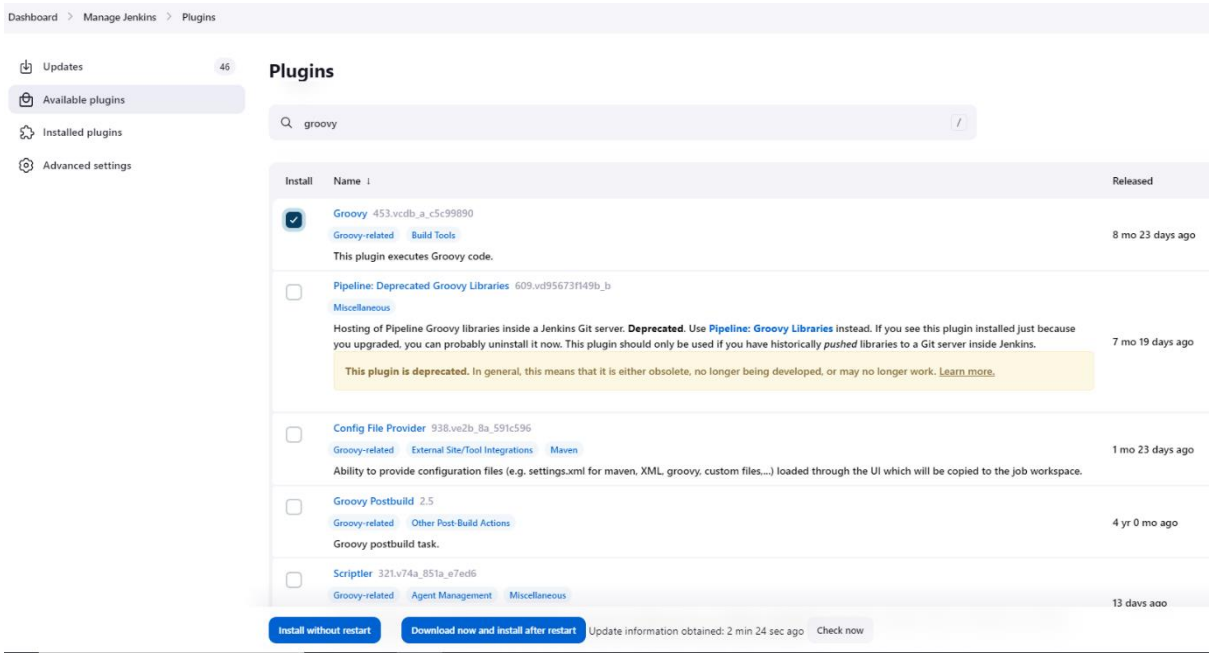
BuildPulse Jenkins Plugin Development

Pre-requisite: (Groovy DSL script to execute run.sh file)

1. Linux machine with Jenkins automation server, with internet connection
2. Install groovy dsl plugin in Jenkins.
3. Defining environment variables configuration change.
4. Defining new job in Jenkins
5. Defining optional build parameters
6. Groovy dsl script definition to execute run.sh file.
7. Github Credentials creation using personal access token.
8. Configure Source Code Management in Jenkins Job
9. Configuring specific directory to clone the GitHub code.
10. Build to view report submission status.

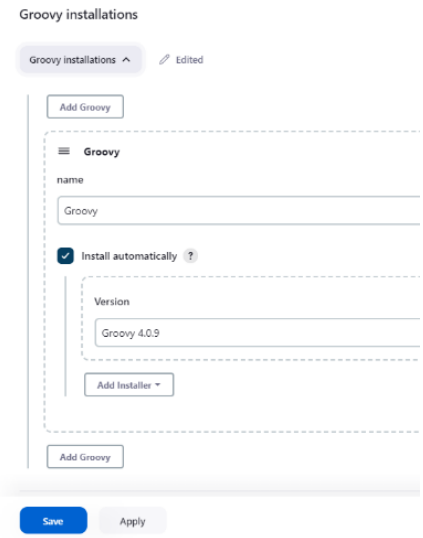
Install groovy dsl plugin in Jenkins.

Manage Jenkins -> Plugins -> search the name “groovy” and select the checkbox “Groovy” and then click button **“Install without restart”**



Jenkins -> Manage Jenkins -> Tools -> in **Groovy** title select “Groovy Installations” option.

Note: if the Groovy title is not available restart the jenkins instance.
“Groovy Installations” click the button “Add Groovy” it will display a container to define available groovy version select the version and click **save** button.



