Git Branch Basics

What are Git branches?

Git branches are like separate “sub folder” where you work on a feature. When you are done with the feature, you join the feature back with the main folder.

- **Features:** Add new stuff.
- **Fixes:** Fix bugs.
- **Tests:** Try things out.

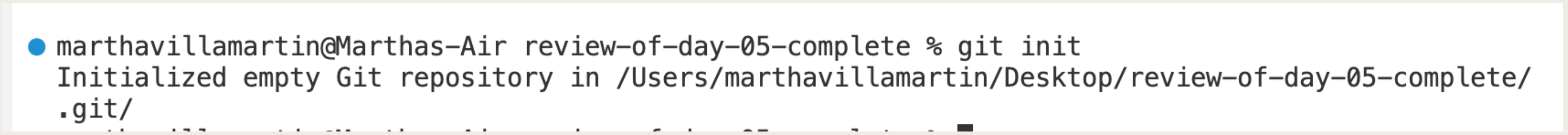
Branches keep your main code safe while you work on a new feature.

Exercise #1

- Set up a local Git repository for the “review-of-day-05-complete”.
- Add and commit the files to the local repository

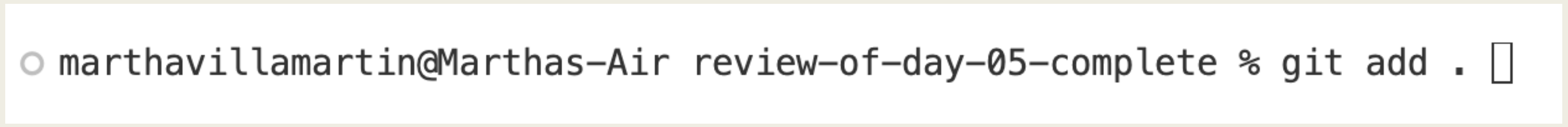
Steps to follow:

- Initialize a git repo in review-of-day-05-complete folder: Run git init in the terminal



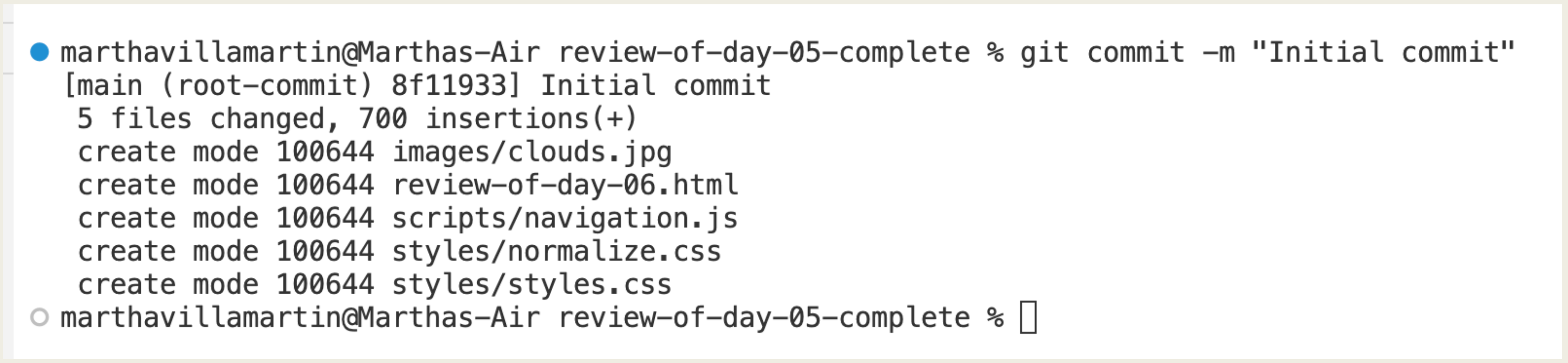
```
● marthavillamartin@Marthas-Air review-of-day-05-complete % git init
  Initialized empty Git repository in /Users/marthavillamartin/Desktop/review-of-day-05-complete/
  .git/
```

- Stage all the files: Run the git add . command



```
○ marthavillamartin@Marthas-Air review-of-day-05-complete % git add . ▯
```

- Commit the content of the review-of-day-05-complete folder with a message: Run git commit -m ”initial commit” command



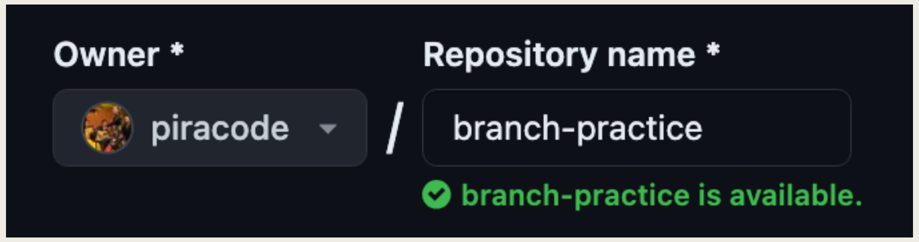
```
● marthavillamartin@Marthas-Air review-of-day-05-complete % git commit -m "Initial commit"
  [main (root-commit) 8f11933] Initial commit
   5 files changed, 700 insertions(+)
   create mode 100644 images/clouds.jpg
   create mode 100644 review-of-day-06.html
   create mode 100644 scripts/navigation.js
   create mode 100644 styles/normalize.css
   create mode 100644 styles/styles.css
○ marthavillamartin@Marthas-Air review-of-day-05-complete % ▯
```

Exercise #2

1. Create a remote repository on GitHub
2. Link the local repository to the remote
3. Push the commits from the local main branch to the remote repository

Initialize & Push to Remote Repo

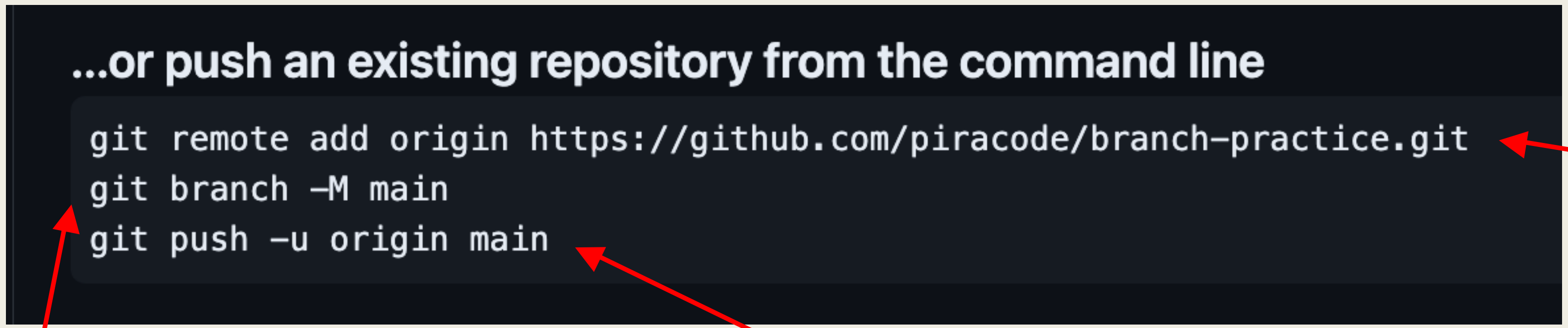
1. In GitHub create a new repository :



Owner * piracode / Repository name * branch-practice

branch-practice is available.

2. Link the local repository to GitHub & push the initial commit:
   - *Copy and paste the provided GitHub commands in your local repo's terminal to connect and push to the remote repo*



...or push an existing repository from the command line

```
git remote add origin https://github.com/piracode/branch-practice.git
git branch -M main
git push -u origin main
```

Sets the new remote repository URL you will push to and pull from.

Renames the current branch to main, forcefully if necessary.

Pushes your main branch to the remote repository (origin) and sets it to track upstream changes.

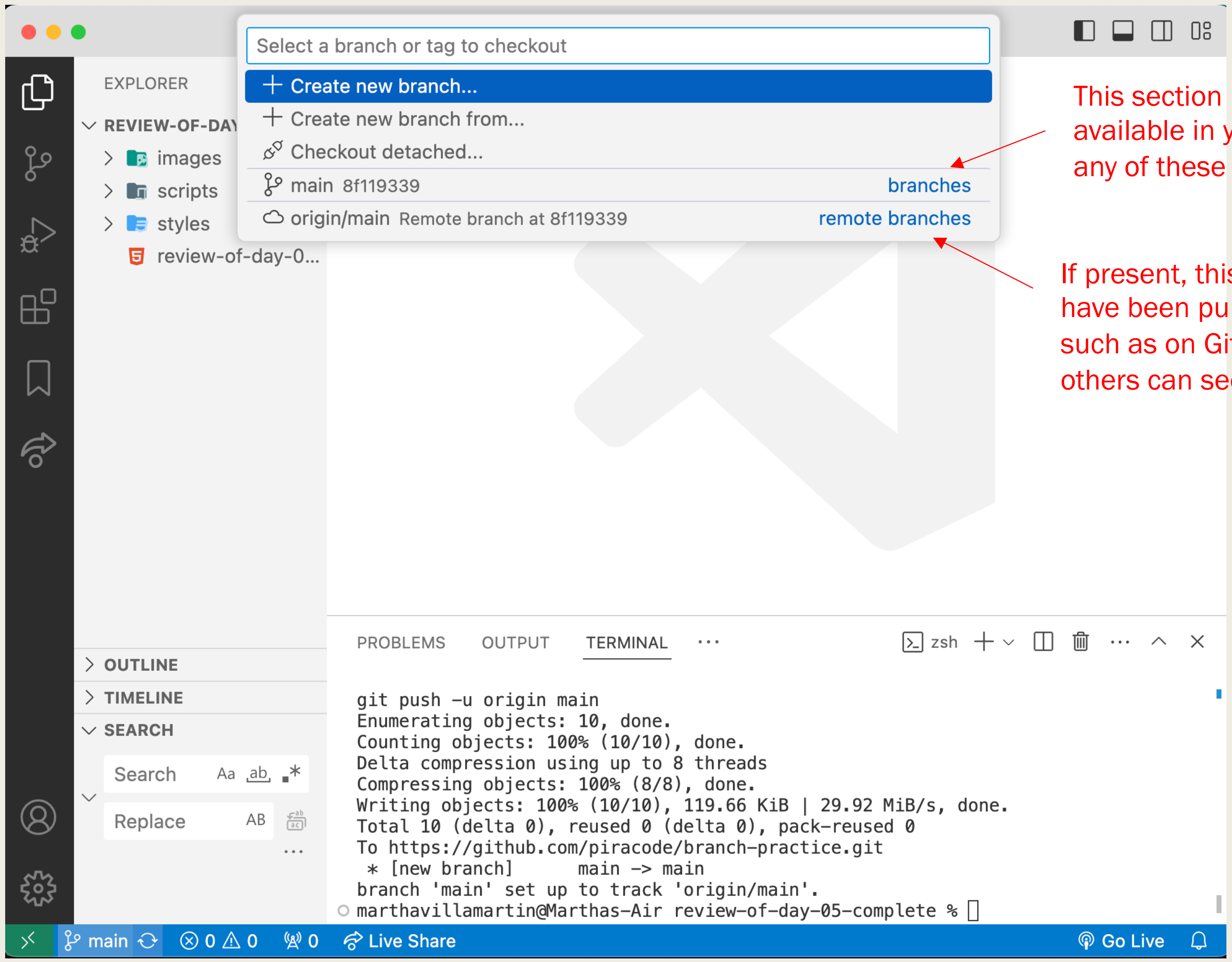
Exercise #3

- Create a branch in review-of-day-05-complete local repo called “my-branch”

Note: In real projects, use descriptive branch names, like feature/login or bugfix/header

First, let's observe the code editor

This dropdown appears when you click on the branch name. It allows you to select, create, and manage your branches directly from the VSCode interface.



```
git push -u origin main
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 8 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (10/10), 119.66 KiB | 29.92 MiB/s, done.
Total 10 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/piracode/branch-practice.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
marthavillamartin@Marthas-Air review-of-day-05-complete %
```

This section lists all branches that are currently available in your local repository. You can switch to any of these branches by clicking on them.

If present, this section would list the branches that have been published to the remote repository, such as on GitHub. These are the branches that others can see and interact with.

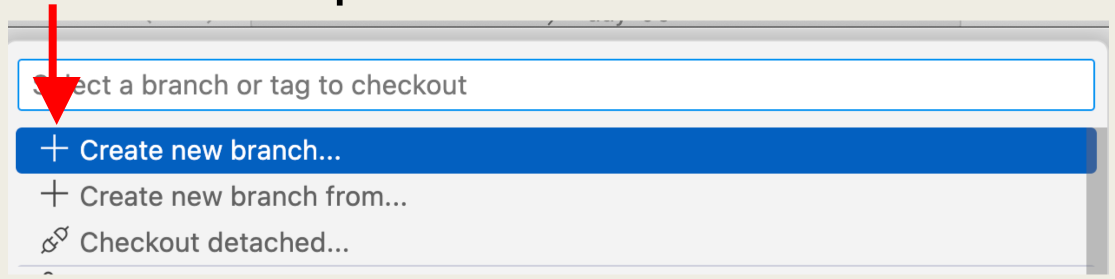
This displays the currently checked-out branch.
Click here to switch branches or create a new one.

Create a Branch

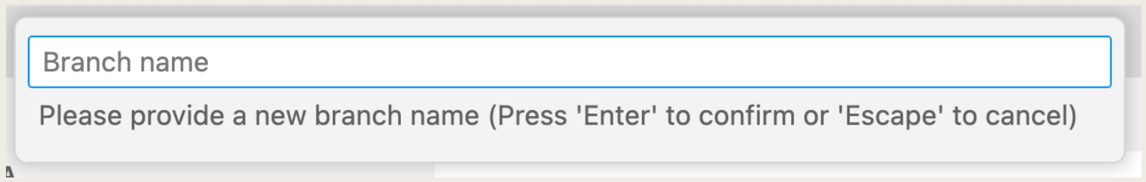
VSCode View

1) Click on the current branch at the bottom left of the app

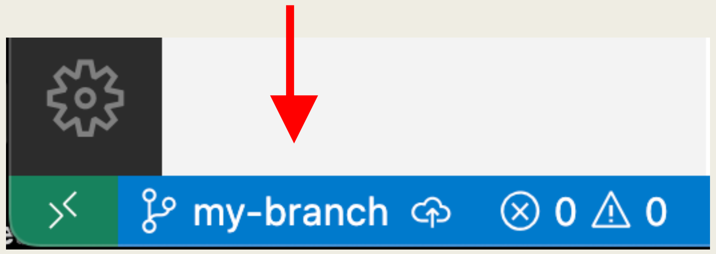
2) Click on “create new branch” in the dropdown menu.


Select a branch or tag to checkout
+ Create new branch...
+ Create new branch from...
Checkout detached...

3) Write the branch name and press enter


Branch name
Please provide a new branch name (Press 'Enter' to confirm or 'Escape' to cancel)

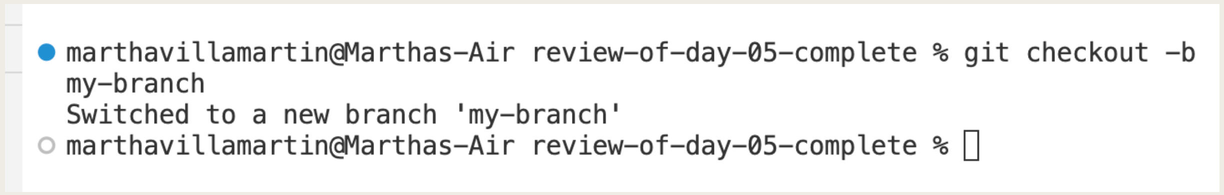
Notice how it switched to the newly created branch



Terminal

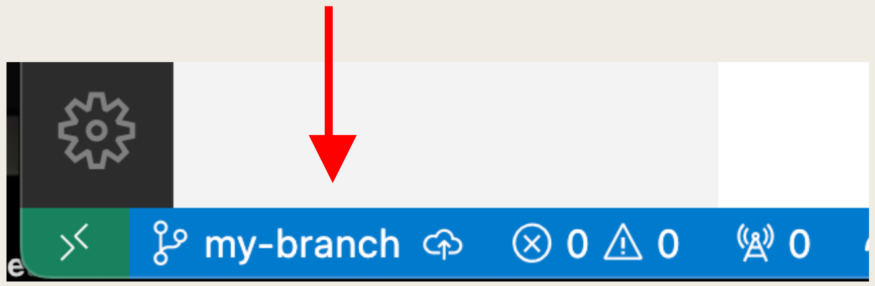
git checkout –b my-branch

This will create the new branch and immediately switch to it.



```
● marthavillamartin@Marthas-Air review-of-day-05-complete % git checkout -b
  my-branch
  Switched to a new branch 'my-branch'
○ marthavillamartin@Marthas-Air review-of-day-05-complete % 
```

Notice how it switched to the newly created branch



What happens when I create a branch?

When you create a new branch, you duplicate the current branch.

For instance, if you're on the 'main' branch and create a new branch called “my-branch”, you essentially duplicate “main” at that moment.

Both branches are initially identical, but they will diverge as changes are made on “my-branch”.

Exercise #4

- Make changes on the ”my-branch".
- Open "review-of-day-05-complete" and modify the background color of the body element.
- Stage and commit the changes on the ”my-branch".
- Switch to the "main" branch to observe the differences between branches.
- View the branches to see the current branch and any newly created branches.

Step 1 - Update styles.css

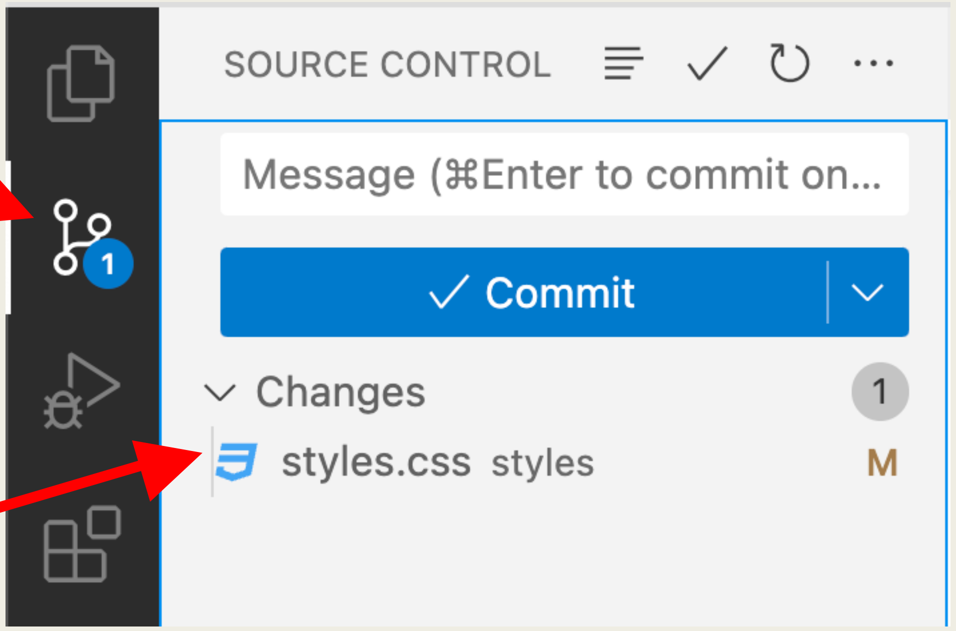
- Modify the background color of the body tag in styles.css to a different light color and save the change.

<u>**Click on the Source Control tab:**</u>

The 'Changes' tab in Source Control highlights modified files, indicating they're not yet committed.

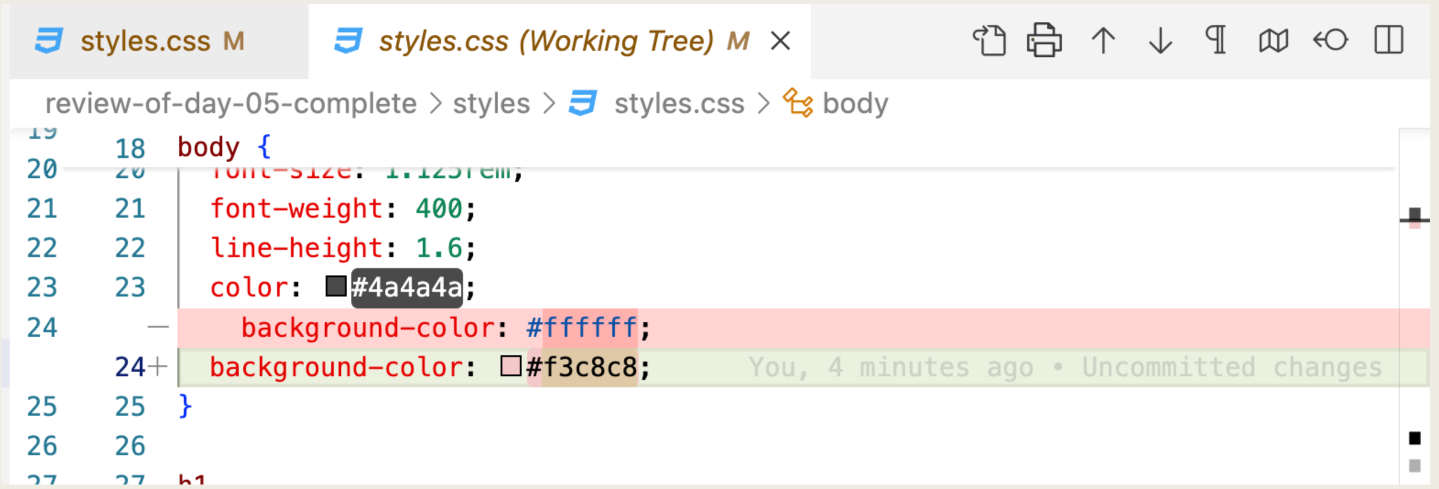
<u>**Click on the modified file:**</u>

This action will open another instance of the styles.css file, reflecting the changes made, known as the working tree view.



<u>**Observe the changes:**</u>

VSCode visually highlights changes, showing additions in green and deletions in red, with annotations like "+" for additions and "-" for deletions next to modified lines.


styles.css M
styles.css (Working Tree) M
review-of-day-05-complete > styles > styles.css > body

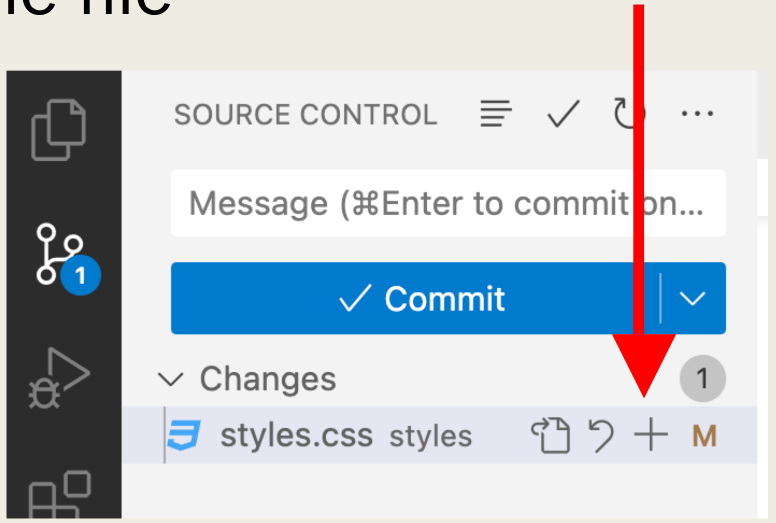
```
body {
  font-size: 1.125rem;
  font-weight: 400;
  line-height: 1.6;
  color: #4a4a4a;
-   background-color: #ffffff;
+ background-color: #f3c8c8;      You, 4 minutes ago • Uncommitted changes
}

h1
```

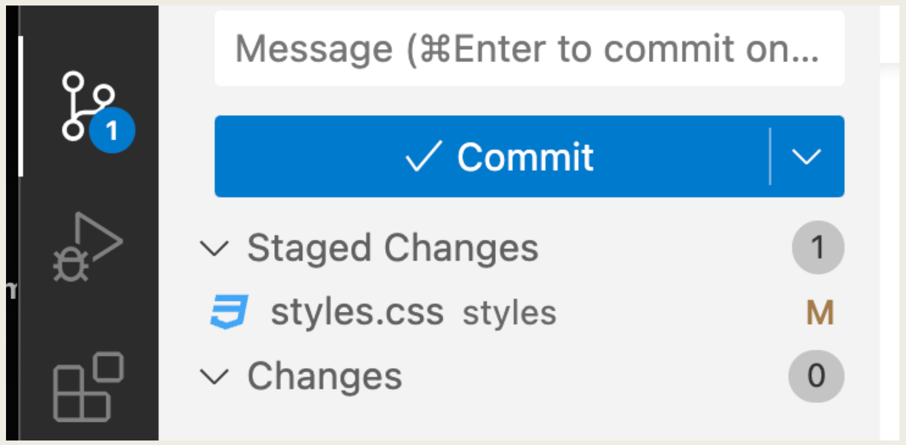
Step 2a - Stage changes

VSCode View	Terminal

Click on the + button to stage the file

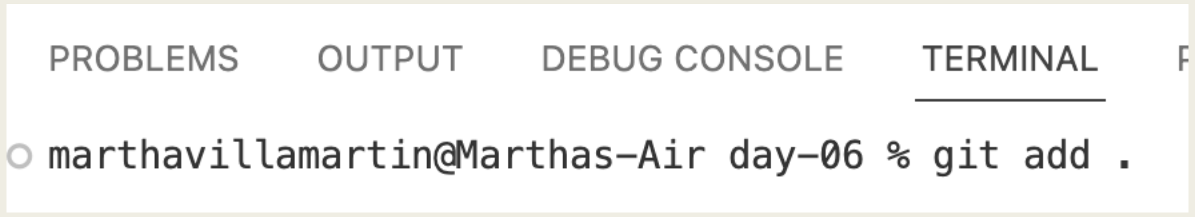


Now the file went from "Changes" to "Staged Changes"



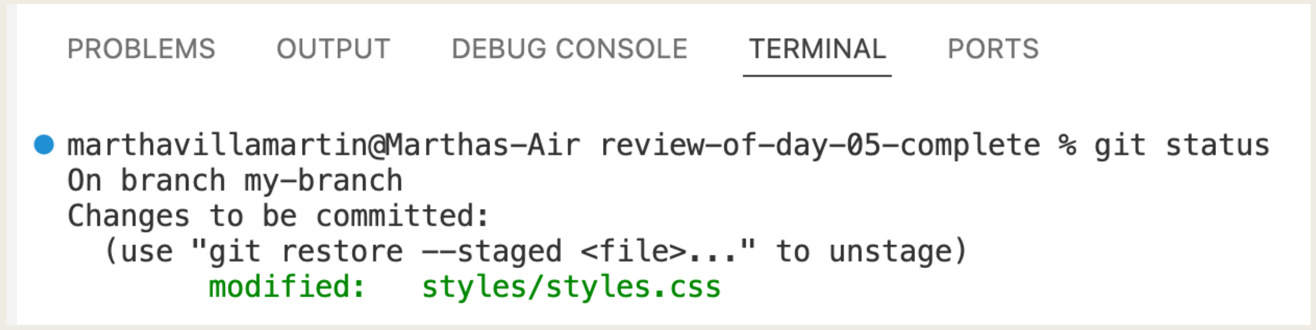
Type the following command to stage all changes

git add .



```
marthavillamartin@Marthas-Air day-06 % git add .
```

(Optional) -- Run "git status" to confirm staged changes.

