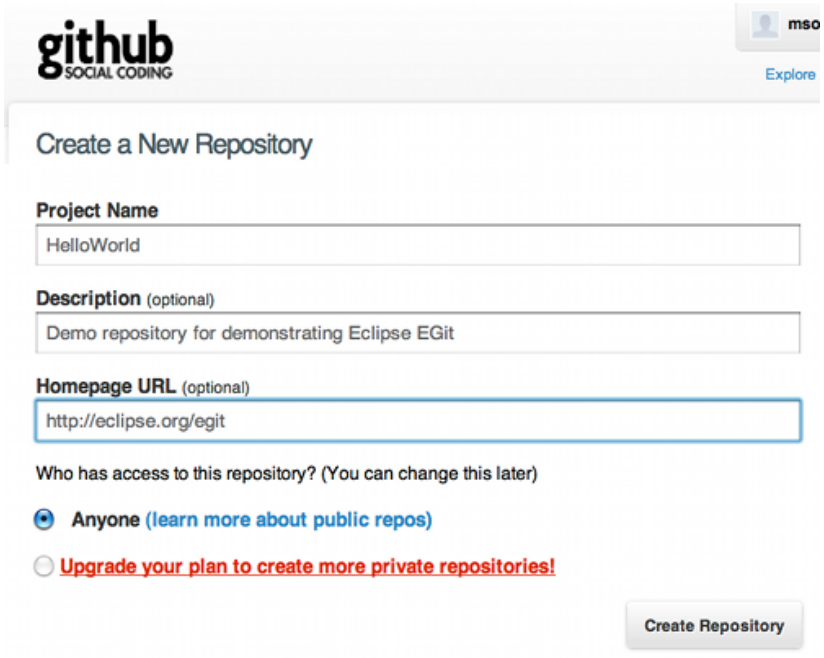
Subversion Demo

- 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



Creating remote repository (cont.)

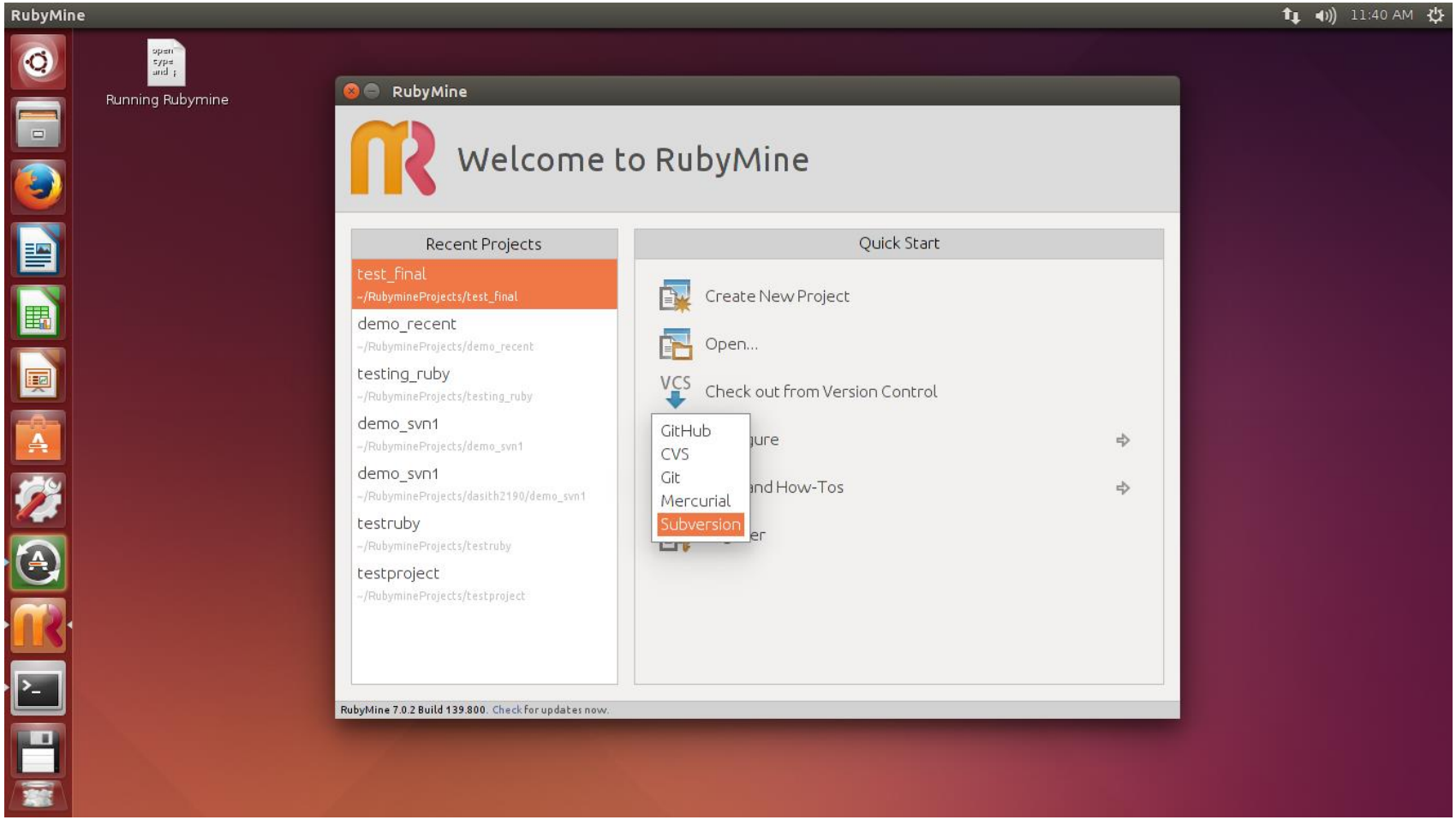
- **Select the option "initialize new repository with README"**
- **Also add .gitignore and pick the option Rails**
  - .gitignore file defines the files that are not submitted in the repository when you commit
  - In Rails – this includes log files, database, and several others
    - You can open the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

