



# Migración y optimización de un e-commerce de un servidor simple a un clúster de AWS

## TFG – Administración de redes y sistemas operativos

Alumno : Adrián Martínez Rodríguez

Consultor : Jaime Jofre Bravo

# Cronología

01. — 02. — 03. — 04.

## Análisis

Es necesario observar las necesidades actuales y la situación actual de la infraestructura

## Objetivos

Es necesario definir los objetivos que queremos cumplir con la nueva infraestructura.

## Investigación

Es necesario investigar sobre las tecnologías a usar para lograr los objetivos definidos.

## Despliegue

Desplegar la nueva infraestructura tras definir los objetivos y estudiar la mejor forma de construirla.



# Situación actual de e-commerces



## Infraestructura limitada

Utilizan servidores simples donde tienen todos los servicios en él.

## Nula escalabilidad

Estos servidores no permiten aumentar en tiempo real las prestaciones de los mismos o añadir nuevos servidores para aguantar picos de carga.

## Sin Alta Disponibilidad

Si un servicio deja de responder todo el sistema deja de funcionar.

## Poca Tolerancia a errores

Si existe un error en el servicio, todo el sistema deja de funcionar.

## Aumento de clientes y ventas online

La tendencia de los consumidores es cada vez realizar más compras por internet.

## Mayor competencia online

Cada vez existen más tiendas electrónicas para realizar las compras lo cual significa que ante un mal servicio podemos perder un cliente por el funcionamiento de nuestra web.

# Diferencias entre **infraestructuras**

## Antigua **Infraestructura**

Todos los servicios se encuentran bajo una misma máquina donde responde un servidor Nginx y un servidor de MYSQL para la resolución del Prestashop.



## Nueva **Infraestructura**

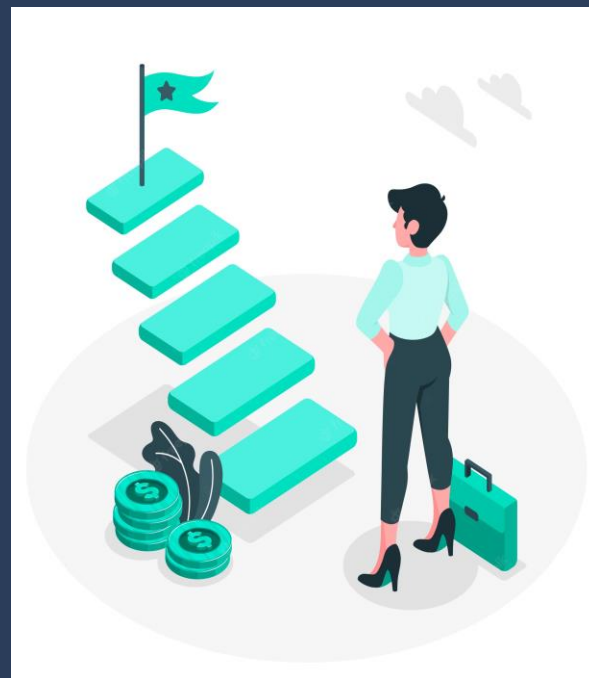
Los servicios están individualizados y clonados en múltiples máquinas para garantizar la alta disponibilidad, también se añaden mejoras como backups, balanceadores de carga, servicios de monitorización, etc...



# Metas a conseguir con el proyecto

Las metas a conseguir en este proyecto son las siguientes:

1. Mejorar la infraestructura de un e-commerce proporcionándole alta disponibilidad, tolerancia a errores y escalabilidad.
2. Añadir servicios como CDN, servicios de monitorización, backups, etc...
3. Conocer y Aplicar tecnologías emergentes al e-commerce.
4. Demostrar que no se trata de un proceso imposible de realizar.



# Tecnologías a **usar**



docker

## Docker

Sistema de contenedores que contiene los servicios necesarios para desplegar aplicaciones



Uno de los tres proveedores más grandes de hosting de la actualidad



## Cloudflare

Gestión de DNS de un dominio, CDN y SSL.



## Prometheus

Software para recopilar métricas de los servidores.



## Grafana

Software para graficar las métricas obtenidas.

