Arcline

November 6, 2026

SAMPLE DELIVERABLES · COMPLETE SET

Arcline sample reports

All 17 pages of sample deliverables from the reports page in one document: the findings report, the cloud cost analysis, the 90-day remediation plan, and the audit evidence pack.

SAMPLE Fictional demonstration data · Prepared by Arcline

CONTENTS

1 **Reliability & Cost Audit: Findings Report** — 6 pp.
The flagship written report: executive summary, prioritized findings, and the next 90 days.

2 **Cloud Cost Analysis** — 3 pp.
Where the spend goes, what is not buying anything, and the seven actions that recover it.

3 **90-Day Remediation Plan** — 5 pp.
The owner-by-owner sequence: what to delete, what to build, and what to leave alone.

4 **Audit Evidence Pack: SOC 2 Controls** — 3 pp.
Ten controls, the evidence behind each one, and the two gaps that would fail an audit.

Fictional client (Northwind Payments). No real client information is included anywhere in these samples.

Arcline

November 6, 2026

SAMPLE DELIVERABLE

Reliability & Cost Audit: Findings Report

A written, prioritized summary of what the platform costs, what it risks, and what to fix first.

SAMPLE Fictional demonstration data · Prepared by Arcline

PREPARED FOR

Northwind Payments

Fintech / payments (PCI DSS scope) · 28 engineers · 3 Kubernetes clusters · Azure and AWS

Cloud spend, the Kubernetes platform, delivery pipeline, on-call practice, and the evidence behind SOC 2 and PCI DSS controls, across production and staging.

01 Executive summary

$24.6K	3	3	2
RECOVERABLE SPEND / MONTH	HIGH FINDINGS	MEDIUM FINDINGS	LOW FINDINGS

This report summarizes a reliability and cost audit of Northwind Payments, completed in November 2026. We reviewed the cloud spend across both providers, the Kubernetes platform, the delivery pipeline, the on-call practice, and the evidence behind the SOC 2 and PCI DSS controls. The goal was to find what the platform costs and what it risks, not to grade the team.

The platform is better built than most at this size: three clusters with consistent namespaces, GitOps already in place for two thirds of services, and a median deploy time of eleven minutes with a working rollback. Nothing we found suggests an active breach or an unrecoverable failure. But $24,600 a month is not buying anything, and the two controls an auditor will test first — privileged audit logging and tested restores — are not yet in place. Both are cheap to fix and expensive to explain.

WHAT'S GOING WELL

- Three clusters, one namespace convention, and real separation between staging and production
- GitOps already covers two thirds of services, and rollback is practised rather than theoretical
- Median deploy time of eleven minutes, with a change history that maps to pull requests
- Image scanning blocks critical vulnerabilities before a deploy
- No evidence of a breach, and no unrecoverable failure in the platform's history

TOP PRIORITIES

1. Agree SLOs for checkout and adopt an error-budget policy
2. Delete the idle capacity and orphaned disks this week
3. Turn on Kubernetes audit logging before the next audit cycle

02 Prioritized findings

Every finding, ranked by what it costs in dollars and in risk. Detail follows in the next section.

ID	FINDING	SEVERITY	OWNER	TIMELINE
R-01	No SLOs or error budgets on the payment path	HIGH	Platform	within 14 days
R-02	38% of production spend is not buying anything	HIGH	Platform	within 14 days

ID	FINDING	SEVERITY	OWNER	TIMELINE
R-03	Kubernetes audit logging is off, and control-plane logs last three days	HIGH	Platform	before the next audit
R-04	Terraform drift: 14 production resources are outside the code	MEDIUM	Platform	within 30 days
R-05	On-call is one engineer, with no escalation and no runbooks	MEDIUM	Platform	within 30 days
R-06	Backups have never been restored, and retention is seven days	MEDIUM	Platform	within 30 days
R-07	40% of workloads have no resource requests or limits	LOW	Platform	within 60 days
R-08	A shared cluster-admin kubeconfig lives in the team vault	LOW	Platform	within 60 days

03 Detailed findings

R-01 · No SLOs or error budgets on the payment path

HIGH

Affected: checkout-api, ledger-api, webhook-dispatcher · Owner: platform · within 14 days

WHY IT MATTERS Without a written objective, reliability is whatever the loudest incident last week made it. The team cannot say whether last month was good or bad, so every conversation about shipping faster becomes an argument rather than arithmetic.

WHAT WE FOUND Three services sit on the critical path, and none has an SLI, an SLO, or an error budget. The dashboards show CPU and memory, so nobody can answer “what share of checkouts succeeded last month”. Alerts fire on host metrics: a busy database pages someone at 2am, while a spike in failed checkouts does not page anyone at all.

