

Manual de Despliegue de la Aplicación Web

‘Zebranalytics’

Equipo 404

Fecha de publicación:

Viernes 26 de Abril de 2024

Índice

Introducción	3
Propósito del Documento	3
Alcance del Despliegue	3
Definiciones y Acrónimos	3
Requisitos	4
Requisitos mínimos de Hardware	4
Requisitos de Software	4
Dependencias Externas	4
Configuraciones Necesarias	4
Entorno de Despliegue	5
Diagrama de despliegue	5
Procedimiento de Despliegue	6
Seguridad	8
Referencias	8

Introducción

Propósito del Documento

Este documento tiene como finalidad establecer una guía estructurada para el despliegue de la aplicación Zebranalytics. Su importancia radica en proporcionar al equipo técnico de Zebrands una referencia detallada que asegure una implementación coherente, eficiente y segura de la aplicación en el entorno de producción. El manual pretende facilitar el proceso de despliegue inicial, así como ser un recurso para actualizaciones y mantenimiento futuro.

Alcance del Despliegue

El presente manual cubre el despliegue de la aplicación Zebranalytics dentro de un servidor Linux, especificando la configuración necesaria del entorno de ejecución, incluyendo Node.js y PM2, así como la configuración del servidor web y proxy inverso con Caddy. Detallará las etapas de conexión a la base de datos, la autenticación mediante Google OAuth 2.0 y la puesta en marcha del código de la aplicación utilizando Express.js.

Definiciones y Acrónimos

Node.js: Entorno de ejecución para JavaScript del lado del servidor.

PM2: Administrador de procesos para aplicaciones Node.js que permite una gestión eficiente de la ejecución y la carga de trabajo.

Caddy: Servidor web y proxy inverso automatizado que simplifica la configuración de encriptación y redireccionamiento.

Express.js: Framework para aplicaciones web de Node.js que permite una rápida configuración de rutas y servicios.

Google OAuth 2.0: Protocolo de autenticación que permite a los usuarios acceder a la aplicación con sus cuentas de Google.

MySQL: Sistema de gestión de bases de datos relacional.

DB: Base de Datos.

API: Interfaz de Programación de Aplicaciones.

SSL/TLS: Protocolos de seguridad para proporcionar comunicaciones seguras por una red de computadoras.

CI/CD: Integración Continua y Despliegue Continuo, prácticas de ingeniería de software que implican la actualización frecuente de aplicaciones.

Requisitos

Requisitos mínimos de hardware

Almacenamiento: 1 GB de espacio disponible en disco para la instalación y datos.

Requisitos de Software

Sistema Operativo: Distribución Ubuntu Linux.

Node.js: Versión LTS actual al momento del despliegue.

PM2: Última versión estable.

Caddy: Última versión estable, incluyendo módulos para SSL/TLS.

MySQL: Versión 5.7 o superior, con soporte para la última versión del motor InnoDB.

Git: Última versión estable para la obtención del código fuente desde el repositorio.

Dependencias Externas

Librerías de Node.js: Listado de paquetes npm necesarios especificados en el archivo package.json.

Google API Console: Acceso configurado para uso de Google OAuth 2.0 con las credenciales correspondientes.

Configuraciones Necesarias

Variables de Entorno: Establecer variables de entorno para configuraciones de base de datos (DB_HOST, DB_USER, DB_PASS), credenciales de Google OAuth (GOOGLE_CLIENT_ID, GOOGLE_CLIENT_SECRET).

