BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Birla Institute of Technology & Science, Pilani
Work Integrated Learning Programmes Division
First Semester 2025-2026

Digital Learning Handout

Part A: Content Design

Course Title	Scalable Services
Course No(s)	SE ZG583
Credit Units	5
Credit Model	3-1-1
Course Author	Prof. Akanksha Bharadwaj
Lead Instructor	Akanksha Bharadwaj
Version No:	2.0
Date:	

Course Description:

Software principles related to scalability. Architectures for Scaling. Microservices - design, service discovery, load balancing, API management. Deployment - container configurations and orchestrations, automated deployments of microservices, integration with CI/CD pipelines. Performance: Scaling and load balancing with containers and microservices, Ensuring QoS and SLAs.

Course Objectives

No	Course Objective
CO1	Build competence to design, develop, implement and manage scalable information systems
CO2	Gain understanding of different techniques & tools for building and managing scalable services
CO3	Gain understanding of challenges and best practices in creating and managing scalable services

Text Book(s):

T1	Microservices patterns by Chris Richardson, Manning Publications 2018
T2	Microservices in action by Morgan Bruce & Paulo Pereira, Manning Publications 2018
T3	Building Microservices by Sam Newman, O'Reilly Media 2015

Reference Book(s) & other resources:

R1	https://kubernetes.io/docs/concepts/
R2	The Art of Scalability: Scalable Web Architecture, Processes, and Organizations for the Modern Enterprise, Second Edition by Michael T. Fisher; Martin L. Abbott Published by Addison-Wesley Professional, 2015
R3	Scalability patterns by Microsoft Azure: https://docs.microsoft.com/en-us/azure/architecture/patterns/category/performance-scalability
R4	Scalability Patterns: Best Practices for Designing High Volume Websites by Chander Dhall

Learning Outcomes: Students will be able to

LO1	Understanding of different scenarios where scaling is needed
LO2	Understanding of different approaches to scaling
LO3	Understanding of microservices technology
LO4	Ability to design, develop and deploy microservices based applications

LO5	Understanding of ways to monitor and manage Microservices
LO6	Confident in using tools used in building scalable services

Modular Content Structure

1. **Getting to know Scalability:**
   - Introduction to Performance, Consistency and availability
   - What is scalability?
   - Need for scalable architectures
   - Principles of Scalability
   - Guidelines for Building Highly Scalable Systems
   - Architecturally scalable requirements
   - Challenges for Scalability
   - YouTube case Study
2. **Popular scaling approaches**
   - Managing & processing high volumes of data
     - Partitioning and sharding
     - Distributed data (CAP theorem, NoSQL, HDFS)
     - Distributed Processing (Map reduce, Spark)
   - Managing high velocity data streams (Kafka)
     - Video streaming: Netflix, YouTube, use of CDN
     - Real time analytics: Credit card fraud detection
     - Web conferencing: WebEx, Zoom
     - Edge computing: IoT systems
   - Managing high volume transactions
     - Service Replicas & load balancing
     - Minimizing event processing: Command Query Responsibility Segregation (CQRS)
     - Asynchronous communication
     - Caching techniques: Distributed cache, global cache
   - Scalability features in the Cloud (AWS, Azure, Google)
     - Auto-scaling
     - Horizontal and vertical scaling
     - Use of Load balancers
     - Virtualization
     - Serverless computing
   - Best Practices for Achieving Scalability
3. **Microservices -** Introduction
   - Challenges with Monolith applications
   - Microservices architecture
   - Advantages and disadvantages of Microservices

- Process & organization for microservices

4. **Decomposition strategies**
    - Decomposition by business capability
    - Decomposition by business domain
    - Decomposition guidelines
    - Obstacles to decomposing an application into services
    - Defining service APIs
5. **Communication between Microservices**
    - Inter-service communication
        - Synchronous communication (REST, gRPC)
        - Asynchronous communication
    - Application boundary
        - API gateway
        - API design
        - Creating and versioning APIs
        - API security
        - Backends for frontends
6. **Transaction management**
    - Distributed transactions
    - Implementation
    - Challenges
    - Solutions
    - Sagas
7. **Building with pipelines**
    - Continuous integration
    - Tooling
    - Repository patterns – Multi-repo, mono-repo
8. **Designing reliable microservices**
    - Sources of failure, cascading failures
    - Designing reliable communication: Retires, async. Comm., circuit breakers
    - Maximizing service reliability: Load balancing, Rate limiting (Queues, Throttling)
    - Service mesh
9. **Securing and Testing scalable services**
    - Securing code and repositories
    - Using Authentication and Authorization
    - Unit testing

