SE6156 Project Report

Doploy: The Docker approach to a reproducible deployment

Zhufeng Xu (zx2245) / Mengyu Han (mh3881)

Core Value

Here we explain what is unique about Doploy and what value it creates.

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.

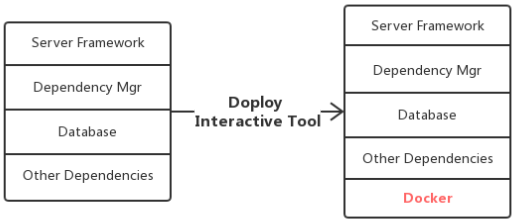


Fig 1. Doploy encapsulation

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

Doploy serves specifically for CS students and TAs and emphasizes a lot on simplicity. The value comes from the requirements of some application based courses(*COMS 4111 Intro to Database*, *COMS 4995 sec4 Intro to Data Visualization, COMS 6998 sec6 Intro to Agile Project*) where students struggled to deploy their codes into servers even though the focus of the course is not deployment.

Usage

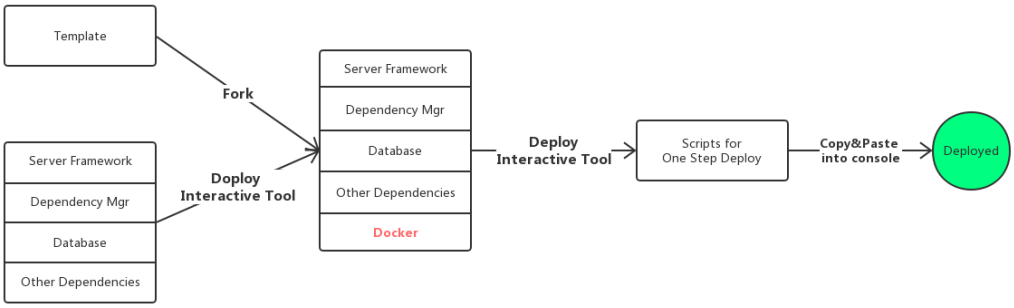


Fig 2. Doploy workflow

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.

We have adopted mainstream "framework + database" combinations, such as *Flask + PostgreSQL, Express + MongoDB, Springboot + MySQL, and etc*. Users can configure the projects with their favored backend combinations.

