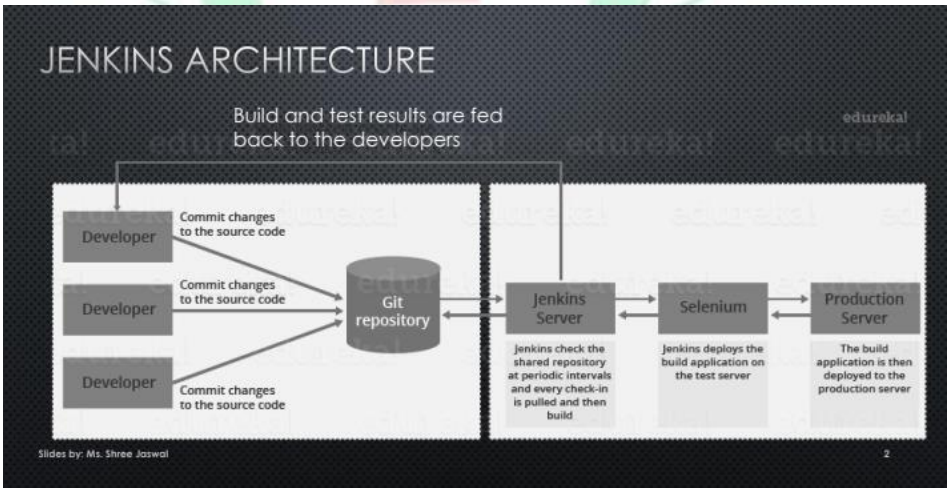
St. Francis Institute of Technology, Mumbai-400 103
Department Of Information Technology

A.Y. 2023-2024
Class: TE-ITA/B, Semester: V

Subject: **DevOps Lab**

Experiment – 7: To understand master-slave architecture and scale your Jenkins standalone implementation by implementing slave nodes.

1. **Aim:** To understand master-slave architecture and scale your Jenkins standalone implementation by implementing slave nodes
2. **Objectives:** Aim of this experiment is that, the students will be able to do
   - Jenkins management
   - Adding a slave node to Jenkins
3. **Outcomes:** After study of this experiment, the students will be able
   - To understand the importance of Jenkins to Build and deploy Software Applications on server environment.
4. **Prerequisite:** Knowledge of Computer Networks concept of Master-slave architecture
5. **Requirements:** Jenkins, JDK, python, Personal Computer, Windows operating system, browser, Internet Connection, Microsoft Word.
6. **Pre-Experiment Exercise:**
   Brief Theory: Refer shared material
7. **Laboratory Exercise**
   A. **Procedure:**
      a. **Answer the following:**
         - Explain the architecture of Jenkins with diagram.

