Deploying PyTorch Models to Production

Janani Ravi

CO-FOUNDER, LOONYCORN

www.loonycorn.com

Overview

Deploy solutions to production

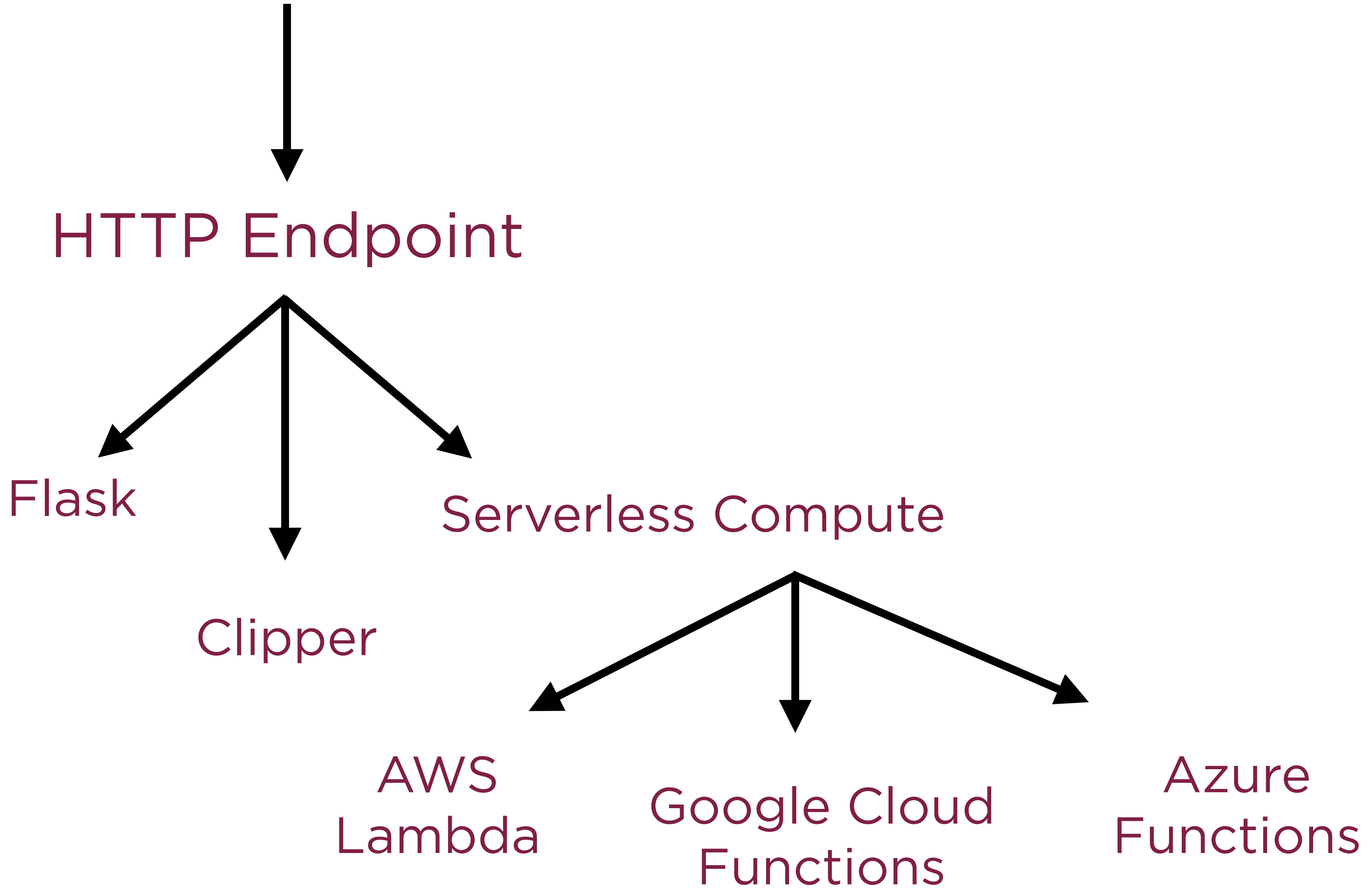
Deploy models for prediction using a Flask web application

