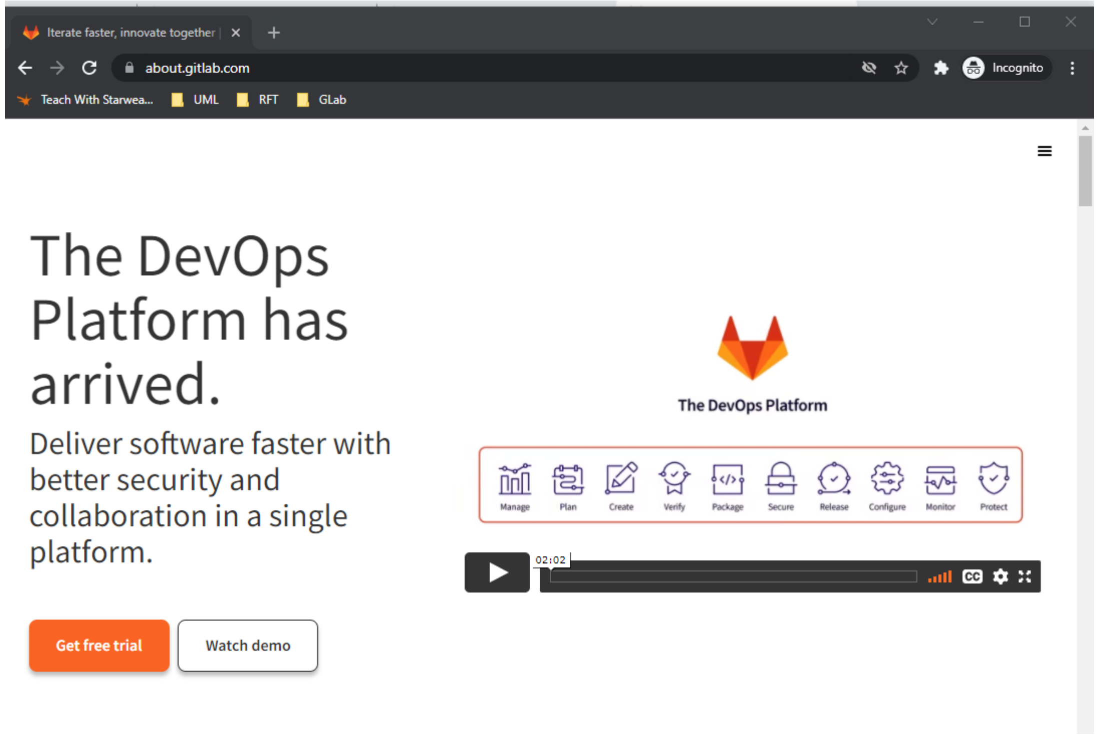
CICD with Jenkins

Module Two: Introduction to CI

GitLab/GitHub Overview

- Primary focus in this module is the CI workflow
  - The repository function of GitLab/GitHub
- Repository events are the usual starting points for CICD pipelines
- We will look at pipelines in the next module

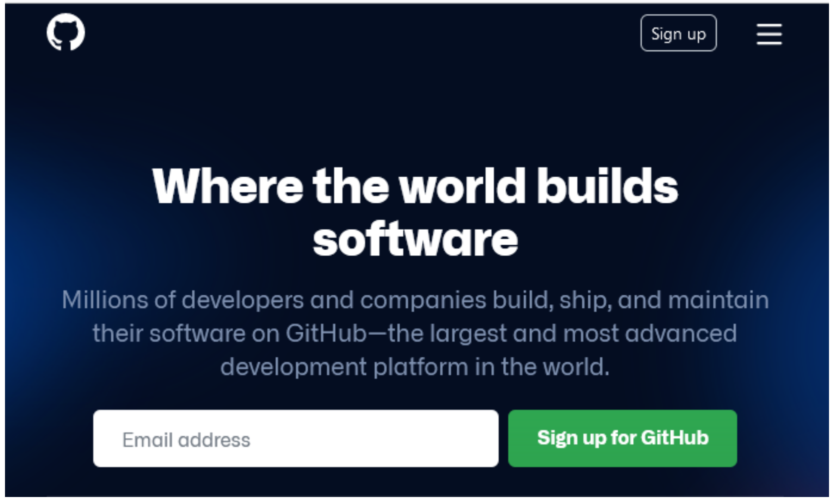


The DevOps Platform has arrived.

Deliver software faster with better security and collaboration in a single platform.

