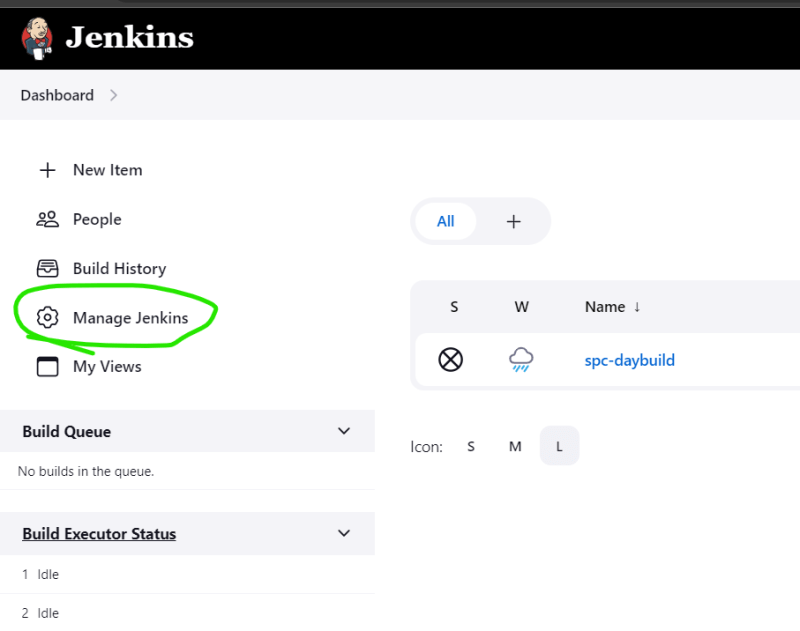
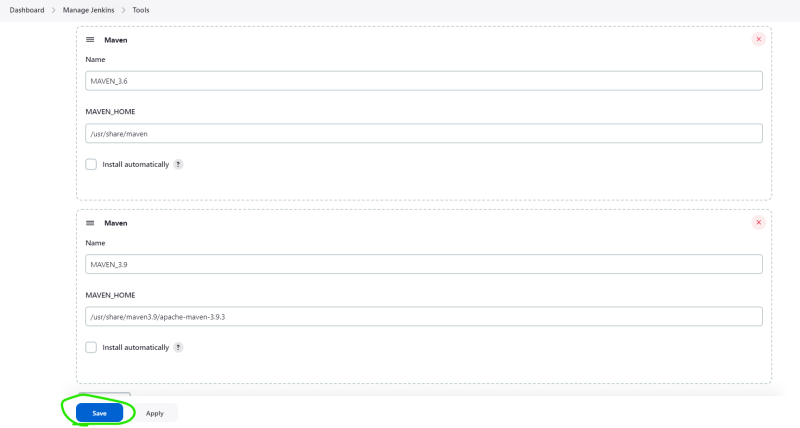
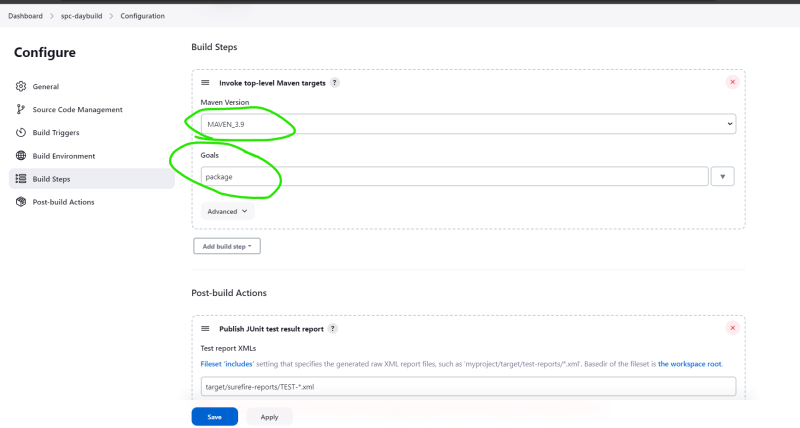
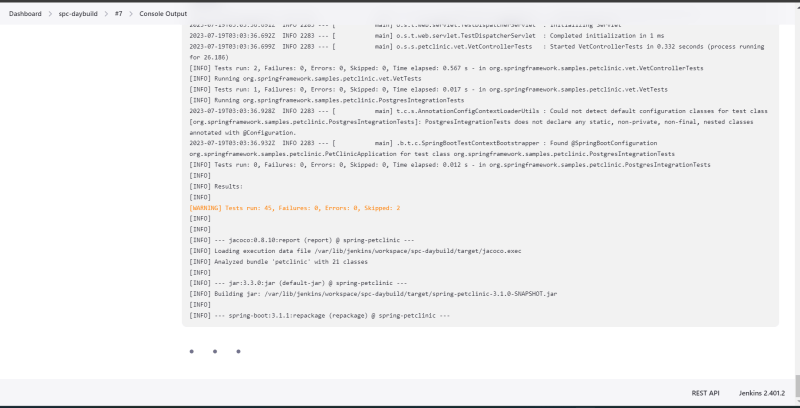
**Managing different versions of the tools using jenkins**

