KORE

PRODUCT BRIEF / 02 / CONTAINERS

Managed Kubernetes (CaaS)

Run Kubernetes without turning every development team into a platform team.

A governed Kubernetes platform covering cluster architecture, worker capacity, networking, persistent storage, observability and lifecycle support.

THE KORE SERVICE PATH

TECHNICAL FOUNDATION

KUBERNETES	WORKER POOLS	POLICY
Governed cluster foundation	Right-sized application capacity	Access and lifecycle controls

PLATFORM	DELIVERY	OWNERSHIP
Governed cluster foundation	Repeatable release path	Clear platform boundary

ENTERPRISE PERFORMANCE. EXECUTIVE SIMPLICITY.

sales@korecs.net / korecs.net

THE BUSINESS CASE

A governed platform without building a platform team

Your teams keep control of applications and deployment intent; KORE operates the infrastructure and platform scope defined in the service model.

01 Focus on applications

Developers own code and releases while KORE operates the platform foundation.

02 Standardise delivery

Namespaces, policies, storage and observability follow an agreed platform pattern.

03 Govern the boundary

Clear responsibilities separate platform operation from application ownership.

WHAT THE SERVICE PROVIDES

Cluster architecture

Topology, availability and the management layer designed for the workload.

Worker pools

Defined compute pools for general or specialist workload profiles.

Networking and ingress

Ingress, DNS, service networking and policy built into the design.

Persistent data

Block or file storage integration with protection options for stateful services.

Observability and lifecycle

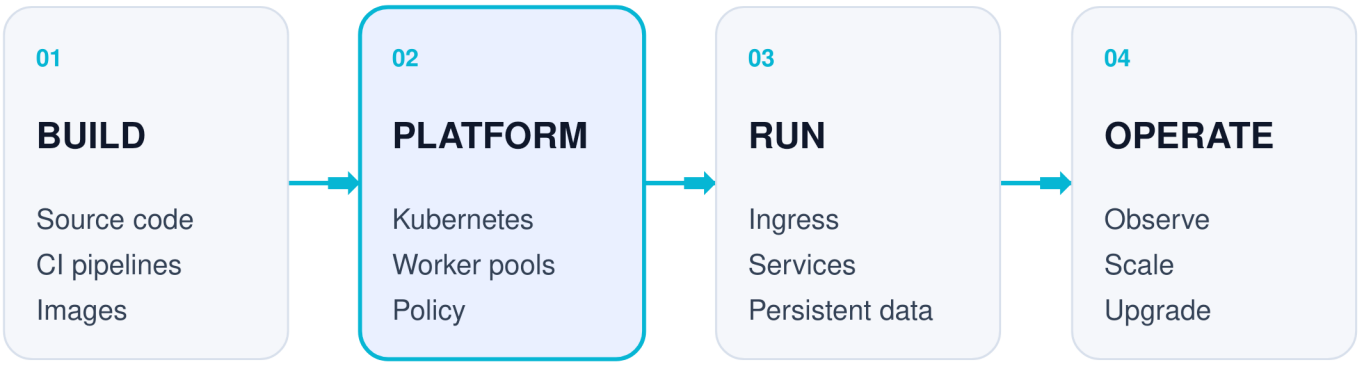
Metrics, logs, alerting, version planning and controlled maintenance.

BEST SUITED TO

- SaaS platforms, APIs and microservices
- Teams needing repeatable environments
- Standardised delivery on Kubernetes
- Stateful services with clear platform boundaries

SERVICE BLUEPRINT

Applications reach production through a governed platform



- IDENTITY & RBAC
- NETWORK CONTROLS
- OBSERVABILITY
- BACKUP OPTIONS

How the platform reaches production

01 Discover

Map application, delivery, data and availability needs.

02 Design

Agree topology, tenancy, networks and responsibilities.

03 Enable

Build, integrate delivery paths and validate workloads.

04 Operate

Monitor health, control change and plan versions.

KORE HANDLES

- Compute, storage and network foundation
- Contracted Kubernetes platform components
- Platform monitoring, maintenance and incidents

CUSTOMER RETAINS

- Application code, images and release quality
- Application secrets, data and vulnerabilities
- Scaling intent, testing and business acceptance

COMMERCIAL DESIGN

Run Kubernetes with clear ownership at every layer

Choose the platform boundary deliberately, then connect application delivery, persistent data and operations through one governed service design.

Complete the service

VMaaS or Bare Metal	Flexible virtual capacity or isolated hardware beneath the platform.
Managed Backup	Protection for persistent data and supporting infrastructure.
Connectivity + Operations	Private dependencies and engineering-led operational coverage.

BRING TO THE DESIGN WORKSHOP

01

Which workloads retain state or external dependencies?

02

Who owns each platform and application decision?

03

What upgrade and recovery behaviour can apps support?

Why KORE

PLATFORM OWNERSHIP

A visible operating boundary

Application and platform responsibilities are agreed before production depends on them.

LIFECYCLE DISCIPLINE

Upgrade paths designed early

Version planning, maintenance and observability are part of the service design.

INTEGRATED FOUNDATION

The stack works as one

Compute, network, storage and protection are engineered together around the workload.

Kubernetes capability without unmanaged platform debt.

sales@korecs.net / www.korecs.net/contact