

RAJALAKSHMI ENGINEERING COLLEGE

RAJALAKSHMI NAGAR, THANDALAM – 602 105



**RAJALAKSHMI
ENGINEERING COLLEGE**

An AUTONOMOUS Institution
Affiliated to ANNA UNIVERSITY, Chennai

CP23211 ADVANCED SOFTWARE ENGINEERING LABORATORY

LABORATORY RECORD

Name : _____SALMAN LATHEEF T A _____

Year / Branch : _____I – M.E. CSE_____

University Register No. : _____2116230711007_____

College Roll no : _____230711007_____

Semester : _____II_____

Academic Year : _____2023 – 2024_____

RAJALAKSHMI ENGINEERING COLLEGE
RAJALAKSHMI NAGAR, THANDALAM – 602 105

BONAFIDE CERTIFICATE

Name : _____SALMAN LATHEEF T A_____

Academic Year : 2023 - 2024 Semester :_II_ Branch :___CSE_

Register No :

2116230711007

Certified that this is the bonafide record of work done by the above student in the CP 23211- Advanced Software Engineering Laboratory during in the year 2023 – 24

Signature of the Faculty In-charge

Submitted for the Practical Examination held on __22/06/2024__

Internal Examiner

External Examiner

INDEX

CONTENT	PAGE NO
SOFTWARE REQUIREMENT SPECIFICATION	04
OVERVIEW OF THE PROJECT	07
SCRUM METHODOLOGY	11
USER STORIES	13
USE CASE DIAGRAM	15
NON FUNCTIONAL REQUIREMENTS	17
OVERALL PROJECT ARCHITECTURE	19
BUSINESS ARCHITECTURE DIAGRAM	21
CLASS DIAGRAM	24
SEQUENCE DIAGRAM	26
ARCHITECTURAL PATTERNS (MVC)	29

SOFTWARE REQUIREMENT SPECIFICATIONS (SRS)

EXP NO : 1

DATE : 05.03.2024

CONTENTS

1. INTRODUCTION	8
1.1 Purpose	8
1.2 Scope	8
2. OVERALL DESCRIPTION	8
2.1 Product Perspective	8
2.2 Features	8
2.3 User Classes and Characteristics	9
3. SPECIFIC REQUIREMENTS	9
3.1 Functional Requirements.....	9
3.1.1 Input & Output	9
3.1.2 Detection Capabilities	9
3.1.3 Scalability and Performance	10
3.1.4 Reporting and Analytics.....	10

3.1.5 Security and Privacy	10
3.2 Non-Functional Requirements	10
3.2.1 Usability	10
3.2.2 Performance	10
3.2.3 Security	11
4. EXTERNAL INTERFACE REQUIREMENTS	11
4.1 Social Media Platforms & Online Communities	11
4.2 Educational Institutions & Businesses	11
4.3 Researchers	11
4.4 Law Enforcement	11
4.5 General Considerations	12
5. CONCLUSION	12

Efficient Web App Deployment with Docker Swarm & Azure DevOps

OVERVIEW OF THE PROJECT:

The project endeavors to redefine cloud infrastructure management through the comprehensive utilization of Microsoft Azure's robust ecosystem. With a focus on automation, predictive analytics, and real-time monitoring, it aims to elevate scalability, optimize operational efficiency, and ensure unwavering application reliability on Azure. Central to its approach are advanced functionalities such as automated resource provisioning, predictive maintenance, and intelligent load balancing. These capabilities are designed to deliver substantial cost savings, enhance overall performance, and facilitate proactive management of cloud resources. By addressing the complexities inherent in cloud management today, the project aims to empower organizations to harness Azure's cutting-edge technologies for sustained growth and maintain a competitive edge in the digital landscape.

The initiative seeks to integrate seamlessly with Azure's extensive suite of services, leveraging its capabilities to streamline operations and drive innovation. Through automated resource provisioning, the project enables dynamic scaling of resources based on demand, optimizing resource allocation and responsiveness. Predictive analytics play a pivotal role in preemptively identifying potential issues and inefficiencies, thereby minimizing downtime and enhancing system reliability. Intelligent load balancing further enhances application performance by efficiently distributing traffic across instances, ensuring optimal utilization of resources during peak periods.

