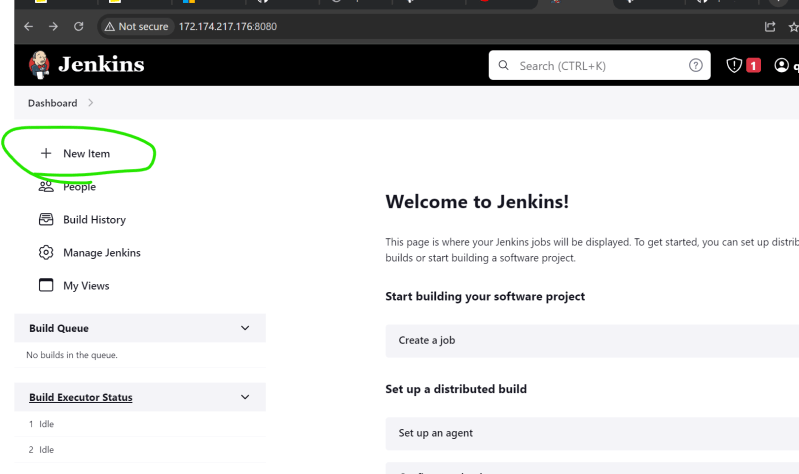
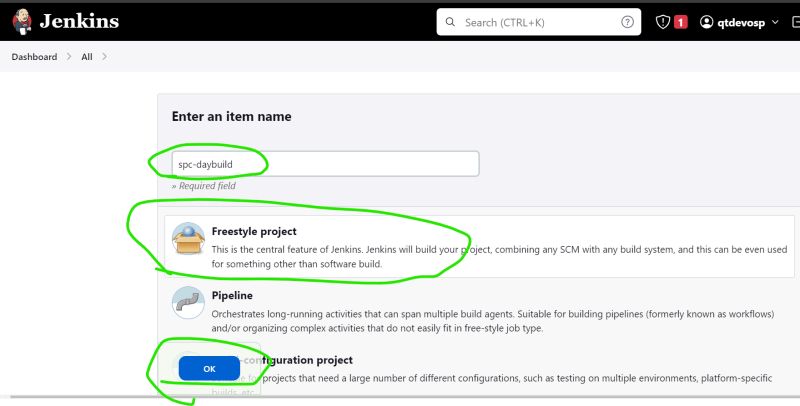
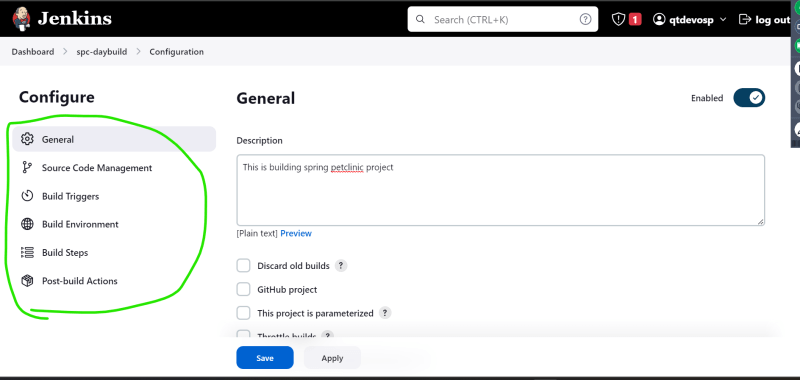
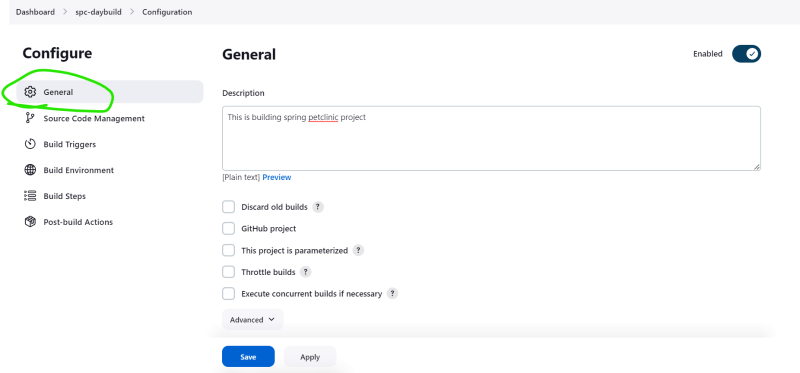
