ط

تعامل Git مع Remote Repository

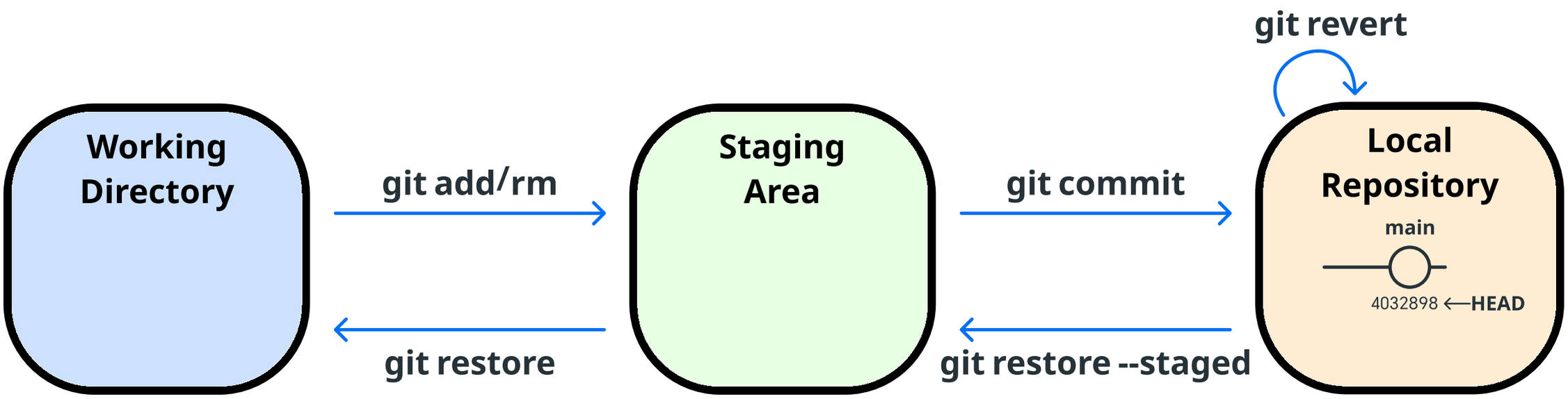
سنتعلم كيف يتعامل الـ Git مع الـ Remote Repository
وإجابة السؤال الأزلي ما الفرق بين الـ GitHub والـ Git

Ahmed El-Tabarani
Back-End Developer

ط

أولًا ما هو الـ Local Repoistory ؟

ببساطة يستخدمه الـ Git ليحتفظ ويتابع كل التعديلات
الموجودة في مشروعك على جهازك الشخصي
ويحتفظ بها على هيئة مجموعة من الـ commit تشير
إلى تعديلات التي قمت بها في المشروع على مر الزمن



هذا ما كنت تتعامل معه في جهازك الخاص بشكل فردي
لكن إن كنت ضمن فريق وكل شيء يقوم بتعديل ويضيف شيء
ما في المشروع
فكيف ستتواصلوا وتتشاركوا هذه التعديلات ؟

