

Curriculum für
CPSA Certified Professional for
Software Architecture®

– Advanced Level –

Modul:
CLOUDINFRA
Infrastruktur, Container und
Cloud Native



Version 1.1 (Juli 2018)

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

Die Nutzung des Lehrplans ist nur unter den nachfolgenden Voraussetzungen möglich:

1. Sie möchten das Zertifikat zum CPSA Certified Professional for Software Architecture Advanced Level® erwerben. Für den Erwerb des Zertifikats ist es gestattet, die Text-Dokumente und/oder Lehrpläne zu nutzen, indem eine Arbeitskopie für den eigenen Rechner erstellt wird. Soll eine darüberhinausgehende Nutzung der Dokumente und/oder Lehrpläne erfolgen, zum Beispiel zur Weiterverbreitung an Dritte, Werbung etc., bitte unter contact@isaqb.org nachfragen. Es müsste dann ein eigener Lizenzvertrag geschlossen werden.
2. Sind Sie Trainer, Schulungsanbieter oder Schulungsdurchführer, ist die Nutzung der Dokumente und/oder Lehrpläne nach Erwerb einer Nutzungslizenz möglich. Hierzu bitte unter contact@isaqb.org nachfragen. Lizenzverträge, die alles umfassend regeln, sind vorhanden.
3. Falls Sie weder unter die Kategorie 1. noch unter die Kategorie 2. fallen, aber dennoch die Dokumente und/oder Lehrpläne nutzen möchten, nehmen Sie bitte ebenfalls Kontakt unter contact@isaqb.org zum iSAQB e. V. auf. Sie werden dort über die Möglichkeit des Erwerbs entsprechender Lizenzen im Rahmen der vorhandenen Lizenzverträge informiert und können die gewünschten Nutzungsgenehmigungen erhalten.

Grundsätzlich weisen wir darauf hin, dass dieser Lehrplan urheberrechtlich geschützt ist. Alle Rechte an diesen Copyrights stehen ausschließlich dem International Software Architecture Qualification Board e. V. (iSAQB® e. V.) zu.

Inhaltsverzeichnis

0	<u>EINLEITUNG: ALLGEMEINES ZUM ISAQB-ADVANCED-LEVEL</u>	5
0.1	WAS VERMITTELT DAS MODUL "CLOUDINFRA"?	5
0.2	WAS VERMITTELT EIN ADVANCED-LEVEL-MODUL?	5
0.3	WAS KÖNNEN ABSOLVENTEN DES ADVANCED-LEVEL (CPSA-A)?	5
0.4	VORAUSSETZUNGEN ZUR CPSA-A-ZERTIFIZIERUNG	5
1	<u>GRUNDLEGENDES ZUM MODUL CLOUDINFRA</u>	7
1.1	GLIEDERUNG DES LEHRPLANS FÜR CLOUDINFRA UND EMPFOHLENE ZEITLICHE AUFTEILUNG	7
1.2	DAUER, DIDAKTIK UND WEITERE DETAILS	7
1.3	VORAUSSETZUNGEN FÜR DAS MODUL CLOUDINFRA	7
1.4	GLIEDERUNG DES LEHRPLANS FÜR CLOUDINFRA	8
1.5	ERGÄNZENDE INFORMATIONEN, BEGRIFFE, ÜBERSETZUNGEN	8
2	<u>GRUNDLAGEN MODERNER INFRASTRUKTUREN</u>	9
2.1	BEGRIFFE UND KONZEPTE	9
2.2	LERNZIELE	9
3	<u>GÄNGIGE ARCHITEKTURKONZEPTE</u>	11
3.1	BEGRIFFE UND KONZEPTE	11
3.2	LERNZIELE	11
4	<u>CLOUD NATIVE JOURNEY</u>	13
4.1	BEGRIFFE UND KONZEPTE	13
4.2	LERNZIELE	13
5	<u>HILFREICHE PATTERNS</u>	15
5.1	BEGRIFFE UND KONZEPTE	15
5.2	LERNZIELE	15
6	<u>DEVELOPMENT, CI/CD UND BETRIEB</u>	17
6.1	BEGRIFFE UND KONZEPTE	17
6.2	LERNZIELE	17
7	<u>AUTOMATISIERUNG</u>	19
7.1	BEGRIFFE UND KONZEPTE	19

7.2	LERNZIELE	19
<u>8</u>	<u>CASE STUDY</u>	<u>21</u>
8.1	LERNZIELE	21
<u>9</u>	<u>REFERENZEN</u>	<u>23</u>

0 Einleitung: Allgemeines zum iSAQB-Advanced-Level

0.1 Was vermittelt das Modul "CLOUDINFRA"?

Microservices, Container und Container-Manager haben die Art und Weise wie wir Software konzipieren, entwickeln und in Produktion bringen in den letzten Jahren stark verändert. Moderne Applikationen müssen in einem Cluster von mehreren Knoten funktionieren, dynamisch platzierbar, skalierbar und fehlertolerant sein.

