



SAVEETHA SCHOOL OF ENGINEERING
SAVEETHA INSTITUTE OF MEDICAL AND
TECHNICAL SCIENCES



COURSE CODE:

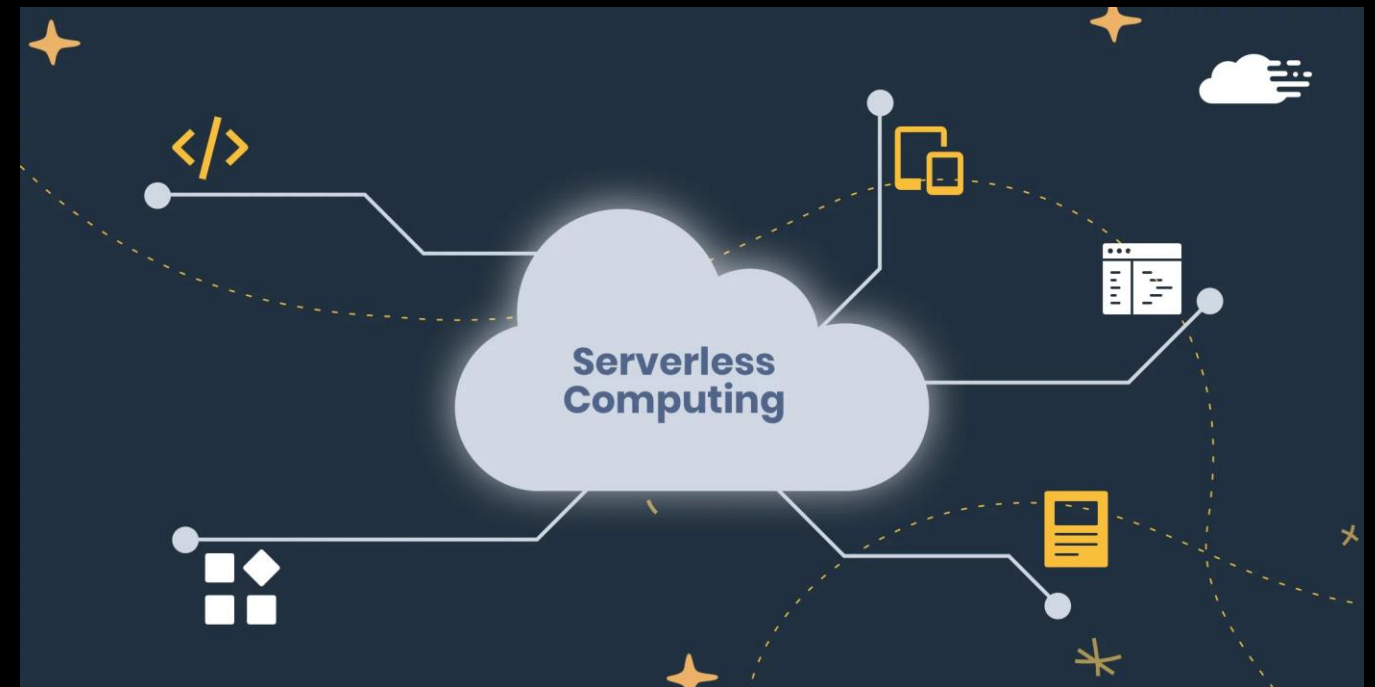
CSA1590 CLOUD COMPUTING FOR BIG DATA ANALYTICS FOR VIRTUAL CLUSTERS

TOPIC:

Project on Developing a Serverless Application using services such as AWS Lambda, Azure Functions, or Google Cloud Functions.

