

Proyecto Integrado

Admón. de Sistemas Informáticos en Red

NETWORKING AUTOMATIZADO

Solucionando los problemas de la infraestructura clásica

AUTOR:

Javier Sánchez Páez

TUTOR/ES:

Víctor Montero Malagón

Javier Pastor Cascales

IES Zaidín-Vergeles (Granada), curso 2021-2023

# Agradecimientos

A mi familia, porque sin ellos no estaría donde estoy.

A Carlos, por ser mi mentor desde que tengo uso de razón.

A Vicky, por darme la paz y la guerra que me hace falta para seguir.

A los buenos profesores que me han acompañado, por enseñarme y mantenerme motivado.

A mis amigos y amigas, por ser quienes son y por estar siempre ahí.

# Abstract

Al montar una infraestructura de red en un entorno siempre se hace difícil o pesado tener que configurar todos los dispositivos manualmente. Este método implica pérdida de tiempo, tener que aprender múltiples comandos de varios sistemas operativos distintos y no tener ninguna seguridad de si funcionará y si, en caso de que no funcionara, poder volver a una versión estable, entre otros.

El objetivo de este proyecto es demostrar que hoy en día podemos crear un entorno de desarrollo seguro también para nuestra infraestructura, con todos los puntos positivos que esto conlleva (testing, rollbacks en caso de fallo, notificaciones, visualización de recursos…)

# Palabras clave

(Jenkins, Ansible, Grafana, Cisco IOS, Python, Telegram)

Contenido

[Agradecimientos 2](#_Toc132702283)

[Abstract 3](#_Toc132702284)

[Palabras clave 3](#_Toc132702285)

[Introducción 5](#_Toc132702286)

# Introducción

Cada vez que tenemos que pensar en hacer cambios en una red nos echamos las manos a la cabeza. Es normal que, al momento, nos vengan pensamientos de:

* ¿Cómo voy a configurar mis dispositivos de uno en uno?
* ¿Cómo voy a aprenderme todos los comandos que tiene cada uno de los sistemas operativos para dispositivos de red?
* ¿Y si no funciona lo que quiero hacer?

Es por ello que nació la idea de Infraestructura como código (**IaC**). De esta manera, podíamos hacer cambios en nuestra infraestructura de forma simple y fácil, pero todavía se nos escapa un posible escenario: ¿Y si no funciona lo que quiero hacer?. Para dar una solución a este problema, tenemos que hablar de **GitOps**.

En resumidas cuentas, lo que GitOps nos permite es tener un mejor control del software que vayamos escribiendo, usando nuestro repositorio (GitHub, Gitlab, Bitbucket) como **fuente de la verdad**. Básicamente la fuente de la verdad significa que todo el código fuente con el que automaticemos la infraestructura vendrá del repositorio y no permitiremos cambios "desde fuera". Implícitamente con GitOps también tendremos un concepto añadido: Guardar todos los datos (incluidos secrets) en el repositorio. De esta manera tendremos una infraestructura "portable". Así, si usamos GitOps en conjunción con herramientas CI/CD (Jenkins, Ansible) tendremos un flujo de trabajo automatizado, estable y que puede reaccionar a errores haciendo rollbacks.

En este proyecto concretamente utilizaremos Netbox junto a GitHub como fuentes de la verdad (Netbox para configuración de dispositivos de red y GitHub para almacenar playbooks), AWX (Ansible Tower) y GitHub Actions para aprovechar el CI/CD, pyTest/pyATS para hacer pruebas sintácticas/de configuración y por último, Grafana para monitorización.

# Objetivos

El objetivo de este proyecto es montar y configurar un clúster local de Kubernetes con todas las partes necesarias para crear el espacio de trabajo que luego aprovechará el equipo de SRE. También se harán algunas pruebas para comprobar el correcto funcionamiento del “workflow”.

# Requisitos previos

Para poder desarrollar este proyecto se han utilizado los siguientes hosts:

|  |  |  |  |
| --- | --- | --- | --- |
| **Hosts** | **CPUs** | **RAM** | **Versión S.O.** |
| Ubuntu on WSL | 2 | 8GB | 22.04 |
| Docker (WSL Backend) | 2 | 8GB | 4.18.0 |
| Kubernetes on Docker | 2 | 8GB | 1.25.4 |

## Instalación de Ubuntu on WSL

Para poder montar toda la infraestructura necesaria podemos hacerlo de dos formas: utilizando Linux como sistema operativo principal o utilizando el Subsistema de Windows para Linux (WSL). En mi caso utilizaré esta última opción por:

* Facilidad de uso.
* Evitar concurrencia de datos al hacer dual boot.
* Evitar conflictos al hacer dual boot.
* Mucha documentación disponible.

Instalaremos Ubuntu on WSL siguiendo [la guía oficial](https://ubuntu.com/tutorials/install-ubuntu-on-wsl2-on-windows-10) de Ubuntu. – añadir al anexo

## Instalación de Docker Desktop with WSL Backend

Para instalar Docker Desktop con el WSL backend seguiremos [esta otra guía](https://docs.docker.com/desktop/windows/wsl/). – añadir al anexo