

Deployment Pipeline

It is the final step and most crucial and important step in end-to-end machine learning, deep learning or data science project. Before going for deployment, I created one web application using HTML, CSS, Flask and verified on my local machine. After that I proceed for the deployment. I have used two approaches to deploy. I used Heroku where I am very much comfortable and the second one is AWS EC2 instance. I am not that much familiar with AWS but I have tried to deploy my model on EC2 instance. The detailed information about both the approaches are discussed below.

1) Heroku Deployment:

Heroku uses platform as a service (paas). It means the user only need to take care of their model and code. All other things such as server, os, load, traffic, scale other services are taken care by the Heroku itself. It is easy to deployment. It works integrated with github. As shown in fig 1 first we have to store our api and their required supporting modules on github. After that we have to link the Heroku and that github repository where our all dependencies are stored. There are two option available for deployment in Heroku. One we can directly deploy as static. Where we can't get the updates of our deployed version if we change something in our dependencies. Other approach is live inferencing where it gets automatically deployed after committing the changes in the github. While deploying we need some dependencies like procfile and requirement.txt file to install all required packages. After successfully deployment it gives us free domain by including their company name in that domain. One can see logs for error and debugging. So, its easy if one can fix the errors otherwise it is very difficult to understand the erros because is uses paas as a cloud service. One can check the deployed app by using this url “https://absap.herokuapp.com/”.

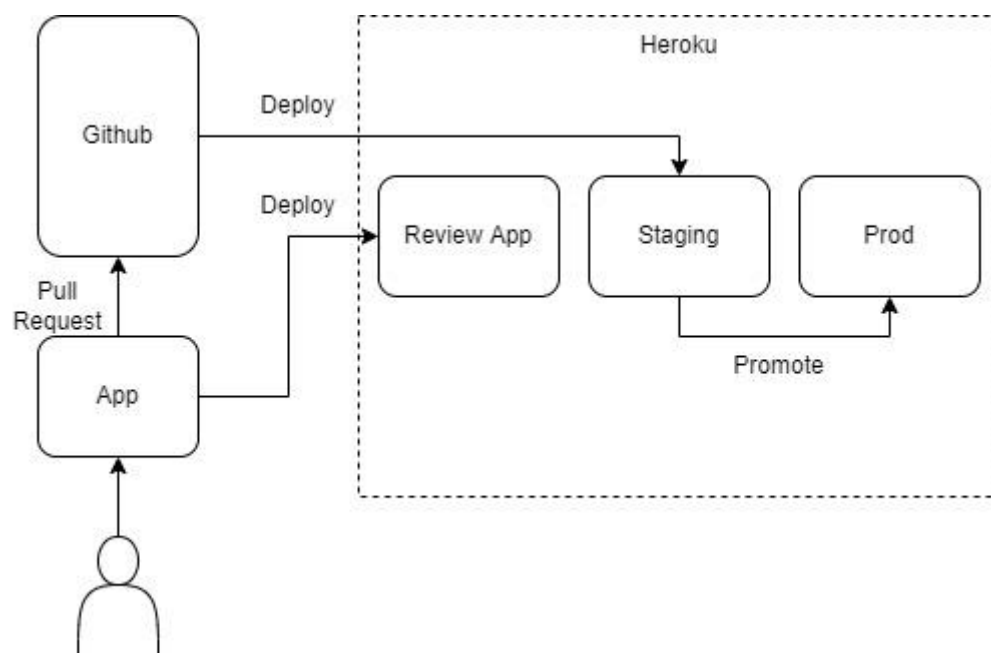


Fig 1: Deployment Pipeline

2) AWS EC2 Deployment:

I am new to AWS so I decided to use EC2 instance for model deployment. I have used putty, puttygen and winscp software for connecting the public domain of EC2 and our local running model. By establishing the connection, I have verified our model also works in EC2 instance. AWS uses infrastructure as a service (IAAS) cloud service. In this type of service, we have to take care of all the requirement except some services. I give the public access while deploying the model and checked on other laptops also and it works fine.

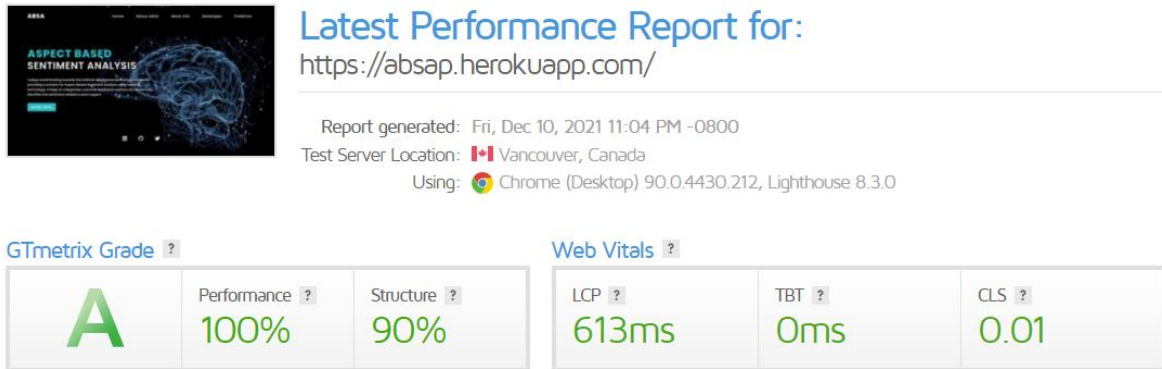


Fig 2: Performance Report of Web Application

Our web application URL is <https://absap.herokuapp.com/>. For checking the performance of our website we used GTmetrix. It is a website performance analytics tool, created by GT.net, who are well known in the digital space for the speed monitoring tool, Pingdom. The key objective of GTmetrix is to analyse the performance of website and provide a list of actionable recommendations to improve it. We, got “A” grade with 100% performance from GTmetrix that shows our web application works great in real time analysis and classification of aspect based sentiment analysis.