Git, Github and Wordpress

Session Organizer Created: **Farhaj Ahmed & Asher Ahsan**

Speaker: **Muhammad Ilyas**

Follow me: **Github.com/ilyasx**

Presentation content credit: **Andrew Kerr & Otto Kekäläinen**

Important Links:

All Slides available here:

Session Feedback:

http://tiny.cc/githubSessionFeedback

Outline

Story of Git

Git /ɡɪt/

”A silly, incompetent, stupid, annoying or childish person.”

http://en.wiktionary.org/wiki/git

"I'm an egotistical bastard, so I name all my projects after myself. First Linux, now Git”

Linus Torvalds, PC World. 2012-07-14

Linus needed a new source code revision manager for Linux, and none of the available options in 2005 where good enough, so he wrote his own in.

Kernel 2.6.12 was the first release managed by Git and version 1.0 of Git was released in December 2005.

Linux kernel Source Code: https://github.com/torvalds/linux

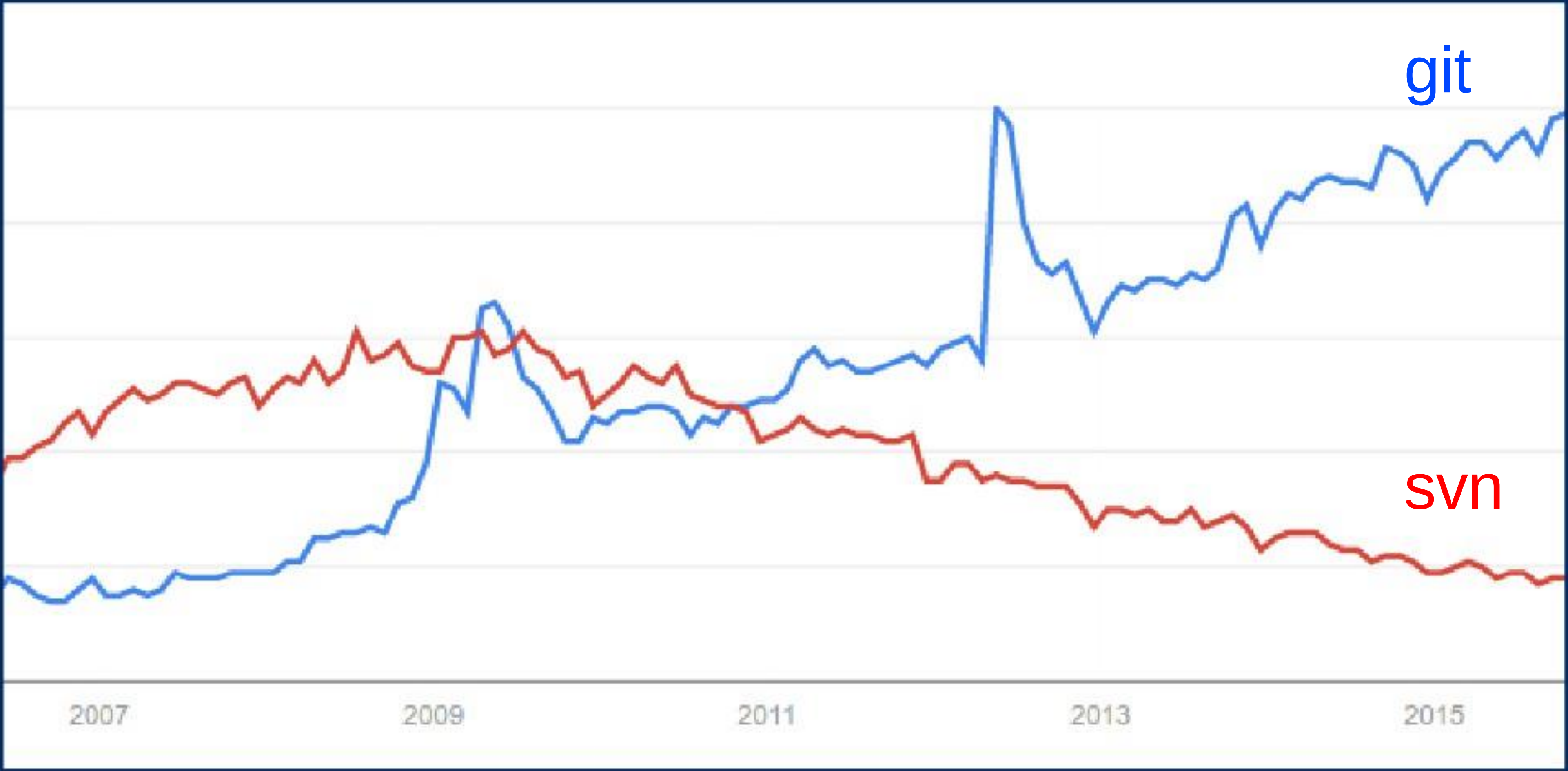
Design goals of Git:

- distributed revision management
- protection against corruption,

both accidental and hostile

- **speed**

Git popularity according to Google Trends



...but adoption would be faster if it was not so difficult to use.

Originally Linus did not intend end users to use Git directly, instead he tried to delegate to somebody else the task of making the actual command line interface. We are still waiting for it...

Luckily Git has been simplified and documentation has improved over time, but some Git commands still refer to Git internals that are difficult to grasp.

E.g. git-push: Update remote refs along with associated objects.

Git might feel difficult at first, but once you learn it, **you never want to go back** to anything less flexible and powerful.

Why is this useful?

Collaboration

How does this work?

How does this work?

1. Download code from remote server

How does this work?

1. Download code from remote server
2. Make your changes

How does this work?

1. Download code from remote server
2. Make your changes
3. Upload code to remote server

How does this work?

1. Download code from remote server
2. Make your changes
3. Upload code to remote server
4. Version control figures out what has changed and applies those changes to the codebase

What experts say about git ??

