

Deploy **fast** and **well**

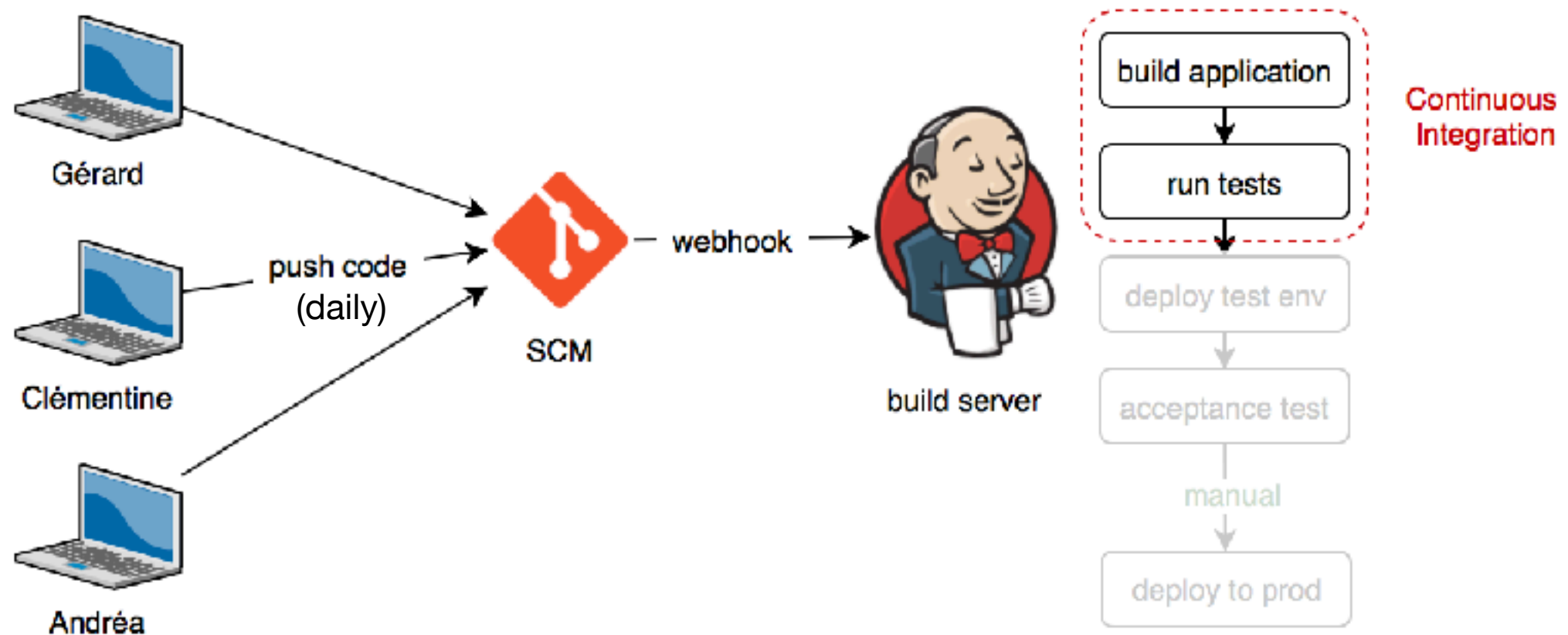
Continuous Integration and Continuous Delivery for
Frontend developers

