

Introducción



En esta sección agregare una lista de docker compose de aplicaciones de interés común que previamente haya probado.

Todos los stacks tendrán su archivo docker-compose.yaml y su archivo .env los cuales deben estar dentro de una carpeta con el nombre del stack.

Además, adjunto una lista de comandos de Docker para hacer todas las actividades cotidianas del mantenimiento de servicios de Docker, en caso de que sea Tapatío puedes instalar Portainer para hacer toda esta gestión mediante una Interfaz Web.

Recomendaciones

Para fines de portabilidad te recomiendo crear una carpeta donde almacenaras toda la data de los servicios de docker, de modo que si necesitas exportar todos tus servicios solo necesitaras copiar esta carpeta al nuevo servidor, ejecutar los contenedores de nuevo y estos preservaran el estado anterior.

Carpeta que recomiendo crear

Carpeta relacionada a toda la data de las apps de docker y su estado

Carpeta: /dockerData/appdata

Nombre en los enviroments : PATH_TO_APPDATA

Carpeta relacionada a toda la data de usuarios de los distintos servicios de Docker

Carpeta: /dockerData/data

Nombre en los enviroments: PATH_TO_DATA

Carpeta dedicada a almacenar todos los stacks de los distintos servicios de Docker

Carpeta: /dockerData/stack

Detalles de la infrestuctura



La documentación esta planeada para el uso de Nginx Proxy Manager para

exponer los servicios por lo que es importante que primero hagas la instancia de Nginx Proxy Manager o su red correspondiente para que todos los demás stacks se monten sobre esta red como externos, Todos los servicios en estos docker compose que sean compartidos mediante HTTP tendrán la configuración de red externo con excepción de Nginx Proxy Manager el cual lo tendrá como red interna.

Comandos Esenciales de Docker

Listar contenedores

```
# Contenedores activos
docker ps

# Todos los contenedores (activos e inactivos)
docker ps -a

# Contenedores con formato personalizado
docker ps --format "table {{.Names}}\t{{.Status}}\t{{.Ports}}"
```

Gestionar stacks/compose

```
# Levantar stack desde archivo compose
docker compose -f /ruta/al/docker-compose.yml up -d

# Detener stack sin eliminar contenedores
docker compose -f /ruta/al/docker-compose.yml stop
```

```
# Detener y eliminar stack
docker compose -f /ruta/al/docker-compose.yml down

# Reiniciar stack
docker compose -f /ruta/al/docker-compose.yml restart

# Ver logs del stack
docker compose -f /ruta/al/docker-compose.yml logs -f

# Tambien puedes posicionarte sobre la carpeta para ahorrarte
# escribir -f y la ruta al stack. ejemplo a continuacion

# Ingresa a la ruta del stack
cd /ruta/a/stacks/filebrowser

# Verificar que el archivo compose es correcto y los enviroments se cargan
docker compose config

# Actualizar
docker compose pull
docker compose up -d

# Forzar recreacion de los contenedores del stack
docker compose up -d --force-recreate
```

Gestionar contenedores individuales

```
# Detener contenedor
docker stop nombre_contenedor

# Iniciar contenedor
docker start nombre_contenedor

# Reiniciar contenedor
docker restart nombre_contenedor

# Eliminar contenedor
docker rm nombre_contenedor

# Ver logs en tiempo real
docker logs -f nombre_contenedor

# Ejecutar comando dentro del contenedor
```

```
docker exec -it nombre_contenedor bash
```

Gestionar imágenes

```
# Listar imágenes
docker images

# Listar imágenes no utilizadas (dangling)
docker images -f "dangling=true"

# Eliminar imágenes no utilizadas
docker image prune

# Eliminar todas las imágenes no utilizadas (incluyendo las no referenciadas)
docker image prune -a

# Actualizar imágenes específicas
docker pull nombre_imagen:tag

# Ver el tamaño de las imágenes
docker system df
```

Gestionar redes

```
# Listar redes
docker network ls

# Inspeccionar red específica
docker network inspect nombre_red

# Listar redes no utilizadas
docker network ls --filter dangling=true

# Eliminar redes no utilizadas
docker network prune

# Crear red personalizada
docker network create mi_red
```

Gestionar volúmenes

```
# Listar volúmenes
docker volume ls

# Inspeccionar volumen
docker volume inspect nombre_volumen

# Listar volúmenes no utilizados
```

```
docker volume ls -f dangling=true

# Eliminar volúmenes no utilizados
docker volume prune

# Crear volumen
docker volume create nombre_volumen
```

Comandos de monitoreo

```
# Estadísticas de contenedores en tiempo real
docker stats

# Uso de disco del sistema Docker
docker system df

# Eventos de Docker en tiempo real
docker events

# Información del sistema
docker info
```

Scripts

Para la creación de scripts te recomendamos revisar la sección de [Crons y Bash Files](#)

Limpieza completa de Docker

Este script es para limpiar Docker por completo

cleanup-docker.sh

```
#!/bin/bash
echo "=== Limpieza de Docker ==="
echo "1. Deteniendo contenedores..."
docker stop $(docker ps -q)
echo "2. Eliminando contenedores..."
docker rm $(docker ps -aq)
echo "3. Eliminando imágenes no utilizadas..."
docker image prune -af
echo "4. Eliminando redes no utilizadas..."
docker network prune -f
echo "5. Eliminando volúmenes no utilizados..."
```

```
docker volume prune -f
echo "6. Limpieza del sistema..."
docker system prune -af
echo "□ Limpieza completada"
```

Fin



Revisión #16

Creado 2025-11-25 15:50:25 UTC por Bruno

Actualizado 2025-12-03 01:36:21 UTC por Bruno