International Journal of Current Science Research and Review
ISSN: 2581-8341
Volume 07 Issue 10 October 2024
DOI: 10.47191/ijcsrr/V7-i10-28, Impact Factor: 7.943
IJCSRR @ 2024
www.ijcsrr.org

Strategic plan for OmniScripts Deployments and Activation in Vlocity Packages

Rishitha Kokku
Senior Software Engineer, Optum Services INC

ABSTRACT: Vlocity Package deployment involves careful consideration of core concepts, OmniScripts, Vlocity architecture and a clear plan to the CICD pipelines and workflows. Vlocity packages have many essential tools to offer for business processes and one critically acclaimed concept is the OmniScripts and OmniStudio. DataRaptors within the OmniStudio let companies handle the data in an unique way. The deployment and activation of the OmniScripts would require a curated pipeline as there are various factors to consider. CICD pipelines for OmniScripts throws many challenges due to its complex architecture. This paper highlights the challenges and also a strategic plan to efficiently manage the OmniScript deployments.

KEYWORDS: Vlocity, DataRaptors, OmniScripts, Jenkins, CICD pipelines, DataPacks, Salesforce CLI, Vlocity deployments, Integration Procedures.

1. INTRODUCTION

OmniScripts have well-defined reusable elements that eases developer efforts to satisfy business needs. Vlocity is a package built by Salesforce to offer specific solutions to organizations especially in healthcare, insurance, security and communications. Vlocity package key components are DataRaptors, OmniScripts and Integrations. OmniScripts help developers and business teams to create guided interactions for their customers. This paper concentrates on how the OmniScripts are deployed using Jenkins Pipelines and how they are activated in an automated process.

2. CHALLENGES WITH DEPLOYING OMNISCRIPTS

2.1 Complex Dependencies: The deployment should ensure that the DataRaptors and other Integration Procedures are deployed before the activation of the OmniScripts. The reason for this is because of the dependencies OmniScripts have on the above mentioned key components. The sequential order of the deployment should be carefully handled to avoid any errors.

2.2 Data Inconsistencies: Apart from the data coming from the Vlocity packages, there resides transactional and reference data within the environments. It is possible that this data might interact with multiple versions of data created by OmniScripts leading to conflicts.

2.3 Multi Step Deployment: Unlike other metadata, OmniScript deployment and activation is a two step process that needs to be handled in the pipeline. There is also a high need for a rollback process established with OmniScripts. DataPacks from the previous versions should be planned to deploy incase of failures and outage.

2.4 Version Conflicts: Often than not, OmniScripts are modified by multiple users in the Salesforce environments leading to conflicts and differences in their versions. This behavior leads to irregularities in the workflows.

3. PIPELINE STRATEGY

3.1 Vlocity Build tool: Vlocity build tool is widely used to deploy OmniScripts. The integration between source file (likely version control) and destination environment, handling dependencies and resolving the errors is controlled by this tool. The entire Datapacks are retrieved and deployed in an automated approach using the Vlocity Build Tool. DataPacks can also be retrieved, sorted by their dependencies and deployed to the target environment using the Vlocity Build Tool.

3.2 CICD pipeline using Jenkins and GitHub: Continuous Integration and Continuous Deployment is essential for OmniScript because of the challenges striked with managing them manually. There are multiple steps involved in building this pipeline using

International Journal of Current Science Research and Review
ISSN: 2581-8341
Volume 07 Issue 10 October 2024
DOI: 10.47191/ijcsrr/V7-i10-28, Impact Factor: 7.943
IJCSRR @ 2024
www.ijcsrr.org

Jenkins which will be discussed in later sections. Salesforce CLI commands can be introduced in the Jenkins scripts to deploy the OmniScripts.

3.3 Version Control: It is critical to leverage a version control tool for OmniScripts to track the versions and maintain their sanity. A well defined branching strategy will help deploy the right version of OmniScripts to the destination environments.

