Prayer for a smooth demo

Oh, benevolent Demo Gods, keepers of quality presentations and flawless functionality, I humbly beseech you. Banish all glitches. Let the audience be captivated. I offer this prayer in the hope of a smooth and glorious demo.

Amen

Our nextgen CI/CD pipeline

and the tools that make it

My goals for today

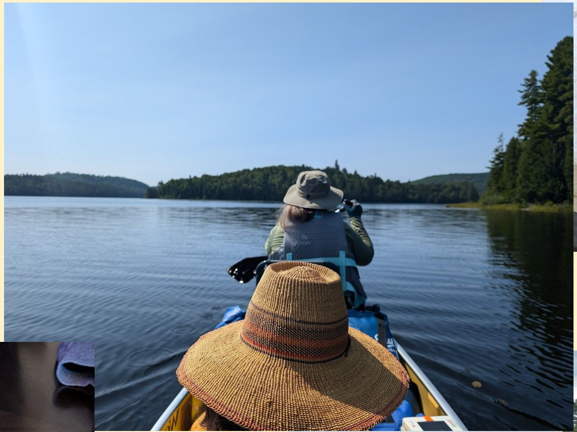
- Give a bit of history
- Show off our new CI/CD pipeline through demo
- Show you the collection of tools

Who am I?

Name: Matt McLane

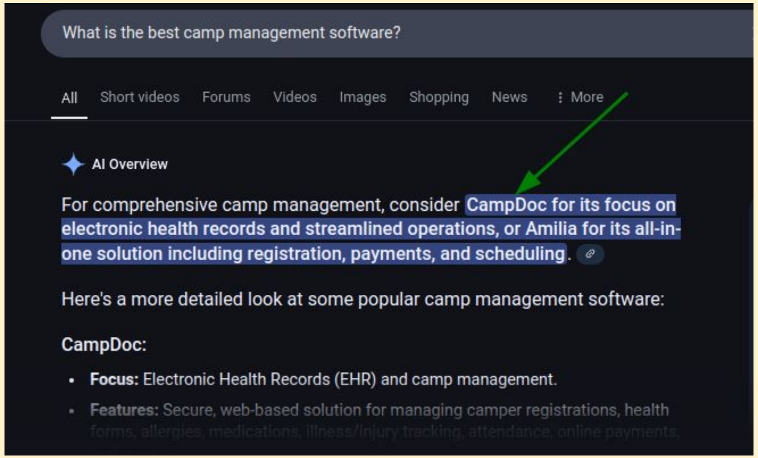
Company: DocNetwork

Profession: DevOps Engineer Supreme



DocNetwork

- Small software company in Ann Arbor, MI
- ~ 50 employees
- Camp Management software that provides Electronic Health Records for Camps, Child Care and Schools.



For comprehensive camp management, consider CampDoc for its focus on electronic health records and streamlined operations, or Amilia for its all-in-one solution including registration, payments, and scheduling.

Here's a more detailed look at some popular camp management software:

CampDoc:

- **Focus:** Electronic Health Records (EHR) and camp management.
- **Features:** Secure, web-based solution for managing camper registrations, health forms, allergies, medications, illness/injury tracking, attendance, online payments,

CampDoc
SchoolDoc

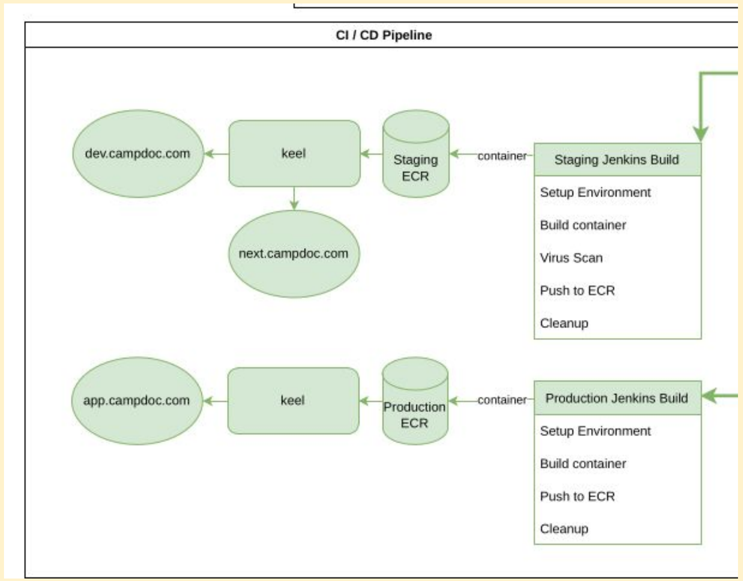
Tenacious Dev - DocNetwork's Dev Team

- 1 Director
- 1 DevOps Lead
- 3 Dev teams
    - 1 Dev Lead
    - 3 Devs
- 1 Product team
    - Product Manager
    - QA software tester

DocNetwork's Tech Stack

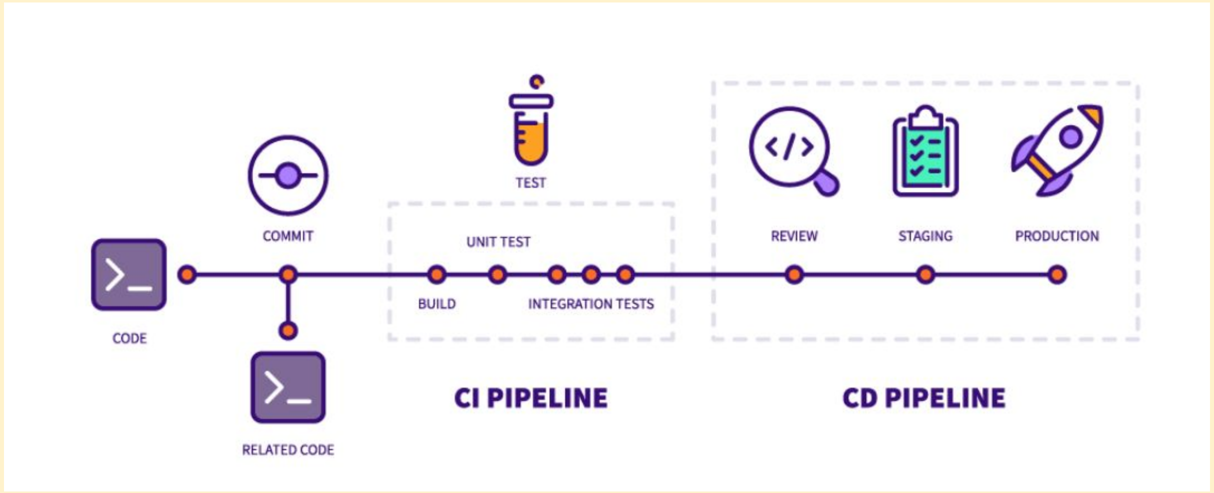
- NodeJS application with a Postgres Database
- Kubernetes Cluster running on AWS EKS
- GitOps workflows using ArgoCD

- Jenkins
  - Triggered by merges filtered by branch name
- ECR
- Keel



New CI/CD Pipelines

- Allow for easy rollout and rollback
- Embed and enforce real security
- Deploy to all environments
- Enable automated functional testing
- Manage configuration holistically
- Allow Devs to spend less time deploying.
    - 30 Man hours per week
- Replace older/problematic software like Jenkins, keel, ECR, SealedSecrets
- Embrace best practices
- Surface issues as early as possible
- Enable QA/Product to deploy
- Set ourselves up for more advanced scenarios



Continuous Integration (CI) - Automated process to create deployable artifacts

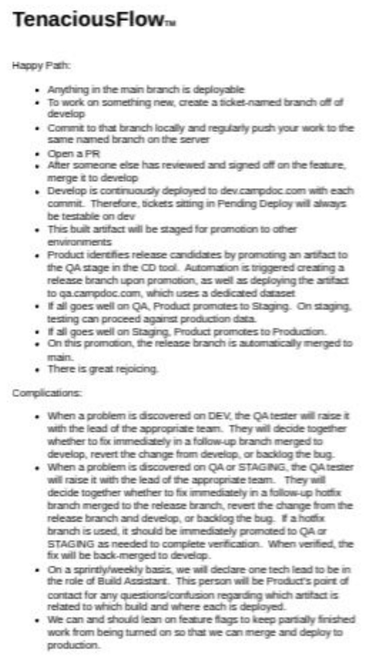
- Build and Test

Continuous Deployment (CD) - Automation to deploy versioned artifacts to your environments

- Deployment and promotion

CI vs. CD

Tenacious Flow



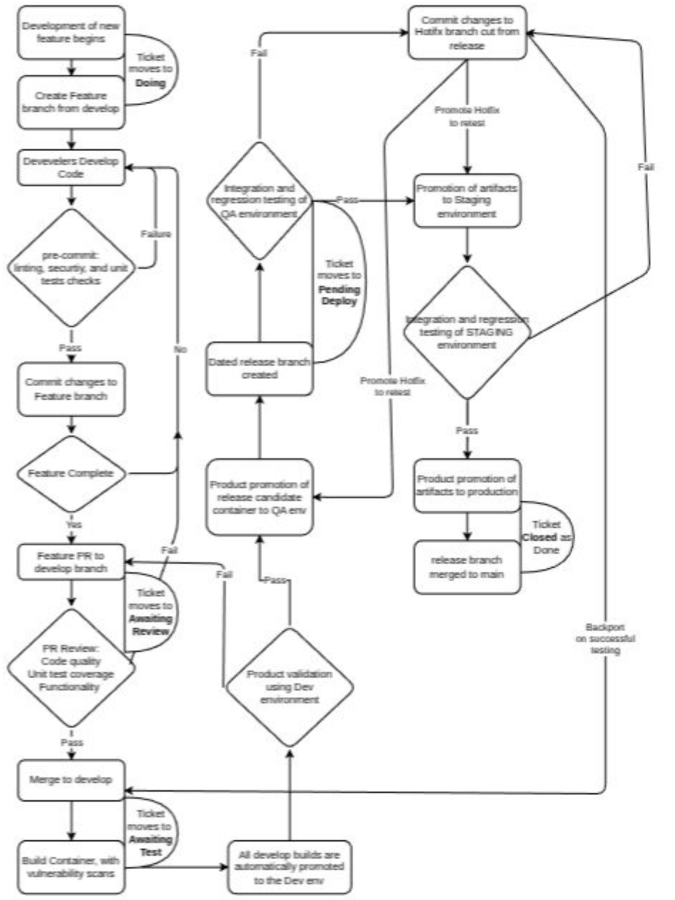
TenaciousFlow™

Happy Path:

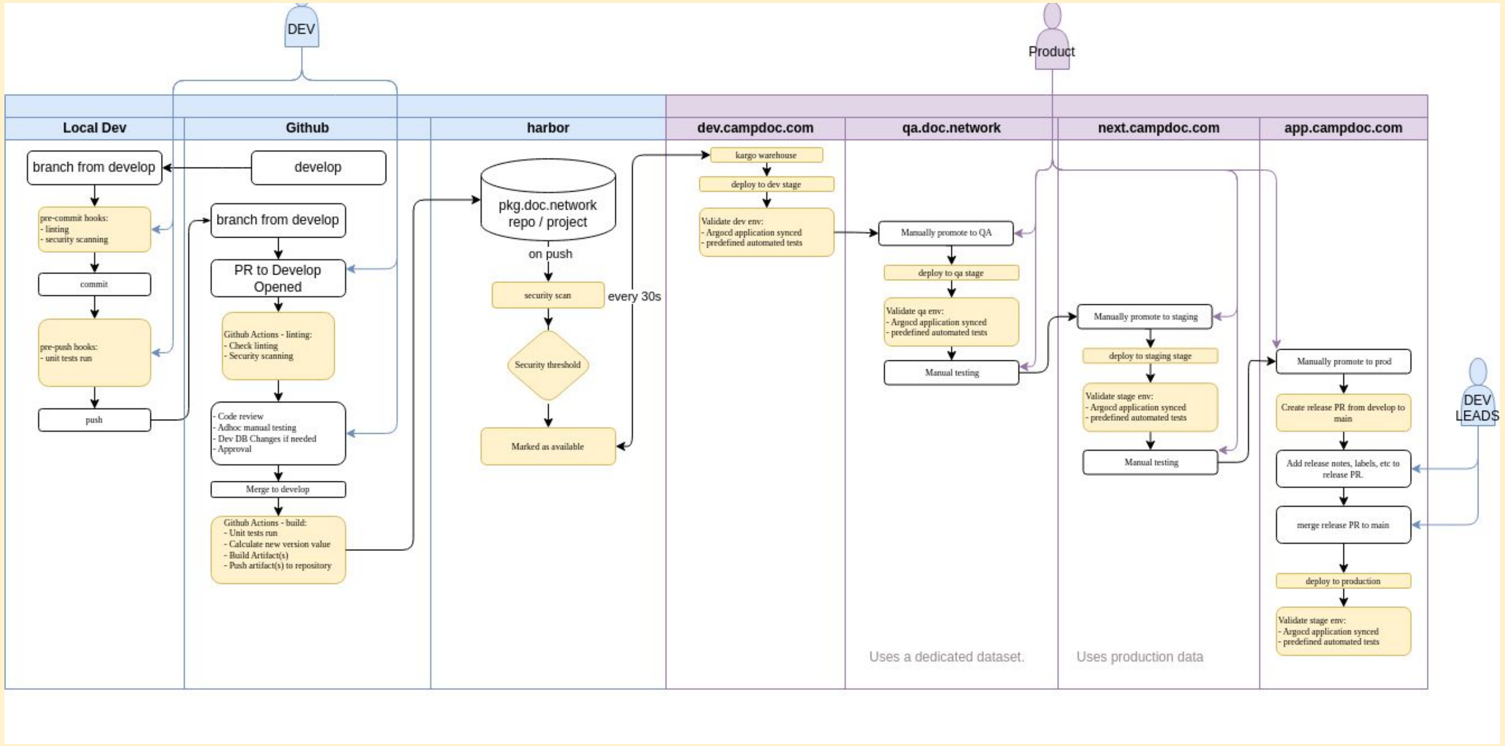
- Anything in the main branch is deployable
- To work on something new, create a ticket-named branch off of develop
- Commit to that branch locally and regularly push your work to the same named branch on the server
- Open a PR
- After someone else has reviewed and signed off on the feature, merge it to develop
- Develop is continuously deployed to dev.campdoc.com with each commit. Therefore, tickets sitting in Pending Deploy will always be testable on dev
- This built artifact will be staged for promotion to other environments
- Product identifies release candidates by promoting an artifact to the QA stage in the CD tool. Automation is triggered creating a release branch upon promotion, as well as deploying the artifact to qa.campdoc.com, which uses a dedicated dataset
- If all goes well on QA, Product promotes to Staging. On staging, testing can proceed against production data.
- If all goes well on Staging, Product promotes to Production.
- On this promotion, the release branch is automatically merged to main.
- There is great rejoicing.

Complications:

- When a problem is discovered on DEV, the QA tester will raise it with the lead of the appropriate team. They will decide together whether to fix immediately in a follow-up branch merged to develop, revert the change from develop, or backlog the bug.
- When a problem is discovered on QA or STAGING, the QA tester will raise it with the lead of the appropriate team. They will decide together whether to fix immediately in a follow-up hotfix branch merged to the release branch, revert the change from the release branch and develop, or backlog the bug. If a hotfix branch is used, it should be immediately promoted to QA or STAGING as needed to complete verification. When verified, the fix will be back-merged to develop.
- On a sprintly/weekly basis, we will declare one tech lead to be in the role of Build Assistant. This person will be Product's point of contact for any questions/confusion regarding which artifact is related to which build and where each is deployed.
- We can and should lean on feature flags to keep partially finished work from being turned on so that we can merge and deploy to production.



Development of new feature begins
Ticket moves to **Doing**
Create Feature branch from develop
Devevelers Develop Code
pre-commit: linting, securtiy, and unit tests checks
Failure
Pass
Commit changes to Feature branch
Feature Complete
No
Yes
Feature PR to develop branch
Ticket moves to **Awaiting Review**
Fail
PR Review: Code quality Unit test coverage Functionality
Pass
Merge to develop
Ticket moves to **Awaiting Test**
Build Container, with vulnerability scans
All develop builds are automatically promoted to the Dev env
Product validation using Dev environment
Fail
Pass
Product promotion of release candidate container to QA env
Dated release branch created
Ticket moves to **Pending Deploy**
Integration and regression testing of QA environment
Fail
Pass
Commit changes to Hotfix branch cut from release
Promote Hotfix to retest
Promotion of artifacts to Staging environment
Integration and regression testing of STAGING environment
Fail
Promote Hotfix to retest
Pass
Product promotion of artifacts to production
Ticket Closed as Done
release branch merged to main
Backport on successful testing



Lots of planning later...

Demo

CI

- Pre-commit
- Makefile
- Linting
- Trivy Scans
- Github Actions
- Harbor

CD

- ArgoCD and Helm Charts
- Kargo
- Build Helm Chart
- Promotion from Dev to Production
- Deployment PR

Future Enhancements

- Automation around handling hotfixes
- Renaming freight with deploy pr title
- Focus on automated testing
- Hand over deployments to product
- Adding lambda function deploy to these workflows

Questions?