

Andhra Kesari University

CLOUD COMPUTING MINOR SYLLABUS

(W.e.f. 2025–26 Batch)

I. PROGRAMME STRUCTURE

The Cloud Computing Minor consists of six courses distributed across Semesters III–VI. Each course is designed as a 3 Theory + 2 Practical components with 4 credits. The theory component is planned for 45 hours per course, with approximately 9 hours allotted to each of the five units. This 9-hours-per-unit distribution is a teaching/completion plan for the syllabus.

Course	Semester	Course Title	Theory + Practical	Credits
Minor–I	III	Cloud Computing Fundamentals	3T + 2P	4
Minor–II	IV	APEX AND VISUALFORCE PROGRAMMING	3T + 2P	4
Minor–III	V	Cloud Infrastructure, Virtualization and AWS	3T + 2P	4
Minor–IV	V	BUSINESS INTELLIGENCE FOR CLOUD	3T + 2P	4
Minor–V	VI	Cloud Security and DevOps	3T + 2P	4
Minor–VI	VI	Cloud-Native Applications, Storage and Serverless Computing	3T + 2P	4

Total Minor Credits: 24

II. GENERAL COURSE DESIGN

- Five theory units per course.
- Approximately 9 teaching hours per unit; total 45 theory hours per course.
- Unit topics are presented in-line in the conventional syllabus style.
- Each unit contains a manageable range of approximately 10–18 content components, allowing the teacher to adjust depth according to classroom needs.
- Practical component: 15–20 experiments per course;
- Hands-on activities should use free/open-source tools wherever possible and cloud-provider educational/free-tier facilities where available.
- Curricular and co-curricular activities are included to support application-oriented learning.

CCM-1: CLOUD COMPUTING FUNDAMENTALS

Semester: III | Pattern: 3 Theory + 2 Practical | Credits: 4 | Theory: 45 Hours | Unit Allocation: 9 Hours per Unit

I. Course Objectives

- To introduce the concept, evolution, characteristics and applications of cloud computing.
- To explain cloud deployment and service models and their practical use cases.
- To introduce SaaS, PaaS and IaaS and their responsibilities and architectures.
- To develop understanding of cloud storage, compute services and virtualization.
- To provide practical familiarity with basic cloud services and virtual machines.

II. Course Outcomes

- **CO1:** Explain the concept, evolution and essential characteristics of cloud computing.
- **CO2:** Differentiate public, private, hybrid and community cloud deployment models.
- **CO3:** Explain IaaS, PaaS and SaaS and select an appropriate model for an application.
- **CO4:** Describe cloud compute, storage and virtualization concepts.
- **CO5:** Demonstrate basic cloud-service usage and virtual-machine operations.

III. Unit-wise Theory Syllabus, Unit Outcomes, Curricular and Co- Curricular Activities:

UNIT-I: Introduction to Cloud Computing

Syllabus: Cloud Computing – Definition, Evolution of Cloud Computing, Need for Cloud Computing, Importance of Cloud Computing, On-Demand Self-Service, Resource Pooling, Rapid Elasticity, Measured Service, Broad Network Access, Advantages of Cloud Computing, Limitations of Cloud Computing, Cloud Applications, Cloud Users and Stakeholders, Cloud Computing versus Traditional Computing.

Unit Outcome: Explain the concept and evolution of cloud computing, describe essential characteristics, identify advantages and limitations, and compare cloud with traditional computing.

Co-Curricular: Group seminar on cloud evolution; cloud-awareness poster activity; case-study discussion on migration to cloud.

Curricular: Prepare a cloud-computing concept map; compare traditional and cloud computing; classify real-world cloud applications.

UNIT-II: Cloud Models and Service Models

Syllabus: NIST Cloud Computing Model, Essential Characteristics, Public Cloud, Private Cloud, Hybrid Cloud, Community Cloud, Cloud Service Models, Software as a Service (SaaS), Platform as a Service (PaaS), Infrastructure as a Service (IaaS), Cloud Architecture Layers, Major Cloud Service Providers, Cloud Service Selection, Cloud Service Integration and Comparison of Cloud Service Models.

Unit Outcome: Explain the NIST model, differentiate deployment models, explain IaaS/PaaS/SaaS, and select an appropriate cloud model for a given requirement.



Co-Curricular: Cloud service-model quiz; group presentation by deployment model; case-study competition on selecting a cloud model.

Curricular: Map real services to IaaS/PaaS/SaaS; compare public/private/hybrid/community clouds; prepare a service-selection matrix.

UNIT–III: Platform as a Service

Syllabus: Platform as a Service – Concept, Evolution of PaaS, PaaS Architecture, PaaS Components, Application Development on PaaS, Application Deployment, Elastic Beanstalk, Google App Engine, Other PaaS Examples, Application Frameworks, Developer Responsibilities, Operator Responsibilities, Advantages of PaaS, Challenges of PaaS and SaaS versus PaaS.

Unit Outcome: Explain PaaS architecture and components, describe application development and deployment on PaaS, identify responsibilities, and compare SaaS and PaaS.

Co-Curricular: PaaS demonstration; student seminar on application platforms; group discussion on vendor lock-in and portability.

Curricular: Design a simple PaaS deployment workflow; compare PaaS providers/features; identify developer and operator responsibilities.

UNIT–IV: Infrastructure as a Service and Cloud Storage

Syllabus: Infrastructure as a Service – Concept, Evolution of IaaS, IaaS Architecture, IaaS Service Components, Cloud Compute Services, Storage Area Network (SAN), Network Storage, Data as a Service (DaaS), Database as a Service (DBaaS), Cloud Storage, Object Storage, Benefits of Cloud Storage, Limitations of Cloud Storage, Cloud Storage Providers and Storage Service Selection.

Unit Outcome: Explain IaaS architecture, describe compute and storage services, differentiate storage approaches, and select suitable cloud storage services.

