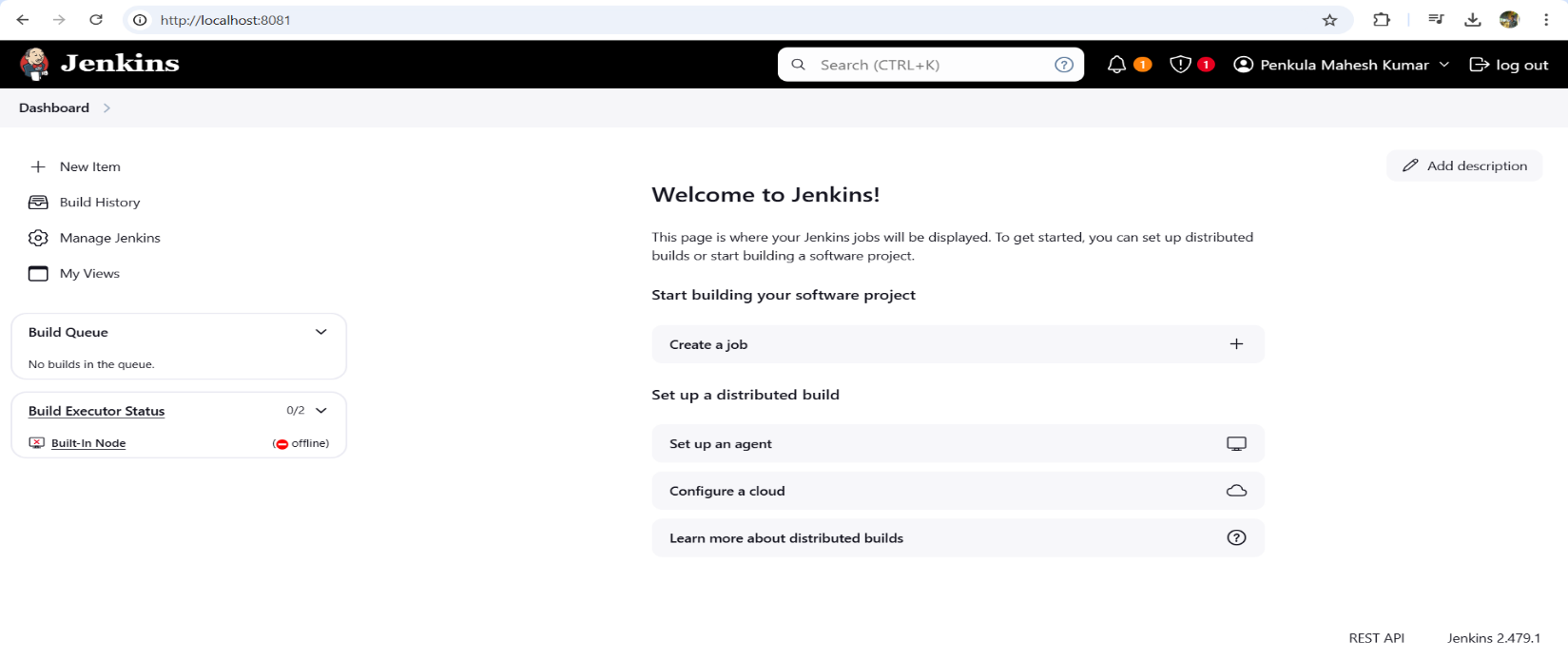
**Jenkins Automation**

**Steps for MavenJava Automation:**

**Step 1: Open Jenkins (localhost:8080)**

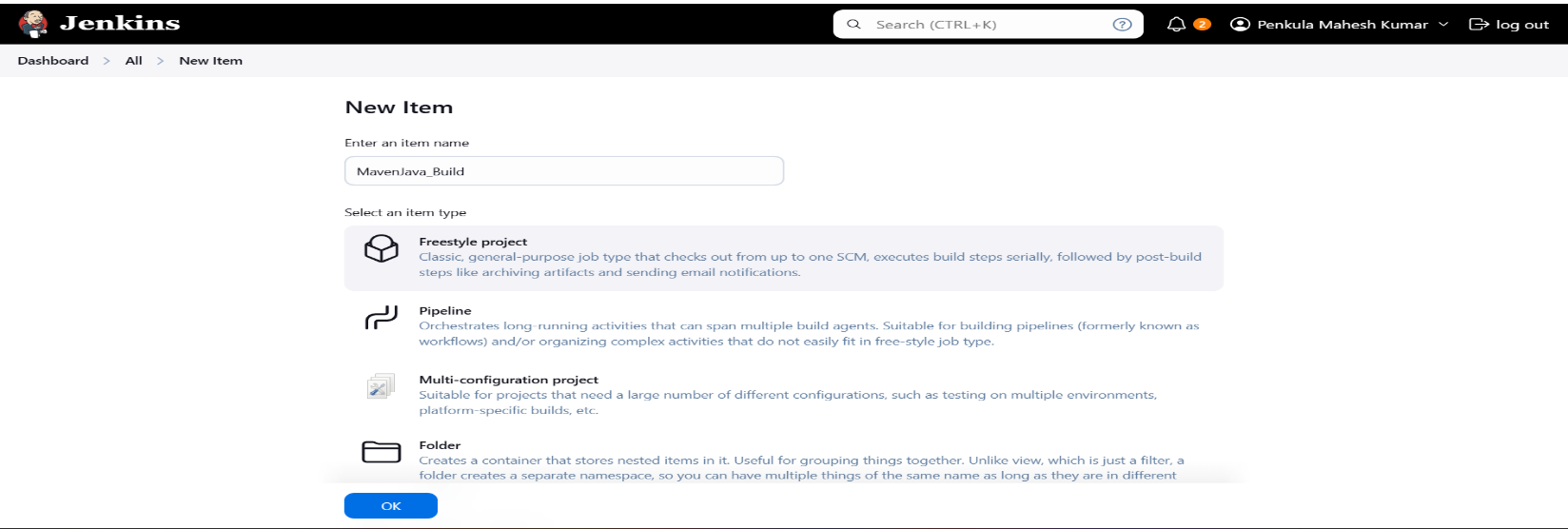
Click on "New Item" (left side menu

****

**Step 2: Create Freestyle Project (e.g., MavenJava\_Build)**

Enter project name (e.g., MavenJava\_Build)

Click "OK"

