Intrucciones montar CI/CD con Docker en VPS usando GitHub Actions para reactJs vite

# Documentos necesarios para dockerizar

En el directorio raíz crear:

## docker-compose.yml:

version: '3'

services:

  react-app:

    image: tuusuario/react-app:latest

    container\_name: react-app

    restart: always

    ports:

      - "80:80"

## Docker-file

# Etapa 1: Build

FROM node:20-alpine as build

WORKDIR /app

COPY . .

RUN npm install

RUN npm run build

# Etapa 2: Producción con nginx

FROM nginx:stable-alpine

COPY --from=build /app/dist /usr/share/nginx/html

COPY nginx.conf /etc/nginx/conf.d/default.conf

## nginx.conf

server {

  listen 80;

  server\_name localhost;

  root /usr/share/nginx/html;

  index index.html;

  location / {

    try\_files $uri /index.html;

  }

}

## /.github\workflows\deploy.yml

name: Build & Deploy React App

on:

  push:

    branches:

      - main

env:

  IMAGE\_NAME: tuusuario/react-app

  CONTAINER\_NAME: react-app

jobs:

  build-and-deploy:

    runs-on: ubuntu-latest

    steps:

    - name: Checkout del código

      uses: actions/checkout@v3

    - name: Log in en Docker Hub

      uses: docker/login-action@v3

      with:

        username: ${{ secrets.DOCKER\_USERNAME }}

        password: ${{ secrets.DOCKER\_PASSWORD }}

    - name: Build de la imagen Docker

      run: |

        docker build -t $IMAGE\_NAME:latest .

    - name: Push de la imagen al Docker Hub

      run: |

        docker push $IMAGE\_NAME:latest

    - name: SSH al VPS y actualizar

      uses: appleboy/ssh-action@v0.1.10

      with:

        host: ${{ secrets.VPS\_HOST }}

        username: ${{ secrets.VPS\_USER }}

        key: ${{ secrets.VPS\_SSH\_KEY }}

        script: |

          docker pull $IMAGE\_NAME:latest

          docker-compose -f ~/docker-compose.yml down

          docker-compose -f ~/docker-compose.yml up -d

# crear repositorio

git init

git add .

git commit -m "Primer commit"

git branch -M main

git remote add origin https://github.com/TU\_USUARIO/NOMBRE\_REPOSITORIO.git

git push -u origin main

## Agregamos Secrets

En tu repositorio > **Settings → Secrets → Actions**, añade:

| **Nombre** | **Valor** |
| --- | --- |
| DOCKER\_USERNAME | Tu usuario de Docker Hub |
| DOCKER\_PASSWORD | Tu contraseña o token de acceso |
| VPS\_HOST | IP del VPS |
| VPS\_USER | Usuario del VPS (ej: ubuntu) |
| VPS\_SSH\_KEY | Tu clave privada SSH (id\_rsa sin pass) |
| Generamos la key publica y privada SSH ssh-keygen -t rsa -b 4096 -C "github-ci"  queda almacenada en :  C:\Users\tu\_usuario\.ssh\github\_ci\_key  Verificar la llave con este comando:  type C:\Users\tu\_usuario\.ssh\github\_ci\_key.pub  una vez verificada copiarla en el servidor VPS y en git secrects |  |

## Crear directorios y archivo de despiegue en VPS

Navegamos al /home del usuario

Como la composición del VPS es tipo hosting Ya que tienes carpetas como htdocs y logs, parece que tu VPS está usando una estructura tipo **hosting (como Apache o Nginx por defecto)**.

Pero como estás usando **Docker + Nginx** para servir la app React (según nuestro plan), lo ideal es crear una nueva carpeta en tu home para ese propósito.

Procedemos a crear los directorios:

mkdir -p ~/react-cicd

cd ~/react-cicd

creamos el archive docker-compose.yml con el comando

nano docker-compose.yml

y pegamos lo siguiente:

version: '3'

services:

react-app:

image: nginx:stable-alpine

container\_name: react-app

restart: always

ports:

- "80:80"

volumes:

- ./app:/usr/share/nginx/html

- ./nginx.conf:/etc/nginx/conf.d/default.conf

Hacemos lo mismo con el nginx.conf

Nano nginx.conf

server {

listen 80;

server\_name localhost;

root /usr/share/nginx/html;

index index.html;

location / {

try\_files $uri /index.html;

}

}

## Creamos el dir de la APP

mkdir app

## Actualizamos el workflow

En el paso de **SCP** (dist/\*) y **SSH**, actualiza los paths:

**En .github/workflows/deploy.yml:**

yaml

CopiarEditar

# Enviar la carpeta build a la carpeta app en el VPS

name: Deploy to VPS

on:

  push:

    branches:

      - main

jobs:

  deploy:

    runs-on: ubuntu-latest

    steps:

      # Enviar la carpeta build a la carpeta app en el VPS

      - name: Subir build a VPS

        uses: appleboy/scp-action@v0.1.3

        with:

          host: ${{ secrets.VPS\_HOST }}

          username: ${{ secrets.VPS\_USER }}

          key: ${{ secrets.VPS\_SSH\_KEY }}

          port: 22

          source: "dist/\*"

          target: "/home/${{ secrets.VPS\_USER }}/react-cicd/app"

      # Reiniciar nginx desde docker-compose

      - name: Reiniciar contenedor nginx

        uses: appleboy/ssh-action@v0.1.10

        with:

          host: ${{ secrets.VPS\_HOST }}

          username: ${{ secrets.VPS\_USER }}

          key: ${{ secrets.VPS\_SSH\_KEY }}

          script: |

            cd ~/react-cicd

            docker compose down

            docker compose up -d

## dar permisos de docker al usuario

sudo usermod -aG docker lmcadev-master

## subir el deploy

git add .

git commit -m "Reintento despliegue"

git push origin main

## configuración de nginx para servir con react y docker

/etc/nginx/sites-available

server {

listen 443 ssl http2 default\_server;

listen [::]:443 ssl http2 default\_server;

server\_name lmcadev.com www.lmcadev.com;

ssl\_certificate /etc/nginx/ssl-certificates/lmcadev.com.crt;

ssl\_certificate\_key /etc/nginx/ssl-certificates/lmcadev.com.key;

root /home/lmcadev-master/react-cicd/app;

index index.html;

location / {

try\_files $uri $uri/ /index.html;

}

location ~\* \.(css|js|jpg|jpeg|gif|png|ico|svg|ttf|woff|woff2|eot|otf|mp4|webm|zip|gz|webp)$ {

expires max;

access\_log off;

add\_header Cache-Control "public";

}

error\_page 500 502 503 504 /50x.html;

}