14.10 Container isolation

Docker provides namespace isolation between different containers, and between the containers and the host platform. Note however, that all containers and the host share the same operating system kernel. Some kernel configuration changes will affect all containers and the host. For example, use of sysctl to modify Address Space Layout Randomization (ASLR) will effect all containers and the effects will persist in the host after the containers are stopped. However, some tuning parameters such as net.ipv4.ip_forward are isolated, i.e., local to the container. These do get reset in ways that are hard to predict, so it is suggested that sysctl tuning be done in rc.local scripts so that they happen on each boot.

Note also, that the Docker group (in which containers execute) is root equivalent, and thus a hostile container can do damage to the Linux host.

14.11 Test registry setup

The test registry is a Docker container that runs on the host, i.e., native OS upon which the VMs run. The same test registry is shared by multiple development VMs. The test registry is created via `host_scripts/registry/start_reg.sh`. It listens to port 5000 on the localhost.

A VM is configured to use the test registry via `setup_scripts/./prep-testregistry.sh`

The test registry is populated using publish.py -t

14.12 CentOS containers

CentOS base containers do not run 32-bit binaries. Add the following to your dockerfile to do that:

```
RUN yum install -y compat-libstdc++-296.i686 compat-libstdc++-33.i686
```