

# Manual de Desplegament

Aquest document descriu els passos necessaris per desplegar l'aplicació `2024_plomes_al_mar` en un servidor AWS, incloent-hi la instal·lació de tots els requisits de programari.

## Chapter 1. Configuració d'instàncies/màquines virtuals a AWS

En el present document, s'expliquen les accions bàsiques a AWS per configurar instàncies amb IPs públiques per poder accedir-hi des de qualsevol lloc. Accés al curs: [https://www.awsacademy.com/vforcesite/LMS\\_Login](https://www.awsacademy.com/vforcesite/LMS_Login)

## Chapter 2. Creació d'instància

Creem una instància des del botó Lanzar una instancia. Aquí s'ha de configurar la màquina que es vol crear. En el nostre cas:

1. Nombre y etiquetas
  1. Nom: El nom que vulguem .Imágenes de aplicaciones y sistemas operativos (Imagen de máquina de Amazon)
  2. Ubuntu
  3. Imágenes de máquina de Amazon (AMI): La que indiqui que és apta per la capa gratuïta.
2. Tipo de instancia: t2.nano o màxim, t2.micro
3. Par de claves (inicio de sesión): vockey
4. Configuraciones de red: Per defecte tot excepte:
  1. Marcar els checks de permetre trànsit http i https.

Finalment, donar al botó de Lanzar instancia.

## Chapter 3. Accedir a la instància

Un cop creada la instància, escollir el check de la instància. Anar al menú a acciones → connectar. Hi ha diferents maneres de connectar a la màquina:

1. Connexió de la instància EC2 s'entra per web com si fos una màquina isard.
2. Client SSH: haurem de descarregar el fitxer .PEM que apareix a la pàgina des d'on iniciem el lab. Descarreguem el PEM si estem a linux, o el PPK si és des de connexió des de Putty.

## Chapter 4. Creació de IP fixa a les instàncies

Per crear i assignar una IP fixa a la màquina (IP Elàstica), anar al menú de la dreta Red y seguridad → direcciones de IP elásticas i prémer el botó Crear o Asignar la dirección IP elástica.

### 4.1. Assignar la IP elàstica a una instància

Al llistat d'adreces IP elàstiques, s'ha d'escollir la IP que interressi. Després, anar al menú actions, escollir l'opció d'associar IP i a la pantalla, escollir la instància EC2 que s'ha creat i deixar la resta d'opcions per defecte.

Aquest document descriu els passos necessaris per desplegar l'aplicació `2024_plomes_al_mar` en un servidor AWS, incloent-hi la instal·lació de tots els requisits de programari.

## Chapter 5. Obrir seguretat

Seleccionar la instancia creada i als detalls anar a Seguretat → ex(sg-001d4fc225ff0e05c (launch-wizard-14) → Editar reglas de entrada → Agregar regla

Tenim que assignar a Tipo → TCP Personalizado, Intervalo de Puertos → 8080 i a la lupa del costat buscar el 0.0.0.0/0 i al final de tot a Guardar reglas.

## Requisits de Programari Necessari

- Servidor amb Ubuntu instal·lat
- Apache 2
- PHP 8.2 i extensions necessàries
- Composer
- Git
- Docker i Docker Compose

## Passos a Fer per Desplegar l'Aplicació

### Actualització i Instal·lació d'Apache

Primer actualitzem els paquets disponibles i instal·lem Apache:

- `sudo apt update`: Actualitza la llista de paquets disponibles.
- `sudo apt install apache2`: Instal·la el servidor web Apache.
- `sudo systemctl enable apache2`: Configura Apache perquè s'iniciï automàticament en arrencar el sistema.

# Configuració del Tallafocs

Actualitzem el sistema i configurem el tallafocs per permetre el tràfic web i SSH:

- `sudo apt update`: Actualitza el sistema operatiu.
- `sudo apt upgrade`: Actualitza el sistema operatiu.
- `sudo ufw status`: Comprova l'estat del tallafocs.
- `sudo ufw app list`: Llista les aplicacions configurades per UFW.
- `sudo ufw allow in "Apache Full"`: Permet el tràfic HTTP i HTTPS per Apache.
- `sudo ufw allow in "OpenSSH"`: Permet les connexions SSH.
- `sudo ufw allow 8080`: Per habilitar el port 8080 per a la API d'Imatges.
- `sudo ufw enable`: Activa el tallafocs UFW.

