LogZilla AI Copilot

AppOps Analysis Report

Generated: December 6, 2025 17:37:00 EST | Analysis Period: Last 2 hours vs. 26-24 hours ago

QUERY PROMPT

"Analyze all application events from the last 2 hours compared to the same 2-hour window yesterday. Correlate cache failures, container crashes, web server errors, and database timeouts to identify cascading failures. Map service dependencies and provide remediation commands."

Executive Summary

SEVERITY: CRITICAL

MAJOR INFRASTRUCTURE INCIDENT DETECTED - Cascading failure affecting multiple layers of the application stack with NO comparable events in the baseline period, indicating a significant new incident.

Comparative Analysis Summary

Metric	Current (2h)	Baseline (2h)	Status
Total Application Events	100+	0	⚠ Critical
Cache Miss Events	726	0	⚠ Critical
Cache Error Events	256	0	⚠ Critical
Container Crashes	100+	0	⚠ Critical
Web Server Errors	19,256	0	⚠ Critical
Network Device Crashes	50+	0	⚠ Critical

Root Cause Analysis

Primary Root Cause: Cascading Infrastructure Failure

Timeline of Cascade

```
T+0 (16:10:50) - INITIAL HARDWARE FAILURES
└─> Chassis intrusion on app-srv-01
└─> Memory allocation failures on nexus-core-01
└─> Power supply failures on sensor-2

T+27s (16:11:17) - NETWORK INFRASTRUCTURE DEGRADATION
└─> ASA firewall failover communications lost
└─> Multiple Nexus switches experiencing service crashes
└─> Memory allocation failures spreading across network devices

T+1m (16:12:00) - APPLICATION LAYER IMPACT
└─> Nginx upstream connection failures
└─> SSL handshake failures
└─> Cache systems reporting high miss rates (80-99%)

T+2m (16:13:00) - FULL CASCADE EFFECT
└─> Container health check failures
└─> Database table crashes
└─> etcd compaction delays (147s-4982s)
└─> Kubernetes DNS resolution failures
```

Service Dependency Mapping

Dependency Chain Analysis

```
Hardware Failures (Power/Memory/Temp)
    ↓
Network Infrastructure (Switches/Firewalls)
    ↓
Cache Layer (Redis/Memcached/DNS)
    ↓
Container Orchestration (K8s/Docker)
    ↓
Application Services (Nginx/API/Workers)
    ↓
End User Impact
```

Most Affected Hosts

Host	Error Count	Primary Issues
nginx-server	19,256	Upstream timeouts, SSL failures, connection resets
cisco-ftd	15,428	SQL injection attempts, malware detection
snort-sensor	10,265	Security event processing
fortigate	8,643	DDoS attack mitigation
nexus-core-01	12+	Service crashes, memory failures, power issues

Cache Layer Analysis

Cache Miss Rates

Service	Miss Rate	Status
Redis (redis-01, redis-02)	80-98%	Critical
Memcached	81-99%	Critical
Varnish	81-98%	Critical
Kubernetes CoreDNS	16-347%	Critical

Performance Degradation

Service	Issue	Response Time	Status
etcd	Compaction delays	147s - 4982s	Critical
nginx	Upstream timeouts	0.3s - 4.88s	Critical
notification-svc	/metrics endpoint	502 error	Critical
Application timeouts	Various services	2.8s - 23.8s	Critical

Remediation Commands

IMMEDIATE ACTIONS (Next 15 minutes)

1. Stabilize Hardware Layer

```
# Check and address temperature issues
ssh esxi-01.company.com
esxcli hardware ipmi sdr list | grep -i temp

# Verify power supply status
ssh nexus-core-01
show environment power

# Check memory status
ssh app-srv-01.company.com
ipmitool sdr list | grep -i dimm
```

2. Restore Network Infrastructure

```
# Check ASA failover status
ssh asa-firewall
show failover
failover reload-standby

# Restart crashed Nexus services
ssh nexus-core-01
show system internal sysmgr service crashed
system reset
```

3. Rebuild Cache Layer

```
# Restart Redis instances
docker restart lz_redis
ssh redis-01.company.com
redis-cli FLUSHALL
systemctl restart redis

# Clear Memcached
ssh memcache-01.company.com
echo "flush_all" | nc localhost 11211
systemctl restart memcached
```

```
# Restart CoreDNS in Kubernetes
kubectl rollout restart deployment/coredns -n kube-system
```

SHORT-TERM ACTIONS (Next 1 hour)

4. Stabilize Container Orchestration

```
# Check Kubernetes node health
kubectl get nodes
kubectl describe node <failing-nodes>

# Restart failed pods
kubectl get pods --all-namespaces | grep -E 'Error|CrashLoopBackOff'
kubectl delete pod <pod-name> -n <namespace>

# Check etcd health
kubectl exec -it etcd-0 -n kube-system -- etcdctl endpoint health
kubectl exec -it etcd-0 -n kube-system -- etcdctl defrag
```

5. Restore Application Services

```
# Check nginx upstream health
ssh nginx-server
nginx -t
systemctl reload nginx

# Verify backend service availability
curl -I http://backend-service:8080/health
```

Validation Checklist

After remediation, verify the following:

- **Hardware:** All temperature, power, and memory alerts cleared
- **Network:** No failover events, all services running
- **Cache:** Miss rates < 10%, all instances responding
- **Containers:** No crash loops, all health checks passing
- **Applications:** Response times < 1s, no timeout errors
- **Database:** No crashed tables, replication lag < 1s
- **Security:** DDoS attacks mitigated, no active intrusions

> **Execute Priority 1 remediation steps immediately to prevent further cascade.**

Report Generated: 2025-12-06 17:37:00 EST

Analysis Period: Last 2 hours vs. 26-24 hours ago

Analysis Confidence: High (based on comprehensive event data)