

GitHub Implementation Guide



Case Study: Visual Path is training institution and it provides world class trainings on Cloud Computing, DevOps with respective to the Azure Cloud.

As part of this Visual path wants to manage their Azure DevOps Engineering Course Material and its related infrastructure Code automation in centralized place and would like to track the continuous changes to their training material and code for the project.

Solution: GitHub is SaaS (Software-as-a-Service) software and helps to the visual path to address their version control system and track their training material in efficient way.

Working With Git Hub

Objective: This document gives the understanding about what is version control system and GitHub, helps in managing Centralized project source code.

- What is Git?
- What is GitHub
- Create a GitHub Repo
- Organizations
- Repository Roles
- Key Features
- Issues
- Notifications
- Branches
- Commits
- Pull Requests
- Labels
- Actions
- Cloning and Forking
- Organizations
- Repository Roles

Implementation Guide:

What is Git?

A version control system, or VCS, tracks the history of changes as people and teams collaborate on projects together. As developers make changes to the project, any earlier version of the project can be recovered at any time.

Developers can review project history to find out:

- Which changes were made?
- Who made the changes?
- When were the changes made?
- Why were changes needed?

What is a Repository?

A repository, or Git project, encompasses the entire collection of files and folders associated with a project, along with each file's revision history

The file history appears as snapshots in time called commits

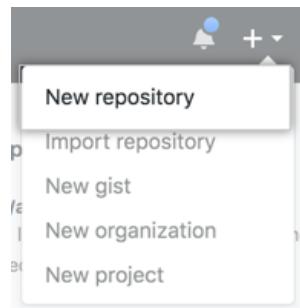
What is GitHub?

GitHub is a cloud development platform that enables us to

- Host and review code
- Manage projects
- Build software
- Handles version control

Creating GitHub Repository: Login to the github.com after the registration with your login credentials

Step-1: In the upper-right corner of any page, use the drop-down menu, and select **New repository**



Step-2:

1. In the **Repository name** box, enter Azure-DevOps.
2. In the **Description** box, write a short description.
3. Select **Add a README file**.
4. Select whether your repository will be **Public** or **Private**.
5. Click **Create repository**.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Owner *



kbattiprolu88 ▾

Repository name *

Azure-DevOps



Great repository names are short and memorable. Need inspiration? How about [miniature-octo-dollop](#)?

Description (optional)

Public

Anyone on the internet can see this repository. You choose who can commit.

Private

You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

Add a README file

This is where you can write a long description for your project. [Learn more](#).

Add .gitignore

Choose which files not to track from a list of templates. [Learn more](#).

.gitignore template: None ▾

Choose a license

A license tells others what they can and can't do with your code. [Learn more](#).

License: None ▾

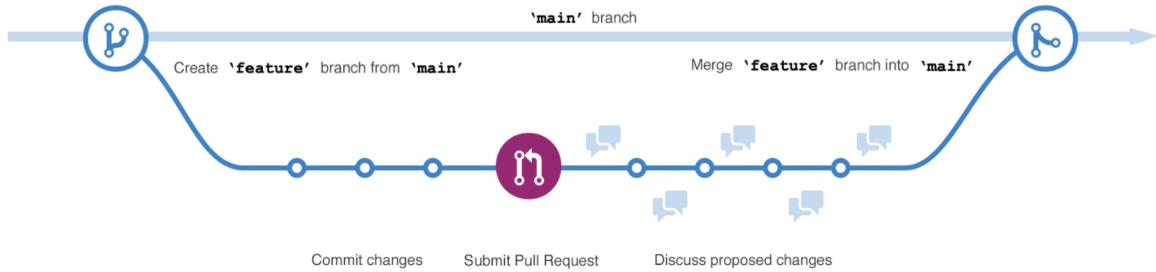
This will set main as the default branch. Change the default name in your [settings](#).

You are creating a public repository in your personal account.

Create repository

What is a Branch ?

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



Create a Branch

1. Click the **Code** tab of your Azure-DevOps repository.
2. Click the drop down at the top of the file list that says **main**.

