St. Francis Institute of Technology, Mumbai-400 103

**Department Of Information Technology**

A.Y. 2024-2025

Class: TE-ITA/B, Semester: V

Subject: **DevOps Lab**

**Experiment – 6: 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

1. **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.

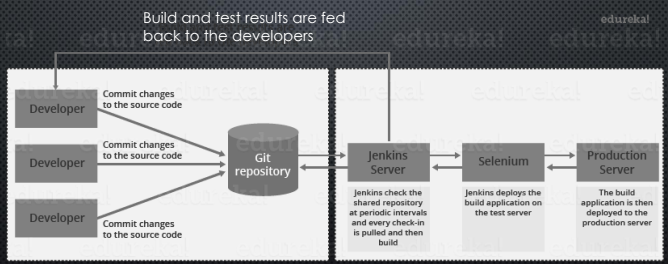
1. **Prerequisite:** Knowledge of Computer Networks concept of Master-slave architecture
2. **Requirements:** Jenkins,JDK, python,Personal Computer, Windows operating system, browser, Internet Connection, Microsoft Word.
3. **Pre-Experiment Exercise:**

**Brief Theory:** Refer shared material

1. **Laboratory Exercise**
   * + 1. **Procedure:**

**a. Answer the following:**

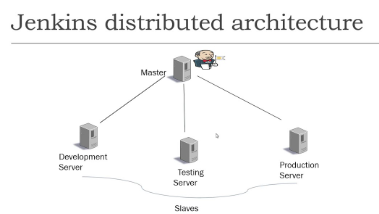
* Explain the architecture of Jenkins with diagram.



Jenkins is a widely used open-source automation server that supports continuous integration (CI) and continuous delivery (CD). The Jenkins architecture consists of the following components:

1. **Jenkins Master:**
   * **Primary Role:** It is responsible for orchestrating the complete CI/CD pipeline. The master controls the execution of jobs, schedules builds, and sends builds to the appropriate agents.
   * **Core Responsibilities:**
     + Schedule and manage jobs.
     + Monitor the build process and agents.
     + Provide a web UI for users to configure jobs and view results.
     + Centralize all plugin management and configuration.
2. **Jenkins Agent (Slave):**
   * **Primary Role:** Agents are machines that run the jobs sent by the master. These jobs can include compiling code, running tests, and deploying applications.
   * **Core Responsibilities:**
     + Execute tasks as per the master’s instructions.
     + Provide resources for running build tasks (CPU, memory, disk space).
     + Report build results back to the master.
3. **Job:**
   * A task or activity configured in Jenkins that gets executed by agents. Jobs can be for building code, running tests, deploying software, etc.
4. **Plugins:**
   * Jenkins’ functionalities can be extended through plugins, enabling support for version control, build tools, testing frameworks, and many more.
5. **Distributed Architecture:**
   * Jenkins uses a master-agent (or master-slave) architecture to achieve distributed builds. This allows Jenkins to offload build tasks to different agents, making it scalable for large projects.

* Explain the distributed architecture of Jenkins with diagram

****

Jenkins' distributed architecture allows for scaling across multiple machines to run jobs in parallel, improving performance, and handling large-scale builds. In this architecture, Jenkins Master and Jenkins Agents (Slaves) work together in a master-agent setup.

#### Components:

1. Jenkins Master:
   * Responsibilities:
     + Manages the configuration of the jobs.
     + Schedules jobs to be executed on agents based on load and availability.
     + Provides a web UI for configuration, monitoring, and reporting.
     + Stores job configurations, build artifacts, and logs.
     + Orchestrates the overall CI/CD pipeline, making sure agents get the jobs and collecting results.
2. Jenkins Agents (Slaves):
   * Responsibilities:
     + Execute build tasks that are assigned by the master.
     + Agents can be located on different machines to spread the load across the network, reducing resource contention.
     + Each agent has its own environment and can be configured to handle specific tasks such as running tests, building projects, etc.
     + After the build execution, agents send the build results back to the master.

