Library Management System

Introduction

Library Book Management System is a web-based application designed to facilitate the management and discovery of books in the library environment. This design document describes the design process, material, design, database model, authentication process, error handling, and other functions.

Project Overview

The library management system provides many functions such as user registration, searching and searching, transferring books, reading materials, managing books and notes. The system is designed to provide a user-friendly interface for both users and librarians/administrators.

Scope

This technical design document covers the following aspects of the Libro Book Management System:

- **Architecture**: Clean Architecture, promoting modularity and testability.
- **Design Patterns**: Service Pattern, Repository Pattern, and Unit of Work Pattern.
- **Error Handling**: Middleware-based logging for capturing and tracking important events, requests, and errors.
- **Authentication**: Form based authentication for secure user authentication and authorization.
- **Database Design**: Utilizing a suitable database structure for storing book information, user profiles, and other relevant data.
- **Additional Features**: Reading Lists, Book Reviews and Ratings, Notifications, and Book Recommendations.

Clean Architecture

Libraries maintain books in accordance with clean design standards. This architecture supports change, control, and testing by separating concerns and making the code base flexible and adaptable to future changes. It includes domains, applications, infrastructure, and presentation layers, each with their own roles and dependencies.

Service Pattern

System management libraries use service models to organize business logic and functionality into reusable and interoperable components. This model promotes flexibility and separation of concerns, allows for a clean organization, increases security and provides a variety of services to achieve higher levels of implementation.

Repository Pattern

Library Management System

The repository model is used in library management systems to provide an abstraction layer between the application and the data persistence layer. This model encourages clear separation of concerns and standards. It provides a consistent and standardized approach to accessing and managing data, developing regulatory frameworks, measurement metrics and benchmarking.

Error Handling and Logging

Library management library includes middleware-based decision making to capture and track critical events, requests, and errors throughout the application process. This approach allows centralized access, making it easier to debug, monitor and analyse usage behaviour, performance and security. Agent software logging provides critical information for troubleshooting, auditing, and compliance purposes.

Fluent Validation Package

Fluent Validation package is used in the library manual administration process to simplify and improve the validation process. It provides simple and clear instructions for validating code by separating the validation logic from the written or view model. This separation of concerns promotes a cleaner and more manageable approach. Fluent Validation improves user experience by providing error messages for invalid validation.

Getting Started

To run the Library Book Management System locally, follow the instructions below:

Prerequisites

Web library management system based on ASP.NET MVC.

Relationship Object Model is Entity framework

Environment Requirements: Visual Studio 2017+, MVC, SQL server

Steps

Clone the repository: https://github.com/vigneshkumar-957/LibraryManagement_1.git

Navigate to the project directory:

- Visual studio Open LMS solution
- Open the PM console
- Type cd LMS.DataAccess to enter the project folder

