**中软国际考题-笔试**

1. 单选题

1、以下哪个不是--net选项后面可以跟的参数？（D）

A. None B. host C. bridge D. NAT

2、如何临时退出一个正在交互的容器的终端，而不终止它（A）

A．按Ctrl+p，后按Ctrl+q。

B．按Ctrl+p，后按Ctrl+c。

C．按Ctrl+c，后按Ctrl+q。

D．按Ctrl+d，后按Ctrl+q。

3、查看镜像支持的环境变量采用（ ）命令 ( B )

A. sudo docker run Image path

B. sudo docker run Image env

C. sudo docker run Image variable

D. sudo docker run Image build

4、Ubuntu系统下Docker的配置文件放在那里（A）

A./etc/default/docker B./etc/user/docker C./etc/sysconfig/docker D./etc/sys/docker

5、下列哪个是在容器内运行一个应用程序的命令？（A）

A． docker run ubuntu:15.10 /bin/echo "Hello world" Hello world

B． run docker ubuntu:15.10 /bin/echo "Hello world" Hello world

C． run docker ubuntu:15.10 /bin/echo "Hello world"

D． docker run ubuntu:15.10 /bin/echo Hello world

6、在服务器虚拟化领域,出现了新的虚拟化容器,它是（A）

A.LXC,Docker B.JVM C.VDI D.INTEL-VT

7、下列选项不属于Docker概念是（B）

A.镜像 B.分布式 C.容器 D.仓库

8、docker run命令的哪个参数是在新容器内指定一个伪终端或终端（B）

A. -i B. -t

C.-e D. -f

1. 多选题
2. 以下哪些是Cgroup的功能？（A、B、C、D）

A.资源限制    B.优先级分配   C.资源统计   D.任务控制

2、构建Docker镜像应该遵循哪些原则？（A、B、C、D、E）

A．尽量选取满足需求但较小的基础系统镜像，建议选择debian:wheezy镜像，仅有86MB大小

B．清理编译生成文件、安装包的缓存等临时文件

C．安装各个软件时候要指定准确的版本号，并避免引入不需要的依赖

D．从安全的角度考虑，应用尽量使用系统的库和依赖

E．使用Dockerfile创建镜像时候要添加.dockerignore文件或使用干净的工作目录

3、一个完整的Docker由以下（ ）部分组成？（A、B、C、E）

A. DockerClient客户端

B. Docker Daemon守护进程

C. Docker Image镜像

D. DockerContainer管理工具

E. DockerContainer容器

4、docker的存储驱动程序有哪些？（A、B、C、D）

A. AUES

B. Btrfs

C. Device mapper

D. OverlayFS

5、下列哪些关于Docker的描述是正确的？（ABCDE）

1. 速度快以及优雅的隔离框架
2. 物美价廉
3. CPU/内存的低消耗
4. 快速开/关机
5. 跨云计算基础构架

6、Docker的存储驱动程序有哪些？（A、B、C、E）

A. AUES B. Btrfs C. Device mapper

D. VFZ; E. ZFS

7、下面哪些项有助于Docker构建与环境无关的系统？（A、B、C）

A. Volumes B. 环境变量注入

C. 只读文件系统 D. Docker镜像

8、以下哪种系统是属于Linux的发行版？（A、B）

A.ubuntu B. centos C. macos D.ios

1. 填空题（每空1分，共10分）
2. 使用docker build命令有\_\_\_\_\_\_\_\_\_\_ 、\_\_\_\_\_\_\_\_\_两种方法构建镜像。(通过源代码路径的方式 、 通过标准输入流的方式)
3. 查看容器的进程的命令是\_\_\_\_\_\_\_\_\_\_（Docker  top   容器id）
4. 使用\_\_\_\_\_\_\_\_\_\_命令下载centos 镜像（Docker  pull  centos）
5. 打开容器test2去链接test1容器，并映射宿主机的80端口到容器的5000端口的命令是\_\_\_\_\_\_\_\_\_\_（sudo Docker run -it -v 80:5000 -name test2 --network=container test1 centos /bin/bash）
6. 如何批量删除或者停止运行的容器：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。(docker kill/rm `docker ps -aq`)
7. Docker启动Container的起始点在daemon/container.go/Start方法中，该方法主要做了\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_、\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_、\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_的事情。(配置启动Container所需要的环境、启动Container进程、如果失败，进行资源回收)
