

# GitHub- Training



Mithun Technologies  
devopstrainingblr@gmail.com  
+91-9980923226





## - Agenda

- ✓ Introduction
- ✓ GitHub Administration (Org/Repo/Users/Team Creation)
- ✓ Personal Access Token Generation
- ✓ SSH Key Generation
- ✓ GIT Commands
- ✓ Pull Request Creation and Merging
- ✓ Create Tag / Releases





## - Introduction

### **Pre Requisite Software Download/ Registration :**

- ❖ Register @ <https://github.com/>
- ❖ Install git bash @ <https://git-scm.com/downloads>





## - Introduction

### What is **git** and why do we use it ?

- Git is an open-source version control system that was started by Linus Trovalds – the same person who created Linux.
- Git is similar to other version control systems – like SVN (Subversion), CVS (Concurrent Versions System), and TFS (Team Foundation Server), it is Distributed model.
- In the distributed approach, each developer works directly with his or her own local repository, and changes are shared between repositories as a separate step.

### Popular Open sources

Git : written in a collection of Perl, C, and various shell scripts.

Mercurial : Written in Python.





## - Administration

- Org Creation
- Repository Creation
- Team Creation
- Adding Users to Team
- Provide the access to Team to Repository
- Importing Repository





## - Administration - Org

### Steps to create Organization

As soon as you login into GitHub, in the upper-right corner, click on + symbol, then click on **New Organization**,

Under "**Organization name**", give the organization a name,

Under "**Billing email**", type the email where receipts for your organization's paid plan should be sent, then click on **Create organization** button.





## - Administration

### Sign up your team

Completed Create personal account Step 2: Create organization Step 3: Invite members

Create an organization account

Organization name: devops-training-blr  
The organization account will live at <https://github.com/devops-training-blr>

Billing email: devopstrainingblr@gmail.com  
Receipts will be sent here

Choose your plan

<input checked="" type="radio"/> <b>Free</b> Unlimited users and public repositories	\$0
<input type="radio"/> <b>Team</b> Starts at \$25 / month which includes your first 5 users. Unlimited public repositories Unlimited private repositories	\$9 per user / month
<input type="radio"/> <b>Business</b> Includes everything in the Team plan, plus: SAML based single sign-on (SSO) 99.95% Guaranteed uptime SLA 24/5 email support with < 8 hour response time <a href="#">Need help getting started or on-premises hosting? Contact us.</a>	\$21 per user / month

This account is owned by a business (see the [Corporate Terms of Service](#) for details)

By clicking on "Create organization" below, you are agreeing to the [Terms of Service](#) and the [Privacy Policy](#).

[Create organization](#)





# - Administration

## Invite organization members

Completed Create personal account Step 2: Create organization Step 3: Invite members

Search by username, full name or email address

Finish

**Organization members**

- ✓ See all repositories ⓘ
- ✓ Create repositories
- ✓ Organize into teams
- ✓ Review code
- ✓ Communicate via @mentions

As an organization owner, you'll have complete access to all of the organization's repositories and have control of what members have access using fine-grained permissions.

You'll also be able to change billing info and cancel organization plans.

Learn more

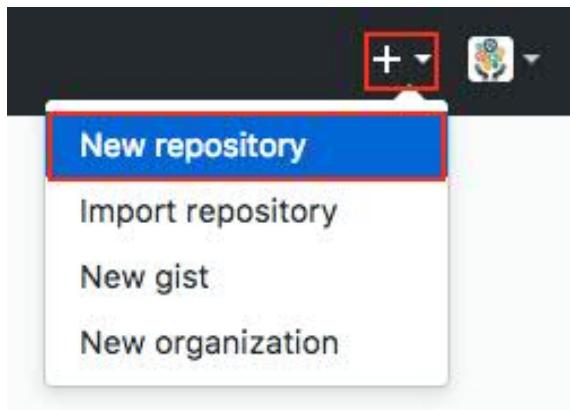
Members will receive their invitation via email. They can also visit <https://github.com/devops-training-blr> to accept the invitation right away.





## - Administration - Repo

As soon as you login into GitHub, in the upper-right corner, click on + symbol, then click on **New repository**.





## - Administration - Repo

Provide the details as below and click on Create repository.

### Create a new repository

A repository contains all the files for your project, including the revision history.

Owner                    Repository name

devops-training-blr / devops-practice ✓

Great repository names are short and memorable. Need inspiration? How about [literate-barnacle](#).

Description (optional)

This repository is used to save all devops related tools documents.

Public

Anyone can see this repository. You choose who can commit.

