Accelerate application modernization

for serverless, containers, and microservices-based architectures with persistent and shared data

1 | Application modernization

Speed up time to market, increase innovation, improve reliability and TCO, and use the right tool for each workload

A modern application represents a combination of modern technologies, architectures, software delivery practices, and operational processes that leads teams to deliver value faster, more frequently, and more consistently. Modern applications take advantage of distributed technologies and focus on agile, event-driven serverless components that allow teams to offload undifferentiated heavy lifting, such as provisioning and managing compute infrastructure. This enables them to spend more time on delivering value for their customers.

Organizations are modernizing applications with microservices, serverless, and containerized applications to respond to customers' needs and scale to millions of users. It removes infrastructure management, helping you to implement more agile processes and adopt a DevOps culture. In turn, these enable you to deliver applications and features faster and continuously improve customer experiences.

Serverless »

AWS Lambda »

Containers »

Data »

Why serverless?

Organizations want their development teams to focus on innovation and move with speed. Yet, most time is spent on operations and maintenance.

A serverless strategy is based on the following tenets: no server management, pay-for-value services, nearly continuous scaling, and built-in fault tolerance. When adopting a serverless service or building a serverless architecture, these ideals are fundamental to application development strategy. AWS helps you build and iterate faster by taking best practices and building them into our services, so you have less and less to manage over time. We design them to be as simple to use and control as possible—so you can focus more on serving your customers.

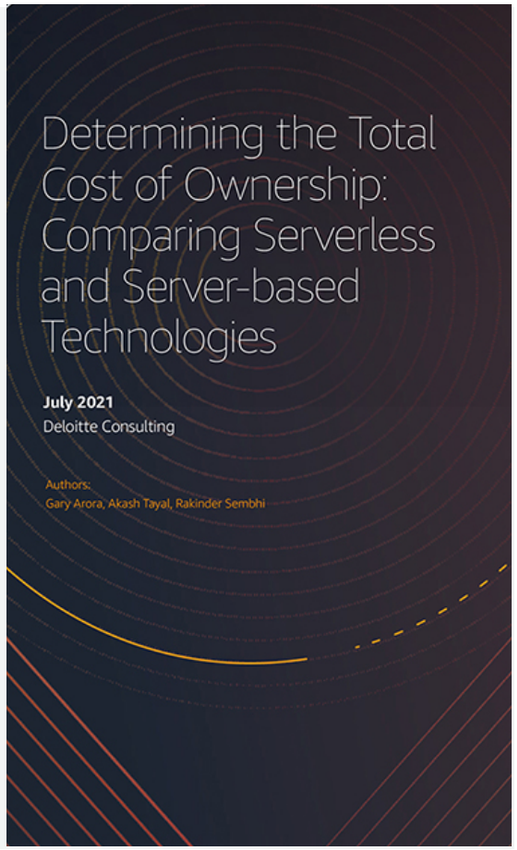
CASE STUDY

Determining the TCO of serverless compared to server-based approaches

Serverless technologies provide the opportunity for faster time to market by dynamically and automatically allocating compute and memory based on user requests. They also provide cost savings through hands-off infrastructure management, which enables companies to redirect IT budget and human capital from operations to innovation. It's no wonder that demand for serverless is on the rise.

Learn why running your applications through serverless is significantly cheaper compared to server-based approaches.

Read the whitepaper »



AWS Lambda

Let's look at one of our key serverless compute services: **AWS Lambda**. It's one of the first services adopted when starting a modernization journey. AWS Lambda is a serverless compute service that lets you run code without provisioning or managing servers, creating workload-aware cluster scaling logic, maintaining event integrations, and managing runtimes. With AWS Lambda, you can run code for virtually any type of application or backend service—all with zero administration.

Just upload your code as a .zip file or container image, and AWS Lambda automatically and precisely allocates compute execution power and runs your code based on the incoming request or event for any scale of traffic. You can set up your code to automatically trigger from over 200 AWS services and SaaS applications or call it directly from any web or mobile app. You can write AWS Lambda functions in your favorite language (Node.js, Python, Go, Java, and more) and use both serverless and container tools, such as AWS Serverless Application Model (SAM) or Docker CLI, to build, test, and deploy your functions.