Die Teilnehmer lernen hier Wege zur Realisierung dynamischer Cloud-Native-Architekturen, Container Application Design, Logging/Monitoring/Alerting, Container Native Storage und Möglichkeiten zur UI-Integration. Ebenso werden typische Konzepte aktueller Container-Manager aufgezeigt und vermittelt, wie sich damit für größere Webanwendungen gängige Qualitätsanforderungen realisieren lassen. Zusätzlich werden gängige Cloud-Anbieter klassifiziert, Möglichkeiten zur Automatisierung aufgezeigt, Ansätze der Softwareentwicklung und des Application Lifecycle besprochen.

Bei CLOUDINFRA stehen betriebliche Aspekte im Vordergrund. Die im Modul CLOUDINFRA detaillierten Konzepte zu Entwurf und Umsetzung von Architekturen werden, wo zum Verständnis notwendig ist, übersichtsweise erläutert.

0.2 Was vermittelt ein Advanced-Level-Modul?

- Der iSAQB-Advanced-Level bietet eine modulare Ausbildung in drei Kompetenzbereichen mit flexibel gestaltbaren Ausbildungswegen. Er berücksichtigt individuelle Neigungen und Schwerpunkte.
- Die Zertifizierung erfolgt als Hausarbeit. Die Bewertung und mündliche Prüfung wird durch vom iSAQB benannte Experten vorgenommen.

0.3 Was können Absolventen des Advanced-Level (CPSA-A)?

CPSA-A-Absolventen können:

- Eigenständig und methodisch fundiert mittlere bis große IT-Systeme entwerfen.
- In IT-Systemen mittlerer bis hoher Kritikalität technische und inhaltliche Verantwortung übernehmen.
- Maßnahmen zur Erreichung nichtfunktionaler Anforderungen konzeptionieren, entwerfen und dokumentieren. Entwicklungsteams bei der Umsetzung dieser Maßnahmen begleiten.
- Architekturrelevante Kommunikation in mittleren bis großen Entwicklungsteams steuern und durchführen.

0.4 Voraussetzungen zur CPSA-A-Zertifizierung

- Eine erfolgreiche Ausbildung und Zertifizierung zum CPSA-F (Certified Professional for Software Architecture, Foundation Level).
- Mindestens drei Jahre Vollzeit-Berufserfahrung in der IT-Branche, dabei Mitarbeit an Entwurf und Entwicklung von mindestens zwei unterschiedlichen IT-Systemen.
 - Ausnahmen auf Antrag zulässig (etwa: Mitarbeit in OpenSource-Projekten)
- Aus- und Weiterbildung im Rahmen von iSAQB-Advanced-Level-Schulungen im Umfang von mindestens 70 Credit Points aus allen drei Kompetenzbereichen (Details siehe iSAQB-Website).
 - Bestehende Zertifizierungen können ggfs. auf diese Credit-Points angerechnet werden. Die Liste der aktuellen Zertifikate, für die Credit Points angerechnet werden, ist auf der iSAQB-Homepage zu finden.
- Erfolgreiche Bearbeitung der CPSA-A-Zertifizierungsprüfung.





1 Grundlegendes zum Modul CLOUDINFRA

1.1 Gliederung des Lehrplans für CLOUDINFRA und empfohlene zeitliche Aufteilung

Inhalt	Empfohlene Minstdauer
Grundlagen moderner Infrastrukturen	2 h
Gängige Architekturkonzepte	2 h
Cloud Native Journey	4 h
Hilfreiche Muster (Pattern)	4 h
Development, CI/CD und Betrieb	4 h
Automatisierung	2 h
Case Study	2 h
Gesamt (3 Tage à ca. 6 Stunden)	20 h

1.2 Dauer, Didaktik und weitere Details

Die unten genannten Zeiten sind Empfehlungen. Die Dauer einer Schulung zum CLOUDINFRA sollte mindestens 3 Tage betragen, kann aber länger sein. Anbieter können sich durch Dauer, Didaktik, Art- und Aufbau der Übungen sowie der detaillierten Kursgliederung voneinander unterscheiden. Insbesondere die Art der Beispiele und Übungen lässt der Lehrplan komplett offen.