1. Introduction

In today's digital landscape, efficient management of cloud resources is crucial for organizations aiming to optimize costs, enhance scalability, and ensure robust performance of their applications. Our project focuses on leveraging Microsoft Azure's cloud services to achieve these goals through advanced automation, predictive analytics, and intelligent monitoring.

2. Objectives

The primary objective of this project is to develop a comprehensive Azure-based solution that streamlines resource management, improves application deployment efficiency, and enables proactive maintenance through predictive insights. By harnessing Azure's capabilities, we aim to deliver a scalable, secure, and cost-effective cloud infrastructure solution that meets the dynamic needs of modern businesses.

3. Scope

3.1 Product Perspective

Our solution integrates with various Azure services such as Azure Machine Learning, Azure Kubernetes Service (AKS), Azure Functions, and Azure Monitor. It provides a unified platform for managing virtual machines, containers, serverless functions, and other Azure resources efficiently.

Features

3.2 Functional Requirements :

- **Automated Resource Provisioning:** Dynamically scale Azure resources based on demand to optimize performance and cost-efficiency.
- **Predictive Maintenance:** Utilize machine learning models to forecast potential issues and proactively mitigate downtime.
- **Intelligent Load Balancing:** Automatically distribute traffic across instances to optimize application performance during peak loads.
- **Automated Deployment Pipelines:** Implement CI/CD pipelines using Azure DevOps for seamless application updates and releases.
- **Real-time Monitoring and Alerting:** Monitor Azure resources in real-time, with customizable alerts for critical thresholds and performance metrics.
- **Cost Optimization Recommendations:** Provide actionable insights and

recommendations to optimize Azure spending based on usage patterns and trends.

- **Security Compliance Automation:** Enforce Azure security best practices and regulatory compliance through automated checks and reporting.

3.3 Non Functional Requirements :

1. Performance

Response Time: All system actions, including automated provisioning, predictive maintenance alerts, and load balancing decisions, should respond within 5 seconds to ensure optimal user experience and operational efficiency.

2. Reliability

Uptime: The core services of the system should maintain a minimum uptime of 99.9% to ensure continuous availability and reliability for critical business operations.

3. Security

Data Encryption: Utilize encryption mechanisms (e.g., TLS/SSL) to secure data in transit between components and services within the Azure environment.

4. Usability

User Interface: Design an intuitive and user-friendly interface for administrators and operators to monitor, manage, and configure Azure resources, including dashboards for real-time monitoring and actionable insights.

4. External Interface Requirements

The project involves several external interface requirements to ensure seamless integration and functionality across different platforms and services:

1. **Integration with Azure Active Directory (AAD):** Utilize AAD for user authentication and authorization, ensuring secure access to Azure resources.
2. **API Endpoints for Third-Party Integration:** Provide well-documented API endpoints to facilitate integration with external systems and tools, enabling data exchange and interoperability.
3. **Support for Monitoring and Logging Protocols:** Implement support for standard monitoring and logging protocols such as Prometheus for metrics

collection and ELK stack for log management, ensuring comprehensive visibility and operational insights.

5. Conclusion

In conclusion, the project aims to transform cloud infrastructure management through Microsoft Azure's powerful capabilities in automation, predictive analytics, and real-time monitoring. By leveraging these advanced features such as automated resource provisioning, predictive maintenance, and intelligent load balancing, the project promises significant cost savings, improved performance, and proactive management of cloud resources. This initiative not only addresses current challenges in cloud management but also positions organizations to achieve sustainable growth and maintain a competitive edge in today's digital landscape. With a focus on scalability, operational efficiency, and application reliability, the project sets out to empower businesses to innovate and thrive in an increasingly dynamic and competitive environment powered by Azure's cutting-edge technologies.

SCRUM METHODOLOGY

EXP.NO: 2

DATE: 01.3.2024

1. Introduction

The SCRUM methodology provides a structured framework for managing software development projects. It emphasizes iterative development, collaboration, and flexibility to adapt to changing requirements throughout the project lifecycle.

