



Apache mod_proxy

Outline

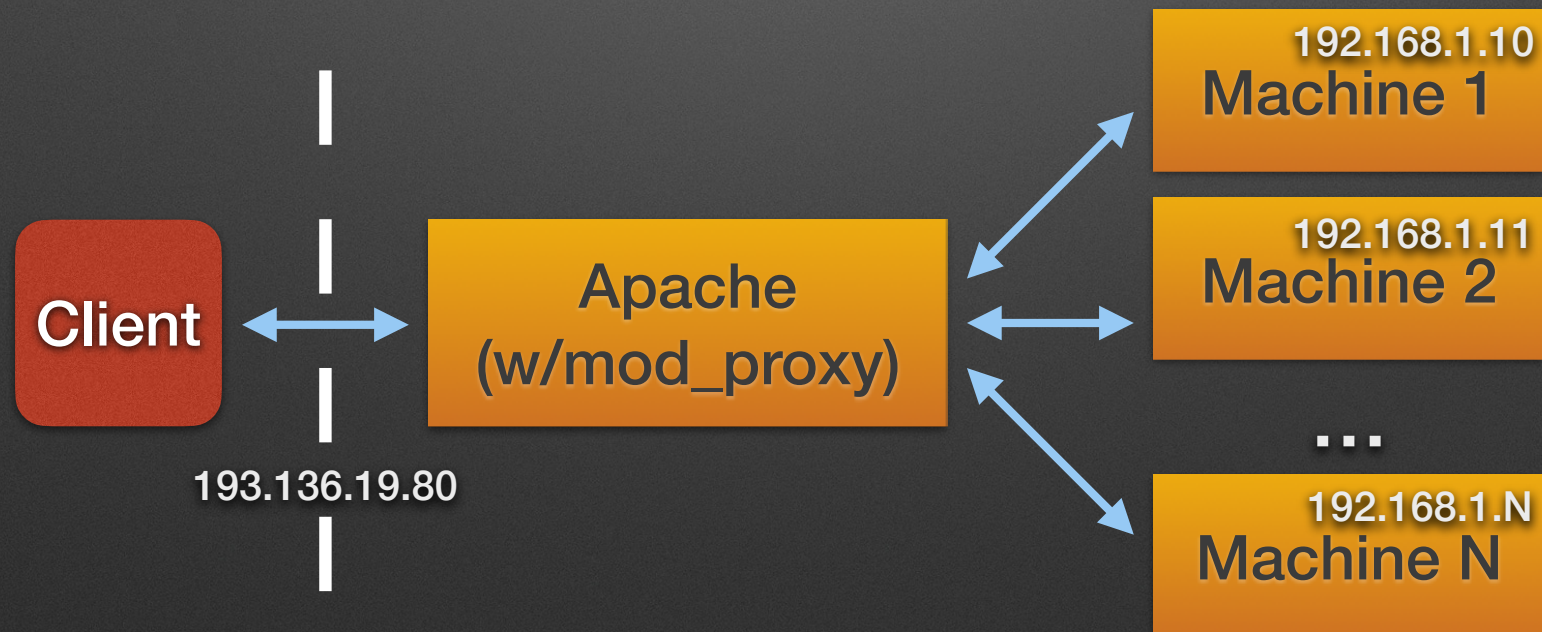
- Introduction
- Configuration
- Example

Introduction

- `mod_proxy` is a proxy module for the Apache HTTP server.
- It implements routing features as well as load balancing.
- It is bundled with apache by default.

Introduction

- With mod_proxy, a machine can be easily configured to be the load balancer.
- The process is transparent to the user.



