 for x86_64 - AppStream (RPMs)
rhel-9-for-x86_64-baseos-rpms    Red Hat Enterprise Linux 9 for x86_64 - BaseOS (RPMs)
[root@workstation ~]# nano /etc/yum.repos.d/redhat.repo
[root@workstation ~]# subscription-manager repos --list
+-----+
Available Repositories in /etc/yum.repos.d/redhat.repo
```

- Fichero donde se guardan todos los paquetes de programas y que hay que habilitar

```
root@workstation:~
GNU nano 5.6.1 /etc/yum.repos.d/redhat.repo Modificado
baseurl = https://cdn.redhat.com/content/dist/rhel9/$releasever/x86_64/supplementary/os
enabled = 0
gpgcheck = 1
gpgkey = file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
sslverify = 1
sslcacert = /etc/rhsm/ca/redhat-uep.pem
sslclientkey = /etc/pki/entitlement/7982300608717604386-key.pem
sslclientcert = /etc/pki/entitlement/7982300608717604386.pem
sslverifystatus = 1
metadata_expire = 86400
enabled_metadata = 0

[ansible-automation-platform-2.3-for-rhel-9-x86_64-rpms]
name = Red Hat Ansible Automation Platform 2.3 for RHEL 9 x86_64 (RPMs)
baseurl = https://cdn.redhat.com/content/dist/layered/rhel9/x86_64/ansible-automation-platform/2.3-for-rhel-9-x86_64-rpms
enabled = 1
gpgcheck = 0
gpgkey = file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
sslverify = 1
sslcacert = /etc/rhsm/ca/redhat-uep.pem
```

2) Localización de Ansible

```
[root@workstation ~]# yum search ansible-core
Error al cargar el complemento "config_manager": '*prog'
Actualización de repositorios de Subscription Management.
Última comprobación de caducidad de metadatos hecha hace 0:08:11, el mar 16 may 2023 23:37:46.
===== Coincidencia exacta en Nombre: ansible-core =====
ansible-core.x86_64 : SSH-based configuration management, deployment, and task execution system
===== Coincidencia en Resumen: ansible-core =====
ansible.noarch : Curated set of Ansible collections included in addition to ansible-core
[root@workstation ~]#
```

3) Instalación

```

ansible-collection-redhat-rhel_mgmt.noarch
ansible-core.x86_64
[root@workstation ~]# yum install ansible-core.x86_64
Error al cargar el complemento "config_manager": '*prog'
Actualización de repositorios de Subscription Management.
Última comprobación de caducidad de metadatos hecha hace 0:10:30, el mar 16 may 2023 23:37:46.
Dependencias resueltas.
=====
Paquete      Arq.  Versión      Repositorio      Tam.
=====
Instalando:
  ansible-core x86_64 2.14.5-1.el9ap ansible-automation-platform-2.3-for-rhel-9-x86_64-rpms 2.0 M
Instalando dependencias:
  git-core x86_64 2.39.1-1.el9 rhel-9-for-x86_64-appstream-rpms 4.2 M
  python3-cffi x86_64 1.15.0-2.el9pc ansible-automation-platform-2.3-for-rhel-9-x86_64-rpms 241 k
  python3-cryptography x86_64 36.0.1-2.el9 rhel-9-for-x86_64-baseos-rpms 1.2 M
  python3-jinja2 noarch 3.1.2-1.el9ap ansible-automation-platform-2.3-for-rhel-9-x86_64-rpms 243 k
  python3-markupsafe x86_64 2.1.0-2.el9ap ansible-automation-platform-2.3-for-rhel-9-x86_64-rpms 28 k

```

- Paquetes instalados.

