LOADBALANCING, AUTOSCALING AND SERVERLESS COMPUTING

Service & Capability Comparison between
AWS, Azure and GCP

Junyuan Gu 001825583

Azure Load Balancing

- Azure Load Balancer is available in two different SKUs: Basic and Standard. There are differences in scale, features, and pricing. Any scenario possible with Basic Load Balancer can also be created with Standard Load Balancer, although the approach might differ slightly. As you learn about Load Balancer, it is important to familiarize yourself with the fundamentals and SKU-specific differences.
- Both SKUs have the same general API and structure. Standard Load Balancer is a new Load Balancer product and largely a superset of Basic Load Balancer. There are important and deliberate differences between both products.
- New designs should consider using Standard Load Balancer.

SKU: Basic vs Standard

	Standard SKU	Basic SKU
Backend pool size	up to 1000 instances	up to 100 instances
Backend pool endpoints	any virtual machine in a single virtual network, including blend of virtual machines, availability sets, virtual machine scale sets.	virtual machines in a single availability set or virtual machine scale set
Availability Zones	zone-redundant and zonal frontends for inbound and outbound, outbound flows mappings survive zone failure, cross-zone load balancing	/
Diagnostics	Azure Monitor, multi-dimensional metrics including byte and packet counters, health probe status, connection attempts (TCP SYN), outbound connection health (SNAT successful and failed flows), active data plane measurements	Azure Log Analytics for public Load Balancer only, SNAT exhaustion alert, backend pool health count
HA Ports	internal Load Balancer	/
Secure by default	default closed for public IP and Load Balancer endpoints and a network security group must be used to explicitly whitelist for traffic to flow	default open, network security group optional
Outbound connections	Multiple frontends with per rule opt-out. An outbound scenario must be explicitly created for the virtual machine to be able to use outbound connectivity. VNet Service Endpoints can be reached without outbound connectivity and do not count towards data processed. Any public IP addresses, including Azure PaaS services not available as VNet Service Endpoints, must be reached via outbound connectivity and count towards data processed. When only an internal Load Balancer is serving a virtual machine, outbound connections via default SNAT are not available. Outbound SNAT programming is transport protocol specific based on protocol of the inbound load balancing rule.	Single frontend, selected at random when multiple frontends are present. When only internal Load Balancer is serving a virtual machine, default SNAT is used.

Elastic Load Balancing - AWS

- Elastic Load Balancing automatically distributes incoming application traffic across multiple targets, such as Amazon EC2 instances, containers, and IP addresses. It can handle the varying load of your application traffic in a single Availability Zone or across multiple Availability Zones. Elastic Load Balancing offers three types of load balancers that all feature the high availability, automatic scaling, and robust security necessary to make your applications fault tolerant.
- You can also use hybrid load balancing to benefit separate applications where one is in a VPC and the other is in an on-premises location. Simply put the VPC targets in one target group and the on-premises targets in another target group, and then use content based routing to route traffic to each target group.

GOOGLE CLOUD LOAD BALANCING

- Cloud Load Balancing comes in a variety of flavors and is integrated with Google Cloud CDN for optimal application and content delivery.
- Global Load Balancing with Single Anycast IP