Lizenzierte Schulungen zu CLOUDINFRA tragen zur Zulassung zur abschließenden Advanced-Level-Zertifizierungsprüfung folgende Punkte (Credit Points) bei:

Methodische Kompetenz:	10 Punkte
Technische Kompetenz:	20 Punkte
Kommunikative Kompetenz:	0 Punkte

1.3 Voraussetzungen für das Modul CLOUDINFRA

Teilnehmer sollten folgende Kenntnisse und/oder Erfahrungen mitbringen:

- Praktische Erfahrung in Entwurf und Entwicklung kleiner bis mittelgroßer Softwaresysteme
- Erste praktische Erfahrung in Wartung oder Weiterentwicklung von Softwaresystemen
- Erste praktische Erfahrung im Umgang mit Containern und deren Deployment.

Hilfreich für das Verständnis einiger Konzepte sind darüber hinaus:

- Kenntnisse oder erste praktische Erfahrung in der Herleitung und Inbetriebnahme moderner Microservice-Architekturen
- Erste praktische Erfahrung im Umgang mit Container-Managern
- Erste praktische Erfahrung mit gängigen Cloud-Anbietern.

1.4 Gliederung des Lehrplans für CLOUDINFRA

Die einzelnen Abschnitte des Lehrplans sind gemäß folgender Gliederung beschrieben:

- **Begriffe/Konzepte:** Wesentliche Kernbegriffe dieses Themas.
- **Unterrichts-/Übungszeit:** Legt die Unterrichts- und Übungszeit fest, die für dieses Thema bzw. dessen Übung in einer akkreditierten Schulung mindestens aufgewendet werden muss.
- **Lernziele:** Beschreibt die zu vermittelnden Inhalte inklusive ihrer Kernbegriffe und -konzepte.

Dieser Abschnitt skizziert damit auch die zu erwerbenden Kenntnisse in entsprechenden Schulungen.

- (R1) Was sollen die Teilnehmer **können**? Diese Inhalte sollen die Teilnehmer nach der Schulung selbständig anwenden können. Innerhalb von Schulungen werden diese Inhalte unterrichtet und sollten zusätzlich durch Übungen oder Diskussion vertieft werden.
- (R2) Was sollen die Teilnehmer **verstehen**? Diese Inhalte werden in Schulungen thematisiert und können durch Übungen oder Diskussion unterstützt werden.
- (R3) Was sollen die Teilnehmer **kennen**? Diese Inhalte (Begriffe, Konzepte, Methoden, Praktiken oder Ähnliches) können das Verständnis unterstützen oder das Thema motivieren. Diese Inhalte werden in Schulungen bei Bedarf thematisiert, aber nicht notwendigerweise ausführlich unterrichtet.

1.5 Ergänzende Informationen, Begriffe, Übersetzungen

Soweit für das Verständnis des Lehrplanes erforderlich, haben wir Fachbegriffe ins iSAQB-Glossar aufgenommen, definiert und bei Bedarf durch die Übersetzungen der Originalliteratur ergänzt.

2 Grundlagen moderner Infrastrukturen

Dauer: 120 Min

Übungszeit: keine

2.1 Begriffe und Konzepte

IaaS, SaaS, PaaS, CaaS, FaaS etc. Infrastructure as Code, Umsetzung moderner Microservice-Architekturen, verteilte Anwendungen, Container, Container-Manager, Cloud-Anbieter, Bare Metal, Time-to-Market, Verfügbarkeit, Skalierung, Performance, IOPS, Decoupling Operations, Private/Public Cloud, Networking, Applikations-Orientierung.

2.2 Lernziele

Die Teilnehmer...

