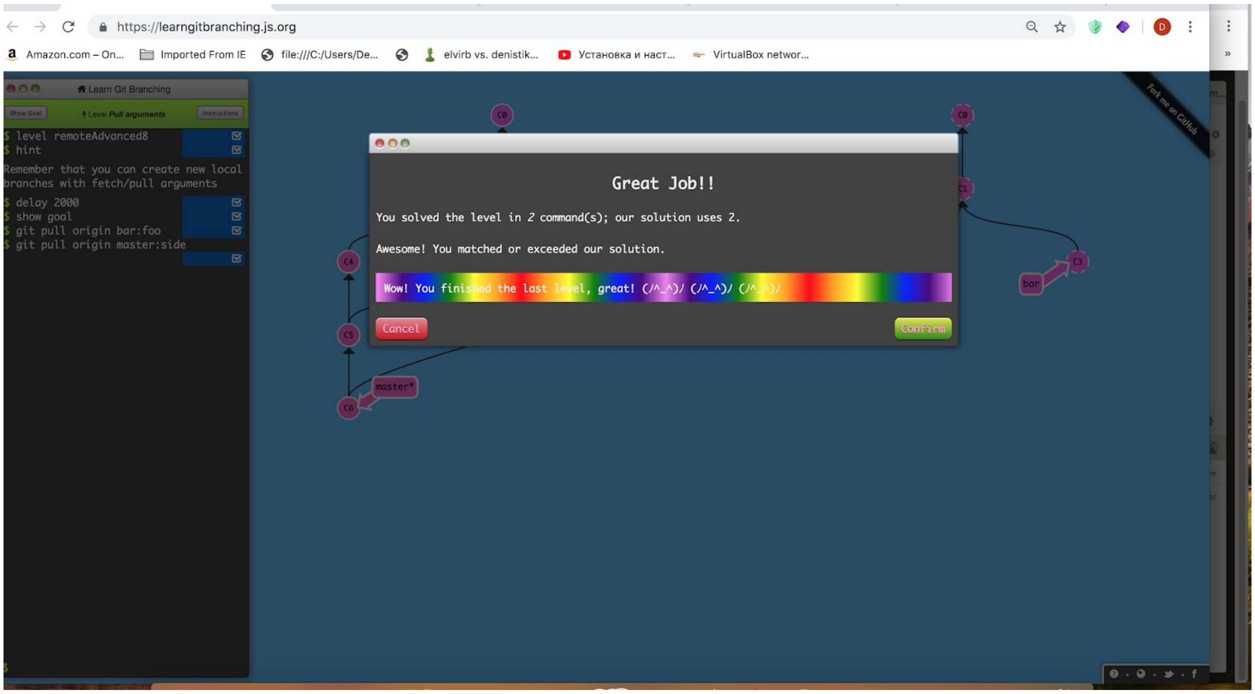
Dennis Tikhomirov **DevOps training, Spring’19 Homework**

Task 8. Git

Part I. Complete all tasks at https://learngitbranching.js.org/ Attach screenshots of completed tasks to the report.



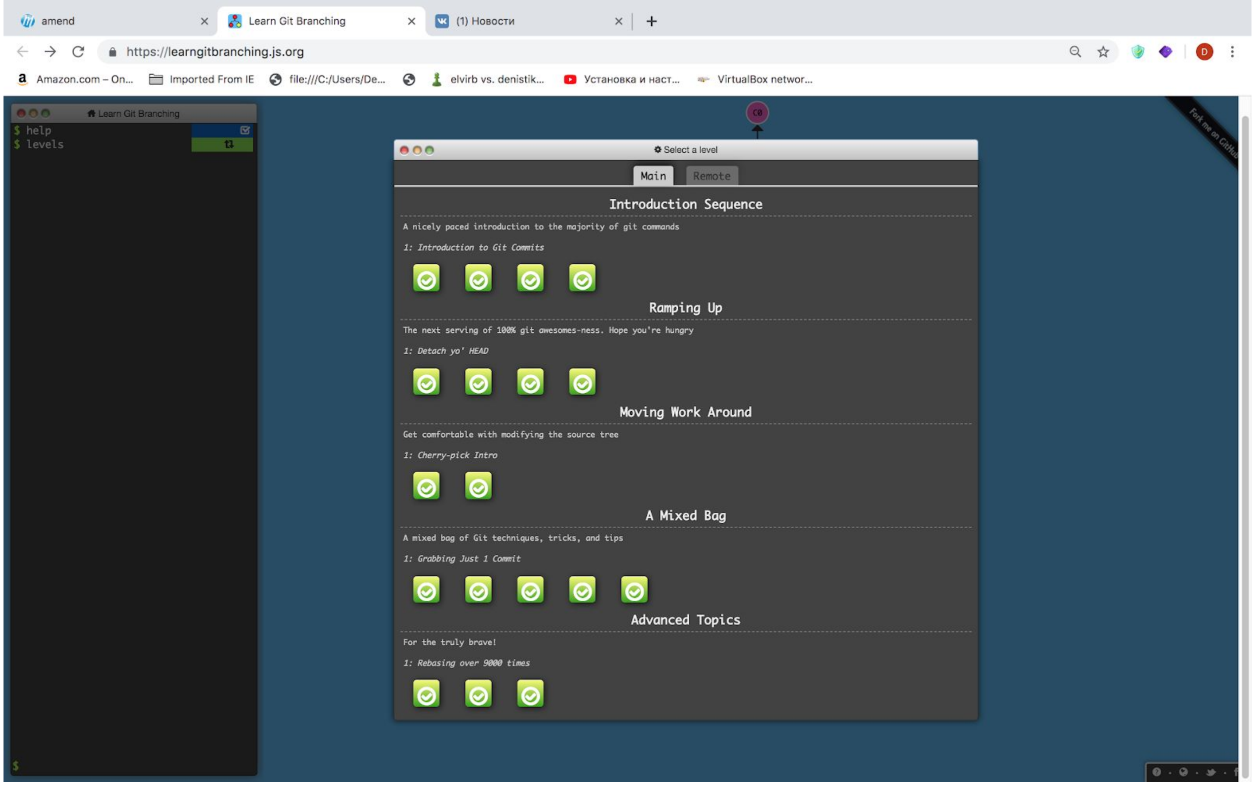
```
$ level remoteAdvanced8
$ hint
Remember that you can create new local
branches with fetch/pull arguments
$ delay 2000
$ show goal
$ git pull origin bar:foo
$ git pull origin master:side
```

