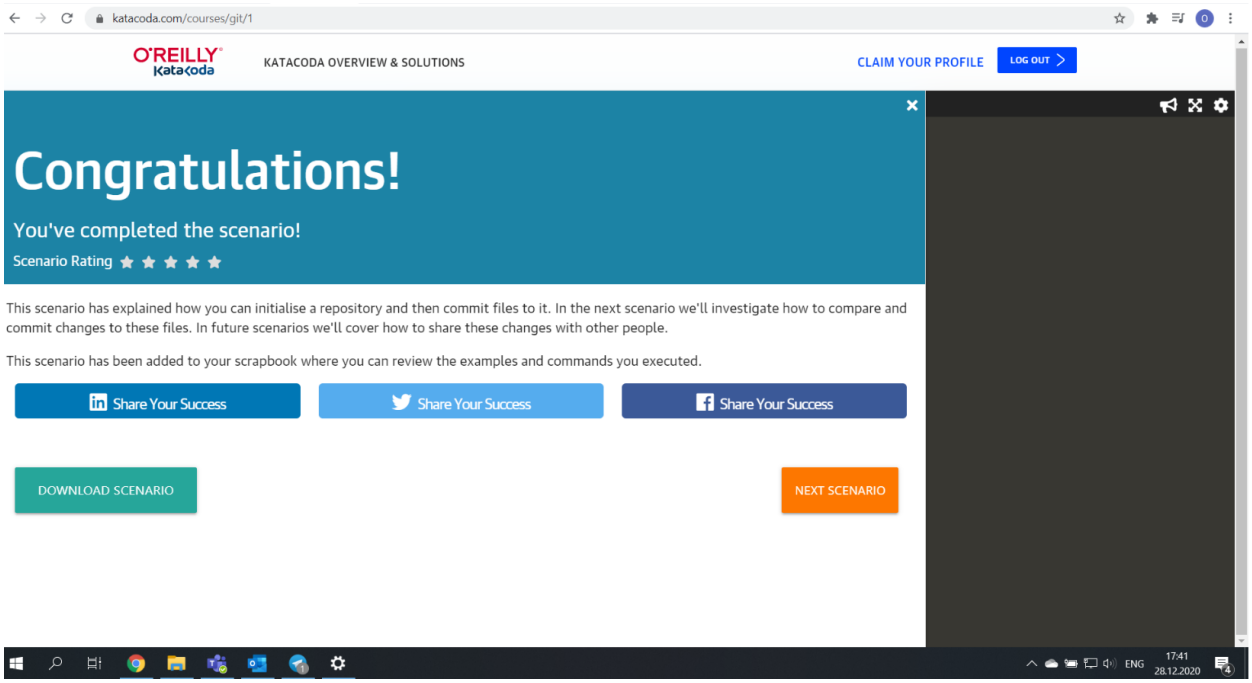
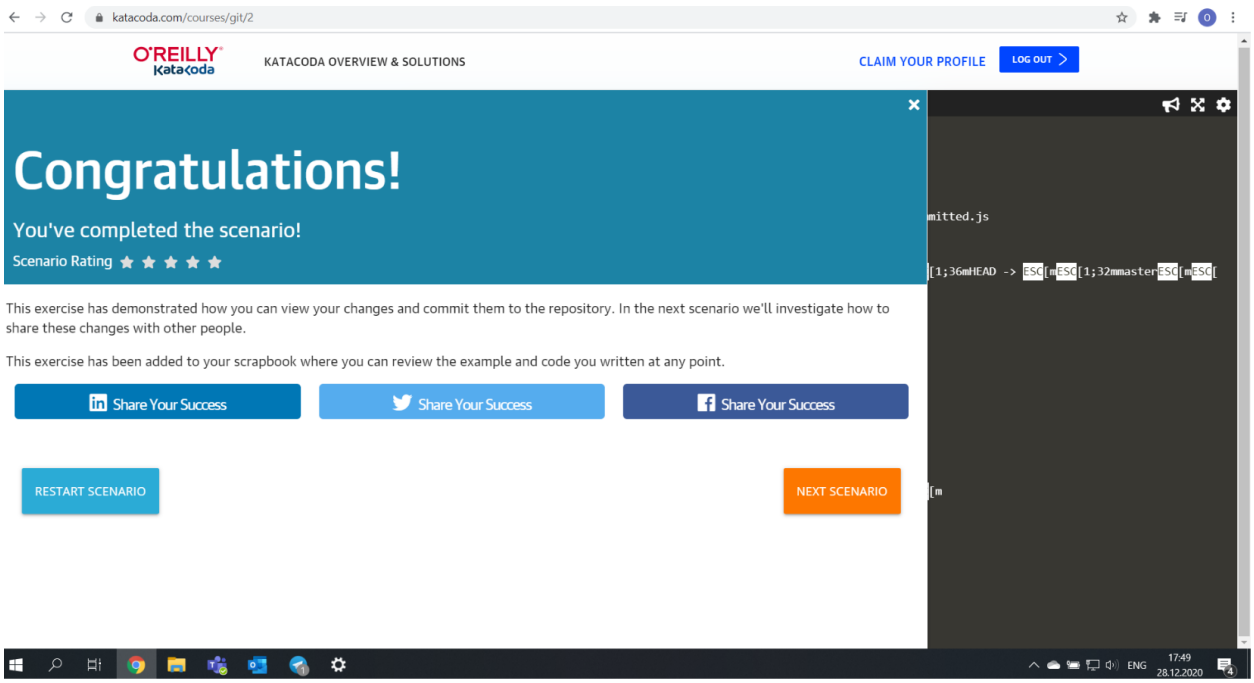
