KUBERNETES AND CLOUD NATIVE ASSOCIATE (KCNA)

CLOUD NATIVE
COMPUTING FOUNDATION

Mastering the Cloud-Native Landscape with KCNA

KCNA Certification Overview

Entry-level certification

Validates foundational knowledge of Kubernetes, cloud-native architectures, and the CNCF ecosystem

Prepares for hands-on certifications and real-world DevOps or cloud-native roles

Provides the groundwork for Kubernetes architecture and operations

Covers core Kubernetes concepts

Pods, Services, Control Plane, container orchestration, and scheduling benefits

Explores microservices and cloud-native design patterns

Observability through metrics, logs, and tracing, CI/CD pipelines and GitOps practices

The KCNA certification empowers candidates with a robust foundational knowledge of Kubernetes and the cloud-native ecosystem, serving as a launching pad for further cloud-native specialization.

Kubernetes Fundamentals

Kubernetes Definition

Kubernetes is an open-source platform for automating deployment, scaling, and operations of application containers.

Core Components

Pods, Nodes, Namespaces, Deployments, ReplicaSets, Services

Control Plane Components

API Server, Scheduler, Controller Manager, etcd

Worker Node Components

Kubelet, Kube-proxy, Container Runtime

Basic Kubernetes Objects and Configuration

YAML, kubectl

Practical kubectl Operations

Deploying and scaling applications, Inspecting cluster health and logs, Debugging with logs and events

Container Orchestration Benefits

- ## High Availability

  Kubernetes automatically replaces failed containers, ensuring applications stay up and running.

- ## Scalability

  Kubernetes can dynamically scale the number of replicas based on demand, enabling applications to handle increased traffic.

- ## Load Balancing

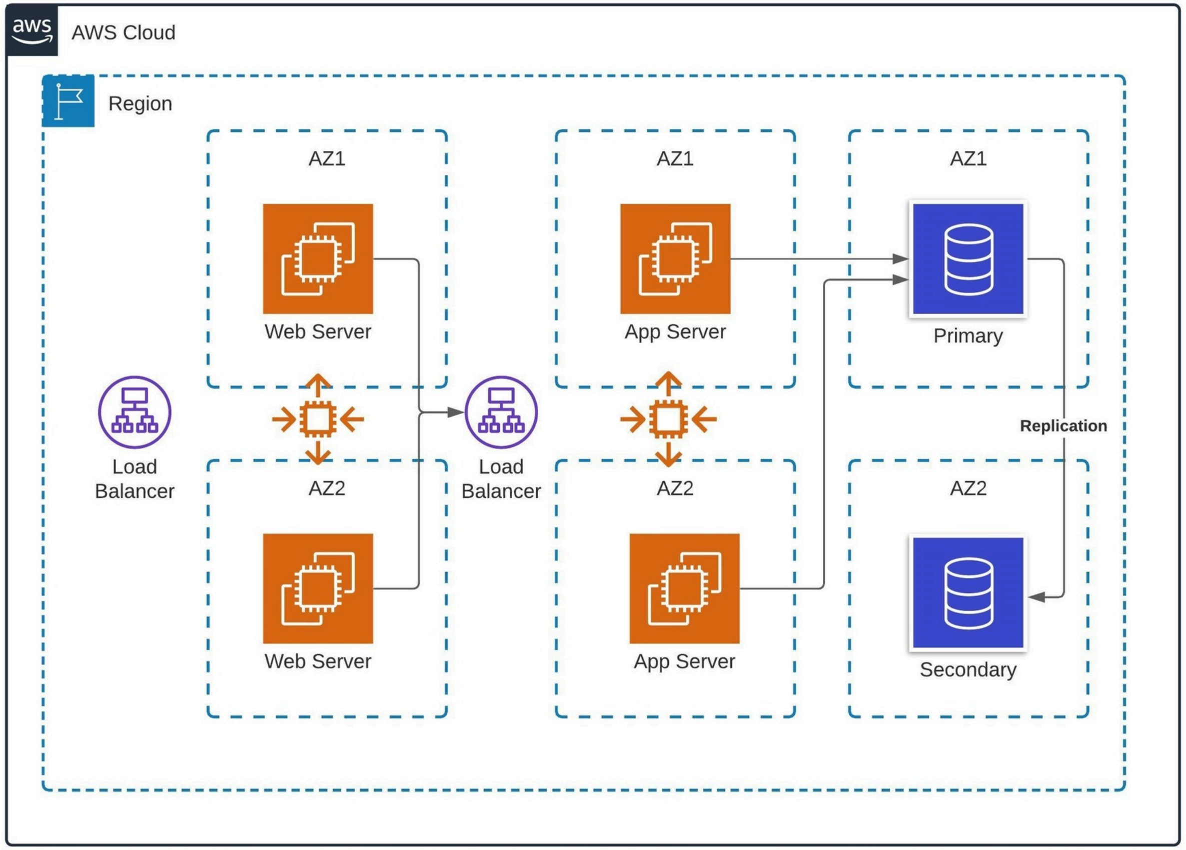
  Kubernetes distributes network traffic across multiple containers, improving performance and resilience.