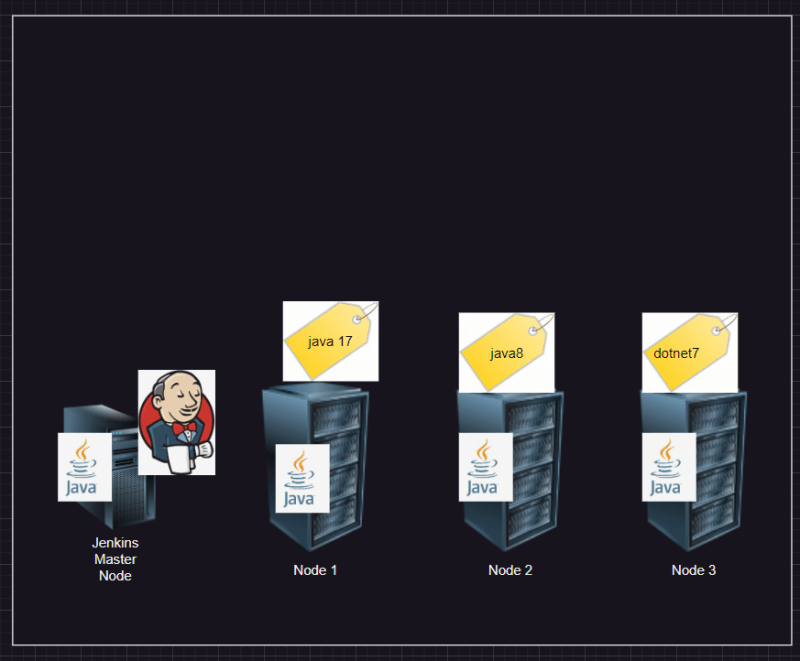
Let’s manage jenkins and navigate to tools => Maven installations  
  


Now let’s configure the spring pet clinic to use top level maven targets  


* Now build the project manually  
    
  