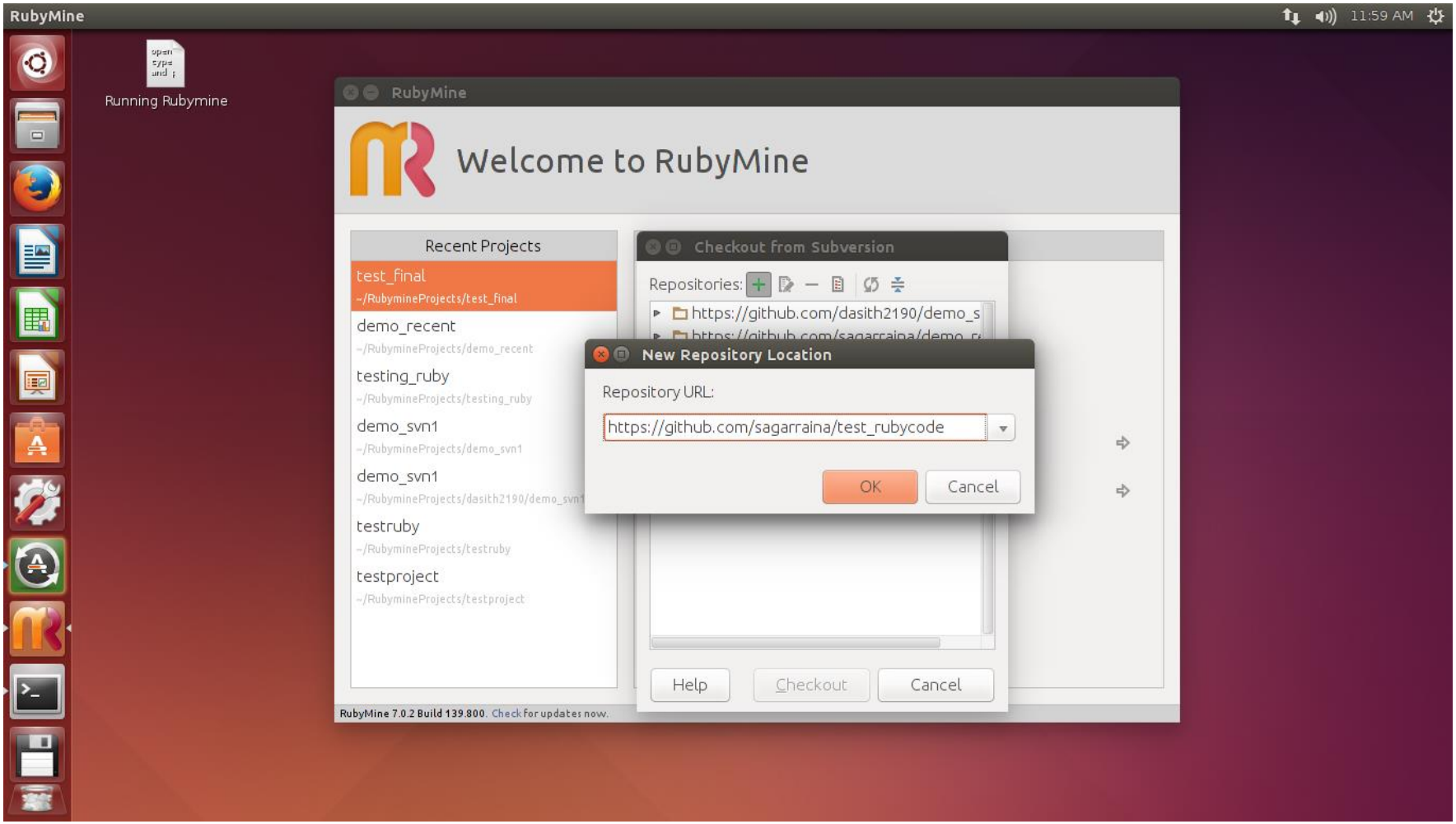
Pushing the initial code

- Click Check out from version control -> choose subversion



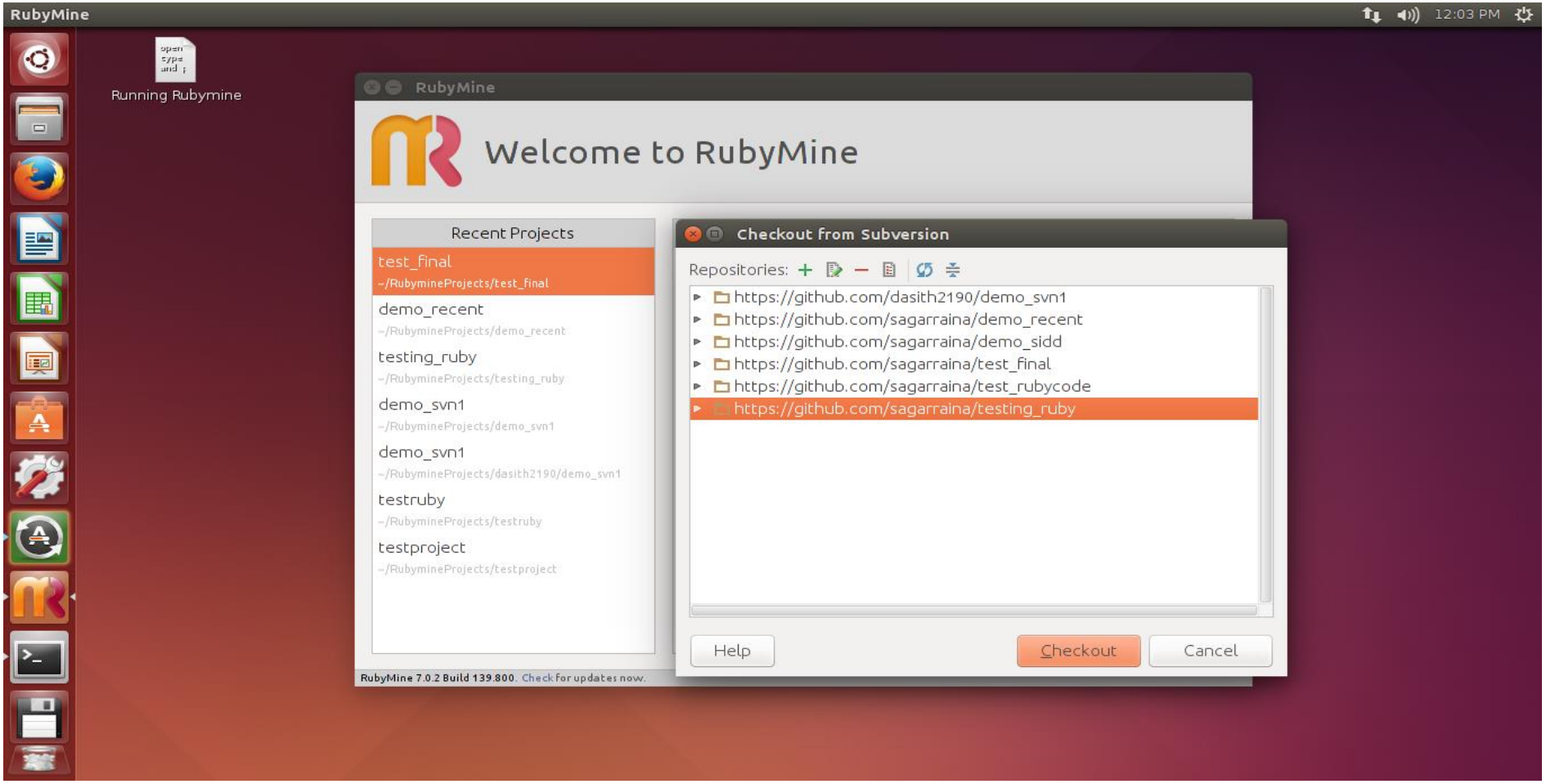
