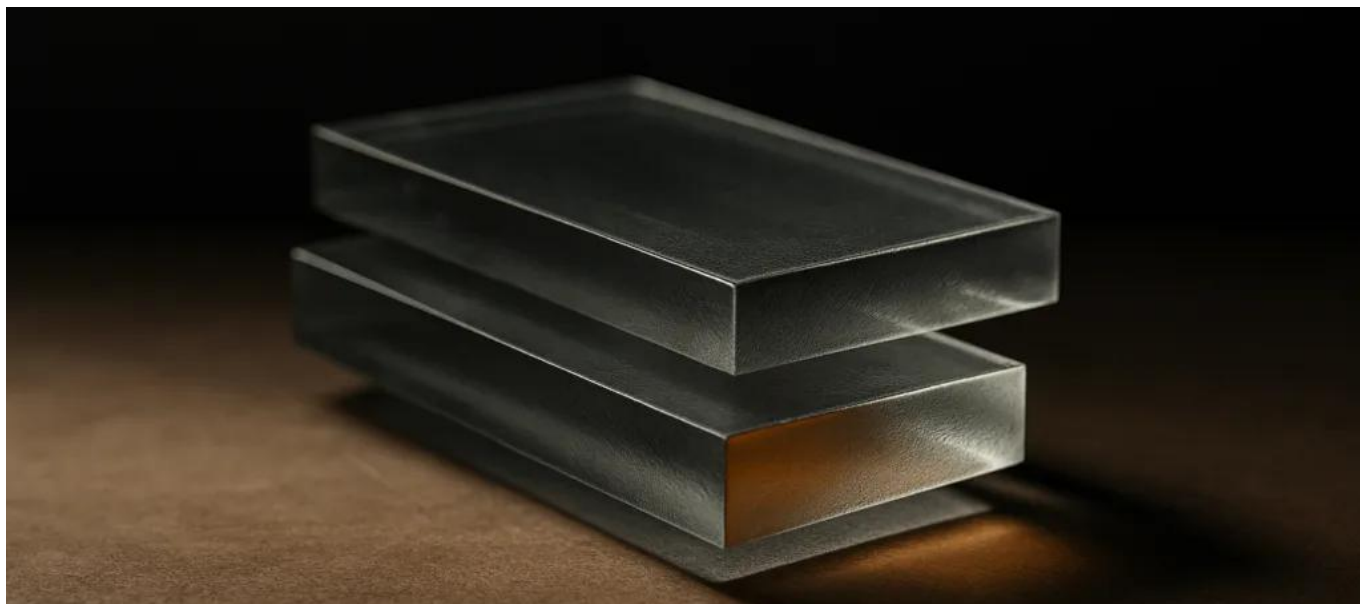




Setting up a self-hosted ComfyUI image pipeline: a CTO's guide to licences, GPU costs and node risk

What exactly do you own when you self-host ComfyUI? The core is GPL-3.0, and that licence covers the software and almost nothing else in your deployment.

THE SHORT VERSION

Self-hosting ComfyUI means assembling three independent licence chains. ComfyUI core is GPL-3.0 per the October 2026 OpenAPI spec metadata. Model weights carry their own terms (FLUX.2 Klein 4B is Apache 2.0, Klein 9B is non-commercial). Custom nodes carry no licence guarantee and no security boundary. Budget the GPU bill second: Comfy API rates run \$4.54/hr for an RTX PRO 6000 96GB up to \$11.23/hr for a B200 180GB.

Through September and October 2026, ComfyUI quietly became enterprise-shaped. The [Comfy API](#) arrived in September 2026: package a workflow, deploy it on selected GPU



infrastructure, call it from your application. Qwen-Image-2.1 landed in ComfyUI the same month at 7B parameters with open weights, native 2K output, RGBA transparency and editing across up to ten input images. FLUX 3 Image Klein followed by 15 to 18 October 2026 at 5.6B parameters, with base and distilled open weights, bounding-box support and multi-reference editing.

So the capability is there, on hardware you control, with models whose weights you can actually download. For a European SME that wants image generation inside its own network boundary, that is the first time the whole chain has been plausible.

The chain has three separate failure points, and three different organisations document them in three separate places. Nobody puts them in one view. That is what this piece is for.

The GPL-3.0 licence stops at the node boundary

The licence metadata in the [official API specification](#) says ComfyUI core is GPL-3.0. Models, checkpoints, LoRAs and custom nodes each retain their own, separate licences. That sentence is the whole legal architecture, and most teams read only the first half of it.

Take the FLUX family as the concrete case. Black Forest Labs splits its line sharply. FLUX.2 Klein 4B is Apache 2.0 and commercially usable with no revenue test. Klein 9B sits under the FLUX Non-Commercial Licence. FLUX.2 dev is free for non-commercial local deployment only. Three models from one vendor, three different answers to “can we ship output from this to a paying customer”.

