Automating Deployments with CI & CD Using BitBucket Pipelines

Maintained by

Eric Jiang

This presentation's code/slides can be found on *https://github.com/lorderikir/cicd-techtalk*

Bitbucket Google Cloud Platform

Slide Deck Version: v1.0.0

Memes Warning: You might see many memes in this talk

So what is this talk about?

1. Introduction to BitBucket Pipelines
2. Branch Workflow and Setting up Permissions
3. Automating Testing with Pipelines
4. Automating Deployments to Google App Engine
5. Building advanced pipelines

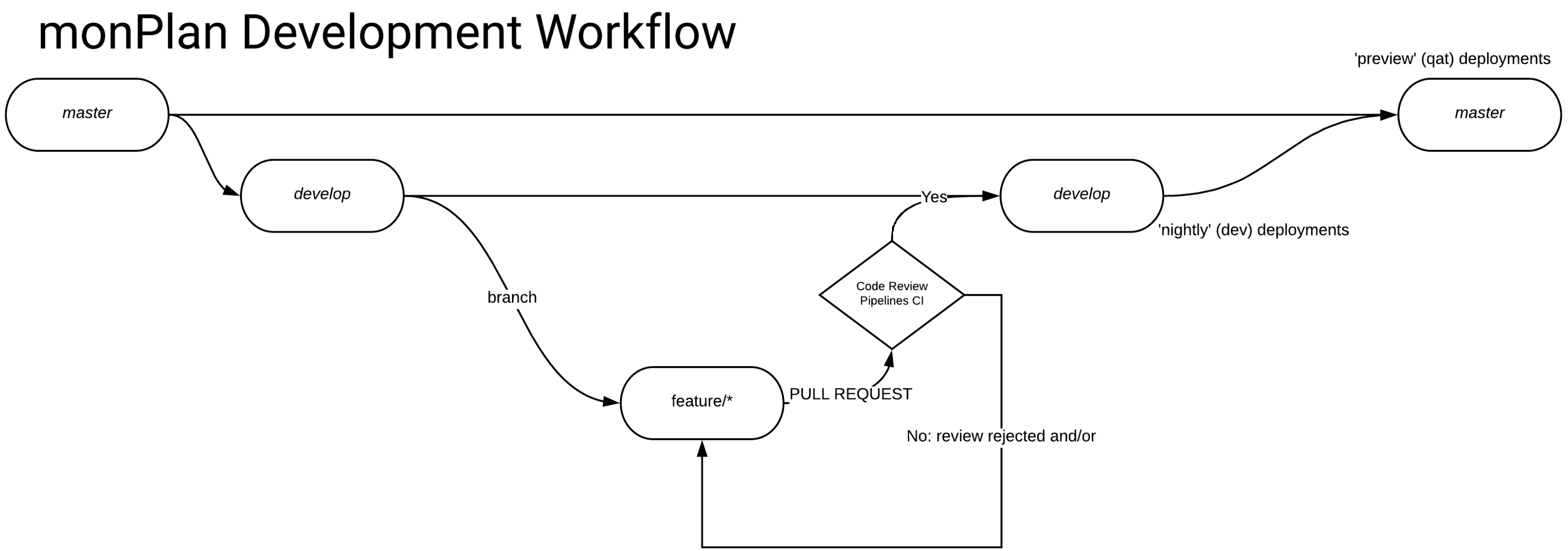
Introduction to BitBucket Pipelines

First of all, what is Pipelines?

> Integrated CI/CD for Bitbucket Cloud that’s trivial to set up, automating your code from test to production.
> *BitBucket Site (Atlassian)*

So how to do it?

monPlan Git Workflow


monPlan Development Workflow
'preview' (qat) deployments
master
master
develop
develop
Yes
'nightly' (dev) deployments
branch
Code Review
Pipelines CI
feature/*
PULL REQUEST
No: review rejected and/or

- **master**: branch is the key branch, everytime for release
- **develop**: *unstable*, most of the PRs should go here
- **'feature/*'**, **'fix/***, etc.: are 'for purpose' branches, these branches are for development
- **deploy** (not shown), is for **manual** deployments to prod

