Tutorial 2: VPC Network, Load Balancer, and Cloud Architectures

Question Set:

Q1: What is a Virtual Private Cloud (VPC) network?

Answer:
A virtual private cloud (VPC) is an on-demand configurable pool of shared resources allocated within a public cloud environment, providing a certain level of isolation between the different organizations (denoted as users hereafter) using the resources. The isolation between one VPC user and all other users of the same cloud (other VPC users as well as other public cloud users) is achieved normally through the allocation of a private IP subnet and a virtual communication construct (such as a VLAN or a set of encrypted communication channels) per user. In a VPC, the previously described mechanism, providing isolation within the cloud, is accompanied by a VPN function (again, allocated per VPC user) that secures, by means of authentication and encryption, the remote access of the organization to its VPC resources. With the introduction of the described isolation levels, an organization using this service is in effect working on a 'virtually private' cloud (that is, as if the cloud infrastructure is not shared with other users), hence the name VPC. VPC is most commonly used in the context of cloud infrastructure as a service.

Q2: Please discuss the differences between VPCs in AWS and GCP?

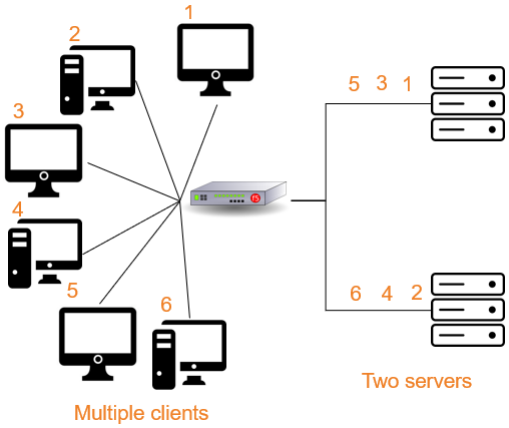
Answer:
In GCP, VPC is used as a global resource while VPC in AWS is regional and needs extra settings to communicate across VPCs, e.g. VPC peering. In GCP, subnets are confined to regions and the traffic can cross multiple zones transparently, while in AWS, the subnets are bonded to specific zones and need routing between multiple subnets for communications. Generally speaking, Google Cloud Platform VPCs are relatively flat with controls targeting the instance, whereas Amazon Web Services VPCs are hierarchical with multiple layers of control at the region, zone, subnet, and instance.

Q3: What is Load Balancing? Please discuss the learned LB algorithms and compare them.

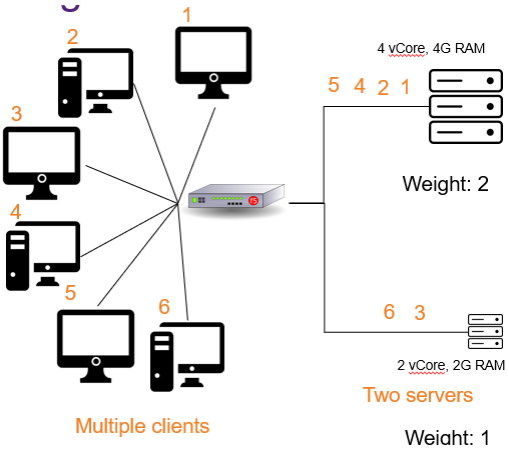
Answer:
In computing, load balancing refers to the process of distributing a set of tasks over a set of resources (computing units), with the aim of making their overall processing more efficient. Load balancing can optimize the response time and avoid unevenly overloading some compute nodes while other compute nodes are left idle.

We have introduced five Load Balancing algorithms: Round Robin, Weighted Round Robin, Least Connections, Weighted Least Connections, and Random.
Round-robin load balancing is one of the simplest methods for distributing client requests across a group of servers. Going down the list of servers in the group, the round-robin load balancer forwards a client request to each server in turn. When it reaches the end of the list, the load balancer loops back and goes down the list again (sends the next request to the first listed server, the one after that to the second server, and so on). **Round-robin load balancing** is suitable for some cases:

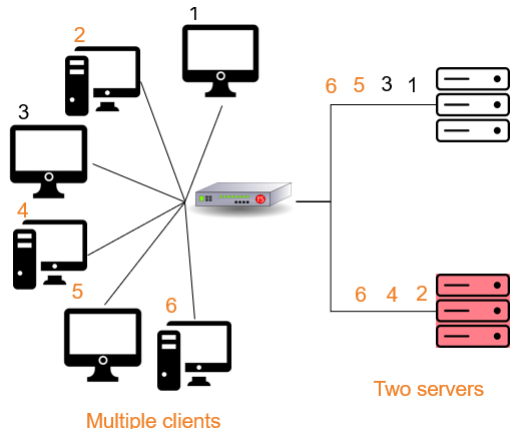
- Not identical hardware specifications between nodes
- Round-robin load balancing can result in the overloading of the imbalanced cluster.
- Round-robin is best for clusters consisting of servers with identical specs.



