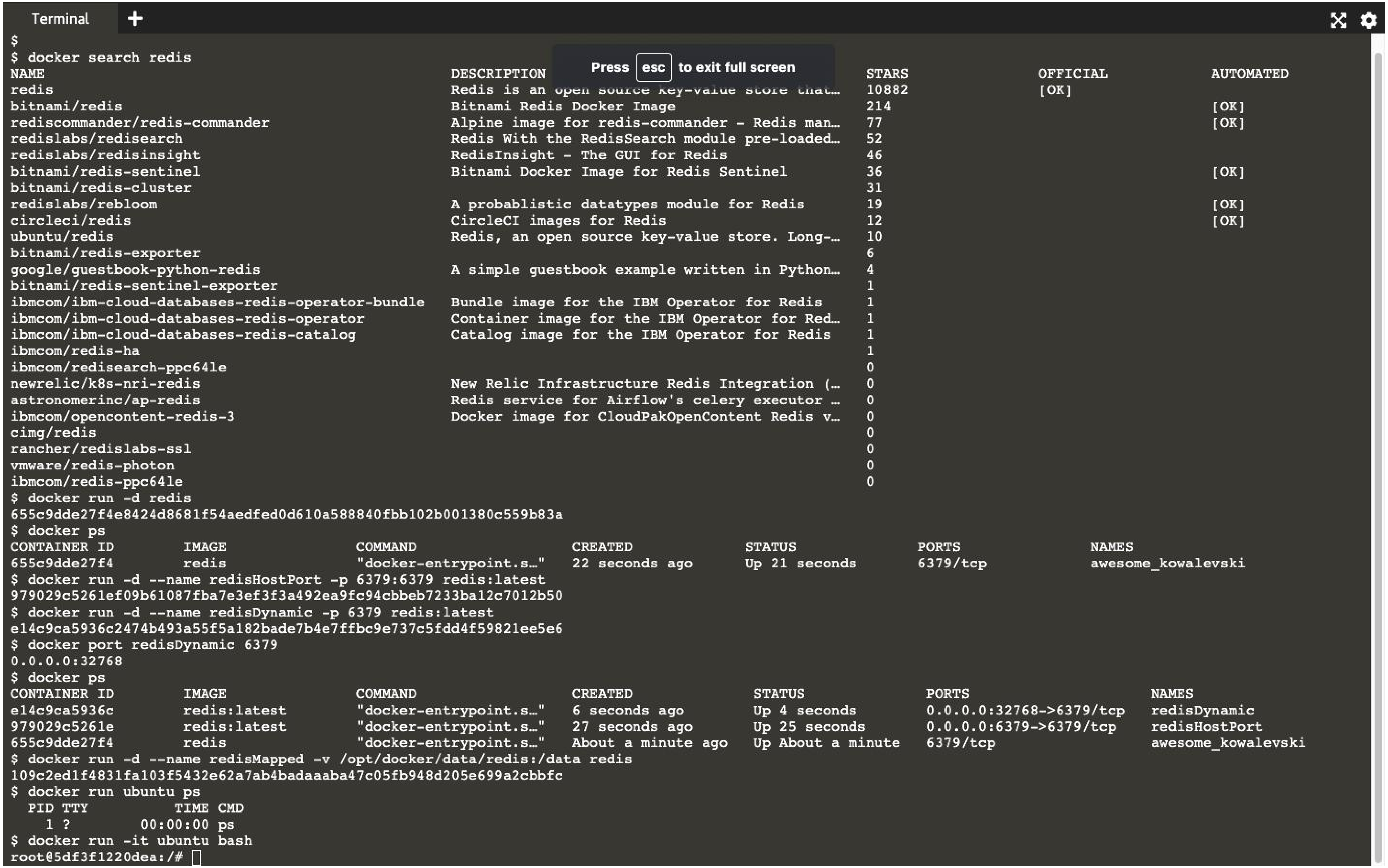
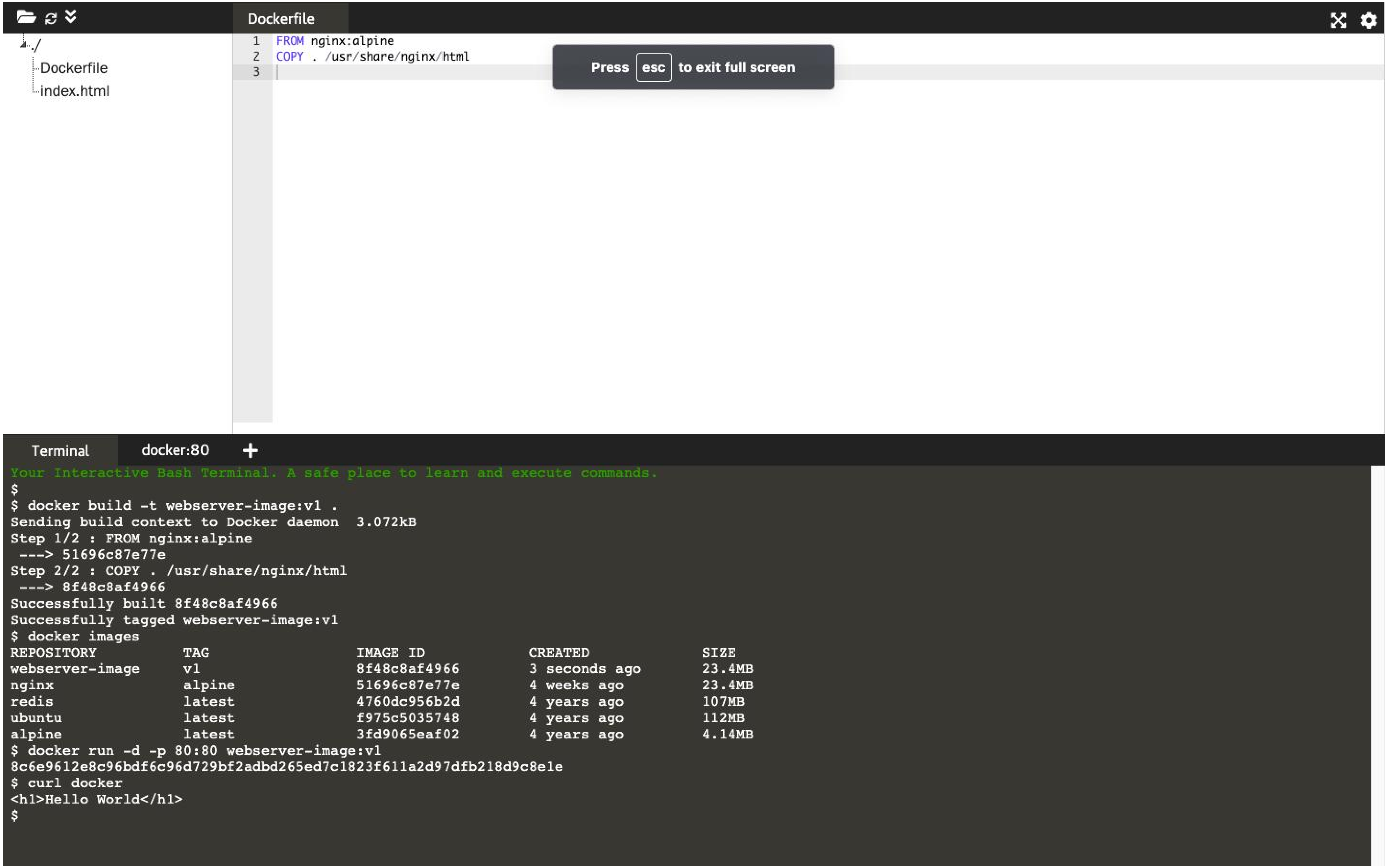
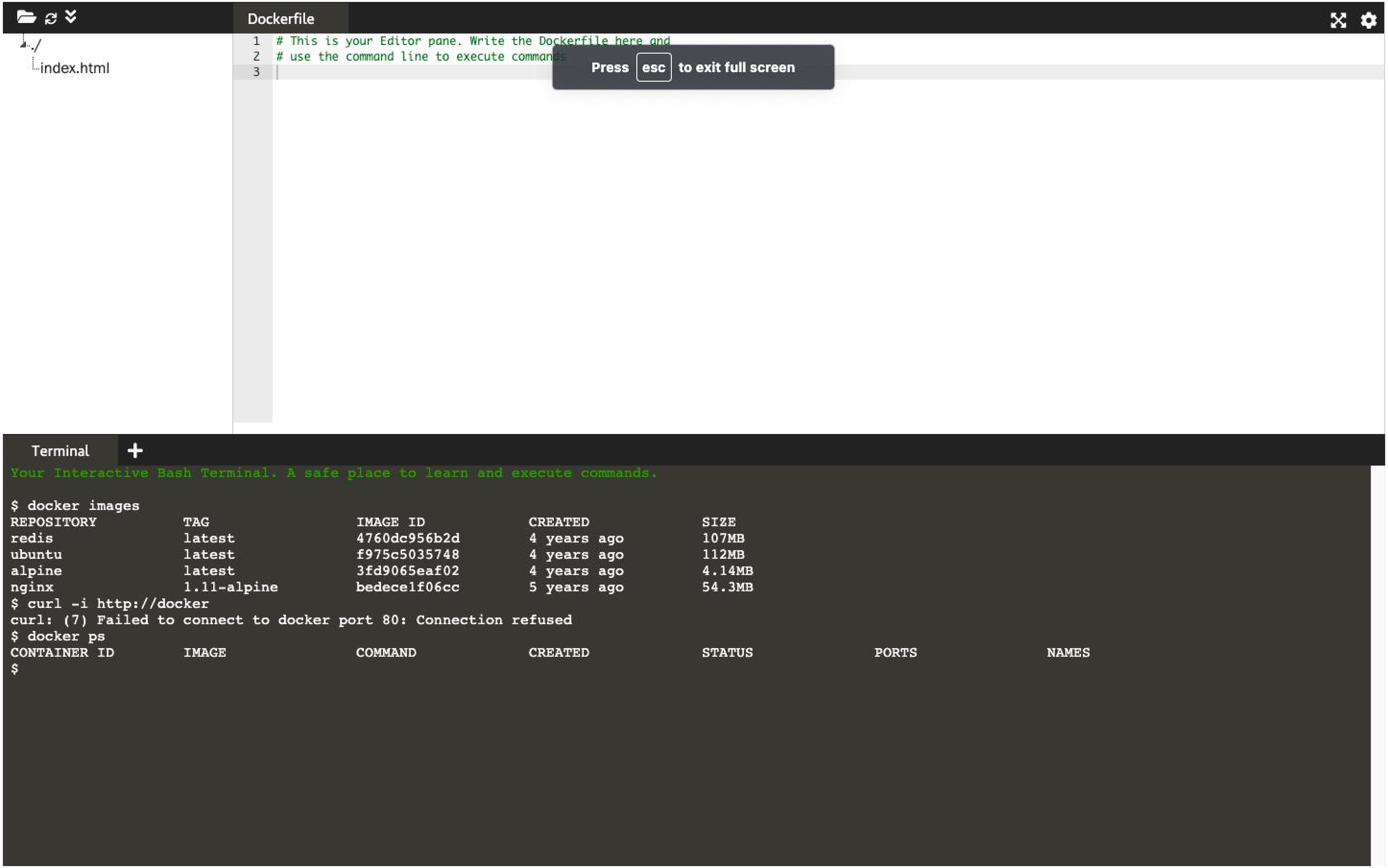


```
$
$ docker search redis
NAME                                               DESCRIPTION                                     STARS               OFFICIAL            AUTOMATED
redis                                              Redis is an open source key-value store that…   10882               [OK]
bitnami/redis                                      Bitnami Redis Docker Image                      214                                     [OK]
rediscommander/redis-commander                     Alpine image for redis-commander - Redis man…   77                                      [OK]
redislabs/redisearch                               Redis With the RedisSearch module pre-loaded…   52
redislabs/redisinsight                             RedisInsight - The GUI for Redis                46
bitnami/redis-sentinel                             Bitnami Docker Image for Redis Sentinel         36                                      [OK]
bitnami/redis-cluster                                                                              31
redislabs/rebloom                                  A probablistic datatypes module for Redis       19                                      [OK]
circleci/redis                                     CircleCI images for Redis                       12                                      [OK]
ubuntu/redis                                       Redis, an open source key-value store. Long-…   10
bitnami/redis-exporter                                                                             6
google/guestbook-python-redis                      A simple guestbook example written in Python…   4
bitnami/redis-sentinel-exporter                                                                    1
ibmcom/ibm-cloud-databases-redis-operator-bundle   Bundle image for the IBM Operator for Redis     1
ibmcom/ibm-cloud-databases-redis-operator          Container image for the IBM Operator for Red…   1
ibmcom/ibm-cloud-databases-redis-catalog           Catalog image for the IBM Operator for Redis    1
ibmcom/redis-ha                                                                                    1
ibmcom/redisearch-ppc64le                                                                          0
newrelic/k8s-nri-redis                             New Relic Infrastructure Redis Integration (…   0
astronomerinc/ap-redis                             Redis service for Airflow's celery executor …   0
ibmcom/opencontent-redis-3                         Docker image for CloudPakOpenContent Redis v…   0
cimg/redis                                                                                         0
rancher/redislabs-ssl                                                                              0
vmware/redis-photon                                                                                0
ibmcom/redis-ppc64le                                                                               0
$ docker run -d redis
655c9dde27f4e8424d8681f54aedfed0d610a588840fbb102b001380c559b83a
$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS               NAMES
655c9dde27f4        redis               "docker-entrypoint.s…"   22 seconds ago      Up 21 seconds       6379/tcp            awesome_kowalevski
$ docker run -d --name redisHostPort -p 6379:6379 redis:latest
979029c5261ef09b61087fba7e3ef3f3a492ea9fc94cbbeb7233ba12c7012b50
$ docker run -d --name redisDynamic -p 6379 redis:latest
e14c9ca5936c2474b493a55f5a182bade7b4e7ffbc9e737c5fdd4f59821ee5e6
$ docker port redisDynamic 6379
0.0.0.0:32768
$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED              STATUS              PORTS                      NAMES
e14c9ca5936c        redis:latest        "docker-entrypoint.s…"   6 seconds ago        Up 4 seconds        0.0.0.0:32768->6379/tcp    redisDynamic
979029c5261e        redis:latest        "docker-entrypoint.s…"   27 seconds ago       Up 25 seconds       0.0.0.0:6379->6379/tcp     redisHostPort
655c9dde27f4        redis               "docker-entrypoint.s…"   About a minute ago   Up About a minute   6379/tcp                   awesome_kowalevski
$ docker run -d --name redisMapped -v /opt/docker/data/redis:/data redis
109c2ed1f4831fa103f5432e62a7ab4badaaaba47c05fb948d205e699a2cbbfc
$ docker run ubuntu ps
  PID TTY          TIME CMD
    1 ?        00:00:00 ps
$ docker run -it ubuntu bash
root@5df3f1220dea:/# 
```



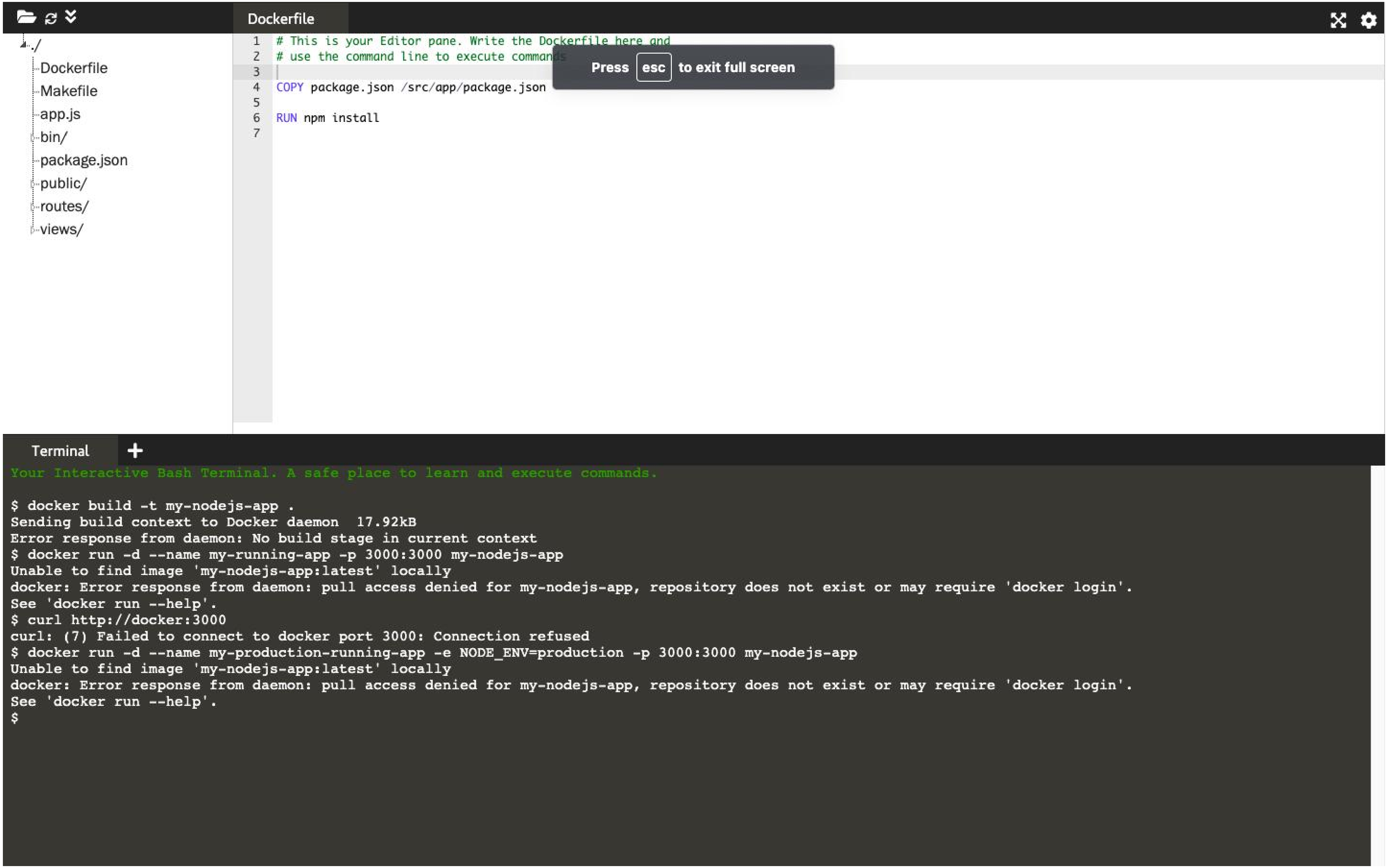
```
FROM nginx:alpine
COPY . /usr/share/nginx/html
```

```
Your Interactive Bash Terminal. A safe place to learn and execute commands.
$
$ docker build -t webserver-image:v1 .
Sending build context to Docker daemon  3.072kB
Step 1/2 : FROM nginx:alpine
 ---> 51696c87e77e
Step 2/2 : COPY . /usr/share/nginx/html
 ---> 8f48c8af4966
Successfully built 8f48c8af4966
Successfully tagged webserver-image:v1
$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
webserver-image     v1                  8f48c8af4966        3 seconds ago       23.4MB
nginx               alpine              51696c87e77e        4 weeks ago         23.4MB
redis               latest              4760dc956b2d        4 years ago         107MB
ubuntu              latest              f975c5035748        4 years ago         112MB
alpine              latest              3fd9065eaf02        4 years ago         4.14MB
$ docker run -d -p 80:80 webserver-image:v1
8c6e9612e8c96bdf6c96d729bf2adbd265ed7c1823f611a2d97dfb218d9c8e1e
$ curl docker
<h1>Hello World</h1>
$
```



```
# This is your Editor pane. Write the Dockerfile here and
# use the command line to execute commands
```

