**Index**

1. **Linux\_naukri**
2. **Market\_place & Shine**
3. **Converting\_docs**

**1) Linux\_naukri**

This suite Linux\_naukri is build with four different suites like

1. naukriapi\_pre\_setup
2. naukriapi\_client\_setup
3. naukriapi\_post\_setup
4. naukriapi\_patch\_updates
5. **Naukriapi\_pre\_setup**

When we are building naukri\_pre\_step we use a freestyle project and create a suite.

* In general we select “**This project is parameterized**” to pass the client ip as hip , username as uname and client server password as passwd.
* In the Build Environment we select the **SSH Agent option**  - to do ssh private key authentication (to connect server with ssh key authentication).
* In Build we select the “**Execute shell”** option to execute sync of bash script to install required packages and etc and execute these script in client machine to ful-fillment of dotnet application requirements.

To check the username and client ip whether the passed parameters in the parameterized dynamic values are correct or not.

**echo "$uname@${hip}"**

To sync the script from jenkins server path to client path we use rsync command

**rsync -avzh /var/api-pre-suite.sh $uname@${hip}:/tmp/**

To do ssh authentication and execute the script which was synced from jenkins we use the following command

**ssh $uname@${hip} "bash /tmp/api-pre-suite.sh $passd"**

1. **Naukriapi\_client\_setup**

Here in this suite we pull the code from svn and deploy it in the jenkins server and later sync the code using code.sh script .

* In general we select “**This project is parameterized**” to pass the client ip as client.
* In Source Code Management we share svn credentials to pull the code.
* In Build we select the “**Execute shell”** option to execute the client\_ip,Client\_name and execute the following commands

hip=$(echo ${client} | cut -d'-' -f 2)

cname=$(echo ${client} | cut -d'-' -f 1)

uname=$(echo ${client} | cut -d'-' -f 3)

To do release and publish the code in the jenkins server itself

**sudo -S su - root -c "dotnet publish /var/lib/jenkins/jobs/naukriapi\_client\_setup/workspace/Naukri\_A -f net6.0 -c Release"**

By executing this below command we sync the code.sh file from jenkins to client server path.  
  
**rsync -avzh /var/code-sync.sh $uname@${hip}:/tmp/**

To exclude the .svn files and sync remaining files and .dll files to the client machine and execute the code.sh files.

**rsync -avzh --exclude='\*.svn' --no-perms -og --chown=root:root /var/lib/jenkins/jobs/naukriapi\_client\_setup/workspace/Naukri\_A/bin/Release/net6.0/publish/ $uname@${hip}:/tmp/naukri/**

**ssh $uname@${hip} "bash /tmp/code-sync.sh $pass"**

**c) naukriapi\_post\_setup**

In this we will execute the post setup script and complete new naukri client setup.

* In general we select “**This project is parameterized**” to pass the client ip as client.
* In Build we select the “**Execute shell”** option to execute the client\_ip,Client\_name and execute the following commands

**hip=$(echo ${client} | cut -d'-' -f 2)**

**cname=$(echo ${client} | cut -d'-' -f 1)**

**uname=$(echo ${client} | cut -d'-' -f 3)**

By executing this below command we sync the api-post-setup.sh file from jenkins to client server path.

**rsync -avzh /var/api-post-suite.sh $uname@${hip}:/tmp/**

TO execute the script we use below command

**ssh $uname@${hip} "bash /tmp/api-post-suite.sh $passd"**

**d) naukriapi\_patch\_updates**

We use these suite to deploy the latest and Patch updated code to deploy in client machines.

* In general we select “**This project is parameterized**” to pass the client ip as client.
* In Source Code Management we share svn credentials to pull the code.
* In Build we select the “**Execute shell”** option to execute the client\_ip,Client\_name and execute the following commands

hip=$(echo ${client} | cut -d'-' -f 2)

cname=$(echo ${client} | cut -d'-' -f 1)

uname=$(echo ${client} | cut -d'-' -f 3)

To do release and publish the code in the jenkins server itself

**sudo -S su - root -c "dotnet publish /var/lib/jenkins/jobs/naukriapi\_patch\_updates/workspace/Naukri\_A -f net6.0 -c Release"**

By executing this below command we sync the code.sh file from jenkins to client server path.  
  
**rsync -avzh /var/code-sync.sh $uname@${hip}:/tmp/**

To exclude the .svn files and sync remaining files and .dll files to the client machine and execute the code.sh files.

**rsync -avzh --exclude='\*.svn' --no-perms -og --chown=root:root /var/lib/jenkins/jobs/naukriapi\_patch\_updates/workspace/Naukri\_A/bin/Release/net6.0/publish/ $uname@${hip}:/tmp/naukri/**

**ssh $uname@${hip} "bash /tmp/patch-update.sh $pass"**

**2) Market\_place & shine**

In these jenkins suites you just need to pull the code from svn repository and release it in jenkins server and will sync the DLL files to client servers.

* In Source Code Management we share svn credentials to pull the code.
* In Build we select the “**Execute shell”** option to execute the below mentioned commands

By using these commands we publish the updated code itself in the jenkins server.

**dotnet publish /var/lib/jenkins/jobs/path/to/the\_project/workspace/projetc -f net6.0 -c Release**

Using the below command we will sync the DLL files to a specific path.

**rsync -avzh --exclude='\*.svn' --no-perms -og --chown=root:root /var/lib/jenkins/jobs/path/to/the/project/ /workspace/project/bin/Release/net6.0/publish/ root@client\_ip:/var/www/build/path/to/deploy/code/**

* Select execute shell script on remote host using ssh and fill the below fields  
    
  ssh site server\_ip

**Command : -**

service market\_place/shine restart

**3) Converting\_docs**

In these jenkins suites you just need to pull the code from the svn repository and then deploy the code in the client machine and publish there and itself.

* In Source Code Management we share svn credentials to pull the code.
* In Build we select the “**Execute shell”** option to execute the below mentioned commands