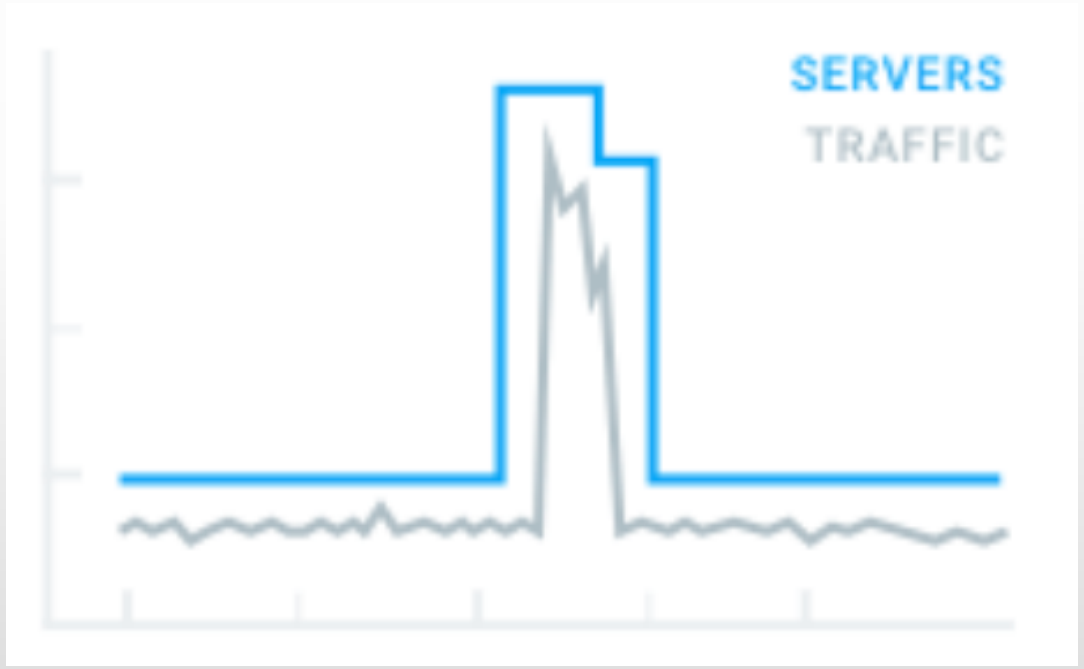
It provides cross-region load balancing including automatic multi-region failover which gently moves traffic in fractions if backends become unhealthy. In contrast to DNS-based Global Load Balancing solutions, Cloud Load Balancing reacts instantaneously to changes in users, traffic, network, backend health and other related conditions.

- Software-Defined Load Balancing

It is a fully distributed, software-defined, managed service for all traffic. It is not an instance or device based solution, so you won't be locked into physical load balancing infrastructure or face the HA, scale and management challenges inherent in instance based LBs. You can apply Cloud Load Balancing to all of your traffic: HTTP(S), TCP/SSL, and UDP. You can also terminate your SSL traffic with HTTPS Load Balancing and SSL proxy.

Cloud Auto Scaling

- Cloud Load Balancing can scale as your users and traffic grow, including easily handling huge, unexpected and instantaneous spikes by diverting traffic to other regions in the world that can take traffic. Autoscaling does not require pre-warming, you can scale from zero to full throttle in a matter of seconds.



AWS Auto Scaling

- PLAN SCALING DECISION

You can build scaling plans that automate how groups of different resources respond to changes in demand and optimize availability, costs, or a balance of both.

- AUTOMATICALLY MAINTAIN PERFORMANCE

You maintain optimal application performance and availability, even when workloads are periodic, unpredictable, or continuously changing. AWS Auto Scaling continually monitors your applications to make sure that they are operating at your desired performance levels.

- PAY ONLY FOR WHAT YOU NEED

When demand drops, AWS Auto Scaling will automatically remove any excess resource capacity so you avoid overspending.

Azure Autoscale

- Scale by any metric

Autoscale is a built-in feature of Cloud Services, Mobile Services, Virtual Machines, and Websites that helps applications perform their best when demand changes.

- Quickly know when something’s wrong

As with Autoscale, you can set alerts based on just about any metric, such as CPU status or response time. You can even create alert for events—including when autoscale itself is triggered.

- Horizontal vs vertical scaling.

Horizontal is more flexible in a cloud situation as it allows you to run potentially thousands of VMs to handle load. In contrast, vertical scaling is different. It keeps the same number of VMs, but makes the VMs more ("up") or less ("down") powerful. Power is measured in memory, CPU speed, disk space, etc. Vertical scaling has more limitations. It's dependent on the availability of larger hardware, which quickly hits an upper limit and can vary by region. Vertical scaling also usually requires a VM to stop and restart.

Serverless Computing

- Serverless Computing is a new form of cloud based computing similar to VM’s and containers running on a cloud provider.
- The management of servers, scaling, and capacity planning are taken care by the underlying cloud provider.
- Application developers only need to focus on functionality and business logic.

