

RUNNING OPEN-WEIGHT LLMS AT RISC

llamafire, Ollama@RISC, Ollama@MACH-2, ollama.com



Wolfgang Schreiner

Research Institute for Symbolic Computation (RISC)

Wolfgang.Schreiner@risc.jku.at



Programs with LLMs

```
qftquad1> source /zvol/formal/python3-agentic/bin/activate  
(python3-agentic) qftquad1!4> python ollama-risc.py
```

```
Model: qwen3.5:2b
```

```
Prompt: tell me a (short) joke
```

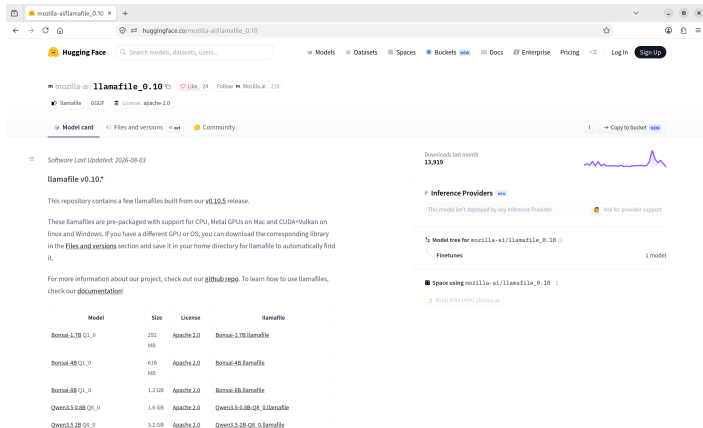
```
Why don't scientists trust atoms?
```

```
Because they **make up everything**!
```

We have various options to run LLMs and LLM-based programs.

llamafire: Stand-alone LLM Executables

https://huggingface.co/mozilla-ai/llamafire_0.10



mozilla-ai/llamafire_0.10 Like 24 Follow Mozilla AI 238

llamafire GGUF License: apache-2.0

[Model card](#) [Files and versions](#) [Community](#) [Copy to bucket](#)

Software Last Updated: 2026-08-03

llamafire v0.10.*

This repository contains a few llamafiles built from our [v0.10.5](#) release.

These llamafiles are pre-packaged with support for CPU, Metal GPUs on Mac and CUDA/Vulkan on linux and Windows. If you have a different GPU or OS, you can download the corresponding library in the [Files and versions](#) section and save it in your home directory for llamafire to automatically find it.

For more information about our project, check out our [github repo](#). To learn how to use llamafiles, check our [documentation](#)!

Model	Size	License	llamafire
Bonpai-1.7B Q1_0	252 MB	Apache 2.0	Bonpai-1.7B.llamafire
Bonpai-4B Q1_0	616 MB	Apache 2.0	Bonpai-4B.llamafire
Bonpai-8B Q1_0	1.2 GB	Apache 2.0	Bonpai-8B.llamafire
Qwen3.5-8B-Q1_0	1.6 GB	Apache 2.0	Qwen3.5-8B-Q1_0.llamafire
Qwen3.5-72B-Q1_0	3.2 GB	Apache 2.0	Qwen3.5-72B-Q1_0.llamafire

Downloads last month: **13,919**

Inference Providers

This model isn't deployed by any Inference Provider. [Ask for provider support](#)

Model tree for mozilla-ai/llamafire_0.10

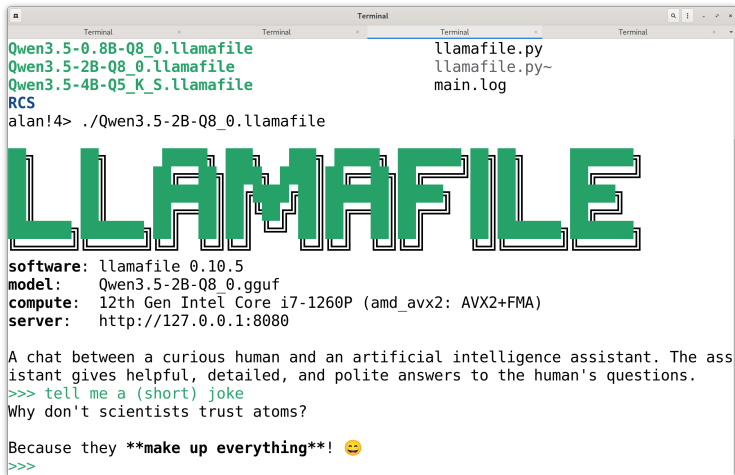
Finetunes: 1 model

Space using mozilla-ai/llamafire_0.10

[WURLPRAJANT/llama-ai](#)

Download and run executable LLM.

