



BOHIOSYSTEM

● SERVICE DATASHEET · BOHIOSYSTEM

# DevOps & Operations

We turn deployment into routine: delivery automation, observability, and reliability practices that reduce the risk of every change.

---

DOCUMENT

BS-SVC-06

VERSION

1.0

DATE

August 1, 2026

PREPARED BY

BohioSystem

BohioSystem

bohiosystem.dev · Confidential

## THE SERVICE

If shipping to production is frightening, teams ship rarely. And when they ship rarely, every release is huge, which is why it fails.

We automate the path from the moment someone commits a change to the moment that change is running for users, with tests, controls, and the ability to roll back in minutes.

The second half of the service is knowing what is happening: metrics, logs, traces, and alerts that warn you before the client calls, not after.

**Daily**ACHIEVABLE DEPLOYMENT  
FREQUENCY**< 15 min**

ROLLBACK TIME

**Automated**

TESTS AND CONTROLS

**Traceable**

EVERY CHANGE AUDITED

01

**What we solve**

The symptoms of a delivery process that is not automated.

| COMMON SITUATION                           | WHAT IT MEANS FOR THE BUSINESS                                                                                                     |
|--------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------|
| Deploying is a middle-of-the-night event   | Releases happen at night, with several people online and a manual procedure written in a document that is almost never up to date. |
| Nobody knows what is running in production | There is no quick way to tell which version is deployed or what changes it includes, turning every incident into an investigation. |
| Issues surface through client complaints   | The first sign that something is broken is a phone call. There are no alerts, or so many that nobody looks at them.                |
| Rolling back is impossible                 | When a deployment goes wrong, the only way out is to fix forward under pressure, which is when serious mistakes get made.          |
| Works on my machine                        | Environments differ from one another, so problems appear only in production and cannot be reproduced to debug them.                |
| The knowledge lives in one person          | Only one person knows how to deploy. Their vacation is an operational risk to the business.                                        |

02

## What the service includes

### 01 Continuous integration and delivery

---

An automated, repeatable path from the repository to production, with controls at every step.

Pipelines per environment · Automated tests as a gate · Versioned artifact builds · Approvals where needed · Zero-downtime deployment

### 02 Observability

---

Knowing what is happening inside the system without guessing. Correlated metrics, logs, and traces.

Business and technical metrics · Centralized logging · Distributed tracing · Per-service dashboards · Alerts with threshold and context

### 03 Reliability (SRE)

---

Practices that keep the system standing: explicit service objectives and an error budget that orders priorities.

Service level objectives · Error budget · On-call and escalation · Root cause analysis · Load testing

### 04 Containers and orchestration

---

Consistent packaging across environments and orchestration when volume justifies it.

Optimized images · Kubernetes or simpler alternatives · Autoscaling · Configuration and secrets management · Health checks

### 05 Incident management

---

A clear procedure for when something breaks, agreed on before it happens.

Severity classification · On-call rotation · Channel and roles during the incident · Blameless post-mortem · Corrective action tracking

### 06 Delivery-chain security

---

Security controls built into the pipeline, not bolted on as an audit at the end.

Dependency analysis · Image scanning · Leaked-secret detection · Artifact signing · Component inventory

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

### Automation

- Documented, version-controlled pipelines.
- Consistent environments.
- Tested rollback procedure.
- Versioned, traceable artifacts.

### Operations

- Runbooks for frequent tasks.
- Incident management procedure.
- Escalation matrix.
- On-call guide.

### Observability

- Dashboards by service and by business metric.
- Alerts with defined recipient and threshold.
- Centralized, searchable logs.
- Agreed data retention.

### Internal capability

- Training for the client's team.
- Recorded handover sessions.
- Delivery architecture documentation.
- Support during the first deployments.

05

## Technologies we work with

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

GITHUB ACTIONS

GITLAB CI

JENKINS

ARGOCD

DOCKER

KUBERNETES

HELM

TERRAFORM

ANSIBLE

PROMETHEUS

GRAFANA

LOKI

OPENTELEMETRY

SENTRY

PAGERDUTY

SONARQUBE

TRIVY

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                                                                                                                        |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| Rollback always available   | No deployment is signed off unless a tested path back to the previous version exists.                                                |
| Alerts that someone acts on | Every alert has a recipient, a justified threshold, and an associated action. An alert that gets ignored is noise, and we remove it. |
| Blameless post-mortems      | After a serious incident, we analyze the system and the process, not the person. It is the only way mistakes get reported.           |

| COMMITMENT                               | HOW IT IS MET                                                                              |
|------------------------------------------|--------------------------------------------------------------------------------------------|
| <b>Automate what repeats</b>             | Any manual task performed more than three times gets automated or documented as a runbook. |
| <b>No dependency on external on-call</b> | We train the client's team so it can operate independently, even as we remain as backup.   |

#### ON ADOPTING KUBERNETES

Not every organization needs it. It pays off when there are many services, independent teams, and variable demand. Below that threshold it adds operational overhead that tends to outweigh the benefit, and a simpler alternative performs better. We evaluate it with numbers before proposing it.

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-06 · v1.0