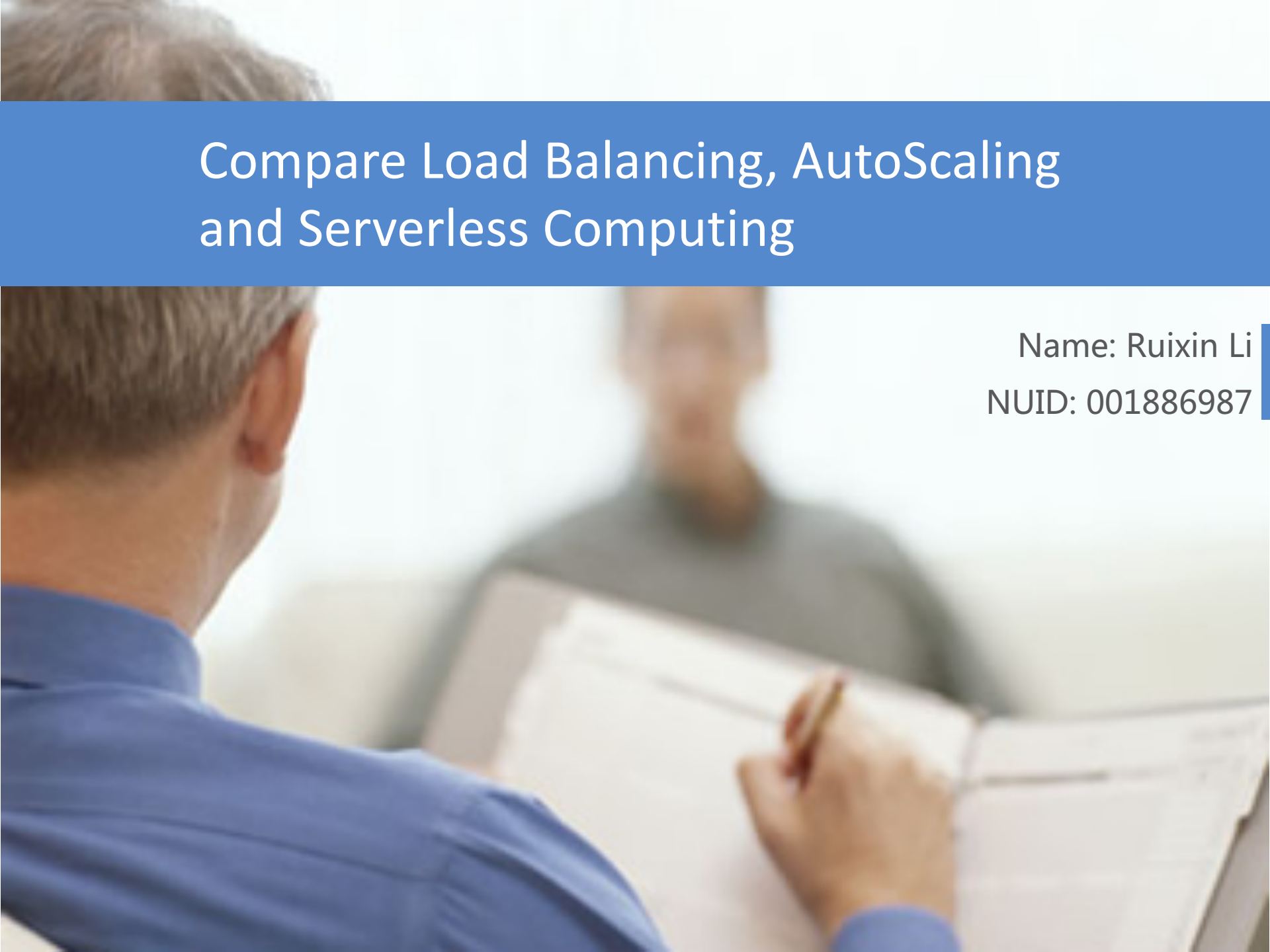
Compare Load Balancing, AutoScaling and Serverless Computing



Name: Ruixin Li

NUID: 001886987

Load Balancing

Introduction

A load balancer is a device that acts as a reverse proxy and distributes network or application traffic across a number of servers.