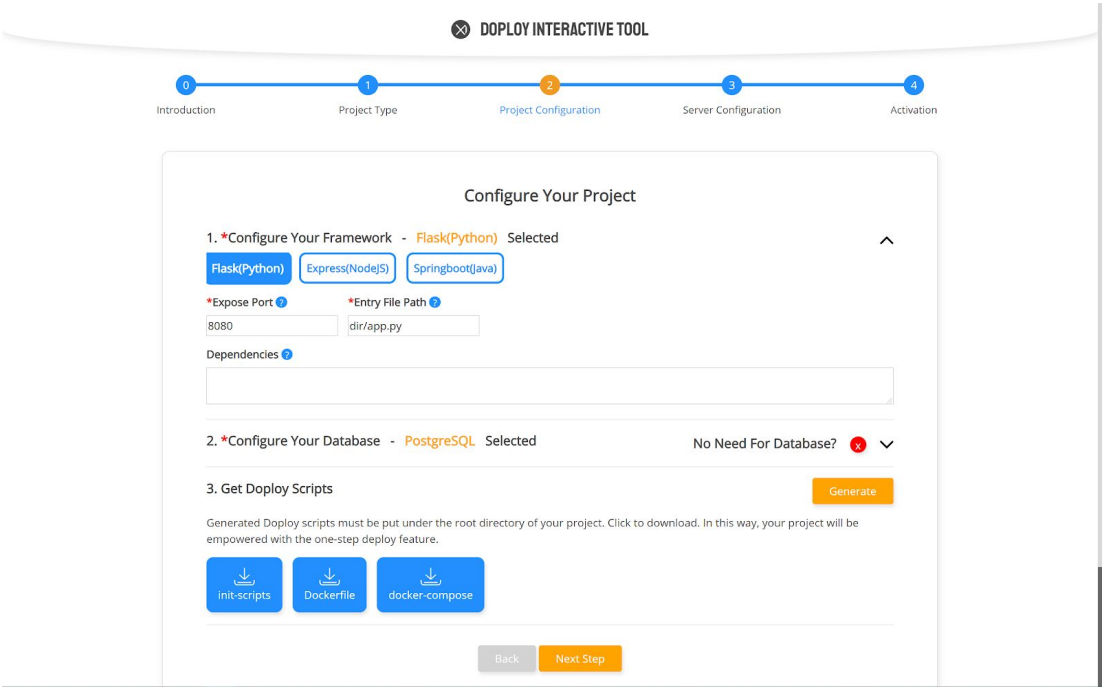


Fig 3. Doploy website

Doploy works in 2 scenarios

- Templates: uniformed templates for TAs to deliver starter codes

- Customized project: generate scripts that empower your project with the one-step deploy feature.

In both cases, users only need to select or input related information about the projects. This allows you to focus more on your project instead of the complicated deployment configuration. Also note that though Doploy is using Docker, it works as a black box, any knowledge outside the range of your project setting is not required.

Research Question and Evaluation

Our research question is to compare Doploy with different deployment tools. We listed and made trials on several approaches for comparison, including virtual machine, oneinstack script tool, Occam, and AWS CodePipeline. The evaluation will be focused on capability(functionality), compatibility(whether it can work under different environments), flexibility(whether it's extensible), and usability(whether it has a learning barrier) of the approach.

Framework/Tool	Capability	Compatibility	Flexibility	Usability
Virtual Machine	**excellent**. VM can almost do everything if it's prepared well, packaging all dependencies together with a suitable OS.	**fair**. While VM is compatible in many scenarios, it's not feasible in some OS that doesn't allow for virtualization.	**poor**. The basic configuration of a VM is set once it's created. Also, the size of the VM is significantly larger than other approaches.	**excellent**. Basically, a VM has already done every setup for users
Oneinstack	**excellent**. Technically, scripts can do everything if you're familiar with them.	**excellent**. The codes are written in Shell language, resulting in a high compatibility	**excellent**. Small size. Extra scripts can be easily added.	**poor**. High learning barrier. Require learning of every tool. Lack of GUI, which requires users to interact directly with terminal.
Occam	**fair**. For research field, it's an excellent tool with full functionality. But when it comes to more general cases like web development, Occam shows its limit.	**excellent**. When creating the workflow, Occam allows you to choose the tool that is the most suitable. Compatible with most of the build and deploy methods.	**fair**. Occam is amazing at sharing research projects. Also, it has a version control system. What's wrong here is about its closed ecosystem. Every tool we use, the build, integration, deploy tools, are all pre-set in the system. And the projects themselves run in the Occam platform.	**fair**. It takes some time to understand the structure and make use of it.
AWS CodePipeline	**excellent**. As mentioned, one of the ultimate industrial solution. Technically capable of doing anything.	**excellent**. Again, it's technically compatible with any case since it allows for a third-party tool to integrate as long as you set up the correct connection	**excellent**. Truly flexible with every stage pluggable. Add, attach, or detach different stages within minutes.	**fair**. Although it has an excellent visualized dashboard that helps a lot in establishing the pipeline, it still costs time to learn the notion of the pipeline, like how each stage is standardized and how they can be connected.
Do-ploy	**excellent**. Combine the scripting approach with Docker. Technically capable of doing anything.	**fair**. Compatible with all mainstream environment. In some cases where there are some private dependencies that can not be found on the DockerHub, we may need to add the dependencies manually to build a customized image.	**excellent**. Small size. Configurations can be easily added or modified in the DockerFile	**excellent**. The progressive way of building the tool helps a lot to reduce the learning barrier. For normal users who don't care about how these dependencies work, a simple installation has zero barrier. While for advanced users, Do-ploy also has interfaces exposed like AWS CodePipeline.