Entorno de Despliegue

Diagrama de despliegue

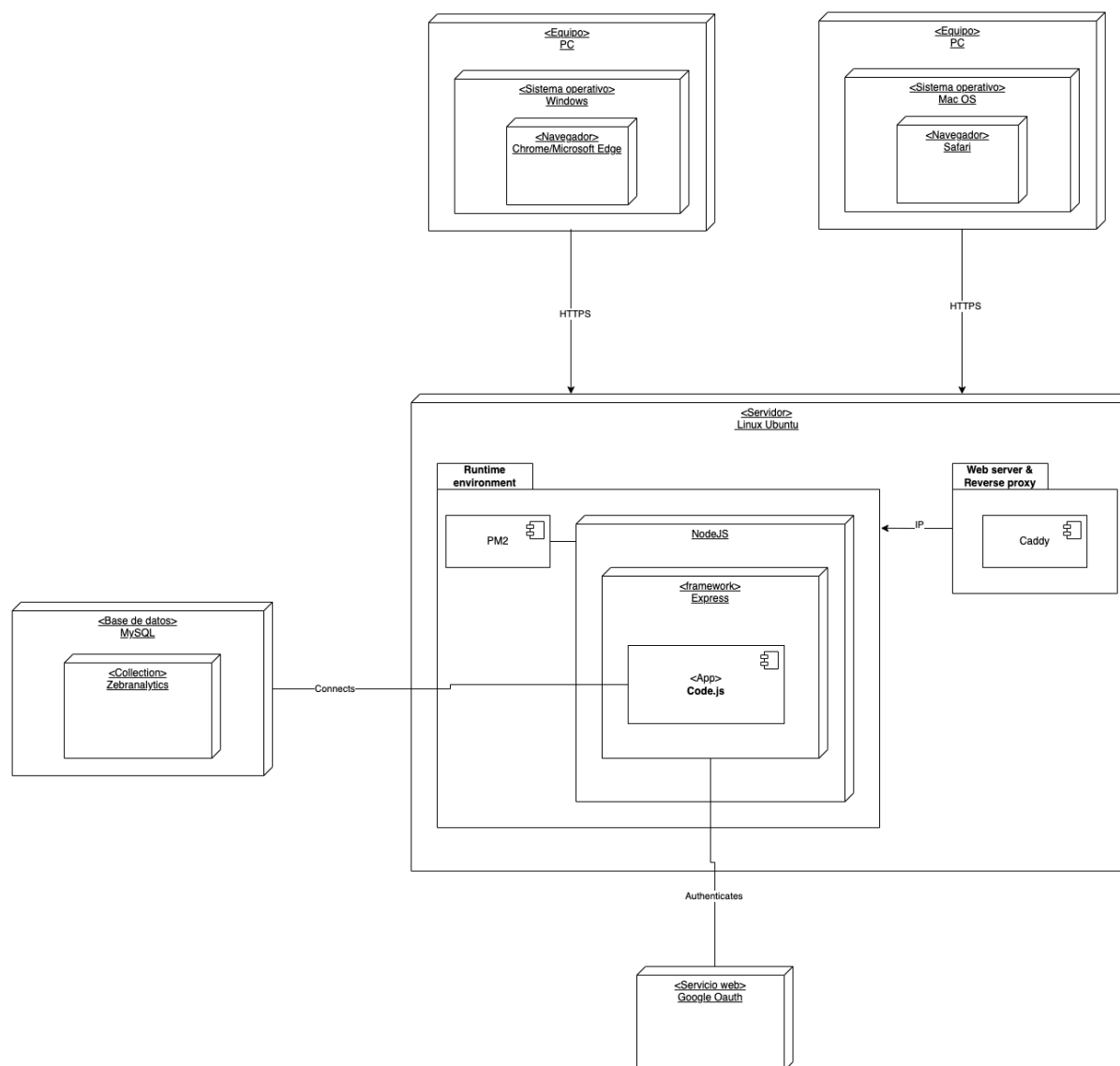


Figura 1. Diagrama de despliegue

Procedimiento de Despliegue

El primer paso en el despliegue de la aplicación Zebranalytics es establecer una conexión segura con el servidor de producción. Como se ilustra en la Figura 2, inicie sesión en el servidor como root, cree un usuario dedicado para la gestión de la aplicación y conceda los permisos necesarios. A continuación, instale las herramientas fundamentales, como Git y Node.js, que son cruciales para el manejo del código fuente y la ejecución de la aplicación. Los pasos detallados para estas tareas se muestran a continuación.

```
1 # Conectar a un servidor remoto por SSH
2 ssh root@<ip del servidor>
3
4 # Ingresar contraseña
5
6 # Crear un nuevo usuario
7 adduser <nombre de usuario>
8 # Darle privilegios de superusuario
9 usermod -aG sudo <nombre de usuario>
10
11 # Desconectar del servidor, y reconectarse con el nuevo usuario
12 exit
13 ssh <nombre de usuario>@<ip del servidor>
14
15 # Instalar Git
16 sudo apt update
17 sudo apt install git
18
19 # Instalar NVM (Node Version Manager)
20 curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.7/install.sh | bash
21
22 # Instalar Node.js 21
23 nvm install 21
24
25 # Verificar instalación de node y npm
26 node -v # should print 'v21.7.2'