**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
* Apply cron command on a project

1. **Post-Experiments Exercise**
2. **Extended Theory:**

Nil

1. **Questions:**

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

1. **Conclusion:**

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

1. **References:**

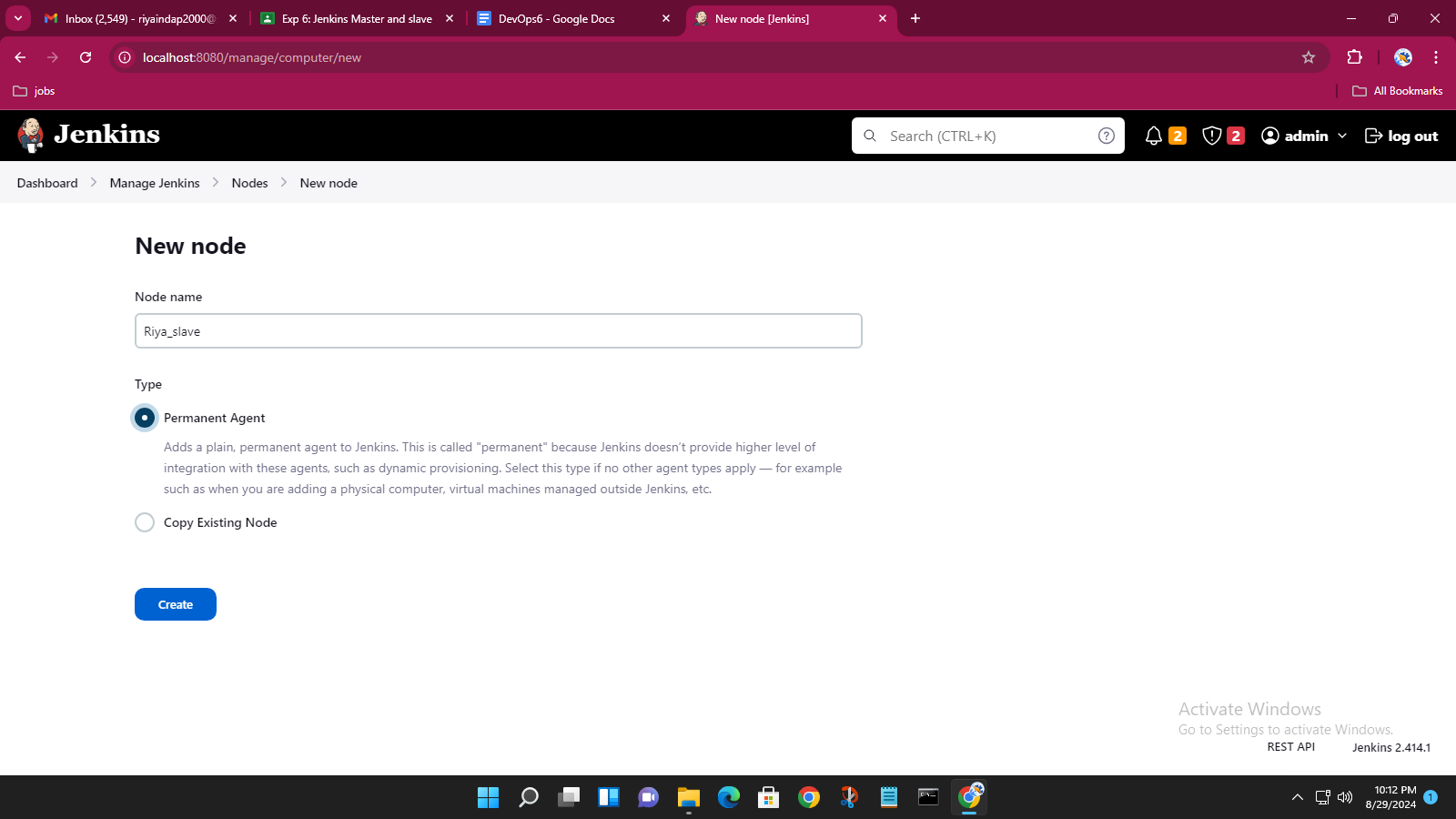
<https://jenkins.io/doc/>

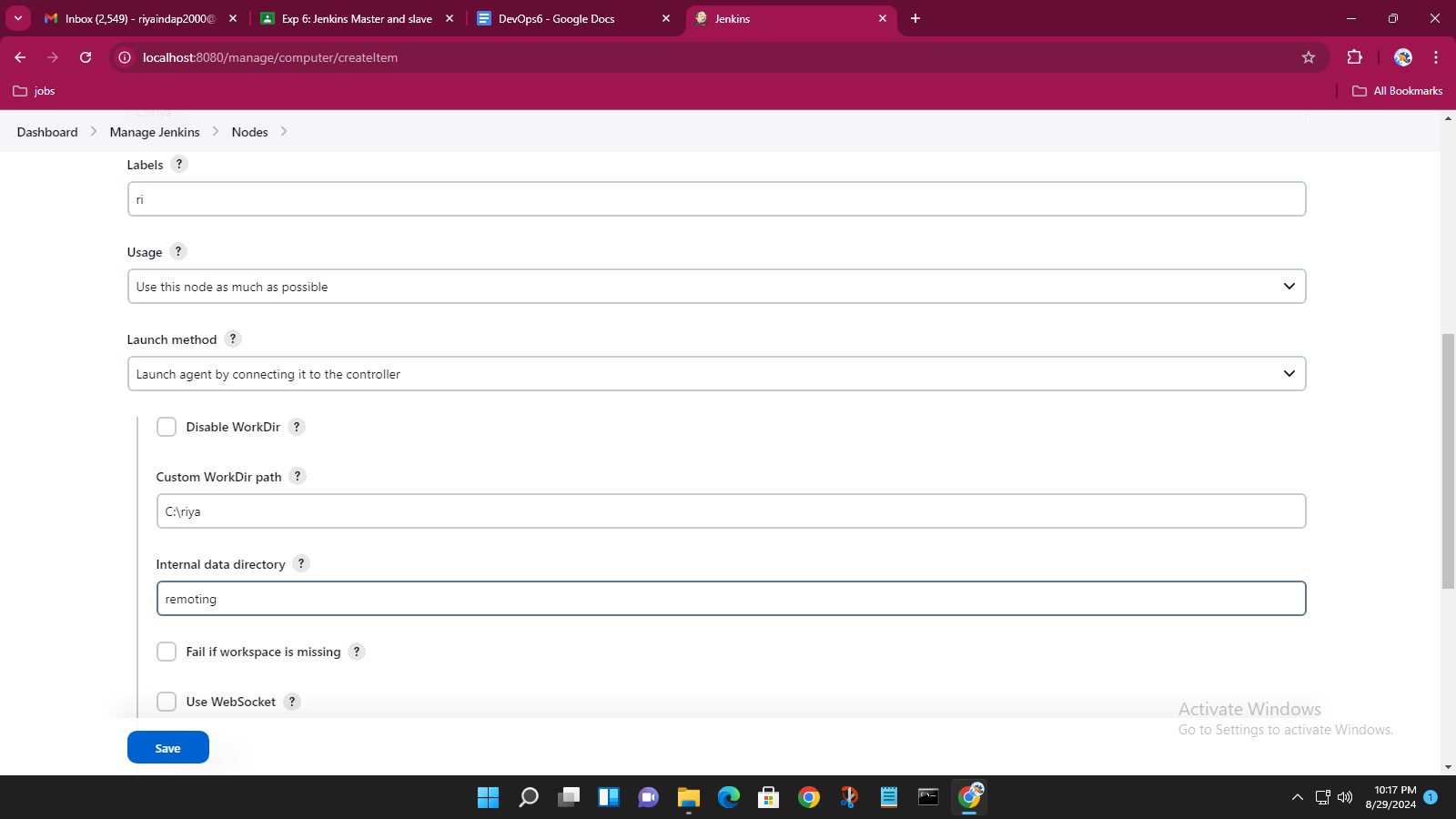
<https://www.slideshare.net/abediaz/introduction-to-jenkins>

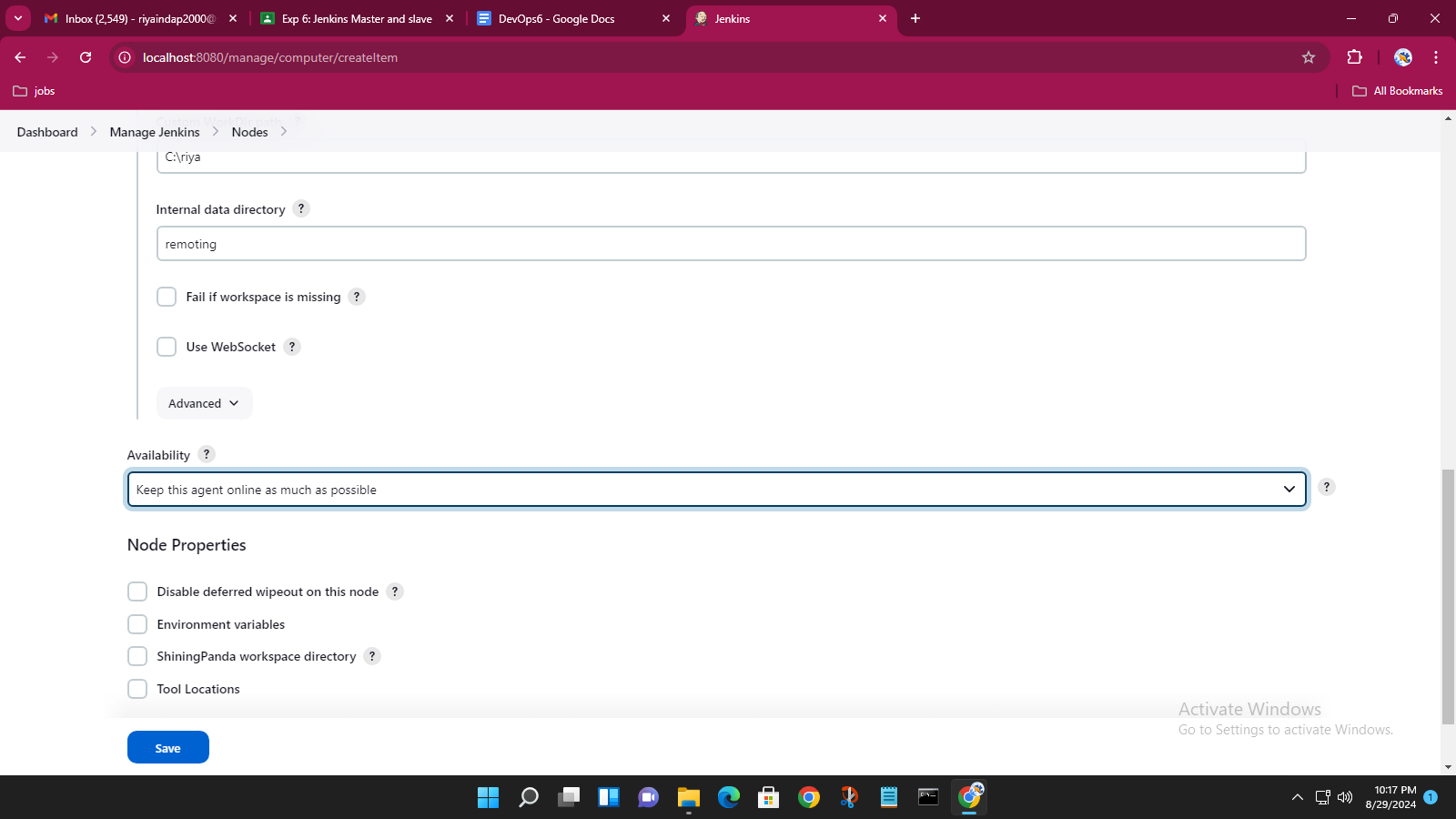
https://www.studytonight.com/jenkins/jenkins-master-slave-configuration

