

Exam Guide – Response Types

Cheat sheets, Practice Exams and Flash cards  www.exampro.co/ghf



There are 75 Questions

- 60 Scored Questions
- 15 Unscoered Questions

You can afford to get 15 scored questions wrong



There is no penalty for wrong questions

Format of Questions

- Multiple Choice
- Multiple Answer



Exam Guide – Duration

Cheat sheets, Practice Exams and Flash cards  www.exampro.co/ghf



Duration of 2 hours

You get ~1.6mins per question

Exam Time is: 120mins

Seat Time is: 150mins

Seat time refers to the amount of time that you should allocate for the exam.

It includes:

- Time to review instructions
- Show online proctor your workspace
- Read and accept NDA
- Complete the exam
- Provide feedback at the end.



Version Control Systems (VCS)

Cheat sheets, Practice Exams and Flash cards  www.exampro.co/ghf

Version Control Systems (VCS) are **designed to track changes or revisions to code.**



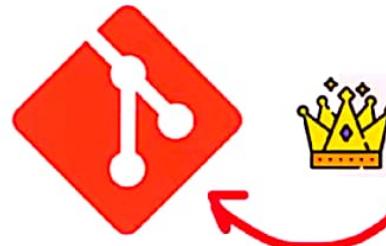
1990



2000



2005



The most popular VCS is **Git**.



Centralized VCS (DVCS)



Decentralized VCS (DVCS)

DVCS (and git specifically) became **very popular** for many reasons:

- Full local history and complete a copy of the repo locally
- Straightforward and efficient branching and merging
- Better performance, Improved fault tolerance, flexible workflows, work fully offline



Version Control normally represents revisions being represented as graph like structure. So you'll hear terms like **tree**, **trunk** and **branches**



Git

Cheat sheets, Practice Exams and Flash cards  www.exampro.co/ghf



Git is a distributed version control systems (DVCS) created by **Linus Torvald.**



creator of the linux kernel

Branches:	Description	Date	Author	Commit ID
origin/65-final-category	correct the town	11 Oct 2023 12:11	Andrew Brown	16796d58
main origin origin/HEAD	commit files (...)	11 Oct 2023 11:48	Andrew Brown	abf341eb
origin/63-payday-home	commit files	7 Oct 2023 14:01	Andrew Brown	a8b839a9
origin/terraform-cloud	yes!	30 Sep 2023 13:59	Andrew Brown	1256b515
please work!!!!		30 Sep 2023 13:54	Andrew Brown	e5baa1d5
lets go		30 Sep 2023 13:49	Andrew Brown	33b8ea39
comment out more		30 Sep 2023 13:42	Andrew Brown	a8e47468
comment out code make things work		30 Sep 2023 13:40	Andrew Brown	c98c7107
try and fix our issues.		30 Sep 2023 13:31	Andrew Brown	47b32b8f
wip		30 Sep 2023 13:23	Andrew Brown	ad732e57
2.6.0	#56 fix issues with provider, setup two diff...	29 Sep 2023 12:21	Andrew Brown	02fa4eb7
origin/56-multi-home-terraform-cloud	#56 fix issu...	29 Sep 2023 12:21	Andrew Brown	8e026527
2.5.0	#52 test our terra home on terratowns test ...	29 Sep 2023 10:07	Andrew Brown	89dfc0e6
origin/52-terratown-test	#52 test our terra home ...	29 Sep 2023 10:06	Andrew Brown	895e72c3
2.4.0	47 terratowns provider (#51)	28 Sep 2023 15:31	Andrew Brown	f0ef63f1
origin/47-terratowns-provider	Merge branch 'main'...	28 Sep 2023 15:31	Andrew Brown	d23bf312
\$47	finish implementation for terraform provider	28 Sep 2023 15:29	Andrew Brown	004a68fc
wip		28 Sep 2023 12:56	Andrew Brown	46f53c2e
2.3.0	47 terratowns provider (#50)	28 Sep 2023 11:52	Andrew Brown	5f9189ee
#47	setup skeleton for resource	28 Sep 2023 11:51	Andrew Brown	103d77d1
Merge branch 'main' into 47-terratowns-provider		27 Sep 2023 16:35	Andrew Brown	257ca11b
2.2.0	47 terratowns provider (#49)	27 Sep 2023 16:33	Andrew Brown	976bd7cb
Merge branch 'main' into 47-terratowns-provider		27 Sep 2023 16:33	Andrew Brown	ad0c0347
#47	implement terraform provider block in terraform	27 Sep 2023 16:31	Andrew Brown	72c519cb

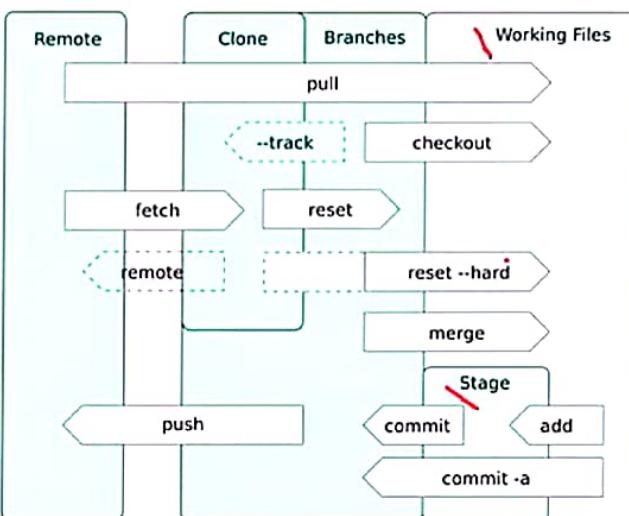
```
diff --git a/public/index.html b/public/index.html
@@ -6,6 +6,6 @@
 6   6      <title>Hello Terraformers</title>
 7   7      </head>
 8   8      <body>
 9 -  -      <h1>Hello Terraformers!!!!!!</h1>
 9 +  +      <h1>Hello there Terraformers!</h1>
10  10      </body>
11  11      </html>
```

Each change of your code (**git commit**) can be captured and tracked throughout the history of your project (**git tree**)



Common Git Terms

Cheat sheets, Practice Exams and Flash cards  www.exampro.co/ghf



- **Repository:** Represents the logical container holding the codebase
- **Commit:** Represents a change of data in the local repository.
- **Tree:** Represent the entire history of a repo
- **Remote:** A version of your project hosted elsewhere, used for exchanging commits.
- **Branches:** Divergent paths of development, allowing isolated changes.
 - **Main (formally known as master)** the most common name for the default branch
- **Clone:** Creates a complete local copy of a repository, including its history.
- **Checkout:** Switches between different branches or commits in your repo.
- **Pull:** Downloads changes from a remote repository and merges them into your branch.
- **Push:** Uploads your local repository changes to a remote repository.
- **Fetch:** Downloads data from a remote repo without integrating it into your work.
- **Reset:** Undoes local changes, with options to unstage or revert commits.
- **Merge:** Combines multiple commit histories into one.
- **Staging files:** Prepares and organizes changes for a commit.
 - **Commit:** Saves your changes as a snapshot in the local repository.
 - **Add:** Adds changes to the staging area for the next commit.



Version Control Services (VCS)

Cheat sheets, Practice Exams and Flash cards  www.exampro.co/ghf

Version Control Services (VCS) are **fully managed cloud services that hosts your version controlled repositories.**

These services often have additional functionality going beyond just being a remote host for your repos.

Git is the most popular and often the only choice for VCS. Often we call these services "**git providers**"



Github

Owned by Microsoft. The most popular VCS offering due to ease of use and being around the longest. Github is primarily where open-source projects are hosted and offer rich functionality such as Issues Tracking, Automation pipelines and a host of other features.



GitLab

GitLab was an emerging competitor to Github and at the time had unique features such as a CI/CD pipeline and improved security measures.



BitBucket

Owned by Atlassian. Originally hosted Mercurial, later adding Git and then sunsetting Mercurial. Tight integration with Atlassian's industry standard agile project management software **JIRA**. Larger organizations commonly use BitBucket because of JIRA adoption.



SourceForge

One of the oldest places to host your source code and was the first to offer free of charge to open-source projects.



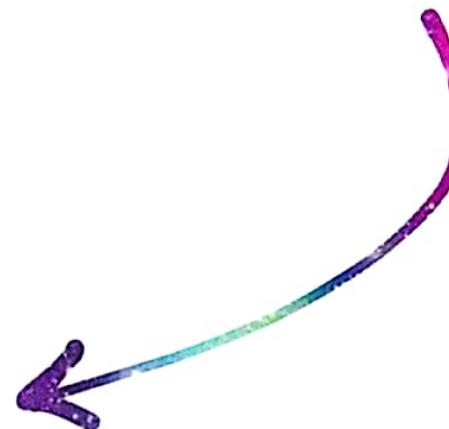
GitHub

Cheat sheets, Practice Exams and Flash cards  www.exampro.co/ghf



GitHub is a **Version Control Service** that initially offered hosting managed remote git repositories and has **expanded to provide other offerings** around hosted codebases.

- Git Repository Hosting
- Project Management Tools
- Issue Tracking
- Pull Requests and Code Review
- GitHub Pages and Wikis
- GitHub Actions
- GitHub Copilot
- GitHub Codespaces
- GitHub Marketplace
- GitHub Gists
- GitHub Discussions
- Collaboration Features (Organizations and Teams)
- API Access and Development Tools (Github Desktop, Github CLI)
- Security Features (eg. Autodetecting credentials in repos)
- Educational Resources and Course Automation (Github Classroom)



Fun Fact! GitHub was originally built using **Ruby on Rails**



GitHub Octocat

Cheat sheets, Practice Exams and Flash cards  www.exampro.co/ghf

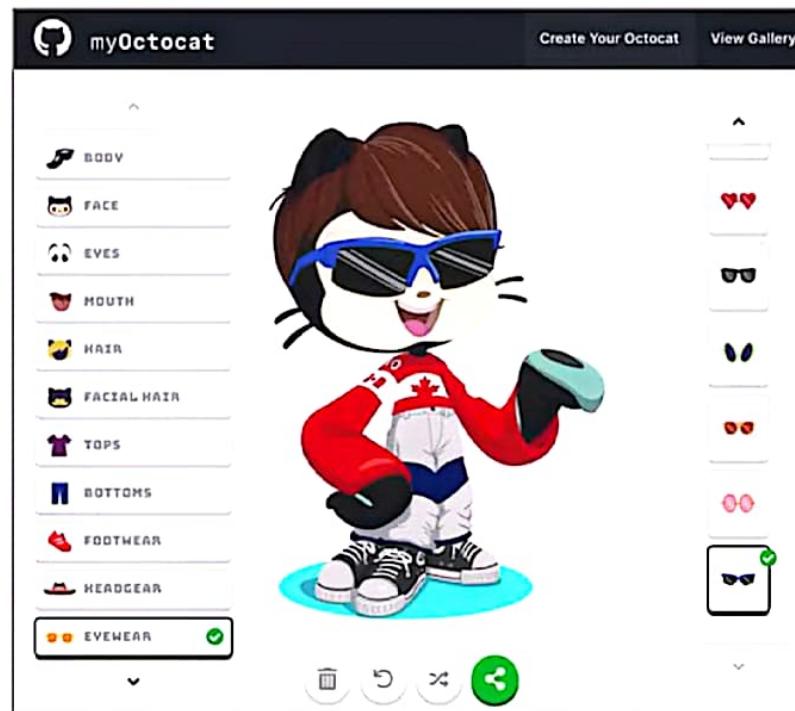
GitHub's Octocat is the platform's **official mascot**.

The character, a hybrid of an octopus and a cat, symbolizes GitHub's friendly and engaging nature in the world of software development and collaboration.



You can **create your own** Octocat

<https://myoctocat.com>



Git vs Github

Cheat sheets, Practice Exams and Flash cards  www.exampro.co/ghf

Git	GitHub
Nature	Distributed Version control system (DVCS)
Functionality	Manages source code history
Access	Local system installation
Scope	Local repository management
Collaboration	Local changes, requires manual sharing
Usage	Command-line interface
	Version Control as a Service (VCaaS)
	Provides cloud storage for Git repositories
	Accessed via web interface
	Online collaboration and repository hosting
	Integrated tools for collaboration (issues, PRs)
	Graphical interface and additional features



Github Repo

Cheat sheets, Practice Exams and Flash cards  www.exampro.co/ghf

A Github Repo is your git repo that you push upstream to GitHub

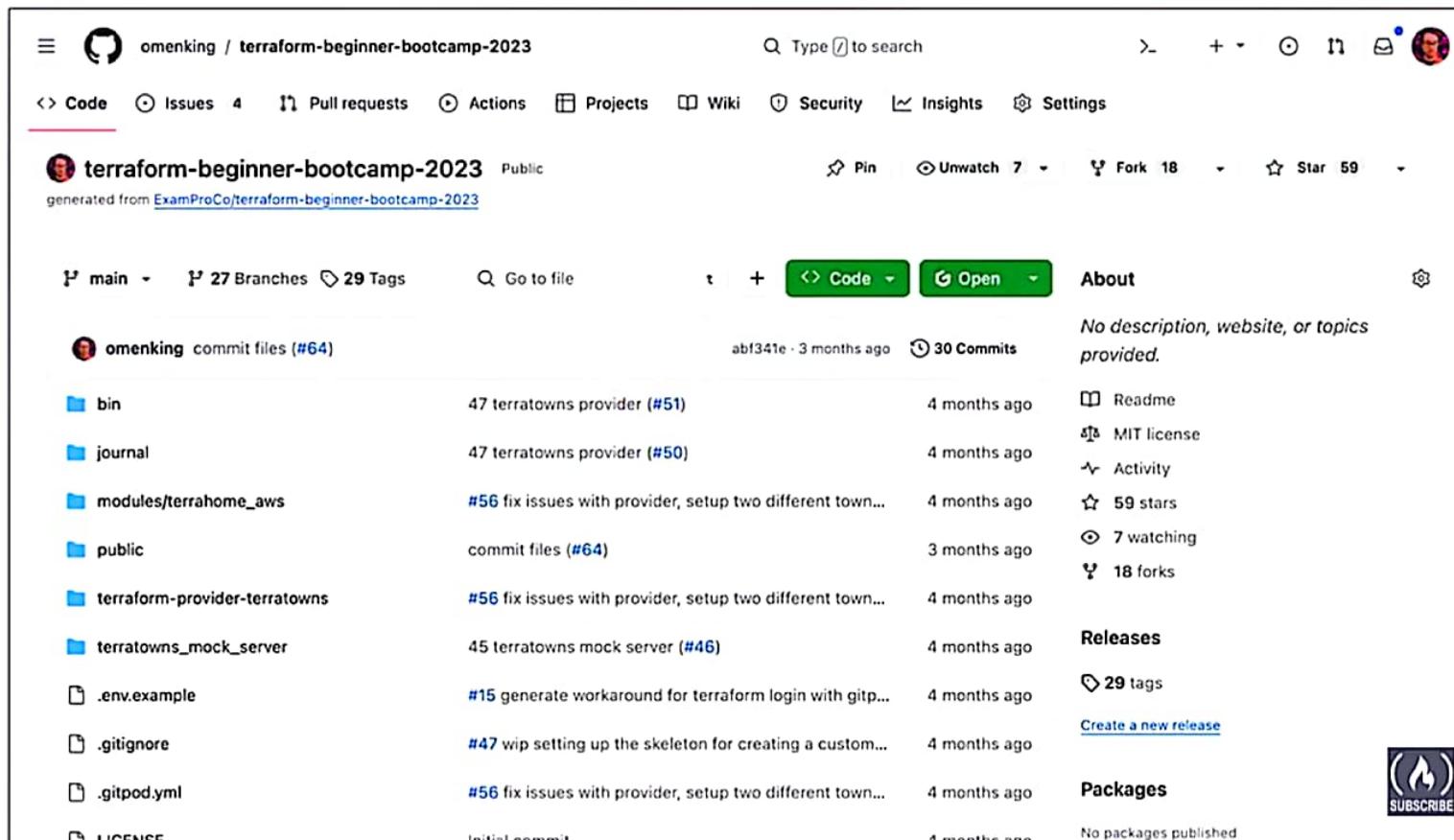
GitHub allows you access and manage your git repo with several functionality

From your Github repo page you can:

- View different branches
- View tags
- View commit history
- Explore repo's files
- View releases
- See codebase language breakdown
- View the top level markdown files
 - Readme.md
 - Licence.md

You can perform actions such as:

- Pinning
- Watching
- Forking
- Starring
- Cloning (downloading)



The screenshot shows a GitHub repository page for 'terraform-beginner-bootcamp-2023' by 'omenking'. The repository has 27 branches and 29 tags. The main branch has 30 commits. The repository is public and was generated from ExamProCo/terraform-beginner-bootcamp-2023. It includes sections for About, Releases, Tags, and Packages.

About: No description, website, or topics provided.

Releases: Create a new release

Tags: 29 tags

Packages: No packages published

Commit	Message	Date
abf341e · 3 months ago	30 Commits	
bin	47 terratowns provider (#51)	4 months ago
journal	47 terratowns provider (#50)	4 months ago
modules/terrahome_aws	#56 fix issues with provider, setup two different town...	4 months ago
public	commit files (#64)	3 months ago
terraform-provider-terratowns	#56 fix issues with provider, setup two different town...	4 months ago
terratowns_mock_server	45 terratowns mock server (#46)	4 months ago
.env.example	#15 generate workaround for terraform login with gitp...	4 months ago
.gitignore	#47 wip setting up the skeleton for creating a custom...	4 months ago
.gitpod.yml	#56 fix issues with provider, setup two different town...	4 months ago
LICENSE	Initial commit	4 months ago

Git Commit

Cheat sheets, Practice Exams and Flash cards  www.exampro.co/ghf

A Git Commit represent incremental changes to a codebase represented with a git tree (graph) at a specific time.



```
47 terratowns provider (#49) ▾
  omenking committed 4 months ago

#47 wip setting up the skeleton for creating a custom terraform provider (#48)
  omenking committed 4 months ago

45 terratowns mock server (#46) ▾
  omenking committed 4 months ago
  ⚡ Commits on Sep 18, 2023

39 add git graph to gitpod configuration (#41) ▾
  omenking committed 4 months ago

add the known extension to our gitpod configuration task file (#40)
  omenking committed 4 months ago
  ⚡ Commits on Sep 15, 2023

#35 upload assets using a for_each (#36)
  omenking committed 4 months ago

#31 invalidate cloudfront stack when content version changes (#34)
  omenking committed 4 months ago
  ⚡ Commits on Sep 14, 2023
```

A git commit contains:

- Additions, **modifications** and deletions of files
- Additions and deletions of file contents.
- Not the whole files themselves.

Each Commit has a SHA hash that acts as an ID

- Eg. 661a91ad3b66926c4591f9d3c73c087906945f3b

You can use this **Commit SHA** to *checkout* specific commits.



```
Showing 9 changed files with 47 additions and 5 deletions

Filter changed files
gitpod.yml
house_aws
page.tf
arcانan-box-art.jpg
screenshot.png
index.html
terraformfile-example
variables.tf
journal/week1.md

gitpod.yml
11 11 00 -11,6 +11,11 @@ tasks:
12 12 before:
13 13 source ./bin/set_tf_alias
source ./bin/install_aws_cli
14 • + name: http-server
15 • + before: 1
16 • + npm install --global http-server
17 • + command:
18 • + http-server
19
20 vscode:
21 extensions:

journal/week1.md
272 272 )
273 273 }
274 274 ...
275 - https://developer.hashicorp.com/terraform
275 + https://developer.hashicorp.com/terraform
276 +
277 + ## For Each Expressions
```



Git does not store the whole files in each commit but rather the state of changes. This greatly reduces the file size. To the developer the files will appear whole

Git Commit

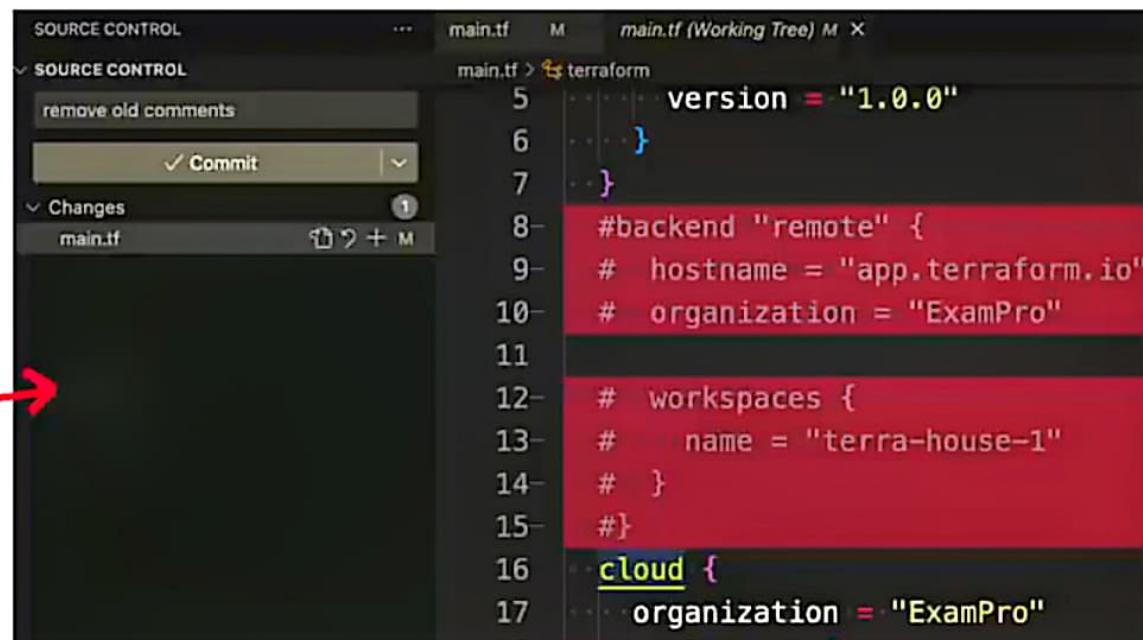
Cheat sheets, Practice Exams and Flash cards  www.exampro.co/ghf

Components of Git Commit:

- **Commit Hash:** A unique SHA-1 hash identifier for the commit.
- **Author Information:** The name and email of the person who made the commit.
- **Commit Message:** A description of what changes the commit contains.
- **Timestamp:** The date and time when the commit was made.
- **Parent Commit Hash(es):** The SHA-1 hash of the commit(s) this commit is based on.
- **Snapshot of Content:** A snapshot of the project at the time of the commit (not the actual files, but references to them).

Commit messages are often written in a tool as its more convenient to quickly add, remote files, audit changes

VS Code has that ability via the **Source Control** window



```
version = "1.0.0"
}
}

#backend "remote" {
#  hostname = "app.terraform.io"
#  organization = "ExamPro"

# workspaces {
#   name = "terra-house-1"
# }

#}

cloud {
  organization = "ExamPro"
```

You need add or remove files to be stage changes



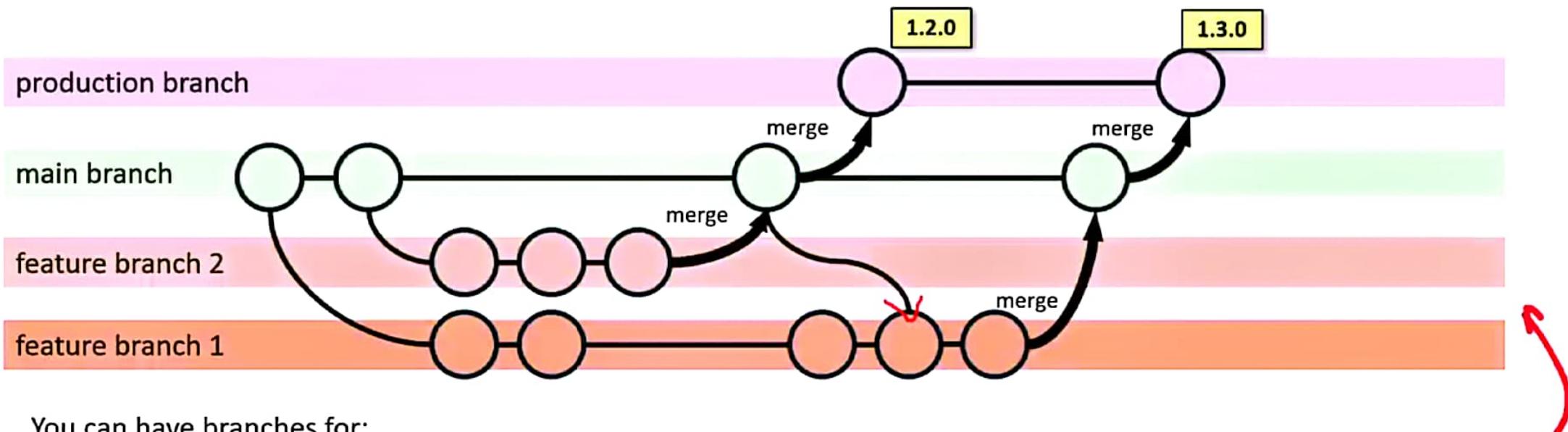
```
# Files to be staged  
git add my-file  
git add . # add all files  
git rm my-other file # remove a specific file  
  
# Commits staged changes with a message  
git commit -m "Commit message"  
  
# Automatically stages all tracked, modified files before the commit  
git commit -a -m "Commit message"  
  
#Modifies the most recent commit  
git commit --amend  
  
#Creates an empty commit, useful as a placeholder  
git commit -m "Initial commit" --allow-empty  
  
# Commits with a specified author.  
git commit -m "Message" --author="Author Name <email@example.com>"  
  
# checkout to a specific commit based on SHA hash  
git checkout 661a91ad3b66926c4591f9d3c73c087906945f3b
```

Git Branch

Cheat sheets, Practice Exams and Flash cards  www.exampro.co/ghf

A git branch is a divergence of the state of the repo.

You can *think* of branches as being copies of a point in time that have been modified to be different.



You can have branches for:

- Specific environments: eg. Staging, development, production
- Specific to developers: eg. andrew, bayko, cindy
- branches per feature: eg. payment-gateway-feature
- Branches per bug eg. hotfix-blank-hoempage

This type of workflow is very close to **GitHub Flow**

#Lists all local branches
`git branch`



#Creates a new branch
`git branch [branch-name]`

Checkout (switch to) a branch
`git checkout [branch-name]`

Create and checkout a branch
`git checkout -b [branch name]`

#Deletes a branch
`git branch -d [branch-name]:`

Renames a branch
`git branch -m [old-name] [new-name]`

Lists both remote and local branches
`git branch -a`

Git Remote

Cheat sheets, Practice Exams and Flash cards  www.exampro.co/ghf

A git **remote** represents **the reference to remote location** where a copy of the repository is hosted.

You can have multiple remote entries for your git repo.

- "origin" as a remote name almost always seen for a repo.
 - It indicates the central or golden repo everyone is working from and represents the source of truth.

The remote entries are stored in **.git/config**

```
[remote "origin"]
url = https://github.com/examproco/exampro.git
fetch = +refs/heads/*:refs/remotes/origin/*
[branch "production"]
remote = origin
merge = refs/heads/production
[branch "main"]
remote = origin
merge = refs/heads/main
```

Notice remote names can be **referenced**

```
# Lists all remote repositories along with their URLs.
git remote -v

git remote add [name] [URL]
git remote remove [name]
git remote rename [old-name] [new-name]

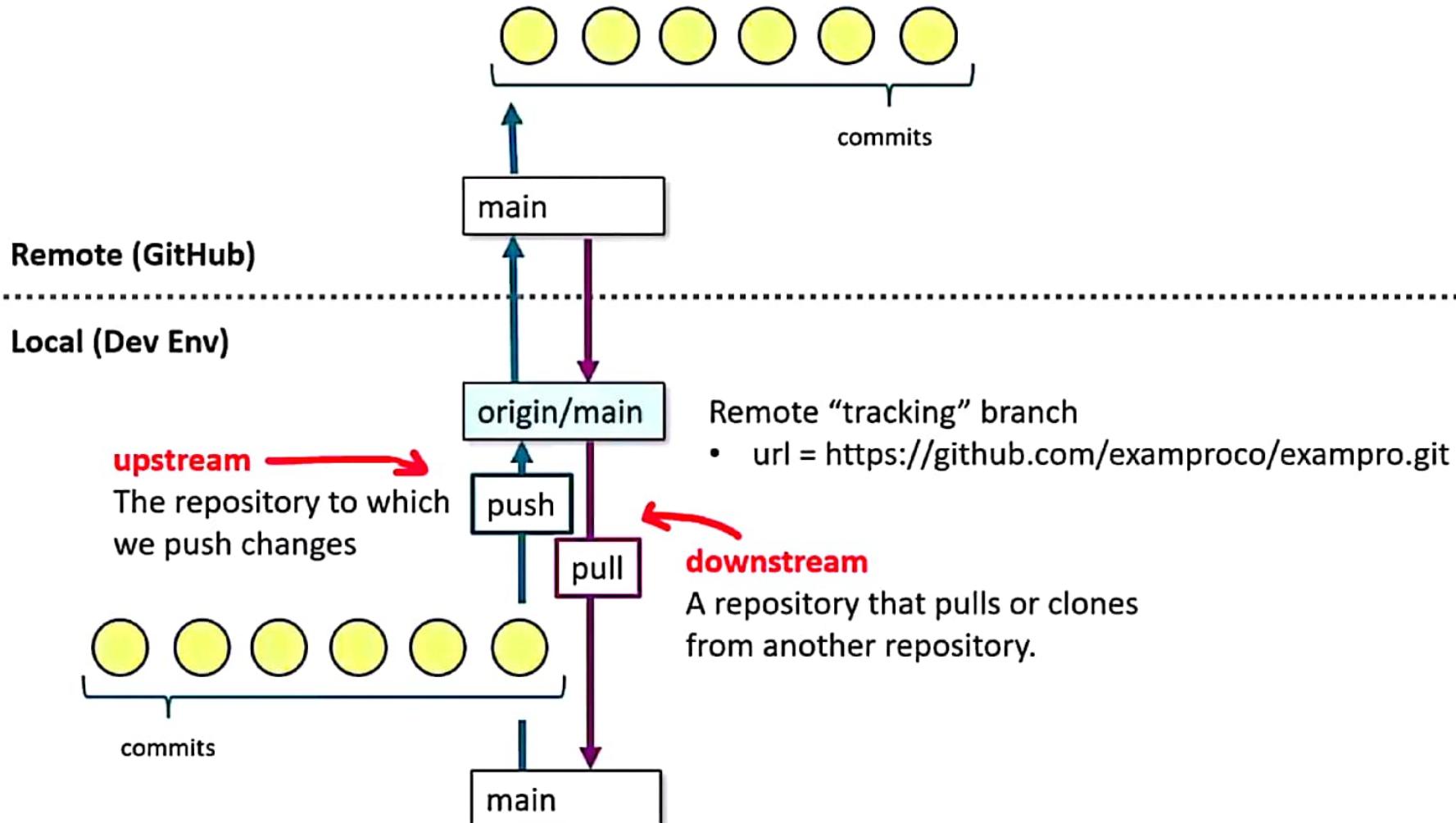
#Pushes a branch and its commits to the specific remote.
git push [remote-name] [branch]

#Pulls updates from a remote branch.
git pull [remote-name] [branch]

# Fetch updates without pulling.
git fetch [remote-name]
```

Git Remote – Upstream and Downstream

Cheat sheets, Practice Exams and Flash cards  www.exampro.co/ghf



GitHub Flow

Cheat sheets, Practice Exams and Flash cards  www.exampro.co/ghf

GitHub Flow is a **light-weight workflow** for multiple developers working on a single repository

Create a Branch: For each new task or feature, create a new branch off the main branch.

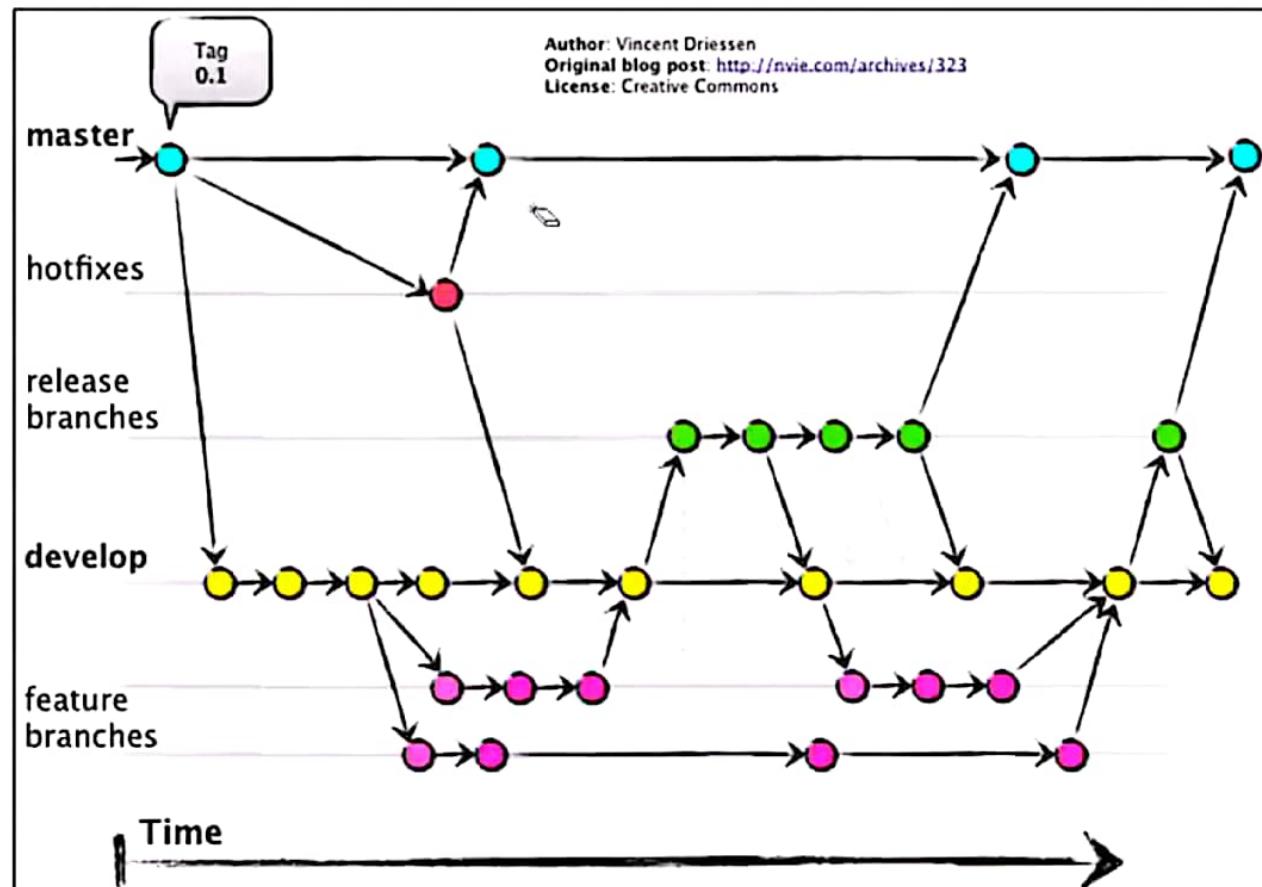
Add Commits: Make changes and commit them to your branch.

Open a Pull Request: Start a discussion about your commits, reviewing code in the pull request.

Discuss and Review: Share your pull request with teammates for feedback.

Deploy: Test your changes in a production environment.

Merge: Once your changes are verified, merge them into the main branch.



GitHub CLI

Cheat sheets, Practice Exams and Flash cards  www.exampro.co/ghf

The GitHub CLI is a **command line interface to interact with your GitHub Account.**

You can quickly perform common GitHub actions without leaving your developer environment

```
gh auth login  
gh repo create github-examples --public  
gh issue create --title "Issue title" --body "Issue body"  
gh pr review --comment -b "interesting"
```

The GitHub CLI can be installed on

Windows, Linux, and macOS

```
brew install gh
```

You can quickly install the CLI for GitHub Codespaces via Features:

```
"features": {  
    "ghcr.io/devcontainers/features/github-cli:1": {}  
}
```

Core commands

- gh auth
- gh browse
- gh codespace
- gh gist
- gh issue
- gh org
- gh pr
- gh project
- gh release
- gh repo

Additional commands

- gh alias
- gh api
- gh completion
- gh config
- gh extension
- gh gpg-key
- gh label
- gh ruleset
- gh search
- gh secret
- gh ssh-key
- gh status
- gh variable

GitHub Actions commands

- gh cache
- gh run
- gh workflow

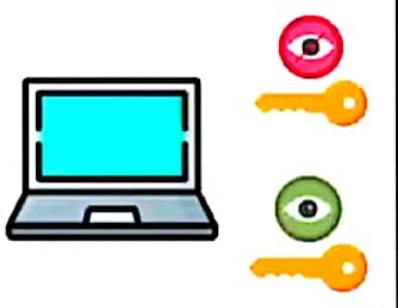
SSH Keys

Cheat sheets, Practice Exams and Flash cards  www.exampro.co/ghf

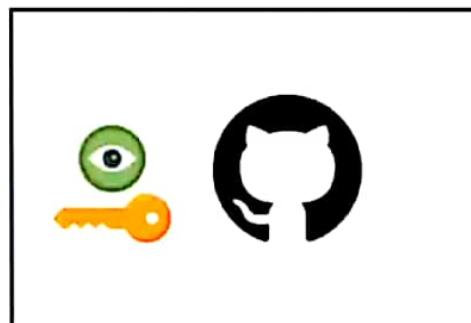
A common strategy for **authenticating to perform git operations** on your remote GitHub repo is using SSH keys

```
ssh-keygen -t rsa
```

SSH uses a public and private key to authorize remote access to servers.
ssh-keygen is a **well known command** to generate a public and private key



A copy of the public and private key reside on your local computer.



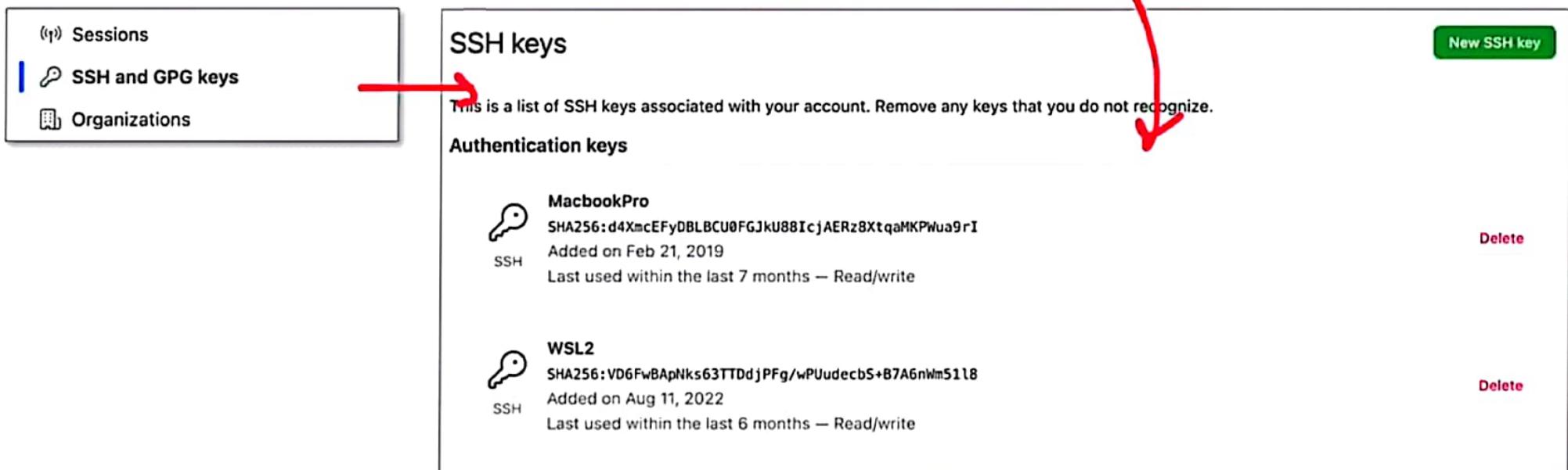
A copy of the public key is stored on GitHub.

- the server checks if you have the same public key
- If you do a challenge message is sent
 - Its an encrypted message with your public key
- Only a private key can decrypt the message
- Once decrypted the client signs the message also using the private key
- The server using the public key will verify the signature
- A connection is established.

SSH Keys

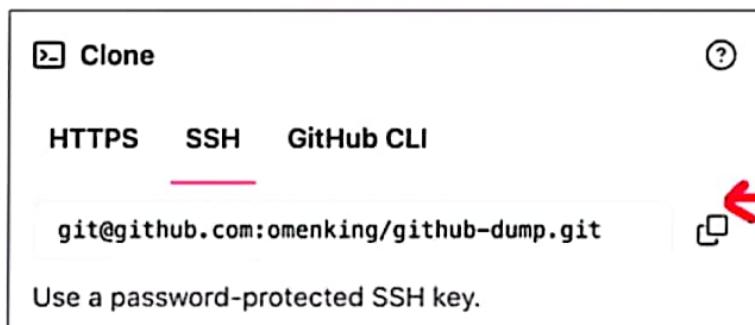
Cheat sheets, Practice Exams and Flash cards  www.exampro.co/ghf

Under your Account Settings of **SSH and GPG keys** you can **add your Public Keys** to your GitHub account



The screenshot shows the 'SSH keys' section of the GitHub account settings. On the left, there's a sidebar with 'Sessions', 'SSH and GPG keys' (which is highlighted with a red arrow), and 'Organizations'. The main area is titled 'SSH keys' and contains a note: 'This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.' Below this is a heading 'Authentication keys'.

Key	Name	SHA256	Added	Last Used	Action
	MacbookPro	SHA256:d4XmcEFyDBLBCU0FGJkU88IcjAERz8XtqaMKPWua9rI	Added on Feb 21, 2019	Last used within the last 7 months — Read/write	
	WSL2	SHA256:VD6FwBApNks63TTDdjPFg/wPUudecbS+B7A6nWm51l8	Added on Aug 11, 2022	Last used within the last 6 months — Read/write	



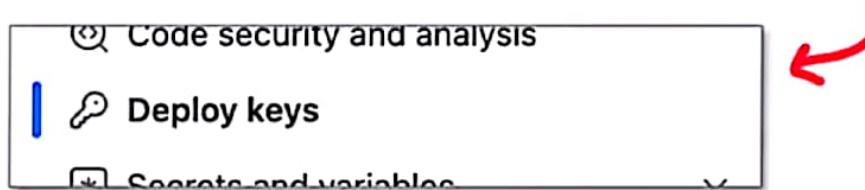
The screenshot shows the 'Clone' interface for a GitHub repository. It has options for 'HTTPS', 'SSH' (which is selected and highlighted with a red arrow), and 'GitHub CLI'. The URL 'git@github.com:omenking/github-dump.git' is displayed. A note at the bottom says 'Use a password-protected SSH key.'

Then you would clone your repo using the **ssh style address**.

Deploy Keys

Cheat sheets, Practice Exams and Flash cards  www.exampro.co/ghf

Deploy Keys allows you **attach public keys directly to a Git Repo**



The use-case for deploy keys are for:

- **Build servers or CI/CD third-party** services that need to clone the repo so they perform a build or a deploy
- **Single Repo Access** so instead of using a shared key pair for multiple repos you have a single key pair for a single Git Repo
- Avoid Using **Personal Account Access Tokens**

Personal Account Access Tokens

Cheat sheets, Practice Exams and Flash cards  www.exampro.co/ghf

Personal Access Tokens (PAT) are an **alternative to using passwords** for authentication to GitHub when using the GitHub API or the command line

GitHub no longer lets you use passwords when interacting with the API

GitHub has two types of PATS:

- **Classic tokens**
 - They are less secure and no longer recommended for use
 - Customer's with legacy systems may be still using classic token
- **Fine-grained personal access tokens**
 - granted specific permissions
 - Must have an expiry date
 - can only access specific repositories
 - can only access resources owned by a single user or organization

 Sponsorship log

<> Developer settings

Personal Access Tokens is found under **Developer Settings**



Personal Account Access Tokens

Cheat sheets, Practice Exams and Flash cards  www.exampro.co/ghf

When you **clone a git repo using HTTPS** you would use your personal access token to authorize the clone.

```
$ git clone https://github.com/USERNAME/REPO.git  
Username: YOUR_USERNAME  
Password: YOUR_PERSONAL_ACCESS_TOKEN
```

Instead of using gh login you could set the **GH_TOKEN**

```
export GH_TOKEN="MY_PERSONAL_ACCESS_TOKEN"  
gh pr create --title "The bug is fixed" --body "Everything works again"
```

When using **GitHub SDKs** you need to pass your PAT to authorize API calls

```
require 'octokit'  
octokit = Octokit::Client.new(access_token: 'YOUR-TOKEN')
```

GitHub APIs

Cheat sheets, Practice Exams and Flash cards  www.exampro.co/ghf

GitHub has two APIs **REST API** and **GraphQL API**

	REST API	GraphQL API
Design Philosophy	Resource-based. Different endpoints per resource	Single endpoint, query for precise data requests
HTTP Methods	Uses GET, POST, PUT, DELETE for operations	Mainly uses POST for all operations
Data Fetching	Multiple requests for related data	Single request for complex, related data
Over-fetching/Under-fetching	Can have over-fetching or under-fetching	Precise data fetching, no over/under-fetching
Performance	Less efficient for complex systems	More efficient for complex queries
Caching	Easier due to HTTP methods	More challenging due to POST method
Learning Curve	Generally easier to learn	Steeper but offers powerful features
Flexibility	Server-driven structure	Client-driven, highly flexible
Versioning	Often requires API versioning	Less need for versioning
Ecosystem	Mature, with extensive tools	Growing, with unique tools for queries

GitHub SDKs

Cheat sheets, Practice Exams and Flash cards  www.exampro.co/ghf



Octokit are the official SDKs to programmatically interact with the Git REST API

Terraform managed a git repo as Infrastructure as Code

```
resource "github_repository" "example" {
  name      = "example"
  description = "My awesome codebase"

  visibility = "public"

  template {
    owner          = "github"
    repository     = "terraform-template-module"
    include_all_branches = true
  }
}
```

GitHub official maintains SDKs for these languages

- JavaScript / TypeScript
- C# / .NET
- Ruby
- Terraform provider

Creating a GitHub repo using **Ruby**

```
require 'octokit'

client = Octokit::Client.new(
  access_token: ENV['GH_TOKEN']
)

repo_name = "my_new_repo"
options = {
  description: "This is my new repository",
  private: false,
  has_issues: true,
  has_projects: true,
  has_wiki: true
}
client.create_repository(repo_name, options)
```

Current Repository AWS-Examples

Current Branch main

Fetch origin Never fetched

Changes History refactor bash scripts, and provide etag example

No Branches to Compare

object locking Andrew Brown · 28 days ago

minor fix Andrew Brown · 28 days ago

metadata example Andrew Brown · 28 days ago

prefixes Andrew Brown · 28 days ago

additional checksums Andrew Brown · 28 days ago

refactor bash scripts, and provide e... Andrew Brown · 28 days ago

s3 versioning Andrew Brown · 28 days ago

simple s3 cdk example Andrew Brown · 2 months ago

terraform simple s3 example Andrew Brown · 2 months ago

add a gitignore Andrew Brown · 2 months ago

refactor bash scripts, and provide etag example · Andrew Brown · be3a698 · +76 -21

7 changed files .gitpod.yml

			.gitpod.yml
4	4	+	@@ -4,12 +4,7 @@ tasks:
5	5	+	npm i -g aws-cdk
6	6	+	- name: terraform
7	-	+	before:
8	-	+	sudo apt-get update && sudo apt-get install -y
		+	gnupg software-properties-common
		+	wget -O- https://apt.releases.hashicorp.com/gpg
		+	gpg --dearmor sudo tee /usr/share/keyrings/hashicorp-archive-keyring.gpg
		+	gpg --no-default-keyring --keyring /usr/share/k
		+	eyrings/hashicorp-archive-keyring.gpg --fingerprint
		+	echo "deb [signed-by=/usr/share/keyrings/hashic
		+	orp-archive-keyring.gpg] https://apt.releases.hashic
		+	rp.com \${lsb_release -cs} main" sudo tee /etc/apt/s
		+	ources.list.d/hashicorp.list
11	-	+	sudo apt update
12	-	+	sudo apt-get install terraform
13	8	+	./bin/terraform_cli_install.sh
14	9	+	# random utilities
15	10	+	- name: utils
		+	before:
19	14	+	@@ -19,23 +14,10 @@ tasks:
		+	# https://docs.aws.amazon.com/powershell/latest/re
		+	ference/

Github Desktop vs Github

Cheat sheets, Practice Exams and Flash cards  www.exampro.co/ghf

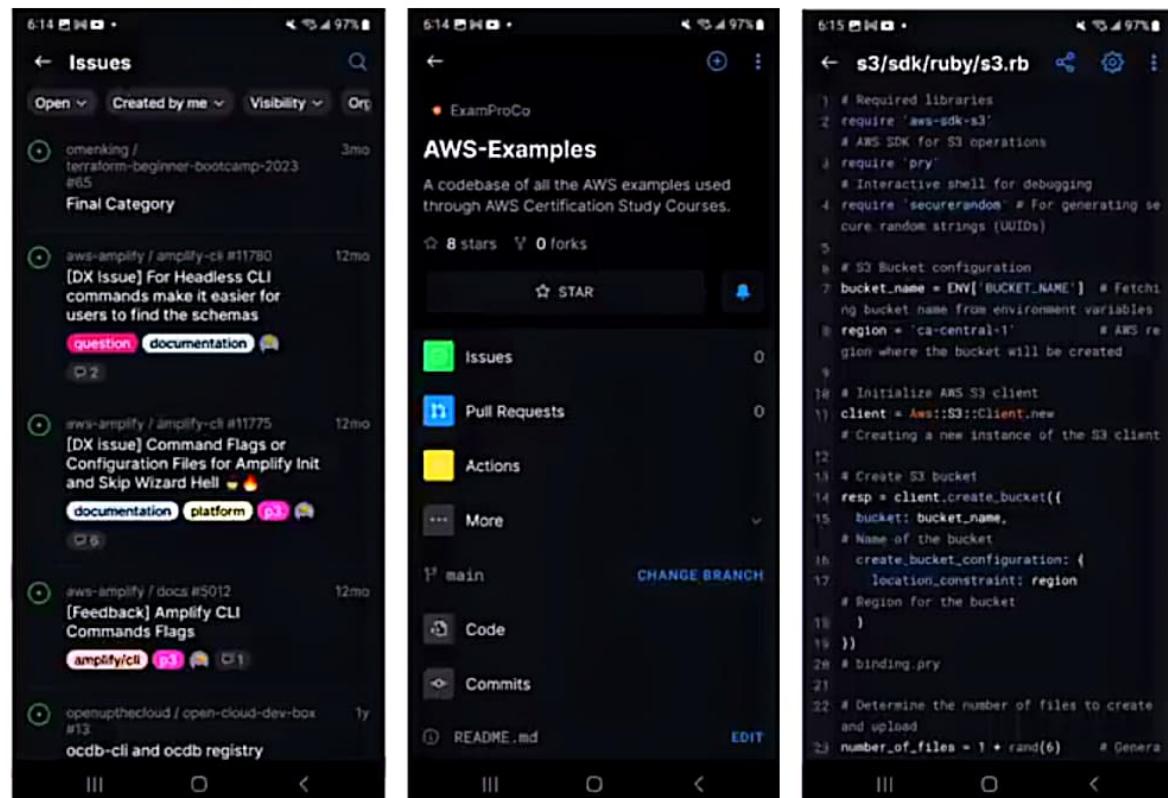
Github.com		Github Desktop
Interface	Web-based interface accessed via browser	Standalone desktop application
Accessibility	Accessible anywhere with internet	Accessible only on installed computer
Repo Management	Directly manage repositories online	Manage local copies of repositories
Basic Git Ops.	Perform operations like fork, star, watch	Simplified interface for commit, pull, push
Collaboration	Direct collaboration tools like issues, PRs	Focuses on local repo management
Advanced Git Features	Supports advanced features via commands	Simplified experience for basic features
Integration with Tools	Integrates with various online CI/CD tools	Limited to local Git tools

GitHub Mobile

Cheat sheets, Practice Exams and Flash cards  www.exampro.co/ghf

GitHub Mobile is a mobile application you can install on iOS or Android phones to
perform read-only or basic GitHub repo management tasks

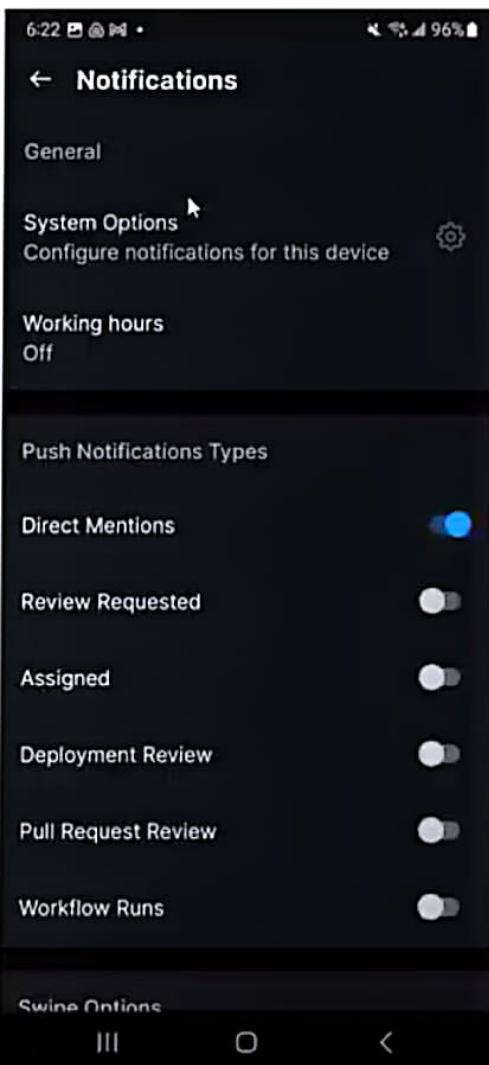
- Manage, triage, and clear notifications
- Read, review, and collaborate on issues and pull requests
- Edit files in pull requests
- Search for, browse, and interact with users, repositories, and organizations
- Receive a push notification when someone mentions your username
- Search through code in a specific repository
- **Secure your GitHub.com account with two-factor authentication**
- Verify your sign in attempts on unrecognized devices



Github Mobile can be used for UFA, which is a convenient way over other methods

GitHub Mobile Notifications

Cheat sheets, Practice Exams and Flash cards  www.exampro.co/ghf



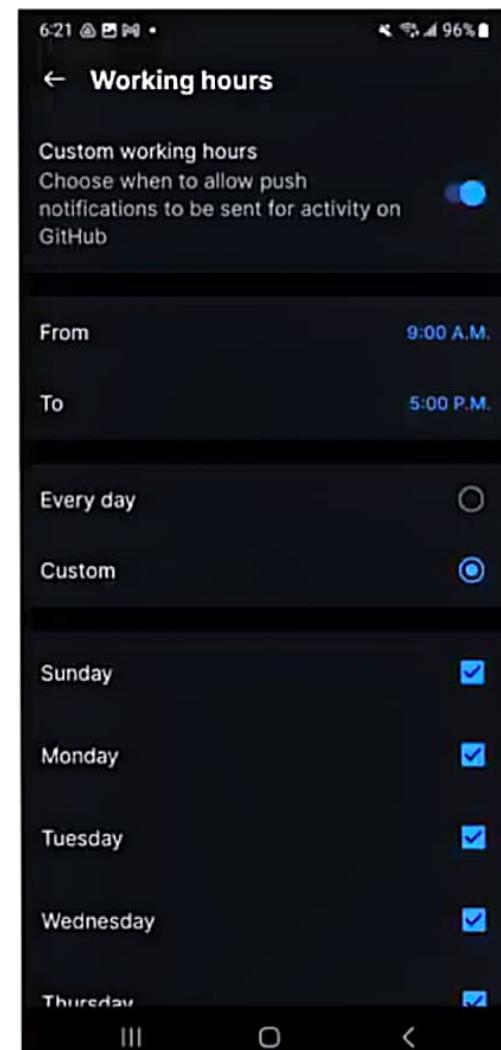
You set what **Push Notifications**

you want to receive:

- Direct Mentions
- Review Requested
- Assigned
- Deployment Review
- Pull Request Review
- Workflow Runs

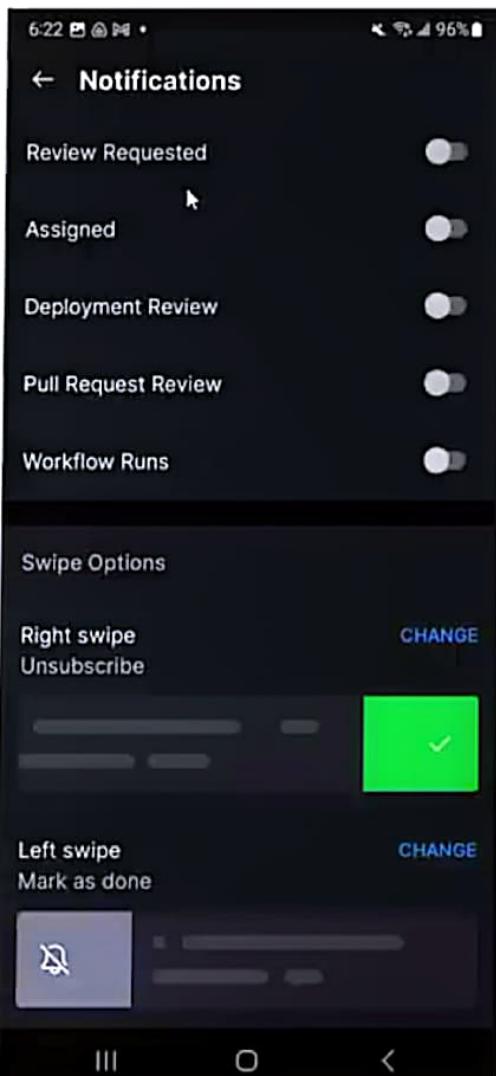


You set **Working Hours** to only receive push notifications for specific periods of time.



