

Modern PHP Deployments using CI and Laravel Envoy

From GitHub to Production

Overview

- Typical Deployment Scenarios
- Propose a New Solution
- Implementation
- Automation

Typical Deployment Process

Deploying Code is Hard

Deployment Scenario 1 - FTP

- Local files on a shared network folder
- Each new project a cut-paste of core code
- Manual copying of files over FTP
- Shared core files using symlinks

Scenario 1 Problems

- File Conflicts
- No Revision History
- No Development/Production Parity
- Fragile Core Code

Deployment Scenario 2 - Git

- SSH into the production server
- cd to the web server root
- git fetch origin (update)
- git reset origin/master --hard (deploy)
- composer install (download dependencies)

Scenario 2 Problems

- Lots to remember
- Easy to make a mistake
- Downtime while Composer is installing
- Non-trivial rollbacks

Propose a New Solution

Deploying Code Should be Easy

Automated Deployment Requirements

- A simple command to run locally
- Same tooling for production and staging
- Define a specific *branch* or *commit* to release
- Ability to *rollback* to the previous release
- Zero-downtime switch between releases

Deployment Automation Tools

- **Capistrano** - Ruby
- **Fabric** - Python
- **Shunt** - PHP (League)
- **Laravel Envoy** - PHP (& Bash)
- *many, many others...*¹

¹ http://software.wikia.com/wiki/List_of_build_automation_software

Choosing Laravel Envoy

- PHP-based solution
- Blade-style syntax
- Parallel execution (Multi-Server)
- Slack/HipChat notifications
- Not Laravel dependent

Our Envoy Commands

envoy run deploy

Deploys the master branch to all servers.

envoy run deploy --branch=my_branch --target=web

Deploys my_branch to all servers in the web pool.

Our Envoy Commands (continued)

envoy run rollback

Rollback all servers to the previous release.

envoy run cleanup

Manual cleanup of the releases directory, keeps the last 5.

Implementation

Stringing it Together

envoy run deploy

- git clone to a new directory in /var/www/releases
- composer self-update and composer install
- Create symlinks to shared folders and config files
- Update symlinks for previous and current releases
- Restart Apache

envoy run rollback

- Update the current symlink to point to the previous release
- Restart Apache

Note: Only supports automated rollback to previous version.

Manual rollbacks can be performed "on the box" for any of the last 5 releases.

Automation

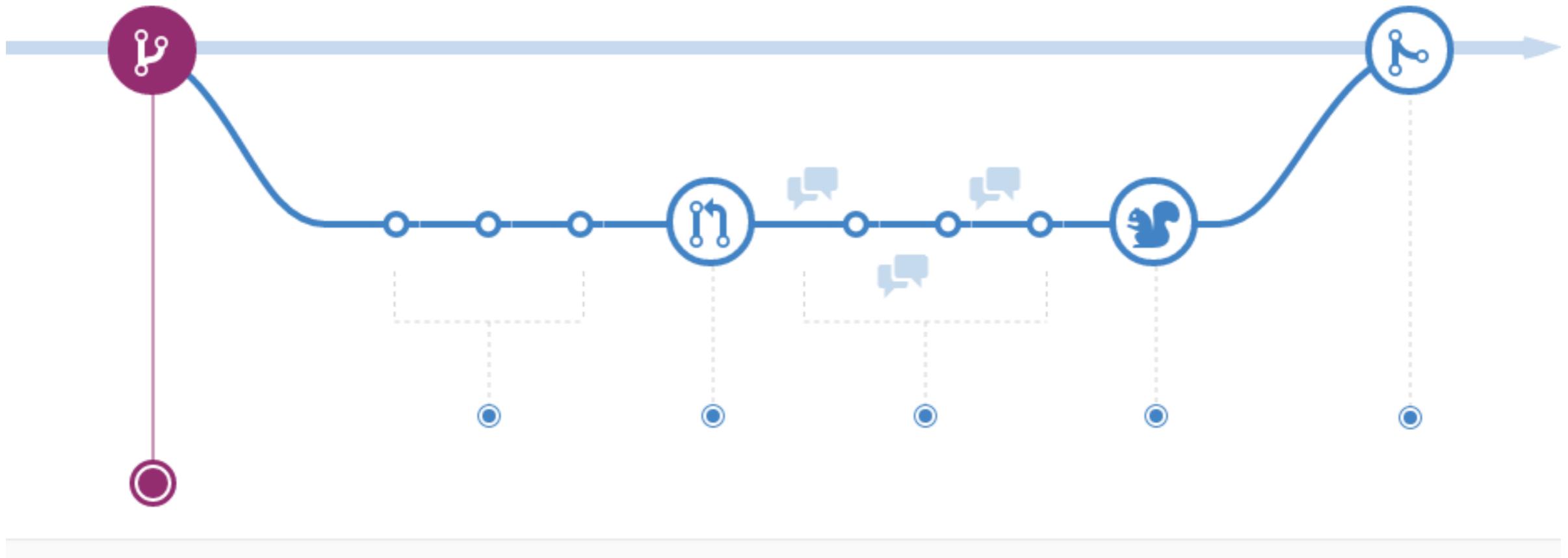
Hello Continuous Deployments!