2. Objectives

The objectives of implementing SCRUM in this project include:

- Facilitating regular and transparent communication among team members.
- Enhancing project transparency and stakeholder engagement through incremental delivery.
- Delivering high-quality software increments at the end of each sprint.
- Enabling continuous improvement through regular retrospectives and feedback loops.

3. Product Backlog Introduction

The product backlog serves as the master list of all features, enhancements, and fixes prioritized for implementation in the project. It is managed and prioritized by the Product Owner based on business value and stakeholder feedback.

4. Product Backlog

The product backlog includes:

- List of high-level features and functionalities.
- User stories detailing specific requirements and functionalities.
- Technical tasks and improvements identified during backlog grooming sessions.

5.1 User Stories

User stories are concise, user-focused descriptions of desired system functionality. They typically follow the format: "As a [role], I want [goal], so that [reason]."

6. Sprint

Sprints are time-boxed iterations (usually 2-4 weeks long) where development work takes place. Each sprint begins with Sprint Planning and concludes with a Sprint Review and Retrospective.

7. Sprint Backlog

The sprint backlog is a subset of the product backlog items selected for implementation during the current sprint. It includes user stories and tasks that the development team commits to completing.

8. Sprint Review

The Sprint Review meeting is held at the end of each sprint to demonstrate completed work to stakeholders. It provides an opportunity for feedback and ensures alignment with project goals.

9. Software Used

Key software tools and platforms used in the project include:

- Azure DevOps for project management, backlog tracking, and CI/CD pipelines.
- Git for version control and collaborative development.
- Microsoft Teams or similar for communication and collaboration among team members.

10. Conclusion

In conclusion, adopting the SCRUM methodology enables our project team to effectively manage complexity, respond to changes swiftly, and deliver value incrementally. By fostering collaboration and transparency, SCRUM empowers us to meet project goals while ensuring stakeholder satisfaction and continuous improvement throughout the development lifecycle.

USER STORIES

EXP.NO: 3

DATE: 12.3.2024

Automated Resource Provisioning

- **As a DevOps engineer**, automate the provisioning of Azure resources based on predefined templates to ensure rapid deployment and scalability.

Predictive Maintenance

- **As a system administrator**, receive proactive alerts and notifications about potential system failures predicted by machine learning models, enabling preemptive action to minimize downtime.

Intelligent Load Balancing

- **As an application owner**, dynamically distribute incoming traffic across Azure instances based on real-time analytics to optimize performance and ensure seamless user experience during peak usage periods.

Automated Deployment Pipelines

- **As a developer**, configure and automate CI/CD pipelines using Azure DevOps, enabling continuous integration and deployment of software updates across different environments with minimal manual intervention.

Real-time Monitoring Dashboard

- **As an operations manager**, access a centralized dashboard displaying real-time metrics and performance indicators of Azure resources and applications, facilitating proactive monitoring and rapid response to anomalies.

Cost Optimization Recommendations

- **As a business analyst**, receive automated reports and recommendations on optimizing Azure resource usage and costs based on usage patterns and historical data, ensuring efficient budget management without compromising performance.

Security Compliance Automation

- **As a security analyst**, implement automated security checks and audits across Azure resources to ensure compliance with industry standards and regulatory requirements, mitigating security risks and vulnerabilities proactively.

