Branching Strategy for ROSS Application

Bitbucket

Hotfix branch — Merge directly into MASTER branch → MASTER Branch — Using MASTER branch we are deploying code to prod environment → Ross Application Prod environment

Merging Release branch into master

Release branch — Using Release branch we are deploying code to Preprod environment → Ross Application Pre-Prod environment

Creating Release branch from dev branch

DEV branch — Using DEV branch we are deploying code to QA environment → Ross Application QA environment

Dev branch — Using Dev branch we are deploying code to Dev environment → Ross Application DEV environment
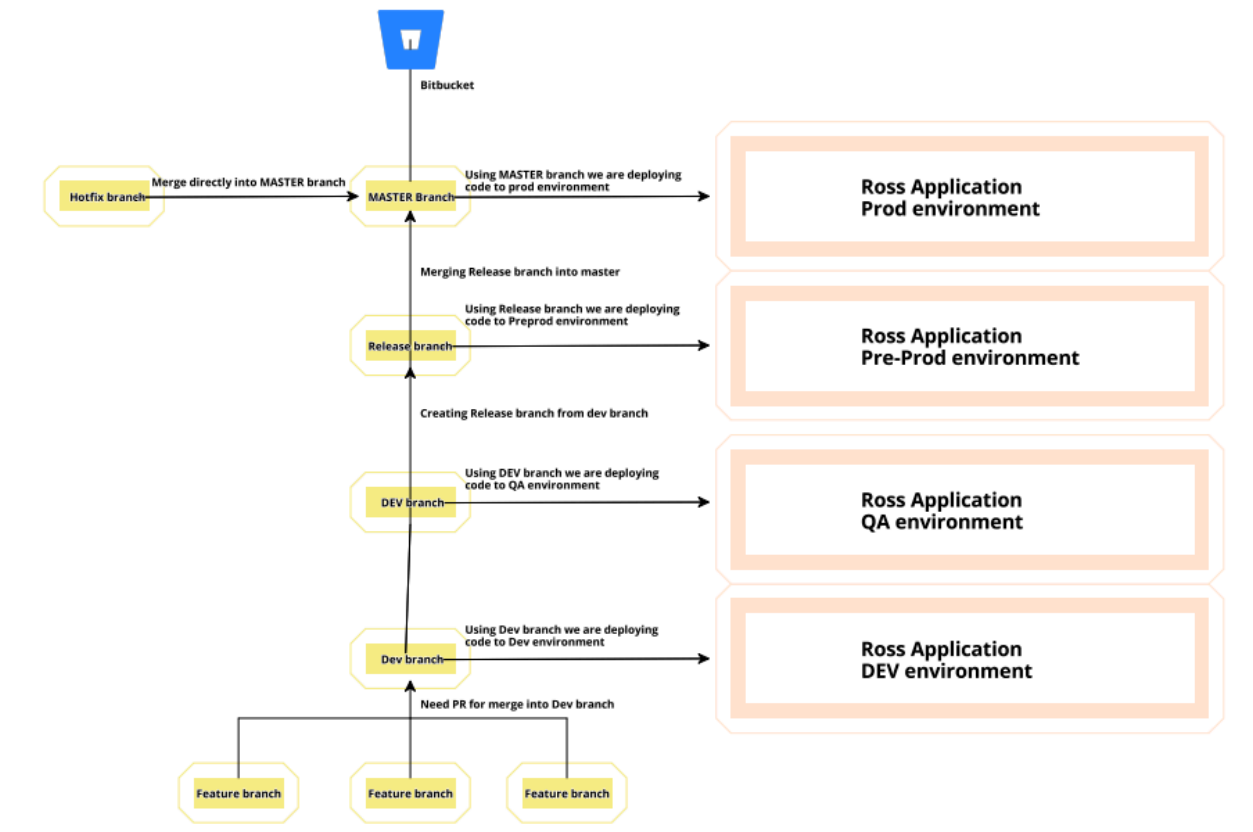
Need PR for merge into Dev branch

Feature branch   Feature branch   Feature branch

Branch Types

Main Branch

- **Purpose:** Production deployment
- **Naming Convention:** `main` or `master`
- **Deployment:** Manually triggered as per production deployment schedule
- **Approval:** Requires approver's approval for deployment to production

Environment-Specific Branches

- **Release Branch**
  - **Purpose:** Pre-production deployment
  - **Naming Convention:** `release/{version}`

- ○ **Deployment:** Automatically triggers pre-prod deployment pipeline
- ● **Dev Branch**
  - ○ **Purpose:** Development and QA
  - ○ **Naming Convention:** dev
  - ○ **Deployment:** Automatically triggers Dev and QA environment pipelines

Hotfix and Bugfix Branches

- ● **Hotfix Branch**
  - ○ **Purpose:** Fix critical issues in production
  - ○ **Naming Convention:** hotfix/{issue-id}
  - ○ **Deployment:** Merges back into main and dev branches after fixes
- ● **Bugfix Branch**
  - ○ **Purpose:** Fix bugs found during QA
  - ○ **Naming Convention:** bugfix/{issue-id}
  - ○ **Deployment:** Merges back into dev branch after fixes

Workflow

1. **Feature Development**
   - ○ Developers create feature branches from the dev branch.
   - ○ Naming Convention: feature/{story-id}
   - ○ Once development is complete, a pull request (PR) is created to merge the feature branch into the dev branch.
   - ○ PR approval and passing tests (Snyk and SonarCloud) are required before merging.
2. **Development Environment Deployment**
   - ○ After merging a feature branch into the dev branch, the Dev environment pipeline is triggered automatically.
   - ○ Deployment occurs in the Dev environment for initial testing by developers.
3. **QA Environment Deployment**
   - ○ After successful Dev testing, the pipeline approval triggers deployment to the QA environment.
   - ○ QA tests the changes; if issues are found, a bugfix branch is created from the dev branch.
   - ○ Fixes are merged back into the dev branch, and the cycle repeats until QA approval.
4. **Pre-Production Deployment**
   - ○ Once QA approves, a release branch is created from the dev branch.
   - ○ The release branch is deployed to the pre-prod environment.
5. **Production Deployment**

- After successful pre-prod testing, the `release` branch is merged into the `main` branch.
- The production pipeline is manually triggered according to the deployment schedule.
- An additional approval layer ensures no deployment without approver's consent.

6. **Hotfix Deployment**
   - For critical issues in production, a `hotfix` branch is created from the `main` branch.
   - Fixes are applied and merged back into both the `main` and `dev` branches.