We are planning to upgrade our jenkins prod server to higher version, so for that what are the parameters I should validate before and after upgrade

What steps should I follow to make the upgrade/restart process as seamless as possible?

Issue

- We need to restart Jenkins on a regular basis to upgrade Jenkins and its plugins.
- We need to patch Linux on a regular basis. Since our Jenkins is installed on Linux, we often need to restart Jenkins that has jobs running.

Resolution

safeRestart

The safeRestart function tells Jenkins what to do *before* restarting. It is built in Jenkins core (just navigate to /safeRestart); this will prevent any new builds from starting and will allow existing builds to complete before Jenkins restarts.

JVM

JVM is **the core of the Java ecosystem** and makes it possible for Java-based software programs to follow the **"write once, run anywhere"** approach. You can write Java code on one machine and run it on any other machine using the JVM. JVM was initially designed to support only Java.

JVM is not the part of OS. There are separate JVMs for each OS. You must install the JDK as per your OS. You are correct that the JVM differs between operating systems while the byte code generated from compiling a Java program is the same no matter which OS you compile it on.

Jenkins is an open-source continuous integration/continuous delivery and deployment (CI/CD) automation software DevOps tool **written in the Java programming language**. It is used to implement CI/CD workflows, called pipelines.

Upgrading Jenkins Java version from 8 to 11

There are a few details and steps to upgrade the JVM used to run Jenkins, more specifically from Java 8 to Java 11.

If you are upgrading the JVM used to run Jenkins, and particularly if you're upgrading from Java 8 to Java 11, there are some details you should know and precautions you should take.

Backup

As with any upgrade, we recommend backing up JENKINS_HOME and testing the upgrade with the backup before performing the upgrade on your production instance.

Upgrading Jenkins

If you need to upgrade Jenkins as well as the JVM, we recommend that you:

1. Back up JENKINS_HOME locally or to s3 bucket.
2. Upgrade Jenkins to the most recent version
    - How you upgrade Jenkins depends on how you installed Jenkins in the first place.
    - We recommend that you use the package manager of your system (such as apt or yum (this is in case you installed a war file Jenkins). In case it is a container Jenkins, go to docker hub and pull the latest Jenkins image and configure the volumes for the JENKINS_HOME and docker.sock. https://hub.docker.com/r/jenkins/jenkins
3. Validate the upgrade to confirm that all plugins and jobs are loaded.
4. Upgrade the required plugins.

Upgrading Plugins

It is important not just to upgrade Jenkins and the JVM but also to upgrade the plugins which support Java 11. Plugin upgrades assure compatibility with the most recent Jenkins releases.

Discovering issues with Java 11

Between June 2018 and February 2019, the community performed many exploratory tests to discover as many Java 11 issues as possible.

As a result, the community solved a lot of problems before announcing Java 11 support in Jenkins. However, it is still possible that some plugins haven't been updated to support Java 11.

Compatibility issues are being tracked in the Jenkins issue tracker as Known Java 11 Compatibility Issues.

If you discover a Java 11 incompatibility, please report the issue in the Jenkins bug tracker using the java11-compatibility label.

For security issues, please use the standard vulnerability reporting process.

Javax XML Bind libraries

Some plugins use JAXB libraries provided by the JDK. However, the java.xml.bind and javax.activation modules are no longer included in OpenJDK 11, and plugins might fail if no replacement is offered.

