GitHub Continuous Deployment Survey

Welcome!

We are studying how people use Docker-enabled Continuous Deployment technologies in their GitHub projects, and the decisions and tradeoffs that they have to make on which tools to use and how to orchestrate them together. Our goal is to identify pain points and distill best practices that will hopefully help many other open-source projects as well.

Your participation is voluntary and confidential, and is expected to take no more than 10 minutes.

Thank you for participating!

* Required

1. **Do you consent to participate in this survey?** *
   Mark only one oval.

   - ◯ Yes
   - ◯ No *Stop filling out this form.*

Background Information

2. **How many years of experience do you have in open-source software?** *

   _____

3. **How many years of experience do you have with continuous integration / continuous deployment?** *

   _____

The remaining questions are specific to a GitHub project.

* Because we are collecting Docker-enabled answers, if you choose your Docker-related project will be very helpful to us.

4. **Which project (owner/repo) of yours will your answers below refer to?**

_____

5. **What is the application domain of this project?**

_____

Current Pipeline

6. **Why do you use continuous deployment (CD)?**

_____

_____

_____

_____

_____

Based on our previous survey, we found two CD workflows：
(1) Docker-Hub automated builds workflow: Code change/Commit-->Tigger Docker-Hub Auomated builds (https://docs.docker.com/docker-cloud/builds/automated-build);
(2) CI Docker workflow: Code change/Commit-->Trigger CI tool to build and push images (e.g. Using Travis CI, https://docs.travis-ci.com/user/docker;
Using Circle CI, https://circleci.com/docs/1.0/docker;
Using Jenkins, https://www.cloudbees.com/continuous-delivery/jenkins-docker).

7. **Which workflow do you use in your current CD pipeline?** *

Mark only one oval.

- ◯ Docker-Hub automated builds workflow
- ◯ CI Docker workflow *Skip to question 10.*
- ◯ Other workflow *Skip to question 14.*

Current Pipeline

8. **Why do you use Docker-Hub automated builds?** *
   Check all that apply.

   - [ ] Deploy more frequently
   - [ ] Increase confidence in build quality and results
   - [ ] Reduce the time spent on setting up how to build and push images
   - [ ] Create more visibility into team workflow via mechanisms like logs and dashboards
   - [ ] Allow higher flexibility of builds
   - [ ] Convenient custom settings and modifications
   - [ ] Less CD processing (build, push, etc.) latency
   - [ ] Other: _____

9. **Has your CD pipeline always looked like this?** *
   Mark only one oval.

   - ◯ Yes *Skip to question 19.*
   - ◯ No *Skip to question 17.*

Current Pipeline

10. **Which CI tool(s) do you use in your CD pipeline?** *
    Check all that apply.

    - [ ] Travis CI
    - [ ] Circle CI
    - [ ] Jenkins CI
    - [ ] Other: _____

11. **Why do you use the CI Docker workflow to build your CD pipeline?** *

Check all that apply.

- [ ] Deploy more frequently
- [ ] Increase confidence in build quality and results
- [ ] Reduce the time spent on setting up how to build and push images
- [ ] Create more visibility into team workflow via mechanisms like logs and dashboards
- [ ] Allow higher flexibility of builds
- [ ] Convenient custom settings and modifications
- [ ] Less CD processing (build, push, etc.) latency
- [ ] Other: _____

12. **If you use multiple CI tools in your current CD pipeline, what are their respective jobs? how do they work with each other?**

_____

_____

_____

_____

_____

13. **Has your CD pipeline always looked like this?** *

Mark only one oval.

- ◯ Yes *Skip to question 19.*
- ◯ No *Skip to question 17.*

Current Pipeline

14. **What does your CD workflow consist of at the moment?** *

_____

_____

_____

_____

_____

15. **Why do you use this workflow to build your CD pipeline?** *

Check all that apply.

- [ ] Deploy more frequently
- [ ] Increase confidence in build quality and results
- [ ] Reduce the time spent on setting up how to build and push images
- [ ] Create more visibility into team workflow via mechanisms like logs and dashboards
- [ ] Allow higher flexibility of builds
- [ ] Convenient custom settings and modifications
- [ ] Less CD processing (build, push, etc.) latency
- [ ] Other: _____

16. **Has your CD pipeline always looked like this?** *

Mark only one oval.

- ◯ Yes *Skip to question 19.*
- ◯ No

Pipeline Evolution

17. **What tools / workflow did you use before this for your continuous deployment pipeline?** *

18. **What issues did you or your team face when you used the old CD pipeline? Why did you switch to your current pipeline?** *

Containers

19. **What role do Docker containers play in your CD pipeline, if any?**

20. **What role does Docker Hub play in your CD pipeline, if any?**

Latency concerns

21. **Have you noticed your CD processing latency worsening over time? (e.g. the latency of each build gradually increases over time)** *

Mark only one oval.

- Yes
- No

22. **Do you plan to change your workflow/tools you use because of the increasing latency?**

Mark only one oval.

- Yes
- No

Pain Points

23. **Is your current CD pipeline sufficient for your needs?** *

Mark only one oval.

- Yes
- No

24. **What are currently the main pain points and how would you improve things?**

_____

_____

_____

_____

_____