Curriculum for

CPSA Certified Professional for Software Architecture®

– Advanced Level –

Module:

CLOUDINFRA

Infrastructure, Container and Cloud Native

Version 1.1 (July 2018)

© (Copyright), International Software Architecture Qualification Board e. V. (iSAQB® e. V.) 2018

The curriculum may only be used subject to the following conditions:

1. You wish to obtain the CPSA Certified Professional for Software Architecture Advanced Level® certificate. For the purpose of obtaining the certificate, it shall be permitted to use these text documents and/or curricula by creating working copies for your own computer. If any other use of documents and/or curricula is intended, for instance for their dissemination to third parties, for advertising etc., please write to contact@isaqb.org to enquire whether this is permitted. A separate license agreement would then have to be entered into.

2. If you are a trainer, training course provider or training delivery organization, it shall be possible for you to use the documents and/or curricula once you have obtained a usage license. Please address any enquiries to contact@isaqb.org. License agreements with comprehensive provisions for all aspects exist.

3. If you fall neither into category 1 nor category 2, but would like to use these documents and/or curricula nonetheless, please also contact the iSAQB e. V. by writing to contact@isaqb.org. You will then be informed about the possibility of acquiring relevant licenses through existing license agreements, allowing you to obtain your desired usage authorizations.

We stress that, as a matter of principle, this curriculum is protected by copyright. The International Software Architecture Qualification Board e. V. (iSAQB® e. V.) has exclusive entitlement to these copyrights. The abbreviation "e. V." is part of the iSAQB's official name and stands for "eingetragener Verein" (registered association), which describes its status as a legal person according to German law. For the purpose of simplicity, iSAQB e. V. shall hereafter be referred to as iSAQB without the use of said abbreviation.

Table of Contents

0 Introduction: General information on the iSAQB Advanced Level 5

0.1 What is covered by the "CLOUDINFRA" module? 5
0.2 What does an Advanced Level Module teach? 5
0.3 What capabilities do graduates of the Advanced Level (CPSA-A) acquire? 5
0.4 Requirements for CPSA-A certification 5

1 Basic information about the CLOUDINFRA module 7

1.1 Structure of the curriculum for CLOUDINFRA and recommended time schedule 7
1.2 Duration, didactics and other details 7
1.3 Prerequisites for the module CLOUDINFRA 7
1.4 Structure of the curriculum for CLOUDINFRA 8
1.5 Supplementary information, terms, translations 8

2 Fundamentals of modern infrastructures 9

2.1 Terms and Concepts 9
2.2 Learning Objectives 9

3 Common architectural concepts 11

3.1 Terms and Concepts 11
3.2 Learning Objectives 11

4 Cloud Native Journey 13

4.1 Terms and Concepts 13
4.2 Learning Objectives 13

5 Helpful Pattern 15

5.1 Terms and Concepts 15
5.2 Learning Objectives 15

6 Development, CI/CD and Operations 17

6.1 Terms and Concepts 17
6.2 Learning Objectives 17

7 Automation 19

7.1 Terms and Concepts 19

7.2 LEARNING OBJECTIVES 19

8 CASE STUDY 21

8.1 LEARNING OBJECTIVES 21

9 REFERENCES 23

0 Introduction: General information on the iSAQB Advanced Level

0.1 What is covered by the "CLOUDINFRA" module?

Microservices, containers and container managers have changed considerably the way we design, develop and deploy software in recent years. Modern applications must work in a cluster of several nodes, be dynamically placeable, scalable and fault-tolerant.

Participants will learn ways to implement dynamic cloud-native architectures, container application design, logging/monitoring/alerting, container native storage and options for UI integration. Typical concepts of current container managers will also be demonstrated and how they can be used to implement common quality criteria for larger web applications.

In addition, current cloud providers will be classified, possibilities for automation will be demonstrated and aspects of software development and the application lifecycle will be discussed.

CLOUDINFRA focuses on operating aspects. The detailed concepts for the design and implementation of architectures in the CLOUDINFRA module are explained in an overview where necessary for understanding.

