

# Docker 在公有云中的一些实践

—— 论势不可挡的容器化趋势

# 关于我

- 王子亭
- 20 岁
- Node.js 开发者
- LeanCloud
- <https://jysperm.me>
- GitHub: jysperm





价 格： **24.00**元

商家承诺： （此商品性质不支持7天退货 [详情](#)）

付费周期： 1季度

主机所在地： 日本

产品套餐： 套餐1

！ 您现在查看的是[交易快照](#)

但目前暂时不提供查看该商品的信息。

所在地区： 辽宁沈阳

商品类型： 全新

掌柜档案

**精英盒子** [给我留言](#)

卖家信用： **378**

卖家好评率： 99.74%

买家信用： **173**

买家好评率： 100.0%

注册时间： 2011年06月25日

**挑选更多商品**  
**进入掌柜店铺**

商品详情

付款方式

购物保障

主机空间： 500M-1G

主机线路环境： 海外

主机功能支持： FTP 管理 ASP.NET PH...

附带邮局空间： 不带邮局

操作系统： UNIX系列

主机WEB方式管理： 支持

付费周期： 1个月 1季度

虚拟主机价格： 0-99.99元

主机服务商： 其它服务商

主机所在地： 美国 日本

产品套餐： 套餐1

RP 主机是一款为技术宅(Geek)提供的 Linux 虚拟主机，支持 Python, Node.js, Golang 等小众语言。

RP 主机提供经典的 Linux 环境，你可以充分利用来自 Linux 的工具进行网站开发和服务器架设。

相比于其他虚拟主机，RP主机具有更高的使用门槛，购买前请确认你对 Linux 以及你要使用的技术有一定了解。客服会提供一定程度的指导，如果你抱着学习的目的，请准备好足够的耐心。