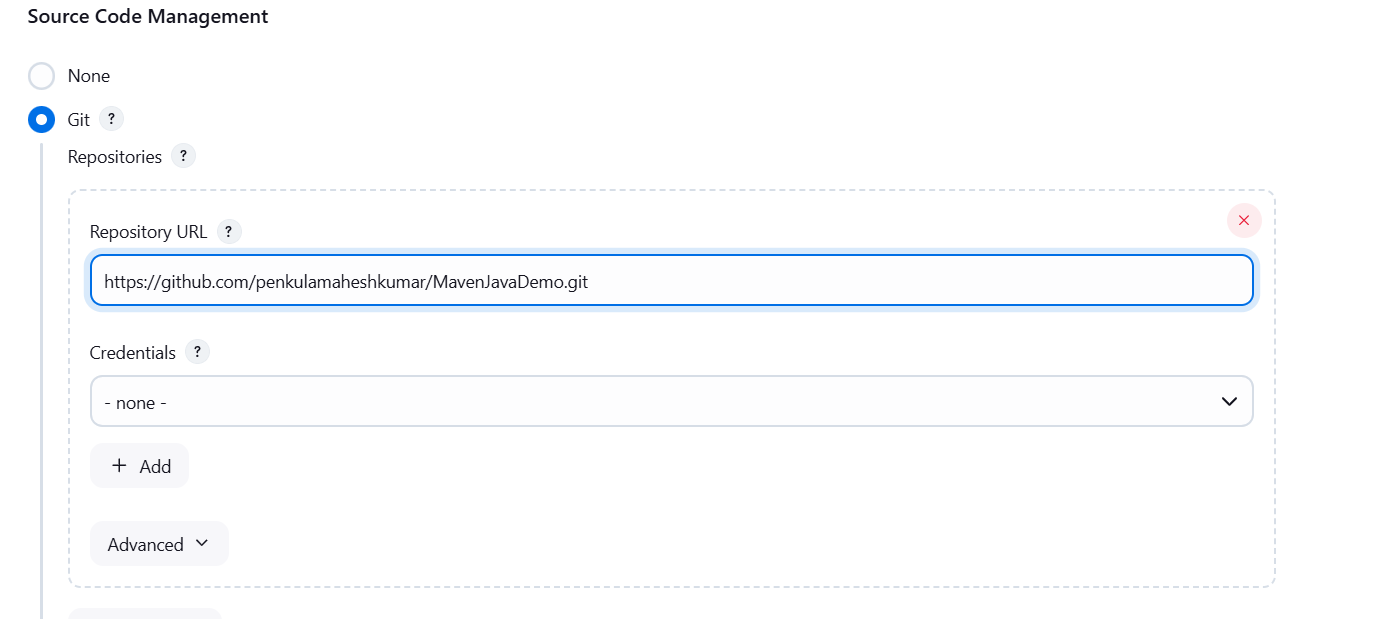


**Configure the project:**

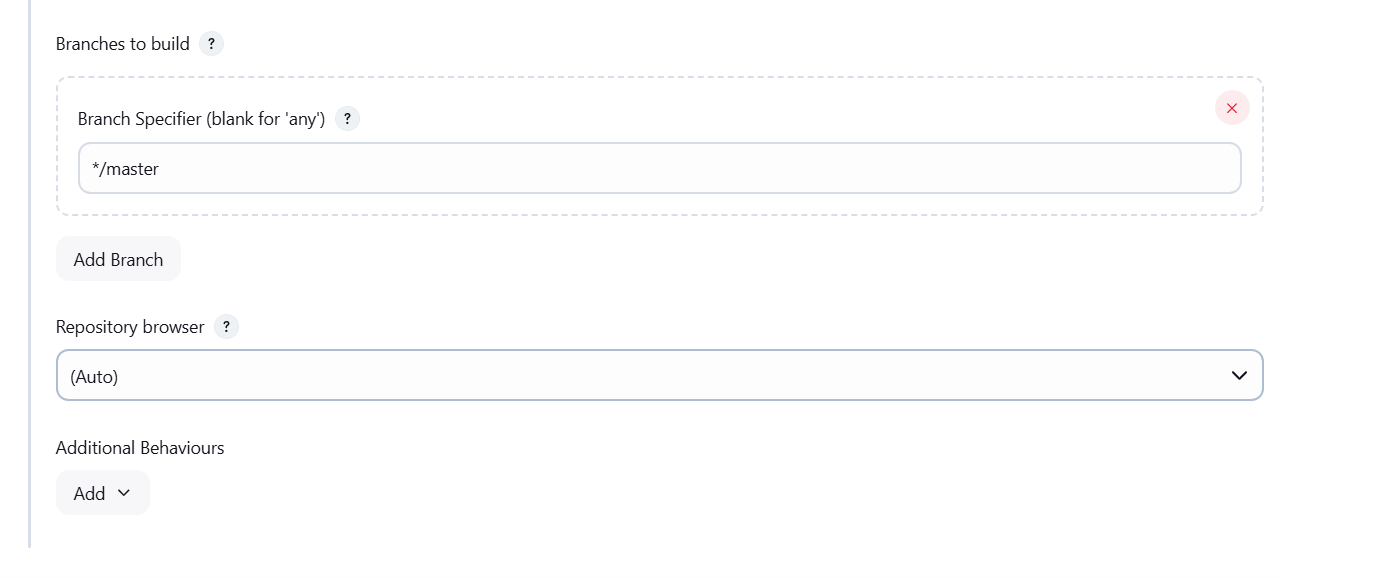
**Description**: "Java Build demo"

**Source Code Management**:

Git repository URL: [GitMavenJava repo URL]



**Branches to build**: \*/Main or \*/master



**Build Steps**:

**Add Build Step** -> "Invoke top-level Maven targets"

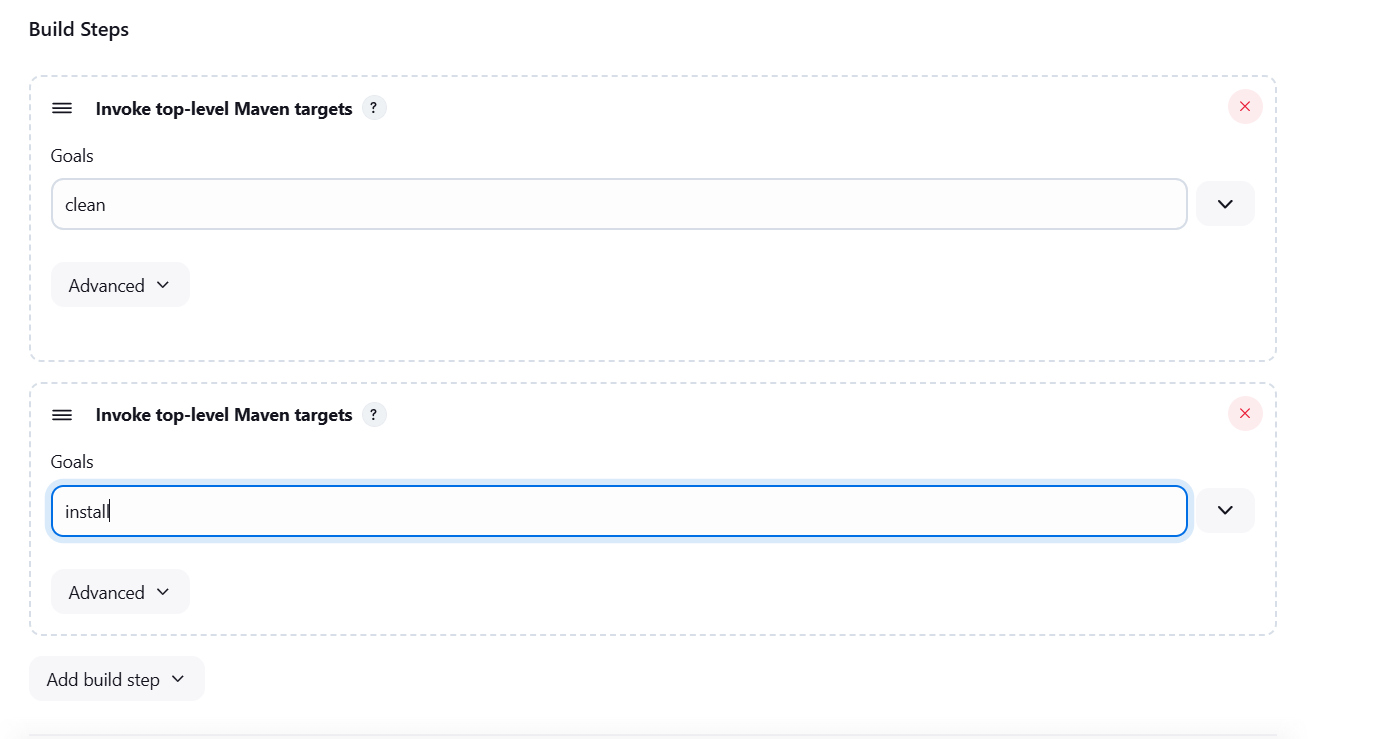
Maven version: MAVEN\_HOME

Goals: clean

**Add Build Step** -> "Invoke top-level Maven targets"

Maven version: MAVEN\_HOME

Goals: install



**Post-build Actions:**

Add Post Build Action -> "Archive the artifacts"

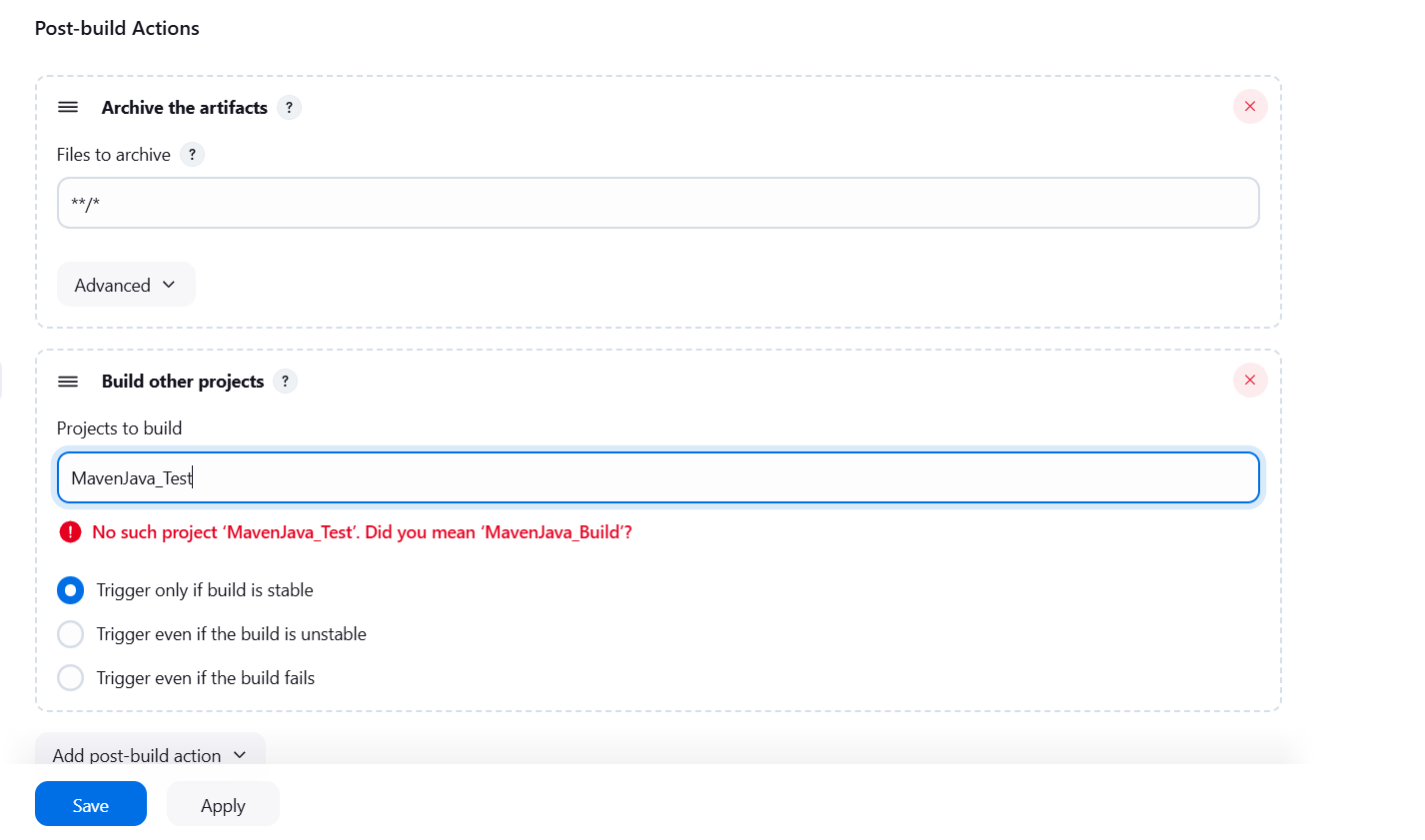
Files to archive: \*\*/\*

Add Post Build Action -> "Build other projects"

