cd /e/Jenkins/workspace/iPUB\_APP\_Build\_Deployment\_Dev

rm -rf \*

git remote add origin http://inet:7990/scm/ipub/ipubapp.git

git init

git checkout dev

git submodule foreach git checkout dev

git clone http://inet:7990/scm/ipub/ipubapp.git --recurse -submodules

cd /e/Jenkins/workspace/iPUB\_APP\_Build\_Deployment\_Dev/ipubapp

echo "\*\*\*\*\*\*\*\*\*\*\*\*\*\*Install&Build\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

npm install && npm run build --production

echo "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Fineshed Build\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

cd /e/Jenkins/workspace/iAuthor\_Deploy\_Demo

REPO='/e/Jenkins/workspace/iAuthor\_Deploy\_Demo'

if [ ! -d ${REPO}/.git ]; then

echo "${REPO} is not a valid git repo! Aborting..."

exit 0

else

echo "${REPO} is a valid git repo! Proceeding..."

fi

cd /e/Jenkins/workspace/iAuthor\_Deploy\_Demo

git remote add origin http://is-srv008:7990/scm/iaut/live-iauthor.git

git checkout release/iAuthorDemo

git pull

gulp --gulpfile E:/Scripts/iAuthor\_Install/iAuthor\_demo/gulpfile.js defaulteditorjs

gulp --gulpfile E:/Scripts/iAuthor\_Install/iAuthor\_demo/gulpfile.js defaultcss

gulp --gulpfile E:/Scripts/iAuthor\_Install/iAuthor\_demo/gulpfile.js htmlminify

node E:/Scripts/iAuthor\_Install/iAuthor\_demo/versionchange.js

echo "========================= Checking status for Bitbucket git start ========================="

git status

echo "========================= Checking status for Bitbucket git end ========================="

echo "========================= Adding all changes for Bitbucket git start ========================="

git add .

echo "========================= Adding all changes for Bitbucket git end ========================="

echo "========================= Committing changes into Bitbucket Git start========================="

git commit -m "Automatic Git commit checking from Jenkkins"

echo "========================= Committing changes into Bitbucket Git end ========================="

git push

echo "========================= Pushing changes into Bitbucket Git start ========================="

echo "========================= Pushing changes into Bitbucket Git end ========================="

PFE-D

Sh gulp clean && gulp build

E:/Scripts/iPMP/iPMP\_Deploy\_Dev.sh

#!/bin/sh

cd /e/Jenkins/workspace/iPMP\_DEV

rm -rf \*

rm -rf .git

mkdir webapi

cd /e/Jenkins/workspace/iPMP\_Build\_Deploy\_FE\_Dev/dist

rm -rf app/

cp -r \* /e/Jenkins/workspace/iPMP\_DEV

cd /e/Jenkins/workspace/iPMP\_Build\_Deploy\_FE\_Dev

cp -r Web.config /e/Jenkins/workspace/iPMP\_DEV

cd /e/Jenkins/workspace/iPMP\_Build\_Deploy\_BE\_Dev/outproject/\_PublishedWebsites/iPMS.Web

cp -r \* /e/Jenkins/workspace/iPMP\_DEV/webapi

cd /e/Jenkins/workspace/iPMP\_DEV/

git init

echo "========================= Checking status for Azure git start ========================="

#git remote set-url origin https://integra-ipmp-dev-nsp2.scm.azurewebsites.net:443/integra-ipmp-dev-nsp2.git

git remote add origin https://integra-ipmp-dev-nsp2.scm.azurewebsites.net:443/integra-ipmp-dev-nsp2.git

git init

git status

echo "========================= Checking status for Azure git end ========================="

echo "========================= Adding all changes for Azure git start ========================="

git add .

echo "========================= Adding all changes for Azure git end ========================="

echo "========================= Committing changes into Azure Git start========================="

git commit -m "Automatic Git commit checking from Jenkkins"

echo "========================= Committing changes into Azure Git end ========================="

git push -f origin master

echo "========================= Pushing changes into Azure Git end ========================="

**PBE-D:**

Msbuild

$workspace\iPMS.sln

/t:Rebuild /p:outdir="E:\Jenkins\workspace\iPMP\_Build\_Deploy\_BE\_Dev\outproject\\" /p:Configuration=Release /p:Platform="Any CPU"

E:/Scripts/iPMP/iPMP\_Deploy\_Dev.sh

clean install

#!groovy

node('node') {

currentBuild.result = "SUCCESS"

try {

stage('Checkout'){

checkout scm

}

stage('Test'){

env.NODE\_ENV = "test"

print "Environment will be : ${env.NODE\_ENV}"

