The background features several large, organic, wavy shapes in shades of orange and pink. One large pink shape is in the top right, another orange shape is in the top left, and a large orange shape is in the bottom right. A pink shape is also in the bottom left. The text is centered in the white space between these shapes.

An UX Case Study

# **Enhancing Safety for Field Workers in Oil Industry**



# Modern applications are changing how you deliver customer value

Businesses worldwide are focusing on innovation, customer satisfaction, and operational efficiency as top priorities. For software companies, this means finding new ways to reduce costs, drive productivity and speed up the development process. Modernization will play a pivotal role in helping developers increase efficiency, shorten time to market, and keep pace with a global ISV market set to be worth \$4.07 billion by 2027.

For many businesses, a move to the cloud has helped increase the pace of digital innovation by providing on-demand capabilities, shifting the burden of server management to the cloud provider, and reducing overall infrastructure costs. Moving to the cloud is just the first step toward modernization. In an increasingly competitive market, forward-thinking software developers rely on Amazon Web Services (AWS) to leverage the full value of the cloud and software-as-a-service (SaaS) to deliver modernized applications.

In this eBook, we'll guide you through three routes for modern application development in your business. We'll also explore how you can work with AWS to innovate, bring new products to market faster, reduce costs, and improve flexibility, security, and reliability.



# Modern apps empower digital innovators

## Innovation means listening to your customers

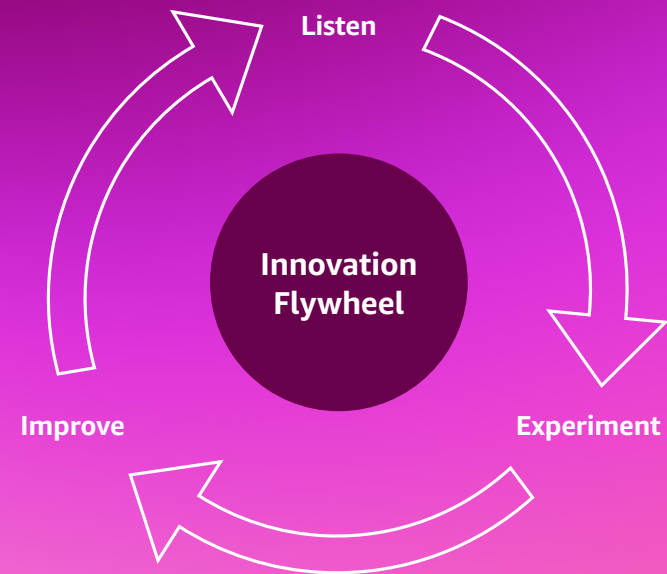
In a recent **IDC Market Perspective**, the importance of retaining a customer-centric viewpoint was referenced as a critical element in driving successful digital innovation. A digital innovator does not present a solution in search of a problem. Rather, innovation is driven by viewing the customer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

Truly focusing on your customer means making business decisions by working backwards from their point of view. It means constantly evolving products and services to better deliver outcomes that will delight. 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 known as the innovation flywheel.

The basic idea of the flywheel is that meaningful innovation is created out of customer demand, improves with customer feedback, and repeats 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 more your teams adopt modern application development, the faster you can get your flywheel spinning and 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."**<sup>1</sup>

Marci Maddox, Research VP, Digital Experience Strategies, IDC





# 40%

of firms will adopt  
a cloud-native-first  
strategy from  
2023 onwards<sup>2</sup>

# 45%

of YoY growth in cloud  
and service adoption<sup>3</sup>

# 82%

of organizations are  
struggling to manage  
their cloud spend<sup>4</sup>

Building modern applications on AWS can help you **get to market faster**. By accelerating the build cycle and increasing productivity, your developers can create and deploy new features at speed. With on-demand access to scalable compute, storage, and networking resources across your applications, you can quickly adjust capacity to meet demand.

Automating test procedures and monitoring at every stage of the development lifecycle can also help to **improve reliability while a pay-for-value model reduces** the cost of over-provisioning or paying for idle resources.

