

# Everything You Use Today Runs on Linux

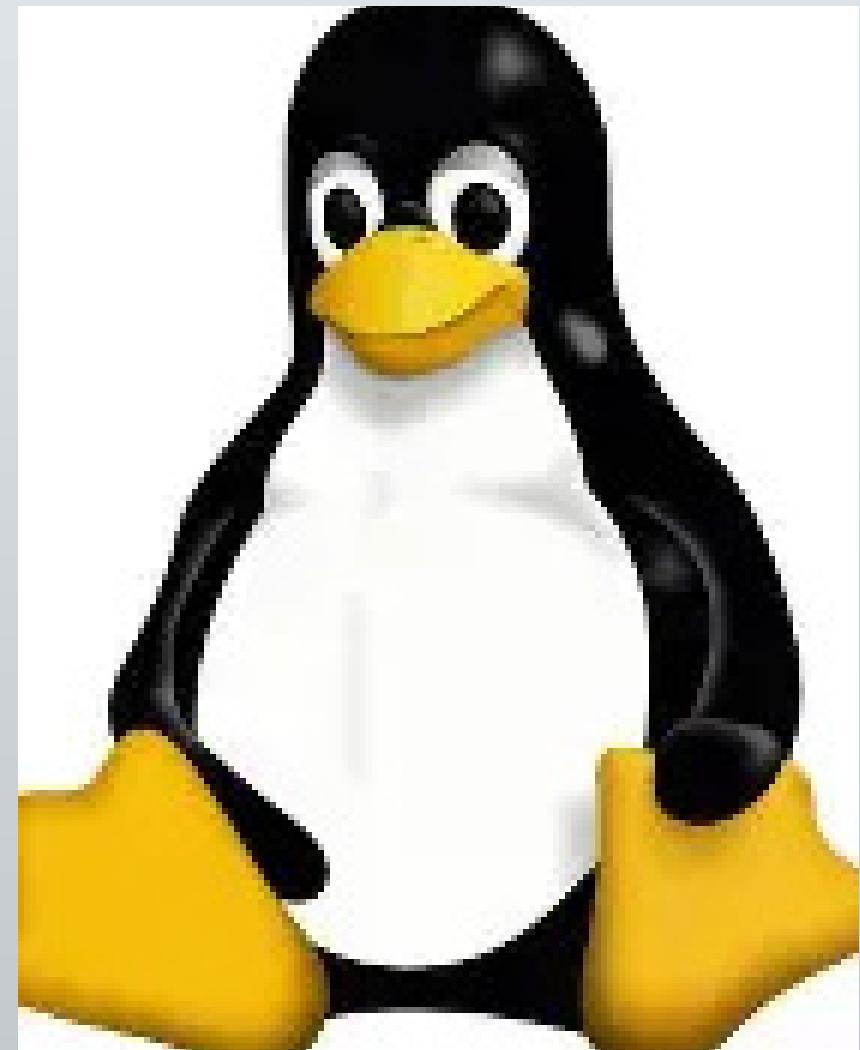
Docker? Linux.

Kubernetes? Linux.

AWS? Linux.

Google Cloud? Linux.

Azure? Linux.



Your "serverless" functions? Linux containers.

ChatGPT? Linux.

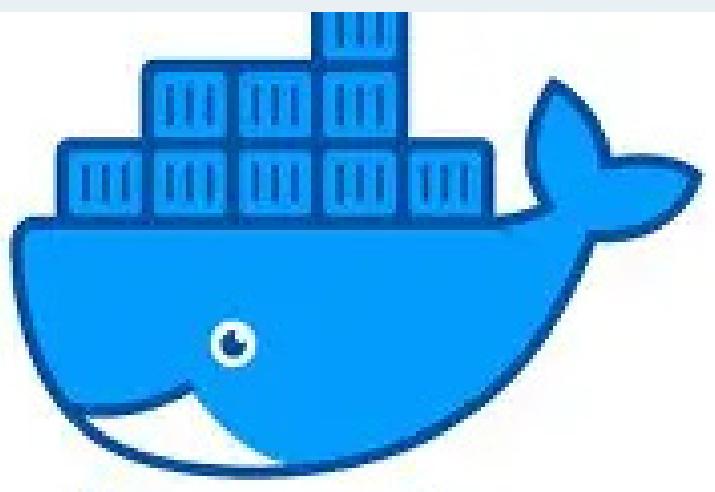
Netflix? Linux.