- Integration testing
- Load testing

10. **Deploying Microservices**
    - Service startup
    - Running multiple instances
    - Adding load balancer
    - Service to host models
        - Single Service Instance to Host
        - Multiple static Service Per Host
        - Multiple scheduled services per host
    - Deploying services without downtime: Canaries, Blue-Green, & rolling deploys
    - Deploying microservices using Serverless deployment
11. **Deployment with Containers**
    - Introduction to containers
    - Containerizing a service
    - Deploying to a cluster
12. **Monitoring**
    - Golden signals
    - Types of metrics
    - Recommended practices
    - Collecting metrics
    - Instrumenting
    - Raising sensible & actionable alerts
    - Using logs & traces
    - Useful info in log entries
    - Tools for logging
    - Logging the right information
    - Tracing interaction between services
    - Visualizing traces
13. **Kubernetes**
    - Dockers for CaaS
    - What is Kubernetes
    - Deployment of Microservices using Kubernetes
    - Scalability in Kubernetes
    - Security in Kubernetes
    - CI/CD using Kubernetes
    - Kubernetes Dashboard

Part B: Learning Plan

Contact	List of Topic Title	Sub-Topics	Reference

Session			
1	**Getting to know Scalability**	● Introduction to Performance, Consistency and availability ● What is scalability? ● Need for scalable architectures ● Principles of Scalability ● Guidelines for Building Highly Scalable Systems ● Architecturally scalable requirements ● Challenges for Scalability ● YouTube case Study	R2 and R4
2	**Popular scaling approaches**	● Managing & processing high volumes of data ○ Partitioning and sharding ○ Distributed data (CAP theorem, NoSQL, HDFS) ○ Distributed Processing (Map reduce, Spark) ● Managing high velocity data streams (Kafka) ○ Video streaming: Netflix, YouTube, use of CDN ○ Real time analytics: Credit card fraud detection ○ Web conferencing: WebEx, Zoom ○ Edge computing: IoT systems	R2
3	**Popular scaling approaches**	● Managing high volume transactions ○ Service Replicas & load balancing ○ Minimizing event processing: Command Query Responsibility Segregation (CQRS) ○ Asynchronous communication ○ Caching techniques: Distributed cache, global cache ● Scalability features in the Cloud (AWS, Azure, Google) ○ Auto-scaling ○ Horizontal and vertical scaling ○ Use of Load balancers ○ Virtualization ○ Serverless computing ● Best Practices for Achieving Scalability	R2
4	**Microservices - Introduction**	● Challenges with Monolith applications ● Microservices architecture ● Netflix case study	T1
5	**Microservices – Introduction** **Decomposition**	● Advantages and disadvantages of Microservices ● Process & organization for Microservices ● Decomposition by business capability ● Decomposition by business domain	T1

	strategies		
6	**Decomposition strategies** **Communication between Microservices**	● Decomposition guidelines ● Obstacles to decomposing an application into services ● Defining service APIs ● Inter-service communication ○ Synchronous communication (REST, gRPC) ○ Asynchronous communication	T1
7	**Communication between Microservices** **Transaction management**	● Application boundary ○ API gateway ○ API design ○ Creating and versioning APIs ○ API security ○ Backends for frontends ● Distributed transactions ● Implementation ● Challenges ● Solutions ● Sagas	T1
8	**Review**	Review of contact session 1 to 7	
9	**Building with pipelines**	● Continuous integration ● Tooling ● Repository patterns – Multi-repo, mono-repo	T3
10	**Designing reliable Microservies**	● Sources of failure, cascading failures ● Designing reliable communication: Retires, async. Comm., circuit breakers ● Maximizing service reliability: Load balancing, Rate limiting (Queues, Throttling) ● Service mesh	T2
11	**Securing and Testing scalable services**	● Securing code and repositories ● Using Authentication and Authorization ● Unit testing ● Integration testing ● Load testing	T1 and T3
12	**Deploying microservices**	● Service startup ● Running multiple instances ● Adding load balancer ● Service to host models ○ Single Service Instance to Host ○ Multiple static Service Per Host ○ Multiple scheduled services per host	T2
13	**Deploying microservices**	● Deploying services without downtime: Canaries, Blue-Green, & rolling deploys	T2

