



BUILD MODERN APPLICATIONS ON AWS



CONTENTS

Introduction	3
Modern Apps Empower Digital Innovators.....	4
Digital Innovators.....	6
Re-Platform To Managed Container Services	7
Build New on Serverless	8
Key Considerations For Building Scalable Modern Apps.....	9
Serverless Operational Model	11
Leveraging AWS Lambda And Managed Container Services On AWS.....	12
Re-Platform To Managed Container Services	13
Transform To A Modern Dev+Ops Model	14
Creating a Culture of Ownership: Manage Less, Innovate More With Modern Dev+Ops	16
Building a Culture of Innovation.....	17
Closing Summary	18
Building Modern Applications With AWS and DMI.....	19
Benefits of Using DMI To Help With Modernization.....	20
Case Study: A Leading Global Insurance Company	21
Start Your Application Modernization Journey.....	22



INTRODUCTION

Modern applications are changing how you deliver customer value

Many organizations are building new applications the hard way as they struggle to find a balance between managing technology and delivering new features.

While the cloud promises agility, that doesn't happen automatically. As organizations look to accelerate innovation, get more out of their data, and build new customer experiences, they need to modernize the way they build and operate applications. Modern applications are built with a combination of modular architecture patterns, serverless operational models, and agile developer processes.

In this eBook, we will guide you through the 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.



MODERN APPS EMPOWER DIGITAL INNOVATORS

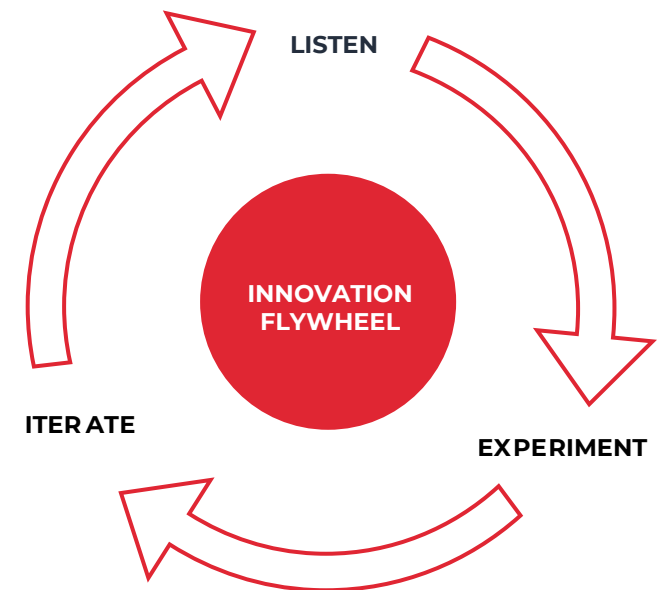
Innovation means listening to your customers

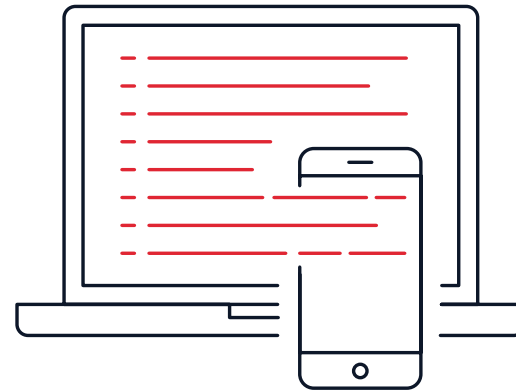
In a recent Vision Report, Digital Rewrites the Rules of Business, Forrester Research defines the customer-centric mindset of a digital innovator. The core mission of these modern disruptors is to:

“...Harness digital assets and ecosystems to continually improve customer outcomes and, simultaneously, improve operational excellence...by applying digital thinking to customer experiences, operations, ecosystems, and innovation.”

Focusing on your customer means making business decisions by working backward from your customer’s point of view. It means constantly evolving products and services to better deliver the outcomes that delight customers. And it means listening to what your customers truly care about so that you can continue inventing and iterating on their behalf. This is called the “innovation flywheel.”