- (R1) wissen die unterschiedlichen *aaS Begriffe, Public/Private Cloud zu unterscheiden und können bestehende Angebote klassifizieren
- (R1) kennen und verstehen die Gründe für den Betrieb in der Cloud und können die Vor- und Nachteile einschätzen
- (R2) verstehen, dass der hohe mögliche Grad von Automatisierung bei der Provisionierung/Skalierung und die Verfügbarkeit von höherwertigen Diensten Hauptgründe für die Verwendung eines Cloud-Anbieters sind
- (R2) wissen, welcher Typ von Anbieter für welchen Einsatz am besten geeignet ist und wann dennoch der Einsatz von eigener Hardware (Bare Metal) sinnvoller sein kann
- (R1) verstehen, wie und in welchem Umfang bei gängigen Cloud-Anbietern VMs und Netzwerke eingerichtet und konfiguriert werden können
- (R1) verstehen die technischen Grundlagen und den Mehrwert von Containern und Container-Managern
- (R2) kennen den Unterschied zwischen konventionellem Deployment und dem mit Containern
- (R1) verstehen, wie die Verwendung von Containern und Container-Manager zu einer besseren Entkopplung von Verantwortlichkeiten und damit zu einer effektiveren Organisation führt
- (R3) lernen, dass sowohl die Aspekte der Datensicherheit für die Entscheidung für oder gegen eine Public Cloud relevant sind, wie auch die Performanz der VMs und die maximal möglichen IOPS
- (R3) lernen, dass Moderne Architekturen und deren Anforderungen an die Verfügbarkeit die Verteilung der Applikation auf mehrere Maschinen und Rechenzentren verlangen
- (R3) lernen, wie der reduzierte Umfang von notwendigen Hosting-Kompetenzen bei der Verwendung eines Cloud-Anbieters Risiko und Kosten reduziert.



3 Gängige Architekturkonzepte

Dauer: 120 Min	Übungszeit: 20 Min
----------------	--------------------

3.1 Begriffe und Konzepte

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

3.2 Lernziele

Die Teilnehmer...

- (R1) verstehen die Ansätze moderner Architekturen (speziell Microservices, Self-contained Systems) und die Gründe, die zu diesen Lösungen führen.
- (R2) verstehen die unterschiedlichen Möglichkeiten der Kommunikation bzw. des Datenaustauschs der Services untereinander, zum Beispiel:
 - Event Sourcing
 - Messaging Middleware
 - RSS-Feeds
- (R1) verstehen, wie man mehrere Services über unterschiedliche Wege zu einer Applikation integrieren kann, zum Beispiel durch Integration über:
 - Backend (Message-Bus, Datenbank, etc.)
 - Frontend (UI-Integration)
- (R3) verstehen die Auswirkung der Managed Services der Cloud-Anbieter auf die Modularisierung und Architektur
- (R2) kennen die Auswirkungen einer Shared-Something/Nothing-Architektur, die insbesondere durch eine Skalierung über Container entsteht.



4 Cloud Native Journey

Dauer: 240 Min

Übungszeit: 30 Min

4.1 Begriffe und Konzepte

"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 Lernziele

Die Teilnehmer...

- (R1) verstehen die Vorteile des Cloud-Native-Konzepts.
Insbesondere durch einheitliche:
 - Schnittstellen zur Administration und Konfiguration
 - Ressourcen-Abstraktion
 - Lieferartefakte
 - Isolation der Applikation u. a. bezüglich des Hostsystems
- (R1) verstehen die Abstraktionskonzepte aktueller Container-Manager und kennen Pattern und Best Practices, wie die Erfüllung nichtfunktionaler Anforderungen (NFR) auf sie übertragen werden kann.
Zum Beispiel:
 - Finite Workloads wie (Cron-) Jobs
 - Health Check und Self Healing
 - Skalierung und Load Balancing
 - Placement
- (R1) kennen die Prinzipien eines Overlay-Networks (VLAN) für Container und Container-Manager und verstehen, wie man verschiedene Services oder Systeme isolieren kann.
z. B. Realisierung einer verteilten Firewall zur Isolation von:
 - verschiedenen Mandanten
 - Testsystemen im Rahmen einer CI/CD
- (R1) verstehen die Linux-Kernel-Isolationsmechanismen, die bei Containern Anwendung finden (Container Security), u. a.:
 - Capabilities
 - Privilegierte Container
 - AppArmor
- (R1) verstehen, wie aktuelle Core/Container-Linux-Distributionen auf den Betrieb von Containern optimiert sind, welche Vorteile sie bieten (u. a. bzgl. Kernel- und Distributionsupdates) und welche Folgen das für die betriebenen Applikationen hat.
- verstehen die verschiedenen Wege um Daten zu persistieren:
 - (R2) verstehen den Unterschied von Object- und Block-Storage
 - (R1) können die verschiedenen Storage-Dienste der Cloud-Anbieter, unterschiedliche COTS-Produkte und selbstverwaltete Lösungen für unterschiedliche Einsatzszenarien klassifizieren
 - (R1) verstehen die Vor- und Nachteile einer Cloud-Native-Storage-Lösung
 - (R2) verstehen die höherwertigen Managed Services (wie etwa RDBMS) der Cloud-Anbieter mit der Verwendung selbstverwalteter Lösungen zu vergleichen
 - (R3) kennen verschiedene Konzepte, um Backup und Restore zu realisieren.



