

## Master II BDCC- Big Data et Cloud Computing

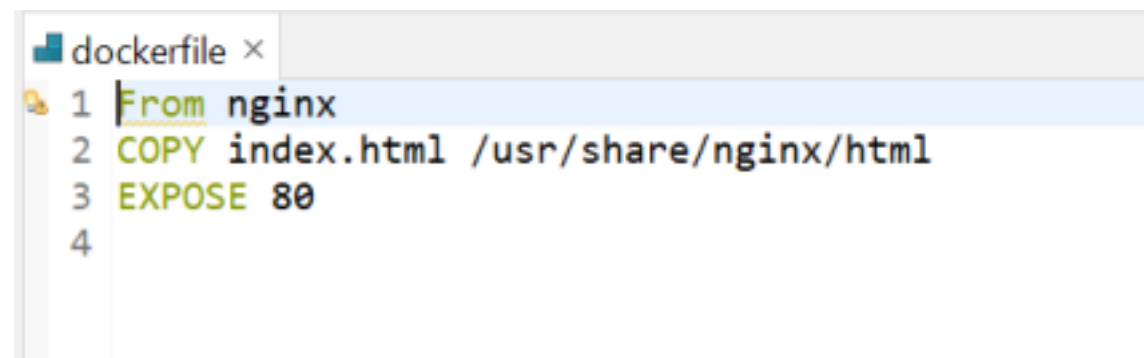
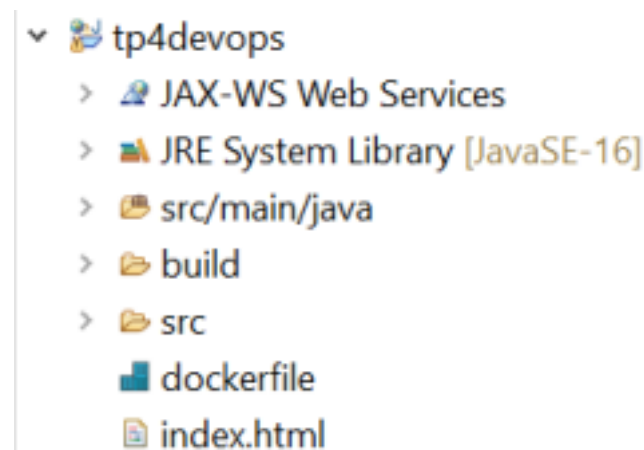
### TP 4 : Docker Engine, Jenkins, CI/CD

#### Partie 1: CI

1. Installer Jenkins.

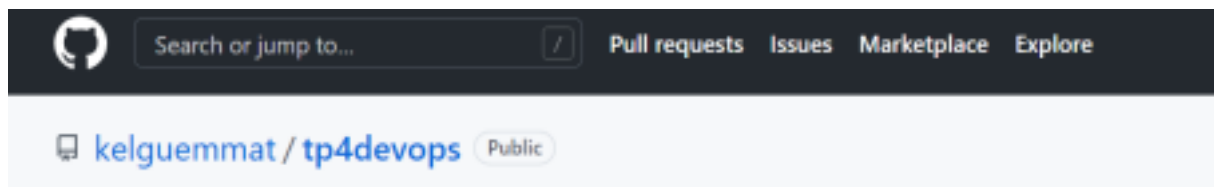
<https://www.jenkins.io/doc/book/installing/>

2. Créer un projet « tp4 » contenant une page web index.html, qui affiche « Welcome BDCC » et un fichier de configuration docker au répertoire du projet (un docker file qui permet de lancer cette page sur un serveur web nginx).



```
dockerfile *index.html x
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="ISO-8859-1">
5 <title>Insert title here</title>
6 </head>
7 <body>
8 welcome bdcc
9 </body>
10 </html>
```

3. Créer un répertoire Git hub nommée tp4 pour partager le code de l'application locale (tp4).



Gît dans le répertoire du projet avec la commande **git init**

• Ajouter tous les fichiers du projet au répertoire local avec la commande **git add \*** • **git config --global user.email** adresse email

• Enregistrer les changements dans le répertoire avec la commande **git commit -m "tp4 v1"** • Lier le répertoire local au répertoire git hub avec la commande **git remote add origin**

<https://github.com/kelguemmat/tp4devops.git>

• Pusher le code vers le répertoire GitHub avec la commande **git push origin master** 4. Créer et configurer un Job Jenkins (job1tp4) du type free style



2

UH2C/ENSET Travaux pratiques/ Ingénierie des Infrastructures Bigdata et Cloud 2022/2023 Pr.  
Kamal EL GUEMMAT