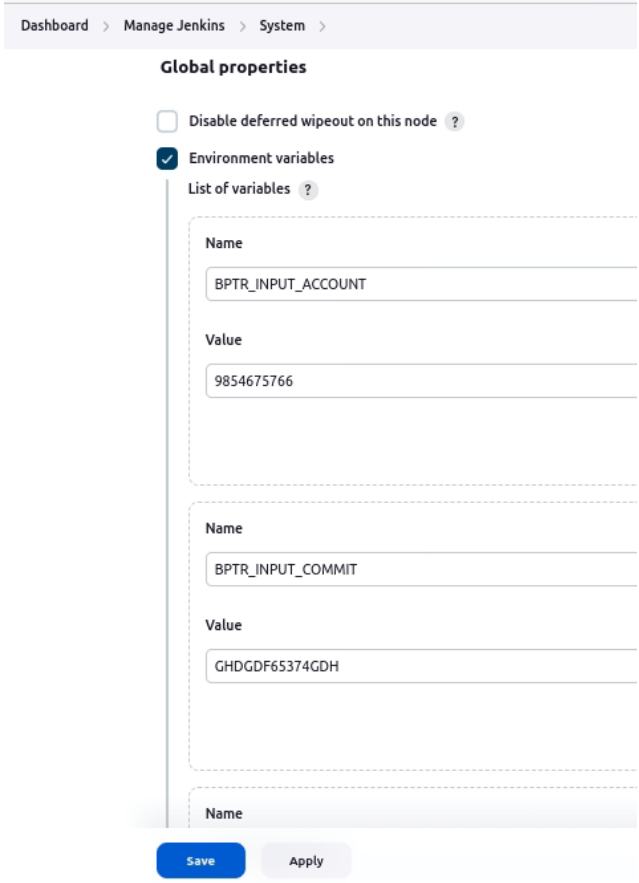
Defining environment variables

Environment variables are common for all Jenkins jobs, to differentiate it we used the prefix BPTR_ we need to define following data in Jenkins environment variables.

S.no	In run.sh used variables	Allocated new name with prefix, same is used in run.sh file.	Value
1.	INPUT_ACCOUNT	BPTR_INPUT_ACCOUNT	
2.	INPUT_COMMIT	BPTR_INPUT_COMMIT	
3.	INPUT_KEY	BPTR_INPUT_KEY	
4.	INPUT_PATH	BPTR_INPUT_PATH	
5.	INPUT_REPOSITORY	BPTR_INPUT_REPOSITORY	
6.	INPUT_REPOSITORY_PATH	BPTR_INPUT_REPOSITORY_PATH	
7.	INPUT_SECRET	BPTR_INPUT_SECRET	
8.	RUNNER_OS	BPTR_RUNNER_OS	**Linux**
9.	BUILDPULSE_BUCKET	BPTR_BUILDPULSE_BUCKET	**buildpulse-uploads-staging**

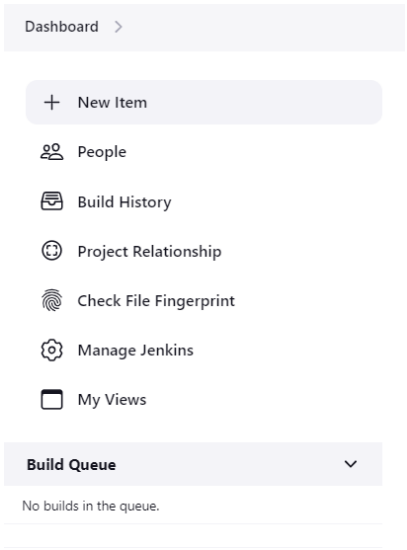
Jenkins Dashboard -> Manage Jenkins -> System -> search “Global Properties” -> select the checkbox of “Environment variables.”

Click **the Add** button for more variable definition and finally click **Save** button.

