



BOHIOSYSTEM

• SERVICE DATASHEET • BOHIOSYSTEM

Cloud & Platform

We design, migrate, and govern your cloud: reproducible infrastructure, costs under control, and a platform your teams can deploy on without help.

DOCUMENT

BS-SVC-05

VERSION

1.0

DATE

August 1, 2026

PREPARED BY

BohioSystem

BohioSystem

bohiosystem.dev • Confidential

THE SERVICE

The cloud does not cut costs by itself. It cuts costs when someone governs consumption and infrastructure stops being configured by hand.

We work across AWS, Azure, and Google Cloud. We design the cloud architecture, migrate existing workloads, and define everything as code, so any environment can be recreated without relying on anyone's memory.

The second half of the service is governance: who can create what, how much each thing costs, and how to keep the bill from growing unnoticed until the end of the month.

Multi

AWS · AZURE · GCP

IaCINFRASTRUCTURE AS
CODE**FinOps**

COST UNDER CONTROL

ReproducibleIDENTICAL
ENVIRONMENTS

01

What we solve

What we find most often when we take over an existing cloud account.

COMMON SITUATION	WHAT IT MEANS FOR THE BUSINESS
The bill keeps growing and nobody knows why	Resources still running that are no longer used, oversized, or duplicated across teams. Without tags, it is impossible to attribute spend to anyone.
Everything was configured by hand	The infrastructure exists only in the web console. Recreating the environment is an archaeology exercise, and there is no way to know what changed or who changed it.
Environments do not match	Production and testing differ in configuration, so what works in one fails in the other and nobody trusts the tests.
Overly broad permissions	Half a dozen accounts with full access because it was the quick option at the time. Any one of them can bring down production by mistake.
Half-finished migration	The easy part was moved and half remained in the in-house data center, so the organization pays for two infrastructures and the complexity of connecting them.
No plan for an availability zone outage	Everything lives in a single availability zone and it was never tested what happens if that zone fails.

02

What the service includes

01 Cloud architecture

Design of the account, network, compute, and storage structure based on actual load and available budget.
Account and subscription structure · Network design and segmentation · Service selection · High availability by layer · Sizing

02 Cloud migration

Moving existing workloads from an in-house data center or another provider, in phases and with a rollback path.

Inventory and dependencies · Per-workload strategy · Data migration · Temporary coexistence · Rollback plan · Controlled cutover

03 Infrastructure as code

All infrastructure defined in version-controlled files. No more manual, untracked changes.

Terraform or OpenTofu · Reusable modules · Parameterized environments · Protected remote state · Change review before apply

04 Governance and security

Rules that keep the cloud from becoming disorganized as more people use it.

Identities and least-privilege permissions · Preventive policies · Encryption and secrets management · Audit logging · Verifiable compliance

05 Cost control (FinOps)

Making spend visible, attributable, and predictable before the bill arrives.

Mandatory tagging · Attribution by area or project · Budgets and alerts · Correct sizing · Usage commitments and reserved capacity

06 Internal platform

Letting development teams deploy without filing a ticket or waiting on anyone, within safe guardrails.

Project templates · On-demand environments · Automatic quality gates · Guarded self-service · Operational documentation

03

How we work

In short stages, with something verifiable at the end of each one.

PHASE	WHAT HAPPENS	WHAT YOU KEEP
01 · Discovery	We understand the business, the current state and the real constraint.	Diagnosis and agreed scope.
02 · Design	We define the solution, its risks and its execution plan.	Architecture and milestone plan.
03 · Build	We implement in verifiable increments, not in a single delivery.	Functional progress per milestone.
04 · Go-live	Controlled deployment, testing and knowledge transfer.	Solution in production and team trained.
05 · Evolution	Operation, measurement and continuous improvement under a service agreement.	Periodic report and living roadmap.

04

What you receive

Infrastructure

- Infrastructure-as-code repository.
- Environments reproducible from scratch.
- Network and architecture diagram.
- Tagged resource inventory.

Costs

- Spend dashboard by area and project.
- Budgets with configured alerts.
- Savings-opportunity report.
- Monthly projection.

Governance

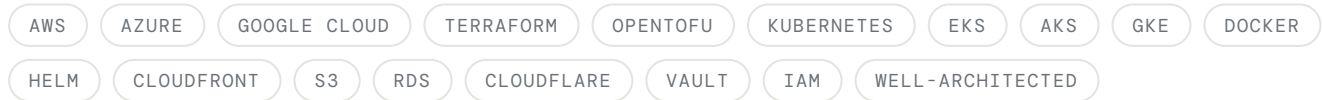
- Identity and permissions matrix.
- Applied security policies.
- Active audit log.
- Emergency access procedure.

Operations

- Runbooks for frequent tasks.
- Tested recovery plan.
- Monitoring and alerts.
- Handover sessions with the team.

05 Technologies we work with

We choose the tool for the problem, not by trend or by agreement with a vendor.



06 Engagement models

MODEL	HOW IT WORKS	WHEN IT FITS
Fixed project	Defined scope and deliverables, agreed price and timeline.	Bounded needs with a clear objective.
Monthly retainer	Reserved team capacity month to month.	Continuous evolution and sustained support.
Dedicated team	Profiles assigned exclusively under your direction.	Long programs or high workload.
Hour bank	Consumable hours with agreed priority.	Variable demand or one-off work.

07 Our commitments

COMMITMENT	HOW IT IS MET
No manual changes in production	Every change goes through reviewed code. Anything touched by hand gets lost, and nobody knows how to rebuild it.
Cost visible from day one	Tagging and spend attribution are set up at the start of the project, not when a surprise bill arrives.
Least privilege necessary	Nobody gets full access for convenience, including us. Access is documented and revoked when the engagement ends.
Tested recovery	A backup that has never been restored is not a backup. We run at least one real recovery test.
No unnecessary lock-in	We flag when a proprietary service ties us to a vendor and what the exit cost is, before adopting it.

ON MIGRATING "TO THE CLOUD" FOR ITS OWN SAKE

Moving a server as-is to a cloud virtual machine almost always costs more than leaving it where it was. Migration pays off when it takes advantage of what the cloud does differently: scaling on demand, turning off what is not in use, and delegating operation of common components. If the case does not justify it, we say so.

08

Next step

GET STARTED

A one-hour session, at no cost, to review your current situation and define together the minimum scope to start with.

CONTACT

contacto@bohiosystem.dev

WEB

bohiosystem.dev

DOCUMENT

BS-SVC-05 · v1.0