Build modern applications on AWS

Manage less. Build faster. Innovate more.

Modern applications are changing how you deliver customer value

Organizations worldwide are focusing on innovation, customer satisfaction, and operational efficiency as today's top business priorities. To achieve these goals, decision makers understand the need for their applications to play a key role in leading the way. In fact, 64 percent of US enterprise decision makers cite modernization as their top IT/operations priority for the next 12 months.[1] However, many companies continue to struggle with building and running their applications—with 78 percent of organizations citing a lack of technology management expertise as a top challenge.[2]

For many businesses, a move to the cloud has helped accelerate the pace of digital innovation by providing on-demand capabilities, shifting the burden of server management to the cloud provider, and reducing the total cost of ownership (TCO). But moving to the cloud is just the first step toward modernization. As business and technical leaders consider an increasingly competitive market, many are building modern applications on Amazon Web Services (AWS) to leverage the full value of the cloud.

In this eBook, we will guide you through three pathways that will help lay the foundation for modern application development in your own organization. We will also explore how modern application development with AWS can help your organization innovate, reduce costs, accelerate time to market, and improve reliability.

[1] Sustar L., et al., "Aiming For Innovation: Cloud In The US Today," June 2022
[2] "2023 State of the Cloud Report," Flexera, 2023

Modern applications empower digital innovators

Innovation means listening to your customers

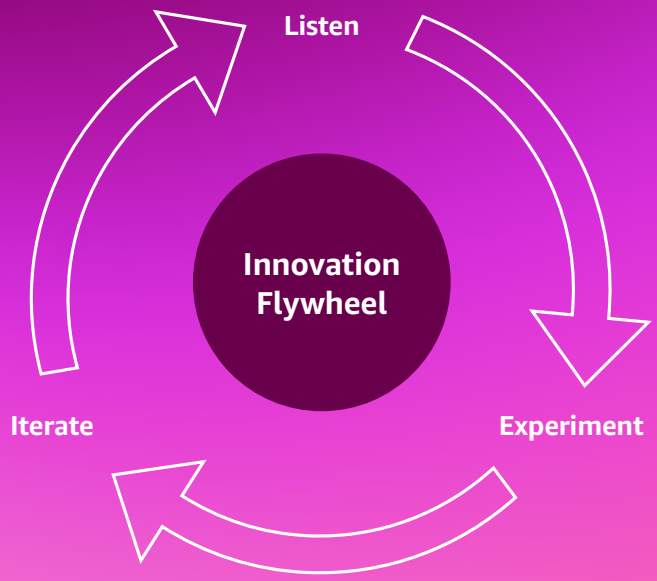
In a recent **IDC Market Perspective**, the importance of retaining a customer-centric viewpoint was referenced as a critical element in driving successful digital innovation. A digital innovator does not present a solution in search of a problem. Rather, innovation is driven by viewing the customer journey through the customer's lens and building with the goal of removing points of friction in their experience.

Focusing on your customer means making business decisions by working backwards from the customer's point of view. It means constantly evolving products and services to better deliver the outcomes that delight customers. It also means listening to what your customers truly care about so you can anticipate the experience that will keep them engaged as you continue inventing and iterating on their behalf. This approach is called the "innovation flywheel."

The basic idea of the innovation flywheel is that meaningful innovation is created out of customer demand, improves with customer feedback, and constantly repeats (and profitably) until the demand changes and the cycle begins again. Modern application development technologies and techniques help unlock your ability to iterate quickly and scale continuously. The deeper your teams adopt modern application development, the faster you can get your innovation flywheel spinning—and the sooner you can leverage innovation to differentiate your business.

"Taking advantage of new MACH-ready architectures — microservices, API, cloud, and headless — allows organizations to quickly plug in new capabilities that fit their technology stack and address opportunities of growth as a digital-first customer-centric enterprise."[3]

Marci Maddox, Research VP, Digital Experience Strategies, IDC



[3] Maddox, M., et al., "Reframing the Digital-First Customer Journey: Key Takeaways from IDC's Global CIO Advisory Board – April 2022 Edition," IDC, May 2022

40%

of firms will be adopting a cloud-native-first strategy in 2023[4]

45%

of YoY growth in cloud and service adoption[5]

82%

of organizations are struggling to manage their cloud spend[6]

Building modern applications on AWS can help you get to market faster. By accelerating the build cycle and offloading operational overhead, your developers can quickly build new features. You will increase innovation with a modular architecture that frees teams to experiment independently with individual application components without risking the entire application. By automating test procedures and monitoring at every stage of the development lifecycle, you will improve reliability. And you will improve TCO with a pay-for-value pricing model that reduces the cost of over-provisioning or paying for idle resources.

While digital transformation and the shift to modern application development can feel complex at an organizational level, the process does not need to be overly complicated. Many organizations, inspired by the promise of the cloud, move all their new application development to the cloud in a single step. Plenty of others take a hybrid approach, which involves building applications to run on premises and in the cloud. These two paths don't have to be mutually exclusive. In fact, the process typically involves parallel paths—the first, a team-by-team experience; the second, a workload-by-workload journey—both moving opportunistically, a step at a time.

[4] Sustar, L., et al., "Predictions 2023: Cloud Computing," Forrester, October 2022
[5] "Closing the cloud strategy, technology, and innovation gap: Deloitte US Future of Cloud Survey Report," Deloitte, 2022
[6] "2023 State of the Cloud Report," Flexera, 2023

