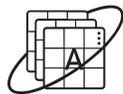# EIGHT STEPS TO CLOUD-NATIVE APPLICATION DEVELOPMENT

BRIEF

"Digitally advanced enterprises are 8x more likely to grow share but still lag behind digital natives."[1]

LEARN MORE
Understanding cloud-native applications

## MODERNIZING AND CREATING APPLICATIONS IN THE DIGITAL ERA

Digital business requires a culture of organizational agility, as the rapid pace of demand can only be satisfied by faster and more flexible development and delivery models. Most organizations do not have the luxury of completely rebuilding their technology foundation or immediately adopting new practices and mindsets, so they are embracing gradual yet fundamental shifts in culture, processes, and technology to support greater velocity and agility. The cloud-native approach modernizes existing applications and builds new applications based on cloud principles, using services and adopting processes optimized for the agility and automation of cloud computing.

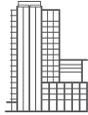## THE CLOUD-NATIVE APPLICATION DEVELOPMENT JOURNEY

The path to cloud-native applications varies by organization. Simply creating microservices does not lead to the service quality and delivery frequency required by digital business. And just adopting tools that support agile development or IT automation will not lead to the increased velocity of cloud-native approaches. Rather, success is achieved from a combination of practices, technologies, processes, and mindsets.

There are two complementary components to cloud-native application development: application services, or middleware, that speeds up the development of a cloud-native application, and infrastructure services, or a container platform, that speeds up its delivery and deployment.

Cloud-native application development is an approach to building and running applications based on four key tenets: service-based architecture, application programming interface (API)-based communication, container-based infrastructure, and DevOps processes.

Figure 1. Key tenets of cloud-native application development

---
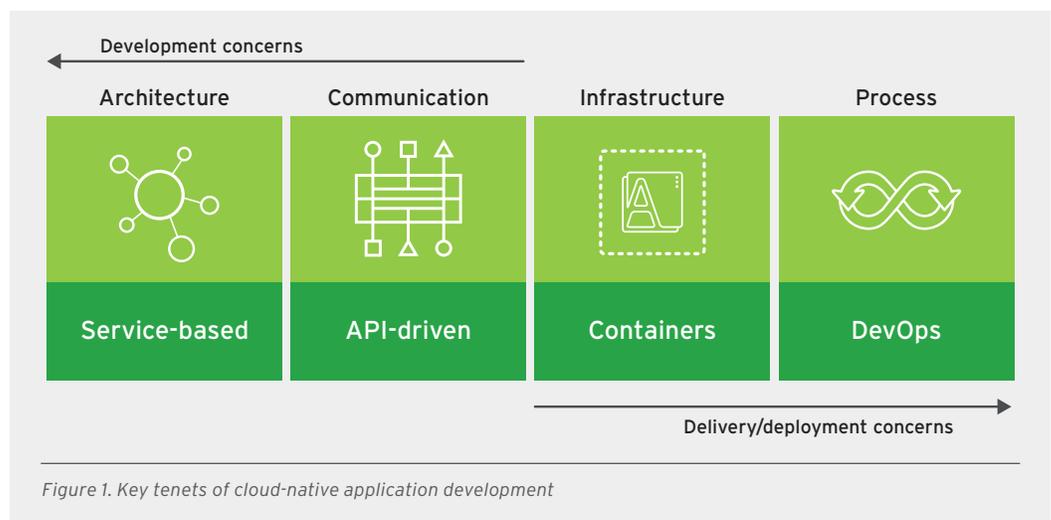
1 *Bain Survey: For Traditional Enterprises, the Path to Digital and the Role of Containers. December 2016.*
https://www.redhat.com/en/about/press-releases/digitally-advanced-traditional-enterprises-are-eight-times-more-likely-grow-share-committing-advanced-technologies-still-lag-behind-digital-natives

NORTH AMERICA
1 888 REDHAT1

EUROPE, MIDDLE EAST, AND AFRICA
00800 7334 2835
europe@redhat.com

ASIA PACIFIC
+65 6490 4200
apac@redhat.com

LATIN AMERICA
+54 11 4329 7300
info-latam@redhat.com

facebook.com/redhatinc
@RedHat
linkedin.com/company/red-hat

A service-based architecture, such as microservices or miniservices, advocates building modular, loosely coupled services. Services are exposed through lightweight, technology-agnostic APIs that reduce the complexity and overhead of deployment, scalability, and maintenance. Cloud-native applications rely on containers for true application portability across different environments and infrastructure, including public, private, and hybrid. DevOps principles focus on building and delivering applications collaboratively with delivery teams, including development, quality assurance, security, and IT operations teams.

EIGHT STEPS TO CLOUD-NATIVE APPLICATION SUCCESS

These recommendations, which can be completed in any order, provide a smooth transition to a cloud-native application approach:

1. **Evolve a DevOps culture and practices** to take advantage of new technology, faster approaches, and tighter collaboration.

2. **Speed up existing, monolithic applications** by either migrating to a container-based platform or migrating and then breaking the applications into microservices or miniservices.

3. **Use application services, i.e., middleware, to speed up the development of business logic.** These are effectively ready-to-use developer tools that have been optimized for containers.

4. **Choose the right tool for the right task** by using a container-based application platform that supports the right mix of frameworks, languages, and architectures.

5. **Provide self-service, on-demand infrastructure for developers** using containers and container orchestration technology to simplify access to underlying infrastructure, give control and visibility to IT operations, and provide application life-cycle management across environments.

6. **Automate IT to accelerate application delivery** with automation sandboxes; collaborative dialog for defining service requirements; self-service catalogs that empower users; and metering, monitoring, and chargeback policies and processes.

7. **Implement continuous delivery and advanced deployment techniques** to accelerate the delivery of your cloud-native applications.

8. **Evolve a more modular architecture** by choosing a design that makes sense for your specific needs, such as microservices, a monolith-first approach, or miniservices.