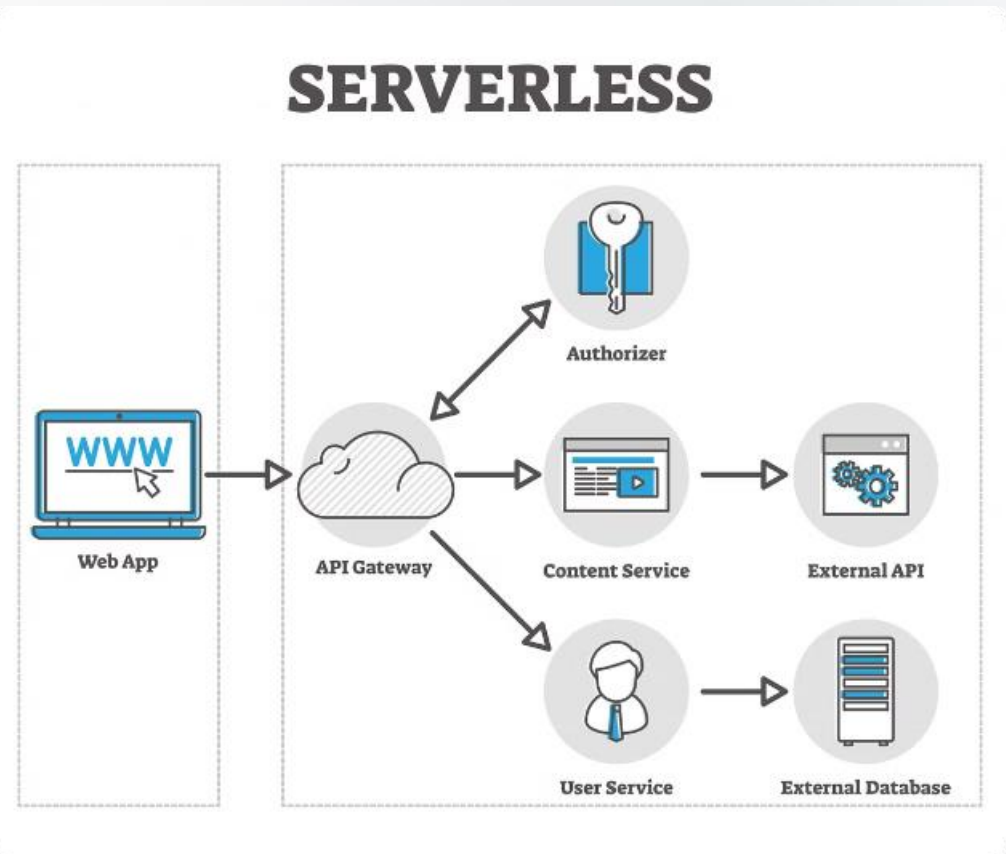
FACULTY NAME:Dr.Gnana Soundari

By,
NAME :D.DEEPAK
REG.NO:192210646



Introduction to Serverless Architecture

Serverless computing is a cloud-based execution model that automatically provisions and manages the infrastructure needed to run your code. It allows developers to focus on building applications without the burden of managing servers.



Benefits of Serverless Computing

Scalability

Serverless platforms automatically scale your applications up or down based on demand, eliminating the need to provision and manage resources.

Cost Optimization

You only pay for the resources you use, which can lead to significant cost savings compared to traditional server-based architectures.

Reduced Overhead

Serverless platforms handle server provisioning, scaling, patching, and other operational tasks, allowing developers to focus on writing code.

Overview of Serverless Platforms



AWS Lambda

AWS's serverless computing service, which allows you to run code without managing servers.



Azure Functions

Microsoft's serverless computing service, which provides event-driven, serverless computing.



Google Cloud Functions

Google's serverless computing service, which enables you to create and run event-driven applications.



Choosing the Right Serverless Platform

1 Maturity and Ecosystem

Consider the platform's market share, feature set, and integrations with other cloud services.

2 Pricing and Cost Model

Evaluate the pricing structure and how it aligns with your application's usage patterns.