Continuous Deployments

- Deployments enacted by your CI suite
- Only triggered by a merge into the master branch
- Minimal risk when adopting GitHub Flow²

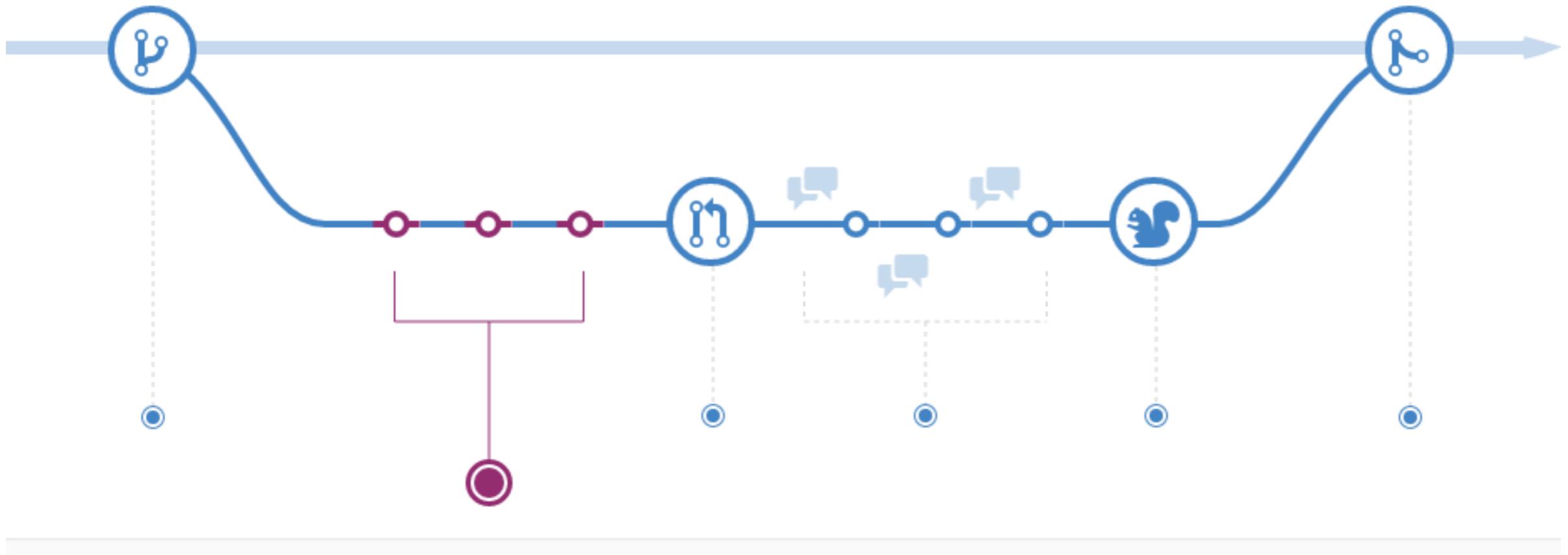
² <https://guides.github.com/introduction/flow/>

