Episode Book

EPISODE #5830

Serving Your Own Fine-Tuned Model in the Cloud

PUBLISHED	10 October 2026 at 10:23 UTC
DURATION	24:20
LISTEN	myweirdprompts.com/episode/serving-fine-tuned-models-cloud/
HOSTS	Corn, Herman, Hilbert
SCRIPT MODEL	deepseek-flash
VOICES	chatterbox-regular (chatterbox-local)
PIPELINE	V5.2
SOURCES	14 cited by the research (listed at the end)
BOOK GENERATED	19 October 2026 at 17:40 UTC

SYNOPSIS

You've fine-tuned an open-weight model like DeepSeek and now you need to serve it behind an API. This episode walks through the two main patterns — dedicated GPU hosting with weights loaded on an always-on instance, and serverless GPU inference where you pay for compute consumed — and untangles the three very different things "serverless GPU" actually means. We cover which platforms let you bring your own weights (RunPod, SageMaker, Modal, Baseten, Replicate, Fireworks, Cerebrium), what the LoRA hosting shift means for adapter-based fine-tunes, and why cold starts are a bandwidth problem, not a compute problem. Most importantly, we work through the utilization math that determines whether serverless is genuinely cheaper or just a way to end up back on a dedicated box.

TRANSCRIPT



Corn

You get the thing working on your own machine, the loss curve looks right, you run a few prompts through it, and then the actual question arrives, which is how anyone else is supposed to talk to it.



Herman

That's the moment the hobby ends and the invoice begins.



Corn

And that's what Daniel's asking about. He's put it in writing this time. You've fine-tuned an open-weight model, maybe something out of the DeepSeek family, and now you want to serve it in the cloud behind an API. He's naming two patterns. Dedicated GPU hosting, where the weights sit loaded on an always-on instance. And serverless GPU inference, where you deploy behind an endpoint and pay for compute consumed instead of holding a card warm. Which platforms actually support your own weights, how bad the cold starts really are, and whether serverless has become a genuine production option or just a cheaper-looking way to end up back on a dedicated box.



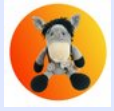
Herman

That last part is the honest question. Everything before it is just the map.



Corn

Start with the map, then. Because the first thing to untangle is that serverless GPU is not one thing.



Herman

It's three, and Threat Frontier made exactly this point in September. Category one is a token API. You pay per million tokens, you never see a GPU, and you can't deploy a fine-tune unless the provider happens to support your adapter. Category two is managed model deployment. You bring a checkpoint or a container, the platform autoscales, it scales to zero, you pay for compute time. Category three is raw GPU rental. You rent by the hour or the second and you run your own stack on it.



Corn

And the expensive mistake lives in that taxonomy.



Herman

Constantly. Somebody fine-tunes a model, gets excited, provisions a GPU endpoint, and the base model they fine-tuned was already being served by a hosted API for a fraction of what they're now paying. For a DeepSeek fine-tune specifically, the first question isn't which platform. It's whether your base model is already on a token API somewhere, and whether the fine-tune even needs to move.



Corn

So somebody buying category two for a workload category one would serve is the most common way this goes wrong.



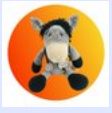
Herman

It's not close. And the giveaway is that they never asked the first question, because the fine-tune felt like the achievement. The deployment is the achievement. The fine-tune is just the artifact.



Corn

Right. Who actually lets you bring your own weights, then.



Herman

On the dedicated side, RunPod Pods are the least lock-in you'll find. Raw Docker, no enforced packaging format, single-tenant instances, and you can pick Community or Secure Cloud. An H100 PCIe on Secure runs two dollars eighty-nine an hour. An A100 SXM is a dollar fifty-nine. Nothing is stopping you from putting vLLM in a container and shipping it.



Corn

And the closer you get to a hyperscaler, the more the abstraction starts charging rent.



Herman

SageMaker is the AWS path for custom weights. You bring a Docker container running vLLM or SGLang or TensorRT-LLM, you control the instance type, and you get real-time endpoints with scale-to-zero. AWS's own decision framework says choose SageMaker over Bedrock when you're deploying fine-tuned, proprietary or open-source models, or when you're using your own container.



