Course Code	Course name	L	T	P	C
CSDV3018P	Containerization and DevOps	4	0	0	4
Total Units to be Covered: 06	Total Contact Hours: 60				
Prerequisite(s):	Cloud Computing Architecture and Deployment Models - CSVT3029P	Syllabus version: 1.0			

Course Objectives

The objective of this course are

1. To understand Containerization with Docker, emphasizing automation and orchestration.

2. To master DevOps fundamentals for collaborative and automated software delivery.

3. To gain practical knowledge of DevOps applications in real-world scenarios.

4. To explore DevOps principles, lifecycles, and essential tools for efficient software development and deployment.

Course Outcomes

On completion of this course, the students will be able to

CO1: Comprehensive understanding of containerization and the principles of DevOps.

CO2: Apply Docker fundamentals by creating and managing Docker containers, demonstrating comprehension and application of containerization concepts.

CO3: Demonstrate a deep understanding of container orchestration and apply it practically by creating and managing services using container orchestration platforms.

CO4: Assess DevOps adoption strategies, analyze its impact on business, and synthesize cross-functional team dynamics and tool selection within the DevOps context.

CO-PO Mapping

Program Outcomes Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	1	-	-	2	-	-	-	-	-	-	1	-	-	3
CO 2	-	1	-	-	2	-	-	-	-	-	-	1	2	1	3
CO 3	-	1	-	-	2	-	-	-	-	-	-	1	2	1	3
CO4	-	1	-	-	2	-	-	-	-	-	-	1	1	1	3
Average	-	1	0.75	-	2	-	-	-	-	-	-	1	1.25	0.75	3

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **“_” means there is no correlation**

Syllabus

Unit I: Introduction to Containerization and DevOps **8 Lecture Hours**
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 **12 Lecture Hours**
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 12 Lecture Hours

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 8 Lecture Hours

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 8 Lecture Hours

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 12 Lecture Hours

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.

Total lecture Hours 60

Textbooks

1. Container Orchestration & Infrastructure Automation, IBM ICE Publications.
2. James Turnbull, “The Docker Book: Containerization is the new virtualization”, First Edition, Shroff Publishers, 2019.
3. Jason Cannon, “Docker: A Project-Based Approach to Learning”, Independently published, 2021.

4. Deepak Gaikwad, Viral Thakkar, “DevOps Tools from Practitioner's Viewpoint”, Wiley, 2019.

Reference Books

1. Ian Miell and Aidan Hobson Sayers, “Docker in Practice”, 2nd Edition , Manning Publications, 2019.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	**IA**	**MID SEM**	**End Sem**	**Total**
Weightage (%)	50	20	30	100

Course Code	Course name	L	T	P	C
CSDV3118P	**Containerization and DevOps Lab**	0	0	2	1
Total Units to be Covered: 12	**Total Contact Hours: 30**				
Prerequisite(s):	**Cloud Computing Architecture and Deployment Models Lab - CSVT3129P**	**Syllabus version: 1.0**			

Course Objectives

1. To demonstrate on Containerization technology and use of Docker container automation and orchestration.
2. To gain practical knowledge and hands-on experience on DevOps tools, their roles and importance in the DevOps lifecycle, including continuous integration, infrastructure provisioning, configuration management, and code quality analysis.

Course Outcomes

At the end of this course student should be able to
CO1. Demonstrate the usage of Docker Containers.
CO2. Demonstrate orchestration of Containers images and microservices.
CO3. Apply DevOps tools to effectively implement DevOps practices, including continuous integration, infrastructure provisioning, configuration management, and code quality analysis.

CO-PO Mapping

Program Outcomes Course Outcomes	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO1	PSO2	PSO3
CO 1	1	-	-	1	2	-	-	1	1	-	-	1	-	-	3
CO 2	1	-	-	1	2	-	-	1	1	-	-	1	-	-	3
CO 3	1	-	-	1	2	-	-	1	1	-	-	1	-	-	3
Average	1	-	-	1	2	-	-	1	1	-	-	1	-	-	3

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

List of Experiments

Experiment 1: Introduction to Dockers

a. Setup & Running a Container
b. Performance evaluation of Virtual box machine deployment and Dockers containers deployment.

Experiment 2: Docker Installation, configuration, Running Images

Experiment 3: Deploying web applications with Docker

Experiment 4: Dockerfile - Containerizing application, Building Images, Tagging, Publishing

Experiment 5: Docker - Volumes, Env, Monitoring (Docker stats), Docker Networks

Experiment 6: Docker Compose - Installation, Creating Compose files, Running Images using Docker-compose

Experiment 7: Create a CI CD pipeline for deploying web application using jenkins

Experiment 8: Ansible

Experiment 9: Chef

Experiment 10: Install, setup and run SonarQube for a local scan.

Experiment 11: Orchestration using Docker compose on multi container applications.

Experiment 12: Study and Analyse container orchestration using Kubernetes.

Total Lab hours 30

Text Books

1. IBM ICE Publications
2. The Docker Book: Containerization is the new virtualization by James Turnbull, Shroff Publishers, First Edition, 2019
3. Docker: A Project-Based Approach to Learning by Jason Cannon, **September 2021**
4. DevOps Tools from Practitioner's Viewpoint, by Deepak Gaikwad, Viral Thakkar, Wiley, Jan 2019

Reference Books

1. Ian Miell, and Aidan Hobson Sayers, “Docker in Practice”, 2nd Edition , Manning Publications, 2019.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continuous Assessment

Components	Quiz & Viva	Performance & Lab Report
Weightage (%)	50 %	50 %