COURSE PLAN

UPES/Course Plan/Theory & lab/Ver 10/ Effective December 2025

Name of School

Name of the Faculty	Prateek Raj Gautam
Designation	Assistant Professor Senior Scale
Cluster	CSO
Program	B.Tech.
Course	Containerization and DevOps
Course Code	CSDV3018P
No. of credits	4
Semester	VI
Session	Jan – May
Academic Year	2025-26

Signature of Faculty Member

Name: _____

Designation: _____

Date: _____

Signature of Cluster Head

Name: _____

Designation: _____

Date: _____

COURSE PLAN

Prerequisite	**Cloud Computing Architecture and Deployment Models - CSVT3029P**	
Credit	**4**	
Lecture	**Tutorial**	**Practical**
4	**00**	**0**

A. The expected Program Outcome are:

PO1	**Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO2	**Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO3	**Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
PO4	**Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO5	**Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
PO6	**The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
PO7	**Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO8	**Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9	**Individual and teamwork:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO10	**Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO11	**Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO12	**Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

B. Expected Program specific Outcome are:

PSO1	Perform system and application programming using computer system concepts, the principles of data structures, algorithm development, problem-solving, and optimization techniques.
PSO2	Apply software development and project management methodologies, integrating principles from both front-end and back-end development, and effectively utilize contemporary tools and technologies.
PSO3	Exhibit a commitment to ethical practices, societal responsibilities, and continuous learning, contributing to the advancement of technology and addressing challenges in diverse computing domains.

C. **The expected Course Outcomes are:**

THEORY	
CO 1	**Explain** the fundamentals of containerization, virtualization, and DevOps principles, including their architecture, concepts, and role in modern software development.
CO 2	**Apply** Docker concepts to create, manage, and deploy containers by using Docker images, Dockerfiles, storage, and networking features.
CO 3	**Implement** container orchestration techniques by configuring and managing services using orchestration platforms such as Docker Swarm and Kubernetes.
CO 4	**Analyze and evaluate** DevOps adoption strategies, tools, and practices to assess their impact on business outcomes and cross-functional team collaboration.

D. CO-PO Relationship Matrix

Indicate the relationships by1- Slight (low) 2- Moderate (Medium) 3-Substantial (high)

Program Outcomes / Course Outcomes	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO 11	PO 12	PSO 1	PSO 2	PSO3
CO 1	3	2	--	--	2	--	--	1	--	--	--	2	3	2	1
CO 2	3	3	2	2	3	--	--	--	1	2	1	1	3	3	--
CO 3	3	3	3	2	3	--	--	--	2	2	2	1	3	3	--
CO 4	1	2	1	3	1	2	2	2	3	3	3	3	1	3	3
Average	2.50	2.50	2.00	2.33	2.25	2.00	2.00	1.67	2.00	2.33	2.00	1.75	2.50	2.75	2.00

E. Course Outcomes assessment plan:

a) Mapping of Course Outcomes with assessment tools

Components / Course Outcomes	Assignment	Test/Quiz	Mid Semester	End Semester	Any other
CO 1	☑	☐	☑	☑	☐
CO 2	☐	☑	☑	☑	☐
CO3	☑	☐	☐	☑	☐
CO4	☐	☑	☐	☑	☐

b) Assessment plan and Analysis: THEORY

	Assessment Tools	Planned	Actual	Highest (Score)	Lowest (Score)
CIA	Quiz 1	2 weeks Before Mid Sem			
	Quiz 2	2 weeks After Mid Sem			
	Assignment 1	3 weeks Before Mid Sem			
	Assignment 2	3 weeks After Mid Sem			
	Class Test 1				
	Class Test 2				
	Group Project 1/PBL 1	After mid Sem			
	Group Project 2/PBL 2				
	Case Studies 1				
	Case Studies 2				
Mid Term Exam					
End Term Exam					

c) Assessment plan and Analysis: Lab

	Assessment Tools	Planned	Actual	Highest (Score)	Lowest (Score)
CIA	Viva- Voce	**---**			
	Lab Records	**---**			

