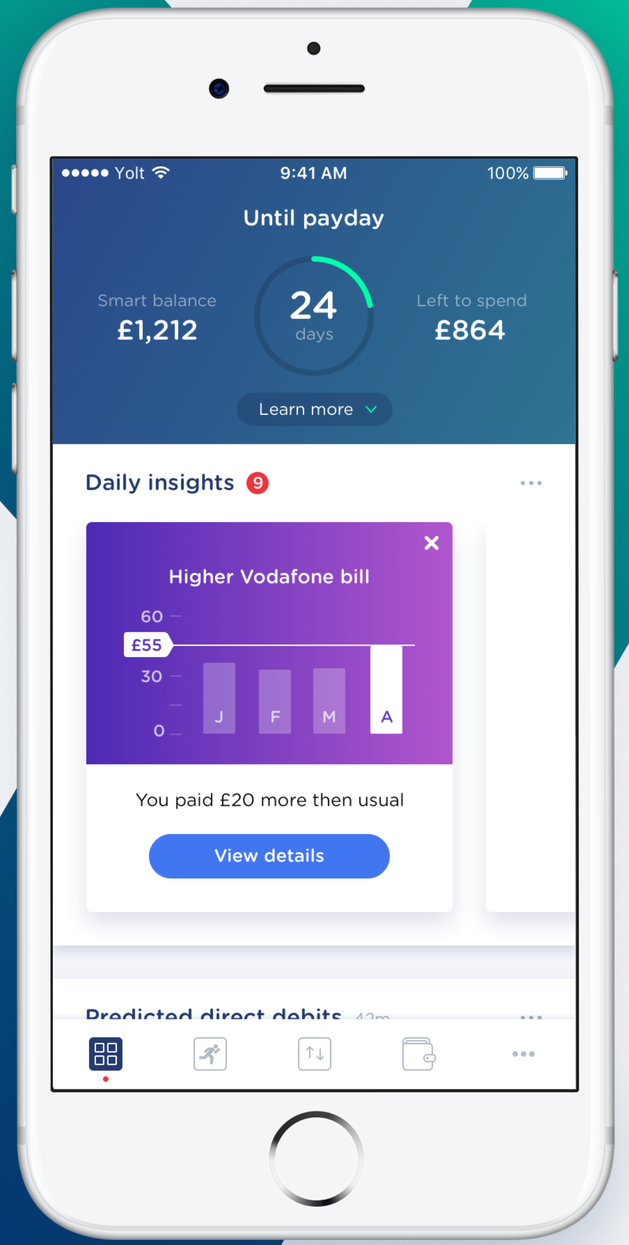
YOLT

Continuous delivery at Yolt Android

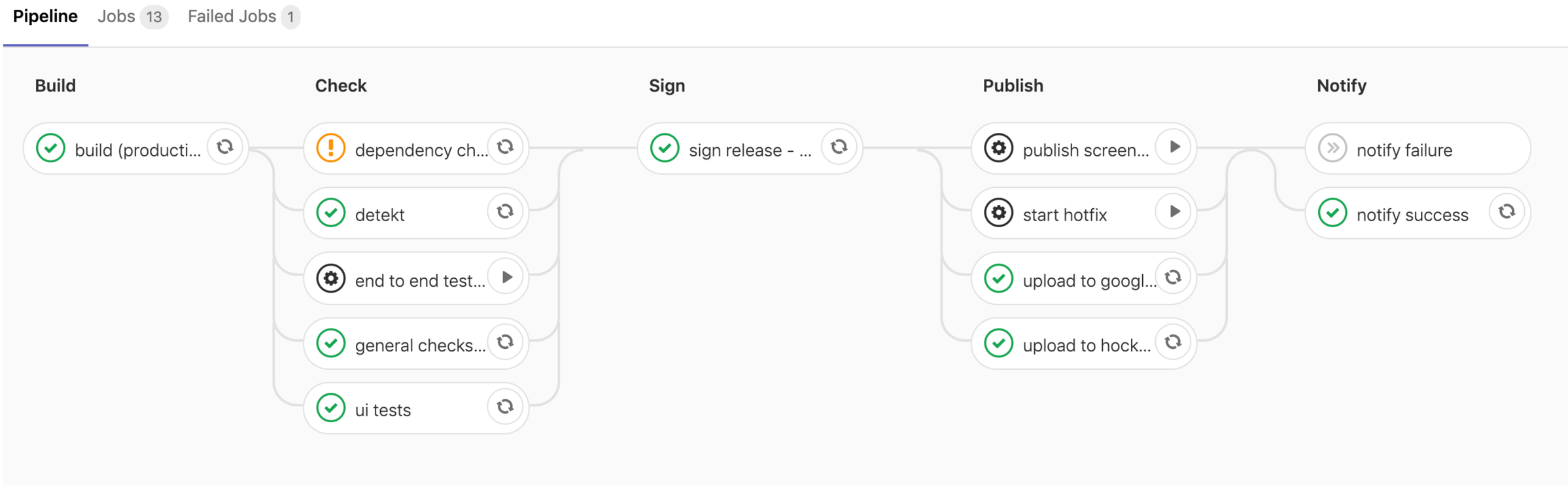
Meetup, September 19



Yolt

Money platform

- 1m installs and registration
- 1 non-full time QA for whole company
- 3.9 start in Google Play
- 99.4% crash free sessions
- 224k Kotlin and 18.5k Java lines
- 7700+ unit tests, 638 UI tests, 12 E2E tests
- 78.4% line coverage