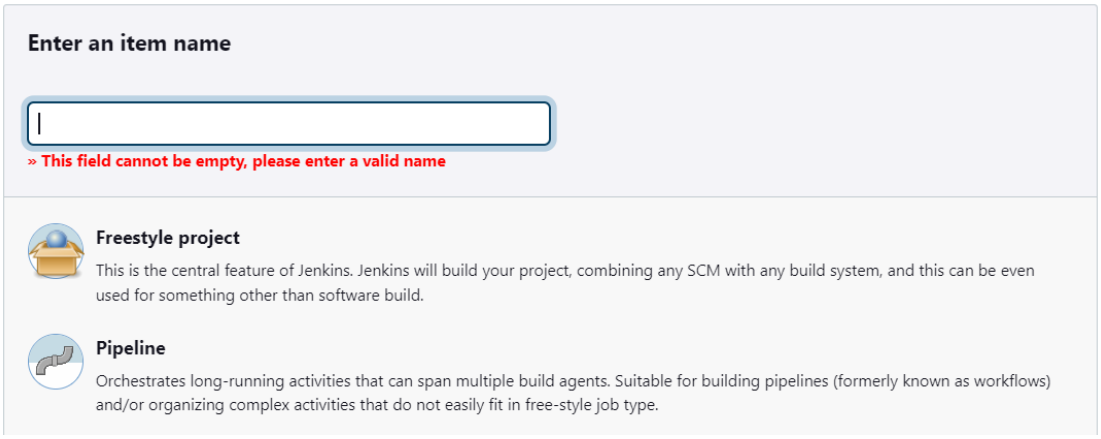


Defining new job in Jenkins

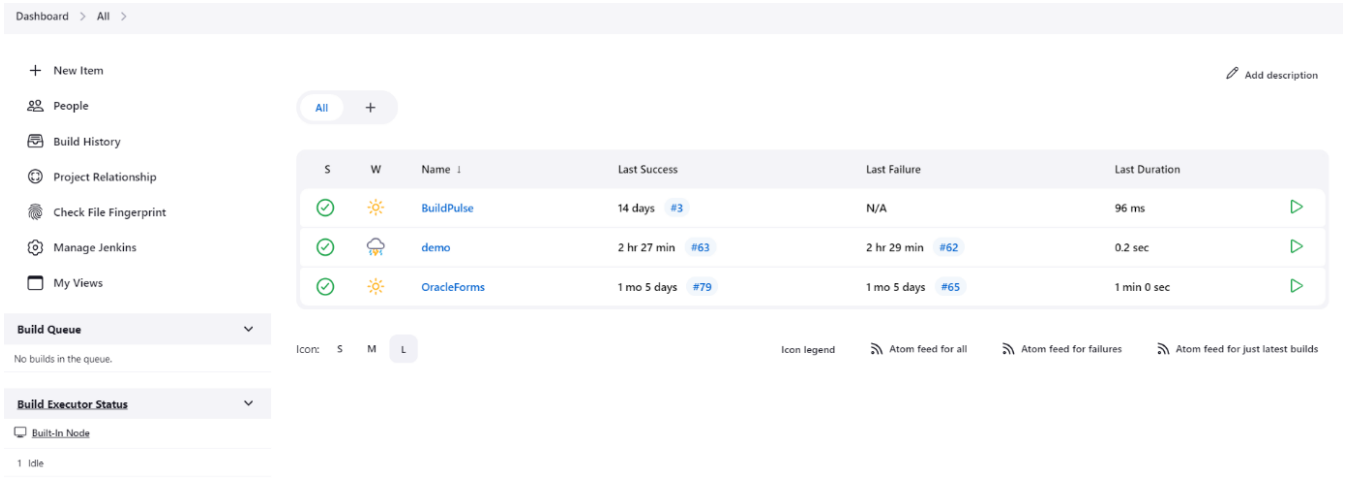
In Jenkins, Dashboard -> New Item -> define your job name.



After entering the new **job name**, select **Freestyle project**, and click **the OK** button.



Available jobs listed in Dashboard as follows.

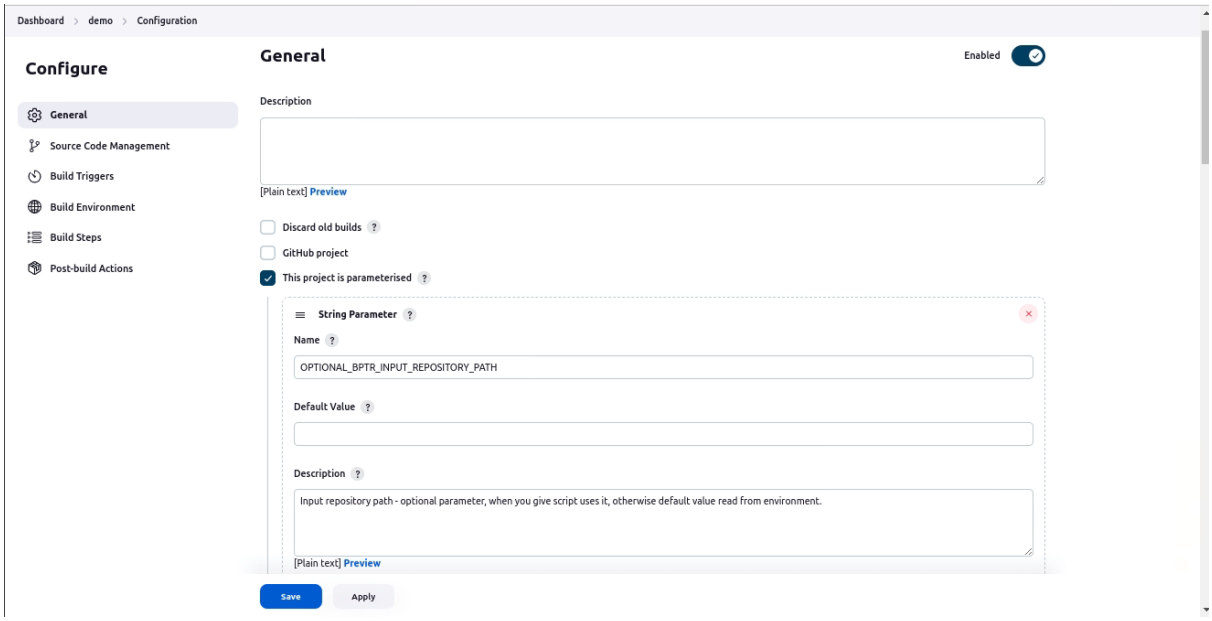


Defining Optional Build Parameters

Open your job -> configuration-> General -> select the check box **“This project is parameterized”** -> click the button **“Add Parameter”** and select the parameter type **“String parameter”** new container display in that define your job parameter variables with description, finally click **save** button.

