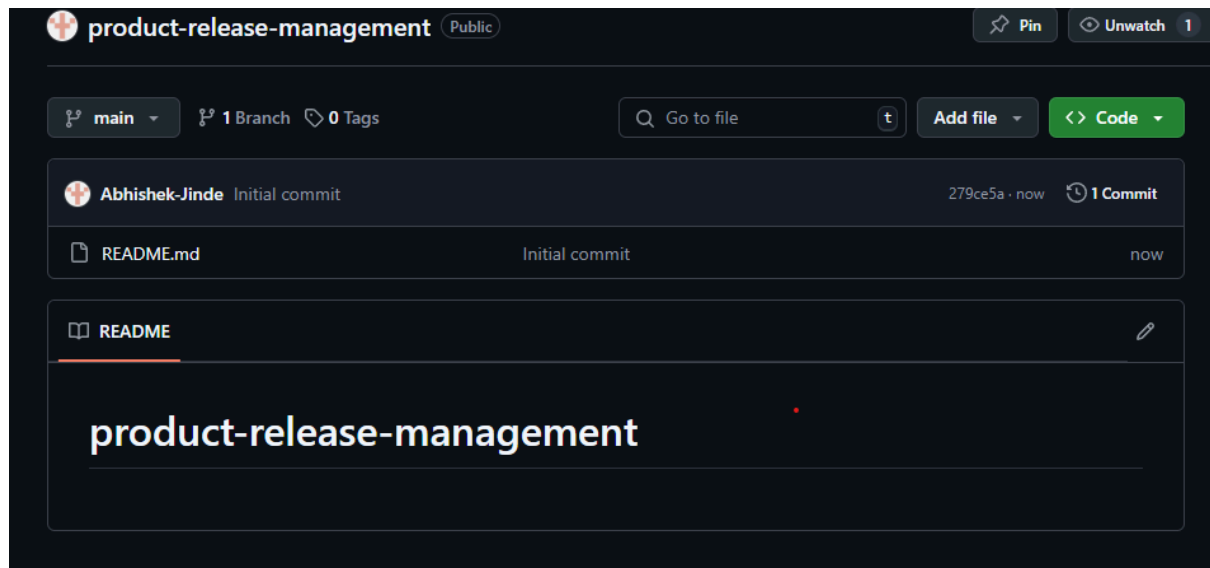**You work as a DevOps Architect in Zendrix Softwares. The company has been struggling to manage their product releases. The releases should happen on the 25th of every month. Suggest a Git workflow architecture for this requirement. Simulate this workflow by creating pseudo code files and branches and upload the same to your GitHub account. As a part of the solution, share the link to your GitHub repository**

Solution : Simplifying Release Management: Automating Git Workflow for Zendrix Softwares' Monthly Releases

Step 1: Set Up the Repository

Create a new repository called `product-release-management` on GitHub with an initial `README.md` and `.gitignore`. You can do this on your GitHub account.

Step 2: Clone the Repository Locally

```
git clone
https://github.com/your-username/product-release-management.git
cd product-release-management
```

```
boxuser@Ubuntu:~$ git clone https://github.com/Abhishek-Jinde/product-release-m
nagement.git
loning into 'product-release-management'...
emote: Enumerating objects: 3, done.
emote: Counting objects: 100% (3/3), done.        .
emote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
eceiving objects: 100% (3/3), done.
boxuser@Ubuntu:~$
```

Step 3: Create the Main Branches (`master` and `develop`)

Create the `develop` branch:

```
git checkout -b develop
```

```
git push -u origin develop
```

```
vboxuser@Ubuntu:~/product-release-management$ git checkout -b develop
Switched to a new branch 'develop'

vboxuser@Ubuntu:~/product-release-management$ git push -u origin develop
Username for 'https://github.com': Abhishek-Jinde
Password for 'https://Abhishek-Jinde@github.com':
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'develop' on GitHub by visiting:
remote:      https://github.com/Abhishek-Jinde/product-release-management/pull/n
ew/develop
remote:
To https://github.com/Abhishek-Jinde/product-release-management.git
 * [new branch]      develop -> develop
branch 'develop' set up to track 'origin/develop'.
vboxuser@Ubuntu:~/product-release-management$
```