Weighted Round Robin load balancing is similar to the Round Robin (cyclic distribution). The node with the higher specs will be apportioned a greater number of requests. Set up the load balancer with assigned "weights" to each node according to hardware specs. Higher specs, higher weight.

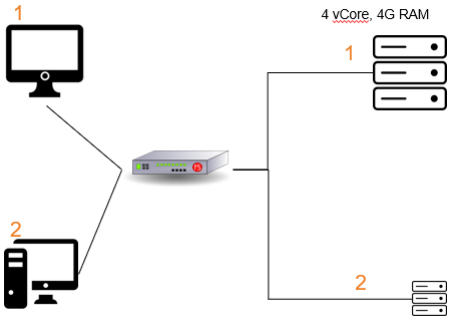


Least Connections algorithm considers the number of current connections each server has when load balancing. Less connection, higher priority for assignment.

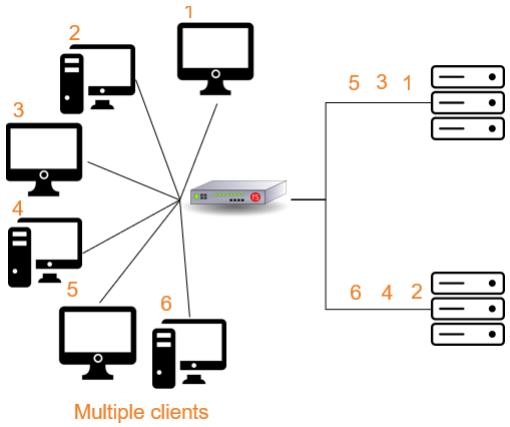


The Weighted Least Connections algorithm applies a "weight" component based on the computing capacities of each server. Similar to Weighted Round Robin, set up a weight for each server. When

directing an access request, a load balancer now considers two things: the weights of each server; the number of clients currently connected to each server.



Random algorithm, as its name implies, this algorithm matches clients and servers by random, i.e. using an underlying random number generator. In cases wherein the load balancer receives a large number of requests, a Random algorithm will be able to distribute the requests evenly to the nodes. Like Round Robin, the Random algorithm is suitable for clusters consisting of nodes with similar configurations (CPU, RAM, etc.).



Q4: Please discuss the Layer 4 and Layer 7 load balancers.

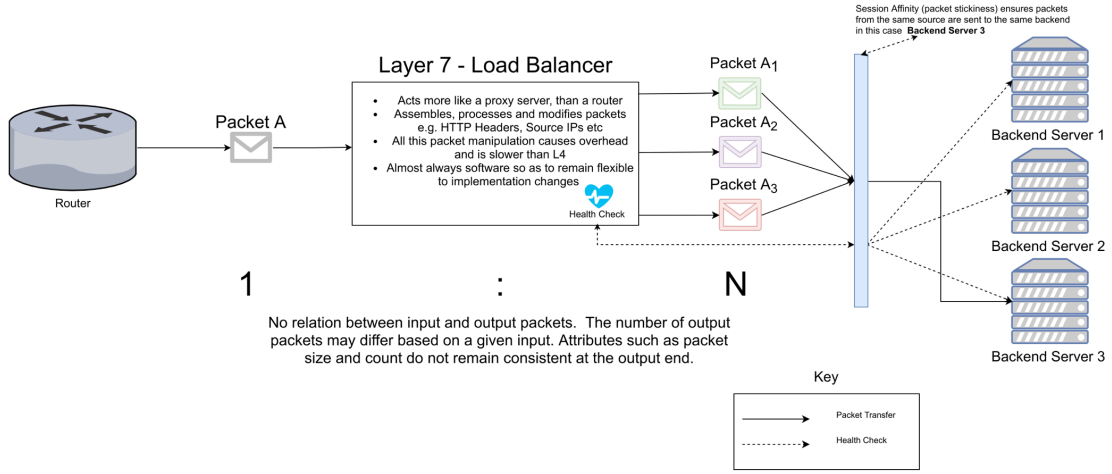
Answer:
Layer 7 load balancers operate at the highest level in the OSI model, the application layer (on the Internet, HTTP is the dominant protocol at this layer). Layer 7 load balancers base their routing decisions on various characteristics of the HTTP header and on the actual contents of the message, such as the URL, the type of data (text, video, graphics), or information in a cookie.