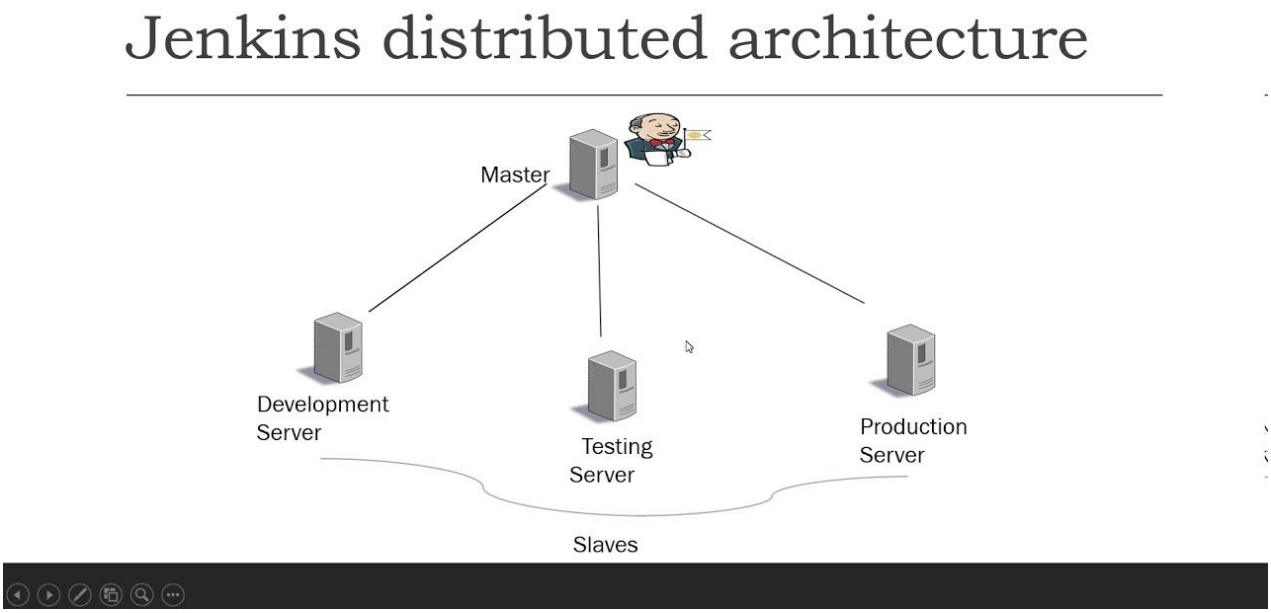


Jenkins Architecture

Here's how Jenkins elements are put together and interact:
Developers commit changes to the source code, found in the repository.
The Jenkins CI server checks the repository at regular intervals and pulls any newly available code.
The Build Server builds the code into an executable file. In case the build fails, feedback is sent to the developers.
Jenkins deploys the build application to the test server. If the test fails, the developers are alerted.
If the code is error-free, the tested application is deployed on the production server.

The files can contain different code and be very large, requiring multiple builds. However, a single Jenkins server cannot handle multiple files and builds simultaneously; for that, a distributed Jenkins architecture is necessary.

- Explain the distributed architecture of Jenkins with diagram?
  In a distributed Jenkins architecture, multiple build agents (slaves) are used to distribute the workload across different machines. This approach is beneficial for scalability, load balancing, and supporting a variety of build environments. Here's a diagram illustrating the distributed architecture of Jenkins:
  In this distributed setup:



The Jenkins Master manages the overall configuration and job scheduling.
Multiple build agents (slaves) can be added as needed, and they can run on different hardware or virtual machines.
Jobs can be configured to run on specific agents, allowing for parallel execution and efficient resource utilization.
Distributed builds can improve build performance and reduce queue times, making it suitable for large and complex projects.
Jenkins also supports other advanced features such as cloud-based agents (e.g., Amazon EC2 instances) that can be dynamically provisioned and decommissioned based on workload, further enhancing its scalability and flexibility.

b. Execute following (Refer the shared material) and attach screenshots:

- Create a slave node and connect it to master
- Use an existing project or a new project to run in the slave node

8. Post-Experiments Exercise

A. Extended Theory:

Nil

B. Questions:

- What are the ways to configure Jenkins node agent to communicate with Jenkins master?
- Which architecture is recommended for a scalable Jenkins environment?

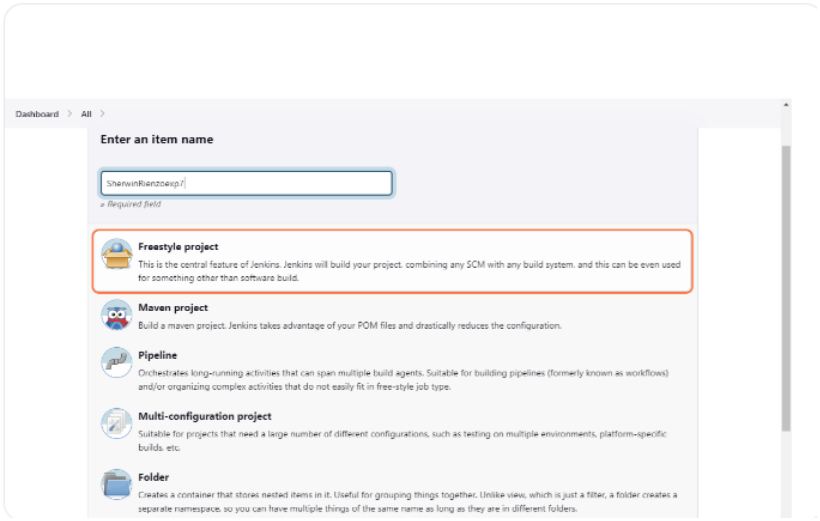
C. Conclusion:

- Write what was performed in the experiment.
- Write the significance of the topic studied in the experiment.

