Deploying Models

* Flask
  + Easily extensible
  + Modular
  + RESTful
  + Request-dispatching web application framework
* Heroku
  + Container-based
  + Platform as a service (PaaS)
  + Can deploy, manage, and scale applications

## TDS Blog

[TDS: Create an API to Deploy Machine Learning Models Using Flask and Heroku](https://towardsdatascience.com/create-an-api-to-deploy-machine-learning-models-using-flask-and-heroku-67a011800c50)

* Intro:
  + Flask:
    - a micro web framework that does not require tools or libraries to create web applications
  + Heroku:
    - is a cloud platform that can host web applications
  + Files required:
    - Model.pkl
    - App.py
    - Requirements.txt
    - Procfile
* Table of contents:
  + Create GitHub Repository
  + Create and Pickle a Model Using Titanic Data
  + Create Flask App
  + Test Flask App Locally
  + Deploy to Heroku
  + Test Working App

[YT: Deploying Python models to Production](https://www.springboard.com/workshops/data-science-career-track/learn#/curriculum/19349)

### High level overview

1. Gitlab: Code lives here
2. Jenkins: Does checkout + builds container
3. Docker: Container Engine
4. Kubernetes: Running/ Monitoring : Elk

* Gitlab: Gitlab unifies issues, code review, CI and CD into a single UI
  + Separate repository for each model
  + Features branches
    - When a feature is ready -> create a merge request
  + After merge
    - Master automatically deployed to dev
* Jenkins, why?
  + The leading open source automation server, Jenkins provides hundreds of plugins to support building, deploying, and automating any project
  + Since Jenkins 2.0, pipelines as code, e.g. a Jenkins file which describes your CI process
  + CI, CD = Continuous Integration, Continuous Deployment
* Jenkins
  + Sep