Other Docker-based tools like "deployment pipeline best practice" and "Jenkins-Docker", they also have fairly high learning barriers of learning the Docker scripts.

Doploy does well in average and works the best with respect to usability, hence its core value of simplicity.

User Study

Doploy is a tool designed to assist with more efficient server configuration and course project deployment, therefore the target user are those students in the Engineering Department of Columbia

Doploy - Project Report - 05/03/2019

University, which can be divided into two groups,

1. Students especially those who take courses that require them to deploy web project to Cloud Platforms.
2. Teaching Assistants that want to generate a project template for the course project.

Recruitment Forum

We recruited students by posting a link to our project on Piazza, and we recruited teaching assistants by personal invitation.

Details About Users

There were a total of seven people participated in our user study, with three teaching assistants (from both sections of COMS 4111 Introduction to Database) and four students who have taken these courses (COMS 4156 Advanced Software Engineering and COMS 4111 Introduction to Database).

What participants did

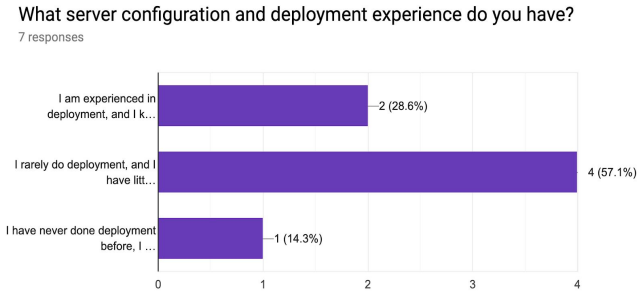
Participants were asked to do the following tasks:

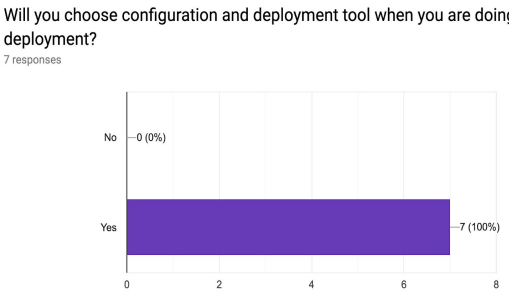
1. Create a new instance on Google Cloud Computing Engine Platform (Link: https://console.cloud.google.com/compute).
2. Open Doploy website and select the backend template and database type they wanted to use.
3. a. If users choose to use the standard project template (project generated by Doploy), they fork Doploy's template project to their own GitHub repositories.
   b. If users choose to use customize the project, they download the scripts generated by Doploy and add to the root directory of their project on Github.
4. Copy the Github link of their project to Doploy's website.
5. Copy the command generated by Doploy to the terminal of Google instance and execute the command.
6. Visit the IP of the instance and test the deployed project.
7. Complete the survey form (Link: https://docs.google.com/forms/d/e/1FAIpQLScwLDwzxLPTRSX5_4RrYmELrbiYy8Qhj7Y5vLBuKeYdYLG2Eg/viewform?usp=sf_link).
8. Give suggestions on Doploy.

* 3. a was used by five of our participants.
* 3. b was used by two of our participants.

