**Git** is a revision control system, a tool to manage your source code history.

**GitHub** is a hosting service for Git repositories.

So they are not the same thing: **Git** is the **tool**, **GitHub** is the **service for projects that use Git**.

1. Download the latest [Git for Windows installer](https://git-for-windows.github.io/).
2. When you've successfully started the installer, you should see the **Git Setup** wizard screen. Follow the **Next** and **Finish** prompts to complete the installation. The default options are pretty sensible for most users.
3. Open a Command Prompt (or Git Bash if during installation you elected not to use Git from the Windows Command Prompt).
4. Run the following commands to configure your Git username and email using the following commands, replacing Emma's name with your own. These details will be associated with any commits that you create:
5. git config --global user.name “PrathapReddy456”

git config --global user.email "prathapreddy876@gmail.com”

1. *Optional: Install the Git credential helper on Windows*

Bitbucket supports pushing and pulling over HTTP to your remote Git repositories on Bitbucket. Every time you interact with the remote repository, you must supply a username/password combination. You can store these credentials, instead of supplying the combination every time, with the [Git Credential Manager for Windows](https://github.com/Microsoft/Git-Credential-Manager-for-Windows).

**IMP Commands:**

**Git Ref log:** To see commit Id’s

**Git branch Branchname:** To create new branch (it just create new branch, but u still in master only)

**Git Checkout Branchname(Which branch u would like to move):** To change one branch to another branch.

**Git checkout –b branch name:** It creates new branch and it will move u from parent to that new created branch.

**Git Merge Branchname(parent branch name):** To merge child branch to parent.

**To push the file to Repository:**

* Get File Or Folder location from Eclipse(IDE)
* Go to Windows Command Prompt and Enter **cd (above)file location.**
* In cmd enter **git init**
* **.git** file is creates in above folder

### Enter from github

### push an existing repository from the command line

git remote add origin git@github.com:PrathapReddy456/CucumberFrm1.git

* To check the status in cmd
  + Enter **git status**

These files in Red colour those are pending.

* Enter **git add .** in cmd

All files added from folder

* Again check the status in cmd
  + Enter **git status**

come in green colour, and all files ready to push

* Before push we have to commit
* To commit in cmd we have to enter **git commit –m “ Any Comment”.**
* Now time to push the code

**But we need “SSH key ” permission to push the code to repository. We have to setup ssh key in our github a/c.**

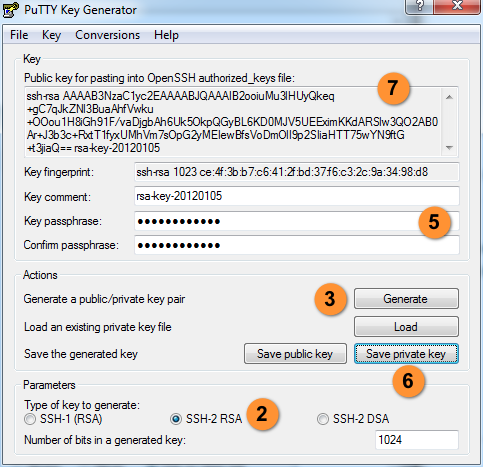
**With Putty**

# Manually generating your SSH key in Windows

### [Generating an SSH key](https://docs.joyent.com/public-cloud/getting-started/ssh-keys/generating-an-ssh-key-manually/manually-generating-your-ssh-key-in-windows#generating-an-ssh-key)

To generate an SSH key with PuTTYgen, follow these steps:

1. Open the PuTTYgen program.
2. For **Type of key to generate**, select **SSH-2 RSA**.
3. Click the **Generate** button.
4. Move your mouse in the area below the progress bar. When the progress bar is full, PuTTYgen generates your key pair.
5. Type a passphrase in the **Key passphrase** field. Type the same passphrase in the **Confirm passphrase** field. You can use a key without a passphrase, but this is not recommended.
6. Click the **Save private key** button to save the private key. Warning! You **must** save the private key. You will need it to connect to your machine.
7. Right-click in the text field labeled **Public key for pasting into OpenSSH authorized\_keys file** and choose **Select All**.
8. Right-click again in the same text field and choose **Copy**.



### [Importing your SSH key](https://docs.joyent.com/public-cloud/getting-started/ssh-keys/generating-an-ssh-key-manually/manually-generating-your-ssh-key-in-windows#importing-your-ssh-key)

Now you must import the copied SSH key to the portal.

