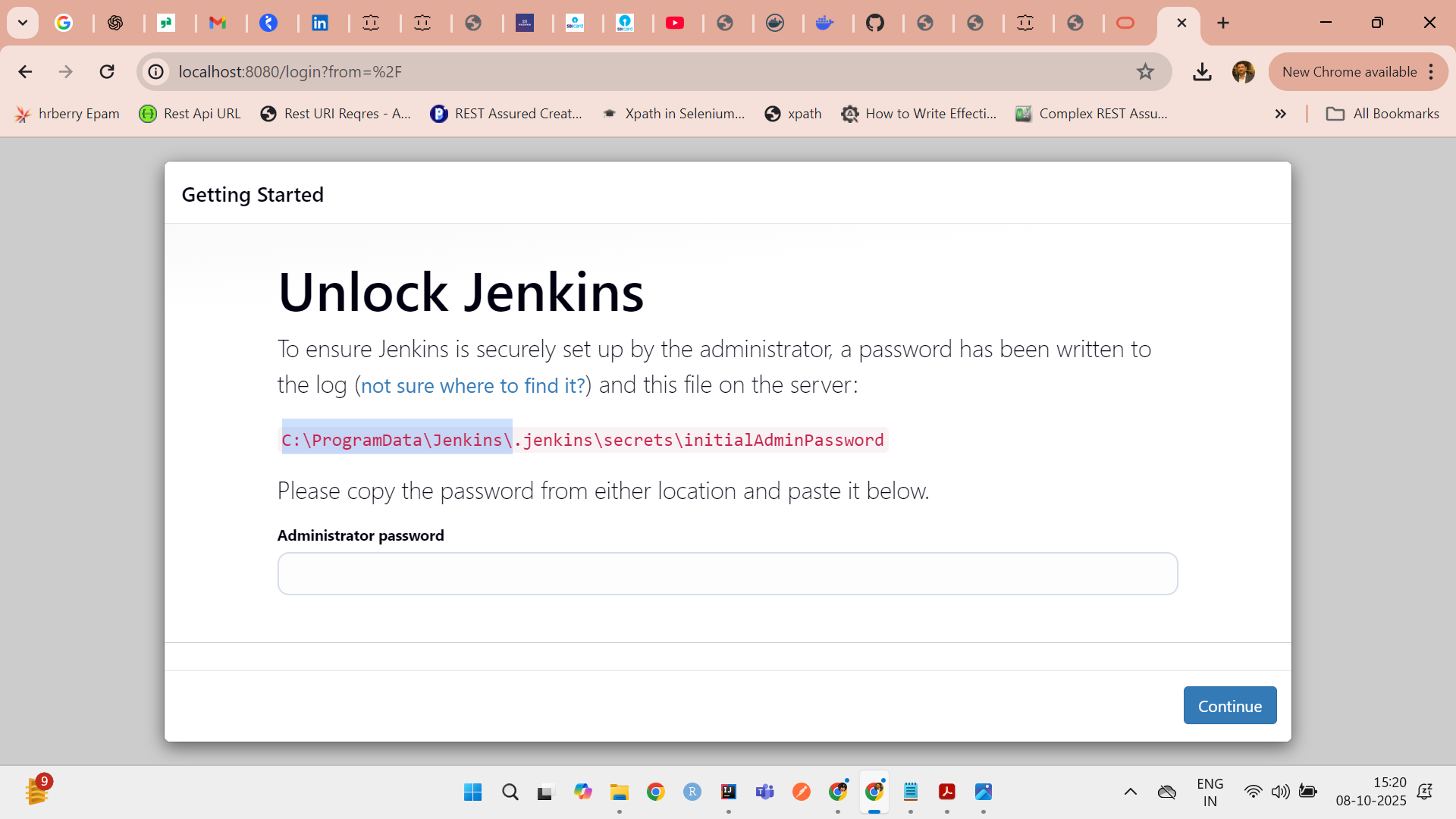
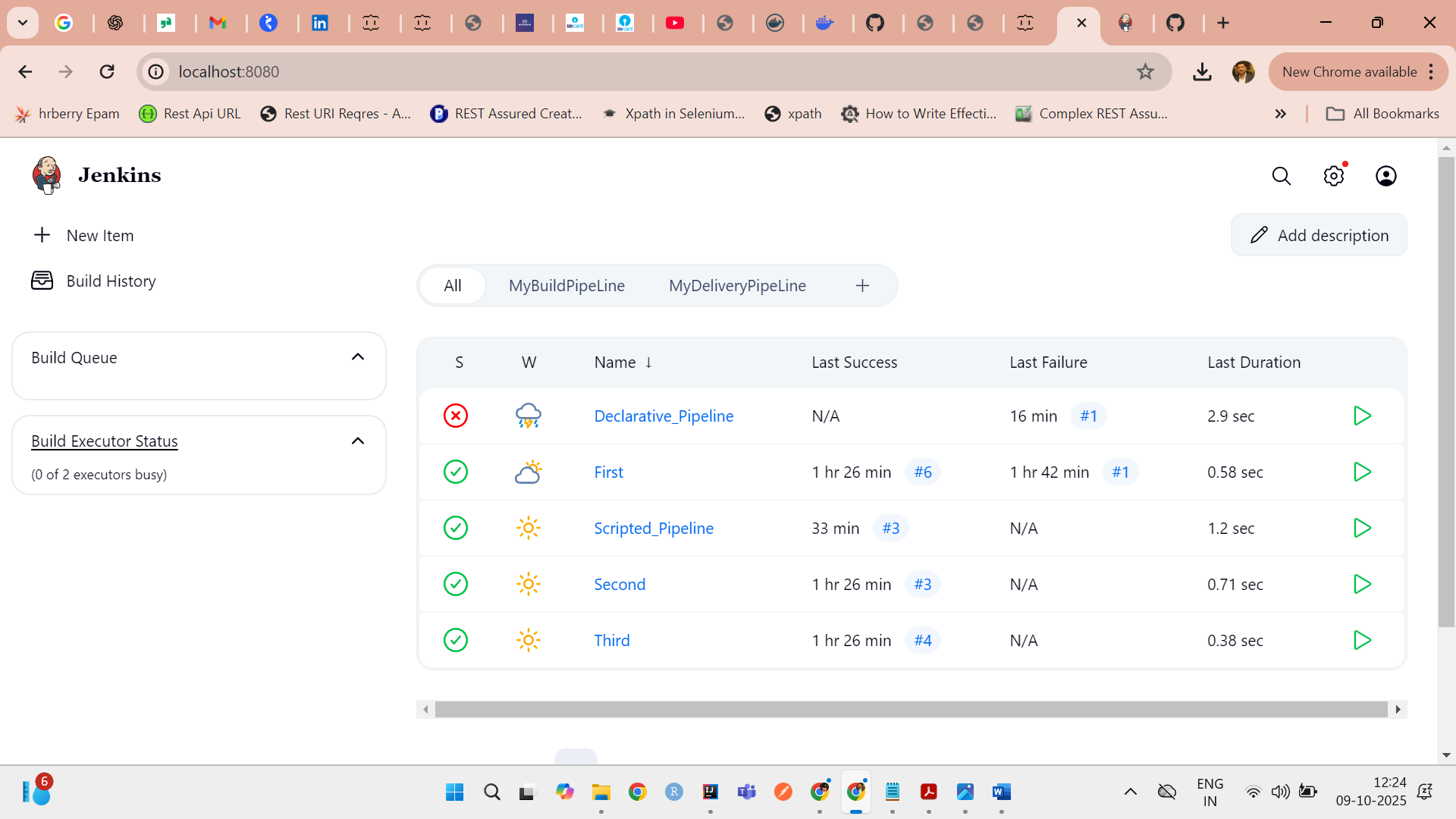
[Download ChromeDriver - Latest Version | ChromeDriver Installe](https://www.chromedriverdownload.net/?utm_source=chatgpt.com)Always runs on localhost  
http://localhost:8080/  
C:\ProgramData\Jenkins\.jenkins\secrets\initialAdminPassword  
amitgoyal/goyal  
9b7159c2d8094b238923b2c8ed4bfe53  
  
