

Pipeline (Script)

✓ Declarative pipeline

- Easier to read , validate , & maintain
- uses modern approach
- DSL (Domain specific language) based on Groovy
- 95+ projects are using declarative pipeline.

Scripted pipeline

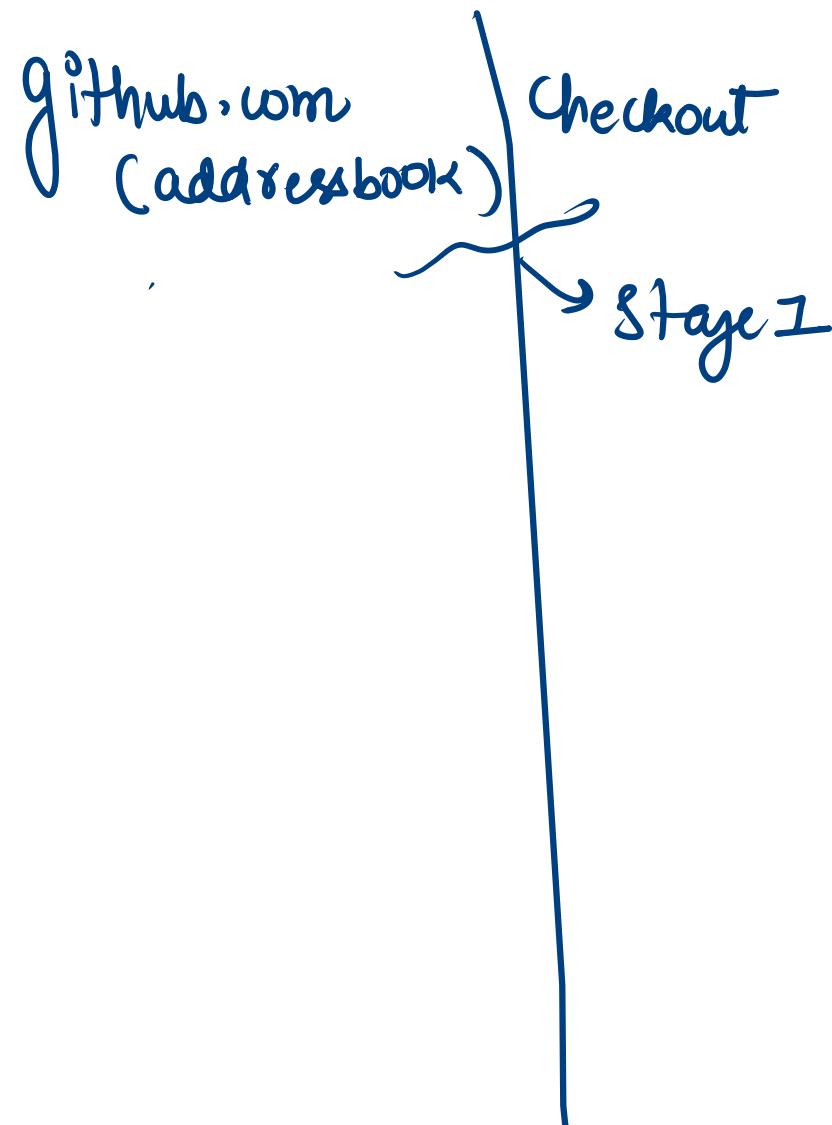
- More difficult to write, read & validate
- Java knowledge is required
 - Purely Groovy script
 - Not much used ~~now~~ now'

```
pipeline {  
  agent any  
  stages{  
    stage("checking out code from github"){  
      steps{  
        git url: "giturl"  
      }  
    }  
  }  
}
```

Stage

name of the stage
(any name)

