docker 常见面试题

一、 Docker

1、什么是 docker？

Docker 是一个容器化平台，它将应用程序及其所有依赖项以容器的形式打包在一起，以确保应用程序在任何环境（无论是开发环境、测试环境还是生产环境

2、什么是容器？

容器就是在隔离的环境运行的一个进程，如果进程停止，容器就会退出。隔离的环境拥有自己的系统文件，ip 地址，主机名等

Docker 容器，将一个软件包在一个完整的文件系统中，其中包含运行所需的一切：代码、运行时、系统工具、系统库等任何可以安装在服务器上的东西。它都将始终运行相同的程序，无论软件的环境如何。

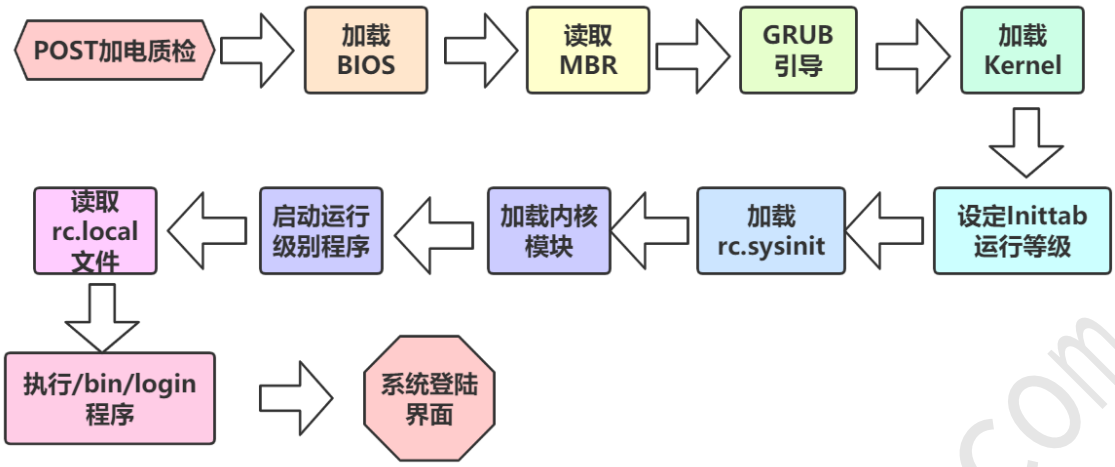
3、容器与传统虚拟化的区别？

Docker 以及其他容器技术，都属于操作系统虚拟化范畴，操作系统细腻化最大的特点就是不需要额外的 supervisor 支持。Docker 虚拟化方式之所以有众多优势，跟操作系统虚拟化技术自身的设计和实现分不开。

传统方式是在硬件层面实现虚拟化，需要有额外的虚拟机管理应用和虚拟机操作系统层。Docker 容器时在操作系统层面实现虚拟化，直接复用本地主机的操作系统，因此更加轻量级。

1)虚拟机

虚拟机：硬件 cpu 支持(vt 虚拟化),模拟计算硬件,走正常的开机启动



2)容器

2、容器：不需要走开机启动流程，不需要硬件 cpu 的支持,共用宿主机内核去启动容器的第一个进程。

3、容器优势：启动快,性能高,损耗少,轻量级

100 个虚拟机运行 100 个服务需要 10 台物理机

100 个容器运行 100 个服务需要大约 6 台物理机

特性	Docker	VM
启动速度	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
隔离性	安全隔离	完全隔离

4、Docker 的优势

作为一种轻量级的虚拟化方式，Docker 在运行应用上跟传统的虚拟机的方式相比具有如下显著优势：

1、Docker 容器启动很快，启动和停止可以实现秒级，相比传统的虚拟机方式（分钟级）要快速很多。

2、Docker 容器对系统资源需求很少，一台主机上可以同时运行数千个 Docker 容器。

3、Docker 通过类似 git 设计理念的操作来方便用户获取、分发和更新应用镜像，存储复用，增量更新。

4、Docker 通过 Dockerfile 支持灵活的自动化创建和部署机制，可以提高工作效率，并标准化流程。

5、什么是镜像？

Docker 镜像是 Docker 容器的源代码。换句话说，Docker 镜像用于创建容器。使用 build 命令创建镜像，并且在使用 run 启动时它们将生成容器。镜像存储在 Docker 注册表中， registry.hub.docker.com 因为它们可能变得非常大，镜像被设计为由其他镜像层组成，允许在通过网络传输镜像时发送最少量的数据。

