**QUESTION 1**

What is the default text editor for the Bash shell with a Windows-based Git install?

Emacs

Vim

Notepad++

Bash

CORRECT ANSWER - Vim

**QUESTION 2**

Before you install Git, which of the following prerequisite products must be present and configured on your local OS?

Jakarta Enterprise Edition web profile compliant server

Java Development Kit 1.8 or newer

Apache Maven

Nothing

CORRECT ANSWER - Nothing

**QUESTION 3**

After you install Git and prior to issuing the first commit, which two configuration properties does the tool expect to be configured?

username and email address

username and password

email address and password

username and IP address

CORRECT ANSWER - username and email address

**QUESTION 4**

Which of the following is not a Git configuration scope?

Local

User

System

Global

CORRECT ANSWER - User

**QUESTION 5**

Who is attributed with inventing Git?

Junio C. Hamano

James Gosling

Kohsuke Kawaguchi

Linus Torvalds

CORRECT ANSWER - Linus Torvalds

**QUESTION 6**

After you initialize a new Git repository and create a file named git-quiz.html, which of the following commands will not work if issued?

git add git-quiz.html

git status

git add .

git commit -m "git quiz web file added"

CORRECT ANSWER - git commit -m "git quiz web file added"

**QUESTION 7**

Under which circumstance should you use a single dash within a Git command, as opposed to a double dash?

When using a single-character option

When using a multicharacter option

When issuing Git commands with multiple options

When referencing a Git alias

CORRECT ANSWER - When using a single-character option

## QUESTION 8

Which vendor acquired GitHub for $7.5 billion in June 2018?

Oracle

Microsoft

IBM

Google

CORRECT ANSWER - Microsoft

## QUESTION 9

Which command should you use to initialize a new Git repository?

git bash

git install

git init

git start

CORRECT ANSWER - git init

## QUESTION 10

Which file can you configure to ensure that certain file types are never committed to the local Git repository?
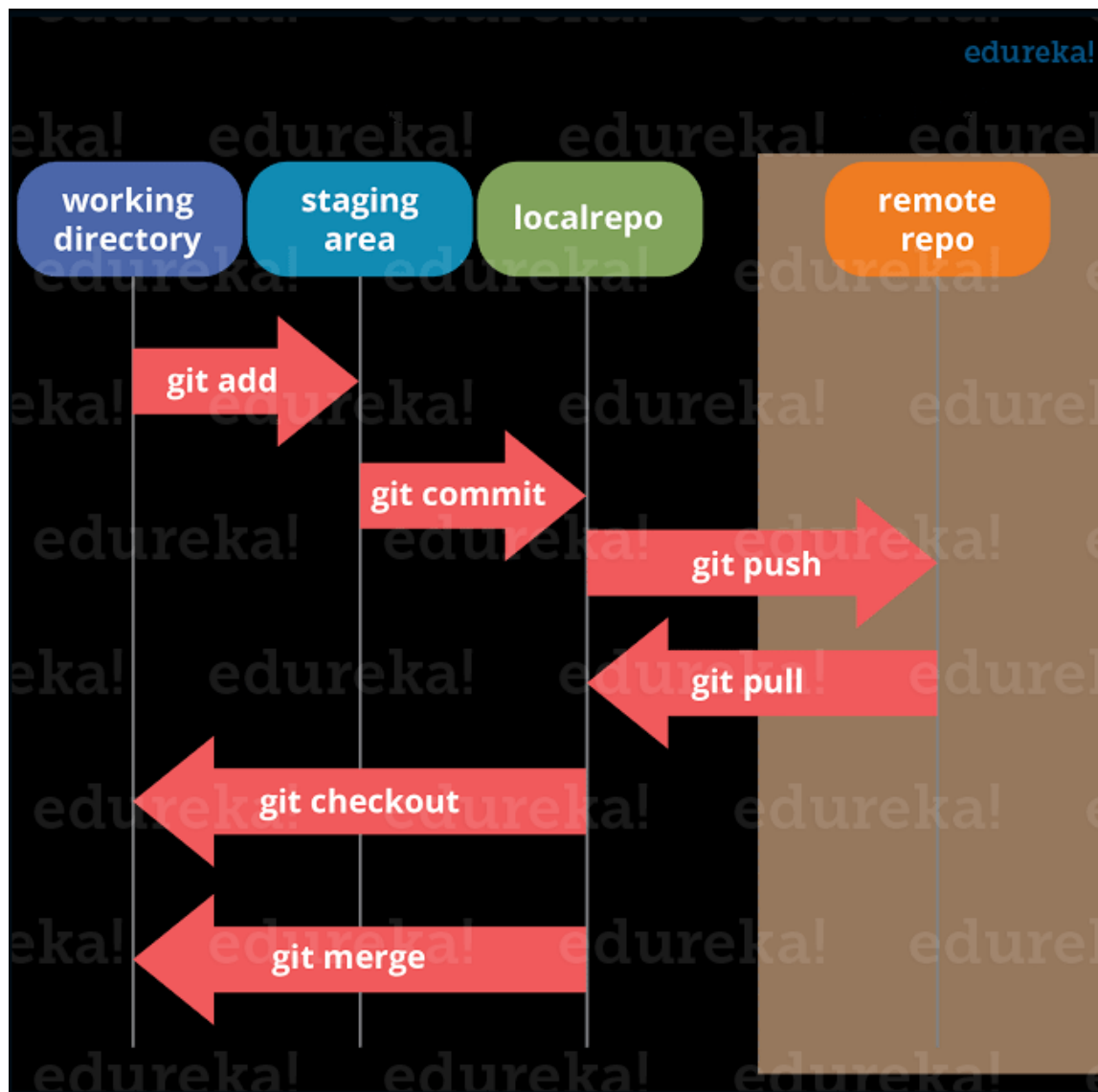
ignore.git