Projects to build: MavenJava\_Test

Trigger: Only if build is stable

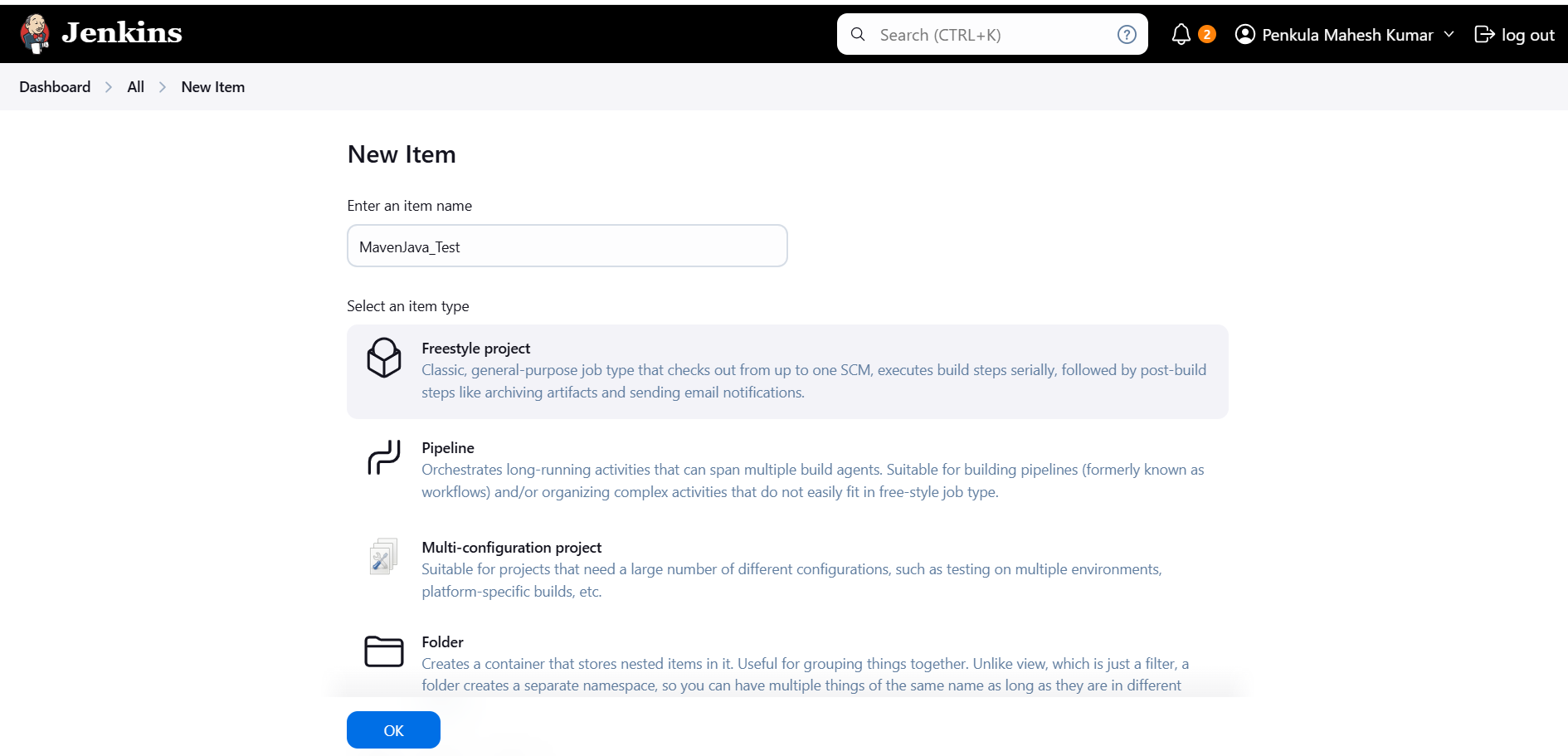
Apply and Save



**Step 3: Create Freestyle Project (e.g., MavenJava\_Test)**

Enter project name (e.g., MavenJava\_Test)

Click "OK"



**Configure the project**:

**Description**: "Test demo"

**Build Environment**:

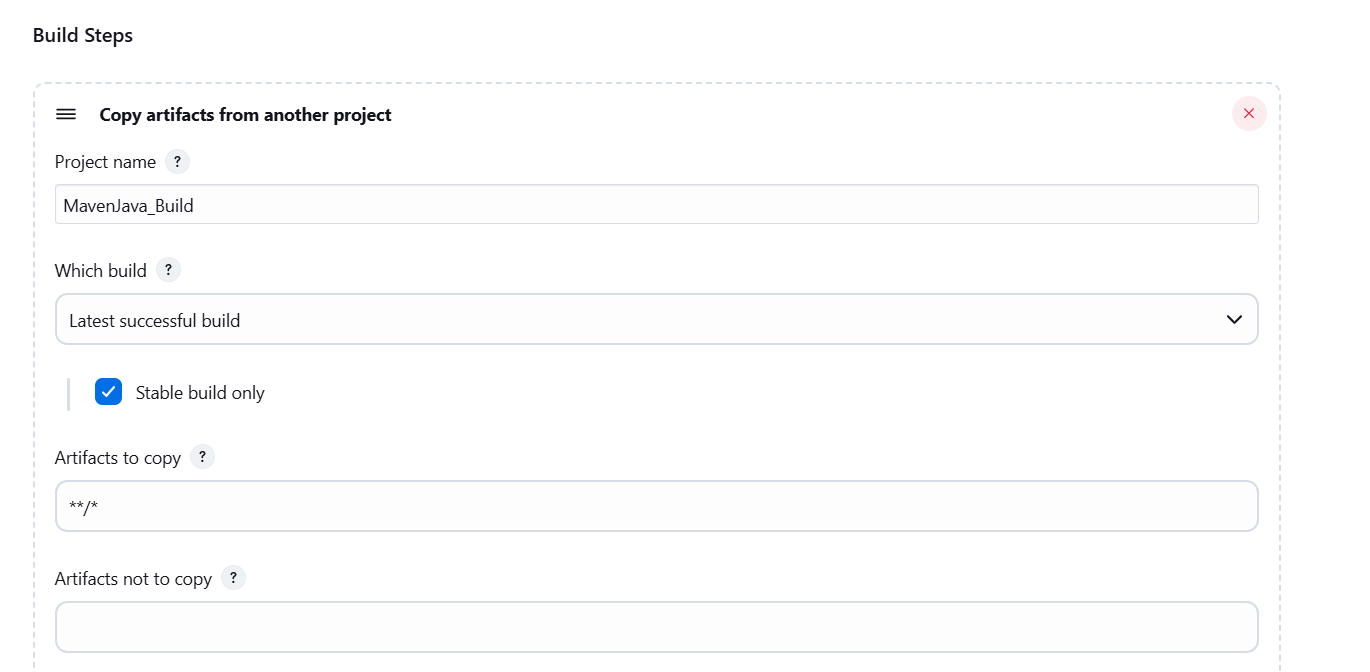
Check: "Delete the workspace before build starts"

**Add Build Step** -> "Copy artifacts from another project"

Project name: MavenJava\_Build

Build: Stable build only **//** tick at this

Artifacts to copy: \*\*/\*



**Add Build Step** -> "Invoke top-level Maven targets"

Maven version: MAVEN\_HOME

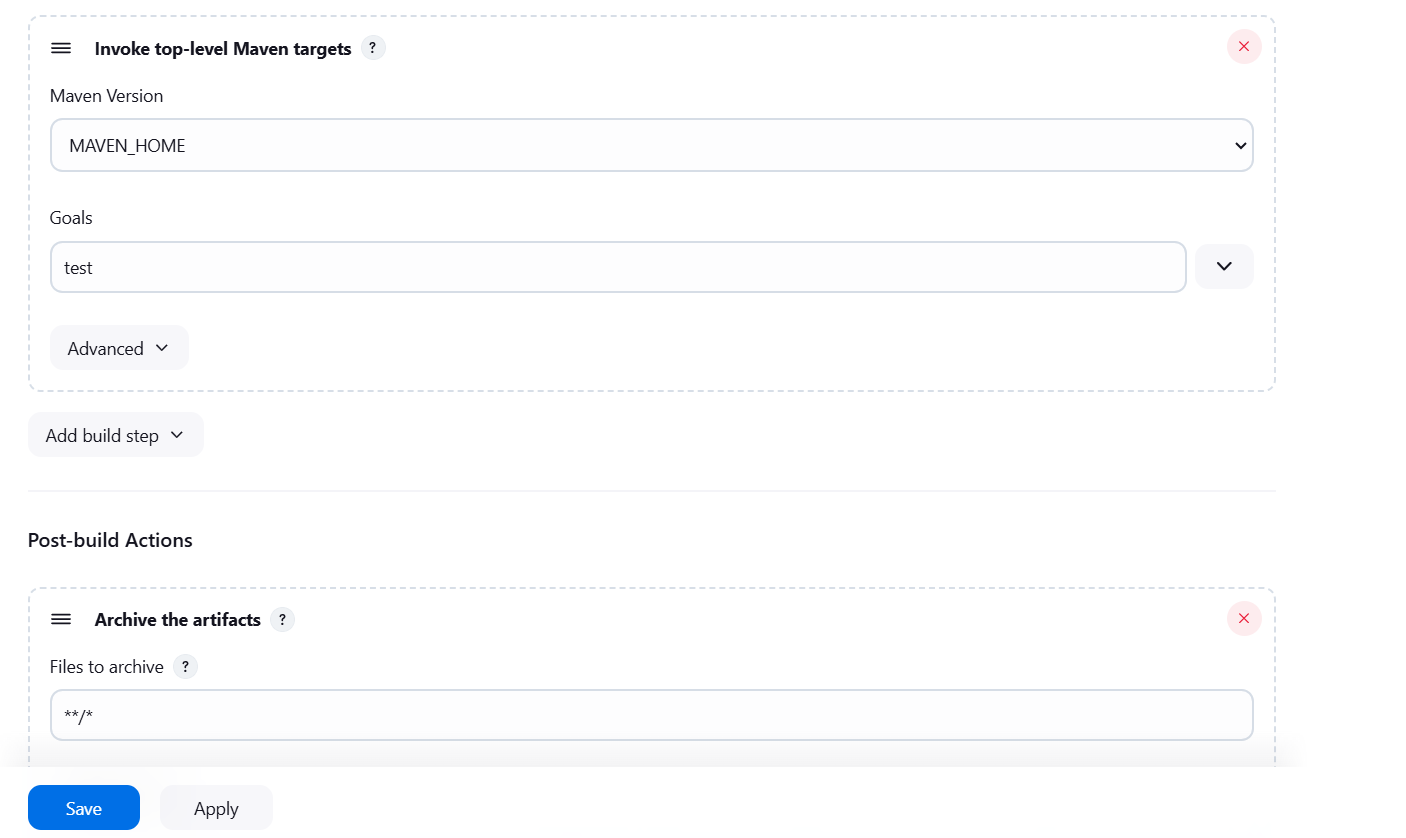
Goals: test

Post-build Actions:

**Add Post Build Action** -> "Archive the artifacts"

Files to archive: \*\*/\*

Apply and Save

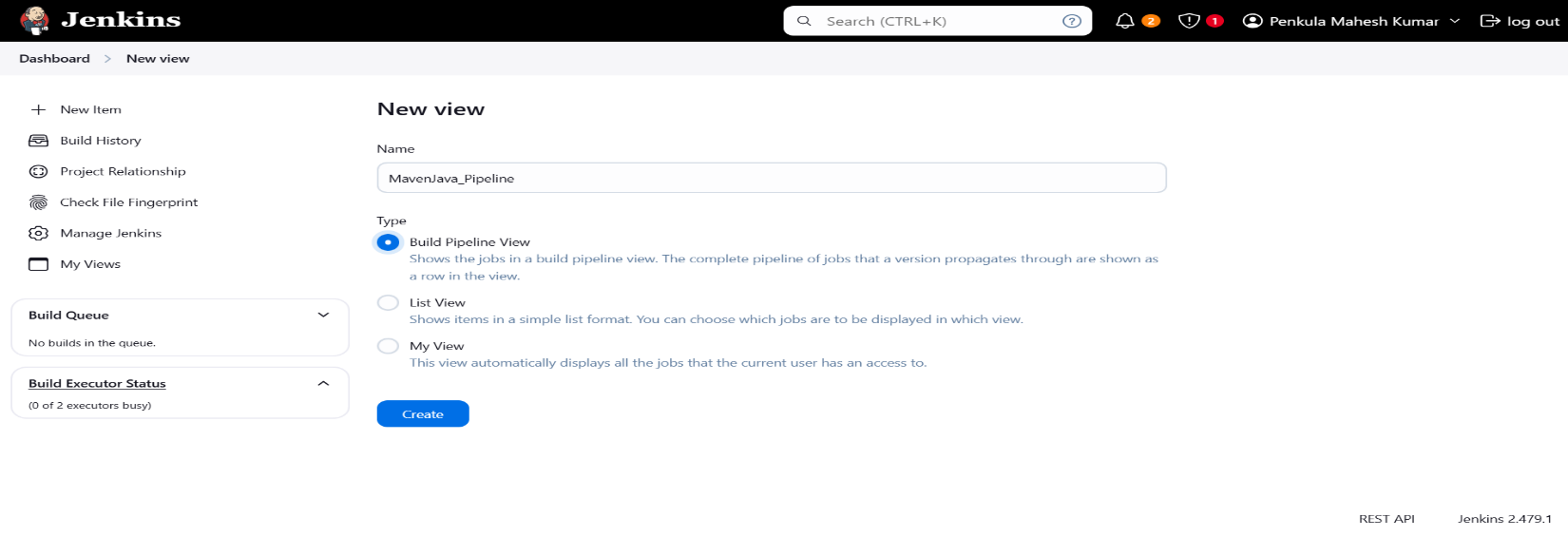


**Step 4: Create Pipeline View for Maven Java project**

Click "+" beside "All" on the dashboard

Enter name: MavenJava\_Pipeline

**Select "**Build pipeline view**" // tick here**

****

**create**

**Pipeline Flow**:

**Layout**: Based on upstream/downstream relationship

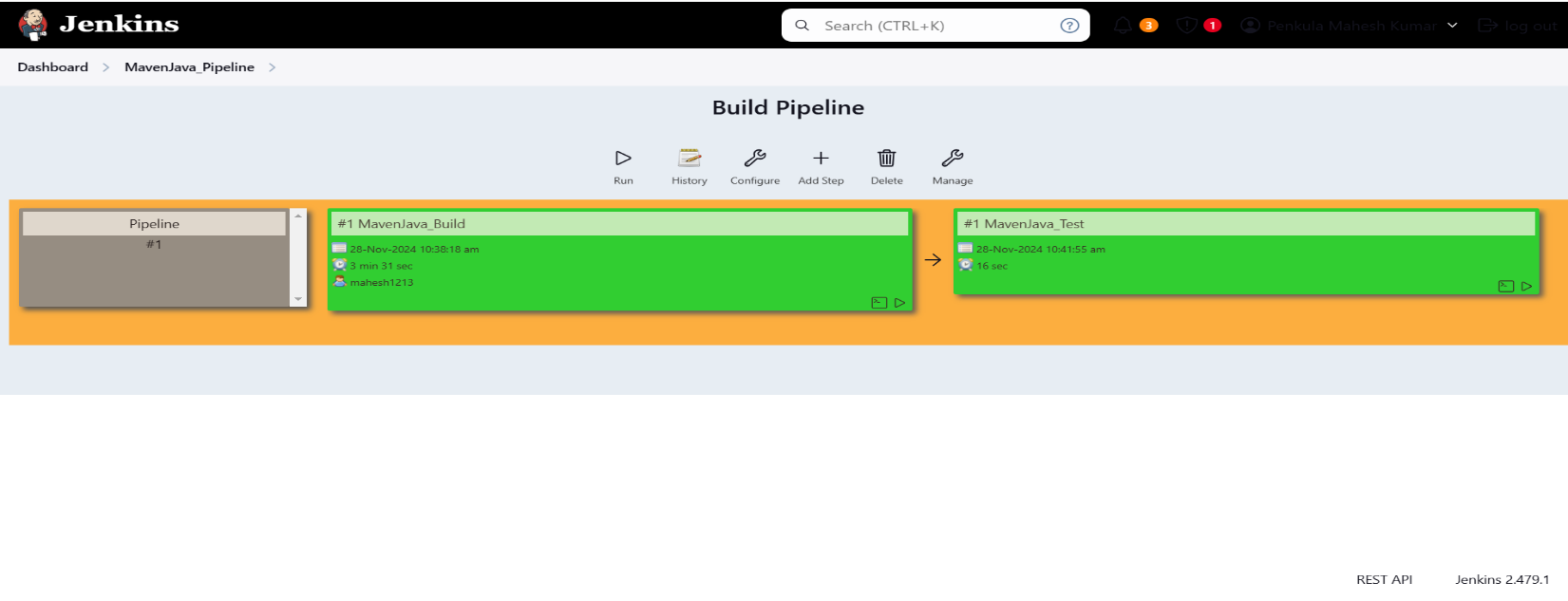
Initial job: MavenJava\_Build

Apply and Save OK

**Step 5**: Run the Pipeline and Check Output

Click on the trigger to run the pipeline

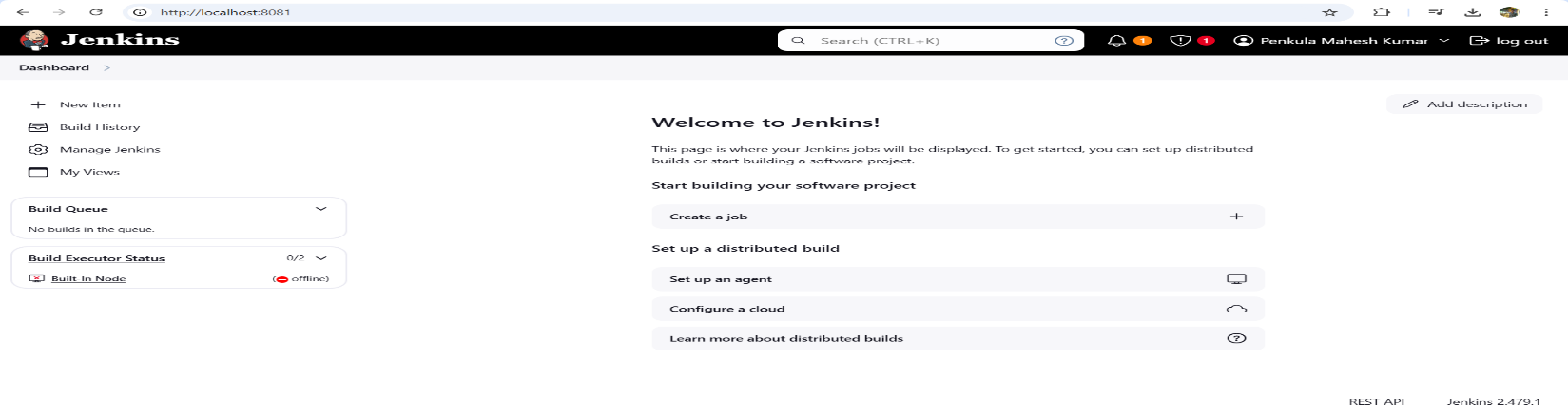
Click on the small black box to open the console to check if the build is success

