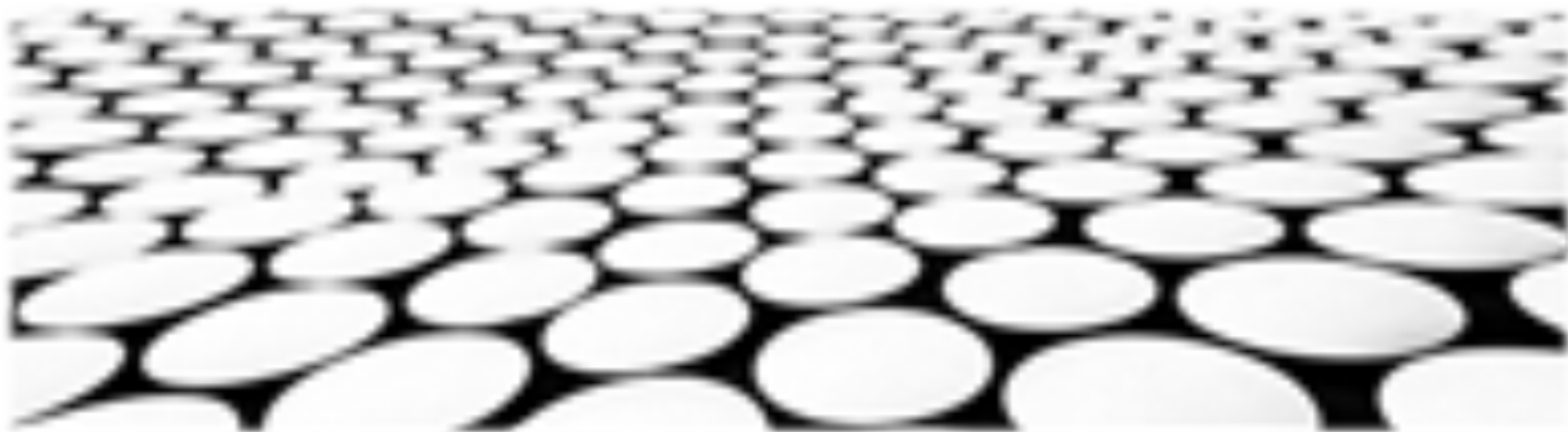


Linux系统的虚拟化与容器化技术





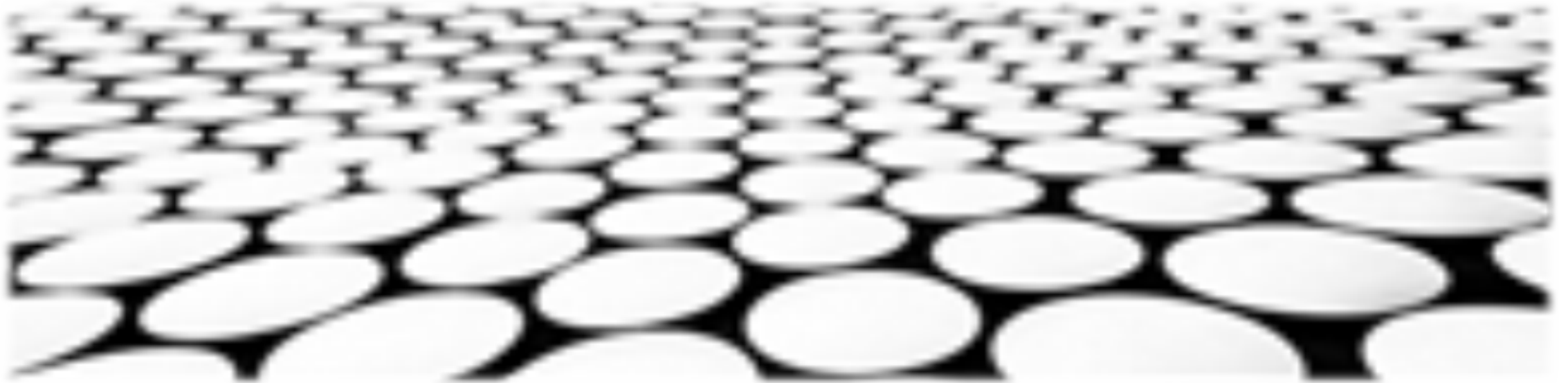
目录页

Contents Page

1. 虚拟化与容器化的区别与联系
2. Linux虚拟化平台的类型与特点
3. 容器化技术的本质与实现机制
4. Docker容器的架构与工作原理
5. 虚拟化与容器化的优势与劣势
6. 虚拟化和容器化技术的结合应用
7. 容器化技术在云计算中的应用场景
8. 虚拟化与容器化技术的未来发展趋势



Linux虚拟化平台的类型与特点





一、KVM

- 1、基于Linux内核的开源虚拟化平台，无需修改Guest操作系统，直接运行虚拟机；
- 2、硬件虚拟化技术，提供接近裸机性能的高效虚拟化体验；
- 3、集成在Linux内核中，开销低、可扩展性和灵活性高。

