Docker和微服务

汇报：第9小组

成员：李铭杨、蒋相云、沈佳雯、孟嫄

1 /微服务的定义/

2 /Docker的定义/

3 /部署Docker服务/

4 /部署自己的程序/

Part 01

什么是微服务

概念、应用场景、部署方法

Concept, Application Scenarios, Deployment Methods

/微服务简介/

概念定义

一种将应用程序构建为一系列小型服务的开发方法，每个服务都运行在自己的进程中，并通过轻量级通信机制进行通信。

主要功能

- 服务解耦：将大型应用拆分为多个小型、独立的服务。
- 独立部署：每个服务可以独立开发、测试和部署。
- 技术多样性：允许使用不同的技术栈来开发各个服务。

发展历史

源于互联网公司，如Netflix和Amazon等，它们在处理大规模应用时遇到了传统单体架构的限制，于是逐渐采用微服务架构。

优点总结

提高了系统的灵活性、可伸缩性和可维护性，也可提升系统的安全性和可扩展性

/微服务应用场景/

电商平台

- 如亚马逊、淘宝等大型电商平台，将用户管理、订单管理、支付管理等不同功能拆分成独立的微服务，以应对高并发和快速迭代的需求。

社交媒体

- 将用户信息、消息推送、社交关系等功能拆分成微服务，以实现灵活的扩展和个性化的服务。

数据分析

- 在大数据处理领域，微服务能够帮助企业实时分析海量数据，提供决策支持。

/微服务部署方法/

01

Docker

- 容器化技术，可以实现应用的快速部署。
- 通过Dockerfile定义应用环境和依赖，创建轻量级的、可移植的容器。
- 适用于快速迭代开发和测试环境。

02