Co-Curricular: Cloud-storage hands-on showcase; storage-service comparison presentation; case-study activity on choosing DBaaS/storage.

Curricular: Compare compute and storage services; design a basic IaaS architecture; classify object/network storage use cases.

UNIT–V: Virtualization

Syllabus: Virtualization – Concept, Evolution of Virtualization, Virtual Machines, Hypervisors, Virtualization Components, Server Virtualization, Desktop Virtualization, Storage Virtualization, Network Virtualization, Application Virtualization, Benefits of Virtualization, Resource Isolation, Virtualization and Cloud Computing, Virtual Machine Management and Virtualization Security.

Unit Outcome: Explain virtualization and virtual machines, describe types of virtualization, explain hypervisors, and analyze the relationship between virtualization and cloud computing.

Co-Curricular: Virtual-machine demonstration; virtualization poster/model activity; group discussion on virtualization security.

Curricular: Draw a hypervisor/VM architecture; compare server, storage and network virtualization; analyze resource-isolation scenarios.

IV. Text Books

1. Rajkumar Buyya, Christian Vecchiola and S. Thamarai Selvi, Mastering Cloud Computing, McGraw-Hill.
2. Anthony T. Velte, Toby J. Velte and Robert Elsenpeter, Cloud Computing: A Practical Approach, McGraw-Hill.

V. Reference Books / Online References

1. Dan C. Marinescu, Cloud Computing: Theory and Practice, Elsevier.
2. Michael Miller, Cloud Computing: Web-Based Applications That Change the Way You Work and Collaborate Online, Que.
3. Arshdeep Bahga and Vijay Madisetti, Cloud Computing: A Hands-on Approach, Universities Press.
4. NIST cloud computing publications and cloud-provider learning/documentation resources.

A handwritten signature or mark in blue ink, possibly reading '14000', with a horizontal line underneath.

Cloud Computing Fundamentals Practical / Laboratory Experiments

1. Create/access an educational cloud account and explore the cloud service dashboard.
2. Create folders/containers in a cloud storage service.
3. Upload, download, organize and share files using cloud storage.
4. Create and collaborate on a cloud-based document.
5. Create and collaborate on a cloud-based presentation.
6. Create and use a cloud-based spreadsheet.
7. Compare two cloud-storage services using a feature matrix.
8. Analyze public, private and hybrid cloud cases for given scenarios.
9. Create and configure a virtual machine using an educational/free-tier environment.
10. Install or access a Linux virtual machine and perform basic commands.
11. Configure basic virtual-machine networking.
12. Demonstrate basic cloud identity and access permissions.
13. Compare SaaS, PaaS and IaaS using real service examples.
14. Deploy a simple static web page using a cloud platform.
15. Demonstrate cloud backup and restoration of selected files.
16. Draw a cloud architecture for a small application.
17. Prepare a cloud service-selection report for a given institutional requirement.
18. Mini project: propose and demonstrate a simple cloud-based solution.

A handwritten signature in blue ink, consisting of a large loop followed by the letters 'H' and 'Q' and a flourish, underlined.

CCM-2: APEX AND VISUALFORCE PROGRAMMING

Semester: IV | Pattern: 3 Theory + 2 Practical | Credits: 4 | Theory: 45 Hours | Unit Allocation: 9 Hours per Unit

I. COURSE OBJECTIVES: After completing this course, the student will be able to:

1. Understand the fundamentals of Apex programming, including data types, operators, control structures, methods, classes and objects.
2. Understand object-oriented programming concepts and their implementation in Apex.
3. Understand Salesforce objects, Apex classes, data access, testing and debugging.
4. Develop Apex applications using collections, DML operations, exception handling and efficient programming practices.
5. Develop Apex applications using triggers, interfaces and asynchronous processing.
6. Develop basic cloud-based user interfaces using Visualforce components and controllers.

II. COURSE OUTCOMES: After successful completion of this course, the student will be able to:

CO1: Apply Apex syntax, data types, operators, control structures, methods and object-oriented programming concepts.

CO2: Develop Apex classes and work with standard/custom sObjects, queries and test classes.

CO3: Use Apex collections and DML/database operations for processing Salesforce records.

CO4: Develop Apex applications using triggers, interfaces and asynchronous processing.

CO5: Develop Visualforce applications using components, data binding and controllers.

III. Unit-wise Theory Syllabus, Unit Outcomes, Curricular and Co- Curricular Activities:

UNIT-I: APEX FUNDAMENTALS AND OBJECT-ORIENTED PROGRAMMING: Introduction to Apex, Features of Apex, Apex Development Environment, Apex Syntax and Structure, Identifiers and Variables, Primitive Data Types, sObjects and Complex Data Types, Expressions and Operators, Conditional Statements, Iterative Statements, Methods and Parameters, Classes and Objects, Constructors, Static and Instance Members, Encapsulation, Inheritance, Polymorphism and Abstraction.

Unit-wise Learning Outcomes: After completing this unit, the student will be able to:

1. Apply Apex syntax, data types, operators, control structures and methods.
2. Explain classes, objects and core object-oriented concepts in Apex.
3. Develop simple Apex classes using object-oriented programming principles.



Curricular Activities

- Demonstration of Apex syntax and programming constructs.
- Programming exercises on data types and operators.
- Exercises on conditional and iterative statements.
- Development of methods and simple classes.
- Problem-solving exercises based on OOP concepts.

Co-Curricular Activities

- Individual programming assignment.
- Student seminar on object-oriented programming in Apex.
- Quiz on Apex fundamentals.
- Peer programming activity.
- Online coding/problem-solving exercises.

