

Continuous deployment of a jenkins job.

- Creation of job in Jenkins
- Creating declarative pipeline
- Remote access without password

As all jenkins jobs we are using, this is two parts:

1. Creation of the job in jenkins
2. Creating the declarative pipeline in repository [jenkinsjobs](#).

Creation of job in Jenkins

The job created in Jenkins will be stored as an xml document inside the jenkins machine. This xml could be stored in Source Control, but for some reason, it isn't. So take care while modifying jenkins jobs, as there is no backup unless you explicitly ask for a backup of the machine (Alberto Robles).

So the job I have created for now is this one:



This job will be triggered when the Server Integration one finishes building. The first job needs to be archiving their artifacts (Archive Artifact plugin), which was already doing so, and the second job, the one doing the deployment needs to read such artifacts as needed. To do so, the first job must explicitly say which jobs are allowed to read its archived artifacts.

As a general rule, working with Pipelines in Jenkins is confusing, as there are parts that are defined in the jenkins job, while there are parts defined in the pipeline. This is one of the examples.

So, for the first job, Tajikistan_Server_Git, the archiving part is defined in the pipeline:

```
→ success {
  →   junit 'Server/*TEST*.xml'
  →   archiveArtifacts artifacts: 'Server/Solutions/IMS/build/libs/*, Test/Staging/staging_conf.json, Test/Integration/development_conf.json, Test/Integration/validation_conf.json, Test/Integration/dmz_conf.json', onlyIfSuccessful
  → }
```

while defining which projects can read from this, one, is done in the job, in the jenkins ui:



As you can see, we need to define that we give permission to the tailing job to read these artifacts.

In the new job, Deploy_Tajkistan_Server, we need to define some things:
Trigger the build automatically, after the build has finished:

Build Triggers

Build after other projects are built ?

Projects to watch

Tajikistan_Server_Git

Trigger only if build is stable

Trigger even if the build is unstable

And then, the pipeline we are using:



And if you want, the environment in which to deploy:



Now, this is defined to deploy the last successful build, but that is not defined here, but in the pipeline.

Creating declarative pipeline

Pipeline file is Jenkinsfile_Tajikistan_Deploy_Server.groovy.

Just two steps:

- Copy artifacts.
- Deploy to JBoss.

Copy:

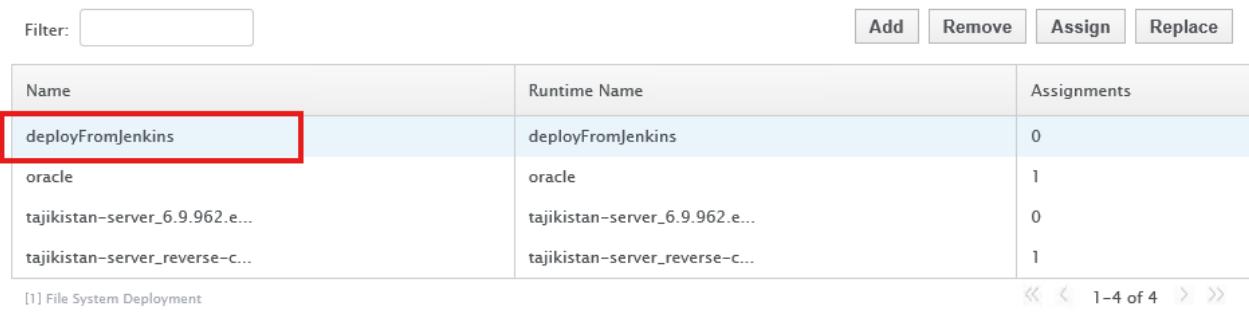
```
.....steps {  
.....    copyArtifacts(  
.....        projectName: 'Tajikistan_Server_Git',  
.....        selector: [$class: 'StatusBuildSelector', stable:false],  Which build  
.....        filter: 'Server/Solutions/IMS/build/libs/tajikistan-server.ear',  
.....        target: '.'  
.....    )  
.....}  
.....}
```

Deploy into jboss:

Use jboss command line to do so. Also, an auxiliary file for the deployment, as there were too many interpreters to escape:

```
....// 3. Deploy into JBoss  
....def jbossCliCmd = 'C:/Installation/jboss-eap-6.4/bin/jboss-cli.bat --connect --controller=localhost:9999 --file=C:/DeploymentStaging/deploy.cli'  
....sh """  
....echo "Complete CLI command: Set-Location C:\\; ${jbossCliCmd}"  
....ssh -i "$KEYFILE" -o StrictHostKeyChecking=no \  
....          -o PubkeyAcceptedKeyTypes=+ssh-rsa \  
....          -o HostKeyAlgorithms=ssh-rsa \  
....          -o UserKnownHostsFile=/dev/null \  
....${USER}@${remoteIp} \  
....powershell -Command "${jbossCliCmd}"  
...."
```

Finally, you will be able to see the deployed artifact in your jboss instance:



Name	Runtime Name	Assignments
deployFromJenkins	deployFromJenkins	0
oracle	oracle	1
tajikistan-server_6.9.962.e...	tajikistan-server_6.9.962.e...	0
tajikistan-server_reverse-c...	tajikistan-server_reverse-c...	1

[1] File System Deployment << < 1-4 of 4 > >>

Attributes Path

Remote access without password

As the process needs to be non interactive, we need to create an ssh key. Ideally we would create it in the jenkins machine, but as we don't have access we create it locally:

```
ssh-keygen -t rsa -b 2048 -m PEM -f jenkins_deploy_rsa_legacy
```

We are using an old format here because the servers are old.

