

Workshop

開源・工作坊

WORKSHOP



H4



COSCUP 開源工作坊： Git Workflows

Hacking Thursday
Carl Su

甚麼是

Hacking Thursday ?

- 幾位好友相約做專案，前身為 Skalde
- Hacker 愈來愈多，約好星期四晚上固定聚會
- 專案在 COSCUP 後結束，改名 Hacking Thursday
- 成員把 H4 帶到上海，因此上海也有聚會

在這裡，您可以.....

- ✓ 分享技術上的新發現
- ✓ 工作上遇到難題卡關
- ✓ 找人一起討論技術
- ✓ 發起開源自由專案
- ✓ 交換分享工作機會



Welcome to HackingThursday

聚會地點：

餐廳：伯朗咖啡（建國店）

地址：台北市大安區建國南路一段 166 號 2 樓

(捷運忠孝新生站三號出口，沿忠孝東路走至建國南路右轉)

時間：7:30pm ~ 10:00pm

Restaurant : Mr. Brown Coffee (Jian Guo)

Venue : 2nd Floor, No. 166, Sec. 1, Jianguo S. Rd., Da-an Dist., Taipei City

(MRT Zhongxiao Xinsheng Station, Entry 3)

Time : 7:30pm ~ 10:00pm

詳細資訊請參見 => [聚會通告](#)

<https://hackingthursday.hackpad.com/>

<https://www.facebook.com/groups/hackingday/>

課程安排

- 集中式管理
- 功能分支
- Git Flow
- GitHub Flow
- GitLab Flow

Git 觀念複習

甚麼是 bare repository ?

- central repository
- 不能直接進行操作
- 通常都是以 .git 結尾
- GitHub 上面存的都是 bare repository

甚麼是工作目錄？

- working directory
- local repository
- `git clone` 抓回來的目錄
- `git init` 可將現有目錄轉為 Git repository

Git 同步機制

- git remote: 管理 bare repository 連結
- git fetch: 將 remote 分支同步到本地
- git pull: `git fetch` 加上 `git merge`
- git push: 提交變更到指定的 remote



課程講義連結

<https://goo.gl/YXhzdV>

初始環境設定

- 在 **Documents** 底下建立目錄：
 - `coscup2015_workshop`
- 建立 `bare repository`：
 - `h4_workshop.git`
- 開兩個終端畫面，初始化工作目錄（不加`.git`）
 - `john_h4_workshop`, `mary_h4_workshop`

集中式管理

使用 merge

John 新增 README 後 push

Mary 新增 LICENSE 後 push

Rejected (fetch first)

```
To /home/hackingthursday/Documents/h4_workshop.git  
! [rejected]          master -> master (fetch first)
```

Mary 執行 fetch 後 push

Rejected (non-fast-forward)

```
To /home/hackingthursday/Documents/h4_workshop.git  
! [rejected]          master -> master (non-fast-forward)
```

Mary 執行 merge 後 push

Merged (non-fast-forward)

Merge made by the 'recursive' strategy.

```
 README.md | 1 +  
 1 file changed, 1 insertion(+)  
 create mode 100644 README.md
```

Graph	Description	Commit	Author
	master 2 ahead Merge remote-tracking branch 'origin/master'	2ad38cf	Mary <mary@ha...
	origin/master Add a README file	6b65a07	John <john@hacki...
	Add the MIT license	7997616	Mary <mary@hack...

fast-forward v.s. no fast-forward

Fast-forward

- 不保留分支紀錄
- 只保留提交紀錄
- 直接移動分支指標
- 可以想成 rebase

No fast-forward

- 保留分支紀錄
- 保留提交紀錄
- 進行合併 (merge)
- 產生新的 commit

集中式管理

使用 pull --rebase

John 改了 LICENSE 後 push

Mary 改了 README 後 push

Rejected (fetch first)

```
To /home/hackingthursday/Documents/h4_workshop.git  
! [rejected]           master -> master (fetch first)
```

Mary 執行 pull --rebase 後 push

Merged (pull --rebase)

First, rewinding head to replay your work on top of it...
Applying: Add agenda section in README

Graph	Description	Commit	Author
	master 1 ahead Add agenda section in README	26d2eb3	Mary <mary@ha...
	origin/master Update year and copyright holders	059f401	John <john@hacki...
	Merge remote-tracking branch 'origin/master'	2ad38cf	Mary <mary@hack...
	Add a README file	6b65a07	John <john@hacki...
	Add the MIT license	7997616	Mary <mary@hack...