Interview with Jeffrey Middleton

Cool, so let's talk about git

Install on Linux:

```
sudo apt-get install git

sudo yum install git
```

Install on Windows:

https://desktop.github.com/

Basic commands

Define the author of commits

```
git config --global user.name “Username"
git config --global user.email “your@email.com”
```

Git Basics

1. Make a new directory

Open up a git bash!

or

Linux terminal

2. **git init**

3. **git status**

4. Make a file

5. **git status**

6. **git add .**

6. **git add .**

- "staged files" are files that are ready to be committed to git
- "unstaged files" are files that have changes that have not been prepared to be committed

can also use **git add [filename]**
for individual files

7. **git status**

8. **git commit -m 'Summary of changes'**

9. **git log**

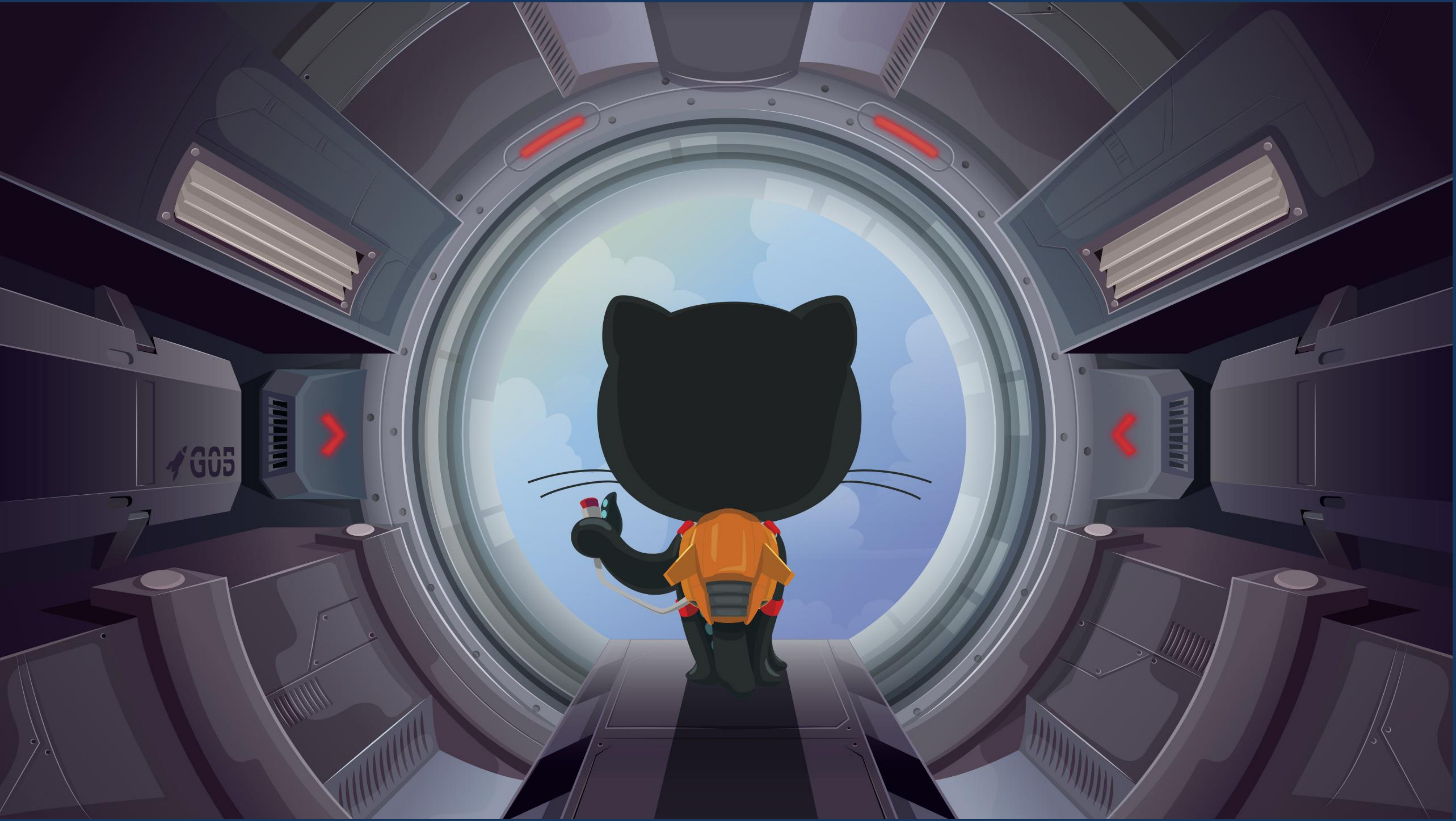
Any questions?

How to write a good commit message

- Your attitude towards commit messages should be the same as for code: it is written once, but read thousands of times.
- Don't explain how was done, that is visible in the diff anyway. Explain what the intention was and **why** it was made.
- Use imperative form “Fix typo” (instead of “Fixed typo”)
- Keep subject line short and sweet, under 72 chars. Body can be verbose.
- Use proper English. Capital letters. Reference issue identifiers is possible.
- Looking for a good example? How about one by Linus himself?

  https://github.com/torvalds/linux/commit/fc90888

Let Get started Github !!!



GitHub

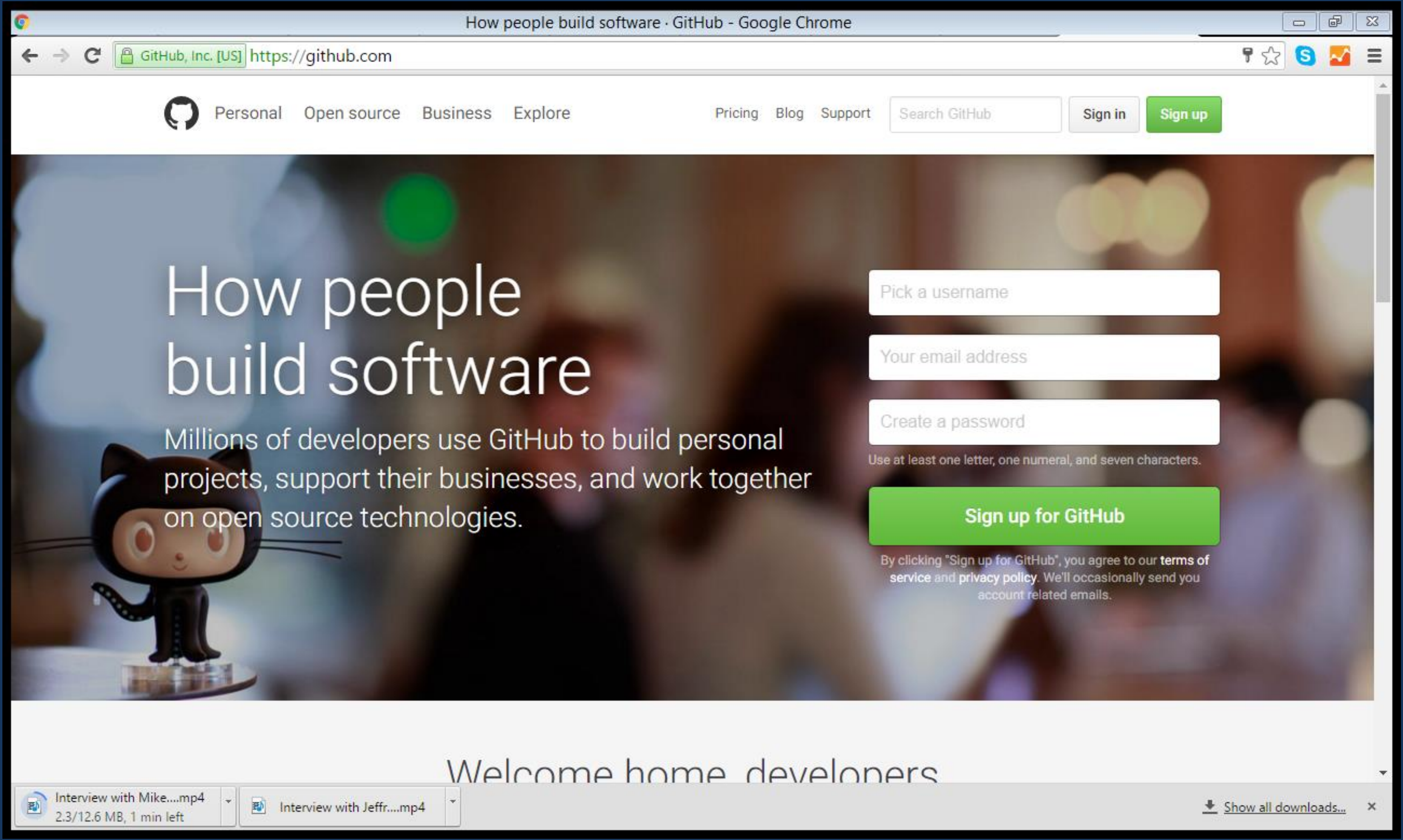
Powerful collaboration, code review, and code management for