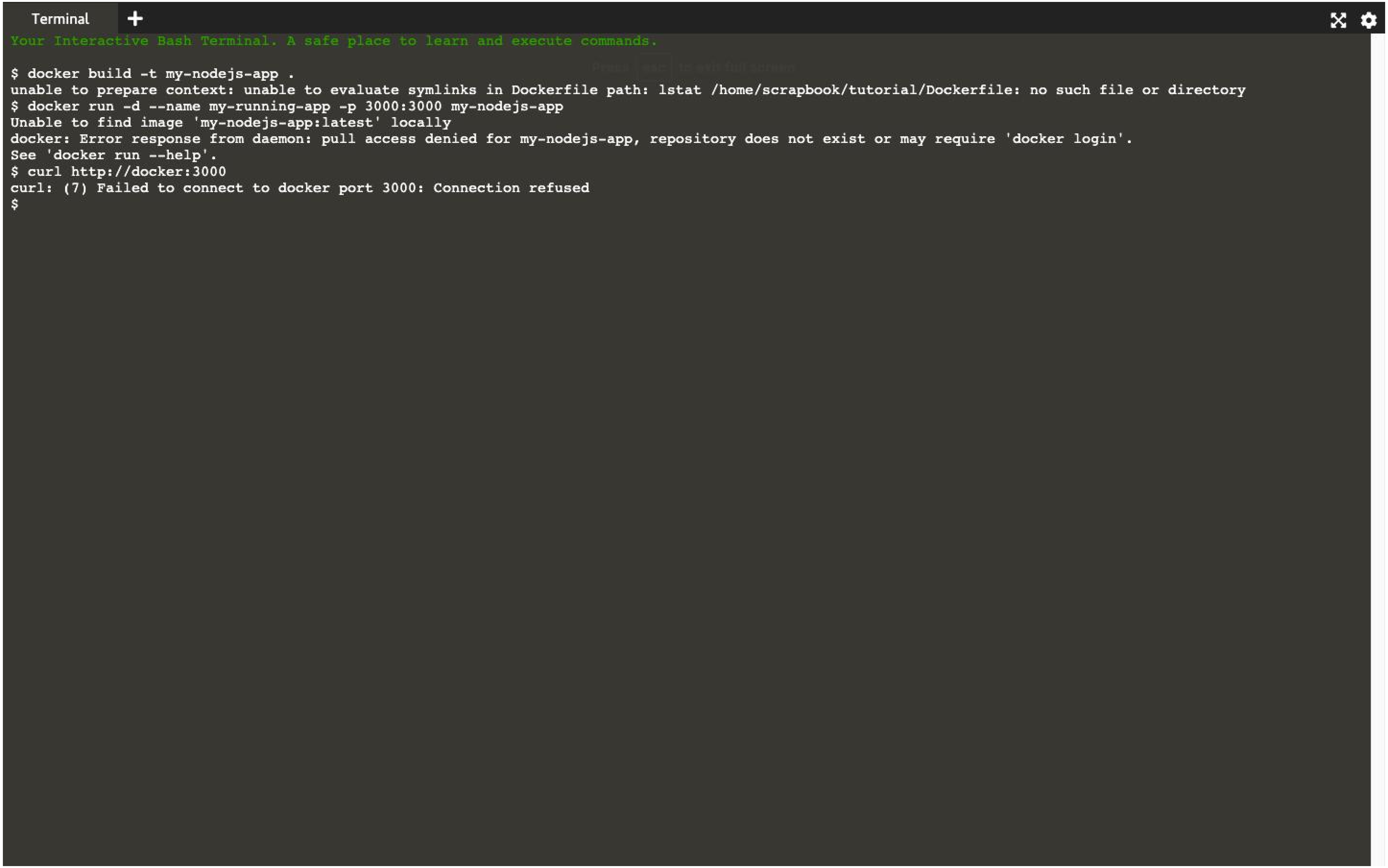
```
Your Interactive Bash Terminal. A safe place to learn and execute commands.

$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
redis               latest              4760dc956b2d        4 years ago         107MB
ubuntu              latest              f975c5035748        4 years ago         112MB
alpine              latest              3fd9065eaf02        4 years ago         4.14MB
nginx               1.11-alpine         bedece1f06cc        5 years ago         54.3MB
$ curl -i http://docker
curl: (7) Failed to connect to docker port 80: Connection refused
$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
$
```



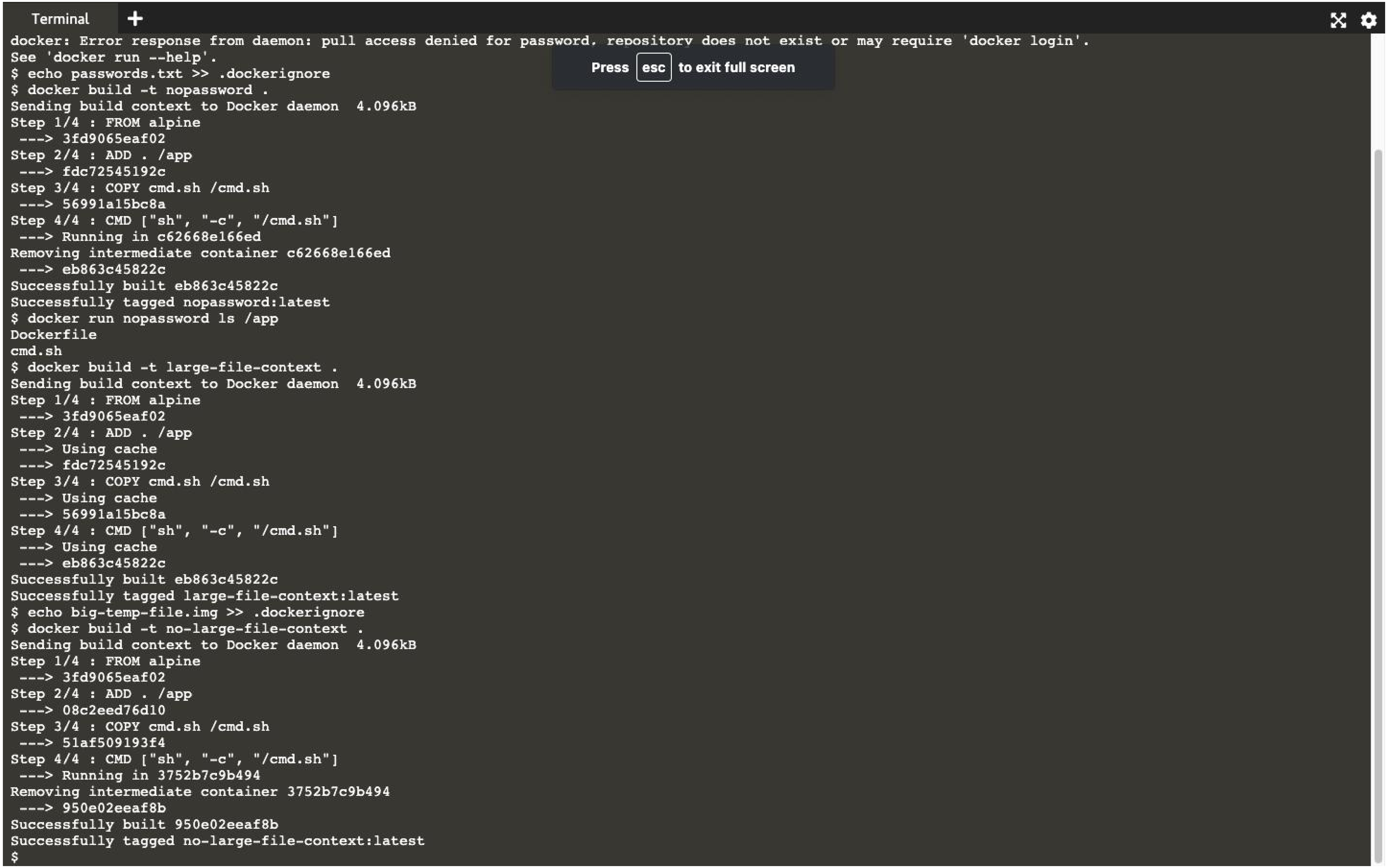
Press esc to exit full screen

```
# This is your Editor pane. Write the Dockerfile here and
# use the command line to execute commands

COPY package.json /src/app/package.json

RUN npm install

```

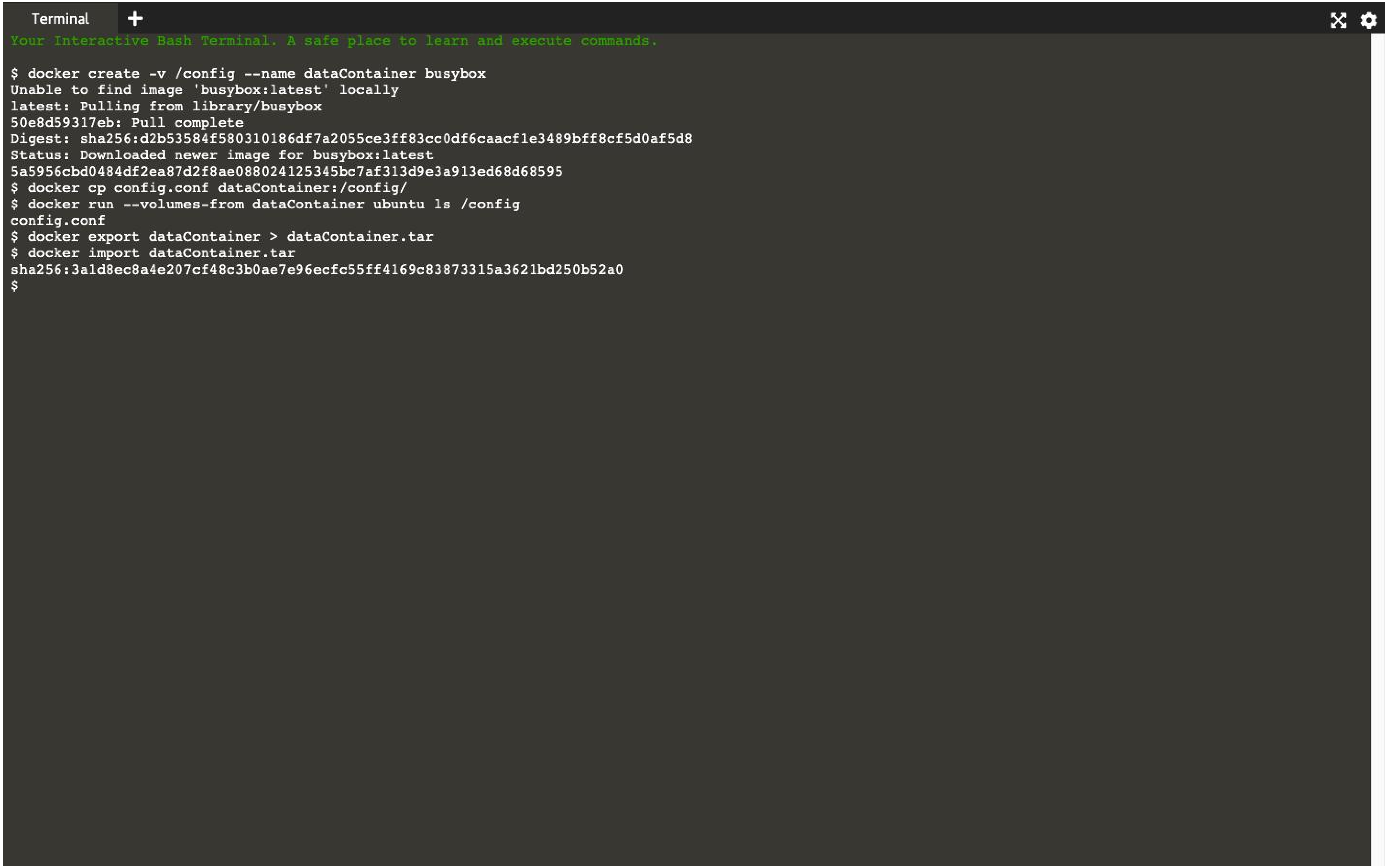
```
Your Interactive Bash Terminal. A safe place to learn and execute commands.

$ docker build -t my-nodejs-app .
Sending build context to Docker daemon  17.92kB
Error response from daemon: No build stage in current context
$ docker run -d --name my-running-app -p 3000:3000 my-nodejs-app
Unable to find image 'my-nodejs-app:latest' locally
docker: Error response from daemon: pull access denied for my-nodejs-app, repository does not exist or may require 'docker login'.
See 'docker run --help'.
$ curl http://docker:3000
curl: (7) Failed to connect to docker port 3000: Connection refused
$ docker run -d --name my-production-running-app -e NODE_ENV=production -p 3000:3000 my-nodejs-app
Unable to find image 'my-nodejs-app:latest' locally
docker: Error response from daemon: pull access denied for my-nodejs-app, repository does not exist or may require 'docker login'.
See 'docker run --help'.
$
```



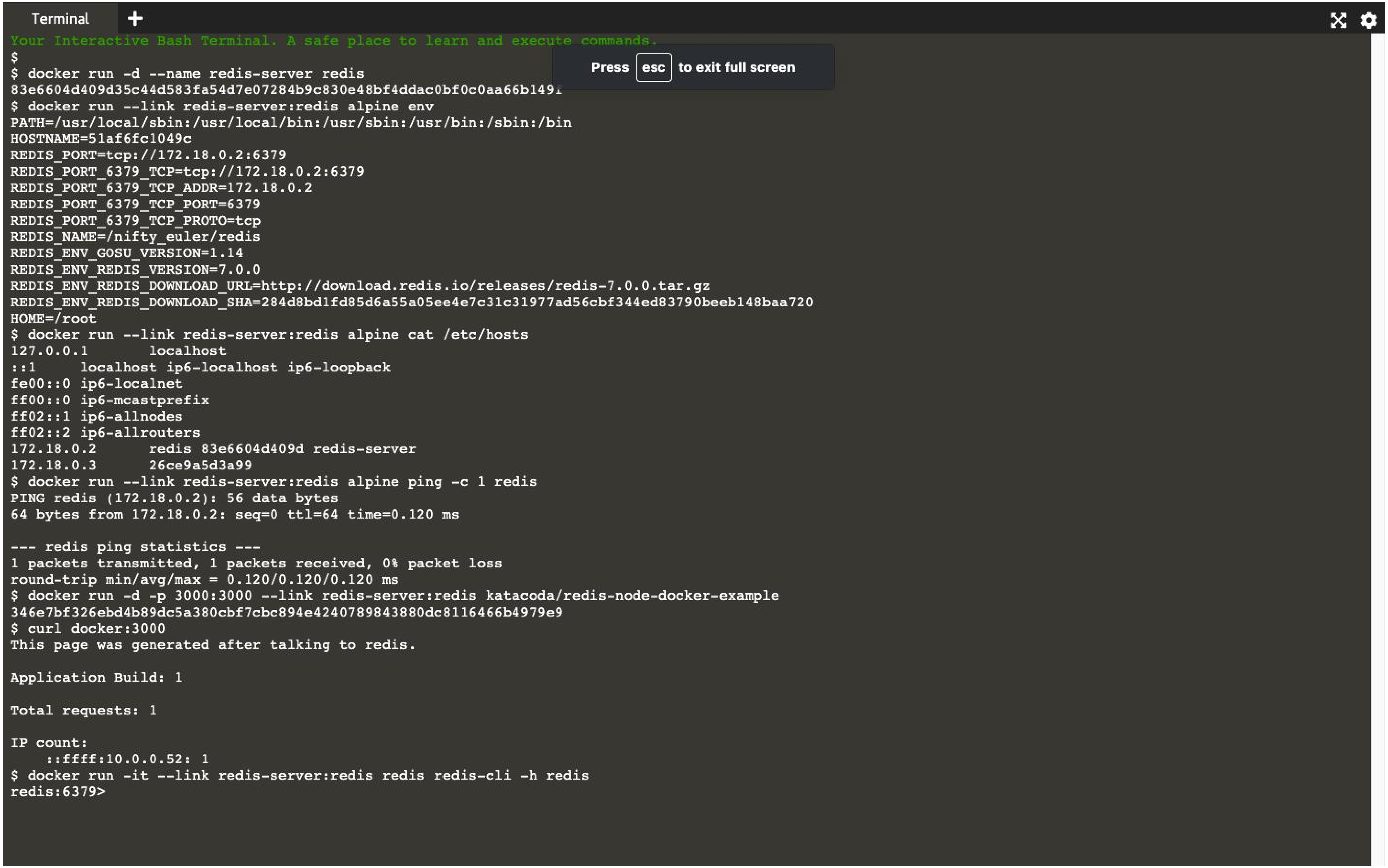
```
Your Interactive Bash Terminal. A safe place to learn and execute commands.

$ docker build -t my-nodejs-app .
unable to prepare context: unable to evaluate symlinks in Dockerfile path: lstat /home/scrapbook/tutorial/Dockerfile: no such file or directory
$ docker run -d --name my-running-app -p 3000:3000 my-nodejs-app
Unable to find image 'my-nodejs-app:latest' locally
docker: Error response from daemon: pull access denied for my-nodejs-app, repository does not exist or may require 'docker login'.
See 'docker run --help'.
$ curl http://docker:3000
curl: (7) Failed to connect to docker port 3000: Connection refused
$
```



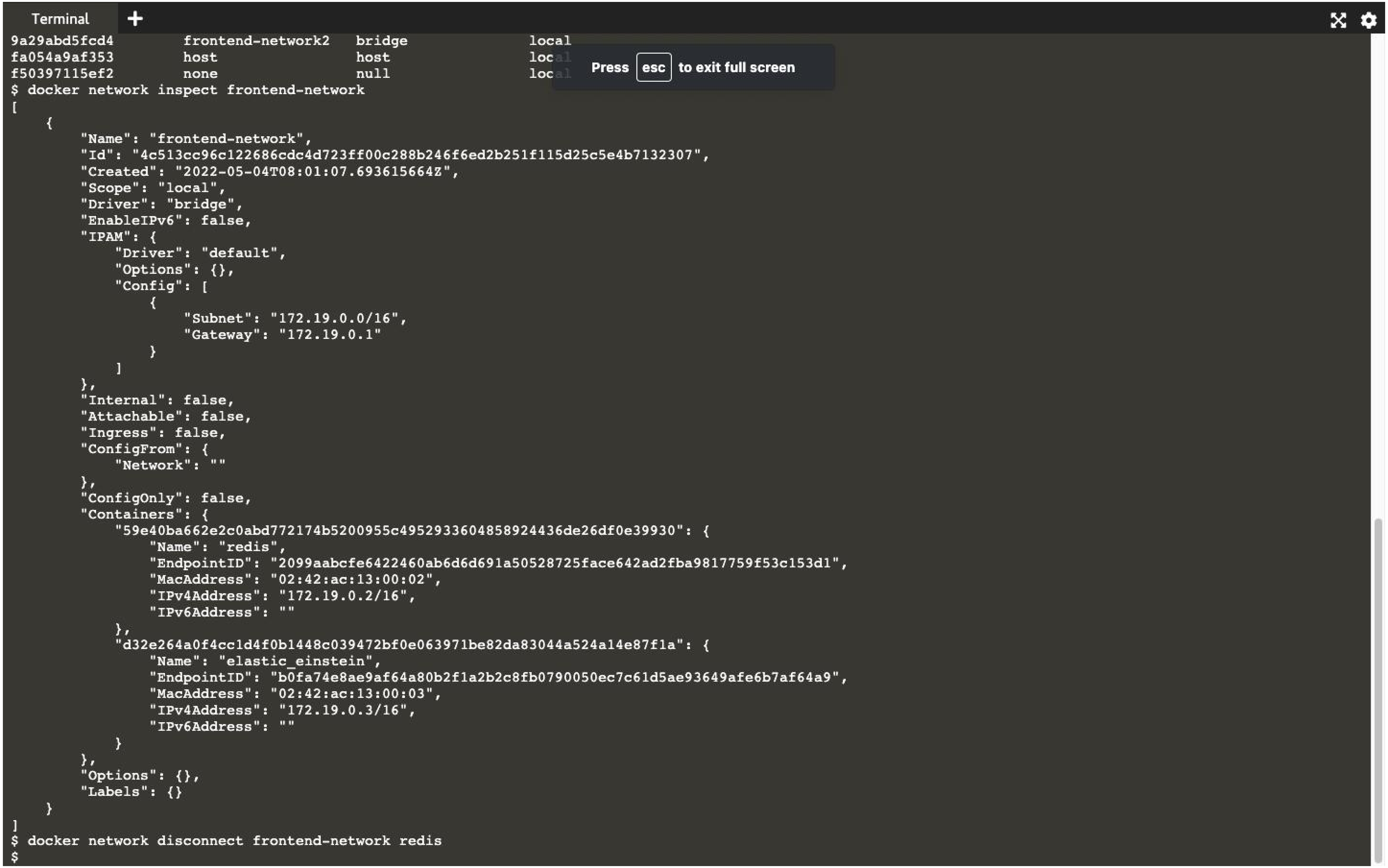
```
docker: Error response from daemon: pull access denied for password, repository does not exist or may require 'docker login'.
See 'docker run --help'.
$ echo passwords.txt >> .dockerignore
$ docker build -t nopassword .
Sending build context to Docker daemon  4.096kB
Step 1/4 : FROM alpine
 ---> 3fd9065eaf02
Step 2/4 : ADD . /app
 ---> fdc72545192c
Step 3/4 : COPY cmd.sh /cmd.sh
 ---> 56991a15bc8a
Step 4/4 : CMD ["sh", "-c", "/cmd.sh"]
 ---> Running in c62668e166ed
Removing intermediate container c62668e166ed
 ---> eb863c45822c
Successfully built eb863c45822c
Successfully tagged nopassword:latest
$ docker run nopassword ls /app
Dockerfile
cmd.sh
$ docker build -t large-file-context .
Sending build context to Docker daemon  4.096kB
Step 1/4 : FROM alpine
 ---> 3fd9065eaf02
Step 2/4 : ADD . /app
 ---> Using cache
 ---> fdc72545192c
Step 3/4 : COPY cmd.sh /cmd.sh
 ---> Using cache
 ---> 56991a15bc8a
Step 4/4 : CMD ["sh", "-c", "/cmd.sh"]
 ---> Using cache
 ---> eb863c45822c
Successfully built eb863c45822c
Successfully tagged large-file-context:latest
$ echo big-temp-file.img >> .dockerignore
$ docker build -t no-large-file-context .
Sending build context to Docker daemon  4.096kB
Step 1/4 : FROM alpine
 ---> 3fd9065eaf02
Step 2/4 : ADD . /app
 ---> 08c2eed76d10
Step 3/4 : COPY cmd.sh /cmd.sh
 ---> 51af509193f4
Step 4/4 : CMD ["sh", "-c", "/cmd.sh"]
 ---> Running in 3752b7c9b494
Removing intermediate container 3752b7c9b494
 ---> 950e02eeaf8b
Successfully built 950e02eeaf8b
Successfully tagged no-large-file-context:latest
$
```