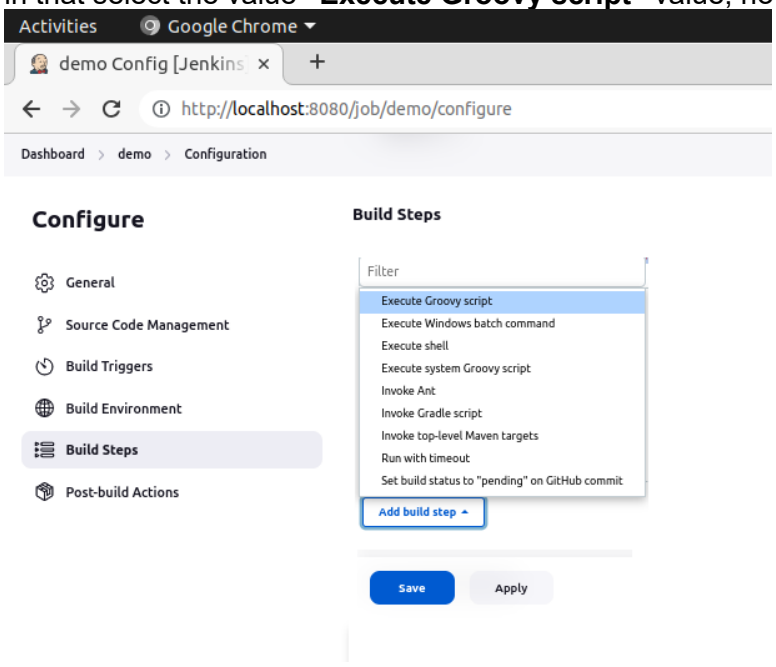
Following 2 variables we need to define.

S.no	Allocated new name with prefix, same is used in run.sh file.	Value
1.	OPTIONAL_BPTR_INPUT_COMMIT	Each build time user may enter data, if not in the environment variable defined value to be used by run.sh file.
2.	OPTIONAL_BPTR_INPUT_REPOSITORY_PATH	



Groovy dsl script definition to execute run.sh file.

Note: BUILDPULSE_BUCKET key we need to use in run.sh file. Modified run.sh file available at end of the document.
Open your job -> configuration-> in **Build Steps** title below dropdown box **“Add build step”** is available in that select the value **“Execute Groovy script”** value, now new container will be added.



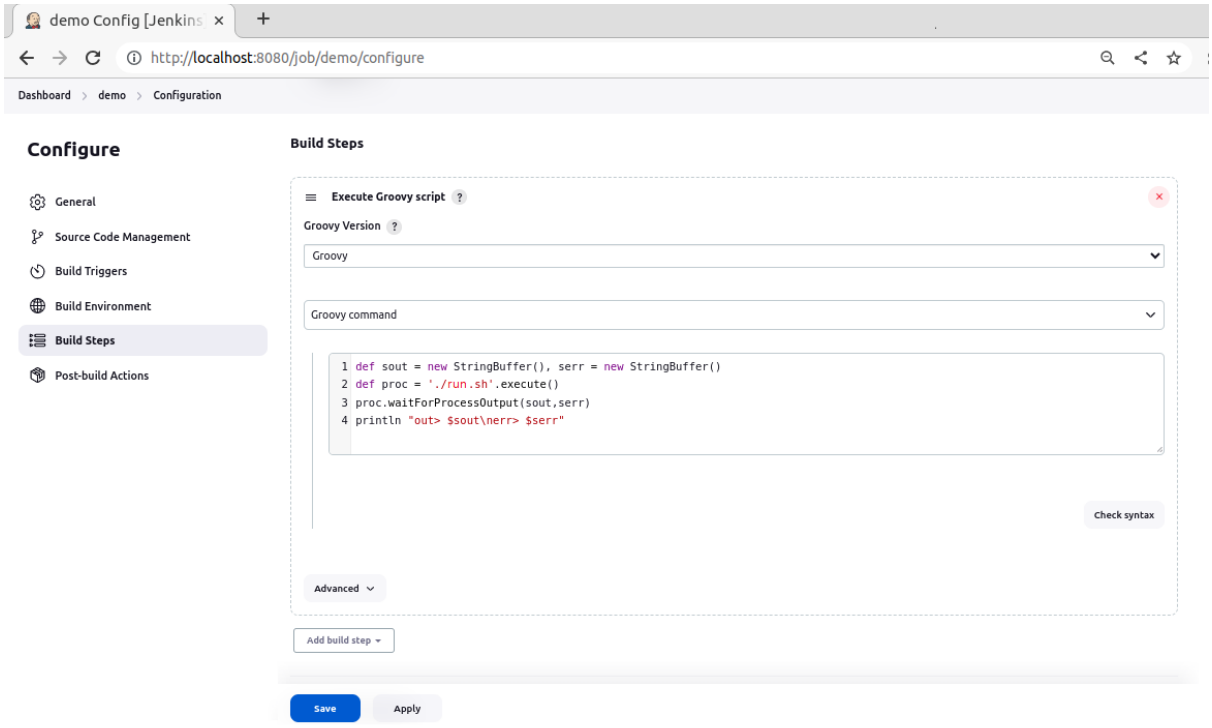
In the new container select the **“Groovy version”** as **“Groovy”**, and in the next drop down select the value **“Groovy Command”.**

Next command place holder places the groovy dsl script.

```
def sout = new StringBuffer(), serr = new StringBuffer()
def proc = './run.sh'.execute()
proc.waitForProcessOutput(sout,serr)
println "out> $sout\nerr> $serr"
```

Notes: place your **run.sh** file inside the workspace of job.
Example **/var/lib/jenkins/workspace/demo/**

finally click save button.



Github Credentials creation using personal access token:

Go to Dashboard -> Manage jenkins -> Configure System you will find the GitHub Servers. In that we need to provide necessary details.

