Junction 2026: Modal onboarding guide

13 to 15 November 2026, Helsinki (Espoo). For organizers, mentors and participants.

$100 in Modal credits per participant

1. Welcome

Modal is the compute partner for Junction 2026. Every participant gets $100 in credits for GPUs and CPUs, and Modal engineers will be on site. Your Modal contact is Filip Makraduli (filip@modal.com). The event page for participants is https://autohackathon.modal.dev/e/junction-2026.

Event	Junction 2026, 13 to 15 November 2026
Credits	$100 per workspace, code VVX-2QX-8HW, redeem by 16 November 2026
Support	Modal docs, the Modal Community Slack and the examples repo.
Workshop	Modal workshop, Saturday 14 Nov 13:30, workshop stage

2. Modal in five minutes

Modal lets you run Python on cloud GPUs without setting anything up: write a function, add a decorator, run `modal run`, and it executes on an A100 or H100 in seconds. `modal deploy` gives you a public URL for your demo. Your credits cover everything.

```
pip install modal
modal setup
```

```
modal run hello_world.py        # runs your function in the cloud
modal deploy my_app.py          # gives it a public URL
```

- Run your agent's tool calls and generated code inside Modal Sandboxes instead of on your laptop
- Serve an open-weight LLM behind a URL with vLLM or SGLang, billed per second, scaling to zero between demos
- Spin up isolated containers per user or per task for code execution, evals and browser automation
- Transcribe or synthesise speech on GPUs (Whisper, Parakeet, TTS) with streaming WebSocket endpoints
- Generate or analyse images and video (Flux, YOLO, VLMs) with weights cached in a Volume

3. Credits and how to redeem them

VVX-2QX-8HW

1. Create a free Modal account at modal.com/signup (GitHub login). One redemption per workspace.
2. Open the redemption link (or scan the QR) and submit your workspace name and the code.
3. Credits appear on modal.com/settings/usage within a minute. Install the client with `pip install modal` and run `modal setup`.

Link: https://modal.fillout.com/t/qMXCmRGseUus?code=VVX-2QX-8HW

One redemption per workspace. Teams: one shared workspace, redeem once, invite teammates.
Expiry gates redemption only; credits last well beyond the event.

Common problems

- Code already redeemed: each workspace can redeem a code once; check Settings -> Usage first.
- Wrong workspace name: use the workspace slug shown in the dashboard URL, not your email.
- Code expired: ask the organizer or the Modal contact; expiry gates redemption only, credits last well beyond the event.
- GPU job pending: try another GPU type (gpu="A100" or "L40S") or lower the concurrency.

What to say on stage

"Modal gives every one of you $100 of GPU credits. Scan the QR on the screen, redeem once per team workspace, and you can run an H100 from a Python function in the next ten minutes. Modal mentors are at the booth all weekend."

4. Hardware, limits and how Modal fits this event

Credits work on every GPU Modal offers (T4, L4, A10, L40S, A100, H100, H200, B200, B300) and on CPUs. Pick the GPU in code with gpu="H100" and fall back to another type if a job sits pending; no single GPU type is guaranteed at a given moment. New workspaces start on the Starter plan with its concurrency limits; organizers can ask Modal to raise them for the weekend.

- GPU-heavy use: expect H100/A100 demand; new workspaces start on Starter limits, so ask Modal to raise concurrency for the weekend and tell teams to fall back to A10/L40S when a type is pending.
- Agent and code-execution projects: Sandboxes are the fit; budget is CPU-dominated, so credits go a long way.
- Real-time audio or video: use WebSocket or WebRTC endpoints and keep containers warm with min_containers during demos.
- Mostly first-time Modal users: run the kick-off workshop early, point teams to hello_world and the examples before they design around a GPU.
- Large event (1200 participants): expect a redemption spike at kick-off; the code has enough uses and the Fillout link is prefilled, so redemption takes under a minute.

Relevant links

- Sandboxes for agents: https://modal.com/docs/guide/sandbox
- Build a coding agent: https://modal.com/docs/examples/agent
- LLM inference on Modal: https://modal.com/docs/examples/vllm_inference
- GPU guide: https://modal.com/docs/guide/gpu
- Sandbox snapshots: https://modal.com/docs/guide/sandbox-snapshots
- Whisper transcription: https://modal.com/docs/examples/batched_whisper

- Streaming and WebSockets: https://modal.com/docs/guide/webhooks
- Image generation with Flux: https://modal.com/docs/examples/flux
- Model weights in Volumes: https://modal.com/docs/guide/model-weights
- Modal docs: https://modal.com/docs/guide
- Examples repo: https://github.com/modal-labs/modal-examples
- Pricing and GPUs: https://modal.com/pricing
- Community Slack: https://modal.com/slack

5. Examples for Junction 2026

Clone github.com/modal-labs/modal-examples and run from the repo root.