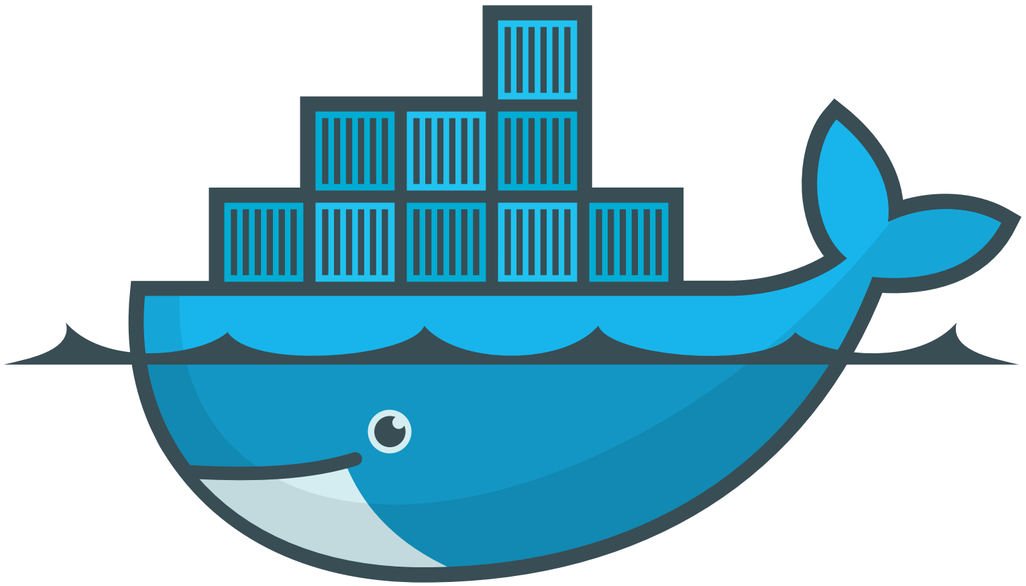
Vercel

- 现代化的云平台，提供无服务器计算和静态网站托管服务。
- 允许开发者快速部署和扩展Web应用。
- 适用于前端应用的快速部署和静态资源托管。

03

虚拟机

- 虚拟机技术可以在物理服务器上模拟出多个独立的计算机系统。
- 每个虚拟机可以运行不同的操作系统和应用。
- 适用于需要隔离性和稳定性的生产环境部署



Part 02

什么是Docker

简介、为什么用、优缺点

Introduction, Why Use It, Pros and Cons

/Docker简介/

发展历史

由DotCloud公司开发的开源项目，于2013年推出。它的发展源于Linux容器技术的进步

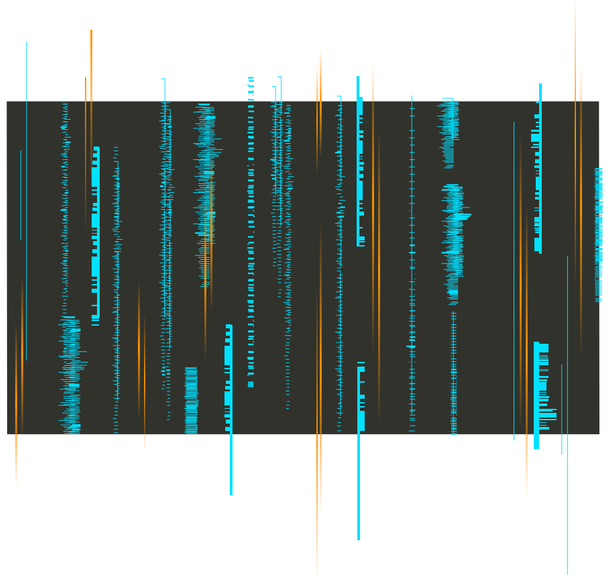
概念定义

Docker是一种容器化平台，提供了轻量级的虚拟化解决方案，可以将应用程序及其所有依赖项打包到一个可移植的容器中，并在任何环境中运行。

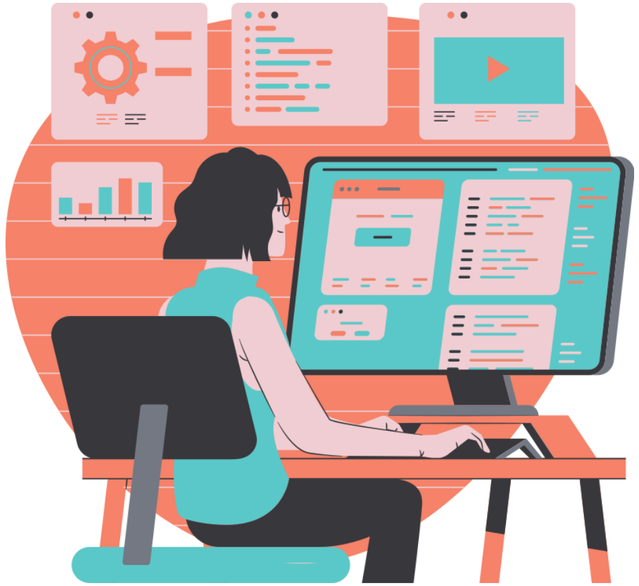
名词介绍

- 镜像： Docker镜像是一个只读的模板，包含了运行容器所需的文件系统、运行时环境等。镜像可以用来创建容器的实例。
- 容器： Docker容器是从Docker镜像创建的运行实例，它包含了应用程序及其所有运行时依赖项，可以被快速部署和运行。
- 仓库： Docker仓库是用来存储和分享Docker镜像的地方，可以是公共的或私有的，例如Docker Hub是一个公共的镜像仓库。

/为什么用Docker/



想在不同平台部署同一套代码?



不想配环境？不想配依赖项?