Corn

Which is a sentence Bedrock's marketing team probably doesn't have on a mug.



Herman

Bedrock's standard catalog is off-the-shelf models. The bridge is Custom Model Import, generally available since October of twenty twenty-four. You can bring compatible weights — Mistral, Mixtral, Flan, Llama two through three point three, Mllama, GPTBigCode, Qwen two, 2.5 and 3, GPT-OSS. But it excludes Batch inference and CloudFormation, which for some shops is the whole reason they'd have used it.



Corn

Then there's Modal, which is a strange shape, because it's both.



Herman

Modal is general-purpose serverless compute with GPU support, and you can absolutely keep containers warm on it. Python decorators, no separate artifact to build, per-second billing, scale to zero. Thirty dollars a month in free compute on the Starter plan, two hundred fifty on Team. H100 SXM5 at about a tenth of a cent per second, which lands near three ninety-five an hour. A100 eighty gigabyte at two fifty an hour. And the per-second billing is the thing that makes it interesting for the sporadic case, because a job that runs eleven seconds bills for eleven seconds.



Corn

Unless you pick a region.



Herman

Region selection costs between one point one five and one point seven five times base, and non-preemptible execution costs three times. So the headline rate is the floor, not the price.



Corn

RunPod again on the serverless side.



Herman

Flex workers scale to zero. Active workers run twenty-four seven at a lower per-GPU rate, which is the tell that they know exactly what they're selling. H100 at four seventy-nine an hour, A100 at two seventy-two, B200 at eight sixty-four. FlashBoot is their cold-start play, and they claim forty-eight percent of serverless cold starts complete in under two hundred milliseconds.



Corn

I'd want to know what model.



Herman

Everyone would. That number is doing a lot of quiet work in that sentence.



Corn

Baseten.



Herman

Dedicated single-tenant deployments through their Truss format, and it's aimed upmarket — compliance, mission-critical serving, that world. Per-minute billing with true scale-to-zero. H100 at about ten point eight cents a minute, roughly six fifty an hour. B200 at sixteen point six cents a minute, near ten dollars an hour. And Baseten raised a billion and a half at up to a thirteen billion valuation over the summer, after being worth five in January, so whatever they're doing is being funded generously.



Corn

Replicate.



Herman

Cog container format, about nine thousand four hundred stars on the repo. Public models bill only active time, so cold starts are free to the user. Private deployments bill setup, idle and active separately, which is where people get surprised. H100 at five forty-nine an hour.



Corn

And the LoRA story, because that's the part that changed.



Herman

Fireworks does dedicated-only LoRA. Either a live merge with one adapter and no added overhead, or multi-LoRA addons. H100 or H200 at eight an hour, B200 at thirteen, B300 at fifteen, billed per minute. Together AI discontinued serverless LoRA and multi-LoRA — adapters now run on dedicated endpoints, up to sixteen per endpoint, five if you're on a small Qwen three point five. And andrew dot ooo's read on this in October was blunt: per-token serverless hosting for LoRA adapters is mostly gone this year.



Corn

Gone where, exactly?



Herman

Gone as a cheap default. The adapters didn't disappear. The assumption that you could host an adapter per-token and pay pennies did. The choice now is which GPU platform serves your adapters, which is a category-two question wearing a category-one costume.



Corn

Cerebrium fits in there too.



Herman

Serverless with CPU and GPU memory checkpointing. SOC two Type Two as of July this year. They're the ones who've published the most aggressive cold-start numbers, which we should get to, because that's where the actual engineering is.



Corn

Here's the thing I keep circling, though. Every price you just said is a rate. None of them tell you what a month costs.



Herman

Because the answer depends entirely on one number nobody knows before launch.



Corn

Utilization.



Herman

Duty cycle. How many hours out of the month the thing is actually doing something. And the reason that dominates everything is simple. An idle GPU bills like a busy one. There's no version of a dedicated instance where you only pay when a request arrives. You're renting the possibility of a request.



Corn

Which is exactly the problem serverless claims to solve.