GitHub Mobile Notifications

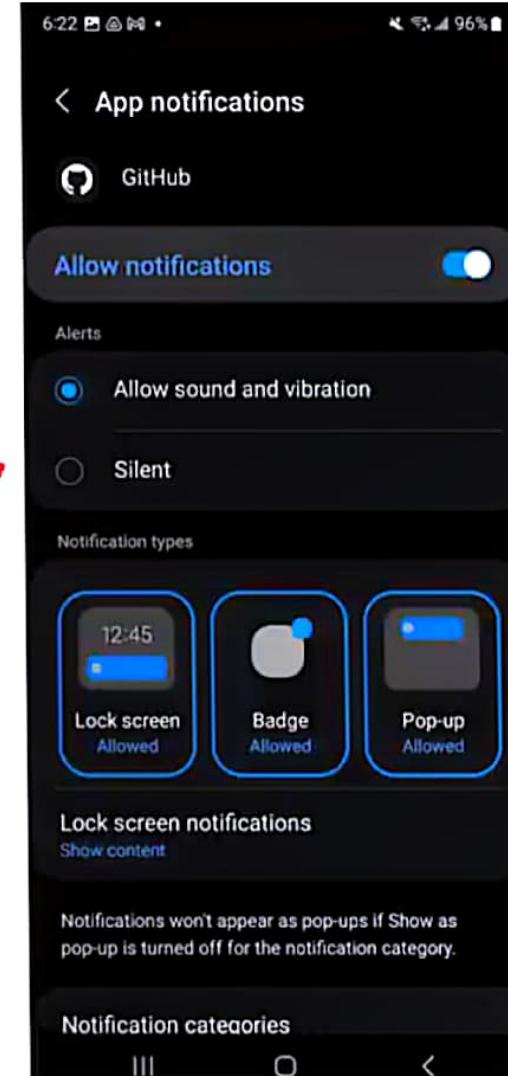
Cheat sheets, Practice Exams and Flash cards  www.exampro.co/ghf



You can set **swipe options** on notifications to perform:

- Save
- Mark as read
- Mark as done
- Unsubscribe

You can change app notifications style



Types of GitHub Accounts

Cheat sheets, Practice Exams and Flash cards  www.exampro.co/ghf

There are three types of GitHub Accounts: **Personal, Organizational, Enterprise**



Personal Accounts:

These are individual accounts with a username and profile.

They can own resources like repositories and projects, and actions taken are attributed to the personal account.

They can use either GitHub Free or GitHub Pro.



Organization Accounts:

Shared accounts where multiple people collaborate on projects.

They can also own resources like repositories but are managed through individual personal accounts.

Organizations offer different roles with varying levels of access and come with security features.



Enterprise Accounts

Part of GitHub Enterprise Cloud and Server, these accounts allow for central management of multiple organizations.

They're geared towards larger setups needing centralized policy and billing management.

GitHub Personal

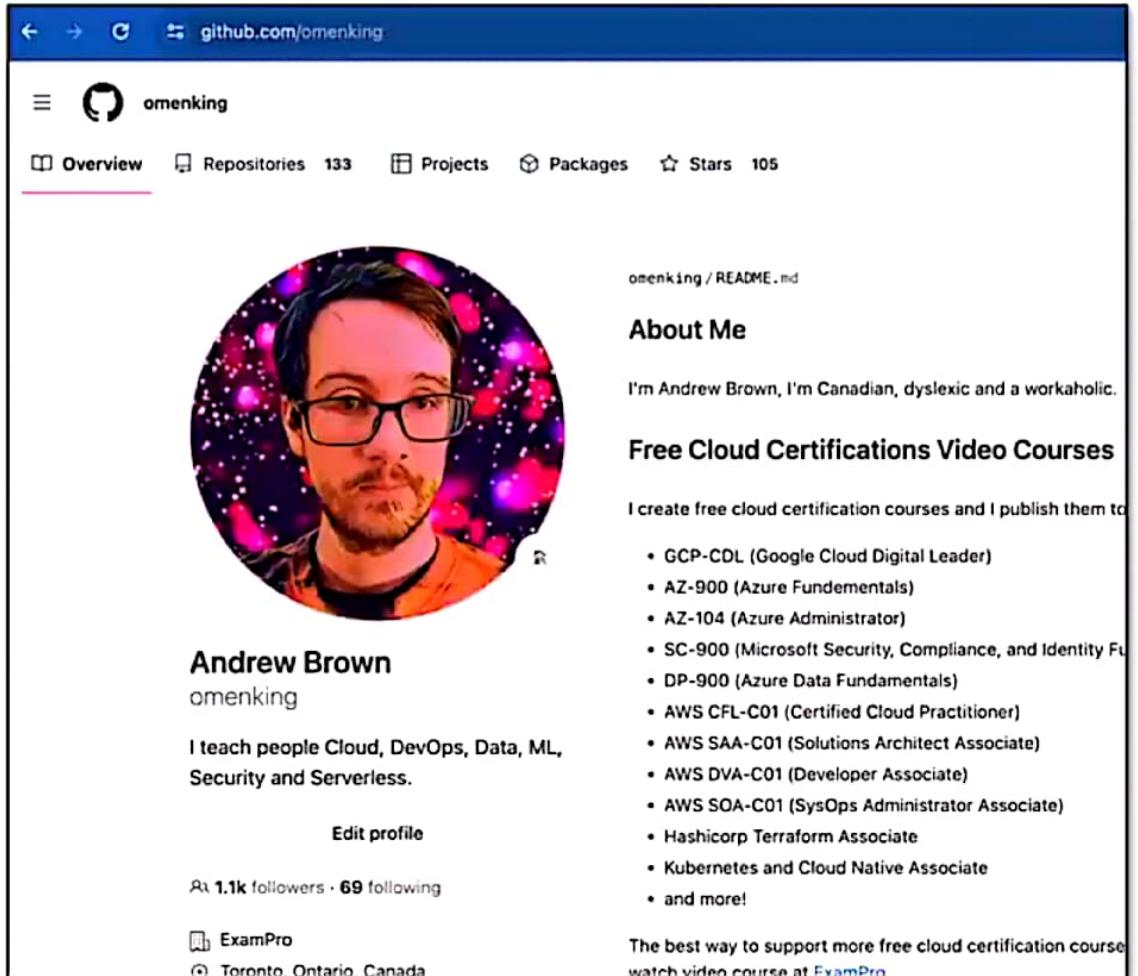
Cheat sheets, Practice Exams and Flash cards  www.exampro.co/ghf

With a GitHub Personal account you can have a **public profile** and also host your own public website.

Personal accounts have their own usernames:
Eg. omenking

Repo urls for personal accounts look like:

https://github.com/omenking/hello



The screenshot shows a GitHub profile page for the user 'omenking'. At the top, there's a navigation bar with links for Overview, Repositories (133), Projects, Packages, and Stars (105). Below the navigation is a large circular profile picture of a man with glasses and a beard, set against a colorful, star-filled background. The user's name, 'omenking', is displayed above the picture. To the right of the profile picture, there's a link to 'omenking / README.md'. Below the profile picture, the user's name is listed again as 'Andrew Brown' and 'omenking'. A bio section states: 'I teach people Cloud, DevOps, Data, ML, Security and Serverless.' There are 'Edit profile' and 'Edit bio' buttons below this. Further down, it shows '1.1k followers' and '69 following'. At the bottom left, there are links for 'ExamPro' and 'Toronto, Ontario, Canada'. On the right side, there's a sidebar with the heading 'About Me' containing the bio text. Below that is a section titled 'Free Cloud Certifications Video Courses' listing various certifications like GCP-CDL, AZ-900, and SC-900. At the very bottom right, there's a note: 'The best way to support more free cloud certification course watch video course at ExamPro'.

GitHub Organization

Cheat sheets, Practice Exams and Flash cards  www.exampro.co/ghf



With a GitHub Organization account can have its own **public profile** page.

It can show who belongs within the organization.

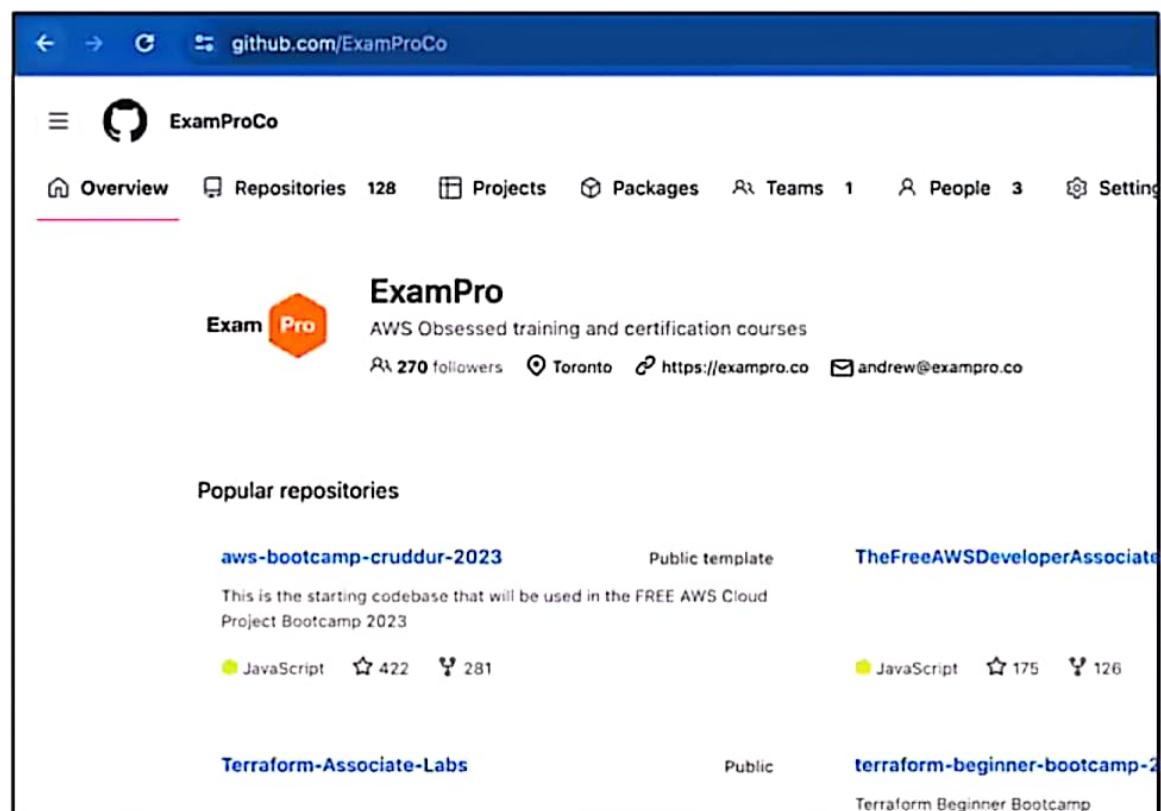
You can manage people into teams with different levels of access

Organizations have their own usernames.

Eg. ExamPro

To access a Organization repo would look like:

`https://github.com/ExamPro/AWS-Examples`



The screenshot shows the GitHub organization profile for 'ExamProCo'. The URL in the browser bar is 'github.com/ExamProCo'. The main header shows the organization's name 'ExamProCo' with a profile picture. Below the header, there are navigation links for Overview, Repositories (128), Projects, Packages, Teams (1), People (3), and Settings. The 'Overview' tab is selected, indicated by a pink underline. In the center, there is a section for the organization 'ExamPro' with a logo, a brief description 'AWS Obsessed training and certification courses', and stats: 270 followers, located in Toronto, with links to the website and email. Below this, there is a 'Popular repositories' section listing three repos: 'aws-bootcamp-cruddur-2023', 'Terraform-Associate-Labs', and 'terraform-beginner-bootcamp-2'. Each repo has a brief description, language (JavaScript), star count (422, 175), and fork count (281, 126).

GitHub Free vs Pro

Cheat sheets, Practice Exams and Flash cards  www.exampro.co/ghf

GitHub Free

- GitHub Community Support
- Dependabot alerts
- Deployment protection rules for public repositories
- Two-factor authentication enforcement
- 500 MB GitHub Packages storage
- 120 GitHub Codespaces core hours per month
- 15 GB GitHub Codespaces storage per month
- GitHub Actions features:
 - 2,000 minutes per month
 - Deployment protection rules for public repositories

GitHub Pro

- **Everything from GitHub FREE** and....
- GitHub Support via email
- 3,000 GitHub Actions minutes per month
- 2 GB GitHub Packages storage
- 180 GitHub Codespaces core hours per month
- 20 GB GitHub Codespaces storage per month
- Advanced tools and insights in private repositories:
 - Required pull request reviewers
 - Multiple pull request reviewers
 - Protected branches
 - Code owners
 - Auto-linked references
 - GitHub Pages
- Wikis
- Repository insights graphs

GitHub Organizations Plans

Cheat sheets, Practice Exams and Flash cards  www.exampro.co/ghf

Github Organizations has three plans: **Free, Teams and *Enterprise**

FREE

- **Everything from GitHub FREE** and....
- Team access controls for managing groups
- 2,000 GitHub Actions minutes per month
- 500 MB GitHub Packages storage

Teams (continued...)

- Draft pull requests
- Team pull request reviewers
- Protected branches
- Code owners
- Scheduled reminders
- GitHub Pages
- Wikis
- Repository insights graphs
- The option to enable or disable GitHub Codespaces

Teams

- **Everything from GitHub Organizations FREE** and....
- GitHub Support via email
- 3,000 GitHub Actions minutes per month
- 2 GB GitHub Packages storage
- Advanced tools and insights in private repositories
- Required pull request reviewers
- Multiple pull request reviewers

GitHub Enterprises Deployment Options

Cheat sheets, Practice Exams and Flash cards  www.exampro.co/ghf

GitHub Enterprise includes two deployment options

- GitHub Enterprise Cloud (hosted on GitHub.com)
- GitHub Enterprise Server (self-hosted)

GitHub Enterprise (both Cloud and Server)

- **Everything from GitHub Organizations Team**
- GitHub Enterprise Support
- Additional security, compliance, and deployment controls
- Authentication with SAML single sign-on
- Access provisioning with SAML or SCIM
- Deployment protection rules with GitHub Actions for private or internal repositories
- GitHub Connect
- option to purchase GitHub Advanced Security

GitHub Enterprise Cloud (specific features)

- 50,000 GitHub Actions minutes per month
- 50 GB GitHub Packages storage
- A service level agreement for 99.9% monthly uptime
- option to centrally manage policy and billing for multiple GitHub.com organizations with an enterprise account
- option to provision and manage the user accounts for your developers, by using Enterprise Managed Users

Markdown

Cheat sheets, Practice Exams and Flash cards  www.exampro.co/ghf

What is a Markup language?

A Markup language is a way of formatting and presenting text data in a different format.
A common use-case for a markup language is to present data in **HTML**.



Markdown is markup language that provides a shorthand syntax to format information into HTML.
Markdown is popular due to its easy syntax, and being readable in its raw format.

Markdown files extensions are **.md** and **.markdown**

Raw Markdown	HTML	HTML Preview
<pre># Sample Markdown This is some basic sample markdown. ## Second Heading • Unordered lists, and: 1. One 1. Two 1. Three • More > Blockquote And **bold**, *italics*, and even *italics a later **bold***. Even ~~strikethrough~~. [A (https://markdowntohtml.com) to somewhere.</pre>	<pre><h1 id="sample-markdown">Sample Markdown</h1> <p>This is some basic sample markdown.</p> <h2 id="second-heading">Second Heading</h2> Unordered lists, and: One Two Three More <blockquote> <p>Blockquote</p> </blockquote> <p>And bold, italics italics and later bold. Even</pre>	<p>Sample Markdown</p> <p>This is some basic sample markdown.</p> <p>Second Heading</p> <ul style="list-style-type: none">• Unordered lists, and:<ol style="list-style-type: none">1. One2. Two3. Three• More <p>Blockquote</p> <p>And bold, <i>italics</i>, and even <i>italics and later bold</i>. Even strikethrough. A link to somewhere.</p>

Markdown Basic Syntax

Cheat sheets, Practice Exams and Flash cards  www.exampro.co/ghf

All markdown parsers supports the following markdown syntax

Headings

H1
H2
H3

Text Formatting

italicized text
bold text

Blockquote

> blockquote

Unordered List

- First item
- Second item
- Third item

Ordered List

1. First item
1. Second item
1. Third item

the numbers will change to be the correct numbers

Link

[title](https://www.example.com)

Horizontal Rule

Image

![alt text](image.jpg)

Code Inline

`puts "hello world"'

*codeblocks use **backticks** and not single quotes.
Backticks key is above the tab key on the keyboard*



Markdown Extended Syntax

Cheat sheets, Practice Exams and Flash cards  www.exampro.co/ghf

Markdown parsers may support the following extended markdown syntax

Table

Syntax	Description
----- -----	
Header Title	
Paragraph Text	

Fenced Code Blocks

```
```rb
def hello_world
 puts "hello world"
end
````
```

You can sometimes have syntax highlighting for codeblocks by telling the codeblock the language being used.

Heading ID

```
### My Great Heading {#custom-id}
```

Definition List

term
: definition

Strikethrough

~~The world is flat.~~

Footnote

Here's a sentence with a footnote. [^1]

[^1]: This is the footnote.

Task List

- [x] Write the press release
- [] Update the website
- [] Contact the media

Emoji

That is so funny! :joy:

Highlight

I need to highlight these ==very important words==.

Subscript

H²O

Superscript

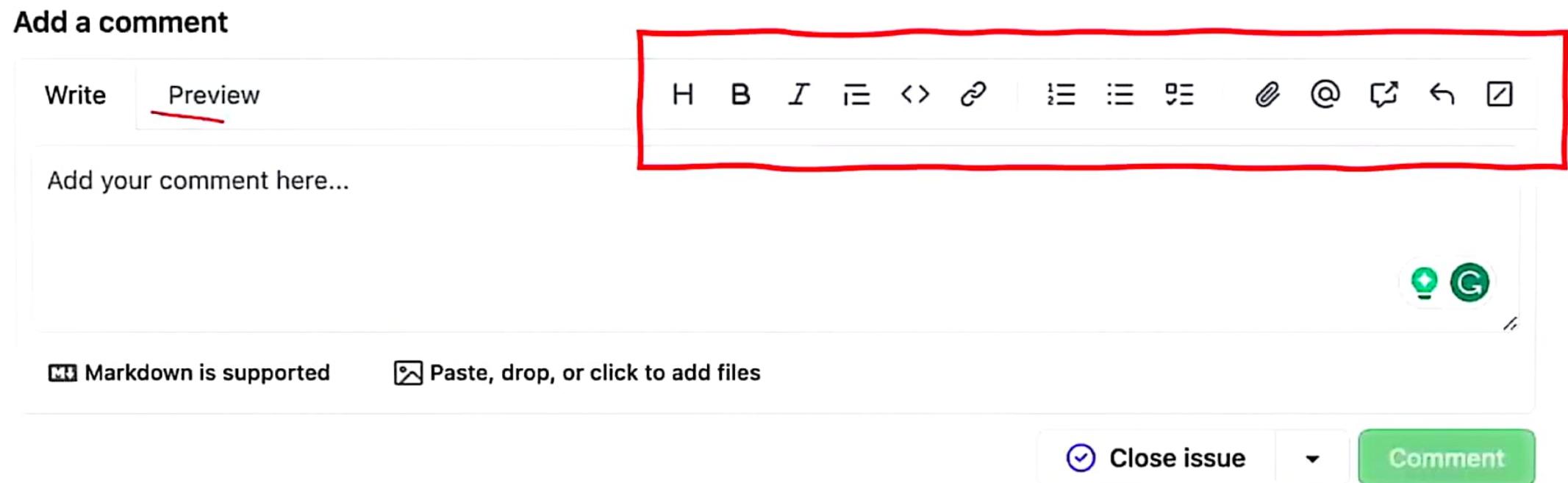
X²A

Text Formatting Toolbar

4:39:41 / 9:54:07

Cheat sheets, Practice Exams and Flash cards  www.exampro.co/ghf

Every comment field on GitHub contains a **text formatting toolbar**, which allows you to format your text without learning Github-flavoured Markdown syntax.

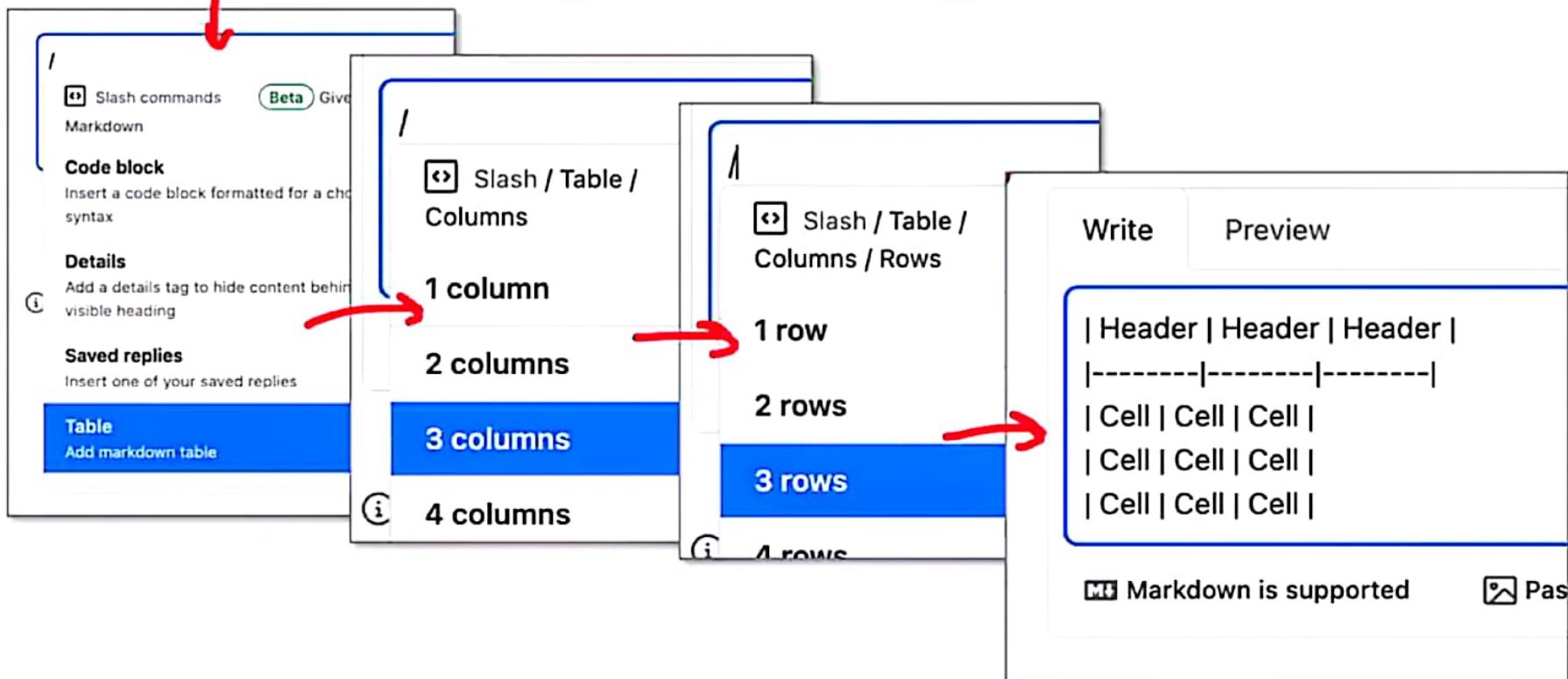


Slash Commands

Cheat sheets, Practice Exams and Flash cards  www.exampro.co/ghf

Slash Commands (within comment boxes) on GitHub.com provides convenience features such as formatting markdown.

Example of creating a markdown table using the slash command



Github Flavoured Markdown (gfm)

Cheat sheets, Practice Exams and Flash cards  www.exampro.co/ghf

Github Flavoured Markdown (GFM) is the **dialect of Markdown** that is currently **supported for user content on GitHub.com and GitHub Enterprise**

GFM provides powerful functionality:

- Collapsible sections
- Embed mathematical expressions
- Embed diagrams
- Relative paths or linking to files in the same repo
- Tasks lists can be converted into Issues
- Extended formatting for tables
- Shorthands to autolink to issues, pull requests and repos
- Render code snippets from other codesbase via linking
- And more...

The detailed **GFM spec** is located here:

 <https://github.github.com/gfm/>

Github Flavoured Markdown (gfm)

Cheat sheets, Practice Exams and Flash cards  www.exampro.co/ghf

Supports embedded syntax within tables

| Command | Description |
|--------------|---|
| --- | --- |
| `git status` | List all <i>*new or modified*</i> files |
| `git diff` | Show file differences that haven't been staged |

Supports advanced cell alignment

| Left-aligned | Center-aligned | Right-aligned |
|--------------|----------------|---------------|
| :--- | :---: | ---: |
| git status | git status | git status |
| git diff | git diff | git diff |

Command Description

| | |
|------------|--|
| git status | List all <i>new or modified</i> files |
| git diff | Show file differences that haven't been staged |

Left-aligned Center-aligned Right-aligned

| | | |
|------------|------------|------------|
| git status | git status | git status |
| git diff | git diff | git diff |

Github Flavoured Markdown (gfm)

Cheat sheets, Practice Exams and Flash cards  www.exampro.co/ghf

The **<details>** tag can be used to create collapsible sections

```
<details>  
  <summary>Tips for collapsed sections</summary>  
  
  ### You can add a header  
  
  You can add text within a collapsed section.  
  
  You can add an image or a code block, too.  
  
  ```ruby  
 puts "Hello World"
  ```  
  
</details>
```

► Tips for collapsed sections



GitHub Flavoured Markdown (gfm)

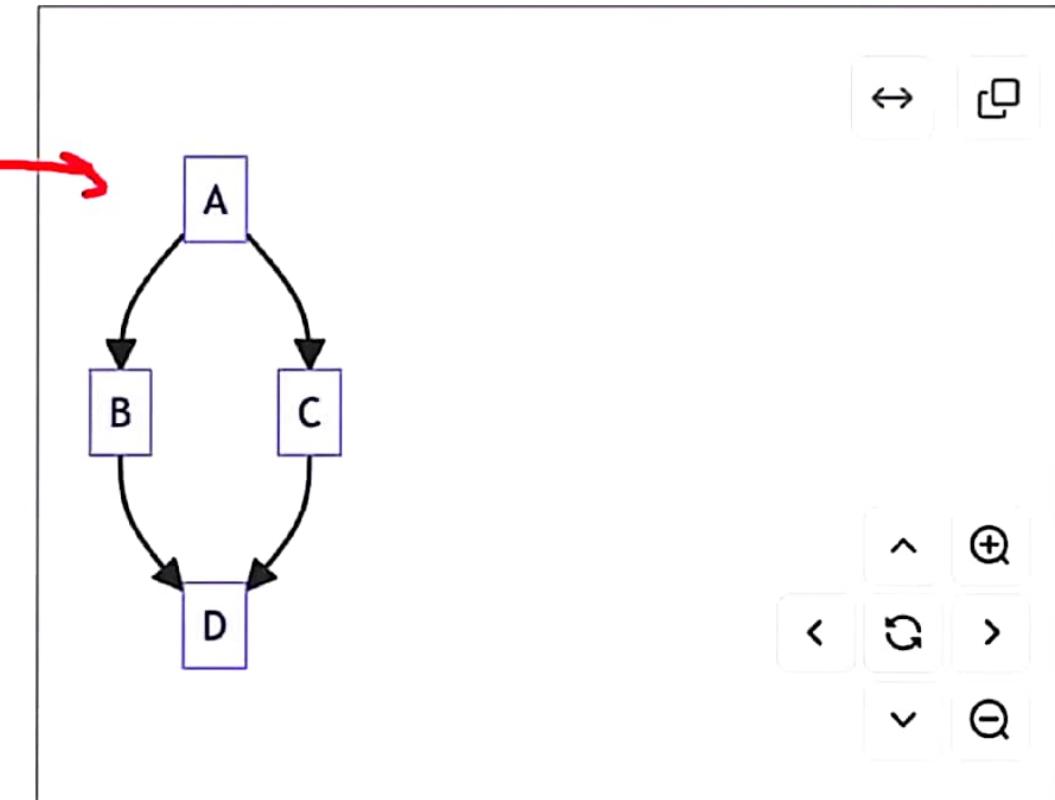
Cheat sheets, Practice Exams and Flash cards  www.exampro.co/ghf



Mermaid is a Markdown-inspired tool that renders text into diagrams

Here is a simple flow chart:

```
```mermaid
graph TD;
 A-->B;
 A-->C;
 B-->D;
 C-->D;
```
```



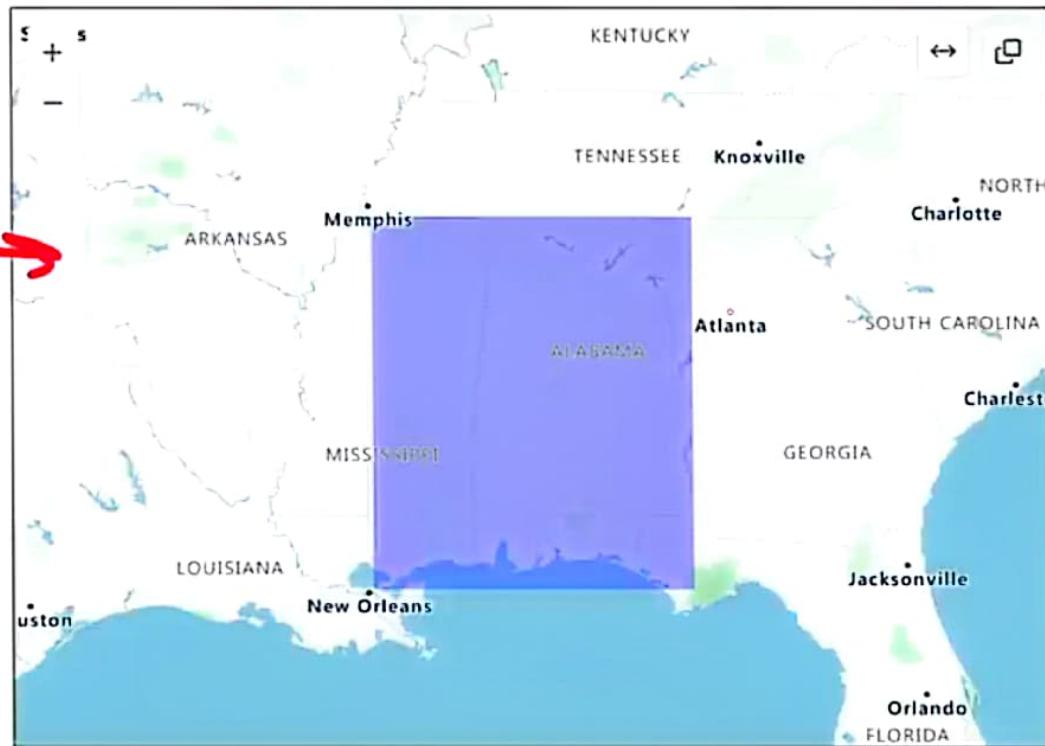
GitHub Flavoured Markdown (gfm)

Cheat sheets, Practice Exams and Flash cards  www.exampro.co/ghf



GeoJSON or TopoJSON syntax to create interactive maps.

```
```geojson
{
 "type": "FeatureCollection",
 "features": [
 {
 "type": "Feature",
 "id": 1,
 "properties": {
 "ID": 0
 },
 "geometry": {
 "type": "Polygon",
 "coordinates": [
 [
 [
 [-90,35],
 [-90,30],
 [-85,30],
 [-85,35],
 [-90,35]
]
]
]
 }
 }
]
}
````
```



Geojson example

```
```topojson
{
 "type": "Topology",
 "transform": {
 "scale": [0.0005000500050005, 0.0001000100010001000],
 "translate": [100, 0]
 },
 "objects": {
 "example": {
 "type": "GeometryCollection",
 "geometries": [
 {
 "type": "Point",
 "properties": {"prop0": "value0"},
 "coordinates": [4000, 5000]
 },
 {
 "type": "LineString",
 "properties": {"prop0": "value0", "prop1": 0},
 "arcs": [0]
 },
 {
 "type": "Polygon",
 "properties": {"prop0": "value0", "prop1": {"this": "that"}},
 "arcs": [[1]]
 }
]
 },
 "arcs": [[[4000, 0], [1999, 9999], [2000, -9999], [2001, 0], [4000, 0]]]
 }
}
```

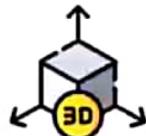
```



Topojson example

GitHub Flavoured Markdown (gfm)

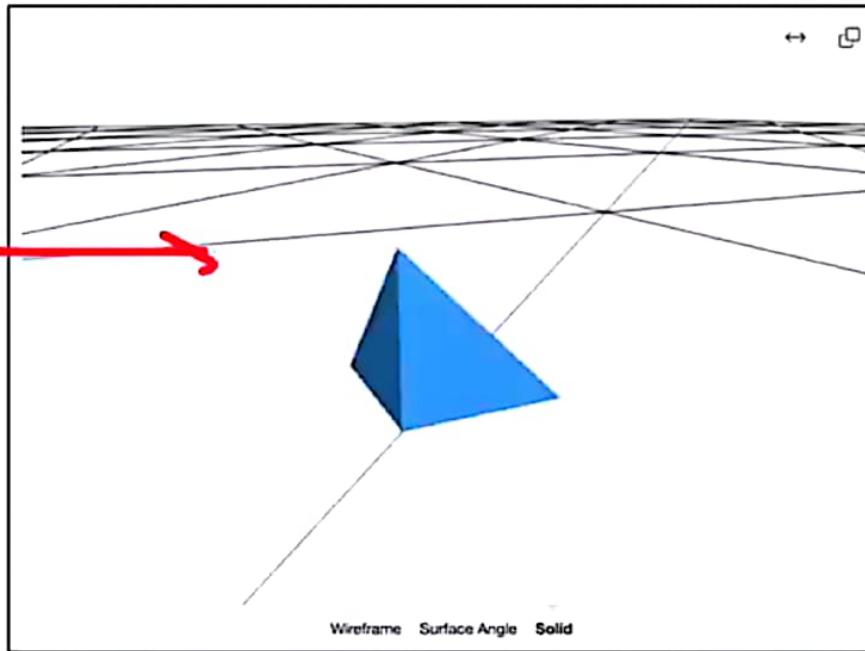
Cheat sheets, Practice Exams and Flash cards  www.exampro.co/ghf



ASCII STL syntax directly in markdown to create interactive 3D models

```
```stl
solid cube_corner
facet normal 0.0 -1.0 0.0
 outer loop
 vertex 0.0 0.0 0.0
 vertex 1.0 0.0 0.0
 vertex 0.0 0.0 1.0
 endloop
endfacet
facet normal 0.0 0.0 -1.0
 outer loop
 vertex 0.0 0.0 0.0
 vertex 0.0 1.0 0.0
 vertex 1.0 0.0 0.0
 endloop
endfacet
facet normal -1.0 0.0 0.0
 outer loop
 vertex 0.0 0.0 0.0
 vertex 0.0 0.0 1.0
 vertex 0.0 1.0 0.0
 endloop
endfacet
facet normal 0.577 0.577 0.577
 outer loop
 vertex 1.0 0.0 0.0
 vertex 0.0 1.0 0.0
 vertex 0.0 0.0 1.0
 endloop
endfacet
endsolid
...```

```



# GitHub Flavoured Markdown (gfm)

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

GFM can render mathematical expressions via Latex formatted math.

*GFM uses MathJax underneath as the tool to render*

Use the **\$\$** to denote start and end of latex

```
The Cauchy-Schwarz Inequality
$$\left(\sum_{k=1}^n a_k b_k \right)^2 \leq \left(\sum_{k=1}^n a_k^2 \right) \left(\sum_{k=1}^n b_k^2 \right)$$
```

Or use the **```math**  
Fencing block

```
The Cauchy-Schwarz Inequality

```math
\left( \sum_{k=1}^n a_k b_k \right)^2 \leq \left( \sum_{k=1}^n a_k^2 \right) \left( \sum_{k=1}^n b_k^2 \right)
```

```

The Cauchy-Schwarz Inequality

$$\left( \sum_{k=1}^n a_k b_k \right)^2 \leq \left( \sum_{k=1}^n a_k^2 \right) \left( \sum_{k=1}^n b_k^2 \right)$$

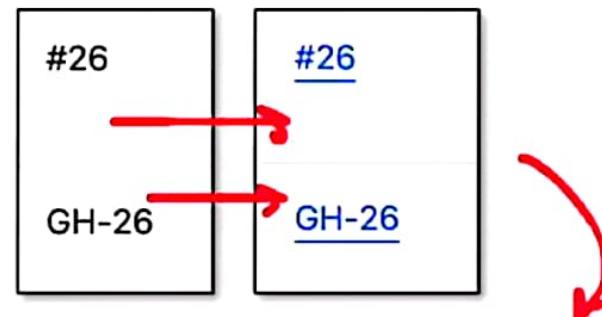
# GitHub Flavoured Markdown (gfm)

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

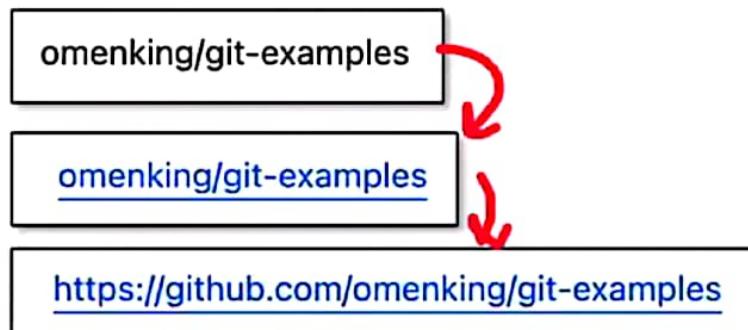
GFM will turn raw link automatically into a link



Within comments you can quickly link to issues and pull requests



Within comments you can quickly link to repos:



| Reference type              | Raw reference                                                                                                                                                                                | Short link                                     |
|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------|
| Commit URL                  | <a href="https://github.com/jlord/sheetsee.js/commit/a5c3785ed8d6a35868bc169f07e40e889087fd2e">https://github.com/jlord/sheetsee.js/commit/<br/>a5c3785ed8d6a35868bc169f07e40e889087fd2e</a> | <a href="#">a5c3785</a>                        |
| SHA                         | a5c3785ed8d6a35868bc169f07e40e889087fd2e                                                                                                                                                     | <a href="#">a5c3785</a>                        |
| User@SHA                    | jlord@a5c3785ed8d6a35868bc169f07e40e889087fd2e                                                                                                                                               | <a href="#">jlord@a5c3785</a>                  |
| Username/<br>Repository@SHA | jlord/<br>sheetsee.js@a5c3785ed8d6a35868bc169f07e40e889087fd2e                                                                                                                               | <a href="#">jlord/<br/>sheetsee.js@a5c3785</a> |

Commit SHAs can be autolinked as well:

# GitHub Flavoured Markdown (gfm)

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

Tasks lists can within an open issue can be created.

- You can associate tasks to specific issues (creating subtasks)
- GitHub can keep track of open or closed state of linked issue
- You can onclick turn a task and associate with an issue

```
- [x] #739
- [] https://github.com/octo-org/octo-repo/issues/740
- [] Add delight to the experience when all tasks are complete :tada:
```

 Convert text into issues #739  
  Keep issue state and checkboxes in sync #740  
 Add delight to the experience when all tasks are complete 🎉

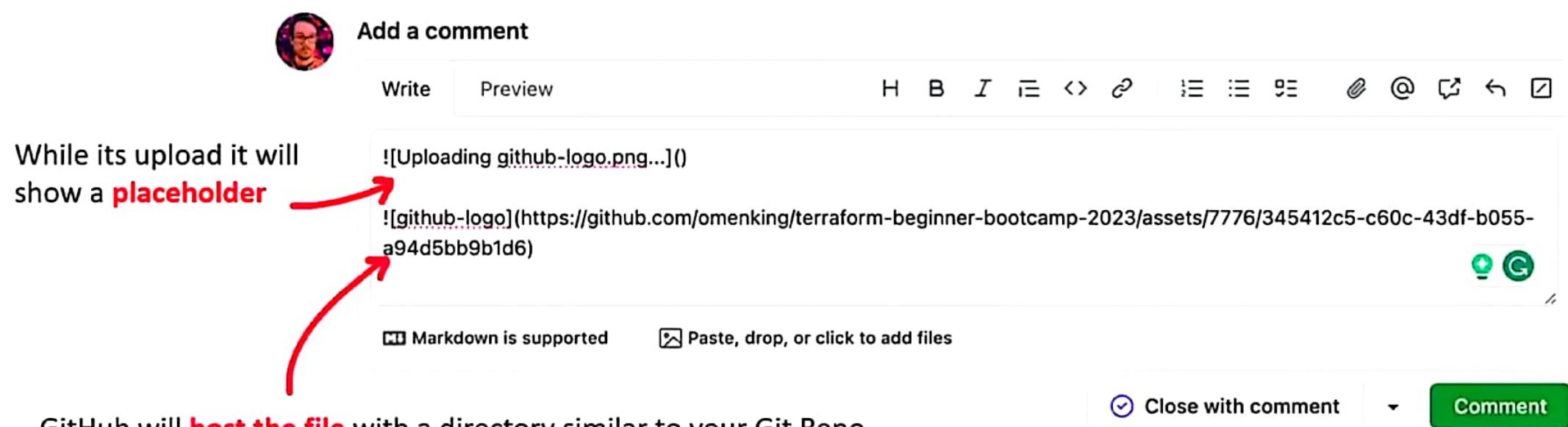
You can also easily reorder tasks

  
  Convert text in  
⋮  Add delight to the

# Uploading Files

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

You can directly upload files by dragging them into the markdown supported textareas in GitHub



The screenshot shows a GitHub comment interface. At the top, there's a placeholder for a profile picture and the text "Add a comment". Below that is a toolbar with "Write" and "Preview" tabs, and various rich text editing icons (bold, italic, etc.). A red arrow points from the text "While its upload it will show a placeholder" to the placeholder text "[Uploading github-logo.png...]()". Another red arrow points from the text "GitHub will host the file with a directory similar to your Git Repo" to the URL "[github-logo](https://github.com/omenking/terraform-beginner-bootcamp-2023/assets/7776/345412c5-c60c-43df-b055-a94d5bb9b1d6)". At the bottom, there are buttons for "Markdown is supported", "Paste, drop, or click to add files", "Close with comment" (with a checked checkbox), and a green "Comment" button.

While its upload it will show a **placeholder**

![Uploading github-logo.png...]()

![github-logo](https://github.com/omenking/terraform-beginner-bootcamp-2023/assets/7776/345412c5-c60c-43df-b055-a94d5bb9b1d6)

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

Close with comment

Comment

GitHub will **host the file** with a directory similar to your Git Repo

- Assume files you upload for public repos are public.

# Uploading Files

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

The maximum file size is:

- **10MB** for images and gifs
- **10MB** for videos uploaded to a repository owned by a user or organization on a free GitHub plan
- **100MB** for videos uploaded to a repository owned by a user or organization on a paid GitHub plan
- **25MB** for all other files

Supported files:

- PNG (*.png*)
- GIF (*.gif*)
- JPEG (*.jpg, .jpeg*)
- SVG (*.svg*)
- Log files (*.log*)
- Markdown files (*.md*)
- Microsoft Word (*.docx*)
- PowerPoint (*.pptx*)
- Excel (*.xlsx*) documents
- Text files (*.txt*)
- PDFs (*.pdf*)
- ZIP (*.zip, .gz, .tgz*)
- Video (*.mp4, .mov, .webm*)

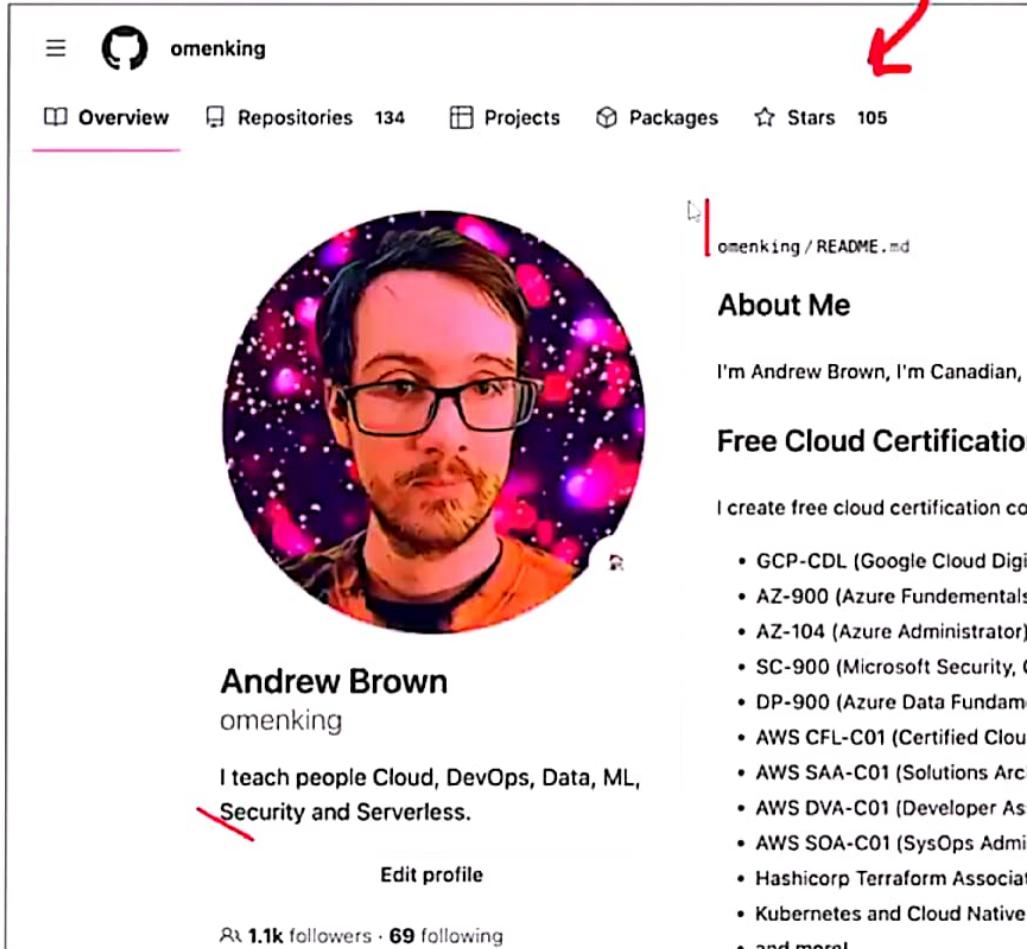
# GitHub User Profile Features

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

For Personal Accounts you can have a public GitHub User Profile to **showcase yourself as a developer.**

Features of your GitHub user profile include:

- **Profile Picture and Bio:** You can upload a profile picture and write a short bio about yourself.
- **User Profile Readme:** a readme file rendered on your profile.
- **Repositories:** Showcases your repositories, including options to pin your favorite ones.
- **Contributions Graph:** A graph displaying your contributions over the past year
  - (commits, pull requests, issues, and code reviews).
- **Followers and Following:** Displays the number of followers you have and the number of users you are following.
- **Stars:** Shows repositories you have starred.
- **Organizations:** Lists the GitHub organizations you are a part of.
- **Contribution Activity:** Detailed list of recent activity on GitHub.



The screenshot shows a GitHub user profile for the account 'omenking'. At the top, there's a navigation bar with 'Overview' (underlined), 'Repositories 134', 'Projects', 'Packages', 'Stars 105', and a dropdown menu. Below the navigation is a large circular profile picture of a man with glasses and a beard, set against a colorful, glowing background. To the right of the picture is the user's name, 'Andrew Brown', and their GitHub handle, 'omenking'. Below this, a bio states: 'I teach people Cloud, DevOps, Data, ML, Security and Serverless.' There are 'Edit profile' and 'Edit bio' buttons. At the bottom, it shows 'At 1.1k followers · 69 following'.

A red arrow points from the text 'showcase yourself as a developer.' in the yellow box above to the 'About Me' section of the profile screenshot.

# GitHub User Profile Features

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

In your account settings under Public Profile you can change:

- Display name
- Bio
- URL
- Social Account URLs

You can also specify if you are looking for work

**Jobs profile**

Available for hire

**Save jobs profile**



**Public profile**

**Name**  
Andrew Brown  
Your name may appear around GitHub where you contribute or are mentioned. You can remove it at any time.

**Public email**  
andrew@monsterboxpro.com    
You can manage verified email addresses in your [email settings](#).

**Bio**  
I teach people Cloud, DevOps, Data, ML, Security and Serverless.  
You can @mention other users and organizations to link to them.

**Pronouns**  
Don't specify

**URL**  
<https://www.exampro.co>

**Social accounts**  
 <https://twitter.com/andrewbrown>  
 <https://www.linkedin.com/in/andrew-wc-brown/>

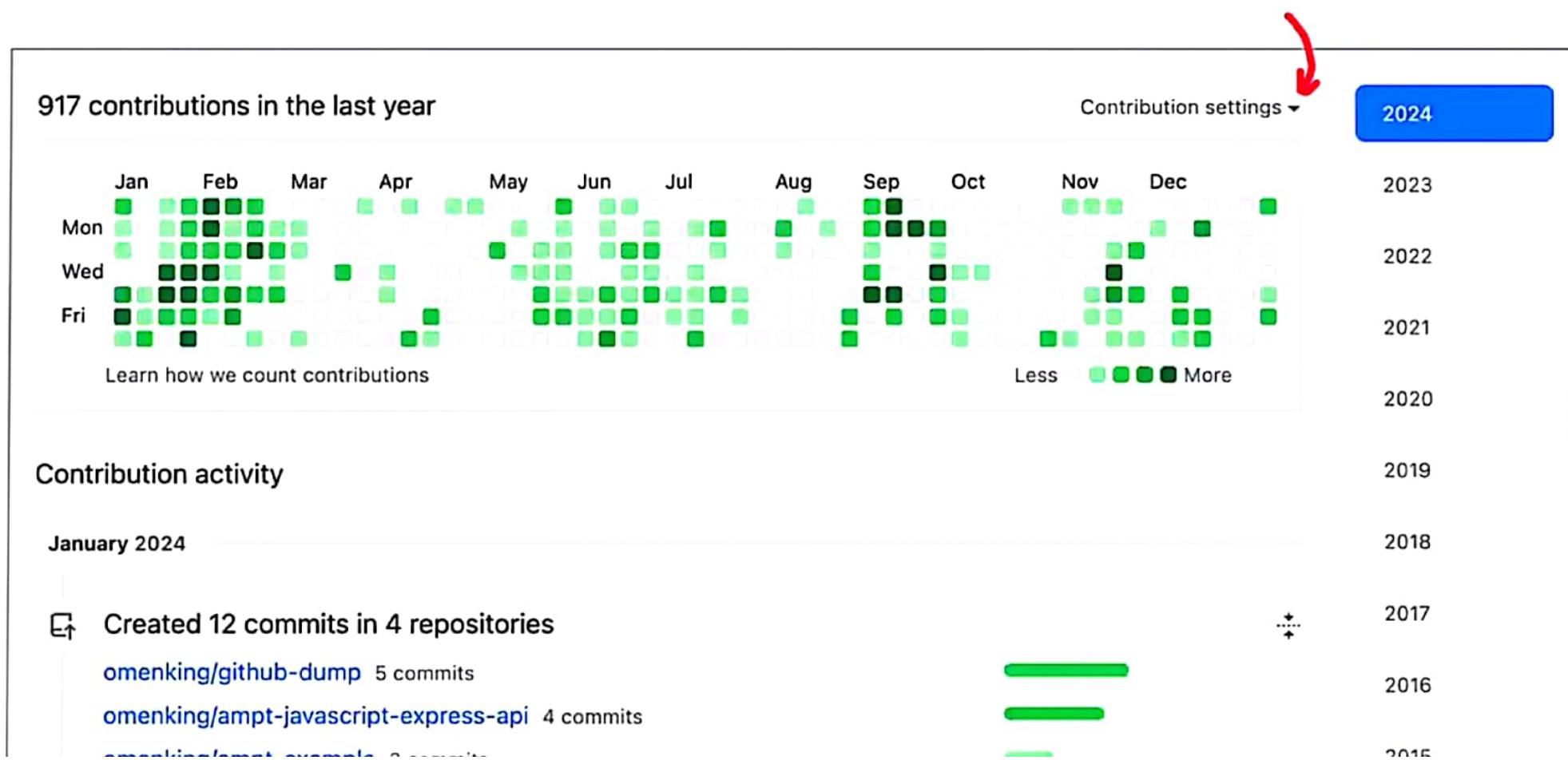
 [Link to social profile](#)

**Profile picture**  
  


# GitHub User Profile Features

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

Contributor Activity in your profile will show **how active (for public projects) you are on GitHub.**

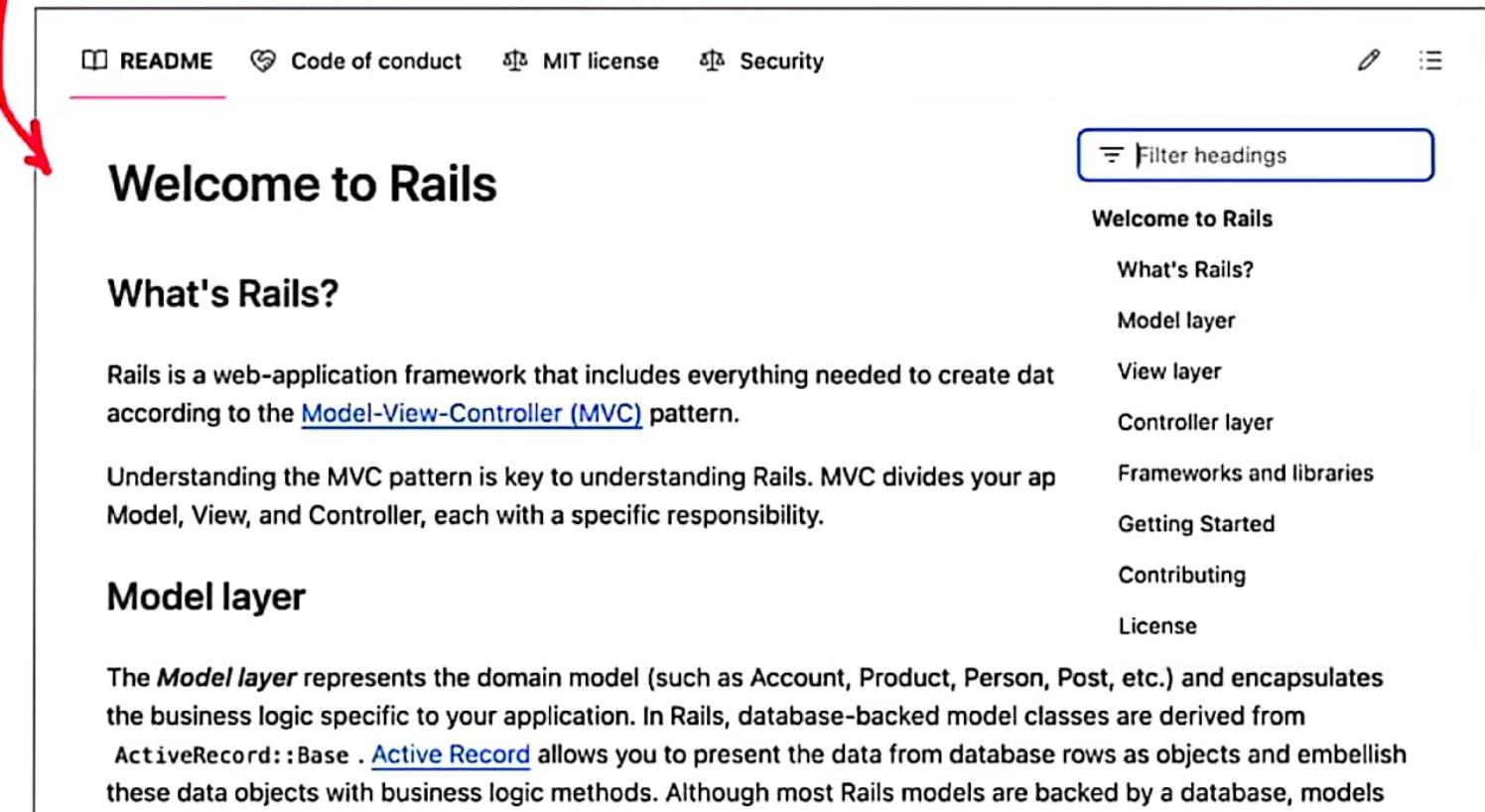


# README Files

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

Readme files are markdown files that provided documentation/instructional information.

A repo that has a **Readme.md or README file in project's root** will be rendered on the Github Repo page for each access.



The screenshot shows a GitHub repository page with a rendered README file. A red arrow points from the text "A repo that has a **Readme.md or README file in project's root** will be rendered on the Github Repo page for each access." to the "README" tab in the top navigation bar. The main content area displays the rendered README file, which includes sections like "Welcome to Rails", "What's Rails?", "Model layer", and "License". The sidebar on the right lists navigation links such as "Welcome to Rails", "What's Rails?", "Model layer", "View layer", "Controller layer", "Frameworks and libraries", "Getting Started", "Contributing", and "License".

README    Code of conduct    MIT license    Security

Welcome to Rails

Filter headings

Welcome to Rails

What's Rails?

Model layer

View layer

Controller layer

Frameworks and libraries

Getting Started

Contributing

License

Welcome to Rails

What's Rails?

Model layer

View layer

Controller layer

Frameworks and libraries

Getting Started

Contributing

License

What's Rails?

Rails is a web-application framework that includes everything needed to create dat according to the [Model-View-Controller \(MVC\) pattern](#).

Understanding the MVC pattern is key to understanding Rails. MVC divides your ap Model, View, and Controller, each with a specific responsibility.

The *Model layer* represents the domain model (such as Account, Product, Person, Post, etc.) and encapsulates the business logic specific to your application. In Rails, database-backed model classes are derived from `ActiveRecord::Base`. [Active Record](#) allows you to present the data from database rows as objects and embellish these data objects with business logic methods. Although most Rails models are backed by a database, models

# Basic Repo Navigation

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

Within a Github repo you'll have a **navigation bar** to the various feature of your GitHub repo:



- **Code:** The main tab where the repository's source code, files, and folders are located.
- **Issues:** Tracks problems or ideas for the project, allowing collaboration and discussion.
- **Pull Requests:** Used for managing contributions from other users, enabling code review and discussion before merging changes.
- **Actions:** Manages continuous integration and continuous deployment (CI/CD) workflows.
- **Projects:** A board for organizing and prioritizing work, similar to kanban or task management boards.
- **Wiki:** A space for the project's documentation.
- **Security:** Features security-related resources, including security policies and vulnerability reports.
- **Insights:** Provides statistics and insights on the project's activity and contributions.
- **Settings:** Contains repository settings, including access controls, webhooks, and other configurations

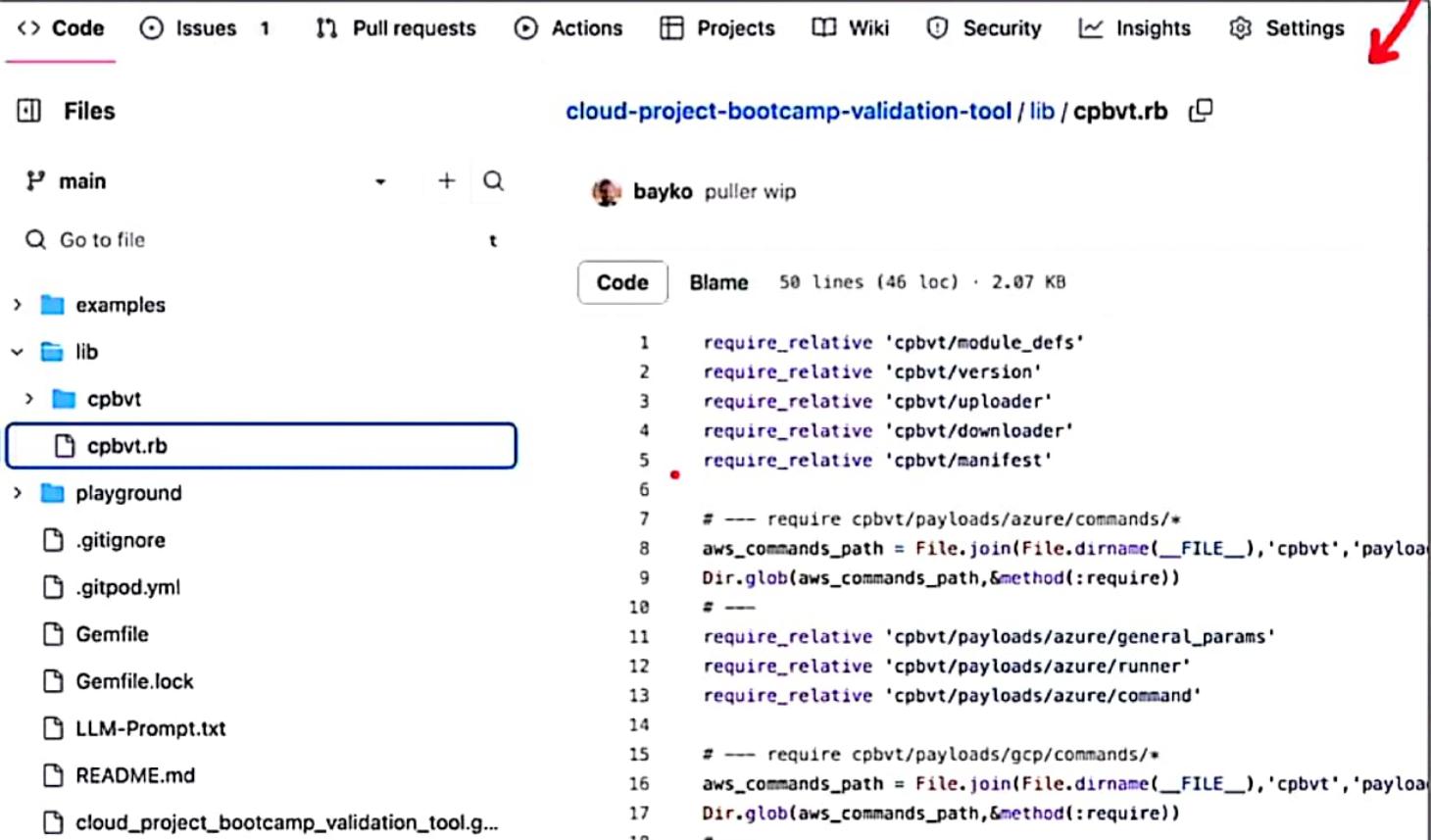
# Basic Repo Navigation

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

Within a Github Repo when you click into a file you can then easily navigate all repo files

You can jump to **specific files** using the search

You can see preview of the **file content**



cloud-project-bootcamp-validation-tool / lib / cpbvt.rb

Code Issues 1 Pull requests Actions Projects Wiki Security Insights Settings

Files

main

Go to file

- examples
- lib
- cpbvt
- cpbvt.rb
- playground
- .gitignore
- .gitpod.yml
- Gemfile
- Gemfile.lock
- LLM-Prompt.txt
- README.md
- cloud\_project\_bootcamp\_validation\_tool.g...

bayko puller wip

Code Blame 50 lines (46 loc) · 2.07 KB