Make models available using a Clipper cluster

Deploy to serverless environment using Google Cloud Functions

Deploying Models for Prediction

Deploying Models for Prediction



HTTP Endpoint

Flask

Clipper

Serverless Compute

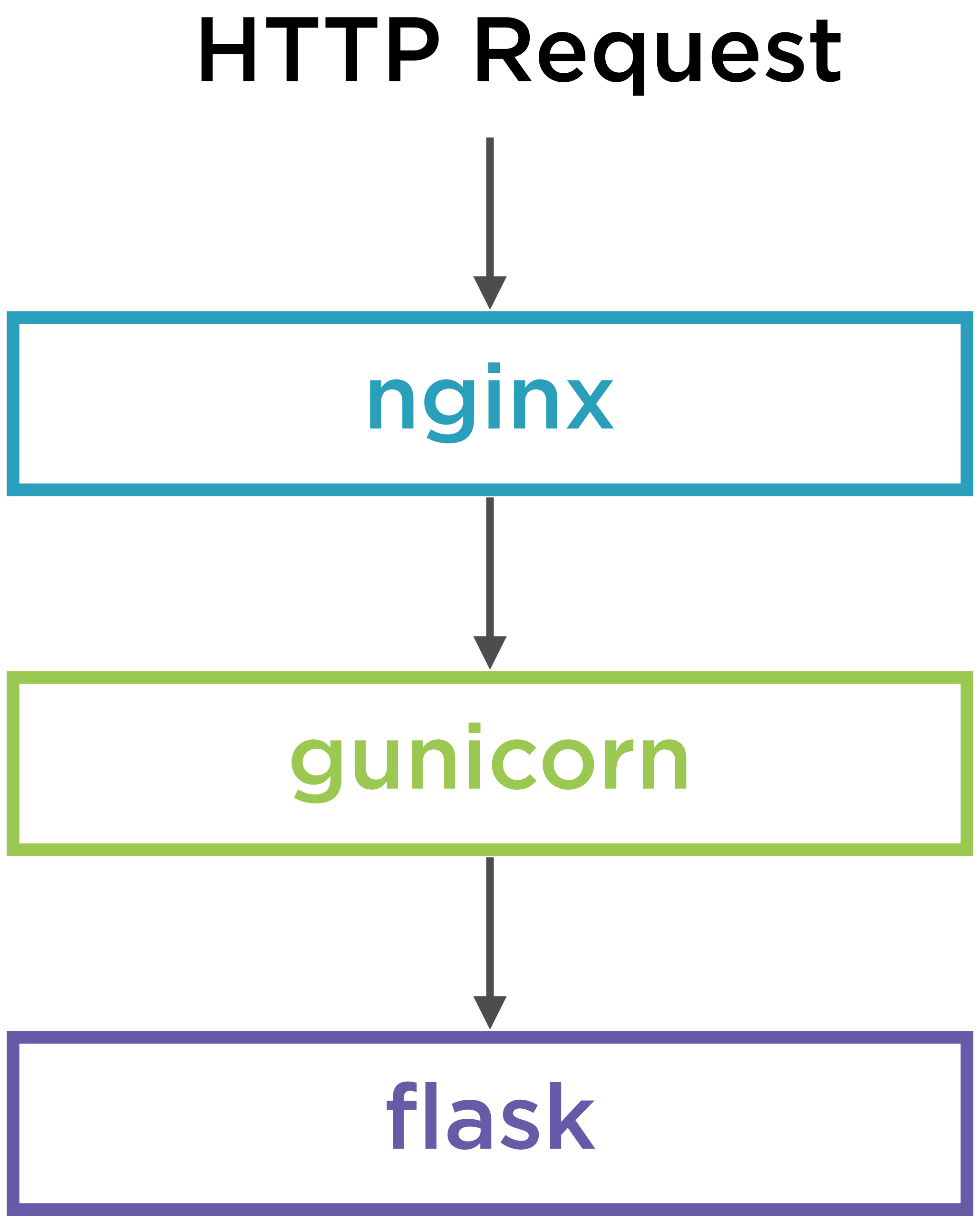
AWS Lambda

Google Cloud Functions

Azure Functions

Flask: Lightweight web framework for making models available as HTTP endpoints

