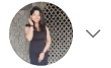


[Open in app](#)[Get unlimited access](#)

Sanjana thamke

Feb 17 · 4 min read · [Listen](#)

Save



# Day 7 Task: Understanding package manager and systemctl



## What is a package manager in Linux?

In simpler words, 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.

*package manager* keeps track of what software is installed on your computer, and allows you to easily install new software, upgrade software to newer versions, or remove software that you previously installed.

## What is a package?



A package is usually referred to an application but it could be a GUI application, command line tool or a software library (required by other software programs). A package is essentially an archive file containing the binary executable, configuration file and sometimes information about the dependencies.

## **Different kinds of package managers**

Package Managers differ based on packaging system but same packaging system may have more than one package manager.

### Types of Linux Package Managers

All Linux distributions have some form of a package manager. They all handle the same job, though:

- Installing applications
- Upgrading applications
- Managing application dependencies
- Removing applications
- Handling OS updates

Depending on the Linux distribution you are using, it will have a different Linux package manager. Here's a quick list of each package manager for popular distributions:

Linux Distribution	Linux Package Manager
REHL	Yum
Ubuntu / Debian	Apt
Arch	Pacman
Gentoo	Portage
Zyper	Open Suse

## Tasks

1. You have to install docker and jenkins in your system from your terminal using package managers
2. Write a small blog or article to install these tools using package managers on Ubuntu and CentOS

Now let's install the docker

1. Update the apt package index

```
sudo apt-get update
```

2. Install the docker

```
sudo apt install docker.io
```

3. Run Command to check docker is installed or not

```
ubuntu@ip-172-31-10-54:~$ docker -v
Docker version 20.10.12, build 20.10.12-0ubuntu4
```

## Now let's install the Jenkins

Reference :<https://www.trainwithshubham.com/blog/install-jenkins-on-aws>

### *Step — 1 Install Java*

#### Update your system

```
sudo apt update
```

#### Install java

```
sudo apt install openjdk-11-jre
```

#### Validate Installation

```
java -version
```

It should look something like this

```
openjdk version "11.0.12" 2021-07-20 OpenJDK Runtime Environment (build 11.0.12
```

### *Step — 2 Install Jenkins*

Just copy these commands and paste them onto your terminal.

```
curl -fsSL https://pkg.jenkins.io/debian/jenkins.io.key | sudo tee \ /usr/sha
```

```
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \ https://pkg.je
```

```
sudo apt-get update
```

```
sudo apt-get install jenkins
```

### Step -3 Start jenkins

```
sudo systemctl enable jenkins
```

```
sudo systemctl start jenkins
```

```
sudo systemctl status jenkins
```

```
sudo cat /var/lib/jenkins/secrets/initialAdminPassword
```

Step — 4 Open port 8080 from AWS Console using security group section

and using ip:8080 ,we can see jenkins page

### systemctl and systemd

systemctl is used to examine and control the state of “systemd” system and service manager. systemd is system and service manager for Unix like operating

systems(most of the distributions, not all).

## Tasks

1. check the status of docker service in your system (make sure you completed above tasks, else docker won't be installed)

```
ubuntu@ip-172-31-10-54:~$ service docker status
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2023-02-17 14:26:29 UTC; 33min ago
   TriggeredBy: ● docker.socket
     Docs: https://docs.docker.com
    Main PID: 2067 (dockerd)
      Tasks: 7
     Memory: 36.8M
        CPU: 430ms
    CGroup: /system.slice/docker.service
            └─2067 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
```

2. stop the service jenkins and post before and after screenshots

Before:

```
ubuntu@ip-172-31-10-54:~$ sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
   Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2023-02-17 15:06:41 UTC; 2min 18s ago
     Main PID: 5546 (java)
        Tasks: 36 (limit: 1143)
     Memory: 295.3M
        CPU: 43.992s
    CGroup: /system.slice/jenkins.service
            └─5546 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080
```

After:

```
ubuntu@ip-172-31-10-54:~$ sudo systemctl stop jenkins
ubuntu@ip-172-31-10-54:~$ sudo systemctl status jenkins
○ jenkins.service - Jenkins Continuous Integration Server
   Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor preset: enabled)
   Active: inactive (dead) since Fri 2023-02-17 15:09:50 UTC; 2s ago
     Process: 5546 ExecStart=/usr/bin/jenkins (code=exited, status=143)
    Main PID: 5546 (code=exited, status=143)
   Status: "Jenkins stopped"
        CPU: 44.155s
```

3. read about the commands systemctl vs service

eg. systemctl status docker VS service docker status

1. **systemctl** is a command-line utility used to control and manage the **systemd** system and service manager on Linux systems. It can be used to start, stop, and restart services, enable and disable them to start at boot, and check the status of services.

2.**service** is a more generic utility that is used to manage services on a wider range of systems, including those that do not use **systemd** . If your system uses **systemd**, you should use **systemctl** to manage services. If your system does not use **systemd** ,or if you are not sure, you can use **service** as a more generic utility.

### **systemctl commands :**

- `systemctl start docker`
- `systemctl stop docker`
- `systemctl restart docker`
- `systemctl enable docker`
- `systemctl disable docker`
- `systemctl status docker`

### **service commands :**

- `service docker start`
- `service docker status`
- `service docker stop`
- `service docker restart`

\_Thank you

\_Sanjana 🙌

DevOps

Systemctl

Package Management