Great Job!!

You solved the level in *2* command(s); our solution uses 2.

Awesome! You matched or exceeded our solution.

Wow! You finished the last level, great! (/^_^)/ (/^_^)/ (/^_^)/



```
$ help
$ levels
$
```

Introduction Sequence

A nicely paced introduction to the majority of git commands

1: Introduction to Git Commits

✓ ✓ ✓ ✓

Ramping Up

The next serving of 100% git awesomes-ness. Hope you're hungry

1: Detach yo' HEAD

✓ ✓ ✓ ✓

Moving Work Around

Get comfortable with modifying the source tree

1: Cherry-pick Intro

✓ ✓

A Mixed Bag

A mixed bag of Git techniques, tricks, and tips

1: Grabbing Just 1 Commit

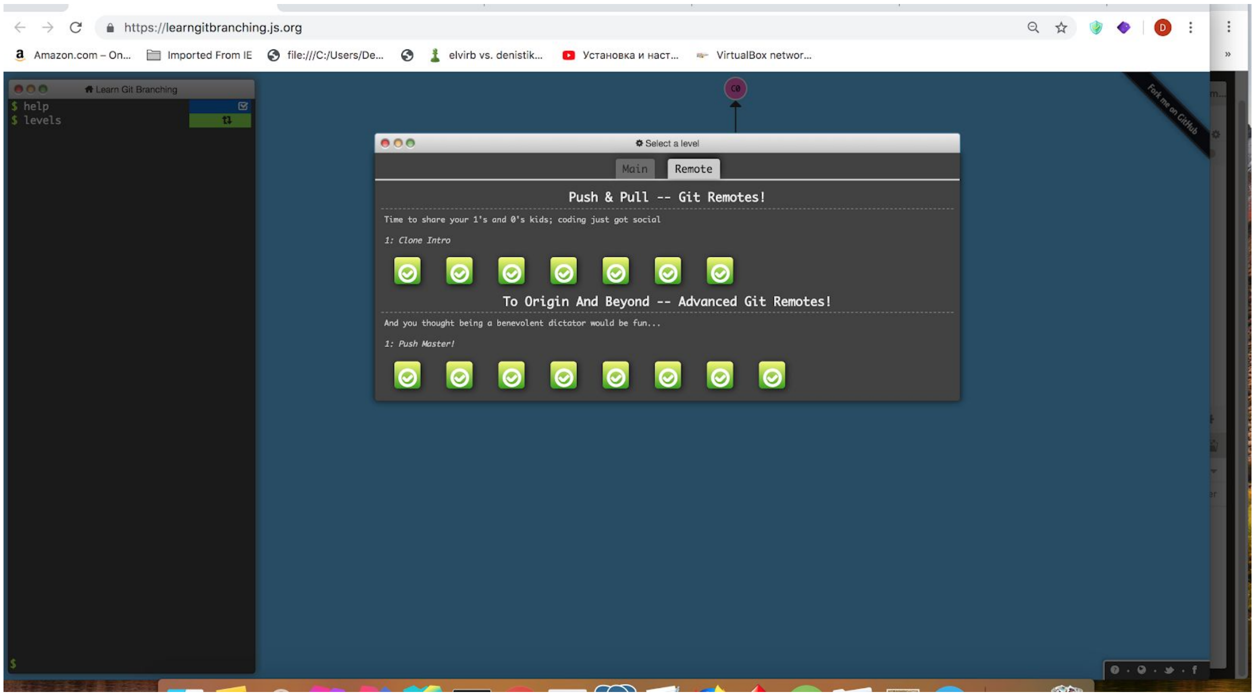
✓ ✓ ✓ ✓ ✓

Advanced Topics

For the truly brave!

1: Rebasing over 9000 times

✓ ✓ ✓



```
$ help
$ levels
$
```

Push & Pull -- Git Remotes!