sami@boudrai.com

**Pushing code to
prod on Friday
afternoon**



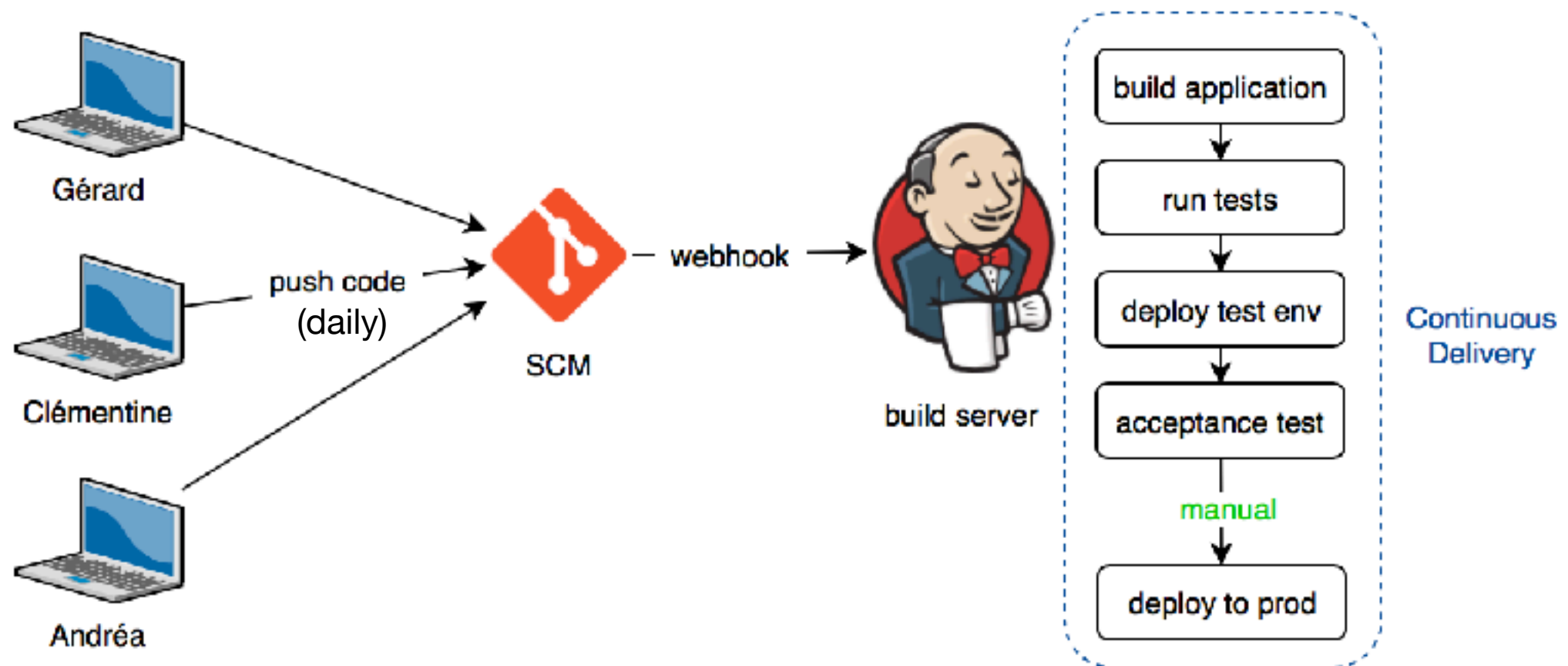
Continuous Integration



