Project Progress Report

Do-ploy: The Docker approach to a reproducible deployment

Zhufeng Xu (zx2245) / Mengyu Han (mh3881)

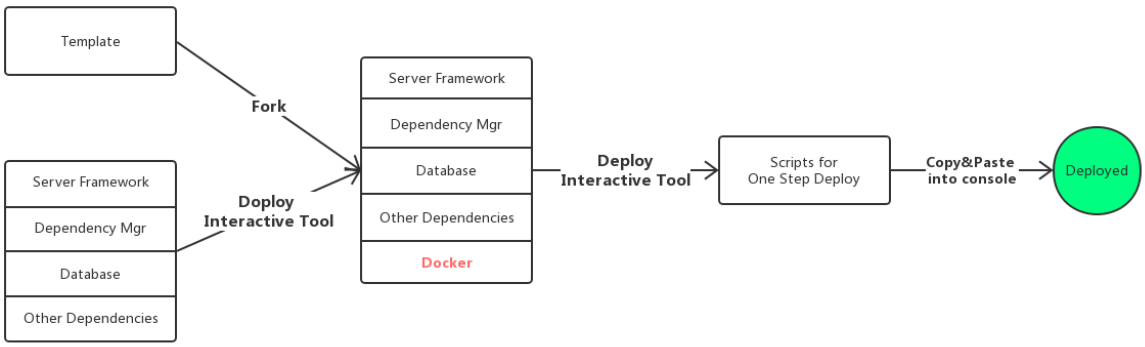
Core Value & Usage

The project is the implementation of the midterm paper. The ultimate goal is to build a Docker-based configuration tool that helps set up a deployment pipeline of the backend server for a web application in a faster and simpler way.

Our core value is simple and efficient deployment. It will encapsulate a traditional web-backend code base into a product that can be deployed to any machine within one step.

Users first select their favored configuration on the interactive website and then the tool will generate scripts that help build the environment. Further, the users only need to run the given scripts at any server from any online server providers (AWS, Azure, Google Cloud, etc.) or local machines with a Docker platform, and they will get the environment they want within minutes.

Current progress

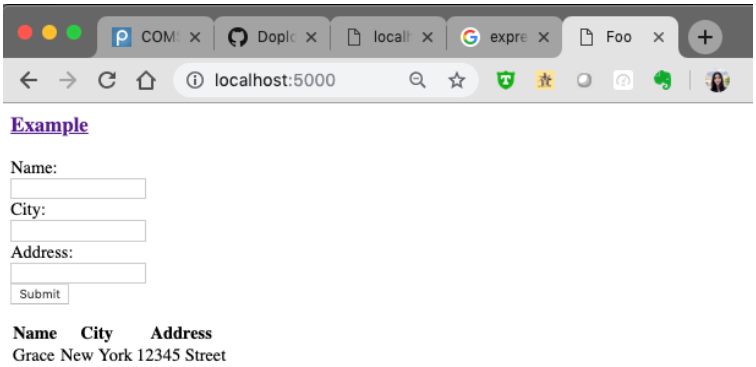


Following the original design for the project, we separated the development into 2 parts, template development (Mengyu is in charge) and auto generator (Zhufeng is in charge).

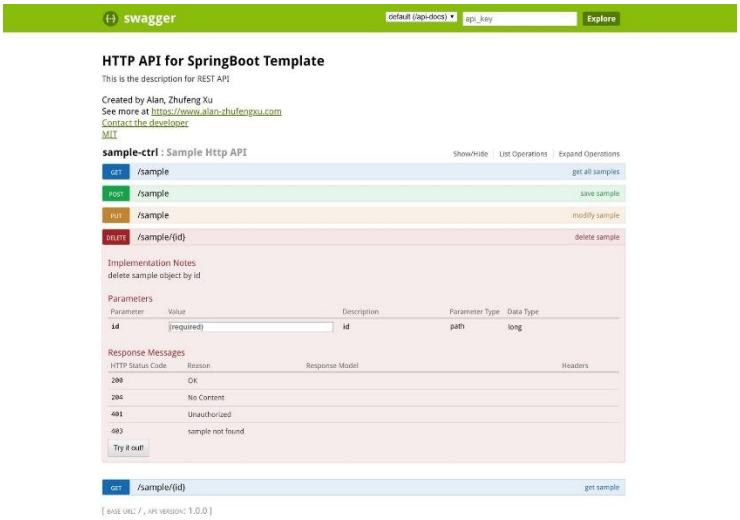
Templates work for greenfield projects (a work which is not following a prior work). You can easily find popular tech stacks for web backend here. Template development is the procedure starting from top left part in the graph. Users can fork templates with their favored tech stacks, and deploy them easily because templates themselves have already been featured with Docker related scripts.

Currently, we've done the following popular templates for users to choose from (since they're just the templates, the display of these demos can be pretty raw):

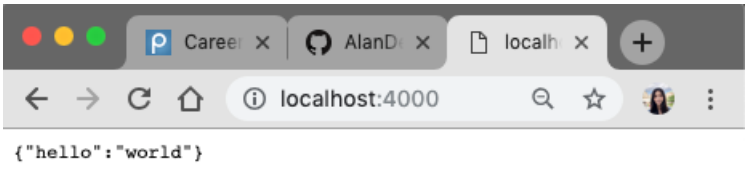
- Flask + PostgreSQL



- Springboot + MySQL



- Express + MongoDB



They can be found in different branches of Doploy repository.

Auto generator works for brownfield projects (a project based on prior work or to rebuild a product from an existing one). The procedure starts from bottom left part in the graph where developers have their existed projects to be renovated. Doploy generates a Docker layer (consists of multiple script files) to empower the original project with a deployable feature.

Currently, we've done a "Flask + PostgreSQL" auto generator. In this case, users can input some existed information about their projects, like "db name", "db username", "db password", "port", to get the customized Docker scripts. Copy these scripts to their projects. And then their projects will be empowered with deployable feature just like templates mentioned above.

It can be found in the master branch of Doploy repository.

Evaluation

Comparing with what we have on the market, like "*deployment pipeline best practice*" and "*Jenkins-Docker*", they're tools with learning barriers. It's not solving the problem of easy deployment which is the core value of our tool. For those comprehensive platforms like *AWS CodePipeline*, it may not be suitable for everyone to understand the whole ecosystem of an AWS deployment pipeline.

We can easily see that learning barrier is the biggest issue here. But Doploy works as a blackbox. Extra knowledge is not required.

Also note that the tool doesn't aim at building an online platform that holds management of users' projects. Instead, it's for the initialization of the project.

To evaluate our tool, we will conduct a survey inviting students in our department to use our tool. Feedbacks will be collected to evaluate Do-ploy. Here are some survey questions:

1. What server configuration and deployment experience do you have?
2. What are the reasons that you choose configuration and deployment tool?
3. Does Do-ploy help you achieve your goal?
4. Do you think Do-ploy is easy to use?
5. Do you think Do-ploy can help you save time in configuration and deployment?
6. Will you recommend this tool to your classmates?

Question 1 and 2 aim at categorizing user groups by evaluating users' familiarity of deployment and their potential scenarios; Question 3, 4 and 5 aim at evaluating Doploy itself, whether it achieves simplicity, usability we expect; Last question is about the general feedback of Doploy.

Code & Demo

All relative codes can be found in Doploy github repository.

The final demo will be presented in the form of website, which is the Doploy interactive tool in the paper and the design graph.