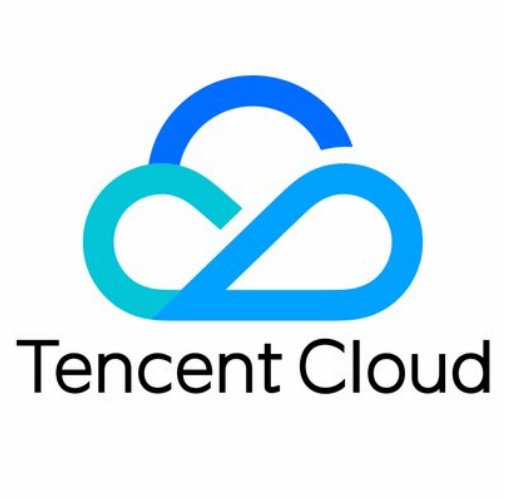
CODING Continuous Integration

Introduction

Product Documentation



Contents

Introduction
- Overview
- Benefits
- Scenarios

Introduction
Overview

Last updated：2023-12-29 11:44:51

Continuous Integration Overview

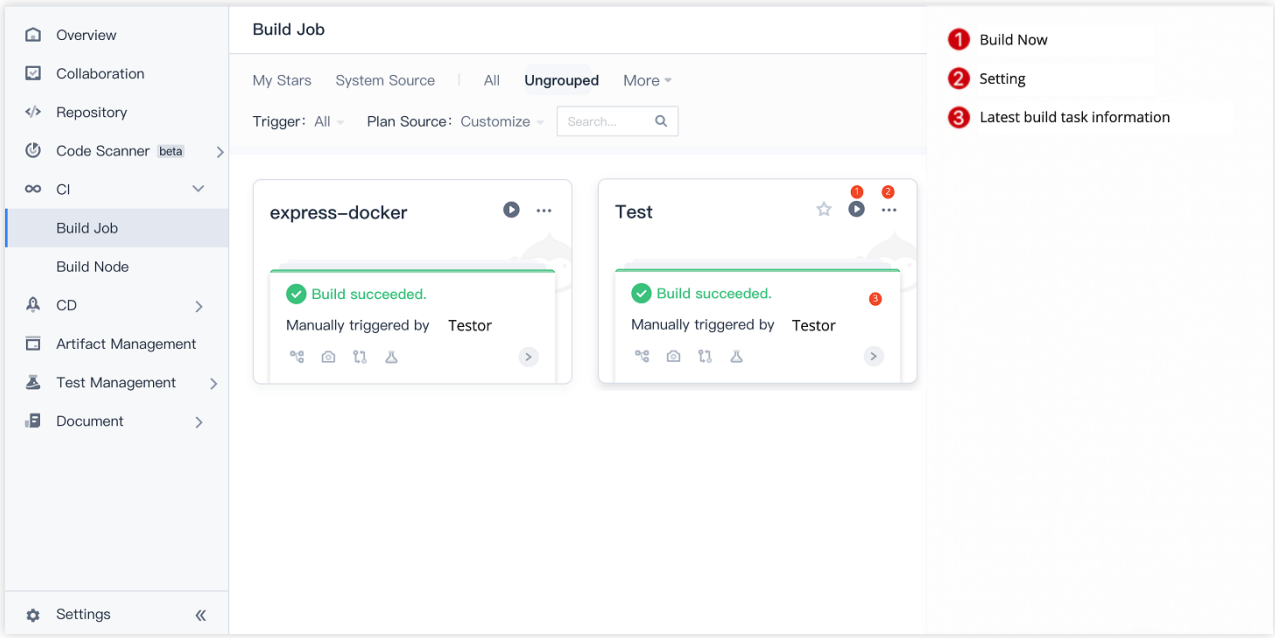
Continuous integration (CI) plays an important role in modern software development. In normal projects, some of the work is mechanical and repetitive, but prone to errors (such as during packaging and deployment). Users can instruct machines to complete this work with the click of a mouse and then take a break while the CI build plan automatically performs unit testing, code inspections, compilation and building, contract testing, and even deployment. This significantly reduces the workload of developers while improving code quality and development efficiency.

CODING Continuous Integration (CODING-CI)

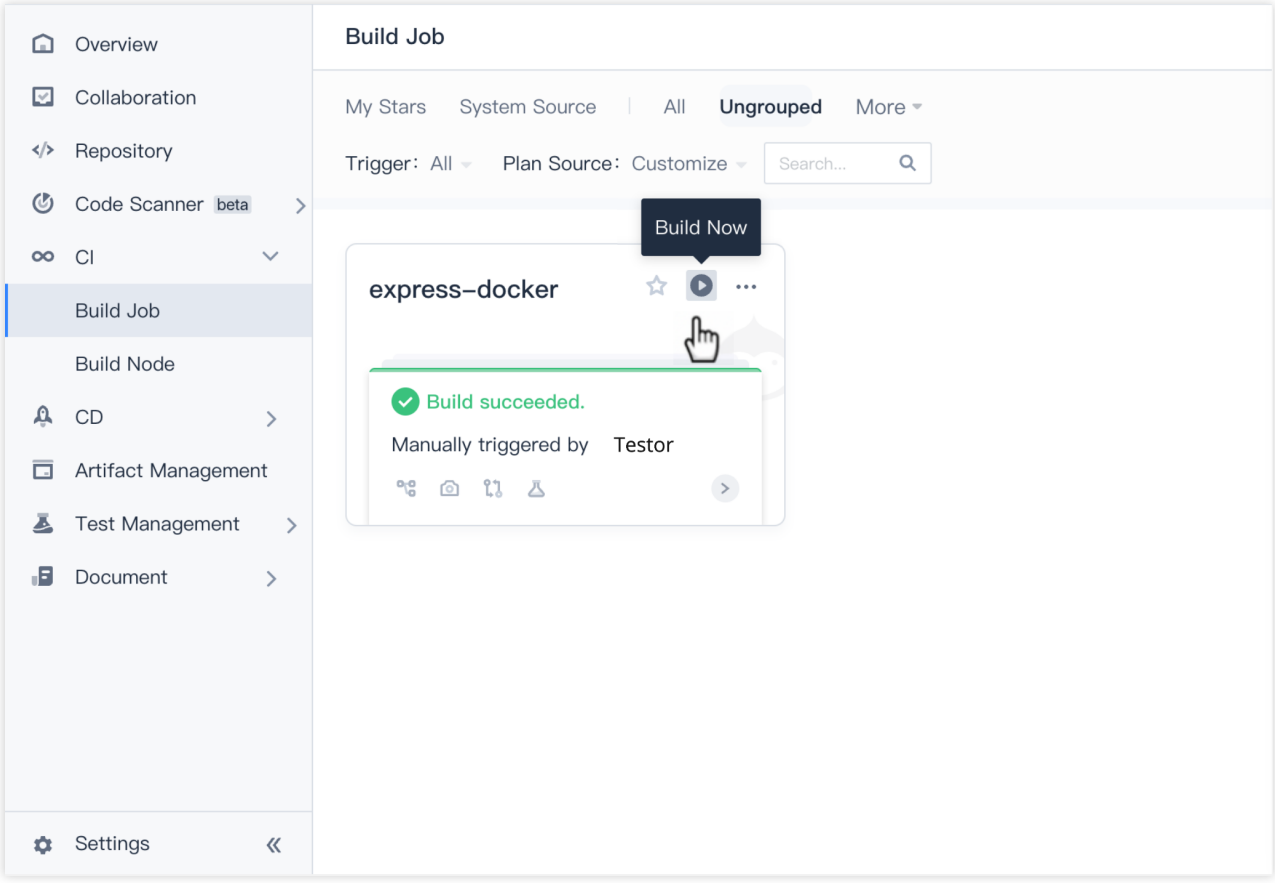
CODING Continuous Integration (CODING-CI) is fully compatible with Jenkins' continuous integration services. It supports popular programming languages, such as Java, Python, and Node.js, and building Docker images. Its graphical orchestration, high-spec clusters, and multi-plan parallel builds help you comprehensively accelerate your build tasks. It supports mainstream Git code repositories, including CODING Code Repositories (CODING-CR), GitHub, and GitLab. In terms of build dependency pull, it has dedicated network optimizations for major image sources such as Maven and npm to ensure a high pull speed and further accelerate the build process.

Function Guide

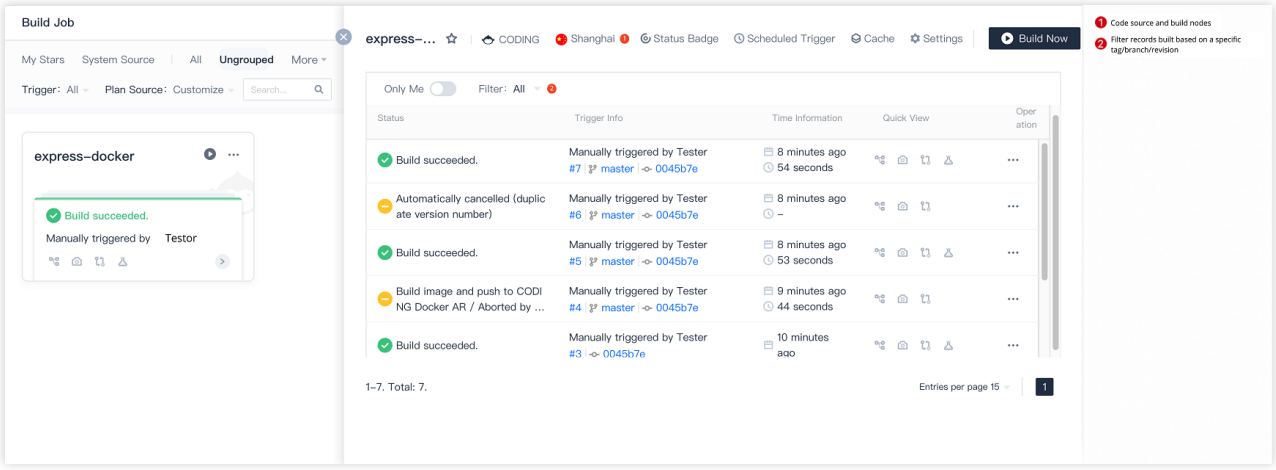
The core components of continuous integration are build plans, which are displayed in cards. The buttons are described below:



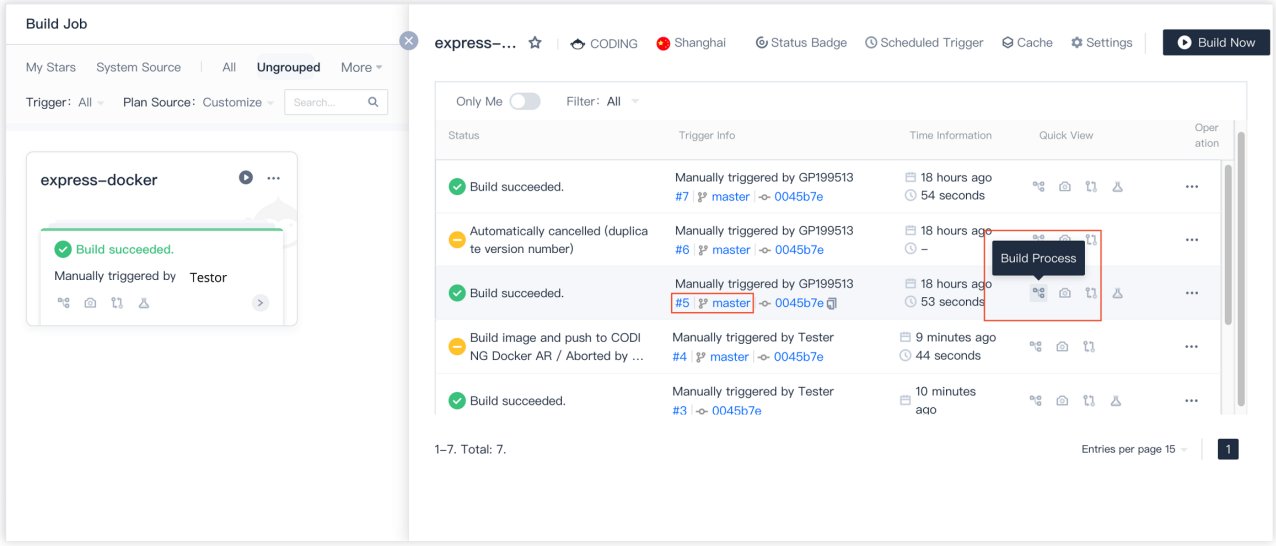
Hover your cursor over a chart to display button descriptions.



On the build records page, users can conveniently set build plans and filter build records.



In a single build record, you can follow the link to quickly go to the build branch and revision version. In the **Quick View** area, you can follow the link to view the build process, build snapshot, and change history.



Build plans

Build plans (jobs) are the basic units of continuous integration. You can open a build plan to set its code source, build process, trigger rules, environment variables, and notifications. In the future, this plan will be triggered based on the set rules to implement an automated build pipeline.

Build tasks

After configuring a build plan, each build execution generates a specific build task. You can go to a build task to see its build process, change history, test reports, build artifacts, build snapshot, and other execution information.

Jenkinsfiles

A Jenkinsfile defines a continuous integration pipeline, which implements stream encapsulation and management of steps. Pipelines are the basic units in continuous integration. Pipelines can be executed in series or parallel.

Benefits

Last updated：2023-12-29 11:44:50

Benefits of CODING Continuous Integration (CODING-CI):

Comprehensive build types

In addition to Docker images, CODING-CI also supports the builds of JAR, APK, and other software packages. Plus, it has many preset build environment images such as Java, Python, and Node.js.

Parallel execution of multiple build plans

CODING-CI supports parallel builds of single projects to meet the high-level continuous integration needs of users. Its backend server cluster can schedule responsive computing resources according to user needs, ensuring that build tasks start quickly and spend less time queuing.

Cache acceleration

During the continuous integration and build process, the repetitive downloading of dependent files may result in a long build time. CODING-CI supports caching between different build tasks to accelerate repetitive builds by an average of 300%.

Graphical orchestration

In addition to manual build script editing, CODING-CI also offers high-quality graphical orchestration capabilities for greater ease of use. For each step of the build, it offers various build script templates that you can choose from, delivering an intuitive editing experience.

Full compatibility with Jenkins

The build scripts of CODING-CI have full syntactical compatibility with Jenkins, the most widely used continuous integration tool in the world, which means that you can seamlessly and easily migrate Jenkins builds to CODING.

Scenarios

Last updated：2023-12-29 11:44:51

Auto building

CODING Continuous Integration (CODING-CI) can be used to automatically build code. Based on the user-defined build environment and build script instructions, it can quickly build the source code into runnable artifacts with the aid of the high computing and networking power of the cloud and store them in build version repositories.

Supplementary review

CODING-CI makes complicated yet important supplementary code review easier. It can perform prep work such as automated building, automated testing, style checks, quality scanning, and security assessments before the review starts, which greatly reduces the complexity and difficulty of the review work.

Automated testing

DevOps requires a high degree of automation for the software delivery process, in which automated testing is particularly important. CODING-CI can participate in automated tests at project release, code review, project acceptance, and other stages to improve the efficiency of the delivery process.