```

Descargando paquetes:
(1/11): python3-markupsafe-2.1.0-2.el9ap.x86_64.rpm 92 kB/s | 28 kB 00:00
(2/11): python3-jinja2-3.1.2-1.el9ap.noarch.rpm 759 kB/s | 243 kB 00:00
(3/11): sshpass-1.09-4.el9ap.x86_64.rpm 166 kB/s | 28 kB 00:00
(4/11): python3-cffi-1.15.0-2.el9pc.x86_64.rpm 1.1 MB/s | 241 kB 00:00
(5/11): ansible-core-2.14.5-1.el9ap.x86_64.rpm 7.5 MB/s | 2.0 MB 00:00
(6/11): python3-pyparsing-2.4.7-9.el9.noarch.rpm 878 kB/s | 154 kB 00:00
(7/11): python3-cryptography-36.0.1-2.el9.x86_64.rpm 4.8 MB/s | 1.2 MB 00:00
(8/11): python3-resolvelib-0.5.4-5.el9.noarch.rpm 17 kB/s | 38 kB 00:02
(9/11): git-core-2.39.1-1.el9.x86_64.rpm 16 MB/s | 4.2 MB 00:00
(10/11): python3-packaging-21.3-1.el9pc.noarch.rpm 17 kB/s | 71 kB 00:04
(11/11): python3-pycparser-2.21-2.el9pc.noarch.rpm 42 kB/s | 195 kB 00:04
-----
Total 1.6 MB/s | 8.4 MB 00:05
Red Hat Enterprise Linux 9 for x86_64 - BaseOS (RPMs) 3.5 MB/s | 3.6 kB 00:00
Importando llave GPG 0xFD431D51:
ID usuario: "Red Hat, Inc. (release key 2) <security@redhat.com>"
Huella : 567E 347A D004 4ADE 55BA 8A5F 199E 2F91 FD43 1D51
Desde : /etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
¿Está de acuerdo [s/N]?: s
La llave ha sido importada exitosamente
Importando llave GPG 0x5A6340B3:
ID usuario: "Red Hat, Inc. (auxiliary key 3) <security@redhat.com>"
Huella : 7E46 2425 8C40 6535 D56D 6F13 5054 E4A4 5A63 40B3
Desde : /etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
¿Está de acuerdo [s/N]?: 

