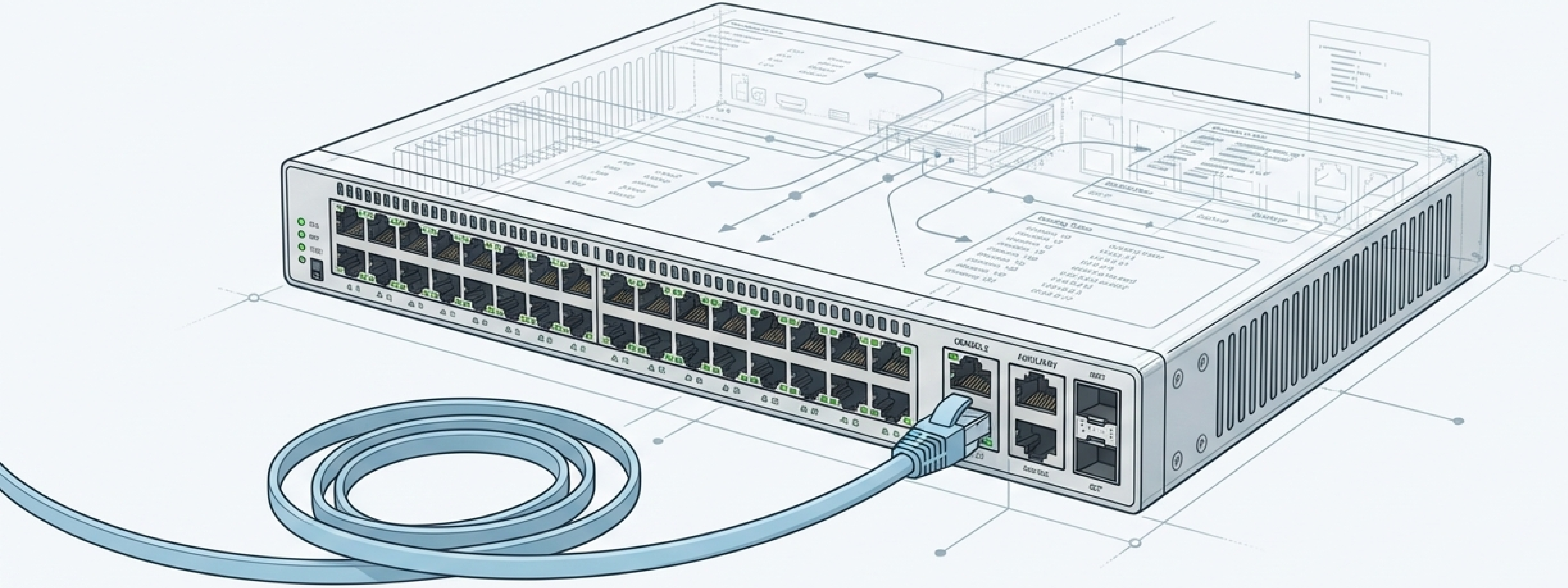
From Box to Baseline

Mastering the Initial Router Configuration



The essential workflow every network professional must know.

Your Mission: Forge a Trustworthy Device

Your task is to take an unconfigured, unknown, or partially configured router and transform it into a secure, manageable, and trustworthy network device.

Take Control

Gain local administrative access.

Establish Identity

Give the device a unique name and configuration.

Lock Down Access

Secure all entry points (console, remote).

Enable Connectivity

Configure interfaces to pass traffic.

