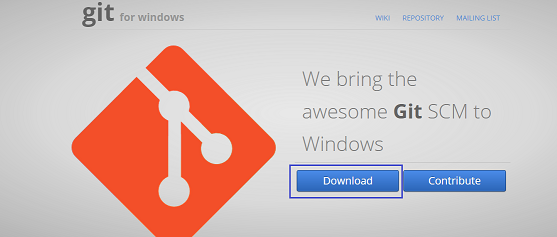
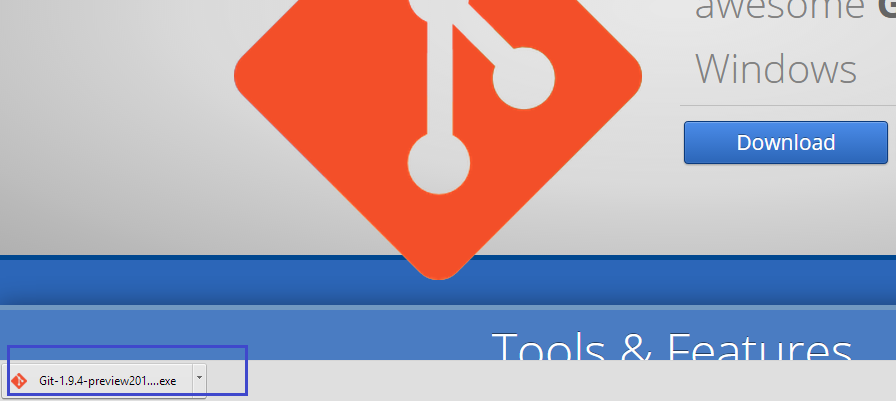
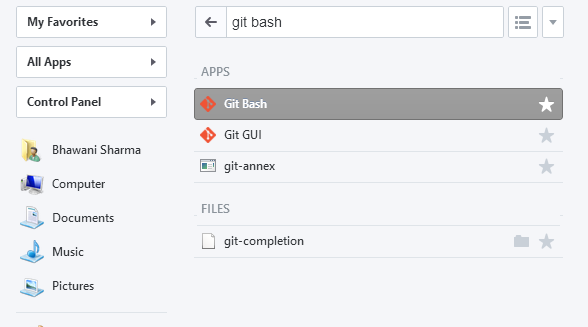
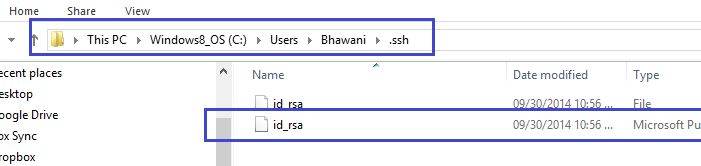