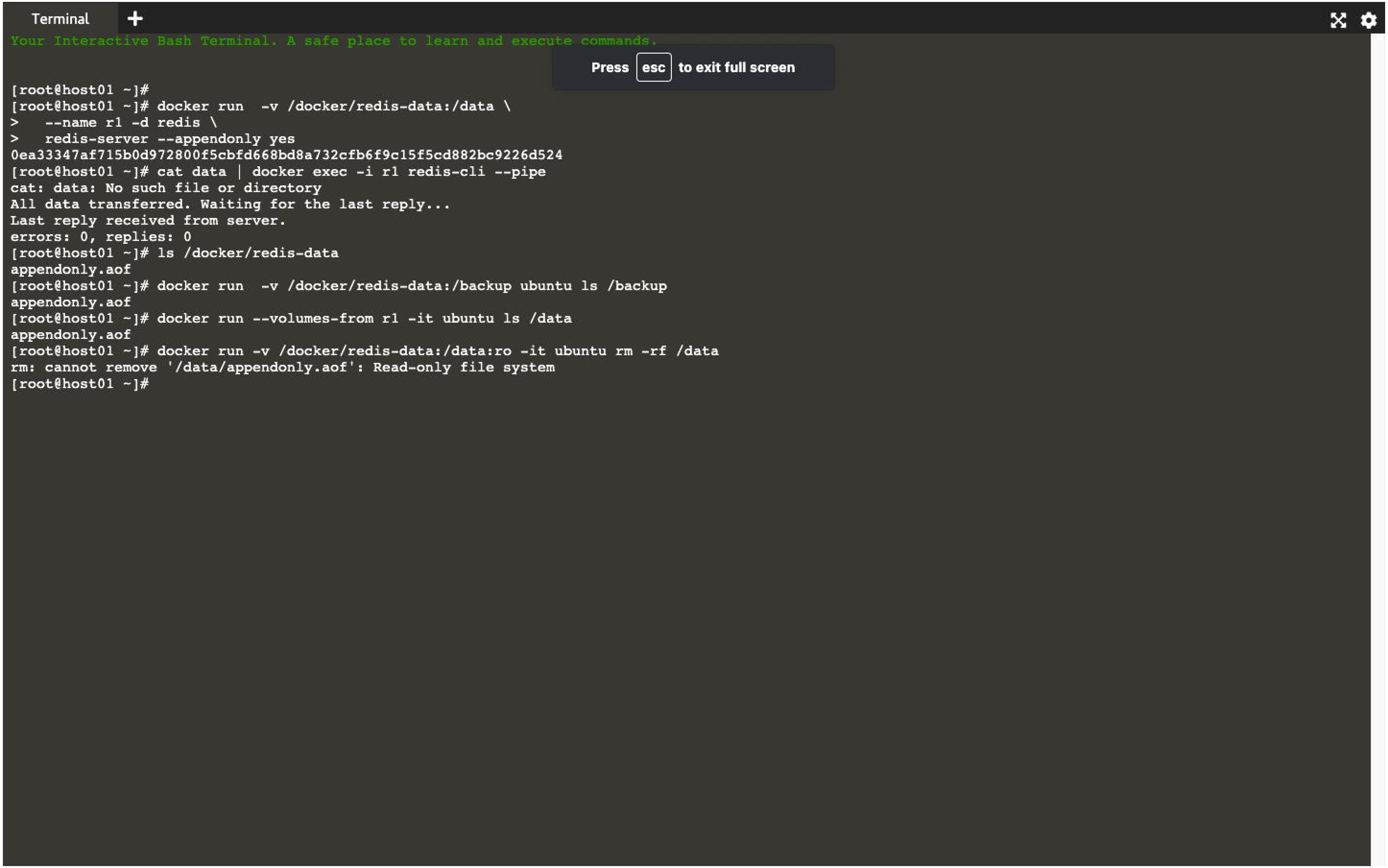
```
Your Interactive Bash Terminal. A safe place to learn and execute commands.

$ docker create -v /config --name dataContainer busybox
Unable to find image 'busybox:latest' locally
latest: Pulling from library/busybox
50e8d59317eb: Pull complete
Digest: sha256:d2b53584f580310186df7a2055ce3ff83cc0df6caacf1e3489bff8cf5d0af5d8
Status: Downloaded newer image for busybox:latest
5a5956cbd0484df2ea87d2f8ae088024125345bc7af313d9e3a913ed68d68595
$ docker cp config.conf dataContainer:/config/
$ docker run --volumes-from dataContainer ubuntu ls /config
config.conf
$ docker export dataContainer > dataContainer.tar
$ docker import dataContainer.tar
sha256:3a1d8ec8a4e207cf48c3b0ae7e96ecfc55ff4169c83873315a3621bd250b52a0
$
```



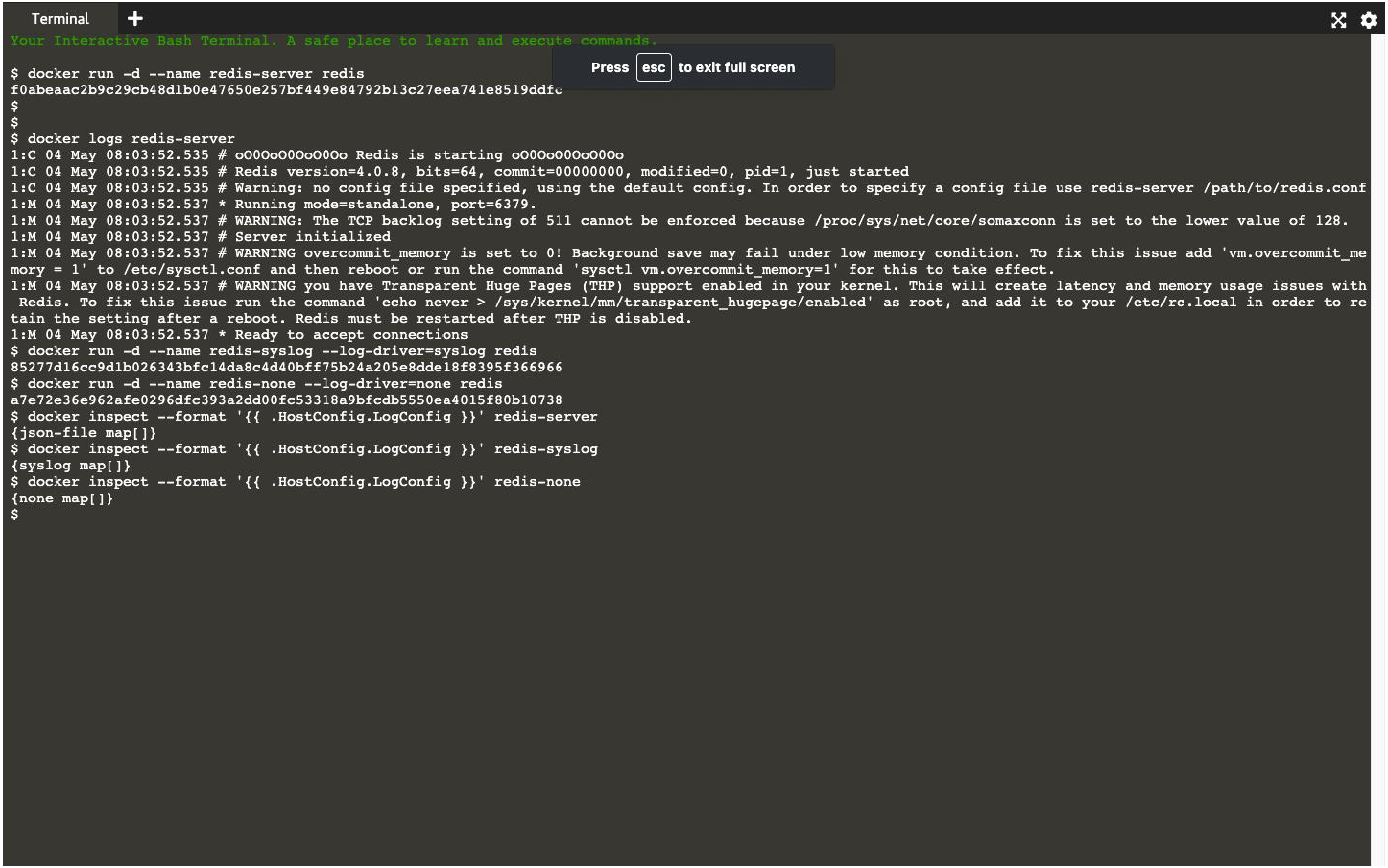
```
Your Interactive Bash Terminal. A safe place to learn and execute commands.
$
$ docker run -d --name redis-server redis
83e6604d409d35c44d583fa54d7e07284b9c830e48bf4ddac0bf0c0aa66b1491
$ docker run --link redis-server:redis alpine env
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOSTNAME=51af6fc1049c
REDIS_PORT=tcp://172.18.0.2:6379
REDIS_PORT_6379_TCP=tcp://172.18.0.2:6379
REDIS_PORT_6379_TCP_ADDR=172.18.0.2
REDIS_PORT_6379_TCP_PORT=6379
REDIS_PORT_6379_TCP_PROTO=tcp
REDIS_NAME=/nifty_euler/redis
REDIS_ENV_GOSU_VERSION=1.14
REDIS_ENV_REDIS_VERSION=7.0.0
REDIS_ENV_REDIS_DOWNLOAD_URL=http://download.redis.io/releases/redis-7.0.0.tar.gz
REDIS_ENV_REDIS_DOWNLOAD_SHA=284d8bd1fd85d6a55a05ee4e7c31c31977ad56cbf344ed83790beeb148baa720
HOME=/root
$ docker run --link redis-server:redis alpine cat /etc/hosts
127.0.0.1       localhost
::1     localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
172.18.0.2      redis 83e6604d409d redis-server
172.18.0.3      26ce9a5d3a99
$ docker run --link redis-server:redis alpine ping -c 1 redis
PING redis (172.18.0.2): 56 data bytes
64 bytes from 172.18.0.2: seq=0 ttl=64 time=0.120 ms

--- redis ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 0.120/0.120/0.120 ms
$ docker run -d -p 3000:3000 --link redis-server:redis katacoda/redis-node-docker-example
346e7bf326ebd4b89dc5a380cbf7cbc894e4240789843880dc8116466b4979e9
$ curl docker:3000
This page was generated after talking to redis.

Application Build: 1

Total requests: 1

IP count:
    ::ffff:10.0.0.52: 1
$ docker run -it --link redis-server:redis redis redis-cli -h redis
redis:6379>
```



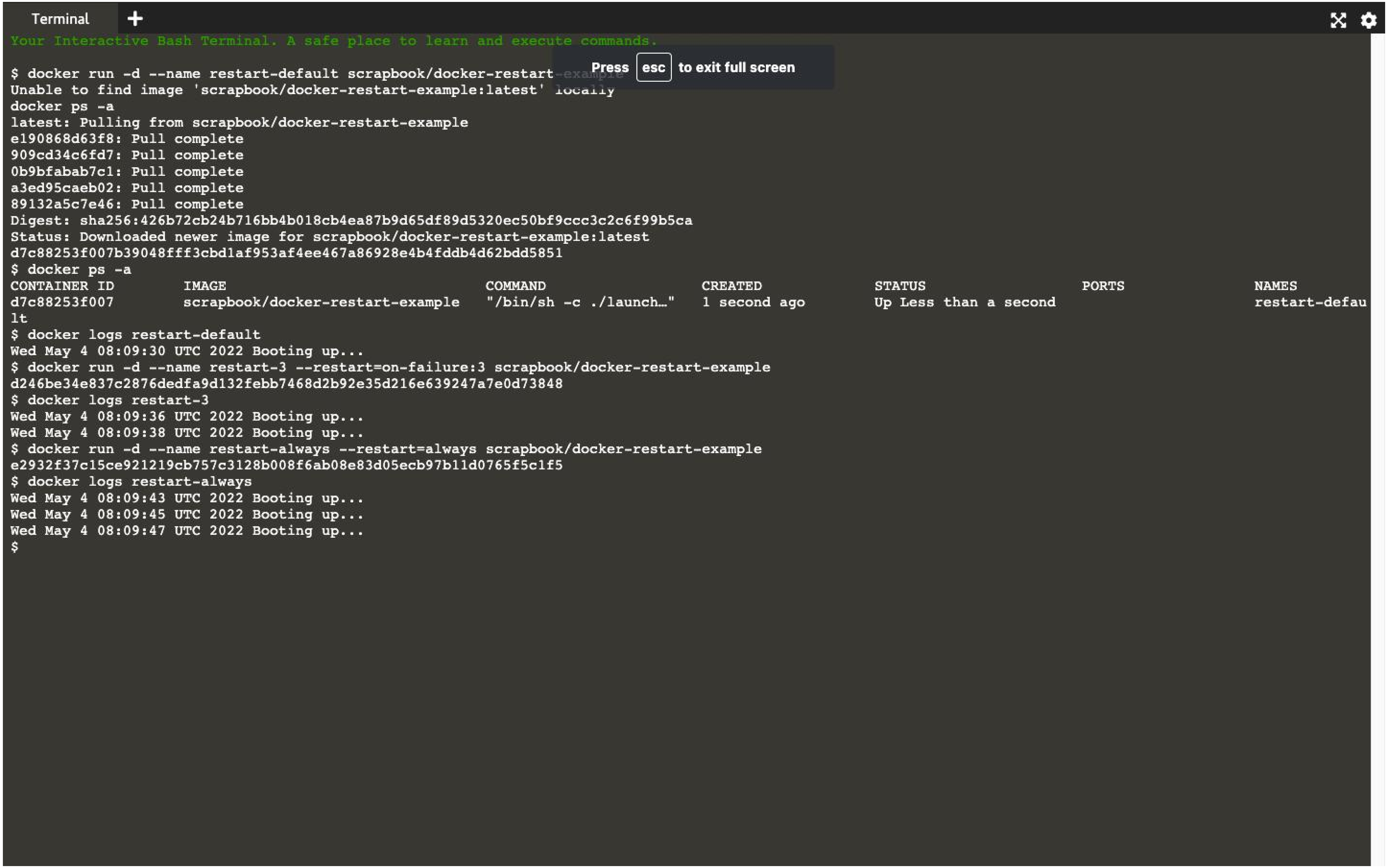
```
9a29abd5fcd4        frontend-network2   bridge              local
fa054a9af353        host                host                local
f50397115ef2        none                null                local
$ docker network inspect frontend-network
[
    {
        "Name": "frontend-network",
        "Id": "4c513cc96c122686cdc4d723ff00c288b246f6ed2b251f115d25c5e4b7132307",
        "Created": "2022-05-04T08:01:07.693615664Z",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": {},
            "Config": [
                {
                    "Subnet": "172.19.0.0/16",
                    "Gateway": "172.19.0.1"
                }
            ]
        },
        "Internal": false,
        "Attachable": false,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
        "Containers": {
            "59e40ba662e2c0abd772174b5200955c4952933604858924436de26df0e39930": {
                "Name": "redis",
                "EndpointID": "2099aabcfe6422460ab6d6d691a50528725face642ad2fba9817759f53c153d1",
                "MacAddress": "02:42:ac:13:00:02",
                "IPv4Address": "172.19.0.2/16",
                "IPv6Address": ""
            },
            "d32e264a0f4cc1d4f0b1448c039472bf0e063971be82da83044a524a14e87f1a": {
                "Name": "elastic_einstein",
                "EndpointID": "b0fa74e8ae9af64a80b2f1a2b2c8fb0790050ec7c61d5ae93649afe6b7af64a9",
                "MacAddress": "02:42:ac:13:00:03",
                "IPv4Address": "172.19.0.3/16",
                "IPv6Address": ""
            }
        },
        "Options": {},
        "Labels": {}
    }
]
$ docker network disconnect frontend-network redis
$
```



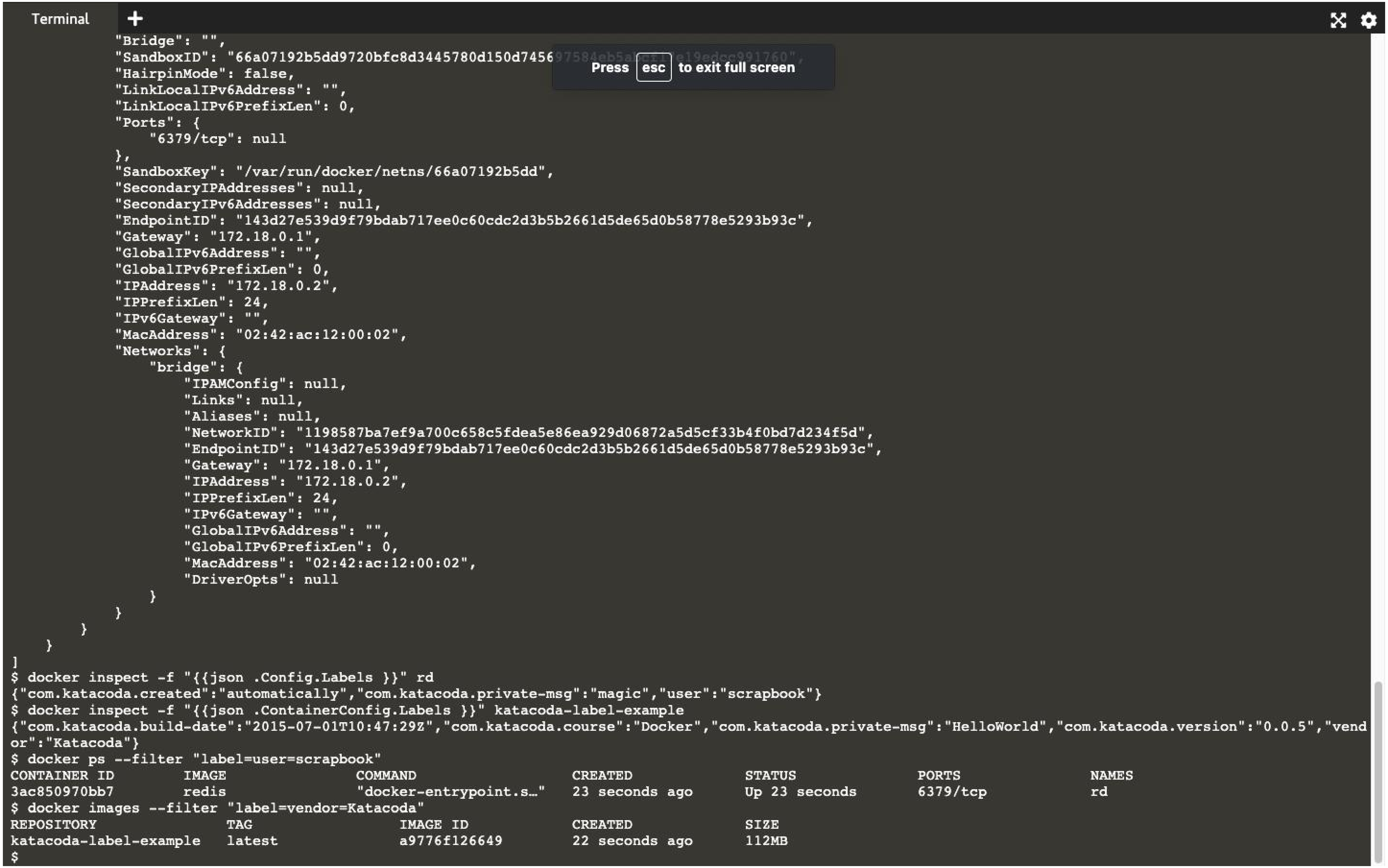
```
Your Interactive Bash Terminal. A safe place to learn and execute commands.


[root@host01 ~]#
[root@host01 ~]# docker run  -v /docker/redis-data:/data \
>   --name r1 -d redis \
>   redis-server --appendonly yes
0ea33347af715b0d972800f5cbfd668bd8a732cfb6f9c15f5cd882bc9226d524
[root@host01 ~]# cat data | docker exec -i r1 redis-cli --pipe
cat: data: No such file or directory
All data transferred. Waiting for the last reply...
Last reply received from server.
errors: 0, replies: 0
[root@host01 ~]# ls /docker/redis-data
appendonly.aof
[root@host01 ~]# docker run  -v /docker/redis-data:/backup ubuntu ls /backup
appendonly.aof
[root@host01 ~]# docker run --volumes-from r1 -it ubuntu ls /data
appendonly.aof
[root@host01 ~]# docker run -v /docker/redis-data:/data:ro -it ubuntu rm -rf /data
rm: cannot remove '/data/appendonly.aof': Read-only file system
[root@host01 ~]#
```