EVIDENCE No SLO definitions in the repository or the monitoring configuration. 63 alert rules, 58 of them on infrastructure metrics, 5 on error rate, none tied to a user-visible objective.

RECOMMENDATION Define three SLIs (checkout success, checkout latency p95, webhook delivery), agree the SLOs with product, and adopt an error-budget policy. Retire any alert that is not tied to a symptom threatening an SLO.

R-02 · 38% of production spend is not buying anything

HIGH

Affected: Azure and AWS production accounts · Owner: platform · within 14 days

WHY IT MATTERS Every dollar of waste is a dollar not spent on reliability, and the waste compounds quietly because no single line on the invoice looks wrong.

WHAT WE FOUND Production runs at $64,200 a month across two clouds. $24,600 of that is idle or oversized: a node pool left running after a cancelled migration, a 32-vCore Postgres averaging 8% CPU, 42 unattached disks, and 90-day log retention that nobody queries. None of it appears as an anomaly on the bill.

EVIDENCE Billing export, 90 days. Idle node pool: zero pods for 61 days. Postgres average CPU 8.2%, peak 19%. Orphaned disks: 42 unattached, $2,900/month.

RECOMMENDATION Delete the idle pool and the orphaned disks this week, rightsize the database next sprint and commit the stable baseline to a savings plan. The detail is in the Cloud Cost Analysis.

R-03 · Kubernetes audit logging is off, and control-plane logs last three days

HIGH

Affected: Both production AKS clusters · Owner: platform · before the next audit

WHY IT MATTERS	An auditor will ask who changed what, and when. Today the platform cannot answer, which turns a routine audit into a reconstruction project and an incident into a guess.
WHAT WE FOUND	The clusters run the default audit policy, so most write operations are never recorded. Control-plane logs are retained for three days. There is no record of a kubectl exec into production and no way to distinguish a reviewed deploy from a manual change made during an incident.
EVIDENCE	Audit policy unset on both clusters. Log Analytics retention 3 days. No audit-log export configured. No RBAC change history beyond what Git happens to contain.
RECOMMENDATION	Apply a baseline audit policy (Metadata at minimum, RequestResponse for secrets and RBAC), ship the logs to immutable storage, and set 12-month retention. This is also the evidence for CC7.2.

R-04 · Terraform drift: 14 production resources are outside the code

MEDIUM

Affected: Production workspace · Owner: platform · within 30 days

WHY IT MATTERS	The repository says one thing and the cluster says another. The next apply either overwrites someone's hotfix or silently reverts a security setting, and nobody knows which until it happens.
WHAT WE FOUND	A plan against the production workspace shows 14 resources would change with no code change: two network security groups, a node-pool autoscaler setting, nine Kubernetes objects applied by hand during an incident, and two secrets managed outside the pipeline.
EVIDENCE	terraform plan, production: 14 to change, 0 to add, 0 to destroy. Git history contains no corresponding commit.
RECOMMENDATION	Import or codify the drifted resources, then make the pipeline the only path to production. Add a nightly drift check that opens a ticket instead of waiting to be discovered during an incident.

R-05 · On-call is one engineer, with no escalation and no runbooks

MEDIUM

Affected: All production services · Owner: platform · within 30 days

WHY IT MATTERS	A rotation of one is not a rotation. When that person is asleep, ill, or on a plane, nobody is on call, and the bus factor is exactly the same person.
WHAT WE FOUND	The same engineer takes every page, every week. There is no secondary, no escalation policy beyond a message in Slack, and no runbooks for the five alerts that page most often. Two of those alerts fired more than 40 times this quarter.
EVIDENCE	PagerDuty schedule: one member. Escalation policy: notify the same user. 213 pages in 90 days; the top two alerts account for 88 of them.
RECOMMENDATION	Design a two-person rotation with an escalation path, write runbooks for the top five alerts, and make alert quality a standing monthly review. This is on-call design, not 24/7 staffing.

R-06 · Backups have never been restored, and retention is seven days

MEDIUM

Affected: Managed Postgres, persistent volumes · Owner: platform · within 30 days

WHY IT MATTERS	An untested backup is a hope. A seven-day window also means a slow, quiet corruption is discovered after the good copies have already expired.
WHAT WE FOUND	Nightly snapshots run and report success. Nobody has restored one in the platform's history. Database retention is seven days, persistent volumes have no snapshot policy at all, and no recovery time is documented.
EVIDENCE	Backup policy: 7-day retention, no point-in-time recovery. No restore record. PVC snapshot policy absent.
RECOMMENDATION	Enable point-in-time recovery, extend retention to 35 days, configure volume snapshots, and run a quarterly restore drill that is timed and written down. That record is the evidence A1.2 asks for.