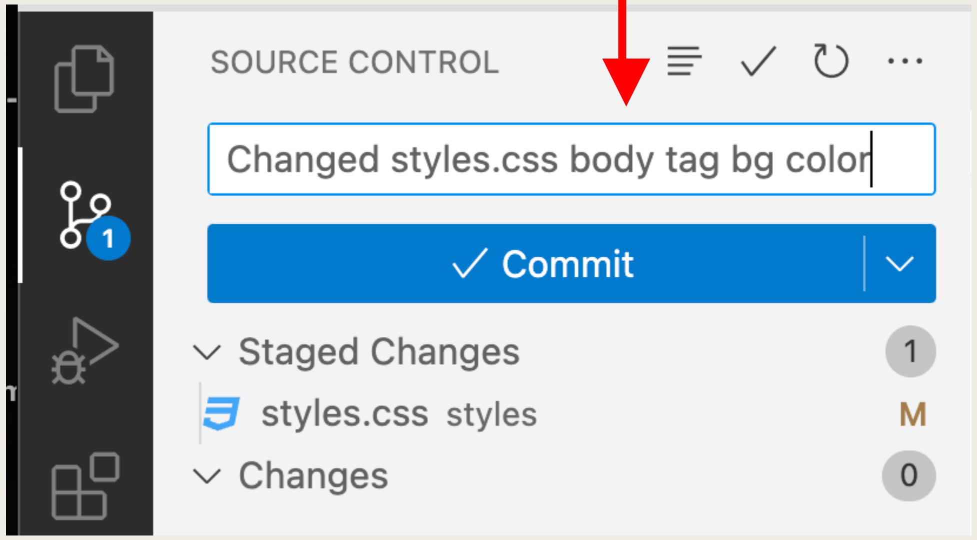


```
marthavillamartin@Marthas-Air review-of-day-05-complete % git status
On branch my-branch
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   styles/styles.css
```

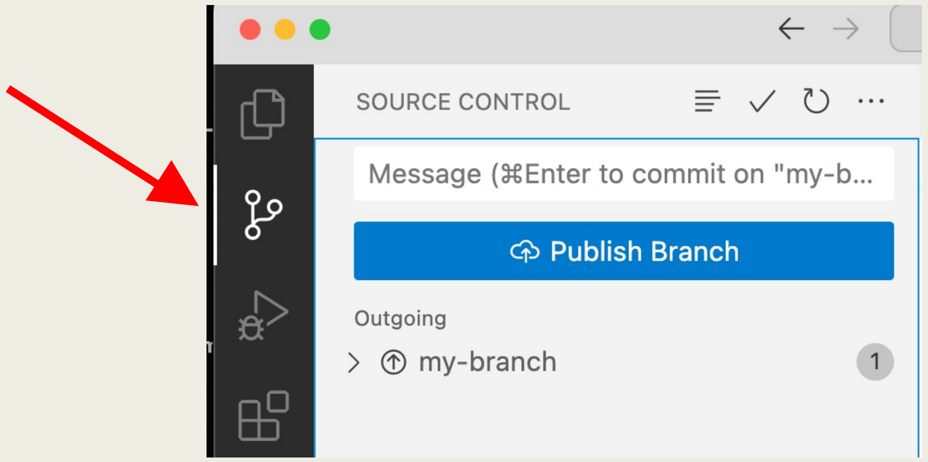
Step 2b - Commit changes

VSCode View	Terminal

Write the commit message and click on "Commit"

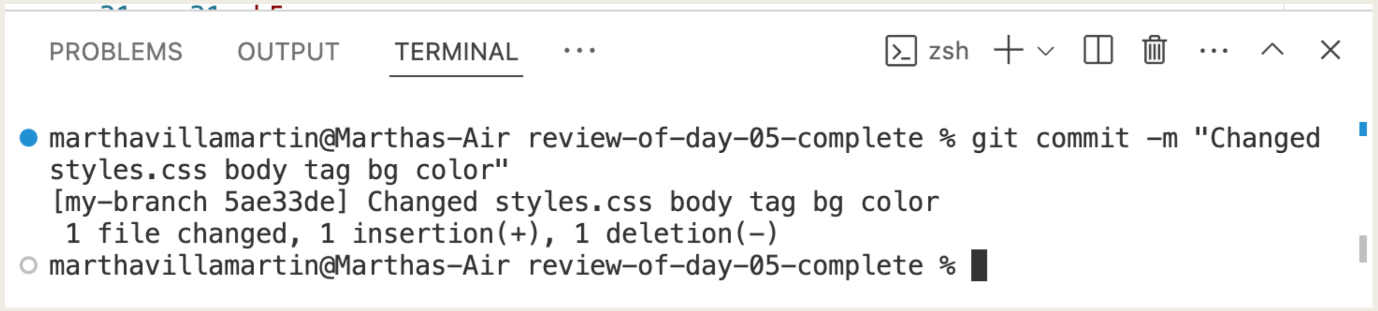


Now, the source control tab is empty



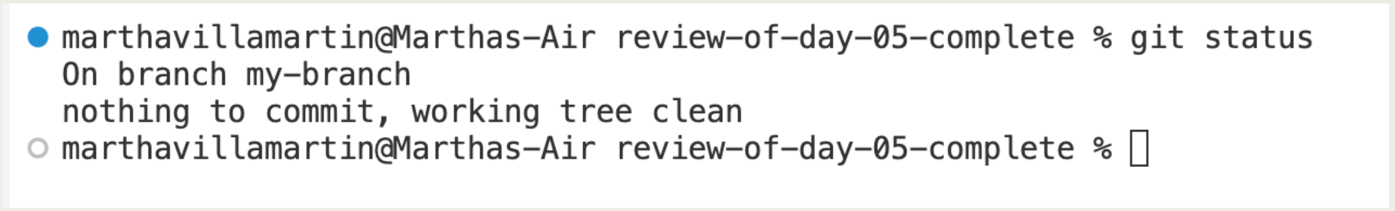
Type the following command to commit:

git commit –m "Changed styles.css body tag bg color"



```
marthavillamartin@Marthas-Air review-of-day-05-complete % git commit -m "Changed
styles.css body tag bg color"
[my-branch 5ae33de] Changed styles.css body tag bg color
 1 file changed, 1 insertion(+), 1 deletion(-)
marthavillamartin@Marthas-Air review-of-day-05-complete %
```

(Optional) -- Run "git status" to verify your commit.

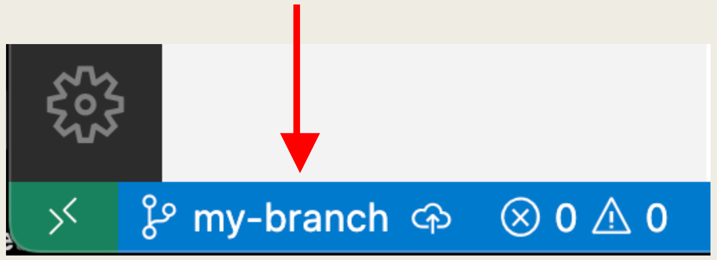
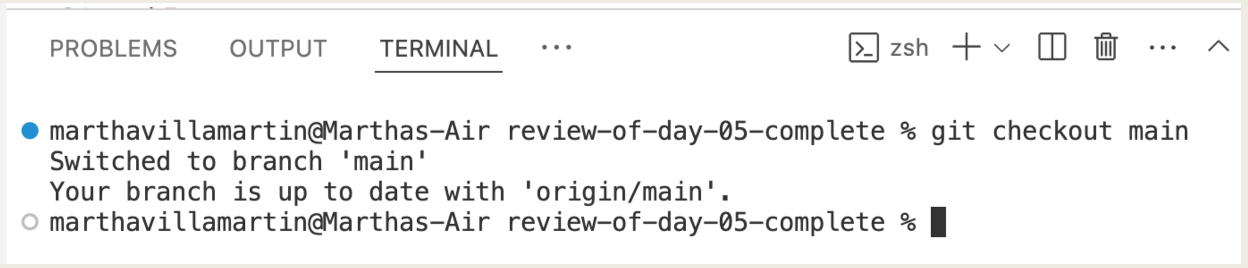
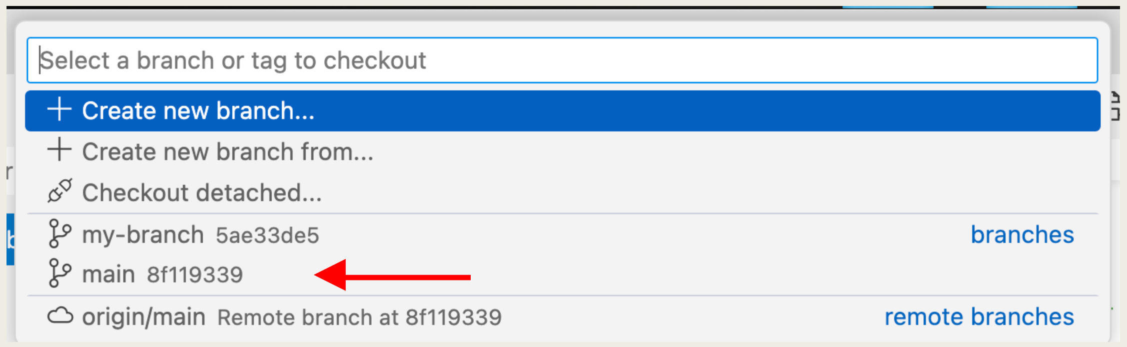


```
marthavillamartin@Marthas-Air review-of-day-05-complete % git status
On branch my-branch
nothing to commit, working tree clean
marthavillamartin@Marthas-Air review-of-day-05-complete %
```

Step 3 - Switch to the main branch

Now that you have added and committed the changes, you can switch between branches.

Note: Committing changes is required to switch branches. You can also use 'git stash' to temporarily save changes if you prefer not to commit them, but we won't cover this here.

VSCode View	Terminal
1) Click on the current branch at the bottom left of the app	git checkout main
	This will switch to the branch called "main"
2) Click on the branch you want to switch to, in our case "main"	
	

```
marthavillamartin@Marthas-Air review-of-day-05-complete % git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
marthavillamartin@Marthas-Air review-of-day-05-complete %
```

```
Select a branch or tag to checkout
+ Create new branch...
+ Create new branch from...
Checkout detached...
my-branch 5ae33de5                              branches
main 8f119339
origin/main Remote branch at 8f119339    remote branches
```

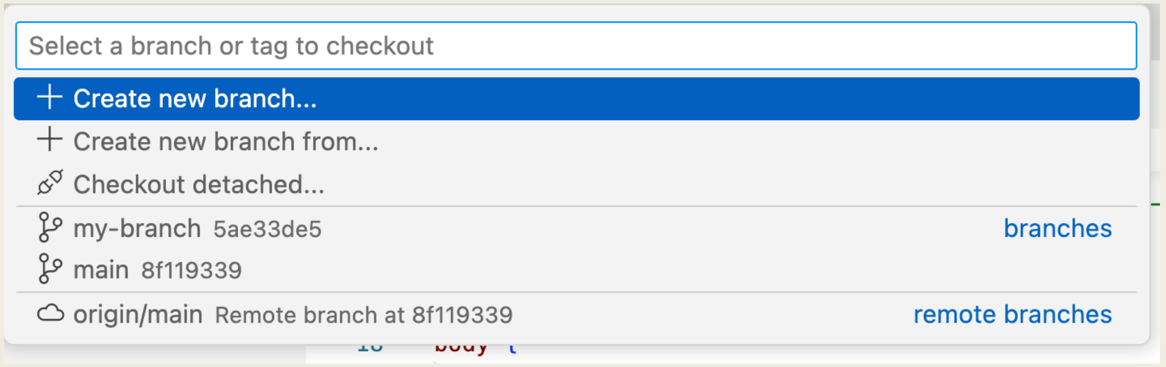
Note: The changes made in the body tag apply only to the "practice-branch" and not the "main" branch.

Step 4 (Optional) - View branches

VSCode View

1) Click on the current branch at the bottom left of the app

2) See all the remote & local branches in the dropdown menu.

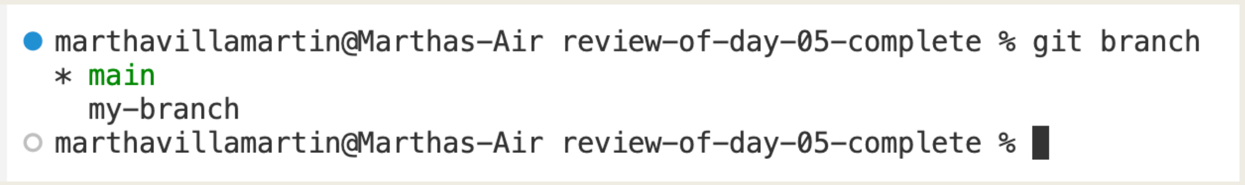


```
Select a branch or tag to checkout
+ Create new branch...
+ Create new branch from...
Checkout detached...
my-branch 5ae33de5                                  branches
main 8f119339
origin/main Remote branch at 8f119339        remote branches
```

Terminal

git branch

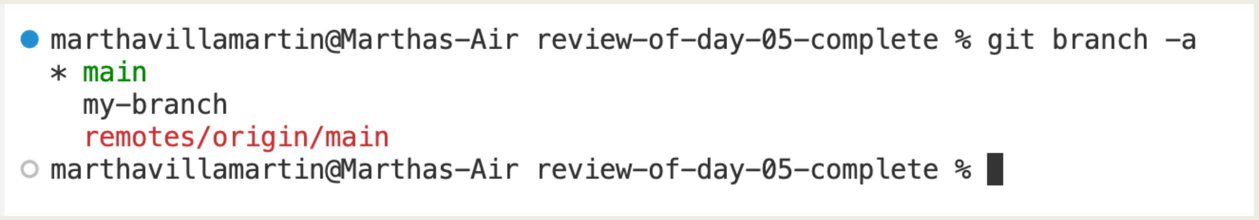
This command lists all branches in the repository. The current branch is highlighted & marked with an asterisk.



```
● marthavillamartin@Marthas-Air review-of-day-05-complete % git branch
  * main
    my-branch
○ marthavillamartin@Marthas-Air review-of-day-05-complete % 
```

git branch -a

This command provides a complete overview of all branches in your repository, including both local and remote branches.



```
● marthavillamartin@Marthas-Air review-of-day-05-complete % git branch -a
  * main
    my-branch
    remotes/origin/main
○ marthavillamartin@Marthas-Air review-of-day-05-complete % 
```

<u>Note</u>: Local branches are for individual work on your machine; remote branches track shared changes on a server like GitHub.

Git Workflow: Solo vs Team

Working Alone

- Create and work on branches locally.
- Merge changes into your local “main” branch.
- Push “main” to the remote repository when ready.

Working in a Team

- Create a feature branch locally for new work.
- Push feature branches to the remote repository.
- Use pull requests for review and merge into “remote main”.
- Regularly pull updates from “remote main” to “local main”.

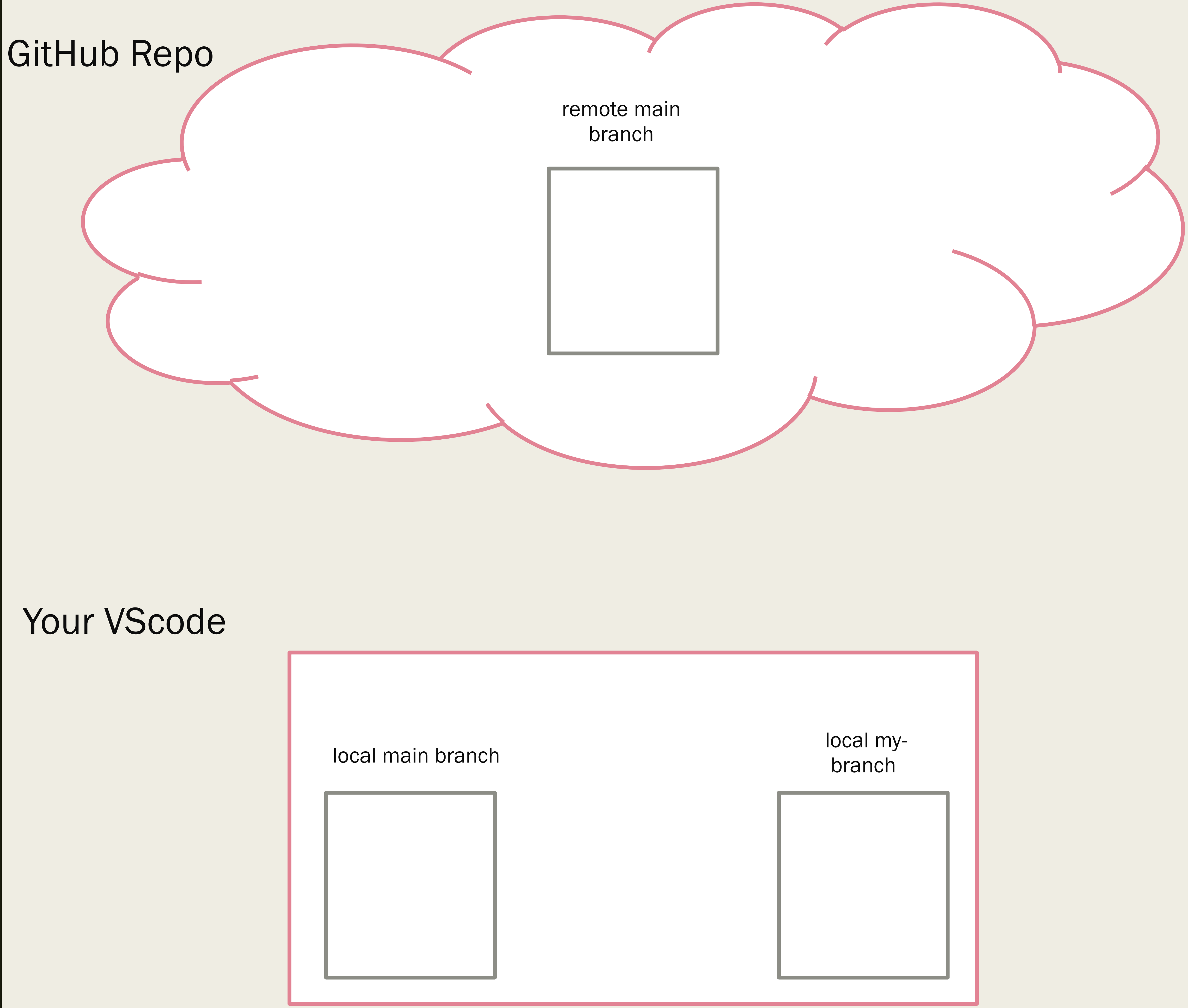
Solo Workflow

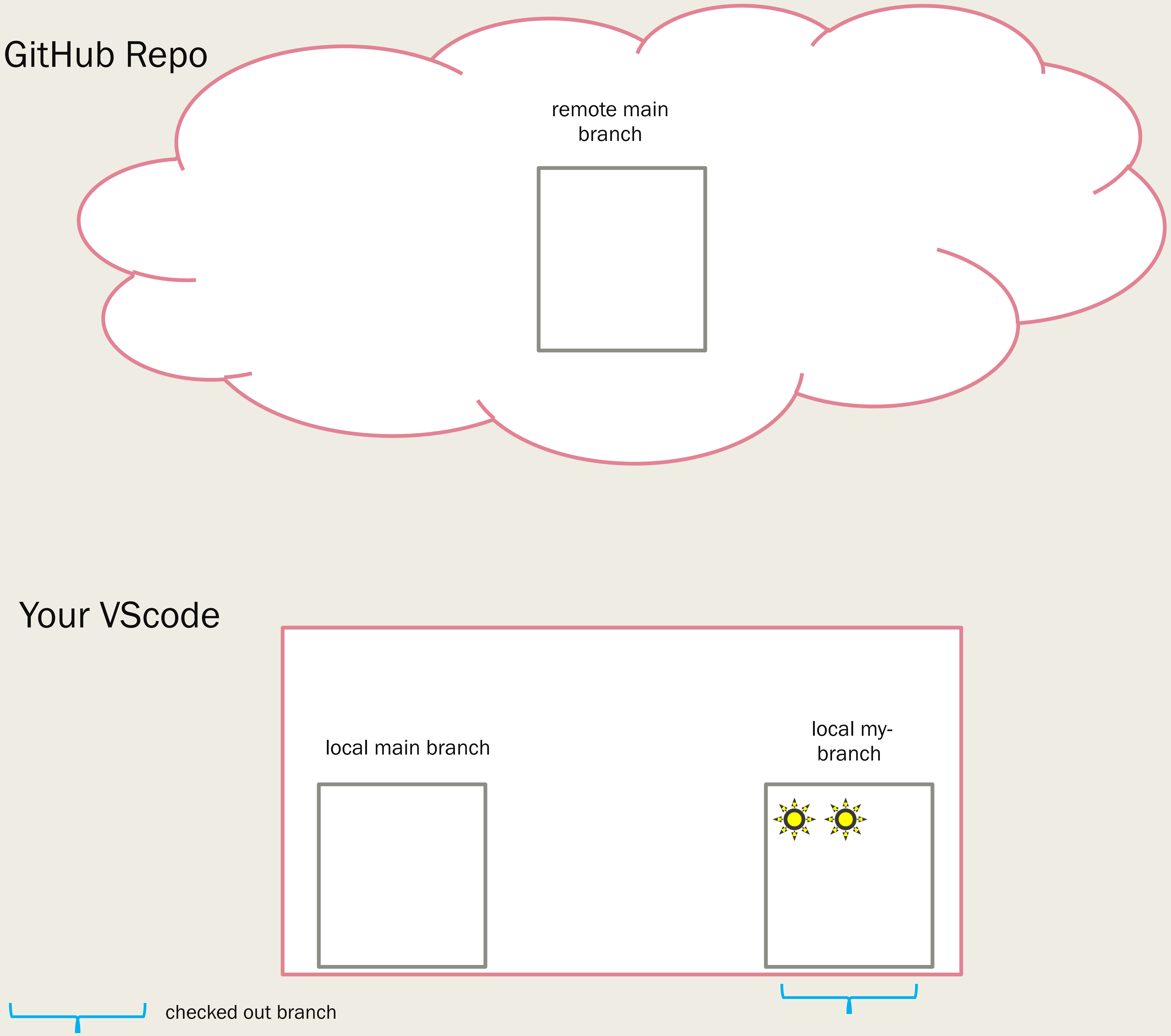
Workflow Overview in Visuals

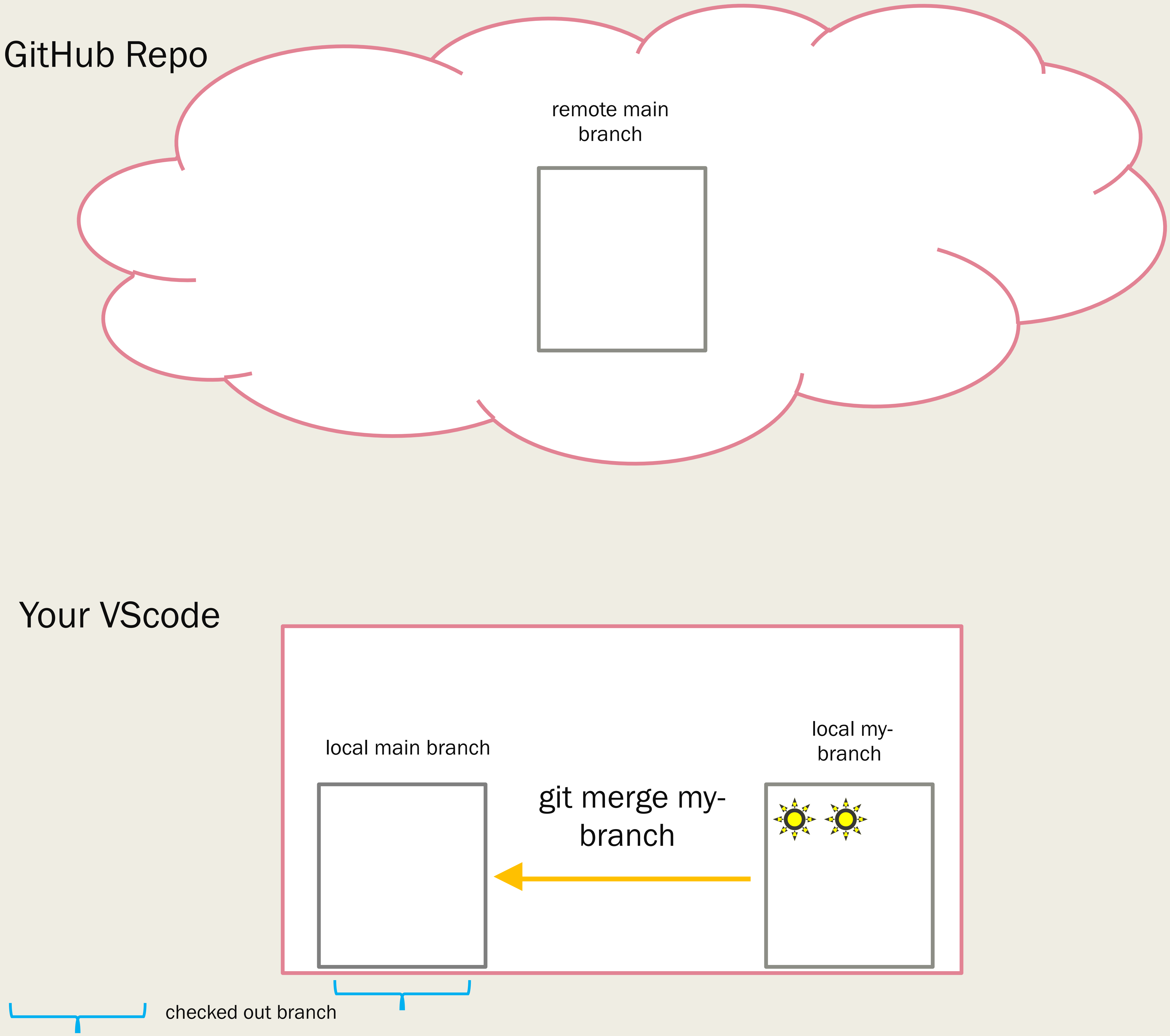
Note:

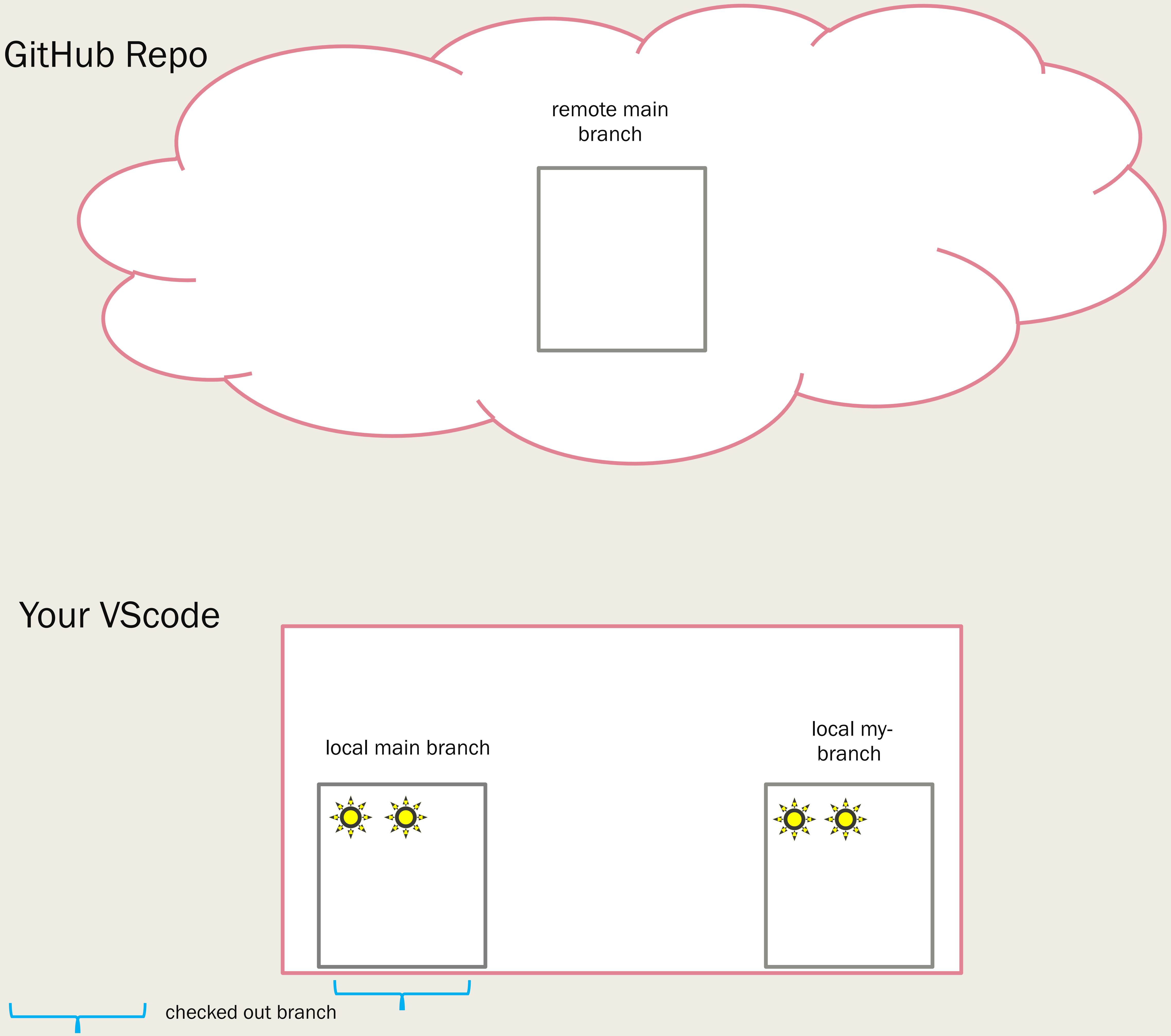
This diagram provides a high-level view of the workflow for solo work. It begins with the assumption that you already have a remote repository, and you already created a local branch to start your work.

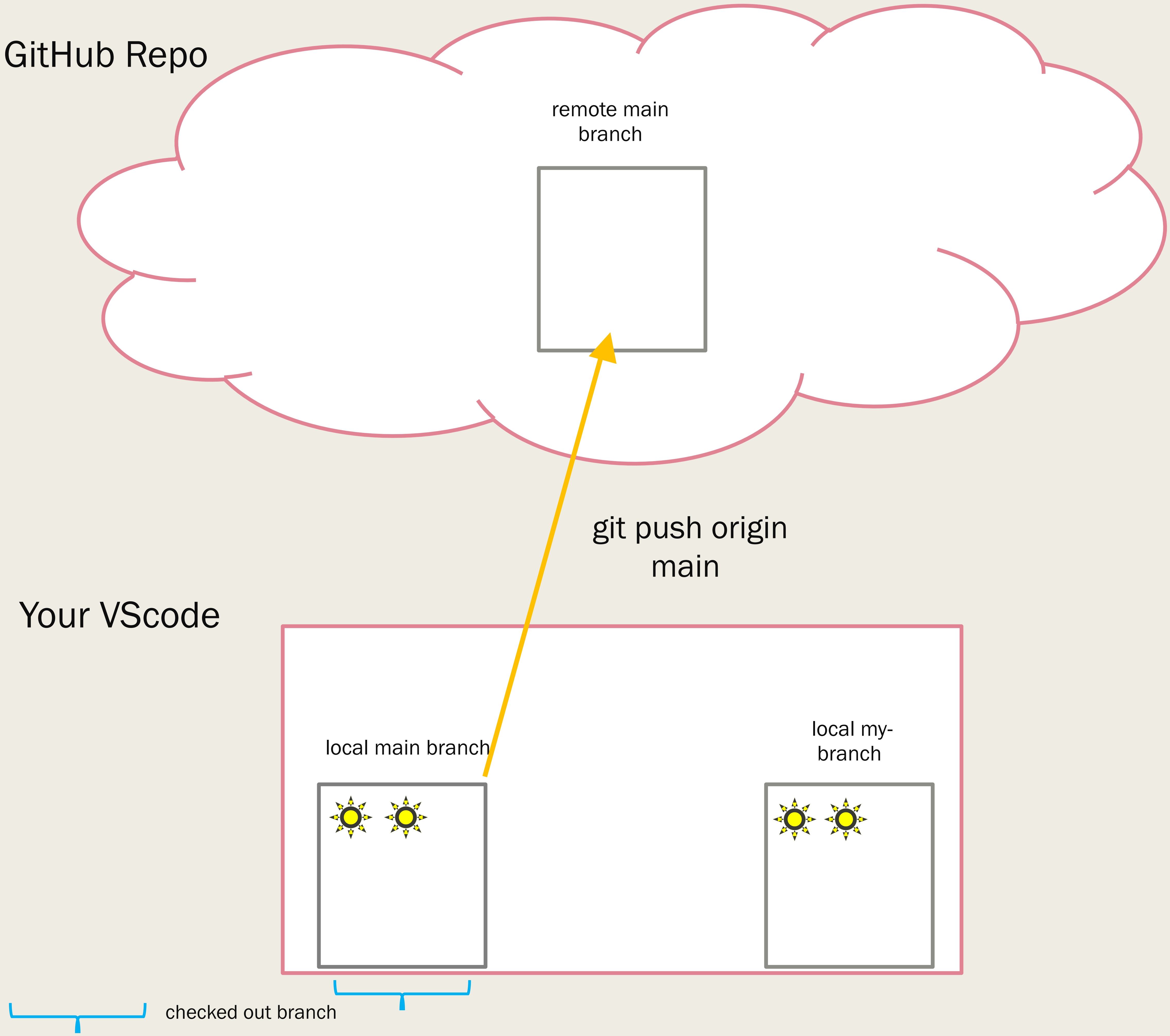
Don’t adopt this workflow when working in a team!

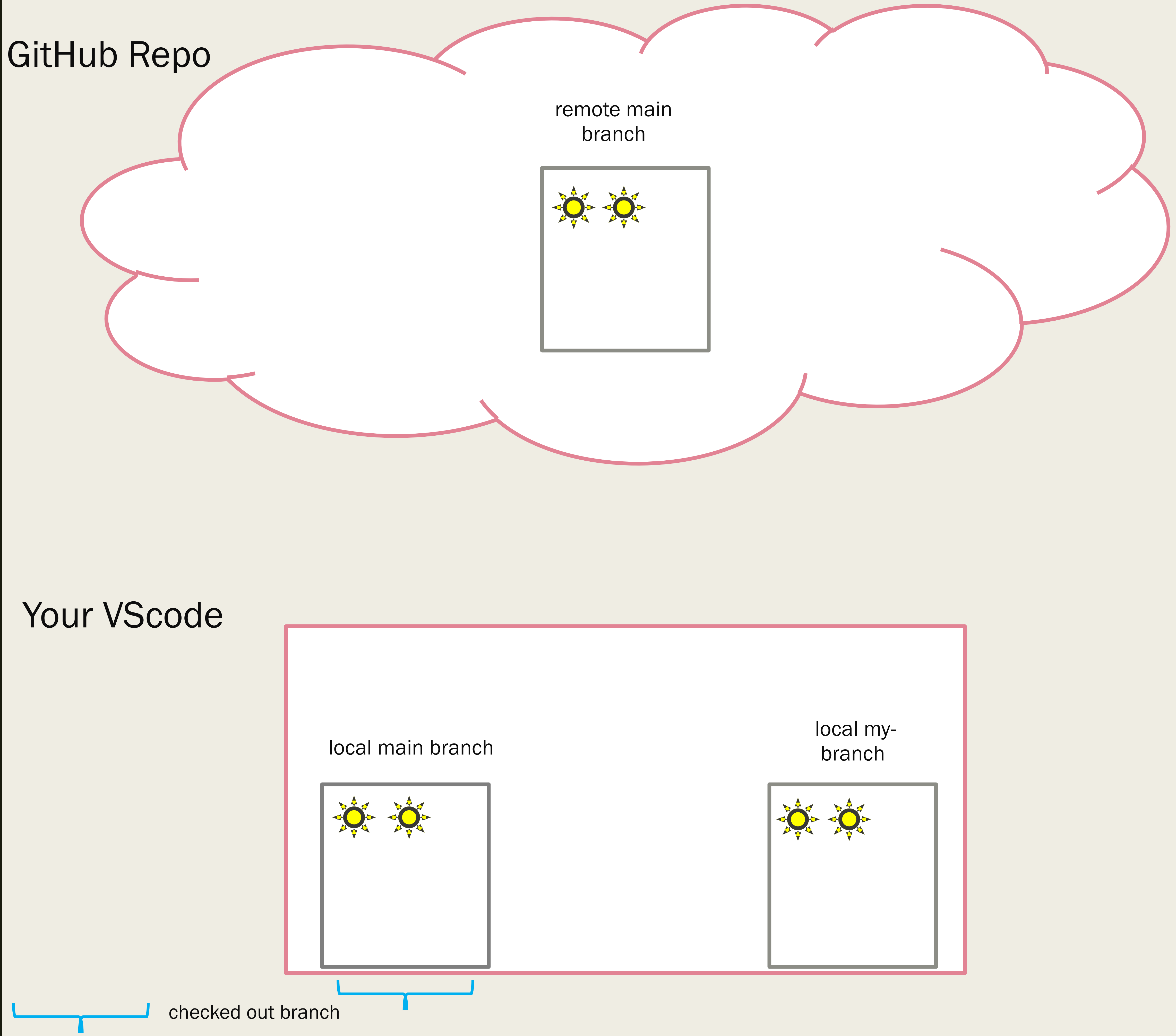

GitHub Repo
remote main branch
Your VScode
local main branch
local my-branch

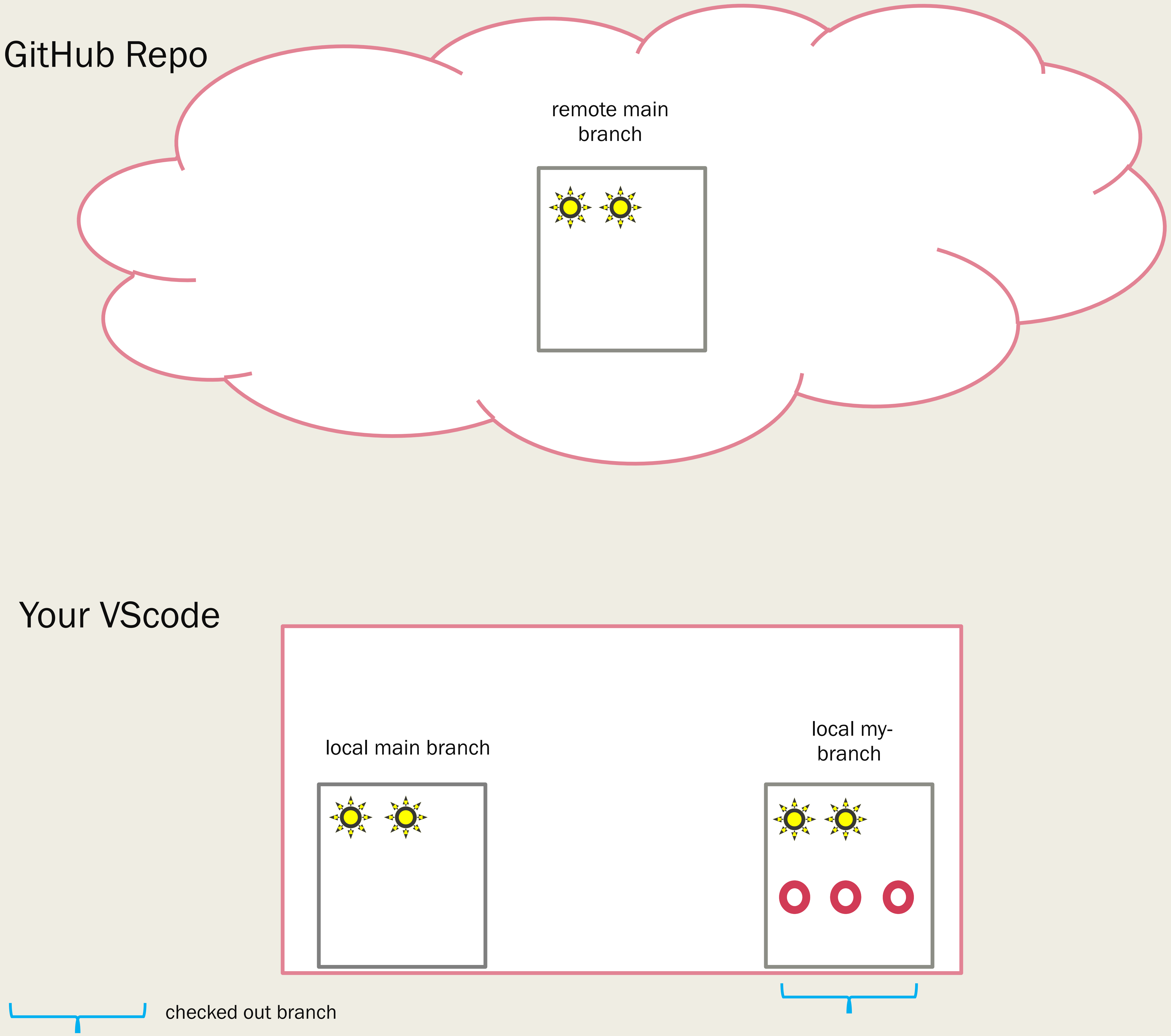

GitHub Repo
remote main branch
Your VScode
local main branch
local my-branch
checked out branch

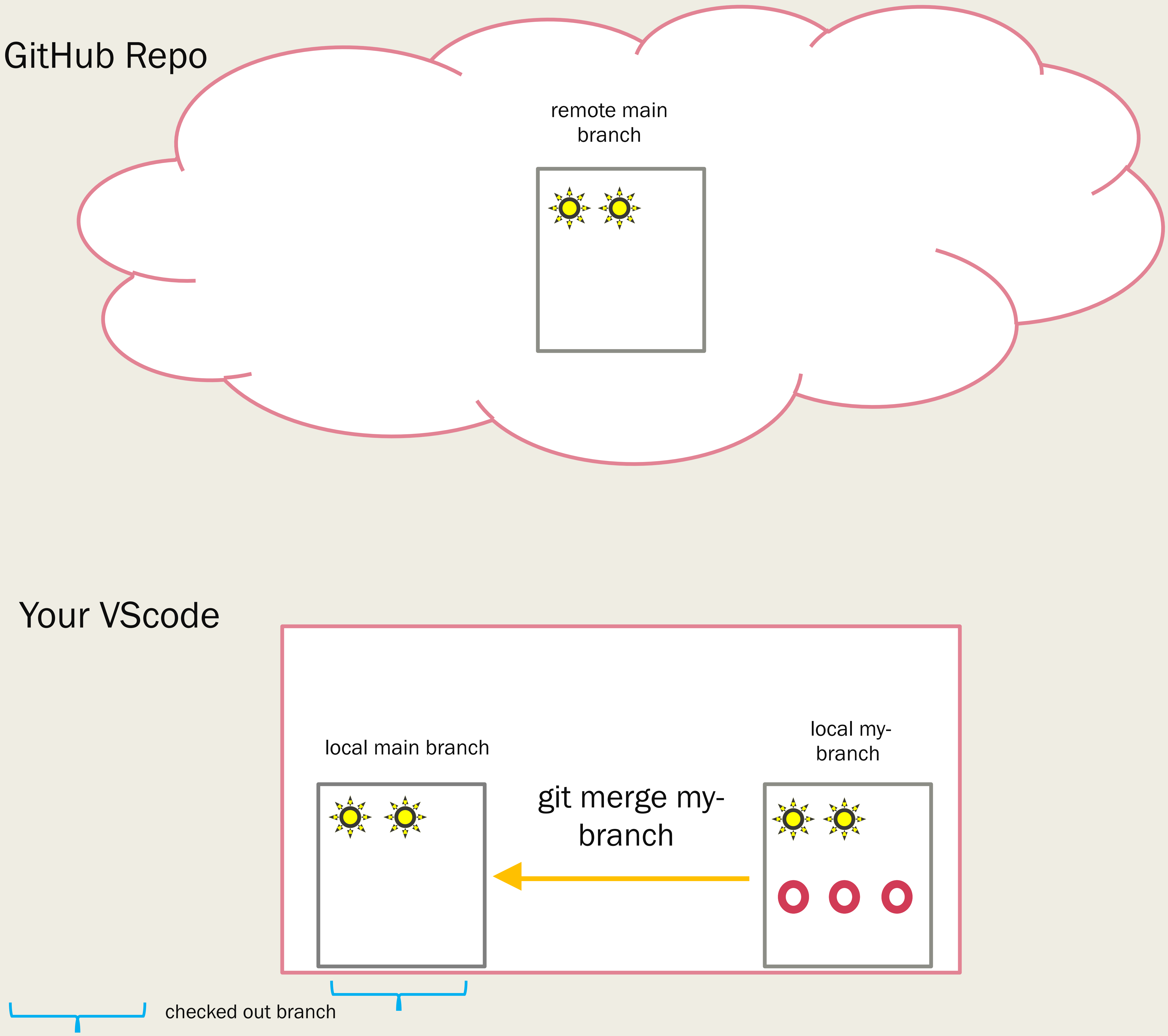

GitHub Repo
remote main branch
Your VScode
local main branch
local my-branch
git merge my-branch
checked out branch


GitHub Repo
remote main branch
Your VScode
local main branch
local my-branch
checked out branch


GitHub Repo
remote main branch
git push origin main
Your VScode
local main branch
local my-branch
checked out branch

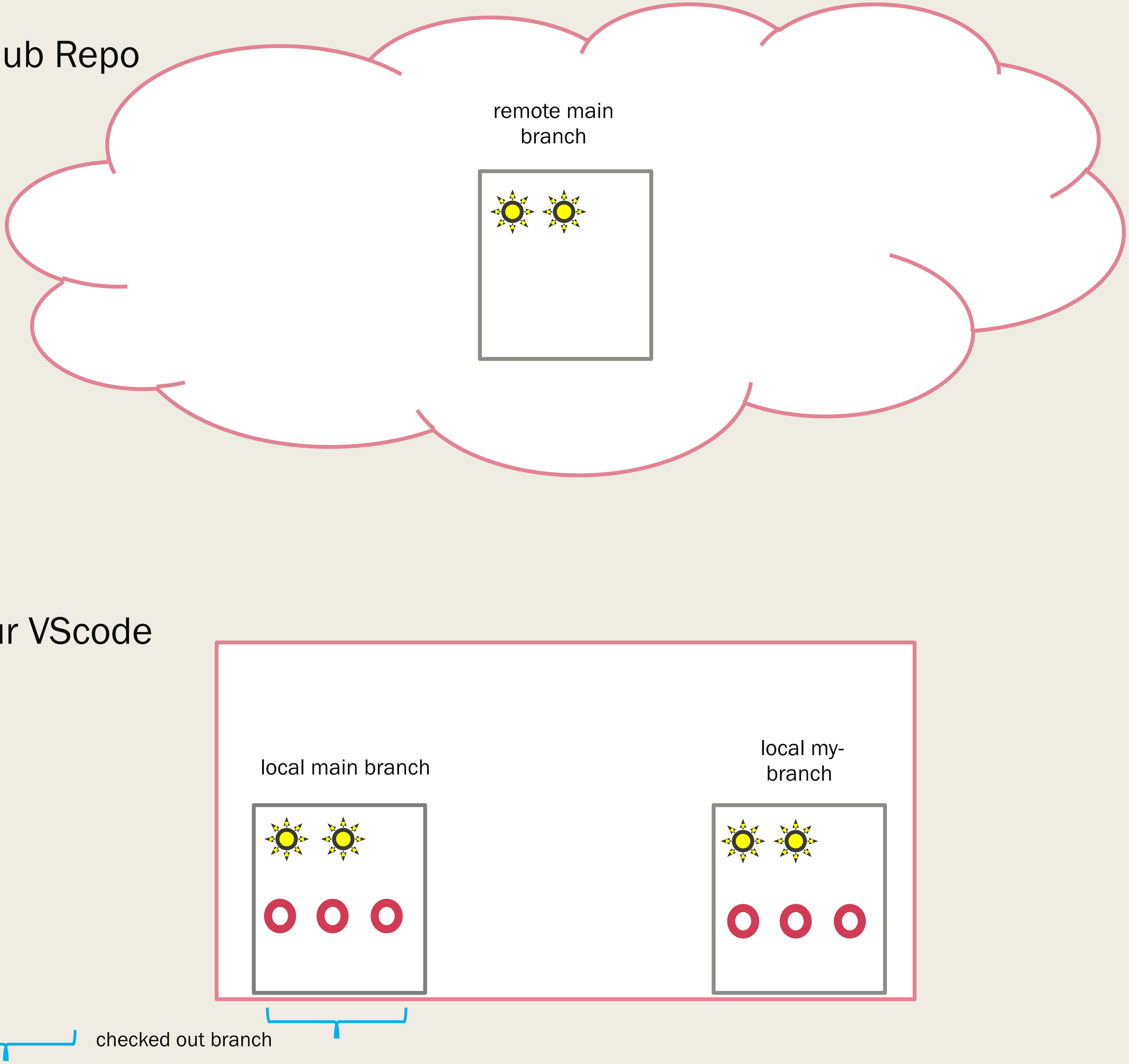

GitHub Repo
remote main branch
Your VScode
local main branch
local my-branch
checked out branch