# Docker

**Docker** se trata de un proyecto open source que permite crear una imagen con todos los servicios necesarios para poder correr una aplicación, esto significa que no es necesario instalar todos los servicios cada vez que se quiere desplegar si no simplemente montar la imagen en un contenedor y esto nos permitirá lanzar la aplicación. Entre sus beneficios se encuentran:

- Facilidad para migrar servicios.
- Rapidez a la hora de implementar una aplicación en un nuevo servidor.
- Evitar errores de versiones al siempre tener los servicios necesarios instalados en la imagen.
- Solo se necesita tener docker instalado y utilizar la imagen creada para desplegar la aplicación.

CONTENEDOR

Aplicación A

Aplicación B

Aplicación C

Binarios y  
Librerías

Binarios y  
Librerías

Binarios y  
Librerías

Docker

SO huésped

Infraestructura

# AWS

AWS se trata de uno de los 3 grandes proveedores de alojamiento que nos proporciona muchos sistemas para poder desplegar nuestra infraestructura en ellos algunas de ellas son:

## EC2

Máquinas escalables de la propia infraestructura de aws.

## VPC

Red privada virtual de la infraestructura totalmente configurable, contiene grupos de seguridad, subredes, etc...

## ECR

Repositorio de imágenes de docker dentro de nuestro usuario de AWS.

## ECS

Servicio de creación de clusters de máquinas a través de contenedores

## EFS

Repositorios de ficheros que se pueden montar como volumen en cualquier máquina de la infraestructura.

## RDS

Servicio de bases de datos relacionales de AWS que permite montar un cluster de bases de datos.

## S3

Unidades de almacenamiento que nos proporciona aws en nuestra cuenta.

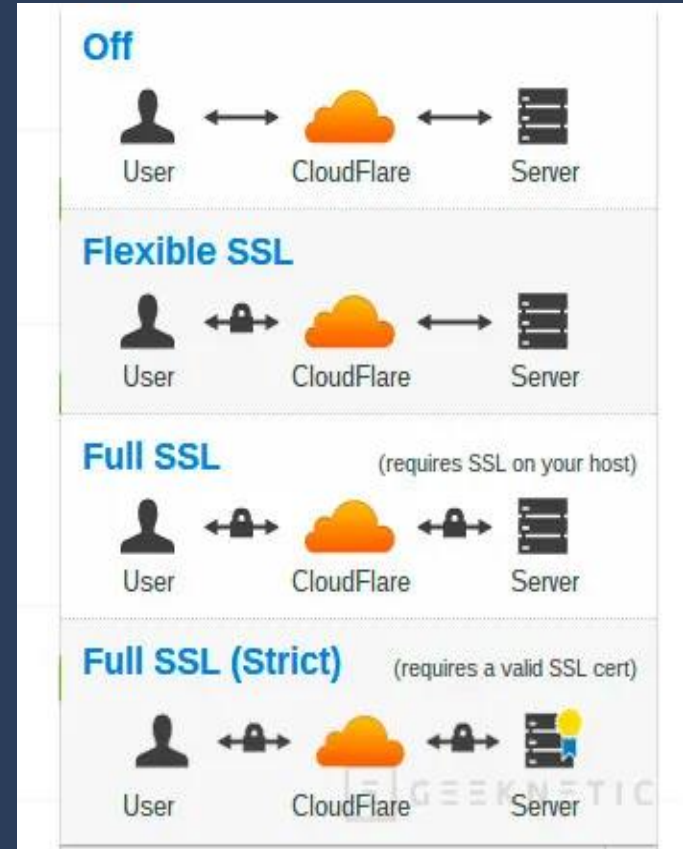


# Cloudflare

Cloudflare se trata de un servicio de resolución de dominios que permite la resolución mediante una CDN que cachea el contenido de nuestra web, a mayores Cloudflare también proporciona servicios de WAF, AntiDDOS, SSL, Firewall, etc

