International Journal of Science and Research (IJSR)
ISSN: 2319-7064
Impact Factor 2024: 7.101

CA Lisa - CI CD CT Micro Service Test Implementation Guide

Project : CI CD CT Automation-Project

Ram Aurovind

University of Madras

Abstract: ***The CI CD CT Automation-Project reflects a timely and well-structured response to a persistent challenge within the international shipping ecosystem delivering a seamless, transparent, and intelligent customer experience across a fragmented logistics chain. What stands out is its use of cutting-edge technologies like Big Data, Microservices, and Machine Learning not merely as buzzwords, but as working components that power real-time classification of commodities, automate customs estimation, and streamline transit projections. This isn't just about technical efficiency it's about rebuilding trust in data accuracy and turnaround times for shipping clients, who often feel lost in outdated systems. That said, the integration of smoke and regression testing within a Jenkins-based CI/CD pipeline adds a much-needed layer of resilience, ensuring that deployments across development, QA, staging, and production environments remain error-free and predictable. It is evident that this project is not only a technical upgrade but a strategic step toward reshaping how global trade tools interact with complex regulatory frameworks. The consistent auto-triggering, environment-specific validations, and intelligent automation mechanisms present a solid blueprint for future digital transformation initiatives in logistics tech. From my perspective, this initiative paves the way for other sectors to rethink automation not as a support function—but as a driver of customer satisfaction and operational clarity.***

Keywords: commodity classification, big data logistics, CI/CD automation, shipping analytics, customer experience in trade

1. Introduction

SHIPPING_SECTORS Trade Tools team began a groundbreaking new initiative named the CI CD CT Automation-Project reflects a timely and well-structured response to a persistent challenge within the international shipping ecosystem delivering a seamless, transparent, and intelligent customer experience across a fragmented logistics chain.

What stands out is its use of cutting-edge technologies like Big Data, Microservices, and Machine Learning not merely as buzzwords, but as working components that power real-time classification of commodities, automate customs estimation, and streamline transit projection Any Operating company looking for accurate commodity classification and accurate duties, tax and transit and also other non-clearance and brokerage platforms can leverage Trade Tools intelligent micro services

This isn't just about technical efficiency it's about rebuilding trust in data accuracy and turnaround times for shipping clients, who often feel lost in outdated systems. That said, the integration of smoke and regression testing within a Jenkins-based CI/CD pipeline adds a much-needed layer of resilience, ensuring that deployments across development, QA, Staging, and Production environments remain error-free and predictable.

It is evident that this project is not only a technical upgrade but a strategic step toward reshaping how global trade tools interact with complex regulatory frameworks. The consistent auto-triggering, environment-specific validations, and intelligent automation mechanisms present a solid blueprint for future digital transformation initiatives in logistics tech. From my perspective, this initiative paves the way for other sectors to rethink automation not as a support function—but as a driver of customer satisfaction and operational clarity

2. Problem Summary

Any Major retail or Shipping sector is having trouble in bringing shipping analytics, customer experience in trade For getting accurate dates for the shipping information and estimated due dates or delivery delay in a Single window.

3. Objective

CI CD CT Automation-Project involves innovative idea of improving the customers experience by providing accurate product code, estimated duties, estimated taxes, transit time, and digitalized experience by utilizing the cutting-edge technologies like Big Data and Analytics.

CI CD CT Automation-Project is to build an innovative, intelligent platform that classifies commodities to a fully qualified code with a single click.

CI CD CT Automation-Project is responsible for ingesting data from various data sources into the Big Data Lake, preparing Data engineering pipelines and machine learning models to create datasets which will be used to render data to Business services

4. Document Overview

In **CI CD CT Automation-Project,** we are continuously developing the application and deploying the code in different environment.

Paper ID: SR25623221815 DOI: https://dx.doi.org/10.21275/SR25623221815

We have designed Smoke Test to make sure that all the components of **CI CD CT Automation-Project** is up and running in all the environments.

The Smoke Test is integrated with build deploy process. Whenever the build is deployed successfully, smoke test is triggered and upon successful execution, an email report generated and sent to all the stakeholders.

Assumptions and Constraints

Topic	Description
None	Not Captured any Error

Core Competencies

Skill	Description
Automation	CA Dev Test used for automating the test cases.