Novel Findings in the Study

1. Almost all type of users need tools to help with server configuration and project deployment



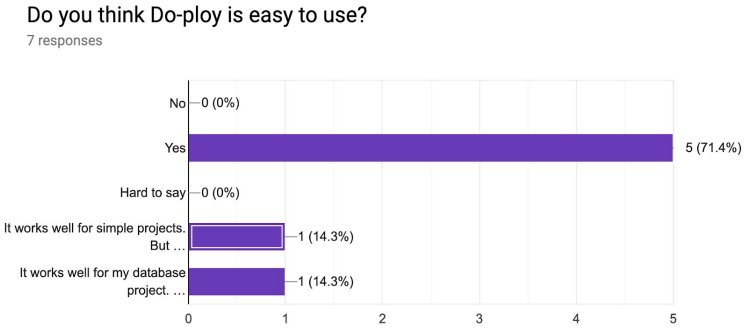


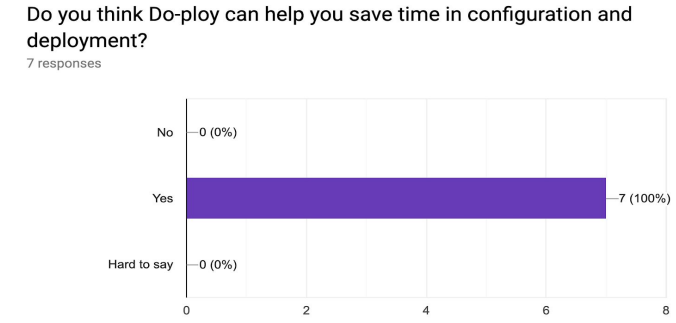
Doploy - Project Report - 05/03/2019

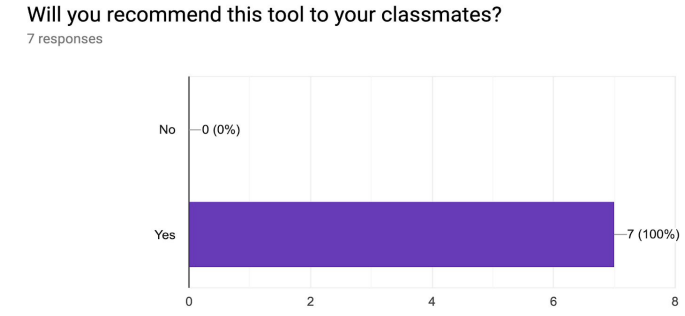
According to the responses to our first two survey questions, we found that no matter how much experience that users have in deployment, they still want to accelerate their server configuration and deployment process by using tools like Doploy. That is contradictory to our assumption, which was that experienced were less likely to use tools to help with the server configuration and deployment process.

2. Users prefer to customize their project instead of just fork the template.
Although users like Doploy website, we received similar suggestions from half of them on extending the customizability of Doploy, which means they prefer to customize the project instead of using the template generated by Doploy. That may be because users will only turn to deployment tools after they finish the development of their web project, which is too late to change the structure of their project to the standard one.

Conclusion







Simplicity of Doploy: According to the response of the participants, more than 70 percent users think Doploy is easy to use, while others think Doploy will be difficult to use when their project becomes

complex and has many dependencies. Considering the fact that Doploy is designed as a tool especially for small web project deployment, we think the problems are a not a general case, and we can reach the conclusion that Doploy is simple to use for students and TAs in Columbia Engineering.

Efficiency of Doploy: All of the seven participants finished the deployment process (task 1 to task 6) in ten minutes, and their response in the form shows that they think Doploy can help shorten the process of server configuration and deployment.

Work Division

Zhufeng Xu

- Design of Doploy framework
- Implementation of Customized scripts generator module
- Implementation of Doploy interactive website

Mengyu Han

- Implementation of templates module, 9 possible combinations in *Flask/Express/Springboot* and *MySQL/PostgreSQL/MongoDB.*
- User study and research questions

What We've Learned

Zhufeng Xu

During the process of building Do-ploy, I learned a lot about Docker, as well as Shell and Linux knowledge. What's more important is the decoupling and recombination notion in the design process. Docker allows me to decouple different dependencies into different containers and then connect them using a virtual network. This new way of resolving dependencies provides Doploy with much more potentials in building applications of different configuration combinations.

Mengyu Han

I mainly learned two of this project. First, by writing Docker file as well as generating template projects, I learned how Docker works and how a container is different from a typical virtual machine. Second, I learned how to design a user survey and how to organize a user study. Actually, I think there is more information that we could have collected by the survey, and I will be more aware of how to design the user study questions next time.

Link

Github Repository: https://github.com/AlanDelip/Doploy/

Doploy Interactive Website: https://www.do-ploy.com/doploy/

*Doploy will still be an active project after the course, keep track on what we're doing!