Time to share your 1's and 0's kids; coding just got social

1: Clone Intro

✓ ✓ ✓ ✓ ✓ ✓ ✓

To Origin And Beyond -- Advanced Git Remotes!

And you thought being a benevolent dictator would be fun...

1: Push Master!

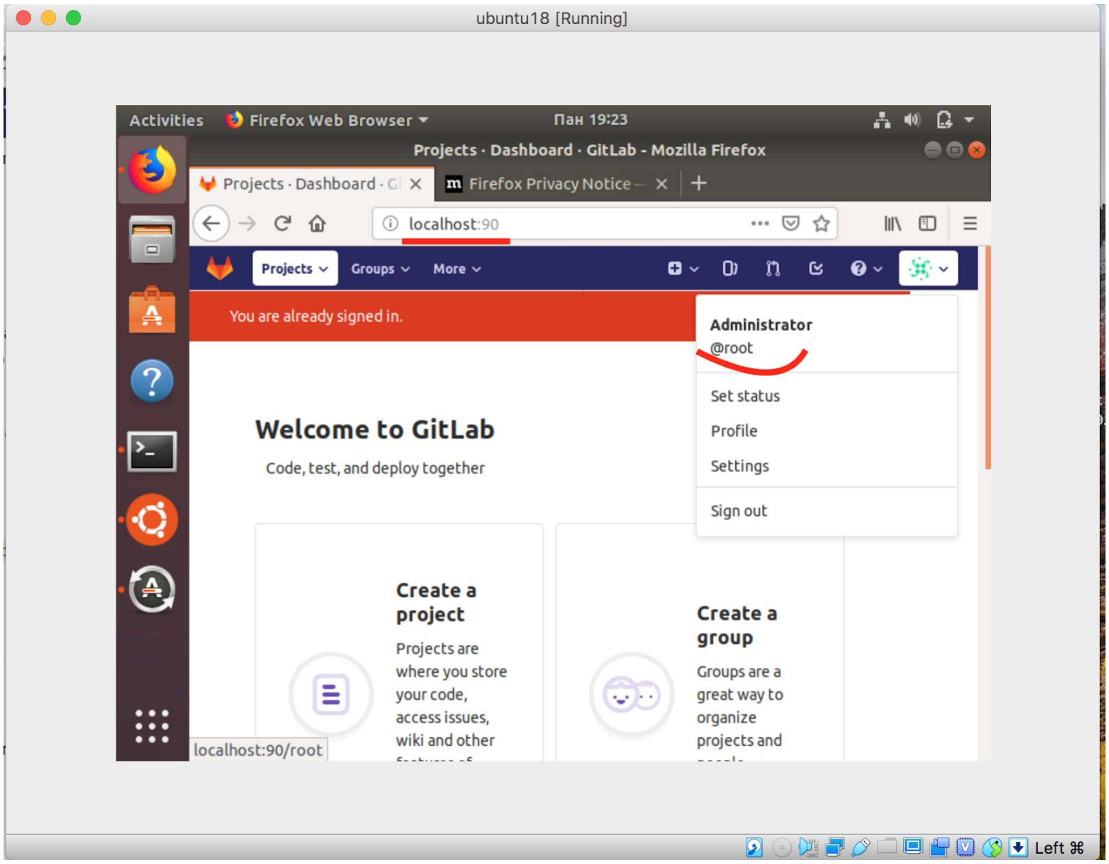
✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓

Part II Prerequisites

Ubuntu 18.04 virtual machine or baremetal, 64bit is preferred

Install Gitlab

1. You need to install Gitlab on your machine. To do this, follow the instruction on official site: https://about.gitlab.com/downloads/#ubuntu1604
2. After the last step of the instruction, you should have working GitLab instance and be logged in as root user.



You are already signed in.

Welcome to GitLab

Code, test, and deploy together

Create a project

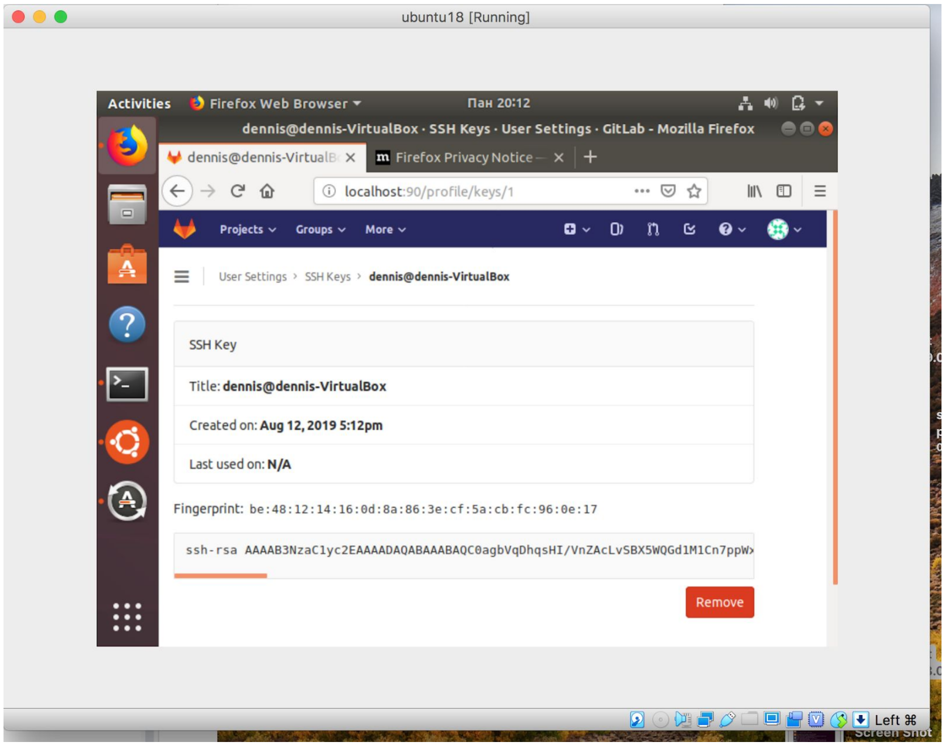
Projects are where you store your code, access issues, wiki and other