Name

Reference name for our GitHub Credential

API URL

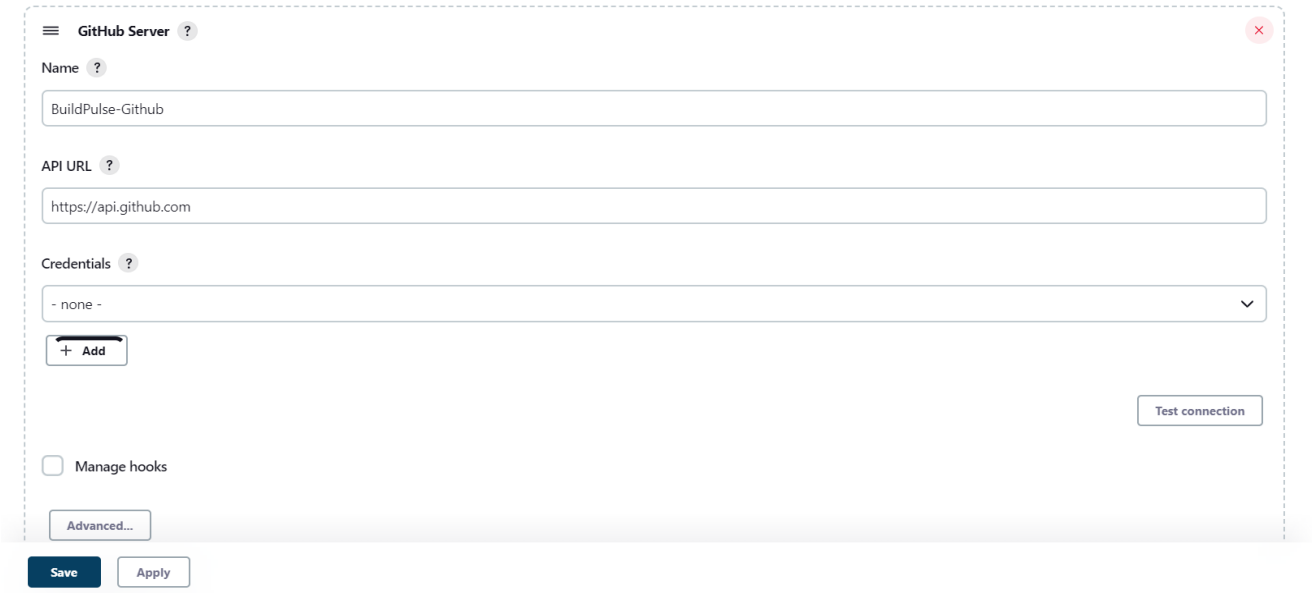
API endpoint of a GitHub server. To use public github.com, leave this field to the default value of https://api.github.com. Otherwise, if you use GitHub Enterprise, specify its API endpoint here (e.g., https://ghe.acme.com/api/v3/)

Credentials

You can create your own **personal access token** in your account GitHub settings. Token should be registered with scopes:

1. **admin:repo_hook** - for managing hooks (read, write, and delete old ones)
2. **repo** - to see private repos.
3. **repo: status** - to manipulate commit statuses.

In Jenkins, create credentials as **«Secret Text»**, provided by Plain Credentials Plugin.