Pushing the initial code

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



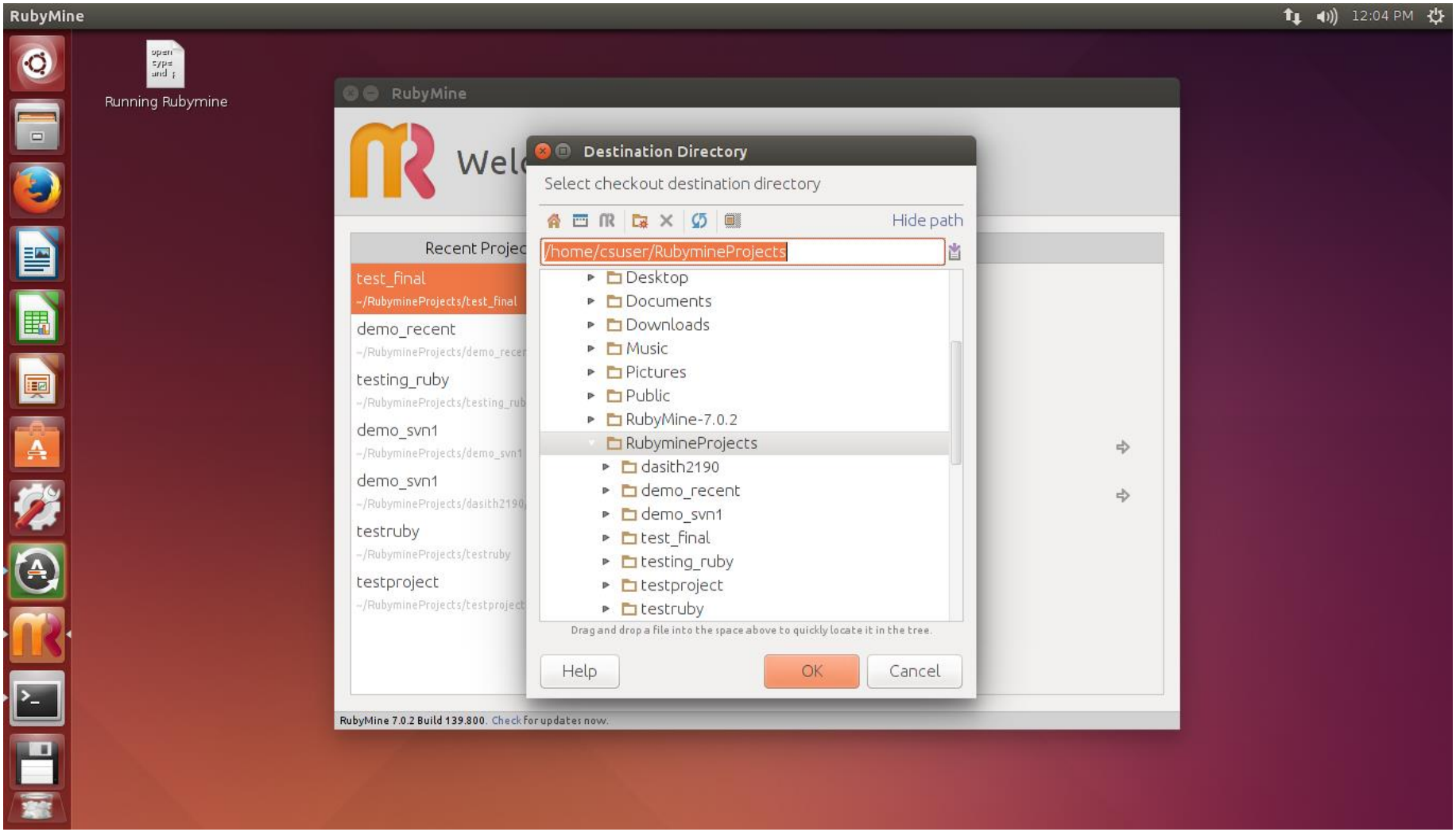
Pushing the initial code