Private

You choose who can see and commit to this repository.

Initialize this repository with a README

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: None ▾

Add a license: None ▾



**Create repository**



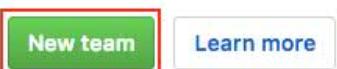


## - Administration - Team

Select the organization where you need to create the team and click on Team.

A screenshot of a GitHub organization's dashboard. At the top, there is a navigation bar with links for 'This organization', 'Search', 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. Below the navigation bar, the organization name 'devops-training-blr' is displayed. Underneath the organization name, there are several tabs: 'Repositories' (2), 'People' (1), 'Teams' (0), 'Projects' (0), and 'Settings'. The 'Teams' tab is highlighted with a red border. The main content area shows a message: 'No teams found. Create your first team now!' with a 'New team' button.

Click on New team.





## - Administration - Team

devops-training-blr

Repositories 2 People 1 Teams 0 Projects 0 Settings

### Create new team

Team name  
 Mention this team in conversations as @devops-training-blr/devops-engineers-team.

Description

Parent team

Team visibility

**Visible** Recommended  
A visible team can be seen and @mentioned by every member of this organization.

**Secret**  
A secret team can only be seen by its members.





## - Administration – Add Users to Team

Go to the organization and Click on Teams and select the team name to which you want to add users.

<https://github.com/devops-training-blr>

A screenshot of the GitHub organization dashboard for 'devops-training-blr'. The top navigation bar shows 'This organization', 'Search', 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. Below the bar, there are links for 'Repositories 2', 'People 1', 'Teams 1' (which is highlighted with a red box), 'Projects 0', and 'Settings'. The 'Teams' section shows one team named 'DevOps-Engineers-Team'.

A screenshot of the 'DevOps-Engineers-Team' details page. It shows a checkbox labeled 'Select all', a checkbox for 'DevOps-Engineers-Team' which is checked and highlighted with a red box, and a note: 'This team mainly providing the solutions for DevOps'. To the right, there are buttons for 'Visibility' and 'Members', both with dropdown menus. Below these are statistics: a user icon, '1 member', and '0 teams'.





# - Administration – Add Users to Team

Teams / DevOps-Engineers-Team @devops-training-blr/devops-engineers-team

This team mainly providing the solutions for DevOps — Edit

**Members 1** Teams 0 Repositories 0 Settings

Find a member... Add a member

1 member 0 child team members Role ▾

DevOps Training devopstrainingblr Maintainer

Invite members to devops-training-blr

mithuntechnologiesblr mithuntechnologiesblr Mithun Technologies

A screenshot of the GitHub Team administration interface for the 'DevOps-Engineers-Team'. It shows one member, 'DevOps Training' (mithuntechnologiesblr), listed as a Maintainer. A modal window titled 'Invite members to devops-training-blr' is open, showing the user 'mithuntechnologiesblr' selected in a search bar. The GitHub logo is visible in the top left corner.





# - Administration – Give the admin access to user

Teams / DevOps-Engineers-Team @devops-training-blr/devops-engineers-team

This team mainly providing the solutions for DevOps — Edit

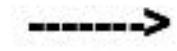
**Members 2** Teams 0 Repositories 0 Settings

Find a member...

1 member selected... ▾

**Change role...** Remove from team

Mithun Technologies mithuntechnologiesblr Maintainer



Change the team role of mithuntechnologiesblr? ×

This action has no effect on Organization owners.

Select a new role:

**Maintainer**  
Can add and remove team members and create child teams.

**Member**  
Has no administrative permissions on the team.

**Change role**





## - Administration – Give the repo access to Team

A screenshot of the GitHub Teams interface for the 'DevOps-Engineers-Team'. The top navigation bar shows 'Teams / DevOps-Engineers-Team @devops-training-blr/devops-engineers-team' and a 'Leave' button. Below the bar, there are tabs for 'Members 2', 'Teams 0', 'Repositories 0' (which is highlighted with a red border), and 'Settings'. A search bar says 'Find a repository...' and a green 'Add repository' button is visible. A checkbox labeled 'Select all' is present. The main content area displays a message: 'This team doesn't have any repositories.' with a small computer monitor icon above the text.

Teams /  
DevOps-Engineers-Team @devops-training-blr/devops-engineers-team  
Leave

This team mainly providing the solutions for DevOps — Edit

Members 2 Teams 0 Repositories 0 Settings

Find a repository... Add repository

Select all

This team doesn't have any repositories.