- ## Rolling Updates and Rollbacks

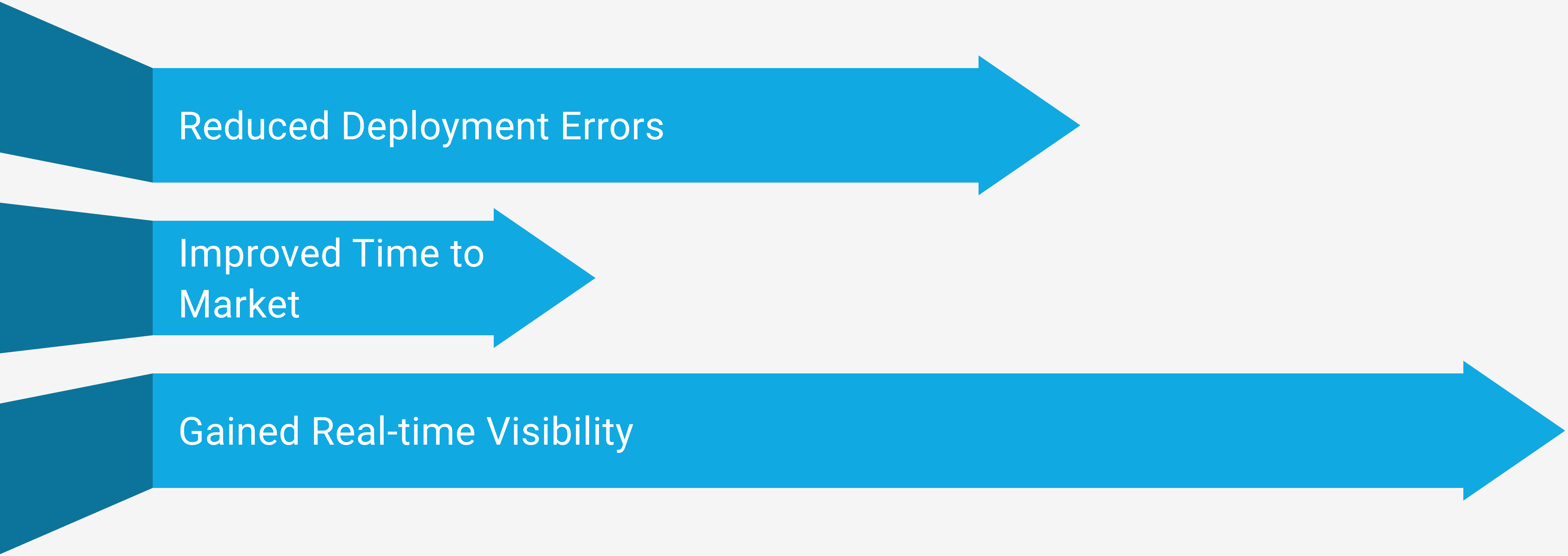
  Kubernetes allows for gradual, controlled deployment of new application versions and the ability to quickly roll back to a previous stable version.

Cloud-Native Architecture Principles

The context provided discusses the key concepts and design principles of cloud-native architecture, which is an approach to building scalable and maintainable applications in distributed, dynamic environments. Cloud-native architectures are characterized by the use of microservices, containerization, and dynamic orchestration.



Observability and Delivery



Reduced Deployment Errors

Improved Time to Market

Gained Real-time Visibility

Building a Cloud-Native Bank

A FinTech startup is launching a digital bank using cloud-native technologies. They have hired a KCNA-certified associate to design the initial architecture, implementing various solutions to improve deployment, visibility, and time-to-market.

- A model of a Cloud Native ecosystem without local persistency ("12 Factor") and fit for a regulated Enterprise
- Purpose: To make you familiar with the concepts & terminology
- *Not to be confused with the CNCF's Cloud Native landscape (https://landscape.cncf.io/)*

The Cloud Native ecosystem Kube - The real short explanation

- It's not just about Kubernetes !
- Clear **demarcation** between **provider** and **consumer** ("Namespace -as-a-Service")
- DevOps team is **autonomous** within its namespace(s)
- Workloads are " **Immutable**", "**Stateless**" & "**Short Living**"
- Data is persisted **externally** in **Data Services**
- "**Shift Left**" of security controls into the pipeline & the production cluster is **hands-off**
- **Automate Everything** !
- *(full explanation in the reserve slides ;-))*