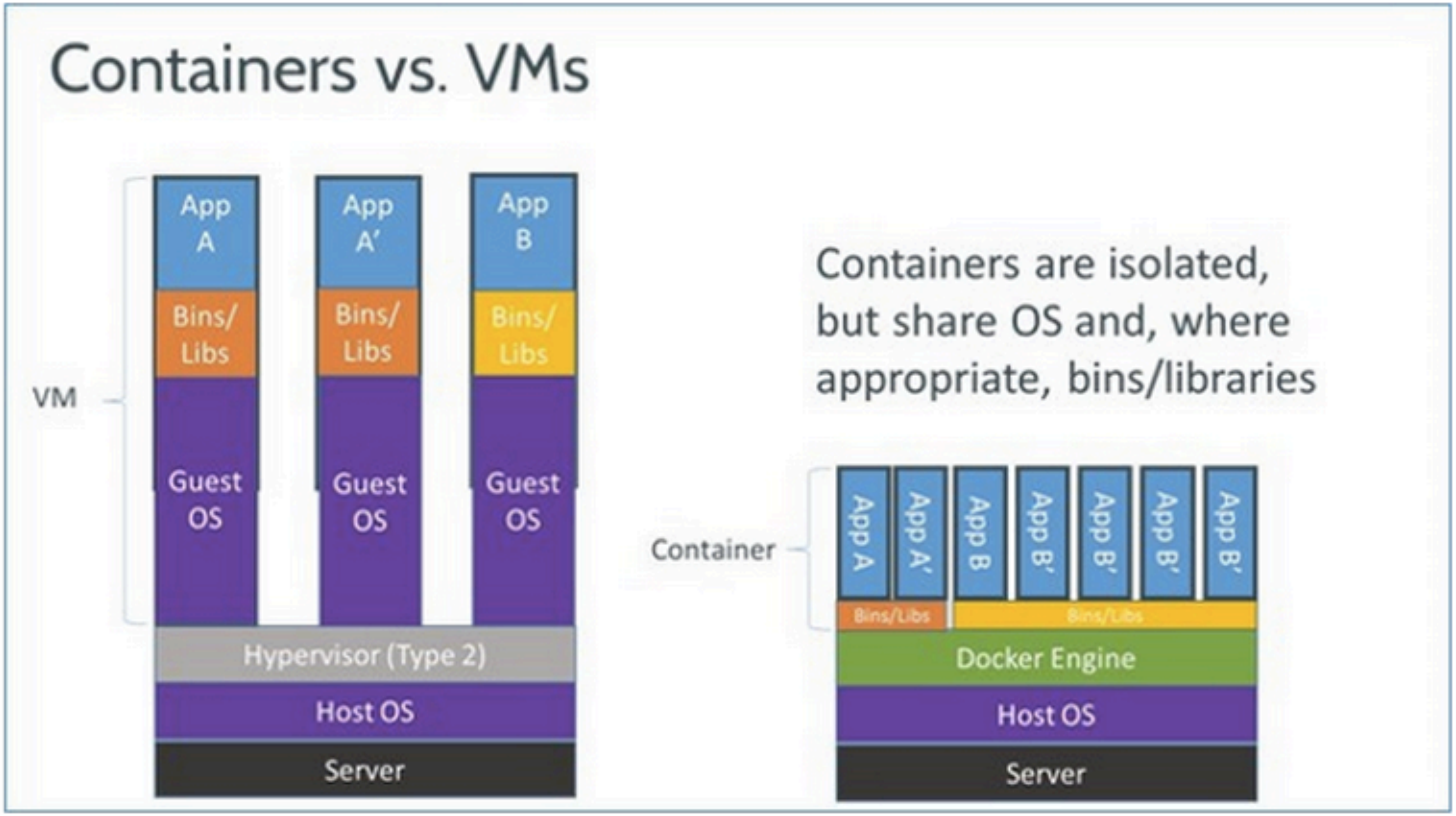
Introduction to Docker

burt.walsh@ast.myflorida.com

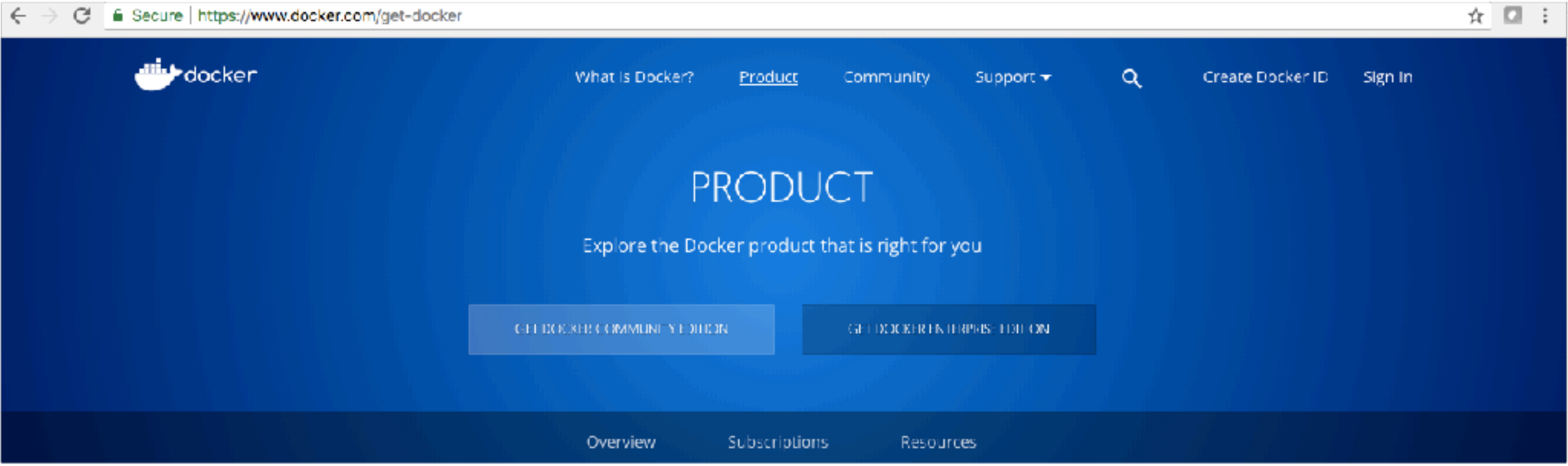
Docker Approach



Containers vs. VMs

Containers are isolated, but share OS and, where appropriate, bins/libraries

Where to Get Docker

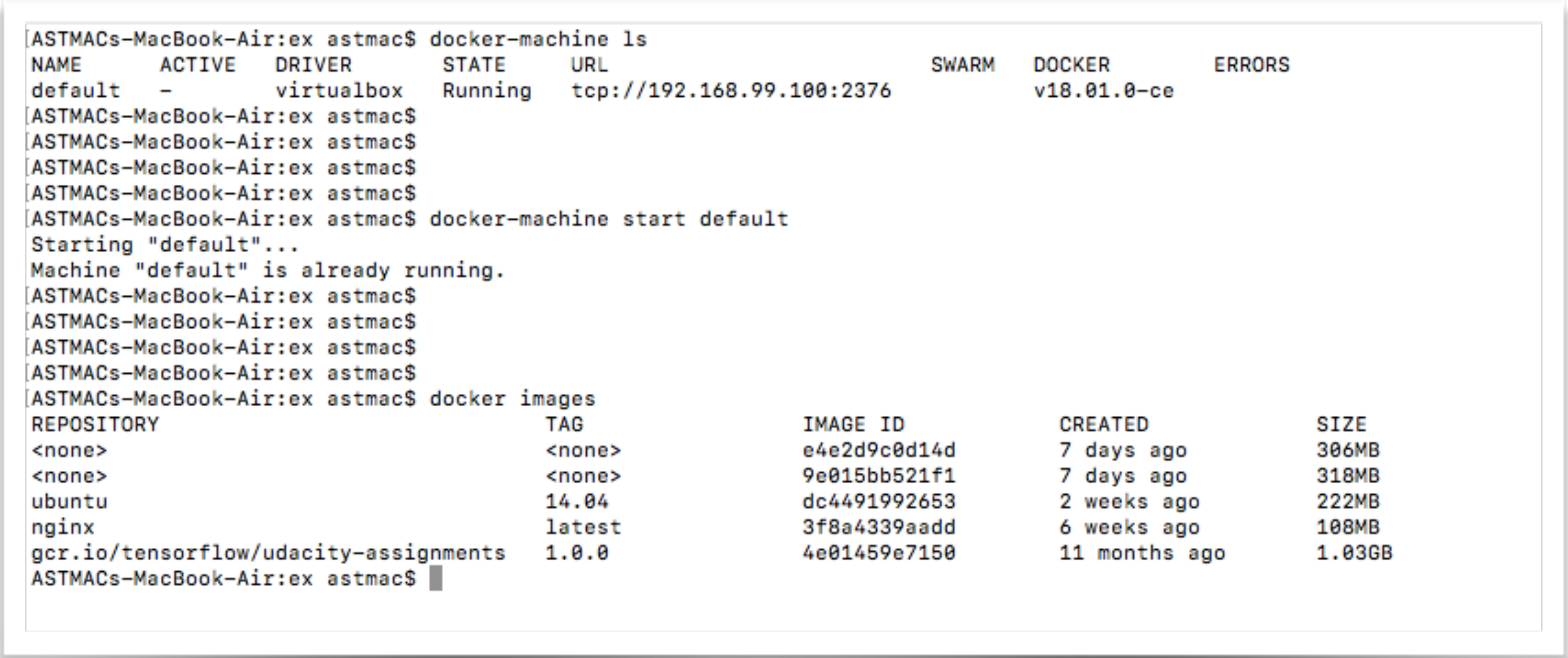


Secure | https://www.docker.com/get-docker

PRODUCT

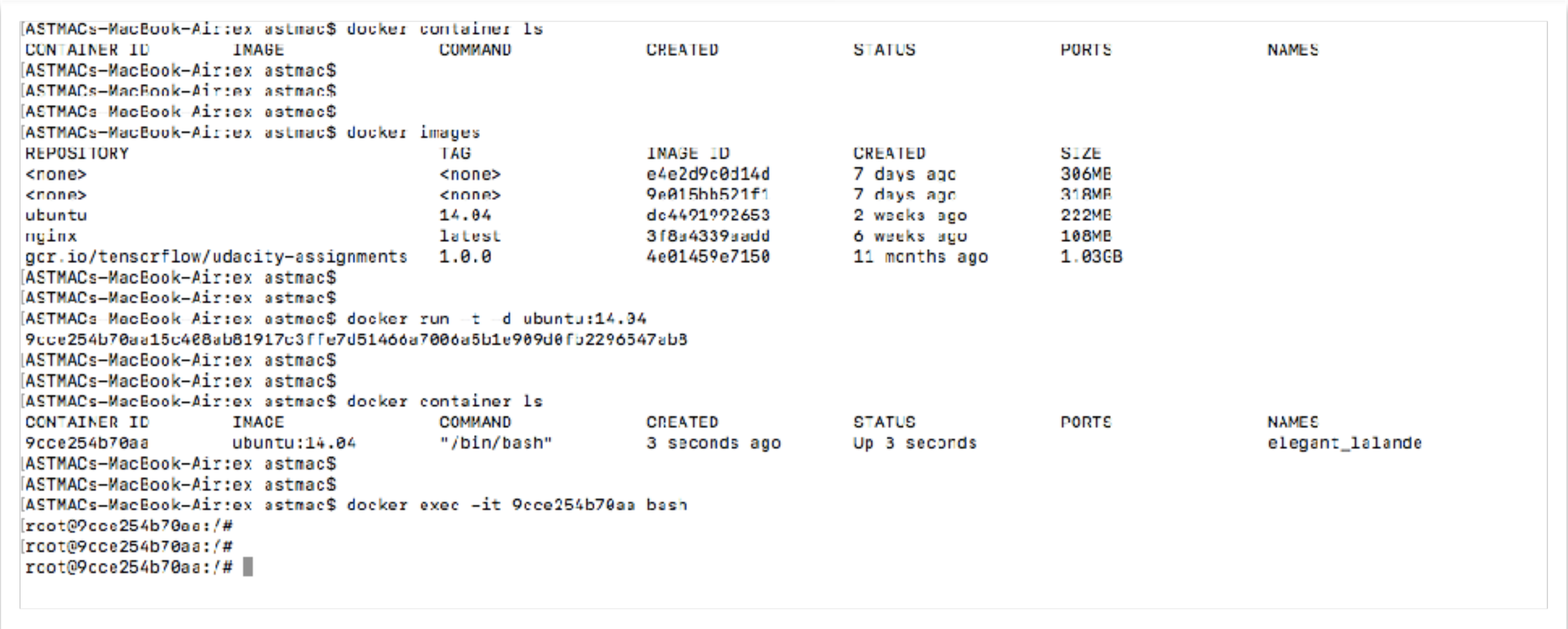
Explore the Docker product that is right for you

Starting the Docker Machine



```
[ASTMACs-MacBook-Air:ex astmac$ docker-machine ls
 NAME      ACTIVE   DRIVER       STATE     URL                         SWARM   DOCKER        ERRORS
 default   -        virtualbox   Running   tcp://192.168.99.100:2376           v18.01.0-ce
[ASTMACs-MacBook-Air:ex astmac$
[ASTMACs-MacBook-Air:ex astmac$
[ASTMACs-MacBook-Air:ex astmac$
[ASTMACs-MacBook-Air:ex astmac$
[ASTMACs-MacBook-Air:ex astmac$ docker-machine start default
 Starting "default"...
 Machine "default" is already running.
[ASTMACs-MacBook-Air:ex astmac$
[ASTMACs-MacBook-Air:ex astmac$
[ASTMACs-MacBook-Air:ex astmac$
[ASTMACs-MacBook-Air:ex astmac$
[ASTMACs-MacBook-Air:ex astmac$ docker images
 REPOSITORY                              TAG                 IMAGE ID            CREATED             SIZE
 <none>                                  <none>              e4e2d9c0d14d        7 days ago          306MB
 <none>                                  <none>              9e015bb521f1        7 days ago          318MB
 ubuntu                                  14.04               dc4491992653        2 weeks ago         222MB
 nginx                                   latest              3f8a4339aadd        6 weeks ago         108MB
 gcr.io/tensorflow/udacity-assignments   1.0.0               4e01459e7150        11 months ago       1.03GB
 ASTMACs-MacBook-Air:ex astmac$ 
```

What no containers? Lets start one and log on!