The basic idea is that the driver for any innovation begins with customer demand, improves with customer feedback, and repeats constantly (and profitably) until the demand changes and the whole cycle begins again. The faster your teams can get your own innovation flywheel spinning, the better you’ll be able to build modern applications and the more you will stand apart from competitors.





Building modern applications on AWS will get you to market faster. By speeding up the build-and-release cycle and offloading operational overhead, developers can quickly build new features. You will increase innovation with a modular architecture that lets teams experiment 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. Finally, you will improve total cost of ownership (TCO) with a pay-for-value pricing model that reduces the cost of over-provisioning or paying for idle resources.

To build modern applications, you may need to reconsider the foundation on which they are built. While shifting this architecture may be dramatic at an organizational level, the process does not need to be brutal. Many organizations take an inspired leap to build new modern apps in the cloud, but plenty of others take a hybrid approach, often taking a team-by-team and workload-by-workload journey, moving opportunistically one step at a time.

DIGITAL INNOVATORS

Through our experience building applications for Amazon.com and from serving millions of AWS customers, we have observed three pathways that organizations can take for translating their vision of application modernization into a reality, generating value for business in the process.

1

Re-platform to managed container

services. Organizations already running containers on-premises, or thinking about moving applications to containers, can re-platform those workloads to container services on AWS to simplify operations and reduce management overhead costs, such as orchestration and infrastructure provisioning.

2

Build new apps on serverless

architecture. As organizations build new applications or features, we recommend they use serverless technologies and purposebuilt databases to maximize agility, as well as advanced development tools to accelerate development.

3

Transform to a Modern Dev+Ops model.

To create a cultural shift to build modern applications at scale, organizations can leverage DevOps services and tools while maintaining a high bar on security and governance.

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 at the onset.



RE-PLATFORM TO MANAGED CONTAINER SERVICES

Containers are a lightweight and portable way to run and deploy applications. Containerizing existing applications is often a first step in an organization's modernization journey. If you are considering moving your applications to containers, you would benefit from re-platforming those workloads to AWS managed services like [Amazon Elastic Kubernetes Service \(Amazon EKS\)](#) or [Amazon Elastic Container Service \(Amazon ECS\)](#) with [AWS Fargate](#). A managed container service helps to reduce operational burden while improving scalability, reliability, security, and availability. With managed container services on AWS, you no longer have to worry about managing containers. Instead, you can focus your resources on education for the upskilling you need to develop modern applications with serverless computing.





BUILD NEW ON SERVERLESS

Modernizing app development is the adoption of services, practices, and strategies that enable developers to build more agile applications. And with the speed and reliability of modern infrastructure, developers can deliver secure apps that scale from prototype to millions of users automatically, so they can innovate and respond to change faster. Many modern applications are built serverless-first, a strategy that prioritizes the adoption of serverless services so customers can increase agility throughout the application stack. With serverless technologies, you no longer have to manage physical servers and you will benefit from automatic scaling, built-in high availability, and a pay-for-value billing model. Instead of worrying about managing and operating servers or runtimes, you can focus on product innovation while enjoying faster time to market. In addition, front-end web and mobile tools and services can be built on top of AWS. The reliability of this infrastructure helps brands deliver secure, highly available apps that can scale automatically across the globe.



KEY CONSIDERATIONS FOR BUILDING SCALABLE MODERN APPS

Architectural patterns: microservices

Although your monolithic app might be easy to manage today, challenges often arise as you grow, 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 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.

With a monolith, you have many developers all pushing changes through a shared release pipeline, which causes friction at many points of the lifecycle. 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.

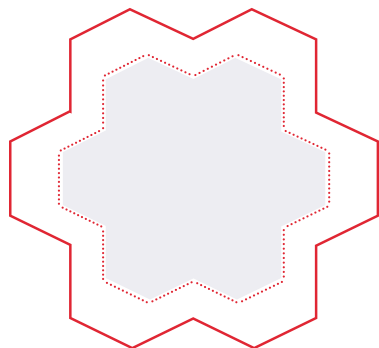
After development, you also face overhead when you're pushing the changes through the delivery pipeline. Even when making a one-line change in a tiny piece of code, engineers need to coordinate their changes ahead of time, merge their code, resolve conflicts within releases, rebuild the entire app, run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



With a microservices architecture, an application is made up of independent components that run each application process as a service. Services are built for business capabilities, and each service performs a single function. Because it runs independently and is managed by a single development team, each service can be updated, deployed, and scaled to meet the demands for specific functions of an application. For example, an online shopping cart can be used by many more users during a sale. 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 application scalability and resiliency while also reducing costs.