Configuration

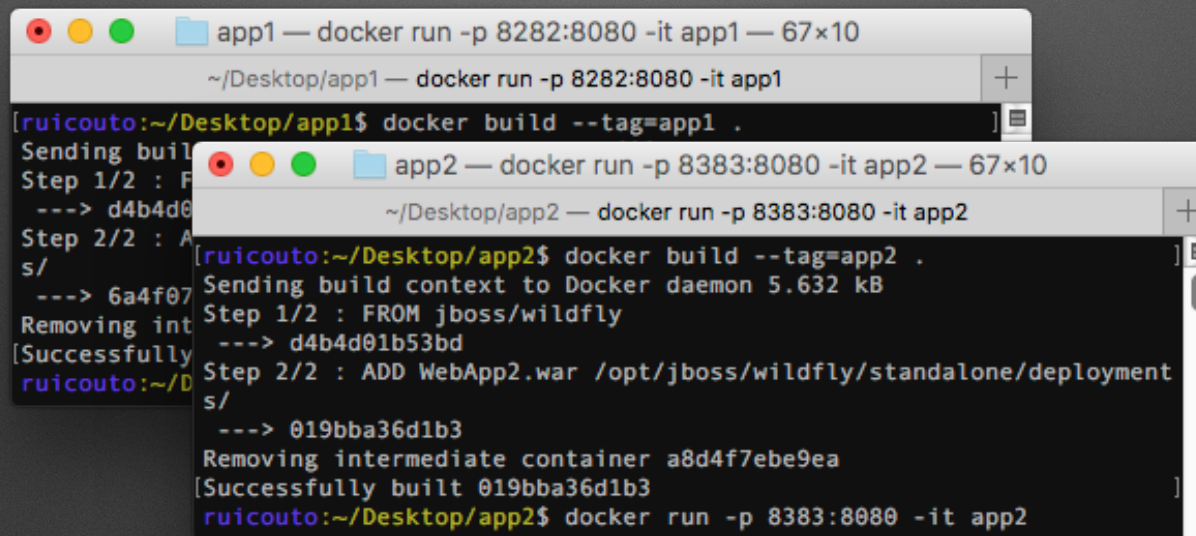
- `mod_proxy` is enabled and configured in the apache configuration file.

```
LoadModule proxy_module libexec/apache2/mod_proxy.so
...
ProxyPass / balancer://mycluster/ stickysession=JSESSIONID|jsessionid
...
ProxyPassReverse / http://localhost:8080/
...

<Proxy balancer://mycluster/>
    BalancerMember http://<url> ... loadfactor=A
    ...
</Proxy>
```

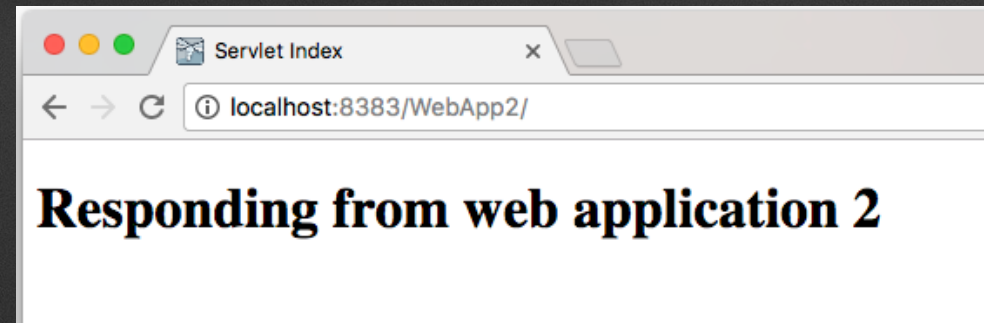
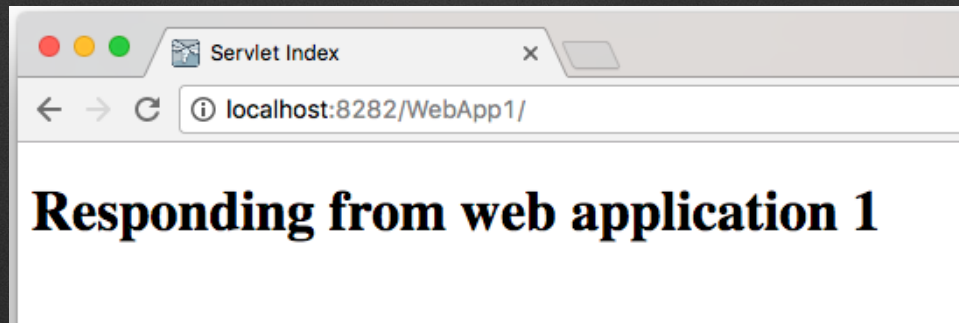
Example