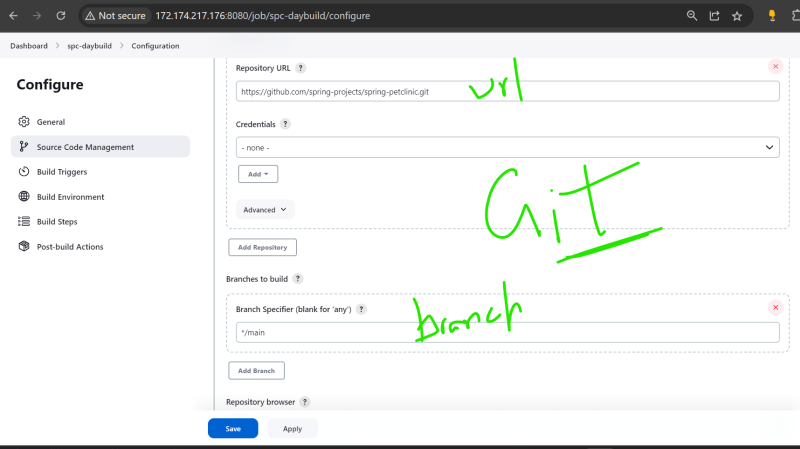
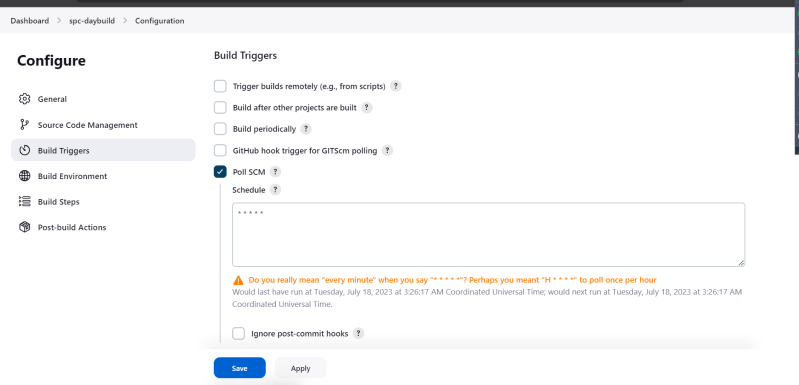
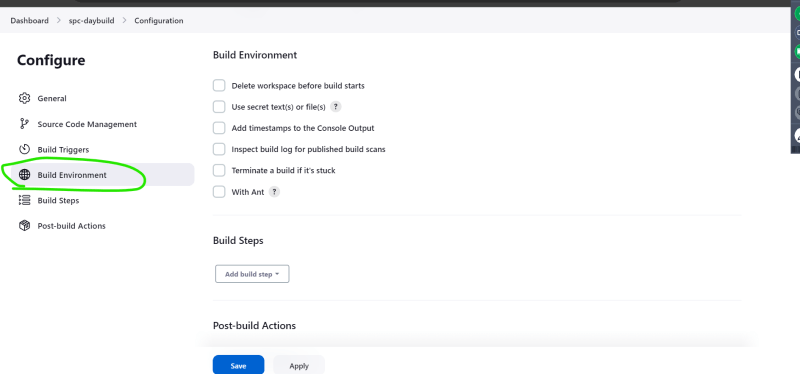
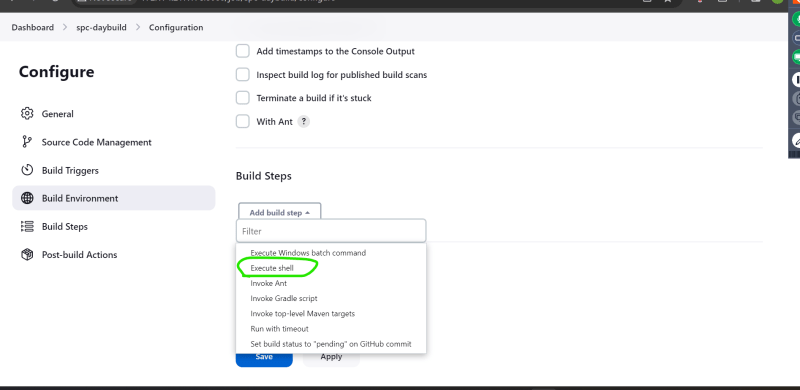
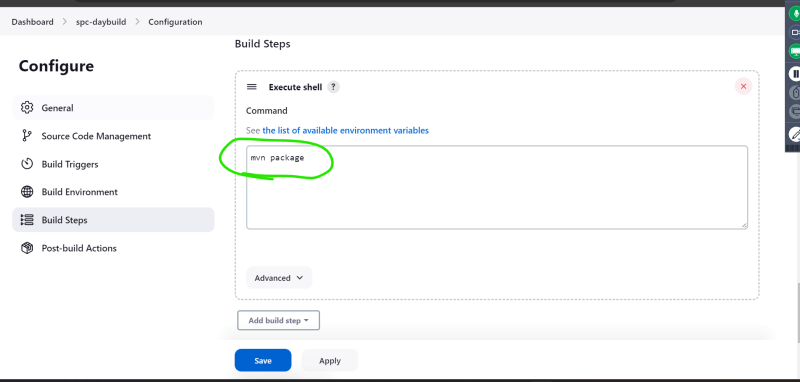