3 modernization pathways that generate real business value

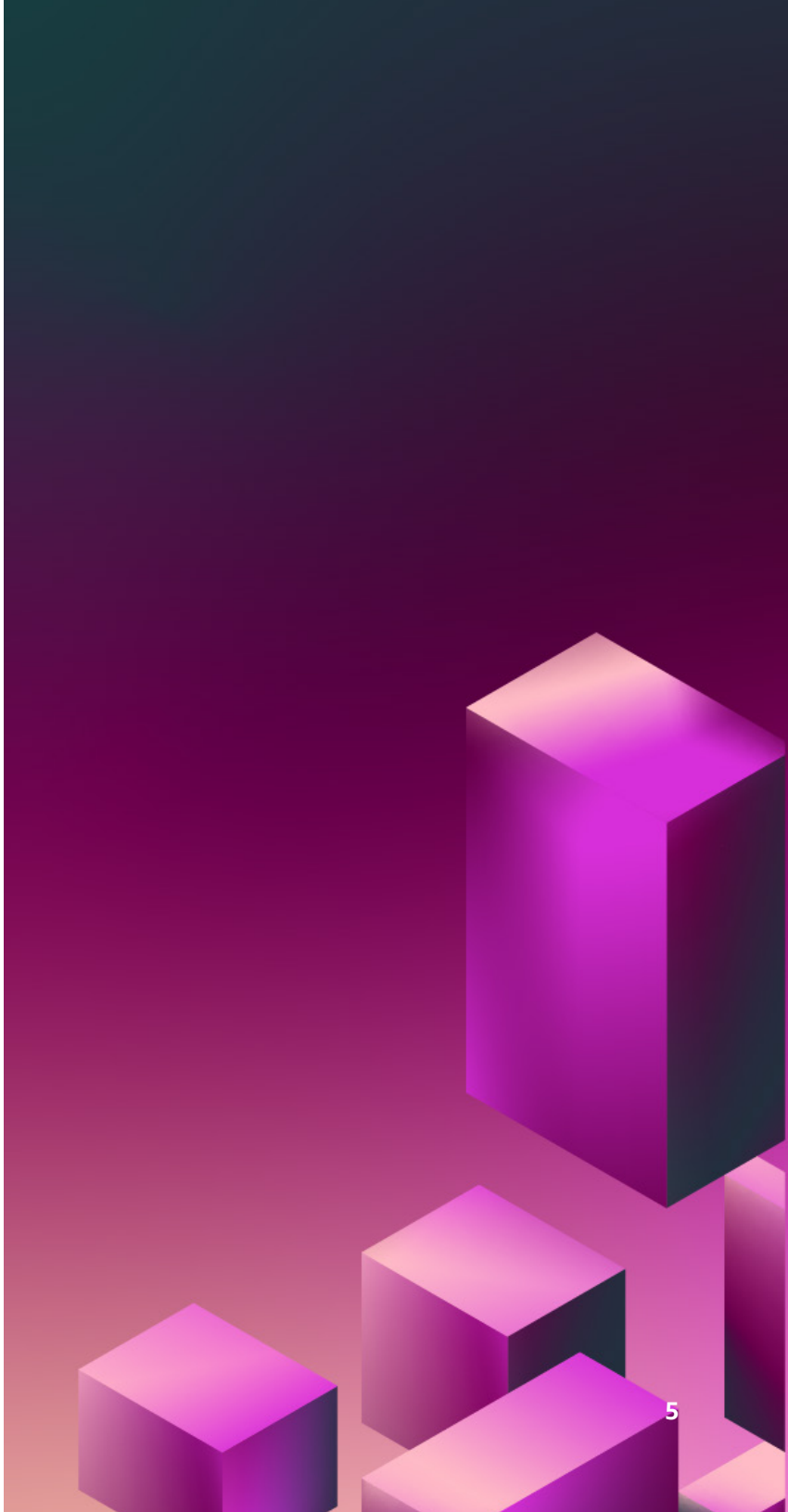
Through our experience building applications for Amazon.com—and for millions of AWS customers—we've observed three pathways customers often take to translate their vision of application modernization into a reality, generating value for their business in the process:

1. **Run managed Kubernetes on AWS and extend existing Kubernetes to AWS for hybrid distribution:** Organizations running Kubernetes in the cloud or on premises can run these workloads on AWS to increase reliability, security, and scalability. This approach provides server-level control, common operational tooling across environments, and integrated access to the vast AWS services portfolio.

2. **Build new apps with a serverless operational model:** As organizations build new applications or features, a serverless operational model provides the easiest approach for scaling and managing cloud applications. The serverless model shifts as much management of the underlying compute resources to AWS as possible, so development teams can focus on writing code that solves business problems.

3. **Transition to a modern DevOps model:** To empower the cultural shift that produces modern applications at scale, organizations can leverage DevOps services and tools. Ensure security and governance while rapidly building and delivering new products.

We will explore each pathway in more detail, demonstrating how each can help lead to increased agility, lower costs, and building better apps that support business success. While you can modernize applications from any starting point, the outcome needs to be the same: applications that are secure, reliable, scalable, and quickly available for your customers and partners from the onset of your combined efforts.



Modern applications in practice

The 3 pathways of modern application development

Modern application development is a powerful approach to designing, building, and managing software in the cloud. In practice, it increases the agility of your development teams and the reliability and security of your applications, allowing your teams to build better products faster.

We've worked with a wide range of organizations, from large, experienced operations teams looking to maintain maximum control with portable orchestration tools to resource-constrained teams preferring to offload operational tasks as they shift their focus to building and iterating on applications.

From these experiences, we've identified three solution pillars of modern application development to help you on your journey toward modernization:

1. **Run managed Kubernetes and extend existing on-premises investments**
2. **Build with a serverless operational model**
3. **The Modern DevOps model**

1 Run managed Kubernetes and extend existing on-premises investments

Today, more and more customers are taking a path of reinvention, building new applications that enable their organizations to take full advantage of the cloud. In fact, there's no single way to modernize; applications can coexist in different states (some highlighted in this eBook). Organizations worldwide are continuing to adopt and grow their use of containers as a lightweight and portable way to run and deploy many new applications. For many, containers have become the preferred route to support complex application architectures that require flexibility, rapid deployment, and continuous innovation in an environment that can run almost anywhere.

When moving to containerized applications, organizations need to consider where and how to invest their operational resources. Many development teams have strong opinions about how their applications run. Seeking to standardize the deployment, scale, security, and management of their containerized applications, these dev teams have turned to Kubernetes. Due to its vibrant open-source ecosystem, community, consistent APIs, and broad flexibility, Kubernetes has gained significant adoption.

Organizations planning to or already running Kubernetes—or thinking about building new applications on Kubernetes—can run them on **Amazon Elastic Kubernetes Service** (EKS). Amazon EKS takes full advantage of AWS's operational excellence to provide organizations with a highly scalable, reliable, and secure Kubernetes platform in the cloud. It is the best way to run Kubernetes, which is why the fastest-growing startups, largest enterprises, and security-conscious government agencies and financial institutions trust it to run their most important production workloads. According to the Cloud Native Computing Foundation (CNCF), more containers run on Amazon EKS than any other cloud. Amazon EKS runs native Kubernetes and manages cluster lifecycle and other key tasks. And it includes integrations with AWS services that enhance management and security. For example, **Amazon GuardDuty** threat detection continuously monitors and profiles container runtime activity to identify suspicious behaviors. By choosing Amazon EKS, organizations can eliminate undifferentiated heavy lifting and focus on delivering value to their end users. Additionally, Amazon EKS is customizable to support any needs. For example, it helps customers take full advantage of the broadest selection of compute choices and instance types, such as **AWS Graviton**-based instances, which enable up to 40 percent better price performance.

