Load Balancing Internal Web Services
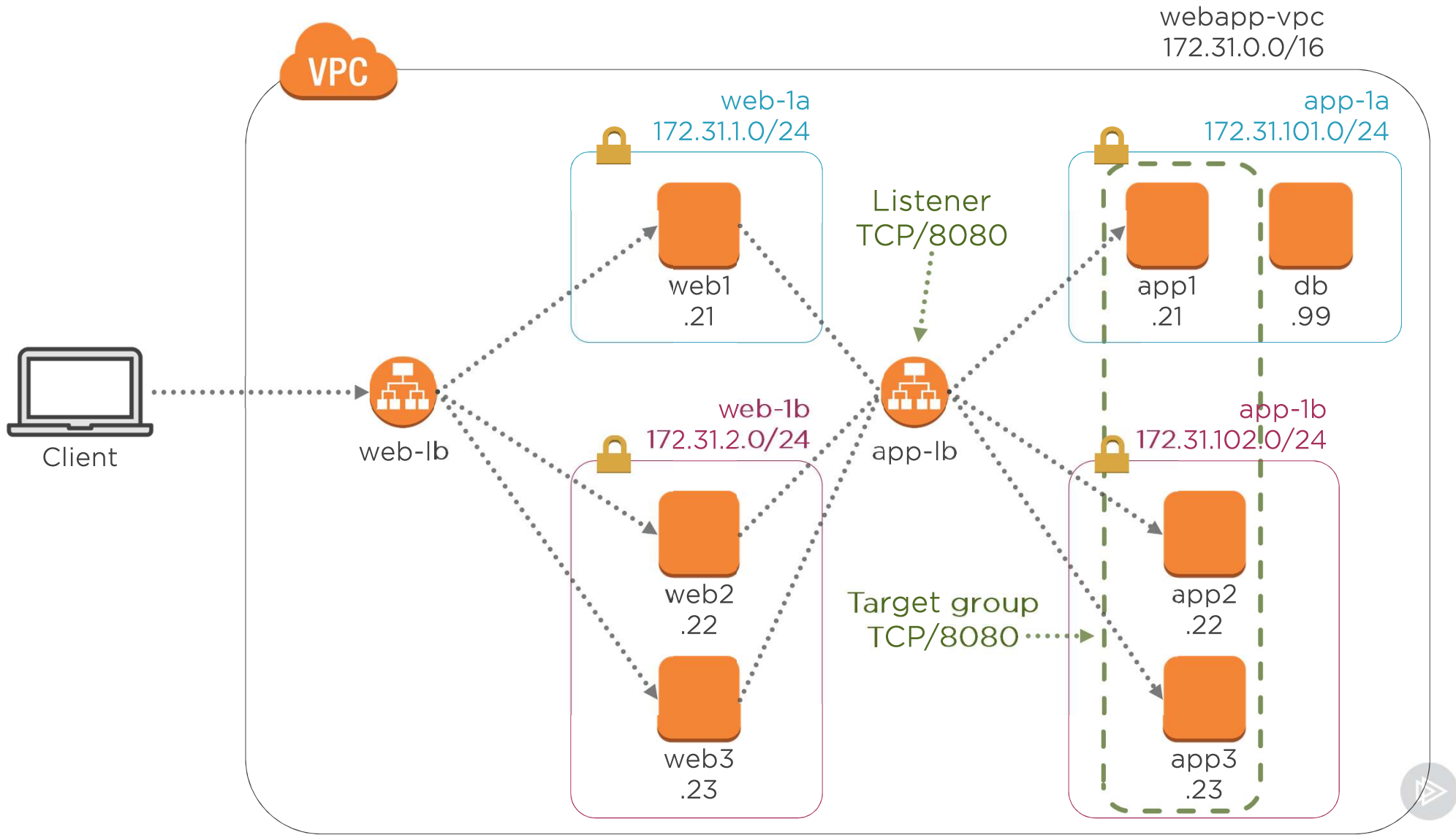
Ben Piper
AWS CERTIFIED SOLUTIONS ARCHITECT

https://benpiper.com

Creating an Internal Load Balancer

```
sudo docker run -d \

-p 8080:8080 -p 8443:8443 \

-h app1 \

benpiper/mtwa:app
```

Deploying the Application Tier Components

The application tier components execute inside of a Docker container

HTTP Request and Response

Instance: app1 (172.31.101.21)

