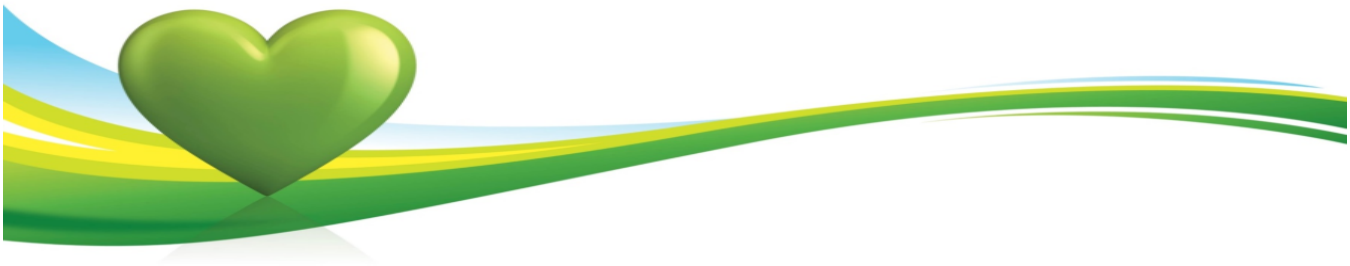
Release Management Process-v1



LV Data Transformation Programme
Release Management Process This document contains information that is confidential and proprietary to LV= and is restricted in usage. You are not authorized to release any information contained in this document to any third party without prior written permission from LV=.

Contents

Objective

The objective is to showcase how code is pushed to higher environment.
Databricks GitHub Repo & Databricks Pipeline Brief Explanation :-