****

**Maven Web Automation Steps:**

**Step 1:** Open Jenkins (localhost:8080)

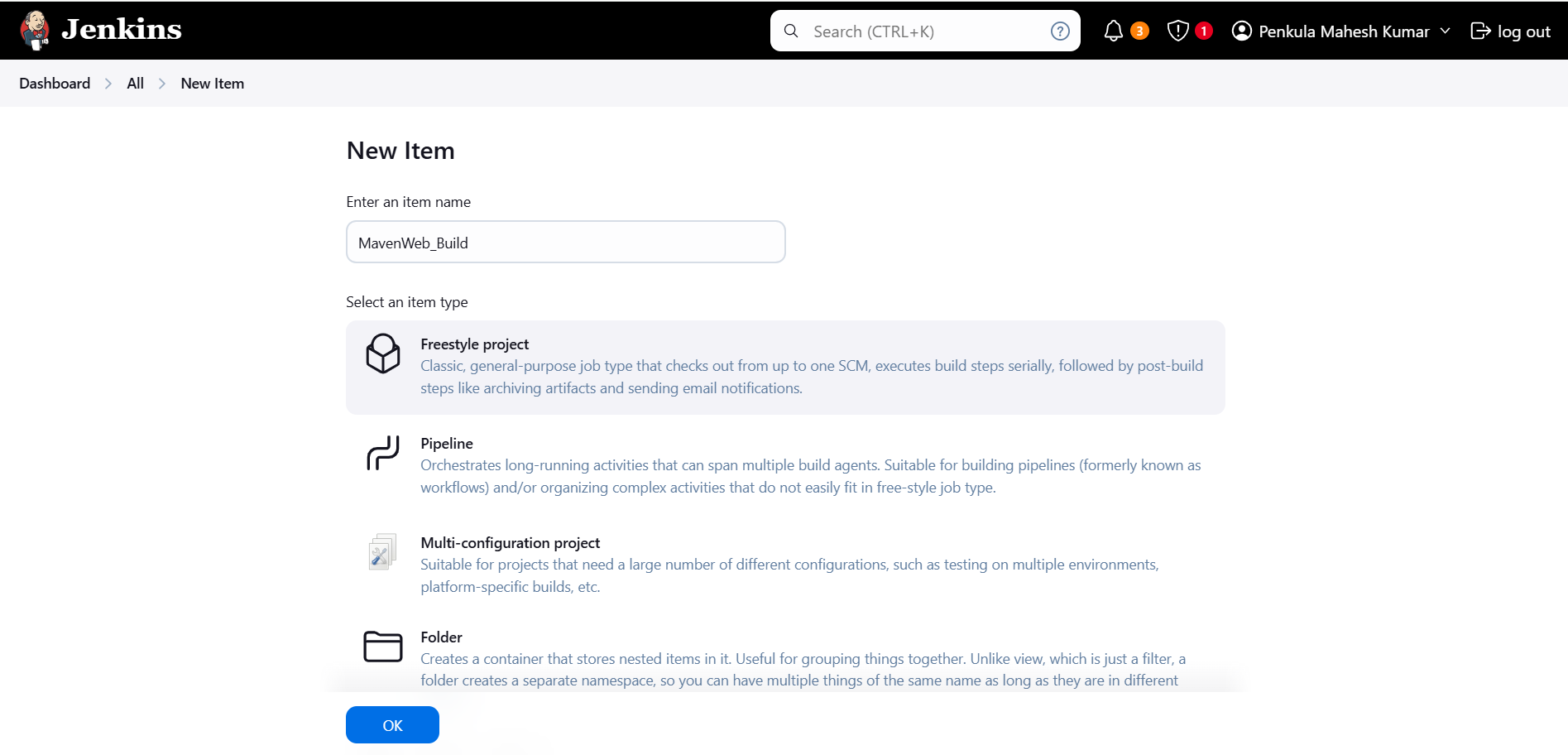
Click on "New Item" (left side menu)



**Step 2**: Create Freestyle Project (e.g., MavenWeb\_Build)

Enter project name (e.g., MavenWeb\_Build)

Click "OK"



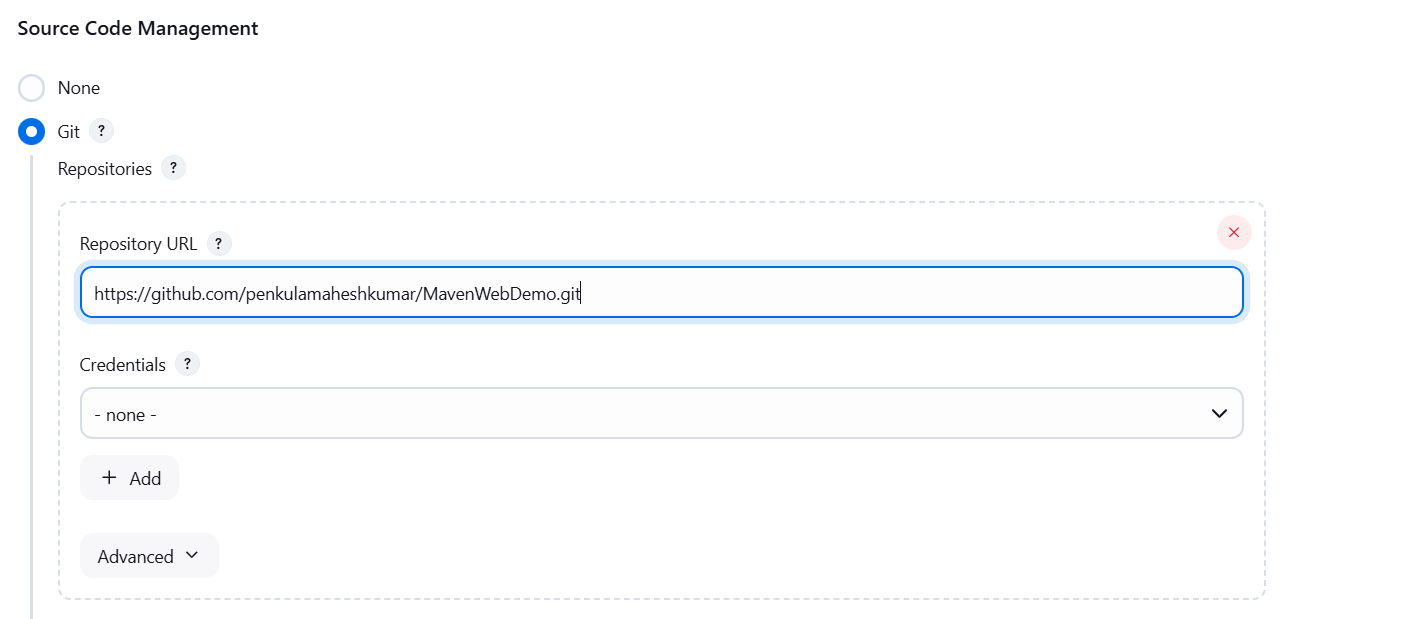
**Configure the project**:

**Description**: "Web Build demo"

**Source Code Management:**

Git repository URL: [GitMavenWeb repo URL]

