Red Hat

Eight steps to

Cloud-native application development

What is cloud-native application development?

Cloud-native application development is an approach to building and running applications. It speeds time to market by using the cloud computing model, which is based on these key tenets:

Service-based architecture

Uses modular, loosely coupled services, such as microservices. Increases development speed without increasing complexity.

Container-based infrastructure

Uses containers as a common operational model across application technology stacks, offering portability, horizontal scaling, and automation with low overhead and high density.

DevOps processes

Follows agile methodology, which builds and delivers applications collaboratively.

API-driven communication

Uses lightweight application programming interfaces (APIs) that reduce the complexity and overhead of deployment, scalability, and maintenance. Composes new business capabilities and opportunities with the exposed APIs.

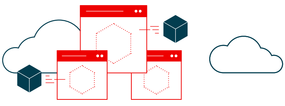
8 steps

Recommendations to help you succeed in cloud-native application development

Step 01 Evolve a DevOps culture and practices

Take advantage of new technology, faster approaches, and tighter collaboration by embracing the principles and cultural values of DevOps and organizing your organization around those values.

Step 02 Speed up existing applications using fast monoliths



Accelerate existing applications by migrating to a modern, container-based platform—and break up monolithic applications into microservices or miniservices for additional efficiency gains.

Step 03 Use application services to speed up development

Speed software development with reusability. Cloud-native application services are ready-to-use developer tools. However, these reusable components must be optimized and integrated into the underlying cloud-native infrastructure to maximize benefits.

Step 04 Choose the right tool for the right task

Spring Boot	Dropwizard	Node.js
Eclipse MicroProfile	Python	Golang
Eclipse Vert.X	Apache OpenWhisk	Jakarta EE

Use a container-based application platform that supports the right mix of frameworks, languages, and architectures—and can be tailored to your specific business application need.

Step 05 Provide developers with self-service, on-demand infrastructure

Use containers and container orchestration technology to simplify access to underlying infrastructure, give control and visibility to IT operations teams, and provide robust application life-cycle management across various infrastructure environments, such as datacenters, private clouds, and public clouds.

Step 06 Automate IT to accelerate application delivery

Lay the foundation for IT automation with:

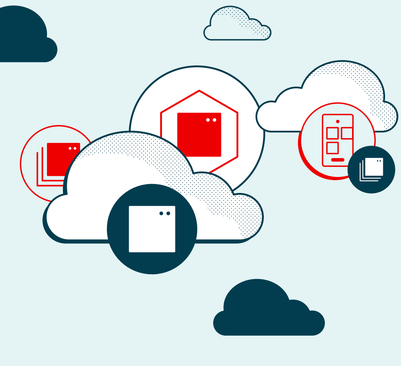
- Automation sandboxes for learning the automation language and process.
- Collaborative dialog across organizations for defining service requirements.
- Self-service catalogs that empower users and speed delivery.
- Metering, monitoring, and chargeback policies and processes.

Step 07 Implement continuous delivery and advanced deployment techniques

Accelerate the delivery of your cloud-native applications with automated delivery, continuous integration/continuous delivery (CI/CD) pipelines, rolling blue/green and canary deployments, and A/B testing.

Step 08 Evolve a more modular architecture

Choose a modular design that makes sense for your specific needs, using microservices, a monolith-first approach, or miniservices—or a combination.



Learn more at

www.redhat.com/en/topics/cloud-native-apps

Read the e-book

The path to cloud-native applications

https://red.ht/CNADebook

Red Hat