Library Management System

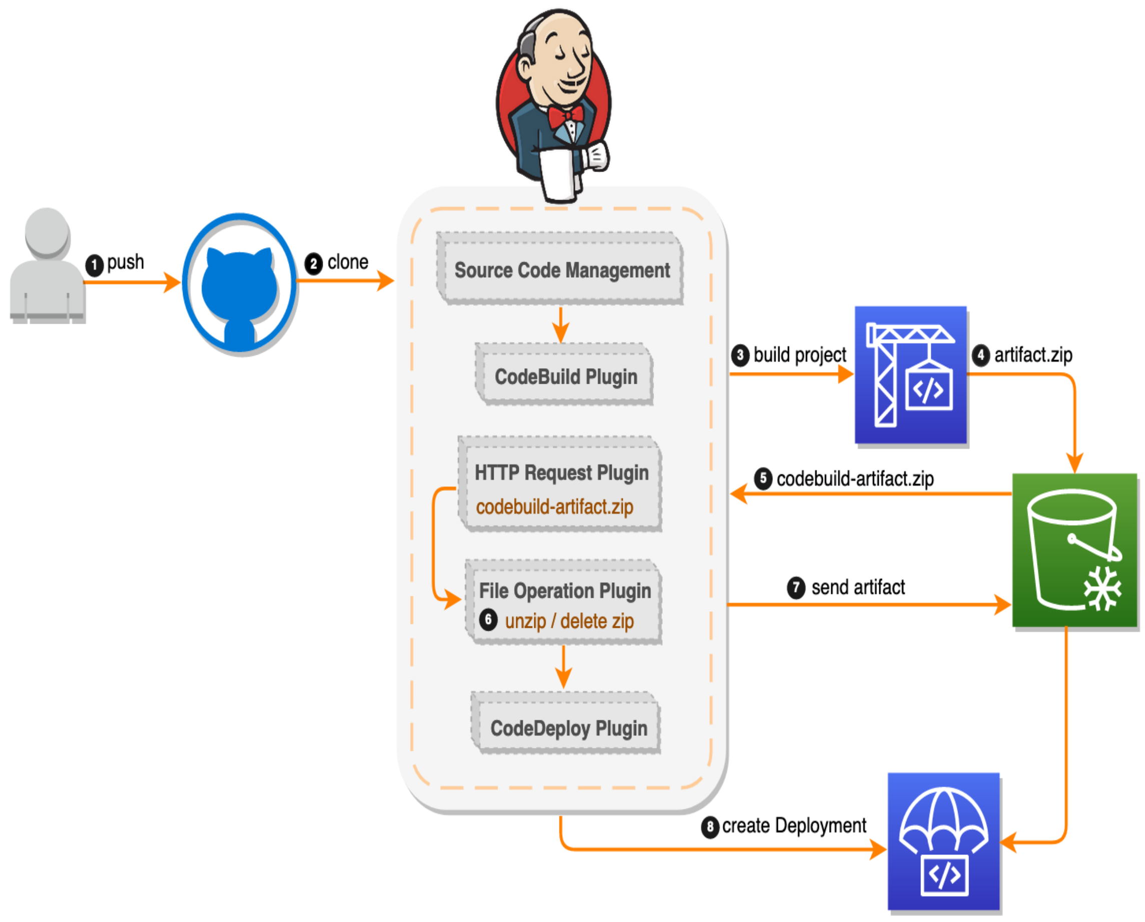
- Please enter **update-database –verbose** after configuring the sql server connection string
- Run IIS press

When the web page refreshes without the error, it can be run.

CI –CD Implementation

Pre-requisites

1. Jenkins Server with Windows Slave attached.
2. AWS Account



Jenkin file

This is the sequential script for the pipeline to execute the stages one by one. Here we used 4 steps - Checkout, Build, Release, Deploy.

Library Management System

Stage 1 - Checkout: In this stage, we provide the URL and Git repository, branch as main and git credentials id. I pull the source code to workspace. Default workspace is - C:\ProgramData\Jenkins.jenkins\workspace

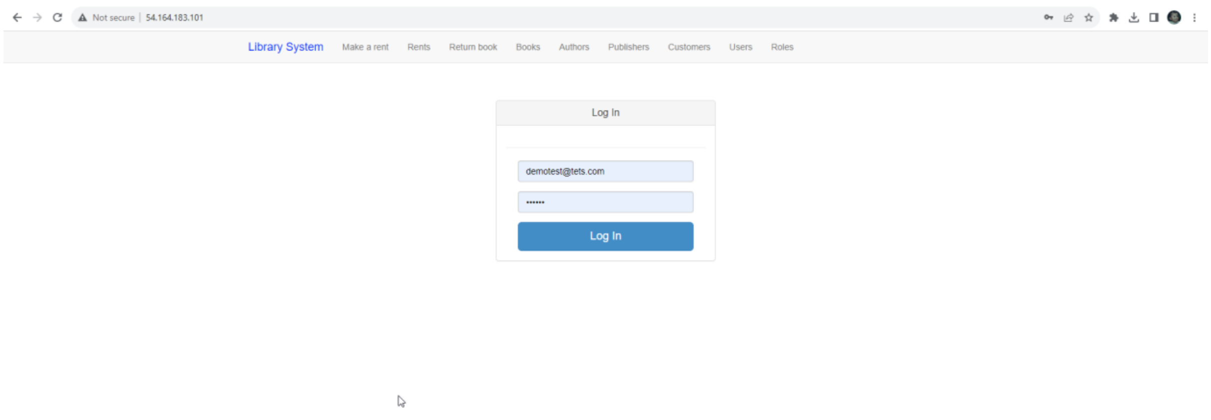
Stage 2 - Build: In this stage, I have build the specific project. Here LMS.API. Keep in mind you have to direct the location where sln file exists.

