**1. Опишіть для чого використовують git, які основні дії та команди в ньому виконують.**  
**Git** is a distributed version control system that tracks changes in any set of computer files, usually used for coordinating work among programmers who are collaboratively developing source code during software development. Its goals include speed, data integrity, and support for distributed, non-linear workflows.  
  
**-Distributed system:** Instead of relying on a central server, Git stores a complete copy of the project history on each developer's machine. This enables offline work and easier collaboration.  
**- Version control:** Git tracks changes made to files over time, allowing you to revert to older versions, compare different versions, and see who made what changes.  
**- Speed and efficiency:** Git is designed for fast performance, even with large projects.  
- Data integrity: Git uses checksums to ensure that data is not corrupted.  
**- Non-linear workflows:** Git supports branching and merging, which allows developers to work on different parts of a project independently and then merge their changes together.  
  
**Storing and tracking changes**:  
-Git stores "snapshots" of your project as commits. Each commit contains information about the changes made to the files, as well as the time and author of those changes.  
-You can easily browse the history of the project to see how it has evolved over time.  
-Reverting to previous versions: Git allows you to go back to any previous version of the project, if necessary.

**Collaboration:**  
-Distributed system: Git allows developers to collaborate on a project, even if they are not on the same network.  
-Branches: You can create branches to work on different features or experiments, without risking damaging the main codebase.  
-Merging: Git makes it easy to merge changes from different branches, which allows for collaborative coding without conflicts.

**Efficiency:**  
-Speed: Git is optimized for fast operation, even with large projects.  
-Local storage: Git stores a full copy of the project history on your computer, allowing you to work offline.  
-Flexibility: Git offers a wide range of commands and tools to automate tasks and optimize the workflow.

**Security:**  
-Data integrity: Git uses checksums to ensure that data is not corrupted.  
-Data recovery: Git allows you to recover lost or damaged files.  
-Version history: Thanks to the version history, you can track who made what changes to the code, which can be useful in case of errors or security issues.  
  
**Additional features**:  
-Script support: Git allows you to automate routine tasks using scripts.  
-Visualization: There are many tools for visualizing Git history and code changes, which can help you better understand the project.  
-Integration with other tools: Git easily integrates with other software development tools, such as IDEs, bug tracking systems, and CI/CD.