This is a solid proof of concept for setting up Jenkins on RHEL. To enhance it further and clarify certain parts, here’s a more detailed and structured breakdown of the steps.

Jenkins Proof of Concept for Setup on RHEL

Prerequisites

1. **Operating System:** RHEL (Red Hat Enterprise Linux)
2. **Java:** JDK 17 (Make sure you have this version installed for Jenkins compatibility)
3. **Network Access:** Ensure access to Jenkins repositories and external servers for plugins.
4. **SSH Access:** Set up SSH keys for agent nodes.

Step 1: User Creation

Create a dedicated user to run Jenkins:

```
sudo useradd -m jenkins_user
sudo passwd jenkins_user
```

This isolates Jenkins processes for security and ease of management.

Step 2: Install Java

Install OpenJDK 17 on the Jenkins server:

```
sudo yum install -y java-17-openjdk
```

Verify the installation:

```
java -version
```

Ensure that the output shows `openjdk version "17.x.x"`.

Step 3: Install Fontconfig (Optional)

This step is only needed for Debian-based systems. If you're using RHEL, you can skip it. However, if needed:

For Debian-based systems:

```
sudo apt update
sudo apt install -y fontconfig
```

Step 4: Install Jenkins

1. **Create the Jenkins installation directory:**

```
mkdir -p /applications/jenkins_dir
cd /applications/jenkins_dir
```

2. **Download the Jenkins WAR file:**

```
wget https://get.jenkins.io/war-stable/2.462.2/jenkins.war
```

3. **Create a directory for Jenkins logs:**

```
mkdir -p /applications/jenkins_dir/jnlogs
```

Step 5: Configure Jenkins Start Script

Create a script to start Jenkins:

```
#!/bin/bash

JAVA_HOME=/applications/java_dir/jdk-17.0.11
JENKINS_WAR=/applications/jenkins_dir/jenkins.war
JENKINS_HOME=/applications/jenkins_dir/jenkins_data
LOGGING_PROPERTIES=/applications/jenkins_dir/jnlogs/logging.properties
LOG_FILE=/applications/jenkins_dir/jnlogs/log1.log

$JAVA_HOME/bin/java -DJENKINS_HOME=$JENKINS_HOME -Djava.util.logging.config.file=$LOGGING_PROPERTIES -jar $JENKINS_WAR --httpPort=8080 > $LOG_FILE 2>&1 &
```

Make the script executable:

```
chmod +x start_jenkins.sh
```

This script will start Jenkins on port 8080 and output logs to the specified log file.

Step 6: Configure Logging

Create a simple logging configuration file:

```
cat <<EOL > /applications/jenkins_dir/jnlogs/logging.properties
handlers= java.util.logging.ConsoleHandler
java.util.logging.ConsoleHandler.level= INFO
java.util.logging.ConsoleHandler.formatter = java.util.logging.SimpleFormatter
EOL
```

This logging setup will ensure that Jenkins logs are captured for troubleshooting.

Step 7: Create Jenkins Stop Script

Create a script to stop Jenkins:

```
#!/bin/bash

stop_jenkins() {
    PID=$(ps aux | grep '[j]enkins.war' | awk '{print $2}')
    if [ -n "$PID" ]; then
        echo "Stopping Jenkins (PID: $PID)..."
        kill "$PID"
        sleep 10
        if ps -p "$PID" > /dev/null; then
            echo "Forcing stop..."
            kill -9 "$PID"
        else
            echo "Jenkins stopped gracefully."
        fi
    else
        echo "No Jenkins process found."
    fi
}

stop_jenkins
```

Make the script executable:

```
chmod +x stop_jenkins.sh
```

This script will gracefully stop Jenkins, or force it to stop if necessary.

Step 8: Check Jenkins Status

To verify that Jenkins is running, check for the Jenkins process:

```
ps aux | grep '[j]enkins.war'
```

Step 9: Setting Up Jenkins Users

After starting Jenkins, navigate to `http://<your_server_ip>:8080` in a browser. Follow the setup wizard to create the admin user and configure Jenkins.

Step 10: Configure Master-Agent Setup

SSH Configuration

Ensure SSH access between the Jenkins master and agent nodes:

```
ssh-keyscan <agent_node_ip> >> ~/.ssh/known_hosts
```

This will add the agent node to the master node's `known_hosts`.

Install Java on Agent Node

Install Java on the agent node:

```
sudo yum install -y java-17-openjdk
```

Launch Jenkins Agent via SSH

- Set up the agent node in Jenkins by adding its SSH credentials in the Jenkins interface.
- Ensure that the Jenkins master’s public key is added to the agent node’s `~/.ssh/authorized_keys` file.

Step 11: Install Git on Agent Node

To enable Jenkins to pull from source control, install Git on the agent node:

```
sudo yum install -y git
```

Step 12: Bitbucket Integration

Integrate Jenkins with Bitbucket for source control:

1. Set up webhooks in Bitbucket to trigger Jenkins builds.
2. In Jenkins, configure the Bitbucket plugin to pull code from your repository.
3. Add the necessary credentials (SSH keys, API tokens) in Jenkins to authenticate with Bitbucket.

Conclusion

This proof of concept outlines the basic steps to set up Jenkins in an enterprise environment, including user creation, Jenkins installation, Java setup, logging configuration, agent setup, and Bitbucket integration.

```
    --------------------------MANI KANTA BANDLA------------------
```