部屬基本設定

vm1

docker swarm init --advertise-addr IP # vm1 IP

設定工作節點

vm2

將剛剛vm1獲得的節點輸入至vm2

vm3

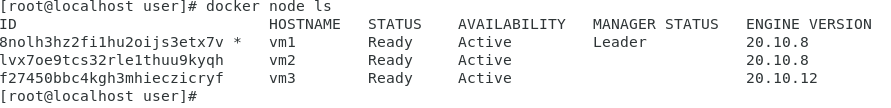
將剛剛vm1獲得的節點輸入至vm3

忘記token的話

docker swarm join-token -q worker

vm1

輸入docker node ls可看到當前的節點



docker node promote [HOSTNAME ]: 升級工作節點

docker node demote [HOSTNAME] : 降級工作節點

更新節點

docker node update [OPTIONS] [HOSTNAME] OPTIONS：

--availability active/drain (執行/不執行)

--role worker/manager

移除節點

docker node rm -f [HOSTNAME]

再次新增 node 為 manager / worker

docker swarm join-token manager/worker

離開 swarm

docker swarm leave –force

圖形化介面

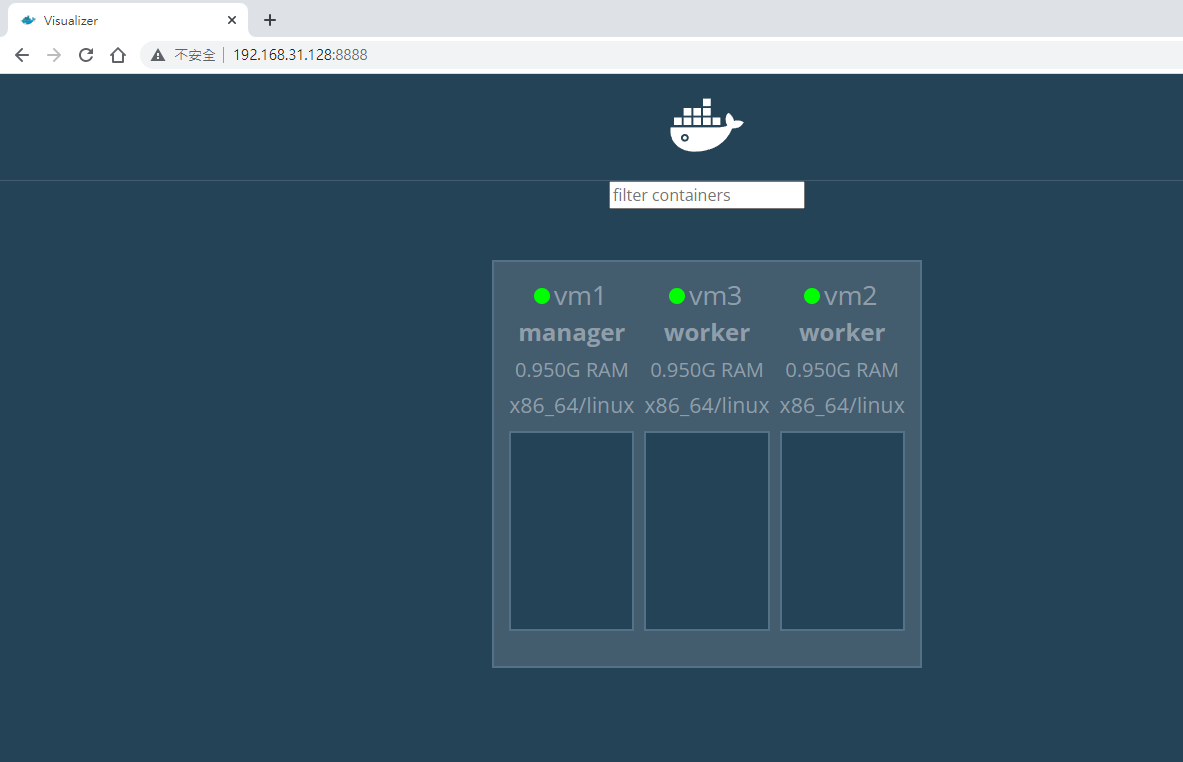
vm1

輸入docker pull dockersamples/visualizer

輸入systemctl restart docker重啟docker

輸入docker run -itd -p 8888:8080 -e HOST=192.168.138.135 -e PORT=8080 -v /var/run/docker.sock:/var/run/docker.sock --name visualizer dockersamples/visualizer