更多信息请参见官网：<http://rpvhost.net>

请先在RP主机官网注册帐号，**购买时备注你在 RP 主机的节点和用户名**，如忘记备注，请在 RP 主机发工单并提供淘宝订单号。

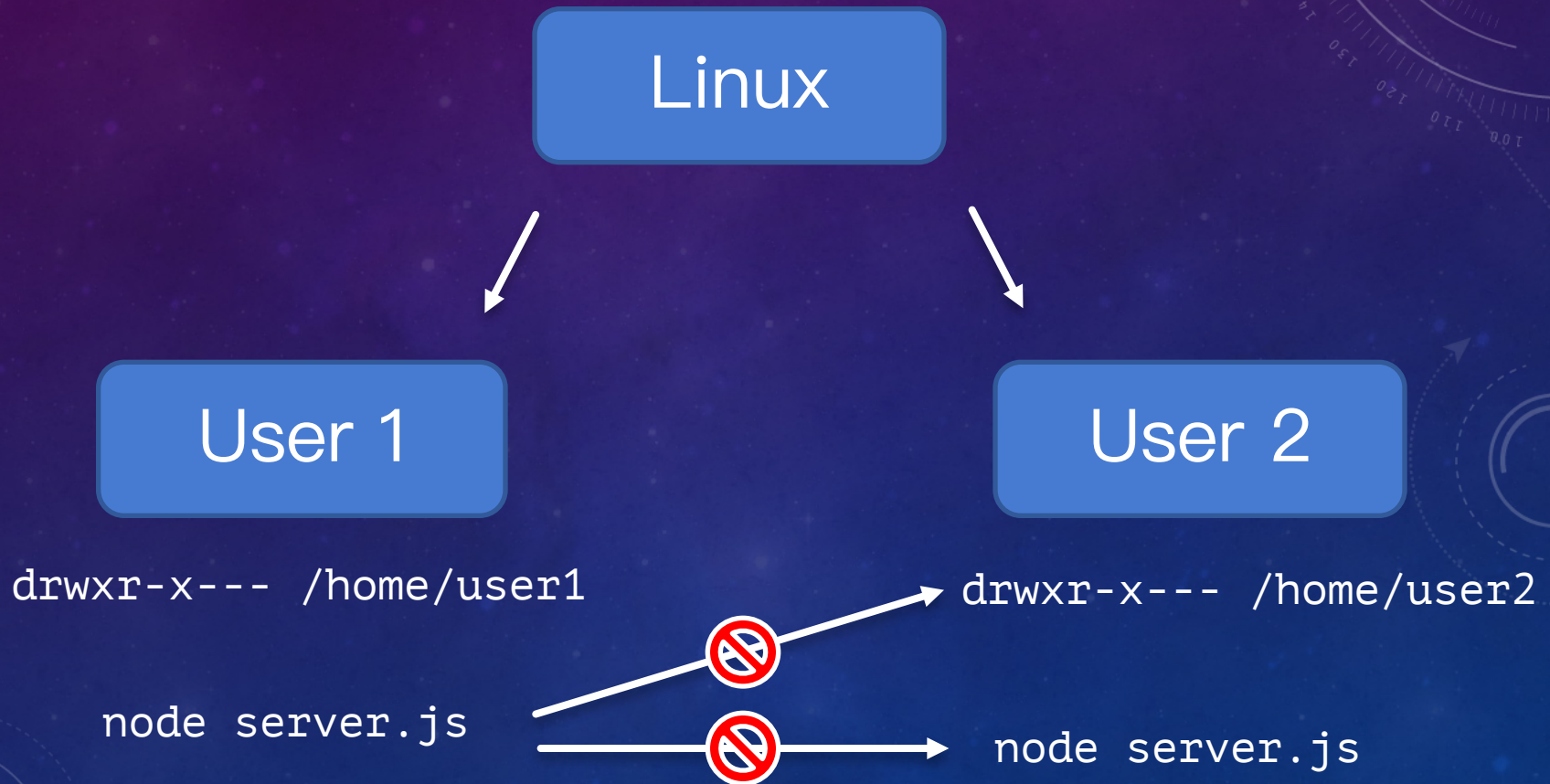
如有疑问请通过官网的**工单或邮件**进行咨询，客服并不经常登录阿里旺旺。

**FAQ**(详细条款参见官网)

Q: 这主机相当于多少金额?



# UNIX Permissions



# UNIX Socket

0.0.0.0:80

Nginx

User 1

node server.js => ~/server.sock srw-rw----

redis-server => ~/redis.sock srw-----

# MongoDB

## 新建数据库

## 设置 MongoDB 密码

数据库

磁盘占用

test\_1

16M



# MySQL

## 设置 MySQL 密码

数据库

体积

磁盘占用

test\_blog

12M

18M

## 财务

余额: 99.00 CNY

充值

工单

首选项

## 付费方案

ShadowSocks	按量付费, 0.6 CNY / G	退订方案
-------------	-------------------	------

### GreenShadow

- v0.8.0
- [Source on Github](#)

## ShadowSocks

连接

端口12345

密码DzizKKVly22aHQwm重置

加密算法rc4-md5修改

流量

余额折合约 165.0 G 流量

0.8G

过去 24 小时

1.1G

过去 7 天

0.0G

过去 30 天



# CPU 和内存限制

```
for user in system.users
    if system.cpuUsage > 80
        if user.cpuUsages > user.cpuLimit
            for proc in users.processes
                proc.renice 10

    if system.memoryUsage > 80
        if user.memoryUsages > user.memoryLimit
            sortByMemory(user.processes).last.kill()
```



# 隔离（并没有）

```
$ ps aux
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
user2	19230	0.1	0.1	125884	1209	?	S	Jun04	0:54	node
user3	18222	0.0	0.2	127028	2611	?	S	22:55	0:00	python

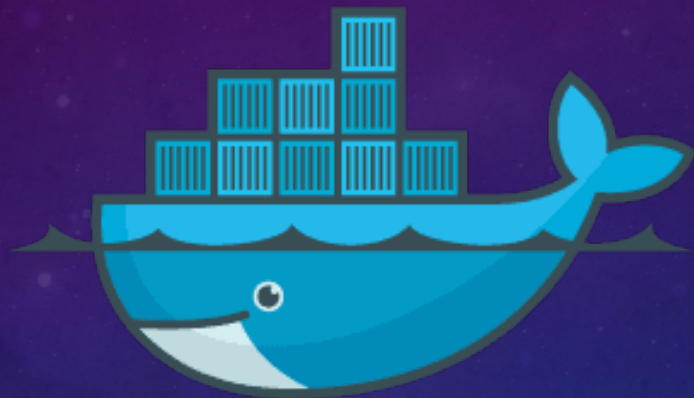
```
$ ls -l /home
```

drwxr-x---	8	user2	user2	4096	Jun	4	17:03	user2/
drwxr-x---	3	user3	user3	4096	Apr	21	2014	user3/

