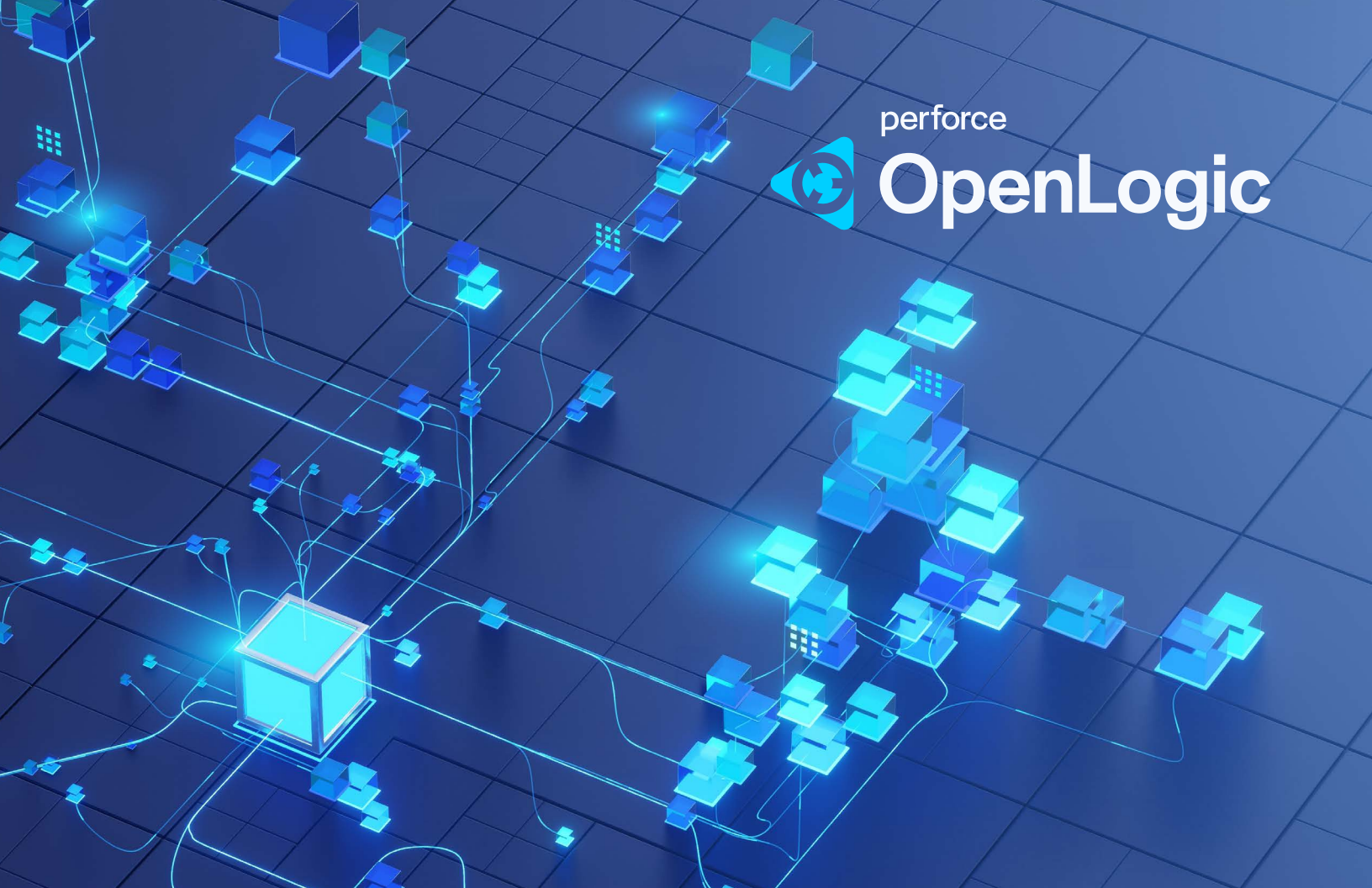




WHITE PAPER

Open Source Digital Autonomy Playbook

A Practical Framework for Reducing Dependence on Proprietary Managed Services

Executive Summary

Cloud computing has helped organizations innovate and scale, but heavy reliance on hyperscaler-managed services can limit flexibility when costs, regulations, markets, or business priorities change. Rather than abandoning the cloud, digital autonomy enables organizations to choose where workloads run and how critical services are operated, giving them more portability and control.

