Below is a **realistic Junior SRE Competency Checklist** based on what hiring managers actually probe for (not what job posts exaggerate).

Use it as a **gap detector**, not a perfection list.

🎯 Junior SRE Competency Checklist

🟢 Tier 1 — Must-Have (Hireability Core)

If these are weak, interviews usually fail.

1️⃣ Linux Operational Competence

You should comfortably be able to:

- navigate filesystem quickly
- analyze CPU, memory, disk usage
- inspect running processes
- read system logs
- manage services
- understand basic networking

Command fluency expected:

- top, htop
- ps, kill
- df, du, free
- journalctl
- systemctl
- ss or netstat
- ip a, ip route
- curl, wget
- basic bash piping (|, grep, awk basics)

✅ **Hire signal:** You debug methodically, not randomly.

2️⃣ Kubernetes Day-1 Operations

For junior roles, they are NOT expecting deep internals.

You should be comfortable with:

- Pods, Deployments, Services
- ConfigMaps & Secrets
- resource requests/limits
- liveness & readiness probes
- rolling updates
- basic troubleshooting with kubectl

You must be able to debug:

- CrashLoopBackOff
- ImagePullBackOff
- pending pods
- failing probes

✅ **Hire signal:** You use kubectl describe and logs first.

3️⃣ Monitoring & Alerting Fundamentals

You already started strong here.

Expected competence:

- what metrics vs logs vs traces are
- Prometheus scraping concept
- Grafana dashboard basics
- Node Exporter meaning
- alert rule basics
- Alertmanager purpose

You should explain clearly:

"How would you detect high CPU on a node?"

✅ **Hire signal:** You think in symptoms → signals → alerts.

4️⃣ Basic Networking Understanding

Not CCIE level — just solid fundamentals.

You should understand:

- DNS basics
- HTTP vs HTTPS
- TCP vs UDP
- ports and services
- load balancer concept
- what happens when you curl a service

Debug comfort:

- connection refused
- timeout
- DNS failure

✅ **Hire signal:** You troubleshoot from network outward.

5️⃣ Incident Thinking Mindset

This is where many juniors fail.

You should naturally think in:

- detection
- mitigation
- root cause
- prevention

Be ready to answer:

"A service is down. What do you do first?"

Good answer pattern:

1. Confirm impact
2. Check recent changes
3. Look at metrics/logs
4. Mitigate quickly

5. Investigate root cause

✅ **Hire signal:** Calm, structured thinking.

🟡 **Tier 2 — Strong Differentiators (Not Required, But Powerful)**

These move you from "maybe hire" → "strong junior candidate."

6️⃣ **Containers & Docker Comfort**

Expected level:

- build simple Dockerfile
- understand image vs container
- basic troubleshooting
- push/pull from registry

7️⃣ **Basic Scripting Ability**

You do NOT need to be a software engineer.

But you should:

- write small bash scripts
- automate simple tasks
- parse logs with shell tools

Even simple scripts impress at junior level.

8️⃣ **Logging Pipeline Awareness**

Since you already touched:

- Fluent Bit
- Graylog

You are ahead of many juniors.

Just ensure you can explain clearly:

"How logs move from app to dashboard."

🔵 Tier 3 — Nice to Have (Ignore for Now)

Do NOT delay applications for these.

- Terraform
- advanced cloud architecture
- service mesh
- advanced CI/CD
- distributed tracing deep dive

These matter later, not for first job.

🚨 Fast Self-Assessment (Be Honest)

Rate yourself 1–5 on each Tier 1 area:

- Linux
- Kubernetes ops
- Monitoring
- Networking
- Incident thinking

Interpretation:

- Mostly 4–5 → start applying aggressively
- Mostly 3 → apply + keep improving (this is normal)
- Any 1–2 → patch that area this month

🎯 What Hiring Managers Secretly Look For

At junior level, the winning signal is:

"This person will not panic in production."

Your hydro control room background actually strengthens this narrative — use it.