課程回顧

- Git 同步機制有哪些？
- fetch 和 pull 有何不同？
- Fast-forward 和 no fast-forward 有何不同？
- Bare repository 和 working directory 有何不同？
- merge 和 rebase 的使用時機為何？

學習資源

- 互動練習
 - Try Git
 - Learn Git Branching
- 教學
 - Pro Git
 - 連猴子都能懂的Git入門指南

功能分支

feature branch

功能分支

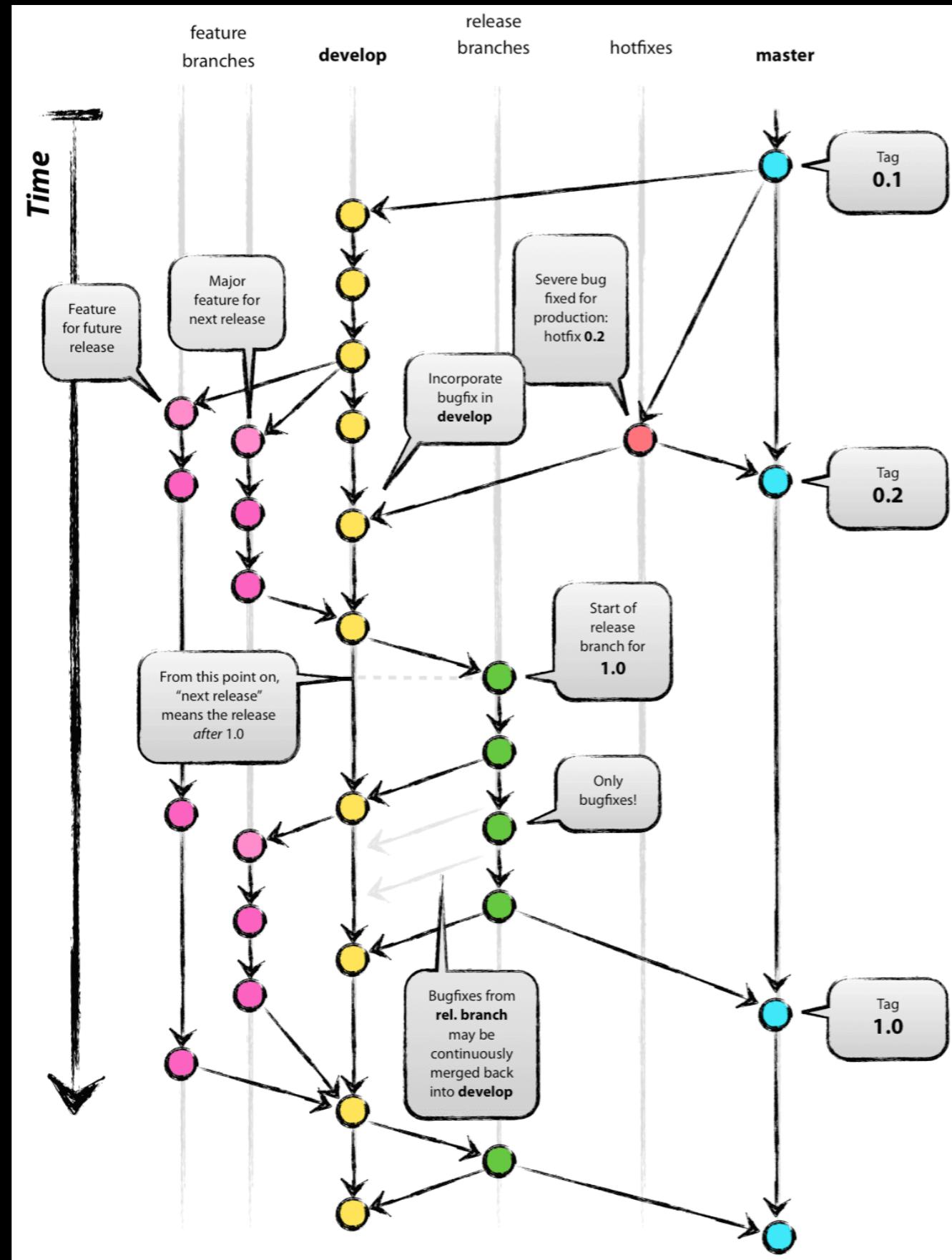
- 從主分支出來，分支名稱與該功能相關
- 暫時不必考慮衝突，專注完成某項功能
- 可以保留開發過程紀錄，包含分支名稱
- 可以在回到主分支之前進行修改與調整
- 建議您 Git 多人協作一定要用功能分支