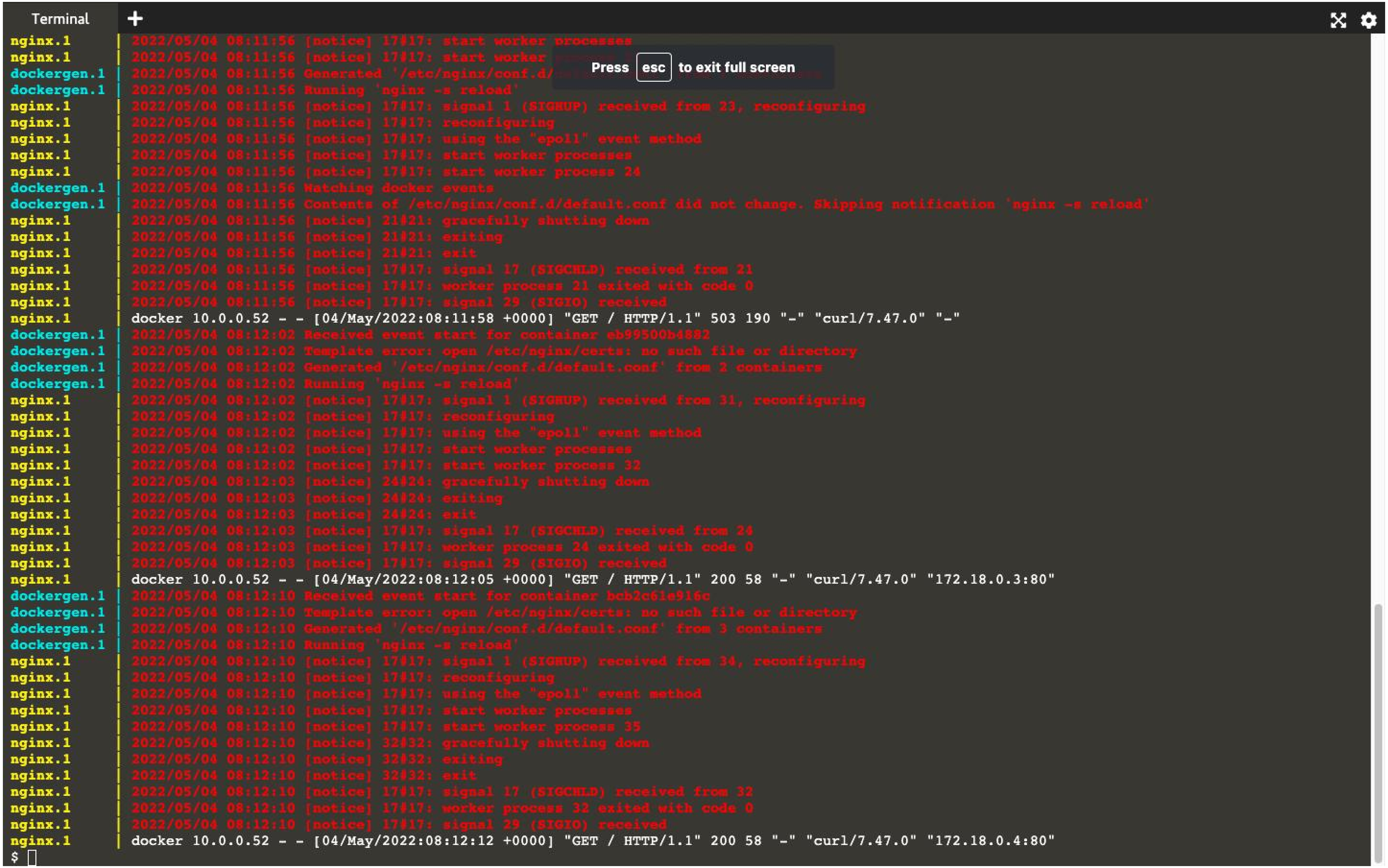
```
Your Interactive Bash Terminal. A safe place to learn and execute commands.

$ docker run -d --name redis-server redis
f0abeaac2b9c29cb48d1b0e47650e257bf449e84792b13c27eea741e8519ddfc
$
$
$ docker logs redis-server
1:C 04 May 08:03:52.535 # oO0OoO0OoO0Oo Redis is starting oO0OoO0OoO0Oo
1:C 04 May 08:03:52.535 # Redis version=4.0.8, bits=64, commit=00000000, modified=0, pid=1, just started
1:C 04 May 08:03:52.535 # Warning: no config file specified, using the default config. In order to specify a config file use redis-server /path/to/redis.conf
1:M 04 May 08:03:52.537 * Running mode=standalone, port=6379.
1:M 04 May 08:03:52.537 # WARNING: The TCP backlog setting of 511 cannot be enforced because /proc/sys/net/core/somaxconn is set to the lower value of 128.
1:M 04 May 08:03:52.537 # Server initialized
1:M 04 May 08:03:52.537 # WARNING overcommit_memory is set to 0! Background save may fail under low memory condition. To fix this issue add 'vm.overcommit_me
mory = 1' to /etc/sysctl.conf and then reboot or run the command 'sysctl vm.overcommit_memory=1' for this to take effect.
1:M 04 May 08:03:52.537 # WARNING you have Transparent Huge Pages (THP) support enabled in your kernel. This will create latency and memory usage issues with
 Redis. To fix this issue run the command 'echo never > /sys/kernel/mm/transparent_hugepage/enabled' as root, and add it to your /etc/rc.local in order to re
tain the setting after a reboot. Redis must be restarted after THP is disabled.
1:M 04 May 08:03:52.537 * Ready to accept connections
$ docker run -d --name redis-syslog --log-driver=syslog redis
85277d16cc9d1b026343bfc14da8c4d40bff75b24a205e8dde18f8395f366966
$ docker run -d --name redis-none --log-driver=none redis
a7e72e36e962afe0296dfc393a2dd00fc53318a9bfcdb5550ea4015f80b10738
$ docker inspect --format '{{ .HostConfig.LogConfig }}' redis-server
{json-file map[]}
$ docker inspect --format '{{ .HostConfig.LogConfig }}' redis-syslog
{syslog map[]}
$ docker inspect --format '{{ .HostConfig.LogConfig }}' redis-none
{none map[]}
$
```



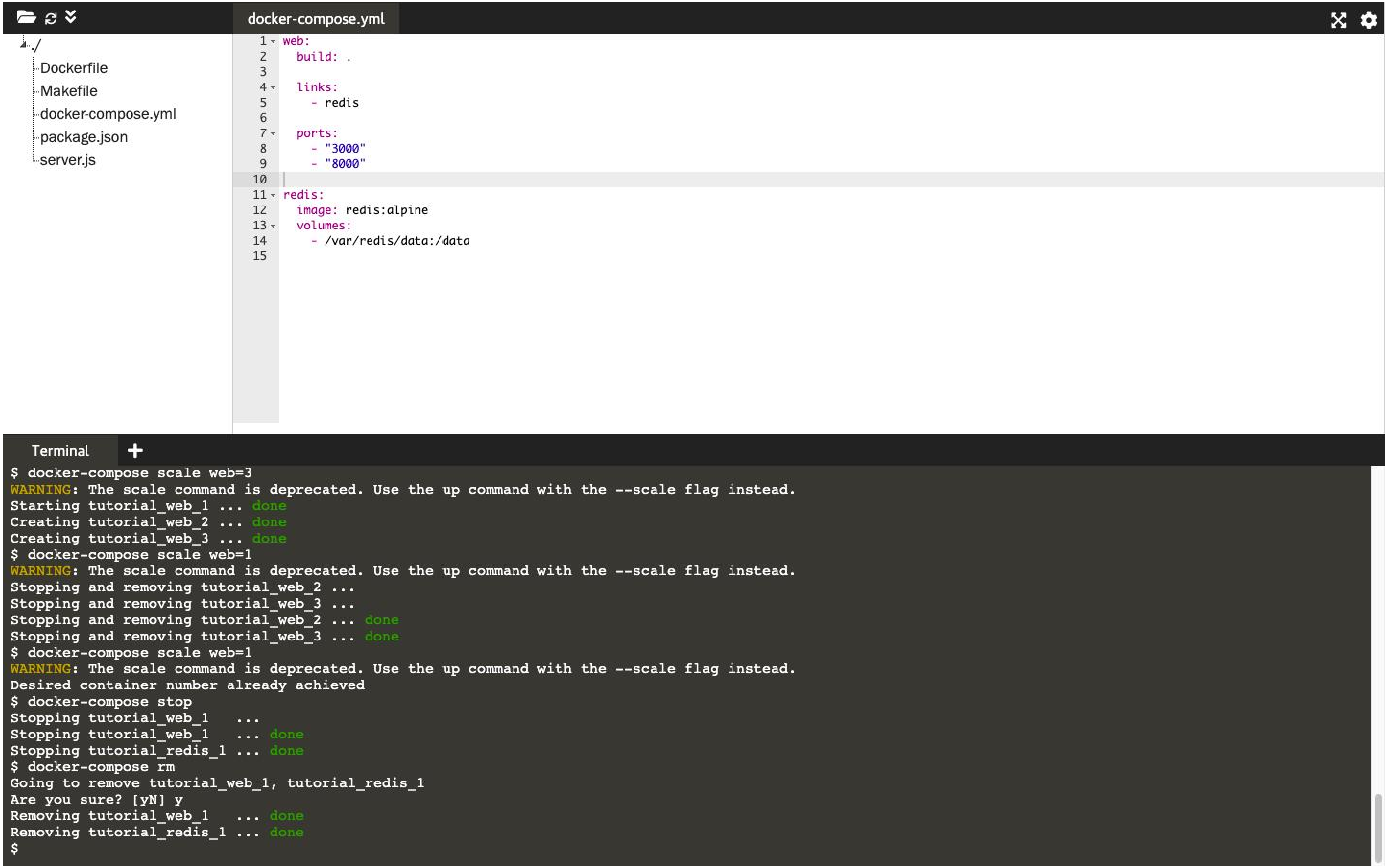
```
Your Interactive Bash Terminal. A safe place to learn and execute commands.

$ docker run -d --name restart-default scrapbook/docker-restart-example
Unable to find image 'scrapbook/docker-restart-example:latest' locally
docker ps -a
latest: Pulling from scrapbook/docker-restart-example
e190868d63f8: Pull complete
909cd34c6fd7: Pull complete
0b9bfabab7c1: Pull complete
a3ed95caeb02: Pull complete
89132a5c7e46: Pull complete
Digest: sha256:426b72cb24b716bb4b018cb4ea87b9d65df89d5320ec50bf9ccc3c2c6f99b5ca
Status: Downloaded newer image for scrapbook/docker-restart-example:latest
d7c88253f007b39048fff3cbd1af953af4ee467a86928e4b4fddb4d62bdd5851
$ docker ps -a
CONTAINER ID        IMAGE                              COMMAND                  CREATED             STATUS                   PORTS               NAMES
d7c88253f007        scrapbook/docker-restart-example   "/bin/sh -c ./launch…"   1 second ago        Up Less than a second                        restart-defau
lt
$ docker logs restart-default
Wed May 4 08:09:30 UTC 2022 Booting up...
$ docker run -d --name restart-3 --restart=on-failure:3 scrapbook/docker-restart-example
d246be34e837c2876dedfa9d132febb7468d2b92e35d216e639247a7e0d73848
$ docker logs restart-3
Wed May 4 08:09:36 UTC 2022 Booting up...
Wed May 4 08:09:38 UTC 2022 Booting up...
$ docker run -d --name restart-always --restart=always scrapbook/docker-restart-example
e2932f37c15ce921219cb757c3128b008f6ab08e83d05ecb97b11d0765f5c1f5
$ docker logs restart-always
Wed May 4 08:09:43 UTC 2022 Booting up...
Wed May 4 08:09:45 UTC 2022 Booting up...
Wed May 4 08:09:47 UTC 2022 Booting up...
$
```



```
            "Bridge": "",
            "SandboxID": "66a07192b5dd9720bfc8d3445780d150d745697584eb5a[illegible]e19edcc991760",
            "HairpinMode": false,
            "LinkLocalIPv6Address": "",
            "LinkLocalIPv6PrefixLen": 0,
            "Ports": {
                "6379/tcp": null
            },
            "SandboxKey": "/var/run/docker/netns/66a07192b5dd",
            "SecondaryIPAddresses": null,
            "SecondaryIPv6Addresses": null,
            "EndpointID": "143d27e539d9f79bdab717ee0c60cdc2d3b5b2661d5de65d0b58778e5293b93c",
            "Gateway": "172.18.0.1",
            "GlobalIPv6Address": "",
            "GlobalIPv6PrefixLen": 0,
            "IPAddress": "172.18.0.2",
            "IPPrefixLen": 24,
            "IPv6Gateway": "",
            "MacAddress": "02:42:ac:12:00:02",
            "Networks": {
                "bridge": {
                    "IPAMConfig": null,
                    "Links": null,
                    "Aliases": null,
                    "NetworkID": "1198587ba7ef9a700c658c5fdea5e86ea929d06872a5d5cf33b4f0bd7d234f5d",
                    "EndpointID": "143d27e539d9f79bdab717ee0c60cdc2d3b5b2661d5de65d0b58778e5293b93c",
                    "Gateway": "172.18.0.1",
                    "IPAddress": "172.18.0.2",
                    "IPPrefixLen": 24,
                    "IPv6Gateway": "",
                    "GlobalIPv6Address": "",
                    "GlobalIPv6PrefixLen": 0,
                    "MacAddress": "02:42:ac:12:00:02",
                    "DriverOpts": null
                }
            }
        }
    }
]
$ docker inspect -f "{{json .Config.Labels }}" rd
{"com.katacoda.created":"automatically","com.katacoda.private-msg":"magic","user":"scrapbook"}
$ docker inspect -f "{{json .ContainerConfig.Labels }}" katacoda-label-example
{"com.katacoda.build-date":"2015-07-01T10:47:29Z","com.katacoda.course":"Docker","com.katacoda.private-msg":"HelloWorld","com.katacoda.version":"0.0.5","vend
or":"Katacoda"}
$ docker ps --filter "label=user=scrapbook"
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS               NAMES
3ac850970bb7        redis               "docker-entrypoint.s…"   23 seconds ago      Up 23 seconds       6379/tcp            rd
$ docker images --filter "label=vendor=Katacoda"
REPOSITORY               TAG                 IMAGE ID            CREATED             SIZE
katacoda-label-example   latest              a9776f126649        22 seconds ago      112MB
$
```

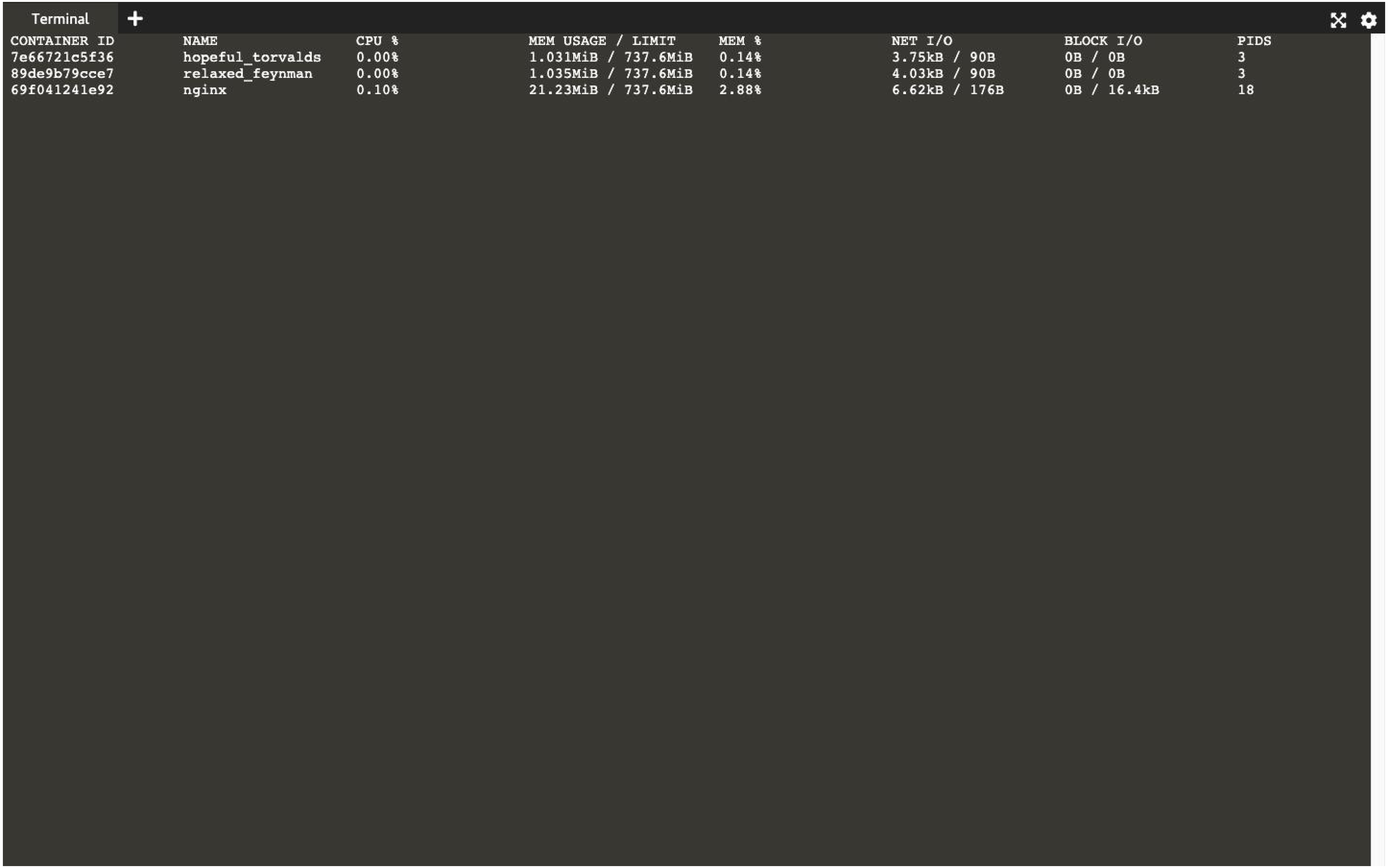


```
nginx.1      | 2022/05/04 08:11:56 [notice] 17#17: start worker processes
nginx.1      | 2022/05/04 08:11:56 [notice] 17#17: start worker
dockergen.1  | 2022/05/04 08:11:56 Generated '/etc/nginx/conf.d/
dockergen.1  | 2022/05/04 08:11:56 Running 'nginx -s reload'
nginx.1      | 2022/05/04 08:11:56 [notice] 17#17: signal 1 (SIGHUP) received from 23, reconfiguring
nginx.1      | 2022/05/04 08:11:56 [notice] 17#17: reconfiguring
nginx.1      | 2022/05/04 08:11:56 [notice] 17#17: using the "epoll" event method
nginx.1      | 2022/05/04 08:11:56 [notice] 17#17: start worker processes
nginx.1      | 2022/05/04 08:11:56 [notice] 17#17: start worker process 24
dockergen.1  | 2022/05/04 08:11:56 Watching docker events
dockergen.1  | 2022/05/04 08:11:56 Contents of /etc/nginx/conf.d/default.conf did not change. Skipping notification 'nginx -s reload'
nginx.1      | 2022/05/04 08:11:56 [notice] 21#21: gracefully shutting down
nginx.1      | 2022/05/04 08:11:56 [notice] 21#21: exiting
nginx.1      | 2022/05/04 08:11:56 [notice] 21#21: exit
nginx.1      | 2022/05/04 08:11:56 [notice] 17#17: signal 17 (SIGCHLD) received from 21
nginx.1      | 2022/05/04 08:11:56 [notice] 17#17: worker process 21 exited with code 0
nginx.1      | 2022/05/04 08:11:56 [notice] 17#17: signal 29 (SIGIO) received
nginx.1      | docker 10.0.0.52 - - [04/May/2022:08:11:58 +0000] "GET / HTTP/1.1" 503 190 "-" "curl/7.47.0" "-"
dockergen.1  | 2022/05/04 08:12:02 Received event start for container eb99500b4882
dockergen.1  | 2022/05/04 08:12:02 Template error: open /etc/nginx/certs: no such file or directory
dockergen.1  | 2022/05/04 08:12:02 Generated '/etc/nginx/conf.d/default.conf' from 2 containers
dockergen.1  | 2022/05/04 08:12:02 Running 'nginx -s reload'
nginx.1      | 2022/05/04 08:12:02 [notice] 17#17: signal 1 (SIGHUP) received from 31, reconfiguring
nginx.1      | 2022/05/04 08:12:02 [notice] 17#17: reconfiguring
nginx.1      | 2022/05/04 08:12:02 [notice] 17#17: using the "epoll" event method
nginx.1      | 2022/05/04 08:12:02 [notice] 17#17: start worker processes
nginx.1      | 2022/05/04 08:12:02 [notice] 17#17: start worker process 32
nginx.1      | 2022/05/04 08:12:03 [notice] 24#24: gracefully shutting down
nginx.1      | 2022/05/04 08:12:03 [notice] 24#24: exiting
nginx.1      | 2022/05/04 08:12:03 [notice] 24#24: exit
nginx.1      | 2022/05/04 08:12:03 [notice] 17#17: signal 17 (SIGCHLD) received from 24
nginx.1      | 2022/05/04 08:12:03 [notice] 17#17: worker process 24 exited with code 0
nginx.1      | 2022/05/04 08:12:03 [notice] 17#17: signal 29 (SIGIO) received
nginx.1      | docker 10.0.0.52 - - [04/May/2022:08:12:05 +0000] "GET / HTTP/1.1" 200 58 "-" "curl/7.47.0" "172.18.0.3:80"
dockergen.1  | 2022/05/04 08:12:10 Received event start for container bcb2c61e916c
dockergen.1  | 2022/05/04 08:12:10 Template error: open /etc/nginx/certs: no such file or directory
dockergen.1  | 2022/05/04 08:12:10 Generated '/etc/nginx/conf.d/default.conf' from 3 containers
dockergen.1  | 2022/05/04 08:12:10 Running 'nginx -s reload'
nginx.1      | 2022/05/04 08:12:10 [notice] 17#17: signal 1 (SIGHUP) received from 34, reconfiguring
nginx.1      | 2022/05/04 08:12:10 [notice] 17#17: reconfiguring
nginx.1      | 2022/05/04 08:12:10 [notice] 17#17: using the "epoll" event method
nginx.1      | 2022/05/04 08:12:10 [notice] 17#17: start worker processes
nginx.1      | 2022/05/04 08:12:10 [notice] 17#17: start worker process 35
nginx.1      | 2022/05/04 08:12:10 [notice] 32#32: gracefully shutting down
nginx.1      | 2022/05/04 08:12:10 [notice] 32#32: exiting
nginx.1      | 2022/05/04 08:12:10 [notice] 32#32: exit
nginx.1      | 2022/05/04 08:12:10 [notice] 17#17: signal 17 (SIGCHLD) received from 32
nginx.1      | 2022/05/04 08:12:10 [notice] 17#17: worker process 32 exited with code 0
nginx.1      | 2022/05/04 08:12:10 [notice] 17#17: signal 29 (SIGIO) received
nginx.1      | docker 10.0.0.52 - - [04/May/2022:08:12:12 +0000] "GET / HTTP/1.1" 200 58 "-" "curl/7.47.0" "172.18.0.4:80"
$ 
```