F. Course Syllabus Template:

Syllabus Template for Theory

Unit Number	Content
Unit I	**Introduction to Containerization and DevOps** Virtualization and its types-Server Virtualization, Operating System Virtualization **Containers & DevOps:** Introduction to Containers, Understanding DevOps Principles and Practices, Benefits of Containerization in DevOps, Docker vs. Virtualization, Different vendors for containers in the market, **Docker:** Introduction to Docker: What is Docker and What isn't Docker?, Overview of Docker editions, Installation of the Docker engine, Docker terminology, Docker community, Docker community edition, Docker enterprise edition, Build Kit features, Docker compose, Docker architecture-Docker host, Docker Daemon, Docker Hub, Docker API, Docker Objects, Docker and DevOps tools **Software Designing Architectural Approaches:** Server based and Serverless architecture-Monolithic, Service Oriented Architecture, Microservices, Comparison of different architectural approaches, Docker and Microservices-apart and together
Unit II	**Docker Fundamentals** **Basic Commands**: Docker container Lifecycle commands, Docker desktop , Checking docker version, Running your first NGINX application, Docker repository, Docker tags, Docker TAG examples, Docker TAG scenario, Tagging scheme, Docker images, Layers, Docker file, Docker file instructions, Managing containers and images, Creating Docker Images, Dockerfile Best Practices, Pushing Images to Docker Hub, Running your own Docker container.h **Data Management in Docker:** Persisting data in docker, Approaches: Volumes, bind mount, Differences, Volumes-creation, listing, --mount flag, -v flag, removing, inspecting bind mount- creation, --mount flag, -v flag, removing, tmpfs mount, Use case scenarios volumes, bind mount and tmpfs mount. **Networking in Docker:** Networking in Docker, Docker network drivers- bridge, host, overlay

	ipvlan, macvlan, Publishing ports, IP address and host name, DNS services, Network Drivers use case summary, creating and removing a user-defined bridge, managing a user-defined bridge- connect a container to a user defined bridge, Disconnect a container from a user-defined bridge, Connect a container to the default bridge network. **Continuous Integration**: Docker as a build environment, GitHub Actions, Building CI/CD Pipleine using Git Hub actions.
Unit III	**Automation and Orchestration** **Container Orchestration need and Overview:** Key concepts in orchestration, popular orchestra platforms: Swarm Docker, Kubernetes, Apache mesos **Docker Compose:** Features, use cases, history and using Docker Compose **Docker Swarm:** Feature highlights, Swarm mode key concepts-swarm, nodes, services & tasks, load balancing; Create a swarm, Add nodes to swarm, Deploy a service, Inspect the service, Scale the service, Delete the service, Apply rolling updates, Drain a node, Use swarm mode routing mesh **Kubernetes:** Overview, Traditional deployment era, virtualized deployment era , container deployment era, Need of Kubernetes, what Kubernetes is not?, Kubernetes components: control plane components, Node components, Addons, Kubernetes API, Cluster Architecture-Nodes, Communication between nodes and the control Plane, Controllers, Leases, Cloud Controller Manager, Container runtime Interface, Garbage Collection, Containers: Images, Container Environment, Runtime class, Container Lifecycle hooks, Workloads- Pods and Workload Resources, Services, Load balancing and Networking, Storage, Configuration, Policies and security **Case Study:** Docker Swarm vs Kubernetes **Case Study:** Amazon ECS and EKS services
Unit IV	**DevOps: Principles and Practices** Overview, Working, Benefits, DevOps history, DevOps principles and lifecycle, DevOps practice, DevOps adoption: Deming, lean manufacturing, and Kaizen, Lean manufacturing, Lean standards of manufacturing, DevOps: IBM view, Four DevOps adoption paths- Steer adoption path, Develop and test adoption path,

	Collaborative development, Continuous testing, Way to deployment, Continuous customer feedback and optimization, DevOps architecture and resilience, Cloud resiliency, DevOps resiliency, Four stages of the resilience process-Detect, DevOps style; Alert with a cloud and DevOps mindset; Respond & recover using automation and appropriate failover strategy; Refine & test, achieving incremental improvements
Unit V	**DevOps Adoption and Business Patterns** Business needs for DevOps , DevOps teams and cross functioning of teams, Silos in the world of software development and their role in project delivery, DevOps teams and cross functioning of teams, Application team v/s. Platform team, System admins and other stakeholders, Continuous integration vs continuous deployment vs continuous delivery, DevOps tools- Continuous development, Continuous integration, Continuous Testing, Continuous Deployment, Continuous Monitoring, Lean Thinking and Methods – Kaizen, Agile Vs DevOps, DevOps impact on developers, DevOps impact on operations, Successful DevOps adoption, Challenges of DevOps adoption, Introduction to Kanban, Types of kanban board, create a kanban board, Kanban with IBM tools. Scrum application delivery pipeline and support team, an orchestration framework for continuous delivery, Software release plan, Feedback and learning from feedback and improving the delivery, DevOps toolchain, DevSecOps, DevOps vs SRE. Select the right tool for DevOps: Docker, Kubernetes, Puppet, Ansible, Other tools, DevOps monitoring tools, Version control, and code repository
Unit VI	**DevOps Tools** **Version Control Tools:** GitHub-GitLab-BitBucket, GitHub-CLI, Desktop, Branches, forks, and Pull requests, Repositories, GitHub actions, GitHub packages, Webhooks, API **Continuous Integration Tools:** Jenkins, TravisCI, Jenkins- Introduction to Jenkins, Jenkins architecture and components: Installation and setup of Jenkins, Creating and Configuring Jenkins Jobs Jenkins Pipelines Source Code Management (SCM) Integration Building and Testing with Jenkins **Infrastructure as Code (IaC) Tools:** Understanding IaC, Popular IaC tools-Terraform, Ansible