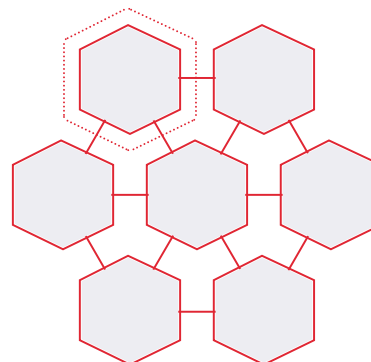
Everything vs. One thing: two types of applications

MONOLITH APPS



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



SERVERLESS OPERATIONAL MODEL

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 that can enable rapid innovation, we recommend building microservices architecture, operating and deploying software using automation for things like monitoring, provisioning, cost management, deployment, and security and governance of applications. Choosing a serverless-first strategy—opting for serverless technologies wherever possible—enables you to maximize the operational benefits of AWS.

With a serverless operational model, you can build and run applications and services without provisioning and managing servers. This eliminates server management, provides flexible scaling, enables you to pay only for value, and automates high availability. This model lets you build and manage the aspects of your application that deliver customer value without having to worry about the underlying detail.

Whether you are building net-new applications or migrating legacy, building with serverless primitives for compute, data, and integration will enable you to benefit from the most agility the cloud has to offer.



A serverless operational model is ideal for high-growth companies that want to innovate quickly. Serverless enables teams to move even faster and keep a laser focus on the activities that differentiate your business so you can speed up your innovation flywheel.



LEVERAGING AWS LAMBDA AND MANAGED CONTAINER SERVICES ON AWS

Leveraging AWS Lambda and managed container services on AWS

With the rise of containers and serverless computing, instances are no longer your only cloud computing option. 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 container services like Amazon ECS and Amazon EKS or event-driven serverless compute services like [AWS Lambda](#).

In reality, most customers use a combination of both container services and event-driven serverless compute services. 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.




RE-PLATFORM TO MANAGED CONTAINER SERVICES

So how do you frame the decision?

Customers choose AWS Lambda when they have teams focused primarily on writing code and no limitations on existing instances or container platforms. AWS Lambda offers the maximum abstraction from infrastructure, and so it enables customers to release fastest which is why new applications are a great fit for AWS Lambda.

Customers often choose containers when they have existing containers investments, open-source preferences for Kubernetes, or specific requirements for managing or configuring infrastructure. Containers are the most popular way to package code and are a great choice for modernizing legacy applications.

	AWS Manages	Customer Manages
Least  AWS Lambda Serverless functions	<ul style="list-style-type: none"> • Data source integrations • Physical hardware, software, networking, facilities • Provisioning 	<ul style="list-style-type: none"> • Application code • Data source integrations • Physical hardware, software, networking, facilities • Provisioning
AWS Fargate Serverless containers	<ul style="list-style-type: none"> • Container orchestration, provisioning duster scaling • Physical hardware, host OS/ kernel, networking, facilities 	<ul style="list-style-type: none"> • Application code • Data source integrations • Security config and updates, network config • Management tasks
Amazon ECS/ Amazon EKS Container management as a service	<ul style="list-style-type: none"> • Container orchestration control plane physical hardware, networking, facilities 	<ul style="list-style-type: none"> • Application code • Data source integrations • Security config and updates, network config, firewall • Work clusters • Management tasks
Most Amazon EC2 Infrastructure as a service	<ul style="list-style-type: none"> • Physical hardware, networking, facilities 	<ul style="list-style-type: none"> • Application code • Data source integrations scaling • Management tasks • Security config and updates, network config • Provisioning, managing scaling and patching of servers