- Select the recently added url and click checkout



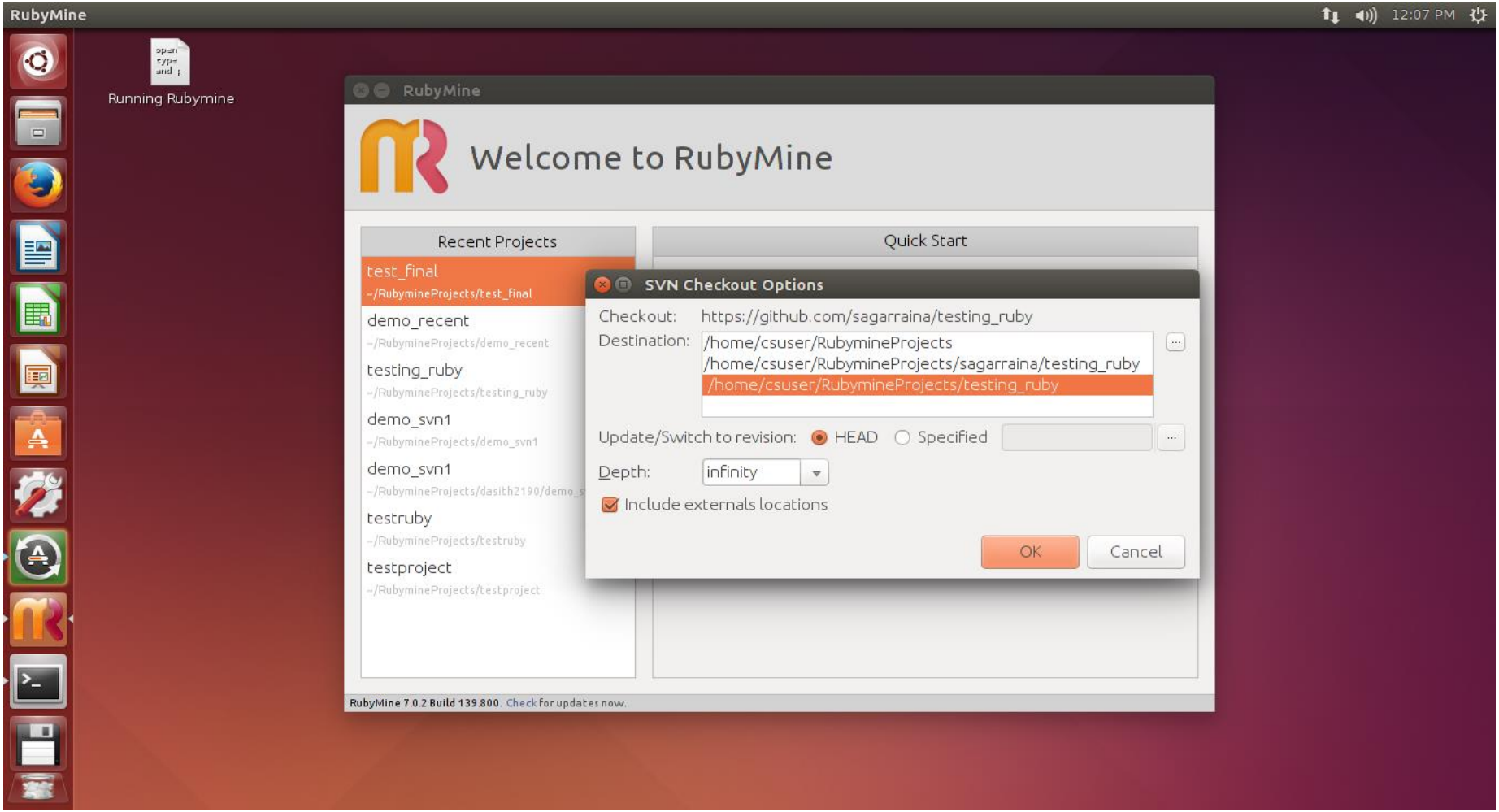
Pushing the initial code

- Select the destination directory and click OK