Hosting



HTTP Request

nginx

gunicorn

flask

nginx

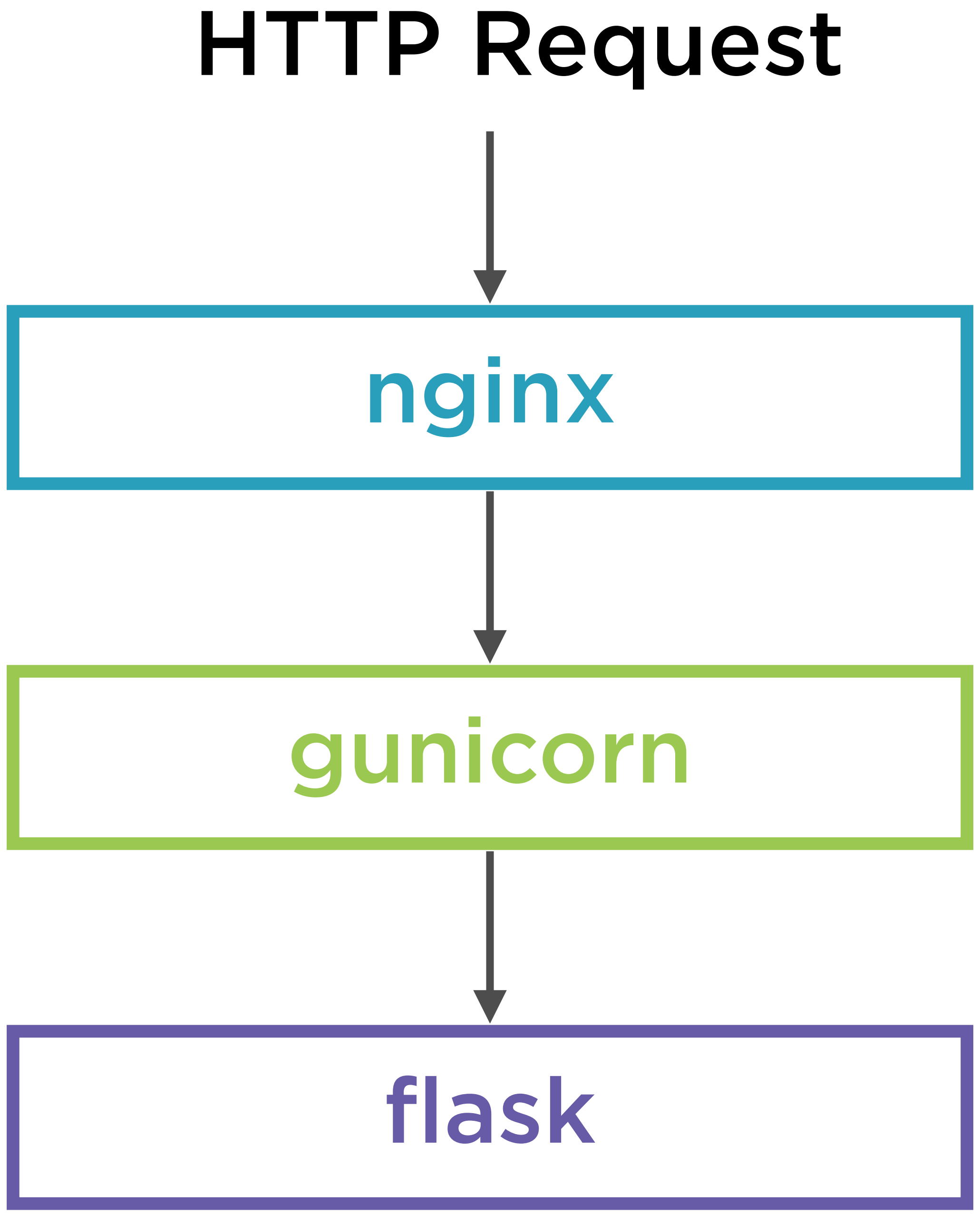
nginx

Open source software for web serving, reverse proxying, caching, load balancing

Reverse proxy:

- Sits behind a firewall and directs requests to the appropriate backend
- Additional level of abstraction between client and server