```
1 require_relative 'cpbvt/module_defs'
2 require_relative 'cpbvt/version'
3 require_relative 'cpbvt/uploader'
4 require_relative 'cpbvt/downloader'
5 require_relative 'cpbvt/manifest'
6
7 # --- require cpbvt/payments/azure/commands/*
8 aws_commands_path = File.join(File.dirname(__FILE__), 'cpbvt', 'payments')
9 Dir.glob(aws_commands_path, &method(:require))
10 #
11 require_relative 'cpbvt/payments/azure/general_params'
12 require_relative 'cpbvt/payments/azure/runner'
13 require_relative 'cpbvt/payments/azure/command'
14
15 # --- require cpbvt/payments/gcp/commands/*
16 aws_commands_path = File.join(File.dirname(__FILE__), 'cpbvt', 'payments')
17 Dir.glob(aws_commands_path, &method(:require))
18 # ---
```

# Creating a new repo

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

When you create a repo you choose a owner:

- Personal account
- An organization that you belong to

You need to choose an available github repo name

- Names are unique based on the scope of the owner

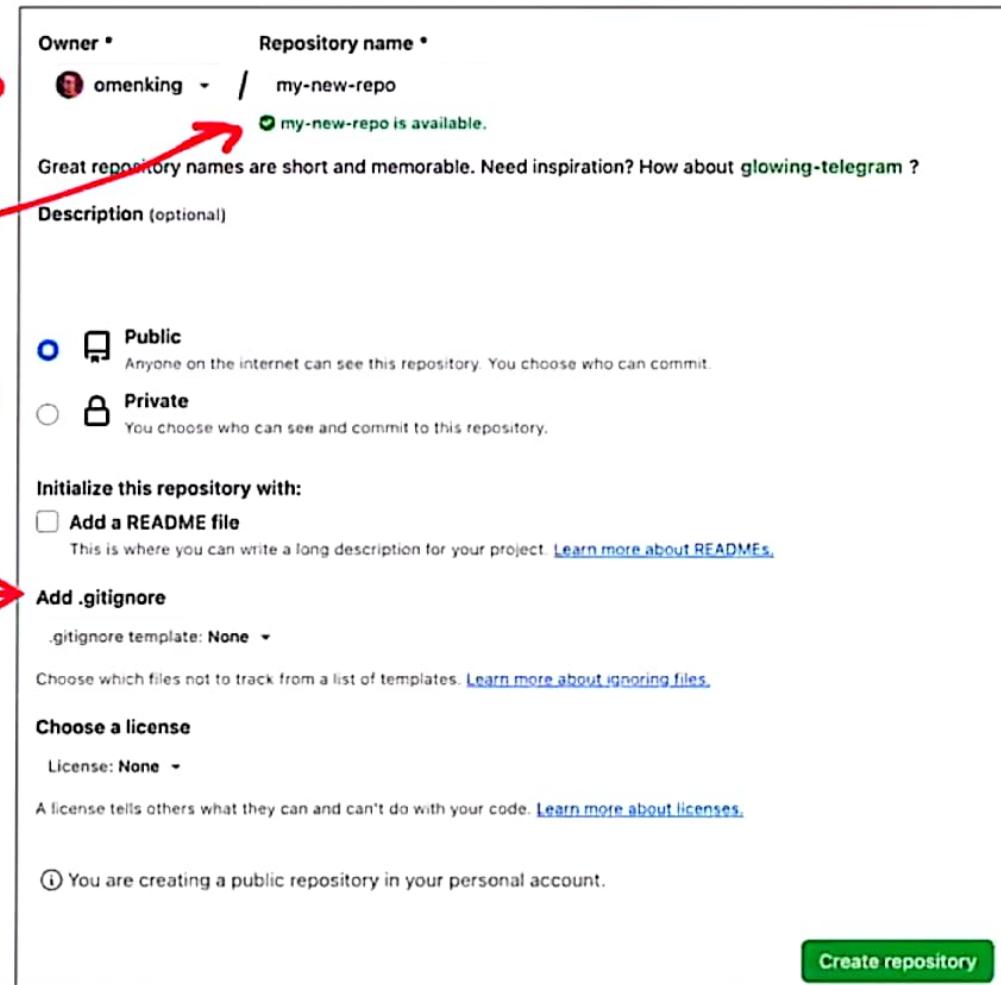
Your repo can either be public or private

You can quickly add a:

- Readme file
- .gitignore file
- License file

Github CLI can be used to create a GitHub repo

```
gh repo create my-project --public --clone
```



The screenshot shows the GitHub 'Create repository' form. It includes fields for 'Owner' (set to 'omenking'), 'Repository name' ('my-new-repo'), and a note that it's 'available'. There's a 'Description (optional)' field, a visibility section with 'Public' selected (described as 'Anyone on the internet can see this repository. You choose who can commit.'), and a 'Private' option (described as 'You choose who can see and commit to this repository.'). Below these are sections for initializing the repository with a 'README file' (with a link to 'Learn more about READMEs.') and an '.gitignore' file (with a link to 'Learn more about ignoring files.' and a dropdown for 'None'). A 'Choose a license' section is present with a 'None' dropdown and a link to 'Learn more about licenses.'. A note at the bottom states 'You are creating a public repository in your personal account.' and a green 'Create repository' button.

# Maintaining a repo

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

## Repository name

github-dump

Rename



You can **change your repo name**, (but try not to because it can break external links and documentation pointing to a public repo)

Repo names are scoped based on the personal or organizational account.

## Default branch

The default branch is considered the “base” branch in your commits are automatically made, unless you specify a diff

main



You can **change the base branch** (default branch).  
You can rename the default branch.  
“main” is the unspoken best practice for naming your base branch

You can **opt-in-and-out of some features for your Github Repo**

Features will appear in the Github repo navigation bar

## Features

### Wikis

Wikis host documentation for your repository.



### Upgrade or make this repository public

GitHub Wikis is a simple way to let others contr pages to use for documentation, examples, sup

[Upgrade](#)

[Learn more about wikis](#)

### Issues

Issues integrate lightweight task tracking into your repository. K reference them in commit messages.

# Maintaining a repo

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)



The **Danger Zone** contains actions you need to think twice about because they cannot be undone.

## Change repo visibility

When you make a private repo public:

- The code will be visible to everyone who can visit <https://github.com>
- Anyone can fork your repository.
- All push rulesets will be disabled.
- Your changes will be published as activity.

## Disable Branch Protection Rules

Branch protect rules are strict workflow rules like disallowing anyone pushing to main.

You can disable all protections temporarily eg. For quick fixes

## Transfer ownership

When you want someone else to be the owner of the repo.

## Archive this repository

When the repo to be read-only.

## Danger Zone

### Change repository visibility

This repository is currently public.

[Change visibility](#)

### Disable branch protection rules

Disable branch protection rules enforcement and APIs

[Disable branch protection rules](#)

### Transfer ownership

Transfer this repository to another user or to an organization where you have the ability to create repositories.

[Transfer](#)

### Archive this repository

Mark this repository as archived and read-only.

[Archive this repository](#)

### Delete this repository

Once you delete a repository, there is no going back. Please be certain.

[Delete this repository](#)

## Delete this repository

When you want to delete your repo.

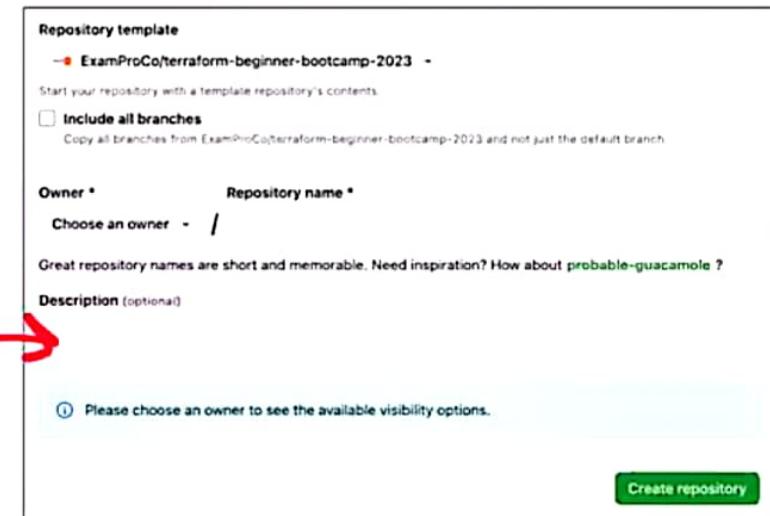
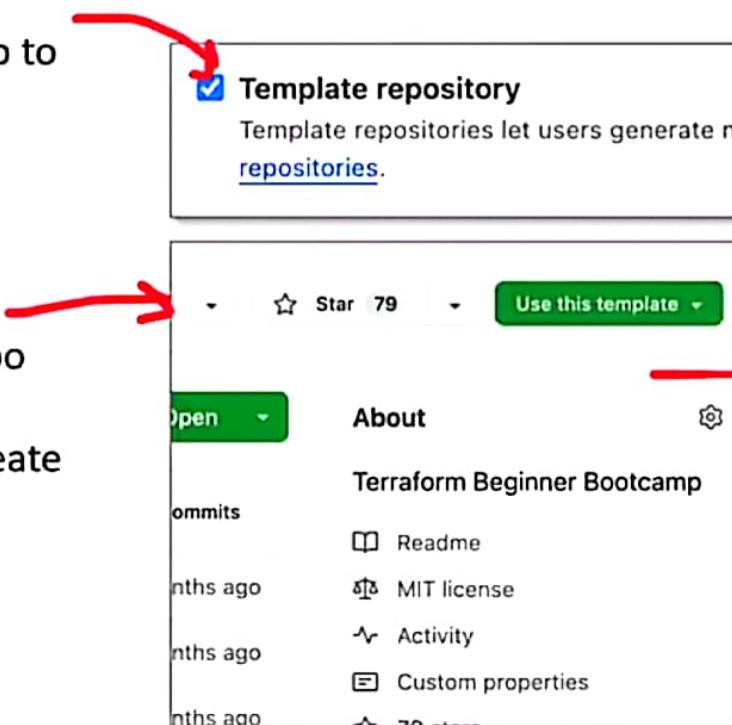
Once its gone its gone, there is not trash bin you can pull it out of.

# Repo Templates

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

**Github Repo Templates** is a feature for public repos that **allow other GitHub users to make a copy of the contents of the template repo** to use as starting point of their own repo.

We set the repo to be a template



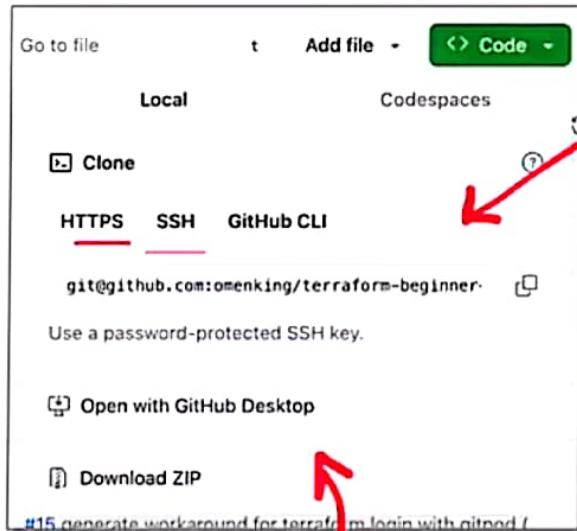
Templates are different from cloning because it's a clean repo with no previous history.

Their repo will indicate they **generated** their repo from the repo template



# Cloning a repo

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)



You can also:

- Open in GitHub Desktop which will perform the clone for you
- Download a ZIP containing the contents of the repo

You can **clone a repo programmatically** three different ways:

## 1. HTTPS

```
git clone https://github.com/omenking/terraform-beginner-bootcamp-2023.git
```

You supply Github username and password on clone, you need to set git to cache credentials

## 2. SSH

```
git clone git@github.com:omenking/terraform-beginner-bootcamp-2023.git
```

Authorization is based on SSH keypair  
you have to upload the Public key to your GitHub account

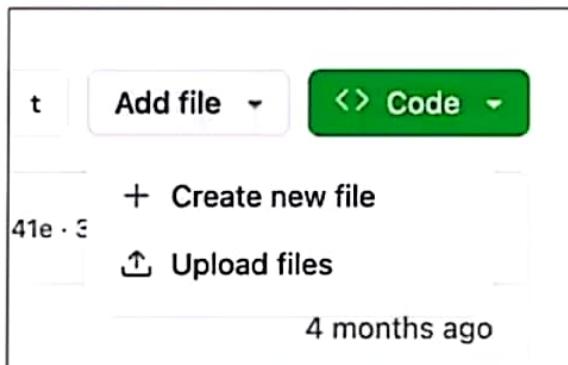
## 3. Github CLI

```
gh repo clone omenking/terraform-beginner-bootcamp-2023
```

Authorization is based on the credentials when did a gh login

# Adding files to a repo

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

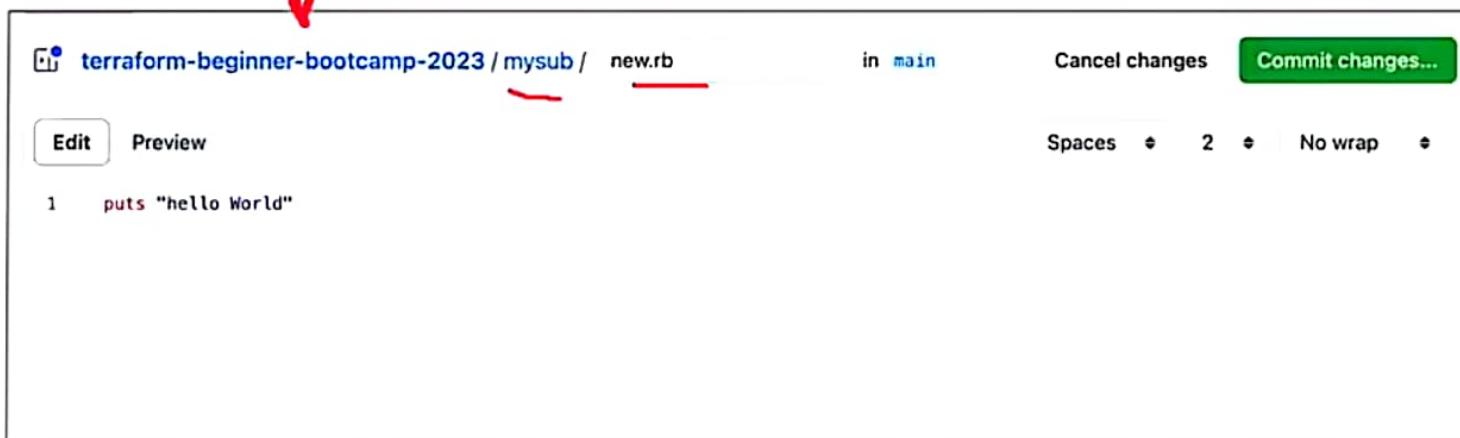


Within the Github UI:

- Multiple Text and Binary files can be uploaded.
- New single text file can be **created**

When creating files in place, you have a basic text editor.

You can also **create nested folders by typing a forward slash in the filename.**



A screenshot of the GitHub text editor. The file path shown is 'terraform-beginner-bootcamp-2023 / mysub / new.rb'. The editor shows the code: '1 puts "hello World"'. There is a red arrow pointing from the text 'You can also ...' to the folder path in the file selector.

When adding multiple files that you also need to edit, you can also quickly use **Github.dev** at no cost

Using git we can create a new branch from an existing branch.  
We then want to push upstream our branch to GitHub

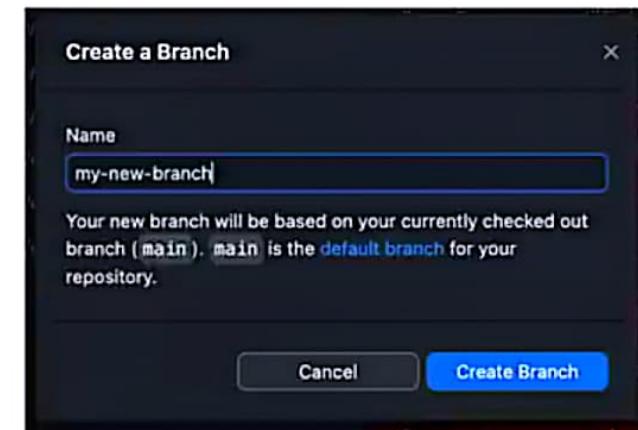
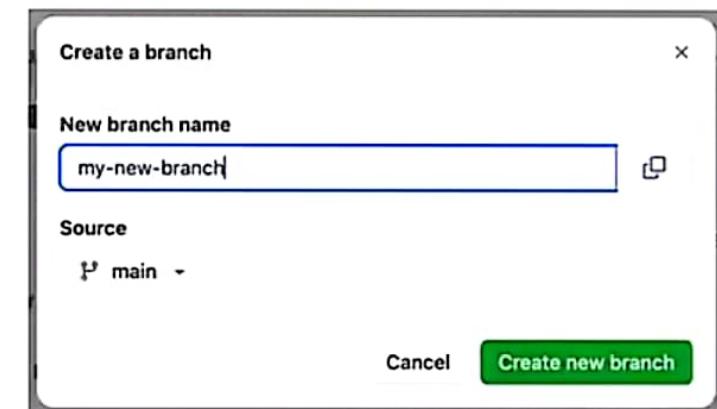
```
git checkout -b staging
change files, create commits
git push -u origin staging
```

*-u is the short hand flag for set-upstream.*

Branches can be created from Issues.  
The branch and issues will be associated/link

Development   
[Create a branch](#) for this issue or link a pull request.

Branches can be created  
in GitHub Desktop



# Starred repos

Cheat sheets, Practice Exams and Flash cards ➡ [www.exampro.co/ghf](http://www.exampro.co/ghf)

A screenshot of a GitHub user profile. At the top, there are statistics: 'Fork 21.5k' and 'Starred 54.2k'. A red arrow points from the text 'Similar to bookmarking...' to the 'Starred' count. Below this is a 'Lists' section with three items: 'Future ideas', 'My stack', and 'Inspiration', each with a checkbox. There is also a '+ Create list' button. At the bottom are tags: 'framework', 'mvc', and 'activejob'.

Similar to bookmarking, it's a way to mark a repository as interesting or to keep track of it for reference.

Stars are public and can be seen by everyone, often used as a **measure of a project's popularity**.

You easily find pages you starred at: <https://github.com/stars>

A screenshot of the GitHub 'stars' page at [github.com/stars](https://github.com/stars). The page has a blue header with the URL. A red arrow points from the text 'Sort: Recently starred' to the 'Recently starred' sorting option in the header. The main content shows a search bar 'Search stars...', a 'STARRED REPOSITORIES' section, and two repository entries:

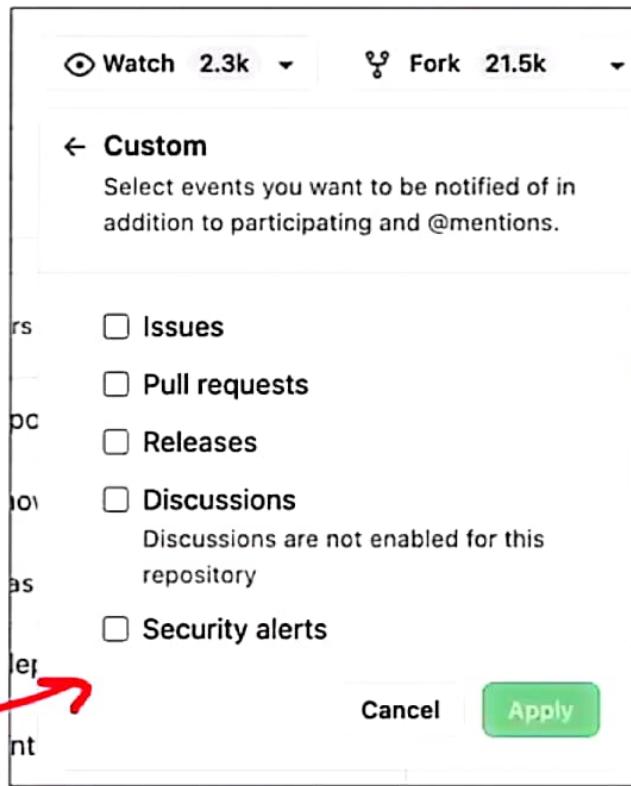
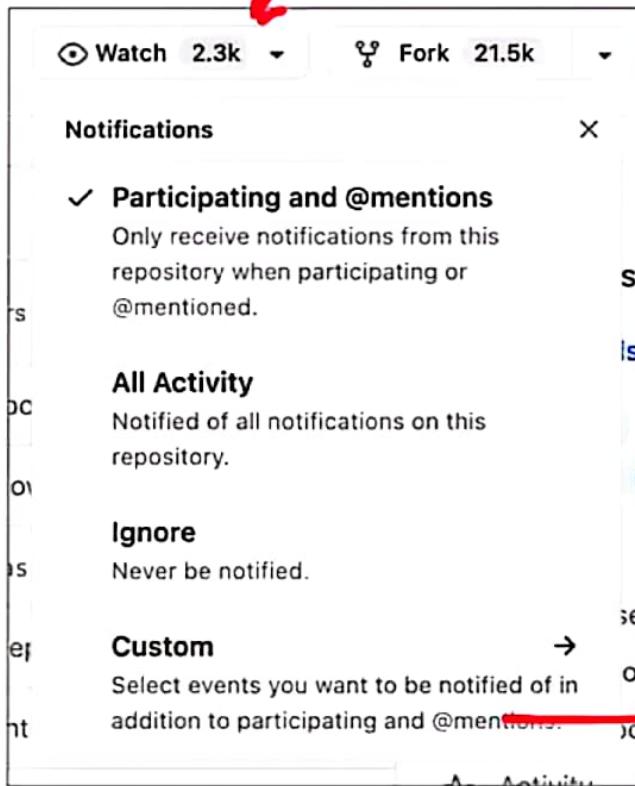
- hightechu / hightechu-discord-bot**  
HighTechU - Discord Bot (Summer 2021)  
JavaScript ⭐ 1 🏷 3  
Starred 3 weeks ago
- Marconiusiii / OhCraps**  
Python-based Craps game for Terminal.  
Python ⭐ 6 🏷 1  
Starred on Jul 14, 2022

On the right side, there are filters: 'All stars', 'All repositories', 'Your repositories', 'Others' repositories', 'Topics', and 'Filter by languages'.

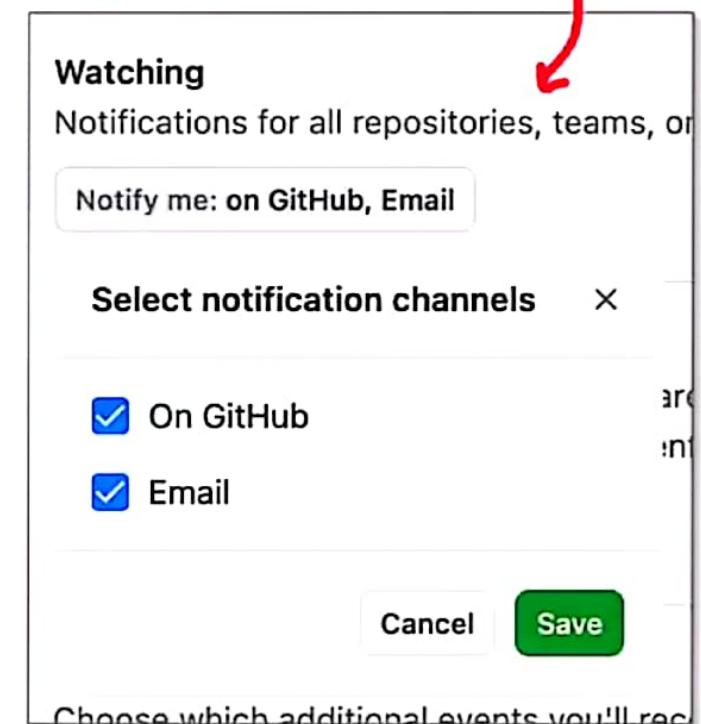
# Watching Repos

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

Watching a repo allows you to stay informed about activities occurring within a repo.



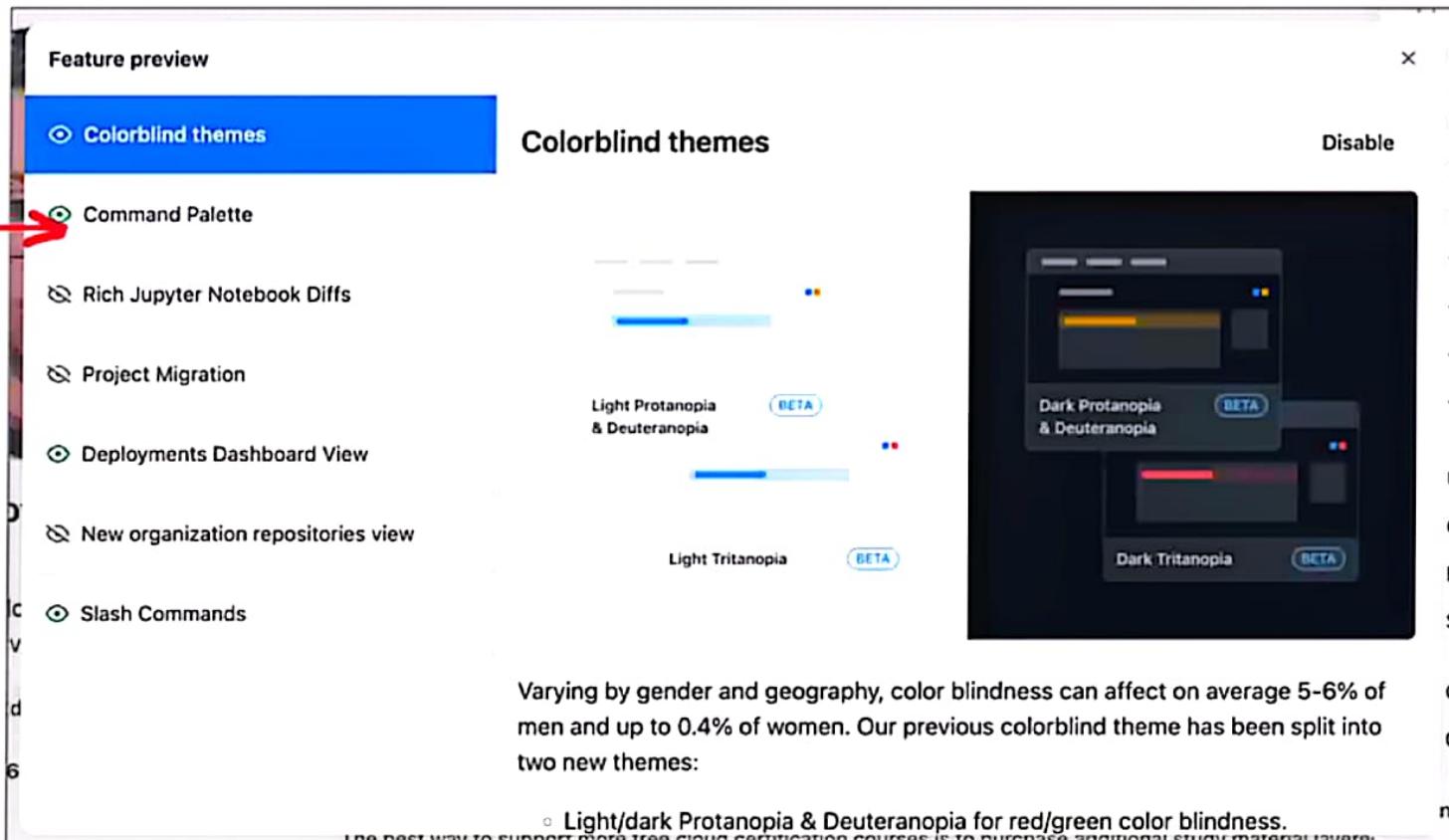
In your Github Account you can specify how you want to be notified



# Feature previews

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

Feature previews allows you to **enable or disable features that are in beta** in your personal account



The screenshot shows the GitHub Feature Preview interface. On the left, there's a sidebar with icons for Upgrade, Copilot, Feature preview (which has a red arrow pointing to it), and Settings. The main area is titled "Feature preview" and lists several beta features:

- Colorblind themes (selected, highlighted in blue)
- Command Palette
- Rich Jupyter Notebook Diffs
- Project Migration
- Deployments Dashboard View
- New organization repositories view
- Slash Commands

For the "Colorblind themes" feature, there are two sliders: "Light Protanopia & Deutanopia" (BETA) and "Light Tritanopia" (BETA). To the right, there's a preview window showing three colorblind themes: "Dark Protanopia & Deutanopia" (BETA), "Dark Tritanopia" (BETA), and another variant. Below the preview, text explains the prevalence of color blindness and details about the new themes.

Varying by gender and geography, color blindness can affect on average 5-6% of men and up to 0.4% of women. Our previous colorblind theme has been split into two new themes:

- Light/dark Protanopia & Deutanopia for red/green color blindness.

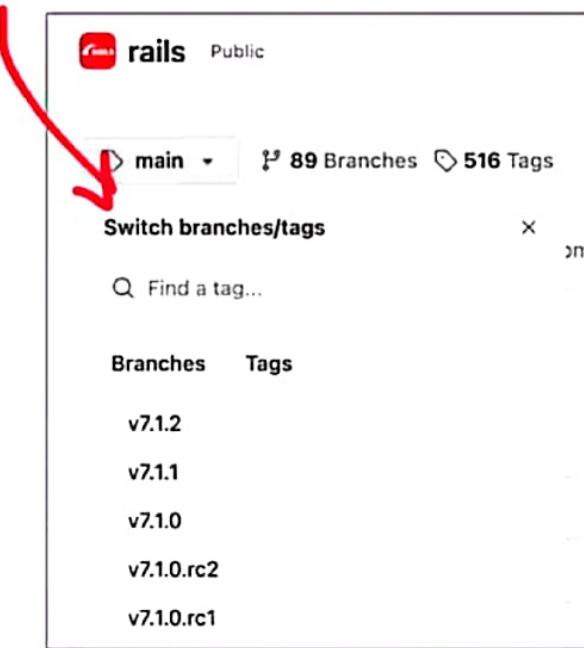
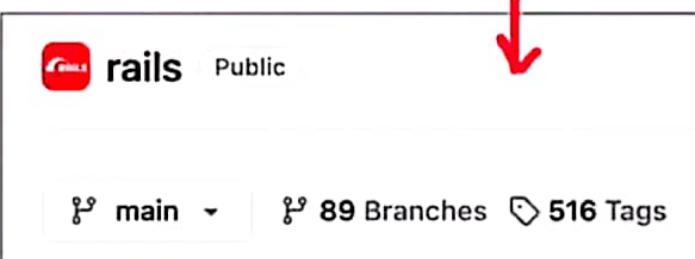
The best way to support more free cloud certification courses is to purchase additional study material layers!

# Tags

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

**Tagging** is used to capture a point in history to mark version release of your codebase

GitHub makes it **easy to explore tagged versions** of Git repo.



The image shows two screenshots of a GitHub repository page for 'rails'. The left screenshot shows the main repository summary with a red arrow pointing to the '516 Tags' link. The right screenshot shows the 'Switch branches/tags' dropdown menu, with a red arrow pointing to the 'Tags' tab, which lists several tag names: v7.1.2, v7.1.1, v7.1.0, v7.1.0.rc2, and v7.1.0.rc1.

Common git tag commands:

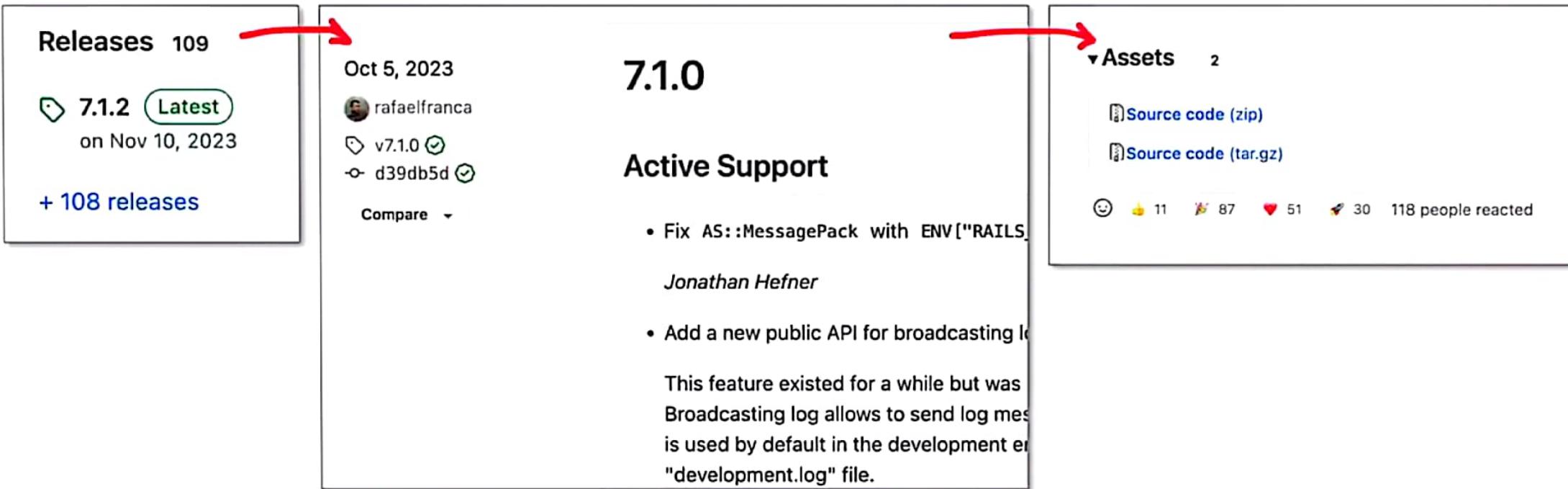
```
git tag v1.0.0 # tag a commit
git push --tags # push tag to remote
git checkout v1.0.0 # checkout codebase at tag
git tag -d v1.0.0 # delete local tag
git push --delete origin tagname # delete remote tag
```

CI/CD pipelines may be configured to deploy on the presence of a new tag on production branch

# GitHub Releases

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

GitHub Releases allows you to **create releases with release notes and linked assets** such as zip source or binaries for specific platforms



The screenshot shows the GitHub Releases interface for a repository. On the left, a sidebar displays "Releases 109" with a red arrow pointing to the "Latest" release. The main content area shows a release titled "7.1.0" made on Oct 5, 2023, by user rafaelfranca. A red arrow points from the sidebar to this release. The release notes include a fix for AS::MessagePack and the addition of a new public API for broadcasting logs. The "Assets" section on the right contains two items: "Source code (zip)" and "Source code (tar.gz)". Below the assets, social interaction metrics are shown: 11 likes, 87 comments, 51 stars, 30 forks, and 118 people reacted.

Releases 109

7.1.2 Latest on Nov 10, 2023 + 108 releases

Oct 5, 2023  
rafaelfranca  
v7.1.0 d39db5d

Compare ▾

**7.1.0**

**Active Support**

- Fix AS::MessagePack with ENV["RAILS"]
- Add a new public API for broadcasting logs

This feature existed for a while but was removed in v7.0.0. Broadcasting log allows to send log messages to a log file. It is used by default in the development environment, but can be disabled by setting "development.log" file.

Assets 2

Source code (zip) Source code (tar.gz)

11 87 51 30 118 people reacted

# GitHub Packages

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

**GitHub Packages** is a platform for **hosting and managing packages, including containers** and other dependencies.

Pushing Docker container to **GitHub Packages**

```
GH_USERNAME="andrew-wc-brown"
IMAGE_NAME="hello-world"
VER="1.0.0"

Sign-in to docker
echo $GH_TOKEN | docker login ghcr.io -u $GH_USERNAME --password-stdin
Tag docker container
docker tag $IMAGE_NAME:$VER ghcr.io/$GH_USERNAME/$IMAGE_NAME:$VER
Push to Docker Packages
docker push ghcr.io/$GH_USERNAME/$IMAGE_NAME:$VER
```

Supported package registry:

- Javascript (npm package)
- Ruby (gems)
- Java (maven and gradle packages)
- .Net (don't net packages)
- Docker images