1. The GitHub Repo having below files.

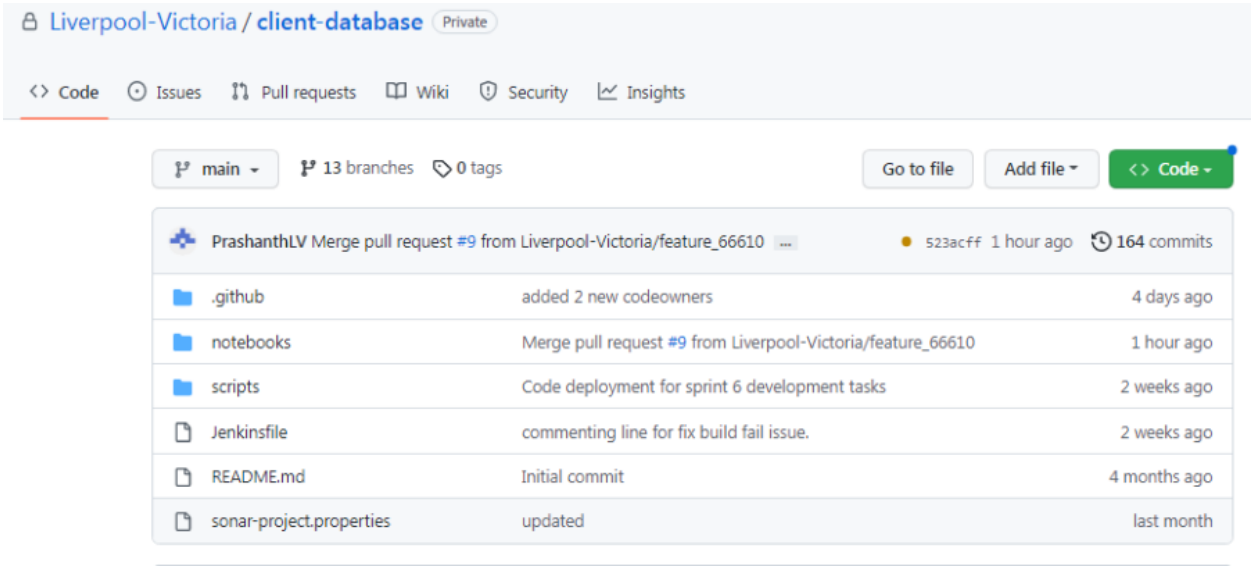


Fig.a

1. Notebooks:- This folder contains all python scripts.
2. Scripts:- This file contains scripts to create and schedule a databricks job.
3. Jenkinsfile:- This file having pipeline stages like Static Analysis, Installing py modules, Package, Package to Artifact, Databricks Setup, Deploy to Dev, Email Notification, Create & Run job, Authorize UAT deploy.
4. Sonar-project-properties:- This file contains required properties to scan Github repo code.
5. Developers can write there python scripts into there local branch in GitHub Repo. If script is working fine then the developer commit there changes to main branch.
6. When the changes are committed to main branch the Jenkins pipeline will be trigger.
7. In first stage of pipeline all notebooks are package into zip then later on they are uploaded to artifact.

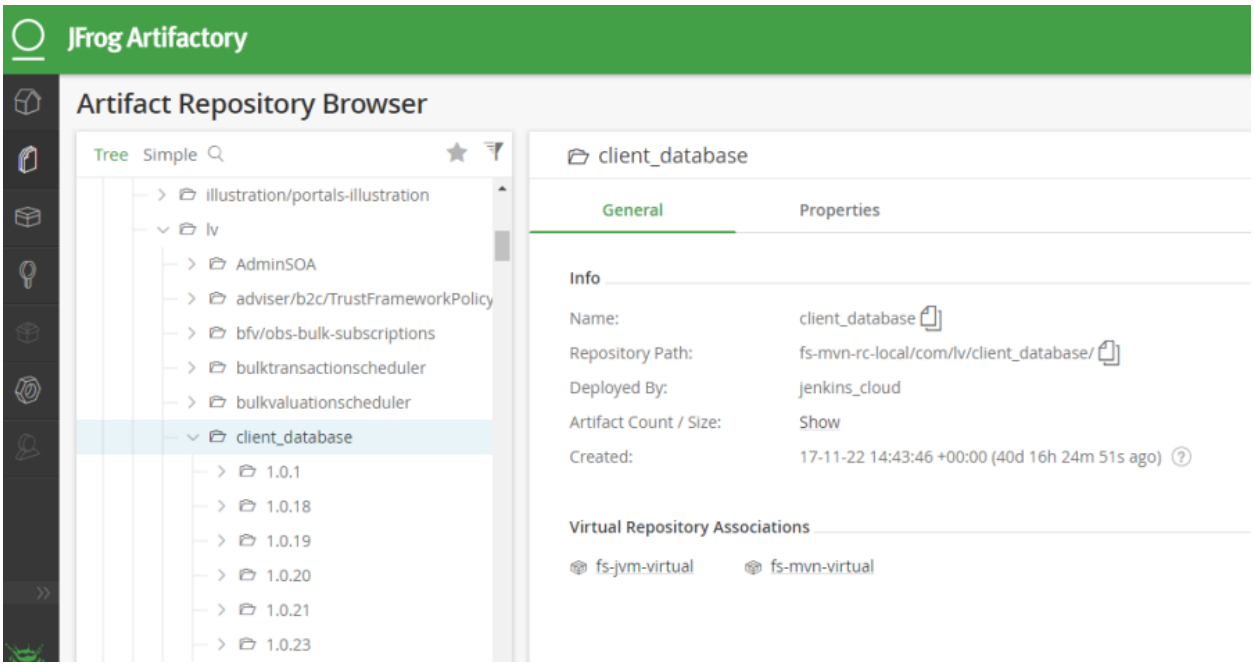


Fig.b

1. At this stage Artifact will be created then Authorization will be needed to continue further execution of pipeline. Once Dev push is authorized, artifact will be download first then deployed to **Azure Databricks Dev environment**.
2. Once pipeline trigger authorization mail is sent to **Dev** environment authorized person.

