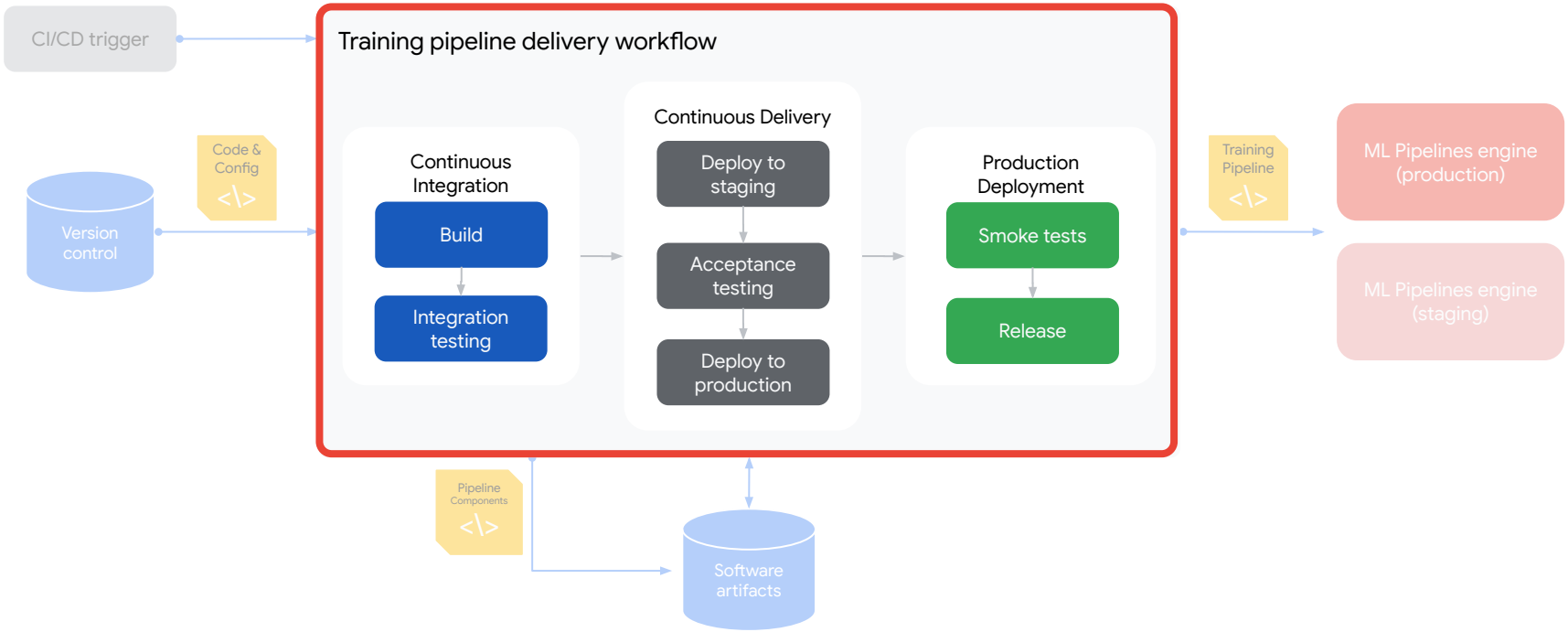
Training Operationalization for TinyML

MLOps: Training Operationalization



The MLOps **Personas**

ML Engineer

ML Researcher

Data Scientist

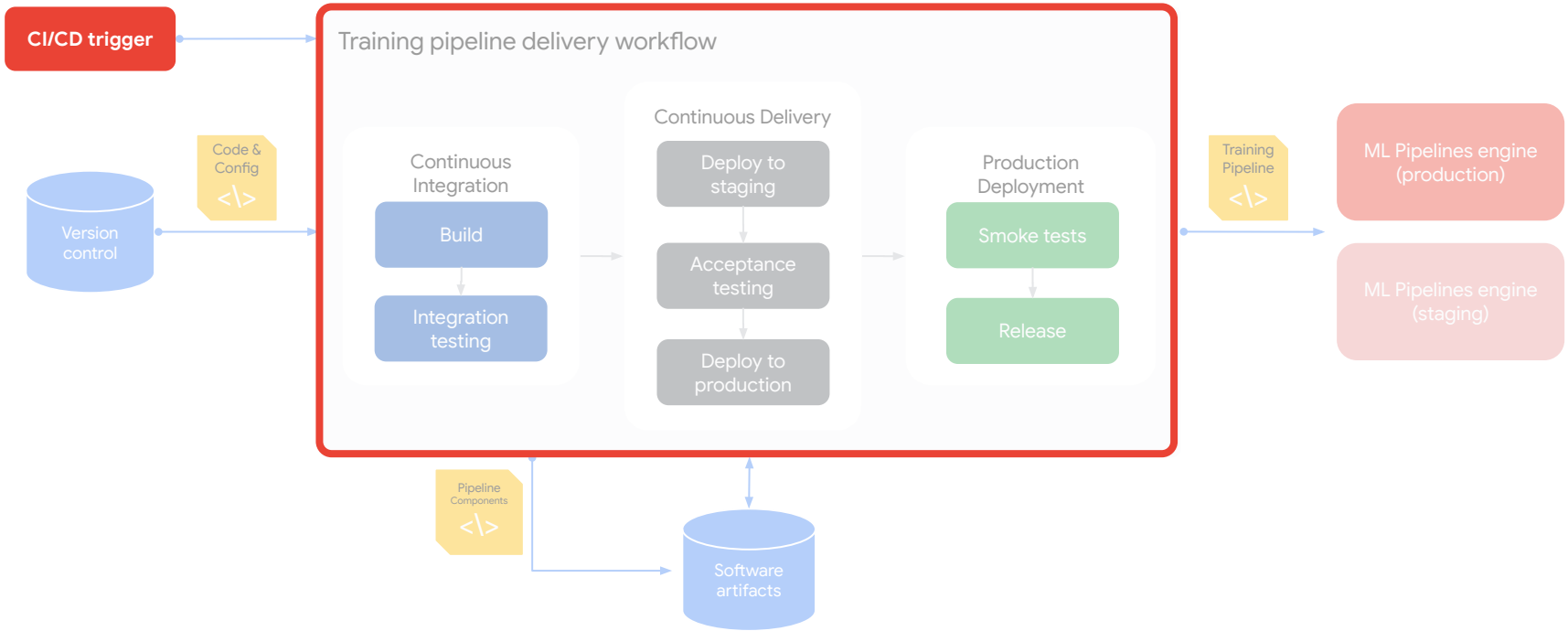
Data Engineer

Software Engineer

DevOps

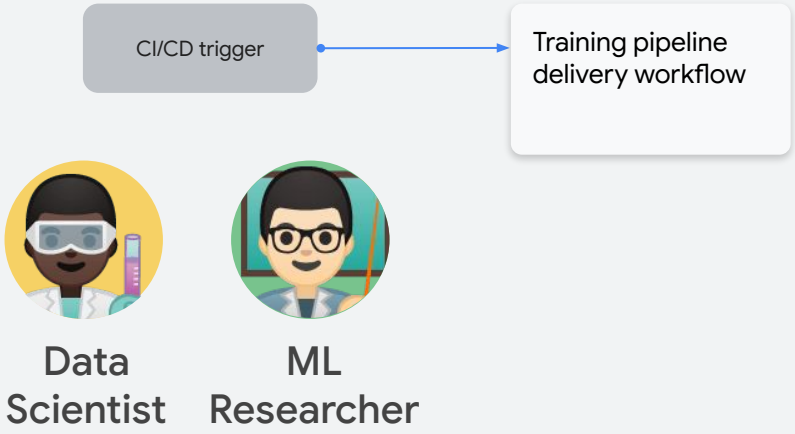
Business Analyst

MLOps: Training Operationalization



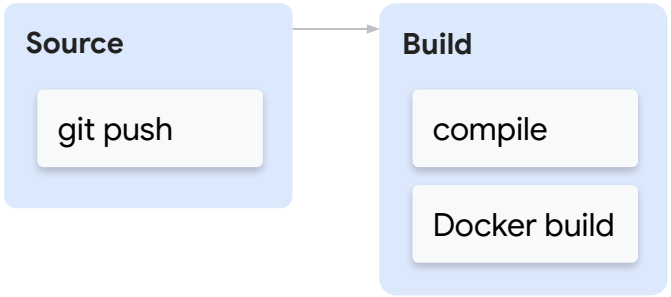
What Triggers CI/CD

1. “Experimentation & prototyping” phase is **complete**
2. ML researcher or data scientist **pushes** the software artifacts to the central repository
3. Push **automatically triggers** CI/CD