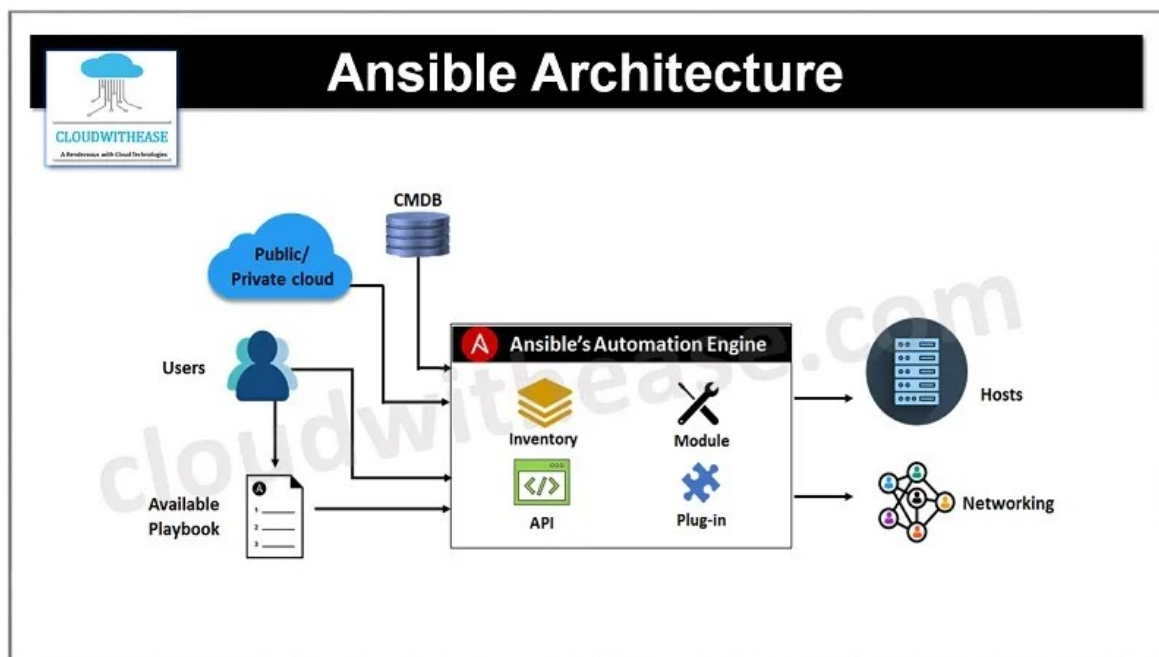
```

Aunque ya se dispone de un usuario root, para mayor seguridad se ha creado otro usuario en el servidor WORKSTATION para que ejecute las automatizaciones:

Conceptos claves

Arquitectura

En esta imagen se puede apreciar los elementos que compone la arquitectura de ansible desde el archivo de inventarios, sus módulos, etc, hasta los hosts, redes, usuarios que se necesite administrar de manera remota.



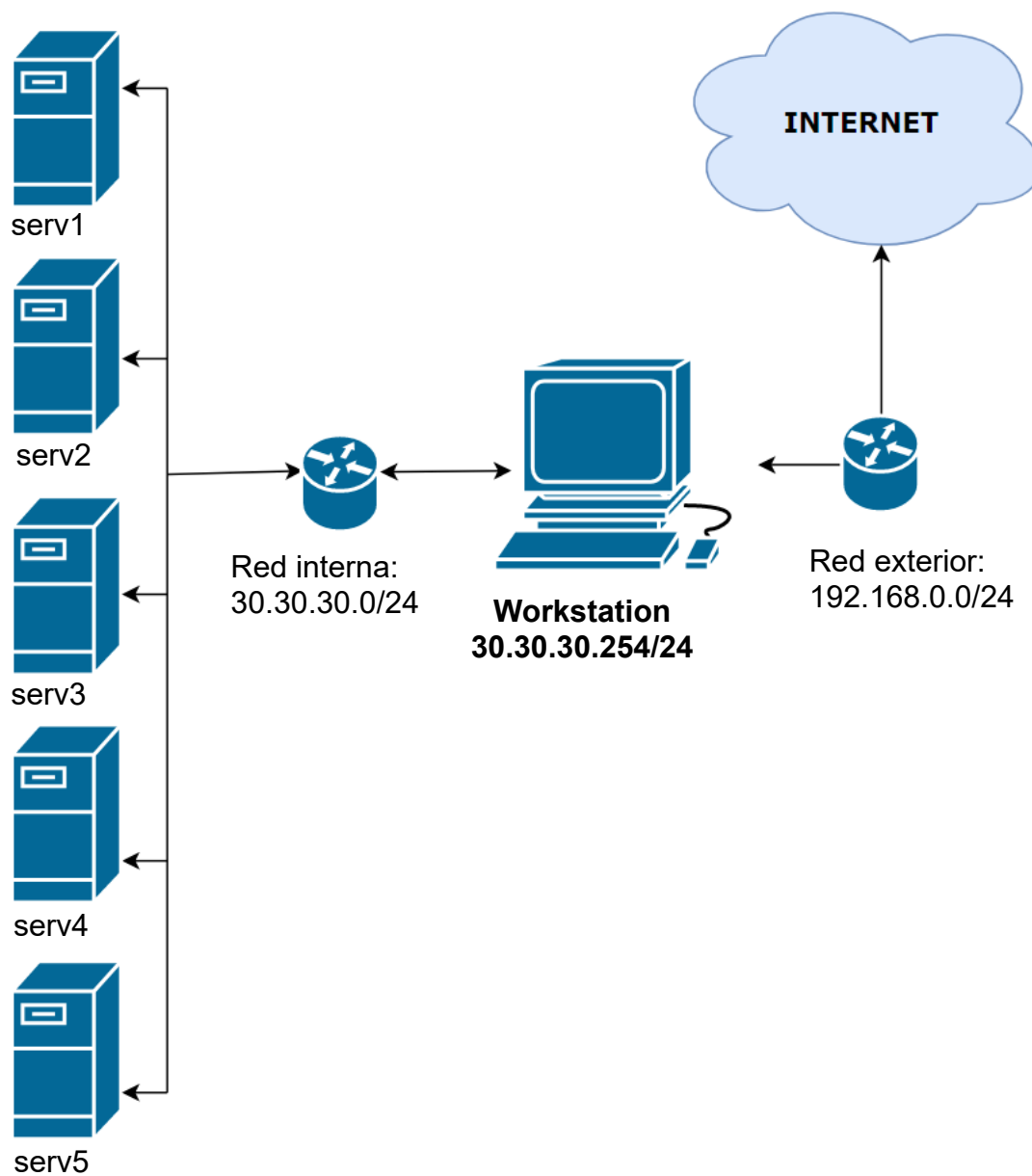
4.2. Diseño

Entorno de trabajo

A continuación se indica todo lo necesario para crear el entorno productivo dentro de una Intranet para el proyecto.

Nombre del servidor	Direcciones IP	Representación
Workstation: Adaptadores de red: - Adaptador 1: tipo puente - Adaptador 2: red interna intnet	0	<ul style="list-style-type: none"> Nodo administrador: <ul style="list-style-type: none"> - contiene interfaz gráfica de RHEL 9 - contiene repositorio personal que sirve para ser usado por los dispositivos en la intranet
Serv1: - Adaptador: red interna intnet	0	• 0
Serv2: - Adaptador: red interna intnet	0	• 0
Serv3: - Adaptador: red interna intnet	0	• 0
Serv4: - Adaptador: red interna intnet	0	• 0