AWS Lambda

- AWS Lambda natively supports a variety of languages including Node.js, Python, Java, and C# (.NET Core). Additional languages can be supported by spawning a child process from one of the supported languages which is allowed in the AWS Lambda sandbox. AWS sandbox isolation does not rely on any language constructs which allows this flexibility.
- Deployment requires creating and uploading a deployment package to bundle the code and dependencies together.
- It can serve as one-click triggers providing a lot of flexibility, triggers for everything from Amazon Kinesis for event streams to updates in DynamoDB or S3 or even be triggered from an Alexa skills app.

Azure Functions

- Azure supports a wider variety of languages compared to AWS. (like F#, PHP, Bash, and PowerShell.)
- Furthermore, their logic apps and flow are enabling non-developers to set up their own logic for business processes just like how Zapier can connect multiple business tools.
- Unlike AWS and Google, Azure provides more infrastructure around deployment. You can set up continuous build and deploy deployment from sources in Visual Studio Team Services, Bitbucket, and Github.
- Unlike AWS, more of the HTTP trigger functionality is built natively in Azure Functions without requiring the set up of a separate API gateway. Much of this has to do with Azure's logically grouping of functions under an App Service container. Azure supports a variety of other triggers. Such triggers could include Azure Blob Storage, Azure Event Hubs, and Queues.

Google Cloud Functions

- It is the most limiting in terms of languages as they only support Node.js.
- Part of this is not due to Google Cloud Functions but that Google has least amount of offerings in Google Cloud that could serve as triggers. AWS has Kinesis and Azure has EventHubs, but Google has no equivalent offering to serve as a trigger.
- Unlike AWS, Google is not pushing Cloud Functions front and center. Internal development at Google relies more on tools like Kubernetes (Borg) rather than Cloud Functions.
- Google allows up to 9 minutes of execution before the process is killed and also has a separate product *Cloud Function for Firebase*, which may be useful if you’re a mobile app startup already reliant on Firebase.
- Unlike AWS, Google integrated HTTP functionality directly in Cloud Functions without requiring the set up of a separate API gateway.

Serverless Comparison I

Feature	AWS Lambda	Google Cloud	Azure Functions
Scalability & availability	Automatic scaling (transparently)	Automatic scaling	Manual or metered scaling (App Service Plan), or sub-second automatic scaling (Consumption Plan)
Max # of functions	Unlimited functions	1000 functions per project	Unlimited functions
Concurrent executions	1000 parallel executions per account, per region (soft limit)	No limit	No limit
Max execution	300 sec (5 min)	540 seconds (9 minutes)	300 sec (5 min)
Supported languages	JavaScript, Java, C#, and Python	Only JavaScript	C#, JavaScript, F#, Python, Batch, PHP, PowerShell
Dependencies	Deployment Packages	npm package.json	Npm, NuGet
Deployments	Only ZIP upload (to Lambda or S3)	ZIP upload, Cloud Storage or Cloud Source Repositories	Visual Studio Team Services, OneDrive, Local Git repository, GitHub, Bitbucket, Dropbox, External repository
Environment variables	Yes	Not yet	App Settings and ConnectionStrings from App Services
Versioning	Versions and aliases	Cloud Source branch/tag	Cloud Source branch/tag

Serverless Comparison II

Feature	AWS Lambda	Google Cloud	Azure Functions
Event-driven	S3,SNS,SES, DynamoDB, Kinesis, CloudWatch, Cognito, API Gateway, Code Commit, etc.	Cloud Pub/Sub or Cloud Storage Object Change Notifications	Blob, EventHub, Generic WebHook, GitHub WebHook, Queue, Http, ServiceBus Queue, Service Bus Topic, Timer triggers
HTTP(S) invocation	API Gateway	HTTP trigger	HTTP trigger
Orchestration	AWS Step Functions	Not yet	Azure Logic Apps
Logging	CloudWatch Logs	Stackdriver Logging	App Services monitoring
Monitoring	CloudWatch & X-Ray	Stackdriver Monitoring	Application Insights
In-browser code editor	Yes	Only with Cloud Source Repositories	Functions environment, App Service editor
Granular IAM	IAM roles	Not yet	IAM roles
Pricing	1M requests for free, then $0.20/1M invocations, plus $0.00001667/GB-sec	1M requests for free, then $0.40/1M invocations, plus $0.00000231/GB-sec	1 million requests for free, then $0.20/1M invocations, plus $0.000016/GB-s