E-book

Build Modern Applications on AWS with Rackspace Technology

rackspace technology | aws

Contents

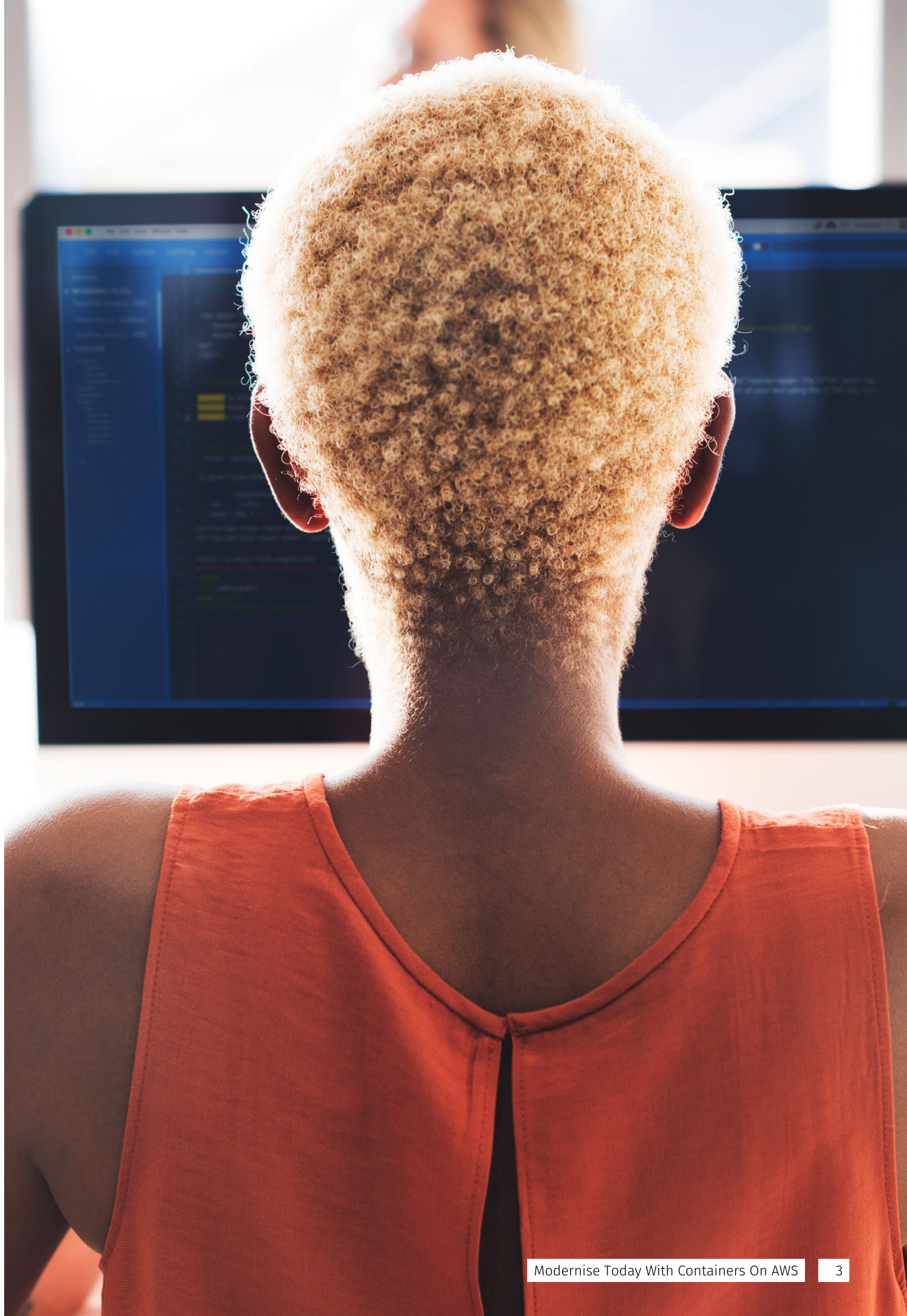
Introduction

Modern applications are changing how you deliver customer value

Many organisations 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 organisations look to accelerate innovation, get more out of their data, and build new customer experiences, they need to modernise 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 e-book, we will guide you through the three pathways that will help lay the foundation for modern application development in your organisation. We will also explore how modern application development with AWS and Rackspace Technology® can help your organisation innovate, reduce costs, accelerate time to market and improve reliability.