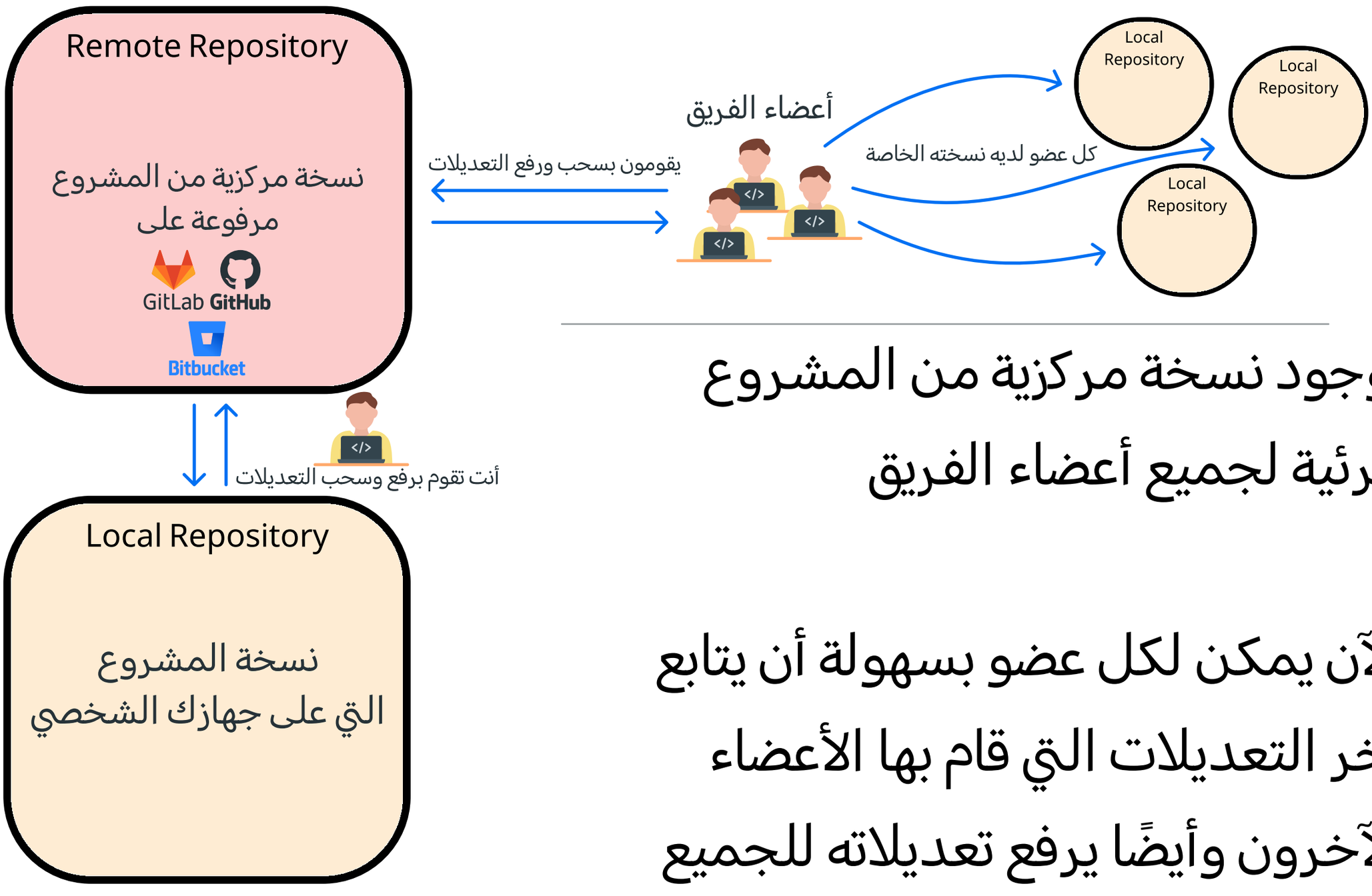
Ahmed El-Tabarani
Back-End Developer

السر في الـ Remote Repository

هو كالـ Local Repository لكنه نسخة رئيسية من المشروع يتم استضافتها في مكان ما

مثل GitHub أو GitLab أو BitBucket أو غيرهما

بالتالي سيكون لدينا نسخة مركزية من المشروع مرفوعة ومرئية لجميع أعضاء الفريق



بوجود نسخة مركزية من المشروع مرئية لجميع أعضاء الفريق

الآن يمكن لكل عضو بسهولة أن يتابع آخر التعديلات التي قام بها الأعضاء الآخرون وأيضًا يرفع تعديلاته للجميع

Ahmed El-Tabarani
Back-End Developer

ط

السؤال الأزلي: ما الفرق بين الـ GitHub والـ Git

- الـ Git هو أداة تستخدمها في جهازك الشخصي لإدارة نسختك من المشروع
- يتحكم وينظم ويسجل كل شيء يحدث في نسختك في الـ Local Repository

---

- الـ Github كغيره من المنصات التي تقوم باستضافة نسخة رئيسية من المشروع، تكون النسخة المركزية للمشروع وتسمى الـ Remote Repository
- هذه النسخ تكون متاحة لجميع الناس أو للفريق الخاص بك
- كل شخص يستطيع مشاركة إضافاته وتعديلاته في مكان واحد مشترك
- بها خدمات تساعد الفرق والشركات على تنظيم مشروعاتهم
- واجهه بسيطة للمشروع وكل ما يحصل فيه
- تسهل رؤية التعديلات ومراجعتها
- تسهل مناقشة التعديلات والتعليق عليها وعرض اقتراحات
- تسهل عملية رفض أو قبول ودمج التعديلات

Ahmed El-Tabarani
Back-End Developer

طـ
أهم أوامر الـ Git في التعامل مع الـ Remote Repository

سنقوم بعرض مثال ثم نقوم بشرح أهم الأوامر

من خلال هذا المثال

سنفترض أننا دخلنا إلى شركة ما وهذه الشركة لديها

مشروع يتم استضافته على GitHub

لنفترض أن الرابط الخاص بالـ Remote Repository
الخاص بمشروع الشركة هو
github-url-project-remote-repository.git

