GitHub For Admins

Enabling 2FA Authentication

- Install an 2FA authenticator app like **Google's Authenticator** on your mobile device
- Sign in to your account on https://github.com/
- From the top right icon of your account, click on the dropdown and select **Settings**
- From under **Access** section on the left, select **Password and Authentication**
- In the "Two-factor authentication" section of the page, click **Enable two-factor authentication**.
- Follow the instructions on the screen to setup the authentication

Detailed document can be found here

Member Privileges

1. Sign in to your account on https://github.com/
2. From the top right icon of your account, click on the dropdown and select **Your organizations**
3. From under **Access** section on the left, select **Member Privileges**
    a. Under **Member team permissions**
        i. Uncheck & Save - **Allow members to create teams**
4. From under **Access** section on the left, select **Moderation**
    a. Add moderators who can block and unblock users from the organization, minimize comments, and manage interaction limits for all public organization repositories.
5. From under **Repository** section on the left, select **Repository Defaults**

   a. Set **Commit Signoff** as **All Repositories**
   What's Commit SignOff?- will require contributors to sign off on commits they make through GitHub's web interface. Signing off is a way for contributors to affirm that their commit complies with the repository's terms

Creating a Team

1. Open https://github.com/orgs/OdiaGenAI/teams
2. Click **New Team**
3. Enter a name for the team, and a description
4. Set the visibility desired
5. Click **Create Team**
6. You can now see an option to add team members
7. Ensure the team can be a child of another larger teams

Per Repository Settings

1. Click open a repository and click **Settings** (this option is visible only to admins)
2. **General Settings**
   a. Under **Pull Requests**
      i. Uncheck option **Allow merge commits**
      ii. Uncheck option **Allow rebase merging**
      iii. Only select **Allow Squash Merging**, uncheck everything else
      iv. Under **Allow Squash Merging**, select **Default message**
      v. Check option **Always suggest updating pull request branches**
      vi. Check option **Automatically delete head branches**
3. Under **Access**

a. select **Collaborator & Teams**

i. add the appropriate team for the repo

4. Under **Code and Automation**

a. select **Branches**

i. Click on **Add branch protection rule**

ii. Under **Branch Name Pattern,** use `main`

iii. Check option **Require a pull request before merging**

iv. Check option **Dismiss stale pull request approvals when new commits are pushed**

v. Check option **Require conversation resolution before merging**

vi. Check option **Require linear history**

vii. Check option **Lock branch**

viii. Check option **Do not allow bypassing the above settings**

ix. Click **Create**

b. select **Tags**

a. Under Tag Name Pattern, use `release-*`