Herman

And does solve, below a certain threshold. The dreaming dot press analysis in August put serverless per-second rates at roughly one and a half to three times a cheap on-demand hourly rate, and worked the break-even duty cycle out to somewhere between thirty and sixty percent. The formula is just dedicated hourly rate divided by serverless effective hourly rate.



Corn

Give me the example, because that's the bit that makes it click.



Herman

An H100 busy six hours a day. On a dedicated instance at two fifty an hour, that's about eighteen hundred dollars a month. Same workload on Modal serverless — about seven hundred eleven dollars. Serverless wins by more than half. Now push it to sixteen busy hours a day. Dedicated stays at eighteen hundred, because the rate doesn't move. Serverless reaches about eighteen ninety-six. The lines cross and they cross hard.



Corn

Six hours a day is not a hobbyist number, either.



Herman

It's a healthy small app. Which is why I don't buy the framing that serverless is only for toy workloads. Six hours a day spread across a business day is a real product with real users.



Corn

And NotCheapAI ran the same math per-GPU.



Herman

At low volume, per-second platforms cost tens of dollars a month, while an always-on A100 pod at a dollar fifty-nine an hour is eleven hundred sixty dollars and seventy cents. Break-even utilization for Modal GPU-only against an A100 pod came out at sixty-three point six percent. With CPU and memory included it drops to fifty-four. RunPod Serverless against a pod was fifty-eight point five. And once you're running two H100s at eighty percent, the rented PCIe pods win outright — forty-two nineteen a month against forty-six twelve on Modal and fifty-five ninety-four on RunPod Serverless.



Corn

So the line moves with the shape of the fleet.



Herman

The pod-versus-serverless line sits between fifty-eight and seventy-three percent utilization, depending on what you're counting. And what you're counting is where most of these comparisons quietly cheat.



Corn

Counting CPU and memory is the honest version.



Herman

It's the honest version because a GPU endpoint still needs a CPU, still needs RAM, still needs a process holding the socket open. If you're comparing a GPU-only serverless rate against a full dedicated instance, you've rigged the comparison. Either count both or count neither.



Corn

Now the piece that actually decides it for most people. Cold starts.



Herman

And the framing people get wrong is that a cold start is a compute problem. It isn't. It's bandwidth. When a model comes back from zero, the container initializes in a second or two and then sits there for a minute and a half while a hundred and forty gigabytes of weights get dragged into VRAM.



Corn

A seventy billion parameter model in fp16.



Herman

A hundred forty gigabytes. Which is already too big for a single eighty gigabyte H100, so you're tensor-parallel across at least two cards before you've served a single token. Anyscale measured a naive Hugging Face load of Llama two seventy billion taking up to ten minutes before they optimized it. Ten minutes to answer the first request.



Corn

That's not a cold start, that's a lunch break.



Herman

Modal's the one with the headline here. They got a vLLM cold start down from four hundred sixty seconds to about seventy using GPU memory snapshots — capturing the post-warmup GPU and CPU state and restoring it instead of rebuilding it. Six point five times faster, and they claim no steady-state throughput loss, which is the important half of that sentence, because a snapshot that costs you throughput in steady state is a bad trade.



Corn

Seventy seconds is still a lot of seconds.



Herman

It's still a lot of seconds. But it changes the category. A hundred twenty seconds is long enough that you build your product around avoiding it. Seventy is long enough that some users notice and most don't.



Corn

Cerebrium published the counter-benchmark.



Herman

They ran a hundred cold-start requests per workload over twenty-four hours across six workloads. Snapshots cut cold starts seventy-one percent on average against their own no-snapshot baseline, up to eighty-eight percent on vLLM. They claim to be eighty-five percent faster than Baseten's cached cold-start path on average, up to ninety-four percent on vLLM, and about twenty-one percent lower average restore time than Modal with twenty-seven percent lower worst case.



Corn

And the concrete numbers underneath.



Herman

A nine gigabyte container on a g5 point twelve x large had a full vLLM cold start of about fifty seconds. Restoring from a nine gigabyte checkpoint took two point two five seconds from S3 and nine seconds from local NVMe. And then the number that should end the conversation — restoring DeepSeek V4 FP8 with vLLM would be a six hundred forty gigabyte checkpoint.