Chat Interface



```
Terminal
Qwen3.5-0.8B-Q8_0.llamafile
Qwen3.5-2B-Q8_0.llamafile
Qwen3.5-4B-Q5_K_S.llamafile
RCS
alan!4> ./Qwen3.5-2B-Q8_0.llamafile
llamafile.py
llamafile.py~
main.log

LLAMAFILE

software: llamafile 0.10.5
model:   Qwen3.5-2B-Q8_0.gguf
compute: 12th Gen Intel® Core i7-1260P (amd_avx2: AVX2+FMA)
server:  http://127.0.0.1:8080

A chat between a curious human and an artificial intelligence assistant. The ass
istant gives helpful, detailed, and polite answers to the human's questions.
>>> tell me a (short) joke
Why don't scientists trust atoms?

Because they **make up everything**! 😊
>>>
```

We can execute the LLM interactively.

Python Program

```
from langchain_openai import ChatOpenAI

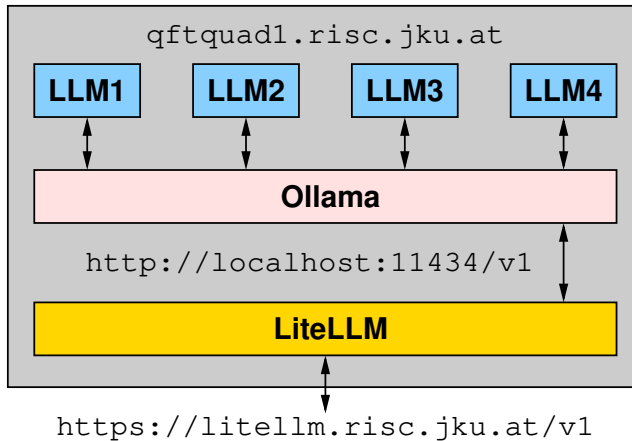
OLLAMA_URL="http://localhost:8080/v1"
OLLAMA_API_KEY="dummy"

llm = ChatOpenAI(
    openai_api_base=OLLAMA_URL,
    openai_api_key=OLLAMA_API_KEY,
    temperature=0,           # prefer reliability over creativity
    reasoning_effort="none", # turn off reasoning/thinking mode
)

prompt = input("Prompt: ")
response = llm.invoke(prompt)
print(response.content)
```

We can connect to LLM servers via a URL.

Ollama@RISC: Self-Hosted LLMs



An interface to multiple LLMs that can be used locally and remotely (installed by Werner Danielczyk-Landerl).

Usage

- **Chat Interface:** on qftquad1 (needs RISC account)

```
qftquad1> ollama run qwen3.5:2b --think=false
```

```
>>> tell me a (short) joke
```

```
Q: What do you call a fake noodle?
```

```
A: An Impasta.
```

- **Local program:** on qftquad1 (needs RISC account)

```
OLLAMA_URL="http://localhost:11434/v1"
```

```
OLLAMA_API_KEY="dummy"
```

```
llm = ChatOpenAI(model="qwen3.5:2b", ...)
```

- **Remote program:** on any machine in the internet (needs API key)

```
OLLAMA_URL="https://litellm.risc.jku.at/v1"
```