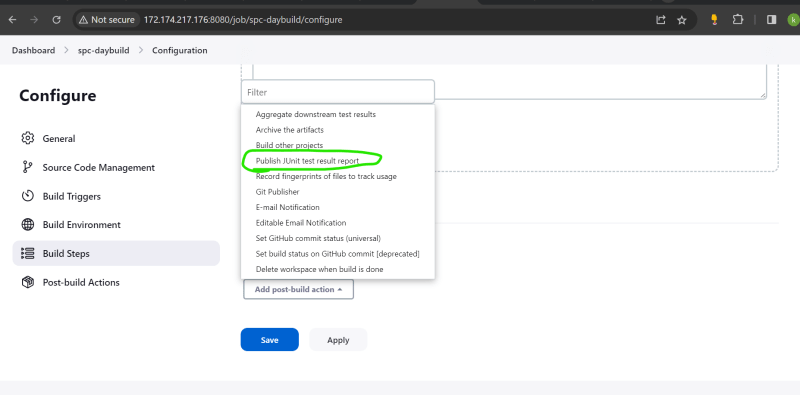
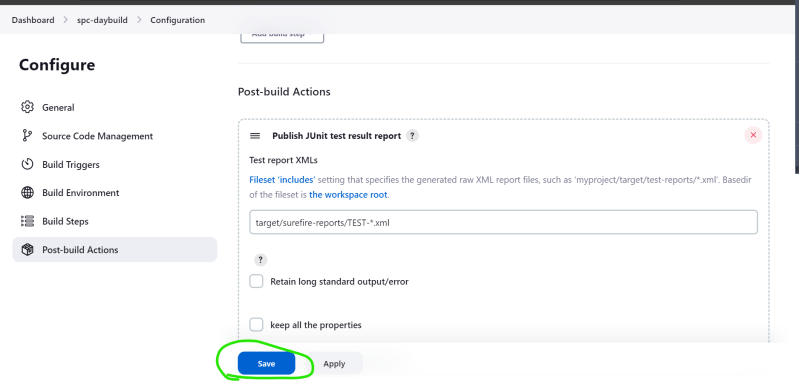
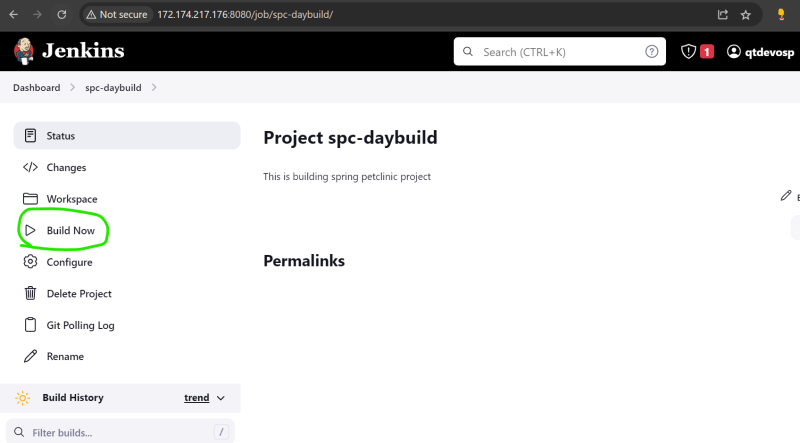
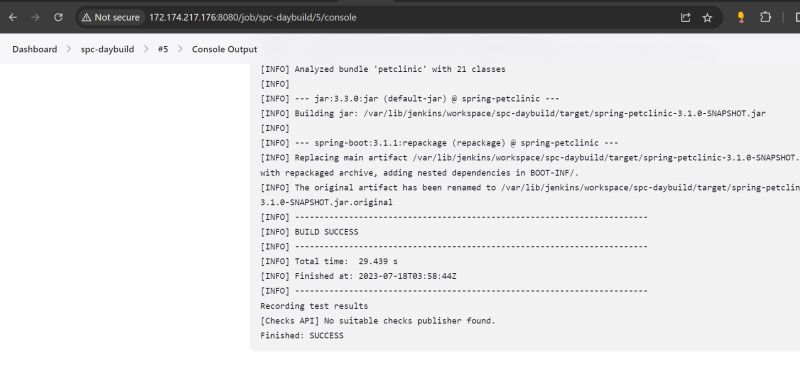