	Monitoring and Logging Tools: Monitoring and logging in DevOps**,** Popular monitoring and logging tools- Prometheus, ELK stack **Collaboration in a DevOps Team:** Popular collaboration and communication tools- Slack, Microsoft Teams, Integrating tools into the DevOps workflow **Case study:** CI/CD technique.

COURSE PLAN DELIVERY

UNIT-I

	Session Plan	Actual Delivery			
Lecture	Topics to be Covered	Lecture	Date	Topics Covered	CO covered
1	Virtualization and its types: Server Virtualization, Operating System Virtualization				
2	Introduction to Containers, Understanding DevOps Principles and Practices				
3	Benefits of Containerization in DevOps, Docker vs. Virtualization				
4	Different vendors for containers in the market				
5	**Docker:** Introduction - What is Docker and What isn't Docker? Overview of Docker editions				
6	Installation of the Docker engine, Docker terminology, Docker community				
7	Docker community edition, Docker enterprise edition, Build Kit features				
8	Docker compose, Docker architecture: Docker host, Docker Daemon, Docker Hub				
9	Docker API, Docker Objects, Docker and DevOps tools				
10	**Software Designing Architectural Approaches:** Server based and Serverless architecture - Monolithic, Service Oriented Architecture, Microservices, Comparison of approaches, Docker and Microservices				

COURSE PLAN DELIVERY

UNIT-II

	Session Plan	Actual Delivery			
Lecture	Topics to be Covered	Lecture	Date	Topics Covered	CO covered
11	**Basic Commands:** Docker container Lifecycle commands, Docker desktop, Checking docker version				
12	Running your first NGINX application, Docker repository, Docker tags, Docker TAG examples				
13	Docker TAG scenario, Tagging scheme, Docker images, Layers				
14	Dockerfile, Docker file instructions, Managing containers and images				
15	Creating Docker Images, Dockerfile Best Practices				
16	Pushing Images to Docker Hub, Running your own Docker container				
17	**Data Management in Docker:** Persisting data in docker, Approaches: Volumes, bind mount, Differences				
18	Volumes: creation, listing, --mount flag, -v flag, removing, inspecting				
19	Bind mount: creation, --mount flag, -v flag, removing, tmpfs mount				
20	Use case scenarios for volumes, bind mount and tmpfs mount				
21	**Networking in Docker:** Networking in Docker, Docker network drivers: bridge, host, overlay				
22	ipvlan, macvlan, Publishing ports, IP address and host name, DNS services, Network Drivers use case summary				

COURSE PLAN DELIVERY

UNIT-III

	Session Plan	Actual Delivery			
Lecture	Topics to be Covered	Lecture	Date	Topics Covered	CO covered
23	**Container Orchestration need and Overview:** Key concepts in orchestration, popular orchestration platforms: Swarm Docker, Kubernetes, Apache mesos				
24	**Docker Compose:** Features, use cases, history and using Docker Compose				
25	**Docker Swarm:** Feature highlights, Swarm mode key concepts: swarm, nodes, services & tasks, load balancing				
26	Create a swarm, Add nodes to swarm, Deploy a service, Inspect the service				
27	Scale the service, Delete the service, Apply rolling updates, Drain a node				
28	Use swarm mode routing mesh				
29	**Kubernetes:** Overview, Traditional deployment era, virtualized deployment era, container deployment era				
30	Need of Kubernetes, what Kubernetes is not?, Kubernetes components: control plane components				
31	Node components, Addons, Kubernetes API, Cluster Architecture: Nodes				
32	Communication between nodes and the control Plane, Controllers, Leases, Cloud Controller Manager				