Environment set-up

The following environment setup is required:

- Installing latest version of Java on the LINUX box: As the CADevTest requires JRE to run some of the script, hence Java was installed and environment variables configured. Java version should be 1.7 or higher.
- Installation of MS office
- Installation of CADevTest workstation on build machine to execute the automated test. For getting the CADevTest installer file we need to contact GTM_CLM team (gtm_ca@corp.ds.SHIPPING_SECTORS.com). Below are the steps to install CADevTest Workstation on (LINUX: http://c0008666.test.cloud.SHIPPING_SECTORS.com) & windows (Windows (http://uwn04530.corp.ds.SHIPPING_SECTORS.com, http://uwn04531.corp.ds.SHIPPING_SECTORS.com)) build machine:

Process for Environment set-up

The following process for environment setup is required:

1) In a terminal window, navigate to the directory where the installer file is located.
2) Ensure that the installer file has the execute permission. chmod 777 devtest_platform_x64.sh This gives rwxrwxrwx permissions on the file.
3) Run the installer file. Double-click the icon or enter a command similar to the following command from a terminal window: ./devtest_platform_x64.sh The DevTest Solutions Setup wizard opens.
4) Click Next. The CA End User License Agreement step opens.
5) Read the license agreement, select the I accept the terms of the License Agreement check box, and click Next. The Select Destination Directory step opens.
6) Specify the directory where you want to install DevTest Workstation. Do not use a directory that contains spaces. (The default is /opt/CA/DevTest.)
7) Click Next. The Installation Type step opens.
8) Select one of the following options and click Next.

Local

Installs all DevTest Solutions components into a single directory on the local computer. By default, all data is stored in this directory, and each user has a personal temp directory. Local is the most common installation type that is used in most environments.

Shared

Used by administrators to install all of the DevTest Solutions components to a shared location that multiple users from multiple computers can access. All data and temporary files are stored in user-specified directories. Each user has personal data, but they share a common DevTest Solutions installation. With a shared installation, users only need read access to the DevTest Solutions programs directory.

The Select Components step opens.

1) Clear the Server check box, ensure that the Workstation check box is selected, and click Next. If you chose the shared installation type, the following steps prompt you to specify the data directory and the temporary files directory.
2) Specify the directories, clicking Next after each step. The Specify Demo Server step opens.
3) If you want the installer to unzip the demo server into the LISA_HOME directory, select the Install demo server check box and specify the fully qualified path of the demo server zip file.
4) Click Next. The Select Additional Tasks step opens.
5) If you do not want to create a desktop icon for DevTest, clear the check box.
6) Click Install. When the installation is finished, the Information step opens.
7) Read the information and click Next. The Completing the DevTest Solutions Setup Wizard step opens.
8) Click Finish.

- All the necessary jars to be kept in LISA lib directory whose location is /opt/SHIPPING_SECTORS/tibco/CADevTest/lib/
- Creation of a CADevTest user for executing the Trunner command. A dummy user is created for this by requesting to the SHIPPING_SECTORS manger concerned.
- Firewall to be configured to allow access to CADevTest registry. Request InfoSec team to allow CADevtset to access lisa registry server from the build box
- Connection to all the component

Smoke Test/ Sanity Test Implementation

Scenarios covered in Smoke Test

Copy all the Smoke test scenarios in below folder

LINUX : http://c0008666.test.cloud.SHIPPING_SECTORS.com **/opt/SHIPPING_SECTORS/tradetools/LISA/lisadevtest/ Projects/CI CD CT AUTOMATION-PROJECT_DEMO/Tests/**

Create Test Runner command for all test suite:

```
Ex:./TestRunner -a -u 3592564 -p May20198 -t /opt/SHIPPING_SECTORS/tradetools/LISA/lisadevtest/Projects/CI CD CT AUTOMATION-PROJECT_DEMO/Tests/All_Services_new_framework.tst -m ssl://srh00457.ute.SHIPPING_SECTORS.com:2010/Registry
```

