Batch A — 40 Advanced Scenarios (detailed)

1) EC2 unreachable but system checks pass

Diagnose: Check route/SG/NACL, ENI status, instance system logs.
aws ec2 describe-instance-status, view CloudWatch & EC2 System Log.
Immediate mitigation: Reattach ENI or stop/start instance (non-destructive) if reachable. Use EC2 Serial Console.
Root fix: Correct SG/NACL/router misconfig or repair corrupted network config.
Prevent: Monitoring on ENI detach events, alerting, runbook.

---

2) S3 PUT requests spiked 10x

Diagnose: Use CloudTrail to find caller; S3 ServerAccess logs or S3 Storage Lens.
Immediate mitigation: Block the offending IAM user/role; apply bucket policy deny.
Root fix: Fix buggy application, SDK retry logic; remove malicious process.
Prevent: Alerts on unusual PUT/GET rates, IAM permissions review, VPC endpoints.

---

3) NAT Gateway costs exploded

Diagnose: VPC Flow Logs to find sources/destinations. Check CloudWatch NAT metrics and billing.
Immediate mitigation: Create VPC endpoints for S3/ECR etc. Restrict internet egress.
Root fix: Move frequent S3/ECR calls to endpoints; fix cron jobs/service pulling packages.
Prevent: Daily cost monitors, route auditing, autoscaling of data transfers.

---

4) Performance degraded after EBS encryption

Diagnose: Check CPU usage (encryption CPU overhead), instance type capabilities (AES instructions).
Immediate mitigation: Move I/O heavy workloads to instances with better crypto performance.
Root fix: Use EBS optimized instances with modern CPUs or provision gp3/io2 with higher IOPS.
Prevent: Benchmark encryption overhead in staging before rollout.

---

5) RDS latency increase during peak

Diagnose: CloudWatch RDS metrics (CPU, read/write latency, queue depth), slow query log.
Immediate mitigation: Add read replicas / scale vertically; increase connection pool.
Root fix: Query tuning, indexes, offload reads to replicas, optimize schema.
Prevent: Use performance insights, baseline query SLAs, capacity planning.

---

6) RDS storage skyrockets

Diagnose: Check binary log retention, slow query generation of temp tables, large backups.
SHOW BINARY LOGS; review snapshot schedule.
Immediate mitigation: Purge or archive logs, increase storage if urgent.

Root fix: Adjust log retention, clean temp tables, schedule backups at off-peak times.
Prevent: Alerts at storage thresholds, automated auto-scale.

---

7) DynamoDB Global Table replication lag
Diagnose: CloudWatch for throttling metrics, check hot keys, CloudWatch Alarms on consume.
Immediate mitigation: Increase WCU/WCU temporarily or switch to on-demand mode.
Root fix: Partition key redesign, implement exponential backoff, use adaptive capacity.
Prevent: Load testing across regions, autoscaling policies.

---

8) Lambda timeouts
Diagnose: Check CloudWatch logs, X-Ray traces; VPC ENI cold start delays if in VPC.
Immediate mitigation: Increase timeout; provisioned concurrency for critical functions.
Root fix: Reduce cold start by keeping function warm, avoid heavy SDK init on cold start, move out of VPC if possible with VPC endpoints.
Prevent: Instrumentation (X-Ray), SLOs for Lambda durations.

---

9) CloudFront 504s
Diagnose: CloudFront logs -> origin response timeouts; check origin health.
Immediate mitigation: Increase origin timeout in distribution; scale origin.
Root fix: Address backend performance, tune origin timeouts and caching.
Prevent: Cache dynamic content if possible; use origin shielding.

---

10) ELB deregistering healthy targets
Diagnose: ELB health check path/port mismatch, check target group health, app process listening port.
Immediate mitigation: Correct health check path or increase timeout; re-register targets.
Root fix: Fix app readiness to respond quickly.
Prevent: Test health checks during deployment; use lifecycle hooks in ASG.

---

11) Pod running but app dead
Diagnose: kubectl exec into pod, check process list, application logs. Confirm readiness vs liveness probe configuration.
Immediate mitigation: Restart container or pod to restore state.
Root fix: Adjust liveness probe (should kill hung processes) and readiness probe to prevent traffic to unhealthy pods.
Prevent: Use proper probes and graceful shutdowns.