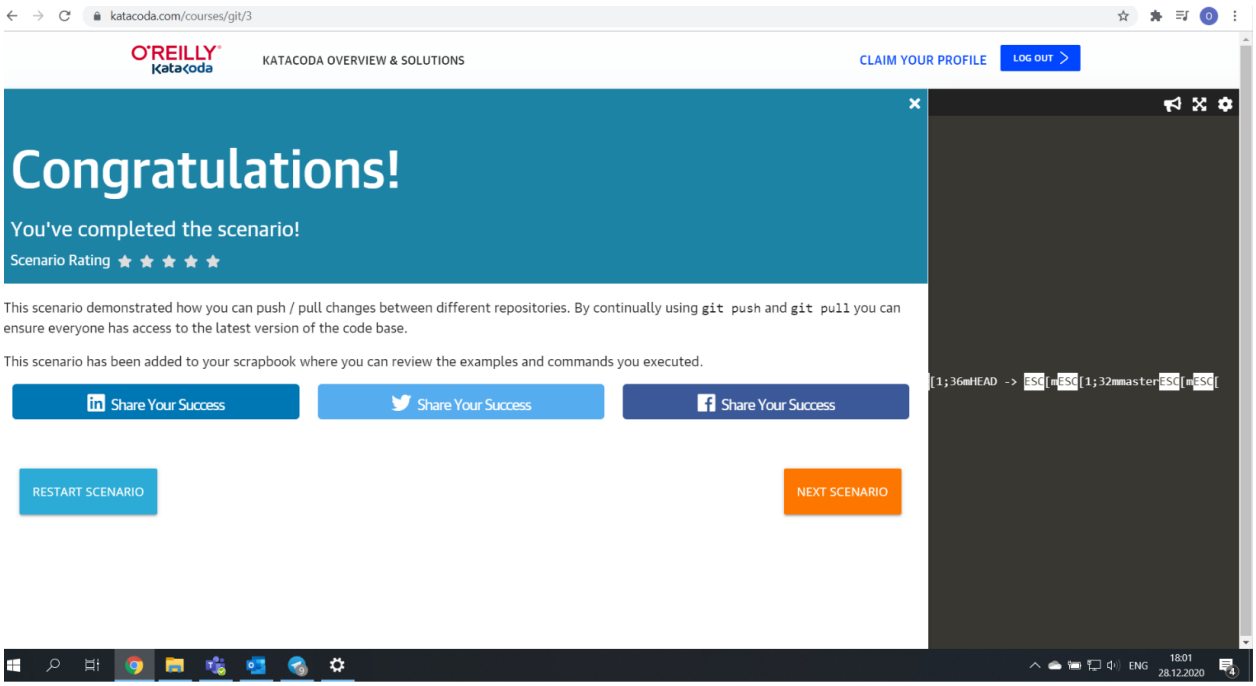
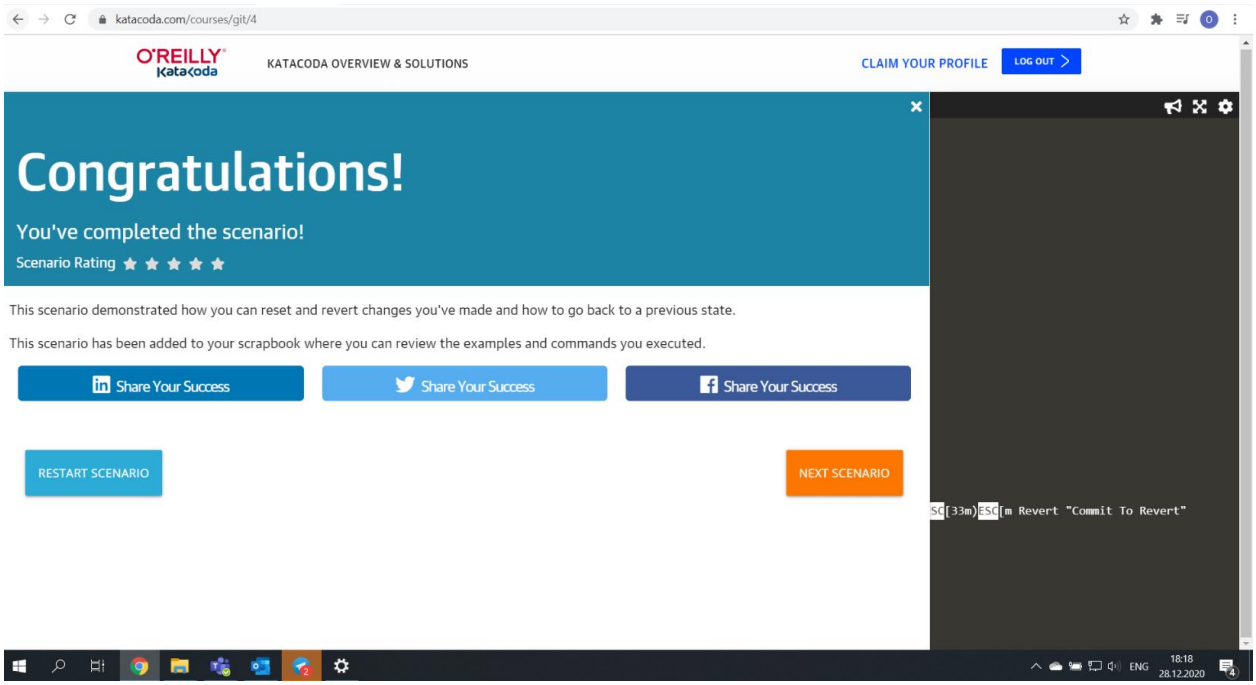