再chrome 網址輸入192.168.138.135:8888



部屬服務到集群中

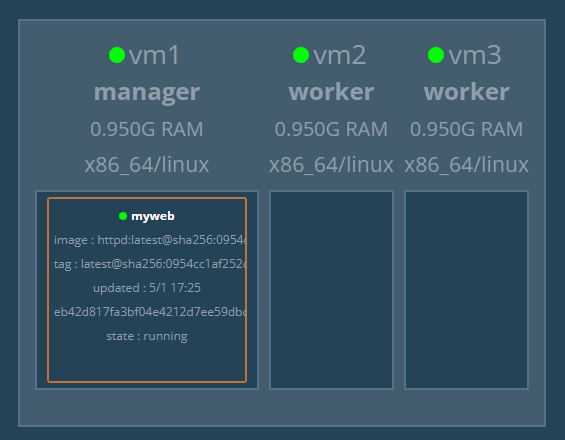
vm1

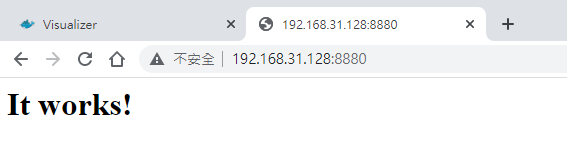
docker service create --name myweb -p 8880:80 httpd