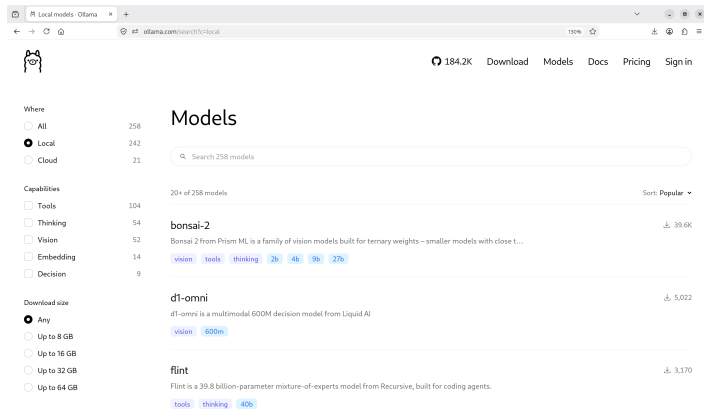
```
OLLAMA_API_KEY="insert your API key here"
```

```
llm = ChatOpenAI(model="qwen3.5:2b", ...)
```

See API key for RISC users in `/zvol/formal/ollama/Ollama@RISC.txt`.

Models

<https://ollama.com/models>



The screenshot shows the Ollama website's 'Models' page. The browser address bar shows 'ollama.com/search?c=local'. The page features a sidebar on the left with filters for 'Where' (All, Local, Cloud) and 'Capabilities' (Tools, Thinking, Vision, Embedding, Decision). The main content area displays a list of models, including 'bonsai-2', 'd1-omni', and 'flint'. Each model entry includes its name, a brief description, and a download size. The 'bonsai-2' model is highlighted with a blue background. The 'd1-omni' model is described as a multimodal 600M decision model from Liquid AI. The 'flint' model is described as a 39.8 billion-parameter mixture-of-experts model from Recursive, built for coding agents.

Local models - Ollama

ollama.com/search?c=local

184.2K Download Models Docs Pricing Sign in

Where

- All 258
- Local 242**
- Cloud 21

Capabilities

- Tools 104
- Thinking 54
- Vision 52
- Embedding 14
- Decision 9

Download size

- Any**
- Up to 8 GB
- Up to 16 GB
- Up to 32 GB
- Up to 64 GB

Models

Search 258 models

20+ of 258 models Sort: Popular

bonsai-2 ↓ 39.6K

Bonsai 2 from Prism ML is a family of vision models built for ternary weights – smaller models with close t...

vision tools thinking 2b 4b 9b 27b

d1-omni ↓ 5,022

d1-omni is a multimodal 600M decision model from Liquid AI

vision 600m

flint ↓ 3,170

Flint is a 39.8 billion-parameter mixture-of-experts model from Recursive, built for coding agents.

tools thinking 40b

More than 240 models available for local installation.

Models

```
qftquad1> ollama ls
```

NAME	ID	SIZE	MODIFIED
codellama:7b	8fdf8f752f6e	3.8 GB	8 days ago
codestral:22b	0898a8b286d5	12 GB	7 days ago
deepseek-coder:1.3b	3ddd2d3fc8d2	776 MB	8 days ago
deepseek-coder:33b	accec7c0b0fd9	18 GB	7 days ago
deepseek-coder:6.7b	ce298d984115	3.8 GB	8 days ago
deepseek-r1:1.5b	e0979632db5a	1.1 GB	9 days ago
deepseek-r1:32b	edba8017331d	19 GB	8 days ago
deepseek-r1:8b	6995872bfe4c	5.2 GB	8 days ago
dolphin-mistral:7b	5dc8c5a2be65	4.1 GB	7 days ago
gemma3:12b	b92b218582c5	8.2 GB	7 days ago
gemma3:1b	97558784bdc9	806 MB	7 days ago
gemma3:27b	6be2e2ba6843	17 GB	7 days ago
gemma3:4b	68ceab7e468a	3.4 GB	7 days ago
gemma4:12b	6114515d63c1	8.0 GB	8 days ago
...			

Currently more than 60 models installed (more on demand).

Performance

The server has 2 Xeon CPUs with 8 cores each but *no GPU/TPU/NPU*.