## - Administration – Give the repo access to Team

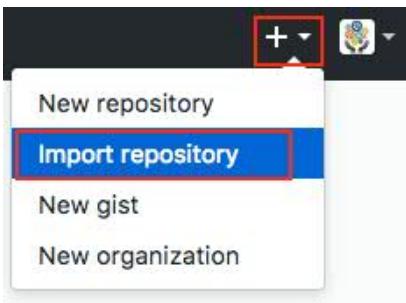
The screenshot shows the GitHub interface for adding a repository to a team. At the top, it says "Add repository to DevOps-Engineers-Team". Below that, a list of repositories is shown, with "Ant-WebProject" selected. A confirmation message "1 repository selected..." is displayed. The main repository card for "devops-training-blr / Ant-WebProject" is shown, with the "Read" permission level selected. A permission level dialog box is open, showing three options: Admin, Write, and Read. The "Read" option is checked. The dialog box has a red border around the "Read" button.





## - Administration - Repo

As soon as you login into GitHub, in the upper-right corner, click on + symbol, then click on **Import repository**.





# - Administration - Repo

## Import your project to GitHub

Import all the files, including the revision history, from another version control system.

### Your old repository's clone URL

`https://github.com/devops-bangalore/Ant-WebProject.git`

Learn more about the types of [supported VCS](#).

### Your new repository details

#### Owner

devops-training-blr ▾

#### Name

/ Ant-WebProject

### Privacy

ⓘ Your new repository will be **public**. In order to make this repository private, you'll need to [upgrade your account](#).

Cancel

**Begin import**





## - Introduction

### **What is Version Control System?**

When developers are creating something (an application, for example), they are making constant changes to the code and releasing new versions, up to and after the first official (non-beta) release.

Version control systems keep these revisions straight, and store the modifications in a central repository. This allows developers to easily collaborate, as they can download a new version of the software, make changes, and upload the newest revision. Every developer can see these new changes, download them, and contribute.





## - Introduction

### Branch

- Branches are used to create another line of development.
- By default, Git has a master branch, which is same as trunk in Subversion (SVN). Usually, a branch is created to work on a new feature.
- Once the feature is completed, it is merged back with the master branch and we delete the branch.

### Tags

- Tags similar to branches, but the difference is that tags are immutable.
- It means, tag is a branch, which nobody intends to modify. Once a tag is created for a particular commit, even if you create a new commit, it will not be updated.
- Usually, developers create tags for product releases.





## - Personal Access Token - Generation

Go to **Personal settings ---> Developer settings**

**→ Personal access tokens ---> Generate New Token** → Provide some info about token in **Token description** input box → Select the appropriate the Scopes and Click on **Generate token** button.





## - SSH Key Generation

SSH keys are a way to identify trusted computers without involving passwords. You can generate an SSH key and add the public key to your GitHub account.

#ls -al ~/.ssh ---> To see if existing SSH keys are present in machine.

### Generate SSH Key

```
# ssh-keygen -t rsa -b 4096 -C "JENKINS_USERNAME@HOSTNAME"
```

Here-t --> Specifies the type of key to create. The possible values are "rsa1" for protocol version 1 and "rsa" or "dsa" for protocol version 2.

-b --> Specifies the number of bits in the key to create. For RSA (Rivest, Shamir, and Adelman) keys, the minimum size is 768 bits and the default is 2048 bits. Generally, 2048 bits is considered sufficient. DSA (Digital Signature Algorithm)keys must be exactly 1024 bits as specified by FIPS 186-2.

-C --> Provides a new comment.

```
cat ~/.ssh/id_rsa.pub
```

### Add SSH key to GitHub

Click on Setting ---> **SSH and GPG keys** ---> **New SSH key** or **Add SSH key** ---> Provide the name for **Title** and copy SSH key **Key** field ---> Click on Add SSH key button





## - SSH Key Generation

```
Bhaskars-MacBook:~ BhaskarReddy$ ssh-keygen -t rsa -b 4096 -C "devopstrainingblr@gmail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/BhaskarReddy/.ssh/id_rsa): [REDACTED]
Enter passphrase (empty for no passphrase): [REDACTED]
Enter same passphrase again: [REDACTED]
Your identification has been saved in /Users/BhaskarReddy/.ssh/id_rsa.
Your public key has been saved in /Users/BhaskarReddy/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:P49RMpILQthwWW/paCLQcY9peaHMY3yiPl120G7Stv4 devopstrainingblr@gmail.com
The key's randomart image is:
+---[RSA 4096]---+
|   ooo          |
| . @.* o .      |
| . o / + +      |
| . = = = .      |
| o o O S o .    |
| . o 0 + + +    |
| o o = . +      |
| . + . =        |
| .o.E . .       |
+---[SHA256]---+
Bhaskars-MacBook:~ BhaskarReddy$
```





## - Pull Request Creation and Merging

<https://help.github.com/articles/creating-a-pull-request/>

**Pull requests** let you tell others about changes you've pushed to a GitHub repository. Once a **pull request** is sent, interested parties can review the set of changes, discuss potential modifications, and even push follow-up commits if necessary.