A screenshot of the Azure-DevOps repository interface. The 'Code' tab is selected. A dropdown menu is open over the 'main' branch selector, showing options like 'Switch branch' and 'Create branch: feature from main'.

1. Type a branch name, **feature** into the text box.
2. Click **Create branch: feature from main**.

A screenshot of the Azure-DevOps repository interface. The 'Code' tab is selected. A dropdown menu is open over the 'main' branch selector, with the 'feature' branch name typed into the search input field.

Now you have two branches, `main` and `feature`. Right now, they look exactly the same. Next you'll add changes to the new branch.

Making and Committing changes ?

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

How to make changes and commit to branch:

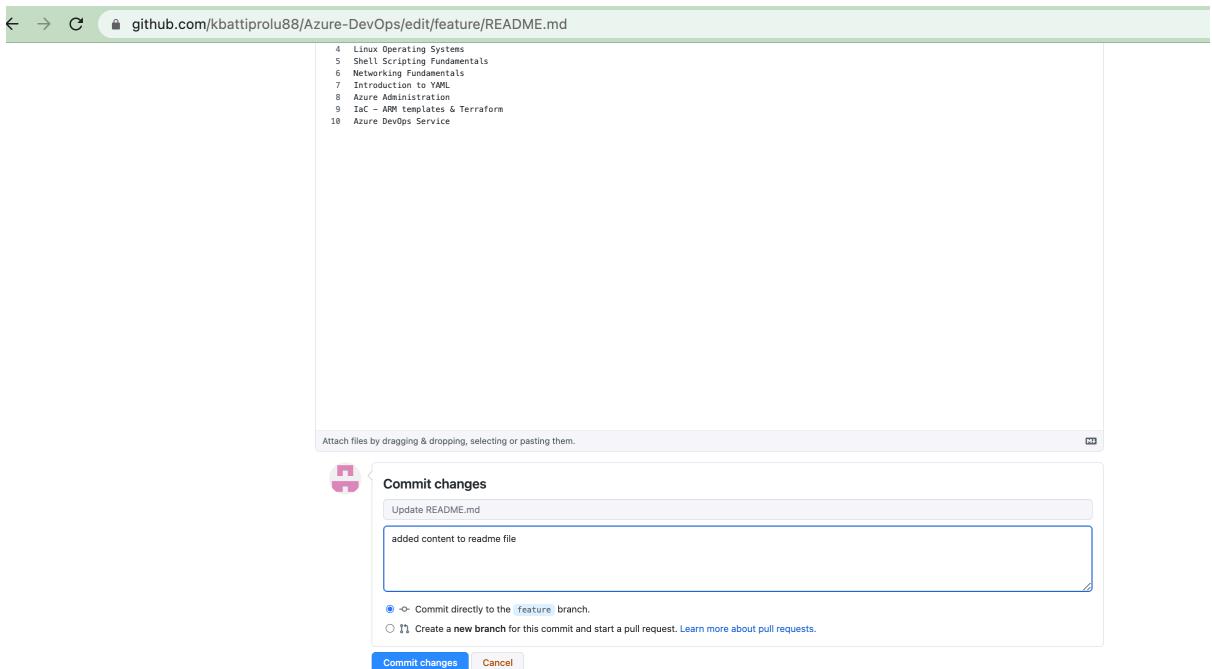
1. Under the `feature` branch you created, click the `README.md` file.
2. Click to edit the file.
3. In the editor, write a bit about yourself. Try using different Markdown elements.
4. In the **Commit changes** box, write a commit message that describes your changes.
5. Click **Commit changes**

The image consists of two vertically stacked screenshots of the GitHub interface, illustrating the steps to make and commit changes to the `README.md` file in the `feature` branch.

Screenshot 1 (Top): This screenshot shows the GitHub repository page for `Azure-DevOps`. The `feature` branch is selected in the dropdown menu. The `README.md` file is listed under the `Code` tab. The commit history shows a single commit from `kbattiprolu88` with the message "Initial commit". The commit details show "1 contributor". The file content is displayed as "Azure-DevOps".