open source and private projects.

Helps put a GUI on top of git!

Sign up for GitHub at github.com

Open link github.com


How people build software · GitHub - Google Chrome
GitHub, Inc. [US] https://github.com
Personal
Open source
Business
Explore
Pricing
Blog
Support
Search GitHub
Sign in
Sign up

How people build software

Millions of developers use GitHub to build personal projects, support their businesses, and work together on open source technologies.

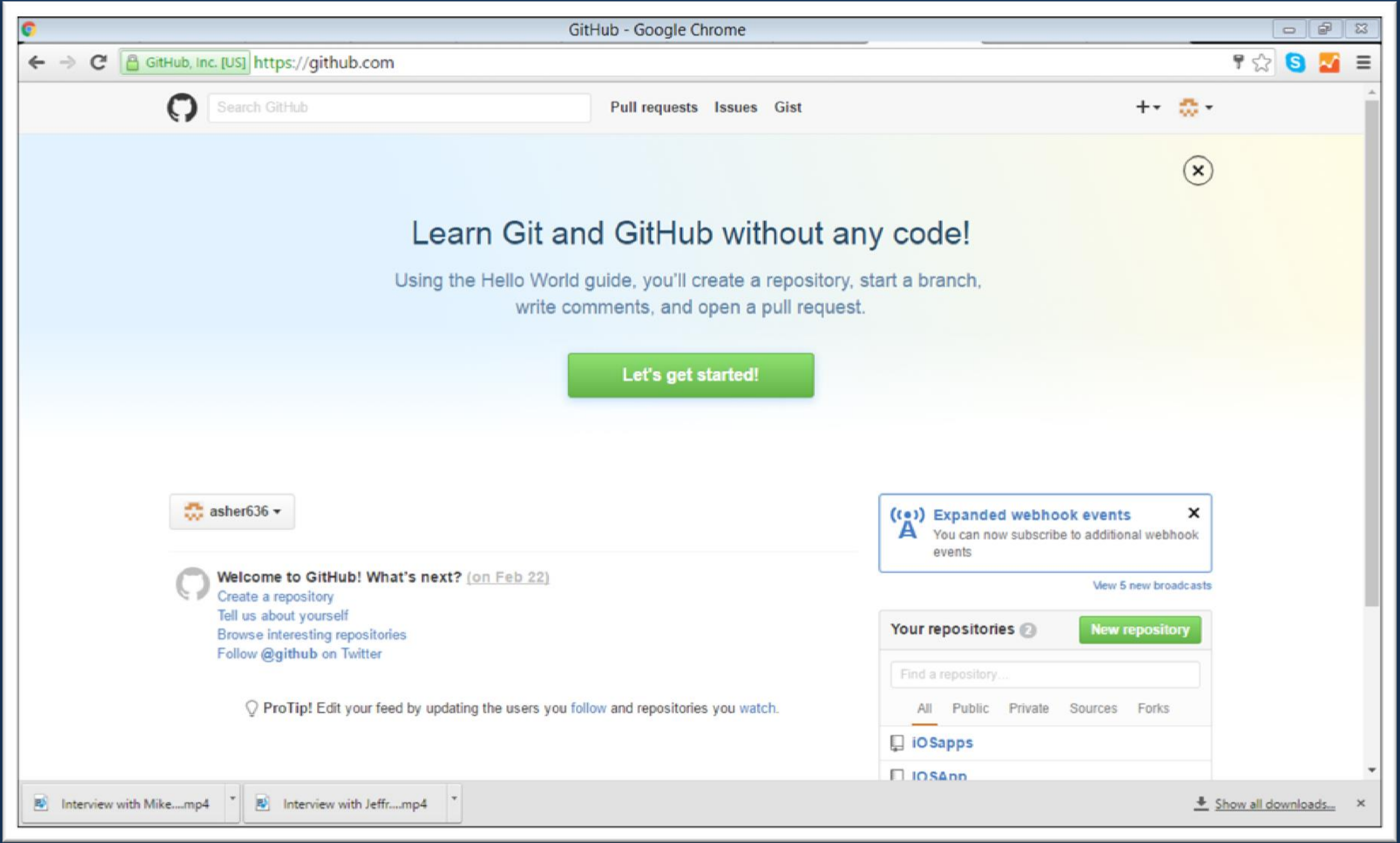
Use at least one letter, one numeral, and seven characters.

By clicking "Sign up for GitHub", you agree to our terms of service and privacy policy. We'll occasionally send you account related emails.

Welcome home, developers

Once signed up, create a repository

Register and Login your account



GitHub - Google Chrome

GitHub, Inc. [US] https://github.com

Search GitHub

Pull requests Issues Gist

Learn Git and GitHub without any code!

Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.

Let's get started!

asher636

Welcome to GitHub! What's next? (on Feb 22)
Create a repository
Tell us about yourself
Browse interesting repositories
Follow @github on Twitter

ProTip! Edit your feed by updating the users you follow and repositories you watch.

Expanded webhook events
You can now subscribe to additional webhook events

View 5 new broadcasts

Your repositories 2 New repository

Find a repository...