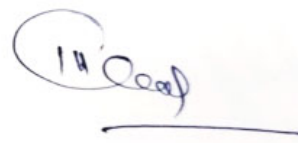
UNIT-II: APEX CLASSES, SUBJECTS AND TESTING: Apex Classes, Class Members, Methods and Constructors, Access Modifiers, Static and Instance Methods, sObjects, Standard and Custom Objects, Creating Custom Objects, Accessing sObject Data, Basic SOQL and SOSL, Anonymous Apex, Apex Development Process, Test Classes, Test Methods, Test Data, Assertions, Code Coverage and Debugging Apex.

Unit-wise Learning Outcomes: After completing this unit, the student will be able to:

1. Develop Apex classes and work with standard and custom sObjects.
2. Retrieve Salesforce data using basic SOQL and SOSL.
3. Create test classes, use assertions and understand code coverage and debugging.

Curricular Activities

- Creation and execution of Apex classes.
- Exercises on methods, constructors and access modifiers.
- Creation and use of custom objects.
- Basic SOQL/SOSL exercises.
- Development and execution of Apex test classes.



Co-Curricular Activities

- Assignment on Apex classes and sObjects.
- Seminar on Salesforce cloud development.
- Quiz on SOQL/SOSL and Apex classes.
- Group discussion on Apex testing.
- Small study project using Apex and Salesforce objects.

UNIT–III: COLLECTIONS, DML AND EXCEPTION HANDLING: Apex Collections, List, Set and Map, Collection Methods, Iteration and Processing Records, sObjects and Collections, Data Manipulation Language (DML), Insert, Update, Upsert, Delete, Undelete and Merge, Database Methods, Bulk Processing, Governor Limits and Efficient Apex, Exception Handling, Try-Catch-Finally, DML Exceptions, Custom Exceptions, Assertions and Debugging.

Unit-wise Learning Outcomes: After completing this unit, the student will be able to:

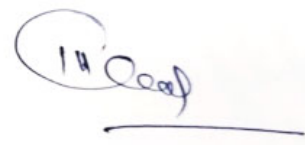
1. Apply List, Set and Map collections for processing Salesforce records.
2. Perform DML and database operations using Apex.
3. Develop efficient Apex programs using exception handling and awareness of governor limits.

Curricular Activities

- Programming exercises using List, Set and Map.
- DML operation exercises.
- Multiple-record processing activities.
- Database-method demonstrations.
- Exception-handling and debugging exercises.
- Exercises on bulk processing and governor-limit awareness.

Co-Curricular Activities

- Individual assignment on Apex collections.
- Quiz on DML and database operations.
- Coding challenge on collections.
- Group discussion on governor limits and efficient Apex.
- Seminar on exception handling.



UNIT-IV: TRIGGERS AND ASYNCHRONOUS APEX: Apex Triggers, Trigger Syntax, Trigger Events, Before and After Triggers, Trigger Context Variables, Trigger.new, Trigger.old, Trigger.newMap and Trigger.oldMap, Trigger Design and Bulkification, Trigger-Based Validation and Automation, Interfaces in Apex, Asynchronous Apex, Queueable Apex, Batch Apex, Scheduled Apex, Apex Jobs and Monitoring Asynchronous Execution.

Unit-wise Learning Outcomes: After completing this unit, the student will be able to:

1. Develop Apex triggers using trigger events and context variables.
2. Apply bulkification, validation and automation principles in triggers.
3. Explain and implement basic asynchronous processing using Queueable, Batch and Scheduled Apex.

Curricular Activities

- Development of basic Apex triggers.
- Exercises using trigger context variables.
- Development of before and after triggers.
- Interface implementation exercises.
- Demonstration of Queueable, Batch and Scheduled Apex.
- Case study on Salesforce automation.

Co-Curricular Activities

- Assignment on Apex triggers.
- Seminar on asynchronous Apex.
- Quiz on trigger context variables.
- Group discussion on Salesforce automation.
- Mini project using triggers.

UNIT-V: VISUALFORCE PAGES, COMPONENTS AND CONTROLLERS: Introduction to Visualforce, Visualforce Page Structure, Visualforce Markup, Visualforce Components, Component Attributes, Output Components, Input Components, Action Components, Formatting Components, Forms and Data Binding, Expression Language, Standard Controllers, Custom Controllers, Controller Extensions, Apex Controllers, Page Navigation and Basic Visualforce Application Development.

Unit-wise Learning Outcomes: After completing this unit, the student will be able to:

1. Develop Visualforce pages using components, expressions and data binding.

2. Use input, output, action and formatting components.
3. Develop Visualforce applications using standard controllers, custom controllers and controller extensions.

Curricular Activities

- Creation of basic Visualforce pages.
- Exercises using Visualforce components.
- Development of input and output forms.
- Exercises using action and formatting components.
- Development using standard controllers.
- Development using custom controllers and controller extensions.

Co-Curricular Activities

- Assignment on Visualforce components.
- Student seminar on Visualforce.
- Quiz on Visualforce tags and controllers.
- Peer review of Visualforce pages.
- Mini project integrating Apex and Visualforce.

IV. TEXTBOOKS

1. **Apex Complete Developer Guide**, Salesforce.com.
2. **Visualforce Developer Guide**, Salesforce.com.

V. REFERENCE BOOKS / ONLINE RESOURCES

1. **Apex Reference Guide**, Salesforce.com.
2. **Salesforce Developer Documentation**, Salesforce.com.
3. **Apex Programming Basics**, Salesforce Trailhead.
4. **Visualforce Fundamentals**, Salesforce Trailhead.
5. **Salesforce Extensions for Visual Studio Code – Apex Development**, Salesforce Developers.

A handwritten signature in blue ink, possibly reading '14 Qad', is written above a horizontal line.