```
$ netstat -lnp
```

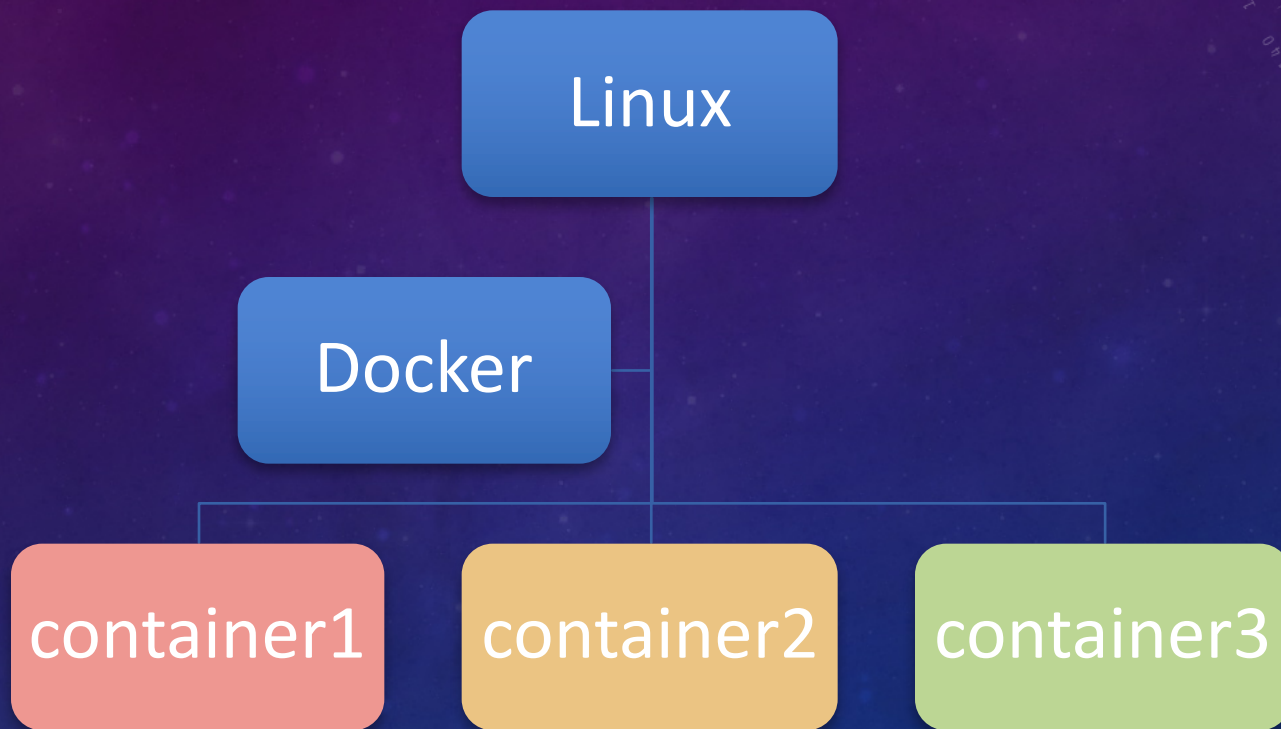
Proto	Local Address	State	PID
tcp	0.0.0.0:80	LISTEN	5824
tcp	0.0.0.0:29437	LISTEN	29833
tcp	0.0.0.0:12644	LISTEN	19157

# Docker



—— 基于 Linux 的轻量级虚拟化技术

# 「容器 (Container)」





# 「隔离」和「资源控制」

## Linux namespaces

- mount (文件系统)
- UTS (主机名)
- PID & IPC (进程)
- network (网络)
- user (用户)

## Linux cgroups

- cpuset & cpu & cpuacct
- memory (内存)
- blkio (磁盘 IO)



# Root Filesystem

container1

/  
/bin  
/etc  
/home  
/usr  
/var

container2

/  
/bin  
/etc  
/home  
/usr  
/var

container3

/  
/bin  
/etc  
/home  
