

CashLine ERP

Hosting And Cost Strategy

Cloudflare Pages / DNS / WAF -> Azure App Service or Azure Container Apps -> Azure SQL Database -> Azure Blob Storage -> Azure Monitor

Document Field	Value
Version	1.0 - Branded Cloudflare Share Pack
Document Date	2026-04-20
Formats	Word, HTML, PDF
Lean MVP Estimate	\$80 - \$250/month
Production-Comfortable MVP Estimate	\$250 - \$600/month

CashLine ERP Hosting And Cost Strategy

Document date: 2026-04-20

Document status: Controlled production MVP hosting recommendation

Prepared for: Business owners, product owners, implementation leads, engineering, infrastructure, security, operations, finance, and release governance

Recommended hosting path:

Cloudflare Pages / DNS / WAF -> Azure App Service or Azure Container Apps -> Azure SQL Database -> Azure Blob Storage -> Azure Monitor

1. Executive Summary

CashLine ERP V1 should use a cost-efficient but production-credible hosting model. The recommended MVP strategy is a hybrid Cloudflare and Azure setup. This combines low-cost global frontend delivery through Cloudflare with enterprise-grade backend, database, storage, monitoring, and governance capabilities through Azure.

This approach is better suited to CashLine than a purely hobby-style hosting stack because the platform includes ERP workflows, user governance, role-based access, audit trails, reporting, document handling, desktop parity, and future finance/compliance requirements.

The recommended model is:

Cloudflare Pages / DNS / WAF -> Azure App Service or Azure Container Apps -> Azure SQL Database -> Azure Blob Storage -> Azure Monitor

2. Recommended MVP Hosting Stack

Layer	Recommended Option	Purpose	Why Recommended
Web frontend	Cloudflare Pages	Host public portal, authenticated shell, static assets, and branded documentation pages	Very low cost, global CDN, automatic HTTPS, fast deployment, strong static hosting
DNS, CDN, WAF	Cloudflare	Domain, SSL, caching, edge protection, basic WAF and traffic control	Cost-efficient protection and performance at the edge
Backend API	Azure App Service or Azure Container Apps	Host backend APIs, workflow endpoints, auth/session logic, reporting APIs, and business services	Stronger enterprise fit for .NET/API workloads, managed runtime, staging support, scaling path
Database	Azure SQL Database	Production relational database for users, roles, workflows, transactions, audit, reporting, and configuration	Best fit for the current SQL Server-aligned production path
Documents/files	Azure Blob Storage	Store KYC documents, generated exports, report files, and operational attachments	Reliable object storage with access controls and lifecycle management
Monitoring	Azure Application Insights and Azure Monitor	Logs, traces, exceptions, performance, alerting, and production diagnostics	Required for MVP hypercare and production readiness
Secrets/configuration	Azure App Configuration or Key Vault	Store secrets, credentials, connection strings, and environment settings	Better security than hardcoded or manually managed secrets
Backup/recovery	Azure SQL backup and storage backup policies	Point-in-time restore, backup evidence, and rollback support	Required for controlled production deployment

3. Recommended Environment Layout

Environment	Hosting Approach	Data Type	Cost Strategy
Development	Local development plus optional low-cost Cloudflare/Azure preview	Synthetic development data	Keep minimal and avoid always-on cloud resources where possible

Environment	Hosting Approach	Data Type	Cost Strategy
QA/System Test	Small backend instance and small database	Controlled test data	Use low-tier resources and scheduled shutdown where possible
Staging/Pre-production	Same architecture as production, smaller size	Sanitized or approved test data	Mirror production behavior while keeping compute/database smaller
Production MVP	Cloudflare + Azure backend + Azure SQL + Blob Storage + monitoring	Real production data	Start modestly, monitor, then scale based on actual load
Wider rollout	Same architecture with scaled compute/database and stronger resilience	Real production data	Upgrade after MVP stability and usage patterns are proven

4. Recommended Target Architecture

Component	MVP Recommendation	Later Scale Recommendation
Frontend hosting	Cloudflare Pages	Cloudflare Pages with stricter WAF, caching rules, custom access policies, and deployment approvals
Backend hosting	Azure App Service Basic/Standard or Azure Container Apps low tier	App Service Premium or Container Apps autoscaling with private networking
Database	Azure SQL Basic/S0/serverless or low vCore tier	Higher Azure SQL tier, zone redundancy, longer backup retention, read replicas or reporting replica if needed
Storage	Azure Blob Storage hot tier	Lifecycle rules, archive tier, private endpoints, retention policy
Monitoring	Application Insights with controlled retention	Centralized logging, dashboards, alert rules, log export, SIEM integration
Security	Cloudflare SSL/WAF + Azure secrets	Private networking, Key Vault, managed identity, advanced WAF, DDoS plan if needed
Deployment	Manual-approved pipeline	Full CI/CD with approvals, rollback automation, environment promotion gates

5. Cost Estimation

The following estimates are directional planning ranges, not vendor quotes. Final cost depends on selected region, traffic, database size, data retention, support plan, storage volume, log volume, and integration usage.