}

stage('Node Version'){

sh 'node -v'

}

stage('Node clean'){

sh 'npm clean'

}

stage('Package Instalation'){

sh 'npm install'

}

stage('Build'){

sh 'npm build'

}

stage('Deploy'){

echo 'Push to Repo'

sh 'copy the file to azure repo'

echo 'ssh to web server and tell it to pull new image'

sh 'ssh deploy@xxxxx.xxxxx.com running/xxxxxxx/azure.sh'

}

stage('Cleanup'){

echo 'prune and cleanup'

sh 'npm prune'

sh 'rm node\_modules -rf'

mail body: 'project build successful',

from: 'xxxx@yyyyy.com',

replyTo: 'xxxx@yyyy.com',

subject: 'project build successful',

to: 'yyyyy@yyyy.com'

}

}

catch (err) {

currentBuild.result = "FAILURE"

mail body: "project build error is here: ${env.BUILD\_URL}" ,

from: 'xxxx@yyyy.com',

replyTo: 'yyyy@yyyy.com',

subject: 'project build failed',

to: 'zzzz@yyyyy.com'

throw err

}

}

pipeline {

agent any

stages {

stage('first') {

steps {

echo 'first, non-parallel stage'

}

}

stage('top-parallel') {

stages {

stage('first-parallel') {

steps {

echo 'First of the parallel stages without further nesting'

sleep 60

}

}

stage('second-parallel') {

stages {

stage('first-nested-parallel') {

steps {

echo 'the first of the nested parallel stages'

sleep 30

}

}

stage('second-nested-parallel') {

steps {

echo 'the second of the nested parallel stages'

sleep 30

}

}

}

}

}

}

node {

currentBuild.result = "SUCCESS"

try {

stage('SCM Chekout'){

git branch: 'test', credentialsId: 'iPMP', url: 'http://172.44.10.22:7990/scm/ipmp/ipmp-front-end.git'

}

stage('Gulp Clean') {

// Clean the Gulp modules.

bat label: '', script: 'gulp clean'

}

stage('Build') {

// Building the code.

bat label: '', script: 'gulp build'

}

stage('Deployment on Azure') {

// Deploying the code into Azure Webapp.

sh "hostname"

mail body: "Project build successful: ${env.BUILD\_URL}",

from: 'samsani.madusudhana@integra.co.in',

replyTo: 'samsani.madusudhana@integra.co.in',

subject: 'project build successful',

to: 'samsani.madusudhana@integra.co.in'

}

}

catch (err) {

currentBuild.result = "FAILURE"

mail body: "Project build error is here: ${env.BUILD\_URL}" ,

from: 'samsani.madusudhana@integra.co.in',

replyTo: 'samsani.madusudhana@integra.co.in',

subject: 'project build failed',

to: 'samsani.madusudhana@integra.co.in'

throw err

}

}

Deployment.file

cd /e/Jenkins/workspace/iPub\_API\_Build\_Deploy\_Test

rm -rf .git

git init

git remote add origin http://172.16.200.141:7990/scm/ipubs/ipubapi.git

#Get latest from Dev Bitbucket

git pull

git checkout TEST -f

git pull

# Auto-Versioning

node E:/Scripts/iPUB/versionchange-TEST-API.js

PACKAGE\_VERSION=$(cat /e/Jenkins/workspace/iPub\_API\_Build\_Deploy\_Test/package.json | grep version | head -1 | awk -F= "{ print $2 }" | sed 's/[version:,\",]//g' | tr -d '[[:space:]]')

#echo $PACKAGE\_VERSION

git add -A

git commit -m "$PACKAGE\_VERSION"

git push -f origin TEST

git pull

#Tagging to Bitbucket

git tag $PACKAGE\_VERSION

git checkout tags/$PACKAGE\_VERSION

git push origin $PACKAGE\_VERSION

#&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&

cd /e/Jenkins/workspace/iPub\_API\_Build\_Deploy\_Test

rm -rf .git

git init

REPO='/e/Jenkins/workspace/iPub\_API\_Build\_Deploy\_Test'

if [ ! -d ${REPO}/.git ]; then

echo "${REPO} is not a valid git repo! Aborting..."

exit 0

else

echo "${REPO} is a valid git repo! Proceeding..."

fi

echo "========================= Checking status for Azure git start ========================="

