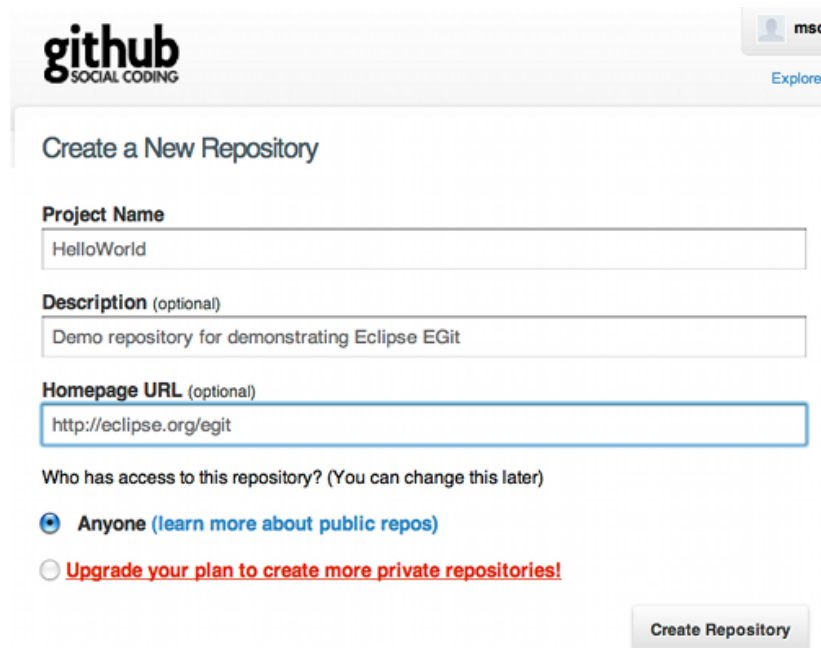


Subversion Demo Outline

- Create Repository
- Push initial code to repository
- Import code in local machines
- Change and Commit code
- Resolve Conflicts

Creating remote repository

- For remote repository – use GitHub
- Create an account on GitHub (<https://github.com/>)
 - All team members should create one
- Create a new repository on Github (search their tutorial)
 - Only one team member should create repository



The screenshot shows the GitHub 'Create a New Repository' page. At the top, the GitHub logo and 'SOCIAL CODING' tagline are on the left, and a user profile icon labeled 'mso' with an 'Explore' link is on the right. The main heading is 'Create a New Repository'. Below this, there are three input fields: 'Project Name' with the value 'HelloWorld', 'Description (optional)' with the value 'Demo repository for demonstrating Eclipse EGit', and 'Homepage URL (optional)' with the value 'http://eclipse.org/egit'. Below the input fields, a question asks 'Who has access to this repository? (You can change this later)'. There are two radio button options: 'Anyone (learn more about public repos)' which is selected, and 'Upgrade your plan to create more private repositories!'. At the bottom right, there is a 'Create Repository' button.

github
SOCIAL CODING

mso
Explore

Create a New Repository

Project Name
HelloWorld

Description (optional)
Demo repository for demonstrating Eclipse EGit

Homepage URL (optional)
http://eclipse.org/egit

Who has access to this repository? (You can change this later)

☒ Anyone (learn more about public repos)

☐ Upgrade your plan to create more private repositories!

Create Repository

Creating remote repository (cont.)

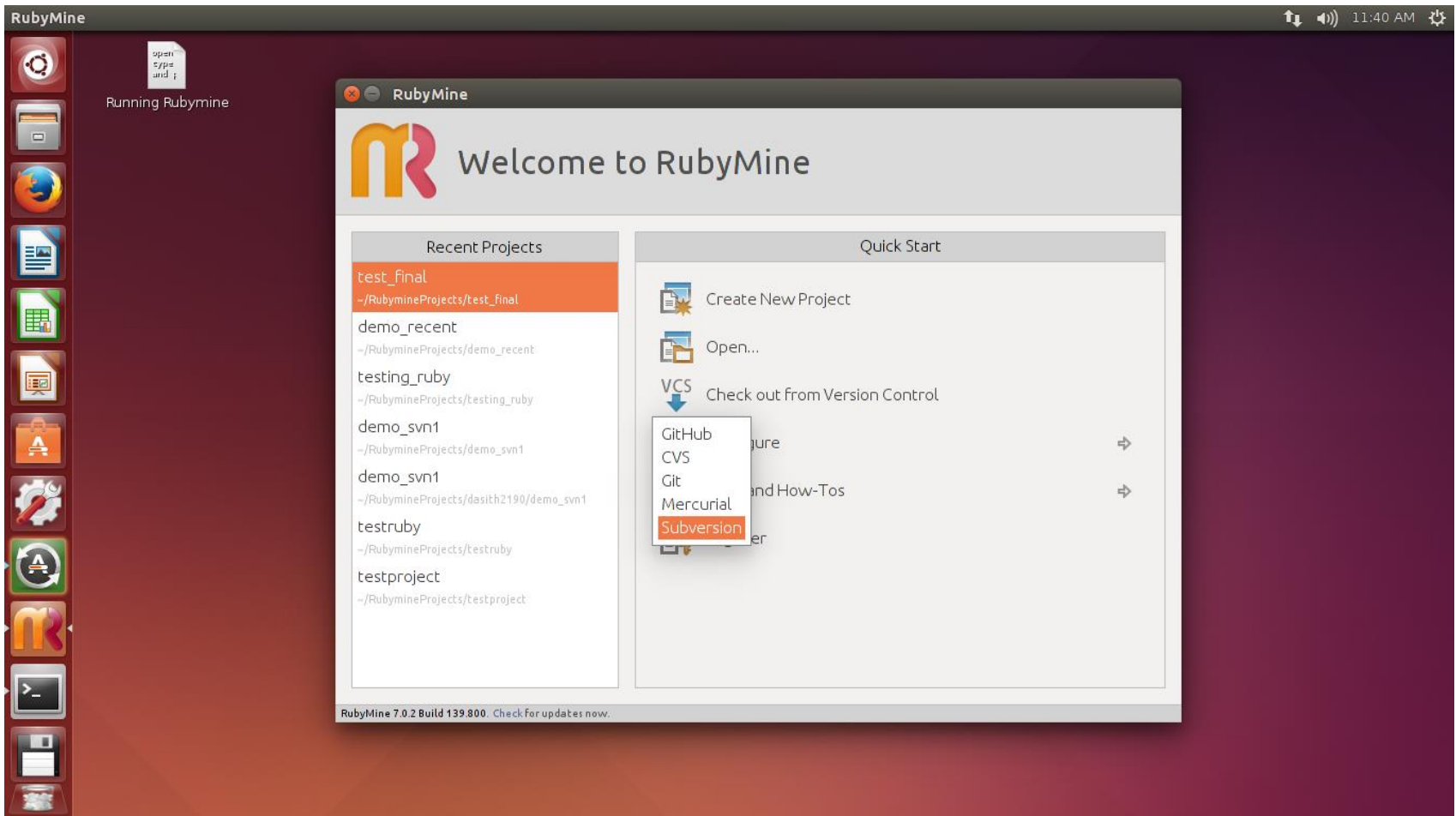
- **Select the option “initialize new repository with README”**
- **Also add .gitignore and pick the option Rails**
 - .gitignore file defines the files that are not submitted in the repository when you commit
 - In Rails – this includes log files, database, and several others
 - You can open the newly created .gitignore to view them

Push the initial Code

- We are going to use your course VM (ubuntu)
- Check if subversion client is installed
 - All team members will do this step
 - Go to terminal and enter `svn --help`
 - If it gives options, means, svn is already installed
 - **if svn is not installed, open a terminal and enter “sudo apt-get install subversion”**
- Only one team member who created the repository will do the following steps.
- Open Rubymine IDE
- Go to File menu and click close project