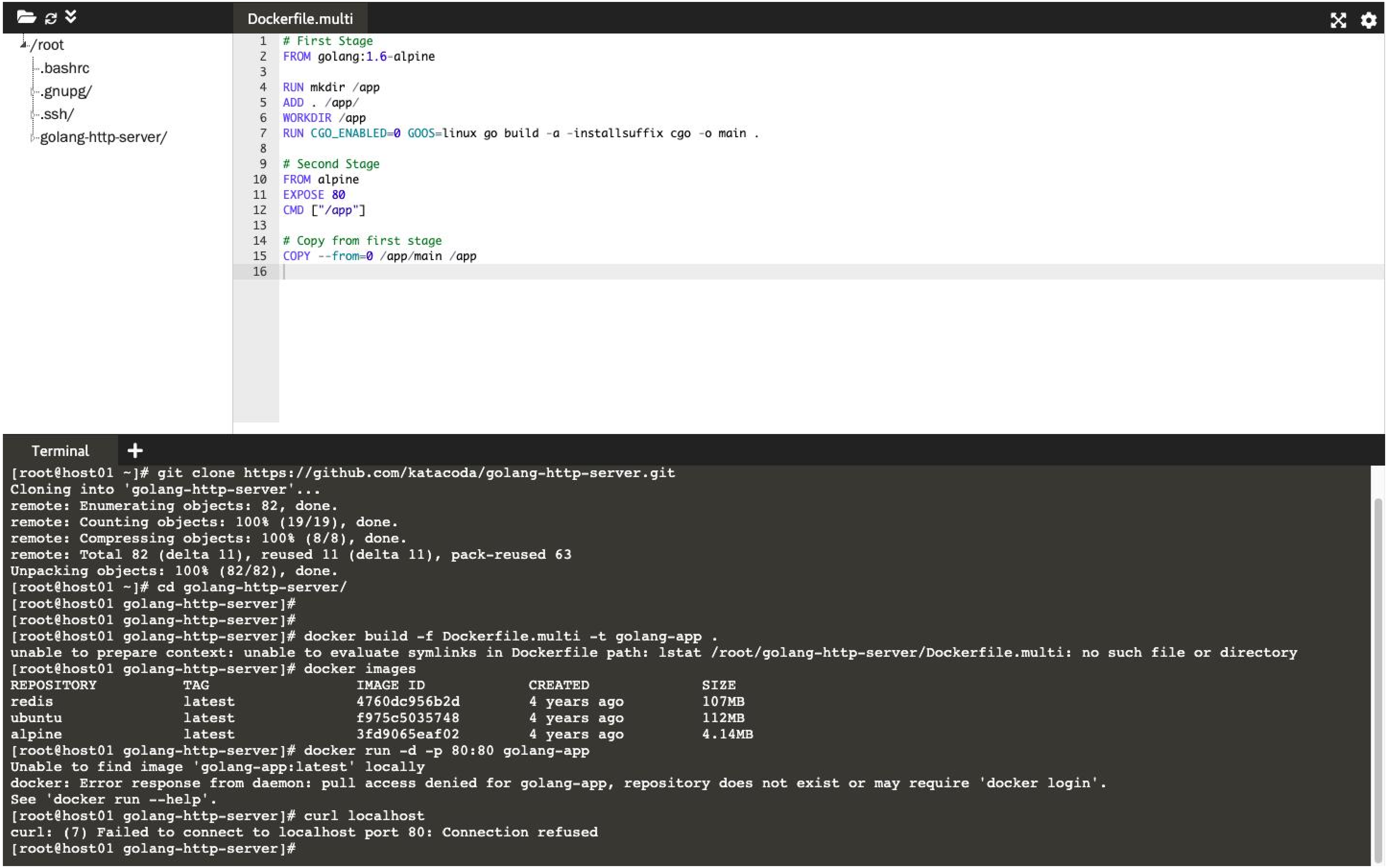


```yaml
web:
  build: .

  links:
    - redis

  ports:
    - "3000"
    - "8000"

redis:
  image: redis:alpine
  volumes:
    - /var/redis/data:/data

```

```
$ docker-compose scale web=3
WARNING: The scale command is deprecated. Use the up command with the --scale flag instead.
Starting tutorial_web_1 ... done
Creating tutorial_web_2 ... done
Creating tutorial_web_3 ... done
$ docker-compose scale web=1
WARNING: The scale command is deprecated. Use the up command with the --scale flag instead.
Stopping and removing tutorial_web_2 ...
Stopping and removing tutorial_web_3 ...
Stopping and removing tutorial_web_2 ... done
Stopping and removing tutorial_web_3 ... done
$ docker-compose scale web=1
WARNING: The scale command is deprecated. Use the up command with the --scale flag instead.
Desired container number already achieved
$ docker-compose stop
Stopping tutorial_web_1   ...
Stopping tutorial_web_1   ... done
Stopping tutorial_redis_1 ... done
$ docker-compose rm
Going to remove tutorial_web_1, tutorial_redis_1
Are you sure? [yN] y
Removing tutorial_web_1   ... done
Removing tutorial_redis_1 ... done
$
```

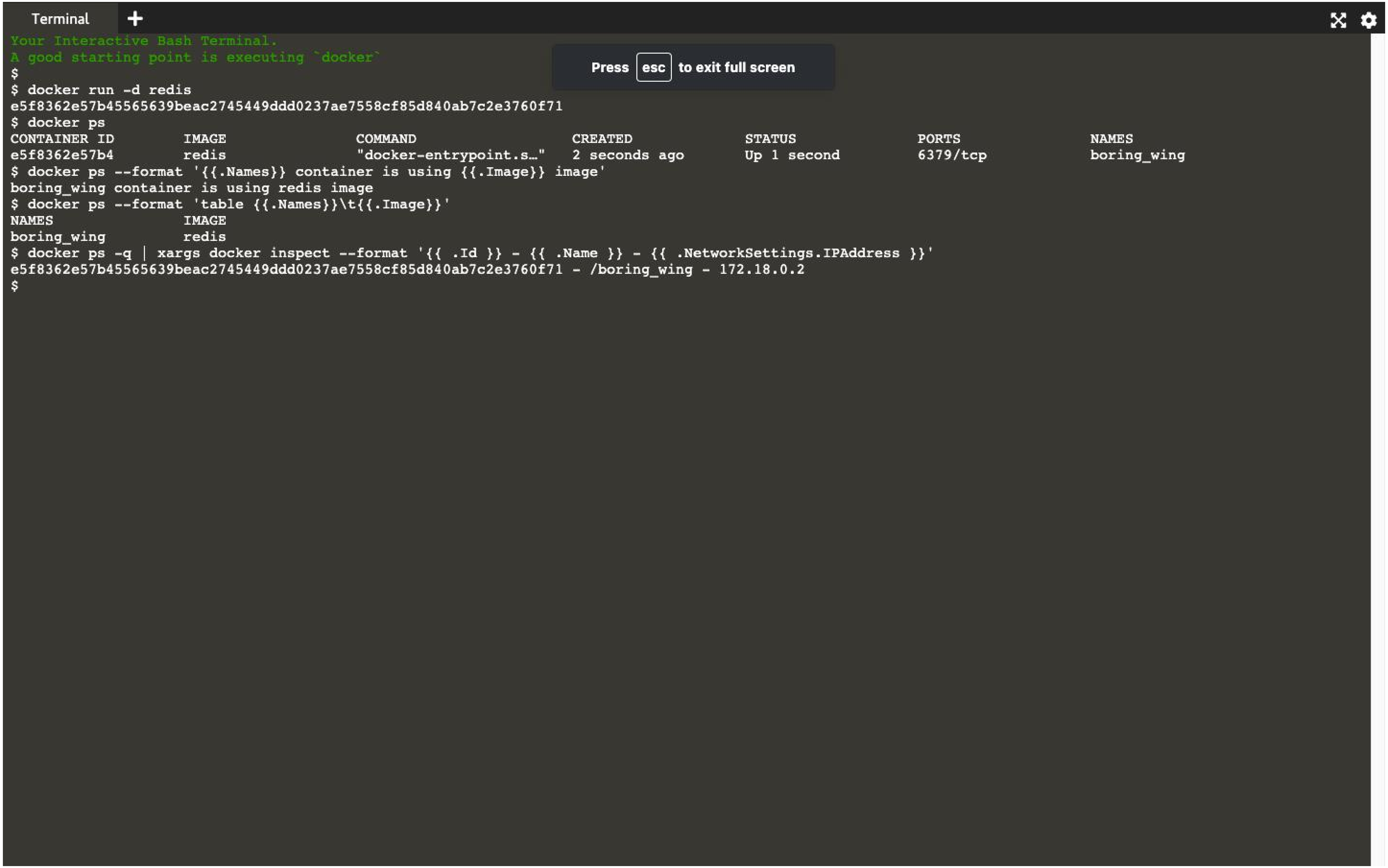


```
CONTAINER ID        NAME                CPU %               MEM USAGE / LIMIT     MEM %               NET I/O             BLOCK I/O           PIDS
7e66721c5f36        hopeful_torvalds    0.00%               1.031MiB / 737.6MiB   0.14%               3.75kB / 90B        0B / 0B             3
89de9b79cce7        relaxed_feynman     0.00%               1.035MiB / 737.6MiB   0.14%               4.03kB / 90B        0B / 0B             3
69f041241e92        nginx               0.10%               21.23MiB / 737.6MiB   2.88%               6.62kB / 176B       0B / 16.4kB         18
```

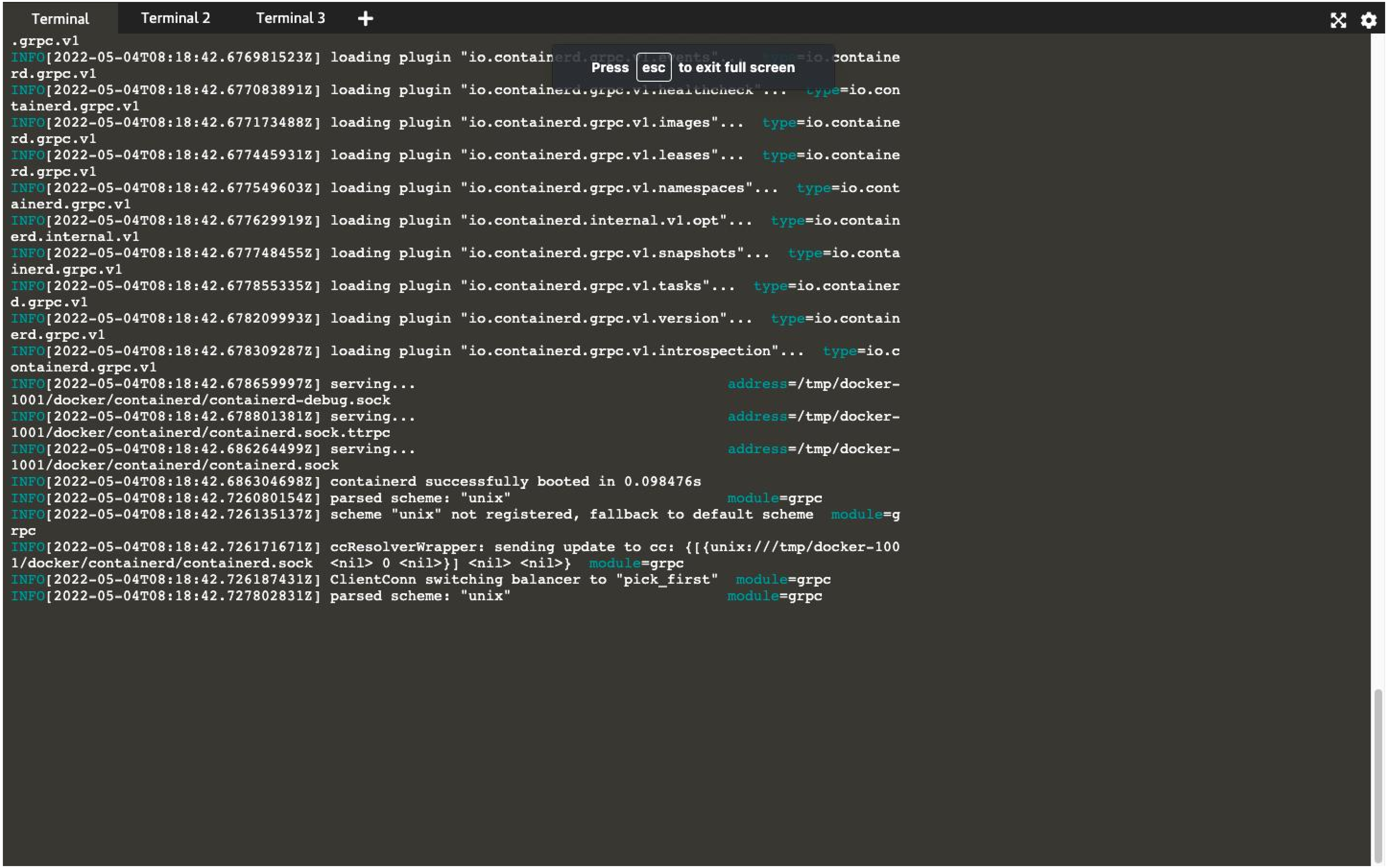


```
# First Stage
FROM golang:1.6-alpine

RUN mkdir /app
ADD . /app/
WORKDIR /app
RUN CGO_ENABLED=0 GOOS=linux go build -a -installsuffix cgo -o main .

# Second Stage
FROM alpine
EXPOSE 80
CMD ["/app"]

# Copy from first stage
COPY --from=0 /app/main /app

```

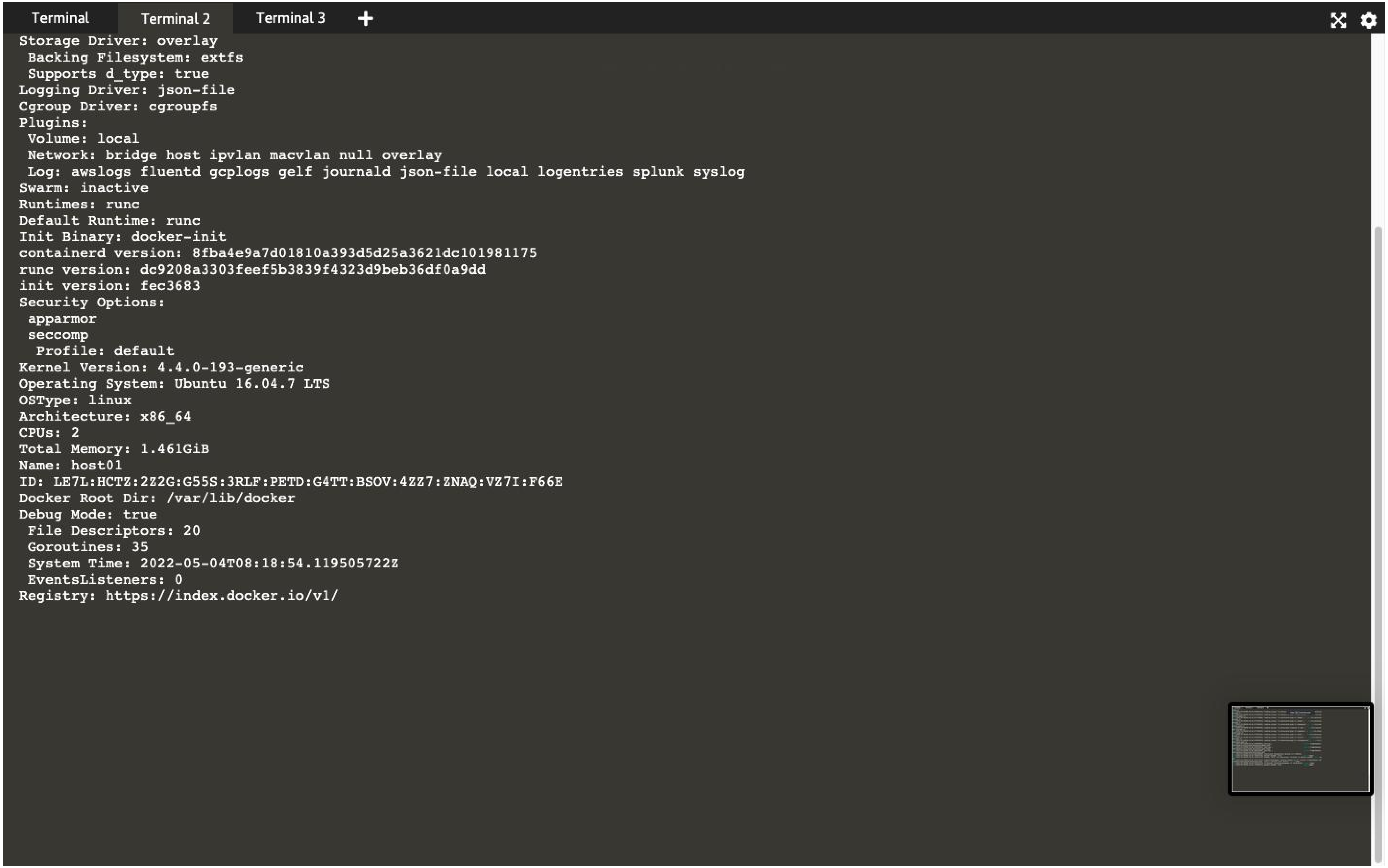
```
[root@host01 ~]# git clone https://github.com/katacoda/golang-http-server.git
Cloning into 'golang-http-server'...
remote: Enumerating objects: 82, done.
remote: Counting objects: 100% (19/19), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 82 (delta 11), reused 11 (delta 11), pack-reused 63
Unpacking objects: 100% (82/82), done.
[root@host01 ~]# cd golang-http-server/
[root@host01 golang-http-server]#
[root@host01 golang-http-server]#
[root@host01 golang-http-server]# docker build -f Dockerfile.multi -t golang-app .
unable to prepare context: unable to evaluate symlinks in Dockerfile path: lstat /root/golang-http-server/Dockerfile.multi: no such file or directory
[root@host01 golang-http-server]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
redis               latest              4760dc956b2d        4 years ago         107MB
ubuntu              latest              f975c5035748        4 years ago         112MB
alpine              latest              3fd9065eaf02        4 years ago         4.14MB
[root@host01 golang-http-server]# docker run -d -p 80:80 golang-app
Unable to find image 'golang-app:latest' locally
docker: Error response from daemon: pull access denied for golang-app, repository does not exist or may require 'docker login'.
See 'docker run --help'.
[root@host01 golang-http-server]# curl localhost
curl: (7) Failed to connect to localhost port 80: Connection refused
[root@host01 golang-http-server]#
```