Screenshot 2 (Bottom): This screenshot shows the GitHub repository page for `kbattiprolu88/Azure-DevOps`. The `feature` branch is selected in the dropdown menu. The `Code` tab is active, showing the `README.md` file. The file content is now updated to:

```
1 This is Repository is for Azure DevOps October batch students
2
3 GitHub Documentation
4 Linux Operating Systems
5 Shell Scripting Fundamentals
6 Networking Fundamentals
7 Introduction to YAML
8 Azure Administration
9 IaC - ARM templates & Terraform
10 Azure DevOps Service
```



These changes will be made only to the README file on your feature branch, so now this branch contains content that's different from main.

Pull Request:

- A pull request is the mechanism used to signal that the commits from one branch are ready to be merged into another branch.
- one or more reviewers to verify the code and approve the merge and once merge
- Pull requests are the heart of collaboration on GitHub. When you open a pull request, you're proposing your changes and requesting that someone review and pull in your contribution and merge them into their branch. Pull requests show diffs, or differences, of the content from both branches. The changes, additions, and subtractions are shown in different colors.
- As soon as you make a commit, you can open a pull request and start a discussion, even before the code is finished.
- Click the **Pull requests** tab of your feature repository.
- Click **New pull request**
- In the **Example Comparisons** box, select the branch you made, **feature**, to compare with **main** (the original).
- Look over your changes in the diffs on the Compare page, make sure they're what you want to submit.

github.com/kbattiprolu88/Azure-DevOps/tree/feature

Search or jump to... Pull requests Issues Marketplace Explore

kbattiprolu88 / Azure-DevOps Private

Code Issues Pull requests Actions Projects Security Insights Settings

feature had recent pushes 4 minutes ago Compare & pull request

feature 2 branches 0 tags Go to file Add file Code

This branch is 2 commits ahead of main. Contribute

kbattiprolu88 Update README.md beed921 4 minutes ago 3 commits

README.md Update README.md 4 minutes ago

README.md

#This is Repository is for Azure DevOps October batch students

GitHub Documentation Linux Operating Systems Shell Scripting Fundamentals Networking Fundamentals Introduction to YAML Azure Administration IaC - ARM templates & Terraform Azure DevOps Service

kbattiprolu88 / Azure-DevOps Private

Code Issues Pull requests Actions Projects Security Insights Settings

feature had recent pushes 4 minutes ago Compare & pull request

Filters Q is:pr is:open Labels 9 Milestones 0 New pull request

Welcome to pull requests!

Pull requests help you collaborate on code with other people. As pull requests are created, they'll appear here in a searchable and filterable list. To get started, you should [create a pull request](#).

ProTip! Adding no:label will show everything without a label.

github.com/kbattiprolu88/Azure-DevOps/compare/main...feature

Code Issues Pull requests Actions Projects Security Insights Settings

Comparing 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](#).

base: main compare: feature Able to merge. These branches can be automatically merged.

Discuss and review the changes in this comparison with others. [Learn about pull requests](#) Create pull request

2 commits 1 file changed 1 contributor

Commits on Oct 16, 2022

Update README.md (1 commit) Verified 3a7edb

kbattiprolu88 committed 6 minutes ago

Update README.md (1 commit) Verified beed921

kbattiprolu88 committed 5 minutes ago

Showing 1 changed file with 10 additions and 1 deletion.

11 README.md

@@ -1 +1,10 @@

1 - # Azure-DevOps

1 + #This is Repository is for Azure DevOps October batch students

2 +

3 + GitHub Documentation

4 + Linux Operating Systems

5 + Shell Scripting Fundamentals

6 + Networking Fundamentals

7 + Introduction to YAML

8 + Azure Administration

9 + IaC - ARM templates & Terraform

10 + Azure DevOps Service

The screenshot shows two consecutive screenshots of the GitHub pull request interface.