PRACTICAL / LABORATORY ACTIVITIES FOR APEX AND VISUALFORCE PROGRAMMING

The following practical exercises are recommended:

1. Write Apex programs using variables, data types and operators.
2. Implement conditional and iterative statements in Apex.
3. Develop Apex methods and functions.
4. Implement classes, objects and OOP concepts.
5. Create and execute an Apex class.
6. Create a Salesforce custom object and access it using Apex.
7. Write basic SOQL and SOSL queries.
8. Create and execute an Apex Test class.
9. Implement List, Set and Map collections.
10. Perform Insert, Update and Upsert operations.
11. Perform Delete, Undelete and Merge operations.
12. Implement exception handling and assertions.
13. Develop an Apex interface.
14. Create Apex triggers using trigger context variables.
15. Implement validation and automation using triggers.
16. Develop a Queueable Apex program.
17. Develop a Batch Apex program.
18. Develop and schedule a Scheduled Apex program.
19. Create Visualforce pages using input, output and action components.
20. Develop Visualforce applications using standard and custom controllers.
21. Develop a Visualforce application using a controller extension.
22. Mini Project: Develop a simple Salesforce cloud application integrating custom objects, Apex, triggers and Visualforce.



CCM-3: CLOUD INFRASTRUCTURE, VIRTUALIZATION AND AWS

Semester: V | Pattern: 3 Theory + 2 Practical | Credits: 4 | Theory: 45 Hours | Unit Allocation: 9 Hours per Unit

Course Objectives

- To develop an understanding of cloud infrastructure, distributed computing and virtualization.
- To explain cloud architecture, availability, scalability, elasticity and fault tolerance.
- To introduce disaster recovery, backup, migration and business continuity concepts.
- To introduce AWS global infrastructure, VPC, EC2 and related services.
- To develop practical skills in basic AWS networking, compute, storage and IAM.

Course Outcomes

- CO1: Explain cloud infrastructure and distributed/parallel computing foundations.
- CO2: Describe virtualization technologies and cloud architecture models.
- CO3: Analyze availability, scalability, disaster recovery and fault-tolerance requirements.
- CO4: Explain AWS Regions, Availability Zones, VPC and EC2 concepts.
- CO5: Configure basic AWS compute, networking, storage and IAM resources in an educational environment.

III. Unit-wise Theory Syllabus, Unit Outcomes, Curricular and Co- Curricular Activities:

UNIT-I: Cloud Infrastructure Foundations

Syllabus: Cloud Computing Foundations, Evolution of Cloud Infrastructure, Parallel Computing, Distributed Computing, Elasticity, On-Demand Resource Provisioning, Scalability, Availability, Fault Tolerance, Cloud Application Characteristics, Service-Oriented Architecture (SOA), Representational State Transfer (REST) Architecture, Web Services, Publish-Subscribe Model and Cloud Infrastructure Requirements.

Unit Outcome: Explain cloud infrastructure foundations, distributed and parallel computing, scalability, elasticity and availability, and architectural requirements of cloud applications.

Co-Curricular: Cloud-architecture seminar; infrastructure case-study discussion; group poster on elasticity and availability.

Curricular: Compare parallel and distributed computing; draw a basic cloud-infrastructure architecture; analyze scalability and fault-tolerance scenarios.

UNIT-II: Virtualization and Cloud Architecture

Syllabus: Virtualization Fundamentals, Levels of Virtualization, Server Virtualization, CPU Virtualization, Memory Virtualization, I/O Virtualization, Hypervisors, Virtual Machine Management, Virtual Machine Support, Cloud Architecture Layers, Public, Private and Hybrid Clouds, IaaS, PaaS and SaaS, Cloud Delivery Models, Cloud Deployment Models and Virtualization in Cloud Infrastructure.



Unit Outcome: Explain levels of virtualization, hypervisors and VMs, relate virtualization to cloud architecture, and differentiate delivery and deployment models.

Co-Curricular: Hypervisor/VM demonstration; virtualization technology seminar; group activity comparing cloud deployment models.

Curricular: Draw different virtualization layers; compare hypervisor types; map virtualization components to cloud services.

UNIT–III: Cloud Design, Disaster Recovery and Availability

Syllabus: Disaster Recovery, Cloud Backup, Recovery Strategies, Recovery Objectives, Availability, Reliability, Fault Tolerance, Cloud Architecture Principles, Resource Pooling, Elastic Resource Allocation, Load Distribution, Cloud Service Selection, Application Migration, Business Continuity and Cloud Infrastructure Risks.

Unit Outcome: Explain disaster recovery and backup, describe availability and fault tolerance, analyze cloud architecture principles, and identify suitable continuity strategies.

Co-Curricular: Disaster-recovery tabletop exercise; group presentation on cloud outages; case-study competition on recovery strategies.

Curricular: Prepare a backup and disaster-recovery plan; identify availability and fault-tolerance requirements; design a basic continuity workflow.

UNIT–IV: AWS Cloud and Networking

Syllabus: Introduction to Amazon Web Services (AWS), AWS Global Infrastructure, AWS Regions, Availability Zones, AWS Management Console, Cloud Service Providers, Amazon Virtual Private Cloud (VPC), VPC Components, Subnets, IP Addressing, Route Tables, Security Groups, Amazon Elastic Compute Cloud (EC2), EC2 Instance Types and EC2 Pricing Concepts.

Unit Outcome: Explain AWS global infrastructure, Regions and Availability Zones, VPC architecture and networking components, and EC2 configuration concepts.

Co-Curricular: AWS console demonstration; VPC design challenge; group seminar on AWS global infrastructure.

Curricular: Draw AWS Region/AZ architecture; design a basic VPC with subnets and routes; compare EC2 instance choices for sample workloads.

UNIT–V: AWS Storage and Security

Syllabus: Cloud Storage Concepts, Object Storage, Amazon Simple Storage Service (S3), S3 Buckets, S3 Objects, S3 Features, Storage Classes, Lifecycle Management, Access Control, AWS Identity and Access Management (IAM), Users and Roles, Policies and Permissions, Cloud Governance, Virtual Machine Security and Network and Application Security.