查看服務

docker service ls

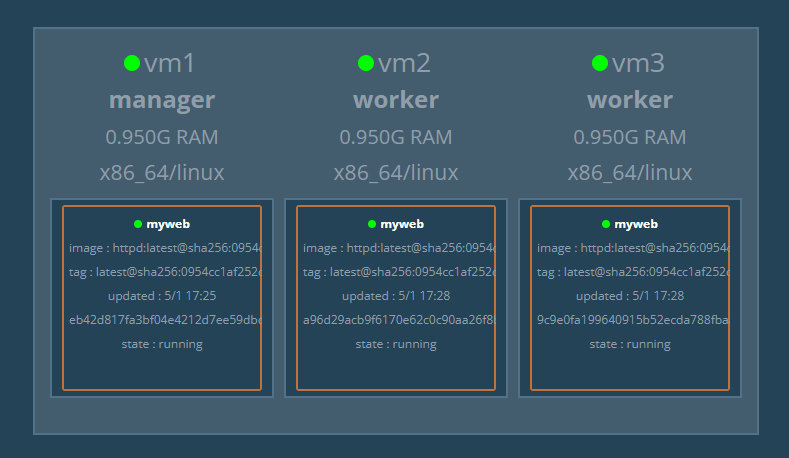






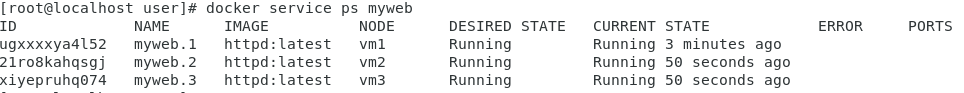
拓展服務

docker service scale myweb=3



以指令方式查看服務

docker service ps myweb



刪除服務

docker service rm myweb

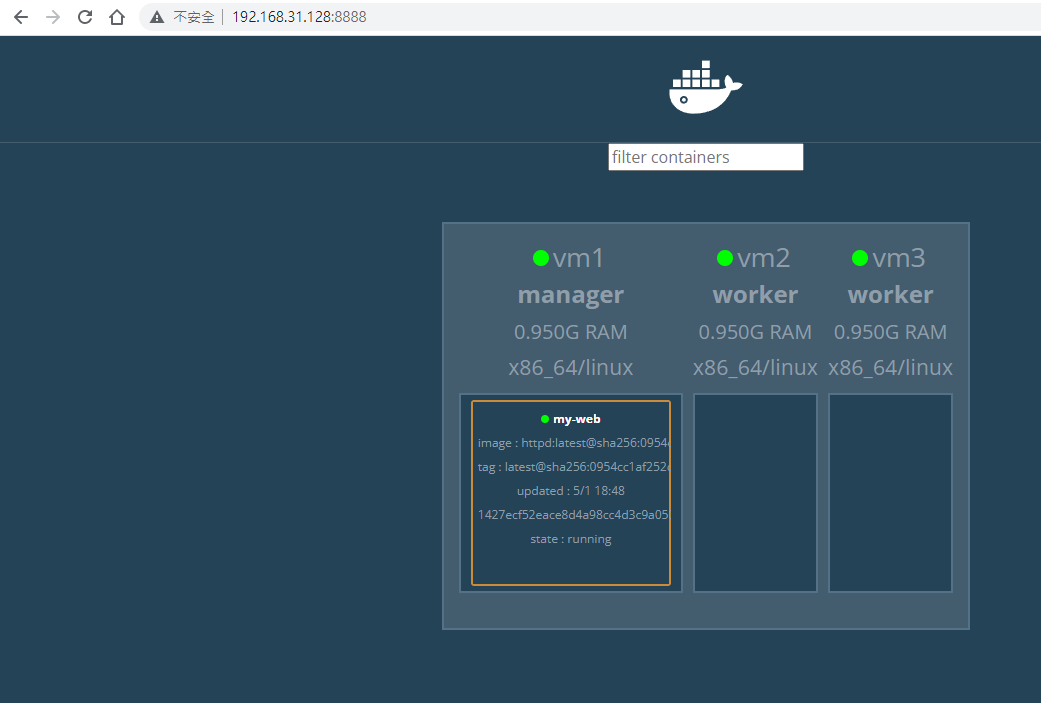
如果不想服務部屬到機器1

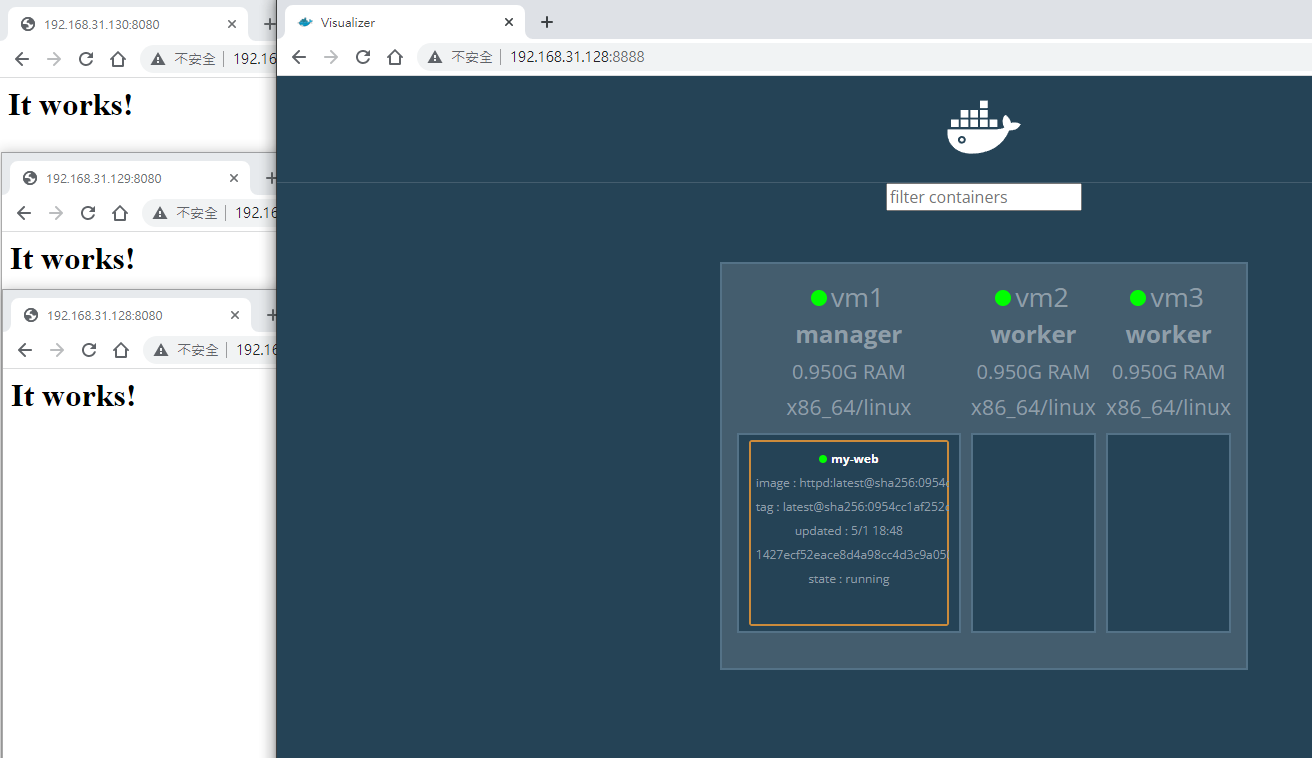
docker node update --availability active vm1