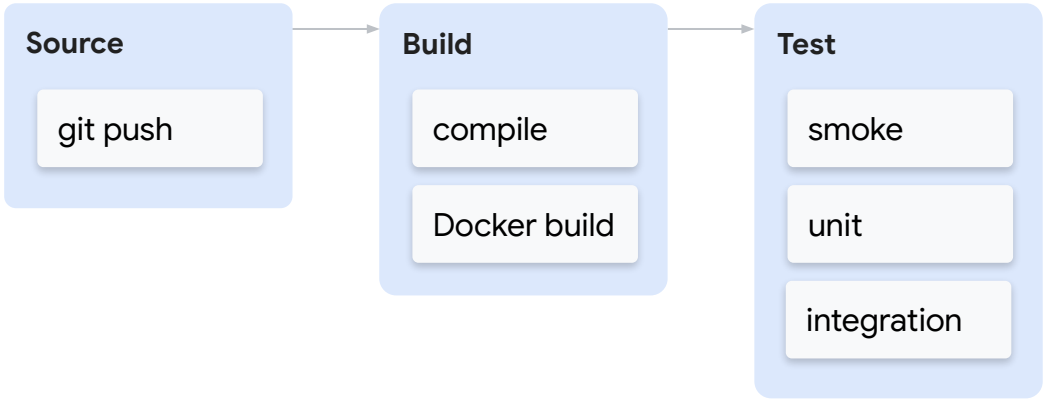


Example CI/CD Flow

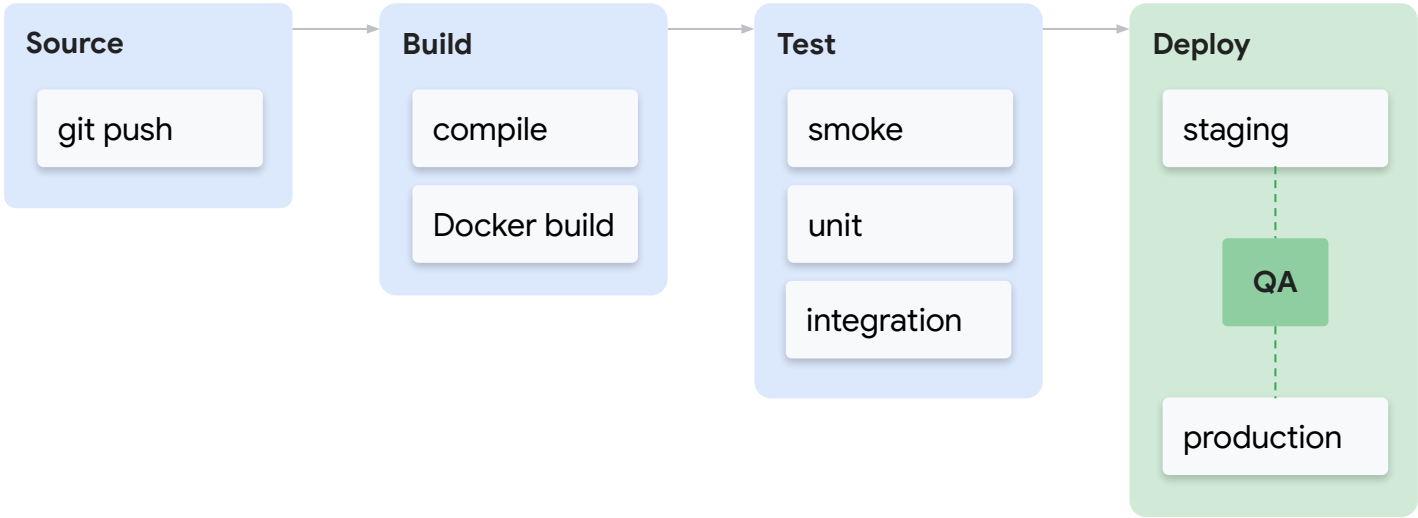
Example CI/CD Flow



