



Jenkins

David Hubber

Hologate Gmbh

November 17th 2020


What is Jenkins?


- Jenkins is a Java server programme designed to manage and/or automate any number of tasks such as :
 - Continuous Integration (i.e. regular testing of code)
 - Compiling, building and deploying of code
 - Simply running a script or executable program at a set time every day/week
- For many Linux administrators, it might be able to replace many of the tasks that could be achieved with simple cron files
- Jenkins contains a huge variety of plug-ins that offer support for using Jenkins in combination with local or online services (e.g. Github)




Quick set-up


- Download Jenkins
 - I'm directly using the java war file, although it can be used in different ways
 - `java -jar jenkins.war --httpPort=8080`
- Open Jenkins in a browser
 - `http://localhost:8080`
- Copy the 'secret' administrator password following the prompts
- Install recommended plugins
- Create an admin account
- Update Jenkins and/or plugins
- Create pipeline to run Jenkins


Jenkins GUI


 **Jenkins**


search 


 1  David Hubber  log out


Jenkins 


 New Item


 People


 Build History

 Manage Jenkins


 My Views

 Lockable Resources

 New View


Build Queue 


No builds in the queue.







Build Executor Status 

1 Idle

2 Idle

 add description

All 

S	W	Name ↓	Last Success	Last Failure	Last Duration	
		NodeExample	47 sec - #2	N/A	0.1 sec	
		PipelineExample	4 min 20 sec - #1	N/A	0.35 sec	

Icon: [S](#) [M](#) [L](#)

[Legend](#) [Atom feed for all](#) [Atom feed for failures](#) [Atom feed for just latest builds](#)

The Jenkins Pipeline

- Pipelines are the most commonly used way of running jobs on Jenkins
- A pipeline is a script that defines the full structure of the job, including each stage, what commands are run, and in what order
 - They also determine what happens depending whether any stage fails with an error
- To create and run a pipeline :
 - Go to the main Jenkins menu (the dashboard)
 - Click on 'New Item'
 - Add a name, select 'Pipeline' from the list of options, and click 'OK'
 - Select all parameters for the pipeline, add in the script code, and click 'Save'
 - Click 'Build Now' in the pipeline menu to run the pipeline

An example pipeline (aka a Jenkinsfile)

- A pipeline contains various keywords defining different sections of the script
 - **pipeline**
 - entire script body
 - **agent**
 - external container that runs Jenkins
 - **stages**
 - contains all defined stages of pipeline
 - **stage**
 - an individual user-defined stage
 - **steps**
 - contains individual script code lines

```
pipeline {  
  agent any  
  stages {  
    stage('Hello') {  
      steps {  
        sh '''  
          echo "Hello Jenkins"  
          ls -lah  
          '''  
      }  
    }  
  }  
}
```

Scheduling

- Jenkins jobs can be easily scheduled for any time in the day, day of the week, or day in the month
- To schedule a pipeline build
 - Click on 'Configure' in the pipeline menu
 - Go to Build Triggers -> Build Periodically
 - Type in a schedule of the form
 - MIN (0 - 59) HOUR (0 - 23) DAY (1 - 31) MONTH (1 - 12) WEEKDAY (0 - 7)
 - e.g. To build every day at 8:30am Mon-Fri, 30 08 * * 1-6

Continuous Integration

- One of the most common uses of Jenkins is to create and maintain in-house Continuous Integration systems (e.g. like Travis CI on Github)
- This almost always requires the use of some version control software, such as Git or Subversion
- Such a system typically has the following stages
 - Build (i.e. compile the code)
 - Unit and integration tests (check for code correctness)
 - Install (on local machine) or package (i.e. prepare files to download and install elsewhere)

A simple CI example

```
pipeline {
  agent any
  stages {
    stage('Build')
    {
      steps {
        sh '''
            git submodule update --init
            cmake .
            make
            '''
      }
    }
    stage('Tests')
    {
      steps {
        sh '''
            git submodule update --init
            cmake .
            make
            ./Tests
            '''
      }
    }
  }
}
```

Processing script results

- In the script, you can define what to do based on the results of the stages in the post section of the script
 - **always**
 - **success**
 - **failure**
 - **unstable**
 - **changed**
- You can also email results using the 'mail' command

```
post {  
  success {  
    echo "Success!"  
  }  
  failure {  
    echo "Fail!"  
  }  
}
```

What else can Jenkins do?

(i.e. stuff I don't know how to do exactly)

- Execute using parameters
- Blue Ocean