How to release Interne system from Jenkins.

Naming conventions on servers:-

* Interne-Ams
* Review(Angular)-Appraisal
* Recruito-Recruito

Interne System Folder structure:

\*All path refer from wwwroot

On staging-

Ftp site(staging and production):-

On dev server:-

Ftp site(dev server):-

Jobs of jenkins and their functions:-

1.[amsApiFetchBuildZip](http://localhost:8080/job/amsApiFetchBuildZip/):-this job fetch interne api code from git and build it and prepared the distribution package inside backup folder.

2.[amsApiTransferDeploy](http://localhost:8080/job/amsApiTransferDeploy/):- this job is taking distribution package from backup folder and transfer and deploy that package to chosen server.

3.[amsAppFetchBuildZip](http://localhost:8080/job/amsAppFetchBuildZip/):- this job fetch interne app code from git and build it and prepared the distribution package inside backup folder.

4.[amsAppTransferDeploy](http://localhost:8080/job/amsAppTransferDeploy/):- :- this job is taking distribution package from backup folder and transfer and deploy that package to chosen server.

5.[appraisalApiFetchBuildZip](http://localhost:8080/job/appraisalApiFetchBuildZip/):- this job fetch interne api code from git and build it and prepared the distribution package inside backup folder.

6.[appraisalApiTransferDeploy](http://localhost:8080/job/appraisalApiTransferDeploy/):- this job is taking distribution package from backup folder and transfer and deploy that package to chosen server.

7.[appraisalAppFetchBuildZip](http://localhost:8080/job/appraisalAppFetchBuildZip/):- this job fetch interne api code from git and build it and prepared the distribution package inside backup folder.

8.[appraisalAppTransferDeploy](http://localhost:8080/job/appraisalAppTransferDeploy/):- this job is taking distribution package from backup folder and transfer and deploy that package to chosen server.

9.[recruitoApiFetchBuildZip](http://localhost:8080/job/recruitoApiFetchBuildZip/):- this job fetch interne api code from git and build it and prepared the distribution package inside backup folder.

10.[recruitoApiTransferDeploy](http://localhost:8080/job/recruitoApiTransferDeploy/):- this job is taking distribution package from backup folder and transfer and deploy that package to chosen server.

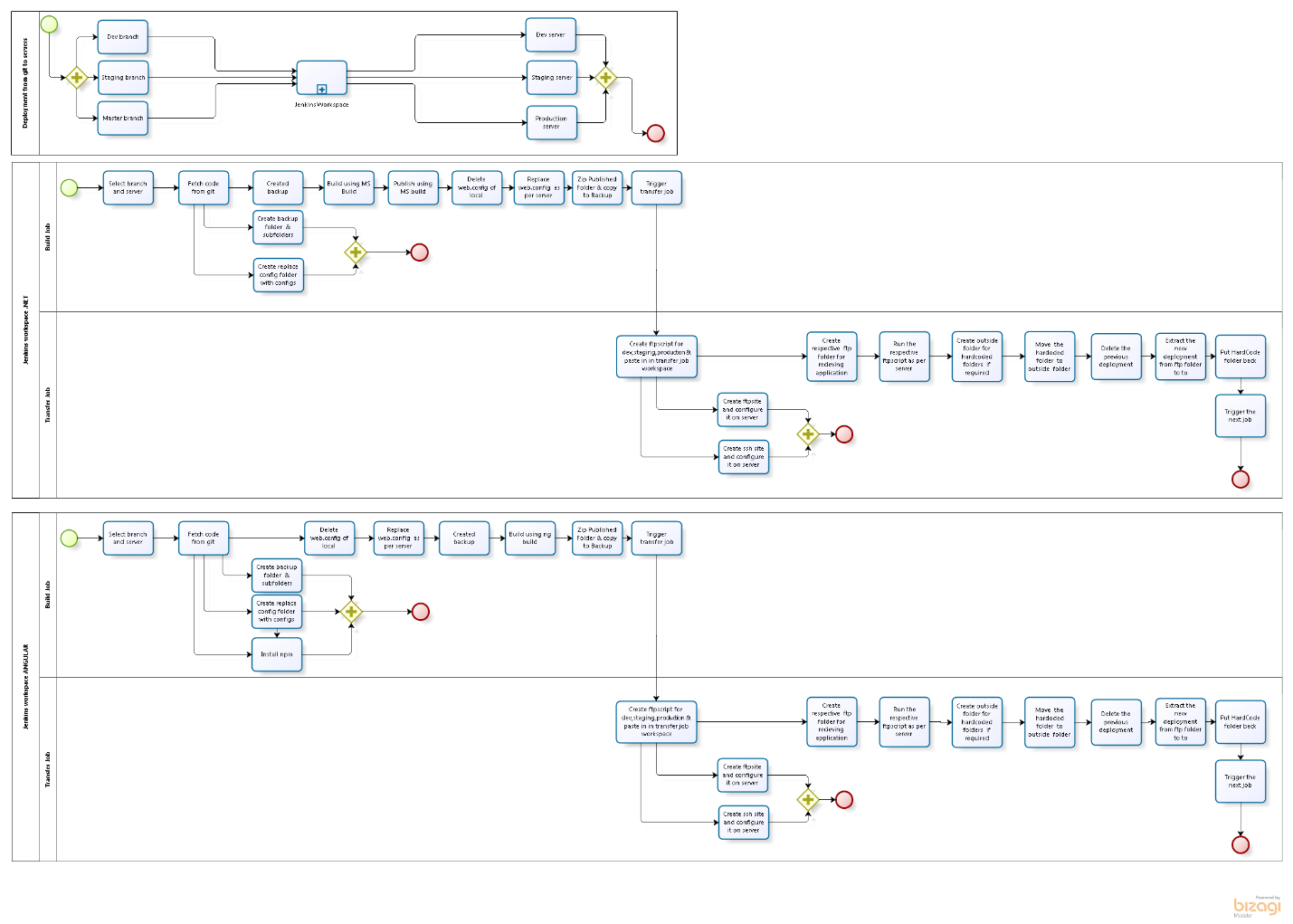
11.[recruitoAppFetchBuildZip](http://localhost:8080/job/recruitoAppFetchBuildZip/):- this job fetch interne api code from git and build it and prepared the distribution package inside backup folder.