For on-premises or hybrid deployments, **Amazon EKS Anywhere** brings the Amazon EKS experience to your data center, allowing you to retain latency-sensitive or regulated applications in a private cloud on premises all while leveraging a common toolset across environments.

Why use Kubernetes today

Many successful modern application developers have already made the transition to Kubernetes. These developers like Kubernetes because the technology makes it easier to automate and standardize deployment, scaling, security, management, and observability of containerized applications. This helps them reduce development time, increase operational efficiency, and improve their overall reliability and security. Kubernetes enables customers to leverage the same APIs and skill sets for running infrastructure across environments across their organizations or in the cloud and on premises.

Run applications at scale

Kubernetes lets you define complex containerized applications and run them at scale across a cluster of servers.

Seamlessly move applications

Containerized applications can be seamlessly moved from local development machines to production deployments in the cloud using the same operational tooling.

Run your apps anywhere

Run highly available and scalable Kubernetes clusters on AWS while maintaining full compatibility with your Kubernetes deployments running on premises.

Add new functionality

Kubernetes is an open-source project, so adding new functionality is easy. A large community of developers and companies build extensions, integrations, and plugins that help Kubernetes users do more.

Organizations possessing in-house Kubernetes and operational expertise can run Kubernetes on AWS using Amazon EKS to leverage scalable, secure, and highly available infrastructure. Amazon EKS automatically manages the availability and scalability of the Kubernetes control plane responsible for scheduling containers, managing application availability, storing cluster data, and automating other key tasks. For organizations that have chosen to use Kubernetes, allowing AWS to operate and maintain the Kubernetes cluster has made it easier to achieve their goals.

For organizations that want to use Amazon EKS without the complexity associated with running and scaling the underlying infrastructure, **AWS Fargate** is a serverless compute environment for containers that removes the requirement for you to own, run, and manage the lifecycle of a compute infrastructure yourself. With Fargate, you can:

- Eliminate operational overhead of scaling, patching, securing, and managing servers
- Improve security through workload isolation by design (Amazon EKS pods run in their own dedicated runtime environment)
- Pay only for what you use and automatically scale to meet business demands by launching thousands of containers within minutes across worldwide **AWS Availability Zones**

CUSTOMER SNAPSHOT

New Relic

New Relic is a leader in the observability space, providing engineers with a software-as-a-service (SaaS) platform that analyzes, troubleshoots, and optimizes their entire software stack. In 2020, New Relic began a journey to transform its business, moving from a host-based pricing model to a consumption-based pricing model. In addition, the company introduced a free service tier to all customers and prepared for a significant increase in demand on its platform. New Relic made the decision to migrate its entire platform to AWS to accommodate the projected trajectory of growth. In a period of just eight months, the business leveraged its own tools and processes and AWS capabilities and offerings to migrate over 20,000 servers and to refactor its entire services platform. The refactoring of the platform utilized Amazon EKS. In this migration and refactoring program, New Relic saw material improvements in engineering efficiency and platform resiliency, along with a path for long-term scalability to support its aggressive growth targets.

Read more ›

new relic

"Consuming AWS lets us focus on our core competencies, enabling us to release better products faster and more frequently."

Andrew Harnett, VP of Engineering, New Relic

CUSTOMER SNAPSHOT

Volkswagen Group

Global automotive manufacturer **Volkswagen Group** is building augmented reality and virtual reality (AR/VR) applications to save time, reduce costs, and improve performance for use cases across the product lifecycle—from design reviews and training simulations to remote guidance in the factory. To scale AR/VR, Volkswagen needed a faster, simpler, and more efficient way to prepare and deliver 3D content. The company developed a cloud architecture on which it could accomplish two things:

1. Automate Volkswagen's 3D data preparation pipelines for higher optimization of 3D models
2. Test the ability to remotely render and stream 3D graphics to AR/VR headsets, replacing headsets tether-constrained to expensive workstations

To achieve its goals, Volkswagen is using AWS to migrate specific VR rendering and 3D data optimization workloads to the cloud. It's also using Innoactive Portal, a custom application developed alongside VR software company **Innoactive** to deploy and manage VR applications. Leveraging several AWS services, including **Amazon Elastic Container Service** (Amazon ECS)—a fully managed container orchestration service that automates the deployment, management, and scaling of containerized applications—Volkswagen is making its 3D data preparation pipeline more efficient and increasing its rendering power and speed in the cloud.

See the full story ›

VOLKSWAGEN
GROUP

"It's simple and effective for us to use AWS because there are so many native services that support our goals and architecture drafts."

Jan-Paul Brückmann, Product Owner, Volkswagen Digital Realities Hub

2 Build with a serverless operational model

When running applications in the cloud, some teams and organizations may choose to manage the cloud infrastructure themselves. However, this means they need to invest resources in cloud administration tasks to ensure that their infrastructure is provisioned, maintained, upgraded, patched, load balanced, secure, available, and scaled based on application needs and customer demands during the build and run phases of their application. For teams looking for faster ways to turn ideas into modern production applications and reduce ongoing infrastructure maintenance in production, these tasks are operational overhead and slow their pace of innovation. They need to build applications rapidly with the ability to scale quickly, have global availability, manage large amounts of data, and respond in milliseconds, all while lowering total costs and maintaining robust security postures.

A serverless operational model alleviates this challenge and provides the easiest model for scaling and managing cloud applications by shifting as much management of the underlying compute resources to the cloud provider as possible so development teams can focus on writing code that solves business problems. With AWS serverless solutions, you leverage AWS best practices and expertise on performance, scalability, availability, and security as AWS takes on tasks such as server management, resource allocation, and scaling on your behalf. With those tasks taken care of, development teams can accelerate time to production while lowering TCOs by paying for only what is used and focusing valuable human resources on innovation.

Key considerations for building scalable modern apps

Architectural patterns: microservices

Monolithic apps can be easy to manage at first, but as your app begins to change and evolve, this growth can create challenges for scalability and agility, including how to distribute ownership of the app across your teams. You can build a strong culture of ownership but still struggle to scale up if your application architecture includes hard dependencies that prevent teams from taking ownership of the final product. This is why we often recommend building microservices architectures for applications that grow and change rapidly. Microservices are the architectural expression of a culture of ownership—they neatly divide complex applications into components that a single team can own and run independently.