From: jenkinslvfsnet@lv.com [mailto:jenkinslvfsnet@lv.com]
Sent: 19 December 2022 11:35
To: Saroliya, Chirag
Subject: Authorization request for UAT Deployment 'DATA_SCIENCE/client-database/main [121]'

Hi,

Jenkins needs your Authorization to proceed further with UAT deployment : 'DATA_SCIENCE/client-database/main [121]':

Kindly approve it, if artifact needs to be deployed on UAT environment. Refer this link to take further action "DATA_SCIENCE/client-database/main [121]"

Fig.c

- Once we open link from mail we will go to the console output window then pipeline proceed for the further execution.

- If pipeline build successfully it means all notebooks deployed successfully to **Azure Databricks Dev environment**.

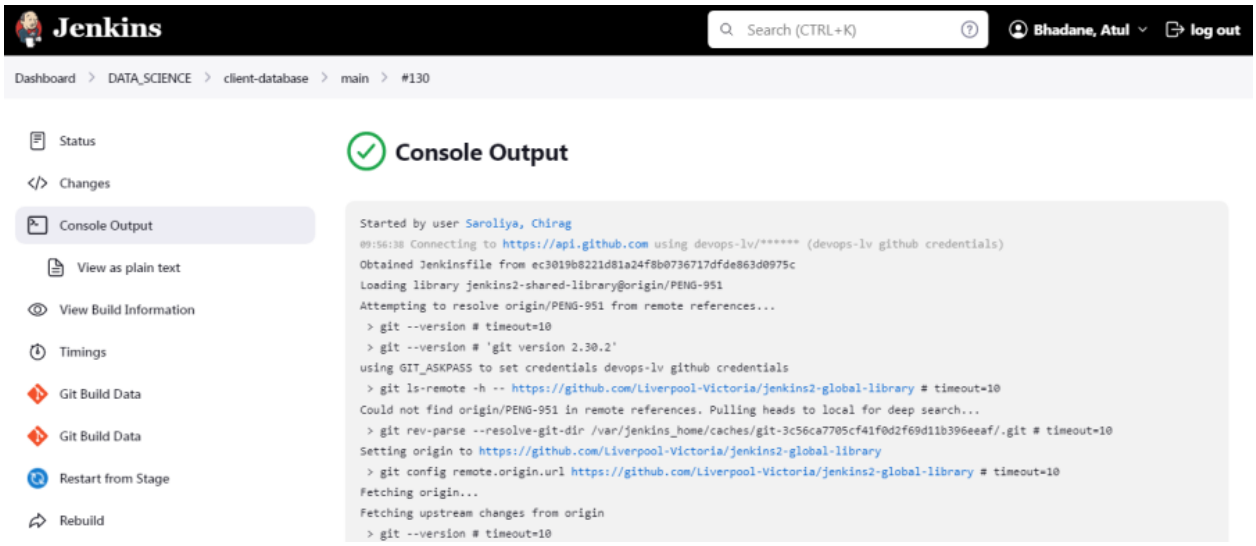


Fig.e

- Databricks Code Deployed on Dev Workspace

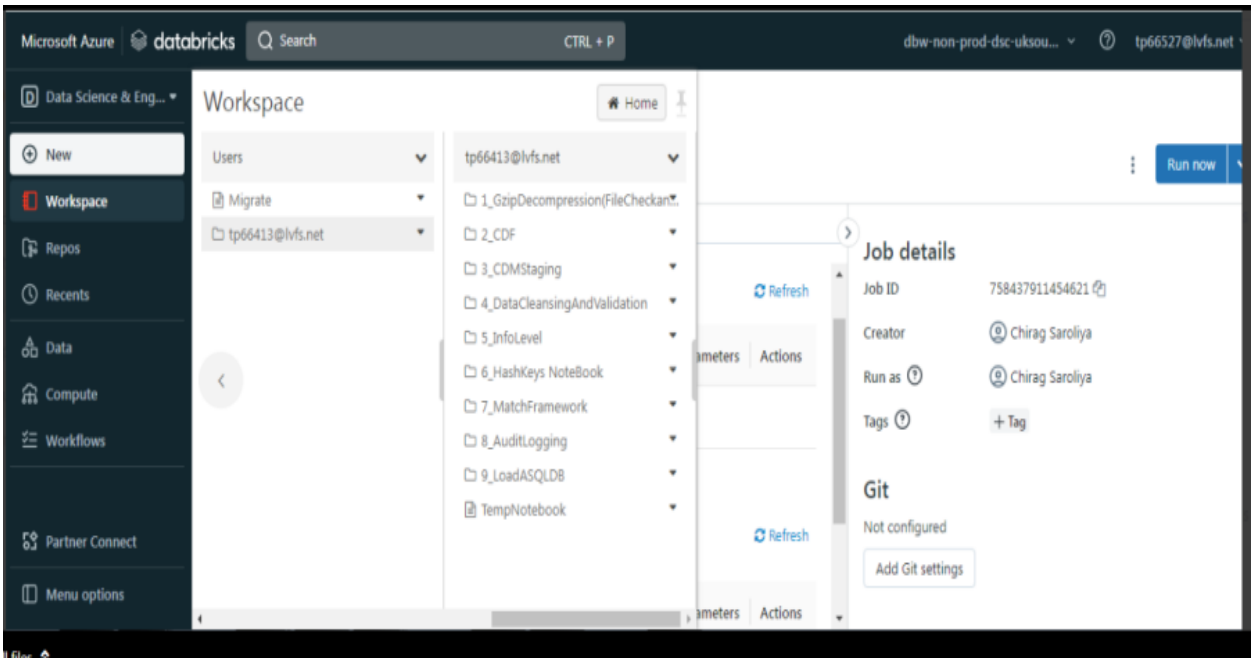


Fig.f

As shown in above **Fig.f,** Once pipeline build successfully all notebooks are deployed to Azure Databricks Dev environment.

