

APPLICATION LOAD BALANCER

KRANTHI APURI, MASTER SOFTWARE ENGINEER, CAPITAL ONE



KRANTHI APURI



@KRANTHIAPURI

MAY 24, 2017

WHAT IS AN ELASTIC LOAD BALANCER?

ELASTIC LOAD BALANCING AUTOMATICALLY DISTRIBUTES INCOMING APPLICATION TRAFFIC ACROSS MULTIPLE AMAZON EC2 INSTANCES:

CLASSIC LOAD BALANCER:

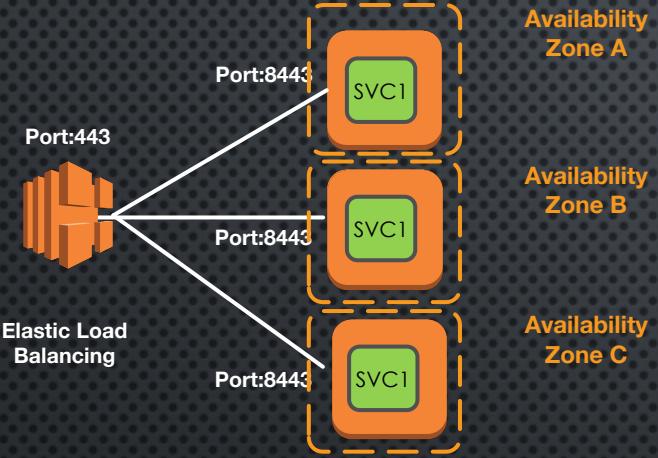
- ROUTES BASED ON APPLICATION OR NETWORK LEVEL INFORMATION
- SIMPLE LOAD BALANCING ACROSS EC2 INSTANCES.

APPLICATION LOAD BALANCER:

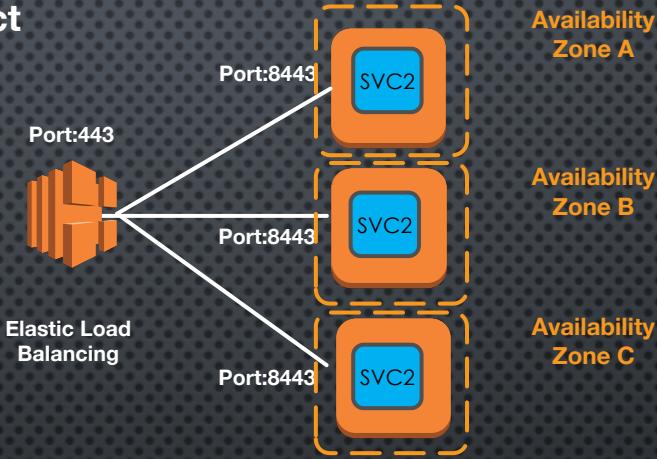
- ADVANCED ROUTING CAPABILITIES.
- ROUTES TO MULTIPLE SERVICES OR MULTIPLE PORTS WITHIN EC2 INSTANCES
- SUITABLE FOR MICRO SERVICES AND CONTAINER BASED ARCHITECTURES.

CLASSIC – ELASTIC LOAD BALANCER (NON-DOCKER)

