Linux Fedora 44: Installation & Customize.

; 28.06.2026.17.25.wib [Indonesia]

Fedora 44 was officially released on Tuesday, April 28, 2026.

The release was originally planned for mid-April, but the development team pushed it back a couple of weeks to squash a few final blocker bugs—sticking to Fedora's classic "quality over schedule" philosophy. Since we just walked through the step-by-step installation flow and basic optimization steps (like setting up DNF5 and RPM Fusion), let's dive straight into advanced customization to turn your Fedora 44 workstation into a sleek, productive powerhouse.

[Part 1: Installation Steps]
Before you begin, grab a USB drive (at least 8GB) and download Fedora Media Writer along with the Fedora 44 Workstation ISO from the official site: URL: https://mirrormanager.fedoraproject.org/mirrors/Fedora.

1. Flash the ISO:
5 mins.
Open Fedora Media Writer, select the Fedora Workstation 44 image, choose your USB drive, and write it. If use Ubuntu 26.04LTS use balenaEtcher. Warning: This wipes the USB.

2. Boot into Live Media:
Restarting PC. Shut down your target PC. Boot into your BIOS/UEFI menu (usually by tapping F2, F12, or Del) and change the boot order to prioritize your USB drive. Save and exit.

3. Partition & Select Destination:
Critical step. Once the live desktop loads, click Install to Hard Drive. Select your language, then go to Installation Destination. For a clean install, select your main drive, leave storage configuration on Automatic, and click Done.

4. Create User & Begin:
10-15 mins.
Set up your local user account, make yourself an administrator, and set a strong password. Click Begin Installation. Once finished, click Finish Installation and manually restart your machine (remember to unplug the USB).

[Part 2: Customization & Optimization]

Once you boot into your shiny new Fedora 44 desktop, skip the initial tour and open up the terminal. Let's optimize it right away.

1. Supercharge DNF5
Fedora 44 uses DNF5, which is already way faster than older versions. However, you can still optimize it to download updates concurrently:
$ sudo vim /etc/dnf/dnf.conf

Add the following line under the [main] section, save (Ctrl+O, Enter), and exit (Esc+:, then wq, Enter):
max_parallel_downloads=10

2. Update and Enable Third-Party Repositories

Fedora is notoriously strict about shipping only free, open-source software. To get standard codecs, hardware acceleration, and restricted drivers, you need the RPM Fusion repositories. First, fetch the latest updates:
$ sudo dnf -y update

Next, run this single command to pull both the free and non-free RPM Fusion repositories:
$ sudo dnf install https://mirrors.rpmfusion.org/free/fedora/rpmfusion-free-release-$(rpm -E %fedora).noarch.rpm https://mirrors.rpmfusion.org/nonfree/fedora/rpmfusion-nonfree-release-$(rpm -E %fedora).noarch.rpm

3. Install Media Codecs
Without this, video playback on certain browsers or media players will break. Swap out the generic barebones tools for the full versions:
$ sudo dnf swap ffmpeg-free ffmpeg --allowerasing

$ sudo dnf groupinstall multimedia --allowerasing

Note: Make sure to reboot your computer after swapping media libraries so your desktop apps don't throw cryptic errors.

4. Graphics Drivers (NVIDIA Users Only)
If you are running an NVIDIA graphics card, do not download drivers from the NVIDIA website. Use the RPM Fusion package you just unlocked.

5. Desktop Polishing (GNOME Tweaks)
If you chose the default Workstation edition, GNOME can feel a bit bare out of the box. Let's add customization power:

Open Tweaks from your app grid. From here you can:
+ Enable Minimize and Maximize window buttons (under Windows Titlebars).
+ Change system fonts and default themes.
+ Manage GNOME extensions (install the Extension Manager flatpak from the Software Store to easily browse and install community extensions like Dash to Panel or Blur my Shell).

1.a. Complete Flatpak & Flathub Setup
While Fedora 44 enables a filtered version of Flathub out of the box, it restricts proprietary software. To unlock the full catalogue of Flatpaks (Spotify, Discord, Steam, VS Code, etc.), run:
$ sudo flatpak remote-add --if-not-exists flathub https://dl.flathub.org/repo/flathub.flatpakrepo

Restart GNOME Software or your terminal, and you will now have access to thousands of sandboxed, up-to-date applications.

