



Introducing DevOps Techniques in a Software Construction Class

Mark Hills (@hillsma on Twitter, <http://www.cs.ecu.edu/hillsma/>)
Associate Professor, Department of Computer Science
East Carolina University (Greenville, NC, USA)

32nd IEEE International Conference on Software Engineering Education & Training (CSEE&T 2020)
November 11, 2020
Munich, Germany (Virtually!)

Context: MS in Software Engineering at ECU

- Graduate program offered at ECU for more than 10 years
- Students can attend either on-campus or online
- Mixture of students: some just earned at BS in CS, some have professional experience, some have a BS in another area and are switching careers

Context: Software Construction

- Originally part of software development track
- Focus is on scaling up development skills for larger software systems
- Important concepts: abstraction, code comprehension/understanding, professional practice
- Lots of non-cloud skills: IDEs, version control, build automation, unit testing, test mocks, code coverage, lightweight static analysis
- Originally, no focus on cloud

First attempt: Reuse online material!

- Wanted to start to work on cloud concepts, especially related to software development and DevOps – important material for their careers
- First attempt: use existing vendor docs (“getting started” materials, online readings)
- Activities not graded beyond participation, goal is to get students to try things
- This...didn't go so well...

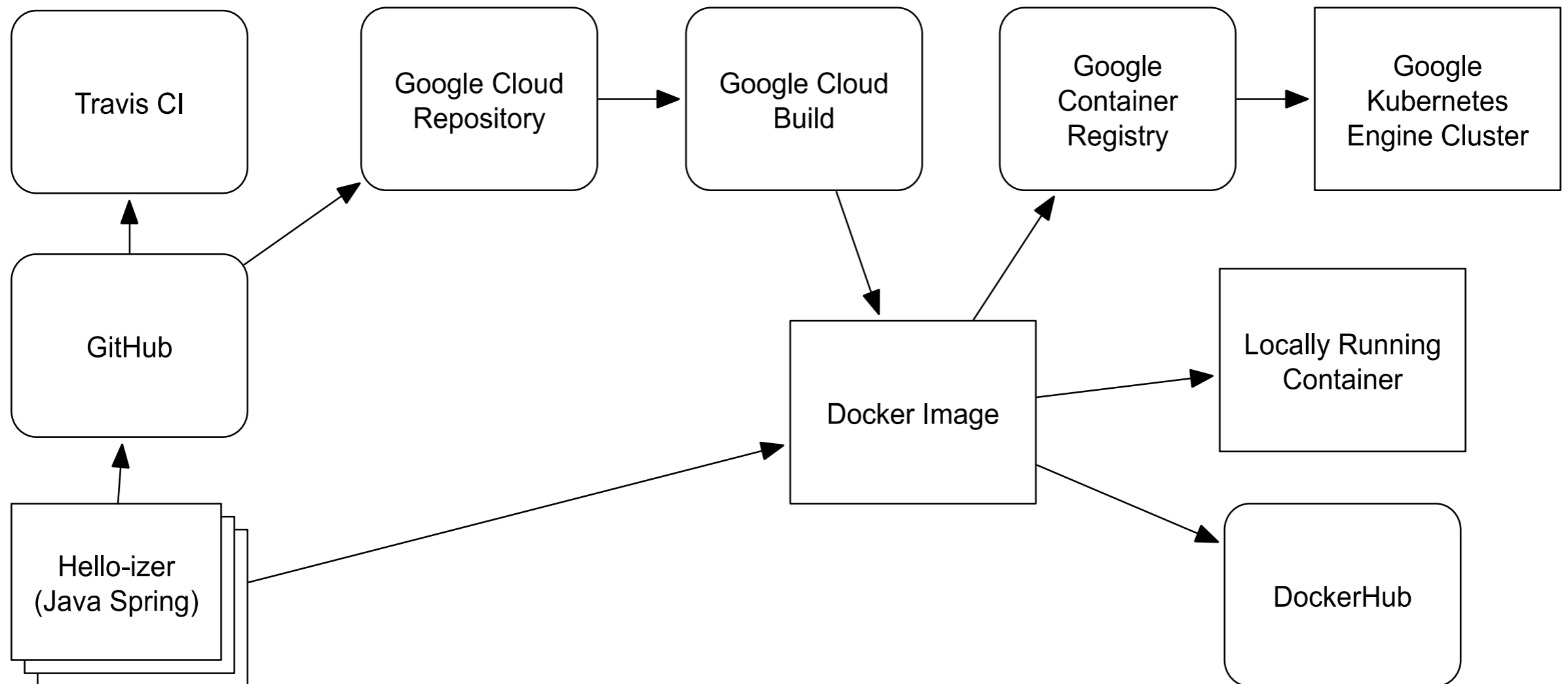
First attempt: Problems

- Sometimes, technologies change faster than docs (minor issue)
- Many docs written for professionals with richer backgrounds (major issue)
- Students could not differentiate fundamentals from examples (major issue)
- Students would get lost, not know how to get back on track (major issue)

Second attempt: start from scratch

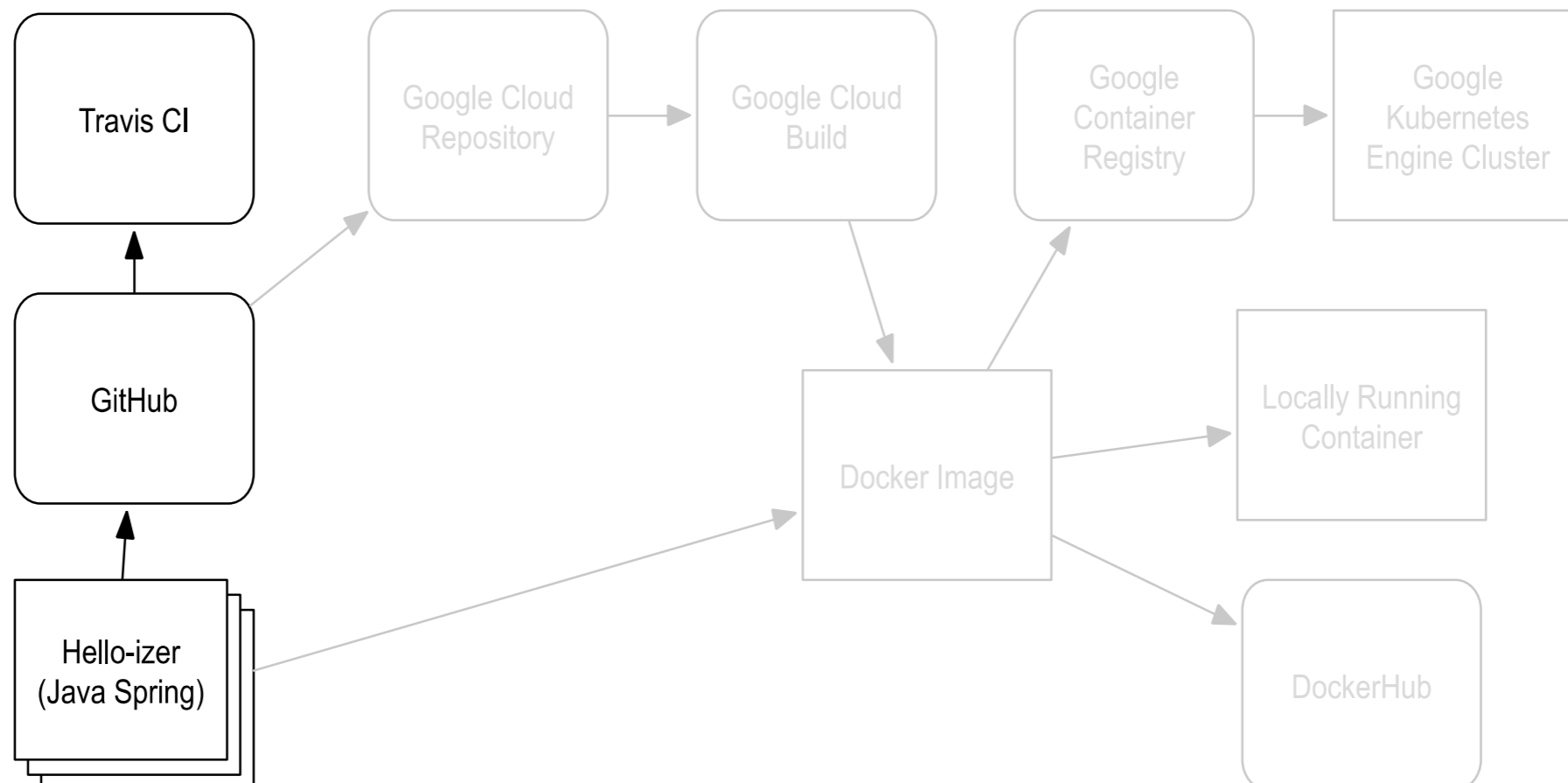
- 4 Activities: Continuous Integration, Docker, Kubernetes, Continuous Delivery
- Mixture of instructor-provided material, videos (LinkedIn Learning), vendor docs
- Ready-made “starters” for each step
- Participation credit, not formally graded (low stress)

