Atitit 容器化的艺术 attilax著 v3 s66.docx

Atitit 容器化技术之道 attilax著

[1. 概念概念组成与原理成分 2](#_Toc16593)

[1.1. 容器是应用服务器中位于组件和平台之间的接口集合。 2](#_Toc14417)

[1.2. 有时候也指集合的概念，里面可以存放不同对象 2](#_Toc4125)

[1.3. 、新兴的虚拟化方式,Docker 2](#_Toc20166)

[2. 性状 3](#_Toc32210)

[3. 适用场景 主治 适应症 为什么要使用容器 3](#_Toc14343)

[3.1. 方便迁移 3](#_Toc23050)

[3.2. Docker在开发和运维中的优势 3](#_Toc19977)

[3.3. 更快速的交付和部署。可以使用镜像来快速构建一套标准的开发环境 4](#_Toc16417)

[3.4. 更轻松的迁移和扩展。 4](#_Toc11022)

[4. 缺点 不良反应 5](#_Toc25692)

[5. 禁忌 不适应场合 5](#_Toc20772)

[6. 注意事项 5](#_Toc27623)

[7. 相互作用与其他类库模式 5](#_Toc14007)

[8. 过量与滥用后果 5](#_Toc2476)

[9. 理解曲线动力学 5](#_Toc10386)

[9.1. 学习曲线 5](#_Toc11968)

[10. 原理容器特点 5](#_Toc13571)

[10.1. 隔离 5](#_Toc21385)

[10.2. 自包含 5](#_Toc12538)

[11. 常见的容器 6](#_Toc18121)

[11.1. 虚拟机 6](#_Toc12133)

[11.2. 模拟器 安卓模拟器等 游戏模拟器 6](#_Toc6696)

[11.3. Vm jdk net cli等 6](#_Toc17422)

[11.4. 微服务容器 6](#_Toc16379)

[11.5. 模块容器 6](#_Toc30408)

[11.6. Ui组件容器 iframe 6](#_Toc19543)

[11.7. Jee容器 解释器 6](#_Toc12467)

[11.8. 功能级别容器 模块化 servless 6](#_Toc7095)

[11.9. 代码级别 集合类list map等容器 泛型 6](#_Toc31956)

[11.10. 视频容器 mp4等 6](#_Toc29619)

[11.11. 文件夹foloder 6](#_Toc32354)

[11.12. Ns class method三级容器 6](#_Toc5793)

[11.13. Docker与集装箱 7](#_Toc21101)

[12. 具体实现 7](#_Toc18715)

[12.1. Docker容器就跟VM差不多， 7](#_Toc32718)

[12.2. 下表总结了使用Docker容器技术与传统虚拟机技术的特性比较。 7](#_Toc178)

[12.2.1. 对比传统虚拟机总结 7](#_Toc28541)

[13. 储藏 8](#_Toc28004)

[14. 包装与api接口 8](#_Toc30919)

[15. 执行标准 8](#_Toc23433)

[16. 生产组织与个人 attilax 8](#_Toc10228)

[17. 参考资料 8](#_Toc6734)

# 概念概念组成与原理成分

## 容器是[应用服务器](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=4341240" \t "http://baike.sogou.com/_blank)中位于组件和[平台](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=3467677" \t "http://baike.sogou.com/_blank)之间的[接口](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=77734" \t "http://baike.sogou.com/_blank)集合。

容器是伴随着[瘦客户端](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=53281869" \t "http://baike.sogou.com/_blank)系统的发展而诞生的。在开发[瘦客户端](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=53281869" \t "http://baike.sogou.com/_blank)系统时，开发人员要花费大量的精力去关注[线程安全](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=8425921" \t "http://baike.sogou.com/_blank)、事务、网络、资源等等[细节](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=101249" \t "http://baike.sogou.com/_blank)，从而降低了开发效率。由于这些对这些[细节](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=101249" \t "http://baike.sogou.com/_blank)的解决方法一般是固定不变，或者只有参数改变的，所以从代码重用和[设计模式](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=123729" \t "http://baike.sogou.com/_blank)的角度出发，开发人员将这些底层[细节](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=101249" \t "http://baike.sogou.com/_blank)提取出来，做成平台，并提供一定的接口。这样，业务开发人员就不需要在关注与这些底层[细节](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=101249" \t "http://baike.sogou.com/_blank)的实现，而专注于业务逻辑的实现。