- Databricks Job Scheduling & Run

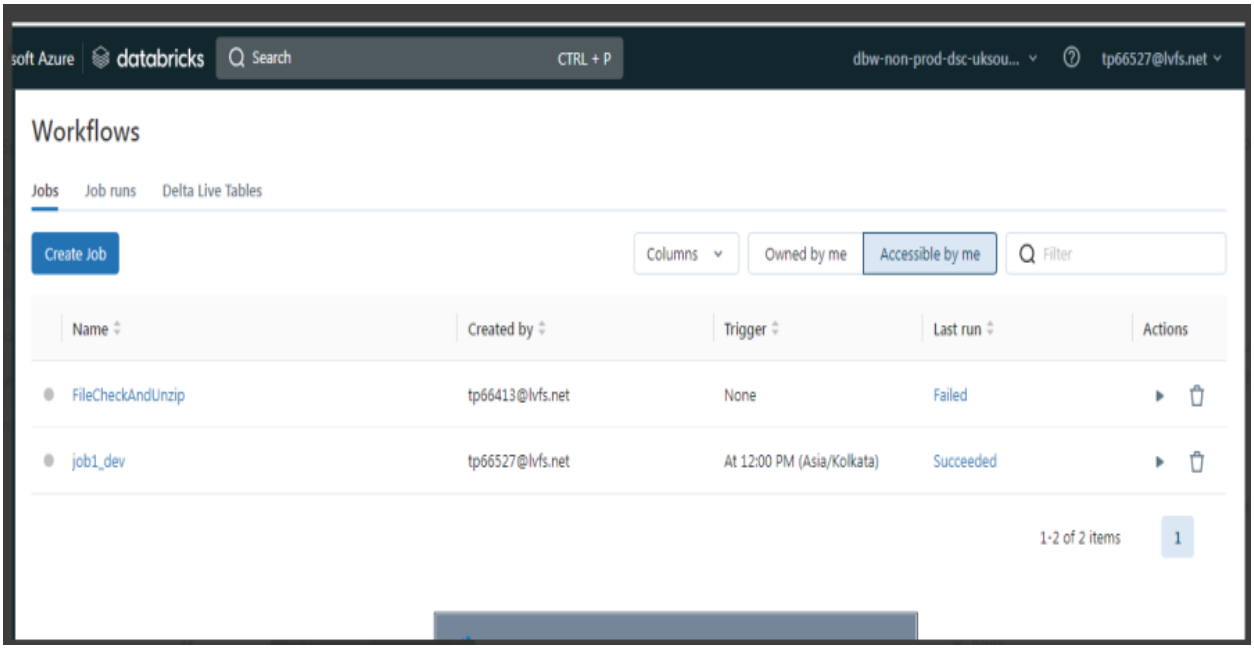


Fig.g

As shown in above **Fig.g** Using Jenkins pipeline we can schedule a job and assign cluster to that job. Notebook which we need to run on regular basis will be attached to that job and that job will get executed at scheduled time.