33	Container runtime Interface, Garbage Collection, Containers: Images, Container Environment, Runtime class				
34	**Case Study:** Docker Swarm vs Kubernetes, **Case Study:** Amazon ECS and EKS services				

COURSE PLAN DELIVERY

UNIT-IV

	Session Plan	Actual Delivery			
Lecture	Topics to be Covered	Lecture	Date	Topics Covered	CO covered
35	Overview, Working, Benefits, DevOps history, DevOps principles and lifecycle				
36	DevOps practice, DevOps adoption: Deming, lean manufacturing, and Kaizen				
37	Lean manufacturing, Lean standards of manufacturing, DevOps: IBM view				
38	Four DevOps adoption paths: Steer adoption path, Develop and test adoption path				
39	Collaborative development, Continuous testing, Way to deployment				
40	Continuous customer feedback and optimization				
41	DevOps architecture and resilience, Cloud resiliency, DevOps resiliency				
42	Four stages of the resilience process: Detect (DevOps style); Alert with a cloud and DevOps mindset; Respond & recover using automation; Refine & test, achieving incremental improvements				

COURSE PLAN DELIVERY

UNIT-V

	Session Plan	Actual Delivery			
Lecture	Topics to be Covered	Lecture	Date	Topics Covered	CO covered
43	Business needs for DevOps, DevOps teams and cross functioning of teams				
44	Silos in software development and their role in project delivery, Application team v/s. Platform team, System admins and other stakeholders				
45	Continuous integration vs continuous deployment vs continuous delivery				
46	DevOps tools overview: Continuous development, Continuous integration, Continuous Testing, Continuous Deployment, Continuous Monitoring				
47	Lean Thinking and Methods - Kaizen, Agile Vs DevOps, DevOps impact on developers and operations				
48	Successful DevOps adoption, Challenges of DevOps adoption, Introduction to Kanban				
49	Types of kanban board, create a kanban board, Kanban with IBM tools, Scrum application delivery pipeline				
50	Orchestration framework for continuous delivery, Software release plan, Feedback and learning, DevOps toolchain, DevSecOps, DevOps vs SRE				

COURSE PLAN DELIVERY

UNIT-VI

	Session Plan	Actual Delivery			
Lecture	Topics to be Covered	Lecture	Date	Topics Covered	CO covered
51	**Version Control Tools:** GitHub, GitLab, BitBucket, GitHub-CLI, Desktop				
52	Branches, forks, and Pull requests, Repositories, GitHub actions, GitHub packages				
53	Webhooks, API				
54	**Continuous Integration Tools:** Jenkins, TravisCI, Jenkins - Introduction to Jenkins				
55	Jenkins architecture and components: Installation and setup of Jenkins, Creating and Configuring Jenkins Jobs				
56	Jenkins Pipelines, Source Code Management (SCM) Integration, Building and Testing with Jenkins				
57	**Infrastructure as Code (IaC) Tools:** Understanding IaC, Popular IaC tools: Terraform, Ansible				
58	**Monitoring and Logging Tools:** Monitoring and logging in DevOps, Popular tools: Prometheus, ELK stack				
59	**Collaboration in a DevOps Team:** Popular collaboration tools: Slack, Microsoft Teams, Integrating tools into DevOps workflow				
60	**Case study:** CI/CD technique				

PERIODIC MONITORING- THEORY

Actual date of completion and remarks, if any

Components		From	To	From	To	From	To
Duration (Mention from and to dates)							
Percentage of Syllabus covered							
Lectures	Planned						
	Taken						
Tutorials	Planned						
	Taken						
Test/quizzes	Planned						
	Taken						
	CO's Addressed						
	CO's Achieved						
Assignments	Planned						
	Taken						
	COs Addressed						
	Cos						

Observations(If any)_____

_____

_____

Signature of Faculty **Signature of Head**

Date **Date**

PLANNING FOR REMEDIAL CLASSES

Sl. No.	Name of Student	Roll No.	Sap ID	Mid Sem Marks	Remedial Classes Held							Class test based on Remedial Classes	End Sem Marks	Improvement (Y/N)
					Date									
					Venue									
					Time									

Signature of Faculty

Signature of Head

G. Target

Target	50% (marks)
Level-1	40% (population)
Level-2	50% (population)
Level-3	60% (population)

H. Method of Evaluation*

UG/PG (Indicative)	
Quizzes/Tests, Assignments (50%)	
Mid Examination (20%)	
End examination (30%)	

*It can be revised as per the assessment scheme of the respective School/Course & In case of the courses running under continuous assessment this scheme is not applicable.