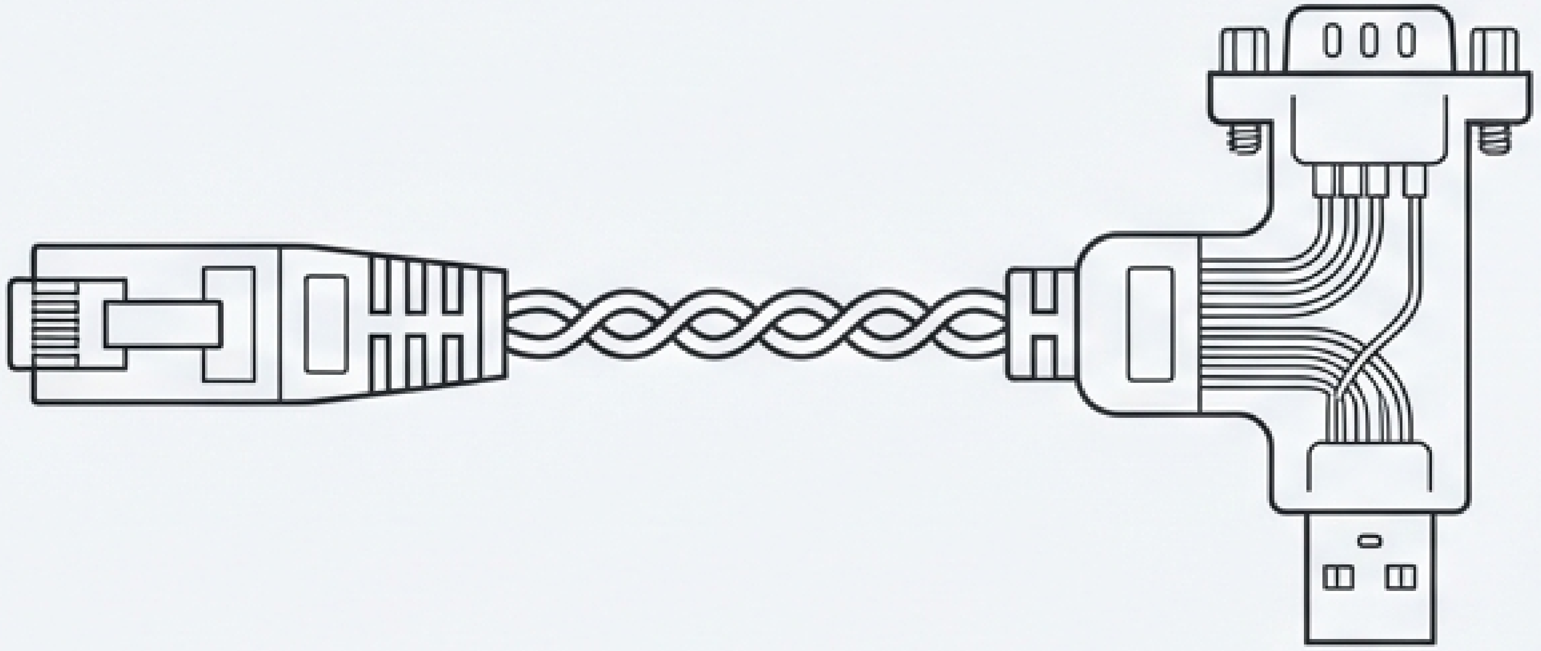
Ensure Persistence

Save your work so it survives a reboot.

The First Step: Making the Physical Connection

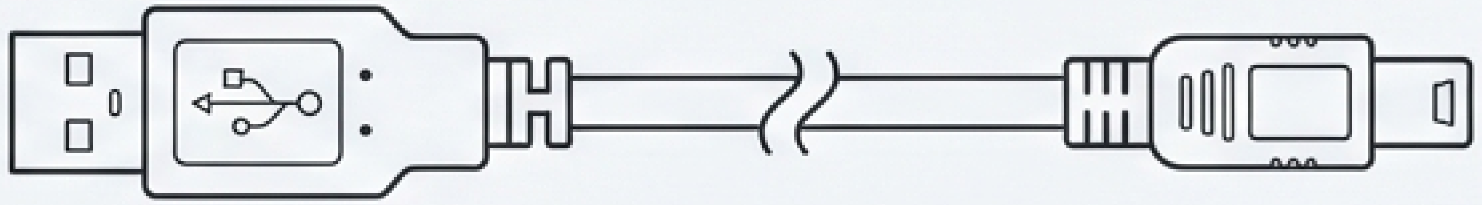
The Classic RJ-45

Uses a "rollover" console cable. Your PC may need a serial-to-USB adapter with the correct driver if it lacks a native DB-9 serial port.