Database GitHub Repo & Database Pipeline Brief Explanation :-

1. The GitHub Repo having below files.

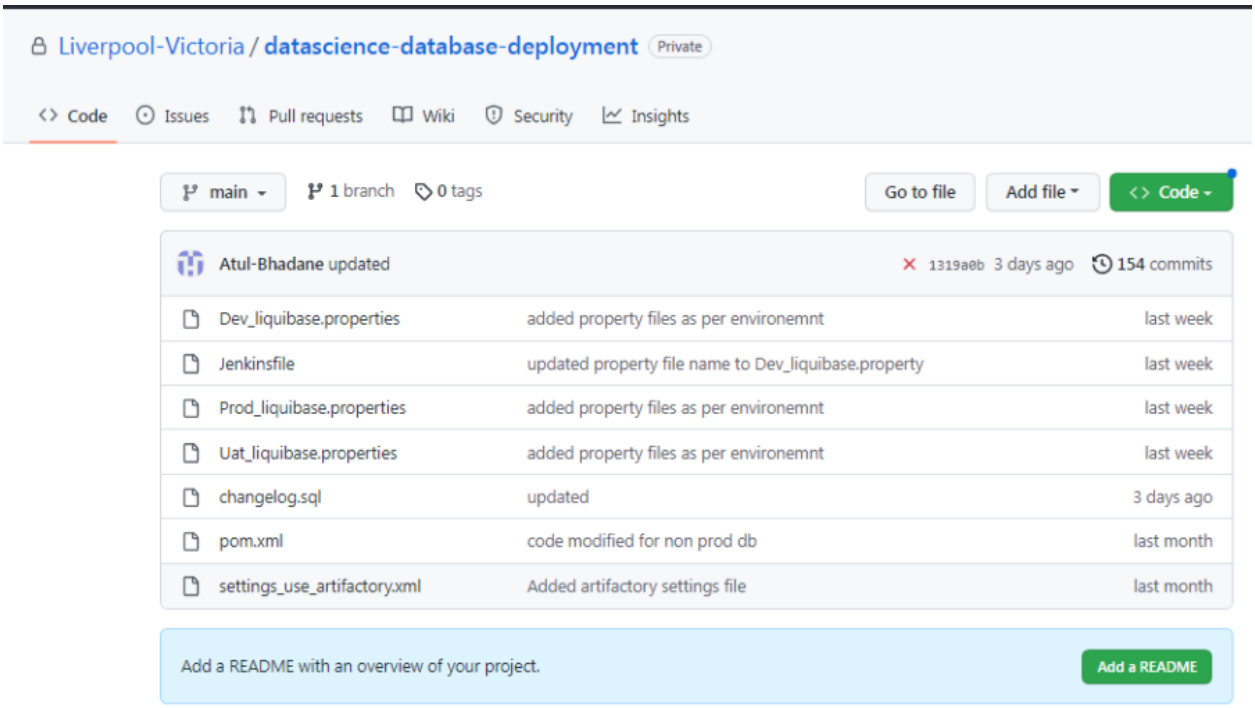


Fig.a

1. Jenkinsfile:- This file contains various stages of pipeline. Like 'Upload Artifact ', 'Authorize Dev Deploy', 'Deploy to