

CI/CD AND MORE

BAEKGROUND

Pankaj Agrawal

Director , Data & Analytics | Governance & Enablement | Produ



JUN 16



Pankaj Agrawal ✅ · 7:18 AM

Let's do the guest lecture on Jun 24 as the week after is a Canada Day. I will flip you the course structure , let's do CI/CD best practices, learnings , your journey into Meta, how Software Engineering course like this helped you despite it's vagueness 😊 (it may help some more positive feedback).

thanks aryan for bailing

Pankaj Agrawal

Director , Data & Analytics | Governance & Enablement | Produ



APR 8



Pankaj Agrawal ✅ · 8:12 AM

Hey Man, good to hear from you!
All good and hope all is well at your end as well!

Would you be open for guest lecture for C01 in summer

I have tried Aryan , and he's been bailing me for 2 semesters LOL



Navinn Ravindaran ✅ (He/Him) · 9:38 AM

Awesome! I honestly miss being a TA lol



Pankaj Agrawal ✅ · 1:29 PM

We can talk more about what you would like to cover .. I haven't prepared full summer schedule yet but would be around third/fourth week of June

I can always play around the date , depending upon your availability



Navinn Ravindaran ✅ (He/Him) · 5:22 PM

Pankaj: We can talk more about what you would like to cover .. I haven't prepared full summer schedule yet but would be around third/fourth week of June

Perfect, mid-late June works well for me



**Everyone always asks
“Who’s there?”**

But never “How’s there? 😞”



About me

- SWE @ Meta
 - Server Eng; Instagram GenAI
 - PHP/Hack 😭, Python, React
- UTSC CS '24
 - CS TA: B07 (x2), B20, **C01**, C09, D01
 - Co-ops: CaseWare, Halo, RBC, HubSpot, AmEx
 - MLH Fellowship
- ~~been coding since i was 5~~

haha chatgpt meme

Caffeine Induced / Code Disasters

noun

The phenomenon where a developer's over-reliance on caffeinated beverages leads to a temporary boost in productivity, followed by a series of coding mishaps that require extensive debugging and troubleshooting.

CONTINUOUS

INTEGRATION

CONTINUOUS

DEPLOYMENT

Prerequisites

- ✓ Pushing code changes (`git push`)
- ✓ Building/Compiling code (`gcc` , Mobile app builds, Generating static webpages)
- ✓ Testing code (Manual, Unit, E2E Integration)
- 🤔 Deploying code
 - Static sites: GitHub Pages, Netlify, Cloudflare Pages, ...
 - Server(less): Vercel, Railway, Firebase, ...
 - Full-stack: Amazon Web Services, Google Cloud Platform, Azure, Heroku, ...

Story Time





This is Aryan (pretend)

- Just started his first new grad SWE role
- Doesn't know much about the team and the codebase they work on yet
- The codebase is **huge**: Thousands of files, rough idea on where to look
- Opened the repo, needs to make a small change

Aryan makes this change

```
<<GraphQLField(  
    'comments_info',  
    'Get comments for an IG Media (feed, story, reel)',  
>>  
public function getComments(  
-     ?IGMediaCommentsInfoInputObject $input,  
+     IGMediaCommentsInfoInputObject $input,  
): ICommentsInfo;
```

Question: Is there anything incorrect/hacky here?

Aryan tests the code change

```
[arypat@] ➔ hh  
No errors!  
[arypat@] ➔ meerkat  
Starting Meerkat instance 60d43606-a864-43c4-82ba-35a31d34ba12  
Calculating last good mergebase  
Querying changed files  
Resetting artifacts to mergebase  
Running rebuild steps  
Updating source control  
  
Meerkat finished in 105.48 seconds
```

Question: What else can/should Aryan do to test? The change was for getting comments in graphql

IT'S SHOWTIME!

Looking good, he pushes the change: <https://www.internalfb.com/diff/D77156122>

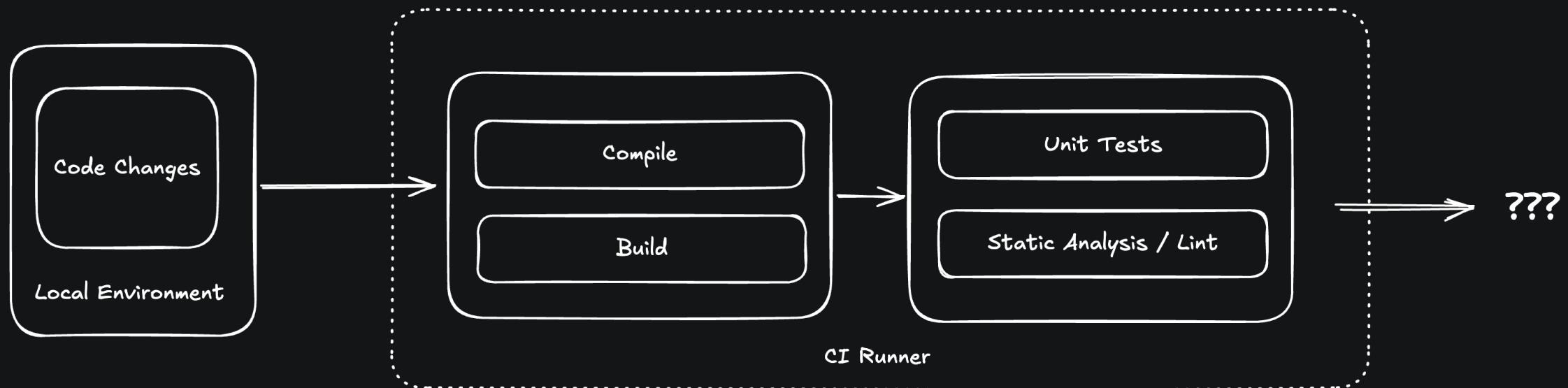
This is only the first part, CI/CD comes after approval and tests are passing.

