

ANDREW DENNER, CIALUG

REVERSE PROXIES AND LOAD BALANCERS

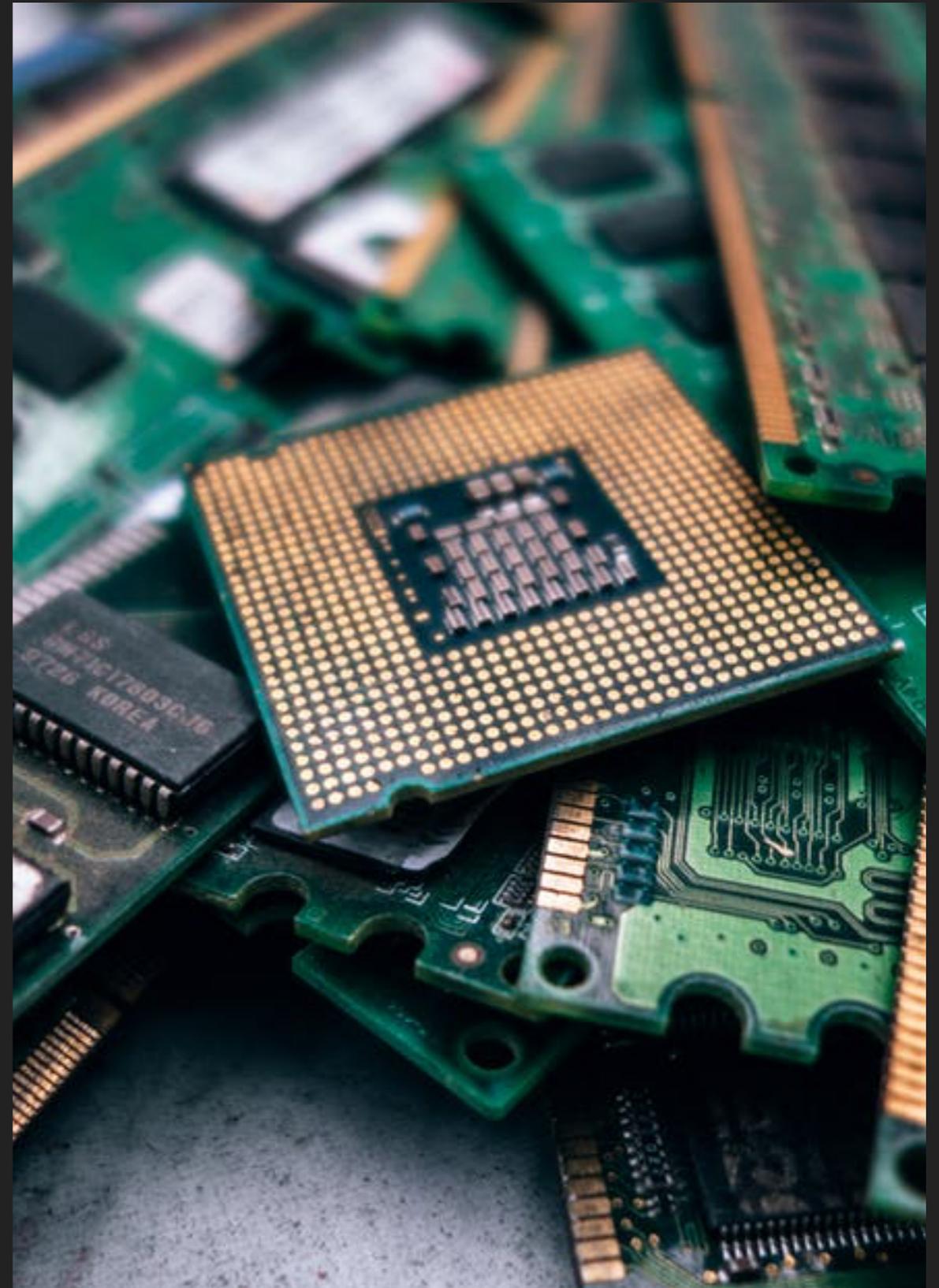
A LITTLE ABOUT ME

- ▶ Slides will be posted to <https://denner.co> later tonight
- ▶ Twitter: @adenner
- ▶ Email: denner@gmail.com