There is a harder clause underneath. The [self-hosted commercial licence terms](#) do not authorise exposing the models via an API, reselling, or distributing to third parties. API use is governed separately under the FLUX API Terms. Read that carefully against your architecture. If your plan is “self-host FLUX on our own GPUs and put an internal REST endpoint in front of it so the marketing app can call it”, you have built the exact thing the self-hosted licence does not cover, and you may need the API terms instead.

My read: this is the clause that will catch the most European teams, because “self-hosted” and “we expose it internally over HTTP” feel like one decision to an engineer and two decisions to a lawyer. I would get this specific question answered in writing before the first GPU is provisioned.



Setting up a self-hosted ComfyUI image pipeline: a CTO's guide to licences, GPU costs and node risk

Stability AI takes a different approach entirely. Their [Core Models licence](#) is free unless the user organisation generates more than \$1 million in annual revenue, and that threshold counts revenue from any source, not just revenue attributable to the model. A profitable 12-person agency crosses it without ever selling an AI product. Platform credits, if you go that route, are \$0.01 each.

Then there is the sentence that ought to reshape how you evaluate Comfy Cloud. Comfy states that models available through [Comfy Cloud](#) are cleared for commercial use. That assurance covers the Cloud offering. It does not travel with the weights when you download the same model into a private deployment. The clearance is a property of the service, not of the file.

! The catch most teams miss

Downloading a model you used successfully on Comfy Cloud into your own cluster does not carry Comfy's commercial-use clearance with it. On your own hardware you are back under the model vendor's own terms, which for Klein 9B and FLUX.2 dev means non-commercial.

What a self-hosted ComfyUI pipeline actually costs next to Comfy Cloud

If you are comparing self-hosting against Comfy's managed path, here is the published side. Comfy Cloud runs Free at \$0 with no API access, Standard at \$20/mo (\$192/yr), Creator at \$35/mo (\$336/yr), Pro at \$100/mo (\$960/yr) and Team at \$700/mo (\$7,560/yr). Above that sits Comfy Cloud Professional, starting at \$5,000/month for up to ten licensed users, which explicitly lists commercial outputs, fine-tuning and LoRA training, and downstream client work as included.

That is a 7x jump from Team to Professional. What the jump buys is the tier that spells out commercial outputs and client work. For an agency doing production work for clients, the \$700 tier and the \$5,000 tier answer different questions.

GPU (COMFY API CATALOGUE)	MEMORY	RATE
RTX PRO 6000	96 GB	\$4.54/hr
H100 SXM	80 GB	\$6.23/hr



Setting up a self-hosted ComfyUI image pipeline: a CTO's guide to licences, GPU costs and node risk

GPU (COMFY API CATALOGUE)	MEMORY	RATE
H200 SXM	141 GB	\$7.71/hr
B200	180 GB	\$11.23/hr

That is a 2.5x spread across one catalogue. The RTX PRO 6000 at 96GB and \$4.54/hr is the interesting entry: for 5.6B and 7B parameter models like FLUX 3 Image Klein and Qwen-Image-2.1, 96GB is a lot of headroom. My take is that most image pipelines at this parameter scale have no business on a B200, and the \$11.23/hr tier exists for workloads that are not this one. Price your batch jobs against the cheapest card that fits the weights, then measure before you move up.

Storage is \$0.20/GB/month on Comfy's network storage. Cloud bills GPU time in credits, and Comfy states that idle time and workflow editing do not consume credits. That detail matters more than its size suggests: an iteration-heavy team pays for execution, not for sitting in the graph editor. On your own infrastructure you pay for the card whether anyone is using it or not, and that is the biggest structural cost difference between the two models.

Two more published limits. Workflow runtime is capped at 30 minutes on Standard and Creator, and one hour on Pro. The Comfy API also requires a paid Cloud subscription; the Free tier includes no API access, per the [developer overview](#).

One thing to settle before you sign anything: Comfy's own materials contradict each other on concurrency. The Cloud page lists Standard/Creator/Pro concurrency as 1/3/5. The API pricing lists per-deployment workers as 2/2/10/20 for Standard/Creator/Pro/Team. These may measure different things, but if your capacity plan depends on either number, get it confirmed in writing before procurement.

The custom node is where your real attack surface lives

Custom nodes run arbitrary Python inside your ComfyUI process. That is the whole value proposition of the ecosystem and also the whole risk. The GPL-3.0 licence on core says nothing about what a third-party node does, who wrote it, or what it has permission to reach.