Dev', 'Authorize Test Deploy', 'Deploy to Test', 'Authorize Prod Deploy', 'Deploy to Prod' environment.
2. changelog.sql :- This file contains sql scripts which we want to deploy to database.

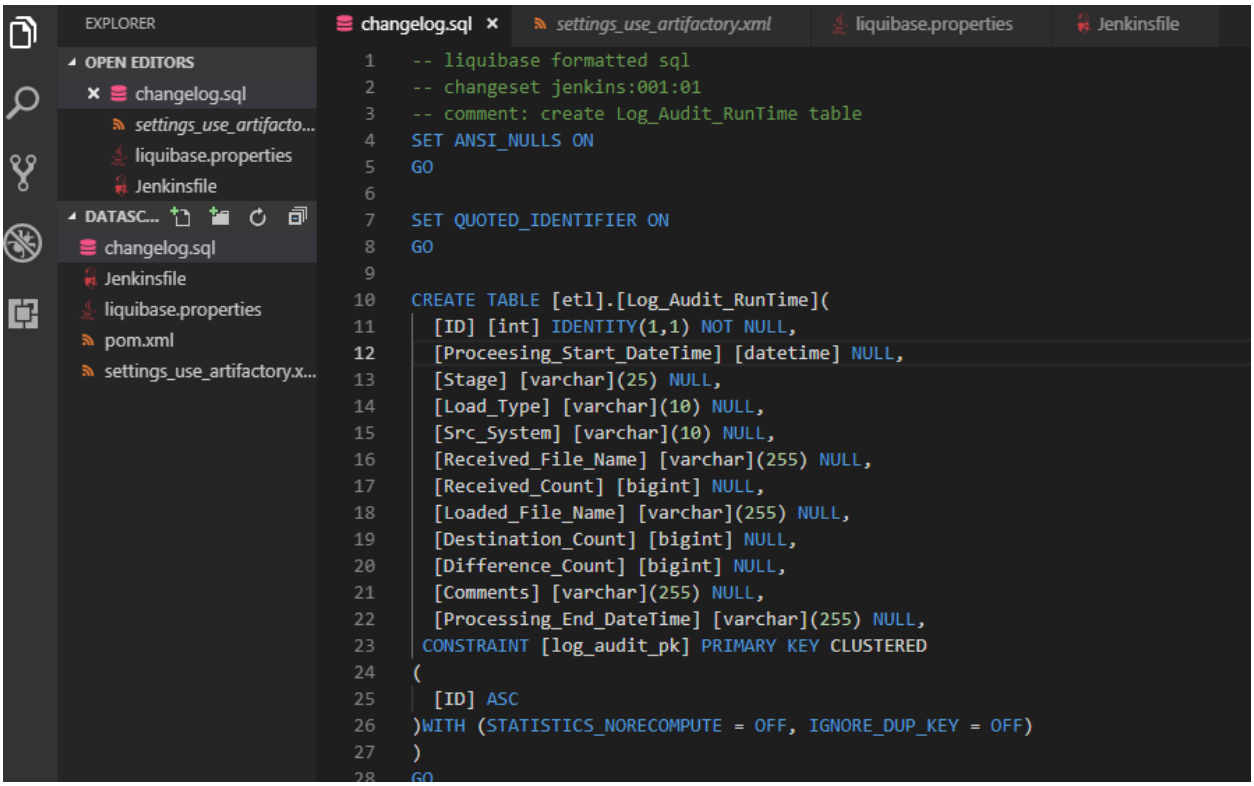
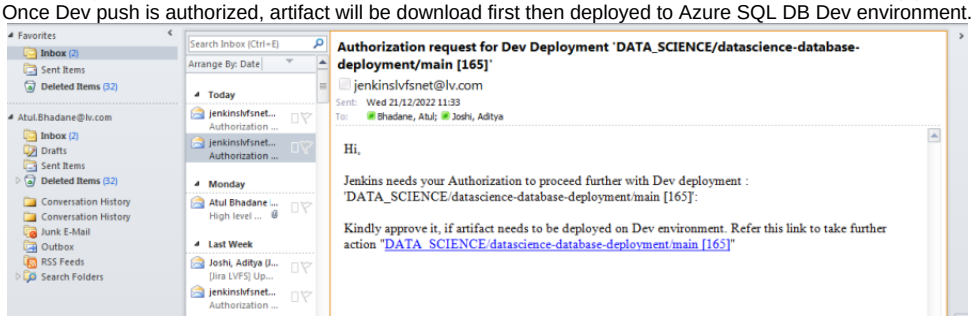