Branches to build: \*/Main or master



**Build Steps**:

**Add Build Step** -> "Invoke top-level Maven targets"

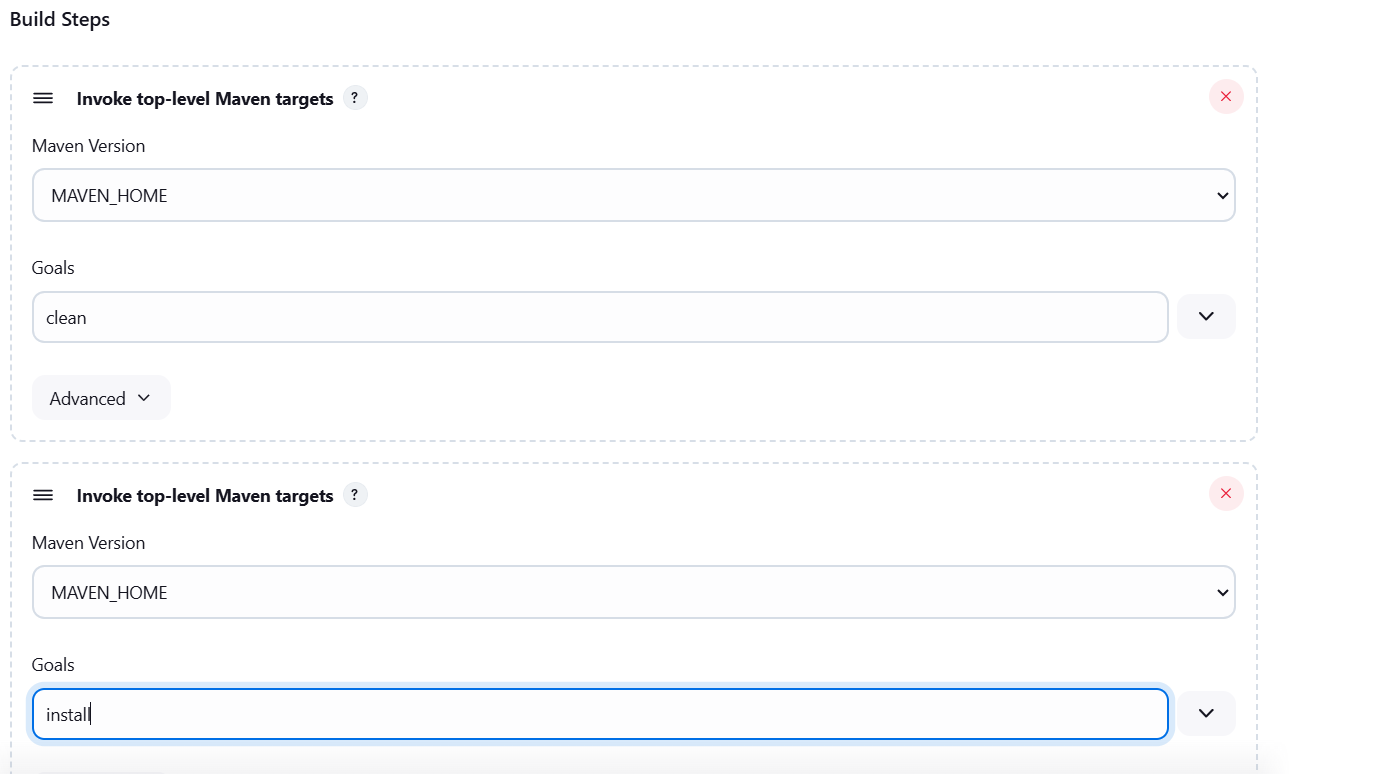
Maven version: MAVEN\_HOME

Goals: clean

**Add Build Step** -> "Invoke top-level Maven targets"

Maven version: MAVEN\_HOME

Goals: install



**Post-build Actions**:

**Add Post Build Action** -> "Archive the artifacts"

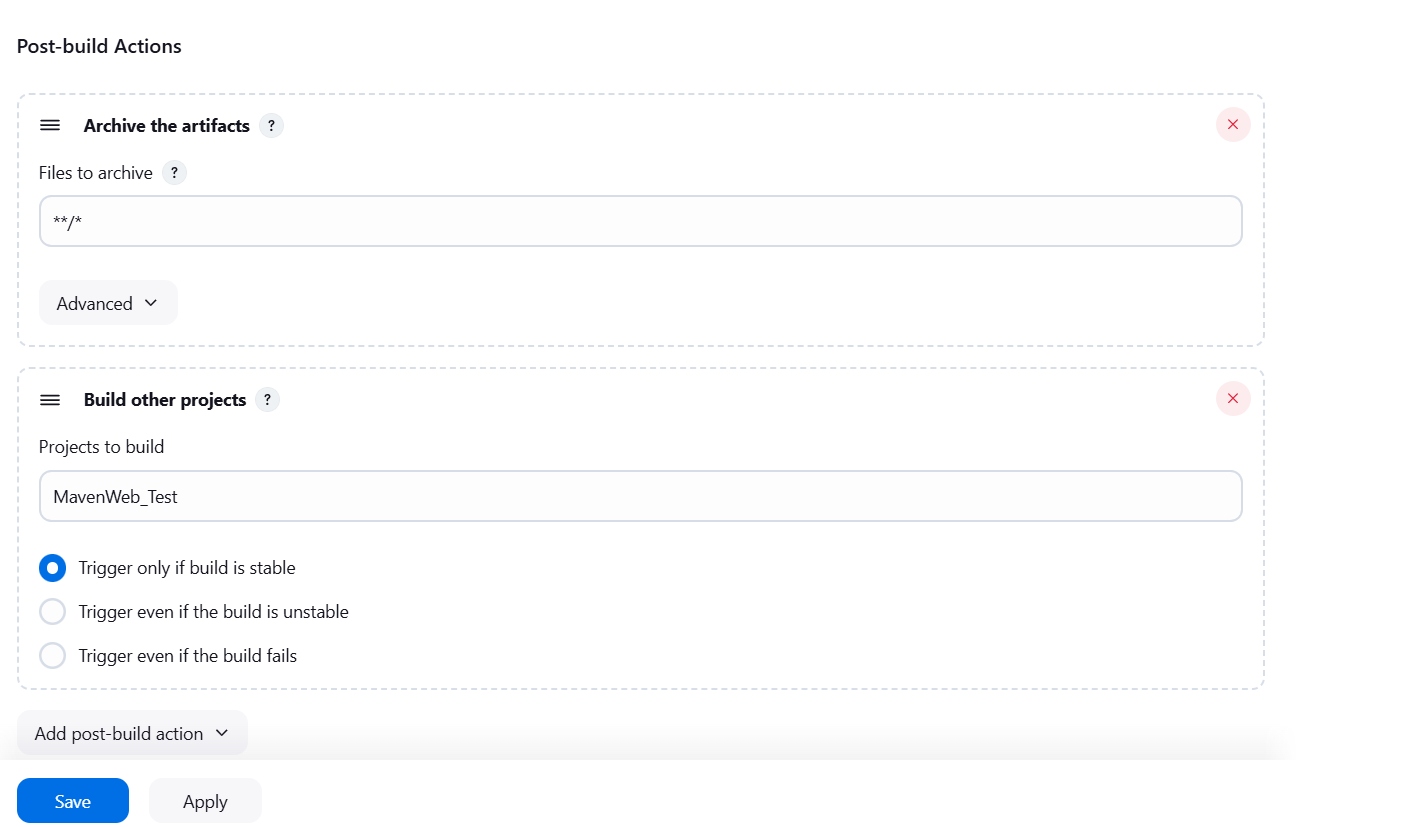
Files to archive: \*\*/\*

**Add Post Build Action** -> "Build other projects"

Projects to build: MavenWeb\_Test

Trigger: Only if build is stable

Apply and Save



**Step 3**: Create Freestyle Project (e.g., MavenWeb\_Test)

Enter project name (e.g., MavenWeb\_Test)

Click "OK"

**Configure the project:**

**Description:** "Test demo"

**Build Environment**:

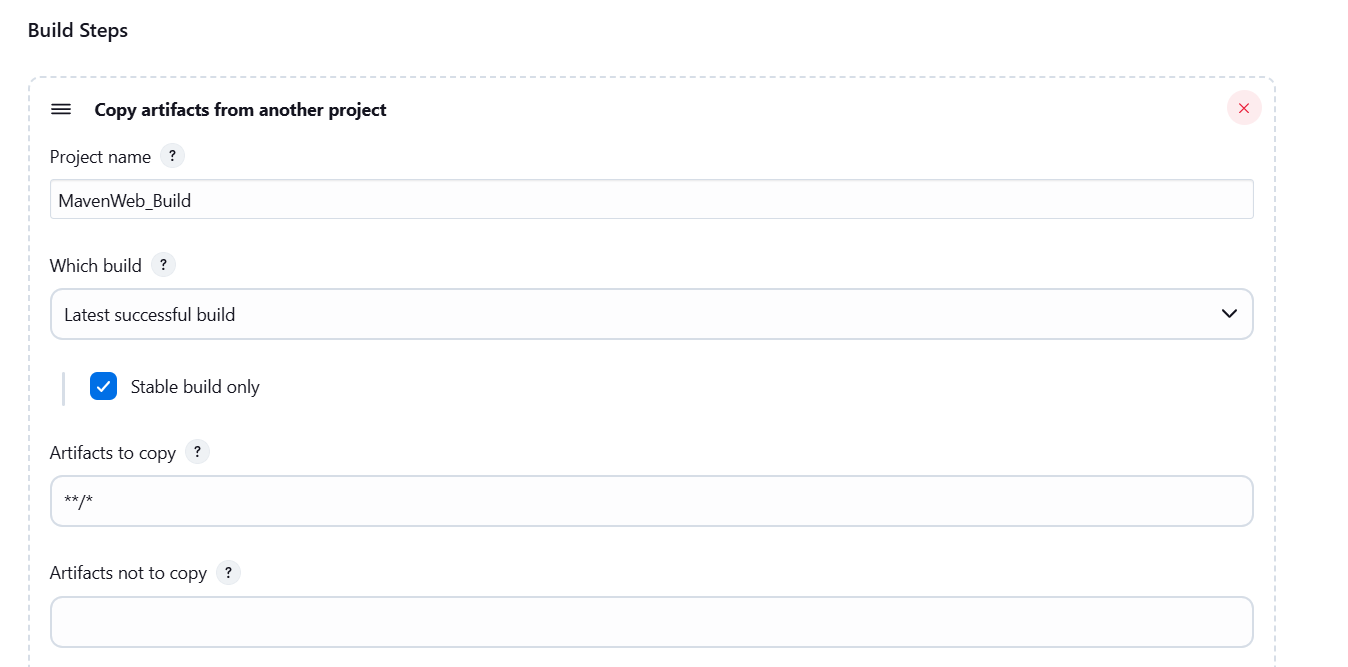
Check: "Delete the workspace before build starts"