CASE STUDY

Asurion builds on-demand ML, using AWS Lambda and Amazon Elastic File System (Amazon EFS)

Asurion is a leading provider of device insurance, warranty, and support service, and its experts can repair, replace, and resolve nearly any consumer tech issue. Asurion used an array of AWS technologies and performed real-time analysis of customer experiences during support calls using ML on AWS Lambda but struggled with fitting those call recordings in the AWS Lambda /tmp space. Asurion solved the issue by using **Amazon EFS** to give extended storage space to its ML functions running in AWS Lambda. Now, Asurion's ML inference infrastructure scales elastically with call volume, and the company has reduced operational overhead as compared to maintaining instances and auto scaling.

Watch the AWS re:Invent session »

asurion

"We really wanted to use AWS Lambda to make our ML inference elastic but thought we wouldn't be able to because of the size of data the process required. With Amazon EFS, we were easily able to give our function all of the storage space it needed."

Jeff Tougas,
Former Senior Principal Software Engineer, Asurion

Evolving containers with persistent data

Containerizing your applications provides a predictable and simple way to create, package, and deploy software across different computing environments.

When an application running in a container is terminated, associated data becomes inaccessible by the application. While this is fine for a stateless application—one that neither reads nor stores information about its state from one time to the next—stateful applications are different. A stateful application can remember some things about its state each time it runs.

Containerizing applications that require state means connecting to persistent storage. These applications may also need to share data among concurrent instances of a single-scale-out application or among multiple applications. Persistent, shared file storage enables containerizing stateful workloads like ML models and DevOps workloads such as GitLab, Jenkins, and Elasticsearch.

CASE STUDY

Acquia modernizes web hosting with Amazon Elastic Kubernetes Service (Amazon EKS) and Amazon EFS

Acquia's software and services are built around Drupal to give enterprise companies the ability to build, operate, and optimize websites, applications, and other digital experiences. Acquia is continuously seeking to improve its web hosting environment to enhance customer experience while operating as efficiently as possible. By containerizing and running its hosting application on **Amazon EKS** and using Amazon EFS for persistent storage, Acquia was able to dynamically scale its customer environments. Acquia lowered its TCO through improved storage and compute utilization and further reduced administrative burden by leveraging fully managed services from AWS.

Watch the AWS re:Invent session »

Acquia®

"By containerizing our hosting applications and running them on Amazon EKS and Amazon EFS, we have improved our customer experience while considerably reducing our infrastructure and operational maintenance overhead."

Jake Farrell,
Senior Director of Engineering,
Acquia

Advancing modern applications with data

Modern, data-intensive applications require fast access to large volumes of shared data. Because containers are transient in nature, long-running applications can benefit from keeping state in durable storage. Distributed applications, like ML training, web serving, and content management systems, benefit from persistent storage in a shared storage layer.

AWS Lambda lets you run large-scale and mission-critical serverless applications. **Amazon EFS** provides highly available and durable serverless storage for those applications and simplifies the sharing of data that needs to persist beyond and between executions of AWS Lambda functions and **AWS Fargate** tasks. This powerful combination is ideal for building ML applications, loading large models, libraries, and other reference data, processing and backing up large amounts of data, hosting web content, and developing internal build systems.

CASE STUDY

Simplify persistent storage for modern applications

As modern organizations leverage more microservices and containerized applications, the need for persistent storage for containers becomes increasingly critical. This whitepaper looks at the evolution of containers—from ephemeral to persistent—and discusses ways for developers and DevOps teams to utilize scalable, performant, and economical storage solutions from AWS for traditionally provisioned and serverless containers across geographies worldwide.

Read the whitepaper Persistent File Storage for Modern Applications »

2 | Expand possibilities with persistent and shared data

Persistent and shared data use cases for application modernization

Modernize application development
Modern, distributed applications, like ML training and stateful microservices, benefit from a shared storage layer. Amazon Elastic File System (Amazon EFS) enables you to build modern applications and persist and share data from your AWS containers and serverless applications—with zero management required. With Amazon EFS, you can focus on your applications, not on managing infrastructure.