.gitignore

gitignore.txt

git.ignore

CORRECT ANSWER - .gitignore


# Git WorkFlow:

**Q1 . What is 'bare repository' in Git?**

A "bare" repository in Git just contains the version control information and no working files (no tree) and it doesn't contain the special .git sub-directory. Instead, it contains all the contents of the .git sub-directory directly in the main directory itself, where as working directory consist of:

A .git subdirectory with all the Git related revision history of your repo.

A working tree, or checked out copies of your project files.

**Q2. What language is used in Git?**

Git uses 'C' language. GIT is fast, and 'C' language makes this possible by reducing the overhead of run times associated with high level languages.

**Q3. In Git how do you revert a commit that has already been pushed and made public?**

Remove or fix the bad file in a new commit and push it to the remote repository. This is the most natural way to fix an error. Once you have made necessary changes to the file, commit it to the remote repository for that I will use

git commit -m "commit message"

Create a new commit that undoes all changes that were made in the bad commit.to do this I will use a command

git revert <name of bad commit>

**Q4. What is the difference between git pull and git fetch?**

Git pull command pulls new changes or commits from a particular branch from your central repository and updates your target branch in your local repository.

Git fetch is also used for the same purpose but it works in a slightly different way. When you perform a git fetch, it pulls all new commits from the desired branch and stores it in a new branch in your local repository. If you want to reflect these changes in your target branch, git fetch must be followed with a git merge. Your target branch will only be updated after merging the target branch and fetched branch. Just to make it easy for you, remember the equation below:

Git pull = git fetch + git merge

**Q5. What is 'staging area' or 'index' in Git?**

That before completing the commits, it can be formatted and reviewed in an intermediate area known as 'Staging Area' or 'Index'. From the diagram it is evident that every change is first verified in the staging area I have termed it as "stage file" and then that change is committed to the repository.

Staging Area - Git Interview Questions - Edureka

Course Curriculum

DevOps Certification Training

Instructor-led SessionsReal-life Case StudiesAssignmentsLifetime Access

If your interviewer has good knowledge on Git he/she will dig in deep, so the next set of Git interview questions will be more challenging.

**Q6. What is Git stash?**

Often, when you've been working on part of your project, things are in a messy state and you want to switch branches for sometime to work on something else. The problem is, you don't want to do a commit of half-done work just so you can get back to this point later. The answer to this issue is Git stash.

Stashing takes your working directory that is, your modified tracked files and staged changes and saves it on a stack of unfinished changes that you can reapply at any time.

**Q7. What is Git stash drop**

Git 'stash drop' command is used to remove the stashed item. It will remove the last added stash item by default, and it can also remove a specific item if you include it as an argument.

If you want to remove a particular stash item from the list of stashed items you can use the below commands:

git stash list: It will display the list of stashed items like:

stash@{0}: WIP on master: 049d078 added the index file

stash@{1}: WIP on master: c264051 Revert "added file_size"

stash@{2}: WIP on master: 21d80a5 added number to log

If you want to remove an item named stash@{0} use command git stash drop stash@{0}.

**Q8.How do you find a list of files that has changed in a particular commit?**

To get a list files that has changed in a particular commit use the below command:

git diff-tree -r {hash}

Given the commit hash, this will list all the files that were changed or added in that commit. The -r flag makes the command list individual files, rather than collapsing them into root directory ames only.

The output will also include some extra information, which can be easily suppressed by including two

flags:

git diff-tree –no-commit-id –name-only -r {hash}

Here –no-commit-id will suppress the commit hashes from appearing in the output, and –name-only will only print the file names, instead of their paths.

**Q9 What is the function of 'git config'?**

First tell why we need 'git config'.

Git uses your username to associate commits with an identity. The git config command can be used to change your Git configuration, including your username.

Suppose you want to give a username and email id to associate commit with an identity so that you can know who has made a particular commit. For that I will use:

git config –global user.name "Your Name": This command will add username.

git config –global user.email "Your E-mail Address": This command will add email id.

**Q10.hat does commit object contains?**

Course Curriculum

DevOps Certification Training

A set of files, representing the state of a project at a given point of time

Reference to parent commit objects

An SHAI name, a 40 character string that uniquely identifies the commit object.