



# Template repos

So, you've created a really cool chain - now what? How do you deploy it and make it easily shareable with the world?

This section covers several options for that. Note that these options are meant for quick deployment of prototypes and demos, not for production systems. If you need help with the deployment of a production system, please contact us directly.

What follows is a list of template GitHub repositories designed to be easily forked and modified to use your chain. This list is far from exhaustive, and we are EXTREMELY open to contributions here.

## Streamlit

---

This repo serves as a template for how to deploy a LangChain with Streamlit. It implements a chatbot interface. It also contains instructions for how to deploy this app on the Streamlit platform.

## Gradio (on Hugging Face) #

---

This repo serves as a template for how to deploy a LangChain with Gradio. It implements a chatbot interface, with a "Bring-Your-Own-Token" approach (nice for not wracking up big bills). It also contains instructions for how to deploy this app on the Hugging Face platform. This is heavily influenced by James Weaver's [excellent examples](#).

## Chainlit

---

This repo is a cookbook explaining how to visualize and deploy LangChain agents with Chainlit. You create ChatGPT-like UIs with Chainlit. Some of the key features include intermediary steps visualisation, element management & display (images, text, carousel, etc.) as well as cloud deployment. Chainlit [doc](#) on the integration with LangChain

## Beam

---

This repo serves as a template for how to deploy a LangChain with [Beam](#).

It implements a Question Answering app and contains instructions for deploying the app as a serverless REST API.

## Vercel

---

A minimal example on how to run LangChain on Vercel using Flask.

## FastAPI + Vercel

---

A minimal example on how to run LangChain on Vercel using FastAPI and LangChain/Uvicorn.

## Kinsta

---

A minimal example on how to deploy LangChain to [Kinsta](#) using Flask.

## Fly.io

---

A minimal example of how to deploy LangChain to [Fly.io](#) using Flask.

## DigitalOcean App Platform

---

A minimal example on how to deploy LangChain to DigitalOcean App Platform.

## CI/CD Google Cloud Build + Dockerfile + Serverless Google Cloud Run

---

Boilerplate LangChain project on how to deploy to Google Cloud Run using Docker with Cloud Build CI/CD pipeline

## Google Cloud Run

---

A minimal example on how to deploy LangChain to Google Cloud Run.

## SteamShip

---

This repository contains LangChain adapters for Steamship, enabling LangChain developers to rapidly deploy their apps on Steamship. This includes: production-ready endpoints, horizontal scaling across dependencies, persistent storage of app state, multi-tenancy support, etc.

## Langchain-serve

---

This repository allows users to deploy any LangChain app as REST/WebSocket APIs or, as Slack Bots with ease. Benefit from the scalability and serverless architecture of Jina AI Cloud, or deploy on-premise with Kubernetes.

## BentoML

---

This repository provides an example of how to deploy a LangChain application with [BentoML](#). BentoML is a framework that enables the containerization of machine learning applications as standard OCI images. BentoML also allows for the automatic generation of OpenAPI and gRPC endpoints. With BentoML, you can integrate models from all popular ML frameworks and deploy them as microservices running on the most optimal hardware and scaling independently.

## OpenLLM

---

OpenLLM is a platform for operating large language models (LLMs) in production. With OpenLLM, you can run inference with any open-source LLM, deploy to the cloud or on-premises, and build powerful AI apps. It supports a wide range of open-source LLMs, offers flexible APIs, and first-class support for LangChain and BentoML. See OpenLLM's [integration doc](#) for usage with LangChain.

## Databutton

---

These templates serve as examples of how to build, deploy, and share LangChain applications using Databutton. You can create user interfaces with Streamlit, automate tasks by scheduling Python code, and store files and data in the built-in store. Examples include a Chatbot interface with conversational memory, a Personal search engine, and a starter template for LangChain apps. Deploying and sharing is just one click away.