/usr  
/var

500M

+

500M

+

500M

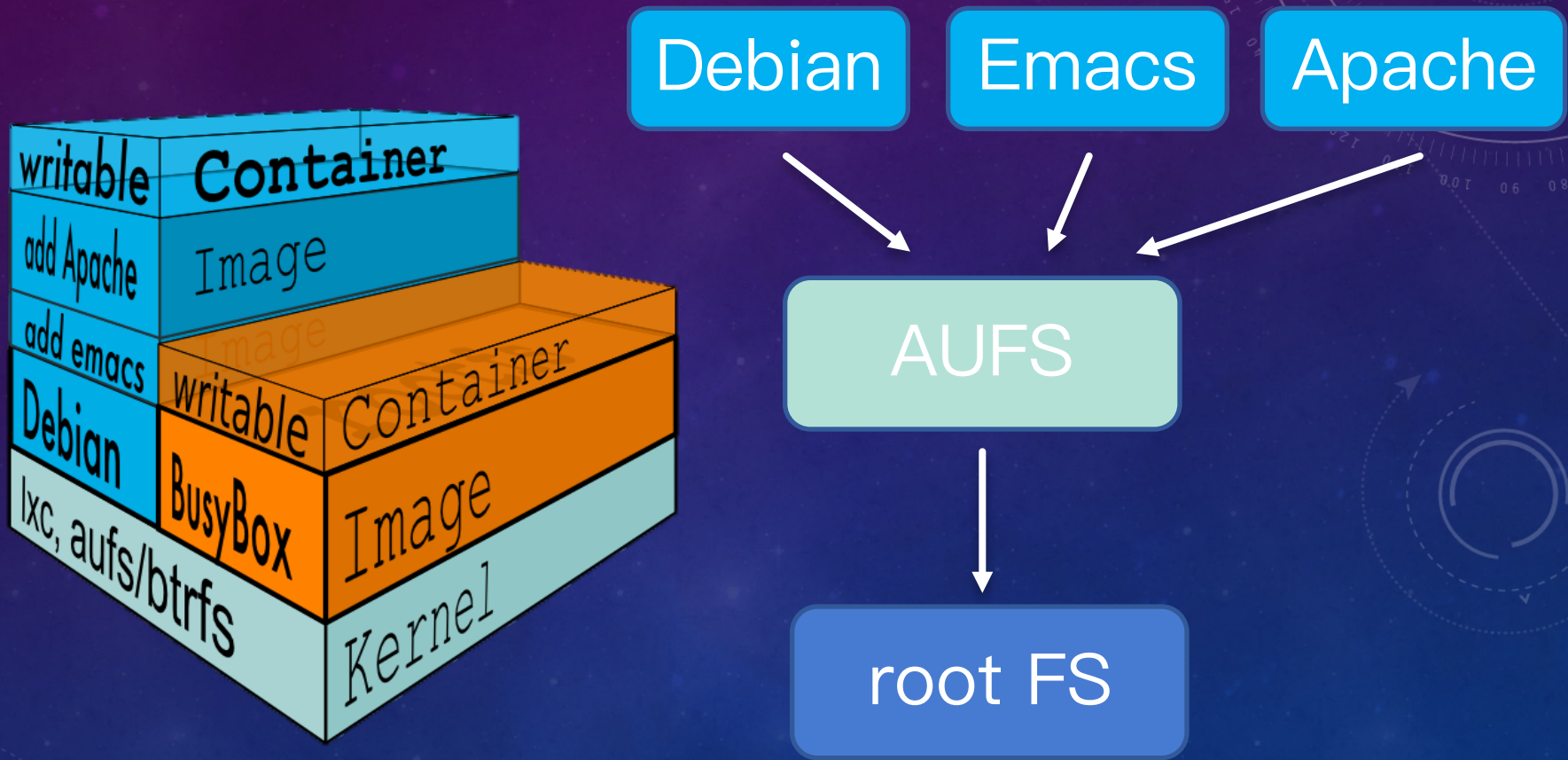
+

...

=

?

# Union Filesystem (AUFS、OverlayFS)



/ /bin /etc /home /usr /var ...

# Root FS with Union FS

readonly:

Apache

Debian

Nginx

container1

container2

container3

writable

writable

writable

/

/bin

/etc

/home

/usr

/var

/

/bin

/etc

/home

/usr

/var

/

/bin

/etc

/home

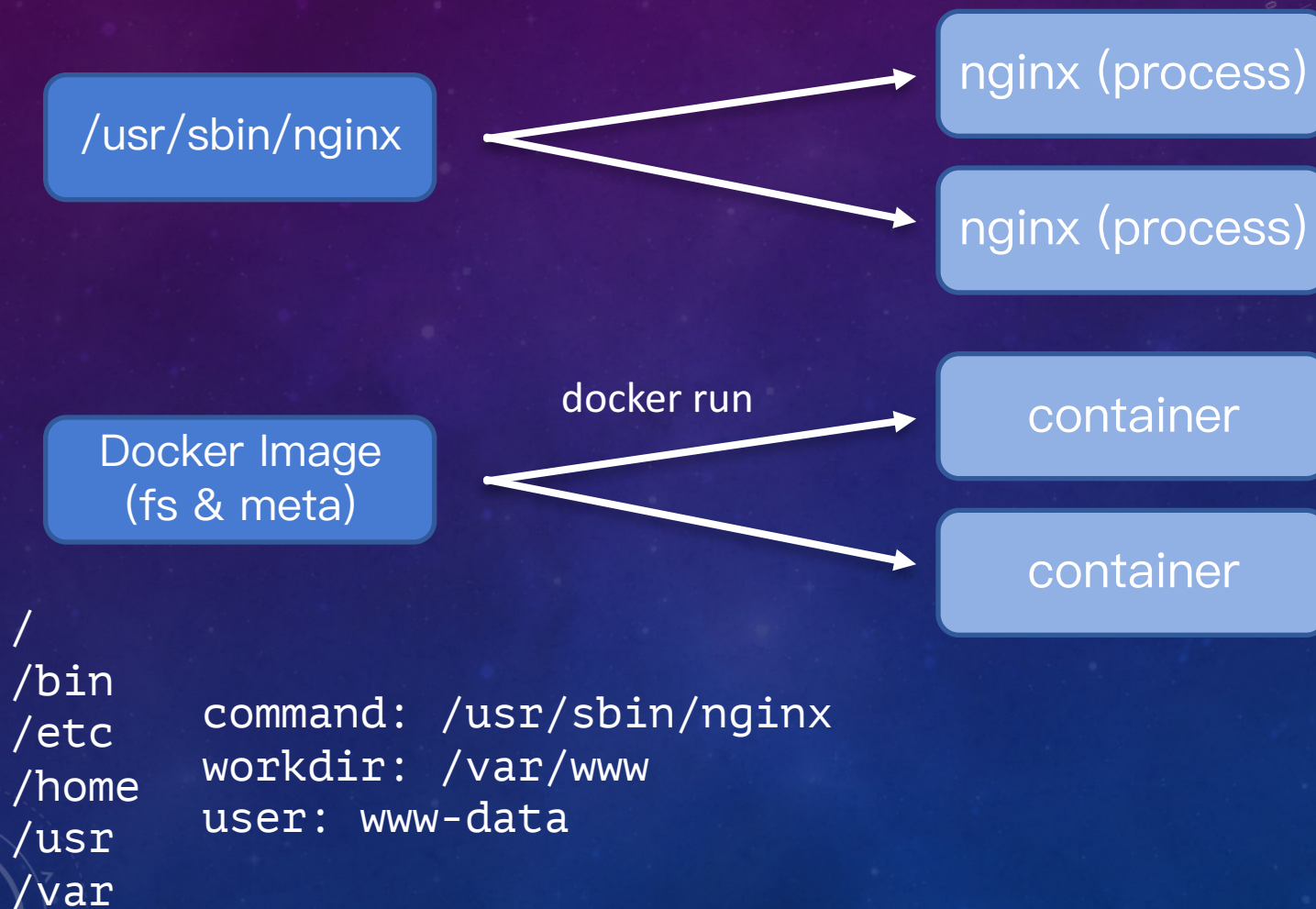
/usr

/var





# Docker 镜像 (Image)



/  
/bin  
/etc  
/home  
/usr  
/var

command: /usr/sbin/nginx  
workdir: /var/www  
user: www-data



# 构建镜像

## Dockerfile

