# Monitorización de sistemas IoT en un entorno DevOps

## Resumen

El presente proyecto se enfoca en la aplicación de metodología DevOps en el ámbito de Internet de las cosas (IoT) mediante el uso de Vagrant en un entorno simulado. El objetivo principal es desplegar el stack ELK (Elasticsearch, Logstash y Kibana) y un clúster de servidores de K3S, cuyos pods emularán dispositivos del Edge. La implementación de este proyecto busca aprovechar la virtualización y el uso de contenedores para lograr una gestión eficiente y escalable de los entornos de desarrollo en el contexto de IoT.

El stack ELK es ampliamente reconocido por su capacidad para recopilar, procesar y visualizar grandes volúmenes de registros y datos. Al desplegar este stack en un entorno simulado de IoT mediante Vagrant, se brinda a los equipos de desarrollo y profesionales de IoT la oportunidad de explorar y evaluar su funcionalidad y rendimiento en un entorno controlado. Esto les permitirá adquirir una comprensión más profunda de las capacidades y limitaciones del stack ELK en el contexto de IoT, así como desarrollar estrategias efectivas para la gestión y análisis de datos a gran escala.

Por otro lado, el clúster de servidores de K3S es una distribución ligera de Kubernetes que facilita la creación y gestión de clústeres de contenedores. En este proyecto, el clúster de K3S se utilizará para simular dispositivos IoT en el entorno virtualizado. Al emular dispositivos a través del clúster de K3S, se podrá evaluar cómo interactúan y se comunican estos dispositivos en un entorno escalable y eficiente. Esta simulación permitirá a los equipos de desarrollo y profesionales de IoT analizar y optimizar el rendimiento de las aplicaciones y servicios IoT, así como explorar estrategias de escalabilidad y tolerancia a fallos.

El uso de Vagrant en este proyecto tiene como objetivo principal lograr una configuración consistente y reproducible de los entornos de desarrollo. Vagrant facilita la definición y compartición de archivos de configuración que describen el entorno deseado, asegurando que todos los miembros del equipo tengan una base común y evitando problemas causados por configuraciones inconsistentes. Esto ahorra tiempo y reduce la posibilidad de errores en la configuración de los entornos.

Además, la metodología DevOps se aplica de manera integral en este proyecto. DevOps se basa en la colaboración estrecha entre los equipos de desarrollo y operaciones para acelerar la entrega de software, mejorar la calidad y garantizar la estabilidad del sistema. En el contexto de IoT, la aplicación de DevOps es fundamental para gestionar la complejidad de los entornos distribuidos y heterogéneos. La virtualización proporcionada por Vagrant y la gestión de contenedores ofrecida por el clúster de K3S son pilares clave para la implementación exitosa de DevOps en IoT.

En resumen, este proyecto tiene como objetivo aplicar metodología DevOps en el ámbito de Internet de las cosas (IoT) mediante la utilización de Vagrant en un entorno simulado. El despliegue del stack ELK y el clúster de servidores de K3S permitirá a los equipos de desarrollo y profesionales de IoT explorar y evaluar el rendimiento de estas tecnologías en un entorno controlado.

## Abstract

This project focuses on the application of DevOps methodology in the Internet of Things (IoT) domain using Vagrant in a simulated environment. The main objective is to deploy the ELK stack (Elasticsearch, Logstash, and Kibana) and a K3S server cluster that will emulate devices. The implementation of this project aims to leverage virtualization and containerization to achieve efficient and scalable management of development environments in the context of IoT.

The ELK stack is widely recognized for its ability to collect, process, and visualize large volumes of logs and data. By deploying this stack in a simulated IoT environment using Vagrant, development teams and IoT professionals are provided with an opportunity to explore and evaluate its functionality and performance in a controlled environment. This will enable them to gain a deeper understanding of the capabilities and limitations of the ELK stack in the context of IoT and develop effective strategies for managing and analyzing large-scale data.