Your Tesla? Linux.

**95%** of the internet runs on Linux.

And you don't know Linux?

Here's what you're missing: 



**docker**

"I Don't Need Linux, I  
Use Docker"  
**Docker IS Linux.**

Every container = Linux namespaces + **cgroups** + **chroot**  
When you run: `docker run **ubuntu**`

You're running a Linux kernel with isolated processes.

Docker on Windows? **WSL2** (Windows Subsystem for Linux)

Docker on Mac? Linux VM running underneath

**There is no "Docker without Linux."**

If you use Docker and don't know Linux:  
You're **driving a car without knowing it has an engine.**



# kubernetes

"I Know Kubernetes,  
That's Enough"

Kubernetes **IS** a Linux orchestrator.

Every **pod** = Linux container

Every **node** = Linux VM

Every network **policy** = Linux **iptables**

Every **volume** mount = Linux filesystem

Kubernetes doesn't **REPLACE** Linux knowledge.

It **REQUIRES** it.

When your pod crashes:

- Check Linux logs (**journalctl, dmesg**)
- Debug Linux networking (**netstat, ss, iptables**)
- Fix Linux resource limits (**cgroups, ulimit**)

"I know **K8s** but not Linux" =

"I know how to schedule meetings but not how  
calendars work"



# "I Use Cloud, I Don't Need Linux" Part 1

**The cloud IS Linux.**

**AWS EC2** = Linux VMs

**AWS Lambda** = Linux containers

**Google Cloud Run** = Linux containers

**Azure Functions** = Linux containers

**"Serverless"** = Linux (you just don't manage it)

**Every major cloud provider:**

→ Runs on Linux infrastructure

→ Offers Linux VMs as default

→ Uses Linux for ALL container services



# "I Use Cloud, Don't Need Linux"

## Part 2

**Cloud abstracts Linux.**

It doesn't **ELIMINATE** it.

When your **Lambda** fails:

- Container crashed (Linux)
- Out of memory (Linux **OOM killer**)
- Network timeout (Linux networking)

**You can't debug what you don't understand.**



# The Production AI Stack (All Linux)

## Part1

### Training Infrastructure:

- **NVIDIA** drivers (Linux kernel modules)
- **CUDA** toolkit (Linux libraries)
- **PyTorch/TensorFlow** (compiled for Linux)
- Distributed training (Linux networking: **RDMA**, **InfiniBand**)

### Data Pipelines:

- **Airflow** (runs on Linux)
- **Spark** (JVM on Linux)
- **Kafka** (Linux processes)

### Container Orchestration:

- Docker (Linux containers)
- Kubernetes (Linux orchestrator)
- containerized (Linux runtime)