R-07 · 40% of workloads have no resource requests or limits

LOW

Affected: 11 of 27 Deployments in production · Owner: platform · within 60 days

WHY IT MATTERS	Without requests the scheduler is guessing, so one noisy workload can starve its neighbours and the node pool is sized from fiction rather than from measured demand.
WHAT WE FOUND	Eleven Deployments define no CPU or memory requests and 16 define no limits. The node pools are sized from peak observed usage, which is inflated by exactly this absence.
EVIDENCE	kubectl audit across production namespaces: 11/27 without requests, 16/27 without limits. No LimitRange in any namespace.
RECOMMENDATION	Set requests from a two-week usage baseline, add a LimitRange per namespace, and let the autoscaler reclaim the difference. Expect a smaller node pool as a result.

R-08 · A shared cluster-admin kubeconfig lives in the team vault

LOW

Affected: Both production clusters · Owner: platform · within 60 days

WHY IT MATTERS	One credential that can do anything, held by everyone, is the finding an auditor writes first — and it makes every other access control in the report unprovable.
WHAT WE FOUND	A single cluster-admin kubeconfig is stored in the shared password manager and used for break-glass access by four engineers. There is no rotation schedule and no record of who used it or when.
EVIDENCE	Vault item shared with four users. Last rotated 14 months ago. No session logging on either cluster.
RECOMMENDATION	Replace it with per-person, time-bound access through the identity provider, and log every session. Rotate the shared credential now and retire it once the replacement is in place.

04 The next 90 days

The full owner-by-owner sequence is in the 90-Day Remediation Plan. This is the order of work and what it is worth.

WHEN	WORK	MONTHLY EFFECT	EFFORT
Week 1–2	Delete the idle node pool and 42 orphaned disks	+$7,200	1 day
Week 1–2	Agree SLOs and an error-budget policy for checkout	Risk	2 days
Week 3–4	Enable audit logging and 12-month retention	Compliance	3 days
Week 3–4	Rightsize the managed database	+$5,200	1 week
Week 5–8	Codify drift and add a nightly drift check	Risk	1 week
Week 5–8	Two-person rotation, escalation, five runbooks	Risk	1 week
Week 9–12	Point-in-time recovery and the first timed restore drill	Compliance	3 days
Week 9–12	Commit the stable baseline to a savings plan	+$6,800	1 week

Arcline

November 6, 2026

SAMPLE DELIVERABLE

Cloud Cost Analysis

Where $64,200 a month goes, what is not buying anything, and the seven actions that recover $24,600 of it.

SAMPLE Fictional demonstration data · Prepared by Arcline

PREPARED FOR

Northwind Payments

Fintech / payments (PCI DSS scope) · 28 engineers · 3 Kubernetes clusters · Azure and AWS

Billing period: trailing 90 days · Read-only access to the Azure and AWS accounts

01 Spend summary

$64.2K	$24.6K	$295K	38%
MONTHLY CLOUD SPEND	RECOVERABLE / MONTH	ANNUALISED	NOT BUYING ANYTHING

Northwind spends $64,200 a month across Azure and AWS. Of that, $24,600 is idle, oversized, or retained past any use. That is 38% of the bill and roughly $295,000 a year.

The important number is not the total. It is that the waste is concentrated: seven actions recover essentially all of it, and four of those are delete-this-now with no engineering trade-off. The rest requires a week of work each, and the largest single item — the savings plan — only becomes safe once the idle capacity is gone.

02 Where the money goes

CATEGORY	MONTHLY	WHAT IS IN IT
AKS node pools	$22,400	6 pools across 3 clusters; 2 are oversized, 1 is idle
Managed Postgres	$11,800	32 vCores, 8.2% average CPU, peak 19%
Storage and disks	$8,600	42 unattached disks and 31 old snapshots
Egress and bandwidth	$6,900	Cross-region traffic between clusters and a legacy peer
Observability	$5,400	90-day retention, 71% of ingestion from debug logs
Everything else	$9,100	Key Vault, DNS, load balancers, managed identities
Total	$64,200	Azure and AWS, trailing 90-day average