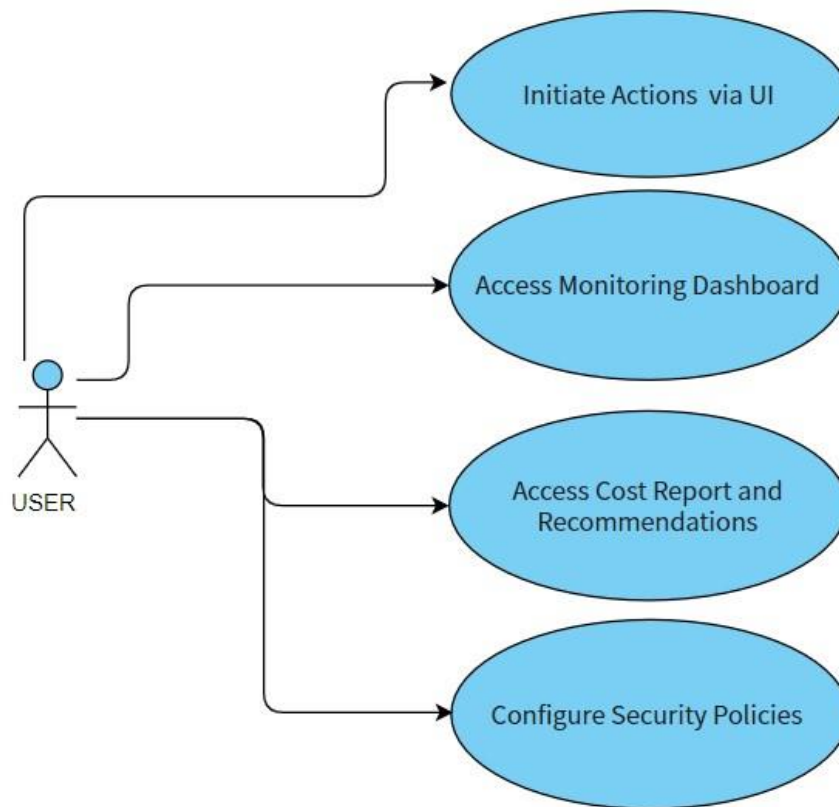
Integration with External Systems

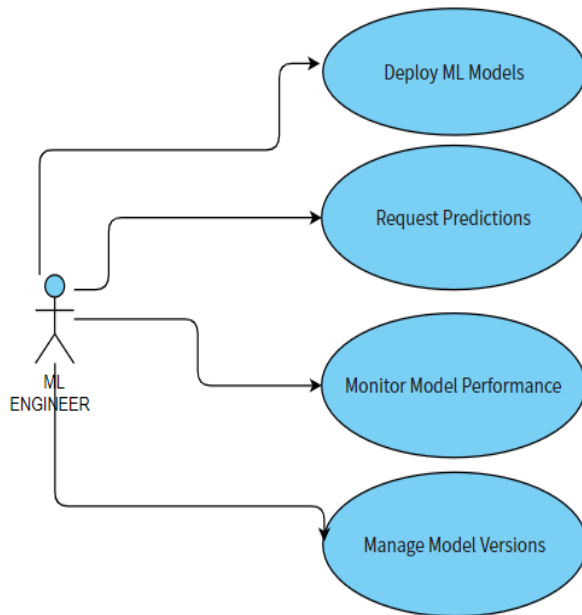
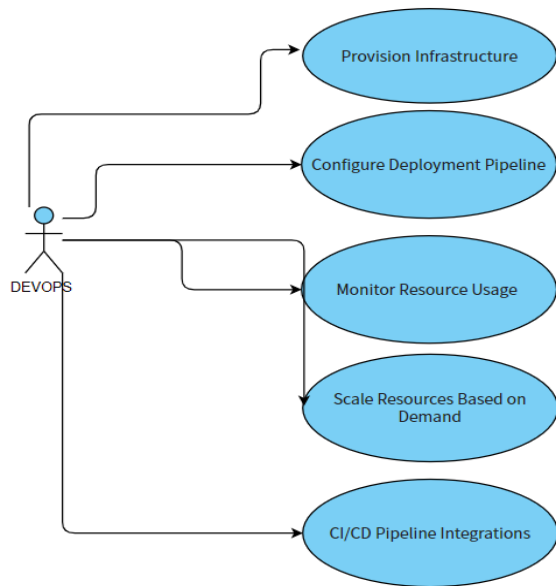
- **As an IT administrator**, configure API endpoints and integrate Azure services with third-party applications and systems, enabling seamless data exchange and interoperability while maintaining security and reliability.

USE CASE DIAGRAM

EXP.NO: 4

DATE: 19.3.2024





NON FUNCTIONAL REQUIREMENT

EXP.NO: 5

DATE: 29.3.2024

1.Performance:

- Ensure that system actions such as resource provisioning and scaling respond within 5 seconds to maintain responsiveness and user satisfaction.
- Support a minimum of 1000 concurrent users accessing the application without degradation in performance.

