PRODUCTION SRE HANDBOOK

Kubernetes Real World Problems: 12 Production Postmortems

Battle-Tested Incident Triage, Diagnostic Playbooks, Linux Kernel Failure Modes, and Permanent Architectural Solutions for Enterprise Clusters

Naveed Ahmed (Kumbhar)
Lead Cloud Platform Architect & Kubernetes Specialist
Official Publication · naveedkumbhar.com

Table of Contents

1. **Executive Architecture Summary & Systemic Risk**
2. **CoreDNS 5-Second Latency Spikes (glibc ndots & conntrack race)**
3. **Worker Node OOMKilled Cascades (cgroup v2 & memory limit traps)**
4. **AWS VPC CNI Subnet IP Exhaustion (ENI allocation & Karpenter)**
5. **Mutating Webhook Timeout Deadlocks (Control plane failure modes)**
6. **Ingress-NGINX 502 Bad Gateway Outages (Endpoint propagation race)**
7. **StatefulSet EBS Multi-Attach Volume Deadlocks**
8. **CFS CPU Bandwidth Throttling Microsecond Latency Anomalies**
9. **etcd Fsync Disk Bottlenecks on Shared Cloud Storage**
10. **Silent Cluster-Wide TLS Certificate Expiration**
11. **Calico / Cilium Default-Deny NetworkPolicy Lockouts**
12. **Docker Hub Container Registry Anonymous Rate Limiting**
13. **Ephemeral Storage Disk Exhaustion via Unlinked Zombie Descriptors**
14. **Production Incident Triage Cheatsheet & Rapid Diagnostic Commands**

1. Executive Architecture Summary

Kubernetes abstraction layers hide distributed system complexity until production workloads expose edge cases in the Linux kernel, cloud control planes, and network overlays. In mission-critical environments, outages rarely originate from bad YAML manifests; they stem from race conditions in asynchronous subsystems.

> **Key Principle:** Container orchestration requires defense-in-depth: NodeLocal DNSCache for connection tracking, graceful termination hooks for reverse proxies, dedicated NVMe volumes for etcd write-ahead logs, and strict resource quotas with topology spread constraints.

2. Postmortem 1: CoreDNS 5-Second Latency Spike

Root Cause: Linux Netfilter connection tracking table race condition between concurrent A and AAAA DNS queries over UDP when `ndots:5` causes glibc to append 5 search domains.

```
# Check conntrack insertion drops
conntrack -S | grep insert_failed

# Trace CoreDNS UDP requests from inside container
tcpdump -i eth0 udp port 53 -nn
```

Permanent Remediation:

- Deploy NodeLocal DNSCache as a DaemonSet using link-local IP `169.254.20.10` to handle queries locally over TCP.
- Configure pod `dnsConfig` with `options: [{name: "ndots", value: "2"}, {name: "single-request-reopen"}]`.

3. Postmortem 2: Cascading Worker Node OOMKilled Storm

Root Cause: Memory limits set equal to memory requests without kernel swap or system headroom reserves, leading Linux cgroup OOM Killer to terminate pods and trigger scheduler re-placement cascades.

```
# Inspect container exit status (Exit Code 137 indicates OOMKilled)
kubectl get pod <pod-name> -o jsonpath='{.status.containerStatuses[0].lastState.terminated}'

# Verify Kubelet node allocatable reservations
kubectl describe node <node-name> | grep -A 8 "Allocatable"
```

Permanent Remediation: Configure `system-reserved` and `kube-reserved` on Kubelet, implement Vertical Pod Autoscaler in recommendation mode, and enforce soft PodDisruptionBudgets.

4. Production Troubleshooting Diagnostic Cheatsheet

Failure Mode	Initial Symptom	Primary Diagnostic Command	Root Cause Solution
CoreDNS Race	5000ms outbound API latency	`conntrack -S | grep insert_failed`	NodeLocal DNSCache DaemonSet + ndots:2
Node OOM Storm	Exit Code 137, node flapping	`kubectl get pods -A -o jsonpath='{...}'`	Kubelet system-reserved & cgroup v2 tuning
IP Exhaustion	Pods stuck in Pending	`kubectl describe pod | grep "FailedCreatePodSandBox"`	WARM_IP_TARGET tuning + secondary CIDR
Webhook Lock	API server rejects deployments	`kubectl get mutatingwebhookconfigurations`	failurePolicy: Ignore on critical namespaces
502 Gateway Race	Spikes during rolling deploy	`kubectl logs -n ingress-nginx ...`	preStop sleep hook + graceful endpointdrain
Volume Deadlock	Multi-Attach error for volume	`kubectl describe pvc <pvc-name>`	volumeAttributesClass / EBS CSI volume driver

5. Architectural Recommendations for Enterprise Kubernetes

1. **GitOps Drift Elimination:** Enforce automated declarative state with ArgoCD or Flux to eliminate configuration divergence between clusters.
2. **Observability Baseline:** Export Prometheus metrics for `container_cpu_cfs_throttled_periods_total` and `etcd_disk_wal_fsync_duration_seconds` to catch saturation before user degradation.
3. **Full Scenario Handbook:** Explore interactive scenarios, practice runbooks, and deep-dive technical tutorials at blog.naveedkumbhar.com.