# The Production AI Stack (All Linux)

## Part 2

### Monitoring:

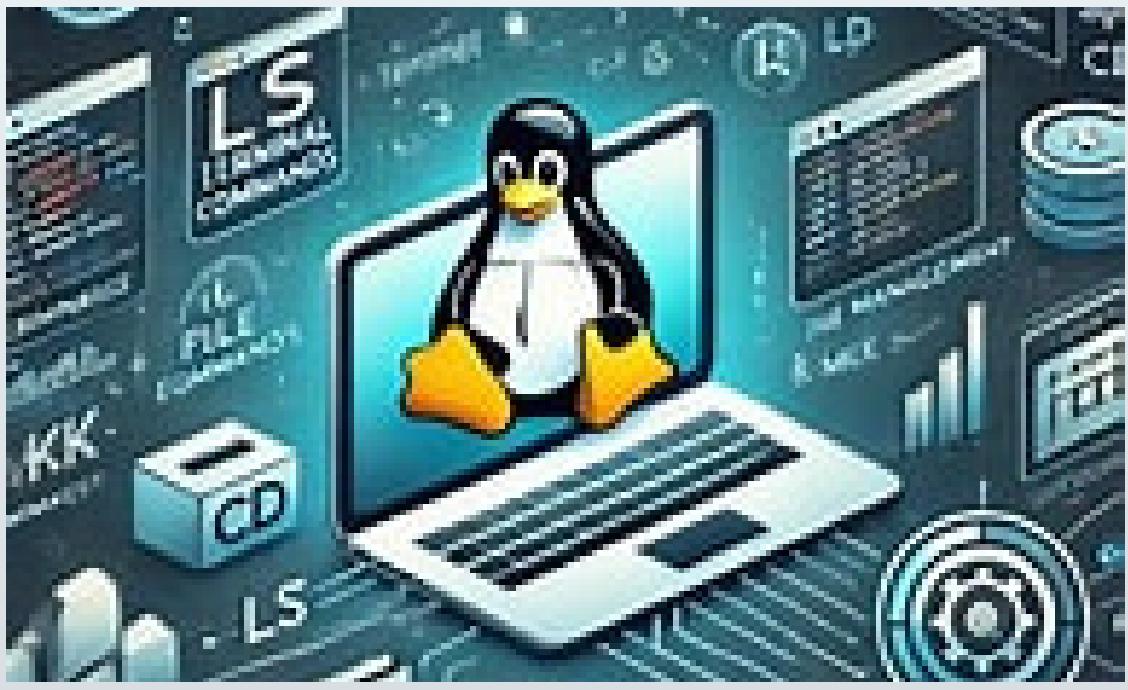
- **Prometheus** (Linux metrics)
- **Grafana** (Linux dashboards)
- **ELK stack** (Linux log aggregation)

### CI/CD:

- **Jenkins** (Linux agents)
- **GitHub Actions** (Linux runners)
- **GitLab CI** (Linux executors)

**Every layer = Linux.**

"I don't need Linux" = "I don't do production"



# 🔧 Linux Skills That Matter (Not Theory, Production) Part 1

## Shell Scripting:

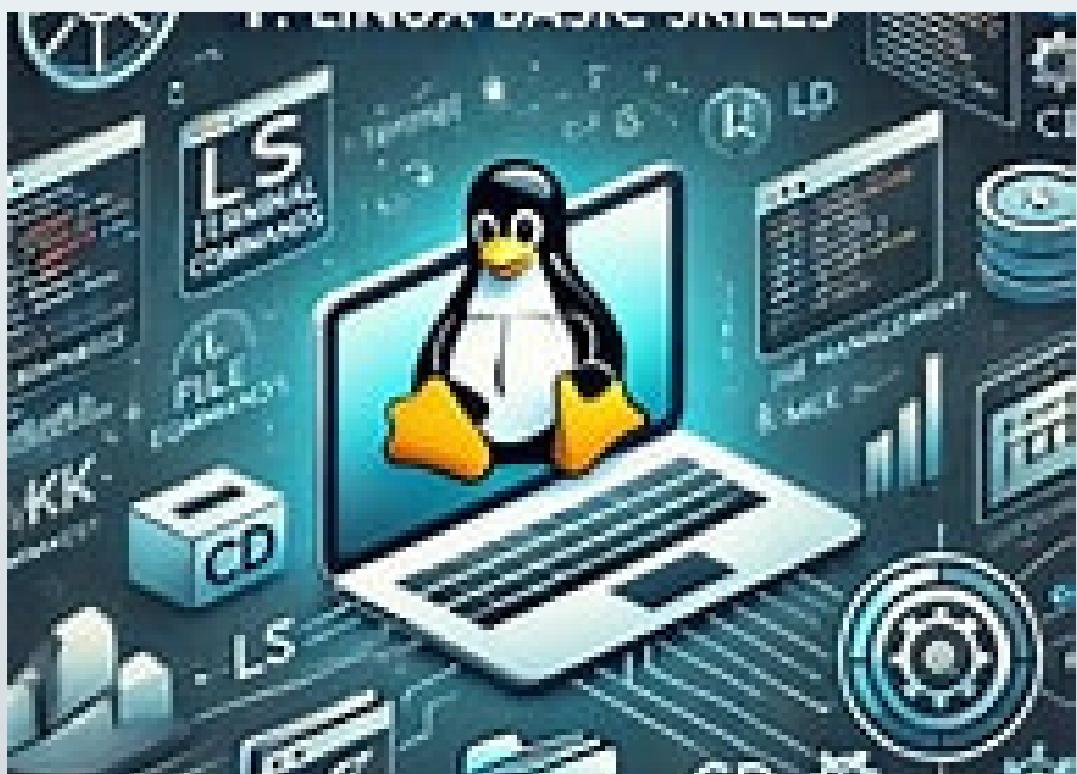
- Automate everything (deployments, monitoring, cleanup)
- Bash, not just copy/paste from Stack Overflow