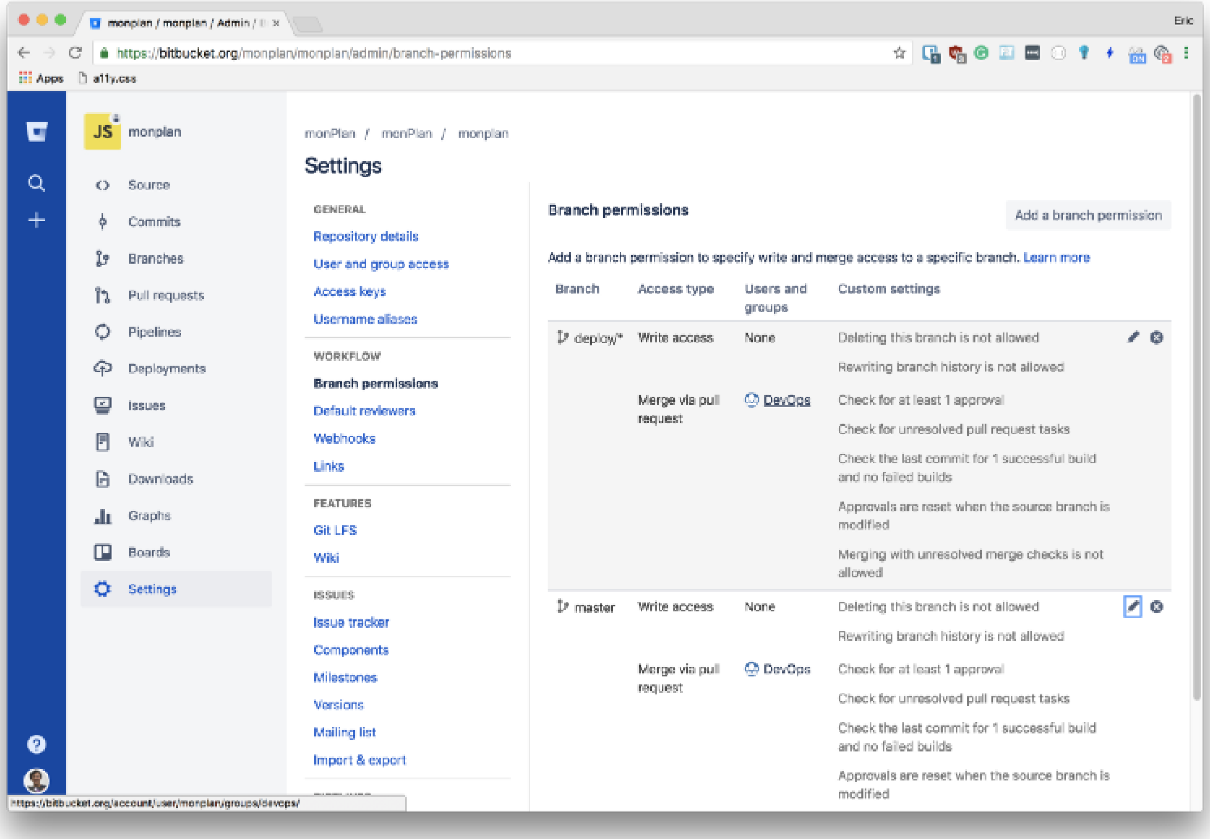
Branch Permissions

- **no-one** has write permissions to *master*, *develop* or *deploy*, not even DevOps or Admin.
- **everyone** has PR write access, only when
  - 1 Tests Pass
  - 1 Approval from a Code-review
- that way, we 'silly'-proof our codebase.

Setting Branch Permissions

Branch permissions by people who have *administrator* privellages inside a repo.

To set this, Go to Settings and under workflow select Branch Permissions.



