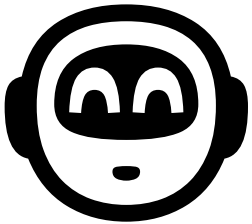
Continuous Integration

with Travis CI

What is Continuous Integration (CI)?

- Simply and practically:

  Continuous Integration just means **automatically running all of your automated tests** (unit, integration, etc) every time you change the code.



What is Continuous Integration (CI), more generally?

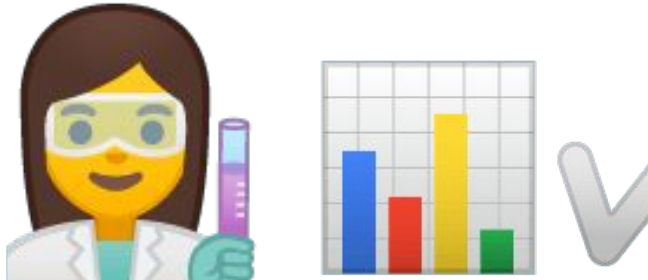
– – –

- CI means continuously **integrating new code** into existing code.
- In a sense, new code has not been “integrated” into the rest of the code until it is all tested together.
- That’s why CI is called “**continuous**” - it tests every new batch of code with existing tests.

Continuous Integration (CI) is about tests.

– – –

- CI is a **natural extension of automated tests**, like unit and integration tests.
- Tests (should) tell you about correctness of your code.
- Running them locally is nice.
- Having them **run automatically** is way better.



Why Continuous Integration?

- For any given new pull request, **most of your tests are safe**. They probably won't break.

- EX: You change how `User.follow(friendUser)` works.

  Do you expect a test for `Mood.updateLocation(friendUser)` to break?

  Do you expect a tests for `FollowActivity` to break?

Why Continuous Integration?

Code can be fragile.

Tests help **protect the code you didn’t expect to break.**

Sometimes you just forget to run tests locally before pushing.

Or you just don’t have time.

Either way, CI has your back.

Which CI Tools are available?

– – –

CircleCI

Travis CI

Bamboo

Jenkins

TeamCity

Many others.

How Travis CI Works

We'll use Travis CI. At the basic level it's the same as the rest.

- You point it at a cloud source control repository. (EX: a GitHub repo)
- Travis **listens for new commits** that come from pushes.
- Travis builds your code and **runs your tests** automatically.
- Travis **notifies you of failures** so you don't break master. (Or worse.)

How does Travis automatically build and test?

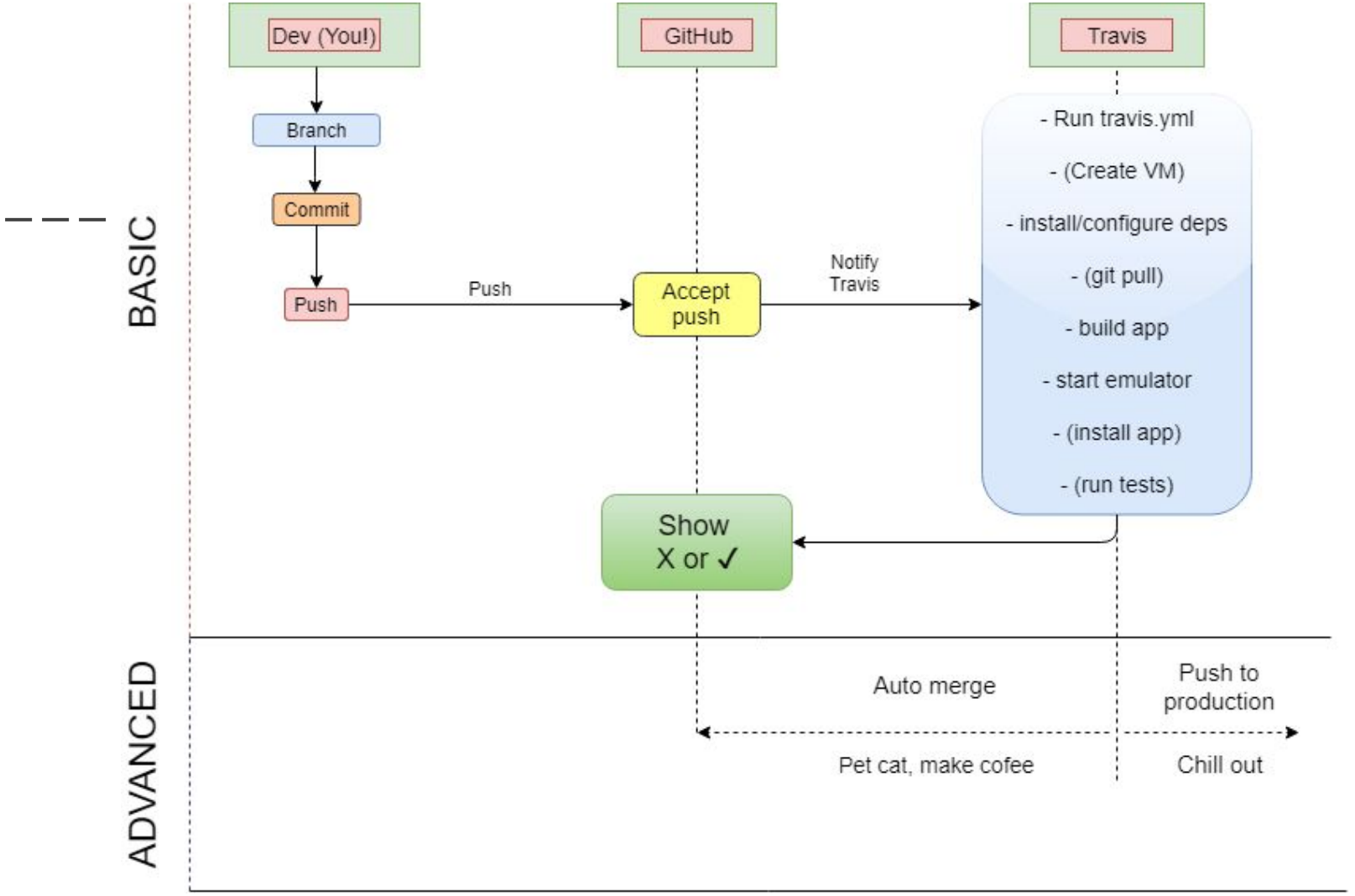
Travis sits in the cloud. When Travis notices a new commit, it does the following:

- Starts a virtual machine.
- Installs and configures your basic dependencies (EX: java, gradle, android SDK)
- Performs a `git pull` to get your code into the VM.
- Uses gradle to build your app.
- Starts an android emulator.
- Installs your app on the emulator.
- Runs your tests.

How does Travis automatically build and test?

Travis sits in the cloud. When Travis notices a new commit, it does the following:

- Starts a virtual machine.
- Installs and configures your basic dependencies (EX: java, gradle, android SDK)
- Performs a `git pull` to get your code into the VM.
- Uses gradle to build your app.
- Starts an android emulator.
- Installs your app on the emulator.
- Runs your tests.


Dev (You!)
GitHub
Travis
Branch
Commit
Push
Push
Accept push
Notify Travis
- Run travis.yml
- (Create VM)
- install/configure deps
- (git pull)
- build app
- start emulator
- (install app)
- (run tests)
Show X or ✓
BASIC
ADVANCED
Auto merge
Push to production
Pet cat, make cofee
Chill out