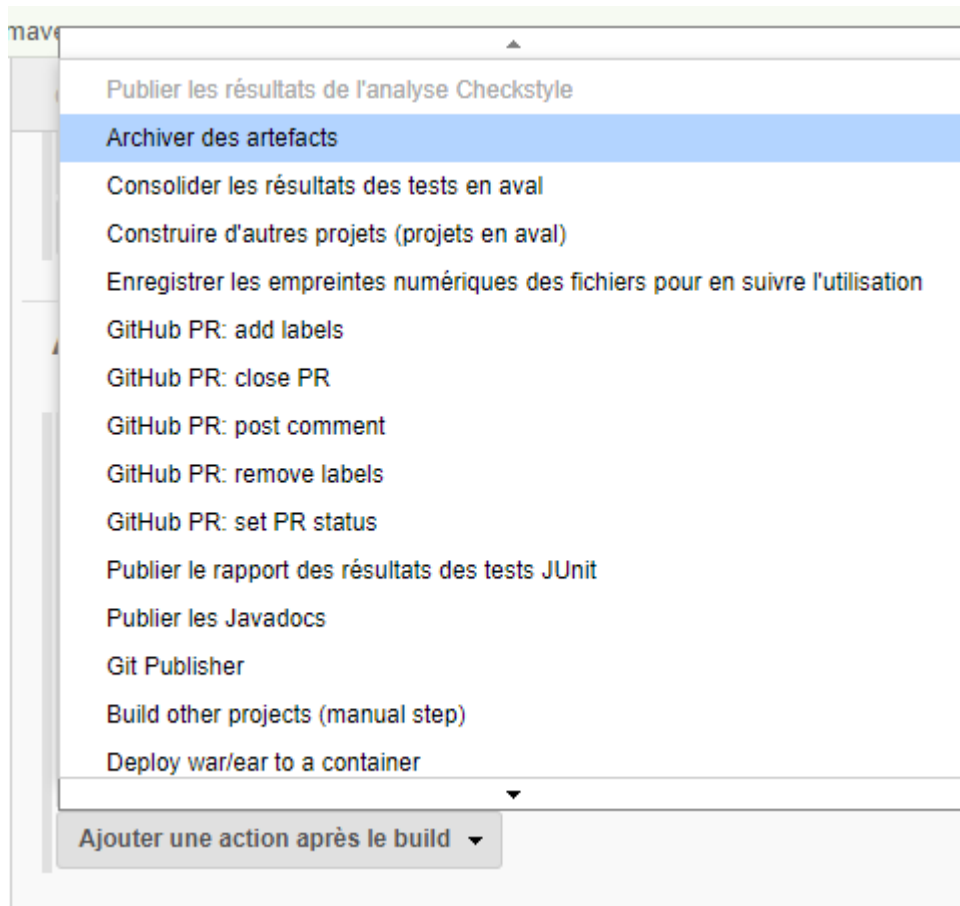


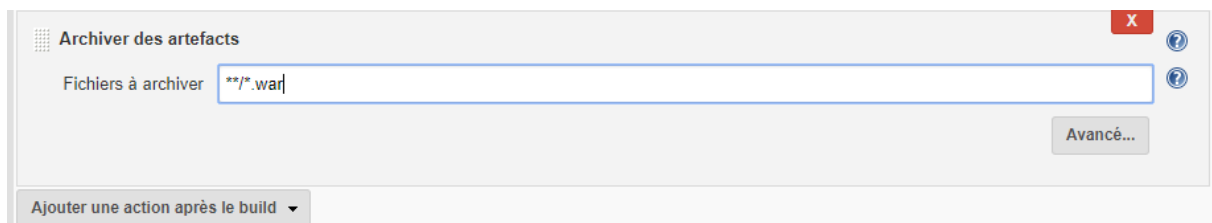
SECTION 4

TP – ARCHIVER UN WAR

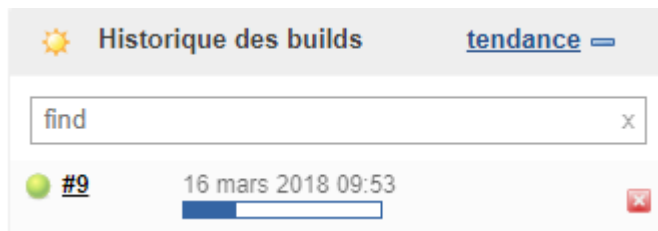
- Dans les configurations du projet « projet-maven », Allez à : **Actions à la suite du build**
- Ajoutez une action : Archiver des artefacts