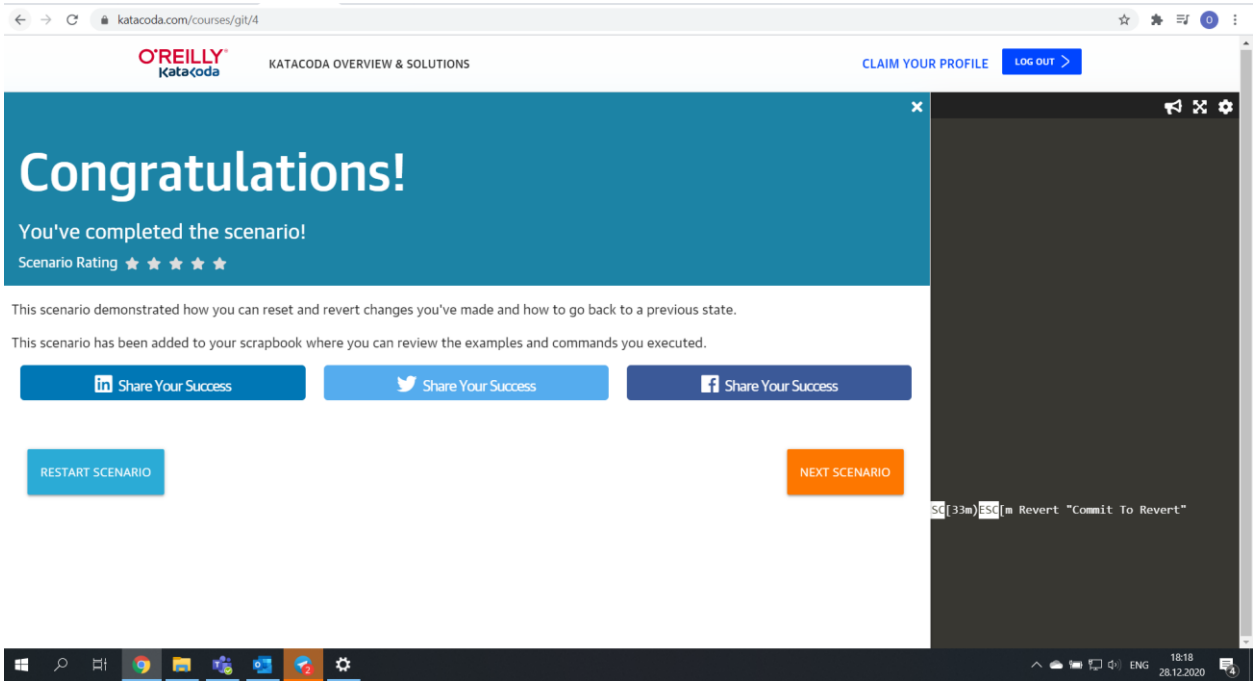
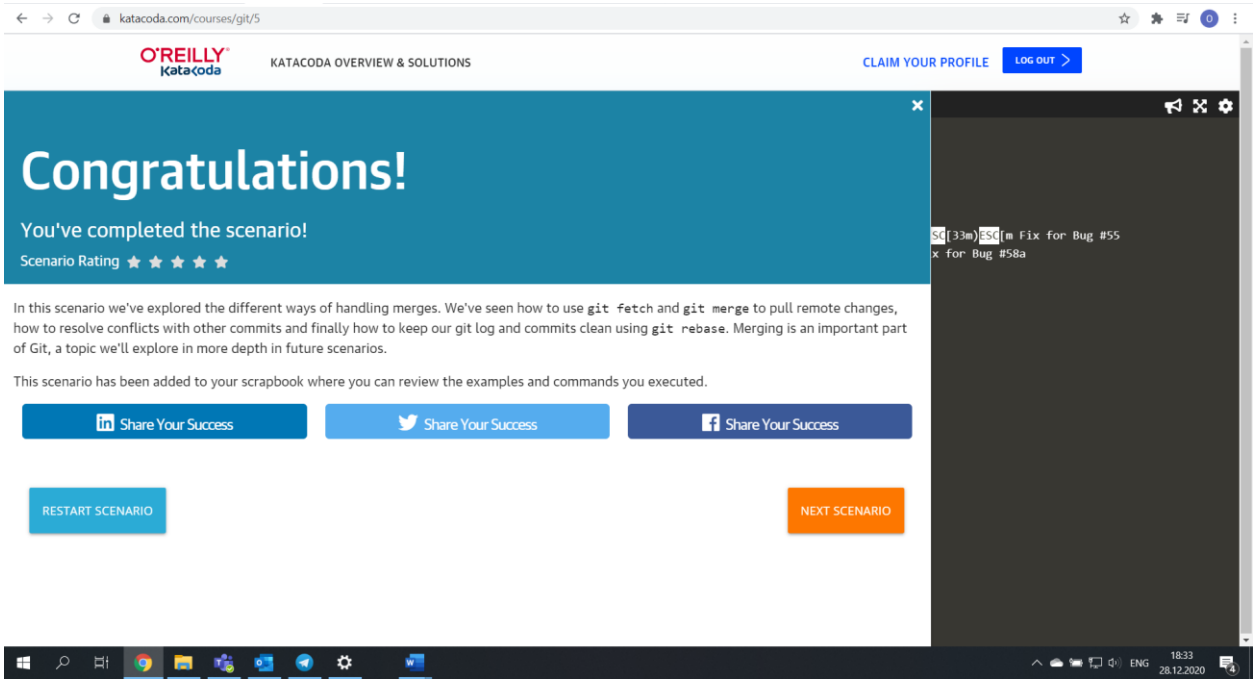