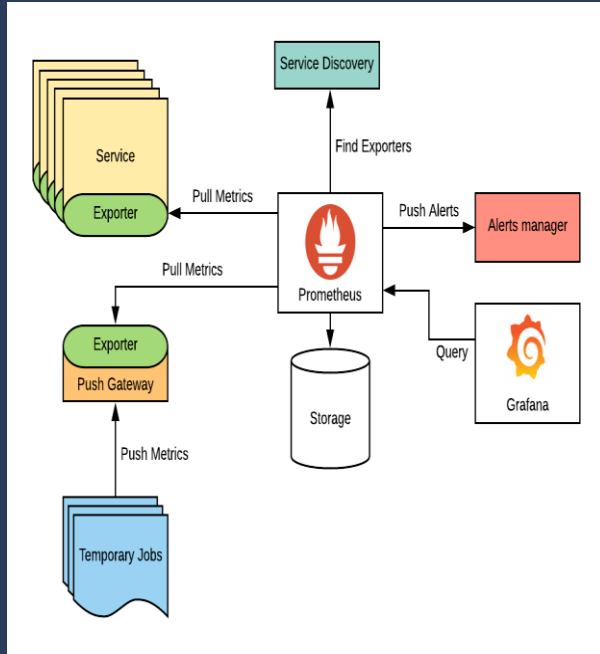
Entre sus beneficios podemos encontrar:

- Ocultación a nivel de resolución DNS de la ip del servidor web que aloja la web.
- Resolución geolocalizada a través de su CDN.
- Protección del cliente que accede a los servicios mediante certificados SSL.
- Reglas de firewall y waf personalizables para proteger los servidores de acciones malintencionadas.
- Sistemas AntiDDos para evitar bots o ataques de denegación de servicio.



# Prometheus

**Prometheus** es una herramienta open-source que permite recolectar métricas de otros servicios o servidores para crear alertas o enviarlas a servicios de graficación para que muestren estos datos de forma gráfica. Prometheus necesita que de exporters en las máquinas para que estos le envíen las métricas y él se encarga de almacenar esas métricas y distribuirlas según se requiera.

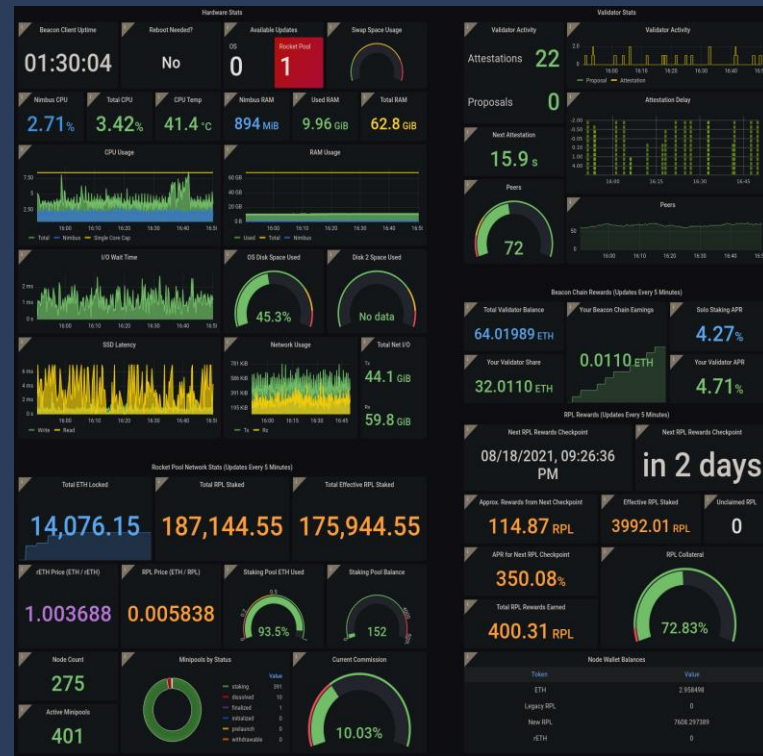


- Gran variedad de exporters según las métricas que se quieran obtener.
- Alta compatibilidad con servicios como docker, kubernetes y clusters.
- Personalización de alertas según las necesidades del servicio.
- Compatible con los grandes sistemas de graficación como Grafana, Kibana, etc...

# Grafana

Grafana es un software open-source que a través de la obtención de métricas permite graficarlas en paneles configurables por el usuario. Permite la creación de alertas, definición de umbrales de riesgo, plantillas por defecto según las métricas obtenidas y también una alta configuración de los dashboards según las necesidades.

