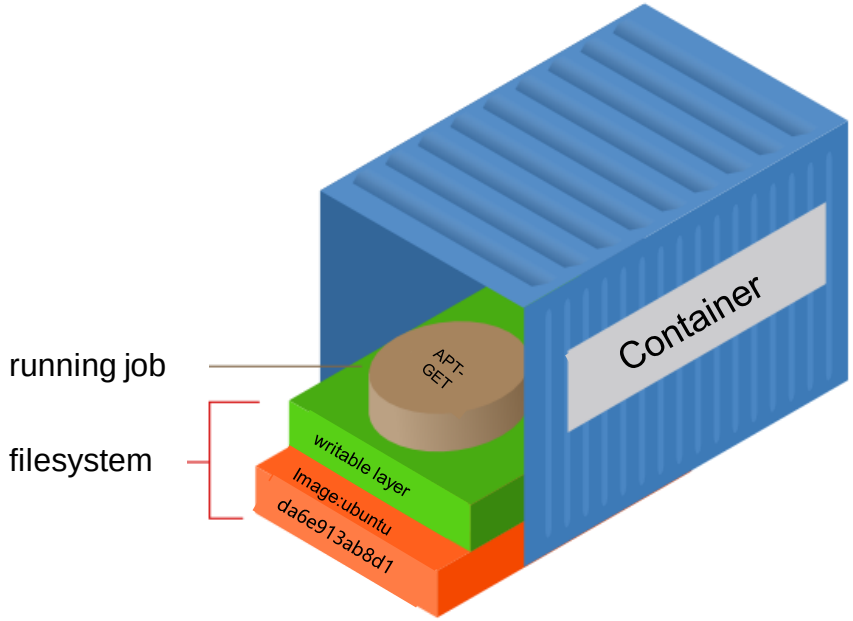
什么是容器?

- 容器是一种**轻量级**、**可移植**（同处理器平台移植）、自包含的**软件打包技术**，使应用程序可以在几乎任何地方以相同的方式运行。
- 开发人员在自己笔记本上创建并测试好的容器，**无需任何修改**就能够在生产系统的虚拟机、物理服务器或公有云主机上运行。
- 当前最流行的容器技术**Docker支持X86/ARM等多平台**

运输业



Container

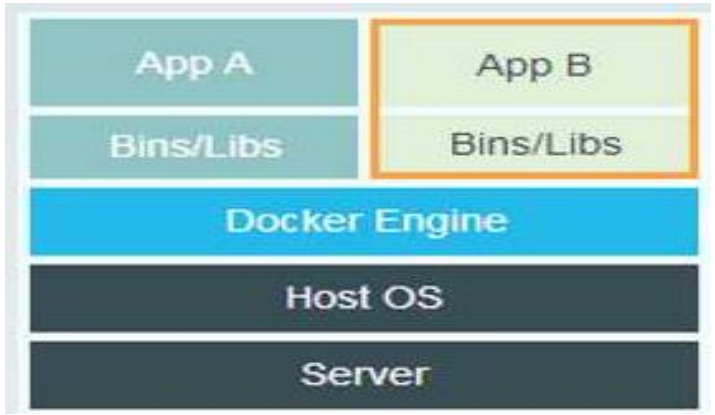
容器在X86和鲲鹏平台无差异的运行，应用无感知

容器与虚拟机区别是什么?