03 What is not buying anything

RESOURCE	MONTHLY	WHY IT IS WASTE	ACTION
aks-migration-temp pool	$4,300	Node pool for a migration that finished in September	Delete
Postgres prod-primary	$5,200	32 vCores at 8.2% average; peak 19%	Resize to 16
42 unattached managed disks	$2,900	Left behind by deleted PVCs and old VMs	Delete
31 snapshots older than a year	$1,100	Outside retention policy, never referenced	Delete
Log Analytics, 90 days	$3,400	71% of ingestion is debug logs; no query older than 30 days	Tier + archive

RESOURCE	MONTHLY	WHY IT IS WASTE	ACTION
Cross-region peer (legacy)	$2,700	Carries one retired batch job and nothing else	Retire

04 Savings plan

ACTION	MONTHLY	EFFORT	CONFIDENCE	BASIS
Delete the idle node pool	+$4,300/mo	1 day	Certain	zero pods for 61 days
Rightsize Postgres to 16 vCores	+$5,200/mo	1 week	High	average 8.2% CPU, peak 19%
Delete orphaned disks and snapshots	+$2,900/mo	1 day	Certain	42 unattached, 31 stale
Commit the stable baseline to a savings plan	+$6,800/mo	1 week	High	68% of spend is steady-state
Rightsize the remaining node pools	+$2,000/mo	1 week	Medium	after requests and limits are set
Tier logs and cut debug ingestion	+$1,900/mo	3 days	High	71% of ingestion is debug
Cut Log Analytics retention to 30 days + archive	+$1,500/mo	2 days	High	nobody queried past 30 days
Total recoverable	+$24,600/mo	≈ 3 weeks of work		

ABOUT THE NUMBERS

Figures are from a 90-day billing export and a read-only review of the account. Savings are monthly run-rate, not credits, and they assume the resources are deleted rather than paused — a paused resource tends to come back. The savings plan is the only line that is a commitment rather than a deletion, which is why it comes last.

Arcline

November 6, 2026

SAMPLE DELIVERABLE

90-Day Remediation Plan

The owner-by-owner sequence: what to delete, what to build, and what to deliberately leave alone.

SAMPLE Fictional demonstration data · Prepared by Arcline

PREPARED FOR

Northwind Payments

Fintech / payments (PCI DSS scope) · 28 engineers · 3 Kubernetes clusters · Azure and AWS

Cloud spend, the Kubernetes platform, delivery pipeline, on-call practice, and the evidence behind SOC 2 and PCI DSS controls, across production and staging.

01 How to use this plan

This is the owner-by-owner sequence behind the findings report. It is ordered by what removes risk or cost fastest per day of effort, not by severity: two of the highest-value items are deletions that take a day, and the compliance work is scheduled so it lands before the audit window rather than during it.

Every item names an owner, an effort estimate, and its monthly effect. Nothing here requires a migration, a new platform, or a freeze on product work.

02 First 14 days

Delete the idle node pool

+$4,300/MO

Platform · 1 day

Zero pods for 61 days. Nothing depends on it; confirm with the workload team and delete.

Delete 42 unattached disks and 31 stale snapshots

+$4,000/MO

Platform · 1 day

Cross-check tags against the retention policy first, then delete. No application impact.

Agree the three SLIs and the SLOs

RISK

Platform + product · 2 days

Checkout success, checkout p95, webhook delivery. Three numbers, one owner each, agreed in writing.

Adopt an error-budget policy

RISK

Platform + product · 1 day

Above 50% budget: ship freely. Below 25%: reliability work takes priority. Exhausted: freeze and fix.

03 Days 15–30

Enable Kubernetes audit logging

COMPLIANCE

Platform · 3 days

Metadata level baseline, RequestResponse for secrets and RBAC. Ship to immutable storage.

Set 12-month retention on audit and application logs

+$1,500/MO

Platform · 2 days

30 days hot, 12 months archive. This is the CC7.2 evidence and it costs less than the current 90-day hot tier.

Rightsize the managed database

+$5,200/MO

Platform · 1 week

16 vCores from a 14-day baseline. Schedule a maintenance window and watch p95 during the cutover.

04 Days 31–60

Codify the drift and add a nightly check

RISK

Platform · 1 week

Import the 14 resources, apply from the pipeline only, and open a ticket when drift is detected.

Design a two-person on-call rotation

RISK

Platform · 1 week

Primary and secondary, a real escalation policy, and five runbooks for the alerts that actually fire.

Set resource requests and limits per namespace

+$2,000/MO

Platform · 1 week

Two-week usage baseline, then a LimitRange per namespace. Let the autoscaler shrink the pools.

Tier logs and drop debug ingestion

+$1,900/MO

Platform · 3 days

Debug goes to a 7-day tier, or nowhere. Production keeps what an incident actually needs.