8. Docker的核心组件有\_\_\_\_\_\_\_，\_\_\_\_\_\_\_和\_\_\_\_\_\_\_。(镜像，容器，仓库)
9. Docker有三个组件和三个基本元素。三个组件分别是：（ ）是用户界面，它支持用户与Docker Daemon之间通信。（ ）运行于主机上，处理服务请求。（ ）是中央registry，支持拥有公有与私有访问权限的Docker容器镜像的备份。（Docker Client ，Docker Daemon，Docker Index）
10. Docker核心组件有哪些：\_\_\_\_\_\_\_\_\_\_ 、\_\_\_\_\_\_\_\_\_ 、\_\_\_\_\_\_\_\_\_。(镜像,容器,仓库)
11. Docker容器的四种状态分别是：\_\_\_\_\_\_\_\_\_\_ 、\_\_\_\_\_\_\_\_\_、\_\_\_\_\_\_\_\_\_\_ 、\_\_\_\_\_\_\_\_\_。(运行、已暂停、重新启动、已退出)
12. Dockerfile中最常见的指令包含：\_\_\_\_\_\_\_\_\_\_ 、\_\_\_\_\_\_\_\_\_、\_\_\_\_\_\_\_\_\_\_ 、\_\_\_\_\_\_\_\_\_。(FROM、LABEL、RUN、CMD)
13. docker run命令允许对容器内的标准输入 (STDIN) 进行交互的参数是\_\_\_\_\_。(-i)
14. docker \_\_\_\_\_\_\_命令可以通过容器ID来查看对应的容器发生了什么。（ps）
15. docker \_\_\_\_\_\_命令，查看容器内的标准输出。（logs）
16. 使用 docker \_\_\_\_\_ 命令来停止容器。（stop）
17. 使用 docker \_\_\_\_\_ 命令从镜像仓库中拉取或者更新指定镜像。（pull）
18. 使用 docker \_\_\_\_\_ 命令将本地的镜像上传到镜像仓库,要先登陆到镜像仓库（push）
19. 使用 docker \_\_\_\_\_ 命令列出本地镜像（images）
20. 使用 docker \_\_\_\_\_ 命令删除本地一个或多个镜像（rmi）
21. 使用 docker \_\_\_\_\_ 命令从容器创建一个新的镜像（commit）
22. 使用 docker \_\_\_\_\_ 命令将指定镜像保存成 tar 归档文件对应的导入命令是docker \_\_\_\_\_（save、 load）
23. 判断题
24. Docker可以运行在32位和64位的Linux操作系统上（ F ）
25. 容器退出后，通过docker ps 命令查看不到，数据会丢失（ F ）
26. Docker利用[Linux核心](https://zh.wikipedia.org/wiki/Linux%E6%A0%B8%E5%BF%83)中的资源分离机制，例如[cgroups](https://zh.wikipedia.org/wiki/Cgroups)，以及Linux核心[名字空间](https://zh.wikipedia.org/w/index.php?title=Linux%E5%91%BD%E5%90%8D%E7%A9%BA%E9%96%93&action=edit&redlink=1)（namespaces），来创建独立的[容器](https://zh.wikipedia.org/wiki/%E4%BD%9C%E6%A5%AD%E7%B3%BB%E7%B5%B1%E5%B1%A4%E8%99%9B%E6%93%AC%E5%8C%96)（containers）。（ T ）
27. Dockers是没有办法打包应用程序及其虚拟容器。（ F ）
28. **在CentOS中，Docker的配置文件放在**/etc/sysconfig/docker/conf**。 （** F **）**
29. Docker提倡“一次封装，到处运行”。（ T ）
30. Kubernetes(k8s)是Apache开源的容器集群管理系统。（ F ）
31. Docker不能在非Linux平台上运行。（ F ）
32. 将Docker环境迁移到另外一台宿主主机上需要先停止Docker服务。（ T ）
33. Docker的镜像可以跨多个服务器进行部署。（ T ）
34. Docker能够确保每个容器都拥有自己的资源，并且和其他容器是隔离的。（ T ）
35. Docker将宿主机操作系统上的敏感挂载点（例如/proc和/sys）作为只读挂载点，并且使用一种[写时复制](http://zh.wikipedia.org/wiki/%E5%AF%AB%E5%85%A5%E6%99%82%E8%A4%87%E8%A3%BD)系统去改变其他容器的数据。（ F ）
36. Docker可以在32位的系统里运行。（ F ）
37. 容器退出后，通过docker ps 命令查看不到，数据不会丢失。（ T ）
38. Docker Containers负责应用程序的运行，包括操作系统、用户添加的文件以及元数据。（ T ）
39. Docker Images是一个只读模板，用来运行Docker容器。（ F ）
40. DockerFile是文件指令集，用来说明如何自动创建Docker镜像（T）
41. Docker能不能在32位的系统里运行（ F ）
42. 可以通过"docker ps –a"运行命令来识别Docker容器的状态。（ T ）
43. 将文件系统作为一个tar归档文件导出到STDOUT，对应的导入命令是docker import（T）
44. Docker容器完全使用沙箱机制，相互之间不会有任何接口（T）

