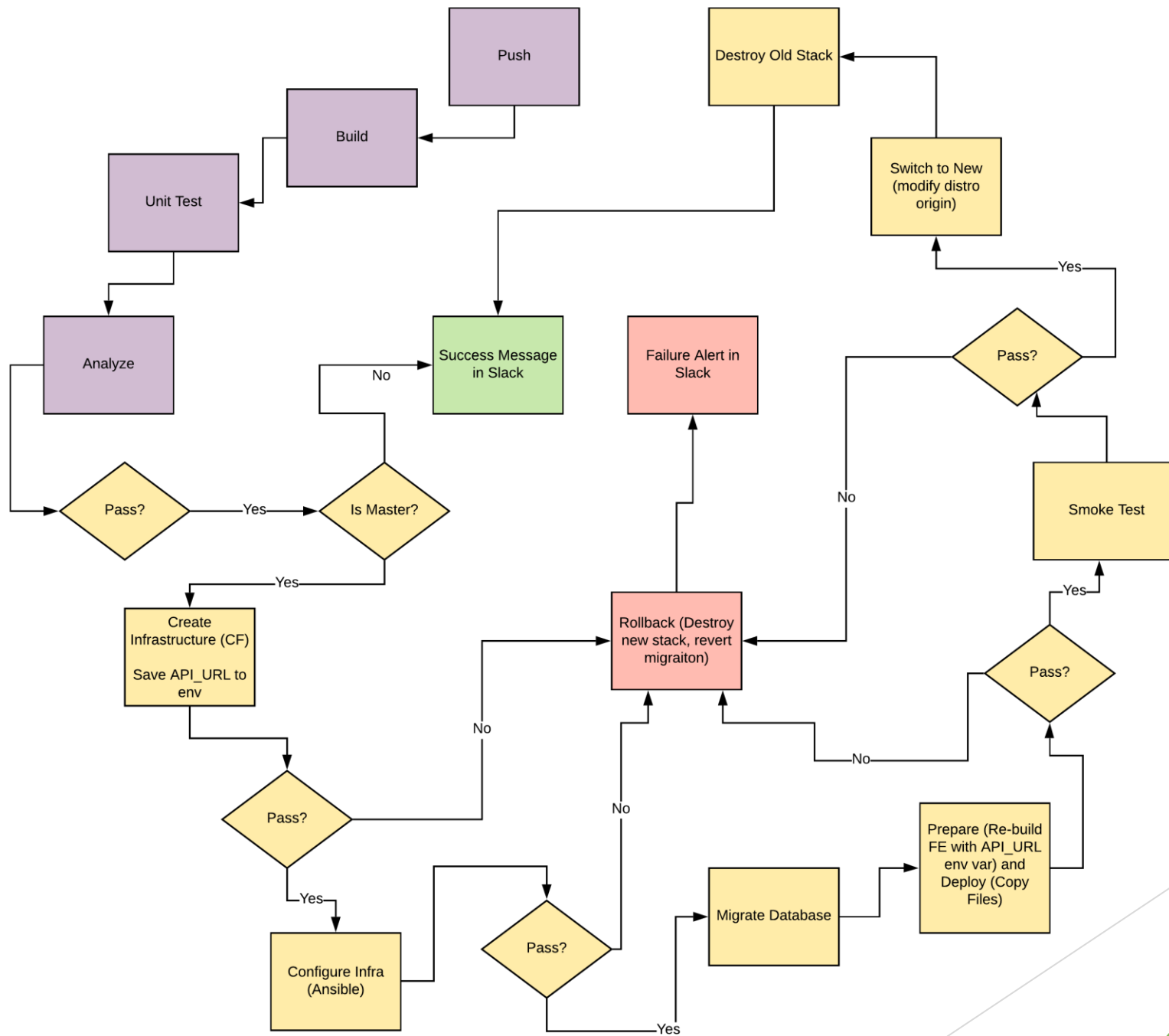# CICD adaptation Proposal

CONTINUOUS DELIVERY IMPLICATIONS •

- ▶ Reduce feedback loop with stakeholders
- ▶ More time adding real value and Less time configuring infrastructure
- ▶ Reduce costs detecting errors earlier
- ▶ Increase revenue with faster and more frequent production deployments
- ▶ Protect revenue with automated tests
- ▶ Avoid responsiblity silos with Continuous Delivery

Push

Build

Unit Test

Analyze

Destroy Old Stack

Switch to New (modify distro origin)

Success Message in Slack

Failure Alert in Slack

Pass?

Pass? — Yes → Is Master?

No → Success Message in Slack

Is Master? — Yes → Create Infrastructure (CF) Save API_URL to env

Rollback (Destroy new stack, revert migraiton)

Smoke Test

Pass? — Yes → Switch to New (modify distro origin)

No → Rollback

Pass? — No → Rollback

Pass? — Yes → Smoke Test

No → Rollback

Create Infrastructure (CF) Save API_URL to env

Pass? — No → Rollback

Pass? — Yes → Configure Infra (Ansible)

Configure Infra (Ansible)

Pass? — Yes → Migrate Database

No → Rollback

Migrate Database

Prepare (Re-build FE with API_URL env var) and Deploy (Copy Files)

PROPOSED PROCESS FLOW