When Containers Are Persistent, Data Storage Needs Evolve

WHITE PAPER

As modern enterprises become more sophisticated in the type of applications they enable with microservices and containerized applications, the need for persistent storage for those containers becomes increasingly critical.

This white paper looks at the evolution of containers from ephemeral to persistent and discusses ways for developers and DevOps teams to utilize scalable, performant and inexpensive storage solutions for containers—whether traditionally provisioned or serverless—across geographies worldwide.

Containers and agility

Organizations are adopting containers versus virtual machines for many reasons. To start, containers can start up rapidly and allow developers to package all of their code's dependencies together in one container image. Containers are also more efficient with system resources than VMs and more easily portable.

In addition, they enable agile development by providing a perfect platform for microservices, with each container running a function or independent piece of the complete application. So it's no surprise that the container market is exploding, with IDC predicting that 70% of IT organizations will adopt containers by 2023 to improve agility.

Containers have evolved from roots in Unix chroot and BSD jails to today's Docker and have seen a concurrent evolution from an ephemeral development platform to the platform of choice for agile, persistent applications. This evolution means it is becoming commonplace for new applications to be built using modern containers

aws

TechTarget | Custom Media

techniques. Yet, for some applications to take advantage of the agility offered by containers, data must be more accessible, and shareable. Without persistent storage, containers could not deliver the high availability and durability needed for stateful workloads, development tools and shared notebook systems.

Where containers live

A recent study by Nucleus Research found that four out of five containers run in the cloud, and 80% of them run on Amazon Web Services (AWS) as well as 82% of all Kubernetes workloads. Why is AWS so popular for containers?

First, there are many deployment options. Organizations can use Amazon Elastic Compute Cloud (Amazon EC2) or can choose from two container-specific orchestration tools: Amazon Elastic Container Service (Amazon ECS) or Amazon Elastic Kubernetes Service (Amazon EKS). Further, AWS Fargate offers a serverless compute engine for Amazon ECS and Amazon EKS, completely eliminating compute infrastructure management by providing dynamically scalable compute for any containerized application so organizations never overprovision or pay for unused resources.

Like Amazon Elastic File System (Amazon EFS), Amazon ECS and Amazon EKS are architected to be regional, meaning they run in all availability zones (AZs) in an AWS region to support availability and scalability for virtually any environment.

Why persistent storage for containers?

As organizations increasingly standardize applications on container orchestration systems, they are running more stateful workloads on containers. This often presents a challenge if the container orchestrator cannot support persistent storage. The lack of persistent access does not hinder modern stateless microservices, but it does present an obstacle to a stateful application's ability to track changes over time, and to share data with each other.

Many types of applications, such as databases and web servers, require data to persist from the storage layer. These include stateful, long-running applications like content management systems, whose files change frequently as editors publish new content or administrators install plugins; and applications that need to share data with one another, possibly spanning across multiple availability zones, such as machine learning frameworks and collaborative data science workspaces.

Like Amazon Elastic File System (Amazon EFS), Amazon ECS and Amazon EKS are architected to be regional, meaning they run in all availability zones (AZs) in an AWS region to support availability and scalability for virtually any environment.

As organizations become increasingly cloud-native, IT and business units both demand applications that evolve from "just working" to applications that are:

- Durable
- Scalable to any production demand
- Elastic and adaptive to fluctuations in demand
- Ready for day-to-day operational challenges

Even businesses just starting on their cloud journey are increasingly finding that containers are the right way to deliver elasticity in both compute and storage while meeting the demands of developers, DevOps and production environments.

Although enterprises can increase the elasticity and scalability of applications—while reducing the administrative overhead by modernizing apps and adopting a containerized, microservices approach—not every application can be quickly redeployed as microservices. Those enterprise applications are next to take advantage of the speed and efficiency of containers and serverless compute architectures.

Persistent storage for containers: The ideal

From IT's perspective, what's needed is simply the ability to build and run containerized apps and operations seamlessly. Unfortunately, on-premises silo-ed IT hasn't evolved to meet these needs—limited capacity to scale, management overhead and constant on-premises technology refresh cycles all inhibit agility and scalability.

For organizations to get the most out of persistent storage for their containerized workloads, a solution should offer the following benefits:

- Self-service, as-a-service design that's easy to deploy, use and manage
- Set-and-forget simplicity without the infrastructure worries
- Policy-driven automation
- Compatibility with containerized and serverless platforms
- Regional availability and durability for performance, redundancy and compliance concerns
- Rock-solid security providing only the data access needed
- Ability to support statefulness across application instances
- Easy sharing of data, either among concurrent instances of a single scale-out application or between multiple applications

What's at stake

Inaction is not an option—businesses that do not modernize their applications and move to agile development and deployment methodologies can find themselves at a competitive disadvantage. And to support a broad array of application types, persistent storage is a requirement. The inability to adopt microservices for containers or support stateful transactions between containers can stall modernization efforts.