Ensure that the `main` branch is the default production branch:
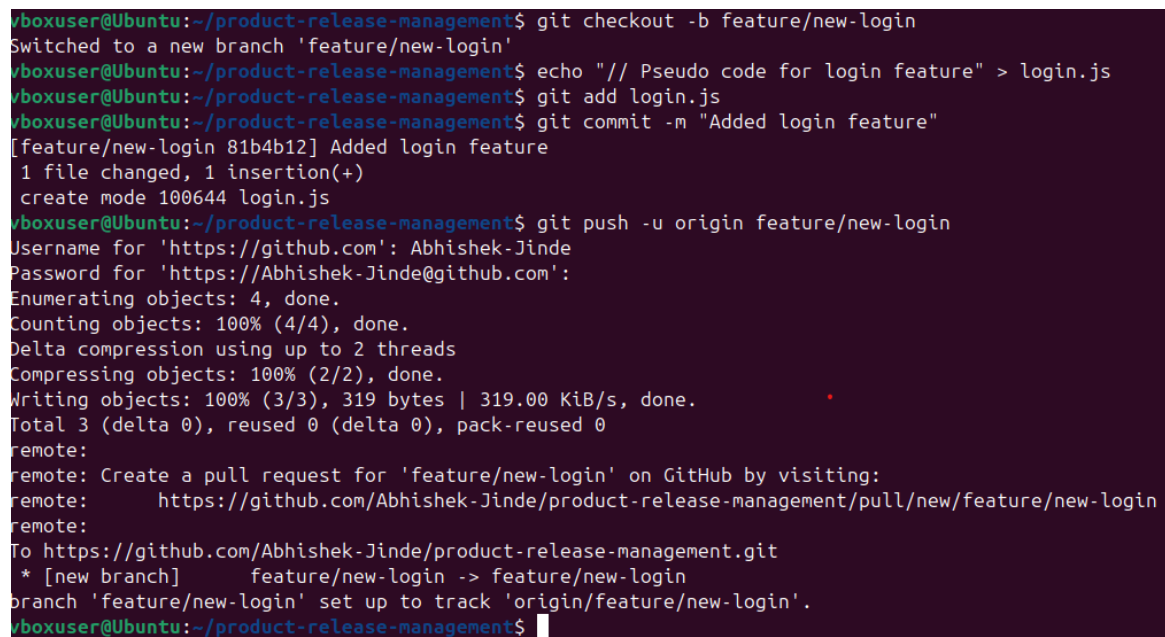
```
git checkout main
```

```
vboxuser@Ubuntu:~/product-release-management$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
vboxuser@Ubuntu:~/product-release-management$
```

Step 4: Simulate Feature Development (`feature/*`)

Create two feature branches (`feature/new-login` and `feature/new-signup`):

For `feature/new-login`:

```
git checkout -b feature/new-login
echo "// Pseudo code for login feature" > login.js
git add login.js
git commit -m "Added login feature"
git push -u origin feature/new-login
```

```
vboxuser@Ubuntu:~/product-release-management$ git checkout -b feature/new-login
Switched to a new branch 'feature/new-login'
vboxuser@Ubuntu:~/product-release-management$ echo "// Pseudo code for login feature" > login.js
vboxuser@Ubuntu:~/product-release-management$ git add login.js
vboxuser@Ubuntu:~/product-release-management$ git commit -m "Added login feature"
[feature/new-login 81b4b12] Added login feature
 1 file changed, 1 insertion(+)
 create mode 100644 login.js
vboxuser@Ubuntu:~/product-release-management$ git push -u origin feature/new-login
Username for 'https://github.com': Abhishek-Jinde
Password for 'https://Abhishek-Jinde@github.com':
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 2 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 319 bytes | 319.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'feature/new-login' on GitHub by visiting:
remote:      https://github.com/Abhishek-Jinde/product-release-management/pull/new/feature/new-login
remote:
To https://github.com/Abhishek-Jinde/product-release-management.git
 * [new branch]      feature/new-login -> feature/new-login
branch 'feature/new-login' set up to track 'origin/feature/new-login'.
vboxuser@Ubuntu:~/product-release-management$
```

