**Continuous Integration (CI) & Continuous deployment Best Practices with SAP: Java Web on SAP Cloud Platform using a Cloud-based Build Service**

This scenario focuses on using publicly available services to build Java applications quickly and efficiently using the cloud services GitHub and Travis CI. There is no need to set up any local infrastructure.

This pipeline adheres to basic CI and CD practices and can be easily extended. It ensures that each change to your project’s repository is built centrally, thus applying the “Build Every Change” CI practice which is crucial for collaborative development.

GitHub’s public Source Code Management (SCM) service, and Travis CI as the build system, provide an enormous ecosystem. Travis CI is a distributed continuous integration service for building and testing software projects, which are hosted at GitHub. Both GitHub and Travis CI offer various plans to suit your individual requirements, ranging from a free public offering to an on-premise installation of the full service.

**Prerequisites**

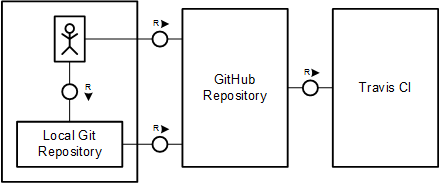
* An account on GitHub and Travis CI. If you already have an account, you can reuse it.

GitHub and Travis CI are both also available in enterprise on-premise versions. You can also set up mixed scenario, using one part in the cloud and the other on-premise.

[GitHub](https://github.com/)  
[Travis CI](https://travis-ci.org/) – Login using the option “Sign in with Github”

Basic setup

The example shows how to create a GitHub repository containing a small sample application, and configure Travis CI to react on a commit event in GitHub by triggering a build.



Procedure:

1. Enter the GitHub site and login to your account. [GitHub](https://github.com/)
2. The next step is creating a sample application.

* Set the HTTP proxy:

set HTTP\_PROXY\_HOST=proxy

set HTTP\_PROXY\_PORT=8080

set HTTPS\_PROXY\_HOST=proxy

set HTTPS\_PROXY\_PORT=8080

* Fork and clone the repository:

Fork from <https://github.com/antopraveen/proto-java_hcp_bluegreen> and clone it **using git clone**

1. Open Pom.xml in the cloned project and change the line number 670 with your account and password (example as below)

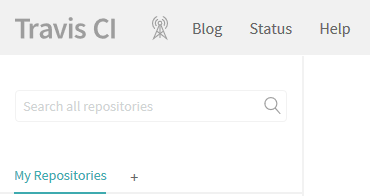
<configuration>

<consoleCommand>rolling-update -a i312624trial -b helloworld -h hanatrial.ondemand.com -u i312624 -p XXXXXXX -source ${project.build.directory}/${project.artifactId}.war</consoleCommand>

<configuration>

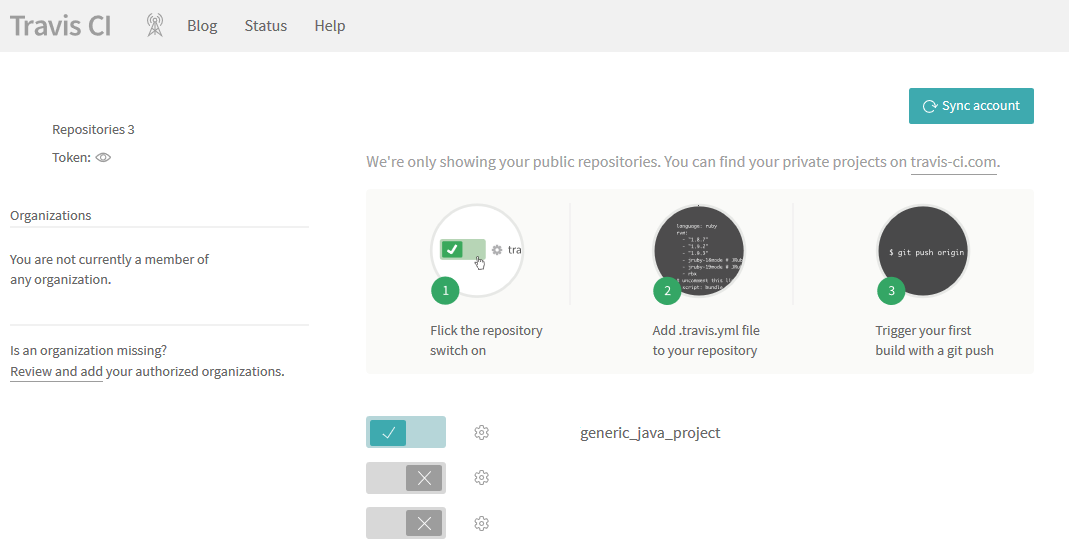
1. Open the URL of Travis CI in your browser and select Sign in with GitHub. You log in to Travis CI with your GitHub account.

[Travis CI](https://travis-ci.org/)

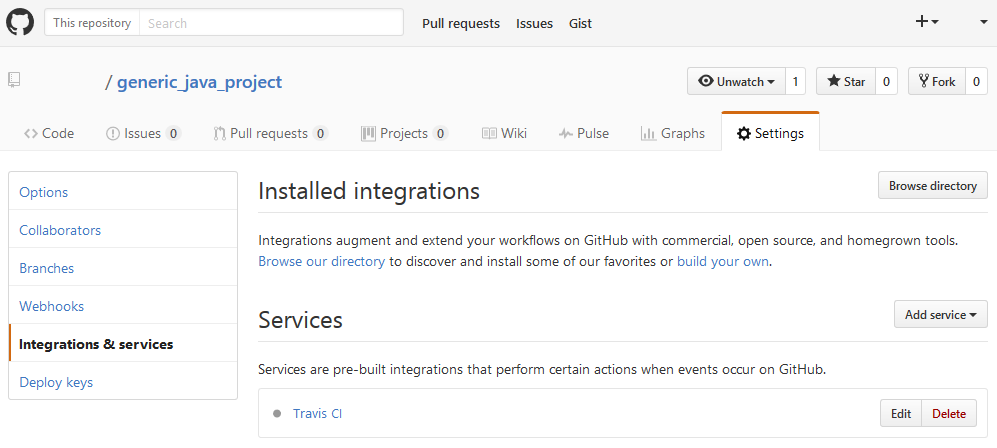
1. In the Travis CI front end, select **My Repositories** and select **+**. 