**Add Build Step** -> "Copy artifacts from another project"

Project name: MavenWeb\_Build

Build: Stable build only

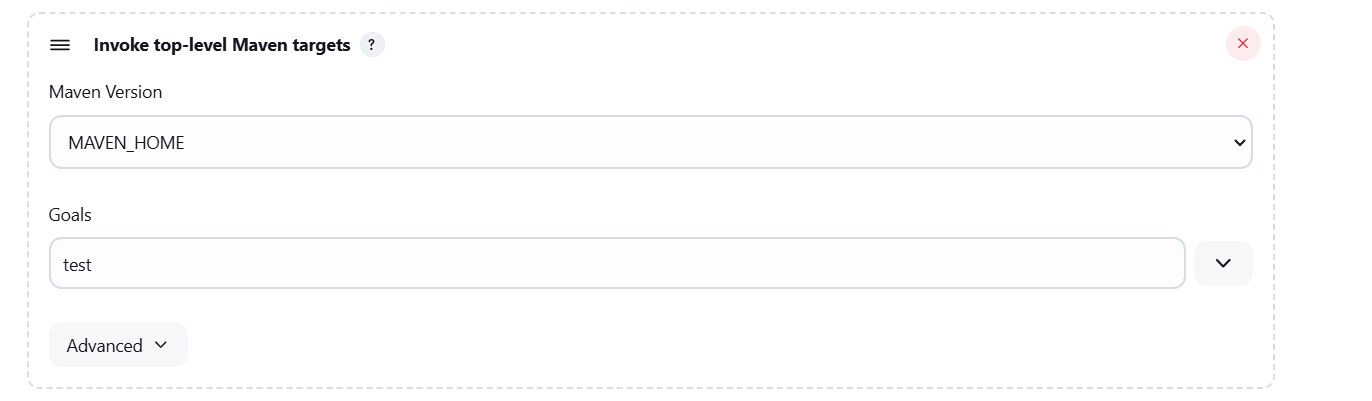
Artifacts to copy: \*\*/\*



**Add Build Step** -> "Invoke top-level Maven targets"

Maven version: MAVEN\_HOME

Goals: test



**Post-build Actions**:

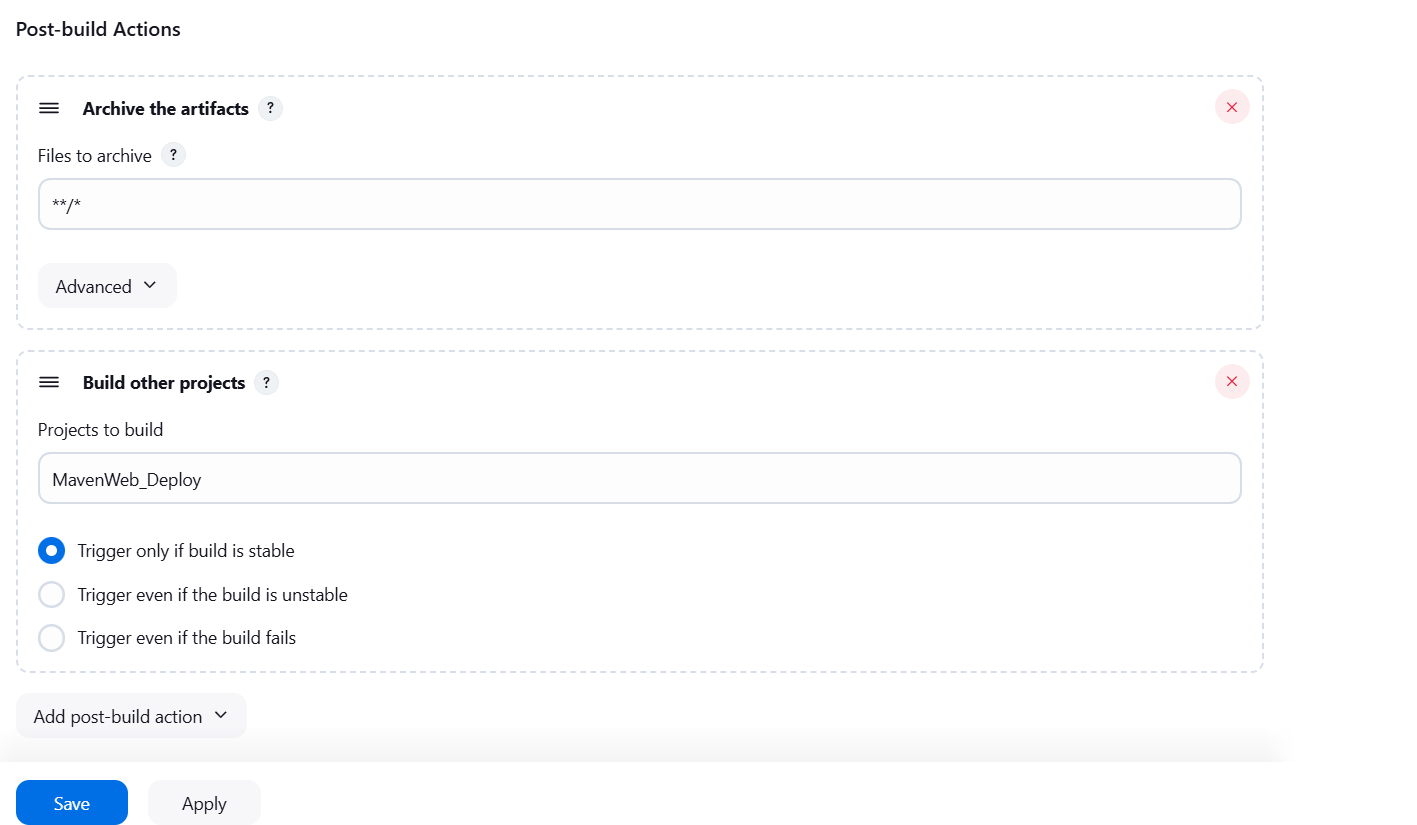
**Add Post Build Action** -> "Archive the artifacts"

Files to archive: \*\*/\*

**Add Post Build Action** -> "Build other projects"

Projects to build: MavenWeb\_Deploy

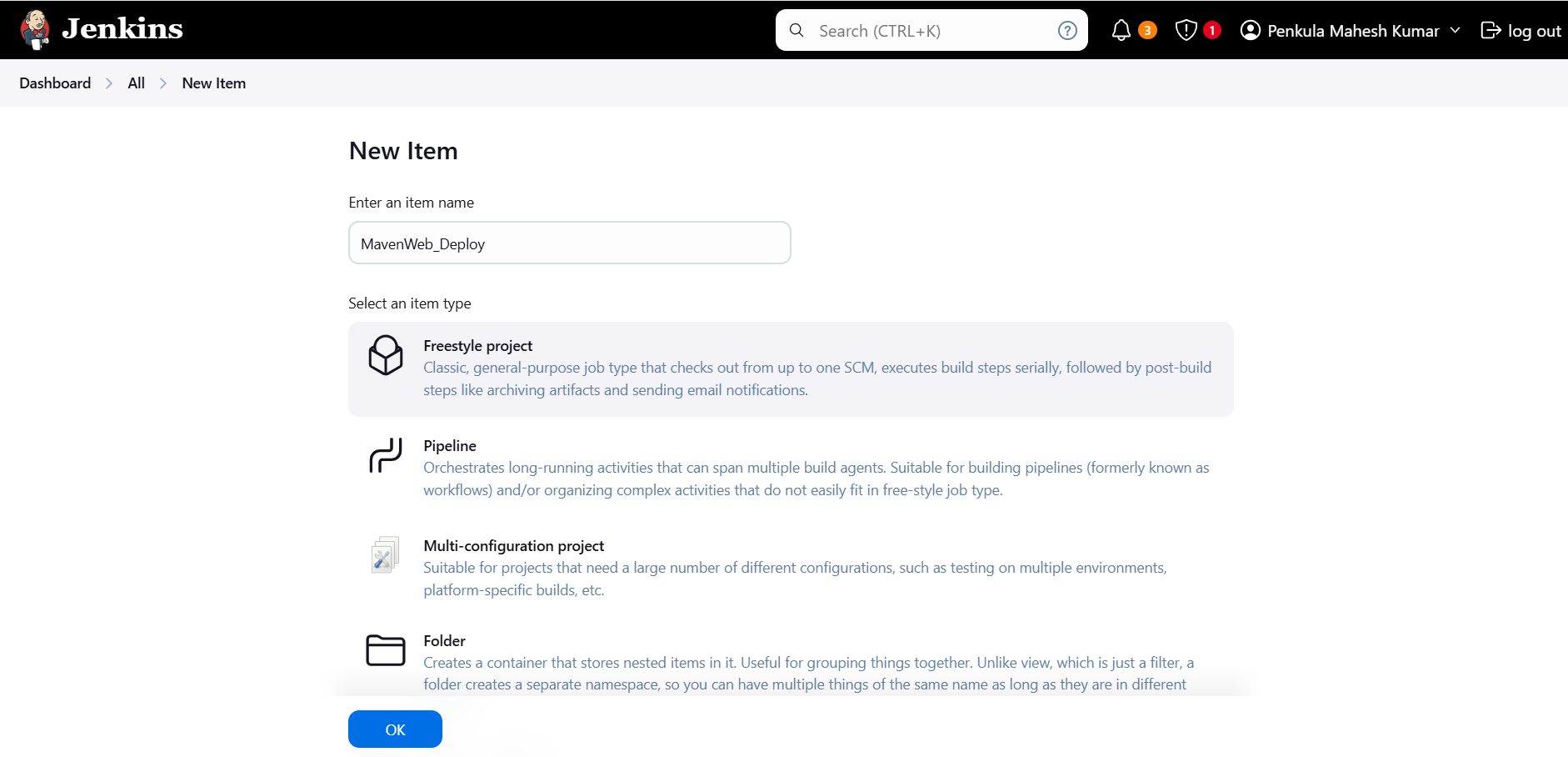
Apply and Save



**Step 4**: Create Freestyle Project (e.g., MavenWeb\_Deploy)

Enter project name (e.g., MavenWeb\_Deploy)

Click "OK"



**Configure the projec**t:

**Description**: "Web Code Deployment"

**Build Environment**:

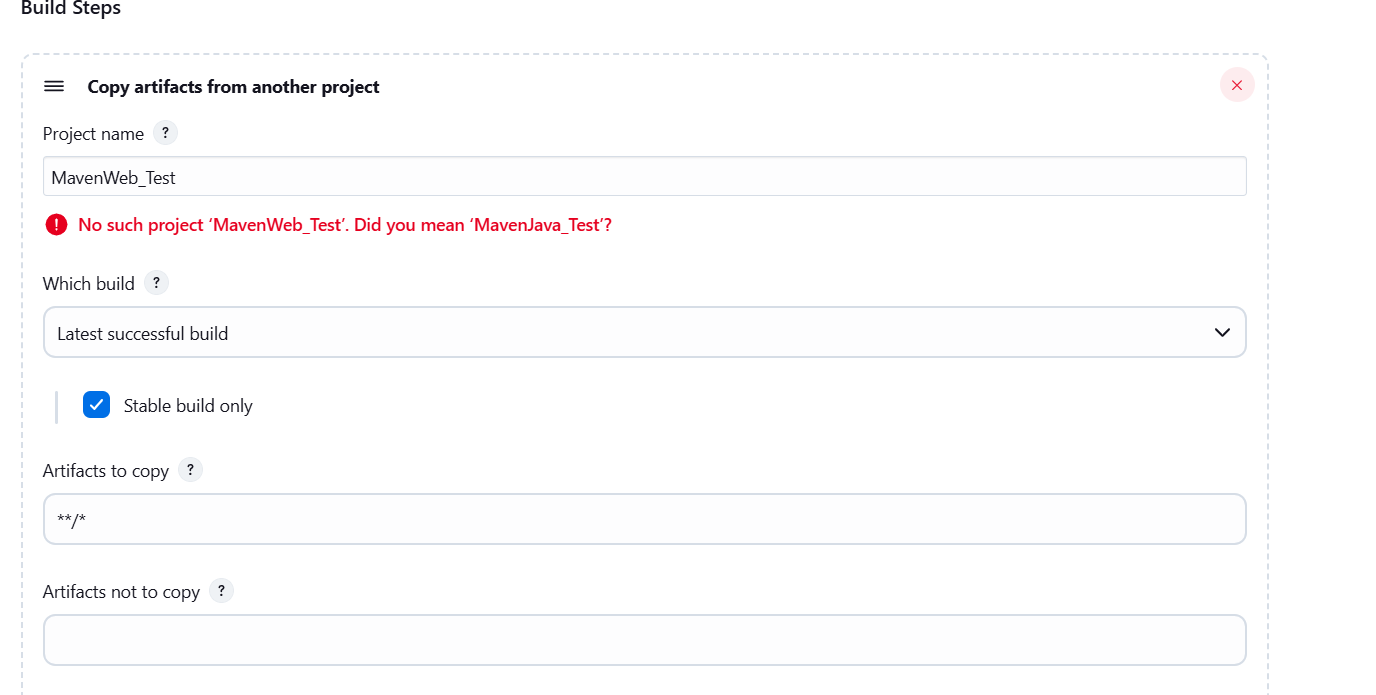
Check: "Delete the workspace before build starts"

