

How to Automate the Deployment of a Web application

A Guide to build a CI/CD pipeline for automated deployment of web application using Git and Jenkins.

Tools/Technologies used –

- Docker (Containerization platform)
- GitHub (for Source Code)
- Gradle (Build Tool)
- Jenkins (Automation CI/CD)
- AWS (Infrastructure/ Cloud platform)

Created two instances - one for master one for stagging(demo)

Instances (3) Info							
<input type="text" value="Search"/>		Refresh Connect		Instance state ▼	Actions ▼	Launch instances	▼
<input type="checkbox"/>	Name ▼	Instance ID	Instance state ▼	Instance type ▼	Status check	Alarm status	A
<input type="checkbox"/>	demo	i-09079ef4945804f73	✔ Running 🔍 🔍	t2.micro	✔ 2/2 checks passed	No alarms +	u
<input type="checkbox"/>	stagging	i-01dfa8405db613ceb	✔ Running 🔍 🔍	t2.micro	✔ 2/2 checks passed	No alarms +	u
<input type="checkbox"/>	master	i-0de4110e850692bfe	✔ Running 🔍 🔍	t2.micro	✔ 2/2 checks passed	No alarms +	u

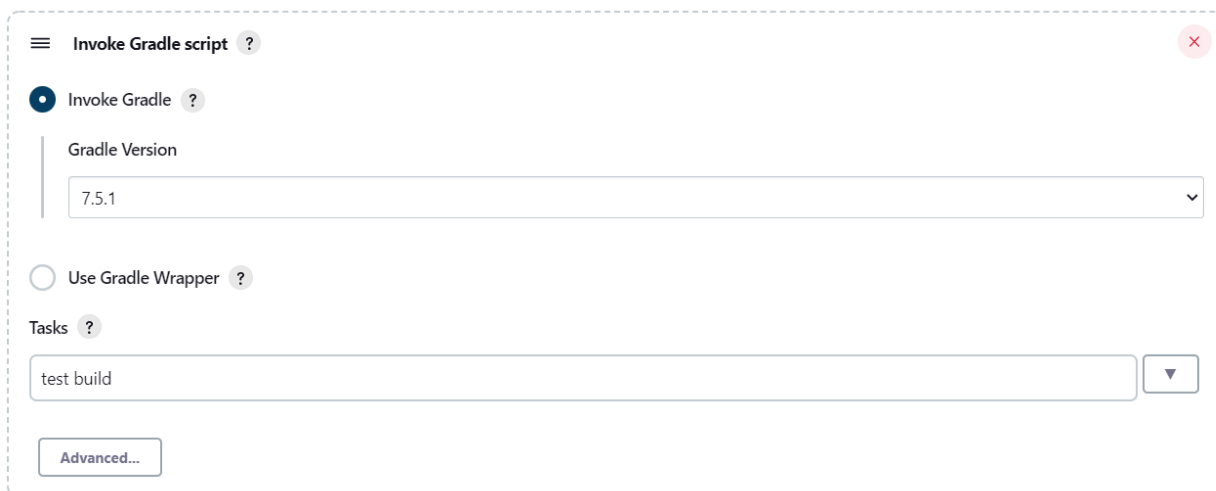
Source code from GitHub repository



The screenshot shows a GitHub repository page for the file `SpringBootTestApp-gradle- / src / main / java / com / kkjavatutorials / SpringBootTestApp / controller / MyController.java`. The repository is owned by `sajalrasto` and has a latest commit of `1d1b4`. The file is 14 lines long (10 sloc) and 347 Bytes. The code is as follows:

```
1 package com.kkjavatutorials.SpringBootTestApp.controller;
2
3 import org.springframework.web.bind.annotation.GetMapping;
4 import org.springframework.web.bind.annotation.RestController;
5
6 @RestController
7 public class MyController {
8
9     @GetMapping(path = "/welcome")
10    public String welcome() {
11        return "Happy to be Devops Engineer Sajal rastogi";
12    }
13
14 }
```