1.b. Desktop Rise: Next-Level GNOME Customization

If you want to move away from the stock GNOME look and build a customized layout, you'll need the Extension Manager.
$ sudo dnf –y install extension-manager

Open **Extension Manager** from your applications menu. Click the **Browse** tab and search for these community-favorites:

Extension Name	What It Does	Why You Want It
Dash to Dock *or* **Dash to Panel**	Modifies the app dock.	Brings a macOS-style dock to your desktop, or combines the top bar and dock into a classic Windows-like single panel.
Blur My Shell	Adds dynamic aesthetic blur.	Applies a beautiful frosted-glass blur effect to the overview screen, menus, and panel.
Just Perfection	Desktop element toggles.	Allows you to hide the clock, remove the accessibility icon, resize the panel, or completely redesign UI layouts.
AppIndicator and KStatusNotifierItem	System tray icons.	Forces background apps (like Discord, Steam, or Dropbox) to show up as icons in your top bar.

1.c. Power Management & Laptop Optimization

Fedora 44 uses power-profiles-daemon by default. If you are on a laptop and want to maximize battery life, you can optimize your power states right from the terminal or your quick settings panel.

To check your current profile status:
$ sudo powerprofilesctl

If you find your battery draining too fast on an AMD or Intel laptop, consider replacing it with TLP (a deeper power management tool), or simply use the GNOME quick-settings toggle to switch to Power Saver when unplugged.

1.d. Enable System Auditing & Rollbacks (Btrfs)

Fedora uses the Btrfs file system by default. You can use this to take absolute control of system safety. If an update ever breaks a custom theme or driver, you can roll back the entire OS in seconds.

Install Timeshift or Snapper via the Software Center to configure automatic, zero-overhead background snapshots of your system before every single dnf upgrade.

[Part 3: Intermezzo]
Because you are on Fedora 44, using yum will still work seamlessly—but under the hood, it is actually redirecting to Fedora's next-generation package manager, DNF5.

Instructs the system to grab both networking packages in a single command.
$ sudo dnf install -y vim curl wget lynx

Enterprise Ship: LAMP: Linux, Apache, MariaDB, PHP
In Fedora 44, running dnf -y install httpd installs the Apache HTTP Server, one of the most widely used web servers in the world.

Here is how to complete the setup so your web server actually starts up, handles traffic through the firewall, and allows you to test it.

Step 1: Start and Enable the Apache Service

By default, Fedora installs services in a disabled and stopped state for security reasons. Run the following commands to start it and ensure it launches automatically every time your system boots:

$ sudo systemctl enable httpd
$ sudo systemctl status httpd

To verify that it is running perfectly, check its status:
$ sudo systemctl status httpd

Step 2: Configure the Firewall

Fedora ships with an active firewall (firewalld) that blocks incoming web traffic by default. You need to explicitly allow HTTP (port 80) and HTTPS (port 443) traffic:

```
# Allow web traffic through the firewall
sudo firewall-cmd --permanent --add-service=http
sudo firewall-cmd --permanent --add-service=https

# Reload the firewall to apply the new rules
sudo firewall-cmd --reload
```

Step 3: Test Your Installation

Your server is now live! You can verify it locally or from another machine on your network.
+ Locally: Open your web browser and type http://localhost or http://127.0.0.1 into the address bar.
+ Via Terminal: Use the curl tool you installed earlier:
$ lynx http://localhost

You will see the default Fedora Web Server Test Page, confirming that Apache is successfully hosting content.

Step 4: Add Your Own Content

The default root directory for your website's files is /var/www/html/. You can replace the test page by creating an index.html file there:

```
$ echo "<h1>Welcome to my Fedora 44 Web Server!</h1>" >
/var/www/html/index.html
```

Refresh http://localhost in your browser, and your custom message will appear.

Let's finish the backend by installing and hardening MariaDB (which officially updates to the blazing fast 11.8 series in Fedora 44). Once this is done, it will fully link up with the PHP libraries you set up earlier.

