GITHUB-INTRODUCTION

What is GitHub?

GitHub is a cloud development platform that enables us to

- host and review code
- manage projects
- build software
- Handles version control

Key features

Issues

Discussions

Pull requests

Notifications

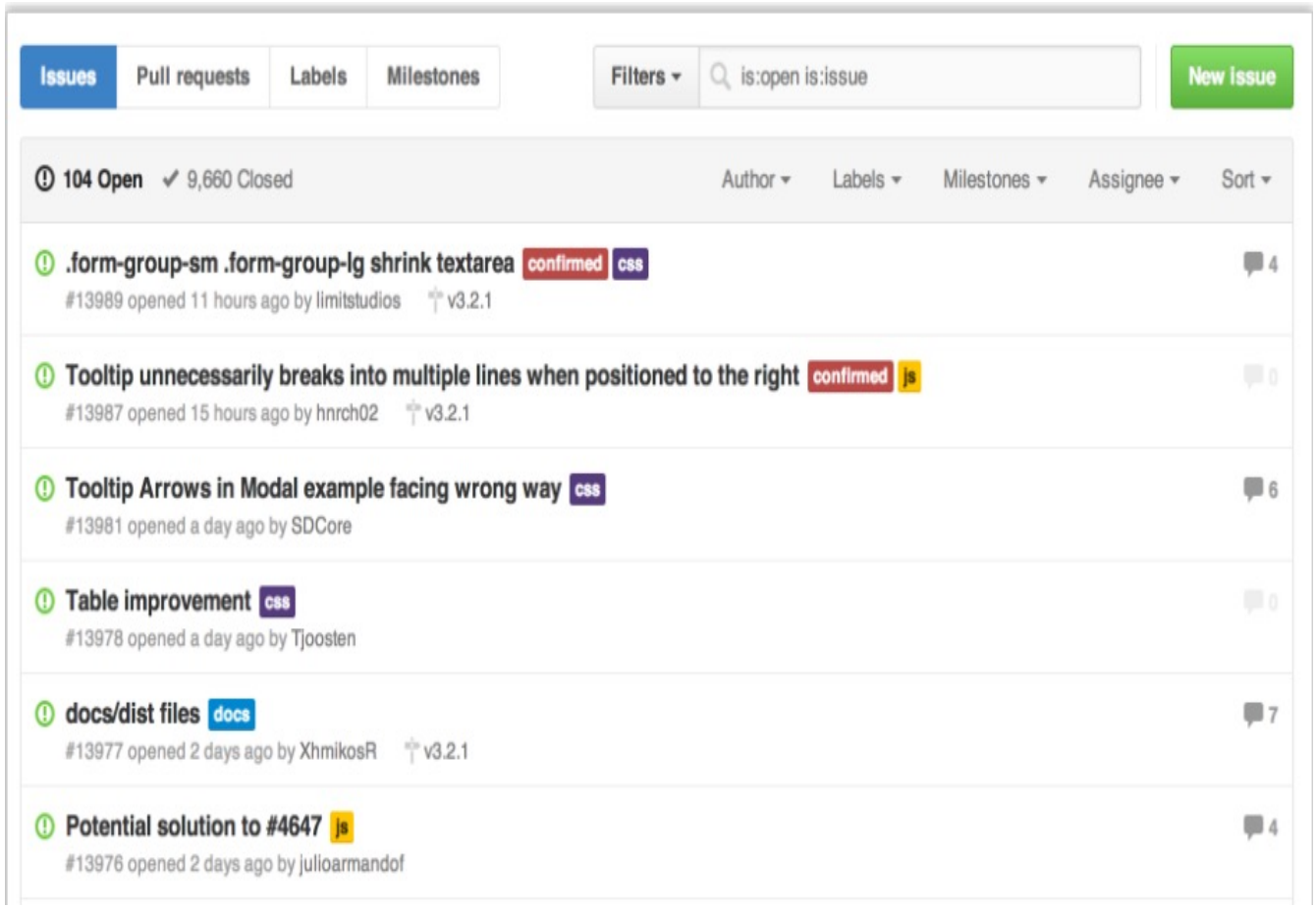
Labels

Actions

Forks

Projects

issues



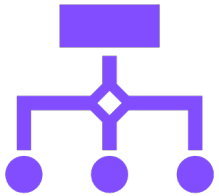
Issues are where most of the communication between a project's consumers and development team occurs. An issue can be created to discuss a broad set of topics, including