```
FROM debian
RUN apt-get install nginx
ADD nginx.conf /etc/nginx/
CMD /usr/sbin/nginx
```

docker build

layer1

FROM debian

/bin /etc /home /usr /var ...

layer2

RUN apt-get install nginx

/usr/sbin/nginx ...

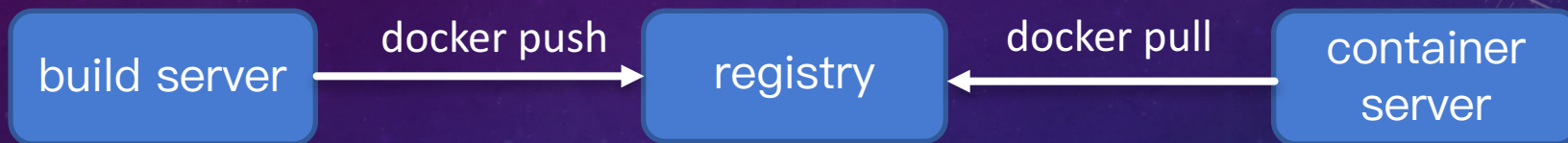
layer3

ADD nginx.conf /etc/nginx/

/etc/nginx/nginx.conf

Nginx (image)

# 传输和存储



```
FROM debian  
RUN apt-get ...  
ADD nginx.conf ...
```

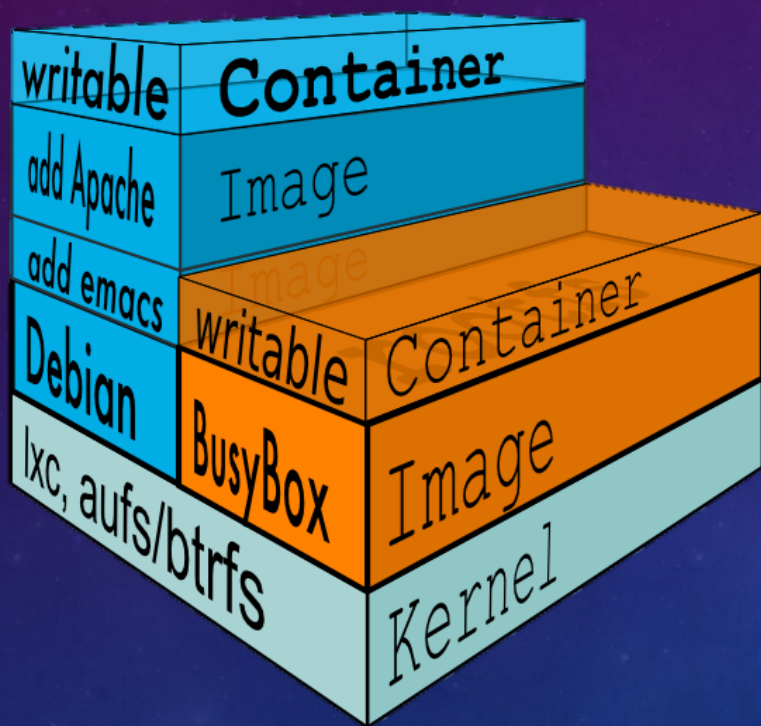