5.1 Lean Controlled MVP Estimate

Cost Item	Recommended Entry Setup	Estimated Monthly Range
Cloudflare Pages / DNS / SSL	Free or Pro depending team needs	\$0 - \$25
Azure backend API	Low-tier App Service or small Container Apps setup	\$15 - \$80
Azure SQL Database	Basic/S0/serverless or low-tier SQL setup	\$10 - \$120
Azure Blob Storage	Low-volume documents and exports	\$2 - \$25
Azure Monitor / Application Insights	Controlled retention and sampling	\$5 - \$40
Backups and recovery storage	Built-in plus minimal retained exports	\$5 - \$30
Miscellaneous bandwidth/configuration	Low MVP traffic	\$5 - \$30
Estimated Total	Lean MVP	\$80 - \$250/month

5.2 Production-Comfortable MVP Estimate

Cost Item	Recommended Production-Comfortable Setup	Estimated Monthly Range
Cloudflare Pro or Business consideration	Pro for MVP; Business if stronger edge support is required	\$20 - \$250

Cost Item	Recommended Production-Comfortable Setup	Estimated Monthly Range
Azure backend API	Standard App Service or more stable Container Apps allocation	\$75 - \$200
Azure SQL Database	Standard/serverless/vCore with better performance headroom	\$75 - \$250
Azure Blob Storage	Documents, exports, backup artifacts, retention	\$10 - \$60
Azure Monitor / Application Insights	Dashboards, alerts, retention, log search	\$20 - \$100
Backups and recovery	More structured backup retention and restore testing	\$20 - \$75
Miscellaneous bandwidth/configuration	Moderate controlled production traffic	\$20 - \$75
Estimated Total	Production-comfortable MVP	\$250 - \$600/month

5.3 Excluded Costs

The estimate excludes:

- SMS provider charges.
- Email provider charges if paid.
- Payment provider fees.
- External credit bureau or iScore-related costs.
- CRIF/D&B reports or APIs.
- Lender or banking integration fees.
- Paid support plans.
- Security penetration testing.
- Domain purchase if not already owned.
- Production Oracle or enterprise database licensing.
- Enterprise SIEM or compliance tooling.
- High-availability multi-region architecture.

6. Cost-Control Recommendations

Cost Control	Recommendation
Start small	Begin with controlled MVP tiers and scale only after real usage data
Separate environments	Keep QA and staging smaller than production
Disable unnecessary always-on resources	Use low-tier or scheduled resources for non-production where possible
Control logs	Use sampling, retention limits, and log-volume alerts
Use Cloudflare for static delivery	Avoid serving static files from the backend API
Keep document storage in object storage	Do not store uploaded documents directly in the database
Set budget alerts	Configure monthly budget thresholds from day one
Review monthly	Reassess cost after the first production month and after wider rollout
Avoid premature high availability	Do not pay for enterprise redundancy before MVP usage justifies it
Avoid direct production experiments	Use staging for load, migration, and rollback rehearsals

7. Platform Comparison

Hosting Option	Strength	Weakness	CashLine Recommendation
Cloudflare Pages + Azure	Strong cost/reliability balance, enterprise growth path, SQL Server fit	Requires managing two platforms	Recommended for MVP production
Full Azure	Strong enterprise governance and single-vendor operations	Higher cost than hybrid frontend approach	Good later if operations prefer one cloud
Cloudflare Pages + Railway	Very fast and cheap for demos	Less enterprise governance and database maturity for finance workflows	Acceptable for demo, not preferred for production MVP
Cloudflare Pages + Render	Simple app deployment	Less aligned if SQL Server is mandatory	Possible for prototype/pilot only
DigitalOcean App Platform	Predictable pricing and simple operations	Better fit for Postgres/MySQL than SQL Server	Good for non-SQL Server MVPs, less ideal for CashLine
Supabase/Neon	Excellent Postgres-first platforms	Requires database direction change from SQL Server	Not recommended unless CashLine moves to Postgres
AWS	Very strong and scalable	More setup complexity and DevOps overhead	Good only if team already has AWS capability