katacoda.com/courses/git/1
O'REILLY Katacoda
KATACODA OVERVIEW & SOLUTIONS
CLAIM YOUR PROFILE
LOG OUT
Congratulations!
You've completed the scenario!
Scenario Rating
This scenario has explained how you can initialise a repository and then commit files to it. In the next scenario we'll investigate how to compare and commit changes to these files. In future scenarios we'll cover how to share these changes with other people.
This scenario has been added to your scrapbook where you can review the examples and commands you executed.
Share Your Success
Share Your Success
Share Your Success
DOWNLOAD SCENARIO
NEXT SCENARIO
ENG
17:41
28.12.2020

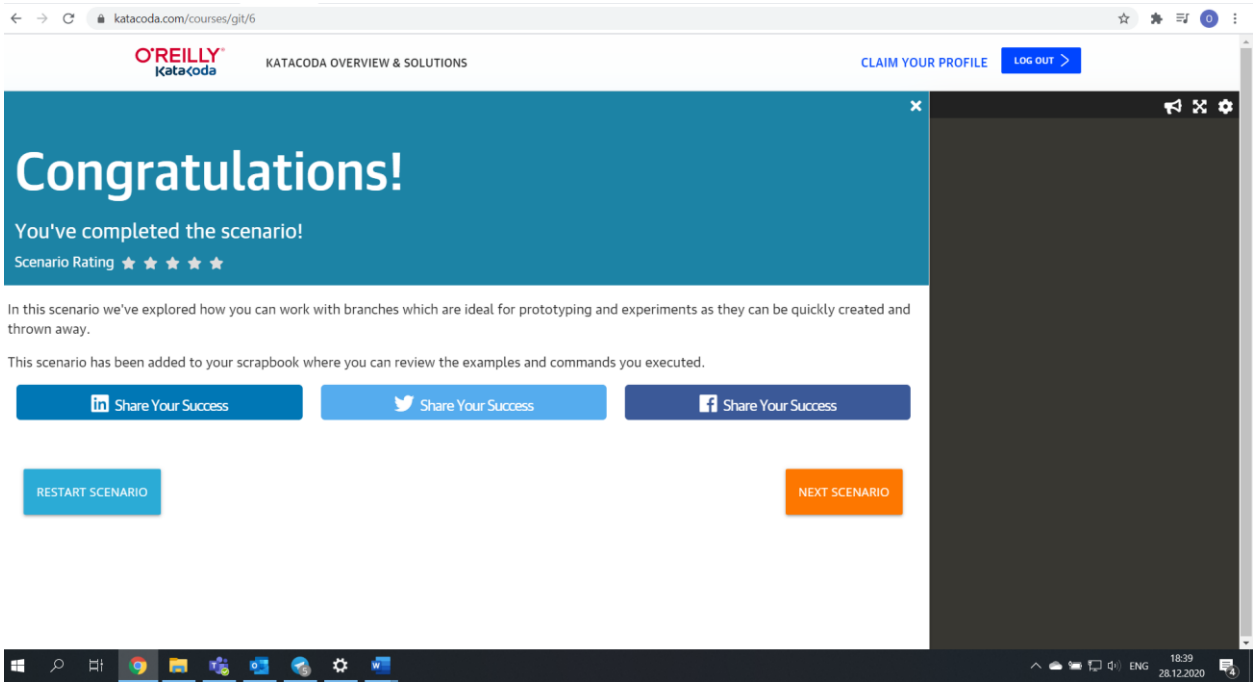

katacoda.com/courses/git/2
O'REILLY Katacoda
KATACODA OVERVIEW & SOLUTIONS
CLAIM YOUR PROFILE
LOG OUT
Congratulations!
You've completed the scenario!
Scenario Rating
This exercise has demonstrated how you can view your changes and commit them to the repository. In the next scenario we'll investigate how to share these changes with other people.
This exercise has been added to your scrapbook where you can review the example and code you written at any point.
Share Your Success
Share Your Success
Share Your Success
RESTART SCENARIO
NEXT SCENARIO
mitted.js
[1;36mHEAD -> ESC[mESC[1;32mmasterESC[mESC[
[m
ENG
17:49
28.12.2020

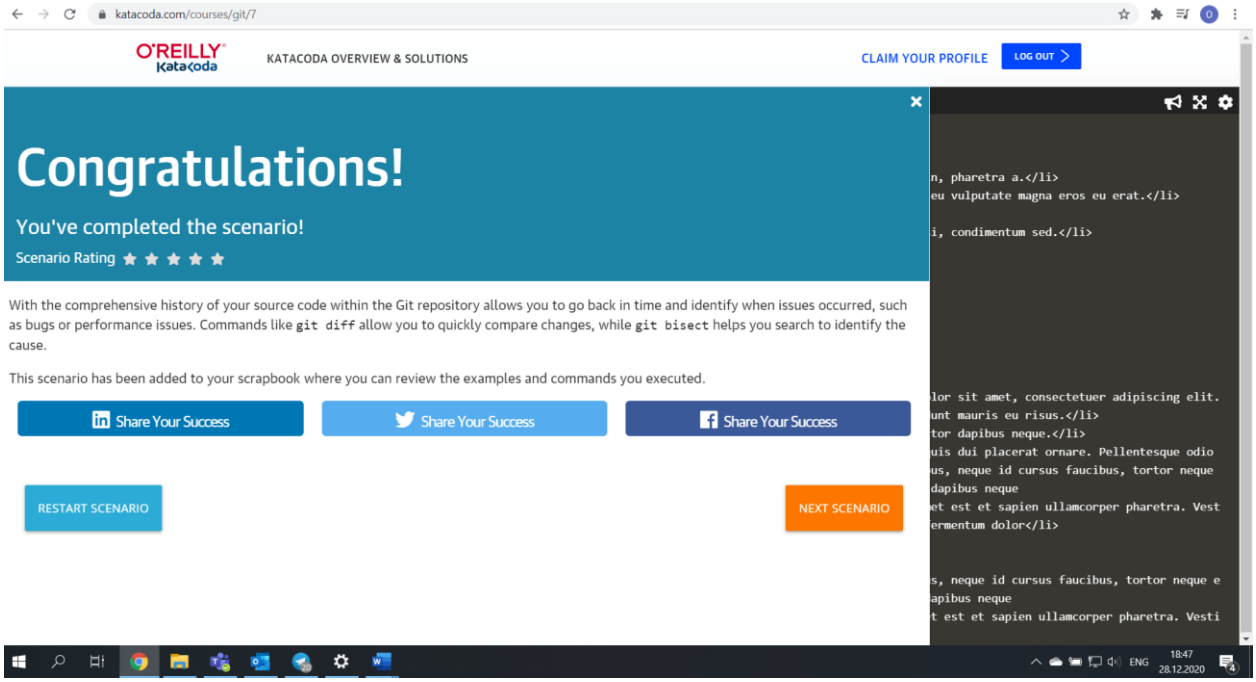

katacoda.com/courses/git/3
O'REILLY Katacoda
KATACODA OVERVIEW & SOLUTIONS
CLAIM YOUR PROFILE
LOG OUT
Congratulations!
You've completed the scenario!
Scenario Rating
This scenario demonstrated how you can push / pull changes between different repositories. By continually using git push and git pull you can ensure everyone has access to the latest version of the code base.
This scenario has been added to your scrapbook where you can review the examples and commands you executed.
Share Your Success
Share Your Success
Share Your Success
RESTART SCENARIO
NEXT SCENARIO
[1;36mHEAD -> ESC[mESC[1;32mmasterESC[mESC[
ENG
18:01
28.12.2020