Automation-Pipeline Summary

1) Parametrized build will help customizing the build and deployment process with minimum code
2) Single Job Triggered across Multiple Environments in 3 levels (DEV-1, DEV-2, QA) in a single stage, STAGING, PROD in 2 different stages as shown below
3) Application components under HSValidation in git lab, which was created as a separate pipeline has
4) been merged into a single pipeline where in it will execute the code based upon the selection of
5) Application component parameter
6) please find below
7) Pipeline code has been decreased more than 60% lines of code as compared to the existing code
8) Stages are rather executed based on the condition/parameter
9) We can use Jenkins shared library which will give more benefits across the projects, Single shared
10) library will be access all the projects.
11) Through Shared library Jenkins file/Pipeline code will decrease a lot, only reference of the shared
12) library will be given in pipeline code

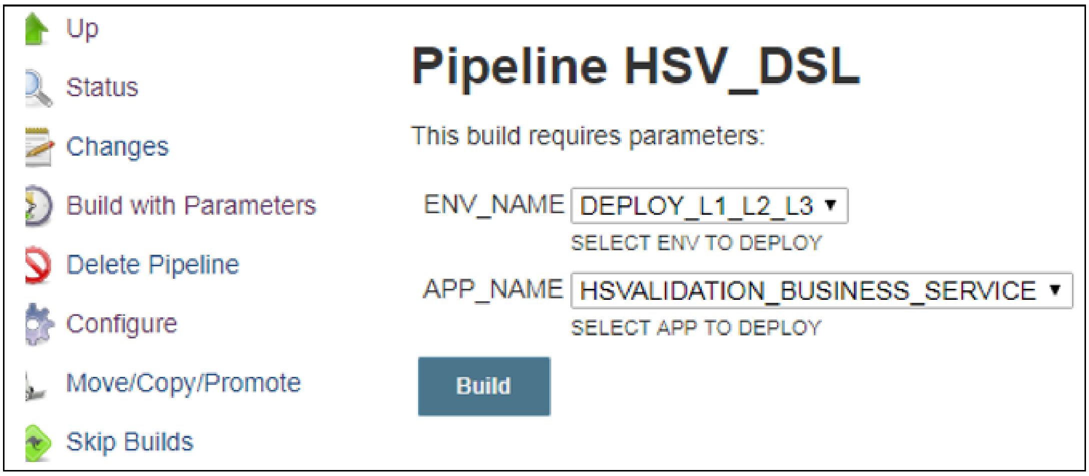
Automation of the smoke test scenarios

CI - CD Team will create pipeline in Jenkins to Trigger Smoke Test suites across all the levels

https://jenkins.prod.cloud.SHIPPING_SECTORS.com:8443/jenkins/job/CI CD CT AUTOMATION-PROJECT/job/MicroServices/job/Scripted_Microservice/
https://jenkins.prod.cloud.SHIPPING_SECTORS.com:8443/jenkins/job/CI CD CT AUTOMATION-PROJECT/job/MicroServices/job/Scripted_Microservice/job/HSV_DSL/

- DEV-1,DEV-2,QA
- L4
- L6

Pipeline structure:



Pipeline HSV_DSL

This build requires parameters:

ENV_NAME DEPLOY_L1_L2_L3
SELECT ENV TO DEPLOY

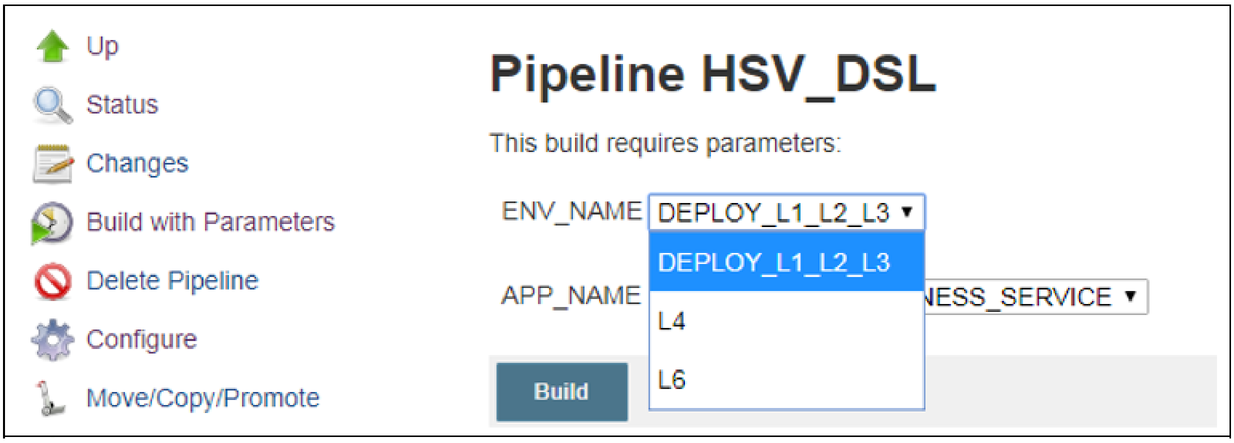
APP_NAME HSVALIDATION_BUSINESS_SERVICE
SELECT APP TO DEPLOY

Execution of Smoke Test

Auto Triggering Smoke test along with Build

- DEV-1, DEV-2, QA (Single deployment and Load master will confirm check point to next levels)
- L4
- L6

Select the environment and application need to deploy.

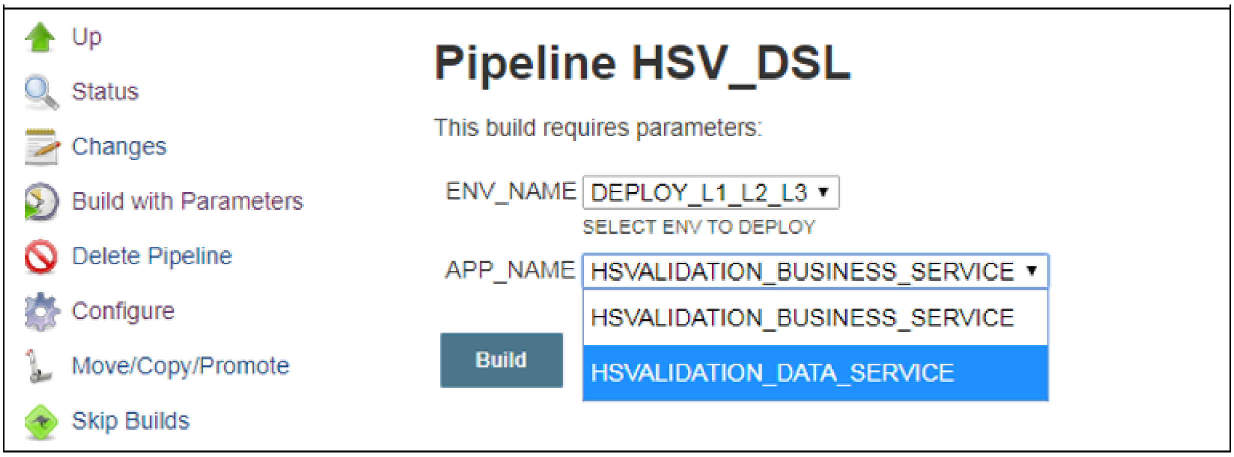


Pipeline HSV_DSL

This build requires parameters:

ENV_NAME DEPLOY_L1_L2_L3
DEPLOY_L1_L2_L3
L4
L6

APP_NAME NESS_SERVICE



Pipeline HSV_DSL

This build requires parameters:

ENV_NAME DEPLOY_L1_L2_L3 ▾
SELECT ENV TO DEPLOY

APP_NAME HSVALIDATION_BUSINESS_SERVICE ▾
HSVALIDATION_BUSINESS_SERVICE
HSVALIDATION_DATA_SERVICE

Send an automated email after the successful completion of test suite execution. In each of the test suite at respective Test level
AD-HOC Smoke Test Execution from Jenkins

Apart from daily execution of smoke test, it can also be triggered manually via Jenkins
Below is the process for the same;

1) Go to below link
https://jenkins.prod.cloud.SHIPPING_SECTORS.com:8443/jenkins/job/CI CD CT AUTOMATION-PROJECT/job/MicroServices/job/Scripted_Microservice/
https://jenkins.prod.cloud.SHIPPING_SECTORS.com:8443/jenkins/job/CI CD CT AUTOMATION-PROJECT/job/MicroServices/job/Scripted_Microservice/job/HSV_DSL/

2) Select Build with Parameters
3) Select Environment and Application
4) Click on the build

this will trigger the chosen smoke test.