docker swarm mode routing mesh

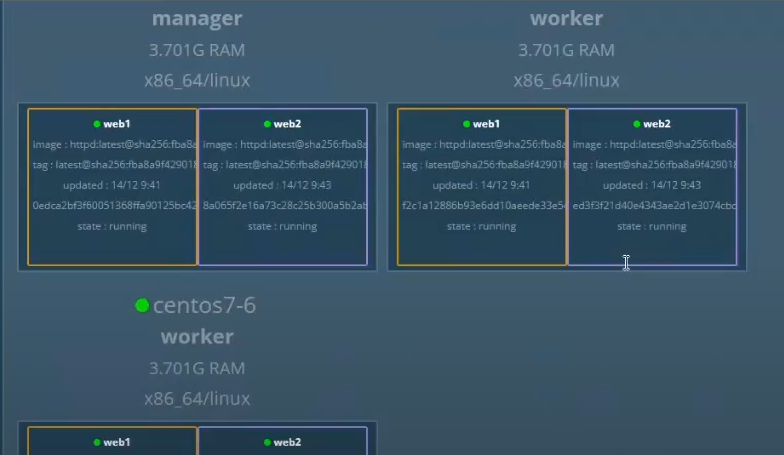
輸入docker service create --name my-web --publish published=8080,target=80 --replicas 1 httpd





若無法連接其他IP網址則參考: <https://stackoverflow.com/questions/48360577/docker-swarm-mode-routing-mesh-not-working>

Service有两种运行模式，一是replicated，指定一个Service运行容器的数量； 二是global，在所有符合运行条件的Node上，都运行一个这类容器。 docker service命令可以操作Swarm中的Service。