You see a list of your GitHub repositories.

## Add your repository (in this example, [proto-java\_hcp\_bluegreen](https://travis-ci.org/antopraveen/proto-java_hcp_bluegreen) to Travis CI by turning on the switch.



A web hook to the settings of your GitHub repository is automatically added and is called when you push new changes. You can verify your settings in GitHub by selecting **Settings > Integrations & services** from your project:



1. Open Travis CI and go to your project. Select **More options > Settings** and add the values below. You must switch off **Display value in build log** to avoid making your settings public.
2. Deploy hello-world in trial Landscape

<https://account.hanatrial.ondemand.com/cockpit#/home/overview>



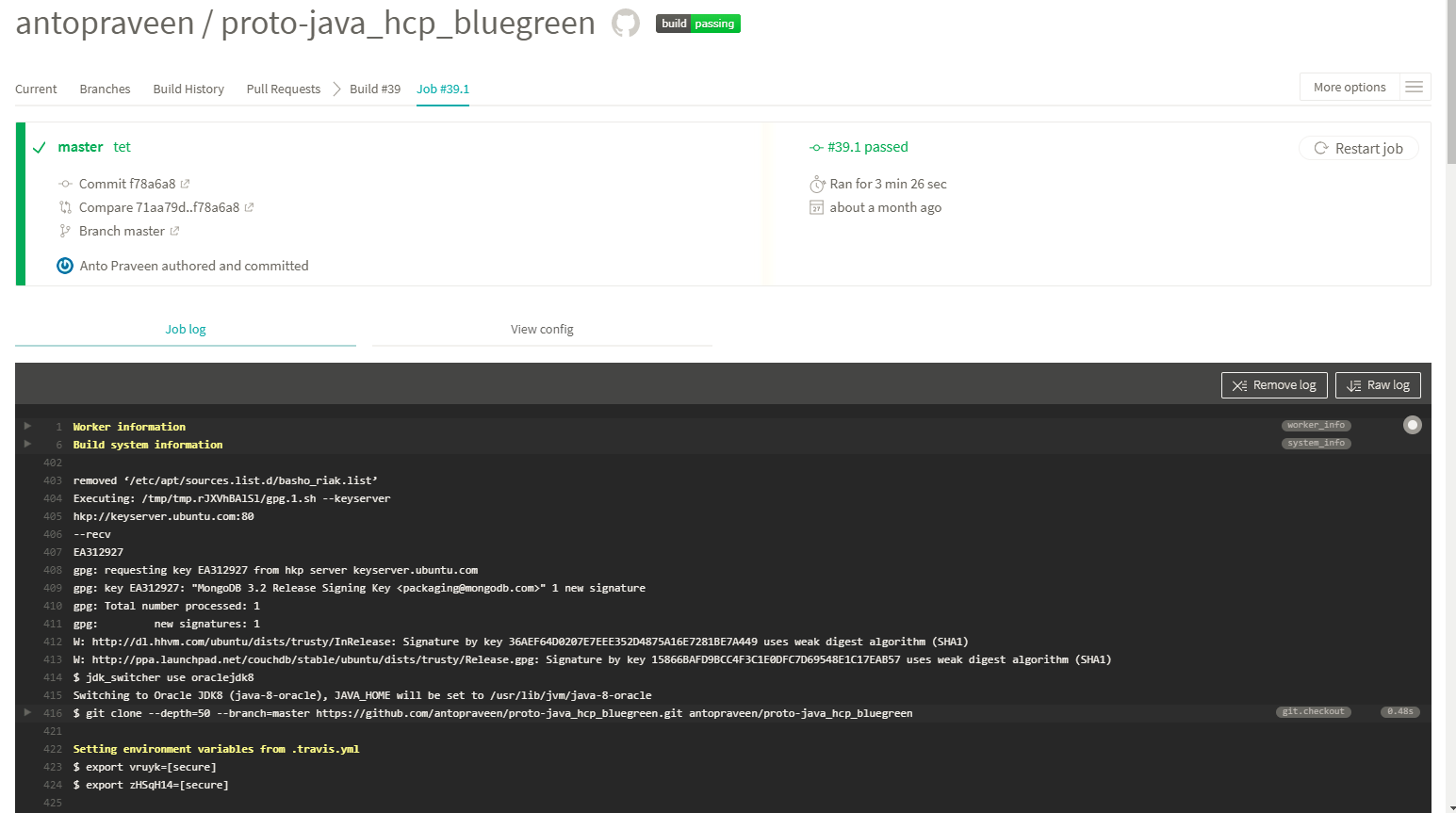
1. Add, commit, and push the sources. The initial project is now available in GitHub

git add .

git commit -m "Initial version of HelloWorld"

git push origin master

1. Check the status of build in Travis



1. Check the application in

<https://account.hanatrial.ondemand.com/cockpit>