6、什么是 docker 容器？

1、Docker 容器包括应用程序及其所有依赖项，但与其他容器共享内核，在主机操作系统的用户空间中作为独立进程运行。

2、Docker 容器不依赖于任何特定的基础架构：它们可以在任何计算机，任何基础架构和任何云中运行。

7、什么是镜像仓库？

用来保存镜像的仓库。当我们构建好自己的镜像之后，需要存放在仓库中，当我们需要启动一个镜像时，可以在仓库中下载下来。

8、什么是 docker hub？

Docker hub 是一个基于云的注册表服务，允许您链接到代码存储库，构建映像并测试它们，存储手动推送的镜像以及指向 Docker 云的链接，以便您可以将镜像部署到主机。它为整个开发流程中的容器发现，分发和变更管理，用户和团队协作以及工作流自动化提供了集中资源。

9、Docker 容器的四种状态？

在任何给定的时间点，Docker 容器都可以有四种状态。如下：

运行

已暂停

重新启动

已退出

10、如何识别 Docker 容器的状态？

我们可以通过运行命令来识别 Docker 容器的状态

```
docker ps -a
```

这将依次列出所有可用的 docker 容器及其在主机上的相应状态。我们可以很容易地识别感兴趣的容器，以相应地检查其状态。

11、说出你常用的 dockerfile 的指令，并说出作用？

Dockerfile 中的一些常见指令如下：

FROM：我们使用 FROM 为后续指令设置基本镜像。在每个有效的 Dockerfile 中，FROM 是第一条指令。

LABEL：我们使用 LABEL 根据项目，模块，许可等组织我们的镜像。我们也可以使用 LABEL 来帮助实现自动化。在 LABEL 中，我们指定一个键值对，以后可用于以编程方式处理 Dockerfile。

RUN：我们使用 RUN 命令在当前图像之上的新图层中执行任何指令。使用每个 RUN 命令，我们在图像上添加一些内容，并在 Dockerfile 的后续步骤中使用它。

CMD：我们使用 CMD 命令提供执行容器的默认值。在 Dockerfile 中，如果我们包含多个 CMD 命令，则只使用最后一条指令。

12、解释 docker 的使用流程

1. 一切都从 Dockerfile 开始。Dockerfile 是镜像的源代码。

2. 创建 Dockerfile 后，您可以构建它以创建容器的镜像。图像只是“源代码”的“编译版本”，即 Dockerfile。

3. 获得容器的镜像后，应使用注册表重新分发容器。注册表就像一个 git 存储库 - 你可以推送和拉取镜像。

4. 接下来，您可以使用该图像来运行容器。在许多方面，正在运行的容器与虚拟机（但没有虚拟机管理程序）非常相似。

13、docker 常用的命令？

docker pull 拉取或者更新指定镜像
docker push 将镜像推送至远程仓库
docker rm 删除容器
docker rmi 删除镜像
docker images 列出所有镜像
docker ps 列出所有容器

14、ADD 和 COPY 的区别？

COPY 与 ADD 的区别 COPY 的 SRC 只能是本地文件，ADD 会自动解压 tar 包

15、解释一下 dockerfile 的 ONBUILD 指令？

当镜像用作另一个镜像构建的基础时，ONBUILD 指令向镜像添加将在稍后执行的触发指令。如果要构建将用作构建其他镜像的基础的镜像（例如，可以使用特定于用户的配置自定义的应用程序构建环境或守护程序），这将非常有用。

16、生产中如何监控 docker？

Docker 提供 docker stats 和 docker events 等工具来监控生产中的 Docker。我们可以使用这些命令获取重要统计数据的报告。

Docker stats：当我们使用容器 ID 调用 docker stats 时，我们获得容器的 CPU，内存使用情况等。它类似于 Linux 中的 top 命令。

Docker events：Docker events 是一个命令，用于查看 Docker 守护程序中正在进行的任务。

17、常见的 Docker 事件？

一些常见的 Docker 事件是：attach，commit，die，detach，rename，destroy 等。我们还可以使用各种选项来限制或过滤我们感兴趣的事件。

18、停止 docker 时会不会丢失数据？

不，当 Dcoker 容器退出时，不会丢失数据。应用程序写入磁盘的所有数据都会保留在其容器中，直到您明确删除该容器为止。即使在容器停止后，该容器的文件系统仍然存在