Pipeline
Jobs 13
Failed Jobs 1
Build
Check
Sign
Publish
Notify
build (producti...
dependency ch...
detekt
end to end test...
general checks...
ui tests
sign release - ...
publish screen...
start hotfix
upload to googl...
upload to hock...
notify failure
notify success

Agenda

- Small incremental and correct parts
- Automation
- Current infra limitations and roadmap for the pipeline

Small, incremental and “correct” changes

Quiz

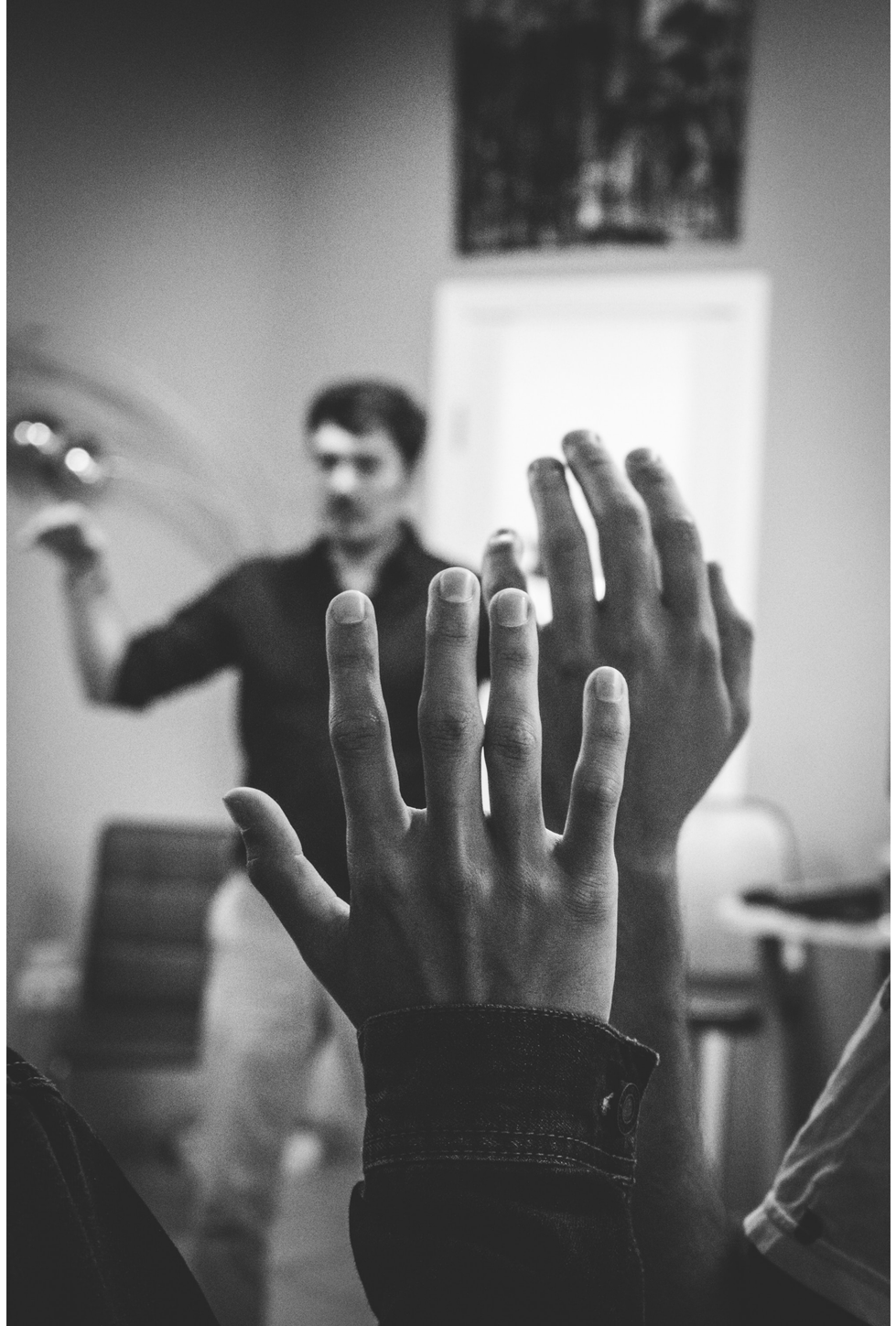


Photo by Artem Maltsev on Unsplash

Dependencies update

- We update dependencies every start of the sprint
- It is partially automated
- If dependency update brings incompatibility or require big changes we create technical story for the next sprint
- Some dependencies never update

```
minSdk                  : 21,
targetSdk               : 29,
compileSdk              : 29,
buildTools              : '29.0.2',

// fix @SuppressLint("DefaultLocale") with Kotlin 1.4.0
kotlin                  : '1.3.50',

androidGradlePlugin     : '3.5.0',
// DexGuard 8.5.05 should have ability to digest java8 code from kotlin (check DGD-1378)
dexGuardPlugin          : '8.6.00beta01',
jacocoGradlePlugin      : '0.8.2',
versionsGradlePlugin    : '0.22.0',
hockeyappGradlePlugin   : '3.6',
owaspGradlePlugin       : '5.2.1',
sonarQube               : '2.7.1',
googleServices          : '4.3.1',
gmsLocation             : '17.0.0',
detekt                  : '1.0.0',
```