Here your project url will come

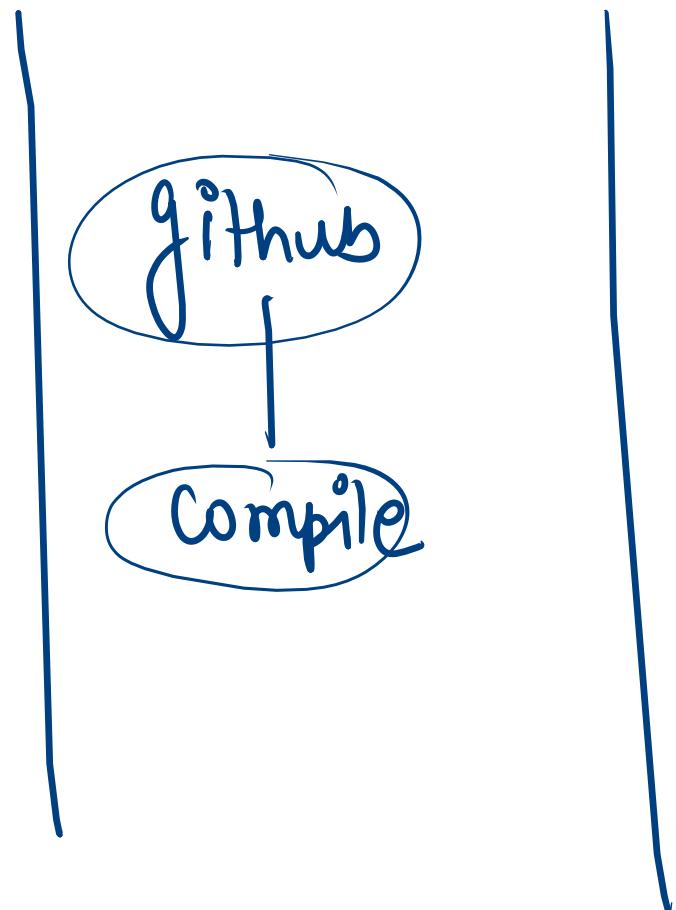


```
pipeline {  
    agent any  
    stages{  
        stage("checking out code from github"){  
            steps{  
                git url: "giturl"  
            }  
        }  
        stage("compling the project"){  
            steps{  
                sh "mvn compile"  
            }  
        }  
    }  
}
```

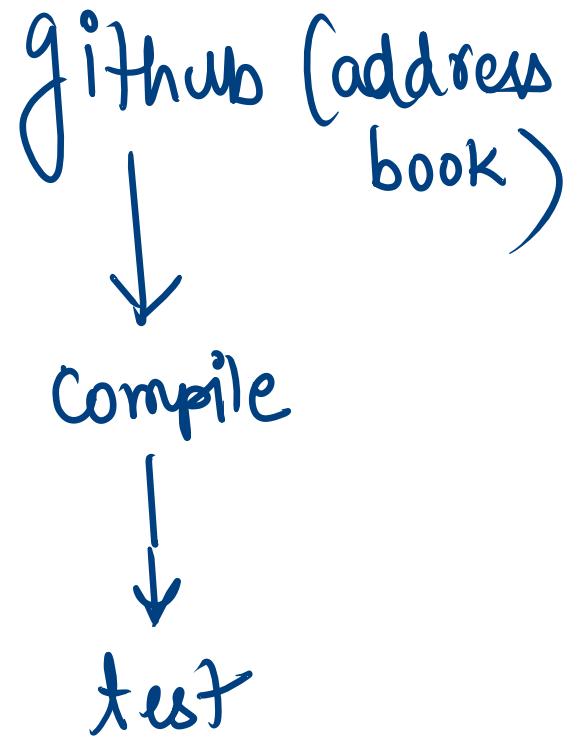
→ Stage 1

→ Stage 2

Shell terminal of machine



```
pipeline {  
    agent any  
    stages{  
        stage("checking out code from github"){  
            steps{  
                git url: "giturl"  
            }  
        }  
        stage("compling the project"){  
            steps{  
                sh "mvn compile"  
            }  
        }  
        stage("test the project"){  
            steps{  
                sh "mvn test"  
            }  
        }  
    }  
}
```



Create Item in Jenkins



New Item

Enter an item name

my-addressbook-pipeline

give any name

Select an item type



Freestyle project

Classic, general-purpose job type that checks out from up to one SCM, executes build steps like archiving artifacts and sending email notifications.



Pipeline

Orchestrates long-running activities that can span multiple build agents. Suitable for workflows) and/or organizing complex activities that do not easily fit in free-style job



Multi-configuration project

Suitable for projects that need a large number of different configurations, such as testing, platform-specific builds, etc.



Folder

Creates a container that stores nested items in it. Useful for organizing related items.