For `feature/new-signup`:

```
git checkout develop
git checkout -b feature/new-signup
echo "// Pseudo code for signup feature" > signup.js
git add signup.js
git commit -m "Added signup feature"
git push -u origin feature/new-signup
```

```
vboxuser@Ubuntu:~/product-release-management$ git checkout develop
Switched to branch 'develop'
Your branch is up to date with 'origin/develop'.
vboxuser@Ubuntu:~/product-release-management$ git checkout -b feature/new-signup
Switched to a new branch 'feature/new-signup'
vboxuser@Ubuntu:~/product-release-management$ echo "// Pseudo code for signup feature" > signup.js
vboxuser@Ubuntu:~/product-release-management$ git add signup.js
vboxuser@Ubuntu:~/product-release-management$ git commit -m "Added signup feature"
[feature/new-signup 8f5331f] Added signup feature
 1 file changed, 1 insertion(+)
 create mode 100644 signup.js
vboxuser@Ubuntu:~/product-release-management$ git push origin -u feature/new-signup
Username for 'https://github.com': Abhishek-Jinde
Password for 'https://Abhishek-Jinde@github.com':
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 2 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 323 bytes | 323.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'feature/new-signup' on GitHub by visiting:
remote:      https://github.com/Abhishek-Jinde/product-release-management/pull/new/feature/new-signup
remote:
To https://github.com/Abhishek-Jinde/product-release-management.git
 * [new branch]      feature/new-signup -> feature/new-signup
branch 'feature/new-signup' set up to track 'origin/feature/new-signup'.
vboxuser@Ubuntu:~/product-release-management$ 
```

Step 5: Merge Features into `develop`

After feature development is complete, merge them into the `develop` branch.

Switch to the `develop` branch:
```
git checkout develop
```

Merge each feature branch into `develop`:
```
git merge feature/new-login
git merge feature/new-signup
```

Push the changes to the remote `develop` branch:
```
git push origin develop
```

```
vboxuser@Ubuntu:~/product-release-management$ git checkout develop
Switched to branch 'develop'
Your branch is up to date with 'origin/develop'.
vboxuser@Ubuntu:~/product-release-management$ git merge feature/new-login
Updating 279ce5a..81b4b12
Fast-forward
 login.js | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 login.js
vboxuser@Ubuntu:~/product-release-management$ git merge feature/new-signup
Merge made by the 'ort' strategy.
 signup.js | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 signup.js
vboxuser@Ubuntu:~/product-release-management$
```

Step 6: Prepare for Release (`release/*`)

Create a `release` branch from `develop` to prepare the release:

Create a release branch (e.g., `release/v1.0.0`):
```
git checkout -b release/v1.0.0
git push -u origin release/v1.0.0
```

Add a pseudo bug fix or feature stabilization code:
```
echo "// Fix for release" > fix.js
git add fix.js
git commit -m "Bug fix for release"
git push origin release/v1.0.0
```

```
vboxuser@Ubuntu:~/product-release-management$ git checkout develop
Already on 'develop'
Your branch is ahead of 'origin/develop' by 3 commits.
  (use "git push" to publish your local commits)
vboxuser@Ubuntu:~/product-release-management$ git checkout -b release/v1.0.0
Switched to a new branch 'release/v1.0.0'
vboxuser@Ubuntu:~/product-release-management$ git push -u origin release/v1.0.0
Username for 'https://github.com': Abhishek-Jinde
Password for 'https://Abhishek-Jinde@github.com':
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 2 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 358 bytes | 358.00 KiB/s, done.
Total 2 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'release/v1.0.0' on GitHub by visiting:
remote:      https://github.com/Abhishek-Jinde/product-release-management/pull/new/release/v1.0.0
remote:
To https://github.com/Abhishek-Jinde/product-release-management.git
 * [new branch]      release/v1.0.0 -> release/v1.0.0
branch 'release/v1.0.0' set up to track 'origin/release/v1.0.0'.
vboxuser@Ubuntu:~/product-release-management$ echo "Fix for release" > fix.js
vboxuser@Ubuntu:~/product-release-management$ git add fix.js
vboxuser@Ubuntu:~/product-release-management$ git commit -m "Bug fix for release"
[release/v1.0.0 7e52060] Bug fix for release
 1 file changed, 1 insertion(+)
 create mode 100644 fix.js
vboxuser@Ubuntu:~/product-release-management$ git push -u origin release/v1.0.0
Username for 'https://github.com': Abhishek-Jinde
Password for 'https://Abhishek-Jinde@github.com':
Enumerating objects: 4, done.
```

Step 7: Merge `release` into `master` on the 25th of the Month

When it's time to release on the 25th, merge the `release` branch into `master` and tag the release.

