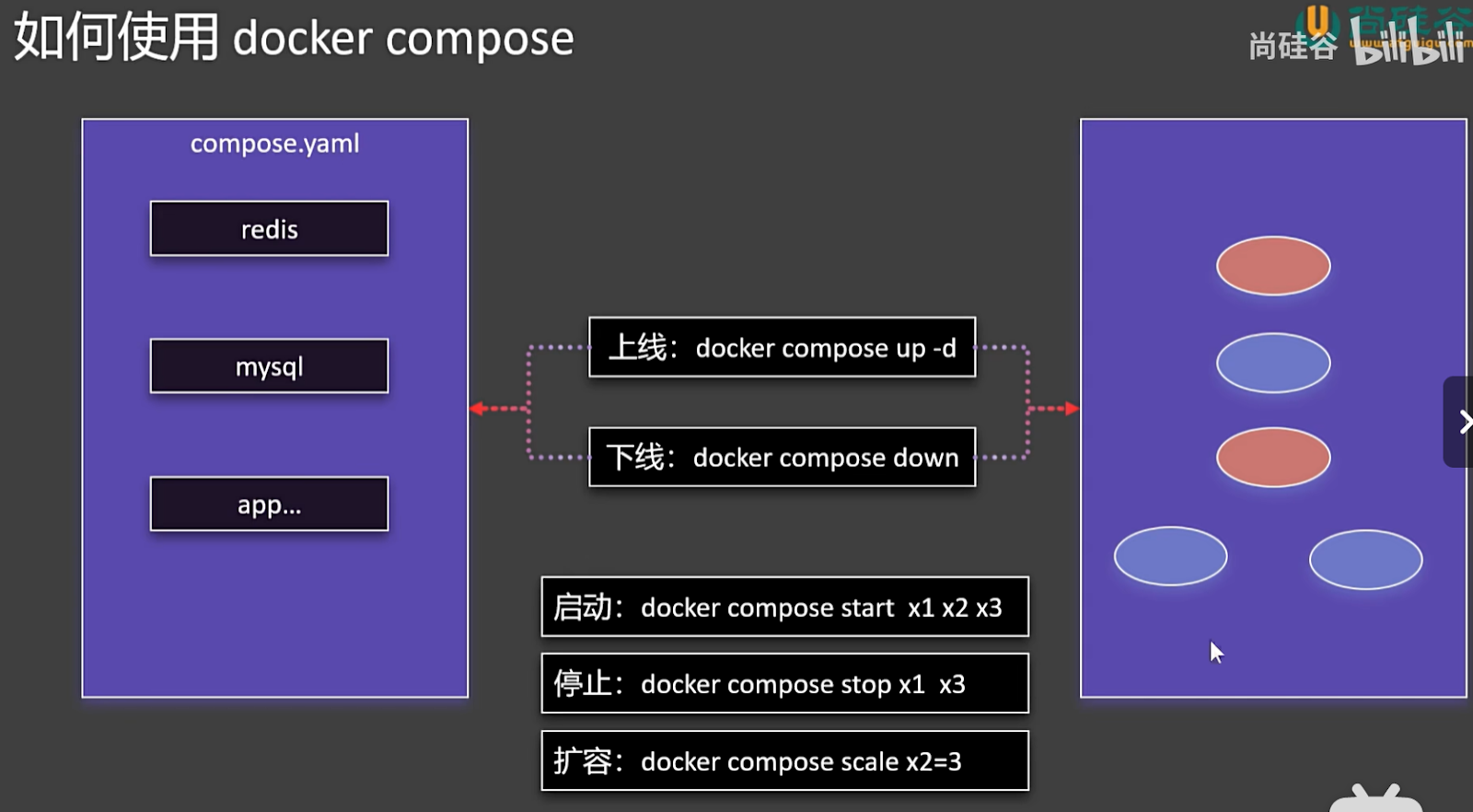
