Reading 4.3: Optimizing Solutions on AWS

What Is Availability?

The availability of a system is typically expressed as a percentage of uptime in a given year or as a number of nines. Below, you can see a list of the percentages of availability based on the downtime per year, as well as its notation in nines.

Availability (%)	Downtime (per year)
90% ("one nine")	36.53 days
99% ("two nines")	3.65 days
99.9% ("three nines")	8.77 hours
99.95% ("three and a half nines")	4.38 hours
99.99% ("four nines")	52.60 minutes
99.995% ("four and a half nines")	26.30 minutes
99.999% ("five nines")	5.26 minutes

To increase availability, you need redundancy. This typically means more infrastructure: more data centers, more servers, more databases, and more replication of data. You can imagine that adding more of this infrastructure means a higher cost. Customers want the application to always be available, but you need to draw a line where adding redundancy is no longer viable in terms of revenue.

Improve Application Availability

In the current application, there is only one EC2 instance used to host the application, the photos are served from Amazon Simple Storage Service (S3) and the structured data is stored in Amazon DynamoDB. That single EC2 instance is a single point of failure for the application.

Even if the database and S3 are highly available, customers have no way to connect if the single instance becomes unavailable. One way to solve this single point of failure issue is by adding one more server.

Use a Second Availability Zone

The physical location of that server is important. On top of having software issues at the operating system or application level, there can be a hardware issue. It could be in the physical server, the rack, the data center or even the Availability Zone hosting the virtual machine. An easy way to fix the physical location issue is by deploying a second EC2 instance in a different Availability Zone.

That would also solve issues with the operating system and the application. However, having more than one instance brings new challenges.

Manage Replication, Redirection, and High Availability

Create a Process for Replication
The first challenge is that you need to create a process to replicate the configuration files, software patches, and application itself across instances. The best method is to automate where you can.

Address Customer Redirection
The second challenge is how to let the clients, the computers sending requests to your server, know about the different servers. There are different tools that can be used here. The most common is using a Domain Name System (DNS) where the client uses one record which points to the IP address of all available servers. However, the time it takes to update that list of IP addresses and for the clients to become aware of such change, sometimes called propagation, is typically the reason why this method isn't always used.

Another option is to use a load balancer which takes care of health checks and distributing the load across each server. Being between the client and the server, the load balancer avoids propagation time issues. We discuss load balancers later.

Understand the Types of High Availability
The last challenge to address when having more than one server is the type of availability you need—either be an active-passive or an active-active system.

- *Active-Passive:* With an active-passive system, only one of the two instances is available at a time. One advantage of this method is that for stateful applications where data about the client's session is stored on the server, there won't be any issues as the customers are always sent to the same server where their session is stored.
- *Active-Active:* A disadvantage of active-passive and where an active-active system shines is scalability. By having both servers available, the second server can take some load for the application, thus allowing the entire system to take more load. However, if the application is stateful, there would be an issue if the customer's session isn't available on both servers. Stateless applications work better for active-active systems.

Resources

- External Site: High Availability and Scalability on AWS
- *External Site:* AWS: AWS Reliability Pillar: AWS Well-Architected Framework