Create a group

Groups are a great way to organize projects and

sshkeygenCreate a project

1. In Gitlab, choose “New Project” and name it whatever you like.
2. In your Ubuntu machine, generate a pair of SSH keys by running sshkeygen
3. After that, you will have new files in ~/.ssh: id_rsa and id_rsa.pub
4. Copy contents of id_rsa.pub and go to Gitlab > Profile Settings >SSH keys. Paste your public key and save. Now you’re able to access your repositories via SSH without typing your credentials every time.


ubuntu18 [Running]
Activities
Firefox Web Browser
Пан 20:12
dennis@dennis-VirtualBox · SSH Keys · User Settings · GitLab - Mozilla Firefox
dennis@dennis-VirtualB
Firefox Privacy Notice —
localhost:90/profile/keys/1
Projects
Groups
More
User Settings › SSH Keys › dennis@dennis-VirtualBox
Remove
Left ⌘

SSH Key

Title: **dennis@dennis-VirtualBox**

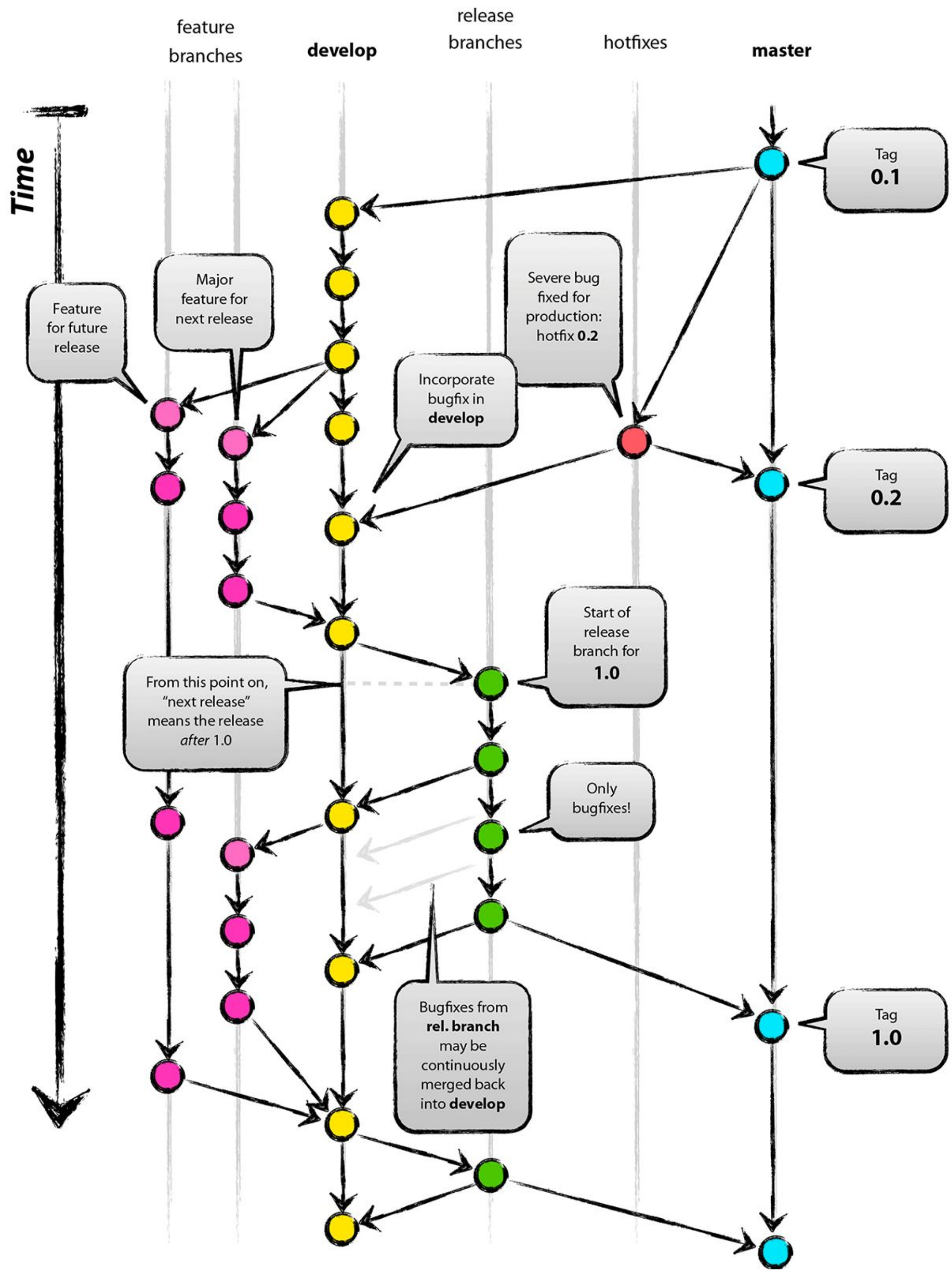
Created on: **Aug 12, 2019 5:12pm**

Last used on: **N/A**