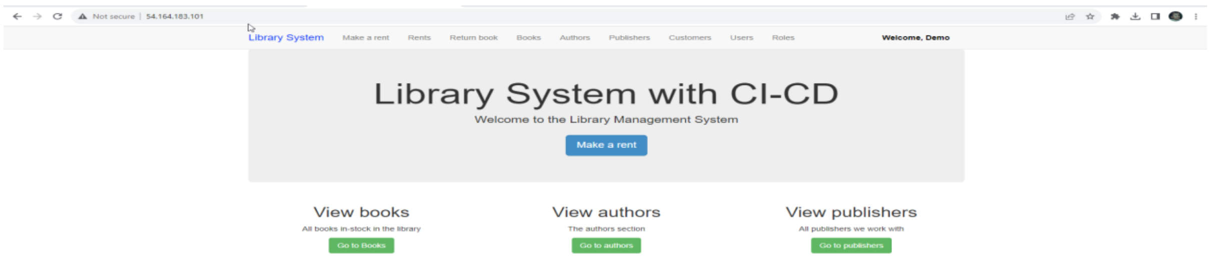
Stage 3 - Release: In this stage, A release package LMS.zip is created in the mentioned location which is provided on JenkinsProfile.pubxml. You may check each steps output using console after executing pipeline.

Stage 4 - Deploy: In this stage, first stop IIS. Then deploy package to IIS and then Start IIS again. So, I have used three bat commands here.

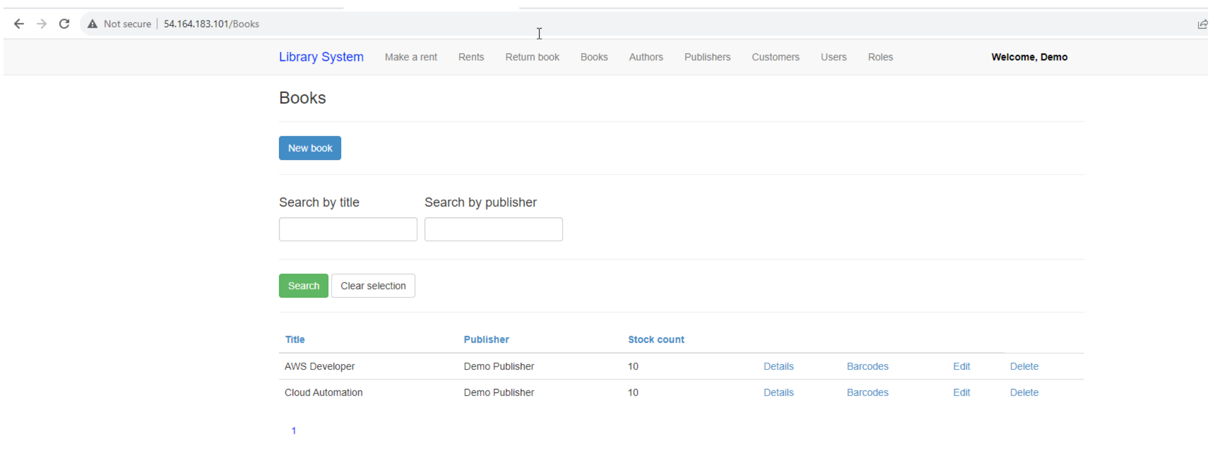
Screenshots:

Url: http://54.164.183.101/





Library Management System

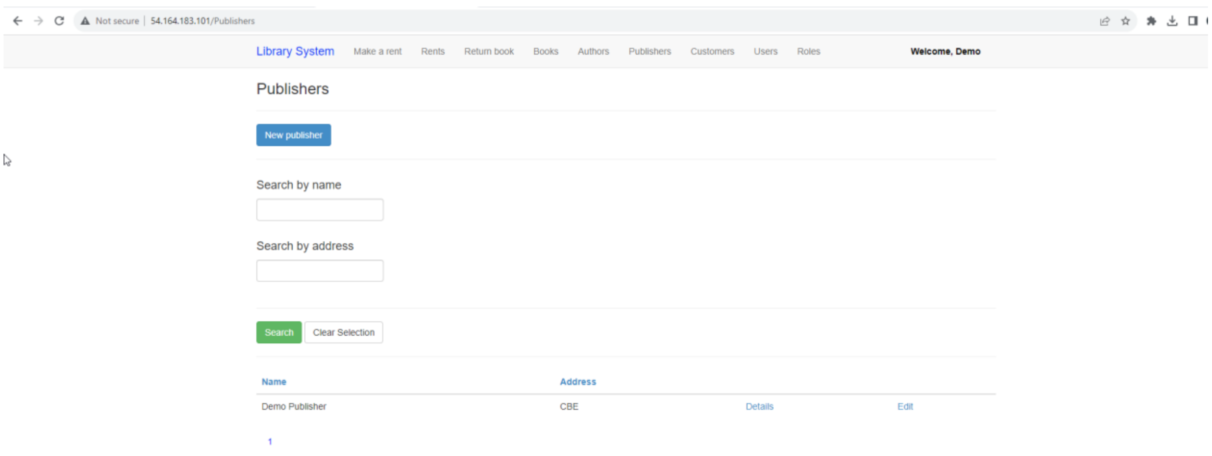


Books

Search by title

Search by publisher

Title	Publisher	Stock count				
AWS Developer	Demo Publisher	10	Details	Barcodes	Edit	Delete
Cloud Automation	Demo Publisher	10	Details	Barcodes	Edit	Delete