- LLMs run with 16 CPU threads without any special hardware acceleration.
 - 10–20 times slower than with dedicated AI hardware.
- LLM execution time breakup:

1. Load LLM into memory.

- Disk speed is bottleneck: 5 s–2 min (linear in number of model parameters).
- Ollama is configured to keep up to 5 idle LLMs in memory for 4 mins.

```
qftquad1> ollama ps
```

NAME	ID	SIZE	PROCESSOR	CONTEXT	RUNNER	UNTIL
qwen3.5:2b	0689d44085e0	4.6 GB	100% CPU	128000	llamacpp	4 minutes from now

2. Process system prompt.

- CPU speed is bottleneck: 0.1 s–X min (quadratic in number of input tokens).

3. Output tokens.

- Memory speed is bottleneck: 2–20 tokens/s (linear in number of output tokens).

Typically ≤ 2 B parameter models run “fast”, ≤ 4 B models “reasonably”, ≤ 8 B models “acceptably”; also some ≤ 32 B models may be useful.

Benchmarks (Highlights)

- “Fast”:

```
[2] deepseek-coder:1.3b: load=6.2s run=2.3s first=0.1s output=2.2s tokens=49 rate=21.0tokens/s
[10] gemma3:1b: load=6.0s run=1.4s first=0.2s output=1.2s tokens=31 rate=22.7tokens/s
[23] llama3.2:3b: load=34.5s run=1.6s first=0.1s output=1.4s tokens=19 rate=12.2tokens/s
[45] qwen2.5-coder:3b: load=14.4s run=1.2s first=0.2s output=1.0s tokens=16 rate=13.4tokens/s
[49] qwen2.5:3b: load=14.9s run=2.0s first=0.2s output=1.9s tokens=25 rate=12.3tokens/s
[61] smollm2:1.7b: load=14.0s run=1.3s first=0.2s output=1.1s tokens=17 rate=13.4tokens/s
[62] tinyllama:1.1b: load=6.4s run=3.2s first=0.2s output=3.0s tokens=46 rate=14.5tokens/s
[63] smollm3:3b-q4_k_m: load=19.2s run=2.8s first=0.2s output=2.6s tokens=34 rate=12.1tokens/s
```

- “Reasonable”:

```
[12] gemma3:4b: load=22.4s run=4.2s first=0.4s output=3.8s tokens=36 rate=8.6tokens/s
[15] gemma4:e2b: load=21.8s run=3.1s first=0.3s output=2.9s tokens=31 rate=9.8tokens/s
[19] granite4.2:3b: load=26.7s run=5.9s first=0.2s output=5.7s tokens=50 rate=8.5tokens/s
[22] llama3.1:8b: load=41.0s run=3.0s first=0.3s output=2.7s tokens=21 rate=7.0tokens/s
[36] phi3.5:3.8b: load=129.5s run=6.3s first=0.2s output=6.1s tokens=63 rate=9.9tokens/s
[38] phi3:3.8b: load=117.9s run=1.9s first=0.2s output=1.7s tokens=20 rate=10.7tokens/s
[40] phi4-mini:3.8b: load=31.0s run=2.4s first=0.2s output=2.2s tokens=23 rate=9.7tokens/s
[53] qwen3.5:2b: load=14.9s run=1.7s first=0.2s output=1.5s tokens=17 rate=10.0tokens/s
[59] qwen3:4b: load=47.4s run=36.4s first=0.3s output=36.1s tokens=268 rate=7.4tokens/s
```

Benchmarks (Highlights)

- “Acceptable”:

[0] codellama:7b: load=16.8s run=7.4s first=0.5s output=6.9s tokens=38 rate=5.2tokens/s
[4] deepseek-coder:6.7b: load=25.9s run=11.0s first=0.2s output=10.8s tokens=60 rate=5.4tokens/s
[8] dolphin-mistral:7b: load=28.4s run=3.3s first=0.3s output=3.0s tokens=18 rate=5.5tokens/s
[13] gemma4:12b: load=42.6s run=3.2s first=0.8s output=2.4s tokens=17 rate=5.3tokens/s
[16] gemma4:e4b: load=12.9s run=2.8s first=0.5s output=2.3s tokens=19 rate=6.7tokens/s
[20] granite4.2:8b: load=56.1s run=4.5s first=0.3s output=4.2s tokens=19 rate=4.2tokens/s
[25] mathstral:7b: load=28.1s run=45.8s first=0.3s output=45.6s tokens=235 rate=5.1tokens/s
[27] ministral-3:3b: load=41.7s run=4.7s first=0.3s output=4.4s tokens=26 rate=5.5tokens/s
[29] mistral-nemo:12b: load=54.0s run=3.0s first=0.3s output=2.7s tokens=14 rate=4.6tokens/s
[30] mistral-openorca:7b: load=23.0s run=10.1s first=0.2s output=9.9s tokens=57 rate=5.6tokens/s
[32] mistral:7b: load=28.3s run=5.9s first=0.3s output=5.6s tokens=33 rate=5.6tokens/s
[33] mistrallite:7b: load=26.5s run=33.9s first=0.2s output=33.6s tokens=185 rate=5.5tokens/s
[39] phi4-mini-reasoning:3.8b: load=105.2s run=74.1s first=3.1s output=71.0s tokens=504 rate=6.8tokens/s
[46] qwen2.5-coder:7b: load=23.6s run=5.0s first=0.3s output=4.7s tokens=26 rate=5.2tokens/s
[50] qwen2.5:7b: load=27.0s run=4.2s first=0.2s output=4.0s tokens=25 rate=6.0tokens/s
[54] qwen3.5:4b: load=22.7s run=2.9s first=0.4s output=2.6s tokens=17 rate=5.8tokens/s
[55] qwen3.5:9b: load=32.9s run=4.1s first=0.4s output=3.7s tokens=17 rate=4.1tokens/s
[60] qwen3:8b: load=40.1s run=4.7s first=0.3s output=4.4s tokens=22 rate=4.7tokens/s

Benchmarks (Highlights)

Also some large models may perform reasonably well (typically “mixture of expert” models that employ for a given prompt only a fraction of their parameters).

- “Large”:

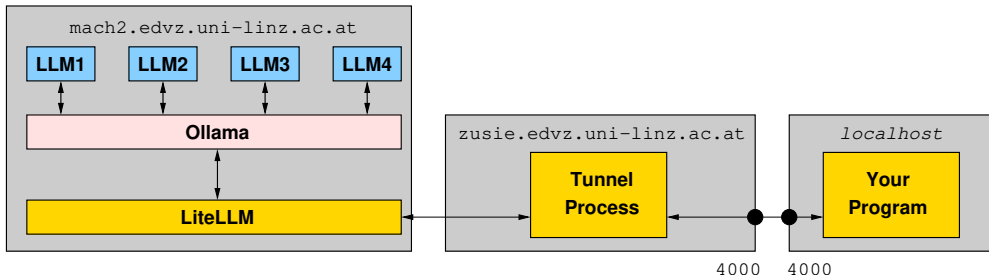
```
[17] gpt-oss:20b: load=21.3s run=4.8s first=0.9s output=3.9s tokens=23 rate=4.7tokens/s
[35] nemotron-3-nano:30b: load=80.9s run=2.3s first=0.3s output=2.0s tokens=16 rate=6.8tokens/s
[51] qwen3-coder:30b: load=85.7s run=1.5s first=0.2s output=1.3s tokens=15 rate=9.9tokens/s
[58] qwen3:30b: load=69.3s run=35.3s first=0.2s output=35.1s tokens=284 rate=8.0tokens/s
```

Note that all benchmarks were performed with prompt “Tell me a (short) joke” and thus did not include any significant prompt processing time.

Ollama@MACH-2: A Self-Hosted Alternative

Ollama is also installed at the JKU supercomputer MACH-2 (by Johann Messner).

<https://www.risc.jku.at/projects/mach2/>



Access via a tunnel from your computer to the JKU front-end computer "zusie".