8. Reliability Recommendations

Area	MVP Control	Wider Rollout Control
Uptime	Use managed Azure hosting and Cloudflare frontend	Add higher App Service/Container tier and stronger SLA/support
Database recovery	Azure SQL point-in-time restore	Longer retention, tested restores, optional replicas
Backups	Pre-deployment backups and regular backup checks	Formal backup policy and restore drills
Deployment safety	Staging rehearsal and production smoke test	Automated CI/CD gates and rollback automation
Monitoring	Application Insights, health checks, log alerts	Dashboards, SLA reports, incident workflow
Security	HTTPS, WAF, role-based access, secret management	Key Vault, private endpoints, advanced WAF, SIEM
Performance	Start modest and measure real use	Scale compute/database from metrics
Desktop parity	Desktop points to the same controlled backend/API	Desktop update channel and version control

9. Production Versus Backup/Recovery Separation

Production and backup/recovery must be treated as two different operating concerns. Production is the active live environment used by real users to process business transactions. Backup/recovery is the protected fallback capability used to restore service, recover data, support rollback, and prove business continuity.

Backup/recovery must not be considered another production environment and must not be used for normal user activity, testing, reporting experiments, or operational processing.

9.1 Production Environment

Production Area	Purpose	Expected Behavior
Production frontend	Serves the live CashLine web portal to approved users	Publicly reachable through approved domain, Cloudflare, HTTPS, and WAF rules
Production backend API	Processes live user actions, workflows, permissions, reports, notifications, and integrations	Always points to production database and production configuration
Production database	Stores live operational records, users, roles, transactions, workflow states, audit records, and reportable data	Receives live writes only from approved production services
Production document storage	Stores live uploaded documents, generated exports, and operational attachments	Governed by production access rules and retention policy
Production monitoring	Tracks live health, errors, latency, security events, and user-impacting incidents	Used for hypercare, alerting, and operational support

Production must contain only approved live configuration and approved live data. It must exclude UAT users, demo users, sample journeys, test records, placeholder transactions, and non-operational guidance content.

9.2 Backup And Recovery Capability

Backup/Recovery Area	Purpose	Expected Behavior
Database point-in-time restore	Recover production database to a known previous point	Used only during approved recovery, rollback, or restore testing
Pre-release backup	Capture production state immediately before deployment	Mandatory before production migration or production release
Blob/document backup controls	Protect uploaded documents, exports, and attachments from accidental deletion or corruption	Uses soft-delete, versioning, retention, or backup policy where feasible
Release package archive	Preserve deployed backend, frontend, desktop, and configuration package	Enables rollback to last known good release
Configuration backup	Preserve production environment variables, secrets references, and integration settings	Supports environment rebuild and incident recovery
Restore drill evidence	Prove that backups can actually be restored	Required before go-live and repeated periodically

Backup/recovery data should be isolated from active production processing. A restored copy may be used for controlled validation or incident recovery, but it must not silently become the live production environment without formal cutover approval.

9.3 Separation Rules

Rule	Requirement
No shared test data	Backup/recovery copies must not be mixed with QA, staging, or demo data
No direct user access	Business users should not log into backup/recovery stores or restored copies unless a formal recovery event is declared
No routine reporting from backups	Normal operational reports must come from production or an approved reporting replica, not from backup stores
No backup overwrite	Production backups must be protected from accidental overwrite or deletion
Restore validation required	A backup is not considered reliable until a restore test proves it can be used
Approval before restore	Production restore or rollback requires release owner and technical owner approval
Evidence retained	Backup creation, restore tests, and recovery actions must be recorded in the release/governance evidence pack

9.4 MVP Backup Baseline

Backup Item	MVP Recommendation	Owner
Azure SQL automatic backups	Enable point-in-time restore with at least 7 days minimum retention, preferably up to 35 days if budget allows	Technical lead / DBA
Pre-deployment database backup	Take before every production deployment or migration	Release team / DBA
Document storage protection	Enable soft-delete/versioning or equivalent retention for production Blob Storage	Technical lead
Release package archive	Keep every deployed backend, frontend, desktop, and Cloudflare package	Release owner
Configuration export	Record production configuration baseline before release	Technical lead
Restore drill	Run before go-live, then monthly during MVP stabilization	QA + technical lead
RPO target	MVP baseline: 24 hours or better unless live finance data requires tighter recovery	Business + technical owners

Backup Item	MVP Recommendation	Owner
RTO target	MVP baseline: same business day recovery unless business requires tighter SLA	Business + technical owners

9.5 Cost Distinction

Production cost covers the active runtime used by users. Backup/recovery cost covers the safety layer that protects production.