While many companies move all new app development to the cloud in a single step, plenty of others take a hybrid on-premises and cloud approach. This 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.

# Three modernization pathways to generate real business value

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

## **1. Increase reliability, security, and scalability by running managed Kubernetes on AWS**

Extending existing Kubernetes to AWS for hybrid distribution provides server-level control, common operational tooling across environments, and integrated access to the vast AWS portfolio.

## **2. Go serverless to simplify management and scale with ease**

A 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. Produce modern applications at scale with a DevOps model**

Leverage DevOps services and tools to ensure security and governance while rapidly building and delivering new products.

We’ll explore each pathway in more detail, demonstrating how they can lead to increased agility, lower costs, and better apps. 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.

# The Modern DevOps model

## The three 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.

Based on our experience helping software developers of all sizes, we've identified three solution pillars of modern application development to guide you on your journey toward modernization:

- 1 Run managed Kubernetes and extend existing on-premises investments**
- 2 Build with a serverless operational model**
- 3 Transform to a modern DevOps model**

# Run managed Kubernetes and extend existing on-premises investments

Today, more and more software developers are reinventing the way they work and building new applications to take full advantage of the cloud. As part of this process, many are growing their use of containers as a lightweight and portable way to run and deploy new applications. In fact, 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, you need to consider where and how to invest your resources. Dev teams looking to standardize the deployment, scale, security, and management of containerized applications are turning to Kubernetes for its vibrant open-source ecosystem, community, consistent APIs, and broad flexibility.

Kubernetes is a powerful container orchestration technology that allows you to deploy and manage containerized applications at scale. Due to its extensible nature and the large number of design choices available, building a tailored Kubernetes cluster can be a long and time-consuming process. Until now, a deep knowledge of the system has been required to connect a wide range of open-source tools.

To simplify this process, **Amazon Elastic Kubernetes Service (Amazon EKS) Blueprints** helps to integrate the landscape of Kubernetes, making it easier to meet your specific application needs using familiar tools.

Progressive modern application developers are already transitioning to Kubernetes as it makes it easier to automate and standardize deployment, scaling, security, management, and observability of containerized applications. This helps to reduce development time, increase operational efficiency, and improve overall reliability and security. With Kubernetes, you can leverage the same APIs and skill sets for running infrastructure across environments in the cloud and on premises.



Developers planning to or already running Kubernetes—or thinking about building new applications on Kubernetes—can run them on Amazon EKS.

Amazon EKS takes full advantage of AWS's operational excellence to provide developers with a highly scalable, reliable, and secure Kubernetes platform in the cloud.

Amazon EKS 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 lifecycles and other key tasks. It also includes integrations with AWS products 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, you can eliminate undifferentiated heavy lifting and focus on delivering value to your users. Additionally, Amazon EKS is customizable to support any need. For example, Amazon EKS 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.

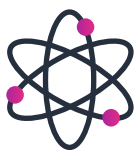
For on-premises or hybrid deployments, **Amazon EKS Anywhere brings the Amazon EKS experience to your data center**, allowing you to retain latency-sensitive or regulated applications in a private cloud on premises, all while leveraging a common toolset across environments.

## Why use Kubernetes today



### Run applications at scale

Kubernetes lets software teams define and quickly scale complex, containerized applications.



### Accelerate speed to market

With Kubernetes, your software teams can automate application updates and speed up the deployment process.



### Save time, boost productivity

Automate tasks associated with developing and deploying cloud-native apps to save time and free up resources.



### Stay secure

With strict version control, Kubernetes permits rapid rollbacks if a vulnerability in new code is discovered.



### Seamlessly move applications

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



### Add new functionality

A large community of developers and companies build extensions, integrations, and plugins that help Kubernetes users do more.



### Run your apps anywhere

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

Software developers with in-house Kubernetes and operational expertise can run on AWS using Amazon EKS for more scalable, secure, and highly available infrastructure. Amazon EKS automatically manages the availability and scalability of the Kubernetes control plane responsible for scheduling containers, storing cluster data, and automating other key tasks.

