目录

1 Docker 核心概念总结

Docker 概述：

Docker 是一个开源的容器化平台，用于自动化应用程序的部署、扩展和管理。它通过将应用程序及其所有依赖项打包成一个容器，确保它们在任何环境中都能一致地运行。

容器化：Docker 容器是轻量级的虚拟化技术，允许在一个宿主操作系统上运行多个隔离的应用。

Docker 的组成：

Docker Engine：包括客户端和服务器，负责容器的构建和管理。

Docker 镜像：包含应用运行所需的所有文件、库、依赖等，是容器的可执行包。

Docker 容器：镜像的运行实例，是可以启动、停止、删除和管理的应用环境。

Docker Hub：Docker 的公共镜像仓库，用户可以从中获取共享的镜像。

常用 Docker 命令：

docker run <image>：从指定镜像创建并启动一个新的容器。

docker ps：查看当前运行中的容器。

docker stop <container>：停止正在运行的容器。

docker rm <container>：删除停止的容器。

docker pull <image>：从 Docker Hub 或指定仓库拉取镜像。

docker build -t <image-name> <dockerfile>：使用 Dockerfile 构建镜像。

docker exec -it <container> /bin/bash：进入容器进行交互式操作。

docker logs <container>：查看容器的日志输出。

docker images：查看本地存储的所有镜像。

Docker 容器生命周期：

创建：使用 docker run 或 docker create 命令从镜像创建容器。

启动：容器启动后，可以使用 docker start 命令来启动一个已创建的容器。

停止：使用 docker stop 停止正在运行的容器。

删除：使用 docker rm 删除已停止的容器。

Docker 网络：

Bridge 网络：默认网络模式，用于容器间的通信，容器通过虚拟网桥互联。

Host 网络：容器与主机共享网络栈，直接使用主机的 IP 和端口。

None 网络：不为容器分配网络，容器无法与外界通信。

Docker 的存储：

Volumes：Docker 容器中的数据存储。数据保存在宿主机上，且容器重启时数据不会丢失。

Bind Mounts：将宿主机的目录挂载到容器内，容器的修改会反映到宿主机上。

Tmpfs：将数据存储在内存中，当容器停止时数据会丢失。

Docker 的安全性：用户权限：Docker 容器以 root 用户身份运行，但可以通过 USER 指令指定非 root 用户运行。安全扫描：使用 docker scan 命令扫描镜像中的安全漏洞。隔离：通过容器的命名空间和 cgroups 技术来实现容器之间的隔离。

常用 Kubernetes 命令总结：集群管理命令：

kubectl get nodes：查看集群中的所有节点。

kubectl get pods：查看所有 Pod 的状态。

kubectl get services：查看所有 Service。

kubectl get deployments：查看所有 Deployment。

Pod 管理命令：

kubectl create -f pod.yaml：通过 YAML 文件创建 Pod。

kubectl describe pod <pod-name>：查看指定 Pod 的详细信息。

kubectl delete pod <pod-name>：删除指定 Pod。

kubectl logs <pod-name>：查看 Pod 中的容器日志。

Deployment 管理命令：

kubectl create -f deployment.yaml：创建 Deployment。

kubectl get deployments：查看所有 Deployment。

kubectl set image deployment/<deployment-name> <container-name>=<image>：更新 Deployment 的镜像。

kubectl scale deployment <deployment-name> –replicas=<number>：调整 Deployment 副本数。

kubectl rollout status deployment/<deployment-name>：查看 Deployment 的滚动更新状态。

kubectl rollout undo deployment/<deployment-name>：回滚 Deployment。

Service 管理命令：

kubectl expose pod <pod-name> –port=<port>：为 Pod 暴露 Service。

kubectl expose deployment <deployment-name> –type=LoadBalancer –port=<port 为 Deployment 暴露 LoadBalancer 类型的 Service。

kubectl get svc：查看所有 Service 的状态。

调度命令：

kubectl get pod <pod-name> -o wide：查看 Pod 的详细信息，包括所在节点。

kubectl logs <pod-name> -c <container-name>：查看指定容器的日志。

集群调试命令：

kubectl describe pod <pod-name>：获取 Pod 的详细描述信息。

kubectl get events：查看集群的事件，帮助排查问题。

kubectl exec -it <pod-name> – /bin/bash：进入 Pod 中的容器进行交互。

用 listings 制作类 minted 风格

2025 年 4 月 19 日

不少用户不会配置 minted 环境，我们也制作了视频教程教大家如何配置，可以到这里看看视频配置。https://www.bilibili.com/video/BV1sT4y1A7KR

如果还是不会配置，今天给大家一个用 listings 宏包定制的类 minted 风格，这样就不用配置环境，也可以体验到 minted 样式了。

显示效果如下：

```
var loginLayer=(function(){
    var div=document.createElement("div");
    div.innerHTML="windows1";
    div.style.display="none";
    document.body.appendChild(div);
    return div;
})();
/*document.getElementById('loginBtn').onclick=function(){
    loginLayer.style.display="block";
};*/
alert(loginLayer.style.display);

    public class HelloWorld {
        public static void main(String[] args) {
            System.out.println("Hello, World!");
        }
    }
```

N.53 最大子数组和的问题

```
public int maxSubArray(int[] nums) {
       int max = nums[0];
       for(int i=0;i<nums.length;i++){
          int tmpMax = nums[i];
          if(tmpMax>max) {
              max = tmpMax;
          }
        for(int j=i+1;j<nums.length;j++) {
          tmpMax += nums[j];
          if(tmpMax>max) {
              max = tmpMax;
          }
        }
       }
       System.out.println("max--"+max);
       return max;
   }
```

1

这是什么?