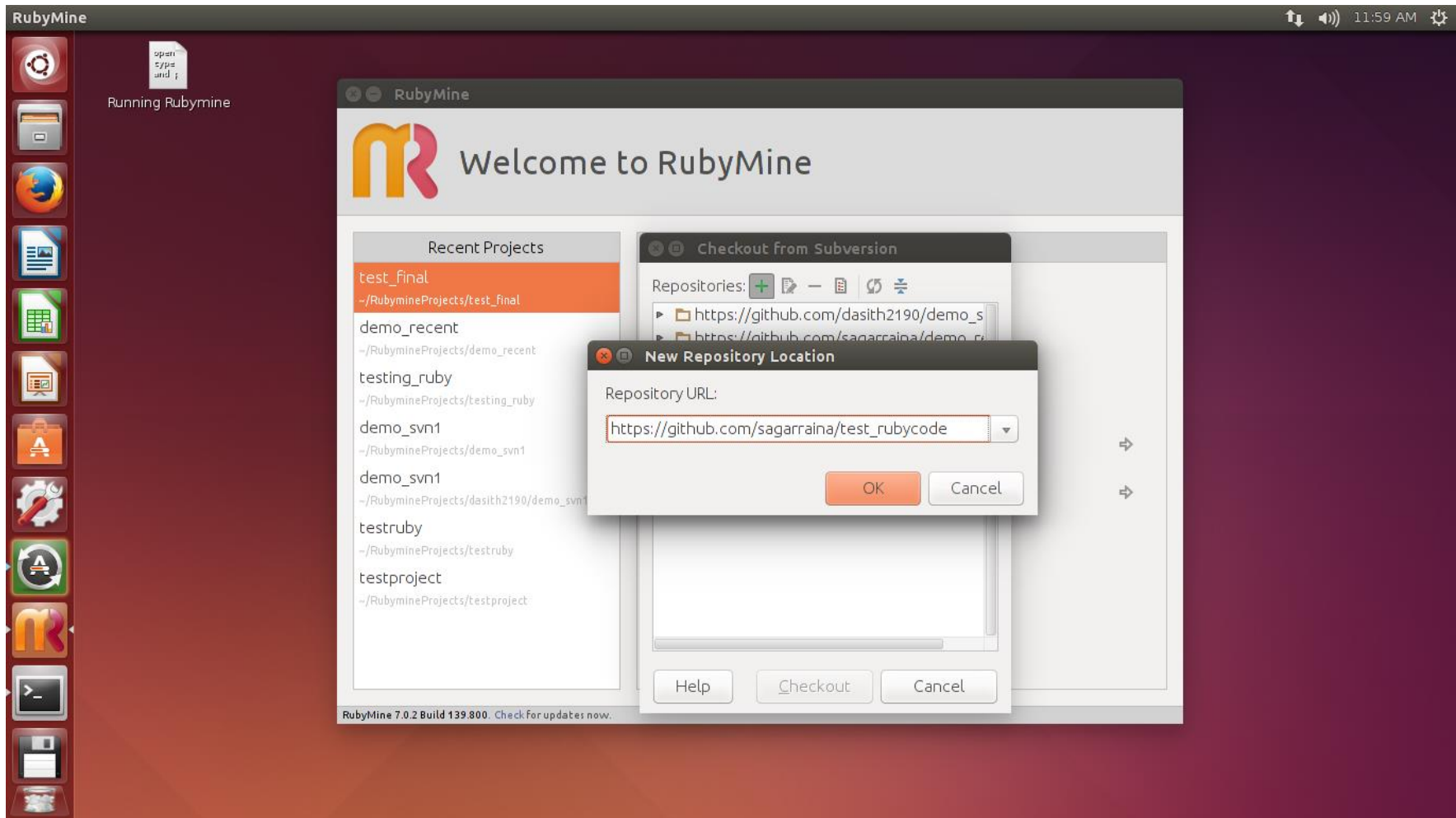
Pushing the initial code

- Click Check out from version control -> choose subversion



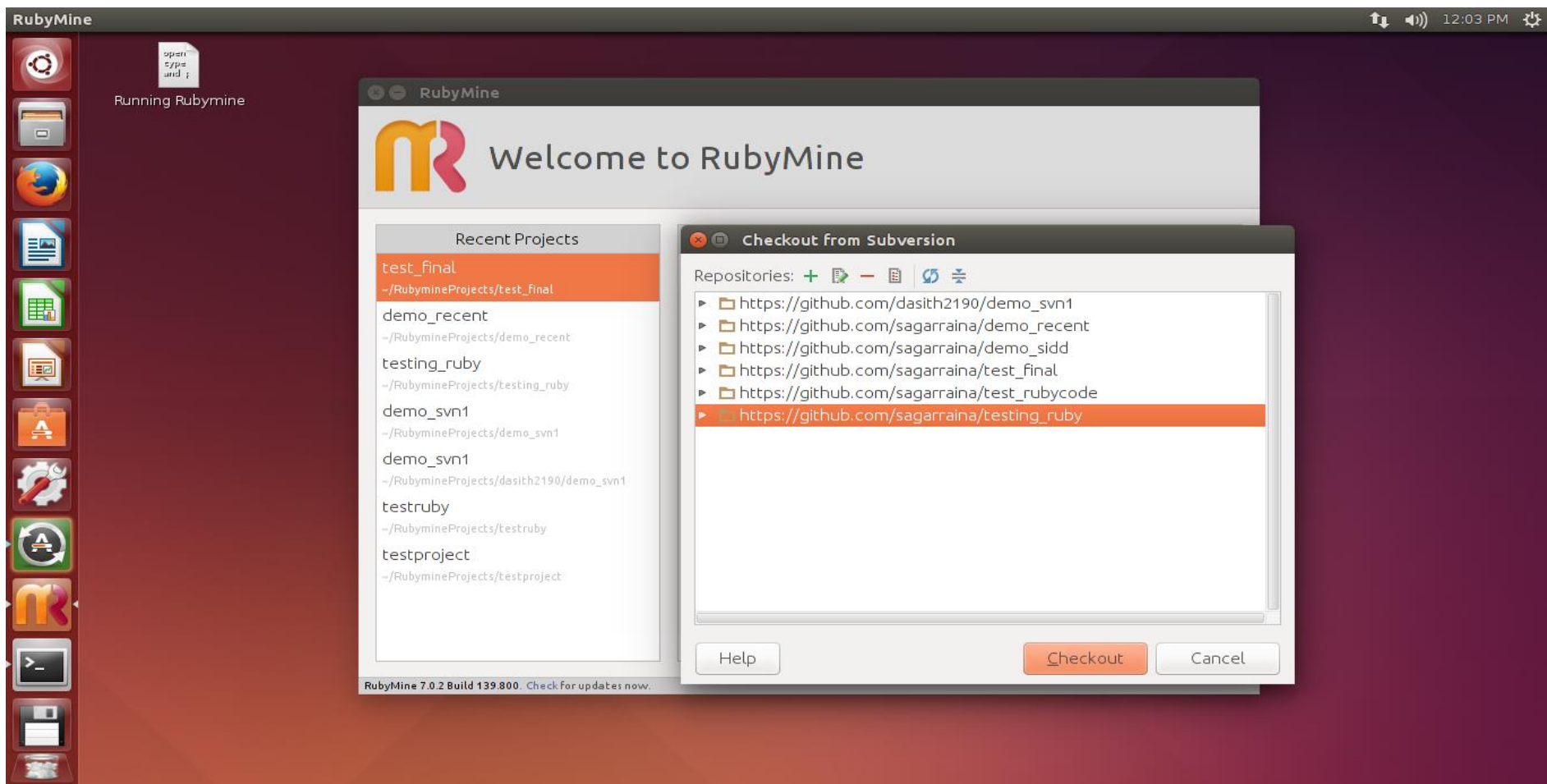
Pushing the initial code

- Click (+) button to add the repository url
- Go back to github.com and get the repository url



