Streamlining Linux Workflows: Easy Docker and Jenkins Setup Guide

Certainly! Before jumping into the installation process, let's take a closer look at some concepts.

What is a Package?

A package is a bundle of software files and metadata. This bundle includes everything necessary to install and manage an application on a Linux system.

What exactly is metadata?

Metadata is like a label on a jar that tells you what's inside.

Package Metadata (in Linux):

Package Name: The name of the software package.

Version: The version number of the software.

Dependencies: Other packages required for the software to work.

Description: A summary of what the package does.

Metadata makes data more understandable and easier to work with by providing essential information about its characteristics and context.

What is a Package Manager?

A package manager is a tool that allows users to install, remove, upgrade, configure and manage software packages on an operating system. The package manager can be a graphical application like a software center or a command line tool like apt-get or pacman.

Different kinds of package managers:

Different package managers exist because different Linux distributions have unique needs and preferences, leading to the development of tools customized for each distribution. The variety of packages in Linux shows how flexible and open the system is. Users can pick the tools that suit them best, which encourages new ideas and improvements in the Linux community.

Commonly Used Package Managers:

APT (Advanced Package Tool): It is used in Debian-based distributions like Ubuntu and Linux Mint.

YUM (Yellowdog Updater, Modified) / DNF (Dandified YUM): Used in Red Hat-based distributions such as CentOS, Fedora, Red Hat Enterprise Linux (RHEL), and their derivatives. DNF is the newer version used in Fedora and CentOS 8+.

Pacman (Package Manager): Widely used in the Arch Linux community and its derivatives like Manjaro . It is popular for its simplicity and speed.

All package managers handle:

Dependency Resolution: Ensuring all necessary software components are installed for proper functioning.

Repository Management: Accessing online repositories to install trusted software from official and third-party sources.

Package Management: Handling package operations, including installation, upgrades, and removals, and maintaining system stability through version locking.

Cache Management: Storing recently downloaded packages locally to speed up future installations and updates.

How to install docker and Jenkins using package managers on Ubuntu:

You can install docker just by using below two terminal commands.

Step 1: Update Package Index

Updating the package index (sudo apt update) helps your system stay up to date with the latest versions of available packages.

```
sudo apt update
```

Step 2:

```
sudo apt install docker.io -y
```

This command installs Docker along with its dependencies without requiring.

manual confirmation.

That's it! With these two commands, you'll have Docker up and running on your Ubuntu system.

Jenkins Installation on Ubuntu:

Jenkins is a Java-based application. It is built on top of the Java Virtual Machine (JVM) and requires Java to run. Java provides the necessary runtime environment for Jenkins to execute its processes and manage the automation of software development tasks, such as building, testing, and deploying applications. Therefore, installing Java on the system is a prerequisite for setting up Jenkins.

Step 1: Install Java

```
sudo apt install default-jdk -y
```

Step 2: Install Jenkins

```
wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key | sudo apt-key add -
```

```
sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'

sudo apt update

sudo apt install jenkins -y
```

wget -q -O -https://pkg.jenkins.io/debian/jenkins.io.key| sudo apt-key add -: This command fetches the Jenkins repository key and adds it to the system's list of trusted keys, allowing for secure installation.

sudo sh -c 'echo debhttp://pkg.jenkins.io/debian-stablebinary/ > /etc/apt/sources.list.d/jenkins.list': This adds the Jenkins repository to the system's list of package sources.

sudo apt update: This updates the package index to include the newly added Jenkins repository.

sudo apt install jenkins -y: This installs the Jenkins package. The -y flag automatically confirms any prompts during installation.

Step 3: Start Jenkins

```
sudo systemctl start jenkins

sudo systemctl enable Jenkins
```

sudo systemctl start jenkins: This starts the Jenkins service.

sudo systemctl enable jenkins: This configures Jenkins to start automatically at boot time.

Jenkins is now installed and ready to use! You can access Jenkins by navigating to http://localhost:8080 in your web browser.

Installing Docker on CentOS:

Step 1: Update Package Index

```
sudo yum check-update
```

Step 2: Install Docker

```
sudo yum install -y yum-utils device-mapper-persistent-data lvm2

sudo yum install docker-ce docker-ce-cli containerd.io -y
```

- sudo yum install -y: Installs packages without asking for confirmation (-y flag).
- yum-utils device-mapper-persistent-data lvm2: Installs necessary dependencies

  for Docker.
- docker-ce docker-ce-clicontainerd.io: Installs Docker components

Step 3: Start and Enable Docker Service

```
sudo systemctl start docker

sudo systemctl enable docker
```

Installing Jenkins on CentOS:

Step 1: Install Java

```
sudo yum install java-1.8.0-openjdk-devel -y
```

- sudo yum install: Installs packages.
- java-1.8.0-openjdk-devel: Installs the OpenJDK Development Kit version 8.

Step 2: Import Jenkins GPG Key and Add Repository

```
sudo rpm --import https://pkg.jenkins.io/redhat/jenkins.io.key

sudo wget -O /etc/yum.repos.d/jenkins.repo
https://pkg.jenkins.io/redhat-stable/jenkins.repo
```

- sudo rpm --import: Imports the Jenkins GPG key for package verification.
- sudo wget -O: Downloads the Jenkins repository configuration file.

Step 3: Install Jenkins

```
sudo yum install jenkins -y
```

- sudo yum install: Installs the Jenkins package.

Step 4: Start and Enable Jenkins Service

```
sudo systemctl start Jenkins

sudo systemctl enable Jenkins
```

- sudo systemctl start jenkins: Starts the Jenkins service.
- sudo systemctl enable jenkins: Sets Jenkins to start automatically on boot.

What is systemd?

Systemd is a service manager for operating systems. It is responsible for intializing the system,starting services , managing deamons and monitoring the system state.

Comparing Systemctl and Service: Understanding Linux Service Control

systemctl is part of the systemd system and service manager, which is increasingly becoming the standard init system on modern Linux distributions. It offers more advanced features and capabilities for managing services, including dependency management, control group management, and logging integration.

On the other hand, the service command is a legacy tool inherited from older init systems like SysVinit. It is still supported on many systems for compatibility but lacks some of the more advanced functionalities provided by systemctl.

You're all set! Docker and Jenkins are now installed on your CentOS system. With Docker, you can containerize applications, and with Jenkins, you can automate your development workflows. Enjoy the efficient development process ahead!