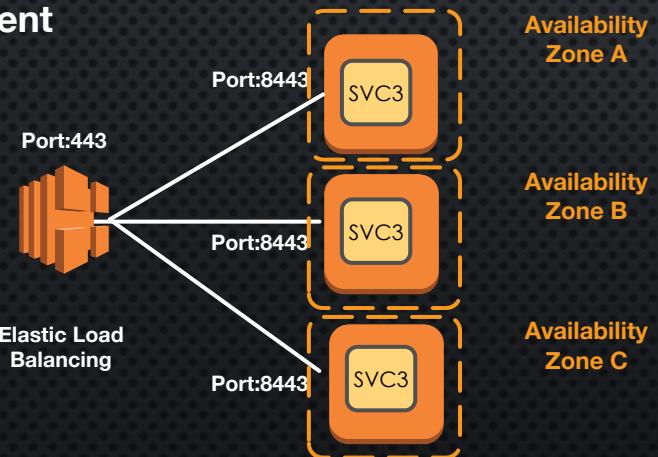
/login



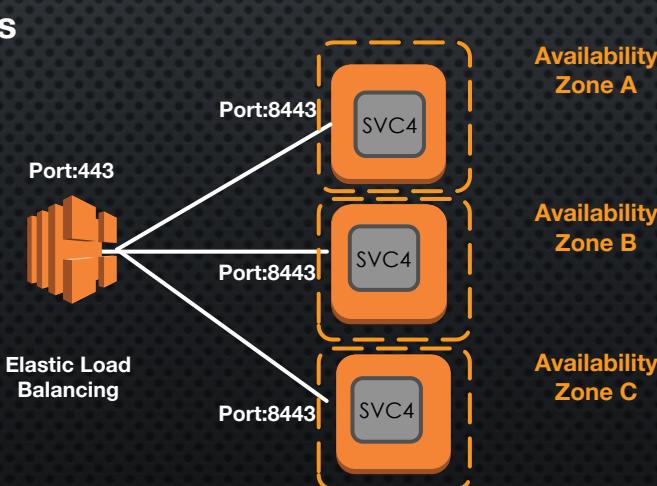
/select



/payment