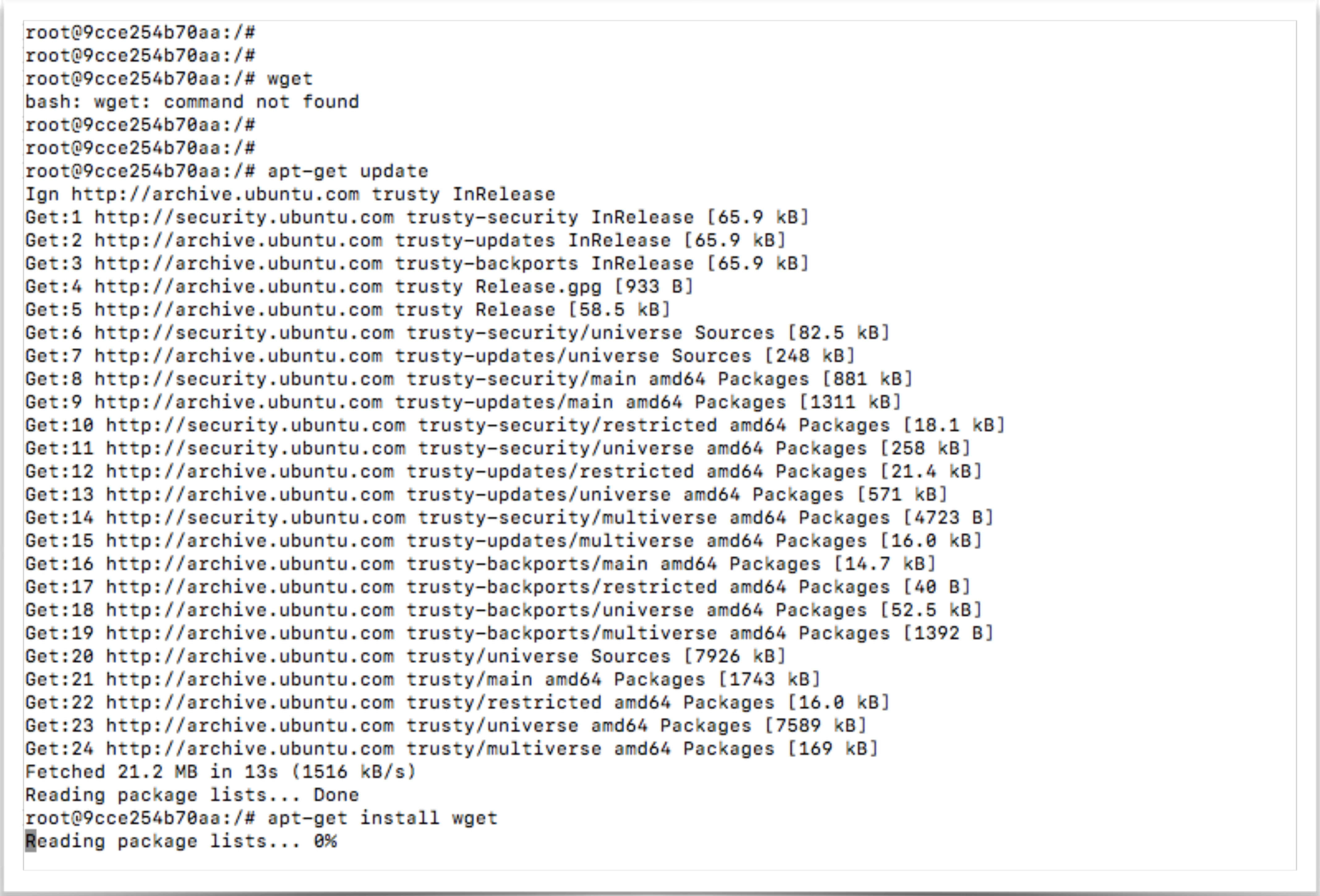


```
[ASTMACs-MacBook-Air:ex astmac$ docker container ls
 CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
[ASTMACs-MacBook-Air:ex astmac$
[ASTMACs-MacBook-Air:ex astmac$
[ASTMACs-MacBook-Air:ex astmac$
[ASTMACs-MacBook-Air:ex astmac$ docker images
 REPOSITORY                              TAG                 IMAGE ID            CREATED             SIZE
 <none>                                  <none>              e4e2d9c0d14d        7 days ago          306MB
 <none>                                  <none>              9e015bb521f1        7 days ago          318MB
 ubuntu                                  14.04               dc4421992653        2 weeks ago         222MB
 nginx                                   latest              3f8a4339aadd        6 weeks ago         108MB
 gcr.io/tensorflow/udacity-assignments   1.0.0               4e01459e7150        11 months ago       1.03GB
[ASTMACs-MacBook-Air:ex astmac$
[ASTMACs-MacBook-Air:ex astmac$
[ASTMACs-MacBook-Air:ex astmac$ docker run -t -d ubuntu:14.04
 9cce254b70aa15c408ab81917c3ffe7d51466a7006a5b1e909d0fb2296547ab3
[ASTMACs-MacBook-Air:ex astmac$
[ASTMACs-MacBook-Air:ex astmac$
[ASTMACs-MacBook-Air:ex astmac$ docker container ls
 CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
 9cce254b70aa        ubuntu:14.04        "/bin/bash"         3 seconds ago       Up 3 seconds                            elegant_lalande
[ASTMACs-MacBook-Air:ex astmac$
[ASTMACs-MacBook-Air:ex astmac$
[ASTMACs-MacBook-Air:ex astmac$ docker exec -it 9cce254b70aa bash
[root@9cce254b70aa:/#
[root@9cce254b70aa:/#
 root@9cce254b70aa:/# 
```

Lets install some software

- apt-get update -y (updates all current packages)
- apt-get install -y nodejs (installs nodejs)
- apt-get install -y npm. (installs package manager)
- mkdir myweb (creates a new directory)
- cd myweb (changes directory)
- npm init (create a node project)
- npm install -y express —save (install web software)

Now that we are in the container… lets load some software

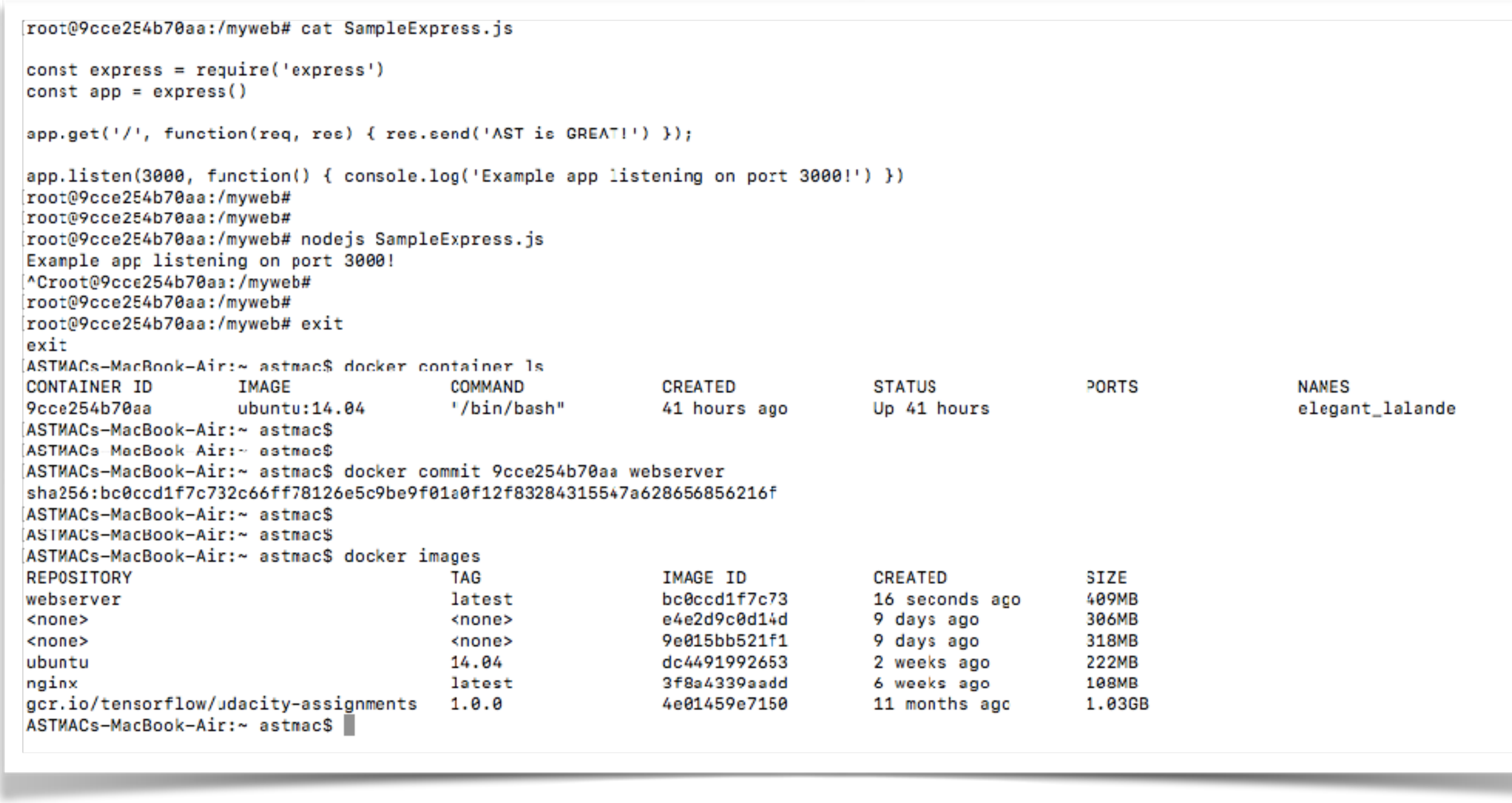


```
root@9cce254b70aa:/#
root@9cce254b70aa:/#
root@9cce254b70aa:/# wget
bash: wget: command not found
root@9cce254b70aa:/#
root@9cce254b70aa:/#
root@9cce254b70aa:/# apt-get update
Ign http://archive.ubuntu.com trusty InRelease
Get:1 http://security.ubuntu.com trusty-security InRelease [65.9 kB]
Get:2 http://archive.ubuntu.com trusty-updates InRelease [65.9 kB]
Get:3 http://archive.ubuntu.com trusty-backports InRelease [65.9 kB]
Get:4 http://archive.ubuntu.com trusty Release.gpg [933 B]
Get:5 http://archive.ubuntu.com trusty Release [58.5 kB]
Get:6 http://security.ubuntu.com trusty-security/universe Sources [82.5 kB]
Get:7 http://archive.ubuntu.com trusty-updates/universe Sources [248 kB]
Get:8 http://security.ubuntu.com trusty-security/main amd64 Packages [881 kB]
Get:9 http://archive.ubuntu.com trusty-updates/main amd64 Packages [1311 kB]
Get:10 http://security.ubuntu.com trusty-security/restricted amd64 Packages [18.1 kB]
Get:11 http://security.ubuntu.com trusty-security/universe amd64 Packages [258 kB]
Get:12 http://archive.ubuntu.com trusty-updates/restricted amd64 Packages [21.4 kB]
Get:13 http://archive.ubuntu.com trusty-updates/universe amd64 Packages [571 kB]
Get:14 http://security.ubuntu.com trusty-security/multiverse amd64 Packages [4723 B]
Get:15 http://archive.ubuntu.com trusty-updates/multiverse amd64 Packages [16.0 kB]
Get:16 http://archive.ubuntu.com trusty-backports/main amd64 Packages [14.7 kB]
Get:17 http://archive.ubuntu.com trusty-backports/restricted amd64 Packages [40 B]
Get:18 http://archive.ubuntu.com trusty-backports/universe amd64 Packages [52.5 kB]
Get:19 http://archive.ubuntu.com trusty-backports/multiverse amd64 Packages [1392 B]
Get:20 http://archive.ubuntu.com trusty/universe Sources [7926 kB]
Get:21 http://archive.ubuntu.com trusty/main amd64 Packages [1743 kB]
Get:22 http://archive.ubuntu.com trusty/restricted amd64 Packages [16.0 kB]
Get:23 http://archive.ubuntu.com trusty/universe amd64 Packages [7589 kB]
Get:24 http://archive.ubuntu.com trusty/multiverse amd64 Packages [169 kB]
Fetched 21.2 MB in 13s (1516 kB/s)
Reading package lists... Done
root@9cce254b70aa:/# apt-get install wget
Reading package lists... 0%
```