- Two docker containers wit Java web applications were deployed in ports 8282 and 8383.



```
app1 — docker run -p 8282:8080 -it app1 — 67x10
~/Desktop/app1 — docker run -p 8282:8080 -it app1
[ruicouto:~/Desktop/app1$ docker build --tag=app1 .
Sending build context to Docker daemon 5.632 kB
Step 1/2 : FROM jboss/wildfly
---> d4b4d01b53bd
Step 2/2 : ADD WebApp1.war /opt/jboss/wildfly/standalone/deployment
s/
---> 6a4f07
Removing intermediate container a8d4f7ebe9ea
[Successfully built 6a4f07
ruicouto:~/Desktop/app1$ docker run -p 8282:8080 -it app1

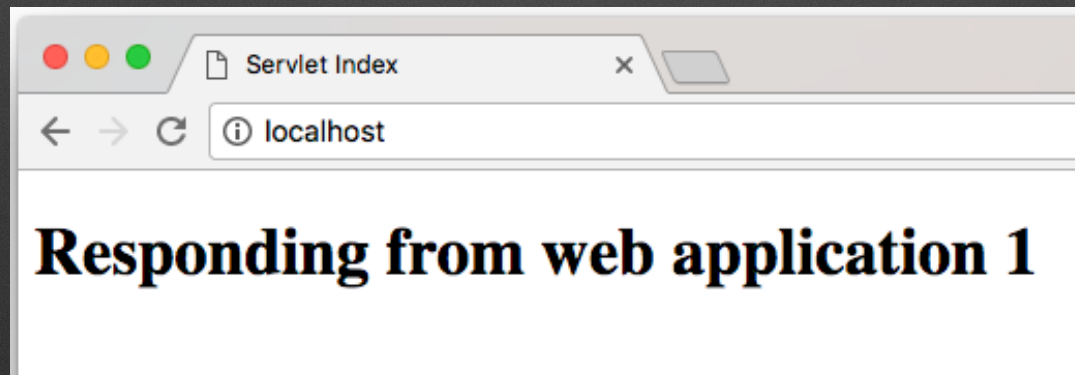
app2 — docker run -p 8383:8080 -it app2 — 67x10
~/Desktop/app2 — docker run -p 8383:8080 -it app2
[ruicouto:~/Desktop/app2$ docker build --tag=app2 .
Sending build context to Docker daemon 5.632 kB
Step 1/2 : FROM jboss/wildfly
---> d4b4d01b53bd
Step 2/2 : ADD WebApp2.war /opt/jboss/wildfly/standalone/deployment
s/
---> 019bba36d1b3
Removing intermediate container a8d4f7ebe9ea
[Successfully built 019bba36d1b3
ruicouto:~/Desktop/app2$ docker run -p 8383:8080 -it app2
```