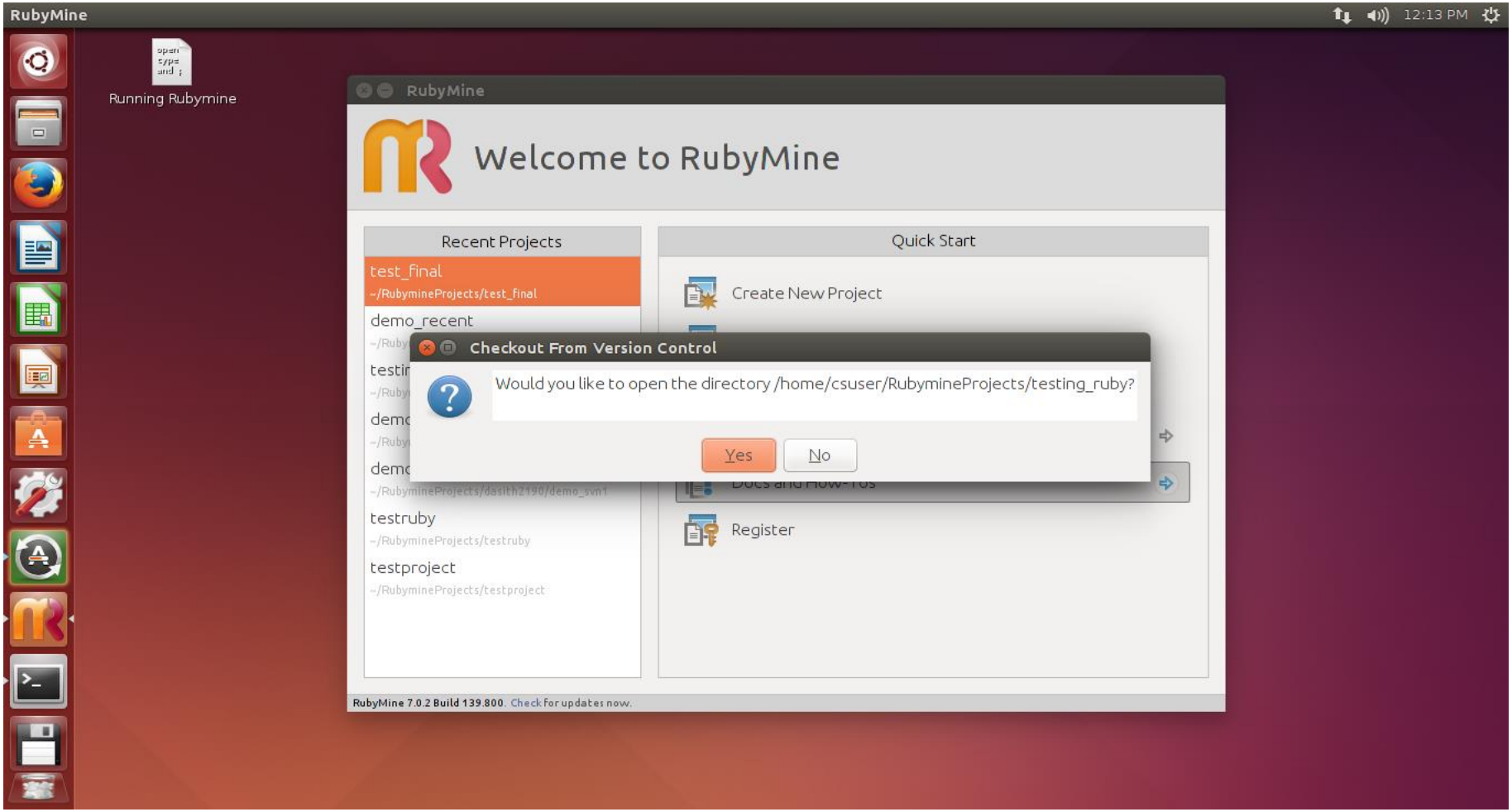
Pushing the initial code

- Select destination folder to local working copy



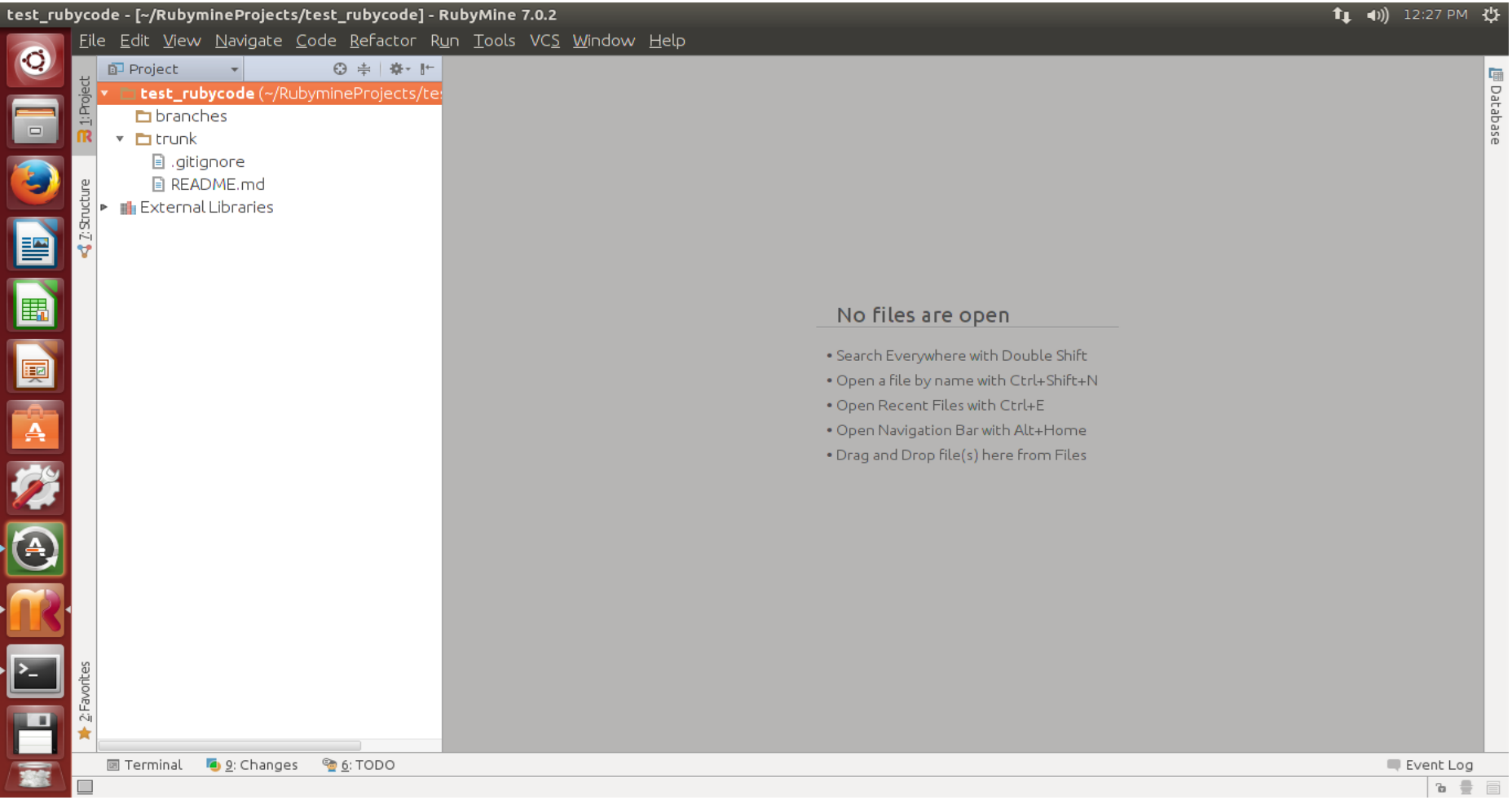
Pushing the initial code