## Instal·lació de PHP

Instal·lem PHP i les extensions necessàries per al projecte:

- `sudo apt install php libapache2-mod-php php-mysql`: Instal·la PHP i les extensions necessàries per funcionar amb Apache i MySQL.

## Instal·lació del Projecte

Afegim el repositori de PHP, instal·lem la versió específica i clonem el projecte des de Git:

- `sudo apt update`: Actualitza el sistema operatiu.
- `sudo add-apt-repository ppa:ondrej/php`: Afegeix el repositori de PHP.
- `sudo apt update`: Actualitza el sistema operatiu.
- `sudo apt install php8.2 php8.2-cli php8.2-xml php8.2-mbstring php8.2-curl`: Instal·la PHP 8.2 i diverses extensions.
- `sudo apt install php8.2-xml php8.2-dom php8.2-curl php8.2-mysql php8.2-gd`: Instal·la PHP 8.2 i diverses extensions.
- `curl -sS https://getcomposer.org/installer | php`: Descarrega i instal·la Composer.
- `sudo mv composer.phar /usr/local/bin/composer`: Mou Composer a una ubicació accessible globalment.
- `cd /var/www/html`: Aquesta ordre canvia el directori de treball actual al directori /var/www/html, que és el directori arrel dels documents per al servidor web Apache.
- `sudo git clone https://git.copernic.cat/valls.berengueras.albert/2024_plomes_al_mar.git`: Clona el repositori del projecte.

# Projecte principal site

Preparació dins del projecte principal (site)

- `cd /var/www/html/2024_plomes_al_mar/site`: Canvia el directori de treball actual al directori site dins del projecte clonat 2024\_plomes\_al\_mar.
- `sudo update-alternatives --set php /usr/bin/php8.2`: Canvia la versió predeterminada de PHP que s'utilitza al sistema a PHP 8.2.
- `sudo composer install`: Instal·la totes les dependències del projecte.
- `sudo cp .env.example .env`: Copia el fitxer de configuració d'exemple .env.example a un fitxer .env nou, que és on s'emmagatzemen les configuracions específiques de l'entorn.
- `sudo php artisan key:generate`: Genera una clau d'aplicació nova per al projecte Laravel. Aquesta clau es fa servir per assegurar les sessions i altres dades xifrades.
- `sudo php artisan storage:link`: Crea un enllaç simbòlic des de public/storage a storage/app/public. Això és necessari perquè els fitxers emmagatzemats al directori d'emmagatzematge siguin accessibles públicament.

## Instal·lació de Docker

Instal·lem Docker i Docker Compose:

- `sudo apt-get update`: Actualitza la llista de paquets disponibles i les seves versions des dels dipòsits configurats al sistema.
- `sudo apt-get install apt-transport-https ca-certificates curl software-properties-common`: Instal·la paquets necessaris per utilitzar repositoris HTTPS.
- `cd ~`: Canvia el directori de treball actual al directori home de l'usuari.
- `curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg`: Descarrega i afegeix la clau GPG de Docker.
- `echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null`: Afegeix el repositori de Docker.
- `sudo apt-get update`: Actualitza la llista de paquets disponibles i les seves versions des dels dipòsits configurats al sistema.
- `sudo apt-get install docker-ce docker-ce-cli containerd.io`: Instal·la Docker.
- `sudo curl -L "https://github.com/docker/compose/releases/download/VERSION/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose`: Descarrega Docker Compose.
- `sudo chmod +x /usr/local/bin/docker-compose`: Fa Docker Compose executable.

## Configuració de Docker Compose

Creem el fitxer `docker-compose.yml` per configurar el servei MySQL:

- `cd /var/www/html/2024_plomes_al_mar/site`: Canvia el directori de treball actual al directori site

dins del projecte clonat 2024\_plomes\_al\_mar.

