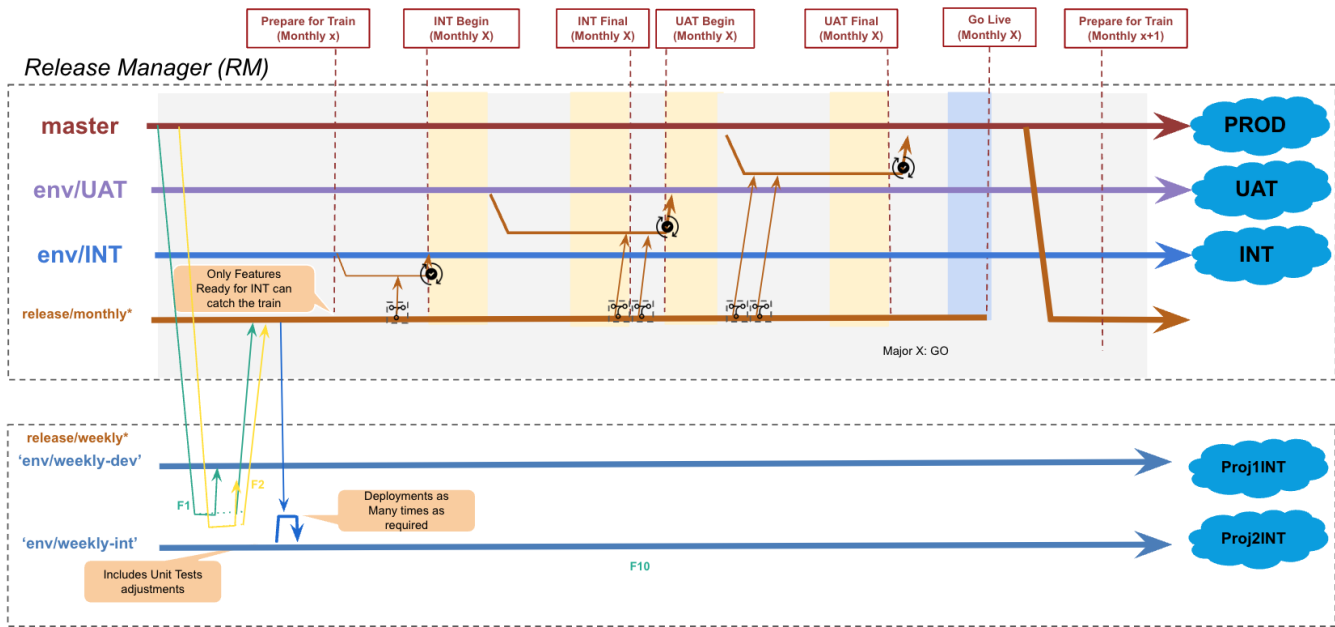


Gitlab Branching Way of Working - Major (Monthly)

Overview

New Flow figure will be updated soon:



Steps

- **[RM]** Creates a new branch `release/monthly_<releaseName>_<releaseNumber>` after the previous major or minor/bugfix release has been deployed to Production (see conventions in "Branching Definitions & Naming Conventions").
- **[dev]** Develops the change in the `majordev1` sandbox.
- **[dev]** Creates a new `feature/*` branch from `master` (per "Branching Definitions & Naming Conventions").
- **[dev]** Opens a Merge Request (MR) from `feature/*` to `majorint` and deploys the changes.
- **[dev]** Once validated in `majorint`, updates the Jira user story to "Ready for INT", assigns it to the Release Manager, and provides required details (e.g., Fix Version).
- **[dev]** Opens an MR from `feature/*` to the target release branch.
- **[Tech lead / Tech lead backup]** Reviews and approves the MR if it meets the required quality checks.
- **[RM]** Creates the INT validation MR (`<releaseName> - Validation to INT`) (one-time per release).
- **[RM]** After successful INT validation, merges the release branch into the `INT` branch.
- **[RM]** Coordinates component deployments and supports resolution of unit test issues in INT.
- **[RM]** Update Jira status to "IN INT" and assign the jiras to testing team
- **[PO]** After successful INT testing, updates the Jira user story to "Ready for UAT", assigns it to the Release Manager, and provides required details (e.g., Fix Version). *Note: UAT validation and deployment occur only after the minor/bugfix release has been promoted from UAT to Production.*
- **[RM]** Creates the UAT validation MR (`<releaseName> - Validation to UAT`) (one-time per release).
- **[RM]** Merges the release branch into the `UAT` branch.
- **[RM]** Coordinates component deployments and supports resolution of unit test issues in UAT.
- **[RM]** Update Jira status to "IN UAT" and assign the jiras to testing team
- **[PO]** After successful UAT testing, updates the Jira user story to "Ready for Release", assigns it to the Release Manager, and provides required details (e.g., Fix Version).
- **[RM]** Creates the Production validation MR (`<releaseName> - Validation to PROD`).
- **[RM]** Merges the release branch into `master`, which triggers validation again.
- **[RM]** Initiates the Quick Deploy and completes the required post-deployment activities.
- **[RM]** Coordinates component deployments and supports resolution of unit test issues in PRD.
- **[RM]** Update Jira status to "Released" and assign the jiras to testing team
- **[RM]** Communicates the deployment completion and confirmation status to all SF team members.
- **[RM]** Performs the retrofit/back promotion of the changes to `minordev` and `bugfix` branches, as applicable.