/orders



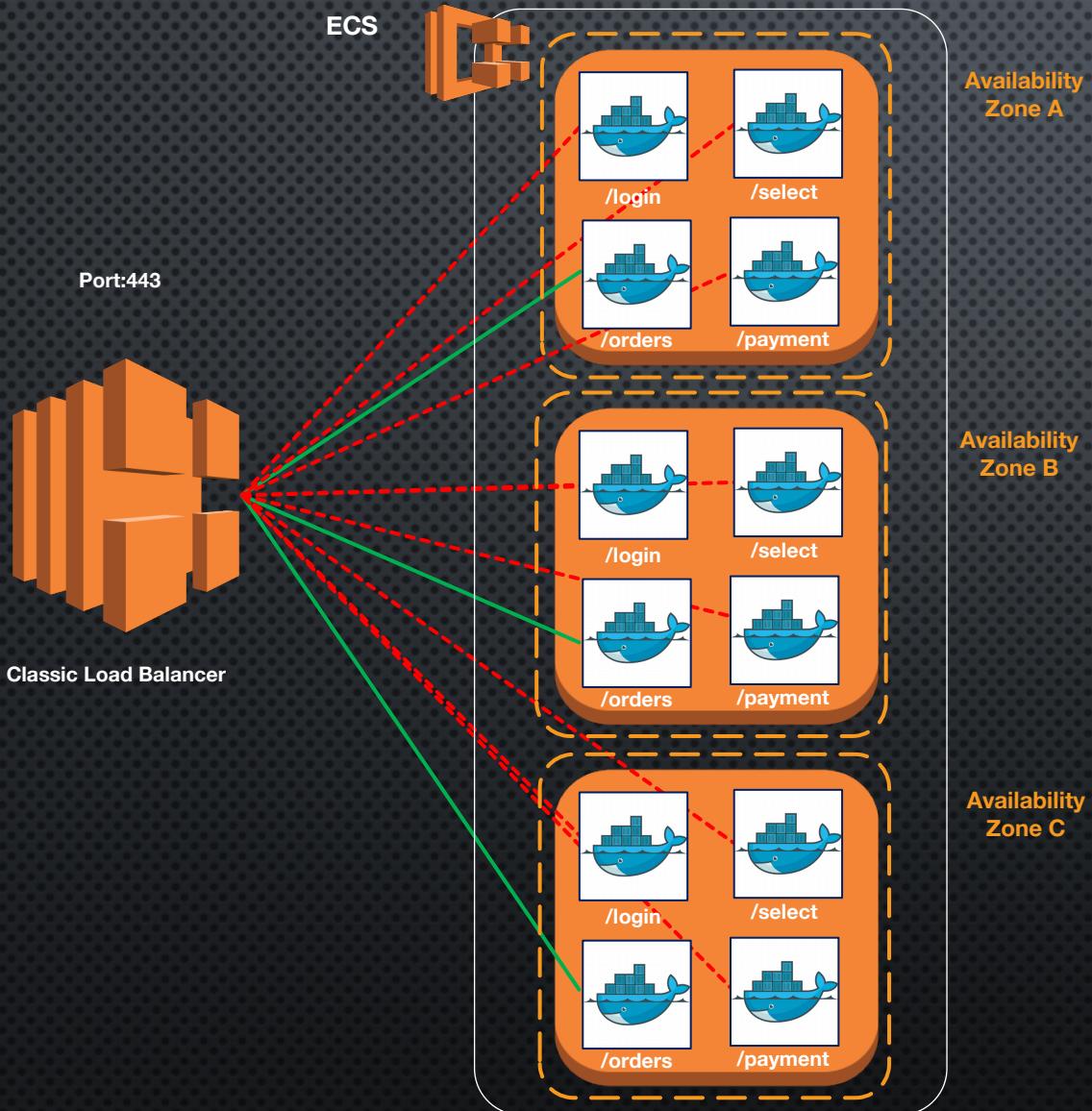
Availability Zone A
Availability Zone B
Availability Zone C

