

**Modul: Cloud-native Programmierung**

<b>Niveau</b>	Master	<b>Kürzel</b>	CloudProg
<b>Modulname englisch</b>	Cloud-native Programming		
<b>Modulverantwortliche</b>	Nane Kratzke		
<b>Fachbereich</b>	Elektrotechnik und Informatik		
<b>Studiengang</b>	Informatik, Master		
<b>Verpflichtungsgrad</b>	Pflicht	<b>ECTS-Leistungspunkte</b>	5
<b>Fachsemester</b>	1	<b>Semesterwochenstunden</b>	4
<b>Dauer in Semestern</b>	1	<b>Arbeitsaufwand in Stunden</b>	150
<b>Angebotshäufigkeit</b>	WiSe	<b>Präsenzstunden</b>	45
<b>Lehrsprache</b>	Deutsch/Englisch	<b>Selbststudiumsstunden</b>	105

Der folgende Abschnitt ist nur ausgefüllt, wenn es **genau eine** modulabschließende Prüfung gibt.

<b>Prüfungsleistung</b>	Projektarbeit	<b>Prüfsprache</b>	Deutsch/Englisch
<b>Dauer PL in Minuten</b>		<b>Bewertungssystem PL</b>	Drittelnoten
<b>Lernergebnisse</b>	<ul style="list-style-type: none"> <li>• Die Studenten kennen die Besonderheiten Cloud-nativer Applikationen und deren Programmierung.</li> <li>• Die Studenten können Methoden wie die Zwölf-Faktoren Methode, das IDEAL Modell und DevOps Prinzipien methodisch anwenden.</li> <li>• Die Studenten sind sich der Besonderheiten des Polyglot Programming bewusst und berücksichtigen diese in Cloud-nativen Projekten.</li> <li>• Die Studenten können polyglott entwickelter Services Cloud-nativer Applikationen als standardisierte Deployment Units (Container) bereitstellen.</li> <li>• Die Studenten können Serverless Functions entwickeln und sind sich deren Möglichkeiten und Grenzen bewusst.</li> <li>• Die Studenten kennen gängige Plattformen zur Container Orchestrierung und können mindestens eine davon zum Betrieb polyglott entwickelter Dienste nutzen.</li> <li>• Die Studenten können polyglot entwickelte Dienste mit Mitteln von Integrationssprachen (wie bspw. Ballerina, Jolie) zu Service-of-Services integrieren.</li> </ul>		
<b>Teilnahmevoraussetzungen</b>	Die Studenten haben einen Bachelor in Informatik oder vglb. Studiengängen und entsprechende Kenntnisse in Betriebssystemen (insb. Linux), Programmierung (mind. 1 Programmiersprache fließend) Client-Server-Programmierung und Kommunikationsprotokollen wie HTTP, TCP, UDP.		

Der vorige Abschnitt ist nur ausgefüllt, wenn es **genau eine** modulabschließende Prüfung gibt.

<b>Berücksichtigung von Gender- und Diversity-Aspekten</b>	<ul style="list-style-type: none"> <li>✘ Verwendung geschlechtergerechter Sprache (THL-Standard)</li> <li>✘ Zielgruppengerechte Anpassung der didaktischen Methoden</li> <li>✘ Sichtbarmachen von Vielfalt im Fach (Forscherinnen, Kulturen etc.)</li> </ul>
<b>Verwendbarkeit</b>	
<b>Bemerkungen</b>	Die Berücksichtigung von Gender- und Diversity-Aspekten erfolgt nicht nach formalen Vorgaben, sondern wird im Rahmen des allgemein üblichen und gesellschaftlich akzeptierten Miteinanders praktiziert.