Regression Test Implementation

Integrate Regression Test suite

Copy all the Regression test scenarios in below folder in Linux/ windows Box

/opt/SHIPPING_SECTORS/tradetools/LISA/lisadevtest/Projects/CI CD CT AUTOMATION-PROJECT_DEMO/Tests/

Create Test Runner command for all test suite:

Ex:**./TestRunner -a -u 3592564 -p May20198 -t /opt/SHIPPING_SECTORS/tradetools/LISA/lisadevtest/Projects/CI CD CT AUTOMATION-PROJECT_DEMO/Tests/All_Services_new_framework.tst -m ssl://srh00457.ute.SHIPPING_SECTORS.com:2010/Registry**

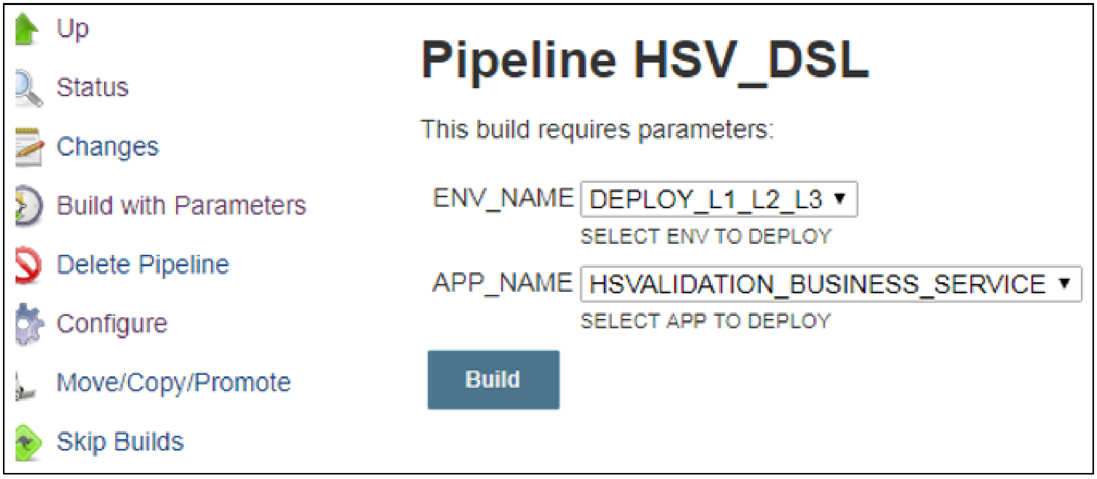
Automation of Regression test Scenarios

CI - CD Team will create pipeline in Jenkins to Trigger Regression Test suites across all the levels

https://jenkins.prod.cloud.SHIPPING_SECTORS.com:8443/jenkins/job/CI CD CT AUTOMATION-PROJECT/job/MicroServices/job/Scripted_Microservice/
https://jenkins.prod.cloud.SHIPPING_SECTORS.com:8443/jenkins/job/CI CD CT AUTOMATION-PROJECT/job/MicroServices/job/Scripted_Microservice/job/HSV_DSL/

- DEV-1, DEV-2, QA
- STAGING
- PROD

Pipeline structure:



Pipeline HSV_DSL

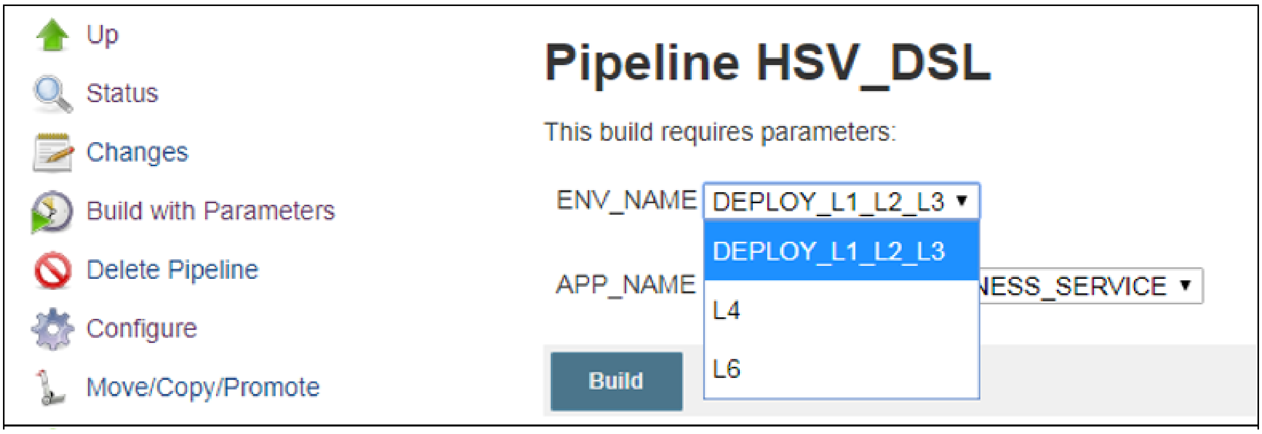
This build requires parameters:

ENV_NAME DEPLOY_L1_L2_L3 ▾
SELECT ENV TO DEPLOY

APP_NAME HSVALIDATION_BUSINESS_SERVICE ▾
SELECT APP TO DEPLOY

Execution of Regression Test Suite

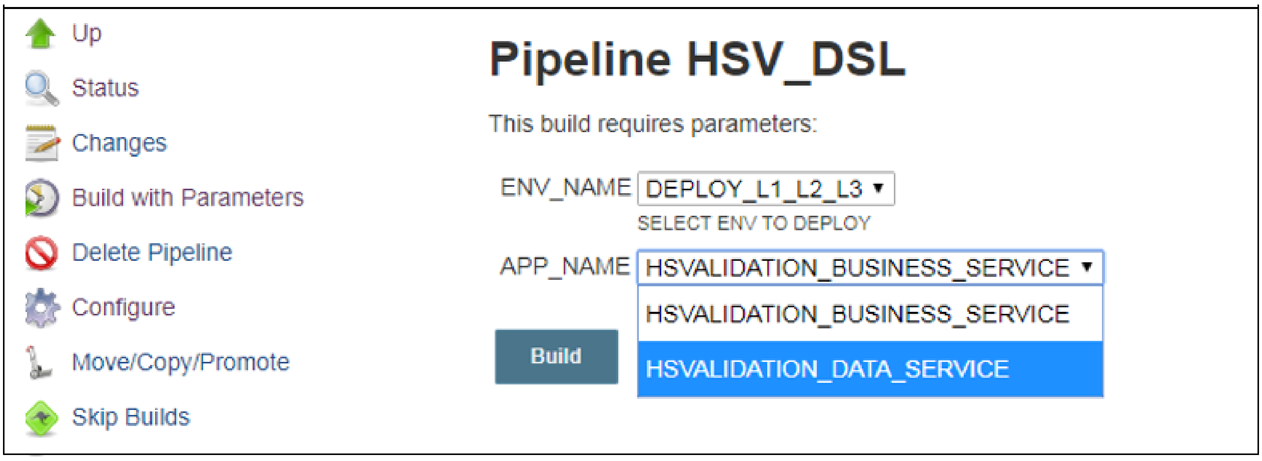
- Auto Triggering along with build
- DEV-1, DEV-2, QA (Single deployment and Load master will confirm check point to next levels)
- STAGING
- PROD
- Ad-Hoc Triggering of Regression test suite on Required environment


Pipeline HSV_DSL

This build requires parameters:

ENV_NAME DEPLOY_L1_L2_L3 ▾

APP_NAME …NESS_SERVICE ▾


Pipeline HSV_DSL

This build requires parameters:

