

Jenkins lab 1

Part 1 – Explore IDE and create a Jenkin job

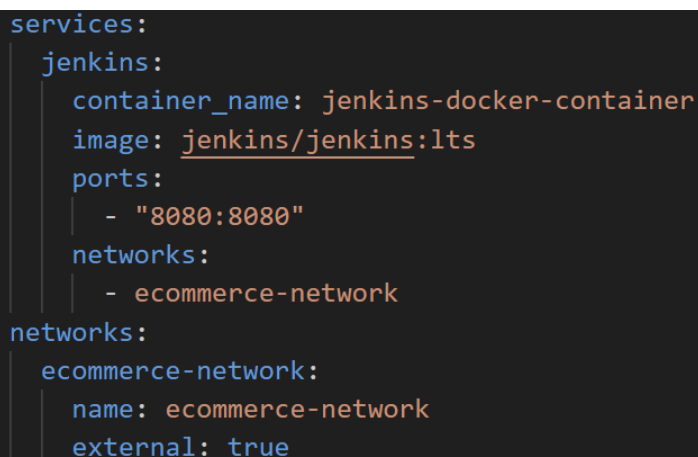
Objectives

- In this part you will install Jenkins using Docker technology.
- Investigate the Jenkins IDE.
- Create a basic Job and explore its options

- 1- Create a folder
- 2- Open the folder in VS-Code
- 3- Create a file called docker-composed.yml
- 4- Copy the following to the yml file

```
services:
  jenkins:
    container_name: jenkins-docker-container
    image: jenkins/jenkins:lts
    ports:
      - "8080:8080"
    networks:
      - ecommerce-network
networks:
  ecommerce-network:
    name: ecommerce-network
    external: true
```

Here is a picture showing the correct indentation



```
services:
  jenkins:
    container_name: jenkins-docker-container
    image: jenkins/jenkins:lts
    ports:
      - "8080:8080"
    networks:
      - ecommerce-network
networks:
  ecommerce-network:
    name: ecommerce-network
    external: true
```

- 5- Test your file by typing the following command in a Terminal window: **docker-compose config**
This command will type the contents of the file if it is valid.
- 6-
- 7- Run the command: **docker-compose up -d**
And then wait for a minute. This command will install Jenkins in detached and will not echo the secret password which you need to start Jenkins.
- 8-

- 9- Get the Jenkin's docker ID (**docker ps**). In this example we assume the ID starts with **e7b**
- 10- Run the following command to get the Jenkins secret password:
docker exec -it e7b cat /var/jenkins_home/secrets/initialAdminPassword
copy the secret password
- 11- In a browser type: **localhost:8080**
Copy the secret Administrator password from step 10

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log ([not sure where to find it?](#)) and this file on the server:

```
/var/jenkins_home/secrets/initialAdminPassword
```

Please copy the password from either location and paste it below.

Administrator password

- 12- Install the suggested plugins

Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

Install suggested plugins

Install plugins the Jenkins community finds most useful.

Select plugins to install

Select and install plugins most suitable for your needs.

- 13- Create the admin user (any name and password will do. The email is not checked)

Create First Admin User

Username

Password

Confirm password

Full name

E-mail address

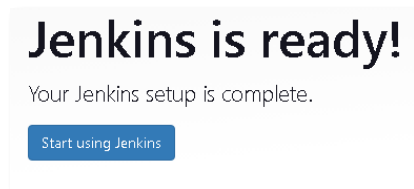
- 14- Follow the rest of the dialogs

Instance Configuration

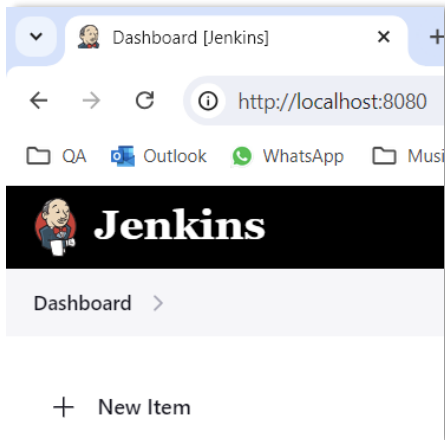
Jenkins URL:

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the BUILD_URL environment variable provided to build steps.