Screenshot 1: Open a pull request

- Header:** github.com/kbattiprolu88/Azure-DevOps/compare/main...feature
- Form:**
 - Base: main ▾
 - Compare: feature ▾
 - Able to merge**: These branches can be automatically merged.
 - Title:** Feature
 - Description:** adding changes to the main branch
 - Text Area:** Attach files by dragging & dropping, selecting or pasting them.
 - Buttons:** Create pull request
- Commit History:**
 - > 2 commits
 - Commits on Oct 16, 2022
 - Update README.md ... (kbattiprolu88 committed 6 minutes ago)
 - Update README.md (kbattiprolu88 committed 5 minutes ago)
- Labels:**
 - bug
 - documentation
 - duplicate
 - enhancement
 - good first issue
 - help wanted
 - invalid
 - question
 - Verified
 - beed921

Screenshot 2: Pull request details

- Header:** github.com/kbattiprolu88/Azure-DevOps/pull/1
- Repository:** kbattiprolu88 / Azure-DevOps (Private)
- Navigation:** Code, Issues, Pull requests 1, Actions, Projects, Security, Insights, Settings
- Title:** Feature #1
- Status:** Open
- Comments:**
 - kbattiprolu88 commented now: adding changes to the main branch
 - kbattiprolu88 added 2 commits 6 minutes ago:
 - Update README.md ... (Verified, 3a77edb)
 - Update README.md (Verified, beed921)
 - kbattiprolu88 added the documentation label now
- Reviewers:** No reviews
- Assignees:** No one—assign yourself
- Labels:** documentation
- Projects:** None yet
- Milestone:** No milestone
- Development:** Successfully merging this pull request may close these issues. None yet
- Notifications:** Customize, Unsubscribe

Merging Pull Request

In this final step, you will merge your `readme-edits` branch into the `main` branch. After you merge your pull request, the changes on your feature branch will be incorporated into `main`.

Sometimes, a pull request may introduce changes to code that conflict with the existing code on `main`.

If there are any conflicts, GitHub will alert you about the conflicting code and prevent merging until the conflicts are resolved. You can make a commit that resolves the

conflicts or use comments in the pull request to discuss the conflicts with your team members.

In this walk-through, you should not have any conflicts, so you are ready to merge your branch into the main branch.

Add more commits by pushing to the **feature** branch on kbattiprolu88/Azure-DevOps.

The screenshot shows the GitHub pull request merge interface. At the top, there's a message: "Add more commits by pushing to the **feature** branch on kbattiprolu88/Azure-DevOps.". Below it, a blue icon with a gear and wrench is followed by the text: "Continuous integration has not been set up". A note below says: "GitHub Actions and several other apps can be used to automatically catch bugs and enforce style." A large blue checkmark icon is next to the text: "This branch has no conflicts with the base branch". Below that, a note says: "Merging can be performed automatically." At the bottom, a blue button says "Merge pull request" with a dropdown arrow, and a note says: "You can also open this in GitHub Desktop or view command line instructions."

This screenshot shows the GitHub pull request merge interface again. It displays the same initial message and CI status. Below the merge button, there's a note: "This commit will be authored by 113235653+kbattiprolu88@users.noreply.github.com". At the bottom, there are two buttons: "Confirm merge" (blue) and "Cancel".

Changes are merged with the main branch

The screenshot shows the GitHub repository main page for kbattiprolu88 / Azure-DevOps. The repository is private. The navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Security, Insights, and Settings. The "Code" tab is selected. Below the navigation, it shows "main" branch, "2 branches", and "0 tags". There are buttons for "Go to file", "Add file", and "Code". A list of recent activity shows a merge pull request from kbattiprolu88/feature to main, made 36 seconds ago by kbattiprolu88, with 4 commits. A file named README.md was updated 9 minutes ago. The README.md content is displayed in a box, starting with "#This is Repository is for Azure DevOps October batch students" and listing various topics like GitHub Documentation, Linux Operating Systems, Shell Scripting Fundamentals, Networking Fundamentals, Introduction to YAML, Azure Administration IaC - ARM templates & Terraform, and Azure DevOps Service.

Key Features of GitHub:

Below are the key features of GitHub



Issues



Discussions



Pull requests



Notifications



Labels



Actions



Forks



Projects

Issues in GitHub?