Open source technologies play a key role because they can run across public clouds, private infrastructure, and hybrid environments. This playbook outlines how to assess cloud dependencies and build a roadmap toward greater sovereignty and resilience.

Contents

3.....	Why Digital Autonomy Matters Now
3.....	The Hidden Trade-Off Behind Managed Services
4.....	From Cloud Migration to Architectural Portability
4.....	Visibility First: Building a Services and Dependency Inventory
5.....	Identifying Candidates for Decoupling
7.....	Kubernetes and the Portability Layer
7.....	The Operational Reality of Digital Autonomy
8.....	Building Internal Capability Without Building a New IT Department
9.....	A Practical Roadmap for Digital Autonomy
10	Digital Autonomy Readiness Assessment
12	Conclusion

Why Digital Autonomy Matters Now

Digital autonomy has become one of the most significant strategic conversations taking place within enterprise IT.

Historically, decisions about infrastructure placement were often driven by technical capabilities, cost considerations, and internal resource constraints. Today, those factors remain important, but they are no longer the only considerations shaping cloud strategy.

Across regulated industries, governments and private organizations are placing greater emphasis on where data resides, who controls critical infrastructure, and how technology providers influence operational risk. In Europe particularly, conversations around digital sovereignty have moved beyond policy discussions and are increasingly influencing real-world technology decisions. Organizations are examining whether critical systems should remain tightly coupled to infrastructure providers headquartered outside their region and whether they would be able to adapt quickly if regulations, geopolitical circumstances, or business priorities changed.

At the same time, cloud environments have grown substantially more complex. What began as virtual machines and storage has evolved into ecosystems of managed databases, messaging services, container platforms, analytics engines, monitoring tools, AI services, and data pipelines. Many organizations now rely on dozens of managed services that sit at the heart of business-critical applications. While these services reduce operational burden, they can also make workloads increasingly difficult to relocate or redesign.

For IT leaders, the question to ask is whether current architectures provide sufficient flexibility for future business needs.

Organizations pursuing digital autonomy are ultimately seeking four forms of control:

- Greater control over infrastructure decisions
- Greater control over data placement and governance

- Greater control over technology roadmaps
- Greater control over long-term operating costs

This does not necessarily mean eliminating cloud providers or abandoning hyperscaler platforms altogether. However, organizations should be looking for ways to ensure those relationships remain strategic choices rather than dependencies.

The Hidden Trade-Off Behind Managed Services

Managed services have become one of the defining features of modern cloud computing.

Whether an organization deploys PostgreSQL through Amazon RDS, runs Kubernetes on EKS, or relies on a managed messaging platform, the underlying value proposition is consistent: reduce operational complexity by outsourcing infrastructure management.

There is nothing inherently wrong with this approach. In many situations, managed services are the fastest and most efficient way to deploy new capabilities. Teams can move quickly, avoid staffing challenges, and rely on cloud providers to handle routine maintenance tasks such as monitoring, patching, backups, and upgrades.

However, convenience creates trade-offs.

Every managed service introduces a degree of coupling between the application and the provider operating that service. Sometimes that coupling is minimal. In other cases, it becomes deeply embedded in deployment processes, operational procedures, backup strategies, monitoring systems, and application architectures.

This challenge often becomes visible only when organizations attempt to move.

For example, an organization may discover that migrating a managed database involves much more than moving data. Backup processes must be recreated. Monitoring workflows need to be redesigned. Operational responsibilities that previously belonged to the cloud provider must now be assigned internally. Teams may require new skills, new tools, and new governance processes. What initially looked like a

straightforward migration effort quickly becomes an operational transformation initiative.

As organizations evaluate digital autonomy initiatives, it is important to recognize that the objective is not necessarily to eliminate managed services. Instead, the objective is to understand where dependencies exist, determine which ones introduce meaningful business risk, and evaluate whether alternative architectures might provide greater flexibility.