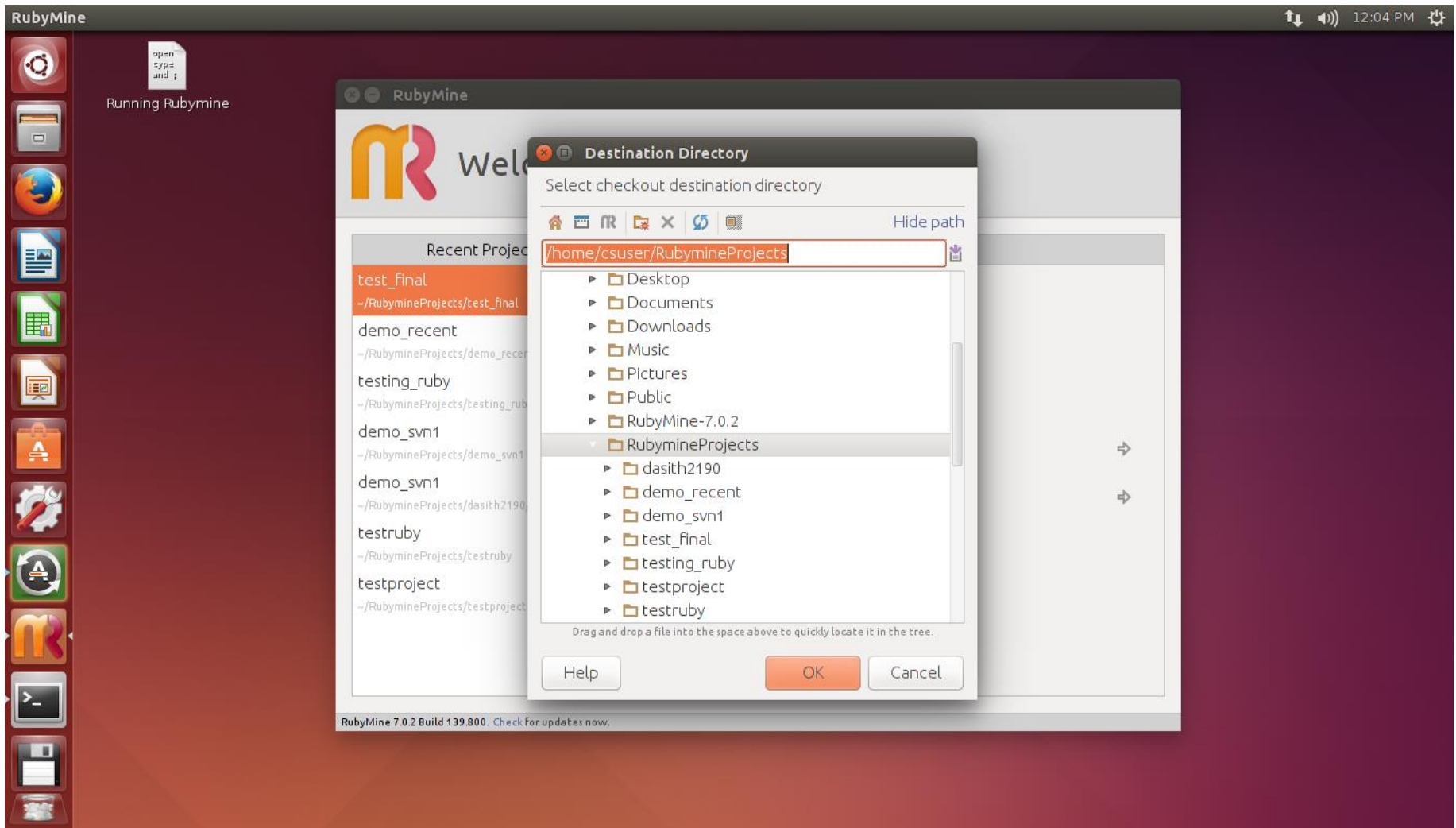
Pushing the initial code

- Select the recently added url and click checkout



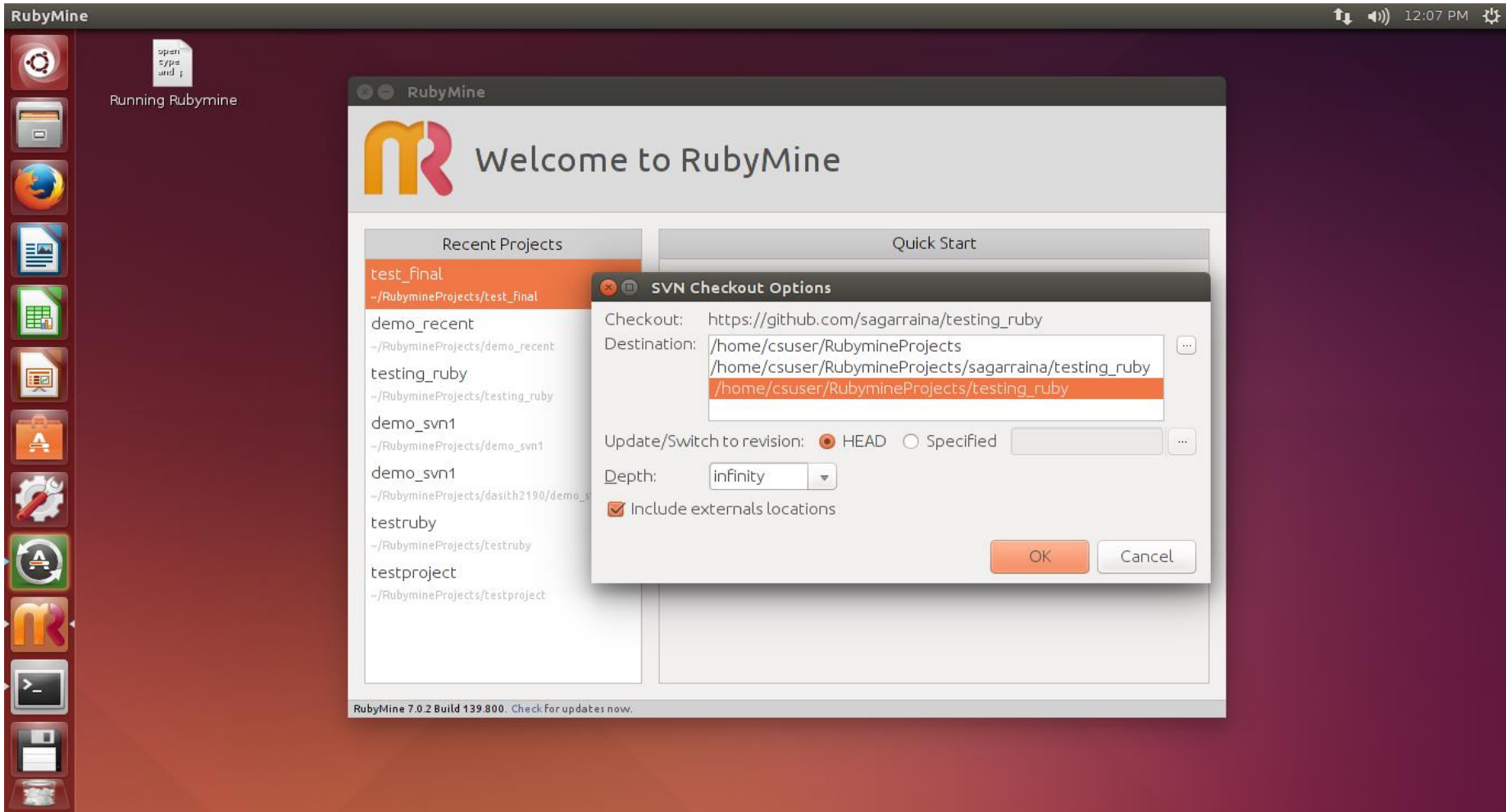
Pushing the initial code

- Select the destination directory and click OK



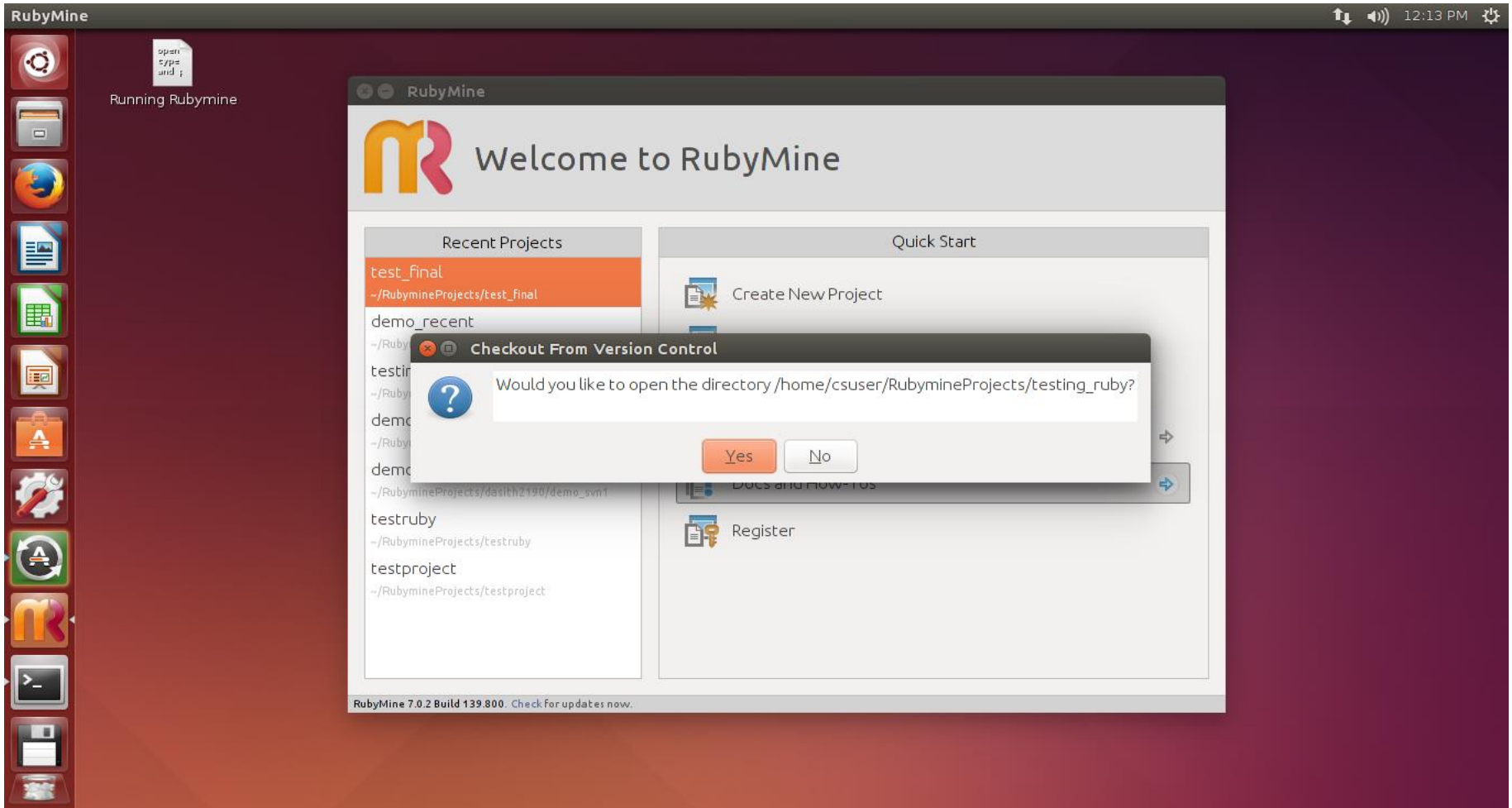