The Modern USB

Uses a standard USB cable (e.g., Type-A to mini-B). Requires a specific Cisco USB console driver for your OS (Windows, macOS, or Linux).



Pro-Tip: Console Port Priority

If a router has both port types, only one is active. Plugging in the USB cable deactivates the RJ-45 port. Unplugging the USB cable reactivates it. This is a common reason for console access suddenly failing.

Speaking the Right Language: Terminal Settings

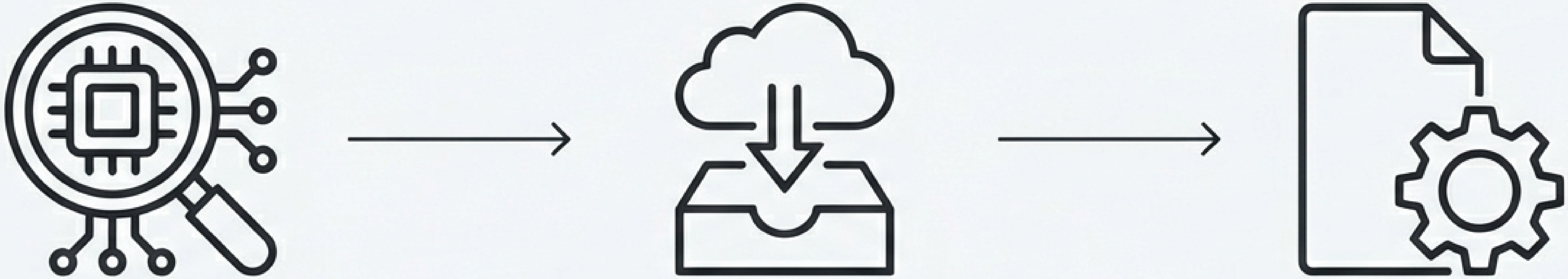
To communicate with the router, you must configure your terminal emulator (like PuTTY or Tera Term) with these exact settings. Memorize them.

Setting	Value
Speed (Baud Rate)	**9600**
Data Bits	**8**
Parity	**None**
Stop Bits	**1**
Flow Control	**None**

Getting Junk Characters or a Blank Screen?

- Is the correct COM/TTY port selected?
- Is the driver installed (especially for USB-to-Serial or USB Console)?
- Are the baud rate and flow control settings *exactly* as listed above?

The Awakening: Inside the Router Boot Process



POST

The **Power-On Self-Test** runs diagnostics on the CPU, memory, and physical interfaces.