Things to think about

Remember, Aryan *just* started.

- How long did it take to code → test → push?
- What would happen if there was no CI?
- How long would it have taken to test 'everything' manually?
 - **Could** he do that?
 - **Should** he do that?

FLOW SO FAR...



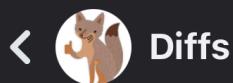
TL;DR: Automated service to quickly test code changes and detect errors early

Question: What happens if this is really slow?

Additional CI Capabilities

- Static Analysis
 - Code Coverage
 - Security Vulnerability Scanning (leaking env. variables, secrets)
- Build/Test Artifacts
 - Log of passing/failing tests
 - Screenshots of UI changes end-to-end
- Async/Offline Triggers
 - Sending notifications
 - Database schema changes
 - Sending notifications on failure/success

9:16



CI signals failed for D77036349:
[Edits][Comment Stream] Add LLM
summary of comments in view

[Go to first failure](#)

3:40 PM

Tests or Static Analysis error

9:16



Metamate for Testing published a diff
that requires your review:
D77041604: [Generated Tests] Add
tests to cover changes in D77036349

(View Butterfly Rule: <https://fburl.com/butterfly/xf9dq729>)

4:18 PM

Automated unit test generation

9:15



Butterfly

Navinn Ravindaran has a new
diff:

D77039005: [Edits][Comment
Stream] Add support for short/
long themes

(View Butterfly Rule: <https://fburl.com/butterfly/cr55721m>)



Shared engineering group

✖ feat(C01): add guest lecture

Build & Deploy #122: Commit [2f72ca3](#) pushed by navn-r

main

2 days ago

6s

...

✖ feat(C01): add guest lecture

Build & Deploy #121: Commit [f31e251](#) pushed by navn-r

main

2 days ago

6s

...

✖ feat(C01): add guest lecture

Build & Deploy #120: Commit [8b261ad](#) pushed by navn-r

main

2 days ago

14s

...

✖ feat(C01): add guest lecture

Build & Deploy #119: Commit [a66c6e7](#) pushed by navn-r

main

2 days ago

17s

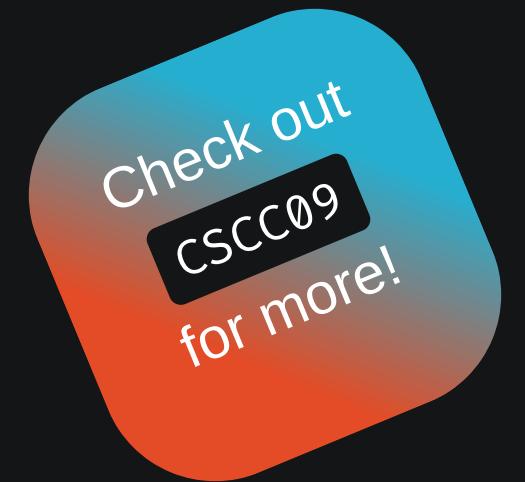
...

After CI Passes



DEPLOYMENT

- Depends on the company / team / project / architecture
- Staggered Rollout
 - i. α / Development (latest , unstable , main)
 - ii. Staging (test , qa , staging)
 - iii. Production (prod , stable)
- Types of Deployment:
 - A/B: Experiment between two versions, see metrics and choose
 - Blue/Green: Run both, then instant swap
 - Canary: Gradual release based on % of population
 - Rolling: Gradual release based on # of services



Question: Does this need to be (as) fast?

To Conclude

Code ⇒ Test ⇒ Build ⇒ Review ⇒ Deploy ⇒ Pray nothing breaks

- The earlier you set it up, the more it helps
- Bad CI/CD is worse than none
- Don't reinvent the wheel
- Green check ≠ Things look good (sometimes)
- CI/CD and Devops at scale - usually someone else's ~~problem~~ responsibility

Resources:

- <https://docs.github.com/en/actions/writing-workflows/>
- <https://about.gitlab.com/topics/ci-cd/>
- Meta AI ChatGPT (lmao)

TALK IS CHEAP



SHOW ME THE CODE

Closing Thoughts

Why this course matters.