	Deployment with Containers	● Deploying microservices using Serverless deployment ● Introduction to containers ● Containerizing a service ● Deploying to a cluster	
14	**Monitoring**	● Golden signals ● Types of metrics ● Recommended practices ● Monitoring o Collecting metrics o Instrumenting o Raising sensible & actionable alerts ● Using logs & traces o Useful info in log entries o Tools for logging o Logging the right information o Tracing interaction between services o Visualizing traces	T2
15	**Kubernetes**	● Dockers for CaaS ● What is Kubernetes ● Deployment of Microservices using Kubernetes ● Scalability in Kubernetes ● Security in Kubernetes ● CI/CD using Kubernetes ● Kubernetes Dashboard	R1 and T3
16	**Review**	Review of contact session 9 to 15	

Experiential Learning Components:

Describe objective, outcome of Experiential Learning Component and the lab infrastructure needed (virtual, remote, open source etc...) number of lab exercises needed, etc.

1. Lab work: 10
2. Project work: 0
3. Case Study: 4 Webinars
4. Simulation: 0
5. Work Integrated Learning Assignment- 2 Assignments
6. Design work/ Field work: 0

Objective of Experiential Learning Component:
Hands on sessions on implementation of microservices based application and its deployment
Scope of Experiential Learning Component:
Technology: No restriction on technology stack
Lab Infrastructure: Online/ Open source
Case studies:

1. YouTube: https://www.womenwhocode.com/blog/youtube-system-architecture

2. Netflix: https://netflixtechblog.com/tagged/cloud-architecture
3. Amazon Prime: https://aws.amazon.com/solutions/case-studies/amazon-prime-video/ , https://www.primevideotech.com/video-streaming/scaling-up-the-prime-video-audio-video-monitoring-service-and-reducing-costs-by-90
4. Google: https://highscalability.com/google-architecture/
5. Facebook: https://highscalability.com/facebook-an-example-canonical-architecture-for-scaling-billi/ , https://medium.com/swlh/an-introduction-to-facebooks-system-architecture-47cfcf597101

List of Experiments:

Lab No	**Lab Objective**	**Session Reference**
1	Design and develop a Microservices based application	4 and 5
2	Attach a database to a service and perform basic CRUD operations	6
3	Exploring the communication between services by using shared database pattern	6
4	Exploring the communication using RabbitMQ	6 and 7
5	Deploying the application using Docker desktop	12 and 13
6	Configuring Minikube and running a local cluster	15
7	Deploying application on Minikube	15

Evaluation Scheme:

Legend: EC = Evaluation Component; AN = After Noon Session; FN = Fore Noon Session

Evaluation Component	Name (Quiz, Lab, Project, Mid-term exam, End semester exam, etc.)	Type (Open book, Closed book, Online, etc.)	Weigh t	Duration	Day, Date, Session, Time
EC – 1*	Quiz	Online	10%	1 week	September 01-10, 2025
	Assignment/Lab Assignment / Lab Exams	Online	20 %	10 days	November 01-10, 2025
EC - 2	Mid-Semester Test	Closed Book	30%	2 hours	20/09/2025 (FN)
EC - 3	Comprehensive Exam	Open Book	40%	2 ½ Hours	29/11/2025 (FN)

EC1* (20% - 30%): Quiz (optional): 5-10 %, Lab Assignment/Assignment: 20% - 30%

Syllabus for Mid-Semester Test (Closed Book): Topics in Contact session: 1 to 8

Syllabus for Comprehensive Exam (Open Book): All topics

Important Links and Information:

eLearn Portal: https://elearn.bits-pilani.ac.in

Students must visit the eLearn portal regularly and stay updated with the latest announcements and deadlines.

Contact Sessions: Students should attend the online lectures as per the schedule provided on the eLearn portal.

Evaluation Guidelines:

1. EC-1 consists of either two Assignments or three Quizzes. Students will attempt them through the course pages on the eLearn portal. Announcements will be made on the portal in a timely manner.
2. For Closed Book tests: No books or reference material of any kind will be permitted.
3. For Open Book exams: “open book” means text/ reference books (publisher copy only) and does not include any other learning material. No other learning material will be permitted during the open book examinations. For Detailed Guidelines refer to the attached document.
   EC3 Guidelines
4. If a student is unable to appear for the Regular Test/Exam due to genuine exigencies, the student should follow the procedure to apply for the Make-Up Test/Exam, which will be made available on the eLearn portal. The Make-Up Test/Exam will be conducted only at selected exam centres on the dates to be announced later.

It shall be the responsibility of the individual student to be regular in maintaining the self-study schedule as given in the course handout, attend the online lectures, and take all the prescribed evaluation components such as Assignments/Quizzes, Mid-Semester Tests and Comprehensive Exams according to the evaluation scheme provided in the handout.

****