NET402

AWS re:INVENT

Elastic Load Balancing Deep Dive and Best Practices

David Pessis

November 30, 2017

AWS re:Invent

aws

Elastic Load Balancing automatically distributes incoming application traffic across multiple targets, such as Amazon **EC2 instances**, **containers**, and **IP addresses**

Layer 4 (network)

Supports TCP and SSL

Incoming client connection bound to server connection

No header modification

Proxy Protocol prepends source and destination IP and ports to request

Layer 7 (application)

Supports HTTP and HTTPS

Connection terminated at the load balancer and pooled to the server

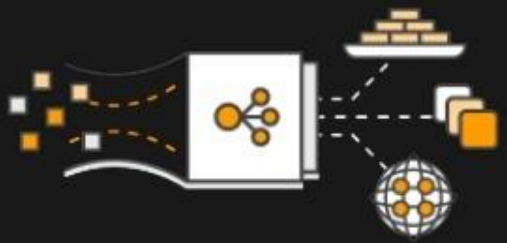
Headers may be modified

X-Forwarded-For header contains client IP address

The Elastic Load Balancing family

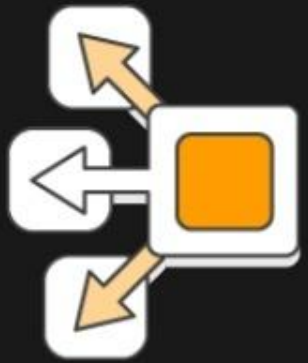
Application Load Balancer

HTTP & HTTPS (VPC)



Network Load Balancer

TCP Workloads (VPC)



Classic Load Balancer

Previous Generation
for HTTP, HTTPS, TCP
(Classic Network)

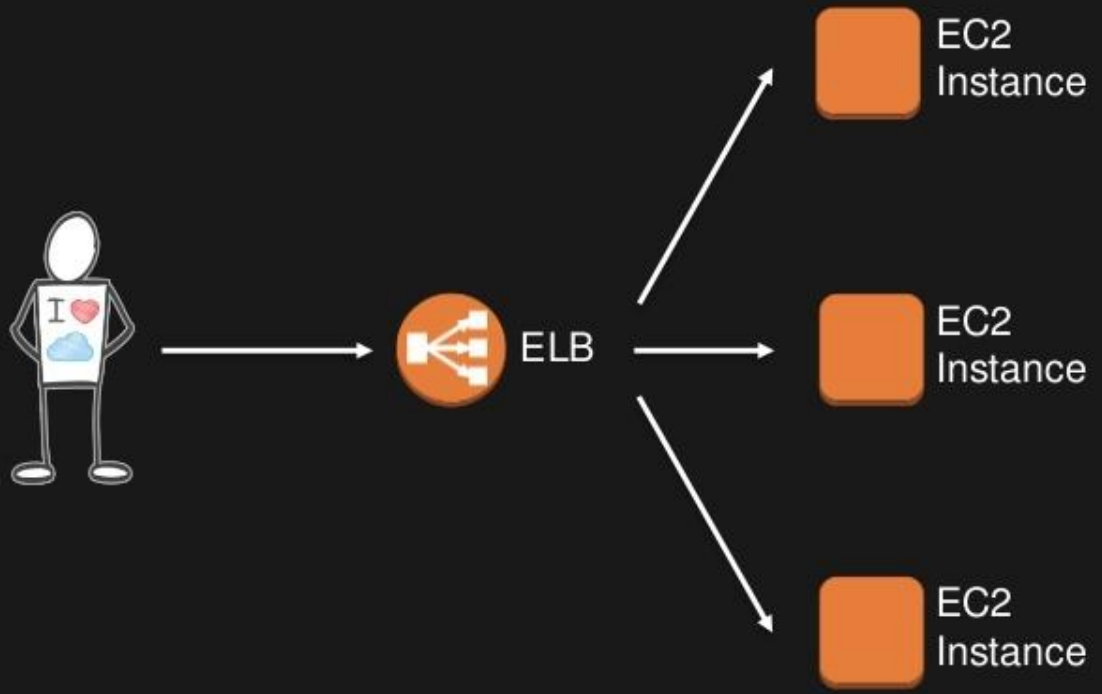
Elastic

Secure

Integrated

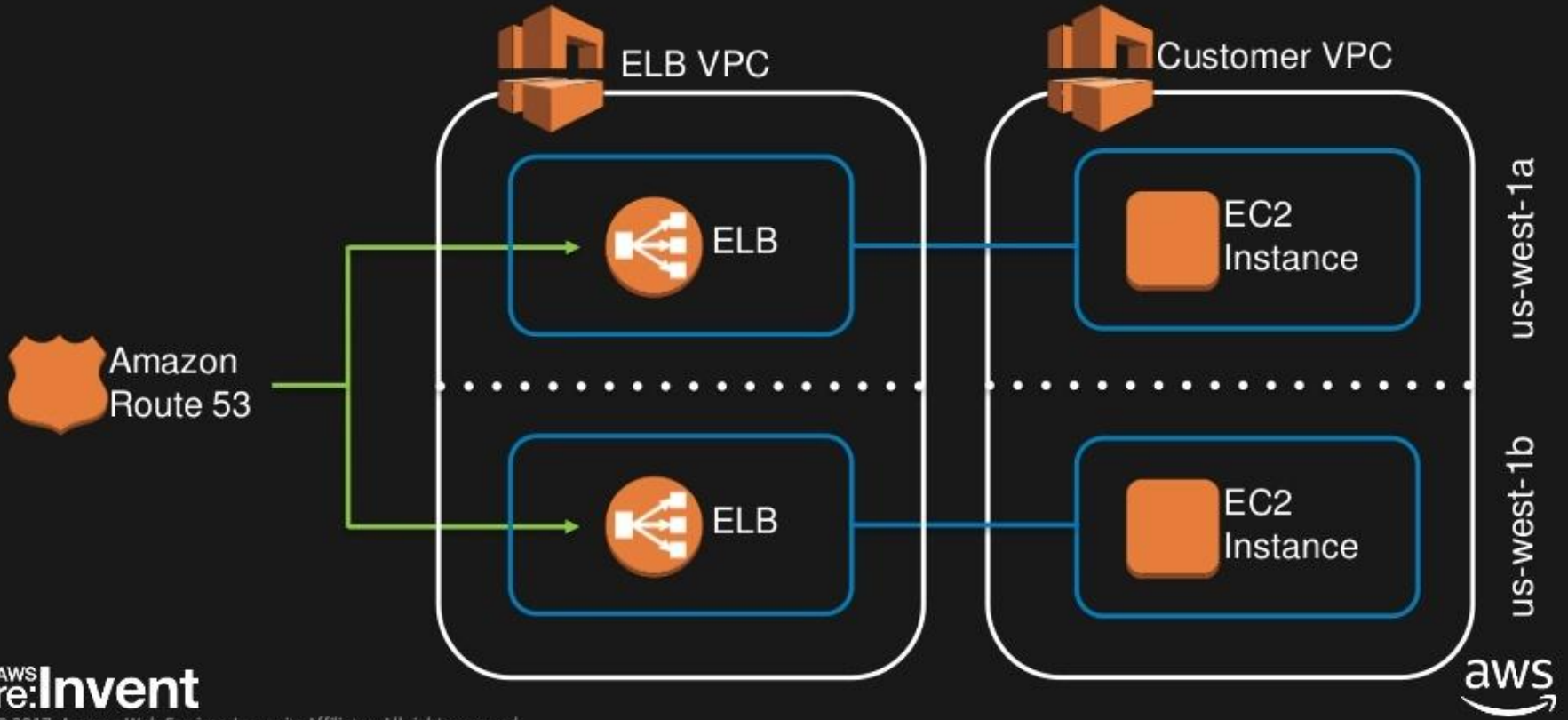
Cost Effective

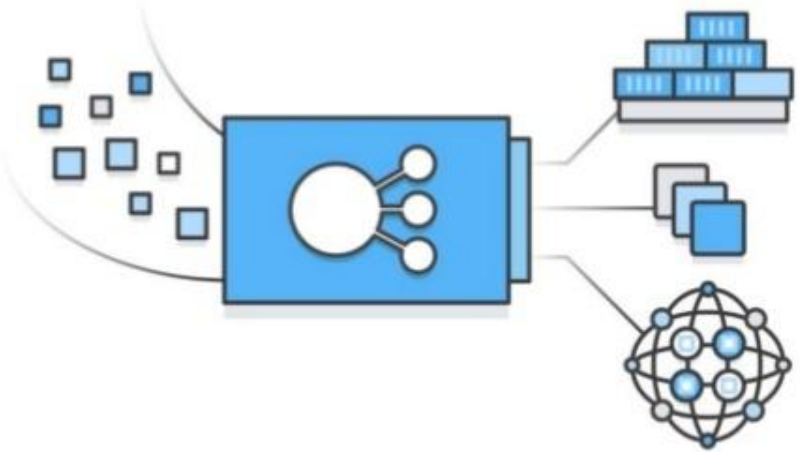
EC2
Instance



Load balancer used to route incoming requests to multiple EC2 instances, containers, or IP addresses in your VPC

Architecture





Application Load Balancer

Advanced request routing with support for microservices and container-based applications

Application Load Balancer

New, feature-rich, Layer 7 load-balanced platform

Content-based routing allows requests to be routed to different applications behind a single load balancer

Support for microservices and container-based applications, including deep integration with Elastic Container Service

Application Load Balancer

Support for WebSockets and HTTP/2

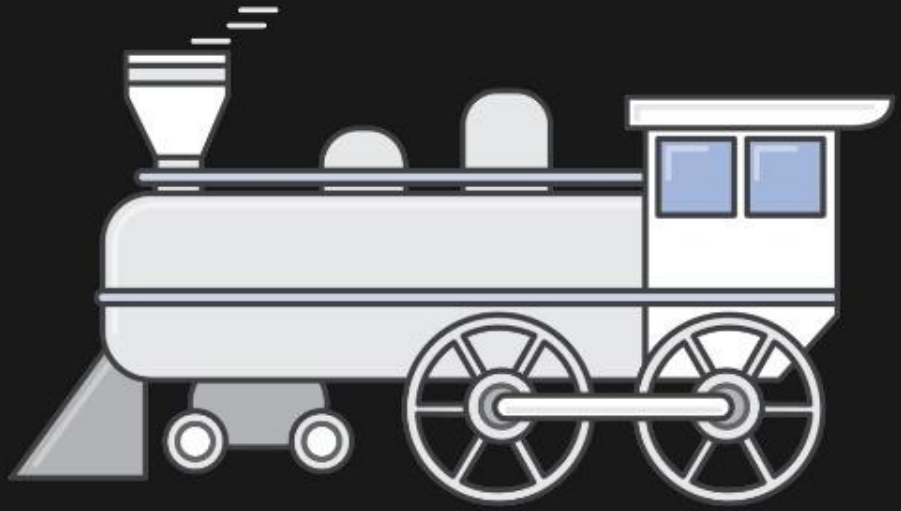
Path and host-based routing

Improved health checks and additional Amazon CloudWatch metrics

Improved performance for real-time and streaming applications

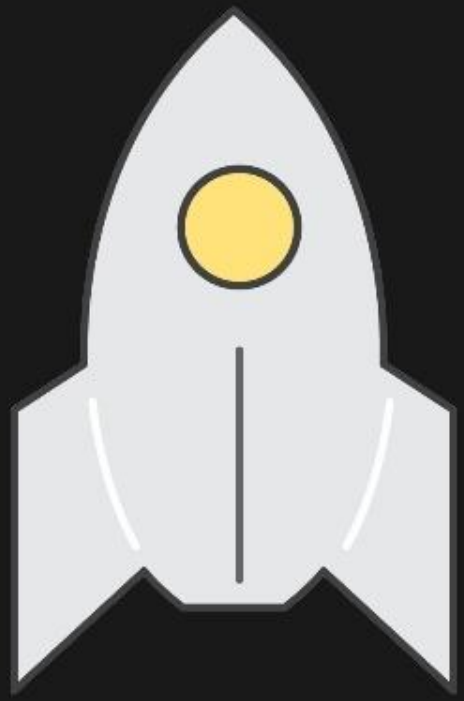
Improved Elastic Load Balancing API

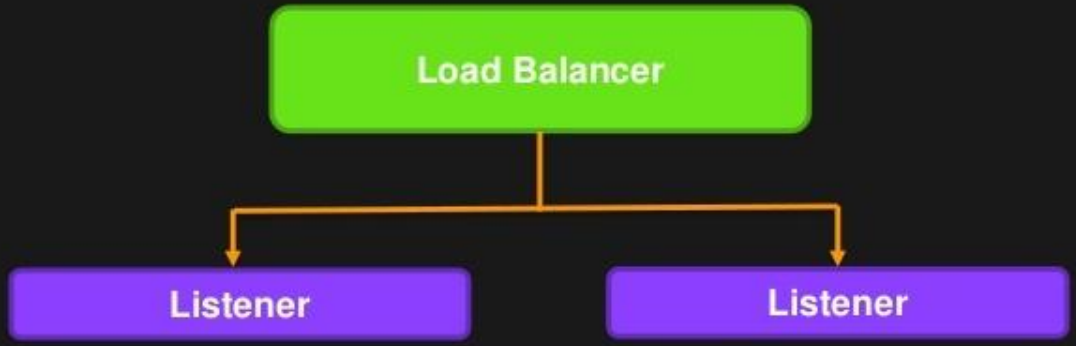
Load balancer API deletion protection



Feature launches in the last year…

- Host-based routing
- Server Name Identification (SNI)
- CloudWatch percentiles support
- Request tracing
- Native IPv6
- AWS WAF support
- New predefined security policies
- IP as a target





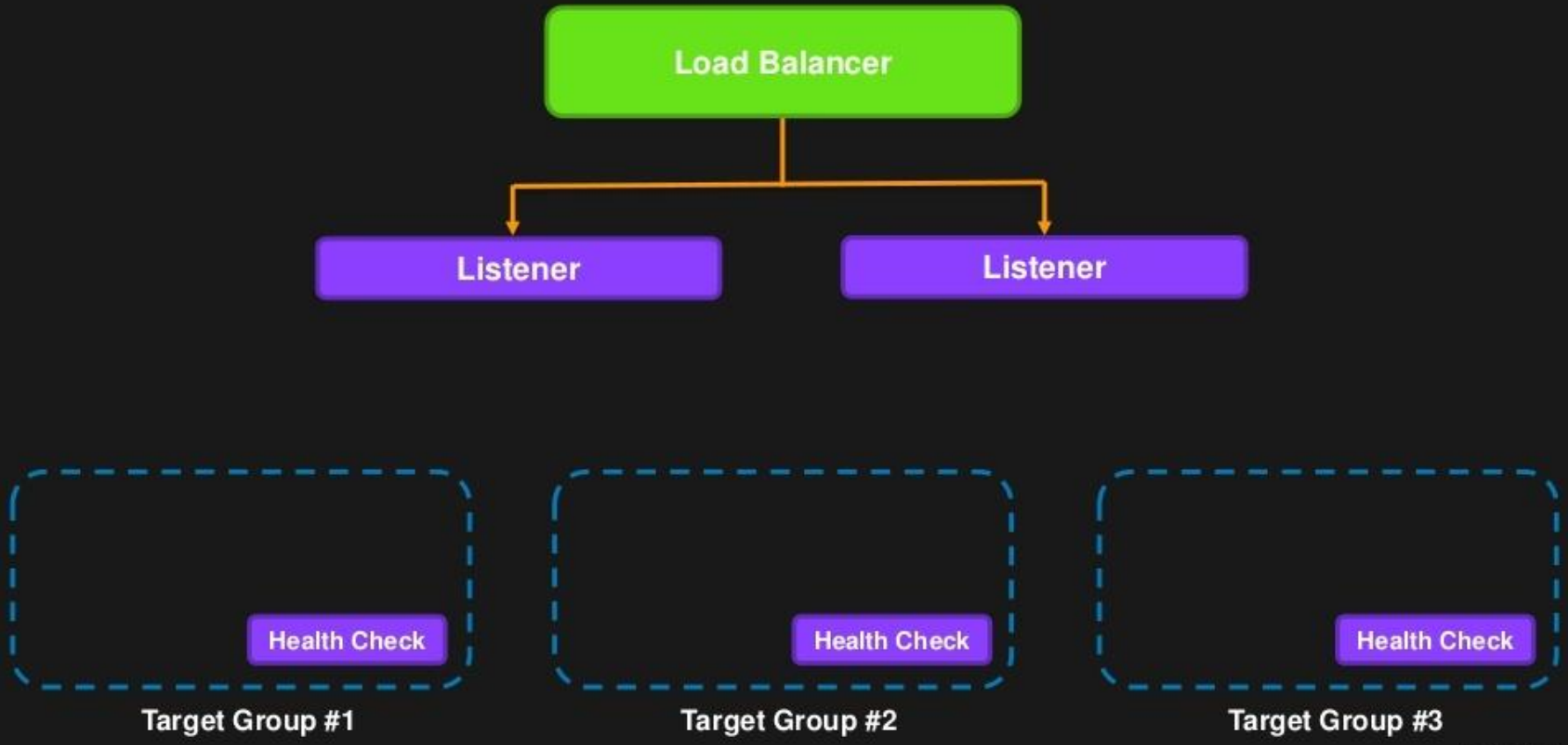
Listeners

Define the port and protocol which the load balancer must listen on

Each Application Load Balancer needs at least one listener to accept traffic

Each Application Load Balancer can have up to 10 listeners

Routing rules are defined on listeners



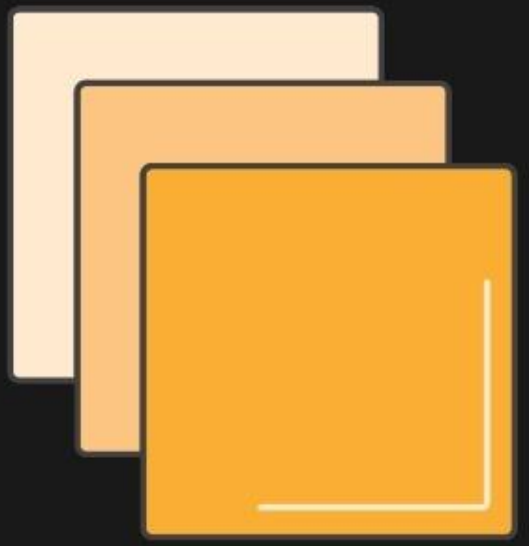
Target groups

Logical grouping of targets behind the load balancer

Target groups can exist independently from the load balancer

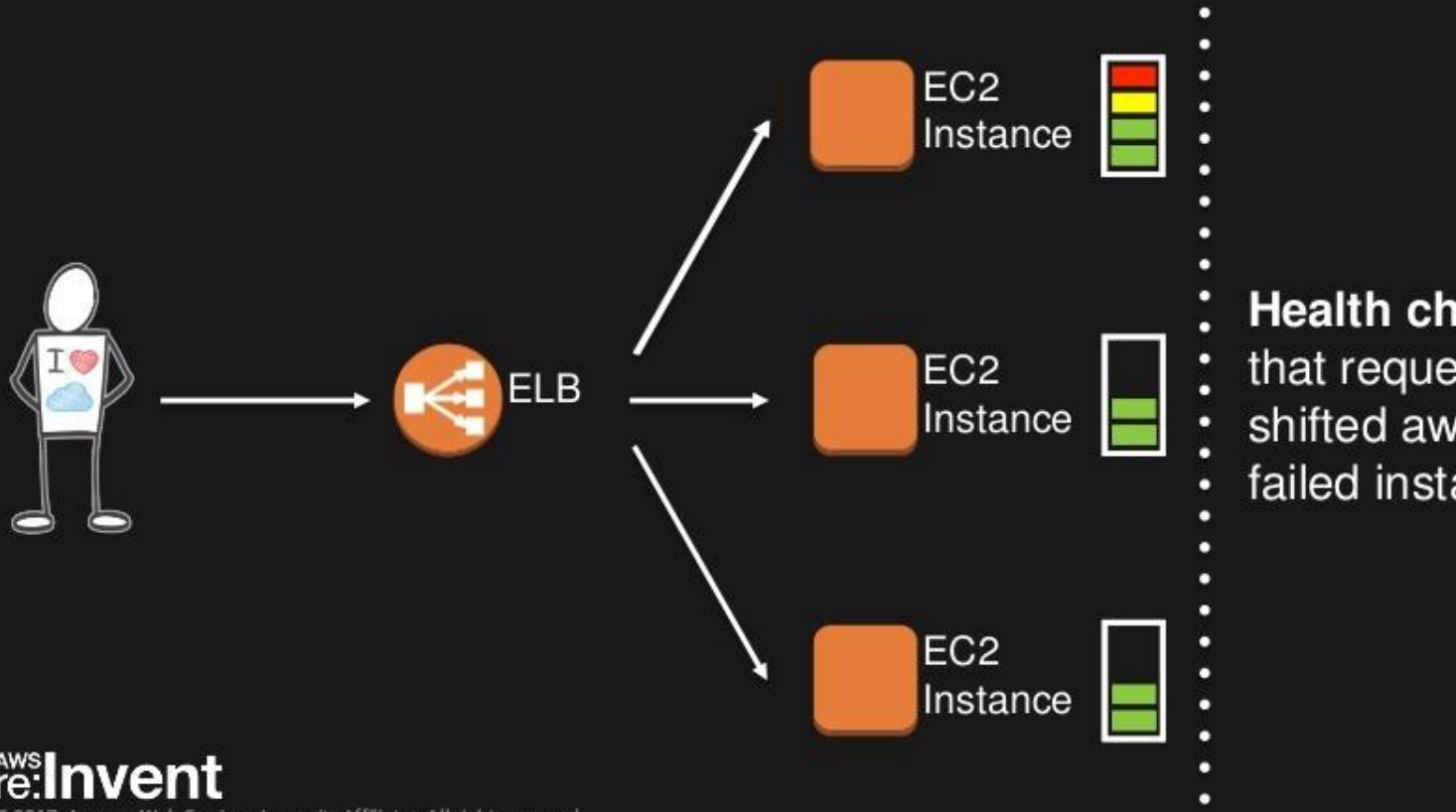
Regional construct that can be associated with an Auto Scaling group

Target groups can contain up to 1,000 targets



Health checks allow for traffic to be shifted away from **failed instances**

Health checks



Health checks ensure that request traffic is shifted away from a failed instance

Health checks

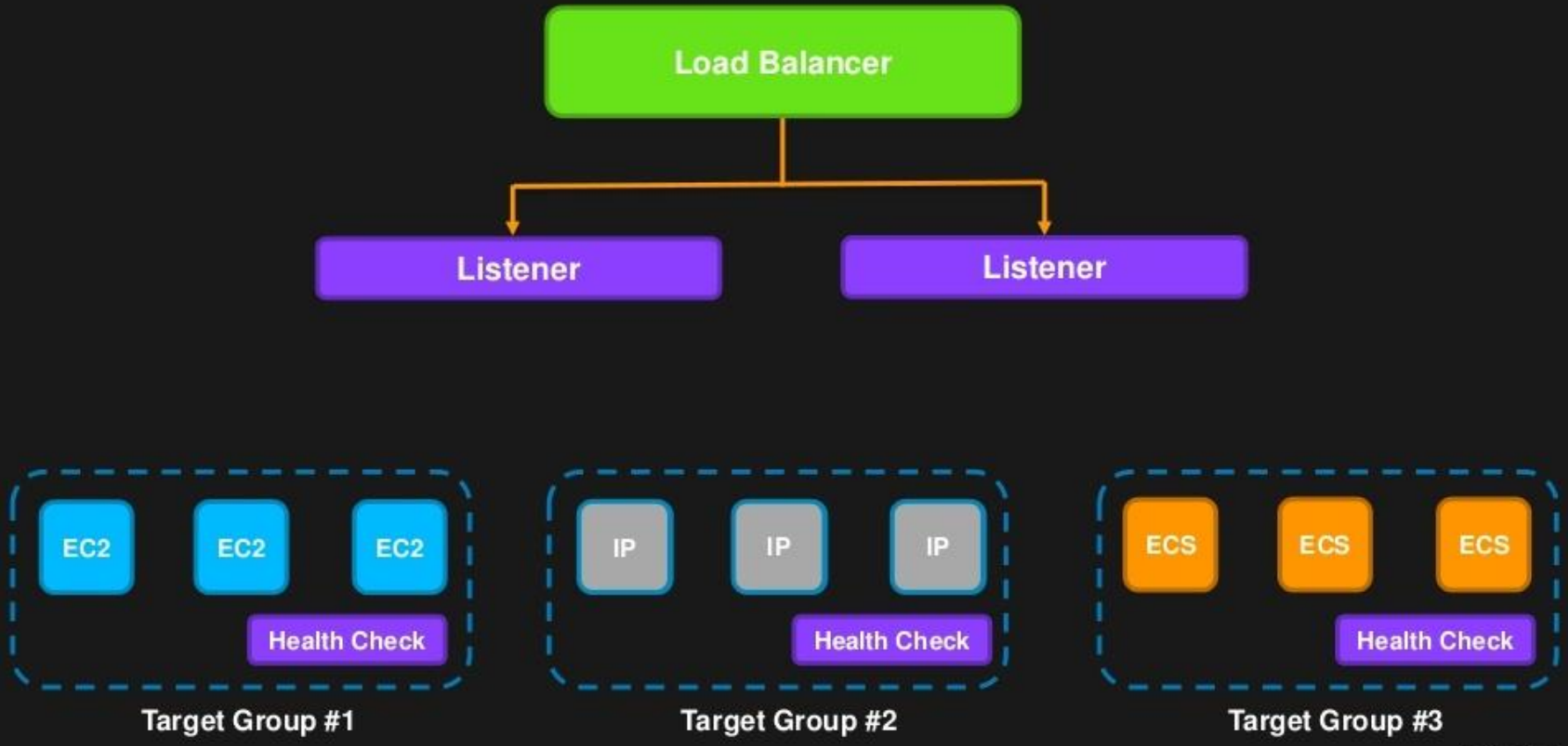
Support for HTTP and HTTPS health checks

Customize the frequency and failure thresholds

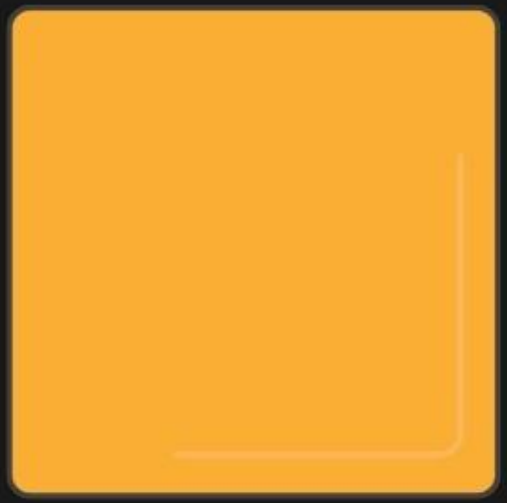
Consider the depth and accuracy of your health checks

Customize list of successful response codes, for example, 200–300

Details of health check failures are returned via the API and the AWS Management Console



Targets

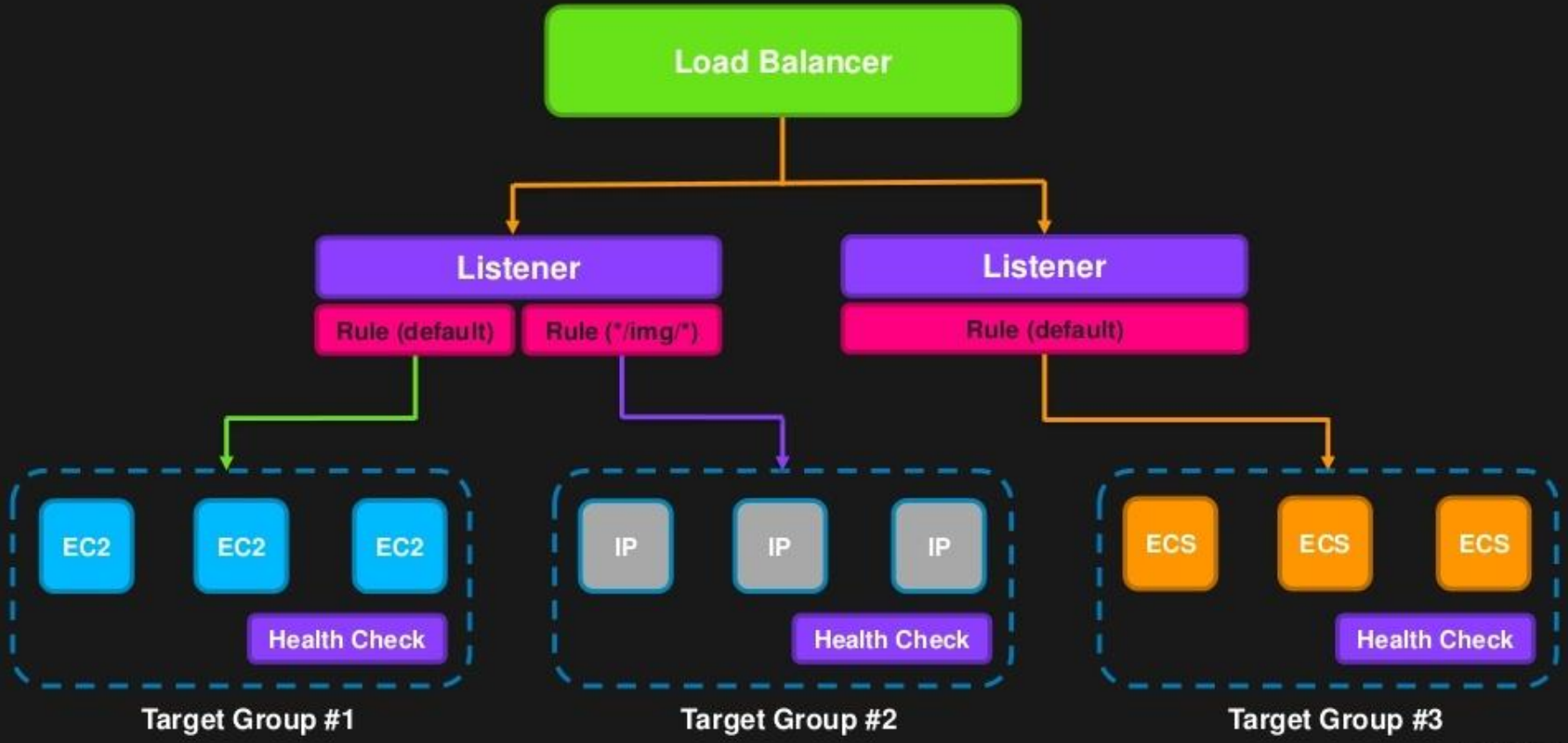


Support for EC2 instances and Amazon ECS containers, and IP Addresses

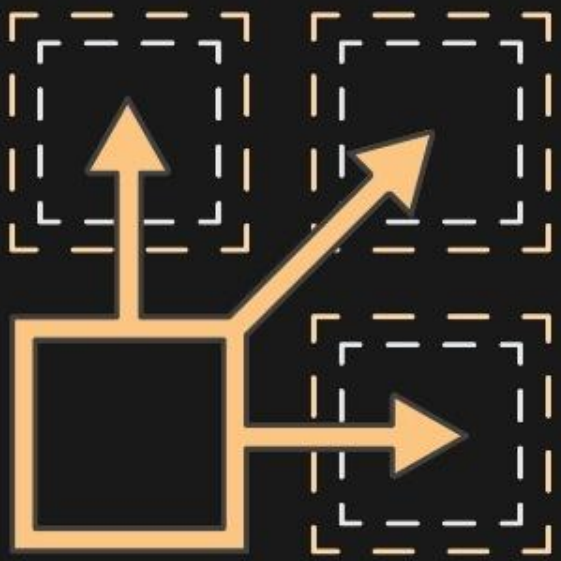
EC2 instances can be registered with the same target group using multiple ports

A single target can be registered with multiple target groups

IP addresses both accessible within your VPC or via DX and VPN



Rules



Each listener can have one or more rules for routing requests to target groups

Rules consist of conditions and actions

When a request meets the condition of the rule, the action is taken

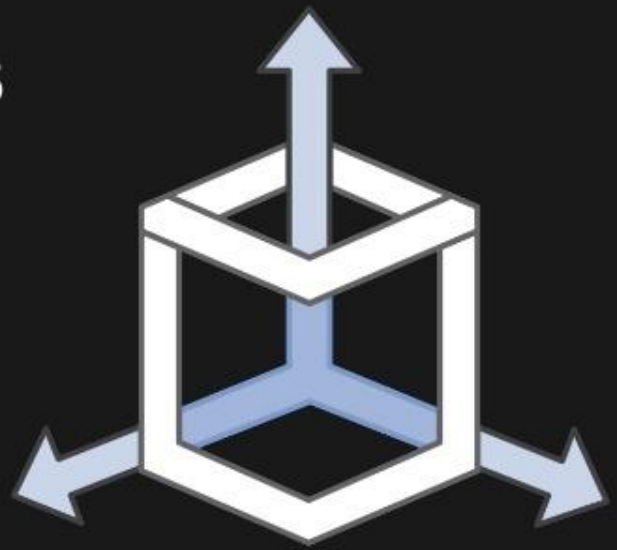
Today, rules can forward requests to a specified target group

Rules (continued)

Conditions can be specified in path pattern format

A path pattern is case-sensitive, can be up to 255 characters in length, and can contain any of the following characters:

- A-Z, a-z, 0-9
- _ - . $ / ~ " ' @ : +
- & (using &)
- * (matches 0 or more characters)
- ? (matches exactly 1 character)



Host-based routing

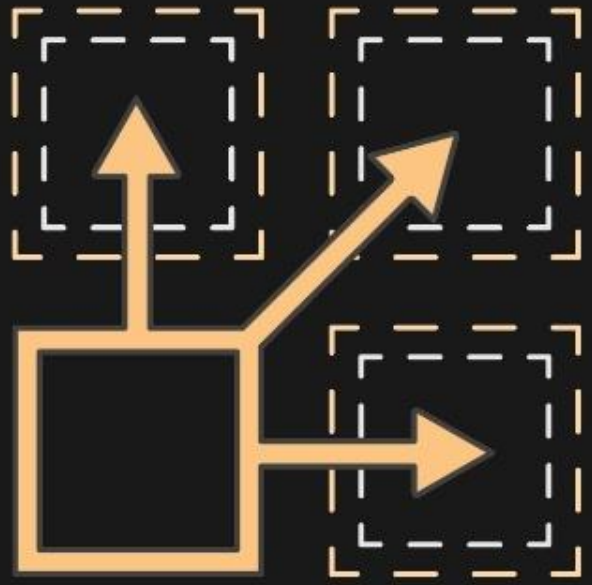
Route based on host field in the HTTP header

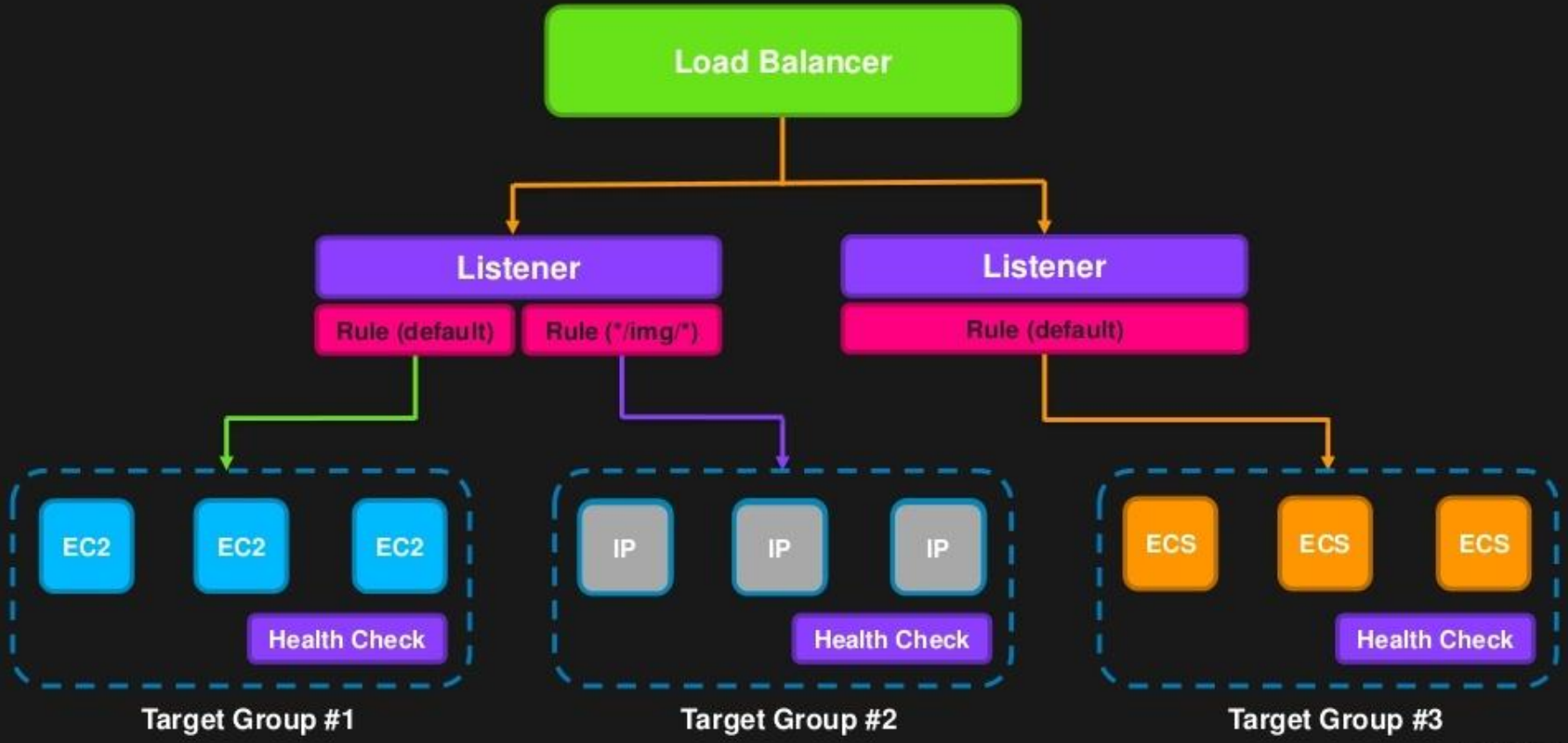
Support multiple domains using a single load balancer

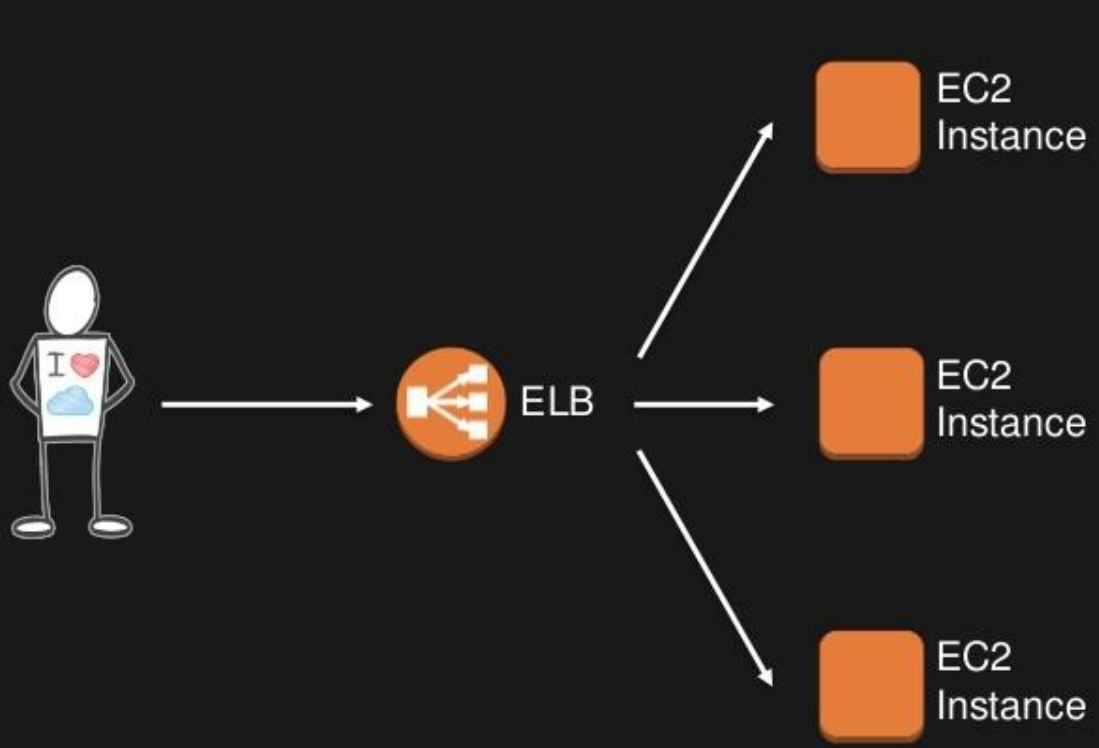
Route each host name to a different target group

Combine host-based routing and path-based routing

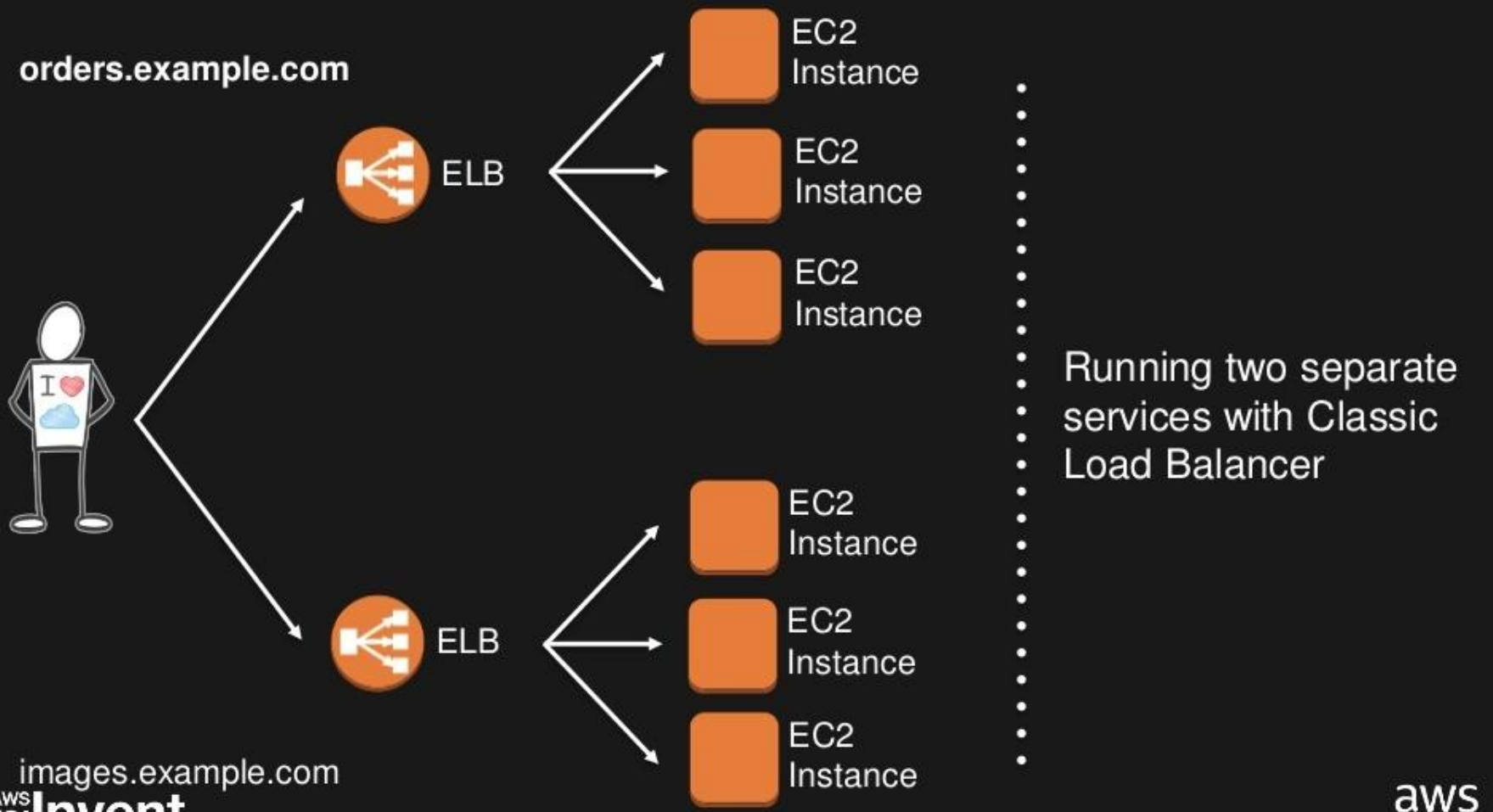
- 128-character limit
- A–Z, a–z, 0–9, -, .
- * (matches 0 or more characters)
- ? (matches exactly 1 character)





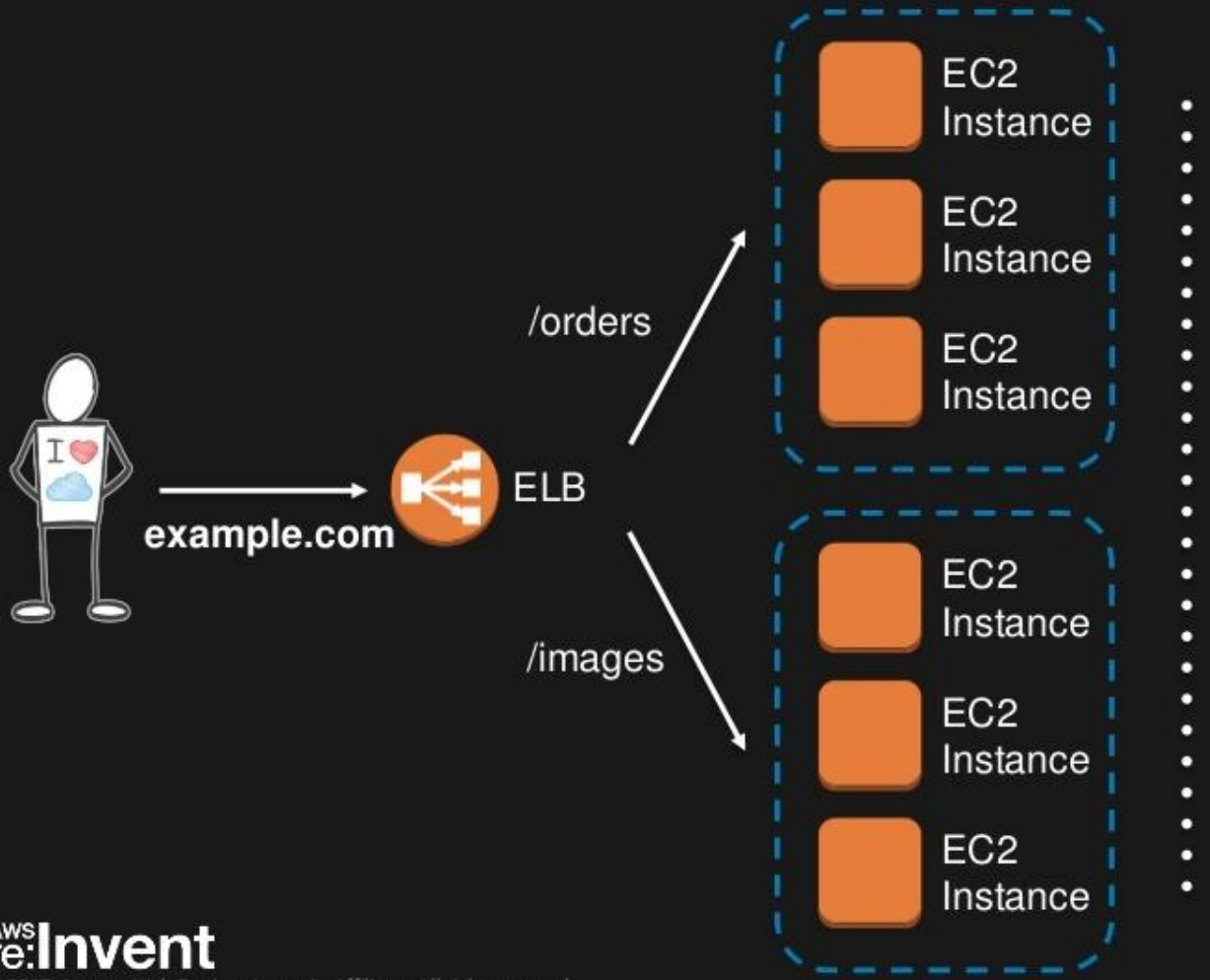


Amazon EC2 instances registered behind a Classic Load Balancer

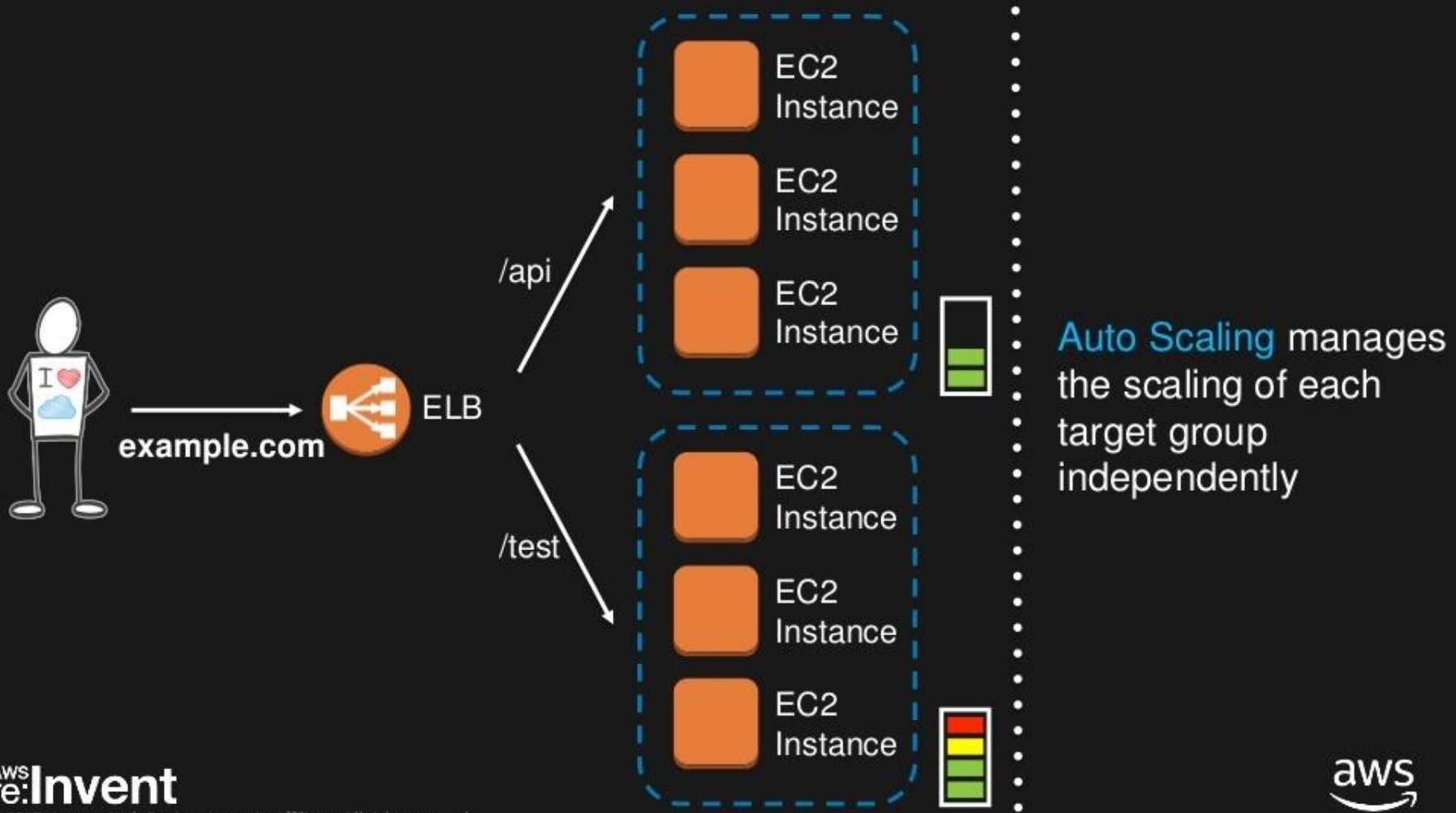


AWS re:Invent

aws



Application Load Balancer allows for multiple services to be hosted behind a single load balancer

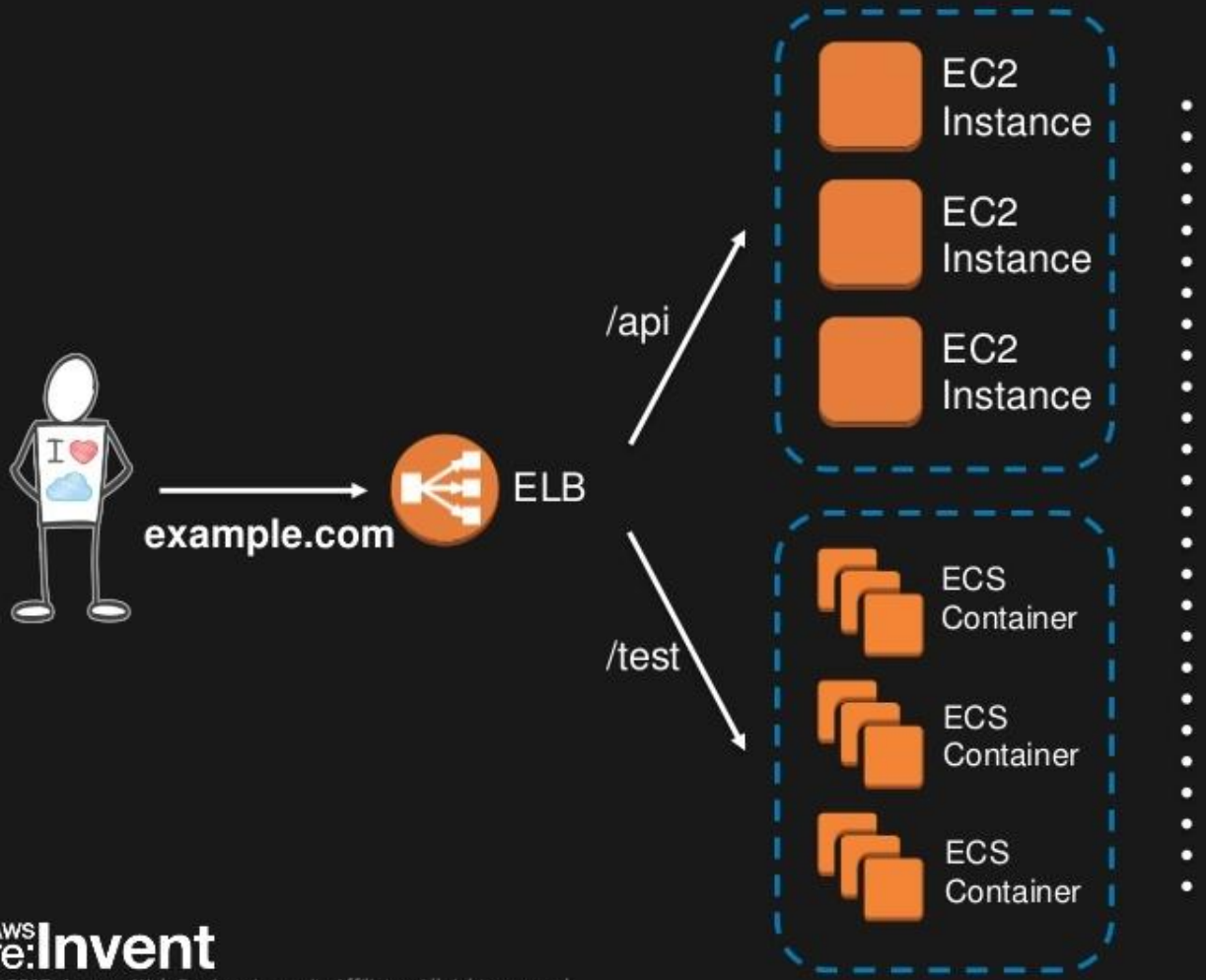


ECS integration

Application Load Balancer (ALB) is fully integrated with Amazon EC2 Container Service (Amazon ECS), managing target groups, paths, and targets

ECS will automatically register tasks with the load balancer using a dynamic port mapping

Can also be used with other container technologies



Application Load Balancer allows containers to be included in the target group

Predefined security policies

ELBSecurityPolicy-TLS-1-1-2017-01—Supports TLS 1.1 and above

ELBSecurityPolicy-TLS-1-2-2017-01—Strictly supports TLS1.2

ELBSecurityPolicy-2016-08—New default policy, same as Classic Load Balancer default policy

Windows XP Security Policy

Windows XP supported policy – Coming soon

Server Name Indication (SNI)

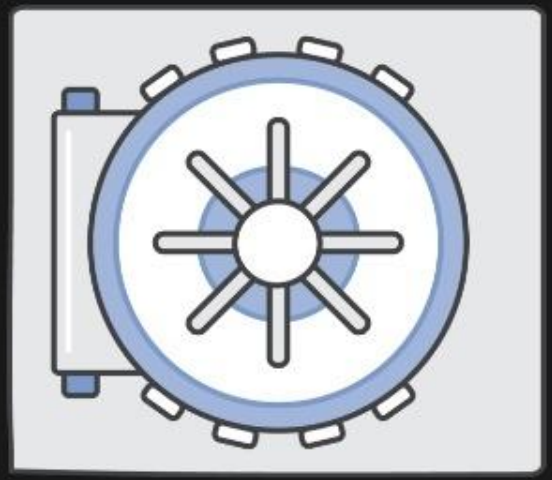
Host multiple TLS secured applications, each with its own TLS certificate

Bind multiple certificates to the same secure listener on your load balancer

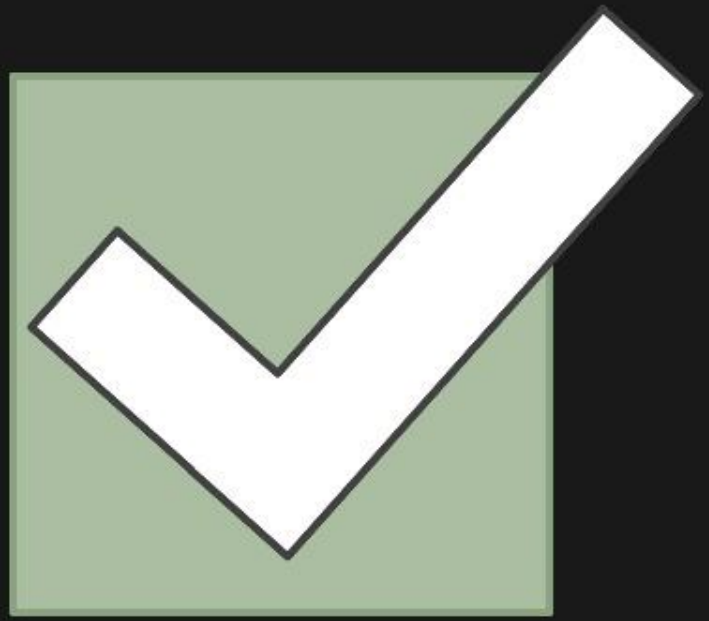
ALB will automatically choose the optimal TLS certificate for each client

Support for both the classic RSA algorithm and the newer, faster elliptic-curve-based ECDSA algorithm

Integration with ACM



Native IPv6 support



Application Load Balancer with WAF

Monitor web requests and protect web applications from malicious requests at the load balancer

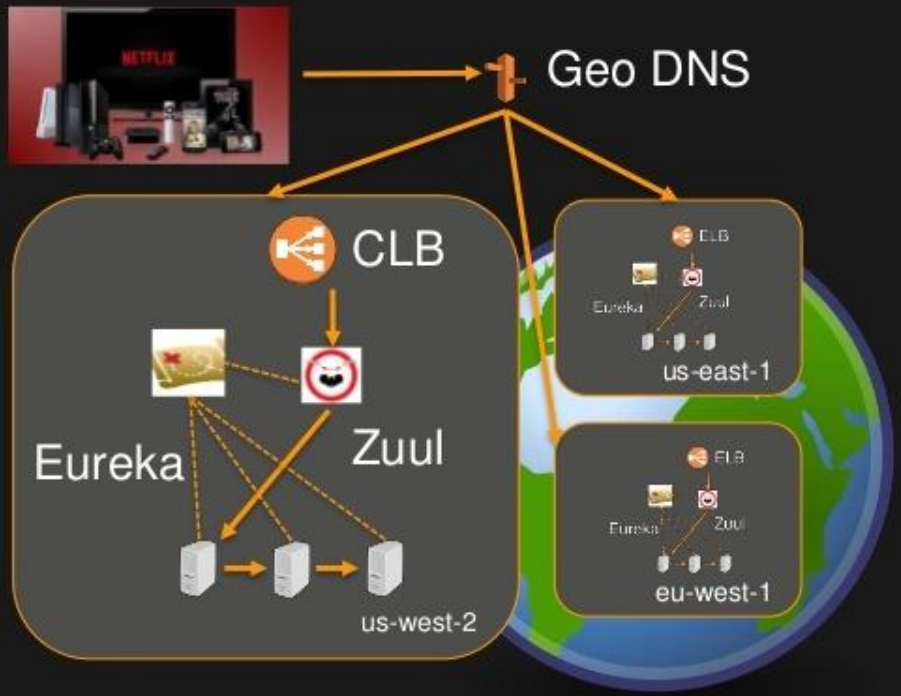
Block or allow requests based on conditions such as IP addresses

Preconfigured protection to block common attacks, such as SQL injection or cross-site scripting

Set up web ACLs and rules from WAF console and apply them to the load balancer

IP as target demonstration

Streaming service Load Balancing



Studio, content, partner Load Balancing

NETFLIX

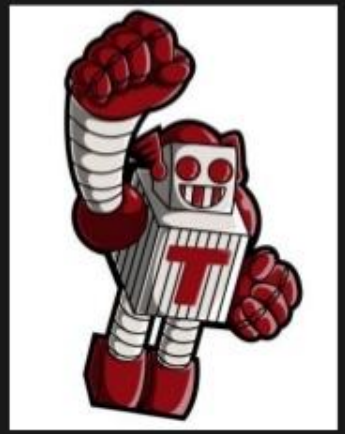
Netflix containers—Titus

EC2 applications should work in containers unchanged

- Provide IP per container
- Native VPC, security group, IAM role support

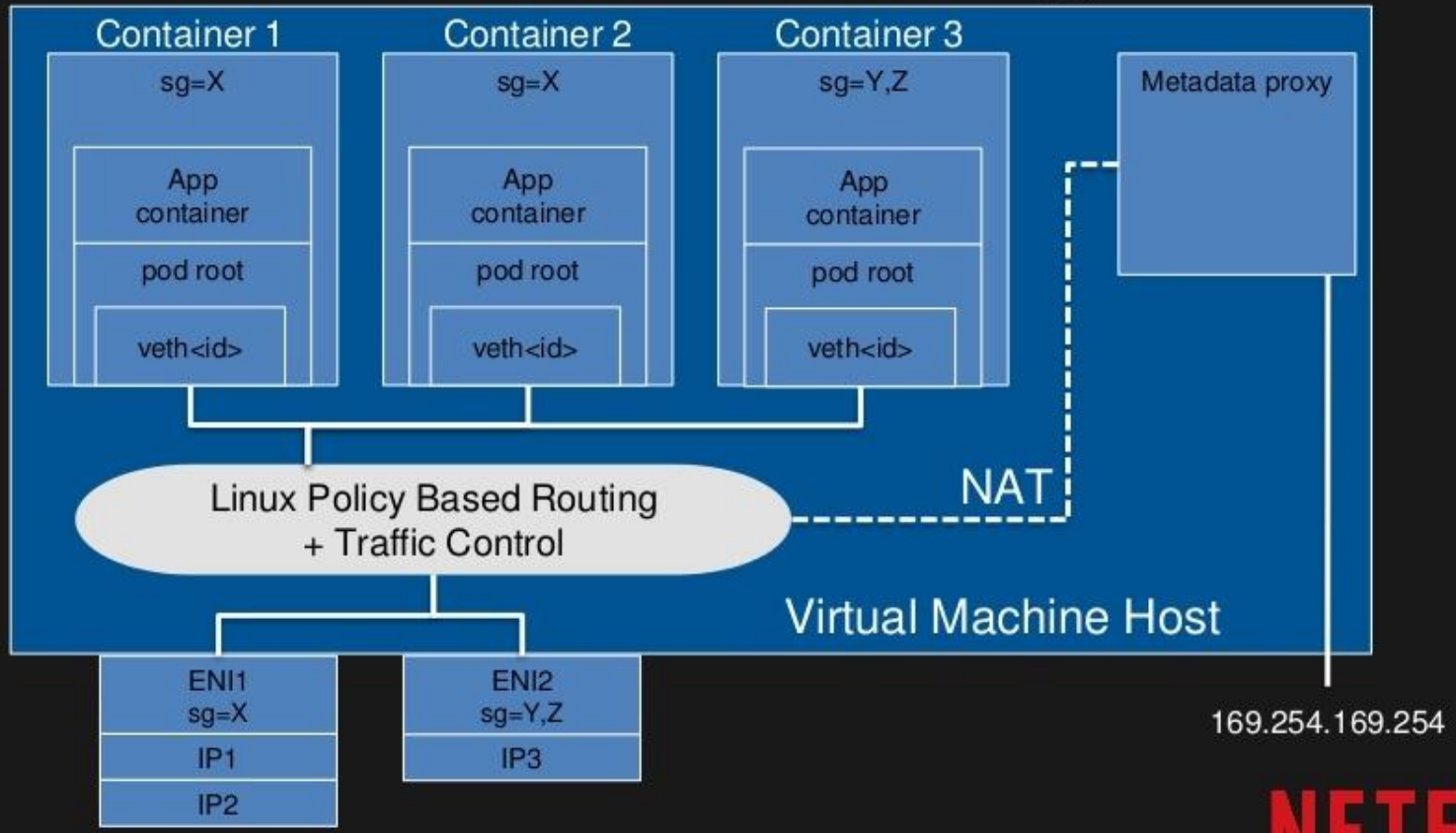
Problems for load balancing

- Multitenant container EC2 hosts presented problems



NETFLIX

Titus container networking



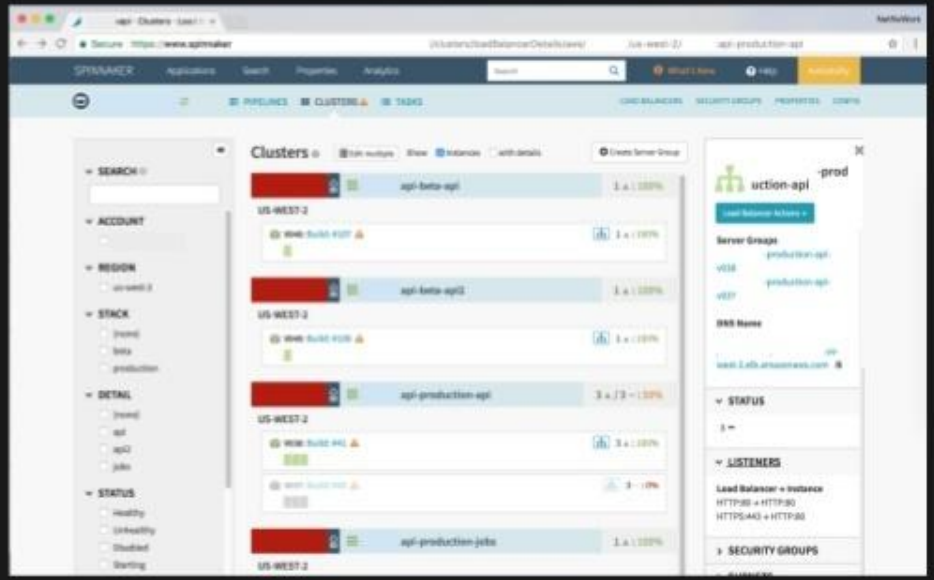
NETFLIX

Configuring ALBs with Spinnaker

Define Cluster

Associate IP Target Groups

Titus Scheduler

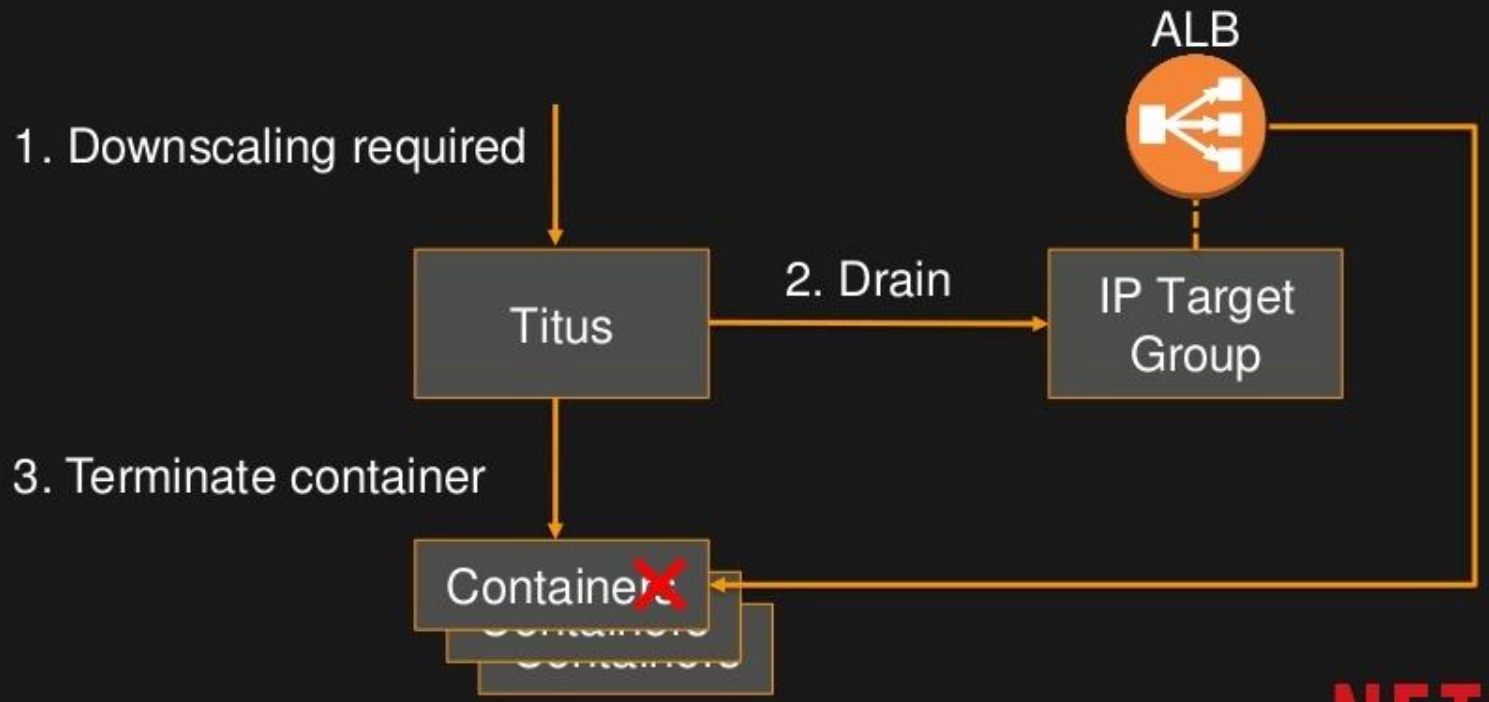


NETFLIX

ALB IP target group registration

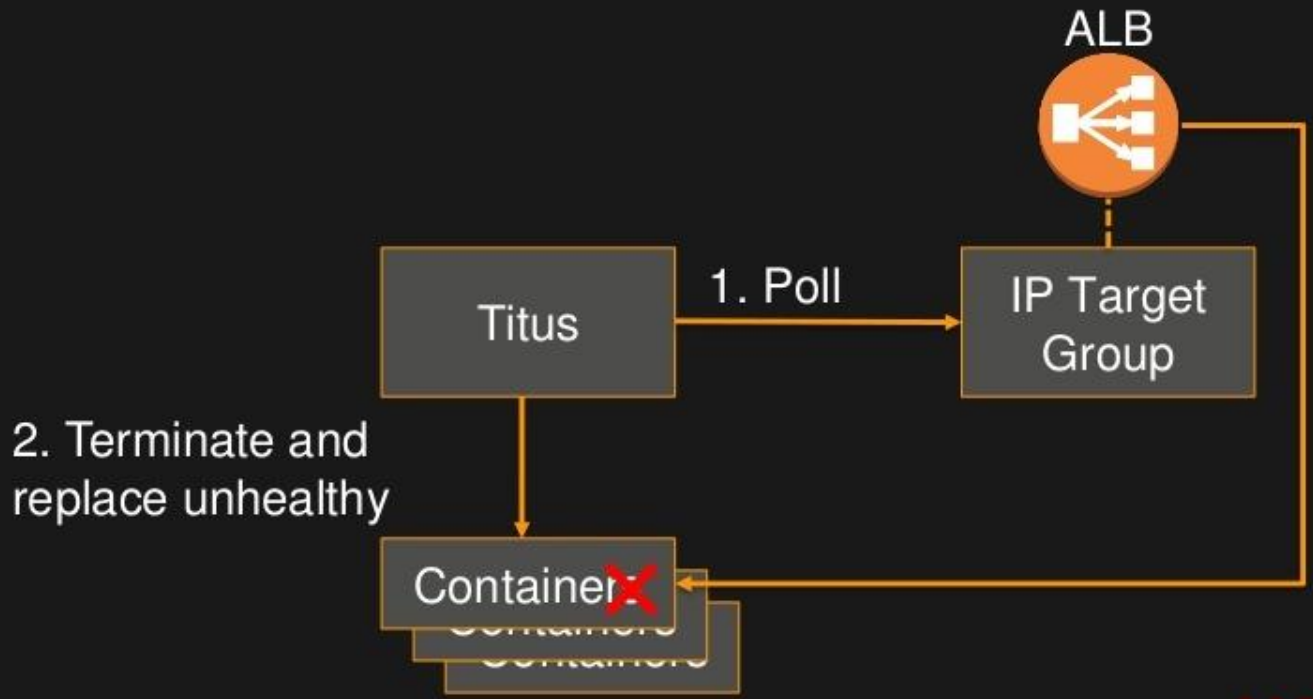
NETFLIX

Advanced features—downscaling



NETFLIX

Advanced features—healthchecks



NETFLIX

IP as a target

Use any IPv4 address from the load balancer's VPC CIDR for targets within load balancer's VPC in RFC 1918 ranges (10.0.0.0/8, 172.16.0.0/12, and 192.168.0.0/16)

Use any IP address from the RFC 6598 range (100.64.0.0/10) for targets located outside the load balancer's VPC (this includes Peered VPC, EC2-Classic, and on-premises targets reachable over Direct Connect or VPN).

Cross-zone load balancing

Requests distributed evenly across multiple Availability Zones

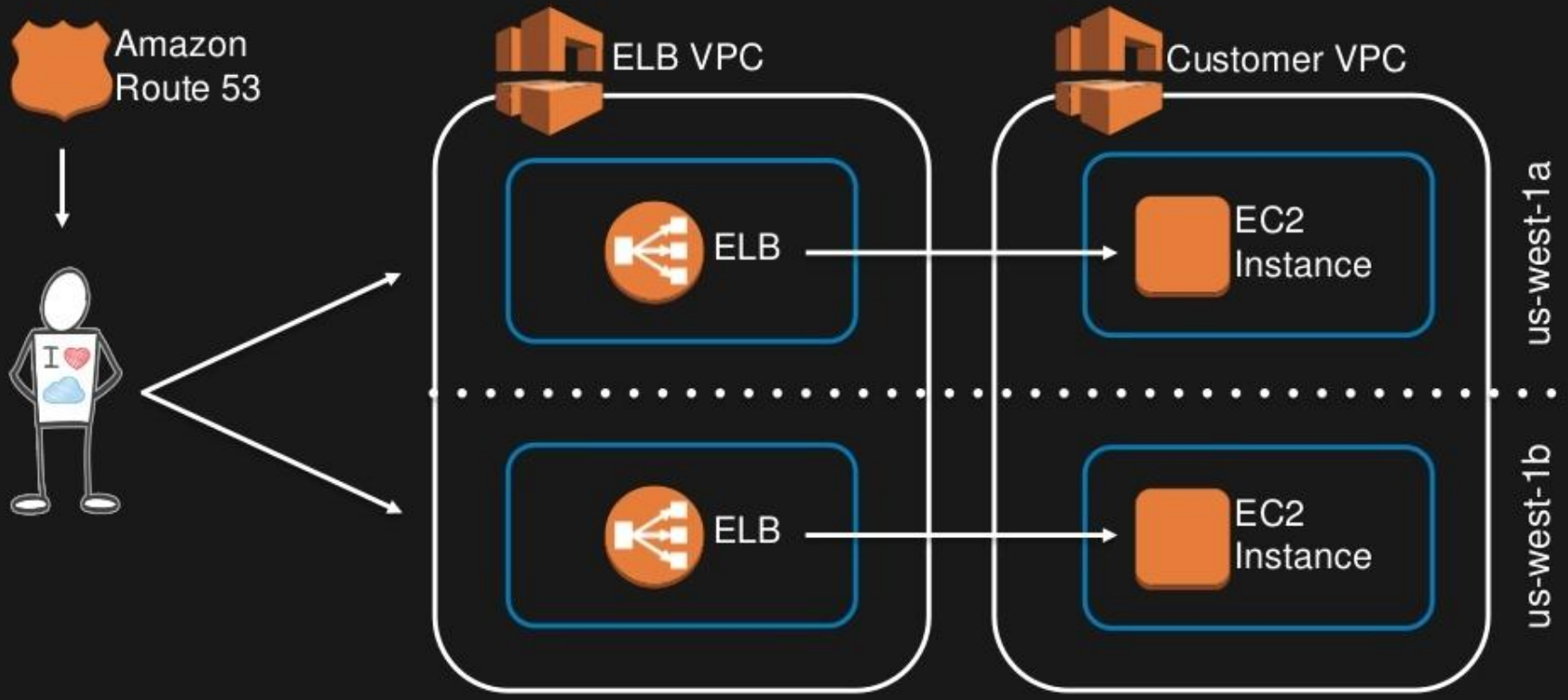
Load balancer absorbs impact of DNS caching

Eliminates imbalances in backend instance utilization

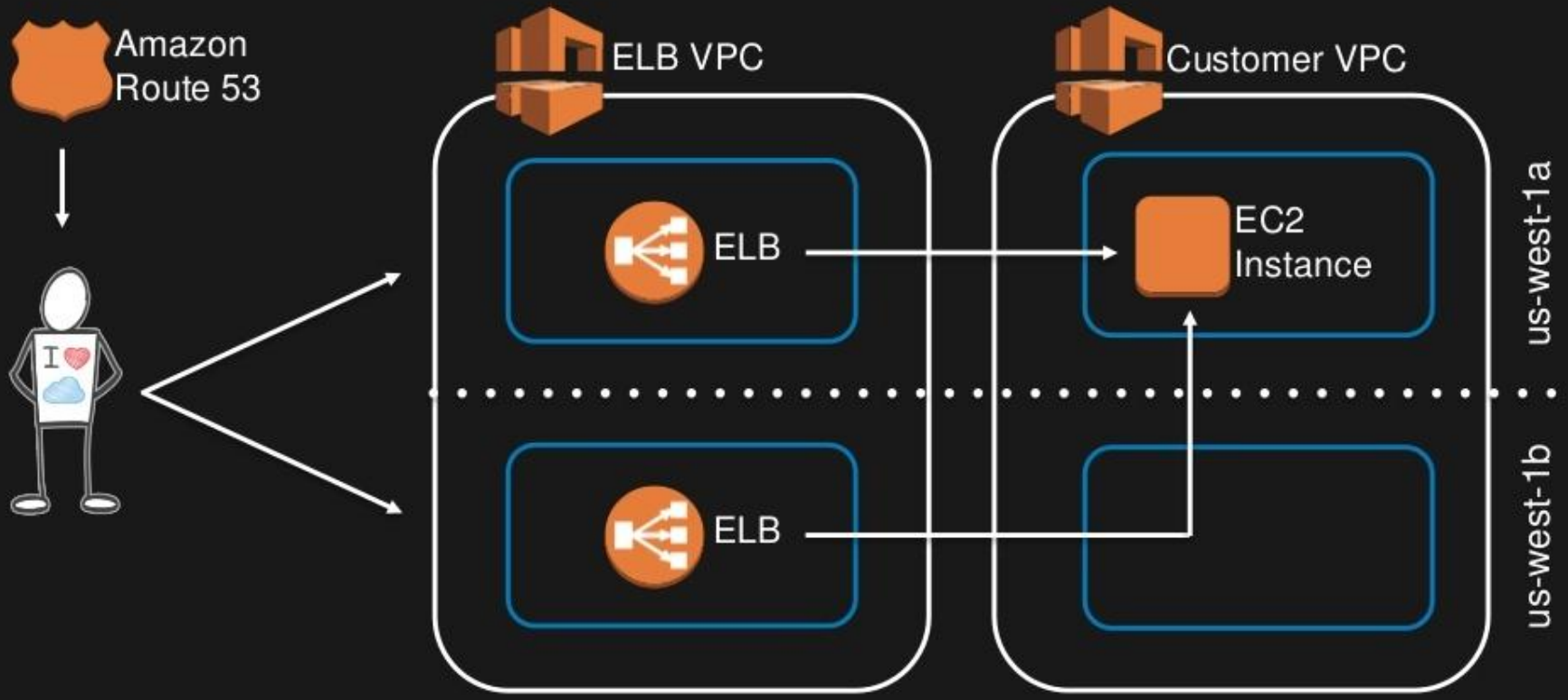
No additional bandwidth charge for cross-zone traffic

Enabled on all ALBs

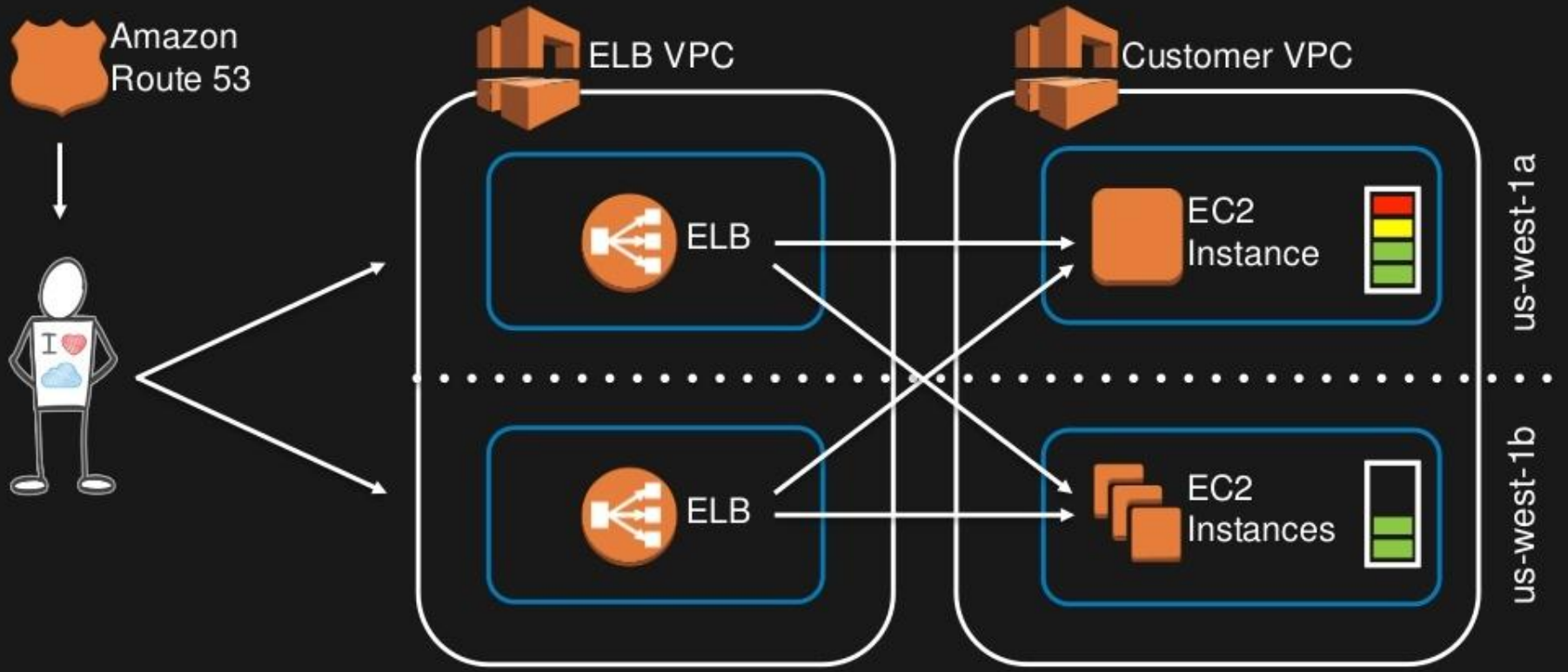
Multiple Availability Zones



Multiple Availability Zones



Cross-zone load balancing



Amazon CloudWatch metrics

CloudWatch metrics provided for each load balancer at 1-minute granularity

Request response times provided with percentile dimensions in the 90th, 95th, 99th or 99.9th percentile

Provide detailed insight into the health of the load balancer and application stack

CloudWatch alarms can be configured to notify or take action if any metric goes outside of the acceptable range

Healthy host count

The count of the number of healthy instances in each Availability Zone

Most common cause of unhealthy hosts is health check exceeding the allocated timeout

Test by making repeated requests to the back-end instance from another EC2 instance

View at the zonal dimension

Latency

Measures the elapsed time, in seconds, from when the request leaves the load balancer until the response is received

Test by sending requests to the back-end instance from another instance

Using min, average, and max CloudWatch stats, provide upper and lower bounds for latency

Debug individual requests using access logs

Rejected connection count

The number of connections that were rejected because the load balancer could not establish a connection with a healthy target in order to route the request

This replaces the surge queue metrics that are used by the Classic Load Balancer

Surge queues often impact client applications, which fast request rejection improves

Normally a sign of an under-scaled application

Target group metrics

The following metrics are now provided at the target group level, allowing for individual applications to be closely monitored:

- RequestCount
- HTTPCode_Target_2XX_Count
- HTTPCode_Target_3XX_Count
- HTTPCode_Target_4XX_Count
- HTTPCode_Target_5XX_Count
- TargetResponseTime (Latency)
- UnHealthyHostCount
- HealthyHostCount
- Request Count per Target

Access logs

Provide detailed information on each request processed by the load balancer

Includes request time, client IP address, latencies, request path, and server responses

Delivered to an Amazon S3 bucket every 5 or 60 minutes

Application Load Balancer pricing

With the Application Load Balancer, you pay only for what you use. You are charged for each hour or partial hour your Application Load Balancer is running and the number of Load Balancer Capacity Units (LCUs) used per hour.

- $0.0225 per Application Load Balancer-hour (or partial hour)
- $0.008 per LCU-hour (or partial hour)

Hourly charge is 10% cheaper than Classic Load Balancer today, reducing the cost for the virtually all of our customers.

Load balancer capacity units

An LCU measures the dimensions on which the Application Load Balancer processes your traffic (averaged over an hour). The three dimensions measured are:

- 25 new connections per second
- 3,000 active connections per minute
- 2.22 Mbps (which translates to 1 GB per hour)
- 1,000 rule evaluations per second

You are charged only on the dimension with the highest usage over the hour

Migrating to Application Load Balancer

Publishing LCU Metrics for Classic Load Balancer, which allows customers to estimate pricing if they migrate from Classic to ALB

Migration is as simple as creating a new Application Load Balancer, registering targets, and updating DNS to point to the new CNAME

Migration Wizard

The migration wizard in the AWS console makes it simple to create an Application Load Balancer with a configuration that is equivalent to your Classic Load Balancer

Enables you to quickly test your application with a new type of load balancer

After migration, you can configure the advanced features offered by the new load balancer

When should I use Application Load Balancer?

	Application Load Balancer	Network Load Balancer	Classic Load Balancer
Protocol	HTTP, HTTPS,HTTP/2	TCP	TCP, SSL, HTTP, HTTPS
SSL offloading	✓		✓
IP as target	✓	✓	
Path-based routing, Host-based routing	✓		
Static IP		✓	
WebSockets	✓	✓	
Container support	✓	✓	

For TCP, use Network Load Balancer

For all other use cases, use Application Load Balancer

Learn more

https://aws.amazon.com/elasticloadbalancing/

https://aws.amazon.com/documentation/elastic-load-balancing/

Follow me on Twitter: @davidpessis

AWS re:Invent

Thank you!