容器一般位于应用服务器之内，由应用服务器负责加载和维护。一个容器只能存在于一个应用服务器之内，一个应用服务器可以建立和维护多个容器。

容器一般遵守可配置的原则，即容器的使用者可以通过对容器参数的配置，来达到自己的使用需求，而不需要修改容器的代码。

## 有时候也指集合的概念，里面可以存放不同对象

、

## 、新兴的虚拟化方式,Docker

跟传统的虚拟化方式相比具有众多的优势。 更高效的利用系统资源 由于容器不需要进行硬件虚拟及运...

# 性状

# 适用场景 主治 适应症 为什么要使用容器

## 方便迁移

举个简单的应用场景的例子。假设用户试图基于最常见的LAMP（Linux + Apache + MySQL + PHP）组合来运维一个网站。按照传统的做法，首先，需要安装Apache、MySQL 和PHP以及它们各自运行所依赖的环境；之后分别对它们进行配置（包括创建合适的用户、配置参数等）；经过大量的操作后，还需要进行功能测试，看是否工作正常；如果不正常，则意味着更多的时间代价和不可控的风险。可以想象，如果再加上更多的应用，事情会变得更加难以处理。

更为可怕的是，一旦需要服务器迁移（例如从阿里云迁移到腾讯云），往往需要重新部署和调试。这些琐碎而无趣的“体力活”，极大地降低了工作效率。

而Docker提供了一种更为聪明的方式，通过容器来打包应用，意味着迁移只需要在新的服务器上启动需要的容器就可以了。这无疑将节约大量的宝贵时间，并降低部署过程出现问题的风险。

## Docker在开发和运维中的优势

对开发和运维（DevOps）人员来说，可能最梦寐以求的就是一次性地创建或配置，可以在任意环境、任意时间让应用正常地运行。而Docker恰恰是可以实现这一终极目标的瑞士军刀。

## 更快速的交付和部署。可以使用镜像来快速构建一套标准的开发环境

使用Docker，开发人员可以使用镜像来快速构建一套标准的开发环境；开发完成之后，测试和运维人员可以直接使用相同环境来部署代码。Docker可以快速创建和删除容器，实现快速迭代，大量节约开发、测试、部署的时间。并且，各个步骤都有明确的配置和操作，整个过程全程可见，使团队更容易理解应用的创建和工作过程。

更高效的资源利用。Docker容器的运行不需要额外的虚拟化管理程序（Virtual Machine Manager，VMM，以及Hypervisor）支持，它是内核级的虚拟化，可以实现更高的性能，同时对资源的额外需求很低。

## 更轻松的迁移和扩展。

Docker容器几乎可以在任意的平台上运行，包括物理机、虚拟机、公有云、私有云、个人电脑、服务器等。 这种兼容性让用户可以在不同平台之间轻松地迁移应用。

# 缺点 不良反应

# 禁忌 不适应场合

# 注意事项

# 相互作用与其他类库模式

# 过量与滥用后果

# 理解曲线动力学

## 学习曲线

# 原理容器特点

## 隔离

## 自包含

# 常见的容器

## 虚拟机

## 模拟器 安卓模拟器等 游戏模拟器

## Vm jdk net cli等

## 微服务容器

## 模块容器

## Ui组件容器 iframe

## Jee容器 解释器

## 功能级别容器 模块化 servless

## 代码级别 集合类list map等容器 泛型

## 视频容器 mp4等

## 文件夹foloder

## Ns class method三级容器

class 是method的容器。Method是statement的容器

## Docker与集装箱

# 具体实现

## Docker容器就跟VM差不多，

从运维的角度，会希望像管理VM那样管理Docker容器，但是Docker容器很难troubleshooting，因为默认来说它没有SSH，你要怎么登陆到一个容器里去查看里面发生了什么问题?Troubleshooting，这是一个最大的问题。

## 下表总结了使用Docker容器技术与传统虚拟机技术的特性比较。

[](http://s3.51cto.com/wyfs02/M00/59/DE/wKiom1TtaGbjjaPQAABVeQuym7E377.jpg)