- Bug Reports
- Feature Requests
- Documentation clarification

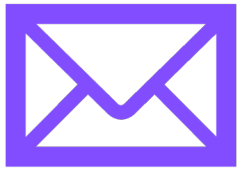
Once an issue has been created, It can be assigned to owners, labels,

projects, and milestones

Notifications



As a collaborative platform, GitHub offers **notifications** for virtually every event that takes place within a given workflow. These notifications can be finely tuned to meet your preferences

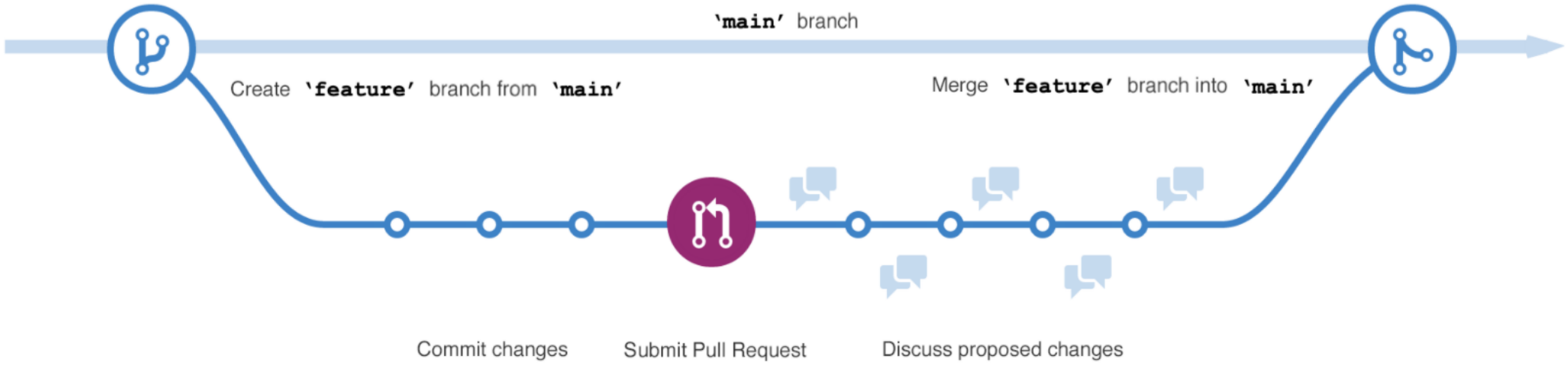


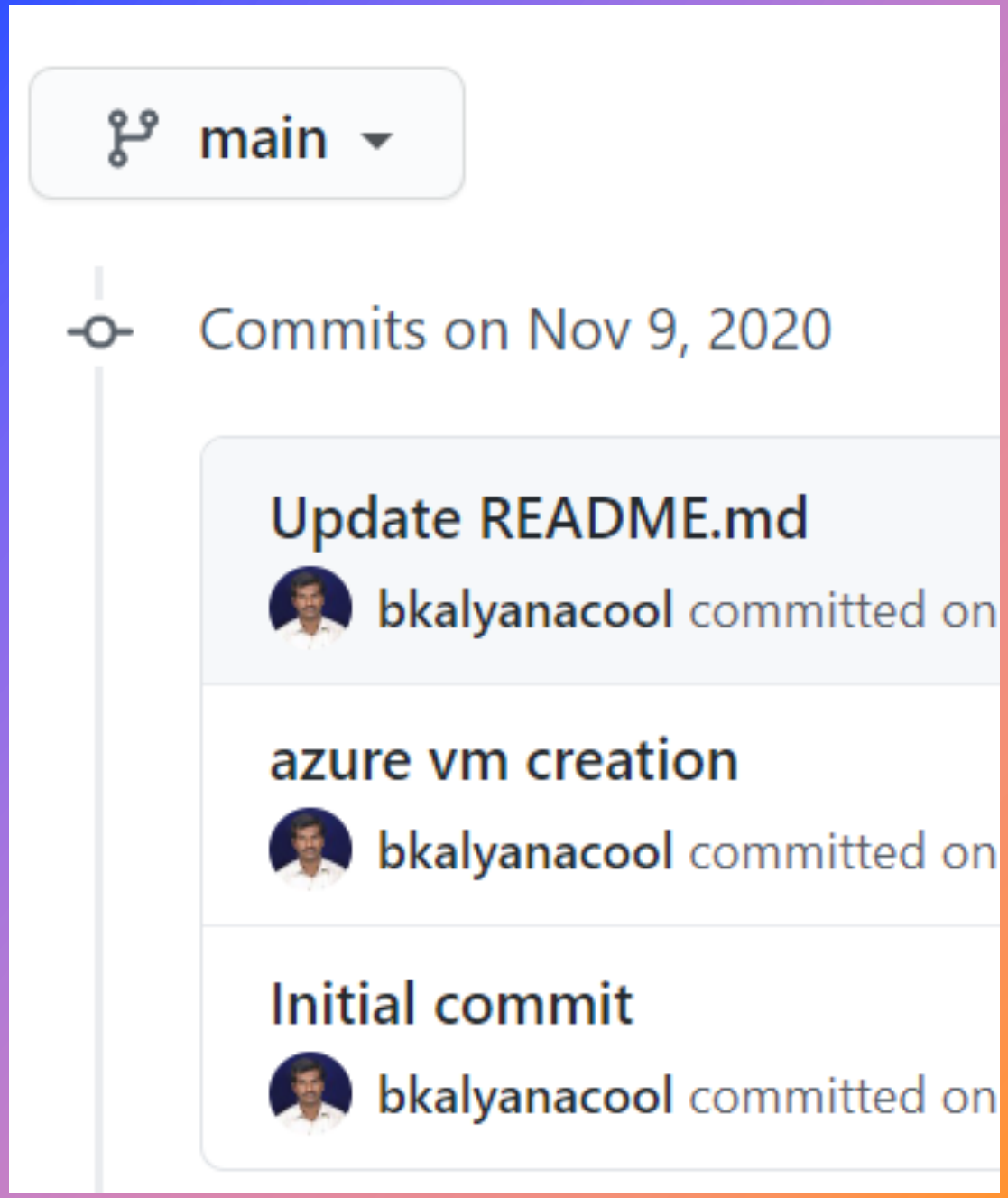
You can also decide whether you receive notifications via email, web & mobile, or both

To keep track of all of your notifications across different projects, use the GitHub Notifications dashboard

Branches

- Branches are the preferred way to create changes in the GitHub flow