With a monolith, you have many developers all pushing changes through a shared release pipeline, which causes friction at many points of the lifecycle. Upfront, during development, engineers need to coordinate their changes to make sure they're not breaking someone else's code. To upgrade a shared library to take advantage of a new feature, you need to convince everyone else to upgrade at the same time—a tough ask! And if you want to quickly push an important fix for your feature, you still need to merge it with changes in progress.

After development, you also face overhead when you're pushing the changes through the delivery pipeline. Even when making a one-line change in a tiny piece of code, engineers need to coordinate their changes ahead of time, merge their code, resolve conflicts within releases, rebuild the entire app, run all of the test suites, and redeploy once again.

Monolith apps

Do everything the app requires in a single self-contained unit

- Single app
- Must deploy entire app
- One database
- Organized around technology layers
- State in each runtime instance
- One technology stack for entire app

Microservices

Decompose the app into smaller, loosely coupled components, each responsible for a specific capability

- Each component does one thing
- Minimal function services
- Deployed separately, interact together
- Each has its own database
- Organized around business capabilities
- State is externalized
- Choice of technology for each microservice

Agility, scalability, and resilience

With a microservices architecture, applications are composed of independent components that run each application process as a service. Services are built for business capabilities, and each service performs a focused function. Because services run independently, each can be updated, deployed, and scaled to meet the demand for specific functions of an application. For example, an online shopping cart built as a microservice can be used by many more users during a sale as it independently scales to support demand. Microservices communicate data with each other via well-defined interfaces using lightweight APIs, events, or streams. Our customers are increasingly relying on event-driven architectures (EDAs)—those in which actions are triggered in response to changes in data—to improve scalability and resiliency while also reducing costs.

CUSTOMER SNAPSHOT

Bosch Thermotechnology North America

Bosch Thermotechnology North America (Bosch TTNA) built a smart source of heating, ventilating, and air-conditioning (HVAC) systems by modernizing and migrating its business to AWS. The new solution monitors products remotely while removing the undifferentiated heavy lifting of infrastructure management. Bosch TTNA was new to smart device development and wanted a cost-effective solution to expand its infrastructure capacity and scalability while creating new smart technologies. The company used AWS to build solutions to connect its devices to **AWS Internet of Things** (AWS IoT). The solution uses AWS serverless technologies for data processing, application integration, and the scaling required to manage its business. Bosch TTNA can now improve customer service by remotely monitoring its new smart energy and building devices with minimal operational overhead.

See the full story ›

BOSCH

"We use AWS to achieve our business goals and to innovate in the technology space. Using AWS, we accelerate the change that we're driving."

Pablo Ferreyra, Head of Software Development for Bosch CI Americas, Bosch Thermotechnology North America

CUSTOMER SNAPSHOT

Liberty Mutual

With $40 billion in annual revenue, **Liberty Mutual** is the world's sixth-largest property and casualty insurance company and is an industry front-runner in technological innovation. To achieve its goal of becoming a global digital company, the insurance provider focused on three main areas of transformation: customer centricity, agility, and cloud-native development. To that end, the company made a strategic decision to pursue a serverless-first approach—a move designed to give it an edge in its competitive global market. Liberty Mutual used AWS to migrate its on-premises systems to the cloud to modernize and drive business-wide transformation. By using serverless architecture and letting AWS handle infrastructure management tasks like capacity provisioning and patching, Liberty Mutual reduced its operational burden and realized substantial cost savings. The company also used serverless solutions so that it could rapidly build more agile, high-quality applications. By eliminating the operational overhead, the serverless architecture facilitates experimentation, empowering Liberty Mutual's teams to release quickly, get feedback, and iterate to get to market faster.

Read more ›

Liberty Mutual INSURANCE

"Our collaboration to figure things out feels like more than a customer-vendor relationship. It genuinely feels like AWS is part of our team."

Dave Anderson, Former Director of Technology, Liberty Mutual

As serverless as possible

As your architectural patterns and software delivery processes change, you will probably want to adopt an operational model that enables you to offload any activity that isn't a core competency of your business. To gain agility for rapid innovation, we recommend building microservices architecture and operating and deploying software using automation for things like monitoring, provisioning, cost management, deployment, security, and governance of apps. Choosing a serverless operational strategy—opting for serverless technologies wherever possible—enables you to maximize the operational benefits of AWS. Whether you are building net-new apps or migrating legacy software, choosing to build with serverless primitives for compute, data, analytics, and integration will enable you to benefit from the most agility the cloud has to offer.

How do we define serverless at AWS?

When we say serverless, we mean the removal of the undifferentiated heavy lifting that is server operations. This is an important distinction because it allows you to focus on building the application rather than managing and scaling the infrastructure to support the application. The tenets of a serverless operational model are:

1. **No server management** – There is no need to provision, operate and patch, or maintain any servers. There is no software or runtime to install, maintain, or administer.
2. **Flexible scaling** – Your application can be scaled automatically or by adjusting its capacity through toggling the units of consumption (throughput, memory) rather than units of individual servers.
3. **Pay for value** – Instead of paying for server units, pay for what you value—consistent throughput or execution duration.
4. **Automated high availability** – Serverless provides built-in availability and fault tolerance. You don't need to architect for these capabilities because the services running the application provide them by default.
5. **Native integration** – Built-in service integrations with AWS solutions, third-party SaaS solutions, and the ability to connect with first-party custom-built applications make it easier to access and use the resources necessary to build your applications.
6. **Secure environment** – Serverless offers built-in security best practices that natively integrate with AWS's identity and governance tools that you already trust with compliance to SOC, HIPAA, PCI, and ISO.

A serverless operational model is ideal for high-growth companies that need to innovate quickly. Serverless enables your teams to move even faster while keeping a laser focus on the activities that differentiate your business—all so you can accelerate your innovation flywheel.

Build with serverless containers

Leveraging AWS Lambda and Amazon ECS with Fargate

With the rise of containers and serverless computing, AWS continues to provide the broadest and deepest offerings for compute to ensure you have the right compute option to fit your needs. Choosing the optimal compute for your modern application starts with exploring several questions. Does self-managing infrastructure improve your business results? Do you have the expertise to do it? And will the extra effort ultimately drive value?

Increasingly, customers are choosing to offload server management by adopting serverless container services like Amazon ECS on Fargate or just running code with serverless function services like **AWS Lambda**.