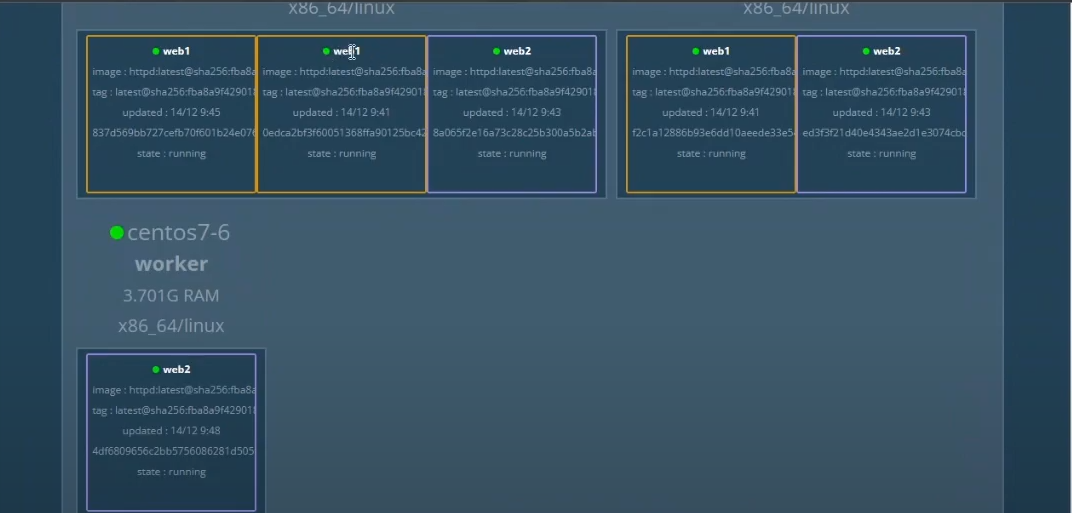




web1為replicated

web2為global

在其中一台機器停止後為上圖



待機器重新開啟後為上圖

global會重新生成

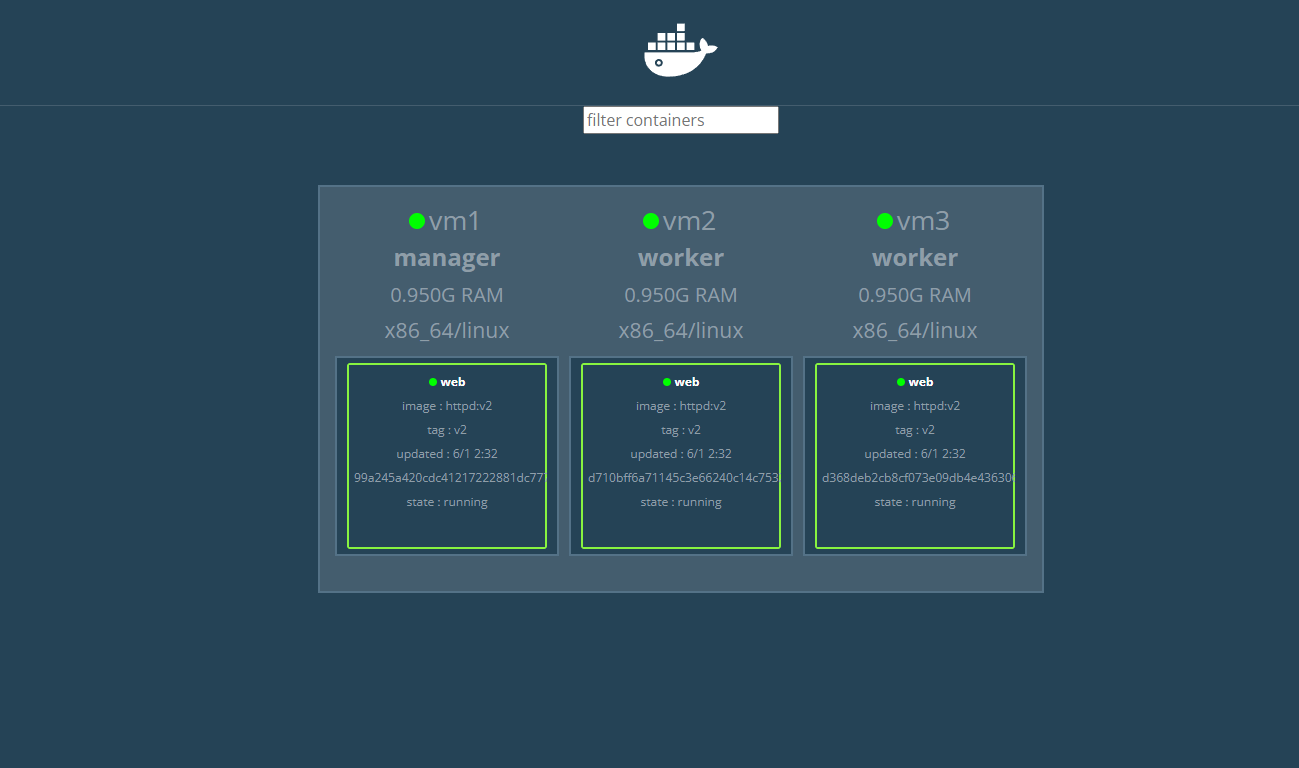
Docker Swarm 滾動更新

先將httpd:v1 v2 上傳至vm2 vm3

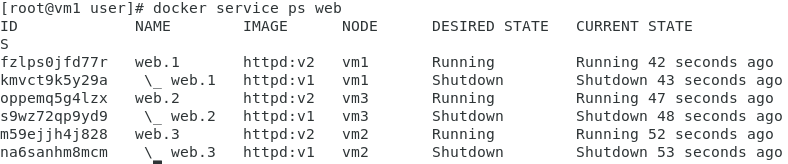
vm1

輸入docker service create --replicas 3 --name web -p 8000:80 httpd:v1 部屬服務

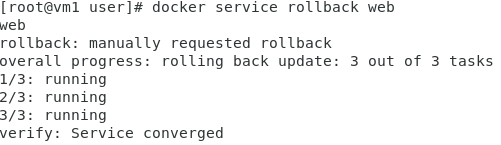
輸入docker service update --image httpd:v2 web 來更新服務



