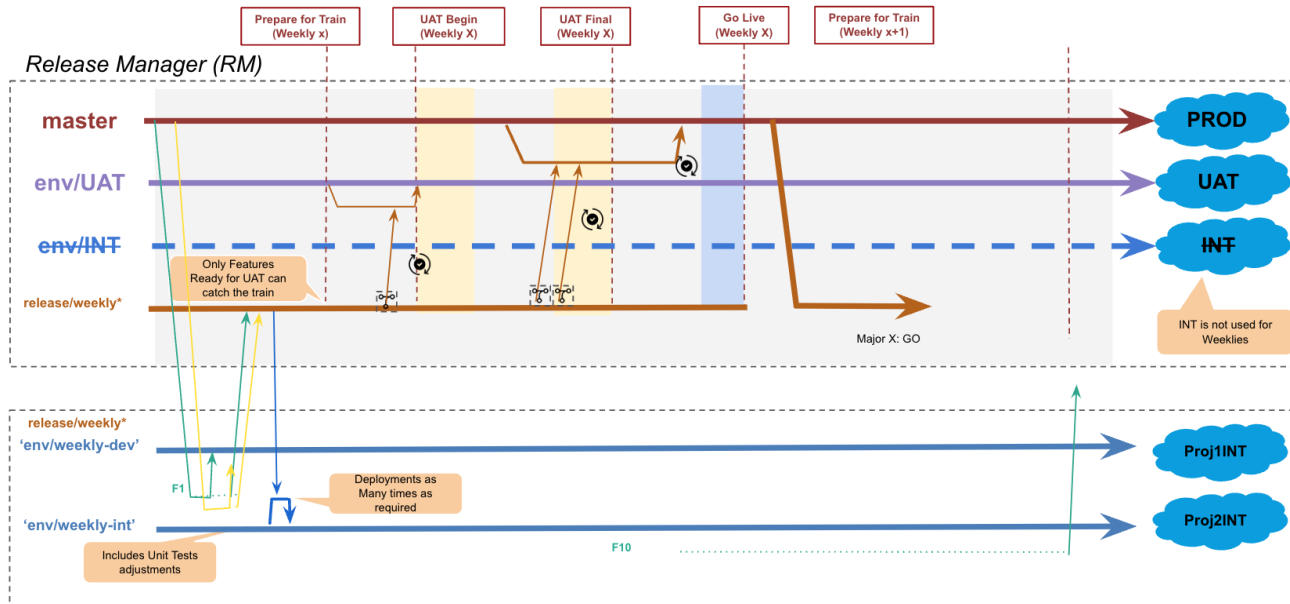


Gitlab Branching Way of Working - Minor release

Overview

New Flow figure will be updated soon:



Steps

Note

- At this time, we do not create a release branch for minor releases.
- The RM team deploys minor updates on an as-needed basis. We do not use a single consolidated feature branch; for example, if there are five stories, each change is validated through its own merge request and deployed independently. This is aligned with our current deployment cadence.

Minor / Bugfix process

- **[dev]** Develops the change in the `minor` sandbox.
- **[dev]** Creates a `minor/*` branch from `master` (per "Branching Definitions & Naming Conventions").
- **[dev]** Creates an MR from `minor/*` to the target `env/uat` branch.
- **[Tech Lead / Tech Lead backup]** Reviews and approves the MR if it meets the required checks.
- **[dev]** Updates the Jira user story to "Ready for UAT" (skipping "Ready for INT"), assigns it to the Release Manager, and provides required details (e.g., Fix Version).
- **[RM]** After successful validation of the MR, merges the `minor/*` branch into the `UAT` branch.
- **[RM]** Coordinates component deployment and supports resolution of unit test issues in UAT.
- **[RM]** Updates the Jira status to "IN UAT" and assigns the Jira items to the 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).
- **[dev]** Creates an MR from `minor/*` to the target `master` branch.
- **[RM]** After successful validation of the MR, merges the `minor/*` branch into `master`.
- **[RM]** Coordinates component deployment and supports resolution of unit test issues in PRD.
- **[RM]** Updates the Jira status to "Released" and assigns the Jira items to the testing team.