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

### **With AWS 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.

To bootstrap clusters with the operational software needed to deploy and operate workloads, use Amazon EKS Blueprints. You describe the configuration for the desired state of your environment, such as the control plane, worker nodes, and Kubernetes add-ons, and once configured, use this blueprint to stamp out consistent environments across AWS accounts and regions.



## With Amazon EKS Blueprints, you can:

- Easily integrate with Amazon EKS and open-source add-ons, including Prometheus, Karpenter, Nginx, Traefik, Fluent Bit, Keda, and more.
- Implement relevant security controls needed to operate workloads from multiple teams in the same cluster.
- Improve work flow between platform and application teams to streamline code development.

For the creation and operation of on-premises Kubernetes clusters, use Amazon EKS Anywhere. With operational support from AWS, Amazon EKS Anywhere supports Bare Metal, CloudStack and VMware vSphere as deployment targets.

## With Amazon EKS Anywhere, you can:

- Simplify on-premises Kubernetes management while providing tools for automating cluster management.
- Reduce support costs and avoid maintenance of redundant open-source and third-party tools.
- Build an on-premises Kubernetes environment that is more reliable than self-managed offerings.





## CUSTOMER SNAPSHOT

# New Relic

**New Relic** is a leader in the observability space, providing engineers with a 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, as well as 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 here ›](#)



**"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

# Uniphore

Uniphore built a platform that leverages artificial intelligence (AI) to realize the value of every conversation a business has with its customers, employees, and partners. To support its cloud-based product line, Q for Sales, Uniphore turned to AWS and AWS Partner, DuploCloud.

Uniphore is running its application and infrastructure stack on AWS solutions including AWS Fargate and Amazon RDS. Using DuploCloud's DevSecOps-as-a-Service platform, Uniphore empowers its developers to focus on product innovation versus infrastructure management.

By leveraging DuploCloud's DevSecOps-as-a-Service platform on AWS, Uniphore gains three major benefits: self-service capabilities for developers, a comprehensive monitoring system, and robust application security. Most importantly, it is able to bring Q to Sales products to market quickly. New Q for Sales customers can get started in a few minutes to immediately augment their client interactions by realizing the value of each conversation.

[See full story here ›](#)



**"We have automated deployment and monitoring at the click of a button with DuploCloud on AWS."**

Saurabh Saxena, Vice President of Engineering, Uniphore



# Build with a serverless operational model

Modern software teams need faster ways to turn ideas into action. Their rapidly built applications must be able to scale quickly, have global availability, manage large amounts of data, and respond in milliseconds.

A serverless operational model is the answer, offering easier scaling and management of cloud applications by shifting much of the underlying compute resources to your cloud provider. This allows your development teams to focus on writing code that solves business problems.

With AWS serverless solutions, you can hand over tasks from server management to resource allocation to improve performance, scalability, availability, and security. With those tasks taken care of, your teams can accelerate speed to market and reduce costs, 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 might be easy to manage today, but challenges often arise as you grow.

Difficulty distributing ownership of your apps can slow the development process, impede scaling, and ultimately impact the agility and profitability of your business. 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 recommend building microservices architectures for apps 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.

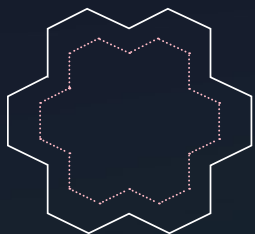
# Agility, scalability, and resilience

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 accessed 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—those in which actions are triggered in response to changes in data—to improve scalability and resiliency while also reducing costs.

## Monolith apps



Break down the app into smaller, loosely coupled components each responsible for a specific capability. Each component does one thing.

---

Do everything

---

---

Single app

---

---

Must deploy entire app

---

---

One database

---

---

Organized around technology layers

---

---

State in each runtime instance

---

---

One technology stack for entire app

---

## Microservices



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

---

Do 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

---

## The trouble with monoliths

By design, monoliths result in multiple developers pushing changes through a shared release pipeline, causing friction throughout the lifecycle.

During development, engineers need to coordinate their changes to make sure that they're not breaking someone else's code, and upgrading a shared library means convincing everyone to upgrade at the same time—a tough ask.