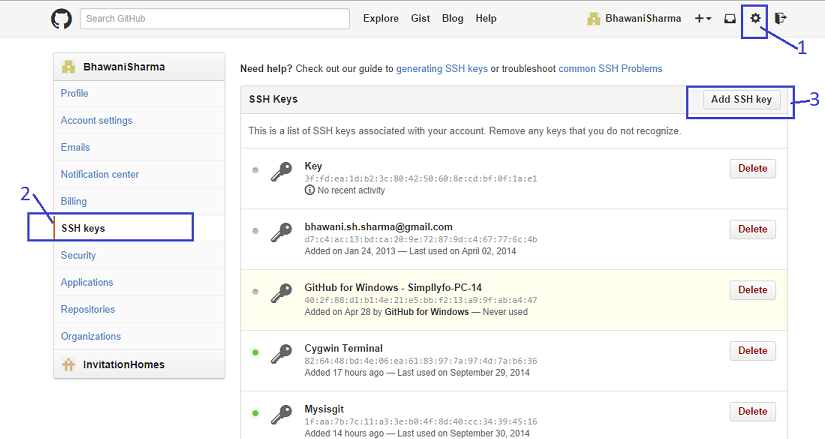
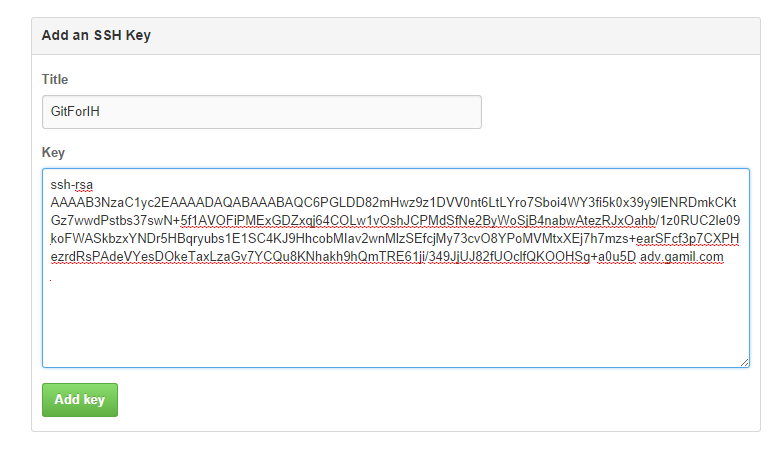
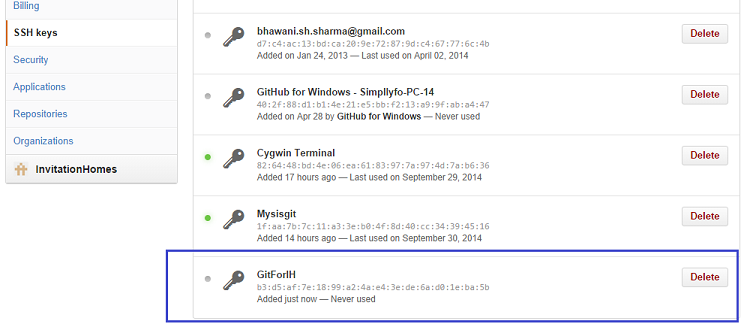
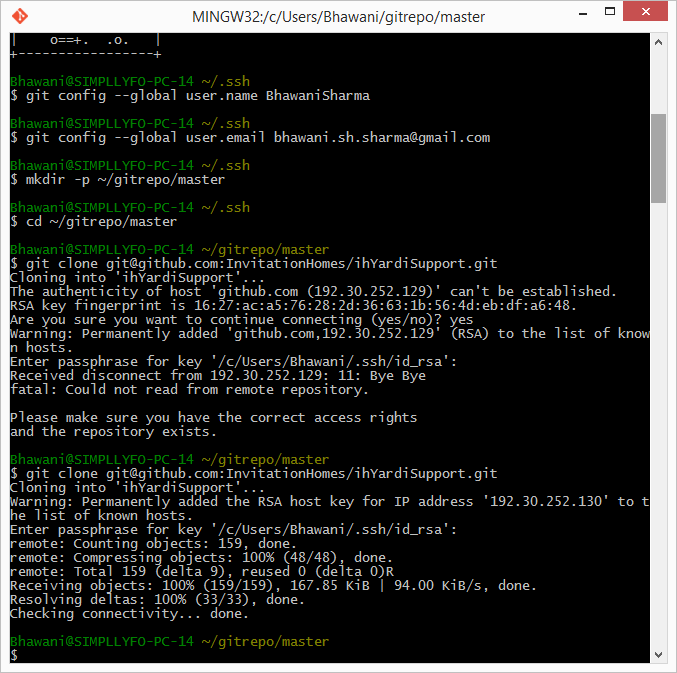
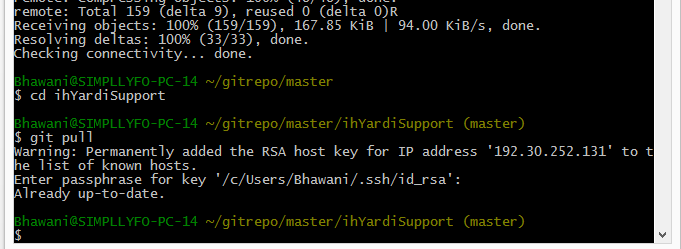