Access

1. Create a tunnel: (requires an account on zusie)

```
ssh -l <login-name> <secret> -L 4000:localhost:4000 zusie.edvz.uni-linz.ac.at
```

2. Connect to the tunnel:

```
OLLAMA_URL="http://localhost:4000/v1"  
OLLAMA_API_KEY="insert your API key here"  
llm = ChatOpenAI(model="qwen35:2b-2.3B-Q8_0-s0", ...)
```

See API key for RISC users in /zvol/formal/ollama/Ollama@MACH-2.txt;
see ssh “secret” in /zvol/formal/ollama/zusie.sh;
get zusie account by sending form to scadmin@jku.at (Johann Messner).

https://help.jku.at/download/attachments/2066317/NutzungsKennung-fuer-ComputeServer_ger.pdf

Models

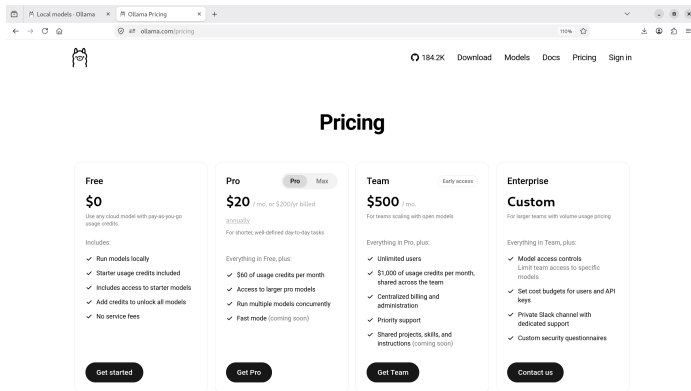
Currently more than 10 models installed (more on demand).

```
[0] deepseek0: -
[1] gemma4_e2b_s1: load=8.5s run=2.7s first=0.2s output=2.5s tokens=34 rate=12.4tokens/s
[2] ibmgranite43b-h_s1: load=8.4s run=2.6s first=0.3s output=2.3s tokens=25 rate=9.6tokens/s
[3] llama3.2B-Q4_K_M-s0: load=66.6s run=1.5s first=0.1s output=1.3s tokens=19 rate=12.9tokens/s
[4] phi4-mini3.8B-Q4_K_M-s1: load=86.7s run=2.2s first=0.2s output=2.1s tokens=23 rate=10.4tokens/s
[5] phi4-mini-reasoning3.8B-Q4_K_M-s0: load=64.7s run=28.3s first=0.3s output=28.0s tokens=245 rate=8.7tokens/s
[6] qwen2.5-coder3.1B-Q4_K_M-s1: load=7.8s run=2.2s first=0.1s output=2.0s tokens=26 rate=12.0tokens/s
[7] qwen35:2b-2.3B-Q8_0-s0: load=8.8s run=2.3s first=0.2s output=2.0s tokens=17 rate=7.5tokens/s
[8] qwen3.5:4b-4.7B-Q4_K_M-s0: load=14.0s run=2.7s first=0.3s output=2.4s tokens=17 rate=6.2tokens/s
[9] qwen3.5-9.7B-Q4_K_M-s1: load=26.1s run=4.2s first=0.4s output=3.8s tokens=17 rate=4.0tokens/s
[10] smollm3_3b_q6_k_s0: load=27.7s run=2.2s first=0.2s output=2.0s tokens=15 rate=6.8tokens/s
[11] smollm3-3.1B-Q4_K_M-s1: load=21.2s run=1.3s first=0.1s output=1.2s tokens=15 rate=11.6tokens/
```

Models are executed on one MACH-2 node by 2 Xeon CPUs with 12 cores each (using two Ollama instances that create 12 threads each); performance is typically by a factor of 1.5–2 higher than on qftquad1.

ollama.com: Remotely Hosted LLMs

LLM server with OpenAI endpoint <https://ollama.com/v1>.



The screenshot shows the Ollama Pricing page in a web browser. The browser tabs include 'Local models - Ollama' and 'Ollama Pricing'. The address bar shows 'ollama.com/pricing'. The page features a navigation bar with links for 'Download', 'Models', 'Docs', 'Pricing', and 'Sign in'. The main heading is 'Pricing'. Below it, there are four pricing tiers: Free, Pro, Team, and Enterprise. Each tier has a list of features and a 'Get' button.