---

12) kube-proxy high CPU
Diagnose: kubectl get endpoints/services huge counts, iptables rule explosion.
Immediate mitigation: Restart kube-proxy, switch mode to IPVS.
Root fix: Reduce service churn, optimize controllers creating endpoints, use headless services for many clients.
Prevent: Use IPVS mode and monitor service counts.

13) CoreDNS SERVFAIL
Diagnose: kubectl logs -n kube-system coredns ; check upstream DNS, node DNS resolution, CoreDNS config.
Immediate mitigation: Restart CoreDNS pods; scale up replicas.
Root fix: Fix upstream DNS or VPC DNS settings, ensure CoreDNS has enough resources.
Prevent: Alerting on CoreDNS error rates; caching.

---

14) K8s API server slow
Diagnose: Check control plane metrics, etcd performance, large CRD list or large number of objects.
Immediate mitigation: Scale control plane (managed EKS or control plane autoscaling), reduce watchers.
Root fix: Cleanup unused CRDs, reduce event rate, tune etcd resources.
Prevent: Limit scale of controllers and use leader election.

---

15) PVC bind delays
Diagnose: StorageClass mismatch, check kubectl describe pvc for events and PVs; AZ mismatch.
Immediate mitigation: Create PV manually if urgent or reschedule pod to correct AZ.
Root fix: Use multi-AZ storage classes or dynamic provisioning with correct parameters.
Prevent: Volume topology awareness and validating YAML.

---

16) CrashLoopBackOff although code OK
Diagnose: Check container entrypoint, permissions, environment variables, missing files. kubectl logs -p.
Immediate mitigation: Adjust entrypoint or fix config; restart pod.
Root fix: Fix Dockerfile entrypoint, config management, secrets mounting.
Prevent: Integration tests in CI that start containers with actual entrypoints.

---

17) OOMKilled persists despite more memory
Diagnose: Heap leak, duplicated memory usage, or a process that grows over time. Collect heap/heapdump.
Immediate mitigation: Increase limit short-term; restart pod with HPA.
Root fix: Fix leak in code, tune JVM GC, memory limits/requests properly.
Prevent: Memory profiling, alerts for memory growth trends.

---

18) Node NotReady due to disk pressure
Diagnose: Node kubectl describe node, check kubelet logs, df -h on node.
Immediate mitigation: Clean large files (logs), cordon and drain node, replace node.
Root fix: Add log rotation, increase disk sizes, move ephemeral storage to EBS.
Prevent: Disk usage alerts, log forwarding to central store.

---

19) Autoscaler adds nodes but pods still Pending

Diagnose: Check pod events for pod affinity, taints, wrong instance types or insufficient resources per pod (GPU, local SSD).
Immediate mitigation: Add nodegroup with required instance type/labels or remove taints.
Root fix: Ensure cluster-autoscaler supports the nodegroup/labels required, correct resource requests.
Prevent: Pod resource request tuning and nodegroup planning.

---

20) Downtime during rolling update
Diagnose: Check Deployment strategy maxUnavailable, readiness probes, pod startup time.
Immediate mitigation: Increase replicas, set PDBs, tweak rollingStrategy to smaller batches.
Root fix: Improve startup time, use readiness probes to delay serving.
Prevent: Canary deployments and pre-warming.

---

21) Terraform state lost
Diagnose: Check S3 versioning, backend misconfiguration; list state file versions.
Immediate mitigation: Restore state from S3 versioning or backup, use terraform import to reconstruct resources if needed.
Root fix: Strict backend management, enable state locking and versioning.
Prevent: Enforce state snapshots, automated backups.

---

22) Two devs applied Terraform simultaneously
Diagnose: Locking failure or overwritten state. Check DynamoDB lock table if used.
Immediate mitigation: Reconcile resource changes manually, run terraform refresh, import missing resources.
Root fix: Enforce remote state locking (DynamoDB) and PR process for infra changes.
Prevent: Pre-apply approvals, CI gated infrastructure changes.