Corn

Hold on.



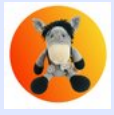
Herman

Six hundred forty gigabytes.



Corn

That's the model you'd actually fine-tune if you were following Daniel's example.



Herman

That's the point. Everything we just said about snapshots and sub-second restores is about a nine gigabyte container. Nobody is snapshotting six hundred forty gigabytes into a warm state in two seconds. The mechanics don't scale with the model.



Corn

So "serverless is viable now" is true at a size, and false above it.



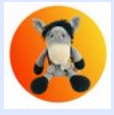
Herman

That's the honest version of the claim, and you almost never hear it stated that way. Threat Frontier put it plainly — a model in the hundreds of gigabytes is not a scale-to-zero candidate at all, no platform cleverness makes that instantaneous. Which is a hard wall, not a slow ramp.



Corn

There's a third camp on loading, though. The streaming loaders.



Herman

CoreWeave's Tensorizer does zero-copy direct-to-GPU, and NVIDIA's Run:ai Model Streamer reports about eighty gigabits sustained, several times the default loader. That attacks the biggest single stage directly — instead of loading to host memory and then copying to the card, you stream straight into VRAM. If the bottleneck is bandwidth, the fix is a faster pipe, not a faster GPU.



Corn

Which is the sentence the whole episode turns on.



Herman

dreaming dot press said it best — compute is not the bottleneck for coming back from zero. Bandwidth is. The frontier isn't a cheaper GPU, it's a faster way to fill it.



Corn

Should we say the obvious thing about those benchmarks?



Herman

That they're competitive. Cerebrium is claiming to beat Modal on worst-case restore while Modal is claiming four sixty to seventy. Baseten's snapshotting is referenced in their own marketing, and Cerebrium says it couldn't use it as a generally available, self-serve feature and benchmarked against their cached path instead. Two vendors telling two different stories about the same feature.



Corn

Which means you can't take any of these numbers as neutral.



Herman

You can't. The only reliable test is your own model, your own workload, your own region, measured. A cold-start number without the model size attached is marketing.



Corn

Eric Ma's benchmark is at least the right shape.



Herman

Ollama versus vLLM versus SGLang on Modal. vLLM generated sixty percent more tokens per second and restored from cold start significantly faster with GPU snapshots. The useful part is that it's apples to apples on the same platform — it tells you the engine matters as much as the platform, which people forget when they're comparing vendors.



Corn

There's a trap in the middle of all this that I want to name.



Herman

The keep-warm trap. Set a minimum replica count above zero for latency and you've bought a reserved GPU with extra steps. Threat Frontier's line is exactly that, and they add the bit that stings — budget it as one. You're now paying hourly at a serverless premium rate, which is the worst of both columns.



Corn

And the init-time billing catches the small jobs.



Herman

RunPod Serverless bills init time, and so do Replicate and Baseten private deployments. A thirty second cold boot on a ten second job triples that call's cost. It's not a bug, it's just the meter running while the weights load.



Corn

So a bursty workload that never keeps anything warm is cheap.



Herman

And a workload that keeps one replica warm for latency is expensive, because you're paying the serverless premium on hours you'd have bought cheaper as a pod. The min-replica setting is the single line of config that decides which world you're in, and it's usually set by whoever's on call that week.



Corn

The packaging choice is the other one that follows you.



Herman

Truss on Baseten, Cog on Replicate, raw Docker on RunPod, Python decorators on Modal. dreaming dot press called it — the choice that follows you longest is the packaging abstraction, not the price. Prices change quarterly. The artifact format is what you have to rebuild when you move.



Corn

And yet nobody puts it on the evaluation spreadsheet.



Herman

Because it costs nothing today and everything in eighteen months.



Corn

What about the fine-tune itself, though. If I've trained a DeepSeek variant, what does it actually take to serve it?



