✓ 100 XP

Summary

6 minutes

Here are some of the key points presented in this module about elasticity:

- VMs and other cloud resources rarely experience constant loads. Instead, they experience variable loads -- sometimes loads that vary by an order of magnitude or more over time.
- Sizing compute capacity to fit peak loads ensures quality of service (QoS) but results in increased costs and energy usage.
- Elasticity refers to the ability to add resources to handle higher loads and remove resources when the load decreases.
- Elasticity is achieved in the cloud by scaling resources such as VMs and databases.
- Scaling in and out (horizontal scaling) refers to increasing and decreasing the number of resources devoted to a task -- for example, increasing the number of VMs serving website users from 10 to 15.
- Scaling up and down (vertical scaling) refers to replacing existing resources with more or less powerful ones -- for example, replacing a web-server VM containing 2 cores and 4 GB of RAM with one containing 4 cores and 8 GB of RAM.
- Scaling resources to match demand keeps resource utilization relatively constant, lowers costs, and improves energy usage.
- Autoscaling allows scaling to occur based on rules or policies established by a cloud administrator. The rules or policies can be time-based, metrics-based, or both. An example of metrics-based autoscaling is bringing additional instances online when average CPU utilization reaches a predetermined threshold such as 70%.
- Time-based autoscaling, also known as scheduled autoscaling, is most appropriate when loads are cyclical and predictable.
- Metrics-based autoscaling can handle both predictable and unpredictable loads.
- Effective load balancing is crucial to implementing scalable cloud services.
- Load balancers use different kinds of algorithms to distribute load, including round-robin and hashed-based algorithms.
- Some load balancers attempt to dispatch requests more intelligently by using metrics such as request-execution time and CPU utilization at each node.
- Load balancers also increase availability by monitoring the health of back-end resources and recognizing when those resources aren't available.
- Because a single load balancer represents a single point of failure, load balancers are often deployed in pairs.
- Serverless computing offers benefits that include consumption-based pricing, automatic scalability, and reduced administrative costs

- One example of serverless computing is serverless functions, which let you upload code to the cloud and define when it executes.
- Another example is serverless workflows, which let you define business workflows (typically using graphical designers and without writing code) and specify when they execute.
- Serverless computing also extends to databases, which scale to meet the demand placed on them.

Explore other modules

AZ-400: Develop a Site Reliability Engineering (SRE) strategy
Manage cloud resources