**Name: Parmita Patre**

**Class: Final Year CSE Div: B**

**Roll: B\_58**

**Assignment 6**

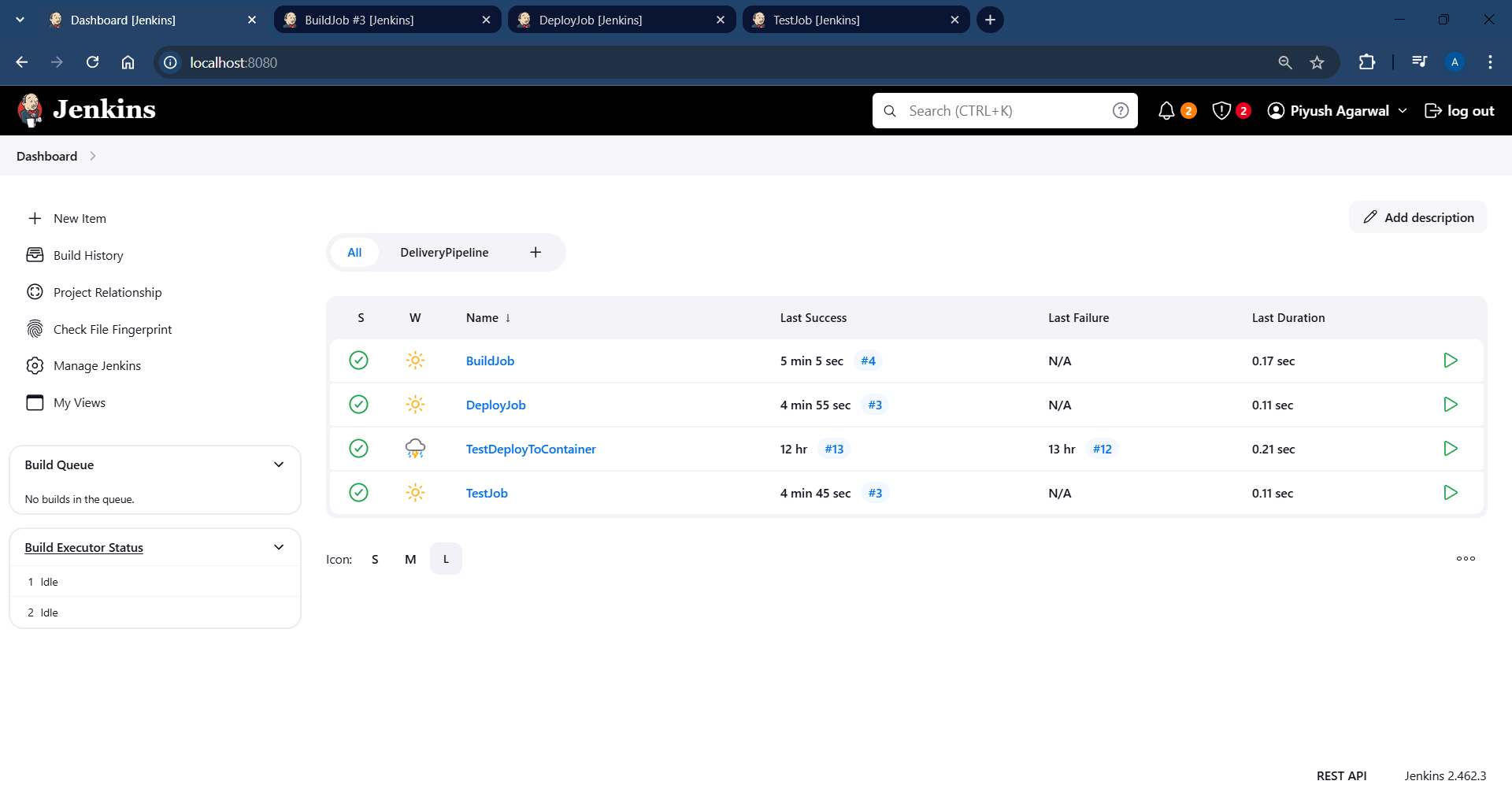
**Problem Statement:** Use Jenkins “Deploy to Container Plugin” and “Build Pipeline Plugin”.

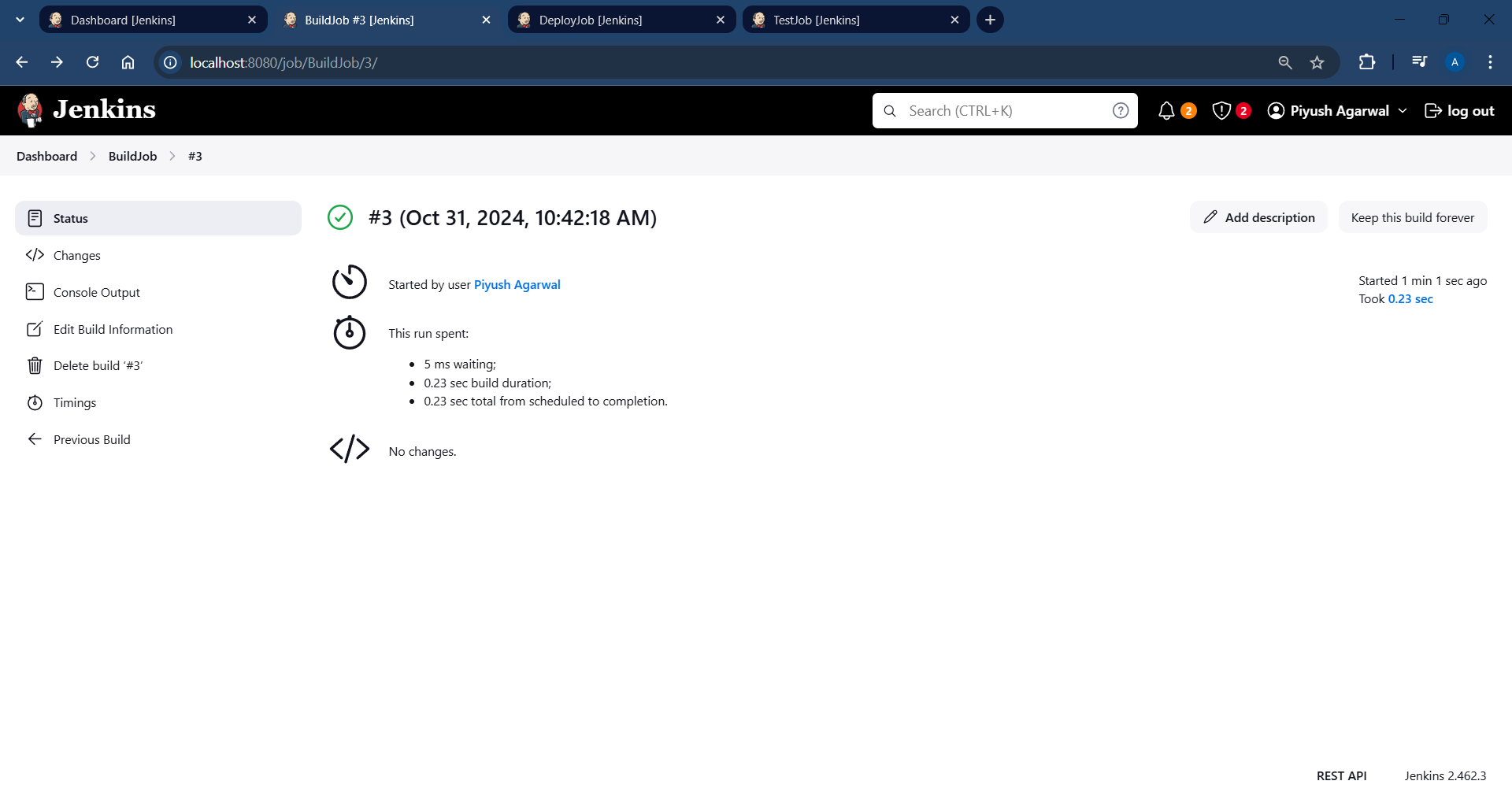
In Continuous Integration/Continuous Deployment (CI/CD), Jenkins plays a crucial role in automating the software development process by building, testing, and deploying applications. The Deploy to Container Plugin and Build Pipeline Plugin are commonly used in Jenkins to streamline deployment processes and visualise the build stages in a pipeline format.

