

|  |
| --- |
|  |

**CI Guidelines and Best Practices**

# Table of Contents

1. INTRODUCTION…………….……………………………………………….…………………………………………………………………………1

2. BEST PRACTICES OF Continous Integration…………….………………………………………………………………………..2

# 

# Introduction

Continuous Integration (CI) is the process of building software with every change committed to a project's version control repository. CI implemented projects aims to reduce rework and thus reduce cost and time.

Block Diagram

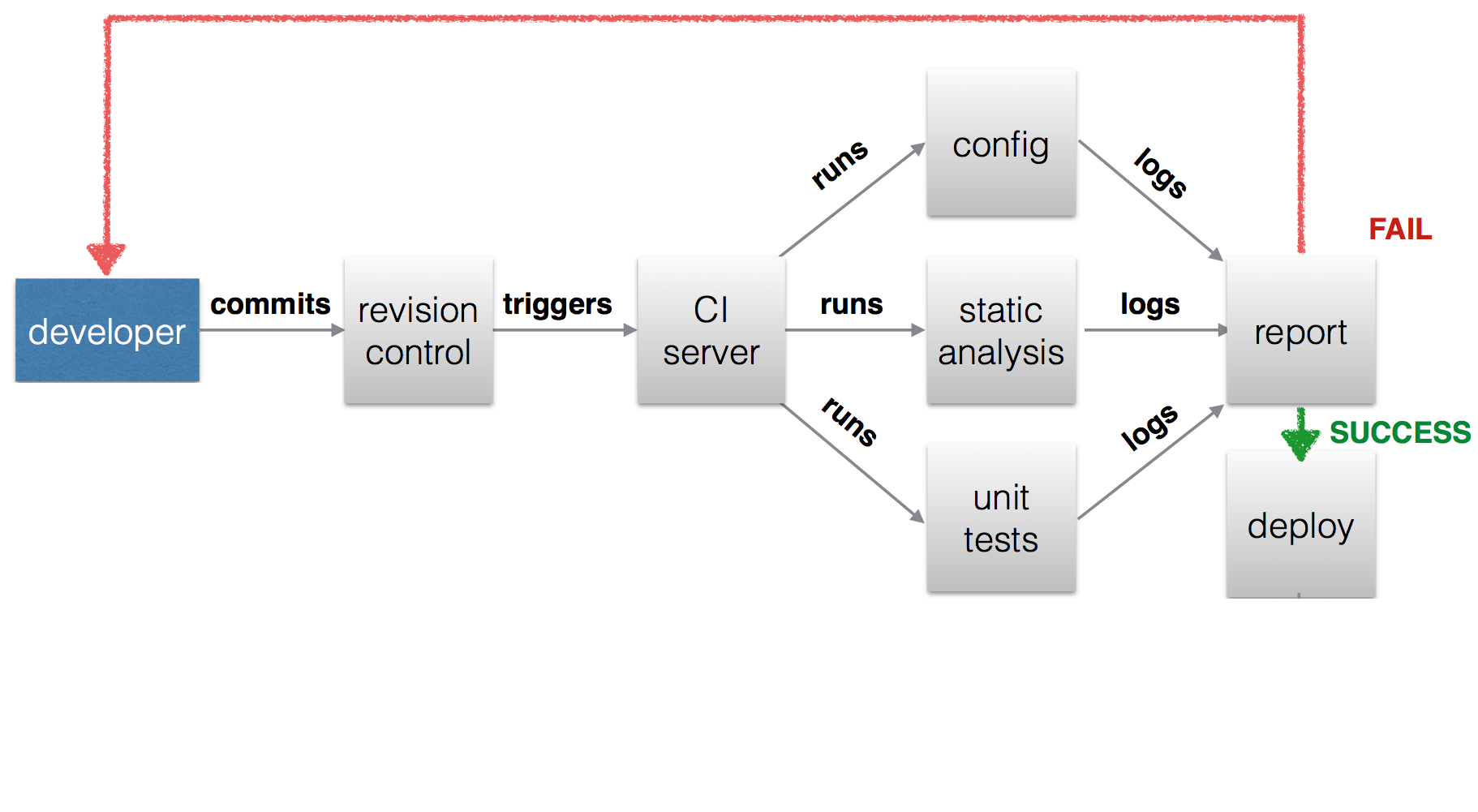


Fig: CI generic block diagram

# BEST PRACTICES OF CI

Following are the guidelines for developers:

1. **Maintain a code repository :**

There should be single repo for entire project so that it will be easy for source code management ,

All developer’s should clone that repo into their local systems and should create feature branch and start coding on it and once it is reviewed merge it into master branch**.**

1. **Everyone commits to the baseline every day:**

Every developer should commit his whole day work at the end of the day into repository.

1. **Automate the build :**

It is best practice to use build tool to automate build process.  Build tool automates the creation of executable applications from source code. As per programming language , there are multiple build tools available for automation. Such as maven , ant, gradle, grunt, gulp etc.

1. **Every commit should be build**

For build automation it is advisable to use tools like Jenkins or Bamboo . These tools are used for integration and to automate the build process using build tools. For every code commit on repo , Jenkins build should be triggered.

1. **Use of static code analysis in build process**

It is always advisable to follow certain sets of coding standards , which results in less code duplication ,less coding standard violation, less coding complexity .It comes under code review. Previously it was done manually but in automated CI process (using Jenkins/bamboo), there are multiple tools by means of which we can carry out static code analysis in automated CI process (using Jenkins/bamboo) by means of which we can display it’s reports on Jenkins/bamboo dashboard.

It is advisable to use static code analysis tools such as sonarqube , Istanbul etc.

1. **Every change should be tested :**

Every changes in codebase should be tested .It is advisable to use different frameworks for testing.

For eg. Phpunit , Junit , Selenium framework etc.

1. **Use of code coverage tools in build process.**

Your test cases should be such that it should cover entire code .In current CI build process these things should be automated .This can be done by using code coverage tools like emma , clover , cobertura , etc. It is possible to render reports generated by code coverage tools on Jenkins/bamboo for more readability and understanding in case of build failures.

1. **Generate various reports through CI servers to have good visibility and readability.**

Reports generated by using build tools are in csv or xml format , which are not readable . It is advisable to use CI servers to display (render ) reports on CI server’s dashboard in tabular or statistical format.

1. **Securing CI servers**

There should be levels of authentications provided in CI servers for group of users. For some CI servers this facility is in build but for some CI servers we need to explicitly provide it by adding some plugins related to it.

1. **Integrating CI server with defect management tool and test management tool.**

There should be traceability between Defect management tool (JIRA ) and Test management tools (Testlink), so that you can easily navigate from one tool to another tool if necessary.

1. **Integrating User Authentication Management system. (SSO)**

In CI process, we use many applications and For every application authentication is e=required. so it is better to use some Authentication Management system which provides facility of single sign on.

1. **Tagging sprint/releases/important builds through CI servers.**

There should be some provision for tagging of sprints, releases, builds in CI build.

1. **End to end build process:**

Every developer should have knowledge of entire build process dedicated for entire sprint**.**

1. **Violation limits for stable build**

There are certain violations like checkstyle , pmd etc . We can set certain values for these violations.

If your build is not fulfilling limits which were set in jenkins/bamboo , then it should result in build failure.

1. **Create stable and fast builds**

The build should be stable .there should not be any long running builds or any build queues. These things affects the overall performance of the build

1. **Easy access of artifacts to all developers**

Any successful build will result in creation of artifacts , so keep track of these artifacts it is advisable to integrate Jenkins with artifactory. All developers should have access to view the build results of Jenkins/bamboo.

1. **Significant of Environment variable.**

It is best practice to make use of environment variables provided by CI server’s for decision making purpose.