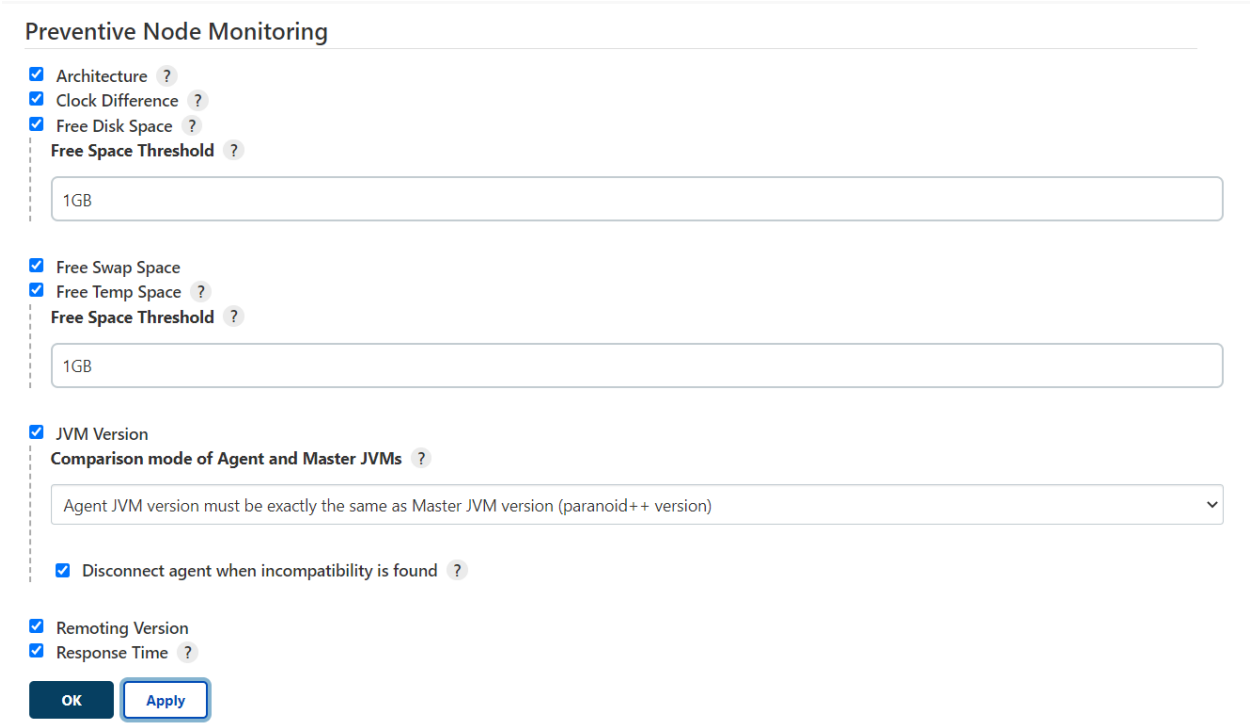
To fix this problem, we've bundled those libraries into a new detached plugin: JAXB plugin. When any Jenkins core more recent than 2.163 is running on Java 11, this plugin is automatically installed. However, if you manage your plugins outside Jenkins (for example, if you use plugins.txt in your Docker images), you might need to install the plugin explicitly.

JVM version on agents

All agents must be running on the same JVM version as the controller (because of how controllers and agents communicate). If you're upgrading your Jenkins controller to run on Java 11, you also need to upgrade the JVM on your agents.

You can validate the version of each agent with the Versions Node Monitors plugin. This plugin provides information about the JVM version of each agent on the node management screen of your Jenkins instance. You can also configure this plugin to automatically disconnect any agent with an incorrect JVM version.

Versions Node Monitors plugin



Java Web Start

Connect agents to Jenkins on Java 11 with plugins like SSH Build Agents Plugin, with operating system command line calls to java -jar agent.jar, or using containers

5. Make a second backup of JENKINS_HOME after upgrading Jenkins and the required plugins
6. Stop the Jenkins instance
7. Upgrade the JVM on which Jenkins is running
    - Use a package manager to install the new JVM.
    - Make sure the default JVM is the newly installed version. If it is not, run systemctl edit jenkins and set either
      the JAVA_HOME environment variable or
      the JENKINS_JAVA_CMD environment variable.

If you use Docker containers to run Jenkins, the default Docker containers use Java 11 beginning with Jenkins 2.303.1. If you need to remain with Java 8 in a Docker container, append -jdk8 to the Docker image tag.