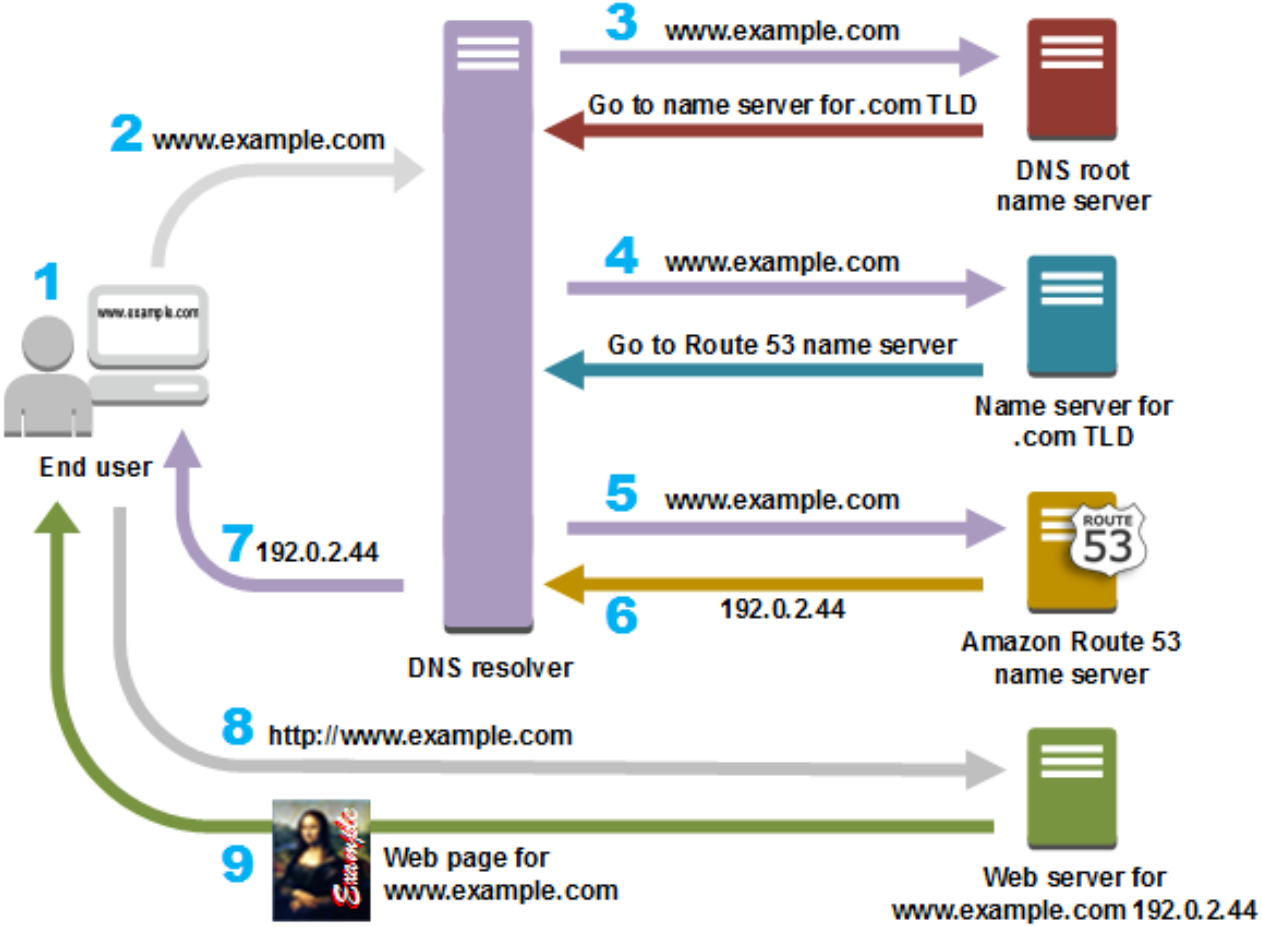
In computing, load balancing improves the distribution of workloads across multiple computing resources, Load balancing aims to optimize resource use, maximize throughput, minimize response time, and avoid overload of any single resource. Using multiple components with load balancing instead of a single component may increase reliability and availability through redundancy.