WELCOME TO CIALUG

- ▶ Website: <http://cialug.org>
- ▶ Mailing list
- ▶ IRC and Slack
- ▶ Videos on youtube



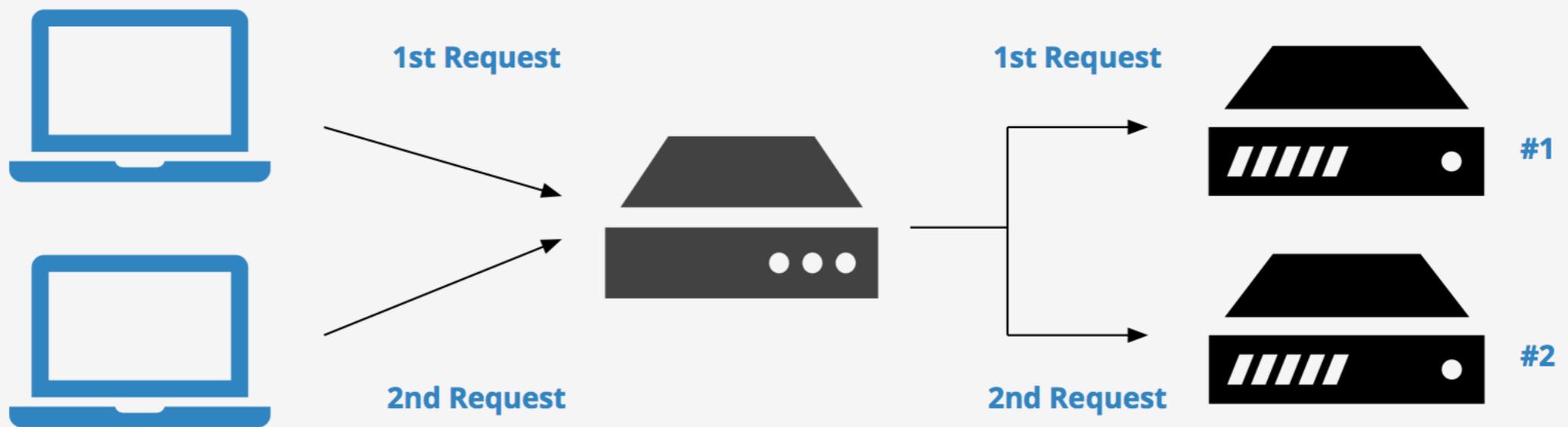
A close-up photograph of a microscope. The microscope is white and black, with a prominent light source on the left side that is glowing brightly. The light is directed towards a specimen on the right, which is also glowing. The background is dark and out of focus. The text "NOW ON TO OUR" is overlaid on the left side of the image, and "FEATURE PRESENTATION" is overlaid in large blue letters across the bottom.

NOW ON TO OUR

FEATURE PRESENTATION

REVERSE PROXY VS LOAD BALANCER

- ▶ A reverse proxy accepts a request from a client, forwards it to a server that can fulfill it, and returns the server's response to the client. A load balancer distributes incoming client requests among a group of servers, in each case returning the response from the selected server to the appropriate client.



