Git Flow & Branching Strategy

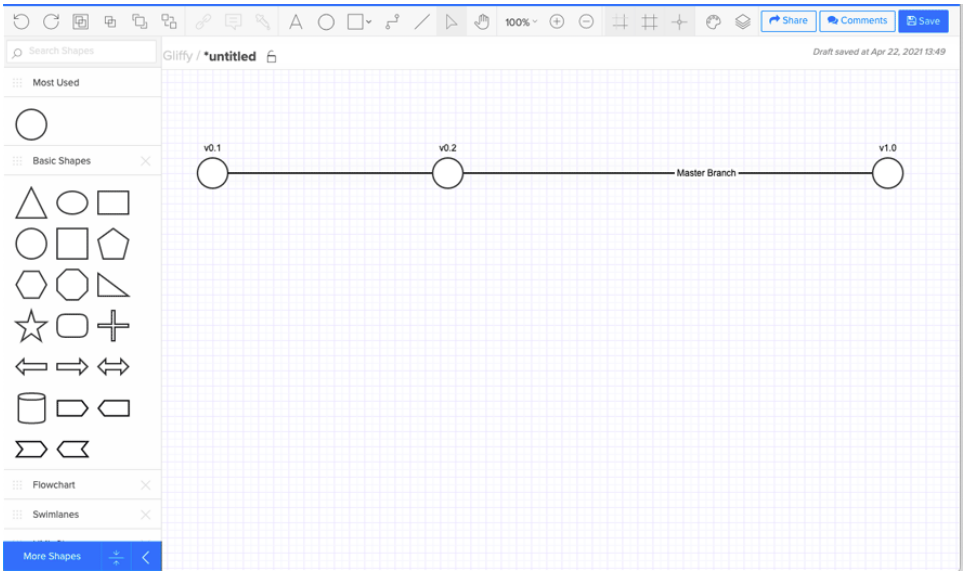
What is Gitflow?

Gitflow comes from the combination of “Git” and “workflow”. Gitflow is a popular Git branching model that organizes work into five or more branch types.

These types of branches are the main branch, develop branch, feature branch, release branch, and hotfix branch. Gitflow helps teams organize their work so that they can quickly and visually describe the workflows associated with their branching strategy and releases.

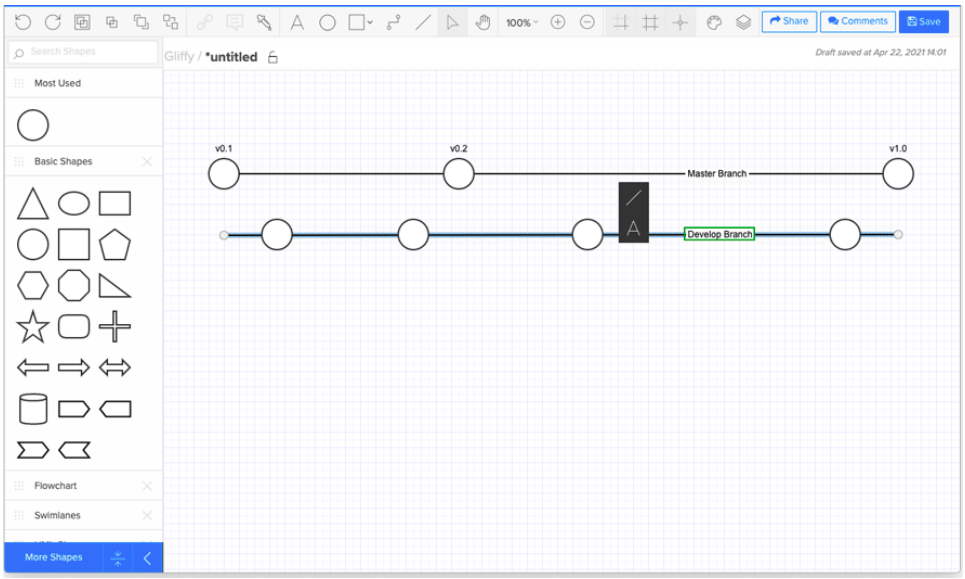
Step 1: Start with Your Master Branch

The master branch of your gitflow diagram describes production releases, so all changes and development work completed funnel up to this branch.



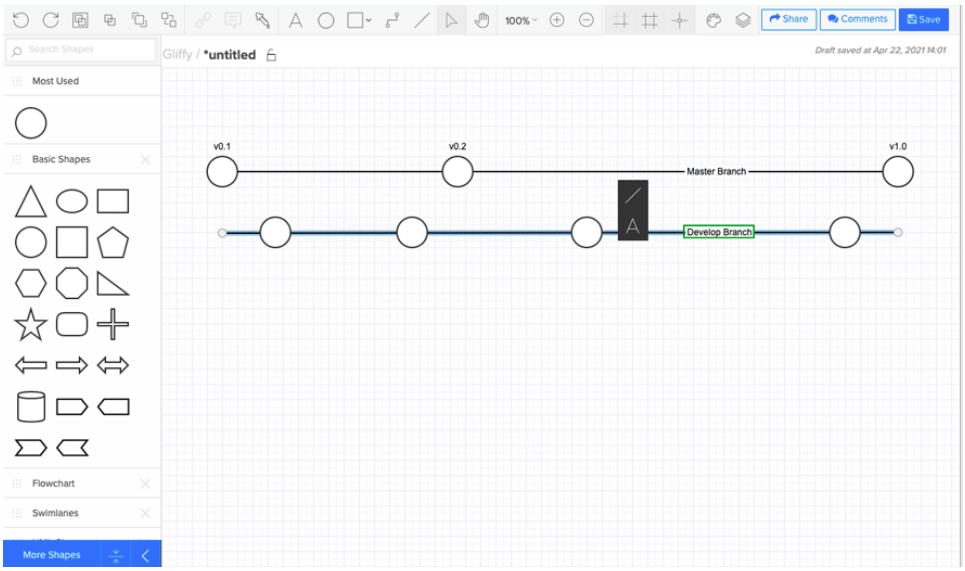
Step 2: Draw a Development Branch for Work in Progress