Activity overview: moving parts



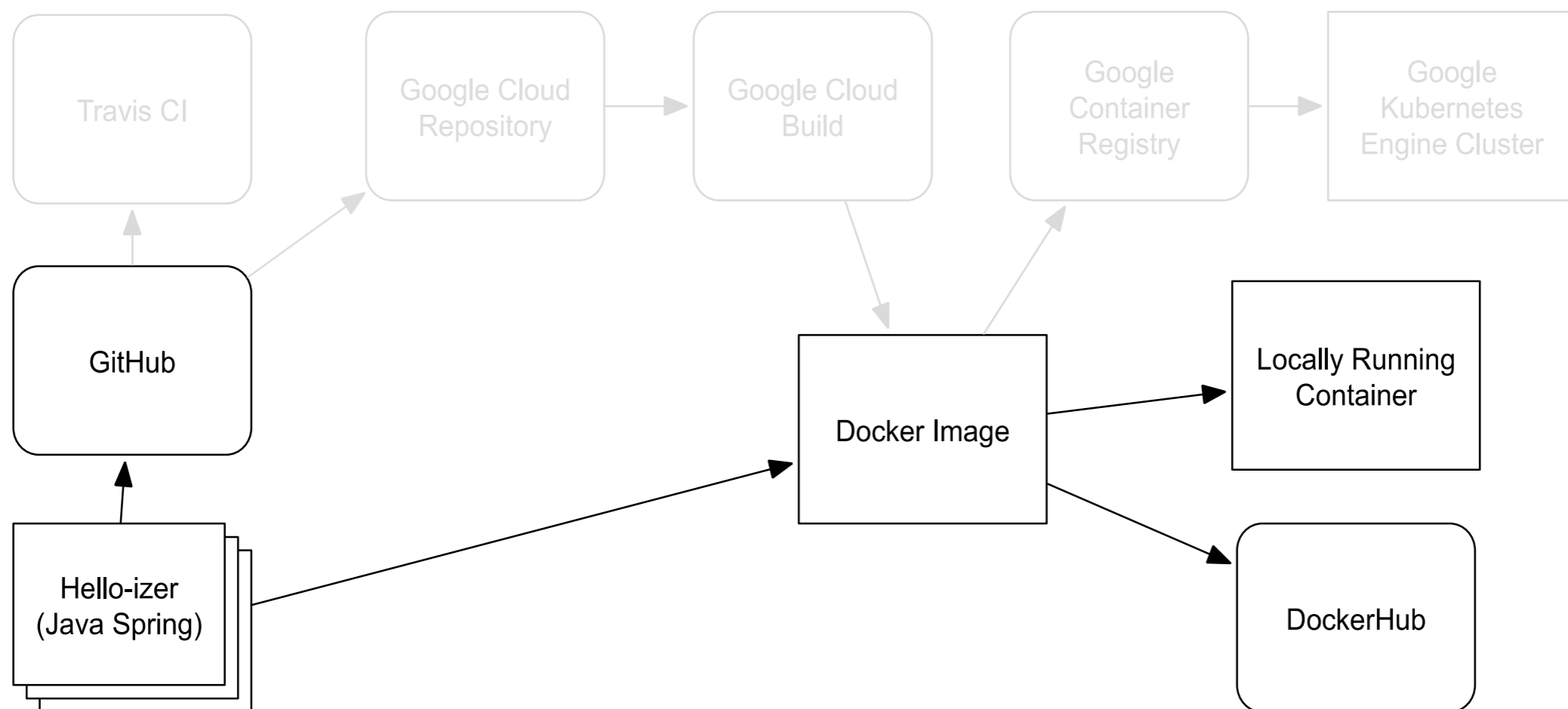
Activity 1: Continuous integration

- Overall goal: students enable CI, see it in action
- Students fork repo, enable Travis-CI integration, trigger a failing build, fix code, push changes to GitHub, and see passing code