Availability Zone A
Availability Zone B
Availability Zone C

LIMITATIONS:

- LOW CPU UTILIZATION
- HIGH COSTS
- HIGH MAINTENANCE

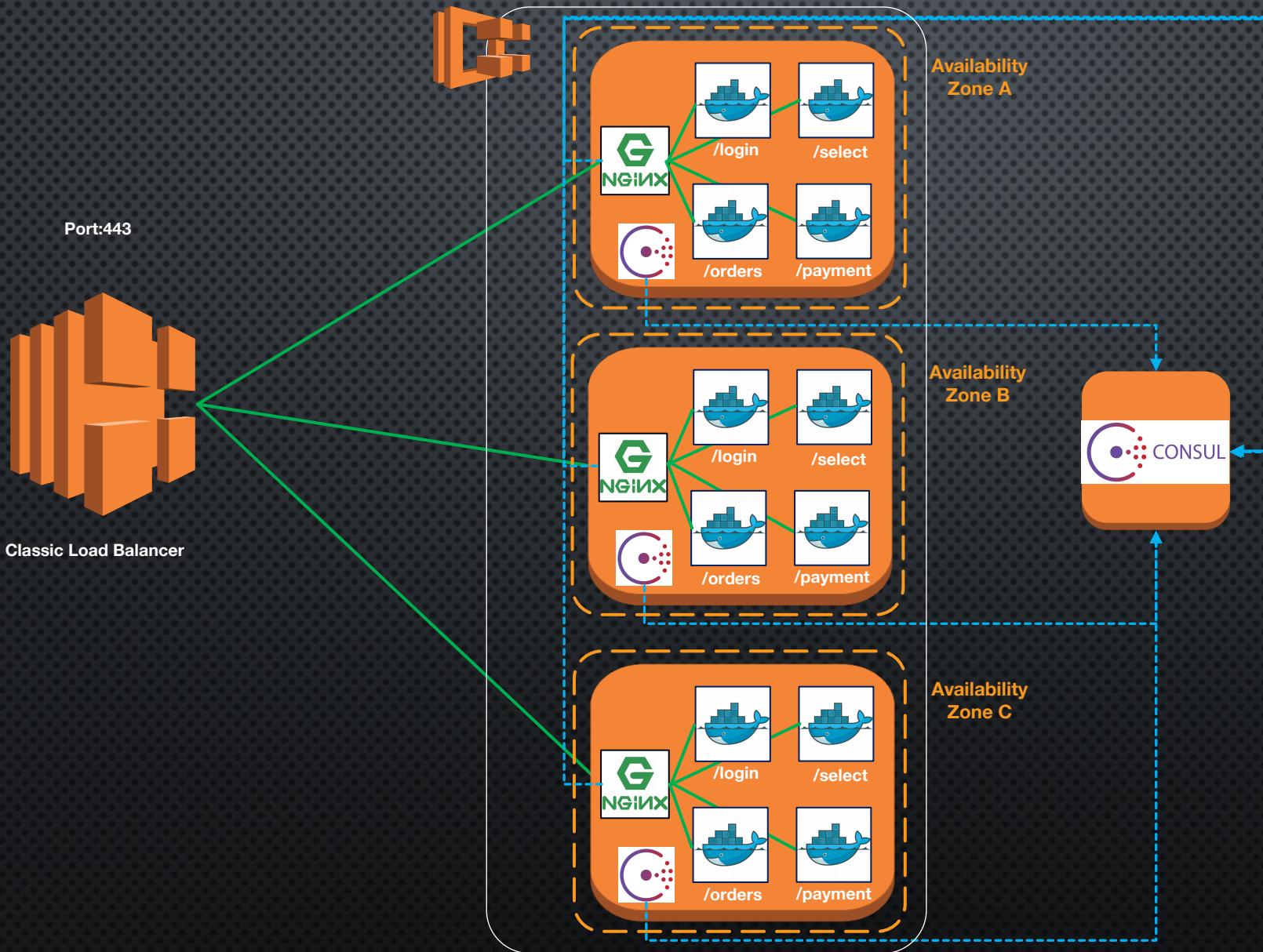
CLASSIC – ELASTIC LOAD BALANCER (DOCKER)