<https://www.edureka.co/blog/jenkins-master-and-slave-architecture-a-complete-guide/>

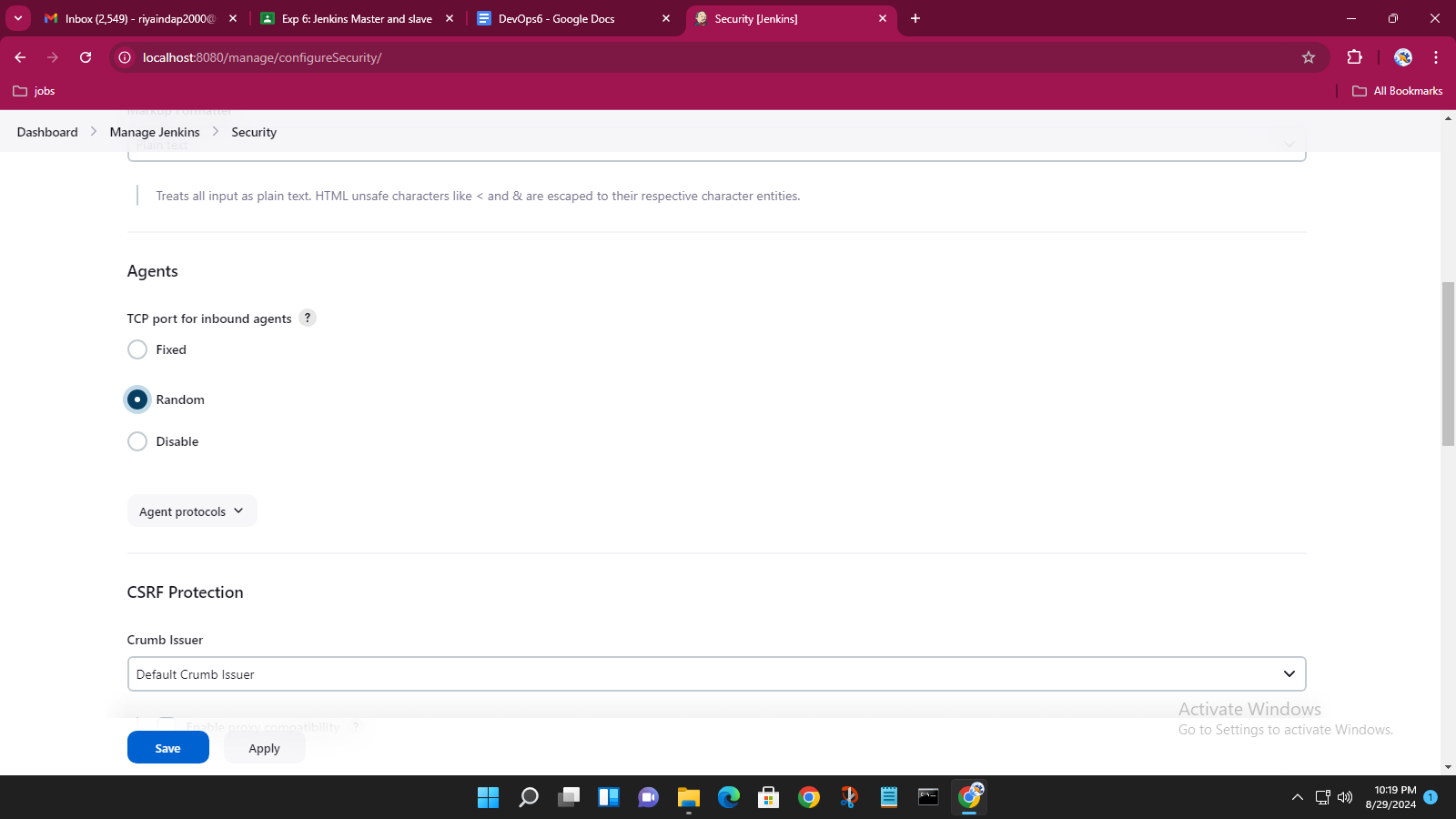
Creating a node:



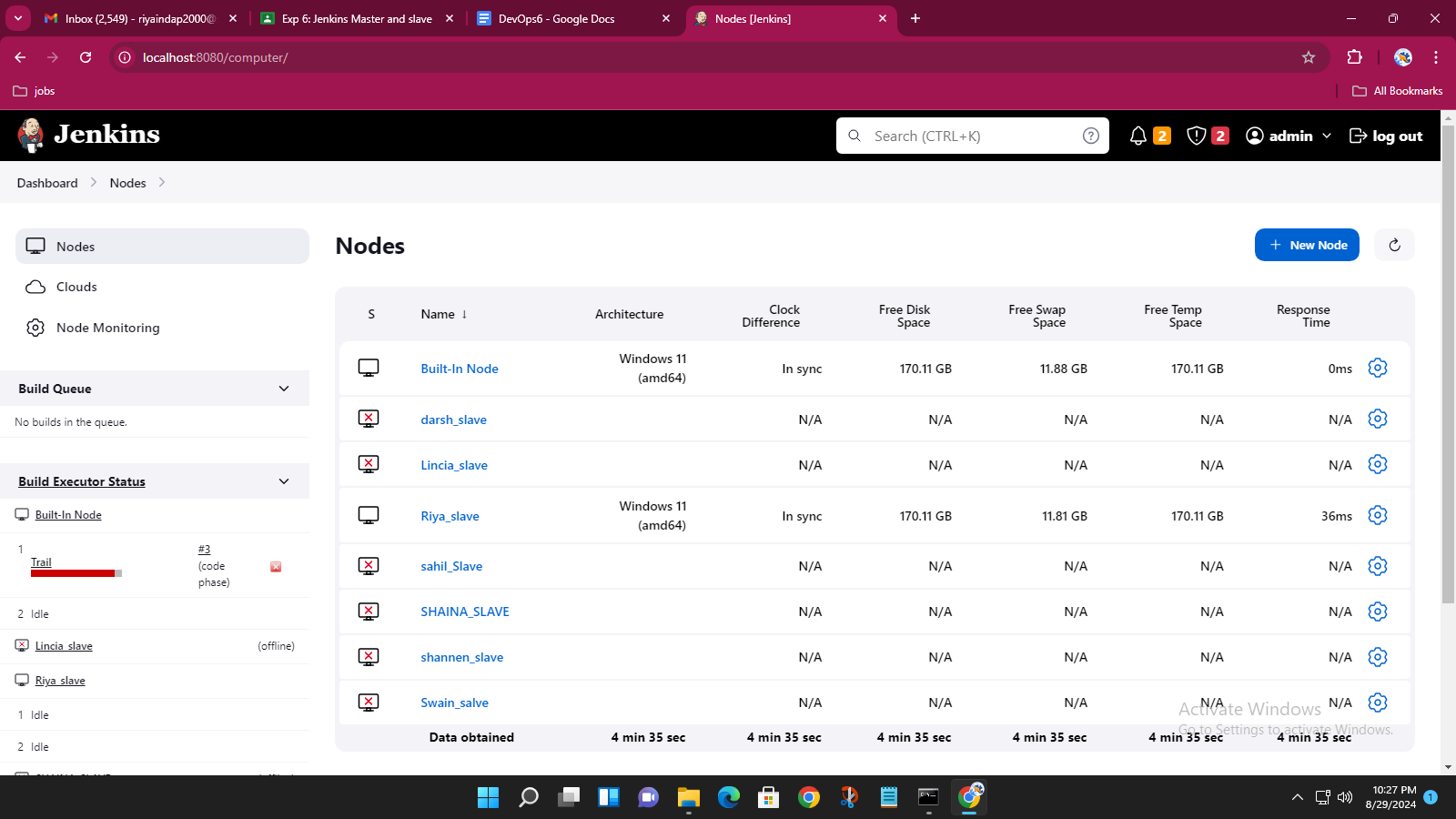
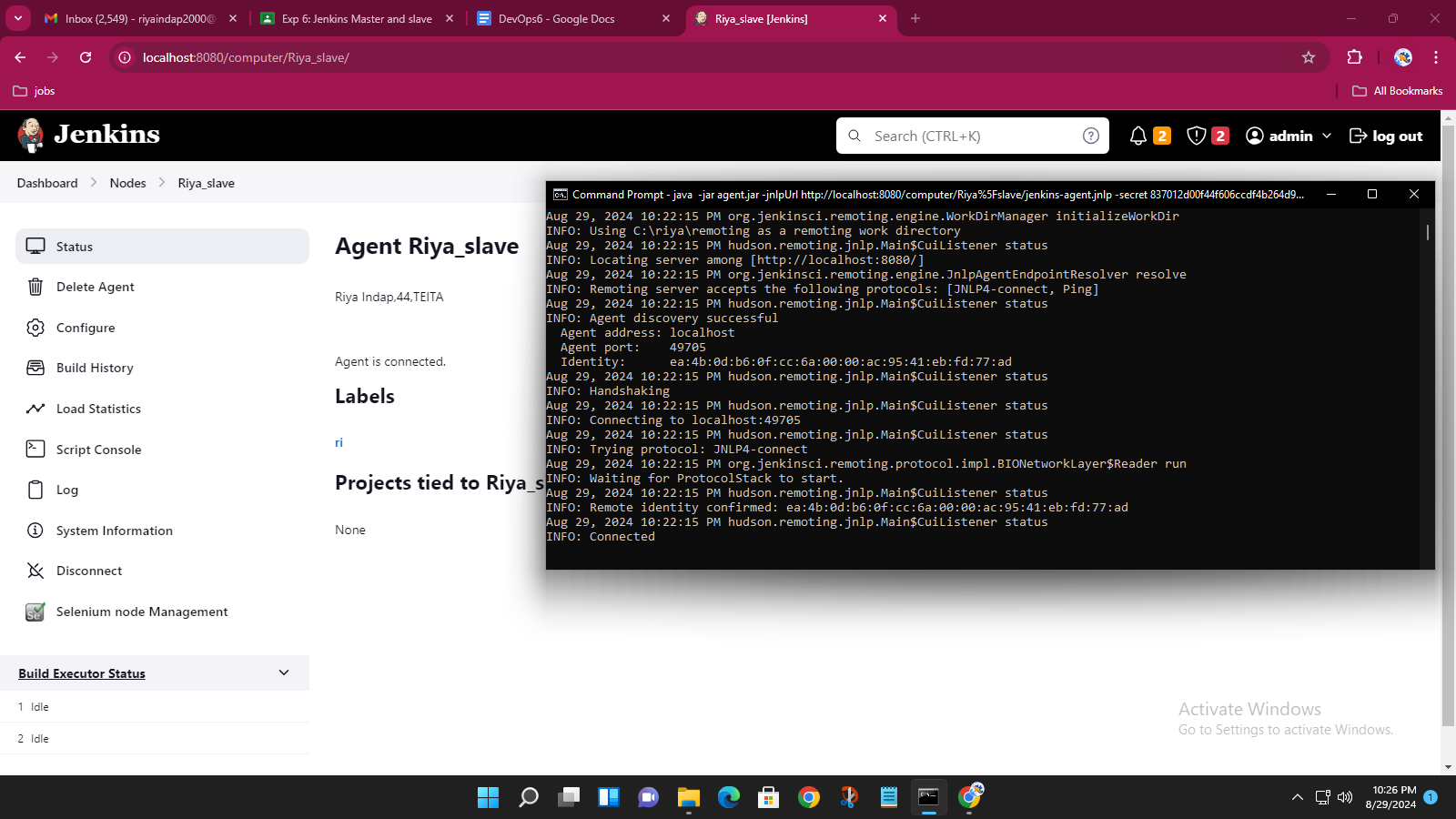