All Public Private Sources Forks

iOSapps

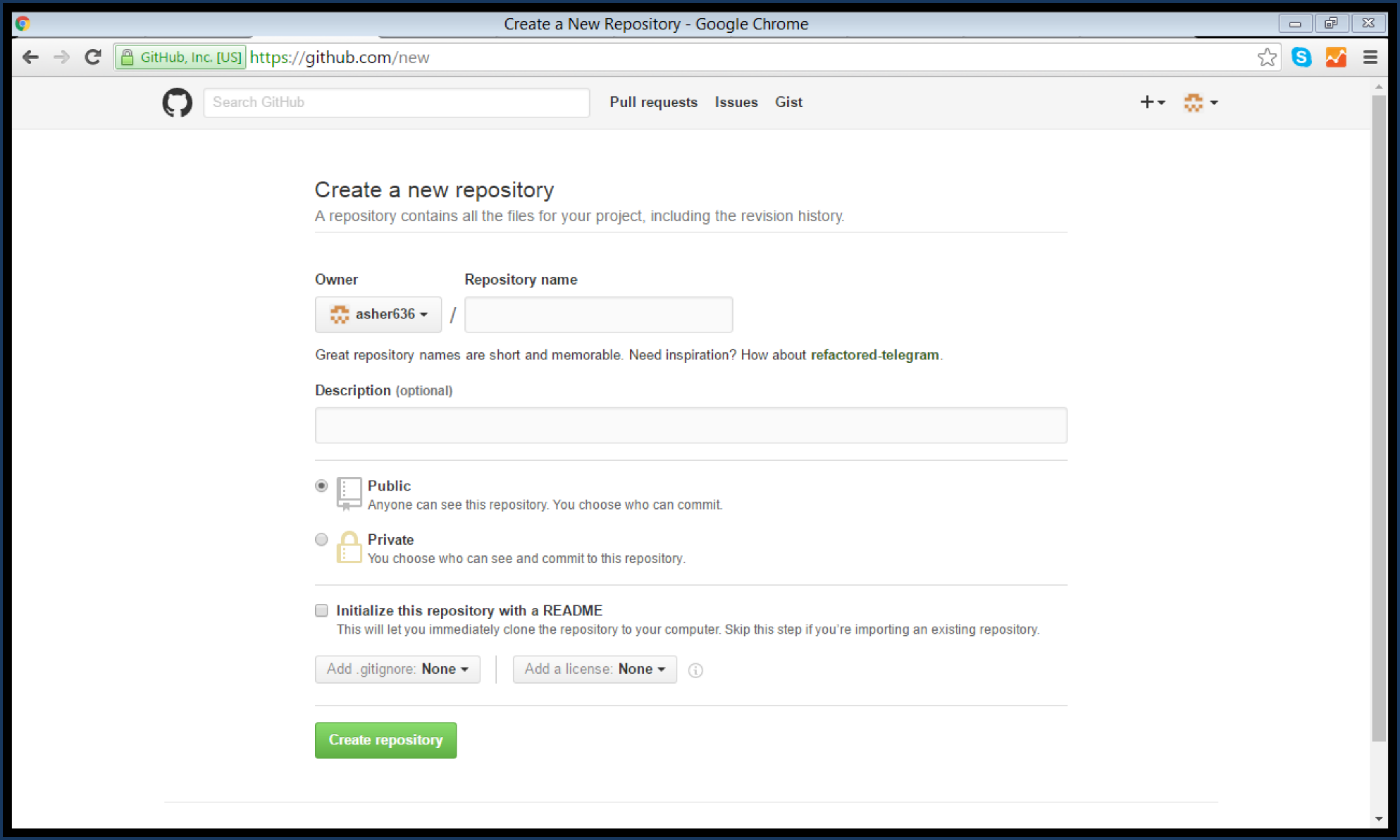
IOSApp

Interview with Mike....mp4
Interview with Jeffr....mp4
Show all downloads...

Hello world to Github

Create a hello world repository

New repository


Create a New Repository - Google Chrome

GitHub, Inc. [US] https://github.com/new

Create a new repository

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

Owner: asher636 / Repository name:

Great repository names are short and memorable. Need inspiration? How about **refactored-telegram**.

Description (optional)

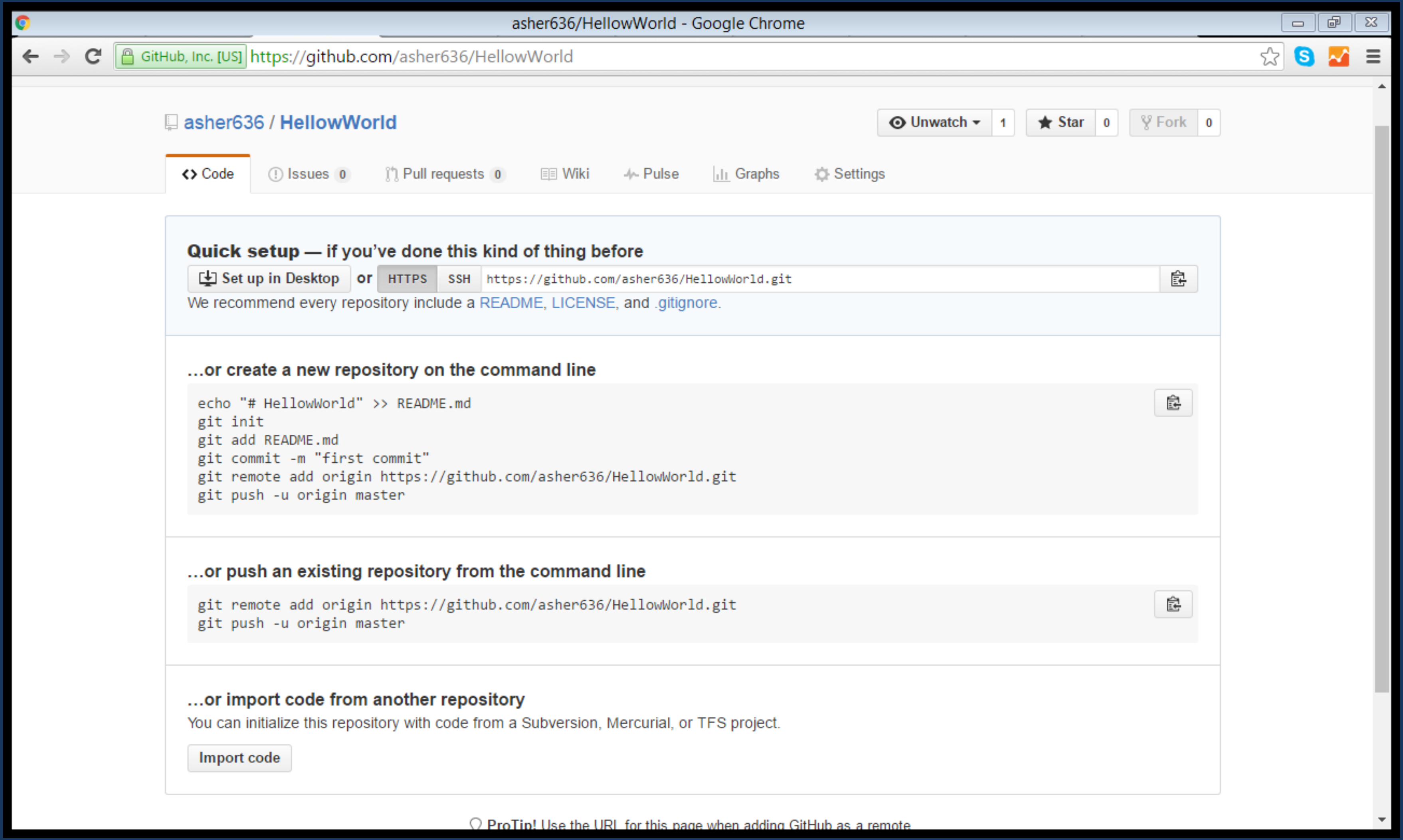
(●) **Public**
Anyone can see this repository. You choose who can commit.