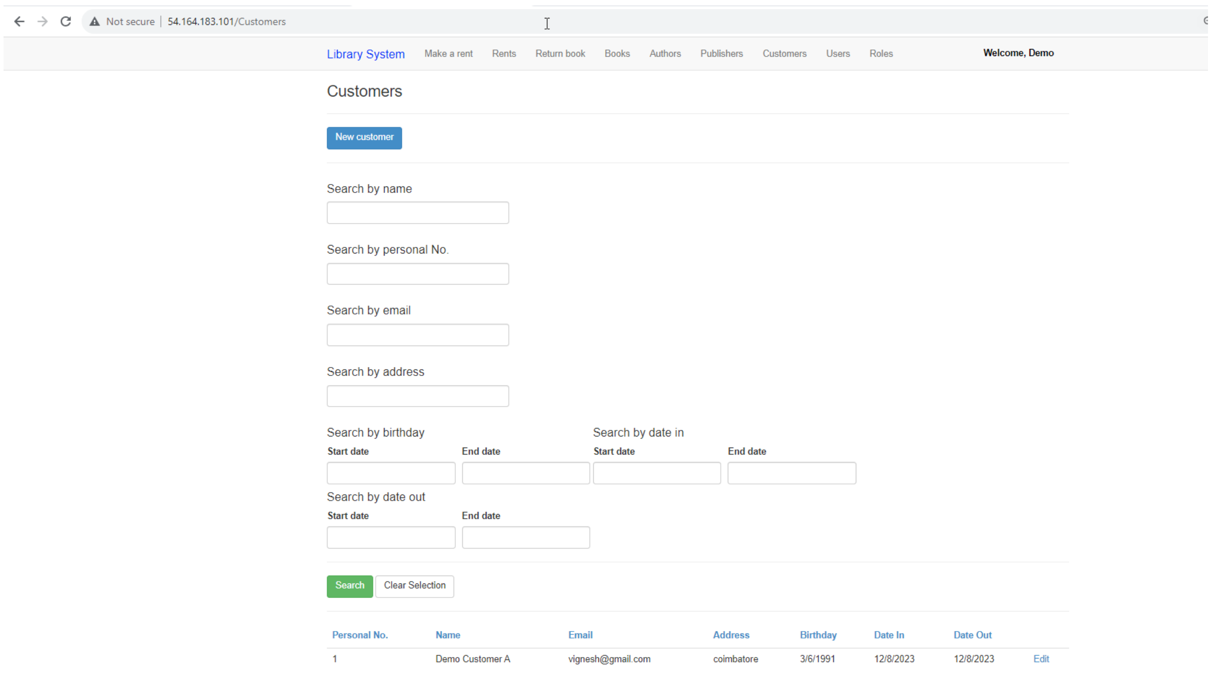


Publishers

Search by name

Search by address

Name	Address		
Demo Publisher	CBE	Details	Edit



Customers

Search by name

Search by personal No.

Search by email

Search by address

Search by birthday
Start date
End date

Search by date in
Start date
End date

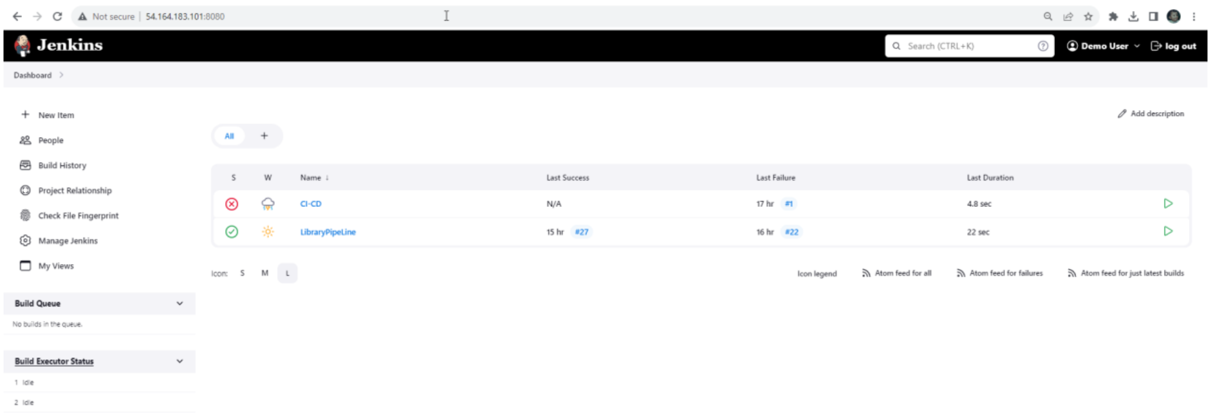
Search by date out
Start date
End date

Personal No.	Name	Email	Address	Birthday	Date In	Date Out	
1	Demo Customer A	vignesh@gmail.com	coimbatore	3/6/1991	12/8/2023	12/8/2023	Edit