Run OpenAI-compatible LLM inference with Gemma and vLLM
Fits llm, code. https://modal.com/docs/examples/vllm_inference

```
modal run 06_gpu_and_ml/llm-serving/vllm_inference.py
```

Build a coding agent with Modal Sandboxes and LangGraph
Fits agents, llm, agent, code, sandboxes. https://modal.com/docs/examples/agent

```
modal run 13_sandboxes/codelangchain/agent.py
```

Run massively parallel evals on Modal Sandboxes
Fits agents, llm, agents, code, sandboxes. https://modal.com/docs/examples/harbor_evals

```
modal run 13_sandboxes/harbor_evals.py
```

Create a Chatterbox TTS API on Modal
Fits llm, speech, about, voice. https://modal.com/docs/examples/chatterbox_tts

```
modal run 06_gpu_and_ml/text-to-audio/chatterbox_tts.py
```

Run Flux fast on H100s with `torch.compile`
Fits vision. https://modal.com/docs/examples/flux

```
modal run 06_gpu_and_ml/stable_diffusion/flux.py
```

Painting flowers with code
Fits llm, training, code, dojo. https://modal.com/docs/examples/paint_flowers

```
modal run 06_gpu_and_ml/paint_flowers/paint_flowers.py
```

Deploy LangChain and LangGraph applications with LangServe
Fits agents, llm, agent, code. https://modal.com/docs/examples/langserve

```
modal run 13_sandboxes/codelangchain/langserve.py
```

6. The Modal challenge

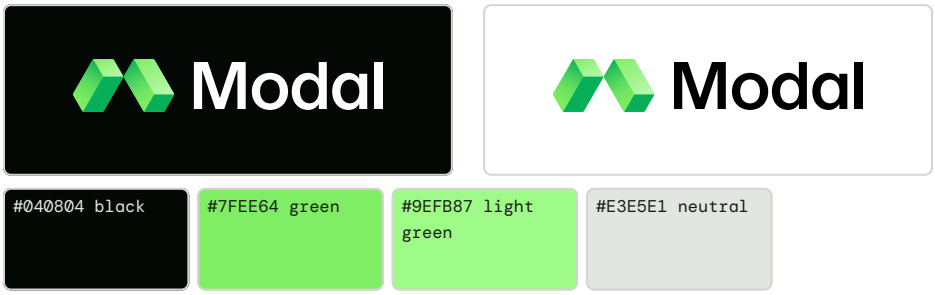
Best use of Modal at Junction 2026. Use Modal to power the project you are already building: serve a model on a GPU, run your agent's code in Sandboxes, fine-tune on your own data or fan out a batch job. We look for working systems where Modal does the heavy lifting: effective use of Modal (40%), technical ambition (25%), impact (20%), demo quality (15%).

Event specifics

Junction 2026, Europe's largest hackathon, 13 to 15 November at Hype Areena, Espoo. Theme: European Tech Renaissance. About 1,200 participants, mostly students and young engineers from across Europe, in teams of 2 to 5. Most teams build AI products: LLM

apps on open-weight models, agents that run code, voice assistants, computer vision, some fine-tuning. Modal is Technology Partner: booth, Best use of Modal tech challenge, 30 minute workshop on Saturday, social and job posts. Every participant gets $100 in Modal credits; GPUs from T4 to B300; Sandboxes for agent code; Dojo for fine-tuning and RL. Mentors from Modal engineering and DevRel at the booth and on Discord. Prize to be determined.

7. Brand guidelines



- Write "Modal", never "Modal Labs", "MODAL" or "modal" in prose. The company is Modal, the product is Modal.
- Use the supplied logo files; do not recolour, rotate, outline or add effects. Keep clear space of at least the height of the M around it.
- On dark backgrounds use the light logo, on light backgrounds the dark logo. Green accent is
- Do not combine the Modal logo with other sponsor logos into one lockup; place it with equal size and spacing.

Blurb, 50 words

Modal is a serverless cloud platform for AI and data teams. Run Python functions on GPUs, serve models as autoscaling endpoints and run agents in secure Sandboxes, billed per second.

Blurb, 150 words

Modal is a serverless platform for running compute-heavy Python in the cloud. Developers attach a decorator to a function and Modal runs it in a container, on CPUs or on any NVIDIA GPU from T4 to B200, scaling from zero to thousands of containers in seconds and billing per second of use. Teams use Modal to serve open-weight LLMs and speech and vision models, run AI agents and LLM-generated code in isolated Sandboxes, fine-tune and post-train models, and fan out batch jobs, without Dockerfiles, Kubernetes or YAML. Modal is used by companies such as Suno, Substack, Lovable and Ramp, and is free to start.

Luma description block

Compute partner: Modal. Every participant gets $100 in Modal credits for GPUs and CPUs. Serve models, run agents in Sandboxes, fine-tune, all in Python. Redeem at the link in the participant guide.

Social post

We are teaming up with @modal for Junction 2026: every builder gets $100 in GPU credits. Ship an AI demo with a real URL this weekend.

8. Support and FAQ

Modal docs, the Modal Community Slack and the examples repo. Contact Filip Makraduli at filip@modal.com. Docs: modal.com/docs. Community: modal.com/slack.

Do credits expire?

The code must be redeemed before its expiry date; the credits themselves last long after the event.

Can a team pool credits?

Credits belong to a workspace. Create one team workspace, redeem once there, and invite teammates (Settings -> Members).

Which GPUs can we use?

All of them, chosen in code with gpu="...". If a type is pending, pick another; no single type is guaranteed at a given moment.

Where do we get help?

Modal docs (modal.com/docs), the Modal Community Slack, the examples repo, and the contacts in this guide.

Generated 6 November 2026 by the Modal DevRel hackathon engine.