

Set Up GIT

To set the global user name

\$ git config --global user.name "Rajesh Shankar"

To view the global user name

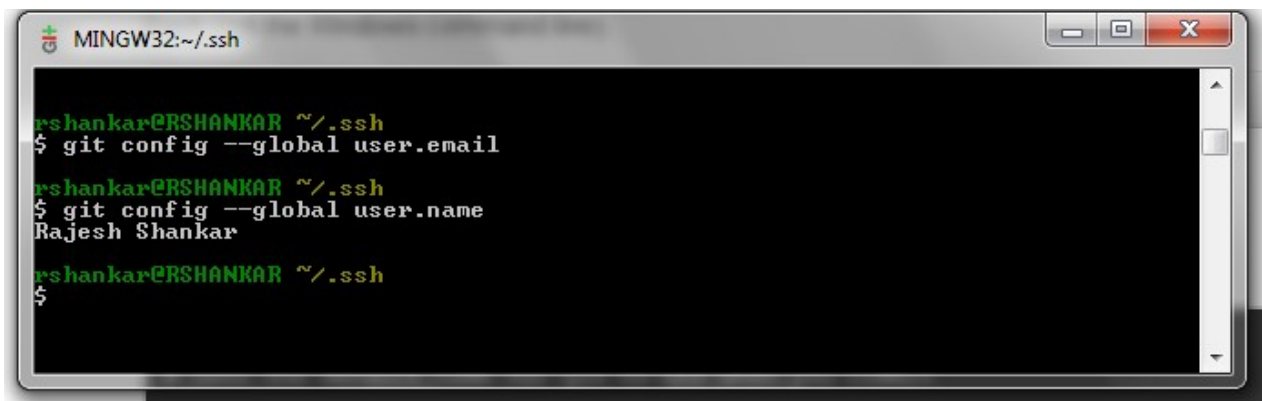
\$ git config --global user.name

To Set the global email Id

\$ git config --global user.email "rshankar@sampatti.com"

To view the global email id

\$ git config --global user.email



```
MINGW32: ~/.ssh
rshankar@RSHANKAR ~/\.ssh
$ git config --global user.email
rshankar@RSHANKAR ~/\.ssh
$ git config --global user.name
Rajesh Shankar
rshankar@RSHANKAR ~/\.ssh
$
```

Taking backup of SSH Keys

1. check for ssh keys on your computer. Openup Git Bash and run
2. Run the command `cd ~/.ssh`
3. Check the listing of ssh keys by command `ls`.
4. Take a back up of the existing ssh keys.
5. For taking a back up make a directory named `Key_backup`
6. `cp id_rsa* key_backup`
7. Remove the existing the ssh keys from the current folder.
8. `Rm id_rsa*`

```
$ cd ~/.ssh
# Checks to see if there is a directory named ".ssh" in your user directory
```

```
$ ls
# Lists all the subdirectories in the current directory
config id_rsa id_rsa.pub known_hosts

$ mkdir key_backup
# Makes a subdirectory called "key_backup" in the current directory

$ cp id_rsa* key_backup
# Copies the id_rsa keypair into key_backup

$ rm id_rsa*
# Deletes the id_rsa keypair
```

Generating the SSH keys

1. To Create a SSH keys type a Command in the Git Bash
2. `ssh-keygen -t rsa -c "Your email address"`.
3. Enter the file name where you want to store the ssh key.
4. Enter passphrase (empty for no pass phrase) : type the passkey
5. enter the confirm passkey again (same).
6. Copy the ssh key by this command: `$ clip < ~/.ssh/id_rsa.pub`

```
$ ssh-keygen -t rsa -C "your_email@youremail.com"
# Creates a new ssh key using the provided email
Generating public/private rsa key pair.
Enter file in which to save the key (/your_home_path/.ssh/id_rsa):
```

```
Enter passphrase (empty for no passphrase): [Type a passphrase]
Enter same passphrase again: [Type passphrase again]
```

```
Your identification has been saved in /your_home_path/.ssh/id_rsa.
Your public key has been saved in /your_home_path/.ssh/id_rsa.pub.
The key fingerprint is:
01:0f:f4:3b:ca:85:d6:17:a1:7d:f0:68:9d:f0:a2:db your_email@youremail.com
```

Adding the ssh keys

1. Go to account settings
2. Click SSH Keys in the left sidebar.
3. Click Add SSH Key
4. Paste your SSH Keys in to the key field.
5. Click Add key
6. Confirm the action by entering your gitHub password.