Create a WebServer Host and Run it

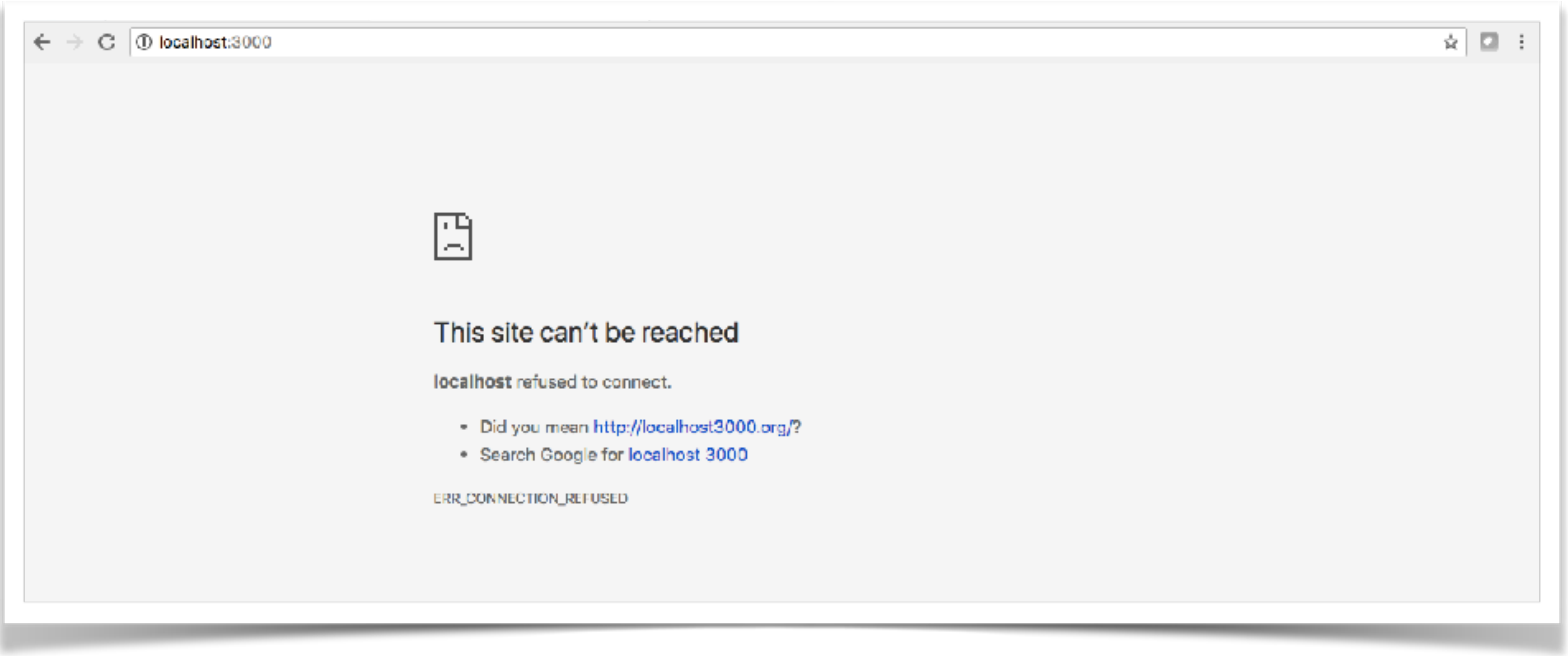
- Create a source file with instructions to start a web server (SampleExpress.js)
- Start the server under nodejs (container port 3000)
- Save the running container as an image so we can reuse it

Accessing the WebServer



```
root@9cce254b70aa:/myweb# cat SampleExpress.js

const express = require('express')
const app = express()

app.get('/', function(req, res) { res.send('AST is GREAT!') });

app.listen(3000, function() { console.log('Example app listening on port 3000!') })
root@9cce254b70aa:/myweb#
root@9cce254b70aa:/myweb#
root@9cce254b70aa:/myweb# nodejs SampleExpress.js
Example app listening on port 3000!
^Croot@9cce254b70aa:/myweb#
root@9cce254b70aa:/myweb#
root@9cce254b70aa:/myweb# exit
exit
ASTMACs-MacBook-Air:~ astmac$ docker container ls
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
9cce254b70aa        ubuntu:14.04        '/bin/bash"         41 hours ago        Up 41 hours                             elegant_lalande
ASTMACs-MacBook-Air:~ astmac$
ASTMACs-MacBook-Air:~ astmac$
ASTMACs-MacBook-Air:~ astmac$ docker commit 9cce254b70aa webserver
sha256:bc0ccd1f7c732c66ff78126e5c9be9f01a0f12f83284315547a628656856216f
ASTMACs-MacBook-Air:~ astmac$
ASTMACs-MacBook-Air:~ astmac$
ASTMACs-MacBook-Air:~ astmac$ docker images
REPOSITORY                              TAG                 IMAGE ID            CREATED             SIZE
webserver                               latest              bc0ccd1f7c73        16 seconds ago      409MB
<none>                                  <none>              e4e2d9c0d14d        9 days ago          306MB
<none>                                  <none>              9e015bb521f1        9 days ago          318MB
ubuntu                                  14.04               dc4491992653        2 weeks ago         222MB
nginx                                   latest              3f8a4339aadd        6 weeks ago         108MB
gcr.io/tensorflow/udacity-assignments   1.0.0               4e01459e7150        11 months ago       1.03GB
ASTMACs-MacBook-Air:~ astmac$
```

Site is not accessible



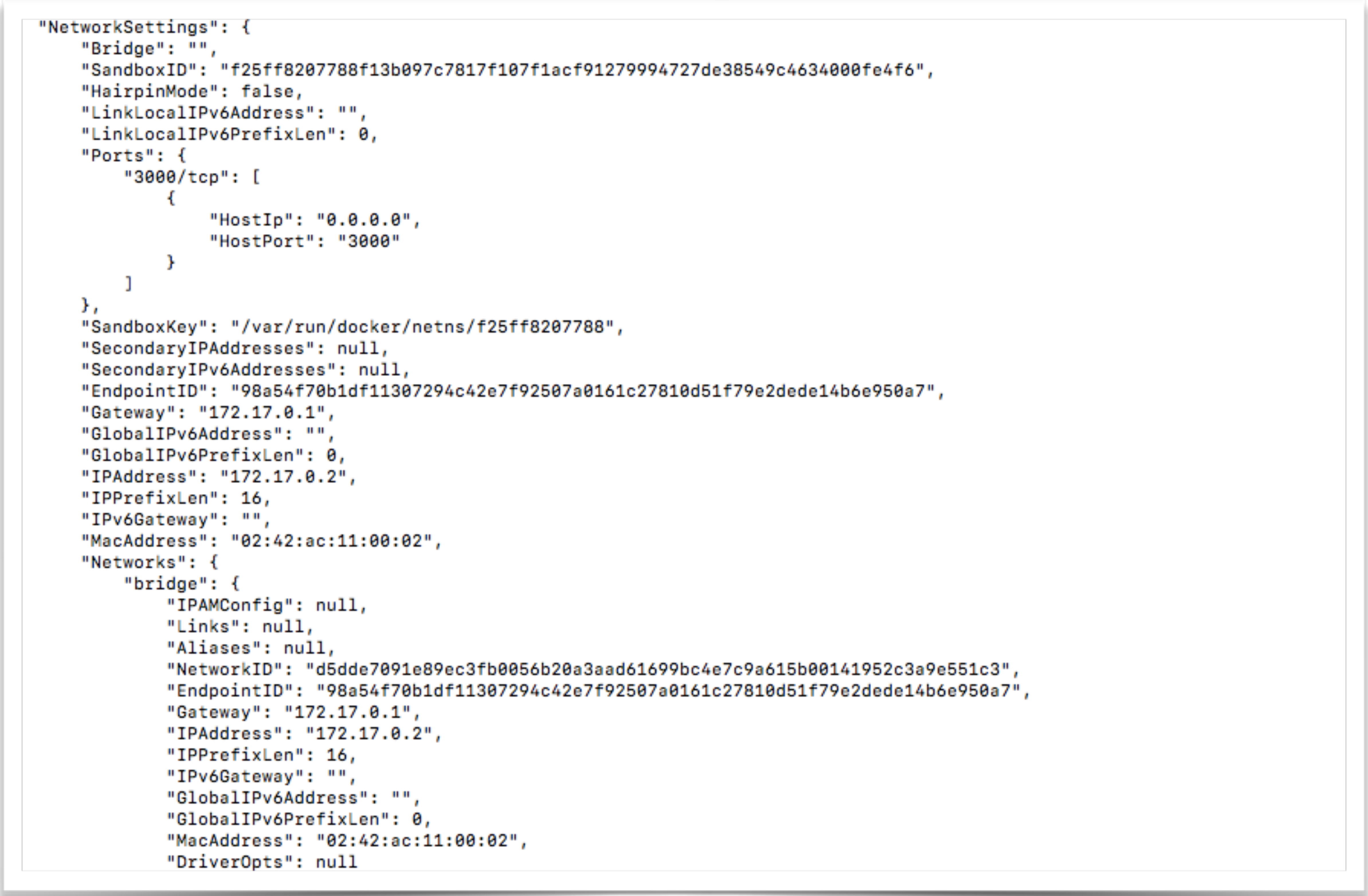
This site can't be reached

localhost refused to connect.