二、Xen

- 1、开源、轻量级的虚拟化平台，专门为虚拟化而设计；
- 2、采用半虚拟化技术，在Guest操作系统中加载额外的驱动程序，提高性能；
- 3、高度可扩展，支持多处理器和大量虚拟机，适用于云计算和高性能计算场景。

Linux虚拟化平台的类型与特点



三、Hyper-V

- 1、微软开发的，Windows Server系统集成的虚拟化平台；
- 2、基于硬件虚拟化技术，提供低开销、高性能的虚拟化体验；
- 3、与Windows生态系统紧密集成，提供丰富的管理工具和功能，适合中小型企业和组织。

四、VMwareESXi

- 1、VMware公司开发的商用虚拟化平台，主要用于企业级应用；
- 2、采用裸机虚拟化技术，提供接近物理机的性能；
- 3、功能强大，支持高可用性、容错和分布式资源管理，适用于大型企业和数据中心环境。



五、QEMU

- 1、自由开源的处理器仿真器和虚拟化管理器，支持多种硬件架构；
- 2、基于软件虚拟化技术，可运行于任何支持的平台，但性能受限；
- 3、用途广泛，可用于仿真、原型制作和开发嵌入式系统。

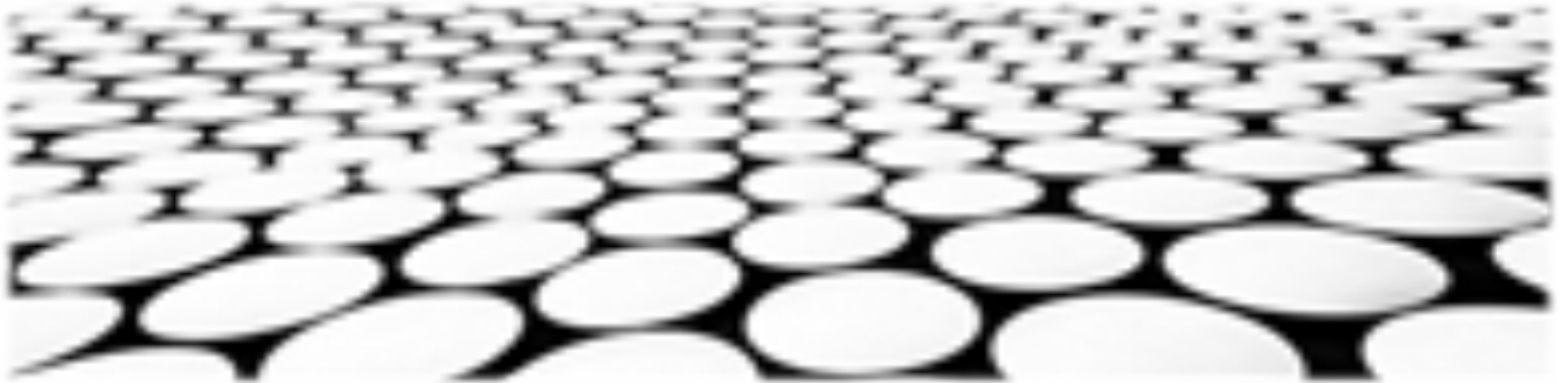
六、LXC

- 1、轻量级的容器化虚拟化技术，基于Linux控制组（cgroups）和命名空间实现；
- 2、在同一主机上创建多个隔离的容器，共享底层操作系统；





Docker容器的架构与工作原理



Docker容器的架构与工作原理一、容器的基础概念

1. 容器是一种轻量级的虚拟化技术，它共享主机内核，从而减少资源消耗。
2. 容器通过隔离应用程序及其依赖项来提高可移植性和可扩展性。
3. 容器通常使用一个镜像来定义其配置和文件系统，该镜像是容器创建过程中的模板。



二、Docker容器架构

1. Docker容器由一个容器镜像和一个容器运行时组成。
2. 容器镜像包含应用程序代码、所需的库和依赖项。
3. 容器运行时负责管理和运行容器，提供资源隔离、网络连接和其他必要功能。

三、容器运行时

1. Docker使用一个称为Docker Engine的容器运行时，它负责管理容器的生命周期。
2. Docker Engine提供了诸如镜像管理、容器启动和停止以及容器网络等功能。
3. 其他流行的容器运行时包括CRI-O、containerd和Kata Containers。

四、容器文件系统

1. Docker容器使用Union文件系统，它允许容器对镜像中的文件系统进行修改。
2. Union文件系统将容器的只读镜像文件系统与容器的可写层叠在一起。
3. 这种方法允许容器对镜像做出更改，而无需修改基础镜像。

五、容器通信

1. Docker容器通过称为Docker网络的网络层进行通信。
2. Docker网络提供了一种将容器相互连接并与外部网络交互的方法。
3. Docker支持多种网络模式，包括桥接、主机和覆盖网络。



六、容器管理

1. Docker提供了一系列命令和工具用于管理容器。
2. 流行工具包括docker build用于构建容器镜像，docker run用于运行容器，以及docker compose用于编排多个容器。

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：
<https://d.book118.com/548115113125006072>