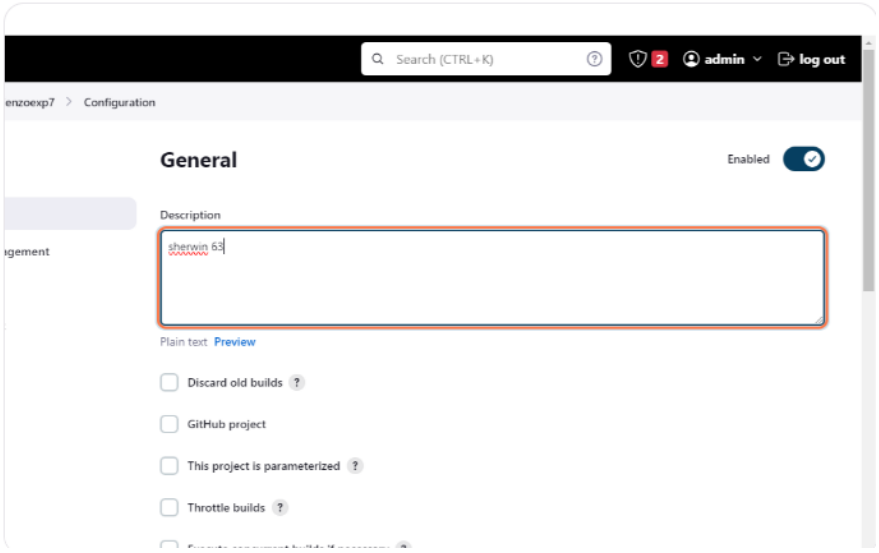
D. References:

https://jenkins.io/doc/
https://www.slideshare.net/abediaz/introduction-to-jenkins
https://www.studytonight.com/jenkins/jenkins-master-slave-configuration
https://www.edureka.co/blog/jenkins-master-and-slave-architecture-a-complete-guide/

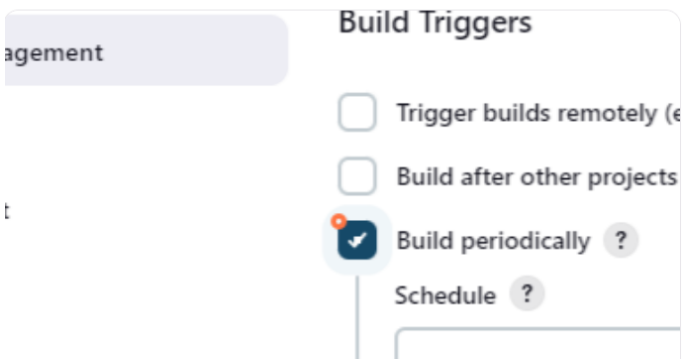
1 Click on Freestyle project...