**Distributed Builds**

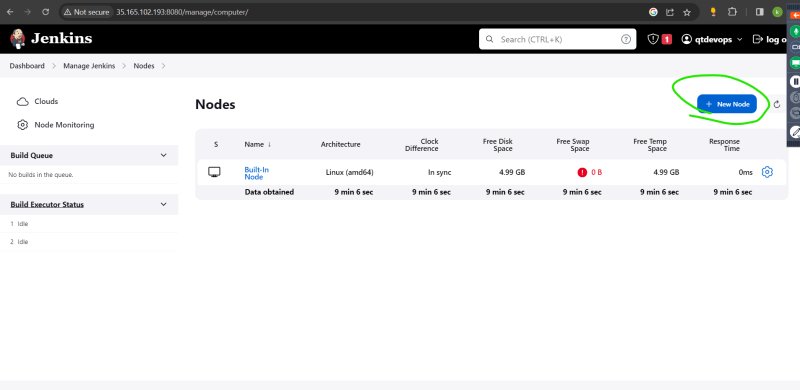
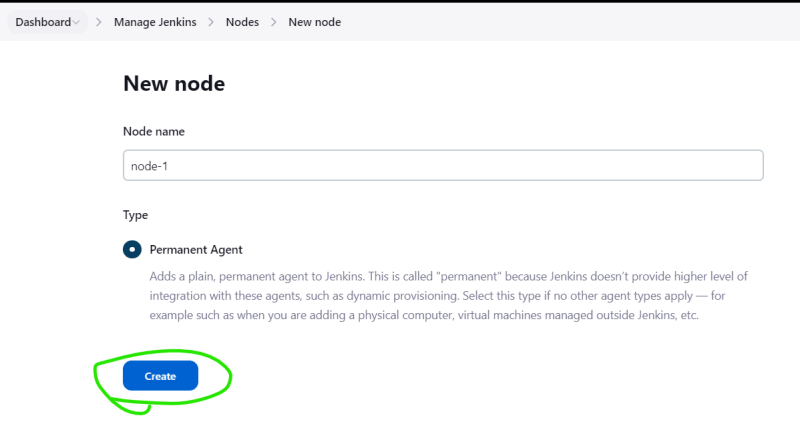
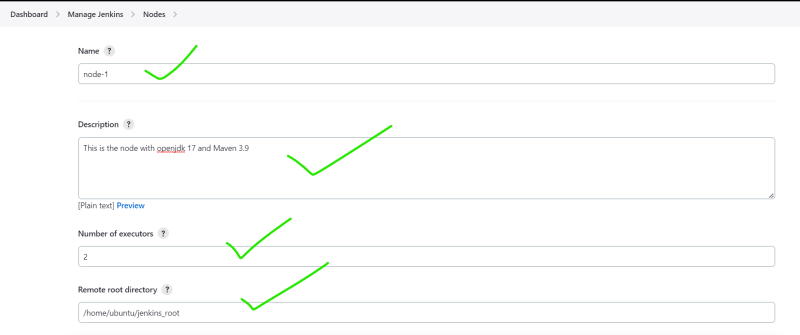
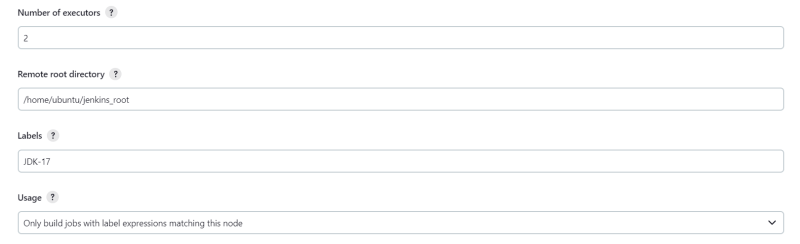
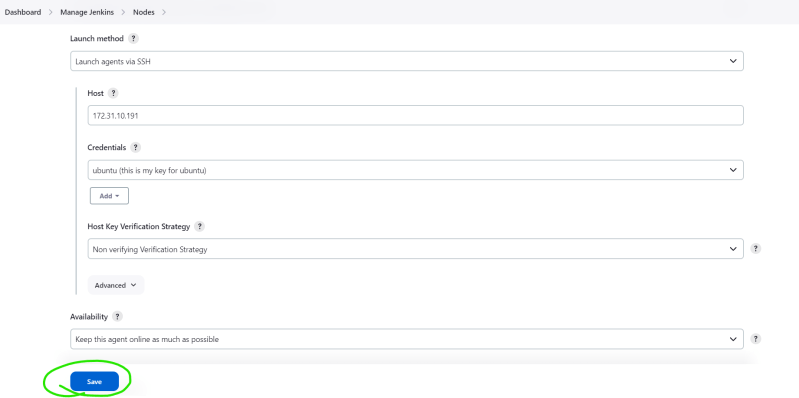
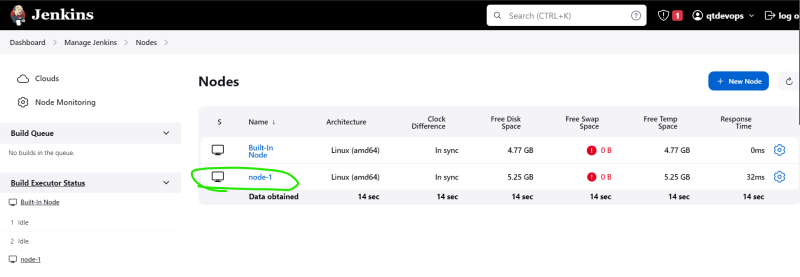
* Setup:
  + Fork Spring petclinic <https://github.com/dummyrepos/spring-petclinic-1>
    - requirements to build
      * java jdk 17
      * maven 3.9
  + Fork game of life <https://github.com/dummyrepos/game-of-life-july23>
    - requirements to build
      * java 8
      * maven
  + Fork nopcommerce <https://github.com/dummyrepos/nopCommerce-july23>
    - requirements to build
      * dotnet 7
* To handle different builds with different software needs, we tend to use different servers.