```
FROM debian  
RUN apt-get ...  
ADD nginx.conf ...
```



```
FROM debian  
RUN apt-get ...  
ADD nginx.conf ...
```

# Docker Image



- 发布应用就像发布可执行文件一样简单
- 通过联合文件系统减少传输和存储开销
- 对应用版本的一个精确快照



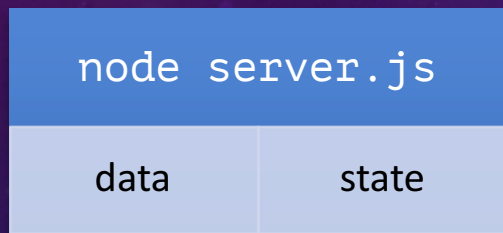
# 在应用容器上可能会发生任何事情

node server.js

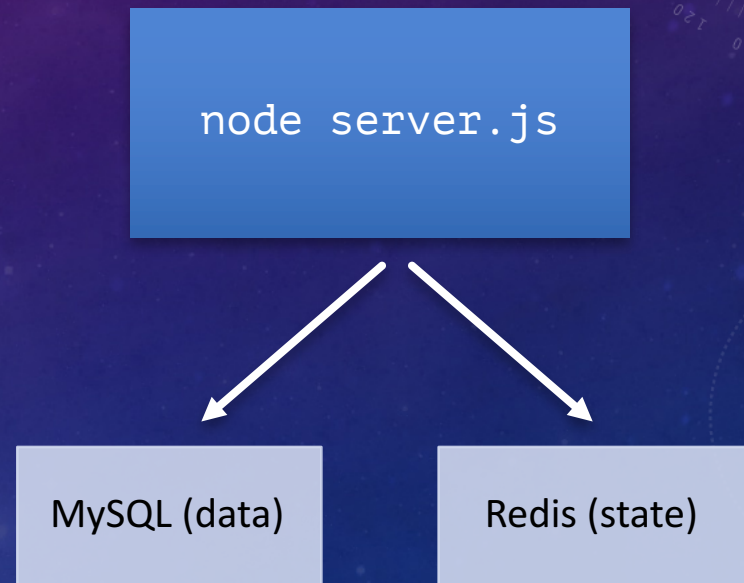
- 崩溃重启
- 内存泄漏
- 需要更新代码
- 不再响应新的请求
- 意外地占用大量 CPU