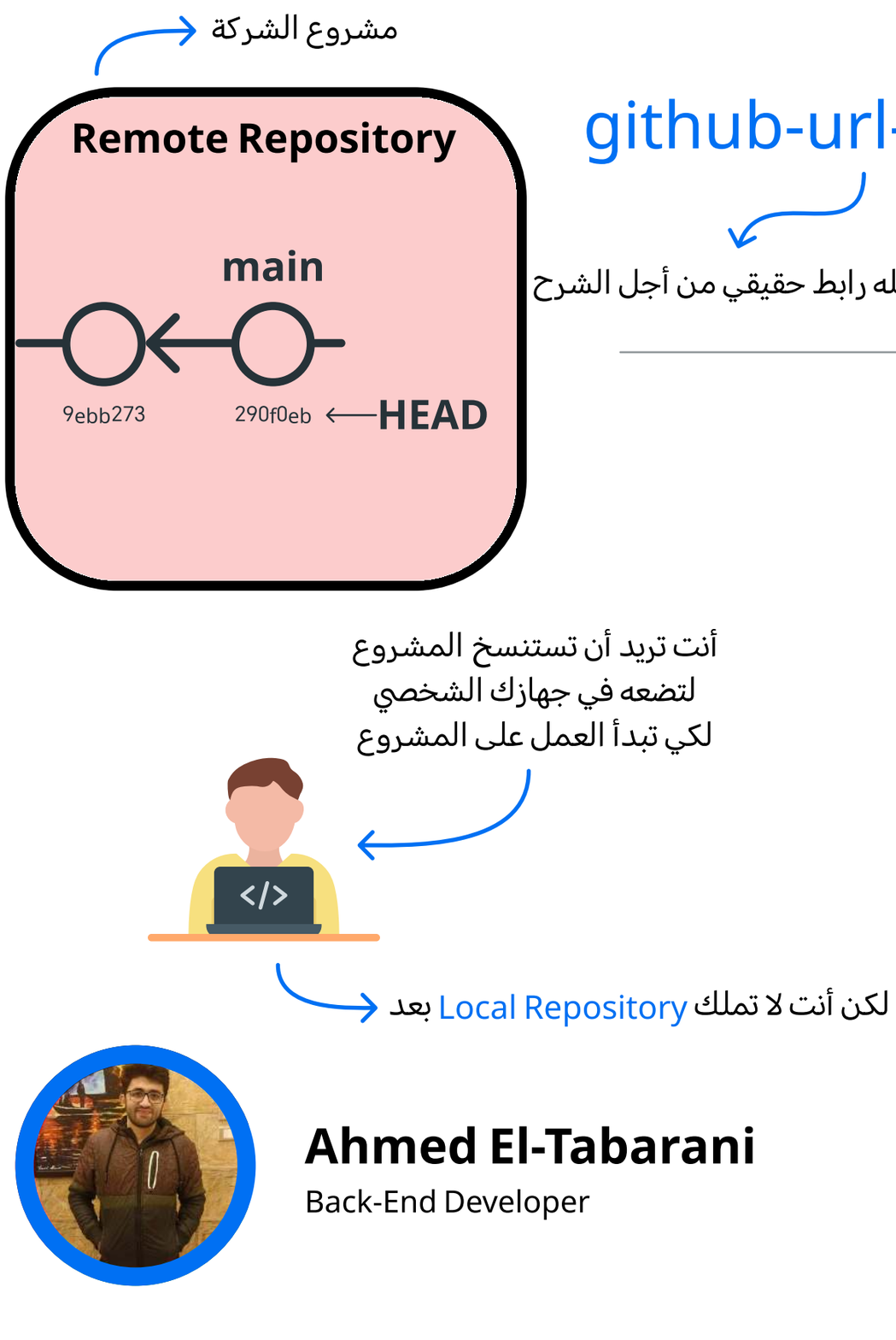
لنتخيله رابط حقيقي من أجل الشرح

الـ Git يقدم لنا أوامر متنوعة تساعدنا

على ربط الـ Local Repository

بالـ Remote Repository

وسنتعرف عليهم تاليًا



Ahmed El-Tabarani
Back-End Developer

git clone

تستخدم لنسخ وتحميل نسخة من
الـ Remote Repository على جهازك الشخصي
ويصبح Local Repository بالنسبة لك
وتُنشيء حلقة تواصل بينهما لسحب ورفع الإضافات والتعديلات

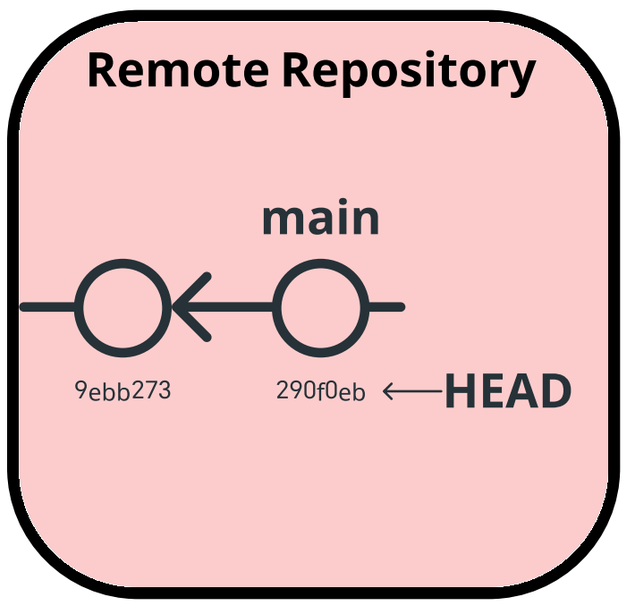
رابط الـ Remote Repository

اسم المجلد الذي سيتم نسخه فيه
والذي سيحتوي على الـ Local Repository

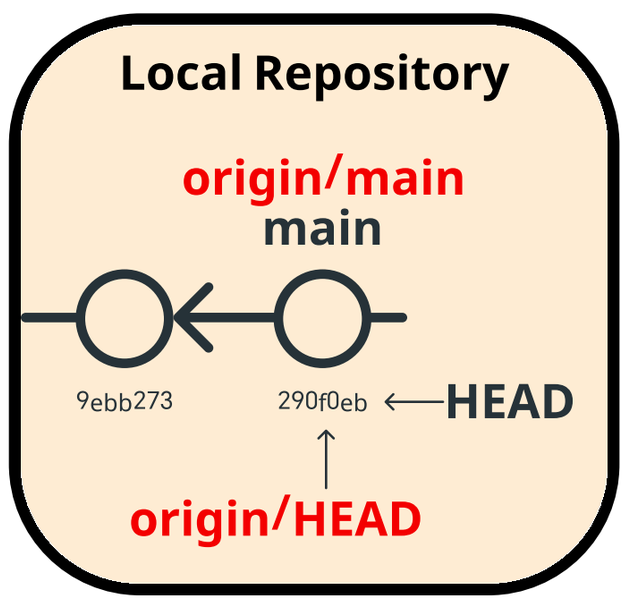
```
> git clone github-url-project-remote-repository.git project-local-repository
Cloning into 'project-local-repository'...
done.
```