5. Ajouter des plugins docker à Jenkins.

MISES À JOUR    DISPONIBLES    **INSTALLÉS**    AVANCÉ

Activé	Nom	Version	Version précédente	Désinstaller
<input checked="" type="checkbox"/>	<b>CloudBees Docker Build and Publish plugin</b> This plugin enables building Dockerfile based projects, as well as publishing of the built images/repos to the docker registry.	1.3.3		Désinstaller
<input checked="" type="checkbox"/>	<b>Docker Commons Plugin</b> Provides the common shared functionality for various Docker-related plugins.	1.17		Désinstaller
<input checked="" type="checkbox"/>	<b>Docker Pipeline</b> Build and use Docker containers from pipelines.	1.26		Désinstaller

6. Configurer job1tp4 afin de générer une image (docker build) et publier une image docker du projet sur docker hub (Tag latest).

General    Gestion de code source    Ce qui déclenche le build    Environnements de Build    Build    Actions à la suite du build

Description
[Plain text] [Prévisualisation](#)

☐ Ce build a des paramètres ?  
☐ GitHub project  
☐ This build requires lockable resources  
☐ Supprimer les anciens builds ?  
☐ Throttle builds ?  
☐ Désactiver le projet ?  
☐ Exécuter des builds simultanément si nécessaire ?

Avancé...

Gestion de code source

☐ Aucune  
☒ [Git](#) ?

Repositories ?

Repository URL ?

Credentials ?  

kelguemmat/\*\*\*\*\* (github)
Ajouter

3

General
Gestion de code source
Ce qui déclenche le build
Environnements de Build
Build
Actions à la suite du build

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?
X
\*/master

Add Branch

Navigateur de la base de code ?

(Auto)

Additional Behaviours

Ajouter

Ce qui déclenche le build

☐ Déclencher les builds à distance (Par exemple, à partir de scripts) ?
☐ Construire après le build sur d'autres projets ?
☐ Construire périodiquement ?
☐ GitHub hook trigger for GITScm polling ?
☒ Scrutation de l'outil de gestion de version ?

Planning ?

\*\*\*\*\*

General
Gestion de code source
Ce qui déclenche le build
Environnements de Build
Build
Actions à la suite du build

Planning ?

\*\*\*\*\*

⚠ Vouslez-vous vraiment dire "chaque minute" avec l'expression "\*\*\*\*\*"? Peut-être voulez-vous dire "H\*\*\*\*\*"?

Aurait été lancé à vendredi 24 décembre 2021 à 20:53:41 heure d'été d'Europe de l'Ouest: prochaine exécution à vendredi 24 décembre 2021 à 20:53:41 heure d'été d'Europe de l'Ouest.

☐ Ignore post-commit hooks ?

Environnements de Build

☐ Delete workspace before build starts
☐ Use secret text(s) or file(s) ?
☐ Abort the build if it's stuck
☐ Add timestamps to the Console Output
☐ Inspect build log for published Gradle build scans
☐ With Ant ?

Build

Docker Build and Publish
X

Repository Name ?

kelguemmat/tp4

General

Gestion de code source

Ce qui déclenche le build

Environnements de Build

Build

Actions à la suite du build

Repository Name ?

kelquemmat/tp4

Tag

Docker Host URI ?

Server credentials

- aucun -

Ajouter

Docker registry URL ?

Registry credentials

kelquemmat/\*\*\*\*\* (dockerhub)

Ajouter

Avancé...

Ajouter une étape au build

Actions à la suite du build

Ajouter une action après le build

Sauver

Apply

## Propriétés globales

☐ Disable deferred wipeout on this node ?

☐ Emplacement des outils

☒ Variables d'environnement

Liste des paires clé-valeur ?

nom

DOCKER\_HOST

valeur

tcp://192.168.1.7:2375

Supprimer

Enregistrer

Appliquer

hub.docker.com/repository/docker/kelguemmat/tp4

**Advanced Image Management**  
View all your images and tags in this repository, clean up unused content, recover untagged images. Available with Pro, Team and Business subscriptions.

**kelguemmat / tp4**  
This repository does not have a description  
Last pushed: 4 days ago

**Docker commands**  
To push a new tag to this repository,  
`docker push kelguemmat/tp4:tagname`

**Tags and Scans**  
This repository contains 20 tag(s).  
VULNERABILITY SCANNING - DISABLED [Enable](#)

TAG	OS	PULLED	PUSHED
latest		4 days ago	4 days ago

**Automated Builds**  
Manually pushing images to Hub? Connect your repository to GitHub or Bitbucket to automatically build and tag your code as it is updated, so you can focus your development.  
Available with Pro, Team and Business subscriptions.