五、简答题

1、Docker的核心组件有哪些？

答案：镜像、容器、仓库

2、docker里有三种方式把数据从宿主机挂载到容器，分别是volmues,blnd mounts,tmpfs volumes，它们有什么区别？

答案：

Volumes方式下：容器内的数据被存放到宿主机（Linux）一个特定的目录下(/var/lib/docker/volumes/)，这个目录只有Docker可以管理，其他进程不能修改。如果想持久保存容器的应用数据，Volumes是Docker推荐的挂载方式。

Bind mounts方式下：容器内的数据被存放到宿主机文件系统的任意位置，甚至存放到一些重要的系统目录或文件中。除了Docker之外的进程也可以任意对他们进行修改。

tmpfs方式下：容器的数据只会存放到宿主机的内存中，不会被写到宿主机的文件系统中，因此不能持久保存容器的应用数据。

3、Orchestration的分类有哪些，其中compose使用的步骤是什么？

答案：Docker Compose, Docker Machine, Docker Swarm

使用Compose仅需要三步·

（1）、使用dockerfile定义你的应用依赖的镜像；

（2）、使用docker-compose.yml定义你的应用具有的服务；

（3）、通过docker-compose up命令创建并运行应用。

4、Docker与LXC（Linux Container）有何不同？（5分）

答案：LXC利用Linux上相关技术实现容器，Docker则在如下的几个方面进行了改进：

移植性：通过抽象容器配置，容器可以实现一个平台移植到另一个平台；

镜像系统：基于AUFS的镜像系统为容器的分发带来了很多的便利，同时共同的镜像层只需要存储一份，实现高效率的存储；

版本管理：类似于GIT的版本管理理念，用户可以更方面的创建、管理镜像文件；

仓库系统：仓库系统大大降低了镜像的分发和管理的成本；

周边工具：各种现有的工具（配置管理、云平台）对Docker的支持，以及基于Docker的Pass、CI等系统，让Docker的应用更加方便和多样化。。

5、**构建Docker镜像应该遵循哪些原则**

答案：

整体原则上，尽量保持镜像功能的明确和内容的精简，要点包括：

* 选取满足需求但较小的基础系统镜像
* 清理编译生成文件、安装包的缓存等临时文件
* 安装各个软件时候要指定准确的版本号，并避免引入不需要的依赖
* 从安全的角度考虑，应用尽量使用系统的库和依赖
* 使用Dockerfile创建镜像时候要添加.dockerignore文件或使用干净的工作目录

6、Docker容器创建后，删除了/var/run/netns 目录下的网络名字空间文件，如何恢复它？

答案：查看容器进程ID，比如1234

sudo docker inspect --format='{{. State.pid}}' $container\_id

1234

到proc目录下，把对应的网络名字空间文件链接到/var/run/netns,然后通过正常的系统命令查看操作容器的名字空间。

7、简述Docker使用了哪些操作系统的功能来提高容器技术效率

答：

Namespaces 充当隔离的第一级。确保一个容器中运行一个进程而且不能看到或影响容器外的其它进程。

Control Groups是LXC的重要组成部分，具有资源核算与限制的关键功能。

UnionFS（文件系统）作为容器的构建块。为了支持Docker的轻量级以及速度快的特性，它创建了用户层。

8、什么是容器

容器是一种轻量级,可移植,自包含的软件打包技术,使用程序可以在任何地方以相同的方式运行

9、Docker与虚拟机有何不同？（5分）

答案：

Docker不是虚拟化方法。它依赖于实际实现基于容器的虚拟化或操作系统级虚拟化的其他工具。为此，Docker最初使用LXC驱动程序，然后移动到libcontainer现在重命名为runc。Docker主要专注于在应用程序容器内自动部署应用程序。应用程序容器旨在打包和运行单个服务，而系统容器则设计为运行多个进程，如虚拟机。因此，Docker被视为容器化系统上的容器管理或应用程序部署工具。

1）与虚拟机不同，容器不需要引导操作系统内核，因此可以在不到一秒的时间内创建容器。此功能使基于容器的虚拟化比其他虚拟化方法更加独特和可取。

2）由于基于容器的虚拟化为主机增加了很少或没有开销，因此基于容器的虚拟化具有接近本机的性能

3）对于基于容器的虚拟化，与其他虚拟化不同，不需要其他软件。

4）主机上的所有容器共享主机的调度程序，从而节省了额外资源的需求。

5）与虚拟机映像相比，容器状态（Docker或LXC映像）的大小很小，因此容器映像很容易分发。

6）容器中的资源管理是通过cgroup实现的。Cgroups不允许容器消耗比分配给它们更多的资源。虽然主机的所有资源都在虚拟机中可见，但无法使用。这可以通过在容器和主机上同时运行top或htop来实现。所有环境的输出看起来都很相似。

