How Cloud Deploys our Application Across the Globe? – Part 2

Deploying an app in multiple data centers cont'd.

We'll cover the following

Multi-regional deployments

When our application is deployed in just one region and it starts receiving traffic from around the globe, users located in other regions will experience a bit of network latency. The data will take some time to travel from the region where the service is deployed to the rest of the world, and this cannot be avoided.

If our service is latency sensitive like a massive, online multiplayer game, players located in other regions of the globe may experience a slight delay in in-game events as opposed to the players located in the region where the service is deployed. This gives the local players an edge and is not good from an end-user experience standpoint.

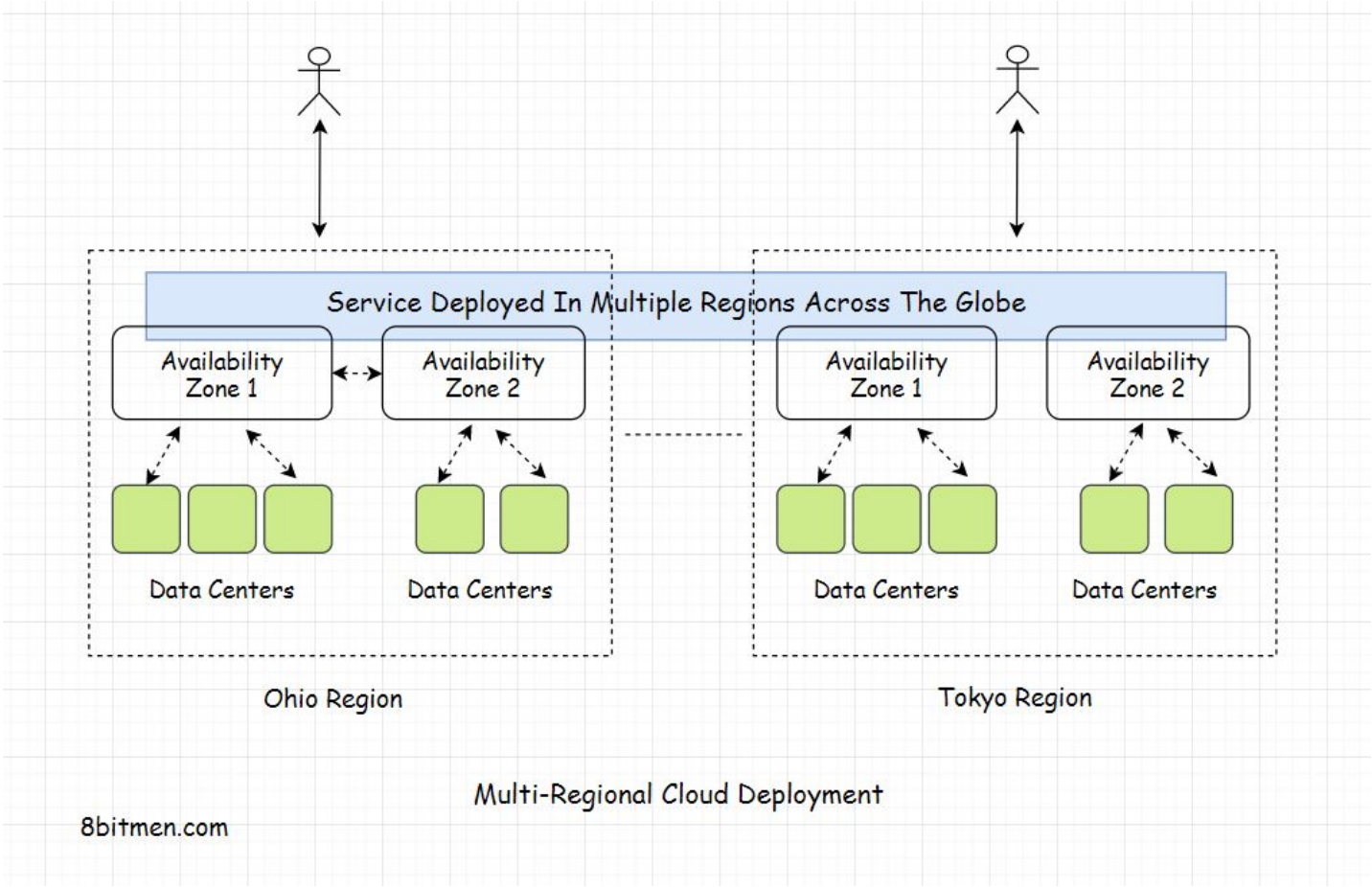
Here is a fun website called gcping.com. It sends requests to all the regions of *Google Cloud* and displays the latency of every region from your location. You will see that the region nearest to your location has the least latency.