12.[recruitoAppTransferDeploy](http://localhost:8080/job/recruitoAppTransferDeploy/):- this job is taking distribution package from backup folder and transfer and deploy that package to chosen server.

13. [InterneDbUpdate](http://localhost:8080/job/InterneDbUpdate/):-this is take a script from git folder as per chosen branch and then run the script with the name current.sql.

Important points to note:-

* All these jobs are connected means if your run first job then it will run the next one if successful and go on like this till last job except job no. 13(database update is not included in pipeline).for this function I also created one pipeline named as Interne Pipeline.
* If your build somehow not working and you want to deploy the last build then you need to put the last successful build from backup folder to one level up (from where transfer jobs picks up the distribution packages). Then you need to disconnect the build job and transfer jobs and then run transfer jobs independently.
* In order to disconnect the jobs just need to delete the name of next project at the bottom of the project inside Build Other Projects. You need to do this at every job in order to disconnect the chain.
* If you want to run a particular job then make sure that you will break forward connection and after completion of your task put the connection back as it was.
* If anyhow due to any error any job is failed then first you need to fix the error then run the job from “[Build with Parameters](http://localhost:8080/job/amsApiFetchBuildZip/build?delay=0sec)” and select the parameters specifically else it will take any parameter by default.



Steps to trigger full Interne system deployment:-

1. Go to chrome and type <http://localhost:8080/> ,this will open jenkin home page.
2. Click on InternePipeline tab.
3. Click on Run & choose Pipeline, server environment & git branch.
4. Click build & wait it will take some time to deploy, and you can refresh the page that will update the status of each job.

**Update(10-feb-2018):-**

* We have changed the folder structure on dev and staging server.
* We have modified the jobs also for pipeline control flow where from dropdown we can select the required option that wheather we want to build & deploy single modules or full Interne system.
* We cannot run a single job independentaly to build and deploy single app or api rather we can build and deploy modules or full system.
* Also git branch are hard coded for making this kind of flow means we have three branches like origin/dev, origin/staging & origin/master. So make sure create your branches in name (case sensitive)- dev,staging,master.
* In case we want to build and deploy separate branch other than this then we need to configure Jenkins accordingly like we need to disconnect those responsible jobs from already build pipeline then we need to give git branch name inside the repo option from where Jenkins fetch the code ($Git\_Branch instead of $Git).
* We have made New build and deployment for interne Console application (InterneCrons), there we build the directories on dev and staging server.
* We have made separate pipeline for this in Jenkins with new tab of interneCrons , while deploying this we need only to select git branch and deployment environment.
* Scheduling is not automated rather it will need to done manually as this is one time activity.