Liang Liu

Case study for Per Scholas: Deploy flask application

Tools: Virtual Box, Jenkins, GitHub, Docker, Dockerhub, Kubernetes(Minikube), Ansible.
GitHub: https://github.com/kg0529/2020_03_DO_Boston_casestudy_part_1
DockerHub: https://hub.docker.com/repository/docker/kg0529/flask

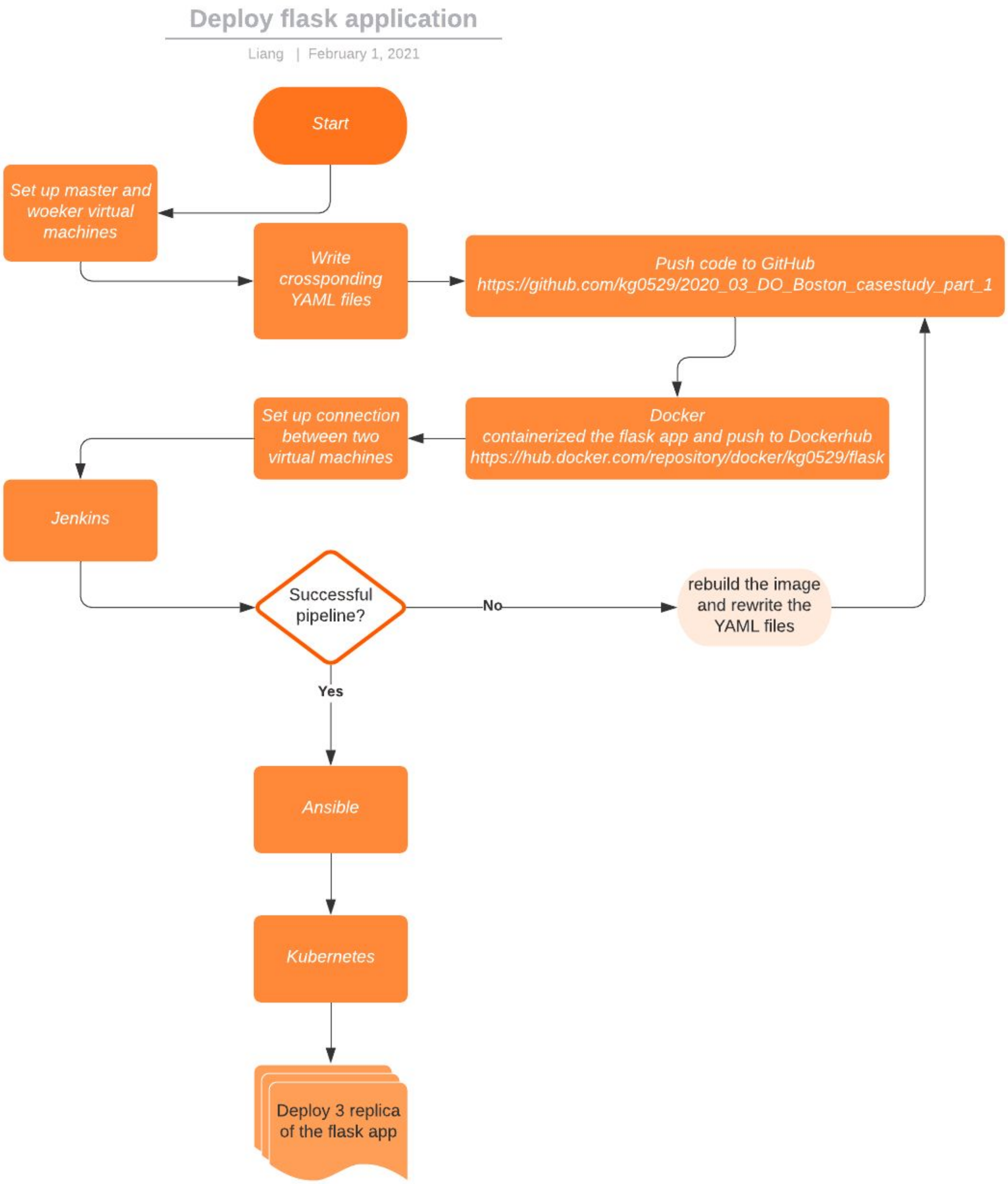
First step is to pull flask application code from the given github repo, making necessary modifications to containerize it and run it in a container as this is a necessary preparation step for the project. Write the proper Dockerfile and build the image with all necessary dependencies then test and run the flask app in a container using Docker, make sure it is up and running, then tag the image and push it to the dockerhub. And push all modifications of code to github, make sure everything is up to date.

Second step would be building the continuous delivery pipeline, Jenkins wound be the tool to build the pipeline. First test it by building a test pipeline by using Jenkins to tell docker to run the flask app by pulling the image from dockerhub which we pushed earlier. Making sure the flask app is up and running through this method. Making any modification if needed. This is just a test, making sure this pipeline would work for the actual project.

Third step, setting up two Virtual Machines, both in Linux environments, master machine needs to be desktop version, since we need GUI for jenkins, worker machine can be server version as everything should and would be in CLI . Install Ansible and Jenkins on the master machine, this machine would be commanding the worker machine to do the actual work. Install Docker and Kubernetes (minikube) on the worker machine, this is the machine that would be doing the actual work (running the flask app). Set up the both machines with other necessary tools such as SSH, ipconfig and etc. Set up the SSH connection between these two machines, so that we can use Ansible on the master machine to configure and run the task on the worker machine. Writing the corresponding YAML file for kubernetes to deploy the app, rebuild the image and push it dockerhub, test and run the kubernetes deployment, making sure it is up and running, then push all code to github.

