要在centos下配置docker有两个前提

1. 当前系统可以访问外网
2. 当前系统版本是否为最新版本

由于配置网络环境已经实现了所以这里只要更新系统就可以

执行命令 yum update 可以实现更新系统

执行命令 yum install docker 可以实现安装docker

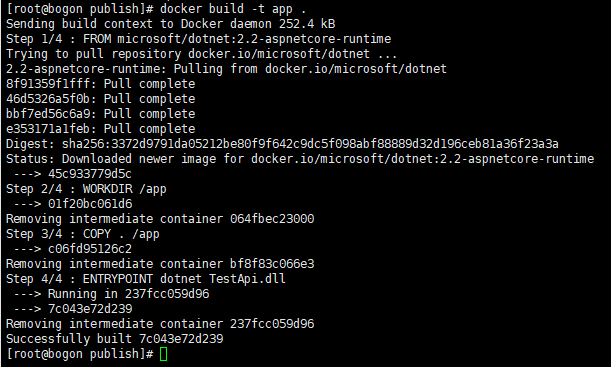
执行命令 systemctl start docker 启动docker

执行命令 systemctl enable docker 确保开机启动docker

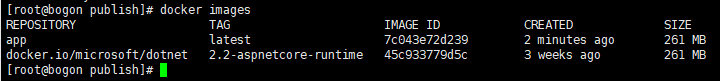
由于执行过程太长了所以不配截图了

但是可以通过发布asp.net core项目至docker 可以验证docker安装是否成功

进入publish文件夹目录下执行命令 docker build -t app . 来创建image如图



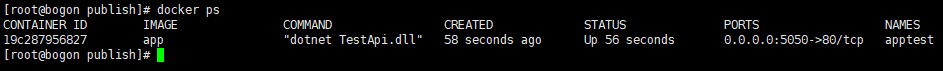
执行命令docker images 查看创建好的image如图



执行命令 docker run –name apptest -d -p 5050:80 app 创建docker container 如图



执行命令 docker ps 查看docker container 发布是否成功 如图



如果执行docker ps 后无任何结果先通过docker ps -a查看全部docker container 无论该容器是否挂载成为守护进程都可以看见 如图



可以通过命令 docker logs apptest 来查看当前container apptest的日志，可以通过日志反馈的信息来查明docker container 发布失败的原因。

注意 5050:80是指将容器的80端口映射至虚拟机的5050也就说宿主机可以通过192.168.136.200:5050来访问该容器。

通过命令curl <http://localhost:5050>，来检测是否可以访问该容器如图

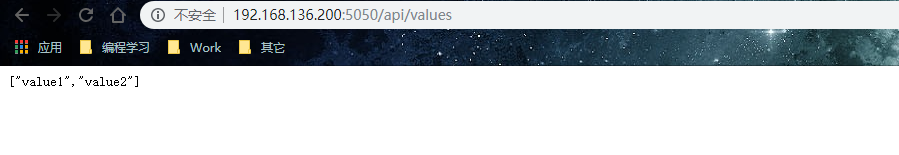


由于发布的项目是web api 所以如果没有配置swagger ui直接访问地址即便访问成功也没有任何结果可以断言访问是否成功，由于创建的web api 项目是默认的项目所以项目中提供了一个web api 的示例路由地址是 /api/values

所以应该执行命令 curl <http://localhost:5050/api/values>如图



通过反馈的结果是[“value1”,”value2”] 发布的容器已经成功挂载在centos系统中了，由于以上过程均是在虚拟机执行的，所以需要确定是否可以通过宿主机访问该容器，由于我们发布的是一个网站所以可以通过宿主机的浏览器访问该容器的地址如图



因此可以确定宿主机是可以访问部署在虚拟机中的docker容器