課程回顧

- 集中式管理和功能分支在使用上有何不同？
- 功能分支與集中式管理相比，優勢在哪裡？
- 功能分支應該在多久的時間內回到主分支？

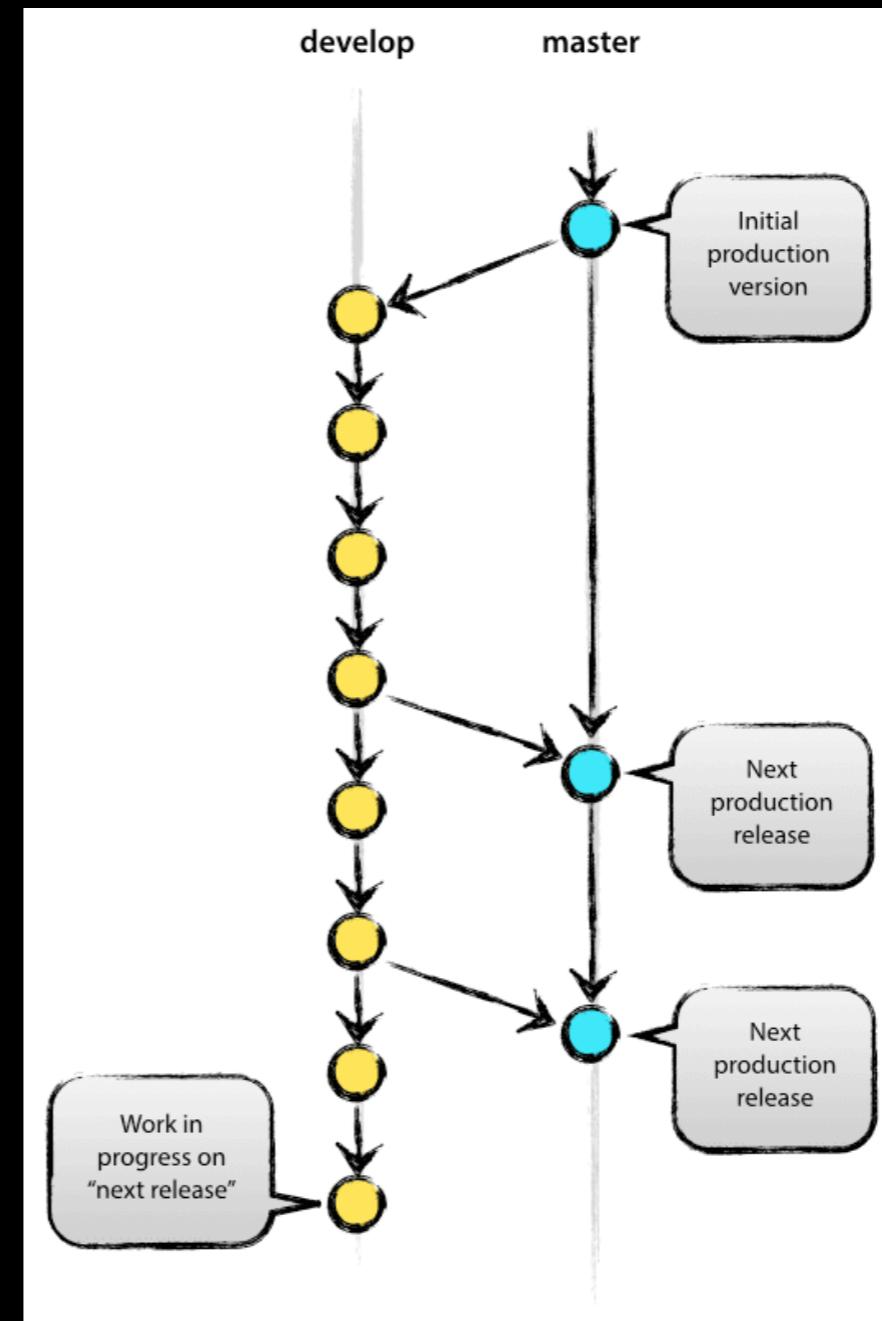
Git Flow

A successful Git branching model



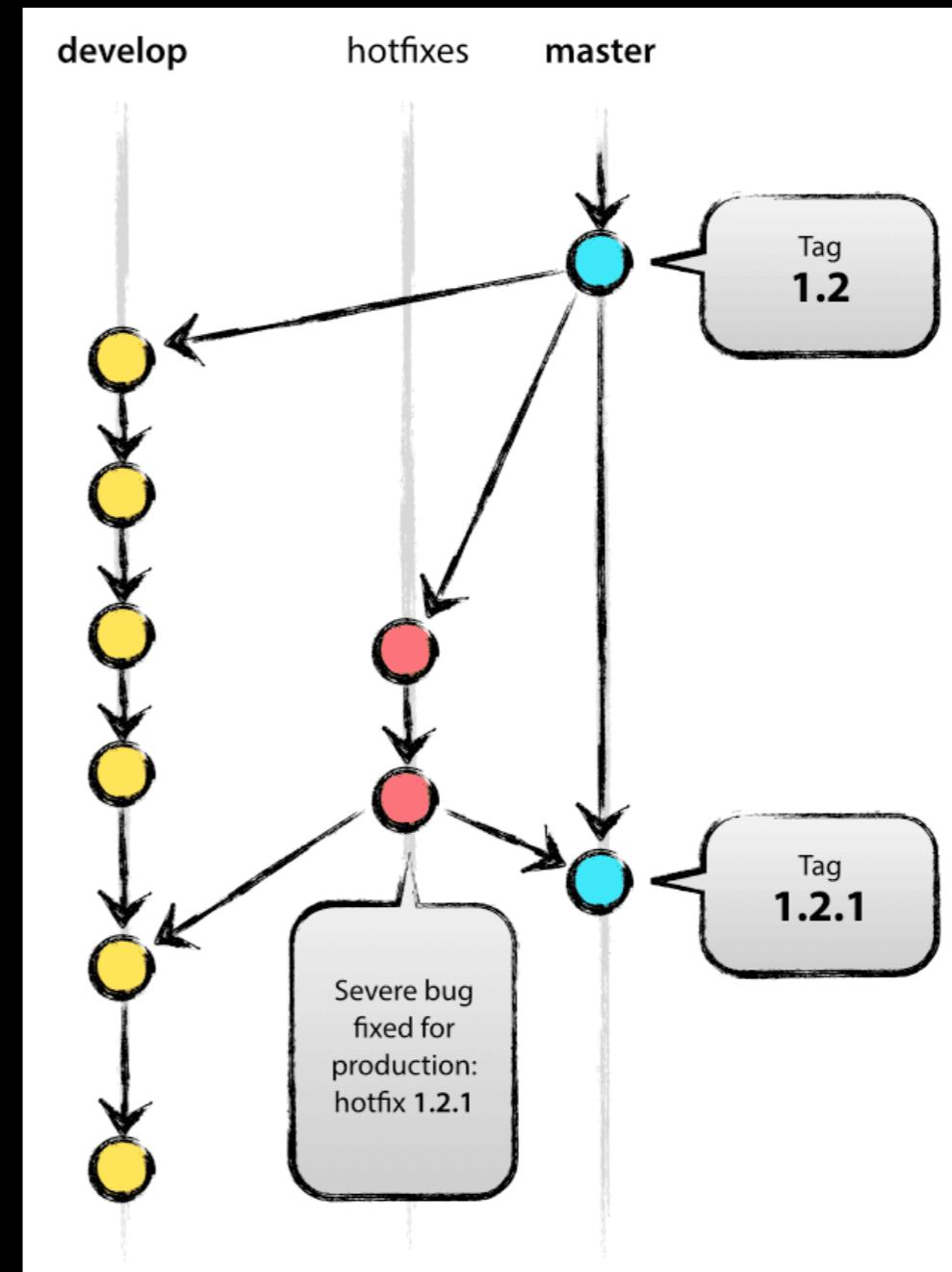
主線分支

- master
- develop



支援分支

- Feature branches
- Release branches
- Hotfix branches



Feature branches

- 從 develop 開分支
- 必須併回 develop
- 命名規則：
 - 第一種：feature/<featureA>
 - 第二種：描述式名稱（例：add-copyright-holders）