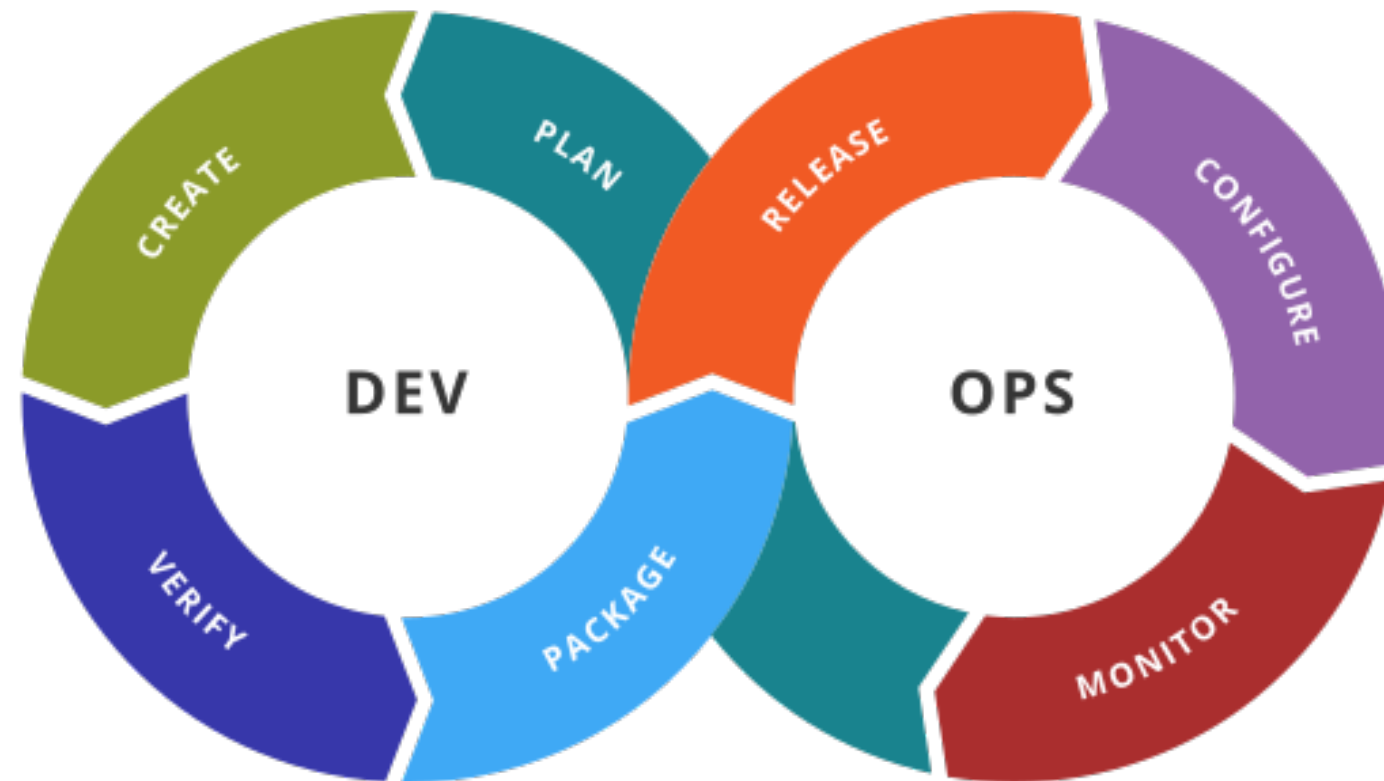
¯_(ツ)_/¯

IT WORKS
on my machine

Continuous Delivery



Dev + Ops



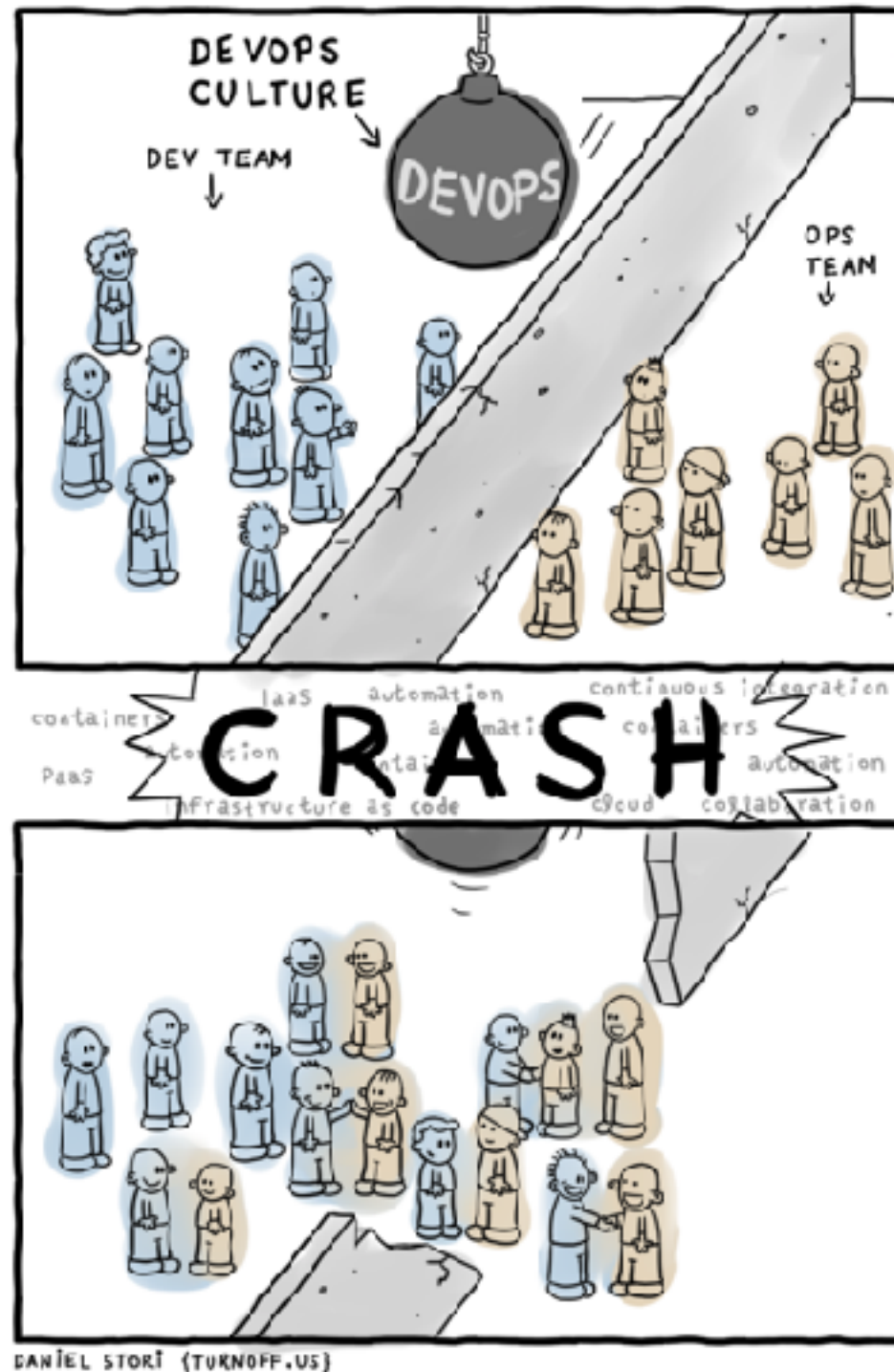
Dev team:

- plan changes
- write code
- write tests
- package the application

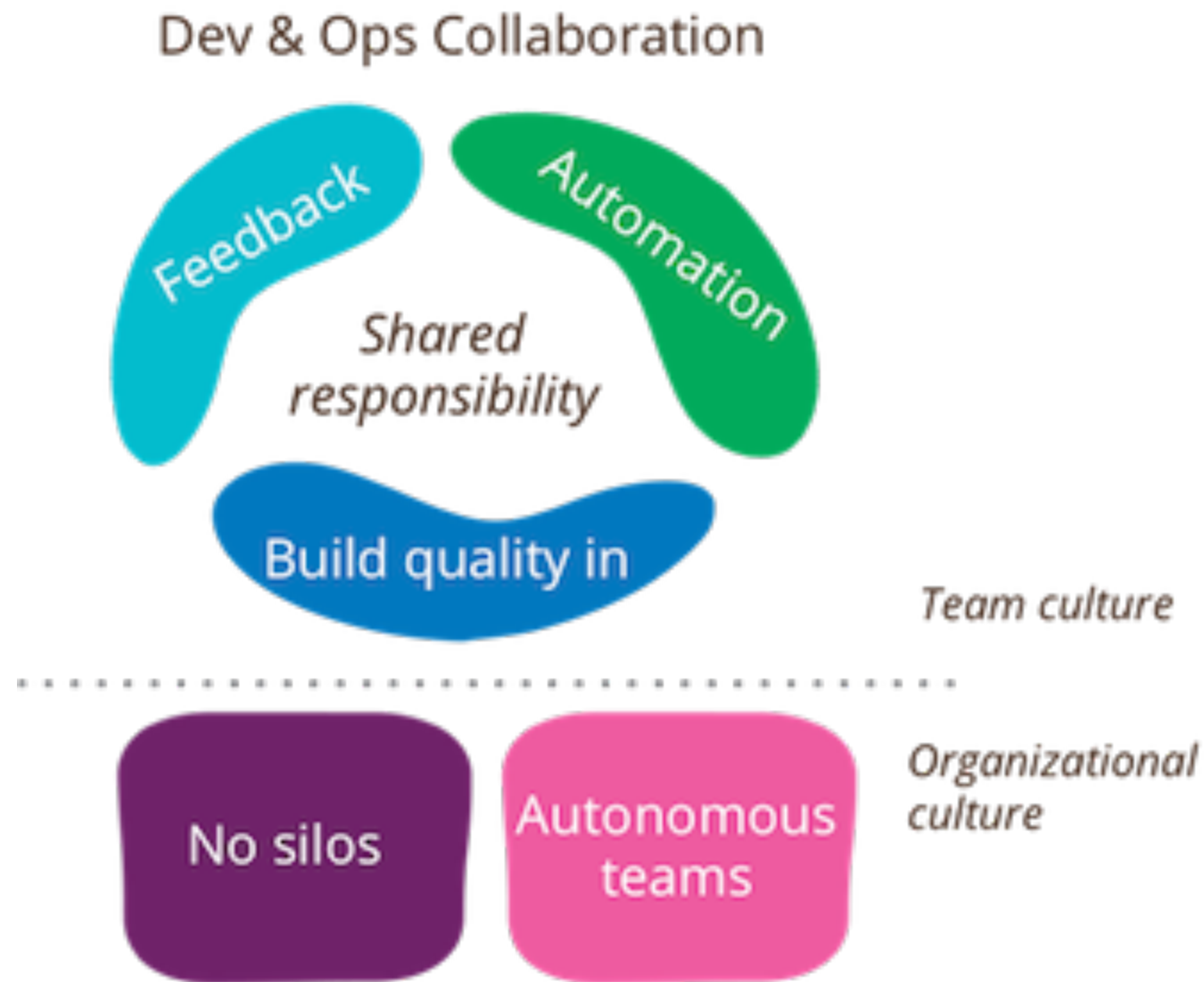
Ops team:

- deploy application
- configure infrastructure
- monitoring & error reporting

DevOps



Devops Culture



<https://martinfowler.com/bliki/DevOpsCulture.html>

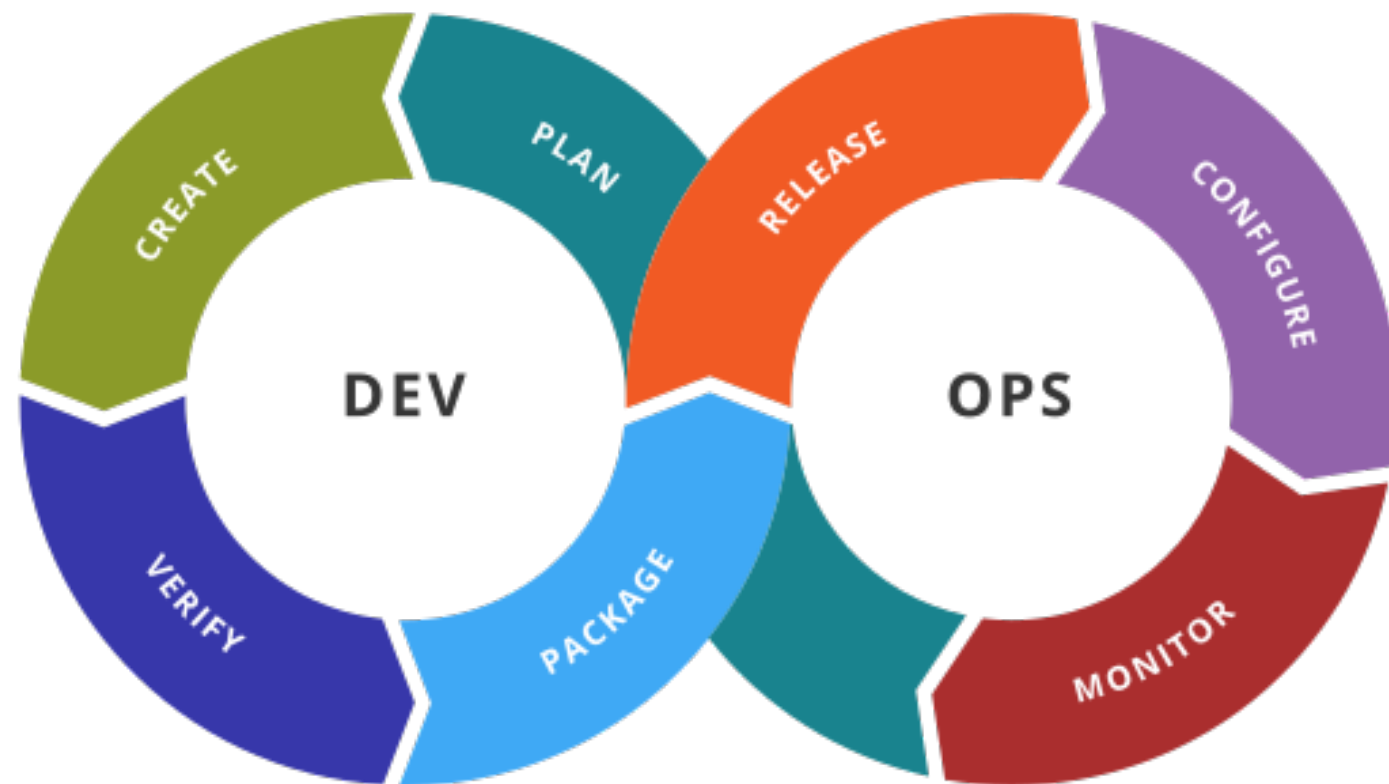
Why so popular (bad)

- Google/Facebook/Netflix/etc... deploy x times per day, so we need to do the same
 - Google has different challenges
 - Google has better code development practises
- Replace 1 dev + 1 ops with 1 devops

Why so popular (good)

- "release often" mantra (cf. Agile, XP, etc...)
 - smaller code increment cause smaller problems
 - tighter customer feedback loop
 - higher reaction speed to urgent fixes (security bugs, etc...)