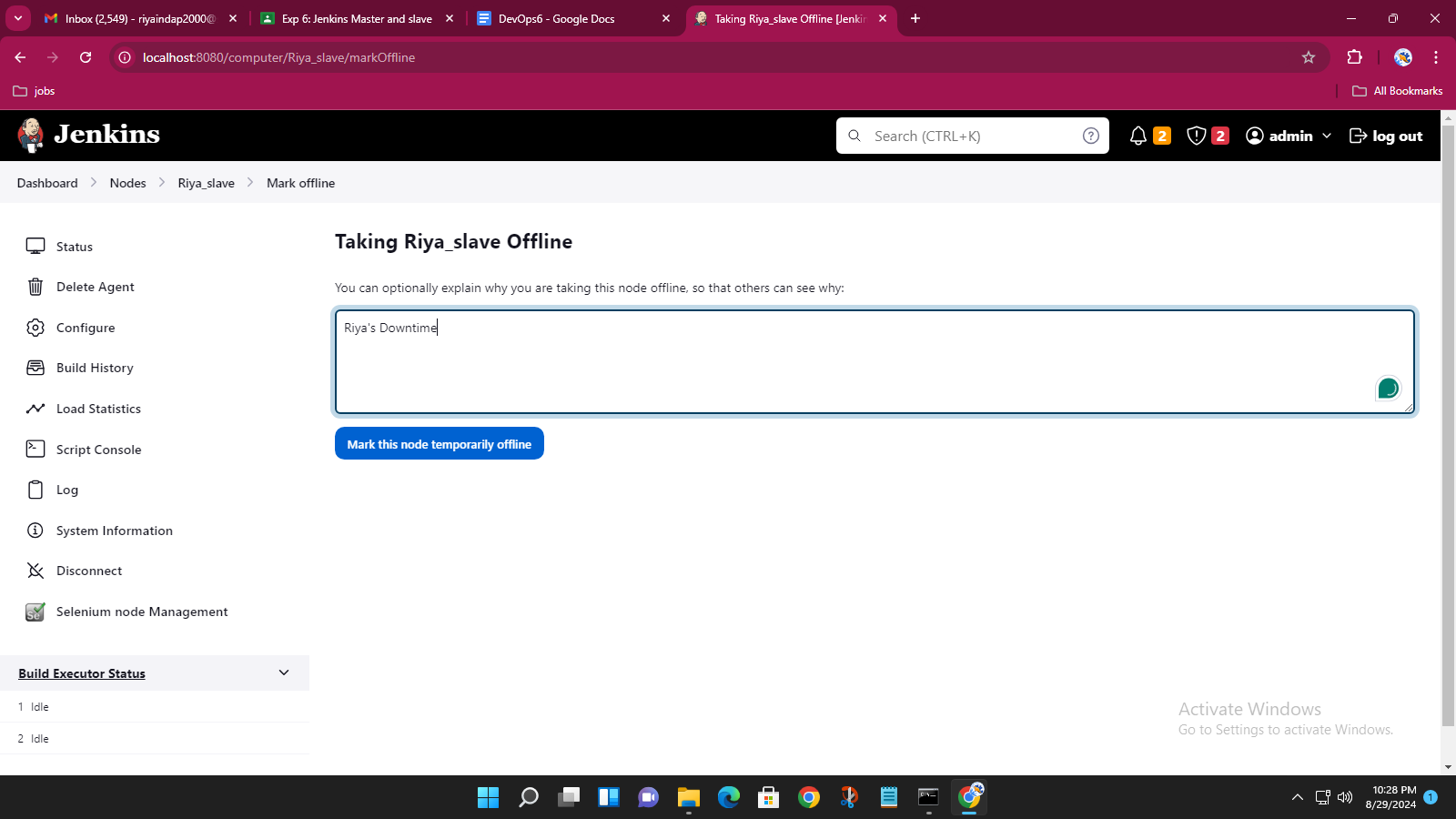


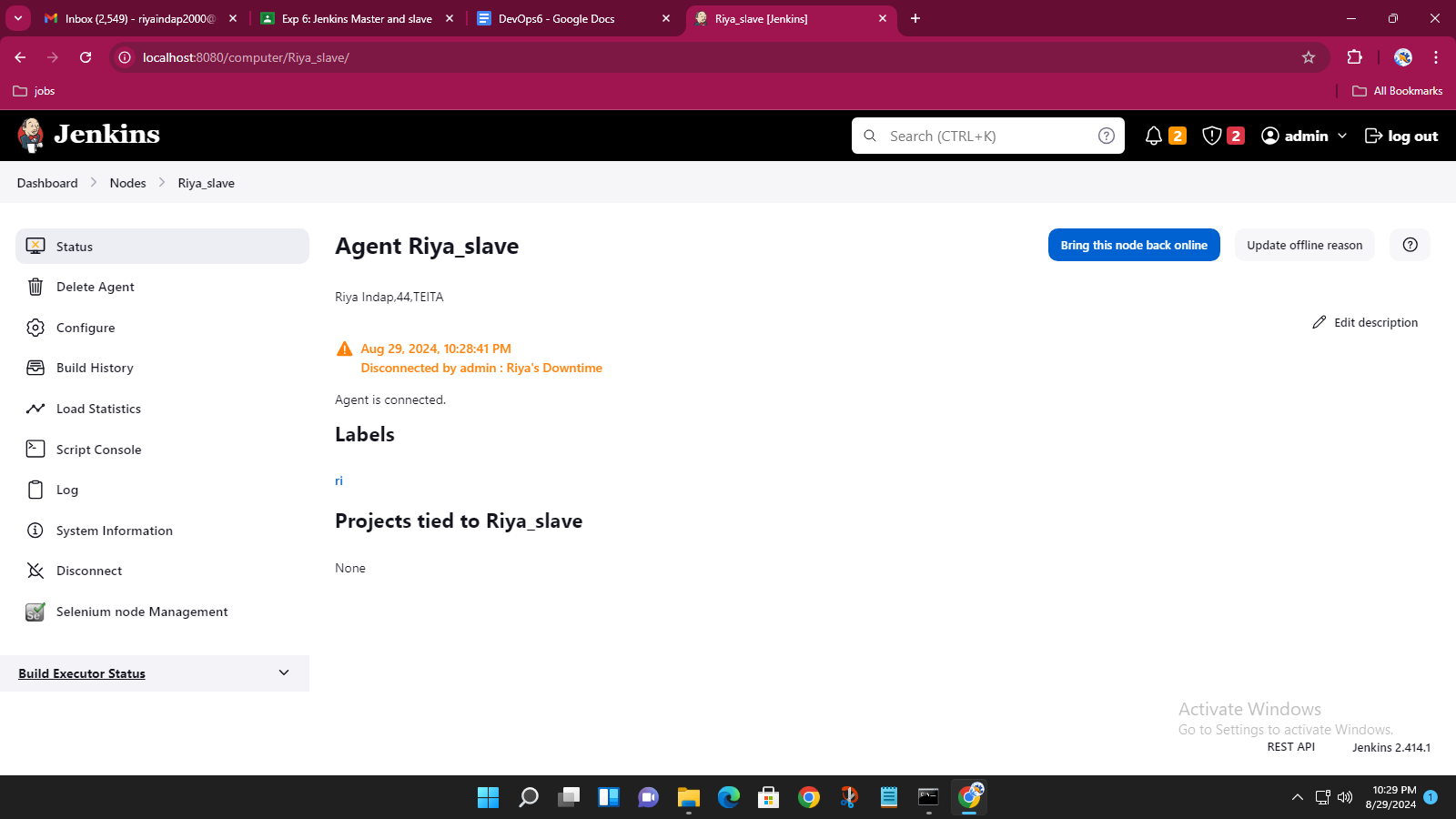
Enabling TCP/IP port for master and slave



