1. docker能干什么：面向开发、测试、运维、自动化的技术
   1. 可以简化配置
   2. 提高开发环境，不需要新进开发部署开发环境
   3. 进行应用隔离
   4. 服务器的整合，与虚拟机类似
   5. 快速部署，依次部署，到处运行
   6. 可以进行开发调试
   7. 可以进行自动化扩容，docker更适合微服务（微型服务，分布式服务），不适合跑数据库哦
2. 安装docker服务：yum install docker.io –y，就是这么简单
3. 搜索镜像：docker search centos
4. 使用镜像：docker pull centos，直接pull下来即可
5. 查看系统存在的镜像：docker images
6. 删除镜像：docker rmi uuid
7. 启动镜像容器：docker run centos /bin/echo “hello world”
8. 查看正在运行的docker容器：docker ps -a
9. docker run –name –d(在后台运行) mydocker –it(启动一个伪终端) centos /bin/bash，这个应用一旦这个程序退出，docker容器就退出了
10. 启动一个镜像：docker start uuid
11. 停止一个docker容器：docker stop uuid
12. 帮助：docker –help
13. 推荐进入docker容器的命令：首先装一下util-linux ，根据容器的pid进入到容器

获取容器的uuid：docker ps –l —》获取pid ：docker start uuid—》docker inspect –format “{{.state.pid}} mydocker(镜像名称)” —》nsenter –target pid (前面获取到的pid)–mount –uts –ipc –net –pid

1. 后台启动一个docker镜像：docker -d –name alias contains-name
2. 随机映射端口：docker -d –P –name alias contains-name 容器内的端口号随机映射到宿主机上的端口
3. 停止一个运行的docker镜像：docker stop [pid|contains-name]
4. 指定映射端口：docker –d –p 91:80 –name mynginx1 nginx，如果容器有多个端口需要映射的话，则可以使用多个-p参数
5. Docker数据管理：有两种方式，一种是数据卷，一种是数据卷容器，使用数据卷使用-v参数，如docker run –it –name volume-test1 –h nginx –v /data nginx 实际上就是将虚拟机中的数据映射到物理主机上，与虚拟机机进行数据共享的一个作用，同时可以指定目录进行映射：-v /data:/data，注意，目录后边不能带斜杠；数据卷容器的使用：启动一个容器专门用来存储数据。使用参数 –volumes-from参数，如：docker run –it –name volume-test –volumes-from volume-test1 centos
6. 进入docker容器的方法：首先获取镜像的pid：docker inspect –f {{State.Pid}} container id ,然后使用 nsenter –target pid –mount –uts –ipc –net –pid，就可以进入容器内部
7. Docker镜像的构建：
   1. 手动构建：首先run一个centos，进入之后安装Nginx的依赖包的，提交制作的镜像：docker commit –m “v1 ” container syge/my-nginx:v1(版本号)

21、创建Dockerfile文件

22、构建命令: docker build -t nginx-file:v1 /opt/docker-file/nginx

23、nginx镜像构建实例：

#This is Ny first Dockerfile

#Version 1.0

#Author:syge

#Base images

From centos

#MAINTAINER

MAINTAINER ysge

ADD pcre-8.37.tar.gz /usr/local/src

ADD nginx.tar.gz /usr/local/src

RUN yum install -y wget gcc-c++ make openssl openssl-devel pcre pcre-devel

RUN useradd -s /sbin/nologin -M www

#WORKDIR

WORKDIR /usr/local/src/nginx

RUN ./configure --prefix=/usr/local/nginx --user=www --group=wwww && make && make install

RUN echo "deamon off;" >>/usr/local/nginx/conf/nginx.conf

ENV PATH /usr/local/nginx/sbin:$PATH

EXPOSE 80

CMD ["nginx"]

24、docker核心原理：

docker的资源隔离：每个docker容器之间的资源都是相互隔离的，-》LXC Kernel namespace ,如pid、net、ipc、mount、uts、user都是通过资源空间相互隔离的

资源限制：通过cgroup完成的，安卓也是通过cgroup来完成的，cpu内存限制：通过加参数来完成，通过-c参数来完成，每个docker容器默认有1024个配额,-c指定份额，如-c 512

如：docker run -d --cpu=1 centos

内存容器的限制：使用-m参数，如：-m 128m，如果内存超出，则容器自动退出（新版本）

25、docker网络模式：docker容器默认的网络模式为桥接模式

26、启动docker仓库：docker run –d –p 5001:5000 registry

27、打包镜像：docker tag container\_name 192.168.199.220:5001/test/es:v1

28、推送：docker push 192.168.199.220:5001/test/es:v1、

29、docker的web端管理：部署shipyard应用，修改/etc/sysconfig/docker文件

添加：other\_args=”-H tcp://0.0.0.0:235 -H unix:///var/run/docker.sock”

Docker再次加强

1. docker的基本组成
   1. docker client客户端
   2. docker daemon守护进程
   3. docker image 镜像（层叠的只读文件系统）
   4. docker container 容器
   5. docker registry 仓库
2. docker依赖的Linux内核特性
   1. namespace命名空间 封装—>代码隔离，包括进程隔离、net隔离、ipc隔离、mnt隔离
   2. control groups控制组 资源限制、优先级设定、资源计量、资源控制
3. docker容器的能力：文件隔离、进程隔离、网络隔离、资源隔离和分组
4. 容器的基本操作：
   1. 启动容器，docker run image [command]
      1. 启动交互式容器：docker run –I –t image /bin/bash
   2. 查看容器，docker ps –l –a
   3. 查看容器的信息，docker inspect image
   4. 自定义容器名称，run命令时，--name选项可以自定义容器的名称
   5. 重新启动和停止容器，docker start –I containerId
5. 守护式容器
   1. Docker run –I –t centos /bin/bash 然后以ctrl+Q和ctrl+P组合键退出
   2. 启动时加上-d参数，然后启动一个脚本，循环不退出
   3. 查看容器日志 docker logs –f -t --tail 容器名
   4. Docker top 容器名称
6. 在运行中的容器内启动新的进程
   1. Docker exec –d –I –t 容器名 command arg
7. docker 容器的端口映射
   1. –P:将所有容器中的暴露的端口映射出来
   2. –p:指定端口的映射
8. docker镜像仓库 对应一组镜像的repository，与registory不同
9. 查看镜像的详细信息 docker inspect,既支持镜像的查看，也支持容器的查看
10. Docker获取和推送镜像
    1. 查找镜像：docker search image\_name
    2. 下载镜像：docker pull，如果只指定仓库名，则默认tag为last
    3. 时候daocloud，获取加速链接，使用这个链接，在docker的启动文件中添加这个链接
    4. 推送docker镜像，docker push container\_name images\_name
11. 构建docker镜像 docker commit containerId –a（作者名）-m（镜像的ID）
    1. 使用dockerfile文件构建镜像 docker build –t image\_name