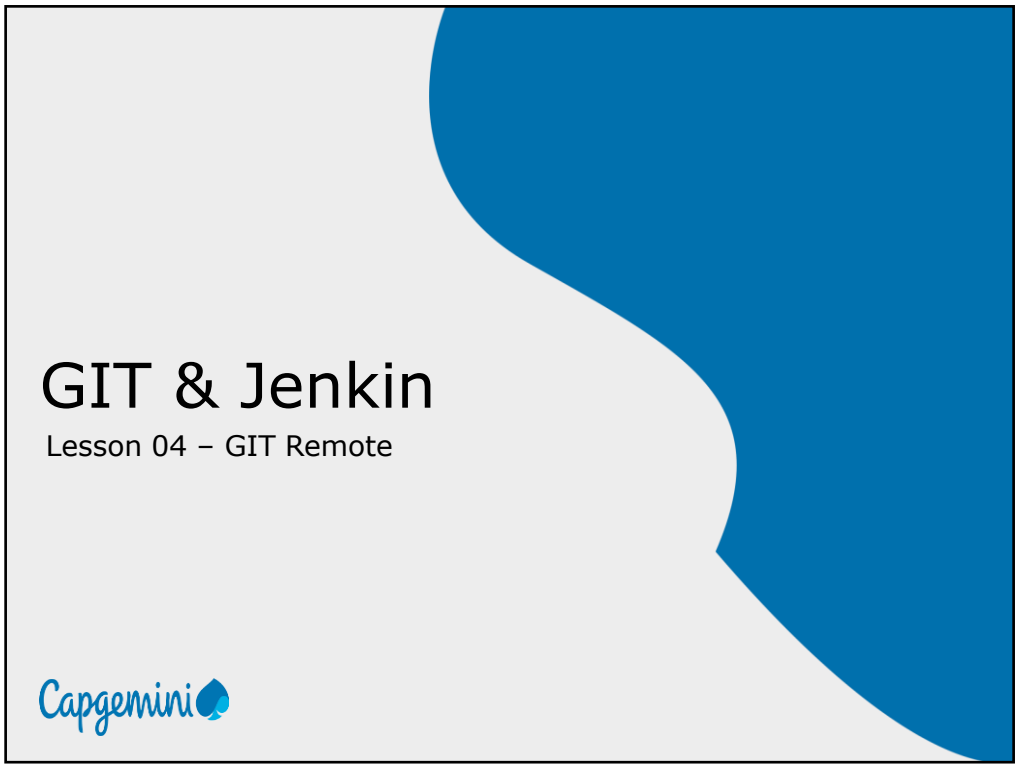

GIT & Jenkin
Lesson 04 – GIT Remote
Capgemini

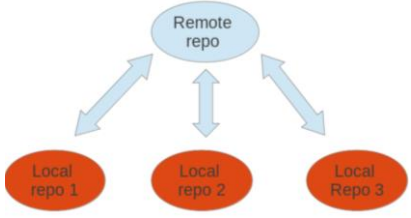
Lesson Objectives

To understand …

- Working with Remotes

GIT Remote repositories

- Remote repositories are repositories that are hosted on the Internet or network. Such remote repositories can be use to synchronize the changes of several Git repositories. A local Git repository can be connected to multiple remote repositories and you can synchronize your local repository with them via Git operations.
- It is possible that users connect their individual repositories directly, but a typically Git workflow involve one or more remote repositories which are used to synchronize the individual repositories.



GIT Remote repositories

- # create a bare repository
- git init --bare
- *# switch to the first repository*
- cd ~/repo01
- *# create a new bare repository by cloning the first one*
- git clone --bare ../remote-repository.git
- *# check the content, it is identical to the .git directory in repo01*
- ls ~/remote-repository.git
- *# Add ../remote-repository.git with the name origin*
- git remote add origin ../remote-repository.git
- *# do some changes*
- echo "I added a remote repo" > test02
- *# commit*
- git commit -a -m "This is a test for the new remote origin"
- *# to push use the command:*
- *# git push [target] # default for [target] is origin*
- git push origin

You can always connect to a remote repository if you know its URL and if you have access to it. If you clone (copy) a repository from another repository, a connection to this original repository is automatically created under the name *origin*. You can use this name to get and retrieve data from the remote repository.

Working with GIT Remote repositories

- git remote show origin *# show the details of the remote repo called origin*
- # show the existing defined remote repositories
- git remote
- # show details about the remote repos
- git remote -v
- *# Switch to home* cd ~ *# Make new directory*
- mkdir repo02
- *# Switch to new directory* cd ~/repo02 *# Clone*
- git clone ../remote-repository.git .
- *# Make some changes in the first repository* cd ~/repo01
- *# Make some changes in the file*
- echo "Hello, hello. Turn your radio on" > test01
- echo "Bye, bye. Turn your radio off" > test02
- *# Commit the changes, -a will commit changes for modified files # but will not add automatically new files*
- git commit -a -m "Some changes"
- git push ../remote-repository.git *# Push the changes*

GIT Remote repositories

- *# switch to second directory*
- cd ~/repo02
- *# pull in the latest changes of your remote repository*
- git pull
- *# make changes*
- echo "A change" > test01
- *# commit the changes*
- git commit -a -m "A change"
- *# push changes to remote repository*
- *# origin is automatically created as we cloned original from this repository*
- git push origin
- # switch to the first repository and pull in the changes
- cd ~/repo01
- git pull ../remote-repository.git/
- # check the changes
- git status

Cloning remote repositories

Git support remote operations with other Git repositories. For communication with these repositories Git supports several transport types; the native protocol for Git is also called git.

The following will clone an existing repository via the Git protocol.

 # switch to a new directory mkdir ~/online cd ~/online *# clone online repository* git clone git@github.com:vogella/gitbook.git

Alternatively you could clone the same repository via the http protocol.

 # The following will clone via HTTP git clone http://vogella@github.com/vogella/gitbook.git

Remote operations via http and a proxy

It is possible to use the HTTP protocol to clone Git repositories. This is especially helpful, if your firewall blocks everything except http.

Git also provides support for http access via a proxy server. The following Git command could, for example, clone a repository via http and a proxy. You can either set the proxy variable in general for all applications or set it only for Git.

This example uses environment variables.

 # Linux export http_proxy=http://proxy:8080 *# On Windows # Set http_proxy=http://proxy:8080* git clone http://dev.eclipse.org/git/org.eclipse.jface/org.eclipse.jface.snippets.git *# Push back to the origin using http* git push origin

This example uses the Git config settings.

// Set proxy for git globally git config --global http.proxy http://proxy:8080 // To check the proxy settings git config --get http.proxy // Just in case you need to you can also revoke the proxy settings git config --global --unset http.proxy

Summary

In this lesson, you have learnt

- GIT Remote repositories

Add the notes here.