IOS Load

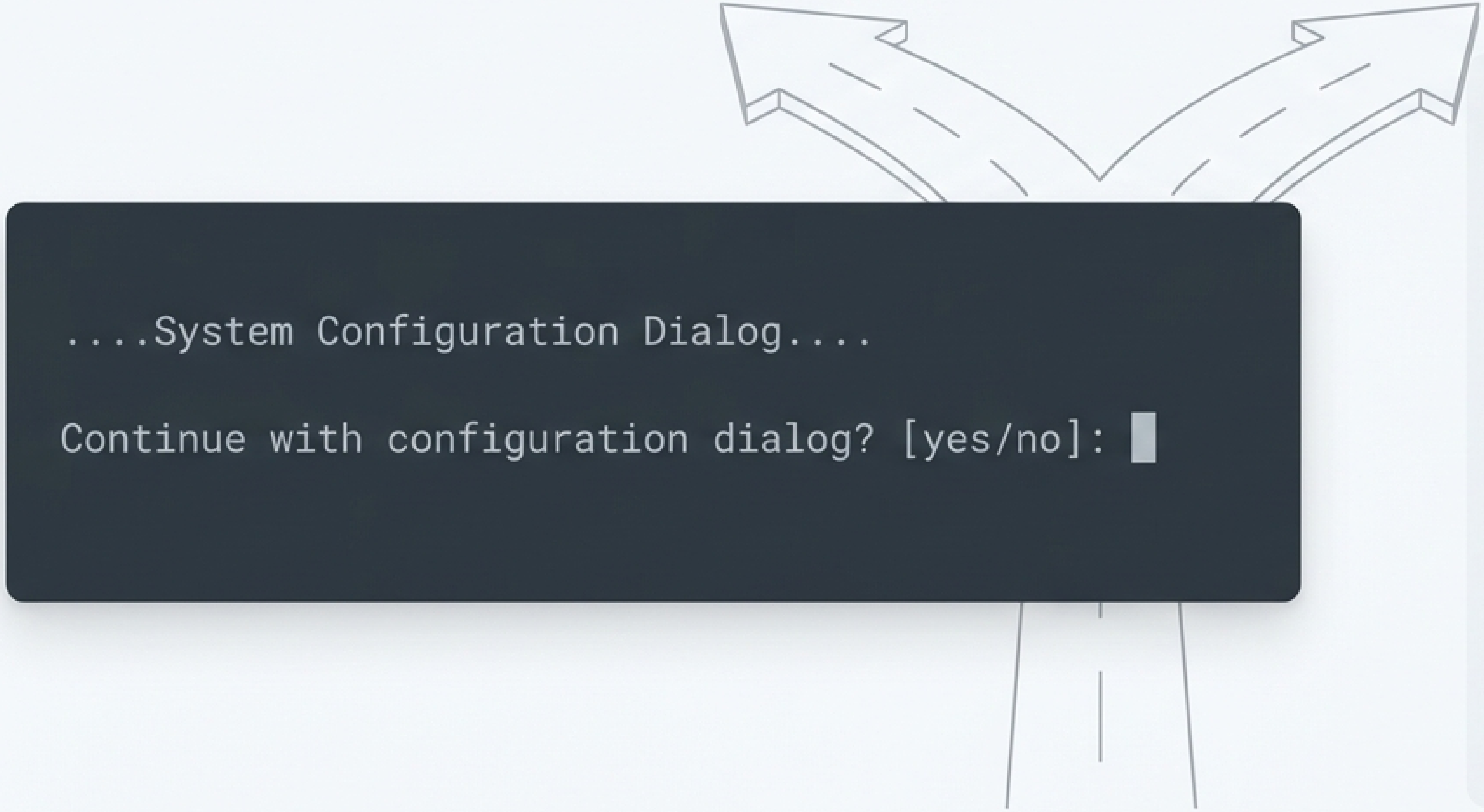
The router finds and loads the Cisco **Internetwork Operating System (IOS)** image, which is the device’s core software.

Config Load

The router looks in NVRAM for the **startup-config** file. If found, it loads this configuration into memory.

The Crossroads: The System Configuration Dialog

After the IOS loads, if no `startup-config` file is found in NVRAM, the router will present you with this prompt:



```
....System Configuration Dialog....

Continue with configuration dialog? [yes/no]: █
```

CCNA Quick Facts

- **What it is**
  A prompted, wizard-like guide for a minimal configuration.
- **What it isn't**
  A tool for complex or professional configurations.
- **The Professional's Choice**
  Always skip it and use the command-line interface (CLI) for full control and precision.

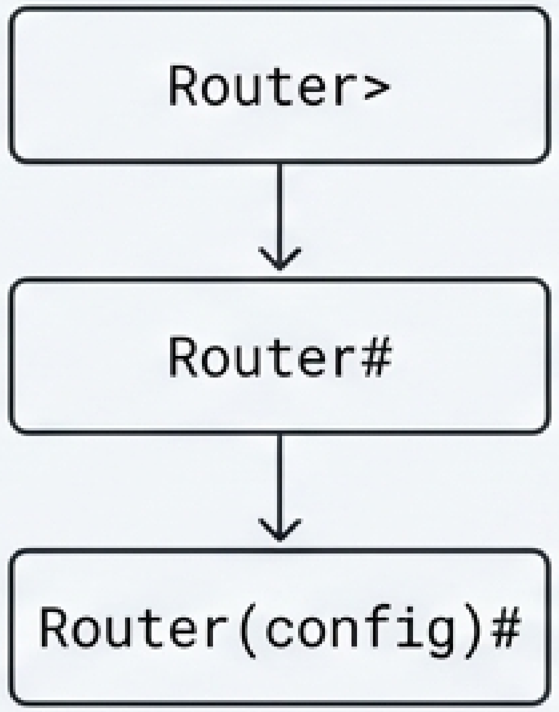
Taking Manual Control: Skipping Setup and Entering the CLI

Two Ways to Skip:

- Simply answer `no` to the prompt.
- Press `Ctrl-C` at any time to abort the dialog.

