MLops

Make life easy

Common problems with scaling ML

Getting started

Mundane steps with lots of boilerplate

Iterating quickly

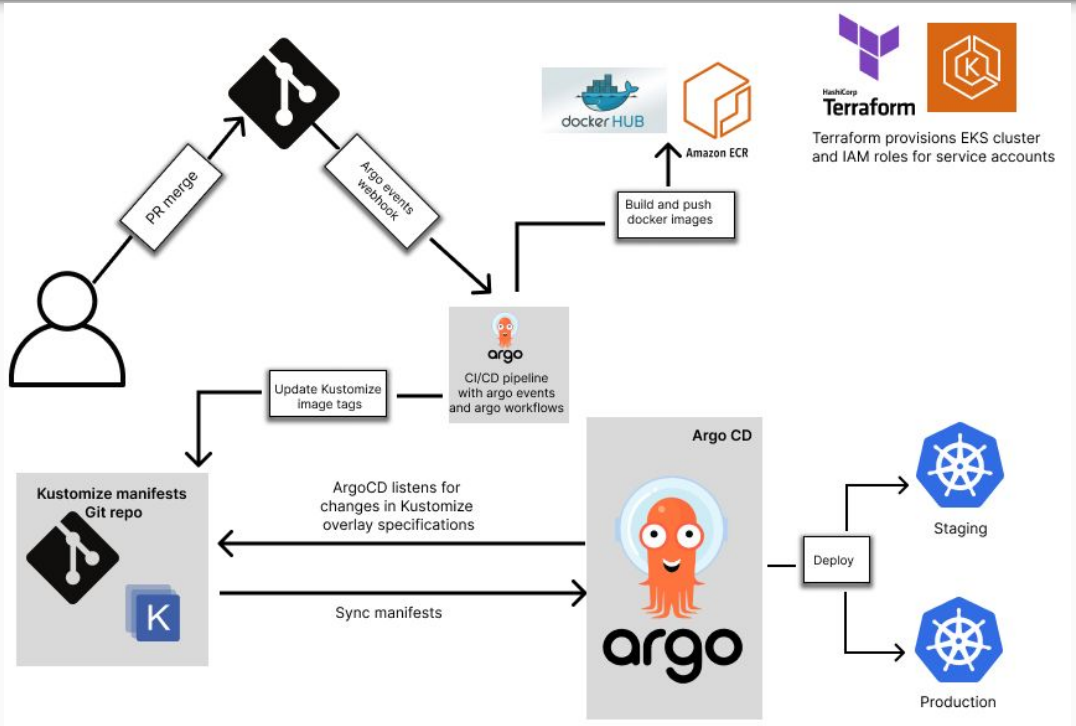
Slow redundant iterative development

Hard to experiment and compare different models/algorithms

Productionizing

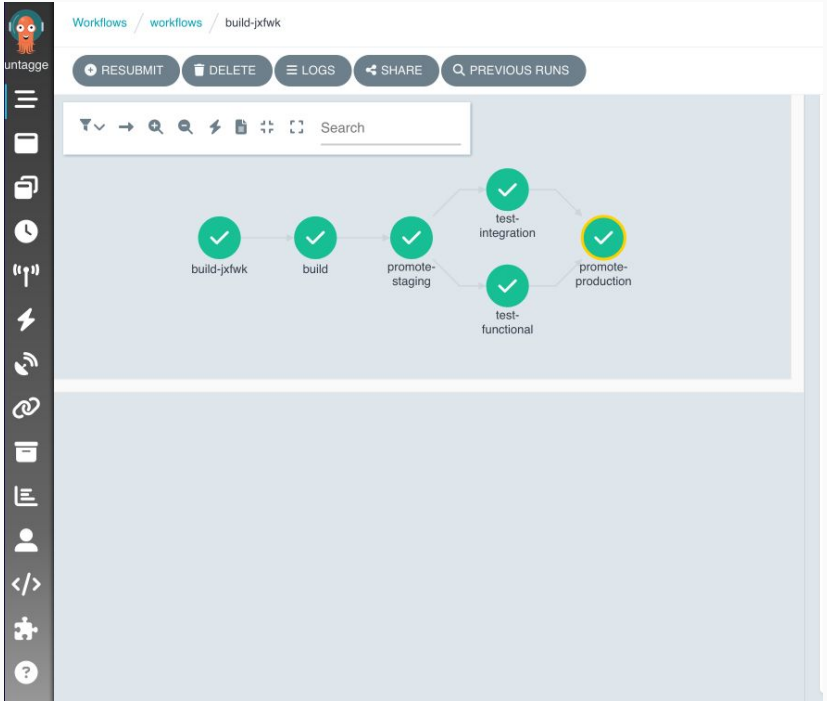
Incorporating ML models into production backend and handing off to engineers is very manual and bottlenecked

Gitops CI/CD for any software product



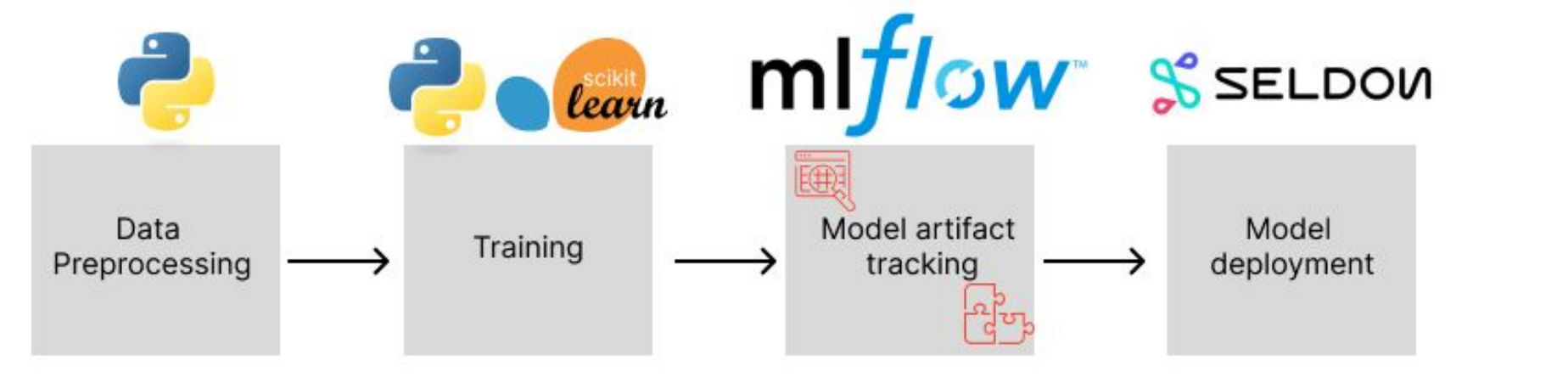
Argo workflows for CI/CD

1. Team makes change to MERN stack application and makes a Github PR
2. Github sends a webhook POST request to Argo event source
3. The CICD workflow trigger source is started
4. Docker image is built for frontend and backend and pushed to docker registry
5. New image tags are updated in Kustomization.yaml file
6. ArgoCD picks this up and updates the deployment
7. Integration tests can be carried out in the staging environment to test that APIs endpoints can be reached and they return the expected results.
8. If all goes well, then we can update manifests in the production cluster

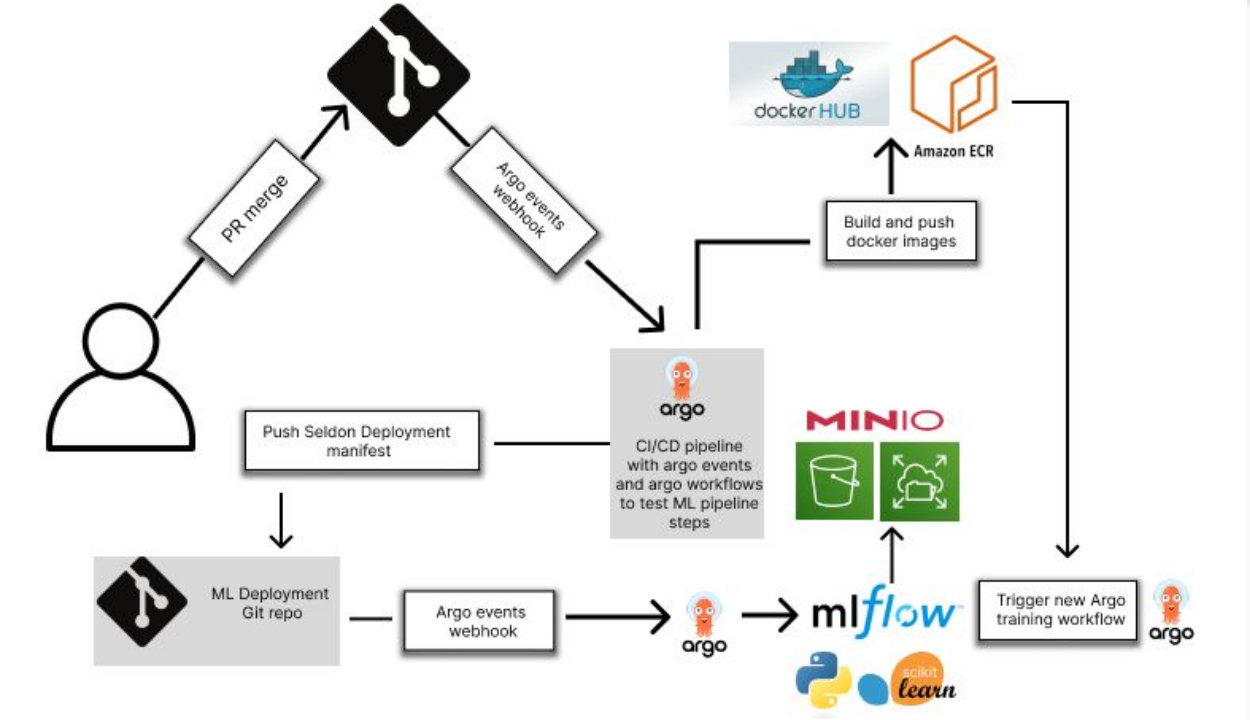


CI/CD for ML pipelines

Machine learning as composable units

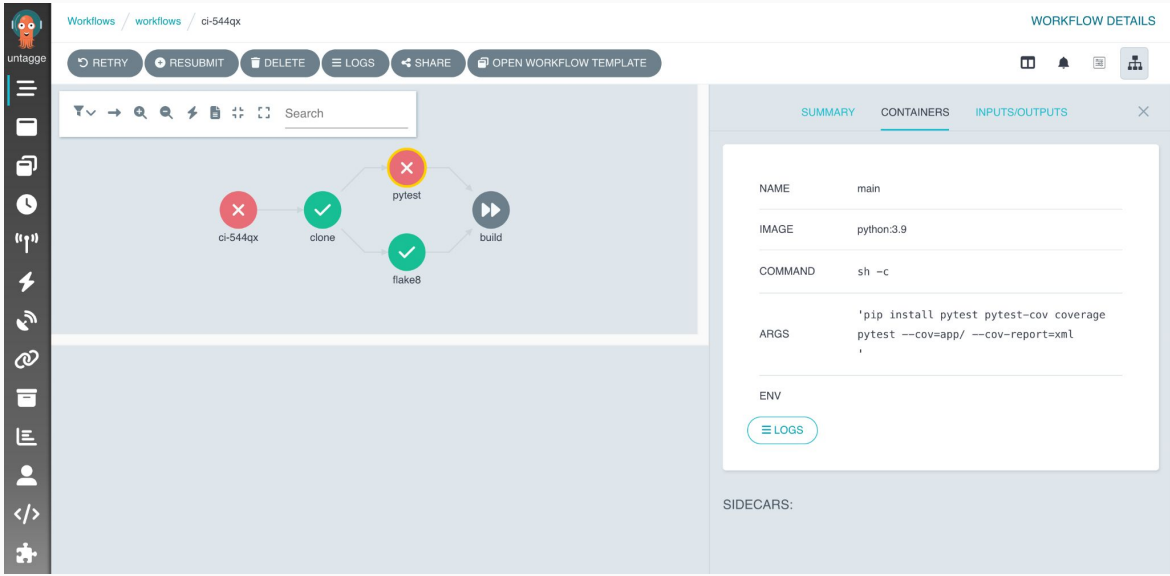


CI/CD for ML training pipeline



CI for Machine Learning

1. Once a model has been experimented and it has found to outperform the model in production, it is then modularized into ML pipeline steps, which are later containerized. Each pipeline step should be unit tested.
2. The next steps would be:
    a. Build docker image with build-kit
    b. Update image tags in staging Kustomize files for argo training workflow
3. This ensures that the argo training workflow will use the most up-to-date image for the training pipeline steps



CICD of the inference pipeline