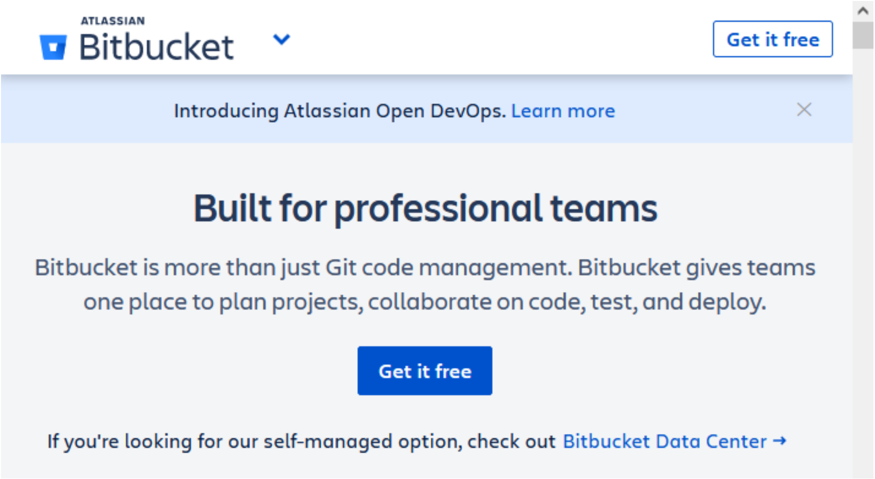
Other Repositories

- The are other major repository competitors to GitLab like Gitub and BitBucket
  - All offer similar repository features
- GitHub is the most popular
  - Supports CICD
  - Favorite for public repositories
  - Owned by MicroSoft
- BitBucket by Atlassian
  - Supports CIDC
  - Integrates with other Atlassian products like JIRA
- Choice often depends on other features



Where the world builds software

Millions of developers and companies build, ship, and maintain their software on GitHub—the largest and most advanced development platform in the world.



Introducing Atlassian Open DevOps. Learn more

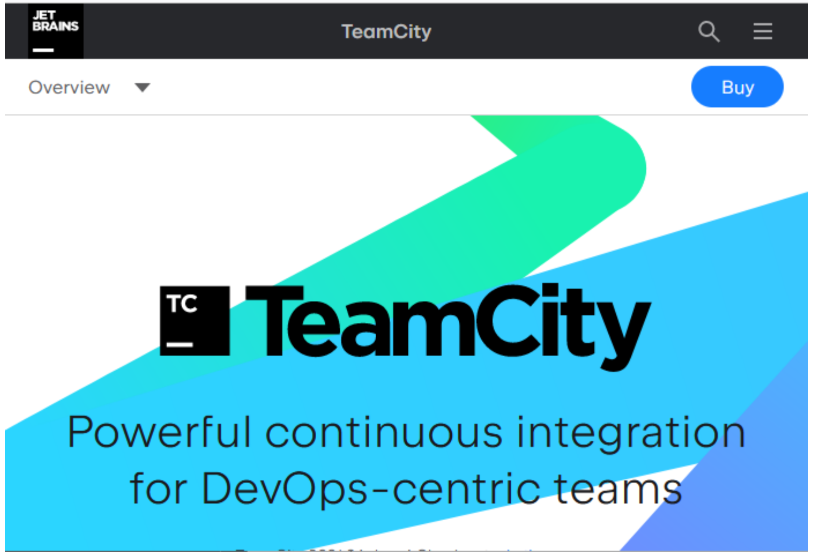
Built for professional teams

Bitbucket is more than just Git code management. Bitbucket gives teams one place to plan projects, collaborate on code, test, and deploy.

If you're looking for our self-managed option, check out Bitbucket Data Center →

Other Pipeline Tools

- Jenkins is the original
  - Currently tied with GitLab in terms of users’
  - Pure pipeline tool, no repository
  - Large community with many plugins
  - Open source and free
- TeamCity
  - Jetbrains product
  - Proprietry, free version available
  - Automation code is in Kotlin



TeamCity

Powerful continuous integration for DevOps-centric teams

Other Pipeline Tools

- Bamboo
  - Atlassian product
  - Integrates with other Atlassian products

- CircleCI
  - Similar to the others

- There are many other pipeline tools
  - Most integrate with the standard repositories
  - Offer similar features