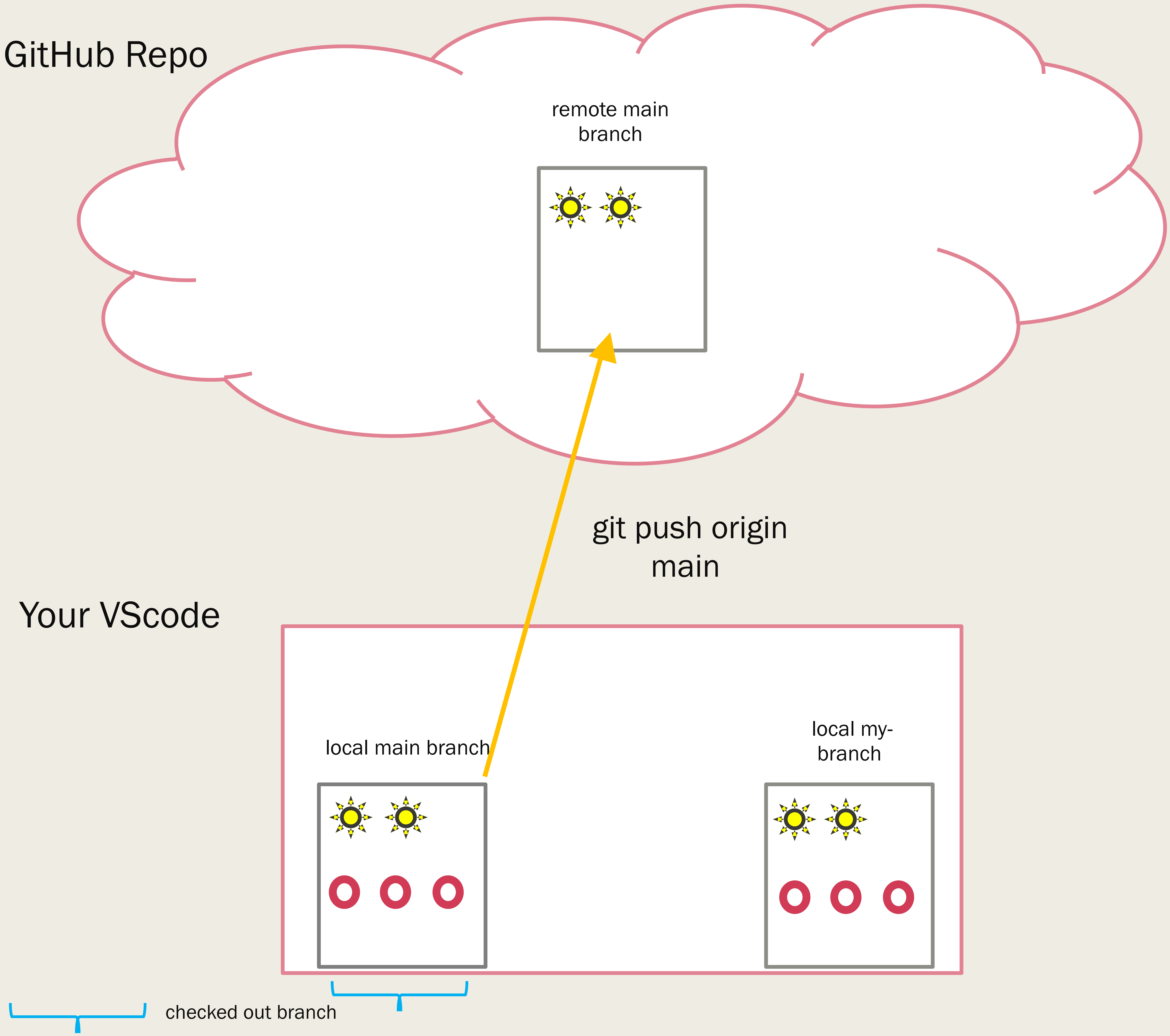

GitHub Repo
remote main branch
Your VScode
local main branch
local my-branch
checked out branch

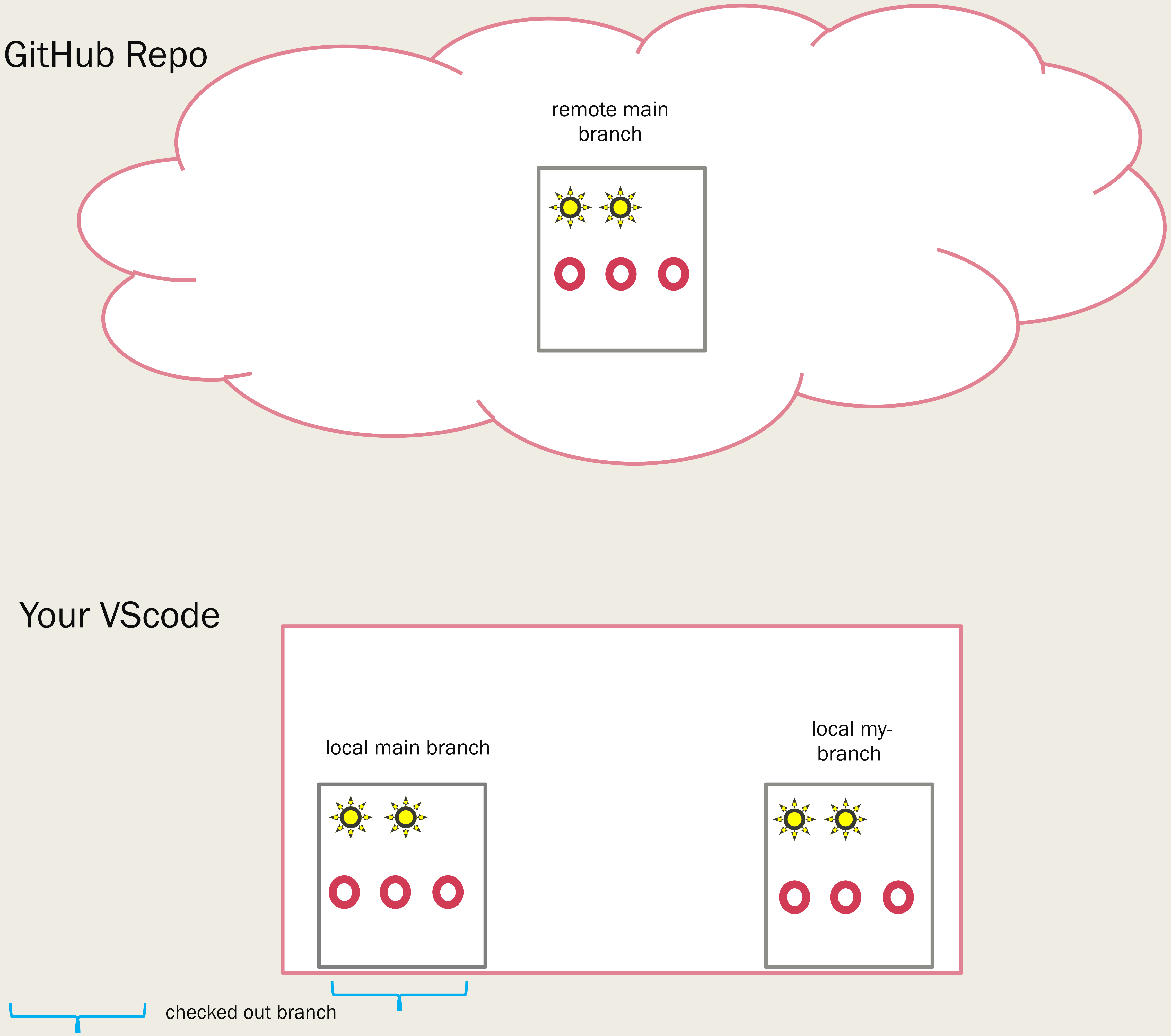

GitHub Repo
remote main branch
Your VScode
local main branch
git merge my-branch
local my-branch
checked out branch

GitHub Repo

Your VScode




GitHub Repo
remote main branch
git push origin main
Your VScode
local main branch
local my-branch
checked out branch


GitHub Repo
remote main branch
Your VScode
local main branch
local my-branch
checked out branch

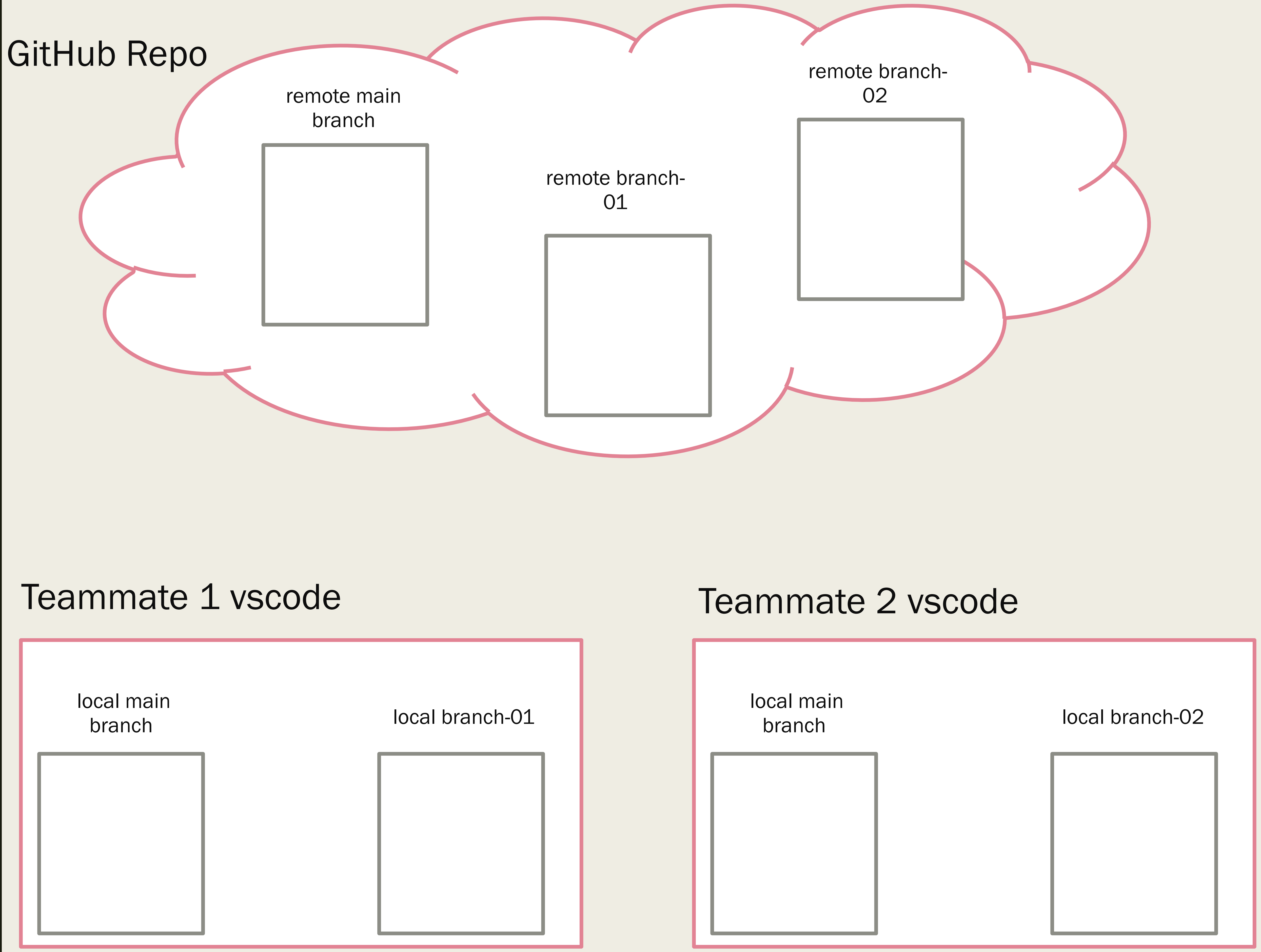
Team Workflow

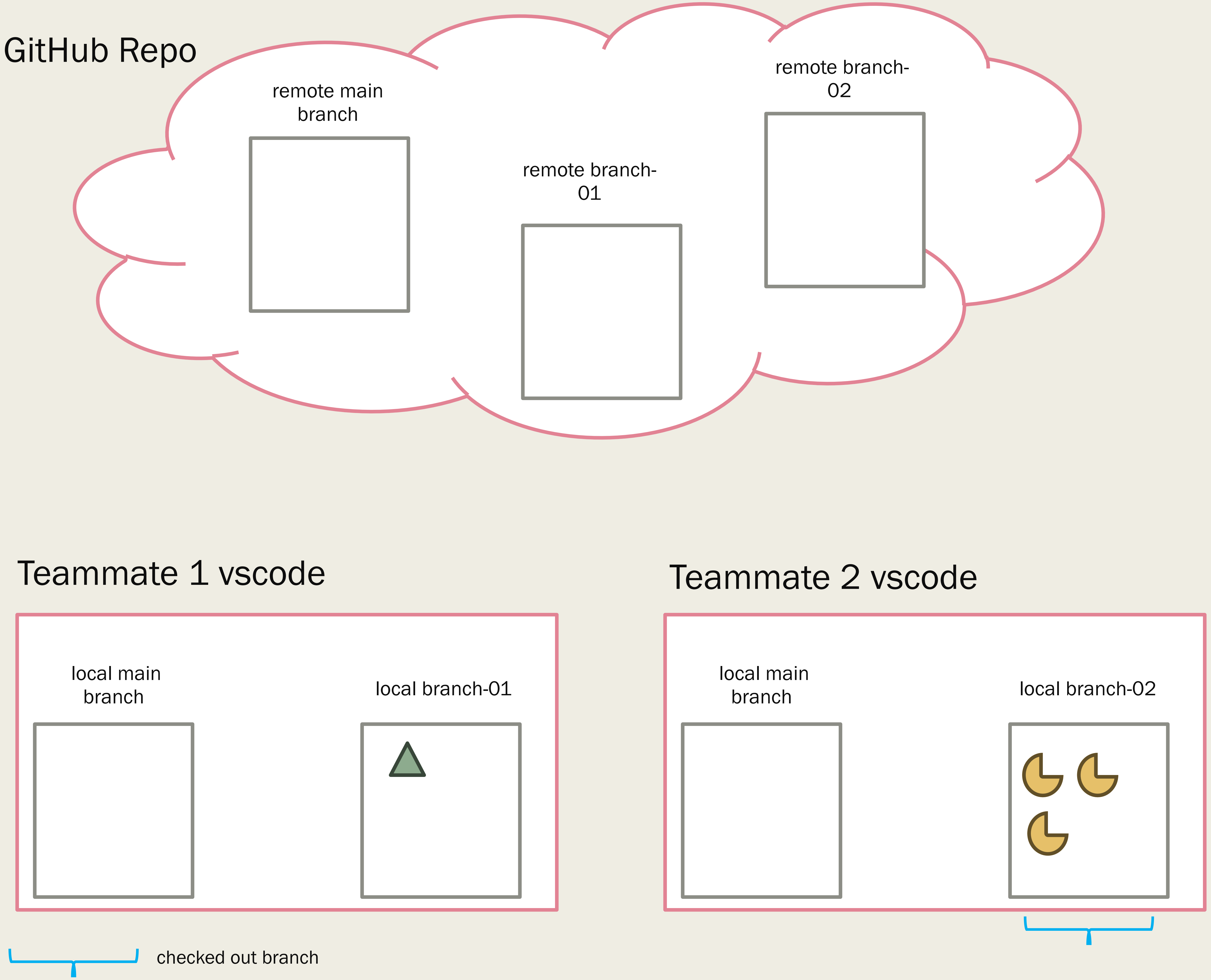
Workflow Overview in Visuals

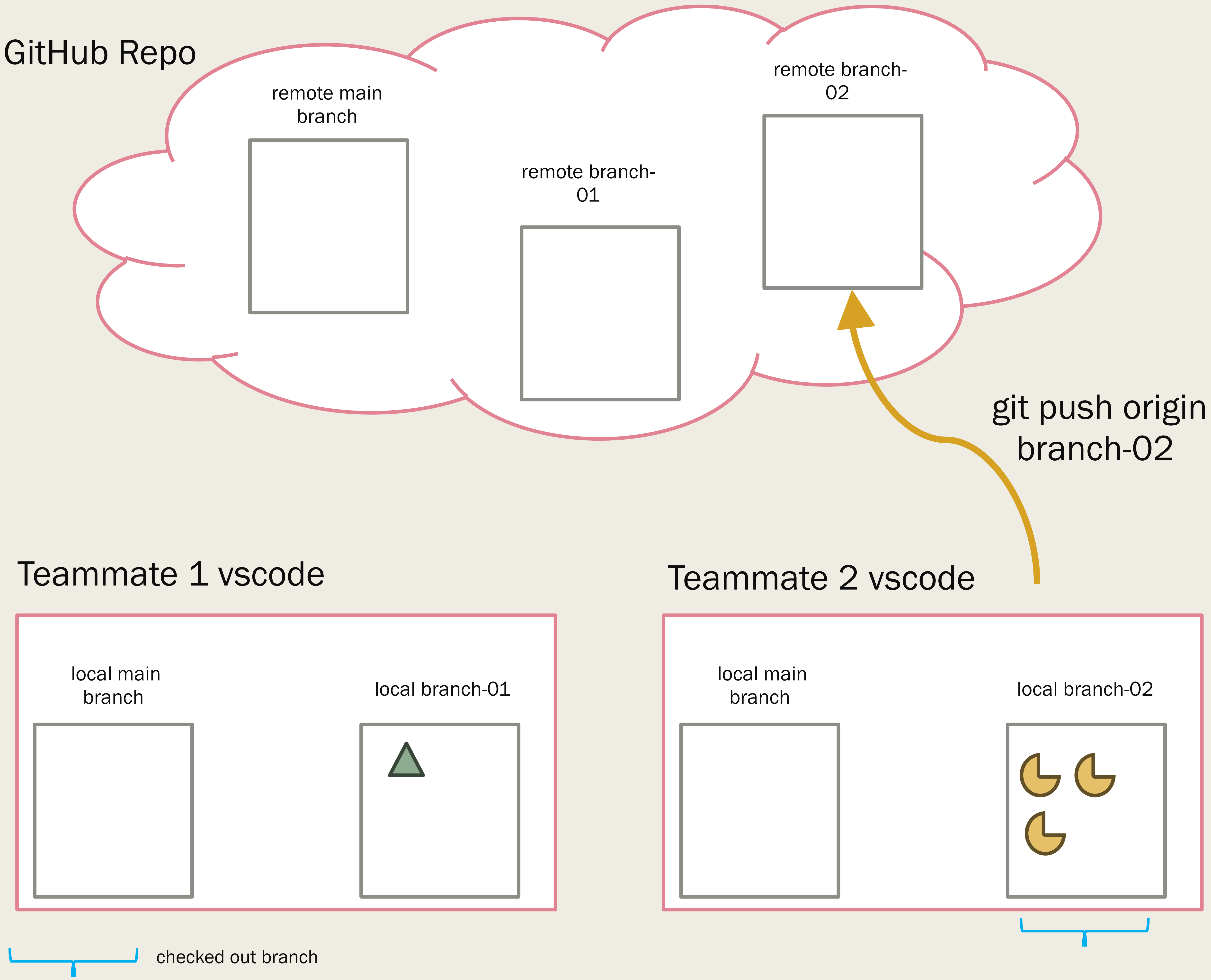
Note:

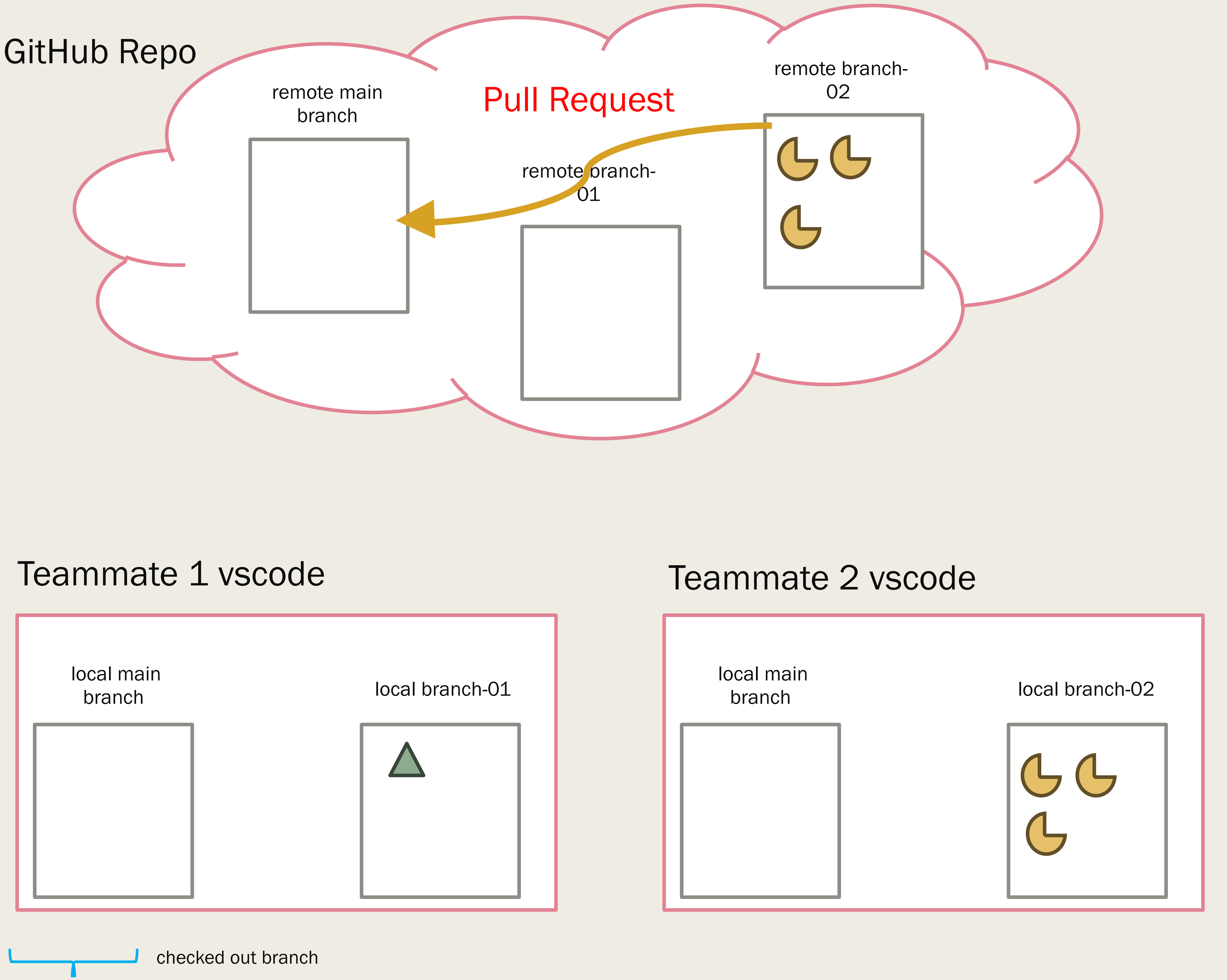
This diagram gives a high-level view of the collaborative workflow with two teammates.