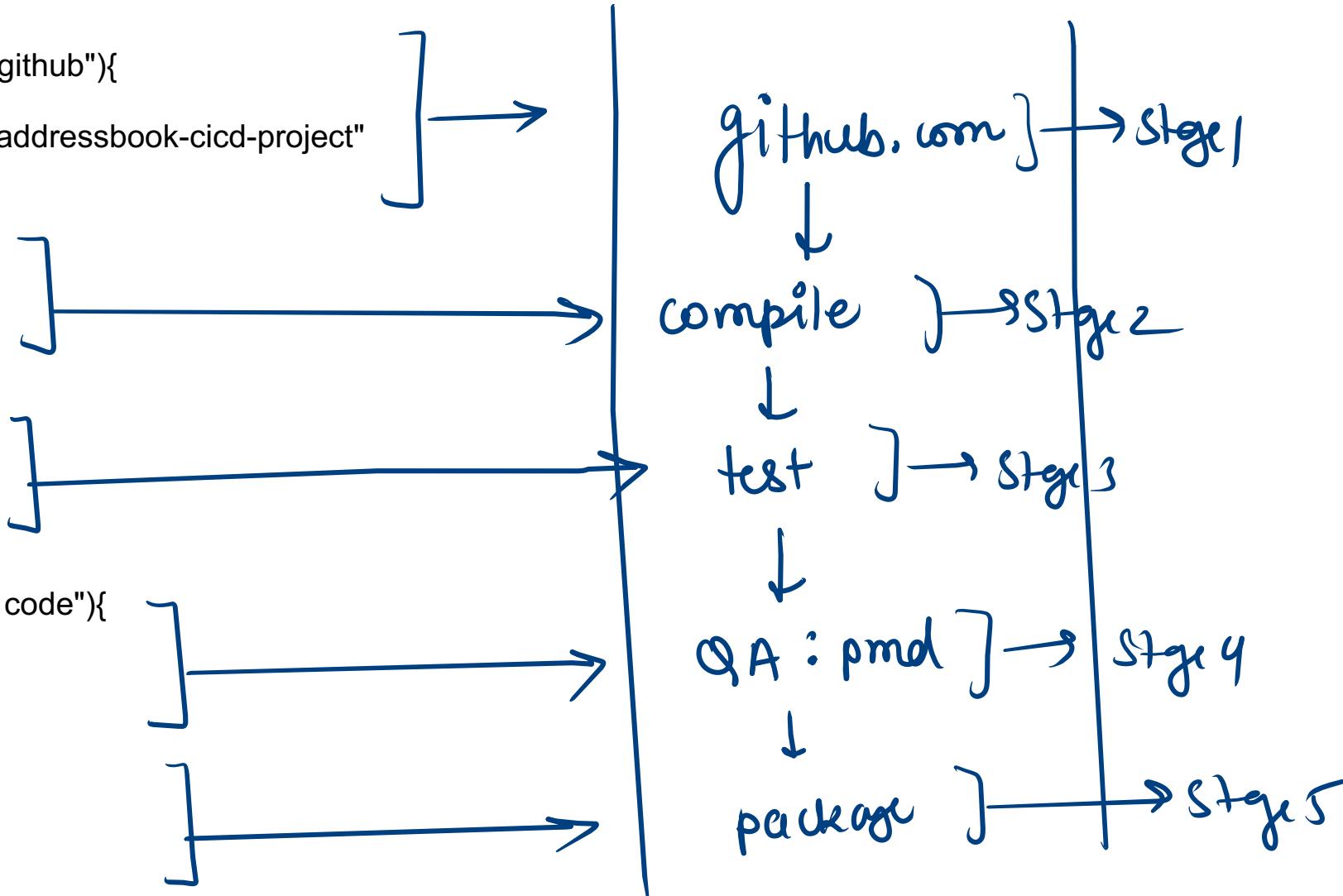
OK

Click on pipeline on left panel

```
pipeline{
    agent any
    stages{
        stage("1st stage: checkout the code from github"){
            steps{
                git url: "https://github.com/akshu20791/addressbook-cicd-project"
            }
        }
        stage("2nd stage: compile the code"){
            steps{
                sh "mvn compile"
            }
        }
    }
}
```