- `sudo nano docker-compose.yml`: Obre l'editor de text nano amb privilegis de superusuari per crear o editar el fitxer docker-compose.yml.

Dins del fitxer docker-compose.yml, afegeix el següent contingut per configurar un servei MySQL utilitzant Docker Compose:

```
services:
  mi_mysql:
    image: mysql:5.7
    container_name: mi_mysql
    environment:
      MYSQL_ROOT_PASSWORD: 1234
      MYSQL_DATABASE: plomesalmar
      MYSQL_USER: mysql
      MYSQL_PASSWORD: 1234
    ports:
      - "3306:3306"
    volumes:
      - mysql_data:/var/lib/mysql
```

```
mi_api:
  image: mysql:5.7
  container_name: mi_api
  environment:
    MYSQL_ROOT_PASSWORD: 1234
    MYSQL_DATABASE: apiimagenes
    MYSQL_USER: mysql
    MYSQL_PASSWORD: 1234
  ports:
    - "3307:3306"
  volumes:
    - api_data:/var/lib/mysql
```

```
volumes:
  mysql_data:
  api_data:
```

- `sudo docker compose up -d`: Per llençar el docker.
- `sudo docker ps`: Verificar que els contenidors estan funcionant.

## Configuració .env

Configurem l'arxiu .env per connectar a la nostra bd:

- `sudo nano .env`: Obre l'editor de text nano amb privilegis de superusuari per editar el fitxer .env,

que conté les configuracions d'entorn.

Cambiar aquestes líneas del fitxer:

```
APP_NAME="Plomes al mar"  
APP_ENV=local  
APP_KEY=base64:w+WN7gBjCTxXtIaU2YNddP3p9gHTDyII60oFoNqZtvM=  
APP_DEBUG=true  
APP_URL=http://localhost  
URL_API=http://localhost:8080
```

```
API_APP_TOKEN=UhheSf0a3wbKFDqS9imnGcDDhVsdON4PDOQvphHC9FzUpkL50nmNXnZVy1p8  
API_PATH=C:/Users/Alfred/Desktop/2024_plomes_al_mar/API_image  
API_PATH=E:/Repositoris/0_Daw/2024/2024_plomes_al_mar/API_image
```

```
LOG_CHANNEL=stack  
LOG_DEPRECATED_CHANNELS=null  
LOG_LEVEL=debug
```

```
DB_CONNECTION=mysql  
DB_HOST=127.0.0.1  
DB_PORT=3306  
DB_DATABASE=plomesalmar2024  
DB_USERNAME=plomesalmar2024  
DB_PASSWORD=plomesalmar2024
```

A aquestes:

```
APP_NAME="Plomes al mar"  
APP_ENV=local  
APP_KEY=base64:/tm8ksDjp99IEhSv9whX+iBRc2YJOA7n93BRVnBUjds=  
APP_DEBUG=true  
APP_URL=http://ip privada  
URL_API=http://ip privada:8080
```

```
API_APP_TOKEN=UhheSf0a3wbKFDqS9imnGcDDhVsdON4PDOQvphHC9FzUpkL50nmNXnZVy1p8  
API_PATH=/var/www/html/2024_plomes_al_mar/API_image
```

```
LOG_CHANNEL=stack  
LOG_DEPRECATED_CHANNELS=null  
LOG_LEVEL=debug
```

```
DB_CONNECTION=mysql
DB_HOST=ip privada
DB_PORT=3306
DB_DATABASE=plomesalmar
DB_USERNAME=mysql
DB_PASSWORD=1234
```

## Creació d'un servei systemd per a Docker Compose

Creació i configuració d'un servei systemd per gestionar i aixecar els contenidors Docker:

- `sudo nano /etc/systemd/system/plomesdemar.service`: Obre l'editor de text nano amb privilegis de superusuari per crear o editar el fitxer de servei systemd anomenat `plomesdemar.service`. Aquest fitxer defineix com i quan cal iniciar el servei.