Simplify DevOps
Be more agile and responsive to your customers' needs with scalable and highly available solutions. Amazon EFS provides a common storage repository to development environments, enabling you to share code and other files in a secure and organized way. Provision, duplicate, scale, and archive your test, development, and production environments with a few clicks.

Modernize content management systems
Modernizing your content management systems helps your organization get its products and services to market faster, more reliably, securely, and at a lower cost. Together, AWS containers, AWS Fargate, and Amazon EFS provide a simple, elastic, highly available, and secure solution to help you modernize websites, online publications, and enterprise content management.

Deploy stateful microservices
For stateful modern applications, Amazon EFS provides a data foundation, operating in concert with containers and serverless technologies to reliably and consistently deploy to AWS and allowing data to persist application state.

Accelerate data science
Amazon EFS provides the ease of use, scale, performance, and consistency needed for ML and big data analytics workloads. Data scientists can use Amazon EFS to create personalized environments with home directories to store notebook files, training data, and model artifacts. Amazon SageMaker integrates with Amazon EFS for training jobs, allowing data scientists to iterate quickly.

The modern applications advantage »

Amazon Elastic File System (Amazon EFS) »

The modern applications advantage

Organizations modernizing their applications with Amazon EFS are typically interested in application migration, application augmentation, or both. Migration involves either updating your build processes to build container images with existing applications or updating applications to run in a function-based compute environment like AWS Lambda. With augmentation, rather than migrate existing applications, you develop microservices with new functionality that can access shared data or state with existing applications. In both cases, this is an opportunity to migrate from self-managed storage to serverless AWS data services, like Amazon EFS.

CASE STUDY

T-Mobile improves performance and reduces costs with containers

Global telecom giant T-Mobile was challenged by a customer-facing app that underwent large spikes in usage based on the time of day and the month of the 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 company now has 16,000 containers under management with Amazon EFS, has reduced storage costs by 70 percent while enjoying reduced storage management overhead, and has improved cycle times for deploying new application services.

Read the T-Mobile case study »

T-Mobile

"We are a large organization that has lots of applications with varying requirements for availability and performance. (Amazon) EFS provides us with a common storage platform that meets these requirements across the board."

Amreth Chandrasehar,
Former Senior Enterprise Architect,
T-Mobile

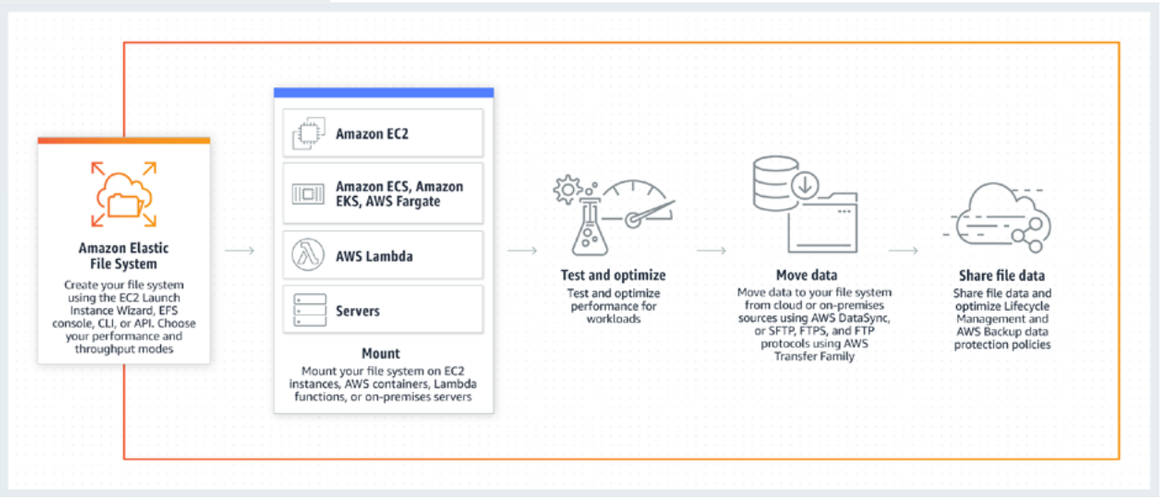
Simple, serverless, set-and-forget, elastic file system

Amazon EFS enables you to build modern applications and persist and share data from your AWS containers and serverless applications—all with zero management required. You can share file data without provisioning or managing storage. And Amazon EFS is built to scale on demand to petabytes without disrupting applications. Grow and shrink your file systems automatically as you add and remove files, eliminating the need to provision and manage capacity to accommodate growth.