أولًا أنت ستقوم بنسخ الملف

git clone remote-url folder-name



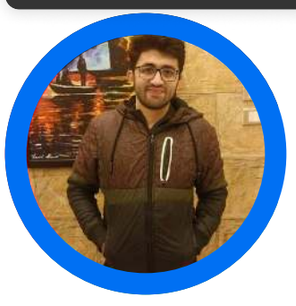
ثم سيتم إنشاء نسخة من المشروع لك
فيه git. والذي يمثل كل شيء بالنسبة للـ Git
كالـ Local Repository الخاص بك وغيره



لاحظ وجود origin/main و origin/HEAD دخل الـ Local Repository

وعند تنفيذ git status ستراه يقول لك
أننا على الـ main وهو متوافق مع origin/main

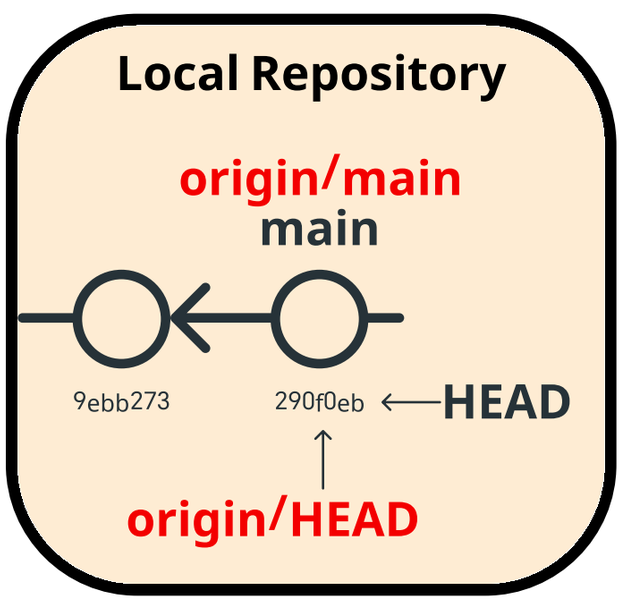
```
> git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```



Ahmed El-Tabarani
Back-End Developer

ما هو الـ origin ؟

هو يشير إلى الـ Remote Repository الذي ارتبطنا به
وهو يحتوي على الـ main والـ HEAD الخاصة بهذا
الـ Remote Repository
ويشار إليهم بـ origin/HEAD و origin/main
والـ origin هو اسم رمزي أو افتراضي وممكن أن تغيره إن أردت



ستجد أن الـ main الخاصة
بالـ Local Repository والـ origin/main
الخاصة بالـ Remote Repository يتطابقان

وأيضًا ستجد أن الـ origin/HEAD والـ HEAD
يقفان على نفس الـ commit

ملحوظة: الـ origin يشاور على آخر التعديلات في الـ Remote Repository
بحسب آخر المعلومات التي حصل عليها الـ Local Repository
بالتالي فيمكن أن يكون الـ origin يملك معلومات قديمة عن الـ Remote Repository
لأنه لا يحدث معلوماته بشكل تلقائي بل يجب عليك أن تحدثها بشكل يدوي

Ahmed El-Tabarani
Back-End Developer

git remote

أولًا كلمة remote نقصد بها

الـ Remote Repository الذي نرتبط به

ونحن بالفعل لدينا حاليًا remote يدعى origin تم إنشاءه

ويمكننا رؤيته عن طريق git remote

```
> git remote
origin
```

ولكي نعرف رابط الـ Remote Repository الذي تشير إليه

نقوم بتنفيذ git remote -v