- Click OK



Pushing the initial Code

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



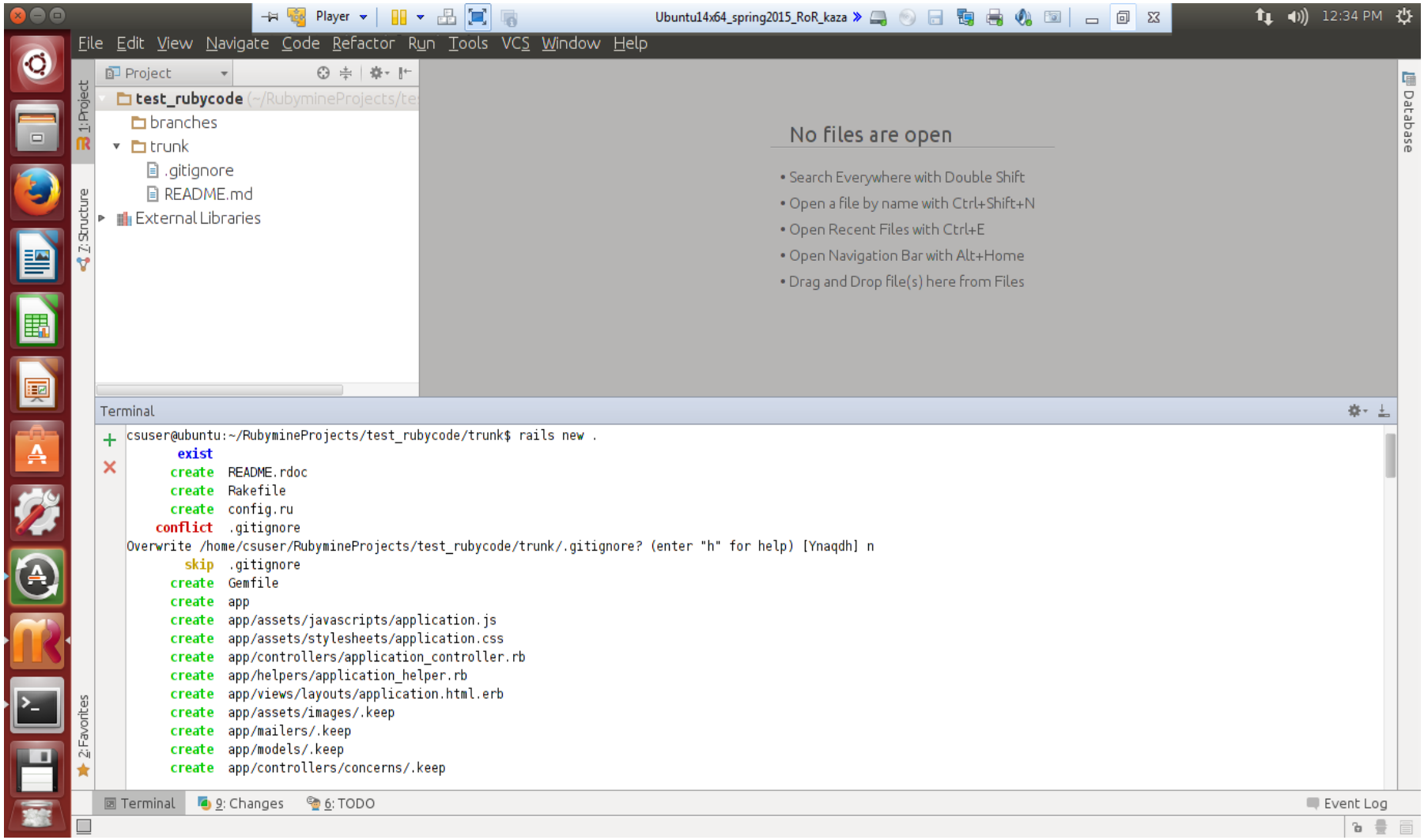
Pushing the initial Code