Herman

Weights and an inference engine. vLLM or SGLang in a container, the checkpoint mounted or baked in, and an endpoint in front of it. That part is identical on a pod and on a serverless platform. The difference is entirely in who holds the idle GPU, and the answer to that comes down to the utilization number we keep coming back to.



Corn

Which is a forecast.



Herman

It's a forecast, and it's the one nobody makes honestly, because the whole reason you're launching is that you don't know yet.



Corn

Here's what I'd say to someone at that stage. If it's a small app with sporadic usage, serverless is not a compromise — it's just correct. You'd be insane to rent an H100 to serve forty requests a day. And if it's regularly used, you cross the line fast, and the crossing is somewhere around half utilization, which sounds like a lot until you plot it against a working day.



Herman

Sixteen hours a day is not a stretch for anything with users in two time zones.



Corn

Which is most things with users at all.



Hilbert

How do you know a card's been idle more than twenty minutes?



Corn

...You can hear it.



Hilbert

Fans spin up different. Softer at first, then it catches. I had a cousin ran a small hosting outfit, and he could call it from the doorway. Twenty minutes, twenty-five, he'd tell you before he looked at anything. He kept a log of it. Every card, every day. Temperature at idle, time since last job, what the fan did when it came back up. Pages of it. Nobody asked him to.



Corn

Why keep that.



Hilbert

Because if you're paying for the electricity anyway, you want to know what the electricity's doing. He ran what he called the idle dance. Little inference jobs, looping, all night, nothing anybody needed answers to. Just enough to keep the silicon at temperature so a real job didn't land on a cold card.



Herman

That's the keep-warm problem with the lights on.



Hilbert

That's the keep-warm problem with a power bill. The rack was never cold. That was the point of it. And the cost of that never showed up on anybody's per-second line, because there was no per-second line. It was a meter on the wall.



Corn

How much of his capacity went to the dance?



Hilbert

Third of it, near enough. Maybe more in the winter. He turned down a customer over it, fellow wanted to run overnight batches, big spikes, nothing in between. My cousin told him he was too spiky. Said it flat, like it was a diagnosis. Fellow went off to one of the token services and paid a tenth of what he'd have paid for the rack. My cousin knew that. He didn't want to lose the account, so he lost the account anyway, just slower.



Herman

That's the category-one mistake running backwards.



Hilbert

He had a man walked the aisle. Twice a shift, ear to the cabinet, listening. That was the whole job. He could hear a fan going before anything on the dashboard moved. Called one three days out. They pulled the card, and it was on its way.



Herman

Three days.



Hilbert

I interviewed for that job. Didn't get it. Couldn't tell an A100 from an H100 by ear. He said it like it was a shame, but it wasn't really. He just didn't want family on the aisle.



Corn

And the fan logging.



Hilbert

Went out of business in ninety-one. Kept too many cards warm for too few customers. The last page in the log is a Tuesday.



Corn

The dance is a real cost and it never appears on the invoice.



Herman

Which is the honest critique of every per-second comparison we just ran. None of them price the thing Hilbert's cousin was actually paying for. The warmth tax.



Corn

And the token service the spiky customer went to charged him a tenth.



Herman

Because it was serving somebody else's base model on somebody else's utilization curve. He got to free-ride on a rack that was already warm.



Corn

So the break-even is utilization, and utilization is a forecast.



Herman

And the number that decides it is the min-replica count. Set it to one and you've bought a reserved GPU with extra steps at a premium rate. Leave it at zero and you accept the cold start, which for a large checkpoint might be minutes.



Corn

The vendor numbers are contested, too. Cerebrium's benchmark says one thing, Modal's blog says another, and Baseten's snapshotting is described two different ways by two different parties. None of that is lying, exactly. It's just everyone measuring the workload that flatters them.



Herman

Which is why the only cold-start figure worth trusting is the one you took yourself, on your model, in your region.



Corn

And the frontier isn't cheaper cards. It's weight-loading speed. Snapshots, streaming loaders, faster pipes into VRAM. As those improve, the size threshold where scale-to-zero stops being viable moves upward — a model that's too big today might be a perfectly good scale-to-zero candidate in two years.