Most customers use a combination of services. About 80 percent of AWS container customers have also adopted Lambda.[7] Leveraging both options has its benefits, including fully managed services that have deep integration with AWS infrastructure, support for a wide range of use cases, abstraction from complexity, and a broad ecosystem of partners.

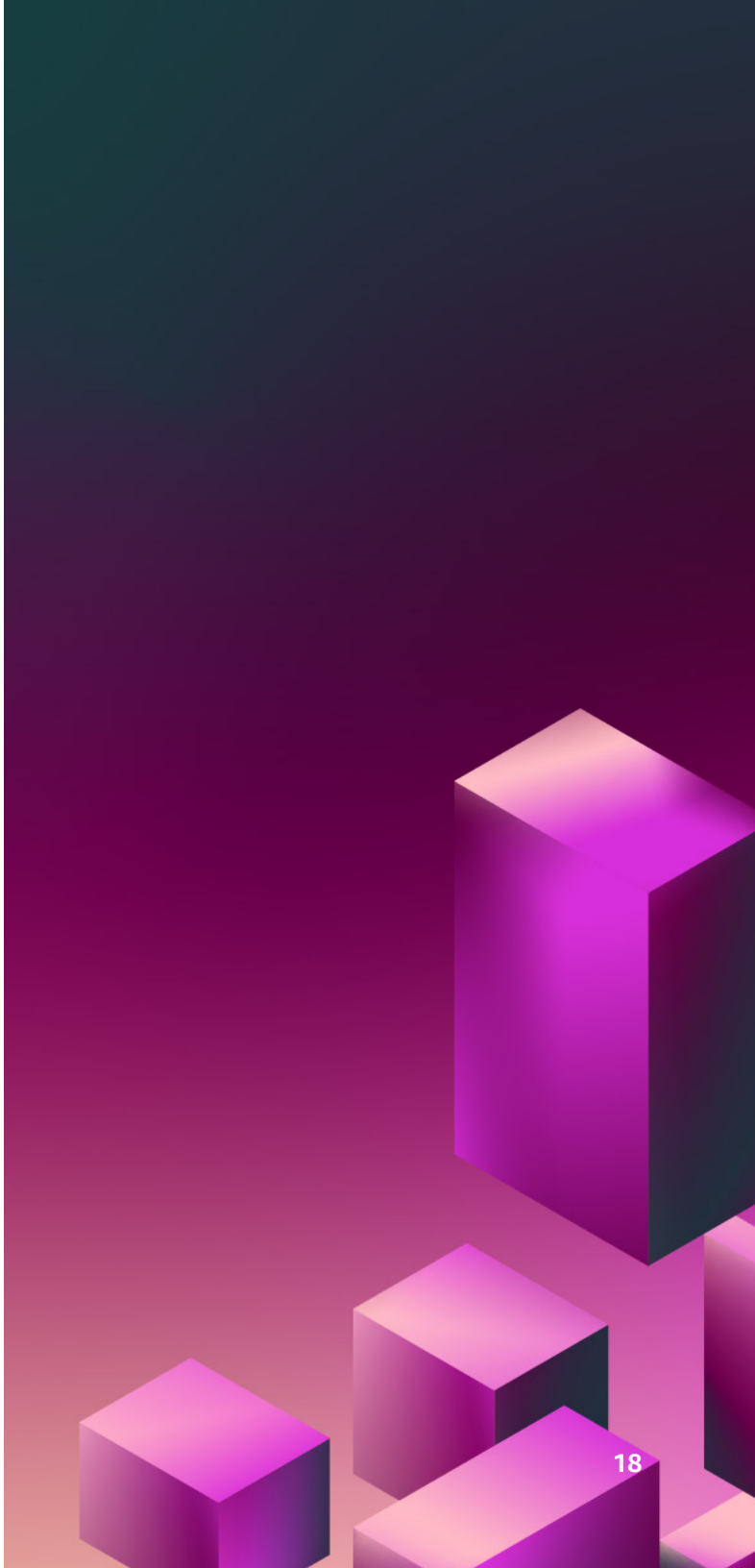
[7] "The State of Serverless," DataDog, June 2022

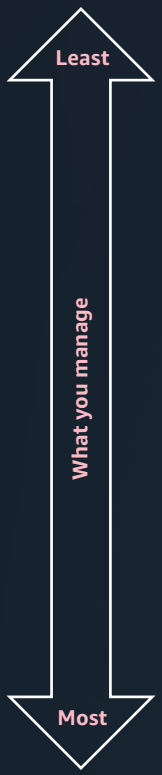
So how do you frame the decision?

Customers choose Lambda when they have teams focused primarily on writing code and no limitations on existing instances or container platforms. Lambda can scale automatically from zero to millions of simultaneous processes in real time with a built-in security posture. With Lambda, you can take full advantage of the most trusted, reliable, and secure cloud provider. Lambda manages almost all server management tasks on your behalf, including hardware, software, networking and facilities management, data source integrations, and provisioning. That's why, every month, Lambda has over a million monthly active customers who generate trillions of invocations.

Customers often choose Amazon ECS on Fargate when they want to leverage containers and container orchestrators for development but do not need to manage servers or clusters. Amazon ECS delivers the easiest way for organizations to deploy and manage containerized applications at any scale.

With Fargate, Amazon ECS supports serverless container orchestration, so you can leverage more of the operational excellence of AWS when it comes to scaling, maintaining availability, and securing your containerized workloads. Amazon ECS provides tooling and built-in support that makes it simple to build and run containerized applications on AWS. When more control over the characteristics of how your applications run is desired, Amazon ECS on **Amazon Elastic Compute Cloud** (Amazon EC2) is available, as well as **Amazon ECS Anywhere** for when you want to run container workloads on your infrastructure. It's no surprise that over 65 percent of all new AWS container customers use the fully managed and versionless Amazon ECS.