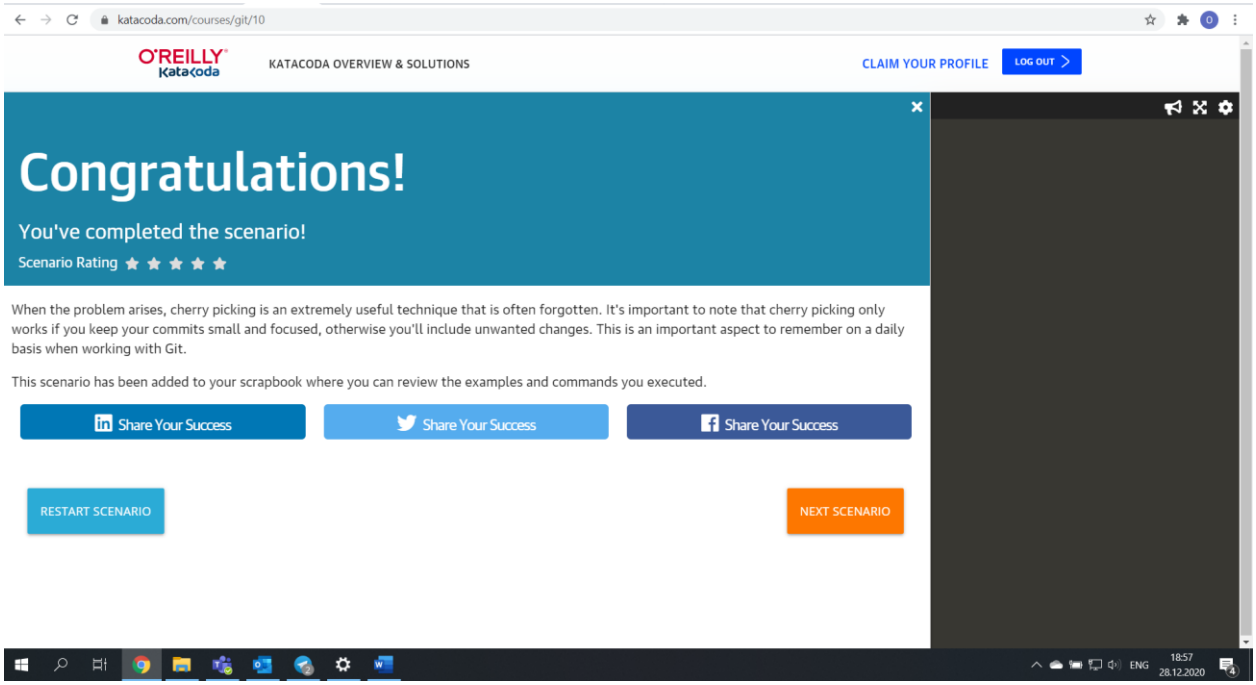

katacoda.com/courses/git/4
O'REILLY Katacoda
KATACODA OVERVIEW & SOLUTIONS
CLAIM YOUR PROFILE
LOG OUT
Congratulations!
You've completed the scenario!
Scenario Rating
This scenario demonstrated how you can reset and revert changes you've made and how to go back to a previous state.
This scenario has been added to your scrapbook where you can review the examples and commands you executed.
Share Your Success
Share Your Success
Share Your Success
RESTART SCENARIO
NEXT SCENARIO
SC[33m)ESC[m Revert "Commit To Revert"
ENG
18:18
28.12.2020

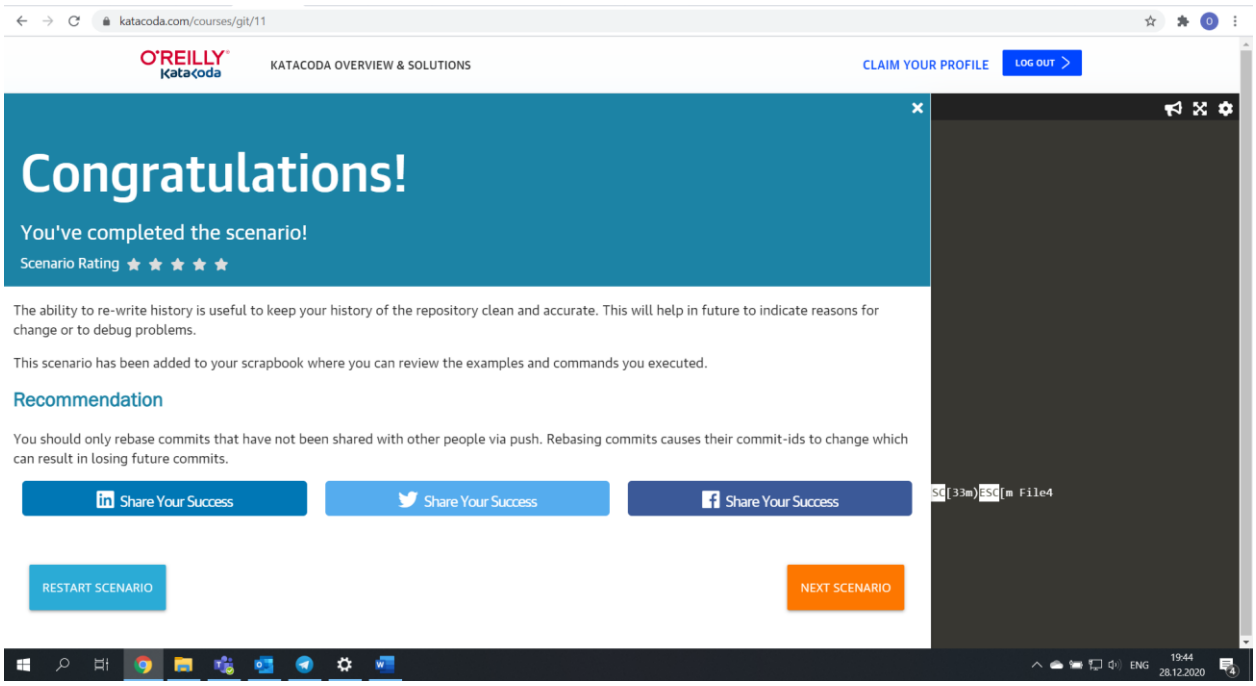

katacoda.com/courses/git/4
O'REILLY Katacoda
KATACODA OVERVIEW & SOLUTIONS
CLAIM YOUR PROFILE
LOG OUT
Congratulations!
You've completed the scenario!
Scenario Rating
This scenario demonstrated how you can reset and revert changes you've made and how to go back to a previous state.
This scenario has been added to your scrapbook where you can review the examples and commands you executed.
Share Your Success
Share Your Success
Share Your Success
RESTART SCENARIO
NEXT SCENARIO
SG[33m)ESG[m Revert "Commit To Revert"
ENG
18:18
28.12.2020