Fingerprint: `be:48:12:14:16:0d:8a:86:3e:cf:5a:cb:fc:96:0e:17`

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQC0agbVqDhqsHI/VnZAcLvSBX5WQGd1M1Cn7ppWx
```

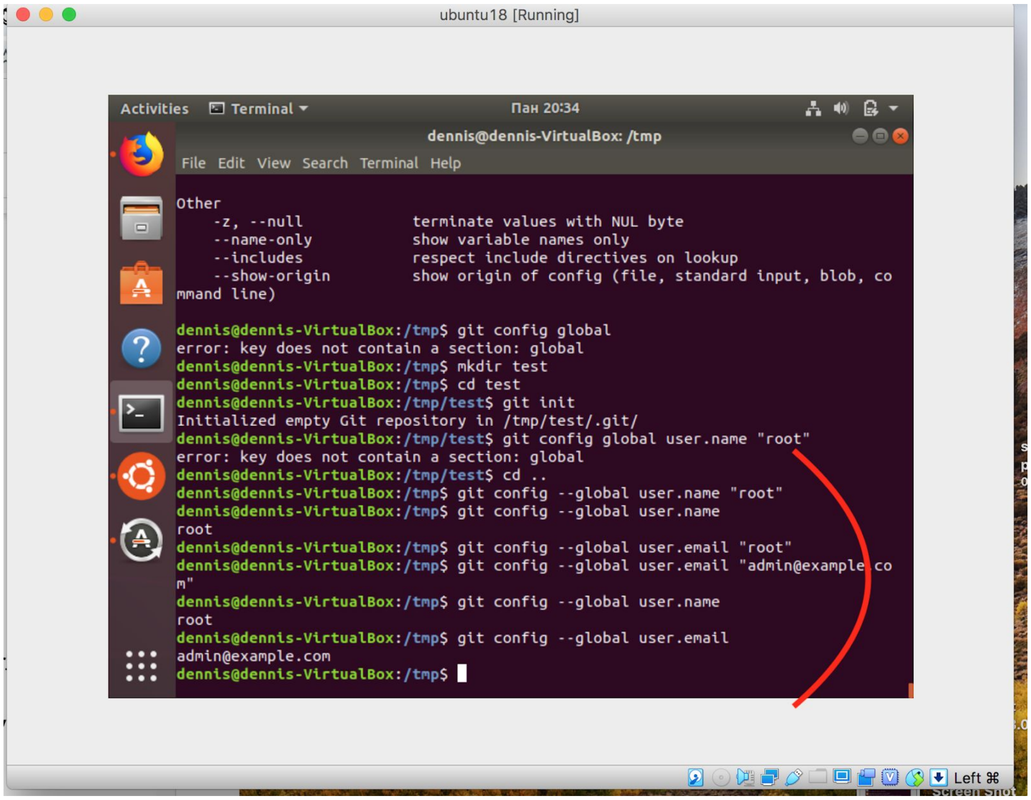
Set up git flow


feature branches
develop
release branches
hotfixes
master
Time
Tag 0.1
Severe bug fixed for production: hotfix 0.2
Major feature for next release
Feature for future release
Incorporate bugfix in develop
Tag 0.2
Start of release branch for 1.0
From this point on, "next release" means the release after 1.0
Only bugfixes!
Bugfixes from rel. branch may be continuously merged back into develop
Tag 1.0

1. Install git on your ubuntu if you don't have it.

2. Set up your user by running commands:

$ git config global
user.name "John Doe"
$ git config global
user.email johndoe@example.com



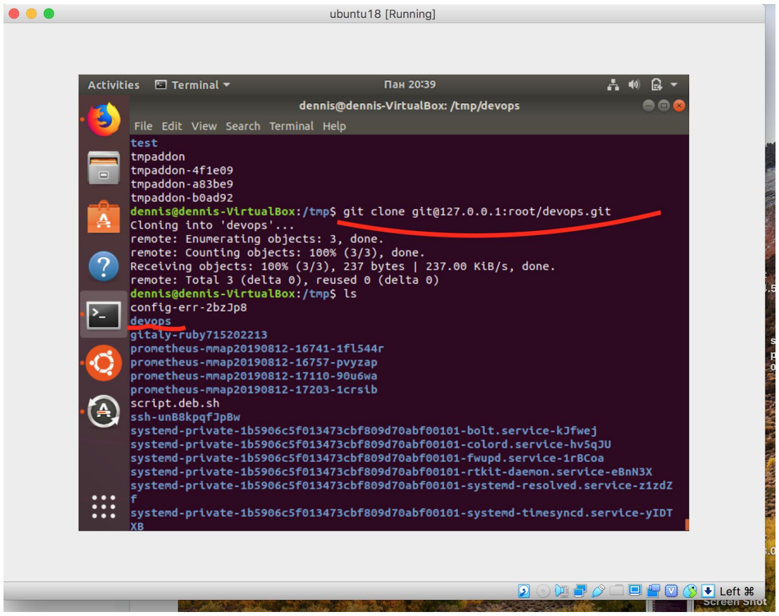
ubuntu18 [Running]
Activities
Terminal
Пан 20:34
dennis@dennis-VirtualBox: /tmp
File Edit View Search Terminal Help
Left ⌘