- Did you mean http://localhost3000.org/?
- Search Google for localhost 3000

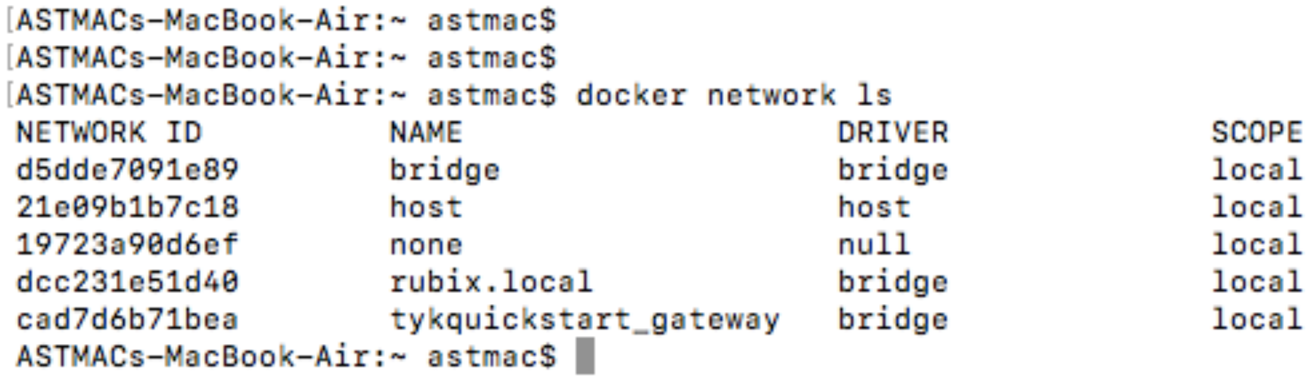
ERR_CONNECTION_REFUSED

Inspecting the Container



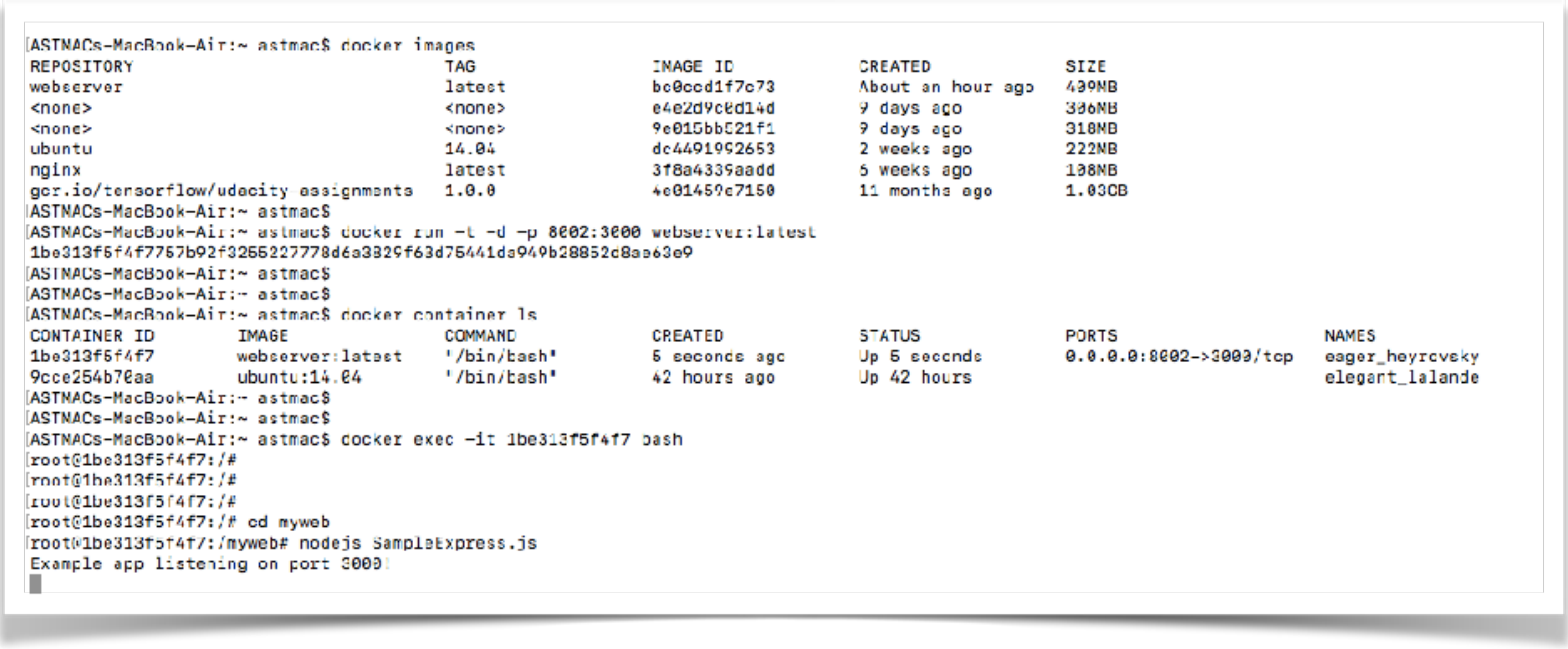
```
"NetworkSettings": {
    "Bridge": "",
    "SandboxID": "f25ff8207788f13b097c7817f107f1acf91279994727de38549c4634000fe4f6",
    "HairpinMode": false,
    "LinkLocalIPv6Address": "",
    "LinkLocalIPv6PrefixLen": 0,
    "Ports": {
        "3000/tcp": [
            {
                "HostIp": "0.0.0.0",
                "HostPort": "3000"
            }
        ]
    },
    "SandboxKey": "/var/run/docker/netns/f25ff8207788",
    "SecondaryIPAddresses": null,
    "SecondaryIPv6Addresses": null,
    "EndpointID": "98a54f70b1df11307294c42e7f92507a0161c27810d51f79e2dede14b6e950a7",
    "Gateway": "172.17.0.1",
    "GlobalIPv6Address": "",
    "GlobalIPv6PrefixLen": 0,
    "IPAddress": "172.17.0.2",
    "IPPrefixLen": 16,
    "IPv6Gateway": "",
    "MacAddress": "02:42:ac:11:00:02",
    "Networks": {
        "bridge": {
            "IPAMConfig": null,
            "Links": null,
            "Aliases": null,
            "NetworkID": "d5dde7091e89ec3fb0056b20a3aad61699bc4e7c9a615b00141952c3a9e551c3",
            "EndpointID": "98a54f70b1df11307294c42e7f92507a0161c27810d51f79e2dede14b6e950a7",
            "Gateway": "172.17.0.1",
            "IPAddress": "172.17.0.2",
            "IPPrefixLen": 16,
            "IPv6Gateway": "",
            "GlobalIPv6Address": "",
            "GlobalIPv6PrefixLen": 0,
            "MacAddress": "02:42:ac:11:00:02",
            "DriverOpts": null
```

Looking at Networks



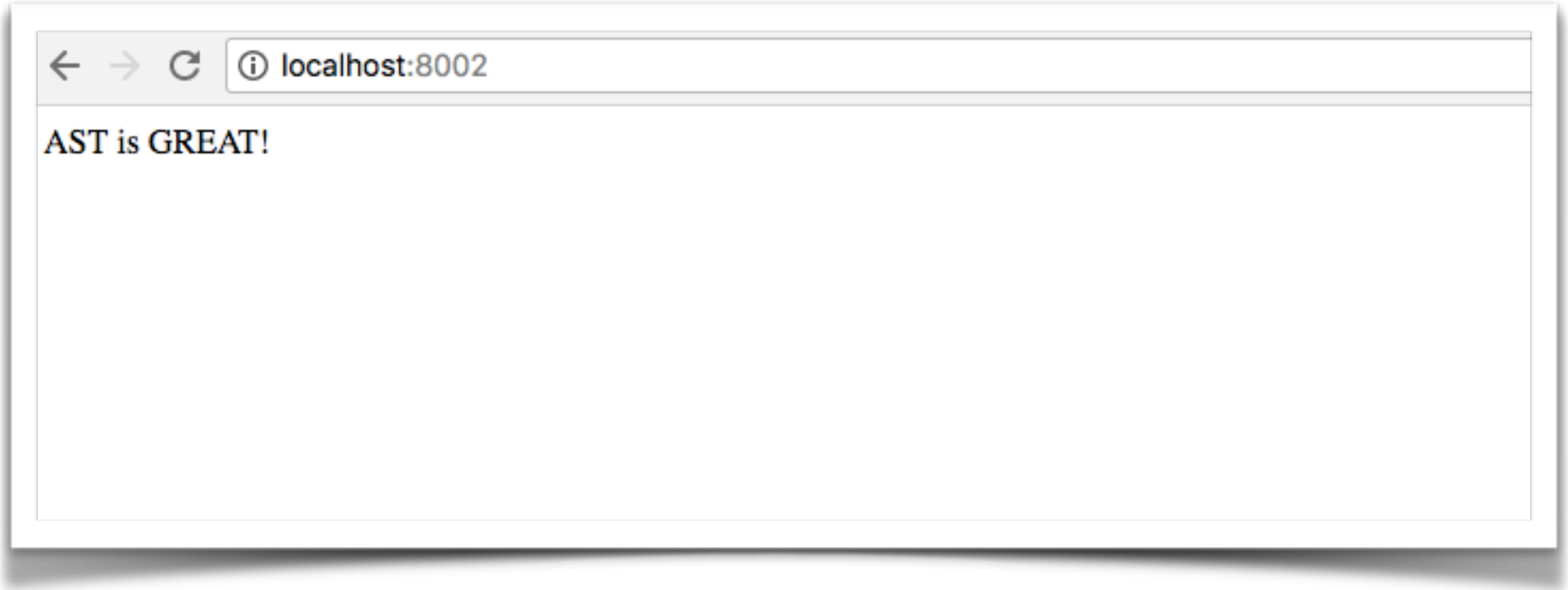
```
[ASTMACs-MacBook-Air:~ astmac$
[ASTMACs-MacBook-Air:~ astmac$
[ASTMACs-MacBook-Air:~ astmac$ docker network ls
 NETWORK ID          NAME                   DRIVER              SCOPE
 d5dde7091e89        bridge                 bridge              local
 21e09b1b7c18        host                   host                local
 19723a90d6ef        none                   null                local
 dcc231e51d40        rubix.local            bridge              local
 cad7d6b71bea        tykquickstart_gateway  bridge              local
 ASTMACs-MacBook-Air:~ astmac$
```

Running with Network Address Translation (NAT)
Bridge Network



```
[ASTNACs-MacBook-Air:~ astmac$ docker images
REPOSITORY                             TAG                 IMAGE ID            CREATED             SIZE
webserver                              latest              bc0ccd1f7c73        About an hour ago   439MB
<none>                                 <none>              e4e2d9c8d14d        9 days ago          386MB
<none>                                 <none>              9e015bb521f1        9 days ago          318MB
ubuntu                                 14.04               dc4491992653        2 weeks ago         222MB
nginx                                  latest              3f8a4339aadd        5 weeks ago         108MB
gcr.io/tensorflow/udacity-assignments  1.0.0               4e01459c7150        11 months ago       1.03GB
[ASTNACs-MacBook-Air:~ astmac$
[ASTNACs-MacBook-Air:~ astmac$ docker run -t -d -p 8002:3000 webserver:latest
1be313f5f4f7767b92f3266227778d6a3329f63d75441da949b28852d8aa63e9
[ASTNACs-MacBook-Air:~ astmac$
[ASTNACs-MacBook-Air:~ astmac$
[ASTNACs-MacBook-Air:~ astmac$ docker container ls
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS                    NAMES
1be313f5f4f7        webserver:latest    "/bin/bash"         5 seconds ago       Up 5 seconds        0.0.0.0:8002->3000/tcp   eager_heyrovsky
9cce254b70aa        ubuntu:14.04        "/bin/bash"         42 hours ago        Up 42 hours                                  elegant_lalande
[ASTNACs-MacBook-Air:~ astmac$
[ASTNACs-MacBook-Air:~ astmac$
[ASTNACs-MacBook-Air:~ astmac$ docker exec -it 1be313f5f4f7 bash
[root@1be313f5f4f7:/#
[root@1be313f5f4f7:/#
[root@1be313f5f4f7:/#
[root@1be313f5f4f7:/# cd myweb
[root@1be313f5f4f7:/myweb# nodejs SampleExpress.js
Example app listening on port 3000!
```

NAT in Action

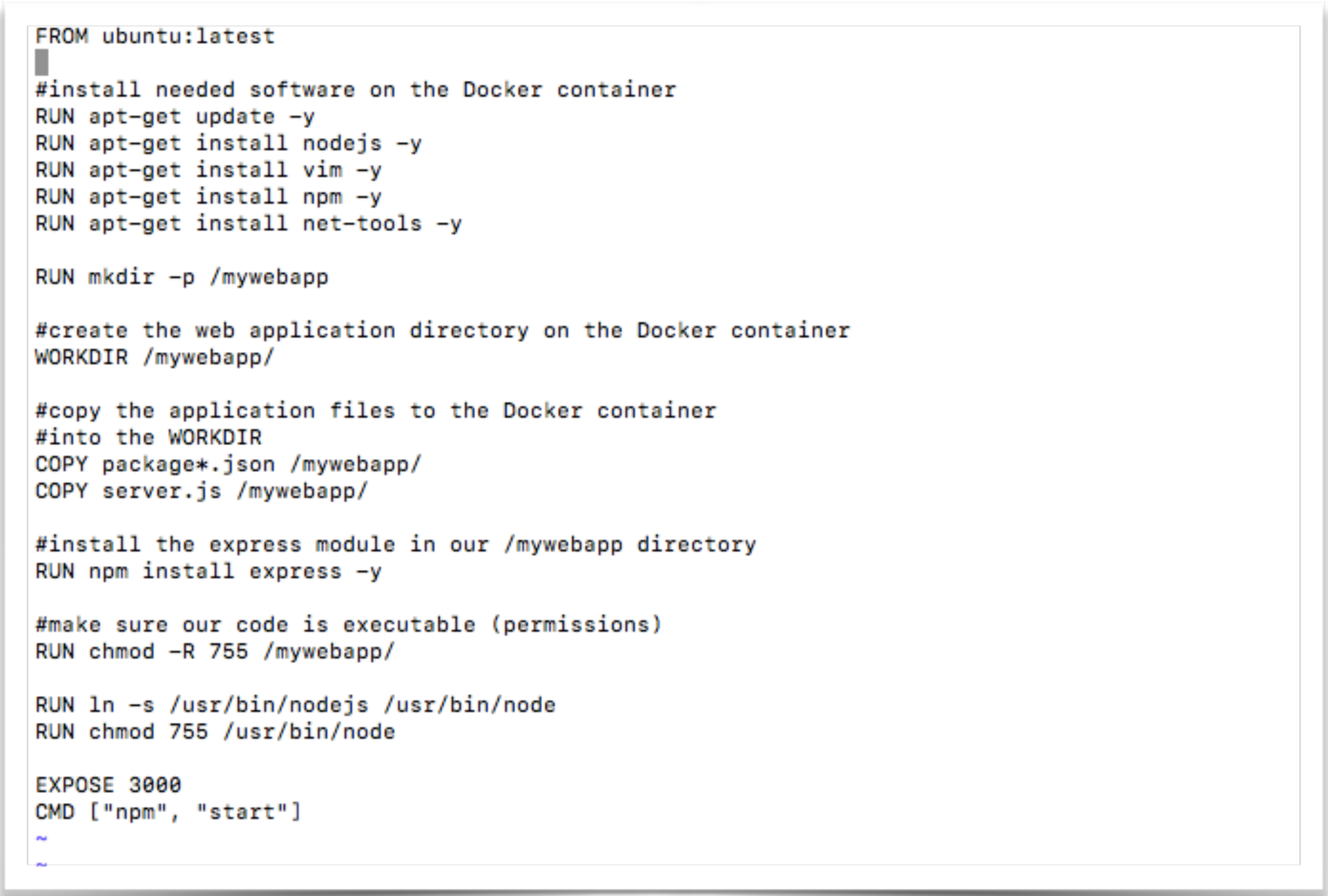


AST is GREAT!

