Compare Load Balancing, AutoScaling and Serverless Computing

Yichuan Zhang& Yang Yuan& Xiao Li

Load Balancing

Wiki:

Load balancing improves the distribution of workloads across multiple computing resources.It aims to optimize resource use, maximize throughput, minimize response time, and avoid overload of any single resource.

Load balancing

Elastic Load Balancing:

Elastic Load Balancing distributes incoming application traffic across multiple EC2 instances, in multiple Availability Zones. This increases the fault tolerance of your applications.

Load balancing

Features of Elastic Load Balancing

Elastic Load Balancing supports three types of load balancers: Application Load Balancers, Network Load Balancers, and Classic Load Balancers. You can select a load balancer based on your application needs.

Google Cloud

Load balancing

Network-Based Load Balancing

Content-Based Load Balancing

Cross-region Load Balancing

Google Cloud

Load balancing

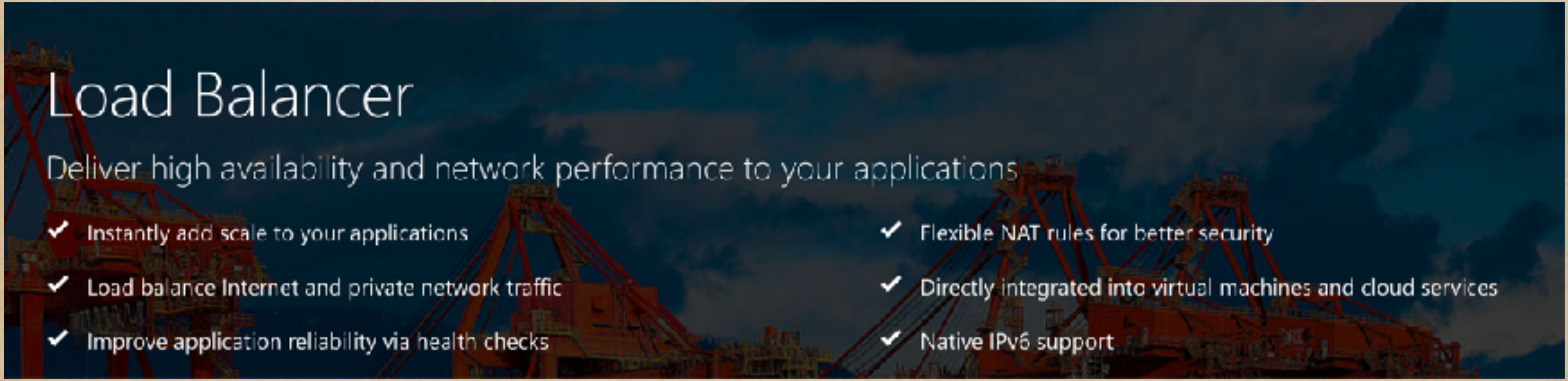
Software-Defined Load Balancing

Cloud Load Balancing is a fully distributed, software-defined managed service for all traffic. You can apply Cloud Load Balancing to all of traffic: HTTP, TCP/SSL, and UDP.

Microsoft Azure

Load balancing

Features:



Load Balancer

Deliver high availability and network performance to your applications

- ✓ Instantly add scale to your applications
- ✓ Load balance Internet and private network traffic
- ✓ Improve application reliability via health checks
- ✓ Flexible NAT rules for better security
- ✓ Directly integrated into virtual machines and cloud services
- ✓ Native IPv6 support

Load balancing

Simplify load balancing for applications:

Create highly-available and scalable applications in minutes. Azure Load Balancer supports TCP/UDP-based protocols such as HTTP, SMTP, and protocols used for real-time voice and video messaging applications

AutoScaling

Wiki:

A method used in cloud computing, whereby the amount of computational resources in a server farm, typically measured in terms of the number of active servers, scales automatically based on the load on the farm. It is closely related to, and builds upon, the idea of load balancing.

Auto scaling

Amazon EC2 Auto Scaling:

Specify the minimum number of instances in each Auto Scaling group, and Auto Scaling ensures that your group never goes below this size. Specify the maximum number of instances in each Auto Scaling group, and Auto Scaling ensures that your group never goes above this size.

Auto scaling

Application Auto Scaling:

Scale the following compute and data resources for cloud-based web applications:

- Amazon ECS services
- Spot Fleet requests
- Amazon EMR clusters
- AppStream 2.0 fleets
- DynamoDB tables and global secondary indexes
- Aurora replicas
- Amazon SageMaker endpoint variants

Google Cloud

Auto scaling

Works by scaling up or down your instance group. Adds more instances to your instance group when there is more load(upscaling), and removes instances when the need for instances is lowered(downscaling)

Serverless computing

Wiki:

A cloud computing execution model in which the cloud provider dynamically manages the allocation of machine resources. Pricing is based on the actual amount of resources consumed by an application, rather than on pre-purchased units of capacity.[1] It is a form of utility computing.

Serverless computing

AWS Serverless Application Repository:

The AWS Serverless Application Repository makes it easy for developers and enterprises to quickly find, deploy, and publish serverless applications in the AWS Cloud.

Serverless computing

Serverless:

A new paradigm of computing that abstracts away the complexity associated with managing servers
for mobile and API backends, ETL, data processing jobs, databases, and more

Serverless computing

serverless computing:

Serverless computing is the abstraction of servers, infrastructure, and operating systems. When you build serverless apps you don't need to provision and manage any servers, so you can take your mind off infrastructure concerns. It is driven by the reaction to events and triggers happening in near-real-time—in the cloud.

Serverless computing

Why build serverless applications?

- Benefit from a fully managed service
- Scale flexibly
- only pay for resources you use

Thank you!