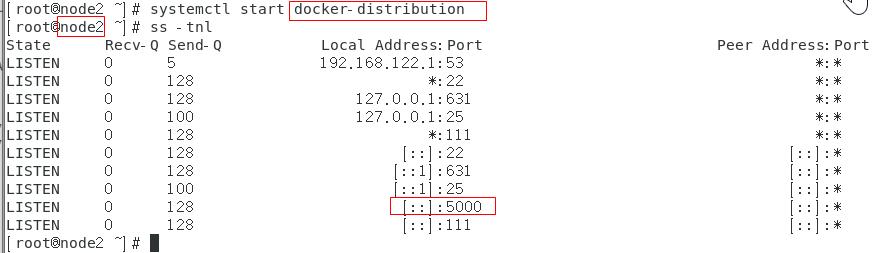
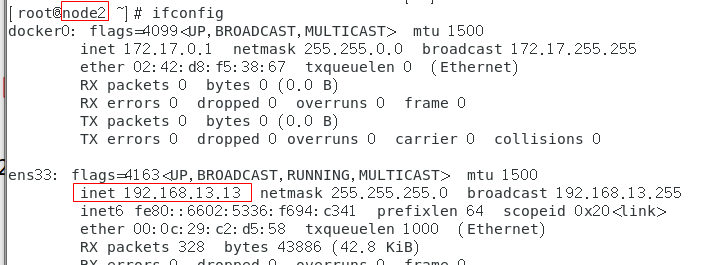


在想要推送镜像的另一台主机(node2)上安装和启动docker-distribution

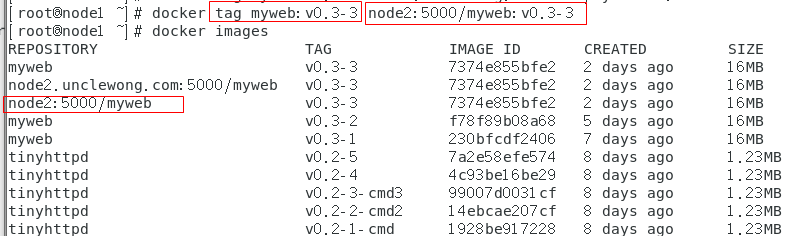


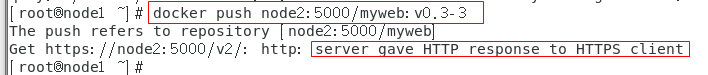
想把镜像推到node2上，先要确认node2对外通信的ip地址+5000端口



非dockerhub情况下必须给定registry服务器的ip地址、端口 甚至用户名

首先给镜像打标：



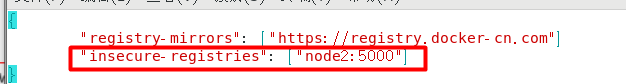


报错：docker push 给客户端自动生成的是https协议， 但服务器发送的是http请求

所以docker registry须要做成HTTPS

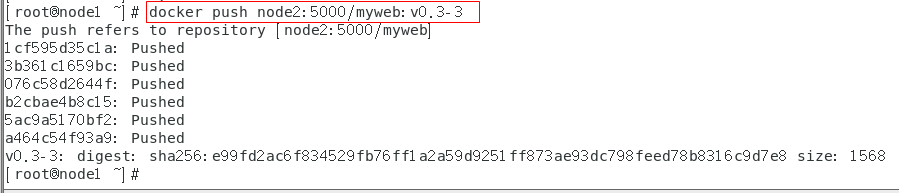
解决方式一：将registry标记为非加密的

vim /etc/docker/daemon.json





这时再push就不再报错

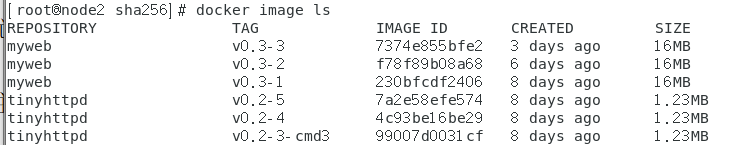


然后进入node2看看有没有收到传来的镜像文件



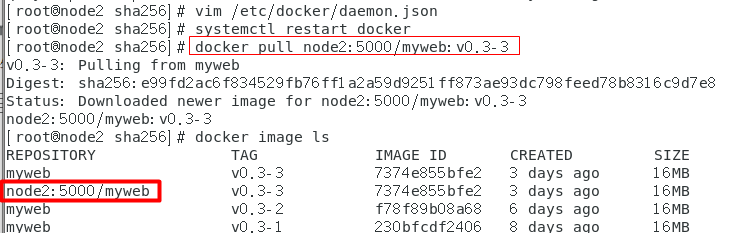
sha256中的每一个文件就代表一个镜像层，sha256中放的是镜像层的链接（link） 实际存储在 v2/blobs中

此时node2的镜像库中还没有拿到镜像，还需要docker pull将镜像拉入仓库



这种方式下 若node2想主动pull镜像的话 也得进行相同的daemon.json设置

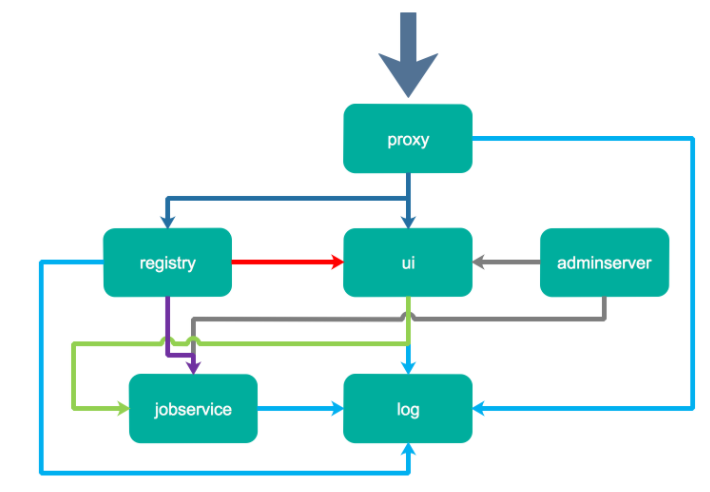
之后才能在node2自己的镜像库里看到从registry拉取的镜像



Harbor

Harbor封装了Docker的registry v2，帮用户提供了许多便捷管理的特性，方便用户操作，提供了管理UI，基于角色的访问控制(Role Based Access Control)，AD/LDAP集成、以及审计日志(Auditlogging) 等企业用户需求的功能，同时还原生支持中文。

harbor的架构:



主要组件包括proxy，他是一个nginx前端代理，主要是分发前端页面ui访问和镜像上传和下载流量，上图通过深蓝色线标志；

ui提供了一个web管理页面，当然还包括了一个前端页面和后端API，底层使用mysql数据库；

registry是镜像仓库，负责存储镜像文件，当镜像上传完毕后通过hook通知ui创建repositry，上图通过红色线标志，当然registry的token认证也是通过ui组件完成；

adminserver是系统的配置管理中心附带检查存储用量，ui个jobservice启动时候需要加载adminserver的配置，通过灰色线标志；

jobservice是负责镜像复制工作的，他和registry通信，从一个registry pull镜像然后push到另一个registry，并记录job\_log，上图通过紫色线标志；

log是日志汇总组件，通过docker的log-dirver把日志汇总到一起，通过蓝色线条标志。