27 npm -v # should print '10.5.0'
28
29 # Clonar su repositorio
30 git clone <url del repositorio>
31
```

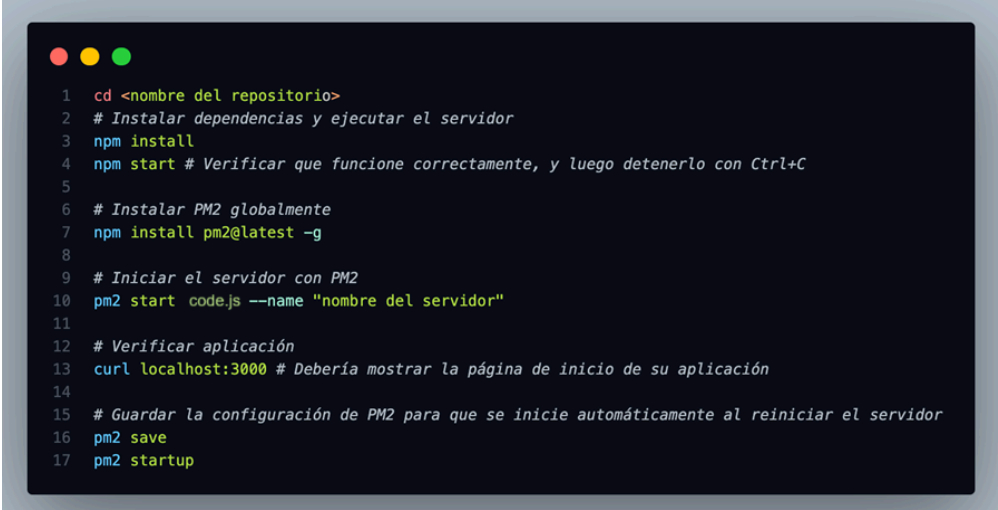
Figura 2. Configuración Básica del Servidor

El siguiente paso es establecer el sistema de gestión de la base de datos MySQL. Esto incluye la instalación del servidor MySQL, la configuración segura inicial, y la creación de la base de datos específica para la aplicación junto con un usuario dedicado. Es vital seguir las buenas prácticas de seguridad como se muestra en la Figura 3, para asegurar que la base de datos esté bien configurada y segura. Después de esto, se debe importar el archivo SQL anexo al manual para cargar la DB.

```
1 # Instalar MySQL
2 sudo apt update
3 sudo apt install mysql-server
4 sudo systemctl start mysql.service
5
6 # Habilitar contraseña root para MySQL
7 sudo mysql
8 ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY '<contraseña>';
9 exit
10
11 # Configuración inicial de MySQL
12 sudo mysql_secure_installation
13
14 # Crear una base de datos y un usuario
15 sudo mysql -u root -p
16 CREATE DATABASE <nombre de la base de datos>;
17 CREATE USER '<nombre de usuario>'@'localhost' IDENTIFIED BY '<contraseña>';
18 GRANT ALL PRIVILEGES ON <nombre de la base de datos>.* TO '<nombre de usuario>'@'localhost';
19 FLUSH PRIVILEGES;
20 exit
```