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

- Bug Reports
- Feature Requests
- Documentation clarification

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

Notifications:

- As a collaborative platform, GitHub offers notifications for virtually every event that takes place within a given workflow. These notifications can be finely tuned to meet your preferences
- You can also decide whether you receive notifications via email, web & mobile, or both
- To keep track of all of your notifications across different projects, use the [GitHub Notifications dashboard](#)

Labels:

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

- Examples of Labels include:
- bug
- documentation
- duplicate
- help wanted
- enhancement
- question

Actions:

- GitHub actions provide task automation and workflow functionality in a repository
- Actions can be used to streamline processes in your software development lifecycle and implement continuous integration and continuous deployment (CI/CD).
- **Workflows:** Automated processes added to your repository.
- **Events:** An activity that triggers a workflow.
- **Jobs:** A set of steps that execute on a runner.
- **Steps:** A task that can run one or more commands (actions).
- **Actions:** Standalone commands that can be combined into steps. Multiple steps can be combined to create a job.
- **Runners:** Server that has the GitHub Actions runner application installed.

Forking & Cloning

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

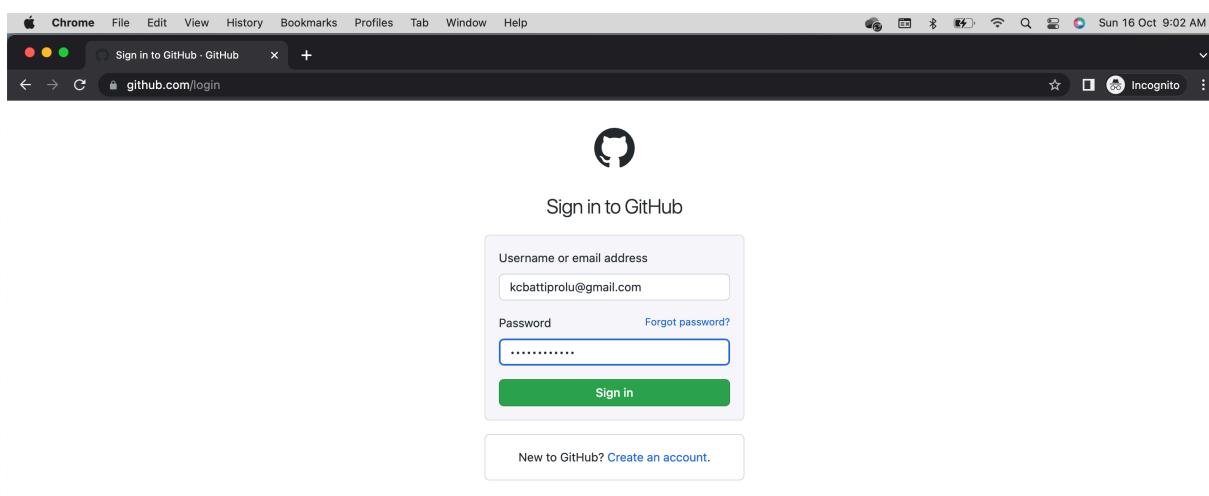
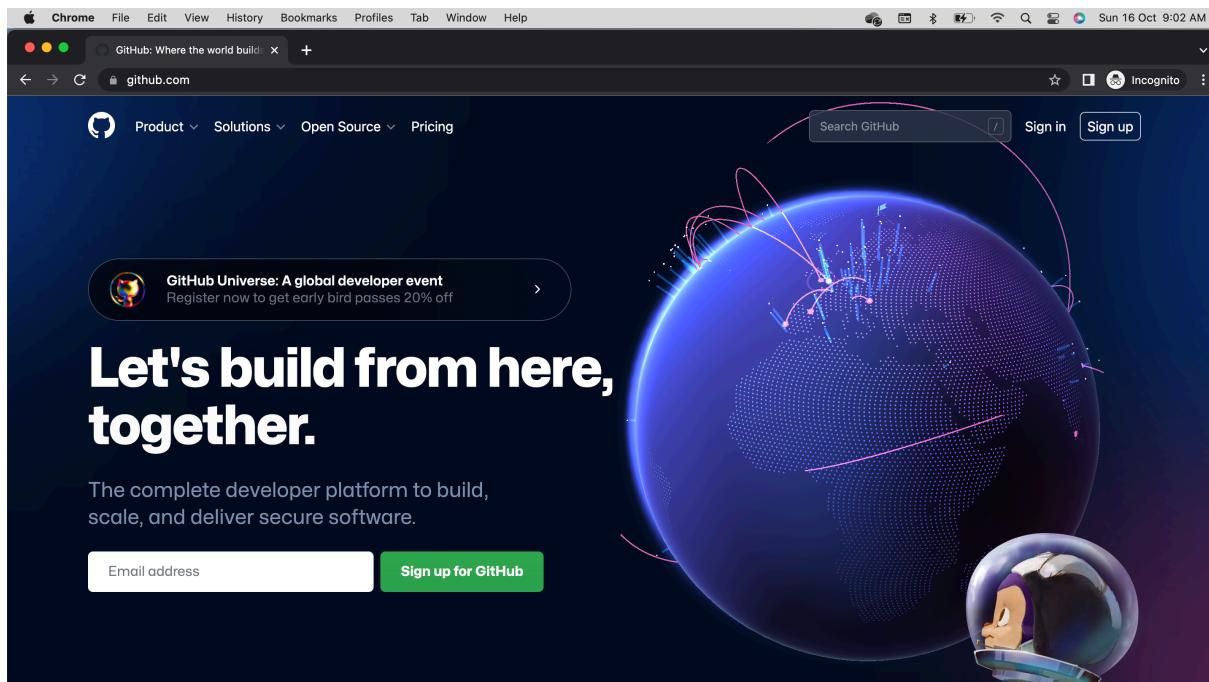
GitHub Organizations:

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

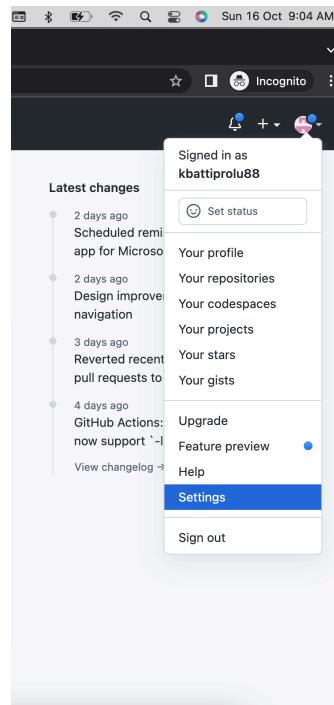
How to Create GitHub Organization

Login to your github account

<https://github.com/>



Once You login to Git Hub Account please navigate to account settings page as shown below



Once you access your accounts settings page you can see the organization option as below

A screenshot of a Chrome browser window displaying the GitHub account settings page at <https://github.com/settings/organizations>. The 'Organizations' tab is selected, indicated by a blue bar and a blue outline around the tab name. Other tabs visible include 'Public profile', 'Account', 'Appearance', 'Accessibility', 'Notifications', 'Billing and plans', 'Emails', 'Password and authentication', 'SSH and GPG keys', 'Moderation', 'Repositories', and 'Packages'. The URL in the address bar is <https://github.com/settings/organizations>.

Create New organization

A screenshot of a Chrome browser window showing the GitHub Organizations settings page. The URL is github.com/settings/organizations. The page displays the user's personal account information, including a profile picture and the handle "kbattiprolu88". Navigation links for "Public profile" and "Organizations" are visible, along with a "New organization" button.

A screenshot of the same GitHub Organizations settings page, focusing on the "Choose Free Subscription" section. It shows the user has selected the "Free" plan, which is \$0 per year forever. Other plans like "Team" (\$48 per user/year) and "Enterprise" (\$252 per user/year) are also listed with their respective features.

Choose Free Subscription

A screenshot of a Chrome browser window showing the GitHub Organizations plan selection page. The URL is github.com/organizations/plan. The heading is "Pick a plan for your organization". It asks "How often do you want to pay?" with options for "Monthly" and "Yearly" (which includes a "Get 1 month free" offer). Three plan options are shown: "Free", "Team", and "Enterprise".

Pick a plan for your organization

How often do you want to pay?