Docker Files

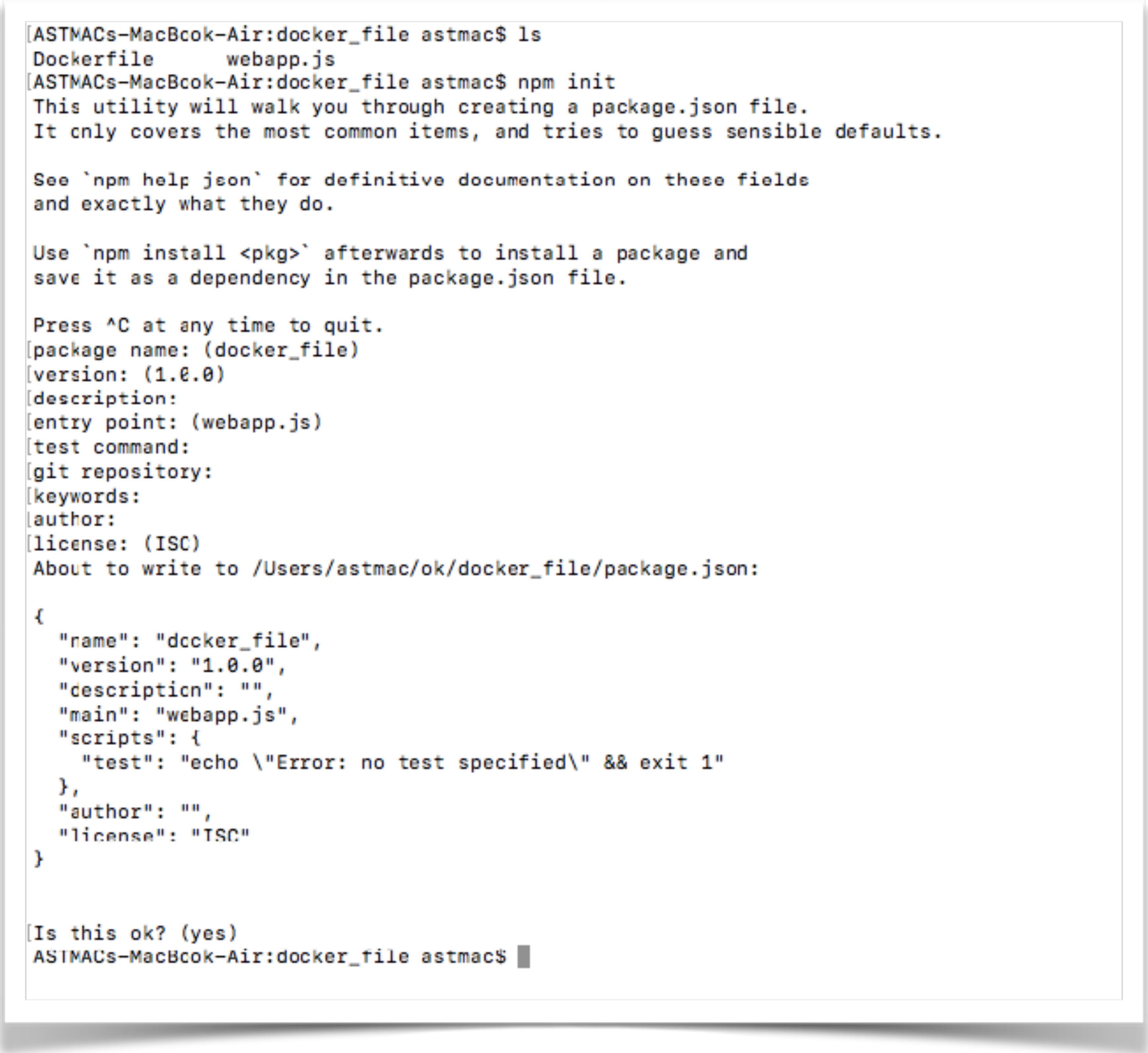
- Descriptive instructions for creation of a docker image
- Install software
- Start processes
- Expose ports

Example Dockerfile



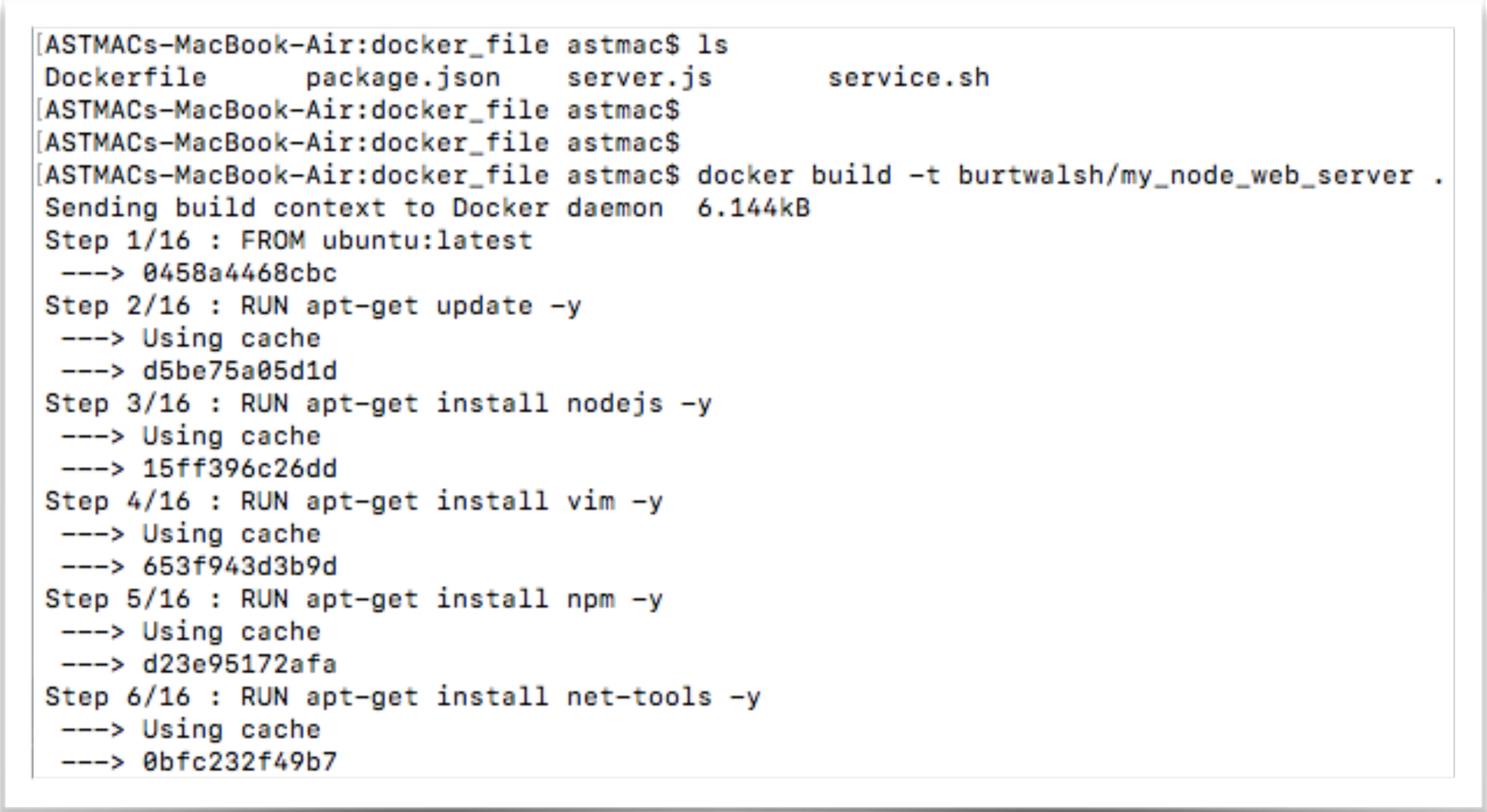
```
FROM ubuntu:latest

#install needed software on the Docker container
RUN apt-get update -y
RUN apt-get install nodejs -y
RUN apt-get install vim -y
RUN apt-get install npm -y
RUN apt-get install net-tools -y

RUN mkdir -p /mywebapp

#create the web application directory on the Docker container
WORKDIR /mywebapp/

#copy the application files to the Docker container
#into the WORKDIR
COPY package*.json /mywebapp/
COPY server.js /mywebapp/

#install the express module in our /mywebapp directory
RUN npm install express -y

#make sure our code is executable (permissions)
RUN chmod -R 755 /mywebapp/

RUN ln -s /usr/bin/nodejs /usr/bin/node
RUN chmod 755 /usr/bin/node

EXPOSE 3000
CMD ["npm", "start"]
~
~
```

Creating the Nodejs Project



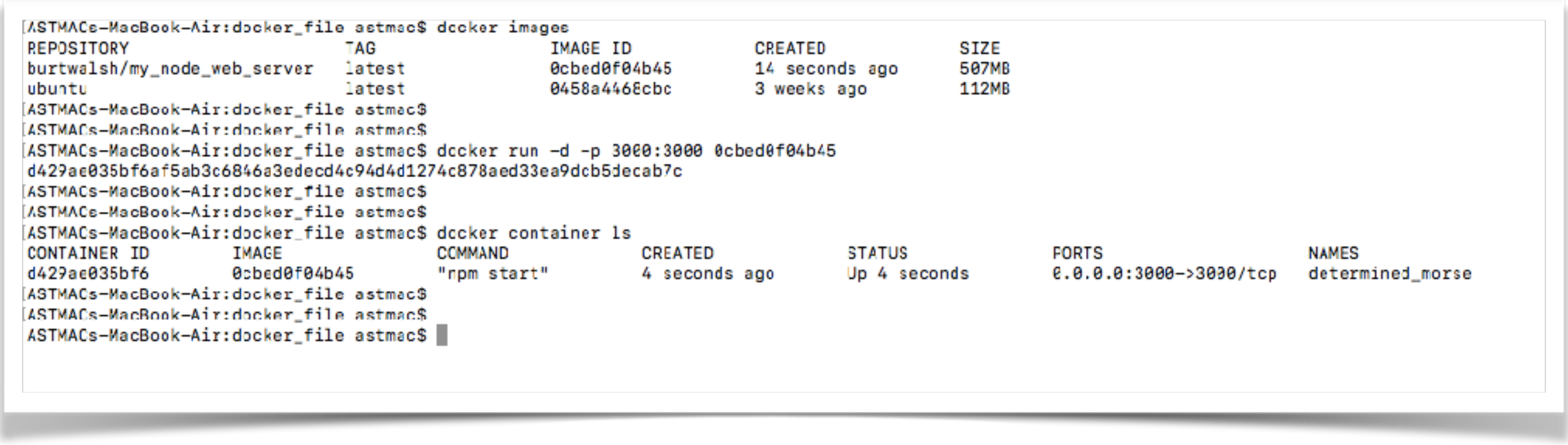
```
[ASTMACs-MacBook-Air:docker_file astmac$ ls
 Dockerfile      webapp.js
[ASTMACs-MacBook-Air:docker_file astmac$ npm init
 This utility will walk you through creating a package.json file.
 It only covers the most common items, and tries to guess sensible defaults.

 See `npm help json` for definitive documentation on these fields
 and exactly what they do.

 Use `npm install <pkg>` afterwards to install a package and
 save it as a dependency in the package.json file.

 Press ^C at any time to quit.
[package name: (docker_file)
[version: (1.0.0)
[description:
[entry point: (webapp.js)
[test command:
[git repository:
[keywords:
[author:
[license: (ISC)
 About to write to /Users/astmac/ok/docker_file/package.json:

 {
   "name": "docker_file",
   "version": "1.0.0",
   "description": "",
   "main": "webapp.js",
   "scripts": {
     "test": "echo \"Error: no test specified\" && exit 1"
   },
   "author": "",
   "license": "ISC"
 }


[Is this ok? (yes)
 ASTMACs-MacBook-Air:docker_file astmac$ 
```

