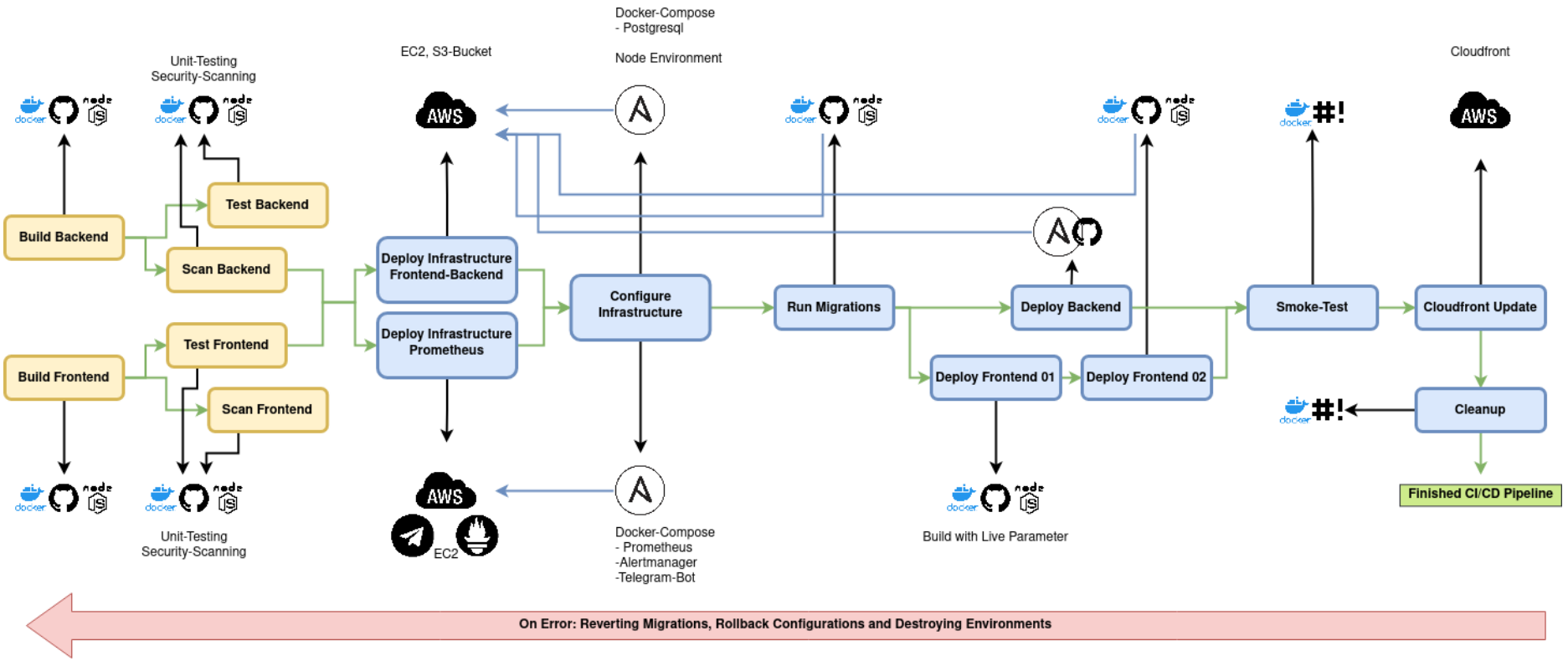
CI/CD Superpowers

How CI/CD will give your Software Development Rocket Speed.

David Lassig

17th January 2021

Why do this?


Unit-Testing
Security-Scanning
EC2, S3-Bucket
Docker-Compose
- Postgresql
Node Environment
Cloudfront
Build Backend
Test Backend
Scan Backend
Deploy Infrastructure
Frontend-Backend
Deploy Infrastructure
Prometheus
Configure
Infrastructure
Run Migrations
Deploy Backend
Smoke-Test
Cloudfront Update
Build Frontend
Test Frontend
Scan Frontend
Deploy Frontend 01
Deploy Frontend 02
Cleanup
Finished CI/CD Pipeline
Unit-Testing
Security-Scanning
EC2
Docker-Compose
- Prometheus
-Alertmanager
-Telegram-Bot
Build with Live Parameter
On Error: Reverting Migrations, Rollback Configurations and Destroying Environments

Commit to Deploy

Git Repository as the Single Source of Trust:

- any new Commit can trigger complete Build-Chain
- depending on Branch only Software-Build/-Test or complete Deployment
- only successful Builds will be merged to master

✅ Everything under Control

Only Commited Code exists in Production. No lost/abandoned files. No missing Code.

Security and Resiliency by Design

Building and Pipelining Language specific Testing and Security Scanning Tools

- Run Unit Tests
- Audit Source and Third Party Dependencies
- Fix Vulnerabilities
- only Continue if Tests and Scans success

✅ Everything under Control

Any piece of Source Code will be tested and scanned. Third party imports will be tested and scanned. On every Commit.

Infrastructure under Version Control

Use Easy Mergeble Configuration Files like yml or json:

- Hardware described in human readable and commitable files
- Easy Organisation of Engineering Teams by using Git Mechanics
- Only running Configurations get documented and into Master Repository

✅ Everything under Control

Easy Contribution of multiple Engineers. Changing configurations are 100% documented.

Measure Costs and Activities on Point

Cloudbased Applications are easily to monitor and provide very granular billing:

- Cloudbased Instances provide easy APIs for monitoring and alerting
- Costs are estimated by allocated Resources and only affects what you really used

✅ Everything under Control

Only successful Builds create Costs. All operation parameters are monitored. You will be alerted on exceptions.