steps

Bringup our docker VM

- Install vagrant Install vagrant
- Install VirtualBox Install VirtualBox
- Install Guest additions Install VirtualBox
- Bring up a centos/8 vm

```
vagrant init centos/8
```

```
# -*- mode: ruby -*-
# vi: set ft=ruby :

Vagrant.configure("2") do |config|
  config.vm.box = 'centos/8'
  config.disksize.size = '20GB'
  #config.vm.network 'public_network', bridge: 'wlp1s0' # attach it to the de
  config.vm.provision "shell", inline: <<-SHELL
    yum install -y yum-utils
    yum-config-manager --add-repo https://download.docker.com/linux/centos/do
    yum install -y docker-ce docker-ce-cli containerd.io
    systemctl start docker
    systemctl enable docker
    usermod -aG docker vagrant
  SHELL

  config.vm.provider 'virtualbox' do |v|

    v.memory = 2048
    v.cpus = 2
  end
end
```

open up 3 terminals withing your editor

- 1st terminal `vagrant up`
- 2nd terminal `vagrant rsync-auto`
- 3rd terminal `vagrant ssh`

Validate if docker is running

```
ps aux | grep docker
# si no hay ningun proceso
systemctl start docker # iniciamos el servicio
systemctl enable docker # habilitamos para que arrante auto
ps aux | grep docker
# agregamos al usuario al grupo de docker
sudo usermod -aG docker vagrant
exit
vagrant ssh
```

How to open multiple terminals in vs-code MS-windows

- Ctrl+Shift+P
- Terminal: Select Default Shell
- Select bash?
- clik on + sign

Docker help

```
vagrant ssh
docker help
docker help cp
```

Building a website monitor

1

```
mkdir mailer && cd $_
```

mailer.sh

```
#!/bin/sh
printf "Mailer has started.\n"

while true
```

```
do
  MESSAGE=`nc -l -p 33333`
  printf "Sending email: %s\n" $MESSAGE
  sleep 1
done
```

Write a mailer `Dockerfile`

```
FROM busybox
COPY . /mailer
WORKDIR /mailer

RUN adduser -DHs /bin/bash example

RUN chown example mailer.sh
RUN chmod a+x mailer.sh
EXPOSE 3333

USER example
CMD ["/mailer/mailer.sh"]
```

Build the image

```
vagrant rsync
vagrant ssh
cd /vagrant/mailer
docker build . -t mvelasquez/mailer:1.0.0
cd ..
```

2

```
mkdir agent && cd $_
```

create watcher.sh

```
#!/bin/sh -x
while true
do
  if `printf "GET / HTTP/1.0\n\n" nc -w 2 $INSIDEWEB_PORT_80_TCP_ADDR $INSIDEW
  then
    echo "System up."
  else
```

```
    printf "To: admin@work Message: The service is down!" | nc $INSIDEMAILER_P
    break
  fi
  sleep 1
done
```

Write agent Dockerfile

```
FROM busybox
COPY . /watcher
WORKDIR /watcher

RUN adduser -DHs /bin/bash example
RUN chown example watcher.sh
RUN chmod a+x watcher.sh

USER example

CMD ["/watcher/watcher.sh"]
```

Build image

```
vagrant rsync
vagrant ssh
cd /vagrant/agent
docker build . -t mvelasquez/agent:1.0.0
cd ..
```

Detached container

```
docker run --detach --name web nginx:latest
```

AKA demon mode

```
docker run -d --name mailer mvelasquez/mailer
```

Runing interactive container

```
docker run --interactive --tty \
--link web:web \
--name web_test \
busybox:1.29 /bin/sh
```

Building up our stack

```
docker run -it \
  --name agent \
  --link web:insideweb \
  --link mailer:insidemailer \
  mvelasquez/agent:1.0.0
```

Practice

Having: nginx servers A B and C monitors D E and F Mailer G Build: