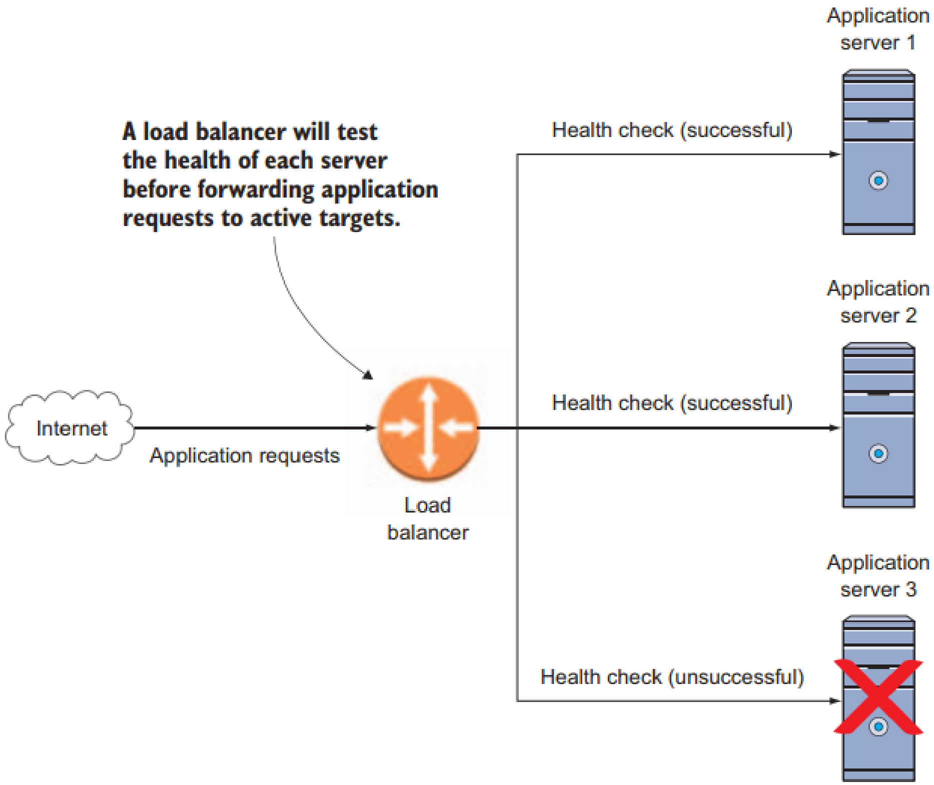
Lesson 11 Elastic Load Balancer

Michael Yang

What is ELB?

- Amazon's Elastic Load Balancing (ELB) system is designed for more than just managing failovers.
- A well designed balancer can do the following:
  1. Listen for traffic aimed at your web application
  2. Keep track of the health and availability of each of your application servers
  3. Redirect incoming traffic among only the servers that are currently able to respond

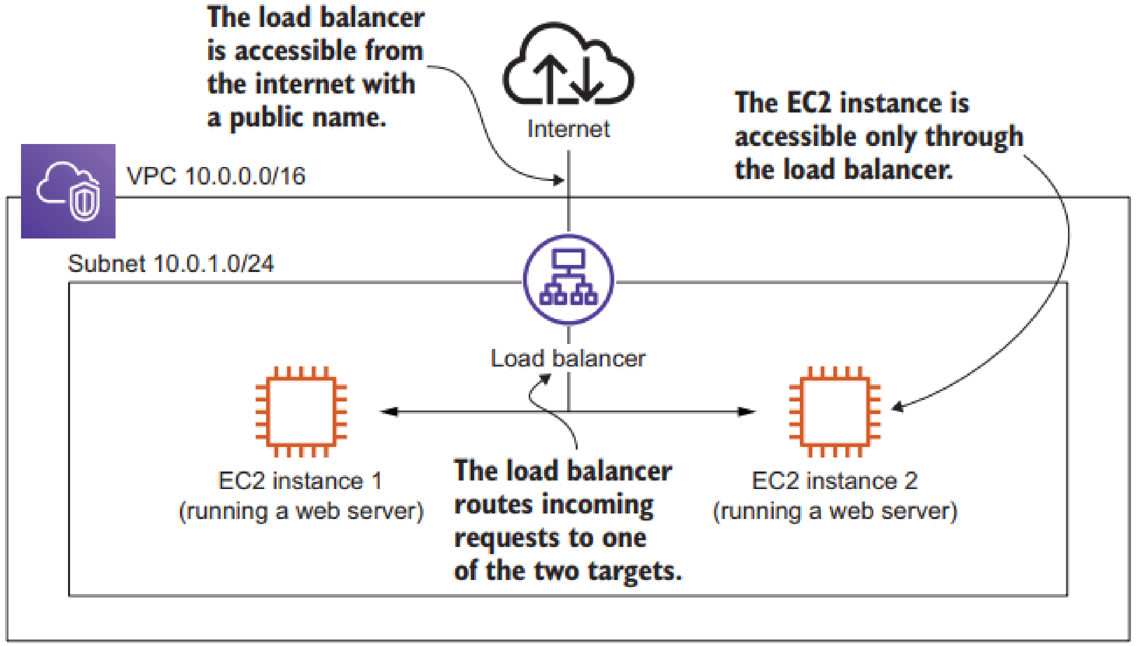


A load balancer will test the health of each server before forwarding application requests to active targets.

Another perspective

- ***synchronous decoupling*** is used when the client expects an immediate response. For example, a user expects an response to the request to load the HTML of a website with very little latency. The Elastic Load Balancing (ELB) service provides different types of load balancers that sit between your web servers and the client to decouple your requests synchronously. The client sends a request to the ELB, and the ELB forwards the request to a virtual machine or similar target. Therefore, the client does not need to

  know about the target; it knows only about the load balancer.



The load balancer is accessible from the internet with a public name.
Internet
The EC2 instance is accessible only through the load balancer.
VPC 10.0.0.0/16
Subnet 10.0.1.0/24
Load balancer
EC2 instance 1 (running a web server)
The load balancer routes incoming requests to one of the two targets.
EC2 instance 2 (running a web server)

ALB , NLB, or CLB?

- AWS offers different types of load balancers through the Elastic Load Balancing (ELB) service. All load balancer types are fault tolerant and scalable. They differ in supported protocols and features as follows:

Application Load Balancer (ALB)—HTTP, HTTPS

Network Load Balancer (NLB)—TCP, TCP TLS

Classic Load Balancer (CLB)—HTTP, HTTPS, TCP, TCP TLS

Consider the CLB deprecated. As a rule of thumb, use the ALB whenever the HTTP/HTTPS protocol is all you need, and the NLB for all other scenarios. *With ALB, it is a requirement that you enable at least two or more Availability Zones.*

ALB

- **Application Load Balancer** is best suited for load balancing of HTTP and HTTPS traffic and provides advanced request routing targeted at the delivery of modern application architectures, including microservices and containers.
- Applications need advanced routing (host-based, URL-based, query string based).
- Run multiple services (microservices) behind a single load balancer.