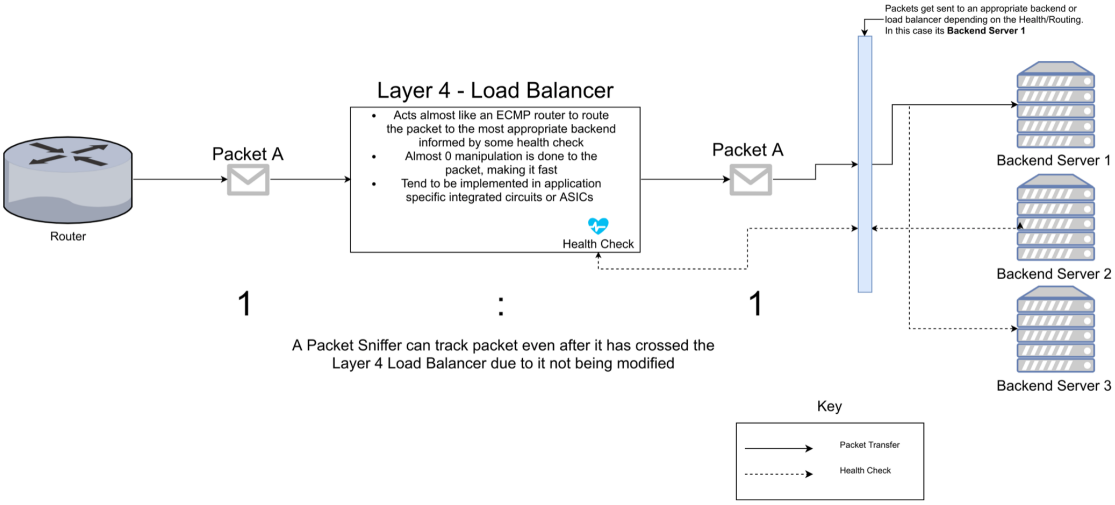
Taking into consideration so many more aspects of the information being transferred can make Layer 7 load balancing more expensive than Layer 4 in terms of time and required computing power, but it can nevertheless lead to greater overall efficiency. For instance, because a Layer 7 load balancer can determine what type of data (video, text, and so on) a client is requesting, you don't have to duplicate the same data on all of the load-balanced servers.

Modern general-purpose load balancers, such as NGINX Plus and the open source NGINX software, generally operate at Layer 7 and serve as full reverse proxies. Rather than manage traffic on a packet-by-packet basis like Layer 4 load balancers that use NAT, Layer 7 load balancing proxies

can read requests and responses in their entirety. They manage and manipulate traffic based on a full understanding of the transaction between the client and the application server.

Some load balancers can be configured to provide Layer 4 or Layer 7 load balancing, depending on the nature of the service. As mentioned previously, modern commodity hardware is generally powerful enough that the savings in computational cost from Layer 4 load balancing are not large enough to outweigh the benefits of greater flexibility and efficiency from Layer 7 load balancing.





	Layer 4 LB (TCP)	Layer 7 LB (HTTPs)
Layer	Transport Layer	Application Layer
Packet Manipulation	No	Yes
SSL Traffic	No	Yes

Logging & Monitoring	Not suitable	Yes
Implementation	Dedicated hardware	Typically software
Throughput Speed	Fast	Relatively lower

Supplementary video material: a detailed video introduction to AWS VPC [5]. Watch it if you are looking for more technical details of VPC.

References

[1]. Peter Mell, Timothy Grance, The NIST Definition of Cloud Computing, National Institute of Standards and Technology, September 2011.
[2]. Erl, Thomas, Ricardo Puttini, and Zaigham Mahmood. Cloud computing: concepts, technology & architecture. Pearson Education, 2013.
[3]. https://en.wikipedia.org/wiki/Virtual_private_cloud.
[4]. https://codeburst.io/vpc-networking-gcp-v-s-aws-77a80bc7cfe2.
[5]. 2019 AWS Summit video about AWS Networking fundamentals (VPC). https://youtu.be/hiKPPy584Mg
[6]. Singh, Navpreet, and Kanwalvir Singh Dhindsa. "Load Balancing in Cloud Computing Environment: A Comparative Study of Service Models and Scheduling Algorithms." *International Journal of Advanced Networking and Applications*8, no. 6 (2017): 3246.
[7]. TCP vs HTTP(S) Load Balancing. https://medium.com/martinomburajr/distributed-computing-tcp-vs-http-s-load-balancing-7b3e9efc6167
[8]. https://www.nginx.com/resources/glossary/layer-4-load-balancing/