- his model enables stability among critical branches, such as `main`, while allowing complete freedom for developers to commit any changes they need to meet their goals.
- Once the code from a branch is ready to become part of the `main` branch, it may be merged via pull request





Commits

- A **commit** is a change to one or more files on a branch. Every time a commit is created, it is assigned a unique ID and tracked, along with the time and contributor.
- This provides a clear audit trail for anyone reviewing the history of a file or linked item, such as an issue or pull request.

Pull Requests

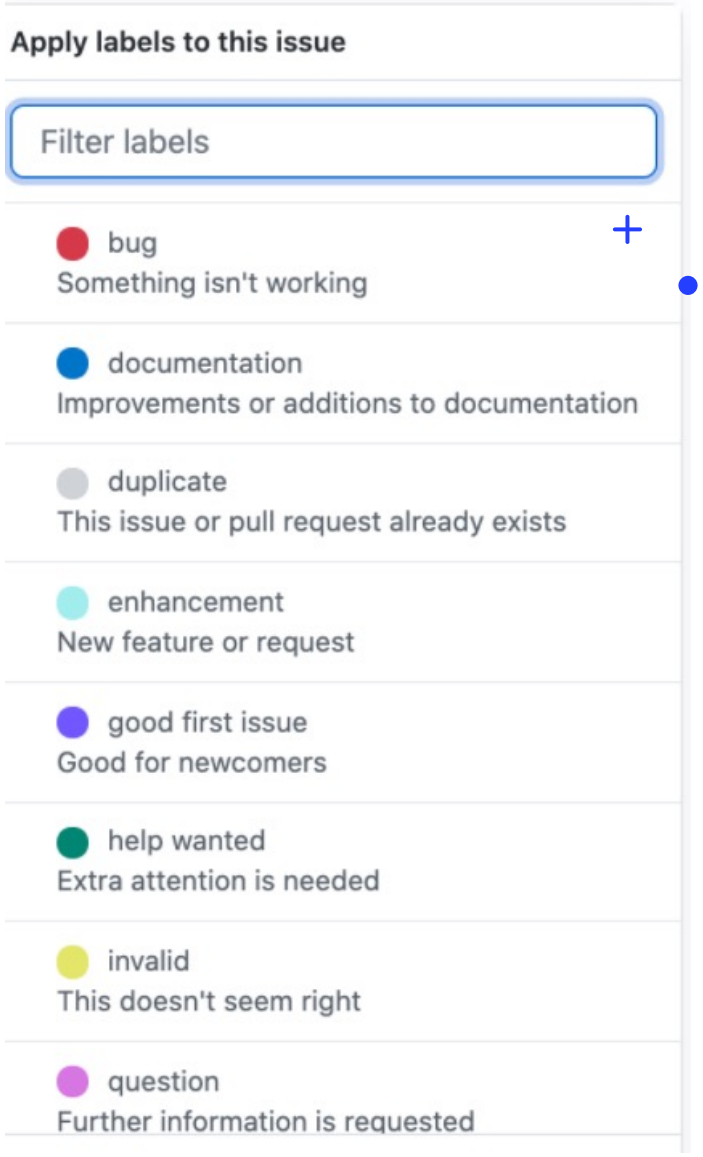
- A **pull request** is the mechanism used to signal that the commits from one branch are ready to be merged into another branch.
- one or more reviewers to verify the code and approve the merge and once merge is approved the code will be merged with the base branch.