( ) **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**

Let get started


asher636/HellowWorld - Google Chrome
GitHub, Inc. [US] https://github.com/asher636/HellowWorld
asher636 / HellowWorld
Unwatch 1
Star 0
Fork 0
Code
Issues 0
Pull requests 0
Wiki
Pulse
Graphs
Settings

Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS SSH https://github.com/asher636/HellowWorld.git

We recommend every repository include a README, LICENSE, and .gitignore.

…or create a new repository on the command line

```
echo "# HellowWorld" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/asher636/HellowWorld.git
git push -u origin master
```

…or push an existing repository from the command line

```
git remote add origin https://github.com/asher636/HellowWorld.git
git push -u origin master
```

…or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

Import code

ProTip! Use the URL for this page when adding GitHub as a remote.

What we have to do now??

1. git pull

To review

1. git pull

2. git checkout -b branch

To review

1. git pull
2. git checkout -b branch
3. Make changes

To review

1. git pull
2. git checkout -b branch
3. Make changes
4. git add .

To review

1. git pull
2. git checkout -b branch
3. Make changes
4. git add .
5. git commit -m 'commit summary'

To review

1. git pull
2. git checkout -b branch
3. Make changes
4. git add .
5. git commit -m 'commit summary'
6. git checkout master

To review (continued...)

1. git pull

To review (continued...)

1. git pull
2. git merge branch

To review (continued...)

1. git pull
2. git merge branch
3. git push

To review (continued...)

1. git pull
2. git merge branch
3. git push
4. Do it all again!

Git and GitHub can help teams work more efficiently on code

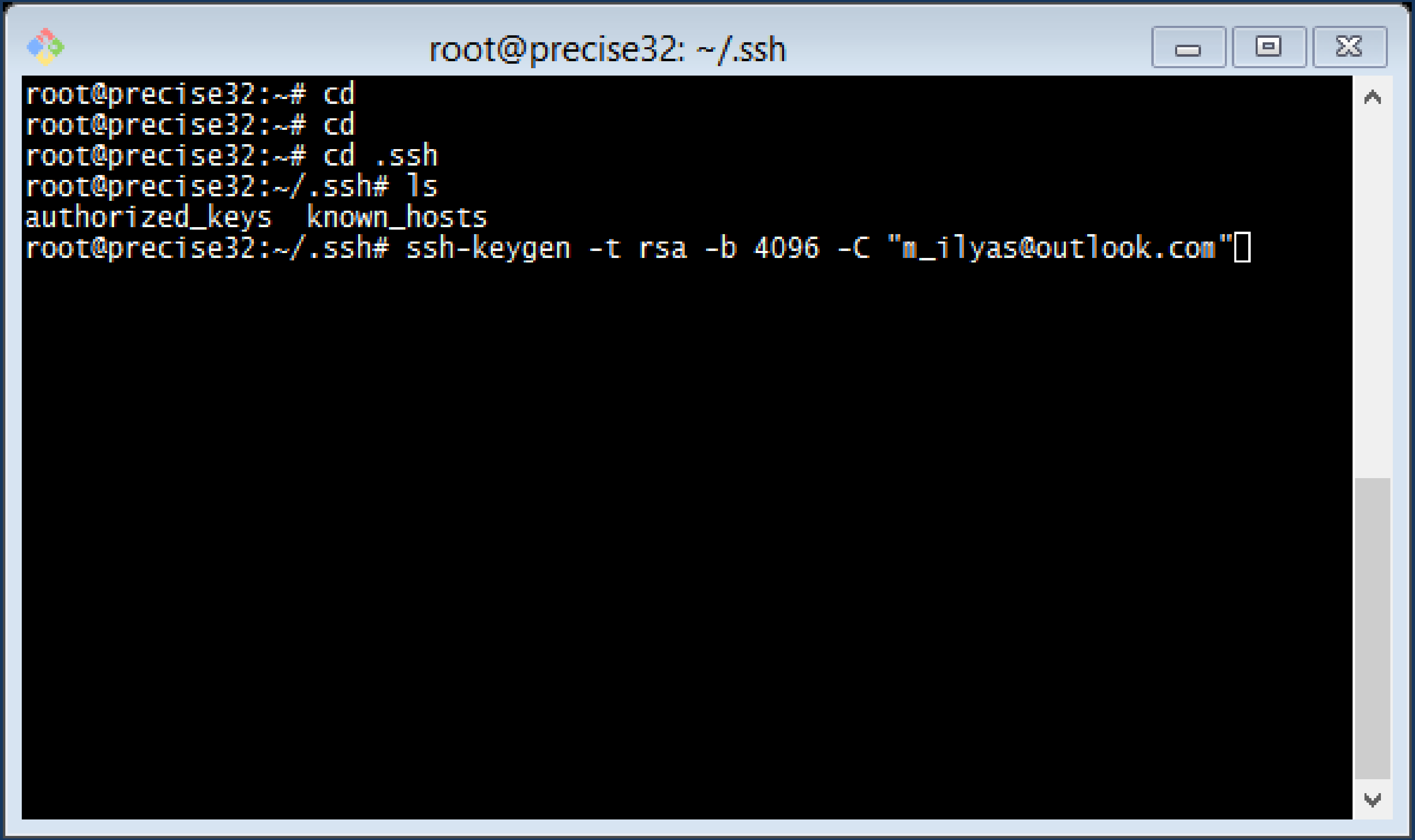
Any questions?