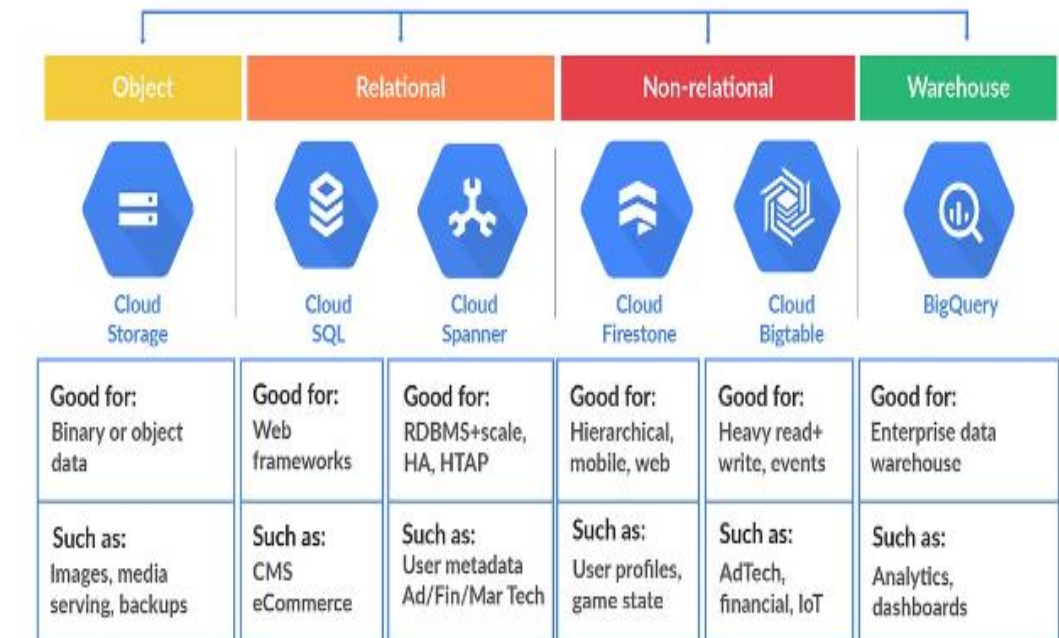
3 Language Support

Ensure the platform supports the programming languages and runtimes you prefer.

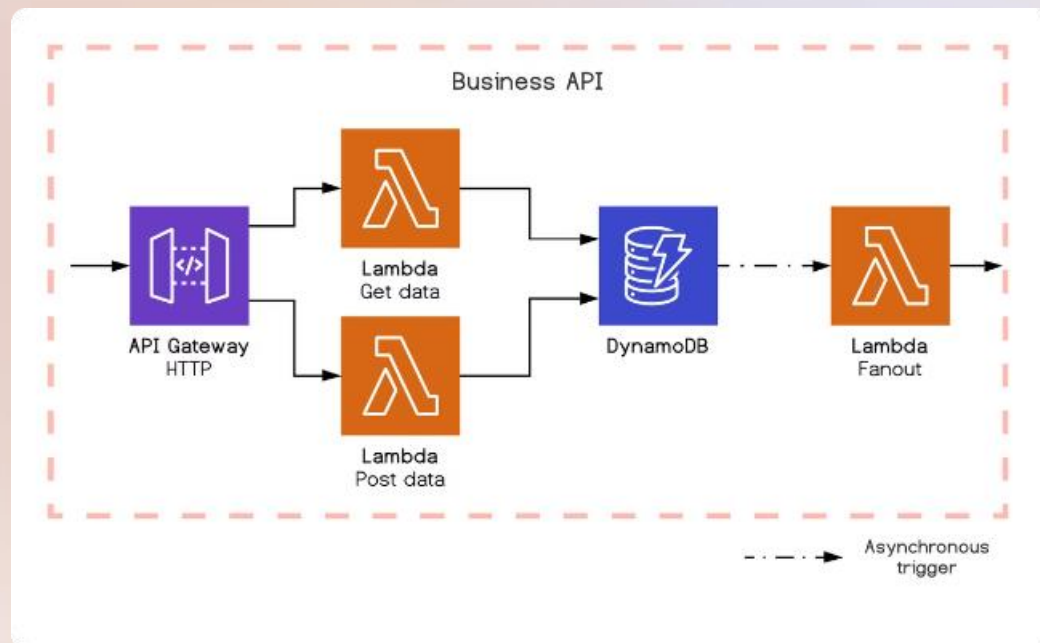
4 Developer Experience

Assess the platform's tooling, documentation, and community support to ensure a smooth development experience.