The Path Forward: Navigating CLI Modes

Once you skip setup, you will enter the CLI. The typical workflow involves moving through these modes:



User EXEC: (Limited view-only commands)

Privileged EXEC: (Full monitoring & access to other modes)

Global Configuration: (Device-wide settings)

Core Command Flow

```
Router> enable
Router# configure terminal
Router(config)#
```

The Blueprint: Goals of a Solid Baseline Configuration

A secure baseline isn’t just a random list of commands; it's a structured approach to achieve key operational goals.

1. Establish Identity

Give the router a unique `hostname`. This is critical for identification in a network.

2. Secure Administrative Access

Protect all modes.

- Privileged EXEC (`enable secret`)
- Console Port (`line console 0`)
- Remote VTY Lines (`line vty 0 4`)

3. Enable Remote Management

Configure SSH for secure remote access, which requires a `domain-name` and `crypto keys`.

4. Activate Connectivity

Assign an `ip address` to an interface and, crucially, enable it with `no shutdown`.

5. Ensure Persistence

Use `copy running-config startup-config` to save the active configuration so it survives a reboot.

The Build, Part 1: Identity and Console Security

Step 1 — Name It

```
Router(config)# hostname R1
R1(config)#
```

Step 2 — Secure Admin Access

```
R1(config)# enable secret Str0ngEnableSecret
```

Step 3 — Secure Console Access

```
R1(config)# line console 0
R1(config-line)# password C0nsoleP@ss
R1(config-line)# login
R1(config-line)# exec-timeout 10 0
R1(config-line)# logging synchronous
R1(config-line)# exit
```

Sets an idle timeout of 10 minutes, 0 seconds.

Prevents console messages from interrupting your typing.

The Build, Part 2: Remote Access and Password Security

Step 4 — Secure Remote Access (SSH Best Practice)

SSH requires a hostname, domain name, and crypto keys before it can be enabled.

```
R1(config)# ip domain-name lab.local
R1(config)# crypto key generate rsa modulus 2048
R1(config)# username admin privilege 15 secret Adm1nP@ss
R1(config)# line vty 0 4
R1(config-line)# login local
R1(config-line)# transport input ssh
```

Tells VTY lines to use the local username database for authentication.

Allows only SSH connections, disabling insecure Telnet.

Step 5 — Hide Weak Passwords

```
R1(config)# service password-encryption
```

Applies a weak encryption to all plaintext passwords (like the console password) in the configuration output.

The Build, Part 3: Bringing the Interface to Life

Step 6 — Set Interface IP and Enable

An interface requires both an IP address and to be administratively enabled before it can pass traffic.

```
R1(config)# interface GigabitEthernet0/0
R1(config-if)# description LAN Connection to SW1
R1(config-if)# ip address 192.168.1.1 255.255.255.0
R1(config-if)# no shutdown
```

The Most Important Command

The `no shutdown` command administratively enables the interface. Without it, the interface will remain down even with a valid valid IP address and a connected cable.