Load Balancing



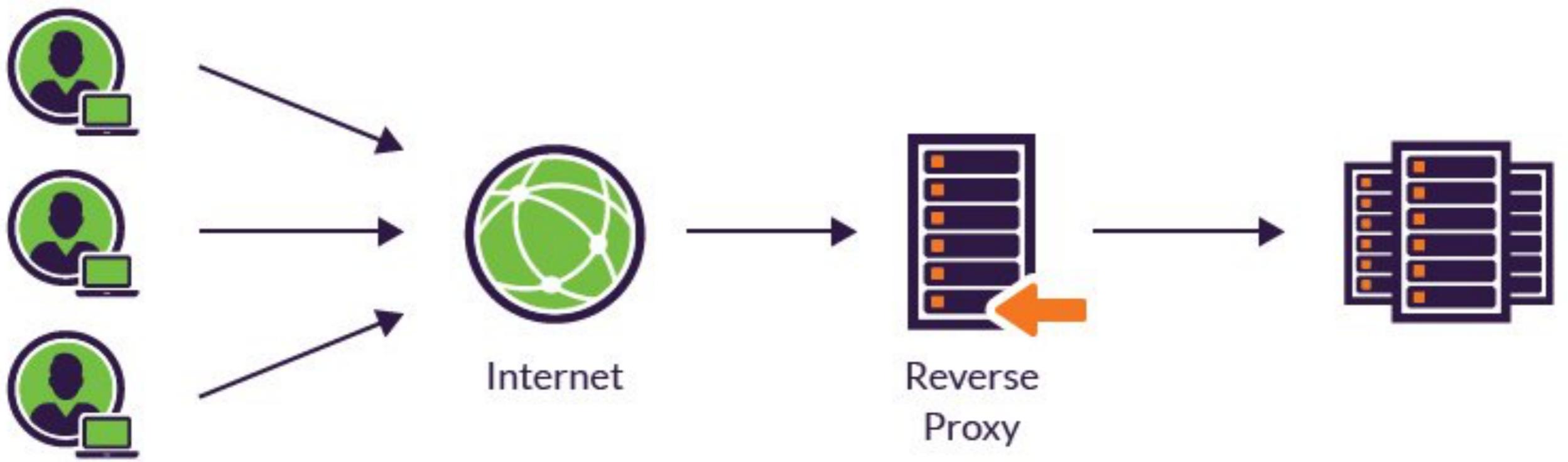
Load Balancer



Server



Client



WHY DO I WANT ONE?