DevOps



Pipeline

- Declarative configuration style
- Commit the pipeline config to your SCM to benefit from versioning
- Examples:
 - Jenkins
 - Drone
 - Travis CI
 - Gitlab CI
 - Concourse
 - Circle CI
 - Azure DevOps
 - GitHub Actions
 - etc...

```
pipeline:  
  frontend:  
    image: node  
    commands:  
      - npm install  
      - npm test  
  backend:  
    image: golang  
    commands:  
      - go test -v  
      - go install  
  
services:  
  database:  
    image: mysql
```


Pipeline

- Most give access to command line, so you can do anything you want
- Having direct integration can simplify writing the pipeline

Travis CI:

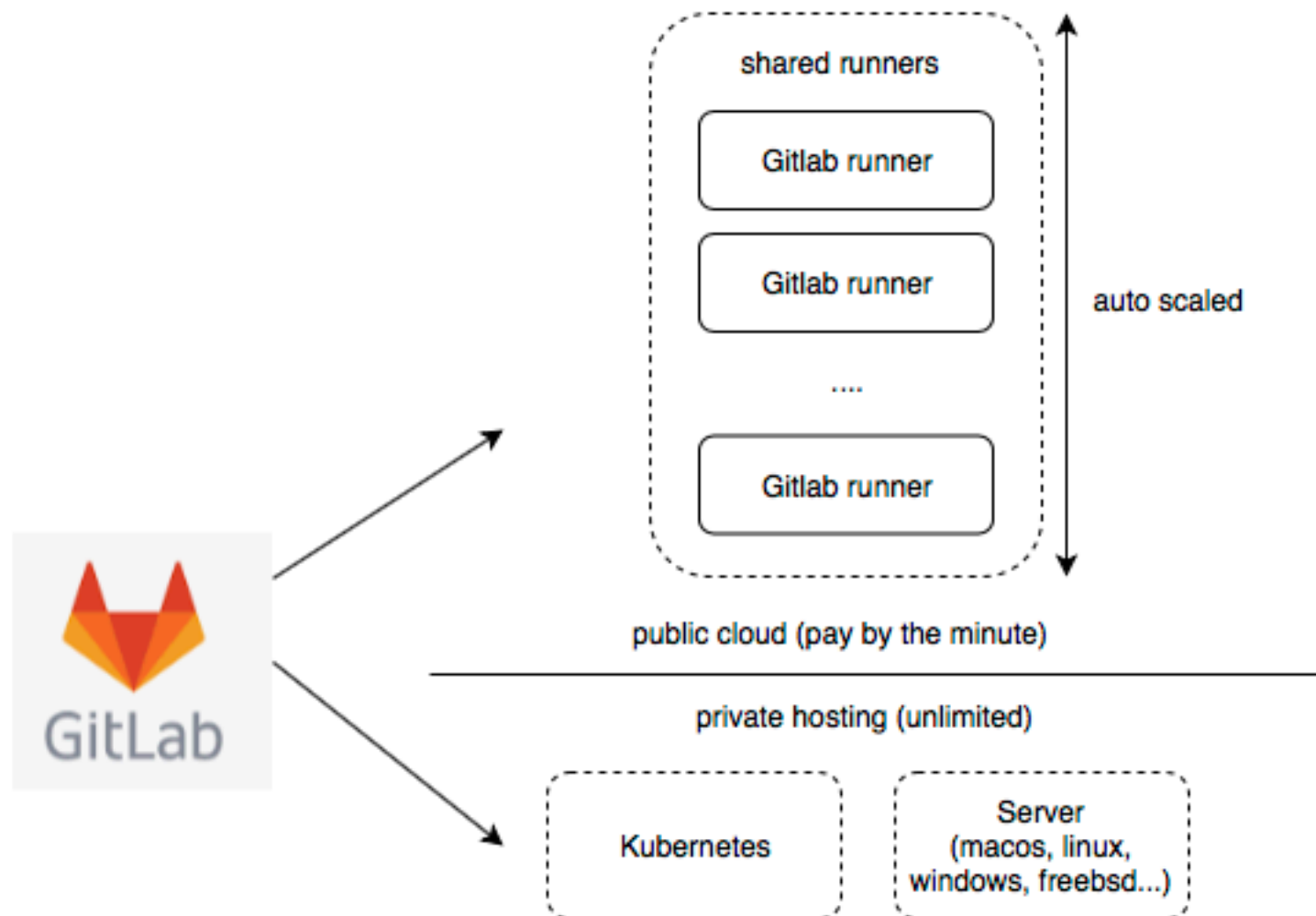
```
deploy:  
  provider: heroku  
  api_key: ...  
  app: my-app-123
```