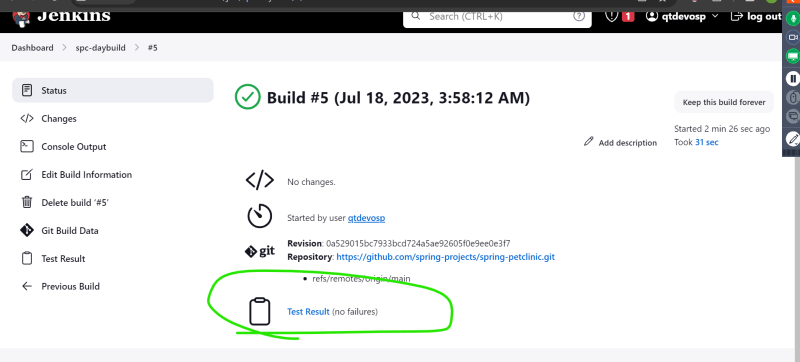
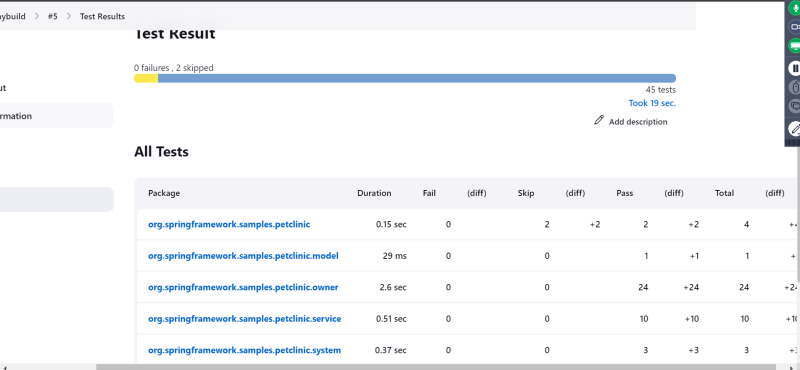
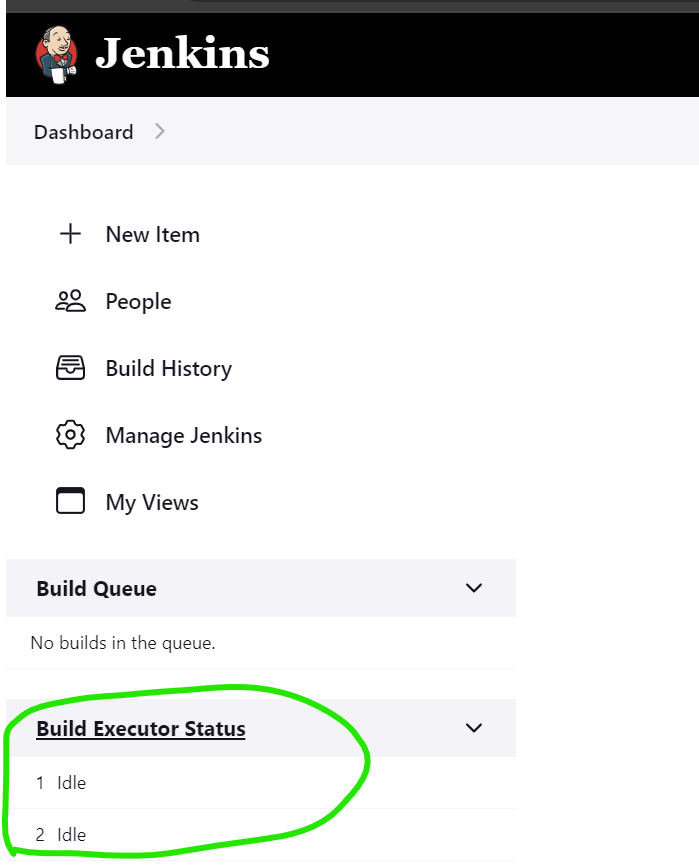
**Using Jenkins to build Maven Projects**