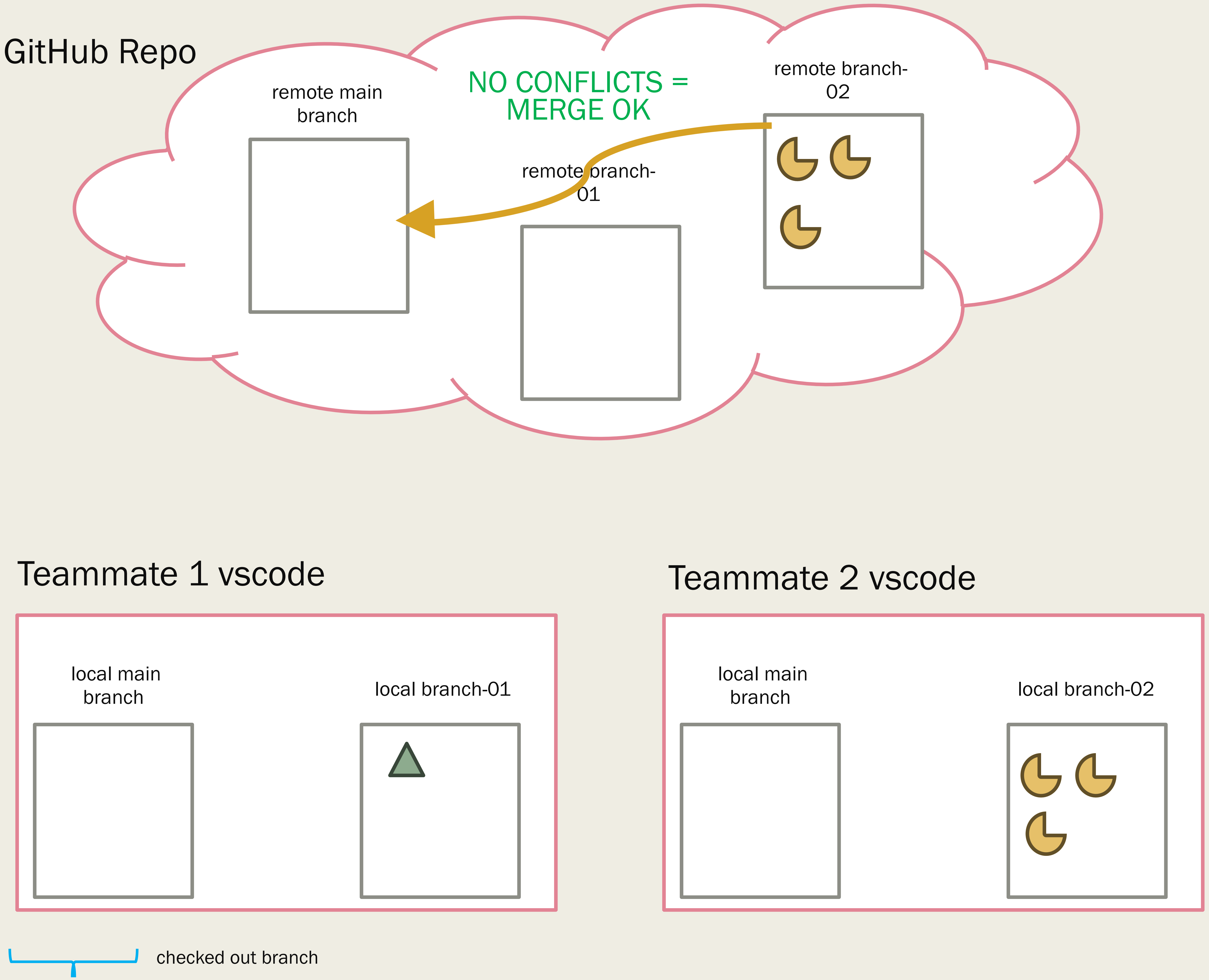
Each teammate has a local feature branch that has already been pushed to GitHub.

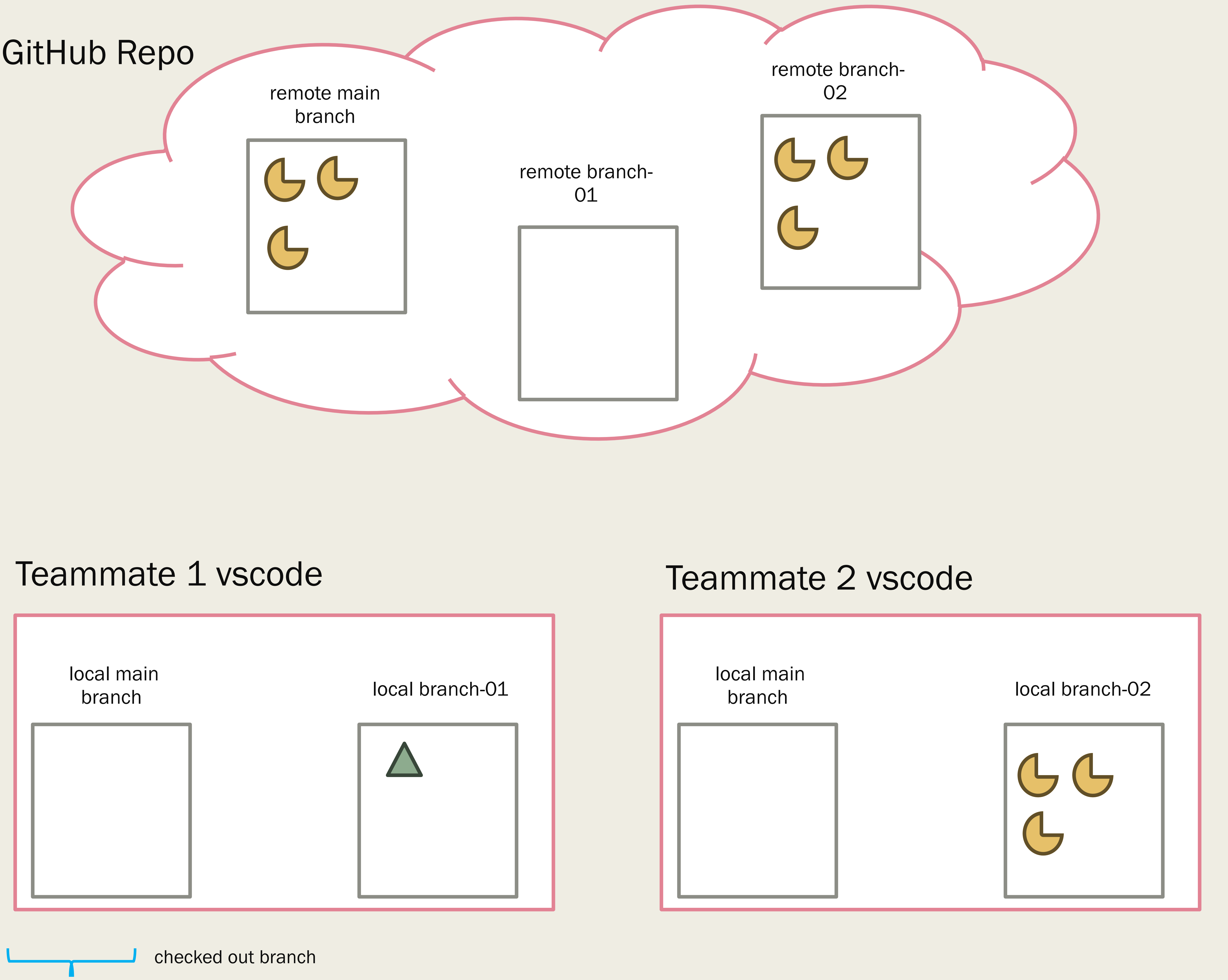

GitHub Repo
remote main branch
remote branch-01
remote branch-02
Teammate 1 vscode
local main branch
local branch-01
Teammate 2 vscode
local main branch
local branch-02

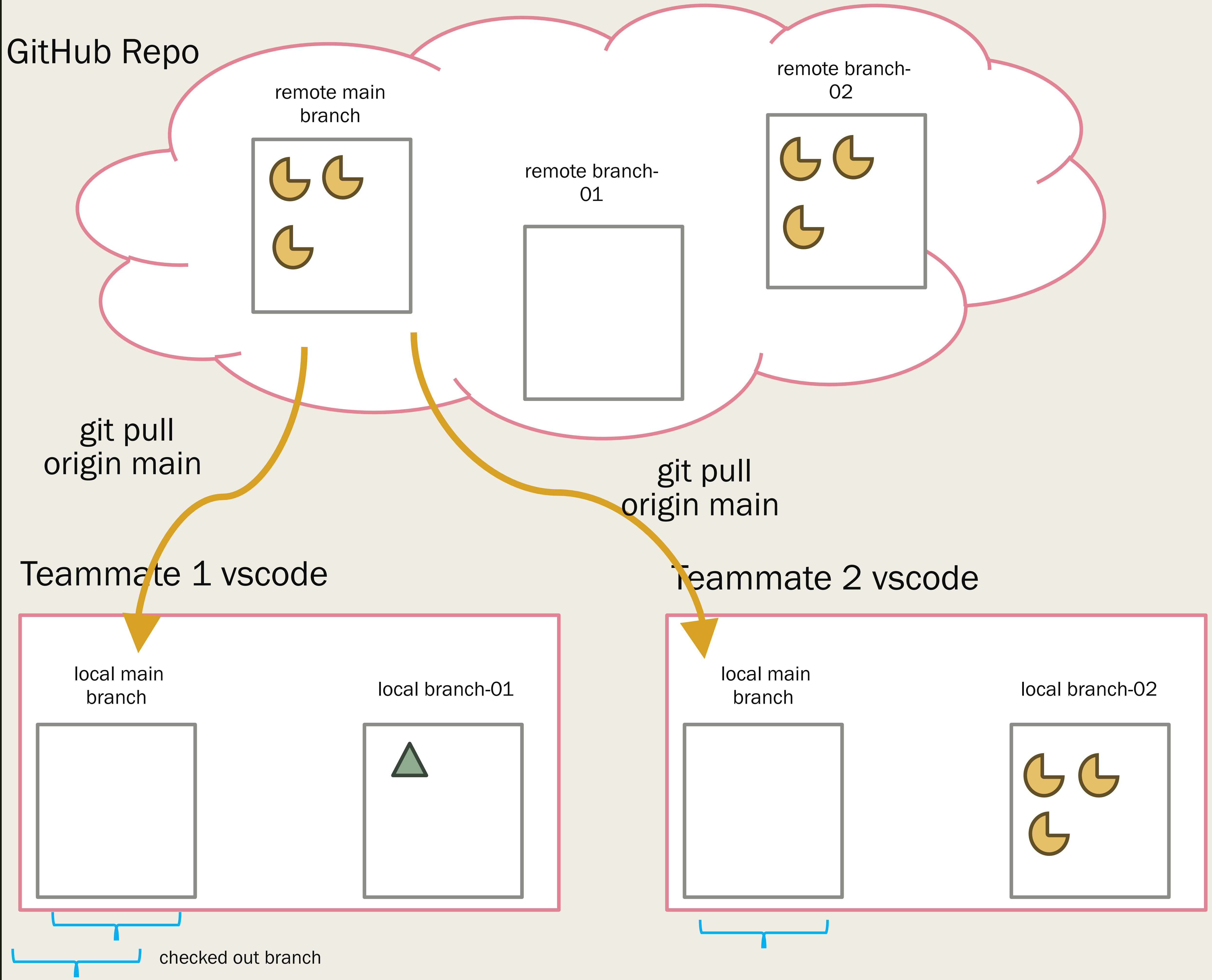

GitHub Repo
remote main branch
remote branch-01
remote branch-02
Teammate 1 vscode
local main branch
local branch-01
Teammate 2 vscode
local main branch
local branch-02
checked out branch

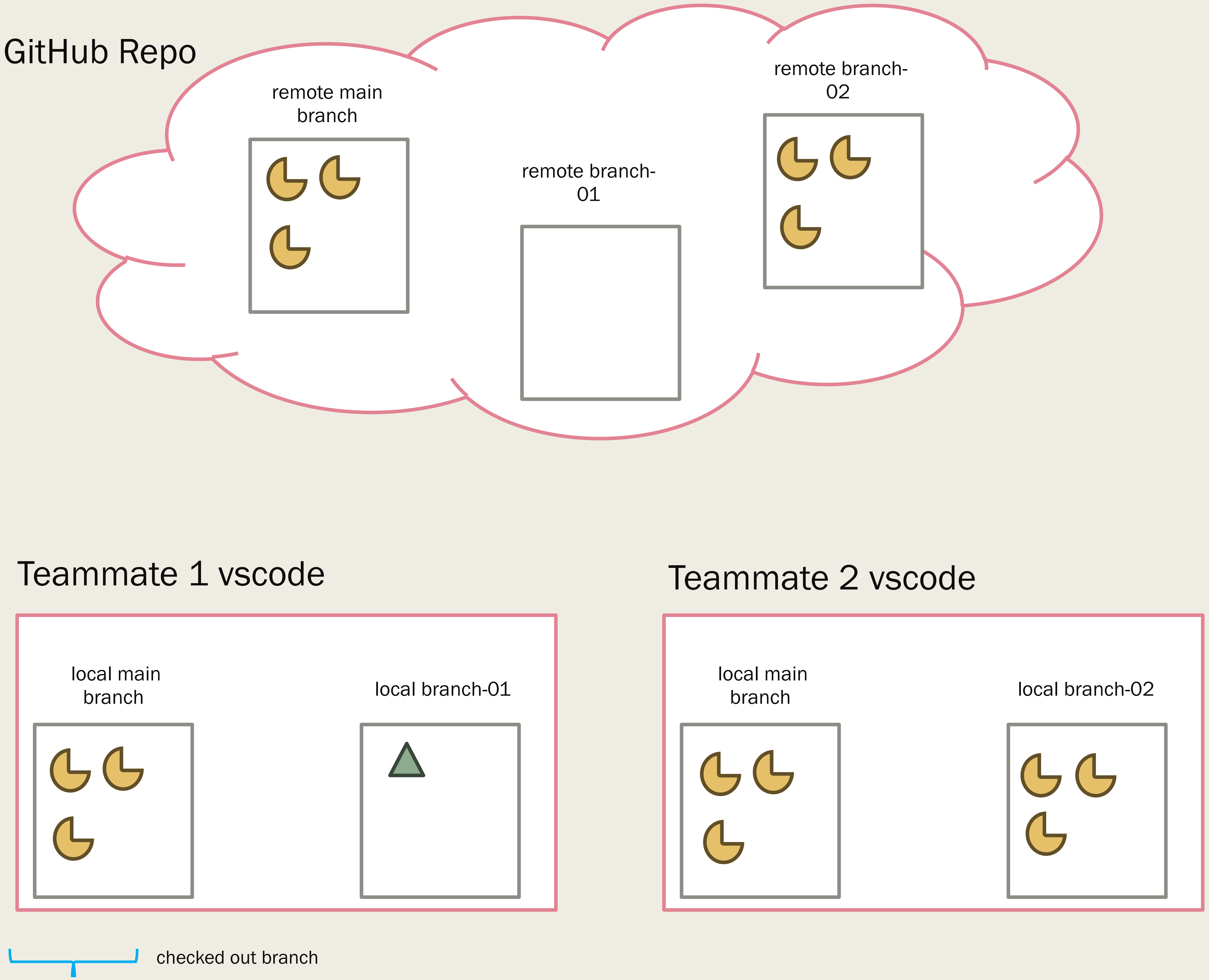

GitHub Repo
remote main branch
remote branch-01
remote branch-02
git push origin branch-02
Teammate 1 vscode
Teammate 2 vscode
local main branch
local branch-01
local main branch
local branch-02
checked out branch

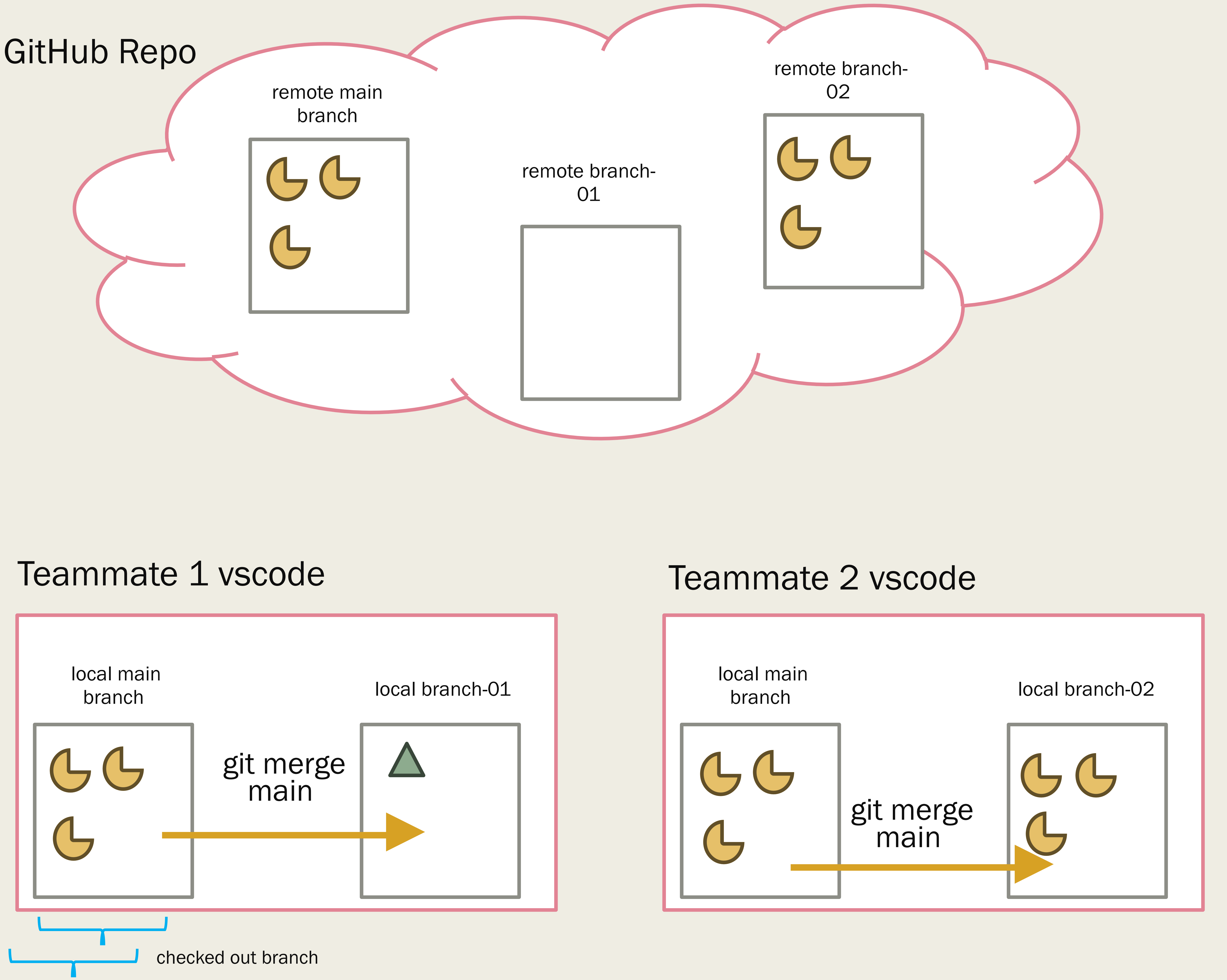

GitHub Repo
remote main branch
Pull Request
remote branch-02
remote branch-01
Teammate 1 vscode
local main branch
local branch-01
Teammate 2 vscode
local main branch
local branch-02
checked out branch

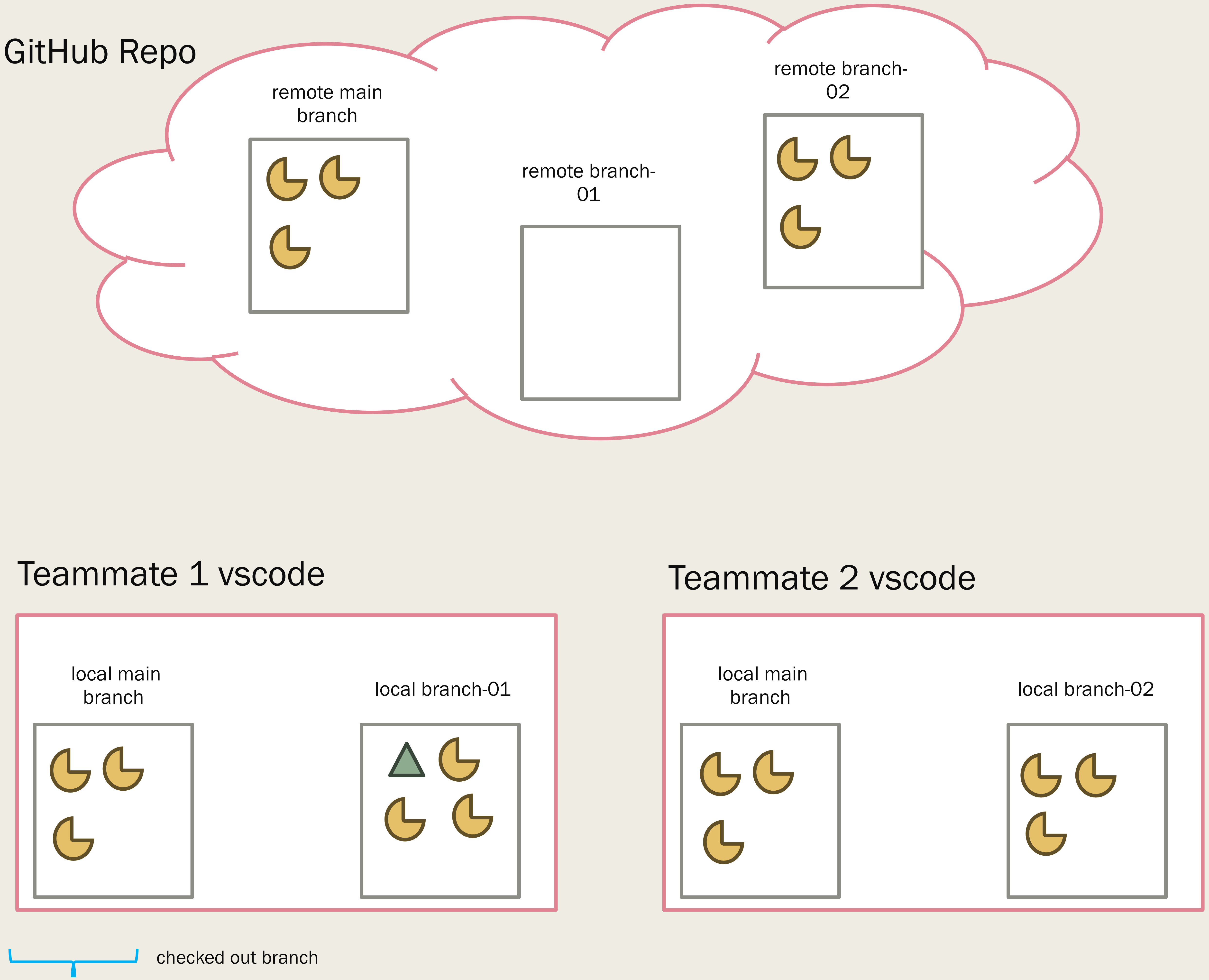

GitHub Repo
remote main branch
NO CONFLICTS = MERGE OK
remote branch-02
remote branch-01
Teammate 1 vscode
local main branch
local branch-01
Teammate 2 vscode
local main branch
local branch-02
checked out branch

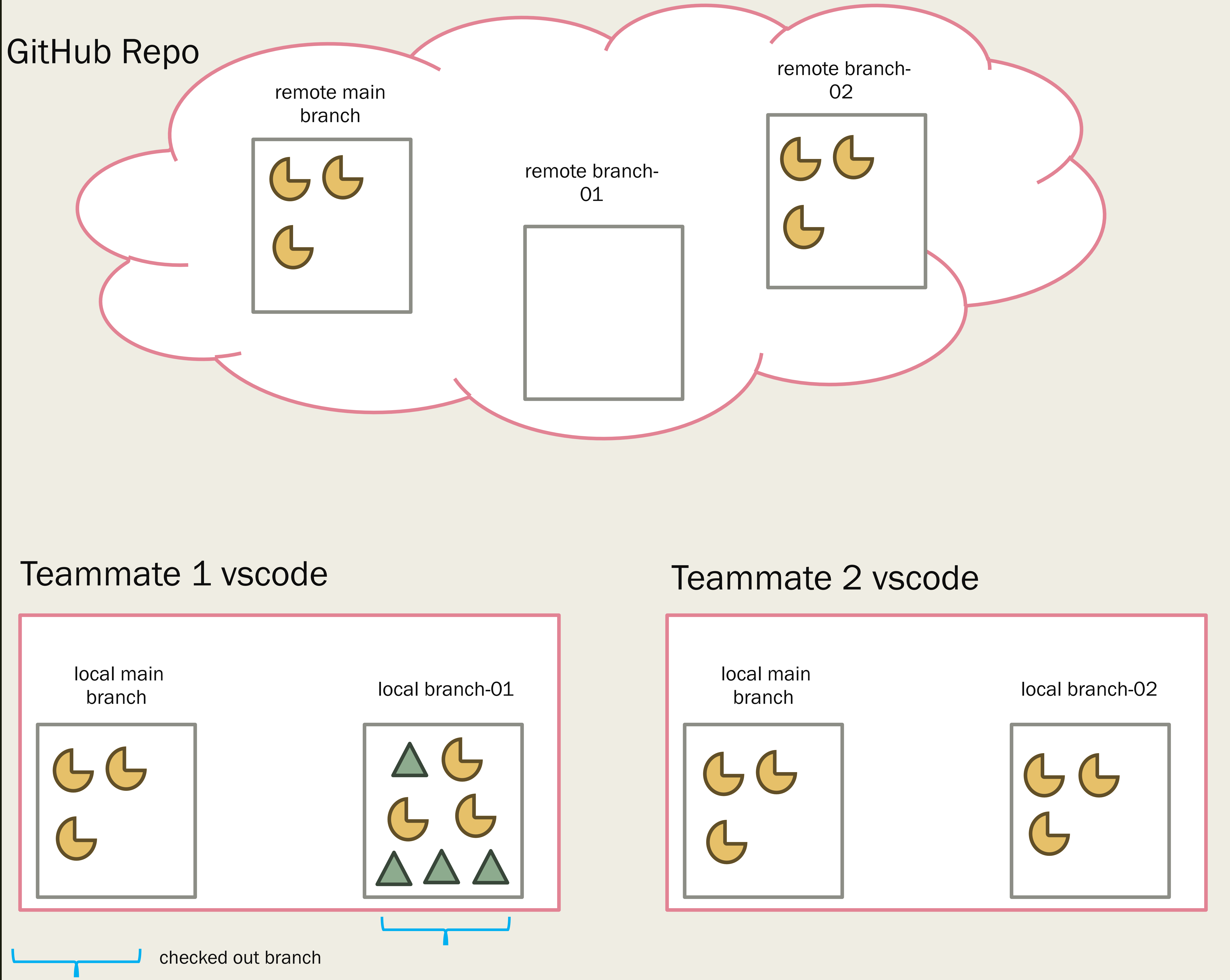

GitHub Repo
remote main branch
remote branch-01
remote branch-02
Teammate 1 vscode
local main branch
local branch-01
Teammate 2 vscode
local main branch
local branch-02
checked out branch