Pushing the initial code

- Select destination folder to local working copy



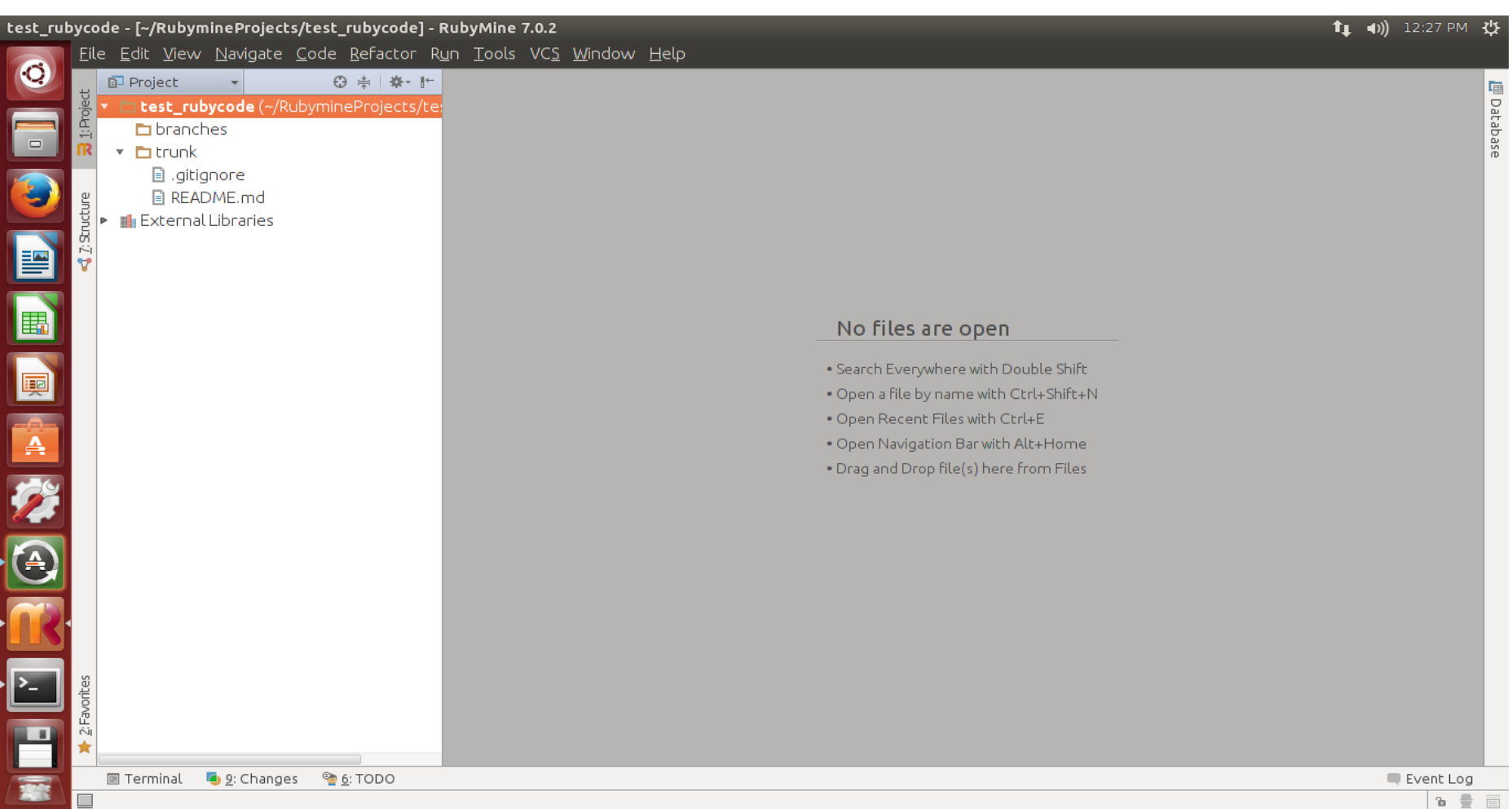
Pushing the initial code

- Click OK



Pushing the initial Code

- You will see the project on left checkout from the repo.



