DevOps Git Session-09 Notes

1. Fast-Forward Merge

Concept: Happens when the base branch has no new commits since the feature branch diverged.

Example:

main → commit-1 → newbranch (commit-a1 → a2 → a3 → a4)

Since there are no commits on main, merging newbranch simply moves the pointer forward.

Command: git merge newbranch (fast-forward occurs automatically if possible)

2. Restore, Reset, Revert

Restore (Undo local changes):

- git restore <filename>: Discards changes from working tree.

- git restore --staged <filename>: Moves from staging area back to working tree.

Reset (Undo commits locally):

- git reset --soft HEAD~1: Moves commit to staging area.

- git reset --mixed HEAD~1: Moves commit to working directory (default).

- git reset --hard HEAD~1: Discards completely.

Revert (Undo commits safely):

- Keeps history intact (used for public/shared branches).

- git revert <commit-id>

3. Git Workflow Stages

Working Tree → Staging Area → Local Repo → Remote Repo

Commands:

- git status: Check state.

- git add . / git add -A: Stage changes.

- git commit -m 'message': Save to local repo.

- git push: Upload to remote.

4. Basic Commands Flow

git status

git branch

git checkout -b newbranch

vi <file>

git add .

git commit -m 'message'

git log

git push

git checkout main

git merge newbranch

5. Untracked vs Tracked Files

Untracked: Created but not added to git (git status shows in red).

Use git add to track them.

6. Log & History

git log: Shows history.

git log --oneline: Simplified view.