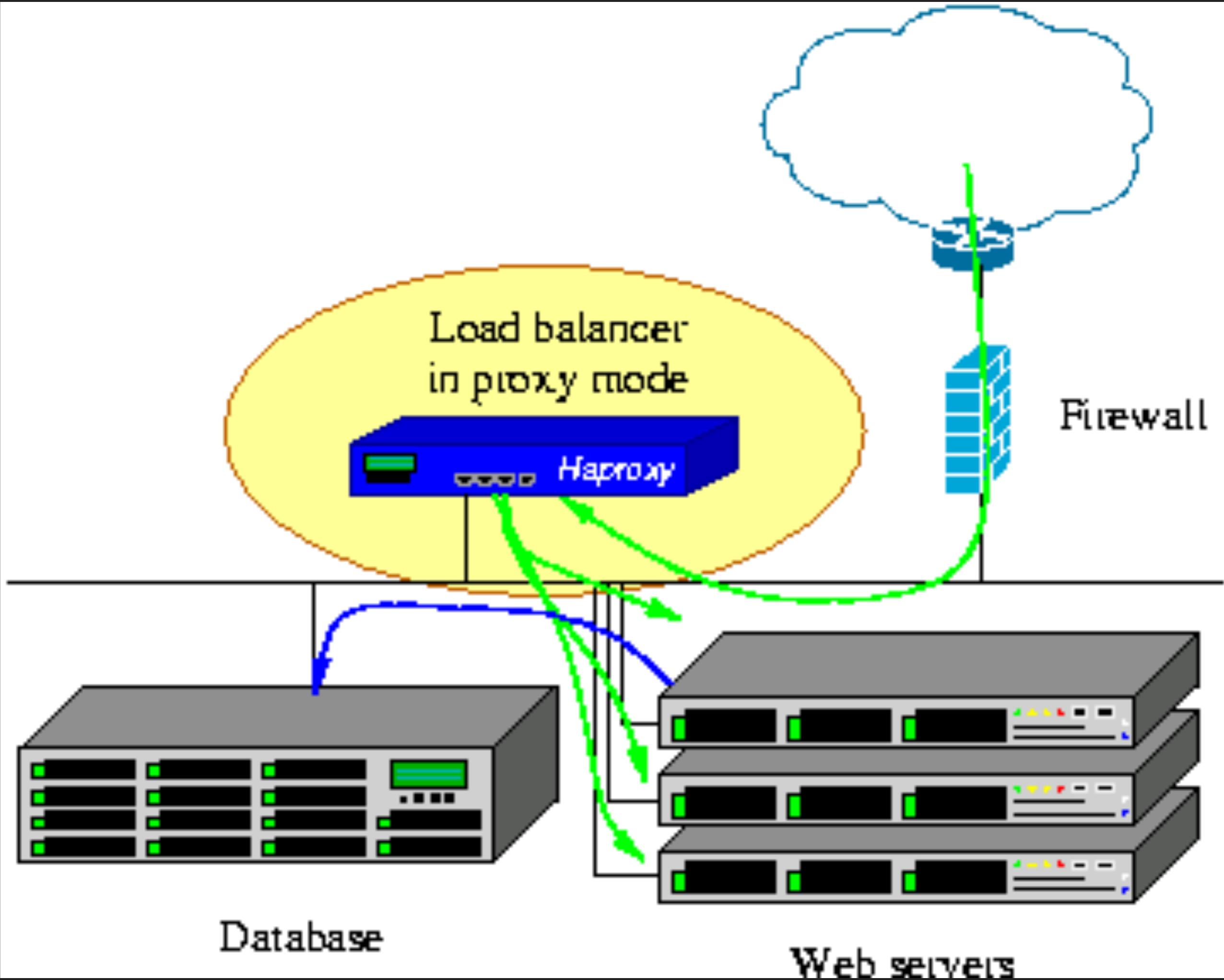
- ▶ SSL offloading
- ▶ URL rewriting
- ▶ Aggregate services
- ▶ High Availability
- ▶ Network cleanup

WHAT ARE MY OPTIONS

- ▶ Apache
 - ▶ Grand daddy of them all
 - ▶ A heavy hammer
 - ▶ Process based
- ▶ Nginx
 - ▶ Younger nimble upstart
 - ▶ Originally designed for reverse proxy
 - ▶ Thread based

WHAT ARE MY OPTIONS (CONT)