Dins del fitxer `plomesdemar.service`, afegeix el següent contingut per configurar el servei:

```
[Unit]
Description=Levantar Docker Compose
After=network.target
```

```
[Service]
Type=simple
WorkingDirectory=/var/www/html/2024_plomes_al_mar/site
ExecStart=/usr/local/bin/docker-compose -f
/var/www/html/2024_plomes_al_mar/site/docker-compose.yml up -d
ExecStop=/usr/local/bin/docker-compose -f
/var/www/html/2024_plomes_al_mar/site/docker-compose.yml down
Restart=always
```

```
[Install]
WantedBy=multi-user.target
```

## Gestió de serveis amb systemd

- `sudo systemctl daemon-reload`: Recarregar la configuració de systemd.
- `sudo systemctl enable plomesdemar.service`: Habilitar el servei `plomesdemar`.
- `sudo systemctl start plomesdemar.service`: Iniciar el servei `plomesdemar`.
- `sudo systemctl restart apache2`: Reiniciar `Apache2`

# Configuració VirtualHost

- `sudo nano /etc/apache2/sites-available/plomesdemar.conf`: Crear o editar el fitxer de configuració de la api. Obre l'editor nano amb privilegis de superusuari per crear o editar el fitxer de configuració de la api.conf.

El seu contingut a de ser:

```
<VirtualHost *:80>
    ServerName (ip elastica)
    DocumentRoot /var/www/html/2024_plomes_al_mar/site/public
    <Directory /var/www/html/2024_plomes_al_mar/site/public>
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>
```

## Api Imatges

Preparació dins de la Api d'Imatges (Api\_Imatges)

- ``cd ..`: Canvia el directori de treball actual anterior.
- `cd API_image/`: Canviar el directori de treball actual al de Api\_image.
- `sudo update-alternatives --set php /usr/bin/php8.2`: Canvia la versió predeterminada de PHP que s'utilitza al sistema a PHP 8.2.
- `sudo composer install`: Instal·la totes les dependències del projecte.
- `sudo cp .env.example .env`: Copia el fitxer de configuració d'exemple .env.example a un fitxer .env nou, que és on s'emmagatzemen les configuracions específiques de l'entorn.
- `sudo php artisan key:generate`: Genera una clau d'aplicació nova per al projecte Laravel. Aquesta clau es fa servir per assegurar les sessions i altres dades xifrades.
- `sudo php artisan storage:link`: Crea un enllaç simbòlic des de public/storage a storage/app/public. Això és necessari perquè els fitxers emmagatzemats al directori d'emmagatzematge siguin accessibles públicament.

## Configuració .env

Configurem l'arxiu .env per connectar a la nostra bd:

- `sudo nano .env`: Obre l'editor de text nano amb privilegis de superusuari per editar el fitxer .env, que conté les configuracions d'entorn.

Cambiar aquestes lines del fitxer:

APP\_URL=http://localhost



DB\_CONNECTION

DB\_HOST

DB\_PORT

DB\_DATABASE

DB\_USERNAME

DB\_PASSWORD

A aquestes:

APP\_URL=http://ip privada

DB\_CONNECTION=mysql

DB\_HOST= privada: Introduir IP Privada

DB\_PORT=3307

DB\_DATABASE=apiimagenes

DB\_USERNAME=mysql

DB\_PASSWORD=1234

## Gestió del servidor web Apache

- `sudo systemctl restart apache2`: Reiniciar el servei Apache.
- `sudo nano /etc/apache2/sites-available/api.conf`: Crear o editar el fitxer de configuració de la api. Obre l'editor nano amb privilegis de superusuari per crear o editar el fitxer de configuració de la api api.conf.

El seu contingut a de ser:

```
<VirtualHost *:8080>
    ServerName (ip elastica)
    DocumentRoot /var/www/html/2024_plomes_al_mar/API_image/public
    <Directory /var/www/html/2024_plomes_al_mar/API_image/public>
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>
```

# Preparació ports

- `sudo nano /etc/apache2/ports.conf`: Afegir port 8080 per api. Afegir contingut a sota del Listen 80:

```
Listen 8080
```

- `sudo a2dismod php8.1`: Desactivar el mòdul PHP 8.1 a Apache.
- `sudo a2enmod php8.2`: Activar el mòdul PHP 8.2 a Apache.
- `sudo a2ensite plomesdemar.conf`: Habilitar el lloc web plomesalmar.
- `sudo a2ensite api.conf`: Habilitar la api d'imatges Api\_image.