Step 1: Install MariaDB Server

We need the server package (mariadb-server), which will also automatically pull in the command-line client utilities and proper SELinux security policies:
$ sudo dnf install -y mariadb-server

Step 2: Start and Enable the Service

Like the web server, the database engine installs in an inactive state. Fire it up and set it to auto-start when the system boots:

# Start the service immediately
sudo systemctl start mariadb

# Enable it to run on system startup
sudo systemctl enable mariadb

Step 3: Run the Install Hardening Wizard

Fresh database installations are vulnerable by default because they contain blank passwords and test access keys. Run the native security binary to clean it up:
$ sudo mariadb-secure-installation

You will step through a series of prompts. Here are the recommended answers:
+ Enter current password for root: Press Enter (it's blank out of the box).
+ Switch to unix_socket authentication: Type Y. This allows you to securely access the database via sudo mariadb from your local machine without needing a hardcoded root password string.
+ Change the root password? Type n (since you enabled unix_socket, you won't need one locally).
+ Remove anonymous users? Type Y.
+ Disallow root login remotely? Type Y.
+ Remove test database and access to it? Type Y.

+ Reload privilege tables now? Type Y.

Step 4: Access Your Database & Create a User

Now you can log directly into the database monitor natively via root privileges:

$ sudo mariadb

Once inside the MariaDB shell prompt (MariaDB [(none)]>), let's create a dedicated database and an application user for your future PHP projects. Run these SQL commands one by one:

```
-- 1. Create a database
CREATE DATABASE app_db;

-- 2. Create a specific user with a strong password
CREATE USER 'app_user'@'localhost' IDENTIFIED BY
'PutAStrongPasswordHere';

-- 3. Grant the user full permissions over the new database
GRANT ALL PRIVILEGES ON app_db.* TO 'app_user'@'localhost';
-- 4. Clear memory cache and exit

FLUSH PRIVILEGES;

EXIT;
```

$ sudo vim /var/www/html/db_test.php

Paste the following code, substituting your database password if you changed it:

```
<?php

$mysqli = new mysqli("localhost", "app_user",
"PutAStrongPasswordHere", "app_db");


if ($mysqli->connect_error) {

    die("× Connection failed: " . $mysqli->connect_error);
```

}

echo "Success! PHP 8.4 is talking to MariaDB 11.8 perfectly.";

?>

Save and exit (Esc+:, wq, then Enter), and give Apache a quick reload:

$ sudo systemctl reload httpd

Open up your browser and navigate to http://localhost/db_test.php. If you see the Success! message, your backend database stack is completely customized and ready to deploy your applications.

The command hw-probe -all -upload triggers a tool called Hardware Probe. It takes a secure snapshot of your computer's hardware configuration, checks component health against diagnostic system logs, and uploads the results to the public Linux Hardware Database.

This is incredibly useful if you are trying to diagnose a hardware issue, check device compatibility with the Linux kernel, or find a missing hardware driver.

Step 1: Install the Utility (if you haven't yet)

On Fedora 44, hw-probe is available directly from the official repositories. Install it via DNF:
$ sudo dnf -y install hw-probe

Step 2: Run the Command Correctly

Because the tool needs to examine low-level hardware structures, hard drives (for SMART health status), and the kernel log (dmesg), you must run it with administrative privileges.

The officially recommended syntax includes the -E flag to preserve your environment variables during the elevation:
$ sudo -E hw-probe -all -upload

What Happens Next?

The terminal will output a short checklist:
Probe for hardware ... Ok
Reading logs ... Ok
Uploaded to DB, Thank you!
Probe URL: https://linux-hardware.org/?probe=xxxxxxxxxx

🔒 Privacy Note: The tool sanitizes your logs before uploading. Sensitive and identifiable data—like username strings, MAC addresses, and serial numbers—are either completely stripped or securely hashed locally on your computer before they hit the server.

[Part 4: Linux Hardening Security]

Linux security hardening is the process of configuring and securing an operating system to minimize its attack surface. By default, most Linux distributions are optimized for user convenience rather than high security.

Here is a tactical guide to hardening your Linux system:
TCP SYN Cookie Protection is a defense mechanism built directly into the Linux kernel to protect servers from SYN Flood Denial of Service (DoS) attacks.

If you are running a public web or database server on Fedora 44, understanding and enabling this feature is critical for maintaining uptime.

The Problem: How a SYN Flood Works

In a standard TCP connection, a client and server perform a 3-way handshake:

+ Client sends a SYN (Synchronize) packet.
+ Server allocates memory, places the connection in a "half-open" queue, and replies with a SYN-ACK.

Client replies with an ACK to complete the connection.

