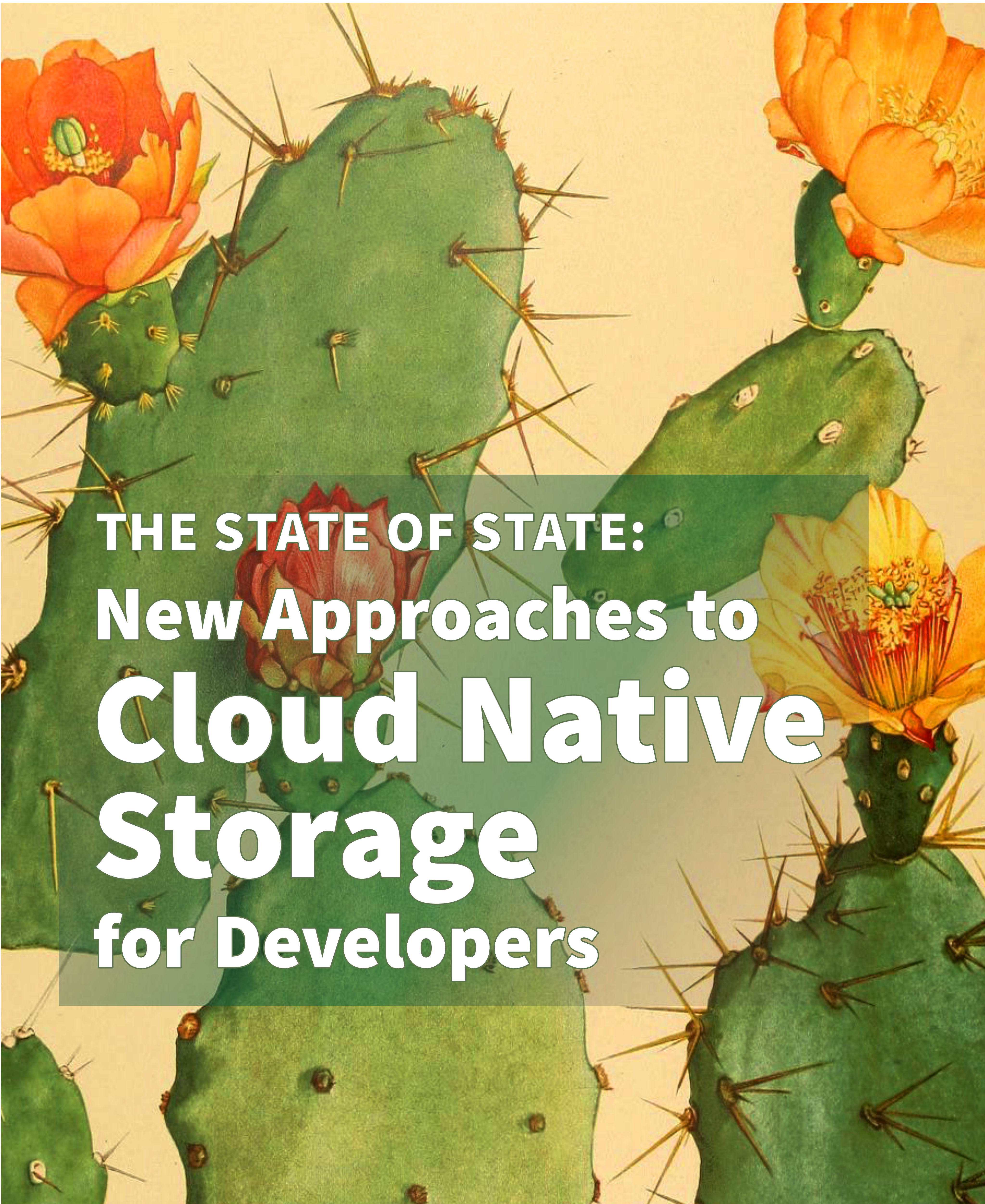
THENEWSTACK



THE STATE OF STATE: New Approaches to Cloud Native Storage for Developers

The New Stack

The State of State: New Approaches to Cloud Native Storage for Developers

Alex Williams, Founder & Editor-in-Chief

Core Team:

Benjamin Ball, Sales & Account Management Director

Emily Omier, Ebook Editor

Gabriel H. Dinh, Executive Producer

Janakiram MSV, Technical Editor

Joab Jackson, Managing Editor

Judy Williams, Copy Editor

Kiran Oliver, Podcast Producer

Lawrence Hecht, Research Director

Libby Clark, Editorial & Marketing Director

Michelle Maher, Editorial Assistant

Norris Deajon, AV Engineer

20190928

Table of Contents

Sponsor

We are grateful for the support of our ebook sponsor:

NetApp®

NetApp built its foundation on data storage but has since expanded into a full range of cloud native capabilities and services to simplify management of applications and data across on-premises and cloud-based environments. NetApp empowers global organizations to unleash the full potential of their data, foster greater innovation and optimize operations.

INTRODUCTION

The rise of containerization and the move towards cloud native development has simultaneously changed the way applications handle state and shifted the responsibility for managing storage from a dedicated storage administrator to application developers. New to the cloud native storage world? Here's what you need to know.

Contributors

Emily Omier is a content marketing consultant and writer specializing in enterprise software engineering tools.

Jean Bozman is vice president and principal analyst with Hurwitz & Associates, focusing on infrastructure for enterprise data centers and the cloud. She previously worked at IDC for 15 years, including 10 years as a research vice president in the worldwide server group.

Maxwell Cooter is launch editor of Cloud Pro and Techworld, a full-time freelance technology journalist and a part-time cricket and rugby coach.

CHAPTER 01

The Current State of State

Successfully managing state is crucial if companies are going to benefit from the speed and agility that a cloud native, microservices-based architecture brings to application development at scale. Yet, a full 68% of companies say that managing state is at least somewhat of an obstacle to moving more applications to microservices, according to a recent survey conducted by The New Stack in partnership with streaming data platform provider Lightbend. [1] The good news is that 18% of companies surveyed said that state was not at all a barrier — the solution is out there, it just takes developer training and the right technology.

So what exactly is state? It is anything an application has to "remember" after it's shut down and then spun up again. This includes both data and application configuration information. Websites were originally designed to be stateless; for example, no records about your visit to the site were stored if you closed the site. Cookies changed that. Now websites routinely remember your language preference and the content of your shopping cart even if you close the website, close your browser and turn off your computer. That is state.

In an enterprise environment, most applications do, in fact, require some kind of state. Managing state has a deep history in enterprise applications. It previously meant storing data in databases that were installed on hardware and were managed to run stateful applications. Just as importantly, the data that enterprise applications need to interact with is often governed by strict service-level agreements (SLAs), with mandates around availability, disaster recovery, security and performance. Enterprises had figured out how to handle state in an on-premises world. But the move to the cloud — and even more importantly, to container architectures — created new challenges. Business requirements haven't changed, but technology has.

Moving to a cloud native architecture is a big leap, explains Jonas Bonér, chief technology officer (CTO) at Lightbend. In his experience, many companies aren't aware of the need to change the application architecture to be cloud native. They continue handling things, like state, in essentially the same way they used to on monoliths — at least until they learn the hard way not to do so.

In the containerized age, applications are broken down into microservices and may only run long enough to perform their duty. When a microservice starts up again, it is a clean slate, knowing nothing of its former, or parallel, instances. It is by this nature that microservices can scale and handle hybrid and multicloud environments, with the downside, at least initially, that the applications no longer behave in the stateful way developers expect. [2]

Containers are also making it much easier for developers to build and manage applications. With containers, developers can build fast without the need to manage the data. Their only requirement is to manage the code. With new operator models, for example, databases are packaged and available for automatic provisioning. [3] An operator turns a complex program, with intricate provisioning and maintenance issues, into an easy-to-run service, noted Sebastien Pahl, who gave a presentation on the technology at the 2018 All Things Open conference in Raleigh, North Carolina.

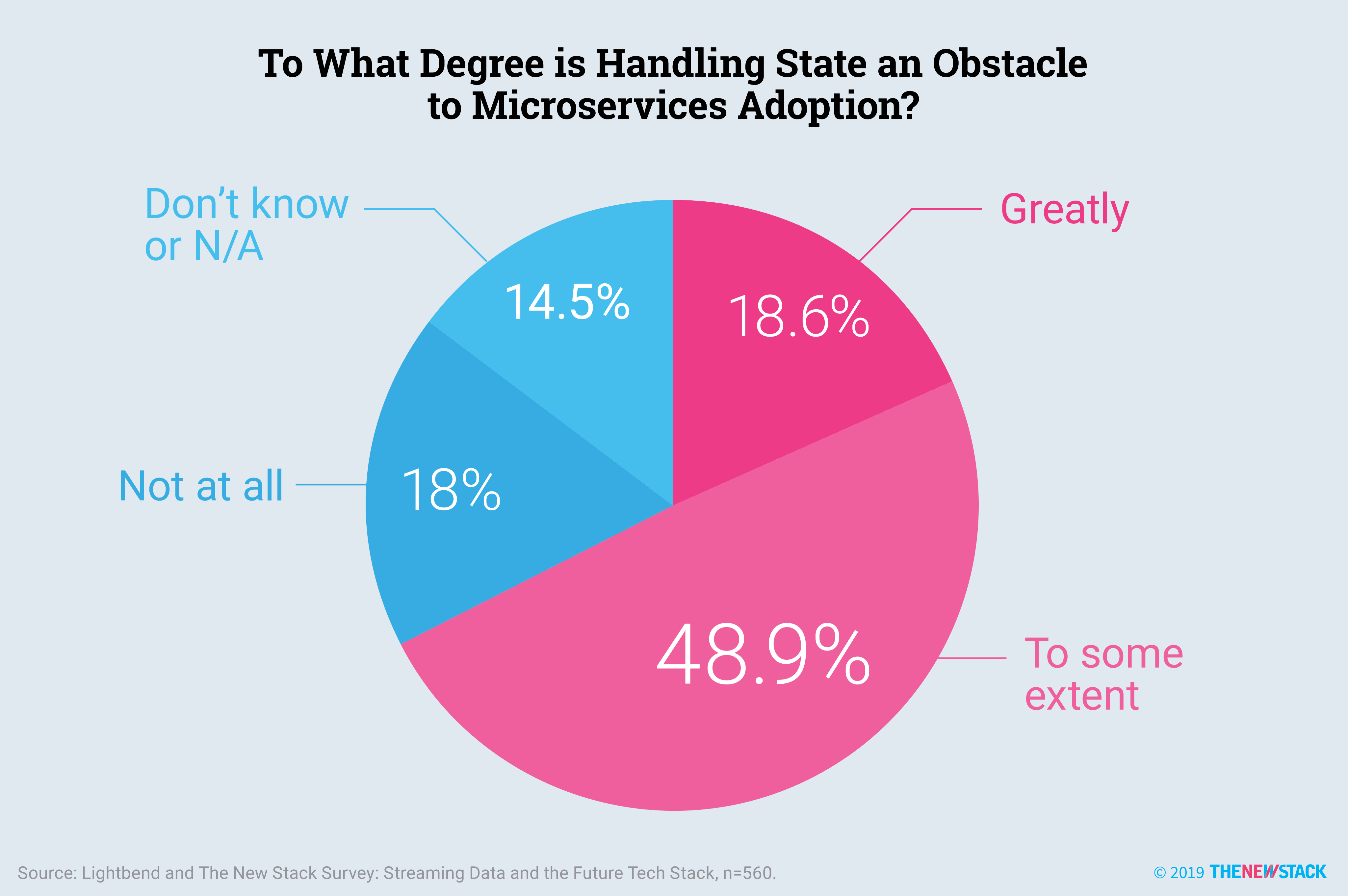


FIG 1.1: *A majority of companies say that managing state is at least somewhat of an obstacle to moving more applications to a microservices-based architecture.*

These advances provide new ways to use data and decrease the time it takes to develop applications, be they stateful or not.

Stateless Beginnings

To understand where we are now when it comes to managing state in cloud native applications, it's important to remember some key facts: Containers were designed to be stateless. Kubernetes was designed to orchestrate stateless, ephemeral, immutable containers. At first, any state that these applications needed to have was just stored externally in siloed storage devices and accessed with volume plugins.

"Even if containers were meant to be stateless, containerized architectures still needed state," explained Alex Chircop, founder and CTO of StorageOS. This state just couldn't be stored in the container or managed by the orchestrator.

As companies start seeing how Kubernetes and containerized architectures can increase application agility and speed, however, there's been an increasing push to package more and more applications in containers, and to use Kubernetes to manage both compute and storage resources.

"Now everyone has started putting stateful applications into containers," explains Chris Merz, principal technologist at NetApp. "Whether or not that is the right design, whether or not that was ever intended, it is what's happening."

State can now be handled inside of stateful containers via storage services or through different kinds of storage systems. The end goal, in either case, is to reduce the state footprint of your infrastructure and allow storage, as well as compute, to behave in a cloud native manner, said Anand Babu Periasamy, co-founder and CEO of MinIO. [4] Storage becomes more resilient, scalable and programmable.

Who Cares About Storage?

You need storage. Every function needs to be able to access some kind of storage. The consequences of problems with either the storage itself, or the ability of applications to access storage, are serious.

"When apps fail, it's often the storage causing problems," explains Irshad Raihan, director of product marketing at Red Hat. This was true when both compute and data were located in data centers, and provisioning more storage meant purchasing a piece of hardware. It is still true in a containerized, cloud native application.

The move to containerized application architectures, as well as the shift towards more cloud-based applications, changes how applications interact with storage, who is responsible for managing storage and some expectations surrounding storage capabilities. At the same time, there are core business

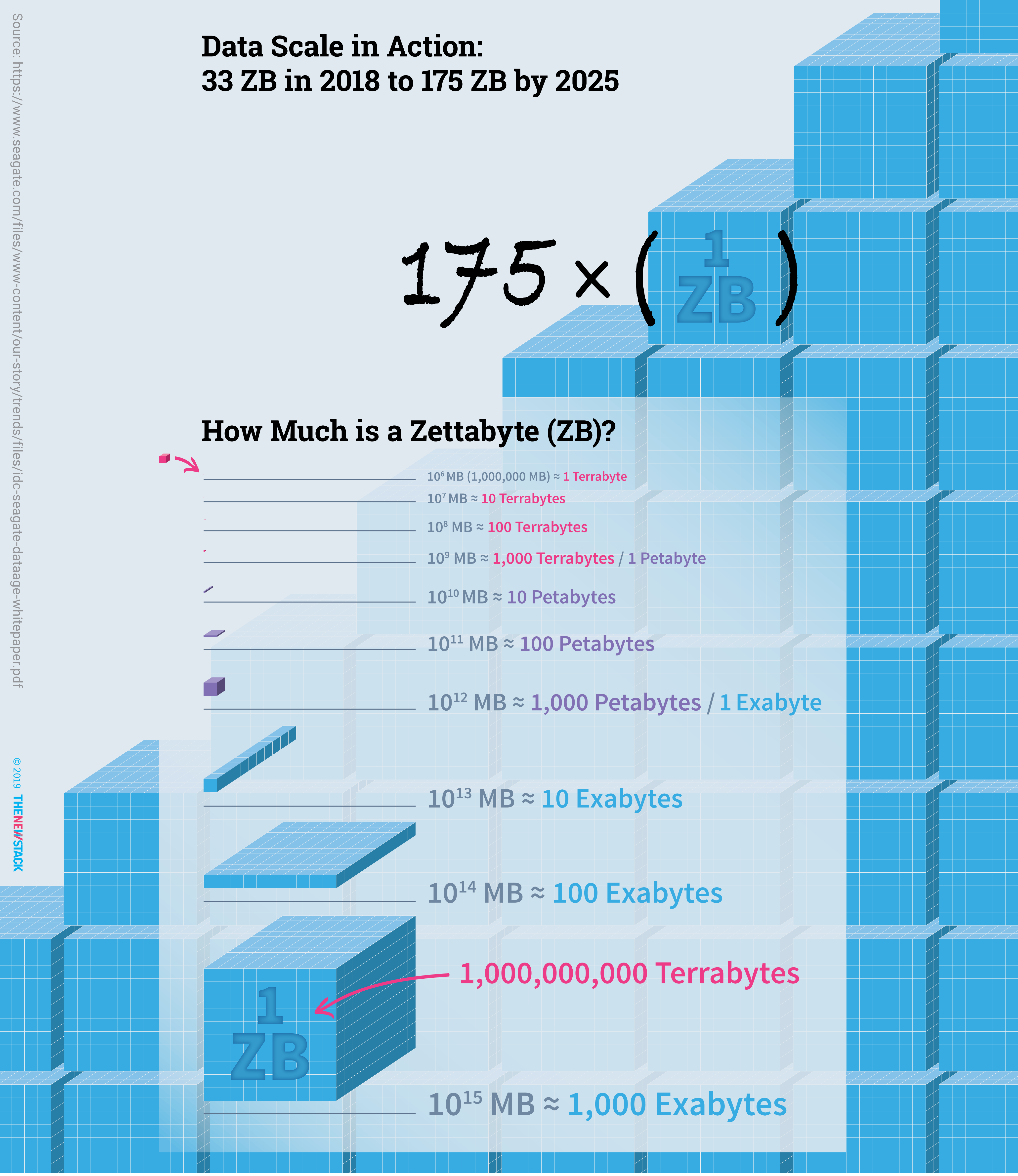


FIG 1.2: *IDC predicts that data created, captured or replicated from traditional and cloud datacenters, at the edge and endpoints, such as mobile and IoT devices, will reach 175 Zettabytes globally by 2025. A Zettabyte (ZB) is a measure of storage capacity and is 2 to the 70th power bytes, also expressed as 10^{21} (1,000,000,000,000,000,000,000 bytes) or 1 sextillion bytes.*

needs related to storage that have not changed. After all, end customers do not care whether or not their bank stores transaction histories in the cloud or on premises, or what kind of application architecture is used. They do care that no transaction history data is ever lost, no matter what disaster befalls the bank's data center or cloud provider.

Regardless of where your application is being deployed, there are three major issues that can arise from storage problems. The most serious issue is data loss or data unavailability. Only slightly less serious are performance problems and the inability to handle spikes in demand.

> "My single most important takeaway is for developers to consider storage attributes up front."
> **— Alex Chircop, founder and CTO of StorageOS.**

Whereas in the "old days" most developers depended on a storage administrator to provide resources and had to build applications around available storage, now developers have the freedom and responsibility to control the storage provisioning process — and to do so in minutes rather than weeks.

There might not be one single best way to connect cloud native workloads to storage. Unlike in the past, developers have the ability to provision storage based on the specific application needs, so each application can connect to data in an optimal way. However, there are specific issues everyone should be aware of as they consider the best storage options for cloud native projects.

Storage Attributes

Like most things in life, provisioning storage involves making tradeoffs. Storage can be optimized for the following attributes: [5]

1. Availability.

2. Scalability.

3. Global consistency.

4. Durability.

5. Performance.

It would be nice if you could get a storage system with top marks in all of those categories, but in practice that isn't possible. Different storage architectures optimize for one or more storage attributes, and choosing among them requires prioritizing your requirements.

Comparison of Key-Value Stores Across the Five Storage Attributes

	Local	Remote	Distributed and nonglobal transactional	Distributed and global transactional
Availability	Limited by local component failures.	Limited by remote component failures.	Partial failures do not affect availability or may be limited to key space.	Partial failures do not affect availability or may be limited to key space.
Scalability	Limited by local resources.	Limited by remote resources.	Scales out as more capacity is added.	Scales out as more capacity is added. API scalability is often limited by a single master.
Global consistency	Strong.	Strong.	Weak.	Strong.
Durability	Limited by local storage failure.	Limited by remote component failures.	Tolerant to partial failures.	Tolerant to partial failures.
Performance	Limited by I/O access latency.	Limited by I/O access latency and network latency.	Limited by I/O access latency and network latency.	Limited by I/O access latency, network latency and usually a single master. Multiple rounds of network latency for cross-shard transactions..

FIG 1.3: *The storage attributes best suited for any given application or architectural pattern will depend on the use case and design objectives.*

Storage can be provisioned either directly as a piece of hardware in a data center, as storage from a cloud provider or through software-defined storage, where a software layer exists between the application and the hardware or cloud provider storage. In the latter two options, the storage is still ultimately connected to a piece of hardware, even if through layers of abstractions. It's important to be aware of the attributes the underlying hardware is optimized for, because it won't be possible to completely circumvent those limitations with software layers.

Container-attached storage services allow storage infrastructure to live side by side with compute resources in the same environment. Both storage and compute can be managed by Kubernetes, simplifying management, reducing cost and improving resource utilization. [6]

At the most basic level, a developer would provision storage through Kubernetes using YAML or JSON to define the need for a volume. [7] Assuming a PersistentVolume is available (meaning an administrator has previously provisioned it), Kubernetes will then make a PersistentVolumeClaim and begin consuming the resources. A StorageClass allows administrators to create different tiers or classes of storage resources, often broken down by quality or how the storage is optimized.

Keeping Data Secure

Though it's not considered a storage attribute, security is also a critical consideration when choosing a cloud native storage solution. When attackers target an enterprise's application, their goal is often to steal data, as was the case in the recent Capital One breach. [8] Containers and the distributed environments they run in create new security challenges. According to Diamanti's 2018 Container Adoption Survey, 22% of companies running containers in production cited security as their top challenge. Keeping data secure in a cloud native environment involves the following:

- Encryption of data both at rest and in transit.
- Role-based access controls.
- Automation tools to ensure consistent application of security policies.

- Security monitoring and logging.

Protecting data goes hand in hand with protecting compute resources — in the Capital One case, a misconfigured firewall provided access to data stored in Amazon Simple Storage Service (S3) buckets. When it comes to both compute and storage security, the biggest security risk is often misconfigurations, which is why most security options focus on policy creation and enforcement to limit the possibility for human error. At the same time, data security cannot be divorced from the rest of the security strategy. If a misconfigured firewall provides access to credentials, as in the Capital One case, the fact that the S3 buckets were encrypted is irrelevant.

Building an Enterprise Storage Strategy

The options companies have as they put more stateful applications into containers fall into three broad categories:

1. Container-attached, software-defined storage.
2. Cloud storage provided by traditional storage hardware vendors.
3. Native options from cloud service providers.

While the latter two categories are self-explanatory, container-attached, software-defined storage is a relatively new type of storage. Software-defined storage is a software layer between the cloud service provider storage — or on-premises storage hardware — that pools storage resources for seamless scaling, makes it possible to pack hundreds of microservices on a single server, and provides additional functionality to make storage act in a more cloud native manner.

In many cases, enterprises will build a strategy for running stateful applications in containers using options from at least two, and often all three, categories. Let's take a look at each of those in the next three chapters.

CHAPTER 02

How the Cloud Changes the Storage Landscape

The Cloud Native Computing Foundation (CNCF) defines cloud native systems as container packaged, dynamically managed and microservices oriented. [9] Being "cloud native" ultimately has nothing to do with where the application is deployed — a monolith lifted and shifted to Amazon Web Services will never be cloud native. In addition to being packaged in containers and built on a loosely coupled microservices architecture, cloud native applications should be declarative and managed through automation tools whenever possible, starting with continuous integration through automated monitoring. [10]

According to Alex Chircop, founder and chief technology officer (CTO) of StorageOS, storage in a cloud native environment should have the same attributes as anything else that is part of the same environment, providing the same experience for managing storage that a container orchestrator like Kubernetes provides for compute. Cloud native storage — also called "container-native" to differentiate between types of storage that act more like traditional storage solutions — should meet the same criteria the CNCF establishes for all cloud native systems. Let's take a look at what that means.

Container Packaged

Container-based environments are dynamic. As each container moves around the cluster, it needs to maintain its connection to the storage volume. If you are connecting to a cloud service provider's storage service — such as Amazon Elastic Block Store (EBS), Google Cloud Storage or Microsoft's Azure Storage — directly, this requires an error prone, manually managed process of detaching and reattaching the volume to new hosts.

"It's not that cloud providers aren't providing a reliable service," says Michael Ferranti, vice president of marketing at Portworx, about the storage options provided natively by cloud service providers. "It's that Kubernetes forces you to use those services in a way they weren't designed for."

Kubernetes was designed to work with immutable, stateless containers. Kubernetes clusters are also generally deployed over several availability zones, sometimes even over multiple clouds. But cloud providers' native storage options don't allow volumes to attach to a host running across several availability zones. Unless you have a software management layer over the cloud provider service, if a container fails over to the second availability zone it would not be able to connect with its data. The same is true for many traditional hardware-based storage solutions, when running Kubernetes in a private cloud.

The Container Storage Interface (CSI) has made it easier to run stateful applications using Kubernetes by providing a standard application programming interface (API) that connects any container orchestrator to storage, whether it's software-defined storage, storage hardware or cloud provider storage. CSI was released for general availability in Kubernetes in early 2019, but while it simplifies the connection between Kubernetes and persistent storage, it doesn't change the fact that storage has to be handled outside of Kubernetes.

Microservices

Running an application made up of hundreds or thousands of microservices, all or most of which need to be connected to storage directly on EBS — or directly on your on-premises server — would not be practical because of the hard limits on the number of volumes you can attach to a Linux or Windows instance: a maximum of 40 volumes for Linux, 17 to 26 for Windows.

"If you want to run, say, 100 pods, or 150 pods, or 250 pods on a [virtual machine] VM and each of those pods needs a volume, you simply cannot do it using native EBS," explains Ferranti.

Building a cloud native application requires densely packing your microservices and their associated data. Kubernetes administrators must aggregate, pool and abstract multiple EBS volumes attached to individual nodes as a single, logical volume to the pods. Doing so requires a software layer over the cloud providers' native storage. Kasten, OpenEBS, Portworx, Rook and StorageOS all provide this software layer that pools storage resources, making it possible to run hundreds of pods on a single instance.

Dynamic Management

Removing opportunities for human error by handling as much as possible through APIs is a key part of cloud native best practices. Not only should you be able to declare your storage needs programmatically through an intuitive user interface (UI), cloud native storage should require minimal human intervention to detach and reattach volumes as containers move in a cluster, to handle scaling, to adapt in case an application fails over to another availability zone and to handle storage degradation as the application runs.

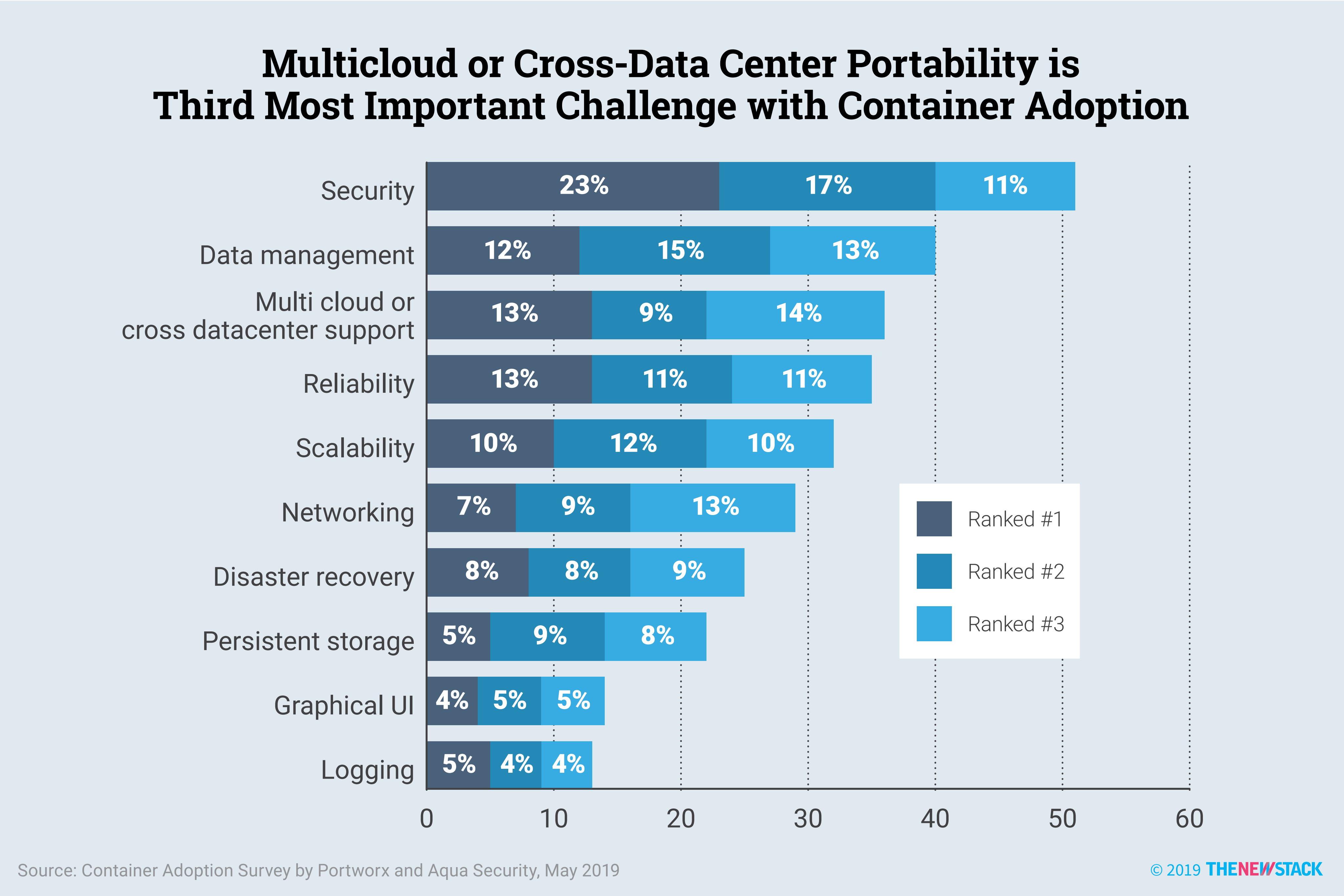
Using a software-defined storage layer, either over the cloud service provider's native storage service or over data center hardware in a private cloud, is the best way to get this cloud native functionality for storage. Software-defined storage also provides solutions for two other areas that companies worry about when it comes to storage: portability and Day 2 Operations.

Portability

"When you move from one cloud to another cloud you realize the part [where] change is required is actually storage, not compute. The APIs are not compatible," said Anand Babu Periasamy, co-founder and CEO of MinIO.

Periasamy says that portability between public clouds and between public and private clouds is among the top concerns MinIO hears about from clients. Portworx and Aqua Security's 2019 Container Adoption Survey found that concerns about multicloud or cross-data center portability was the third most important challenge companies face when adopting containers. [11] Especially considering that running multicloud environments is common, your storage

FIG 2.1: *Moving data between different public clouds and between public and private clouds is a top concern among IT pros surveyed by Portworx and Aqua Security.*



solution should allow data to move between public clouds and private clouds easily.

Day 2 Operations

As applications run, things go wrong: Networks are partitioned, disks fail and containers move around. When thinking about storage, it's important to consider how you will maintain performance, as there is wear and tear on your storage system, and how, as much as possible, the maintenance will be handled through automation. This should include backups and disaster recovery, performance and security monitoring, logging, volume management and de-provisioning. Operations such as taking a snapshot are particularly tricky in a distributed system because the application's data is stored on many virtual machines. Snapshots have to be application-centric, and able to locate all of the relevant data for a particular application, without capturing any extraneous data.

One of the challenges in doing these types of Day 2 Operations for storage is that while application architectures have evolved rapidly, storage infrastructure and tooling has not kept up.

> "Just because you can connect a container to a storage system does not mean you get the operational characteristics that you want in a cloud native application."
>
> **– Michael Ferranti, vice president of marketing at Portworx.**

Getting those operational characteristics requires a software layer between your application and your storage resources. Think of this software layer as a way to translate between your containerized, API-driven, microservices-based application and storage resources. Even in a cloud environment, storage resources are connected to a machine and still have many of the same limitations as on-premises storage options.

For individual developers and entire companies looking for ways to connect their containerized applications to storage, there are two key considerations. First, consider what are the most important storage attributes (see Chapter 1) and make sure that your entire storage stack, down to the hardware, is optimized appropriately. Second, make sure that your storage solution facilitates the dynamic management and portability that you expect from a cloud native application.

According to Gartner, which now includes cloud native storage on its Hype Cycle for Storage Technologies, "container-native storage is specifically designed to support container workloads and focus on addressing unique cloud native scale and performance demands while providing deep integration with the container orchestration systems."

Gartner makes the point that the common foundation for container-native storage systems is deployment based on a single, software-defined pool of storage, where containerized applications and persistent storage use the same platform, and where Kubernetes acts as the orchestration technology.

Container-native storage technology is new. Most of the companies behind both open source and proprietary options are still young. These companies will argue that only a truly container-native storage solution will work for cloud native applications, but there are other options. Traditional storage vendors have solutions optimized for containers that offer their own benefits and drawbacks. The right choice for any individual company depends on the company's level of cloud native maturity, current application architecture and anticipated future needs. This is also not necessarily an either/or decision matrix — many companies will use software-defined storage on top of their storage hardware purchased from a traditional storage vendor.

Storage for Cloud Native DevOps

Part of the transition to DevOps that comes with cloud native application development has been a shift in responsibility for storage, away from dedicated specialists towards developers who are increasingly responsible for provisioning the storage for the applications they build.

“People do not want storage to be a complicated task,” says Chris Merz, principal technologist at NetApp in this episode of The New Stack Makers podcast recorded with Alex Williams, founder and publisher of The New Stack. “It … should follow the same patterns as the systems that DevOps practitioners and cloud native architects are building every day.”

Before Kubernetes, building and operating container-based applications was onerous — it involved manually handling tasks like DNS management, load balancing, scaling and resource monitoring. Now Kubernetes handles all of that — but there needs to be a way to get the same level of automation for storage, Merz says.

Trident, an open source project developed and maintained by NetApp, acts as a storage orchestrator, abstracting away some of the complexities (and decision-making) from developers looking to provision storage. Developers don’t have to worry about the details of how the storage works — Trident integrates Kubernetes with NetApp’s on-premises and cloud-based storage products.

Using a cloud native storage orchestrator that’s integrated into Kubernetes makes applications dramatically easier and faster to deploy, and also makes it easier to set up correctly for monitoring and observability as the application runs in production. This is important for any application, but even more so for stateful applications.

"Stateful applications tend to be systems of record or something more core to your application framework," Merz says. Given that effective, enterprise-scale monitoring involves thousands of metrics, you need something that will automate the container setup every time, and will work for stateful applications.

Whatever the application architecture, Merz says, the challenges remain essentially the same: scale and control — in the form of security, observability and data management. It's entirely possible to get the same scalability and control in a cloud native application that enterprises expect from storage. It just requires using different tools, and making sure that developers who are now in charge of provisioning storage have the knowledge and tools to make the right choices and set up the storage correctly.

Listen on SoundCloud

Chris Merz, *principal technologist, provides market strategy and technology expertise across the NetApp product and cloud services portfolio; specifically in hybrid multicloud, DevOps and cloud native technologies.*

CHAPTER 03

Storage Vendors Adapt to a New Competitive Landscape

Traditional storage hardware was not designed to handle the petabytes, exabytes and zettabytes of data being used in modern applications, nor were they designed to handle that data broken down into tiny chunks, stored in a distributed system. Yet that is exactly what is happening in modern software development. [12]

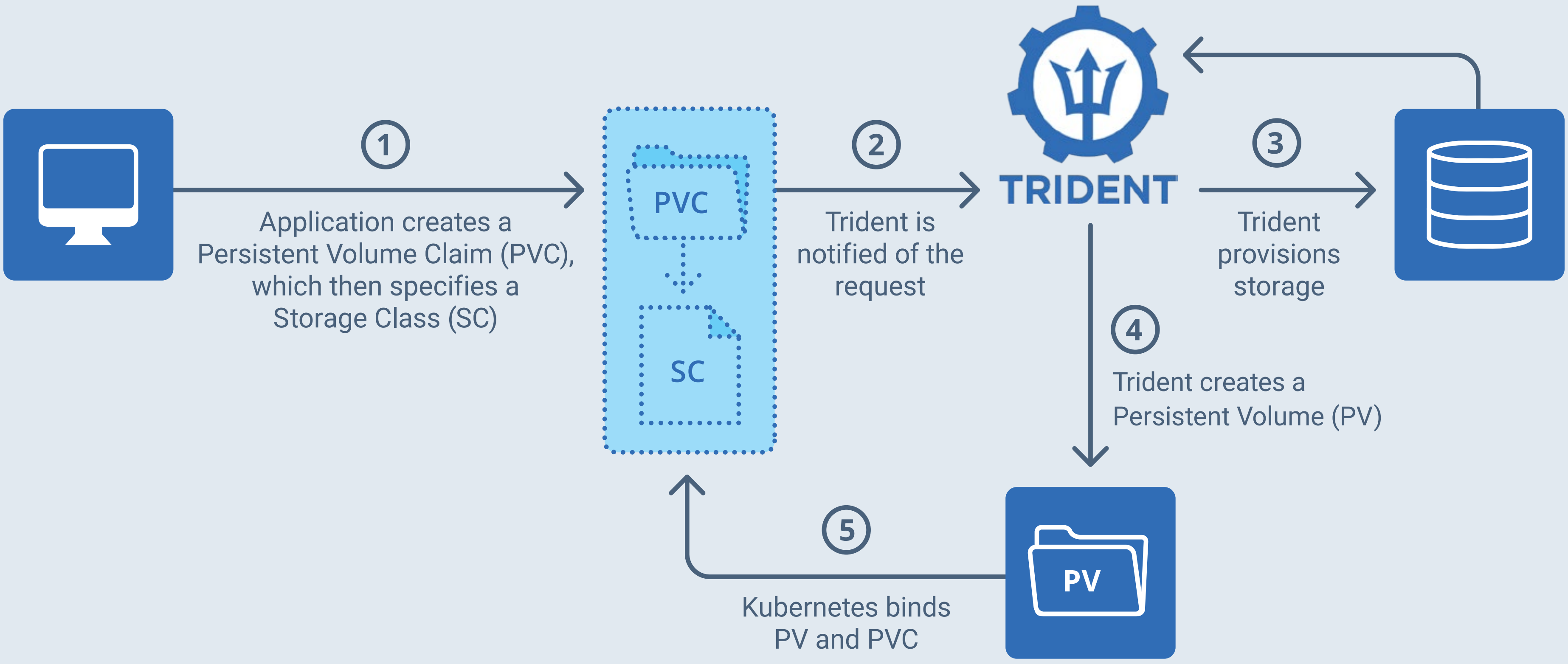
The enterprise storage landscape is shifting. And the companies that have a long history of providing enterprise storage hardware are adapting.

Hardware Vendor or Storage Vendor?

"Some are embracing public cloud and not seeing it as the enemy, while others still say that the future in on-prem," says Julia Palmer, research director at Gartner, about how traditional enterprise storage vendors are adjusting to the changes that containers and the cloud bring to the storage landscape.

NetApp, Palmer says, is an example of a traditional storage company adapting to the reality of a cloud-first future. Ingo Fuchs, chief technologist at NetApp, says that it's the old 'Are we a steamboat company or a transportation company?' question. "The perception is that we are a hardware vendor, but we're a software company that's become a cloud company," he says.

How Trident Connects Containers to Storage With Kubernetes



Source: Trident documentation.
https://netapp-trident.readthedocs.io/en/stable-v19.07/dag/kubernetes/concepts_and_definitions.html#connecting-containers-to-storage

FIG 3.1: *In Kubernetes, applications request access to storage resources with Persistent Volume Claims (PVC). The Storage Class (SC) object describes storage with specific characteristics, such as a provisioner, that will trigger Kubernetes to wait for the volume to be provisioned by the specified provider — in this case, Trident. Persistent Volumes (PVs) are Kubernetes objects that describe how to connect to a storage device.*

Recognizing that being a 'storage' company of the future means getting away from a dependence on hardware sales, NetApp in 2016 released Trident, a storage orchestrator designed to work with persistent volumes (PV) in Kubernetes, and optimized for managing storage on NetApp's platforms. A storage orchestrator provides the dynamic, automated management for storage resources that a container orchestrator like Kubernetes provides for compute.

NetApp is not the only traditional storage company developing a strategy for the cloud. Dell EMC, Hitachi and Pure Storage all offer storage optimized for the cloud — their strategies are focused on allowing enterprises to build hybrid cloud or private cloud architectures that will provide the advantages of the cloud, while still requiring ongoing support for the on-premises data center.

> “Some [storage vendors] are embracing public cloud and not seeing it as the enemy, while others still say that the future is on-prem.”
> **– Julia Palmer, research director at Gartner.**

It’s challenging to make generalizations about the architectures traditional storage vendors are building their cloud storage offerings on. These are large vendors, and often have a number of options marketed as ‘cloud native.’ These range from Storage as a Service models, like Pure Storage’s Evergreen Storage Service, to Data Centers as a Service like VMware Cloud on Dell EMC. While these are all ways to abstract storage needs, they are not the same as the software-defined storage layer characteristic of container-native storage.

Trident, like the similar offerings from other traditional storage vendors, is a way to connect Kubernetes to the vendor’s own products. In contrast, one of the main selling points for many software-defined storage vendors is being hardware- or cloud service-provider agnostic — in other words, providing the ability to move data between different public clouds, between public and private cloud and to run over any type of storage hardware. The business reasons for using Trident are different than those for using a software-defined storage option like OpenEBS, Portworx or Rook. In both cases, the software-defined storage solution works in tandem with the Kubernetes scheduler to ensure that the pods always land on the node that has an associated volume. This approach is important to running stateful workloads on Kubernetes in production.

As a result of these ambiguities and a certain degree of purism around what fits into the cloud native definition, not everyone believes that traditional storage vendors are capable of making the shift. Cody Hill, director of technology at managed Kubernetes provider Platform 9, for example, considers the company’s current setup with Pure Storage “non-cloud native” and would like to move to a software-defined storage option like OpenEBS, Portworx or Rook to get cloud native functionality.

"Traditional storage vendors are claiming they provide 'cloud native storage,'" says Evan Powell, CEO of MayaData, a cloud native storage provider and sponsor of the OpenEBS open source storage platform. "They have so distorted the term already that I prefer the term container-attached storage to describe what solutions like ourselves and Portworx offer, which is a truly cloud native architecture to deliver storage and data management services."

Kubernetes Drives Change, Not the Cloud

Kubernetes has been a game-changer in terms of cloud native development, but it's also a source of major challenges in managing stateful applications in the cloud. The container orchestrator does not offer a way to manage data storage; in fact, it was designed for stateless applications. But with the right integrations, Kubernetes can change the storage game, too.

"[Kubernetes] is a platform for [the] building and operation of distributed software and can be used to deliver storage and data management services to distributed environments," Powell explains.

Especially when paired with other solutions, like Istio, Prometheus, Weave Scope and more, storage can be delivered as a set of loosely coupled microservices. It's also become easier to integrate storage with Kubernetes since the Container Storage Interface (CSI) was released. CSI standardizes the way that all container orchestrators communicate with all storage vendors, whether container-native, software-defined storage or a cloud provider's storage service.

In other words, it's not actually the move 'to the cloud' that is changing the way storage needs to behave; it's the adoption of containers and Kubernetes and the changes to application architectures that have dramatically changed storage requirements.

Storage Specialist Role Evolves

Both Kubernetes and DevOps practices have dramatically changed the role of the storage specialist; most significantly, developers can usually provision storage themselves, without ever talking to a storage specialist. That doesn't necessarily mean they have all the necessary knowledge, but just that storage specialists serve as the experts helping developers optimize for the right attributes and developing higher-level policies, instead of gatekeepers determining who gets what type of storage and how much. 13

Storage used to be an Ops problem, and developers often had no control over the storage provisioning process. In a cloud native, DevOps world developers have both the ability and the responsibility for storage, which involves making choices that will influence application performance.

At the same time, storage specialists must adapt or die. "Storage specialists now no longer need to provision storage or even do most operations tasks if their end users are using a solution like our [MayaData Data Agility Platform] MDAP based on OpenEBS or [a] similar solution," says Powell, from OpenEBS.

"These specialists become operations architects, responsible, for example, for helping to curate storage classes that their end users — developers — consume," Powell said. "This requires an understanding of infrastructure as code or GitOps, as well as observability and chaos engineering. Solutions like the open source Weave Scope for observability — which has visibility down to the persistent volumes — as well as the open source Litmus, which is chaos engineering for stateful workloads, may be of use."

Making the Right Storage Choices

So how should these newly empowered developers make the right choice? "Applications can store state in a lot of different ways," Alex Chircop, founder and chief technology officer (CTO) of StorageOS, explains. Organizations have a plethora of options when it comes to how to manage state for a particular application's needs — and it's not a mature landscape, so in most cases, the companies that are successfully handling state in containers are early adopters and still fairly inexperienced in the process of using containers to handle state.

"There's a decision matrix," explains Michael Ferranti, vice president of marketing at Portworx. First, you decide whether to handle data on Kubernetes or outside Kubernetes. If your data is managed outside of Kubernetes, it will connect to Kubernetes via an application programming interface (API), will be either a cloud provider data service or an on-premises database managed by a database administrator. If you choose to use Kubernetes directly to manage storage, you'll interact with Kubernetes via a CSI to provision volumes.

Typically, a cloud native application will require some combination of object stores, key-value stores and files. The best way to handle the data depends on the particular use case: Video, for example, would probably be best stored using an object store.

The Cloud Native Computing Foundation's (CNCF) storage landscape [14] includes a wide variety of open source and commercial storage options that can work for cloud native workloads. There are container-native storage options, storage offerings from the major cloud service providers and many traditional storage hardware vendors in the list. Some of the technologies focus instead on one type of storage: Elastifile provides file storage; MinIO focuses on object storage. See the next section for the full CNCF list.

Most applications are going to use a combination of the technologies on the CNCF list to build a complete cloud native storage strategy. Developers have the freedom to choose the best storage options for the application at hand, but also need to understand how to select the best options, given the application requirements, and how to tie together the different technologies.

"It is a new problem for developers who are development-centric," Merz explains. "Traditionally, solving state at scale was an operational or database administrator thing. These are hardcore Ops functions. Handling state at scale was like the biggest topic at KubeCon + CloudNativeCon in Barcelona."

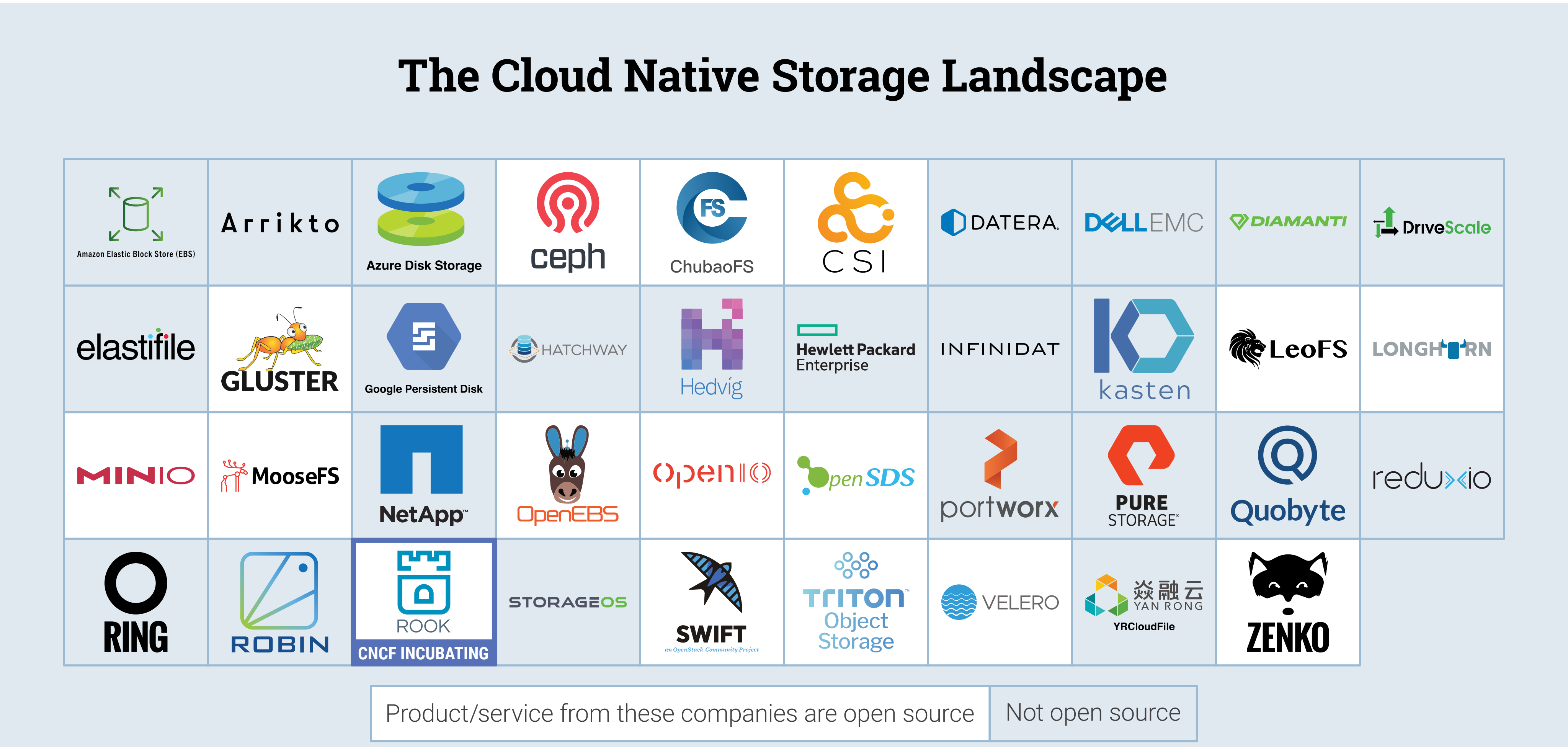
At the same time, storage specialists are being asked to be part of the development equation, too. Instead of being asked just about input/output operations per second (IOPS) and throughput, storage specialists are expected to be focused on helping get the stateful application into production faster — can we deliver in three months instead of 18 months?

Both traditional storage providers and traditional storage specialists have to adapt to the new, cloud-first reality. "If all you care about is on-prem storage," says Fuchs, from NetApp, "well, that's an interesting career choice."

Cloud Native Storage Solutions List

The Cloud Native Computing Foundation's storage landscape includes products based on open source projects, and purely commercial solutions designed for

FIG 3.2: *The Cloud Native Computing Foundation's map of cloud native technologies includes 39 projects in the storage space.*



Source: https://landscape.cncf.io

use with cloud native applications and Kubernetes. The list is not comprehensive, but it includes the storage solutions the CNCF recommends for use in cloud native architectures. The three categories of cloud native storage — container-attached, software-defined storage; cloud storage provided by traditional storage hardware vendors; and native options from cloud service providers — are all represented in the list.

Open Source Projects and Products

1. **Ceph**: An open source software storage platform supported by Red Hat and designed to provide highly scalable object storage.

2. **ChubaoFS**: A distributed file system for cloud native applications.

3. **Container Storage Interface (CSI)**: An open source initiative, now fully supported in Kubernetes, to provide a standard way for container orchestrators to connect with storage, whether it is software-defined, cloud service provider storage or bare metal.

4. **Gluster**: An open source file storage system designed for the cloud and supported by Red Hat.

5. **LeoFS**: Open source, scalable, fault tolerant object storage for web applications.

6. **Longhorn**: A distributed block storage system for Kubernetes developed and supported by Rancher.

7. **MinIO**: Highly scalable, open source object storage for cloud native applications.

8. **MooseFS**: A distributed file storage system developed by Core Technology.

9. **OpenEBS**: Open source, container-attached software-defined storage for cloud native, containerized applications.

10. **OpenIO**: A high performance object storage solution optimized for big data and artificial intelligence (AI) workloads.

11. **OpenSDS**: A community of storage specialists working under the umbrella of The Linux Foundation to solve data challenges in a cloud native world.

12. **Rook**: An open source, software-defined storage solution currently being incubated by the CNCF.

13. **Swift**: An open source object file system backed by Red Hat.

14. **Triton Object Storage**: An object storage system designed for cloud native applications and developed by Joyent.

15. **Velero**: Previously called Heptio Ark, Velero is a tool to manage backups, disaster recovery and migration of Kubernetes applications.

16. **Zenko**: An open source data management platform designed for multicloud and developed by Scality.

Commercial Solutions

1. **Amazon Elastic Block Storage (EBS)**: AWS block storage system.

2. **Arrikto**: The company behind Rok, a data management platform designed for cloud native environments.

3. **Azure Disk Storage**: High performance disk storage to support Microsoft Azure's virtual machines.

4. **Datera**: Software-defined Data center as a Service, focused on providing data management and automation capabilities to companies using a private or hybrid cloud model.

5. **Dell EMC**: A traditional storage vendor that offers Data Center as a Service as well as software interface options to connect its storage hardware to Kubernetes.

6. **Diamanti**: An enterprise Kubernetes platform with integrated persistent storage, targeted towards enterprises using a hybrid cloud strategy.
7. **DriveScale**: An enterprise platform for creating a fully automated private cloud on bare metal, handling both compute and storage.
8. **Elastifile**: File Storage as a Service for cloud native applications. Elastifile was recently acquired by Google Cloud.
9. **Google Persistent Disk**: Google's block file system.
10. **Hatchway**: An open source project from VMware designed to facilitate connecting containers to persistent storage in a vSphere environment.
11. **Hedvig**: A software-defined storage and data management platform designed for the public, private or hybrid cloud. Hedvig is part of Commvault.
12. **HPE Storage**: A traditional storage vendor with various partnerships and software solutions to make connecting Kubernetes to a private or hybrid cloud easier.
13. **Infinidat**: A storage hardware vendor that specializes in petabyte-scale storage.
14. **Kasten**: Data backup and recovery for Kubernetes.
15. **NetApp**: Traditional storage vendor that developed Trident to connect its storage hardware with Kubernetes.
16. **Portworx**: A software-defined, container-attached storage solution and data management platform.
17. **Pure Storage**: Traditional storage vendor with various options for private and hybrid cloud application development.

18. **Quobyte**: A software-defined data center file system that integrates file, block and object storage.

19. **Reduxio**: Container-native data management platform designed for multicloud application architectures.

20. **Robin Systems**: A software-defined data management platform designed for Kubernetes.

21. **Scality RING**: A software-defined file and object storage solution designed for large-scale on-premises data.

22. **StorageOS**: A container-native, software-defined storage solution for running Kubernetes on multicloud or hybrid cloud.

23. **YRCloudFile**: Software-defined storage solution from Beijing-based YanRong.

CHAPTER 04

Cloud Storage Services for Cloud Native Applications

An ebook about cloud storage would be incomplete without a discussion of the dizzying number of storage options offered directly from cloud service providers. For most developers, the first experiences with cloud native storage — and perhaps with provisioning storage themselves — will come from a cloud service provider (CSP). But once developers start moving beyond provisioning storage for a small, non-critical application, manually managing volume attachments and data placement for a distributed application becomes too complex to do successfully.

When applications and teams scale — and when complicated and strict business requirements are tied to data — using only cloud service provider storage is a barrier. Using strictly CSP storage can make it harder to follow a multicloud or hybrid cloud strategy. Heavy reliance on CSP storage can get very expensive at scale, often dwarfing the costs of compute in the cloud. In fact, a Cohesity survey in 2018 found that 47% of IT executives are worried about blowing their budget on storage. A more recent survey, Mass Data Fragmentation in the Cloud, indicates that the cost of moving large volumes of data is a significant concern. [15]

The cloud service provider storage market is not fully commodified, but all

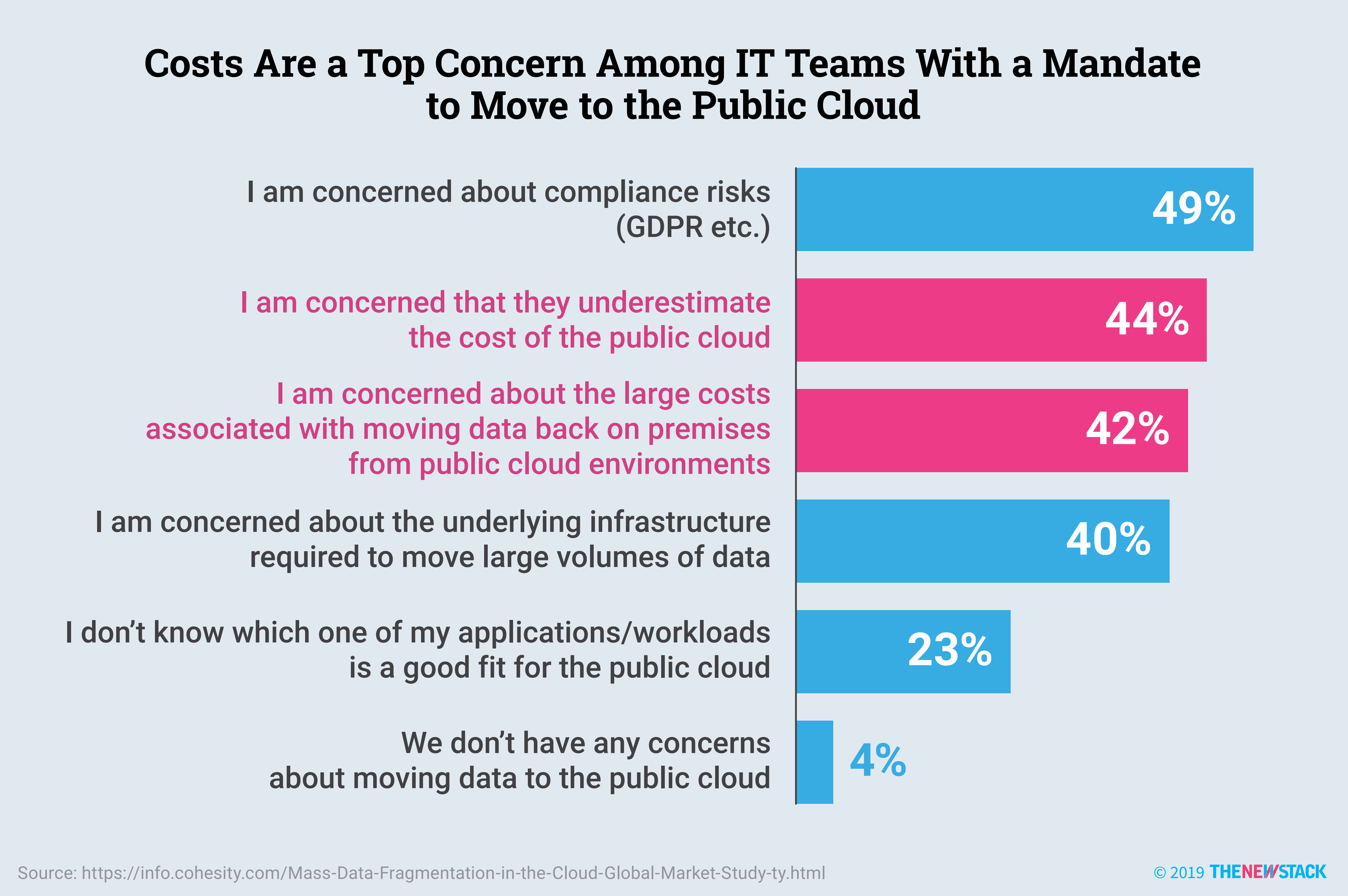


FIG 4.1: *Heavy reliance on cloud service provider storage can get very expensive at scale.*

CSPs offer similar functionality for object, file and block storage. When running at scale, CSP-provided storage is necessary but not sufficient for running stateful applications. It is the bottom layer of your storage infrastructure, but works best when managed by a storage orchestrator.

Storage in a Hybrid Cloud

One of Kubernetes' selling points is the ability to run on a hybrid cloud. Google Anthos, IBM Cloud Private, Red Hat OpenShift and VMware Project Pacific are all examples of how Kubernetes is becoming the de facto standard for hybrid cloud strategies. Kubernetes provides the mobility for compute resources, but what about storage? Distributed applications are creating new storage challenges for all cloud native customers, but those with hybrid clouds or multicloud deployments have unique challenges, said Peder Ulander, technology executive at Amazon Web Services.

> “The biggest challenge is that businesses are moving from data centers to centers of data. Not everything sits in one place. Data may be in Amazon, in Google or out at the edge.”
> **– Peder Ulander, technology executive at Amazon Web Services.**

The large CSPs are all competing to provide services for cloud native, highly-distributed applications. They are not as eager to allow customers to easily run workloads across cloud providers or in their own data centers. Adding a software-defined storage layer over the cloud service provider storage provides the same portability for data that Kubernetes provides for compute resources.

If you are running any part of an application in the public cloud, you will be using cloud provider storage. You need to understand the different types of storage and what kinds of workloads they are best for. Then you can decide whether or not a software storage layer between your application and the cloud service provider storage is needed to provide the cloud native functionality we discussed in Chapter 2. If you have a hybrid cloud setup, you'll be using all three storage categories: traditional storage hardware in the private cloud, cloud service provider storage in the public cloud and likely a software-defined storage option to tie the two together.

CSP Storage Services

Cloud service providers can ease the developer workload with automated cloud storage services, hiding some of the complexity of storage provisioning and management. DevOps teams now work with CSPs who manage the infrastructure. Developers rely on the providers to declaratively provision storage as needed. There is no need for DevOps teams to intervene. It is all managed automatically through the cloud service provider.

However, IT organizations should study their CSP's storage service options before concluding that all of the needed functionality is baked in. Most of the basic functions for cloud object storage are covered by the major CSPs — Amazon Web Services, Google Cloud Platform (GCP) and Microsoft Azure. In particular, when examining cloud storage services that will protect mission-critical data DevOps teams will still need to have a checklist for enterprise workloads migrating to the cloud. And they will need data management software and consoles to give IT a unified view of all stored data — often provided by systems vendors or independent software vendors (ISVs). Third-party SD-WAN providers are also part of the hybrid cloud storage ecosystem. Below is a quick overview of cloud storage services offered by the largest CSPs.

Amazon Web Services (AWS)

AWS is integrating cloud native application support into its Amazon Elastic Compute Cloud (EC2) service and its S3 storage service. It joined the Cloud Native Computing Foundation (CNCF) to gain a deeper level of feedback on technologies it must support for cloud native computing and storage.

At the AWS re:Invent 2018 conference, Adrian Cockcroft, vice president of AWS cloud architecture strategy, outlined new features of EKS, the Elastic Kubernetes Service, which supports container components for cloud native applications, including cloud storage.

- **AWS Cloud Data Migration** supports transfer of large datasets, via Snowball appliances (multiple petabytes per appliance). Another service, the AWS Storage Gateway, connects AWS storage services for purposes of backup/archiving, disaster recovery, cloud data processing, storage tiering and migration.

- **Amazon Elastic Block Store (EBS)** supports block storage, which has long been used in enterprise servers and enterprise storage to store application data.