On the other hand, the K3S server cluster is a lightweight distribution of Kubernetes that facilitates the creation and management of container clusters. In this project, the K3S cluster will be used to simulate IoT devices in the virtualized environment. By emulating devices through the K3S cluster, it will be possible to assess how these devices interact and communicate in a scalable and efficient environment. This simulation will allow development teams and IoT professionals to analyze and optimize the performance of IoT applications and services, as well as explore scalability and fault tolerance strategies.

The use of Vagrant in this project aims to achieve consistent and reproducible configuration of development environments. Vagrant facilitates the definition and sharing of configuration files that describe the desired environment, ensuring that all team members have a common foundation and avoiding issues caused by inconsistent configurations. This saves time and reduces the likelihood of errors in environment configuration.

Furthermore, DevOps methodology is applied comprehensively in this project. DevOps is based on close collaboration between development and operations teams to accelerate software delivery, improve quality, and ensure system stability. In the context of IoT, the application of DevOps is crucial for managing the complexity of distributed and heterogeneous environments. The virtualization provided by Vagrant and container management offered by the K3S cluster are key pillars for the successful implementation of DevOps in IoT.

To sum up, this project aims to apply DevOps methodology in the Internet of Things (IoT) domain using Vagrant in a simulated environment. The deployment of the ELK stack and the K3S server cluster will enable development teams and IoT professionals to explore and evaluate the performance of these technologies in a controlled environment.

## Índice

Contenido

[Monitorización de sistemas IoT en un entorno DevOps 1](#_Toc135998012)

[Resumen 2](#_Toc135998013)

[Abstract 3](#_Toc135998014)

[Índice 4](#_Toc135998015)

[Introducción 5](#_Toc135998016)

[Contexto 5](#_Toc135998017)

[Objetivo 5](#_Toc135998018)

[Justificación 5](#_Toc135998019)

[Fundamentos teóricos: 5](#_Toc135998020)

[IoT 5](#_Toc135998021)

[DevOps 5](#_Toc135998022)

[Vagrant 5](#_Toc135998023)

[ELK Stack 6](#_Toc135998024)

[K3S 6](#_Toc135998025)

[Metodología 6](#_Toc135998026)

[Configuración del entorno simulado: 6](#_Toc135998027)

[Integración de Mosquitto y Logstash: 6](#_Toc135998028)

[Pruebas y validación: 6](#_Toc135998029)

[Resultados Obtenidos 6](#_Toc135998030)

[Referencias 7](#_Toc135998031)

[Glosario 7](#_Toc135998032)

## Introducción

### Contexto

El contexto actual de desarrollo de software se caracteriza por la creciente demanda de implementaciones rápidas, estables y escalables, lo cual ha impulsado la adopción de metodologías y prácticas que permitan abordar estos desafíos. En este sentido, los entornos DevOps han surgido como una aproximación eficaz para mejorar las capacidades de desarrollo, integración y despliegue continuo de aplicaciones.

La implementación exitosa de un entorno DevOps requiere el cumplimiento de varios requisitos fundamentales. En primer lugar, se debe establecer una estrecha colaboración entre los equipos de desarrollo y operaciones, fomentando una comunicación fluida y una visión compartida del proyecto. Es esencial contar con herramientas y procesos automatizados que permitan la integración y entrega continua de software, agilizando el ciclo de vida del desarrollo y minimizando los tiempos de entrega.

Entre los puntos fuertes de la aproximación DevOps se encuentra la mejora en la calidad del software, ya que la automatización de pruebas y la implementación continua permiten identificar y corregir errores de manera temprana. Asimismo, la capacidad de respuesta ante cambios y la rápida adaptación a nuevas necesidades del negocio son características destacadas de estos entornos. Al tener una mayor visibilidad y control sobre el ciclo de vida de la aplicación, los equipos pueden tomar decisiones informadas y reducir el time to market.

