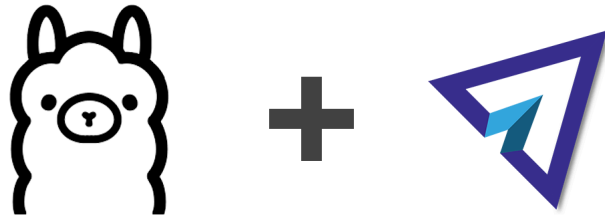


Ollama: Run quantized LLMs on CPUs and GPUs



[Ollama](#) is popular library for running LLMs on both CPUs and GPUs. It supports a wide range of models, including quantized versions of `llama2`, `llama2:70b`, `mistral`, `phi`, `gemma:7b` and many [more](#). You can use SkyPilot to run these models on CPU instances on any cloud provider, Kubernetes cluster, or even on your local machine. And if your instance has GPUs, Ollama will automatically use them for faster inference.

In this example, you will run a quantized version of Llama2 on 4 CPUs with 8GB of memory, and then scale it up to more replicas with SkyServe.

Prerequisites

To get started, install the latest version of SkyPilot:

```
pip install "skypilot-nightly[all]"
```

For detailed installation instructions, please refer to the [installation guide](#).

Once installed, run `sky check` to verify you have cloud access.



Ask AI

[Skip to main content](#)

If you do not have cloud access, you also can run this recipe on your local machine by creating a local Kubernetes cluster with `sky local up`.

Make sure you have KinD installed and Docker running with 5 or more CPUs and 10GB or more of memory allocated to the [Docker runtime](#).

To create a local Kubernetes cluster, run:

```
sky local up
```

► Example outputs:

After running this, `sky check` should show that you have access to a Kubernetes cluster.

SkyPilot YAML

To run Ollama with SkyPilot, create a YAML file with the following content:

► Click to see the full recipe YAML

You can also get the full YAML [here](#).

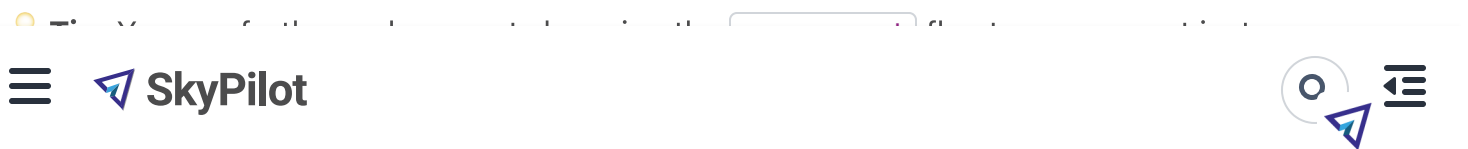
Serving Llama2 with a CPU instance

Start serving Llama2 on a 4 CPU instance with the following command:

```
sky launch ollama.yaml -c ollama --detach-run
```

Wait until the model command returns successfully.

► Example outputs:



```
sky launch ollama.yaml -c ollama --detach-run --env MODEL_NAME=mistral
```

[Skip to main content](#)

Ollama supports `llama2`, `llama2:70b`, `mistral`, `phi`, `gemma:7b` and many more models. See the full list [here](#).


Once the `sky launch` command returns successfully, you can interact with the model via

- Standard OpenAPI-compatible endpoints (e.g., `/v1/chat/completions`)
- [Ollama API](#)

To curl `/v1/chat/completions`:

```
ENDPOINT=$(sky status --endpoint 8888 ollama)
curl $ENDPOINT/v1/chat/completions \
-H "Content-Type: application/json" \
-d '{
  "model": "llama2",
  "messages": [
    {
      "role": "system",
      "content": "You are a helpful assistant."
    },
    {
      "role": "user",
      "content": "Who are you?"
    }
  ]
}'
```

► Example curl response:

 **Tip:** To speed up inference, you can use GPUs by specifying the `accelerators` field in the YAML.

To stop the instance:

```
sky stop ollama
```

To shut down all resources:

```
sky down ollama
```



If you are using a local Kubernetes cluster created with `sky local up`, shut it down with:

[Skip to main content](#)

Serving LLMs on CPUs at scale with SkyServe

After experimenting with the model, you can deploy multiple replicas of the model with autoscaling and load-balancing using SkyServe.

With no change to the YAML, launch a fully managed service on your infra:

```
sky serve up ollama.yaml -n ollama
```


Wait until the service is ready:

```
watch -n10 sky serve status ollama
```

► Example outputs:

Get a single endpoint that load-balances across replicas:

```
ENDPOINT=$(sky serve status --endpoint ollama)
```

 **Tip:** SkyServe fully manages the lifecycle of your replicas. For example, if a spot replica is preempted, the controller will automatically replace it. This significantly reduces the operational burden while saving costs.

To curl the endpoint:

```
curl -L $ENDPOINT/v1/chat/completions \
  -H "Content-Type: application/json" \
  -d '{
    "model": "llama2",
    "messages": [
      {
        "role": "system",
        "content": "You are a helpful assistant."
      },
      {
        "role": "user",
        "content": "Who are you?"
      }
    ]
  }'
```



[Skip to main content](#)

To shut down all resources:

```
sky serve down ollama
```

See more details in [SkyServe docs](#).

Previous

< [TGI: Hugging Face Text
Generation Inference](#)

Next

[SGLang: A Structured
Generation Language](#) >

© Copyright 2024, SkyPilot Team.