Fig.b

1. POM.xml :- This files contains Liquibase-maven plugin & Jdbc- dependency.
2. liquibase.properties :- This file contains target database URL information and path of changelog.sql file according to the Environment.

- Developers can write there sql scripts into there local branch in GitHub Repo. If script is working fine then the developer commit there changes to main branch.
- When the changes are committed to main branch the Jenkins pipeline will be trigger.

- At this stage Artifact will be created then Authorization will be needed to continue further execution of pipeline.
- Once Dev push is authorized, artifact will be download first then deployed to Azure SQL DB Dev environment.



- Once pipeline trigger authorization mail send to **Dev** environment authorized person.

- Once we open link from mail we will go to the console output window then pipeline proceed for the further execution.

Fig.c

- If pipeline build successfully it means the database sql script deployed successfully to Dev environment Database.

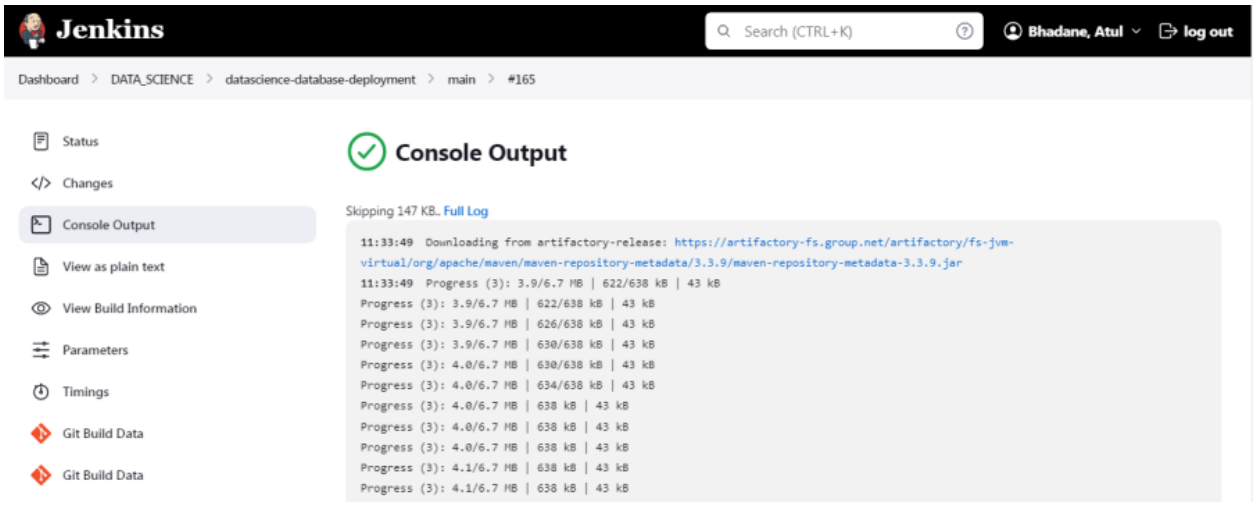


Fig.d.