- 容器和虚拟机之间的主要区别在于虚拟化层的位置和操作系统资源的使用方式。

虚拟机

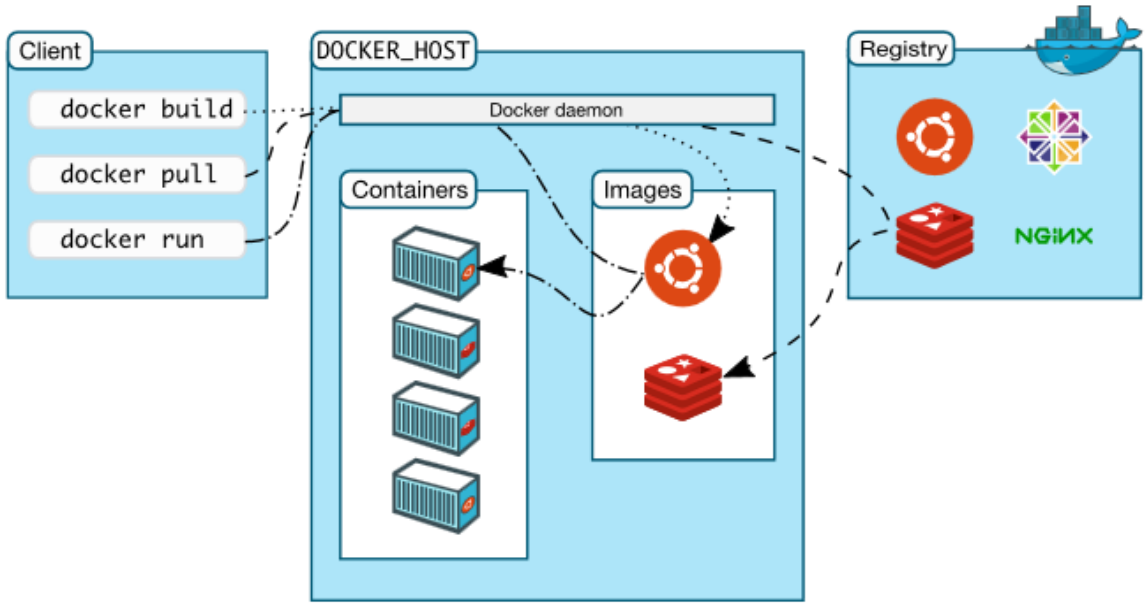
容器



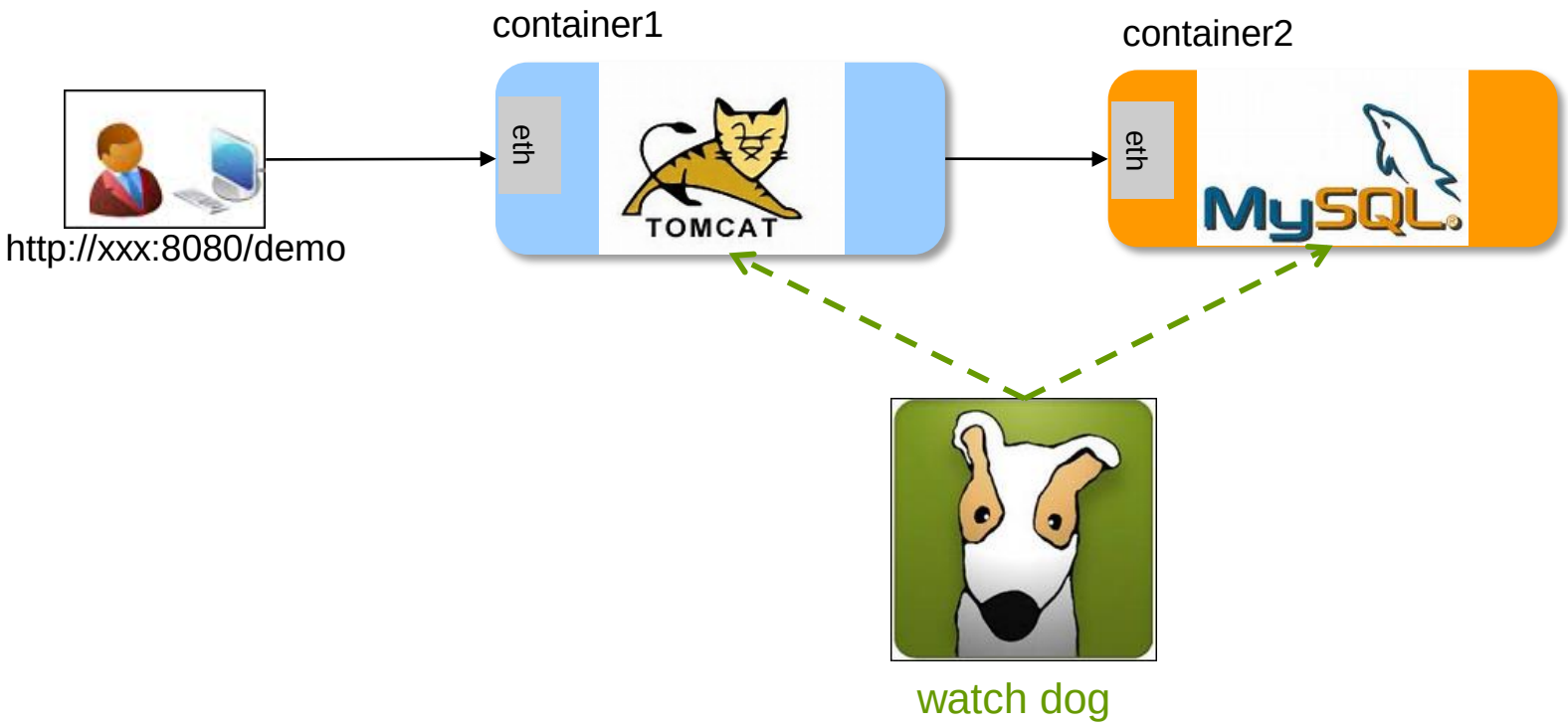
参数	容器（Docker）	对比	虚拟机
快速创建，删除	启动应用	>	启动Guest OS+启动应用
交互，部署	容器镜像	=	虚拟机镜像
密度	单Node 100~1000	>	单Node 10~100
更新管理	迭代式更新，通过修改Dockerfile，对增量内容进行分发，存储，节点启动	>	向虚拟机推送安装，升级应用软件补丁包
启动时间	秒级启动	>	分钟级启动
轻量级	镜像大小通常以M为单位	>	虚拟机以G为单位
性能	docker共享宿主机内核，系统级虚拟化，占用资源少，没有Hypervisor层开销，性能基本接近物理机	>	虚拟机需要Hypervisor层支持，虚拟化一些设备，具有完整的GuestOS，虚拟化开销大，因而降低性能，没有容器性能好。
安全性	Docker具有宿主机root权限，有一定安全隐患	<	硬件隔离，相对安全
高可用性	通过业务本身的高可用性来保证	<	武器库丰富：快照，克隆，HA，动态迁移，异地容灾，异地双活
使用要求	共享宿主机内核，不用考虑CPU是否支持虚拟化技术	>	基于硬件的完全虚拟化，需要硬件CPU虚拟化技术支持