Herman

Six hundred forty gigabytes is still six hundred forty gigabytes, though. The pipe gets wider. The checkpoint doesn't get smaller.



Corn

That's the thing to watch, then — whether bandwidth outruns model size.



Herman

My money's on neither winning, and the line just sitting somewhere new every year.



Corn

That's the note to end on. If you're deploying a custom fine-tune, the first question is whether a hosted API already serves your base model, because if it does, you may be about to rent a GPU for a problem you don't have. If it doesn't, the second question is your expected duty cycle, and the honest answer to that is usually "I don't know yet," which is itself an argument for starting on serverless and watching the meter.



Herman

And mind the keep-warm setting. It's one line of config and it quietly moves you from one business model to the other.



Corn

Thanks to Hilbert Flumingtop, our producer, at the desk as always.



Herman

If you want more of this, try episode forty-eight, Renting vs. Building; episode twenty-four sixty-four, Batch APIs; and episode twenty-one seventy-seven, Skip Fine-Tuning. This has been My Weird Prompts, the human-AI collaboration podcast.



Corn

If you've got a deployment question of your own, send us a prompt on Telegram at t dot me slash MWP listener bot. We'll be back soon.

SOURCES

What the research for this episode read before the script was written. Primary sources first.

1. **Modal pricing page [primary]**
   https://modal.com/pricing

2. **RunPod pricing page (updated 2026-07-27) [primary]**
   https://www.runpod.io/pricing

3. **Cerebrium, Reducing GPU Cold Starts with Memory Snapshots (2026-07-01) [primary]**
   https://cerebrium.ai/blog/reducing-gpu-cold-starts-with-memory-snapshots-restoring-cuda-workloads-in-second

4. **AWS, Amazon Bedrock or Amazon SageMaker AI? [primary]**
   https://docs.aws.amazon.com/decision-guides/latest/decision-guides/bedrock-or-sagemaker.html

5. **AWS re:Post decision framework [primary]**
   https://repost.aws/articles/ARD9QfIv_ZSoCxe-7NqGTAFQ/when-to-choose-amazon-sagemaker-ai-inference-over-amazon-bedrock-a-decision-framework

6. **AWS, SageMaker AI serverless model customization now supports full fine-tuning (2026-08-03) [primary]**
   https://aws.amazon.com/about-aws/whats-new/2026/08/amazon-sagemaker-fft/

7. **NotCheapAI, Modal vs RunPod Serverless vs Replicate: Real AI Inference Hosting Cost in 2026 (rates as of Sep 2026)**
   https://notcheapai.com/guides/modal-vs-runpod-vs-replicate-inference-cost-2026/

8. **andrew.ooo, Best Inference Providers for LoRA and Fine-Tuned Models 2026 (2026-10-07)**
   https://andrew.ooo/answers/best-inference-providers-for-lora-fine-tuned-models-2026/

9. **dreaming.press, Where to Deploy a Custom Model in 2026 (2026-06-22, updated 2026-08-28)**
   https://dreaming.press/posts/2026-06-22-modal-vs-replicate-vs-runpod-vs-baseten.html

10. **Threat Frontier, Serverless GPU Inference Compared (2026-09-12, updated 2026-09-24)**
    https://threatfrontier.com/articles/serverless-gpu-inference-compared-baseten-modal-runpod-together-fireworks

11. **dreaming.press, Scale to Zero for LLM Inference: Why Cold Starts Are a Weight-Loading Problem (2026-06-27)**
    https://dreaming.press/posts/2026-06-27-scale-to-zero-llm-inference-gpu-cold-starts.html

12. **dreaming.press, Serverless GPU vs Dedicated Instances (2026-08-06)**
    https://dreaming.press/posts/serverless-gpu-vs-dedicated-when-per-second-billing-wins.html

13. **Barking Iguana, Bedrock Custom Model Import supported architectures**
    https://barkingiguana.com/writing/importing-custom-weights-into-bedrock/

14. **Eric J. Ma, I Benchmarked Inference Engines on Serverless GPUs: vLLM Won on Everything (2026-07-23)**

https://dspn.substack.com/p/i-benchmarked-inference-engines-on