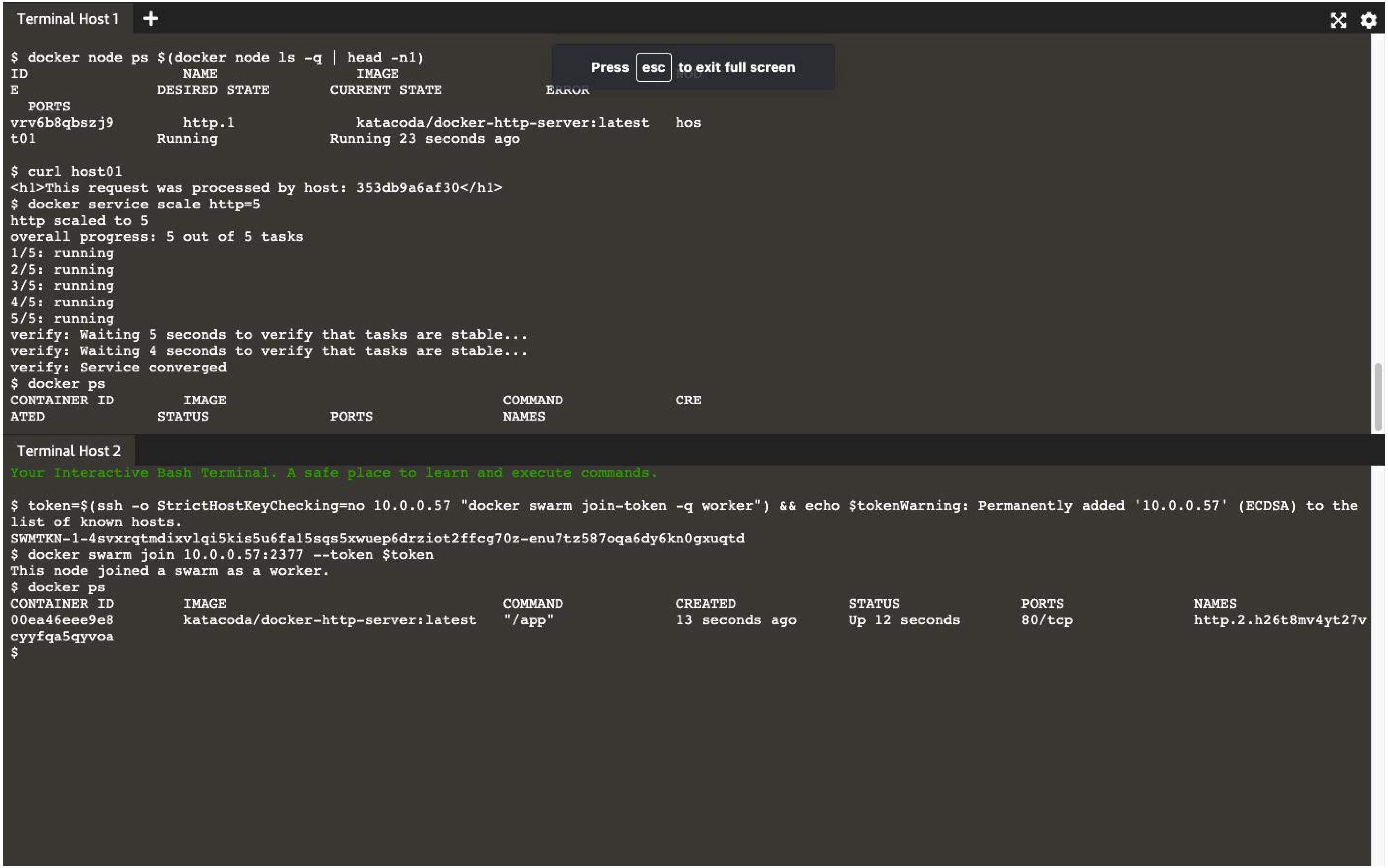
```
Your Interactive Bash Terminal.
A good starting point is executing `docker`
$
$ docker run -d redis
e5f8362e57b45565639beac2745449ddd0237ae7558cf85d840ab7c2e3760f71
$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS               NAMES
e5f8362e57b4        redis               "docker-entrypoint.s…"   2 seconds ago       Up 1 second         6379/tcp            boring_wing
$ docker ps --format '{{.Names}} container is using {{.Image}} image'
boring_wing container is using redis image
$ docker ps --format 'table {{.Names}}\t{{.Image}}'
NAMES               IMAGE
boring_wing         redis
$ docker ps -q | xargs docker inspect --format '{{ .Id }} - {{ .Name }} - {{ .NetworkSettings.IPAddress }}'
e5f8362e57b45565639beac2745449ddd0237ae7558cf85d840ab7c2e3760f71 - /boring_wing - 172.18.0.2
$
```



```
.grpc.v1
INFO[2022-05-04T08:18:42.676981523Z] loading plugin "io.containerd.grpc.v1.events"...  type=io.containe
rd.grpc.v1
INFO[2022-05-04T08:18:42.677083891Z] loading plugin "io.containerd.grpc.v1.healthcheck"...  type=io.con
tainerd.grpc.v1
INFO[2022-05-04T08:18:42.677173488Z] loading plugin "io.containerd.grpc.v1.images"...  type=io.containe
rd.grpc.v1
INFO[2022-05-04T08:18:42.677445931Z] loading plugin "io.containerd.grpc.v1.leases"...  type=io.containe
rd.grpc.v1
INFO[2022-05-04T08:18:42.677549603Z] loading plugin "io.containerd.grpc.v1.namespaces"...  type=io.cont
ainerd.grpc.v1
INFO[2022-05-04T08:18:42.677629919Z] loading plugin "io.containerd.internal.v1.opt"...  type=io.contain
erd.internal.v1
INFO[2022-05-04T08:18:42.677748455Z] loading plugin "io.containerd.grpc.v1.snapshots"...  type=io.conta
inerd.grpc.v1
INFO[2022-05-04T08:18:42.677855335Z] loading plugin "io.containerd.grpc.v1.tasks"...  type=io.container
d.grpc.v1
INFO[2022-05-04T08:18:42.678209993Z] loading plugin "io.containerd.grpc.v1.version"...  type=io.contain
erd.grpc.v1
INFO[2022-05-04T08:18:42.678309287Z] loading plugin "io.containerd.grpc.v1.introspection"...  type=io.c
ontainerd.grpc.v1
INFO[2022-05-04T08:18:42.678659997Z] serving...                                    address=/tmp/docker-
1001/docker/containerd/containerd-debug.sock
INFO[2022-05-04T08:18:42.678801381Z] serving...                                    address=/tmp/docker-
1001/docker/containerd/containerd.sock.ttrpc
INFO[2022-05-04T08:18:42.686264499Z] serving...                                    address=/tmp/docker-
1001/docker/containerd/containerd.sock
INFO[2022-05-04T08:18:42.686304698Z] containerd successfully booted in 0.098476s
INFO[2022-05-04T08:18:42.726080154Z] parsed scheme: "unix"                         module=grpc
INFO[2022-05-04T08:18:42.726135137Z] scheme "unix" not registered, fallback to default scheme  module=g
rpc
INFO[2022-05-04T08:18:42.726171671Z] ccResolverWrapper: sending update to cc: {[{unix:///tmp/docker-100
1/docker/containerd/containerd.sock  <nil> 0 <nil>}] <nil> <nil>}  module=grpc
INFO[2022-05-04T08:18:42.726187431Z] ClientConn switching balancer to "pick_first"  module=grpc
INFO[2022-05-04T08:18:42.727802831Z] parsed scheme: "unix"                         module=grpc
```

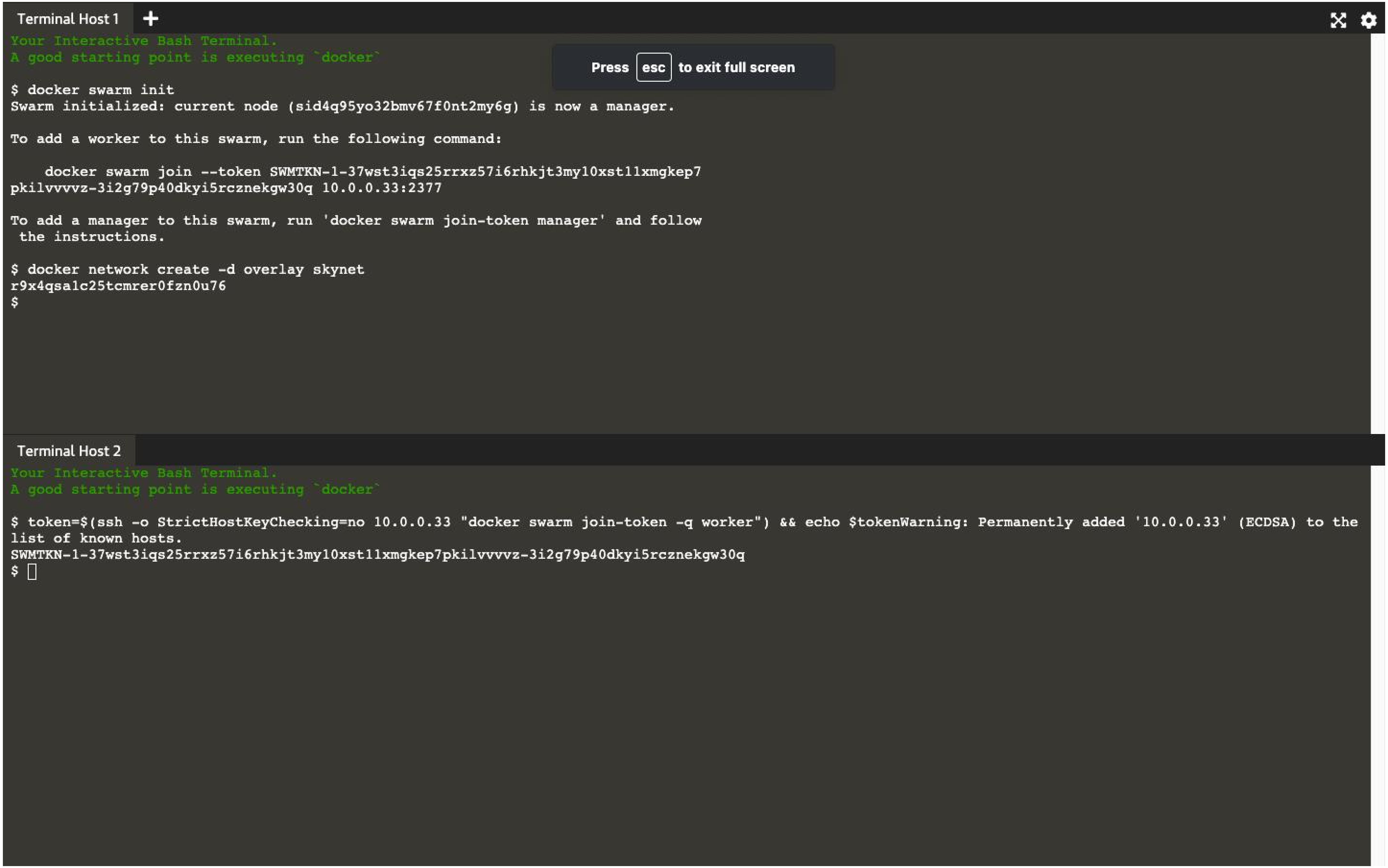


```
Storage Driver: overlay
 Backing Filesystem: extfs
 Supports d_type: true
Logging Driver: json-file
Cgroup Driver: cgroupfs
Plugins:
 Volume: local
 Network: bridge host ipvlan macvlan null overlay
 Log: awslogs fluentd gcplogs gelf journald json-file local logentries splunk syslog
Swarm: inactive
Runtimes: runc
Default Runtime: runc
Init Binary: docker-init
containerd version: 8fba4e9a7d01810a393d5d25a3621dc101981175
runc version: dc9208a3303feef5b3839f4323d9beb36df0a9dd
init version: fec3683
Security Options:
 apparmor
 seccomp
  Profile: default
Kernel Version: 4.4.0-193-generic
Operating System: Ubuntu 16.04.7 LTS
OSType: linux
Architecture: x86_64
CPUs: 2
Total Memory: 1.461GiB
Name: host01
ID: LE7L:HCTZ:2Z2G:G55S:3RLF:PETD:G4TT:BSOV:4ZZ7:ZNAQ:VZ7I:F66E
Docker Root Dir: /var/lib/docker
Debug Mode: true
 File Descriptors: 20
 Goroutines: 35
 System Time: 2022-05-04T08:18:54.119505722Z
 EventsListeners: 0
Registry: https://index.docker.io/v1/
```



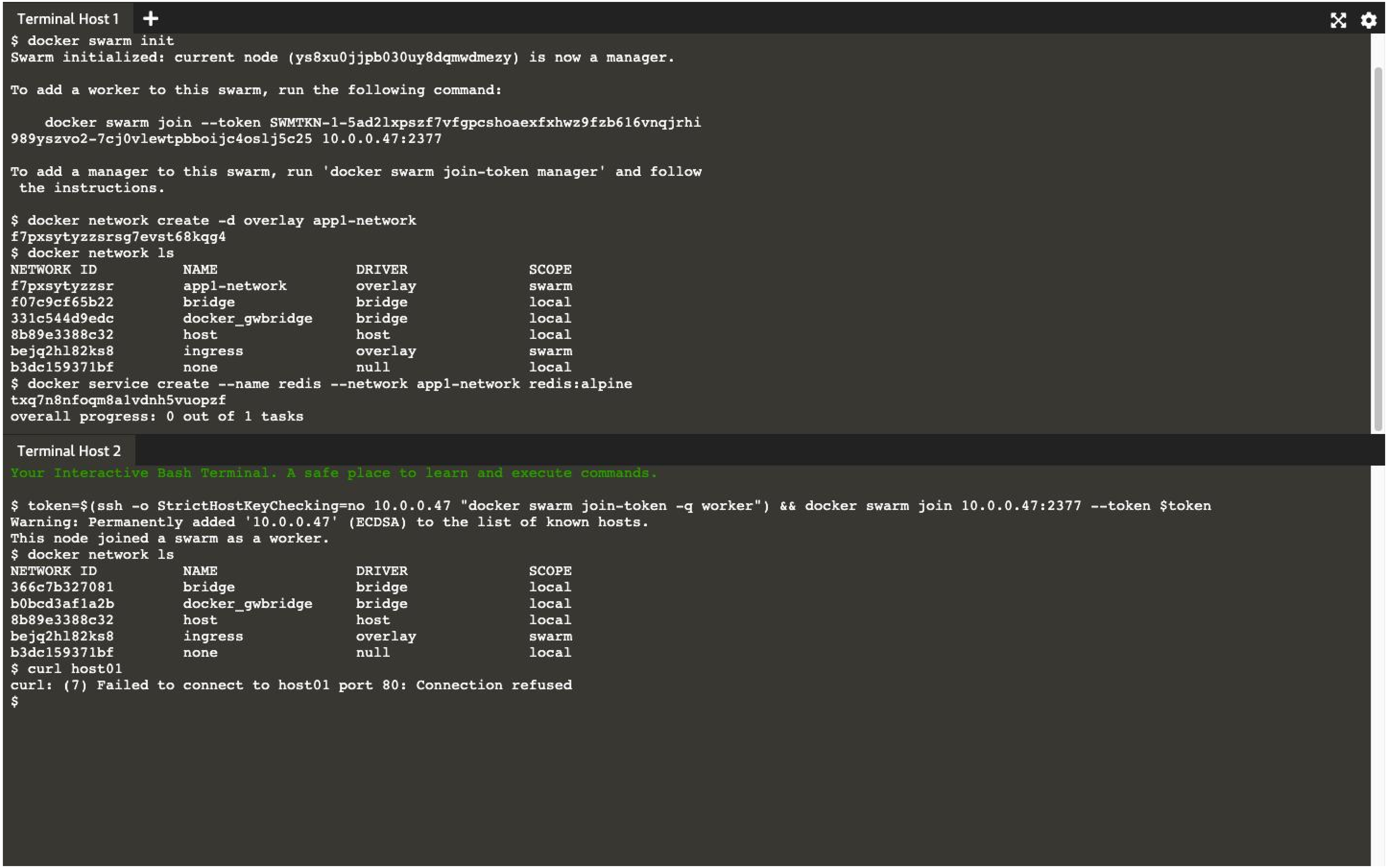
```
$ docker node ps $(docker node ls -q | head -n1)
ID                  NAME                IMAGE
E                DESIRED STATE       CURRENT STATE            ERROR
  PORTS
vrv6b8qbszj9        http.1              katacoda/docker-http-server:latest   hos
t01              Running             Running 23 seconds ago

$ curl host01
<h1>This request was processed by host: 353db9a6af30</h1>
$ docker service scale http=5
http scaled to 5
overall progress: 5 out of 5 tasks
1/5: running
2/5: running
3/5: running
4/5: running
5/5: running
verify: Waiting 5 seconds to verify that tasks are stable...
verify: Waiting 4 seconds to verify that tasks are stable...
verify: Service converged
$ docker ps
CONTAINER ID        IMAGE                                COMMAND             CRE
ATED             STATUS              PORTS               NAMES
```

```
Your Interactive Bash Terminal. A safe place to learn and execute commands.

$ token=$(ssh -o StrictHostKeyChecking=no 10.0.0.57 "docker swarm join-token -q worker") && echo $tokenWarning: Permanently added '10.0.0.57' (ECDSA) to the
list of known hosts.
SWMTKN-1-4svxrqtmdixvlqi5kis5u6fa15sqs5xwuep6drziot2ffcg70z-enu7tz587oqa6dy6kn0gxuqtd
$ docker swarm join 10.0.0.57:2377 --token $token
This node joined a swarm as a worker.
$ docker ps
CONTAINER ID        IMAGE                                COMMAND             CREATED             STATUS              PORTS               NAMES
00ea46eee9e8        katacoda/docker-http-server:latest   "/app"              13 seconds ago      Up 12 seconds       80/tcp              http.2.h26t8mv4yt27v
cyyfqa5qyvoa
$
```



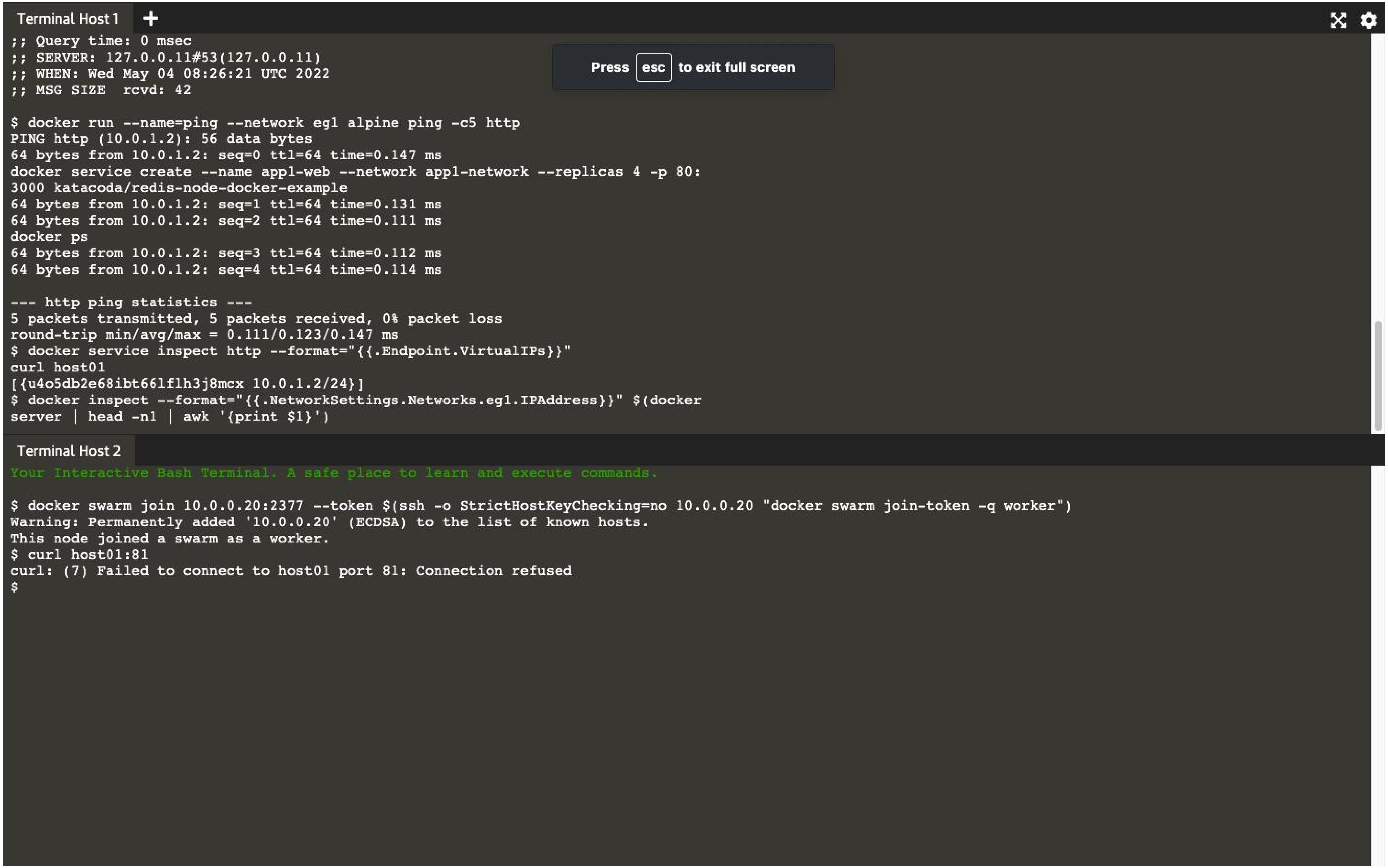
```
Your Interactive Bash Terminal.
A good starting point is executing `docker`

$ docker swarm init
Swarm initialized: current node (sid4q95yo32bmv67f0nt2my6g) is now a manager.

To add a worker to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-37wst3iqs25rrxz57i6rhkjt3my10xst11xmgkep7
pkilvvvvz-3i2g79p40dkyi5rcznekgw30q 10.0.0.33:2377

To add a manager to this swarm, run 'docker swarm join-token manager' and follow
 the instructions.

$ docker network create -d overlay skynet
r9x4qsa1c25tcmrer0fzn0u76
$
```

```
Your Interactive Bash Terminal.
A good starting point is executing `docker`

$ token=$(ssh -o StrictHostKeyChecking=no 10.0.0.33 "docker swarm join-token -q worker") && echo $tokenWarning: Permanently added '10.0.0.33' (ECDSA) to the
list of known hosts.
SWMTKN-1-37wst3iqs25rrxz57i6rhkjt3my10xst11xmgkep7pkilvvvvz-3i2g79p40dkyi5rcznekgw30q
$ 
```