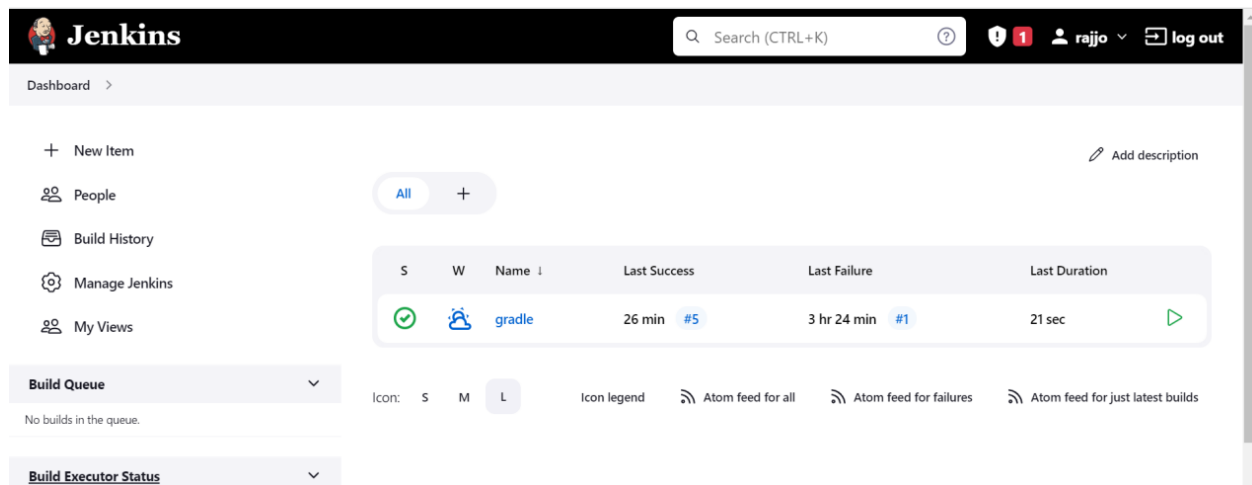
Test and Build using Gradle



The screenshot shows the 'Invoke Gradle script' dialog box. It has a title bar with a hamburger menu icon, the text 'Invoke Gradle script', and a question mark icon. There is a close button (red X) in the top right corner. The dialog contains the following options:

- ☒ Invoke Gradle ?
- Gradle Version: 7.5.1 (dropdown menu)
- ☐ Use Gradle Wrapper ?
- Tasks: test build (dropdown menu)
- Advanced... button

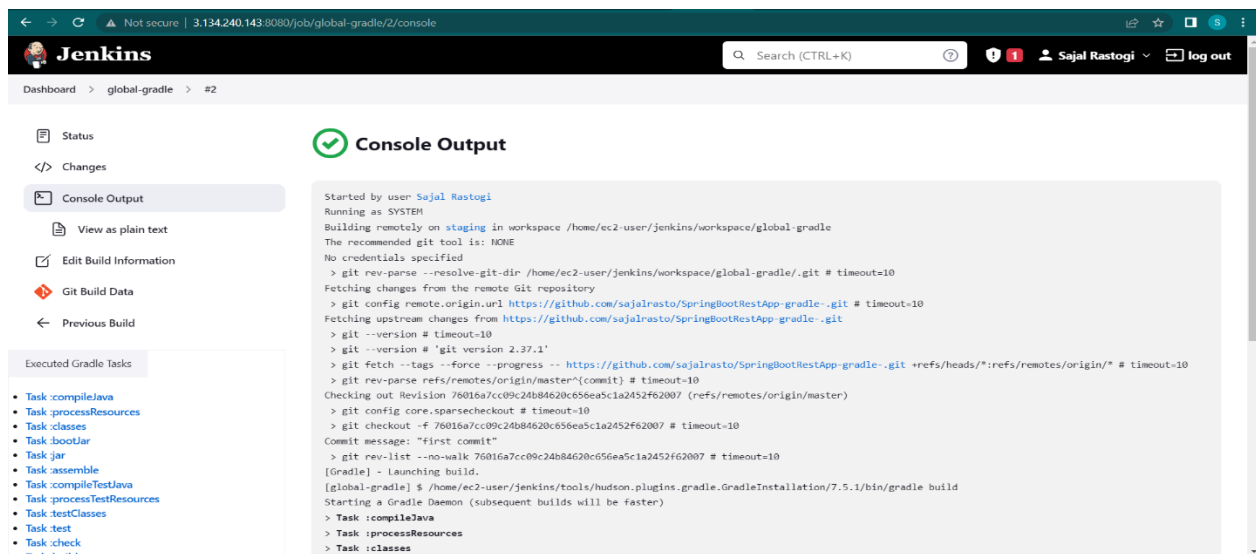
Created a Jenkins job



The screenshot shows the Jenkins Dashboard. At the top, there's a search bar and user information (rajjo). The left sidebar contains links: New Item, People, Build History, Manage Jenkins, and My Views. The main area displays a table of builds for the 'gradle' job. The table has columns for Status (S), Waiver (W), Name, Last Success, Last Failure, and Last Duration. The 'gradle' job shows a successful build with a duration of 21 seconds. Below the table, there's a 'Build Queue' section showing 'No builds in the queue.' and a 'Build Executor Status' section.