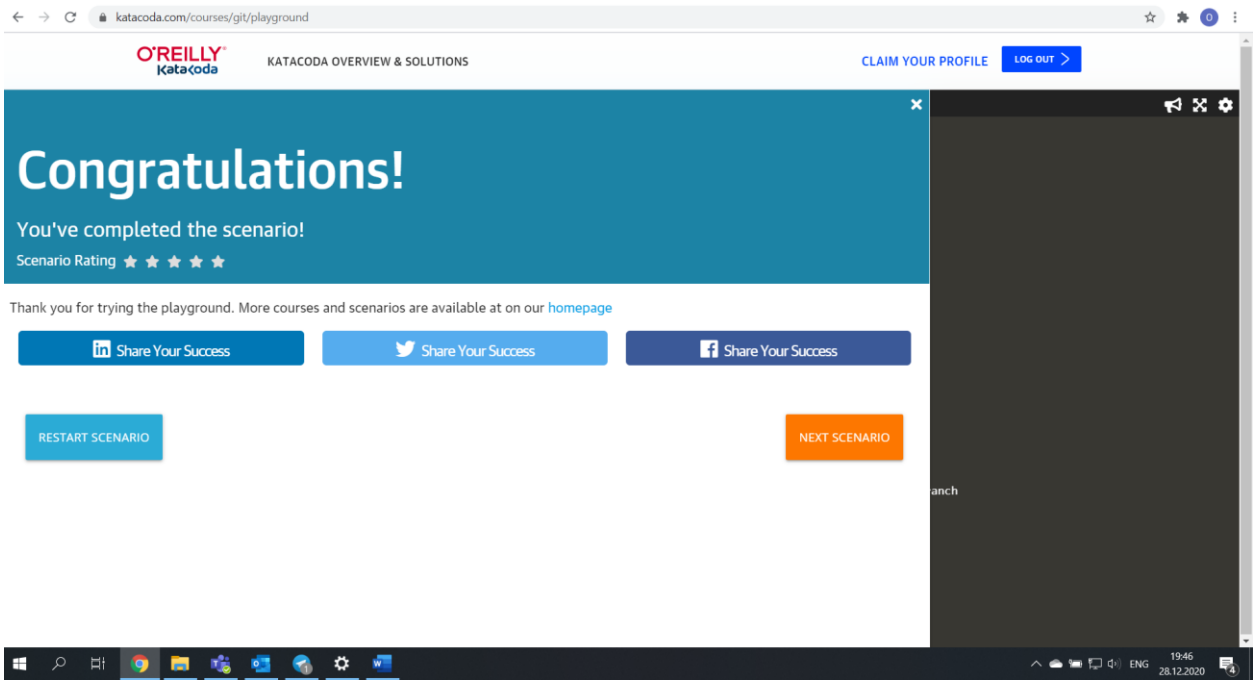

katacoda.com/courses/git/5
O'REILLY Katacoda
KATACODA OVERVIEW & SOLUTIONS
CLAIM YOUR PROFILE
LOG OUT
Congratulations!
You've completed the scenario!
Scenario Rating
In this scenario we've explored the different ways of handling merges. We've seen how to use git fetch and git merge to pull remote changes, how to resolve conflicts with other commits and finally how to keep our git log and commits clean using git rebase. Merging is an important part of Git, a topic we'll explore in more depth in future scenarios.
This scenario has been added to your scrapbook where you can review the examples and commands you executed.
Share Your Success
Share Your Success
Share Your Success
RESTART SCENARIO
NEXT SCENARIO
SG[33m)ESG[m Fix for Bug #55
x for Bug #58a
ENG
18:33
28.12.2020

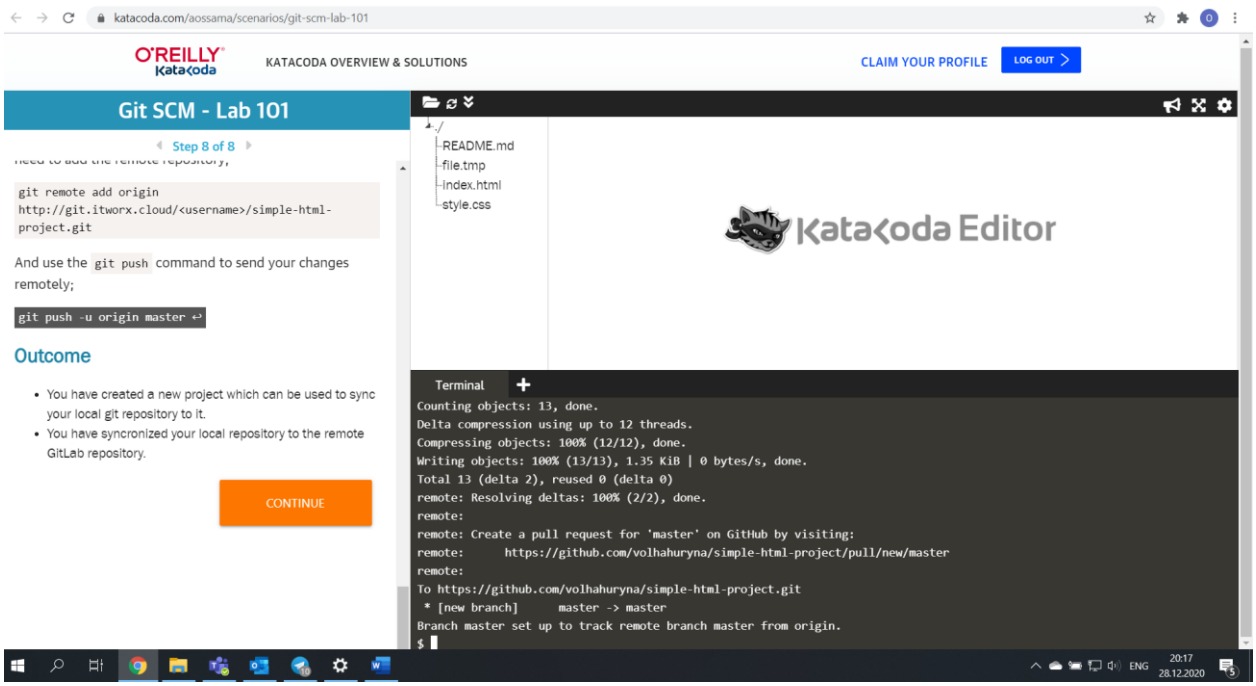

katacoda.com/courses/git/6
O'REILLY Katacoda
KATACODA OVERVIEW & SOLUTIONS
CLAIM YOUR PROFILE
LOG OUT
Congratulations!
You've completed the scenario!
Scenario Rating
In this scenario we've explored how you can work with branches which are ideal for prototyping and experiments as they can be quickly created and thrown away.
This scenario has been added to your scrapbook where you can review the examples and commands you executed.
Share Your Success
Share Your Success
Share Your Success
RESTART SCENARIO
NEXT SCENARIO
ENG
18:39
28.12.2020