LIMITATIONS OF CLASSIC ELB

- 1:1 MAPPING OF LISTENER PORT TO THE INSTANCE PORT OF THE APPLICATION
- LACK OF SERVICE LEVEL HEALTH CHECKS
- MANAGE THE PORTS MANUALLY

SERVICE DISCOVERY USING NGINX & CONSUL



CLASSIC – ELASTIC LOAD BALANCER (DOCKER)

LIMITATIONS OF USING NGINX & CONSUL:

- MAINTENANCE OF ADDITIONAL EC2 CLUSTER
- MULTIPLE POINTS OF FAILURE
- DIFFICULTY IN DEBUGGING AND TROUBLESHOOTING
- REDUCED PERFORMANCE

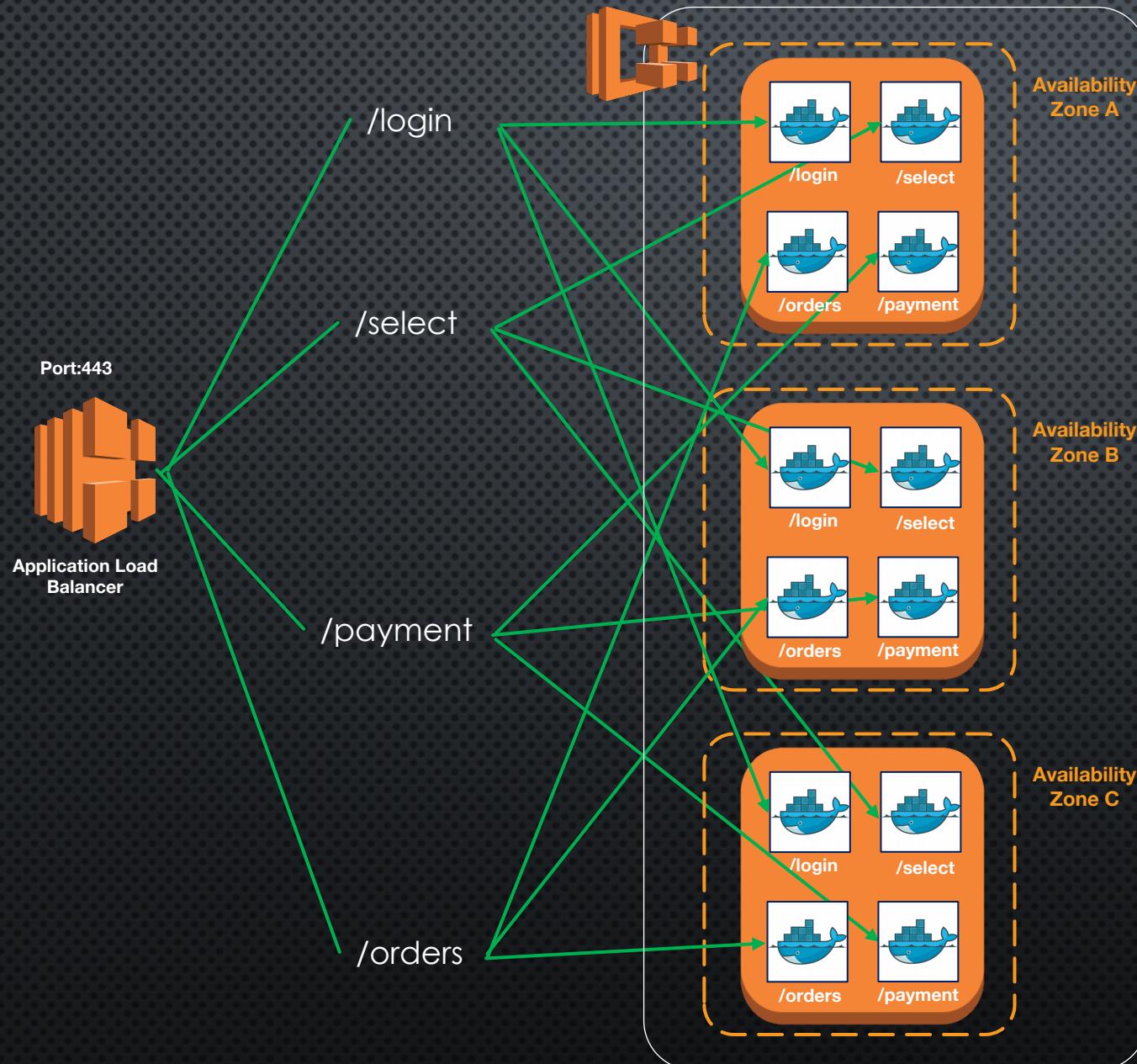
APPLICATION LOAD BALANCER

APPLICATION LOAD BALANCER OPERATES AT THE **APPLICATION LAYER** AND ALLOWS YOU TO **DEFINE ROUTING RULES** BASED ON **CONTENT** ACROSS MULTIPLE SERVICES OR CONTAINERS RUNNING ON ONE OR MORE EC2 INSTANCES.