Modern applications empower digital innovators

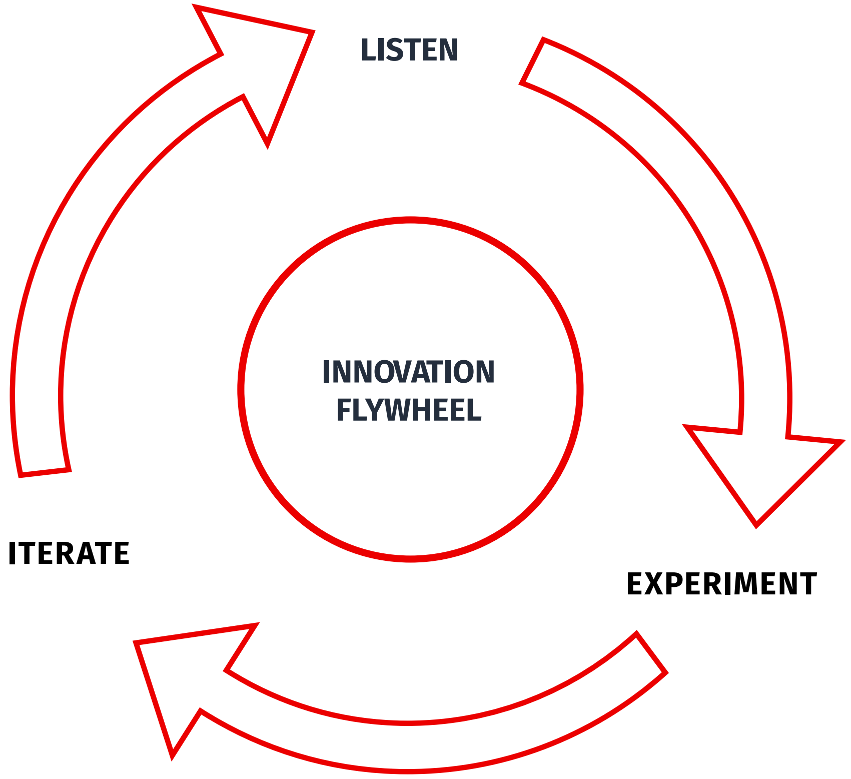
Innovation means listening to your customers.

In a 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 customers means making business decisions by working backward from your customers' 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."

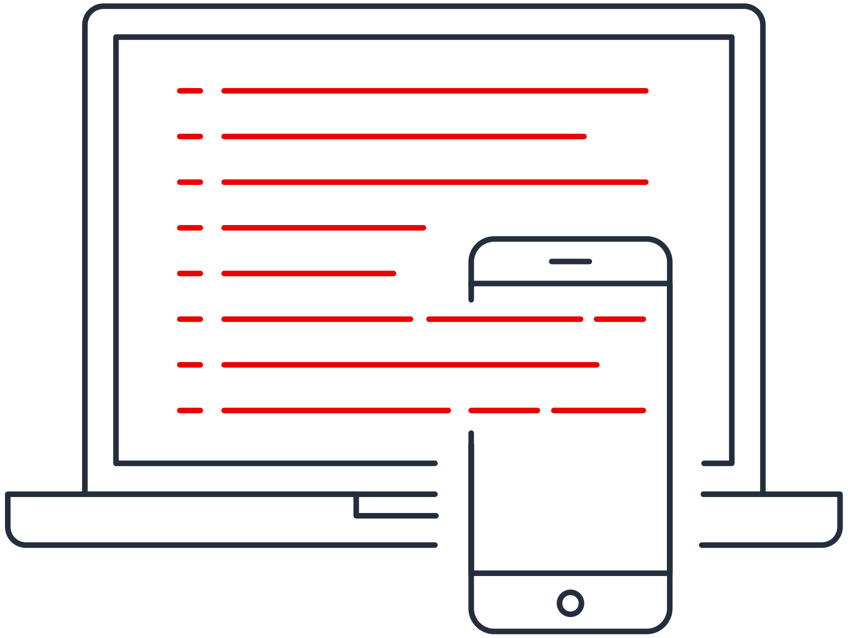
The basic idea is that the driver for any innovation begins with customer demand, improves with customer feedback, and repeats constantly (and profitably) until the demand changes and the whole cycle begins again. The faster your teams can get your own innovation flywheel spinning, the better you'll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