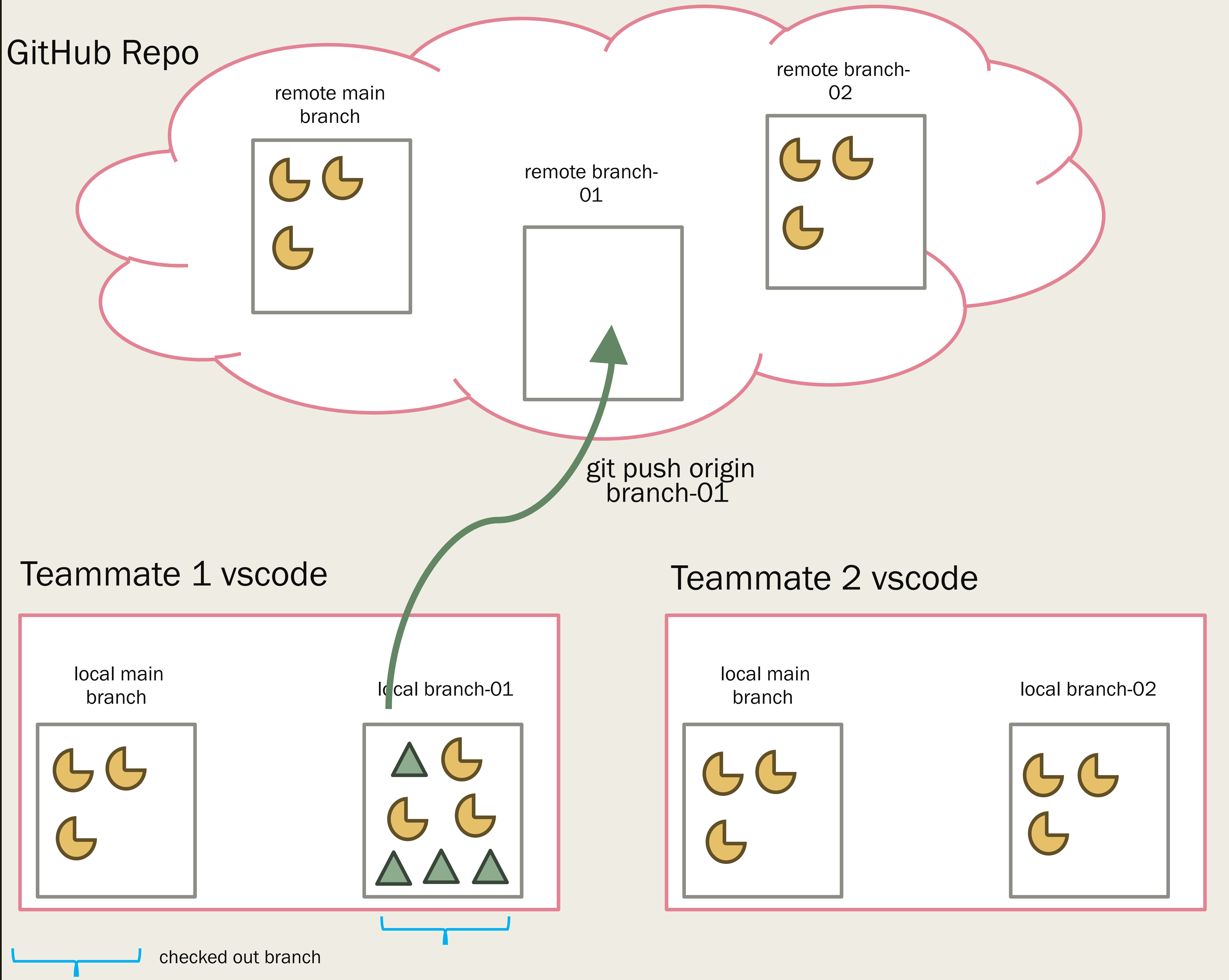

GitHub Repo
remote main branch
remote branch-01
remote branch-02
git pull origin main
git pull origin main
Teammate 1 vscode
Teammate 2 vscode
local main branch
local branch-01
local main branch
local branch-02
checked out branch

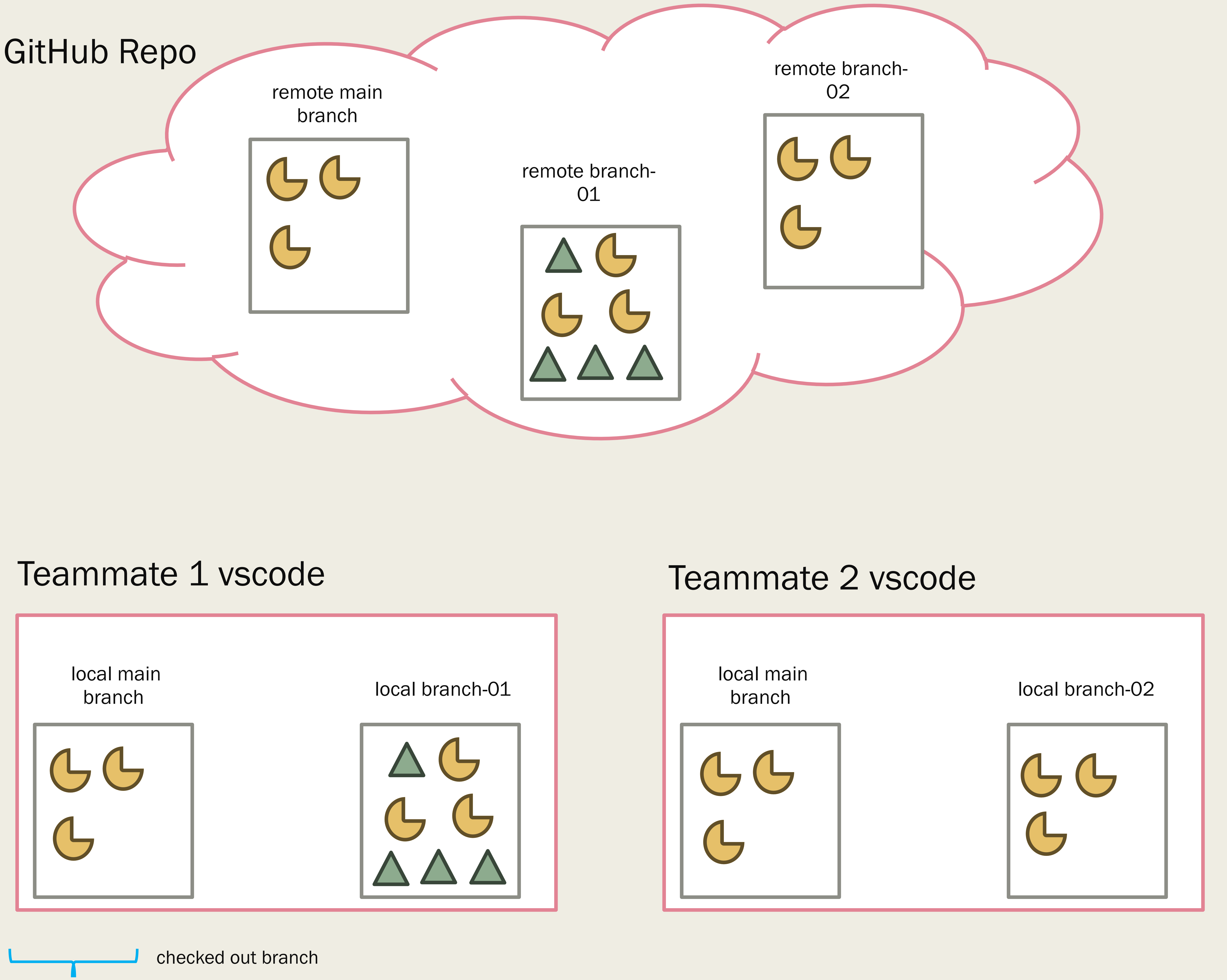

GitHub Repo
remote main branch
remote branch-01
remote branch-02
Teammate 1 vscode
local main branch
local branch-01
Teammate 2 vscode
local main branch
local branch-02
checked out branch

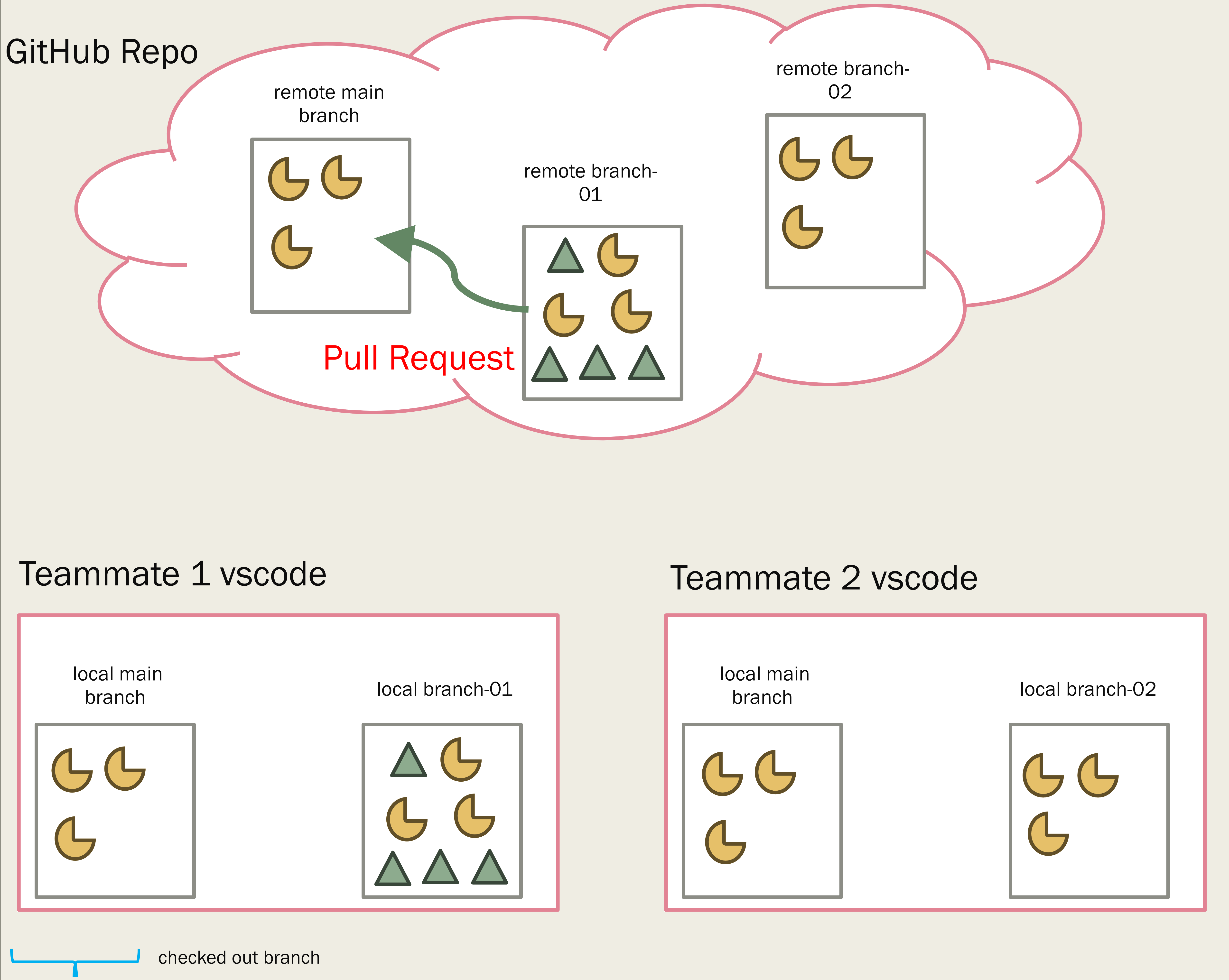

GitHub Repo
remote main branch
remote branch-01
remote branch-02
Teammate 1 vscode
local main branch
git merge main
local branch-01
Teammate 2 vscode
local main branch
git merge main
local branch-02
checked out branch


GitHub Repo
remote main branch
remote branch-01
remote branch-02
Teammate 1 vscode
local main branch
local branch-01
Teammate 2 vscode
local main branch
local branch-02
checked out branch

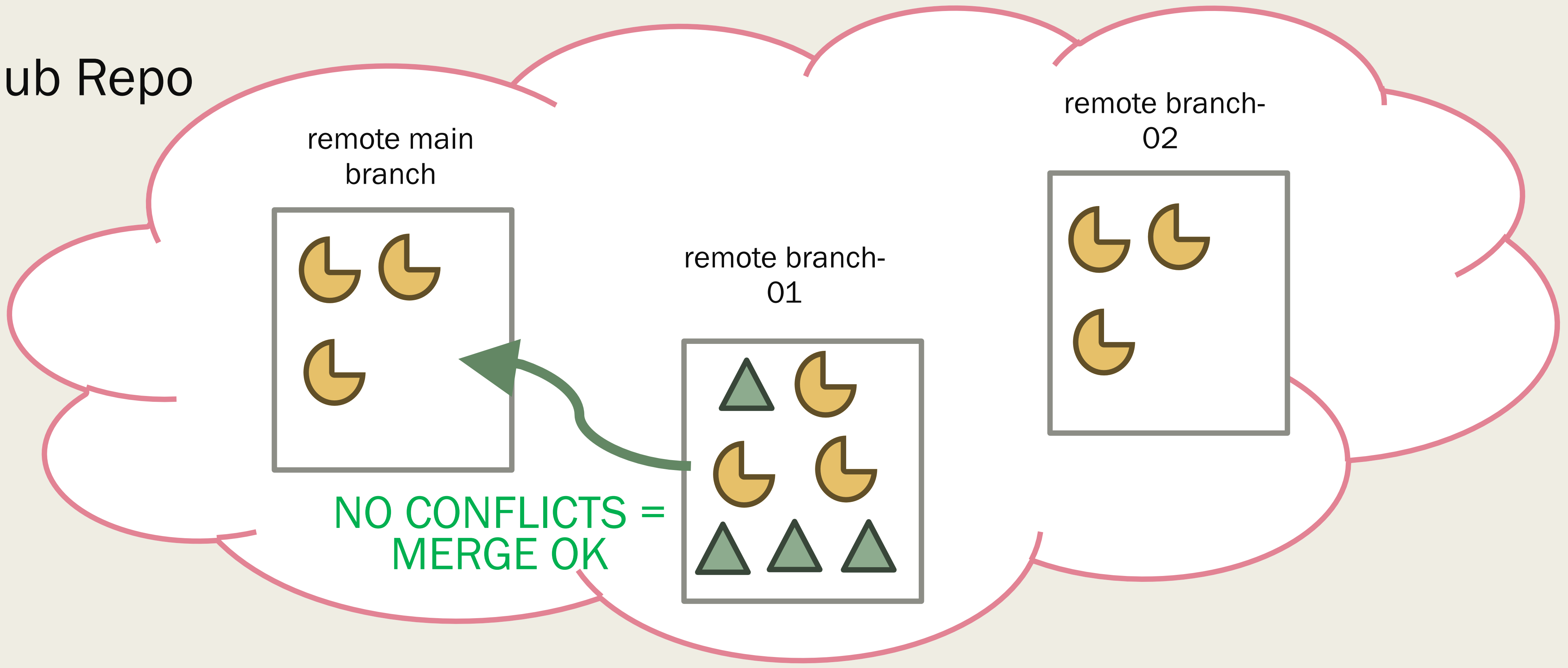

GitHub Repo
remote main branch
remote branch-01
remote branch-02
Teammate 1 vscode
local main branch
local branch-01
Teammate 2 vscode
local main branch
local branch-02
checked out branch

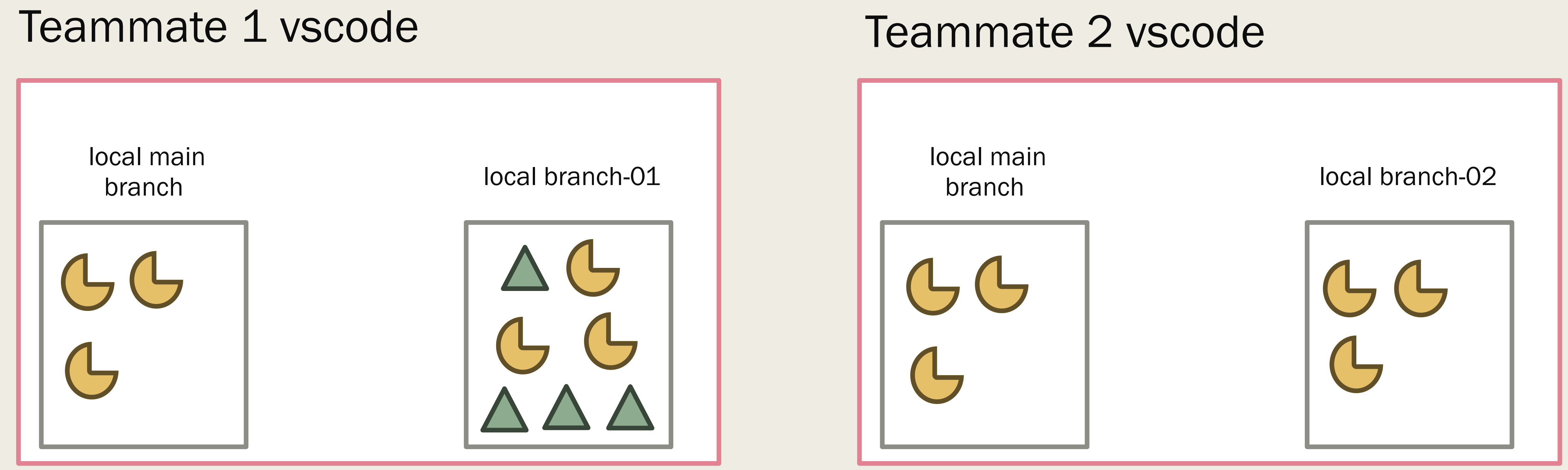

GitHub Repo
remote main branch
remote branch-01
remote branch-02
git push origin branch-01
Teammate 1 vscode
Teammate 2 vscode
local main branch
local branch-01
local main branch
local branch-02
checked out branch


GitHub Repo
remote main branch
remote branch-01
remote branch-02
Teammate 1 vscode
local main branch
local branch-01
Teammate 2 vscode
local main branch
local branch-02
checked out branch

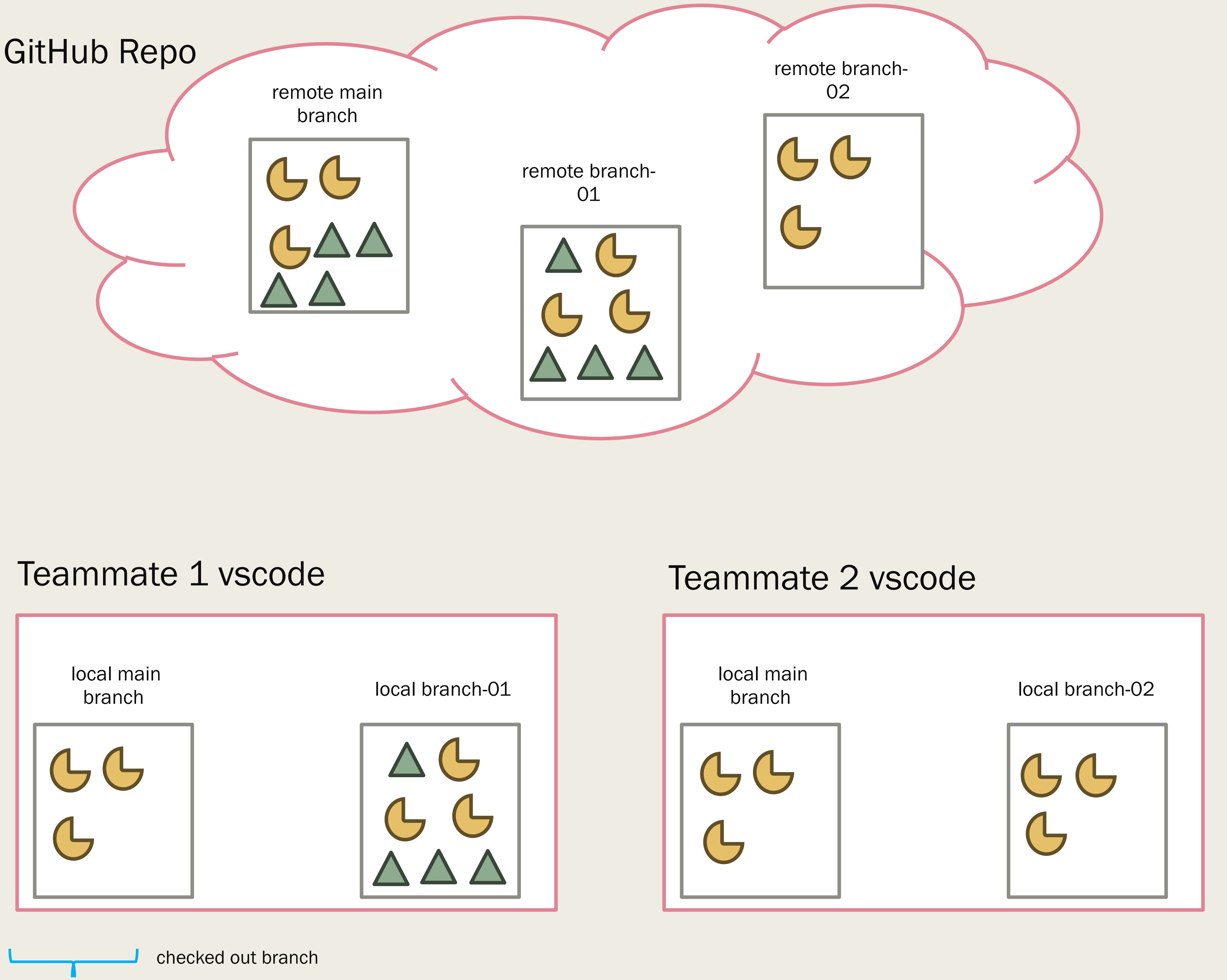

GitHub Repo
remote main branch
remote branch-01
remote branch-02
Pull Request
Teammate 1 vscode
local main branch
local branch-01
Teammate 2 vscode
local main branch
local branch-02
checked out branch