---

23) CI pipeline runs but deploy not reaching cluster
Diagnose: Check kubeconfig used by pipeline, RBAC, network connectivity from runner.
Immediate mitigation: Verify runner credentials and context; redeploy with correct kubeconfig.
Root fix: Secure and centrally manage service accounts; rotate tokens.
Prevent: Pipeline preflight checks confirm cluster connectivity.

---

24) Helm upgrade failed, cluster unstable
Diagnose: Partial resources applied, failed CRDs, hooks stuck. helm history and helm status to inspect.
Immediate mitigation: helm rollback to last stable release. Clean orphan resources.
Root fix: Validate manifests, add precheck hooks and dry runs.
Prevent: Use helm test and CI linting, add resource quotas.

---

25) Applied Terraform to production accidentally
Diagnose: Pause pipelines, review change plan output.
Immediate mitigation: Stop the apply if possible; roll back to last known state; restore from backups.
Root fix: Separate state files per environment, require manual approvals for prod.
Prevent: Protect production workspace with approval gates and limited exec permissions.

---

26) Docker build 10 minutes
Diagnose: Large base images, many layers, dependencies reinstalled each build.
Immediate mitigation: Use caching, build on bigger hardware for now.
Root fix: Multi-stage builds, smaller base images (alpine/distroless), cache dependencies.
Prevent: Use CI cache and prebuilt base images.

---

27) Jenkins agent containers restart often
Diagnose: Check Jenkins logs & agent container logs for OOM or disk full.
Immediate mitigation: Increase agent resources/ephemeral storage.
Root fix: Clean workspace, use ephemeral agents with enough disk size; limit concurrent builds.
Prevent: Monitor disk, set quotas, ephemeral agent lifecycle.

---

28) Secrets pushed to Git
Diagnose: Identify commit(s) using git log and git grep.
Immediate mitigation: Rotate compromised secrets immediately. Remove secrets from repo using git filter-repo or BFG, push forced history rewrite.
Root fix: Move to Secrets Manager/SSM Parameter Store and enforce pre-commit hooks.
Prevent: Secret scanning in CI, deny commits with secrets, rotate secrets regularly.

---

29) Works in staging but fails prod
Diagnose: Compare env vars, IAM roles, VPC/network differences, feature flags.
Immediate mitigation: Recreate prod with same configs as staging for controlled tests; revert to previous stable release.
Root fix: Environment parity, config as code, use the same secrets and permission models.
Prevent: Promote artifacts, not builds; blue/green strategy.

---

30) Unexpected auto-reverts
Diagnose: Auto rollback policies triggered by health checks or custom monitors. Check pipeline and deployment health events.
Immediate mitigation: Identify failing checks and either fix the reason or temporarily disable auto-revert to inspect.
Root fix: Fix underlying instability causing health checks to fail (performance, smoke tests).
Prevent: Tighter pre-prod validation, canary waits.

---

31) 5xx errors but CPU/memory normal

Diagnose: Check DB connection pools, downstream services, thread dumps, timeouts. Use tracing (Jaeger/X-Ray).
Immediate mitigation: Restart flaky downstream services, increase DB pool or connections temporarily.
Root fix: Fix connection leaks, tune timeouts, increase retry/backoff.
Prevent: Instrument and set SLOs for downstream calls.

---

32) Daily latency spikes at same time
Diagnose: Investigate cron jobs/backup snapshots, DB compactions, garbage collection. Correlate with cloudwatch metrics.
Immediate mitigation: Reschedule heavy jobs to off-peak.
Root fix: Stagger jobs, optimize backups.
Prevent: Run capacity planning and schedule maintenance windows.

---

33) Memory leak in production
Diagnose: Capture heap dump, analyze with profiler (YourKit/VisualVM), trace leak path.
Immediate mitigation: Restart pods in a rolling fashion to clear memory.
Root fix: Fix code causing leak, add stream/iterator closures.
Prevent: Add memory growth alarms, periodic Canary restarts.