Pushing the initial Code

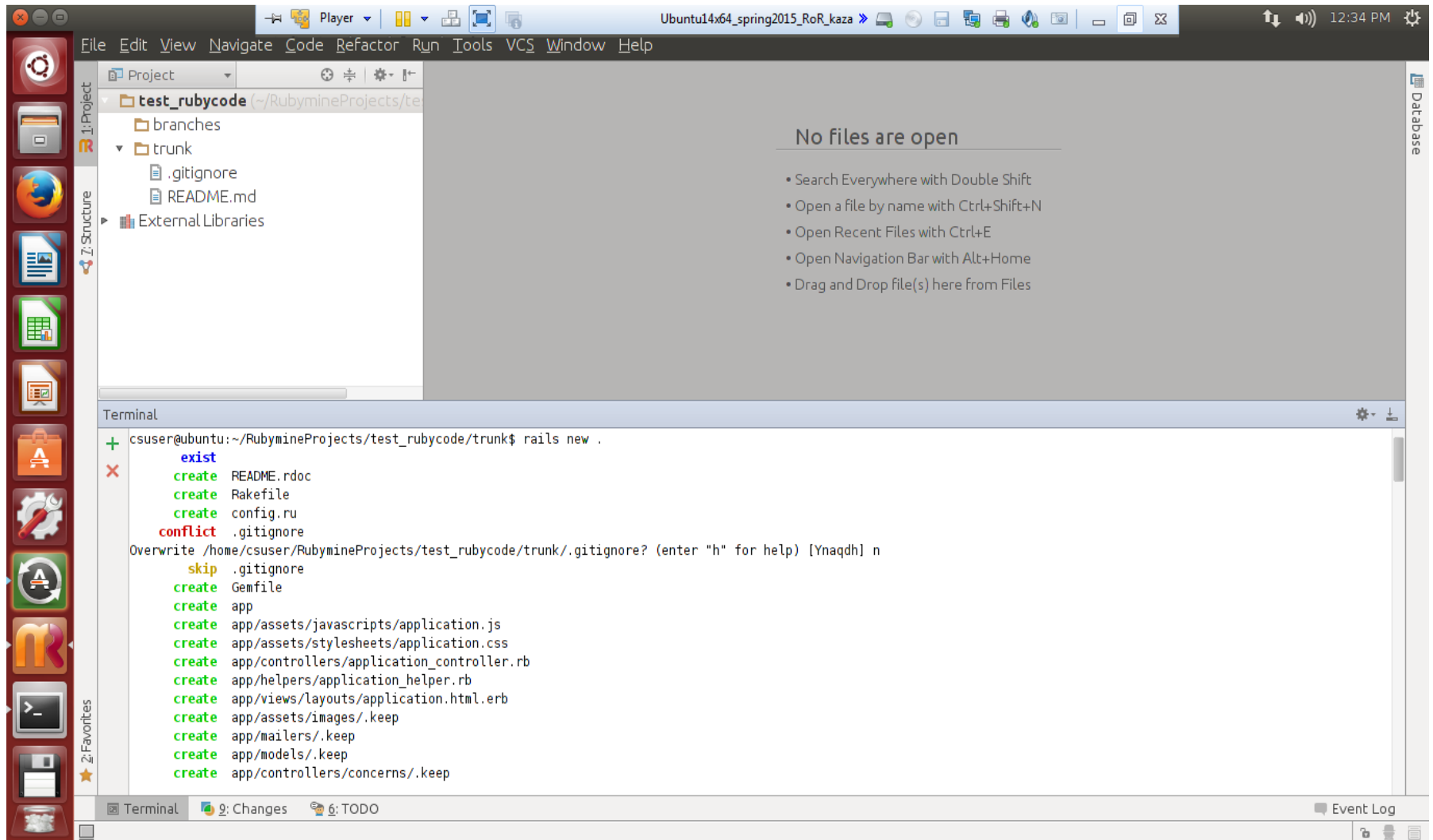
- In Rubymine terminal, enter the following commands:

- `cd trunk`
- `rails new .`

Entering the command “rails new .” – which creates a new rails project in folder. **Don't overwrite .gitignore when prompted.** Make sure your skeleton project is running

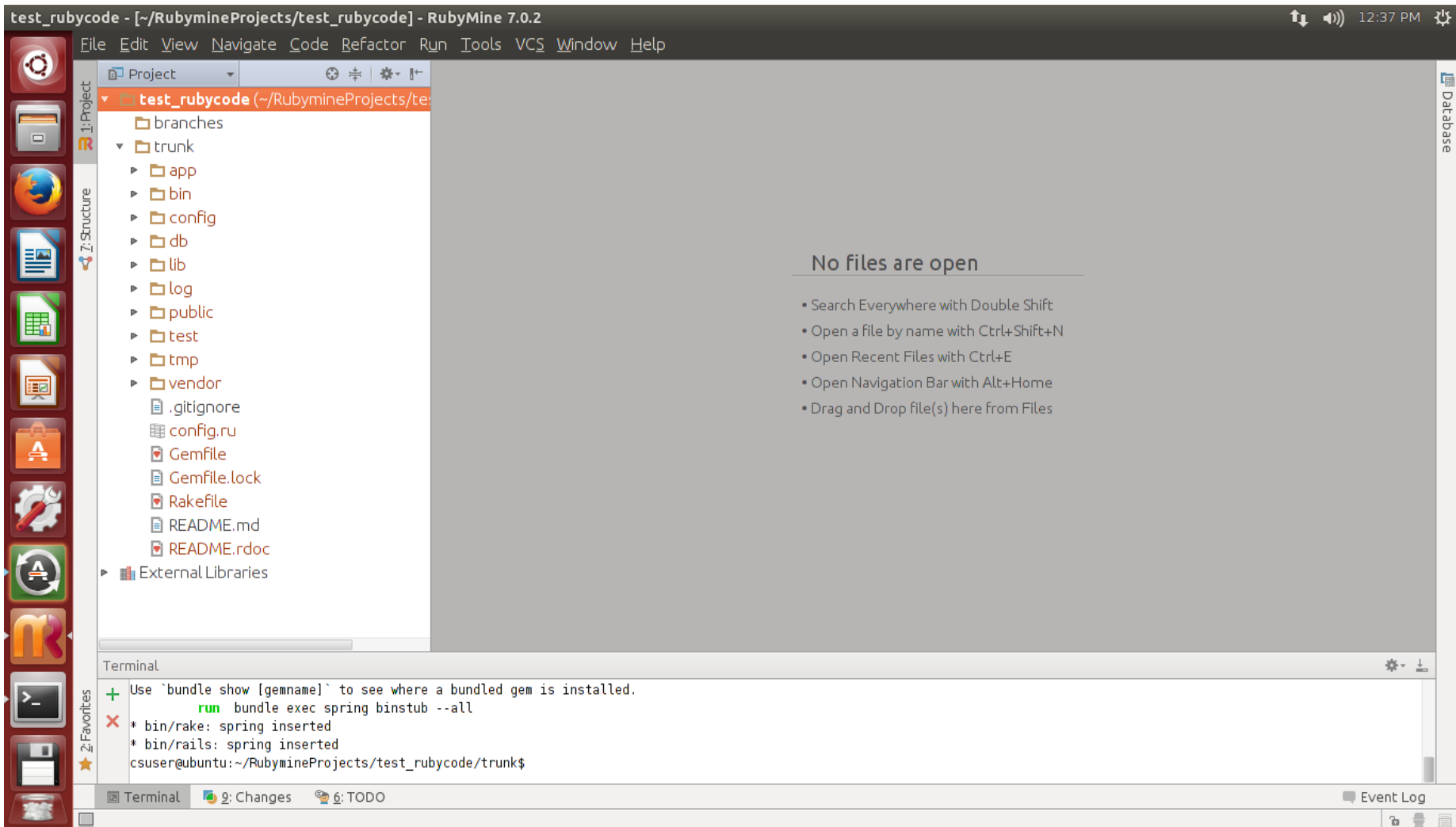
Pushing the initial Code

- New rails app gets created in the trunk folder, see below.



