

1. Create a Git Flow Workflow Architecture on Git

Git Flow is a branching model for Git that defines a strict branching structure for projects. Here's how to set it up:

1. **Master:** The stable branch that holds the production-ready code.
2. **Develop:** The integration branch where features are merged before going to the master.
3. **Feature:** A branch created for a specific feature development.
4. **Release:** A branch created to prepare a new production release.
5. **Hotfix:** A branch created for urgent fixes that need to go directly to production.

2. Create All the Required Branches

Start by initializing a Git repository and creating the develop branch.

```
mkdir git_project
```

```
cd git_project
```

```
git init
```

```
# Create the develop branch
```

```
git checkout -b develop
```

```
# Create a feature branch from develop
```

```
git checkout -b feature/some-feature develop
```

```
# Push develop to remote (optional)
```

```
git push origin develop
```

3. Starting from the Feature Branch, Push the Branch to the Master, Following the Architecture

1. Develop your feature in the feature/some-feature branch:

```
echo "Feature implementation" > feature.txt
```

```
git add feature.txt
```

```
git commit -m "Add feature.txt with feature implementation"
```
2. Merge the feature branch into develop:

```
git checkout develop
```

```
git merge --no-ff feature/some-feature
```

```
git commit -m "Merge feature/some-feature into develop"
```

3. Create a release branch from develop and merge it into master:

```
git checkout -b release/v1.0 develop
```

```
# (Optional) Final tweaks in the release branch
```

```
git commit -am "Final tweaks for release v1.0"
```

```
# Merge the release branch into master
```

```
git checkout master
```

```
git merge --no-ff release/v1.0
```

```
git commit -m "Release v1.0"
```

```
# Tag the release
```

```
git tag -a v1.0 -m "Release v1.0"
```

```
# Push the master branch and the tag
```

```
git push origin master
```

```
git push origin v1.0
```

4. Merge the release/v1.0 branch back into develop:

```
git checkout develop
```

```
git merge --no-ff release/v1.0
```

```
git commit -m "Merge release/v1.0 into develop"
```

```
git branch -d release/v1.0
```

4. Push an urgent.txt on Master Using Hotfix

1. Create a hotfix branch from master:

```
git checkout master
```

```
git checkout -b hotfix/urgent-fix
```

2. Make your urgent fix in the hotfix/urgent-fix branch:

```
echo "Urgent fix content" > urgent.txt
```

```
git add urgent.txt
```

```
git commit -m "Add urgent.txt with urgent fix"
```

3. Merge the hotfix branch into master:

```
git checkout master
```

```
git merge --no-ff hotfix/urgent-fix
```

```
git commit -m "Merge hotfix/urgent-fix into master"
```

```
git push origin master
```

4. Merge the hotfix back into develop:

```
git checkout develop
```

```
git merge --no-ff hotfix/urgent-fix
```

```
git commit -m "Merge hotfix/urgent-fix into develop"
```

```
git branch -d hotfix/urgent-fix
```

```
git push origin develop
```

Summary of Git Commands:

- **mkdir git_project**
- **cd git_project**
- **git init**

- **# Create branches**
- **git checkout -b develop**
- **git checkout -b feature/some-feature develop**

- **# Work on feature branch**
- **echo "Feature implementation" > feature.txt**
- **git add feature.txt**
- **git commit -m "Add feature.txt with feature implementation"**

- **# Merge feature into develop**
- **git checkout develop**
- **git merge --no-ff feature/some-feature**
- **git commit -m "Merge feature/some-feature into develop"**

- **# Create release branch and merge to master**
- **git checkout -b release/v1.0 develop**
- **git commit -am "Final tweaks for release v1.0"**
- **git checkout master**
- **git merge --no-ff release/v1.0**
- **git commit -m "Release v1.0"**
- **git tag -a v1.0 -m "Release v1.0"**
- **git push origin master**
- **git push origin v1.0**

- **# Merge release branch back into develop**
- **git checkout develop**
- **git merge --no-ff release/v1.0**
- **git branch -d release/v1.0**

- **# Create hotfix branch and apply fix**
- **git checkout master**
- **git checkout -b hotfix/urgent-fix**
- **echo "Urgent fix content" > urgent.txt**
- **git add urgent.txt**
- **git commit -m "Add urgent.txt with urgent fix"**
- **git checkout master**
- **git merge --no-ff hotfix/urgent-fix**
- **git commit -m "Merge hotfix/urgent-fix into master"**
- **git push origin master**

- **# Merge hotfix into develop**
- **git checkout develop**
- **git merge --no-ff hotfix/urgent-fix**
- **git commit -m "Merge hotfix/urgent-fix into develop"**
- **git branch -d hotfix/urgent-fix**
- **git push origin develop**