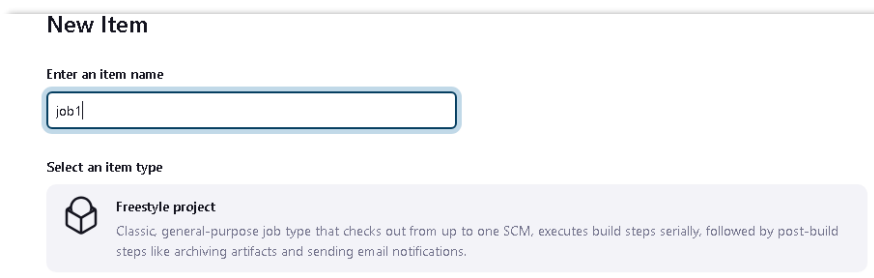
The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.



15- Create a new job by clicking [+ New Item]

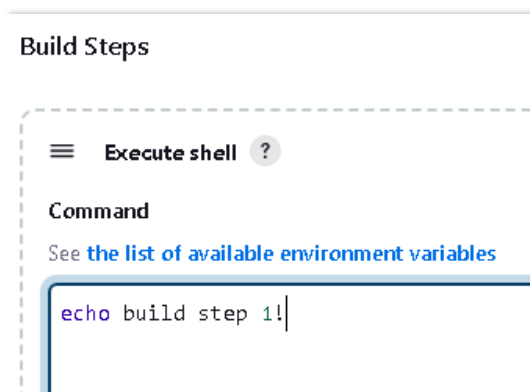
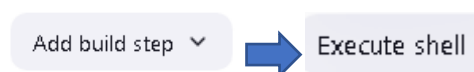


16- Choose a name for this job

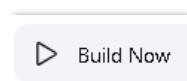


17- Scroll down and find the Build Step section and then select these options:


Build Steps



We will not deploy an application, only a simple script to get to know this tool.




18- Click **Save** and then


Build History
trend ▾


✓ #1
 | Aug 24, 2024, 6:07 PM

📡 Atom feed for all
📡 Atom feed for failures

19- Click on **#1** (build number) and then click  **Console Output**

Status

Changes

 **Console Output**

Edit Build Information

Delete build '#1'

Timings

✓


Console Output

```

Started by user admin
Running as SYSTEM
Building in workspace /var/jenkins_home/workspace/job1
[job1] $ /bin/sh -xe /tmp/jenkins5327622296918982256.sh
+ echo build step 1!
build step 1!
Finished: SUCCESS
          
```

20- Jenkins provides many environmental variables. We will have a look at a few. View these by clicking on the [See the list of available environment variables](#) link.

21- Click on **job1** title and then select the configure option


 **Configure**

☰ **Execute shell** ?

Command
 See [the list of available environment variables](#)

```

echo build step 1!
echo $BUILD_ID
          
```

 **Build Now**

22- Click

23- View the Console log and note the BUILD_ID value.

24- Modify Job1 by adding a few more lines

Add build step ▾

(Execute shell)

Execute shell ?

Command

See [the list of available environment variables](#)

```
echo builds step 2!  
echo $BUILD_NUMBER  
echo $BUILD_URL  
echo $WORKSPACE  
ls -la
```

Build Now

Build History trend ▾

Filter...

/

✓ #2

| Aug 24, 2024, 6:17 PM

✓ #1

| Aug 24, 2024, 6:07 PM

Atom feed for all

Atom feed for failures

25- View the console log

Status

Changes

Console Output

Edit Build Information

Delete build '#3'

