

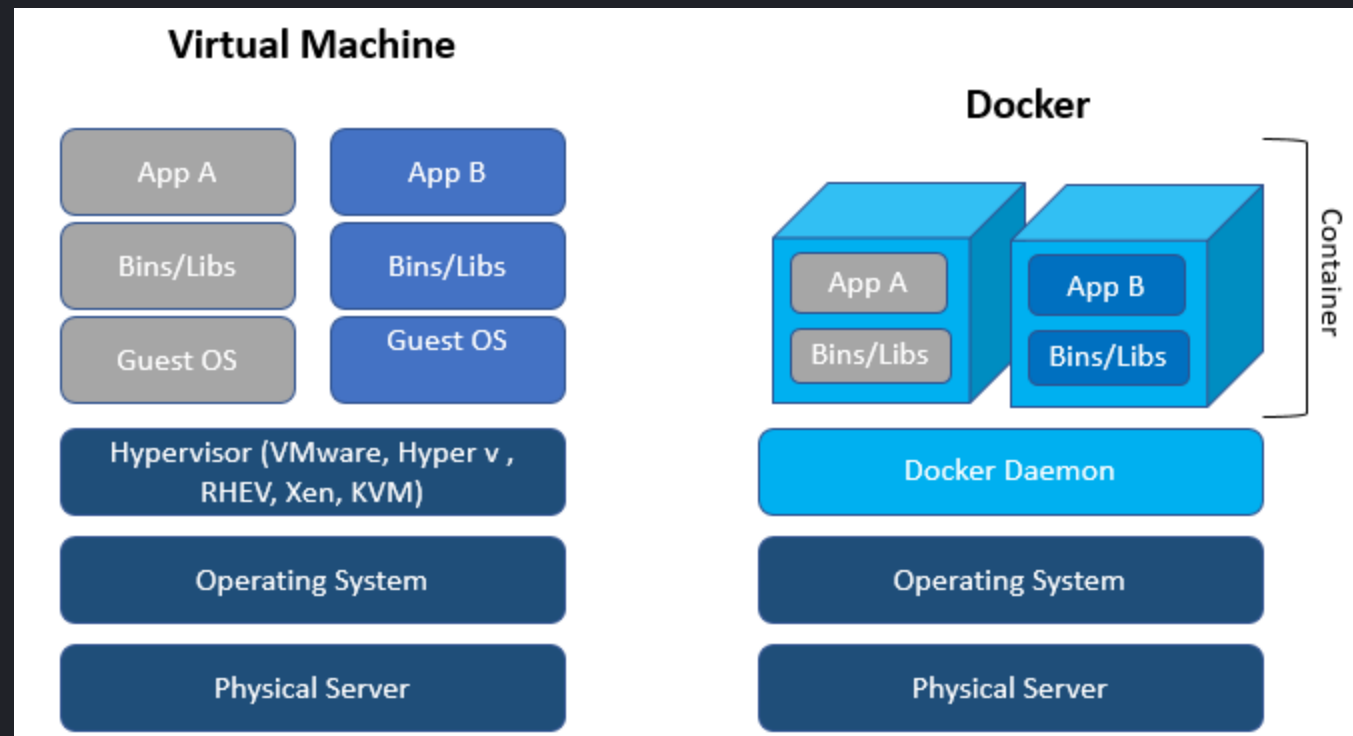


Docker助力敏捷开发

- 什么是docker?
- AI组为什么基于docker开发?
- 一些简单的概念
- 一个简单的应用demo
- Q&A

什么是Docker?