Storage & Database Services



Designing Event-Driven Serverless Applications



1

Trigger Event

An event, such as an API call or a database update, that initiates the serverless function.

2

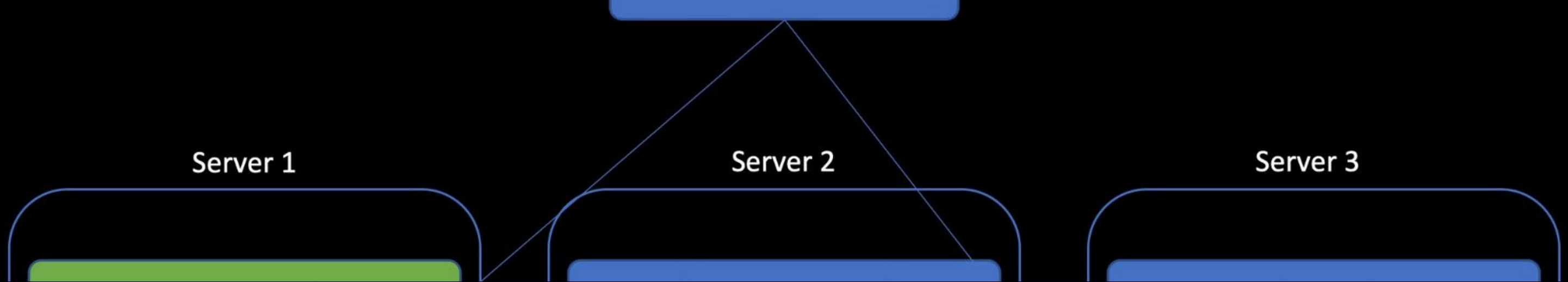
Function Execution

The serverless function is executed in response to the trigger event.

3

Integration

The serverless function may integrate with other cloud services to fulfill the application's requirements.



Implementing Serverless Functions

Function Code

Write modular, stateless functions that perform a specific task, such as processing data or responding to API requests.

Configuration

Define the function's trigger, environment variables, and other settings through the platform's management console or CLI.

Testing

Thoroughly test your serverless functions locally and in the cloud to ensure they meet your application's requirements.

Deployment

Automate the deployment of your serverless functions using CI/CD pipelines for efficient and reliable releases.

Integrating Serverless Functions

1

Data Sources

Connect your serverless functions to databases, message queues, and other data stores to fetch and process information.