	AWS Manages	Customer Manages	
AWS Lambda Serverless functions	Data source integrations Physical hardware, software, networking, facilities Provisioning	Application code	
AWS Fargate Serverless containers	Container orchestration control plane and provisioning Cluster scaling Physical hardware, host OS/kernel, networking, facilities	Application code Data source integrations Security config and updates, network config	Management tasks
Amazon ECS/ Amazon EKS Container management as a service	Container orchestration control plane Physical hardware, networking, facilities	Application code Data source integrations Security config and updates, network config, firewall	Work clusters Management tasks
Amazon EC2 Infrastructure as a service	Physical hardware, networking, facilities	Application code Data source integrations Scaling Management tasks	Security config and updates, network config Provisioning, managing scaling, and patching of servers

Snoop

Snoop is a UK-based fintech startup that helps customers cut their bills, pay off debt, grow their savings, and save where they spend, all without changing banks. The company achieves this by harnessing open banking data to provide hyper-personalized financial insights on its cloud-based app in seconds. Working with lean resources, Snoop's co-founders looked to AWS for solutions to hasten time to market and build an app that remains secure, highly scalable, and available 24/7. By leveraging Amazon ECS with Fargate, Snoop scaled from zero to one billion transactions in just two years, optimizing costs and reducing overhead.

See the full story ›

"All of our Amazon ECS instances use AWS Fargate, which takes off a huge piece of overhead. As a fast-scaling startup, that's exactly what we need."

Jamie West, Senior DevOps Engineer, Snoop

Futuready

Futuready, an online insurance broker operating in Indonesia and Thailand, migrated to the AWS Cloud, seeking to improve its reliability and gain the agility to modernize its monolithic architecture. The company redesigned its architecture using Amazon ECS with Fargate, which helped expand its business regionally without additional IT investments. Futuready also reduced its infrastructure and operations costs by 35 percent and has accelerated its deployment cycle for new insurance providers and products. The time taken to onboard new insurance partners was reduced from six weeks to only two weeks, and the team can now release new features weekly instead of biweekly. Additionally, Furtuready leverages Lambda to automate the process of completing insurance documentation, helping to reduce the amount of time customers spend filling in their account details when purchasing a policy.

See the full story ›

"We can move faster and have gained 99.999% uptime with AWS and its range of available managed services. This leads to higher revenues by running more projects and closing deals more than 50% faster."

Dr. Nicholas Ruwan Dias, CTO, Futuready

Taco Bell

As one of America's most iconic food brands, **Taco Bell** operates over 7,000 restaurants in the United States. During the COVID-19 pandemic, Taco Bell needed to rapidly shift to meet consumer demand for delivery. The company runs nearly all of its infrastructure on AWS and uses serverless on AWS to focus less on managing servers and more on building business logic and data transformations to deliver real-time menu and restaurant information to its delivery partners.

Watch the video ›

"We have a menu that is very complex and (it) has to be shared across multiple digital channels. Serverless fits that model really well."

Vadim Parizher, VP of Technology, Taco Bell

Coca-Cola

When the COVID-19 pandemic hit in 2020, consumer habits changed literally overnight. Coca-Cola responded rapidly with a no-touch drink dispensing experience to go along with its innovative Freestyle drink dispensers. Coca-Cola opted to build with Lambda, and as a result, its team was able to focus on the application rather than security, latency, or scalability. (With Lambda, it's all built in.) The new application launched in just 100 days, and now more than 52,000 machines have the touchless capability.

Watch the video ›

"Low latency is essential to the user experience, which is why we're committed to a serverless solution on AWS."

Michael Connor, Former Chief Architect, Coca-Cola Freestyle Equipment Innovation Center

3 The Modern DevOps model

Modern DevOps is the combination of cultural philosophies, practices, and tools that enable an organization to quickly and safely develop software, release it to production, and maintain its target availability and performance.

AWS has identified a set of common, broadly accepted practices that, when adopted, provides a mechanism for building a high-performing DevOps organization. This approach takes a simple idea—continuous improvement—and applies it to everything in the DevOps lifecycle, from planning and code writing to deployment and monitoring. We call this approach modern DevOps, and it's centered around bringing developers and operations closer by sharing operational tasks like compliance, observability, resilience, and infrastructure earlier in the development process and enhancing it with artificial intelligence (AI) and machine learning (ML).

Developer agility: abstraction, automation, and standardization

Microservices architectures make teams agile and enable them to move faster, which means you're building more things that need to get released—great! However, you won't get new features to your customers faster if your build-and-release process does not keep pace. Traditional development processes and release pipelines are slowed mainly by manual processes and custom code. Custom code is ultimately a long-term liability because it introduces the possibility of errors and ongoing maintenance. Manual steps—from code changes and build requests to testing and deploying—are the greatest drags on release velocity. The solution involves abstraction, automation, and standardization.

To speed the development process, abstract as much code as possible, particularly the lines of non-business logic code required to develop and deliver production-ready apps. One way to do this is by employing frameworks and tooling that reduce the complexity of provisioning and configuring resources. This gives developers the ability to move quickly while also enforcing best practices for security, privacy, reliability, performance, observability, and extensibility throughout the development process. Development frameworks give you confidence that your architecture will support your business growth long term.

By defining your software delivery process with best-practice templates, you can provide a standard for modeling and provisioning all infrastructure resources in a cloud environment. These infrastructure-as-code (IaC) templates help teams get off to a good start because the templates provision the entire technology stack for an application through code rather than using a manual process.

Through automation, you can create a repeatable motion that speeds up your software delivery lifecycle. Automating the release pipeline through continuous integration and continuous delivery (CI/CD) helps teams release high-quality code faster and more often. Teams that practice CI/CD ship more code, do it faster, and respond to issues quicker. In fact, according to the 2020 Puppet State of DevOps Report, teams that employ CI/CD have a failure rate that is five times lower, a commit-to-deploy rate that is 440 times faster, and a rate of deployment that is 46 times more frequent. Most notably, teams that practice CI/CD spend 44 percent more of their time creating new features and code instead of managing processes and tools.[8]

CI/CD pipelines have become the new factory floor for building modern applications. At Amazon, we started using CI/CD to increase release velocity, and the results were dramatic—achieving millions of deployments per year and growing faster every year. To help companies benefit from our experience, we built a suite of developer tools based on the tools we use internally, so our customers can deliver code faster.

A bit more detail

Continuous integration (CI) is a software development practice in which developers regularly merge their code changes into a central repository, then automated builds and tests are run. CI most often refers to the build or integration stage of the software release process and entails both an automation component (a CI or build service, for example) and a cultural component (learning to integrate frequently, for example).

Continuous delivery (CD) is a software development practice in which code changes are automatically prepared for a release to production. CD expands on CI by deploying all code changes to a testing environment or a production environment after the build stage.