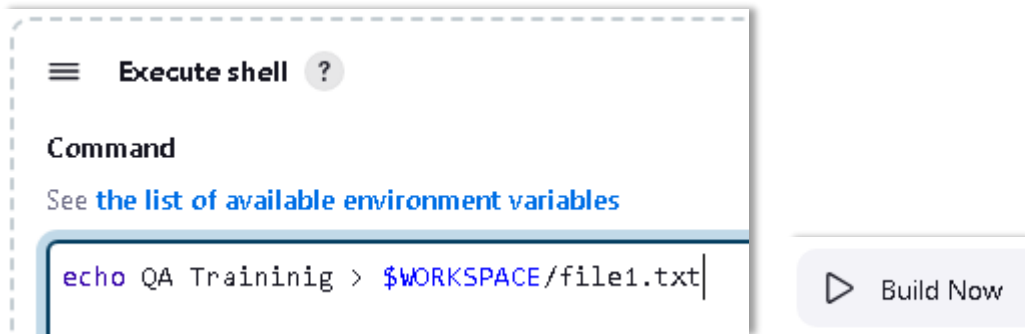
Timings

Previous Build

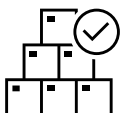
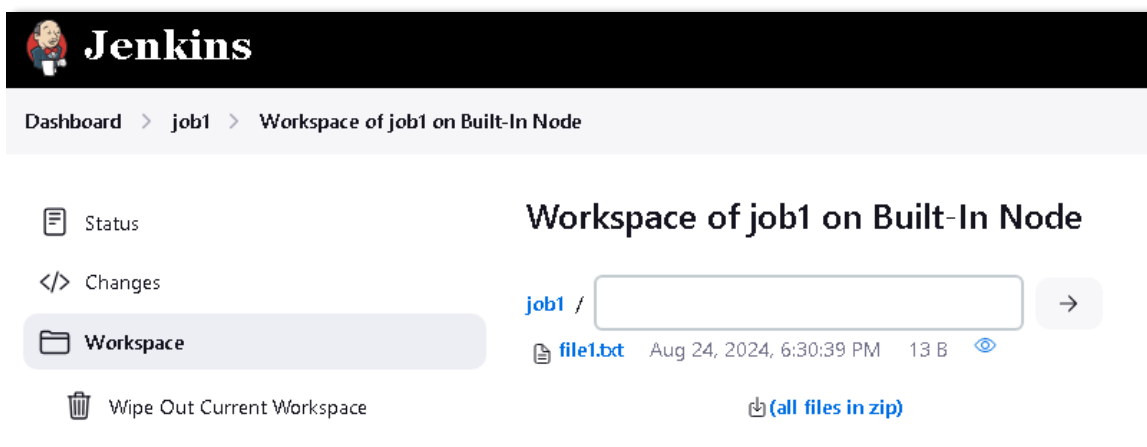
✓ Console Output

```
Started by user admin  
Running as SYSTEM  
Building in workspace /var/jenkins_home/workspace/job1  
[job1] $ /bin/sh -xe /tmp/jenkins2253034970788292338.sh  
+ echo build step 1!  
build step 1!  
+ echo 3  
3  
[job1] $ /bin/sh -xe /tmp/jenkins13255811510427251969.sh  
+ echo builds step 2!  
builds step 2!  
+ echo 3  
3  
+ echo http://localhost:8080/job/job1/3/  
http://localhost:8080/job/job1/3/  
+ echo /var/jenkins_home/workspace/job1  
/var/jenkins_home/workspace/job1  
+ ls -la  
total 8  
drwxr-xr-x 2 jenkins jenkins 4096 Aug 24 18:07 .  
drwxr-xr-x 3 jenkins jenkins 4096 Aug 24 18:07 ..  
Finished: SUCCESS
```

26- Add another step to the build to execute a shell command to reference Jenkin's Workspace which is a directory on the Jenkins server where the files related to a specific job are stored during the build process. The following command create a file called file.txt with content of "QA Training".



27- Lökk for the Job 1's Workspace menu and display the content of file.txt



Congratulations, you have successfully created a job using Jenkins and explored it's various IDE options. Please continue to **Part 2** below.

Part 2- Triggers

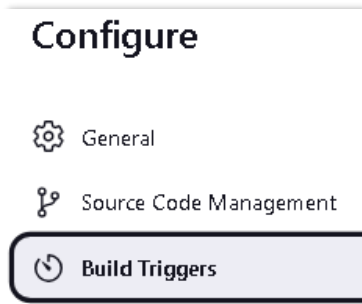
Objectives

In this part you will create a Jenkin **trigger**, and explore other forms of triggers in other labs.

Trigger is an event or action that initiates the execution of a job or pipeline (we will explore pipelines later).

Create a Remote build trigger

- 1- Configure Job1 and find the Buil Triggers section



- 2- Create a token for invoking and API end point for triggering a build (any secret token word will do)

Build Triggers

☒ Trigger builds remotely (e.g. from scripts) ?

Authentication Token

secret-token

Use the following URL to trigger build remotely: `JENKINS_URL/job/job1/build?token=TOKEN_NAME` or `/buildWithParameters?token=TOKEN_NAME`
Optionally append `&cause=Cause+Text` to provide text that will be included in the recorded build cause.