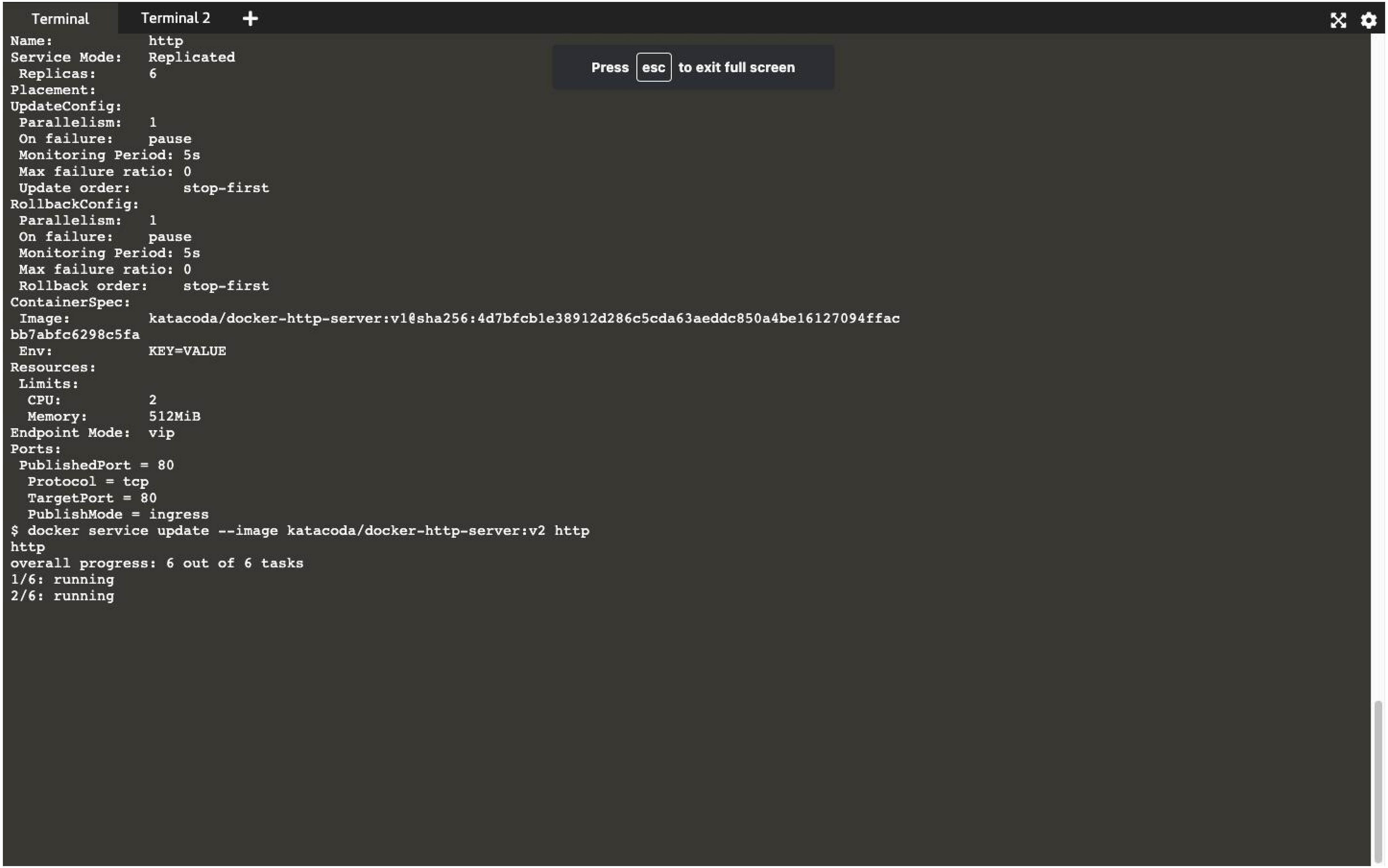
```
$ docker swarm init
Swarm initialized: current node (ys8xu0jjpb030uy8dqmwdmezy) is now a manager.

To add a worker to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-5ad2lxpszf7vfgpcshoaexfxhwz9fzb616vnqjrhi
989yszvo2-7cj0vlewtpbboijc4oslj5c25 10.0.0.47:2377

To add a manager to this swarm, run 'docker swarm join-token manager' and follow
 the instructions.

$ docker network create -d overlay app1-network
f7pxsytyzzsrsg7evst68kqg4
$ docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
f7pxsytyzzsr        app1-network        overlay             swarm
f07c9cf65b22        bridge              bridge              local
331c544d9edc        docker_gwbridge     bridge              local
8b89e3388c32        host                host                local
bejq2hl82ks8        ingress             overlay             swarm
b3dc159371bf        none                null                local
$ docker service create --name redis --network app1-network redis:alpine
txq7n8nfoqm8a1vdnh5vuopzf
overall progress: 0 out of 1 tasks
```

```
Your Interactive Bash Terminal. A safe place to learn and execute commands.

$ token=$(ssh -o StrictHostKeyChecking=no 10.0.0.47 "docker swarm join-token -q worker") && docker swarm join 10.0.0.47:2377 --token $token
Warning: Permanently added '10.0.0.47' (ECDSA) to the list of known hosts.
This node joined a swarm as a worker.
$ docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
366c7b327081        bridge              bridge              local
b0bcd3af1a2b        docker_gwbridge     bridge              local
8b89e3388c32        host                host                local
bejq2hl82ks8        ingress             overlay             swarm
b3dc159371bf        none                null                local
$ curl host01
curl: (7) Failed to connect to host01 port 80: Connection refused
$
```

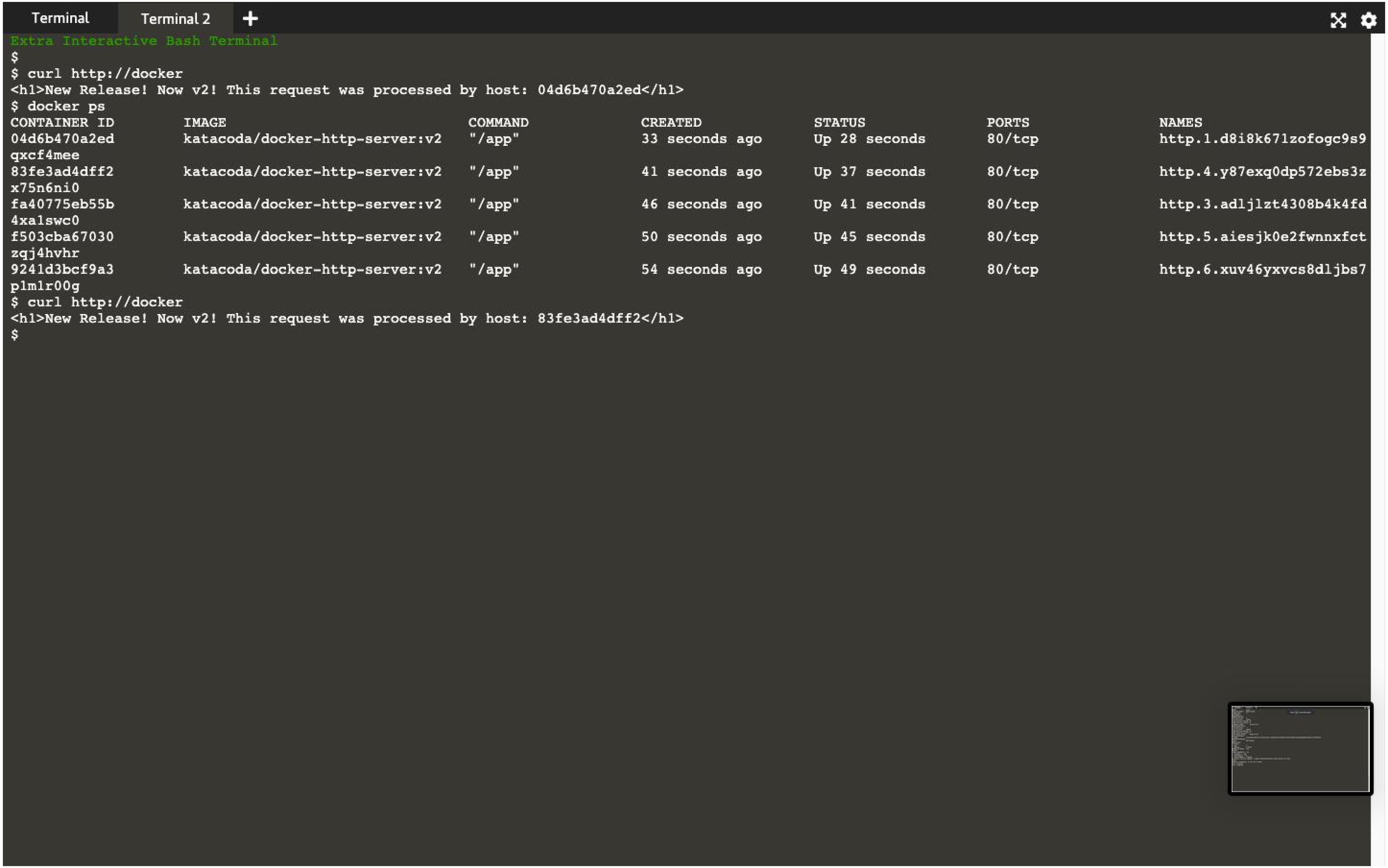


```
;; Query time: 0 msec
;; SERVER: 127.0.0.11#53(127.0.0.11)
;; WHEN: Wed May 04 08:26:21 UTC 2022
;; MSG SIZE  rcvd: 42

$ docker run --name=ping --network eg1 alpine ping -c5 http
PING http (10.0.1.2): 56 data bytes
64 bytes from 10.0.1.2: seq=0 ttl=64 time=0.147 ms
docker service create --name app1-web --network app1-network --replicas 4 -p 80:
3000 katacoda/redis-node-docker-example
64 bytes from 10.0.1.2: seq=1 ttl=64 time=0.131 ms
64 bytes from 10.0.1.2: seq=2 ttl=64 time=0.111 ms
docker ps
64 bytes from 10.0.1.2: seq=3 ttl=64 time=0.112 ms
64 bytes from 10.0.1.2: seq=4 ttl=64 time=0.114 ms

--- http ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.111/0.123/0.147 ms
$ docker service inspect http --format="{{.Endpoint.VirtualIPs}}"
curl host01
[{u4o5db2e68ibt66lflh3j8mcx 10.0.1.2/24}]
$ docker inspect --format="{{.NetworkSettings.Networks.eg1.IPAddress}}" $(docker
server | head -n1 | awk '{print $1}')
```

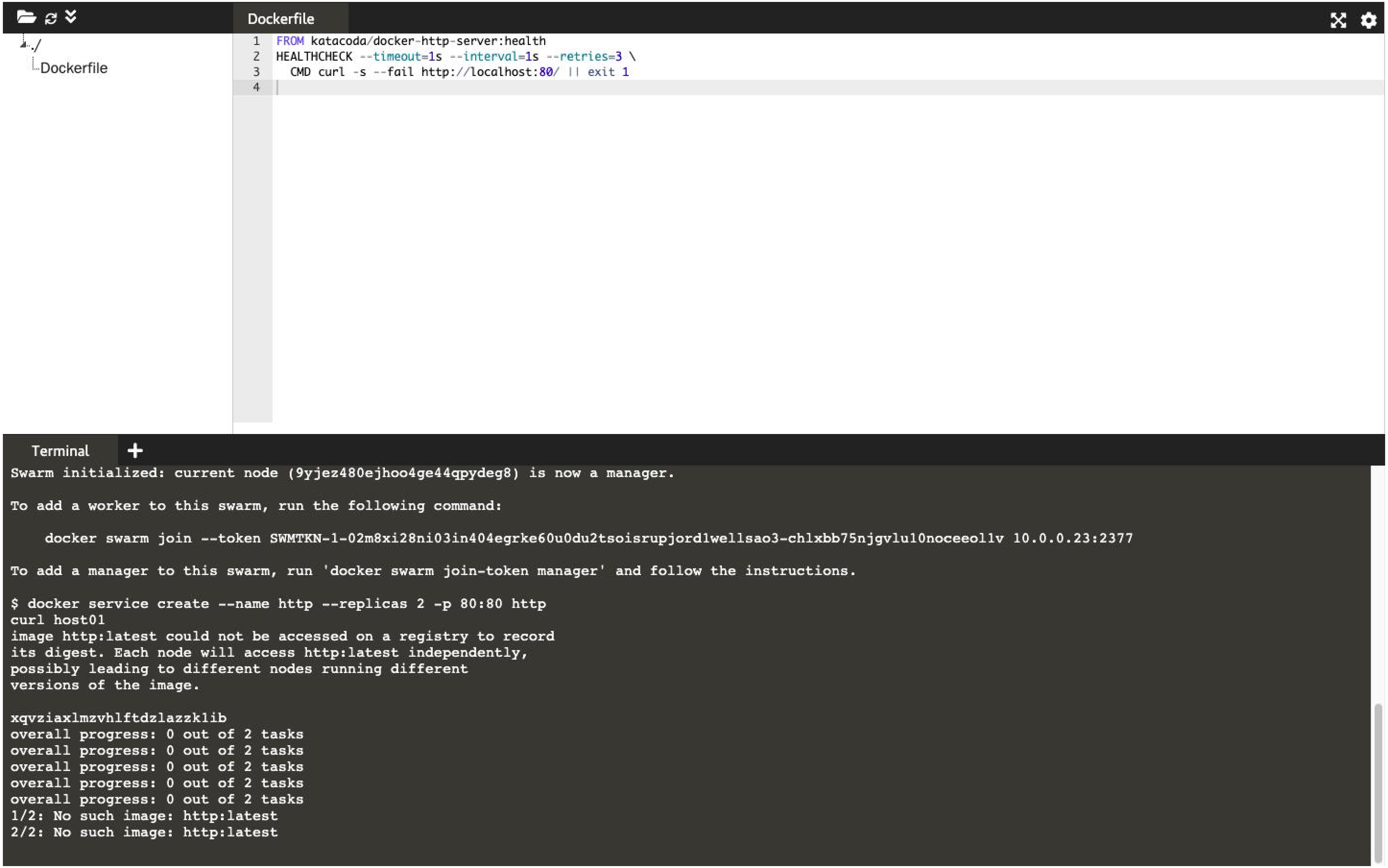
```
Your Interactive Bash Terminal. A safe place to learn and execute commands.

$ docker swarm join 10.0.0.20:2377 --token $(ssh -o StrictHostKeyChecking=no 10.0.0.20 "docker swarm join-token -q worker")
Warning: Permanently added '10.0.0.20' (ECDSA) to the list of known hosts.
This node joined a swarm as a worker.
$ curl host01:81
curl: (7) Failed to connect to host01 port 81: Connection refused
$
```



```
Name:           http
Service Mode:   Replicated
 Replicas:      6
Placement:
UpdateConfig:
 Parallelism:   1
 On failure:    pause
 Monitoring Period: 5s
 Max failure ratio: 0
 Update order:      stop-first
RollbackConfig:
 Parallelism:   1
 On failure:    pause
 Monitoring Period: 5s
 Max failure ratio: 0
 Rollback order:    stop-first
ContainerSpec:
 Image:         katacoda/docker-http-server:v1@sha256:4d7bfcb1e38912d286c5cda63aeddc850a4be16127094ffac
bb7abfc6298c5fa
 Env:           KEY=VALUE
Resources:
 Limits:
  CPU:          2
  Memory:       512MiB
Endpoint Mode:  vip
Ports:
 PublishedPort = 80
  Protocol = tcp
  TargetPort = 80
  PublishMode = ingress
$ docker service update --image katacoda/docker-http-server:v2 http
http
overall progress: 6 out of 6 tasks
1/6: running
2/6: running
```

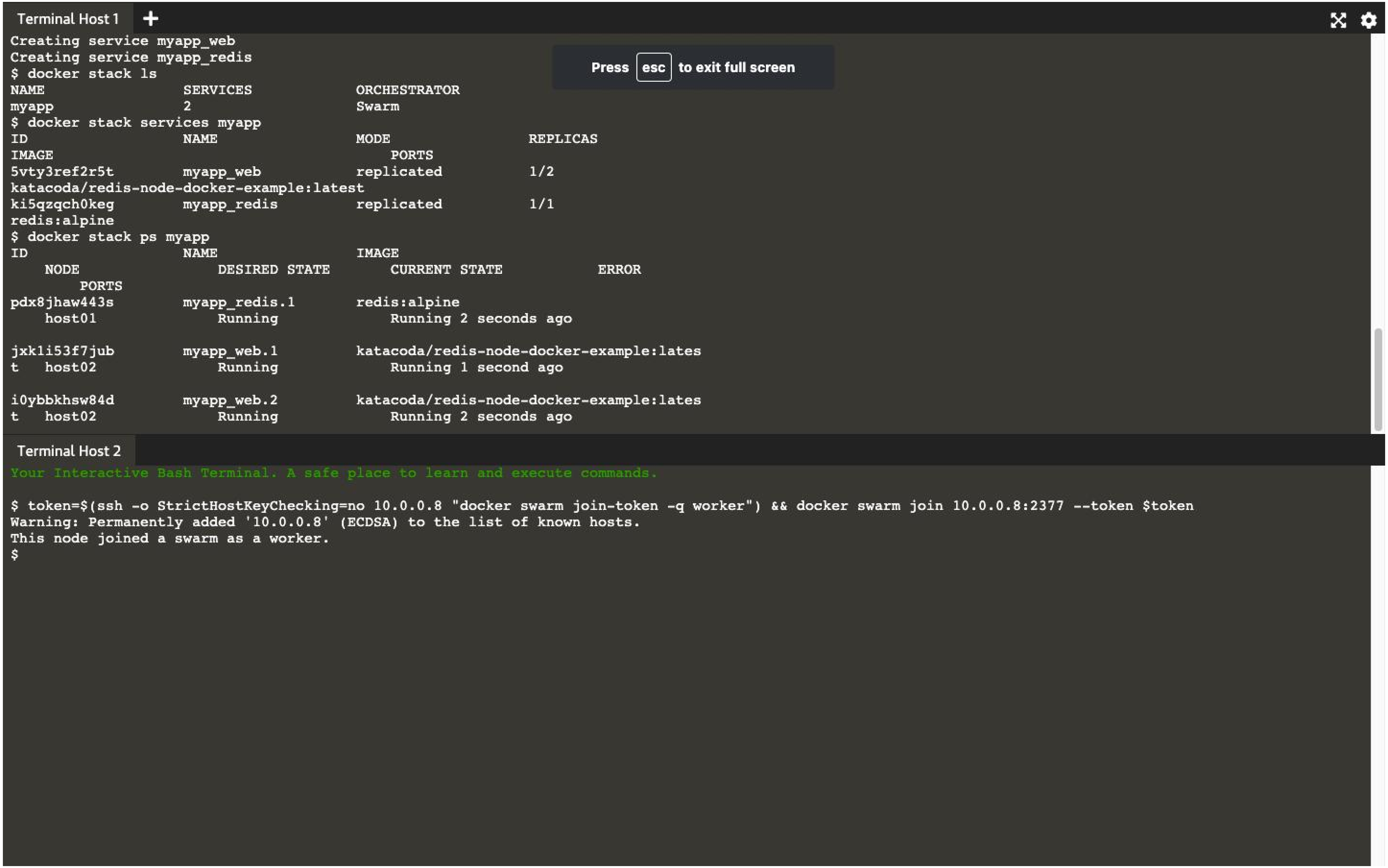


```
Extra Interactive Bash Terminal
$
$ curl http://docker
<h1>New Release! Now v2! This request was processed by host: 04d6b470a2ed</h1>
$ docker ps
CONTAINER ID        IMAGE                            COMMAND             CREATED             STATUS              PORTS               NAMES
04d6b470a2ed        katacoda/docker-http-server:v2   "/app"              33 seconds ago      Up 28 seconds       80/tcp              http.1.d8i8k671zofogc9s9
qxcf4mee
83fe3ad4dff2        katacoda/docker-http-server:v2   "/app"              41 seconds ago      Up 37 seconds       80/tcp              http.4.y87exq0dp572ebs3z
x75n6ni0
fa40775eb55b        katacoda/docker-http-server:v2   "/app"              46 seconds ago      Up 41 seconds       80/tcp              http.3.adljlzt4308b4k4fd
4xa1swc0
f503cba67030        katacoda/docker-http-server:v2   "/app"              50 seconds ago      Up 45 seconds       80/tcp              http.5.aiesjk0e2fwnnxfct
zqj4hvhr
9241d3bcf9a3        katacoda/docker-http-server:v2   "/app"              54 seconds ago      Up 49 seconds       80/tcp              http.6.xuv46yxvcs8dljbs7
p1m1r00g
$ curl http://docker
<h1>New Release! Now v2! This request was processed by host: 83fe3ad4dff2</h1>
$
```