Monthly Yearly

Free

The basics for individuals and organizations

\$ 0 per year forever

[Create a free organization](#)

- > Unlimited public/private repositories
- > Automatic security and version updates
- > 2,000 CI/CD minutes/month
Free for public repositories
- > 500MB of Packages storage
Free for public repositories
- > New issues & projects (in limited beta)

Team

Advanced collaboration for individuals and organizations

\$ 48 \$ 44 per user/year
for the first 12 months*

[Continue with Team](#)

- < Everything included in Free, plus...
- > Access to GitHub Codespaces
- > Protected branches
- > Multiple reviewers in pull requests
- > Draft pull requests
- > Code owners

Enterprise

Security, compliance, and flexible deployment

\$ 252 \$ 231 per user/year
for the first 12 months*

[Start a free trial](#)

[Contact Sales](#)

- < Everything included in Team, plus...

- > Enterprise Managed Users
- > User provisioning through SCIM
- > Enterprise Account to centrally manage multiple organizations
- > Environment protection rules

A screenshot of a Chrome browser window showing the GitHub "Set up your organization" creation page. The URL is github.com/account/organizations/new?plan=free&ref_cta=Create%2520a%2520free%2520organization&ref_loc=cards&ref_page=%2Forganizations%.... The page asks for the organization account name ("visualpath-iac-cloud") and contact email ("kbattiprolu@gmail.com"). It also asks if the organization belongs to "My personal account" or "A business or institution". A large "Verify your account" section contains a green checkmark icon. At the bottom, there is a checkbox for accepting the Terms of Service and a "Next" button.

Organization account name *

This will be the name of your account on GitHub.
Your URL will be: <https://github.com/visualpath-iac-cloud>.

Contact email *

This organization belongs to: *

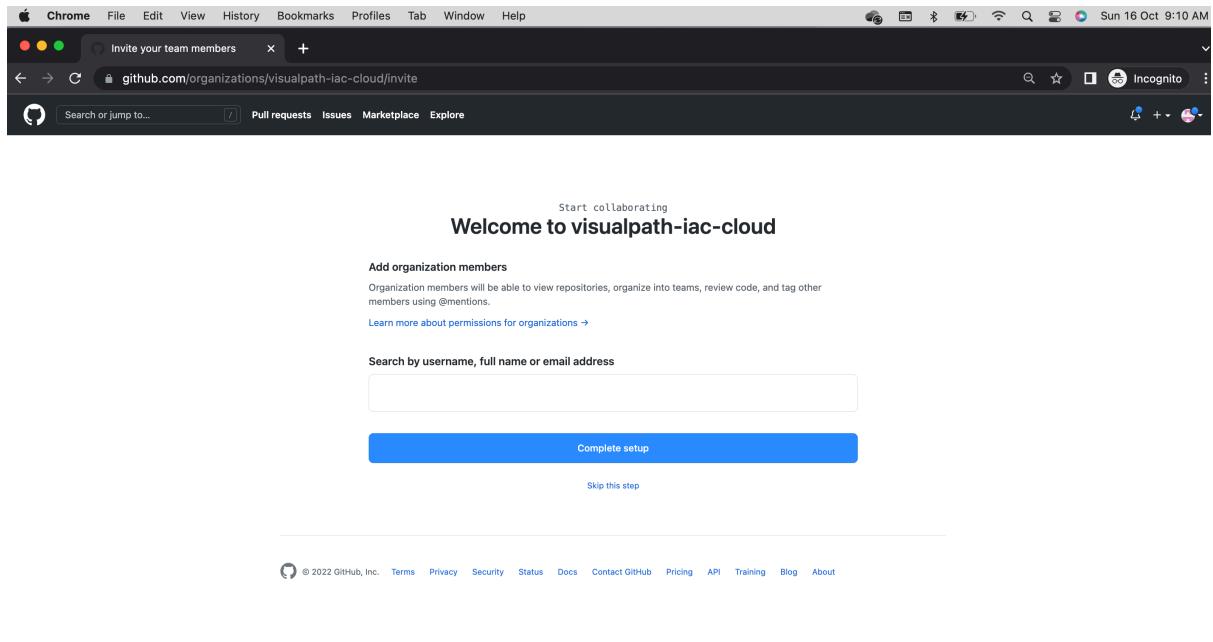
My personal account
I.e., kbattiprolu88

A business or institution
For example: GitHub, Inc., Example Institute, American Red Cross