During an attack, a malicious actor sends thousands of spoofed SYN packets but never replies with the final ACK. The server's half-open connection queue completely fills up, causing it to drop connection requests from legitimate users.

The Solution: SYN Cookies

When SYN Cookie Protection is enabled, if the half-open queue fills up, the server stops creating a state in memory for incoming connections entirely. Instead, it crafts a special cryptographic cookie inside the sequence number of the SYN-ACK reply.

If the client is legitimate, it will reply with an ACK containing that cookie. The server validates the mathematical sequence, confirms it's real, and opens the connection on the spot. Result: The attack fails to exhaust your server's RAM.

How to Check and Enable it on Fedora 44Fedora 44 typically has this enabled by default, but you should verify it—especially if you are configuring a custom server build.

1. Check current status

Run this command to see if it is enabled on your running kernel:
$ sudo sysctl net.ipv4.tcp_syncookies

+ net.ipv4.tcp_syncookies = 1 means Enabled (Protected).
+ net.ipv4.tcp_syncookies = 0 means Disabled.

2. Enable it instantly (Temporary)

If it returns 0, you can turn it on immediately without rebooting the machine:
$ sudo sysctl -w net.ipv4.tcp_syncookies=1
$ sudo sysctl --system

3. Make it Permanent

To ensure this protection survives system reboots and updates, you must drop it into a sysctl configuration file.

Create a dedicated network hardening file:
$ sudo vim /etc/sysctl.d/99-sysctl-hardened.conf

Add the following line to the file:
net.ipv4.tcp_syncookies = 1

Save and close the file (Ctrl+O, Enter, then Ctrl+X). Apply the new configurations seamlessly:

Deep Dive: What These Settings Do Together

net.ipv4.tcp_max_syn_backlog = 4096

+ The Job: Expands the maximum number of "half-open" (SYN-RCVD) connections the kernel will track simultaneously in its buffer.

+ The Impact: The default value on many Linux systems is quite low (often 128 or 256). Raising this to 4096 ensures your Apache or MariaDB stack can confidently absorb sudden, intense bursts of legitimate concurrent connection requests without prematurely dropping users.

net.core.somaxconn = 1024

+ The Job: Sets the absolute maximum queue limit for completely established sockets waiting to be accepted by your backend applications (like Apache/httpd).

+ The Impact: Your tcp_max_syn_backlog handles the incoming handshake, but once the handshake completes, the connection moves to this queue. Pushing this to 1024 acts as a crucial safety cushion, protecting your backend stack from choking under heavy application loads.

net.ipv4.tcp_synack_retries = 3

+ The Job: Dictates how many times the kernel will re-transmit a SYN-ACK response packet to an unresponsive client before abandoning the connection attempt.

+ The Impact: The system default is usually 5 retries, which forces the kernel to wait up to 180 seconds before cleaning up a dead connection. Dropping this to 3 forces a timeout much sooner (around 15–20 seconds), clearing out malicious or dead connections from your backlog quickly and freeing up system RAM.

How to Apply These Right Now

To write these permanently into your system configurations so they persist through updates and system reboots:

1. Open your custom security profile:
$ sudo vim /etc/sysctl.d/99-sysctl-hardened.conf

2. Paste your block of text cleanly inside:
net.ipv4.tcp_syncookies = 1
net.ipv4.tcp_max_syn_backlog = 4096
net.core.somaxconn = 1024
net.ipv4.tcp_synack_retries = 3

3. Save and close (Esc+:, wq, then Enter).

4. Force the Linux kernel to live-load your modifications immediately without requiring a system reboot:
$ sudo sysctl --system

Last finishing of Enterprise Ship: LAMP: Linux, Apache, MariaDB, PHP

Install all available software and system updates on Fedora 44.
$ sudo dnf -y upgrade

Pro Tip: Clean House After Upgrading

Once your system finishes updating, it's a great habit to clean up old, orphaned packages (like dependencies left behind by apps you deleted) and see if any system-critical files require a reboot.

You can do it easily with these two commands:
1. Safely remove unneeded background package leftovers

$ sudo dnf –y autoremove

2. Check if a full system reboot is recommended (e.g., for a new Linux kernel)

$ sudo needs-restarting -r

If the needs-restarting tool tells you a reboot is required, just type reboot into your terminal to safely restart into your freshly patched system environment!

Enjoy your Fedora Workstation.

by: piasDraco [/pias : pias25]