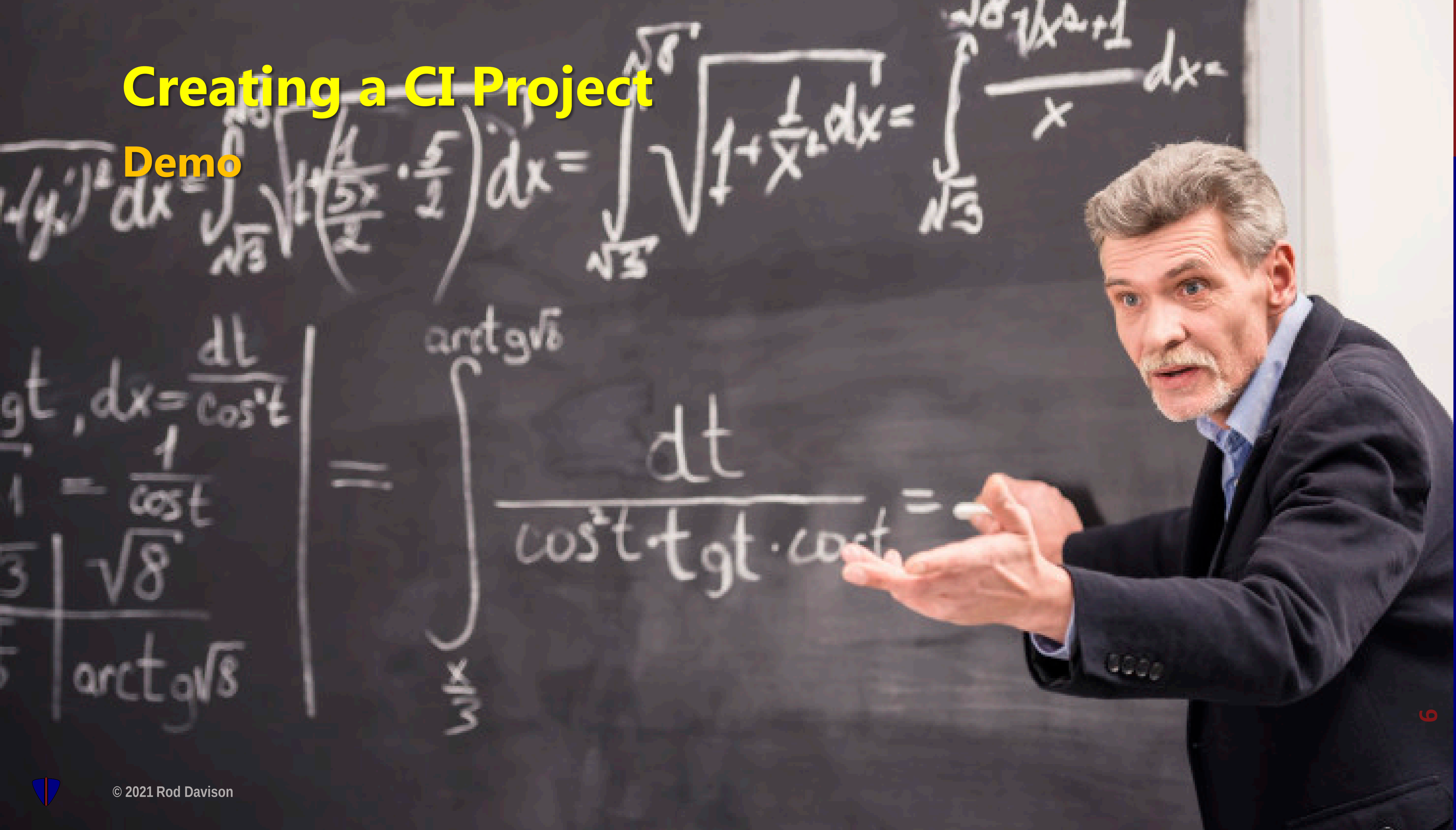
ATLASSIAN BAMBOO

CONTINUOUS INTEGRATION, DEPLOYMENT, AND RELEASE MANAGEMENT

Deployment Options: Data Center** | Server*

Creating a CI Project

Demo



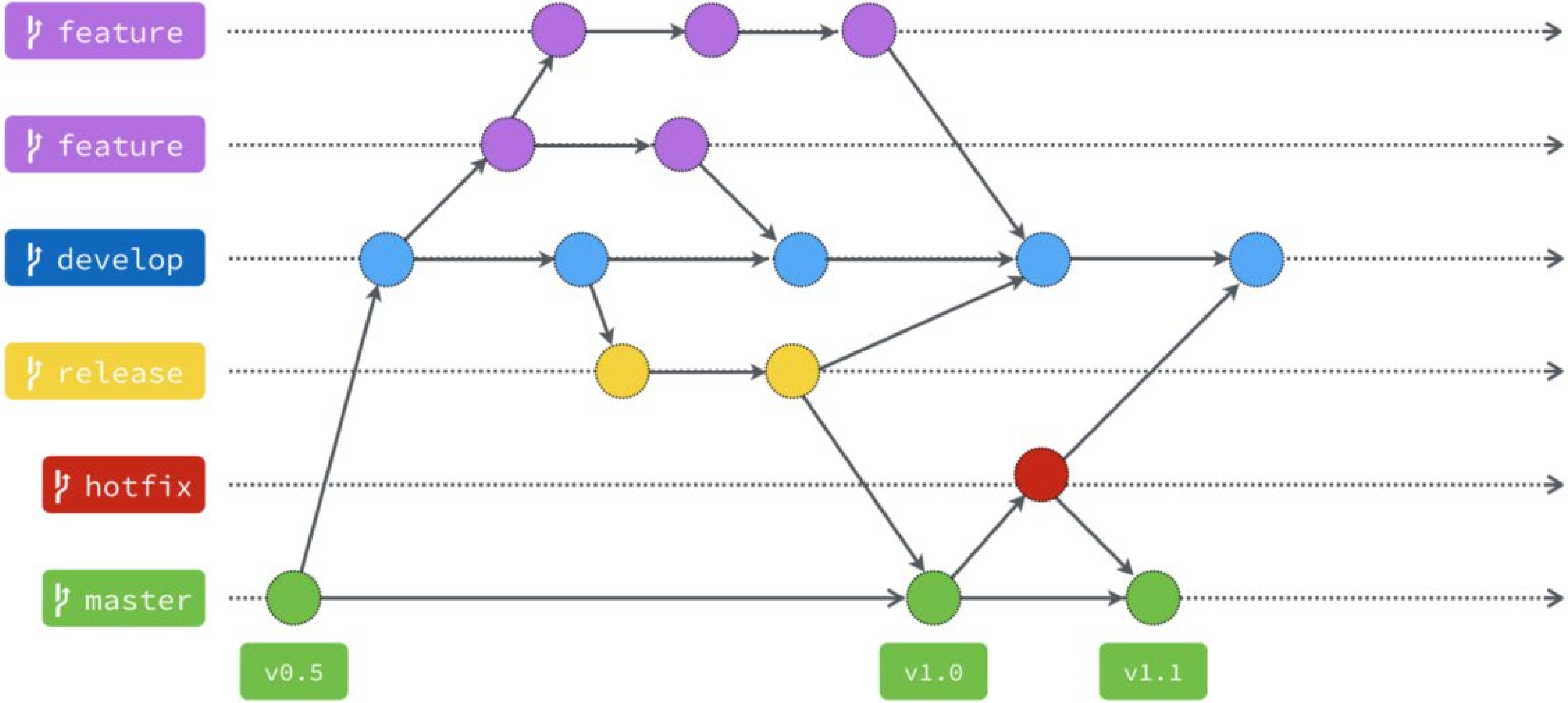
Create a CI Project

Lab Two

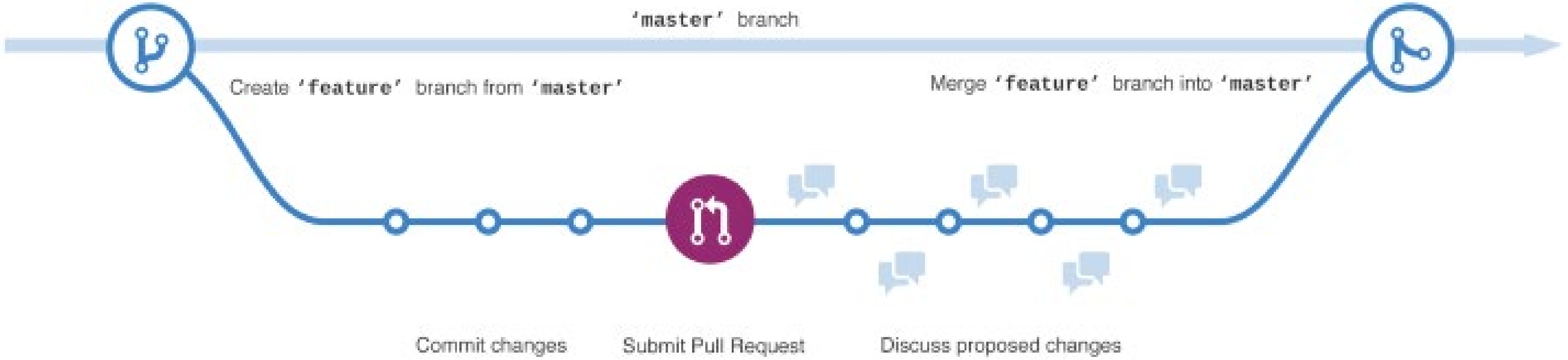
Branching Strategies

- A branching workflow is how developers:
  - Work in parallel on separate tasks and
  - Integrate their work into a codebase
- These are implementations of development models
  - All development on main branch; or
  - All development on feature branches
- All rely on branch and merge events
  - Generally, merges are the events that initiate a CICCD pipeline
- There are three main flows used
  - GitHub flow
  - Git flow
  - GitLab flow