Example

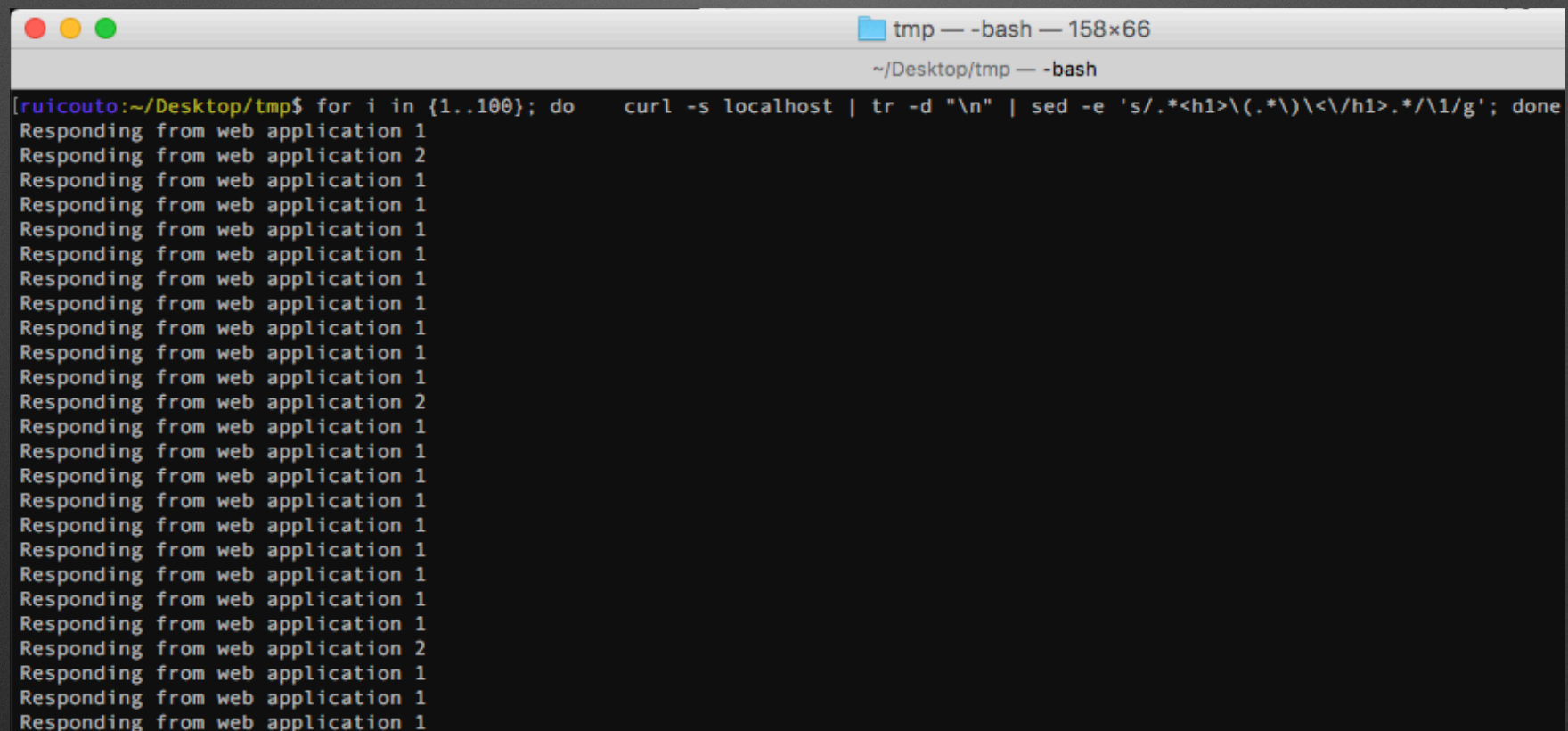
- Apache was configured to route requests on / to both addresses.

```
<Proxy balancer://mycluster/>  
  BalancerMember http://localhost:8282/WebApp1 route=node1 retry=60 loadfactor=80  
  BalancerMember http://localhost:8383/WebApp2 route=node2 retry=60 loadfactor=20  
</Proxy>
```



Example

- For a significative number of requests , it is possible to see that the load is being balanced.



```
tmp — -bash — 158x66
~/Desktop/tmp — -bash
[ruicouto:~/Desktop/tmp$ for i in {1..100}; do curl -s localhost | tr -d "\n" | sed -e 's/.*<h1>\(.*\)\<\/h1>.*\/1/g'; done
Responding from web application 1
Responding from web application 2
Responding from web application 1
Responding from web application 1
Responding from web application 1
Responding from web application 1
Responding from web application 1
Responding from web application 1
Responding from web application 1
Responding from web application 1
Responding from web application 1
Responding from web application 1
Responding from web application 1
Responding from web application 1
Responding from web application 1
Responding from web application 2
Responding from web application 1
Responding from web application 1
Responding from web application 1
Responding from web application 1
Responding from web application 1
Responding from web application 1
Responding from web application 1
Responding from web application 1
Responding from web application 1
Responding from web application 1
Responding from web application 1
Responding from web application 1
Responding from web application 2
Responding from web application 1
Responding from web application 1
Responding from web application 1
```