

Load Balancing Internet-facing Web Applications

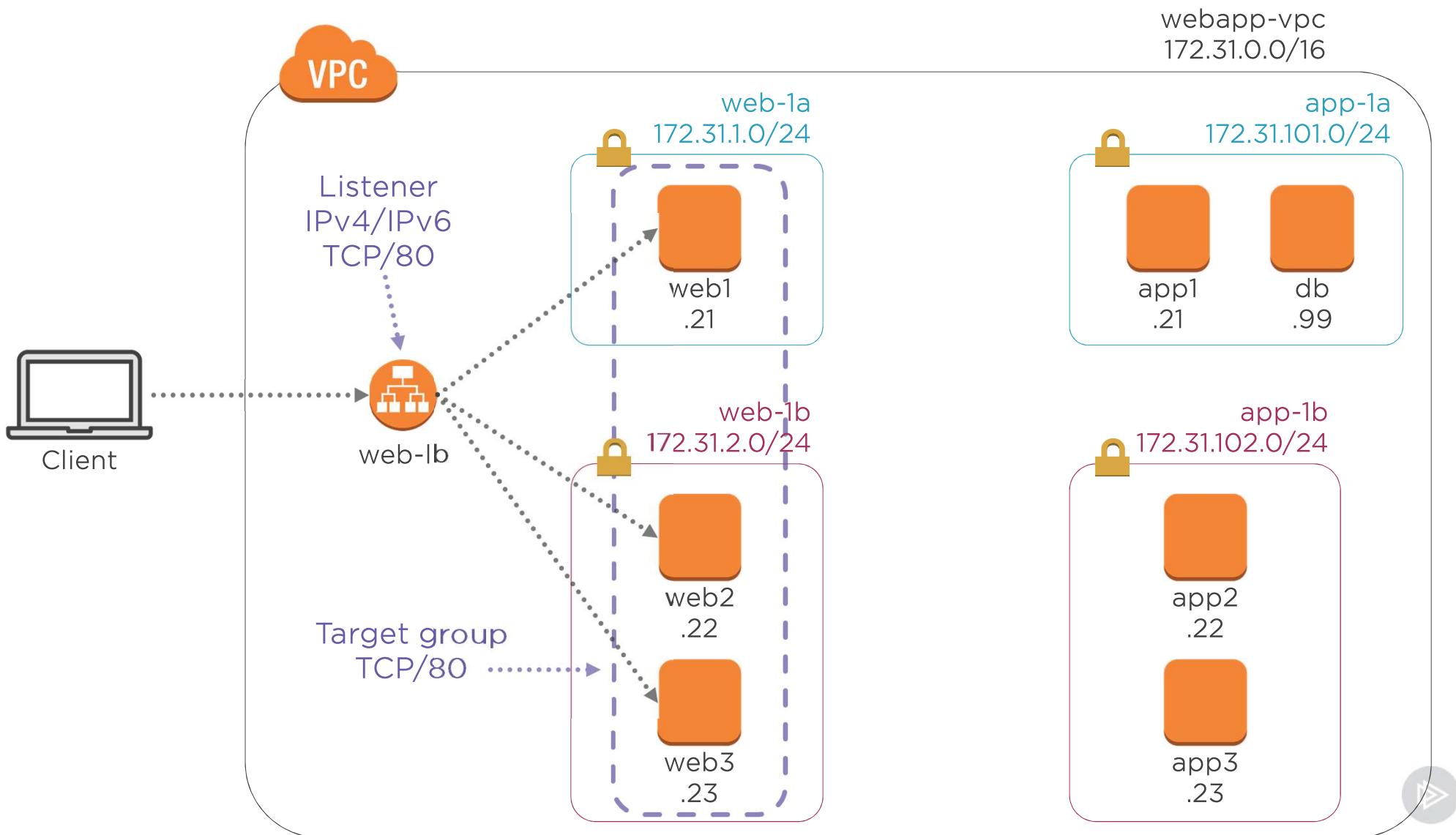


Ben Piper

AWS CERTIFIED SOLUTIONS ARCHITECT

<https://benpiper.com>

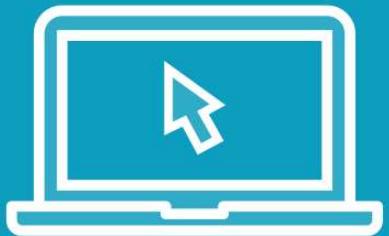




Provisioning the Web Tier



Demo



SSH into web1

Deploy the front end using Docker

Browse to web1's public IP address

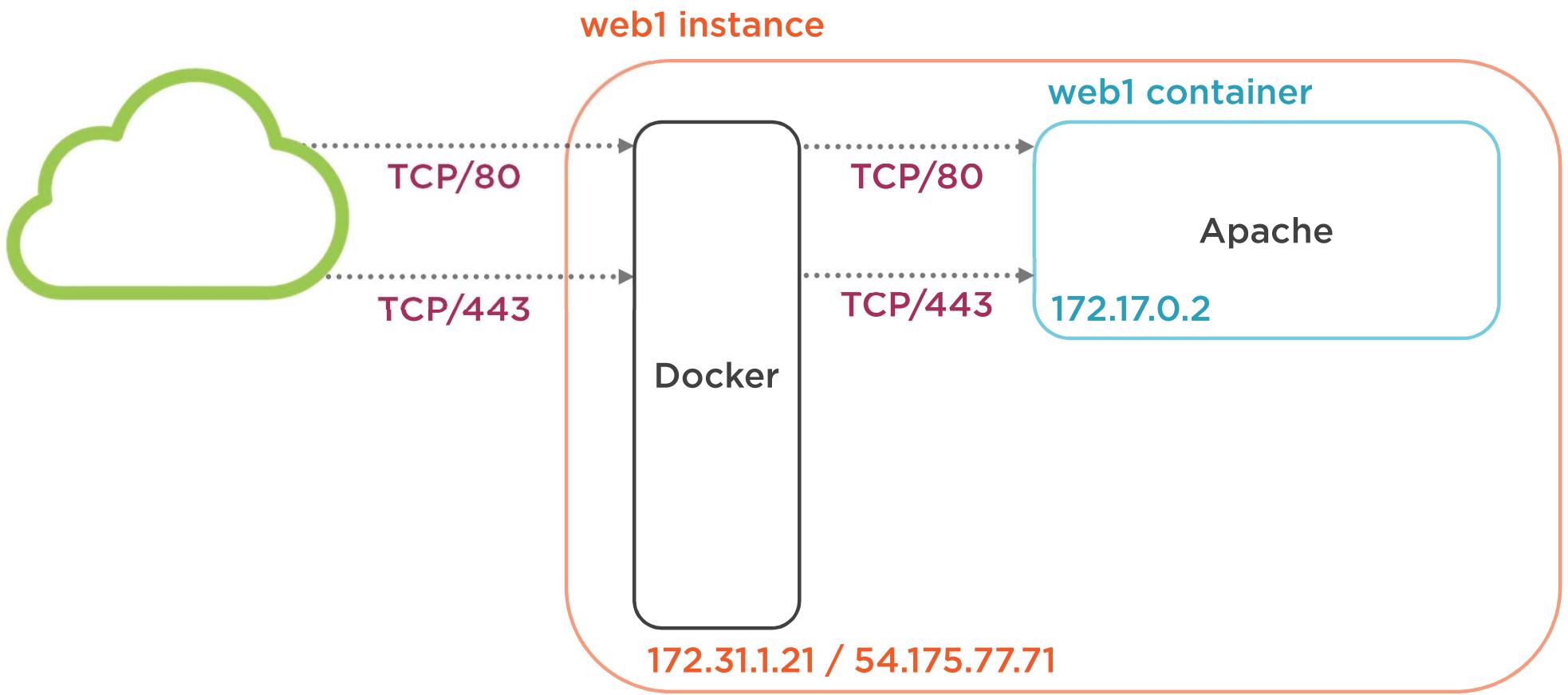


```
sudo docker run -d \
-p 80:80 -p 443:443 \
-h web1 \
benpiper/mtwa:web
```

Deploying the Web Front End

The web front end components execute inside of a Docker container





Creating an HTTP Target Group



Supported Protocols

HTTP

HTTPS

TCP



Port Range

TCP/1-65535

**AWS will change the port number to match
the protocol**

HTTP or TCP: 80

HTTPS: 443



Target Types

Instance

Routes traffic to the primary private IP address of the instance

IP

Routes traffic to a specified IP address

RFC 1918: 10.0.0.0/8, 172.16.0.0/12, and 192.168.0.0/16

RFC 6598: 100.64.0.0/10



Health Checks



Determines whether to send traffic to a given instance

Each instance must pass its health check before receiving traffic

Sends HTTP GET request and looks for a success code

HTTP Request and Response

Instance: web1 (172.31.1.21)

Protocol: HTTP

Port: TCP/80

Path: /

Request URL: http://172.31.1.21/

Request Method: GET

Status Code: 200 OK



Creating the Application Load Balancer



Registering Additional Targets



Testing IPv6 Connectivity





Linux, Mac, and Windows prefer IPv6 by default if it's available

Use the `ipv4` address type to prevent all IPv6 connections



Enable IPv6 on your client!



Summary



Decide on a load balancer type before configuring a target group

If you select the HTTP or HTTPS protocol, you must use an ALB

If you select TCP, you must use the NLB

Health check settings include protocol, port, and path

200 OK response code indicates a healthy instance



Coming Up Next



Load Balancing Internal Web Services

