A preliminary market research indicates that the 3 most important factors game developers look for in a cloud service provider are:

1. Developer Ergonomics
2. Risk Management and Demand Scaling
3. Pricing

It might be a good starting point to explore the market dynamics along these verticals while also modifying the list as we gain better understanding of the ecosystem.

Let's start with **Developer Ergonomics**:

- The evidence suggests that having **a uniform and standardized API** eases the cognitive load on developers, letting them focus on their core business.
- Cloudflare already offers the functionality as **Terraform**. However, **cascading the complexity of the configuration, and letting the users decide on level of customization by offering staggered levels of default optimized values,** based on use cases, can reduce the release period for a product.
- To increase the chances of success, being involved in efforts to **educate and encourage the adoption of Terraform and cascaded configurations**, with possible **figures stating expected reduction in deployment time and performance comparison** to similar services being offered in the market could be helpful before release.
- To measure the success of the product, a limited beta version can be offered free of cost while **analytical as well as subjective feedback** be taken from the users of the service to decide on its acceptance among the users.
- A possible risk of failure could stem from the inefficacy of the proposed cascaded configuration and default optimization, possibly leading **to creation of a separate market catering to the specific demands at lower price points** and across multiple vendors. Though, I think the product could still retain its appeal for the indie developers.

Risk Management and Demand Scaling:

- **Mitigating DDoS / botnet / zero-day attacks** while ensuring scalable system performance has been a consistent priority for game developers and likely to be in the foreseeable future as well.
- Cloudflare has done a pretty good job at risk management. However, some of the risk management practices, for instance, **monitoring for excessive cache misses** to detect a side channel attack**, limiting connection requests per isolate** and having a **50 micro-second hard timeout** limit may not scale well in future, especially as bandwidth keeps increasing but computational power nears saturation in **a post Moore's law** world. The scaling strategies need to be further investigated and innovated upon if we are to maintain competency in the market in future. I think **web assembly** is a great effort and could possibly do away with some of the above limitations by providing **alternate mechanisms to deal with misbehaving isolates**.
- To increase the chances of success**, design guidelines, empirically observed best practices and possible caveats for deployment of services on an isolate model can be published** before a public offering.
- Until web assembly kicks in and the gap between isolates and containers reduce, the current performance scaling efficacy needs to be constantly monitored and feedback from users

assessed to identify the possibly bottlenecks, as and when they arise. **A one to one performance comparison with similar applications running on containerized or bare metal servers could provide for good initial measure of success**.

- The most obvious risk of failure stems from the limitation that current isolates impose on user applications, and **the possibility that modifying them could lead to a loss in performance the isolates provide**.

Pricing:

- Sensitivity to pricing varies across the spectrum and indie game developers appear to be the ones most sensitive to pricing. The pricing market is a complex adaptive system **and requires constantly evolving the pricing strategy to address the morphing values** associated with market attributes.
- The edge computing model of Cloudflare with costs associated with the number of requests and not processor time, works to the advantage customers and Cloudflare, in keeping the system transparent and accountable. A possible addition to the system could be **allowing customers to choose a different metric for billing, letting them do an apple to apple comparison** with the other alternatives available in the market. It would also have the advantage of **exposing flaws in Cloudflare's current pricing model or the alternatives**.
- Before a public release, **an option to switch to an alternate pricing model can be offered to a limited number of current customers** and analyze their behavior and feedback **to fine the model parameters.**
- In this case, **the number of switches to alternate pricing models and new signups** for the same can be a reasonable parameter to judge the potential success of the project.
- The **cost of technical infrastructure** to implement the alternate pricing model may be considerable and in case the market does not really care about alternate pricing strategies or micro-managing savings, the venture could potentially be a failure.

Besides the above, exploring potential scaling issues, use case scenarios, and deployment mechanisms for

- AI / ML based applications
- Computing intensive applications
- True or pseudo multithreaded applications
- Devoting more resources to Web Assembly and possibly bridging the gap between isolates and containers / bare metals

could also be potential endeavors.