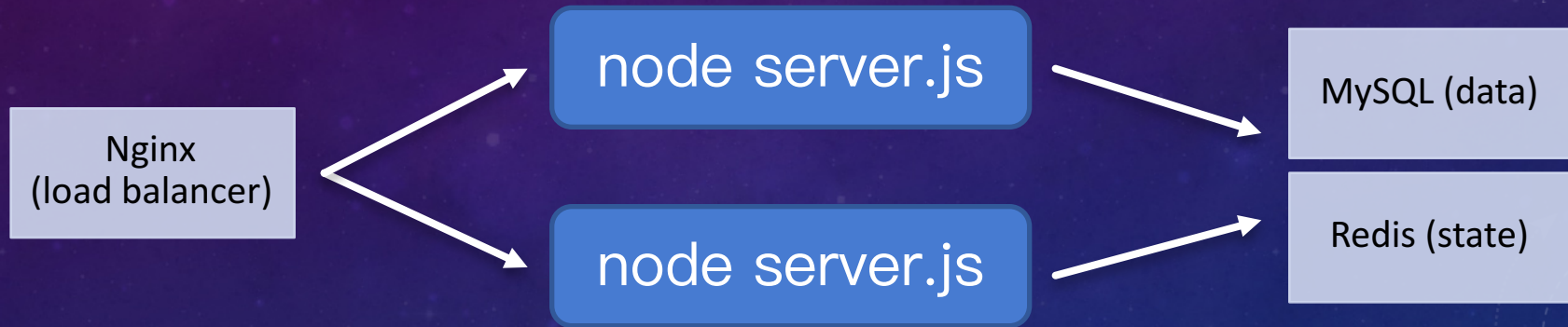
# 有状态



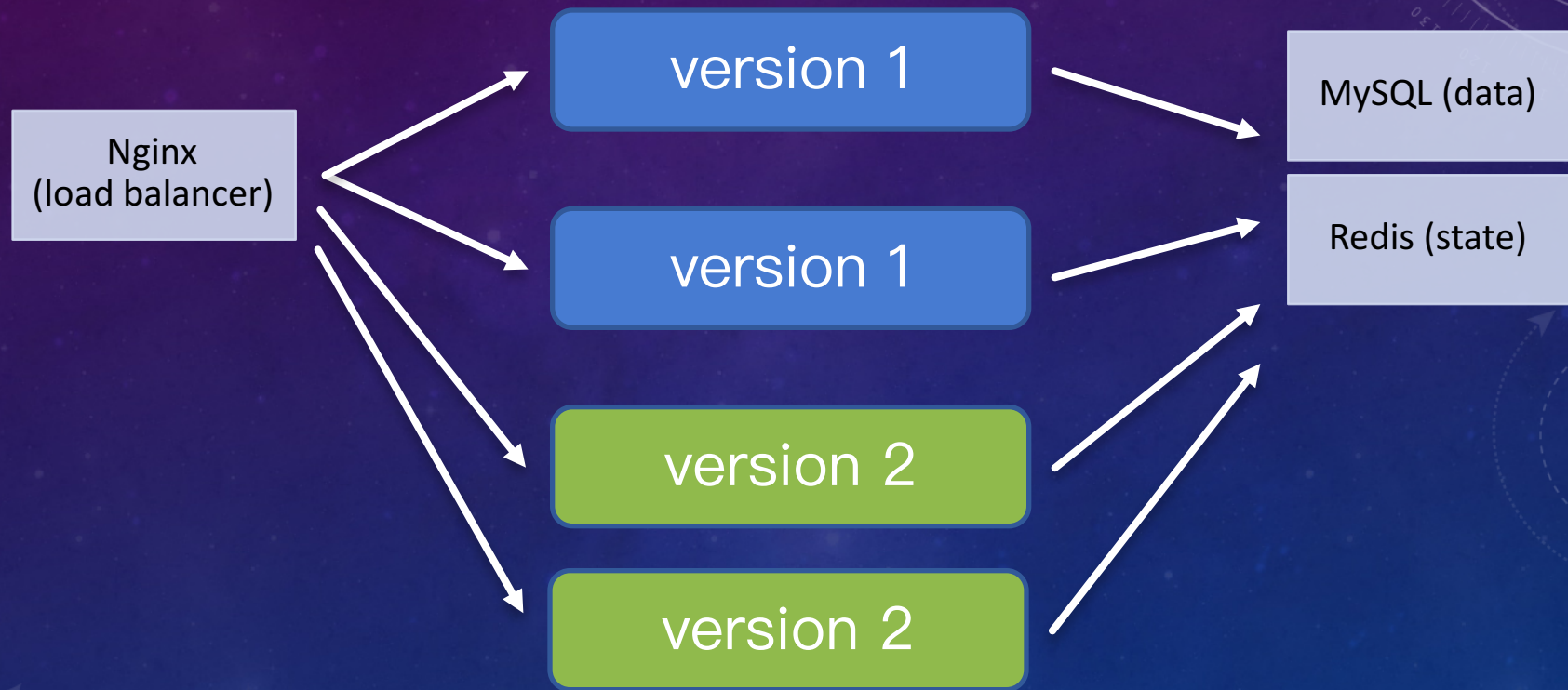
# 无状态



# Stateless Container



# Blue-Green Deployment





## 无法避免的故障

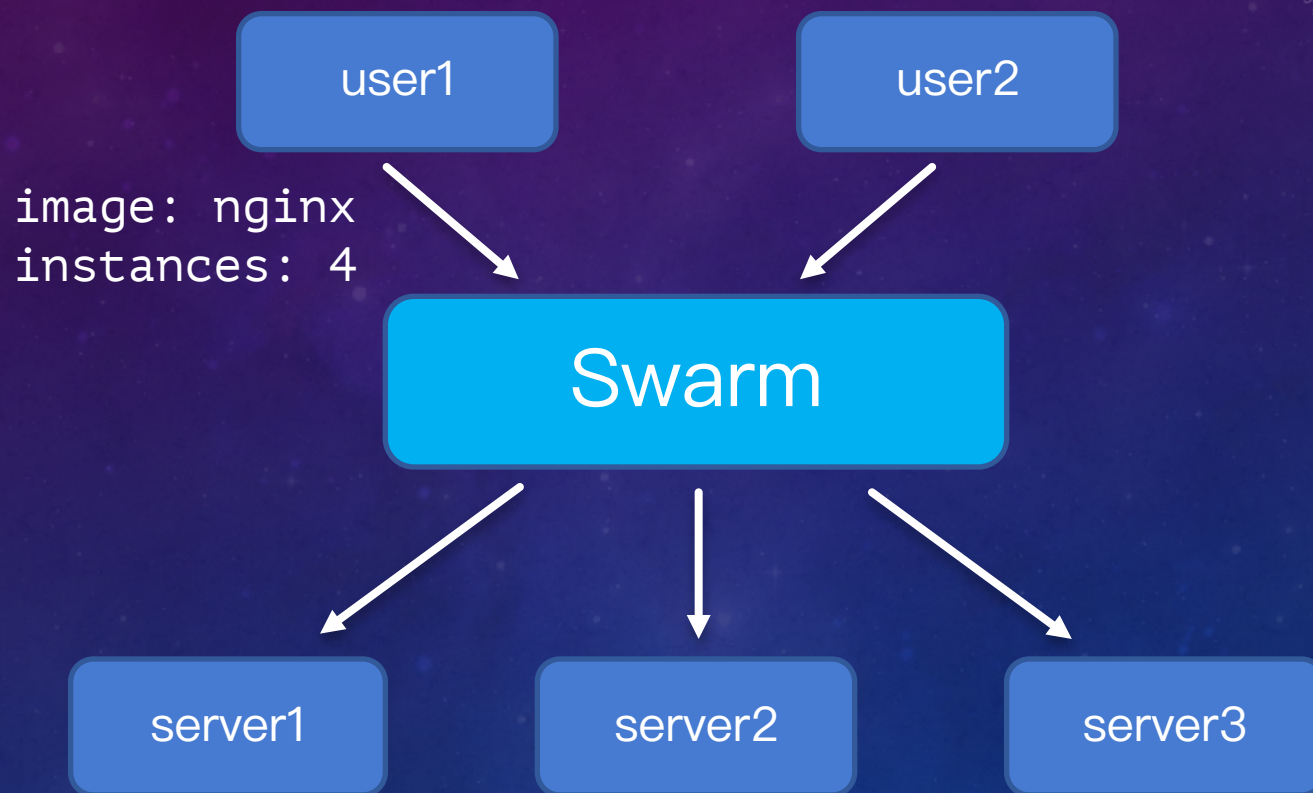
- 容器崩溃（重启）
- 容器僵死（不再处理新的请求）
- 服务器崩溃或失联

## 计划中的任务

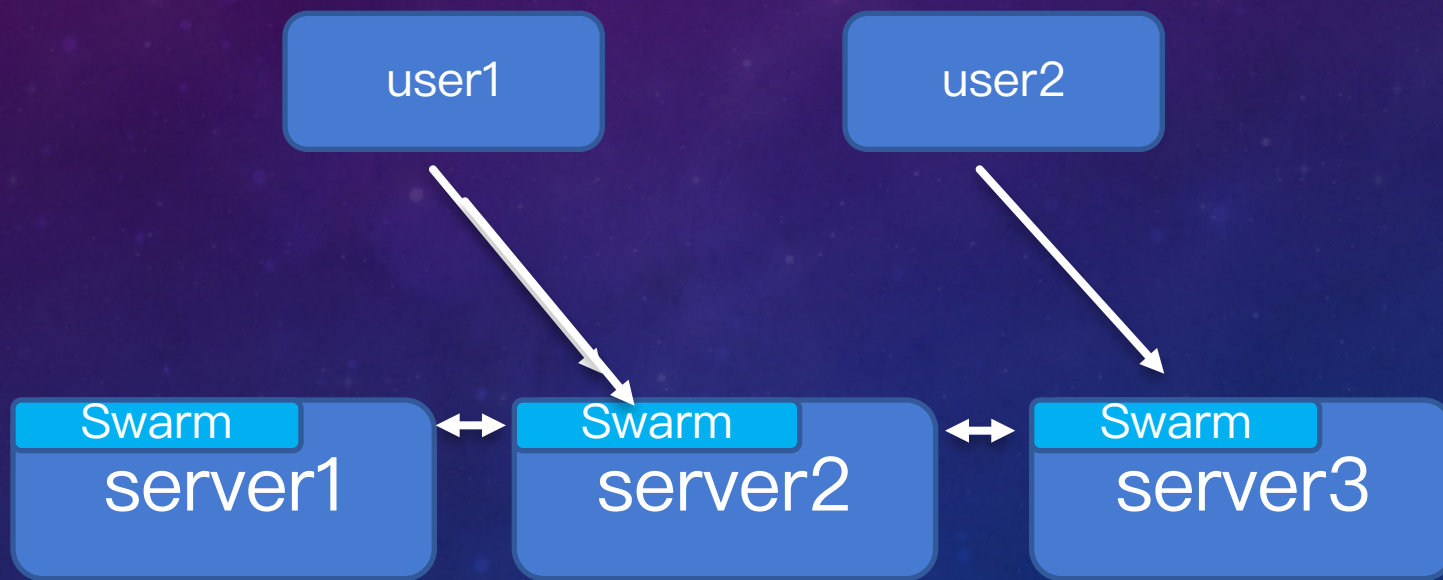
- 上线新版本（蓝绿部署）、回滚到旧版本
- 根据服务器负荷调度容器



# 集群管理器



# 集群管理器（分布式）



# Open Source

- Mesos & Marathon