05 Days 61–90

Enable point-in-time recovery and volume snapshots

COMPLIANCE

Platform · 3 days

35-day retention. This is the control, not the backup job — the evidence is the drill.

Run the first timed restore drill

COMPLIANCE

Platform · 1 day

Restore the database and one volume, record the duration, and file it as the A1.2 evidence.

Replace the shared cluster-admin credential

COMPLIANCE

Platform · 3 days

Per-person, time-bound access through the IdP, with session logging. Rotate and retire the shared key.

Commit the stable baseline to a savings plan

+$6,800/MO

Platform · 1 week

Only after the deletions, so the commitment covers what actually remains.

06 What we would not do yet

SCOPE DISCIPLINE

Three things are deliberately absent. A cluster upgrade, because it changes nothing in this report and would consume the first month. A service mesh, because the current ingress works and the operational cost is not yet justified. And a multi-region active-active deployment, because 99.9% does not require it — the SLO conversation should come before the architecture.

Saying no to those three is most of why this plan fits in 90 days.

Arcline

November 6, 2026

SAMPLE DELIVERABLE

Audit Evidence Pack: SOC 2 Technical Controls

Ten controls, the evidence behind each one, and the two gaps that would fail an audit today.

SAMPLE Fictional demonstration data · Prepared by Arcline

PREPARED FOR

Northwind Payments

Fintech / payments (PCI DSS scope) · 28 engineers · 3 Kubernetes clusters · Azure and AWS

Framework: SOC 2 Type II (technical controls) · Evidence assembled from the platform

01 Control status

5	3	2
CONTROLS MET	PARTIAL	GAPS

This is the evidence pack for one framework's technical controls, assembled from the platform rather than from a policy document. Each row names the control, the evidence that satisfies it, and the owner. The point is that an auditor can be handed this table and the exports behind it, without a reconstruction project.

Five controls are met and evidenced automatically. Three are partial, which in practice means the control exists but the evidence is manual or incomplete. Two are genuine gaps, and both are the same finding an auditor reaches first: privileged actions are not logged, and recovery has never been tested.

02 Control evidence

CONTROL	REF	STATUS	EVIDENCE	OWNER
Access is provisioned from the identity provider	CC6.1	MET	SCIM group to RBAC mapping exported nightly; join table retained 12 months	Platform
Access is removed within one business day of departure	CC6.1 / CC6.6	PARTIAL	Offboarding script runs on a schedule; 90-day average is 2.7 days	IT + Platform
Production changes are reviewed before they merge	CC7.1 / PCI 6.5	MET	Branch protection requires one approval; 100% of merges in the last 90 days have one	Platform
Every production change is traceable to a deploy	CC7.1	MET	Argo CD application history joined to commit SHA; retained with the cluster	Platform
Privileged actions are logged	CC7.2	GAP	Kubernetes audit policy is unset; control-plane logs expire in 3 days	Platform
Logs are retained for 12 months	CC7.2	PARTIAL	Application logs 90 days; audit logs not collected at all	Platform
Data is encrypted in transit	CC6.7	MET	TLS 1.2+ enforced at ingress and between services; cipher policy in the ingress config	Platform
Data is encrypted at rest, with managed keys	CC6.7	MET	Provider-managed keys on Postgres and disks; key rotation record exported quarterly	Platform
The platform restores within the documented RTO	A1.2	GAP	No restore has ever been performed; retention is 7 days and no RTO is documented	Platform

CONTROL	REF	STATUS	EVIDENCE	OWNER
Incidents are reviewed and actions tracked	CC7.3 / CC7.4	PARTIAL	Postmortems exist for 6 of 11 incidents this year; actions are not tracked to closure	Platform

03 The two gaps that would fail an audit

Privileged actions are not logged. Without a Kubernetes audit policy, the platform cannot show who ran a command in production, who changed a role binding, or who read a secret. Every other access control depends on this record, so a single missing audit log undermines the four controls above it that currently pass.

Recovery has never been tested. A backup policy is not evidence that the platform can restore. Auditors increasingly ask for the date, duration, and result of the last drill, and “we have backups” is not an answer. Both gaps close in under a week of engineering, which is why they are first in the remediation plan.

04 What good evidence looks like

The five passing controls have one thing in common: the evidence is produced by the platform, on a schedule, without anyone assembling it. The group-to-role mapping is exported nightly. The merge approval count comes from the Git host. The deploy history comes from Argo CD.

That is the difference between an audit that takes an afternoon and one that takes six weeks. It is also why the work is worth doing beyond the certificate: a platform that can produce this table on demand is a platform where access is reviewed, changes are traceable, and recovery has been proven.