Unit Outcome: Explain S3 object storage, buckets, objects and storage classes; explain IAM users, roles and policies; and apply basic AWS security and governance principles.

Co-Curricular: AWS security checklist activity; IAM policy review exercise; group presentation on secure cloud architecture.

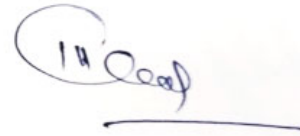
Curricular: Design an S3 storage hierarchy; compare S3 storage classes; create a basic IAM permission model for a sample workload.

IV. Text Books

1. Arshdeep Bahga and Vijay Madisetti, Cloud Computing: A Hands-on Approach, Universities Press.
2. Rajkumar Buyya, Christian Vecchiola and S. Thamarai Selvi, Mastering Cloud Computing, McGraw-Hill.

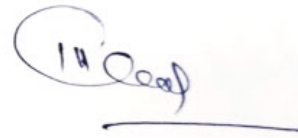
V. Reference Books / Online References

1. AWS Skill Builder learning resources.
2. AWS documentation for VPC, EC2, S3 and IAM.
3. Dan C. Marinescu, Cloud Computing: Theory and Practice, Elsevier.
4. Cloud architecture and security best-practice documentation.

A handwritten signature in blue ink, possibly reading '14/09/2020', is written above a horizontal line.

CLOUD INFRASTRUCTURE, VIRTUALIZATION AND AWS Practical / Laboratory Experiments

1. Explore AWS educational/free-tier environment and AWS Management Console.
2. Identify AWS Regions and Availability Zones for a selected workload.
3. Create and examine a basic VPC.
4. Create a subnet and inspect IP-address configuration.
5. Configure or inspect route tables and basic network routing.
6. Create and apply a security group for an educational EC2 workload.
7. Launch an EC2 instance using an appropriate instance type.
8. Start, stop, reboot and terminate an EC2 instance safely.
9. Connect to an EC2 Linux instance and perform basic administration.
10. Compare selected EC2 instance types for a sample workload.
11. Create an S3 bucket and configure basic properties.
12. Upload, download and organize objects in S3.
13. Demonstrate S3 access permissions and safe sharing.
14. Compare selected S3 storage classes and use cases.
15. Create basic IAM users/roles and examine permissions.
16. Apply a least-privilege policy in an educational environment.
17. Prepare a basic AWS security checklist for an EC2/S3 workload.
18. Design and document a small AWS architecture using VPC, EC2 and S3.

A handwritten signature in blue ink, possibly reading 'I. H. Q. Q. Q.', is written above a horizontal line.

CCM-4: BUSINESS INTELLIGENCE FOR CLOUD

Semester: V | Pattern: 3 Theory + 2 Practical | Credits: 4 | Theory: 45 Hours | Unit Allocation: 9 Hours per Unit

I. COURSE OBJECTIVES: After completing this course, the student will be able to:

1. Understand Business Intelligence concepts and cloud-based analytics using Salesforce Analytics.
2. Explore and analyze business data using cloud analytics applications.
3. Create and configure analytics applications, dashboards and reports.
4. Develop Sales and Service Analytics applications for business decision-making.
5. Apply dashboard customization and visualization techniques for cloud-based business intelligence.

II. COURSE OUTCOMES: After successful completion of this course, the student will be able to:

CO1: Explain Business Intelligence concepts and Salesforce Analytics fundamentals.

CO2: Explore and analyze business data using cloud analytics tools.

CO3: Create analytics applications and dashboards with appropriate permissions.

CO4: Develop Sales Analytics solutions for business reporting.

CO5: Develop Service Analytics dashboards for business decision support.

III. Unit-wise Theory Syllabus, Unit Outcomes, Curricular and Co- Curricular Activities:

UNIT-I: SALESFORCE ANALYTICS FUNDAMENTALS: Introduction to Business Intelligence, Introduction to Salesforce Analytics, Wave Analytics Overview, CRM Analytics Concepts, Analytics Environment Setup, Exploring Salesforce Analytics, Analytics Apps, Creating Analytics Apps, Analytics Navigation, Business Intelligence Applications.

Unit-wise Learning Outcomes: After completing this unit, the student will be able to:

1. Explain Business Intelligence and Salesforce Analytics concepts.
2. Configure the basic analytics environment.
3. Create and explore analytics applications.

Curricular Activities

- Demonstration of Salesforce Analytics environment.
- Analytics App creation exercise.



- Navigation and exploration activity.
- BI use-case discussion.
- Classroom problem-solving exercise.

Co-Curricular Activities

- Assignment on Salesforce Analytics.
- Seminar on Business Intelligence applications.
- Quiz on BI fundamentals.
- Group discussion.
- Online Salesforce Trailhead practice.

UNIT-II: DATA EXPLORATION AND ANALYSIS: Desktop Data Explorer, Exploring Datasets, Selecting Dimensions and Measures, Filtering Data, Grouping Data, Sorting Data, Compare Table, Data Visualization, Desktop Exploration Interface, Mobile Data Explorer, Mobile Analytics Interface.

Unit-wise Learning Outcomes: After completing this unit, the student will be able to:

1. Explore datasets using Salesforce Analytics tools.
2. Analyze business data using tables and visualizations.
3. Perform desktop and mobile analytics exploration.

Curricular Activities

- Desktop Data Explorer demonstration.
- Compare Table exercise.
- Filtering and grouping activity.
- Mobile analytics demonstration.
- Dataset exploration exercise.

Co-Curricular Activities

- Assignment on Data Explorer.
- Seminar on mobile analytics.
- Quiz on data exploration.



- Peer learning activity.
- Mini study project.