---

34) DB connection exhaustion across services
Diagnose: Check max_connections, connection pool usage per service, long transactions.
Immediate mitigation: Increase DB connection limit or add connection pooling (pgbouncer).
Root fix: Implement connection pooling, reduce per-request connections, tune pool sizes.
Prevent: Monitor connection counts, circuit breaker for DB.

---

35) EBS IOPS exhaustion
Diagnose: CloudWatch EBS metrics, identify high I/O disks, check instance type for EBS optimization.
Immediate mitigation: Move hotspot I/O to provisioned IO (io2), use caching (ElastiCache), or add sharding.
Root fix: Restructure workload, use higher IOPS volumes or scale horizontally.
Prevent: Configure appropriate IOPS and monitor.

---

36) Error budget exceeded
Diagnose: Review SLO metrics, determine common failure modes.
Immediate mitigation: Freeze releases to production; prioritize reliability work.
Root fix: Fix the top contributors to errors.
Prevent: Invest part of sprint time for reliability and safety for releases.

---

37) Client sees slowness but server shows healthy
Diagnose: Test from client location (traceroute), CDN edge health, Route53, DNS TTLs, network latency.

Immediate mitigation: Use region-specific endpoints or improve CDN caching.
Root fix: Increase edge caching, check peering issues with ISP.
Prevent: Synthetic monitoring from multiple geographies.

---

38) Kubernetes network jitter rising
Diagnose: Check CNI plugin metrics, node saturation, iptables complexity.
Immediate mitigation: Move high throughput pods to separate nodes, restart CNI.
Root fix: Switch to high-performance CNI/IPVS, upgrade kernel and CNI plugin.
Prevent: Network QoS and node sizing, isolate noisy neighbors.

---

39) Logs increased 20x
Diagnose: Identify source service enabling debug, look for log spam or error loops.
Immediate mitigation: Temporarily reduce log level for that service, clean up disk space.
Root fix: Fix the underlying error causing log spam.
Prevent: Set log rate limits and structured logging with sampling.

---

40) Incident repeats despite fixes
Diagnose: Review RCA, check whether root cause was fixed or only symptoms. Examine change control & test coverage.
Immediate mitigation: Implement temporary controls (circuit breakers, throttling) to stop recurrence.
Root fix: Comprehensive root cause elimination (code fix + infra change + process).
Prevent: Add regression tests, automation to prevent recurrence, documented runbooks.

Batch B — 40 More Advanced Scenarios (detailed answers)
(These correspond to the second set of 40 scenarios I gave earlier — answers follow the same Diagnose → Immediate → Root-Cause → Prevent pattern.)
41) VPC IP exhaustion because of EKS nodes
Diagnose: aws ec2 describe-subnets to check allocated IPs per subnet. Count pod IP allocation using AWS VPC CNI.
Immediate mitigation: Add new subnets with larger CIDR and move nodegroups, or enable WARM_IP_TARGET to reduce pod-per-node IP.
Root fix: Re-architect with larger CIDRs or use secondary CIDR ranges/VPC peering or use CNI overlay (kube-ovn/calico).
Prevent: Plan IP addressing for scale, use pod IP reuse features.

---

42) ALB high target response time but EC2 CPU normal
Diagnose: Check network latency, backend thread pools, unhealthy dependency (DB), disk I/O, and connection reuse. Use X-Ray/tracing.
Immediate mitigation: Scale backend, add cache, increase connection pool.
Root fix: Identify blocked threads or slow downstream service; optimize queries or code.
Prevent: Add distributed tracing and alert on tail latency.

---

43) RDS failover too slow

Diagnose: RDS failover time depends on engine and replication lag; check CloudWatch failover metrics.
Immediate mitigation: Use Multi-AZ provisioned with synchronous replication (Aurora is faster).
Root fix: Migrate to Aurora Global DB or use read-replicas with promoted failover and fast DNS TTLs.
Prevent: Test failovers regularly and tune maintenance windows.

---

44) S3 replication causing duplicate data
Diagnose: Check replication configuration, multiple replication rules, or retry loops from source.
Immediate mitigation: Stop the process causing duplicate writes, delete duplicates if safe.
Root fix: Fix replication rules, ensure object versioning and idempotent writes.
Prevent: S3 events/metrics and deduplication keys.