Setup RSA keys

- Open git bash
- Type cd
- Mkdir .ssh
- Cd .ssh
- Insert this command

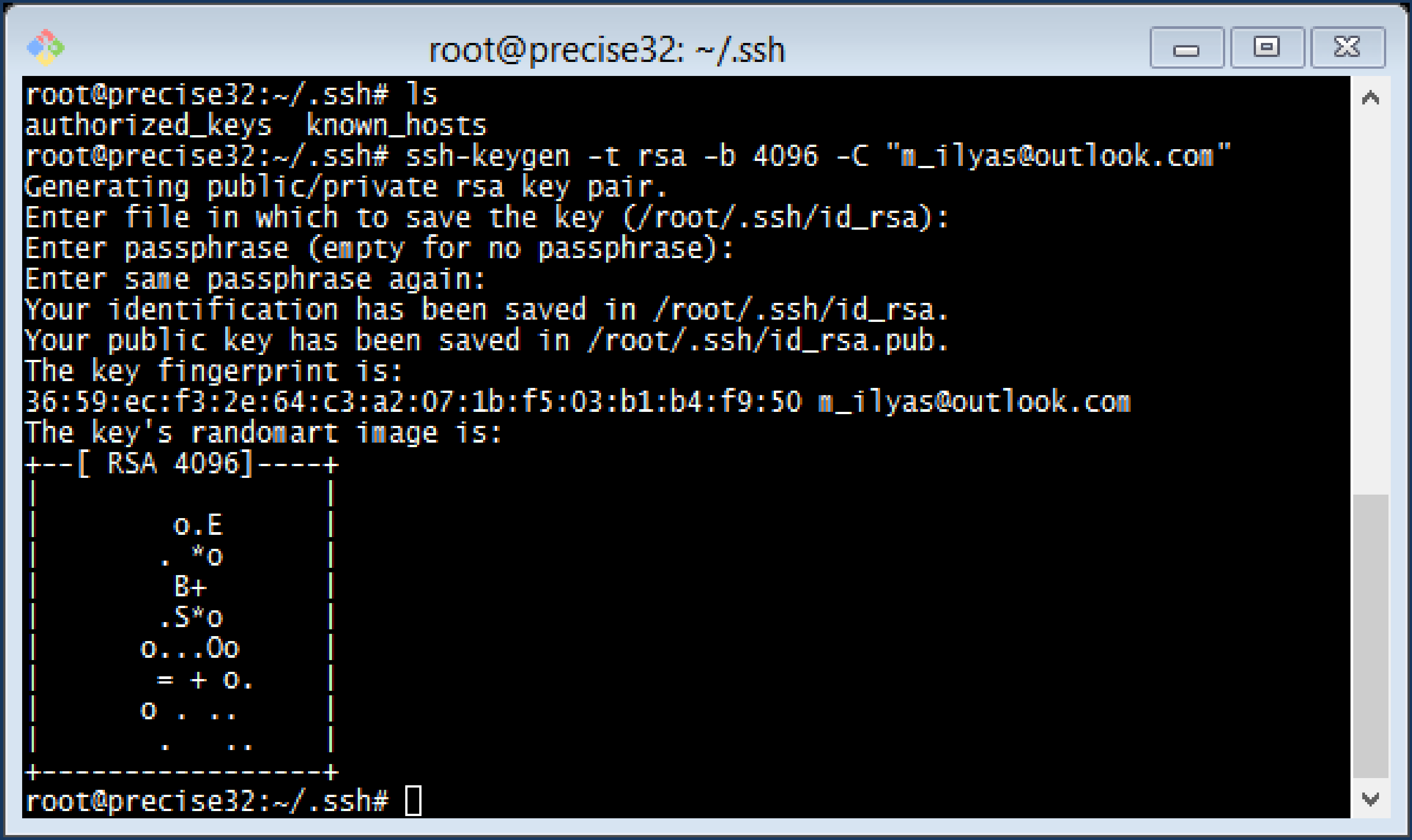
ssh-keygen -t rsa -b 4096 -C “your@email.com“

Setup RSA keys (cont)



```
root@precise32:~# cd
root@precise32:~# cd
root@precise32:~# cd .ssh
root@precise32:~/.ssh# ls
authorized_keys  known_hosts
root@precise32:~/.ssh# ssh-keygen -t rsa -b 4096 -C "m_ilyas@outlook.com"
```

Setup RSA keys (cont)