## Lehrveranstaltung: Cloud-native Programmierung (Vorlesung)

(zu Modul: Cloud-native Programmierung)

<b>Lehrveranstaltungsart</b>	Vorlesung	<b>Lernform</b>	Präsenz
<b>LV-Name englisch</b>	Cloud-native Programming (Lecture)		
<b>Anwesenheitspflicht</b>	nein	<b>ECTS-Leistungspunkte</b>	2
<b>Teilnahmebeschränkung</b>	200	<b>Semesterwochenstunden</b>	3
<b>Gruppengröße</b>	200	<b>Arbeitsaufwand in Stunden</b>	60
<b>Lehrsprache</b>	Deutsch/Englisch	<b>Präsenzstunden</b>	30
<b>Studienleistung</b>		<b>Selbststudiumsstunden</b>	30
<b>Dauer SL in Minuten</b>		<b>Bewertungssystem SL</b>	

Der folgende Abschnitt ist nur ausgefüllt, wenn es eine lehrveranstaltungsspezifische Prüfung gibt.

<b>Prüfungsleistung</b>		<b>Prüfungsprache</b>	
<b>Dauer PL in Minuten</b>		<b>Bewertungssystem PL</b>	
<b>Lernergebnisse</b>			
<b>Teilnahmevoraussetzungen</b>			

Der vorige Abschnitt ist nur ausgefüllt, wenn es eine lehrveranstaltungsspezifische Prüfung gibt.

<b>Lehrinhalte</b>	<ul style="list-style-type: none"> <li>• Definition Cloud-nativer Applikationen</li> <li>• Das IDEAL Modell</li> <li>• Die Zwölf-Faktoren Methode</li> <li>• Polyglot Programming</li> <li>• Standardisierte Deployment Units (Container)</li> <li>• Plattformen zur Container Orchestrierung</li> <li>• Konzepte von Container Orchestrierungsplattformen (Pod, Service, Proxy, Service Meshs, Ingress Controller, Load Balancer, etc.)</li> <li>• Serverless Computing</li> </ul> <p>Integration polyglot entwickelter Dienste</p>
<b>Literatur</b>	<ul style="list-style-type: none"> <li>• Nane Kratzke, Cloud-native Computing, Hanser, 2021</li> <li>• Christoph Fehling, Frank Leymann, Ralph Retter, Walter Schupeck, Peter Arbitter, Cloud Computing Patterns, Springer</li> <li>• Cornelia Davis, Cloud-native – Designing change-tolerant software, Manning</li> <li>• Marko Luksa, Kubernetes in Action, Hanser Verlag</li> </ul>
<b>Bemerkungen</b>	

## Lehrveranstaltung: Cloud-native Programmierung (Praktikum)

(zu Modul: Cloud-native Programmierung)

<b>Lehrveranstaltungsart</b>	Praktikum	<b>Lernform</b>	Präsenz
<b>LV-Name englisch</b>	Cloud-native Programming (Practical Course)		
<b>Anwesenheitspflicht</b>	ja	<b>ECTS-Leistungspunkte</b>	3
<b>Teilnahmebeschränkung</b>	24	<b>Semesterwochenstunden</b>	1
<b>Gruppengröße</b>	12	<b>Arbeitsaufwand in Stunden</b>	90
<b>Lehrsprache</b>	Deutsch/Englisch	<b>Präsenzstunden</b>	15
<b>Studienleistung</b>		<b>Selbststudiumsstunden</b>	75
<b>Dauer SL in Minuten</b>		<b>Bewertungssystem SL</b>	

Der folgende Abschnitt ist nur ausgefüllt, wenn es eine lehrveranstaltungsspezifische Prüfung gibt.

<b>Prüfungsleistung</b>		<b>Prüfsprache</b>	
<b>Dauer PL in Minuten</b>		<b>Bewertungssystem PL</b>	
<b>Lernergebnisse</b>			
<b>Teilnahmevoraussetzungen</b>			

Der vorige Abschnitt ist nur ausgefüllt, wenn es eine lehrveranstaltungsspezifische Prüfung gibt.

<b>Lehrinhalte</b>	Siehe Vorlesung (diese werden in praktischen Aufgaben entsprechend vertieft und angewendet)
<b>Literatur</b>	Siehe Vorlesung
<b>Bemerkungen</b>	