Save

- 3- Make a change to the last build and then save.

≡ Execute shell ?

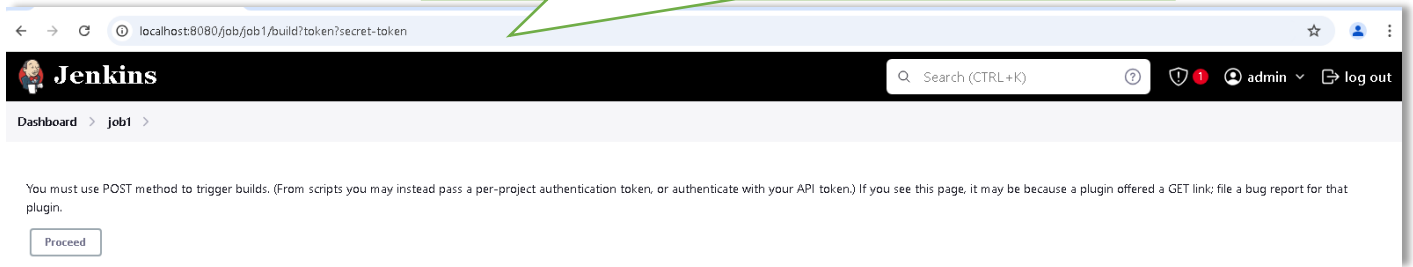
Command

See [the list of available environment variables](#)

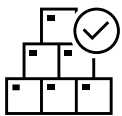
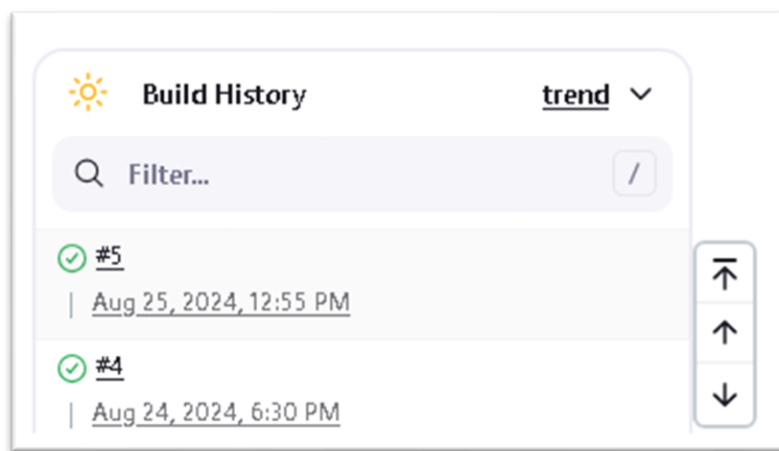
```
echo QA Traininig > $WORKSPACE/file1.txt
echo QA Traininig UK > $WORKSPACE/file2.txt
```

- 4- Type the following text in a browser and click the Proceed button to trigger the build.
Please note, this will result in a blank screen and will only shows errors if there are any.

`http://localhost:8080/job/job1/build?token?secret-token`



- 5- As mentioned, the screen will go blank but a new build is triggered (please wait a moment until it does)



Congratulations, you have successfully created a Jenkins trigger. This is an important concept and you will explore other forms of Jenkin's Triggers in other labs.
Please continue to **Part 3** below.

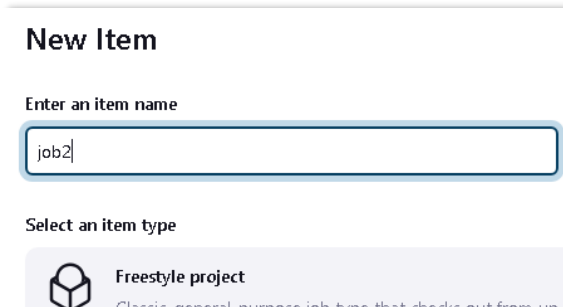
Part 3 - Chaining project builds

Objectives

In this part you will chain 2 or more builds together.