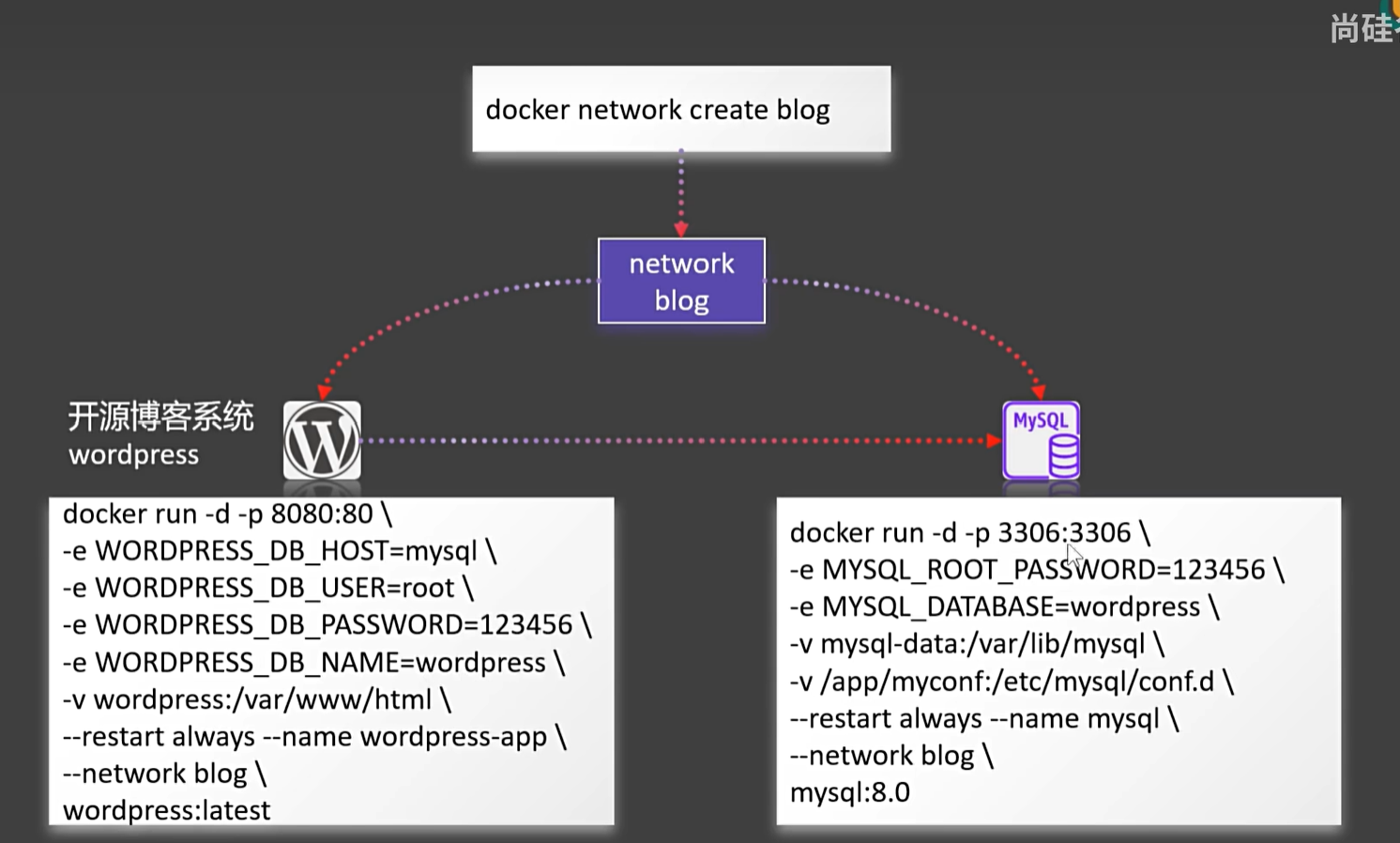
Docker Compose批量管理容器

