Git, AWS EC2, and CI/CD Pipeline Documentation

1. Git Setup and Commands

A. Prerequisites
- A GitHub account
- Git Bash installed
- Internet connection

B. Steps to Set Up a Git Repository
1. Create a GitHub account at GitHub.
2. Create a new repository and configure name, visibility, and initialization.
3. Clone repository using **git clone <repo_url>**.

C. Git Bash Steps
1. Install Git Bash.
2. Navigate to repo directory using **cd <folder>**.
3. List files using **ls**.
4. Add files: **git add .**
5. Commit changes: **git commit -m 'message'**
6. Push to GitHub: **git push**

2. AWS EC2 Instance Setup

1. Create AWS account at AWS.
2. Navigate to EC2 service.
3. Click **Launch Instance**.
4. Name the instance.
5. Choose OS (Ubuntu, Windows, etc.).
6. Select instance type (e.g., t2.micro).
7. Configure network & enable public IP.
8. Add key pair (.pem file).
9. Launch and connect via SSH:
ssh -i "key.pem" ubuntu@public_ip

3. CI/CD Pipeline Setup (GitHub Actions + AWS EC2)

A CI/CD pipeline automates deployment from GitHub to EC2 using GitHub Actions.

Steps to Configure:
1. Open your GitHub repo → Settings → Secrets and Variables → Actions.
2. Add secrets:
- EC2_HOST → Public IPv4 address
- EC2_USER → Usually ubuntu or ec2-user
- EC2_KEY → Private SSH key
3. Create **.github/workflows/deploy.yml** with:

```
name: CI/CD Pipeline on: push: branches: - main jobs: deploy: runs-on: ubuntu-latest steps: - name: Checkout code uses: actions/checkout@v3 - name: Connect to EC2 and deploy uses: appleboy/ssh-action@v0.1.10 with: host: ${{ secrets.EC2_HOST }} username: ${{ secrets.EC2_USER }} key: ${{ secrets.EC2_KEY }} script: | cd /var/www/html git pull origin main
```

Summary

Section	Purpose
Git	Manage and push code from local to remote repository
AWS EC2	Host your application/server in the cloud
CI/CD	Automate deployment using GitHub Actions