Gitlab CI:

```
production:  
  type: deploy  
  script:  
    - apt-get update -qy  
    - apt-get install -y ruby-dev  
    - gem install dpl  
    - dpl --provider=heroku --app=gitlab-ci-python-test-prod --api-key=$HEROKU_PRODUCTION_API_KEY  
  only:  
    - tags
```

Pipeline

- Where does my pipeline run?



PLAN

Issue Board

- Product Owner creates a new issue
- DEMO

CODE

Gitlab's Merge Request

- equivalent to Gitlab's Pull Request
- created directly from Issues
- can integrate test coverage and results by configuring gitlab-ci.yml



Pipeline #35517850 passed for b5843502 on 1-add-operator-for-m...
Coverage 60.00%



Test summary contained 1 failed test result out of 5 total tests



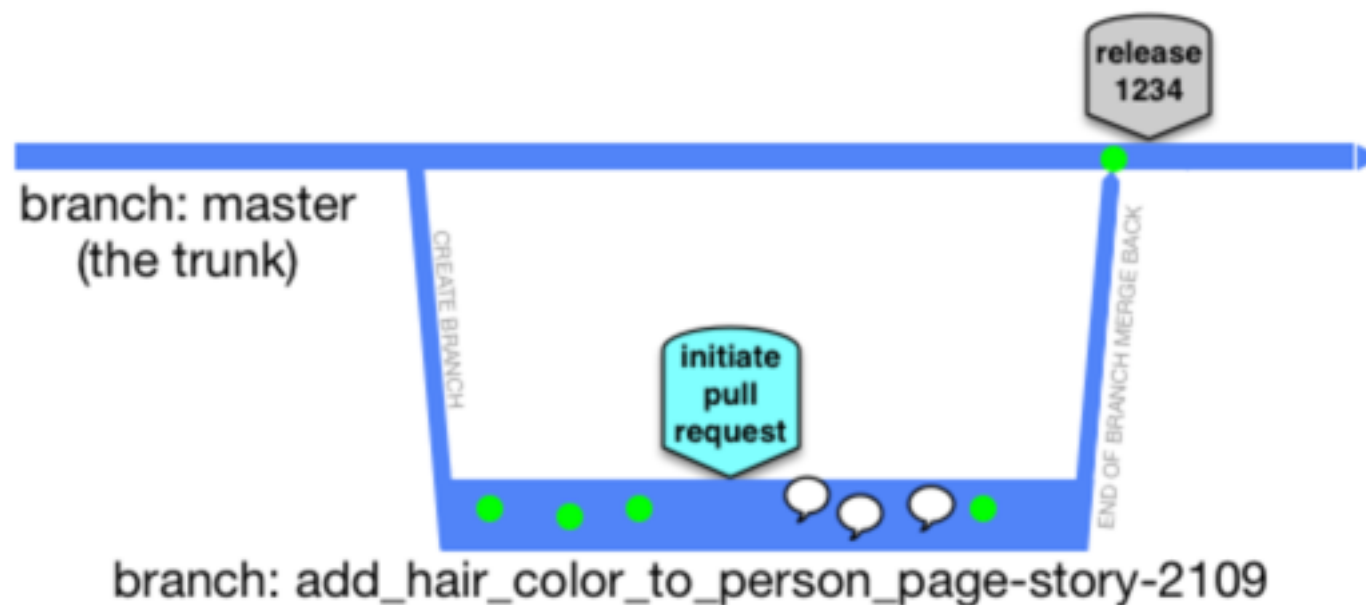
test found 1 failed test result out of 5 total tests



New Calculator.vue Calculates 2 x 2

Branching

- Merging conflicts is a pain, to avoid conflicts, keep branches short-lived
- Trunk Based Development (cf. <https://trunkbaseddevelopment.com/>)