Learn how Amazon automates safe, hands-off deployments ›

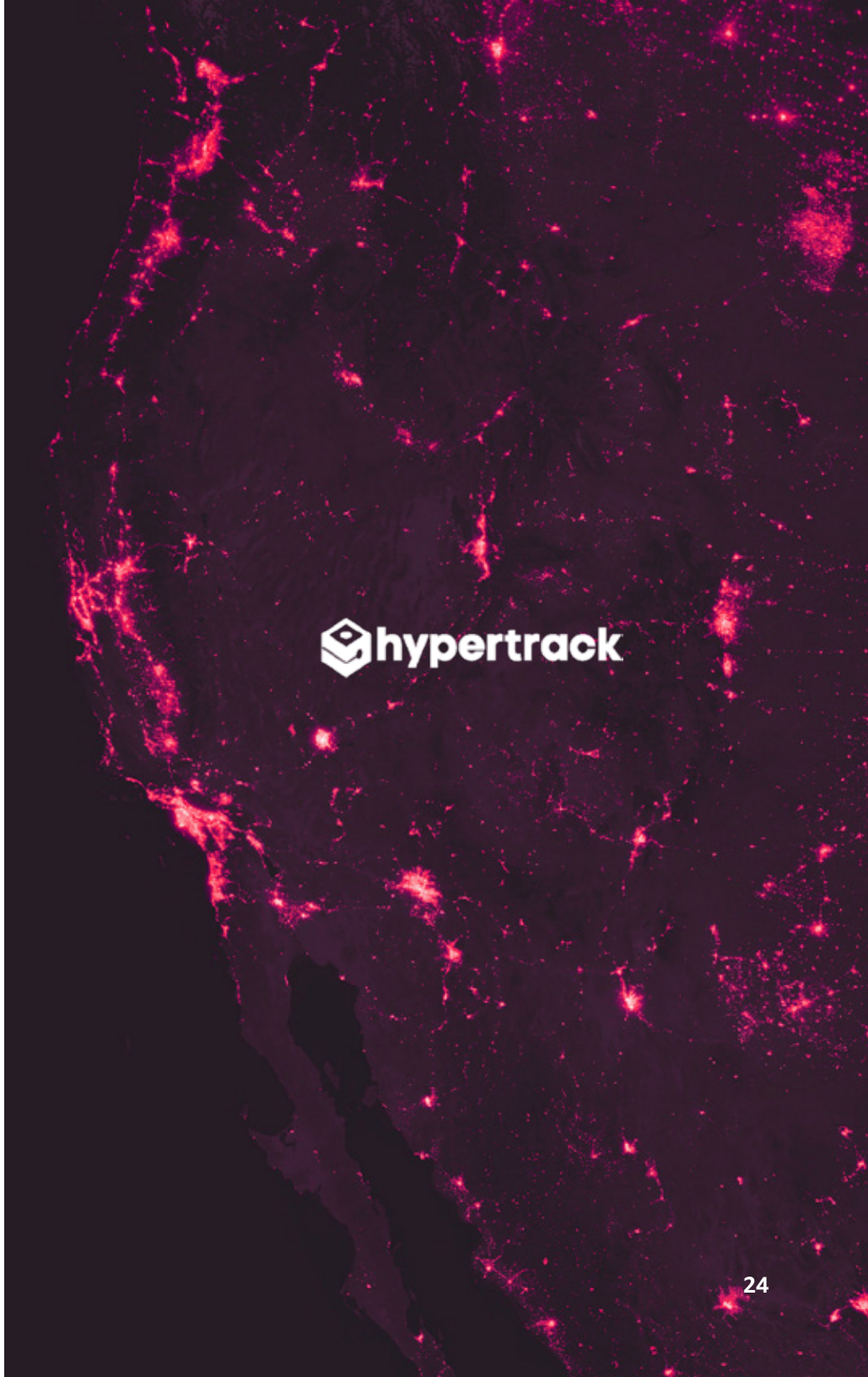
[8] "2020 State of DevOps Report," Puppet, 2020

CUSTOMER SNAPSHOT

HyperTrack

HyperTrack is a self-serve cloud platform for live location tracking through applications. When it launched, the company needed to build a platform that could scale automatically to meet its anticipated growth without reducing the time its developers spent on building new features. HyperTrack opted to use **AWS Amplify** for a mobile development framework and a serverless architecture, allowing it to scale up and down automatically without engineering intervention. As a result, the company has realized a 30 percent cost savings compared to its previous architecture. A big part of that savings comes from the elimination of operational resources for server management; HyperTrack saves 40 hours of work every week while managing millions of events.

See the full story ›



Transform to a Modern DevOps model

Creating a culture of ownership: Manage less, innovate more with modern DevOps

Ultimately, innovation comes from people. So, enabling your people to deliver better customer outcomes is where modern application development starts. At AWS, we use the concept of "products, not projects" to describe how this cultural mindset impacts team structure. Simply stated, it means that the teams that build products are responsible for running and maintaining them. Such a mindset makes product teams accountable for the development of the whole product, not just a piece of it.

After more than a decade of building and running the highly scalable web application that is **Amazon.com**, we understand the importance of giving autonomy to our teams. When we gave our teams ownership of the complete application lifecycle—from taking customer input and planning the road map to developing and operating the application—they became owners, fully empowered to develop and deliver customer outcomes. Autonomy creates motivation, opens the door for creativity, and develops a risk-taking ethos in an environment of trust.

While not inherently technical, embracing a culture of ownership remains one of the most challenging aspects of modern application development. Empowering teams to become product owners involves changing the motivation of your organization, the structure of your teams, and the work for which teams are responsible. Knowing firsthand how difficult yet impactful this transition can be has led AWS DevOps services team to make everything it does about accelerating our customers through it.

That design philosophy culminated in **Amazon CodeCatalyst**, a complete software development service that brings together everything teams need to plan, code, build, test, and deploy applications on AWS.