Building the Image



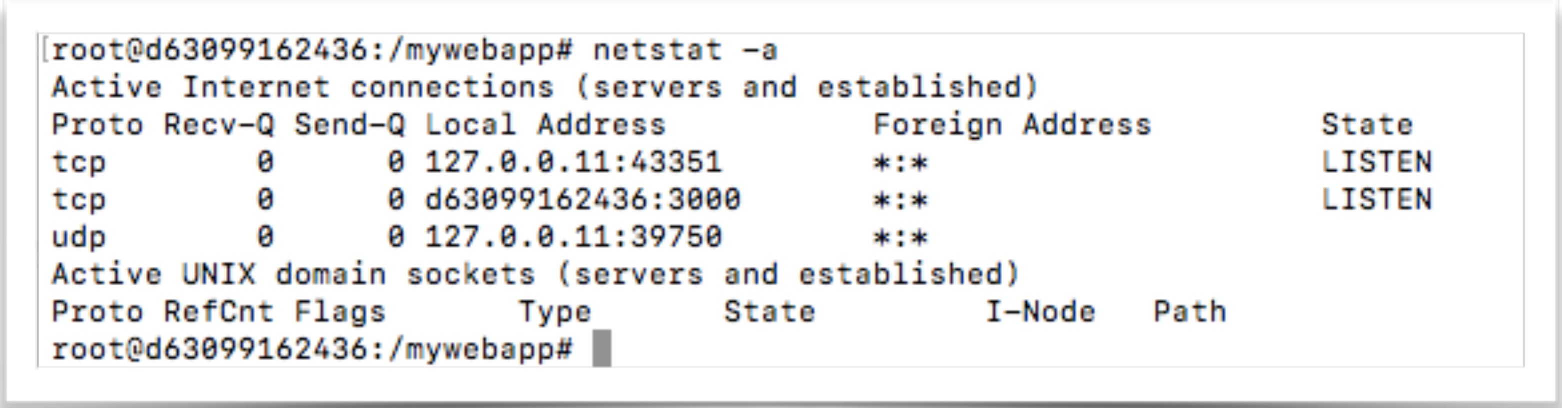
```
[ASTMACs-MacBook-Air:docker_file astmac$ ls
Dockerfile      package.json    server.js       service.sh
[ASTMACs-MacBook-Air:docker_file astmac$
[ASTMACs-MacBook-Air:docker_file astmac$
[ASTMACs-MacBook-Air:docker_file astmac$ docker build -t burtwalsh/my_node_web_server .
Sending build context to Docker daemon  6.144kB
Step 1/16 : FROM ubuntu:latest
 ---> 0458a4468cbc
Step 2/16 : RUN apt-get update -y
 ---> Using cache
 ---> d5be75a05d1d
Step 3/16 : RUN apt-get install nodejs -y
 ---> Using cache
 ---> 15ff396c26dd
Step 4/16 : RUN apt-get install vim -y
 ---> Using cache
 ---> 653f943d3b9d
Step 5/16 : RUN apt-get install npm -y
 ---> Using cache
 ---> d23e95172afa
Step 6/16 : RUN apt-get install net-tools -y
 ---> Using cache
 ---> 0bfc232f49b7
```

Running the Image



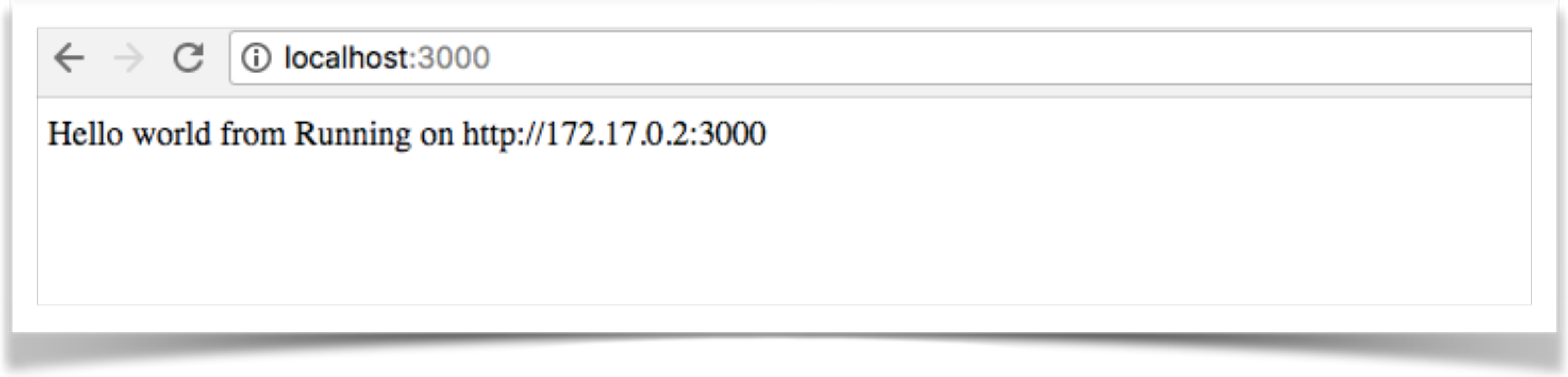
```
[ASTMACs-MacBook-Air:docker_file astmac$ docker images
REPOSITORY                     TAG                 IMAGE ID            CREATED             SIZE
burtwalsh/my_node_web_server   latest              0cbed0f04b45        14 seconds ago      507MB
ubuntu                         latest              0458a4468cbc        3 weeks ago         112MB
[ASTMACs-MacBook-Air:docker_file astmac$
[ASTMACs-MacBook-Air:docker_file astmac$
[ASTMACs-MacBook-Air:docker_file astmac$ docker run -d -p 3000:3000 0cbed0f04b45
d429ae035bf6af5ab3c6846a3edecd4c94d4d1274c878aed33ea9dcb5decab7c
[ASTMACs-MacBook-Air:docker_file astmac$
[ASTMACs-MacBook-Air:docker_file astmac$
[ASTMACs-MacBook-Air:docker_file astmac$ docker container ls
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS                    NAMES
d429ae035bf6        0cbed0f04b45        "npm start"         4 seconds ago       Up 4 seconds        0.0.0.0:3000->3000/tcp   determined_morse
[ASTMACs-MacBook-Air:docker_file astmac$
[ASTMACs-MacBook-Air:docker_file astmac$
ASTMACs-MacBook-Air:docker_file astmac$ █
```

Netstat to see the WebServer



```
root@d63099162436:/mywebapp# netstat -a
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.11:43351        *:*                     LISTEN
tcp        0      0 d63099162436:3000       *:*                     LISTEN
udp        0      0 127.0.0.11:39750        *:*
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags       Type       State         I-Node   Path
root@d63099162436:/mywebapp#
```

Accessing the WebServer

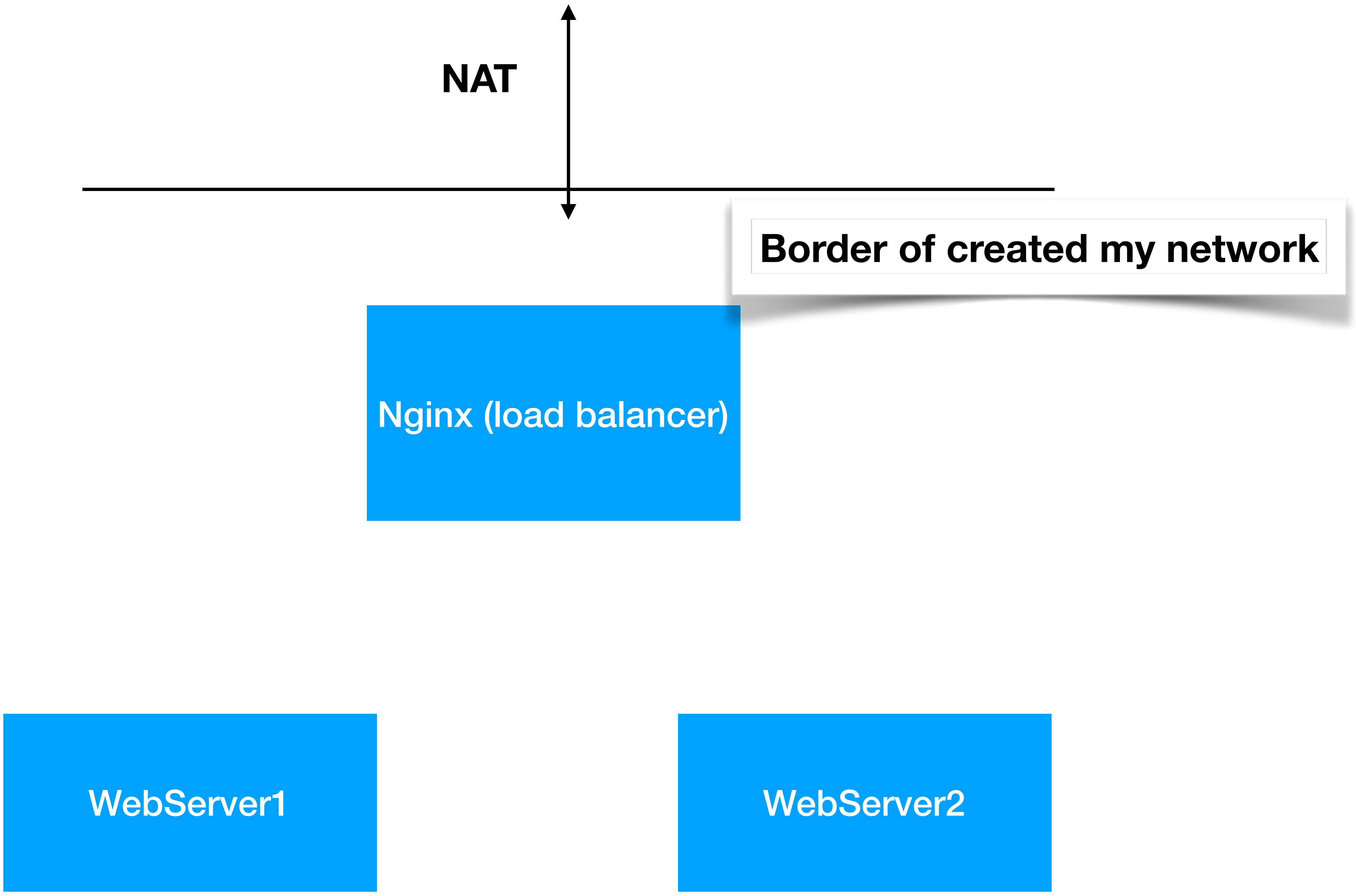


Hello world from Running on http://172.17.0.2:3000

Docker Compose

- Docker compose lets you define a group of containers and relationships between them
- You can bring up the group, restart the group, stop the group as a unit
- This is defined in a simple YAML file