輸入docker service ps web 查看紀錄

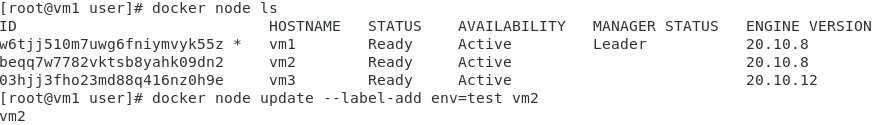


可以輸入docker service rollback web回復服務



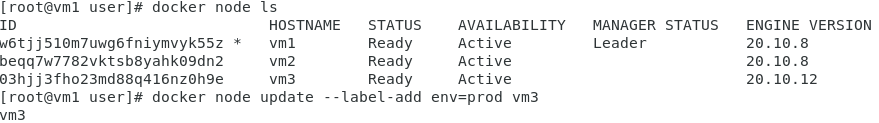
Label

輸入docker node update –label-add env=test vm2



給予vm2 一個標籤 env=test

輸入docker node update –label-add env=prod vm3



給予vm3 一個標籤 env=prod

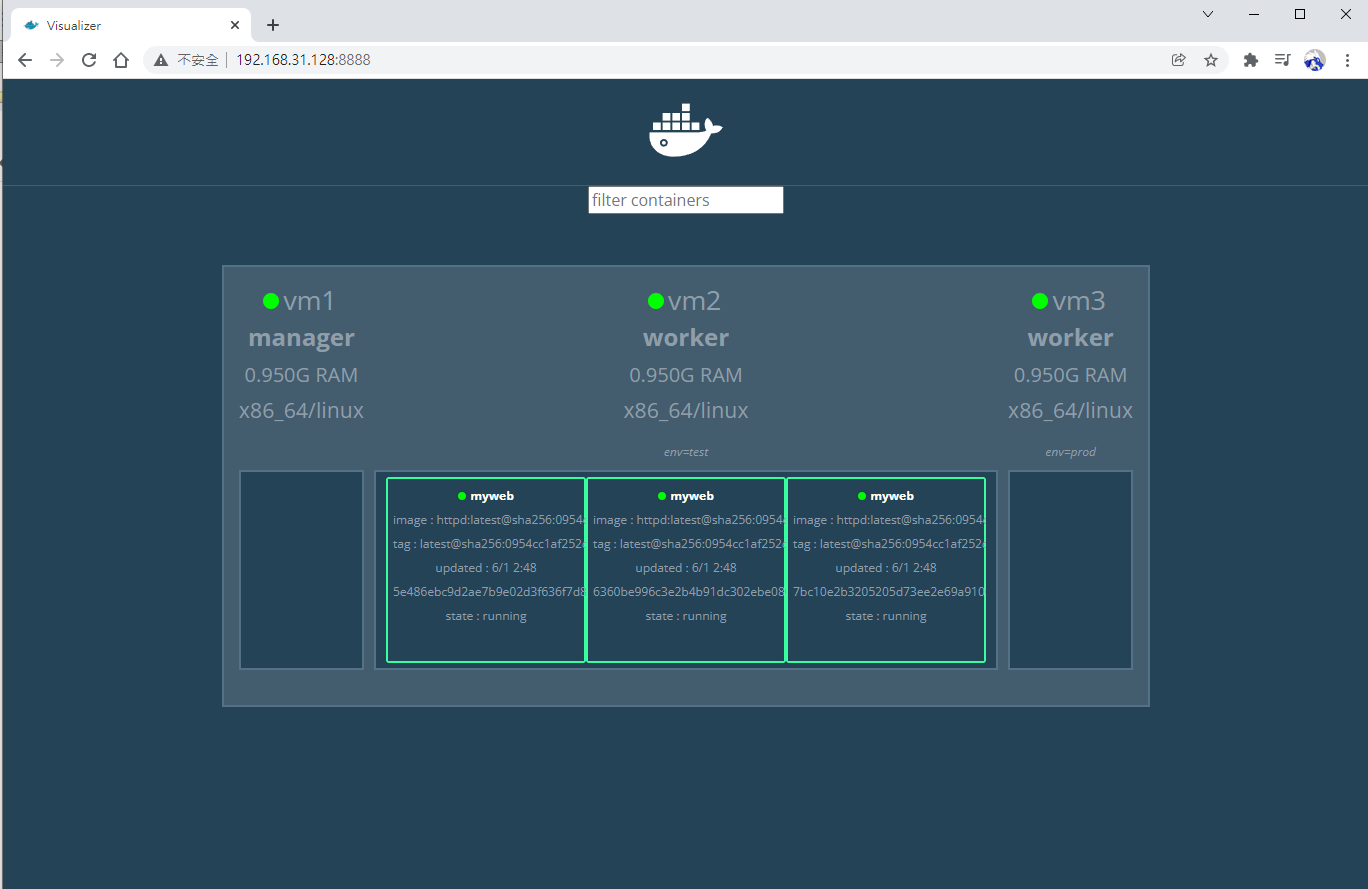
輸入docker node inspect vm2 –pretty 查看



輸入docker node inspect vm3 –pretty 查看

