

- 1. Introduction (<https://github.com/amosproj/amos2023ws02-pitest-ide-plugin/wiki/Build--deploy-documentation#1-introduction>)
  - 1.1. Purpose (<https://github.com/amosproj/amos2023ws02-pitest-ide-plugin/wiki/Build--deploy-documentation#11-purpose>)
  - 1.2. Scope (<https://github.com/amosproj/amos2023ws02-pitest-ide-plugin/wiki/Build--deploy-documentation#12-scope>)
- 2. Prerequisites (<https://github.com/amosproj/amos2023ws02-pitest-ide-plugin/wiki/Build--deploy-documentation#2-prerequisites>)
  - 2.1. Software Dependencies (<https://github.com/amosproj/amos2023ws02-pitest-ide-plugin/wiki/Build--deploy-documentation#21-software-dependencies>)
  - 2.2. GitHub Repository (<https://github.com/amosproj/amos2023ws02-pitest-ide-plugin/wiki/Build--deploy-documentation#22-github-repository>)
- 3. Build Process (<https://github.com/amosproj/amos2023ws02-pitest-ide-plugin/wiki/Build--deploy-documentation#3-build-process>)
  - 3.1. Clone Repository (<https://github.com/amosproj/amos2023ws02-pitest-ide-plugin/wiki/Build--deploy-documentation#31-clone-repository>)
  - 3.2. Gradle Build (<https://github.com/amosproj/amos2023ws02-pitest-ide-plugin/wiki/Build--deploy-documentation#32-gradle-build>)
  - 3.3. Run Tests (<https://github.com/amosproj/amos2023ws02-pitest-ide-plugin/wiki/Build--deploy-documentation#33-run-tests>)
  - 3.4. Create JAR File (<https://github.com/amosproj/amos2023ws02-pitest-ide-plugin/wiki/Build--deploy-documentation#34-create-jar-file>)
- 4. Deployment to GitHub (<https://github.com/amosproj/amos2023ws02-pitest-ide-plugin/wiki/Build--deploy-documentation#4-deployment-to-github>)
  - 4.1. GitHub Actions (<https://github.com/amosproj/amos2023ws02-pitest-ide-plugin/wiki/Build--deploy-documentation#41-github-actions>)
  - 4.2. Workflow Configuration (<https://github.com/amosproj/amos2023ws02-pitest-ide-plugin/wiki/Build--deploy-documentation#42-workflow-configuration>)
  - 4.3. Deployment Triggers (<https://github.com/amosproj/amos2023ws02-pitest-ide-plugin/wiki/Build--deploy-documentation#43-deployment-triggers>)
- 5. Troubleshooting (<https://github.com/amosproj/amos2023ws02-pitest-ide-plugin/wiki/Build--deploy-documentation#5-troubleshooting>)
  - 5.1. Build Issues (<https://github.com/amosproj/amos2023ws02-pitest-ide-plugin/wiki/Build--deploy-documentation#51-build-issues>)
  - 5.2. Deployment Issues (<https://github.com/amosproj/amos2023ws02-pitest-ide-plugin/wiki/Build--deploy-documentation#52-deployment-issues>)

# 1. Introduction

## 1.1 Purpose

This document provides step-by-step instructions for building and deploying a Java-based project using Gradle as a build tool and GitHub as the deployment system.

## 1.2 Scope

The scope of this document covers the entire process from cloning the repository to deploying the project to GitHub.

# 2. Prerequisites

## 2.1 Software Dependencies

Ensure the following software is installed on your machine:

- [Java JDK \(https://www.oracle.com/java/technologies/javase-downloads.html\)](https://www.oracle.com/java/technologies/javase-downloads.html)
- [Gradle \(https://gradle.org/install/\)](https://gradle.org/install/)
- [Git \(https://git-scm.com/book/en/v2/Getting-Started-Installing-Git\)](https://git-scm.com/book/en/v2/Getting-Started-Installing-Git)

## 2.2 GitHub Repository

Create a GitHub repository for your project and obtain the repository URL.

# 3. Build Process

## 3.1 Clone Repository

```
git clone <repository-url>  
cd <project-directory>
```

## 3.2 Gradle Build

```
gradle clean build
```

## 3.3 Run Tests

```
gradle test
```

## 3.4 Create JAR File

```
gradle jar
```

The JAR file will be located in build/libs/.

# 4. Deployment to GitHub

## 4.1 GitHub Actions

Set up GitHub Actions by creating a .github/workflows/build.yml file.

## 4.2 Workflow Configuration

Edit the workflow configuration file according to your project's needs. This example assumes a simple Java project, and additional steps may be required for more complex projects.

## 4.3 Deployment Triggers

GitHub Actions will trigger the workflow on every push to the main branch. Adjust triggers in the workflow file to suit your requirements.

# 5. Troubleshooting

## 5.1 Build Issues

If the build fails, check the error messages and resolve any missing dependencies or configuration issues.

## 5.2 Deployment Issues

If deployment to GitHub fails, review the GitHub Actions logs for error details. Ensure that the workflow configuration is accurate and that necessary permissions are granted.