## - Create Tag / Releases

Step 1: On GitHub, navigate to the main page of the repository.

Step 2: Under your repository name, click **Releases**.

Step 3: Click **Create a new release/Draft a new release**.

Step 4: Type a version number for your release. Select a branch that contains the project you want to release. Usually, you'll want to release against your master branch, unless you're releasing beta software.

Type a title and description that describes your release.

Step 5: If you're ready to publicize your release, click **Publish release**. Otherwise, click **Save draft** to work on it later.





## - GIT Commands

**#git init** : Initialize git repo.

**#git status** : Check current status of the project

**#git add "Mithun.txt"** : Add file/files to staging area also called indexing.

**#git commit -m "Initial commit"** : Commit your indexed files to local git repo.

**#git remote add origin <GIT Remote URL>** : Add Remote repository where you will be pushing your code.

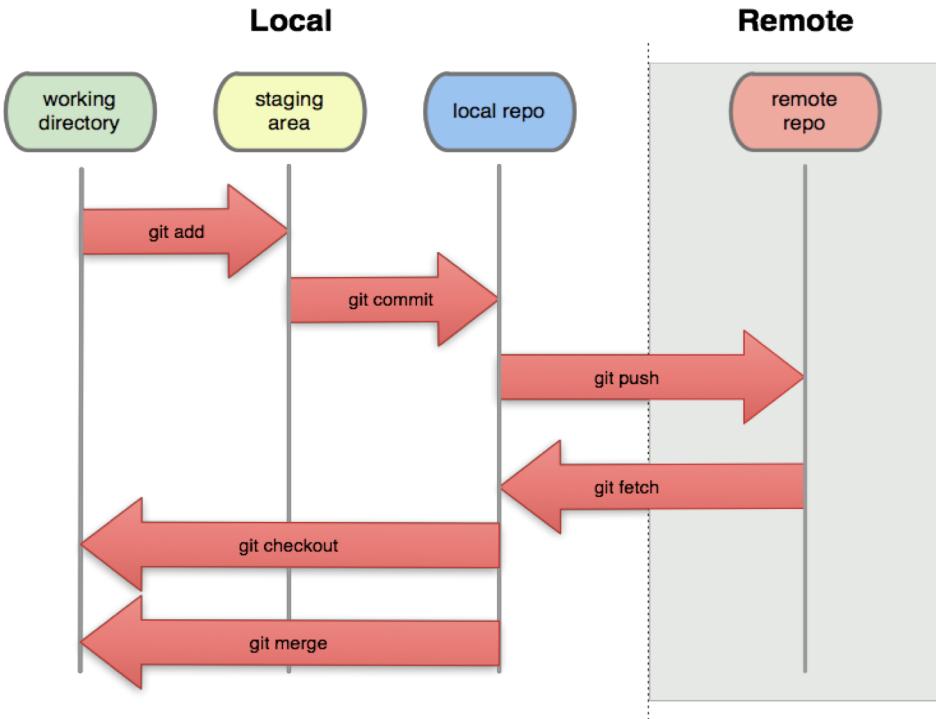
**#git push -u origin master**: Push committed code to remote repository in master branch.

**#git clone <GIT Remote URL>** : Cloning remote git repository.





# - Introduction





## - More GIT Commands

- ❖ \$ git branch <branchName>
- ❖ \$ git checkout <branchName>
- ❖ \$ git merge <branchName>
- ❖ \$ git rebase <branchName>
- ❖ \$ git log
- ❖ \$ git branch -d|-D <branchName>
- ❖ \$ git fetch
- ❖ \$ git pull





## - Reference URLs

<https://help.github.com/articles/creating-releases/> ---> Tag/Releases

<https://help.github.com/articles/creating-a-pull-request-from-a-fork/> ---> Pull Request

<https://help.github.com/articles/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent/> ---> SSH Key

<https://help.github.com/articles/creating-a-pull-request/> ---> Pull Request



# Questions ?



Thank you  
Mithun Technologies  
+91-9980923226  
[devopstrainingblr@gmail.com](mailto:devopstrainingblr@gmail.com)