```
FROM alpine:latest
CMD echo "Hello World!"
```

Dockerfile

```
docker build -t hello-world .
docker run hello-world
```

Build and run Docker image



GitHub Actions could be used to build and then public packages to GitHub Packages

# Repo Insights

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

## Insights

For a Github Repo under the Insights tab you can gain lots of statistical graphs about the repo.

Insights contains:

- **Pulse:** Overview of recent activity (issues, pull requests).
- **Contributors:** Lists contributors and their activity stats.
- **Community Standards:** Checks for essential community health files.
- **Commits:** History of all commits in the repo.
- **Code Frequency:** Graph of code additions and deletions over time.
- **Dependency Graph:** Visualizes code dependencies.
- **Network:** Shows fork relationships and variations.
- **Forks:** Number and links to repository forks.

Repository insights is not listed as an available feature in the GitHub free tier?

# Repo Insights

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

## Pulse contains:

- Pull requests open/close ratio
- Issues open/close ratio
- Summary of activity
- Graph of top contributors
- List of merged pull requests
- List of issues closed
- List of issues opened
- List of unresolved conversations

December 11, 2023 – January 11, 2024

Period: 1 month ▾

### Overview



Excluding merges, 80 authors have pushed 250 commits to main and 389 commits to all branches. On main, 482 files have changed and there have been 5,415 additions and 7,844 deletions.



217 Pull requests merged by 73 people

[Fix #50713] Do not trigger loading of ActiveJob::Base in ActiveJob::TestHelper

#50715 merged 6 hours ago

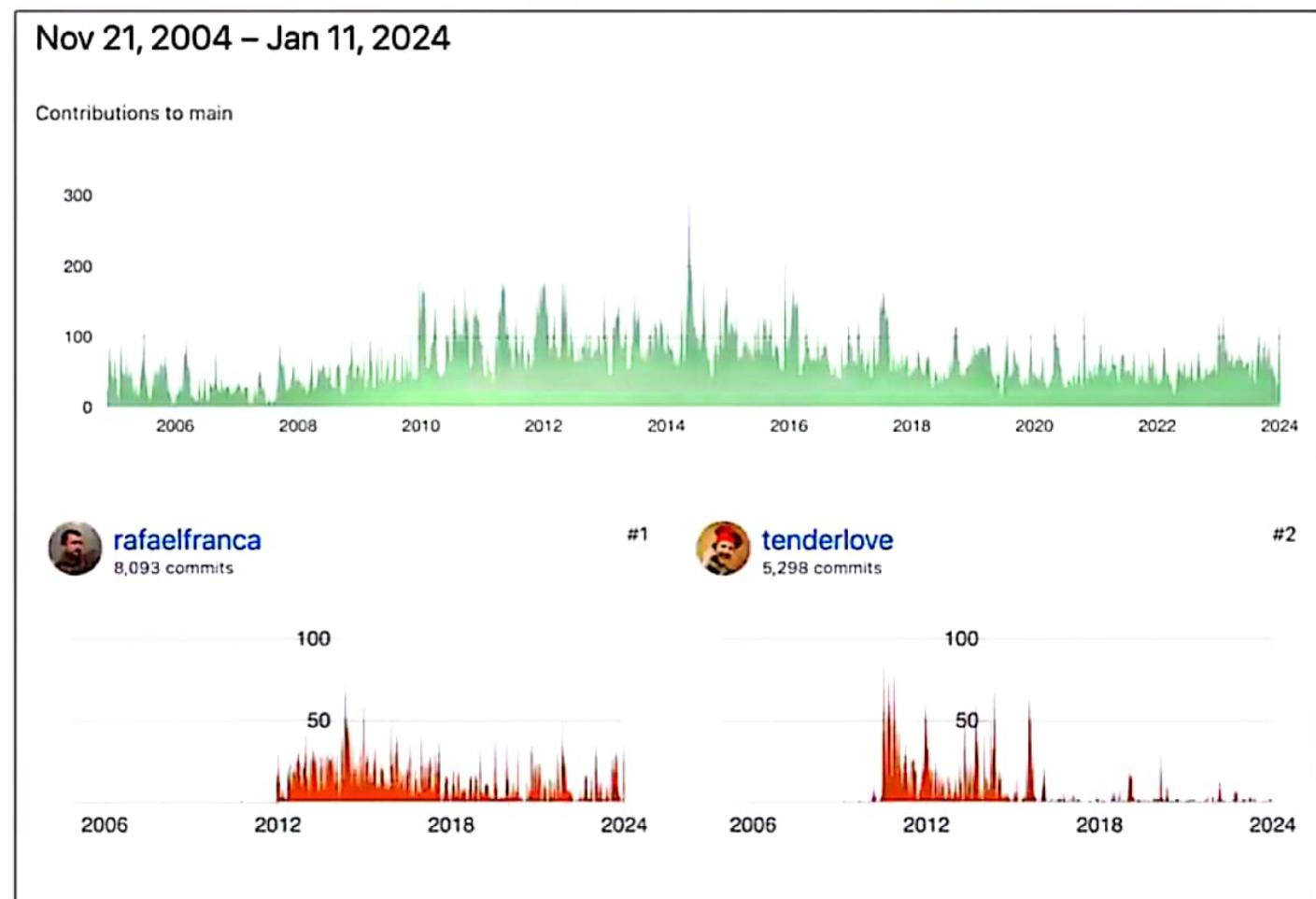
Make sure attribute\_aliases is consistent

# Repo Insights

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

## Contributors contains:

- Graph of all commits over a time period
- Graph of specific contributors commits over a time period



# Repo Insights

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

## Community Standards contains:

- A checklist of recommended community standards and how much a community profile has completed

This graph will only show for repos that have community profiles.

Community profiles are for open-source projects on GitHub.

### Community Standards

Here's how this project compares to [recommended community standards](#).

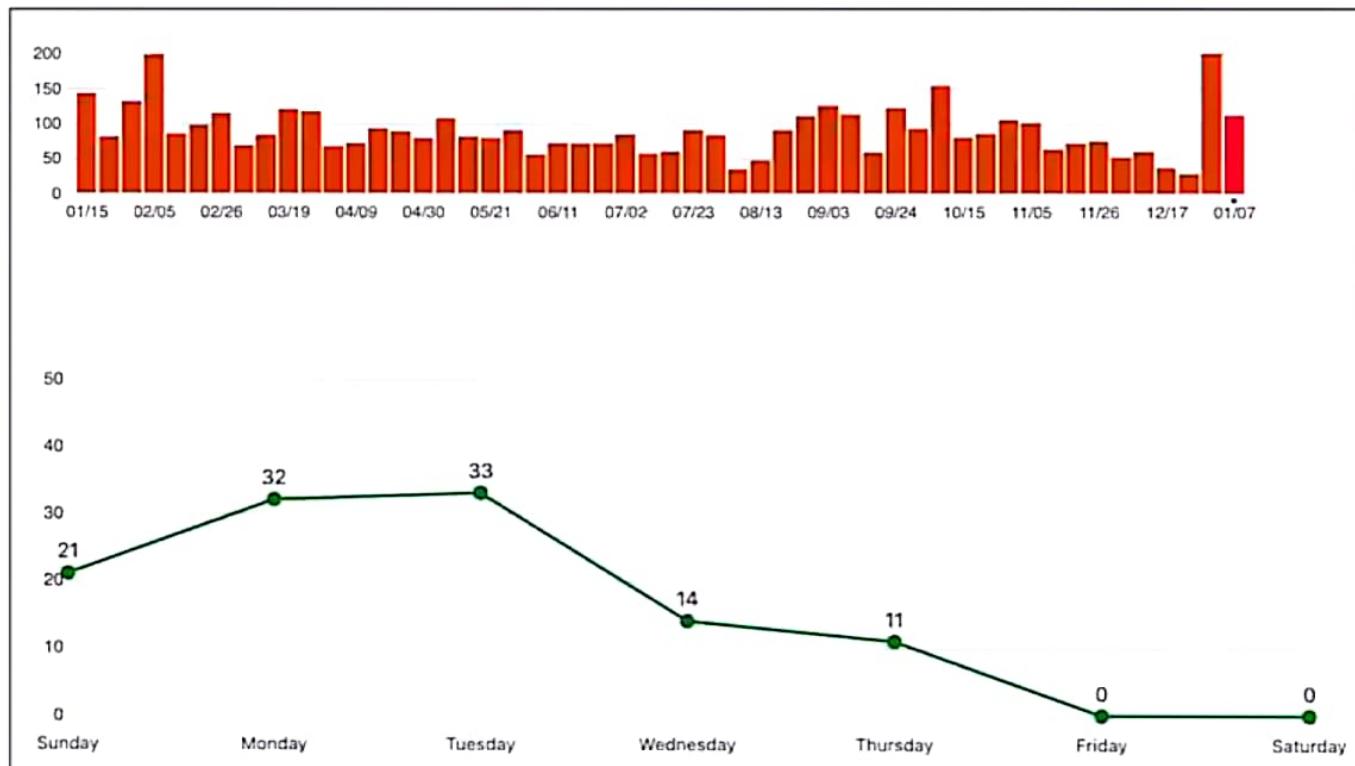
#### Checklist

- 
- ✓ Description
  - ✓ README
  - ✓ Code of conduct
  - ✓ Contributing
  - ✓ License
  - ✓ Security policy
  - Issue templates
  - ✓ Pull request template
  - Repository admins accept content reports

What is [the community profile](#)?

## Commits contains:

- Number of commits per week for the last 52 weeks. (for a year)
- The average number of commits by day of the week for the selected week

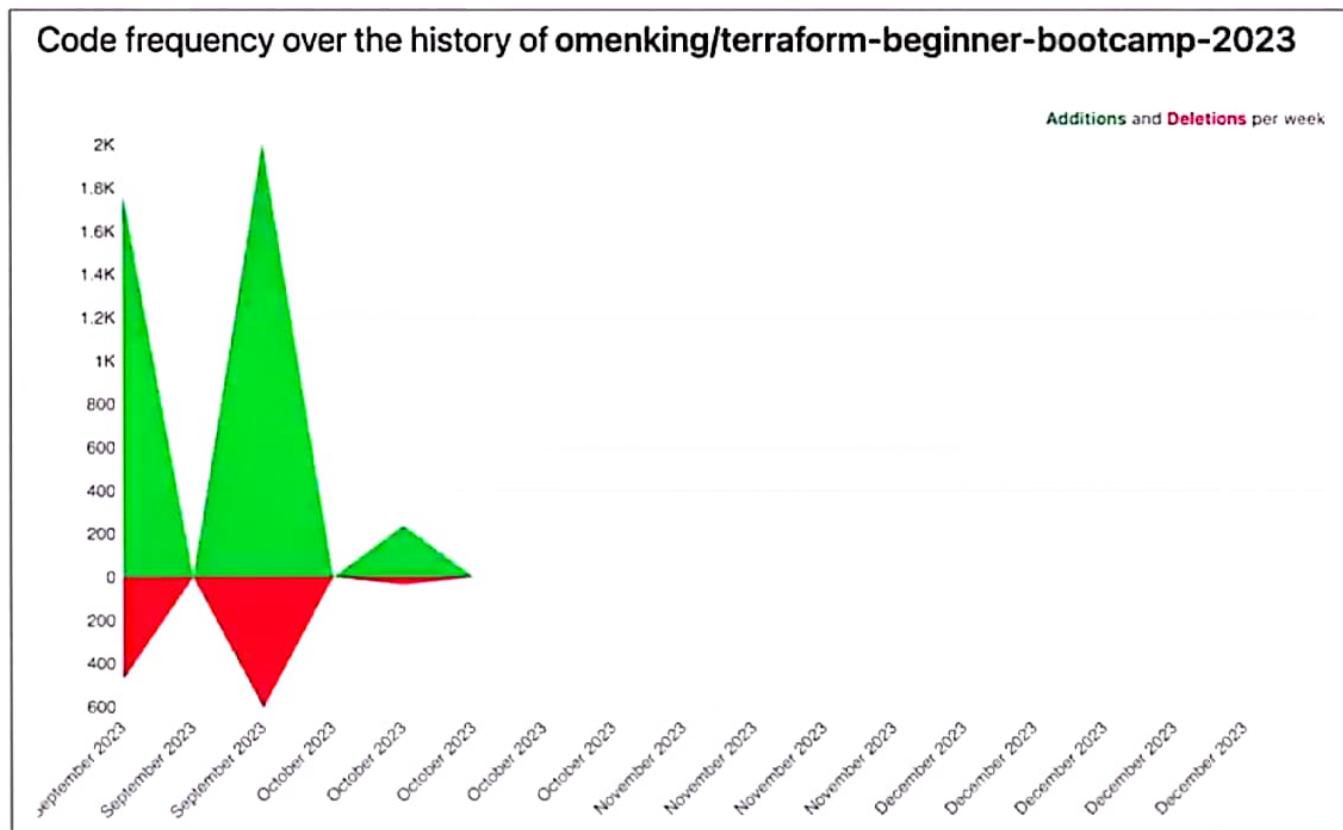


# Repo Insights

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

**Code frequency** contains:

- The amount of additions and deletions of code to a report per month.



If the additions and deletions is too large the information will be able to graph.

# Repo Insights

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

## Dependency graph contains:

- List of dependencies
- List of dependents
- Export Software Bills of Materials (SBOMs)

### Dependency graph

Dependencies      Dependents  Export SBOM

Q Search all dependencies

**addressable** 2.8.6  
Detected automatically on Jan 09, 2024 (RubyGems) · Gemfile.lock

**amq-protocol** 2.3.2  
Detected automatically on Jan 09, 2024 (RubyGems) · Gemfile.lock · MIT

**ast** 2.4.2  
Detected automatically on Jan 09, 2024 (RubyGems) · Gemfile.lock · MIT

**aws-eventstream** 1.3.0

Security and compliance teams increasingly request software bills of materials (SBOMs) to identify the open source components of their software projects, assess their vulnerability to emerging threats, and verify alignment with license policies.

# Repo Insights

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

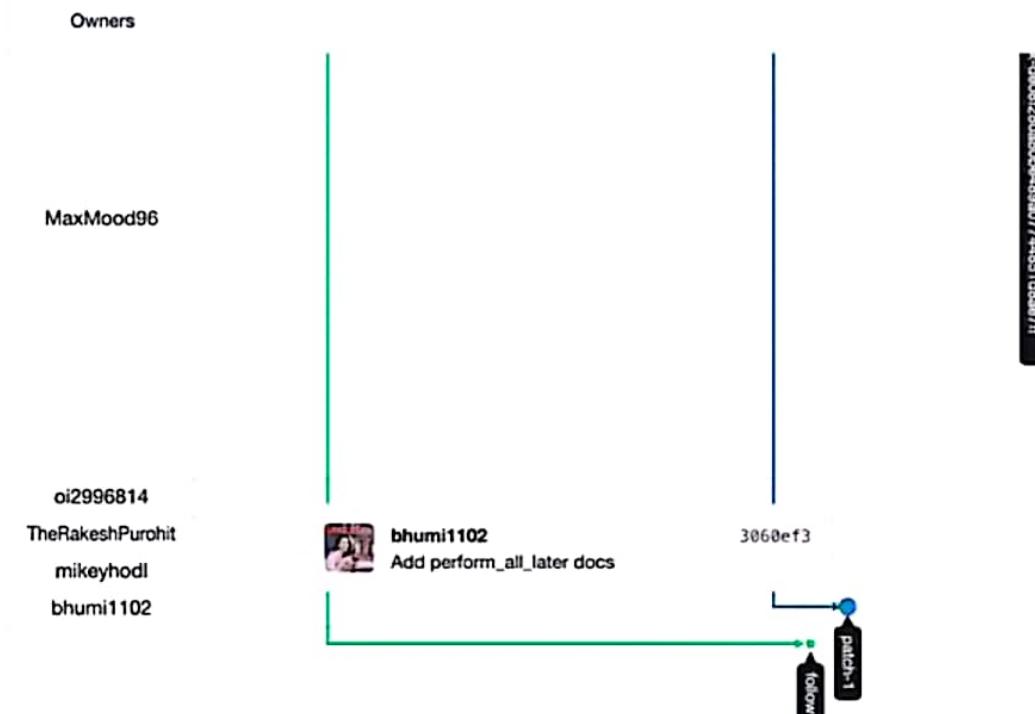
## Network contains:

- 100 most recently pushed forks
  - You can read the commits to determine the difference of these forks.

### Network graph

Timeline of the most recent commits to this repository and its network ordered by most recently pushed to.

The repository network shows the 100 most recently pushed forks.



Keyboard shortcuts available 

# Repo Insights

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

## Forks contains:

- Contains a list of filterable forks

Switch to tree view

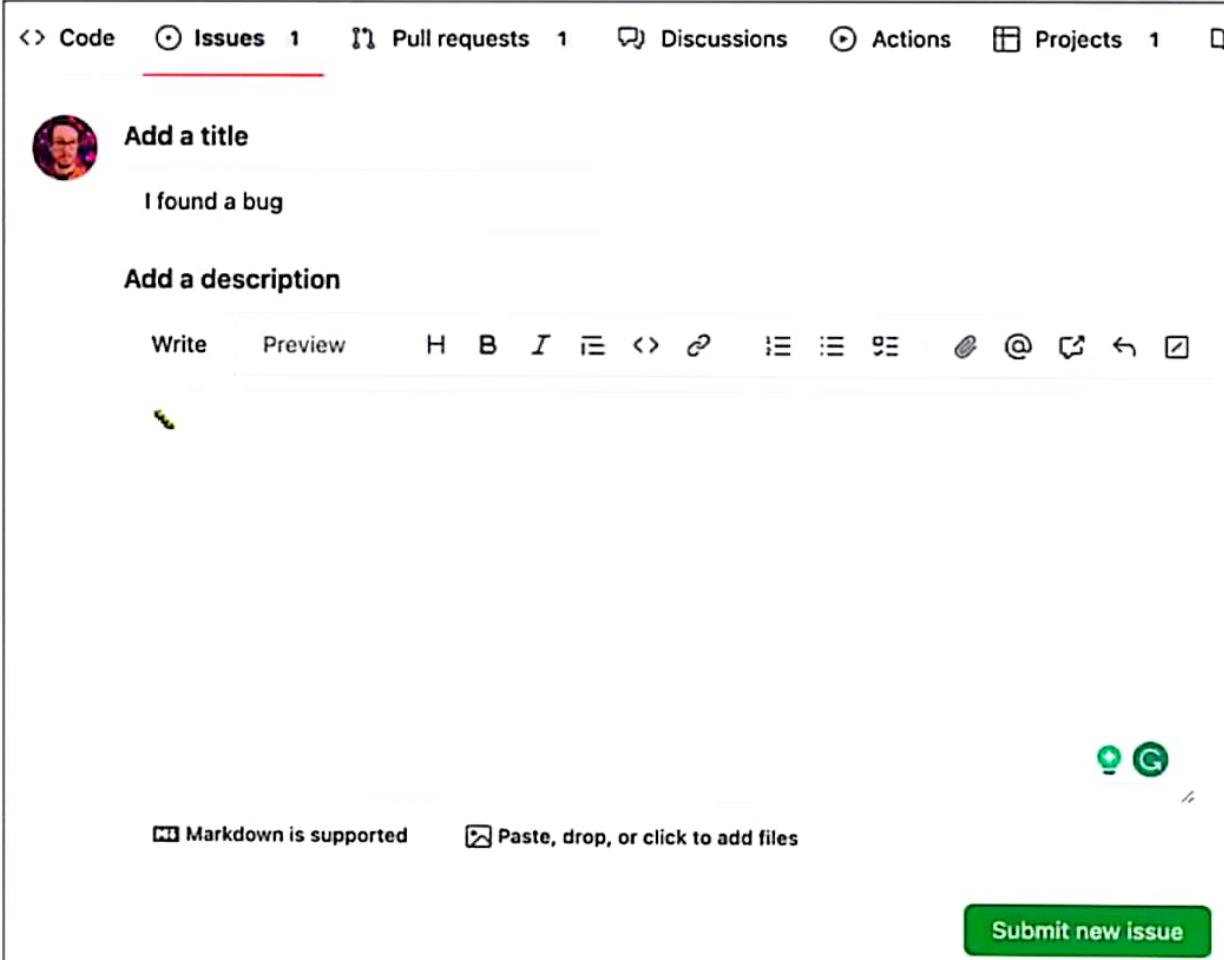
**Forks**      Period: **2 years** ▾      Repository type: **Active** ▾      Sort: **Most starred** ▾      Defaults Saved

|                                                                                                                       |                                                                                      |                                                                                      |                                                                                       |                      |                      |
|-----------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|----------------------|----------------------|
|  62                                  |  0  |  0  |  0  | Created 12 years ago | Updated 3 months ago |
|  37                                  |  0  |  4  |  0  | Created 10 years ago | Updated 7 months ago |
|  29                                  |  0  |  0  |  1  | Created 16 years ago | Updated 6 months ago |
|  14                                 |  0 |  0 |  1 | Created 8 years ago  | Updated 9 hours ago  |
|  <a href="#">sununqa07</a> / rails |                                                                                      |                                                                                      |                                                                                       |                      |                      |

# Creating Issues

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

Opening an Issue is straight forward.  
You enter a **title** and a markdown  
supported **description**



The screenshot shows the GitHub Issues creation page. At the top, there are navigation links: Code, Issues (with 1), Pull requests (with 1), Discussions, Actions, Projects (with 1), and a search bar. Below these are two input fields: 'Add a title' containing 'I found a bug' and 'Add a description' containing 'Write Preview H B I E <> C E E % @ ↵ ↶ ↷ ↸'. A red arrow from the previous slide points to the 'Add a title' field. At the bottom right are two buttons: 'Markdown is supported' and 'Paste, drop, or click to add files', followed by a large green 'Submit new issue' button.

You can use the **GitHub CLI** to create Issues.

```
Create an issue interactively
$ gh issue create
Creating issue in owner/repo
? Title My new issue
? Body [(e) to launch nano, enter to skip]
http://github.com/owner/repo/issues/1
$
```



# Issues vs Discussions vs Pull Requests

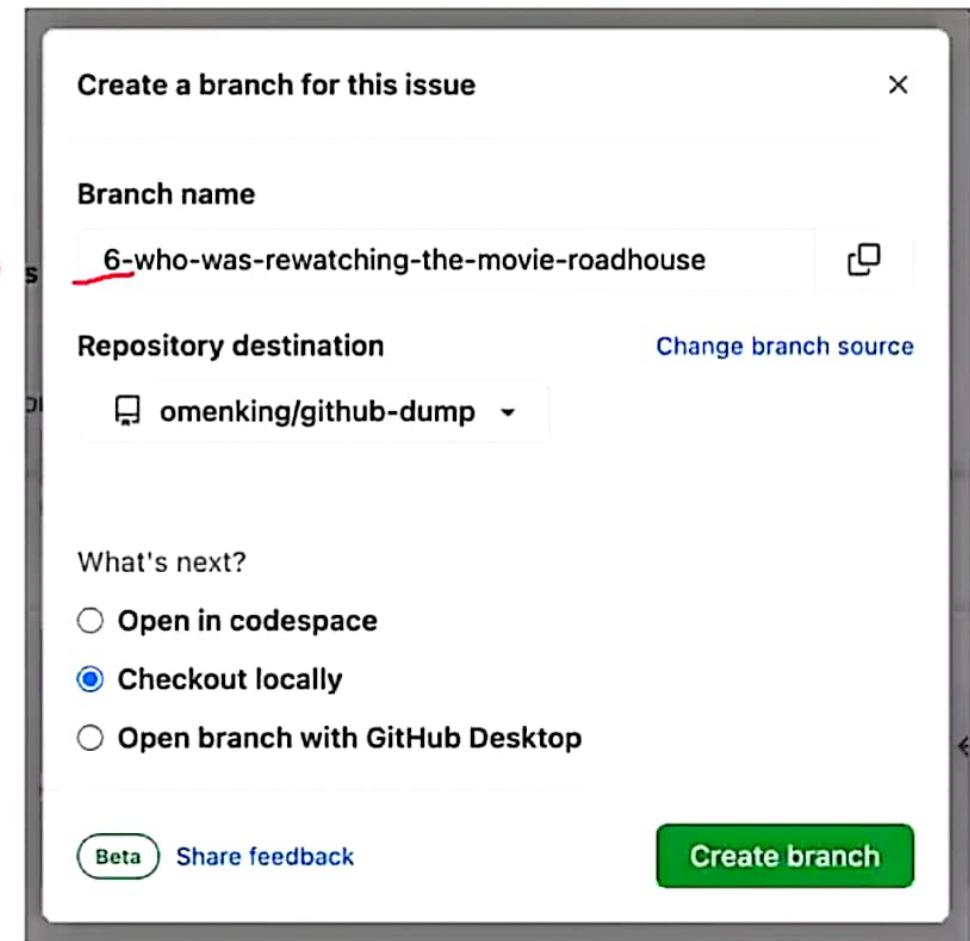
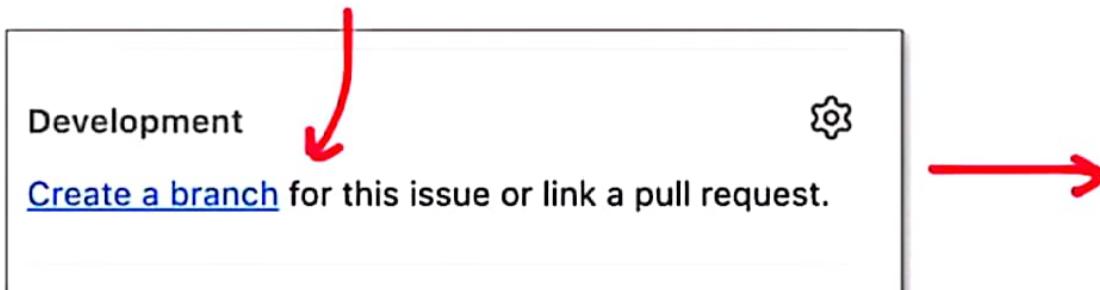
Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

| Issues                                                                                                                                                                                      | Discussions                                                                                                                                                                                                                                                                                                                                                                        | Pull Requests                                                                                                                                                                                       |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>Tracking tasks, bugs, enhancements, and other actionable items.</p> <ul style="list-style-type: none"><li>• Often linked to code changes</li><li>• Can be linked to <b>PRs</b></li></ul> | <p>Facilitating conversations and Q&amp;As about a wide range of topics related to the project.</p> <ul style="list-style-type: none"><li>• Categorized by topics</li><li>• Can be “converted” to issue<ul style="list-style-type: none"><li>• Really just creates an issue with a soft link to the issue</li><li>• <b>Not directly linked to code changes</b></li></ul></li></ul> | <p>Proposing, reviewing, and merging code changes into the codebase.</p> <ul style="list-style-type: none"><li>• Directly linked to code changes</li><li>• Can be linked to <b>Issues</b></li></ul> |

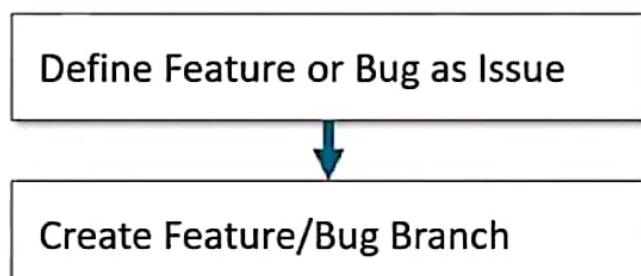
# Create a Branch from an Issue

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

From an Issue you can create  
an **associated branch**.



A common workflow is creating Feature or Bug branches.  
By defining your Issues upfront you quickly create these branches



# Issues Templates

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

Issues Templates are **markdown templates that are preloaded for new Issues**.

They help ensure users creating issues provide all the relevant and expected information

Write Preview H B

```
Steps to reproduce
<!-- (Guidelines for creating a bug report are [available here](https://edgeguides.rubyonrails.org/contributing_to_ruby_on_ra

<!-- Paste your executable test case created from one of the scripts f
(https://edgeguides.rubyonrails.org/contributing_to_ruby_on_rails.ht
```
# Your reproduction script goes here
```

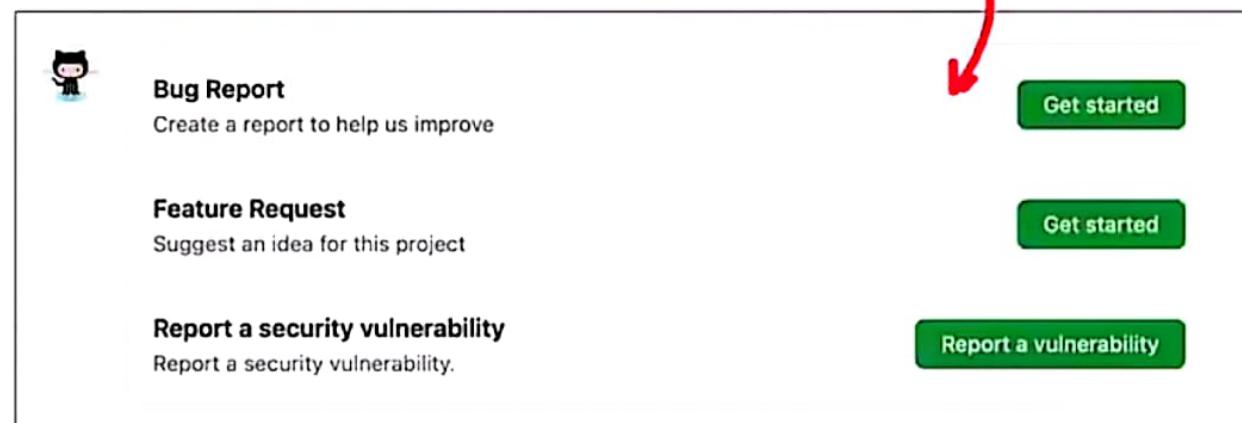
Expected behavior
<!-- Tell us what should happen -->

Actual behavior
<!-- Tell us what happens instead -->

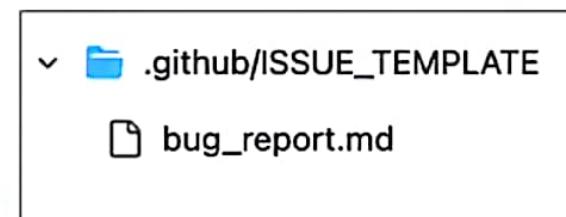
System configuration
Rails version:

Ruby version:
```

You can **create multiple issues templates** to better improve the context of issues:



Issue Templates are stored as markdown files in **.github/ISSUE\_TEMPLATE** folder in you repo



 GitHub has a wizard GUI to easily create Issue Templates

# Issues Forms

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

**Issues Forms** is the evolution of Issues Templates.

You use a **YAML formatted file** to create Issue forms  
for stricter entry of issue information



```
name: Bug Report
description: File a bug report
title: "[Bug]: "
labels: ["bug", "triage"]
projects: ["octo-org/1", "octo-org/44"]
assignees:
- octocat
body:
- type: markdown
 attributes:
 value:
 Thanks for taking the time to fill out
- type: input
 id: contact
 attributes:
 label: Contact Details
 description: How can we get in touch with
 placeholder: ex. email@example.com
validations:
```

## Issue: Bug Report

File a bug report. If this doesn't look right, [choose a different type](#).



[Bug]:

Thanks for taking the time to fill out this bug report!

### Contact Details

How can we get in touch with you if we need more info?

ex. email@example.com

### What happened?

Also tell us, what did you expect to happen?

A bug happened!

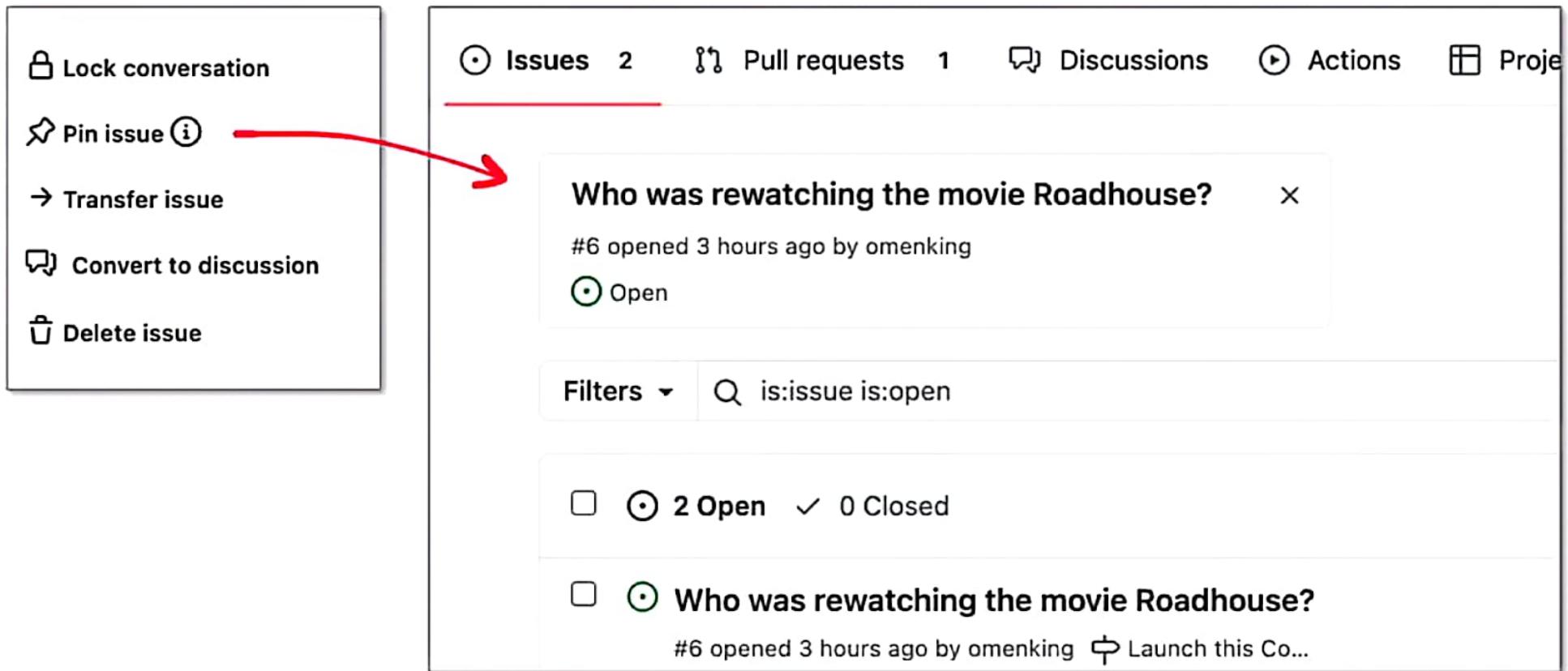
### Version

What version of our software are you running?

# Pinning Issues

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

Up to 3 issues can be pinned so they appear at the top of the Issue page.



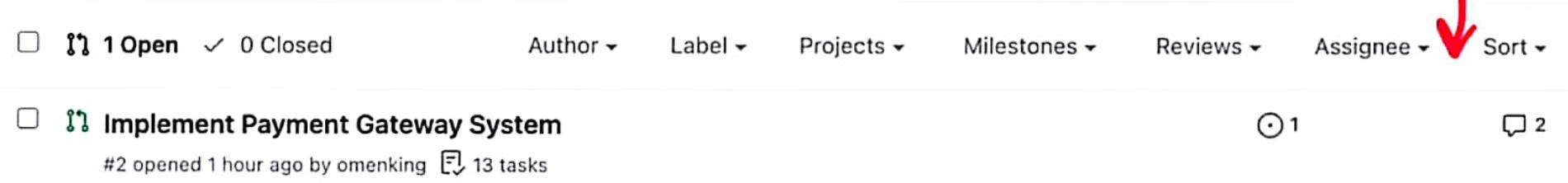
The screenshot shows a GitHub Issues page with the following elements:

- Left sidebar:** A vertical list of actions:
  - Lock conversation
  - Pin issue** (highlighted with a red arrow)
  - Transfer issue
  - Convert to discussion
  - Delete issue
- Header:** Issues 2, Pull requests 1, Discussions, Actions, Projects
- Issue card:** Who was rewatching the movie Roadhouse?
  - #6 opened 3 hours ago by omenking
  - Open
- Filters:** is:issue is:open
- Filter options:**
  - 2 Open ✓ 0 Closed
  - Who was rewatching the movie Roadhouse?
    - #6 opened 3 hours ago by omenking
    - Launch this Co...

# Pull Requests

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

A Pull Request (PR) is a **formal process to put forth changes**, that can be manually or automatically reviewed before its accepted into your base (main) branch.



The screenshot shows a list of pull requests. At the top, there are filters: '1 Open' (unchecked), '0 Closed' (checked), and dropdowns for 'Author', 'Label', 'Projects', 'Milestones', 'Reviews', 'Assignee', and 'Sort'. A red arrow points to the 'Sort' dropdown. Below the filters is a single pull request card:

- Implement Payment Gateway System** #2 opened 1 hour ago by omenking 13 tasks

To the right of the card are icons for reviews (1), comments (2), and other actions.

## Benefits of a Pull Request

- Collaborative Review:** Enhances code quality through team discussions and peer feedback.
- Change Tracking:** Provides a record of code changes and related discussions.
- Automated Testing:** Enables integration with tools for automated checks and tests.
- Controlled Integration:** Manages safe and reviewed merging of code changes.
- Open Source Friendly:** Simplifies contributions and collaboration in open-source projects.

A pull request is not a feature of git, but a workflow. Services like GitHub can automate the Pull Request workflow

# Creating Pull Requests

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

We set the:

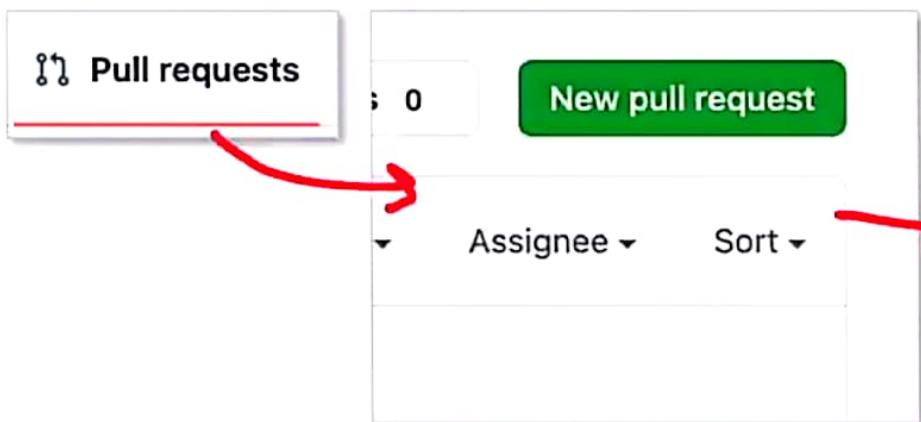
**Base** – who we are going to merge into

**Head** – the changes to pull in

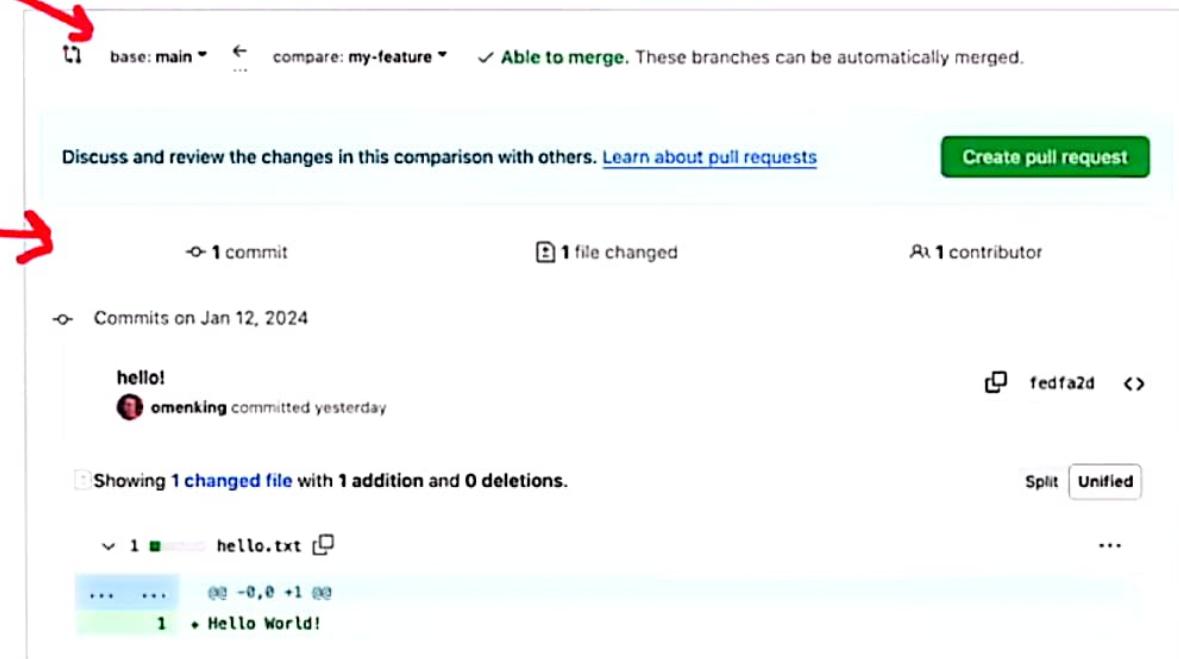
We can use the **Github CLI** to create a pull request from a branch.

```
git checkout -b cool-feature-branch
git commit -am "Your commit message"
git push -u origin cool-feature-branch
gh pr create --base main --head cool-feature-branch
```

In the GitHub repo we  
navigation to **Pull Requests**



We create a **new pull requests**



A screenshot of the GitHub 'Create pull request' interface. It shows a comparison between 'base: main' and 'compare: my-feature'. The status is 'Able to merge'. The pull request summary includes '1 commit', '1 file changed', and '1 contributor'. The commit details show a commit by 'omenking' with the message 'hello!' and hash 'fedfa2d'. The diff view shows a single addition to 'hello.txt' with the content '+ Hello World!'. Navigation buttons 'Split' and 'Unified' are at the bottom right.

base: main ▾ ← ... compare: my-feature ▾ ✓ Able to merge. The

Choose a head ref

X

Discuss and review the

Find a branch

at

Branches

Tags

main

default

✓ my-feature

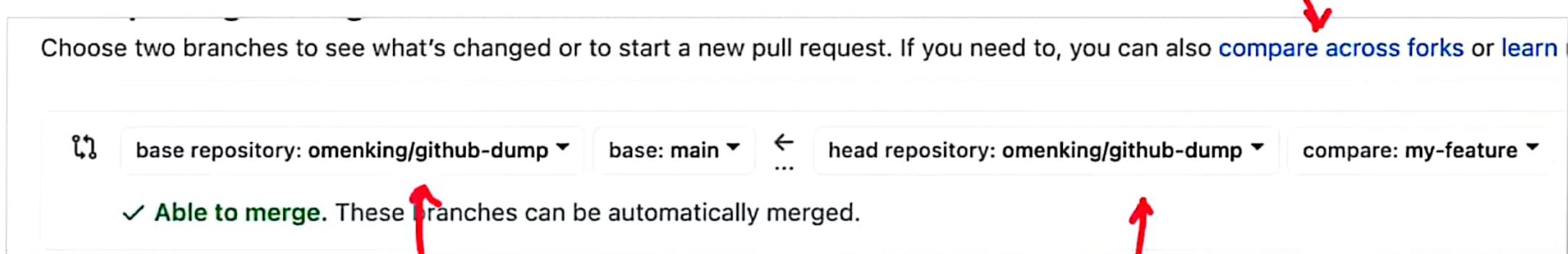
-o Commits on Jan 12,

# Pull Requests – Base and Compare

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

You can also **compare across forks** this is how a fork stays up to date, or how forks can suggest you to accept their changes.

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also compare across forks or learn



base repository: omenking/github-dump ▾ base: main ▾ ↵ ... head repository: omenking/github-dump ▾ compare: my-feature ▾

✓ Able to merge. These branches can be automatically merged.

You'll now be able to choose repo's from forks

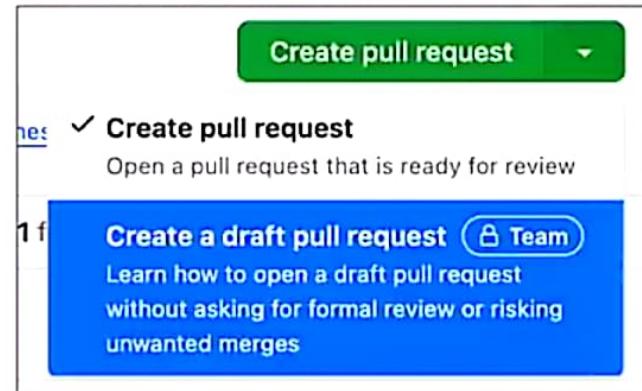
# Draft Pull Requests

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

A Draft Pull Request on GitHub is a feature that **allows you to open a pull request but mark it as a work-in-progress (WIP)**.



- Draft pull requests cannot be merged
- code owners are not automatically requested to review draft pull requests



## Use case for Draft Pull Requests

- **Indicating Work-in-Progress:** Communicates that the pull request is not ready for review or merging.
- **Preventing Premature Merging:** Ensures incomplete work is not accidentally merged.
- **Facilitating Early Feedback and Collaboration:** Allows for early sharing and discussion of code changes.
- **Continuous Integration Testing:** Enables CI tests during the development process.
- **Transitioning to a Ready State:** Easy switch from draft to ready for final review and merging.
- **Organizing Work and Priorities:** Helps in managing and tracking ongoing work in large projects.

Draft Pull Requests is a feature only for **Github Organizations Teams**

**Link a branch or pull request**

Select a repository to search for branches and pull requests or [create a branch](#)

Search for repositories

 [omenking/github-dump](#) →

omenking/github-dump

← Link a branch, pull request, or [create a branch](#)

Search for branches or pull requests

✓  my-feature  
#2 hello!

Cancel

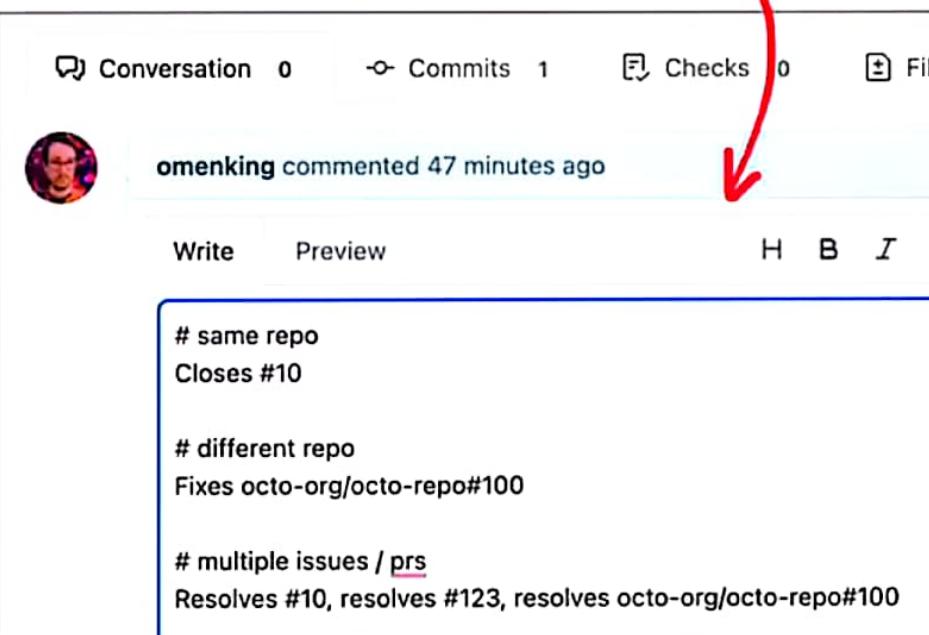
Apply

Within the Issue go to the right

# Linked Activity within a Pull Request

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

You can link a pull request to an issue by using a supported keyword in the **pull request's description** or in a commit message.



The screenshot shows a GitHub pull request interface. At the top, there are statistics: Conversation 0, Commits 1, Checks 0, and Files 0. Below this, a comment from a user named omenking is shown, followed by a red arrow pointing down to the description area. The description contains three examples of linked activity:

- # same repo  
Closes #10
- # different repo  
Fixes octo-org/octo-repo#100
- # multiple issues / prs  
Resolves #10, resolves #123, resolves octo-org/octo-repo#100

## Supported Keywords

- close
- closes
- closed
- fix
- fixes
- fixed
- resolve
- resolves
- resolved



The pull request **must be** on the default branch.

# Pull Request Statuses

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

Open

The default status when a pull request is created. It's open for discussion and review.

Draft

Indicates the pull request is a work-in-progress and not yet ready for review.

Closed

The pull request has been closed without being merged. This status is used when the proposed changes are no longer needed or if the branch has been rejected.

Merged

The pull request's changes have been merged into the target branch.  
This status indicates a successful conclusion of the pull request process.

Changes Requested

This status is used during the review process when a reviewer requests changes before the pull request can be merged.

Review Required

Indicates that the pull request requires a review before it can be merged.  
This status is common in repositories where reviews are a mandatory part of the workflow.

Approved

The pull request has been reviewed and approved for merging by the required number of reviewers.

Conflict

Indicates that there are conflicts between the pull request's branch and the target branch that need to be resolved before merging.

Ready for Review

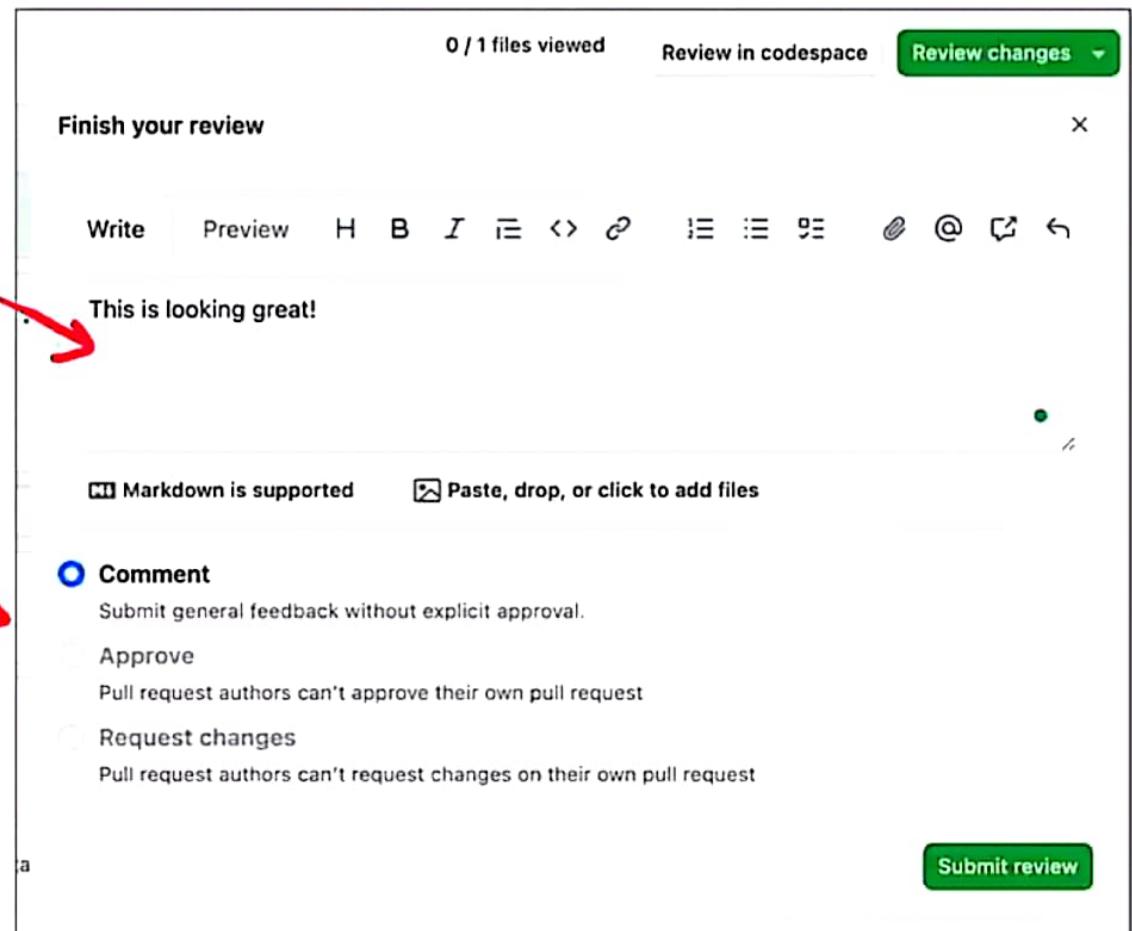
A pull request initially marked as draft can be changed to this status once it's ready for review.

# Commenting in Pull Requests

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

When you comment you are submitting a review to be considered before a Pull Request is being accepted.

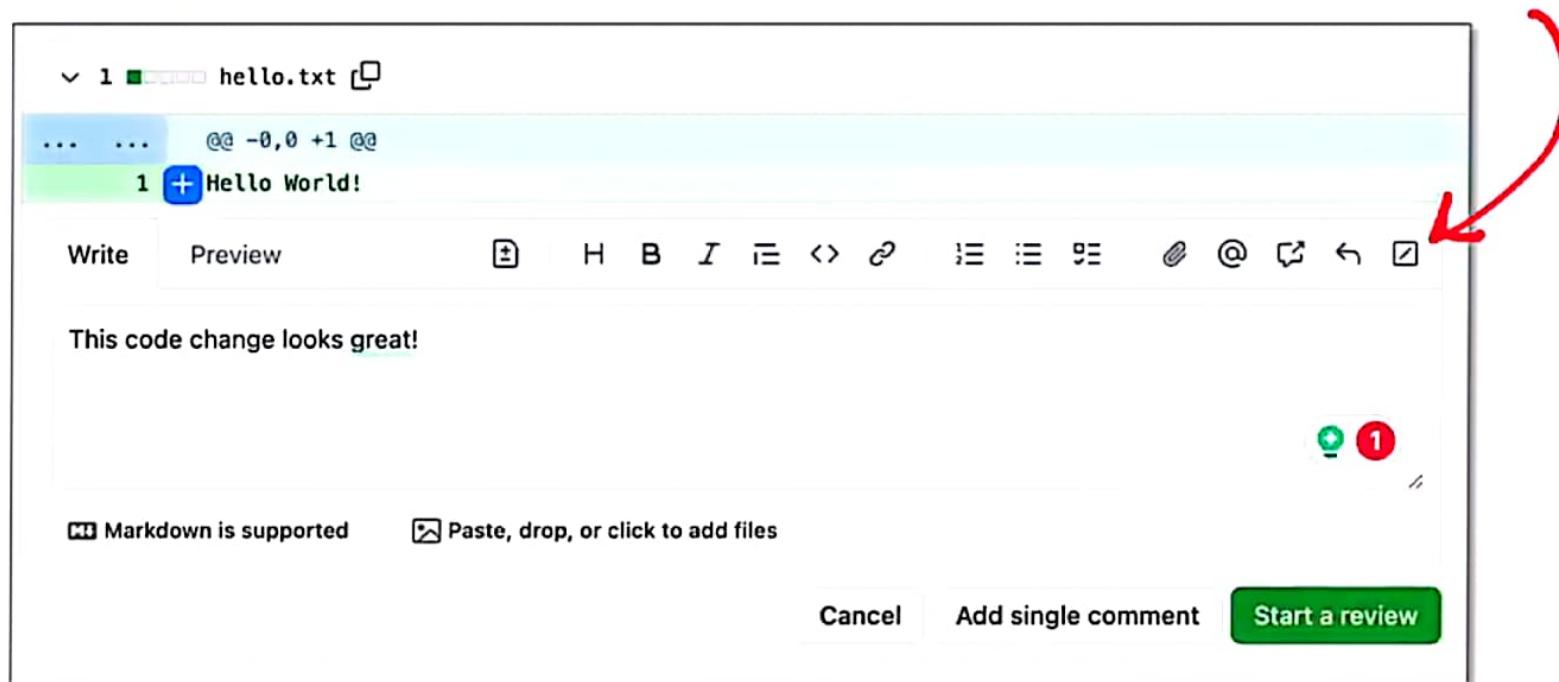
If you are not the author, but also a designated reviewer you can **request changes** or mark the Pull Request as **Approved** so it can be considered for merging.



# Commenting in Pull Requests

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

Under **Files Changed** you can also apply a review on **a very specific line of code**.



# Codereviews with CODEOWNERS file

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

**CODEOWNERS** Github repo specific file to define individuals or teams that are responsible for **specific code** in a repository

```
* @global-owner1 @global-owner2
*.js @js-owner
*.go docs@example.com
*.txt @octo-org/octocats
/build/logs/ @doctocat
docs/* docs@example.com
apps/ @octocat
/docs/ @doctocat
/scripts/ @doctocat @octocat
**/logs @octocat
/apps/ @octocat
/apps/github
/apps/ @octocat
/apps/github @doctocat
```



CODEOWNER files uses a **similar syntax to .gitignore**.

When a pull request is opened that modifies any files matching a pattern in the CODEOWNERS file, GitHub **automatically requests a review from the specified code owners.**

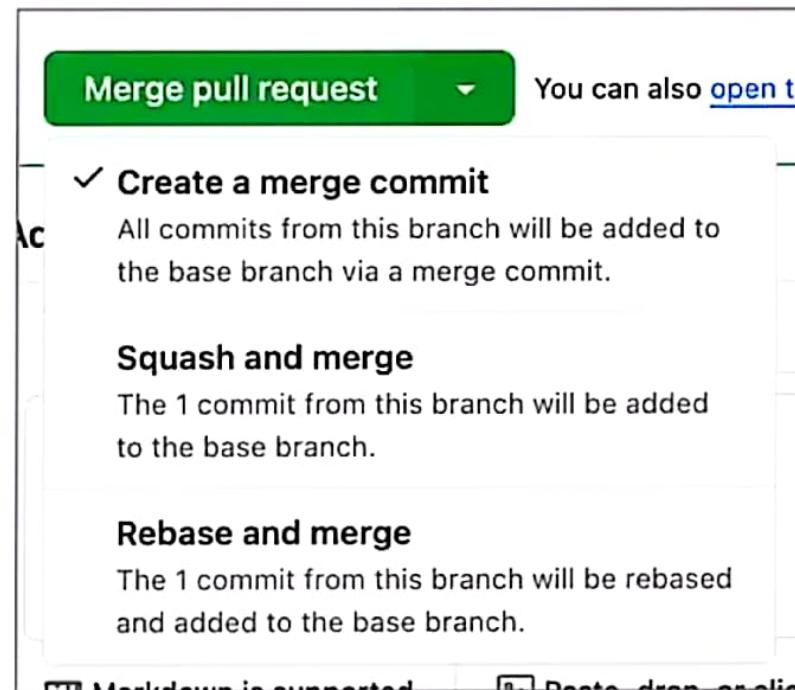
The codeowner files goes in either the project root, .github or docs directory.

# Pull Request Options

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

When merging a pull request there are a few options

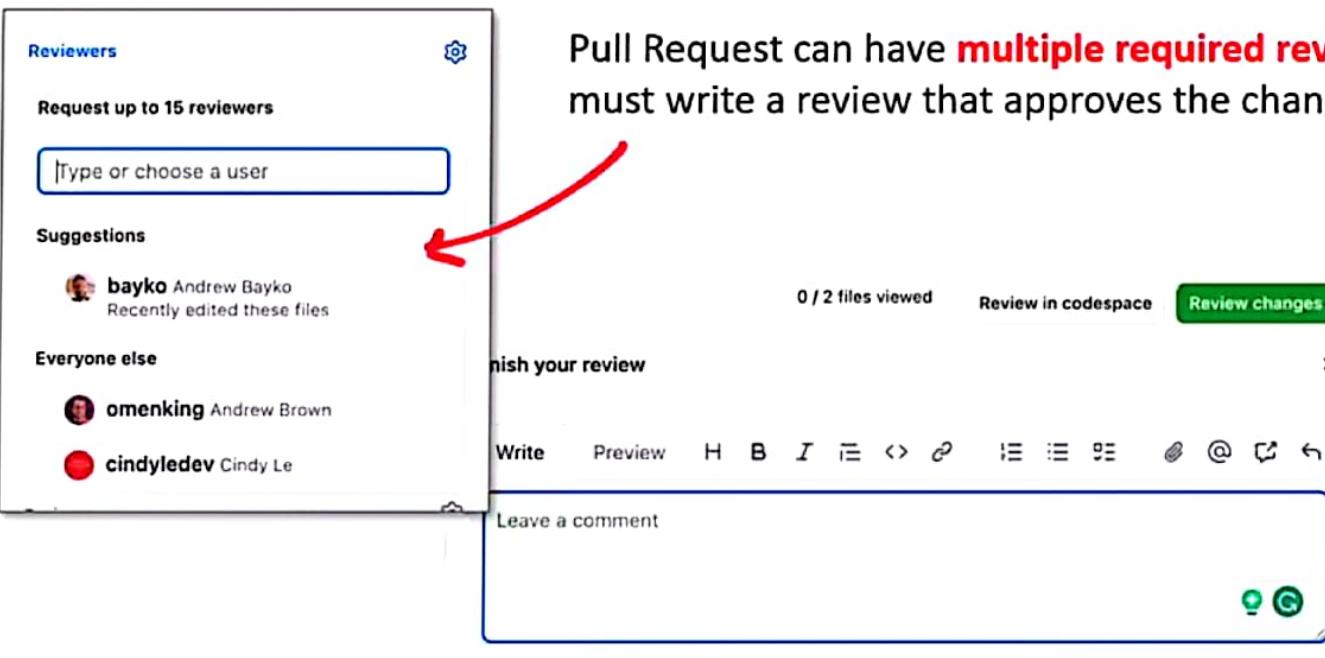
- 1. Create a merge commit**
  - All commits will be added
- 2. Squash and merge**
  - 1 commit will be added
- 3. Rebase and merge**
  - 1 commit will be added and rebased



The use-case depends on your team's workflow. They may prefer only a single commit is added to keep the base branch git tree clean and readable.

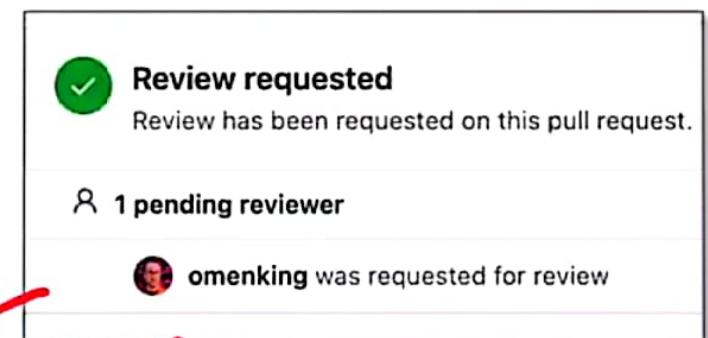
# Required Reviewers

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)



The screenshot shows a GitHub pull request review interface. On the left, a sidebar titled "Reviewers" allows selecting up to 15 reviewers, with a search bar and a list of suggestions including "bayko" and "Everyone else". The main area shows a review form with sections for "Finish your review" and "Leave a comment". Below the comment section are Markdown and file upload options. At the bottom, there are three review actions: "Comment", "Approve" (which is selected), and "Request changes". A red arrow points from the text "The review with Approve is provided" to the "Approve" button.

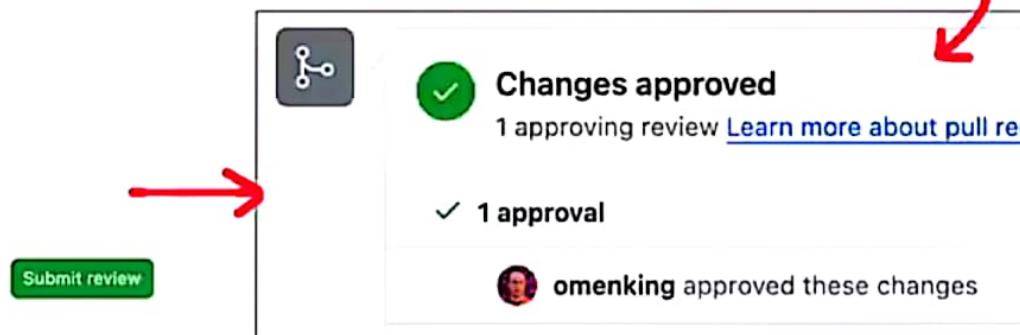
Pull Request can have **multiple required reviewers** who must write a review that approves the changes.



A screenshot of a GitHub notification card. It displays a checkmark icon and the text "Review requested". Below it, it says "Review has been requested on this pull request." and "1 pending reviewer". A user named "omenking" is listed as requested for review. A red arrow points from the text "The review is expected" to this notification.

The review is expected

The review with **Approve** is provided



A screenshot of a GitHub notification card. It displays a checkmark icon and the text "Changes approved". Below it, it says "1 approving review [Learn more about pull requests](#)" and "1 approval". A user named "omenking" is listed as having approved the changes. A red arrow points from the text "The changes are approved passing the check for the repo" to this notification.

The changes are approved passing the check for the repo

# Pull Request Templates

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

**Pull Request Templates** are similar to Issue Templates

They will populate the pull request textarea with the specified template

You create a file in the  
**.github/pull\_request\_template.md**



Add a description

Write Preview H B I B

```

name: Pull Request Checklist
about: pull request checklist

Pull Request Template

Description
Please include a summary of the change and which issue is fixed. Incl dependencies that are required for this change.