2 Click on description

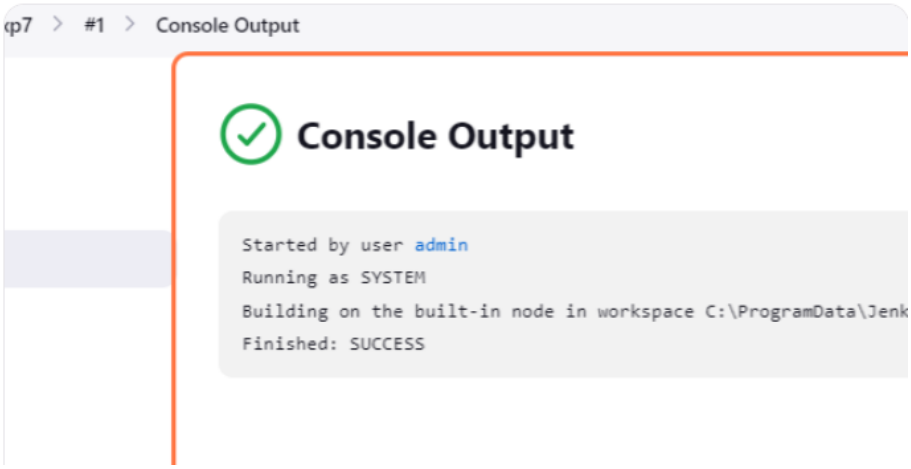


3 Check hudson-triggers-TimerTrigger

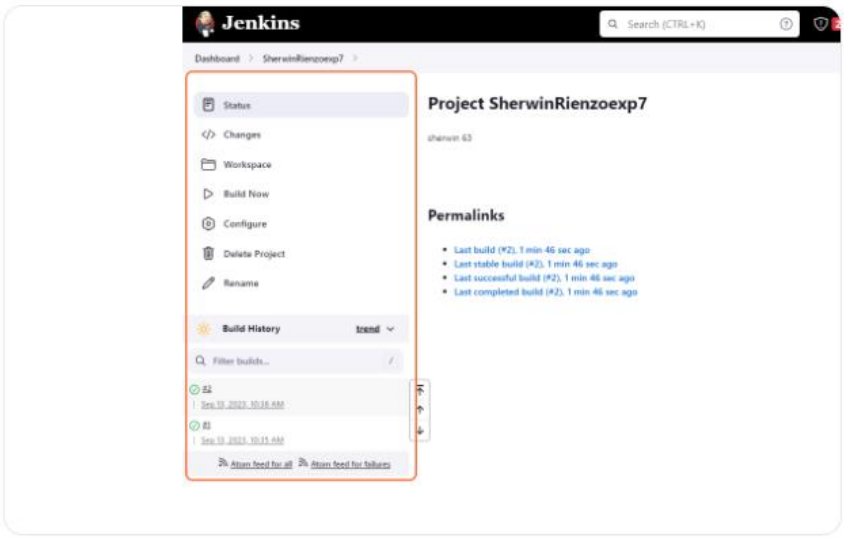


4 Click on _.spec

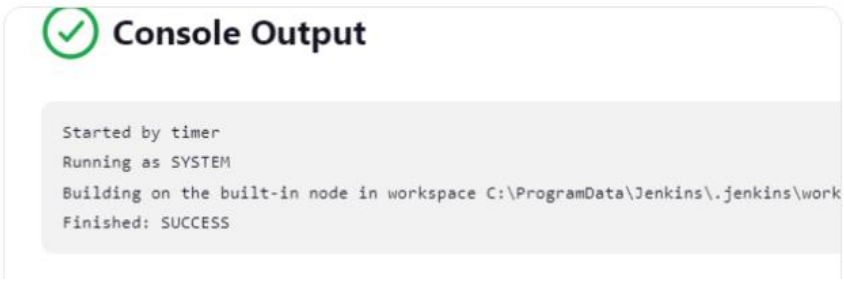
5 Click on Console Output...



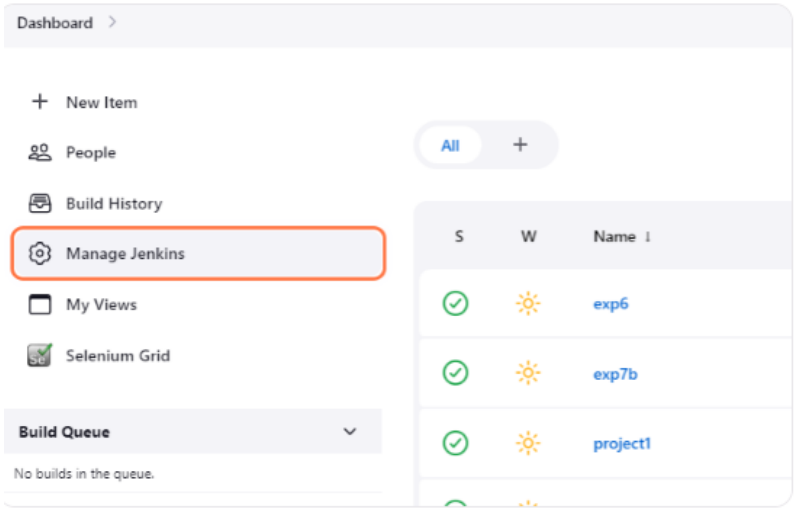
6 Click on Status...



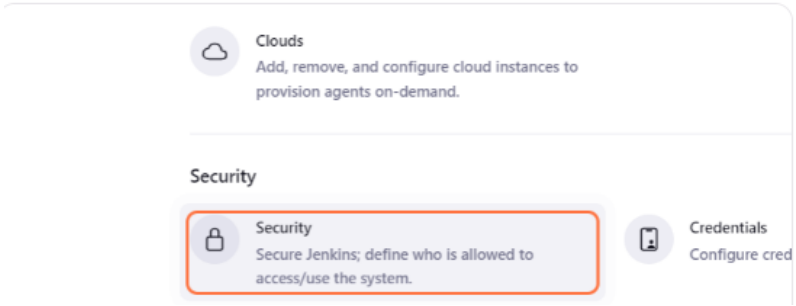
7 Click on Console Output...



