Apache RocketMQ 持续集成

李维军

2018.07.29

目 录

* 持续集成测试框架
* Jenkins
* Sonar
* Committer规约

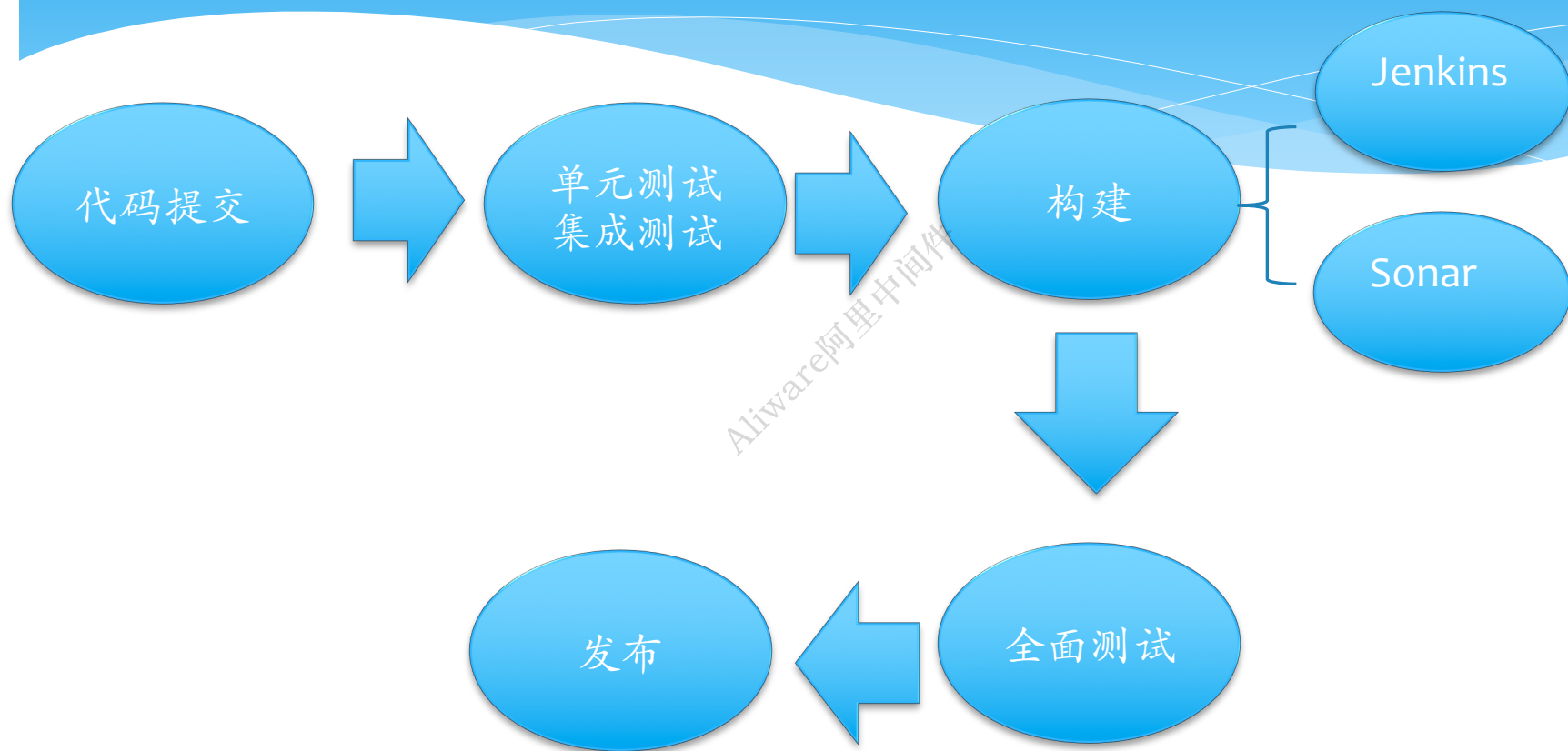# 持续集成测试框架

* 持续集成简介
* Apache RocketMQ的持续集成模型
* Apache RocketMQ的持续集成数据

持续集成简介

* 概念：频繁地将代码集成到主干
* 好处：快速发现错误、防止分支大幅偏离主干
* 目的：高质量地快速迭代
* 核心：自动化测试
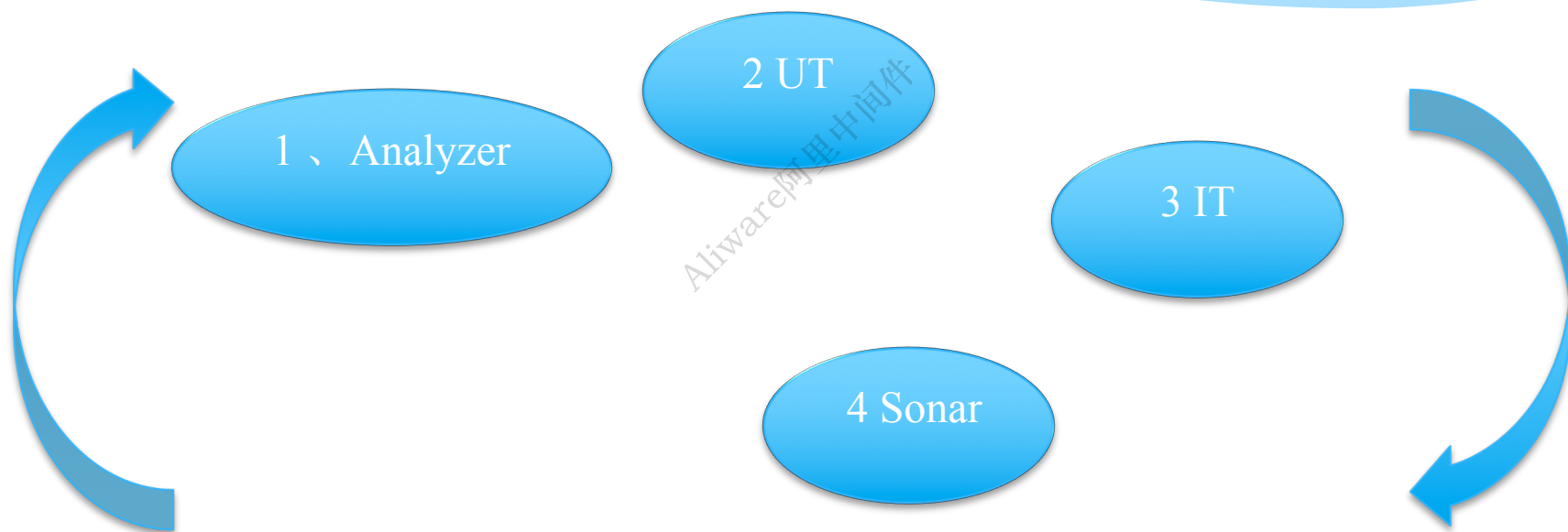* 持续集成并不能消除bug，而是让它们非常容易发现和改正

Apache RocketMQ的持续集成模型

代码提交 → 单元测试 集成测试 → 构建

Jenkins

Sonar

全面测试 ← 构建

发布 ← 全面测试

Jenkins

* Jenkins简介
* Jenkins在Apache RocketMQ中的部署应用模型

Jenkins简介

* 功能强大的java开源持续集成工具
* 图形界面化显示
* 监控持续集成构建错误
* 详细的日志和提醒功能
* 丰富的插件支持

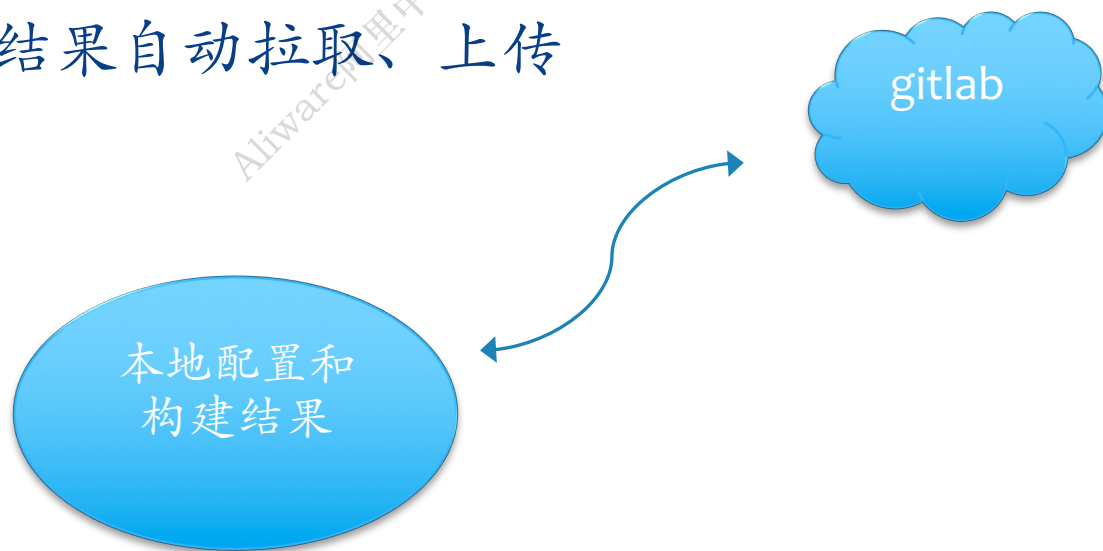# Jenkins在Apache RocketMQ中的部署应用模型

1、Analyzer

2 UT

3 IT

4 Sonar

15min周期，4个executor

Jenkins在Apache RocketMQ中的部署应用模型
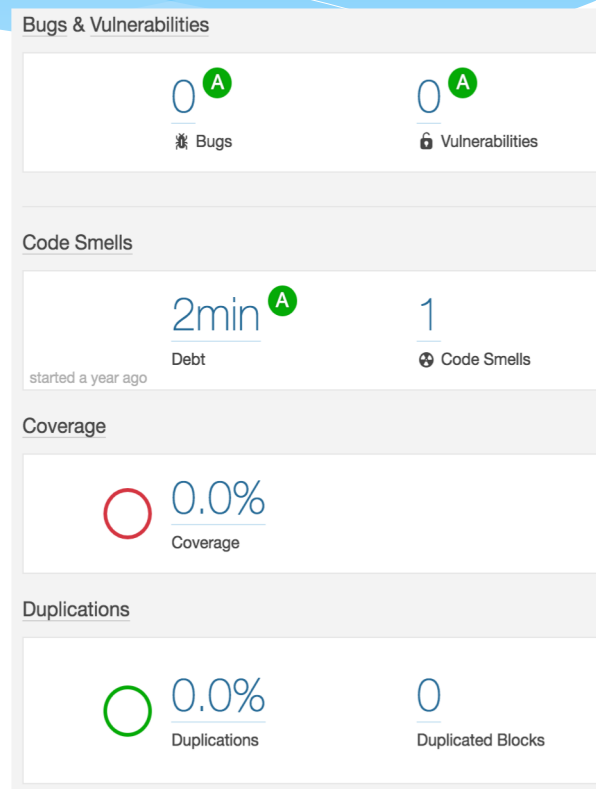
* 构建过程Jacoco覆盖率测算
* 配置和构建结果自动拉取、上传

gitlab

本地配置和构建结果

Sonar

* Sonar简介
* Sonar在Apache RocketMQ中的部署应用模型

Sonar简介

* 开源的代码质量管理平台
* 七个维度（复杂度分布、重复、单测缺乏、代码标准、注释、潜在的 bug、Spaghetti design）
* 二十多种语言支持（java、C/C++、 JS…）

Committer规约

* Issue规约
* Pull requests规约

Issue 规约

* **The issue tracker is **ONLY** used for bug report and feature request. Keep in mind, please check whether there is an existing same report before your raise a new one.**

* ****BUG REPORT****

* **1. Please describe the issue you observed:**

* **- What did you do (The steps to reproduce)?**

* **- What did you expect to see?**

* **- What did you see instead?**

* **2. Please tell us about your environment:**

* **3. Other information (e.g. detailed explanation, logs, related issues, suggestions how to fix, etc):**

Issue规约

* **FEATURE REQUEST**

  * 1. Please describe the feature you are requesting.

  * 2. Provide any additional detail on your proposed use case for this feature.

  * 3. Indicate the importance of this issue to you (blocker, must-have, should-have, nice-to-have). Are you currently using any workarounds to address this issue?

  * 4. If there are some sub-tasks using -[] for each subtask and create a corresponding issue to map to the sub task:

  * - [sub-task1-issue-number](example_sub_issue1_link_here): sub-task1 description here,
  * - [sub-task2-issue-number](example_sub_issue2_link_here): sub-task2 description here,
  * - ...

Pull requests 规约

* Please do not create a Pull Request without creating an issue first.
* ## What is the purpose of the change
* ## Brief changelog
* ## Verifying this change

* Follow this checklist to help us incorporate your contribution quickly and easily.

* - [x] Make sure there is a Github issue filed for the change (usually before you start working on it). Trivial changes like typos do not require a Github issue. Your pull request should address just this issue, without pulling in other changes - one PR resolves one issue.
* - [ ] Format the pull request title like `[ISSUE #123] Fix UnknownException when host config not exist`. Each commit in the pull request should have a meaningful subject line and body.
* - [ ] Write a pull request description that is detailed enough to understand what the pull request does, how, and why.
* - [ ] Write necessary unit-test(over 80% coverage) to verify your logic correction, more mock a little better when cross module dependency exist. If the new feature or significant change is committed, please remember to add integration-test in [test module](https://github.com/apache/rocketmq/tree/master/test).
* - [ ] Run `mvn -B clean apache-rat:check findbugs:findbugs checkstyle:checkstyle` to make sure basic checks pass. Run `mvn clean install -DskipITs` to make sure unit-test pass. Run `mvn clean test-compile failsafe:integration-test` to make sure integration-test pass.
* - [ ] If this contribution is large, please file an [Apache Individual Contributor License Agreement](http://www.apache.org/licenses/#clas).

Thanks