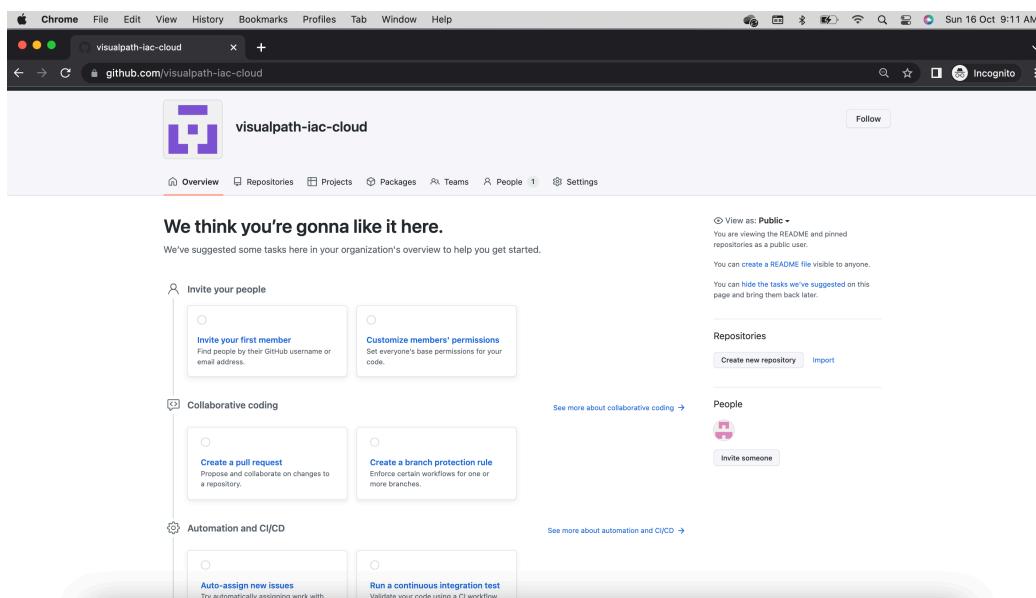
Verify your account

I hereby accept the [Terms of Service](#). For more information about GitHub's privacy practices, see the [GitHub Privacy Statement](#).

[Next](#)



The screenshot shows the GitHub organization setup process for 'visualpath-iac-cloud'. At the top, there's a header with 'Start collaborating' and the organization name. Below it, a section titled 'Add organization members' explains that members can view repositories, organize into teams, review code, and tag others. A link to 'Learn more about permissions for organizations' is provided. A search bar for 'Search by username, full name or email address' is followed by a large blue button labeled 'Complete setup'. Below this, a small link says 'Skip this step'. At the bottom, a footer includes links to GitHub's Terms, Privacy, Security, Status, Docs, Contact GitHub, Pricing, API, Training, Blog, and About.



The screenshot shows the GitHub organization overview for 'visualpath-iac-cloud'. It features a header with the organization logo and a 'Follow' button. Below the header, a section titled 'We think you're gonna like it here.' lists several tasks: 'Invite your people' (with 'Invite your first member' and 'Customize members' permissions'), 'Collaborative coding' (with 'Create a pull request' and 'Create a branch protection rule'), and 'Automation and CI/CD' (with 'Auto-assign new issues' and 'Run a continuous integration test'). To the right, sections for 'View as: Public' (with a note about pinned repositories), 'Repositories' (with 'Create new repository' and 'Import' buttons), and 'People' (with an 'Invite someone' button) are shown.

GitHub Repository Roles:

- **Read:** Recommended for non-code contributors who want to view or discuss your project
- **Triage:** Recommended for contributors who need to proactively manage issues and pull requests without write access
- **Write:** Recommended for contributors who actively push to your project
- **Maintain:** Recommended for project managers who need to manage the repository without access to sensitive or destructive actions
- **Admin:** Recommended for people who need full access to the project, including sensitive and destructive actions like managing security or deleting a repository