Designed for the vast majority of file workloads, from latency-sensitive applications to highly parallelized scale-out jobs requiring high throughput, Amazon EFS is ideal for ML, web serving, content management, and everything in-between. Amazon EFS is designed for 99.999999999% (11 nines) of durability, is highly available, and is natively integrated with AWS Backup, so your data is available whenever it's needed.

Amazon EFS provides concurrent access for tens of thousands of connections for Amazon Elastic Computing (Amazon EC2) Instances, containers, and AWS Lambda functions. You pay only for the storage you use, with no minimum storage requirements.

HOW IT WORKS: AMAZON EFS



Watch the Amazon EFS video »

3 | The solution

Persistent, shared file storage for modern applications

AWS Lambda lets you run large-scale and mission-critical serverless applications. Amazon provides highly available and durable serverless storage for those applications and simplifies the sharing of data that needs to persist beyond and between executions of AWS Lambda functions and **AWS Fargate** tasks.

Amazon Elastic File System (Amazon EFS) attachments are configured in the application metadata, such as the **Amazon Elastic Container Service (Amazon ECS)** task definition or the Kubernetes persistent volume, including connectivity, so developers can focus on their applications, not infrastructure.

AWS containers, AWS Lambda, and Amazon EFS all scale up and down elastically with your applications and have lower operational overhead for supporting continuous integration and continuous delivery (CI/CD) DevOps pipelines.

Modernizing applications also helps you improve reliability. AWS containers, AWS Lambda, and Amazon EFS are all regional services that span multiple availability zones with automatic failover.

Modernizing applications with AWS is cost-optimized. You pay only for the storage and compute you use. Amazon EFS scales on demand from zero to petabytes with no disruptions, automatically growing and shrinking as you add and remove files. **Amazon ECS**, **Amazon Elastic Kubernetes Services (Amazon EKS)**, and **AWS Fargate** Cluster Auto Scaling also enable capacity to grow and shrink to meet demand.

Discover Financial Services accelerated analytics and time to insights using AWS »

How it works »

Get started »

AWS services »

Discover Financial Services accelerated analytics and time to insights using AWS

Discover Financial Services provides banking and credit products to help customers achieve their financial goals. Over the years, individual analytics practices sprang up within Discover's teams and business units. In all, there were about 8–10 toolsets across 12 teams. Each practice required different skill sets and diverse tools. The company's development team was tasked with creating a centralized platform that would allow the company's data scientists to collaborate in a common environment, an internal data science workbench called AIR9.

Discover determined that Kubernetes was a good fit to host AIR9 because many of the data science tools the company already used naturally lent themselves to containerization. Having dedicated containers would allow for isolated workloads and enable users to install custom packages and make changes to their environments that would be difficult to manage in a multi-tenant environment.

CASE STUDY

50 percent
reduction in storage costs, using Amazon EFS Infrequent Access for home directories and Amazon Simple Storage Service (Amazon S3) for backups

10x–20x
improvement in execution time over on-premises systems in a number of use cases

Built
a collaborative platform used by 85 percent of Discover's data scientists

90 percent
Cut storage management time by 90 percent

Read the Discover Financial Services case study »