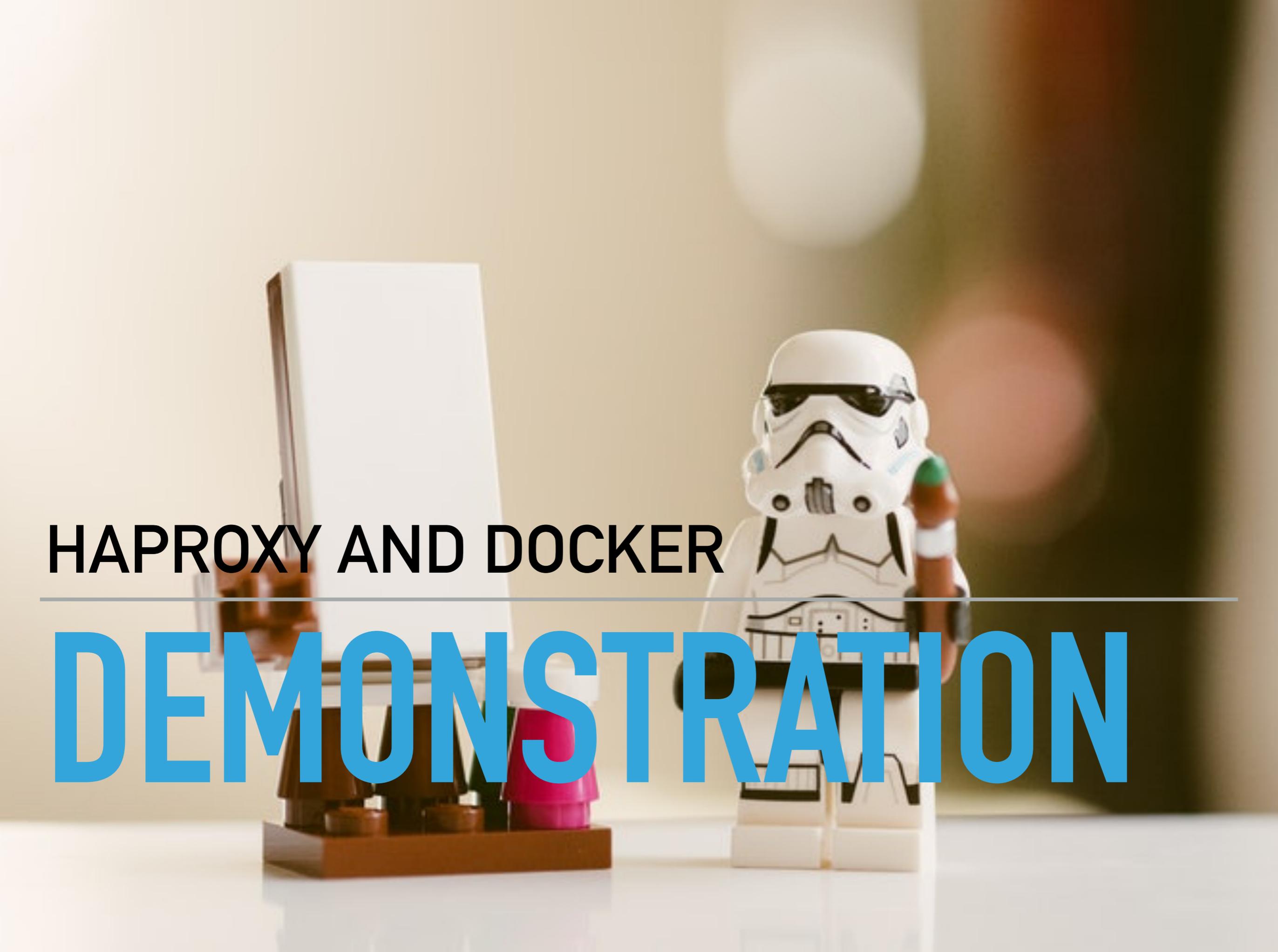
- ▶ HAProxy
 - ▶ TCP and HTTP reverse proxy
 - ▶ Purpose built for only one purpose
 - ▶ Secure and reliable



A LOOK AT THE DOCUMENTS:

- ▶ <http://cbonte.github.io/haproxy-dconv/1.9/intro.html>





HAPROXY AND DOCKER

DEMONSTRATION

BRIEF DESCRIPTION:

- ▶ Directions from writeup: <https://medium.com/@nirgn/load-balancing-applications-with-haproxy-and-docker-d719b7c5b231>
- ▶ Using docker, docker compose, docker swarm, node.js and HAProxy