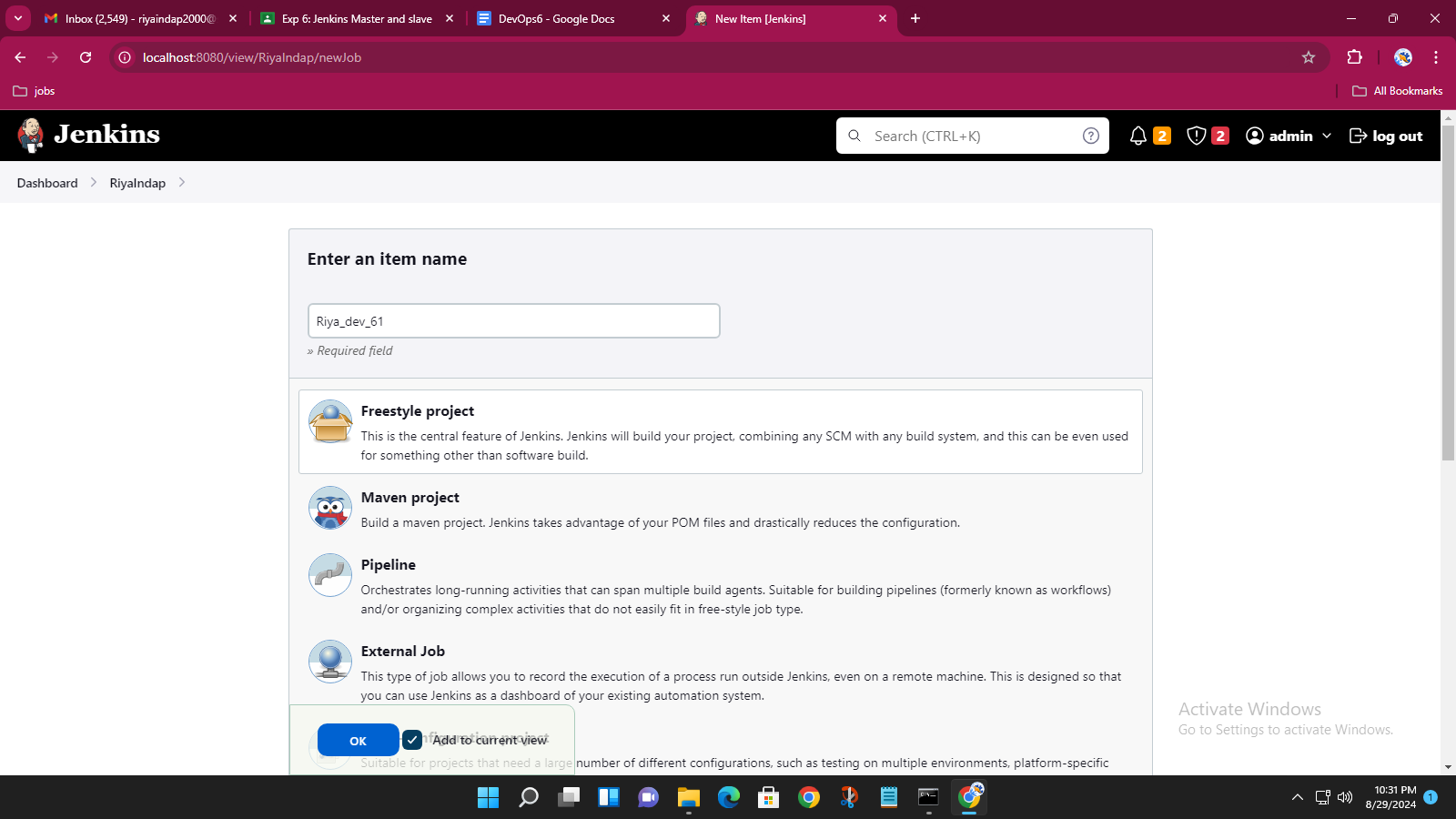
Java executable for slave is connected with the master

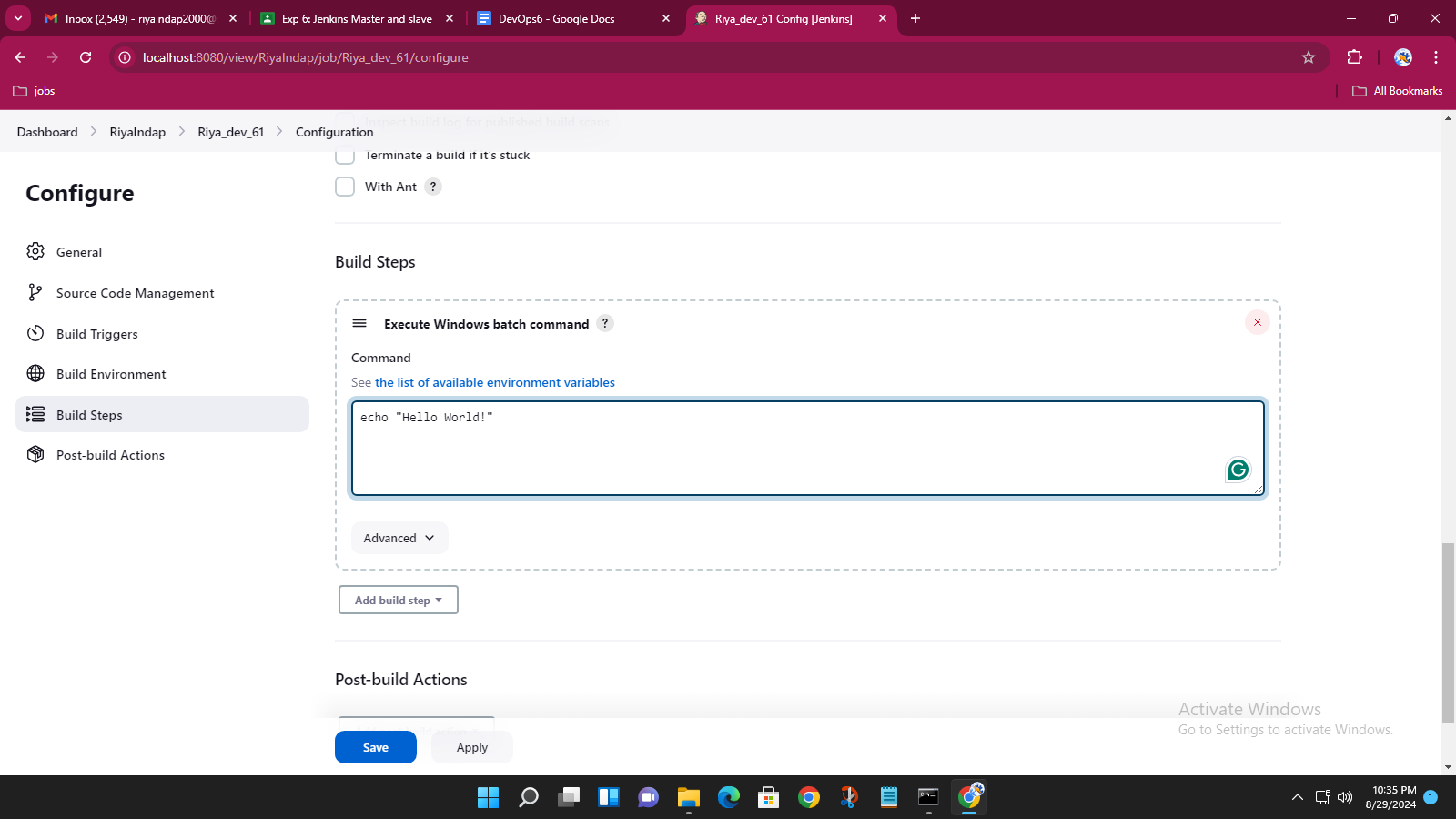
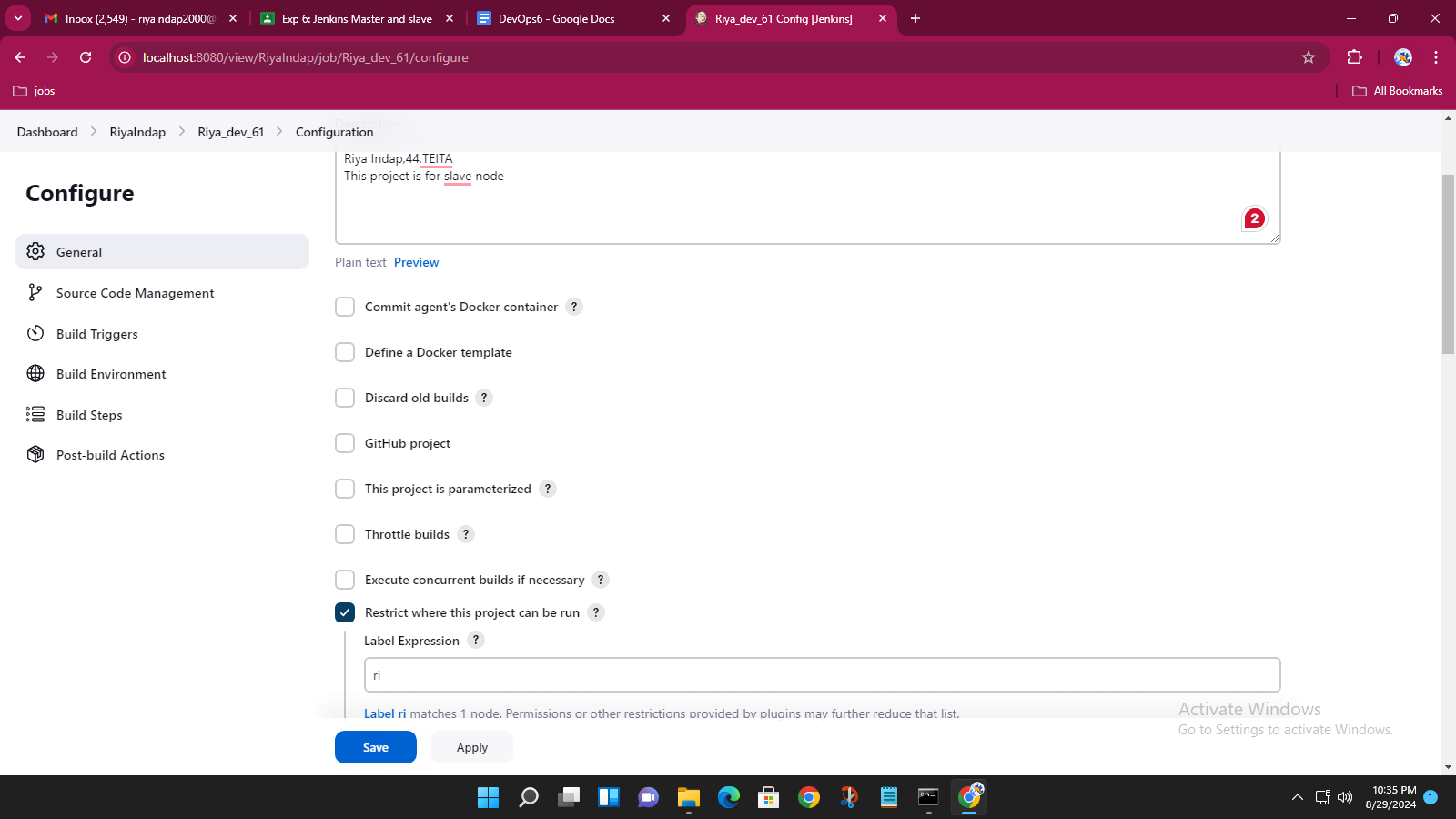