Release branches

- 從 develop 開分支
- 必須併回 develop 或 master
- 命名規則：
 - release-* 或 release/*

Hotfix branches

- 從 master 開分支
- 必須併回 develop 和 master
- 命名規則：
 - hotfix-* 或 hotfix/*

課程回顧

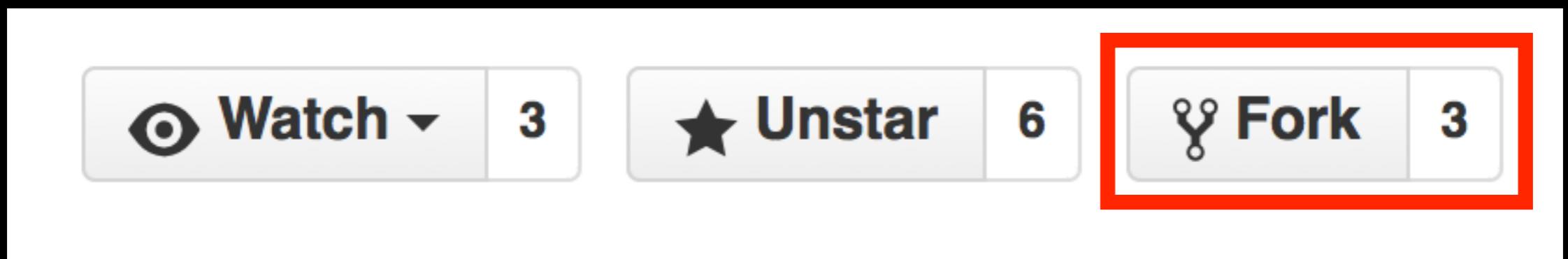
- develop 分支的用途是甚麼？
- 主線分支和支援分支有何差異？
- 為何要有 release 分支？
- feature 分支和 hotfix 分支有何不同？
- 最終 release 會出現在哪一個分支？

GitHub Flow

Introduced by GitHub

GitHub Flow

- fork / create a branch
- add commits
- open a pull request
- discuss and review your code
- deploy and merge



Fork repository or create a branch

-o Commits on Oct 13, 2013

	Update the release notes of Thunderbird to 24.0.1	...		2410192	
	Apply CODE tags to TB 24.0 release notes	...		9cf8c07	

Add commits

A screenshot of a GitHub repository page for the branch `fix-missing-de...`. A red box highlights the green 'Compare' button in the top navigation bar. Below the header, a message states 'This branch is 5 commits ahead of amueller:master.' To the right are links for 'Pull request' and 'Compare'. A commit card is shown with the message 'Add 'Pillow' to fix missing dependencies' by user `bcbcarl` 5 hours ago. The latest commit hash is `1f4c31cf82`.

Branch: `fix-missing-de...` [scipy_2015_sklearn_tutorial](#) / +

This branch is 5 commits ahead of amueller:master.

Add 'Pillow' to fix missing dependencies

`bcbcarl` authored 5 hours ago

latest commit `1f4c31cf82`

Open a pull request

amueller commented 23 hours ago

ok. So you should add pillow, not scikit-image.

Owner

bcbcarl added some commits 5 hours ago

- Revert "Add scikit-image dependencies in requirements.txt" ... a66c00b
- Revert "Fix missing dependencies" ... e74ab85
- Add 'Pillow' to fix missing dependencies 1f4c31c

bcbcarl changed the title from **Fix missing dependencies** to **Add 'Pillow' to fix missing dependencies** 5 hours ago

bcbcarl commented 5 hours ago

Okay. I reverted all changes and added 'Pillow' to `check_env.py`.