S	W	Name	Last Success	Last Failure	Last Duration
✓		gradle	26 min #5	3 hr 24 min #1	21 sec

Console Output



The screenshot shows the Jenkins Console Output for a job named 'global-gradle'. The output text is as follows:

```
Started by user Sajal Rastogi
Running as SYSTEM
Building remotely on staging in workspace /home/ec2-user/jenkins/workspace/global-gradle
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /home/ec2-user/jenkins/workspace/global-gradle/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/sajalrasto/SpringBootRestApp-gradle-.git # timeout=10
Fetching upstream changes from https://github.com/sajalrasto/SpringBootRestApp-gradle-.git
> git --version # timeout=10
> git --version # 'git version 2.37.1'
> git fetch --tags --force --progress -- https://github.com/sajalrasto/SpringBootRestApp-gradle-.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision 76016a7cc09c24b84620c656ea5c1a2452f62007 (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f 76016a7cc09c24b84620c656ea5c1a2452f62007 # timeout=10
Commit message: "first commit"
> git rev-list --no-walk 76016a7cc09c24b84620c656ea5c1a2452f62007 # timeout=10
[Gradle] - Launching build.
[global-gradle] $ /home/ec2-user/jenkins/tools/hudson.plugins.gradle.GradleInstallation/7.5.1/bin/gradle build
Starting a Gradle Daemon (subsequent builds will be faster)
> Task :compileJava
> Task :processResources
> Task :classes
```

After building Gradle project, Container is created automatically using Dockerfile.

```
root@staging/home/ec2-user/jenkins/workspace/global-gradle
[root@staging home]# ls
dockeradmin ec2-user
[root@staging home]# cd ec2-user/
[root@staging ec2-user]# ls
agent.jar jenkins
[root@staging ec2-user]# cd jenkins/
[root@staging jenkins]# ls
maven35-agent.jar maven35-interceptor-commons.jar remoting tools workspace
[root@staging jenkins]# cd workspace/
[root@staging workspace]# ls
global-gradle maven-tomcat newmaven newmaven2 newmaven3
[root@staging workspace]# cd global-gradle/
[root@staging global-gradle]# ls
bin build build.gradle gradle gradlew gradlew.bat README.md settings.gradle src
[root@staging global-gradle]# cd build
[root@staging build]# ls
classes generated libs reports resources test-results tmp
[root@staging build]# cd libs
[root@staging libs]# ls
SpringBootTestApp-0.0.1-SNAPSHOT.jar
[root@staging libs]# pwd
/home/ec2-user/jenkins/workspace/global-gradle/build/libs
[root@staging libs]# ^C
[root@staging libs]# cd ..
[root@staging build]# cd ..
[root@staging global-gradle]# ls
bin build build.gradle Dockerfile gradle gradlew gradlew.bat README.md settings.gradle src
[root@staging global-gradle]# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS
NAMES
212f8177af05   gradle_demo   "catalina.sh run"       2 minutes ago Up 2 minutes   0.0.0.0:8093->8080/tcp, :::8093->8080/tcp
sajal-gradle
```

Content inside the Dockerfile

```
root@demo:/home/ec2-user/jenkins/workspace/gradle/build/libs
FROM openjdk:11
COPY SpringBootTestApp-0.0.1-SNAPSHOT.jar /
WORKDIR /
CMD ["java", "-jar", "SpringBootTestApp-0.0.1-SNAPSHOT.jar"]
~
~
~
~
~
~
~
```

**Final output Displayed on 8093 port no. with get mapping endpoint :
/welcome**



A screenshot of a web browser window. The address bar shows the URL `3.142.49.80:8093/welcome`. Below the address bar, the page content displays the text "Happy to be Devops Engineer Sajal rastogi".