示例程序

/为什么用Docker/

Home Assistant	http://1419cuc.fun:8023
博客	http://1419cuc.fun:8090

Docker技术面面观

优劣分析

优点

- **高效轻量：** 启动快速，性能开销小，资源利用率高。
- **环境一致性：** 确保应用在不同环境中的一致性，解决了“在我的机器上就可以运行”的问题。
- **易于扩展：** 可以通过简单的命令或工具快速扩展应用的实例数量。
- **隔离性：** 每个容器内运行着独立的应用，互不干扰。容器只占用所需资源，不会占满整个系统的内存，提高了资源的利用率。

缺点

- **安全性问题：** Docker容器之间共享主机的内核，存在一定的安全隐患，需要进行适当的安全配置和管理。
- **学习曲线：** 对于初学者来说，Docker的概念和使用方法可能有一定的学习难度。
- **性能损耗：** 相比裸机运行应用程序会有一定的性能损耗，尤其是在I/O密集型应用场景下。
- **持久化存储问题：** Docker容器本身是临时的，不适合存储应用程序的持久化数据，

Part 03

部署Docker服务

MacOS/Windows/Linux

安装方式/基础命令

/Docker在MacOS下的安装/

1. 下载 Docker Desktop for Mac
2. 安装 Docker Desktop
3. 拖拽安装

/Docker在Windows下的安装/

1

前置说明

- 确保你的 Windows 系统满足 Docker Desktop 的系统要求。对于最新版本的 Docker Desktop，通常需要 Windows 10 64-bit: Pro, Enterprise 或 Education (Build 15063 或更高版本)。你也可以在 Windows Home 上安装它，但可能需要额外的步骤来启用 WSL 2。
- 对于 Windows 10 Home，可能需要手动启用 WSL 2 和虚拟化功能。

2

下载 Docker Desktop for Windows

- Docker 官方网站下载 Docker Desktop for Windows 的安装程序。下载链接

3

安装

- 双击下载的 Docker Desktop 安装程序（Docker Desktop Installer.exe）。
- 在安装过程中，如果系统提示你启用 Hyper-V 功能，请选择是。如果你的系统支持 WSL 2 并且你想使用它，也可以在安装过程中选择使用 WSL 2 作为 Docker 的后端。
- 完成安装后，重启你的系统。

/Docker在Linux下的安装/

Linux环境中docker部署

- docker pull [options] NAME[:TAG]

通过此命令可以docker远程仓库拉取镜像到本地.
name是拉取镜像的名称,:TAG表示是可选的,如果不选表明时latest,如果选择表明是指定版本的.
options是拉去的一些参数.

- docker images [options] [REPOSITORY[:TAG]]

options是选项,后面是指定镜像的名称.这个用的不多,可能当本地镜像非常多的时候要指定查看某一个镜像.

```
[root@iZ2ze68ge5c1uwlkmnb9ixZ zcapp]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
hello-world         latest              fce289e99eb9        4 months ago        1.84kB
```

- docker run [options] IMAGE[:TAG] [COMMAND] [ARG..]

IMAGE是镜像的名字.COMMAND是运行起来的时候要执行什么命令.ARG表示这条命令运行需要的参数.

- docker tag NAME[:TAG] NEWNAME[:NEWTAG]