Architecture



NAT

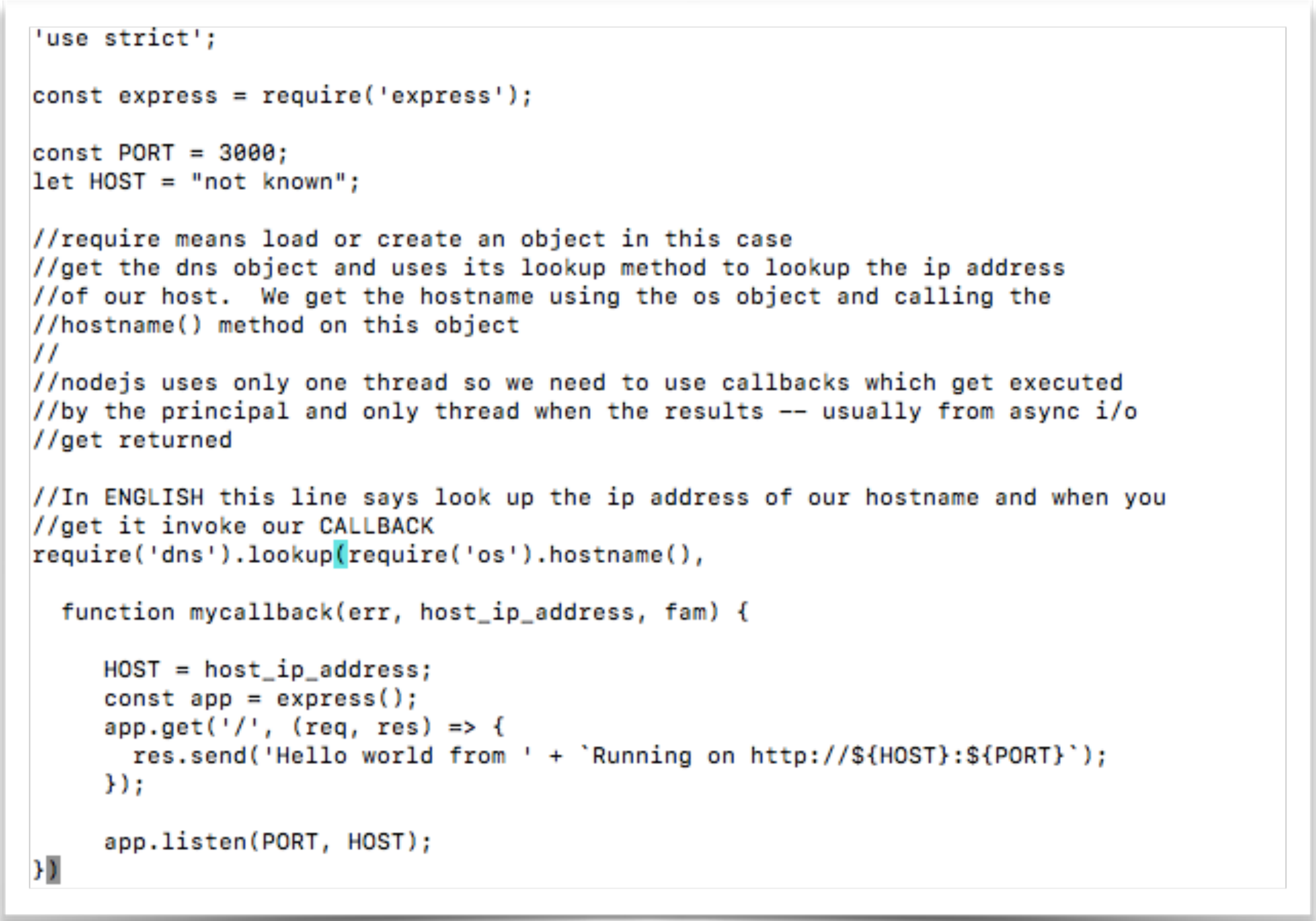
Border of created my network

Nginx (load balancer)

WebServer1

WebServer2

WebServer Code



```
'use strict';

const express = require('express');

const PORT = 3000;
let HOST = "not known";

//require means load or create an object in this case
//get the dns object and uses its lookup method to lookup the ip address
//of our host.  We get the hostname using the os object and calling the
//hostname() method on this object
//
//nodejs uses only one thread so we need to use callbacks which get executed
//by the principal and only thread when the results -- usually from async i/o
//get returned

//In ENGLISH this line says look up the ip address of our hostname and when you
//get it invoke our CALLBACK
require('dns').lookup(require('os').hostname(),

  function mycallback(err, host_ip_address, fam) {

     HOST = host_ip_address;
     const app = express();
     app.get('/', (req, res) => {
       res.send('Hello world from ' + `Running on http://${HOST}:${PORT}`);
     });

     app.listen(PORT, HOST);
})
```

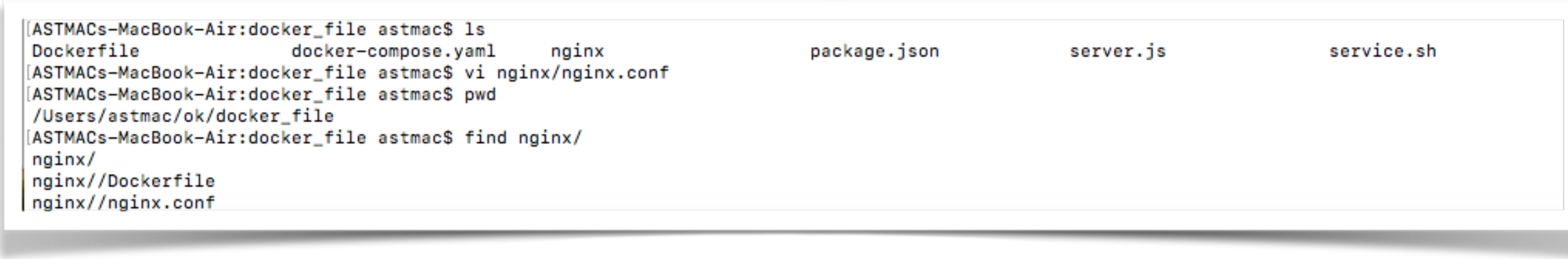
Nginx

- Will be a container like our webservers
- Acts as a load balancer (reverse proxy)
- Will shift requests between our two web servers

Note: A proxy hides the client. Servers think they are talking with the proxy not the client; the proxy makes requests on behalf of the client.

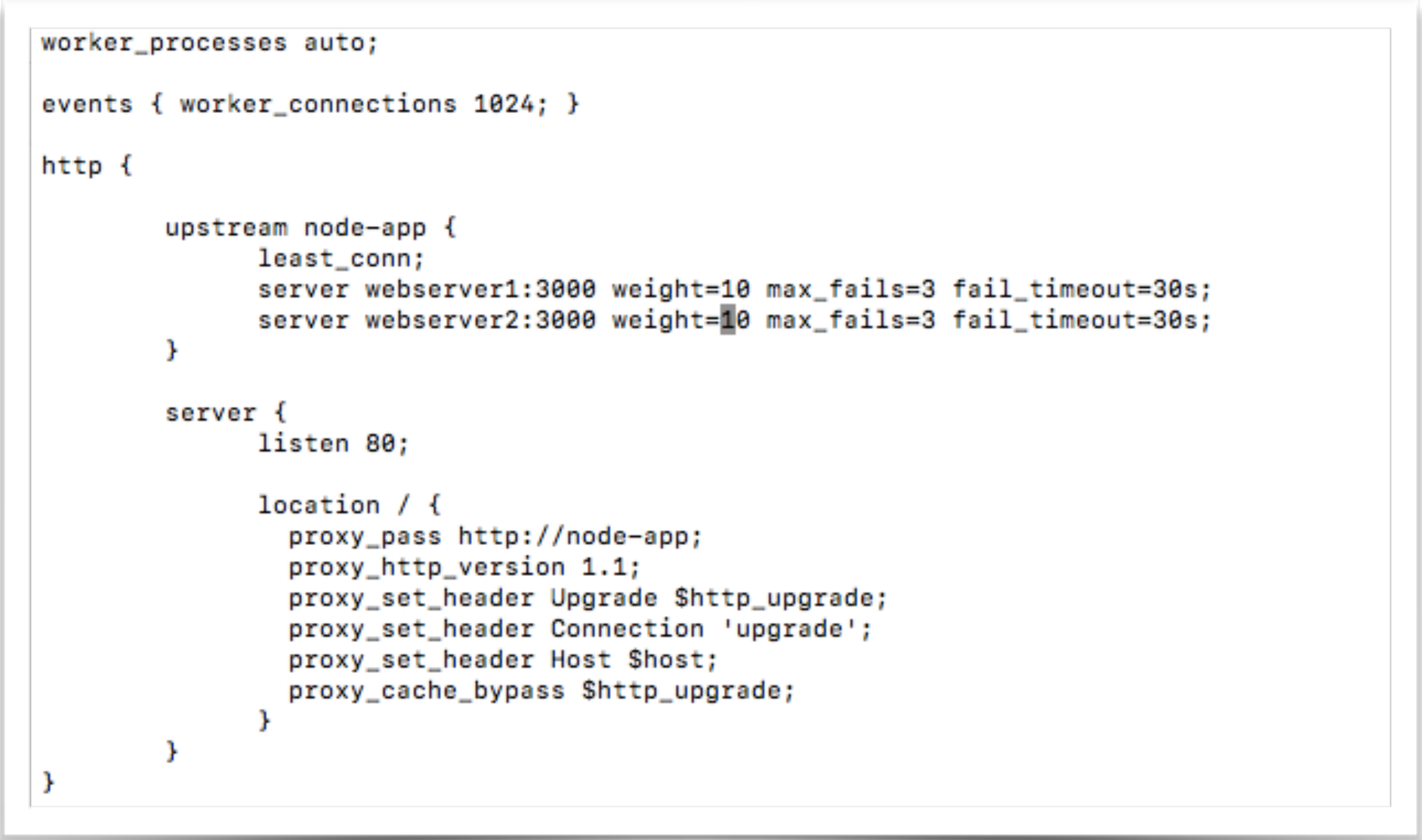
A reverse proxy hides servers the client talks to a server though the reverse proxy. The true server fielding the request is hidden from the client.

Directory for Nginx Resources



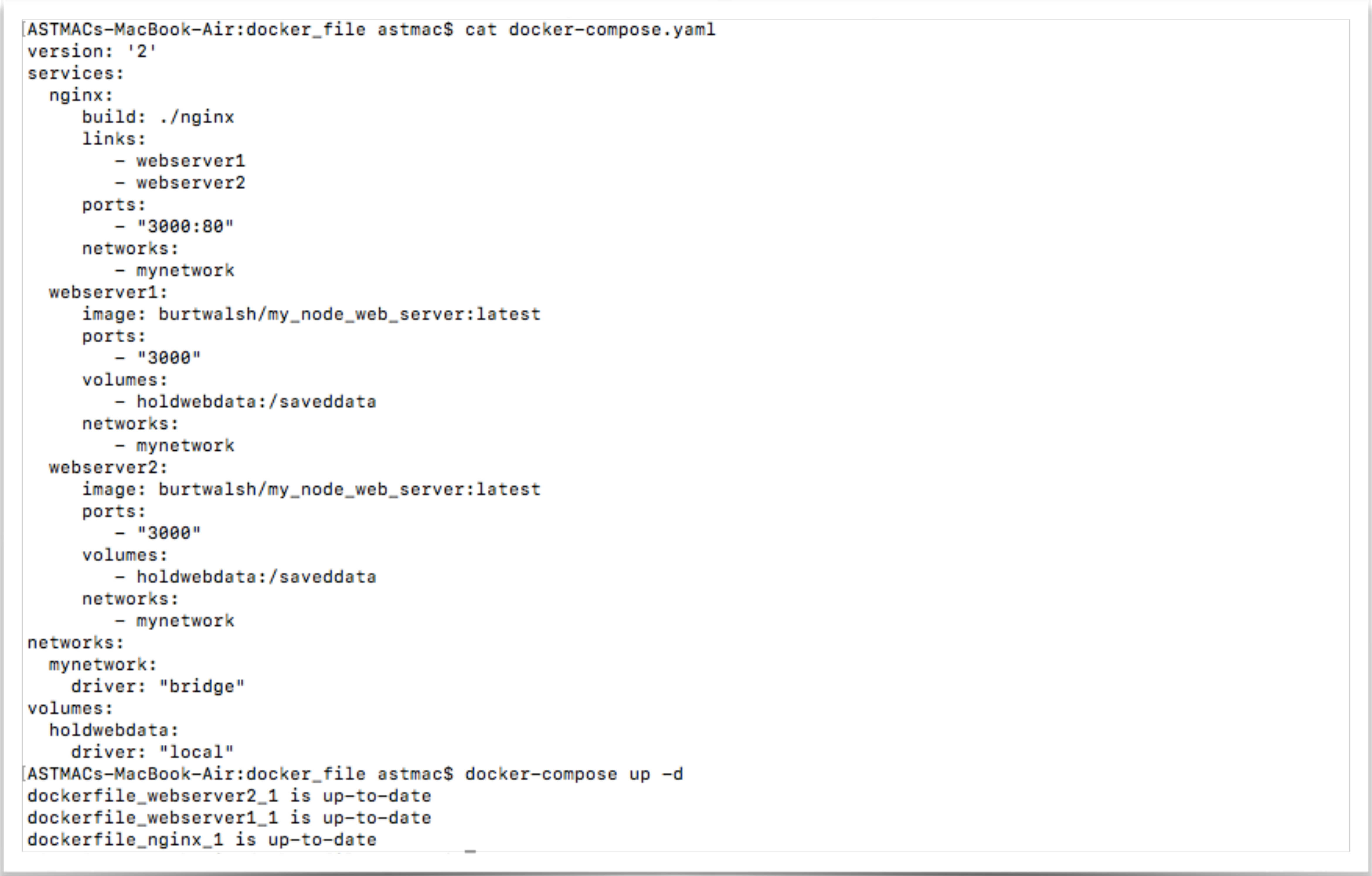
```
[ASTMACs-MacBook-Air:docker_file astmac$ ls
Dockerfile              docker-compose.yaml     nginx                   package.json            server.js               service.sh
[ASTMACs-MacBook-Air:docker_file astmac$ vi nginx/nginx.conf
[ASTMACs-MacBook-Air:docker_file astmac$ pwd
/Users/astmac/ok/docker_file
[ASTMACs-MacBook-Air:docker_file astmac$ find nginx/
nginx/
nginx//Dockerfile
nginx//nginx.conf
```