Pushing the initial Code

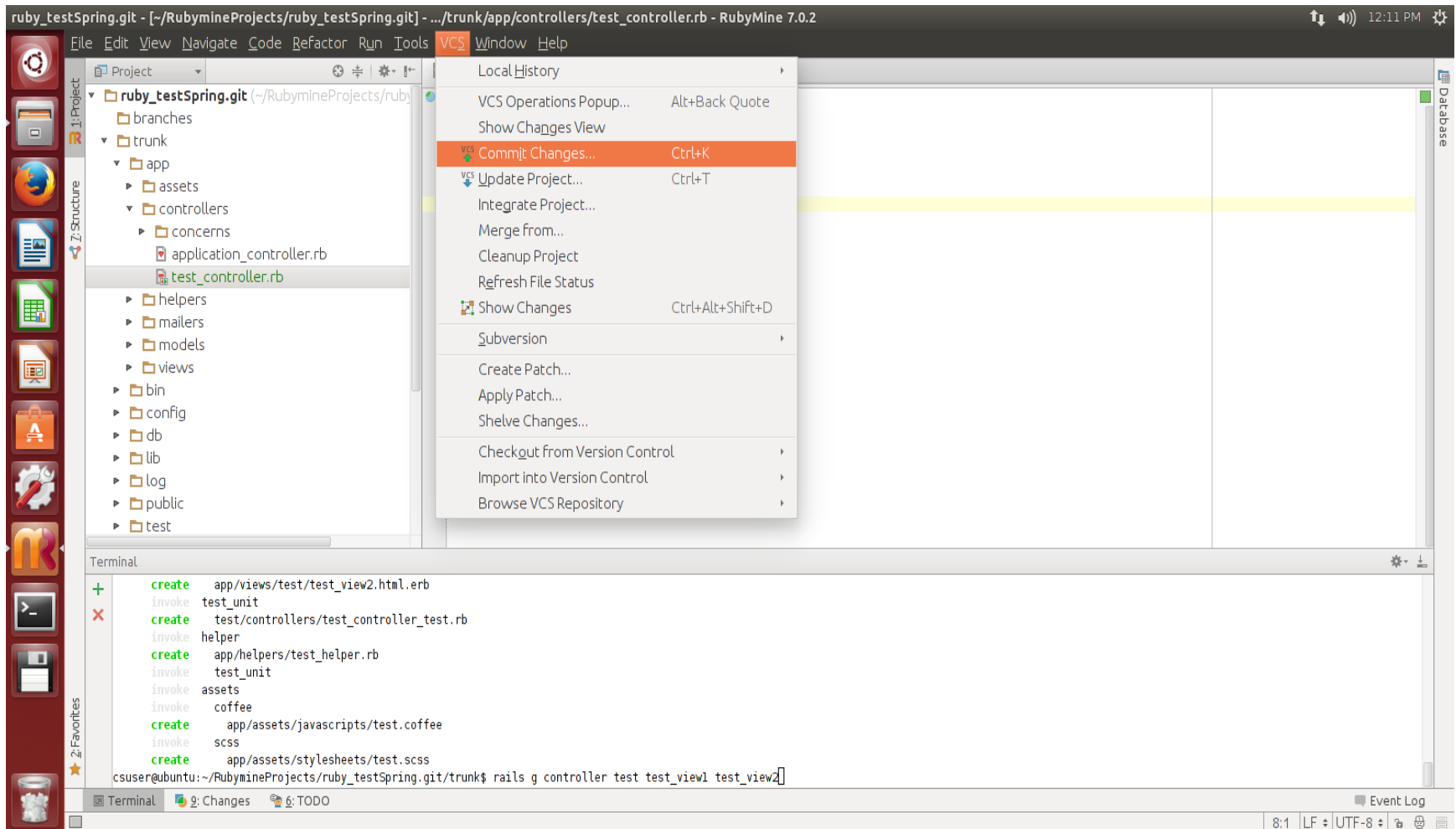
- See the rails app project sub folders on the left.



Pushing the initial Code

- Use the following command to add the project to queue for commit.
 - `svn add *`
- Next, we have to exclude the tmp folder from queue. To do this, enter the following command.
 - `svn remove tmp`
- Use the following command to push the project files to remote repository.
 - `svn commit -m "pushing my first changes"`

Commit Changes



Pushing the initial code

- In github repository you will see the recently committed code

The screenshot shows a web browser window displaying the GitHub repository page for 'sagaraina/test_rubycode'. The browser's address bar shows the URL 'https://github.com/sagaraina/test_rubycode'. The repository page shows the 'master' branch selected, with a commit history table listing files and their commit times. The files listed include 'app', 'bin', 'config', 'db', 'lib', 'log', 'public', 'test', 'vendor/assets', '.gitignore', 'Gemfile', 'Gemfile.lock', 'README.md', 'README.rdoc', and 'Rakefile'. The commit history shows that most files were committed '3 minutes ago' by 'sagaraina' with the message 'pushing my first changes'. The 'Initial commit' for '.gitignore', 'README.md', and 'Rakefile' was made 'an hour ago'. The right sidebar shows options for 'Pull requests', 'Wiki', 'Pulse', 'Graphs', 'Settings', and 'Subversion checkout URL'.

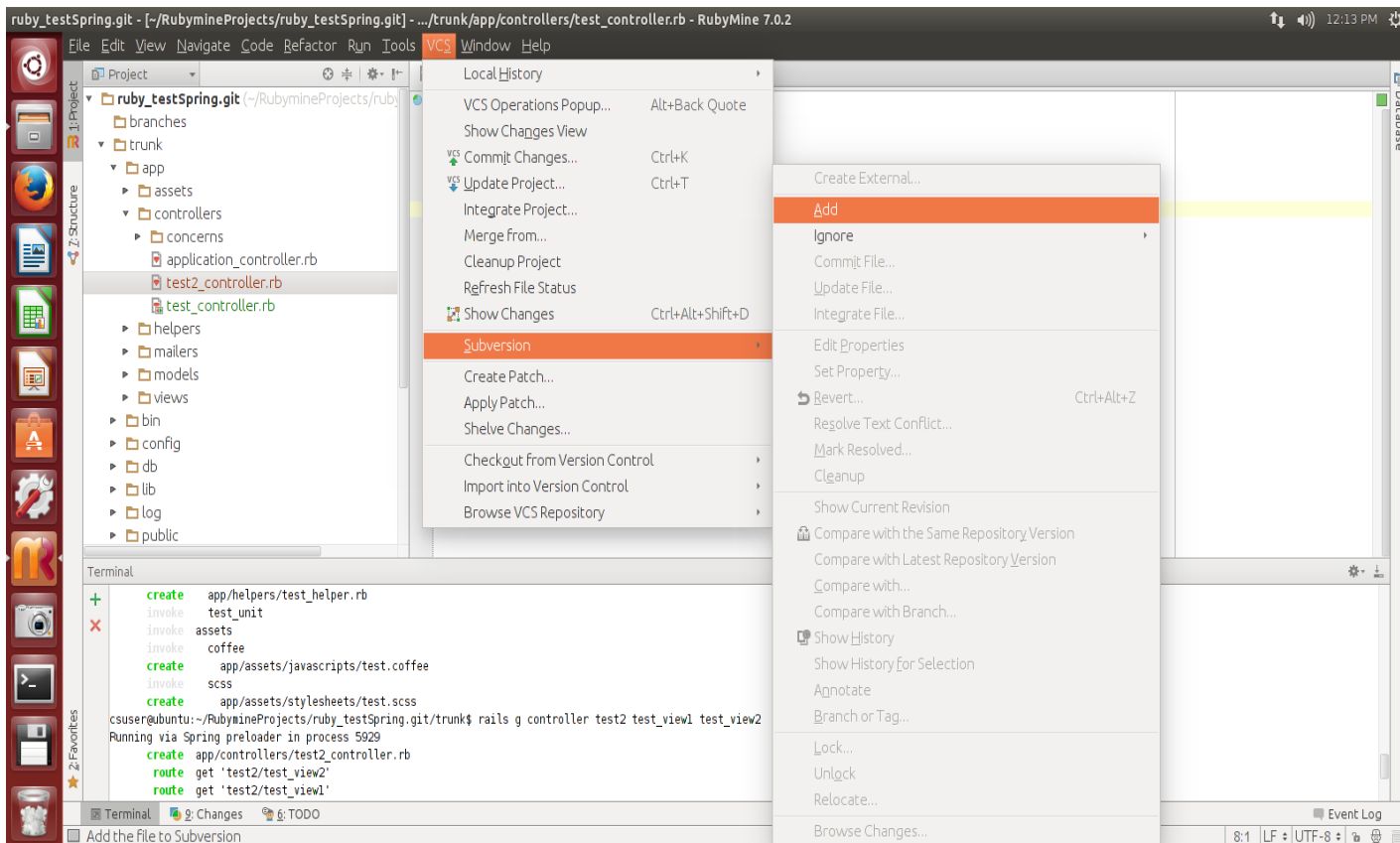
File	Commit Message	Commit Time
app	pushing my first changes	3 minutes ago
bin	pushing my first changes	3 minutes ago
config	pushing my first changes	3 minutes ago
db	pushing my first changes	3 minutes ago
lib	pushing my first changes	3 minutes ago
log	pushing my first changes	3 minutes ago
public	pushing my first changes	3 minutes ago
test	pushing my first changes	3 minutes ago
vendor/assets	pushing my first changes	3 minutes ago
.gitignore	Initial commit	an hour ago
Gemfile	pushing my first changes	3 minutes ago
Gemfile.lock	pushing my first changes	3 minutes ago
README.md	Initial commit	an hour ago
README.rdoc	pushing my first changes	3 minutes ago
Rakefile	pushing my first changes	3 minutes ago

Getting the code from Repo by other team members

- The team member who created the repository needs to give other team members permission to checkout the code from the subversion.
 - For The team member who created the repo:
 - Go to the repository in github.com
 - Click settings on the right side column
 - Click collaborators under options on the left side
 - Then search for the your team mates by username or emails they used used for github.
 - click add collaborator.
 - For the rest of the team members:
 - Go to your email to accept the collaboration
 - Browse to the repository page on GitHub and copy the HTTPS clone URL:
 - Checkout again (go through slide 5 – 11)
 - change working directory: cd trunk
 - bundle install