7. Faire un changement dans index.html, découvrir les changements sur le job1tp4.

Enregistrer les changements dans le répertoire avec la commande **git commit -m "tp4 v2"** Pusher le code vers le répertoire GitHub avec la commande **git push origin master**  
**D'après les changements** Déclenchement automatique du build sur Jenkins

Tableau de bord ▾ » job1tp4 ▸

État

Modifications

Répertoire de travail

Lancer un build

Configurer

Supprimer Projet

Log du dernier accès à Git

Renommer

Historique des builds

tendance ^

Filter builds...

#39 kelguemmat/tp4

20 déc. 2021 12:10

#38 kelguemmat/tp4

20 déc. 2021 11:36

#37 kelguemmat/tp4

20 déc. 2021 11:03

#36 kelguemmat/tp4

20 déc. 2021 10:38

Projet job1tp4

Espace de travail

Changements récents

Liens permanents

- Dernier build (#39 kelguemmat/tp4), il y a 4 j 8 h
- Dernier build stable (#39 kelguemmat/tp4), il y a 4 j 8 h
- Dernier build avec succès (#39 kelguemmat/tp4), il y a 4 j 8 h
- Dernier build en échec (#35 kelguemmat/tp4), il y a 4 j 10 h
- Dernier build non réussi (#35 kelguemmat/tp4), il y a 4 j 10 h
- Last completed build (#39 kelguemmat/tp4), il y a 4 j 8 h

8. Supprimer job1tp4.

7

UH2C/ENSET Travaux pratiques/ Ingénierie des Infrastructures Bigdata et Cloud 2022/2023 Pr.  
Kamal EL GUEMMAT

Tableau de bord ▾ » job1tp4 »

État

Modifications

Répertoire de travail

Lancer un build

Configurer

**Supprimer Projet**

Log du dernier accès à Git

Renommer

Historique des builds **tendance** ^

Filter builds...

- #39 kelguemmat/tp4  
20 déc. 2021 12:10
- #38 kelguemmat/tp4  
20 déc. 2021 11:36
- #37 kelguemmat/tp4  
20 déc. 2021 11:03
- #36 kelguemmat/tp4  
20 déc. 2021 10:38

## Projet job1tp4

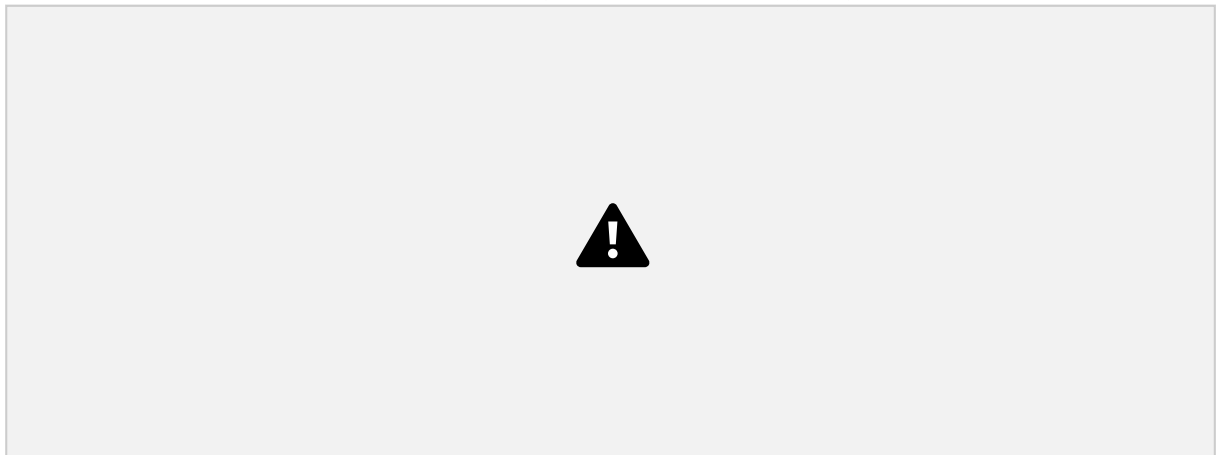
Espace de travail

Changements récents

### Liens permanents

- Dernier build (#39 kelguemmat/tp4), il y a 4 j 8 h
- Dernier build stable (#39 kelguemmat/tp4), il y a 4 j 8 h
- Dernier build avec succès (#39 kelguemmat/tp4), il y a 4 j 8 h
- Dernier build en échec (#35 kelguemmat/tp4), il y a 4 j 10 h
- Dernier build non réussi (#35 kelguemmat/tp4), il y a 4 j 10 h
- Last completed build (#39 kelguemmat/tp4), il y a 4 j 8 h

## Partie 2: CI/CD (continuous delivery/continuous deployment)



1. Créer un autre job freestyle job2tp4 contenant les mêmes instructions du job1tp4 de la première partie tout en ajoutant un script Shell qui déploie l'image sous un nouveau conteneur sur docker engine.