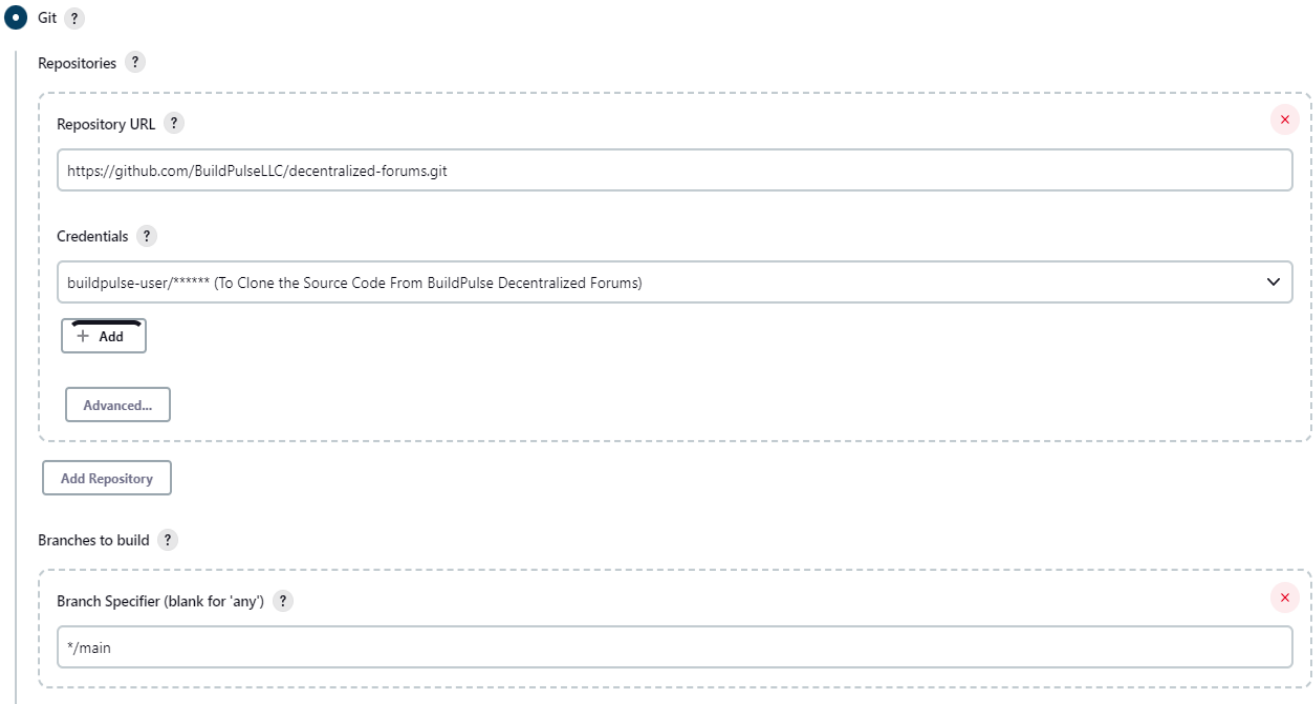


Once you created the credentials, then click **Save**.

Configure Source Code Management in Jenkins Job

Go to Dashboard -> <Your_Job> -> Configurations -> Source Code Management.

Select **Git** option and click **Add Repositories**. In that provide the **Decentralized-Forums GitHub Path** and select the credentials you previously created in the GitHub Server section. Afterwards you need to specify the respective branch name in **"Branch Specifier"** section.



Configuring specific directory to clone the GitHub code

If you want to configure the Specific Directory for the GitHub Clone, you need to click the **"Additional Behaviours"** option. In those options, select **Checkout to Sub-Directory** to provide the specific path for the GitHub Code

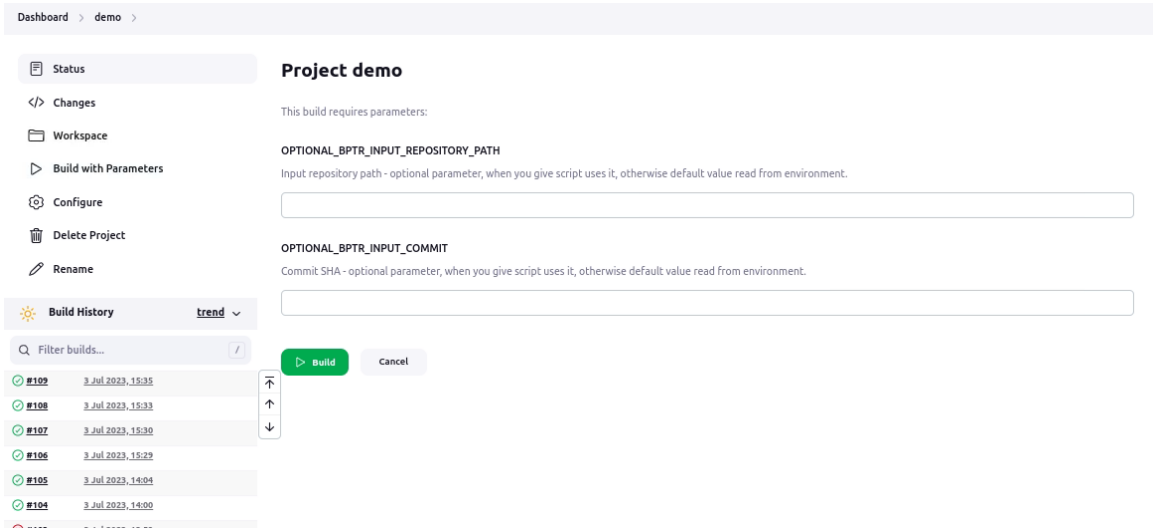
Definition of Code Build Command in Jenkins

For this POC Activity, we build the **decentralized-forums code manually in the server** and used the system generated report XML path in the Jenkins environment.

To make it automation, define the code build commands in the jenkins job build steps.

Build to view report submission status.

In your **job dashboard** click the link **Build with Parameter** now to execute the build, it will run the groovy dsl script which will execute the run.sh file to submit the junit test result to build pulse via buildpulse-test-reporter tool.



Click the **last execution build number** and select the **console output** to view the report submission status.