Fixes # (issue number)
Markdown is supported Paste, drop, or click to add files
```



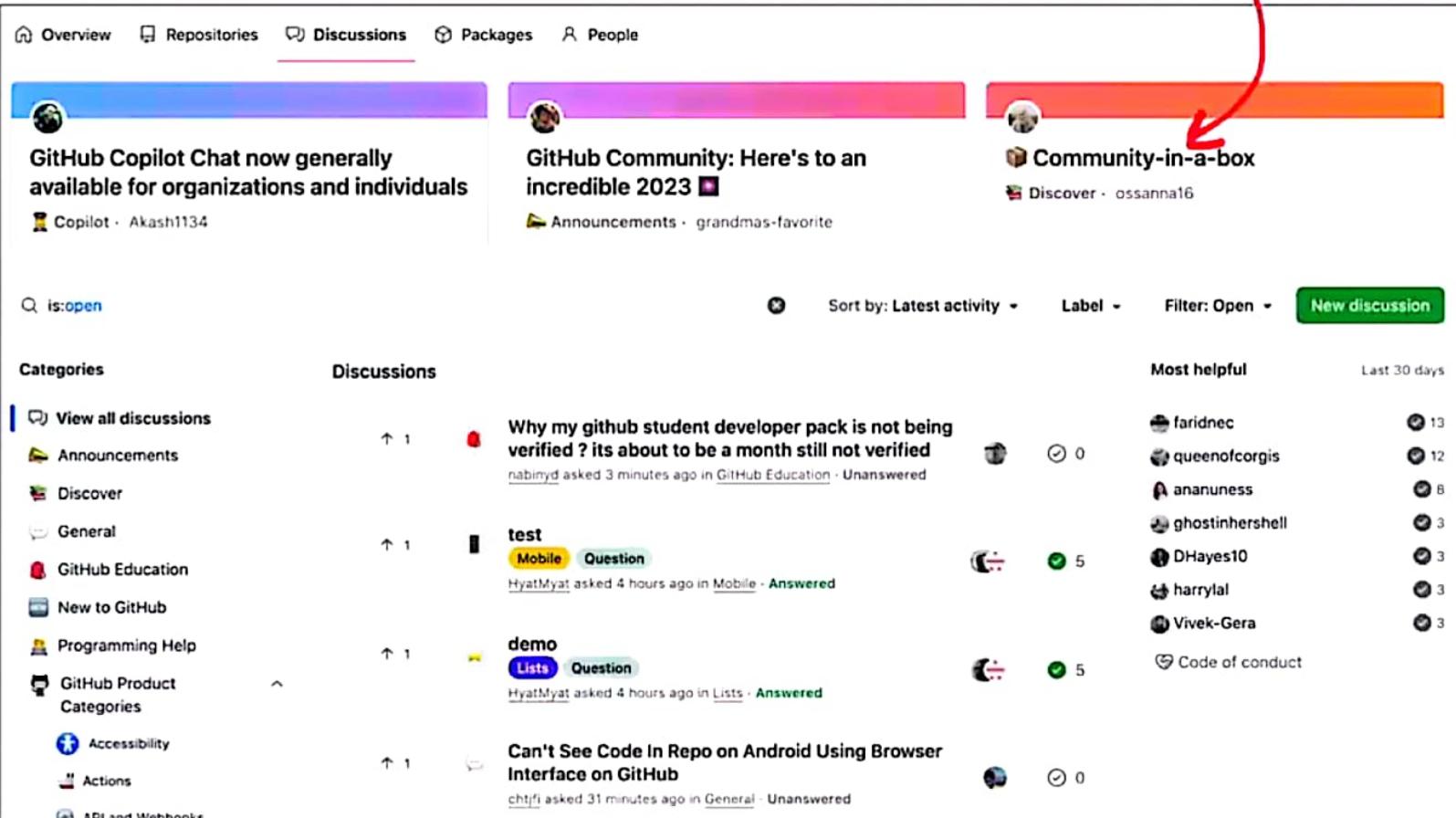
GitHub kind-of supports multiple Pull Requests Templates but you have to assemble your own URL with a querystrings, so **its not as convenient to use as multiple Issue templates.**

# Github Discussions

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

GitHub Discussions is **community communication tool** for your public or private GitHub Repos.

- Threaded Conversations
- Categorization
- Community Interaction
- Markdown Support
- Pin Discussions
- Conversation Polls
- Discussion Voting
- Conversion to Issues
- Notifications
- Searchable and Linkable
- GitHub Integration



A screenshot of the GitHub Discussions interface. At the top, there are four pinned posts:

- GitHub Copilot Chat now generally available for organizations and individuals** by Copilot · Akash1134
- GitHub Community: Here's to an incredible 2023** by Announcements · grandmas-favorite
- Community-in-a-box** by Discover · ossanna16

The main area shows a search bar with "is:open" and filters for "Sort by: Latest activity", "Label", and "Filter: Open". A green button at the bottom right says "New discussion".

**Categories** on the left include:

- View all discussions
- Announcements
- Discover
- General
- Github Education
- New to GitHub
- Programming Help
- Github Product Categories
- Accessibility
- Actions
- API and Webhooks

**Discussions** listed on the right:

- Why my github student developer pack is not being verified ? its about to be a month still not verified** by nabimyd · 3 minutes ago in GitHub Education - Unanswered
- test** by HyatMyat · 4 hours ago in Mobile - Answered
- demo** by HyatMyat · 4 hours ago in Lists - Answered
- Can't See Code In Repo on Android Using Browser Interface on GitHub** by chtfji · 31 minutes ago in General - Unanswered

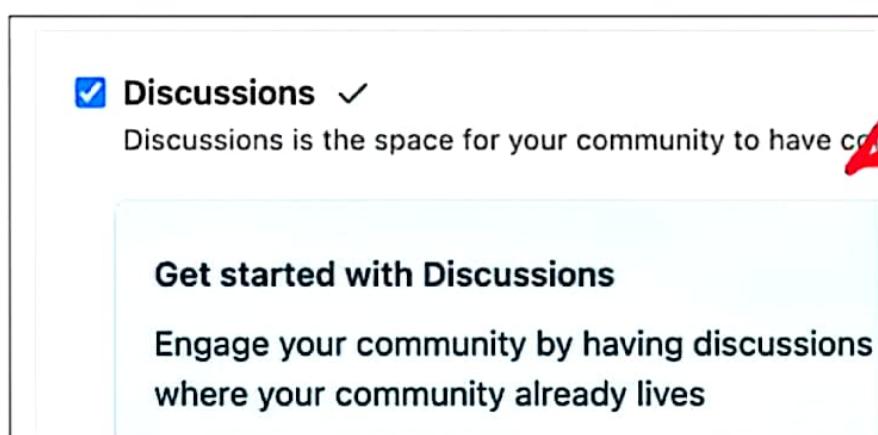
Each discussion entry includes a user icon, name, timestamp, category, and a "Most helpful" section with upvotes and names.

A red arrow points from the text "Community-in-a-box" in the pinned posts to the "Community-in-a-box" pinned post itself.

# Discussions

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

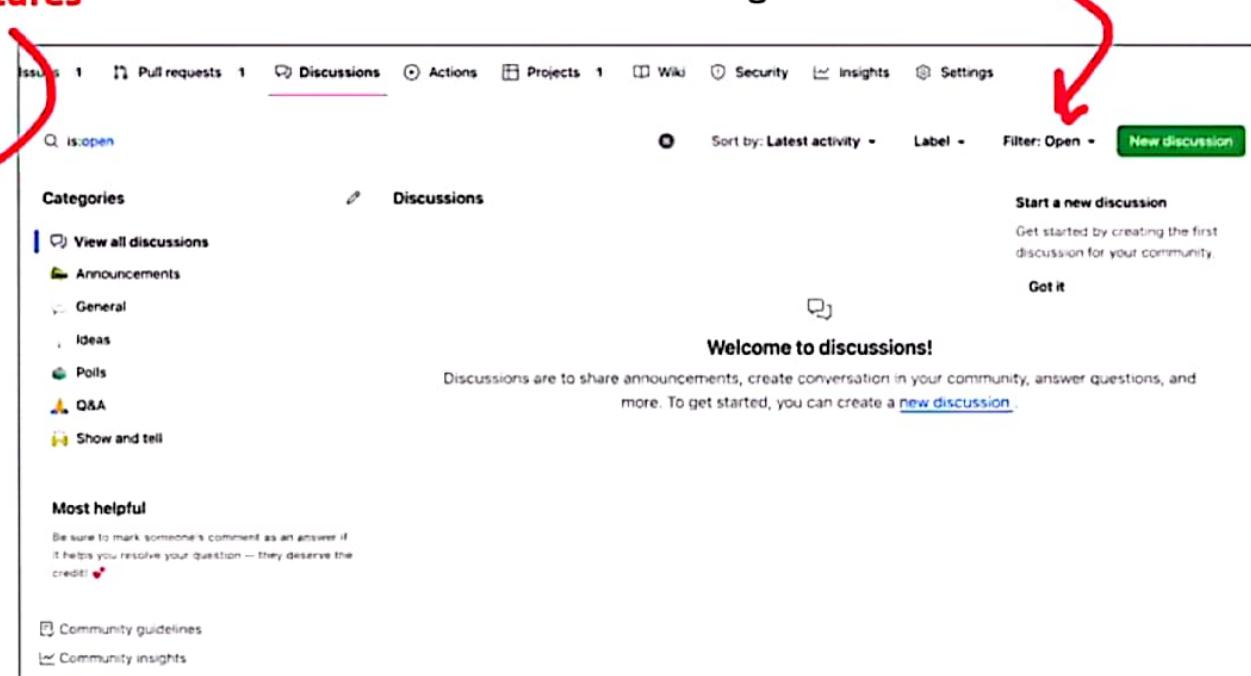
Turn on Discussions under your **Github Repo Features**



**Discussions ✓**  
Discussions is the space for your community to have conversations.

**Get started with Discussions**  
Engage your community by having discussions where your community already lives

A **Discussions tab** will appear in your GitHub Repo.  
You'll have some default categories available.



Issues 1 Pull requests 1 Discussions Actions Projects 1 Wiki Security Insights Settings

Q Is open Sort by: Latest activity Label Filter: Open New discussion

Categories Discussions Start a new discussion

- View all discussions
- Announcements
- General
- Ideas
- Polls
- Q&A
- Show and tell

Welcome to discussions!

Discussions are to share announcements, create conversation in your community, answer questions, and more. To get started, you can create a [new discussion](#).

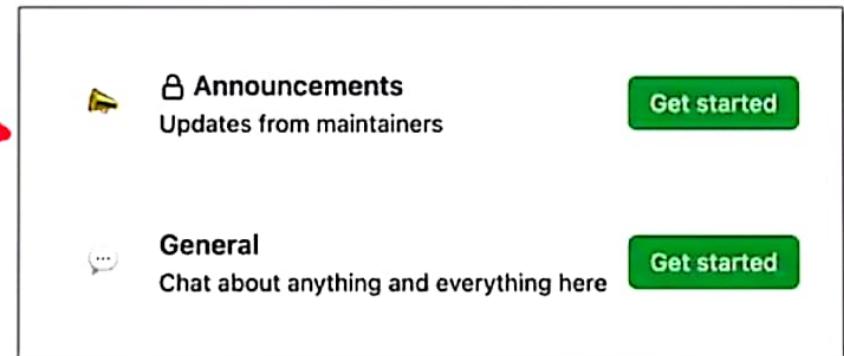
Community guidelines Community insights

# Discussion Options

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

When someone creates a discussion they  
**need to choose a Category.**

GitHub provides some Categories but  
Github repo owners **can create their own.**



## Discussion Format

- Open-ended discussion**  
Enable your community to have conversations, tips and tricks or just chatting.
- Question / Answer**  
Enable your community to ask questions and receive answers.
- Announcement**  
Share updates and news with your community, but anyone can comment.
- Poll**  
Gauge interest, vote, and interact with your community.

When you create a new category you will choose a **Discussion Format**:

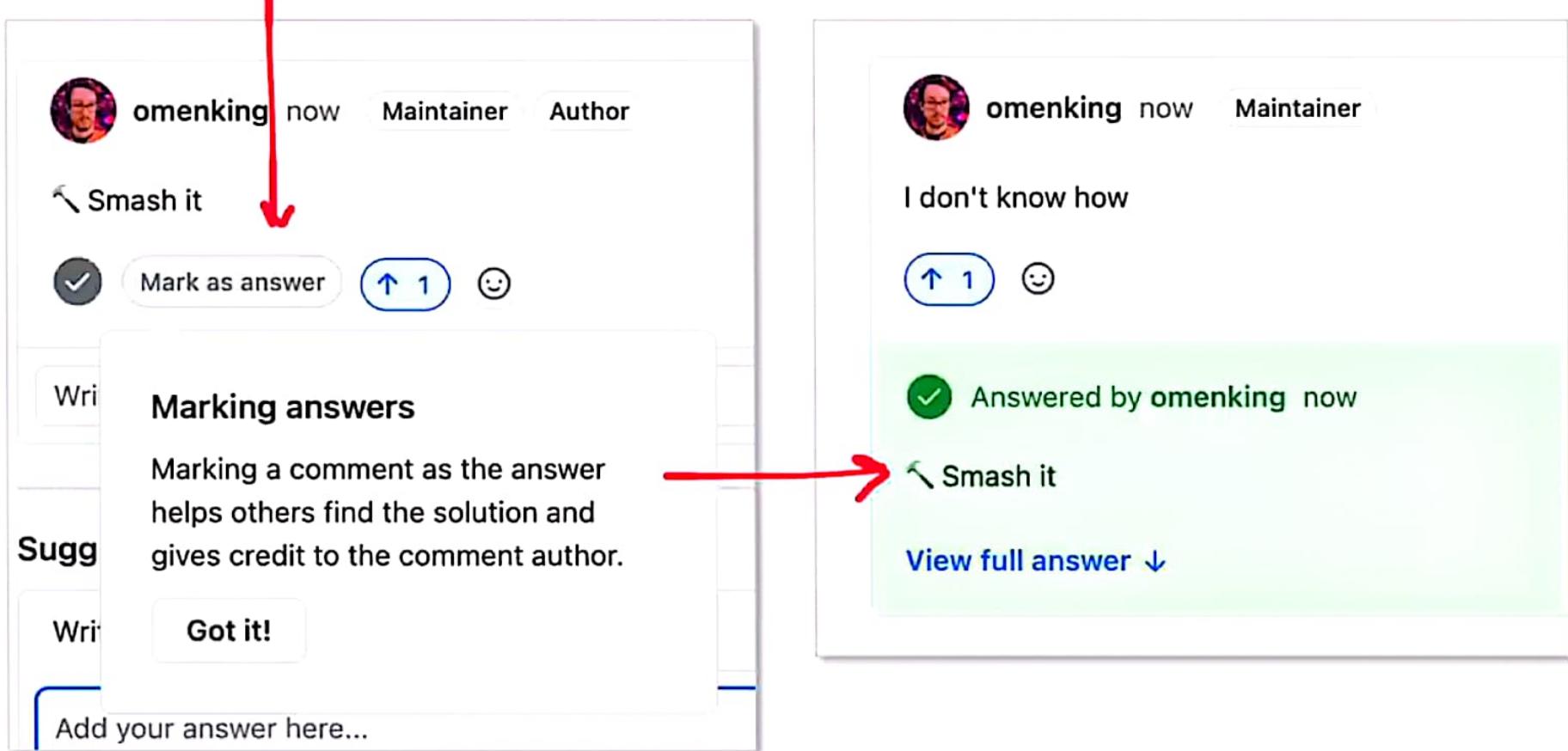
- **Open-ended discussion** — no voting of best answer, just open discussions
- **Question / Answer** — vote on the best suggested answer.
- **Announcement** — admins or maintainers can only post, anyone can reply
- **Poll** — Create Poll and let members vote

**1 =** Categories can be grouped into **sections**  
**2 =**

# Marking Answers in a Discussion

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

For Q&A Discussion Formats you can **mark an answer** as the recommended answer regardless of upvotes to help direct which question you think is the most credible.



The image shows two screenshots of a Q&A discussion interface. The left screenshot shows a comment by user 'omenking' with a red arrow pointing to the 'Mark as answer' button. The right screenshot shows the same comment with a green checkmark indicating it has been marked as the answer, and a red arrow pointing to the 'Smash it' button below it.

omenking now Maintainer Author

Smash it

Mark as answer ↑ 1 ☺

Wri

**Marking answers**

Marking a comment as the answer helps others find the solution and gives credit to the comment author.

Sugg

Wri Got it!

Add your answer here...

I don't know how

↑ 1 ☺

Answered by omenking now

Smash it

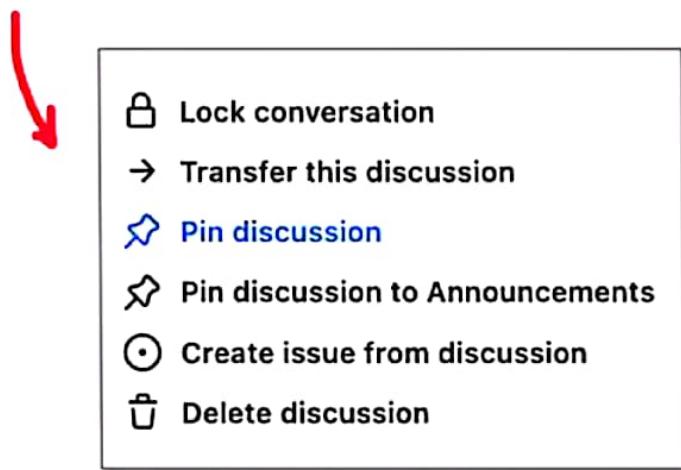
View full answer ↓

# Pinning Discussions

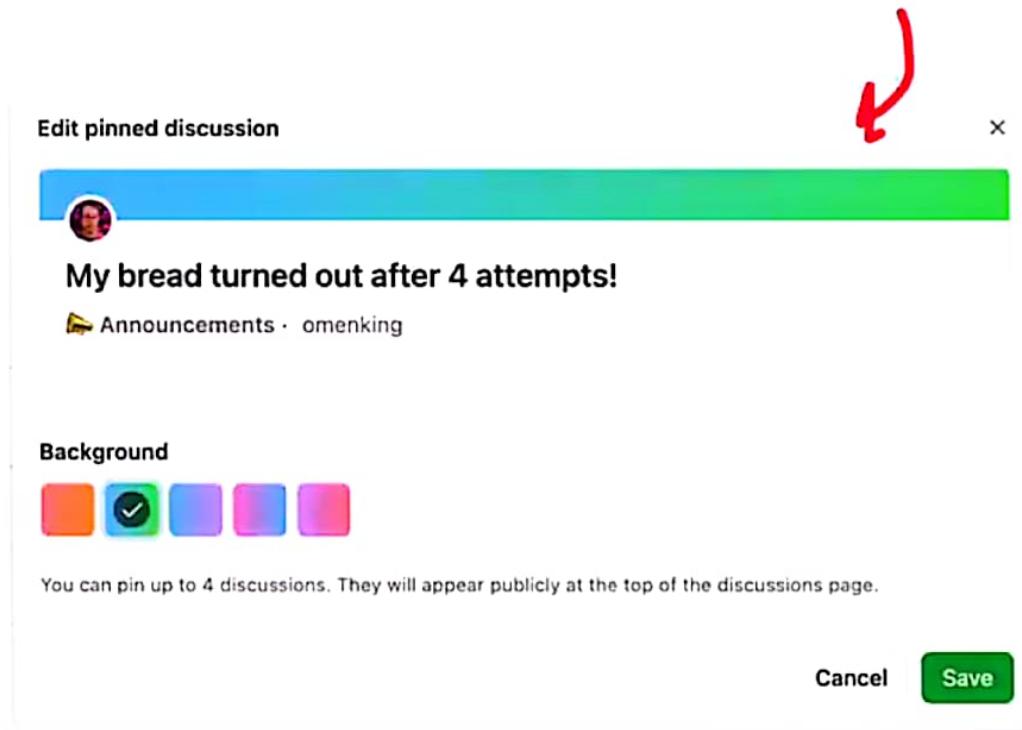
Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

**Pinning Discussions** puts discussions fixed at the top of the community page.

You can **pin** a discussion the Discussions home page or for a specific category



You can **customize** the look of a pinned discussion.



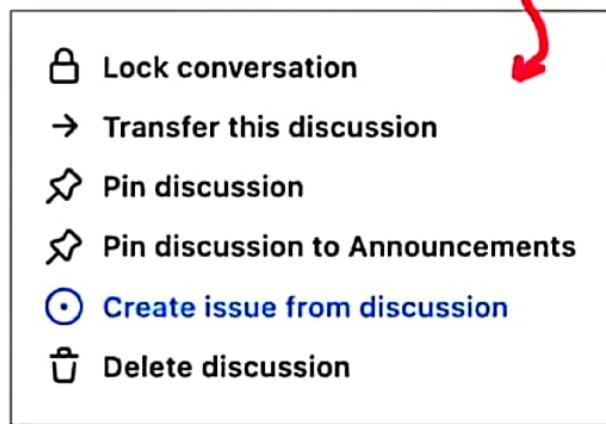
A screenshot of a "Edit pinned discussion" dialog. It shows a pinned discussion card with the title "My bread turned out after 4 attempts!" and the source "Announcements · omenking". Below the card, there's a "Background" section with five color swatches (red, green, blue, purple, pink) and a note: "You can pin up to 4 discussions. They will appear publicly at the top of the discussions page." At the bottom right are "Cancel" and "Save" buttons.

# Convert Discussion to an Issue

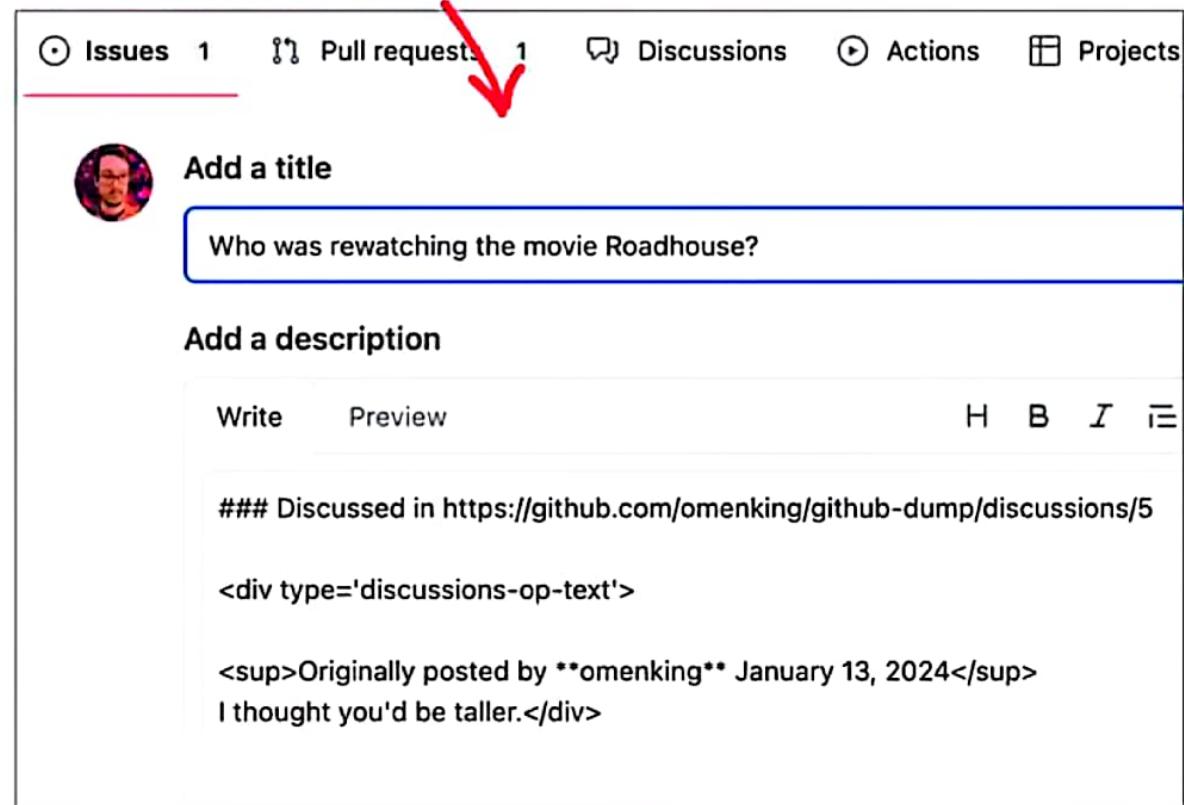
Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

You can "**convert**" a Discussion to an Issue

In the discussion choose to **create issue**.



It will **prepopulate an Issue** and place a Discussion link in the description



A screenshot of a GitHub issue creation interface. The title field contains "Who was rewatching the movie Roadhouse?". The description field contains the following text:

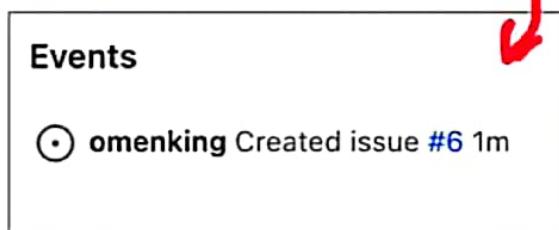
```
Discussed in https://github.com/omenking/github-dump/discussions/5

<div type='discussions-op-text'>

^{Originally posted by **omenking** January 13, 2024}
I thought you'd be taller.</div>
```

Under the **Discussion's Events**

it will have a link to the issue



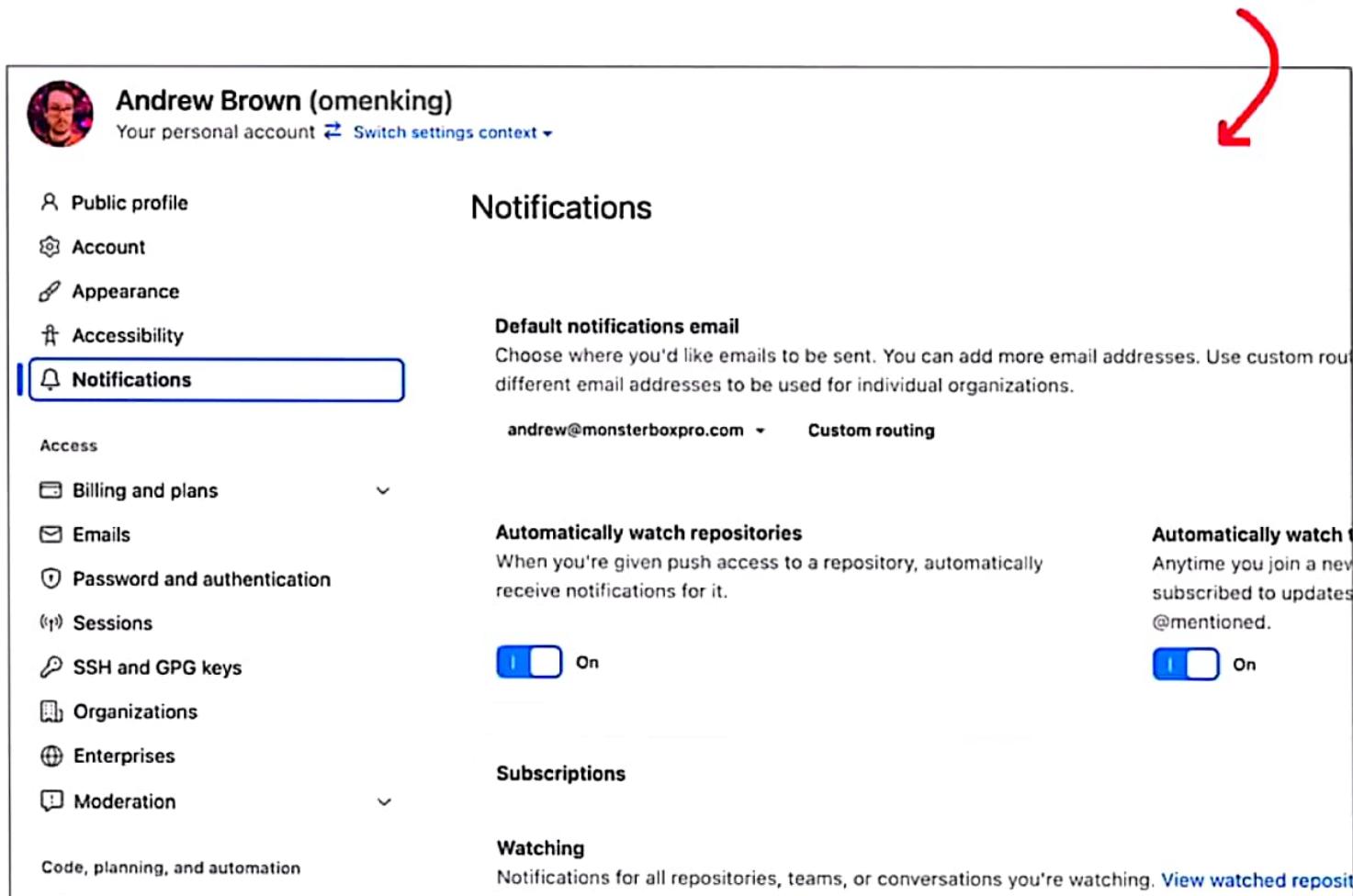
A screenshot of the "Events" section for a discussion. It shows one event:

- omenking Created issue #6 1m

# Notification Configuration Options

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

In your account settings you have a Notifications tab to configure **notifications options**.



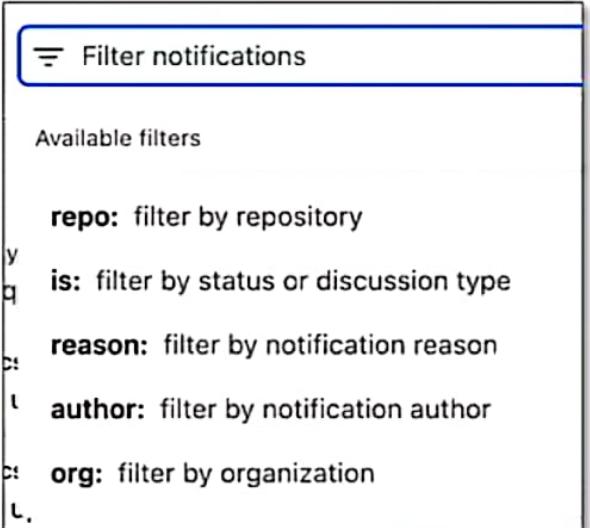
The screenshot shows the GitHub account settings page for user Andrew Brown (omenking). The left sidebar lists various settings categories, and the 'Notifications' tab is highlighted with a blue border. The main content area is titled 'Notifications' and contains several configuration sections:

- Default notifications email:** Set to andrew@monsterboxpro.com, with a 'Custom routing' link.
- Automatically watch repositories:** Described as receiving notifications for repositories you're given push access to. A toggle switch is set to 'On'.
- Automatically watch teams:** Described as receiving notifications for teams you join. A toggle switch is set to 'On'.
- Subscriptions:** Described as notifications for repositories, teams, or conversations you're watching. A 'View watched repositories' link is provided.
- Watching:** Described as notifications for all repositories, teams, or conversations you're watching.

# Notification Filters Options

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

You can filter notifications using a **filter syntax**:

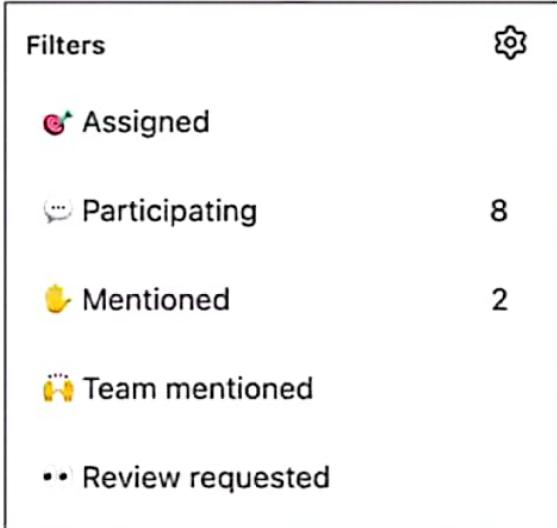


Filter notifications

Available filters

- repo: filter by repository
- is: filter by status or discussion type
- reason: filter by notification reason
- author: filter by notification author
- org: filter by organization

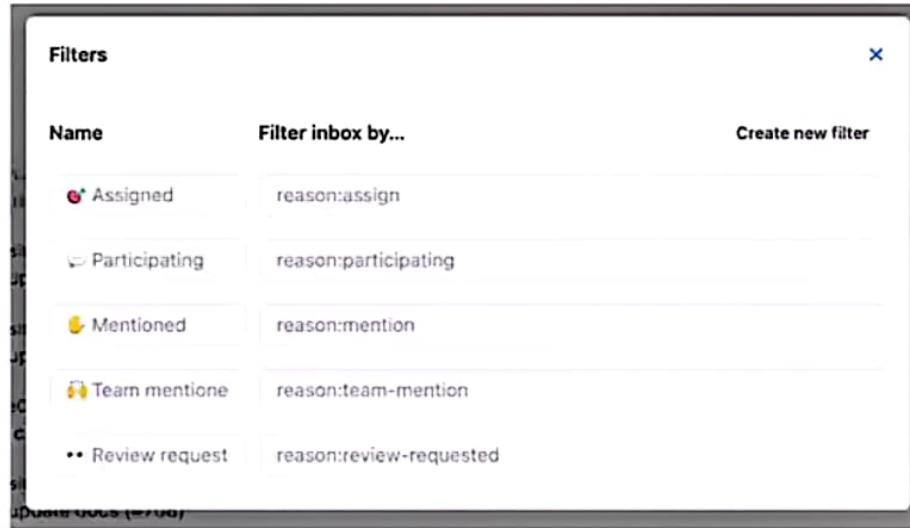
There are **predefined saved filters**



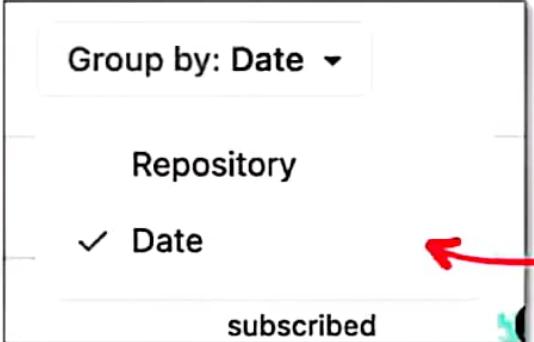
Filters

- Assigned
- Participating 8
- Mentioned 2
- Team mentioned
- Review requested

You can **create more saved filters** using the filter syntax



| Name           | Filter inbox by...      |
|----------------|-------------------------|
| Assigned       | reason:assign           |
| Participating  | reason:participating    |
| Mentioned      | reason:mention          |
| Team mention   | reason:team-mention     |
| Review request | reason:review-requested |



Group by: Date ▾

- Repository
- Date
- subscribed

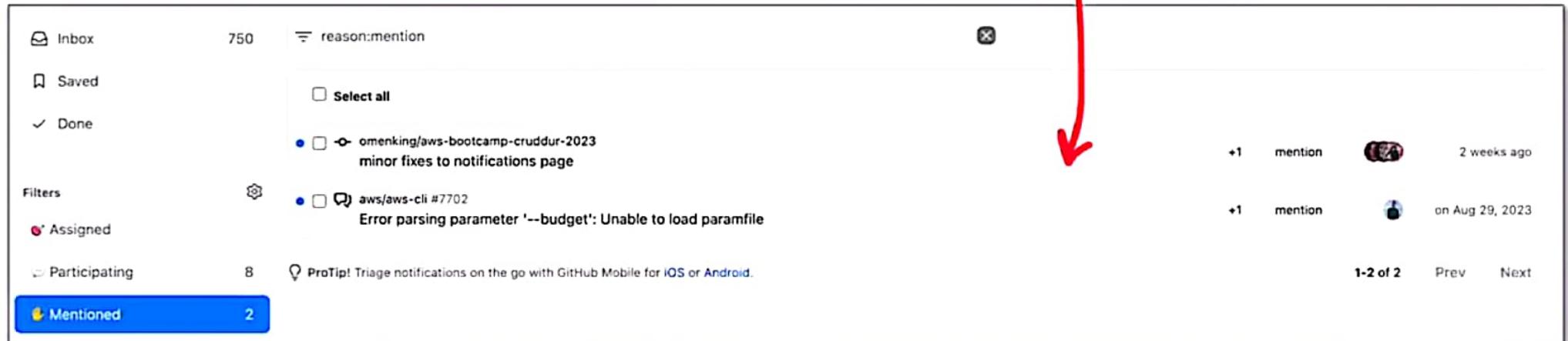
You can **group** Notification Threads by Repo or by Date.

# Mentions in Threads

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

A mention is when someone using  
a @ followed by your username

If someone mentions you in a thread  
than you can find these mention  
under the **Mentioned Filters**.



A screenshot of the GitHub inbox interface, specifically the "Mentioned" filter. The sidebar shows filters like "Inbox" (750 notifications), "Saved", "Done", and "Assigned". The main area lists notifications:

- omenking/aws-bootcamp-crudur-2023 minor fixes to notifications page (mention, 2 weeks ago)
- aws/aws-cli #7702 Error parsing parameter '--budget': Unable to load paramfile (mention, on Aug 29, 2023)

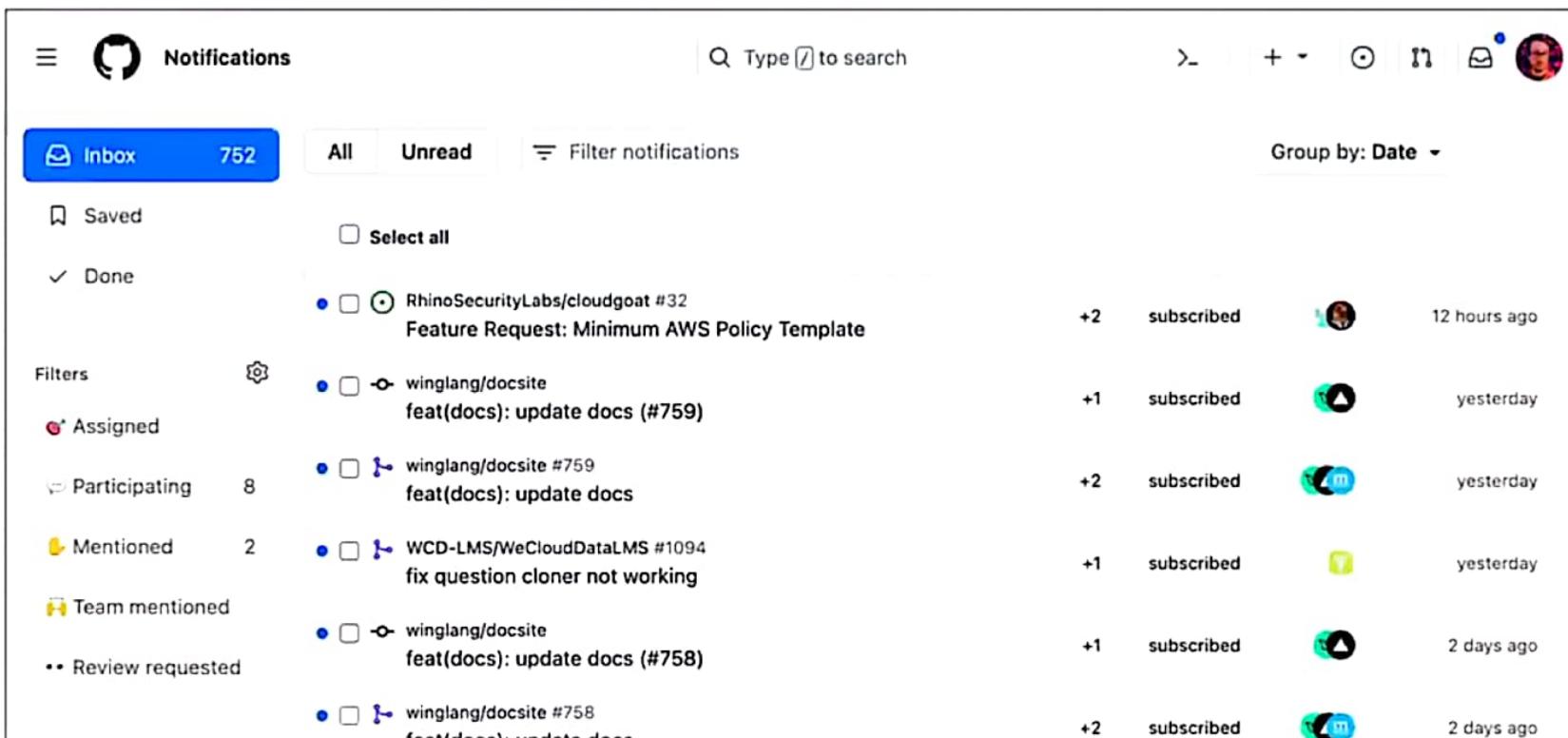
At the bottom, there's a "ProTip! Triage notifications on the go with GitHub Mobile for iOS or Android." message and navigation links "1-2 of 2", "Prev", and "Next". A red arrow points from the "Assigned" filter in the sidebar to the "Assigned" filter in the list of notifications.

# Notifications Inbox

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

The Notifications Inbox provides a chronological list of notifications threads. This allows you to keep on top of all subscribed activities with associated GitHub repos.

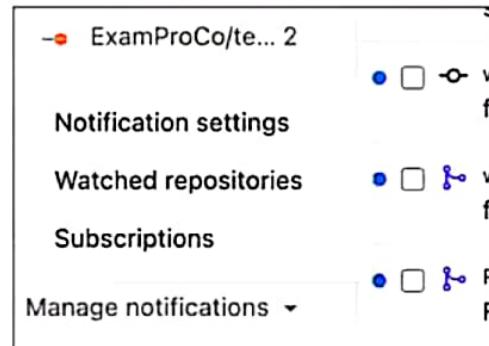
<https://github.com/notifications>



The screenshot shows the GitHub Notifications inbox with 752 unread notifications. The notifications are listed chronologically, grouped by date. Each notification includes the repository name, issue number, title, the user who posted it, the notification type (e.g., subscribed), and the time ago. On the left, there's a sidebar with filters for Saved, Done, Assigned, Participating, Mentioned, Team mentioned, and Review requested. On the right, there's a sidebar with links for ExamProCo/te..., Notification settings, Watched repositories, Subscriptions, and Manage notifications.

| Notification Type | Repository                  | Issue # | Title                                        | Author | Type       | Time Ago     |
|-------------------|-----------------------------|---------|----------------------------------------------|--------|------------|--------------|
| Saved             |                             |         |                                              |        | Subscribed | 12 hours ago |
|                   |                             |         |                                              |        |            |              |
| Done              | RhinoSecurityLabs/cloudgoat | #32     | Feature Request: Minimum AWS Policy Template |        | Subscribed | 12 hours ago |
|                   |                             |         |                                              |        |            |              |
| Filters           | winglang/docsite            | #759    | feat(docs): update docs                      |        | Subscribed | yesterday    |
|                   |                             |         |                                              |        |            |              |
| Assigned          | winglang/docsite            | #759    | feat(docs): update docs                      |        | Subscribed | yesterday    |
|                   |                             |         |                                              |        |            |              |
| Participating     | winglang/docsite            | #759    | feat(docs): update docs                      |        | Subscribed | yesterday    |
|                   |                             |         |                                              |        |            |              |
| Mentioned         | WCD-LMS/WeCloudDataLMS      | #1094   | fix question cloner not working              |        | Subscribed | yesterday    |
|                   |                             |         |                                              |        |            |              |
| Team mentioned    | winglang/docsite            | #758    | feat(docs): update docs                      |        | Subscribed | 2 days ago   |
|                   |                             |         |                                              |        |            |              |
| Review requested  | winglang/docsite            | #758    | feat(docs): update docs                      |        | Subscribed | 2 days ago   |
|                   |                             |         |                                              |        |            |              |

In the far bottom right we can access other **Notifications views:**



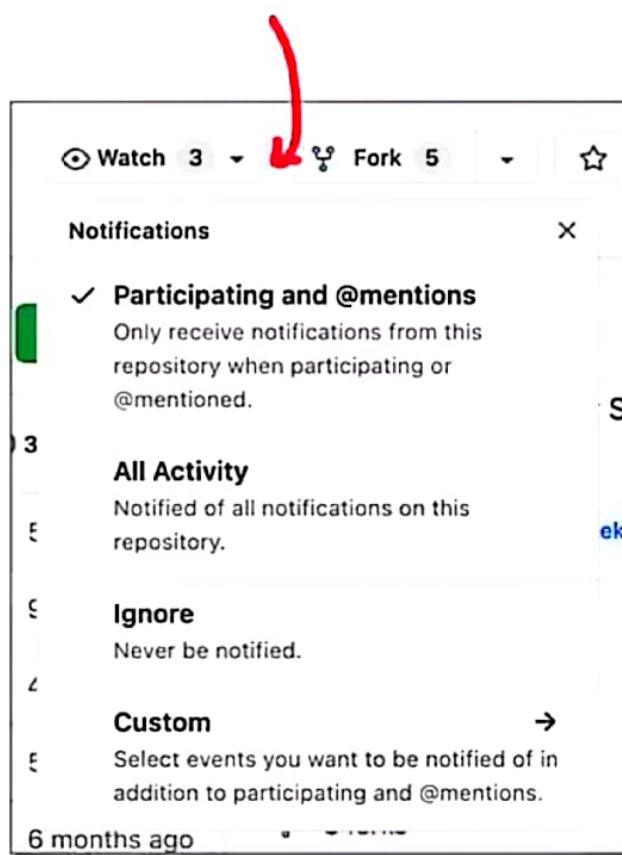
The sidebar on the right contains links to other GitHub notification views:

- ExamProCo/te... 2
- Notification settings
- Watched repositories
- Subscriptions
- Manage notifications ▾

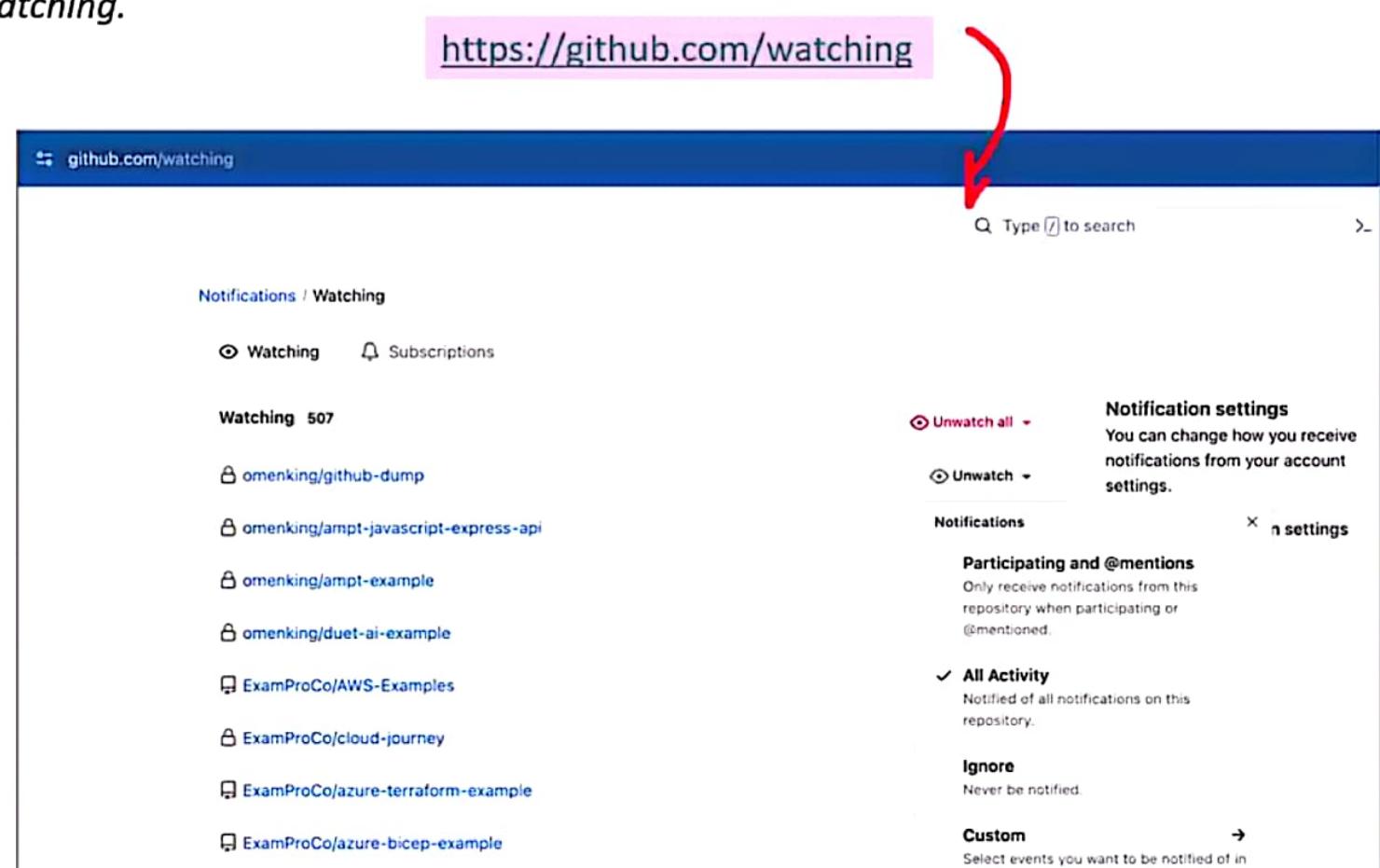
# Notification Watching

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

You can watch repo's in order to get notified of activities.  
*Your own repos you already set as watching.*



You can see a list of watching under notifications



A screenshot of the GitHub 'Watching' list page, accessible via the URL <https://github.com/watching>. The page title is 'Notifications / Watching'. It shows a list of 507 watched repositories, each with a small icon, the repository name, and a 'Unwatch' button. To the right of the list are sections for 'Notification settings' (with 'Participating and @mentions' selected) and 'Notifications' (with 'All Activity' selected). A red arrow points from the top of the slide to the URL in the browser bar.

# Notification Subscriptions

Cheat sheets, Practice Exams and Flash cards ➡ [www.exampro.co/ghf](http://www.exampro.co/ghf)

Subscriptions allow you to follow activities for a **specific Pull Request or Issue.**



You can see all your subscriptions under Notifications.

A screenshot of the GitHub Notifications / Subscriptions page. The URL <https://github.com/notifications/subscriptions> is highlighted with a pink box and a red arrow. The page shows a list of subscriptions with the following details:

| Subscription                                                            | Reason                                        | Repository                                | Sort                     |
|-------------------------------------------------------------------------|-----------------------------------------------|-------------------------------------------|--------------------------|
| omenking/github-dump Finish the Github Foundations Certification course | #1 opened 2 hours ago by omenking             | omenking/github-dump                      | Most recently subscribed |
| omenking/terraform-beginner-bootcamp-2023 My new ticket                 | #68 opened yesterday by omenking              | omenking/terraform-beginner-bootcamp-2023 |                          |
| grpc/grpc Request: Prebuilt musllinux arm artifacts                     | #34998 opened on Nov 16, 2023 by MattDynamite | grpc/grpc                                 |                          |
| omenking/terraform-beginner-bootcamp-2023 Final Category                | #65 opened on Oct 11, 2023 by omenking        | omenking/terraform-beginner-bootcamp-2023 |                          |

# GitHub Gists

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

Gists provide a **simple way to share code snippets with others**  
**Every gist is a Git repository**, which means that it can be **forked** and **cloned**.

<https://gist.github.com>

sebastianlippner / [rspamd.lua](#)    1 file    0 forks    0 comments    2 stars

Last active 9 minutes ago  
rspamd fetchmail plugin