Library Management System

CI-CD Scrrenshots

Url: http://54.84.14.199:8080/



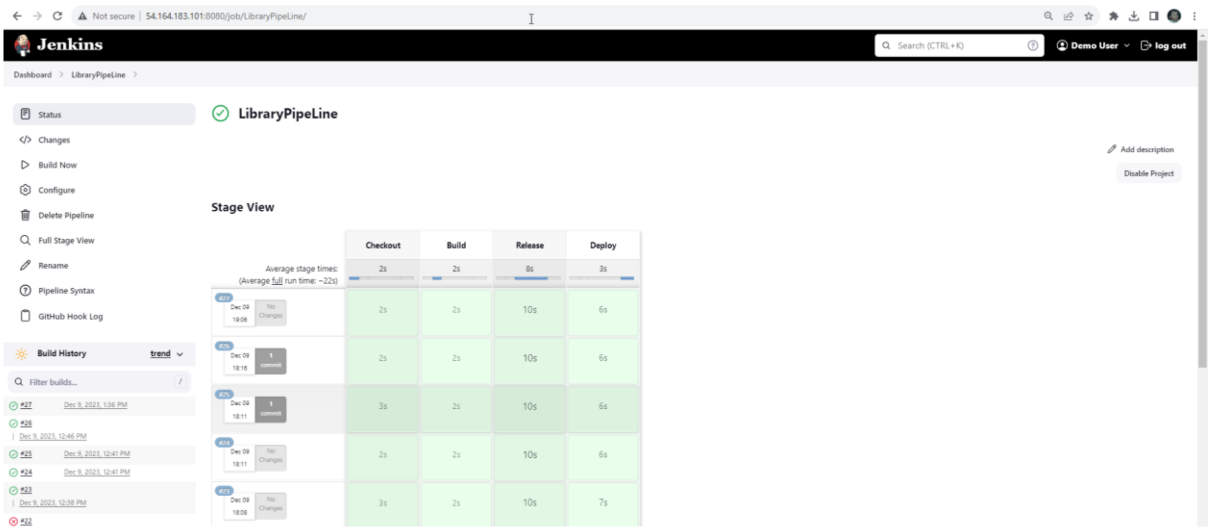
Not secure | 54.164.183.101:8080

Jenkins — Dashboard

S	W	Name ↓	Last Success	Last Failure	Last Duration
✗		CI-CD	N/A	17 hr #1	4.8 sec
✓		LibraryPipeLine	15 hr #27	16 hr #22	22 sec

Build Queue: No builds in the queue.