- If we want push code from Dev to higher environment (i.e UAT, Prod) first need to update **liquibase.properties** file with **Database connection string.**
- Aslo update the **jenkinsfile** with database name & there credentials to connect. The credentials should be store into the jenkins variable.
- The Database deployed successfully according to the **changelog.sql** file.
- According to the **Fig.b** from this document, the table listed in **Fig.e** is successfully deployed and all the sql queries in that changelog.sql file are now executed. Database is updated with it.
- Following is the snapshot of the database, we can see that the same query is deployed to the database. Please refer highlighted section in **Fig.e**.

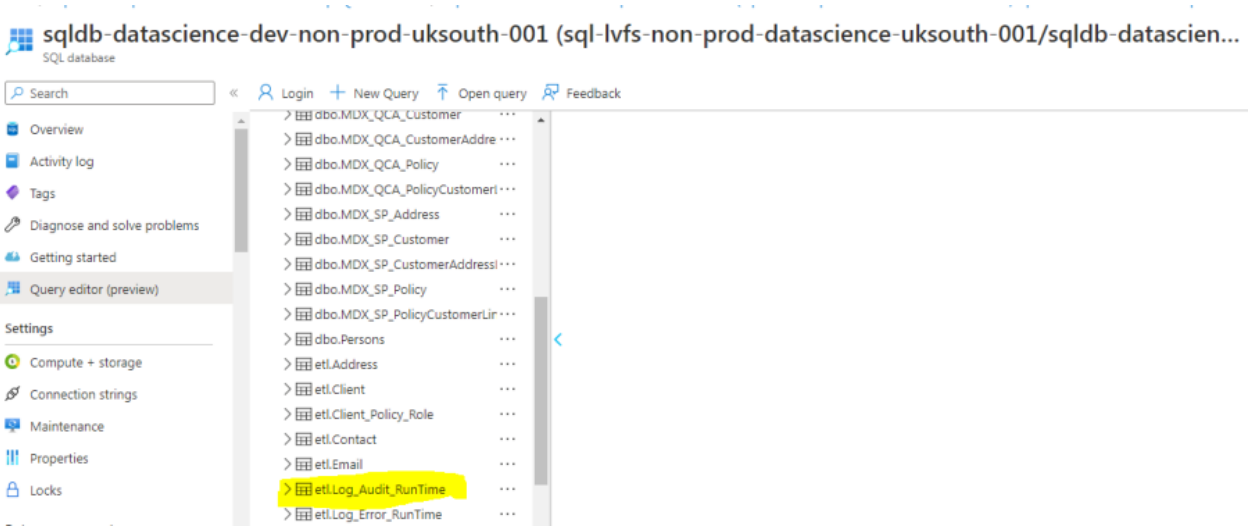

sqldb-datascience-dev-non-prod-uksouth-001 (sql-lvfs-non-prod-datascience-uksouth-001/sqldb-datascien...
SQL database
Search
Login
New Query
Open query
Feedback
Overview
Activity log
Tags
Diagnose and solve problems
Getting started
Query editor (preview)
Settings
Compute + storage
Connection strings
Maintenance
Properties
Locks
dbo.MDX_QCA_Customer
dbo.MDX_QCA_CustomerAddre
dbo.MDX_QCA_Policy
dbo.MDX_QCA_PolicyCustomerI
dbo.MDX_SP_Address
dbo.MDX_SP_Customer
dbo.MDX_SP_CustomerAddressI
dbo.MDX_SP_Policy
dbo.MDX_SP_PolicyCustomerLir
dbo.Persons
etl.Address
etl.Client
etl.Client_Policy_Role
etl.Contact
etl.Email
etl.Log_Audit_RunTime
etl.Log_Error_RunTime

Fig.e