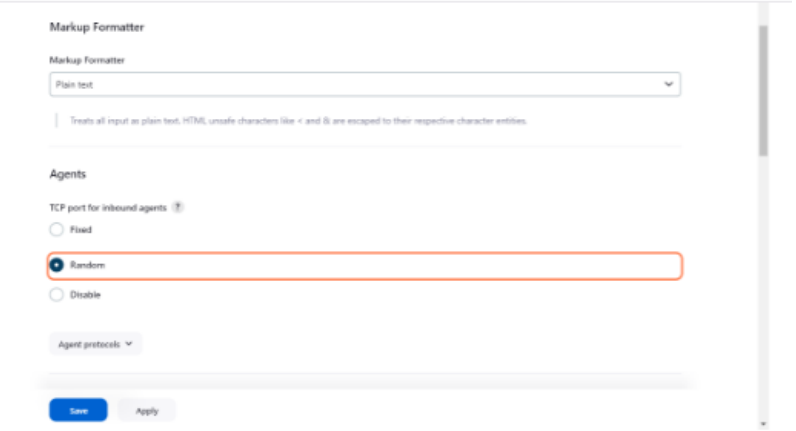
8 Click on Manage Jenkins



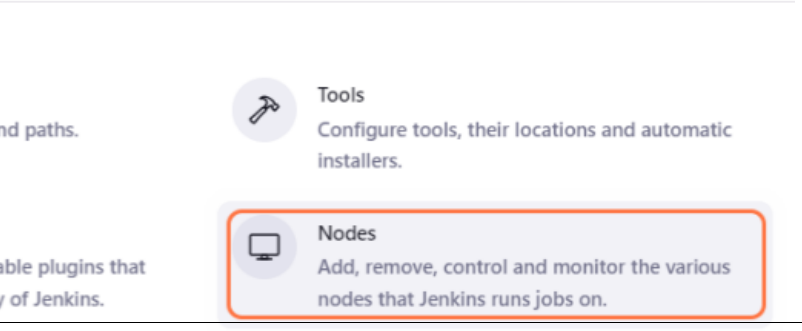
9 Click on Security...



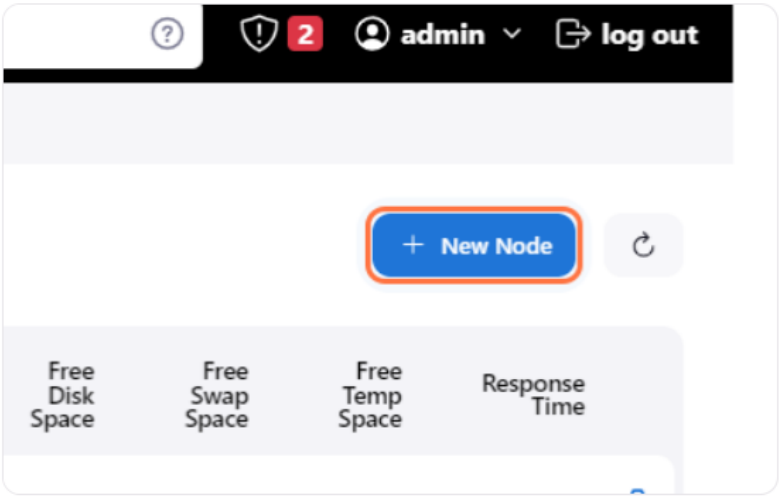
10 Click on Random



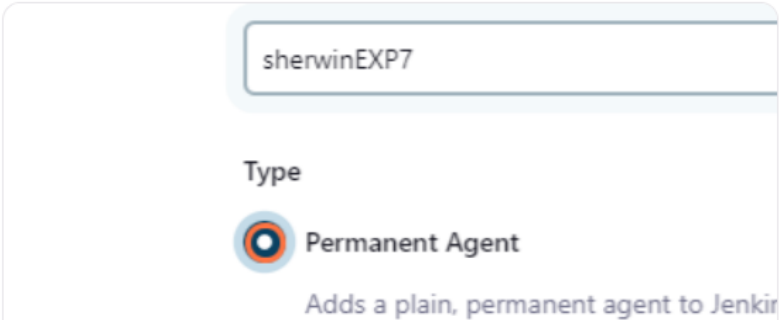
11 Click on Nodes...