这是 Q-book LaTeX 书籍模板，当前版本为 v2.01。

这份模板主要基于上海交大的学位论文模板[1]修改得到，结合少量个人审美喜好，重新定制了定义、定理等环境。由于这个模板本是用于一项书籍翻译计划，因此其中的一些环境，例如“观察”、“规则”、“关键点”等，读者可能用不到，可以根据自己的需求适当修改。

你也可以通过邮箱 jcy74165@163.com 给我发邮件反映遇到的问题。不过作者水平有限，或许有些问题也无法解答，还请见谅。

1.1 文档说明

1.1.1 准备工作

要想灵活使用、魔改这个模板来撰写自己的书籍，需要对 *TeX* 系统有一定的了解，也需要掌握基本的 *TeX* 技能。

- TEX 系统：所使用的 TEX 系统要支持 XELATEX 引擎，且带有 ctex 2.x 宏包。一般来说，只要安装了的完整 TeXLive 或 MacTeX 发行版就不会出现问题。
- TEX 技能：尽管提供了对模板的必要说明，但这并不是一份“LATEX 入门文档”。用户应当有一定的 LATEX 使用经验。

[1] https://github.com/sjtug/SJTUThesis

图 1: 示例图片

1

这是什么?

这是 Q-book LaTeX 书籍模板，当前版本为 v2.01。

这份模板主要基于上海交大的学位论文模板[1]修改得到，结合少量个人审美喜好，重新定制了定义、定理等环境。由于这个模板本是用于一项书籍翻译计划，因此其中的一些环境，例如“观察”、“规则”、“关键点”等，读者可能用不到，可以根据自己的需求适当修改。

你也可以通过邮箱 jey74165@163.com 给我发邮件反映遇到的问题。不过作者水平有限，或许有些问题也无法解答，还请见谅。

1.1 文档说明

1.1.1 准备工作

要想灵活使用、魔改这个模板来撰写自己的书籍，需要对 *TeX 系统*有一定的了解，也需要掌握基本的 *TeX* 技能。

- TEX 系统：所使用的 TEX 系统要支持 XETEX 引擎，且带有 ctex 2.x 宏包。一般来说，只要安装了的完整 TeXLive 或 MacTeX 发行版就不会出现问题。
- TeX 技能：尽管提供了对模板的必要说明，但这并不是一份“LATEX 入门文档”。用户应当有一定的 LATEX 使用经验。

[1]https://github.com/sjtug/SJTUThesis

1

这是什么?

这是 Q-book LaTeX 书籍模板，当前版本为 v2.01。

这份模板主要基于上海交大的学位论文模板[1]修改得到，结合少量个人审美喜好，重新定制了定义、定理等环境。由于这个模板本是用于一项书籍翻译计划，因此其中的一些环境，例如“观察”、“规则”、“关键点”等，读者可能用不到，可以根据自己的需求适当修改。

你也可以通过邮箱 jey74165@163.com 给我发邮件反映遇到的问题。不过作者水平有限，或许有些问题也无法解答，还请见谅。

1.1 文档说明

1.1.1 准备工作

要想灵活使用、魔改这个模板来撰写自己的书籍，需要对 *TeX 系统*有一定的了解，也需要掌握基本的 *TeX* 技能。

- TEX 系统：所使用的 TEX 系统要支持 XETEX 引擎，且带有 ctex 2.x 宏包。一般来说，只要安装了的完整 TeXLive 或 MacTeX 发行版就不会出现问题。
- TeX 技能：尽管提供了对模板的必要说明，但这并不是一份“LATEX 入门文档”。用户应当有一定的 LATEX 使用经验。

[1]https://github.com/sjtug/SJTUThesis

核心定制代码如下：

```
\lstset{
    aboveskip={.3\baselineskip},
    basicstyle=\scriptsize\ttfamily\linespread{4},
    breaklines=false,
    columns=flexible,
    commentstyle=\color[rgb]{0.127,0.427,0.514}\ttfamily\itshape,
    escapechar=@,
    extendedchars=true,
    frame=single,
    identifierstyle=\color{black},
    inputencoding=latin1,
    keywordstyle=\color[HTML]{228B22}\bfseries,
    language=JavaScript,
    ndkeywordstyle=\color[HTML]{228B22}\bfseries,
    numbers=left,
    numberstyle=\tiny,
    prebreak = \raisebox{0ex}[0ex][0ex]{\ensuremath{\hookleftarrow}},
    stringstyle=\color[rgb]{0.639,0.082,0.082}\ttfamily,
    upquote=true,
    showstringspaces=false,
}
```