



TECHNICAL SEMINAR

# Cloud-Native Applications

Microservices, containers and DevOps

---

PRESENTED BY  
**Ishita Verma, Roll No. 21CS045**

GUIDE  
**Dr. S. Banerjee**

DATE  
**March 2026**

# Agenda

---

**01**

**Overview**

**02**

**Core concepts**

**03**

**Applications**

**04**

**Challenges**

**05**

**Future**

---

# OVERVIEW

01

# Introduction

---

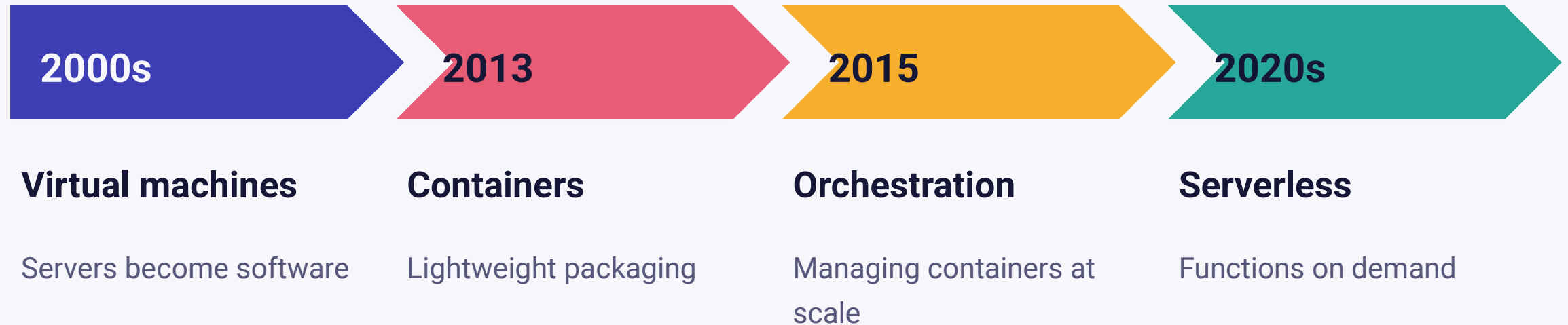
- ▶ Cloud-native apps are built to run on modern cloud platforms
- ▶ They use microservices, containers and automation
- ▶ They scale and update without downtime
- ▶ Most new software is designed this way



Image

# Evolution

---



---

**CORE CONCEPTS**

02

# Key concepts

---

**01**

## **Microservices**

Small services that do one job each.

**02**

## **Containers**

Packaged apps that run anywhere.

**03**

## **CI/CD**

Automated build, test and deploy.

# How it works

---

- |   |                |                             |
|---|----------------|-----------------------------|
| 1 | <b>Code</b>    | Small, independent services |
| 2 | <b>Package</b> | Build container images      |
| 3 | <b>Deploy</b>  | Orchestrator runs them      |
| 4 | <b>Observe</b> | Logs, metrics, traces       |

# Architecture

---

## Services, containers and an orchestrator.

Each microservice ships in its own container. An orchestrator schedules containers across machines, restarts failures and scales services up or down with traffic.



Architecture diagram

---

**APPLICATIONS**

03

# Applications

---



Image 1

## E-commerce

Scale during big sales.



Image 2

## Banking apps

Release features weekly.