INDEX.JS

```
var http = require('http');
var os = require('os');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/html'});
  res.end(`<h1>I'm ${os.hostname()}</h1>`);
}).listen(8080);
```

```
denneran@US-GVM2CM2:/mnt/c/Users/denneran/OneDrive - AgCompany/dockerdemo/nodeapp$ docker build -t awesome .
Sending build context to Docker daemon 5.632kB
Step 1/5 : FROM node
---> 0e2811757f93
Step 2/5 : RUN mkdir -p /usr/src/app
---> Using cache
---> 9faf17410650
Step 3/5 : COPY index.js /usr/src/app
---> Using cache
---> b7212dea8325
Step 4/5 : EXPOSE 8080
---> Using cache
---> c23083bb07a5
Step 5/5 : CMD [ "node", "/usr/src/app/index" ]
---> Using cache
---> f00f920d9827
successfully built f00f920d9827
successfully tagged awesome:latest
SECURITY WARNING: You are building a Docker image from windows against a non-Windows Docker host. All files and directories added to build stages will only be accessible in Windows containers.
denneran@US-GVM2CM2:/mnt/c/Users/denneran/OneDrive - AgCompany/dockerdemo/nodeapp$ _
```

```
denneran@US-GVM2CM2:/mnt/c/Users/denneran/OneDrive - AgCompany/dockerdemo/nodeapp$
denneran@US-GVM2CM2:/mnt/c/Users/denneran/OneDrive - AgCompany/dockerdemo/nodeapp$ docker stack deploy --compose-file=docker-compose.yml
Creating service prod_awesome
Creating service prod_proxy
denneran@US-GVM2CM2:/mnt/c/Users/denneran/OneDrive - AgCompany/dockerdemo/nodeapp$ curl localhost
curl: (52) Empty reply from server
denneran@US-GVM2CM2:/mnt/c/Users/denneran/OneDrive - AgCompany/dockerdemo/nodeapp$ curl localhost
<h1>I'm eb56dc416b1b</h1>denneran@US-GVM2CM2:/mnt/c/Users/denneran/OneDrive - AgCompany/dockerdemo/nodeapp$ curl localhost
<h1>I'm 3c936117cab4</h1>denneran@US-GVM2CM2:/mnt/c/Users/denneran/OneDrive - AgCompany/dockerdemo/nodeapp$ curl localhost
<h1>I'm 5743526d9ea7</h1>denneran@US-GVM2CM2:/mnt/c/Users/denneran/OneDrive - AgCompany/dockerdemo/nodeapp$ curl localhost
<h1>I'm dd8220ef213b</h1>denneran@US-GVM2CM2:/mnt/c/Users/denneran/OneDrive - AgCompany/dockerdemo/nodeapp$ curl localhost
<h1>I'm 6621ba987d4b</h1>denneran@US-GVM2CM2:/mnt/c/Users/denneran/OneDrive - AgCompany/dockerdemo/nodeapp$ curl localhost
<h1>I'm 05013e6fcfcf</h1>denneran@US-GVM2CM2:/mnt/c/Users/denneran/OneDrive - AgCompany/dockerdemo/nodeapp$ curl localhost
<h1>I'm 273a48608cd0</h1>denneran@US-GVM2CM2:/mnt/c/Users/denneran/OneDrive - AgCompany/dockerdemo/nodeapp$ curl localhost
<h1>I'm 44ae63e773fe</h1>denneran@US-GVM2CM2:/mnt/c/Users/denneran/OneDrive - AgCompany/dockerdemo/nodeapp$ curl localhost
<h1>I'm 9dd0aee43923</h1>denneran@US-GVM2CM2:/mnt/c/Users/denneran/OneDrive - AgCompany/dockerdemo/nodeapp$ curl localhost
<h1>I'm adb2589c357b</h1>denneran@US-GVM2CM2:/mnt/c/Users/denneran/OneDrive - AgCompany/dockerdemo/nodeapp$ curl localhost
<h1>I'm 1ecb3371fdb0</h1>denneran@US-GVM2CM2:/mnt/c/Users/denneran/OneDrive - AgCompany/dockerdemo/nodeapp$ curl localhost
<h1>I'm cce24af07484</h1>denneran@US-GVM2CM2:/mnt/c/Users/denneran/OneDrive - AgCompany/dockerdemo/nodeapp$ curl localhost
<h1>I'm 249af8723214</h1>denneran@US-GVM2CM2:/mnt/c/Users/denneran/OneDrive - AgCompany/dockerdemo/nodeapp$ curl localhost
<h1>I'm 62cab6b0d6ea</h1>denneran@US-GVM2CM2:/mnt/c/Users/denneran/OneDrive - AgCompany/dockerdemo/nodeapp$ curl localhost
<h1>I'm 8e2784018b60</h1>denneran@US-GVM2CM2:/mnt/c/Users/denneran/OneDrive - AgCompany/dockerdemo/nodeapp$ curl localhost
<h1>I'm bd107e68c20e</h1>denneran@US-GVM2CM2:/mnt/c/Users/denneran/OneDrive - AgCompany/dockerdemo/nodeapp$ curl localhost