0.2 What does an Advanced Level Module teach?

- The iSAQB Advanced Level offers modular training in three areas of competence with flexible training paths. It takes individual interests and specializations into account.
- The certification takes the form of a term paper. The assessment and verbal examination is carried out by experts nominated by the iSAQB.

0.3 What capabilities do graduates of the Advanced Level (CPSA-A) acquire?

CPSA-A graduates are capable of the following:

- Independent and method-based design of medium- to large-scale IT systems.
- Responsibility for technology and content of IT systems of medium to high criticality.
- Development, design and documentation of measures for achieving non-functional requirements. Support of development teams in implementation of these measures.
- Control and implementation of architecture-related communication in medium to large development teams.

0.4 Requirements for CPSA-A certification

- A successful certification as CPSA-F (Certified Professional for Software Architecture, Foundation Level).
- At least three years of full-time professional experience in the IT industry, working on the design and development of at least two different IT systems.
    - Exceptions possible on request (e.g., participation in open source projects)
- Training and further education within the framework of iSAQB Advanced Level training courses with at least 70 credit points from all three competence areas (see iSAQB website for details).
    - Existing certifications can be added to these credit points, if applicable. The list of current certificates for which credit points are credited can be found on the iSAQB homepage.
- Successful completion of the CPSA-A certification exam.

1 Basic information about the CLOUDINFRA module

1.1 Structure of the curriculum for CLOUDINFRA and recommended time schedule

Content	Recommended minimum duration
Fundamentals of modern infrastructures	2 h
Common architectural concepts	2 h
Cloud Native Journey	4 h
Helpful patterns	4 h
Development, CI/CD and operations	4 h
Automation	2 h
Case study	2 h
Total (3 days, each lasting approx. 6 hours)	20 h

1.2 Duration, didactics and other details

The durations listed above are recommendations. The duration of a CLOUDINFRA course should be at least 3 days, but can be longer. Providers may differ in duration, didactics, type and structure of the exercises as well as the detailed course structure. In particular, the curriculum leaves the type of examples and exercises completely open.

Licensed CLOUDINFRA training courses contribute the following credit points to the final Advanced Level certification exam:

Methodical competence: 10 points
Technical competence: 20 points
Communicative competence: 0 points

1.3 Prerequisites for the module CLOUDINFRA

Participants should have the following knowledge and/or experience:

- Practical experience in the design and development of small to medium-sized software systems
- First practical experience in maintenance or evolution of software systems
- First practical experience in handling containers and their deployment.

In addition, the following concepts are also helpful for understanding:

- Knowledge or first practical experience in the derivation and implementation of modern microservice architectures
- First practical experience in dealing with container managers
- First practical experience with common cloud providers.

1.4 Structure of the curriculum for CLOUDINFRA

The individual sections of the curriculum are described according to the following structure:

- Terms/Concepts: Essential core terms of this topic.
- Teaching/Exercise Time: Determines the minimum teaching and exercise time required for this topic or exercise in an accredited training session.
- Learning Objectives: Describes the content to be conveyed, including its core terms and concepts.

This section also outlines the knowledge to be acquired in respective training courses.

- (R1) What should the participants be able to do? The participants should be able to apply these contents independently after the training. These contents are taught in training courses and should also be deepened through exercises or discussion.
- (R2) What should the participants understand? These contents are discussed in training courses and can be supported by exercises or discussion.
- (R3) What should the participants know? These contents (terms, concepts, methods, practices or similar) can support understanding or motivate for the topic. These contents are discussed in training courses if necessary, but not necessarily in detail.

1.5 Supplementary information, terms, translations

To the extent necessary for understanding the curriculum, we have included technical terms in the iSAQB glossary, defined them and, if necessary, supplemented them with translations of the original publications.

2 Fundamentals of modern infrastructures

Duration: 120 Min	Exercise time: none

2.1 Terms and Concepts

IaaS, SaaS, PaaS, CaaS, FaaS, etc. Infrastructure as Code, implementation of modern microservice architectures, distributed applications, containers, container managers, cloud providers, bare metal, time-to-market, availability, scaling, performance, IOPS, decoupling operations, private/public cloud, networking, application orientation.

2.2 Learning Objectives

The participants...