Sin embargo, también existen algunos puntos débiles en esta aproximación. La complejidad y la curva de aprendizaje asociadas con la adopción de herramientas y prácticas DevOps pueden ser un desafío para algunos equipos. Además, la falta de una cultura colaborativa y resistencia al cambio pueden obstaculizar la implementación exitosa de un entorno DevOps. Es fundamental fomentar una mentalidad de mejora continua y promover la colaboración entre los diferentes roles involucrados.

En este contexto, Vagrant ha demostrado ser una herramienta valiosa dentro de los entornos DevOps. Vagrant es una herramienta de código abierto que permite la creación y gestión de entornos de desarrollo virtualizados de manera reproducible. Al proporcionar una configuración declarativa y fácil de usar, Vagrant permite a los equipos establecer entornos de desarrollo consistentes y portátiles, evitando problemas causados por diferencias en la configuración del entorno.

Además, Vagrant se integra perfectamente con otras herramientas y tecnologías utilizadas en los entornos DevOps, como Ansible o Docker. Esto permite la automatización de tareas de aprovisionamiento y configuración, así como la gestión de contenedores, facilitando la creación de entornos de desarrollo completos y listos para ser desplegados en cualquier infraestructura.

Por otro lado, la monitorización de logs se ha convertido en una práctica esencial. Los logs son registros que contienen información valiosa sobre el funcionamiento de un sistema, incluyendo errores, eventos y actividades relevantes. Los sistemas de monitorización de logs permiten recopilar, analizar y visualizar estos registros de manera centralizada, proporcionando una visión completa y en tiempo real del estado y rendimiento del sistema. Esto facilita la detección temprana de problemas, la toma de decisiones informadas y la optimización de la infraestructura. Algo que sirve de apoyo dentro de la metodología DevOps.

En conclusión, los entornos DevOps han surgido como una respuesta efectiva a los desafíos actuales en el desarrollo de software. Al cumplir con requisitos clave como la colaboración, la automatización y la entrega continua, los entornos DevOps mejoran las capacidades de desarrollo y permiten una mayor eficiencia en la entrega de software. Vagrant, como herramienta de virtualización, desempeña un papel importante en esta aproximación, proporcionando entornos de desarrollo consistentes y portátiles que facilitan la adopción de prácticas DevOps y la implementación exitosa de proyectos, adicionalmente, la monitorización de logs se ha vuelto fundamental para garantizar la fiabilidad y eficiencia de los sistemas en entornos empresariales.

### Objetivo

El principal objetivo es desarrollar un Vagrantfile que se encargue de la

Uno de los principales objetivos de este proyecto es implementar el stack ELK (Elasticsearch, Logstash y Kibana) en el entorno desplegado. Elasticsearch proporcionará una plataforma robusta para el almacenamiento y búsqueda de los registros generados por los dispositivos IoT simulados. Logstash será utilizado para la ingestión y procesamiento de estos registros, permitiendo su transformación y enriquecimiento antes de ser almacenados en Elasticsearch. Kibana se utilizará como herramienta de visualización y análisis para extraer información valiosa a partir de los datos recopilados. El despliegue exitoso del stack ELK en el entorno simulado será un hito clave en el proyecto.

Además, se pretende desplegar un cluster de K3S para simular los dispositivos IoT. K3S es una distribución ligera de Kubernetes que facilita la creación y gestión de clústeres de contenedores. Los contenedores desplegados en este clúster representarán los dispositivos IoT, generando y enviando logs a través del protocolo MQTT. Para ello, se utilizará Mosquitto como servidor MQTT, que actuará como intermediario en la comunicación entre los dispositivos y el resto del sistema. Los dispositivos enviarán sus registros a Mosquitto utilizando una jerarquía de topicos.

Para capturar y enviar los registros de MQTT a Logstash, se desplegará un agente Filebeat en la máquina virtual de MQTT. Filebeat es un componente de la suite de Elastic que permite recopilar, enviar y procesar registros de manera eficiente. El agente Filebeat estará configurado para leer los mensajes de los tópicos de MQTT y enviarlos a Logstash a través de una conexión TCP. Esto permitirá una integración fluida entre Mosquitto, Filebeat y Logstash, asegurando que los registros generados por los dispositivos IoT sean enviados y procesados de manera adecuada en el stack ELK.