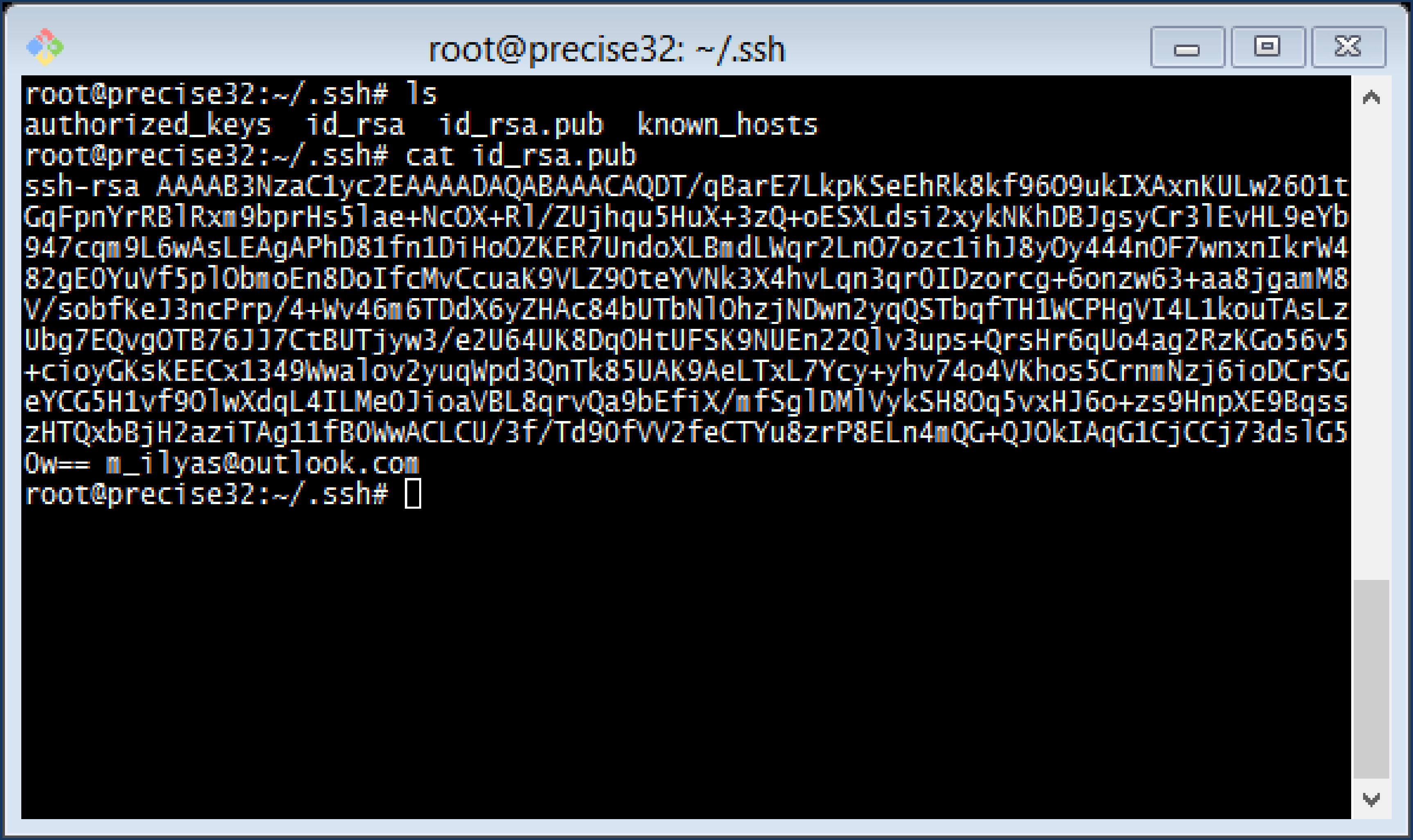


```
root@precise32:~/.ssh# ls
authorized_keys  known_hosts
root@precise32:~/.ssh# ssh-keygen -t rsa -b 4096 -C "m_ilyas@outlook.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
36:59:ec:f3:2e:64:c3:a2:07:1b:f5:03:b1:b4:f9:50 m_ilyas@outlook.com
The key's randomart image is:
+--[ RSA 4096]----+
|                 |
|        o.E      |
|       . *o      |
|        B+       |
|       .S*o      |
|      o...Oo     |
|       = + o.    |
|      o . ..     |
|       .   ..    |
+-----------------+
root@precise32:~/.ssh# 
```

Setup RSA keys (cont)

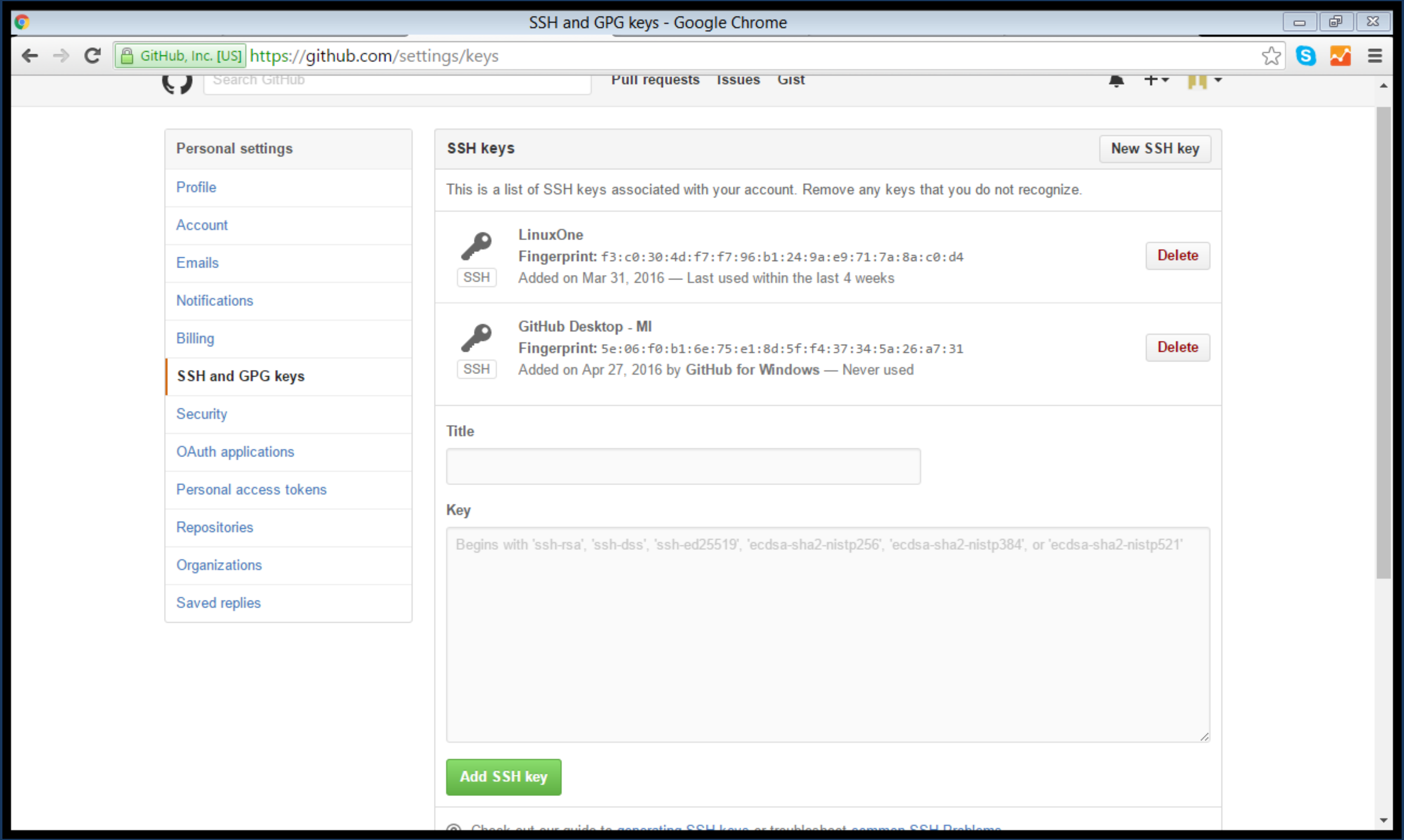
- Insert ls commad
- Insert cat id_rsa.pub
- Copy all data from id_rsa.pub
- Now open github.com ->setting -> ssh and gpg keys ->New SSH keys-> insert your id_rsa.pub key.