- 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

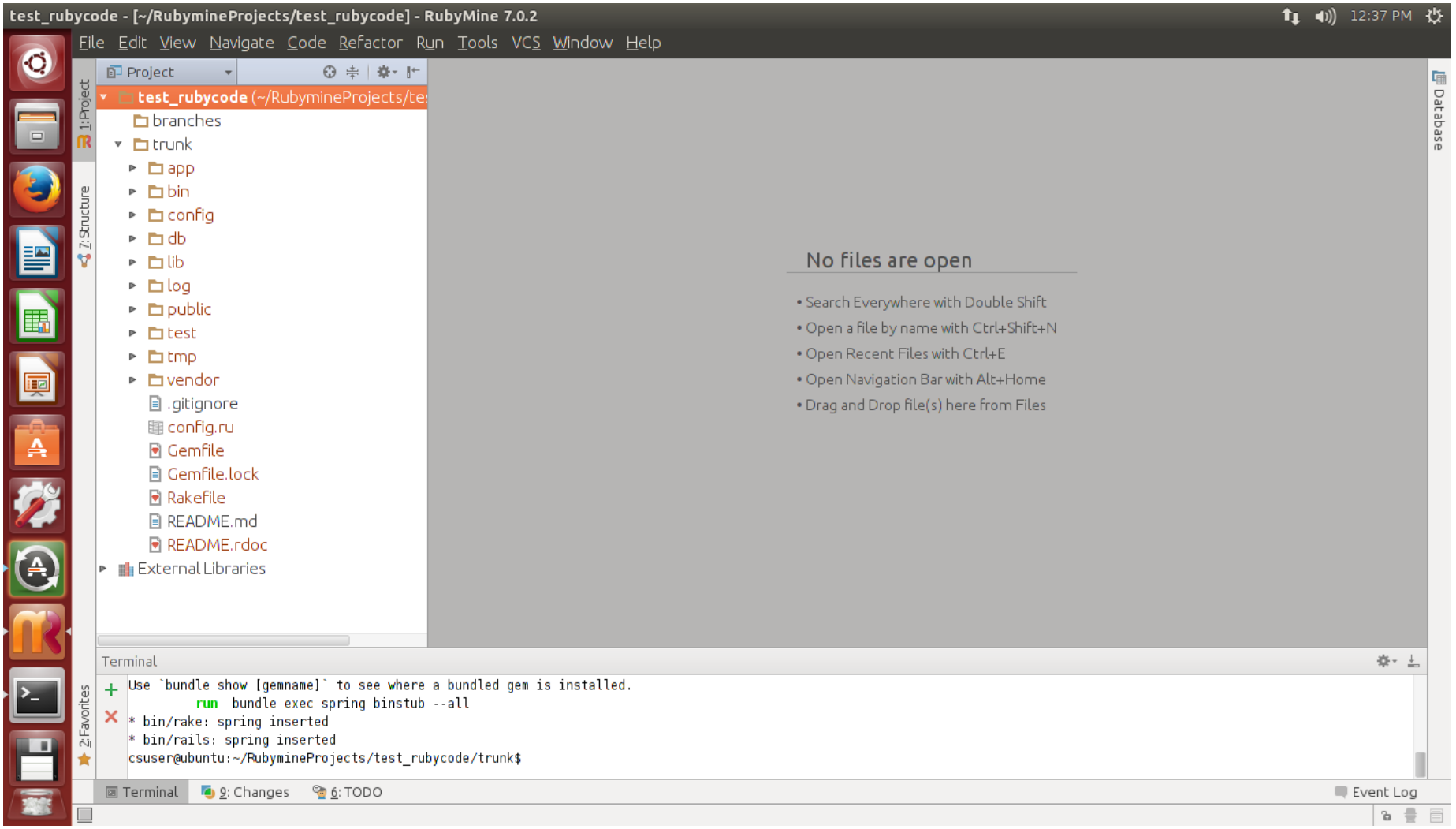
Pushing the initial Code

- 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.