Nginx Config



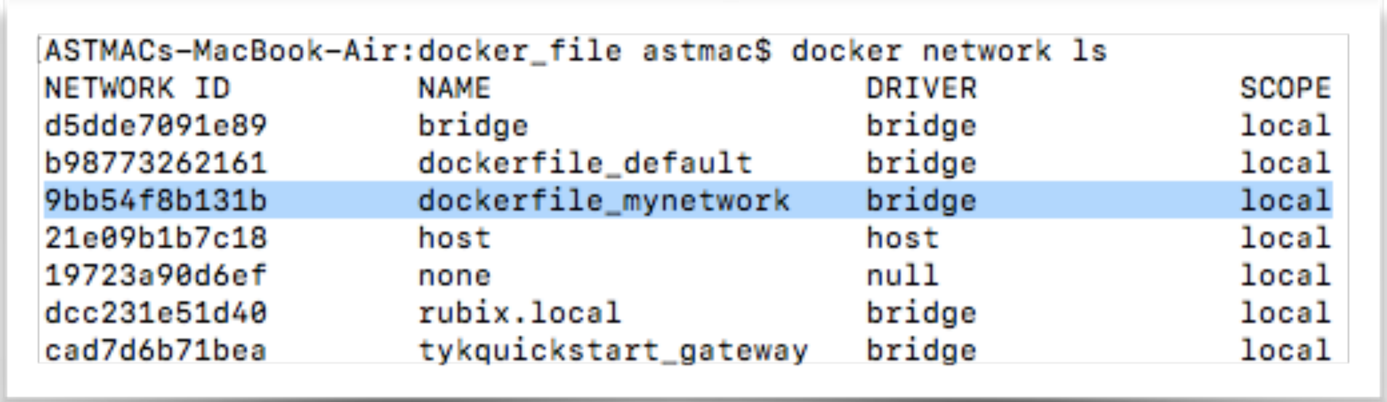
```
worker_processes auto;

events { worker_connections 1024; }

http {

        upstream node-app {
              least_conn;
              server webserver1:3000 weight=10 max_fails=3 fail_timeout=30s;
              server webserver2:3000 weight=10 max_fails=3 fail_timeout=30s;
        }

        server {
              listen 80;

              location / {
                proxy_pass http://node-app;
                proxy_http_version 1.1;
                proxy_set_header Upgrade $http_upgrade;
                proxy_set_header Connection 'upgrade';
                proxy_set_header Host $host;
                proxy_cache_bypass $http_upgrade;
              }
        }
}
```

Docker Compose File



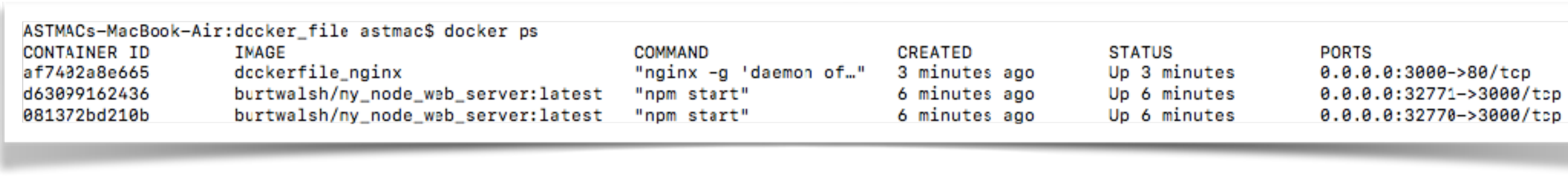
```
[ASTMACs-MacBook-Air:docker_file astmac$ cat docker-compose.yaml
version: '2'
services:
  nginx:
     build: ./nginx
     links:
        - webserver1
        - webserver2
     ports:
        - "3000:80"
     networks:
        - mynetwork
  webserver1:
     image: burtwalsh/my_node_web_server:latest
     ports:
        - "3000"
     volumes:
        - holdwebdata:/saveddata
     networks:
        - mynetwork
  webserver2:
     image: burtwalsh/my_node_web_server:latest
     ports:
        - "3000"
     volumes:
        - holdwebdata:/saveddata
     networks:
        - mynetwork
networks:
  mynetwork:
    driver: "bridge"
volumes:
  holdwebdata:
    driver: "local"
[ASTMACs-MacBook-Air:docker_file astmac$ docker-compose up -d
dockerfile_webserver2_1 is up-to-date
dockerfile_webserver1_1 is up-to-date
dockerfile_nginx_1 is up-to-date
```

Created Network



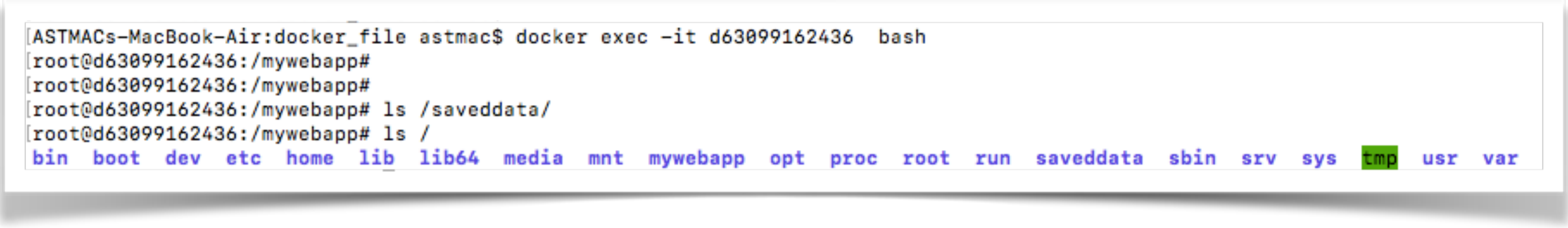
```
ASTMACs-MacBook-Air:docker_file astmac$ docker network ls
NETWORK ID          NAME                    DRIVER              SCOPE
d5dde7091e89        bridge                  bridge              local
b98773262161        dockerfile_default      bridge              local
9bb54f8b131b        dockerfile_mynetwork    bridge              local
21e09b1b7c18        host                    host                local
19723a90d6ef        none                    null                local
dcc231e51d40        rubix.local             bridge              local
cad7d6b71bea        tykquickstart_gateway   bridge              local
```

The containers



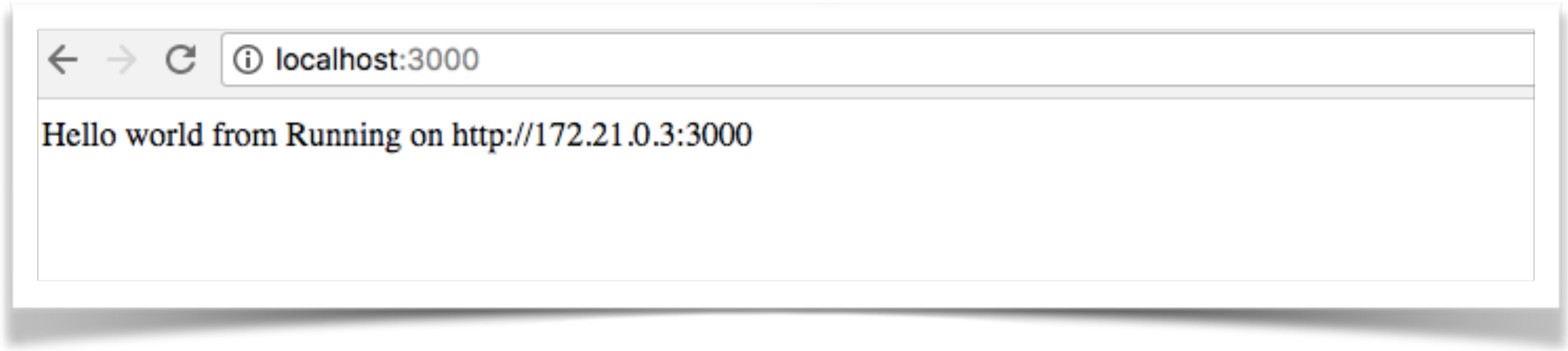
```
ASTMACs-MacBook-Air:docker_file astmac$ docker ps
CONTAINER ID        IMAGE                                 COMMAND                  CREATED             STATUS              PORTS
af7402a8e665        dockerfile_nginx                      "nginx -g 'daemon of…"   3 minutes ago       Up 3 minutes        0.0.0.0:3000->80/tcp
d63099162436        burtwalsh/my_node_web_server:latest   "npm start"              6 minutes ago       Up 6 minutes        0.0.0.0:32771->3000/tcp
081372bd210b        burtwalsh/my_node_web_server:latest   "npm start"              6 minutes ago       Up 6 minutes        0.0.0.0:32770->3000/tcp
```

Mounted Volumes

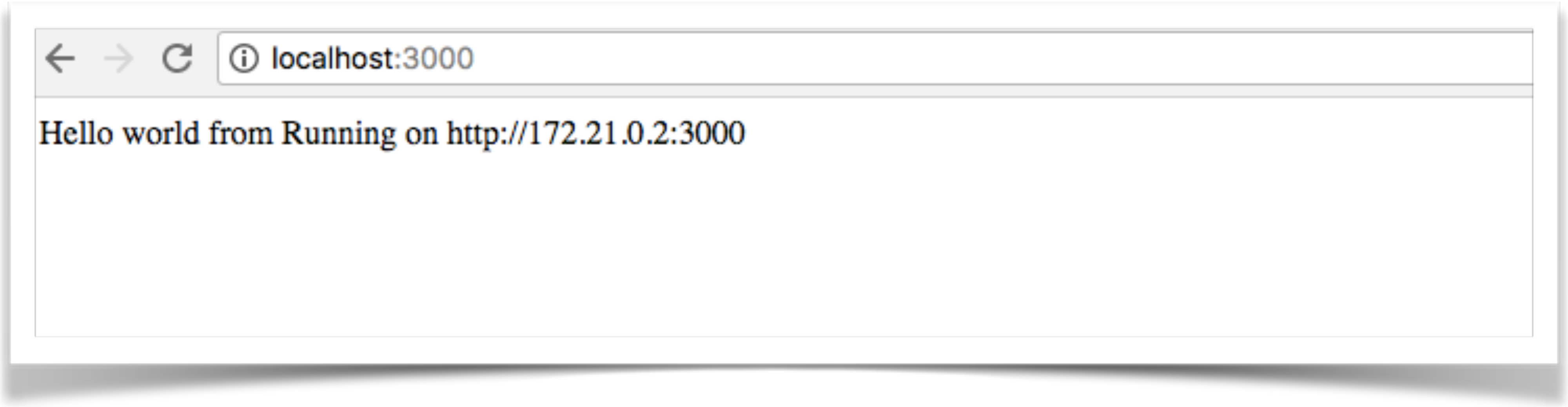


```
[ASTMACs-MacBook-Air:docker_file astmac$ docker exec -it d63099162436  bash
[root@d63099162436:/mywebapp#
[root@d63099162436:/mywebapp#
[root@d63099162436:/mywebapp# ls /saveddata/
[root@d63099162436:/mywebapp# ls /
bin  boot  dev  etc  home  lib  lib64  media  mnt  mywebapp  opt  proc  root  run  saveddata  sbin  srv  sys  tmp  usr  var
```

Alternation between WebServers



Hello world from Running on http://172.21.0.3:3000



Hello world from Running on http://172.21.0.2:3000

References

- Three tier web application with docker compose (https://serversforhackers.com/dockerized-app/docker-compose)
- Great tutorial for around $10 (https://www.udemy.com/docker-mastery/)
- Main docker site (https://www.docker.com/)
- Docker hub for publishing and pulling (https://hub.docker.com/)

Other Things to Look At

- Docker Swarm (https://docs.docker.com/engine/swarm/)
- Nodejs (https://nodejs.org/en/)
- Source code and slides (https://github.com/burtwalsh/dockertutorial)