Setup RSA keys (cont)


root@precise32: ~/.ssh

```
root@precise32:~/.ssh# ls
authorized_keys  id_rsa  id_rsa.pub  known_hosts
root@precise32:~/.ssh# cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQDT/qBarE7LkpKSeEhRk8kf9609ukIXAxnKULw2601t
GqFpnYrRBlRxm9bprHs5lae+NcOX+Rl/ZUjhqu5HuX+3zQ+oESXLdsi2xykNKhDBJgsyCr3lEvHL9eYb
947cqm9L6wAsLEAgAPhD81fn1DiHoOZKER7UndoXLBmdLWqr2LnO7ozc1ihJ8yOy444nOF7wnxnIkrW4
82gEOYuVf5plObmoEn8DoIfcMvCcuaK9VLZ9OteYVNk3X4hvLqn3qrOIDzorcg+6onzw63+aa8jgamM8
V/sobfKeJ3ncPrp/4+Wv46m6TDdX6yZHAc84bUTbNlOhzjNDwn2yqQSTbqfTH1WCPHgVI4L1kouTAsLz
Ubg7EQvgOTB76JJ7CtBUTjyw3/e2U64UK8DqOHtUFSK9NUEn22Qlv3ups+QrsHr6qUo4ag2RzKGo56v5
+cioyGKsKEECx1349Wwalov2yuqWpd3QnTk85UAK9AeLTxL7Ycy+yhv74o4VKhos5CrnmNzj6ioDCrSG
eYCG5H1vf9OlwXdqL4ILMeOJioaVBL8qrvQa9bEfiX/mfSglDMlVykSH8Oq5vxHJ6o+zs9HnpXE9Bqss
zHTQxbBjH2aziTAg11fBOWwACLCU/3f/Td9OfVV2feCTYu8zrP8ELn4mQG+QJOkIAqG1CjCCj73dslG5
Ow== m_ilyas@outlook.com
root@precise32:~/.ssh# 
```

Setup RSA keys (cont)


SSH and GPG keys - Google Chrome

GitHub, Inc. [US] https://github.com/settings/keys

Personal settings

- Profile
- Account
- Emails
- Notifications
- Billing
- **SSH and GPG keys**
- Security
- OAuth applications
- Personal access tokens
- Repositories
- Organizations
- Saved replies

SSH keys New SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

SSH **LinuxOne**
Fingerprint: `f3:c0:30:4d:f7:f7:96:b1:24:9a:e9:71:7a:8a:c0:d4`
Added on Mar 31, 2016 — Last used within the last 4 weeks
Delete

SSH **GitHub Desktop - MI**
Fingerprint: `5e:06:f0:b1:6e:75:e1:8d:5f:f4:37:34:5a:26:a7:31`
Added on Apr 27, 2016 by **GitHub for Windows** — Never used
Delete

Title

Key

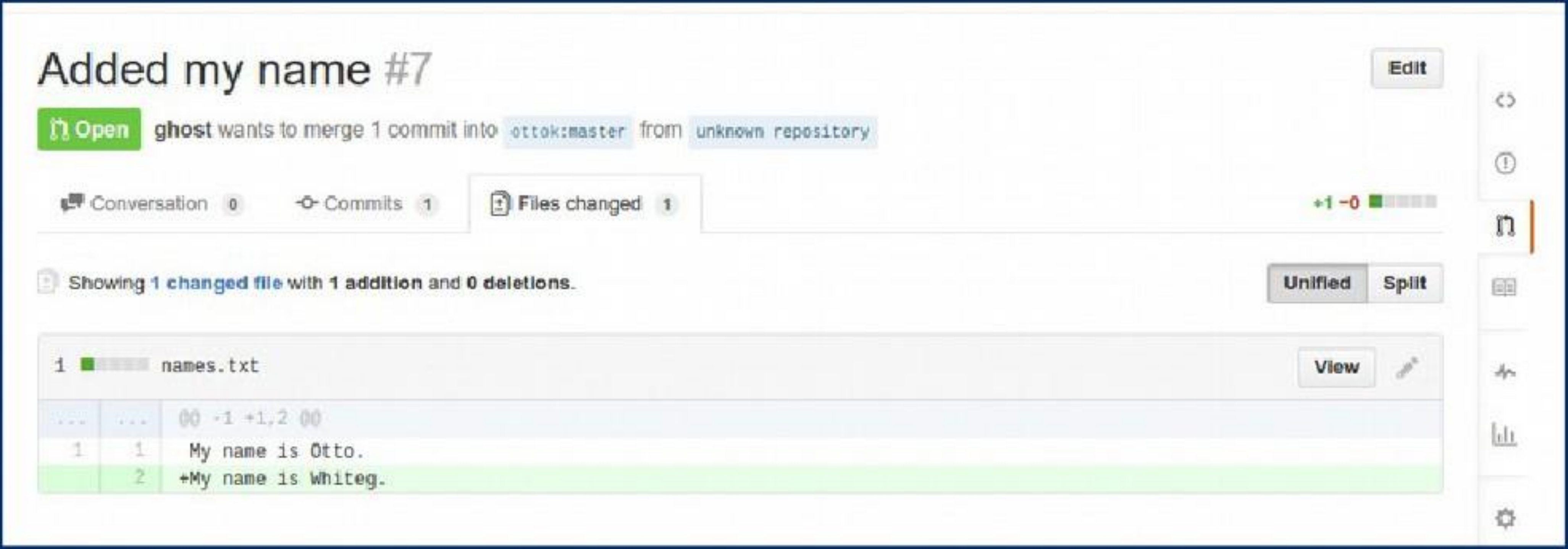
Begins with 'ssh-rsa', 'ssh-dss', 'ssh-ed25519', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', or 'ecdsa-sha2-nistp521'

Add SSH key

Congratulations you are successfully insert your ssh keys

Back to the command line...

```
git clone git://github.com/<yourname>/git-training.git
# edit names.txt
git commit -am “Added my name”
git push
# Open pull request on Github
```



Added my name #7

Open ghost wants to merge 1 commit into `ottok:master` from `unknown repository`

Showing 1 changed file with 1 addition and 0 deletions.

```
1 names.txt
@@ -1 +1,2 @@
 My name is Otto.
+My name is Whiteg.
```

The best part of Github is **pull requests**, and how they enable frictionless **collaboration** among millions of developers

Exercise:

- go to https://github.com and register you account
- Install github bash in your pc
- Create workspace/ new folder at your pc
- Create two files text1.txt and text2.txt and insert some random text.
- Initialize git to new directory by this ‘**git init’**
- Create repository at github.com
- git add .
- git commit –m “file text1 and text2 inserted”.
- git remote add origin [url]
- git push –u origin master

4. Branches and tags

Distributed version control?

Example scenarios at

https://git-scm.com/about/distributed

Open Source Project

Register here:

Thank you