10、容器内部运行机制？（5分）

答案：

每个容器都在自己的命名空间中运行，但使用与所有其他容器完全相同的内核。发生隔离是因为内核知道分配给进程的命名空间，并且在API调用期间确保进程只能访问其自己的命名空间中的资源。

11、如何理解Docker？（5分）

答案：

1）Docker是一个容器化平台，它以容器的形式将您的应用程序及其所有依赖项打包在一起，以确保您的应用程序在开发，测试或生产的任何环境中无缝运行。

2）Docker容器，将一个软件包装在一个完整的文件系统中，该文件系统包含运行所需的一切：代码，运行时，系统工具，系统库等可以安装在服务器上的任何东西。

3）这可以保证软件始终运行相同，无论其环境如何。

12、docker save和docker export的区别？（5分）

答案：

docker save保存的是镜像（image），docker export保存的是容器（container）；

docker load用来载入镜像包，docker import用来载入容器包，但两者都会恢复为镜像；

docker load不能对载入的镜像重命名，而docker import可以为镜像指定新名称

13、Kubernetes是什么？

Kubernetes，因为首尾字母中间有8个字符，所以被简写成 K8s

K8s 是底层资源与容器间的一个抽象层，如果和单机架构类比，可以算作是一个分布式时代的 Linux。

K8s 是 Google 开源的容器集群管理系统。在 Docker 技术的基础上，为容器化的应用提供部署运行、资源调度、服务发现和动态伸缩等一系列完整功能，提高了大规模容器集群管理的便捷性。

六、程序实现题

1、编写dockerfile,实现功能：开启容器就能查看/目录，并且能复写查看/mnt目录。镜像自选

|  |
| --- |
| #vim dockerfile  FROM centos  RUN Is /  CMD ['ls","/mnt"] |

2、编写使用Maven插件编译Docker镜像的主要Pom文件配置及构建命令

|  |
| --- |
| <plugin>  <groupId>com.spotify</groupId>  <artifactId>docker-maven-plugin</artifactId>  <version>0.4.13</version>  <configuration>  <imageName>itmuch/microservice-discovery-eureka:0.0.1</imageName>  <baseImage>java</baseImage>  <entryPoint>["java", "-jar", "/${project.build.finalName}.jar"]</entryPoint>  <resources>  <resource>  <targetPath>/</targetPath>  <directory>${project.build.directory}</directory>  <include>${project.build.finalName}.jar</include>  </resource>  </resources>  </configuration>  </plugin>  mvn clean  mvn package docker:build |

3、请使用Docker安装Tomcat。根据要求，写出命令，完成如下配置和部署操作。（10分）

|  |
| --- |
| $ docker search tomcat  NAME DESCRIPTION STARS OFFICIAL AUTOMATED  tomcat Apache Tomcat is an open source implementa... 744 [OK]  #拉取tomcat镜像  $ docker pull tomcat  #查看库中tomcat镜像  $ docker images|grep tomcat  #启动该tomcat镜像，使用9888端口映射容器8080端口  $ docker run -p 9888:8080 tomcat  #创建/home/root/webapp文件夹，用于映射tomcat容器中的webapp  $mkdiar -p /home/root/webapp  #查看正在运行的docker容器  $ docker ps  #将home/root/webapp挂载到容器的webapp中，启动容器，映射端口还是9888  $ docker run -d -v /home/root/webapps:/usr/local/tomcat/webapps/ -p 9888:8080 tomcat  # 停止容器，id为54215923125b  $ docker stop 54215923125b |

4、如何批量删除或者停止运行的容器

如何查看镜像支持的环境变量

如何停止所有正在运行的容器

如何清理批量后台停止的容器

|  |
| --- |
| docker kill/rm `docker ps -aq`  sudo docker run IMAGE env  docker kill $(sudo docker ps -q)  docker rm $（sudo docker ps -a -q） |

5、用代码、命令演示Docker集成MySQL?

|  |
| --- |
| 创建Dockerfile  首先，创建目录mysql,用于存放后面的相关东西。    data目录将映射为mysql容器配置的数据文件存放路径  logs目录将映射为mysql容器的日志目录  conf目录里的配置文件将映射为mysql容器的配置文件  进入创建的mysql目录，创建Dockerfile              通过Dockerfile创建一个镜像，替换成你自己的名字    创建完成后，我们可以在本地的镜像列表里查找到刚刚创建的镜像 |

6、编写dockerfile,实现功能,开启容器就能查看/目录,并且能复写查看/mnt目录,镜像自选.

（10分）

|  |
| --- |
| #vim dockerfile  FROM centos  RUN ls /  CMD [“ls”,”mnt”] |