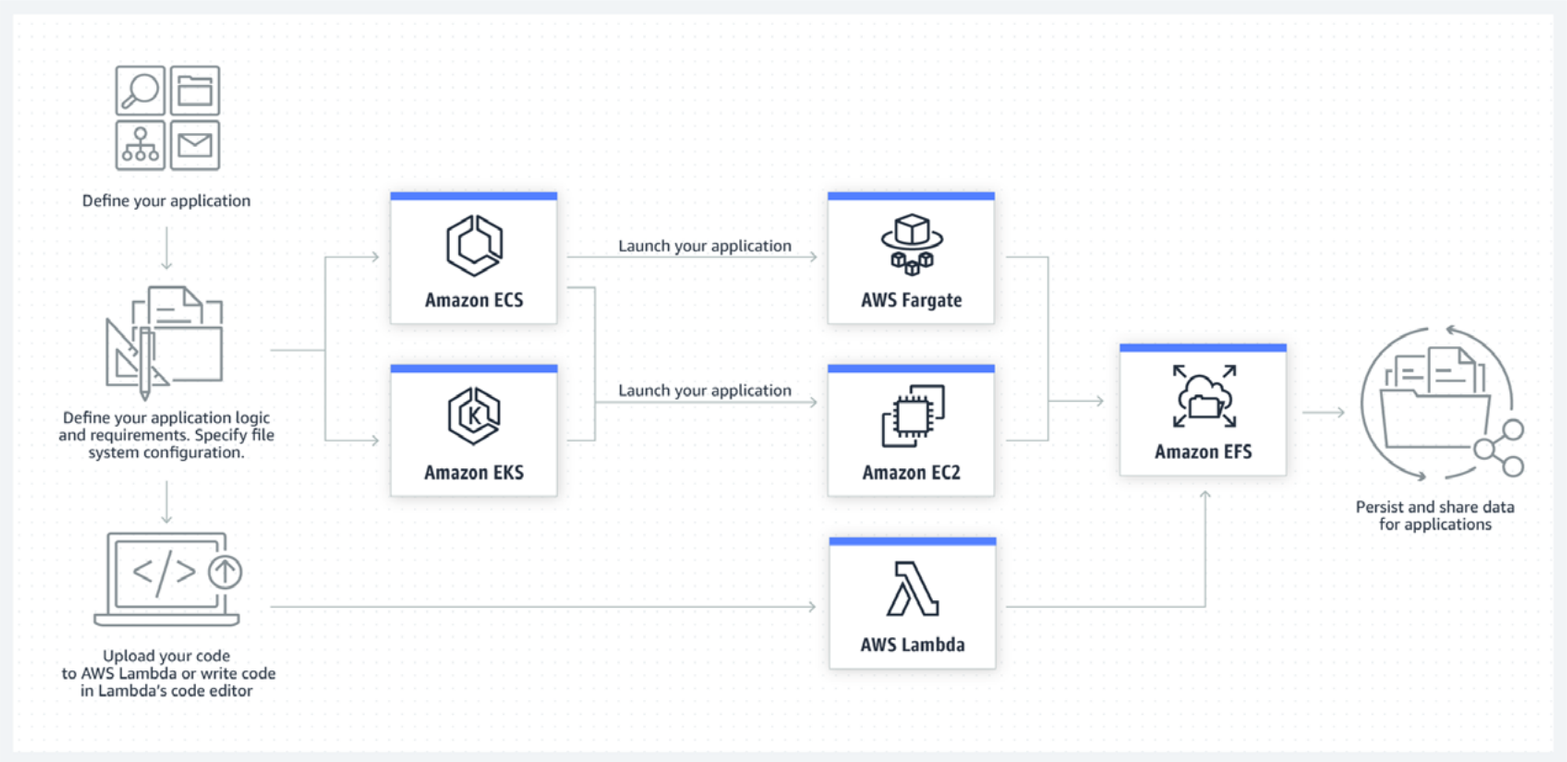
DISCOVER®

"It used to take weeks to get users the tools they needed to do their jobs. Now we can do it in hours so they can start gleaning insights and delivering value for our customers almost immediately."

Brandon Harris,
Director of Data Science Technology,
Discover Financial Services

How it works

Chart your application modernization journey.



Get started now

AWS Lambda and Amazon EFS

Use the resources below to begin your application modernization journey.

AWS Lambda and Amazon EFS
While **AWS Lambda** includes a 512 MB temporary file system for your code, this is an ephemeral scratch resource not intended for durable storage.

With Amazon EFS for AWS Lambda, you can easily share data across function invocations. You can also read large reference data files and write function output to a persistent and shared store. The blog post **Using Amazon EFS for AWS Lambda in your serverless applications** shows how to enable **Amazon EFS for Lambda** in your AWS account and walks through some common use cases.