```
> git remote -v
origin github-url-project-remote-repository.git (fetch)
origin github-url-project-remote-repository.git (push)
```

كما ترى لدينا فقط origin والرابط الـ Remote Repository

الذي يشير إليه

ومسموح لنا فقط بتنفيذ fetch لجلب التعديلات منه وpush لرفع

التعديلات إليه وسنرى هذا بالتفصيل

Ahmed El-Tabarani
Back-End Developer

git remote add remote-name remote-url

إذا كان لديك Local Repository مشروع
وهو لا يرتبط أو يملك أي Remote Repository بعد
لأنه مجرد مشروع تعمل عليه بشكل خاص على جهازك الشخصي

ثم قررت رفعه على GitHub لتكون هناك نسخة مركزية
تكون هي الـ Remote Repository
حينها ستقوم بإضافة remote بشكل يدوي عن طريق تنفيذ الأمر
التالي **git remote add** remote-name remote-url

ويفضل أن تختار origin كاسم لهذا الـ remote لأنه الاسم
الشائع والمتعارف عليه

هذا في حالة أنك لديك Local Repository بالفعل وتريد أن تربطه مع Remote Repository
أما لو هناك Remote Repository وتريد أن تستنسخه إلى Local Repository لديك
فنستخدم git clone كما فعلنا سابقً

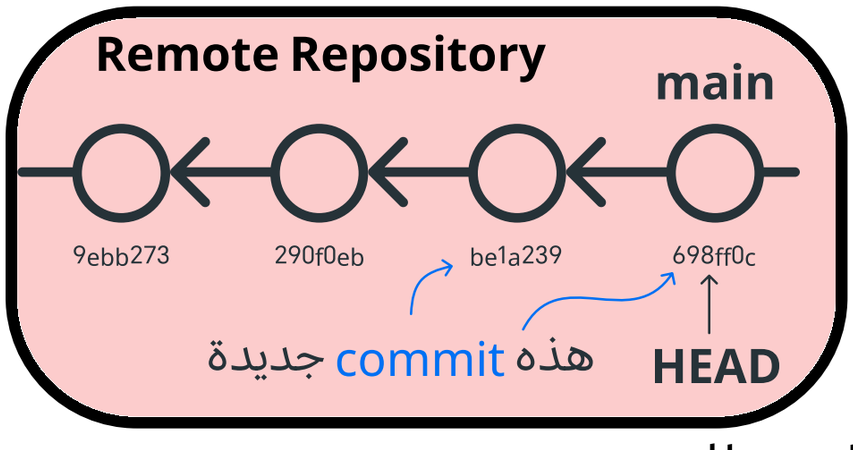
Ahmed El-Tabarani
Back-End Developer

ط

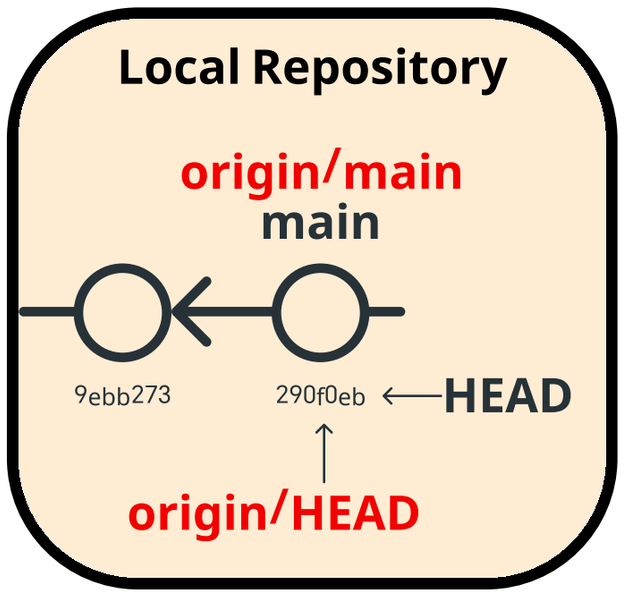
git fetch

نستخدمه لكي نحدث الـ origin ونسحب آخر الإضافات
والتعديلات التي حدثت في الـ Remote Repository
والتي ليست في الـ Local Repository بعد
لكنه لا يقوم بعمل merge لها في الـ HEAD

لنفترض أن عضو ما في الفريق قام برفع بعض التعديلات في commit جديدة
على الـ Remote Repository



وبالطبع الـ Local Repository لن يستطيع
أن يعرف هذه التعديلات من تلقاء نفسه
وإن نفذت git status سيقول أنه متوافق تمامًا مع الـ origin
لكن هذا لا يعني أنه متوافق مع الـ Remote Repository

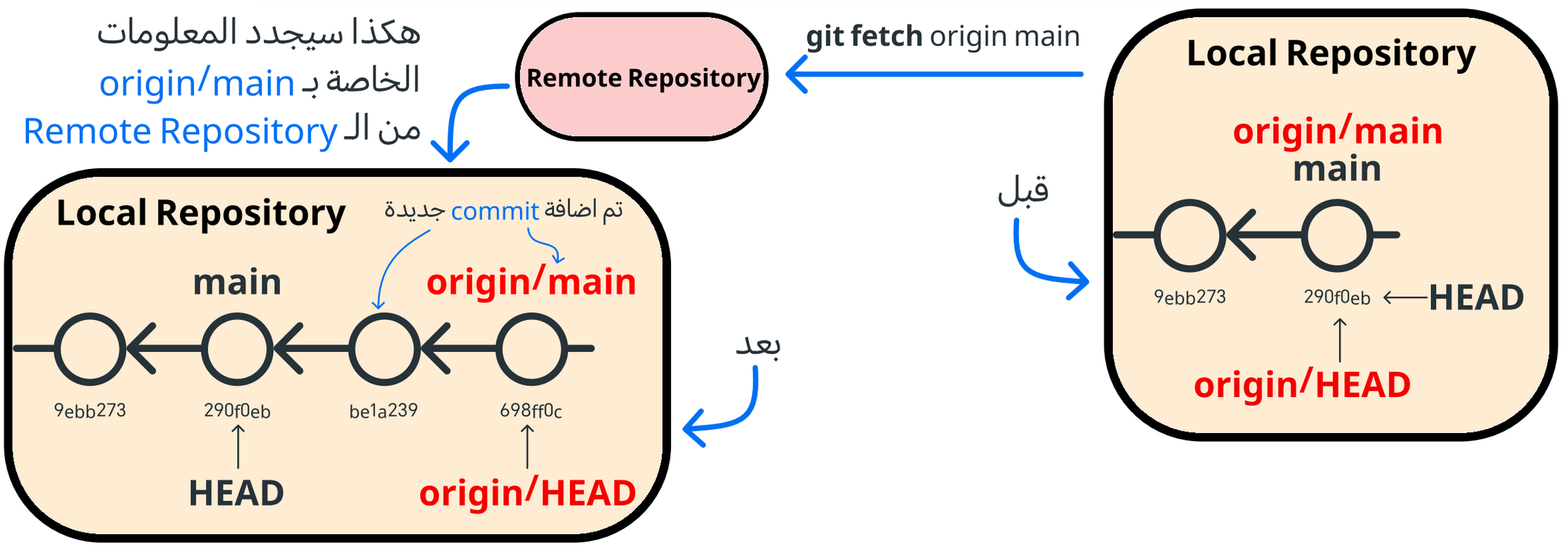


هام:
الـ origin لا يحدث معلوماته بشكل تلقائي
بل يجب عليك ان تحدثها بشكل يدوي
عن طريق تنفيذ الأمر git fetch

Ahmed El-Tabarani
Back-End Developer

نقوم بتنفيذ git fetch