```
1 local rspamd_ip = require "rspamd_ip"
2 local rspamd_logger = require "rspamd_logger"
3 local fun = require 'fun'
4
5 rspamd_config@register_symbol{
6 name = 'FWD_FETCHMAIL',
7 type = 'prefilter',
8 weight = 0.0,
9 callback = function(task)
10 -- Fetchmails runs on localhost
11 end
}
```

A **public gist** on [gist.github.com](https://gist.github.com)

Gists can be **public** or **secret**.

- Public Gists are searchable
- Public Gists show up in Discover
  - <https://gist.github.com/discover>
- Secret gists are not searchable and do not show up in discovery
- Secrets gists are not private
  - You can share the URL with friends
- Secret Gists can be changed to be public
- Public Gists cannot be changed to be private
- You can pin Gists to your profile for other to easily find
- Gists allow other users to comment
- Gists allow you to easily navigation revisions

# Creating GitHub Gists

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

If a Gist is a **markdown file** It will render the HTML.  
This is a great way to share one-off documentation



parmentf / [GitCommitEmoji.md](#)

Last active 14 minutes ago

Git Commit message Emoji

1 file 710 forks 339 comments 3484 stars

Inspired by [dannyfritz/commit-message-emoji](#)

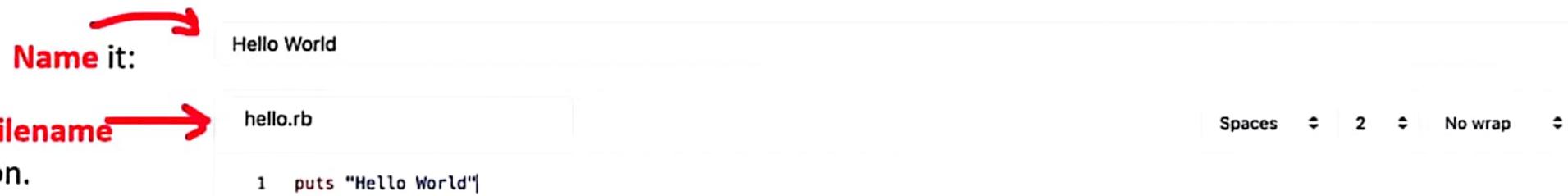
See also [gitmoji](#).

| Commit type    | Emoji        |
|----------------|--------------|
| Initial commit | 🎉 :tada:     |
| Version tag    | 🔖 :bookmark: |
| New feature    | ✨ :sparkles: |
| Bugfix         | 🐛 :bug:      |

# GitHub Gists

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

Create a Gist is a straightforward.



The screenshot shows the GitHub Gist creation interface. A red arrow points from the text "Name it:" to the input field containing "Hello World". Another red arrow points from the text "Provide the filename" to the input field containing "hello.rb". To the right of the code editor, there are settings for "Spaces": a dropdown menu set to "2" and a "No wrap" checkbox.

Provide the **filename** with extension.

- The file extension will allow the file to apply syntax highlighting

Provide your **code** →

Use `Control+Shift+m` to toggle the `tab` key moving focus.



Add file

Attach **associate files** to the gist

Create secret gist ▾

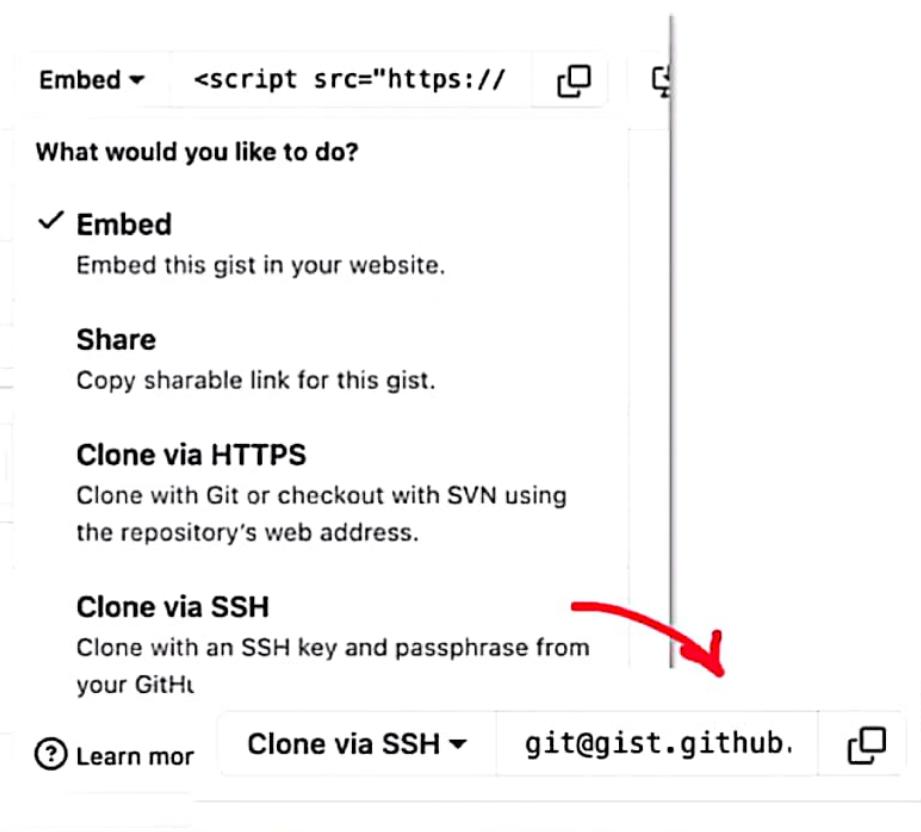


Set it as either **secret** or **private**

# Fork and Clone Gists

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

You can **clone URL** by dropping down embed



Embed ▾ <script src="https://</script>

What would you like to do?

✓ **Embed**  
Embed this gist in your website.

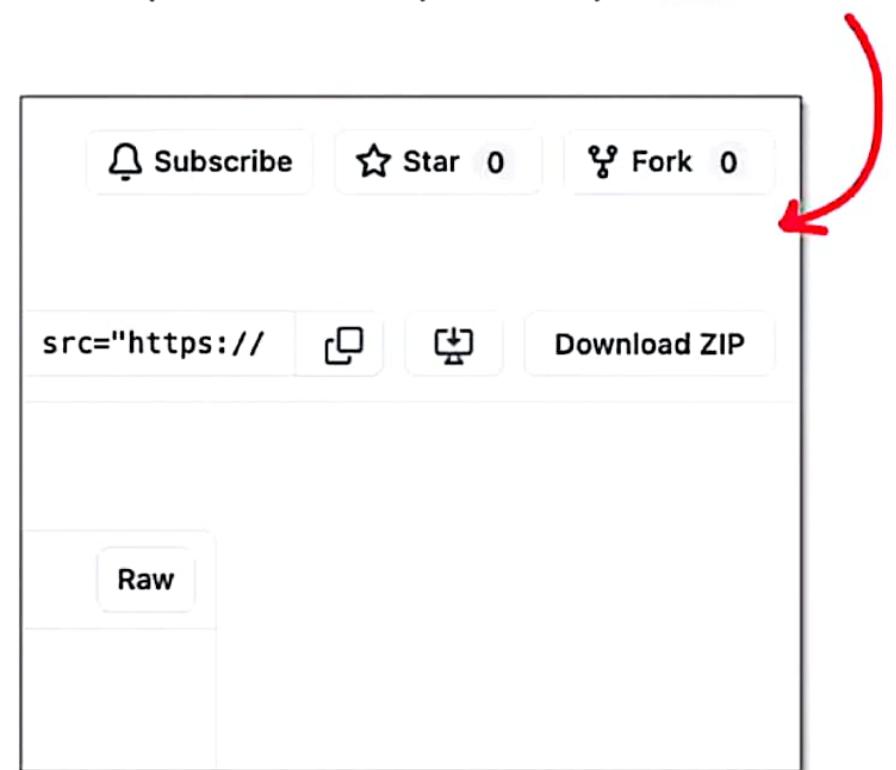
**Share**  
Copy sharable link for this gist.

**Clone via HTTPS**  
Clone with Git or checkout with SVN using  
the repository's web address.

**Clone via SSH**  
Clone with an SSH key and passphrase from  
your GitHub

Clone via SSH ▾ git@gist.github.com

For Gist repos other than your own you **fork** them.



Subscribe Star 0 Fork 0

src="https://</script>

Raw

# GitHub Wiki Pages

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

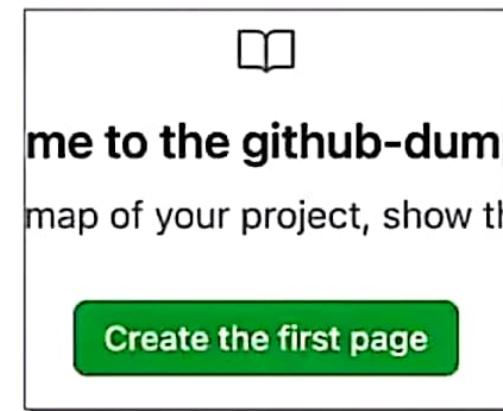
Turn **on the Wiki feature** for your GitHub Repo.



Go to your **Wiki page** of your Repo



Create your **first page!**



GitHub Wiki's can be used for free on Public repos.

# GitHub Wiki Pages

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

A Wiki is collaborative way to quickly **create documentation with multiple nested pages**

A wiki would serve as a **knowledge base** for your repo or organization.

## Configuration files

Justin MacCarthy edited this page on Aug 23, 2021 · 2 revisions

### Files used by Guard

#### Project-specific Guard DSL files:

`Guardfile` (in current directory) - can contain DSL statements, evaluated by Guard during startup

`.Guardfile` (in current directory) - if it exists, it is used instead of the `Guardfile`

#### Global Guard DSL files:

`~/.guard.rb` (in home directory) - can contain DSL statements. If it exists, it is always evaluated by Guard (before evaluating the project-specific DSL).

### Pry-only config

Guard supports the `XDG_CONFIG_HOME` environment value if it is set .

`~/.guardrc` (in home directory) - if it exists, it is evaluated by the Pry interator (and should not contain DSL statements)

This wiki and the Guard [README](#) document contains a lot of information, please take your time and read these instructions carefully.

If you run into any trouble, you may start by [understanding how Guard works](#).

Edit [New page](#)

### Pages

Find a page...

- ▶ Home
- ▶ Add Readline support to Ruby on M...
- ▶ Analysis of inotify events for differ...
- ▶ Command line options for Guard
- ▶ Configuration files

#### Files used by Guard

##### Project-specific Guard DSL files:

##### Global Guard DSL files:

##### Pry-only config

- ▶ Configure keyboard shortcuts
- ▶ Correctly using the watchdir option

**Search pages**

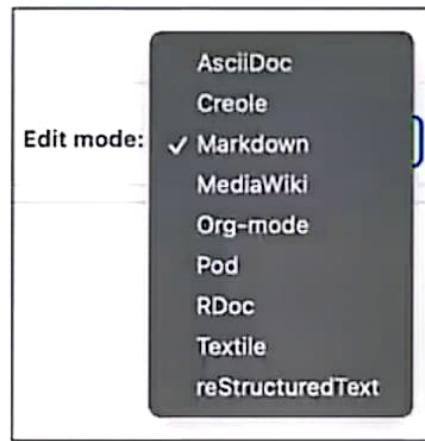


**Nested pages**



# Managing GitHub Wiki Pages

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)



When you create a page you can use  
**various markup languages.**

To **link to other pages** you provide the Github Wiki pages title as the link



A screenshot of the GitHub Wiki 'Write' interface. It shows a toolbar with 'Write' and 'Preview' tabs, and buttons for h1, h2, h3, a link icon, and an image icon. Below the toolbar, there is a text area containing the text '[My Home Page] (Home)'. A red arrow points from the text in the Write area to the 'My Home Page' link in the adjacent 'About' box.



A screenshot of the GitHub Wiki 'About' interface. It displays the text 'About' and 'Andrew Brown edited this page'. Below that, it shows a blue link labeled 'My Home Page'. At the bottom of the box, there is a URL: <https://github.com/omenking/github-dump/wiki/Home>. A red arrow points from the 'My Home Page' link in the 'Write' box to this URL.

If you create a wiki in a **public repository**,  
the wiki is available to the public.

If you create a wiki in a **private repository**,  
only people with access to the repository  
can access the wiki.

# GitHub Pages

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

GitHub Pages allows you to directly host a static website via a GitHub repository.

You need to create a **public repo** with the repo name being `<username>.github.io`

Owner \*      Repository name \*

 omenking    /    omenking.github.io

 omenking.github.io is available.

Great repository names are short and memorable. Need ins...

Description (optional)

Public

Anyone on the internet can see this repository. You cho...

You simply **add HTML files and assets** to your repo.



**Jekyll** is static site generator which allows you to build complex static sites or blogs with ease.

- Jekyll uses yaml files with frontmatter format.
- Jekyll utilizes markdown files
- Jekyll directly integrates with GitHub Pages and is a GitHub project

Your website will be available the domain name: `<username.github.com>`

# Introduction to GitHub Actions

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

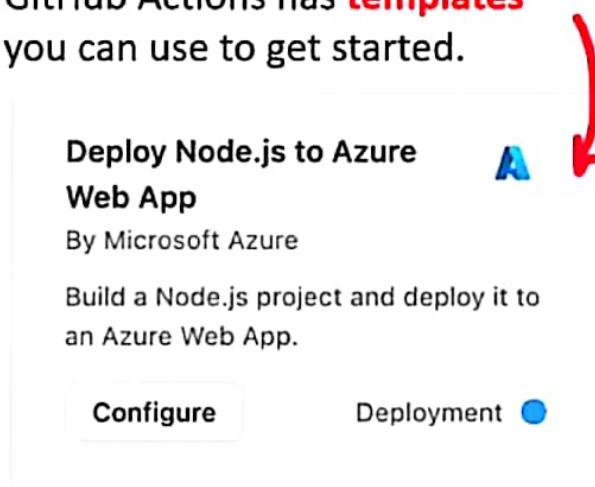


**GitHub Actions is a CI/CD pipeline directly integrated with your GitHub repository.**

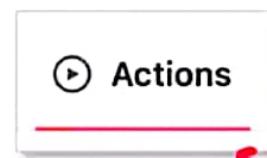
Github Actions allows you to **automate**:

- Running test suites
- Building images
- Compiling static sites
- Deploying code to servers
- and more...

GitHub Actions has **templates** you can use to get started.



A screenshot of a GitHub Actions template card. It shows the title "Deploy Node.js to Azure Web App" by Microsoft Azure. Below the title, it says "Build a Node.js project and deploy it to an Azure Web App." At the bottom, there are two buttons: "Configure" and "Deployment". A red arrow points from the "Deployment" button towards the ".github/workflow" section below.



Within a Github repo you'll have a **tab for Actions**.

```
on:
 push:
 branches: [$default-branch]
 workflow_dispatch:
env:
 AZURE_WEBAPP_NAME: your-app-name
 AZURE_WEBAPP_PACKAGE_PATH: '.'
 NODE_VERSION: '14.x'
permissions:
 contents: read
jobs:
 build:
 runs-on: ubuntu-latest
 steps:
 - uses: actions/checkout@v3
 ...
```

# Introduction to GitHub Actions

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

When you run GitHub Actions you'll get **a history of workflow runs** where it will indicate if it was success a failure, and how long it took to run.



| All workflows                                                                                                                                        | Showing runs from all workflows | Q Filter workflow runs                                                                                                                                                                               |
|------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 113 workflow runs                                                                                                                                    |                                 | Event ▾ Status ▾ Branch ▾ Actor ▾                                                                                                                                                                    |
|  Merge pull request #116 from ExamProCo/main                        | production                      |  6 months ago<br> 10m 10s      |
| Deploy production #113: Commit <a href="#">8cac2e9</a> pushed by omenking                                                                            |                                 |                                                                                                                                                                                                      |
|  Merge pull request #114 from ExamProCo/main                        | production                      |  6 months ago<br> 10m 55s      |
| Deploy production #112: Commit <a href="#">6d2e4e8</a> pushed by omenking                                                                            |                                 |                                                                                                                                                                                                      |
|  Merge pull request #113 from ExamProCo/main                       | production                      |  9 months ago<br> 11m 1s   |
| Deploy production #111: Commit <a href="#">af69015</a> pushed by bayko                                                                               |                                 |                                                                                                                                                                                                      |
|  Merge pull request #112 from ExamProCo/vimeo-submission-feedb... | production                      |  10 months ago<br> 11m 24s |
| Deploy production #110: Commit <a href="#">7c4cd93</a> pushed by bayko                                                                               |                                 |                                                                                                                                                                                                      |
|  Merge pull request #111 from ExamProCo/main                      | production                      |  10 months ago<br> 12m 34s |
| Deploy production #109: Commit <a href="#">f0152e7</a> pushed by bayko                                                                               |                                 |                                                                                                                                                                                                      |
|  Merge pull request #110 from ExamProCo/main                      | production                      |  10 months ago<br> 18m 1s  |
| Deploy production #108: Commit <a href="#">6ed15e7</a> pushed by bayko                                                                               |                                 |                                                                                                                                                                                                      |

# Finding Github Actions

Cheat sheets, Practice Exams and Flash cards [👉 www.exampro.co/ghf](http://www.exampro.co/ghf)

Github has a repo of example workflows you can use to get you started

<https://github.com/actions/starter-workflows>

The screenshot shows two parts of the GitHub interface. On the left, the repository 'starter-workflows' is displayed, which is public and contains 23 branches and 0 tags. A red arrow points from the URL bar above to the 'Public' label on the repository page. On the right, the specific file 'ci/go.yml' is shown in a detailed view. Another red arrow points from the left repository page to the file's code editor area. The file content is a GitHub Actions workflow:

```
1 # This workflow will build a golang project
2 # For more information see: https://docs.github.com/en/act
3
4 name: Go
5
6 on:
7 push:
8 branches: [$default-branch]
9 pull_request:
10 branches: [$default-branch]
11
12 jobs:
13
14 build:
15 runs-on: ubuntu-latest
16 steps:
```

# Different Types of GitHub Actions

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

Event Triggers causes a GitHub Action to run.

The **on** attribute specifies  
the **event trigger** to be used:

Github Actions has 35+ event triggers

```
name: Example Workflow
on: [push]
jobs:
 build:
 runs-on: ubuntu-latest
 steps:
 - uses: actions/checkout@v2
 - name: Run a script
 run: echo "This script runs on every push!"
```

Examples of common GitHub Actions that could be triggered:

- **Pushes:** Trigger an action on any push to the repository.
- **Pull Requests:** Run actions when pull requests are opened, updated, or merged.
- **Issues:** Execute actions based on issue activities, like creation or labeling.
- **Releases:** Automate workflows when a new release is published.
- **Scheduled Events:** Schedule actions to run at specific times.
- **Manual Triggers:** Allow manual triggering of actions through the GitHub UI.

# Introduction to GitHub Copilot

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

GitHub Copilot is **AI developer tool** that can be used with multiple IDEs via an extension

- **Code Completion** 
- Chat in IDE and Mobile
- CLI assistance
- Code referencing
- Public code filter
- IP indemnity

```
game.rb
A terminal-based black jack game with weird scifi theme and new rules.

def game
 puts "Welcome to the game of Black Jack!"
 puts "What is your name?"
 name = gets.chomp
 player = Player.new(name)
 dealer = Dealer.new
 deck = Deck.new
 deck.shuffle
 player.hand = Hand.new
 dealer.hand = Hand.new
 player.hand.add_card(deck.deal)
 dealer.hand.add_card(deck.deal)
 player.hand.add_card(deck.deal)
 dealer.hand.add_card(deck.deal)
 puts "The dealer is showing #{dealer.hand.cards[0].rank} of #{dealer.hand.cards[0].suit}"
 puts "You have a #{player.hand.cards[0].rank} of #{player.hand.cards[0].suit}"
 puts "Your total is #{player.hand.total}."
 puts "Would you like to hit or stay?"
 hit_or_stay = gets.chomp.downcase
 while hit_or_stay == "hit"
 player.hand.add_card(deck.deal)
 puts "You have a #{player.hand.cards[0].rank} of #{player.hand.cards[0].suit}"
 end
end
```

FREE  
TRIAL

You can try GitHub Copilot for free with a one-time 30-day trial.

Github Copilot is free to use for verified students, teachers, and maintainers of popular open source projects.

# GitHub Copilot Individuals vs Business

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

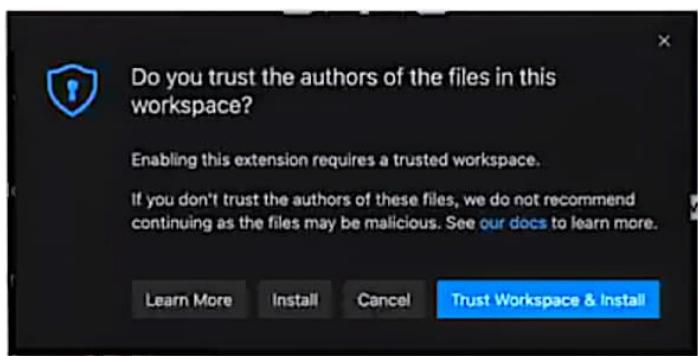
|                                            | Copilot Individuals                      | Copilot Business                    |
|--------------------------------------------|------------------------------------------|-------------------------------------|
| Pricing                                    | \$10 USD per month<br>\$100 USD per year | \$19 USD per user per month         |
| Types of GitHub accounts                   | Personal accounts                        | Organization or enterprise accounts |
| Telemetry                                  | ✓                                        | ✗                                   |
| Blocks suggestions matching public code    | ✓                                        | ✓                                   |
| Plugs right into your editor               | ✓                                        | ✓                                   |
| Offers multi-line function suggestions     | ✓                                        | ✓                                   |
| Organization-wide policy management        | ✗                                        | ✓                                   |
| Exclude specified files                    | ✗                                        | ✓                                   |
| Audit logs                                 | ✗                                        | ✓                                   |
| HTTP proxy support via custom certificates | ✗                                        | ✓                                   |

# Setting up Copilot

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)



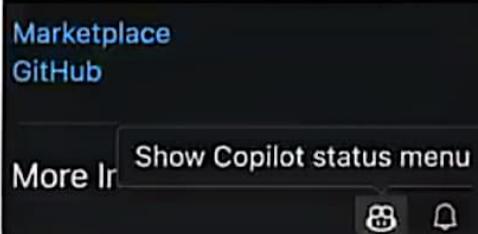
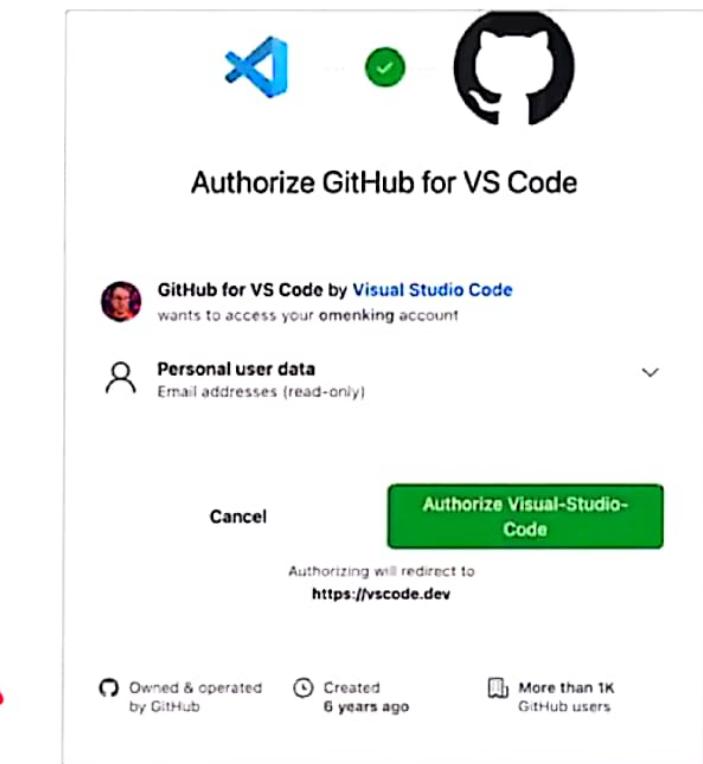
Install the **GitHub Copilot extension** in your VS Code



Choose **Trust Workspace & Install**



**Sign in to GitHub**



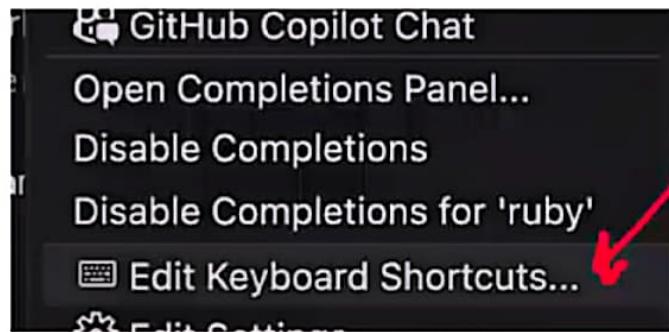
Copilot will show up  
in **bottom right corner**



# Using Copilot

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

Copilot is best used when you  
**know or set hotkeys** to  
perform suggestions



| Command                                        | Keybinding |
|------------------------------------------------|------------|
| GitHub <b>Copilot</b> : Accept panel sugg...   | ^ /        |
| github.copilot.acceptCursorPane...             |            |
| GitHub <b>Copilot</b> : Navigate to the nex... | ^ ]        |
| github.copilot.nextPanelSolution               |            |
| GitHub <b>Copilot</b> : Navigate to the pre... | ^ [        |
| github.copilot.previousPanelSol...             |            |
| GitHub <b>Copilot</b> : Open Completions ...   | ^ Enter    |
| github.copilot.generate                        |            |
| GitHub <b>Copilot</b> : Suggest Terminal C...  | ⌘ /        |
| github.copilot.terminal.suggest...             |            |
| Trigger Inline Suggestion                      | ^ \        |

Its best to check what hotkeys you have since they may vary based on the documentation stated hotkeys

CHAT: GITHUB COPILOT

game.rb U X

lib > game.rb

```
1 # A terminal-based
2
3 def game
4
5 end
```

Here's a basic implementation of this logic in Ruby:

```
class Card
 attr_reader :value

 def initialize(value)
 @value = value
 end
end

class Deck
 def initialize
 @cards = (1..10).to_a * 4
 @cards.shuffle!
 end

 def deal
 Card.new(@cards.pop)
 end
end

def game
 deck = Deck.new
 player_cards = [deck.deal, dec
 ◊ How can I handle user input for the player's actions in the game?
```

Ask Copilot or type '/' for command ➤

You need to have **Github Copilot Chat feature** enabled in your personal account or for your organization.

#### Features

##### Copilot Chat in the IDE

If enabled, members of this organization will have access to GitHub Copilot Chat in IDE.

Enabled ➔

##### Copilot in the CLI Beta

If enabled, members of this organization will get GitHub Copilot assistance in terminal.

Disabled ➔

Once enabled you can  
**chat with Google Copilot**

# Introduction to GitHub Codespaces

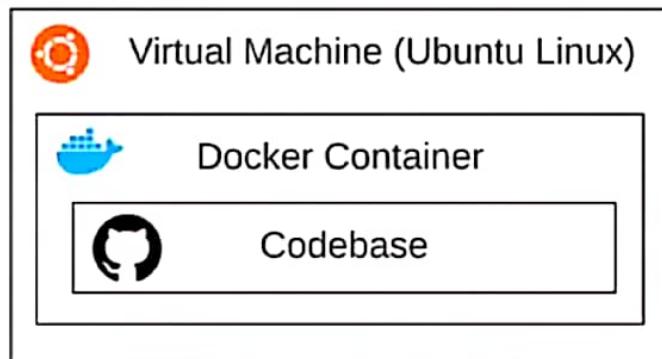
Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

**GitHub Codespaces is Cloud Developer Environment (CDE) integrated with your GitHub repo.**

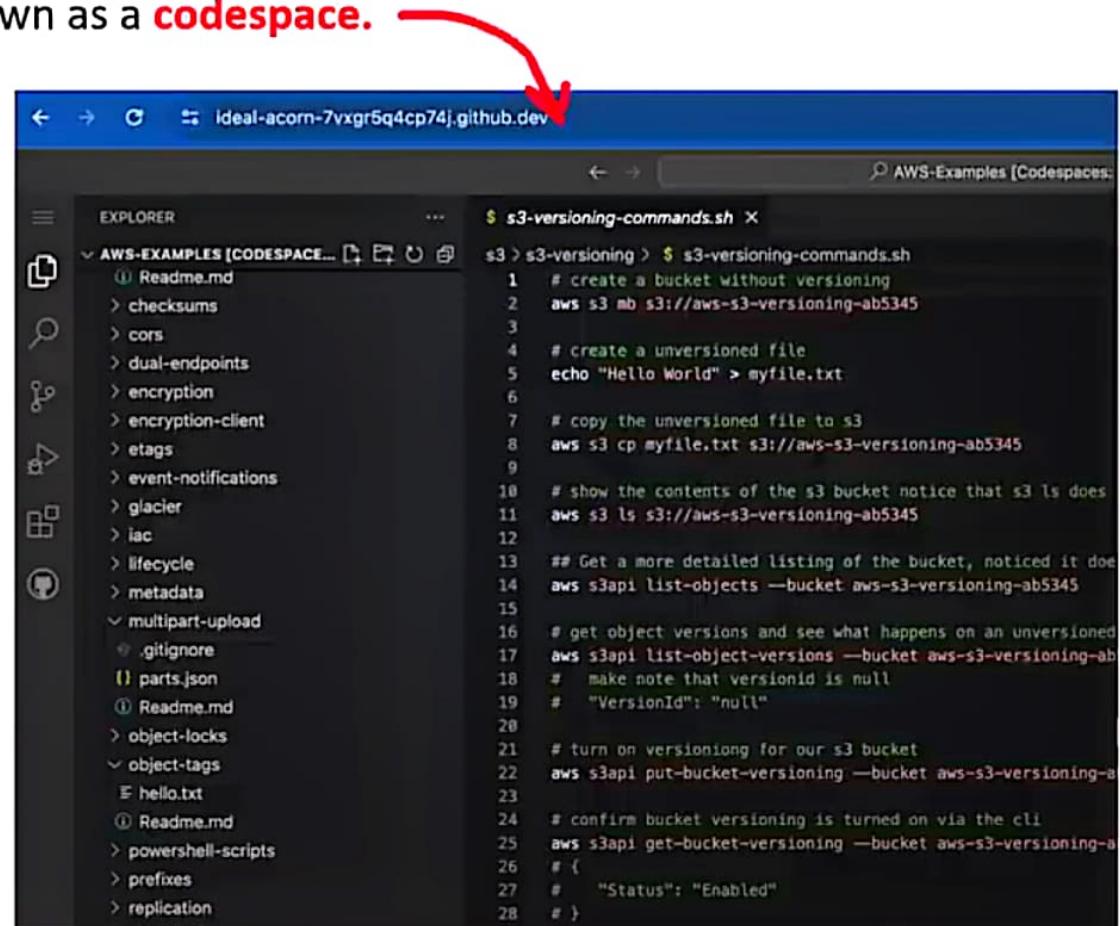
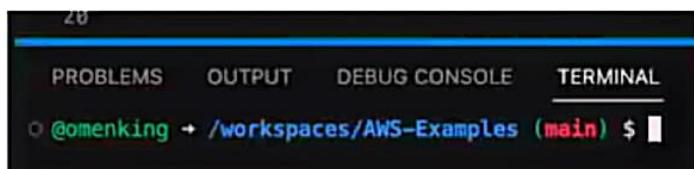
An environment is known as a **codespace**.

A codespace runs:

- in Ubuntu Linux docker container
  - on a virtual machine
  - hosted and managed by GitHub



Your codebase will be cloned into **/workspaces**

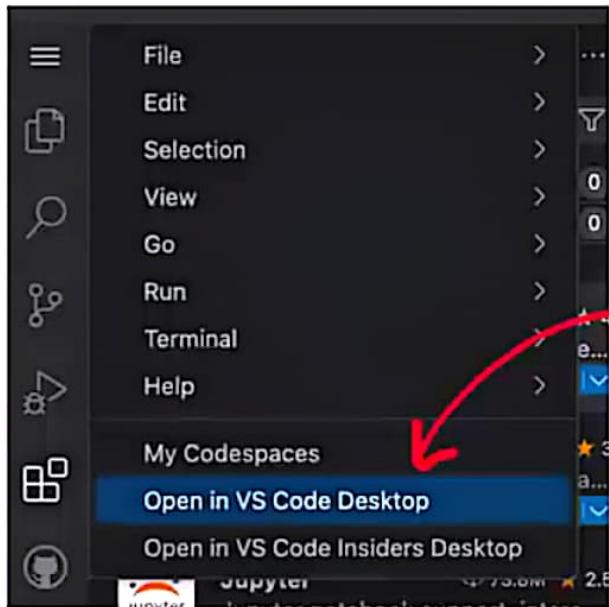


The screenshot shows the GitHub Codespace interface within a web browser. At the top, the URL is 'ideal-acorn-7vxgr5q4cp74j.github.dev'. The interface includes an EXPLORER sidebar with a tree view of files and folders, a TERMINAL pane showing a shell session with AWS commands, and an EDITOR pane where code is being written. A red arrow points from the word 'codespace.' in the text above to the browser tab. Another red arrow points from the '/workspaces' text below to the terminal output.

```
$ s3-versioning-commands.sh
1 # create a bucket without versioning
2 aws s3 mb s3://aws-s3-versioning-ab5345
3
4 # create an unversioned file
5 echo "Hello World" > myfile.txt
6
7 # copy the unversioned file to s3
8 aws s3 cp myfile.txt s3://aws-s3-versioning-ab5345
9
10 # show the contents of the s3 bucket notice that s3 ls does
11 aws s3 ls s3://aws-s3-versioning-ab5345
12
13 ## Get a more detailed listing of the bucket, noticed it doe
14 aws s3api list-objects --bucket aws-s3-versioning-ab5345
15
16 # get object versions and see what happens on an unversioned
17 aws s3api list-object-versions --bucket aws-s3-versioning-ab
18 # make note that versionid is null
19 # "VersionId": "null"
20
21 # turn on versioning for our s3 bucket
22 aws s3api put-bucket-versioning --bucket aws-s3-versioning-a
23
24 # confirm bucket versioning is turned on via the cli
25 aws s3api get-bucket-versioning --bucket aws-s3-versioning-a
26 #
27 # "Status": "Enabled"
28 #
```

# Introduction to GitHub Codespaces

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)



In the VS Code Browser you can tell it to connect this codespace To your locally installed VS Code

You can interact with your codespace via:

- Command Line
  - **SSH via GitHub CLI**
- Code Editors
  - VS Code Browser
  - **VS Code Desktop**
  - JetBrains IDE

The Github CLI commands available for Codespaces:

- gh codespace code
- gh codespace cp
- gh codespace create
- gh codespace delete
- gh codespace edit
- gh codespace jupyter
- gh codespace list
- gh codespace logs
- gh codespace ports
- gh codespace rebuild
- gh codespace ssh
- gh codespace stop
- gh codespace view

# Introduction to GitHub Codespaces

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

When you create a new codespace you **choose**:

- The repo
- The branch
- The Region
- Machine type



Create a new codespace

**Repository**  
To be cloned into your codespace Select a repository ▾

**Branch**  
This branch will be checked out on creation Default branch ▾

**Region**  
Your codespace will run in the selected region US East ▾

**Machine type**  
Resources for your codespace None ▾

Create codespace

The **regions** you can choose:

- US East
- US West
- Europe West
- Southeast Asia
- Australias

The capacity of your environment is based on the virtual **machine type** you choose:

- 2-Core (8GB RAM, 32 GB)
- 4-Core (16GB RAM, 32 GB)
- 8-Core (32GB RAM, 64 GB)
- 16-Core (32GB RAM, 128 GB)

# Introduction to GitHub Codespaces

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

You can see all of your codespaces across all  
your repos at <https://github.com/codespaces>



The screenshot shows the GitHub Codespaces interface. At the top, there are navigation links for 'All' and 'Templates'. Below that, a section titled 'Your codespaces' shows a single item: 'omenking/terraform-beginner-bootcamp-2023'. Underneath this, there's a heading 'Explore quick start templates' followed by four template cards:

- Blank** By github 
- React** By github 
- Jupyter Notebook** By github 
- .NET** By github 

Each template card has a description, a 'Use this template' button, and a 'See all' link. At the bottom, it says 'Owned by omenking' and shows a repository card for 'omenking/terraform-beginner-bootcamp-2023' with a 'dev-env2' branch, 0 changes, and resource details: 2-core + 8GB RAM + 32GB, 0.92 GB, Active, and three dots.

# Starting GitHub Codespaces

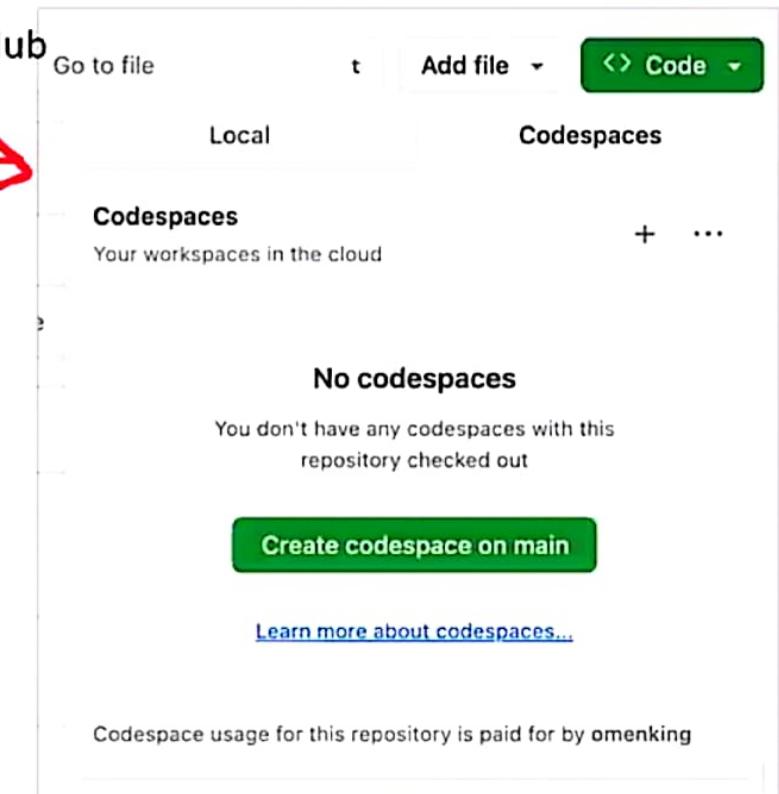
Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

You can start a Codespace via your GitHub repo via the **Code** button



You can also use the **GitHub CLI** to create an open a codespace

```
gh codespace create
gh codespace open
```



Go to file t Add file <> Code Local Codespaces

**Codespaces**  
Your workspaces in the cloud + ...

**No codespaces**  
You don't have any codespaces with this repository checked out

Create codespace on main

[Learn more about codespaces...](#)

Codespace usage for this repository is paid for by omenking

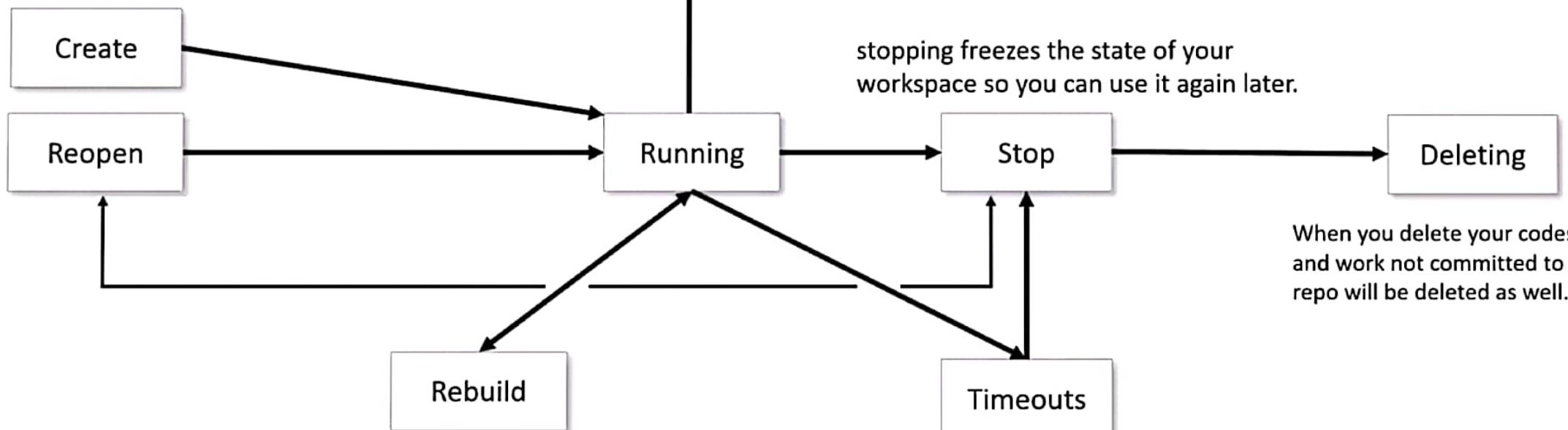


There is a limit of concurrent codespaces allowed to run.  
If you hit the limit you may need to stop other codespaces first.

# GitHub Codespace Lifecycle

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

You choose to **create** a codespace or **reopen** an existing one



If you made changes to your dev container configuration. You need to rebuild your codespace for them to take affect

Codespace by default will timeout after 30 mins of activity. The codespace will **stop** to conserve spend. Your changes will persist on the Virtual Machine

# Renaming Codespaces

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

When you create a codespace it's assigned an **auto-generated display name**.

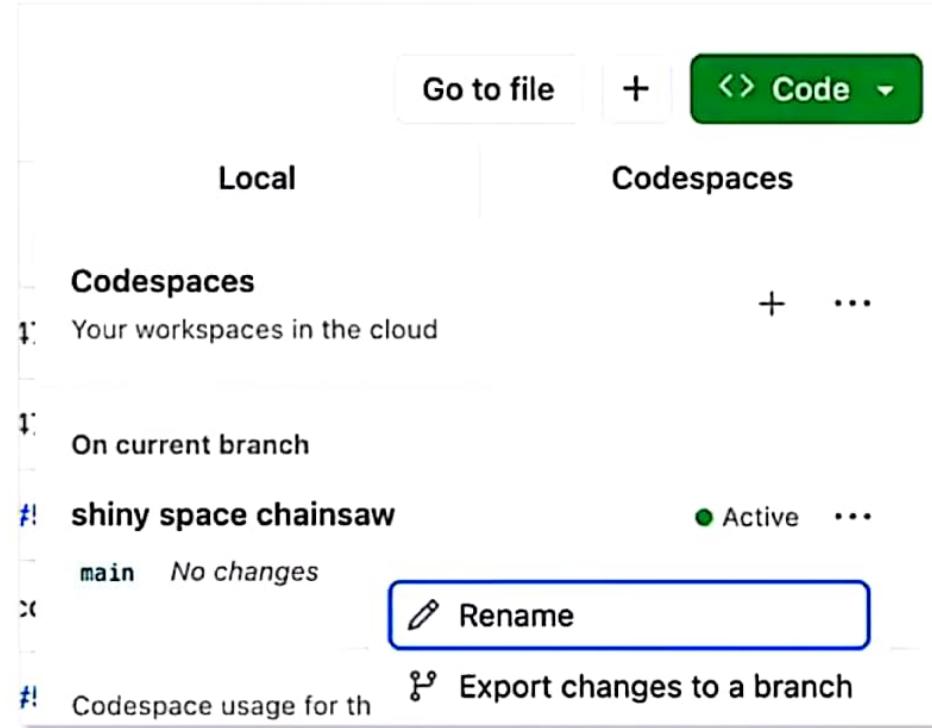
shiny-space-chainsaw-jjgx4w5r3q9xj



You can rename the auto-generated name in the case you have multiple codespaces to easily identify the purpose or state of your codespace.

You can see all of your codespaces at

<https://github.com/codespaces>



The screenshot shows the GitHub Codespaces interface. At the top, there are tabs for "Local" and "Codespaces". Below the tabs, there's a section titled "Codespaces" with two items:

- "Your workspaces in the cloud" (with a plus sign and three dots)
- "On current branch"

Under "On current branch", there is a list of codespaces:

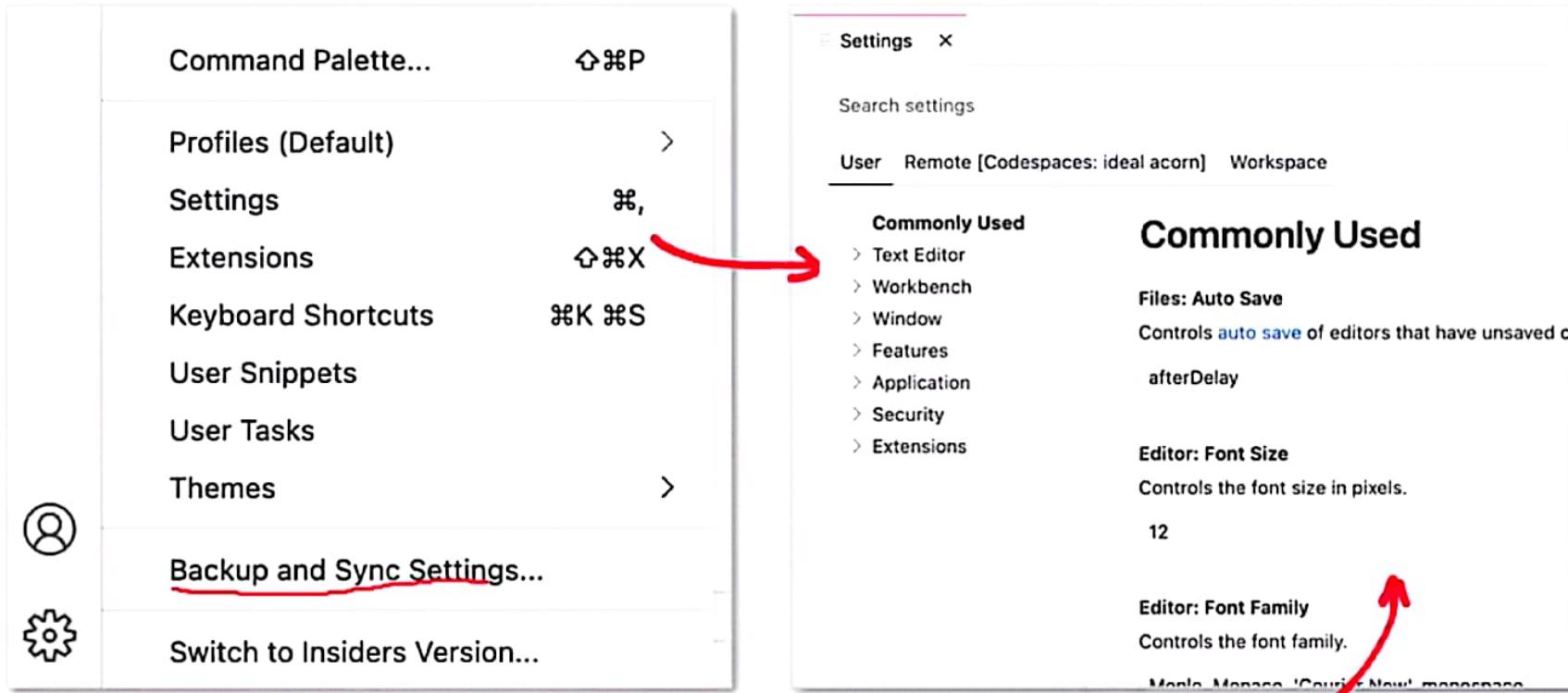
- # shiny space chainsaw (marked as Active) with a "Rename" button highlighted by a blue border and a "..." button.
- main No changes
- # Codespace usage for th

At the bottom right, there is a "Export changes to a branch" button.

# VS Code Settings

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

Github Codespaces **should persist your VS Code settings** that you change in your codespace.



When you edit Settings in the UI you are **actually modifying a settings.json file**.

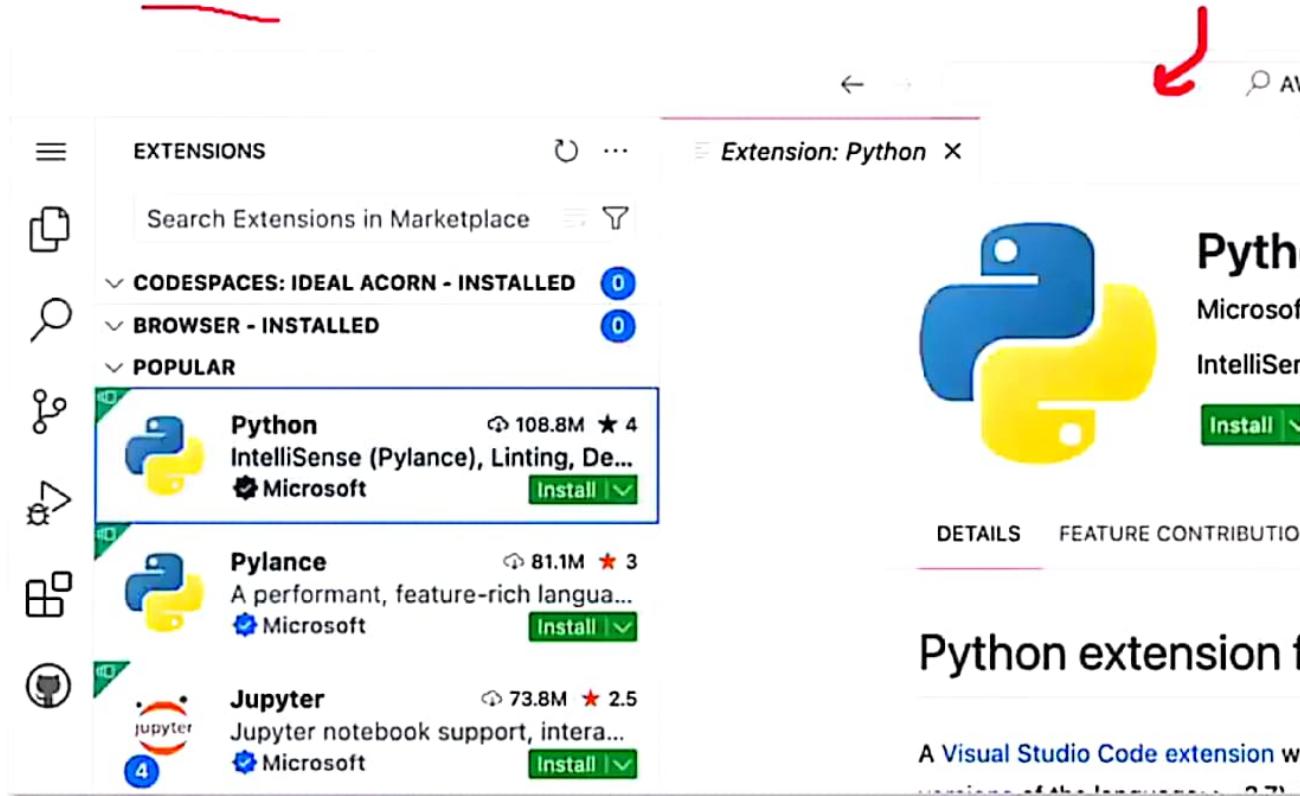
It gets synced and backed up hosted by GitHub

You can override setting.json options in your devcontainer.json file

# VS Code Extensions

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

Github Codespaces **should persist your VS Code extensions** that you change in your codespace.

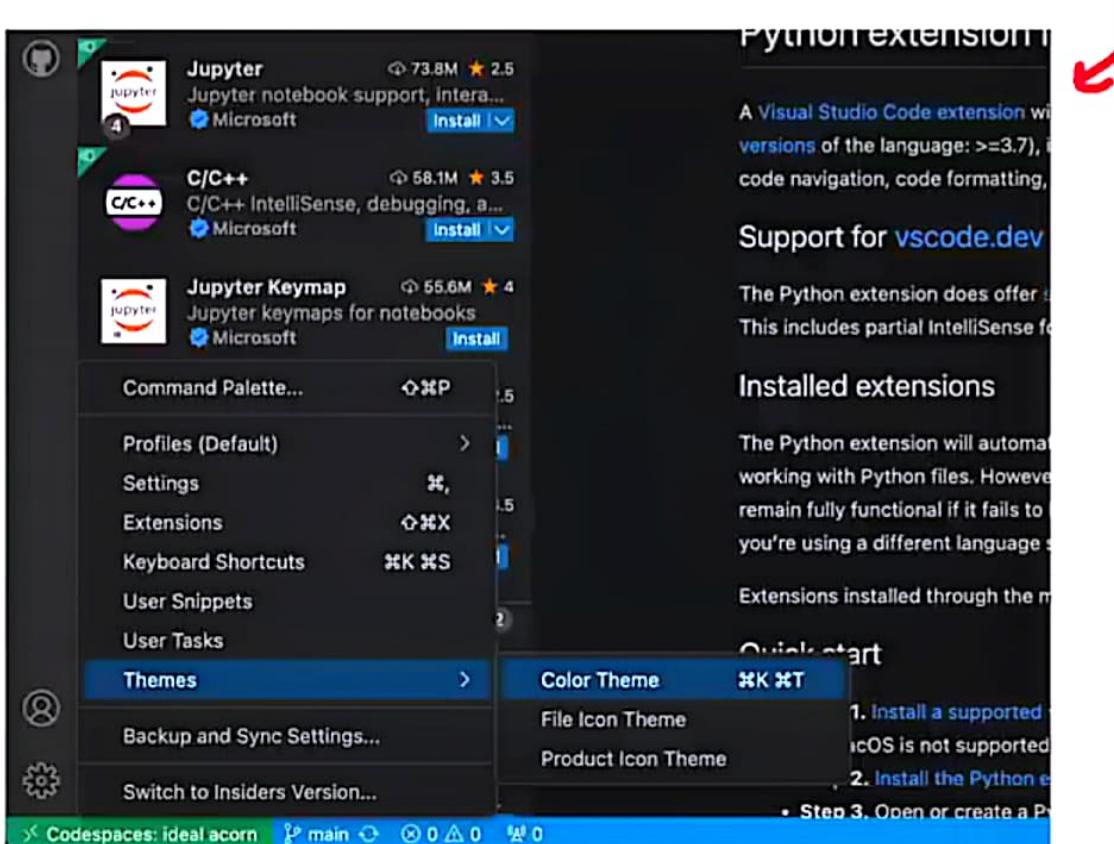


**Most VSCode Marketplace extensions are available** to install in VSCode Browser for GitHub codespaces  
You can ensure your extensions reliably load everytime by setting them in your devcontainer.json

# VS Code Theme

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

Github Codespaces **should persist your VS Code theme** that you change in your codespace.



user interface decorators or themes are personal choices that should not be put in the devcontainer.json file

# Retention Period and Timeout Configuration

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

Under **Github Settings > Codespaces** you can change the **timeout** and **retention period**.

## Default idle timeout

A codespace will suspend after a period of time. You can change this after the default is changed. You will be charged for the codespace after 240 minutes (4 hours).

30 minutes

Save

How long should codespaces wait to stop a codespace when there has been no activity.

- Minimum: 5 minutes
- Maximum 240 minutes (4 hours)
- Default: 30 minutes

## Default retention period

Inactive codespaces are automatically deleted. This applies to all codespaces created going forward.

30 days

Save

How many days should pass until inactive codespaces are automatically deleted?

- Minimum: 0 days
- Maximum 30 days
- Default: 30 days

# Codespaces Sync Settings

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

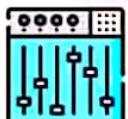
## Settings Sync

By enabling, your codespaces will be able to pull from VS C  
Only enable this for repositories that you trust.

**Enable**

VS Code Settings Sync will be available in Codespaces

You can have codespaces able to pull from  
**VS Code Settings Sync service**



**VS Code Settings Sync** lets you share your Visual Studio Code configurations such as settings, keybindings, and installed extensions across your machines so you are always working with your favorite setup.

# Installing Dotfiles

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

You can configure GitHub Codespaces to **use dotfiles from a specific repository** that will be used for all future codespaces you start.

## Dotfiles

### Automatically install dotfiles

Codespaces can automatically install your dotfiles in

 omenking/my-dot-files ▾



When you create a new codespace with this feature, GitHub does two things:

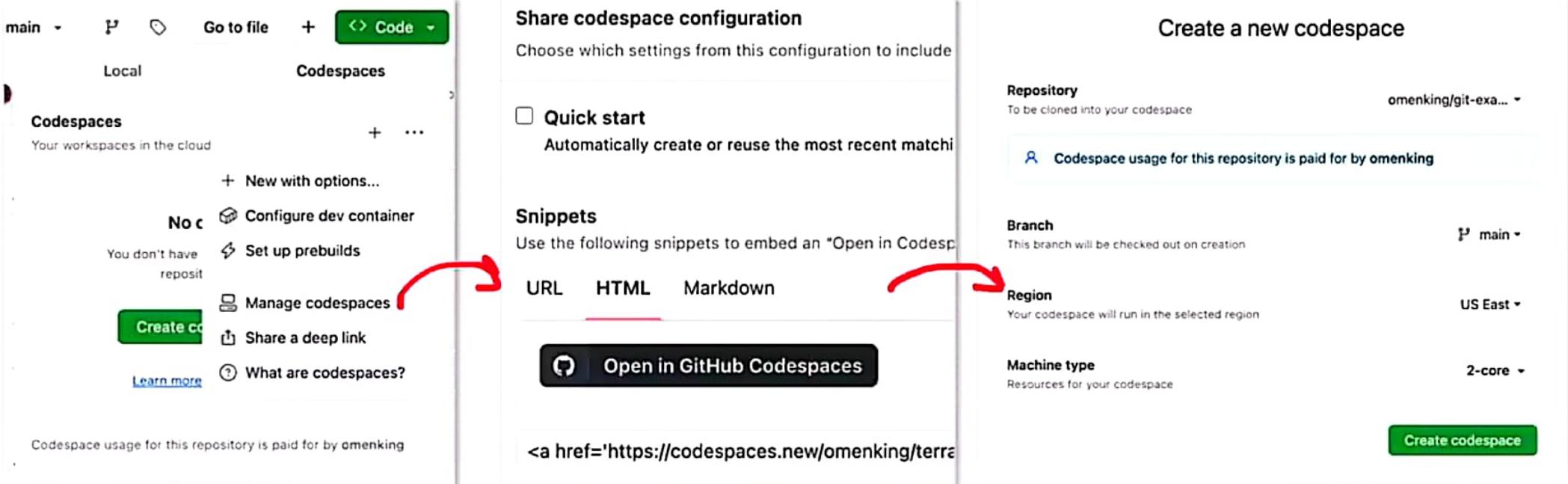
1. GitHub clones your selected dotfiles repository to the codespace environment.
2. Looks for one of the following files to set up the environment.
  - *install.sh*
  - *install*
  - *bootstrap.sh*
  - *bootstrap*
  - *script/bootstrap*
  - *setup.sh*
  - *setup*
  - *script/setup*

# GitHub Codespaces Deep Links

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

Github Codespaces Deep Link is an easy way to generate a shareable link that will launch a Codespace for a specific Github Repo.

<https://codespaces.new/omenking/git-examples>



The screenshot shows the GitHub Codespaces interface. On the left, the sidebar has 'Codespaces' selected. A red arrow points from the 'Share a deep link' button to the central 'Share codespace configuration' dialog. Another red arrow points from the 'HTML' tab in the 'Snippets' section to the generated URL. The central dialog shows settings for sharing a configuration, including 'Quick start' and 'Snippets' sections. The right side shows the 'Create a new codespace' form, which includes fields for 'Repository' (set to 'omenking/git-exa...'), 'Branch' (set to 'main'), 'Region' (set to 'US East'), and 'Machine type' (set to '2-core'). A red arrow points from the 'Open in GitHub Codespaces' button to the generated URL at the bottom of the dialog.

main ▾ P Go to file + <> Code ▾

Local Codespaces

Codespaces Your workspaces in the cloud + ...

+ New with options...

No c Configure dev container Set up prebuilds

You don't have any repositories yet

+ Manage codespaces Share a deep link Create codespace

Learn more What are codespaces?

Codespace usage for this repository is paid for by omenking

Share codespace configuration

Choose which settings from this configuration to include

Quick start

Automatically create or reuse the most recent matching configuration

Snippets

Use the following snippets to embed an "Open in Codespaces" button

URL **HTML** Markdown

<https://codespaces.new/omenking/git-examples>

Open in GitHub Codespaces

<a href='https://codespaces.new/omenking/terra...'

Create a new codespace

Repository To be cloned into your codespace omenking/git-exa...

Branch This branch will be checked out on creation main

Region Your codespace will run in the selected region US East

Machine type Resources for your codespace 2-core

Create codespace

# Codespaces Secrets

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

Under **Github Settings > Codespaces** you can set GitHub Secrets

| Codespaces secrets                                                                                                                                                                  |                            |                         |                        | New secret             |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------|-------------------------|------------------------|------------------------|
| Development environment secrets are environment variables that are <b>encrypted</b> . They are available to any codespace you create using repositories with access to that secret. |                            |                         |                        |                        |
|  AWS_ACCESS_KEY_ID                                                                                 | Available to 1 repository. | Updated on Nov 23, 2023 | <a href="#">Update</a> | <a href="#">Delete</a> |
|  AWS_REGION                                                                                        | Available to 1 repository. | Updated on Nov 23, 2023 | <a href="#">Update</a> | <a href="#">Delete</a> |
|  AWS_SECRET_ACCESS_KEY                                                                           | Available to 1 repository. | Updated on Nov 23, 2023 | <a href="#">Update</a> | <a href="#">Delete</a> |

Secrets allows you to **set sensitive environment variables that will be loaded into your codespace**  
This avoids hardcoding secrets, passwords or sensitive values into your codebase.

# Dev Containers Configuration

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

Visual Studio Code Dev Containers allows you to **configure your docker container via json file.**

**devcontainer.json** file is expected in the root of your project.