Cost Category	Examples	How To Treat It
Production cost	App Service/Container Apps, Azure SQL live database, Cloudflare, active Blob Storage, monitoring	Core monthly operating cost
Backup/recovery cost	Backup retention, backup storage, release archives, restore-test database copies, log retention	Business continuity and risk-control cost
Staging cost	Staging backend, staging database, staging storage	Pre-production validation cost, separate from production and backup

The MVP cost estimate includes a modest backup/recovery allowance, but a stricter backup plan, longer retention, replicated storage, or disaster recovery environment will increase monthly cost.

10. Recommended Deployment Model

The recommended MVP hosting and deployment model should follow the production release path:

Development -> QA -> Staging/Pre-production -> Controlled Production MVP -> Stabilization -> Wider rollout

Each stage should have its own configuration and database. Production should never share a database with QA, staging, demo, or development.

11. Minimum Production Setup

The minimum acceptable production setup for CashLine V1 MVP should include:

- Cloudflare-managed domain and HTTPS.
- Cloudflare Pages frontend hosting.
- Azure backend hosting.
- Azure SQL production database.
- Separate Azure SQL staging database.
- Azure Blob Storage for files and exports.
- Application Insights logging.
- Health check endpoint.
- Budget alerts.
- Pre-deployment database backup.
- Role-based production users only.
- No UAT/demo data.
- No public debug endpoints.
- No hardcoded production secrets.

12. Recommended First-Month Operating Plan

Week	Focus
Week 1	Deploy staging, run migration rehearsal, load test data, verify web/desktop/API/reporting
Week 2	Deploy production MVP, activate limited users, monitor login/workflows/reports/notifications
Week 3	Review defects, tune database/API, validate backup/restore evidence, refine alert thresholds
Week 4	Cost review, usage review, security review, readiness decision for wider rollout

13. External Integration Hosting And Go-Live Uncertainty

ETA/e-invoicing, Fawry, and Paymob are go-live uncertainty paths for the MVP hosting and deployment model. They are not assumed active in the baseline production hosting cost or runtime setup unless third-party credentials, authority/merchant access, endpoint configuration, callback rules, security configuration, and owner sign-off become available.

These paths may be added to the controlled MVP at any time if ETA, Fawry, or Paymob activate the required credentials or access earlier than expected. If that happens, the hosting design must support controlled activation without weakening production governance.

External Path	Hosting/Infrastructure Impact	Activation Control
ETA / e-invoicing	May require outbound API access, certificate/credential handling, secure configuration, tax/audit logging, and exception monitoring	Enable only after staging validation, secure secret storage, tax/finance approval, and production release approval
Fawry	May require payment API access, callback endpoints, webhook security, reconciliation logs, and monitoring alerts	Enable only after payment-flow testing, callback validation, reconciliation check, and production release approval
Paymob	May require payment API access, callback endpoints, webhook security, reconciliation logs, and monitoring alerts	Enable only after payment-flow testing, callback validation, reconciliation check, and production release approval

The MVP budget estimate does not include external provider fees, transaction fees, paid certification fees, production support costs from those providers, or additional hosting/security cost that may arise if these integrations are activated. Any activation must be reviewed as a cost-impacting and release-impacting change.

14. Final Recommendation

CashLine should proceed with a hybrid Cloudflare and Azure hosting model for the controlled production MVP.

The recommended target stack is:

Cloudflare Pages / DNS / WAF -> Azure App Service or Azure Container Apps -> Azure SQL Database -> Azure Blob Storage -> Azure Monitor

This gives the project the best balance of cost efficiency, reliability, SQL Server alignment, enterprise credibility, operational monitoring, and future scalability.

For budget planning, use:

- Lean controlled MVP estimate: \$80 - \$250/month
- Production-comfortable MVP estimate: \$250 - \$600/month

The business should start lean, monitor carefully, avoid premature enterprise spend, and scale only after controlled production usage proves real workload patterns.