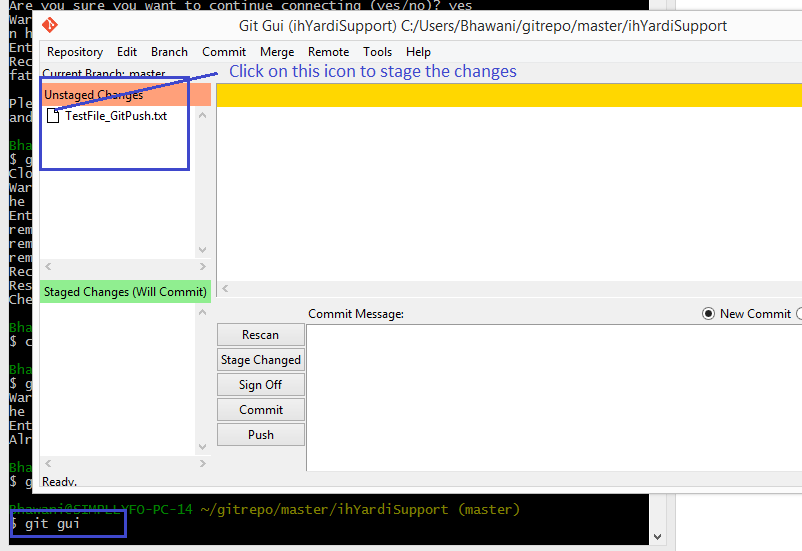
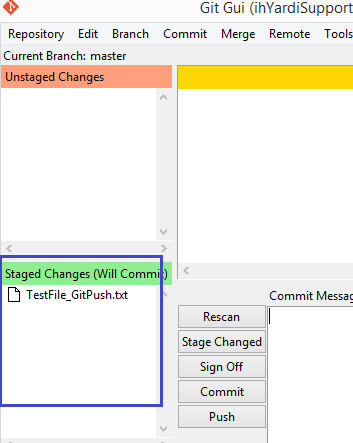
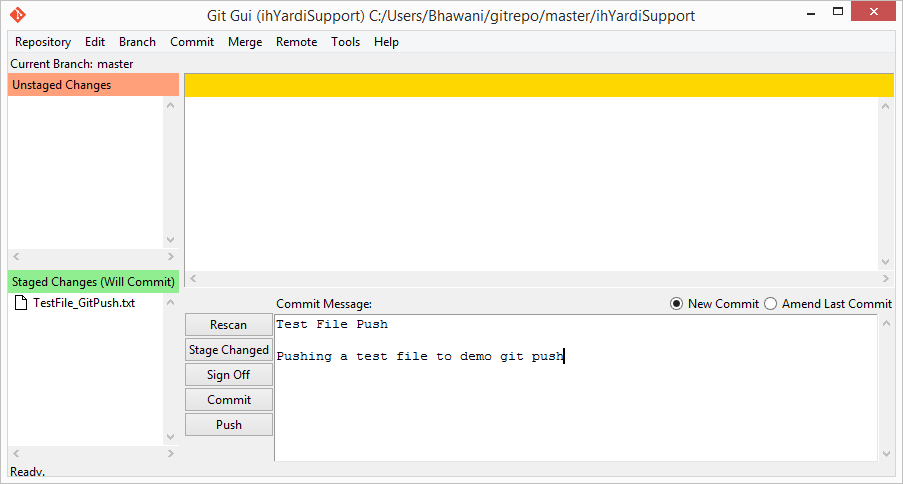
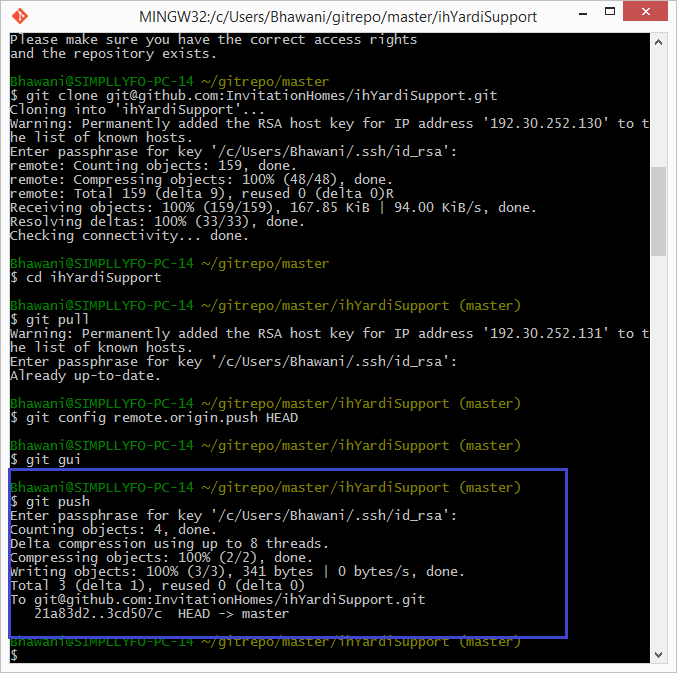
# Git Setup