UNIT–III: ANALYTICS APPLICATIONS AND DASHBOARDS: Wave App Basics, Creating Analytics Apps, Analytics Configuration, Datasets, App Organization, Licenses and Permissions, User Access, Dashboard Components, Charts, Tables, Dashboard Filters and Interactions.

Unit-wise Learning Outcomes: After completing this unit, the student will be able to:

1. Create and configure analytics applications.
2. Apply user permissions and access controls.
3. Develop interactive dashboards.

Curricular Activities

- Analytics App creation exercise.
- Permission configuration activity.
- Dashboard component demonstration.
- Interactive dashboard exercise.
- User-access case study.

Co-Curricular Activities

- Assignment on analytics applications.
- Seminar on dashboard development.
- Quiz on permissions and dashboards.
- Group discussion.
- Dashboard design activity.

UNIT–IV: SALES ANALYTICS: Introduction to Sales Analytics, Sales Wave Overview, Sales Analytics Wizard, Creating Sales Analytics Applications, Sales Datasets, Sales Metrics, Sales KPIs, Sales Dashboards, Sales Reports, Mobile Sales Analytics, Sales Trend Analysis, Business Decision Support using Sales Analytics.

Unit-wise Learning Outcomes: After completing this unit, the student will be able to:

1. Create Sales Analytics applications using Salesforce tools.



2. Analyze sales performance using dashboards and KPIs.
3. Interpret sales reports for business decision-making.

Curricular Activities

- Sales Analytics Wizard demonstration.
- Sales dashboard creation.
- KPI identification exercise.
- Mobile sales analytics activity.
- Sales trend analysis exercise.

Co-Curricular Activities

- Assignment on Sales Analytics.
- Seminar on sales dashboards.
- Quiz on sales KPIs.
- Group activity on business reporting.
- Case study on sales performance.

UNIT-V: SERVICE ANALYTICS AND DASHBOARD CUSTOMIZATION: Introduction to Service Analytics, Service Wave Overview, Creating Service Analytics Applications, Service Datasets, Service Workload Analysis, Service KPIs, Service Dashboards, Dashboard Customization, Dashboard Components, Mobile Service Analytics, Service Performance Analysis, Business Decision Support using Service Analytics.

Unit-wise Learning Outcomes: After completing this unit, the student will be able to:

1. Develop Service Analytics applications.
2. Customize dashboards for service management.
3. Apply Service Analytics for business decision support.

Curricular Activities

- Service Analytics Wizard demonstration.
- Dashboard customization exercise.



- Service workload analysis.
- KPI development exercise.
- Dashboard presentation activity.

Co-Curricular Activities

- Assignment on Service Analytics.
- Student seminar on service dashboards.
- Quiz on dashboard customization.
- Peer review activity.
- Mini project on Service Analytics.

IV. TEXTBOOKS

1. **Business Intelligence Guidebook: From Data Integration to Analytics**, Rick Sherman, Morgan Kaufmann.
2. **Business Intelligence: A Managerial Perspective on Analytics**, Ramesh Sharda, Dursun Delen and Efraim Turban, Pearson.
3. **The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling**, Ralph Kimball and Margy Ross, Wiley.

V. REFERENCE BOOKS / ONLINE RESOURCES

1. **Cloud Computing: Concepts, Technology & Architecture**, Thomas Erl, Ricardo Puttini and Zaigham Mahmood, Prentice Hall.
2. **Building the Data Warehouse**, W. H. Inmon, Wiley.
3. **Fundamentals of Business Analytics**, Ramesh Sharda, Dursun Delen and Efraim Turban, Pearson.
4. **Designing Data-Intensive Applications**, Martin Kleppmann, O'Reilly Media.
5. Documentation and learning resources of leading cloud BI platforms.
6. Documentation and learning resources of cloud data warehouse platforms.
7. Documentation and learning resources of cloud-based data visualization and dashboard platforms.



PRACTICAL / LABORATORY ACTIVITIES for CCM-6: BUSINESS INTELLIGENCE FOR CLOUD

1. Explore the Salesforce Analytics environment.
2. Configure the analytics setup.
3. Create a basic Analytics App.
4. Navigate the Data Explorer.
5. Filter and group datasets.
6. Compare business data using Compare Table.
7. Perform desktop data exploration.
8. Perform mobile data exploration.
9. Create an Analytics App with user permissions.
10. Configure datasets for dashboards.
11. Create dashboard components.
12. Design an interactive dashboard.
13. Create a Sales Analytics application using Wizard.
14. Analyze Sales KPIs.
15. Create a Sales Dashboard.
16. Create a Service Analytics application.
17. Analyze Service workload.
18. Customize a Service Dashboard.
19. Create mobile analytics views.
20. Mini Project: Develop a cloud-based BI dashboard integrating Sales and Service Analytics.



CCM-5: CLOUD SECURITY AND DEVOPS

Semester: VI | Pattern: 3 Theory + 2 Practical | Credits: 4 | Theory: 45 Hours | Unit Allocation: 9 Hours per Unit

Course Objectives

- To introduce the principles, threats and controls associated with cloud security.
- To explain data security, network security and identity/access management.
- To introduce DevOps, Git, CI/CD, containers and DevSecOps practices.
- To develop understanding of monitoring, incident response, backup and disaster recovery.
- To provide practical experience in secure configuration, IAM, version control, containers and CI/CD.

Course Outcomes

- CO1: Explain fundamental cloud-security objectives, threats and shared-responsibility concepts.
- CO2: Apply basic data, network and identity-access security principles.
- CO3: Use basic Git/version-control and CI/CD concepts.
- CO4: Explain containers, Docker and DevSecOps practices.
- CO5: Develop basic monitoring, incident-response, backup and continuity plans.

