# InfraThrone Elite – Week 1 Troubleshooting Guide

---

If you follow this process, you can troubleshoot almost any Linux-based infrastructure issue without panic.

---

### 1. OS Internals & Boot Process

**When a system won't boot**

1. **Establish State & Access**
    - Console access?
    - Remote (SSH) vs Rescue mode
    - Uptime, last boot logs (last -x | head)
2. **Identify Boot Stage Failure**
    - BIOS/UEFI → Kernel → initramfs → init/systemd
    - If stuck in initramfs → check /etc/fstab mounts
    - If kernel panic → review last lines of dmesg
3. **Recovery Actions**
    - Boot into single-user or rescue mode
    - Mount the root FS manually and chroot into it
    - Regenerate initramfs & GRUB configs:
      dracut --regenerate-all --force,  grub2-mkconfig -o /boot/grub2/grub.cfg

4. **Advanced**
    - Detect hidden rootkits with chkrootkit / rkhunter
    - Analyse crash dump (/var/crash, kdump) for cause

---

# 2. Linux Networking & Traffic Control

### When connectivity is broken

1. **Layered OSI Debug**
    - **L1:** Check cables, virtual NIC states (ip link show)
    - **L2:** Verify MAC & ARP tables (arp -n)
    - **L3:** IP config (ip addr), routes (ip route)
    - **L4:** Port state (ss -lntup)
    - **L7:** Service logs & DNS resolution
2. **Packet Tracing**
    - tcpdump -i eth0 host <IP> – verify packets leaving/arriving
    - If service works on IP but not hostname → DNS (dig, /etc/resolv.conf)
3. **Traffic Shaping & Bottlenecks**
    - Check tc qdisc show for throttling rules
    - Detect drops with ethtool -S eth0 | grep drop
4. **Advanced**
    - Namespace-level debugging with ip netns exec
    - Reverse-engineer custom routing/firewall setups



---

# 3. Secure Access & User Management

### When unauthorized access is suspected

1. **Identify Active Sessions**
    - w, who, last for session tracking
    - Inspect ~/.bash_history (check for timestamp mismatch)
2. **Check Authentication Logs**
    - /var/log/secure or /var/log/auth.log
    - PAM logs for abnormal module calls
3. **Harden Access**
    - Lock user (usermod -L user) without deleting home
    - Rotate keys & check /etc/ssh/sshd_config for PermitRootLogin
4. **Advanced**
    - Audit binary integrity with rpm -Va or debsums
    - Use auditd to track file modifications in /etc/ and /root/



---

# 4. Cloud-Native Fundamentals (Containers & Kubernetes Basics)

### When container workloads fail

1. **Container-Level**
    - Logs: docker logs <container>
    - Inspect events: docker inspect <container>
    - Check health checks & resource limits
2. **Host-Level**
    - Verify namespaces: lsns
    - Check storage mounts, overlayfs layers
3. **Networking**
    - Verify bridge: docker network inspect <network>
    - Packet trace inside container: nsenter --target <PID> --net
4. **Advanced**
    - Detect zombie containers impacting others
    - Use strace to capture syscalls on a failing container process



---

# 5. Terraform + Ansible

## When IaC deployments fail

1. **Terraform**
    - terraform plan → compare with desired
    - Check remote state file for drift/corruption
    - Recover partial applies with terraform state rm or taint
2. **Ansible**
    - Run with -vvv for verbose logs
    - Use --step to run playbook interactively
    - Ensure idempotency by re-running and verifying no changes
3. **Integration Issues**
    - Ensure API credentials are valid and scoped
    - Validate modules/providers versions match your environment
4. **Advanced**
    - Rollback infrastructure manually if pipeline is stuck mid-deploy
    - Write a one-off playbook for hotfix without breaking IaC state



---

## 6. Elite-Level “War Room” Checklist

When **anything breaks in prod**, run this **battle flow**:

1. **Stop the bleeding**
    - Mitigate impact before fixing root cause (scale up, route traffic, revert change)

2. **Frame the incident**
    - What changed? Who deployed? Which component? When?

3. **Reproduce or isolate**
    - Can it be recreated in staging?

4. **Trace from outside in**
    - External checks → load balancer → app → OS → hardware

5. **Document EVERYTHING**
    - Commands run, outputs, and timestamps

6. **Escalate with data**
    - Logs, metrics, and your working hypothesis

7. **Postmortem**
    - RCA with preventive steps, automation, and alerts



---

**Saurav Chaudhary**

**Senior DevOps Consultant • Infra Architect • Mentor • War Room Specialist**