<h1>I'm 6d8da2736105</h1>denneran@US-GVM2CM2:/mnt/c/Users/denneran/OneDrive - AgCompany/dockerdemo/nodeapp$ _
```

```
denneran@US-GVM2CM2:/mnt/c/Users/denneran/OneDrive - AgCompany/dockerdemo/nodeapp$
denneran@US-GVM2CM2:/mnt/c/Users/denneran/OneDrive - AgCompany/dockerdemo/nodeapp$ docker stack deploy --compose-file=docker-compose.yml prod
Creating service prod_awesome
Creating service prod_proxy
denneran@US-GVM2CM2:/mnt/c/Users/denneran/OneDrive - AgCompany/dockerdemo/nodeapp$ curl localhost
curl: (52) Empty reply from server
denneran@US-GVM2CM2:/mnt/c/Users/denneran/OneDrive - AgCompany/dockerdemo/nodeapp$ curl localhost
<h1>I'm eb56dc416b1b</h1>denneran@US-GVM2CM2:/mnt/c/Users/denneran/OneDrive - AgCompany/dockerdemo/nodeapp$ curl localhost
<h1>I'm 3c936117cab4</h1>denneran@US-GVM2CM2:/mnt/c/Users/denneran/OneDrive - AgCompany/dockerdemo/nodeapp$ curl localhost
<h1>I'm 5743526d9ea7</h1>denneran@US-GVM2CM2:/mnt/c/Users/denneran/OneDrive - AgCompany/dockerdemo/nodeapp$ curl localhost
<h1>I'm dd8220ef213b</h1>denneran@US-GVM2CM2:/mnt/c/Users/denneran/OneDrive - AgCompany/dockerdemo/nodeapp$ curl localhost
<h1>I'm 6621ba987d4b</h1>denneran@US-GVM2CM2:/mnt/c/Users/denneran/OneDrive - AgCompany/dockerdemo/nodeapp$ curl localhost
<h1>I'm 05013e6fcfcf</h1>denneran@US-GVM2CM2:/mnt/c/Users/denneran/OneDrive - AgCompany/dockerdemo/nodeapp$ curl localhost
<h1>I'm 273a48608cd0</h1>denneran@US-GVM2CM2:/mnt/c/Users/denneran/OneDrive - AgCompany/dockerdemo/nodeapp$ curl localhost
<h1>I'm 44ae63e773fe</h1>denneran@US-GVM2CM2:/mnt/c/Users/denneran/OneDrive - AgCompany/dockerdemo/nodeapp$ curl localhost
<h1>I'm 9dd0aee43923</h1>denneran@US-GVM2CM2:/mnt/c/Users/denneran/OneDrive - AgCompany/dockerdemo/nodeapp$ curl localhost
<h1>I'm adb2589c357b</h1>denneran@US-GVM2CM2:/mnt/c/Users/denneran/OneDrive - AgCompany/dockerdemo/nodeapp$ curl localhost
<h1>I'm 1ecb3371fdb0</h1>denneran@US-GVM2CM2:/mnt/c/Users/denneran/OneDrive - AgCompany/dockerdemo/nodeapp$ curl localhost
<h1>I'm cce24af07484</h1>denneran@US-GVM2CM2:/mnt/c/Users/denneran/OneDrive - AgCompany/dockerdemo/nodeapp$ curl localhost
<h1>I'm 249af8723214</h1>denneran@US-GVM2CM2:/mnt/c/Users/denneran/OneDrive - AgCompany/dockerdemo/nodeapp$ curl localhost
<h1>I'm 62cab6b0d6ea</h1>denneran@US-GVM2CM2:/mnt/c/Users/denneran/OneDrive - AgCompany/dockerdemo/nodeapp$ curl localhost
<h1>I'm 8e2784018b60</h1>denneran@US-GVM2CM2:/mnt/c/Users/denneran/OneDrive - AgCompany/dockerdemo/nodeapp$ curl localhost
<h1>I'm bd107e68c20e</h1>denneran@US-GVM2CM2:/mnt/c/Users/denneran/OneDrive - AgCompany/dockerdemo/nodeapp$ curl localhost
<h1>I'm 6d8da2736105</h1>denneran@US-GVM2CM2:/mnt/c/Users/denneran/OneDrive - AgCompany/dockerdemo/nodeapp$ docker service rm prod_proxy file=docker-compose.yml prod
prod_proxy
denneran@US-GVM2CM2:/mnt/c/Users/denneran/OneDrive - AgCompany/dockerdemo/nodeapp$ docker service rm prod_proxy
Error: No such service: prod_proxy
denneran@US-GVM2CM2:/mnt/c/Users/denneran/OneDrive - AgCompany/dockerdemo/nodeapp$ docker service rm prod_awesome
prod_awesome
denneran@US-GVM2CM2:/mnt/c/Users/denneran/OneDrive - AgCompany/dockerdemo/nodeapp$ docker stack deploy --compose-file=docker-compose.yml prod
Creating service prod_awesome
Creating service prod_proxy
denneran@US-GVM2CM2:/mnt/c/Users/denneran/OneDrive - AgCompany/dockerdemo/nodeapp$ docker service rm prod_proxy
prod_proxy
denneran@US-GVM2CM2:/mnt/c/Users/denneran/OneDrive - AgCompany/dockerdemo/nodeapp$ docker service rm prod_awesome
prod_awesome
denneran@US-GVM2CM2:/mnt/c/Users/denneran/OneDrive - AgCompany/dockerdemo/nodeapp$ _
```

