On-Call Engineer Checklist: High Server Load Incident

Alert Source: Datadog - "High CPU/Memory Utilization on [HOSTNAME]"

PHASE 1: INITIAL DIAGNOSIS (First 5 Minutes)

[ ] Acknowledge the alert in PagerDuty. [ ] SSH into the affected server: `ssh [HOSTNAME]` [ ] Check current load and uptime: `uptime` [ ] Check memory and swap usage: `free -h` [ ] Check disk I/O wait times: `iostat -xz 1 5` [ ] Identify top resource-consuming processes: `top` or `htop`

PHASE 2: AUTOMATED RESPONSE

[ ] **Is the issue with a known web application (e.g., Loyalty Microservice)?** ---> YES: Navigate to `/opt/it-automation/scaling/` and trigger the application restart script. `./restart_service.sh -s loyalty-app` ---> NO: Proceed to manual investigation.

[ ] **Is the server part of an auto-scaling group (e.g., Web Servers)?** ---> YES: Check the AWS/Azure console to confirm if a new instance is already being provisioned. If not, manually trigger the scaling script. `./scale_out_cluster.sh -c web-prod-cluster` ---> NO: This is a standalone server (e.g., Database). Do NOT run scaling scripts.

PHASE 3: MANUAL INTERVENTION & ESCALATION

[ ] If a specific process is identified as hung or looping, attempt to kill it gracefully (`kill [PID]`) and then forcefully (`kill -9 [PID]`). [ ] Check application logs in `/var/log/[appname]/` for recent ERROR or FATAL entries. [ ] Check system logs (`journalctl -f`) for hardware or kernel-level errors. [ ] **If load does not subside within 15 minutes:** ---> Escalate to the Tier-2 application development or database team. ---> Update the incident ticket with all diagnostic information gathered.