DNS Traffic Load Balancing

AWS: Route 53 provides a range of routing and load balancing services such as directing weighted portions of traffic to different endpoints, choosing endpoints based on latency, routing traffic based on the geographical location of the user, and failing over to an alternate endpoint/region if the intended endpoint becomes unavailable. Route 53 can also use regular health traffic to send alerts and notifications via Cloud Watch.

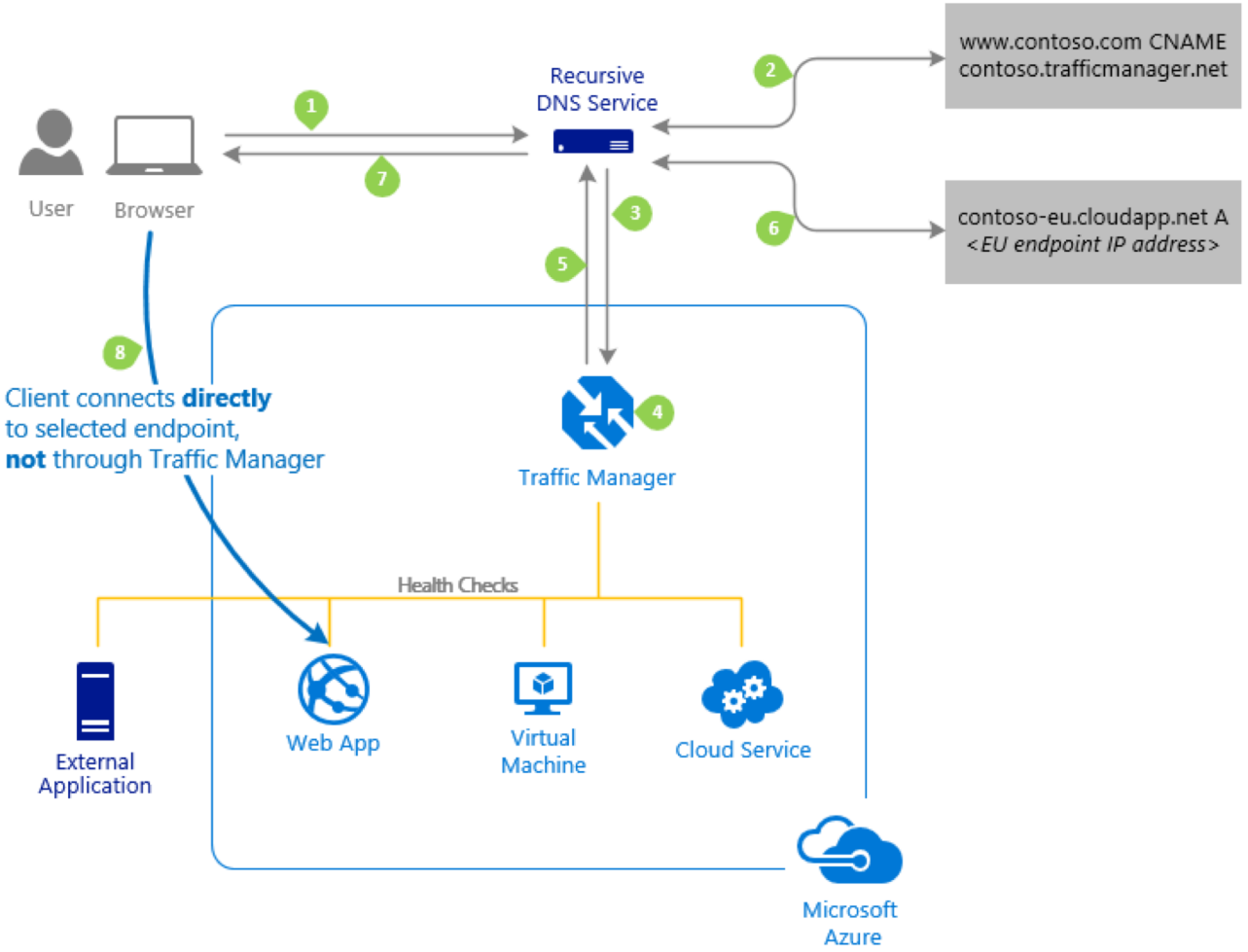
Microsoft Azure: Traffic Manager is a DNS based traffic routing solution. It provides a number of distribution policies including weighted round robin, automatic fail-over to healthy endpoints, and routing traffic to the nearest location. Traffic Manager can route traffic to services in any region as well as non-Azure endpoints. Health checking is achieved by periodically polling an HTTP endpoint. There is an additional charge for each health check endpoint.