## Process Management:

- **systemd** (start/stop services)
- **ps**, **top**, **htop** (what's running, what's hung)
- **kill**, **pkill** (stop runaway processes)

## Networking:

- **netstat**, **ss** (what's listening, what's connected)
- **iptables** (firewall rules)
- **DNS** debugging (**dig**, **nslookup**)



## 🔧 Linux Skills That Matter (Not Theory, Production) Part 2

### File Systems:

- **mount, umount** (attach storage)
- **df, du** (disk space)
- **chmod, chown** (permissions)

### Package Management:

- **apt, yum** (install software)
- Building from source (when packages don't exist)

### Kernel Basics:

- **dmesg** (kernel messages)
- **journalctl** (system logs)
- **lsmod, modprobe** (kernel modules for GPUs)

These aren't "nice to know."

They're "production breaks without them."



# ⚠ What Developers Get Wrong

✗ "Linux is just for servers"

✓ Android = Linux. Your phone runs Linux.

✗ "I use Mac, I don't need Linux"

✓ macOS development ≠ production. Production = Linux.

✗ "Docker abstracts Linux away"

✓ Docker IS Linux. You can't abstract the foundation.

✗ "I'll learn Linux when I need it"

✓ Production breaks at 3am. That's when you need it.

✗ "ChatGPT can write my bash scripts"

✓ ChatGPT can't debug why your GPU driver failed.

✗ "Cloud makes Linux knowledge obsolete"

✓ Cloud REQUIRES Linux knowledge when things break.

# The Uncomfortable Truth

You can learn frameworks in boot camps.

You can learn Python from tutorials.

You CANNOT learn production Linux from YouTube.

It takes years.

In terminals.

Fixing broken systems at 2 am.

**95%** of production infrastructure = Linux

**100%** of container platforms = Linux

Every cloud provider = Linux underneath

If you're building production systems and don't know Linux:

You're not a production engineer.

You're a **configuration manager hoping things work**.

Linux isn't optional. It's **foundational**.

# T-Shaped Expertise

**Technology changes every 6 months**, but the problems we're solving remain constant—**continuous learning ensures you can use NEW tools on OLD problems.**

**T-shaped expertise** (deep in MLOps, broad in security/networking/infrastructure) means that when one domain shifts, you **pivot** using knowledge from others.

**The best time to start was yesterday.**

**The next best time is today.**

**What's your Linux war story? ↴**

#Linux #Production #Engineering #SystemsThinking #Security  
#ContinuousLearning #Docker #MLOPS #Pivot #Networking