En resumen, los objetivos principales de este proyecto de Trabajo de Fin de Máster son desplegar un entorno utilizando Vagrant que contenga el stack ELK, un cluster de K3S con contenedores que simulan ser dispositivos IoT y establecer una arquitectura que permita la recopilación, procesamiento y análisis de los registros generados por estos dispositivos. La integración de Mosquitto, Filebeat y Logstash asegurará la captura y transferencia eficiente de los registros de MQTT a través del stack ELK, proporcionando una solución completa y escalable para la gestión de registros en un entorno de IoT simulado.

### Justificación

## Fundamentos teóricos:

### IoT

### DevOps

DevOps es una metodología y conjunto de prácticas que combina los aspectos del desarrollo de software (Dev) y las operaciones (Ops) en un enfoque integrado. Se basa en la colaboración estrecha y continua entre los equipos de desarrollo y operaciones con el objetivo de acelerar la entrega de software, mejorar la calidad y garantizar la estabilidad del sistema.

La filosofía detrás de DevOps es romper las barreras tradicionales entre los equipos de desarrollo y operaciones, fomentando la comunicación, la colaboración y la responsabilidad compartida. Se enfoca en automatizar los procesos, desde el desarrollo hasta la implementación y el monitoreo, para lograr una entrega continua y confiable de software.

Los principios clave de DevOps incluyen:

* Colaboración: Los equipos de desarrollo y operaciones trabajan juntos de manera estrecha y colaborativa, compartiendo conocimientos, objetivos y responsabilidades.
* Automatización: Se automatizan los procesos de desarrollo, pruebas, implementación y monitoreo para reducir los errores, mejorar la eficiencia y acelerar los tiempos de entrega.
* Entrega continua: Se busca entregar software en incrementos pequeños y frecuentes, lo que permite una retroalimentación rápida, una detección temprana de problemas y una mayor capacidad de respuesta a los cambios.
* Infraestructura como código: La infraestructura se trata como código, utilizando herramientas de gestión y aprovisionamiento automatizado para facilitar la configuración y despliegue de la infraestructura necesaria.
* Monitorización y retroalimentación: Se establecen mecanismos de monitoreo continuo del rendimiento y la calidad del software, lo que permite una retroalimentación rápida y la mejora continua del sistema.

Los beneficios de la implementación de DevOps incluyen una mayor agilidad en el desarrollo y despliegue de software, una mejor calidad del producto, una mayor eficiencia operativa y una mayor satisfacción del cliente.

En resumen, DevOps es una metodología que busca integrar los equipos de desarrollo y operaciones para lograr una entrega continua de software de alta calidad, a través de la colaboración, la automatización y una cultura de mejora continua.

### Vagrant

#### Ventajas de uso de Vagrant

Integrar Vagrant en un entorno de desarrollo de Internet de las cosas (IoT) presenta una serie de ventajas significativas que pueden impulsar la eficiencia y la calidad de los proyectos. Vagrant, una herramienta de administración de entornos de desarrollo, permite la creación y configuración de máquinas virtuales de manera sencilla y automatizada. Al utilizar Vagrant en el desarrollo de proyectos IoT, los equipos de desarrollo pueden experimentar numerosos beneficios.

Una de las ventajas clave de utilizar Vagrant en el desarrollo de IoT es la capacidad de reproducir entornos de manera consistente. Dado que el desarrollo de IoT a menudo involucra una combinación de hardware y software específicos, asegurar que todos los miembros del equipo tengan entornos idénticos puede ser un desafío. Sin embargo, con Vagrant, los desarrolladores pueden definir y compartir archivos de configuración que describen el entorno de desarrollo deseado. Esto garantiza que todos los desarrolladores tengan una base común y evita problemas causados por configuraciones inconsistentes, lo que ahorra tiempo y reduce la posibilidad de errores.