- (R1) know the difference between the *aaS terms, public/private cloud and can assess the existing offers
- (R1) know and understand the reasons for operating in the cloud and can assess the advantages and disadvantages
- (R2) understand that the possible high level of automation in provisioning/scaling and the availability of higher-value services are the main reasons for using a cloud provider
- (R2) know which type of supplier is best suited for which application and when the use of their own hardware (bare metal) can still be more appropriate
- (R1) understand how and to what extent VMs and networks can be set up and configured with common cloud providers
- (R1) understand the technical basics and the added value of containers and container managers
- (R2) know the difference between conventional deployment and container deployment
- (R1) understand how the use of containers and container managers leads to a better decoupling of responsibilities and thus to a more effective organization
- (R3 learn that both, the aspects of data security are relevant to deciding for or against a public cloud, as well as the performance of VMs and the maximum possible IOPS
- (R3) learn that modern architectures and their availability requirements imply the distribution of applications across multiple machines and data centers
- (R3) learn how the reduced scope of necessary hosting skills when using a cloud provider reduces risk and cost.

3 Common architectural concepts

Duration: 120 Min	Exercise time: 20 Min

3.1 Terms and Concepts

Self-contained Systems, Microservices, Independent Systems Architecture, integration concepts, CQRS, Event Sourcing.

3.2 Learning Objectives

The participants...

- (R1) understand the principles of modern architectures (especially microservices, self-contained systems) and the reasons that lead to these solutions.
- (R2) understand the different options for communication and data exchange between services, for example:
  - Event Sourcing
  - Messaging Middleware
  - RSS-Feeds
- (R1) understand how to integrate multiple services into an application in different ways. For example, by integrating them by:
  - Backend (message-bus, database, etc.)
  - Frontend (UI-Integration)
- (R3) understand the impact of managed services of cloud providers on modularization and architecture
- (R2) are familiar with the effects of a shared something/nothing architecture, which is caused in particular by scaling via containers.

4 Cloud Native Journey

Duration: 240 Min	Exercise time: 30 Min

4.1 Terms and Concepts

"*Applications adopting the principles of Microservices, packaged and delivered as Containers, orchestrated by Platforms, running on top of Cloud infrastructure*"

NFRs to the Platform, Cloud Native Storage, Overlay Networking, Network Policies, Container Security, Container Linux.

4.2 Learning Objectives

The participants...

- (R1) understand the advantages of the Cloud-Native concept.
- In particular by standard...
    - Interfaces for administration and configuration
    - Resource abstraction
    - Delivery artifacts
    - Application isolation, among others, with regard to the host system
- (R1) understand the abstraction concepts of current container managers and know patterns and best practices on how to transfer the fulfillment of non-functional requirements (NFR) to them.
  For example:
    - Finite Workloads like (Cron-) Jobs
    - Health Check and Self Healing
    - Scaling and Load Balancing
    - Placement
- (R1) know the principles of an overlay network (VLAN) for containers and container managers and understand how to isolate different services or systems.
  For example, implementation of a distributed firewall to isolate…:
    - Different tenants
    - Test systems in the context of a CI/CD
- (R1) understand the Linux kernel isolation mechanisms applied to containers (container security), including...:
    - Capabilities
    - Privileged Container
    - AppArmor
- (R1) understand how current core/container Linux distributions are optimized for running containers, the advantages they offer (including kernel and distribution updates), and the consequences for the applications they run.
- Understand different approaches to persisting data:
    - (R2) understand the difference between object and block storage
    - (R1) can classify the different storage services of cloud providers, different COTS products and self-managed solutions for different deployment scenarios
    - (R1) understand the advantages and disadvantages of a cloud native storage solution
    - (R2) know how to compare the higher-value managed services (such as RDBMS) of cloud providers with the use of self-managed solutions

(R3) know different concepts to realize backup and restore

5 Helpful Pattern

Duration: 240 Min	Exercise time: 20 Min

5.1 Terms and Concepts

Resilience Pattern, Container Application Design, Container Pattern, Service Mesh.

5.2 Learning Objectives

The participants...