ENV_NAME DEPLOY_L1_L2_L3 ▾
SELECT ENV TO DEPLOY

APP_NAME HSVALIDATION_BUSINESS_SERVICE ▾

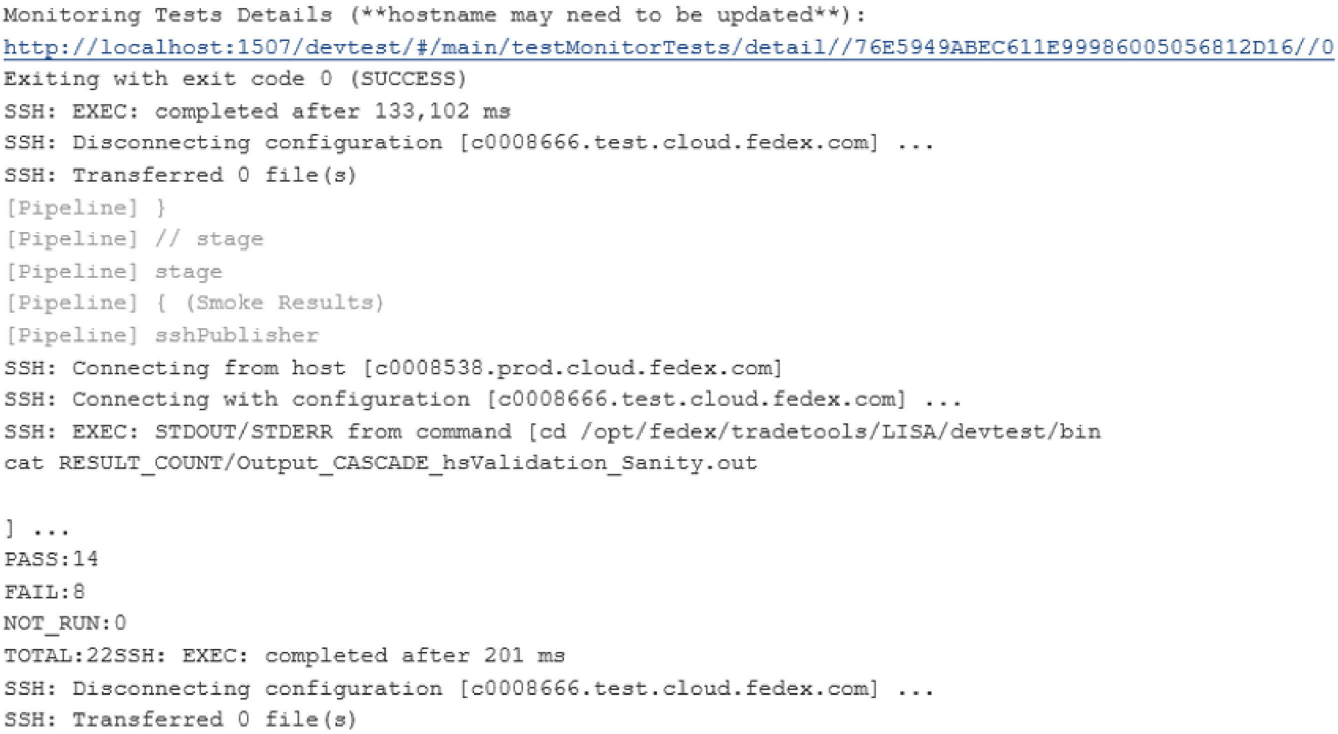
Send an automated email after the successful completion of test suite execution. In each of the test suite at respective Test level

Validation of the Results & Logs

- We can see Pass/ Fail Test scenarios in Jenkins
- We can monitor build status in Jenkins
- Logs can be available in source box

LINUX : http://c0008666.test.cloud.SHIPPING_SECTORS.com **/opt/SHIPPING_SECTORS/tradetools/LISA/lisadevtest/ Projects/CI CD CT AUTOMATION-PROJECT_DEMO/Results**

Jenkins Console output:



```
Monitoring Tests Details (**hostname may need to be updated**):
http://localhost:1507/devtest/#/main/testMonitorTests/detail//76E5949ABEC611E99986005056812D16//0
Exiting with exit code 0 (SUCCESS)
SSH: EXEC: completed after 133,102 ms
SSH: Disconnecting configuration [c0008666.test.cloud.fedex.com] ...
SSH: Transferred 0 file(s)
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Smoke Results)
[Pipeline] sshPublisher
SSH: Connecting from host [c0008538.prod.cloud.fedex.com]
SSH: Connecting with configuration [c0008666.test.cloud.fedex.com] ...
SSH: EXEC: STDOUT/STDERR from command [cd /opt/fedex/tradetools/LISA/devtest/bin
cat RESULT_COUNT/Output_CASCADE_hsValidation_Sanity.out

] ...
PASS:14
FAIL:8
NOT_RUN:0
TOTAL:22SSH: EXEC: completed after 201 ms
SSH: Disconnecting configuration [c0008666.test.cloud.fedex.com] ...
SSH: Transferred 0 file(s)
```

5. Conclusion

CI CD CT Automation-Project made sure that customers can experience the accurate product code, estimated duties, estimated taxes, transit time, and digitalized experience by utilizing the cutting- edge technologies like Big Data and Analytics.

CI CD CT Automation-Project is to build an innovative, intelligent platform that classifies commodities to a fully qualified code with a single click.