```
> git fetch origin main
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 284 bytes | 0 bytes/s, done.
From github-url-project-remote-repository.git
   290f0eb..698ff0c  main       -> origin/main
```



لاحظ أنه قام بسحب أي commit موجود في الـ Remote Repository
وليس موجودًا في الـ Local Repository
وبعد أن سحبه قام بوضعه في الـ origin/main دون دمجه في الـ main

وأيضًا ستلاحظ أن الـ main متأخر على الـ origin/main ببعض الـ commit
ولدمج هذه الـ commit والتعديلات إلى الـ main نقوم بتنفيذ git merge

Ahmed El-Tabarani
Back-End Developer

git pull

مزيج بين git fetch و git merge في آن واحد

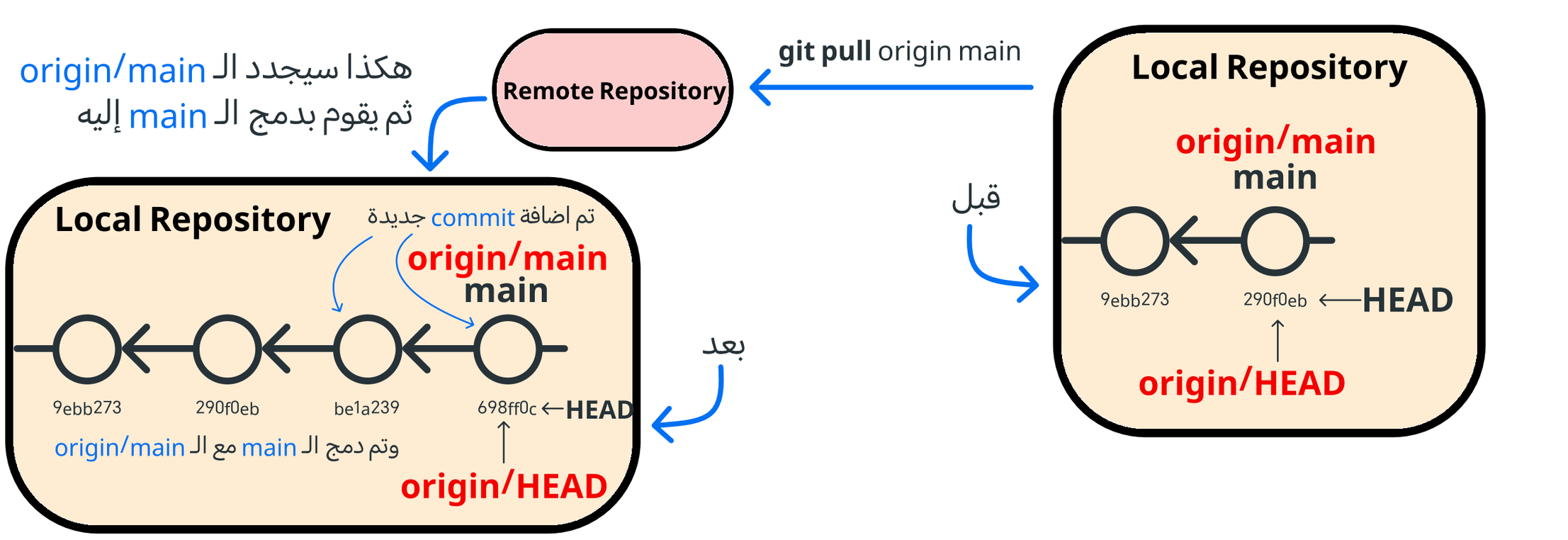
فهو يقوم بتجديد الـ origin ثم يقوم بدمجه

في الـ main مباشرةً في خطوة واحدة