Protocol: HTTP

Port: TCP/8080

Path: /appserverinfo.py

Request Method: GET

Request URL: http://172.31.101.21:8080/appserverinfo.py
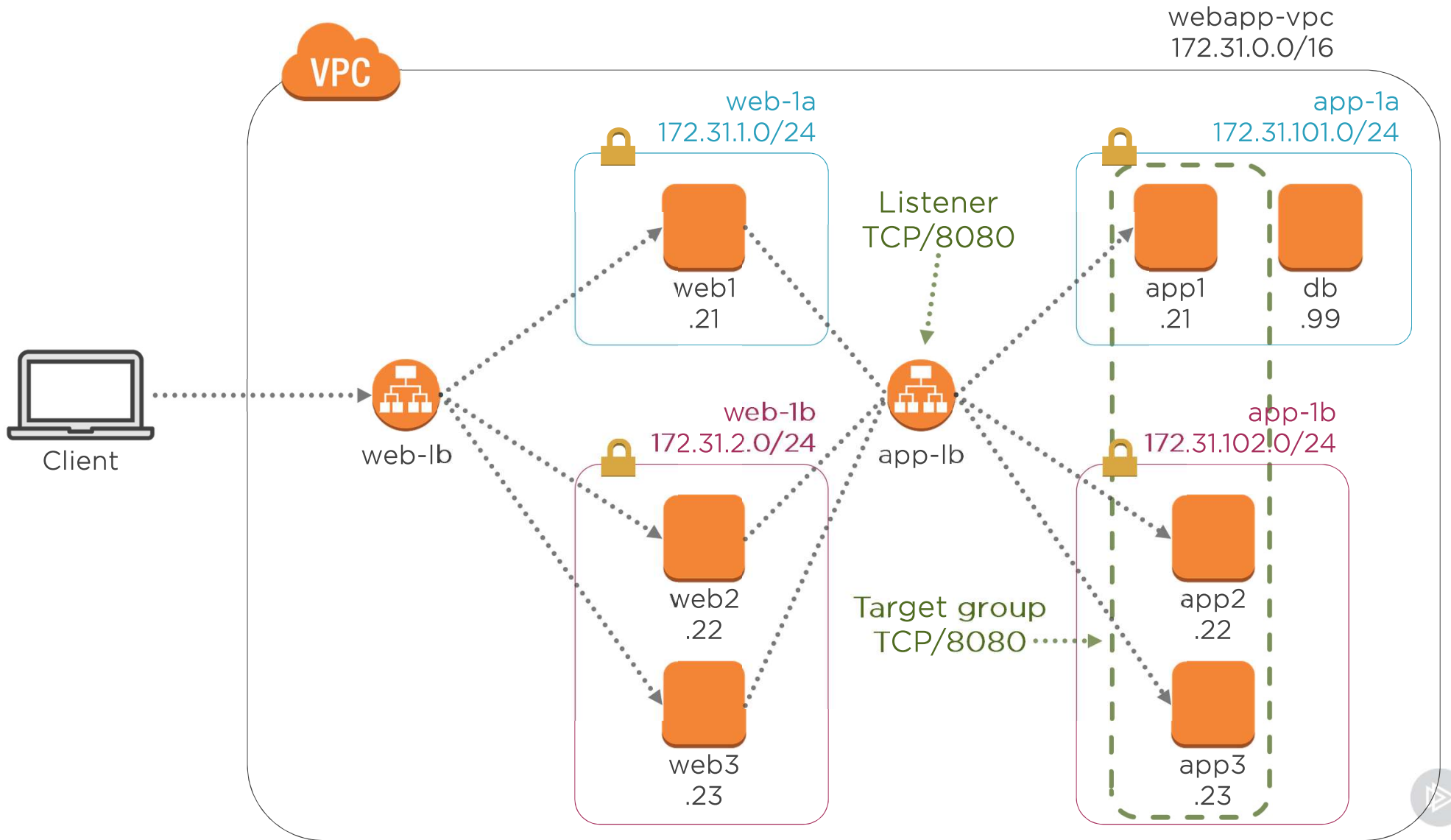
Status Code: 200 OK

Using an Internal Load Balancer

Using the Internal Load Balancer

```
sudo docker run -d \

-p 80:80 -p 443:443 \

-h web1 \

-e APPSERVER="http://internal-app-lb-1295698277.us-east-1.elb.amazonaws.com:8080" \

benpiper/mtwa:web
```

VPC

webapp-vpc
172.31.0.0/16

web-1a
172.31.1.0/24

web1
.21

Listener
TCP/8080

app-1a
172.31.101.0/24

app1
.21

db
.99

web-lb

Client

web-1b
172.31.2.0/24

web2
.22

web3
.23

app-lb

Target group
TCP/8080

app-1b
172.31.102.0/24

app2
.22

app3
.23

Provisioning the Database Tier

```
sudo docker run -d \

-p 3306:3306 \

-h db \

benpiper/mtwa-db
```

Deploying the Database

Summary

Internet-facing load balancer distributes traffic to the web tier

Internal load balancer distributes traffic to the application tier

Internal load balancer doesn't have publicly resolvable DNS or public IP

DNS name resolves to a private IP

Listener uses IPv4 only

Coming Up Next

Sticky Sessions and Idle Timeouts