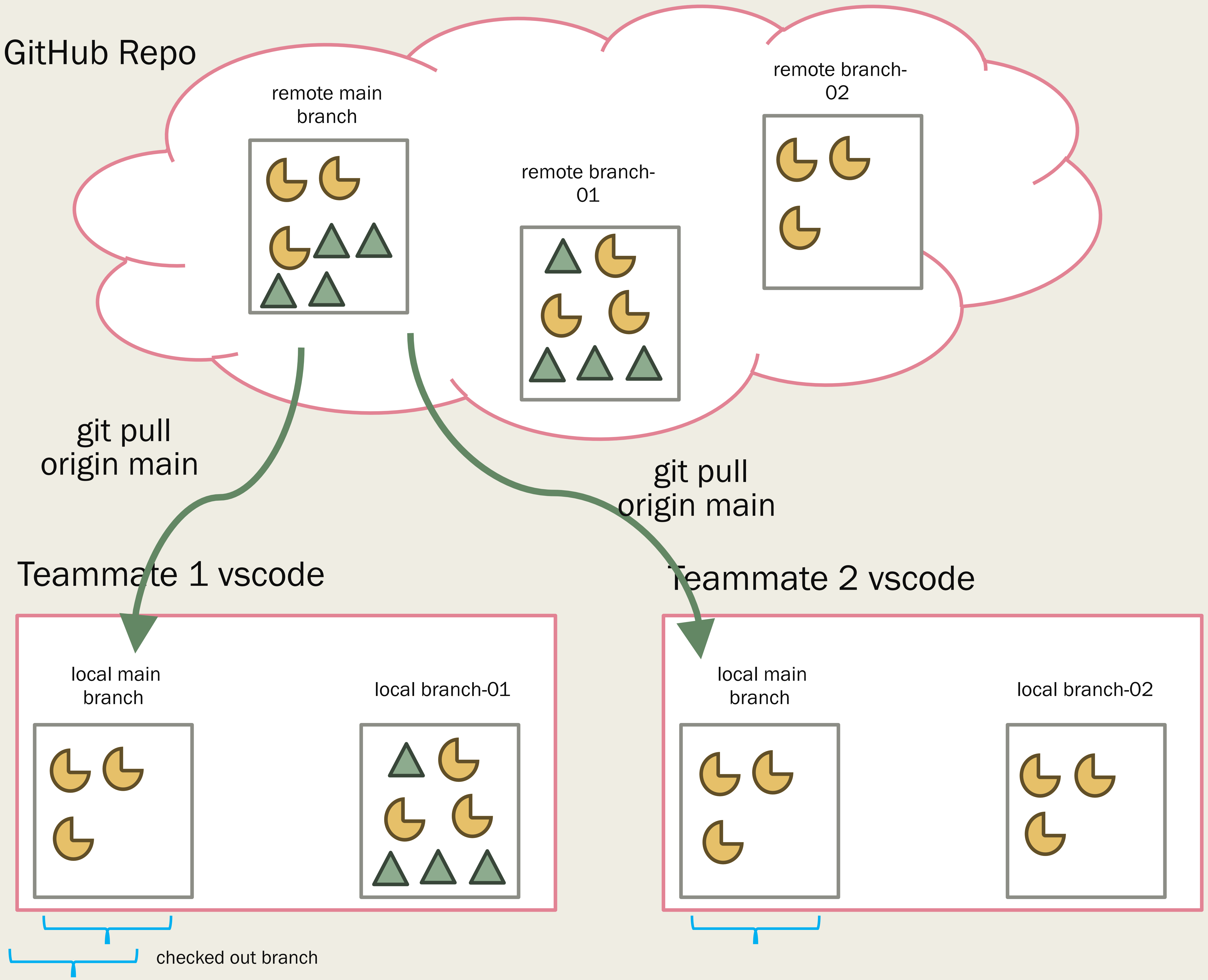
GitHub Repo


remote main branch
remote branch-01
remote branch-02
NO CONFLICTS = MERGE OK

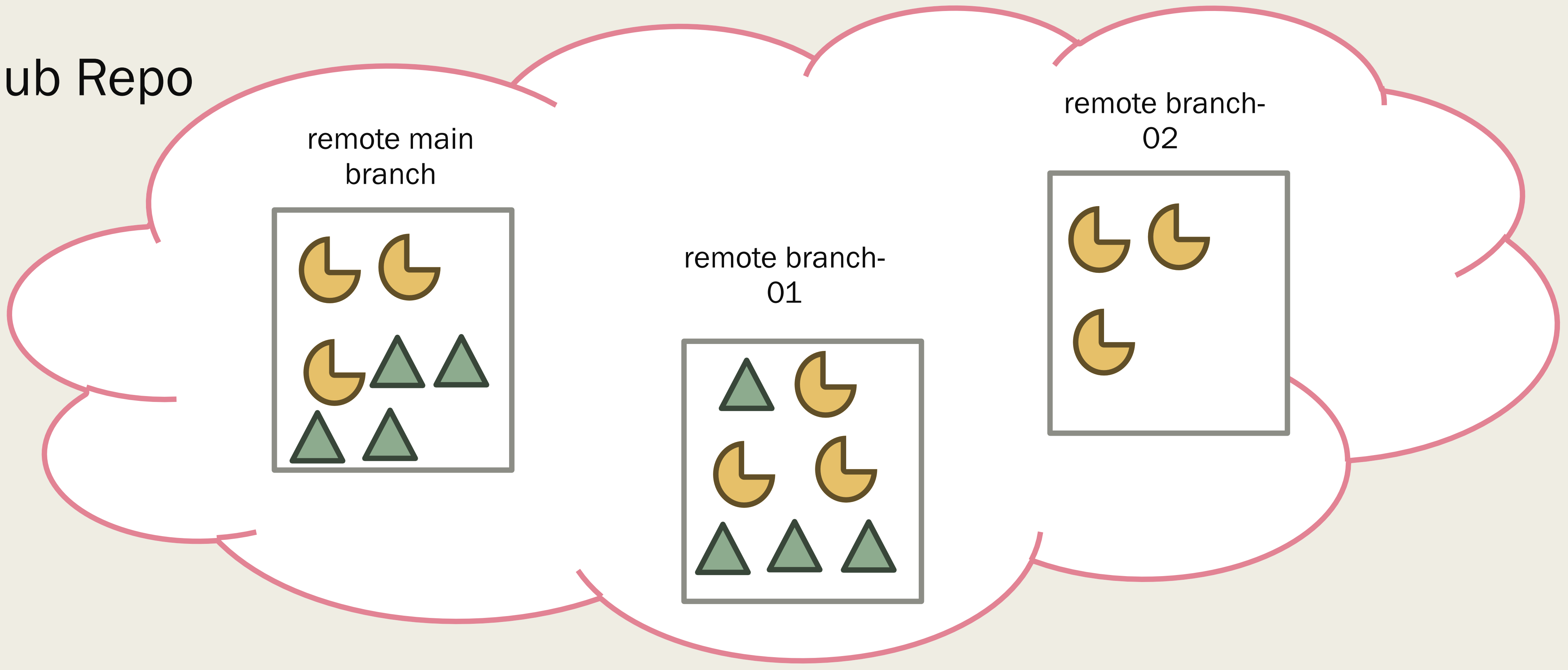

Teammate 1 vscode
local main branch
local branch-01
Teammate 2 vscode
local main branch
local branch-02

checked out branch


GitHub Repo
remote main branch
remote branch-01
remote branch-02
Teammate 1 vscode
local main branch
local branch-01
Teammate 2 vscode
local main branch
local branch-02
checked out branch

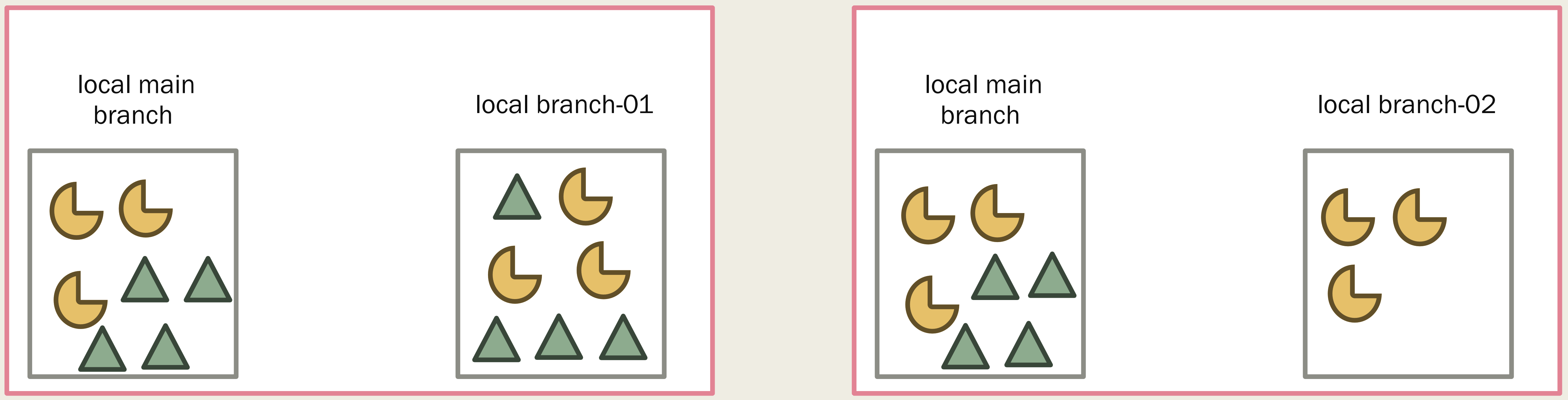

GitHub Repo
remote main branch
remote branch-01
remote branch-02
git pull origin main
git pull origin main
Teammate 1 vscode
Teammate 2 vscode
local main branch
local branch-01
local main branch
local branch-02
checked out branch

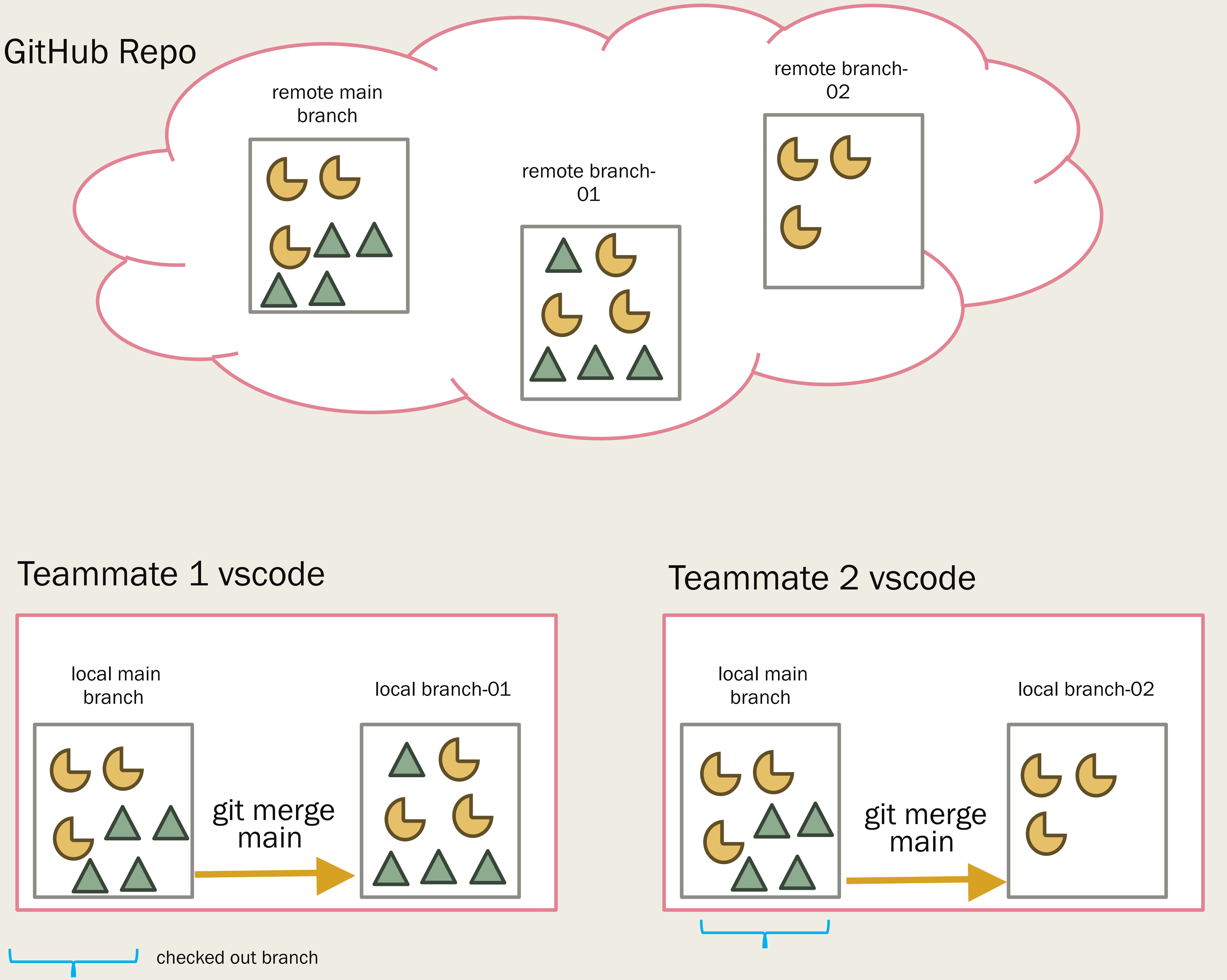
GitHub Repo

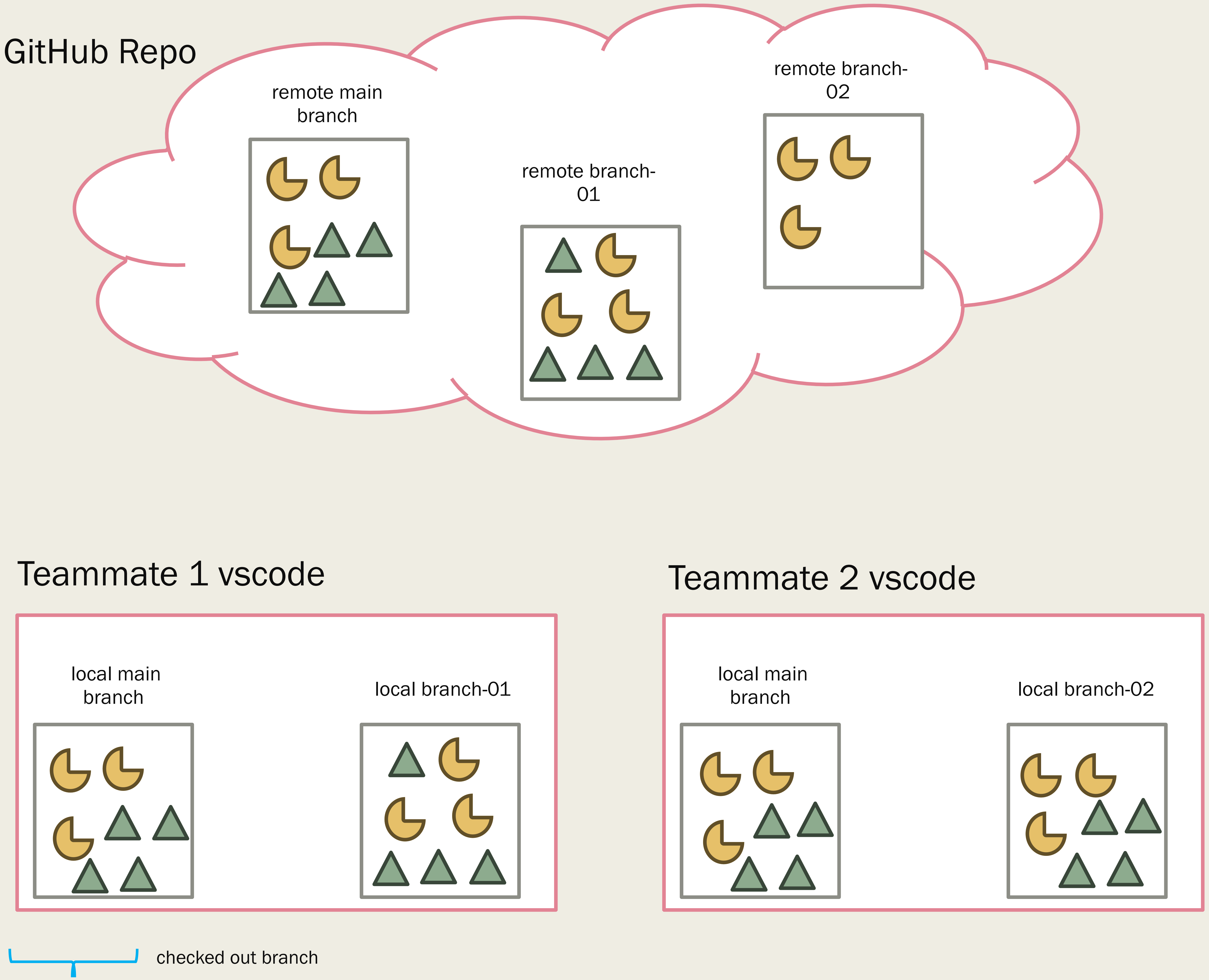


Teammate 1 vscode

Teammate 2 vscode




GitHub Repo
remote main branch
remote branch-01
remote branch-02
Teammate 1 vscode
local main branch
git merge main
local branch-01
Teammate 2 vscode
local main branch
git merge main
local branch-02
checked out branch


GitHub Repo
remote main branch
remote branch-01
remote branch-02
Teammate 1 vscode
local main branch
local branch-01
Teammate 2 vscode
local main branch
local branch-02
checked out branch

Quick Concept: Merging & Pull Requests

- **Merging**: Combines changes from one branch (e.g., a feature branch) into another (e.g., the main branch).
- **Pull Request:** A request to merge your changes. It's a place for review and discussion before the final merge.

  ➢ **When You Might See Merge Conflicts:**

- Conflicts occur when the same part of code is changed in different branches. In solo work, this is less common unless you're making changes to the main branch and your feature branch simultaneously.

Recap: Team Workflow

Communication is Key:

Always communicate with your team to ensure you're not working on the same files, reducing the risk of conflicts.

Pushing Changes:

Step 1: Push your changes from your local feature branch to the same branch on GitHub to share them with teammates.

Handling Pull Requests:

Step 2: Create a pull request from your remote feature branch to remote main. This is where you'll check for and resolve any merge conflicts.

Keeping Your Branch Updated:

Step 3: Regularly update your local main branch with the latest changes from the remote main.

-After updating, merge these changes into your feature branch. This ensures that when you push to GitHub, you're only merging your new work, minimizing conflicts.

Note: Perform these updates frequently, especially before starting new work or when other team members have completed their tasks.

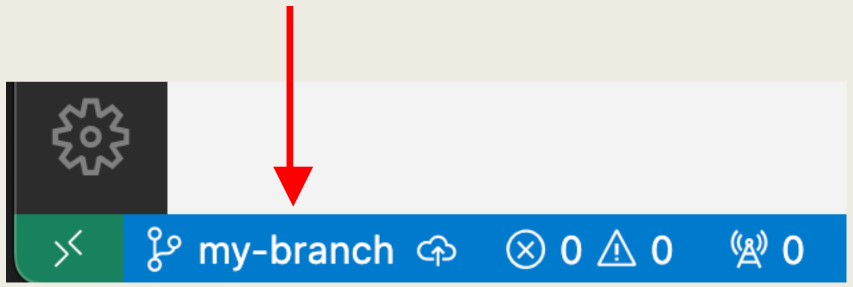
Exercise #4

- Now that we've made our changes locally, let's create a pull request on GitHub to merge the practice-branch into the main branch. This will replicate the workflow we just reviewed in the diagram.

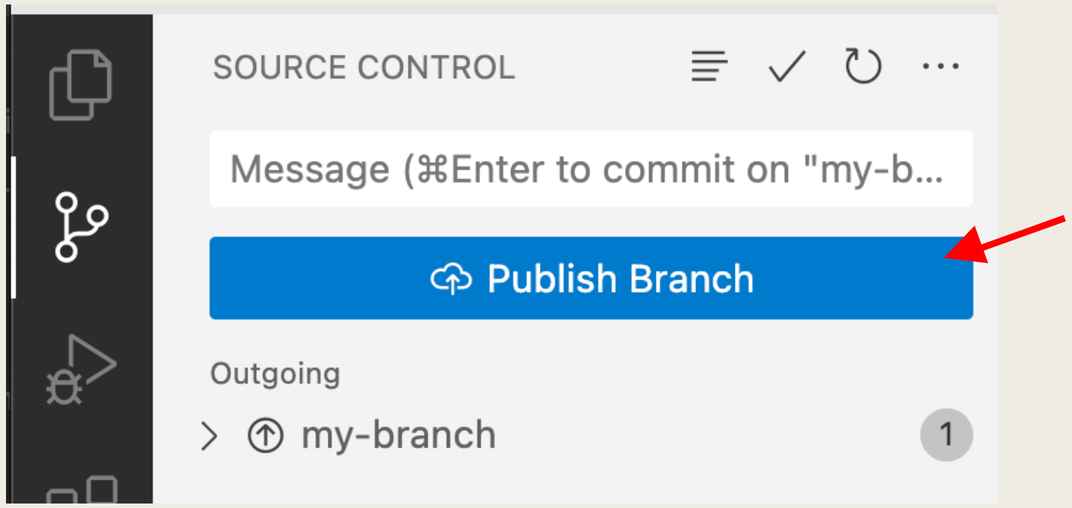
Push Changes to remote repo

VSCode View	Terminal

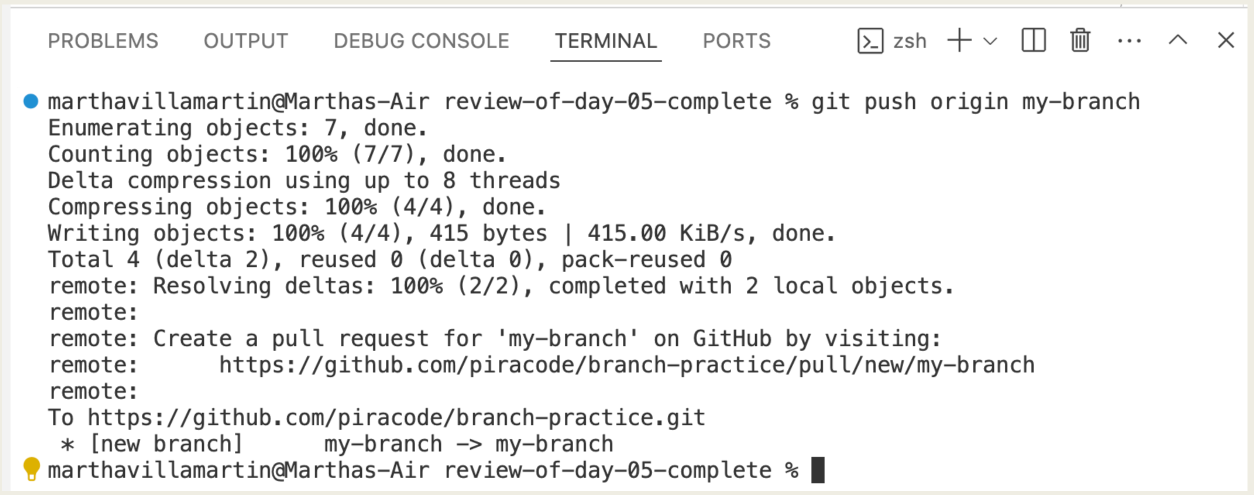
You can publish the branch and the changes we've made by clicking on the icon next to the branch name in the bottom-left corner



Or by clicking on "Publish Branch" in the Source Control tab



git push origin my-branch



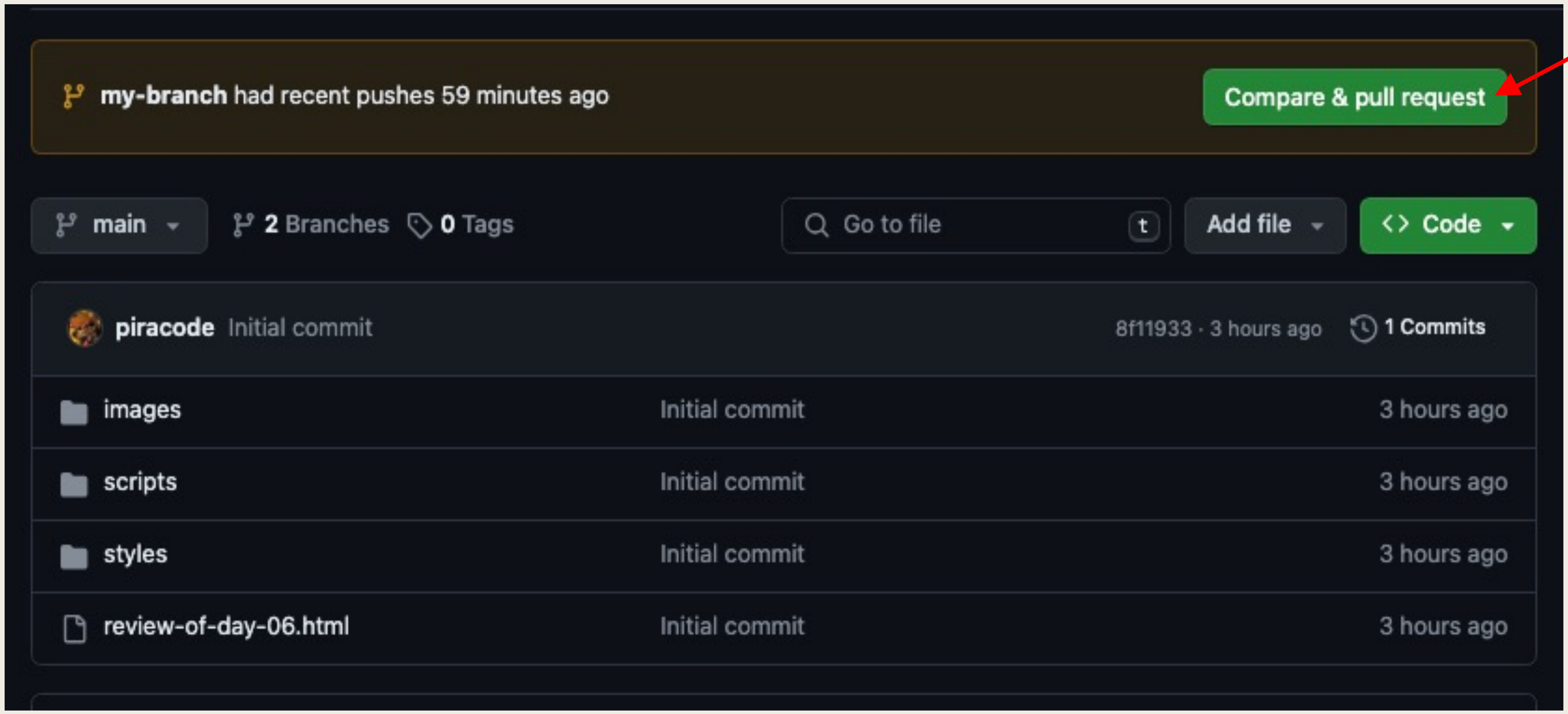
```
marthavillamartin@Marthas-Air review-of-day-05-complete % git push origin my-branch
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 415 bytes | 415.00 KiB/s, done.
Total 4 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
remote:
remote: Create a pull request for 'my-branch' on GitHub by visiting:
remote:      https://github.com/piracode/branch-practice/pull/new/my-branch
remote:
To https://github.com/piracode/branch-practice.git
 * [new branch]      my-branch -> my-branch
marthavillamartin@Marthas-Air review-of-day-05-complete %
```

This command will publish the branch and push the committed changes to the remote repo

Heads-up: If VSCode still shows "Publish Branch" post-push, it's a sync glitch. Click to clear or verify via terminal.