A useful starting exercise is to ask the following questions:

- Which managed services are most critical to our operations?
- What would happen if we needed to relocate these workloads within the next 12 months?
- Do we have sufficient visibility into how these services are being consumed?
- Would changing providers require substantial architectural changes?

The answers to these questions often reveal where an organization is most exposed to vendor dependency and where opportunities for greater portability may exist.

From Cloud Migration to Architectural Portability

One of the most important shifts organizations can make is changing how they think about migration.

Traditional migration projects typically focus on a destination. The objective is to move workloads from one environment to another as quickly and efficiently as possible. Success is measured by completing the migration while minimizing disruption.

The challenge with this approach is that it can unintentionally recreate the same dependency in a new location.

An organization may invest significant time and resources moving workloads from one cloud provider to another only to discover that the resulting architecture remains just as difficult to move in the future. The migration succeeds, but portability does

not improve.

Digital autonomy introduces a different objective.

Instead of asking how to move from Platform A to Platform B, organizations should ask how to create an architecture that can operate across multiple platforms with minimal change.

This distinction may appear subtle, but it has significant long-term implications.

When organizations prioritize portability, they begin making architecture decisions that reduce future migration effort. Workloads become less dependent on provider-specific services. Open source technologies become more attractive because they can be deployed across a variety of environments. Infrastructure components are selected based not only on functionality, but also on their ability to operate consistently across locations.

The ultimate goal is not to simplify the first migration. The goal is to ensure that future infrastructure decisions are no longer constrained by architecture choices made today.

Visibility First: Building a Services and Dependency Inventory

Many organizations begin their digital autonomy journey by focusing on technologies they want to adopt. In practice, a more effective starting point is understanding the technologies they already depend on.

Before developing a migration strategy, organizations need a clear inventory of the managed services supporting their business. This exercise extends beyond infrastructure teams. It requires understanding which applications rely on those services, how deeply they are integrated into existing workflows, and what operational responsibilities would need to be assumed if those services were self-managed.

Many leaders discover that they have excellent visibility into applications, but limited visibility into the underlying services those applications consume. Teams may know they are using PostgreSQL, Kubernetes, Elasticsearch, or object storage, but they may not fully understand how many applications rely on

those services or which provider-specific features have become embedded in day-to-day operations.

A useful inventory should answer four questions:

What managed services are currently in use?

Document every cloud service supporting production workloads, including:

- Databases
- Messaging platforms
- Search platforms
- Container orchestration platforms
- Monitoring services
- Backup services
- Object storage

Which teams use them?

Identify application owners and technical stakeholders responsible for each workload.

How critical are they?

Determine the business impact if the service becomes unavailable or requires migration.

What operational capabilities would need to be replaced?

Cloud platforms often provide additional capabilities beyond the service itself, including:

- Backup and recovery
- Monitoring
- Alerting
- Scaling
- Security updates

- Lifecycle management

Documenting these dependencies early prevents surprises later in the process.

Rather than producing a migration plan immediately, organizations should treat the inventory as a discovery exercise. The goal is to develop a realistic understanding of the current environment and establish a foundation for future decision-making.

Questions to Ask During the Inventory Process

- Which managed services support our most critical applications?
- Which apps would require code changes if services were replaced?
- Are teams using provider-specific features?
- What operational tasks are currently being handled by the cloud provider?
- Do we have sufficient expertise to operate these services internally if required?

The organizations that successfully move toward digital autonomy are rarely those with the most aggressive migration plans. They are the ones with the clearest understanding of their existing dependencies.

Identifying Candidates for Decoupling

Once an inventory is complete, the next step is determining where it makes sense to focus first.

Not every dependency requires immediate attention, and not every cloud service creates the same level of lock-in. Some services are built around widely adopted standards that exist across multiple providers. Others are tightly coupled to proprietary ecosystems and may become more difficult to replace over time.

The objective is not to replace everything at once. Instead, organizations should identify services where portability

improvements can deliver meaningful business value while maintaining an acceptable level of risk.

In many environments, the following categories emerge as strong candidates for evaluation:

Databases

Managed PostgreSQL and MySQL offerings are often logical starting points because organizations are already using open source technologies under the hood. The primary shift is moving responsibility for operation and support from the provider back to the organization.