5 Hilfreiche Patterns

Dauer: 240 Min

Übungszeit: 20 Min

5.1 Begriffe und Konzepte

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

5.2 Lernziele

Die Teilnehmer...

- kennen die Konzepte des Container Application Designs (Container Pattern), um
 - (R1) Software-Komponenten moderner Architekturen in Container modularisieren zu können. Zum Beispiel:
 - Ambassador/Adapter/Sidekick
 - Scatter & Gather
 - Work Queue
 - (R1) technische von fachlichen Aufgaben durch getrennte Container zu realisieren
 - (R2) bei Verwendung eines Container-Managers übliche Aufgaben zu realisieren, insbesondere:
 - Konfiguration und Initialisierung der Applikation
 - Adaptierte Skalierung über Custom Metrics
 - Managed Container durch Operator bzw. Controller
- verstehen, dass bei einer verteilten Anwendung die Kommunikation von außen und zwischen den Services fehlertolerant realisiert werden muss
 - (R1) verstehen durch welche Pattern die notwendige Fehlertoleranz auf der Kommunikationsebene sichergestellt werden kann
 - (R2) verstehen wie man gängige Service-Mesh-Konzepte verwenden kann, um Resilience Patterns vom fachlichen Code zu trennen und warum ein Service Mesh für moderne Webanwendungen sinnvoll ist.
Besprochene Pattern sind u. a.:
 - Circuit Breaker
 - Conditional Rate Limits
 - Traffic Shifting.



6 Development, CI/CD und Betrieb

Dauer: 240 Min

Übungszeit: 20 Min

6.1 Begriffe und Konzepte

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

6.2 Lernziele

Die Teilnehmer...

- (R1) kennen verschiedene Vorgehensweisen, um Projekte in oder mit der Cloud zu durchführen. Zum Beispiel:
 - Organisatorische Best Practices
 - Development- und CI/CD-Umgebungen
- (R1) kennen Möglichkeiten, um in einer Cloud-Umgebung mit Container-Managern Deployments und ein Application Lifecycle Management zu realisieren, insbesondere:
 - Versionierung von Containern und Deployment-Spezifikationen etc.
 - Canary Releases und Blue Green Deployment
- (R2) lernen Komponenten und Vorgehensmodelle kennen, um einen schnellen Test- und Deployment-Prozess zu realisieren. Insbesondere Konzepte zum Dev/Test/Prod Cluster wie:
 - Verantwortlichkeiten und Zugriffskontrolle
 - Best Practices zur Komponentengruppierung
 - Tests und Testbarkeit
- (R1) lernen, wie die für eine verteilte Anwendung essentielle Beobachtbarkeit sichergestellt werden kann.
 - Logging
 - Monitoring/Metriken und Alerting
 - Distributed Tracing
- (R3) lernen Wege und Verantwortlichkeiten zur Erstellung möglichst fehlervorhersagenden Time Series Queries für Alerts kennen.
- (R1) können u. a. folgende Berechnungen überschlägig durchführen:
 - Verfügbarkeit
 - Größe eines Clusters.



7 Automatisierung

Dauer: 120 Min	Übungszeit: 20 Min
----------------	--------------------

7.1 Begriffe und Konzepte

Provisionierung, Infrastructure as Code, Configuration.

7.2 Lernziele

Die Teilnehmer...

- (R1) lernen die Möglichkeiten der Automatisierung zum vorhersehbaren Erzeugen, Verändern und Verbessern der Infrastruktur kennen und zu unterscheiden, u. a.:
 - Ansible
 - Chef
 - Terraform
- (R1) lernen vollständige Container-Management-Plattformen und Tools zur Administration kennen, wie zum Beispiel:
 - Rancher
 - Tectonic, Kops oder Kubeadm
 - OpenShift
- (R2) kennen Best Practices, um unterschiedliche Konfigurationen der Infrastruktur zu verwalten
- (R3) lernen die API unterschiedlicher Cloud-Anbieter kennen und einzusetzen.



8 Case Study

Dauer: 120 Min	Übungszeit: 120 Min
----------------	---------------------

Im Rahmen einer lehrplankonformen Schulung muss eine Fallstudie die Konzepte praktisch erläutern.

8.1 Lernziele

Die Case Study soll keine neuen Lernziele vermitteln, sondern die Themen durch praktische Übungen vertiefen und die Praxis verdeutlichen.



9 Referenzen

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: „Sichere Kubernetes Cluster: Warum trotz Docker und Kubernetes beim Thema Sicherheit noch immer etwas zu tun ist“, Blogpost, <https://www.innoq.com/de/blog/sichere-kubernetes-cluster/>