Console Output

```
Started by user DCEP
Running as SYSTEM
Building on the built-in node in workspace /var/jenkins_home/workspace/BuildPulse
The recommended git tool is: NONE
using credential buildpulse-justin--github-personal-token
 > git rev-parse --resolve-git-dir /var/jenkins_home/workspace/BuildPulse/DecentralizedForumsCode/.git # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url https://github.com/BuildPulseLLC/decentralized-forums.git # timeout=10
Fetching upstream changes from https://github.com/BuildPulseLLC/decentralized-forums.git
 > git --version # timeout=10
 > git --version # 'git version 2.30.2'
using GIT_ASKPASS to set credentials To Clone the Source Code From BuildPulse Decentralized Forums
 > git fetch --tags --force --progress -- https://github.com/BuildPulseLLC/decentralized-forums.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision c7476079a82ac3b66d4e01fa352bbc45b4d40bb7 (refs/remotes/origin/main)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f c7476079a82ac3b66d4e01fa352bbc45b4d40bb7 # timeout=10
Commit message: "reset back to pipeline"
 > git rev-list --no-walk c7476079a82ac3b66d4e01fa352bbc45b4d40bb7 # timeout=10
[BuildPulse] $ /var/jenkins_home/tools/hudson.plugins.groovy.GroovyInstallation/Groovy/bin/groovy /var/jenkins_home/workspace/BuildPulse/hudson4576387500804592300.groovy
out> This bash shell version supports double globbing: '5.1.4(1)-release'.
1. account: 46661156
2. repository: 608133431
3. path: /var/jenkins_home/workspace/BuildPulse/DecentralizedForumsCode/spec/reports/
4. key: ***
5. secret: ***
6. repository path: /var/jenkins_home/workspace/BuildPulse/DecentralizedForumsCode/
7. commit sha: c7476079a82ac3b66d4e01fa352bbc45b4d40bb7
8. operating system: Linux
 report tool download success....
<buildpulse> Current version: BuildPulse Test Reporter 0.27.2 (linux 7e01ff7 go1.19.12)
<buildpulse> Initiating `submit`
<buildpulse> Received args: /var/jenkins_home/workspace/BuildPulse/DecentralizedForumsCode/spec/reports/ --account-id 46661156 --repository-id 608133431 --repository-dir
/var/jenkins_home/workspace/BuildPulse/DecentralizedForumsCode/
<buildpulse> Using working directory: /var/jenkins_home/workspace/BuildPulse
<buildpulse> Looking for git repository at /var/jenkins_home/workspace/BuildPulse/DecentralizedForumsCode/
<buildpulse> Found git repository at /var/jenkins_home/workspace/BuildPulse/DecentralizedForumsCode/
<buildpulse> Gathering metadata to describe the build
<buildpulse> Detected build environment: jenkins
<buildpulse> Using $GIT_COMMIT environment variable as commit SHA: c7476079a82ac3b66d4e01fa352bbc45b4d40bb7
<buildpulse> Looking up info for commit `c7476079a82ac3b66d4e01fa352bbc45b4d40bb7` in git repository
<buildpulse> Found commit info
<buildpulse> Writing metadata to /tmp/buildpulse-1465516518.yml
<buildpulse> Preparing tarball of test results:
<buildpulse> - /var/jenkins_home/workspace/BuildPulse/DecentralizedForumsCode/spec/reports/rspec.xml
<buildpulse> - /var/jenkins_home/workspace/BuildPulse/DecentralizedForumsCode/coverage/lcov/DecentralizedForumsCode.lcov
<buildpulse> Adding buildpulse.yml to tarball
<buildpulse> Flushing log to /tmp/buildpulse-1352517512.log
<buildpulse> Adding buildpulse.log to tarball
<buildpulse> Gzipping tarball (/tmp/buildpulse-3421686409.tar)
<buildpulse> Sending /tmp/buildpulse-230073231.gz to BuildPulse
<buildpulse> Delivered test results to BuildPulse (46661156/608133431/buildpulse-448581da-c3fc-4007-a2cc-aee9d5448a43.gz)

err> + curl -fsSL --retry 3 --retry-connrefused --connect-timeout 5 https://get.buildpulse.io/test-reporter-linux-amd64
+ BUILDPULSE_ACCESS_KEY_ID=AKIA3SYVRUSE3N4K4B6P
+ BUILDPULSE_SECRET_ACCESS_KEY=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
+ GITHUB_SHA=c7476079a82ac3b66d4e01fa352bbc45b4d40bb7
+ BUILDPULSE_BUCKET=buildpulse-uploads-staging
+ ./buildpulse-test-reporter submit /var/jenkins_home/workspace/BuildPulse/DecentralizedForumsCode/spec/reports/ --account-id 46661156 --repository-id 608133431 --repository-dir
/var/jenkins_home/workspace/BuildPulse/DecentralizedForumsCode/

Finished: SUCCESS
```

Modified run.sh File:

Updated run.sh script file is listed below, in the echo message available images are displayed as question mark in this document.

```
#!/bin/bash

# Enable double globbing
shopt -s globstar

set -e

# Enable double globbing if supported by the shell on the base github runner
if shopt -s globstar; then
        echo "This bash shell version supports double globbing: '${BASH_VERSION}'."

else
  echo "This bash shell version does not support double globbing: '${BASH_VERSION}'. Please upgrade to bash 4+."
fi

if ! echo $BPTR_INPUT_ACCOUNT | egrep -q '^[0-9]+$'
then
        echo "?? The given value is not a valid account ID: ${BPTR_INPUT_ACCOUNT}"
        echo "?? To resolve this issue, set the 'account' parameter to your numeric BuildPulse Account ID."
        exit 1
fi
ACCOUNT_ID=$BPTR_INPUT_ACCOUNT

echo "1. account: $ACCOUNT_ID"

if ! echo $BPTR_INPUT_REPOSITORY | egrep -q '^[0-9]+$'
then
        echo "?? The given value is not a valid repository ID: ${BPTR_INPUT_REPOSITORY}"
        echo "?? To resolve this issue, set the 'repository' parameter to your numeric BuildPulse Repository ID."
        exit 1
fi
REPOSITORY_ID=$BPTR_INPUT_REPOSITORY

echo "2. repository: $REPOSITORY_ID"

for path in $BPTR_INPUT_PATH; do
        if [ ! -e "$path" ]
        then
                echo "?? The given path does not exist: $path"
                echo "?? To resolve this issue, set the 'path' parameter to the location of your XML test report(s)."
                exit 1
        fi
done
REPORT_PATH="${BPTR_INPUT_PATH}"

echo "3. path: $REPORT_PATH"

if [ ! -d "$BPTR_INPUT_REPOSITORY_PATH" ]
then
        echo "?? The given path is not a directory: ${BPTR_INPUT_REPOSITORY_PATH}"
        echo "?? To resolve this issue, set the 'repository-path' parameter to the directory that contains the local git clone of your
repository."
        exit 1
fi

# >>>>>> additional input replace START <<<<<<
REPOSITORY_PATH="${OPTIONAL_BPTR_INPUT_REPOSITORY_PATH}"
# from the user received INPUT_REPOSITORY_PATH value empty scenario read INPUT_REPOSITORY_PATH value from environment
variable.
if [ -z "$OPTIONAL_BPTR_INPUT_REPOSITORY_PATH" ]
then
   REPOSITORY_PATH="${BPTR_INPUT_REPOSITORY_PATH}"
fi

COMMIT_SHA="${OPTIONAL_BPTR_INPUT_COMMIT}"
# from the user received INPUT_COMMIT value empty scenario read INPUT_COMMIT value from environment variable.
if [ -z "$OPTIONAL_BPTR_INPUT_COMMIT" ]
then
   COMMIT_SHA="${BPTR_INPUT_COMMIT:-$GITHUB_SHA}"
fi

# >>>>>> additional input replace END <<<<<<

#echo "4. key: $BPTR_INPUT_KEY"
#echo "5. secret: $BPTR_INPUT_SECRET"
```

```
echo "4. key: ***"
echo "5. secret: ***"
echo "6. repository path: $REPOSITORY_PATH"
echo "7. commit sha: $COMMIT_SHA"

if test -z "$BPTR_INPUT_KEY" && test -z "$BPTR_INPUT_SECRET" && test "$GITHUB_ACTOR" = "dependabot[bot]"
then
	echo "::warning ::No value available for the 'key' parameter or the 'secret' parameter. Skipping upload to BuildPulse."
	echo "?? ?? ?? As of March 1, 2021, Dependabot PRs cannot access secrets in GitHub Actions. See details on the GitHub blog at
https://bit.ly/3KAoIBf"
	echo "?? ?? ?? Secrets are necessary in order to authenticate with external services like BuildPulse."
	echo "?? ?? ?? Since secrets aren't available in this build, the build cannot authenticate with BuildPulse to upload test results."
	exit 0
fi

echo "8. operating system: $BPTR_RUNNER_OS"

case "$BPTR_RUNNER_OS" in
	Linux)
		BUILDPULSE_TEST_REPORTER_BINARY=test-reporter-linux-amd64
		;;
	macOS)
		BUILDPULSE_TEST_REPORTER_BINARY=test-reporter-darwin-amd64
		;;
	Windows)
		BUILDPULSE_TEST_REPORTER_BINARY=test-reporter-windows-amd64.exe
		;;
	*)
		echo "::error::Unrecognized operating system. Expected RUNNER_OS to be one of \"Linux\", \"macOS\", or \"Windows\",
but it was \"$RUNNER_OS\"."
		exit 1
esac

BUILDPULSE_TEST_REPORTER_HOSTS=(
	https://get.buildpulse.io
	https://github.com/buildpulse/test-reporter/releases/latest/download
)
[ -n "${INPUT_CLI_HOST}" ] && BUILDPULSE_TEST_REPORTER_HOSTS=("${INPUT_CLI_HOST}"
"${BUILDPULSE_TEST_REPORTER_HOSTS[@]}")

getcli() {
	local rval=-1
	for host in "${BUILDPULSE_TEST_REPORTER_HOSTS[@]}"; do
		url="${host}/${BUILDPULSE_TEST_REPORTER_BINARY}"
		if (set -x; curl -fsSL --retry 3 --retry-connrefused --connect-timeout 5 "$url" > "$1"); then
			return 0
		else
			rval=$?
		fi
	done;

	return $rval
}

if getcli ./buildpulse-test-reporter; then
	: # Successfully fetched binary. Great!
  echo " report tool download success...."
else
  echo " report tool download issue...."
	msg=$(cat <<-eos
		::warning::Unable to send test results to BuildPulse. See details below.

		Downloading the BuildPulse test-reporter failed with status $?.

		We never want BuildPulse to make your builds unstable. Since we're having
		trouble downloading the BuildPulse test-reporter, we're skipping the
		BuildPulse analysis for this build.

		If you continue seeing this problem, please get in touch at
		https://buildpulse.io/contact so we can look into this issue.
	eos
	)

	echo "${msg//$'\n'/%0A}" # Replace newlines with URL-encoded newlines for proper formatting in GitHub Actions annotations
(https://github.com/actions/toolkit/issues/193#issuecomment-605394935)
	exit 0
fi

chmod +x ./buildpulse-test-reporter

set -x
```

```
BUILDPULSE_ACCESS_KEY_ID="${BPTR_INPUT_KEY}" \
	BUILDPULSE_SECRET_ACCESS_KEY="${BPTR_INPUT_SECRET}" \
	GITHUB_SHA="${COMMIT_SHA}" \
	BUILDPULSE_BUCKET="${BPTR_BUILDPULSE_BUCKET}" \
	./buildpulse-test-reporter submit $REPORT_PATH --account-id $ACCOUNT_ID --repository-id $REPOSITORY_ID --repository-dir
"${REPOSITORY_PATH}"
```