Marking the slave offline

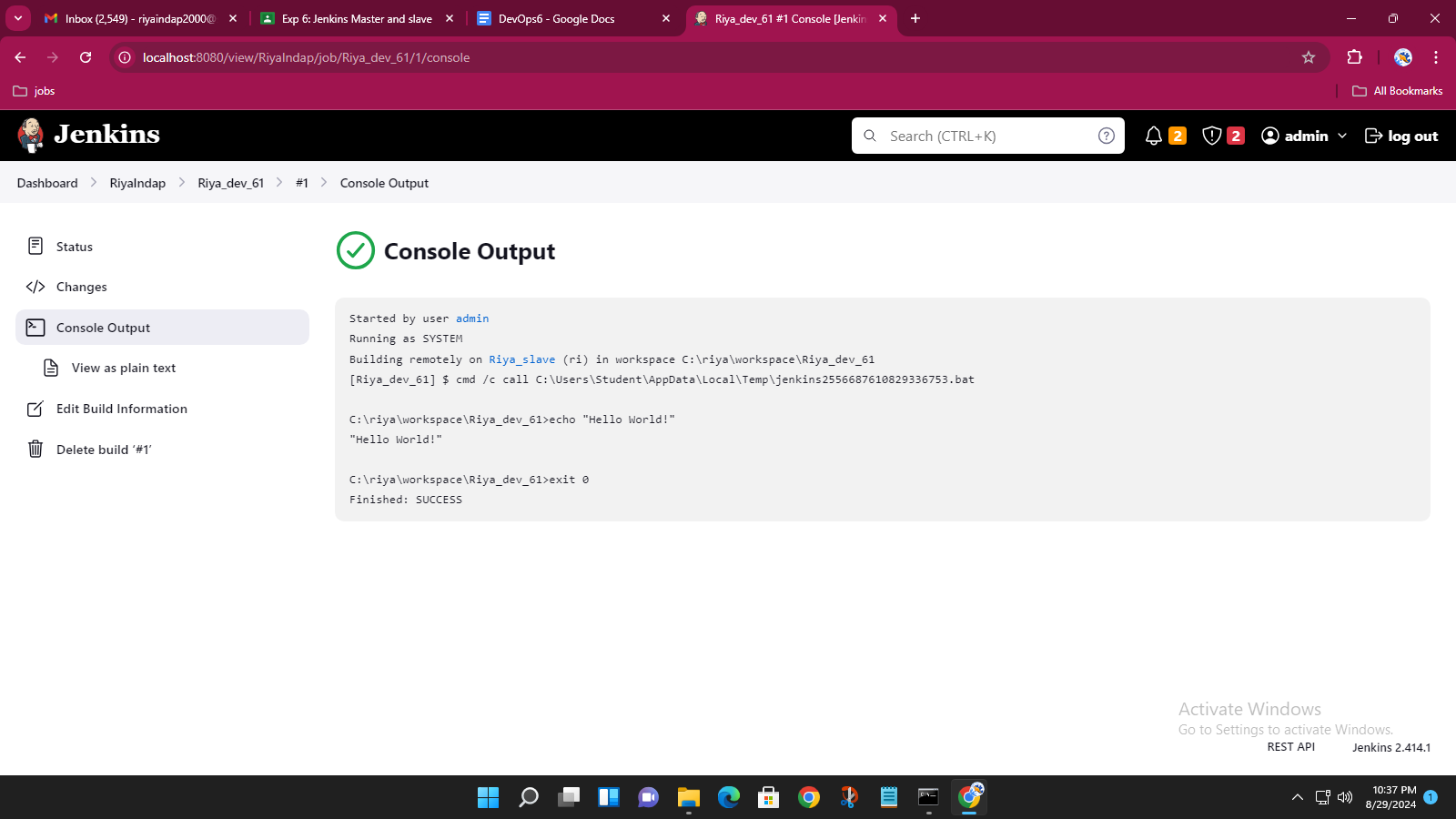


Now to give job to slave,creating a project

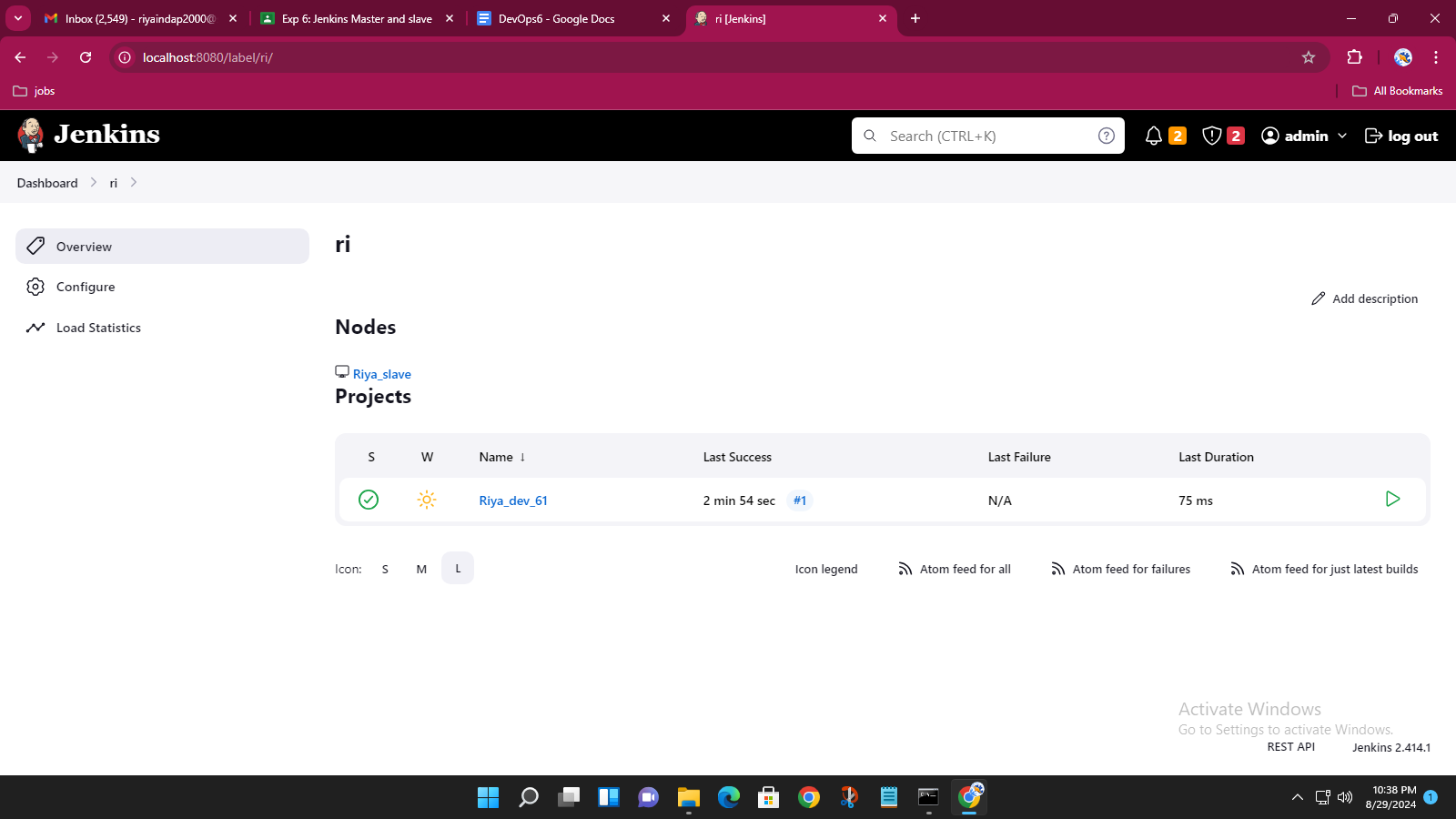




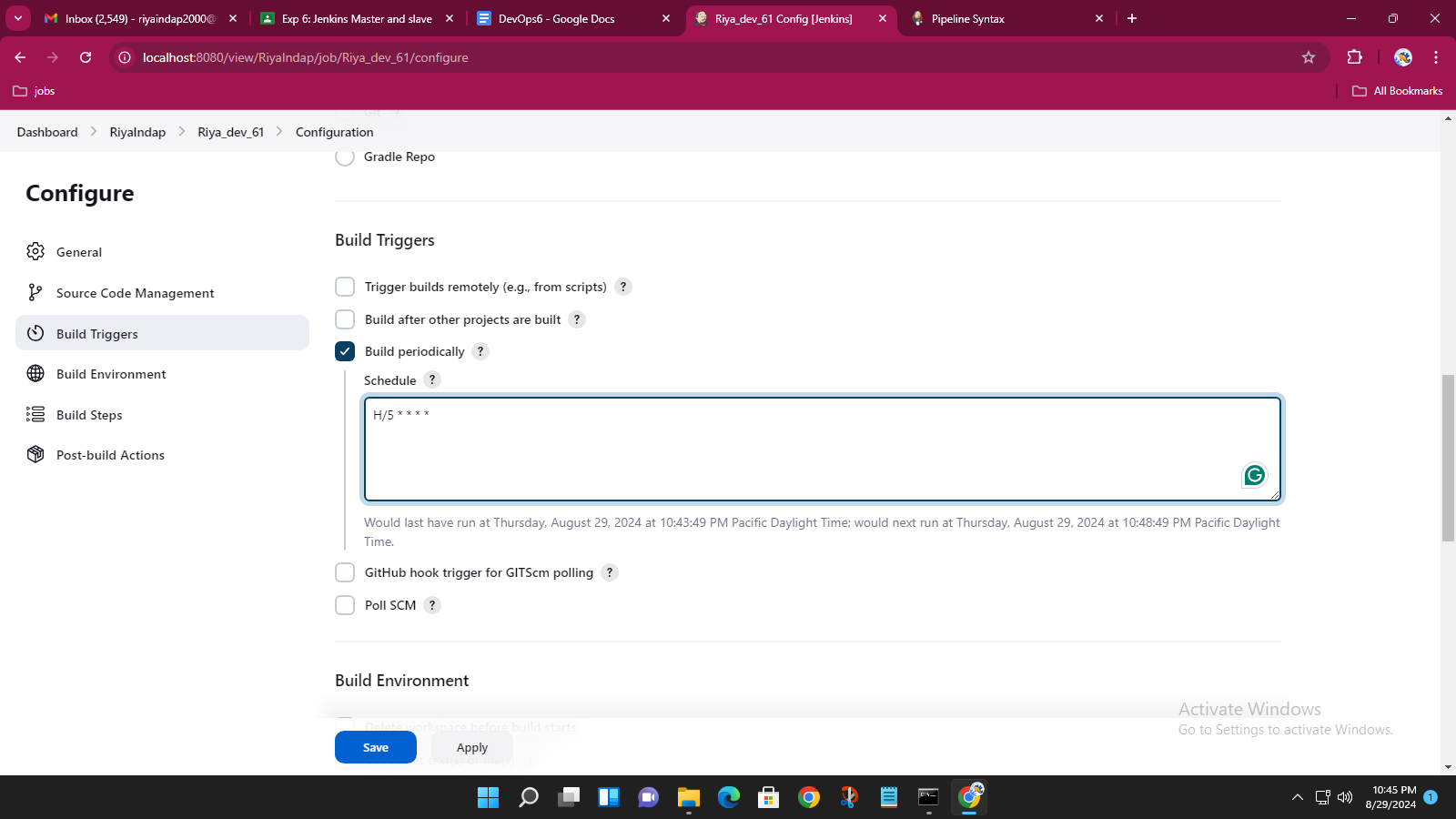
Project running on slave node



List of projects assigned to slave



Applying cron command to build the project every five minutes



Builds happening automatically because of cron cmd in every 5 minutes.

