Incident Post-Mortem: CloudForge API Outage

Incident ID: INC-2024-0715 **Date:** July 15, 2024 **Duration:** 47 minutes (10:23 AM - 11:10 AM PT) **Severity:** P1 (Critical) **Author:** SRE Team **Status:** Complete

---

Executive Summary

On July 15, 2024, CloudForge experienced a 47-minute API outage affecting approximately 60% of customers. The root cause was a database connection pool exhaustion triggered by a configuration change during routine maintenance. The incident was detected within 3 minutes and fully resolved within 47 minutes. No data was lost.

Impact

Customer Impact

Metric	Value
Affected Customers	~60% (1,347 organizations)
Failed API Requests	847,000
Error Rate (peak)	78%
Affected Products	CloudForge API, Web UI, CLI

Business Impact

- SLA breach for 23 Enterprise customers
- Estimated revenue impact: ~$15,000 (credits issued)
- Customer support tickets: 156
- Social media mentions: 34

What Worked

- Provisioning jobs in progress were not affected (async processing)
- Data integrity maintained throughout incident
- Webhooks delivered (with delays) after recovery

Timeline (All times PT)

Time	Event
10:15	Maintenance window begins for DB connection pool tuning
10:20	Configuration change deployed (reduced max connections from 500 to 200)
10:23	Error rates begin increasing
10:26	PagerDuty alert fires (API error rate > 5%)
10:28	On-call engineer acknowledges alert
10:32	Incident declared, war room opened
10:35	Initial assessment: DB connection errors identified
10:42	Root cause identified: connection pool exhaustion
10:47	Rollback initiated for connection pool config
10:52	Rollback complete, connections recovering
10:58	Error rates returning to normal
11:02	API fully operational
11:10	Incident resolved, monitoring continues
11:30	Customer communication sent
14:00	Post-mortem meeting scheduled

Root Cause Analysis

What Happened

During routine maintenance, the database connection pool maximum was reduced from 500 to 200 connections to test more conservative settings in preparation for a planned database migration.

The change was deployed without adequate load testing. At 10:23 AM, normal traffic patterns exceeded the reduced connection limit. New API requests began queueing, and after the connection timeout (30 seconds), requests started failing.

Why It Happened

```
Maintenance scheduled
during business hours


Connection pool reduced
without load testing


Normal traffic exceeds
new connection limit


Requests queue & fail


Cascading failures
across API endpoints
```

Contributing Factors

1. **No staging test:** Change tested in dev environment (10% of prod traffic) only
2. **Insufficient change review:** Change classified as "low risk" without validation
3. **Poor timing:** Deployed during peak usage hours (10 AM PT)
4. **No gradual rollout:** Configuration applied to all instances simultaneously
5. **Connection timeout:** 30-second timeout too long for connection failures

---

Detection

What Alerted Us

Alert	Triggered At	Threshold
API Error Rate	10:26	>5% errors for 2 min
P99 Latency	10:27	>5s for 3 min

Alert	Triggered At	Threshold
DB Connection Pool	10:25	>90% utilization

Detection Gap

While alerts fired within 3 minutes of impact start, the DB connection pool alert should have fired **before** errors propagated to customers. The 90% threshold was too high.

Resolution

Immediate Actions

1. Rolled back connection pool configuration (10:47)
2. Restarted API pods with fresh connections (10:52)
3. Cleared request backlog (10:58)
4. Verified system health (11:02)

What Made Recovery Slow

- Initial 7 minutes spent investigating API layer before identifying DB root cause
- Rollback required manual approval (another 5 minutes)
- Connection pool recovery was gradual (not instant)

Action Items

Immediate (This Week)

Action	Owner	Due	Status
Lower connection pool alert threshold to 70%	SRE	Jul 16	Done
Add connection exhaustion runbook	SRE	Jul 18	Done

Action	Owner	Due	Status
Reduce connection timeout to 5 seconds	Platform	Jul 19	Done

Short-Term (This Sprint)

Action	Owner	Due	Status
Require staging load test for DB changes	SRE Lead	Jul 26	In Progress
Implement gradual config rollout	Platform	Jul 31	In Progress
Add automated rollback on error spike	Platform	Aug 7	In Progress

Long-Term (This Quarter)

Action	Owner	Due	Status
Implement connection pooler (PgBouncer)	Platform	Sep 15	Planned
Change freeze during peak hours	SRE Lead	Aug 15	Planned
Customer-facing status page improvements	Product	Sep 30	Planned

---

Lessons Learned

What Went Well

- Alert fired quickly (3 minutes after impact)
- War room assembled rapidly (4 minutes after alert)
- Clear incident command structure
- Rollback procedure was straightforward
- Customer communication sent within 20 minutes of resolution

What Didn't Go Well

- Configuration change not load tested
- Change deployed during peak hours
- Initial investigation went in wrong direction
- Manual approval slowed rollback
- Customer impact notification delayed

Where We Got Lucky

- No data corruption or loss
- Async jobs (provisioning) were unaffected
- Peak load had already passed
- No secondary failures occurred

Metrics

Response Metrics

Metric	Target	Actual
Time to detect	<5 min	3 min
Time to acknowledge	<10 min	2 min
Time to resolve	<60 min	47 min
Time to communicate	<30 min	20 min

System Metrics During Incident

Metric	Normal	During Incident	Peak
Error rate	<0.1%	78%	78%
P99 latency	150ms	30,000ms	30,000ms
DB connections	40%	100%	100%
Request queue	0	12,847	15,203

Customer Communication

Initial Notice (10:45 AM)

> We are currently investigating increased error rates affecting the CloudForge API. Some users may experience failures when making API requests. We are working to resolve this as quickly as possible.

Update (11:05 AM)

> The issue affecting CloudForge API has been resolved. All systems are operating normally. We apologize for any inconvenience caused.

Follow-up (Next Day)

> [Detailed incident summary sent to affected Enterprise customers with credits issued per SLA terms]

Appendix

Related Incidents

- INC-2024-0312: Similar DB connection issue (different root cause)
- INC-2023-1205: Connection pool exhaustion during traffic spike

References

- Runbook: Database Connection Issues
- Change Management Policy
- Incident Response Plan

Attendees (Post-Mortem)

- SRE Team (primary)
- Platform Team
- CloudForge Engineering Lead
- VP Engineering (observer)

This post-mortem is blameless. We focus on systems and processes, not individuals.