III. Unit-wise Theory Syllabus, Unit Outcomes, Curricular and Co- Curricular Activities:

UNIT-I: Fundamentals of Cloud Security

Syllabus: Cloud Security – Meaning, Security Objectives, Confidentiality, Integrity, Availability, Cloud Governance, Shared Responsibility Model, Cloud Security Reference Models, Security Policies, Security Controls, Cloud Threats, Cloud Vulnerabilities, Misconfiguration Risks, Account Compromise and Cloud Security Best Practices.

Unit Outcome: Explain cloud-security principles and CIA objectives, describe shared responsibility, and identify common threats and vulnerabilities.

Co-Curricular: Cloud-security awareness seminar; threat-modeling group exercise; security poster/quiz activity.

Curricular: Build a CIA/security-controls matrix; identify threats and vulnerabilities from case studies; analyze shared-responsibility scenarios.

UNIT-II: Data and Network Security

Syllabus: Cloud Data Security, Data at Rest, Data in Transit, Encryption, Encryption Keys, Key Management, Network Security, Firewalls, Security Groups, Network Segmentation, Host Security, Virtual Machine Security, Application Security, Secure Configuration and Security Testing.

Unit Outcome: Explain data protection, encryption and key management, describe network-security controls, and apply basic secure-configuration principles.

Co-Curricular: Security-control demonstration; firewall/security-group case study; group discussion on encryption and key management.

Curricular: Classify data-at-rest/in-transit protections; compare encryption and network controls; prepare a secure-configuration checklist.

UNIT–III: Identity and Access Management

Syllabus: Identity Management, Authentication, Authorization, Access Control, Role-Based Access Control (RBAC), Least Privilege, Users, Roles, Policies, Multi-Factor Authentication (MFA), Credentials, Keys and Secrets, Audit Logs, Access Review and IAM Best Practices.

Unit Outcome: Explain authentication and authorization, apply least privilege, describe users/roles/policies, and explain MFA and auditing.

Co-Curricular: IAM policy review activity; MFA awareness session; role/permission scenario-based group exercise.

Curricular: Design a basic RBAC model; apply least-privilege reasoning; analyze sample IAM policies and audit logs.

UNIT–IV: DevOps and DevSecOps

Syllabus: DevOps – Concept, DevOps Culture, Development and Operations Collaboration, Version Control, Git, Repository Management, Branching, Merging, Continuous Integration, Continuous Delivery, Continuous Deployment, Automated Testing, Containers, Docker and DevSecOps.

Unit Outcome: Explain DevOps principles, use basic version-control concepts, describe CI/CD pipelines, and explain containers, Docker and DevSecOps.

Co-Curricular: Git collaboration exercise; CI/CD demonstration day; DevSecOps group seminar and workflow challenge.

Curricular: Create a basic Git workflow; draw a CI/CD pipeline; demonstrate container build/run concepts and identify security checkpoints.

UNIT–V: Monitoring, Incident Response and Continuity

Syllabus: Cloud Monitoring, Metrics, Logs, Alerts, Performance Monitoring, Incident Response, Security Incident Handling, Backup and Restore, Disaster Recovery, Business Continuity, High Availability, Privacy, Compliance, Risk Assessment and Security Review.



Unit Outcome: Explain monitoring and logging, describe incident response, explain backup/disaster recovery/business continuity, and conduct a basic cloud-security review.

Co-Curricular: Incident-response tabletop exercise; monitoring dashboard demonstration; business-continuity case-study presentation.

Curricular: Identify useful metrics/logs/alerts; prepare an incident-response workflow; design backup and disaster-recovery procedures.

IV. Text Books

1. William Stallings, Network Security Essentials: Applications and Standards, Pearson.
2. Rajkumar Buyya, Christian Vecchiola and S. Thamarai Selvi, Mastering Cloud Computing, McGraw-Hill.

V. Reference Books / Online References

1. Pro Git, official Git documentation/book.
2. Docker documentation.
3. OWASP application-security resources.
4. AWS, Microsoft Azure and Google Cloud security documentation.
5. Cloud security best-practice and DevSecOps guidance.



CLOUD SECURITY AND DEVOPS Practical / Laboratory Experiments

1. Create and inspect basic cloud IAM identities and permissions.
2. Apply a least-privilege policy to an educational cloud resource.
3. Configure or demonstrate multi-factor authentication.
4. Examine and modify resource permissions safely.
5. Demonstrate encryption for stored or transmitted sample data.
6. Configure basic firewall/security-group rules in an educational environment.
7. Apply basic VM and host-security hardening steps.
8. Create a Git repository and perform commit operations.
9. Create a Git branch, make changes and merge the branch.
10. Clone and update a remote Git repository.
11. Build and run a basic Docker container.
12. Create or use a Docker image and inspect container configuration.
13. Create a simple CI workflow for build/test execution.
14. Run automated tests in a CI pipeline.
15. Inspect application/cloud logs and define useful alerts.
16. Prepare a basic backup and restore demonstration.
17. Develop a disaster-recovery and business-continuity plan for a sample workload.
18. Prepare a DevSecOps security checklist and conduct a mini security review.



CCM-6: CLOUD-NATIVE APPLICATIONS, STORAGE AND SERVERLESS COMPUTING

Semester: VI | Pattern: 3 Theory + 2 Practical | Credits: 4 | Theory: 45 Hours | Unit Allocation: 9 Hours per Unit

Course Objectives

- To introduce cloud-native application architecture and design principles.
- To explain cloud compute, object/block/file storage and managed databases.
- To introduce serverless computing, FaaS and event-driven application design.
- To explain APIs, messaging, load balancing, auto scaling and service integration.
- To develop understanding of reliability, cost optimization and cloud-native application deployment.

Course Outcomes

- CO1: Explain cloud-native application characteristics and architectural principles.
- CO2: Select appropriate compute, storage and database services.
- CO3: Design basic serverless and event-driven applications.
- CO4: Explain REST APIs, messaging, load balancing and auto scaling.
- CO5: Evaluate reliability, resource utilization and cost considerations in cloud architectures.