Free	Pro	Team	Enterprise
\$0 Use any cloud model with pay-as-you-go usage credits.	\$20 /mo, or \$200/yr billed <i>annually</i> For shorter, well-defined day-to-day tasks	\$500 /mo. For teams scaling with open models	Custom For larger teams with volume usage pricing
Includes: ✓ Run models locally ✓ Starter usage credits included ✓ Includes access to starter models ✓ Add credits to unlock all models ✓ No service fees	Everything in Free, plus: ✓ \$60 of usage credits per month ✓ Access to larger pro models ✓ Run multiple models concurrently ✓ Fast mode (coming soon)	Everything in Pro, plus: ✓ Unlimited users ✓ \$1,000 of usage credits per month, shared across the team ✓ Centralized billing and administration ✓ Priority support ✓ Shared projects, skills, and instructions (coming soon)	Everything in Team, plus: ✓ Model access controls Limit team access to specific models ✓ Set cost budgets for users and API keys ✓ Private Slack channel with dedicated support ✓ Custom security questionnaires
Get started	Get Pro	Get Team	Contact us

RISC has purchased a “Pro” licence (60\$ tokens per month); send email to Wolfgang.Schreiner@risc.jku.at for an API key.

Model pricing

Pay as you go with usage credits. Prices are per million tokens. Off-peak pricing apply outside 12:00 and 18:00 UTC on weekdays and all day on weekends.

Model	Input	Cached Input	Output
clef	\$0.24	-	-
deepseek-v4.1-flash	\$0.30	\$0.006	\$1.20
deepseek-v4.1-flash (Off-Peak)	\$0.15	\$0.003	\$0.60
deepseek-v4-pro	\$1.32	\$0.044	\$3.96
deepseek-v4-pro (Off-Peak)	\$0.66	\$0.022	\$1.98
gemma4	\$0.14	\$0.05	\$0.40
glm-5.3	\$1.40	\$0.26	\$4.40
glm-5.3-flash	\$0.15	\$0.03	\$0.50
glm-5.2	\$1.40	\$0.26	\$4.40
gpt-oss:120b	\$0.15	\$0.014	\$0.60
gpt-oss:20b	\$0.07	\$0.035	\$0.30
kimi-k3	\$3.00	\$0.30	\$15.00
kimi-k2.7-code	\$0.95	\$0.19	\$4.00
kimi-k2.6	\$0.95	\$0.16	\$4.00
minimax-m3.1-flash	\$0.15	\$0.015	\$0.60
minimax-m3	\$0.60	\$0.12	\$2.40
minimax-m2.7	\$0.30	\$0.06	\$1.20
mistral-large-4	\$0.68	\$0.07	\$2.09
mistral-large-3	\$0.50	-	\$1.50
muse-spark-open	\$1.25	\$0.15	\$4.25
nemotron-3-nano	\$0.06	-	\$0.24
nemotron-3-super	\$0.015	\$0.015	\$0.60
nemotron-3-ultra	\$0.10	\$0.10	\$3.00