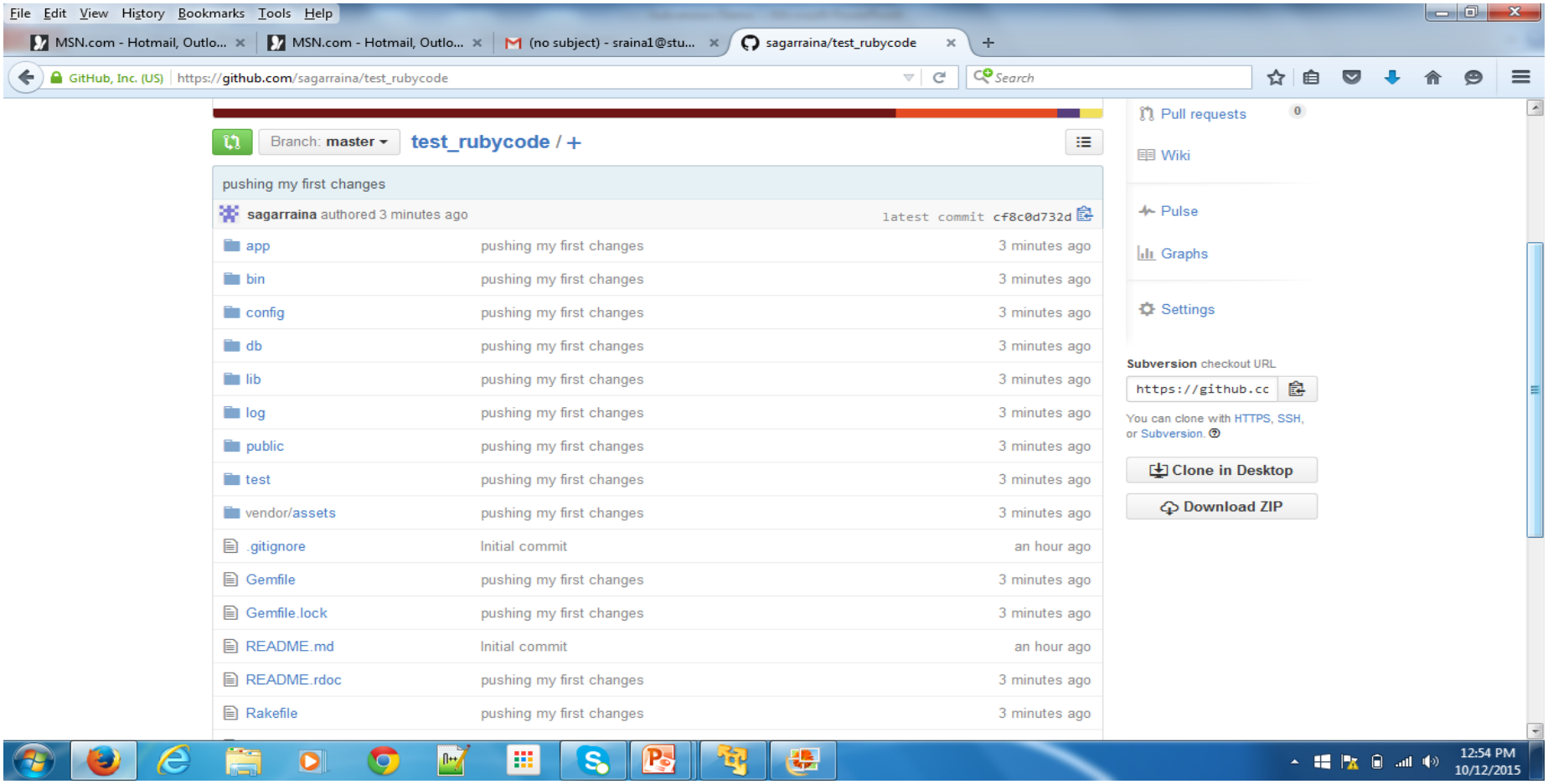


Pushing the initial Code

- Subversion does not commit empty folders, therefore, we
  will not commit the tmp folder.
- 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”
- Go to your remote repository and see the recently committed files

Pushing the initial code

- In github repository you will see the recently committed code



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:
- svn co https://github.com/*user/repo* *(Put the url you copied from repo site)*
- cd *repo (replace 'repo' with your repository name)*
- svn up trunk
- svn update

Changing and committing files

- Whenever you want to make change in your code, before changes get the latest code from the repository.
- Go to your trunk folder from rubymine and enter the following command
  - svn update

- Making changes to your code (this is just an example).
  Go to trunk -> app -> controllers - > application_controller.rb
- Go to trunk folder using rubymine terminal and enter
  - svn commit –m “I am committing the code”

Resolving conflicts

- When multiple people are changing and committing the same file, this might lead to conflicts.
- You will need to resolve the conflict before committing the code.
- The version control system will give you several options to resolve the conflicts.
  - Either manually, or by copying code from one file to another, or merging

External Resources

- Guide to checkout an existing project in Github:

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