After development, pushing changes through the delivery pipeline is a challenge. Even a one-line fix in a tiny piece of code requires engineers 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.

## Do more with microservices

Microservices architecture means applications are composed of independent components that run each process as a service. Each are for business capabilities and perform a focused function.

As services run independently, they can be updated, deployed, and scaled to meet the demand for specific functions of an application.

Our customers are increasingly relying on microservices in event-driven architectures—those in which actions are triggered in response to changes in data—to improve scalability and resiliency while also reducing costs.

## CUSTOMER SNAPSHOT

# Kasada

Kasada is a bot mitigation company that helps customers detect and defend against bot attacks across web, mobile, and application programming interface (API) channels. To help the business run more efficiently at scale, Kasada began a multifaceted modernization exercise on AWS.

Kasada initiated preliminary work to shift from single-tenant to multi-tenant architecture, from monolith to microservices. This strategy wouldn't only reduce the cost to serve per customer, but also remove interdependencies that could hinder scaling. Further standardization and automation using **AWS CloudFormation** to provision infrastructure as code has contributed to increased velocity, by minimizing the burden on operations.

Implementation of an **AWS Landing Zone** with **AWS Control Tower** has also standardized governance and agile best practices across Kasada's IT environment, paving the way for multi-tenant architectures. Kasada now has all its customers on the same version of software, whereas previously 10 out of 15 customers could be using different versions.

[See the full story ›](#)

# kasada

**"It's imperative we have the fastest response times possible on a consistent basis. We need to stay up and scale up, both of which we can do without any concerns on AWS."**

David Turner, Head of Engineering, Kasada.





## CUSTOMER SNAPSHOT

# IBS Software

IBS Software is a leading SaaS solutions provider to the global travel industry, with customers across the globe. IBS was hosting its SaaS solutions in an on-premises environment, which lacked global reach and distribution, fully automated deployments, and the ability to optimize costs. To meet its customers' evolving needs and swiftly roll out changes, IBS chose to re-architect its solutions on the AWS Cloud for better agility, time to market, and scalability.

By using Graviton-based Amazon EC2 instances, IBS is improving customers' application experience, driving down operational costs, and supporting innovation. It can access more compute capacity at any time, which helps the company scale customers' applications to meet rising customer demand as the company grows. Looking ahead, IBS plans to expand customers' implementation of AWS Graviton, with the new Amazon EC2 C7g instances to power future applications.

[Read more here ›](#)

## ibsoftware

**“We want to build a culture of continuous innovation and adopt new technologies as the processor ecosystem is continuously evolving. As part of AWS Graviton adoption, we’ve reengineered our CI/CD pipelines to build, deliver, and run our multi-architecture products in x86 and arm64 processor families.”**

Anil Abraham, Associate Vice President & Head of Technology, Travel, and Hospitality Solutions, IBS Software



# As serverless as possible

As your architectural patterns and software delivery processes change, you will want to adopt an operational model that allows you to offload any activity that isn't a core competency of your business. To gain agility for rapid innovation, we recommend building a 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 help you to benefit most from the agility that 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, manage, or administer.
2. **Flexible scaling:** Your application can be scaled automatically or by adjusting its capacity through toggling the units of consumption. So, you can reliably meet any critical peak in demand.
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 since 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 SOC, HIPAA, PCI, and ISO compliance.



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 AWS Fargate

With the rise of containers and serverless computing, AWS continues to provide the broadest and deepest offerings for compute to make sure that you have the right 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 AWS Fargate or just running code with serverless function services like [AWS Lambda](#).



## AWS Lambda

About 80 percent of AWS container customers have also adopted AWS Lambda.<sup>5</sup> 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.

With Lambda, you can scale automatically from zero to millions of simultaneous processes in real time with a built-in security posture. Lambda also manages almost all server management tasks on your behalf, including hardware, software, networking and facilities management, data source integrations, and provisioning.

- Run code with serverless function services.
- Scale automatically.
- Benefit from the most trusted, reliable, and secure cloud provider.