Discuss and review your code

bcbcarl commented on Oct 13, 2013

Add Thunderbird 24.0.1 release notes

bcbcarl added some commits on Oct 13, 2013

- Update the release notes of Thunderbird to 24.0.1 ... 2410192
- Apply CODE tags to TB 24.0 release notes ... 9cf8c07

irvin merged commit **9d5f581** into `moztw:master` on Oct 13, 2013

Revert

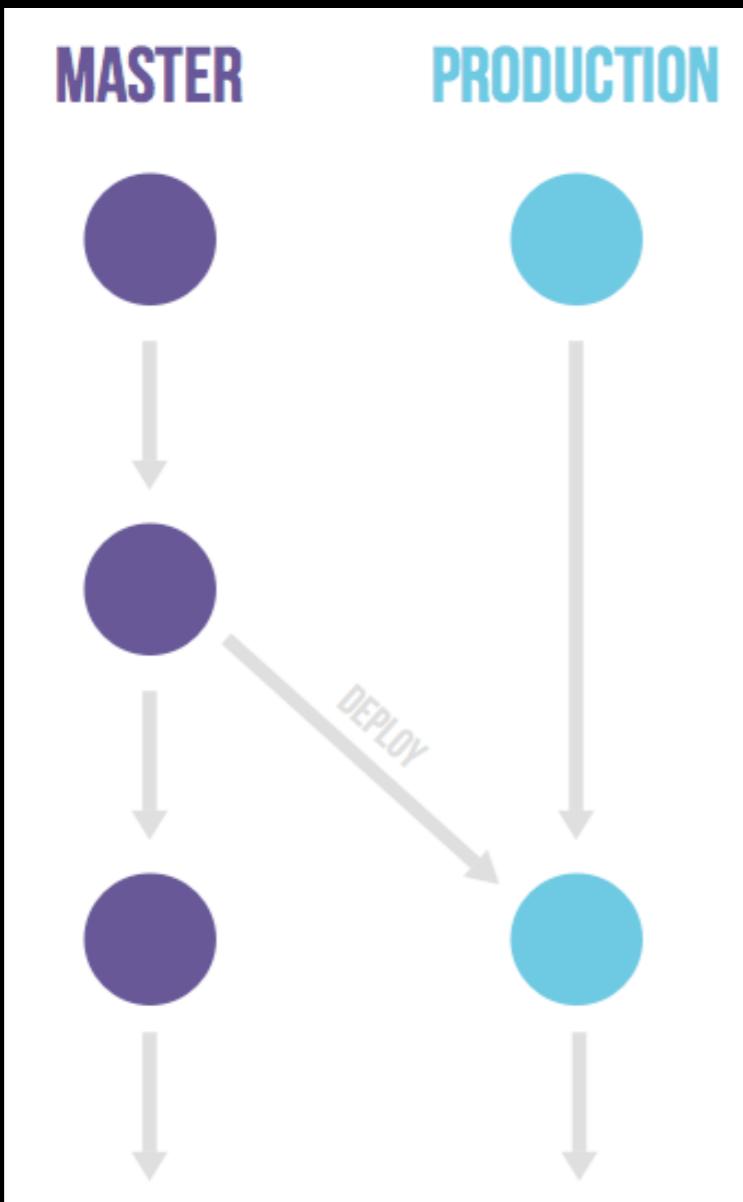
The screenshot shows a GitHub commit history. At the top, a comment from user 'bcbcarl' is shown, followed by a commit message 'Add Thunderbird 24.0.1 release notes'. Below this, another comment from 'bcbcarl' shows two commits: 'Update the release notes of Thunderbird to 24.0.1' and 'Apply CODE tags to TB 24.0 release notes'. Both commits have their commit hash and pull request number listed next to them. At the bottom, a merge commit from user 'irvin' is shown, merging commit '9d5f581' into the 'moztw:master' branch. A 'Revert' button is visible on the right side of this merge commit.