* Install jenkins
  + jdk 17
* Install and configure maven in Jenkins (Master Node)
* Let’s build spring petclinic <https://github.com/spring-projects/spring-petclinic>
  + software requirements
    - jdk 17
    - maven
* Create a free style project to build spring petclinic  
    
  
* Free style project sections  
  
* General: This represents the project information  
  
* Source Code Managent: This represent the code to be used for ci/cd pipelines  
  
* Build Triggers: This represent when to build
  + Build Periodically: If the project has to be build based on schedule, write cron expression into this <https://crontab.guru/>
  + Poll SCM: this represents jenkins polling scm (asking git) and the cron expression represents how frequently should it ask  
    
* Build Environment: This represents the environmental configuration  
  
* Build Steps: These are actual activities that are performed during execution.  
    
  
* Post Build actions: Actions to be performed after completion of build  
    
  
* Now we can wait for the trigger to call the job or trigger build manually  
  
* In Jenkins we can have multiple versions of java, maven etc and we can handle these by configuring jenkins
* To fix the maven 3.6.3 issue we had install 3.9 and used full path for package  
    
  
* Test results  
  

**Jenkins Terms**

* Jenkins Home: Jenkins home is a folder where jenkins stores all of is configuration. In the above case the workspace is /var/lib/jenkins. If you want to change the workspace deal with JENKINS\_HOME
* Backup of Jenkins is backup of Workspace
* Project: This contains the actvity that needs to be performed on triggers.
  + This project is stored as xml file in workspace
  + Types:
    - Freestyle project: This is UI based configuration
    - Pipeline: This is instructions expressed in some code format.
* Build: This represents the execution of project. Every build for a project has a running number called as Build id
* Node: This represents the machine on which build can be executed.
* Each Node can be configured to handle multiple builds by executors.  
  

mvn clean build: Maven clean goal (clean:clean) is bound to the clean phase in the clean lifecycle. Its clean:cleangoal deletes the output of a build by deleting the build directory. Thus, when mvn clean command executes, Maven deletes the build directory.

Concurrent Builds: it allows you to build multiple deployments with Vercel simultaneously. Team accounts allow you to increase the number of Concurrent Builds from the Billing section on the Team Settings page, enabling the team to create deployments faster.

Plugin: Plugin is a UI tool to do configuration, but those configurations must be present in the master node.