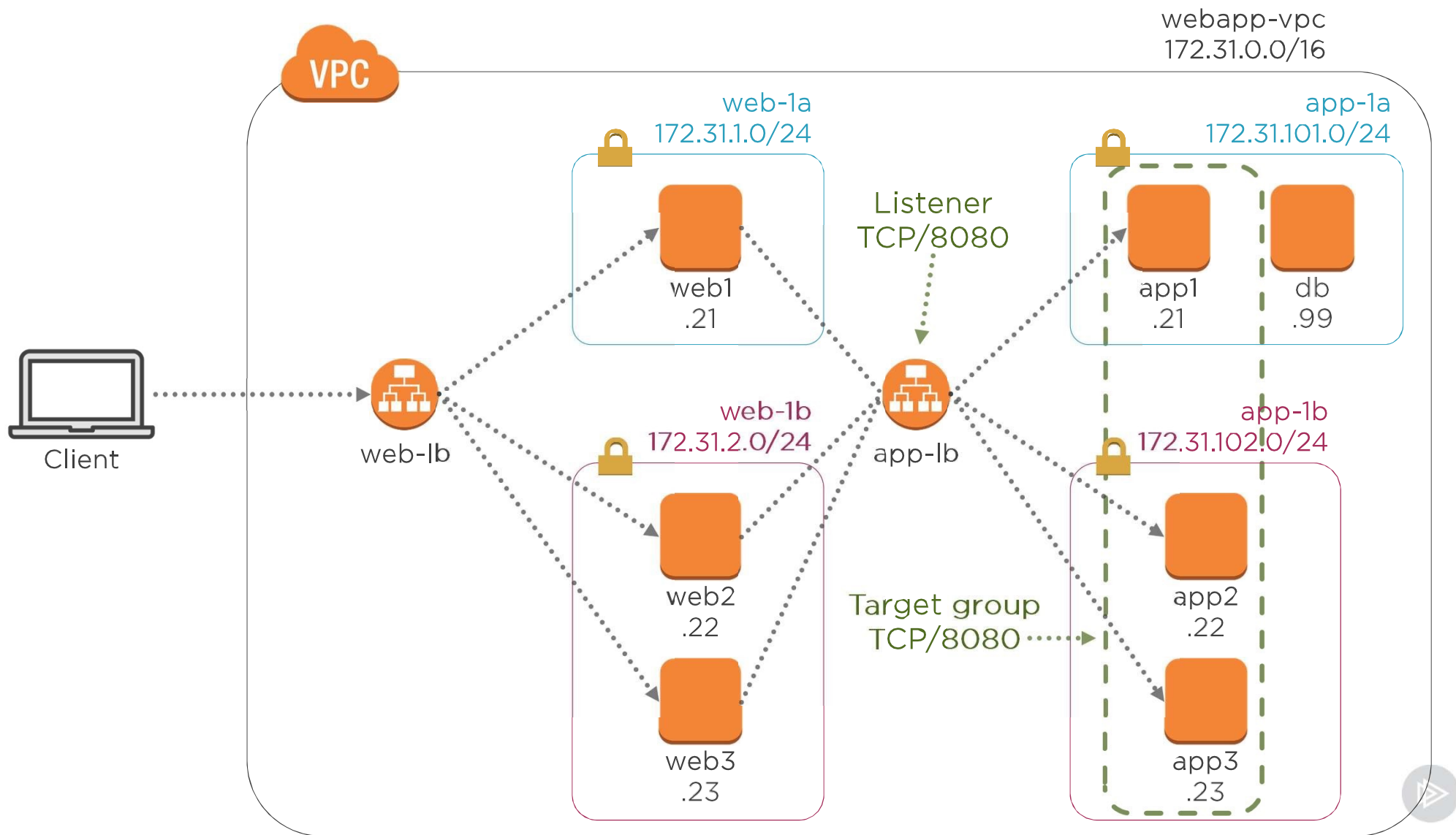


# Load Balancing Internal Web Services

---





# Creating an Internal Load Balancer

---



```
sudo docker run -d \  
-p 8080:8080 -p 8443:8443 \  
-h app1 \  
kienbt/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