```
> git pull origin main
From github-url-project-remote-repository.git
 * branch            main       -> FETCH_HEAD
Updating 290f0eb..698ff0c
Fast-forward
 article_3.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 article_3.txt
```

} جزء الـ fetch

} جزء الـ merge



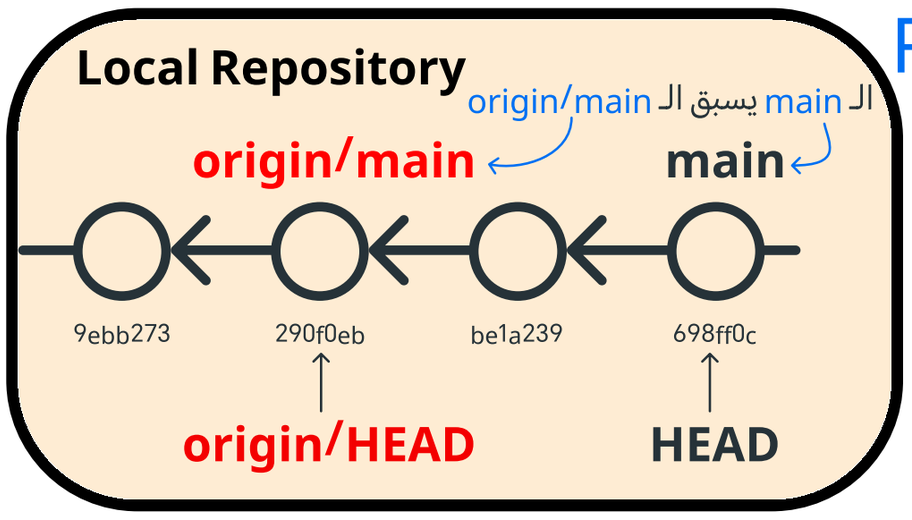
Ahmed El-Tabarani

Back-End Developer

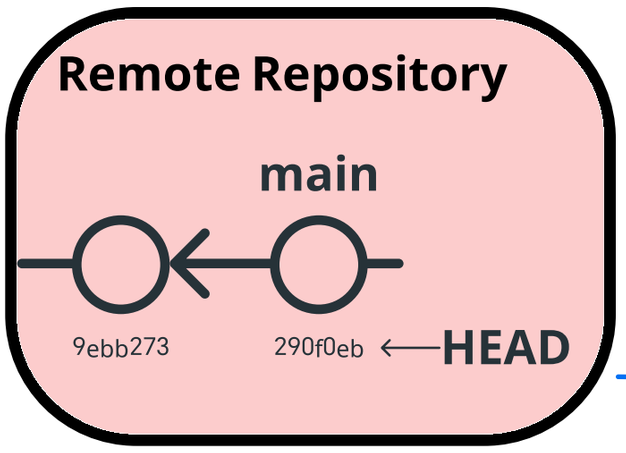
git push

نستخدمه لنقوم برفع التعديلات التي في
الـ Local Repository إلى الـ Remote Repository
وهو الوجه المعاكس للـ git pull

لنفترض أنك قمت بعمل بعض التعديلات في الـ Local Repository
الآن أصبحت لديك commit في الـ Local Repository
وليس موجودًا في الـ Remote Repository



الآن أعضاء الفريق الذين يعملون معك
لن يروا ما قمت بتعديله لأنه على
الـ Local Repository الخاص بك فقط



ولذا يجب عليك أن ترفع ما قمت به
إلى الـ Remote Repository، ليستطيع باقي أعضاء
الفريق سحبه ويروا هذه التعديلات التي قمت بها
في الـ Local Repository الخاصة بهم

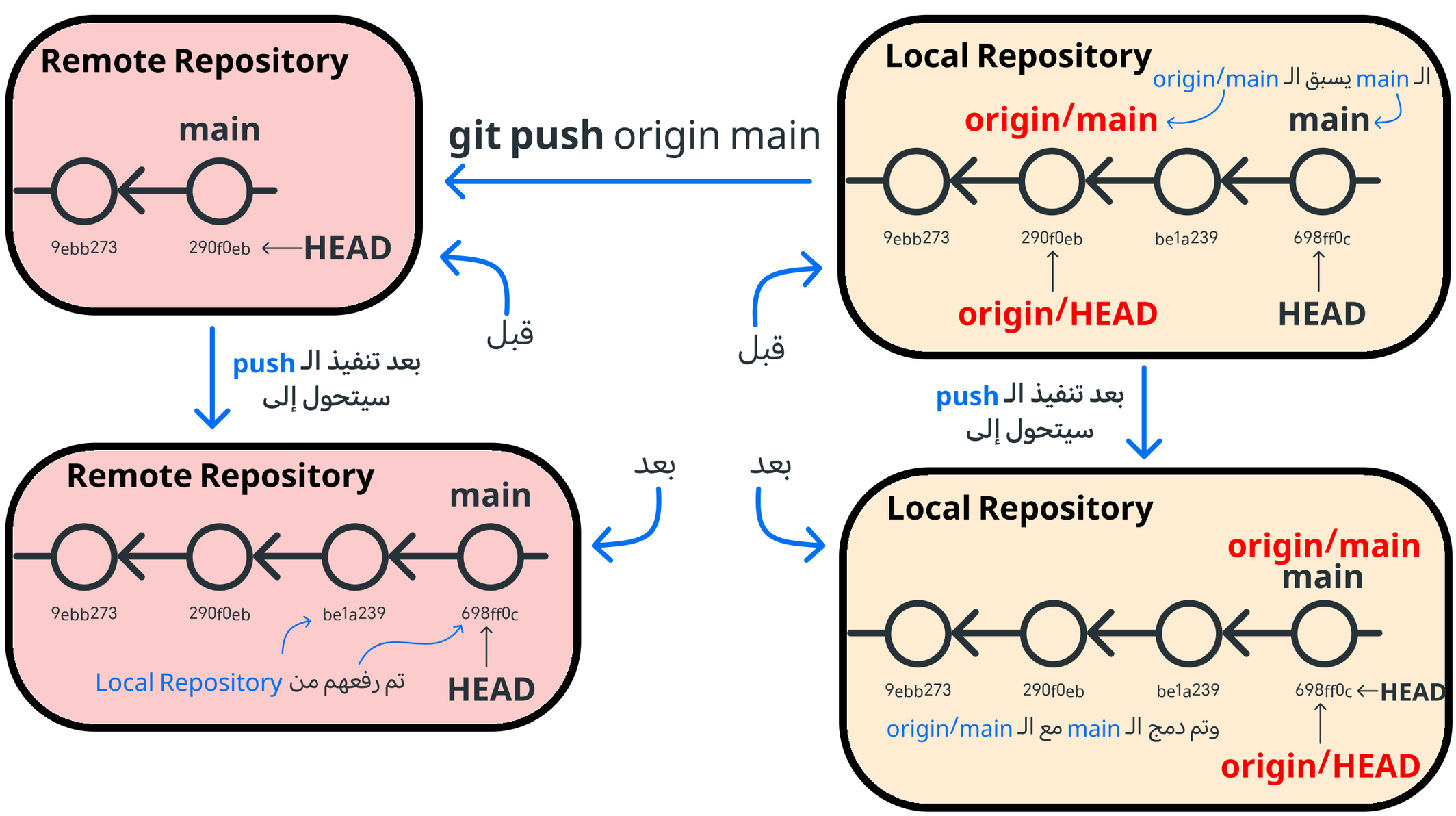
لا يدري ما الذي تقوم به في الـ Local Repository الخاص بك

هنا سنستخدم git push

Ahmed El-Tabarani
Back-End Developer

نقوم بتنفيذ git push