- Indiquez à Jenkins d'archiver tous les fichiers se terminant par l'extension (.war)



- Sauvez les configurations et lancez un build depuis la vue projet



Suite à la fin du build, vous pouvez voir dans le détail de ce dernier, l'archive (.war) du projet

Construction #9 (16 mars 2018 09:53:33)



Artefacts du build



[webapp.war](#)

2,40 KB



[view](#)



No changes.



Lancé par l'utilisateur [admin](#)

- Regardez le détail de la console afin de trouver l'étape d'archivage

TP – INSTALLER ET CONFIGURER TOMCAT

- Allez sur le site d'apache : <https://tomcat.apache.org/download-80.cgi>

8.5.29

Please see the [README](#) file for packaging information. It explains what every distribution contains.

Binary Distributions

- Core:
 - [zip](#) ([pgp](#), [sha1](#), [sha512](#))
 - [tar.gz](#) ([pgp](#), [sha1](#), [sha512](#))
 - [32-bit Windows zip](#) ([pgp](#), [sha1](#), [sha512](#))
 - [64-bit Windows zip](#) ([pgp](#), [sha1](#), [sha512](#))
 - [32-bit/64-bit Windows Service Installer](#) ([pgp](#), [sha1](#), [sha512](#))

- Prenez la version **Tomcat 7** / core / zip, décompressez-le sur le poste de travail et renommez : **tomcat_dev**

Tomcat tourne sur le port **8080** qui est actuellement le port par défaut utilisé par **Jenkins**.

Il faut alors changer le port de Tomcat afin de pouvoir faire tourner ce serveur web.

Allons sur le dossier « **tomcat_dev** » qui contient les fichiers du serveur.

- Ouvrez le fichier : serveur.xml dans le dossier « conf » depuis l'explorateur windows
- Changer le port 8080 par le port : **8090**
- A l'aide de l'invite de commandes : lancer le serveur Tomcat

```
Administrateur : cmd

C:\>cd "Program Files"

C:\Program Files>cd apache-tomcat-8.5.28

C:\Program Files\apache-tomcat-8.5.28>cd bin

C:\Program Files\apache-tomcat-8.5.28\bin>startup.bat
```

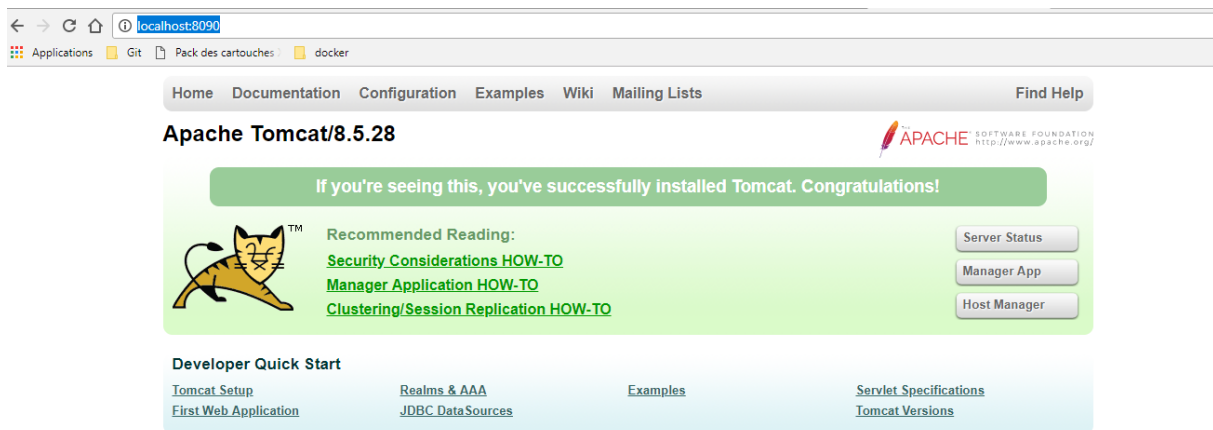
Si erreur liée à la JDK ou JRE, configurez :

Variables d'environnement

Variables utilisateur pour Babacar

Variable	Valeur
Nouvelle variable utilisateur	
Nom de la variable :	JAVA_HOME
Valeur de la variable :	C:\Program Files\Java\jdk1.8.0_161
Parcourir le répertoire...	Parcourir le fichier...