Hosting



gunicorn

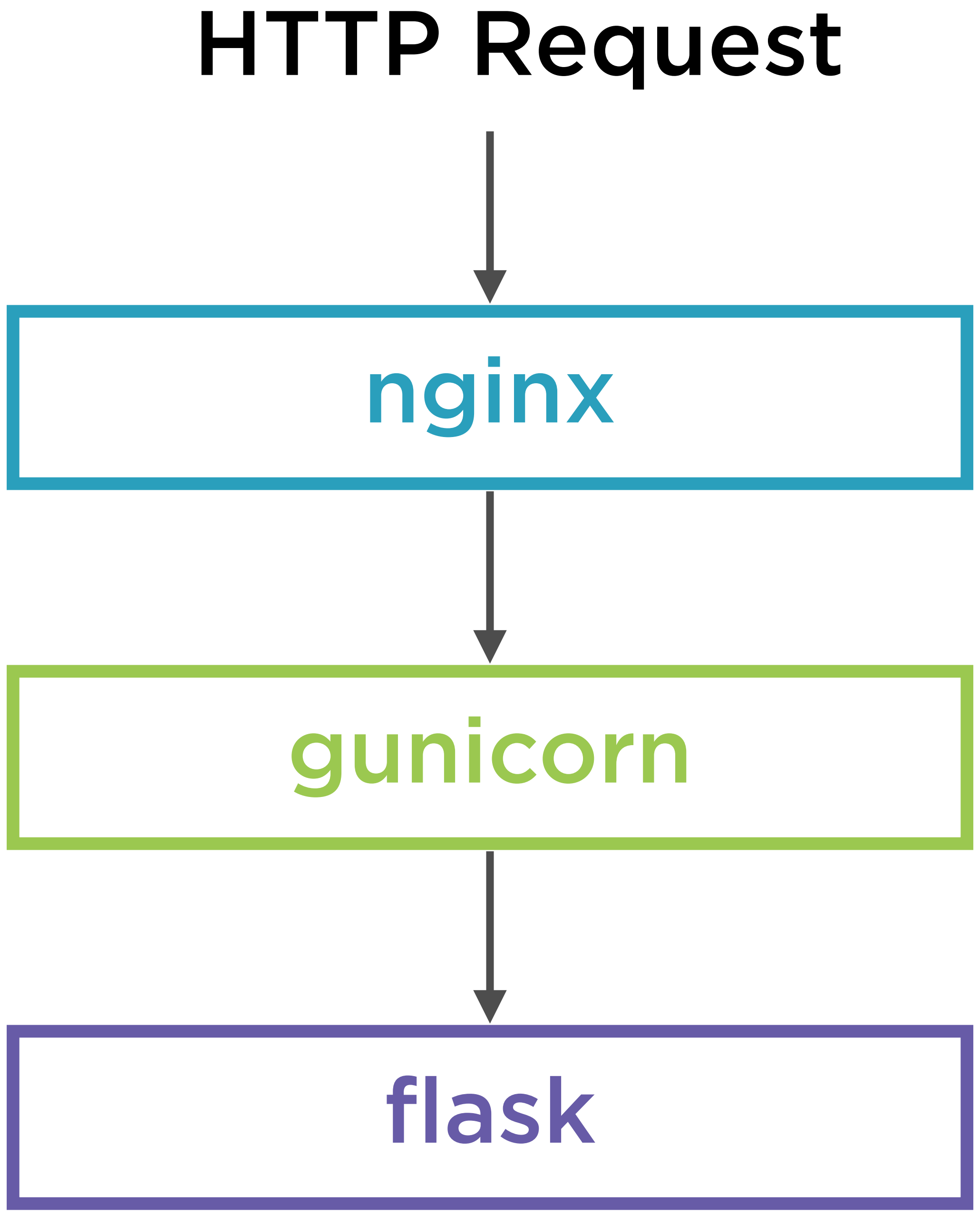
gunicorn

Web server for Unix

WSGI HTTP Server:

- WSGI (Web Server Gateway Interface) is a Python standard which determines how a web server communicates with applications
- Simple, lightweight, fast and works with many web frameworks