Messaging Platforms

Services such as ActiveMQ and Kafka frequently support business-critical workflows. Because they sit at the center of application communication, they present both an opportunity for greater portability and a need for careful planning.

Search and Analytics Platforms

Organizations often discover that search, logging, and observability stacks have become deeply integrated into cloud environments. Evaluating alternatives can help reduce platform dependency while improving consistency across environments.

Container Platforms

Container orchestration services can serve as an important step toward portability because they create a consistent deployment model across infrastructure providers.

A useful prioritization framework evaluates workloads against four criteria:

Factor	Questions to Ask
Business Importance	How critical is the workload?
Migration Complexity	How difficult would migration be?
Lock-In Risk	How dependent is the workload on proprietary features?
Operational Readiness	Can the team support it post-migration?

The resulting roadmap will look different for every organization. A financial institution, global retailer, and healthcare provider may arrive at entirely different priorities despite using similar technologies.

Kubernetes and the Portability Layer

Few technologies have become as closely associated with digital autonomy as Kubernetes. However, many discussions overstate its role.

Kubernetes is not a digital autonomy strategy. Rather, it is an enabling technology that can support portability when used appropriately.

At its core, Kubernetes provides a standardized way to deploy and manage applications. Teams define desired system states using manifests and deployment configurations that can be applied consistently across environments. Because Kubernetes has become widely available across public clouds, private infrastructure, and managed offerings, it enables a common operational model regardless of where workloads ultimately run.

This consistency creates several benefits:

- Standardized application deployment processes
- Greater flexibility across cloud providers
- Improved disaster recovery options
- Reduced application-specific migration effort
- Consistency across development, test, and production environments

Perhaps most importantly, Kubernetes allows organizations to focus on application portability rather than infrastructure differences.

That said, Kubernetes should not be viewed as a complete solution. It does not automatically eliminate lock-in, nor does it remove the need for architecture decisions, governance processes, or operational expertise.

A common misconception is that adopting Kubernetes automatically delivers digital autonomy. In reality, organizations can still create provider dependencies through proprietary tooling, cloud-native integrations, or operational practices layered on top of Kubernetes.

For this reason, IT leaders should think of Kubernetes as a portability layer rather than a destination. The goal is not Kubernetes adoption for its own sake. The goal is enabling greater flexibility in where applications can operate in the future.

The Operational Reality of Digital Autonomy

Conversations about digital autonomy often focus on architecture. In practice, operations are where many organizations encounter the greatest challenges.

When cloud providers manage a service, they are often handling much more than infrastructure. They are also responsible for routine operational tasks that may not always be visible to application owners.

These responsibilities can include:

- Monitoring and alerting
- Backup management
- Disaster recovery
- Patch management
- Capacity planning
- Security updates
- Platform maintenance

As organizations pursue greater ownership of their environments, these responsibilities must be assigned somewhere else. That shift represents one of the most significant changes associated with digital autonomy.

Many organizations discover gaps during this phase.

For example, a team may be comfortable operating PostgreSQL but realize they lack a backup strategy. Another organization may successfully migrate workloads but discover that monitoring assumptions no longer apply. Others may find that support models designed around cloud-managed services no longer fit their new operating environment.

These challenges should not be viewed as reasons to avoid digital autonomy. Rather, they highlight why planning matters.

Successful autonomy initiatives account for operational ownership from the beginning.

Areas Requiring Special Attention

- **Monitoring and Observability:** Organizations need clear strategies for monitoring infrastructure health, application performance, and operational risk.
- **Security and Patching:** Ownership of software means ownership of software updates, vulnerability remediation, and lifecycle management.
- **Disaster Recovery:** Disaster recovery should be evaluated as part of architecture design rather than added later.
- **Governance:** Organizations need clear accountability for decision-making, support, and ongoing platform management.

The operational realities of autonomy are often where long-term success is determined.

Building Internal Capability Without Building a New IT Department

One of the most common misconceptions about digital autonomy is that organizations must dramatically expand their internal teams.

In reality, most organizations pursuing autonomy are not looking to become software vendors or infrastructure specialists. Their objective is simply to gain greater control over critical technology decisions.

As part of the planning process, IT leaders must determine how responsibilities should be distributed between internal teams, and third party partners, consultants, and support vendors.