Quiz

Photo by Ilkka Kärkkäinen on Unsplash

Feature toggles

- Allow you to break feature in smaller MRs
- Allow you to release unfinished or not “ready” things
- Allow you to A/B test features
- Allow you to switch off broken or impactful features
- Remove toggle and cleanup whenever they become obsolete

Question

Photo by Jon Tyson on Unsplash

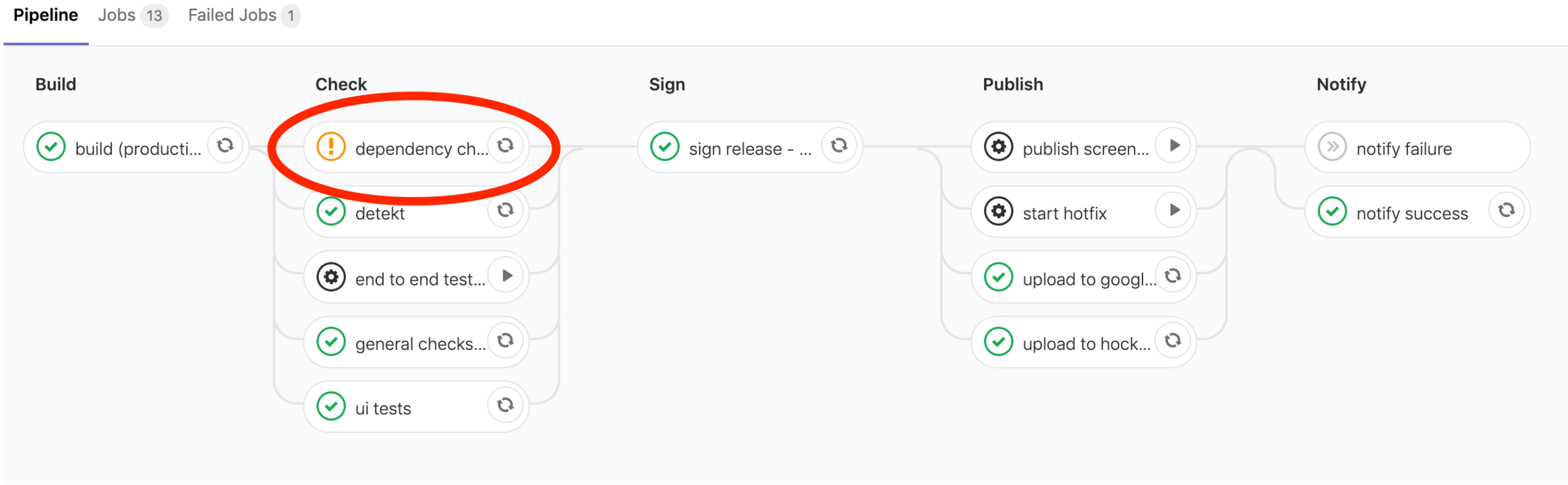
Definition of done

```
* [ ] Latest **dev** branch pulled into branch
* Testing:
    * [ ] Activities and fragments are covered with UI tests
    * [ ] Rest is covered with unit tests
    * [ ] Coverage is the same or better
    * [ ] End to end tests are OK
* New dependencies:
    * [ ] OWASP check completed
    * [ ] License checked and About page is updated
* Translations:
    * [ ] Copy check is done
    * [ ] Non-English translations are checked
* Analytics:
    * [ ] Events are implemented
    * [ ] Analytics are tested and aligned with the 'analytics owner'
* [ ] Obfuscated build is OK (run `./gradlew installDebugWithDexGuard`)
* [ ] No new Sonar issues
* [ ] UX has approved
* [ ] PO has approved
* [ ] CHANGELOG is updated
* [ ] Movie / screenshots done for sprint demo
```