Building modern applications on AWS will get you to market faster

By accelerating 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 reduce total cost of ownership (TCO) with a pay-for-value pricing model that eliminates 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 organisational level, the process does not need to be brutal. Many organisations take an inspired leap to build new modern applications 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, we have observed three pathways that organisations can take for translating their vision of application modernisation into a reality — and generating value in the process.

1

Re-platform to managed container services. Organisations 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 applications on serverless architecture. As organisations build new applications or features, we recommend they use serverless technologies and purpose-built databases to maximise agility, as well as advanced development tools to accelerate development.

3

Transform to a modern DevOps model. To create a cultural shift to build modern applications at scale, organisations 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 applications that support business success. While you can modernise 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.

Re-platform to managed container services

Containerising existing applications is often a first step in an organisation's modernisation journey. If you are considering moving your applications to containers, you would benefit from re-platforming those workloads to a managed service like Rackspace Managed Platform for Kubernetes, Amazon Elastic Kubernetes Service (Amazon EKS) or Amazon Elastic Container Service (Amazon ECS) with AWS Fargate. A managed container service helps to reduce operational burdens 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 your core business and goals.

Build new on serverless

Modern application development incorporates 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 applications 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 prioritises the adoption of serverless services so customers can increase agility throughout the application stack. The adoption of serverless technologies will eliminate the need to manage physical servers and deliver the added benefits of 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 applications that can scale automatically around the globe.

Key considerations for building scalable, modern applications

Architectural patterns: microservices

Although your monolithic applications might be easy to manage today, challenges often arise as you grow, including how to distribute ownership of the application 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 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.

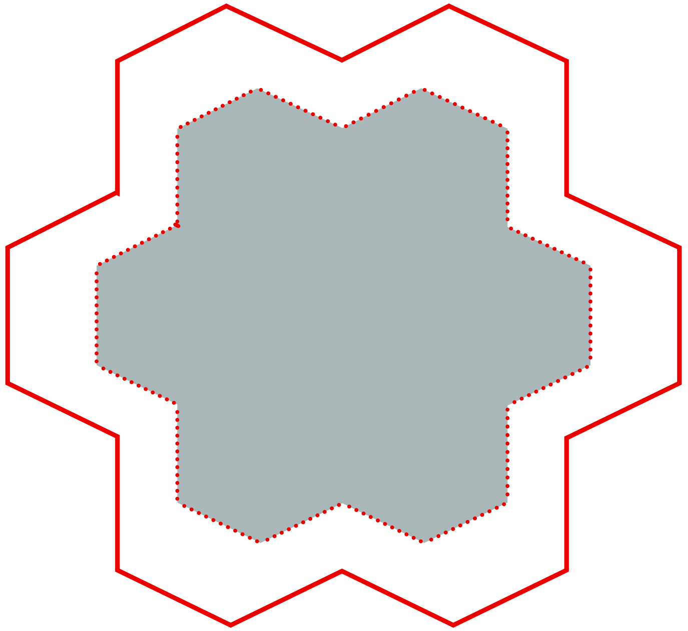
With a monolith, you have many developers all pushing changes through a shared release pipeline, which causes friction at many points of the lifecycle. During development, engineers need to coordinate their changes to make sure they're not breaking someone else's code. To upgrade a shared library and take advantage of a new feature, you need to convince everyone 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 application, run all of the test suites and redeploy.

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 increasingly rely 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.

