

# Prestashop en alta disponibilidad en la nube AWS para Back In Black

## Migración servidor On premises a la nube pública de AWS

*Por Alberto Gómez, AWS Certified Solutions Architect, abril 2023*

### Contenido

1. Descripción general .....	2
2. Evaluación de la Infraestructura.....	2
2.1 Auditoría de la infraestructura actual.....	2
3. Selección de los Servicios de AWS necesarios .....	3
4. Configuración y migración. ....	5
4.1. Migración de los datos On-Premises a la nube de AWS.....	6
4.2 Configuración y Optimización.....	6
5. Puesta en producción. ....	7
6. Conclusión final. ....	7

## 1. Descripción general.

El objetivo principal de este proyecto fue migrar la infraestructura "on premises" de Back In Black Snowboards a la nube pública de Amazon Web Services (AWS) a raíz de un problema de espacio físico en los almacenes y las constantes interrupciones del servicio por parte del proveedor de internet. Se decidió estudiar las diferentes posibilidades de mejora con el fin de migrar todos los servidores a una nube pública. Debido a mi experiencia en dichos servicios, se optó por utilizar AWS y se ha determinado que yo seré el líder del proyecto.

## 2. Evaluación de la Infraestructura.

Este paso fue crítico en el proceso de migración de cara a planificar la estrategia a seguir. Se analizó en detalle el servidor y sus componentes para comprender su estado, configuración y capacidad.

### 2.1 Auditoría de la infraestructura actual.

Antes de comenzar necesitaba una visión completa de la infraestructura on-premises y, de esta manera, poder tomar decisiones informadas y mucho más precisas sobre las necesidades.

1. Identificación del servidor: El servidor web estaba compuesto de cuatro máquinas virtuales. Dos de ellas que contenían el sistema operativo y gestionaban los servidores web. Los clientes acceden a dichas máquinas y son distribuidos utilizando otra máquina con un balanceador de carga, obteniendo una mayor tolerancia a los fallos y una mejor gestión en épocas de alta demanda. Aun así, pude comprobar que la capacidad no era suficiente para garantizar el servicio con calidad, especialmente en los meses de invierno donde la demanda es un 300% mayor.
2. Hardware y recursos: El servidor web se encuentra en el almacén de la compañía, instalado en un armario rack de grandes dimensiones y limitando el uso de ese espacio. Está compuesto por dos equipos, uno de ellos dividido en diferentes máquinas como ya mencionamos anteriormente y otro equipo utilizado como servidor de almacenamiento en el que se guardaban las copias de seguridad y servía a su vez como almacenamiento compartido entre todos los equipos del personal de tienda. Se han documentado las especificaciones de hardware del servidor, incluyendo procesador, cantidad de memoria RAM, capacidad de almacenamiento, número de núcleos, etc., determinando que son insuficientes para el tamaño actual de la web y su infraestructura. Se hace necesario el aumento de memoria RAM, así como un procesador más moderno con mayor número de núcleos.

3. Sistema operativo y software: registro de todo el software y de sus aplicaciones instaladas, incluyendo versiones y licencias. El sistema operativo es CentOS y tiene instalado Apache HTTP Server License version 2.0, PHP License version 3.01, MySQL client License version 2. La virtualización se realiza utilizando Proxmox Version 8.0-1 y por último el servidor que se encarga de las copias de seguridad utiliza Proxmox Backup Server Version 3.0.
4. Configuración de red: la infraestructura de red se encuentra contratada a un proveedor externo, así como un router especial para empresas y una IP fija que suponen un coste elevado mensual que pudimos suprimir, con el consiguiente ahorro, al no necesitar dicha infraestructura. Para obtener mayor nivel de tolerancia a fallos cada equipo integraba dos tarjetas de red, en el caso de que alguna de ellas fallara.
5. Seguridad y rendimiento: se evaluaron las políticas de seguridad implementadas en el servidor, como cortafuegos, antivirus, controles de acceso y configuraciones de seguridad del sistema operativo y se han examinado las políticas de respaldo y recuperación de datos implementadas. Además, se han analizado los registros de eventos de los servidores para identificar problemas anteriores y patrones de errores. Por último, utilicé herramientas de monitoreo de rendimiento para evaluar la utilización de recursos y el rendimiento del servidor en diferentes momentos y así tener una idea más precisa de cuando es necesario añadir capacidad a estos y el hardware necesario.

Por tanto, habiendo evaluado la infraestructura, el objetivo era replicarla y mejorarla utilizando los servicios de AWS.