Build Executor Status: 1 Idle, 2 Idle



Not secure | 54.164.183.101:8080/job/LibraryPipeLine/

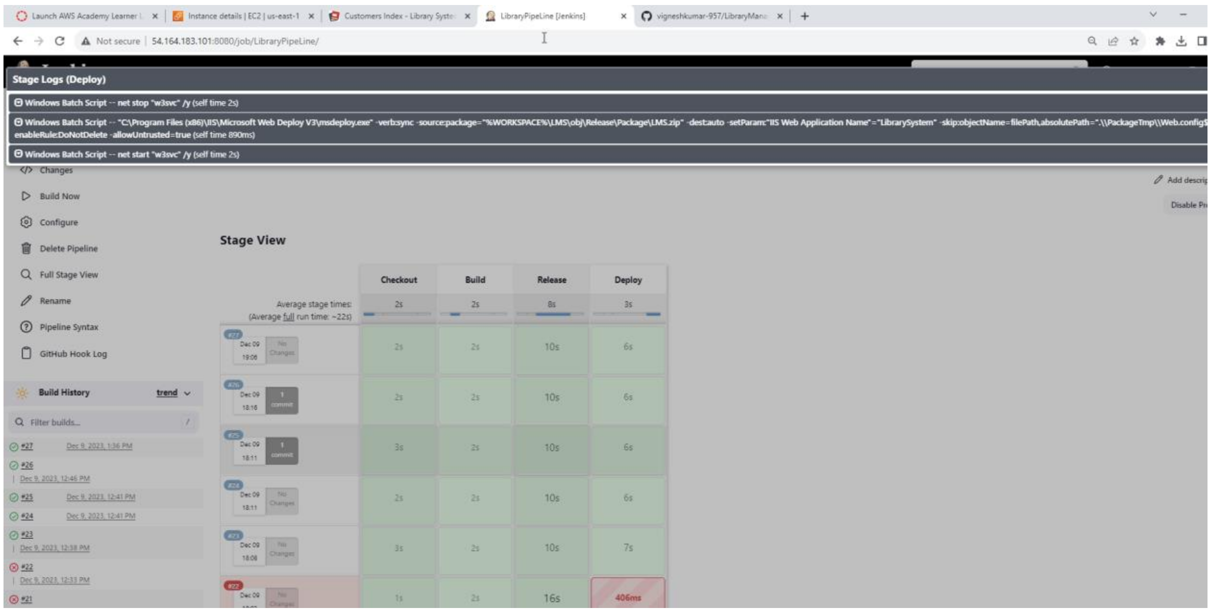
Dashboard > LibraryPipeLine

LibraryPipeLine

Stage View

	Checkout	Build	Release	Deploy
Average stage times: (Average full run time: ~22s)	2s	2s	8s	3s
#27 Dec 09 19:06 No Changes	2s	2s	10s	6s
#26 Dec 09 18:16 1 commit	2s	2s	10s	6s
#25 Dec 09 18:11 1 commit	3s	2s	10s	6s
#24 Dec 09 18:11 No Changes	2s	2s	10s	6s
#23 Dec 09 18:08 No Changes	3s	2s	10s	7s

Build History: #27 Dec 9, 2023, 1:36 PM; #26 Dec 9, 2023, 12:46 PM; #25 Dec 9, 2023, 12:41 PM; #24 Dec 9, 2023, 12:41 PM; #23 Dec 9, 2023, 12:38 PM; #22



Not secure | 54.164.183.101:8080/job/LibraryPipeLine/

Stage Logs (Deploy)

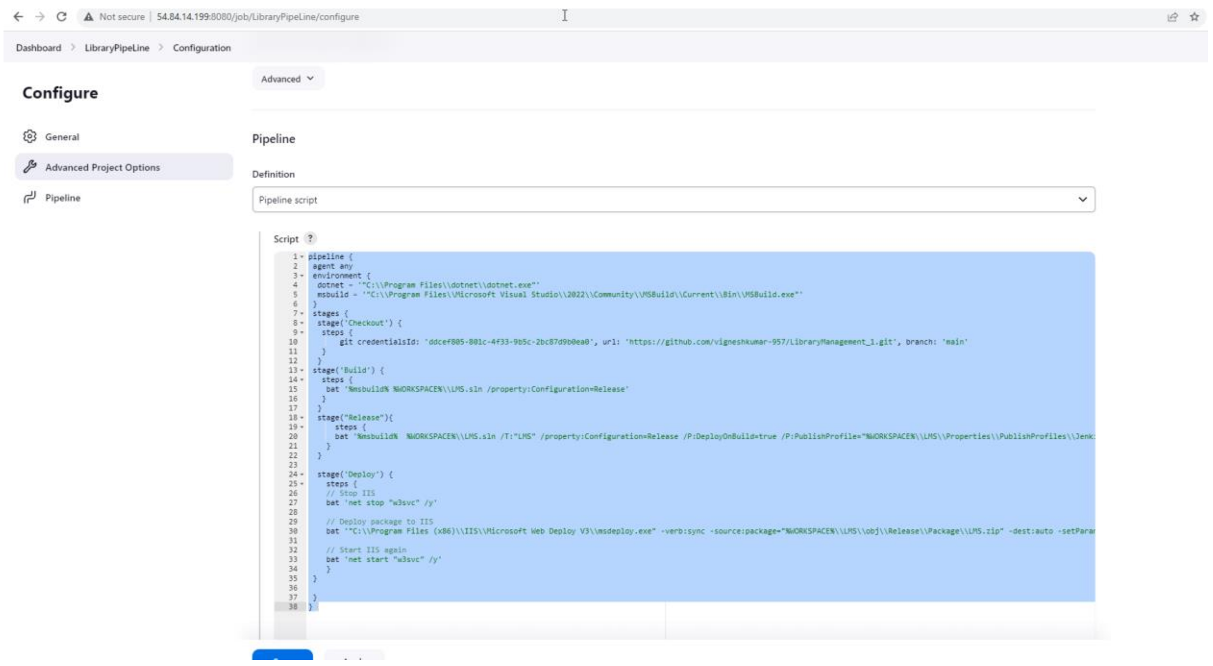
```
Windows Batch Script -- net stop "w3svc" /y (self time 2s)
Windows Batch Script -- "C:\Program Files (x86)\IIS\Microsoft Web Deploy V3\msdeploy.exe" -verb:sync -source:package="%WORKSPACE%\LMS\obj\Release\Package\LMS.zip" -dest:auto -setParam:"IIS Web Application Name"="LibrarySystem" -skip:objectName=filePath,absolutePath=".\\PackageTmp\\Web.config$
enableRule:DoNotDelete -allowUntrusted=true (self time 890ms)
Windows Batch Script -- net start "w3svc" /y (self time 2s)
```