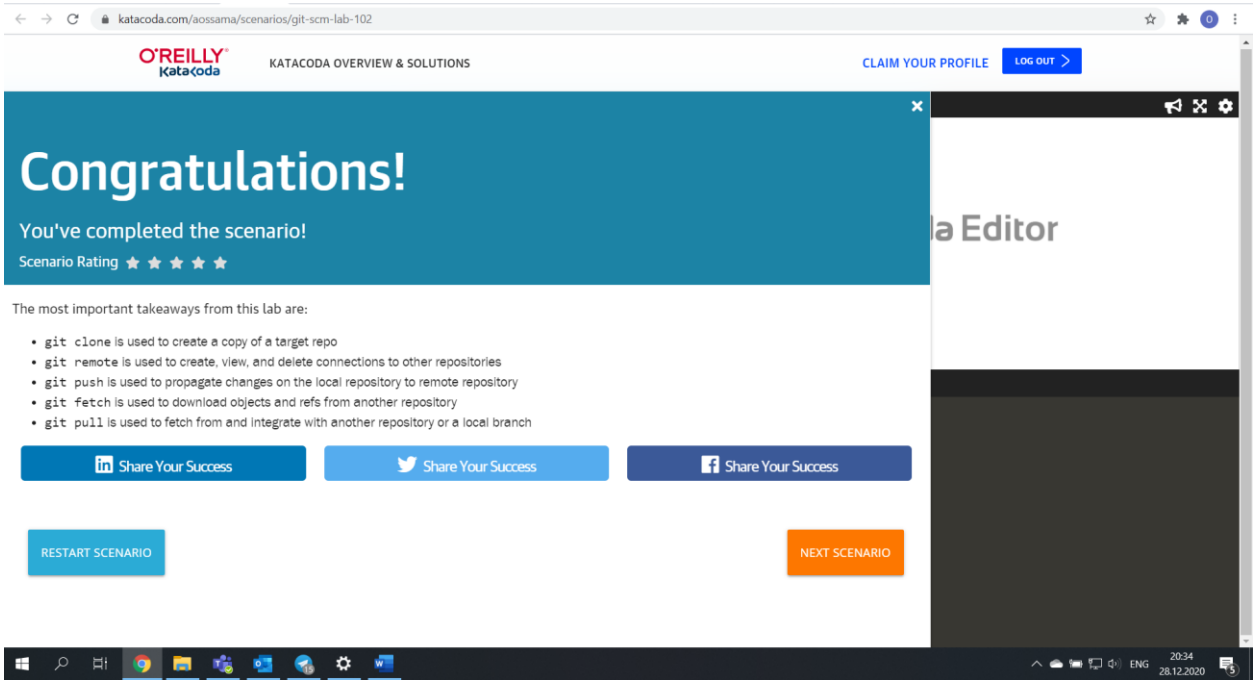

katacoda.com/courses/git/7
O'REILLY Katacoda
KATACODA OVERVIEW & SOLUTIONS
CLAIM YOUR PROFILE
LOG OUT
Congratulations!
You've completed the scenario!
Scenario Rating
With the comprehensive history of your source code within the Git repository allows you to go back in time and identify when issues occurred, such as bugs or performance issues. Commands like git diff allow you to quickly compare changes, while git bisect helps you search to identify the cause.
This scenario has been added to your scrapbook where you can review the examples and commands you executed.
Share Your Success
Share Your Success
Share Your Success
RESTART SCENARIO
NEXT SCENARIO
ENG
18:47
28.12.2020

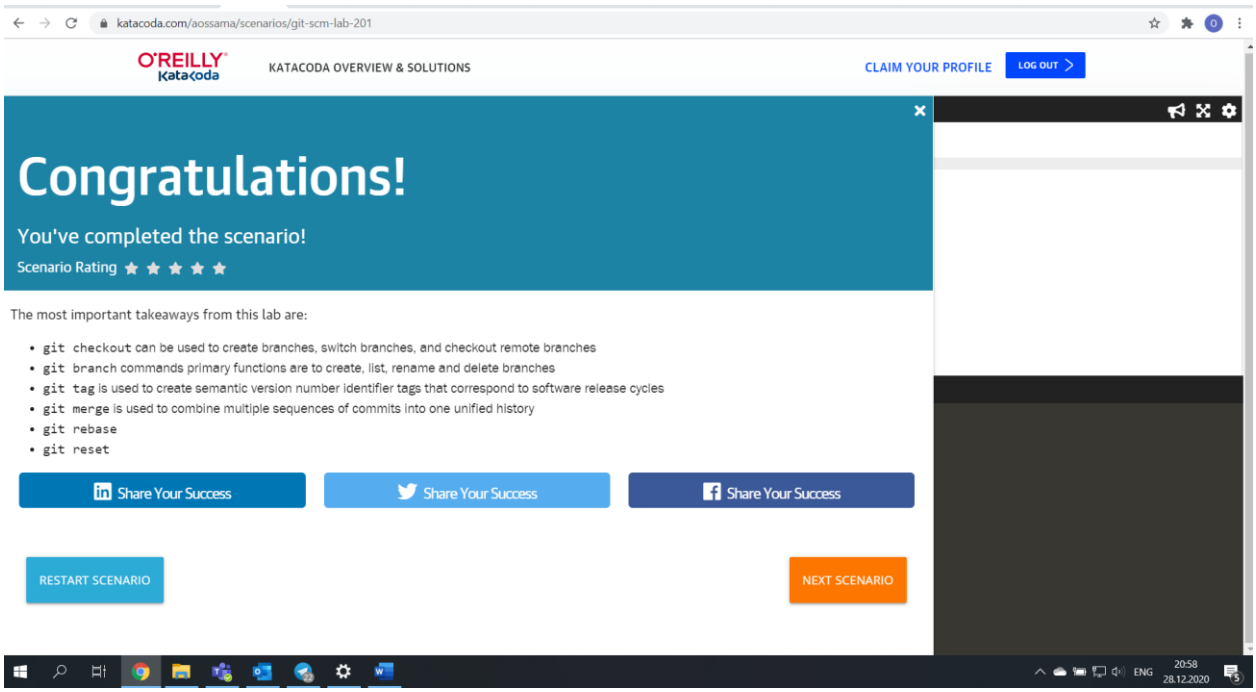

katacoda.com/courses/git/10
O'REILLY Katacoda
KATACODA OVERVIEW & SOLUTIONS
CLAIM YOUR PROFILE
LOG OUT
Congratulations!
You've completed the scenario!
Scenario Rating
When the problem arises, cherry picking is an extremely useful technique that is often forgotten. It's important to note that cherry picking only works if you keep your commits small and focused, otherwise you'll include unwanted changes. This is an important aspect to remember on a daily basis when working with Git.
This scenario has been added to your scrapbook where you can review the examples and commands you executed.
Share Your Success
Share Your Success
Share Your Success
RESTART SCENARIO
NEXT SCENARIO
ENG
18:57
28.12.2020