Serv5: - Adaptador: red interna intnet	0	• 0

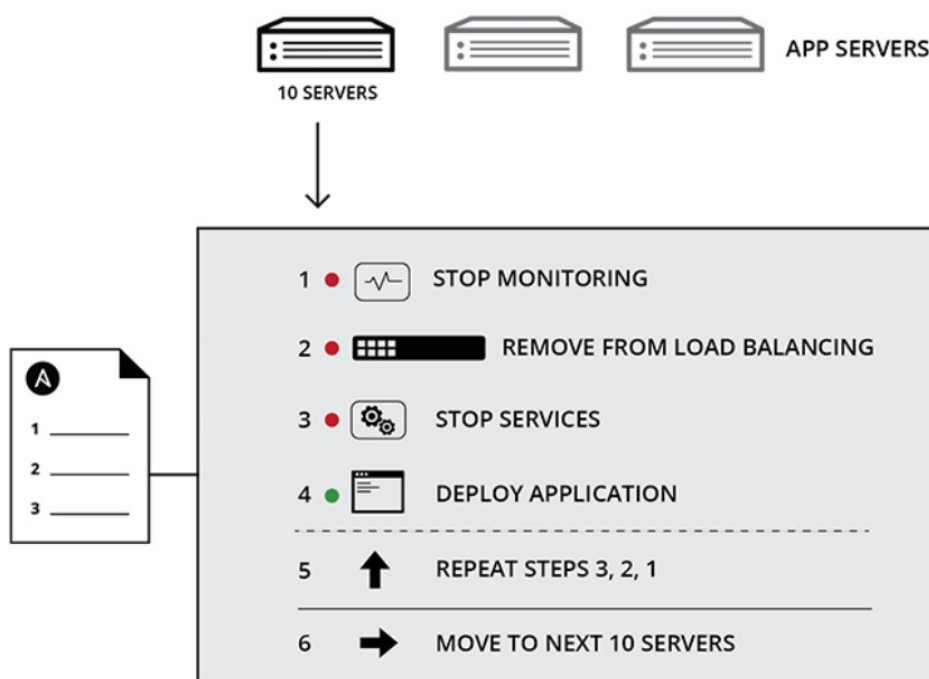
Esquema de red: Intranet



Casos de uso

1. Implementación de aplicaciones:

Configurar servicios claves como algún servicio de red como apache o ftp; servicios del sistema o insertar archivos de aplicaciones, todo esto desde un mismo sistema común en lugar de hacer una configuración personalizada de automatización para un sistema en concreto; de esta manera incluiremos descripciones de todas las tareas que queremos realizar en todos nuestros servidores o grupo de servidores y enviarlo a través de un playbook e implementarlo de manera estándar tantas veces como se quiera.



2. Automatización de redes

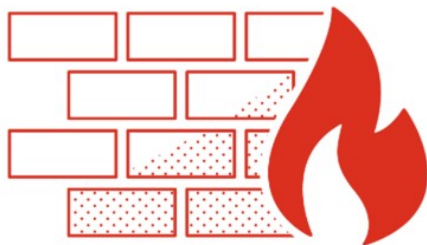
Administrar toda la red y los procesos de TI a través de redes físicas, redes definidas por software y redes basadas en la nube.

Administración de recursos de red



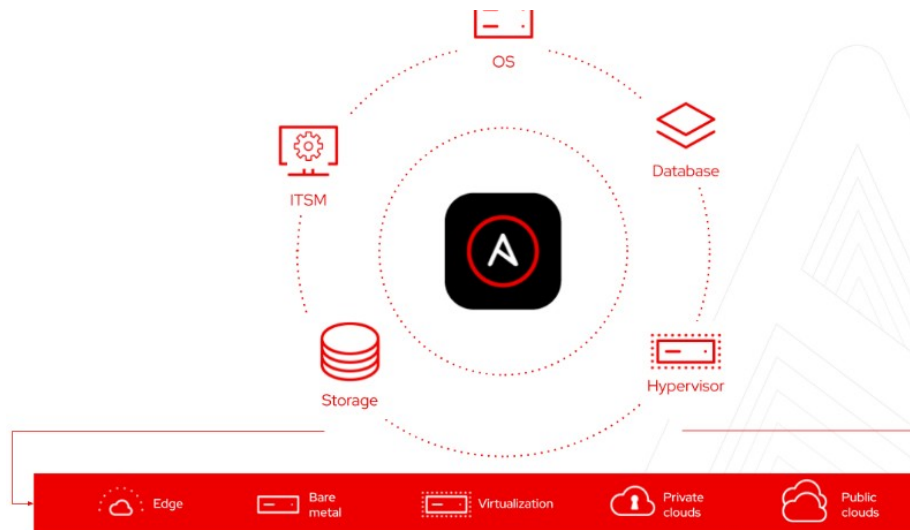
3. Automatización de la seguridad

Organiza los sistemas de seguridad utilizando una colección seleccionada de módulos, roles y manuales para investigar y responder a las amenazas.



4. Infraestructuras

