
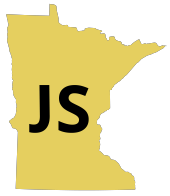



CONTINUOUS INTEGRATION
for web. for mobile. for fun. for profit.

WHAT WE'LL COVER

- 🙋 What CI is, why you'd want to use it
- 🆚 The distinction between CI and CD
- ❓ ♀ Why CI is so darned tricky
- 🙏 One way you can try out CI for yourself
- 📝 A potential thing that might be available for you to use!






WHO AM I?

- Brandon Johnson
- a research software engineer at 150
- I help out with  and 
- I consume lots of , and would be happy to chat about CI/CD, React Native, or other topics irl over a macchiato or something
- I placed contact info at the far side of this deck

WHAT IS CI/CD?

What is Continuous Integration?

A CI system will automatically*:

1.  **PULL** your latest code from source control
2.  **CONFIGURE** a clean testing environment
3.  **RUN** a build
4.  **REPORT** any failing integration tests or steps, or
5.  **REPORT** completion and describe any artifacts generated

“INTEGRATION STEPS”

Integration steps include any checks or specifications you use to verify that the code being tested is ready for production.

Can include:

- Unit tests, functional tests, integration tests
- Linters
- Other things important to you or your project

** You may have a different def. of “integration steps” or “integration tests” than I. That’s ok. I hope we can still be friends.*

BENEFITS TO CONTINUOUS INTEGRATION

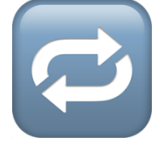

- 🧐 No individual developer is in charge of making sure you're ready to deploy your code
- 🏠 Any linting, testing, or code quality checks are enforced project-wide
- 🚨 Learn about code conflicts and build problems earlier and oftener
- 🚀 Be “ready to deploy” at any time

CI vs. CD?

CONTINUOUS INTEGRATION vs. CONTINUOUS DELIVERY

- Continuous Delivery services *deliver* 🍕 your *build artifact* 📁 to some *environment* 🌐 after its *integration steps* complete ✅
- Most CI services also provide delivery or deployment services nowadays!

BBBBBBBONUS ROUND!

- “CD” can also stand for **Continuous Deployment**, as in releasing the latest version of your code to a *production environment on every commit.* 
- Sound intense? It can be! This “CD” is less about the product that does the deploying, and more about *the way your team works* 

CI CAN BE TRICKY

WHY IS CI TRICKY?

What do you do when:

- 🤔 You don't have any tests?
- 🏛️ You have a monolith (multiple projects, one repo)?
- 🚰 You need to work with hosted GitHub Enterprise/a task tracker/
your boss' kitchen faucet?

WHY IS CI TRICKY?

- 🏢 The infrastructure (*what service you use*) is complex to configure
- ✈️ Your build plans (*how the service tests/delivers your software*) are also complex to configure
- ❓ Unless the infrastructure is already there, picking a CI service/tool can be overwhelming

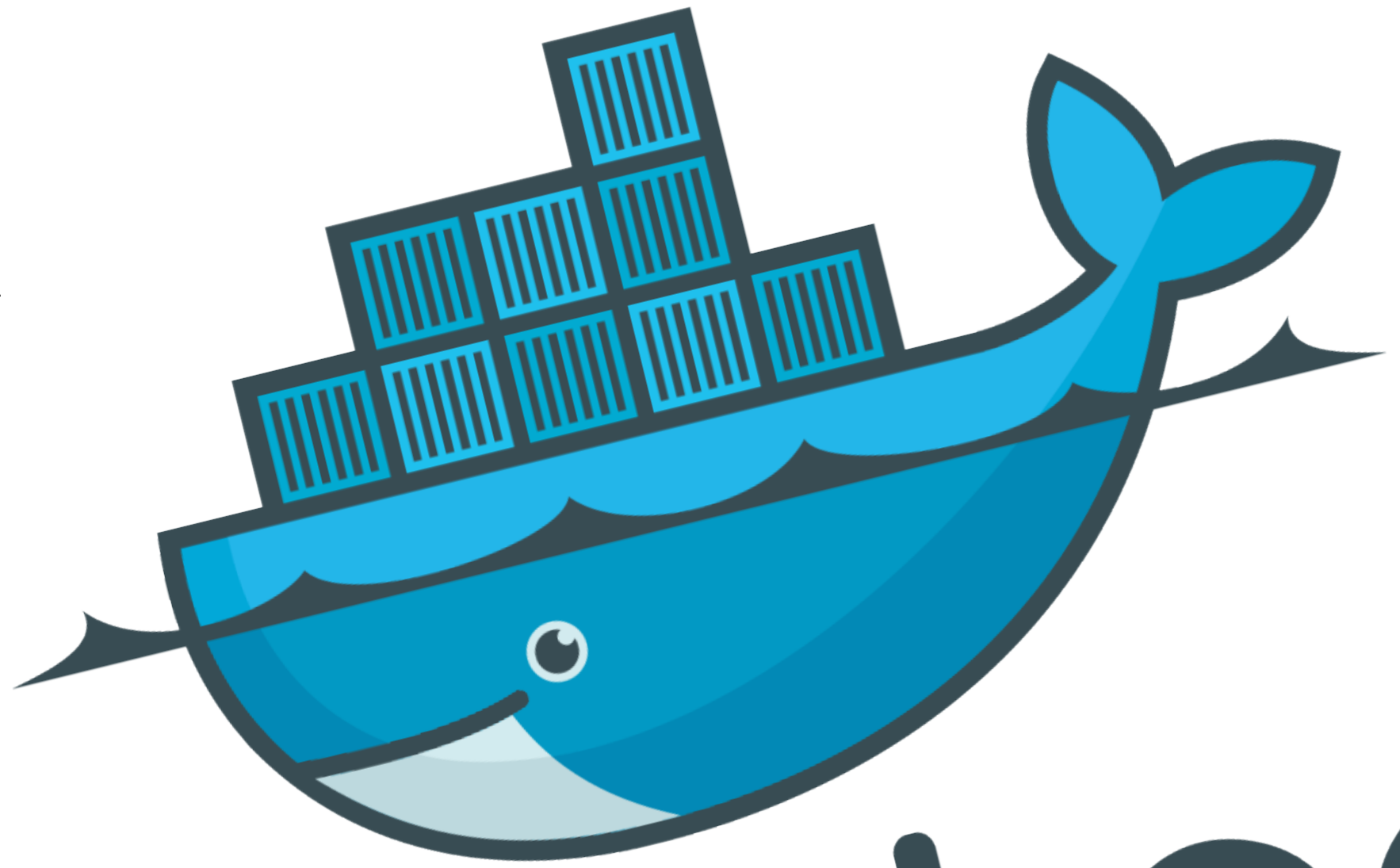


CODESHIP



Jenkins

?????



docker







circleci



START SIMPLE.

LIVE DEMO: CI for a basic web project

- For the purposes of the following demo, we'll be using GitLab CI 
- There are a bunch of other options! GitLab is nice because:
 -  It's free
 -  It has a lot of neat templates
 -  It has a nice interface that's conducive to demonstration
- Your company may have an existing setup that doesn't use GitLab, and that's ok!

LET'S HOP TO IT!

[Interactive demo: from CD to CI and CD](#)

[Full-featured CI and CD example](#)

LINKS + CONTACT

- @brandon in the Mpls Jr Devs slack,
- @brandon_mn on Twitter, @brandon@mastodon.xyz on Mastodon
- @skylineproject on GitHub
- These slides are on brandon.mn/slides/jrdevs-ci.pdf