TRANSFORM TO A MODERN DEV+OPS MODEL

AWS has identified a set of common, broadly accepted practices that, when adopted, provide a mechanism for building a highperforming 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 Dev+Ops, and it's centered around bringing developers and operations closer by sharing operational tasks like compliance, observability, resilience, and infrastructure earlier into the development process and enhancing it with artificial intelligence and machine learning (AI/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. However, you will not get new features to your customers faster if your build-and-release process does not keep up with your team's 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 for errors and long-term 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 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 to employ frameworks and tooling that reduce the complexity of provisioning and configuring resources. This gives developers an ability to move quickly while also enforcing best practices for security, privacy, reliability, performance, observability, and extensibility throughout the development process. Development frameworks like this give you confidence that your architecture will support your business growth long term.



By defining your software delivery process through the use of best-practice templates, you can provide a standard for modeling and provisioning all infrastructure resources in a cloud environment. These “infrastructure as code” templates help teams get started on the right foot because the template provisions 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. 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.

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: we have achieved millions of deployments a year and we grow 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 where developers regularly merge their code changes into a central repository, after which automated builds and tests are run. Continuous integration most often refers to the build or integration stage of the software release process and entails both an automation component (e.g., a CI or build service) and a cultural component (e.g., learning to integrate frequently).

Continuous delivery - (CD) is a software development practice where code changes are automatically prepared for a release to production. Continuous delivery expands on continuous integration by deploying all code changes to a testing environment and/or a production environment after the build stage.

[*Learn how Amazon automates safe, hands-off deployments »*](#)



CREATING A CULTURE OF OWNERSHIP: MANAGE LESS, INNOVATE MORE WITH MODERN DEV+OPS

Innovation ultimately comes from people, and so enabling your people to deliver better customer outcomes is where modern application development starts. We use the concept of “products, not projects” to describe how this impacts team structure. Simply stated, it means that the teams that build products are responsible for running and maintaining them.

It 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 Amazon.com, we have learned firsthand the importance of giving autonomy to our teams. When we gave our teams ownership of the complete application lifecycle, including taking customer input, planning the roadmap, and developing and operating the application, they became owners and felt empowered to develop and deliver new customer outcomes. Autonomy creates motivation, opens the door for creativity, and develops a risk-taking culture in an environment of trust.

While embracing a culture of ownership is not inherently technical, it remains one of the most challenging aspects of modern application development. Empowering teams to become product owners involves changing the mindset of your organization, the structure of your teams, and the work for which they are responsible.

For most organizations, IT falls into one of two camps: it is viewed as a strategic competitive weapon or, more commonly, as a cost center necessary to support business growth.





BUILDING A CULTURE OF INNOVATION

- 1 Start with the customers** – Every innovation should start with a customer need and ultimately delight your customers. Prioritize relentlessly to focus on customer demand.
- 2 Hire 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 flywheel spins.
- 3 Support 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.

CLOSING SUMMARY

Manage less, innovate more

Modern applications create competitive differentiation by enabling rapid innovation. By adopting services, practices, and strategies that underscore speed and agility, you can shift resources from business as usual to differentiating activities with deep customer value. You can experiment more and turn ideas into releases faster.

You can foster 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 AWS Lambda and the AWS IoT platform to manage its Roomba robotic vacuum cleaners

Improve

By automating test procedures and monitoring at every stage of the development lifecycle, modern applications are reliable at deployment. Any issues can be evaluated and addressed in real time

See the proof:

Siemens decreases customer control system alerts by 90 percent and reduces infrastructure costs by 85 percent

Improve TCO

With a pay-for-value pricing model, modern applications reduce the cost of over-provisioning or paying for idle resources. By offloading infrastructure management, maintenance costs are also lower.

See the proof:

Save up to 80 percent on app maintenance with AWS



BUILDING MODERN APPLICATIONS WITH AWS & DMI



Modern application development is a powerful approach to designing, building, and managing software in the cloud. This proven approach increases the agility of your development teams and the reliability and security of your applications, allowing you to build and release better products faster. Many industries are innovating faster by changing the way they design, build, and manage applications, through what we call modern application development. Application modernizations are more complex, time-consuming and expensive undertakings than ever before. To answer all your questions on this journey, we help you focus on your business outcomes.