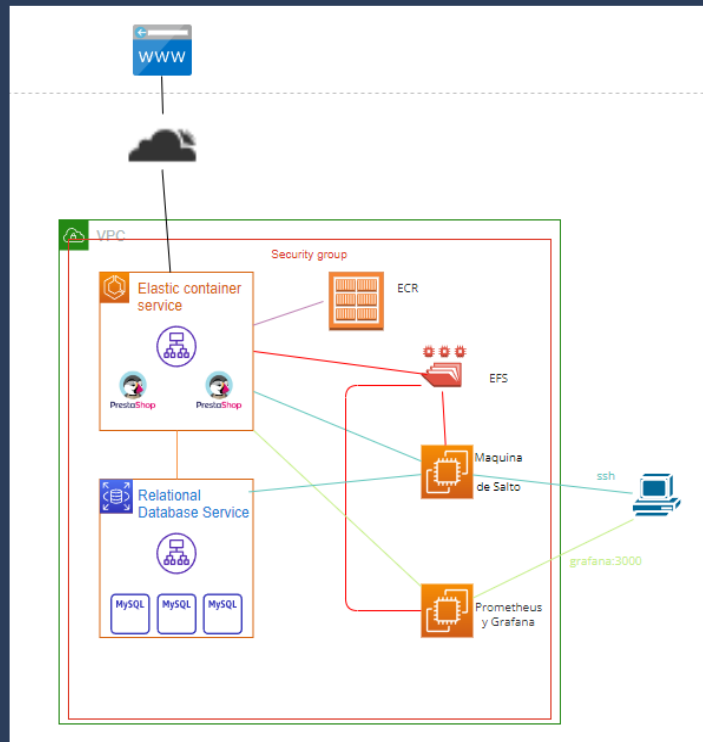
- Integración con sistemas de recopilación de métricas como Prometheus, Zabbix, etc...
- Paneles personalizables y plantillas predefinidas según los exporters o métricas requeridas.
- Personalización de alertas o umbrales según los requerimientos del usuario.
- Fácil visualización y obtención de información en tiempo real del estado de nuestra infraestructura.



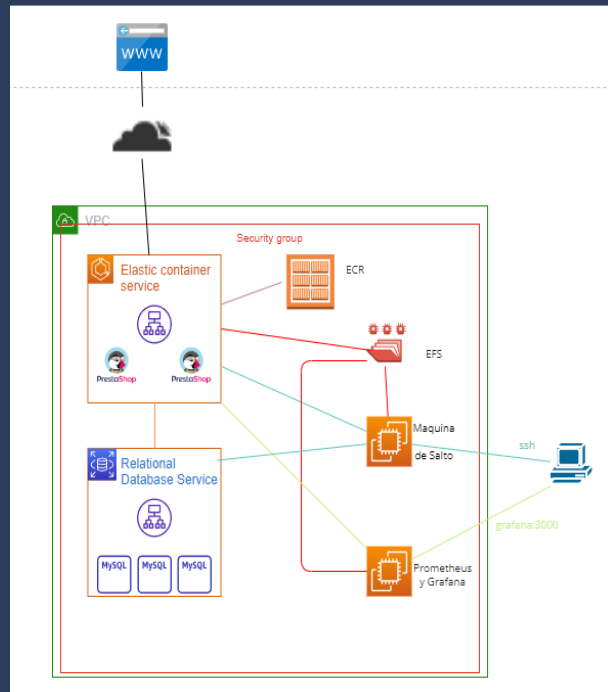
# Esquema de la nueva infraestructura

La nueva infraestructura será desplegada como se muestra en el esquema con los siguientes servicios.

1. El dominio apuntará hacia la CDN de Cloudflare donde responderá bajo un certificado SSL que permitirá el tráfico por el protocolo https.
2. Cloudflare estará apuntando a los contenedores de ECS donde se desplegará un cluster de varias máquinas con el software de Prestashop instalado.
3. Este cluster ECS obtendrá los datos tendrá montado un volumen EFS para cada máquina desplegada donde se encuentran los ficheros de nuestra web.
4. El cluster ECS tendrá conectividad contra el clusters de bases de datos gestionado desde el servicio RDS.



# Esquema de la nueva infraestructura



5. El cluster **ECS** también obtendrá las imágenes de docker necesarias para correr el Prestashop desde el servicio **ECR**.
6. Se desplegará una máquina **EC2** para alojar los servicios de **Prometheus** y **Grafana** que obtendrá las métricas desde los diferentes exporters instalados en los diferentes servicios. Esta máquina dejará conectividad externa para poder acceder al **Grafana** y ver las gráficas.
7. Se desplegará una máquina **EC2** de salto que será la única puerta de acceso a la infraestructura por SSH desde el exterior, permitirá la administración de todos los servicios al tener conectividad con todos.
8. Todos los servicios estarán dentro de una **VPC** con un **security group** cerrado al exterior solo permitiendo el acceso por **https** al **ECS**, por **SSH** a la máquina de salto y al panel de **Grafana**.