The second branch in your gitflow workflow diagram is the development branch. This branch is used throughout the development process for pre-production code — so a lot of work between released versions of your product will branch off of this line.



Step 3: Connect Your Branches to Show Where Work is Created and Merged

Using the connector tool, draw arrows between the circles on the master and development branches to show where work is being created and merged.

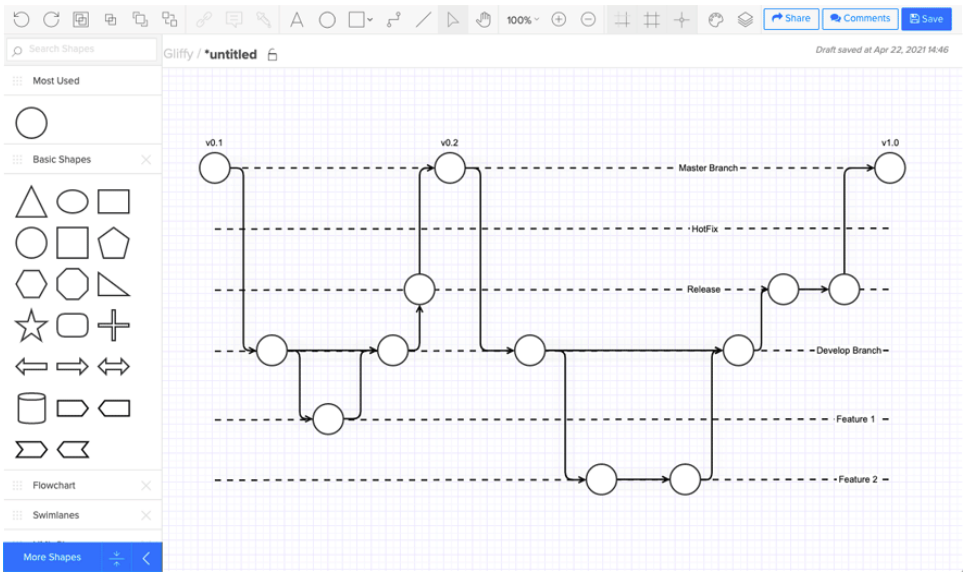


Step 4: Add Additional Branches for Features, Releases, and Hotfixes

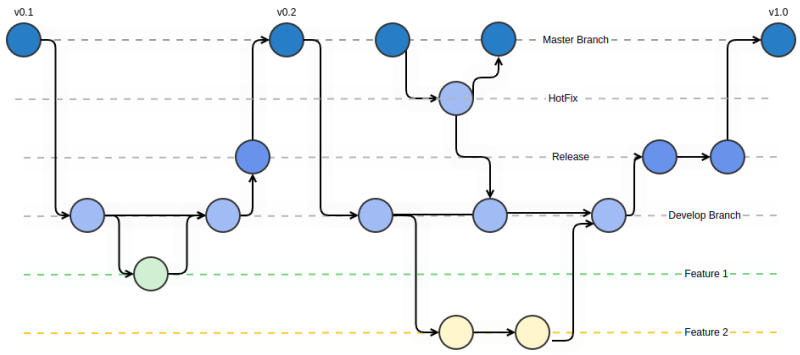
While most of the work will take place in the development branch and the master branch tracks what work is completed and released, there are three additional branches to support your work. These are the release, hotfix, and feature branches.

HOT FIX BRANCH

- A hotfix branch is created directly off the latest commit on master/main.
- The only commits allowed on the hotfix branch are ones that explicitly address the software bug.
- No feature enhancements or chores are allowed on the Gitflow hotfix branch.
- The hotfix branch merges into both master and develop branches when its lifecycle ends.
- The hotfix branch is deleted after it is merged or rebased into master and develop branches



Step 5: Final Git-Flow Diagram



VPC Region 1

Bastion Host

VPC-Peering

VPc Peering

KubeFed k8s API

Zone 1

Zone 2

Federated K8s control plane

Standard K8s API

Amazon EKS

Standard K8s API

Amazon EKS

Amazon EKS

Amazon elastic compute cloud (Node)

pod

Pod

VPC- Peering

Amazon EKS

Amazon elastic compute cloud (Node)

Pod

Pod

Database MultiAz

Lambda

Amazon S3

Amazon EKS

SNS

SQS

Code-Commit

Code-Build

Code-Pipeline

Cloud-Watch