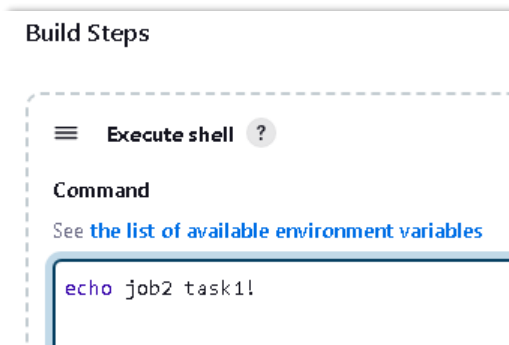
Build chaining in Jenkins automatically triggers one or more jobs after another job completes. This enables multi-step automation pipelines, where one job's output becomes the next job's input.

- 1- Create a new Job called **job2**



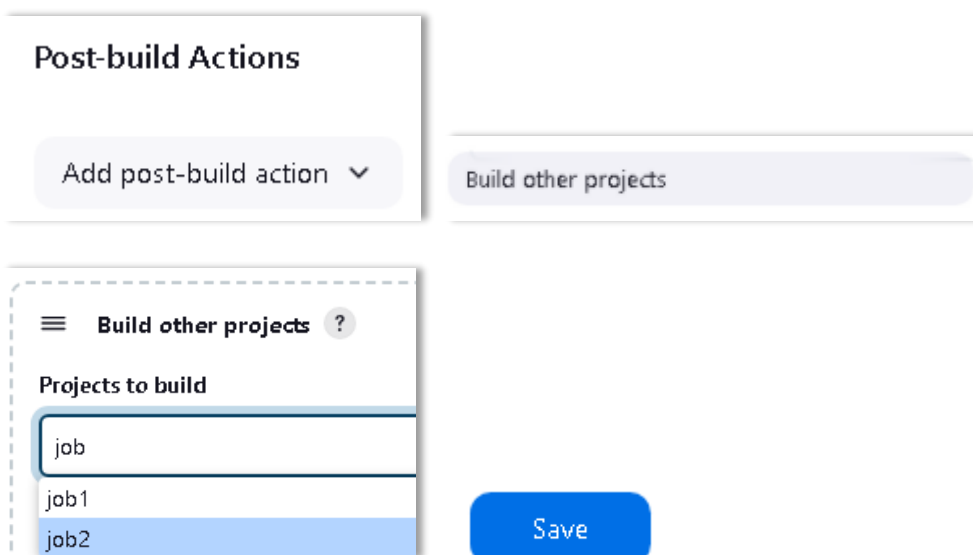
The 'New Item' form in Jenkins. It has a title 'New Item'. Below it is a label 'Enter an item name' followed by a text input field containing 'job2'. Below that is a label 'Select an item type' followed by a list of item types. The first item type is 'Freestyle project' with a cube icon and a description 'Classic, general purpose job type that checks out from un-'. The rest of the list is truncated.

- 2- Assign a build step like




The 'Build Steps' configuration in Jenkins. It has a title 'Build Steps'. Below it is a dashed box containing a menu icon, the text 'Execute shell', and a question mark icon. Below that is a label 'Command' followed by a link 'See the list of available environment variables'. Below that is a text input field containing 'echo job2 task1!'.

- 3- Configure Job1's post-build action to trigger building Job2



The 'Post-build Actions' configuration in Jenkins. It has a title 'Post-build Actions'. Below it is a button 'Add post-build action' with a dropdown arrow. To the right is a button 'Build other projects'. Below the 'Add post-build action' button is a dashed box containing a menu icon, the text 'Build other projects', and a question mark icon. Below that is a label 'Projects to build' followed by a list of projects: 'job', 'job1', and 'job2'. The 'job2' project is selected. To the right of the list is a blue button 'Save'.


 Build Now

- 4- Build Job1 and note that both jobs get built
View the console output of Job1 and Job2 for proof


Console Output

```
Started by upstream project "job1" build number 6
originally caused by:
  Started by user admin
Running as SYSTEM
Building in workspace /var/jenkins_home/workspace/job2
[job2] $ /bin/sh -xe /tmp/jenkins3223712646613369310.sh
+ echo job2 task1!
job2 task1!
Finished: SUCCESS
```

- 5- View Job1's status for the Downstream project (Job2)

 Status

 job1

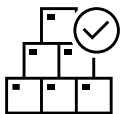
 Changes

Downstream Projects

 Workspace

 job2

 Build Now



Congratulations, you have successfully created a Jenkins build chain.

Please continue to Lab2 to explore building and deploying a Maven app stored on GIT.