# Logros Alcanzados

## Alta disponibilidad

Gracias a los clusters con múltiples servidores y a los balanceadores de carga.

## Tolerancia a fallos

Por tener clusters en caso de fallo de una máquina las otras seguirían resolviendo.



## Escalabilidad

Al tratarse de una infraestructura fragmentada nos permite aumentar nuestros servicios si fuese necesario

## Seguridad

Se limita el acceso a la infraestructura a solo una red externa y a solo una máquina interna.

# Mejoras implementadas



# Dificultades encontradas



- Funcionamiento interno de los contenedores de **docker**, para obtener los servicios persistentes ante un reinicio.
- Encontrar la infraestructura que mejor se adaptaba a la idea entre todos los servicios de **AWS**.
- Comprensión del funcionamiento de las **VPC** y de los **security groups** dentro de **AWS**, así como conocer los servicios ofertados.
- Configuración de volúmenes de datos **EFS** y bases de datos **RDS** en **ECS**.
- Configuración base de datos importada en **RDS** para responder bajo el dominio principal del proyecto.
- Configuración de los **ECS** para desplegar y poder incrementar el número de servidores sin necesidad de realizar cambios.
- Servicios de **AWS** fuera de la capa gratuita.
- Instalación de los exporters de **Prometheus** en servicios como **RDS**.
- Creación de plantilla personalizada en **Grafana** con las métricas y valores que se deseen.




# Puntos positivos y Negativos

## Positivos

1. Mejora del rendimiento, seguridad, escalabilidad de la infraestructura.
2. Uso de tecnologías y o servicios actuales.
3. Amplio abanico de posibilidades en la creación de la nueva infraestructura.
4. Más control por parte de cliente sobre los servicios.
5. Configuración personalidad según las necesidades.

## Negativos

1. Se necesita tener un conocimiento bastante extenso para poder llevar la migración a cabo.
  2. Existen sobrecostos y necesidad de vigilar la facturación de la nueva infraestructura.
  3. Se necesita de la administración de sistemas por parte de un especialista.
  4. Necesidad de conocer gran parte de los servicios disponibles para poder escoger la mejor solución.
- 

# Conclusiones

- El proyecto me ha ayudado a obtener una visión y unos conocimientos sobre las nuevas tecnologías utilizadas para los servicios web.
- Este tipo de acciones, aunque pudieran parecer simples a simple vista, tienen bastante más complejidad por detrás de la aparente.
- No es una acción que pueda realizar cualquier persona que lleve un e-commerce, es necesario tener un conocimiento de administración de sistemas y de los sistemas utilizados.
- Es factible crear una empresa que se encargue de realizar estos trabajos para tiendas online que no contienen con administradores de sistemas en su plantilla.
- Mejora el rendimiento de la web, así como la capacidad de adaptación a fechas como navidades, Black Friday gracias a su escalabilidad.
- Facilita la posible actualización a nuevos sistemas como podrían ser Kubernetes.
- No se trata de un servicio barato, ni simple de utilizar.
- Ha sido satisfactorio lograr el objetivo de realizar la migración y aplicar sus mejoras y poder comprobar en línea el correcto funcionamiento del servicio.

# Gracias!

## Padre y Hermano

Por inculcarme la pasión por las nuevas tecnologías desde pequeño y enseñarme mucho de lo que aún a día de hoy uso y se.

## Madre

Por el simple hecho de ser mi madre, y haber obtenido tanto cariño y apoyo durante toda mi vida.

## Amigos

Por estar en los malos momentos y ayudarme a desconectar durante estos años en momentos de agobios y estrés.

## Especialmente a Alicia

Por toda la motivación y ayuda recibida, y por ser el pilar que ha soportado mi vida durante estos años.

# FIN

¿Tienen alguna consulta?

amartinezrod@uoc.edu



**Alumno:** Adrián Martínez Rodríguez

**Consultor:** Jaume Jofre Bravo

**Año:** 2022-2023