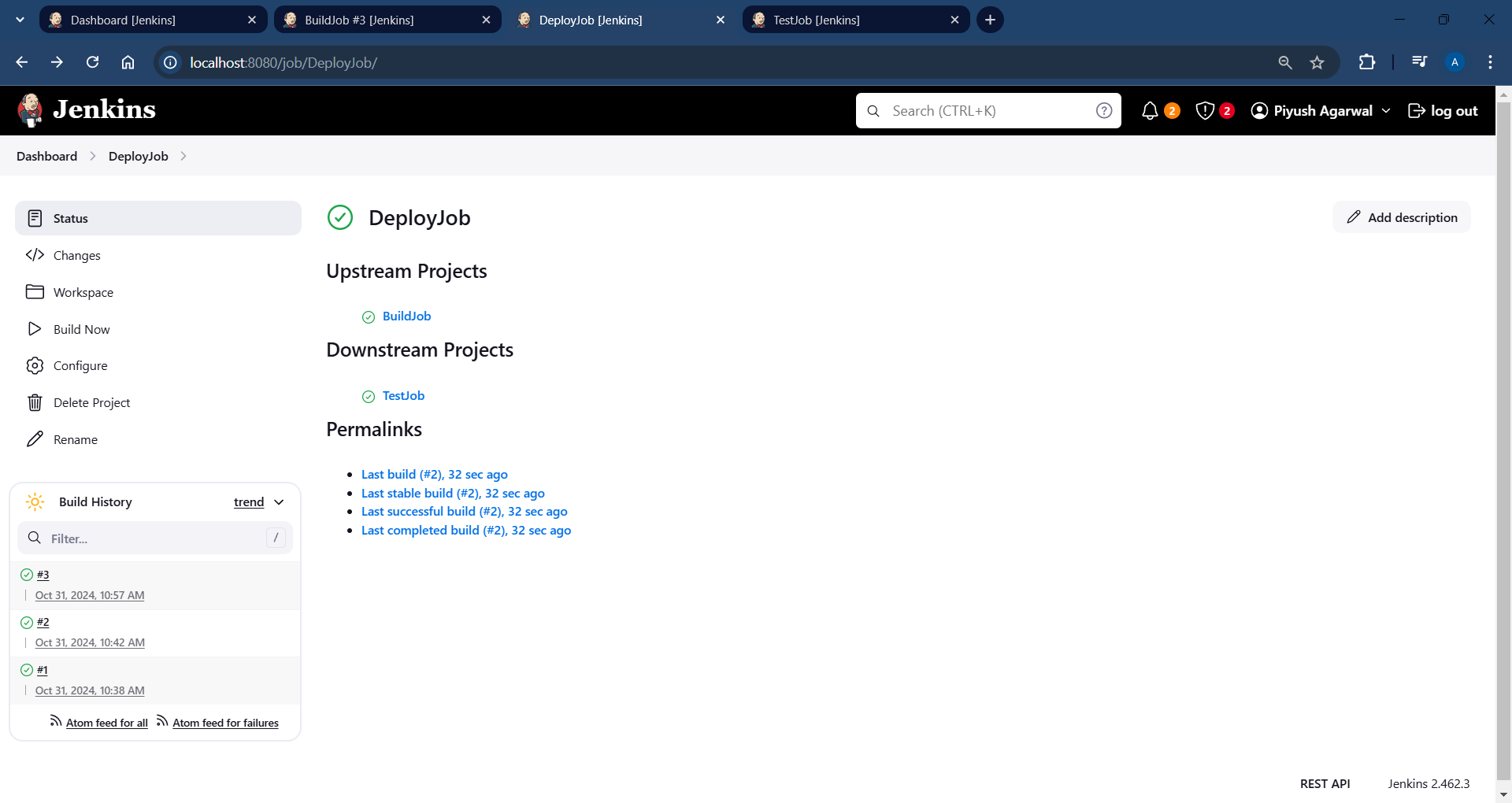
1. Deploy to Container Plugin: This plugin allows Jenkins to automate the deployment of applications to various containerized environments like Tomcat, JBoss, etc.
2. Build Pipeline Plugin: This plugin enables visualisation of a sequence of jobs in a pipeline view, which helps in understanding the flow of stages from code build to deployment.