### 3. Selección de los Servicios de AWS necesarios

El objetivo principal fue identificar los servicios AWS que nos permitieran replicar la infraestructura existente on premises. Mediante la combinación de estos servicios logramos mayor escalabilidad, redundancia y rendimiento. Los servicios utilizados finalmente fueron los siguientes:

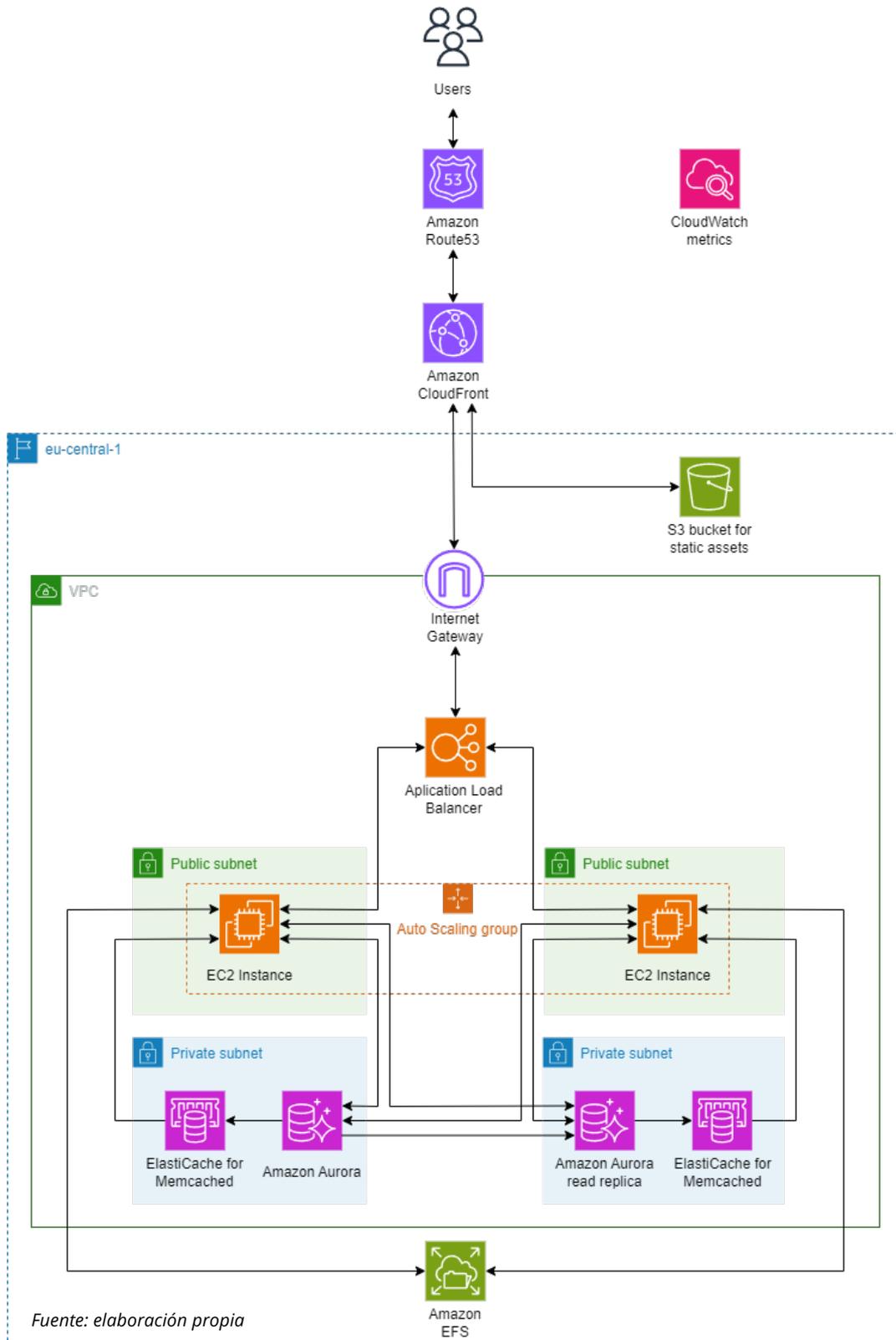
- Amazon EC2: utilizamos instancias de EC2 para alojar los servidores de Prestashop, en conjunto con un grupo de auto escalado de entre 2 a 4 instancias (según la demanda) del tipo t3a.medium.
- Amazon RDS (Relational Database Service): RDS nos permitió migrar fácilmente la base de datos ya existente al ser compatible con MySQL, en este caso y, debido a su rendimiento hemos optado por utilizar Amazon Aurora. La instancia necesaria para

sostener la demanda de la web será una db.r6g.large. Se hizo uso de las opciones que RDS ofrece de respaldo, estableciendo copias de seguridad diarias. Además, gracias a la implementación de un clúster Multi-AZ se obtuvo replicación síncrona y conmutación por error automática en caso de falla de una instancia.