On a laptop that is a personal risk. On a self-hosted node inside your network, with



credentials in the environment and network storage mounted, it is a supply-chain exposure with the same shape as a malicious npm package, except the review culture around ComfyUI nodes is younger and thinner than npm's. **A custom node is a dependency you have granted process-level trust**, chosen from a catalogue optimised for capability rather than provenance.

I have no incident data here, and I will not manufacture any. What I will say is that the mitigations are ordinary and unglamorous: pin node versions, read the source of anything that touches the filesystem or the network, run the ComfyUI process as a low-privilege user with no cloud credentials in its environment, and keep model storage on a mount the process can read but not write. None of that is specific to ComfyUI. All of it gets skipped routinely, because ComfyUI arrives looking like a creative tool rather than a server.

How I would sequence the build, licences first

1 Write down what the output is for, before choosing a model

Internal mockups, client deliverables and resold product are three different licence categories. FLUX.2 Klein 4B under Apache 2.0 answers all three. Klein 9B under the Non-Commercial Licence answers only the first. Deciding the use case last means re-picking the model after you have built around it.

2 Check the revenue threshold, not just the model terms

Stability AI's Core Models are free below \$1 million in annual organisational revenue, regardless of where that revenue comes from. This is an accounting fact about your company rather than a technical fact about your pipeline, so it belongs in the decision before the architecture.

3 Decide whether anything will call the model over HTTP

Black Forest Labs' self-hosted commercial licence does not authorise exposing models via an API or distributing to third parties; API use falls under separate FLUX API Terms. If your pipeline ends in an endpoint, resolve which terms apply before you write the endpoint.

4 Benchmark on the cheapest card that holds the weights

At 5.6B (FLUX 3 Image Klein) and 7B (Qwen-Image-2.1), the RTX PRO 6000 at 96GB and \$4.54/hr is the sane starting point in Comfy's catalogue. Moving to a B200 at \$11.23/hr is a 2.5x cost decision that a measurement should justify, not a spec sheet.



Setting up a self-hosted ComfyUI image pipeline: a CTO's guide to licences, GPU costs and node risk

5 Freeze and audit the node set, then treat it like production code

Custom nodes execute arbitrary Python in your process. Pin versions, review anything touching network or filesystem, and run the process without credentials it does not need. The GPL-3.0 licence on core gives you no assurance about any of this.

6 Add output marking into the workflow, not after it

If you are serving EU users, transparency marking is a pipeline requirement rather than a post-processing step. I covered the deadline and the mechanics in [the Article 50 guide](#).

Those six steps are ordered deliberately. The licence questions come first because they can invalidate the architecture, and the architecture questions cannot invalidate the licence.

When self-hosting is the wrong answer

Self-hosting wins when data residency is a hard requirement, when you need model weights that are not in a managed catalogue, or when your utilisation is high enough that a reserved GPU beats metered time. It loses in a few situations.

If your usage is bursty, the economics invert. Comfy Cloud bills GPU time in credits and does not charge for idle time or workflow editing. A self-hosted card charges for all of it. A team generating in concentrated bursts a few days a month is paying for silence the rest of the time.

If the commercial-use clearance is what you are actually buying, managed wins again. Comfy's statement that Cloud models are cleared for commercial use is a contractual position from an identifiable counterparty. On your own hardware you trade that for a stack of vendor licences you have to read and defend yourself. Some organisations would rather pay \$5,000/month for the Professional tier that names commercial outputs and client work than build an internal licence-review function.

And if nobody on the team owns the node security question, stay managed. A self-hosted ComfyUI without a named owner for dependency review is worse than a managed one, because you have taken on the risk and none of the control that justifies it.

WHERE I LAND

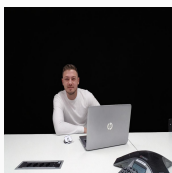


Setting up a self-hosted ComfyUI image pipeline: a CTO's guide to licences, GPU costs and node risk

*For most European SMEs under the \$1 million Stability threshold doing internal and client creative work, the honest starting configuration is Apache 2.0 weights (FLUX.2 Klein 4B), a self-hosted instance with a frozen and reviewed node set, and ****a hard architectural rule that nothing external calls the model over HTTP**** until the licence question is settled in writing. That setup is defensible with documents you can print. The moment any one of those three conditions breaks, revisit the managed path rather than patching around it.*

One honest gap: Comfy's published materials contradict themselves on concurrency, and I do not know which number governs a signed contract. If you want the comparative view on the models themselves rather than the deployment around them, I wrote that up in [the image model comparison for CTOs](#).

The pipeline is buildable today on your own hardware. The GPL-3.0 badge on the core repository is the least interesting licence in the whole stack.



WRITTEN BY

Artur Markus

Software engineer in Basel building AI infrastructure and agentic systems, with a background in pharmaceutical automation, GxP compliance, and validation. I help teams turn AI from a chatbot into a reliable operational layer.

Book a meeting →