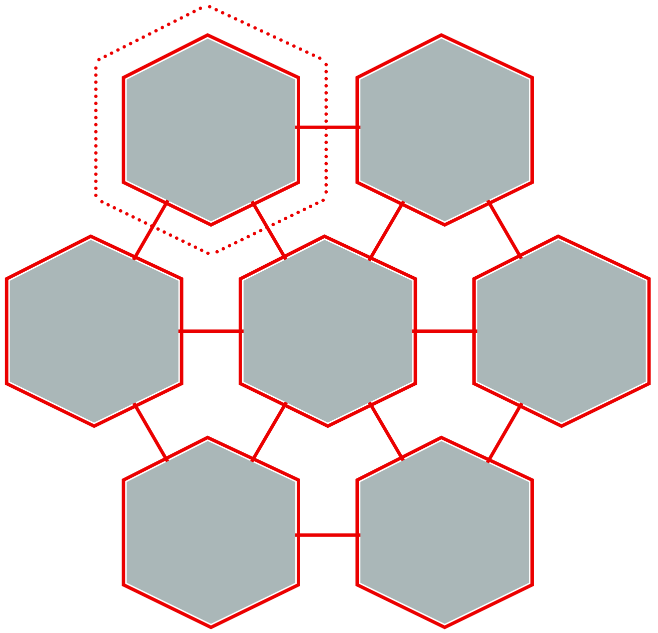
Everything vs. one thing: two types of applications

Monolith applications



- Do everything
- Single application
- Must deploy entire application
- One database
- Organised around technology layers
- State in each runtime instance
- One technology stack for entire application

Microservices



- Do one thing
- Minimal function services
- Deploy separately, interact together
- Each has its own datastore
- Organised around business capabilities
- State is externalised
- 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 maximise 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 details.

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

With the rise of containers and serverless computing, instances are no longer your only cloud computing option. Choosing the optimal compute resources 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 faster, which is why new applications are a great fit for AWS Lambda.

Customers often choose containers when they have existing container 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 modernising legacy applications.

Least
What you manage?
Most

	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, provisioning cluster scaling Physical hardware, host OS/kernel, networking, facilities	Application code, data source integrations Security configuration and updates, network configuration	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 configuration and updates, network configuration, firewall	Work clusters Management tasks
Amazon EC2 Infrastructure as a service	Physical hardware, networking, facilities	Application code Data source integrations, scaling Management tasks	Security configuration and updates, network configuration Provisionining, managing scaling and patching of servers

Transform to a modern DevOps model

AWS has identified a set of common, broadly accepted practices that, when adopted, provide a mechanism for building a high-performing DevOps organisation. 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.

This modern DevOps approach is centred 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 AI and machine learning

Developer agility: abstraction, automation and standardisation

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 standardisation.

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 applications. One way to do this is to employ 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 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 they 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. Most notably, teams that practice CI/CD spend 44% 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. For example, Amazon started using CI/CD to increase its release velocity. The results included making millions of deployments a year and growing faster every year. To help companies benefit from its experience, AWS built a suite of developer tools based on the tools the company used internally to its customers deliver code faster.

A bit more detail:

- **Continuous integration:** A software development practice where developers regularly merge their code changes into a central repository, after which 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 (e.g., a CI or build service) and a cultural component (e.g., learning to integrate frequently).
- **Continuous delivery:** A software development practice where code changes are automatically prepared for a release to production. CD expands on CI by deploying all code changes to a testing environment and/or a production environment after the build stage.

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

Innovation ultimately comes from people, and so enabling your people to deliver better customer outcomes is where modern application development starts. This 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.

When teams are given ownership of the complete application lifecycle, including taking customer input, planning the roadmap, and developing and operating the application, they became owners and feel 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 organisation, the structure of your teams and the work for which they are responsible.

Building a culture of innovation

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

1

Start with the customers: Every innovation should start with a customer need and aim to delight your customers. Prioritise 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.

Why build modern applications on AWS?

	AWS manages	Customer manages
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: Echelon Fitness achieved elastic scaling to meet 1,000% annual growth on 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: AGU condensed 70,000 pages of content into one content management system.
Improve reliability	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: Truecar scaled up critical systems to enable future growth.
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: iPromote reduced operating costs by 30% with .NET Core on AWS.

Building modern applications with AWS and Rackspace Technology

As today's leading cloud service provider, AWS provides an ever-expanding number of tools and services that are essential for innovating in the cloud. As an AWS Premier Consulting Partner and AWS Managed Service Provider, Rackspace Technology provides the expertise companies need to turn their ideas into optimised cloud-based products and services.