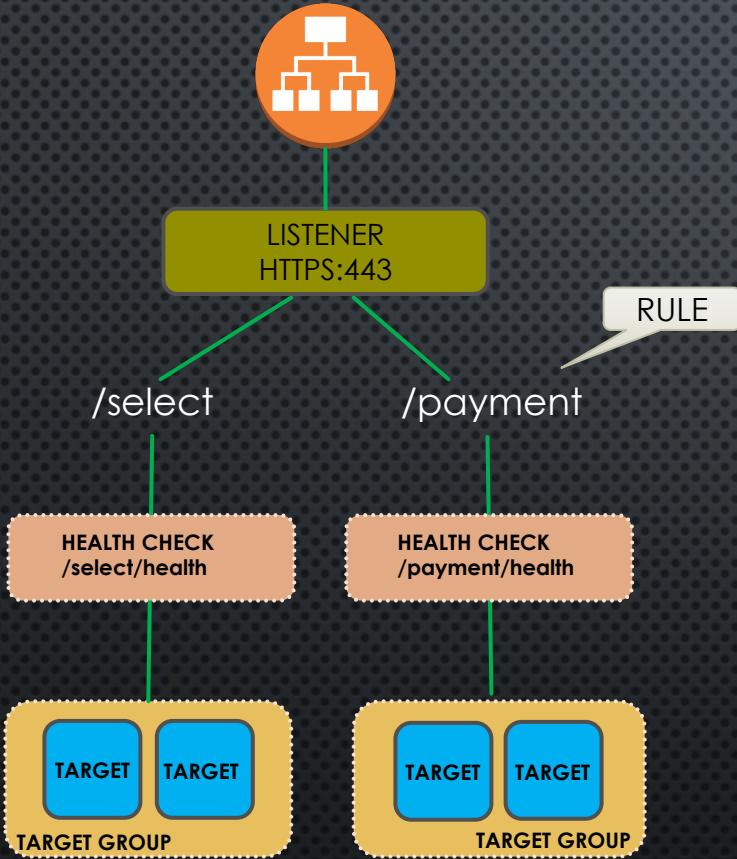
NEW KEY FEATURES:

- CONTENT BASED ROUTING
- DYNAMIC PORT MAPPING
- CONTAINERIZED APPLICATIONS SUPPORT
- IMPROVED LOAD BALANCER PERFORMANCE
- WEB SOCKET & HTTP/2 SUPPORT
- INTEGRATION WITH ECS

APPLICATION LOAD BALANCER (DOCKER)



APPLICATION LOAD BALANCER - CONCEPTS



- **LISTENER** – DEFINES THE PROTOCOL AND PORT FOR THE INCOMING CONNECTIONS.
- **RULE** – A RULE IS MADE UP OF CONDITIONS AND ACTIONS FOR ROUTING REQUESTS. CURRENTLY, ALB ONLY SUPPORTS CONDITION OF PATH AND ACTION OF FORWARD
- **TARGET GROUP** – A COLLECTION OF TARGETS SUCH AS EC2 INSTANCE. HAVE INSTANCE PORT, PROTOCOL AND CONFIGURATIONS FOR THE INSTANCE SIDE OF THE CONNECTION
- **TARGET** – ANY RESOURCE OR ENDPOINT TO WHICH LOAD BALANCER CAN SEND TRAFFIC. EC2 INSTANCE, MICRO SERVICE OR A CONTAINER.

DEMO

Code available at: <https://github.com/kranthi-apuri/DevOps-RVA-0524>

WHEN SHOULD I USE APPLICATION LOAD BALANCER?

	CLASSIC	APPLICATION
PROTOCOL	TCP, SSL, HTTP, HTTPS	HTTP, HTTPS
HEALTH CHECKS	✓	IMPROVED
CLOUDWATCH METRICS	✓	IMPROVED
PATH-BASED ROUTING		✓
CONTAINER SUPPORT		✓
WEBSOCKETS & HTTP/2		✓
PERFORMANCE	✓	IMPROVED
DELETE PROTECTION		✓

REFERENCES

- [HTTPS://AWS.AMAZON.COM/ELASTICLOADBALANCING/](https://aws.amazon.com/elasticloadbalancing/)
- [HTTPS://AWS.AMAZON.COM/BLOGS/DEVOPS/INTRODUCING-APPLICATION-LOAD-BALANCER-UNLOCKING-AND-OPTIMIZING-ARCHITECTURES/](https://aws.amazon.com/blogs/devops/introducing-application-load-balancer-unlocking-and-optimizing-architectures/)
- [HTTPS://WWW.YOUTUBE.COM/WATCH?V=QY7zNADTYGQ](https://www.youtube.com/watch?v=QY7zNADTYGQ)