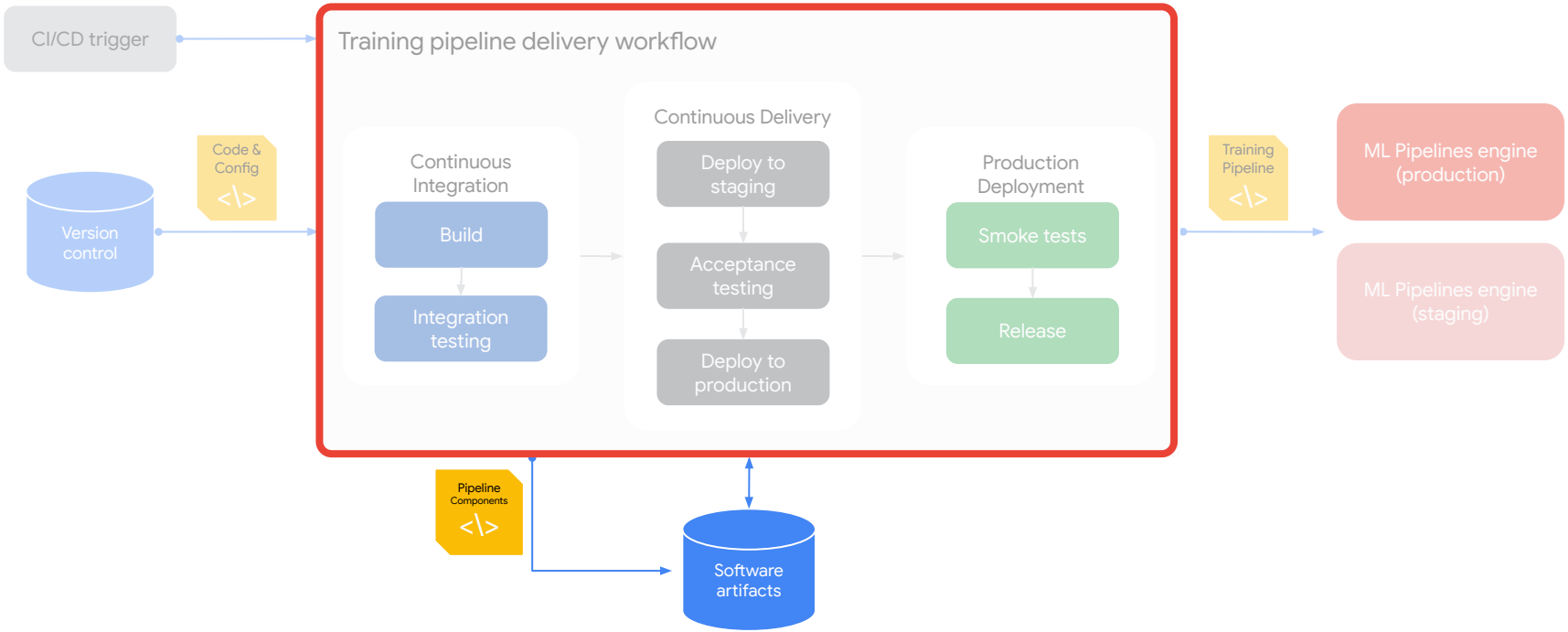
Example CI/CD Flow



Example CI/CD Flow

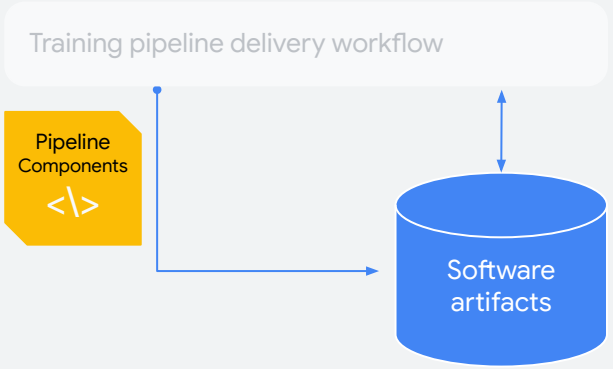


MLOps: Training Operationalization