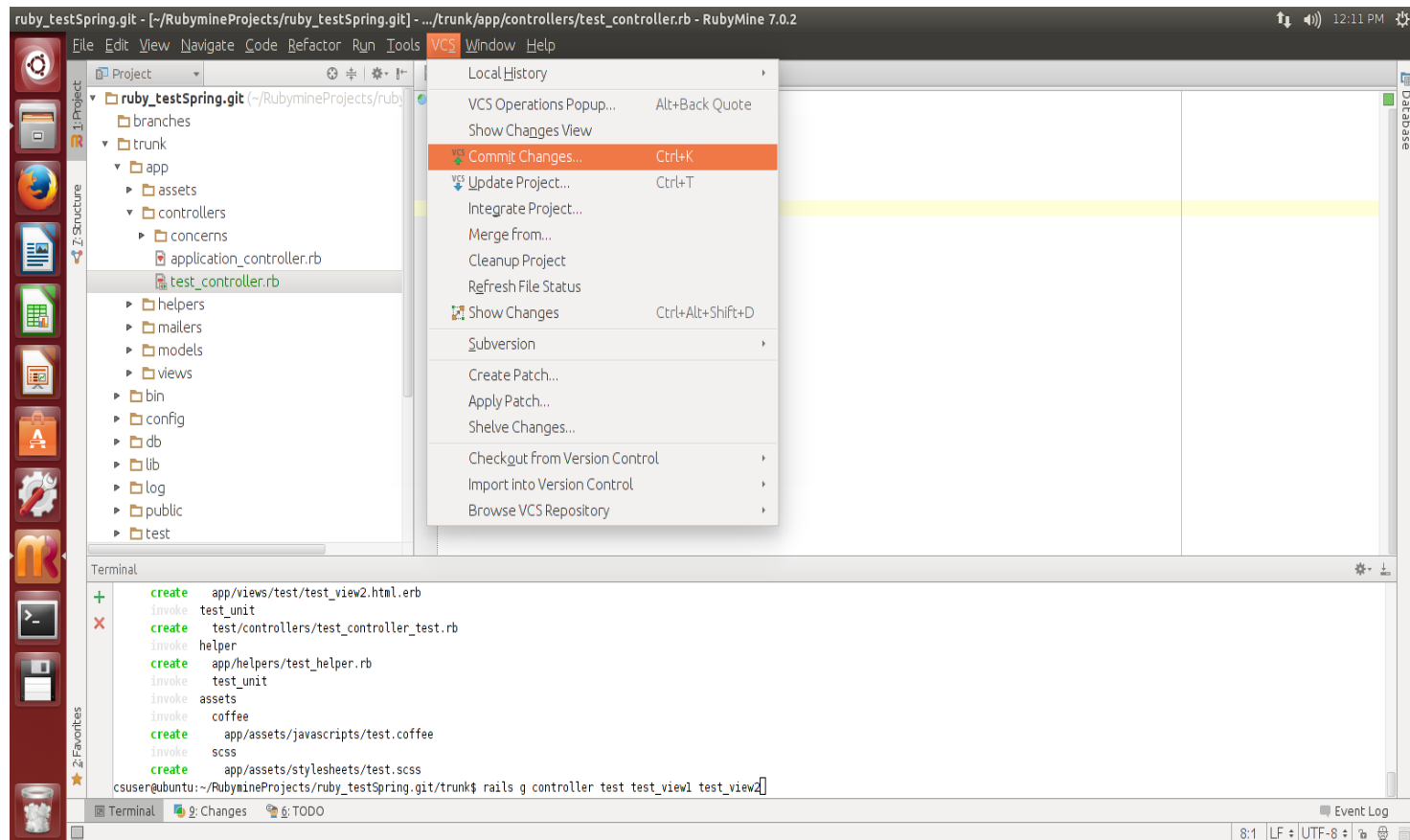
Adding New Files to Version Control

- If you create new files, you need to add it to version control.
 - Select file you want to add to version control -> VCS-> Subversion -> Add



