# Specification

* Harbor
  + harbor有一个管理页面，地址是https://10.12.32.91/
  + username: tvu
  + password: P1dkme17tvU
  + 目前repository只有latest标签的镜像（为了方便不用加tag，默认标签）；
  + 现有的repository都是public的，所有人都可以拉取镜像，不需要登录；
  + docker默认使用https来连接registry，可以使用如下方式绕过，修改完配置后需要systemctl restart docker：
    - 新版本docker在/etc/docker/daemon.json中添加"insecure-registries":["0.0.0.0/0"]
    - 目前老版本的是在/etc/sysconfig/docker中添加INSECURE\_REGISTRY='--insecure-registry 0.0.0.0/0'
  + 不管registry是使用http或者https，都可以通过上条方式修改配置绕过去。但是如果使用https协议，则可以不修改配置文件，只需添加一个ca证书就可以解决，所以目前让registry使用的是https协议，证书是自己生成的自签名证书，添加证书操作如下
  + sudo mkdir -p /etc/docker/certs.d/10.12.32.91   
    sudo scp tvu@10.12.32.91:/data/certifications/ca.crt /etc/docker/certs.d/10.12.32.91/
  + push镜像前需要先登录
  + docker login 10.12.32.91
  + username: tvu
  + password: P1dkme17tvU
* 测试镜像
  + library/webr-proxy
  + 该镜像默认把你的请求转发给测试环境的Nginx服务，所以配置的端口是Nginx服务暴露出来的，比如连接测试机器来时。
  + default **RHOST** is **127.0.0.1** *host*
  + default **RPORT** is **8288** *nginx port*
  + default **RWS** is **8288** *nginx port*
  + docker -d [--name <container name>] -p <external port>:80 [-e RHOST=<r host>] [-e RPORT=<api port>] [-e RWS=<websocket port>] 10.12.32.91/library/webr-proxy
  + example:
  + docker -d --name 23.21 -p 3321:80 -e RHOST=10.12.23.21 10.12.32.91/library/webr-proxy
  + library/webr-direct
  + 如果是本机Debug的话，可以使用该镜像直接连接自己启动的R。
  + default **RHOST** is **127.0.0.1** *external ip*
  + default **RPORT** is **3582** *Api port*
  + default **RWS** is **8298** *websocket port*
  + docker -d [--name <container name>] -p <external port>:80 [-e RHOST=<r host>] [-e RPORT=<api port>] [-e RWS=<websocket port>] 10.12.32.91/library/webr-direct
  + example:
  + docker -d --name debugWebR -p 3366:80 -e RHOST=10.12.23.66 10.12.32.91/library/webr-direct