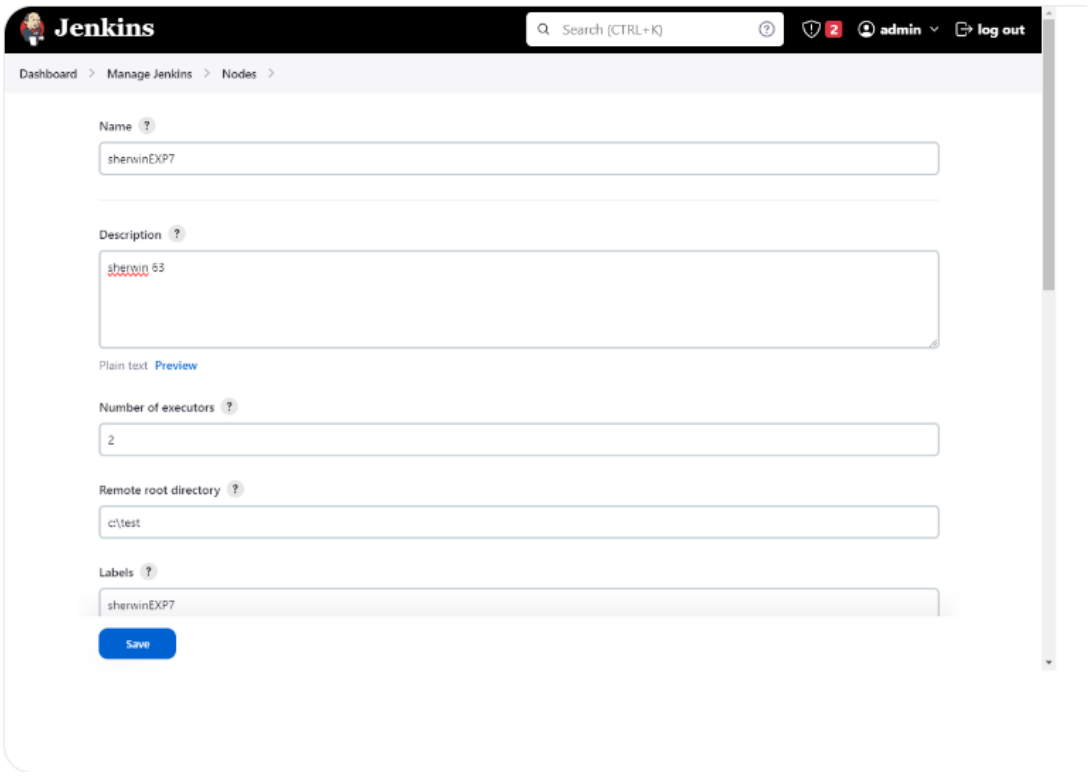
12 Click on New Node



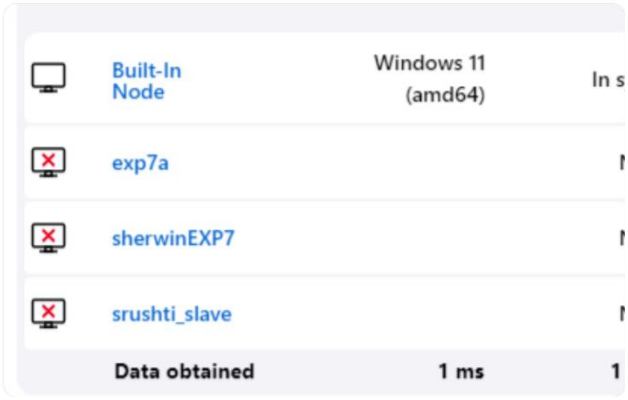
13 Select Permanent Agent



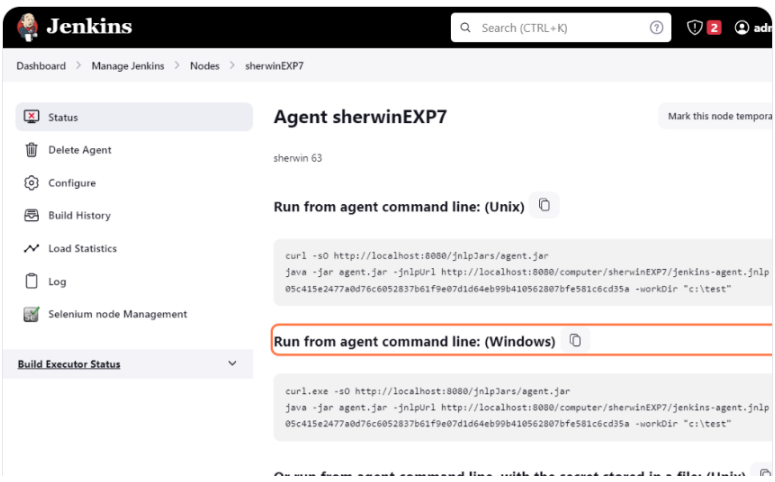
14 Click on highlight

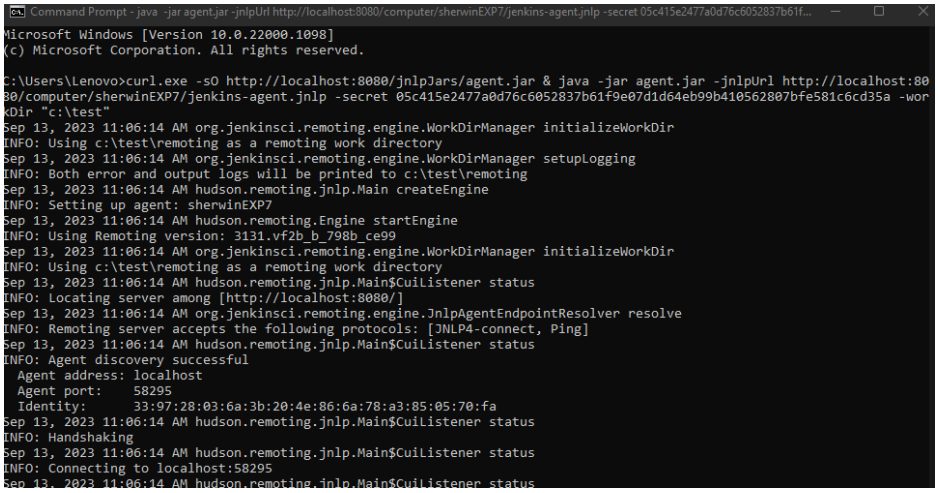


15 Click on Nodes...