## Configuració de permisos i directoris

- `sudo mkdir -p /var/www/html/2024_plomes_al_mar/site/storage/logs`: Creeu el directori logs dins de storage si no existeix.
- `sudo chown -R www-data:www-data /var/www/html/2024_plomes_al_mar/site/public`: Canvia el propietari del directori públic i tots els seus subdirectoris i fitxers a l'usuari i grup www-data.
- `sudo chown -R www-data:www-data /var/www/html/2024_plomes_al_mar/site/storage`: Canvia el propietari del directori storage i tots els seus subdirectoris i fitxers a l'usuari i grup www-data.
- `sudo chown -R www-data:www-data /var/www/html/2024_plomes_al_mar/site/bootstrap/cache`: Canvia el propietari del directori bootstrap/cache i tots els seus subdirectoris i fitxers a l'usuari i grup www-data.
- `sudo chmod -R 775 /var/www/html/2024_plomes_al_mar/site/storage`: Canvia els permisos del directori storage i tots els seus subdirectoris i fitxers a 775 (lectura, escriptura i execució per al propietari i el grup, i només lectura i execució per a altres).
- `sudo chmod -R 775 /var/www/html/2024_plomes_al_mar/site/bootstrap/cache`: Canvia els permisos del directori bootstrap/cache i tots els seus subdirectoris i fitxers a 775.
- `sudo mkdir -p /var/www/html/2024_plomes_al_mar/API_image/storage/logs`: Creeu el directori logs dins de storage si no existeix.
- `sudo chown -R www-data:www-data /var/www/html/2024_plomes_al_mar/API_image/public`: Canvia el propietari del directori públic i tots els seus subdirectoris i fitxers a l'usuari i grup www-data.
- `sudo chown -R www-data:www-data /var/www/html/2024_plomes_al_mar/API_image/storage`: Canvia el propietari del directori storage i tots els seus subdirectoris i fitxers a l'usuari i grup www-data.
- `sudo chown -R www-data:www-data /var/www/html/2024_plomes_al_mar/API_image/bootstrap/cache`: Canvia el propietari del directori bootstrap/cache i tots els seus subdirectoris i fitxers a l'usuari i grup www-data.
- `sudo chmod -R 775 /var/www/html/2024_plomes_al_mar/API_image/storage`: Canvia els permisos del directori storage i tots els seus subdirectoris i fitxers a 775 (lectura, escriptura i execució per al propietari i el grup, i només lectura i execució per a altres).

- `sudo chmod -R 775 /var/www/html/2024_plomes_al_mar/API_image/bootstrap/cache`: Cambia los permisos del directorio bootstrap/cache y todos sus subdirectorios y archivos a 775.

## Crear BD

- `cd /var/www/html/2024_plomes_al_mar/API_image/`
- `php artisan migrate`

Una vegada acabi fer: `* cd .. * cd site * php artisan migrate:refresh --seed`

Cuan acabi cambiar le ip privades de APP\_URL i API\_URL per ip publiques.

## Activar mòdul de reescriptura a Apache

- `sudo a2enmod rewrite`: Habilita el mòdul rewrite a Apache, necessari per a les regles de reescriptura d'URLs.

## Reiniciar el servei Apache

- `sudo systemctl restart apache2`: Reiniciar el servidor web Apache, cosa que implica aturar el servei actual i després iniciar-lo novament. Reiniciar Apache és necessari per aplicar qualsevol canvi a la configuració del servidor o els mòduls.

## Detalls

- Si al fer tots el pasos apareix un error de que falten permisos a alguna carpeta fer la part de permisos de la carpeta que falta, despres `sudo a2enmod rewrite` i `sudo systemctl restart apache2`.
- Cuan s'apaga el servidor y es torna a obrir s'ha d'accedir a la carpeta `cd /var/www/html/2024_plomes_al_mar/site` i `sudo docker compose up -d`.