```
Other
    -z, --null            terminate values with NUL byte
    --name-only           show variable names only
    --includes            respect include directives on lookup
    --show-origin         show origin of config (file, standard input, blob, co
mmand line)

dennis@dennis-VirtualBox:/tmp$ git config global
error: key does not contain a section: global
dennis@dennis-VirtualBox:/tmp$ mkdir test
dennis@dennis-VirtualBox:/tmp$ cd test
dennis@dennis-VirtualBox:/tmp/test$ git init
Initialized empty Git repository in /tmp/test/.git/
dennis@dennis-VirtualBox:/tmp/test$ git config global user.name "root"
error: key does not contain a section: global
dennis@dennis-VirtualBox:/tmp/test$ cd ..
dennis@dennis-VirtualBox:/tmp$ git config --global user.name "root"
dennis@dennis-VirtualBox:/tmp$ git config --global user.name
root
dennis@dennis-VirtualBox:/tmp$ git config --global user.email "root"
dennis@dennis-VirtualBox:/tmp$ git config --global user.email "admin@example.co
m"
dennis@dennis-VirtualBox:/tmp$ git config --global user.name
root
dennis@dennis-VirtualBox:/tmp$ git config --global user.email
admin@example.com
dennis@dennis-VirtualBox:/tmp$ 
```

3. Now clone your repository locally by running git clone <link_to_your_project>. You can get link on a project page in GitLab. Be sure to copy SSH link.

```
git clone git@127.0.0.1:root/devops.git
```

4. Go to your new folder containing downloaded project.

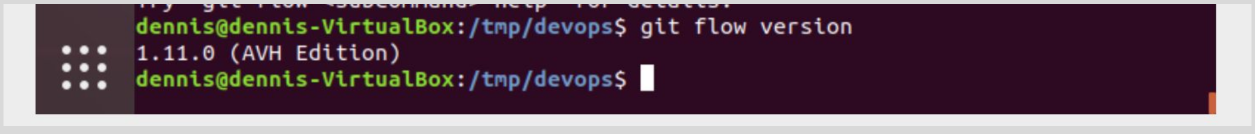


ubuntu18 [Running]
Activities
Terminal
Пан 20:39
dennis@dennis-VirtualBox: /tmp/devops
File Edit View Search Terminal Help
Left ⌘

```
test
tmpaddon
tmpaddon-4f1e09
tmpaddon-a83be9
tmpaddon-b0ad92
dennis@dennis-VirtualBox:/tmp$ git clone git@127.0.0.1:root/devops.git
Cloning into 'devops'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
Receiving objects: 100% (3/3), 237 bytes | 237.00 KiB/s, done.
remote: Total 3 (delta 0), reused 0 (delta 0)
dennis@dennis-VirtualBox:/tmp$ ls
config-err-2bzJp8
devops
gitaly-ruby715202213
prometheus-mmap20190812-16741-1fl544r
prometheus-mmap20190812-16757-pvyzap
prometheus-mmap20190812-17110-90u6wa
prometheus-mmap20190812-17203-1crsib
script.deb.sh
ssh-unB8kpqfJpBw
systemd-private-1b5906c5f013473cbf809d70abf00101-bolt.service-kJfwej
systemd-private-1b5906c5f013473cbf809d70abf00101-colord.service-hv5qJU
systemd-private-1b5906c5f013473cbf809d70abf00101-fwupd.service-1rBCoa
systemd-private-1b5906c5f013473cbf809d70abf00101-rtkit-daemon.service-eBnN3X
systemd-private-1b5906c5f013473cbf809d70abf00101-systemd-resolved.service-z1zdZ
f
systemd-private-1b5906c5f013473cbf809d70abf00101-systemd-timesyncd.service-yIDT
XB
```

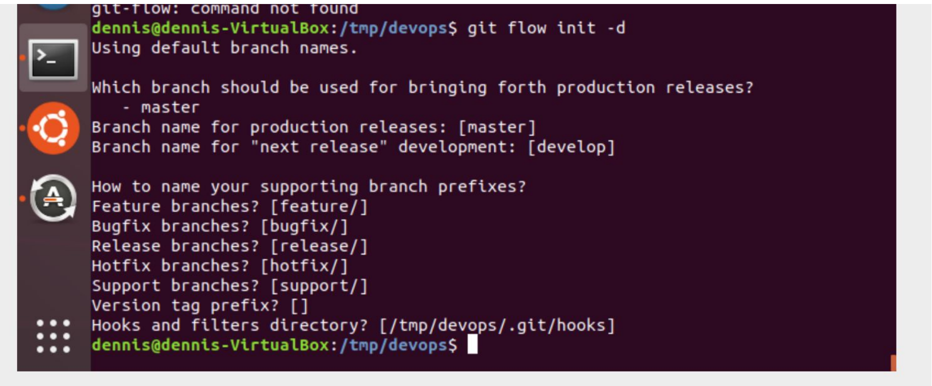
5. Install git flow. Official repo can be found here: https://github.com/nvie/gitflow