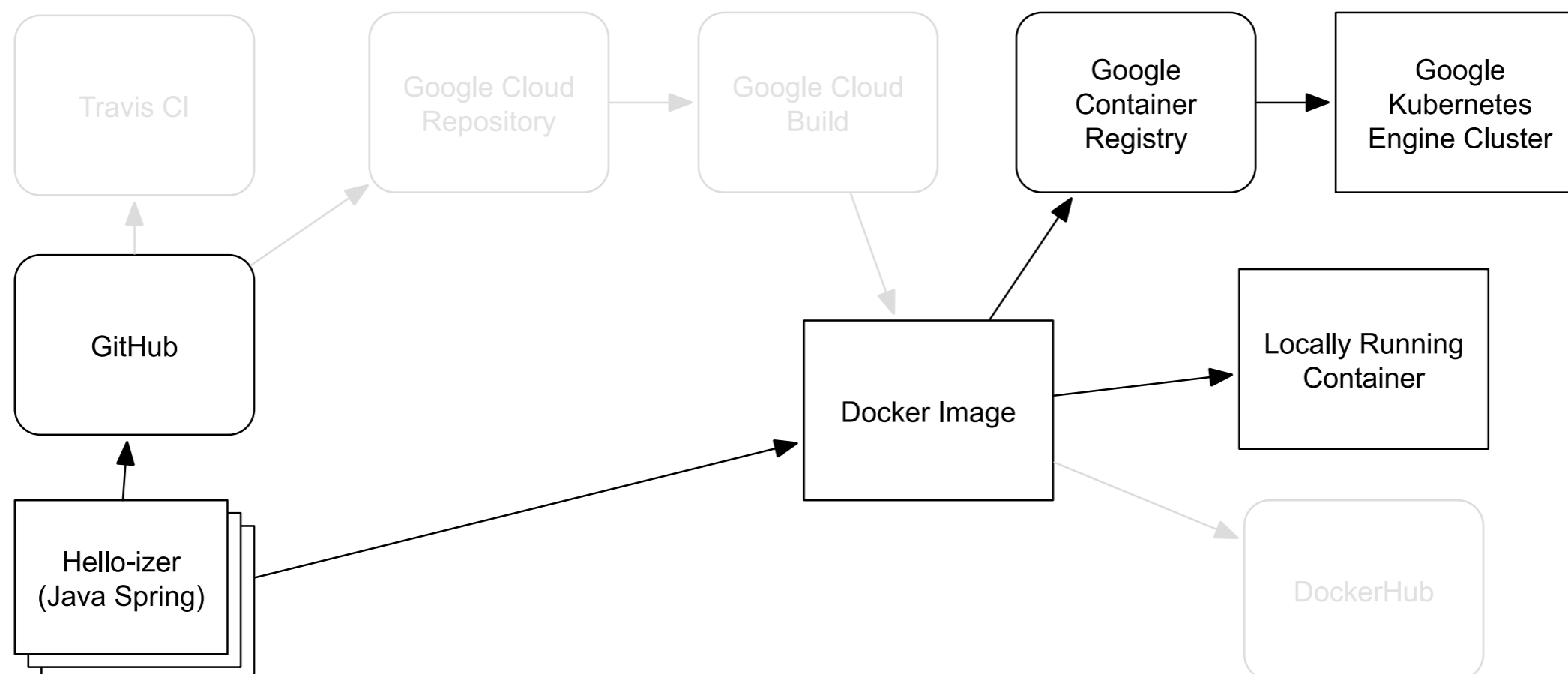
Activity 2: Docker

- Overall goal: students learn the basics of Docker, including build automation to create images
- Students create a Dockerfile, test the image, automate building an image and testing container, push results to DockerHub and GitHub