Docker是如何工作的?

Docker客户端（Client）	Docker客户端通过命令行或者其他工具使用Docker API 与 Docker的守护进程通信。
Docker主机（Host）	一个物理或者虚拟的机器用于执行 Docker 守护进程和容器。
Docker镜像（Images）	Docker镜像是用于创建 Docker 容器的模板。
Docker容器（Container）	容器是独立运行的一个或一组应用。
Docker Daemon	服务端守护进程，负责接收客户端的指令，并处理这些请求（创建、运行、分发容器）。
Docker仓库（Registry）	Docker仓库用来保存镜像，可以理解为代码控制中的代码仓库。 Docker Hub提供了庞大的镜像集合供使用。



基于Docker容器如何部署高可靠Web应用系统?



1. 部署一个MySql的容器
2. 为MySql配置可供内部访问的ip
3. 部署一个TOMCAT容器
4. 为Tomcat配置可供对外访问的ip，并配置从MySql中读取数据
5. 部署watch dog/heartbeat监控两个容器应用的状态
6. 主机宕了怎么办?
7. Watch dog死了怎么办?
8. ...

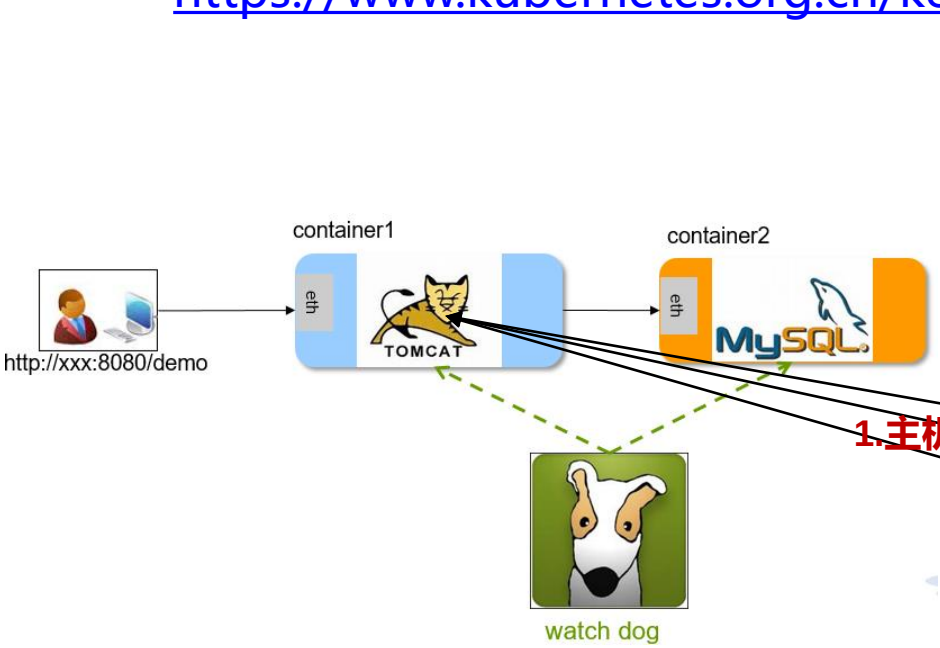
有没有简单可靠的方式?

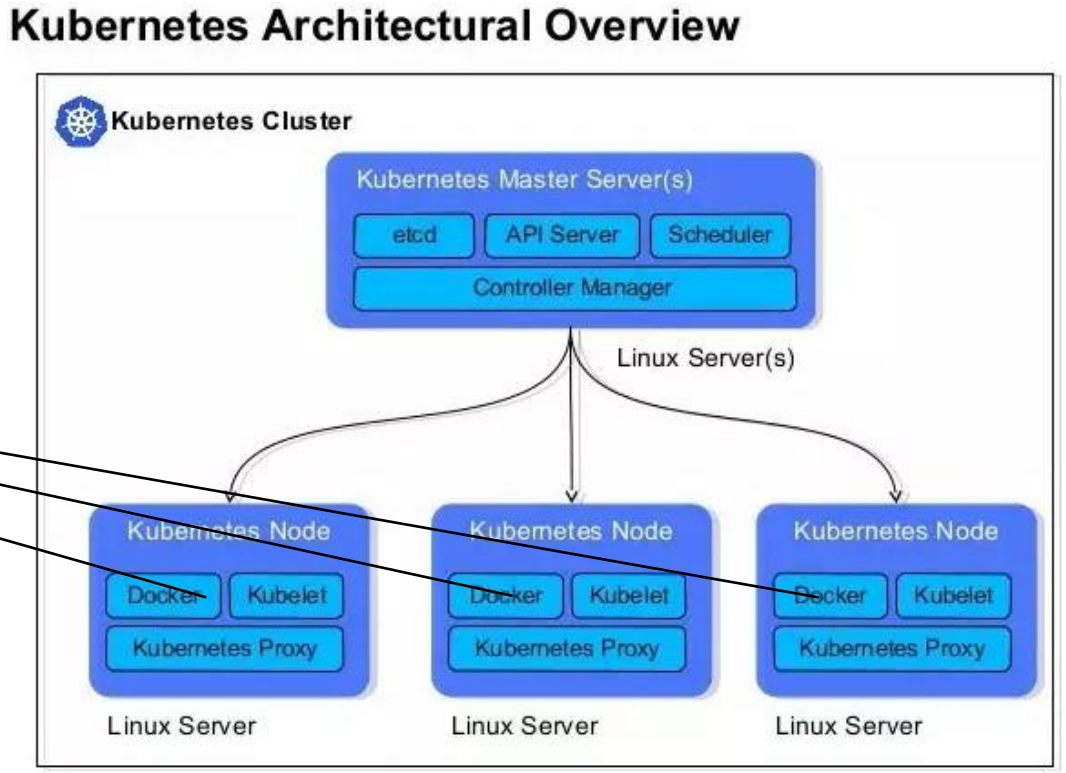
Kubernetes是可以基于Docker容器部署高可靠应用系统的容器管理平台

什么是Kubernetes?