---

45) Lambda cold starts
Diagnose: Check duration histograms, factor in VPC ENI attach time, package size, runtime.
Immediate mitigation: Use provisioned concurrency for critical functions.
Root fix: Reduce package size, lazy-init modules, avoid VPC when not needed, use VPC endpoints.
Prevent: Warmers, metrics on cold start frequency.

---

46) NAT Gateway bill — repeated (detailed)
(See #3) additionally inspect ECR pulls and periodic updates from many instances. Use CloudWatch Contributor Insights to find top talkers.

---

47) ENI attach failures during ASG scaling
Diagnose: Check EC2 instance limit for ENIs per instance type, IAM permissions for CNI to attach ENIs.
Immediate mitigation: Use instance types with higher ENI quotas or change pod density.
Root fix: Update instance types or CNI plugin settings, ensure amazon-vpc-cni has permissions.
Prevent: Capacity planning for ENIs and pod density.

---

48) Intermittent packet drops across VPCs
Diagnose: VPC peering / Transit Gateway route table misconfigurations, SG NACL intermittent rules, network hardware events.
Immediate mitigation: Reconfigure route tables or failover to alternative path, open necessary ports.
Root fix: Fix route or TGW attachments, increase MTU alignment across path.
Prevent: Redundant network paths and monitoring.

---

49) Migrate VPC CIDR without downtime
Diagnose: This is a delicate migration. Plan for new VPC, transit gateway peering, route propagation.
Immediate mitigation: Create new VPC with desired CIDR, set up VPC peering or TGW and gradually move resources via new subnets (use elastic IPs, DNS cutover).
Root fix: Full migration plan.
Prevent: Plan VPC allocation ahead, use secondary CIDRs for expansion.

---

50) CloudFront caching not working for dynamic content
Diagnose: Check Cache-Control headers, query string forwarding, cookies and CacheBehavior.
Immediate mitigation: Adjust Cache-Control to allow caching; configure behaviors for specific paths.
Root fix: Update application to set proper caching headers and TTLs for dynamic/static separation.
Prevent: Test caching behavior in staging.

---

51) Pods stuck Terminating
Diagnose: Finalizers on resource, webhook blocking termination, PV unmount issues. kubectl get pod -o yaml to inspect finalizers.
Immediate mitigation: Remove finalizer if safe; force delete as last resort.
Root fix: Fix webhook handlers or ensure PV unmounts properly.
Prevent: Build graceful shutdown handlers and test.

---

52) CoreDNS pods restarting — more detail
Diagnose: Check CPU/memory, upstream DNS unreachable, recursive loops.
Immediate mitigation: Scale CoreDNS replicas; point CoreDNS to stable resolvers.
Root fix: Fix networking issues or upstream DNS failure.
Prevent: Resource limits and readiness probes for CoreDNS.

---

53) HPA not triggering
Diagnose: Check metrics server is running (metrics-server), proper metrics configured; check target utilization and resource requests.
Immediate mitigation: Manually scale if necessary.
Root fix: Ensure metrics server is healthy or use external metrics adapter.
Prevent: Validate metric availability in pre-prod.

---

54) Cluster Autoscaler adds nodes but pods Pending — again
(See #19) additional angle: pods may have nodeSelector requiring GPU or taints. Add matching nodegroup.

---

55) Worker nodes failing to join cluster
Diagnose: Bootstrap script errors, kubelet certificate issues, wrong AMI, IAM role or security group blocking.

Immediate mitigation: Inspect cloud-init logs on node, replace node.
Root fix: Fix AMI or bootstrap process.
Prevent: Bake AMIs with required kubelet components.

---

56) App slow after rolling update
Diagnose: Check new version resource usage, compatibility changes, database migration steps causing locks.
Immediate mitigation: Rollback to previous version.
Root fix: Improve migration strategy and perform canary testing.
Prevent: Blue/green and feature flags.

---

57) Deployment stuck at 0/5 updated pods
Diagnose: ImagePullBackOff, insufficient resources, failed init containers. kubectl describe rs and pods.
Immediate mitigation: Fix image path/credentials or add resources.
Root fix: CI ensures images are pushed to correct registry with tags.
Prevent: Pre-deploy smoke test.