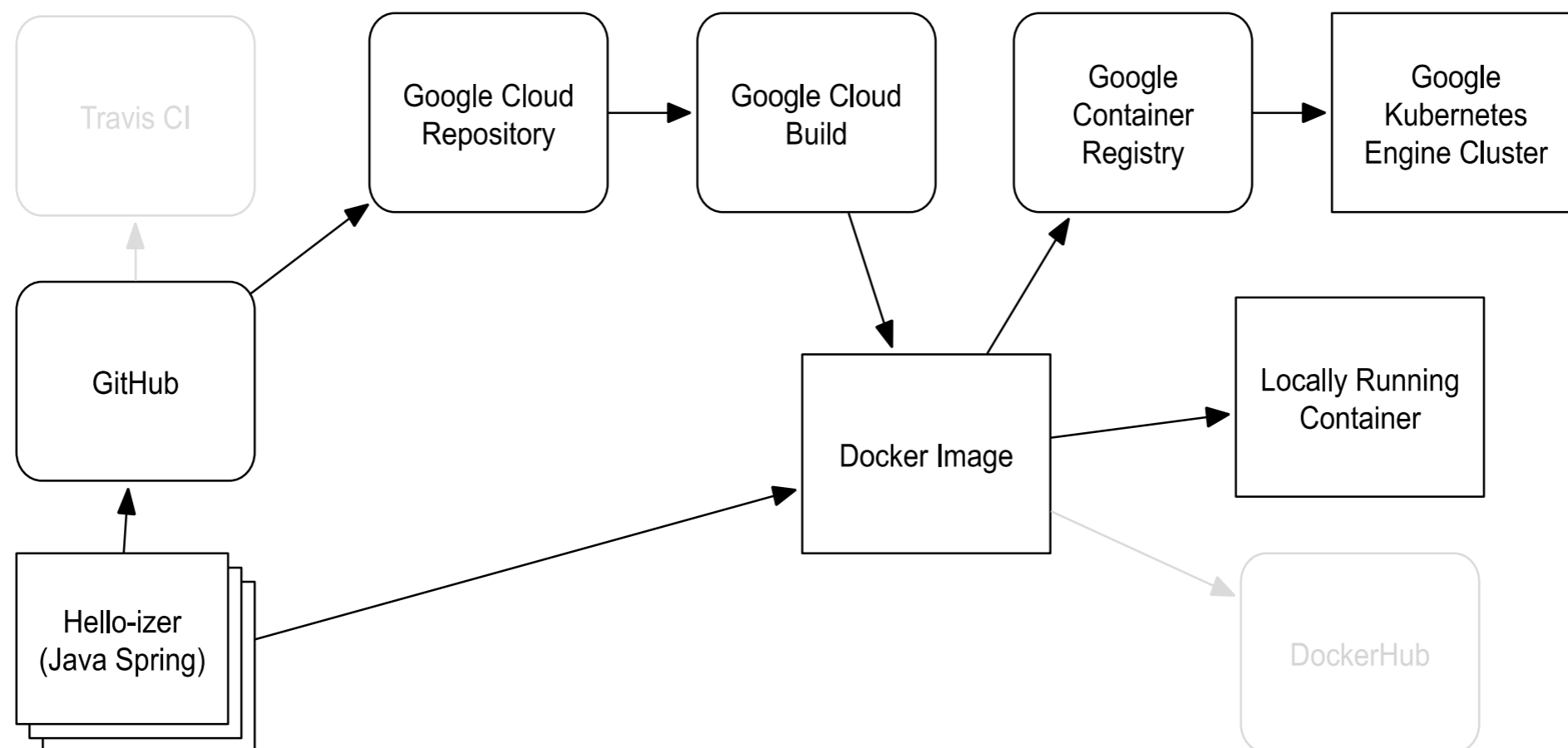
Activity 3: Kubernetes

- Overall goal: students learn how to deploy a Docker image to a Kubernetes cluster
- Students set up project on Google Cloud Platform, push image to Container Registry, create Kubernetes cluster, deploy image, make available



Activity 4: Continuous Delivery

- Overall goal: students put prior steps together to deploy changes to their cluster
- Students mirror repo to a Cloud Source Repository, enable Cloud Build, set up build triggers, check build to ensure Kubernetes updates, automate build