Additionally, without persistent access to data, data scientists, data analysts and other business units will not be able to enjoy the freedom to share—and collaborate around—data generated from containerized application workloads.

Ultimately, the lack of a solid strategy for persistent storage can impede digital transformation efforts and impact market readiness.

Stateful storage for containers

There is a better way. Amazon Elastic File System (Amazon EFS) delivers persistent storage for stateful containers and does so with a DevOps attitude. Designed with the same core properties of Amazon ECS and Amazon EKS, Amazon EFS is elastic and regional, and it integrates seamlessly with Amazon EKS and Amazon ECS as smoothly as peas and carrots, scaling elastically without the need to reconfigure the file system. And Amazon EFS works hand in hand with AWS Fargate to deliver serverless containers with persistent storage.

Amazon EFS provides shared storage that is consistent across container pods and across AZs, ensuring every container always has secure access to the latest version of every file. As a fully managed service, Amazon EFS delivers high availability and rapid failover with a fully managed service.

Additionally, where containers are clustered across AZs, the shared volumes can be accessed concurrently from any of the AZs. This enables organizations to run applications in those AZs with favorable spot pricing, reducing costs without the need to reconfigure the storage layer. And because Amazon EFS is regional by design, when a container moves or restarts, it will reconnect to the shared filesystem, even if it is relaunched in another AZ.

T-Mobile®

Global telecom giant T-Mobile was challenged by a customer-facing app that underwent large spikes in usage based on both time of day and the month of year. Its existing infrastructure wasn't able to support the scalability required without overprovisioning to support peak demand.

T-Mobile modernized the applications to employ a microservices approach and deployed containers utilizing Amazon EFS to provide persistent storage and the ability to dynamically scale without any storage management overhead. The benefits were clear:

- **16,000 containers under management with Amazon EFS**
- **Reduced storage costs by 70% compared with DIY, while enjoying reduced storage management overhead**
- **Improved cycle time for deploying new application services**

"We are a large organization that has lots of applications with varying requirements for availability and performance," says Amreth Chandrasehar, principal architect at T-Mobile. "EFS provides us with a common storage platform that meets these requirements across the board."

Amreth Chandrasehar
Senior Enterprise Architect, T-Mobile

What Amazon EFS delivers

Since all the key components that support containers on AWS—Amazon ECS, Amazon EKS, AWS Fargate and Amazon EFS—are highly available, scalable and high performance, developers don't need to design for those features. They're already incorporated into the AWS platform.

Amazon EFS brings many benefits to organizations adopting persistent containers. First, it brings an agile methodology to container storage deployment and management in line with other DevOps best practices.

Amazon EFS "democratizes" container storage as well. Developers no longer need to worry about OS, storage types or anything infrastructure-related, for that matter. IT and users alike benefit from a design with zero overhead that can scale, grow or contract as needs change.

Organizations also benefit from the ability to continually optimize container storage for the best performance, and they can integrate with AWS Fargate to deploy containers and persistent storage while eliminating the need to manage servers, clusters or any infrastructure at all. Amazon EFS configuration is inside the Amazon ECS task definition, and connectivity is behind the scenes, simplifying and automating stateful application development and data sharing.

To ensure secure data access, organizations can use Amazon EFS Access Points, which simplify access to shared data by integrating with AWS Identity and Access Management (IAM). The resulting fine-grained security ensures that applications, users and other microservices have access only to the data they need and that unauthorized applications have no access to any data at all. Each microservice can have its own directory and controlled or partial access to just those parts of the file system. And with AWS Fargate, each task or pod runs in its own kernel, yielding workload isolation and improved security by design.

Summary

The enterprise container journey is expanding to all applications, stateless and stateful, as containers increasingly are viewed as the modern way to develop and deploy applications. For many, containers are the new normal. As more mission-critical applications are deployed in containers, the need for persistent storage for containers becomes increasingly important.

Development teams, IT and operations all want to deploy containers using an agile methodology, and AWS delivers policy-based automation and no-ops administration to storage and compute that are secure, available and durable even across regions.

The AWS container ecosystem goes beyond Amazon EFS and persistent storage to support development and DevOps efforts to integrate new microservice-based apps into the enterprise. From stateless to stateful to serverless, and completely managed with AWS Fargate, Amazon EFS is a natural fit for any container environment in the AWS cloud.

Click here to find out how Amazon EFS simplifies persistent storage for your containers.