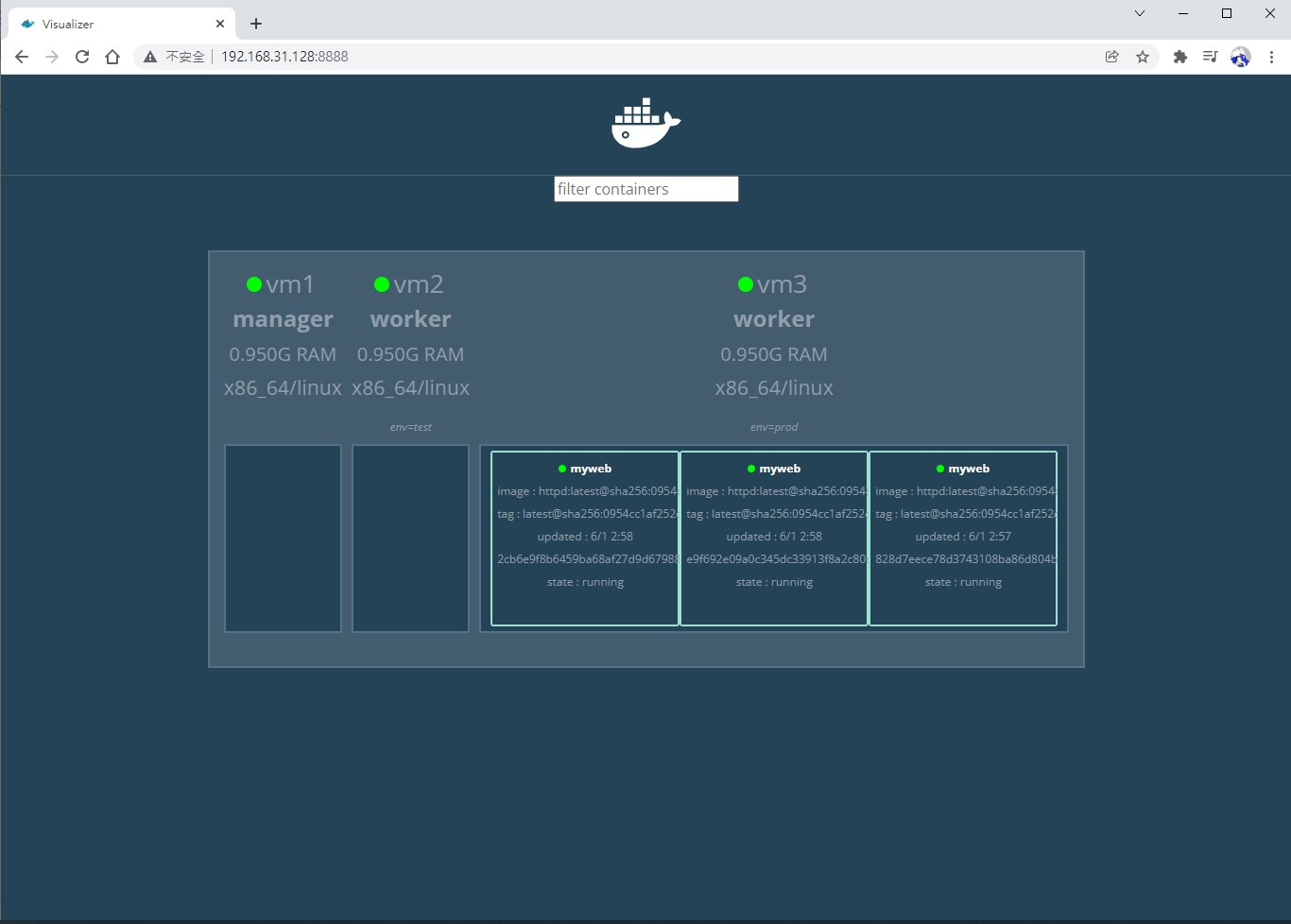


輸入docker service create --constraint node.labels.env==test --replicas 3 --name myweb -p 8000:80 httpd 部屬到特定labels的服務



先輸入docker service update --constraint-rm node.lables.env==test myweb 移除條件限制

再輸入docker service update --constraint-add node.labels.env==prod myweb 新增條件限制



移除標籤

docker node update --label-rm env vm2

docker node update --label-rm env vm3