Deploy and merge

GitLab Flow

Production / Environment / Release branch

Production branch



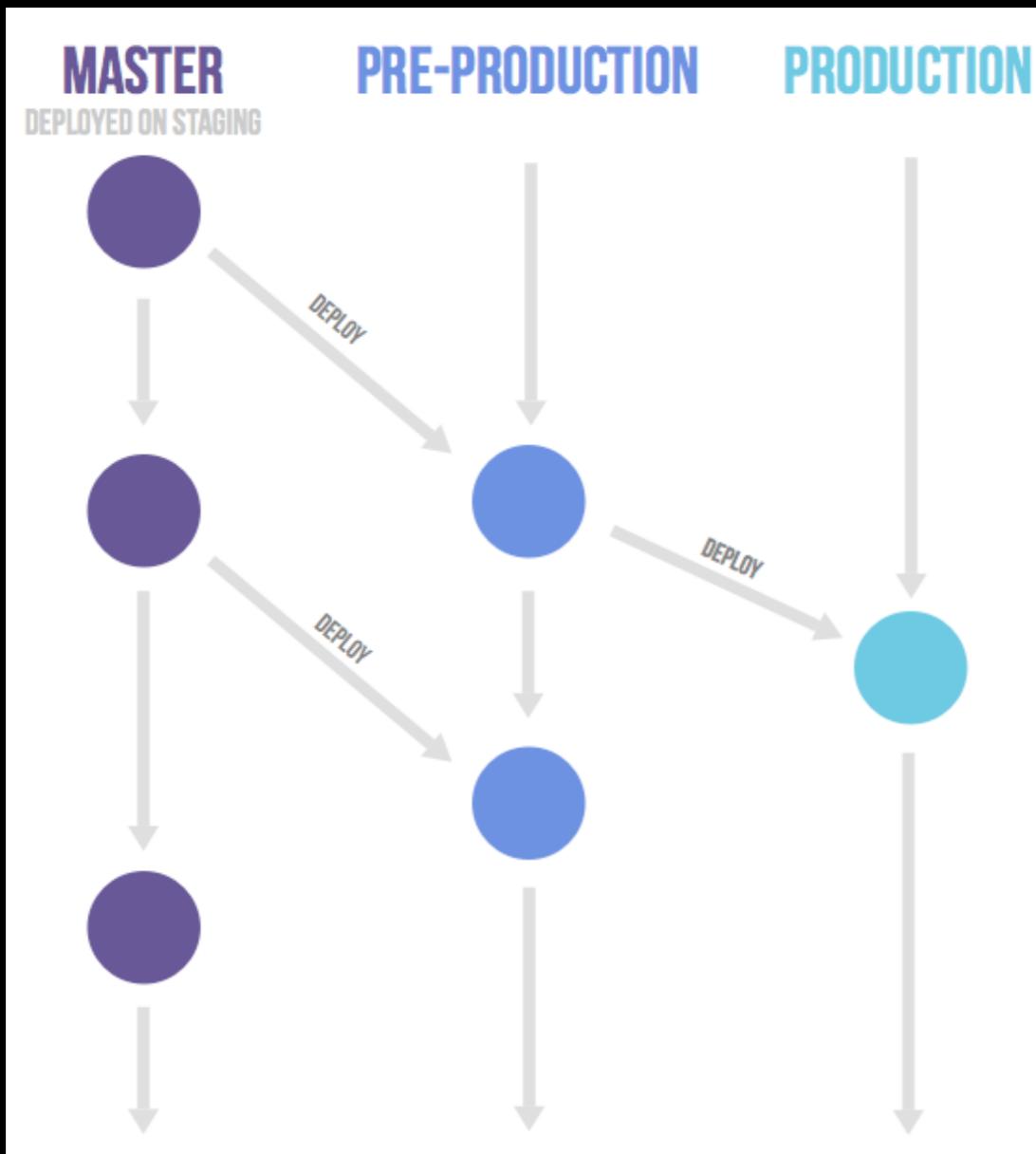
使用時機

- 主分支無法直接 deploy

特色

- 可更精確控制發佈時機
- production 等同 Git Flow 的 master 分支

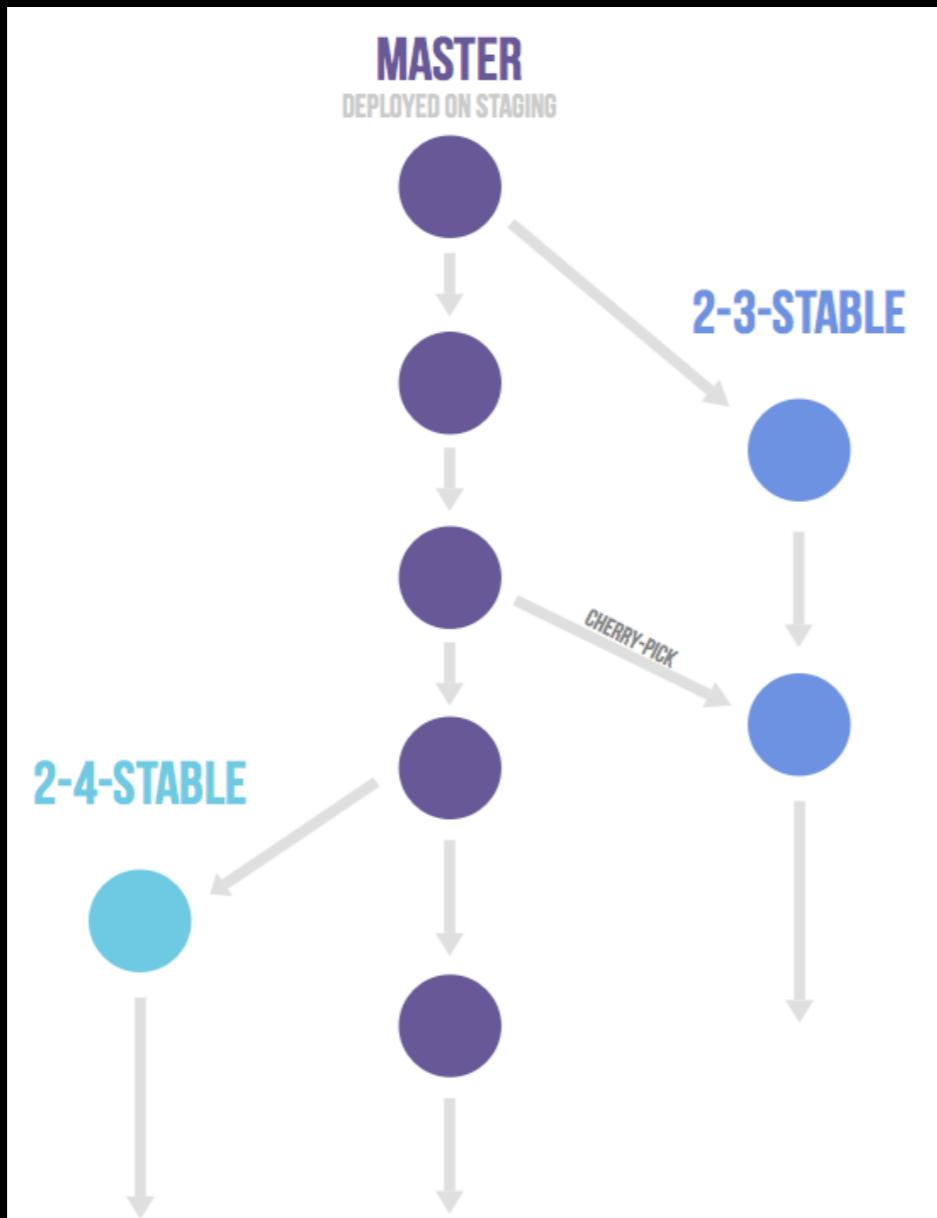
Environment branch



使用時機

- 需要額外測試環境
- 特色
- 主分支是暫存區
- 確保上線前測試

Release branch



使用時機

- 需要對外發佈軟體

特色

- hotfix 從主分支 cherry-pick
- 通常沒有 production 分支

課程回顧

- 本課程介紹了哪些工作流程？
- 這些工作流程的差異在哪裡？
- 哪個工作流程適合您的團隊？

參考連結

- <https://www.atlassian.com/git/tutorials/comparing-workflows/>
- <https://git-scm.com/book/en/v2/Git-Branching-Basic-Branching-and-Merging>
- <http://nvie.com/posts/a-successful-git-branching-model/>
- <https://guides.github.com/introduction/flow/>
- <https://about.gitlab.com/2014/09/29/gitlab-flow/>