- **Kubernetes（简称k8s）**是一个开源的，用于管理多个主机上的容器化的应用，Kubernetes的目标是让部署容器化的应用简单并且高效，k8s提供了应用部署，规划，更新，维护的一种机制。
- k8s始于2014年，源自于google内部的集群管理系统Borg。

https://www.kubernetes.org.cn/k8s





Kubernetes可以在鲲鹏云服务上部署基于Docker容器的高可靠应用系统

如何在鲲鹏云服务器上制作Docker镜像？

- 三种方式：

方式	描述	优点	缺点
镜像仓库拉取	采用docker pull命令在镜像仓库拉取所需镜像	简单实用	不够灵活，且不一定存在匹配版本
镜像打包	基于基础镜像运行容器并安装所需应用，打包成新镜像	可以获取所需版本的软件Docker镜像	这种方式由于应用已经绑定，无法更改，灵活性差，例如修改内容需要重新制作镜像
Dockerfile自动构建	基于Dockerfile规则编写镜像制作脚本，自动生成镜像	基于脚本自动生成，更新镜像只需修改脚本即可	需要学习和了解Dockerfile的编写规则

Docker镜像在鲲鹏与X86平台上的运行差异?

X86平台构建的镜像是否能直接在鲲鹏平台中运行?

程序代码（C/C++）：

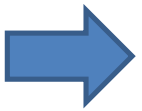
```
int main()
{
        int a = 1;
        int b = 2;
        int c = 0;
        c = a + b;
        return c;
}
```

鲲鹏处理器指令

指令	汇编代码	说明
b9400fe1	ldr x1, [sp,#12]	从内存将变量a的值放入寄存器x1
b9400be0	ldr x0, [sp,#8]	从内存将变量b的值放入寄存器x0
0b000020	add x0, x1, x0	将x1(a)中的值加上x0(b)的值放入x0寄存器
b90007e0	str x0, [sp,#4]	将x0寄存器的值存入内存（变量c）

x86处理器指令

指令	汇编代码	说明
8b 55 fc	mov -0x4(%rbp),%edx	从内存将变量a的值放入寄存器edx
8b 45 f8	mov -0x8(%rbp),%eax	从内存将变量b的值放入寄存器eax
01 d0	add %edx,%eax	将edx(a)中的值加上eax(b)的值放入eax寄存器
89 45 f4	mov %eax,-0xc(%rbp)	将eax寄存器的值存入内存（变量c）



```
dr-xr-xr-x  2 root root 12288 Oct  1  2019 bin
drwxr-xr-x  2 root root  4096 Apr 11  2018 etc
drwxr-xr-x  2 root root  4096 Apr 11  2018 games
drwxr-xr-x  3 root root  4096 Oct  1  2019 include
dr-xr-xr-x 19 root root  4096 Oct  1  2019 lib
dr-xr-xr-x 24 root root 12288 Oct  1  2019 lib64
drwxr-xr-x 11 root root  4096 Oct  1  2019 libexec
drwxr-xr-x 12 root root  4096 Oct  1  2019 local
dr-xr-xr-x  2 root root  4096 Oct  1  2019 sbin
drwxr-xr-x 53 root root  4096 Oct  1  2019 share
drwxr-xr-x  4 root root  4096 Oct  1  2019 src
lrwxrwxrwx  1 root root    10 Oct  1  2019 tmp -> ../var/tmp
[root@ebc53374cbb7 usr]#
```