- Vérifiez que le serveur est bien démarré dans les logs
- Pour confirmer le démarrage du serveur, tapez : <http://localhost:8090> sur votre navigateur



Pour permettre à Jenkins de déployer automatiquement l'artefact sur le serveur Tomcat, ce dernier doit connaître certaines informations confidentielles concernant le serveur notamment sur les « users role ». Pour ce faire :

- Toujours dans le dossier conf, ouvrez le fichier : **tomcat-users.xml**

Pour permettre à Jenkins de déployer l'artefact sur le serveur Tomcat, nous avons besoin de configurer les 2 rôles que sont : « manager-script » et « admin-gui ».

- Zone de config (**Avant**)

```
<!--  
  <role rolename="tomcat"/>  
  <role rolename="role1"/>  
  <user username="tomcat" password="<must-be-changed>" roles="tomcat"/>  
  <user username="both" password="<must-be-changed>" roles="tomcat,role1"/>  
  <user username="role1" password="<must-be-changed>" roles="role1"/>  
-->
```

- Zone de config (**Après**)

```
<role rolename="admin-gui"/>  
<role rolename="manager-script"/>  
<user username="tomcat" password="tomcat" roles="admin-gui,manager-script"/>
```

- **N'oubliez pas d'enlever les commentaires de cette section !**
- Utilisez l'invite de commandes pour **redémarrer** le serveur (shutdown.bat et relancez le serveur par la suite : startup.bat).

TP – DEPLOYER SUR L'ENVIRONNEMENT DE DEV

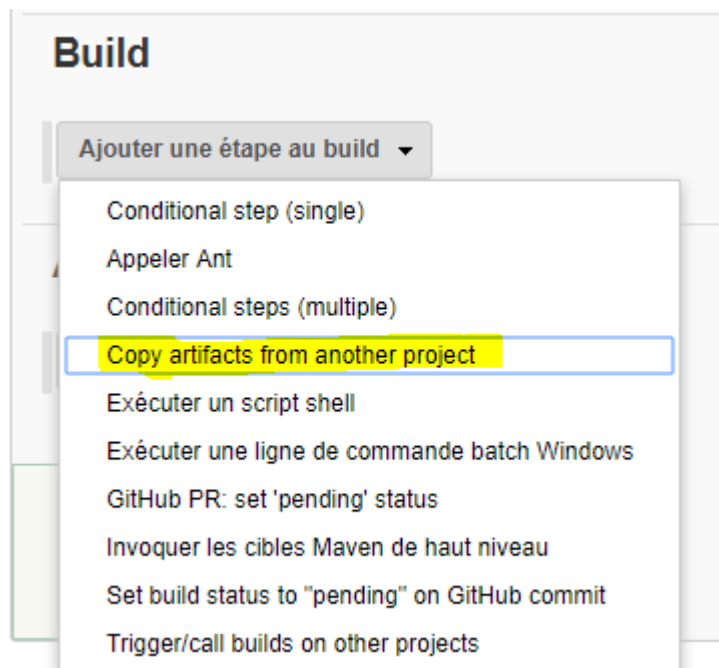
- Dans le même esprit que la section 3, installer les plugins
 - Copy Artificat
 - Deploy to container Plugin

Le job « **maven-project** » que nous utilisons a pour but de valider et de packager le projet. Pour une meilleure compréhension, renommons ce projet « **packaging** »

- Allez sur les configurations du job et changer le nom du job par « **packaging** »
- Sauvez la configuration

Nous allons créer le job qui a pour but de déployer l'application dans le serveur de développement.

- Dans la page d'accueil, créer un nouvel item
- Nommez le « **deploy-staging** » et sélectionner le type « free-style »
- Description : « Déploiement sur le serveur de développement »
- Allez sur la zone « Build » et ajoutez une étape du build



- Donnons à Jenkins le nom du projet concerné et l'extension des fichiers à copier.