#git remote set-url origin https://ipubsuitetestapigitusr@integra-ipubsuite-testapi.scm.azurewebsites.net:443/integra-ipubsuite.git

git remote add origin https://ipubsuitetestapigitusr@integra-ipubsuite-testapi.scm.azurewebsites.net:443/integra-ipubsuite.git

git status

echo "========================= Checking status for Azure git end ========================="

echo "========================= Adding all changes for Azure git start ========================="

git add .

echo "========================= Adding all changes for Azure git end ========================="

echo "========================= Committing changes into Azure Git start========================="

git commit -m "Automatic Git commit checking from Jenkkins"

echo "========================= Committing changes into Azure Git end ========================="

git push -f origin master

echo "========================= Pushing changes into Azure Git end ========================="

**versioncontol.file**

const fs = require('fs')

function getversion(param) {

let version = param;

let today = new Date();

let dd = String(today.getDate()).padStart(2, '0');

let mm = String(today.getMonth() + 1).padStart(2, '0'); //January is 01

let yyyy = today.getFullYear();

version = version.split('.');

(version[0].startsWith("V"))?version[0]=version[0].slice(1):version[0];

//(version[0].startsWith("v"))?version[0]=version[0].slice(1):null;

if(Number(version[0])==Number(yyyy) && Number(version[1])==Number(mm) && Number(version[2])==Number(dd)){

version[3]=Number(version[3])+1;

}

else{

version[0]=yyyy;

version[1]=mm;

version[2]=dd;

version[3]=01;

}

return version.join('.').toString();

}

let file\_content = fs.readFileSync("E:/Jenkins/workspace/iPub\_API\_Build\_Deploy\_Test/package.json");

let content = JSON.parse(file\_content);

let val1 = content.version;

val1=("V"+ getversion(val1) +"");

content.version=val1;

content = JSON.stringify(content, null, 2);

fs.writeFileSync('E:/Jenkins/workspace/iPub\_API\_Build\_Deploy\_Test/package.json',content);

process.stdout.write(val1);

return val1;

**package.json file:**

**{**

**"name": "ipubsuite\_api",**

**"description": "ipubsuite api portal handles user like irights, IOPP, T&F and S&W.",**

**"version": "V2020.03.10.1",**

**"environment": "test",**

**"private": false,**

**"main": "server.js",**

**"scripts": {**

**"start": "node server.js",**

**"post": "node postgrator"**

**},**

**"repository": {**

**"type": "git",**

**"url": "test"**

**},**

**"keywords": [**

**"ipubsuite",**

**"api",**

**"portal",**

**"irights"**

**],**

**"dependencies": {**

**"azure": "^2.3.1-preview",**

**"azure-common": "^0.9.20",**

**"azure-storage": "^2.10.1",**

**"body-parser": "^1.18.3",**

**"chalk": "^2.4.1",**

**"compression": "^1.7.3",**

**"cors": "^2.8.4",**

**"cron": "^1.7.2",**

**"cryptr": "^4.0.2",**

**"docx-parser": "^0.2.1",**

**"events": "^3.0.0",**

**"express": "^4.16.3",**

**"express-fileupload": "^0.4.0",**

**"express-joi-validator": "^2.0.1",**

**"express-jwt": "^5.3.1",**

**"express-validator": "^5.3.0",**

**"fetch": "^1.1.0",**

**"file-stream-rotator": "^0.4.1",**

**"fs": "0.0.1-security",**

**"glob": "^7.1.3",**

**"helmet": "^3.15.0",**

**"http-status-codes": "^1.3.0",**

**"iplocation": "^6.1.0",**

**"is-doc": "^1.0.0",**

**"is-docx": "0.0.3",**

**"joi": "^14.3.1",**

**"jsdom": "^13.0.0",**

**"json2xls": "^0.1.2",**

**"jsonwebtoken": "^8.3.0",**

**"jszip": "^3.1.5",**

**"lodash": "^4.17.11",**

**"method-override": "^3.0.0",**

**"morgan": "^1.9.0",**

**"mssql": "^4.2.1",**

**"multer": "^1.3.1",**

**"nodemailer": "^4.6.7",**

**"nodemon": "^1.18.3",**

**"npm": "^6.4.0",**

**"path": "^0.12.7",**

**"pg": "^7.8.0",**

**"postgrator": "^3.7.0",**

**"request": "^2.88.0",**

**"socket.io-client": "^2.2.0",**

**"xml-js": "^1.6.11",**

**"xmlhttprequest": "^1.8.0"**

**},**

**"engines": {**

**"node": "10.6.0",**

**"npm": "6.1.0"**

**},**

**"author": "integra",**

**"license": "ISC",**

**"devDependencies": {**

**"postgrator-cli": "^3.1.0"**

**}**

**}**