首先 Docker != VM



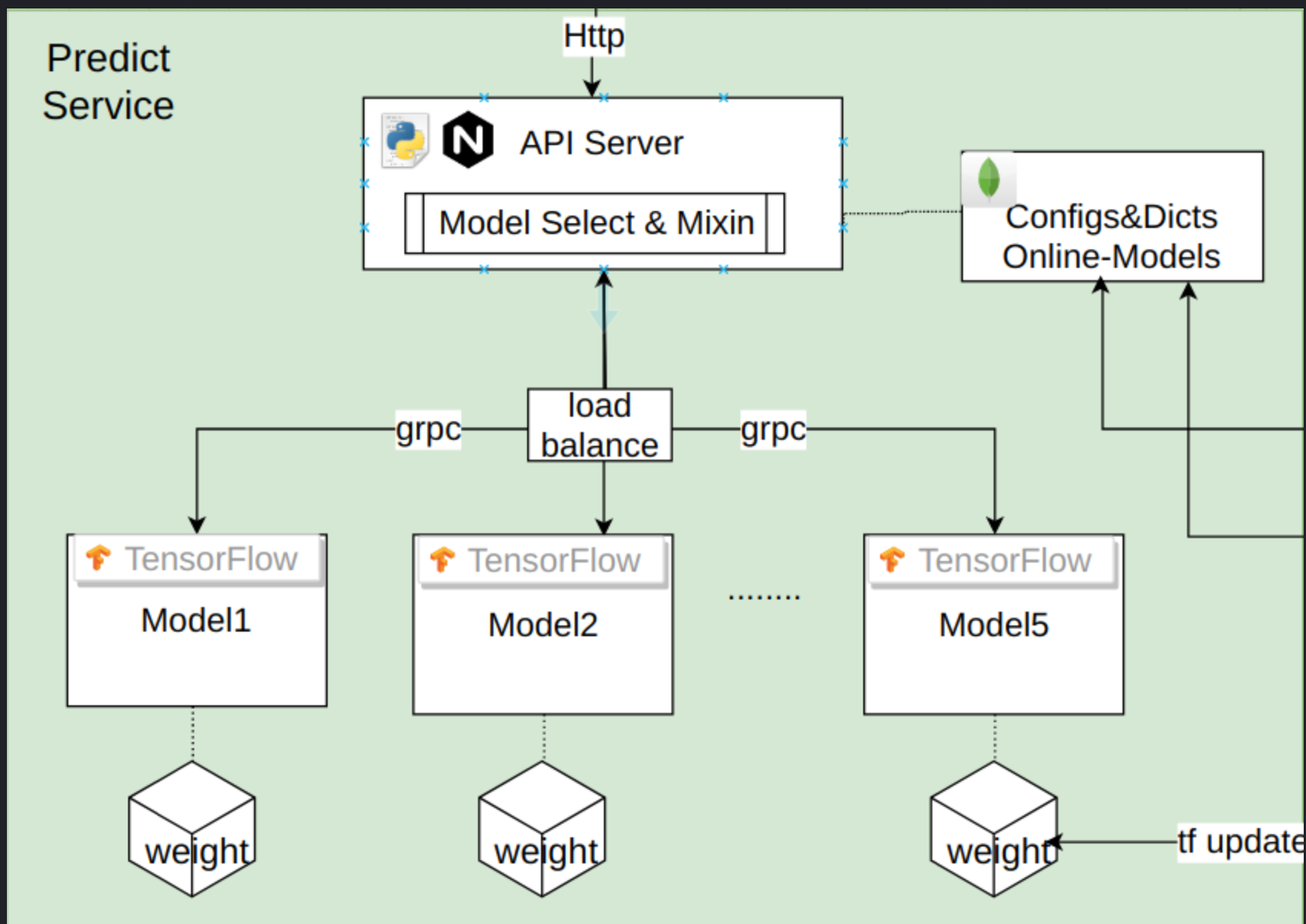
优势

- 基于Linux内核的 `cgroup` , `namespace` 等技术，对进程进行"封装隔离":
 - 启动速度更快
 - 占用系统资源更少
 - 易于迁移

我们为什么基于Docker进行开发？

- 代码一致性
 - Python有序字典的教训
 - 快速统一配置开发环境
- 便于项目部署
 - 混乱的 `Tensorflow api` 与 `cuda`
 - 一键部署一时爽，一直部署一直爽

科目自动匹配项目:



Docker 基本概念

镜像/仓库/容器



镜像

Docker 镜像是一个特殊的文件系统，提供容器运行时所需的程序、库、资源、配置等文件

```
FROM ubuntu:18.04    # 定义基础镜像
RUN apt-get install python3    # 安装系统依赖
COPY app.py /app/app.py    # 拷贝必要文件
CMD ["python3", "/app/app.py"]    # 执行入口
```

仓库

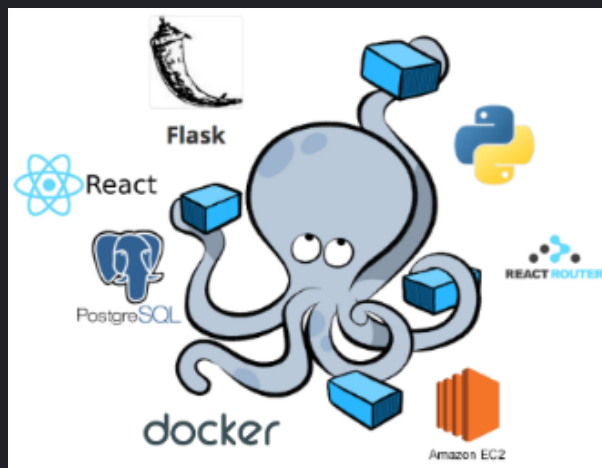
Dockerhub 或私有仓库来保存镜像

```
jupyter/datascience-notebook:latest  
<用户>/<仓库>:标签
```


容器

镜像 和 容器 的关系，就像是 类 和 实例 一样

```
docker run -p 8000:8000 jupyter/datascience-notebook:latest bash  
docker run 端口 <外部端口>:<内部端口> <镜像/仓库地址> <命令>
```



编排文件

- 项目 (project) :
由一组服务组成的完整业务单元，在 `docker-compose.yml` 中定义
- 服务 (service) :
一个应用的容器，实际上可以包括若干运行相同镜像的容器实例



TALK IS CHEAP

SHOW ME THE `</CODE>`

LIVE CODE

- 目标：
使用Vscode创建一个fastapi的开发环境并部署