2. Faire un changement dans index.html, découvrir les changements sur le job2tp4 et sur l'image déployé.

Enregistrer les changements dans le répertoire avec la commande **git commit -m "tp4 v3"** Pusher le code vers le répertoire GitHub avec la commande **git push origin master**

**D'après les changements** □ **Déclenchement automatique du build sur Jenkins**

3. Créer un job du type pipeline job2tp4v2 (qui reprend les mêmes tâches du job freestyle job2tp4 mais d'une autre manière), ajouter sans rien changer dans les

paramètres du job, un script dans la partie script du pipeline assurant les trois stages (Cloning Git, Building image, Publish Image).

```
pipeline {  
  environment {  
    registry = "kelguemmat/tp4"  
    registryCredential = 'dockerhub'  
    dockerImage = "  
  }  
  agent any  
  stages {  
    stage('Cloning Git') {  
      steps {  
        git 'https://github.com/kelguemmat/tp4master21-22'  
      }  
    }  
    stage('Building image') {  
      steps{  
        script {  
          dockerImage = docker.build registry + ":$BUILD_NUMBER" }  
        }  
      }  
    stage('Publish Image') {  
      steps{  
        script {  
          docker.withRegistry( " , registryCredential ) {  
            dockerImage.push()  
          }  
        }  
      }  
    }  
  }  
}
```

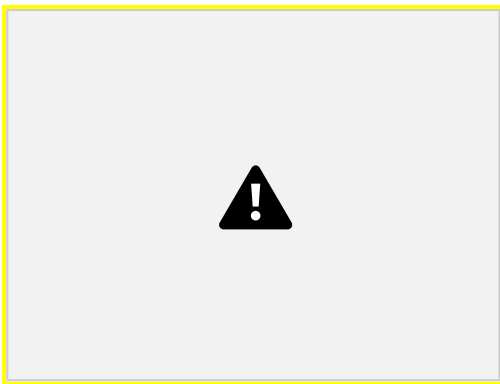
```
}
}
}
}
```

4. Créer un job du type pipeline (job3tp4). Ce dernier contiendra quatre Stages (Cloning Git, Building image, Test image, Publish Image). Sur le même projet TP4, créer un fichier 'jenkinsfile' qui définit le script assurant les quatre stages, par la suite spécifier sur le job le chemin du fichier 'jenkinsfile'.

Créer le Jenkins file sur le dépôt local et faire le push de ce dernier sur le dépôt distant :

```
pipeline {
  environment {
    registry = "kelguemmat/tp4"
    registryCredential = 'dockerhub'
    dockerImage = "
  }
  agent any
  stages {
    stage('Cloning Git') {
      steps {
        git 'https://github.com/kelguemmat/tp4master21-22'
      }
    }
    stage('Building image') {
      steps{
        script {
          dockerImage = docker.build registry + ":$BUILD_NUMBER"
        }
      }
    }
  }
}
```

```
}  
stage('Test image') {  
  steps{  
    script {  
  
    echo "Tests passed"  
  }  
}  
  
stage('Publish Image') {  
  steps{  
    script {  
      docker.withRegistry( " , registryCredential ) {  
        dockerImage.push()  
      }  
    }  
  }  
}  
}
```







5. Afficher stage view après quelques changements dans le projet (par exemple sur index.html).

Enregistrer les changements dans le répertoire avec la commande **git commit -m "tp4 v3"** Pusher le code vers le répertoire GitHub avec la commande **git push origin master**

**D'après les changements → Déclenchement automatique du build sur Jenkins**



6. Modifier le pipeline. Ce dernier contiendra cinq Stages (Cloning Git, Building image, Test image, Publish Image, deploy image). Ajouter sur le fichier jenkinsfile un stage du déploiement de l'image vers docker engine. Tester le changement via le stage view.

```
pipeline {
    environment {
        registry = "kelguemmat/tp4"
        registryCredential = 'dockerhub'
        dockerImage = "
    }
    agent any
    stages {
        stage('Cloning Git') {
            steps {
                git 'https://github.com/kelguemmat/tp4master21-22'
            }
        }
        stage('Building image') {
            steps{
                script {
                    dockerImage = docker.build registry + ":$BUILD_NUMBER"
                }
            }
        }
        stage('Test image') {
            steps{
                script {
                    echo "Tests passed"
                }
            }
        }
    }
}
```

```

    }
  }
  stage('Publish Image') {
    steps{
      script {
        docker.withRegistry( " , registryCredential ) {
          dockerImage.push()
        }
      }
    }
  }
  stage('Deploy image') {
    steps{
      bat "docker run -d $registry:$BUILD_NUMBER" }
    }
  }
}

```

**Ressources :**

<https://www.jenkins.io/>