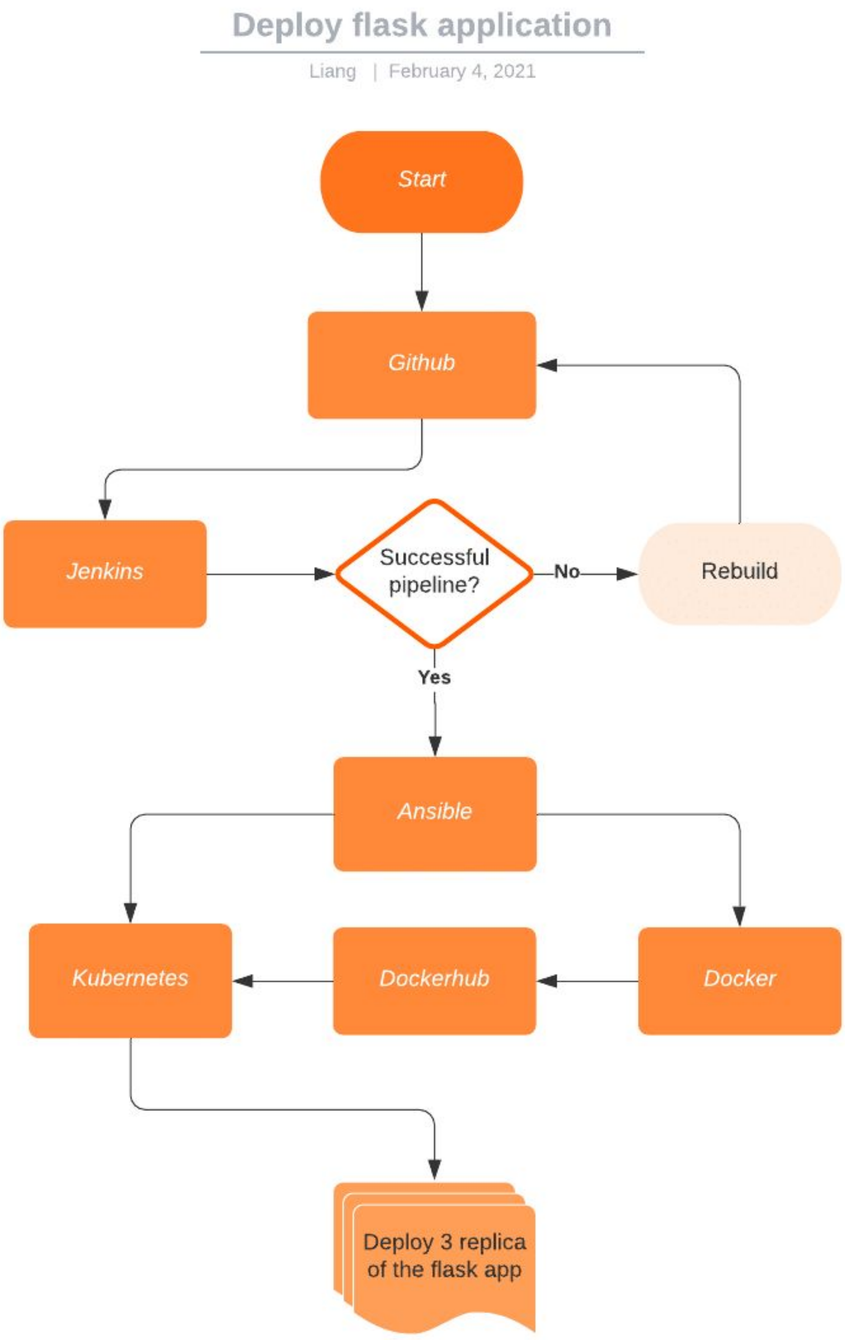
Fourth step, after all preparation steps are done, we have the necessary environment and tools on both machines. In this step, starting to build the actual pipeline for this project. On the master machine, using Jenkins to build a pipeline to command Ansible, Ansible would be configuring the worker machine and telling kubernetes to deploy the flask application. Writing corresponding YAML file for Ansible for this task. Save the pipeline in the Jenkins file. Pushing all modifications to github, Using Jenkins pipeline to start the deployment, using rolling deployment trigger on Github repository changes.

Old



Deploy flask application
Liang | February 1, 2021
Start
Set up master and woeker virtual machines
Write crossponding YAML files
Push code to GitHub https://github.com/kg0529/2020_03_DO_Boston_casestudy_part_1
Docker containerized the flask app and push to Dockerhub https://hub.docker.com/repository/docker/kg0529/flask
Set up connection between two virtual machines
Jenkins
Successful pipeline?
No
rebuild the image and rewrite the YAML files
Yes
Ansible
Kubernetes
Deploy 3 replica of the flask app

New


Deploy flask application
Liang | February 4, 2021
Start
Github
Jenkins
Successful pipeline?
No
Rebuild
Yes
Ansible
Kubernetes
Dockerhub
Docker
Deploy 3 replica of the flask app