Figura 3. Instalación de MySQL en el servidor

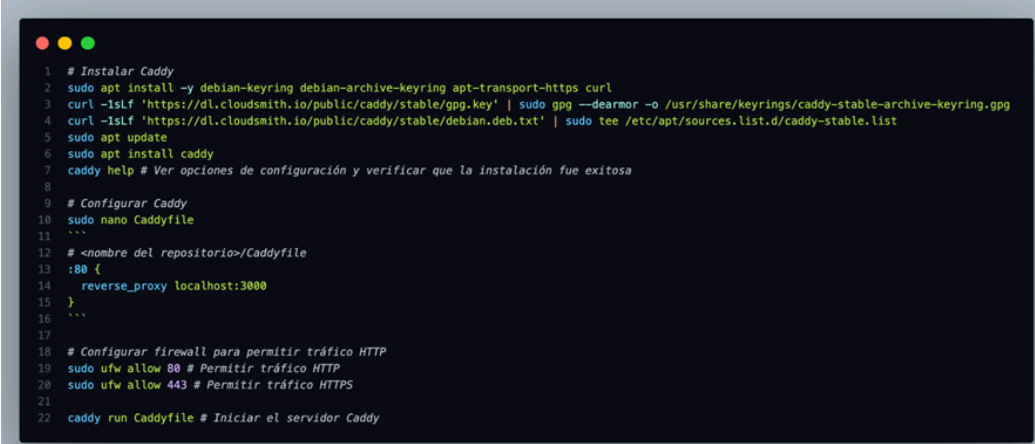
Después de la configuración de la base de datos, el siguiente paso es preparar el entorno de ejecución de la aplicación. Esto incluye la instalación de las dependencias necesarias, el uso de PM2 para gestionar el proceso de la aplicación, y asegurar su correcto funcionamiento como se detalla en la Figura 4. La utilización de PM2 es fundamental para garantizar que la aplicación se mantenga en ejecución de manera continua y que se reinicie automáticamente en caso de fallos, proporcionando así una mayor disponibilidad y estabilidad.

A terminal window with a dark background and light-colored text. It shows a series of commands for setting up a Node.js application with PM2. The commands are numbered 1 through 17. The window has three colored window control buttons (red, yellow, green) in the top-left corner.

```
1 cd <nombre del repositorio>
2 # Instalar dependencias y ejecutar el servidor
3 npm install
4 npm start # Verificar que funcione correctamente, y luego detenerlo con Ctrl+C
5
6 # Instalar PM2 globalmente
7 npm install pm2@latest -g
8
9 # Iniciar el servidor con PM2
10 pm2 start code.js --name "nombre del servidor"
11
12 # Verificar aplicación
13 curl localhost:3000 # Debería mostrar la página de inicio de su aplicación
14
15 # Guardar la configuración de PM2 para que se inicie automáticamente al reiniciar el servidor
16 pm2 save
17 pm2 startup
```

Figura 4. Ejecutar la aplicación con PM2

Con la aplicación en marcha, el paso final es configurar el servidor web y el proxy inverso. En esta etapa, se instala y se configura Caddy, que no solo servirá contenido estático y dinámico, sino que también manejará el cifrado SSL/TLS automáticamente. La correcta configuración del proxy inverso es crucial para dirigir el tráfico web entrante a su destino apropiado, como se ilustra en la Figura 5. Además, se deberán realizar ajustes en el firewall para permitir el tráfico a través de los puertos adecuados.