The GCP region nearest to my location is *Mumbai*; it shows the latency as *45 ms*, and the latency for the *Belgium* region, which is, the max, is *468 ms*. Although we can place a *Content Delivery Network (CDN)* in between to serve static content quickly and, reduce the latency *CDNs* don't help much with the dynamic content.

Going ahead with multi-regional deployment in the latency-sensitive online gaming service scenario would be a good idea. It will reduce the application

latency; you'll go global, your service will have an extra layer of regional level

disaster safety, and so on. The key advantage of deploying our app on the cloud is that we can take our business global with dramatically less hassle than hosting our service ourselves.



Be it a business of moderate size deployed with the local city hosting service or a very small startup with a couple of programmers hosting their service from their basement.

Imagine the efforts it would require for them to put their servers on different locations across the globe. Is it even possible?

Furthermore, we experience some fundamental differences between the traditional computing infrastructures when self-hosting our service and the cloud computing infrastructure that enables the cloud to scale globally. There will be more on this in the next lesson.

Efforts involved in going multi-regional

When running a service on the cloud, the efforts required to go multi-regional largely depend on:

- The cloud service deployment model being used. For instance, if the service is

- The cloud service deployment model being used. For instance, if the service is completely deployed using the serverless approach, the efforts required for the multi-regional deployment would be minimal.
- The expectations of the business from their global deployment, for instance, the latency expectations and the architectural complexity of their service. It's generally a trade-off between the latency and the increase in complexity of the architecture due to the efforts put in to achieve a certain latency number across the regions.

Multi-regional deployments require:

- architectural changes
- configuration changes
- the pricing policy of the cloud providers to vary based on the geographic locations
- the service to stay in compliance with the local data laws
- configuring the primary and the secondary regions
- the data replication policies between the regions have to be written
- implementing geo-aware routing and multi-regional cluster monitoring
- making sure the infrastructure we want is available in the regions we want to deploy our service and so on.

All this stuff is not so trivial to pull off. Everything requires serious consideration before making a move. Let's unpack this further with the help of a simple example.

Single region to multi-regional deployment of an online multiplayer game

Taking that online multiplayer game example further, it was originally deployed in a single region. Now, we want to expand the deployment to multiple regions across the world.

To have a multi-regional deployment we are required to comply with the local data laws. Therefore, a couple of big changes that we have to make in our system are:

- We need to update our data persistence and log management flow.
- We need to rebuild our deployment pipeline.

Data persistence and log management

Originally, all the logs were being stored in a single region, but now we have to split the logs and store them region-wise to keep the data within that specific region. This means updating the log management module. Also, regarding storing the data of users, we have to tweak the flow that takes care of on-boarding new users, as we now have to keep their information, like the payment information for in-game purchases, in the region closest to them. We may also have to move all the existing data we have to the regions closest to the users.

Why are we doing this?

To reduce the latency and to comply with the data laws.

Since the data of the user stays in the region closest to others, the network latency to fetch that data would be minimal.

In compliance with the data law of some regions, we may also have to remove some fields in the user registration form. To deal with this, we may have to create separate onboarding pages for different regions and then route the users to region-specific pages something along the lines of how e-commerce websites route us to the local region-specific versions of their websites.

Let’s continue this discussion in the next lesson.