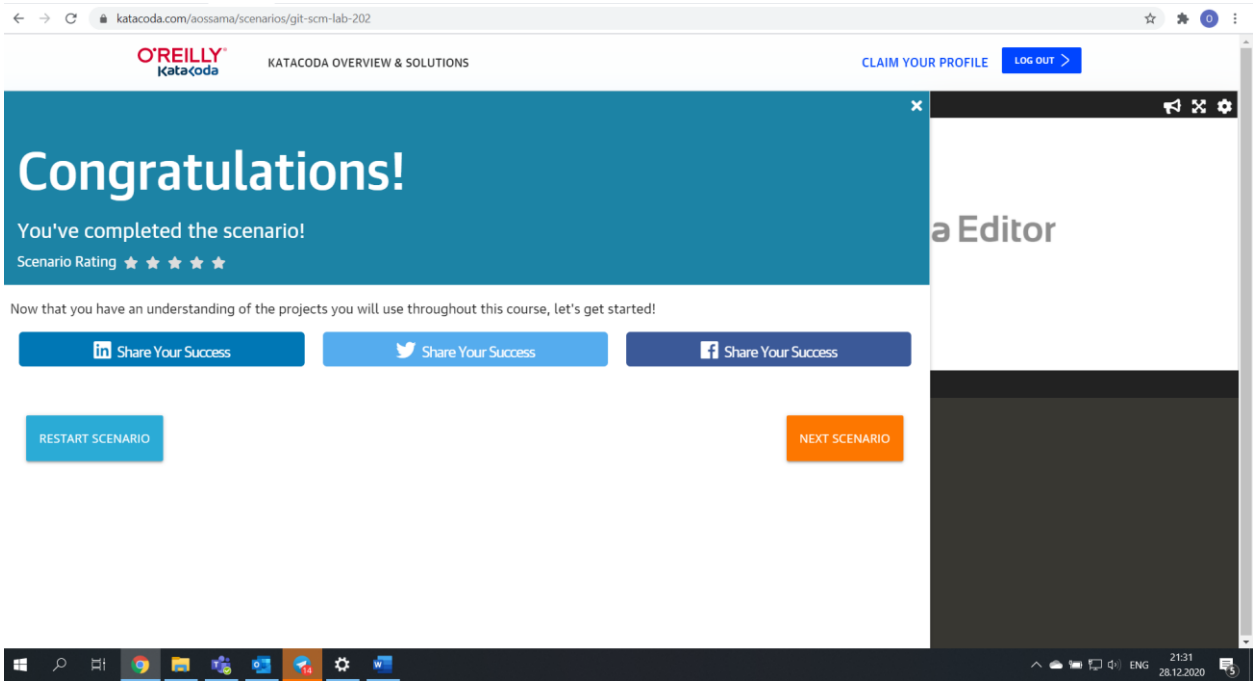

katacoda.com/courses/git/11
O'REILLY Katacoda
KATACODA OVERVIEW & SOLUTIONS
CLAIM YOUR PROFILE
LOG OUT
Congratulations!
You've completed the scenario!
Scenario Rating
The ability to re-write history is useful to keep your history of the repository clean and accurate. This will help in future to indicate reasons for change or to debug problems.
This scenario has been added to your scrapbook where you can review the examples and commands you executed.
Recommendation
You should only rebase commits that have not been shared with other people via push. Rebasing commits causes their commit-ids to change which can result in losing future commits.
Share Your Success
Share Your Success
Share Your Success
RESTART SCENARIO
NEXT SCENARIO
ENG
19:44
28.12.2020


katacoda.com/courses/git/playground
O'REILLY Katacoda
KATACODA OVERVIEW & SOLUTIONS
CLAIM YOUR PROFILE
LOG OUT
Congratulations!
You've completed the scenario!
Scenario Rating
Thank you for trying the playground. More courses and scenarios are available at on our homepage
Share Your Success
Share Your Success
Share Your Success
RESTART SCENARIO
NEXT SCENARIO


katacoda.com/aossama/scenarios/git-scm-lab-101
O'REILLY Katacoda
KATACODA OVERVIEW & SOLUTIONS
CLAIM YOUR PROFILE
LOG OUT
Git SCM - Lab 101
Step 8 of 8
git remote add origin http://git.itworx.cloud/<username>/simple-html-project.git
And use the git push command to send your changes remotely;
git push -u origin master ↩
Outcome
You have created a new project which can be used to sync your local git repository to it.
You have syncronized your local repository to the remote GitLab repository.
CONTINUE
README.md
file.tmp
index.html
style.css
Katacoda Editor
Terminal
Counting objects: 13, done.
Delta compression using up to 12 threads.
Compressing objects: 100% (12/12), done.
Writing objects: 100% (13/13), 1.35 KiB | 0 bytes/s, done.
Total 13 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), done.
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote: https://github.com/volhahuryna/simple-html-project/pull/new/master
remote:
To https://github.com/volhahuryna/simple-html-project.git
* [new branch] master -> master
Branch master set up to track remote branch master from origin.
$


katacoda.com/aossama/scenarios/git-scm-lab-102
O'REILLY Katacoda
KATACODA OVERVIEW & SOLUTIONS
CLAIM YOUR PROFILE
LOG OUT
Congratulations!
You've completed the scenario!
Scenario Rating
The most important takeaways from this lab are:
git clone is used to create a copy of a target repo
git remote is used to create, view, and delete connections to other repositories
git push is used to propagate changes on the local repository to remote repository
git fetch is used to download objects and refs from another repository
git pull is used to fetch from and integrate with another repository or a local branch
Share Your Success
Share Your Success
Share Your Success
RESTART SCENARIO
NEXT SCENARIO
ENG
20:34
28.12.2020


katacoda.com/aossama/scenarios/git-scm-lab-201
O'REILLY Katacoda
KATACODA OVERVIEW & SOLUTIONS
CLAIM YOUR PROFILE
LOG OUT
Congratulations!
You've completed the scenario!
Scenario Rating
The most important takeaways from this lab are:
git checkout can be used to create branches, switch branches, and checkout remote branches
git branch commands primary functions are to create, list, rename and delete branches
git tag is used to create semantic version number identifier tags that correspond to software release cycles
git merge is used to combine multiple sequences of commits into one unified history
git rebase
git reset
Share Your Success
Share Your Success
Share Your Success
RESTART SCENARIO
NEXT SCENARIO
ENG
20:58
28.12.2020


katacoda.com/aossama/scenarios/git-scm-lab-202
O'REILLY Katacoda
KATACODA OVERVIEW & SOLUTIONS
CLAIM YOUR PROFILE
LOG OUT
Congratulations!
You've completed the scenario!
Scenario Rating
Now that you have an understanding of the projects you will use throughout this course, let's get started!
Share Your Success
Share Your Success
Share Your Success
RESTART SCENARIO
NEXT SCENARIO
a Editor
ENG
21:31
28.12.2020