

Лабораторная работа №1 по дисциплине «ТРПО»

Перед выполнением данной лабораторной работы необходимо зарегистрироваться на <https://github.com/>

Каждый шаг каждой задачи необходимо документировать и описывать.

Отчеты по данной лабораторной работе необходимо принести в письменном виде(!)

Task 1

1. Установить консольный git-клиент и настроив его параметры `user.name` и `user.email`, соответствующие вашим данным, не используя опцию `--global`
2. Создать новый репозиторий **entry_lab** с помощью команды `git init`.
3. Создать файл `nginx.conf`, который содержит базовую настройку `nginx` (стандартная конфигурация `nginx` после установки соответствующего пакета в `linux /etc/nginx/nginx.conf`).
4. Создать файл `Readme.md` с описанием содержимого репозитория.
5. Добавить в репозиторий файлы **nginx.conf** и **Readme.md** с помощью команды `git add` и зафиксировать их состояния с помощью `git commit`.
6. Проверить историю коммитов и убедиться, что в истории два коммита.
7. С помощью команды `git status` проверить что в локальном репозитории нет больше файлов/папок, которые в него не добавлены.
8. Залить все изменения в ветку **master** созданного удаленного репозитория.
9. Склонировать ваш удаленный репозиторий в свой локальный, предварительно настроив доступ к репозиторию по **ssh**.
10. Выполнить три произвольных изменения в файле **nginx.conf** и зафиксировать эти изменения.
11. Переключиться на 2-й коммит сначала и просмотреть содержимое файла **nginx.conf**.
12. Добавить легковесный тег **0.1** на второй коммит сначала. Загрузить теги и все изменения в свой удаленный репозиторий.
13. В своем локальном репозитории создать файл **.gitignore**, который должен исключать из репозитория папку `tmp` и все ее содержимое.
14. Создать в своем локальном репозитории папку **tmp** с несколькими файлами внутри. Выполнить фиксацию добавленных изменений.
15. Убедиться, что изменения сделанные в папке **tmp** не попадают в коммит.

Task 2

1. В созданном ранее репозитории создайте две ветки - **develop** и **feature-new-site**
2. В ветке **develop** измените настройки **worker_connection** на **16384** в файле **nginx.conf** и зафиксируйте эти изменения
3. В ветке **develop** включите сжатие ответа для типа **application/json** в файле **nginx.conf** и зафиксируйте эти изменения
4. В ветке **feature-new-site** добавьте файл **conf.d/mysite.domain.com.conf** с базовым описанием статического сайта и зафиксируйте эти изменения
5. Запустите обе ветки в свой удаленный репозиторий и проверьте полученный результат.
6. Выполните объединения(**merge**) ветки **develop** и ветки **master** (прислать графическое представление полученного результата, утилиты **Git Kraken** или **gitk** в помощь).
7. И залейте результат слияния веток в свой удаленный репозиторий.
8. Выполните слияние ветки **feature-new-site** в **develop**, затем выполните изменения в соответствующих ветках в одном и том же файле повторив процедуру слияния, с разрешением конфликтов, оставив последнее изменение по времени.
9. Сделайте несколько коммитов в ветке **feature-new-site**, а затем выполните перемещение коммитов в ветку **develop** с помощью **git rebase**.
10. Аналогично выполнить перемещение истории коммитов из ветки **develop** в ветку **master**.
11. Запустить все изменения в свой удаленный репозиторий
12. Создайте четыре коммита в ветке **develop**
13. Посмотрите историю коммитов в ветке **develop** и объединив 4 первых коммита в один, оставив сообщение 1 коммита (**fixup** или **squash**).
14. Выполните интерактивное перемещение новых коммитов в ветку **master**.
15. * (Со звездочкой - первые два студента, сдавшие лабу, получают бонус на экзамене/зачете) Создайте в ветке **develop** коммит с сообщением «**Formatted code**» и затем еще несколько коммитов в этой ветке с произвольными сообщениями, а затем перенесите коммит с сообщением «**Formatted code**» в ветку **master** используя **git cherry-pick**.

Task 3

1. В настройках своего удаленного репозитория создать 2 группы – **maintainer** и **developer**.
2. Настройте права для указанных групп таким образом, чтобы участникам группы **developer** было запрещено добавлять напрямую изменения в ветку **master**, а можно было только через **pull request**
3. Пришлите перекрестные приглашение друг другу для доступа к вашим репозиториям(в отчете напишите почту друга)
4. Склонируйте ветку **develop** удаленного репозитория друг друга. Выполните несколько изменений и зафиксируйте их.
5. Залейте изменения в соответствующую ветку удаленного репозитория и создайте запрос на слияние **develop** ветки и **master** через **pull request** (рассмотрев различные варианты слияния).