Labels

- Provide a way to categorize and organize **issues** and **pull requests** in a repository
- As you create a GitHub repository several labels will automatically be added for you and new ones can also be created

Examples of Labels include:

- bug
- documentation
- duplicate
- help wanted
- enhancement
- question



Actions

- GitHub actions provide task automation and workflow functionality in a repository
- Actions can be used to streamline processes in your software development lifecycle and implement continuous integration and continuous deployment (CI/CD).

Workflows: Automated processes added to your repository.

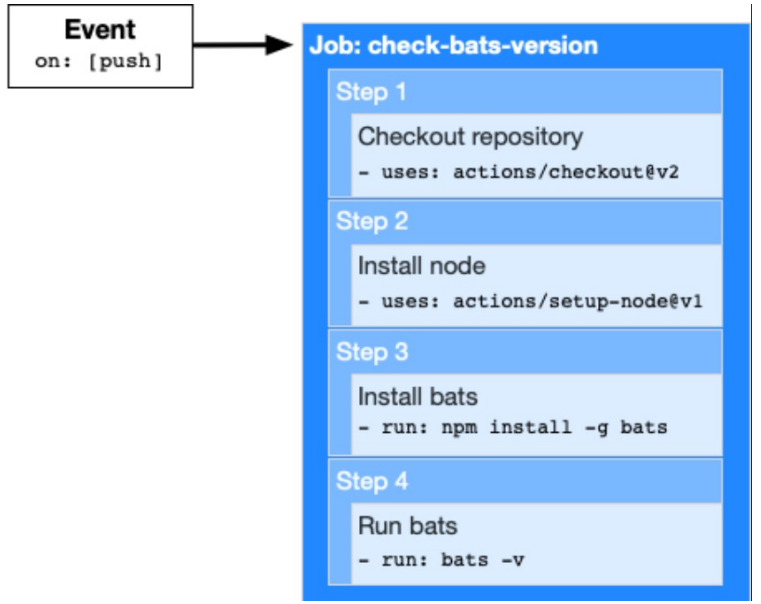
Events: An activity that triggers a workflow.

Jobs: A set of steps that execute on a runner.

Steps: A task that can run one or more commands (actions).

Actions: Standalone commands that can be combined into steps. Multiple steps can be combined to create a job.

Runners: Server that has the GitHub Actions runner application installed.



Cloning and forking

- Cloning a repository will make a copy of the repository and its history on your local machine
- **Forking** a repository makes a copy of the repository in your GitHub account. The parent repository is referred to as the **upstream** while your forked copy is referred to as the **origin**

Organizations

- Organizations are shared accounts where businesses and open-source projects can collaborate across many projects at once, with sophisticated security and administrative features
- You can use organizations to collaborate with an unlimited number of people across many projects at once, while managing access to your data and customizing settings

Repository Roles

- **Read**: Recommended for non-code contributors who want to view or discuss your project

- **Triage**: Recommended for contributors who need to proactively manage issues and pull requests without write access

- **Write**: Recommended for contributors who actively push to your project

- **Maintain**: Recommended for project managers who need to manage the repository without access to sensitive or destructive actions

- **Admin**: Recommended for people who need full access to the project, including sensitive and destructive actions like managing security or deleting a repository