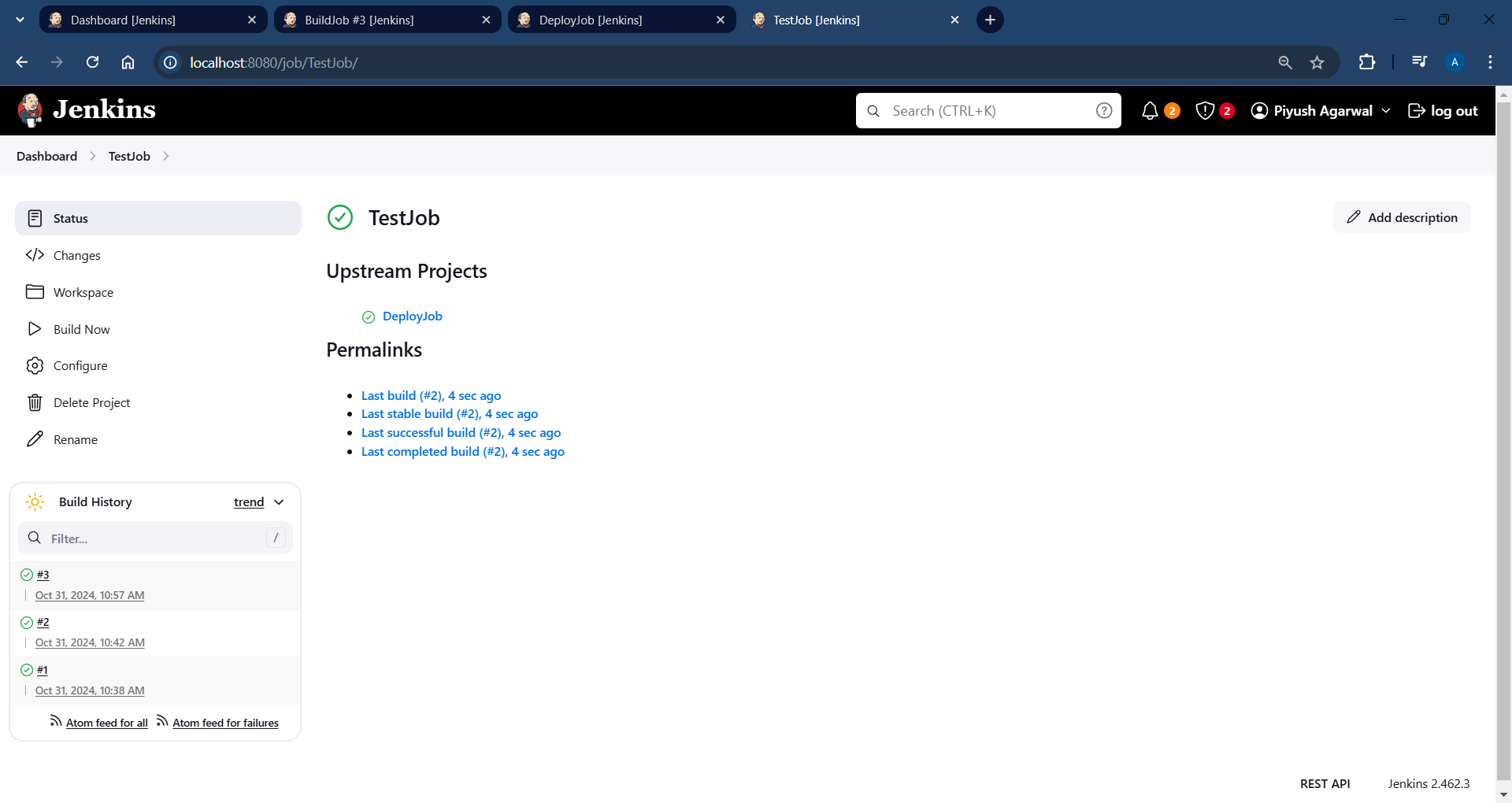
**Create and chain jobs in sequence:**

BuildJob => DeployJob => TestJob

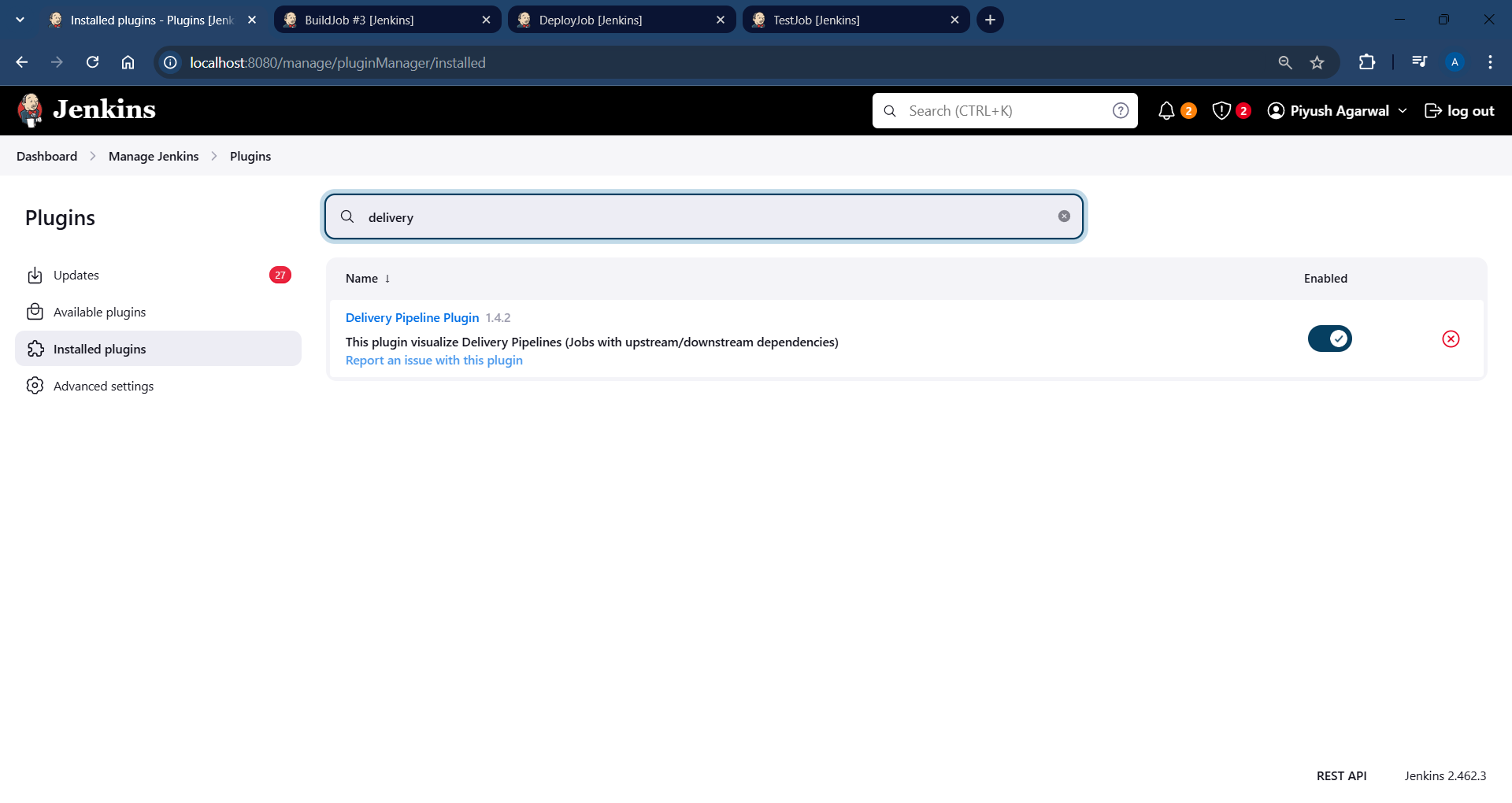




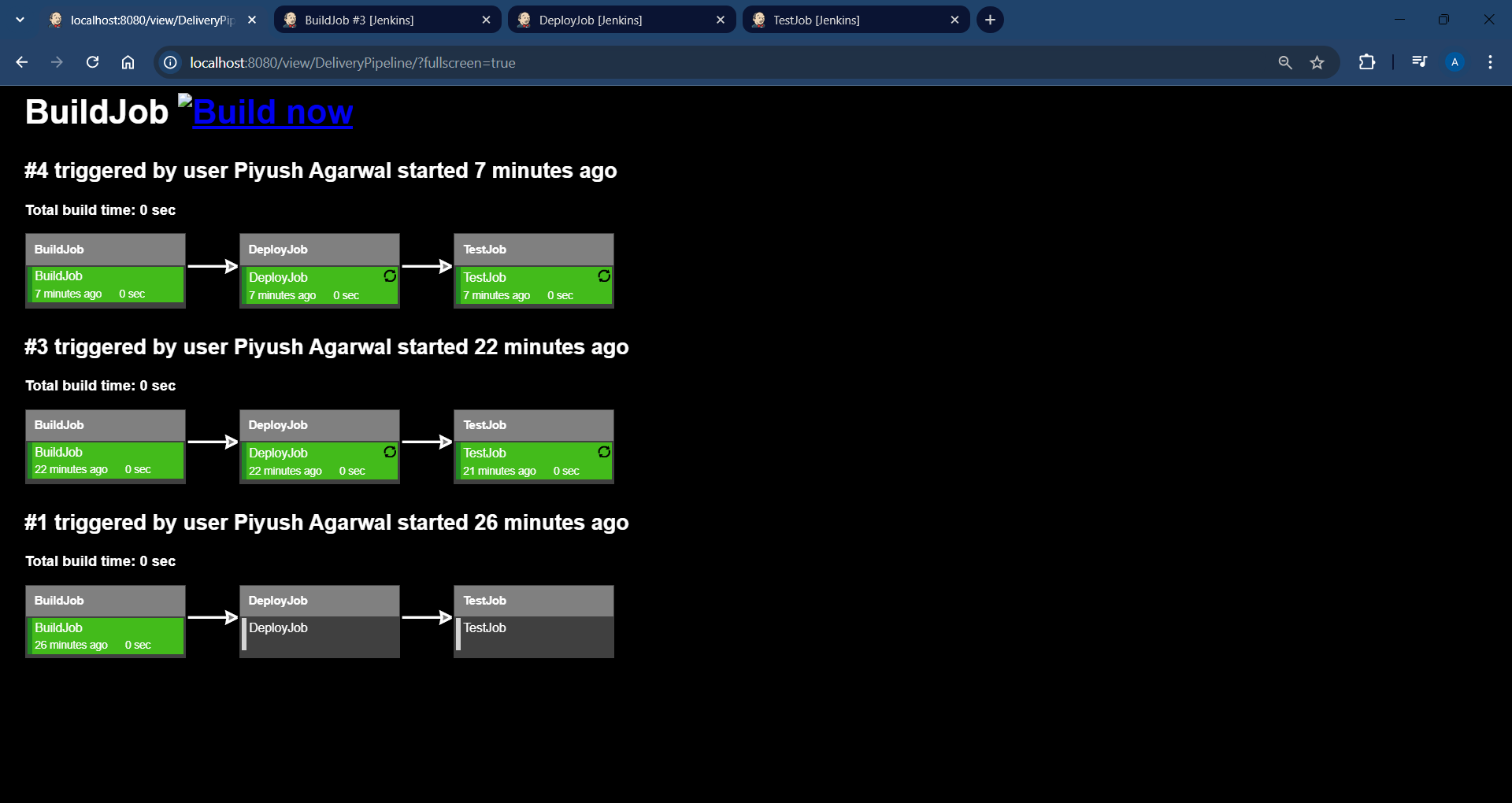