- Know the concepts of the Container Application Design (Container Pattern) to…:
  - (R1) modularize software components of modern architectures into containers. For example:
    - Ambassador/Adapter/Sidekick
    - Scatter & Gather
    - Work Queue
  - (R1) realize technical and business tasks through separate containers
  - (R2) realize common tasks when using a container manager, in particular:
    - Configuration and initialization of the application
    - Adapted scaling via custom metrics
    - Managed container by operators or controllers
- Understand that in a distributed application, communication from outside and between services must be resilient to errors
  - (R1) understand which patterns can be used to ensure the necessary fault tolerance at communication level
  - (R2) understand how to use common Service Mesh concepts to detach resilience patterns from domain code and why a Service Mesh is useful for modern web applications.
    Some of the patterns discussed include:
    - Circuit Breaker
    - Conditional Rate Limits
    - Traffic Shifting.

6 Development, CI/CD and Operations

Duration: 240 Min	Exercise time: 20 Min

6.1 Terms and Concepts

Development Environment, Application Lifecycle Management, Canary Releases, Blue Green Deployment, Backup and Restore, Logging, Monitoring, Alerting, Distributed Tracing.

6.2 Learning Objectives

The participants...

- (R1) know different procedures to execute projects in or with the cloud. For example:
  - Organizational best practices
  - Development and CI/CD environments
- (R1) know options for implementing deployments and application lifecycle management in a cloud environment with container managers, in particular:
  - Versioning of containers and deployment descriptors etc.
  - Canary Releases and Blue Green Deployment
- (R2) get to know components and process models in order to realize a fast test and deployment cycle. In particular concepts for Dev/Test/Prod clusters like:
  - Responsibilities and access control
  - Best practices for component grouping
  - Tests and testability
- (R1) learn how to ensure the essential observability for distributed applications
  - Logging
  - Monitoring/Metrics and Alerting
  - Distributed Tracing
- (R3) learn options and responsibilities for creating Time Series queries for alerts that predict errors as far as possible.
- (R1) can among others roughly perform the following calculations:
  - Availability
  - Size of a cluster.

7 Automation

Duration: 120 Min	Exercise time: 20 Min

7.1 Terms and Concepts

Provisioning, Infrastructure as Code, Configuration.

7.2 Learning Objectives

The participants...

- (R1) learn options of automation to predictably create, change and improve the infrastructure, including:
  - Ansible
  - Chef
  - Terraform
- (R1) get to know container management platforms and administration tools, such as:
  - Rancher
  - Tectonic, Kops or Kubeadm
  - OpenShift
- (R2) know best practices for managing different configurations of the infrastructure
- (R3) get to know and use the API of different cloud providers.

© 2018 iSAQB e. V.

8 Case Study

Duration: 120 Min	Exercise time: 120 Min

As part of a curriculum-compliant training, a case study must explain the concepts in practical terms.

8.1 Learning Objectives

The case study is not intended to add new learning objectives, but to deepen the topics through practical exercises and clarify the practice.

9 References

Bilgin Ibryam, Roland Huß: "Kubernetes Patterns -- Patterns, Principles, and Practices for Designing Cloud Native Applications", Leanpub 2018, https://leanpub.com/k8spatterns

Florian Karlstetter: „Cloud-Architektur konsequent genutzt. Was ist Cloud native?“, Blogpost, https://www.cloudcomputing-insider.de/was-ist-cloud-native-a-669681/

Cornelia Davis: „Cloud Native: Designing change-tolerant software“, Manning MEAP, https://www.manning.com/books/cloud-native

Betsy Beyer, Chris Jones, Jennifer Petoff and Niall Richard Murphy: „Site Reliability Engineering“, O`REILLY, https://landing.google.com/sre/book.html

Christopher Schmidt: „Warum ein Container Manager das bessere Microservice Framework ist“, Blogpost, https://www.innoq.com/de/articles/2017/06/warum-ein-container-manager-das-bessere-microservice-framework-ist/

Christopher Schmidt: „Secure Kubernetes Cluster: Why, even with Docker and Kubernetes, there is still work to be done on security issues “, Blogpost, https://www.innoq.com/en/blog/secure-kubernetes-cluster/