	Checkout	Build	Release	Deploy
Average stage times: (Average full run time: ~22s)	2s	2s	8s	3s
#27 Dec 09 19:06 No Changes	2s	2s	10s	6s
#26 Dec 09 18:16 1 commit	2s	2s	10s	6s
#25 Dec 09 18:11 1 commit	3s	2s	10s	6s
#24 Dec 09 18:11 No Changes	2s	2s	10s	6s
#23 Dec 09 18:08 No Changes	3s	2s	10s	7s
#22 Dec 09 No Changes	1s	2s	16s	406ms

Build History: #27 Dec 9, 2023, 1:36 PM; #26 Dec 9, 2023, 12:46 PM; #25 Dec 9, 2023, 12:41 PM; #24 Dec 9, 2023, 12:41 PM; #23 Dec 9, 2023, 12:38 PM; #22 Dec 9, 2023, 12:33 PM; #21

Launch AWS Academy Learner L
Instance details | EC2 | us-east-1
Customers Index - Library Syste
LibraryPipeLine [Jenkins]
vigneshkumar-957/LibraryMana
Changes
Build Now
Configure
Delete Pipeline
Full Stage View
Rename
Pipeline Syntax
GitHub Hook Log
Stage View
trend
Filter builds...

Library Management System



Not secure | 54.84.14.199:8080/job/LibraryPipeLine/configure

Dashboard > LibraryPipeLine > Configuration

Configure

General

Advanced Project Options

Pipeline

Advanced

Pipeline

Definition

Pipeline script

Script

```
pipeline {
 agent any
 environment {
  dotnet = '"C:\\Program Files\\dotnet\\dotnet.exe"'
  msbuild = '"C:\\Program Files\\Microsoft Visual Studio\\2022\\Community\\MSBuild\\Current\\Bin\\MSBuild.exe"'
 }
 stages {
  stage('Checkout') {
   steps {
       git credentialsId: 'ddcef805-801c-4f33-9b5c-2bc87d9b0ea0', url: 'https://github.com/vigneshkumar-957/LibraryManagement_1.git', branch: 'main'
   }
  }
 stage('Build') {
   steps {
    bat '%msbuild% %WORKSPACE%\\LMS.sln /property:Configuration=Release'
   }
  }
  stage("Release"){
      steps {
      bat "%msbuild%  %WORKSPACE%\\LMS.sln /T:"LMS" /property:Configuration=Release /P:DeployOnBuild=true /P:PublishProfile="%WORKSPACE%\\LMS\\Properties\\PublishProfiles\\Jenk
    }
  }

  stage('Deploy') {
    steps {
    // Stop IIS
    bat 'net stop "w3svc" /y'

    // Deploy package to IIS
    bat '"C:\\Program Files (x86)\\IIS\\Microsoft Web Deploy V3\\msdeploy.exe" -verb:sync -source:package="%WORKSPACE%\\LMS\\obj\\Release\\Package\\LMS.zip" -dest:auto -setParar

    // Start IIS again
    bat 'net start "w3svc" /y'
    }
 }

 }
}
```

Conclusion

I have hosted the LMS application on 80 port on IIS and CICD Jenkins on 8080 port with the following URLs. This URL may get changes if aws EC2 turned off.

http://54.164.183.101/

http://54.164.183.101:8080/

Reference

Pipeline : **https://www.jenkins.io/solutions/pipeline/**
AWS : https://aws.amazon.com/blogs/devops/setting-up-a-ci-cd-pipeline-by-integrating-jenkins-with-aws-codebuild-and-aws-codedeploy/