```
{
 "name": "My Development Container",
 "image": "mcr.microsoft.com/vscode/devcontainers/base:0-focal",
 "settings": {
 "terminal.integrated.shell.linux": "/bin/bash"
 },
 "extensions": [
 "ms-python.python",
 "ms-vscode.cpptools"
],
 "forwardPorts": [3000],
 "postCreateCommand": "echo 'Container ready!'",
 "remoteUser": "vscode"
}
```

# Dev Containers Configuration

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

You can **provide remote image** from a repo

```
"image": "mcr.microsoft.com/vscode/devcontainers/base:0-focal"
```

```
"build": {
 "dockerfile": "Dockerfile",
 "args": {
 "VARIANT": "3.8",
 "INSTALL_NODE": "true"
 },
},
```

You can **build your own image** by including the Dockerfile with your project

# Dev Containers Configuration

Cheat sheets, Practice Exams and Flash cards [👉 www.exampro.co/ghf](http://www.exampro.co/ghf)

To **custom configure program** generally you will install these via your custom Dockerfile

```
"features": {
 "github-cli": "latest",
 "node": {
 "version": "lts"
 },
 "python": {
 "version": "3.8"
 }
},
```

**Features** allows you to quickly install programs on top of a common base image.



You can create your own features to quickly install on top of your own base image.

```
FROM python:3.8-slim
WORKDIR /usr/src/app
ENV TF_VER=1.0.0
ENV TF_PREFIX=https://releases.hashicorp.com/terraform
RUN apt-get update && apt-get install -y --no-install-recommends \
curl \
unzip \
git \
&& rm -rf /var/lib/apt/lists/*

RUN curl -o terraform.zip ${TF_PREFIX}/${TF_VER}/terraform_${TF_VER}_linux_amd64.zip \
&& unzip terraform.zip \
&& mv terraform /usr/local/bin/ \
&& rm terraform.zip

COPY . .
EXPOSE 80
ENV NAME World
CMD ["python", "app.py"]
```

You could use the `postCreateCommand` to **install a bunch of programs** ontop of your base container.

```
"postCreateCommand": "bash ./devcontainer/postCreateCommand.sh",
```

# Dev Containers Configuration

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

Settings allows you to configure the **settings of VS Code**

Eg. Setting editor font and terminal size

```
"settings": {
 "terminal.integrated.shell.linux": "/bin/bash",
 "python.pythonPath": "/usr/local/bin/python"
},
"extensions": [
 "ms-python.python",
 "ms-vscode.cpptools",
 "github.vscode-pull-request-github"
],
```

**Extensions** allows you to install VS Code extensions  
eg. VIM style keyboard controls

```
"forwardPorts": [3000, 5000],
```

You can list **ports you always want to forward**

```
"runArgs": [
 "-e", "MY_ENV_VAR=myvalue",
 "-e", "ANOTHER_ENV_VAR=anothervalue",
 "-v", "/my/host/folder:/container/folder",
 "--network=host",
 "--user=$(id -u):$(id -g)",
 "--cpus=2.0",
 "--memory=2g",
 "--shm-size=1g",
 "--add-host=example.com:192.168.1.10",
 "--privileged",
 "--security-opt", "seccomp=unconfined",
 "--network-alias=myalias",
 "-e", "TZ=Europe/London",
 "-v", "/var/run/docker.sock:/var/run/docker.sock"
]
```

# Dev Containers Configuration

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

## Environment Variables

we want to pass to the running container



```
"containerEnv": {
 "NODE_ENV": "development",
 "PYTHONUNBUFFERED": "1",
 "TZ": "America/New_York",
 "LANG": "en_US.UTF-8",
 "HTTP_PROXY": "http://proxy.example.com:8080",
 "HTTPS_PROXY": "http://proxy.example.com:8080",
 "NO_PROXY": "localhost,127.0.0.1,.example.com",
 "JAVA_HOME": "/usr/lib/jvm/java-11-openjdk-amd64",
 "MAVEN_HOME": "/usr/share/maven",
 "DOCKER_TLS_VERIFY": "1",
 "DOCKER_HOST": "tcp://docker.example.com:2376",
 "DOCKER_CERT_PATH": "/certs/client",
 "DATABASE_URL": "postgresql://user:password@localhost:5432/mydatabase",
 "REDIS_URL": "redis://localhost:6379",
 "API_KEY": "your-api-key",
 "LOG_LEVEL": "debug"
}
```

# Dev Containers Configuration

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

Mounts allows you to **map files and folders to your containers**

```
"mounts": [
 "source=${localWorkspaceFolder}/data,target=/container/data,type=bind,consistency=cached",
 "source=myvolume,target=/container/volume,type=volume",
 "source=/var/run/docker.sock,target=/var/run/docker.sock,type=bind",
 "source=${localWorkspaceFolder}/config/settings.json,target=/container/settings.json,type=bind,consistency=cached",
 "source=${localWorkspaceFolder}/docs,target=/container/docs,type=bind,consistency=cached,readonly"
]
```

- **Bind Mount a Local Directory:** Mount a local directory for direct file access.
- **Mount a Docker Volume for Persistent Storage:** Use a Docker volume for data that persists across container rebuilds.
- **Mount the Docker Socket:** Allow the container to interact with the host's Docker daemon.
- **Mount a Specific File:** Mount a specific configuration file into the container.
- **Mount a Host Directory as ReadOnly:** Mount a directory from the host as read-only.

# Dev Containers Configuration

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

command that **runs after creating the container**.

```
"postCreateCommand": "echo 'Container ready!',
"remoteUser": "vscode"
```

the **default username** that VS Code should use when connecting to the container.

You could use the postCreateCommand to **install a bunch of programs** ontop of your base container.

```
"postCreateCommand": "bash ./devcontainer/postCreateCommand.sh",
```

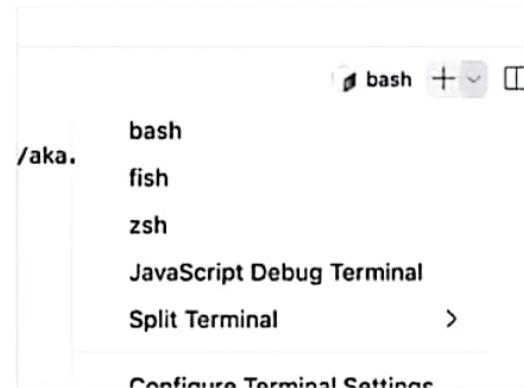
# Changing Shell

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

Github Codespaces VS Code Browser

default devcontainer images comes with:

- Bash (default)
- Zsh
- Fish



If you build or provide an image to devcontainer.json with other shells installed you can define and provide the path to that shell for use:



```
{
 "terminal.integrated.profiles.linux": {
 "csh": {
 "path": "csh"
 }
 },
 "terminal.integrated.defaultProfile.osx": "zsh",
 "terminal.integrated.defaultProfile.linux": "bash",
 "terminal.integrated.defaultProfile.windows": "PowerShell"
}
```

You can configure the settings.json in the VSCode to change the defaults based on Operating System.



# GitHub.dev Editor

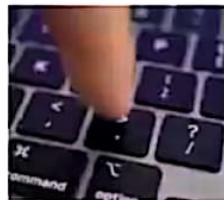
Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

Github.dev Editor is a **VS Code browser** that instantly loads that has no attached compute.



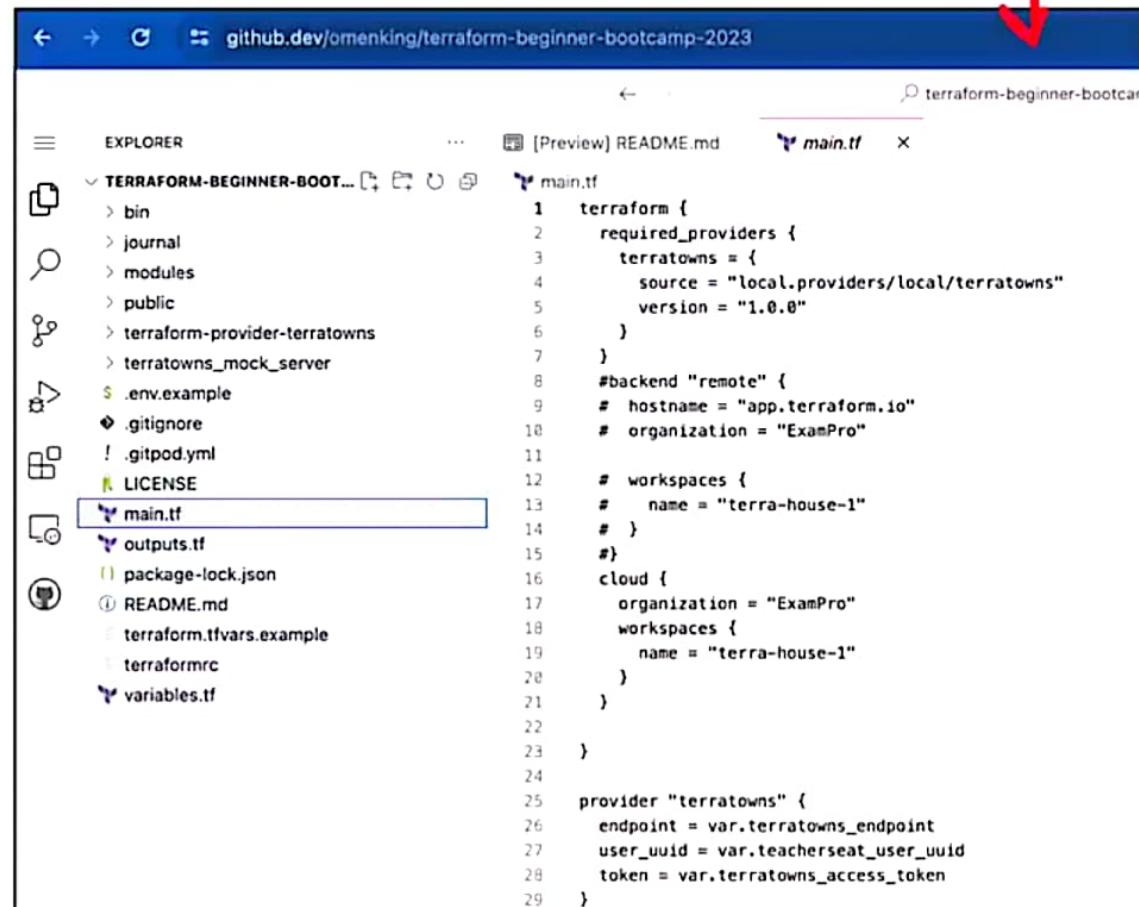
There are two ways you can open your Github repo in Github.dev:

1. Press the period (.) hotkey on your keyboard when you have your Github Repo page opened.



2. Place your Github Path after the url:

github.dev/  /



```
1 terraform {
2 required_providers {
3 terratowns = {
4 source = "local.providers/local/terratowns"
5 version = "1.0.0"
6 }
7 }
8 #backend "remote" {
9 # hostname = "app.terraform.io"
10 # organization = "ExamPro"
11 }
12 # workspaces {
13 # name = "terra-house-1"
14 # }
15 #
16 cloud {
17 organization = "ExamPro"
18 workspaces {
19 name = "terra-house-1"
20 }
21 }
22 }
23
24 provider "terratowns" {
25 endpoint = var.terratowns_endpoint
26 user_uuid = var.teacherseat_user_uuid
27 token = var.terratowns_access_token
28 }
29 }
```



Github.dev is intended for quickly changes to file and committing code in an IDE experience.

# GitHub.dev vs GitHub Codespaces

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

|                 | GitHub.dev                                                                                                          | Codespaces                                                                                                                                                                                 |
|-----------------|---------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Cost            | FREE                                                                                                                | <ul style="list-style-type: none"><li>• Free monthly quota</li><li>• Storage and Core hours per month<ul style="list-style-type: none"><li>• Machine type changes cost</li></ul></li></ul> |
| Availability    | Available to everyone on GitHub.com.                                                                                |                                                                                                                                                                                            |
| Start up        | <ul style="list-style-type: none"><li>• Instantly launches.</li><li>• No devcontainer.json configuration</li></ul>  | <ul style="list-style-type: none"><li>• Takes a few minutes to startup</li><li>• Does load devcontainer.json</li></ul>                                                                     |
| Compute         | <ul style="list-style-type: none"><li>• There is no attached compute</li><li>• You can't run code or apps</li></ul> | <ul style="list-style-type: none"><li>• Dedicated attached VM</li><li>• Run and debug your code and apps</li></ul>                                                                         |
| Terminal access | No terminal                                                                                                         | Terminal access                                                                                                                                                                            |
| Extensions      | Limited subset of extensions                                                                                        | Most extensions from VS Code Marketplace                                                                                                                                                   |

# Open Source

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)



**Open-source is source code made freely available for possible modification and redistribution.**

Open source benefits include:

- Encourages global collaboration.
- Speeds up innovation.
- Offers adaptability and customization.
- Reduces software costs.
- Enhances learning for developers.
- Typically high-quality and reliable.
- Provides transparency for trust and security.

Open-source software has often have **free-community versions** which makes it easy for personal developers or small organizations to quickly adopt technology.

| Ranking | Project                       | Leading company          | Market value   |
|---------|-------------------------------|--------------------------|----------------|
| 1       | <a href="#">Linux</a>         | <a href="#">Red Hat</a>  | \$16 billion   |
| 2       | <a href="#">Git</a>           | <a href="#">GitHub</a>   | \$2 billion    |
| 3       | <a href="#">MySQL</a>         | <a href="#">Oracle</a>   | \$1.87 billion |
| 4       | <a href="#">Node.js</a>       | NodeSource               | ?              |
| 5       | <a href="#">Docker</a>        | Docker                   | \$1 billion    |
| 6       | <a href="#">Hadoop</a>        | <a href="#">Cloudera</a> | \$3 billion    |
| 7       | <a href="#">Elasticsearch</a> | Elastic                  | \$700 million  |
| 8       | <a href="#">Spark</a>         | Databricks               | \$513 million  |
| 9       | <a href="#">MongoDB</a>       | MongoDB                  | \$1.57 billion |
| 10      | <a href="#">Selenium</a>      | Sauce Labs               | \$470 million  |

# Open Source

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)



The Open Source Initiative (OSI) is a non-profit organization is the steward of the Open Source Definition, the set of rules that define open source software

THE OSI maintains **a list of open-source licensing documents** for all different use-cases which are readability available for project owners to adopt

The Top five most popular open-source licenses:

- **MIT License:**
  - Known for its simplicity and permissiveness.
- **GNU General Public License (GPL):**
  - Ensures that modified versions remain open.
- **Apache License 2.0:**
  - Provides explicit patent rights grants.
- **GNU Lesser General Public License (LGPL):**
  - Similar to GPL but with some allowances for linking to proprietary software.
- **BSD License:**
  - Characterized by minimal restrictions on redistribution.

| License Name                                         |
|------------------------------------------------------|
| <a href="#">1-clause BSD License</a>                 |
| <a href="#">Academic Free License v. 3.0</a>         |
| <a href="#">Adaptive Public License 1.0</a>          |
| <a href="#">Apache License, Version 2.0</a>          |
| <a href="#">Apache Software License, version 1.1</a> |

# GitHub Sponsors

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

**Github Sponsors allows GitHub users to collect donations for their Github hosted own-source projects.**

Sponsorship payments are facilitated two ways:

- Payments through GitHub on Github.com
- Payments through Patreon on Github.com

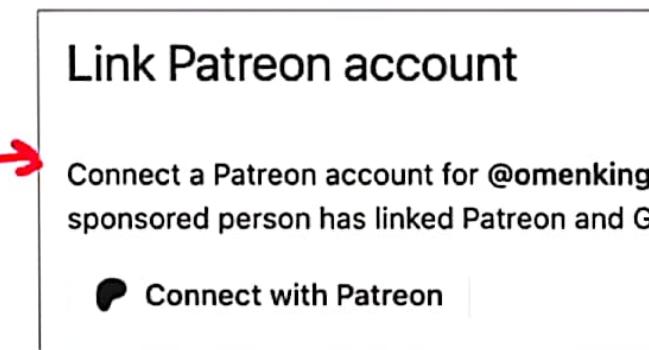
You can turn sponsor on for a **specific repo**

**Sponsorships** ✓  
Sponsorships help your community know how to financially support this repository.

Display a "Sponsor" button  
Add links to GitHub Sponsors or third-party methods your repository accepts for financial contributions to your project. [Set up sponsor button](#)



You can **connect Patreon** via your Account settings



# GitHub Sponsors

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

To receive sponsorship through GitHub on Github.com you'll need to get accepted into **Github Sponsors**.  <https://github.com/sponsors>



omenking

This account has not applied to join GitHub Sponsors.

[Join the waitlist](#)

**Personal accounts** can apply



ExamProCo Organization

This account has not applied to join GitHub Sponsors.

[Join the waitlist](#)

**Organizational accounts** can apply



**Organization bank account**

This organization uses their own bank account

You can use your organization's bank account or a personal bank account. Note that you are responsible for all applicable taxes on payouts. You may tax your GitHub Sponsors payouts as personal income. [Learn more](#)

Country or region where your bank account is located:

Select a country or region 

This organization is using a fiscal host

Organizations can use a fiscal host to join GitHub Sponsors instead. [Learn more about using a fiscal host.](#)

[Join waitlist](#)

Setting up GitHub sponsors via Github requires you to connect your bank account and may have **other fiscal requirements**.

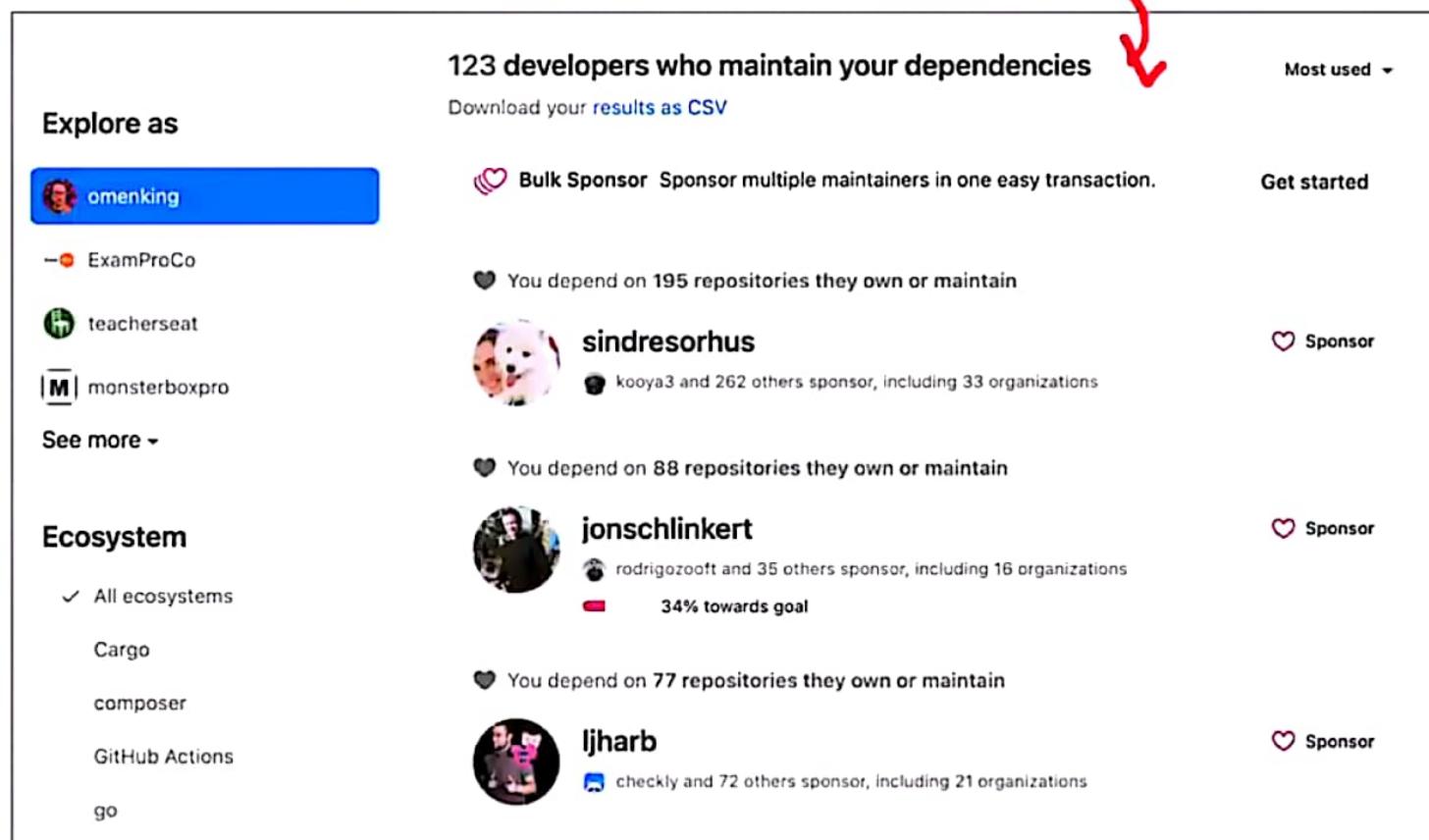


# GitHub Sponsors

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

GitHub makes it easy to locate people who maintains your dependencies so you can **support open-source contributors**

<https://github.com/sponsors/explore>



The screenshot shows the GitHub Sponsors explore page. At the top, it displays "123 developers who maintain your dependencies". A red arrow points from the URL above to this text. Below this, there's a "Most used" dropdown menu. On the left, there's a sidebar titled "Explore as" with a blue button for "omenking" and links for "ExamProCo", "teacherseat", "monsterboxpro", and a "See more" button. The main content area lists maintainers with their profiles, repository counts, and sponsorship status:

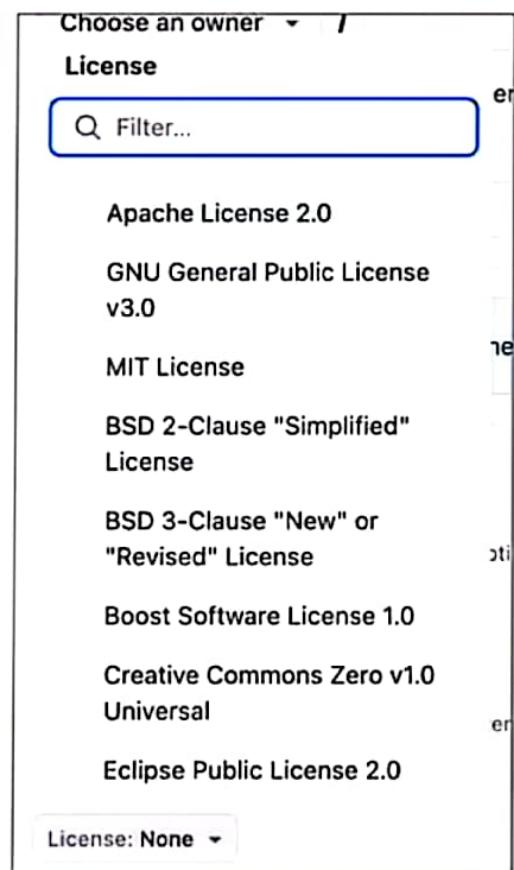
- sindresorhus**: You depend on 195 repositories they own or maintain. 
- jonschlinkert**: You depend on 88 repositories they own or maintain. 
- ljharb**: You depend on 77 repositories they own or maintain. 

Each maintainer entry includes a profile picture, their name, a summary of repositories, and a note about sponsors and organizations. There are also CSV download and get started links.

# GitHub and Open source projects

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

When creating a new repo GitHub makes it easy to  
**quickly add** a selected Open source license to your repo



Choose an owner

License

Filter...

- Apache License 2.0
- GNU General Public License v3.0
- MIT License
- BSD 2-Clause "Simplified" License
- BSD 3-Clause "New" or "Revised" License
- Boost Software License 1.0
- Creative Commons Zero v1.0 Universal
- Eclipse Public License 2.0

License: None

 README  Code of conduct  MIT license  Security

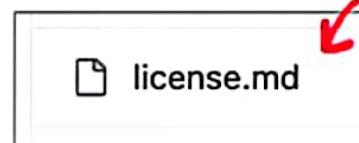
## The MIT License (MIT)

Copyright (c) 2024 Vercel, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.



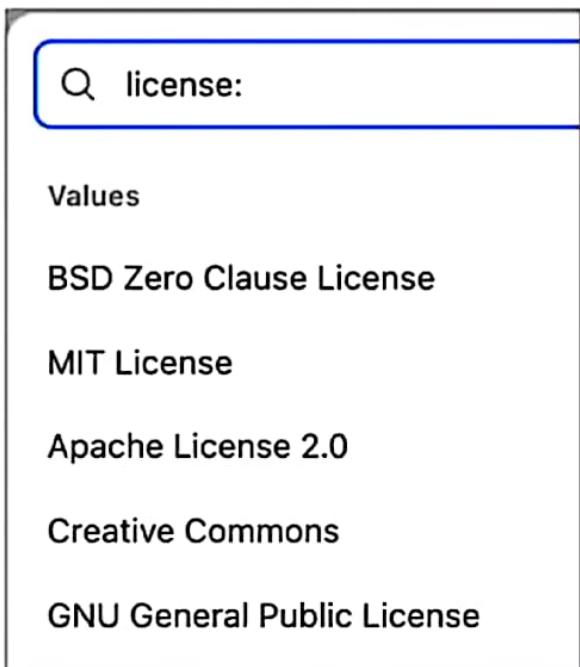
The file in the repo will be listed as either:

- license.md, LICENSE or LICENSE.md

# Open source discovery

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

GitHub **search** makes it easy to find public repo's based on Open source license



Many open-source GitHub repos are hosted or mirrored on GitHub.

Just type the open-source project name into search

645k results (422 ms)

ex [expressjs/express](#)

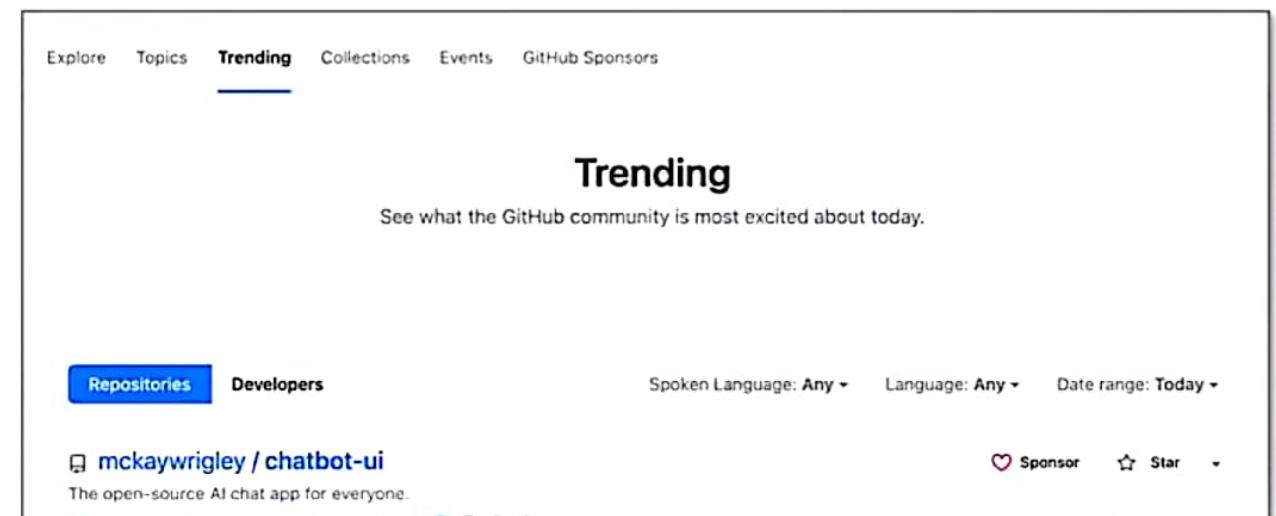
Fast, unopinionated, minimalist web framework for node.

nodejs javascript express server

JavaScript · ⭐ 62.7k · Updated 6 days ago

In Explore:

- Github Topics allows you to find repos for specific categories eg. Azure
- Github Trending: shows you most popular repos



A screenshot of the GitHub Trending page. At the top, there is a navigation bar with links: Explore, Topics, **Trending**, Collections, Events, and GitHub Sponsors. The main section is titled "Trending" with the sub-instruction "See what the GitHub community is most excited about today." Below this, there is a repository card for "mckaywrigley / chatbot-ui". The card includes the repository name, a brief description ("The open-source AI chat app for everyone."), and filtering options for "Repositories" and "Developers". At the bottom right, there are buttons for "Sponsor" and "Star".

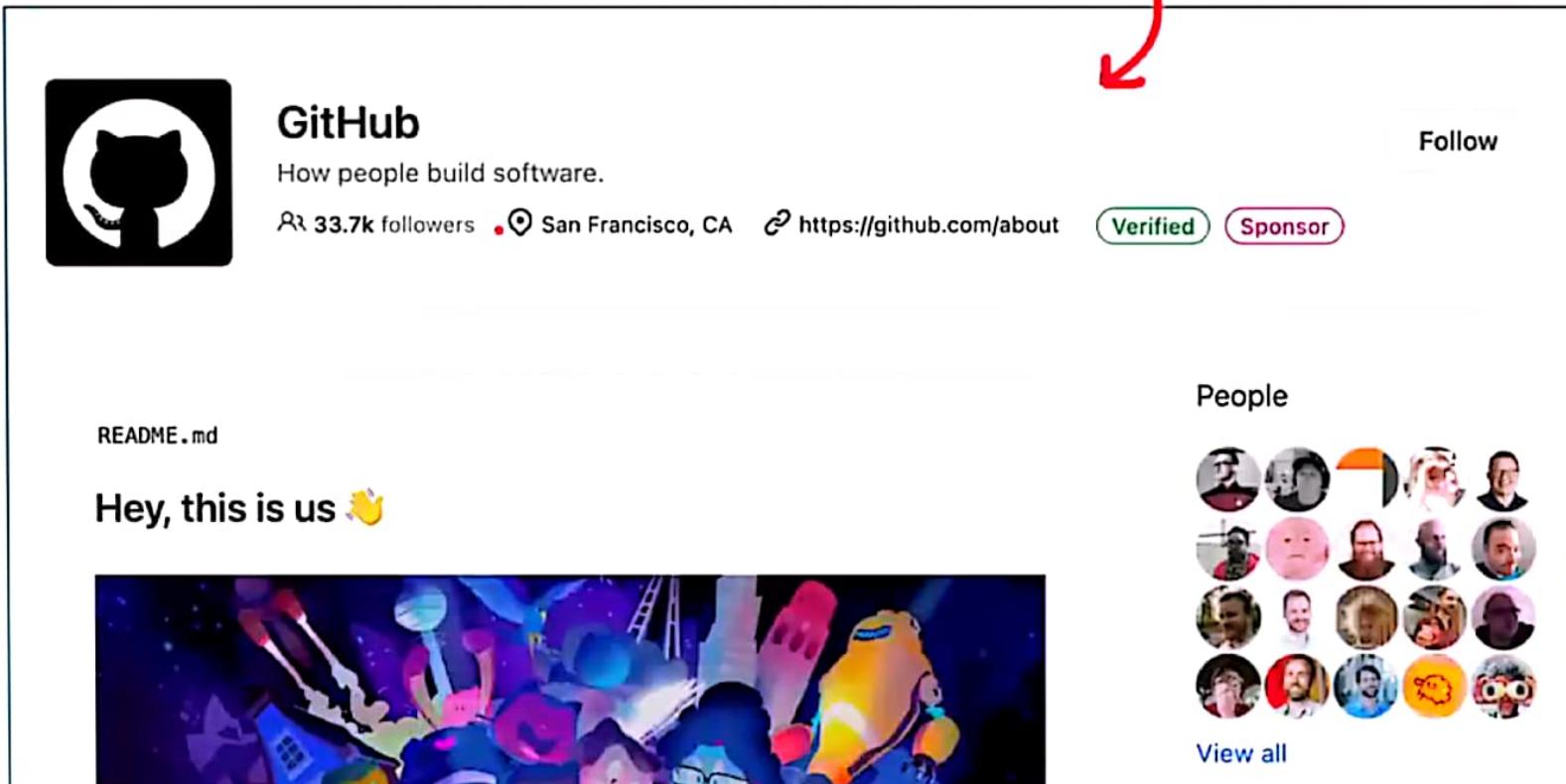
# Following Organizations

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

On a Github Organizations public profile you can **choose to follow** them to notifications of their public activity on your personal dashboard

This activity includes:

- new discussions
- sponsorships
- repositories



A screenshot of a GitHub organization profile page for "GitHub". The profile features the GitHub logo, the name "GitHub", the tagline "How people build software.", and statistics showing 33.7k followers from San Francisco, CA. A "Follow" button is visible on the right. Below the profile, there's a "README.md" section containing the text "Hey, this is us 🙌" and a colorful illustration of a diverse group of people. To the right, there's a "People" section showing a grid of user profiles and a "View all" link.

Follow

GitHub

How people build software.

33.7k followers • San Francisco, CA <https://github.com/about> Verified Sponsor

README.md

Hey, this is us 🙌

People

View all

# GitHub Marketplace

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

GitHub Marketplace are apps that **integrate with your GitHub repos** to provide additional functionality.

Apps can be **free or paid:**



### Pricing and setup

|                                                     |                            |
|-----------------------------------------------------|----------------------------|
| <b>Open Source</b><br>Free for open source projects | \$0                        |
| <b>Pro</b><br>Unlimited private repositories        | \$180<br>per user<br>/year |

**Codacy**  
**Open Source**  
Free for open source projects  
Account:   
Search by name

**Install it for free** Next: Confirm your installation location.

Codacy is provided by a third-party and is governed by separate [terms of service](#), [privacy policy](#), and [support contact](#).

**CircleCI**  
By circleci  
Automatically build, test, and deploy your project in minutes  
Recommended

**CodeFactor**  
By codefactor-io  
Automated code review for GitHub  
Recommended

**Imgbot**  
A GitHub app that optimizes your images  
Recommended

**Rewind Backups for GitHub**  
By backhub  
Automatic daily backups of your GitHub repos and metadata with on-demand restores to protect your business  
Recommended

**Codecov | Code Coverage**  
By codecov  
Automatic test report merging for all CI and languages into a single code coverage

**Zenhub**  
By ZenHubIO  
Agile Task Boards, Epics, Estimates and Reports, all within GitHub's UI

# InnerSource

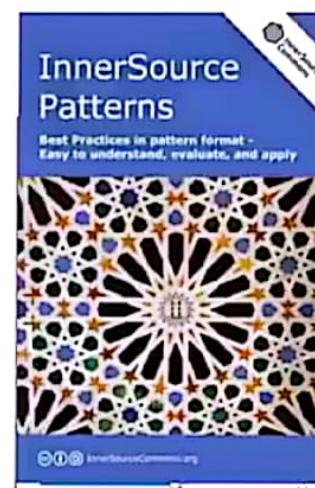
Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

**InnerSource** is organization and development best practices of non-open-source and/or proprietary software.

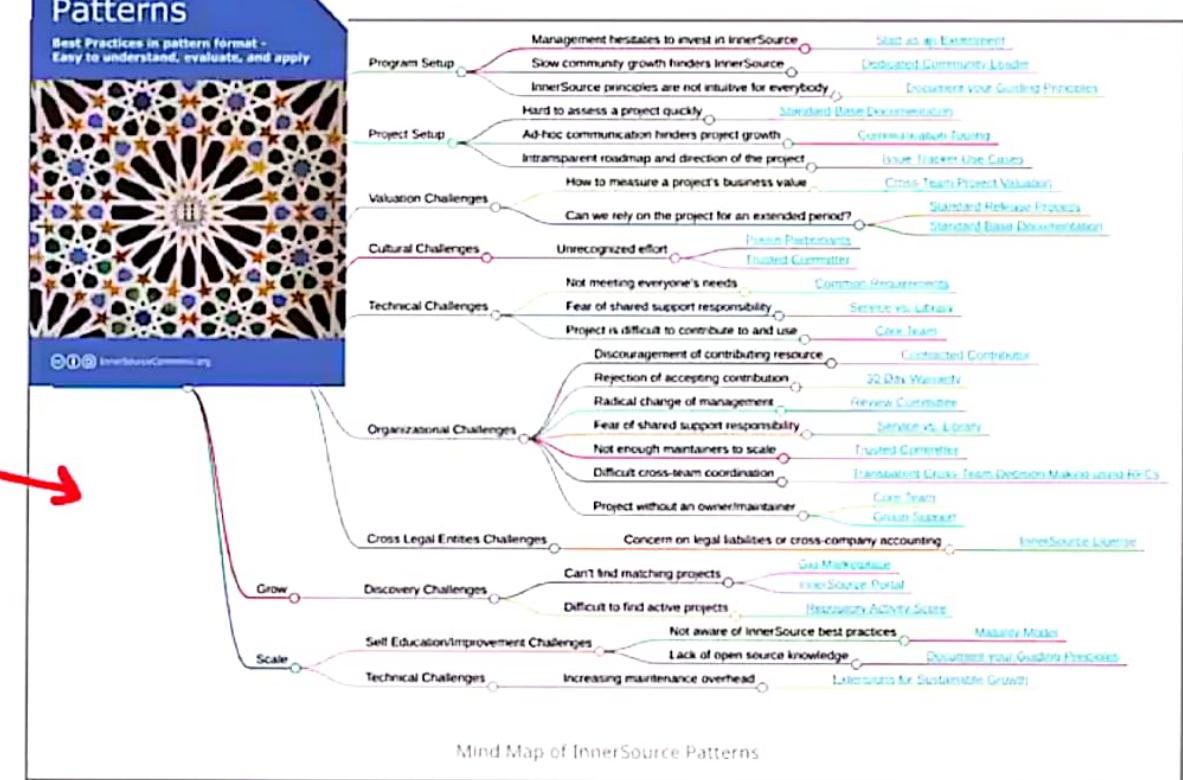


InnerSource is **not a strict guideline** but **a loose strategy** to establish an open source-like culture within organizations

Innersourcecommons.org provide several **free structured resources** to help understand how to adopt InnerSource



The term was coined by Tim O'Reilly in 2000



<https://patterns.innersourcecommons.org/explore-patterns>

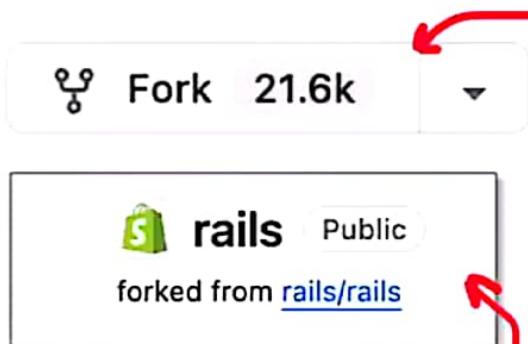
# InnerSource vs Open source

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

|               | InnerSource                                            | OpenSource                                                   |
|---------------|--------------------------------------------------------|--------------------------------------------------------------|
| Scope         | Within an organization                                 | Public, global community                                     |
| Accessibility | Limited to company employees                           | Open to anyone                                               |
| Purpose       | Improve collaboration and efficiency internally        | Share and collaborate on projects globally                   |
| Contribution  | Employees of the organization                          | Anyone (developers, users, etc.)                             |
| Visibility    | Code and discussions often private to the organization | Publicly available and visible                               |
| Governance    | Dictated by internal policies and culture              | Usually governed by open source licenses and community rules |

# Forking

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)



A forked repo is indicated  
underneath its repo name.

You may prefer using a forked  
repo over the original.

You may find original projects  
abandon, and a forks becoming  
the go-to repo for the project.

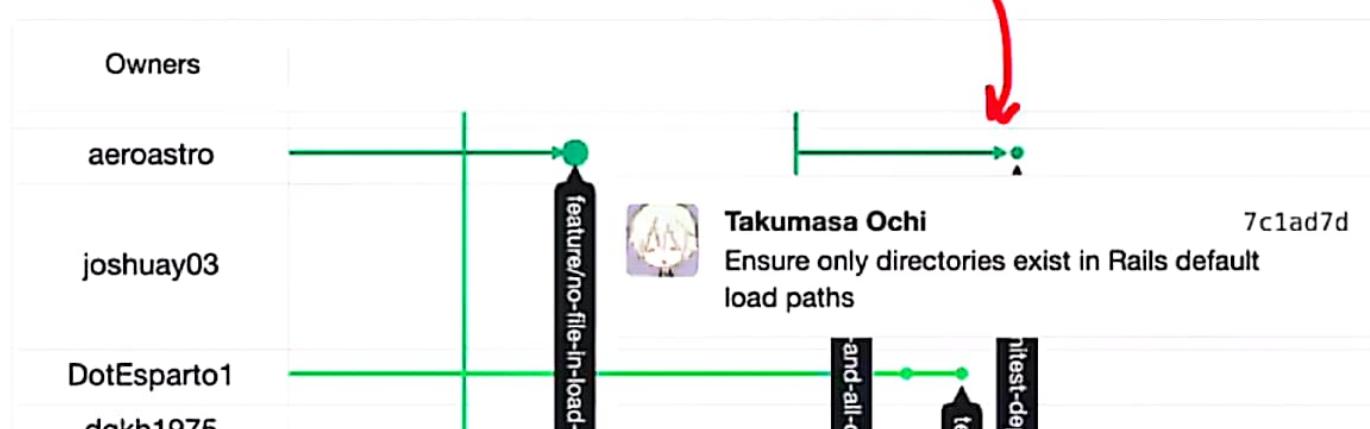
Fork's help keep public projects  
alive, on the edge and collaborative

**Forking repo's** allows you to create a copy of a repo.

Forking allows you to:

- Take a open-source or source available repo and go your own way
- Quickly apply bugs and patch fixes
- A separate repo to work on community contributions
  - You can create cross-repo pull-requests to get accepted in the original repo

Using **Network Graph** under Repo Insights you can often find  
community solutions to bugs or missing features.



# Discoverable repos

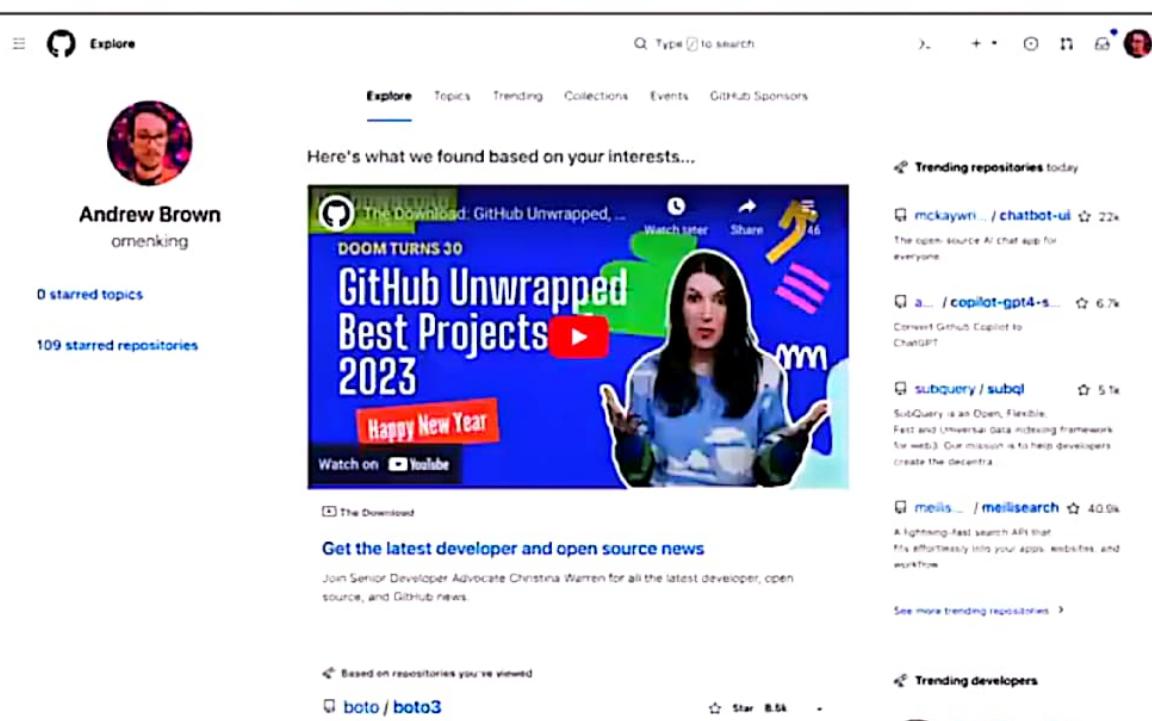
Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

Github Repos can be set as public, making repos easily searchable on GitHub and via search engines. Public repos serve a community purpose for knowledge sharing, educational resources and open-source projects.

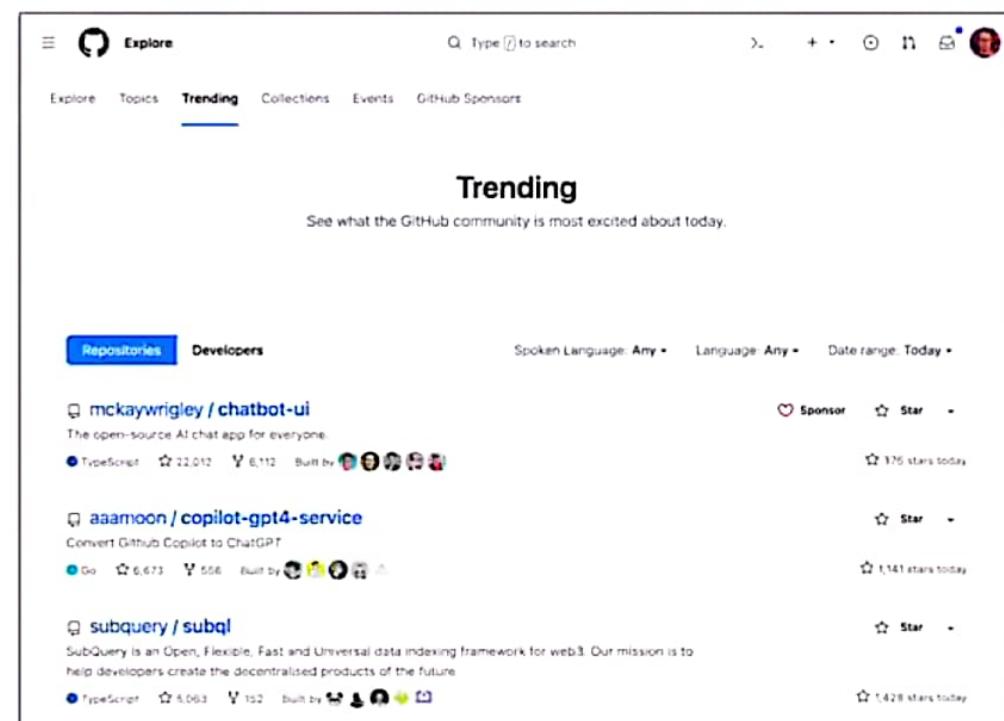
Besides just search Github curates content via their community pages:

<https://github.com/explore>

<https://github.com/trending>



The screenshot shows the GitHub Explore page. On the left, there's a sidebar with the user's profile picture, name (Andrew Brown), and links to starred topics and repositories. The main content area features a banner for "GitHub Unwrapped Best Projects 2023" with a video thumbnail of a woman speaking. Below the banner, it says "Here's what we found based on your interests...". It lists several trending repositories, such as "mckaywrigley / chatbot-ui" and "aaamoon / copilot-gpt4-service". At the bottom, there's a section for "Trending developers" and a link to "See more trending repositories".



The screenshot shows the GitHub Trending page. It has a header "Trending" and a sub-header "See what the GitHub community is most excited about today.". It lists trending repositories, with "mckaywrigley / chatbot-ui" at the top. Each repository entry includes the owner, name, stars, language, and a "Sponsor" or "Star" button. There are also filters for "Repositories" and "Developers", and dropdowns for "Spoken Language", "Language", and "Date range".

GitHub provides a **robust search** that lets us not only repo's names but the contents of their repos.

A screenshot of a GitHub search results page for the query "rails". The search bar at the top shows "rails". Below it, a "Filter by" sidebar includes options for Code, Repositories (selected), Issues, Pull requests, Discussions, Users, and More. Under Languages, there are filters for Ruby, JavaScript, HTML, CSS, and Shell. The main results section displays 479k results, sorted by Best match. The first result is "rails/rails" (8M), followed by "Ruby on Rails" (479k), "capistrano/rails" (378k), and "railsadminteam/rails\_admin" (1M). Each result card includes a thumbnail, the repository name, a star icon, a brief description, and links to the repository page, the organization page, and the topic page.

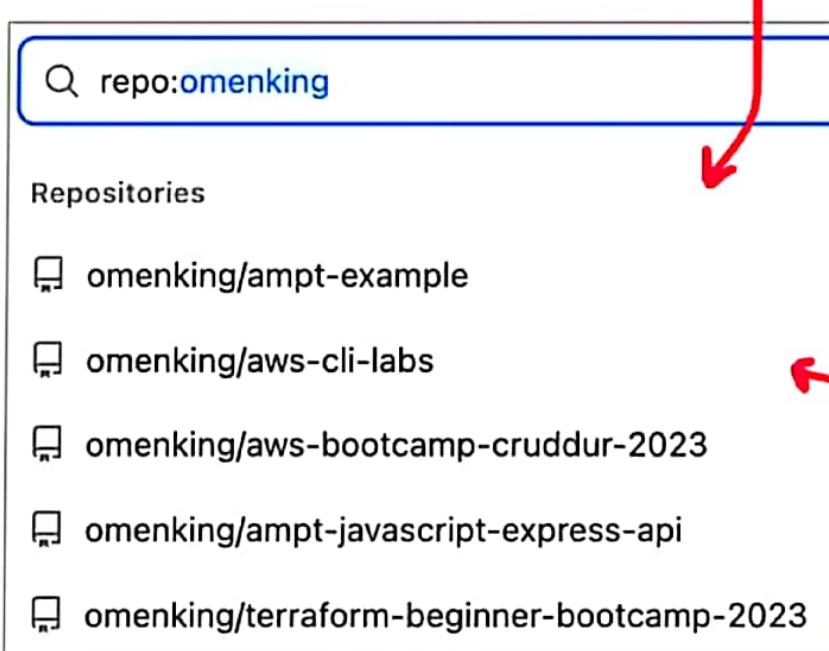
GitHub also has several **advanced search** options

A screenshot of the GitHub Advanced search sidebar. It lists various search filters, each preceded by a plus sign and a small icon: Owner, Size, Number of followers, Number of forks, Number of stars, Date created, Date pushed, Topic, License, Archived, Public, and Private. At the bottom of the sidebar is a link to "Advanced search". A red arrow points from the text "GitHub also has several advanced search options" to this sidebar.

# Discoverable repos – Search Syntax

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

GitHub global search box has a **search syntax** to help quickly apply advanced search options



A screenshot of the GitHub search interface. In the search bar at the top left, the query "repo:omenking" is entered. Below the search bar, the results are displayed under the heading "Repositories". Six repositories are listed, each with a small icon and the repository name: "omenking/ampt-example", "omenking/aws-cli-labs", "omenking/aws-bootcamp-cruddur-2023", "omenking/ampt-javascript-express-api", and "omenking/terraform-beginner-bootcamp-2023". A red arrow points from the "user:username" section of the search syntax table to the "omenking" part of the search bar.

| Search Syntax            | Description                            |
|--------------------------|----------------------------------------|
| language:python          | Finds repositories written in Python.  |
| stars:>100               | Repositories with more than 100 stars. |
| forks:>=50               | Repositories forked at least 50 times. |
| size:<1000               | Repositories smaller than 1000 KB.     |
| <b>user:username</b>     | Repositories owned by a specific user. |
| repo:username/repository | Specific repository from a user.       |
| filename:file.ext        | Search for a specific file.            |
| path:/docs/              | Files in a specific directory.         |
| extension:py             | Files with a .py extension.            |
| is:issue is:open         | Open issues.                           |
| is:pr is:closed          | Closed pull requests.                  |

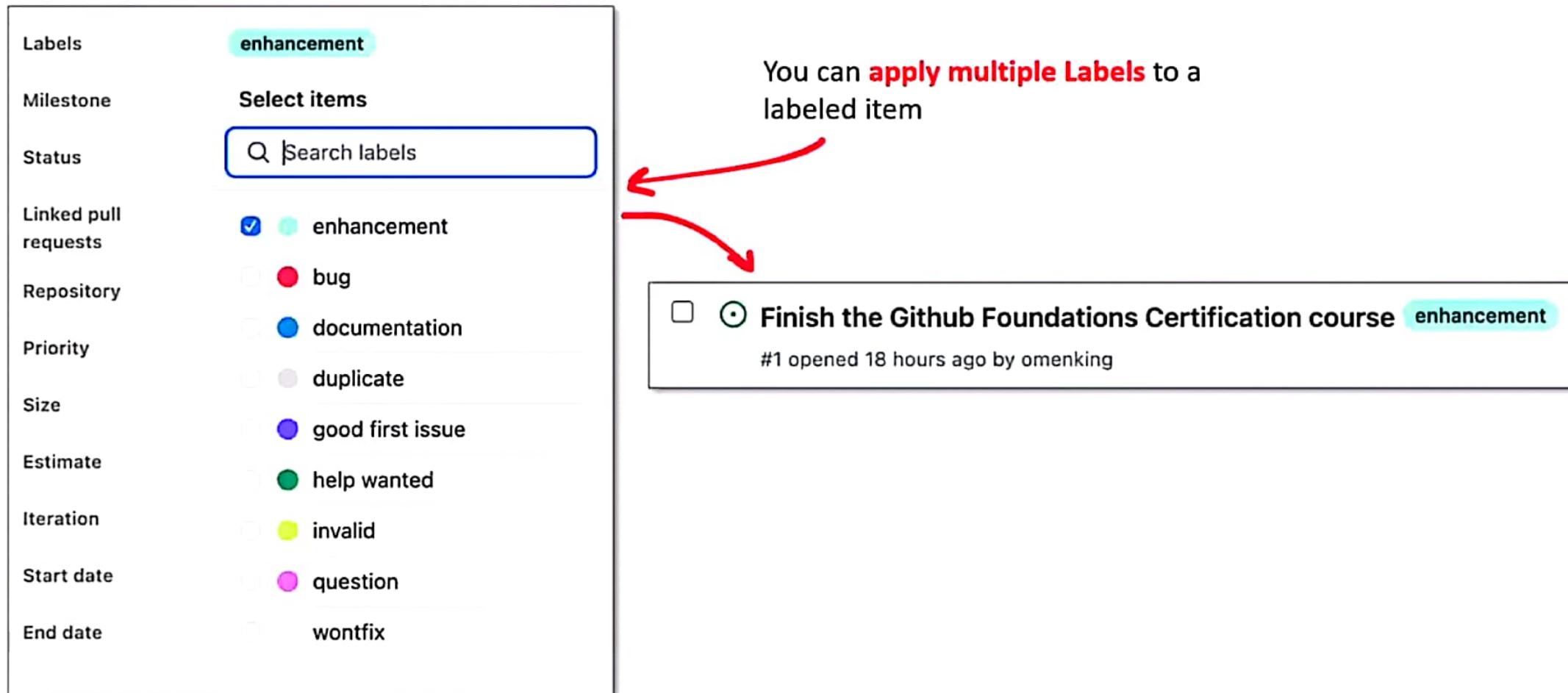


Search syntax also supports : **regular expressions, wildcards, NOT, OR AND, multiple terms and more**

# Labels

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

**GitHub Labels** are used to **categorize** issues, pull requests, and discussions



The image shows a screenshot of the GitHub Labels interface and a detailed view of a labeled issue.

**Labels Interface:**

- Labels: enhancement (highlighted)
- Milestone
- Status
- Linked pull requests
- Repository
- Priority
- Size
- Estimate
- Iteration
- Start date
- End date

**Select items**

- enhancement (checked)
- bug
- documentation
- duplicate
- good first issue
- help wanted
- invalid
- question
- wontfix

**Search labels:** Q Search labels

**Labeled Issue:**

You can **apply multiple Labels** to a labeled item

⚡ Finish the Github Foundations Certification course enhancement

#1 opened 18 hours ago by omenking

Annotations: Two red arrows point from the "enhancement" label in the interface to the "enhancement" label in the issue details.

# Labels

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

You can **manage your labels** or create new labels on the Labels page for your repo

<https://github.com/<username/org>/<repo-name>/labels>



The screenshot shows a GitHub repository interface. At the top, there's a navigation bar with links for Code, Issues (2), Pull requests (1), Discussions, Actions, Projects (1), Wiki, Security, Insights, and Settings. Below the navigation bar, there are tabs for Labels (which is selected and highlighted in blue), Milestones, and a search bar labeled "Search all labels". A green button labeled "New label" is located on the right side of the header. The main content area displays a list of 9 labels. Each label has a colored circular icon, a name, a description, and edit/delete buttons. The labels listed are: bug (red), documentation (blue), duplicate (grey), and enhancement (light blue). There are also several other labels whose names are partially visible or obscured by a blue horizontal bar at the bottom.

| Label         | Description                                | Actions                            |
|---------------|--------------------------------------------|------------------------------------|
| bug           | Something isn't working                    | Edit Delete                        |
| documentation | Improvements or additions to documentation | Edit Delete                        |
| duplicate     | This issue or pull request already exists  | Edit Delete                        |
| enhancement   | New feature or request                     | Convert to discussions Edit Delete |

# Labels

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

Label preview

Label name

Description



Color

 #b60205

Create label

Cancel

**Creating a label** as simple as providing a:

- Name
- Description
- Colour

Github CLI provides commands for labels

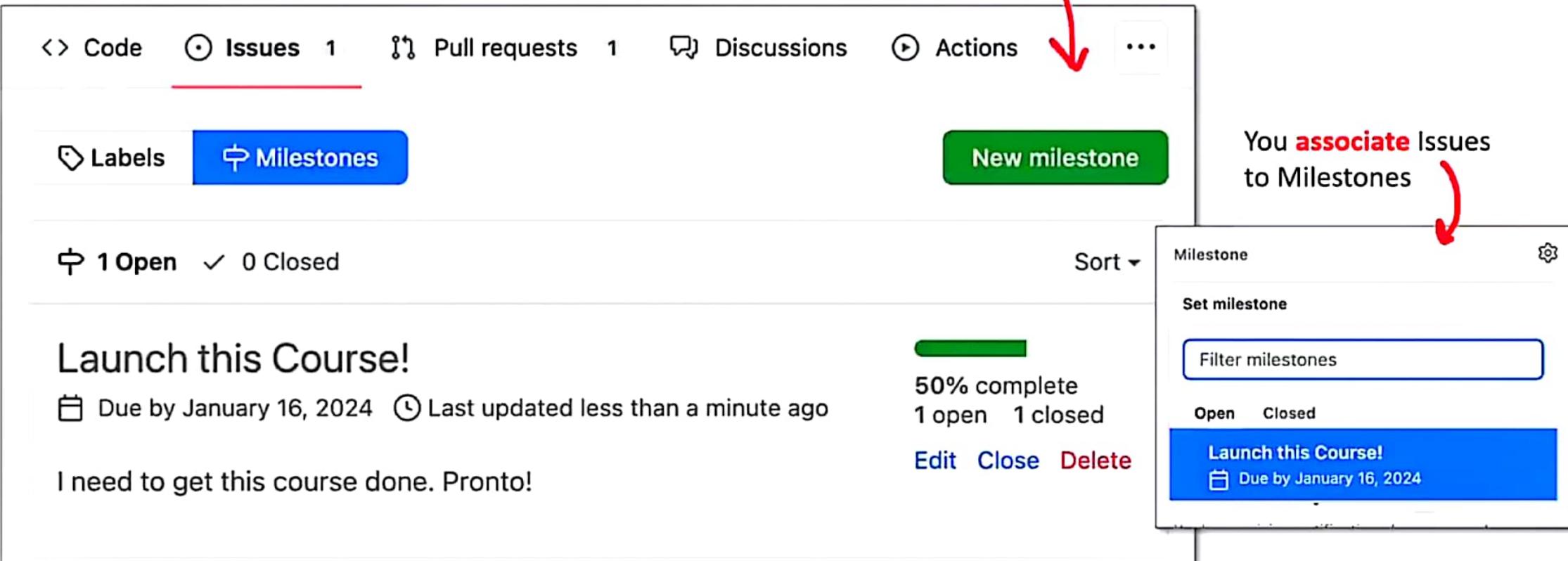
- gh label clone
- gh label create
- gh label delete
- gh label edit
- gh label list