  
  
scripted-on Jenkins itself  
Declarative on Jenkins file in project  
  
  
**What Is a Jenkins Pipeline?**

A **Jenkins Pipeline** is a way to define your build process as code. There are **two main types**:

1. **Declarative Pipeline** ✅ *(recommended for most users)*
2. **Scripted Pipeline** ⚙️ *(more flexible, but complex)*

**🆚 Declarative vs Scripted Pipeline**

| **Feature** | **Declarative Pipeline** | **Scripted Pipeline** |
| --- | --- | --- |
| **Syntax style** | Structured, predefined keywords | Groovy code (flexible scripting) |
| **Ease of use** | Easy to read and write | More complex and verbose |
| **Error handling** | Built-in (post {} blocks) | Must handle manually (try-catch) |
| **Recommended for** | Most users and teams | Advanced use cases only |
| **Introduced in** | Jenkins 2.x+ | Jenkins 1.x (still supported) |
| **Parallel execution** | Supported (easy) | Supported (manual & verbose) |
| **Pipeline as code (Jenkinsfile)** | ✅ Yes | ✅ Yes |

**✅ Declarative Pipeline Example**

pipeline {

agent any

stages {

stage('Build') {

steps {

echo 'Building...'

// run Maven

bat 'mvn clean install'

}

}

stage('Test') {

steps {

echo 'Testing...'

bat 'mvn test'

}

}

}

post {

success {

echo 'Build succeeded!'

}

failure {

echo 'Build failed.'

}

}

}

**👍 Pros:**

* Easy to understand
* Enforces best practices
* Has post, when, environment, etc.

**⚙️ Scripted Pipeline Example**

node {

try {

stage('Build') {

echo 'Building...'

bat 'mvn clean install'

}

stage('Test') {

echo 'Testing...'

bat 'mvn test'

}

} catch (err) {

echo "Build failed: ${err}"

currentBuild.result = 'FAILURE'

} finally {

echo 'Cleaning up...'

}

}

**⚠️ Pros:**

* Full access to Groovy features
* More control and flexibility

**👎 Cons:**

* Verbose and harder to maintain
* No built-in syntax validation in Jenkins UI
* Error handling must be written manually  
    
    
    
    
  **Pipeline setup on Jenkins  
  Need to install Java and set Maven path Manulayy**  
    
  