---

58) OOMKilled despite increase — deeper
Diagnose: Host-level memory fragmentation or swap issues, cgroup limits, memory leak persists.
Immediate mitigation: Add swap or add nodes, restart pods.
Root fix: Profiling the app; fix memory leaks.
Prevent: Memory pressure monitoring.

---

59) Noisy neighbor in shared cluster
Diagnose: Identify pod with high resource consumption using kubectl top pod and node metrics.
Immediate mitigation: Apply quotas and limits; cordon node and move noisy pods.
Root fix: Enforce resource requests & limits, use namespaces with ResourceQuotas.
Prevent: Pod QoS classes and burstable handling.

---

60) CSI driver PVC mount failures
Diagnose: Check CSI driver logs, CSI controller and node plugin status, driver compatibility with K8s version.
Immediate mitigation: Restart CSI pods, re-provision PV.
Root fix: Update CSI driver, fix IAM roles for driver.
Prevent: Compatibility testing and readiness probes for CSI.

---

61) Terraform corrupted state (more detail)
Diagnose: Identify state version in S3, check lock.
Immediate mitigation: Restore previous version from S3, run terraform state rm to remove problematic resources.

Root fix: Add strict locking and pre-apply reviews.
Prevent: Enforce CI validation and backups.

---

62) Terraform plan shows large unwanted changes
Diagnose: Compare local vs remote state, provider upgrades, resource rename.
Immediate mitigation: Pin provider version; import resources back into state if they were created outside TF.
Root fix: Use terraform import carefully and ensure state refresh.
Prevent: Terraform runbook and provider pinning.

---

63) Pipeline runs but deployment not reaching cluster (revisit)
(Same as #23) include checking Kubernetes RBAC and ServiceAccount token TTL.

---

64) Helm upgrade failed — cluster unstable (revisit)
(See #24) add helm hooks idempotency.

---

65) Applied TF in prod mistakenly — further actions
(See #25) and implement immediate freeze and postmortem.

---

66) Optimize Docker build time (more)
Diagnose: Repeated apt-get/ package installs, no build cache.
Immediate mitigation: Use build cache retention; use buildkit.
Root fix: Separate dependencies into separate layers and use cache from registry.
Prevent: CI caching and reproducible builds.

---

67) Jenkins agent container restarts — additional
Include ephemeral storage pressure from logs; use docker system prune cron.

---

68) Secrets in Git — legal/compliance actions
Also audit access logs and notify security team; rotate secrets across services and dependent CI.

---

69) Staging vs Prod mismatch — more
Add feature toggles and ensure IaC templates identical; use environment matrix.

---

70) Auto reverts — more depth
Investigate health checks flapping thresholds, circuit breakers; perhaps overzealous monitoring triggers.

---

71) 5xx errors — deep trace
Use distributed tracing to locate service causing 5xx; inspect rate limiter behavior and queue lengths.

---

72) Nightly latency spike — further

Check snapshot/backup windows for RDS/Aurora; also check S3 lifecycle jobs and repackaging jobs.

---

73) Memory leak in prod — extra
If language is Java, collect GC logs and perform heap dump analysis; if Python, check reference cycles.

---

74) DB connection spikes across microservices — additional
Enforce connection pooling libraries (HikariCP), tune max_connections per app and DB.

---

75) Disk IOPS exhaustion — more
Consider read replicas for read-heavy workloads, upgrade to provisioned IOPS, or use local NVMe for caches.

---

76) Error budget exceeded — extra
Communicate to business stakeholders and schedule reliability sprint.

---

77) Client slowness with server healthy — more
Check CDN edge misconfigurations, large TLS handshakes, renegotiation storms.

---

78) K8s network jitter — more
Examine CNI plugin version, tune MTU, and check node-level CPU steal affecting network stacks.

---

79) Logs increase 10x — additional steps
Set up adaptive logging where high severity levels are retained but debug-level logs sampled.

---

80) Repeat incident — enforce closure
Define success criteria for RCA closure and require proof-of-fix (automated test) before closing.