Many organizations adopt a hybrid model: Internal teams continue to own application strategy and business outcomes, while external experts provide specialized platform knowledge, support, training, and advisory services.

This approach offers several advantages:

- Faster skills development
- Reduced hiring pressure
- Access to specialized expertise
- Lower operational risk
- Greater confidence during transitions

Organizations should evaluate their existing capabilities in several areas:

Technical Expertise

Do teams have practical experience operating open source infrastructure?

Process Maturity

Are incident response, patching, and operational procedures sufficiently mature?

Resource Availability

Can teams absorb additional responsibilities without disrupting current priorities?

Support Strategy

Who serves as the escalation path when complex issues arise?

Digital autonomy does not require organizations to operate in isolation. In most cases, success comes from combining internal ownership with external expertise in a way that aligns with business objectives and risk tolerance.

A Practical Roadmap for Digital Autonomy

Organizations rarely achieve meaningful portability through a single migration effort. Instead, they make incremental improvements that gradually reduce dependency while increasing operational flexibility.

A practical roadmap typically follows six phases.

Phase 1: Inventory

Create a comprehensive understanding of managed services, applications, dependencies, and operational ownership requirements.

Phase 2: Application Mapping

Identify relationships between applications and cloud services.

Determine where dependencies have become deeply embedded.

Phase 3: Architecture Design

Evaluate portability requirements and identify open source alternatives.

Establish operational and governance expectations.

Phase 4: Validation

Create parallel environments and evaluate functionality, performance, and operational readiness before migrating production systems.

Phase 5: Cutover

Transition workloads once confidence has been established and operational processes have been validated.

Phase 6: Repeat

Apply lessons learned to additional workloads and continue reducing dependencies over time.

This final phase is particularly important.

Digital autonomy develops through a series of decisions that gradually improve flexibility, resilience, and operational control. And organizations that approach it as an ongoing discipline tend to achieve the greatest long-term benefits.

Digital Autonomy Readiness Assessment

Use the following assessment to evaluate your organization's preparedness for a digital autonomy initiative.

Assign a score from 1 to 5 for each question:

1 = Strongly Disagree

3 = Neutral

5 = Strongly Agree

Architecture

Question	Score
We understand which cloud services support critical applications.	
We have documented application dependencies.	
We know where provider-specific features are being used.	
Applications can be moved without significant redesign.	

Operations

Question	Score
Monitoring responsibilities are clearly defined.	
Backup and disaster recovery processes are documented.	
We have processes for patching and lifecycle management.	
Operational ownership is clearly assigned.	

Governance

Question	Score
We have policies governing workload placement.	
Regulatory and sovereignty requirements are understood.	
Cloud decisions are evaluated against long-term strategic objectives.	
Platform decisions include portability considerations.	

Skills and Expertise

Question	Score
Teams have experience operating open source infrastructure.	
Internal expertise exists for core platforms and services.	
We have access to specialized support when needed.	
Teams can adopt additional responsibilities without significant disruption.	

Portability

Question	Score
We could relocate a critical workload within 12–18 months.	
Multiple deployment options are available for key applications.	
Disaster recovery environments are not tied to proprietary services.	
Future migrations would require less effort than previous migrations.	

Scoring Guide

20–40: Early Stage

Your organization is beginning to explore autonomy but still depends heavily on provider-managed capabilities.

41–70: Developing

Many foundational capabilities exist, but significant work remains before portability becomes a practical option.

71–90: Advanced

Your organization has established strong operational and architectural foundations for digital autonomy.

91–100: Highly Autonomous

Portability, operational ownership, and governance are embedded within technology decision-making processes, enabling greater flexibility and resilience over the long term.

Conclusion

Digital autonomy is about ensuring that no future infrastructure decision is constrained by architecture choices made in the past. Organizations that invest in portability today position themselves to respond more effectively to regulatory change, business growth, emerging technologies, and whatever comes next. Open source, if it can be properly maintained and governed, makes that agility possible.

Talk to OpenLogic about your autonomy plans and get a complimentary in-depth assessment of your environment:

Contact Us ►

openlogic.com/solutions/digital-autonomy-consultation