Step 1a: Initiate Pull Request on GitHub

- After pushing your branch, use the GitHub interface to open a pull request and prepare to merge your changes.



my-branch had recent pushes 59 minutes ago

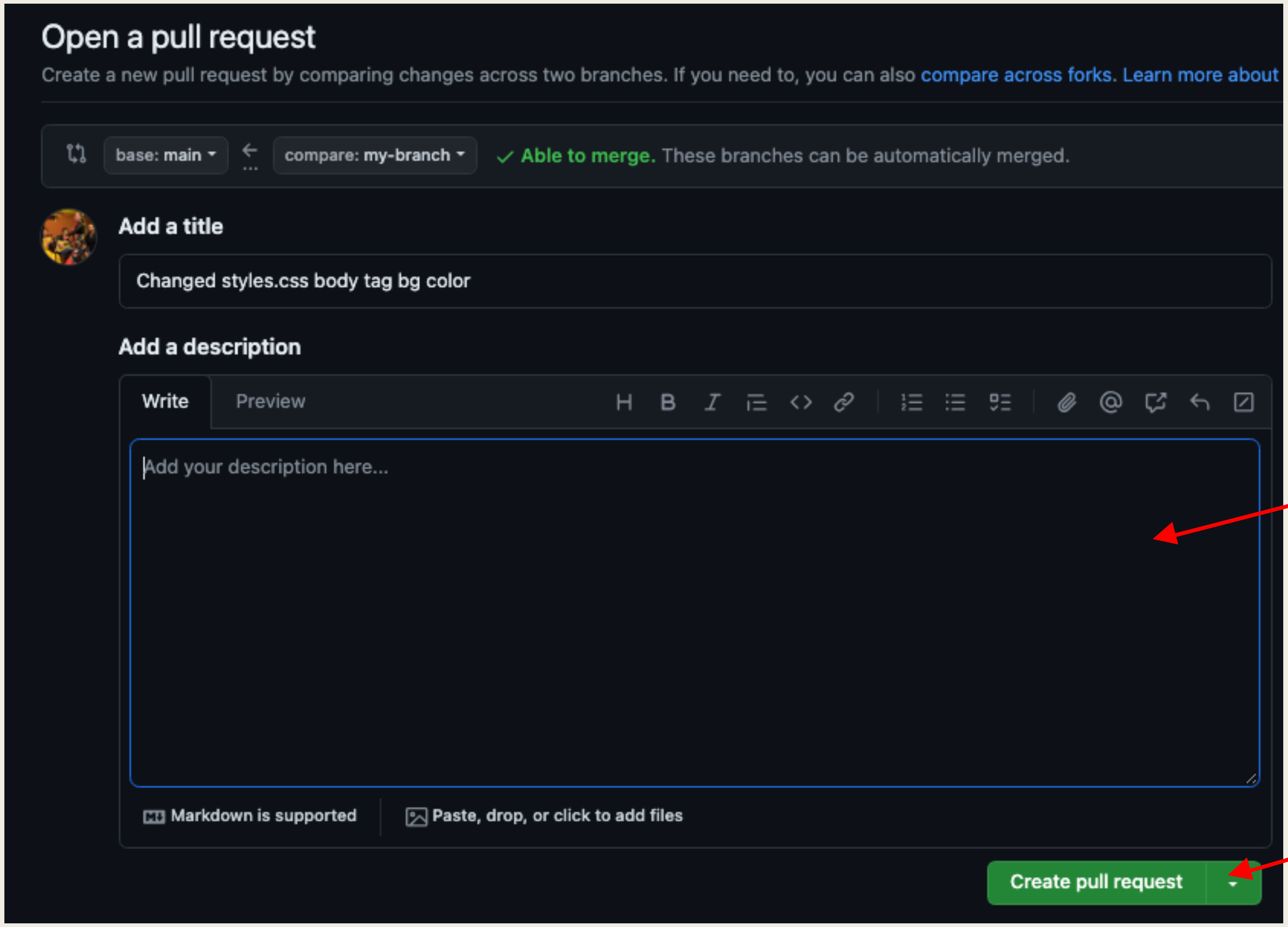
piracode Initial commit 8f11933 · 3 hours ago 1 Commits

images	Initial commit	3 hours ago
scripts	Initial commit	3 hours ago
styles	Initial commit	3 hours ago
review-of-day-06.html	Initial commit	3 hours ago

Click here

Step 1b: Completing Pull Request Details on GitHub

- Ensure the pull request title is clear—it's often pre-populated from your commit message. Adding details in the description is optional but recommended for clarity.


Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks. Learn more about

base: main ← compare: my-branch ✓ Able to merge. These branches can be automatically merged.

Add a title

Changed styles.css body tag bg color

Add a description

Write Preview

Add your description here...

Markdown is supported Paste, drop, or click to add files

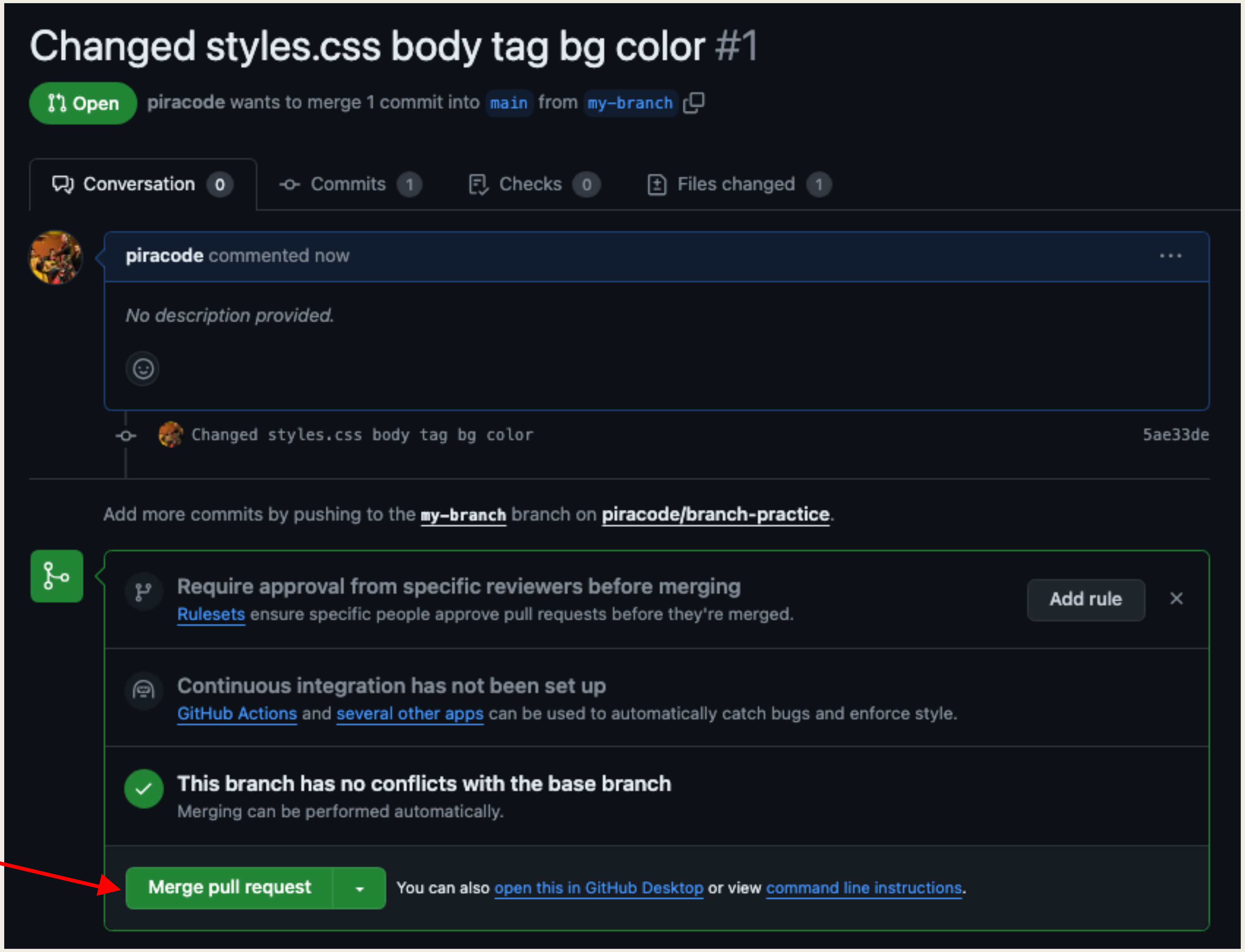
Create pull request

For this practice, we can leave it blank, but in a real-world setting, always include one for clarity and context. This helps reviewers understand the purpose of the changes and any implications they may have.

Click here

Step 1c: Finalizing Pull Request Details

- After ensuring the pull request is clear and ready to merge, additional comments are optional and typically used for collaboration or to request changes.


Changed styles.css body tag bg color #1

Open piracode wants to merge 1 commit into `main` from `my-branch`

Conversation 0 | Commits 1 | Checks 0 | Files changed 1

piracode commented now

No description provided.

`Changed styles.css body tag bg color` 5ae33de

Add more commits by pushing to the **my-branch** branch on **piracode/branch-practice**.

Require approval from specific reviewers before merging
Rulesets ensure specific people approve pull requests before they're merged.

Add rule

Continuous integration has not been set up
GitHub Actions and several other apps can be used to automatically catch bugs and enforce style.

This branch has no conflicts with the base branch
Merging can be performed automatically.

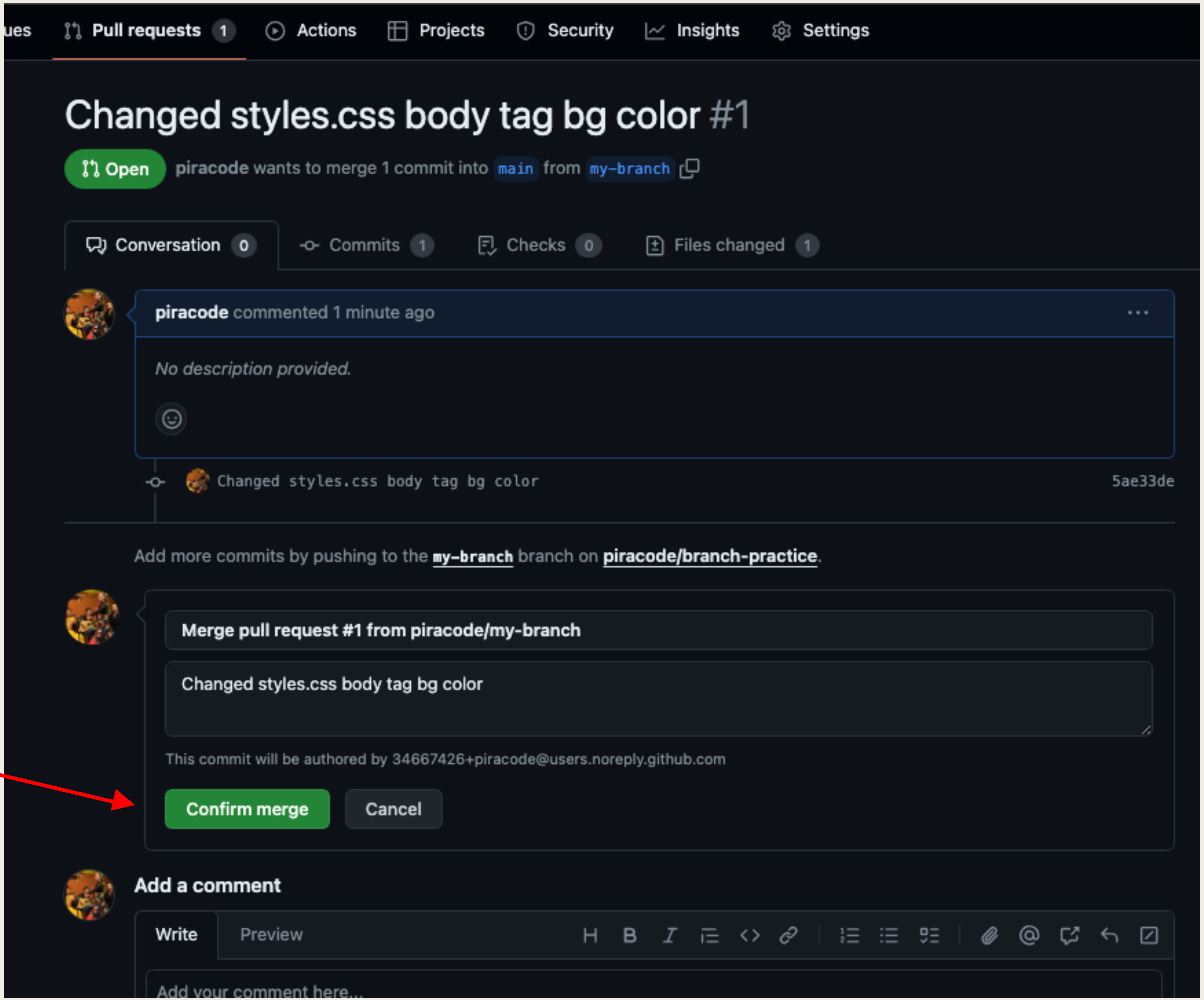
Merge pull request | You can also open this in GitHub Desktop or view command line instructions.

Click here

Step 4: Confirm the Merge of PR

- Click 'Confirm merge' to finalize the merge of your pull request after review, integrating your changes into the main branch

Click here



Changed styles.css body tag bg color #1

Open piracode wants to merge 1 commit into main from my-branch

piracode commented 1 minute ago

No description provided.

Changed styles.css body tag bg color 5ae33de

Add more commits by pushing to the my-branch branch on piracode/branch-practice.

Merge pull request #1 from piracode/my-branch

Changed styles.css body tag bg color

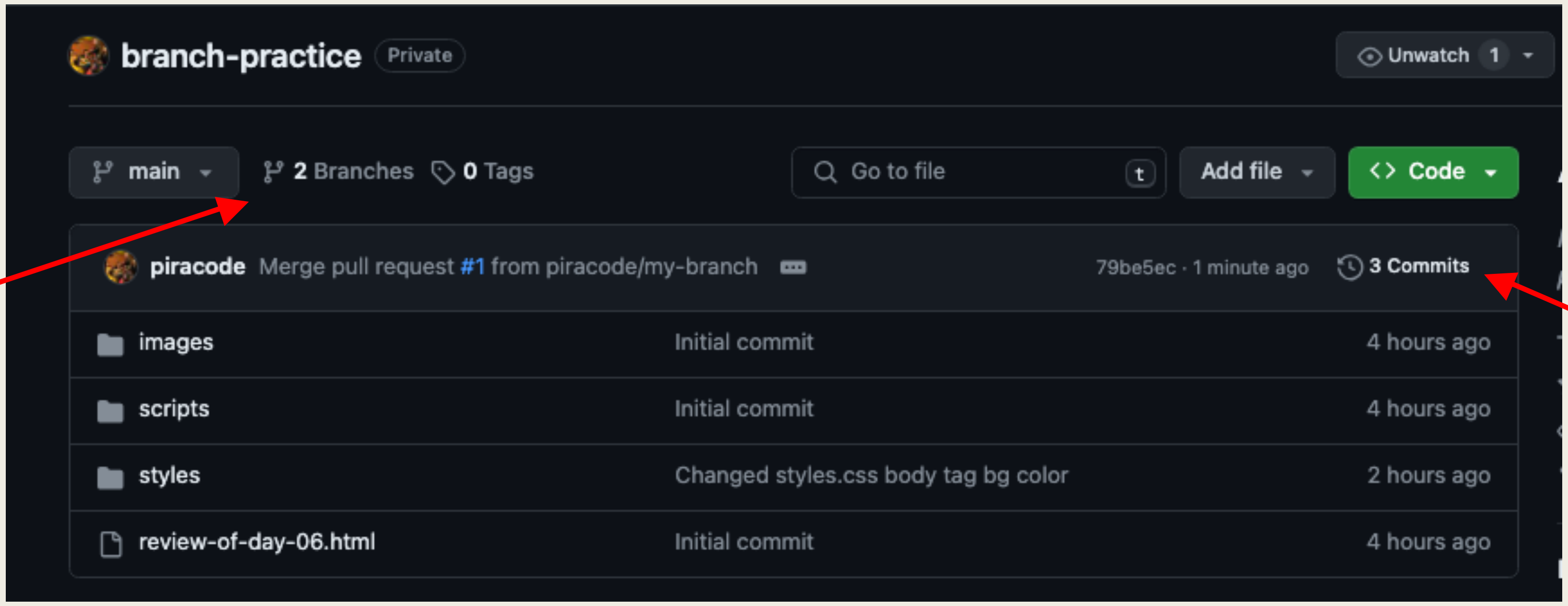
This commit will be authored by 34667426+piracode@users.noreply.github.com

Add your comment here...

(Optional) Check list of commits

- Click on your repo to see details such as the list of branches and the commits made.

Click here for a list of branches and detailed information on each.

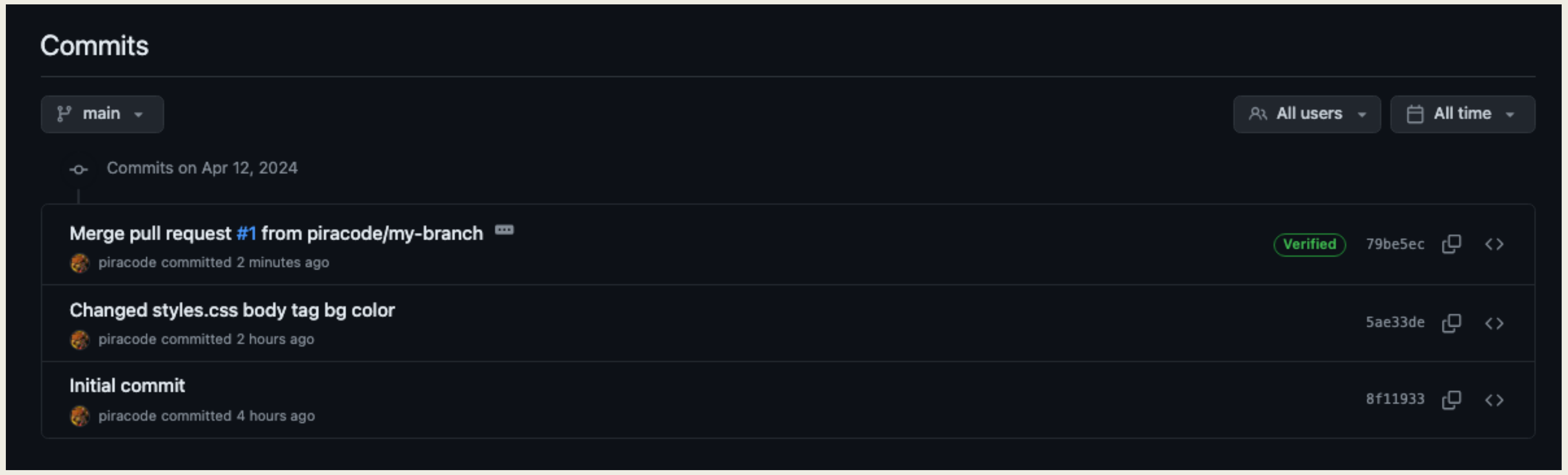


piracode Merge pull request #1 from piracode/my-branch 79be5ec · 1 minute ago

images	Initial commit	4 hours ago
scripts	Initial commit	4 hours ago
styles	Changed styles.css body tag bg color	2 hours ago
review-of-day-06.html	Initial commit	4 hours ago

Click here for a list of commits and detailed information on each.

Once the list of commits is displayed, you can click on each one to view the details, including the deletions and additions made.



Commits on Apr 12, 2024

Merge pull request #1 from piracode/my-branch piracode committed 2 minutes ago	Verified	79be5ec
Changed styles.css body tag bg color piracode committed 2 hours ago		5ae33de
Initial commit piracode committed 4 hours ago		8f11933

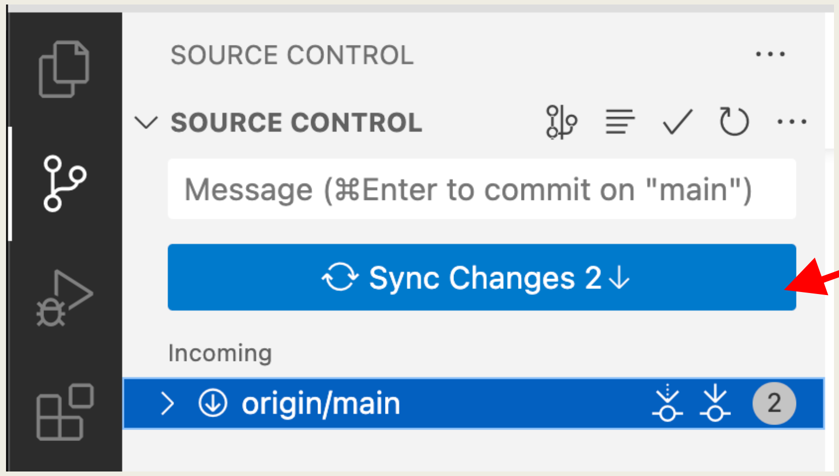
Exercise #5

- Update your local main branch to reflect the changes that we merged to the remote main branch.

Pull Latest Changes from Remote Main

VSCode View	Terminal

Switch to the main branch using the branch selector at the bottom-left corner.



Click the "Sync Changes" button to fetch and merge changes from the remote main branch.

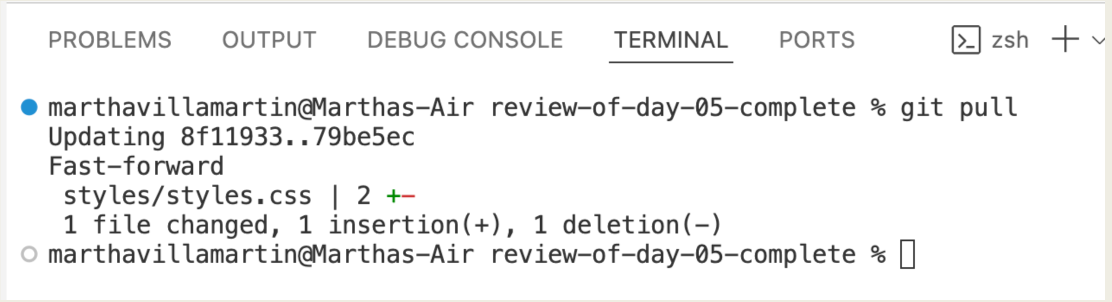
Or by clicking on "Publish Branch" in the Source Control tab

1) Make sure you're on the main branch

   git checkout main

2) Fetch and merge the changes from the remote repository

   git pull origin main*



```
● marthavillamartin@Marthas-Air review-of-day-05-complete % git pull
  Updating 8f11933..79be5ec
  Fast-forward
   styles/styles.css | 2 +-
   1 file changed, 1 insertion(+), 1 deletion(-)
○ marthavillamartin@Marthas-Air review-of-day-05-complete % 
```

*Note: If you have set the upstream command "git push –u origin main", you can just run "git pull"

By doing this, your local main branch will be up-to-date with the remote repository, and you'll be ready to start new work from a current base.

Branch Naming Conventions & Best Practices

Good branch names help everyone understand what you're working on:

- **For New Features:** “feature/contact-form” for adding a new contact form.
- **To Fix Bugs**: “fix/responsive” for fixing responsive design issues.
- **To make improvements**: “update/change-colors” for updating the color scheme.

Tips :

- Start with a keyword (feature, update, fix) to indicate the purpose of the branch.
- Follow with a short description of what you're doing.
- Use hyphens to separate words for readability.
- Keep it short and meaningful.

Remember:

- Always switch to the correct branch before you start working (git checkout branch-name).
- Save and share your work often (git push).

How to get out of VIM mode

1. Press Esc to ensure you're in normal mode.
2. Type :q! to quit without saving changes, or :wq to save changes and quit.
3. Press Enter.