GitHub Flow Explained



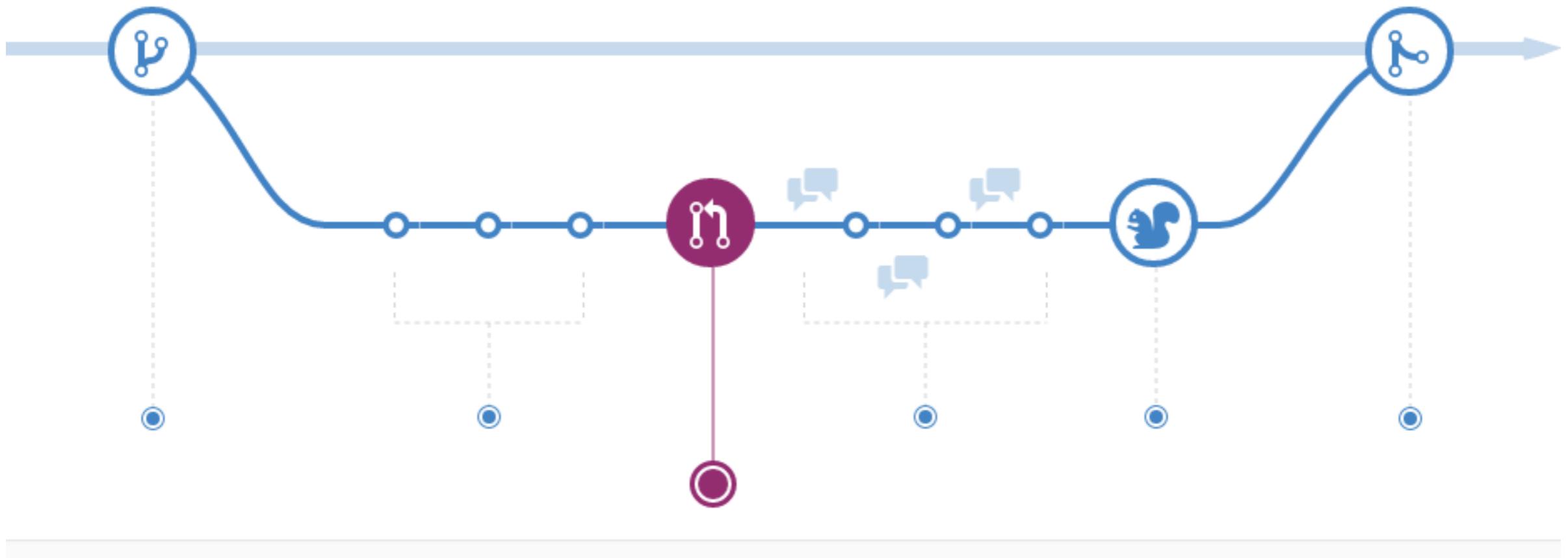
Step 1 - Create a branch

GitHub Flow Explained



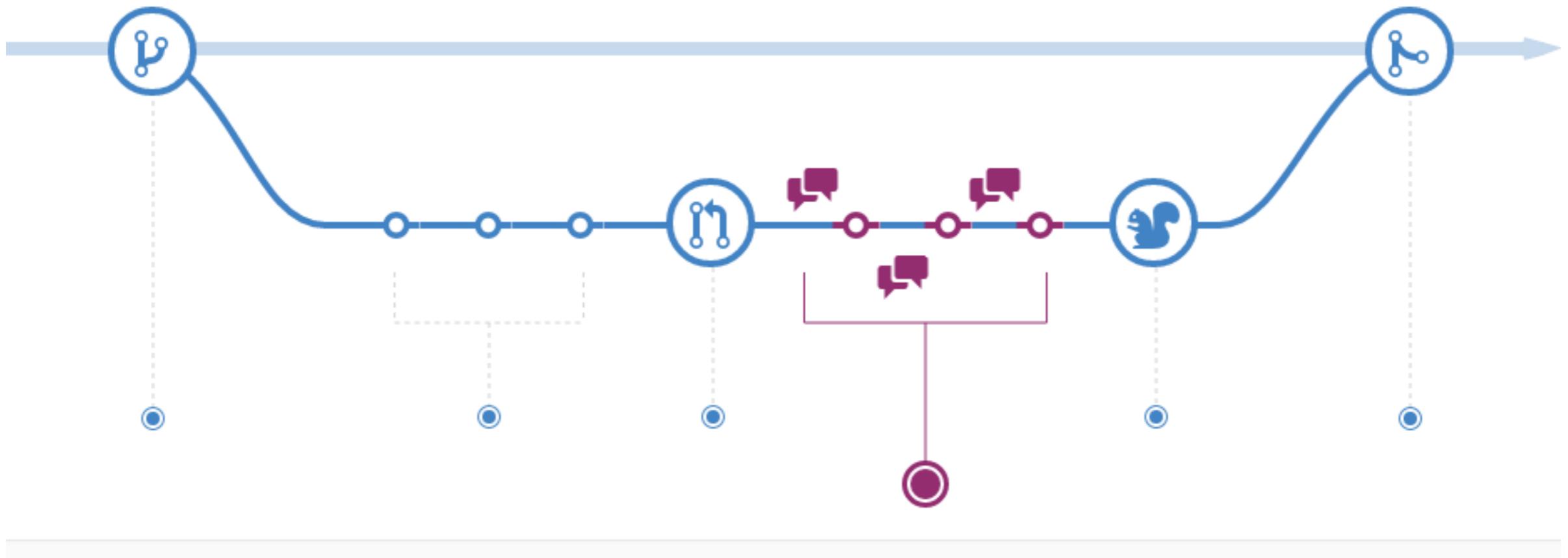
Step 2 - Add commits

GitHub Flow Explained



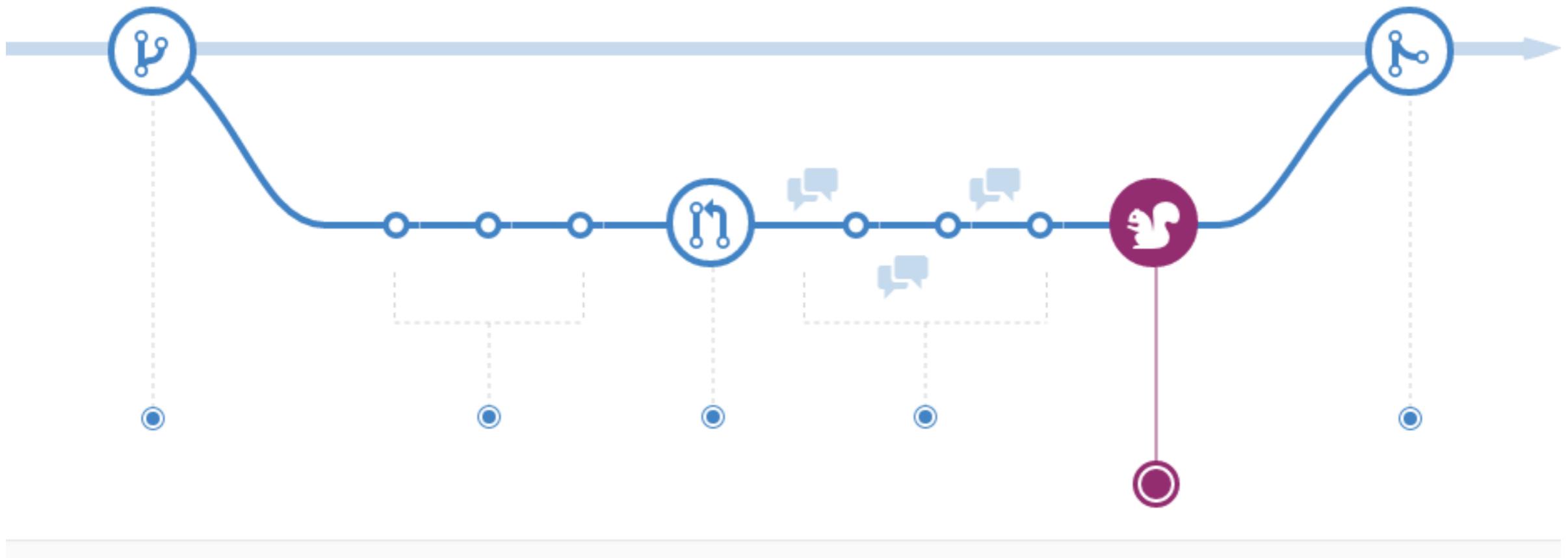
Step 3 - Open a Pull Request

GitHub Flow Explained



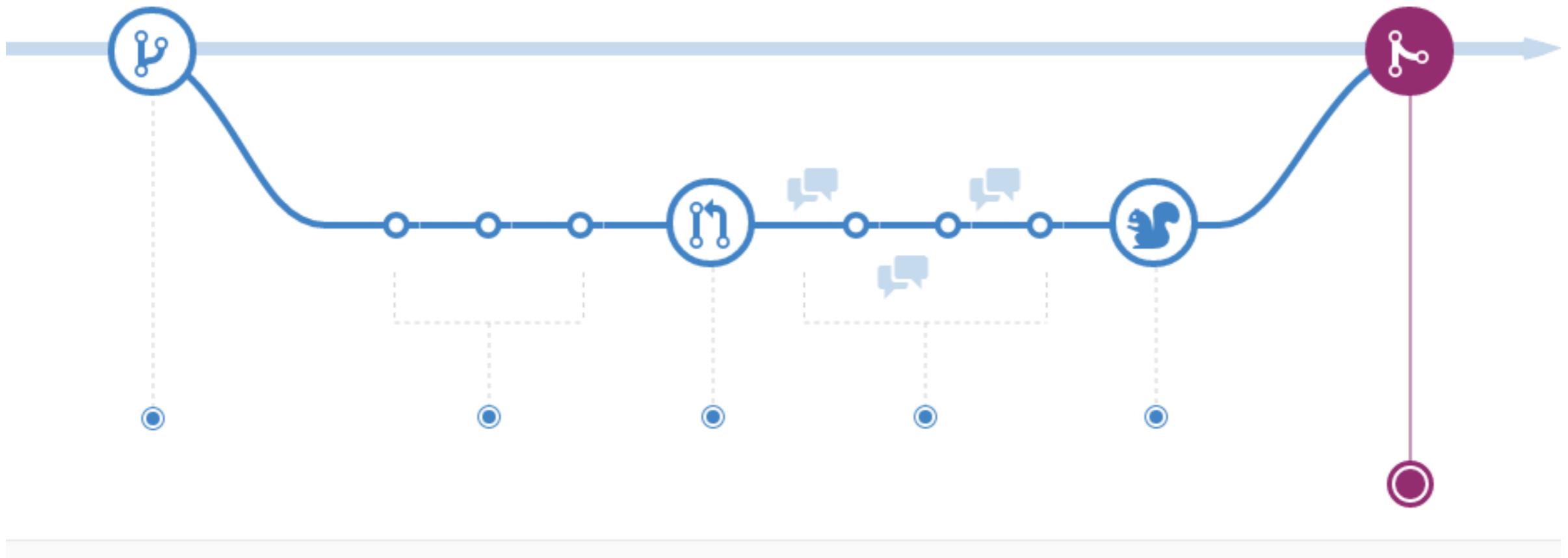
Step 4 - Code Review

GitHub Flow Explained



Step 5 - Ship It!

GitHub Flow Explained



Step 6 - Merge

GitHub Flow - Summary

- The master branch is always deployable
- Create a branch for each new feature
- Commit regularly with clear commit messages
- Open a Pull request (Peer Review, CI & ACR)
- Deploy **the branch** to production
- Merge the pull request

Deploying the Branch

We deploy the feature branch manually:

```
envoy run deploy --branch=my_new_branch
```

After merging the Pull Request, SemaphoreCI runs:

```
envoy run deploy
```

- This resets production servers back to master

Review

- Use version control
- Manual processes cause accidents
- Make processes easy, repeatable and reliable
- Use CI to automate testing, code reviews and quality checks
- GitHub Flow reduces risks of deploying

Questions?

Twitter: @johnblackmore

Web: johnblackmore.com

Email: john@johnblackmore.com

Links and Inspiration

- [Enhancing Envoy Deployment](#)
- [One Step Symlink Switch](#)
- [OpCache and symlink-based deployments](#)
- [Deploying with Envoy](#)
- [Laravel Envoy Deploy](#)
- [An Envoyer-like deployment script using Envoy](#)