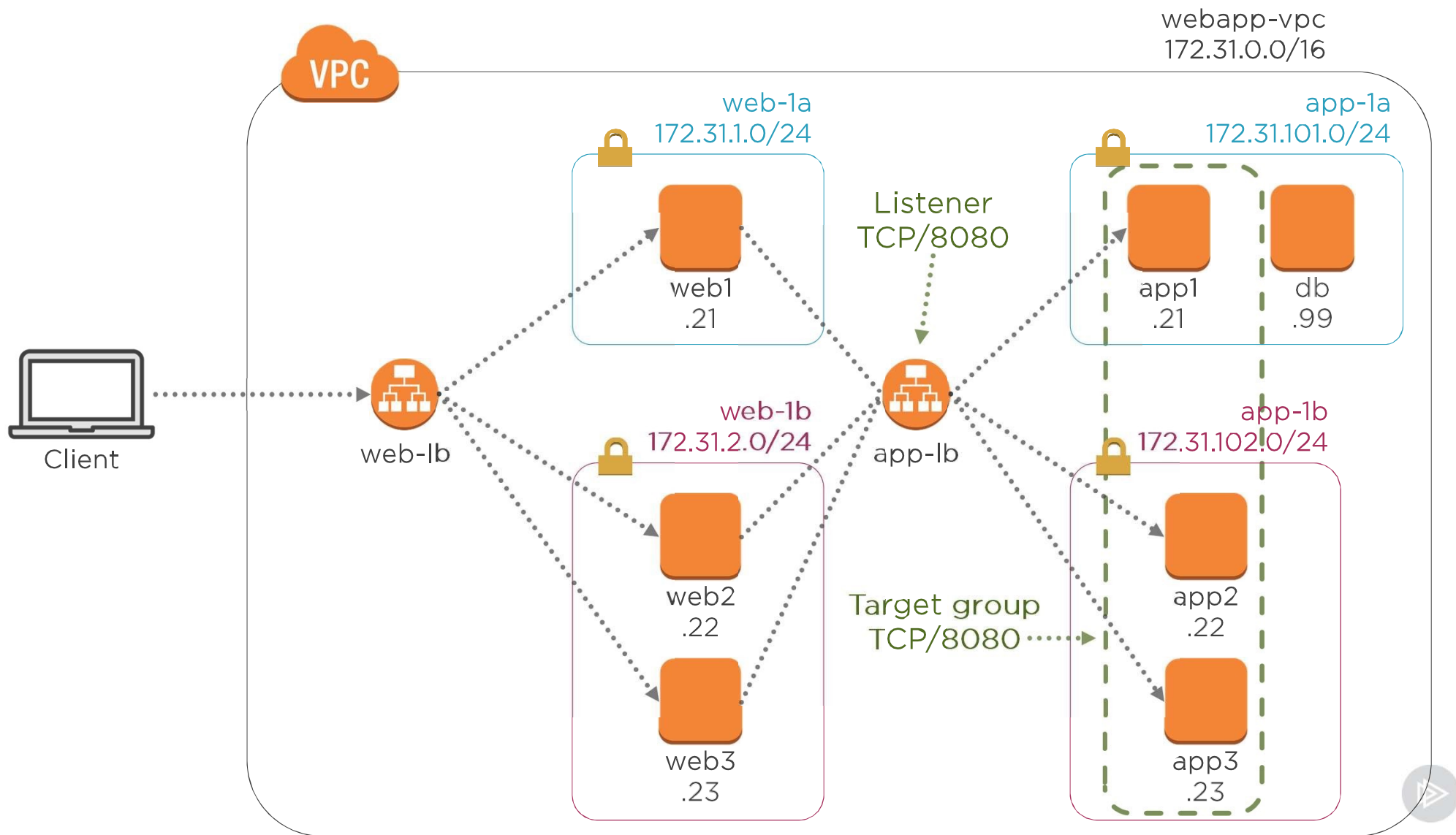
---



# 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" \  
kienbt/mtwa:web
```







# Provisioning the Database Tier

---



```
sudo docker run -d \  
-p 3306:3306 \  
-h db \  
kienbt/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**

