KOENIG
step forward

Architecting and Operating SaaS in the Cloud

This course provides an in-depth exploration of designing, operating, and scaling Software-as-a-Service (SaaS) applications in the cloud. Designed for architects, developers, and product managers, the course covers the core tenets of SaaS business models, architecture patterns such as multi-tenancy and scalability, cloud-native engineering practices, and secure operational strategies. Participants will learn how to optimize tenant onboarding, manage billing and KPIs, implement modern DevOps processes, and ensure compliance in a SaaS environment, particularly with AWS as the deployment backbone.

Course objectives

By the end of this course, participants will be able to:

- Understand key SaaS business models, including subscription, freemium, and metered pricing
- Understand key SaaS business models, including subscription, freemium, and metered pricing.
- Design scalable, multi-tenant SaaS architectures with efficient tenant data management.
- Implement cloud-native SaaS operations using AWS and DevOps best practices.
- Apply SaaS-specific release management strategies such as feature flags and blue-green deployments.
- Manage tenant lifecycle processes — onboarding, billing, and support.
- Define and track KPIs relevant to SaaS product and business performance.
- Incorporate security, compliance, and data privacy into every layer of SaaS design and operations.

Prerequisites

- AWS Technical Essentials
- Familiarity with software architecture and DevOps workflows.
- Some experience with software development, deployment, or product management in a tech environment.

Target Audience
- Cloud & SaaS Architects

- Product and Engineering Managers
- DevOps Engineers
- Software Developers working in SaaS companies
- Solution Architects and Pre-sales Engineers

Course outline

Module 1: SaaS Business Models and Market Landscape

- Introduction to SaaS
  - o Definition and characteristics
  - o Comparison with traditional software delivery models
- Business Models
  - o Subscription-based SaaS
  - o Freemium and premium models
  - o Metered/billed usage models
- Pricing Strategies
  - o Per-user, per-feature, per-usage pricing
  - o Tiered pricing and bundling
- Key SaaS Metrics
  - o MRR/ARR, Churn, Customer Lifetime Value
  - o Engagement and conversion metrics

Module 2: How Do Hypervisors Work? How Does IaaS Function?

- Introduction
- Hardware Virtualization
- Auto-Provisioning
- Data Centre Rack Systems
- Scaling through Software Architecture or Hardware
- Motivation or Need for Scalable Architecture

- Scalable Architecture (of Software)
- Concept of Load Balancer
- Auto-Scaling
- Summary of Capabilities of Hypervisors
- A Simple Model of Infrastructure as a Service (IaaS)
- Example Case Situations

Module 3: Characteristics of Cloud SaaS Software

- Introduction
- Multi-Tenancy
- Customization
    - Web Tier: User Interface
    - Business Tier
    - Data Tier
    - Reports
    - Abilities to Choose Functions at Fine Granular Level
- Scaling (Auto-Scaling and Auto-Provisioning)
- Operational and Billing Support Services
- Software Upgrades and Maintenance
- Maintenance of Database
- Multi-Tenancy Models
    - Introduction to Multi-Tenant Architectures
    - Understanding Tenant Isolation
    - Introduction to Attribute-Based Access Control (ABAC)
    - Implementing Tenant Isolation with ABAC and AWS IAM
    - Security and Compliance

Module 4: Cloud Compatibility Measure

- Introduction
- Motivation to Come Up with Cloud Compatibility Measure

- Definition of 'Cloud Compatibility'
- SaaS (Solutions) Maturity Model
- SaaS Maturity Continuum Scale
- Cloud Compatibility Measure
    - Procedure to Set Up the 'Cloud Compatibility Measuring Scale'
    - Ideal Values for Characteristics
    - Case Study – Measures for Two Products of Similar Functionalities
- **Hands-on Activity:** Combined Discussion about All the Three 'Cloud Compatibility Measures'

Module 5: Architecting SaaS Solutions for Cloud Using Semi-Cloud Compatible SBBs

- Introduction
- Case Study
    - Introduction to Case Study
    - Description of Customer
    - Customers' Requirements
    - Solutions Implications and Constraints
    - Case Model
- Architecting Solution
    - Building Business Capabilities for a Group of Enterprises
    - Calibrating COTS against Cloud Compatibility Criteria
    - Key Challenges and Solutions in Finalizing SBBs
    - Security Requirements and Solutions to the Final Solution
- Summary of Cloud-Based SaaS Solution
    - Deployment Architecture for Minimum Usage
    - Evolving Deployment Architecture
    - Size Software for Scalability
    - Determining Scaling Algorithms

Module 6: Architecting Cloud SaaS Solutions with Cloud Non-Compatible Products

- Introduction
- Classification of Solutions Using Not-at-All Cloud Compatible Products
- Some General Strategies
- Case Study
  - o Use Case 1
  - o Use Case 2
  - o Some Common Observations
  - o Solution Description

Module 7: Cloud Computing Reference Architecture

- Introduction
  - o Review Bias
  - o What Does CCRA Bring to Table for Solution Architects?
- Cloud Computing Architectures are Service-Oriented Architectures
  - o Important Aspects of Cloud (SaaS) Services
  - o Cloud Reference Architecture Derives Experience from SOA in Addressing these Aspects
- A Quick Summary of the SOA RA
- Using the SOA RA with the CCRA

  Hands-on Activity: CCRA – Architecture Overview Diagram
  - o Roles of CCRA 10.5.2 Architectural Elements for Each of These Three Major Roles

Module 8: Cloud-Native Operations & Deployment in SaaS

- Release Management
  - o Continuous Integration / Continuous Deployment (CI/CD)
  - o Blue/Green and Canary deployments
  - o Feature sizing and toggling (feature flags)
- Testing Strategy

- - Integration testing with tenant context
    - Regression and smoke testing for SaaS
- Observability
    - Tenant-aware monitoring and alerts
    - Logging and distributed tracing

Hands-on Activities:

- Simulate blue-green deployment pipeline using AWS CodePipeline/CodeDeploy

Module 9: Deploying a SaaS application using AWS Ecosystem.