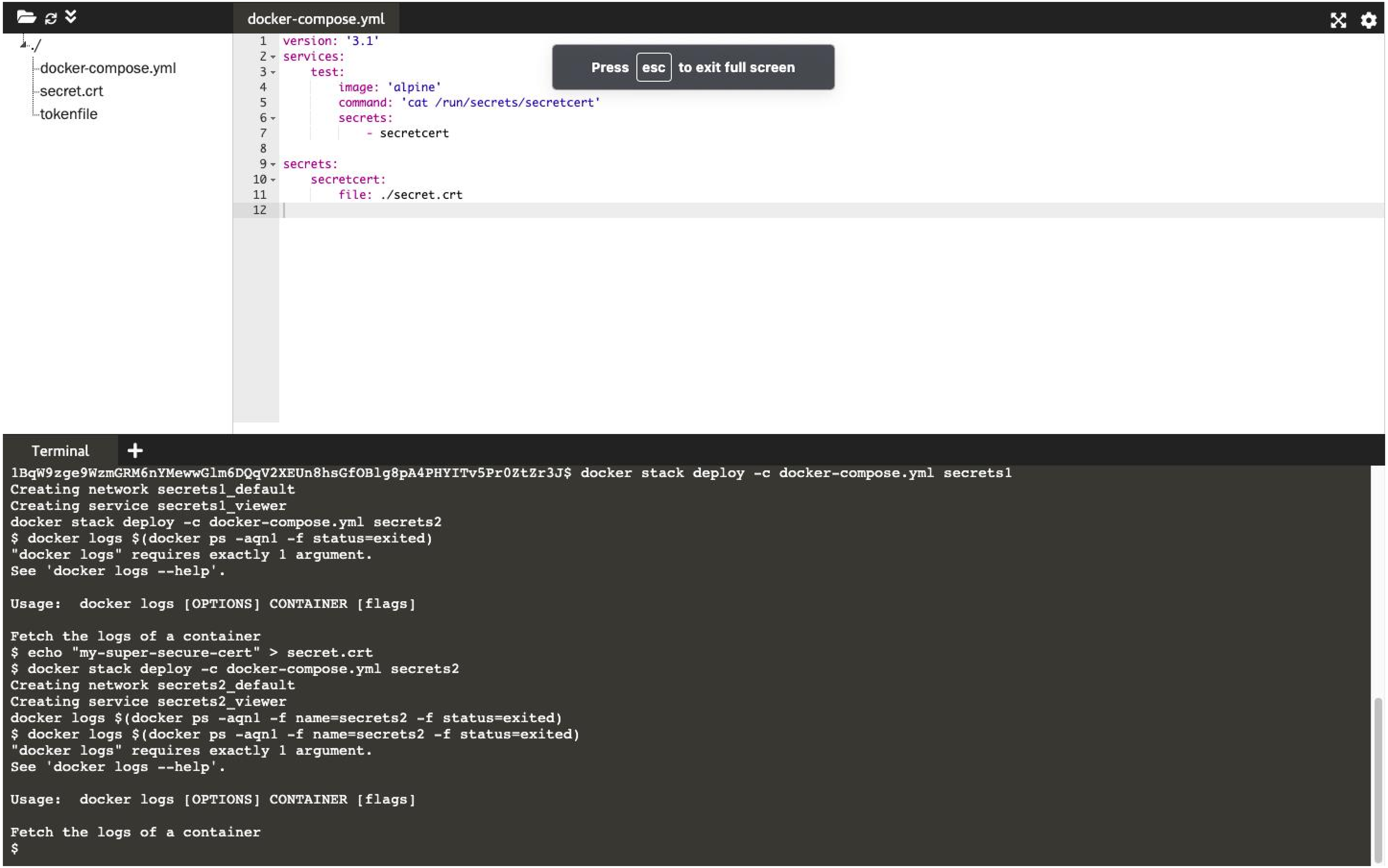
```
FROM katacoda/docker-http-server:health
HEALTHCHECK --timeout=1s --interval=1s --retries=3 \
  CMD curl -s --fail http://localhost:80/ || exit 1
```

```
Swarm initialized: current node (9yjez480ejhoo4ge44qpydeg8) is now a manager.

To add a worker to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-02m8xi28ni03in404egrke60u0du2tsoisrupjord1wellsao3-chlxbb75njgvlu10noceeol1v 10.0.0.23:2377

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.

$ docker service create --name http --replicas 2 -p 80:80 http
curl host01
image http:latest could not be accessed on a registry to record
its digest. Each node will access http:latest independently,
possibly leading to different nodes running different
versions of the image.

xqvziaxlmzvhlftdzlazzk1ib
overall progress: 0 out of 2 tasks
overall progress: 0 out of 2 tasks
overall progress: 0 out of 2 tasks
overall progress: 0 out of 2 tasks
overall progress: 0 out of 2 tasks
1/2: No such image: http:latest
2/2: No such image: http:latest
```



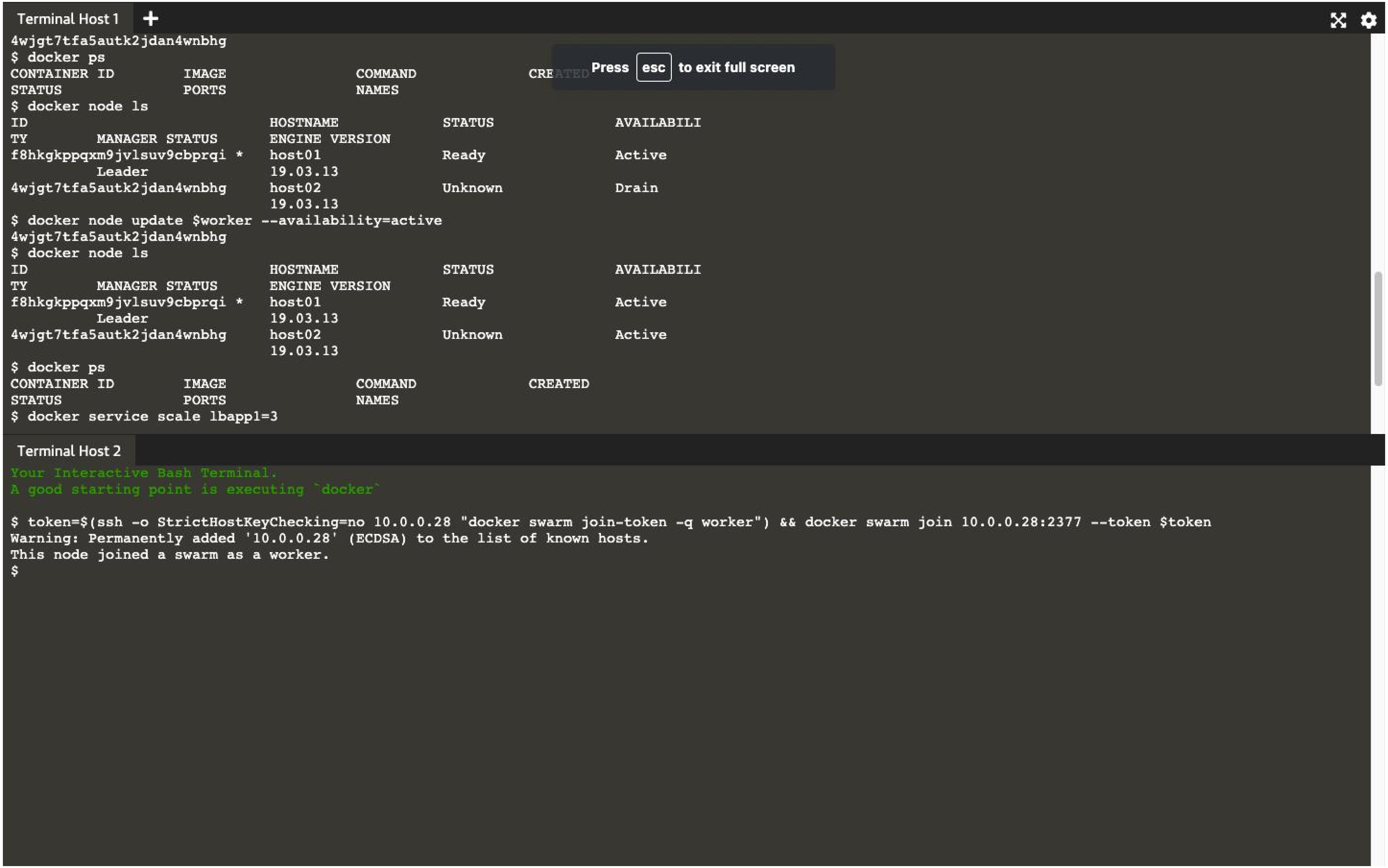
```
Creating service myapp_web
Creating service myapp_redis
$ docker stack ls
NAME                SERVICES            ORCHESTRATOR
myapp               2                   Swarm
$ docker stack services myapp
ID                  NAME                MODE                REPLICAS
IMAGE                                       PORTS
5vty3ref2r5t        myapp_web           replicated          1/2
katacoda/redis-node-docker-example:latest
ki5qzqch0keg        myapp_redis         replicated          1/1
redis:alpine
$ docker stack ps myapp
ID                  NAME                IMAGE
    NODE                DESIRED STATE       CURRENT STATE           ERROR
        PORTS
pdx8jhaw443s        myapp_redis.1       redis:alpine
    host01              Running             Running 2 seconds ago

jxk1i53f7jub        myapp_web.1         katacoda/redis-node-docker-example:lates
t   host02              Running             Running 1 second ago

i0ybbkhsw84d        myapp_web.2         katacoda/redis-node-docker-example:lates
t   host02              Running             Running 2 seconds ago
```

```
Your Interactive Bash Terminal. A safe place to learn and execute commands.

$ token=$(ssh -o StrictHostKeyChecking=no 10.0.0.8 "docker swarm join-token -q worker") && docker swarm join 10.0.0.8:2377 --token $token
Warning: Permanently added '10.0.0.8' (ECDSA) to the list of known hosts.
This node joined a swarm as a worker.
$
```



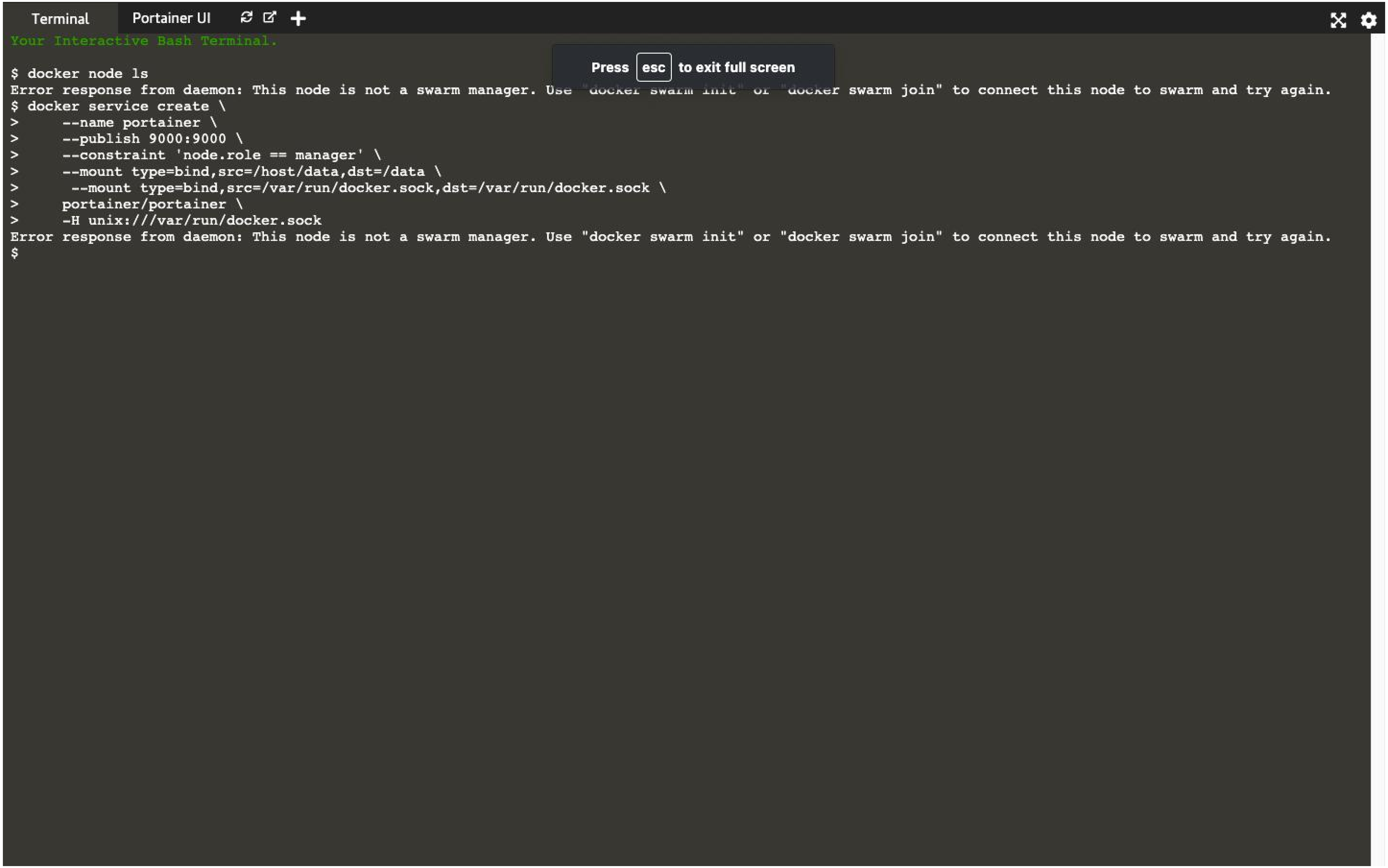
```yaml
version: '3.1'
services:
    test:
        image: 'alpine'
        command: 'cat /run/secrets/secretcert'
        secrets:
            - secretcert

secrets:
    secretcert:
        file: ./secret.crt

```

```
1BqW9zge9WzmGRM6nYMewwGlm6DQqV2XEUn8hsGfOBlg8pA4PHYITv5Pr0ZtZr3J$ docker stack deploy -c docker-compose.yml secrets1
Creating network secrets1_default
Creating service secrets1_viewer
docker stack deploy -c docker-compose.yml secrets2
$ docker logs $(docker ps -aqn1 -f status=exited)
"docker logs" requires exactly 1 argument.
See 'docker logs --help'.

Usage:  docker logs [OPTIONS] CONTAINER [flags]

Fetch the logs of a container
$ echo "my-super-secure-cert" > secret.crt
$ docker stack deploy -c docker-compose.yml secrets2
Creating network secrets2_default
Creating service secrets2_viewer
docker logs $(docker ps -aqn1 -f name=secrets2 -f status=exited)
$ docker logs $(docker ps -aqn1 -f name=secrets2 -f status=exited)
"docker logs" requires exactly 1 argument.
See 'docker logs --help'.

Usage:  docker logs [OPTIONS] CONTAINER [flags]

Fetch the logs of a container
$
```

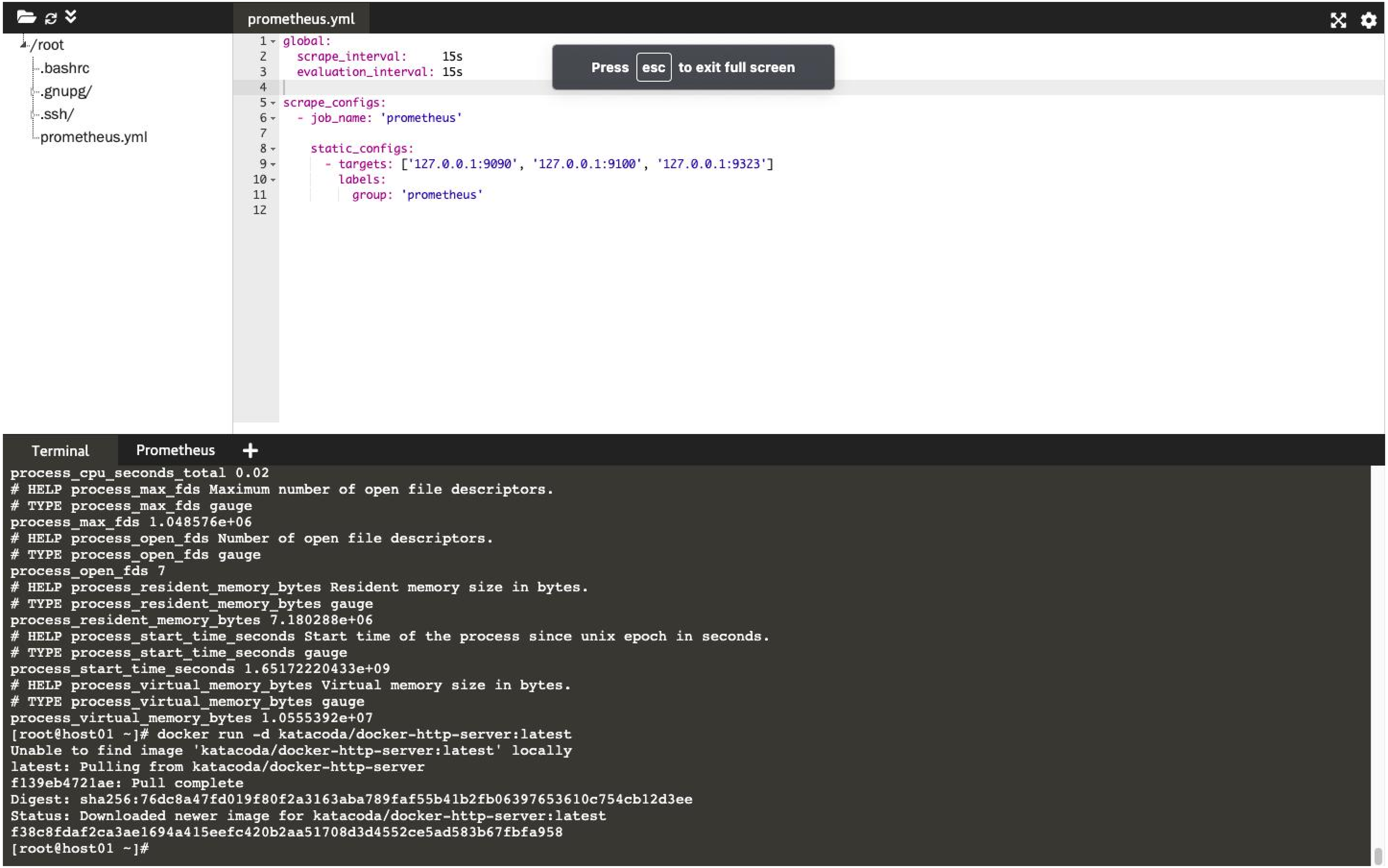


```
4wjgt7tfa5autk2jdan4wnbhg
$ docker ps
CONTAINER ID        IMAGE               COMMAND             CRE
STATUS              PORTS               NAMES
$ docker node ls
ID                            HOSTNAME            STATUS              AVAILABILI
TY          MANAGER STATUS      ENGINE VERSION
f8hkgkppqxm9jvlsuv9cbprqi *   host01              Ready               Active
          Leader              19.03.13
4wjgt7tfa5autk2jdan4wnbhg     host02              Unknown             Drain
                              19.03.13
$ docker node update $worker --availability=active
4wjgt7tfa5autk2jdan4wnbhg
$ docker node ls
ID                            HOSTNAME            STATUS              AVAILABILI
TY          MANAGER STATUS      ENGINE VERSION
f8hkgkppqxm9jvlsuv9cbprqi *   host01              Ready               Active
          Leader              19.03.13
4wjgt7tfa5autk2jdan4wnbhg     host02              Unknown             Active
                              19.03.13
$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS              PORTS               NAMES
$ docker service scale lbapp1=3
```

```
Your Interactive Bash Terminal.
A good starting point is executing `docker`

$ token=$(ssh -o StrictHostKeyChecking=no 10.0.0.28 "docker swarm join-token -q worker") && docker swarm join 10.0.0.28:2377 --token $token
Warning: Permanently added '10.0.0.28' (ECDSA) to the list of known hosts.
This node joined a swarm as a worker.
$
```



```
Your Interactive Bash Terminal.

$ docker node ls
Error response from daemon: This node is not a swarm manager. Use "docker swarm init" or "docker swarm join" to connect this node to swarm and try again.
$ docker service create \
>     --name portainer \
>     --publish 9000:9000 \
>     --constraint 'node.role == manager' \
>     --mount type=bind,src=/host/data,dst=/data \
>      --mount type=bind,src=/var/run/docker.sock,dst=/var/run/docker.sock \
>     portainer/portainer \
>     -H unix:///var/run/docker.sock
Error response from daemon: This node is not a swarm manager. Use "docker swarm init" or "docker swarm join" to connect this node to swarm and try again.
$
```



```yaml
global:
  scrape_interval:     15s
  evaluation_interval: 15s

scrape_configs:
  - job_name: 'prometheus'

    static_configs:
      - targets: ['127.0.0.1:9090', '127.0.0.1:9100', '127.0.0.1:9323']
        labels:
          group: 'prometheus'

```

```
process_cpu_seconds_total 0.02
# HELP process_max_fds Maximum number of open file descriptors.
# TYPE process_max_fds gauge
process_max_fds 1.048576e+06
# HELP process_open_fds Number of open file descriptors.
# TYPE process_open_fds gauge
process_open_fds 7
# HELP process_resident_memory_bytes Resident memory size in bytes.
# TYPE process_resident_memory_bytes gauge
process_resident_memory_bytes 7.180288e+06
# HELP process_start_time_seconds Start time of the process since unix epoch in seconds.
# TYPE process_start_time_seconds gauge
process_start_time_seconds 1.65172220433e+09
# HELP process_virtual_memory_bytes Virtual memory size in bytes.
# TYPE process_virtual_memory_bytes gauge
process_virtual_memory_bytes 1.0555392e+07
[root@host01 ~]# docker run -d katacoda/docker-http-server:latest
Unable to find image 'katacoda/docker-http-server:latest' locally
latest: Pulling from katacoda/docker-http-server
f139eb4721ae: Pull complete
Digest: sha256:76dc8a47fd019f80f2a3163aba789faf55b41b2fb06397653610c754cb12d3ee
Status: Downloaded newer image for katacoda/docker-http-server:latest
f38c8fdaf2ca3ae1694a415eefc420b2aa51708d3d4552ce5ad583b67fbfa958
[root@host01 ~]#
```