Implementa, administra y escala cargas de trabajo en toda la infraestructura híbrida con una plataforma de automatización unificada.



5. Gestión de la configuración

Centraliza la administración e implementación de archivos de configuración con una curva de aprendizaje baja para administradores, desarrolladores y administradores de TI.

****Fuente:** [Casos de uso de Ansible](#)

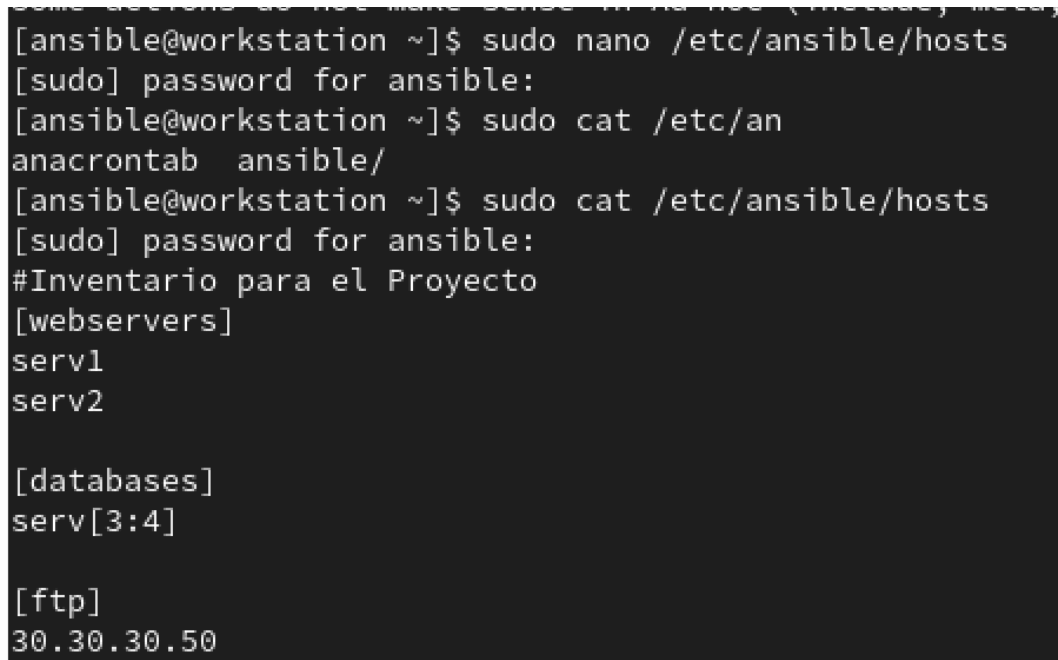
4.3. Desarrollo

El desarrollo de este proyecto pasa por varias etapas para conseguir que funcione como se espera. La primera etapa ha sido la investigación de conceptos claves para entender como se usa esta herramienta para automatizar procesos y aprovisionar nuestros servidores.

Después creamos la simulación del entorno productivo en virtualbox, en donde se crearon cinco máquinas virtuales con sistemas RHEL 9 que son a los que van dirigido las automatizaciones de configuraciones, aprovisionamiento y procesos rutinarios.

Posteriormente a la instalación de ansible, el siguiente paso fue realizar las configuraciones iniciales del fichero de configuración de Ansible y la creación del archivo Inventario para dejar el entorno de trabajo para crear y probar como funcionan los comandos ad-hoc, plays y playbooks.