Automation

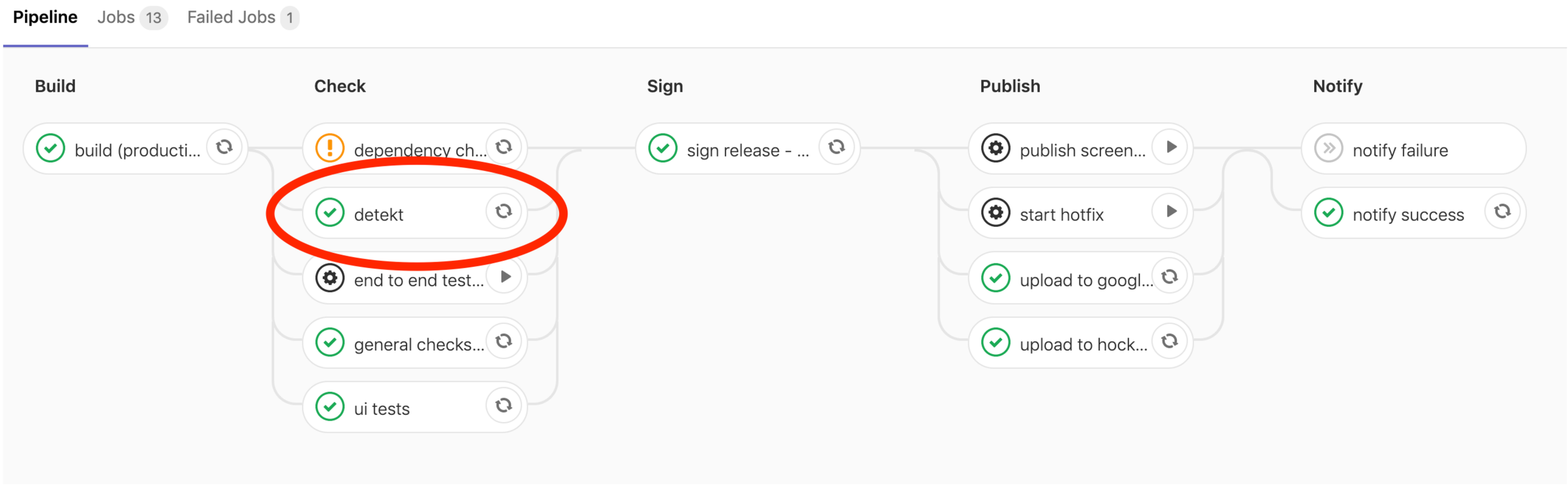
Check automations

OWASP check


Pipeline
Jobs 13
Failed Jobs 1
Build
Check
Sign
Publish
Notify
build (producti...
dependency ch...
detekt
end to end test...
general checks...
ui tests
sign release - ...
publish screen...
start hotfix
upload to googl...
upload to hock...
notify failure
notify success

```
owaspGradlePlugin  : '5.2.1'
owasp              : "org.owasp:dependency-check-gradle:$versions.owaspGradlePlugin"

buildscript {
  dependencies {
    classpath gradlePlugins.owasp
  }
}

apply plugin: 'org.owasp.dependencycheck'

dependencyCheck {
    // Fail the build when there is a finding with "medium" severity.
    failBuildOnCVSS = 4
    // Only check the compile classpath and ignore findings in other configurations.
    scanConfigurations = ['debugCompileClasspath', 'acceptanceCompileClasspath',
'productionCompileClasspath']
    suppressionFiles = ["${rootDir}/extra/owasp.suppression.xml"]

    analyzers {
      // disable .NET analysers to remove log warning
      nuspecEnabled = false
      nugetconfEnabled = false
      assemblyEnabled = false
      pathToDotnet = false
    }
}
```

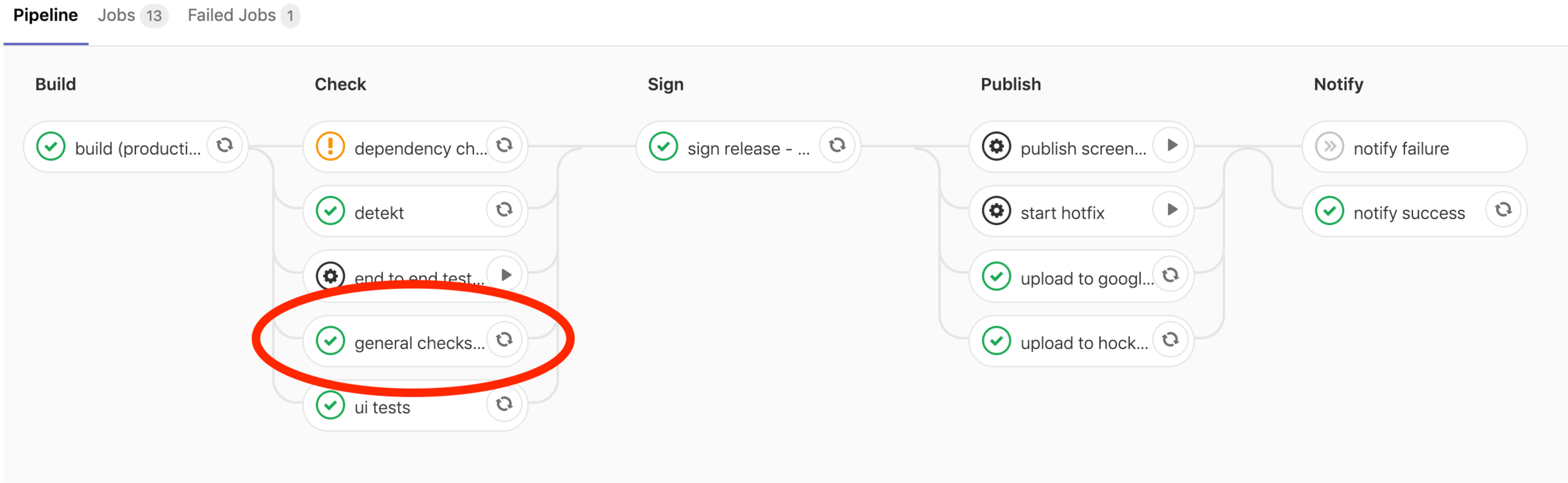
Detekt check



```
detekt  : '1.0.1'
detekt  : "io.gitlab.arturbosch.detekt:detekt-gradle-plugin:$versions.detekt"

buildscript {
  dependencies {
    classpath gradlePlugins.detekt
  }
}

apply plugin: 'io.gitlab.arturbosch.detekt'

detekt {
  version = versions.detekt
  buildUponDefaultConfig = true
  input = files(collectSourceDirs())
  config = files("$gradleDir/tools/detekt/detekt-config.yml")
  baseline = file("$gradleDir/tools/detekt/detekt-baseline.xml")
  parallel = true
}

/**
 * @return List of source directories from al subprojects.
 */
def collectSourceDirs() {
  def sourceDirs = []
  subprojects.each {
    sourceDirs << file("${it.projectDir}/src/main/java")
    sourceDirs << file("${it.projectDir}/src/test/java")
    sourceDirs << file("${it.projectDir}/src/androidTest/java")
    sourceDirs << file("${it.projectDir}/src/debug/java")
    sourceDirs << file("${it.projectDir}/src/acceptance/java")
    sourceDirs << file("${it.projectDir}/src/release/java")
  }
  return sourceDirs.findAll { it.exists() }
}
```

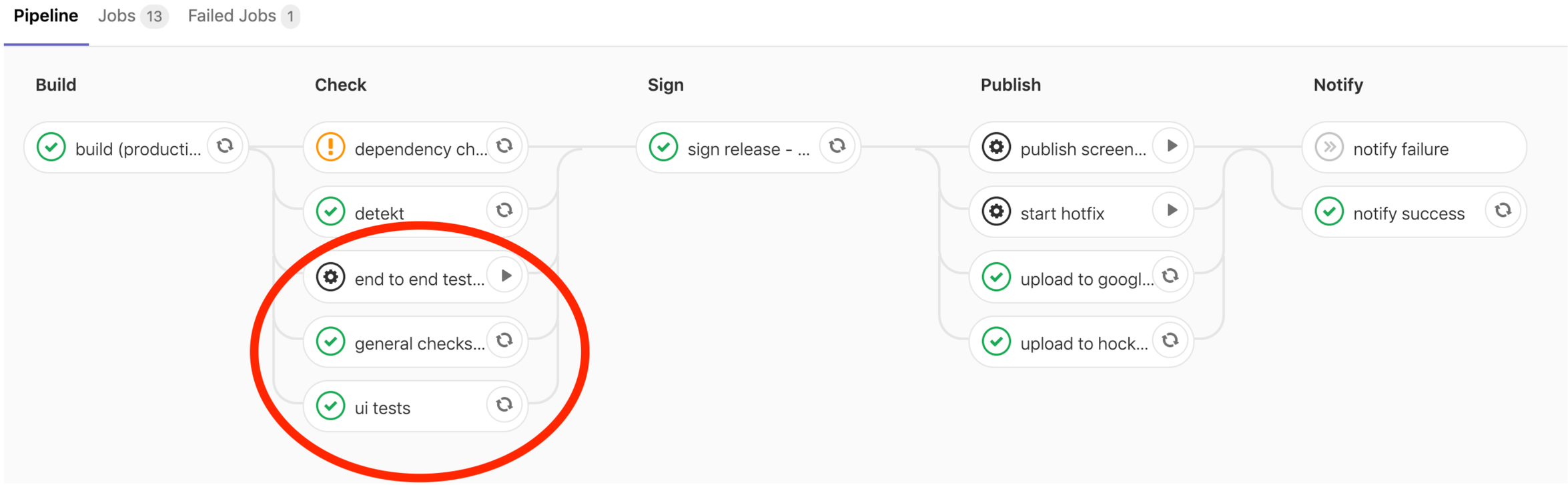
```yaml
autoCorrect: false

build:
  maxIssues: 0 # Fail the build if any issues are found.

complexity:
  active: true
  TooManyFunctions:
    active: true
    thresholdInFiles: 15 # Allow more methods in files than the default configuration.
    thresholdInClasses: 15 # Allow more methods in classes than the default configuration.
    excludes: "**/test/**;**/androidTest/**"
  LongParameterList:
    active: true
    excludes: "**/test/**;**/androidTest/**"
  LongMethod:
    active: true
    excludes: "**/test/**;**/androidTest/**"
  LargeClass:
    active: true
    excludes: "**/test/**;**/androidTest/**"
  ComplexMethod:
    active: true
    excludes: "**/test/**;**/androidTest/**"

style:
  active: true
```

Lint check


Pipeline
Jobs 13
Failed Jobs 1
Build
Check
Sign
Publish
Notify
build (producti...
dependency ch...
detekt
end to end test...
general checks...
ui tests
sign release - ...
publish screen...
start hotfix
upload to googl...
upload to hock...
notify failure
notify success

```
android {

  lintOptions {
    abortOnError true
    warningsAsErrors true
    lintConfig file('lint.xml')

    // Also check all other modules (should be faster and more accurate than linting each module
individually).
    checkDependencies true

    // Enable output to the console so we can quickly spot new issues from the CI logs.
    textReport true
    textOutput 'stdout'
  }

}
```

Automated tests



Unit tests

- JUnit 5
- AssertJ (Trust)
- Mockito and KotlinMockito
- Robolectric for legacy tests
- We use experimental mockito option to mock final classes
- Run on every push to remote

UI and e2e tests

- AndroidX test libraries
- Espresso
- Barista for flaky test rules only
- Composer (improvement of the Square Spoon)
- Flacon to take screenshots

UI and e2e tests

- We use robot pattern for UI and e2e tests
- We mock server responses, hardware (like fingerprint) in UI tests
- UI tests run on small screen android emulators. They are shared on three emulator instances
- **avdmanager** to create image and **emulator** to run
- We disable animations on the emulator start
- We wrote own runner to inject mocked things, add screenshot failure handler
- We wrote own test rule to mock network request/responses
- We wrote couple of idling resources
- We restart once UI tests if they fail
- We set super small timeouts like splash screen timeout, etc
- UI tests run on every push to repo
- UI tests are only in the app module

SUITES LIST/ SUITE 0

Passed	Failed	Ignored	Duration
638	0	15	10:14.467

Search

Reset

TESTS 653

testPinTried8TimesShowsBlockedFor1Hour
ChangePinTests
com.yolt.android.tests
Emulator-5554 Android SDK Built For X86
0:14.012

testChangePin
ChangePinTests
com.yolt.android.tests
Emulator-5554 Android SDK Built For X86
0:13.005

showCountrySelectorFromButton
LandingTests
com.yolt.android.tests
Emulator-5554 Android SDK Built For X86
0:2.002

e2e tests

- Use same robots as UI tests
- Only hardware mocks, the rest is real app connecting to real BE
- Cover only critical flows - registration, add a bank, user deletion, transaction re-categorisation
- Use quick registration when is suitable
- Clean up created user record as test is finished

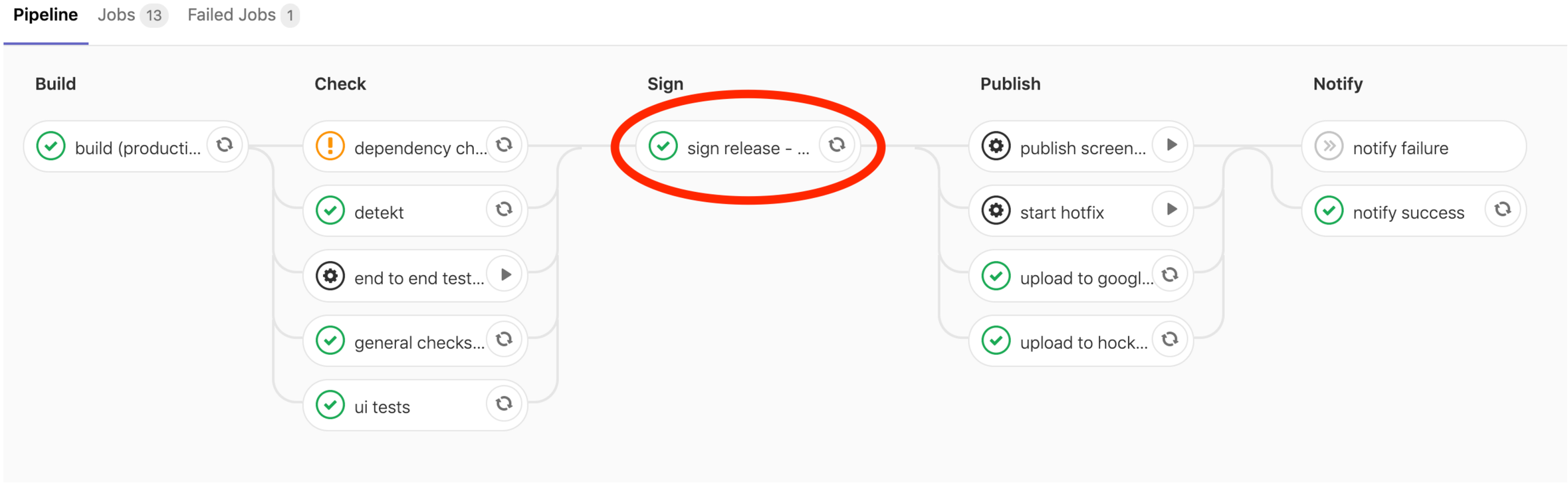
```
tasks.withType(JavaCompile).configureEach {
  if (isCIBuild) {
    // treat all warnings as errors
    options.compilerArgs += '-Werror'
  }
}

tasks.withType(org.jetbrains.kotlin.gradle.tasks.KotlinCompile).all {
  kotlinOptions {
    kotlinOptions.allWarningsAsErrors = isCIBuild
  }
}
```

Other automation

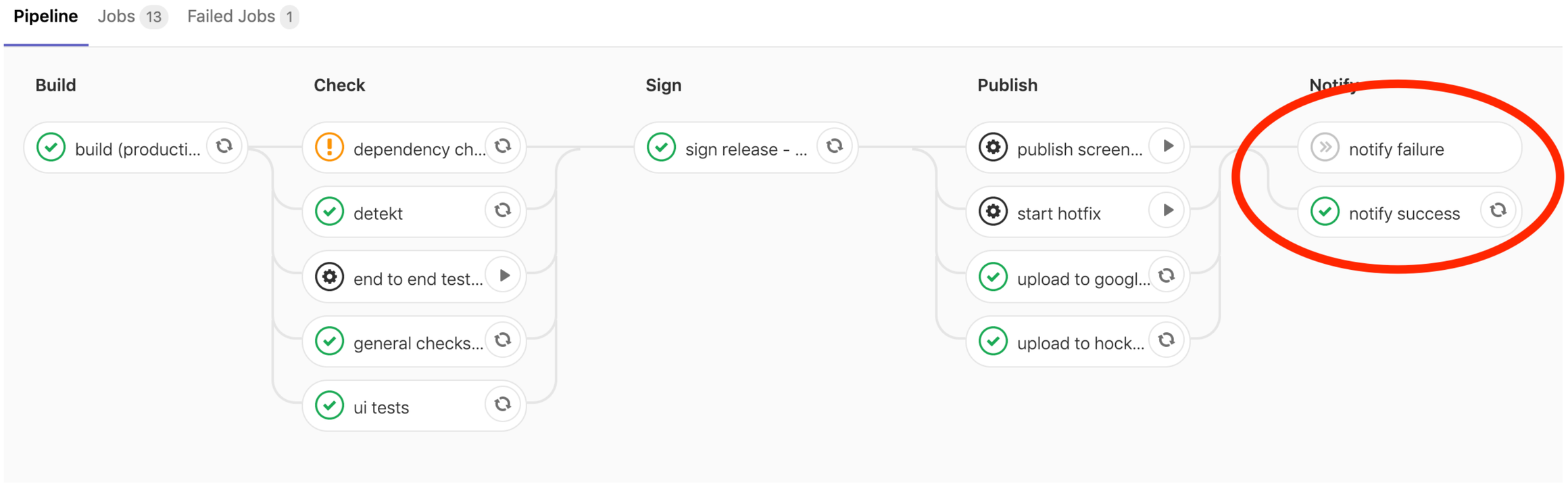
General

- Use gradle as automation is OK
  - Every script defined in project require Android SDK presence
  - There are limitation of the orphan gradle scripts in latests versions
- Use bash scripts
- Use kotlin script?

Sign automation

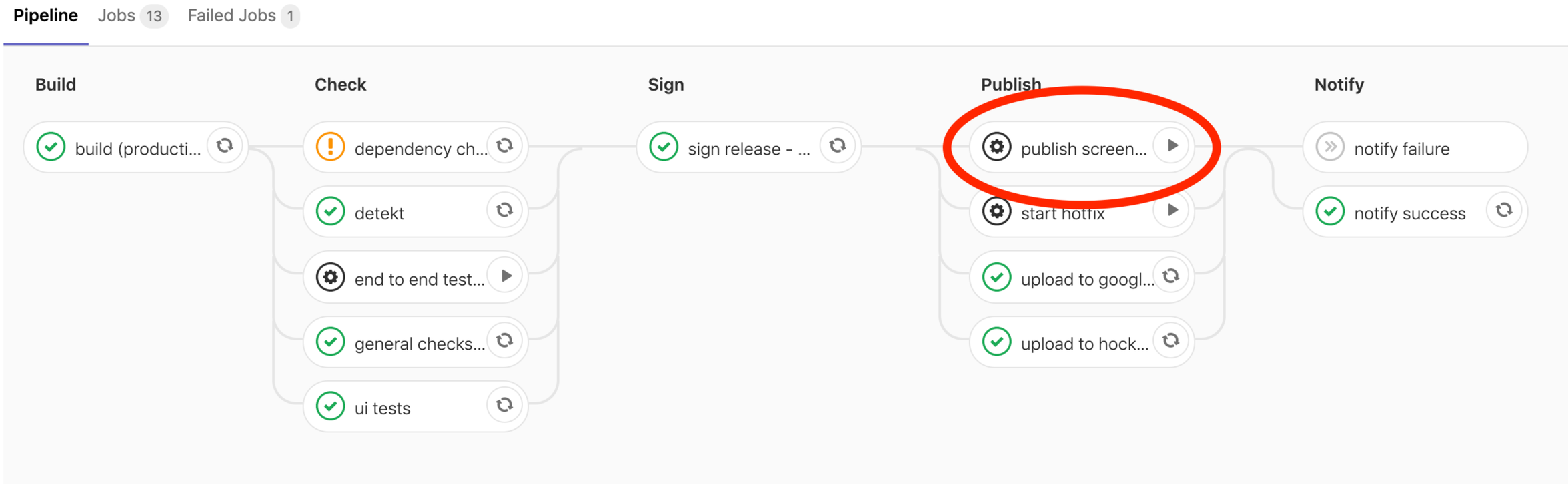


Notifications

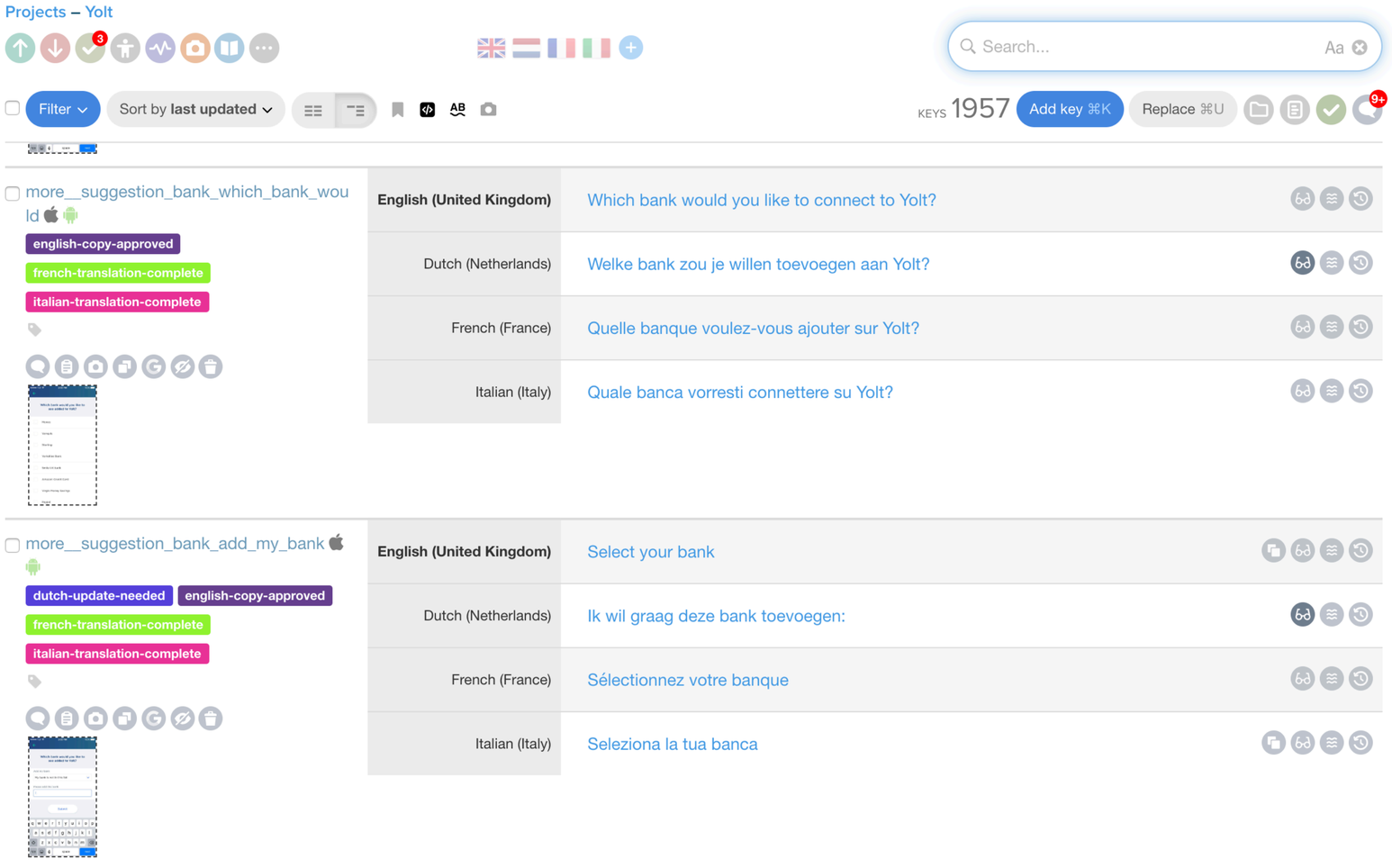


Git automation

Screenshots automation



Download localisations



Infra and current problems

Current infra

- Infra
  - 2 Macs, 1 mac mini and 1 powerful linux machines
  - Pull of “weak” docker runners in AWS
- We user remote cradle build cache to reuse tasks output
- We record build times but no much checks on this data
- It is nightmare to troubleshot disconnections
- Managing updates is also painful

Infra and automation roadmap

- Dockerise the rest of the jobs
  - Utilise AWS updated pull
  - Use third party provider (like Bitrise)
- Plot collected build times in the human consumable view
- Use of third party for emulators (Firebase Test Lab)
- Use Google distributed signing
- Automate “hard” checks - new permissions, new broadcast receivers, etc
- Automate build performance checks
- Automate dependencies update
- Return automatic commenting findings on MRs