

Choose a GPU cloud for your workload

Use a repeatable benchmark and a clear acceptance rule.

Updated October 29, 2026

Start with the result you need. For training, record the model, precision, batch size, data path and checkpoint size. For inference, record request sizes, concurrency and the latency you must meet. Do not use a vendor's peak throughput as your application benchmark.

Define a fair trial

Use the same container, model revision and input set for each candidate. Pin library versions. Save the command and environment description with the result. Measure data loading, model loading and useful work separately. Include retries and failed jobs in your cost log.

Check device availability in the actual region you need. A public device catalogue does not reserve capacity. Ask for the quota and the allocation terms before you plan a launch.

Choose an acceptance rule

Write a pass condition before the trial. Include correctness, runtime, maximum spend and recovery from interruption. Use a small representative workload first. Expand only after the run is repeatable.

Compare total cost per accepted result. Include engineering work to move images, data and monitoring. SkyPilot can help manage jobs across infrastructure; it does not remove the need to test your workload. Kubernetes GPU scheduling depends on device plugins and declared GPU resources. Confirm that the service exposes the controls your deployment needs.

Keep a record of rejected candidates and the failed condition. This helps when prices or capacity change.

Sources

- [SkyPilot documentation](#) — fetched 2026-10-29.
- [Kubernetes GPU scheduling](#) — fetched 2026-10-29.

Frequently asked questions

Is this a vendor quote?

No. Confirm current terms and capacity with the provider before you rent.