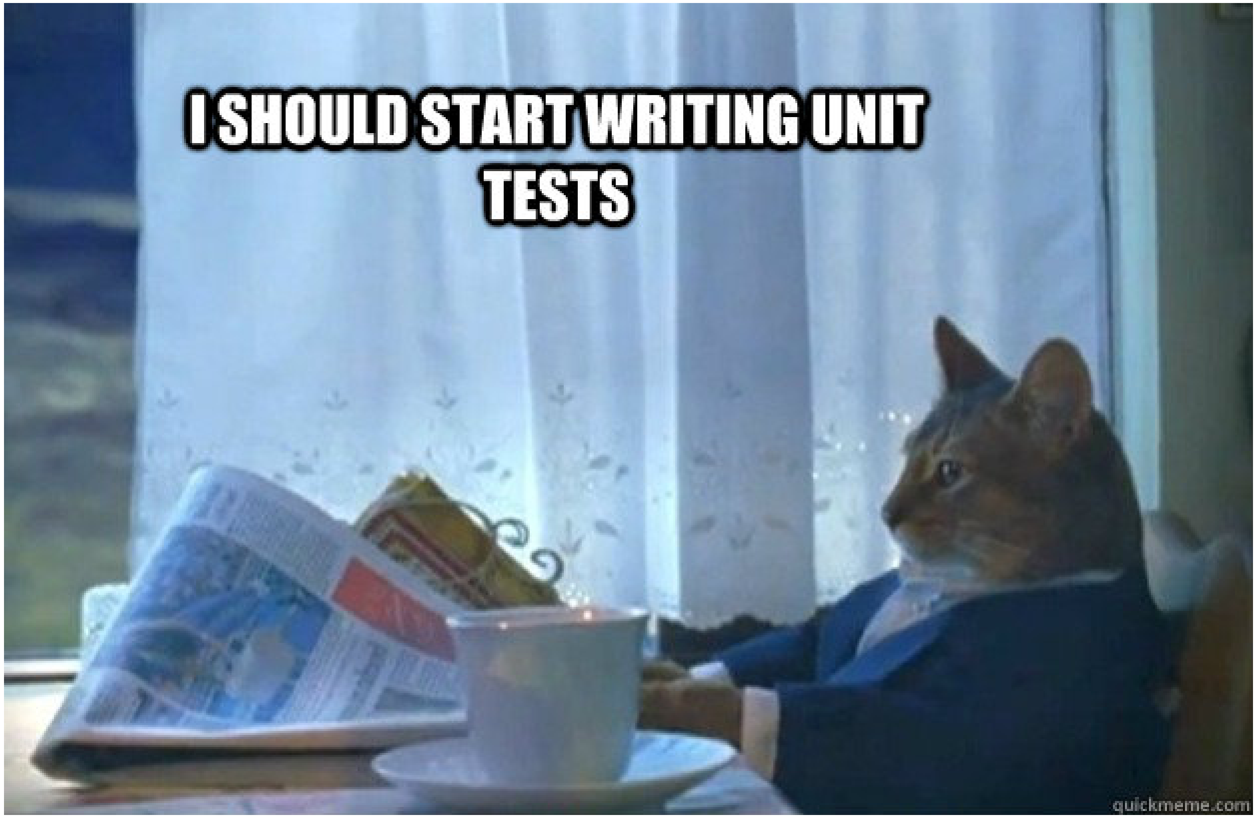
Branch permissions

Add a branch permission to specify write and merge access to a specific branch. Learn more

Branch	Access type	Users and groups	Custom settings
deploy*	Write access	None	Deleting this branch is not allowed Rewriting branch history is not allowed
	Merge via pull request	DevOps	Check for at least 1 approval Check for unresolved pull request tasks Check the last commit for 1 successful build and no failed builds Approvals are reset when the source branch is modified Merging with unresolved merge checks is not allowed
master	Write access	None	Deleting this branch is not allowed Rewriting branch history is not allowed
	Merge via pull request	DevOps	Check for at least 1 approval Check for unresolved pull request tasks Check the last commit for 1 successful build and no failed builds Approvals are reset when the source branch is modified

Demo of Branch Permissions

Automating Testing



Note that we will be using **NodeJS** for the sake of this demo/talk, but the idea is similar

First, It's the Development Team's Problem to write appropriate Tests, not the CI/CDs role

Second, the CI works on an alternative CLI, so running tests on interactive mode won't work

Setting up CI for Testing

default Pipeline runs for all branches (if not matched)

from bitbucket-pipelines.yml

```
image: node:8

pipelines:
  default:
    - step:
        caches:
          - node
        script:
          - yarn # <- Install Dependencies
          - yarn test:ci # <- Test in Non-CI Mode
```