- Continuous Modernization: Leverage this methodology as an alternative to rip-and-replace programs to lower your costs
- Culture: Establish a continuous modernization culture for products and platforms
- Transform Legacy Applications: Turn to platforms by identifying, prioritizing, and eliminating obstacles for digital business one by one

DMI and AWS offers a comprehensive portfolio of services to support your business as you develop modern applications. From our experience helping organizations of every kind build applications, we've identified common characteristics of modern applications that digital innovators rely on for success.



BENEFITS OF USING DMI TO HELP WITH MODERNIZATION



DMI is an AWS Select Tier Services Partner that helps companies enable, operate, and innovate in the cloud. From building modern applications to operational excellence and futuristic transformations in the cloud, we've supported numerous customers transforming their businesses by leveraging AWS. DMI, an AWS Lambda Delivery validated Partner, has extensive experience and helps you attain –

- **Cost Optimization:** DMI can help you run your application for the lowest possible cost, enabling you to pass those savings on to your customers and increase your margins.
- **Composable Business:** Improved stability and agility by provisioning composable applications on demand.
- **Scalability:** Modernize diverse legacy systems for improved agility and scale.
- **Improve TCO:** Reduce app development costs and lower your TCO.



CASE STUDY: A LEADING GLOBAL INSURANCE COMPANY

Founded in the 19th century, the provider is the largest global property and casualty insurer, focused on protecting and preserving the lives and dreams of customers in more than 30 countries. It employs over 45,000 people in 29 countries and economies around the world – offering a wide range of insurance products and services, including personal automobile, homeowners, specialty lines, reinsurance, commercial multiple-peril, workers compensation, commercial automobile, general liability, surety, and commercial property.

CHALLENGE

A Fortune 100 U.S. property and casualty insurance provider faced several challenges associated with the legacy applications and services core to its business. Driven by a need to provide a better customer experience and to retire aging infrastructure, the company enlisted DMI to work with business and technology leaders to create a strategy rooted in customer focused design and cloud native development leveraging AWS Serverless technologies.

SOLUTION

DMI delivered a modern platform that stabilizes the customer's environment and dramatically reduces the cost of operation. Prior to this engagement, end users were faced with disparate systems to perform various functions, resulting in a disjointed experience. The new paradigm is a single consolidated digital experience that empowers business scalability, solidifies the brand, and allows the provider to stay competitive.

OUTCOMES

The engagement began with a customer-centric approach, interviewing more than 50 users to define key personas. Using a Scaled Agile Framework (SAFe) methodology, Lean-Agile product strategy and development, DevOps, and DesignOps. The insurance company now has a modern application and cloud-enabled approach to better meet customer demands today and for years to come.

“DMI has been a true partner to us, having put in the time and effort to deeply understand our business operations and priorities. Their partnership has helped us transform outdated digital experiences and learn new ways of delivering technology – all with a laser focus on our customers.”

Property & Casualty Business
Insurance Leader
A Leading Global Insurance Company



START YOUR APPLICATION MODERNIZATION JOURNEY

Replatform to Managed Container Services

Resources

[Amazon ECS workshop](#)

[Amazon EKS workshop](#)

[AWS AppRunner workshop](#)

Recommended Training (Classroom)

[Running Containers on Amazon Elastic](#)

[Kubernetes Service \(Amazon EKS\)](#)

Recommended Training (Online)

[Amazon Elastic Container Service \(ECS\)](#)

[Primer](#)

Build New Modern Applications with Serverless Technologies and Tools

Resources

[Innovator Island](#) - Serverless web application development workshop. Build a serverless web app video tutorial.

Recommended Training (Classroom)

[Advanced Developing on AWS](#)

Recommended Training (Online)

[Architecting Serverless Solutions](#)

Transform to a Modern DevOps Model

Resources

[The Amazon Builder's Library](#)

[AWS DevOps services](#)

Recommended Training (Classroom)

[DevOps Engineering on AWS](#)

Recommended Training (Online)

[Getting started with DevOps on AWS](#)

[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 »](#)





© Copyright, 2022 reserved DMI:

This message is produced and distributed by
DMI | <https://dminc.com/policies/privacy/>