CODE

Tips for code collaboration

- Commit your yarn.lock

package.json

```
"dependencies": {  
  "vue": "^2.5.17"  
}
```

yarn.lock

```
vue@^2.5.17:  
  version "2.5.17"  
  resolved "https://registry.yarnpkg.com/vue/-/..."
```

CODE

Tips for code collaboration

- Add eslint to as git pre-commit hook (with husky)

package.json

```
"lint-staged": {
  "*.{js, vue}": [
    "eslint --fix",
    "git add"
  ]
},
"gitHooks": {
  "pre-commit": "lint-staged && yarn test:unit"
}
```

Having an enforced code style greatly simplifies reading diffs

VERIFY

Testing dependencies

- Gitlab can make “services” like mysql, postgres, redis (or any docker container) available to your test scripts
- Other CIs can propose more pre-installed services
- But replicating this with docker-compose is easy

VERIFY

Verify more

- Automatic Static Application Security Testing
but frontend Javascript is not supported (cf. <https://gitlab.com/gitlab-org/gitlab-ee/issues/5473> to track progress)

VERIFY

Verify more

- Automatic Static Application Security Testing
but frontend Javascript is not supported (cf. <https://gitlab.com/gitlab-org/gitlab-ee/issues/5473> to track progress)

The screenshot shows a GitLab merge request interface. At the top, it says "Request to merge `awesome-feature` into `master`" with a "Check out branch" button. Below this, a green checkmark indicates "Pipeline #18777035 passed for 8825f6cd." A warning icon and text state "SAST improved on 1 security vulnerability and degraded on 4 security vulnerabilities" with a "Collapse" button. A list of four SAST findings follows: three red 'X' marks for "Medium: Cipher with no integrity", "Medium: ECB mode is insecure", and "Medium: Predictable pseudorandom number generator" (all in `src/main/java/com/gitlab/security_products/tests/App.java`), and one green checkmark for "Medium: Predictable pseudorandom number generator" (also in `src/main/java/com/gitlab/security_products/tests/App.java`). A link "Show complete code vulnerabilities report" is below the list. At the bottom, there is a green "Merge" button, checkboxes for "Remove source branch" and "Squash commits", and a "Modify commit message" button.

Request to merge `awesome-feature` into `master` Check out branch

✓ Pipeline #18777035 passed for 8825f6cd. ✓

⚠ SAST improved on 1 security vulnerability and degraded on 4 security vulnerabilities Collapse

- ✗ Medium: Cipher with no integrity in `src/main/java/com/gitlab/security_products/tests/App.java:29`
- ✗ Medium: ECB mode is insecure in `src/main/java/com/gitlab/security_products/tests/App.java:29`
- ✗ Medium: Predictable pseudorandom number generator in `src/main/java/com/gitlab/security_products/tests/App.java:41`
- ✓ Medium: Predictable pseudorandom number generator in `src/main/java/com/gitlab/security_products/tests/App.java:47`

[Show complete code vulnerabilities report](#)

✓ Merge ☒ Remove source branch ☒ Squash commits Modify commit message

RELEASE

Deploy frontend

- Only requires static site hosting
- Several providers provide optimised solutions for SPAs
 - Now
 - Surge
 - Google Firebase Hosting
 - Netlify
- Advantages:
 - CDN
 - Auto-SSL
 - Cache invalidation
 - Atomic uploads & rollback

Challenges

- Pipelines can sometimes be slow (but many providers propose caching and parallelisation)
- “I have been coding for many years, I don’t need to write tests”
- Faux Agile <https://martinfowler.com/articles/agile-aus-2018.html>
ex: “We have stand-ups so we do Agile”
- Anti-pattern: DevOps as a 3rd silo (cf. <https://web.devopstopologies.com>)



Gitlab?

- (~) No specific endorsement for Gitlab: it's been very good for my team but other teams might have different needs
- (-) gitlab.com is regularly down (although it's better since their migration to GCP, cf. <https://about.gitlab.com/2018/10/11/gitlab-com-stability-post-gcp-migration/>)
- (-) Gitlab prefers breadth over depth (<https://about.gitlab.com/company/strategy/#breadth-over-depth>) so make sure to properly check properly if Gitlab covers your use cases
- (+) but for small teams, Gitlab provides everything you need ootb
- (-) Github has a richer ecosystem
- (+) You can deploy your own dedicated gitlab-ci runners - many Ones supported and jobs can run potentially faster