This will create two keys: public and private.

We will use the private key in our jenkins credentials, and the public in our servers, as authroized keys.

Cat your private key and use it as credentials in jenkins:

```
$ cat jenkins_deploy_rsa_legacy
-----BEGIN RSA PRIVATE KEY-----
MIIEpQIBAAKCAQEAnZMdEDjgggcDQF/77eZgDQta5uLmlzVz267v2lwxtd/i9tJt
.
.
.
.

CIEng57LHHueMdBeZ9jxTzGcXo1U8zY9AIUTtfJzoM9o38rl743h0so=
-----END RSA PRIVATE KEY-----
```

Use it in jenkins, defining the user you want to use: Manage Jenkins Manage Credentials:

T	P	Store	Domain	ID	Name
		Jenkins	(global)	3970d8d0-2163-470a-8432-b32e88eebb1f	3970d8d0-2163-470a-8432-b32e88eebb1f
		Jenkins	(global)	Sonar	Sonar
		Jenkins	(global)	sonar token	sonar token
		Jenkins	(global)	21763f33-dde2-46e1-b60f-92b424152a85	SG_MAD_EJOURNEY_CI/*****
		Jenkins	(global)	FORTIFY_AUTH_TOKEN	Fortify Authentication Token (Expires in 356 days) - ADP
		Jenkins	(global)	SG_MAD_EJOURNEY_CI	SG_MAD_EJOURNEY_CI/***** (Bitbucket)
		Jenkins	(global)	b8db197c-8e86-47fe-8006-e183c9f962ba	invitado/***** (Map Q: network drive)
		Jenkins	(global)	jenkins_deploy	Administrator (Deploy files to legacy servers. You will need your public key for your client.)

Here Administrator is the user with which to log in our machines.

To create a new one, push in the Global Domain:

T	P	Store	Domain	ID	Name
		Jenkins	(global)	3970d8d0-2163-470a-8432-b32e88eebb1f	3970d8d0-2163-470a-8432-b32e88eebb1f
		Jenkins	(global)	Sonar	Sonar
		Jenkins	(global)	sonar token	sonar token
		Jenkins	(global)	21763f33-dde2-46e1-b60f-92b424152a85	SG_MAD_EJOURNEY_CI/*****
		Jenkins	(global)	FORTIFY_AUTH_TOKEN	Fortify Authentication Token (Expires in 356 days) - ADP
		Jenkins	(global)	SG_MAD_EJOURNEY_CI	SG_MAD_EJOURNEY_CI/***** (Bitbucket)
		Jenkins	(global)	b8db197c-8e86-47fe-8006-e183c9f962ba	invitado/***** (Map Q: network drive)
		Jenkins	(global)	jenkins_deploy	Administrator (Deploy files to legacy servers. You will need your public key for your client.)

Icon: S M L

Stores scoped to Jenkins

P	Store	Domain
	Jenkins	(global)

Icon: S M L

and then Add Credentials:

ID	Name	Kind	Description
3970d8d0-2163-470a-8432-b32e88eebb1f	3970d8d0-2163-470a-8432-b32e88eebb1f	Username with password	
	Sonar	Username with password	Sonar
	sonar token	Secret text	
	21763f33-dde2-46e1-b60f-92b424152a85	Username with password	
	FORTIFY.AUTH_TOKEN	Secret text	Fortify Authentication Token (Expires in 356 days) - ADP
	SG_MAD_EJOURNEY_CI	Username with password	SG_MAD_EJOURNEY_CI/***** (Bitbucket)
	b8db197c-8e86-47fe-8006-e183c9f962ba	Username with password	invitado/***** (Map Q: network drive)
	jenkins_deploy	SSH Username with private key	Administrator (Deploy files to legacy servers. You will need your public key for your client.)

Icon: S M L

Global credentials (unrestricted)

Credentials that should be available irrespective of domain specification to requirements matching.

And use your private key there, this is an example:

Dashboard > Credentials > System > Global credentials (unrestricted) >

Back to credential domains

Add Credentials

Kind

SSH Username with private key

Scope ?
Global (Jenkins, nodes, items, all child items, etc)

ID ?
jenkins_deploy2

Description ?

Username
Administrator

Treat username as secret ?

Private Key

Enter directly
 Key

Enter New Secret Below

-----BEGIN RSA PRIVATE KEY-----
MIIEidOBAAKCAoEAnDmED1aaecDOF77ezaQ0taSulnIvvz267v1wxtd1ptjt

Passphrase

OK

And in the clients (In this case, our clients are the servers where the deployment is to be made), we must have the public keys in the authorized_keys files. In our case, we are using the ssh server of cygwin, so the correct file where to place the keys is in /cygdrive/c/Users/Administrator/.ssh/authorized_keys

Here you can see the dir route in linux and windows mode:

```
Administrator@Tajikistan-HQ-Master ~/.ssh
$ cygpath -w $PWD
C:\cygwin\home\Administrator\.ssh

Administrator@Tajikistan-HQ-Master ~/.ssh
$ pwd
/home/Administrator/.ssh
```

Here you need to add your public key (`jenkins_deploy_rsa_legacy.pub`) in the `authorized_keys` file:

```
Administrator@Tajikistan-HQ-Master ~/.ssh
$ ls
authorized_keys  known_hosts
```

That way, you will be able to use the ssh commands from the jenkins pipeline without the host asking for user/password.