Jenkins has distributed builds where we can distribute the builds on differnt nodes by matching labels  


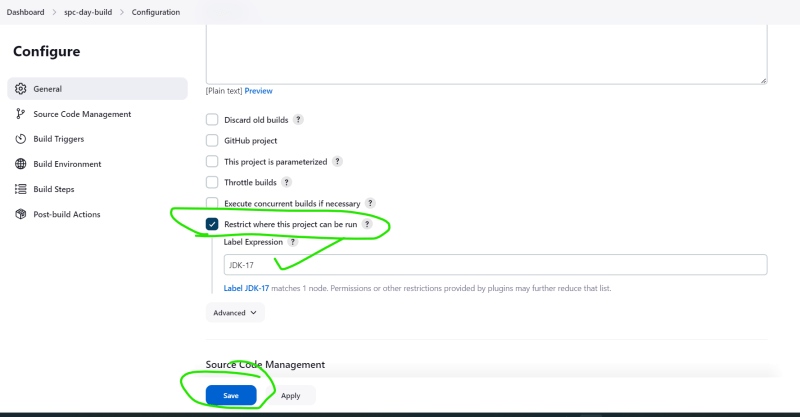
While creating a project we can set labels and expect them to be executed on the node matching labels

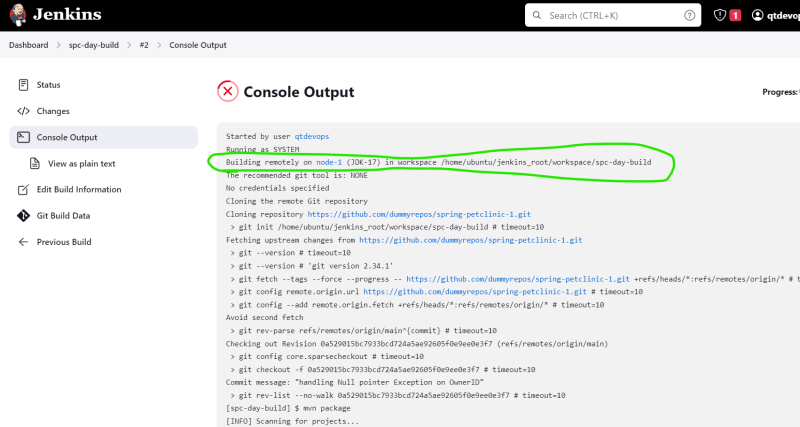
**How to add multiple nodes to jenkins**

* Lets create 2 ubuntu vms and lets make one vm the jenkins master
* one one machine install and configure jenkins
* On the node (we will be using existing credentials)
* install jdk 17
* install maven 3.9
* Now lets configure node to the jenkins master with label JDK-17

On Jenkins UI Navigate to Manage Jenkins => Nodes and Clouds  
  
  
  
  
  


**Let’s setup spring petclinic to execute node-1**

Configure spc-day build same as last session with one restriction in General section  


Now build and verify the console output  


**How to execute command on remote server> by distributed build setup**