**Add Build Step** -> "Copy artifacts from another project"

Project name: MavenWeb\_Test

Build: Stable build only

Artifacts to copy: \*\*/\*



**Post-build Actions**:

**Add Post Build Action** -> "Deploy WAR/EAR to a container"

WAR/EAR File: \*\*/\*.war

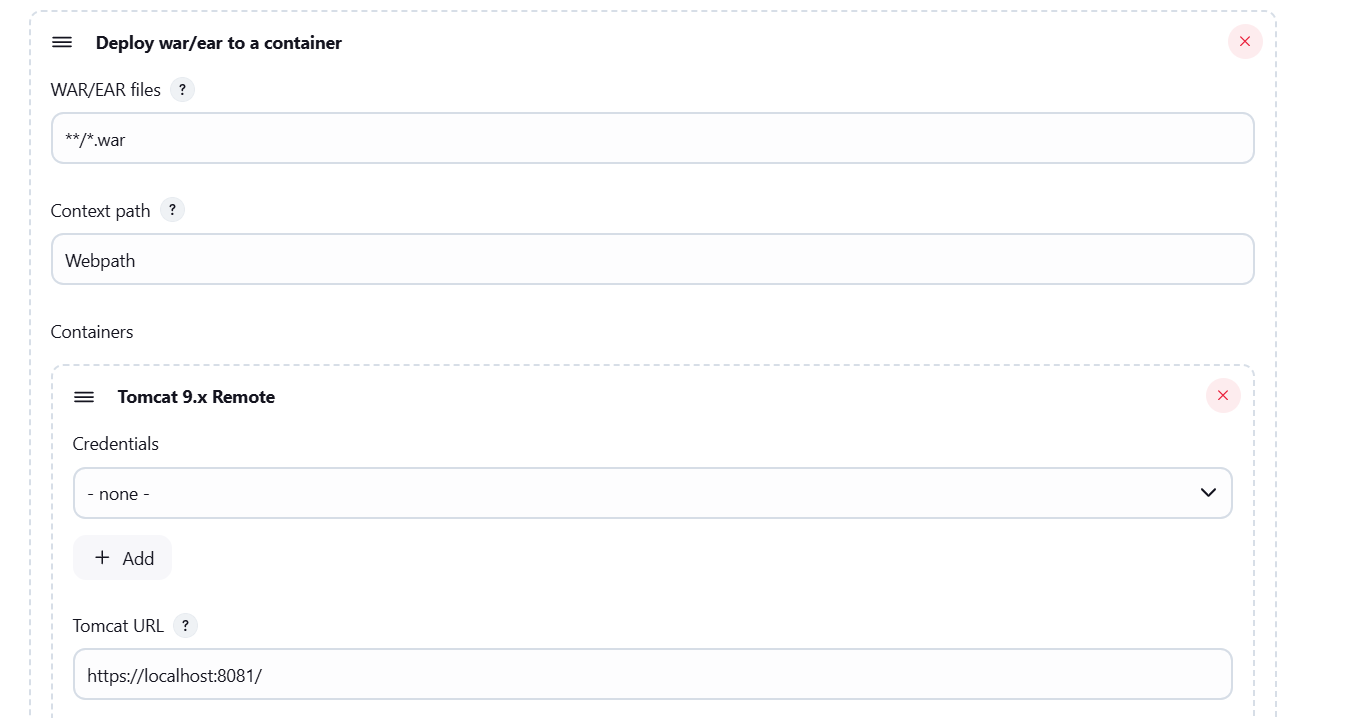
Context path: Webpath

Add container -> Tomcat 9.x remote

Credentials: Username: admin, Password: 1234

Tomcat URL: https://localhost:8085/

Apply and Save

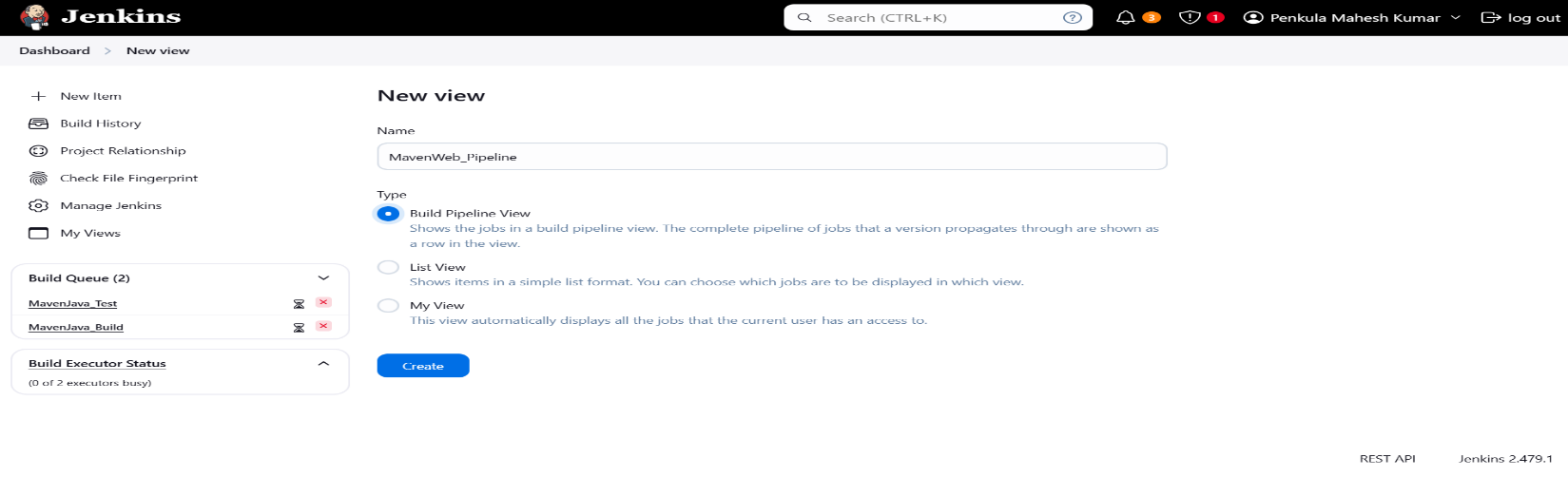


**Step 5**: Create Pipeline View for MavenWeb

Click "+" beside "All" on the dashboard

Enter name: MavenWeb\_Pipeline

**Select "Build pipeline view"**

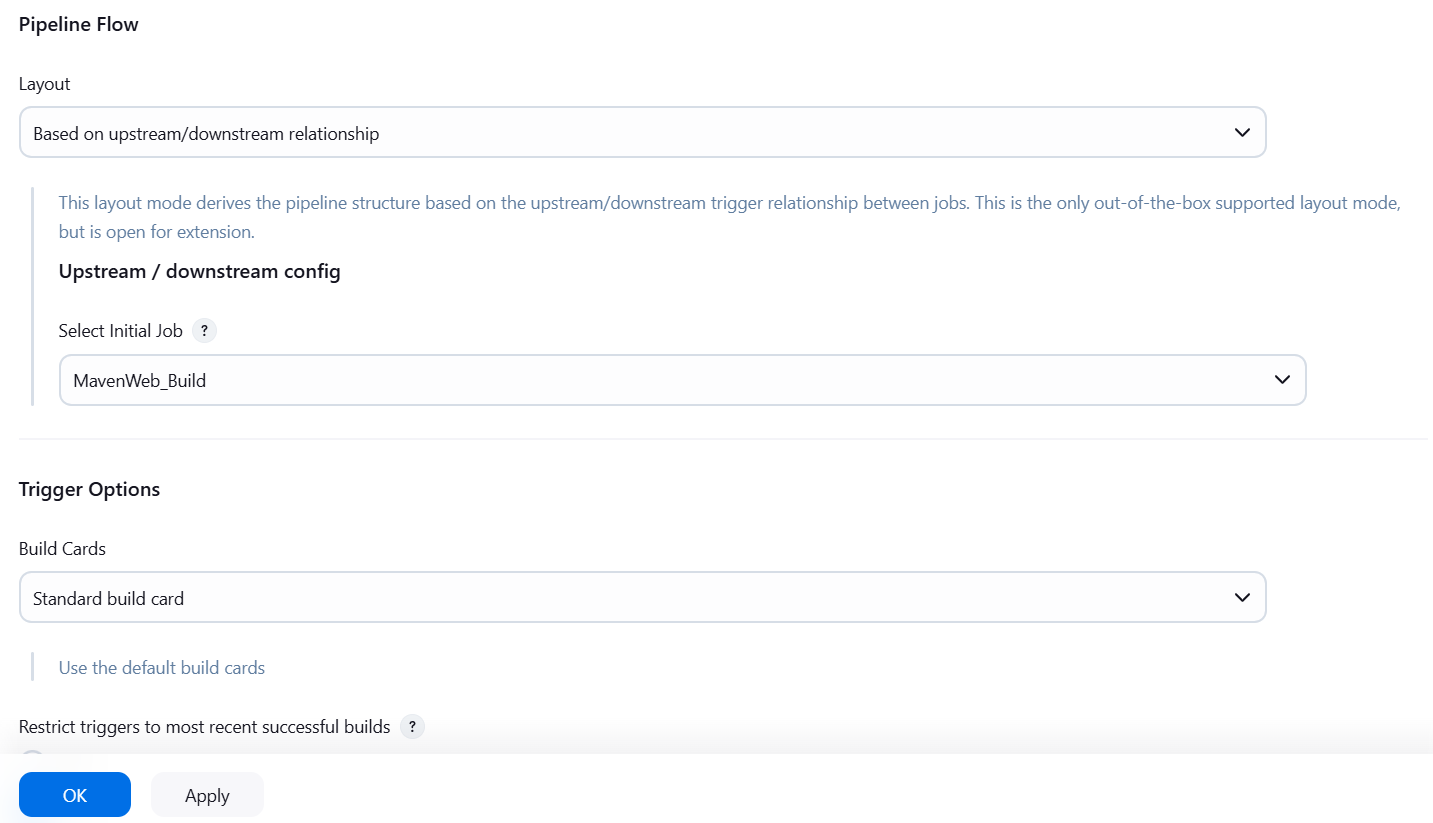


**Pipeline Flow**:

**Layout**: Based on upstream/downstream relationship

Initial job: MavenWeb\_Build

Apply and Save



**Step 6**: Run the Pipeline and Check Output

Click on the trigger **“RUN”** to run the pipeline