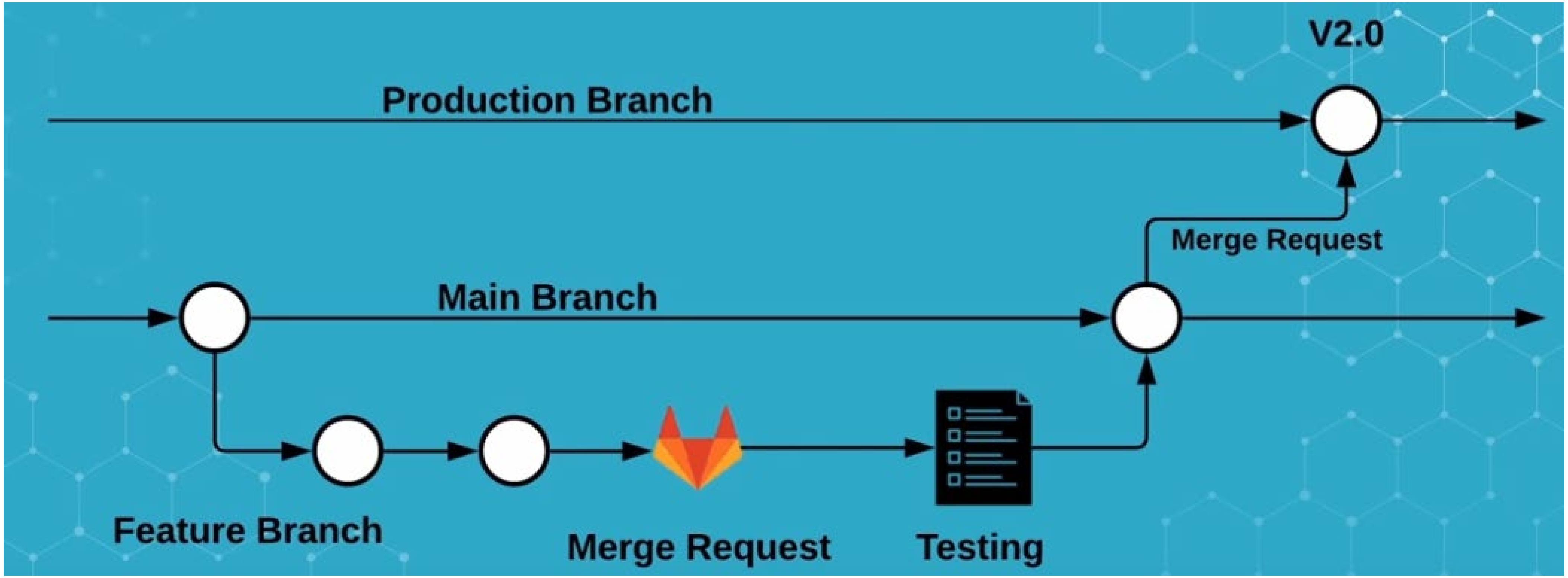
Git Flow



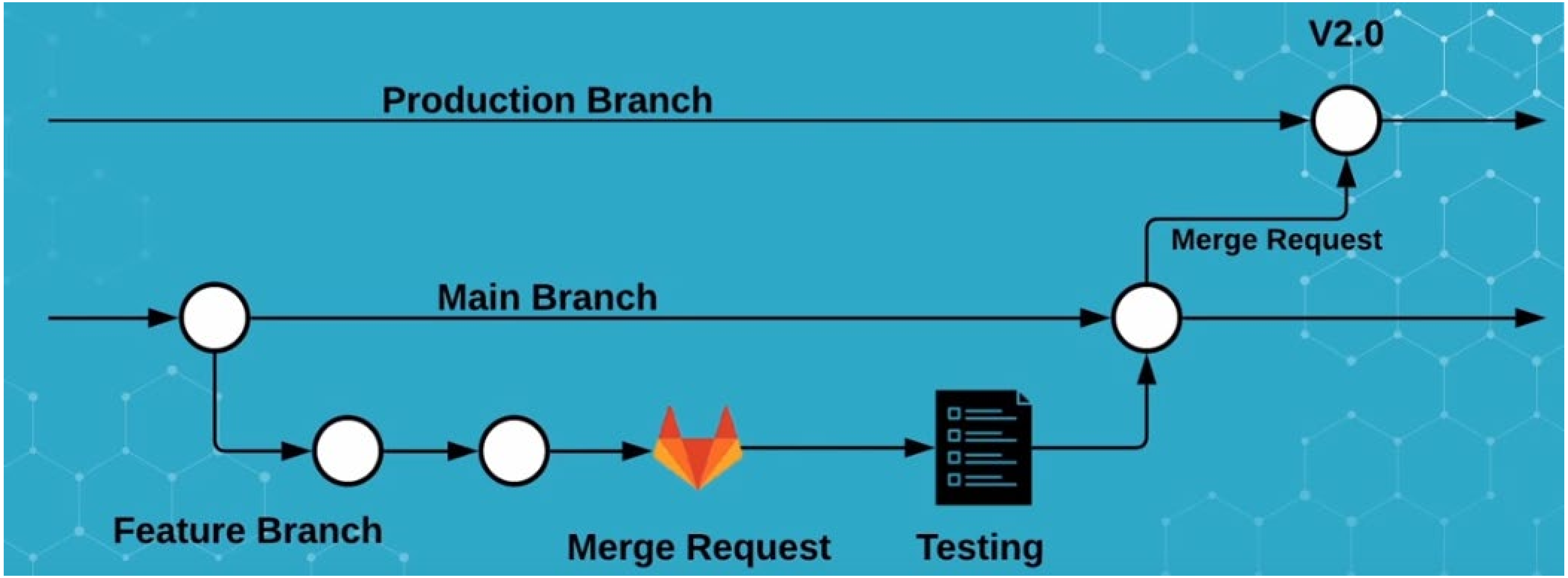
GitHub Flow


'master' branch
Create 'feature' branch from 'master'
Merge 'feature' branch into 'master'
Commit changes
Submit Pull Request
Discuss proposed changes

GitLab Flow Type 1



GitLab Flow Type 2



Feature Branch Workflow

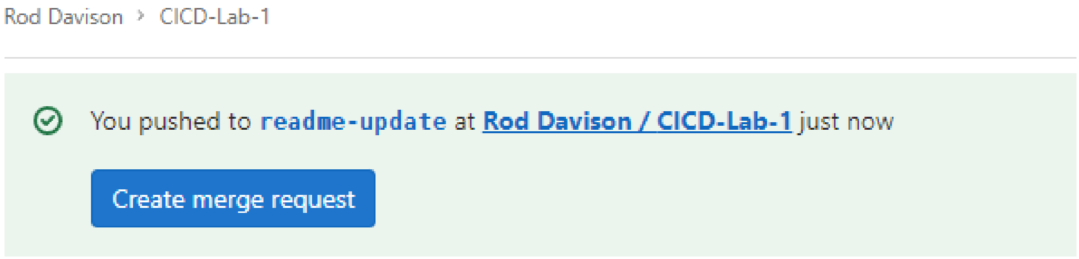
- The main branch is protected
  - Only authorized members can push or merge
- To do any work, create a feature branch
  - These branches should not be created in GitLab
- Clone the main branch to a local directory
  - Create the feature branch
  - Make changes, commit to the feature branch
  - Push the feature branch to GitLab
  - The feature branch will remain until it is merged into the main branch

Create a Feature Branch

Lab Three

Feature Branch Merge

- The feature branch has to be merged into main by creating a merge request
- The feature branch can be deleted after the merge is done
- Feature branches should neve be long lived



Rod Davison › CICD-Lab-1

You pushed to `readme-update` at Rod Davison / CICD-Lab-1 just now

Merge the Feature Branch

Lab Four

End of Module