

Bloom.Host -

Bloom.Host is based in Ashburn, Virginia and has locations all around the world for game hosting. For dedicated servers, they manage on premise systems based in this location and LA, California. They provide support via tickets online and in their Discord which is the much more preferred method. Their hardware works well and none of our clients have had uptime issues.

Pros

- Whenever there is a major fault with the system, they will personally setup a replica system and once you've moved the data over, they remove the old system from your account. They have spares on hand to instantly get you back into production and they don't faff around like other providers giving excuses wasting more time. This is the biggest pro by far and has earned them the extra star.
- They are the cheapest on the market in terms of reliability and performance for both of their locations. They definitely could charge more on their hardware which is an extreme benefit for you as the end customer.
- You can ask them to internally link the servers for a private networking configuration when purchasing two or more dedicated servers.

Cons

- We've tried to strike a partnership deal with the team but unfortunately this has failed multiple times. This is due to being the cheapest they can be and they actually make a loss for a month or so until you've brought the system. Don't attempt this, it won't happen.

Known Issues

- Unless we've been unlucky with some of our clients, hardware faults a little more common than we'd like from a provider like OVH. This isn't their fault as it's been RAM faults, CPU faults or some weird networking issue we've come across twice, just keep this in mind.
- Their firewall is a tad aggressive towards downloading files and will grind them to a halt due to their Anti-DDoS protection systems and to whitelist the

address the data is coming from, a support ticket is needing to be opened.

Revision #3

Created 28 January 2024 02:57:48 by JasmeowTheCat

Updated 16 October 2024 22:50:36 by JasmeowTheCat