2

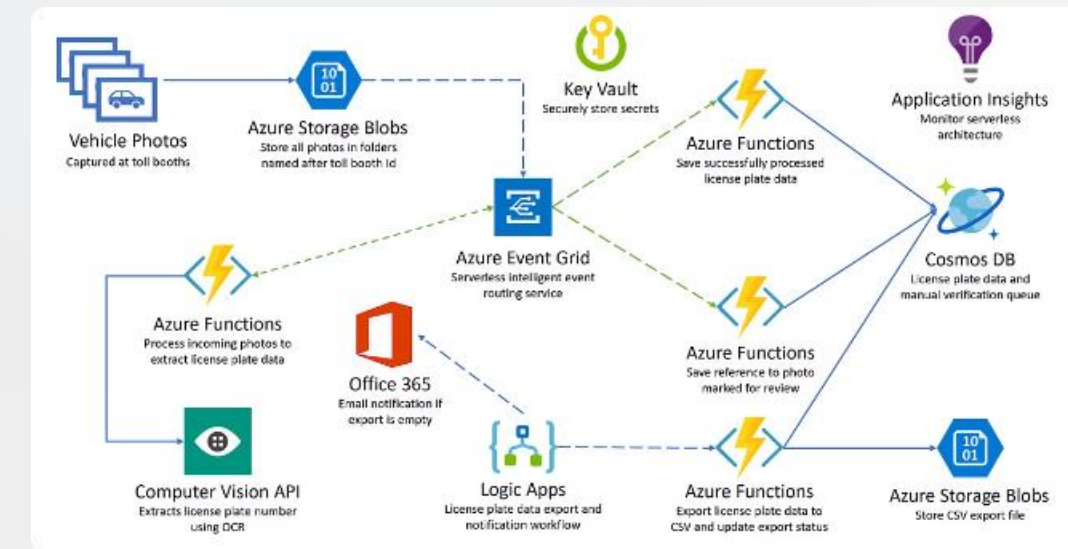
Web Services

Integrate your serverless functions with APIs, third-party SaaS platforms, and other web services to extend your application's capabilities.

3

Event Triggers

Configure your serverless functions to respond to various events, such as file uploads, IoT sensor data, or scheduled tasks.



Monitoring and Debugging Serverless Applications

Logging and Metrics

Use the platform's logging and monitoring services to track function invocations, errors, and performance metrics.

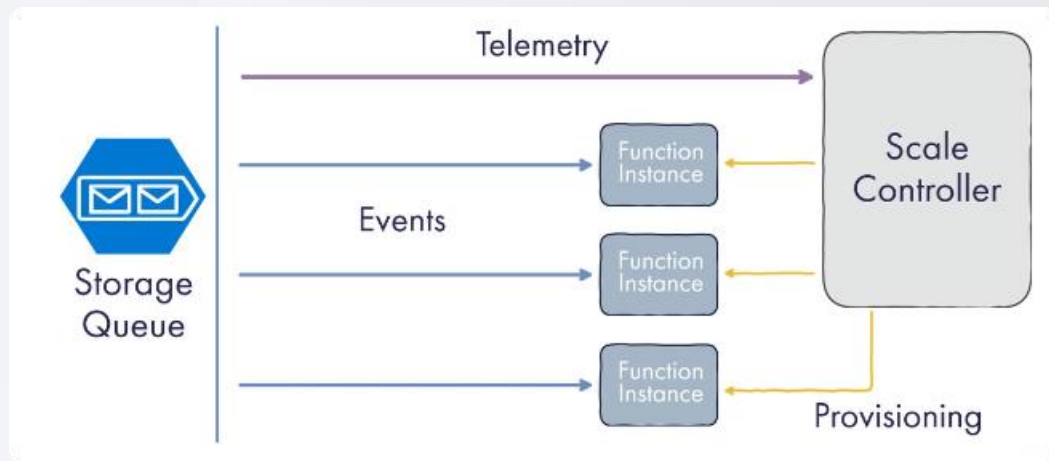
Distributed Tracing

Leverage distributed tracing tools to identify and debug issues across the various components of your serverless application.

Alerting and Notifications

Set up alerts and notifications to quickly respond to critical issues or unexpected events in your serverless application.

Scaling and Performance Considerations



1

Concurrency

Understand the platform's concurrency limits and design your functions to handle concurrent invocations efficiently.

2

Cold Starts

Optimize your functions to minimize cold start latency, which can occur when a function is invoked after a period of inactivity.

3

Resource Utilization

Monitor and optimize your functions' memory usage, CPU allocation, and other resource requirements to ensure optimal performance.

Best Practices and Security Considerations



1

Modular Design

Break down your application into small, reusable serverless functions to promote scalability and maintainability.

2

Secure Configurations

Properly configure your serverless functions' access permissions, environment variables, and event triggers to ensure security.

3

Monitoring and Logging

Implement comprehensive monitoring and logging to detect and respond to security incidents and operational issues.

4

Automated Testing

Develop a robust testing strategy, including unit, integration, and end-to-end tests, to ensure the reliability and security of your serverless application.

Thank You