图37



部署案例：以前的做法，分别开启容器，博客内容会存在MYSQL中

图38



官网可以参考compose文件的写法，Docker Compose 是一个用于定义和运行多容器 Docker 应用的工具。通过使用一个 YAML 文件，你可以配置应用程序中的所有服务、网络和卷，从而简化容器的管理和部署。

docker-compose.yml 文件是 Docker Compose 的配置文件，用于定义应用的服务、网络和卷。以下是一个简单的 docker-compose.yml 文件示例，它定义了一个包含 MySQL 和 WordPress 服务的应用：

version: '3.8'  # 指定 Docker Compose 文件的版本

services:  # 定义服务

  db:  # MySQL 服务

    image: mysql:5.7  # 使用 MySQL 5.7 镜像

    volumes:

      - db\_data:/var/lib/mysql  # 挂载卷

    environment:  # 设置环境变量

      MYSQL\_ROOT\_PASSWORD: example

      MYSQL\_DATABASE: wordpress

      MYSQL\_USER: wordpress

      MYSQL\_PASSWORD: wordpress

  wordpress:  # WordPress 服务

    image: wordpress:latest  # 使用最新的 WordPress 镜像

    ports:

      - "8000:80"  # 映射主机端口 8000 到容器端口 80

    environment:  # 设置环境变量

      WORDPRESS\_DB\_HOST: db:3306

      WORDPRESS\_DB\_USER: wordpress

      WORDPRESS\_DB\_PASSWORD: wordpress

      WORDPRESS\_DB\_NAME: wordpress

    depends\_on:

      - db  # 表示 WordPress 服务依赖于 MySQL 服务

volumes:  # 定义卷

  db\_data:  # 数据卷，用于持久化 MySQL 数据

version: 指定 Docker Compose 文件的版本。

services: 定义应用程序的各个服务。每个服务都有自己的配置。

db: 定义 MySQL 服务。

image: 指定要使用的 Docker 镜像（例如 mysql:5.7）。

volumes: 定义卷，将数据持久化到主机。

environment: 设置环境变量，例如数据库密码和名称。

wordpress: 定义 WordPress 服务。

image: 指定要使用的 Docker 镜像（例如 wordpress:latest）。

ports: 映射主机端口到容器端口。

environment: 设置环境变量，例如数据库连接信息。

depends\_on: 指定服务的依赖关系，在启动 wordpress 服务前，确保 db 服务已启动。

volumes: 定义卷，用于持久化数据。

**使用 Docker Compose**

创建 docker-compose.yml 文件: 在项目目录下创建一个名为 docker-compose.yml 的文件，并将上述内容复制到文件中。

启动服务:在项目目录下运行以下命令以启动所有定义的服务：docker-compose up

后台运行服务: 如果希望在后台运行服务，可以使用 -d 参数：

docker-compose up -d

停止服务:要停止服务并删除容器，可以使用以下命令：

docker-compose down

查看日志:可以使用以下命令查看所有服务的日志：

docker-compose logs

通过 Docker Compose，你可以轻松地定义和管理复杂的多容器应用，确保它们在任何环境中都可以一致地运行。

图39

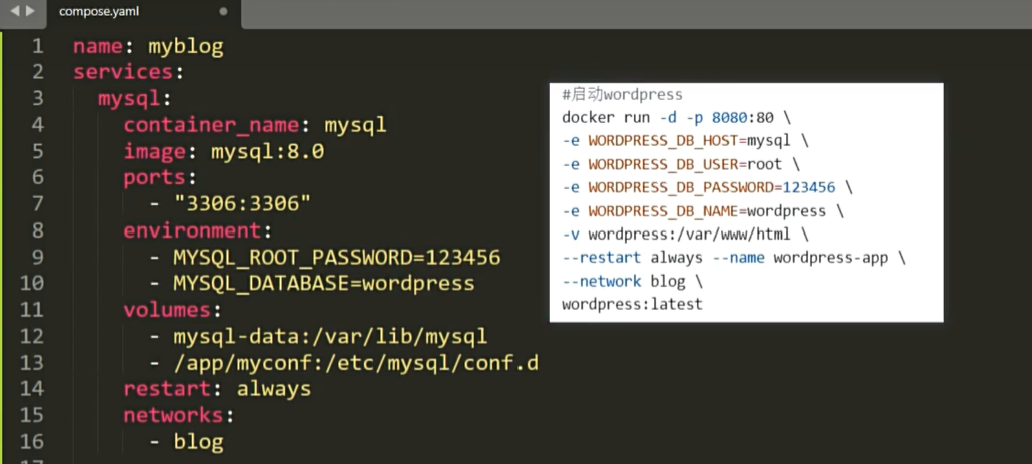
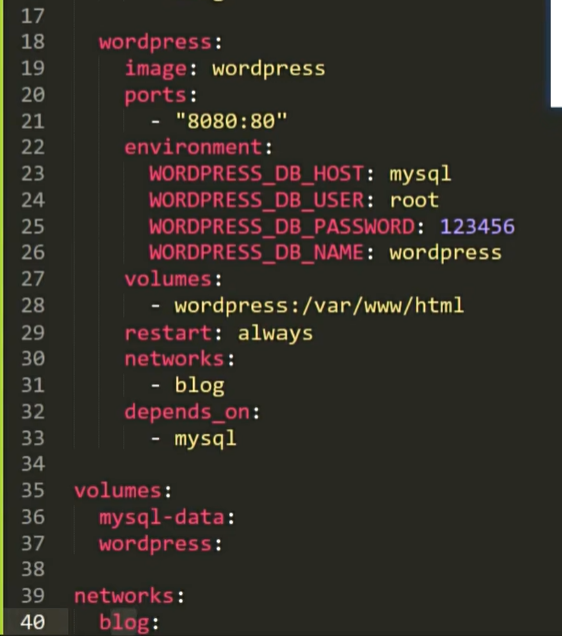
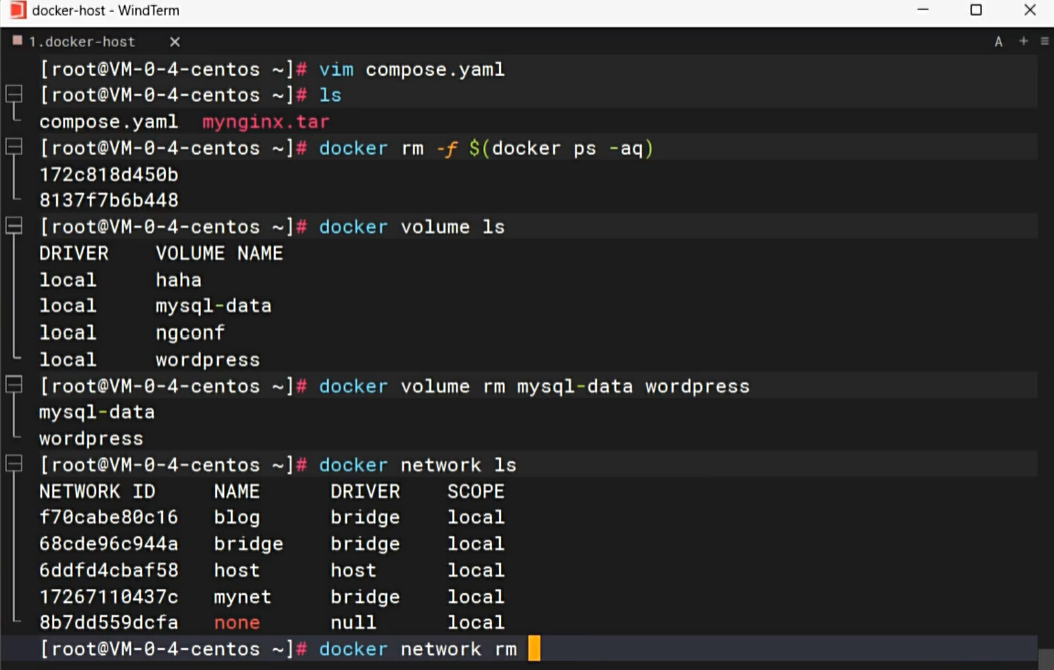


图40



将compose文件加入并删除现在所有的镜像

图41



启动compose文件，名称是应用名\_网络名

图42

