Git Workflow: main/dev vs origin/main/origin/dev

Step 1 — Clone the repo

```
git clone https://github.com/user/repo.git
```

Result:

```
(main)          A---B---C
(dev)           A---B---C---D
(origin/main)   A---B---C
(origin/dev)    A---B---C---D
```

Git stores these pointers inside .git/:

```
.git/
  refs/
    heads/
      main        ← your local main branch pointer
      dev         ← your local dev branch pointer
    remotes/
      origin/
        main      ← remote tracking branch for main
        dev       ← remote tracking branch for dev
```

Step 2 — Make changes in dev

```
git checkout dev
# edit files
git add file1 file2
git commit -m "Dev changes"
```

Result:

```
(dev)          A---B---C---D---E
(origin/dev)   A---B---C---D
```

dev is ahead of origin/dev by 1 commit.

Step 3 — Push dev changes

```
git push origin dev
```

Result:

```
(dev)         A---B---C---D---E
(origin/dev)  A---B---C---D---E
```

Both dev branches are now in sync.

Step 4 — Merge dev into main locally

```
git checkout main
git merge dev
```

Result:

```
(main)          A---B---C-------M
                 \             /
                  D---E----
(dev)           A---B---C---D---E
(origin/main)  A---B---C
(origin/dev)   A---B---C---D---E
```

Here, M is the merge commit.

Step 5 — Push merged main

```
git push origin main
```

Result:

```
(main)          A---B---C-------M
                  \             /
                   D---E---
(dev)           A---B---C---D---E
(origin/main)  A---B---C-------M
                 \             /
                  D---E----
(origin/dev)   A---B---C---D---E
```

Now remote main is updated with the merge.

Step 6 — Creating a merge request locally (simulation)

You can prepare changes locally before pushing for a remote Merge Request:

```
git checkout main
git merge dev
# test, review locally
git push origin main
```

On platforms like GitHub/GitLab/Bitbucket, you would then open a **Pull Request** or **Merge Request** for `dev` → `main`. Locally, the merge is already done; the MR is for review and approval on the remote.

✅ **Key Recap:**

- Local `main` and `dev` have their own history.
- Remote tracking branches `origin/main` and `origin/dev` update via **fetch**.
- **Push** sends your local commits to the corresponding remote branch.
- **Merge** combines histories locally; push afterward to update remote.
- Merge requests are remote-side review processes; local merges can be pushed directly if review isn't needed.