DOCKERFILE

```
FROM node
RUN mkdir -p /usr/src/app
COPY index.js /usr/src/app
EXPOSE 8080
CMD [ "node", "/usr/src/app/index" ]
```

DOCKER-COMPOSE.YML

```
version: '3'

services:
  awesome:
    image: awesome
    ports:
      - 8080
    environment:
      - SERVICE_PORTS=8080
    deploy:
      replicas: 20
      update_config:
        parallelism: 5
        delay: 10s
      restart_policy:
        condition: on-failure
        max_attempts: 3
        window: 120s
    networks:
      - web

  proxy:
    image: dockercloud/haproxy
    depends_on:
      - awesome
    environment:
      - BALANCE=leastconn
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock
    ports:
      - 80:80
    networks:
      - web
    deploy:
      placement:
        constraints: [node.role == manager]

networks:
  web:
    driver: overlay
```

```
→ ~ curl http://localhost
<h1>I'm 40d13f415fe2</h1>%
→ ~ curl http://localhost
<h1>I'm a3d9e45fdccb</h1>%
→ ~ curl http://localhost
<h1>I'm 3ecd1a0921d0</h1>%
→ ~ curl http://localhost
<h1>I'm adbd98102eae</h1>%
→ ~ curl http://localhost
<h1>I'm c921ac20429e</h1>%
→ ~ █
```

```
→ ~ docker service ls
ID                NAME                MODE                REPLICAS                IMAGE                PORTS
lx1ep2uq8lkb     prod_awesome        replicated          20/20                   awesome:latest      *:0->8080/tcp
v14zl3q1bsbx     prod_proxy          replicated          1/1                     dockercloud/haproxy:latest *:80->80/tcp
→ ~ █
```

RESOURCES

- ▶ HAProxy
 - ▶ <http://www.haproxy.org>
 - ▶ <http://cbonte.github.io/haproxy-dconv/1.9/intro.html>
- ▶ Nginx
 - ▶ <https://docs.nginx.com/nginx/admin-guide/web-server/reverse-proxy/>
 - ▶