## Amazon ECS on AWS Fargate

Customers choose Amazon ECS on AWS Fargate when they want to leverage containers and container orchestrators for development but do not need to manage servers or clusters.

With AWS Fargate, Amazon ECS supports serverless container orchestration, so you can leverage more of AWS's operational excellence 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.

- Offload server management.
- Deploy and manage containerized apps at scale.
- Simplify the building and running of applications.

<sup>5</sup> "The State of Serverless," DataDog, June 2022

## Amazon ECS on Amazon EC2

If you are looking for more control over the characteristics of how your applications run, Amazon ECS on Amazon Elastic Compute Cloud (Amazon EC2) can help. Offering the broadest and deepest compute platform, with over 600 instances and choice of the latest processor, storage, networking, operating system, and purchase model, you can best match the needs of your workload.

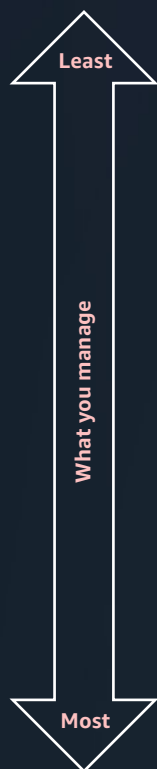
- Access reliable, scalable infrastructure on demand.
- Optimize performance and reduce costs with flexible instances.
- Provide secure compute for your applications.

## Amazon ECS Anywhere

Use Amazon ECS Anywhere when you want to run container workloads on your infrastructure. This approach will help you meet compliance requirements and scale your business without sacrificing your on-premises investments.

- Streamline software management on premises and on AWS with a standardized container orchestrator.
- Ensure a simple and consistent experience no matter where your container-based applications are running.
- Run a familiar, in-region ECS control plane to reduce operational overheads and focus on innovation.





**AWS Lambda**  
Serverless functions

**AWS Manages**

Data source integrations  
Physical hardware, software,  
networking, facilities  
Provisioning

**Customer Manages**

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

## CUSTOMER SNAPSHOT

# Prometeia

**Prometeia** is a leading European company for risk and wealth management solutions, as well as services for institutional investors. It is present in more than 20 locations serving more than 500 customers across all economic segments.

Prometeia's front-office SaaS wealth management platform, PFTPro, allows users to digitalize the entire investment and pension process. It currently supports the management of more than €2.5 trillion in assets under management (AuM) across 30,000 branches, 80,000 relationship managers, and 50 million portfolios.

Working with the AWS SaaS Factory team, Prometeia created a new platform as a service solution on AWS. This has allowed the company to offer a highly scalable and performant solution, and to delegate the management of the underlying infrastructure to AWS. With Amazon API Gateway at the heart of its infrastructure, it has a single standardized access point to its APIs and services.

[See the full story ›](#)



**"The partnership with AWS helped us build a highly scalable and performant solution, and delegate the management of the underlying infrastructure by leveraging managed and serverless services."**

Massimiliano Capaccio, Head of Wealth & Asset Management  
Technology Infrastructure Unit, Prometeia





nexthink



ENSEK

## CUSTOMER SNAPSHOT

### Nexthink

**Nexthink** is a software company focused on bringing clarity to IT departments through a unique combination of real-time analytics, automations, and employee feedback.

The company started work with AWS to migrate their software solution to the cloud. Today, Nexthink has transformed from an on-premises software companies to a SaaS company, with a fully cloud-native solution on AWS.

[Watch the video ›](#)

**“On Egress, we have a unified infrastructure that we control, making it easier to automate and industrialize the deployment of our application. One of our big benefits was the scale we can now target.”**

David Nemeshazy, Lead Architect, Nexthink

## CUSTOMER SNAPSHOT

### ENSEK

With evolving consumer expectations, increasing regulatory pressure, and global stress on climate challenges, the need to continuously adapt is unprecedented for energy retailers. These retailers need to build customer-centric and agile retail businesses to support consumer needs, as well as the energy transition. But for many, executing a successful digital strategy in this complex market can be a challenge.