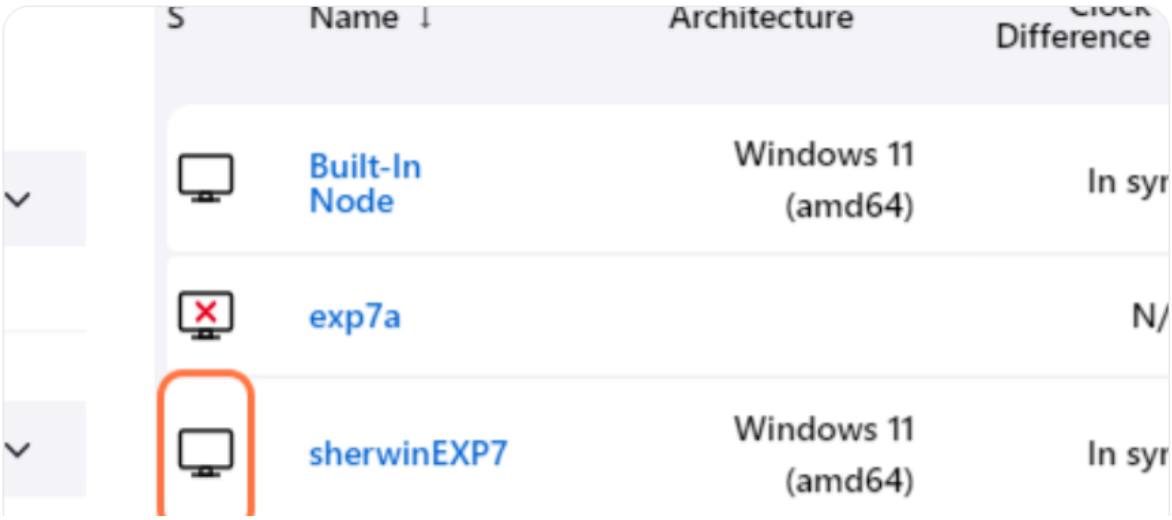


16 Click on Run from agent command line: (Windows)



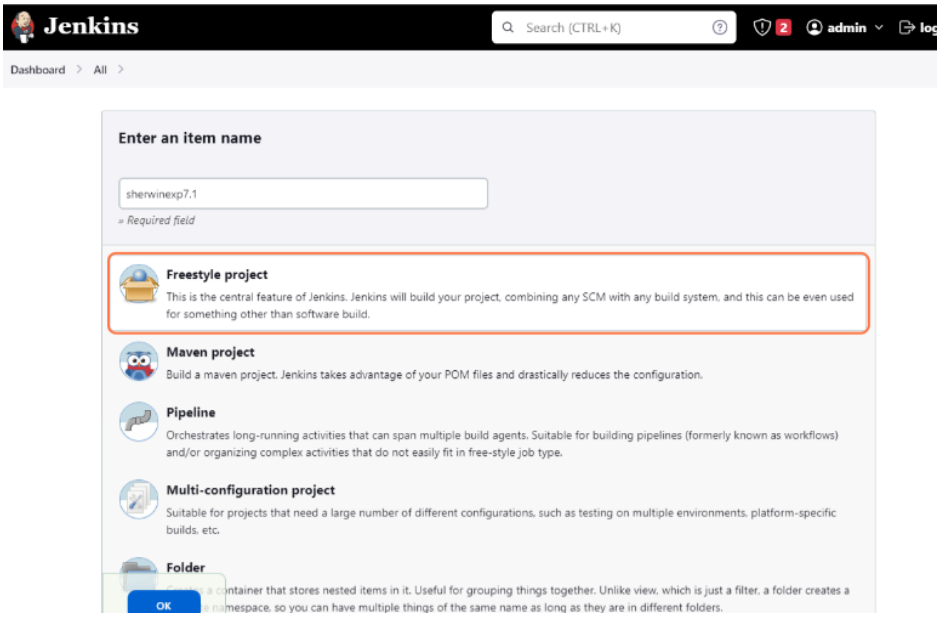


17 Click on [online]

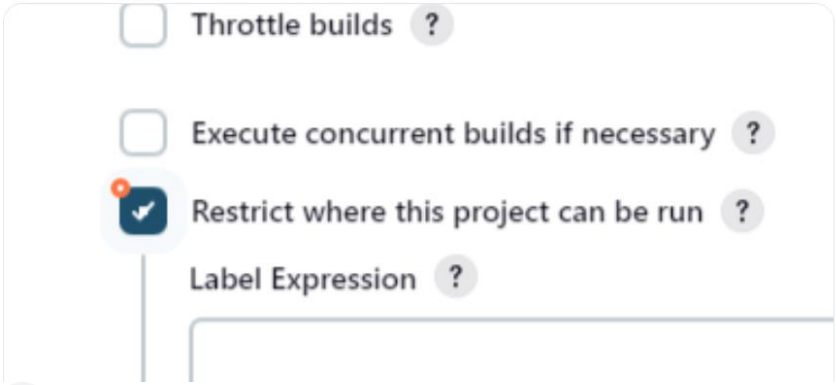


NEW FREE STYLE PROJECT

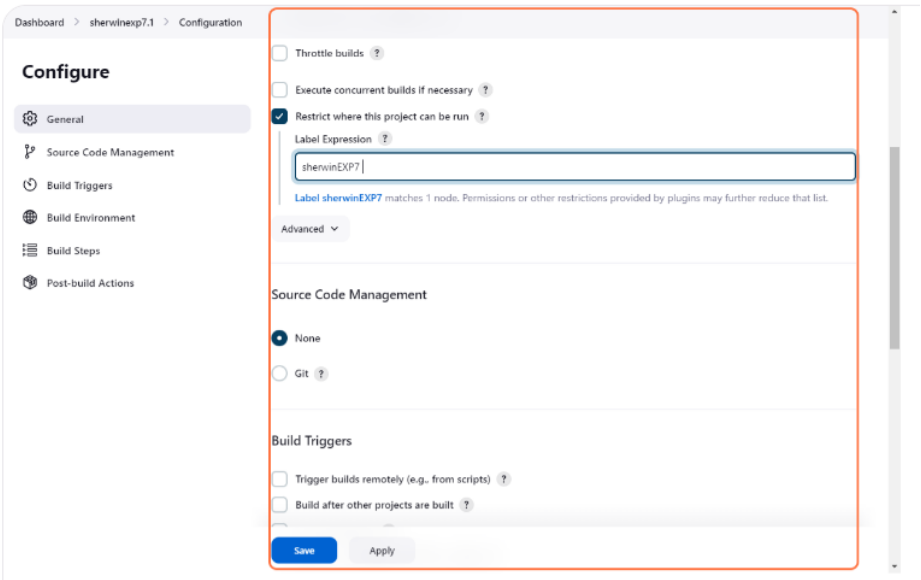