### ****对比传统虚拟机总结****

| **特性** | **容器** | **虚拟机** |
| --- | --- | --- |
| 启动 | 秒级 | 分钟级 |
| 硬盘使用 | 一般为 MB | 一般为 GB |
| 性能 | 接近原生 | 弱于 |
| 系统支持量 | 单机支持上千个容器 | 一般几十个 |

# 储藏

# 包装与api接口

# 执行标准

# 生产组织与个人 attilax

# 参考资料

《Docker 容器与容器云 第2版》(浙江大学SEL实验室)【简介\_书评\_在线阅读】 - 当当图书.mhtml

作者:: 绰号:老哇的爪子claw of Eagle 偶像破坏者Iconoclast image-smasher

捕鸟王"Bird Catcher kok 虔诚者Pious 宗教信仰捍卫者 Defender Of the Faith. 卡拉卡拉红斗篷 Caracalla red cloak 万兽之王 纵火者

简称：： st Emir Attilax Akbar 圣 埃米尔 阿提拉克斯 阿克巴

全名：：st Emir Attilax Akbar bin Mahmud bin attila bin Solomon bin adam Al Rapanui 圣 埃米尔 阿提拉克斯 阿克巴 本 马哈茂德 本 阿提拉 本 所罗门 本亚当 阿尔 拉帕努伊

常用名：艾提拉（艾龙）， EMAIL:1466519819@qq.com

头衔：

|  |  |
| --- | --- |
| uke | Emir Uke部落首席大酋长，ati协会创始人  uke总部o2o负责人，全球网格化项目创始人，  圣阿提拉克斯国王 |
| 科技领域 | UTSC uke技术标准化委员会委员长 uke 首席cto 软件部门总监 技术部副总监 研发部门总监主管 产品部副经理 项目部副经理 uke科技研究院院长 uke软件培训大师  Ati组织科研研究院创始人 |
| 文艺领域 | ， ，， uke机车协会主任 uke纹身协会  uke交友协会会长 uke捕猎协会会长  Ati文艺协会会长 ati文学协会 |
| 行政领域 | Gchsp总裁 gchsp常委 GsP创始人 |
| 媒体传播领域 | uke出版社编辑总编 宣传布道总策划  Ati传媒总部 |
| 渔猎军事领域 | uke保安部首席大队长  Uke 户外运动协会理事长 度假村首席大村长  Ati打猎协会 |
| 法学 | 法学研究会 制度研究会 |
| 管理领域 | 工商管理学 公共管理与社会服务  ,uke制度检查委员会副会长 |
| 教育领域 | uec学院校长， uecip图像处理机器视觉专业系主任 uke文档检索专业系主任  Uke图像处理与机器视觉学院首席院长  uke终身教育学校副校长  靓号研究院 |
| 经济领域 | uke波利尼西亚区大区连锁负责人 汤加王国区域负责人 uke克尔格伦群岛区连锁负责人，莱恩群岛区连锁负责人，uke布维岛和南乔治亚和南桑威奇群岛大区连锁负责人  Uke软件标准化协会理事长理事长 Uke 数据库与存储标准化协会副会长  直达巴士西北区负责人 直达巴士长沙与西安分部部长  润昌通讯软件事业部总裁 执行长 分部负责人 执行委员会主席  Ati经济研究所 |
| 历史领域 | 历史事业部 ati历史研究院 |
| 社会科学领域 | 社科学院 ati文化部 |
| 自然科学领域 | Uke研究院院长兼首席研究员 科学家  Ati自然科学研究院 |
| 宗教神学领域 | uke宗教与文化融合事务部部长 大师master  uke制度与重大会议委员会委员长 ati宗教事务所 |
| 医学领域 | Uke医院 与医学院方面的创始人 |

转载请注明来源：attilax的专栏 <http://blog.csdn.net/attilax>

<http://www.cnblogs.com/attilax/>

Microblog

<http://weibo.com/u/5941179815> (common attilax)

<https://weibo.com/p/1005055941179815> （attilax201707,bek weibo）

<http://weibo.com/u/5487832265> (tech,for blog auto gene)

知乎空间

https://www.zhihu.com/people/ati-att/activities

Qq 1466519819 小号112237553

微信attilax 小号attilax201708

微博 attilax2016 小号attilax201707

--Atiend v19