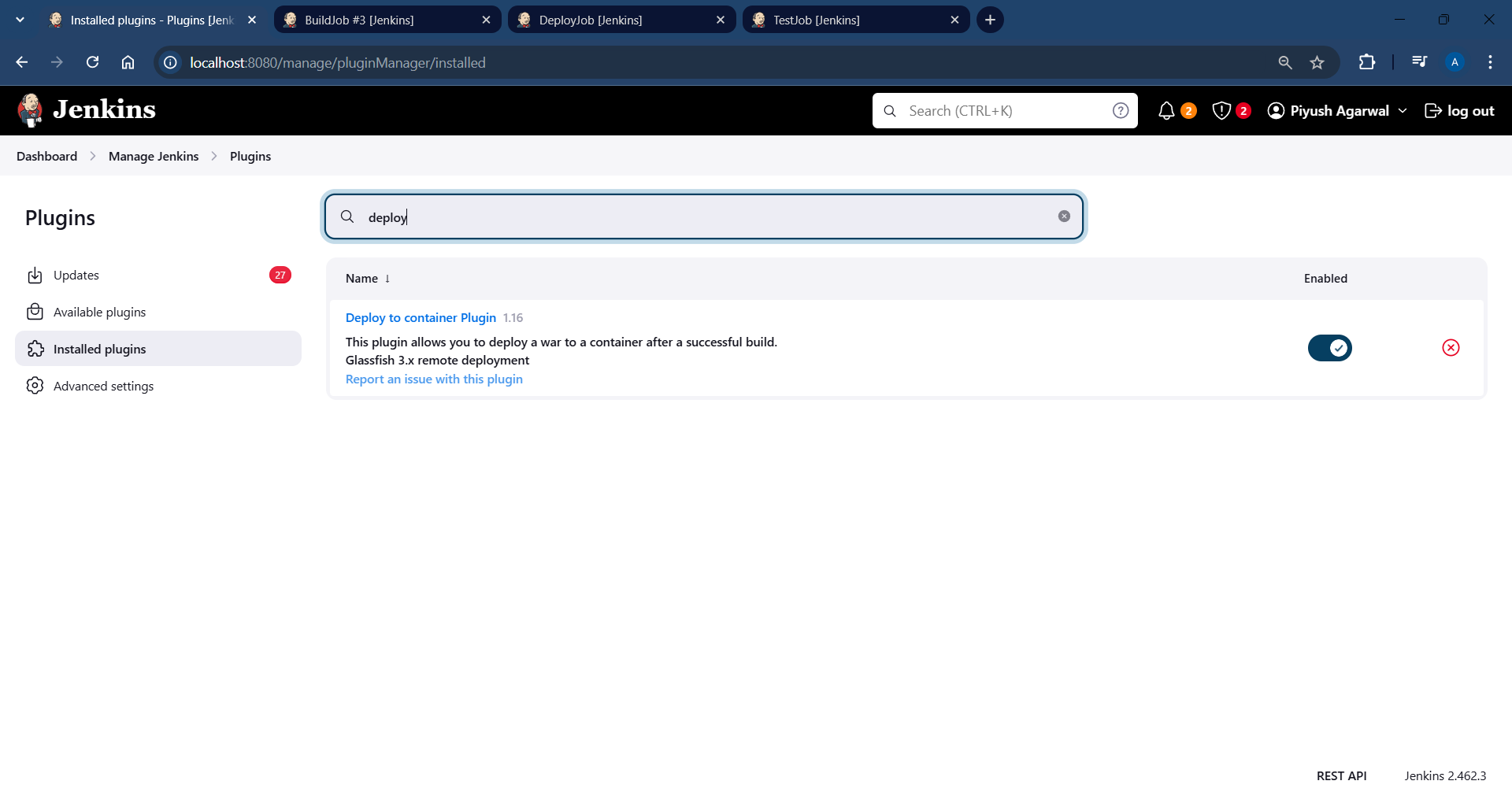
**Install Delivery Pipeline Plugin:**

****

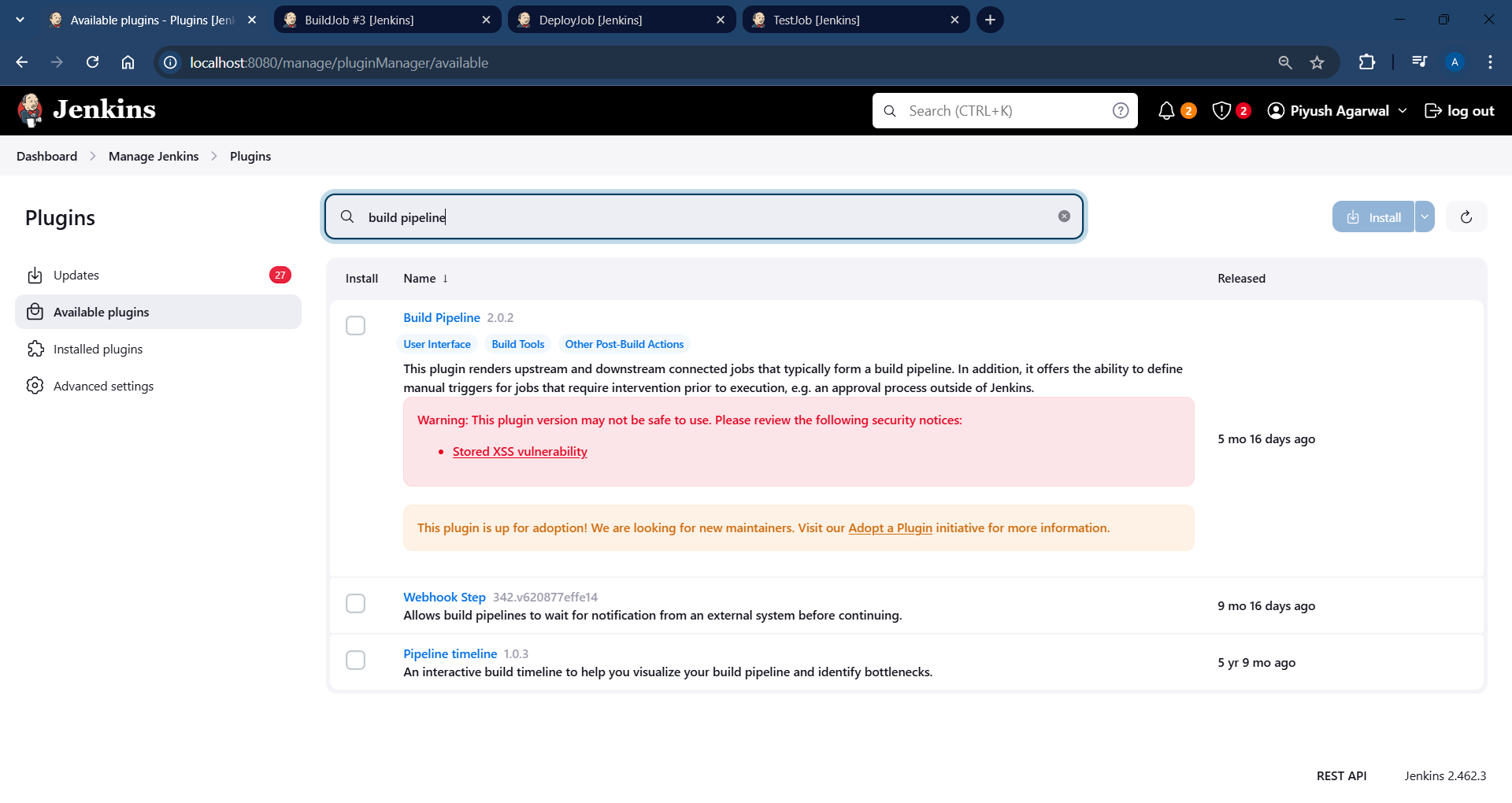
**Test the chained jobs and result it in delivery pipeline view using delivery pipeline plugin**:



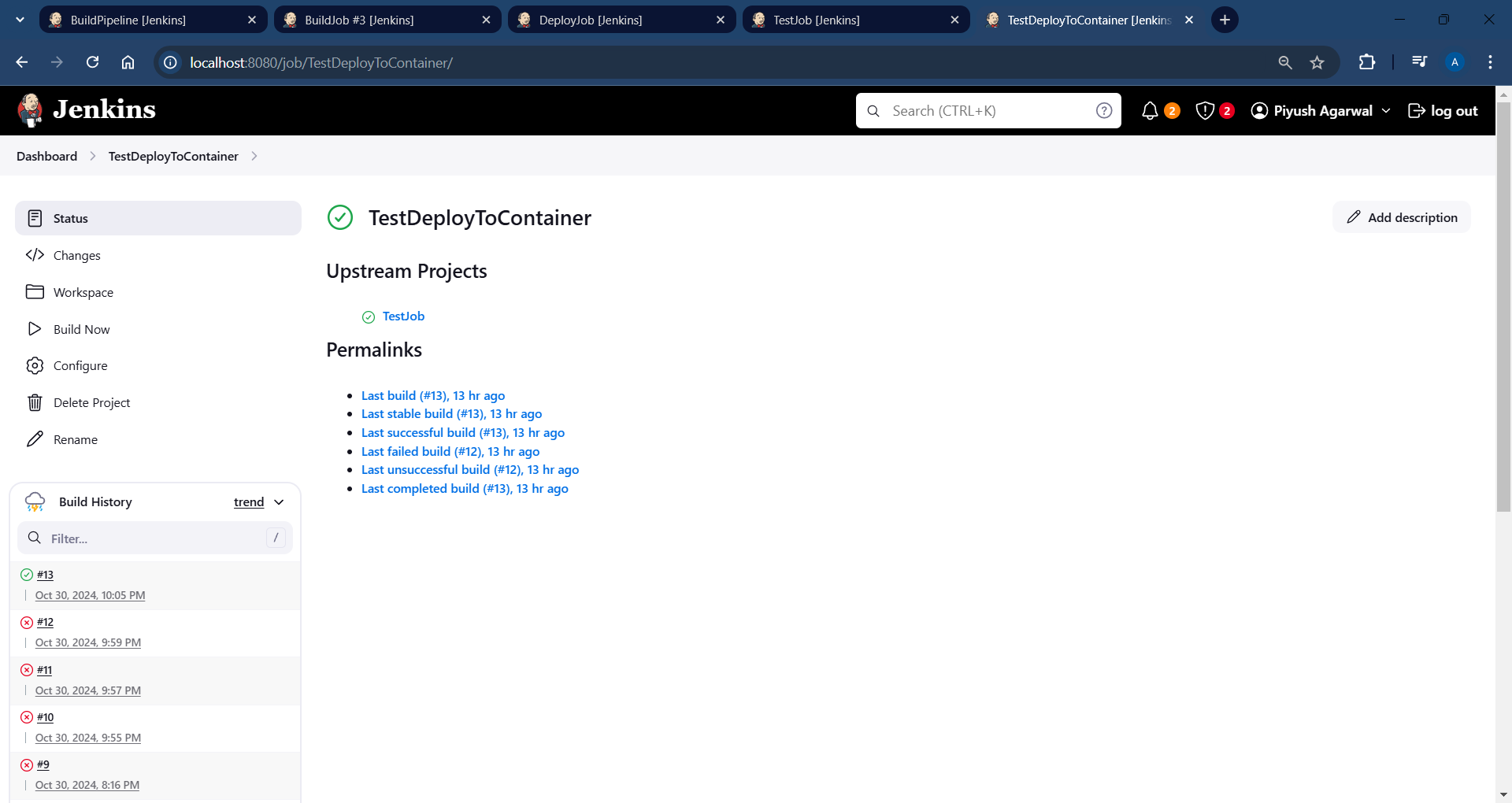
**Install Deploy to Container plugin:**

****

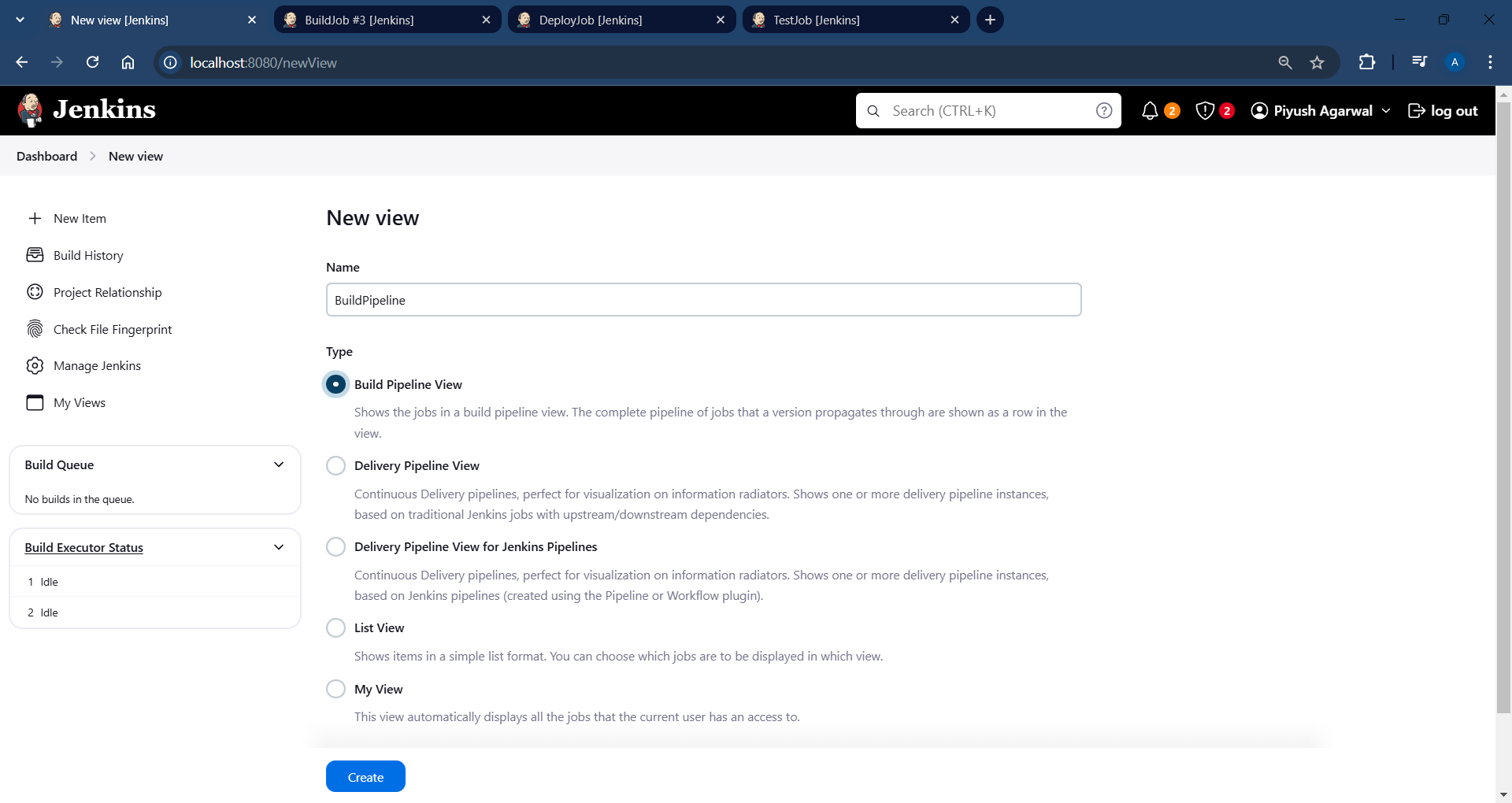
**Install Build Pipeline Plugin:**

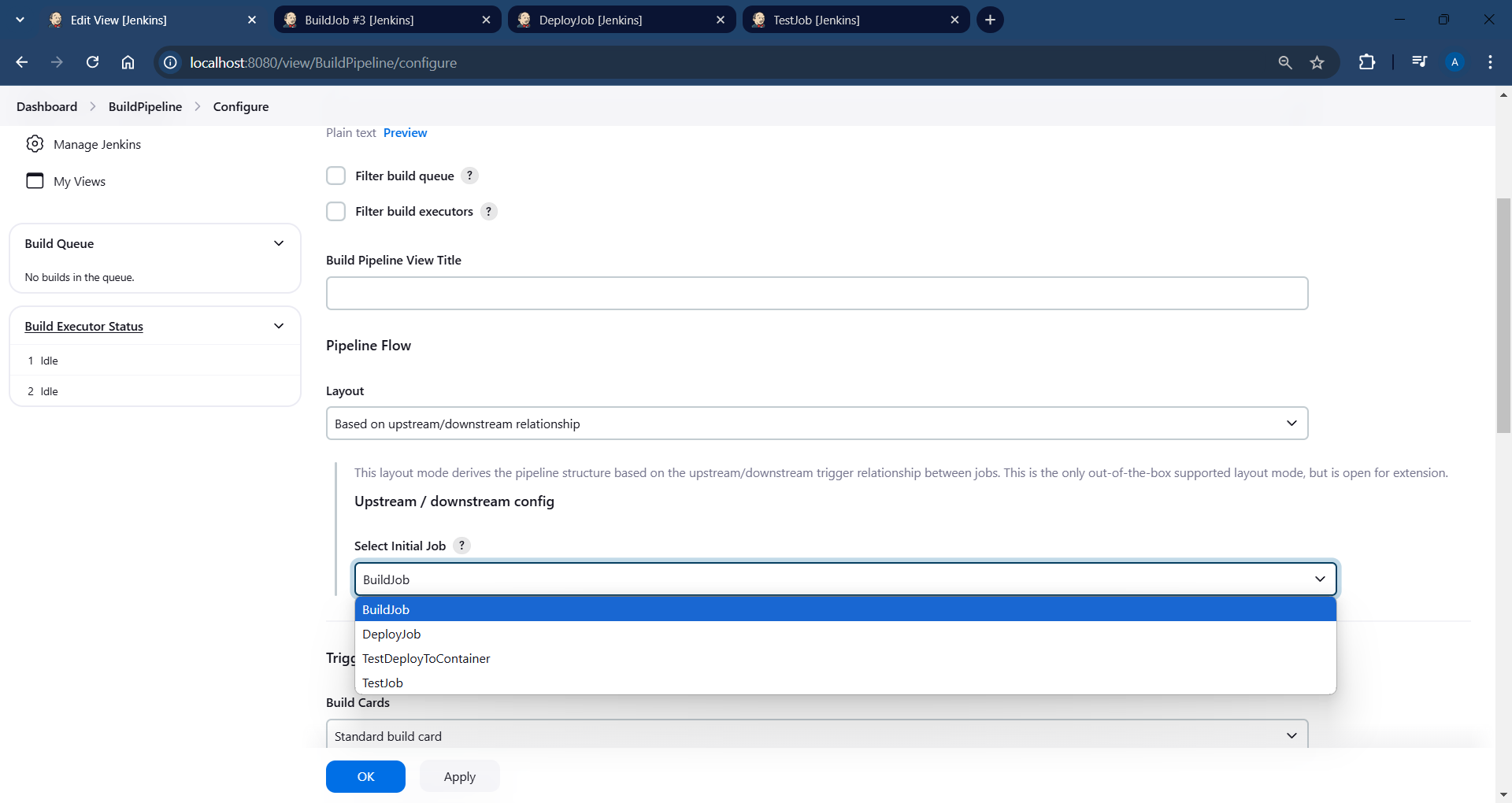
****

**Add TestDeployToContainer job to the chain:**

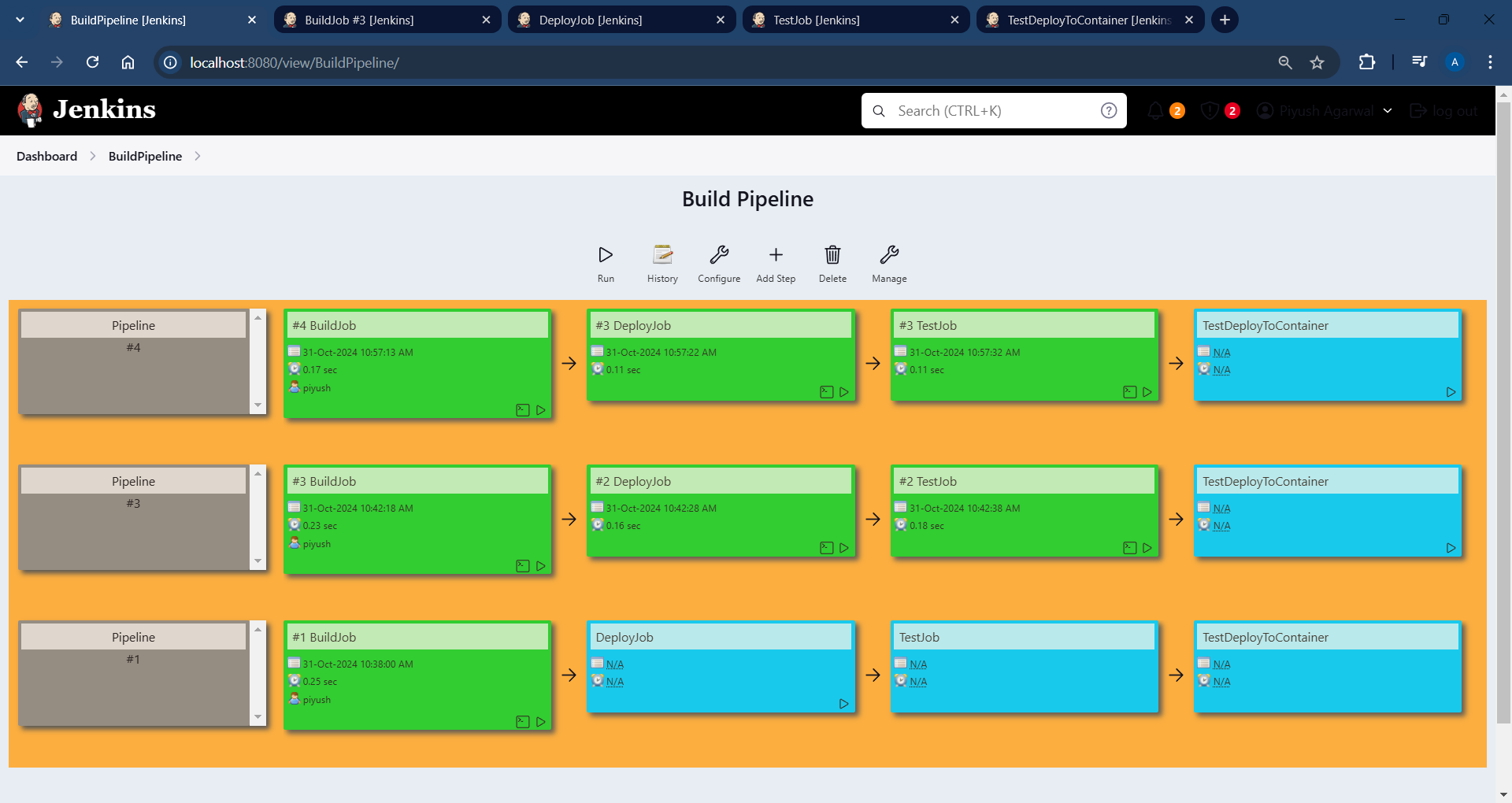
****

**Add Build Pipeline View and configure the view:**

****

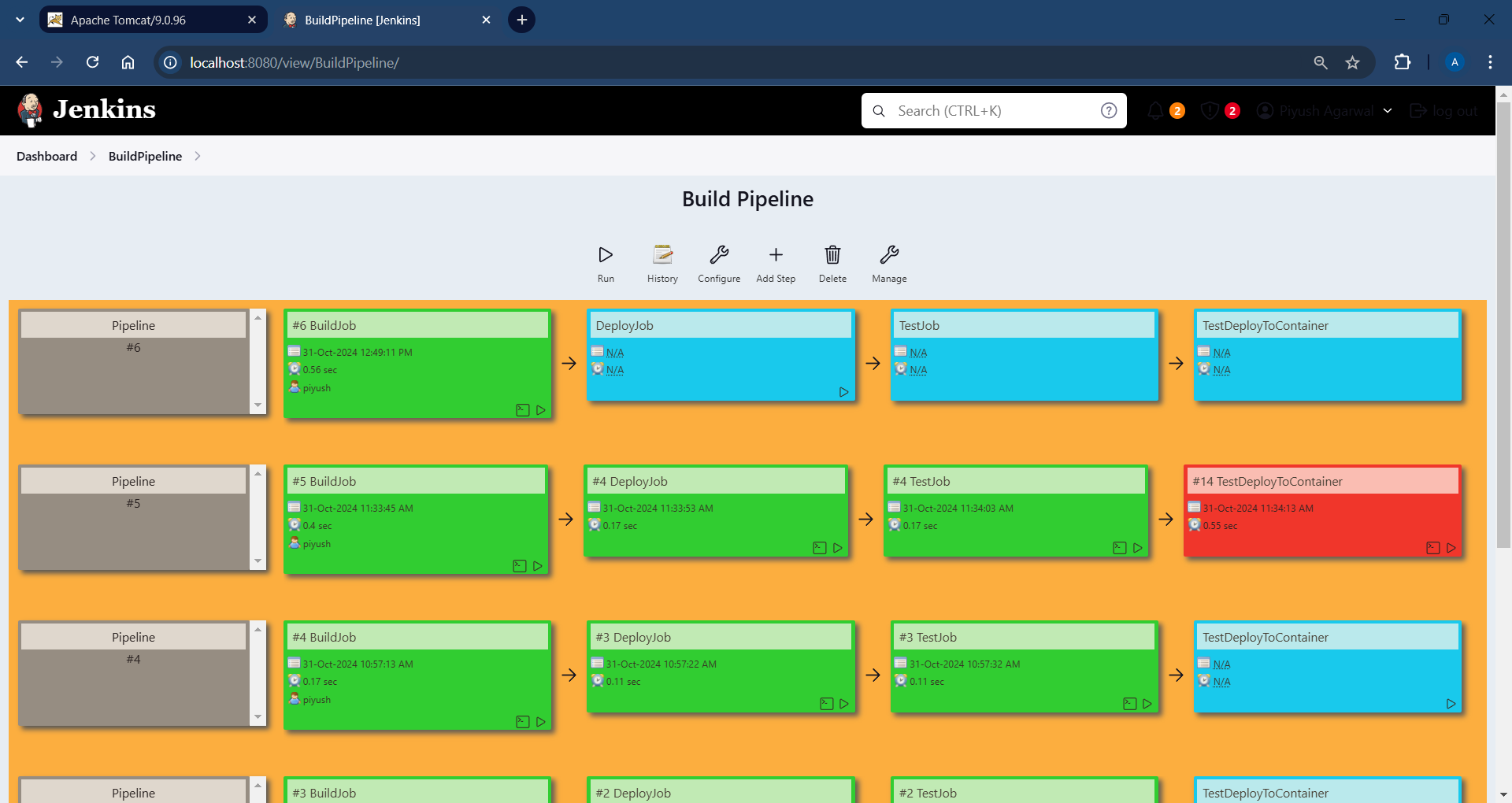