1. Go to <http://msysgit.github.io/>.
2. Click on download button.  
     
   
3. Once the file is downloaded on machine, execute the exe file to install git.  
     
   
4. Choose all the default options.
5. Click on finish button to complete the Installation.
6. Now go to start menu and open Git Bash.  
     
   
7. Execute the following commands in Git Bash window.
   1. $ mkdir ~/.ssh
   2. $ cd ~/.ssh
8. Generate your SSH Key using the below command. Make sure to use the email address which is associated with your git account.  
   $ ssh-keygen -t rsa -C youremailaddress
9. When you run the step# 8, it will ask you to enter a file name. Enter id\_rsa here.
10. After running the step# 9, it will ask you to enter passphrase. This will be your password for pull and push request. Enter any password of your choice.
11. After running the step# 10, it will ask you to confirm your passphrase. Enter the same password as used in step# 10.
12. After running step# 11, you will see a new key fingerprint is generated in your Git Bash console.
13. Screen-Shot for step 7 – 12  
      
    
14. Now go to your Users -> User -> .ssh directory on local machine.
15. Open id\_rsa.pub in notepad or any other text editor.  
      
    
16. Copy the complete text from id\_rsa file. Make sure you don’t make any changes in id\_rsa file. This is encrypted text, so just copy it completely.
17. Now login to you github account and go to settings -> ssh section and click on “Add SSh key” button.  
      
    
18. Provide any unique name in Title box.
19. Paste the complete text in Key section which was copied in step# 16 and click on “Add Key” button.  
      
    
20. Newly added key will be displayed in SSH keys section.  
      
    
21. Now go back to your Git Bash console.
22. Execute following commands in Git Bash to setup your username and email address.  
    $ git config --global user.name "your github username"  
    $ git config --global user.email "your email address"
23. Run the following commands in Git Bash to clone your repository from GitHub.  
    $ mkdir -p ~/gitrepo/master  
    $ cd ~/gitrepo/master  
    $ git clone git@github.com:BhawaniSharma/Test-Automation.git
24. When you run the above clone command, it will ask you to enter passphrase. Use the same passphrase as entered in step# 10.
25. If it fails, use the clone command again and enter your passphrase.
26. Once repository is cloned, you will see a success message.
27. Screen-shot from step# 21-27.  
      
    
28. Run the following commands in Git Bash console.  
    $ cd Test-Automation  
    $ git config remote.origin.push HEAD
29. Now you are all setup to checkin and checkout code.

# Pull from Github Repository

1. To pull the new changes, use the following command in git Bash window.  
   $ git pull
2. Above command will ask you to enter passphrase. Use the same as in step# 10.
3. If there is nothing new to pull, it will say “Already up-to-date”.
4. Screen-shot from step# 28-32.  
     
   

# Push in Github Repository

1. Make your changes in your local folder as needed.
2. Run the following command in Git Bash window.  
   $ git gui
3. Above command will open a new window.
4. All the new files, updated files and deleted files will be displayed in “Unstaged Changes” section.  
     
   
5. Click on the file icon to stage these updated files.
6. Selected files will be moved from “Unstaged Changes” section to “Staged Changes” section.  
     
   
7. Now provide a commit message for the new changes you are pushing and click on “Commit” button.  
     
   
8. Close the “Git Gui” window and come back to the Git Bash console.
9. Run the following command to push your changes to Github repository.  
   $ git push
10. Above command will ask you to enter passphrase. Use the same passphrase as used in step# 10 in first section.
11. Verify success message in Git Console. Now your changes have been pushed to Github repository.
12. Screen-shot for step# 8-11.  
      
    
13. Verify changes on Github.  
      
    