Save & let's see if the
code is running properly or not

```
pipeline{  
    agent any  
    stages{  
        stage("1st stage: checkout the code from github"){  
            steps{  
                git url: "https://github.com/akshu20791/addressbook-cicd-project"  
            }  
        }  
        stage("2nd stage: compile the code"){  
            steps{  
                sh "mvn compile"  
            }  
        }  
        stage("3rd stage: testing the code"){  
            steps{  
                sh "mvn test"  
            }  
        }  
        stage("4th stage: quality assurance of the code"){  
            steps{  
                sh "mvn pmd:pmd"  
            }  
        }  
        stage("5th stage: package the project"){  
            steps{  
                sh "mvn package"  
            }  
        }  
    }  
}
```



```
pipeline{
    agent any
    stages{
        stage("1st stage: checkout the code from github"){
            steps{
                git url: "https://github.com/akshu20791/addressbook-cicd-project"
            }
        }
        stage("2nd stage: compile the code"){
            steps{
                sh "mvn compile"
            }
        }
        stage("3rd stage: testing the code"){
            steps{
                sh "mvn test"
            }
        }
        stage("4th stage: quality assurance of the code"){
            steps{
                sh "mvn pmd:pmd"
            }
        }
        stage("5th stage: package the project"){
            steps{
                sh "mvn package"
            }
        }
    }
}
```

Build now

Status

</> Changes

▷ Build Now

⚙ Configure

Delete Pipeline

☰ Stages

✍ Rename

?

 Pipeline Syntax

✓ my-addressbook-pipeline

Permalinks

- Last build (#1), 9 min 19 sec ago
- Last stable build (#1), 9 min 19 sec ago
- Last successful build (#1), 9 min 19 sec ago
- Last completed build (#1), 9 min 19 sec ago

Builds

...

Filter

today

#2 3:52 PM

A screenshot of a pipeline interface showing a list of builds. The most recent build, labeled '#2 3:52 PM', is highlighted with a green oval. A blue arrow points from this highlighted item to handwritten text on the right.

click on latest build
to see the log created

Status

</> Changes

>- Console Output

✓ Edit Build Information

✗ Delete build '#2'

⌚ Timings

❖ Git Build Data

🕒 Pipeline Overview

⟳ Restart from Stage



#2 (Jun 5, 2025, 3:52:27 PM)



Started by user admin



This run spent:

- 20 ms waiting;
- 28 sec build duration;
- 28 sec total from scheduled to completion.



Revision: c7153b53bc451125ec84fb2d067583008485e154

Repository: <https://github.com/akshu20791/addressbook-cicd-project>

- refs/remotes/origin/master



Pipeline Overview

you would be
able to see the
steps done
in pipeline



< #2

Rebuild

Manually run by admin

Started 2 min 52 sec ago

Queued 3 ms

Took 28 sec

Graph



Bottom of this page we can even see the logs of each step.

To see the location where package is created (war)

The screenshot shows the Jenkins Pipeline Overview for a pipeline named 'my-addressbook-pipeline' with a single job '#2'. The 'Pipeline Overview' page displays the stages of the pipeline:

- 1st stage: checkout the code from git
- 2nd stage: compile the code 3.5 sec
- 3rd stage: testing the code 6 sec
- 4th stage: quality assurance of the co
- 5th stage: package the project 6.6 sec

The 5th stage is highlighted with a yellow oval. The log for this stage shows the command 'mvn package' and its output:

```
mvn package
+ mvn package
[INFO] Scanning for projects...
[WARNING]
[WARNING]
] Some problems were encountered while building the effective model for
com.edureka.demo.tutorial:addressbook:war:2.0
[WARNING]
] Reporting configuration should be done in <reporting> section, not in maven-site-plugin <>
as reportPlugins parameter. @ line 314, column 40
[WARNING]
```

A large blue arrow points downwards from the stage name '5th stage: package the project' towards the log output, with the handwritten note 'Scroll down' written next to it.

Below the log, the build results are summarized:

```
73 Tests run: 23, Failures: 0, Errors: 0, Skipped: 0
74
75 [INFO]
76 [INFO] --- maven-war-plugin:3.3.2:war (default-war) @ addressbook ---
77 [INFO] Packaging webapp
78 [INFO]
    ] Assembling webapp [addressbook] in [/var/lib/jenkins/workspace/my-addressbook-
    pipeline/target/addressbook]
79 [INFO] Processing war project
80 [INFO] Building war: /var/lib/jenkins/workspace/my-addressbook-pipeline/target/addressbook.war
81 [INFO] -----
82 [INFO] BUILD SUCCESS
83 [INFO] -----
84 [INFO] Total time: 4.709 s
85 [INFO] Finished at: 2025-06-05T15:52:55Z
86 [INFO] -----
```