- **Amazon Elastic File System (EFS)** supports shared file storage. This is useful when transferring files from servers supporting Network File System (NFS)-based file storage.

- **Amazon Simple Storage Service (S3)** is durable, scalable and available storage for block, file and object storage. The earliest form of AWS storage, the S3 service provides extensible and highly available data storage, across storage tiers.

Microsoft Azure

Microsoft is providing storage services and software tools that address the needs of highly scalable, distributed applications that will be deployed on Microsoft Azure.

Azure supports open source workloads typical of cloud native applications. The Azure Kubernetes Service (AKS) is aimed at cloud native applications deployed on Azure, providing a Kubernetes-based container solution for software-defined storage. Following is a list of specific Azure storage services, based on storage type, that can be used for cloud native applications:

- **Azure Blob Storage** is a massively scalable object store for text and binary data. This is likely to be the most visible Azure service, given the highly scalable nature of cloud native application deployments.

- **Azure Files** supports managed file shares for cloud and on-premises deployments. Azure Files uses the Server Message Block (SMB) protocol. This service supports enterprise applications that are migrating to the cloud.

- **Azure Queue Storage** is a messaging store that links application components.

- **Azure Table Storage** is a NoSQL store for schema-less storage of structured data.

- **Azure Ultra SSD** cloud service, not yet generally available, provides a tier of low-latency SSD flash storage with high IOPS (input/output operations per second).

Google Cloud Platform (GCP)

Google Cloud Platform has long provided storage for cloud native applications and cloud services. Given Google's role in the development of Kubernetes orchestration software, and its deep support for artificial intelligence (AI) and machine learning (ML), cloud native applications are a focus of its cloud storage services. These include the following:

- **Cloud Bigtable** is a scalable, fully managed NoSQL wide-column database for real-time access and analytics workloads. Real-time computing is a fast-growing segment of the IT market, although real-time applications account for a relatively small slice of the enterprise computing marketplace.

- **Cloud Datastore** is a scalable, fully managed NoSQL document database for web and mobile applications. Cloud native applications often generate data that will be stored in NoSQL databases.

- **Google Cloud Storage** is a scalable, fully managed, reliable and cost-efficient object/blob store for storing unstructured data such as images, pictures and videos. Unstructured data is the fastest growing category for data generated by cloud native applications.

- **Persistent Disk** is a durable, high-performance block storage system.

IBM Cloud and Oracle Cloud

With smaller customer bases than the largest CSPs, IBM Cloud and Oracle Cloud emphasize support for enterprise workloads making the transition to cloud, leveraging containers and software-defined storage.

Both IBM and Oracle offer services that will run cloud services on premises at their customers' sites. These on-premises cloud services are managed by the cloud providers, both of which have roots in enterprise computing. Both give customers the option to store data within the original enterprise corporate firewall, for mission-critical data security, for compliance or for data that must be stored within one geographic region.

Conclusion

There is general agreement in the storage industry that state is possible to manage successfully in container-based, cloud native applications — and, in fact, must be handled successfully if companies are going to move their entire application suite to a cloud native, container-based architecture. In general, companies seem to be moving in the right direction — slowly abandoning practices that work in a monolith, but not in a distributed system.

But there are still some problems that are tricky. Visibility, or understanding what particular data is located where, is difficult in a distributed environment where the actual data placement is handled by software rather than a database administrator. But knowing the exact placement of data is important: If one server is hacked, for example, companies need to know what data was stored on that particular server for compliance reasons. Maintaining data security by default, at all times is also a primary concern as companies move more sensitive data to cloud native applications, Alex Chircop, founder and chief technology officer (CTO) of StorageOS, says.

Particularly, as more and more applications become data-centric and machine learning is increasingly important, it's also necessary to store data close to the compute resources, Jonas Bonér, CTO of Lightbend, says. Conversely, Chris Merz, principal technologist at NetApp, thinks focusing on locality is short-sighted, but acknowledges that the core goal — making it easier for data and compute to interact and reducing processing friction — is important. Merz thinks a very smart orchestration platform is a better way to solve the connection between compute and data in data-centric applications.

Ultimately, Merz says, the types of challenges enterprise customers need to solve, when it comes to managing state, are the same as the challenges from 20 years ago. They need to manage persistence and to do so in a way that is scalable. Data needs to be kept secure, and it needs to be highly available and meet business requirements around compliance. But the ecosystem has changed, the architectures have changed, amounts of data have changed and the extent to which data is integrated into the application and the business has changed.

"I would say that the majority of enterprises are not yet running their stateful services in a cloud native manner," Michael Ferranti, vice president of marketing at Portworx, explains. "But they will be in the next 10 years. The agility gains of running your entire application stack in a cloud native manner are so great that as soon as the business requirements can be addressed, businesses of all sizes will start to do it."

Bibliography

1. **Concern About 'State' Lessens as More Applications Use Stream Processing**

   by Lawrence Hecht, The New Stack, May 16, 2019.

   This article discusses the results of a survey conducted by The New Stack and Lightbend about how companies are handling state in a microservices architecture.

2. **Reduxio Launches a Microservices-Based, Container-Native Storage Platform**

   by Mike Melanson, The New Stack, May 20, 2019.

   This article outlines how and why Reduxio created a new container-native platform designed to handle stateful applications.

3. **Database Operators Bring Stateful Workloads to Kubernetes**

   by Joab Jackson, The New Stack, January 22, 2019.

   This article discusses how database management systems providers are using Red Hat's Operator Framework to make it easier to run stateful services on Kubernetes.

4. **What the Container Storage Interface Means for Storage Evolution**

   The New Stack Makers podcast with Janakiram MSV, a TNS correspondent and principal of Janakiram & Associates, and Anand Babu Periasamy, co-founder and CEO at MinIO, May 9, 2019.

   The New Stack Founder and Publisher Alex Williams hosts this discussion about how using the Container Storage Interface is different from connecting to storage with volume plugins.

5. [**CNCF Storage Landscape White Paper**]
by Alex Chircop, Quinton Hoole, Clinton Kitson, Xiang Li, Luis Pabón, and Xing Yang, Cloud Native Computing Foundation, December 18, 2018.

A summary of the storage attributes and how they relate to cloud native application development.

6. **Why Cloud Native Storage Requires Tightly-Coupled Containers and Microservices**
by Nir Peleg, founder and chief technology officer of Reduxio, The New Stack, May 9, 2019.

This article makes the case that to realize the full benefits of a microservices architecture, storage infrastructure needs to be in the same environment as compute infrastructure rather than in siloed storage devices.

7. **Running Stateful Applications in Kubernetes: Storage Provisioning and Allocation**
by Janakiram MSV, The New Stack, October 3, 2016.

An introduction to how Kubernetes connects to storage and key terms related to storage orchestration.

8. **Capital One's Cloud Misconfiguration Woes Have Been an Industry-Wide Fear**
by Lawrence Hecht, The New Stack, July 30, 2019.

Cutting through the spin from security companies to take a hard look at why the Capital One security breach happened and how similar breaches can be avoided in the future.

9. **CNCF Cloud Native Definition v 1.0**
by the Cloud Native Computing Foundation (CNCF), June 11, 2018.
This is the GitHub repo for the official definition of cloud native technologies approved by the CNCF's technical oversight committee.

10. **10 Key Attributes of Cloud Native Applications**
by Janakiram MSV, The New Stack, July 19, 2018.

Being cloud native is about much more than deploying an application to the public cloud. This article outlines the attributes of cloud native applications, including being container packaged, using a microservices architecture and being API-driven.

11. **2019 Container Adoption Survey**
by Portworx and Aqua Security, May 2019.
The fourth annual Container Adoption Survey done by Portworx shows that while the vast majority of companies are moving towards containerization, there are still concerns related to data management and storage.

12. **The Convergence of Object Storage and Cloud Native Technologies**
by Irshad Raihan, director of storage marketing at Red Hat, The New Stack, January 29, 2019.

An introduction to how projects like Ceph and Rook are helping developers get the scalability, visibility and security they need for storage resources.

13. See #7.

14. **CNCF Cloud Native Interactive Landscape**
by the Cloud Native Computing Foundation (CNCF), graphic generated September 2019.
This interactive graphic lists 40 storage-related open source projects and commercial products in the cloud native ecosystem.

15. **Wasting Money on Public Cloud Storage**
by Lawrence Hecht, The New Stack, June 27, 2019.

Companies are worried about many of the costs associated with storage, from data migration costs to rightsizing public cloud storage. This article discusses the results of several surveys about the cost concerns related to storage.

Disclosure

The following companies are sponsors of The New Stack:

Aspen Mesh, Atomist, Capsule8, CircleCI, CloudBees, Cloud Foundry Foundation, Cloud Native Computing Foundation, Dynatrace, Epsagon, Exoscale, GitLab, HAProxy, Harness, HashiCorp, Humio, InfluxData, KubeCon + CloudNativeCon, Lightbend, NetApp, New Relic, Nirmata, NS1, Oracle, Packet, Palo Alto Networks, Pivotal, Portworx, Puppet, Raygun, Red Hat, Redis Labs, Semaphore, The Linux Foundation, Tidelift, Tricentis, VMware and WSO2.