View the video Modernizing Serverless Applications with AWS Lambda and Amazon EFS »

AWS containers and Amazon EFS

Native integrations between Amazon EFS and AWS container services simplify containerizing long-running stateful applications and applications with shared datasets.

Amazon ECS and AWS Fargate with Amazon EFS
Amazon ECS supports deploying containers on both **Amazon Elastic Compute Cloud (EC2)** and **AWS Fargate**. A three-part blog series *Developers guide to using Amazon EFS with Amazon ECS and AWS Fargate* gives you the information you need to get started: **Part 1** provides the background about the need for the integration—its scope, use cases, and the scenarios it unlocks. **Part 2** provides a deep dive on how Amazon EFS security works in container deployments based on **Amazon ECS** and AWS Fargate and considerations around regional deployments best practices. **Part 3** is a practical example, including reusable code and commands.

Amazon EKS and AWS Fargate with Amazon EFS
With **Amazon EKS**, you have the choice to run Kubernetes pods on Amazon EC2 instances or AWS Fargate. **AWS Fargate**, a serverless compute engine for containers, allows you to run Kubernetes workloads without creating and managing servers, scaling your data plane, rightsizing Amazon EC2 instances, or dealing with worker nodes upgrades. **Amazon EFS** supplies persistent, shared storage needed to run stateful Kubernetes workloads on AWS Fargate. The blog post **Running stateful workloads with Amazon EKS on AWS Fargate using Amazon EFS** shows you how to run stateful Kubernetes workloads on AWS Fargate using Amazon EFS.

Watch the demo Amazon EFS Secure data persistence with Amazon ECS and AWS Fargate »

AWS services

Amazon EFS »

Amazon EFS is a simple, serverless, set-and-forget, cloud-native file system that enables you to build modern applications and persist and share data from your AWS containers and serverless applications with zero management required.

AWS Lambda »

AWS Lambda is a serverless compute service that lets you run code without provisioning or managing servers, creating workload-aware cluster scaling logic, maintaining event integrations, and managing runtimes.

Amazon ECS »

Amazon ECS is a fully managed container orchestration service that helps you easily deploy, manage, and scale containerized applications.

Amazon EKS »

Amazon EKS gives you the flexibility to start, run, and scale Kubernetes applications in the AWS Cloud or on-premises.

AWS Fargate »

AWS Fargate is a serverless compute engine for containers that works with both Amazon ECS and Amazon EKS.

4 | Executive summary

Cloud file storage for your microservices architectures

Modern application development is enabling organizations to innovate faster, respond to customer needs, and scale to millions of users. Modern applications are built on containers and serverless architectures that are complementary to one another. These modern architectures free developers from configuration and compatibility concerns and management overhead, so they can focus on building applications.

It is now commonplace for new applications to be built with these modern techniques, and even in-place applications are being repackaged for deployment on containers and serverless architectures. Yet, there is a data problem to be addressed.

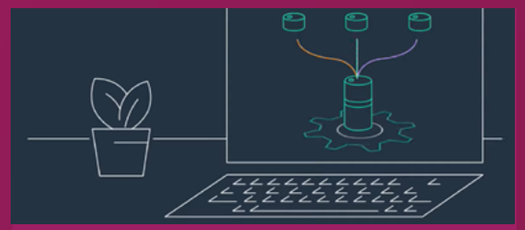
What is needed is a cloud file system that persists data, even when containerized applications and serverless functions are terminated. For these modern applications, a cloud-native file system becomes a data foundation, operating in concert with containers and serverless technologies to reliably and consistently deploy to AWS, allowing data to persist application state and share data among applications when needed.

Resources »

« Prev

Modernize your applications with AWS containers, serverless, and Amazon EFS

Here are more resources to help guide you on your application modernization journey.



Persistent File Storage for Modern Applications »

Persist and share data with zero management required

AWS Modern Applications »

Innovate faster, respond to customer needs, and scale to millions of users

Amazon Elastic File System »

Simple, serverless, set-and-forget, elastic file system

Get started with Amazon EFS in the AWS console »

Create a file system in seconds, with a few clicks in the AWS console