Picking the Right Cloud Provider for Your Application - Part 2

This lesson continues the discussion about picking the right cloud platform to host your app.

We'll cover the following

Here is a good way to be sure that you won't face any technology limitations on the road ahead. With this approach, you can also ascertain whether the solution that the cloud offers fits well with your business requirements.

Run a Proof of Concept (POC) before going all in

Once you zero in on a cloud vendor, the best way to acquire insight into the technology, the platform, and the pricing model is to run a *Proof of Concept (POC)* on the cloud platform.

Let the POC run for a while, and then do a stress test on the app. Check if you are happy with the *monitoring, analytics, logging*, and other services. There is no better way to gauge the issues and the risks that you might have to deal with in future than this.

When you run your app on the cloud, the pricing models aren't that straightforward and simple. For instance, if a vendor says you have *x* hours of instance time in a particular quota, that doesn't mean you'll get *x* instance hours, no matter what. Chances are you might not. Reason being, there are *request-response* data limits associated with the instance hour limit too.

If your *request-response* data limits get exhausted, you might not be able to use even *x/10* instance hours. Once you run a *POC*, you'll understand all these intricacies of the platform.

This is all for understanding your business use case and knowing if a certain platform is right for you.
It is time to move on to other factors.

Check out the smaller niche players in the market

If you have very specific needs, you can check out other smaller cloud vendors in the market that cater to those specific needs such as *serverless, static hosting*, and so on.

An upside of working with smaller providers is that they offer very competitive pricing models, features, tools, and customer support in comparison to the bigger players.

Vendor lock-in

I can't stress this enough, always look for an open-source solution first before you make up your mind to move forward with proprietary cloud technology. This reduces your vendor lock-in risk significantly.

When you are not locked in, if you don't like a service, you can move to another vendor. However, when using a proprietary service, you are in for some serious code refactoring.

Vendor lock-in is something that is unavoidable when using a vendor-managed proprietary technology. If our business faces any sort of issue with the proprietary technology in the future, be it a technical limitation or a billing issue, it's super hard to bail out because the code is tightly coupled with the proprietary tech.

To transition to a new cloud service, we may even have to write things from scratch, which can eat up quite a lot of business resources and time.

When we are locked-in, we have to rely on the vendor to roll out new features of the proprietary product. Since the code is closed source, we cannot write any custom features on our own. This strongly limits our ability to move fast.

Speaking of security, complying with the security protocols and regulations regarding the implementation of the proprietary product is, again, on the vendor.

On the contrary, when using open source, we have the freedom to write custom features and leverage other open industry frameworks, tools, and libraries that are continually patched by a dedicated global community.

We’ll continue this discussion in the next lesson.