# Version Control using Git

## Project

**Scenario:**

You are working on a project with a team of developers. The project repository has a

main branch called main and a development branch called dev. You have been assigned

a task to add a new feature to the project.

**Tasks:**

1. Clone the Repository

2. Create a Feature Branch

3. Implement the Feature

4. Commit Your Changes

5. Push the Feature Branch

6. Create a Pull Request

7. Review and Merge

8. Update Local Branches

9. Resolve Conflicts (if any) and Push the Final Changes

Here's a **step-by-step guide** to complete your Git project on AWS Linux

**1. Clone the Repository**

git clone https://github.com/your-org/repo-name.git

cd repo-name

* Replace the URL with your actual repository link.
* Verify files: ls -l

**2. Create a Feature Branch**

git checkout dev *# Switch to dev branch*

git pull origin dev *# Get latest changes*

git checkout -b feature/add-payment *# Create your feature branch*

* **Naming tip**: Use feature/your-feature-name (e.g., feature/add-payment).

**3. Implement the Feature**

* Edit/add files using nano or vim:

nano payment.html *# Example: Create a new payment page*

* Save changes (Ctrl+O, Enter, Ctrl+X in nano).

**4. Commit Your Changes**

git status *# View changes*

git add payment.html *# Stage specific file (or use `git add .`)*

git commit -m "Added payment page with basic form"

**5. Push the Feature Branch**

git push -u origin feature/add-payment

* -u links your local branch to the remote branch.

**6. Create a Pull Request (PR)**

**On GitHub/GitLab**:

1. Go to your repository → **Pull Requests** → **New PR**.
2. Set:
   * **Base branch**: dev
   * **Compare branch**: feature/add-payment
3. Add a title/description → **Create PR**.

**7. Review and Merge**

* Teammates review your PR.
* Address feedback (if any) by pushing more commits to the same branch.
* Once approved, merge into dev.

**8. Update Local Branches**

git checkout dev *# Switch back to dev*

git pull origin dev *# Sync with merged changes*

**9. Resolve Conflicts (If Any)**

If conflicts occur during PR:

git checkout feature/add-payment

git fetch origin *# Get latest changes*

git rebase origin/dev *# Update your branch*

* **Fix conflicts** in files marked with <<<<<<<, =======, >>>>>>>.
* Continue rebase:

git add .

git rebase --continue

git push --force *# Update remote branch*

**Documentation (Required in Project)**

For each step, record:

1. **Commands used** (copy from above).
2. **Terminal output** (e.g., git status screenshots).
3. **PR screenshots** (GitHub PR page, merge confirmation).

**AWS Linux-Specific Tips**

1. **Generate SSH Key for GitHub** (avoid HTTPS auth):

ssh-keygen -t ed25519 -C "your-email@example.com"

cat ~/.ssh/id\_rsa.pub *# Copy to GitHub SSH settings*

1. **Install GitHub CLI** (for PRs from terminal):

sudo yum install gh -y *# Amazon Linux*

gh auth login *# Follow prompts*

gh pr create --fill *# Create PR interactively*

1. **Cleanup** after merge:

git branch -d feature/add-payment *# Delete local branch*

git push origin --delete feature/add-payment *# Delete remote*

**Troubleshooting**

* **Permission denied (publickey)**?  
  Ensure your SSH key is added to GitHub:

eval "$(ssh-agent -s)"

ssh-add ~/.ssh/id\_rsa

* **Conflict resolution failed**?  
  Abort rebase and retry:

git rebase --abort