Switch to the `main` branch:
```
git checkout main
```

Merge the `release/v1.0.0` branch:
```
git merge release/v1.0.0
```

Tag the release:
```
git tag -a v1.0.0 -m "Release v1.0.0"
git push origin main --tags
```

```
vboxuser@Ubuntu:~/product-release-management$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
vboxuser@Ubuntu:~/product-release-management$ git merge release/v1.0.0
Updating 279ce5a..7e52060
Fast-forward
 fix.js    | 1 +
 login.js  | 1 +
 signup.js | 1 +
 3 files changed, 3 insertions(+)
 create mode 100644 fix.js
 create mode 100644 login.js
 create mode 100644 signup.js
vboxuser@Ubuntu:~/product-release-management$ git tag -a v1.0.0 -m "Release v1.0.0"
vboxuser@Ubuntu:~/product-release-management$ git push origin main --tags
Username for 'https://github.com': Abhishek-Jinde
Password for 'https://Abhishek-Jinde@github.com':          .
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 169 bytes | 169.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Abhishek-Jinde/product-release-management.git
   279ce5a..7e52060  main -> main
 * [new tag]         v1.0.0 -> v1.0.0
vboxuser@Ubuntu:~/product-release-management$ █
```

Step 8: Merge `release` Back into `develop`

To ensure that all bug fixes are included in future development, merge the `release` branch back into `develop`:

```
git checkout develop
git merge release/v1.0.0
git push origin develop
```

```
vboxuser@Ubuntu:~/product-release-management$ git checkout develop
Switched to branch 'develop'
Your branch is ahead of 'origin/develop' by 3 commits.
  (use "git push" to publish your local commits)
vboxuser@Ubuntu:~/product-release-management$ git merge release/v1.0.0
Updating 3effce4..7e52060
Fast-forward
 fix.js | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 fix.js
vboxuser@Ubuntu:~/product-release-management$ git push origin develop
Username for 'https://github.com': Abhishek-Jinde
Password for 'https://Abhishek-Jinde@github.com':
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Abhishek-Jinde/product-release-management.git
   279ce5a..7e52060  develop -> develop
vboxuser@Ubuntu:~/product-release-management$
```
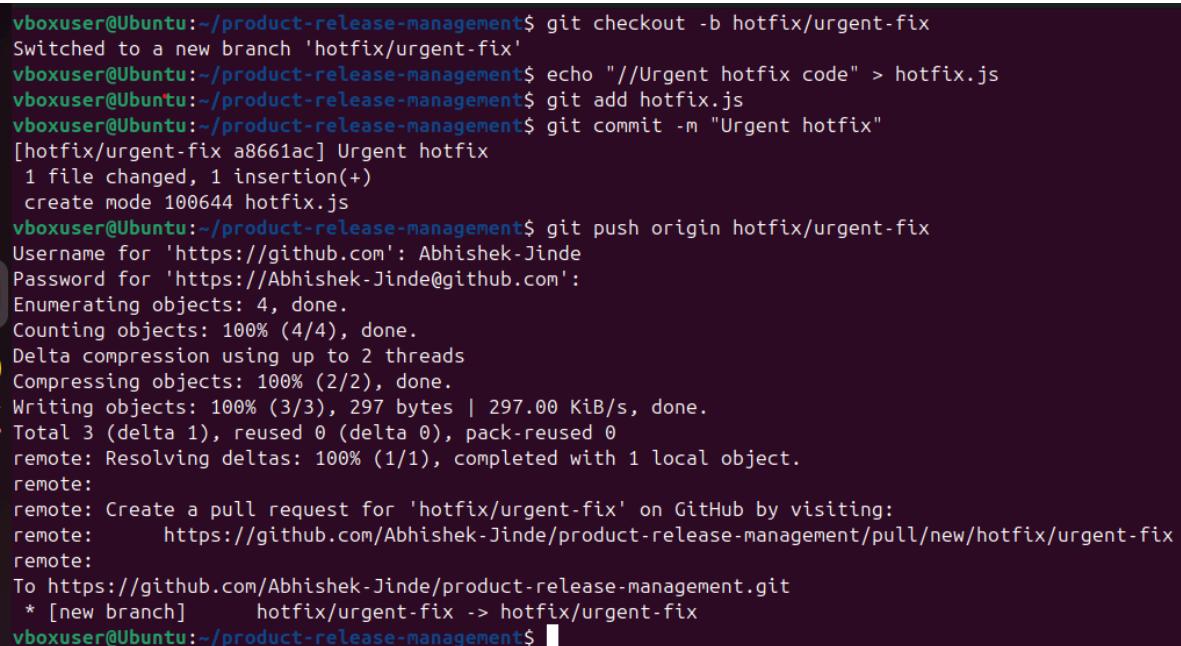
Step 9: Simulate a Hotfix (`hotfix/*`)