```
gh label create bug \
--description "Something isn't working" \
--color E99695
```

# Milestones

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

Milestones allow you to **group multiple issues into an end goal** which show completion towards the goal for each closed Issue



The screenshot shows the GitHub Issues page for a repository. At the top, there are navigation links: Code, Issues (1), Pull requests (1), Discussions, Actions, and a three-dot menu. Below these are sections for Labels and Milestones. A red arrow points from the 'Issues' link to the 'Milestones' button, which is highlighted with a blue background and white text. Another red arrow points from the 'Milestones' button to a modal window titled 'Milestone'. The modal contains fields for 'Set milestone' (with a placeholder 'Filter milestones'), 'Open' (1), and 'Closed' (0). A large blue button at the bottom of the modal says 'Launch this Course!' with the note 'Due by January 16, 2024'. To the left of the main content, there's a summary: '1 Open' and '0 Closed'. Below this, a section titled 'Launch this Course!' lists a single issue: 'Due by January 16, 2024' (Last updated less than a minute ago). A note below says 'I need to get this course done. Pronto!'

Code Issues 1 Pull requests 1 Discussions Actions ...

Labels **Milestones**

1 Open 0 Closed

Sort ▾

New milestone

You associate Issues to Milestones

Milestone

Set milestone

Filter milestones

Open Closed

Launch this Course!

Due by January 16, 2024

Launch this Course!

Due by January 16, 2024

1 Open 1 closed

Edit Close Delete

I need to get this course done. Pronto!

# Introduction to GitHub Projects

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

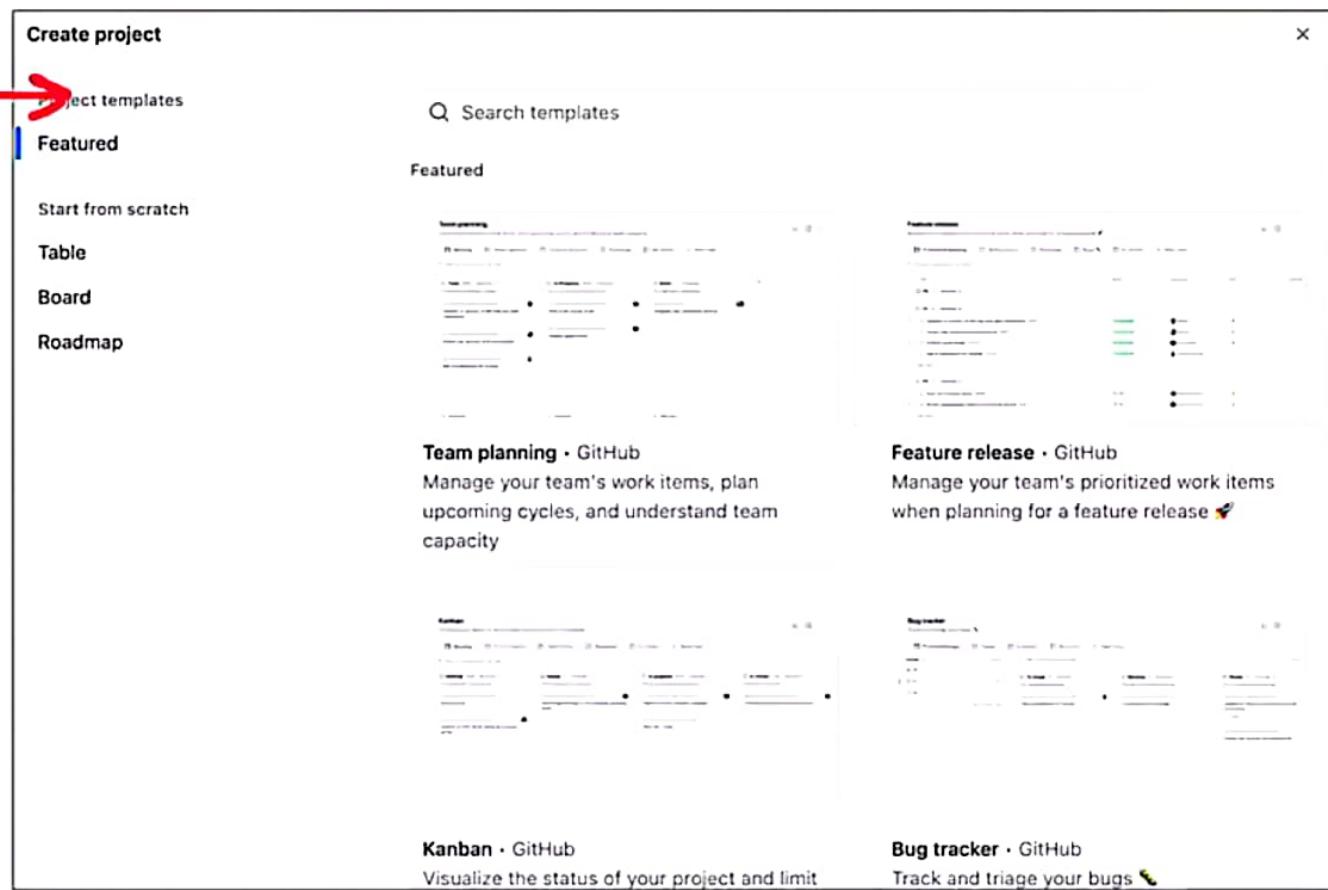
Github Projects is a **planning and tracking tool** when on working on GitHub Repo.

A project has an adaptable view  
that can be changed at anytime  
between:

- Spreadsheet (Table)
- Taskboard (Board)
- Roadmap (Roadmap)

Projects **directly integrate**  
with Issues and Pull Requests.

Projects have built-in  
workflows to **automate**  
**common actions**.

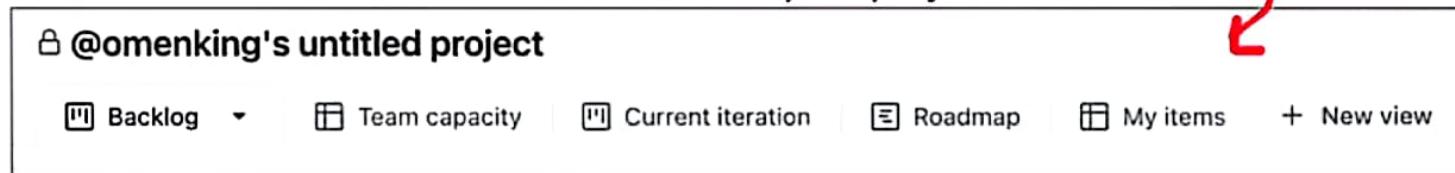


# GitHub Projects Layout Options

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

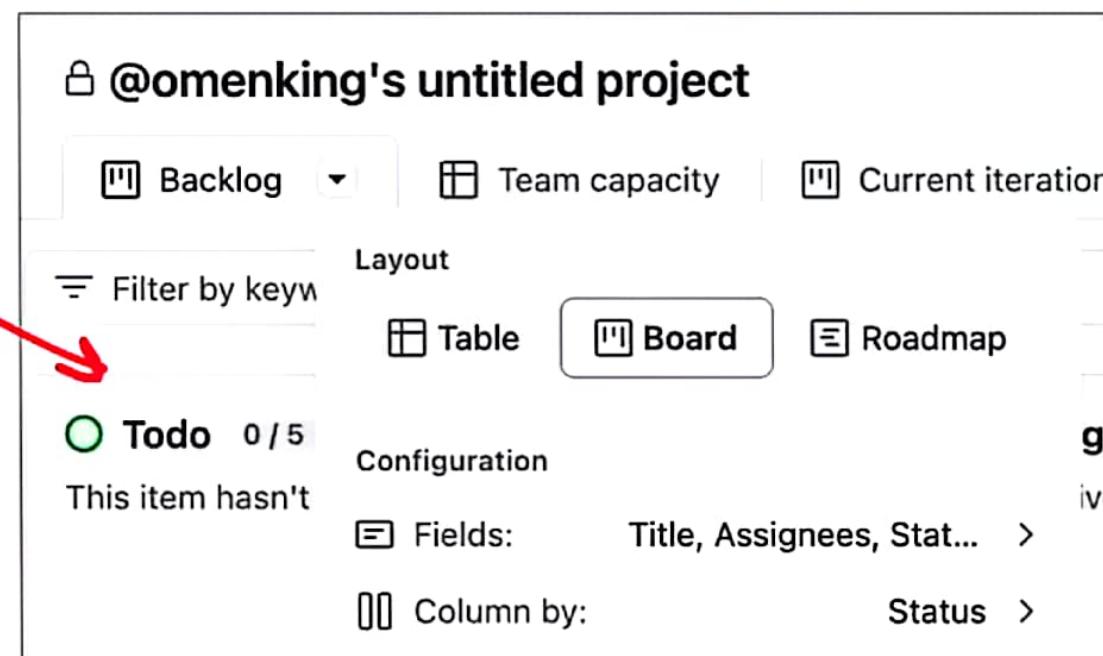
A GitHub Project is **composed of multiple views**

- You **add more views** to your project



The screenshot shows the top navigation bar of a GitHub Project. It includes a lock icon and the text '@omenking's untitled project'. Below the title are several buttons: 'Backlog' (with a dropdown arrow), 'Team capacity', 'Current iteration', 'Roadmap', 'My items', and a '+ New view' button. A red arrow points from the text 'You add more views to your project' to the '+ New view' button.

Each **view's layout can be changed at anytime** to accommodate the project use-case:



The screenshot shows the same GitHub Project interface, but with a different layout selected. The 'Board' button in the 'Layout' section is highlighted with a red border. The other options are 'Table' and 'Roadmap'. The rest of the interface remains the same, showing the backlog, team capacity, current iteration, and my items sections.

**Table** layout is great for your traditional ticket tracker and large amount of tasks



**Feature release**  
Manage your team's prioritized work items when planning for a feature release

Prioritized backlog Status board Roadmap Bugs In review New view

Filter by keyword or by field

| Title                                                      | Status      | Assignees      | Size | Estimate |
|------------------------------------------------------------|-------------|----------------|------|----------|
| 1 P0 2 Estimate: 2                                         |             |                |      |          |
| 2 P1 4 Estimate: 6                                         |             |                |      |          |
| 3 Updates to velocity of the ship and alien movements #802 | In progress | Dedac          | S    |          |
| 4 Create user account and subscription #820                | In progress | j0slepy        | S    |          |
| 5 Finalize upsell model #1323                              | In progress | marywhite      | M    |          |
| 6 Add a leaderboard for Europe #1378                       | In progress | rileybroughten | L    |          |
| + Add item                                                 |             |                |      |          |
| 7 P2 2 Estimate: 3                                         |             |                |      |          |
| 8 Save score across levels #800                            | To-do       | Mattamorphic   | M    |          |
| 9 Review engagement metrics once posts are live #25        | To-do       | ohioveryown    | S    |          |
| + Add item                                                 |             |                |      |          |

## Team planning

Manage your team's work items, plan upcoming cycles, and understand team capacity



Backlog Team capacity Current iteration Roadmap My items + New view

Filter by keyword or by field

### Todo 3/5 Estimate: 6

This item hasn't been started

planning-tracking-demo #802

Updates to velocity of the ship and alien movements

planning-tracking-demo #820

Create user account and subscription

planning-tracking-demo #1328

Add a leaderboard for Europe

### In Progress 2/5 Estimate: 2

This is actively being worked on

planning-tracking-demo #800

Save score across levels

planning-tracking-demo #1323

Finalize upsell model

### Done 1 Estimate: 2

This has been completed

OctoArcade #3

Integrate with Commerce Service

+ Add item

+ Add item

+ Add item

# GitHub Projects Layout Options

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

**Roadmap** layout is great for For planning based on a timeline



**Roadmap**

Roadmap + New view

Filter by keyword or by field

October 2023

GA Launch

Squad 1 7

Fri, Oct 6 - Mon, Oct 30

| Day | Task                                                               | Start Date   | End Date     | Description                                                        |
|-----|--------------------------------------------------------------------|--------------|--------------|--------------------------------------------------------------------|
| 1   | Updates to velocity of the ship and alien movements #802           | Oct 6, 2023  | Oct 24, 2023 | Updates to velocity of the ship and alien movements #802           |
| 2   | Create user account and subscription #820                          | Oct 9, 2023  | Oct 16, 2023 | Create user account and subscription #820                          |
| 3   | Finalize upsell model #1323                                        | Oct 12, 2023 | Oct 30, 2023 | Finalize upsell model #1323                                        |
| 4   | Blowing in to the cartridge does not fix the blinking screen #1378 | Oct 14, 2023 | Oct 18, 2023 | Blowing in to the cartridge does not fix the blinking screen #1378 |
| 5   | Add a leaderboard for Europe #1378                                 | Oct 16, 2023 | Oct 23, 2023 | Add a leaderboard for Europe #1378                                 |
| 6   | Create the design for login screen #39                             | Oct 16, 2023 |              | Create the design for login screen #39                             |
| 7   | Support light and dark themes #39                                  | Oct 17, 2023 |              | Support light and dark themes #39                                  |

+ Add item

Squad 2 14

Tue, Jan 31 - Thu, Oct 26

Markers Start date Date fields M

# GitHub Projects Layout Options

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

For your layouts you can **configure**  
**your data** based on multiple options



**Configuration**

- Group by: *none* >
- Markers: *none* >
- Sort by: *manual* >
- Dates: *Iteration* >
- Zoom level: *Month* >
- Slice by: *none* >

- **Fields:** Add custom data fields to each project item (e.g., status, assignees).
- **Column by:** Organize board columns based on specific fields (e.g., status).
- **Group by:** Group items into rows based on a chosen field (e.g., assignee, label).
- **Sort by:** Define the order of items within columns or groups (e.g., by priority).
- **Fields sum:** Show a summary or total of a particular field for all items in a column/group.
- **Slice by:** Filter visible items on the board by a specific field (e.g., milestone, label).
- **Zoom Level (Roadmap only):** Adjust the view of the roadmap timeline (daily, monthly, quarterly).

# GitHub Projects Configuration Options

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

Your Github Project Items  
have **custom fields**



Custom fields

- Status
- Priority
- Size
- Estimate
- Iteration
- Start date
- End date

+ New field

You can create **new fields**



Field name \*

Agileness

Field type

Aa Text

✓ Aa Text

1/3 Number

Date

Single select

Iteration

And provide **options** to easily fill in the fields.



Options

- Todo This item hasn't been started
- In Progress This is actively being worked on
- Done This has been completed
- Failed

Add option... Add

# Projects vs Projects Classic

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

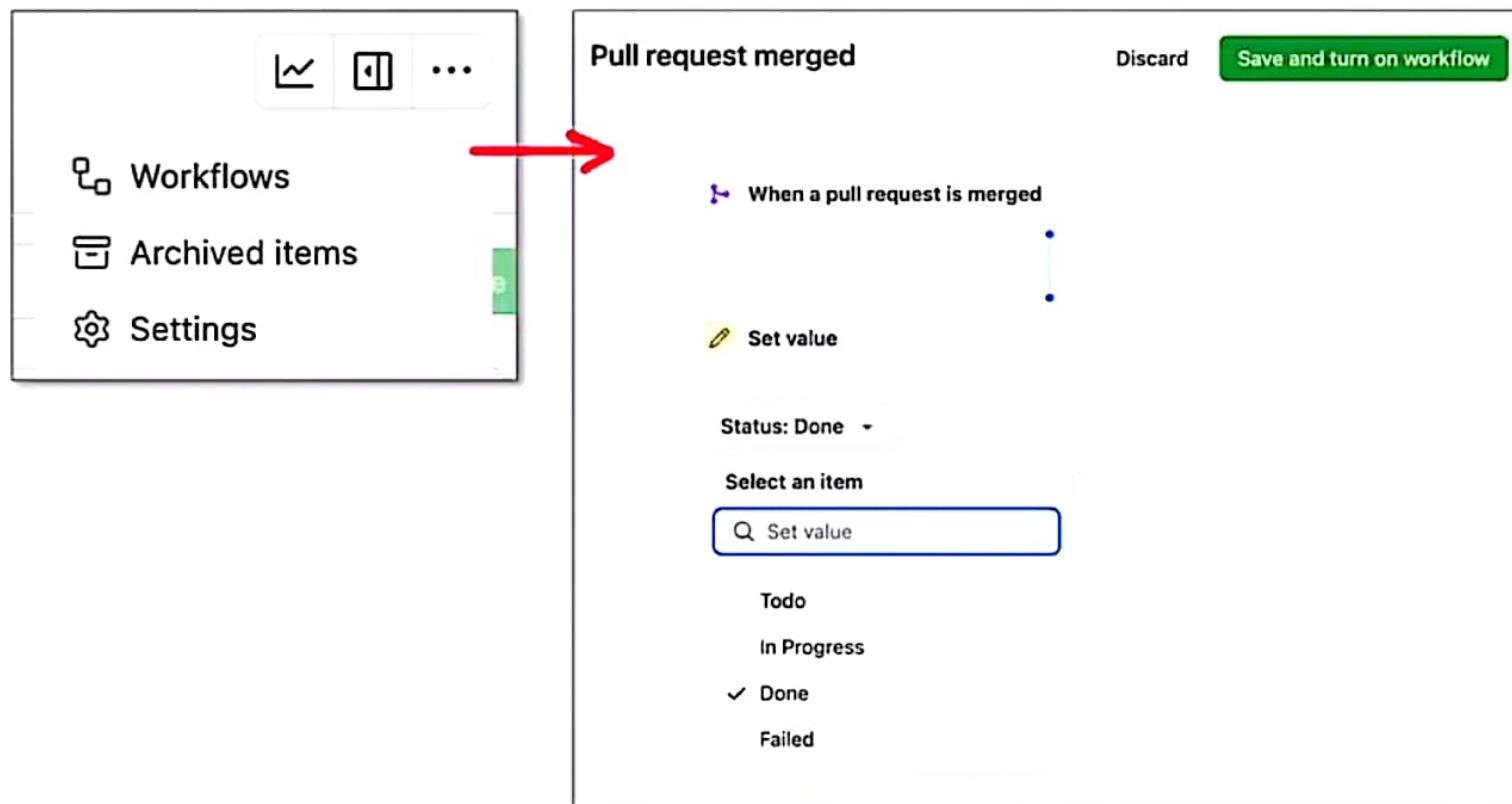
| GitHub Projects |                                                            | GitHub Projects <i>Classic</i>             |
|-----------------|------------------------------------------------------------|--------------------------------------------|
| Interface       | New, more dynamic interface with tables, boards, and views | Traditional board view similar to Trello   |
| Flexibility     | Highly customizable with different views and fields        | Limited customization options              |
| Automation      | Supports automated workflows and field updates             | Basic automation capabilities              |
| Reporting       | Enhanced reporting features and insights                   | Basic reporting and tracking               |
| Integration     | Deeper integration with GitHub issues and pull requests    | Basic integration with GitHub repositories |

GitHub Projects offers more advanced and flexible project management tools compared to GitHub Classic Projects, which is simpler and more straightforward.

# Project Workflows

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

GitHub built-in Project Workflows allows you to **automate what happens based on specific events**



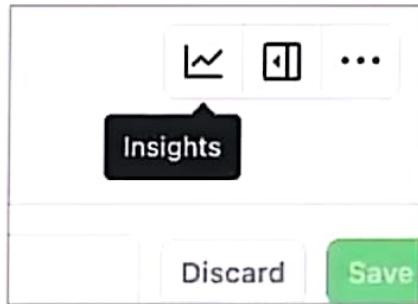
The screenshot shows the GitHub Project Workflows interface. On the left, there's a sidebar with icons for Workflows, Archived items, and Settings. A red arrow points from the 'Workflows' icon in the sidebar to the 'Pull request merged' step in the main panel. The main panel displays a 'Pull request merged' workflow step. It includes a 'When a pull request is merged' trigger, a 'Save and turn on workflow' button, and a 'Discard' button. Below the trigger, there's a 'Set value' section with a dropdown set to 'Status: Done'. A search bar labeled 'Select an item' is below it. At the bottom, there's a list of status options: Todo, In Progress, Done (which is checked), and Failed.

## Built-in Workflows

- Item added to project
- Item reopened
- Item closed
- Code changes requested
- Code review approved
- Pull request merged
- Auto-archive items
- Auto-add to project

# Project Insights

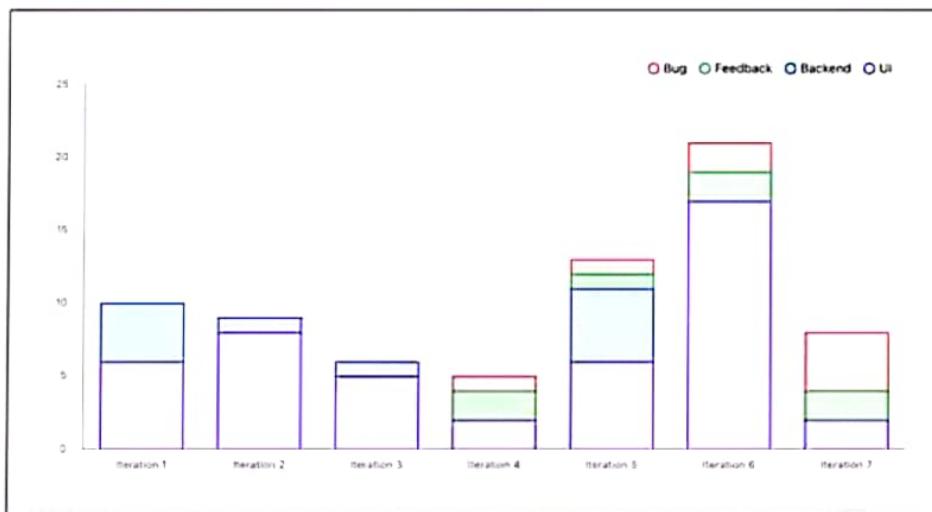
Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)



Project Insights lets you **create charts about your Github Project.**

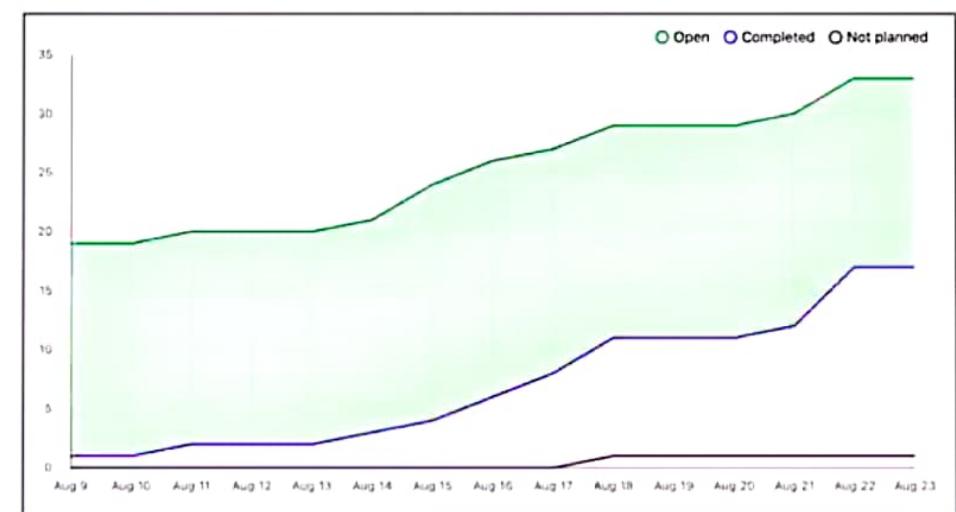
## Current Charts

e.g. how many items are assigned to each individuals



## Historical Charts

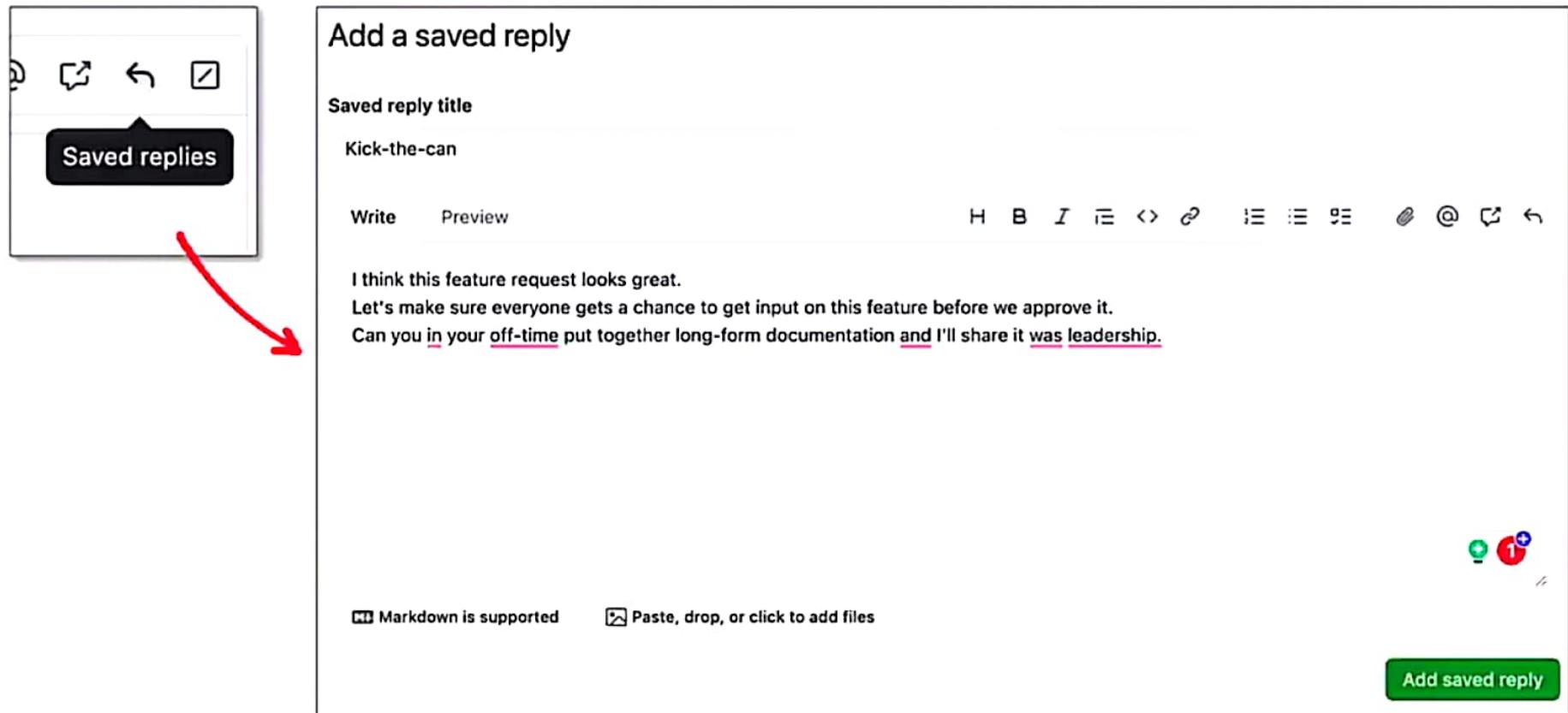
track changes to the state of your project items



# Managing Saved Replies

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

When commenting on an issue or pull request, you can **add a saved reply that you've already set up.**



**Add a saved reply**

Saved reply title  
Kick-the-can

Write Preview

I think this feature request looks great.  
Let's make sure everyone gets a chance to get input on this feature before we approve it.  
Can you in your off-time put together long-form documentation and I'll share it was leadership.

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

Add saved reply

Assignees

Assign up to 10 people to this issue

Type or choose a user

omenking Andrew Brown

omenking@omenking-ubuntu:~\$



You can **assign issues** and pull requests to specific users.  
That's it.

Projects ▾ Milestones ▾ Assignee ▾ Sort ▾

Filter by who's assigned X

Filter users

Assigned to nobody

omenking Andrew Brown



# Securing Account with 2FA

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)



**2-Factor Authentication (2FA)** is when you use a second device (eg. phone) to enter a code as an extra security precaution when signing-in or performing sensitive actions.

**Preferred 2FA method**  
Set your preferred method to use for 2FA

GitHub Mobile

Choose your preferred UFA:

- GitHub Mobile
- SMS/Text message
- Authenticator app



You can download third-party authenticator apps via your phone

- Authy
- Google Authenticator
- Microsoft Authenticator

Get UFA code sent as text message



Hardware devices like YubiKeys

Github's Mobile app

**Recovery codes** are long secret codes you need to print and store somewhere safe in case you get locked out.

## Two-factor methods

 **Authenticator app** Configured  
Use an authentication app or browser extension

 **SMS/Text message** Configured  
You will receive one-time codes at this phone number

## Security keys

Security keys are hardware devices that can be used for 2FA

 **GitHub Mobile** Configured 1 device  
GitHub Mobile can be used for two-factor authentication

## Recovery options

 **Recovery codes** Viewed  
Recovery codes can be used to access your account if you forget your password or lose your device.

# Access Permissions – Personal Accounts

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

Personal account has **two permission levels**:

- **Repository owner** – the person who owns the account, that's you!
  - Repositories owned by personal accounts have one owner
  - Ownership permissions can't be shared with another personal account.
- **Collaborators** – people you add as collaborators

## Repository owner actions — **has full control of the repository**

- Invite collaborators
- Change the visibility of the repository
- Limit interactions with the repository
- Rename a branch, including the default branch
- Merge a pull request on a protected branch, even if there are no approving reviews
- Delete the repository
- Manage the repository's topics
- Manage security and analysis settings for the repository
- Enable the dependency graph for a private repository
- Delete and restore packages
- Customize the repository's social media preview
- Create a template from the repository
- Control access to Dependabot alerts
- Dismiss Dependabot alerts in the repository
- Manage data use for a private repository
- Define code owners for the repository
- Archive the repository
- Create security advisories
- Display a sponsor button
- Allow or disallow auto-merge for pull requests
- Manage deploy keys
- Manage webhooks

# Access Permissions – Personal Accounts

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

**Collaborators actions — can pull (read) the contents of the repository and push (write) changes to the repository**

- Fork the repository
- Rename a branch other than the default branch
- Create, edit, and delete comments on commits, pull requests, and issues in the repository
- Create, assign, close, and re-open issues in the repository
- Manage labels for issues and pull requests in the repository
- Manage milestones for issues and pull requests in the repository
- Mark an issue or pull request in the repository as a duplicate
- Create, merge, and close pull requests in the repository
- Enable and disable auto-merge for a pull request
- Apply suggested changes to pull requests in the repository
- Create a pull request from a fork of the repository
- Submit a review on a pull request that affects the mergeability of the pull request
- Create and edit a wiki for the repository
- Create and edit releases for the repository
- Act as a code owner for the repository
- Publish, view, or install packages
- Remove themselves as collaborators on the repository

## **Repository owner actions**

has full control of the repository

## **Collaborators actions**

can pull (read) the contents of the repository  
and push (write) changes to the repository

VS

# EMUs

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

**Enterprise Managed Users (EMUs)** allows you to **manage the lifecycle and authentication of your users on GitHub.com** from an **external identity management system**, or IdP.

- GitHub partners with some developers of identity management systems to provide a "paved-path" integration with EMUs
- These IdPs mostly provide authentication using SAML.
- Microsoft Entra ID (aka Azure AD) also offers OIDC for authentication.
- The IdP applications provision users with System for Cross-domain Identity Management (SCIM).

| Partner IdP                                                                                           | SAML                                                                                 | OCID                                                                                  | SCIM                                                                                  |
|-------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
|  Microsoft Entra ID |   |   |   |
|  Okta              |  |  |  |
|  PingFederate      |  |  |  |

# Managing Features

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

GitHub Repo have **multiple features we can enable or disable:**

- Wikis
- Issues
- Sponsorships
- Preserve this repo
  - Goes into GitHub Arctic Code Vault
- Discussions
- Projects
- and possibly more...



## Features

- Wikis**  
Wikis host documentation for your repository.
- Restrict editing to collaborators only**  
Public wikis will still be readable by everyone.
- Issues**  
Issues integrate lightweight task tracking into your repository. Keep projects on track with issue labels and milestones, and reference them in commit messages.
- Sponsorships**  
Sponsorships help your community know how to financially support this repository.
- Preserve this repository**  
Include this code in the [GitHub Archive Program](#).
- Discussions**  
Discussions is the space for your community to have conversations, ask questions and post answers without opening issues.
- Projects**  
Projects on GitHub are created at the repository owner's level (organization or user) and can be linked to a repository's Projects tab. Projects are suitable for cross-repository development efforts such as feature work, complex product roadmaps or even issue triage.

# Repo permission levels

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

Role: Write ▾ Remove

Choose role

- Read**  
Can read and clone this repository. Can also open and comment on issues and pull requests.
- Triage**  
Can read and clone this repository. Can also manage issues and pull requests.
- ✓ Write**  
Can read, clone, and push to this repository.  
Can also manage issues and pull requests.
- Maintain**  
Can read, clone, and push to this repository.  
They can also manage issues, pull requests, and some repository settings.
- Admin**  
Can read, clone, and push to this repository.  
Can also manage issues, pull requests, and repository settings, including adding collaborators.

When you add a collaborator (and they accept the invite) you can choose from **predefined roles** with different level of access

- **Read:**
    - View and clone the repository.
    - Open and comment on issues and pull requests.
    - Download releases.
  - **Triage:**
    - Manage issues and pull requests without write access.
    - Label and assign issues and pull requests.
    - Close and reopen issues and pull requests.
  - **Maintain:**
    - Push to the repository.
    - Manage issues, pull requests, labels, and project boards.
    - Create and publish releases.
    - Configure repository settings for non-sensitive fields (like assigning collaborators).
  - **Admin:**
    - Full control over the repository.
    - Push to, delete, or archive the repository.
    - Change repository settings, including sensitive fields like security or billing.
    - Manage collaborators and their permissions.
- Access to all functionalities provided in the Maintain role.

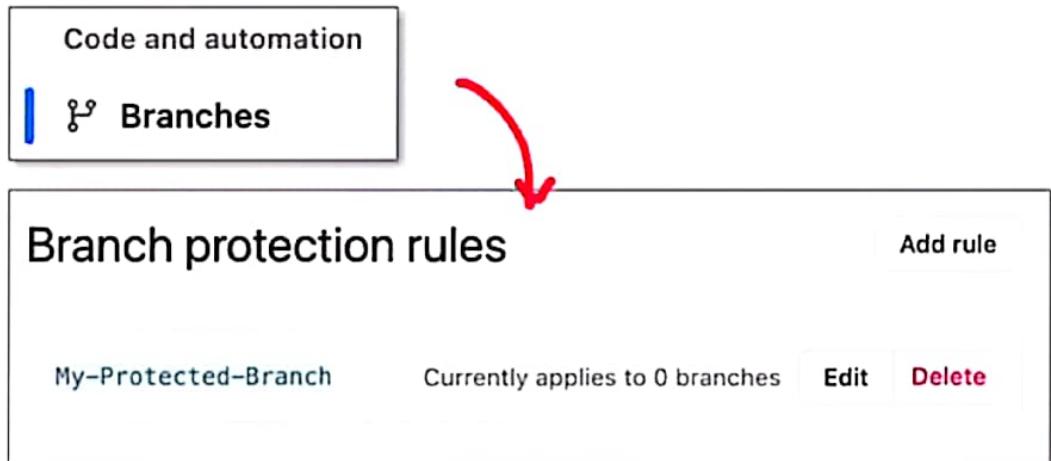


Enterprise accounts can have custom roles.

# Branch Protection Rules

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

Branch protection rules are used to **enforce certain workflows or requirements** before changes can be merged into a branch.



The screenshot shows the GitHub interface for managing branch protection rules. On the left, there's a sidebar with 'Code and automation' and 'Branches' selected. The main area is titled 'Branch protection rules' and shows a single rule named 'My-Protected-Branch'. It indicates 'Currently applies to 0 branches'. There are 'Edit' and 'Delete' buttons. An 'Add rule' button is located at the top right of this section. A red arrow points from the 'Branches' link in the sidebar down to the 'Edit' button in the rule card.

## Protect matching branches rules

- Require a pull request before merging
- Require status checks to pass before merging
- Require conversation resolution before merging
- Require signed commits
- Require linear history
- Require deployments to succeed before merging
- Lock branch
- Do not allow bypassing the above settings

## Rules applied to everyone including administrators

- Allow force pushes
- Allow deletions

# Security Tab features and options

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)



The security tab will present a **repo security checklist** of security options to configure

## Security policy

Create markdown file that explains how security vulnerabilities should be reported

## Security advisories

privately discuss, fix, and publish information about security vulnerabilities in your public repository

## Private vulnerability reporting

Allow your community to privately report potential security vulnerabilities to maintainers and repository



## Dependabot alerts

A bot that alerts you of vulnerabilities due to out-of-date dependencies.

Dependabot can automatically create PRs to update dependencies for you to approve.

## Code scanning alerts

Automatically detect common vulnerabilities and coding errors. Via CodeQL or third-party tools

## Secret scanning alerts

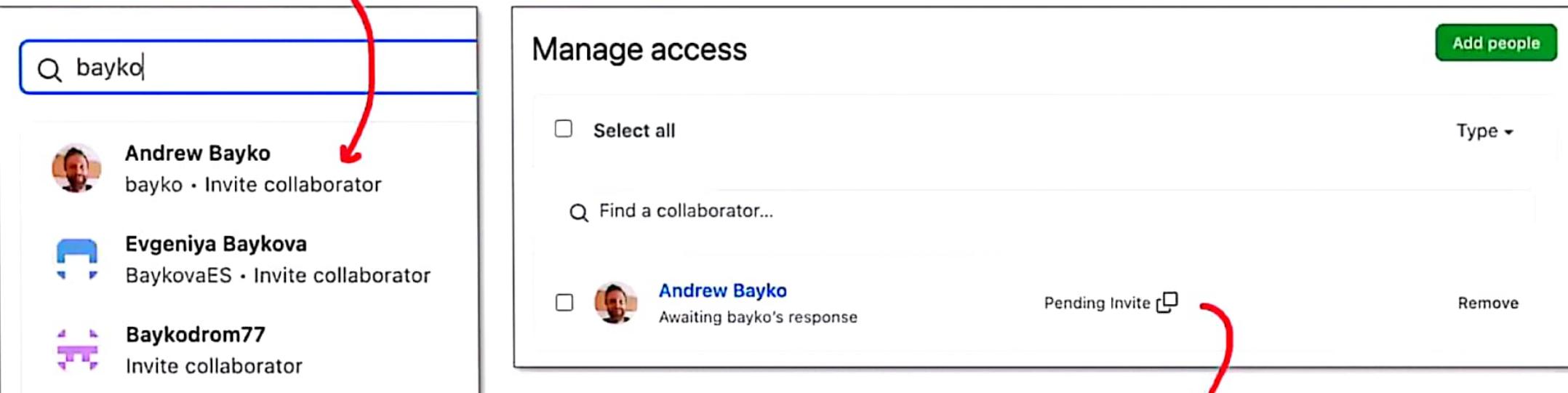
Get notified when a secret is pushed to this repository

# Managing Collaborators

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

Collaborators allows you let other GitHub users have access to your repo based on the permission levels you provide

You search their **username** to invite



The screenshot shows two side-by-side panels from GitHub's 'Manage access' feature.

**Left Panel:** A search bar contains the text 'bayko'. Below it, a list of users includes:

- Andrew Bayko (profile picture, 'bayko · Invite collaborator')
- Evgeniya Baykova (profile picture, 'BaykovaES · Invite collaborator')
- Baykodrom77 (profile picture, 'Invite collaborator')

A red arrow points from the search bar down to the list of results.

**Right Panel:** The title is 'Manage access'. It features a 'Select all' checkbox, a search bar ('Find a collaborator...'), and a list of invited users:

- Andrew Bayko (profile picture, 'Awaiting bayko's response', 'Pending Invite' with a delete icon)

A red arrow points from the 'Pending Invite' link in the list back towards the search bar on the left.

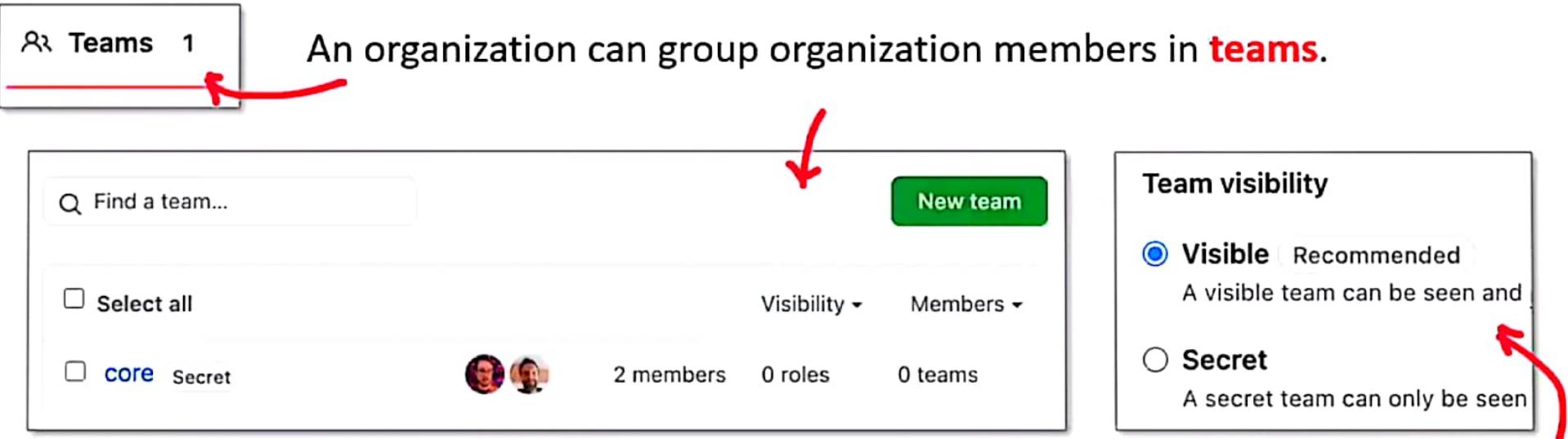
<https://github.com/<username>/<repo-name>/invitations>

The user need to **accept** the invite.

*Its best to share the link directly and manually remind them to accept*

# Organization Teams Tab

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)



An organization can group organization members in **teams**.

Find a team... New team

Select all      Visibility ▾      Members ▾

core Secret      2 members      0 roles      0 teams

**Team visibility**

**Visible** Recommended  
A visible team can be seen and

**Secret**  
A secret team can only be seen

Teams can be **public** or **private**

- Teams can be assigned to projects and issue
- Request reviews from teams
- Teams can be mentioned in discussions, issues, and pull requests
- Control team access to repositories
- New members can be added to teams, instantly giving them access to all relevant repositories and discussions

# GitHub Connect

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)



**GitHub Connect** enhances GitHub Enterprise Server (GHES) by allowing your GHES instance **access some of GitHub.com cloud only offerings**

You can **individually enable these features:**

- Automatic user license sync
- Dependabot
- GitHub.com actions
- Server Statistics
- Unified search
- Unified contributions

# GitHub Internal Repos

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

**GitHub Enterprise Cloud** accounts have a third special GitHub Repo type called **Internal**

- All enterprise members have **read permissions** to the internal repository
- Internal repositories are not visible to people outside of the enterprise, including outside collaborators on organization repositories.
- Unless your enterprise uses Enterprise Managed Users (EMUs), members of the enterprise can fork any internal repository owned by an organization in the enterprise.

Internal repositories are the **default setting** for all new repositories created in an organization owned by an enterprise account.

# GitHub Advanced Security

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

GitHub makes **extra security features available** to customers under a GitHub Advanced Security (GHAS) license.

GitHub Advanced Security is available for enterprise accounts on:

- GitHub Enterprise Cloud
- GitHub Enterprise Server
- And some features ~~of~~ for public repositories on GitHub.com

GitHub Advanced Security license provides the following additional features:

|                   | Public repository | Private repository without Advanced Security | Private repository with Advanced Security |
|-------------------|-------------------|----------------------------------------------|-------------------------------------------|
| Code scanning     |                   |                                              |                                           |
| Secret scanning   |                   |                                              |                                           |
| Dependency review |                   |                                              |                                           |

# SAML and SCIM

Cheat sheets, Practice Exams and Flash cards  [www.exampro.co/ghf](http://www.exampro.co/ghf)

**SAML and SCIM** is available for both GitHub Enterprise Cloud and GitHub Enterprise Server

## SAML

**Security Assertion Markup Language (SAML)** is an open standard for exchanging authentication and authorization between an identity provider and a service provider.  
An important use case for SAML is **Single-Sign-On via web browser**.

## SCIM

**System for Cross-domain Identity Management (SCIM)** is an open standard for automating the exchange of user identity information across different identity management systems. A key use case for SCIM is to enable scalable and automated user provisioning and deprovisioning, often integrated with enterprise identity services.

SAML is about **securely transmitting user authentication and authorization data** for single sign-on purposes.

vs

SCIM is about **managing user identities across different systems** to simplify account maintenance and provisioning



They can complement each other in an ecosystem where you need both federated identity for authentication and a standardized way to manage user accounts across various systems.