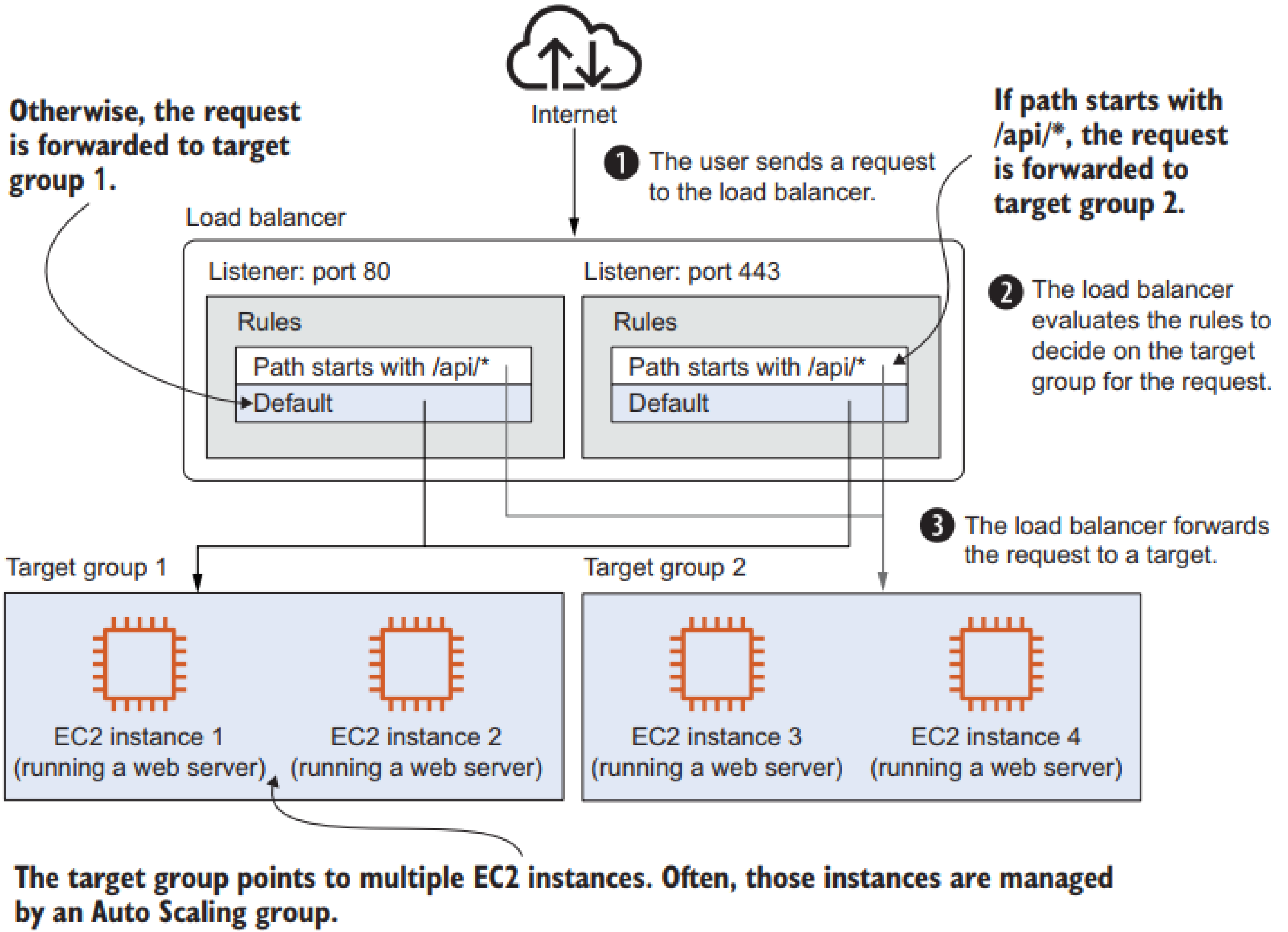
NLB

- **Network Load Balancer** is best suited for load balancing of TCP traffic where extreme performance is required. It is capable of handling millions of requests per second while maintaining ultra-low latencies, and it is optimized to handle sudden and volatile traffic patterns.
- You want to share/expose your services (e.g. SaaS services) to other consumers in different VPCs using PrivateLink VPC Endpoint.
- You need a static IP address that can be used by applications as the front-end IP of the load balancer.

ALB Core Concepts

- ***Load balancer***—Defines some core configurations
- ***Listener***—The listener defines the port and protocol that you can use to make requests to the load balancer.
- ***Target group***—A target group defines your group of backends. The target group is responsible for checking the backends by sending periodic health checks.
- ***Listener rule***—Optional. You can define a listener rule. The rule can choose a different target group based on the HTTP path or host. Otherwise, requests are forwarded to the default target group defined in the listener

ALB



Otherwise, the request is forwarded to target group 1.

1 The user sends a request to the load balancer.

If path starts with /api/*, the request is forwarded to target group 2.

2 The load balancer evaluates the rules to decide on the target group for the request.

3 The load balancer forwards the request to a target.

The target group points to multiple EC2 instances. Often, those instances are managed by an Auto Scaling group.

Create an ALB

- Once the instances are running, you’ll need to jump through these five steps to get a load balancer on the job:
  1 Create a target group, and configure a health check.
  2 Register your four instances with the target group.
  3 Create a load balancer, and associate it with the two subnets hosting your instances.
  4 Create a security group for both instances and for the load balancer.
  5 Associate your target group with the load balancer.

Target Group

- Once a load balancer is configured with the addresses of all of your servers, organized into what AWS called ***target groups***, its own network address becomes the only URL your users need to access. Visitors don’t need to know the individual DNS or IP addresses of each of your servers—nor do they have to worry about which ones are best prepared to handle their needs. As long as you’ve set your load balancer to listen for requests against a particular address and associated your application’s DNS address with the balancer’s endpoint—which, for AWS balancers, is provided as part of the balancer description you’ll see later—all that is taken care of automatically and invisibly.

Pricing

- AWS load balancers are charged per hour of active use and for the data they transfer, which includes just about anything moving in or out through the balancer. To give you a general idea of how that works, at the time of writing you’d be billed $0.025/hour to run a balancer—which is $18 for a month—and $0.008 per GB of transferred data. 100 GB of total monthly transfers would come to $0.80. This, naturally, is over and above any charges you incur for running EC2 instances.