Deploying with Jenkins

Both Amir and I have just recently started at Codurance as apprentices. This week, one of our tasks was to set up Jenkins on a new server then add a client’s project for building and testing. Thanks to an excellent tutorial from Jeff Shantz, it ended up being a relatively straight forward task. We then wanted to create a way for the client to be able to download their tested application. However we got more than a little stumped when it came to deploying those assets to a new location on the server. Frantic Googling had us going around in circles.

Luckily, a pairing session with Codurance Craftsman Samir pointed us in the right direction we thought we would set out our process below in case any other Jenkins newbies got stuck like we did.

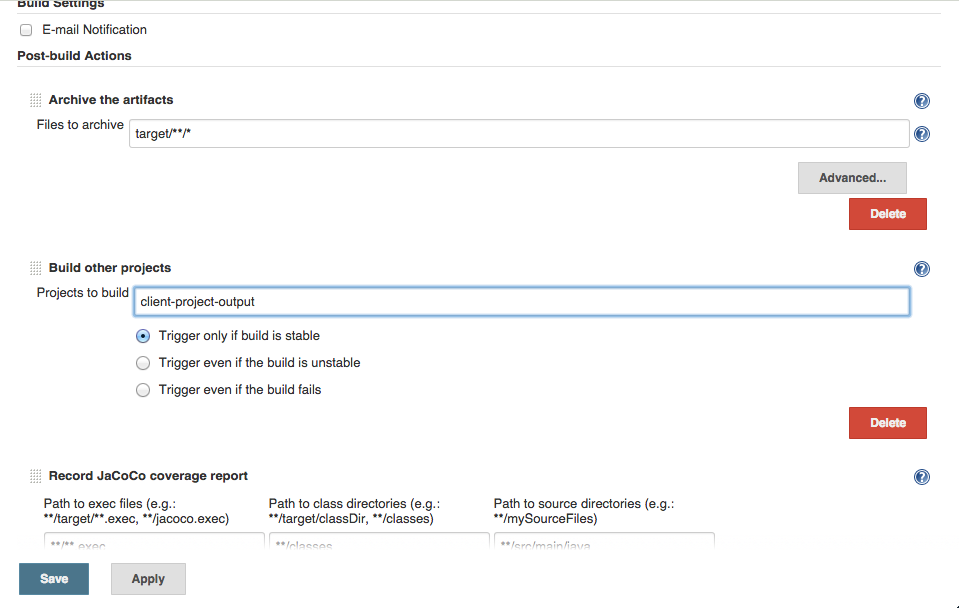
Note, I'll assume you have a client project set up on Jenkins, and it is connected to your version control system. If you have not got this far then checking out the link to the tutorial earlier will help.

Step 1: Create a new Jenkins Item

Select 'New item' from the main menu and call it something like "[ENTER-PROJECT-NAME]-Output". Its purpose will be solely export the files from your client project to a folder of your choosing on the server.

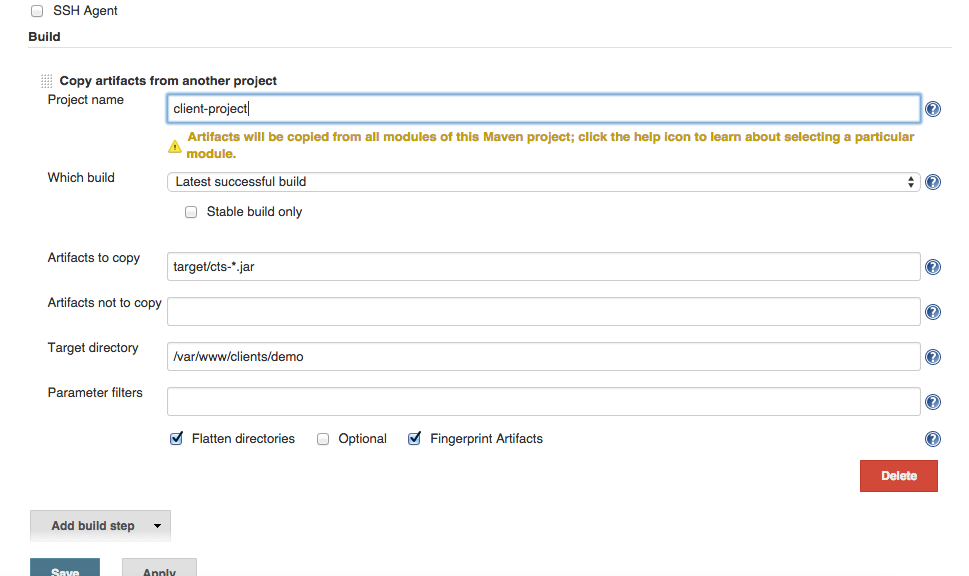
Step 2: Create a post-build action

Go to your client project and select configure. Create a post-build action and select ‘archive artififacts’ from the drop down menu. Add the type of files you want to archive (and eventually, copy and export). Next add another post-build action ‘Build other project’ and enter the name of the build item you created earlier.



Step 3: Install the Copy Artifact plugin

Next you need to install the [Copy Artifact plugin](https://wiki.jenkins.io/display/JENKINS/Copy+Artifact+Plugin) in the Manage Plugins section of Jenkins. Go to "[PROJECT-NAME]-Output" > configure and add a new build step. Because you have installed the Copy Artifact plugin you should see an option called ‘copy artifacts from another project’ in the drop down menu. Specify the folder or files you want copied and set the location path. Notice that we set set our location to "var/www/clients/…". This leads to a new folder on the server (we were using an Apache server on an Amazon EC2 instance). Don't do what we did and set the path using an http address(!).



Step 4: Test it Out

Trigger a build from your client-project. This should then trigger a new build in [PROJECT-NAME]-Output. Check the deployment folder you set up on the server. Hopefully you will see the newly-deployed files but...

...Wait, the build failed! I got a FileException error...

It may mean that Jenkins does not have the right permissions to write to the folder and cannot deploy the files.

SSH into your server and check the permissions of you output folder. As you might have guessed we ran into this problem and manage to fix this by:

1. adding Jenkins to a group, in our case, the www-data group.
2. changing ownership of the output folder to the www-data group with the command *sudo chown -R :www-data clients*.
3. enabling write access on the folder with the command *sudo chmod -R g+w clients*.

Don’t forget to restart your server to finalise the changes!