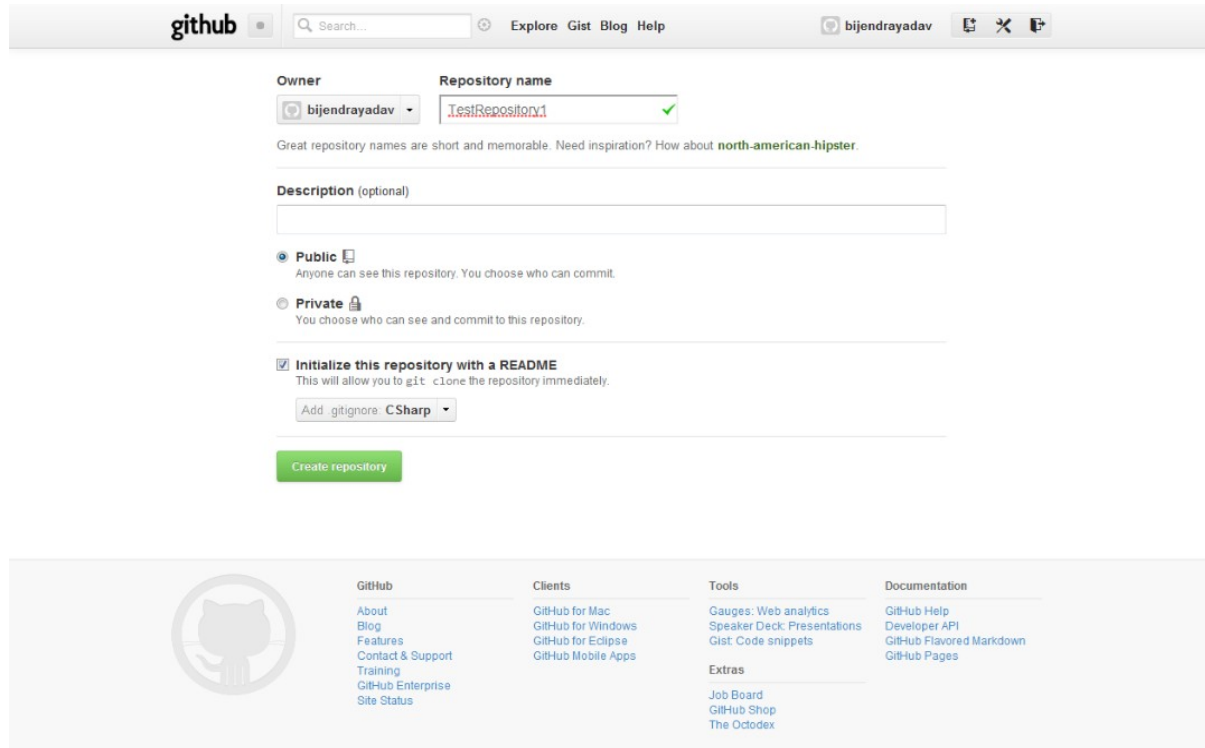
The screenshot shows the GitHub account settings page for user 'bijendrayadav'. The left sidebar contains navigation links: Profile, Account Settings, Emails, Notification Center, Billing, Payment History, SSH Keys (highlighted), Security History, Applications, Repositories, and Organizations. The main content area is titled 'SSH Keys' and includes a link to 'Add SSH Key'. Below this, it states 'There are no SSH keys with access to your account.' A section titled 'Add an SSH Key' contains a 'Title' field with the value 'TestSSHKey' and a 'Key' field containing a long RSA private key. At the bottom of the 'Add an SSH Key' section is a green 'Add key' button. The footer of the page includes the GitHub logo, navigation links for GitHub, Clients, Tools, and Documentation, and a 'PRIVATE REPOS' indicator showing 0 of 0.

Test Your ssh Is verified by the github

1. type a command on the github bash : `$ ssh -T git@github.com`

```
$ ssh -T git@github.com
# Attempts to ssh to github
```

Creating new Repository



The screenshot shows the GitHub 'Create new repository' page. At the top, there's a navigation bar with the GitHub logo, a search bar, and links for 'Explore', 'Gist', 'Blog', and 'Help'. The user 'bijendrayadav' is logged in. The form has two main sections: 'Owner' and 'Repository name'. The 'Owner' is set to 'bijendrayadav'. The 'Repository name' is 'TestRepository1', which is marked as valid with a green checkmark. Below this, a message says: 'Great repository names are short and memorable. Need inspiration? How about [north-american-hipster](#).' There's a 'Description (optional)' text area. Under 'Visibility', 'Public' is selected, with a note: 'Anyone can see this repository. You choose who can commit.' 'Private' is also an option, with a note: 'You choose who can see and commit to this repository.' There's a checkbox for 'Initialize this repository with a README', which is checked, with a note: 'This will allow you to `git clone` the repository immediately.' Below this, there's a dropdown menu for 'Add .gitignore' with 'CSharp' selected. At the bottom of the form is a green 'Create repository' button. The footer of the page contains the GitHub logo, a list of links for 'GitHub' (About, Blog, Features, Contact & Support, Training, GitHub Enterprise, Site Status), 'Clients' (GitHub for Mac, GitHub for Windows, GitHub for Eclipse, GitHub Mobile Apps), 'Tools' (Gauges: Web analytics, Speaker Deck: Presentations, Gist: Code snippets, Extras: Job Board, GitHub Shop, The Octodex), and 'Documentation' (GitHub Help, Developer API, GitHub Flavored Markdown, GitHub Pages). At the very bottom, it says 'github SOCIAL CODING' and 'Terms of Service Privacy Security © 2012 GitHub Inc. All rights reserved.'

Creating the clone of the repository

1. Open the github and login to the git hub.
2. Click on the repository i.e the section of the repositories.
3. Click the ssh small button and copy the url.
4. Right click on the folder where you want this clone upon.
5. Paste the url to the Url section.
6. Click ok button.
7. After pressing the button it'll ask you to putting the ssh password.