Creación del fichero de inventario



```
[ansible@workstation ~]$ sudo nano /etc/ansible/hosts
[sudo] password for ansible:
[ansible@workstation ~]$ sudo cat /etc/ansible/hosts
#Inventario para el Proyecto
[webservers]
serv1
serv2

[databases]
serv[3:4]

[ftp]
30.30.30.50
```

imagen fichero inventario

Fichero de configuración de Ansible: ansible.cfg

- Opciones que se debe tener en cuenta en el fichero de ansible:

Imagen fichero de configuración ansible.cfg

Comandos adhoc

Primer uso: de un comando adhoc para el proyecto

Segundo uso: autenticación basada en claves ejecutada con comandos adhoc

Tercer uso: comprobar conectividad con los servidores remotos usando el modulo ping

Otro interesante

Playbooks en Ansible

Documentación de los modulos

4.4. Pruebas

4.5. Documentación

5. TRABAJOS FUTUROS

6. CONCLUSIONES

7. BIBLIOGRAFÍA Y WEBGRAFÍA

8. ANEXOS:

8.1. Manuales de Instalación

[Instalación de Ansible — Documentación de Ansible](#)

8.2. Partes del código y/o ficheros de configuración importantes

8.3. Descripción de los contenidos incluidos en soporte digital