为本地的镜像添加新的标签.例如

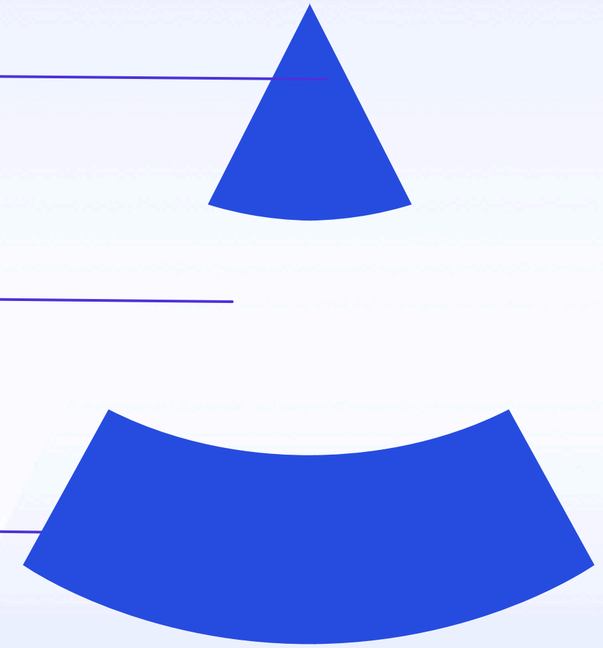
/使用docker命令部署应用程序/

编写Dockerfile文本文件，其中包含了构建Docker镜像所需的指令，在这里，可以定义基础镜像、安装依赖、复制文件等。

构建镜像，使用dockersbuild命令根据Dockerfile构建Docker镜像，镜像中包含了应用程序和依赖。

运行容器，使用docker run命令基于构建的镜像运行Docker容器，可以指定端口映射、环境变量等配置。

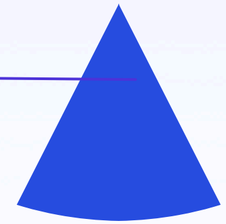
扩展和管理，同意在接著其他编排工具管理多个容器的部署和通信。



/使用docker-compose部署应用程序/

编写docker-compose.yml文件（可以在官网上寻找与自己docker对应的所需版本）

使用docker-compose命令，例如docker-compose up前后台创建镜像并启动容器。其余操作也是调用命令。



Part 4

部署自己的程序

如何编写自己的程序

Docker的商业竞争优势

即使是最复杂的应用程序
也可以容器化。

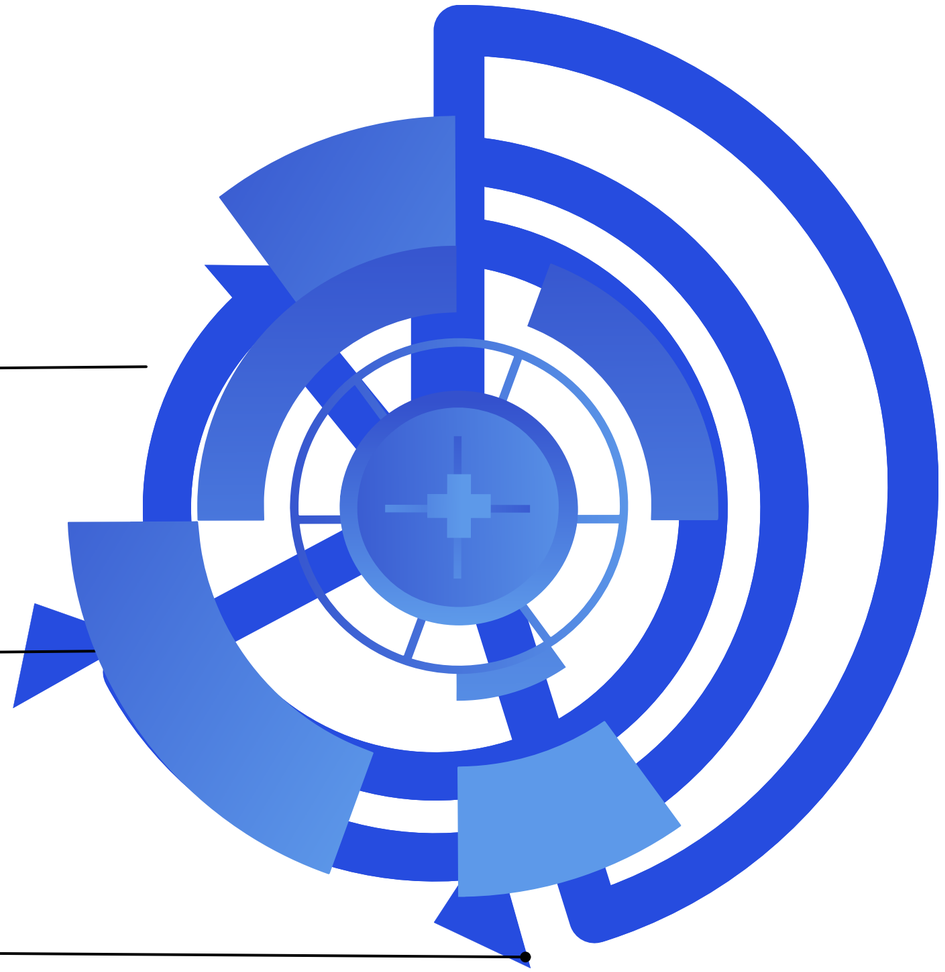
容器利用并共享主机内核,
使它们在系统资源方面
比虚拟机更有效率。

跨数据中心增加容器
自动分发容器副本
方便企业按需自行搭建架构