3.4 OmniScript Activation: Although this can be done manually post deployment, it is advisable to include the activation process in the pipeline to ensure a seamless and smooth deployment. There would be plenty of human intervention if the OmniScripts are to be manually activated in multiple sandboxes and Production after each migration.

3.5 Automated Test Case Execution: Automated testing is vital in verifying the functionality of OmniScripts. This involves unit testing by the developers and regression testing. All of the testing modules can be included in the CICD pipeline to receive faster testing results and improve the quality of the application.

3.6 Monitoring activities: An end-to-end pipeline will have monitoring and alerting mechanisms alongside the adobe mentioned blocks. It is essential to have alerts received when there is a disruption of service or deployment failures.

4. BUILD THE JENKINS PIPELINE

Developing a CICD pipeline using Jenkins requires integration with the Vlocity Build tool. The script uses Salesforce CLI commands to connect with Salesforce and the Jenkins configuration should be handled prior to deploying the OmniScripts. The version control system (Git) should comprise the right version of the DataPacks and a few other configuration files for a successful deployment.

- **Install necessary Jenkins Plugins:** Jenkins requires GIT, NodeJs, Pipeline, Credentials Plugins to retrieve and deploy the OmnScripts to Salesforce orgs.
- **Create JWT Connected App in each envrionment:** Generate a Self-Signed certificate in order to use it in the Connected App for Authentication. Below command generates a certificate.

  ```
  openssl req -new -newkey rsa:2048 -nodes -keyout server.key -x509 -days 365 -out server.crt
  ```

  Start creating the Connected App and provide all the required details and upload the above certificate. Test the connection in a CLI using the below command. If the connection is successful, the same JWT can be used in the Jenkinsfile for JWT authentication.

  ```
  sfdx force:auth:jwt:grant --clientid “Your_clientid_here” --jwtkeyfile “location_to_your_server.key” --username “username” --instanceurl https://login.salesforce.com --setdefaultdevhubusername
  ```

- **Jenkins Credentials:** Use the Credential Plugin to store the Salesforce Credentials in Jenkins. Use the Client ID and Client Secret from the Connected App created above and store them as entries in Jenkins.
- **Jenkins Job:** Create a pipeline job in Jenkins and provide an appropriate name. Configure the job and provide essential details like version control URL (verify the connection), name of the Jenkinsfile to build the pipeline, NodeJS configuration (this has to be configured with the right version under “Manage Jenkins”).
- **Develop the Jenkinsfile:** Jenkinsfile sits in the root directory of the repository and drives the Jenkins job. There are multiple stages involved in a pipeline and these stages are defined in Jenkinsfile.

International Journal of Current Science Research and Review
ISSN: 2581-8341
Volume 07 Issue 10 October 2024
DOI: 10.47191/ijcsrr/V7-i10-28, Impact Factor: 7.943
IJCSRR @ 2024
www.ijcsrr.org

```
pipeline {
    agent ('label')

    environment {

        SF_USERNAME = env.salesforce-username
        SF_CLIENTID = env.salesforce-client-id
        SF_CLIENT_SECRET = env.salesforce-client-secret
        SF_SERVER_URL = env.serverurl
        SF_KEY = credentials('salesforce-jwt-key')
    }

    tools {
        nodejs 'NodeJS-13'
    }

    stages {
        stage('Checkout SCM) {
            steps {
                checkout scm
            }
        }

        stage('Install VBT') {
            steps {
                sh 'npm install -g vlocity'
            }
        }

        stage('Authenticate with Salesforce') {
            steps {
                sh '''
                    sfdx auth:jwt:grant --clientid $SF_CLIENTID \
--jwtkeyfile $SF_KEY \
                    --username $SF_USERNAME \
                    --instanceurl $SF_SERVER_URL \
                    --setalias "givealiasname"
            '''
            }
        }
stage("Vlocity Installation"){
        Steps{
sh label: '', script: 'node -v'
                    sh label:'', script: 'sudo npm install -g n'
                    sh label:'', script: 'sudo n v16.16.0'
                    sh label: '', script: 'sudo n use v16.16.0'
                    sh label: '', script: 'node -v'
                    sh label: '', script: 'npm -v'
                    sh label: '', script: 'sudo mv /usr/local/bin/node /usr/bin'
                    sh label: '', script: 'sudo npm install --global vlocity'
        }
    }
    stage('Deploy Vlocity Package') {
        steps {
            sh '''
                vlocity -sf.username $SF_USERNAME \
                    -sf.password $SF_PASSWORD \
                    packDeploy -projectPath ./vlocity
            '''
        }
    }

    stage('Post-Deployment Tasks') {
        steps {
      }
    }
  }
```