Hosting



flask

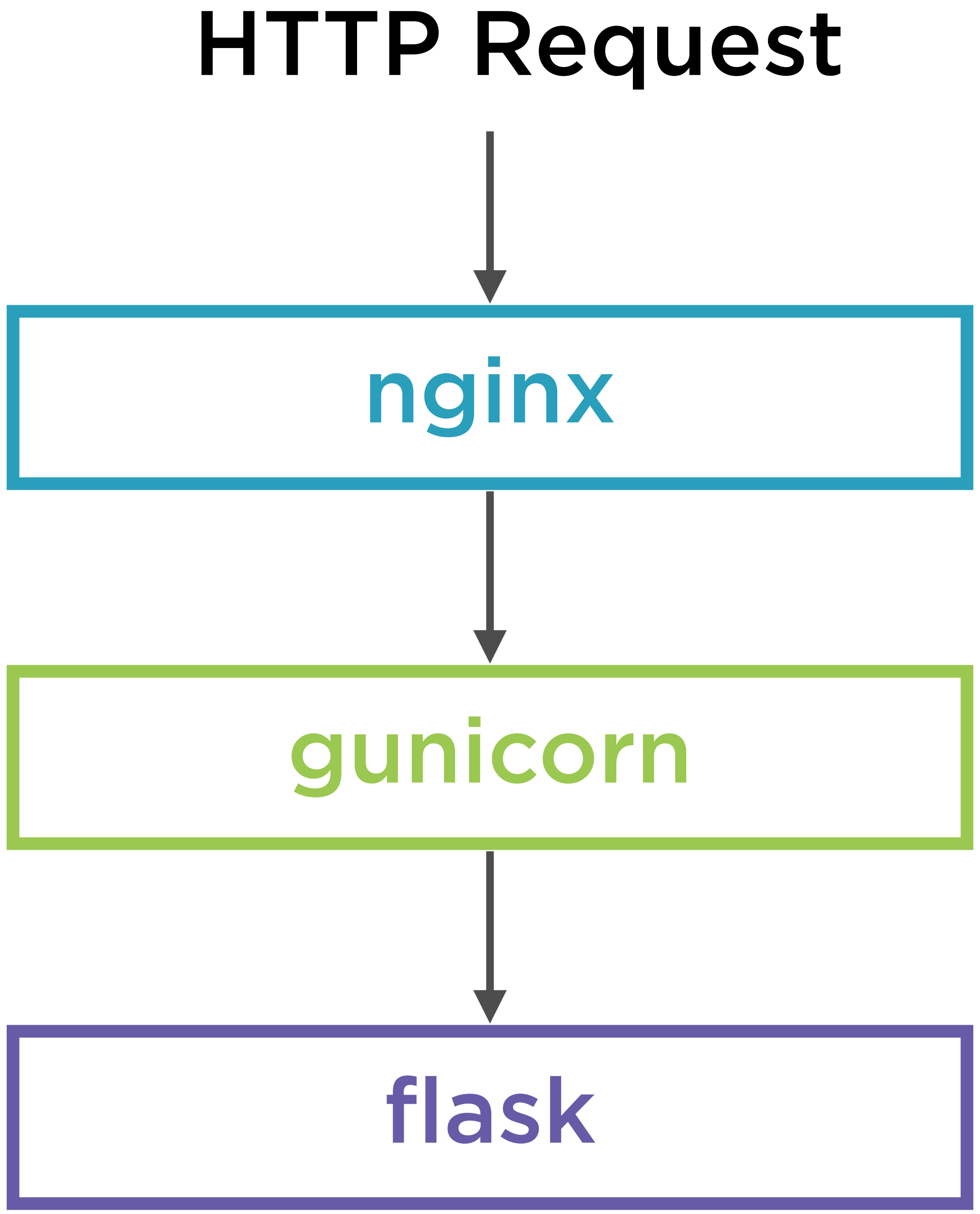
flask

Microframework for Python web app development

Worker:

- The actual instance of the application which hosts the inference code
- Loads the trained model and returns prediction results

Hosting



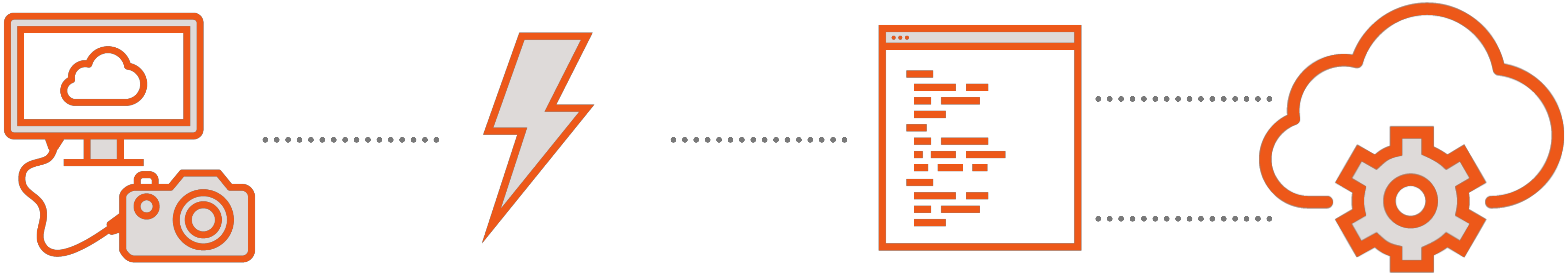
Clipper: Low-latency prediction serving system for ML models

Google Cloud Functions

Event-driven serverless compute service on the Google Cloud Platform

Serverless compute **abstracts away** provisioning, managing servers and configuring software

Event-driven Serverless Compute



Event Occurs

Platform triggers execution

Cloud Function code runs

Invokes other GCP services

Events

Occurs in the external environment

Functions can choose to respond to an event

Events are wired up to trigger functions

Demo

Deploy a trained PyTorch model using a Flask application

Demo

Deploying a PyTorch model to a Clipper cluster for low-latency predictions

Demo

Deploying a PyTorch model to a serverless environment i.e. Cloud Functions on the Google Cloud Platform

Summary

Deploy solutions to production

Deploy models for prediction using a Flask web application

Make models available using a Clipper cluster

Deploy to serverless environment using Google Cloud Functions

Related Courses

Using PyTorch on the Cloud: PyTorch Playbook

Expediting Deep Learning with Transfer Learning: PyTorch Playbook