18 Click on Freestyle project...



19 Check hasSlaveAffinity



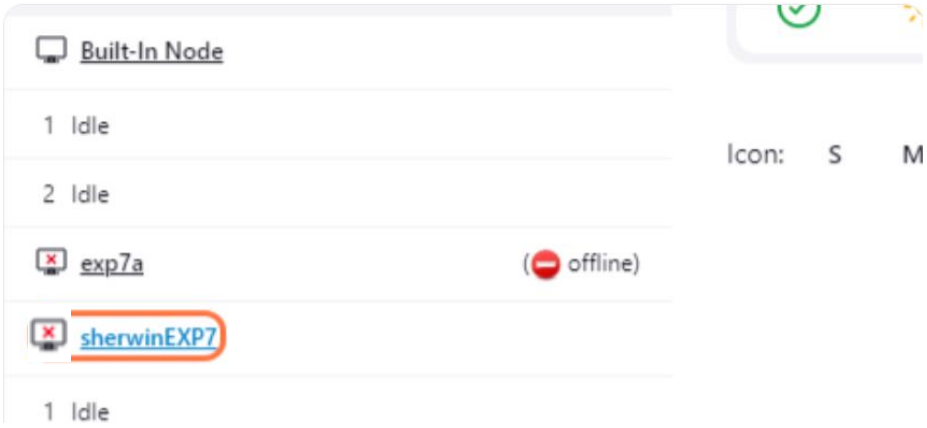
20 Click on General



22 Click on Console Output...

MAKE SLAVE NODE OFFLINE TEMPORARYILY AND MAKE IT ONLINE AGAIN

23 Click on sherwinEXP7



CREATED ANOTHER NODE TIED TO MY PROJECT