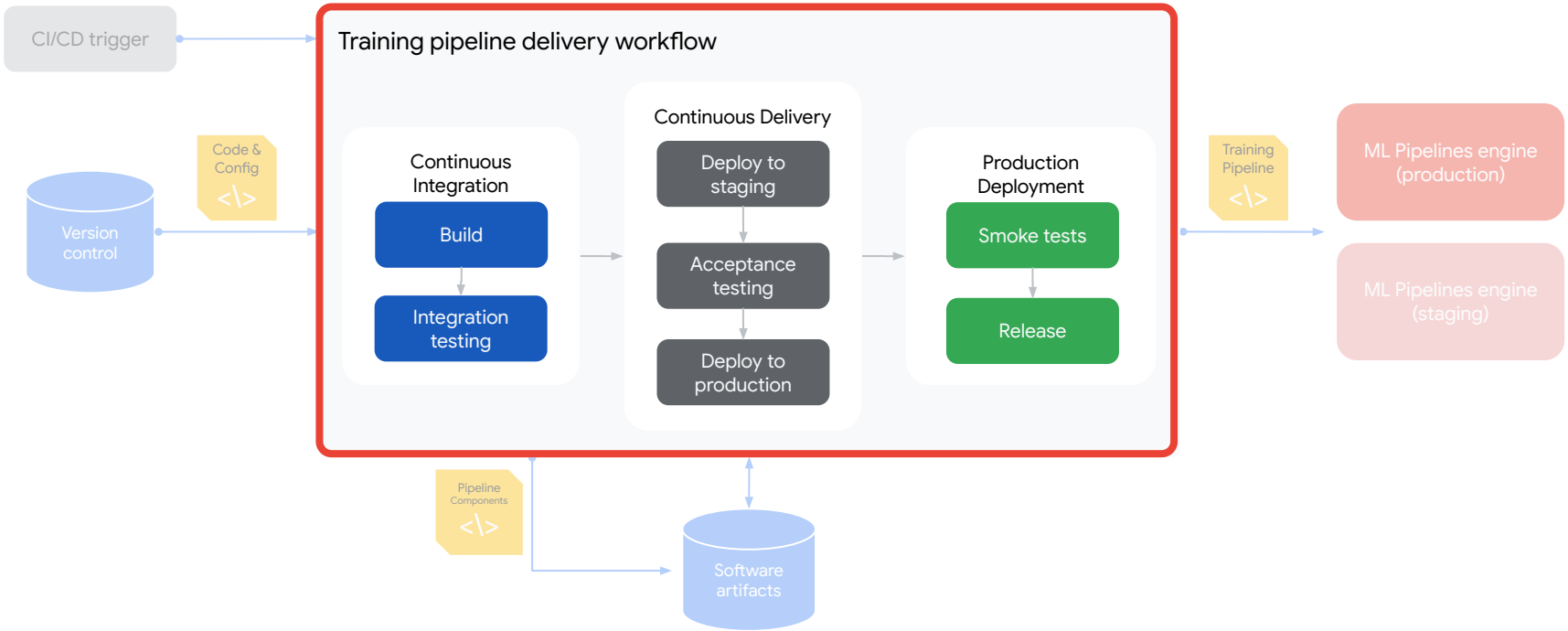


ML Software Artifact

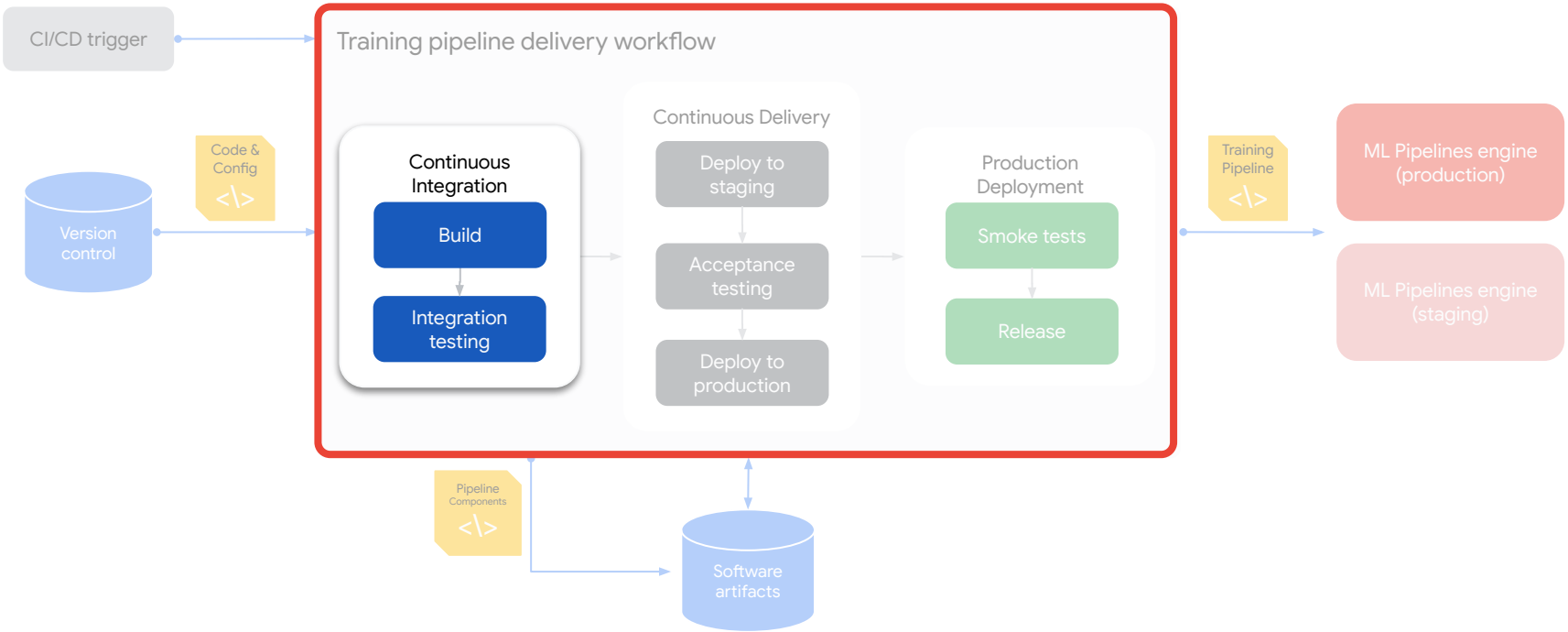
- Code for the model
- Preprocessing code
- Training and validation data
- Trained model in runnable format
- Build and run environment
- Documentation
- Code and data for testing