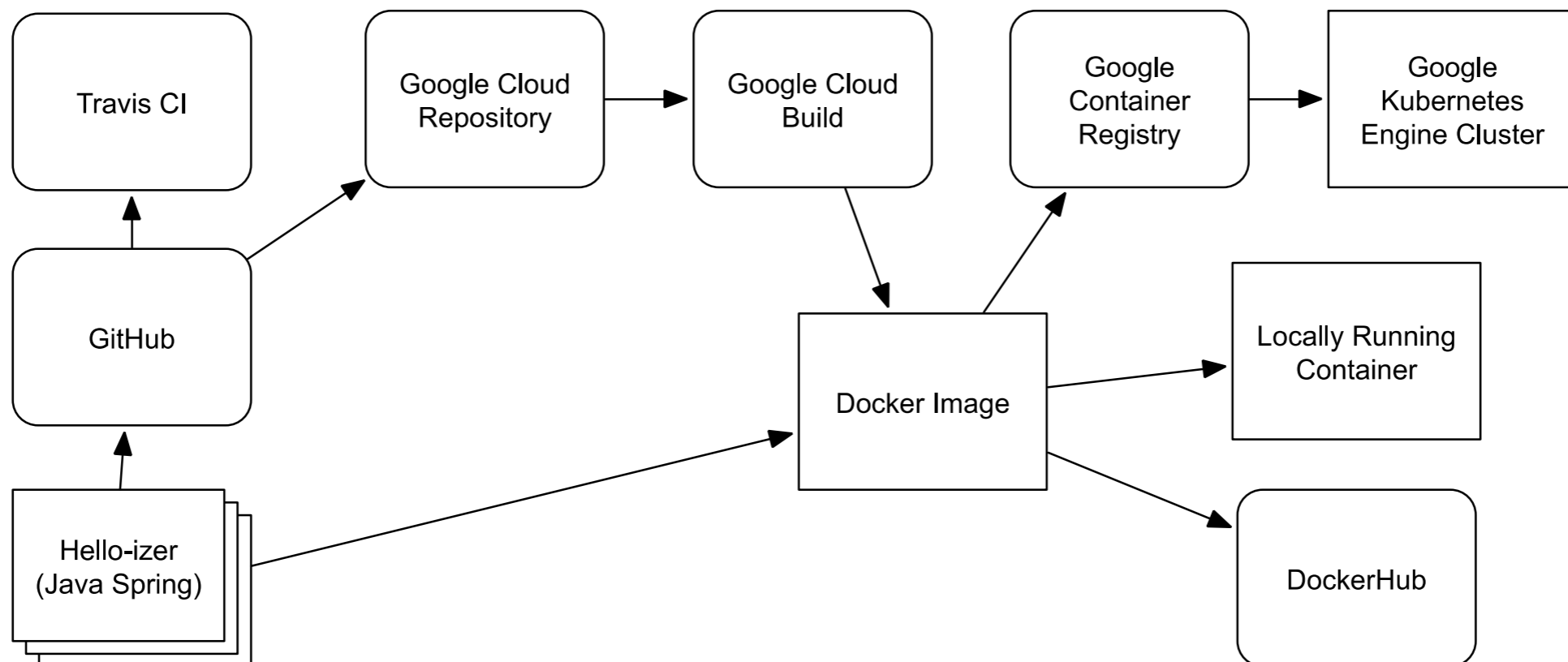
Activity Summary

- By the end, changes to repository trigger test and creation of new Docker image

changes committed to GitHub

- Kubernetes automatically updated using newest

- Students can change API output and quickly see deployed changes



Lessons Learned

- Command line skills are important, but often missing
- Vendor documentation is important, but often missing context students need
- Smaller assignments are easier to follow
- Be ready to work with your students, it's important they learn this material, but they will get stuck on parts of it
- Be patient with yourself, it's hard to keep up with all this stuff!

Student Impressions, Future Work

- Student reaction was generally quite positive