Docker内部，会有很多依赖库和可执行的二进制文件

鲲鹏与X86指令差异：

1、指令长度（鲲鹏等长，指令流水并发度较X86高、X86不等长，擅长串行）

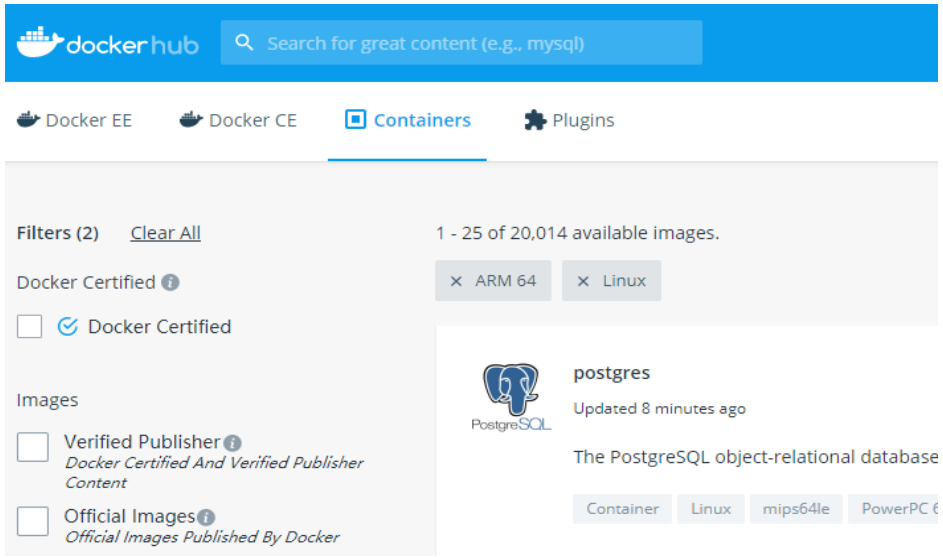
鲲鹏在高性能计算等高并发场景下性价比提升30%

2、汇编代码（X86与鲲鹏汇编指令不一致， X86程序无法直接移植到鲲鹏，必须经过重新编译才能运行）

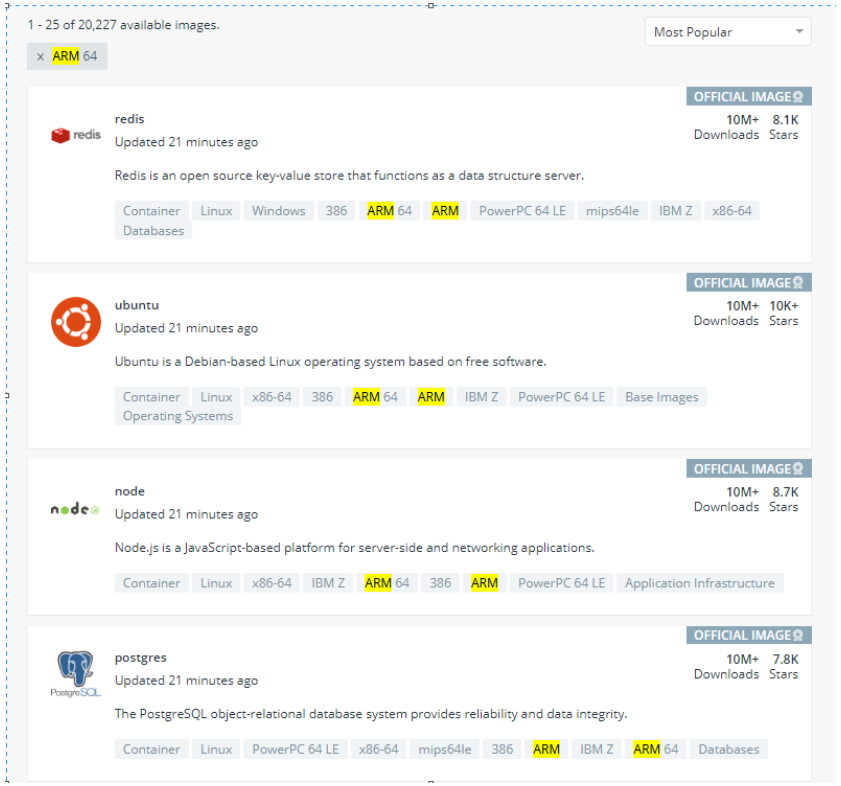
X86平台中三种方式构建的Docker镜像均与平台相关，在鲲鹏平台运行会出现格式不支持等错误。

Docker在鲲鹏上的镜像仓库

1、Docker 官方维护了一个公共仓库 Docker Hub，大部分镜像都可以通过在 Docker Hub 中直接下载。
2、通过docker pull命令可以直接在镜像仓库下载所需镜像



https://hub.docker.com/search?q=&type=image



Docker Hub提供了几乎所有常见应用的镜像

HUAWEI

基于鲲鹏弹性云服务器构建Tomcat镜像

方式一

镜像仓库拉取

- docker pull arm64v8/tomcat
- 运行容器
- 验证容器

方式二

镜像打包

- docker pull arm64v8/centos:7
- docker run交互式启动容器，在容器中安装Tomcat依赖和部署Tomcat
- 创建一个新的镜像作为Tomcat
- 运行容器
- 验证容器

方式三

Dockerfile自动构建

- 编写Dockfile
- 通过Dockerfile构建tomcat镜像
- 运行容器
- 验证容器