- Amazon S3: se han almacenado todos los recursos estáticos, como imágenes, CSS y archivos multimedia e, integrándolo con CloudFront se acelera la velocidad de entrega, a la vez que se libera carga en las instancias de EC2.
- Amazon ElastiCache: se encarga de cachear las consultas de base de datos y mejorar el rendimiento de PrestaShop. Para ello, se usó Memcached como motor de caché.
- Amazon Application Load Balancer (ALB): una pieza fundamental de la arquitectura, dividiendo el tráfico entre las diferentes instancias de EC2 y proporcionando balanceo de carga y alta disponibilidad.
- Amazon VPC (Virtual Private Cloud): se muestra necesario para aislar los recursos y definir la topología de red deseada, incluyendo las subredes públicas y privadas.
- Amazon Route 53: imprescindible para gestionar el sistema de nombres de dominio (DNS).
- AWS Identity and Access Management (IAM): desde su panel gestionamos el acceso y la seguridad de los recursos de AWS.
- Amazon CloudWatch: para garantizar la salud de la web se hace uso de las alarmas y monitorizaciones del rendimiento de las instancias de EC2, bases de datos RDS y otros recursos.

#### 4. Configuración y migración.

Configurar una arquitectura para una aplicación web basada en Prestashop en AWS implica desarrollar un entorno escalable, seguro y de alto rendimiento. Para ello, una vez seleccionados los servicios, se creó el diseño de la arquitectura a utilizar.



## 4.1. Migración de los datos On-Premises a la nube de AWS.

Para la migración de los datos utilizamos AWS DataSync, el cual proporciona una forma sencilla y eficiente de transferir los datos, al mismo tiempo garantiza la integridad y coherencia de estos. Es especialmente útil para las necesidades continuas de sincronización de datos entre los recursos locales y de AWS.

Después de la transferencia inicial, se validó que los datos en AWS coincidieran con los datos del servidor local. La idea principal era realizar un "Green/Blue deployment", dejando la web antigua funcionando mientras configuraba y optimizaba todos los recursos de la nube. Al programar varias transferencias diarias me aseguraba que la coherencia de los datos era exacta y constante.

## 4.2 Configuración y Optimización.

Tras configurar toda la infraestructura y asegurarme de una correcta sinergia de todos los recursos, llegó el momento de optimizarlos tanto para el rendimiento como para el control de costos. Esta parte es esencial para garantizar operaciones eficientes y rentables.

El primer paso fue revisar que todos los tipos de instancias que fueran apropiados para las cargas de trabajo, se optimizó el almacenamiento con las características de rendimiento adecuadas (IOPS aprovisionadas) para las necesidades de la web, así como la habilitación las snapshots de volumen y las políticas de ciclo de vida para automatizar las copias de seguridad y reducir los costos de almacenamiento.

Después, implementé el balanceador de carga y las políticas de auto escalado para así ajustar automáticamente la cantidad de las instancias según los patrones de tráfico, garantizando la capacidad adecuada en todo momento y una distribución del tráfico de manera uniforme para lograr un mejor rendimiento y tolerancia a fallas.

El siguiente paso fue supervisar las instancias de Amazon RDS y ajustar su tamaño según la actividad real de la base de datos. Además, se implementó el modo Multi-AZ para alta disponibilidad y conmutación por error, utilizando réplicas de lectura para descargar tráfico de la instancia principal, además de programar snapshots diarias.

Finalmente, se contrató el plan Enterprise para obtener las recomendaciones de Trusted Advisor sobre optimización de costes, seguridad y rendimiento y se realizaron los presupuestos y alarmas para recibir notificaciones cuando los costes excedan los umbrales predefinidos.

## 5. Puesta en producción.

Las pruebas son cruciales para garantizar que el sitio de comercio electrónico funcione correctamente en el entorno de AWS. Esto implicó realizar pruebas rigurosas posteriores a la migración. Por ello se testeó que el sitio funcionara a la perfección y se corrigieron todos los errores que fuimos detectando. Además, tuve que brindar capacitación al equipo de operaciones para familiarizarlo con los servicios y las mejores prácticas de AWS. Tras dos semanas de testeado se decidió pasar la plataforma de la nube a producción y se realizó entre las 3:00 y las 4:00 am, periodo en el cual tan solo estuvimos inactivos durante 40 minutos.

## 6. Conclusión final.

Tras varios meses de haber concluido la migración podemos afirmar que ha sido un éxito. Se ha logrado minimizar el tiempo de inactividad en casi 83 horas en los últimos 7 meses. Además, ha habido un ahorro de costes de casi un 15% respecto a los servidores On-Premises y los servicios que teníamos contratados. Este ahorro se debe principalmente a la contratación de Savings Plans y Reserved Instances en AWS que nos permite ahorrarnos de media un 37% con respecto al modelo On-Demand.

Las mejoras también se han traducido en mayor número de ventas, especialmente en países extranjeros, con casi un 275% más. Gracias a una mayor inversión en motores de búsqueda, estrategias de marketing y mayor velocidad del sitio en ubicaciones lejanas. Hemos logrado ser referentes a nivel nacional en la venta online de Snowboard y hemos obtenido un 96% de satisfacción de los clientes.