1. After you copy the SSH key to the clipboard, return to [your account page](https://my.joyent.com/main/#!/account).
2. Choose to **Import Public Key** and paste your SSH key into the Public Key field.
3. In the **Key Name** field, provide a name for the key. **Note**: although providing a key name is optional, it is a best practice for ease of managing multiple SSH keys.
4. **Add** the key. It will now appear in your table of keys under SSH.

## With Git Bash

## Generating a new SSH key and adding it to the ssh-agent

After you've checked for existing SSH keys, you can generate a new SSH key to use for authentication, then add it to the ssh-agent.

If you don't already have an SSH key, you must [generate a new SSH key](https://help.github.com/articles/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent/#generating-a-new-ssh-key). If you're unsure whether you already have an SSH key, check for [existing keys](https://help.github.com/articles/checking-for-existing-ssh-keys).

If you don't want to reenter your passphrase every time you use your SSH key, you can [add your key to the SSH agent](https://help.github.com/articles/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent/#adding-your-ssh-key-to-the-ssh-agent), which manages your SSH keys and remembers your passphrase.

### Generating a new SSH key

1. Open Git Bash.
2. Paste the text below, substituting in your GitHub email address.
3. ssh-keygen -t rsa -b 4096 -C "prathap456qa@gmail.com"

This creates a new ssh key, using the provided email as a label.

Generating public/private rsa key pair.

1. When you're prompted to "Enter a file in which to save the key," press Enter. This accepts the default file location.
2. Enter a file in which to save the key (/c/Users/you/.ssh/id\_rsa):[Press enter]
3. At the prompt, type a secure passphrase. For more information, see ["Working with SSH key passphrases"](https://help.github.com/articles/working-with-ssh-key-passphrases).
4. Enter passphrase (empty for no passphrase): [Type a passphrase]
5. Enter same passphrase again: [Type passphrase again]

### Adding your SSH key to the ssh-agent

Before adding a new SSH key to the ssh-agent to manage your keys, you should have [checked for existing SSH keys](https://help.github.com/articles/checking-for-existing-ssh-keys) and [generated a new SSH key](https://help.github.com/articles/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent#generating-a-new-ssh-key).

If you have [GitHub Desktop](https://desktop.github.com/) installed, you can use it to clone repositories and not deal with SSH keys. It also comes with the Git Bash tool, which is the preferred way of running git commands on Windows.

1. Ensure the ssh-agent is running:
   * If you are using the Git Shell that's installed with GitHub Desktop, the ssh-agent should be running.
   * If you are using another terminal prompt, such as Git for Windows, you can use the "Auto-launching the ssh-agent" instructions in "[Working with SSH key passphrases](https://help.github.com/articles/working-with-ssh-key-passphrases)", or start it manually:
   * # start the ssh-agent in the background
   * eval $(ssh-agent -s)
   * Agent pid 59566
2. Add your SSH private key to the ssh-agent. If you created your key with a different name, or if you are adding an existing key that has a different name, replace id\_rsa in the command with the name of your private key file.
3. ssh-add ~/.ssh/id\_rsa
4. [Add the SSH key to your GitHub account](https://help.github.com/articles/adding-a-new-ssh-key-to-your-github-account).

* It’s time to push the code now, in CMD enter
  + - **git push origin master**
* **Now you can see the folder structure with code in your github repository**
* If you add more files to your folder after push the code, just again do
  + - **git add .**
    - **git status**
    - **git commit –m “comment”**
    - **git push origin master**

**How to Pull the code from Repository:**

* Create a directory Where you want to pull code by manually or in cmd
  + In Cmd
    - **Cd..**(.. to go back)
    - **Cd..**
    - **Cd documents**(location)
    - **mkdir Pulling**(folder name where you gonna pull code)
    - **cd Pulling/ (/ to forward)**
* We have to Clone in cmd
* **git clone https://github.com/PrathapReddy456/Prathap** (github url)

**How to Import Above folder to Eclips**

* Go to Eclips
* Select File
* Select Import
* Select General
* Select Existing Projects into workspace
* Click Next
* Browse for Selec root directory
* Click Finish

**If any changes added to pulled code in repository, How to get those:**

* Go to Cmd
  + **Cd C:\Users\RITHWIK\new\Java\_My Work\Frmwrk** (Adding changes Project location from Eclips).
  + **git pull origin master**