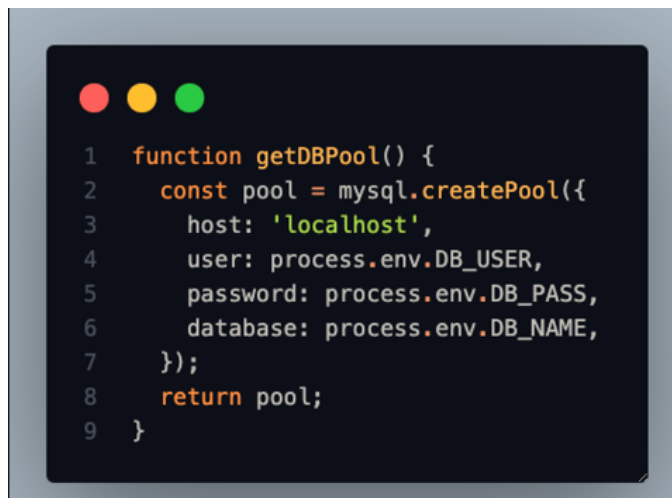
A terminal window with a dark background and light-colored text. It shows commands for installing Caddy, configuring it as a reverse proxy, and setting up the firewall. The commands are numbered 1 through 22. The window has three colored window control buttons (red, yellow, green) in the top-left corner.

```
1 # Instalar Caddy
2 sudo apt install -y debian-keyring debian-archive-keyring apt-transport-https curl
3 curl -sLf 'https://dl.cloudsmith.io/public/caddy/stable/gpg.key' | sudo gpg --dearmor -o /usr/share/keyrings/caddy-stable-archive-keyring.gpg
4 curl -sLf 'https://dl.cloudsmith.io/public/caddy/stable/debian.deb.txt' | sudo tee /etc/apt/sources.list.d/caddy-stable.list
5 sudo apt update
6 sudo apt install caddy
7 caddy help # Ver opciones de configuración y verificar que la instalación fue exitosa
8
9 # Configurar Caddy
10 sudo nano Caddyfile
11 '''
12 # <nombre del repositorio>/Caddyfile
13 :80 {
14     reverse_proxy localhost:3000
15 }
16 '''
17
18 # Configurar firewall para permitir tráfico HTTP
19 sudo ufw allow 80 # Permitir tráfico HTTP
20 sudo ufw allow 443 # Permitir tráfico HTTPS
21
22 caddy run Caddyfile # Iniciar el servidor Caddy
```

Figura 5. Configuración de Servidor Web (Caddy)

Seguridad

- Es de suma importancia el no guardar datos sensibles para la conexión a la base de datos dentro del código fuente. La recomendación es que se mantengan estos datos en variables de entorno como se muestra en la figura 6.



```
1 function getDBPool() {
2   const pool = mysql.createPool({
3     host: 'localhost',
4     user: process.env.DB_USER,
5     password: process.env.DB_PASS,
6     database: process.env.DB_NAME,
7   });
8   return pool;
9 }
```

• **Figura 6.** Variables de entorno

Referencias

Node.js. (s. f.). Recuperado 25 de abril de 2024, de <https://nodejs.org/en/download/package-manager/current>

Virdó, H., & Drake, M. (2022, 11 julio). How To Install MySQL on Ubuntu 20.04. DigitalOcean. Recuperado 25 de abril de 2024, de <https://www.digitalocean.com/community/tutorials/how-to-install-mysql-on-ubuntu-20-04>

PM2 - Quick start. (s. f.). Keymetrics Inc. Recuperado 25 de abril de 2024, de <https://pm2.keymetrics.io/docs/usage/quick-start/>

Caddy. (s. f.). Caddy Documentation. Recuperado 25 de abril de 2024, de <https://caddyserver.com/docs/>

Juell, K., & Tagliaferri, L. (2021, 31 marzo). How To Set Up a Node.js Application for Production on Ubuntu 20.04. DigitalOcean. Recuperado 25 de abril de 2024, de <https://www.digitalocean.com/community/tutorials/how-to-set-up-a-node-js-application-for-production-on-ubuntu-20-04>