****

**Build Pipeline View of initial runs:**

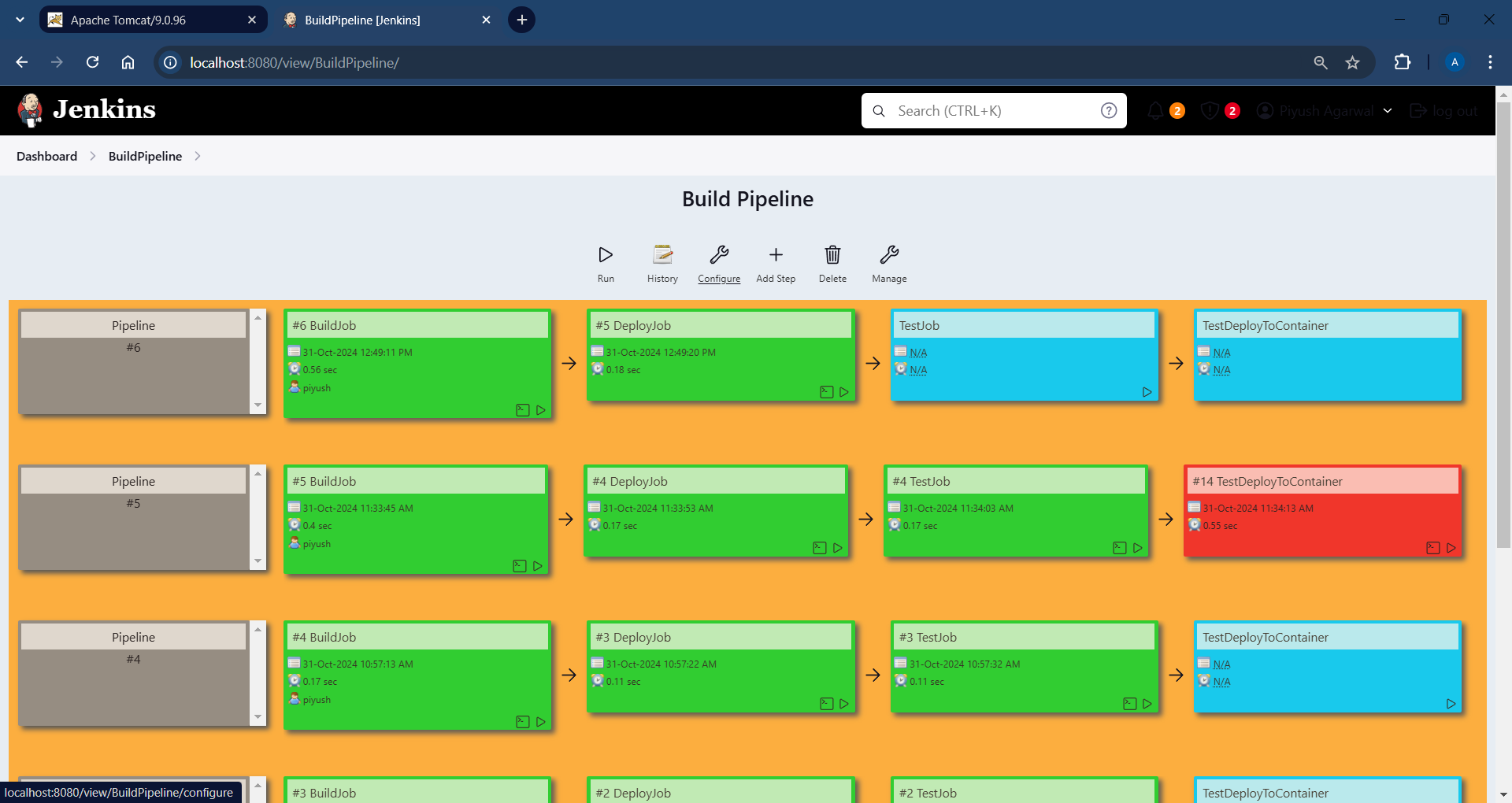
****

**Stages in Build Pipeline after running a new build:**

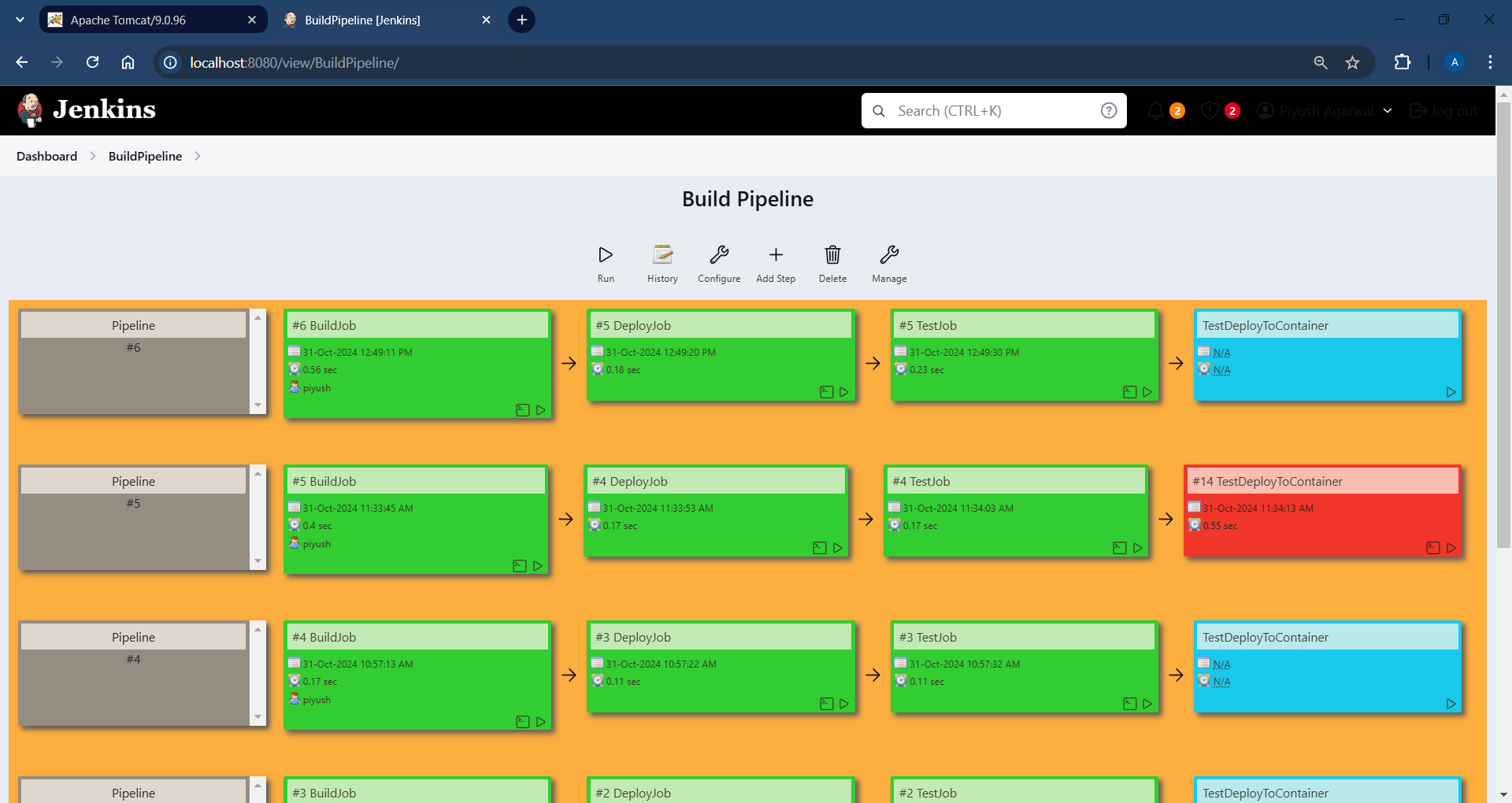
**1.**

****

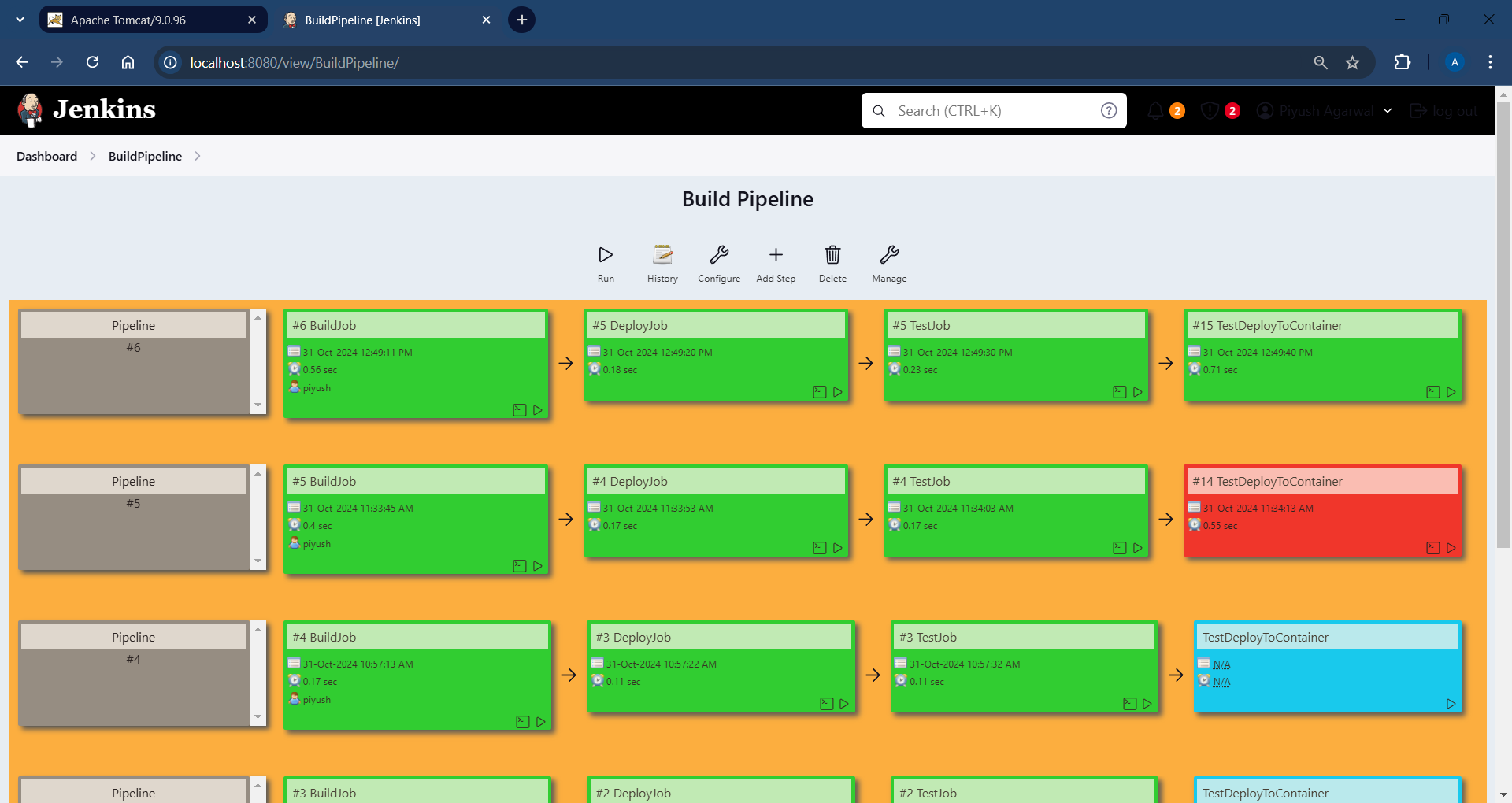
**2.**

****

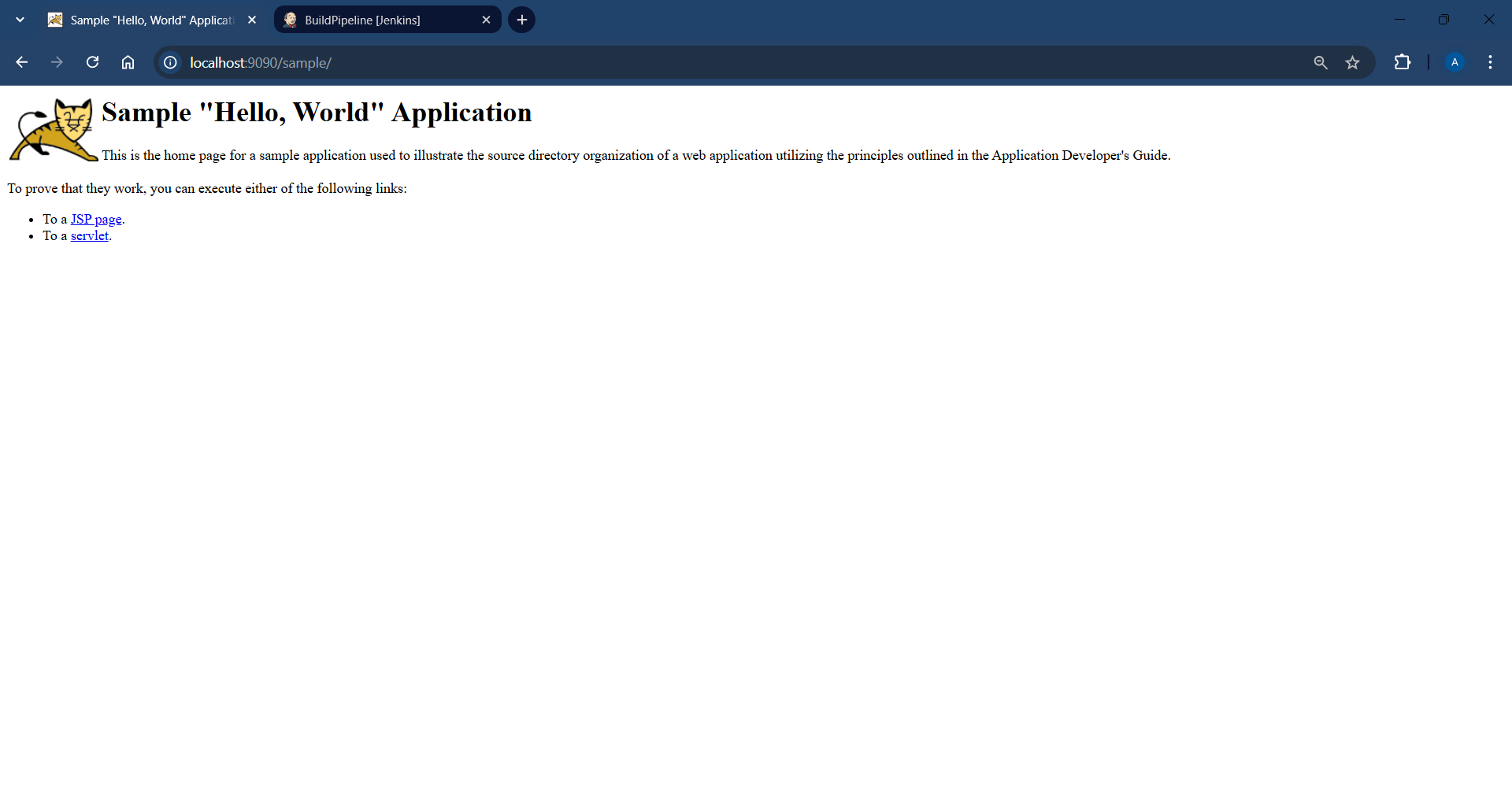
**3.**

****

**4.**

****

**Tomcat server screenshot after last deploy to container job is completed:**

****

**Conclusion:**

Using the Build Pipeline Plugin alongside the Deploy to Container Plugin in Jenkins enables an efficient, automated deployment pipeline. This setup streamlines the process of moving code from build to deployment, with clear visual feedback through the pipeline view. By chaining multiple jobs together, including the deployment job from the previous assignment, this setup effectively demonstrates how Jenkins can be used to automate and visualise each stage of CI/CD. In practice, this approach enhances software delivery by providing transparency, improving deployment speed, and reducing manual errors, making it a robust solution for continuous deployment scenarios.