```
> git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 291 bytes | 145.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github-url-project-remote-repository.git
   290f0eb..698ff0c  main -> main
```

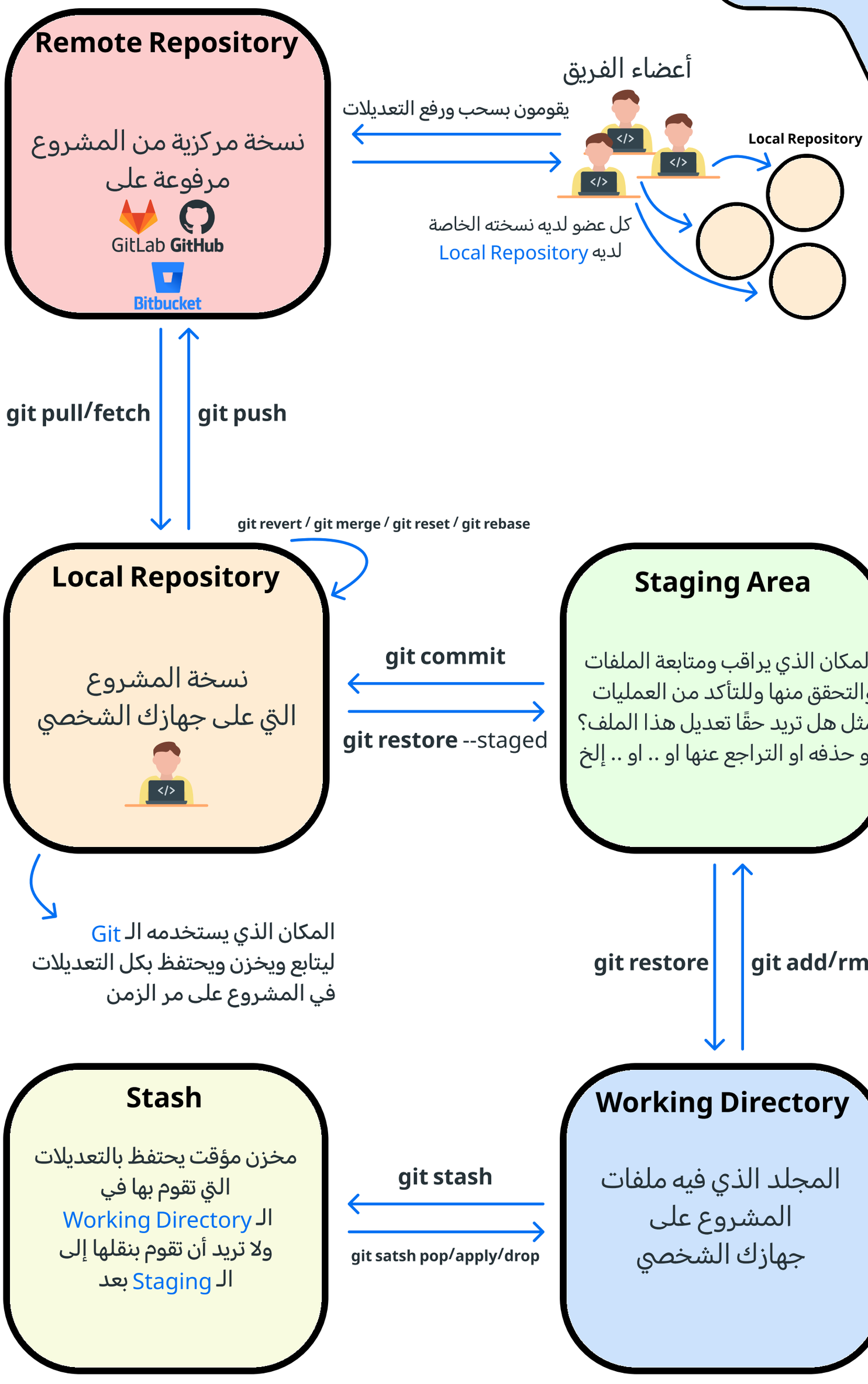


الآن قمنا برفع كل commit جديد لدينا من الـ Local Repository إلى الـ Remote Repository بنجاح

وسترى أن الـ git push قام أيضًا بعمل دمج للـ origin/main مع الـ main في الـ Local Repository

Ahmed El-Tabarani
Back-End Developer

ملخص مرئي بسيط



Ahmed El-Tabarani
Back-End Developer

طبعًا لن أوصيك بتفقد المقالة الكاملة 😄

tabarani.tk/articles/git-with-remote-repository

Ahmed El-Tabarani

Back-End Developer

أنا أحمد الطبراني، مهندس برمجيات SWE 😁

مبرمج متخصص في عالم الـ Backend ⚙️💻

أحب دائمًا أن أُشارك معرفتي المتواضعة مع الآخرين لعله عسى أن يستفيد شخص ما بما أكتبه 🤗
أكتب بعض المقالات عن أشياء مختلفة في عالم البرمجة، أرجوا أن تستفيدوا وتستمتعوا 😉

تابعني علي لينكدإن او علي مدونتي الشخصية