Figure 1. Jenkins file with all the stages involved in the pipeline

International Journal of Current Science Research and Review
ISSN: 2581-8341
Volume 07 Issue 10 October 2024
DOI: 10.47191/ijcsrr/V7-i10-28, Impact Factor: 7.943
IJCSRR @ 2024
www.ijcsrr.org

- **Test the Pipeline:** Run the Jenkins job and monitor the stages mentioned in Figure 1. If there is a failure, it is likely that the connection or configuration is not done in the right way. Debug the error and run the pipeline until it has been successfully executed.

- **Automated Pipeline:** Polling can be enabled for the job to eliminate the manual intervention and deploy when the changes are merged into the repository. This can be enabled inside the build configuration.

5. BEST PRACTICES FOR OMNISCRIPTS DEPLOYMENT MANAGEMENT

5.1 Security Monitoring: There is confidential data involved with OmniScripts and an unsecure deployments can pose a threat to this data. Ensure proper security standards are in place while migrating the OmniScripts between environments. Encryption and secure Authentication is necessary as per the organizational policies and is highly recommended by Salesforce.

5.2 Flexible OmniScripts: Divide larger OmniScripts into chunks for better handling and better deployments. Larger OmniScripts often tend to fail during deployments and can become complicated to maintain in multiple environments.

5.3 Consistency: Ensure the OmniScripts are consistent across the sandboxes and in Production and are matching with version control. Impose strict guidelines to avoid manual changes in sandboxes especially in QA, UAT and Production.

6. CONCLUSION

With a well versed CICD mechanism, OmniScripts can be managed and successfully deployed despite its complexities and architecture. Deployment strategy, version control system, branching strategy and automation are the driving factors of a successful DevOps practice especially when OmnScripts are involved in Salesforce.

REFERENCES

1. Simge Ulusoy, Alper Batıoğlu, Tolga Ovatman, Omni-script: Device independent user interface development for omni-channel fintech applications, Computer Standards & Interfaces, Volume 64, 2019, Pages 106-116, ISSN 0920-5489, https://doi.org/10.1016/j.csi.2019.01.003.
2. P. McCollum, *Practical Salesforce Architecture*, Understanding and Deploying the Salesforce Ecosystem for the Enterprise, O'Reilly Media, Inc., 2019
3. D. Khanine, *Optimizing Salesforce Industries Solutions on the Vlocity OmniStudio Platform*, Implementing OmniStudio best practices for achieving maximum performance, Packt Publishing, 2024.
4. A. Davis, *Mastering Salesforce DevOps*, Apress, 2019.
5. Salesforce.com, Troubleshoot Migration. [Online]. Available: https://help.salesforce.com/s/articleView?id=ind.comms_troubleshoot_migration.htm&type=5
6. Salesforce.com, OmniStudio Spring '21. [Online]. Available: https://help.salesforce.com/s/articleView?id=ind.comms_t_omnistudio_enhancementsspring21_16186.htm&type=5
7. Salesforce.com, OmniScripts in Multiplay Subscription Management. [Online]. Available: https://help.salesforce.com/s/articleView?id=ind.comms_t_omniscripts_in_mobile_subscriptionmanagement_79725.htm&type=5

Cite this Article: K. Rishitha (2024). Strategic plan for OmniScripts Deployments and Activation in Vlocity Packages. International Journal of Current Science Research and Review, 7(10), 7741-7744, DOI: https://doi.org/10.47191/ijcsrr/V7-i10-28