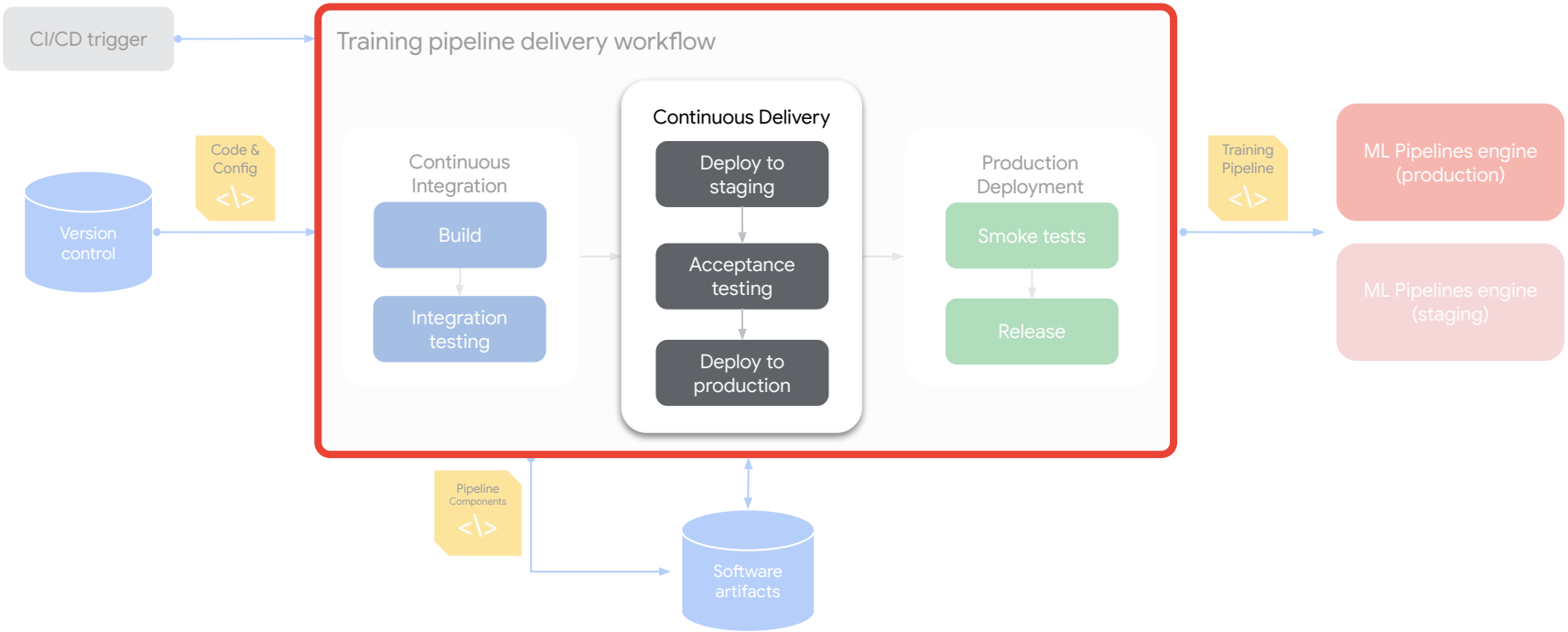
MLOps: Training Operationalization



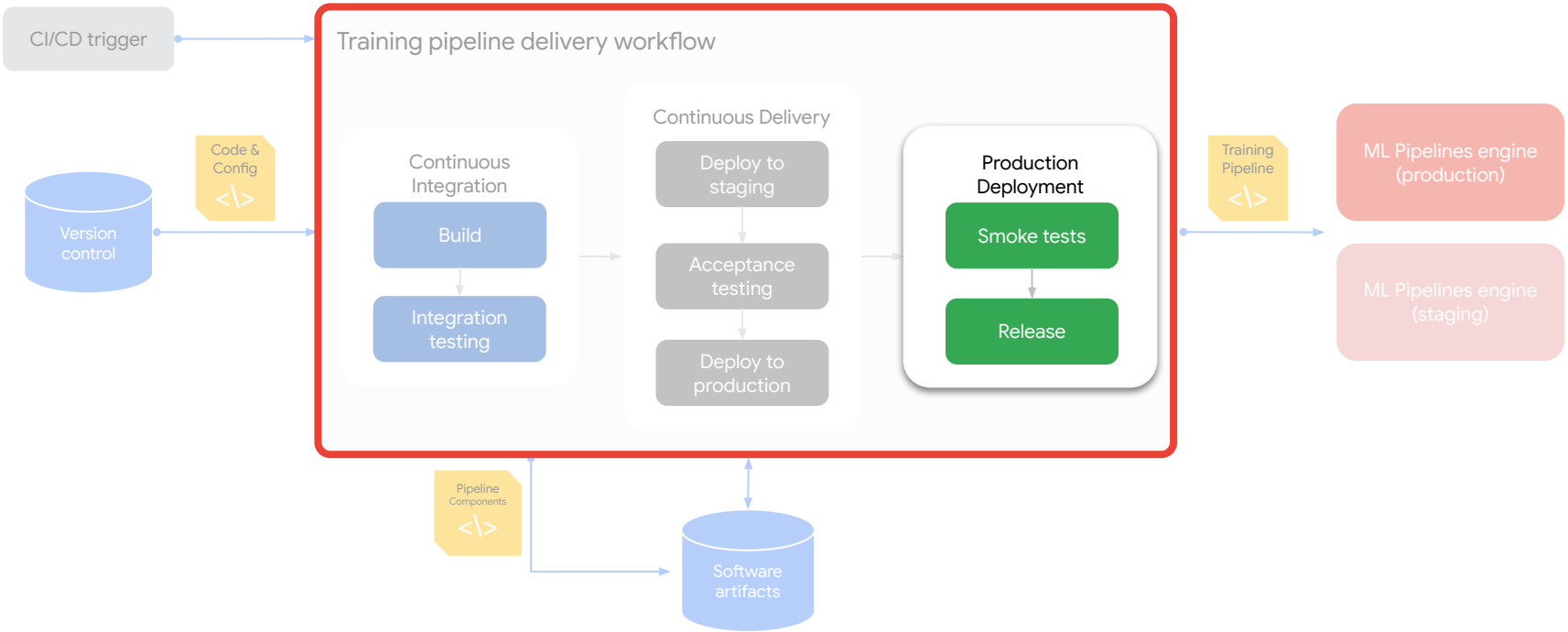
MLOps: Training Operationalization



MLOps: Training Operationalization

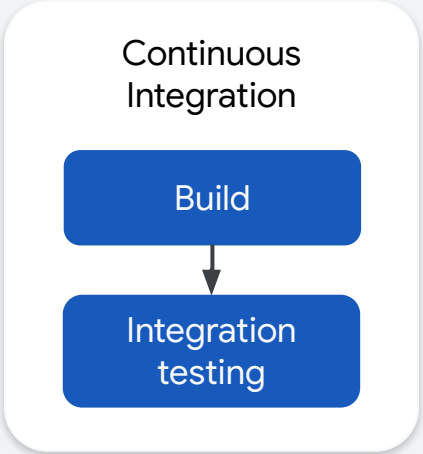


MLOps: Training Operationalization



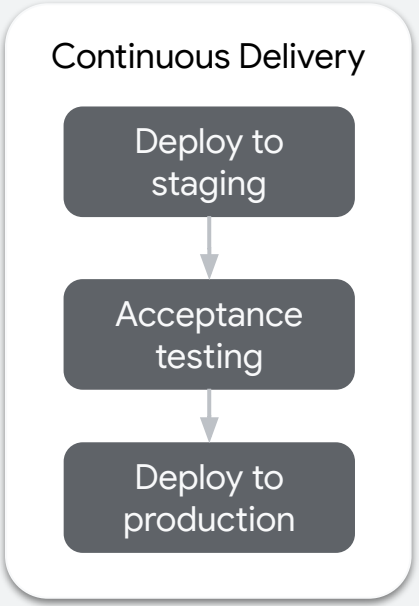
TinyML CI Questions

- What does the **build environment** look like?
- What **types of assets** do I need to consider writing for testing?



TinyML CD Questions

- What does the **staging environment** look like?
- What is considered as **accepted testing**?
- What and **how do I deploy** into production?



TinyML Production Deployment Questions

- What does it mean to do a **smoke test** for embedded machine learning systems?
- How can you do a **production release** with TinyML devices?