SaaS provider **ENSEK** addressed these challenges by building a cloud-native solution for energy retailers on AWS. This solution supports the complete customer life cycle and uses an event-driven, API-first architecture that handles the nondiscretionary tasks involved in running an energy business. This approach helps to drastically reduce operating costs, improve the customer experience, and dedicate more resources to supporting the global energy transition.

[See the full story ›](#)

**“The AWS team has shown a real ability and willingness to help us solve problems for our customers. That’s been key to our business development efforts.”**

James Kirk, Chief Technology Officer and Cofounder, ENSEK

# The Modern DevOps model

Modern DevOps is the combination of cultural philosophies, practices, and tools that help developers to quickly and safely build 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, provide 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 so they can 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 any quicker 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 drag on release velocity. The solution involves abstraction, automation, and standardization.

To speed up the development process, abstract away 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 in the 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 provide 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 and respond to issues quicker. In fact, 67 percent of top-performing development teams use CI/CD to deploy code on demand, with 74 percent doing so several times a day. Businesses that adopt CI/CD practices also have 25 percent faster lead times, 50 percent fewer failures, and can release software two times faster.<sup>6</sup>

For these reasons and more, most development teams now use some form of CI/CD. At Amazon, we started using it to increase release velocity, and the results were dramatic—achieving millions of deployments per year. To help companies benefit from –our experience, we built a suite of developer tools based on the ones we use internally, so our customers can deliver code faster.

## Set the standard with AWS Well-Architected

AWS Well-Architected helps software developers build and deploy a template for secure, high-performing, resilient, and efficient infrastructure for a variety of applications and workloads.

Built around six pillars—operational excellence, security, reliability, performance efficiency, cost optimization, and sustainability—it provides a consistent approach to evaluating architectures and implementing scalable designs.

<sup>6</sup> “2020 State of DevOps Report,” Puppet, 2020

## 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 continuous integration 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 ›**



## CUSTOMER SNAPSHOT

# Innovyze — an Autodesk company

Autodesk provides critical design, make, and operate software solutions across the architecture, engineering, construction, manufacturing, media, and entertainment industries. In March 2021, Autodesk acquired Innovyze, who needed a cost-efficient approach to bridge existing on-premises infrastructure to the cloud with minimal operational disruption for customers.

To meet customer demand for cloud-based services, Innovyze built its Info360 services on AWS and developed three cloud-native SaaS products on top of it. These products provide asset management and operational analytics for telemetry data from systems and sensors at water utility companies so that they can better understand the health of water management systems and water/wastewater treatment plants.

By shifting data from on premises to a SaaS solution, Innovyze improved time to market, reduced deployment time, and increased scalability to cover virtually any number of water sensors and manage a wide range of inspections and conditions data.

[See the full story ›](#)

**Innovyze**  
An  AUTODESK company

**“Using AWS IoT Greengrass, we provide a secure and scalable mechanism to unlock data from our customers’ on-premises environments to give them great visibility into their operations and data in the cloud.”**

Boaz Brudner, Head of SaaS engineering,  
AI, and architecture, Innovyze



## CUSTOMER SNAPSHOT

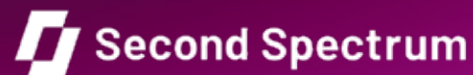
# Second Spectrum

**Second Spectrum's** artificial-intelligence-driven tracking technology helps broadcasters and commentators to provide quick insights accompanied by helpful visuals. To improve these tools and quickly develop new ones, Second Spectrum relies on the efficient use of engineering resources from AWS.

Second Spectrum began using Application Load Balancer for containerized workloads that it initially ran on self-managed Kubernetes clusters. The company soon recognized that it could run its Kubernetes clusters more efficiently by migrating to Amazon EKS.

Second Spectrum's integrated AWS solution helps to reduce Kubernetes hosting costs by 90 percent, reduce errors by up to 75 percent, improve application response time by five milliseconds per request, increase staff productivity, and create a scalable, replicable means of streamlining workload provisioning throughout the company.

[See the full story ›](#)



**“The direct-pod internet protocol target registration from AWS Load Balancer Controller enabled us to clean up and consolidate our configuration into a simple-to-use Kubernetes input.”**

Sheldon Kwok, Senior Director of Infrastructure, Second Spectrum



# 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's DevOps services team to make everything they do 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 that 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.