Además, Vagrant facilita la colaboración y el intercambio de proyectos de IoT. Al proporcionar una configuración fácil de seguir, los equipos pueden compartir sus proyectos con otros miembros o incluso con la comunidad en general. Esto fomenta la colaboración y acelera el proceso de desarrollo, ya que los desarrolladores pueden compartir rápidamente sus avances y permitir que otros los exploren y contribuyan. La capacidad de compartir entornos de desarrollo en forma de archivos Vagrant simplifica enormemente la configuración de nuevos miembros del equipo, permitiéndoles unirse rápidamente al proyecto sin problemas.

Otra ventaja significativa de utilizar Vagrant en el desarrollo de IoT es la capacidad de probar y depurar aplicaciones en diferentes entornos. En el desarrollo de IoT, es esencial garantizar que las aplicaciones funcionen correctamente en diversas configuraciones de hardware y software. Vagrant permite a los desarrolladores crear fácilmente múltiples máquinas virtuales con configuraciones específicas para simular diferentes entornos. Esto les permite probar y depurar sus aplicaciones en diferentes plataformas, sistemas operativos y configuraciones de red, lo que resulta en un producto final más robusto y compatible.

La automatización es otro beneficio clave que Vagrant aporta al desarrollo de IoT. Al aprovechar la funcionalidad de aprovisionamiento de Vagrant, los desarrolladores pueden automatizar la instalación y configuración de software y herramientas dentro de las máquinas virtuales. Esto ahorra tiempo y esfuerzo al eliminar la necesidad de configurar manualmente cada entorno de desarrollo individualmente. Además, la automatización garantiza que los entornos sean consistentes y reproducibles, lo que reduce la posibilidad de errores humanos y facilita la escalabilidad a medida que el proyecto crece.

La seguridad también es una consideración crítica en el desarrollo de IoT, y Vagrant puede ayudar en este aspecto. Al utilizar máquinas virtuales aisladas, los desarrolladores pueden garantizar que sus proyectos se ejecuten en entornos seguros y controlados. Esto es especialmente relevante en el desarrollo de aplicaciones IoT, donde la seguridad y la privacidad de los datos son primordiales. Al utilizar Vagrant, los equipos de desarrollo pueden tener la tranquilidad de que sus proyectos se ejecutan en entornos virtuales protegidos, lo que minimiza el riesgo de filtraciones de datos o ataques malintencionados.

En resumen, la incorporación de Vagrant en el entorno de desarrollo de IoT ofrece numerosas ventajas. La capacidad de reproducir entornos de manera consistente, facilitar la colaboración y el intercambio de proyectos, probar y depurar aplicaciones en diferentes configuraciones, automatizar tareas y garantizar la seguridad de los entornos son solo algunas de las ventajas clave. Al aprovechar las características de Vagrant, los equipos de desarrollo pueden mejorar la eficiencia, la calidad y la colaboración en el desarrollo de proyectos de IoT, acelerando el tiempo de comercialización y proporcionando productos más robustos y seguros.

### ELK Stack

### K3S

## Metodología

### Configuración del entorno simulado:

Descripción detallada de cómo se estableció el entorno simulado utilizando Vagrant, incluyendo la configuración de las máquinas virtuales, la conexión de Mosquitto y Logstash, y la implementación del stack ELK y el clúster de servidores de K3S.

### Integración de Mosquitto y Logstash:

Explicación de cómo se realizó la conexión entre Mosquitto y Logstash utilizando el plugin "logstash-input-mqtt" para la recepción de mensajes MQTT y su posterior procesamiento y envío a los destinos deseados.

### Pruebas y validación:

Descripción de las pruebas realizadas para verificar el funcionamiento correcto del entorno simulado, el procesamiento de los datos y la integración entre los componentes.

## Resultados Obtenidos

Presentación de los resultados obtenidos durante las pruebas y la validación, junto con un análisis de los datos recopilados. Se pueden incluir métricas de rendimiento, eficiencia y escalabilidad, así como ejemplos de visualización y análisis de datos realizados con el stack ELK.

## Referencias

## Glosario