The screenshot shows the 'Build' configuration page in Jenkins. The section 'Copy artifacts from another project' is expanded. The 'Project name' field is set to 'packaging'. The 'Which build' dropdown is set to 'Latest successful build'. The 'Artifacts to copy' field contains the wildcard pattern '**/*.war'. Below this, there are checkboxes for 'Stable build only' (unchecked), 'Flatten directories' (unchecked), 'Optional' (unchecked), and 'Fingerprint Artifacts' (checked). There are also empty fields for 'Artifacts not to copy', 'Target directory', and 'Parameter filters'. A red 'X' button is in the top right corner of the section, and an 'Avancé...' button is at the bottom right.

Dans les actions post-build, nous allons demander à Jenkins de déployer le (.war) obtenu dans le précédent « job »


- Dans la zone « **Actions à la suite du build** », ajoutez l'étape « **Deploy war/ear to a container** » et spécifier les types de fichiers concernés

The screenshot shows the 'Actions à la suite du build' configuration page. The 'Deploy war/ear to a container' action is added. The 'Fichiers WAR/EAR' field is set to '**/*.war'. The 'Context path' field is empty. The 'Containers' section has an 'Add Container' button. The 'Deploy on failure' checkbox is unchecked. At the bottom, there is a button 'Ajouter une action après le build' with a dropdown arrow.

- Ajoutez le serveur Tomcat installé grâce à l'option « add Container »
- Choisissez votre serveur dans la liste déroulante
- Dans la rubrique « credentials », ajoutez vos données utilisateurs confidentiels du fichier (**tomcat-users.xml**) dans le nouvel onglet qui apparaît :



Jenkins Credentials Provider: Jenkins

 **Ajouter des identifiants**

Domain

Type

Portée

Nom d'utilisateur


Mot de passe

ID

Description

- Sélectionner le profil (**user/pass**) dans la liste des « credentials » et renseigner l'adresse de votre serveur web (Tomcat URL)


Actions à la suite du build

 **Deploy war/ear to a container**

Fichiers WAR/EAR

Context path

Containers

 **Tomcat 8.x**





Credentials

Tomcat URL


Deploy on failure ☐

- **Sauvez** la configuration

Dans la page d'accueil de notre serveur Jenkins, nous avons maintenant :

TP			
Tous			
build pipeline			
+			
S	M	Nom du projet ↓	Dernier succès
		deploy-staging	s. o.
		packaging	2 h 0 mn - #9

Icône: [S](#) [M](#) [L](#)

[Légende](#)  [RSS pour tout](#)

Nous avons abouti à configurer « **deploy-staging** » qui récupère l'artefact généré par le build de « **packaging** » et le déploie dans notre serveur Tomcat. Voyons maintenant comment lancer ce processus **automatiquement** après le build du préalable job « packaging »

Retournons à la configuration du job « packaging »

- Allez dans la zone « **Actions à la suite du build** »
- Sélectionner la rubrique : Construire d'autres projets (projets en aval)
- Inscrivez le nom du projet à construire après une construction de celui-ci

Construire d'autres projets (projets en aval)

Projets à construire

☒ Déclencher que si la construction est stable

☐ Déclencher même si la construction est instable

☐ Déclencher même si la construction échoue

Ajouter une action après le build ▼

- **Sauvez** la configuration et revenez à la page d'accueil

Nous allons maintenant lancer le projet « **packaging** » et vérifier si ce dernier fait ensuite appel au projet « **deploy-staging** » afin qu'il **déploie** le (.war) dans notre serveur Tomcat.

- Lancez le job « **packaging** » via le bouton 

File d'attente des constructions 

File d'attente des constructions vide

État du lanceur de compilations 

1 packaging  #10 

2 Au repos

Puis à la fin de celui-ci, le projet « **deploy-staging** » est lancé.

File d'attente des constructions (1) 

deploy-staging 

État du lanceur de compilations 

1 Au repos

2 Au repos

Ensuite, rafraichir la page et vérifier que les 2 projets ont été construits avec **succès**.

Vous pouvez jeter un coup d'œil dans les détails des projets et des Builds pour voir les liens entre les deux projets et voir la console qui détaille les constructions.

Maintenant, nous pouvons aller sur notre navigateur et vérifier que la page d'accueil de notre application est bien disponible.

- Tapez : <http://localhost:8090/webapp/> ;))

Simuler une modification et vérifier le processus !