III. Unit-wise Theory Syllabus, Unit Outcomes, Curricular and Co- Curricular Activities:

UNIT-I: Cloud-Native Application Fundamentals

Syllabus: Cloud-Native Applications, Characteristics of Cloud Applications, Cloud-Native Architecture, Layered Architecture, Service-Oriented Architecture, Scalability, Elasticity, Availability, Reliability, Fault Tolerance, Stateless Applications, Stateful Applications, Monolithic Applications, Microservices and Cloud-Native Case Studies.

Unit Outcome: Explain cloud-native characteristics, differentiate stateful/stateless applications, compare monolithic and microservice architectures, and explain scalability, elasticity and fault tolerance.

Co-Curricular: Architecture poster activity; microservices case-study seminar; peer review of student architecture diagrams.

Curricular: Compare monolithic and microservice architectures; identify stateful/stateless components; design a basic cloud-native architecture.

UNIT-II: Cloud Compute, Storage and Databases

Syllabus: Virtual Machine-Based Compute, Container-Based Compute, Managed Compute Services, Object Storage, Block Storage, File Storage, Storage Classes, Storage Selection, Relational Databases, NoSQL Databases, Managed Database Services, Database Connectivity, Application-Database Integration, Storage Performance and Storage Service Selection.

Unit Outcome: Differentiate VM, container and managed compute services; compare object, block and file storage; explain relational and NoSQL databases; and select suitable services.

Co-Curricular: Storage-selection challenge; database technology seminar; cloud-service comparison presentation.

Curricular: Select compute/storage services for sample workloads; compare object/block/file storage; map relational and NoSQL database use cases.

UNIT–III: Serverless Computing

Syllabus: Serverless Computing – Concept, Function as a Service (FaaS), Serverless Architecture, Event-Driven Computing, Functions, Triggers, API Gateway, Managed Cloud Services, Function Execution, Stateless Functions, Serverless Storage, Advantages of Serverless Computing, Limitations of Serverless Computing, Serverless Use Cases and Serverless Application Design.

Unit Outcome: Explain serverless computing and FaaS, describe event-driven architecture, explain functions/triggers/API gateways, and identify suitable use cases.

Co-Curricular: Serverless demonstration; use-case ideation activity; group presentation on FaaS applications.

Curricular: Design an event-driven workflow; create a function/trigger concept diagram; identify suitable serverless use cases and limitations.

UNIT–IV: APIs, Messaging and Application Integration

Syllabus: REST APIs, API Endpoints, API Communication, Web Services, Messaging Services, Queues, Publish–Subscribe Systems, Load Balancing, Auto Scaling, Caching, Content Delivery Networks (CDNs), Application Monitoring, Performance Management, Service Integration and Fault Handling and Scaling.

Unit Outcome: Explain REST APIs and service integration, describe queues and publish-subscribe systems, explain load balancing and auto scaling, and design basic integrated cloud architectures.

Co-Curricular: API testing demonstration; messaging-system case study; integration-architecture group challenge.

Curricular: Design REST endpoints; model queue and publish-subscribe workflows; prepare a basic load-balancing/auto-scaling architecture.

UNIT–V: Cloud-Native Design, Reliability and Cost Optimization

Syllabus: Cloud-Native Design Principles, Twelve-Factor Application Principles, Configuration Management, Application Reliability, High Availability, Disaster Recovery, Backup Strategies, Cloud Service Selection, Resource Optimization, Right-Sizing, Cloud Cost Concepts, Cost Estimation, Usage Monitoring, Cost-Aware Architecture and Cloud Application Case Studies.

Unit Outcome: Apply cloud-native design principles, explain reliability and recovery, analyze resource utilization and right-sizing, and develop cost-aware cloud architecture recommendations.

Co-Curricular: Cloud-cost optimization case study; architecture peer review; final cloud-native mini-project exhibition.

Curricular: Apply Twelve-Factor principles to a sample application; prepare a right-sizing/cost-estimation exercise; design a reliability and backup plan.

IV. Text Books

1. Arshdeep Bahga and Vijay Madisetti, Cloud Computing: A Hands-on Approach, Universities Press.
2. Rajkumar Buyya, Christian Vecchiola and S. Thamarai Selvi, Mastering Cloud Computing, McGraw-Hill.

V. Reference Books / Online References

1. Dan C. Marinescu, Cloud Computing: Theory and Practice, Elsevier.
2. Sam Newman, Building Microservices, O'Reilly.
3. Twelve-Factor App methodology and related documentation.
4. AWS, Microsoft Azure and Google Cloud documentation for compute, storage, databases, APIs and serverless services.



CLOUD-NATIVE APPLICATIONS, STORAGE AND SERVERLESS COMPUTING Practical / Laboratory Experiments

1. Deploy a simple application using an educational/free-tier cloud environment.
2. Create and use an object-storage bucket for application data.
3. Compare object, block and file storage for a sample application.
4. Create/use a managed relational database or local equivalent for demonstration.
5. Create/use a NoSQL database for a sample application.
6. Connect a simple application to a cloud/managed database.
7. Create a basic serverless function.
8. Configure a trigger for a serverless function.
9. Expose a serverless function through an API endpoint.
10. Develop and test a simple REST API.
11. Demonstrate queue-based messaging.
12. Demonstrate publish-subscribe messaging.
13. Configure or simulate load balancing for a sample service.
14. Demonstrate auto scaling or an equivalent scaling configuration.
15. Use cloud/application monitoring to inspect metrics and logs.
16. Draw and explain a cloud-native application architecture.
17. Prepare a simple cloud cost estimate and resource-optimization recommendation.
18. Mini project: design and demonstrate a small cloud-native/serverless application.