```
apt-get install git-flow
```



```
Try 'git flow <subcommand> help' for details.
dennis@dennis-VirtualBox:/tmp/devops$ git flow version
1.11.0 (AVH Edition)
dennis@dennis-VirtualBox:/tmp/devops$ 
```

6. Run gitflow init d to initialise git flow with all defaults.



```
git-flow: command not found
dennis@dennis-VirtualBox:/tmp/devops$ git flow init -d
Using default branch names.

Which branch should be used for bringing forth production releases?
   - master
Branch name for production releases: [master]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []
Hooks and filters directory? [/tmp/devops/.git/hooks]
dennis@dennis-VirtualBox:/tmp/devops$ 
```

Make yourself familiar with gitflow

1. Create some text file in your project and place it under version control

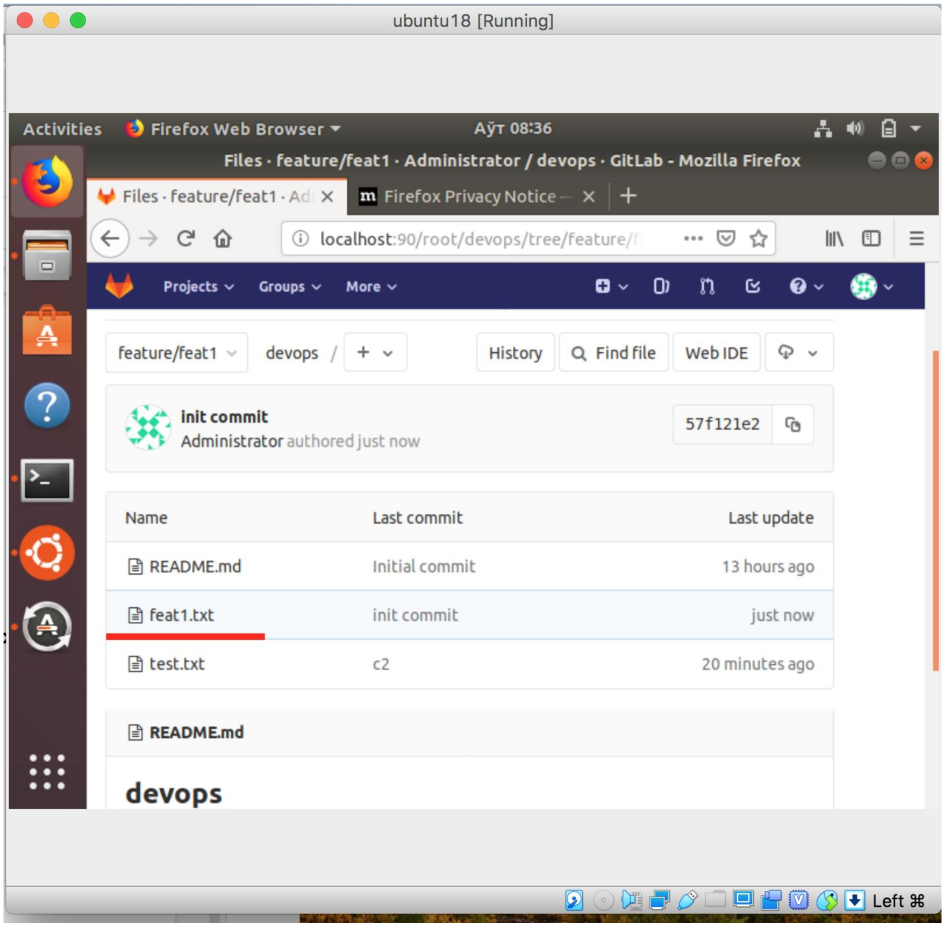
2. Push your changes in master branch

Current branch is develop:

```
    git add .
    git commit -m ‘init commit’
    git merge master
```

3. Create new feature branch, make some changes and finish/push it the remote. Track what’s happening with your repo on each step of working with branch.