There are four foundational elements of CodeCatalyst that accelerate the transition to DevOps:

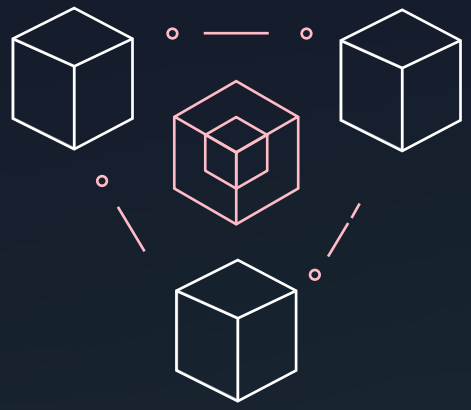
1. Project blueprints that automatically set up everything you need to start a new project
2. Flexible CI/CD automation for managing day-to-day software lifecycle tasks
3. On-demand cloud dev environments for a consistent build experience
4. Project and issue management for streamlined team collaboration

One of the most difficult aspects of the move to DevOps culture is that teams must spend a great deal of time and effort constructing and maintaining bespoke software platforms out of disparate tools, which can lead to frustration. Software teams that use CodeCatalyst get pre-integrated code repositories, cloud infrastructure, DevOps tooling, and collaboration tooling, all delivered as a service they can quickly weave into the fabric of their day-to-day work. CodeCatalyst does all of this in a flexible system that can grow with them: It combines an opinionated experience that works out of the box and provides tooling modularity, so you can integrate or swap out tools as needed. By drastically reducing the amount of time and effort software teams spend configuring software platforms, CodeCatalyst has the ability to turn IT from a cost that is necessary to support business growth to a strategically competitive advantage that organizations cannot live without.

Contino

Contino helps customers migrate their applications to the cloud, improving their architectures by taking full advantage of cloud-native features to enhance agility, performance, and scalability. Contino is using **CodeCatalyst** to bring together everything its software development teams need to plan, code, build, test, and deploy applications on AWS. Providing a streamlined, integrated experience, CodeCatalyst has helped Contino improve collaboration and increase productivity, allowing its engineers to focus more time on rapidly developing application code that captures customer value.

See the full story ›



Building a culture of innovation

1. Start with customers – Every innovation should start with a customer need and, ultimately, lead to delighting your customers. Relentlessly prioritize customer demand.

2. Hire application builders and let them build – Remove any obstacles that slow the process of building and releasing products and features for customers. The faster you iterate, the faster your innovation flywheel spins.

3. Support application builders with a belief system – Don't pay lip service to innovation. Live and breathe innovation in all areas of the business, from leadership to sales to support.

Accelerate development with generative AI-powered tools

Part of enabling your developers to build better applications that deliver exceptional customer experiences is making use of tools that can improve productivity and free up more time for your team to innovate. It is very common for software developers to spend a significant amount of their time writing code that is pretty straightforward and undifferentiated. They also spend a lot of time trying to keep up with a complex and ever-changing tool and technology landscape. All of this leaves developers less time to develop new, innovative capabilities and services. Developers try to overcome this by copying and modifying code snippets from the web, which can result in inadvertently copying code that doesn't work, contains security vulnerabilities, or doesn't track the usage of open-source software. And ultimately, searching and copying still take time away from the good stuff.

Using tools that leverage capabilities like generative AI can take this heavy lifting out of the equation by "writing" much of the undifferentiated code, allowing developers to build faster while freeing them up to focus on the more creative aspects of coding. **Amazon CodeWhisperer**, an AI coding companion, radically improves developer productivity by generating code suggestions in real time based on developers' comments in natural language and prior code in their integrated development environment (IDE). Developers can simply tell CodeWhisperer to do a task such as "parse a CSV string of songs" and ask it to return a structured list based on values such as artist, title, and highest chart rank. CodeWhisperer provides a productivity boost by generating an entire function that parses the string and returns the list as specified. AWS ran a productivity challenge, and participants who used CodeWhisperer completed tasks 57 percent faster, on average, and were 27 percent more likely to complete them successfully than those who didn't use CodeWhisperer. This is a giant leap forward in developer productivity, and we believe this is only the beginning.

Manage less, innovate more

Modern applications help create competitive differentiation by enabling rapid innovation. By adopting services, practices, and strategies that prioritize speed and agility, you can shift resources from business-as-usual operations to differentiating activities with deep customer value. Experiment more and turn ideas into releases faster. Nurture an environment where builders spend more time building and less time managing. Modern applications are how organizations, including Amazon, innovate with speed and agility.

Why build modern applications on AWS?

Faster to market	By speeding up the build-and-release cycle and offloading operational overhead, developers can quickly build new features. Automated test-and-release processes reduce error rates, so products are market-ready faster.	**See the proof:** Urbanbase launches services 20x faster with AWS ›
Increase innovation	With a modular architecture, changes to any individual application component can be made quickly and with a lower risk to the whole application, so teams can experiment with new ideas more often.	**See the proof:** iRobot uses Lambda and the AWS IoT platform to manage its Roomba robotic vacuum cleaners ›
Improve reliability	By automating test procedures and monitoring every stage of the dev lifecycle, modern apps provide greater reliability at deployment. Any issues are evaluated and addressed in real time.	**See the proof:** Siemens decreases customer control system alerts by 90% and reduces infrastructure costs by 85% ›
Improve TCO	With a pay-for-value pricing model, modern applications reduce the cost of over-provisioning or paying for idle resources. Additional maintenance savings can be achieved by offloading infrastructure management.	**See the proof:** Save up to 80% on app maintenance with Lambda ›

Learn more about modern application development

Re-platform to managed container services

More than 80 percent of organizations today are using containers.[9] More containerized applications running in the cloud run on AWS than anywhere else.[10]

Resources

Amazon ECS workshop

Amazon EKS workshop

Recommended training (Classroom)

Running Containers on Amazon EKS

Recommended training (Online)

Amazon ECS Primer

Build new modern applications with serverless technologies and tools

Save up to 80 percent of the time spent on maintenance and approximately 70 percent on development when adopting a serverless strategy for building modern applications.[11]

Resources

Innovator Island – Serverless web application development workshop

Build a serverless app video tutorial

Recommended training (Classroom)

Advanced Developing on AWS

Recommended training (Online)

Architecting Serverless Solutions

Transform to a Modern DevOps model

By 2027, AI will dramatically increase developer velocity by automatically generating code to meet functional business requirements for 80 percent of new digital solutions in development and early deployment.[12]

Resources

The Amazon Builders' Library

Recommended training (Classroom)

DevOps Engineering on AWS

Recommended training (Online)

Getting started with DevOps on AWS

[9] "CNCF Annual Survey," Cloud Native Computing Foundation, 2022

[10] "2023 State of the Cloud Report," Flexera, 2023

[11] "Guidebook: Containers and Kubernetes on AWS," Nucleus Research, October 2019

[12] "IDC FutureScape Webcast: Worldwide Developer and DevOps 2023 Predictions," IDC, December 2022

Need more help?

Learn more about building modern applications on AWS ›

Talk to an expert to implement the best practices of modern application development ›

Connect with an AWS Partner to accelerate your modernization projects ›