- Students generally had few problems with the first two assignments, but were more likely to struggle with the final two
- One part of future work: extend these to improve them, make them easier to share
- Another part: evaluate these more rigorously, this is a promising experience report

Discussion

102 questions

Sort by » date activity ▼ answers votes RSS

Also see the RascalTutor.

Search tip: add tags and a query to focus your search

Operator Overloading

operator overloading support

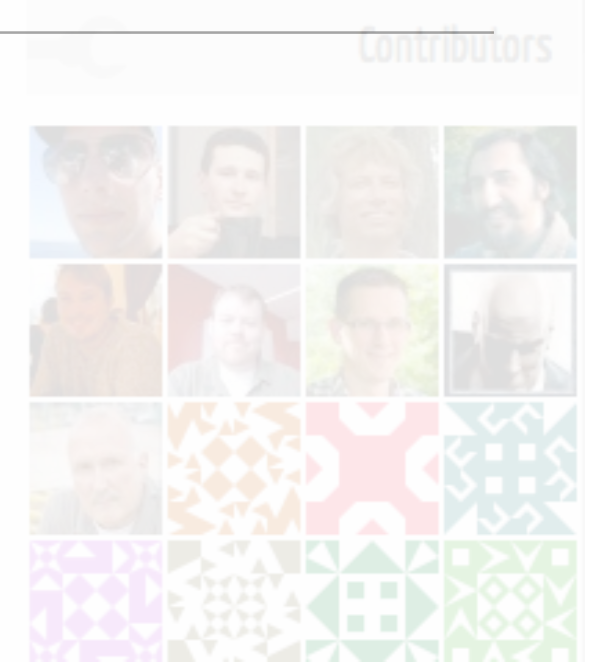
Thank you! Any Questions?

no votes 2 answers 19 views
Mar 07 Hossein

How to solve this MissingFormatArgumentException?

java exception

no votes 1 answer 11 views
Mar 07 Atze



Web: <http://www.cs.ecu.edu/hillsma>

