SkillNewton

Home Lab Setup Checklist

Plan and build a safe networking and security home lab: hardware, hypervisors, isolated networks, VMs, tools and snapshots.

This checklist guides you through building a home lab for practising networking, Linux, Windows Server and security skills. It is for students and early-career engineers who want hands-on practice without risking a real network. It covers goals, hardware, hypervisors, network design, the VMs to build, tools, safety and documentation. Follow the sections in order, keep vulnerable machines isolated, and take snapshots so you can break things and recover quickly.

How to use it

1. **Set your goals.** Decide what you want to practise, because that sets how much RAM, storage and how many VMs you need.
2. **Choose hardware and a hypervisor.** Start with the computer you already have and a free hypervisor before buying anything.
3. **Design the network before building VMs.** Plan isolated and NAT networks and IP ranges first so vulnerable machines never touch your home network.
4. **Build, snapshot and document.** Build one VM at a time, take a clean snapshot, and record settings and passwords in your lab notes.

Goals and planning

- [ ] Write down what you want to learn
  For example routing and firewalls, Linux administration, Active Directory or penetration testing basics.
- [ ] Choose two or three starter projects
  Such as a firewall with two internal networks, a Linux web server and a domain controller.
- [ ] Estimate resources per VM
  A minimal Linux server can run in 1–2 GB of RAM; Windows Server with a desktop needs more. Check each product's stated requirements.
- [ ] Decide on a budget
  You can start for free with an existing PC and free or evaluation software.
- [ ] Plan your IP ranges
  Use private ranges that do not clash with your home network, for example `10.10.10.0/24` if your home uses `192.168.1.0/24`.

Hardware options

- [ ] Old desktop or laptop
  Often the cheapest start. Check the CPU supports virtualisation (Intel VT-x or AMD-V) and enable it in the BIOS or UEFI.
- [ ] Mini PC
  Small, quiet and low power, and many accept extra RAM and an NVMe SSD. Good for an always-on Proxmox host.

- [ ] Raspberry Pi
  Good for Linux, DNS filtering and network services, but it uses an ARM processor, so it cannot run x86 VMs such as Windows Server or most firewall images.
- [ ] Prioritise RAM and SSD storage
  Running several VMs at once is limited mostly by memory; an SSD makes VMs far more responsive.
- [ ] Optional: managed switch and spare NIC
  A small managed switch that supports VLANs (802.1Q) lets you practise real VLANs and trunking.

Hypervisor options

- [ ] Oracle VirtualBox
  Free and runs on Windows, Linux and macOS (Intel and Apple silicon builds). Easy for beginners.
- [ ] VMware Workstation Pro
  Free for personal use on Windows and Linux; Workstation Player has been discontinued. Check Broadcom's current licence terms.
- [ ] Microsoft Hyper-V
  Built into Windows 10 and 11 Pro, Enterprise and Education. Enabling it can affect other hypervisors on the same PC.
- [ ] Proxmox VE
  Free, open-source bare-metal platform for a dedicated lab machine, managed through a web interface, with VMs and containers.
- [ ] Pick one and learn it well
  Type 2 hypervisors (VirtualBox, Workstation) run on your desktop OS; Proxmox is installed directly on the hardware.

Network design

- [ ] Create an isolated internal or host-only network for lab traffic
  VirtualBox "Internal Network", VMware "LAN segment" or a Proxmox bridge with no physical port keeps traffic inside the lab.
- [ ] Use NAT only for VMs that need internet access for updates
  NAT lets VMs reach the internet without being reachable from your home network.
- [ ] Avoid bridged mode for vulnerable or untrusted VMs
  Bridged VMs sit directly on your home network alongside family devices.
- [ ] Put a firewall VM between lab networks
  Route lab segments through pfSense or OPNsense so you control and log traffic between them.
- [ ] Use VLANs if you have a managed switch
  Separate lab, management and home traffic with tagged VLANs and test inter-VLAN routing on the firewall.
- [ ] Draw a diagram of every network and address
  Include VM names, interfaces, subnets and gateways.

VMs to build

- [ ] pfSense CE or OPNsense firewall
  Give it a WAN interface on NAT and one or more LAN interfaces on internal networks. Practise DHCP, firewall rules and VPNs.
- [ ] Ubuntu Server LTS
  Practise the command line, SSH, users, services, a web server and a database.
- [ ] Windows Server evaluation
  Microsoft offers free time-limited evaluation editions. Practise Active Directory, DNS, DHCP and Group Policy.

- [ ] A Windows or Linux client VM
  Join it to your domain and test policies, file shares and troubleshooting.
- [ ] A deliberately vulnerable target, only on an isolated network
  Metasploitable 2 or DVWA (Damn Vulnerable Web Application). Never connect them to your home network or the internet.
- [ ] An attacker or tools VM
  For example Kali Linux, placed on the same isolated network as the target.

Tools

- [ ] Wireshark
  Capture traffic on lab interfaces to see ARP, DHCP, DNS, TCP handshakes and HTTP in detail.
- [ ] Nmap
  Discover hosts and open ports, for example `nmap -sV 10.10.10.0/24`, only on lab networks.
- [ ] Built-in command-line tools
  `ping`, `traceroute/tracert`, `ip`, `ss`, `dig` and `nslookup`.
- [ ] Cisco Packet Tracer for routing and switching
  A free simulator for practising Cisco IOS commands alongside your virtual lab.
- [ ] A text editor and Git for configs and notes
  Store configuration exports and scripts in a private repository.

Safety and legality

- [ ] Only scan or attack systems you own or have written permission to test
  Unauthorised access to computer systems is an offence under India's IT Act, 2000 and similar laws elsewhere.
- [ ] Keep vulnerable VMs on isolated networks only
  Check their network adapter setting before every start-up.
- [ ] Never expose lab services to the internet with port forwarding
  Use a VPN on the firewall if you need remote access.
- [ ] Do not run scans on your ISP, office, college or public Wi-Fi networks
  Even a simple Nmap scan can breach acceptable-use policies.
- [ ] Keep the host machine patched and backed up
  The host holds every VM, so a compromise or disk failure loses the lot.

Snapshots, backups and documentation

- [ ] Take a snapshot after each clean install
  Name it clearly, for example "ubuntu-base-updated", so you can roll back after experiments.
- [ ] Snapshot before risky changes
  Revert instead of rebuilding when something breaks.
- [ ] Remember snapshots are not backups
  Export important VMs or use Proxmox backups to another disk.
- [ ] Keep a lab notebook
  Record what you built, commands used, problems and fixes. It doubles as portfolio material for interviews.
- [ ] Store lab passwords in a password manager
  Use unique passwords even in the lab, and never reuse real ones.

- [ ] Track evaluation licence expiry dates

  Note when evaluation editions expire so you can rebuild or re-arm in time.

Online version, with the matching free guide: skillnewton.com/resources/templates/home-lab-setup-guide/

Learn this hands-on: Networking & Cybersecurity: Foundation & Practical Implementation – skillnewton.com/courses/networking-foundation-course/