## Building a culture of innovation

1

### Start with customers

Every innovation should start with the customer need and, ultimately, lead to delighting your customers. Relentlessly prioritize customer demand.

2

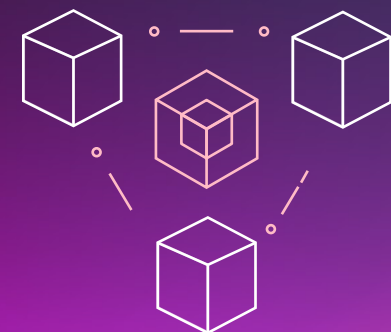
### Hire application builders and let them build

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.

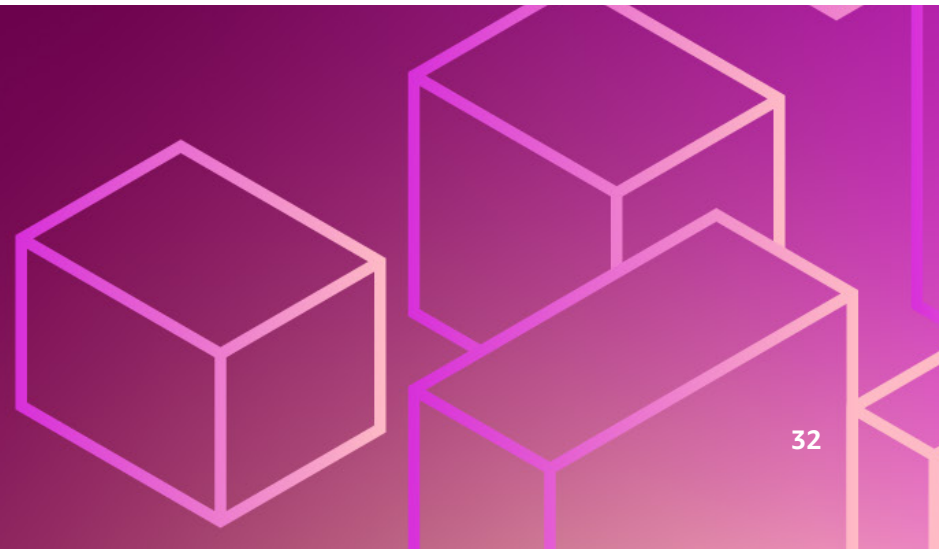
They also spend a lot of time trying to keep up with a complex and ever-changing tool and technology landscape. All of this leaves less time to develop new, innovative capabilities and services.

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 usage of open-source software. And, ultimately, searching and copying still takes time away from the good stuff.

Using tools that leverage capabilities like generative AI can remove this heavy lifting from 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 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 it. 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 through 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 development 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%**

### Optimize costs

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 AWS Lambda**

# Conclusion Section

## Replatform to managed container services

More than 80 percent of organizations today are using containers.<sup>7</sup> More containerized applications running in the cloud run on AWS than anywhere else.<sup>8</sup>

### 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.<sup>9</sup>

### 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.<sup>10</sup>

### Resources

[The Amazon Builder's Library](#)

### Recommended training (Classroom)

[DevOps Engineering on AWS](#)

### Recommended training (Online)

[Getting started with DevOps on AWS](#)

<sup>7</sup> "CNCF Annual Survey," Cloud Native Computing Foundation, 2022

<sup>8</sup> "2023 State of the Cloud Report," Flexera, 2023

<sup>9</sup> "Guidebook: Containers and Kubernetes on AWS," Nucleus Research, October 2019

<sup>10</sup> "IDC FutureScape Webcast: Worldwide Developer and DevOps 2023 Predictions," IDC, December 2022

## Learn more

[Learn more about building modern applications on AWS ›](#)

[Explore our ISV solutions, services and dedicated resources ›](#)

[Talk to an expert to implement the best practices of modern application development ›](#)

[Connect with an AWS partner to accelerate your modernization projects ›](#)