灵活性

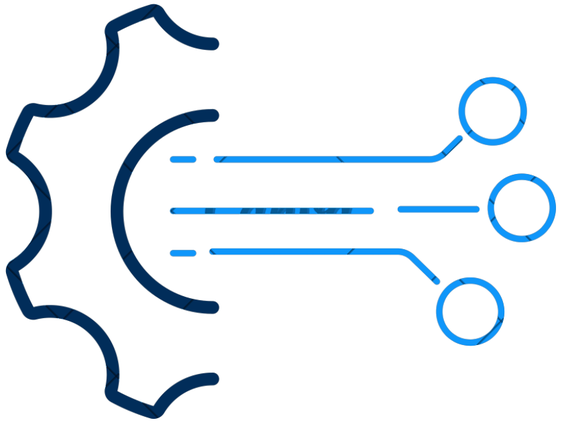
轻量级

可扩展

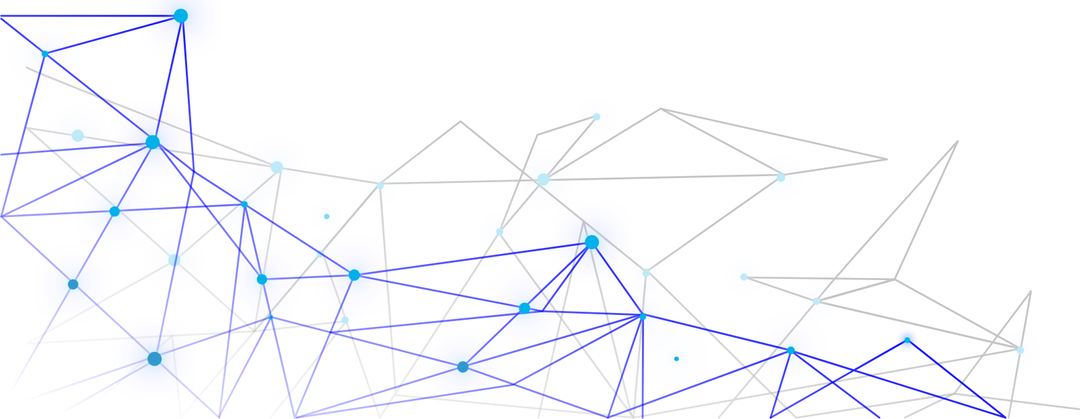


Docker 与微服务的关系

Docker 作为容器工具可以把业务逻辑容器、数据库容器、储存容器、队列容器使得软件可以拆分成若干个标准化容器，然后像搭积木一样组合起来，让彼此通信，从而形成微服务。



这意味着微服务很适合用 Docker 容器实现，每个容器承载一个服务。一台计算机同时运行多个容器，从而就能很轻松地模拟出复杂的微服务架构。



Docker的未来发展

云计算

使用 Docker 容器可以将应用程序及其所有依赖项打包成一个独立的镜像。在云计算环境中，可以使用 Docker 镜像轻松地在不同的云平台或云服务提供商之间部署应用程序，而无需担心环境差异或配置问题。此外，Docker Compose 或 Kubernetes 等工具可以方便地管理多个容器化应用程序的部署和扩展。

边缘计算

边缘设备通常资源有限（如存储、计算能力和内存），Docker 容器启动速度快，占用资源少，能够高效地在边缘设备上运行应用程序。边缘计算涉及各种不同的硬件和操作系统环境。Docker 容器可以提供一致的运行环境，使得应用程序可以在各种边缘设备上运行而无需担心环境差异。这样可以大大简化应用程序的开发和部署过程。

THANK YOU

汇报人：李铭杨

成员：李铭杨、蒋相云、沈佳雯、孟嫄