```
   git flow feature start feat1
   cat > feat1.txt
    git add .
    git commit -m ‘feat1 init commit’
    git flow feature publish feat1
```



init commit
Administrator authored just now
57f121e2

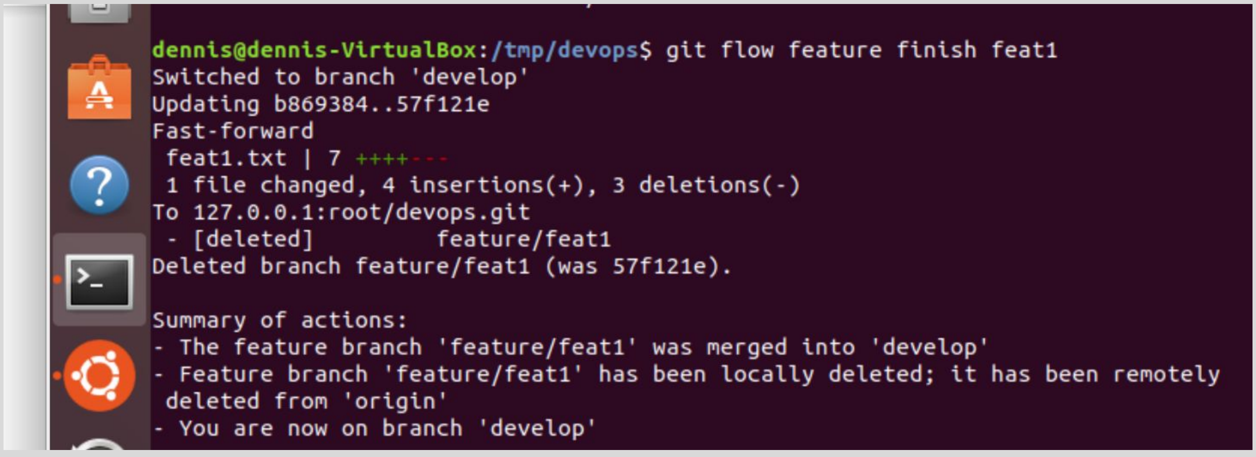
Name	Last commit	Last update
README.md	Initial commit	13 hours ago
feat1.txt	init commit	just now
test.txt	c2	20 minutes ago

README.md

devops

```
git flow feature finish feat1
```

After this command branch feat1 was merged into develop branch and deleted.



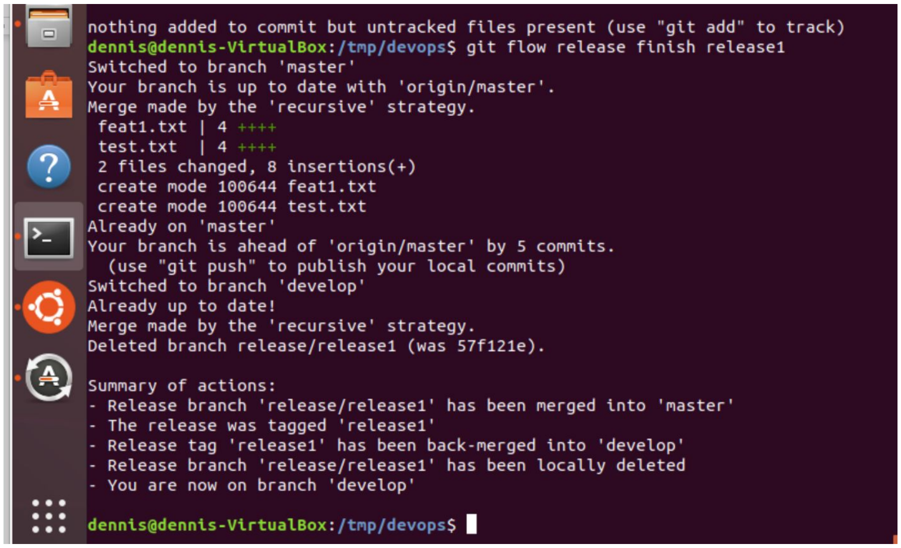
```
dennis@dennis-VirtualBox:/tmp/devops$ git flow feature finish feat1
Switched to branch 'develop'
Updating b869384..57f121e
Fast-forward
 feat1.txt | 7 ++++---
 1 file changed, 4 insertions(+), 3 deletions(-)
To 127.0.0.1:root/devops.git
 - [deleted]         feature/feat1
Deleted branch feature/feat1 (was 57f121e).

Summary of actions:
- The feature branch 'feature/feat1' was merged into 'develop'
- Feature branch 'feature/feat1' has been locally deleted; it has been remotely
  deleted from 'origin'
- You are now on branch 'develop'
```

4. Create new release branch and finish it. Track what’s happening with your repo on each step of working with branch.

```
git flow release start release1
```

New branch release1 was created.

```
 git flow feature start feat1
     cat > feat1.txt
      git add .
      git commit -m ‘release1 init commit’
   git flow release finish release1
# merged into MASTER, merged into DEVELOP, branch RELEASE1 was deleted
```



```
nothing added to commit but untracked files present (use "git add" to track)
dennis@dennis-VirtualBox:/tmp/devops$ git flow release finish release1
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
Merge made by the 'recursive' strategy.
 feat1.txt | 4 ++++
 test.txt  | 4 ++++
 2 files changed, 8 insertions(+)
 create mode 100644 feat1.txt
 create mode 100644 test.txt
Already on 'master'
Your branch is ahead of 'origin/master' by 5 commits.
  (use "git push" to publish your local commits)
Switched to branch 'develop'
Already up to date!
Merge made by the 'recursive' strategy.
Deleted branch release/release1 (was 57f121e).

Summary of actions:
- Release branch 'release/release1' has been merged into 'master'
- The release was tagged 'release1'
- Release tag 'release1' has been back-merged into 'develop'
- Release branch 'release/release1' has been locally deleted
- You are now on branch 'develop'

dennis@dennis-VirtualBox:/tmp/devops$ █
```