If a critical bug is found in production, create a `hotfix` branch:

Create a hotfix branch from `main`:

```
git checkout main
git checkout -b hotfix/urgent-fix
echo "// Urgent hotfix code" > hotfix.js
git add hotfix.js
git commit -m "Urgent hotfix"
git push origin hotfix/urgent-fix
```


After the hotfix is tested, merge it into `main` and `develop`:

```
git checkout main
git merge hotfix/urgent-fix
git push origin main

git checkout develop
git merge hotfix/urgent-fix
git push origin develop
```

```
vboxuser@Ubuntu:~/product-release-management$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
vboxuser@Ubuntu:~/product-release-management$ git merge hotfix/urgent-fix
Updating 7e52060..a8661ac
Fast-forward
 hotfix.js | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 hotfix.js
vboxuser@Ubuntu:~/product-release-management$ git push origin main
Username for 'https://github.com': Abhishek-Jinde
Password for 'https://Abhishek-Jinde@github.com':
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Abhishek-Jinde/product-release-management.git
   7e52060..a8661ac  main -> main
vboxuser@Ubuntu:~/product-release-management$
```

```
vboxuser@Ubuntu:~/product-release-management$ git checkout develop
Switched to branch 'develop'
Your branch is up to date with 'origin/develop'.
vboxuser@Ubuntu:~/product-release-management$ git merge hotfix/urgent-fix
Updating 7e52060..a8661ac
Fast-forward
 hotfix.js | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 hotfix.js
vboxuser@Ubuntu:~/product-release-management$ git push origin develop
Username for 'https://github.com': Abhishek-Jinde
Password for 'https://Abhishek-Jinde@github.com':
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Abhishek-Jinde/product-release-management.git
   7e52060..a8661ac  develop -> develop
vboxuser@Ubuntu:~/product-release-management$
```

Step 10: Push All Changes to GitHub

Ensure that all branches and tags are pushed to the GitHub repository:

```
git push --all
```

```
vboxuser@Ubuntu:~/product-release-management$ git push --all
Username for 'https://github.com': Abhishek-Jinde
Password for 'https://Abhishek-Jinde@github.com':
Everything up-to-date
vboxuser@Ubuntu:~/product-release-management$
```

product-release-management Public

📌 Pin   👁 Unwatch 1

🔀 main ▾   🔀 6 Branches   🏷 1 Tags

🔍 Go to file   t   Add file ▾   <> Code ▾

Switch branches/tags   ✕

🔍 Find or create a branch...

Branches   Tags

✓ main   default

develop

feature/new-login   •

feature/new-signup

hotfix/urgent-fix

release/v1.0.0

View all branches

a8661ac · 5 minutes ago   🕐 6 Commits

Initial commit	36 minutes ago
Bug fix for release	14 minutes ago
Urgent hotfix	5 minutes ago
Added login feature	23 minutes ago
Added signup feature	21 minutes ago

✏️

product-release-management