PLATFORM9

Cloud Lock-In: A Definitive Guide

Table of Contents

Introduction 3

No Easy Fixes 3

Defining Cloud Lock-In 4

Identifying Cloud Lock-In 4

Ways to Avoid Cloud Lock-In 6

The Causes of Cloud Lock-In 8

Strategies for Avoiding Cloud Lock-In 10

Anti-Lock Strategies: The Kubernetes Example 11

Conclusion 13

Introduction

In the world of cloud computing, lock-in is the elephant in the room. It's easy to sing the praises of the cloud, but more difficult to confront the real risks that lock-in poses for organizations that migrate to the cloud -- whether they use a public, private, hybrid or edge cloud architecture.

To be sure, not everyone is on the same page when it comes to recognizing cloud lock-in risks. Some proponents of cloud migration contend that lock-in threats are overstated. Others argue that, although lock-in risks are real, the effort required to avoid them is not worth it. Still others offer a simplistic take on the situation; they suggest, for example, that if you use open source platforms like OpenStack or Kubernetes to deploy cloud workloads, you are automatically protected against the danger of lock-in, which is not true.

This guide offers a different perspective in order to clarify what cloud lock-in risks entail and how to devise cloud migration and management strategies that will allow your organization to steer clear of them. In the pages that follow, we define the meaning of cloud lock-in -- a concept that is more complicated than it may first appear -- and offer examples of the many forms that cloud lock-in may take. We then present specific strategies that offer safeguards against lock-in.

No Easy Fixes

As we note below, there is no silver bullet or simple trick for avoiding cloud lock-in risks entirely. An anti-lock-in strategy requires a multi-pronged approach that combines the right cloud architectures, tooling and management strategies to minimize a team's risk of being unable to migrate from one cloud platform to another, or to extend its cloud architecture or services beyond those that it chooses during its initial cloud migration. And, of course, the strategy that you adopt to combat lock-in must reflect your organization's specific cloud usage patterns; there is no one-size-fits-all solution.

Whatever the reality of your lock-in risks, or which particular cloud platforms you choose to use, the tips in this guide will help empower you to build a cloud strategy that capitalizes fully on the agility, scalability and extensibility that cloud architectures promise, while also making full use of the array of different cloud services and solutions available today.

Defining Cloud Lock-In

Part of the reason there is such a diversity of opinion regarding cloud lock-in risks and whether there is actually a serious problem, is that simply defining and identifying lock-in can be challenging, especially within the context of the cloud.

At a high level, cloud lock-in may be defined as the inability to migrate or extend a cloud deployment in the ways that a team wishes. However, this definition is somewhat simplistic, because it implies that lock-in is a singular phenomenon or a problem that takes a consistent form.

The reality is more complex. Cloud lock-in can occur in many different ways. Some are more restrictive and problematic than others. It's possible to maintain a cloud strategy that is overall quite agile, while still being locked into certain cloud tools or services. Likewise, even organizations with very constrained ability to extend or migrate their cloud deployments, still typically have some flexibility, and in that sense do not face total lock-in.

What all of this means is that you should think of cloud lock-in as a multi-dimensional phenomenon. It comes in many forms and affects each organization -- indeed, each individual cloud workload -- differently. Your team or department will not confront lock-in in the same way as other teams within your organization, let alone other companies.

Identifying Cloud Lock-In

The diverse forms in which cloud lock-in may occur also means that identifying lock-in and understanding which ways it may threaten your organization, requires a multi-faceted analysis of your overall cloud strategy.

In this regard, the "shades" of lock-in defined by Gregor Hohpe on the website of Martin Fowler, a prominent DevOps thought leader, may be instructive. Writing about lock-in in general (as opposed to cloud lock-in in particular), Hohpe identifies multiple ways in which lock-in may occur. From the perspective of the cloud, the most significant lock-in shades include:

- **Vendor**: Perhaps the most obvious form of lock-in is that associated with a particular vendor. Vendors who offer solutions that integrate or are compatible only with other solutions offered by the same vendor lock you into their ecosystems. This is the most classic form of lock-in, dating back to the era of vendor monopoly in the PC industry in the 1990s. In more recent years, however, naked vendor lock-in has become less of a threat, as most vendors now invest more effort in building integrations with third-party solutions. Still, vendor promises of platform-agnosticism don't always live up to the reality.

- **Product**: You may be locked into using a particular product. Even if you can migrate from one vendor's version of that product to another's, you are still locked in if you lack the ability to migrate your workloads to a different product platform entirely. Kubernetes is a good example of a product that, despite being an open source platforms that is offered by multiple vendors, can nonetheless lock users in through vendor-specific add-ons, APIs or other functionality that may make it difficult to migrate Kubernetes configurations or applications from one vendor's implementation to another.

- **Version**: You could end up being dependent on a specific version of a product or platforms, and unable to upgrade when a new version appears. This could happen because you knowingly deploy legacy workloads or tooling that will no longer be supported in future product releases. Or, you may find yourself unwittingly locked into a product version because the product's vendor (or the open source project that develops it, if it's an open source platform) suddenly discontinues backwards-compatibility support for features or services that you depend on.

- **Architecture**: In the cloud especially, architectural lock-in presents a significant risk. You may deploy workloads using one configuration of cloud services, such as object storage and virtual machines, and find it difficult to reconfigure your workload in order to run on different services, such as cloud-based databases and serverless functions. Or, you may be locked into an entire cloud architecture; for example, you may deploy a hybrid cloud architecture at first, then find it difficult to migrate to an architecture that runs solely in the public cloud, due to incompatibility between hybrid cloud services and those running in the public cloud. You may also have to use certain APIs in order to use a particular type of cloud architecture, and those APIs could become a source of lock-in.

- **Support service**: Even if it is technically feasible to migrate from one cloud vendor, product, version or architecture to another, your organization may find itself unable to do so because it depends on support services that either will not support the migration, or will not be available after the migration is complete. Third-party cloud management services that work only with certain types of architectures or products may lead to this type of cloud lock-in. That is because, without the in-house resources required to forego third-party support and manage a migration yourself, you enjoy only as much cloud flexibility as the support service allows you.

- **Team and skillsets**: Likewise, if you manage cloud workloads using your own team, they may lock you into certain cloud patterns or services due to a lack of the skills necessary to migrate or extend to other platforms. You might have built your cloud strategy initially around a specific platform, like the AWS public cloud or a VMware-based hybrid or private cloud. As a result, your staff are well versed in these platforms, but they may lack the ability or willingness to learn to manage to a different platform.

- **Mental**: The final shade of lock-in that Hohpe defines -- and which he identifies as "most subtle, but also the most dangerous type" -- is "mental." By this, Hohpe means a situation where you have become so accustomed to doing things one way that you lack the ability to imagine alternatives, even if those alternatives are technically practical to implement. In the cloud ecosystem, mental lock-in is often a major barrier that prevents organizations from migrating to the cloud in the first place: They are so used to on-premises technologies and architectures that they can't imagine how they'd function without them. Or, you may be mentally locked into a single-cloud architecture because you have used it for so long that migrating to a multi-cloud configuration is simply unthinkable, even if the tools required to do so are readily available.

It's important to recognize that these so-called shades of lock-in are not the causes of lock-in. They are the symptoms you may experience when you fail to adopt a cloud management strategy that empowers you to avoid lock-in. We'll discuss the root causes of lock-in further below in this guide.

Ways to Avoid Cloud Lock-In

Before delving into the causes of cloud lock-in, however, it is important to enumerate the reasons why lock-in poses a real threat. These considerations are especially significant given that, as noted above, there is some debate within the technical community about whether cloud lock-in is truly a problem; and, if it is, whether the effort required to avoid lock-in is worth the results.

"Concern about cloud lock-in is probably overblown and likely counterproductive," one major tech media site would have you believe. Others contend that "there is no hope" when it comes to avoiding lock-in, because most teams are locked into specific cloud services or tools without even realizing it. Even Hohpe points out that "avoiding being locked into one aspect often locks you into another," suggesting that fighting lock-in is, more or less, a futile endeavor.

These are all valid perspectives, but only for certain cases and for certain teams. Obsessing over cloud lock-in may indeed be counterproductive if you have very basic deployment needs, or if your total cloud expenditure is minimal. A team that uses a public cloud to just host a few websites at a cost of only several hundred dollars per year probably does not need to worry much about being locked into one cloud platform or another, because there is just not enough at stake for that team. However, most cloud workloads today are more complex than this, and the stakes of lock-in are higher.

It's also true that, for many teams, total freedom from lock-in is impossible. As we've noted, lock-in comes in many forms and affects organizations to varying extents. The notion that your team can ever have total freedom to migrate or extend its cloud workloads from one platform or another, with zero need to reconfigure anything or learn any new tooling, is false. In that extreme sense, there is indeed no hope for avoiding lock-in entirely.

But just because cloud lock-in isn't always a problem doesn't mean it's never a problem. For the majority of organizations and teams, lock-in does pose real risks.

Likewise, it would be short-sighted to conclude that, because most teams can't expect to avoid cloud lock-in in all of its forms, all of the time, they should not bother trying to combat lock-in risks at all. To take this stance would be to adopt a perfectionist mentality that hinders your ability to take advantage of all that the cloud has to offer. Yes, you may always face some form of lock-in. But by striving to minimize it, you will enable yourself to get the most out of the cloud.

Specifically, by mitigating the challenges of cloud lock-in, you enrich your cloud strategy with the following:

- Cost savings: Maintaining maximum ability to use different cloud services, products, architectures or tools helps ensure that you are able to choose the most cost-effective solutions.

- Respond to changing needs: Your technical requirements are constantly changing. The needs you had to address when you first migrated to the cloud are most likely not those you face today, or those that will arise tomorrow. By being able to modify your cloud strategy at will, you maximize your ability to keep that strategy aligned with your everevolving requirements.

- Resistance to disruptions: If you depend on a particular cloud platform, product or vendor, you will face a major disruption if the solution you use is discontinued or the vendor goes out of business and you cannot migrate to an alternative solution easily. When you are free from lock-in (or your lock-in risks are minimal), however, you can simply pivot to a different platform to avoid disruptions.

- Security enhancements: Just as freedom from lock-in helps you choose the most cost-effective cloud solutions and allows you to update your cloud deployment strategy as your needs change, it also empowers you to update your cloud security strategy as new types of threats emerge and new security solutions become available.

- Team satisfaction: When your engineers enjoy the ability to choose whichever cloud solutions they deem the best, you avoid the risk of your team becoming demoralized because it has to rely on cloud tools, services or processes that it dislikes.

- End-user satisfaction: Your end-users, too, benefit from the ability to use whichever tools or services deliver the best end-user experience. When your users are employees within your organization, the public at large or both, you should strive to be able to offer them solutions based on whichever cloud technologies are the best fit for their needs at a given moment, instead of the ones you are locked into.

These are all excellent reasons for most teams that use the cloud to invest in cloud architectures, tools and management strategies that mitigate lock-in risks, even if they can't totally eradicate all forms of lock-in.

The Causes of Cloud Lock-In

Just as the symptoms of cloud lock-in can take many forms, cloud lock-in has multiple potential causes. Some result from the tools or products that an organization uses, while others arise from architectural patterns or management processes.

To avoid lock-in, you must be cognizant of the lock-in catalysts to which your team or organization is most prone. The greatest risks to watch out for include:

- Use of proprietary products: Although not all proprietary or closed-source software products lock you in, they are more likely than open source solutions to lack interoperability with third-party solutions. In this way, they restrict your ability to extend your cloud strategy. Many proprietary tools also lack solutions for automatically converting or migrating workloads to other platforms, which also causes lock-in.

- API dependency: Some vendors or platforms offer APIs that may be supported by third parties, but that nonetheless offer only limited compatibility. The Amazon S3 storage API is an example; it is compatible with many third-party storage services and applications, but if you write applications that depend on this particular API, you would need to rewrite those applications if you choose to use a different cloud storage service API.

- Organizational culture: Sometimes, the greatest lock-in risk is rooted in your organizational culture. If your team is resistant to change or unwilling to experiment with or learn new technologies, you are likely to face mental lock-in, as described above. A culture that encourages lock-in is often also one that has grown highly dependent on vendor-specific platforms, and lacks the imagination to explore alternatives.

- Legal or compliance constraints: For cloud workloads that are subject to specific legal or compliance requirements such as those associated with HIPAA or the GDPR, your ability to migrate to different architectures, products or services may be constrained. In many such cases, migration is not impossible, but it requires extra effort because you need not only to learn to implement and manage a new solution, but also to deploy it in a compliant manner.

- Cost pressures: In some cases, cloud migration or extension is technically possible, but you lack the financial resources to undergo it, which causes lock-in. Cost as a cause of cloud lock-in often goes hand-in-hand with other sources of lock-in, such as proprietary platforms: If you use a vendor-specific platform that doesn't offer an automated way of migrating to an alternative product, the manual migration effort might require so much time and money that it is infeasible for cost reasons.

- Data gravity: Data gravity refers to situations where you have so much data stored in one cloud or platform that it is very difficult to migrate the data to a different location in order to update your cloud strategy. A variant on this challenge is having a cloud architecture in which integrating data between multiple tools is inefficient or impossible. For example, if you rely on one public cloud to store data, your ability to deploy applications that consume that data on another cloud as part of a multi-cloud architecture may be hindered by the data egress fees that you would need to pay in order to move data between the two clouds, or between an on-premises environment and the public cloud.

- Security commitments: You may become so invested in a security strategy that depends on certain tools or services -- or the expertise required to manage those particular tools and services in a secure way -- that migration is difficult. You may know one public cloud vendor's IAM framework very well, for example, while lacking the time to reconfigure your IAM policies to fit a different cloud. Likewise, you may face challenges in migrating to a multi-cloud strategy because you lack an efficient way of federating identities and access-control policies across clouds.

As we've noted, these causes of cloud lock-in are closely related in many cases. Cost-driven lock-in tends to be associated with proprietary platforms, for example. So does an organizational culture that leads to lock-in.

It's clear, then, that in most cases cloud lock-in arises from a mix of different causes. There is no singular root cause of lock-in that you can simply avoid or eliminate in order to free yourself from lock-in. Once again, you need a multi-faceted anti-lock-in strategy.

Strategies for Avoiding Cloud Lock-In

Although your team's strategy for avoiding cloud lock-in must be tailored to your unique needs, there are a variety of practices that, at a high level, help to mitigate lock-in risks.

Prefer open source

Above all, preferring open source solutions over proprietary ones will go far in reducing lock-in risks. This is not because open source is a silver bullet when it comes to stopping lock-in. It's not, as we explain below in discussing how to manage lock-in on the open source Kubernetes platform. Still, by and large, open source platforms are less likely to lock you into a particular ecosystem, product or version than are those that are controlled by a single vendor.

Granted, choosing to migrate wholesale to open source solutions is often not practical, especially for teams that currently depend on proprietary platforms. In these cases, however, you can nonetheless adopt strategies that will increase your use of open source tools over time. If today you run everything on a specific public cloud vendor's native cloud services, for example, consider migrating your workloads to that same vendor's Kubernetes platform. This will get your applications into an open source environment -- albeit one that is linked to the vendor -- while allowing you to retain many of the IAM configurations and management tooling you already have in place for that cloud. From there, you can eventually move to a different Kubernetes distribution that gives you greater freedom and independence from a specific cloud.

Choose solutions with multiple offerings

When you are selecting new services or products to power your cloud strategy, look for solutions that are offered or supported by multiple vendors. Here again, Kubernetes is a prime example; it's a cloud platform that can run as part of virtually any public, private or hybrid cloud architecture.

This rule is not limited to open source platforms, however. You can also choose proprietary solutions that, despite being closed-source, offer as much interoperability as possible. For instance, as noted above, Amazon's S3 API is supported by multiple platforms, despite being controlled by Amazon.

Prefer third-party support options

It may seem natural to obtain support services directly from the vendor who develops or manages the cloud tools and products you support. However, if professional-class third-party support options exist, they are often a wiser choice. By choosing third-party support offerings, you are more likely to find options that will allow you to migrate between different tools and platforms without having to change your support strategy.

Here, Platform9 is a prime example. Platform9 offers OpenStack and Kubernetes management and support services regardless of which cloud-based or on-premises infrastructure you choose for running OpenStack or Kubernetes. With Platform9, you can migrate to different OpenStack or Kubernetes architectures while keeping the same support team and the same management tooling.

Invest in organizational culture

Because cultural resistance to migration or the acquisition of new expertise may be the biggest barrier to avoiding lock-in, it's critical to invest in building an organizational culture that promotes change and experimentation. Reward engineers who spend time exploring new cloud product or platform possibilities, for example. Reinforce practices -- such as maintaining consistent documentation and inserting clear comments within configuration code -- that make it easier to migrate or change configurations.

Anti-Lock Strategies: Kubernetes Example

To place the preceding points about avoiding cloud lock-in into context, Kubernetes offers a helpful example.

At first glance, Kubernetes may seem to be an antidote to all forms of lock-in, for two reasons. First, it's an open source platform developed by a broad community. There is minimal risk that Kubernetes will be discontinued. Second, a key part of Kubernetes's functionality is to provide an abstraction layer for cloud services and infrastructure. Using Kubernetes, you can deploy and manage containers in a consistent way on any public cloud or on-premises server cluster. You can also take advantage of other types of deployment formats, like serverless functions, in a cloud-agnostic way by running them on top of Kubernetes, thereby freeing yourself from being locked into a cloud vendor's native services.

As analysts have pointed out, however, these factors alone don't guarantee that Kubernetes will be a complete safeguard against lock-in. Sometimes, the Kubernetes deployment and management strategy that you choose becomes a form of lock-in in itself. That is especially true if you choose a Kubernetes service that is tied to a specific public cloud. These services often use, or at least recommend, special deployment and management tools in place of standard Kubernetes tooling, like kubectl. They also typically rely on vendor-specific IAM configurations, monitoring and logging stacks and so on.

That's all true, but it doesn't mean that Kubernetes is never helpful for avoiding cloud lock-in. So long as you adopt a Kubernetes strategy that is not subject to the restrictions described above, Kubernetes is an effective tool for maximizing the flexibility and agility of your cloud strategy. In particular, you should design a Kubernetes strategy that:

- Allows you to use the same deployment and management tools for your Kubernetes clusters, even if you migrate the clusters from one cloud to another, or run them on multiple clouds at once.

- Makes it easy to upgrade Kubernetes whenever you choose, instead of whenever the vendor that hosts your cluster chooses to push out an upgrade. This way, you avoid version lock-in.

- Is founded on Kubernetes's standard, open source tooling, and avoids vendor-specific tools or extensions.

- Lets you configure monitoring, logging, security and other aspects of your Kubernetes clusters in whichever ways you prefer -- including with fully open source solutions like Prometheus and the EFK stack -- rather than requiring specific tooling.

- Conforms with community-defined standards with regard to APIs, container run-times and so on.

The point here is this: Even with a platform as extensible, open source and broadly used as Kubernetes, there is no magic safeguard against lock-in. Instead, the extent to which Kubernetes can protect you from cloud lock-in hinges on the way you manage Kubernetes itself. You must actively pursue a 'lock-in free' agenda in order to leverage Kubernetes as a defense against lock-in, rather than a tool that may increase it.

Conclusion

Cloud lock-in is a complex, multi-layered topic. There is a whole range of ways in which you may become locked into cloud services, platforms or products, as well as an array of different effects that may result from lock-in. What's more, some instances of cloud lock-in are more problematic than others. Depending on your needs and goals, the lock-in threats that your team faces may be more or less serious.

This doesn't mean that you should write off cloud lock-in risks as being over-hyped. Nor should you become complacent, assuming that you can't avoid lock-in entirely, so you might as well not even try.

On the contrary, you must work actively to assess the most serious cloud lock-in risks your team and organization faces. Then, adopt tools and strategies that can mitigate them. By doing so, you place your team in a position to save money, improve cloud reliability and security, delight end-users and, in general, get the very most out of your cloud strategy.

Although, again, there is no simple trick for avoiding cloud lock-in, Platform9 can help. As a cloud-agnostic management and support solution for both OpenStack and Kubernetes, Platform9 makes it easy to deploy these open source platforms on your terms, using whichever infrastructure you like, and to move or extend them at will. By making it practical to leverage OpenStack and Kubernetes as cloud abstraction layers that allow organizations to take full advantage of cloud architectures without becoming dependent on proprietary software platforms or specific public clouds, Platform9 helps companies of all types and sizes to unlock the full potential of the modern cloud.

PLATFORM9

Cloud Lock-In: A Definitive Guide

Platform9.com/contact

Headquarters

Platform9 Systems Inc.
800 W. El Camino Real
Suite 180
Mtn. View, CA 94040
650-898-7369
info@platform9.com

About Platform9: Platform9 enables freedom in cloud computing for enterprises that need the ability to run private, edge or hybrid clouds. Our SaaS-managed cloud platform makes it easy to operate and scale clouds based on open-source standards such as Kubernetes and OpenStack; while supporting any infrastructure running on-premises or at the edge. Enterprises such as S&P Global, Kingfisher Retail, Cadence Design, Juniper Networks and Autodesk are using Platform9 to easily manage large scale private and edge clouds. The company is headquartered in Mountain View, CA and is backed by Redpoint Ventures, Menlo Ventures, Canvas Ventures, NGP Capital, Mubadala Capital and HPE Pathfinder.