Rackspace Technology cloud architects will assess your applications and infrastructure in the context of your business goals and your team's strengths to determine which combination of cloud technologies and architectural patterns will best fit your long-term roadmap, including:

- **Serverless refactoring:** Build self-healing, auto-scaling applications, unchained from the limitations of servers.
- **Container adoption:** When serverless isn't an option, containers are the preference for deploying modern, complex, distributed applications.
- **Cloud native replatforming:** Incrementally modernise your application by adopting managed platform services as drop-in replacements for databases, messaging, API management, logging, monitoring, alerting and more.

Through a collaborative process, our team of business analysts and architects will review your application portfolio to understand existing architectural patterns, infrastructure, tools and processes. We'll evaluate the current state against business, functional, technical and cost objectives, and align your teams' skills and gaps with modern development processes and technologies.

Based on the assessment and your business goals, our solution architects will categorise your applications using a modernisation approach and design a reference blueprint architecture. We'll select tools and technologies and define a roadmap that focuses on early wins that build momentum and maximise ROI.

Following the adage of "think big, start small", we'll start the modernisation with a pilot project that focuses on defining repeatable patterns, rapid delivery and demonstrating business value.

Building on the momentum from the pilot project, our team will consolidate best practices into a foundational platform that enables building, deploying and operating modern applications at scale.

Rackspace Technology case study: Cando Rail Services

See how we helped our customer optimise in the cloud on AWS.

Challenge

To develop a serverless environment to process and push through IoT sensor data from train carriages to customers, implementing CI/CD best practices, and improving cost efficiency, scalability and stability.

Solution

Onica by Rackspace Technology™ began by fixing the existing environment to resolve high latency and poor system performance. Scaling and costs were also addressed. This helped drive a serverless solution implementation to achieve lower operational costs.

Outcome

The completed project resulted in predictability and reliability, improved performance and provided cost management. What once took 90 seconds, now takes less than two seconds.

Read the full story here

"All I have to say is we are really pleased with Onica by Rackspace Technology's work. Our architecture is not simple. It's quite complicated. We have old legacy technology that we're integrating it using AWS cloud-based services."*

Corrie Banks
Director of Logistics, Cando Rail Services

Benefits of using Rackspace Technology to help with modernisation

Innovation begins with the people who envision new ways to perform critical tasks and identify new business opportunities and ideas. Innovation is the by-product of the people who know how to leverage the vast number of technology tools and services available to turn ideas into reality.

Rackspace Technology is uniquely positioned to accelerate your application modernisation initiatives. Our team is comprised of solution architects and engineers focused on building modern applications that take advantage of cloud native technologies and microservices architectures. With specialisations in .NET to .NET Core, cloud native SaaS platforms and a proven framework for monolith to microservices migrations, we're ideally positioned to help you achieve your transformation goals.

Our delivery process prioritises agility and transparency, enabling your teams through our "do with" development approach. Embracing technology and empowering customers to deliver the future is our mission.

Rackspace Technology is your trusted partner across cloud, applications, security, data and infrastructure.

- 2,600+ certified technical experts
- Audited managed service provider
- 15 AWS competencies
- 13 AWS service delivery designations

See our Application Modernisation Playbook here.

About Rackspace Technology

Rackspace Technology is the multicloud solutions expert. We combine our expertise with the world's leading technologies — across applications, data and security — to deliver end-to-end solutions. We have a proven record of advising customers based on their business challenges, designing solutions that scale, building and managing those solutions, and optimising returns into the future.

As a global, multicloud technology services pioneer, we deliver innovative capabilities of the cloud to help customers build new revenue streams, increase efficiency and create incredible experiences. Named a best place to work, year after year according to Fortune, Forbes, and Glassdoor, we attract and develop world-class talent to deliver the best expertise to our customers. Everything we do is wrapped in our obsession with our customers' success — our Fanatical Experience® — so they can work faster, smarter and stay ahead of what's next.

Learn more at www.rackspace.com or call:

DACH: +49 (0)800 723 87 49
MEA: +9714 3999496
NE: +31 20 753 3200
UK: +44 (0)20 8734 8107

TSK-7340_Build-Modern-Applications-on-AWS_E-book_v3 :: September 7, 2022