I. Passing Criteria

Scale	PG	UG
Out of 10point scale	SGPA – "6.00" in each semester CGPA – "6.00" Min. Individual Course Grade – "C" Course Grade Point – "4.0"	SGPA – "5.0" in each semester CGPA – "5.0" Min. Individual Course Grade – "C" Course Grade Point – "4.0"

*for PG, passing marks are 40/100 in a paper (Composite)

*for UG, passing marks are 35/100 in a paper (Composite)

J. References:

Text Books	Container Orchestration & Infrastructure Automation, IBM ICE Publications.
	James Turnbull, "The Docker Book: Containerization is the new virtualization", First Edition, Shroff Publishers, 2019.
	Jason Cannon, "Docker: A Project-Based Approach to Learning", Independently published, 2021.
	Deepak Gaikwad, Viral Thakkar, "DevOps Tools from Practitioner's Viewpoint", Wiley, 2019.

Web resources	
	https://upessocs.github.io/#dir=/Lectures/Containerization%20and%20DevOps/&file=list.txt
Journals	
Reference books	Ian Miell and Aidan Hobson Sayers, “Docker in Practice”, 2nd Edition , Manning Publications, 2019.
MOOCs, Online courses	

SUGGESTIONS FOR FACULTY

- Faculty should keep track of the students with low attendance and counsel them regularly.
- The course coordinator will arrange to communicate the short attendance (as per UPES policy) cases to the students and their parents monthly.
- Topics covered in each class should be recorded in the table of RECORD OF CLASS TEACHING (Suggested Format).
- Internal assessment marks should be communicated to the students twice in a semester.
- The file will be audited by respective IQAC members for theory as well as for lab as per schedule.
- The faculty is required to maintain these files for a period of at least three years.
- This register should be handed over to the head of department, whenever the faculty member goes on long leave or leaves the Colleges/University.
- For labs, continuous evaluation format (break-up given in the guidelines for result preparation in the same file) should be followed.
- The department should monitor the actual execution of the components of continuous lab evaluation regularly.
- Instructor should maintain record of experiments conducted by the students in the lab weekly.
- Instructor should promote students for self-study and to make concept diary, due weightage in the internal should be given under faculty assessment for the same.
- Course outcome assessment: To assess the fulfilment of course outcomes two different approaches have been decided. Degree of fulfillment of course outcomes will be assessed in different ways through direct assessment and indirect assessment. In Direct Assessment, it is measured through quizzes, tests, assignment, Mid-term and/or End-term examinations. It is suggested that each examination is designed in such a way that it can address one or two outcomes (depending upon the course completion). Indirect assessment is done through the student survey which needs to be designed by the faculty (sample format is given below) and it shall be conducted towards the end of course completion. The evaluation of the achievement of the Course Outcomes shall be done by analyzing the inputs received through Direct and Indirect Assessments and then corrective actions suggested for further improvement.
- At the completion of the course, course attainment and other documents should be shared with the program coordinator for computation of Program attainment.
- At the completion of the course Faculty members are suggested to share the innovative teaching techniques along with the course plan (format provided by IQAC).
- Faculties are encouraged to share the master/expert classes evidence (as per the event report format)
- Faculties are also encouraged to include MOOCs,,SWAYAM any other online content and share the evidence of MOOCs courses /online courses referred (as per the event report format).
- Faculties are encouraged to share the evidence related to interventions or initiatives focusing the unique/slow and Fast Leaners along with Course Completion files.

INDIRECT ASSESSMENT

Sample format for Indirect Assessment of Course outcomes:

NAME:
ENROLLMENT NO:
SAP ID:
COURSE:
PROGRAM:

Please rate the following aspects of course outcomes of ----------.

Use the scale 1-3*

Course Outcomes	Statement	1	2	3
CO1	**Explain** the fundamentals of containerization, virtualization, and DevOps principles, including their architecture, concepts, and role in modern software development.			
CO2	**Apply** Docker concepts to create, manage, and deploy containers by using Docker images, Dockerfiles, storage, and networking features.			
CO3	**Implement** container orchestration techniques by configuring and managing services using orchestration platforms such as Docker Swarm and Kubernetes.			
CO4	**Analyze and evaluate** DevOps adoption strategies, tools, and practices to assess their impact on business outcomes and cross-functional team collaboration.			

* 1 → Low level of attainment

2 → Moderate level of attainment

3 → High level of attainment