Benchmarks

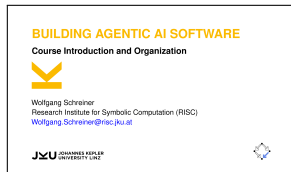
```
[0] clef:27b: -
[1] deepseek-v4.1-flash: load=0.1s run=0.6s first=0.4s output=0.1s tokens=16 rate=28.8tokens/s
[2] deepseek-v4-pro: load=0.0s run=0.9s first=0.6s output=0.3s tokens=16 rate=18.2tokens/s
[3] gemma4:31b: load=0.1s run=0.4s first=0.3s output=0.1s tokens=18 rate=42.1tokens/s
[4] glm-5.3: load=-0.4s run=1.8s first=0.7s output=1.1s tokens=217 rate=121.6tokens/s
[5] glm-5.3-flash: load=0.2s run=1.4s first=0.3s output=1.1s tokens=79 rate=56.9tokens/s
[6] glm-5.2: load=0.1s run=0.7s first=0.6s output=0.2s tokens=18 rate=25.4tokens/s
[7] gpt-oss:120b: load=0.1s run=0.7s first=0.4s output=0.3s tokens=22 rate=31.3tokens/s
[8] gpt-oss:20b: load=0.1s run=0.7s first=0.4s output=0.3s tokens=31 rate=41.6tokens/s
[9] kimi-k3: load=0.4s run=1.3s first=1.1s output=0.2s tokens=10 rate=7.7tokens/s
[10] kimi-k2.7-code: load=0.2s run=0.8s first=0.5s output=0.3s tokens=15 rate=19.0tokens/s
[11] kimi-k2.6: load=0.1s run=0.8s first=0.6s output=0.2s tokens=11 rate=14.0tokens/s
[12] minimax-m3.1-flash: load=0.1s run=2.1s first=1.7s output=0.4s tokens=6 rate=2.9tokens/s
[13] minimax-m3: load=0.4s run=0.9s first=0.8s output=0.2s tokens=5 rate=5.4tokens/s
[14] minimax-m2.7: load=-3.8s run=9.3s first=0.9s output=8.4s tokens=116 rate=12.4tokens/s
[15] mistral-large-4: load=-0.3s run=2.3s first=2.0s output=0.3s tokens=11 rate=4.7tokens/s
[16] mistral-large-3:675b-cloud: load=1.3s run=1.1s first=0.6s output=0.5s tokens=15 rate=13.7tokens/s
[17] muse-spark-open: load=0.2s run=1.7s first=1.5s output=0.2s tokens=18 rate=10.5tokens/s
[18] nemotron-3-nano:30b: load=0.1s run=0.4s first=0.3s output=0.1s tokens=17 rate=39.9tokens/s
[19] nemotron-3-super: load=-0.1s run=0.7s first=0.5s output=0.2s tokens=13 rate=19.3tokens/s
[20] nemotron-3-ultra: load=0.1s run=0.7s first=0.6s output=0.1s tokens=8 rate=11.3tokens/s
```

About 10 times faster than the self-hosted installation (but with limited model selection and token usage).

New Course

Building Agentic AI Software

Wolfgang Schreiner, SS 2027



This is a one-semester elective university course (3 ECTS) aimed at master degree students in mathematics, computer science, or related areas. Its aim is to give a basic introduction into the development of "agentic AI" software where a problem is solved with the help of a large language model (LLM). This LLM drives the control flow of the program and may use external "tools" for solving specific subproblems and/or interacting with the environment. Students learn the basic principles of agentic AI and their practical utilization in the Python-based agent development framework LangChain/LangGraph. The overall goal is to let students add agentic AI to their "tool belt" of software development techniques for their career in science or engineering.

The course does not assume any background in artificial intelligence or machine learning. However, it does assume some programming skills in Python (or any other programming language; in this case, the student must be ready to self-study Python during the semester). The course consists of 12 lectures of 90 minutes where the basic principles and the practical software development are presented (the lectures take base in class but are also live-streamed and recorded on a best-effort basis).

Resources

- Slides, source code, API keys:

```
qftquad1> ls /zvol/formal/ollama
```

Ollama@MACH-2.txt	ollama-com-benchmark.py	ollama-mach2-benchmark.py	ollama-risc-benchmark.py	slides.pdf
Ollama@RISC.txt	ollama-com.py	ollama-mach2.py	ollama-risc.py	zusie.sh

- RISC portal How-To:

The screenshot shows the RISC portal website. The header includes the RISC logo and navigation links. The main content area displays the 'Ollama@RISC' page, which includes a list of links for 'How to add a trusted user role', 'How to Use ssh Tunnel to Send Email with Thunderbird', 'How to make RISC Summer better', 'Conferences', and 'How to Adjust Phone for'. The page also features a calendar for October 2026 and a section for 'Upcoming Events'.

RESEARCH INSTITUTE FOR SYMBOLIC COMPUTATION | RISC

Navigation: Home, About, News, Events, Projects, Research, Education, RISC Forum, Internal, Conferences

You are here: Home / Internal / How To / Ollama@RISC

Contents View Edit Rules Sharing Actions Display Add new... State: Private

Ollama@RISC

by Wolfgang Schreiner — last modified Oct 22, 2026 12:08 PM — History

Running open-weight LLMs at RISC

RISC Forum Presentation — by Wolfgang Schreiner — last modified Oct 22, 2026 12:09 PM

Upcoming Events

RISC Forum
Oct 26, 2026 01:30 PM - 02:30 PM