
IAW - Práctica de ampliación 1

IAW - Implantación de Aplicaciones Web

Curso 2025/2026

Índice general

- 1 Práctica de ampliación: Réplicas Maestro-Eslavo y Maestro-Maestro en MySQL Server** **1**
- 1.1 Réplicas de MySQL para conseguir alta disponibilidad 1
- 1.2 Proxies para MySQL 1
 - 1.2.1 MySQL Router 2
 - 1.2.2 ProxySQL 2
 - 1.2.3 MariaDB MaxScale 2
- 1.3 Entregables 2

- 2 Créditos** **4**

- 3 Referencias** **5**

- 4 Licencia** **6**

1 Práctica de ampliación: Réplicas

Maestro-Eslavo y Maestro-Maestro en MySQL Server

Crea otra máquina virtual en [Amazon Web Services \(AWS\)](#) con la última versión de [Ubuntu Server](#). En esta nueva máquina vamos a instalar otra instancia de [MySQL Server](#). El objetivo de esta práctica es crear una arquitectura de **alta disponibilidad** que sea **escalable** y **redundante**, de modo que si un servidor [MySQL](#) falla por algún motivo, nuestra aplicación web podrá conectar con el otro servidor [MySQL](#).

Básicamente, podemos tener redundancia de un servicio de MySQL aplicando dos configuraciones:

- **Redundancia Maestro-Eslavo:** El servidor que actúa como maestro gestiona las operaciones de lectura/escritura y el servidor que actúa como esclavo sólo gestiona las operaciones de lectura.
- **Redundancia Maestro-Maestro:** Ambos servidores gestionan las operaciones de lectura/escritura.

La arquitectura estará formada por:

- Un balanceador de carga, implementado con un [Apache HTTP Server](#) configurado como [proxy inverso](#).
- Una capa de *front-end*, formada por dos servidores web con [Apache HTTP Server](#).
- Una capa de *back-end*, formada por dos servidores [MySQL](#).

Necesitará crear cinco máquinas virtuales:

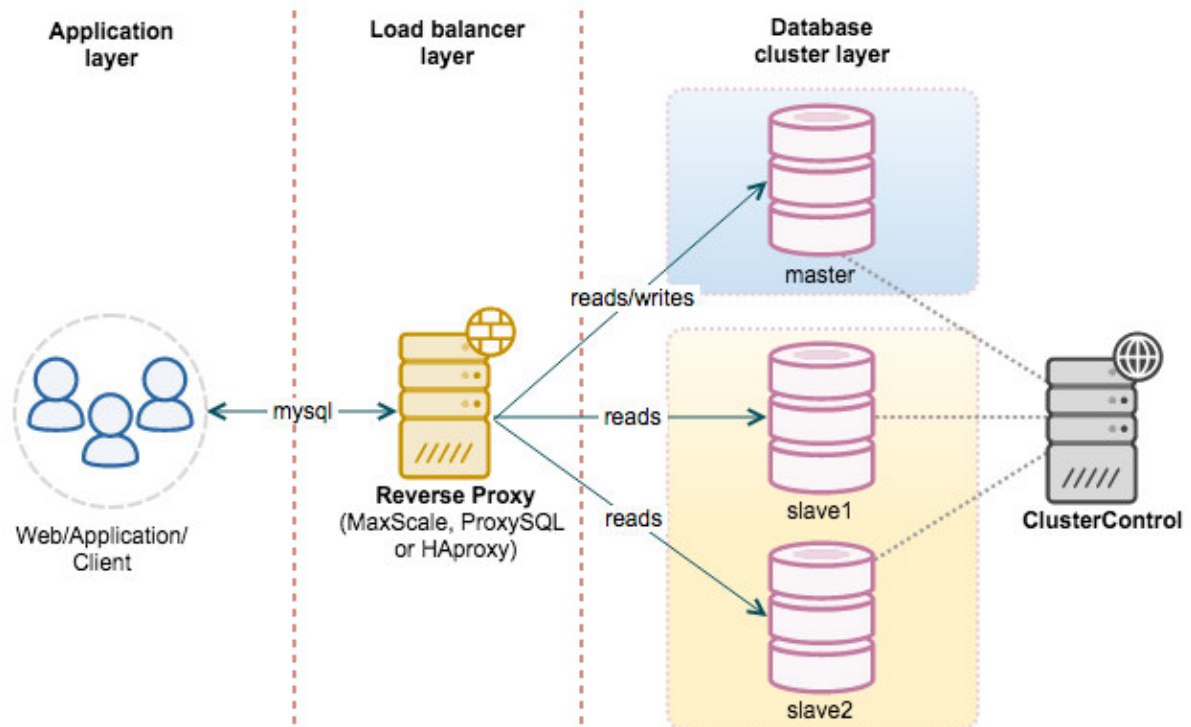
- Balanceador.
- Frontal Web 1.
- Frontal Web 2.
- Servidor de Base de Datos 1.
- Servidor de Base de Datos 2.

1.1. Réplicas de MySQL para conseguir alta disponibilidad

- [MySQL replication for High Availability](#). [pdf]. SeveralNines.
- [MySQL replication for High Availability](#). [web]. SeveralNines.

1.2. Proxies para MySQL

- [Choosing a database proxy for MySQL and MariaDB](#).



1.2.1. MySQL Router

- [Documentación oficial de MySQL Router.](#)

1.2.2. ProxySQL

- [Documentación oficial de ProxySQL.](#)

1.2.3. MariaDB MaxScale

- [Documentación oficial de MariaDB MaxScale.](#)

1.3. Entregables

En esta práctica habrá que entregar un **documento técnico** con la descripción de los pasos que se han llevado a cabo.

El documento debe incluir **como mínimo** lo siguientes contenidos:

- URL del repositorio de GitHub donde se ha alojado el documento técnico escrito en [Markdown](#).

- Tenga en cuenta que el aprovisionamiento de las máquinas virtuales se realizará mediante un *script* de *bash*. Cada máquina usará su propio *script*. El contenido de cada uno de los *scripts* deberá ser incluido en el documento y **deberá describir qué acciones se han ido realizando en cada uno de ellos**.
- El documento deberá incluir cuáles han sido los comandos de Vagrant que ha necesitado utilizar durante el desarrollo de la práctica.

2 Créditos

Las imágenes utilizadas en esta guía se han obtenido de [SeveralNines](#).

3 Referencias

- [VirtualBox](#).
- [Ubuntu Server](#).
- [LAMP Stack](#).
- [PHP](#).
- [Apache HTTP Server](#).
- [MySQL Server](#).
- [Move MySQL to a separate Cloud Database Server](#).
- [How to set up MySQL master-master replication](#).
- [How to set up master-slave replication in MySQL](#).
- [How to optimize WordPress performance with MySQL replication on Ubuntu](#).
- [MySQL Web Reference Architectures. Building Massively Scalable Web Infrastructure](#). Oracle.
- [How to design highly available open source database environments](#). Muy bueno.

4 Licencia

Esta página forma parte del curso Implantación de Aplicaciones Web de José Juan Sánchez y su contenido se distribuye bajo una licencia Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional.