CODING Continuous Integration

Getting Started

Product Documentation

Getting Started

Last updated：2023-12-29 11:44:50

This document demonstrates how to use CODING Continuous Integration (CODING-CI) templates to deploy Node.js + Express + Docker applications.

Prerequisites

To use CODING-CI, you must activate the CODING DevOps service for your Tencent Cloud account.

Open Project

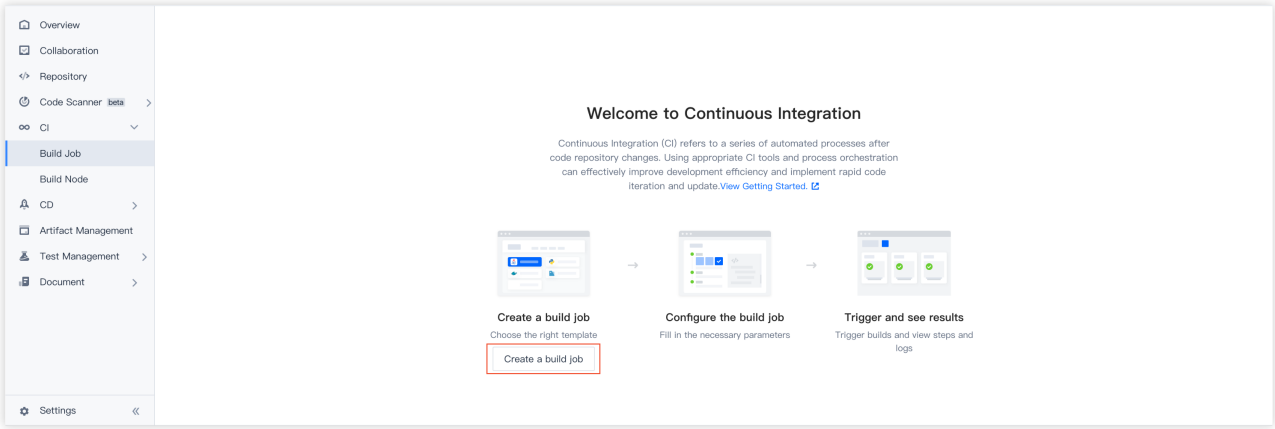
1. Log in to the CODING Console and click the team domain name to go to CODING.
2. Click

in the upper-right corner to open the project list page and click a project icon to open the corresponding project.
3. Select **Continuous Integration** from the menu on the left.

Getting Started

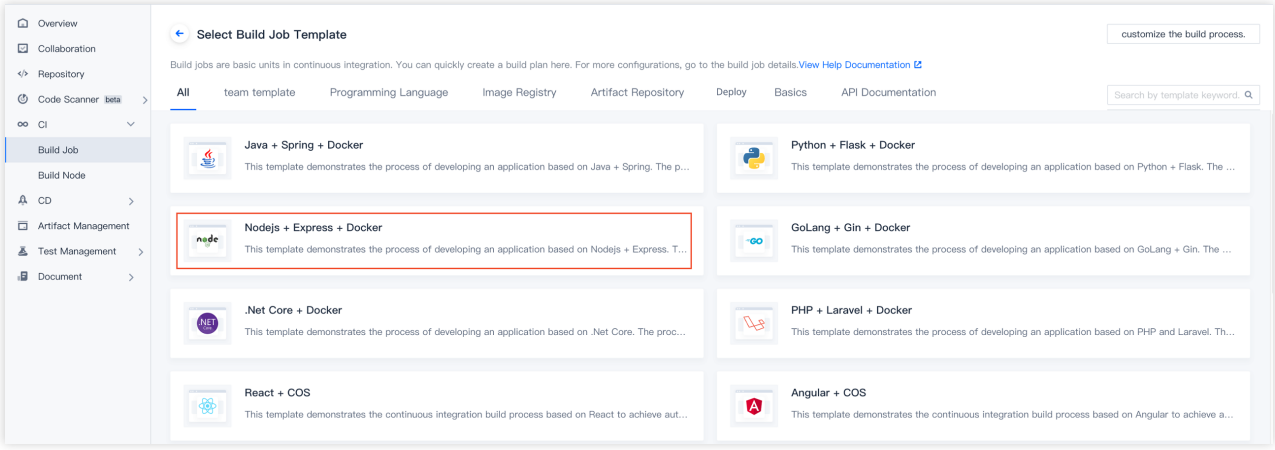
1. Create build plan

After opening a project, select **Continuous Integration** on the left of the page, and then click **Build Plans** > **Create Build Plan**. With this plan, we will demonstrate the following based on Node.js + Express: Implement Auto Code Checkout > Unit Testing > Build Docker Image > Push to Docker Artifact Repository > Deploy to Remote Server (optional).



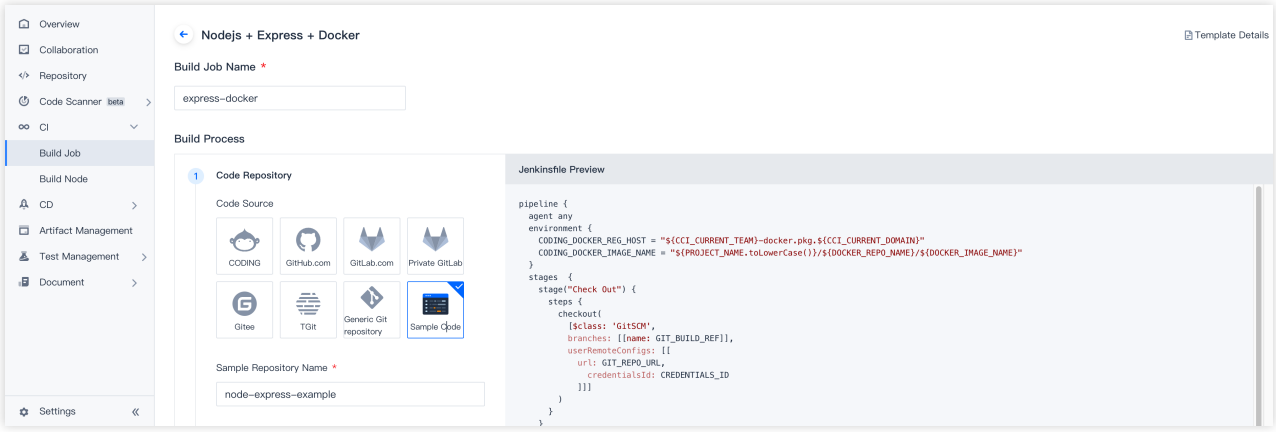
2. Select continuous integration template

Select a Node + Express + Docker continuous integration (CI) template.



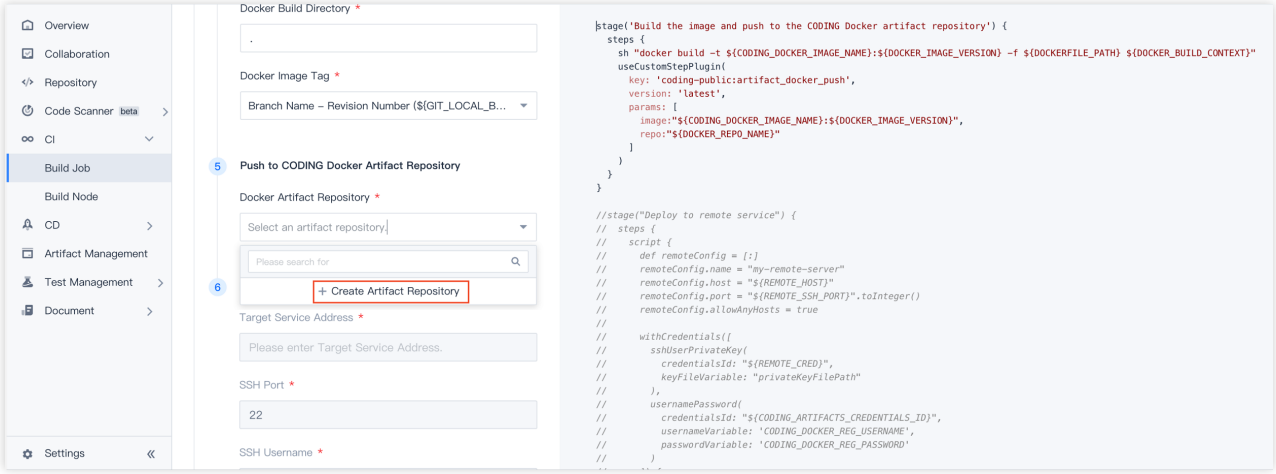
3. Select code source

In the example project, we recommend you select **Sample Code** as the code source in the Code Repository field. Then, the system will automatically establish a sample code repository in your project. You can also select an existing code repository in a custom build plan.



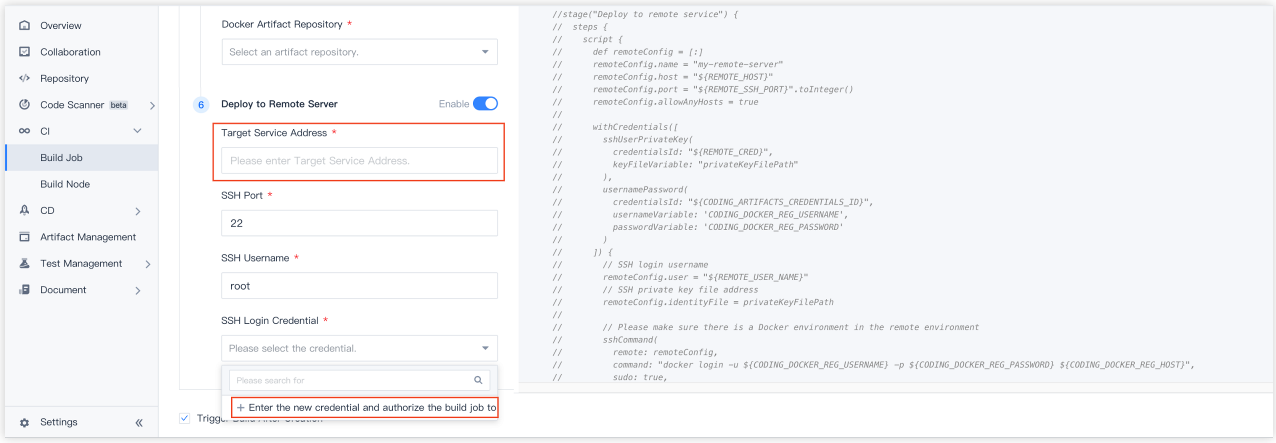
4. Select CODING Docker artifact repository

After the build plan is completed, the system generates a build result. Here, you can select the CODING Docker artifact repository that the results are pushed to. If there is no artifact repository, you can quick create one.

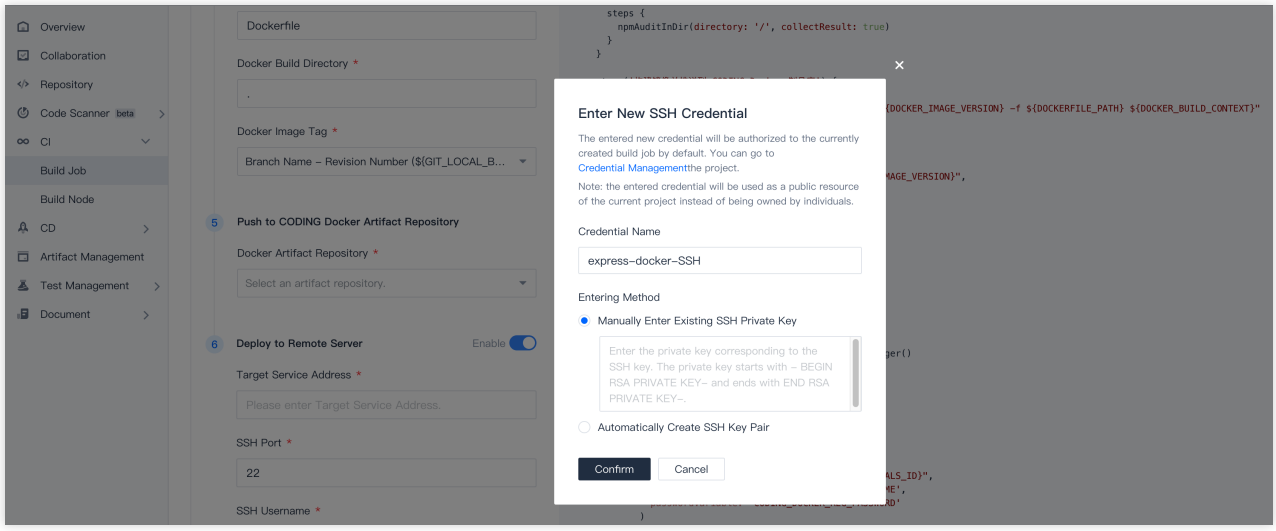


5. Enter remote server information (optional)

Enter the information of the remote server for deployment, including its IP address and port and SSH login credentials. After making sure this information is correct, wait for the build plan to be completed, after which the artifacts are sent to the remote server. You can preview the release result at a URL. If you do not need to deploy to a remote server, you can skip this step.



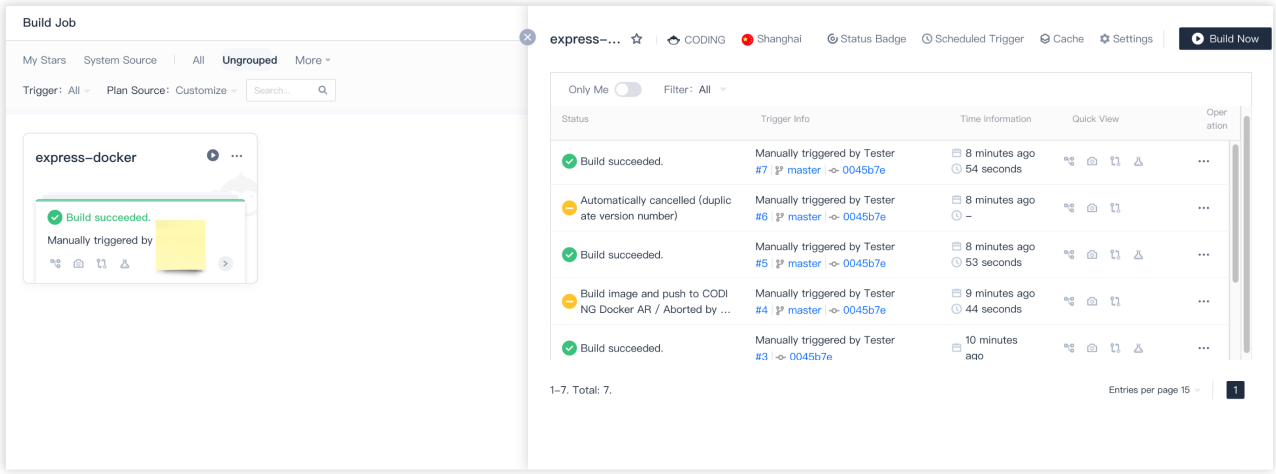
After you click **Enter a new credential and authorize**, if you use an SSH private key to log in to the remote server, just click **Manually enter existing SSH private keys** for the entry method. After entering the key information, you can view it in **Project Settings** > **Developer Options** > **Credential Management**.



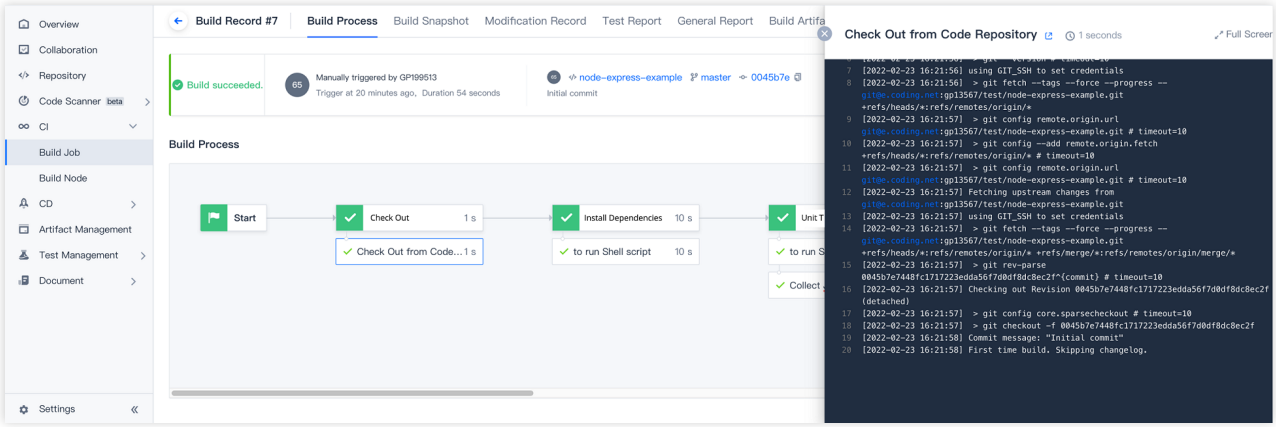
If you do not know how to log in to a remote server with an SSH private key, click **Automatically create an SSH key pair**. Then, you must manually set the public key in the `~ssh/authorized_ keys` folder of the remote server.

6. Click create and view build result

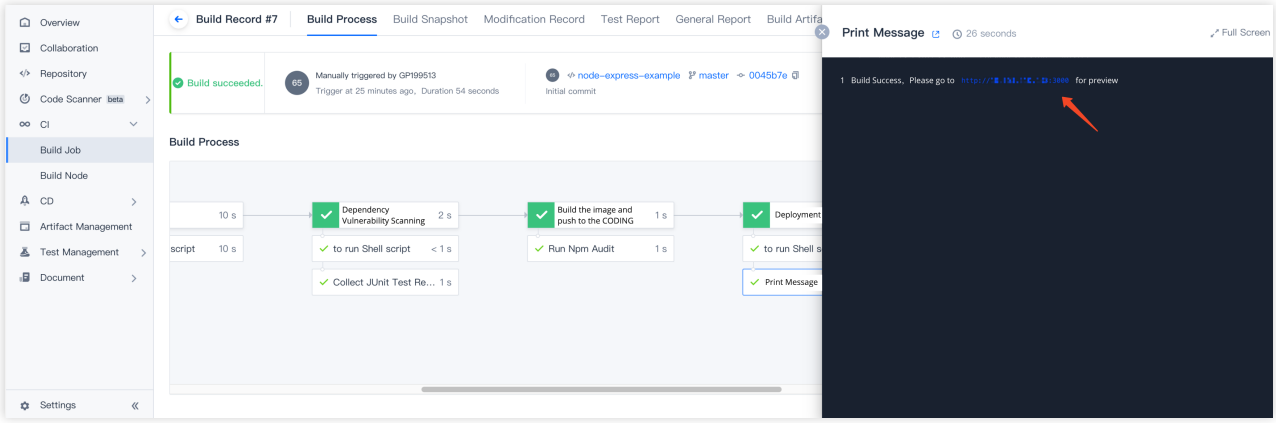
Click **OK** to save the build plan. If you have selected **Trigger build after creation** the build plan will be executed immediately. During build plan execution, you can view build details in the list of build plan records.



Click **Build Records** to view the operation statuses for each stage in the pipeline. You can also see the execution results and logs for the commands in each step.

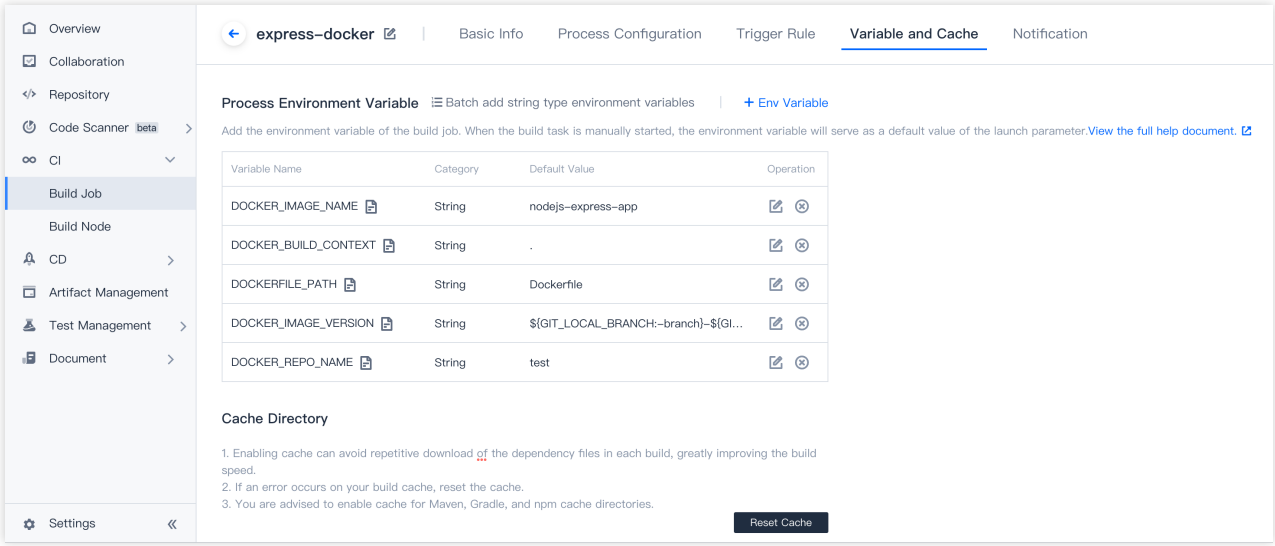


If the operation status of Step 5 is normal, you can view the artifact output URL in the build plan.



7. Modify remote server information

You have already configured a remote server address and SSH key in Step 5. To change this information, go to **CI Plans** > **Settings** > **Variables and Caches**.



More Information

You can use CI plans to set custom CI build nodes.

Basic information

Cloud server and custom build node

Configuration process

Use the graphical interface to configure build processes

Build different types of artifacts and deliver them to the artifact repository

Trigger rules

Configure the CI trigger method

Variables and caches

Invoke security credentials and configure them in environment variables

Configure the project cache