- Docker Swarm
- Kubernetes

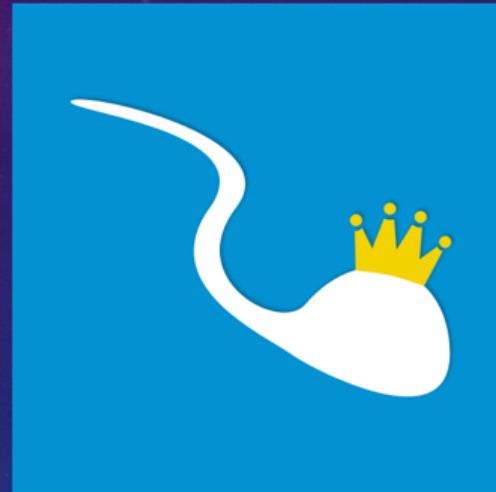


# 小结

- Docker 是一种轻量级的容器虚拟化解决方案
- Docker 基于 Linux 的 namespaces 和 cgroups 来实现隔离和资源控制
- 镜像是用于创建容器的模板，是对一个应用所需要的所有环境的封装
- 我们应该将应用容器设计为无状态的以便灵活地对其进行调整
- 我们应该用分布式的工具去管理集群，隐藏掉具体运行环境的细节

# Q & A

- Docker
- Node.js
- LeanCloud
- Bitcoin
- GPG



- <https://jysperm.me>
- GitHub: jysperm

Connext