The Final Step: Save Your Work

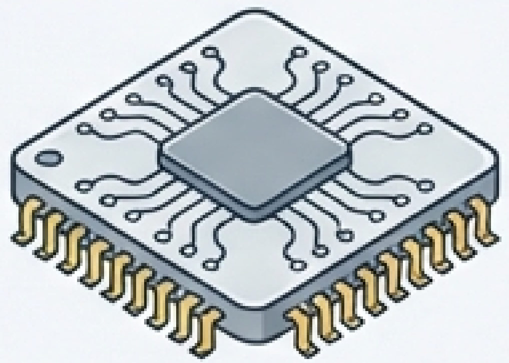
```
R1# copy running-config startup-config
Destination filename [startup-config]? [Enter]
Building configuration...
[OK]
```

The Logic Explained

running-config

- Lives in RAM.
- Is the active, live configuration.
- Is volatile and will be lost on reboot.

startup-config



- Lives in NVRAM (Non-Volatile RAM).
- Is the backup configuration.
- Is loaded into RAM every time the router boots.

You MUST copy the `running-config` to the `startup-config` to make your changes permanent.

The After-Action Report: Key Verification Commands

Never assume the configuration is correct. Always verify with these commands from Privileged EXEC mode (R1#).

```
R1# show running-config
```

Displays the entire active configuration currently in RAM. Use this to check every command you just entered.

```
R1# show ip interface brief
```

The fastest way to check interface IP addresses, and their line status ('up') and protocol status ('up').

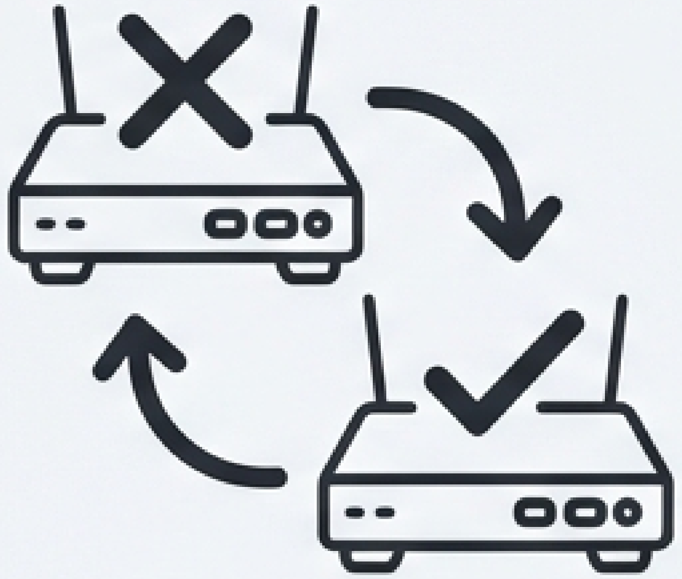
```
R1# show version
```

Confirms the device platform, IOS version, uptime, and the location of the IOS image file.

```
R1# show startup-config
```

The final check. Confirms your running configuration was successfully saved to NVRAM. If this is empty or different from `show run`, your changes will be lost on reboot.

Where This Matters: The Real-World Payoff



Emergency Device Swap

When a router fails, you must be able to configure its replacement from scratch, quickly and accurately, to restore service.

Fixing Lost Remote Access

If a bad configuration change breaks SSH/Telnet, the only way to fix it is with a physical console connection using these exact skills.

Troubleshooting a 'Blank' Router

If a colleague forgets to save the configuration (`copy run start`), the router will boot up 'blank' and enter the setup dialog after a power outage. You'll know immediately that the `startup-config` is missing.

Key Takeaway: This isn't just an academic exercise; it's the core discipline that prevents and resolves network outages.

The Technician’s Field Guide: Initial Config on a Single Page

Console Connection

App
PuTTY, Tera Term, etc.

Settings
9600 / 8-N-1 / No Flow Control

Boot Process
POST -> IOS Load -> Config Load

Skip Setup
Answer `no` or press `Ctrl-C`

Baseline Configuration Workflow

```
enable
configure terminal
hostname R1
enable secret <secret>
line console 0 -> password <pass> -> login
ip domain-name <domain>
crypto key generate rsa modulus 2048
username <name> privilege 15 secret <pass>
line vty 0 4 -> login local -> transport input ssh
interface <type/num>
 ip address <ip> <mask>
 no shutdown
end
```

Save & Verify

Save Command

```
copy running-config startup-config
```

Verification

```
show ip interface brief
show running-config
show startup-config
show version
```