DevCorp

MANAGED HOSTING

HIGH-AVAILABILITY CLOUD PLATFORM

Managed Hosting: Enterprise

A platform architecture for mission-critical services, higher availability goals, and formal 24/7 operations.

Enterprise replaces the single-server model with a load-balanced, orchestrated platform. It is designed around multiple application instances, stronger network controls, centralized observability, managed data services, documented recovery targets, and 24/7 incident response.

Service model

Managed hosting

Package

Enterprise

Document

Version 1.0 - 2025

Package Overview

Enterprise replaces the single-server model with a load-balanced, orchestrated platform. It is designed around multiple application instances, stronger network controls, centralized observability, managed data services, documented recovery targets, and 24/7 incident response.

Best fit

- Mission-critical applications with formal uptime expectations
- Higher transaction volume, stronger brand risk, or compliance pressure
- Teams that require disaster recovery, 24/7 escalation, and change management

Plain-English value

- The infrastructure is monitored and maintained by DevCorp.
- Operational responsibilities are defined before production use.
- Technical controls can be expanded as risk, traffic, or compliance needs grow.

Resilient by design

Multiple application instances and load balancing reduce dependency on any single server.

Enterprise operations

Monitoring, incident process, access control, and reporting are designed for accountable support.

Scalable foundation

The platform can grow through horizontal scaling, managed databases, queues, and regional expansion.

Reference Architecture

The architecture is chosen to match the operational risk of the package: simple where simplicity is safer, layered where availability and control matter more.

Edge

CDN, WAF, DNS and global traffic controls

>

Platform

Load balancers, Kubernetes and worker pools

>

Data

Managed database, backups and DR design

Technical foundation

- Kubernetes or managed container platform
- Load-balanced API and frontend delivery
- Managed database or highly available database cluster
- Separate worker pools for queues and background jobs
- CDN/WAF in front of public traffic
- Disaster recovery design with documented RTO and RPO

Compute Platform

We choose the compute platform based on compatibility, stability, and cost/performance. AMD/x86 is usually the safest default; ARM is considered when dependencies and container images are validated.

Platform

AMD/x86 dedicated or performance-class nodes for predictable workloads; ARM nodes may be used for compatible stateless services.

Sizing

Multiple nodes across failure domains. Capacity includes production load, failover headroom, rollouts, and observability agents.

Scaling

Horizontal pod autoscaling, node pool expansion, separate worker pools, and managed database scaling.

AMD / x86 strengths

- Best default for PHP, Node.js, MySQL, Docker images, and broad package compatibility.
- Predictable choice for production systems with third-party binaries or legacy dependencies.
- Dedicated or performance-class AMD nodes are preferred for sustained production load.

ARM strengths

- Good cost/performance for compatible stateless services, workers, and staging systems.
- Requires image and dependency validation before production use.
- Usually introduced after the baseline environment is stable and measurable.

Security and Firewall Layers

Security is built in layers: public traffic controls, server firewalls, restricted operator access, secret handling, patching, and optional WAF or VPN controls.

Firewall and access controls

- Edge WAF/CDN with rate limiting, bot controls, and DDoS mitigation
- Cloud firewalls and security groups per subnet or node pool
- Network policies between application, workers, database, and observability
- Private database access; public database exposure avoided
- Bastion, VPN, or SSO-backed administrative access
- Central secrets management and least-privilege service accounts

Security baseline

- Least-privilege access for operators and deployment paths
- Secrets stored outside source control
- TLS certificates managed and monitored
- Security updates handled through planned maintenance
- Suspicious login and brute-force activity monitored where applicable

Monitoring, Backup, Operations

A managed service is only useful when it can be observed, recovered, and operated consistently. This package defines what is watched, what is backed up, and how routine operations are handled.

Monitoring

- Platform, node, pod, ingress, and database monitoring
- Synthetic checks from multiple regions
- Centralized logs with retention policies
- SLO dashboards and alert routing
- Incident dashboard and post-incident reporting
- Capacity planning reviews

Backup and recovery

- Managed database backups or replicated backup strategy
- Cross-region backup retention
- Infrastructure-as-code state and cluster configuration backup
- Regular restore or disaster recovery exercise
- Documented RTO/RPO and escalation workflow

Operational support

- 24/7 critical incident support
- Defined escalation path and named operational contacts
- Change management for major releases
- Regular architecture and security reviews
- Post-incident reports for major incidents

Support terms in brief

- 24/7 coverage applies to agreed critical incident categories and escalation paths.
- Formal response targets, service credits, and availability commitments are defined in the service agreement.
- Major incidents include follow-up reporting and improvement actions.

Add-ons and Boundaries

The package can be extended with add-ons. Boundaries are intentionally explicit so customers understand which risks are covered by the selected service level.

Available add-ons

- Active-active multi-region architecture
- Managed SIEM integration
- Dedicated status page and customer communication process
- Annual disaster recovery exercise
- Penetration test coordination
- Compliance documentation package

Important boundaries

- Requires a separate platform implementation phase
- Higher operational complexity than VM-based hosting
- Application may need changes for active-active or advanced failover
- Formal support commitments require agreed scope and responsibilities

Not included by default

- Compliance certification, legal review, and penetration testing are separate workstreams unless included in scope.
- Customer-owned third-party platforms remain under the customer's commercial account control.
- Application changes required for active-active or advanced failover are scoped separately.

Shared responsibility

- DevCorp manages the hosting platform, monitoring, backups, and agreed operational processes.
- The customer remains responsible for business content, third-party account ownership, and timely approval of changes.
- Final support windows, response targets, and legal commitments are defined in the service agreement.

Typical onboarding path

- Confirm application architecture, domains, DNS, secrets, integrations, and expected traffic.
- Select compute platform, region, backup target, firewall model, and monitoring scope.
- Deploy the environment, run smoke checks, validate backups, and document access.
- Agree support contacts, maintenance windows, incident priorities, and escalation path.