DNS Traffic Load Balancing



Route 53

DNS Traffic Load Balancing



Traffic Manager

Four-tier Load Balancing

AWS:

- It has the Ability to handle volatile workloads and scale to millions of requests per second.
- A load balancer node selects a target using a flow hash algorithm, based on the protocol, source IP address, source port, destination IP address, destination port, and TCP sequence number.
- Elastic Load Balancing creates a network interface for each Availability Zone you enable. Each load balancer node in the Availability Zone uses this network interface to get a static IP address.
- You can configure a target group so that you register targets by instance ID or IP address.
- You can add and remove targets from your load balancer as your needs change, without disrupting the overall flow of requests to your application.
- You can configure health checks, which are used to monitor the health of the registered targets so that the load balancer can send requests only to the healthy targets.

Four-tier Load Balancing

Feature	Azure Load Balancer	Compute Engine Network Load Balancer
TCP/UDP load balancing	Yes	Yes
Internal load balancing	Yes	Yes
Internet-facing load balancing	Yes	Yes
Supported application-layer protocols	Any	Any
Supported endpoints	Azure VMs (excluding Basic VMs), Cloud Services role instances	Target pools, target VM instances, backend services (internal load balancing only)
Health monitoring	Yes	Yes
Default load balancing mode	5-tuple (source IP and destination IPs, source and destination ports, protocol type)	5-tuple (source and destination IPs, source and destination ports, protocol type)
Session affinity modes	2-tuple (source and destination IPs), 3-tuple (source and destination IPs, port)	2-tuple (source and destination IPs), 3-tuple (source and destination IPs, protocol type)

Seven-tier Load Balancing

AWS: Application Load Balancer is designed specifically to support modern application workloads such as containerised applications, web sockets and HTTP/2 traffic. It has built-in health monitoring, operational management through Cloudwatch, logging, SSL termination and sticky sessions. Pricing is based on the number of deployed load balancers per hour plus a charge for the amount of data processed.

Microsoft Azure: Application Gateway is a Layer 7 (HTTP/HTTPS) load balancer. It can be used as an internet facing or private load balancer providing round robin, url based routing or cookie affinity traffic distribution. Application Gateway uses HTTP probes to monitor health and can provide SSL termination. Pricing is based on the time that Application Gateway is running and the amount of data transferred.

GCC: HTTP(S) Load Balancing provides global load balancing for modern web based applications. Traffic is routed to the nearest instance group to the calling user via an anycast IP address and SSL can be terminated at the load balancer. Distribution strategies include URL and content based routing.

AutoScaling

AutoScaling

AWS: It is possible to autoscale EC2 instances within a VPC according to a set of performance metric thresholds defined. There are no additional fees for autoscaling other than the underlying EC2 charges.

Microsoft Azure: Virtual machine scale sets allow VM instances to be automatically added or removed from a VNET based on a set of rules. Rules can be defined on a range of criteria includnig performance metric thresholds, day/time and message queue size. When using Azure Load Balancer, new instances will be automatically registered with or removed from the load balanced set.

GCC: Google supports auto scaling groups of compute instances based on various run time performance statistics. Scaling can be triggered through metrics such as CPU utilisation, HTTP load balancing throughput, Stackdriver monitoring alerts or in response to Cloud Pub/Sub queue metrics.

Serverless Computing

Serverless Computing

AWS Lambda: Lambda supports a range of runtime environments including NodeJS, Python, Java and C# and has some advanced features like request chaining and edge processing. The product is quickly evolving towards mobile and IoT based use cases.

Microsoft Azure Functions: Azure Functions launched with a variety of supported runtimes including JavaScript, C#, Python and PHP. Azure's approach is to provide a functional IDE in their portal to help you prototype and deploy functions. Note the similarities between their machine learning studio product, a product philosophy which ties in nicely with Visual Studio.

Google Cloud Functions: Google Cloud Functions still only supports a single runtime environment using NodeJS.

Thanks for your time