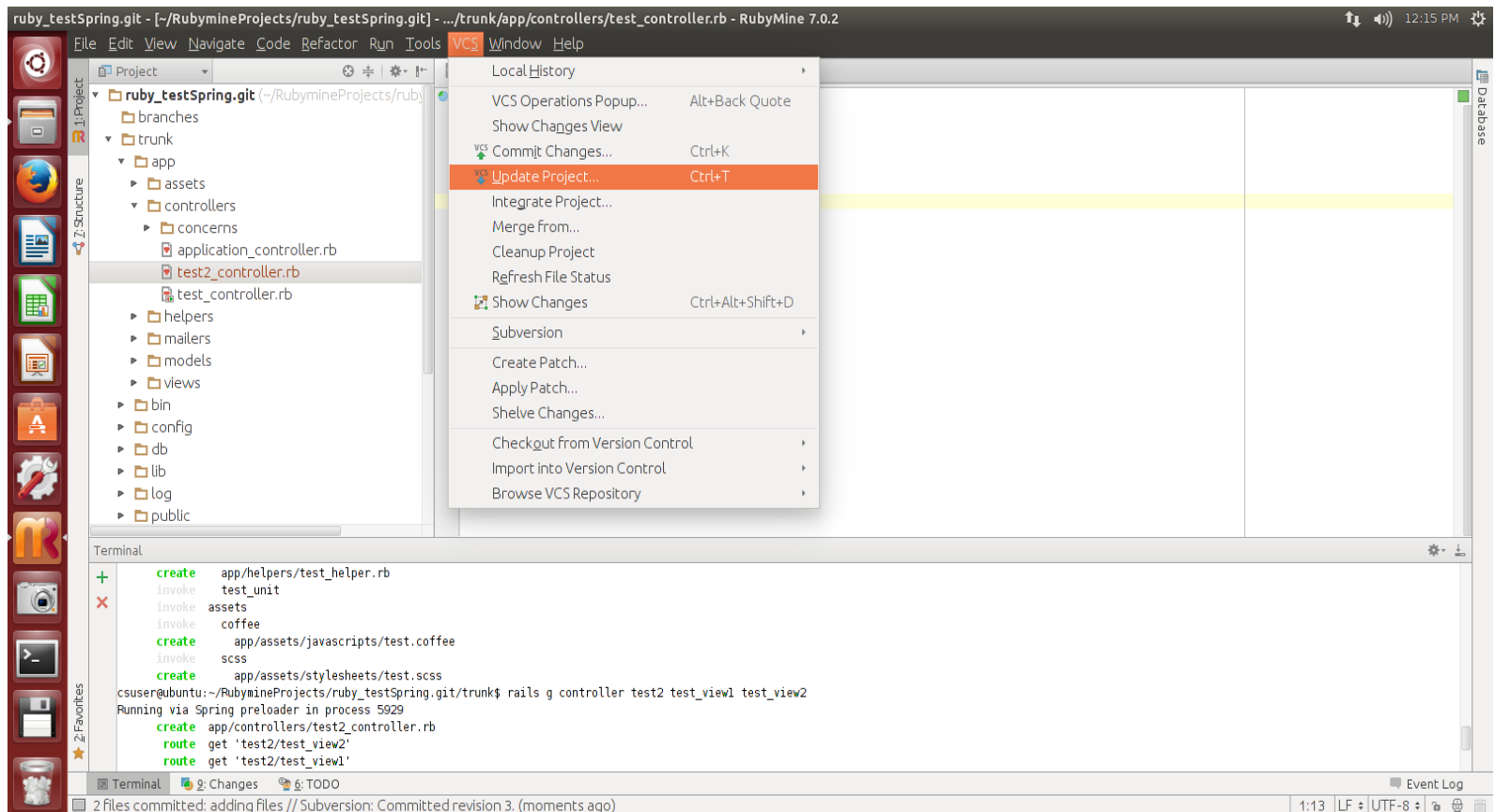
Adding New Files to Version Control

- After “Add”, commit changes to Github Repo: VCS-> Commit Changes



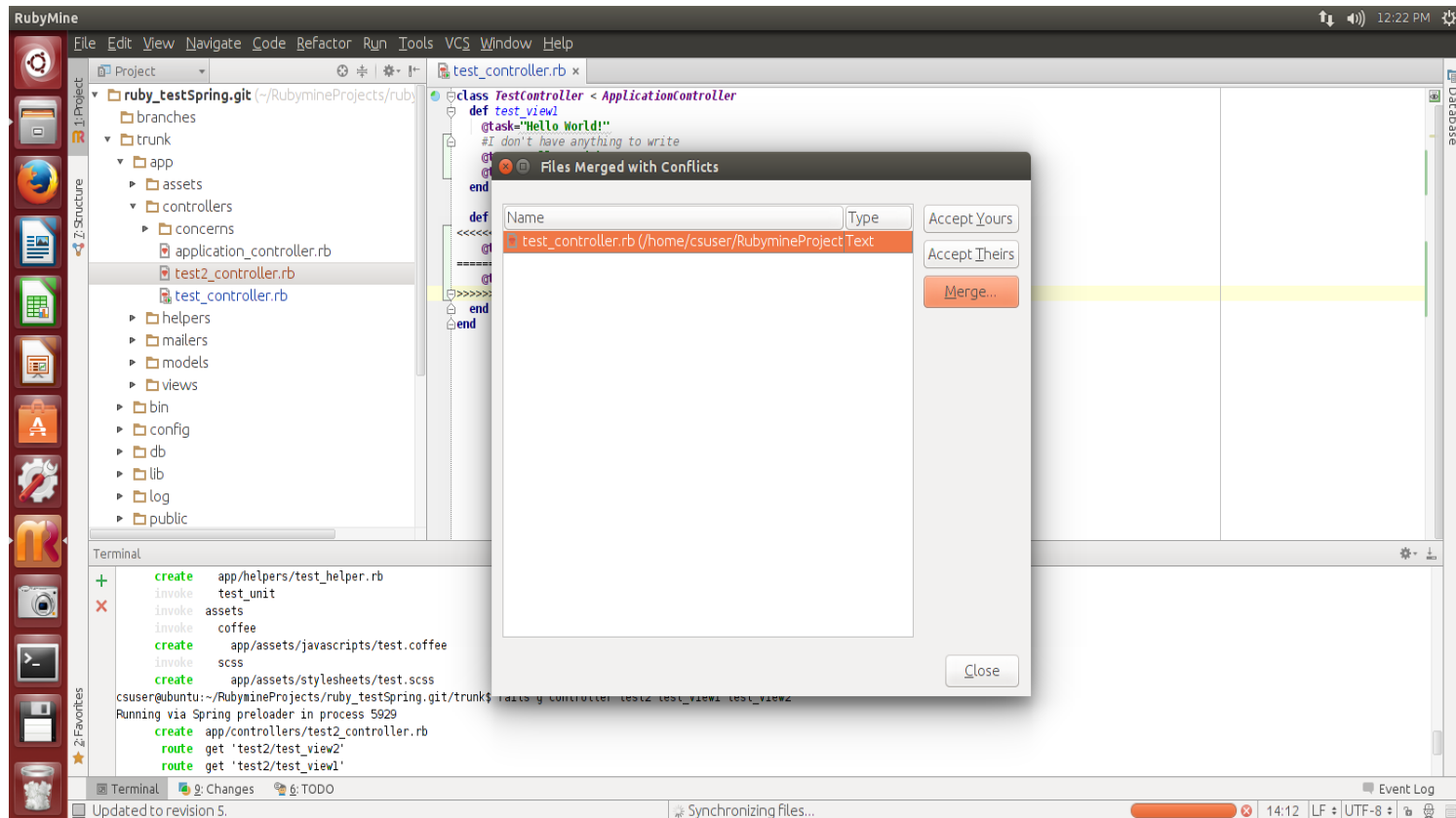
Updating the Project

- Other members should update their project to get the latest working copy in Git: VCS-> Update Project



Resolving Conflicts

- If two or more people have edited the same file, you may see a conflict when you Update Project. The following window will pop up if there is a conflict.

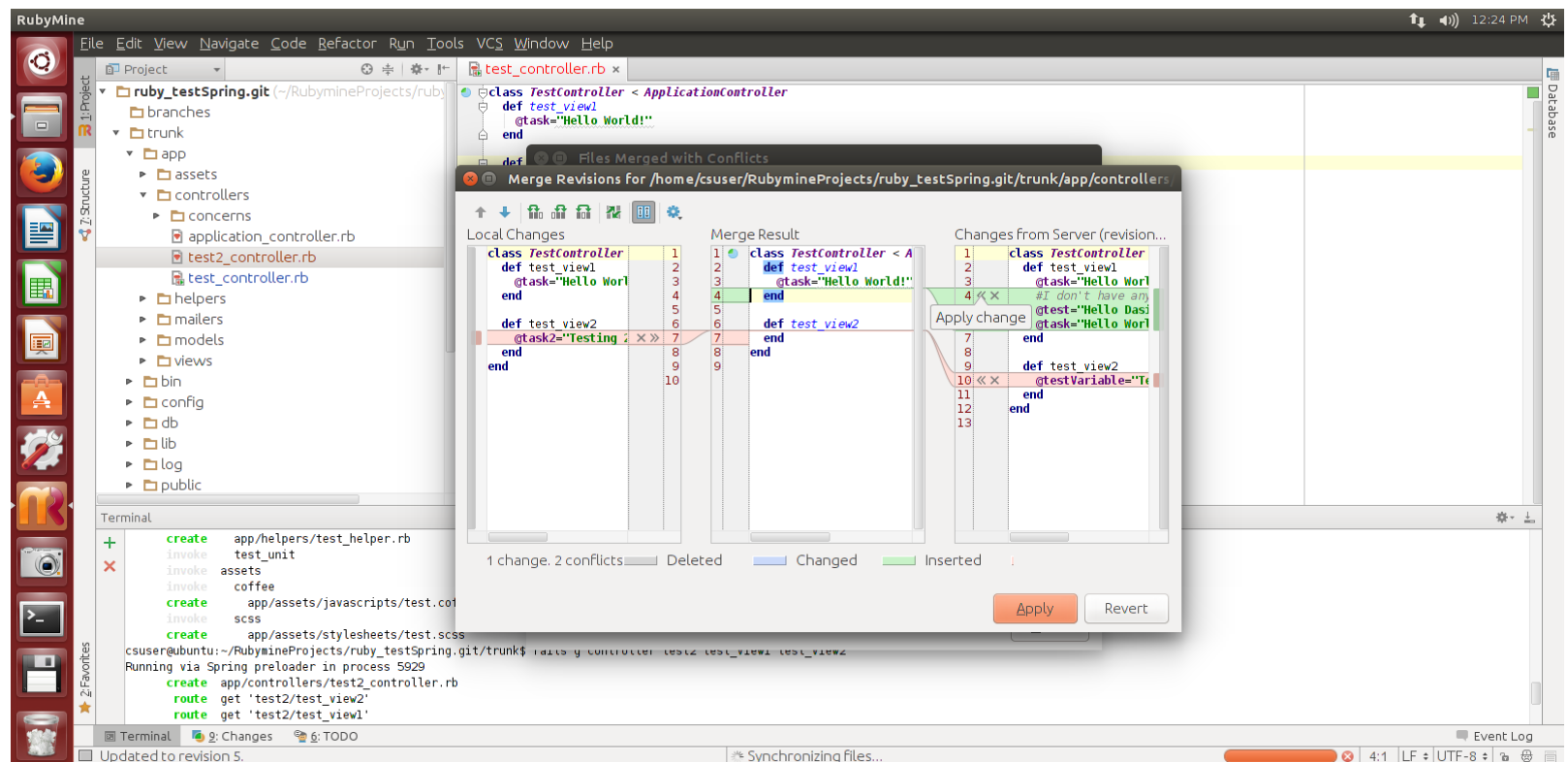


Resolving Conflicts contd.

- To resolve conflict, you can do the following
 - Accept Yours- accept your change
 - Accept Theirs- accept the change in Git
 - Merge- Specify manually what changes need to be made

Resolving Conflicts contd.

- If you select the option to Merge, the following window will pop up
- You can use the pop window to select the changes you want to make.
 - Clicking the arrow will add the selected change.
 - Clicking x will get rid of the conflict.



External Resources

- Guide to checkout an existing project in Github:

<https://help.github.com/articles/support-for-subversion-clients/>