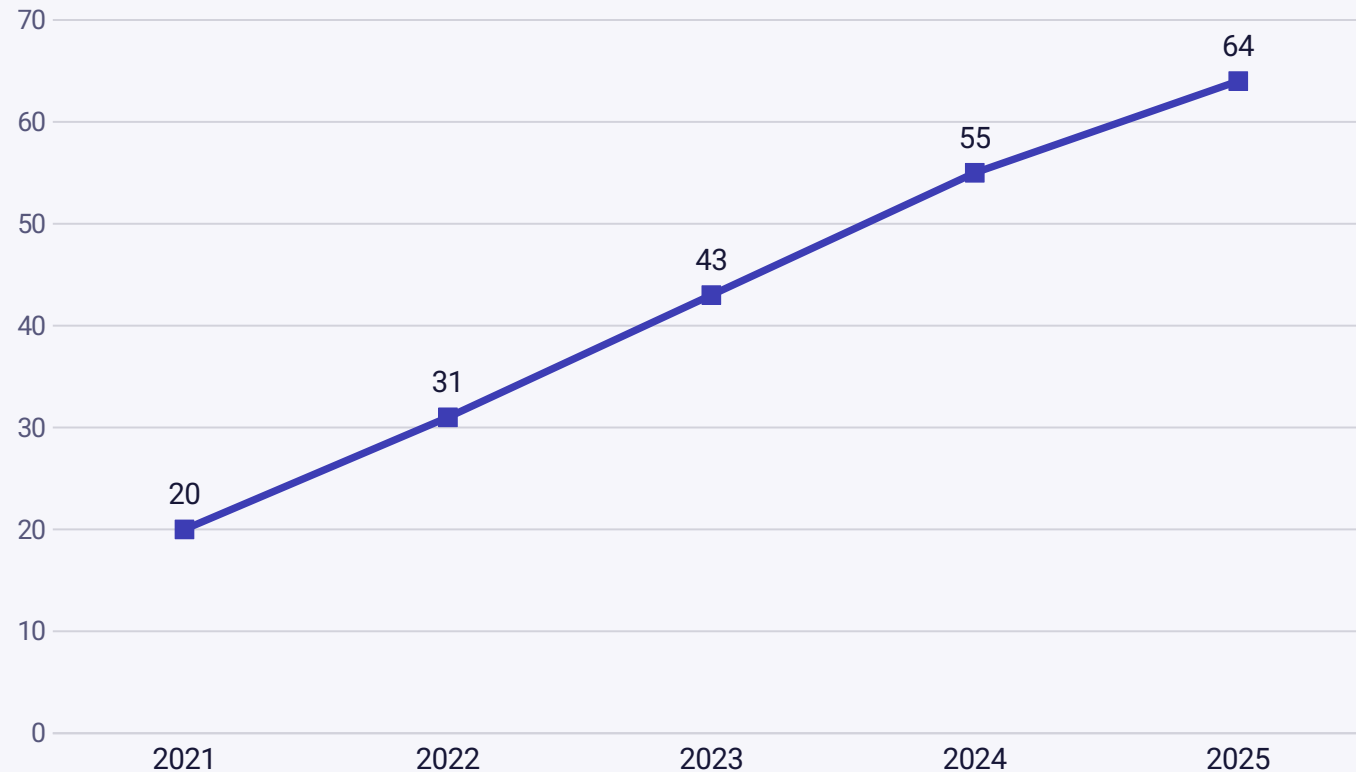


Image 3

## Streaming

Serve millions at once.

# Adoption trend



Organisations running containers in production (sample data).

*Sample data for illustration — replace with your own figures.*

---

**CHALLENGES**

04

# Advantages and limitations

## Advantages

- Scales with demand
- Faster releases
- Resilient to failures
- Portable across clouds

vs

## Limitations

- More moving parts
- Needs DevOps skills
- Monitoring complexity
- Cost control

# Open challenges

---

**01**

## **Complexity**

Many services to manage.

**02**

## **Skills**

Training needed.

**03**

## **Security**

Supply-chain risks.



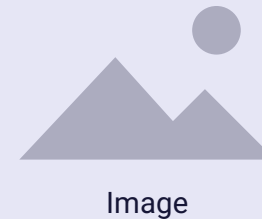
**Platform engineering will  
make cloud-native easier  
for every team.**

Looking ahead

# References

---

- ▶ [1] Survey paper on the topic, IEEE Access, 2024.
- ▶ [2] Textbook chapter covering fundamentals, 2023.
- ▶ [3] Recent conference paper on applications, 2025.
- ▶ [4] Industry white paper (generic reference).



# Questions?

Thank you for listening.

Ishita Verma

Department of Computer Science