2.Scalability:

- Design the system to scale horizontally and vertically to accommodate increasing user loads and resource demands seamlessly.
- Ensure that the system can handle a 50% increase in workload during peak usage periods without performance degradation.

3.Reliability:

- Maintain a minimum uptime of 99.9% for core services to ensure high availability and reliability of the application.
- Implement automated failover and disaster recovery mechanisms to minimize downtime and ensure continuous service availability.

4.Security:

- Implement Azure security best practices and standards to protect sensitive data and ensure compliance with industry regulations (e.g., GDPR, HIPAA).
- Conduct regular security audits and vulnerability assessments to identify and mitigate potential security risks proactively.

5.Usability:

- Design an intuitive user interface with a learning curve of less than 2 hours for new users to navigate and perform basic tasks effectively.
- Provide comprehensive documentation and user guides to facilitate ease of use and minimize the need for extensive training.

6.Maintainability:

- Adopt modular and scalable architecture patterns to facilitate future enhancements and updates to the system components.
- Ensure that codebase adheres to coding standards and practices, enabling efficient maintenance and troubleshooting by development teams.

7.Compatibility:

- Ensure compatibility with latest versions of web browsers (Chrome, Firefox, Edge) and operating systems (Windows, Linux, macOS) to support diverse user environments.
- Integrate seamlessly with Azure services and third-party tools through standardized APIs and protocols to enable interoperability and data exchange.

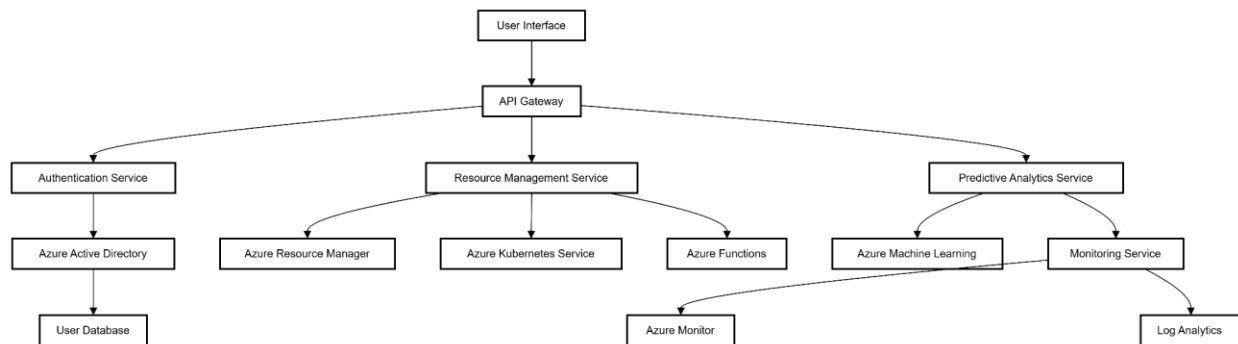
8.Performance Monitoring:

- Implement robust monitoring and logging mechanisms to track system performance metrics (CPU usage, memory utilization, response times) in real-time.
- Configure alerting systems to notify administrators of performance anomalies or threshold breaches for timely intervention and troubleshooting.

OVERALL PROJECT ARCHITECTURE

EXP.NO: 6

DATE: 09.4.2024



Components

User Interface: The web-based dashboard for users to interact with the system, providing a graphical interface for managing Azure resources and viewing performance metrics.

API Gateway: Acts as a centralized entry point for all API requests, handling authentication, authorization, and routing requests to appropriate backend services.

Authentication Service: Manages user authentication and authorization using Azure Active Directory (AAD), ensuring secure access to the application and Azure resources.

Resource Management Service: Orchestrates the provisioning, management, and scaling of Azure resources such as virtual machines (via Azure Resource Manager), containers (via Azure Kubernetes Service), and serverless functions (via Azure Functions).

Predictive Analytics Service: Integrates Azure Machine Learning for predictive analytics, forecasting resource demands, identifying potential issues, and optimizing resource allocation proactively.

Monitoring Service: Collects and analyzes real-time metrics and logs from Azure resources using Azure Monitor and Log Analytics, providing insights into system

health, performance, and operational efficiency.

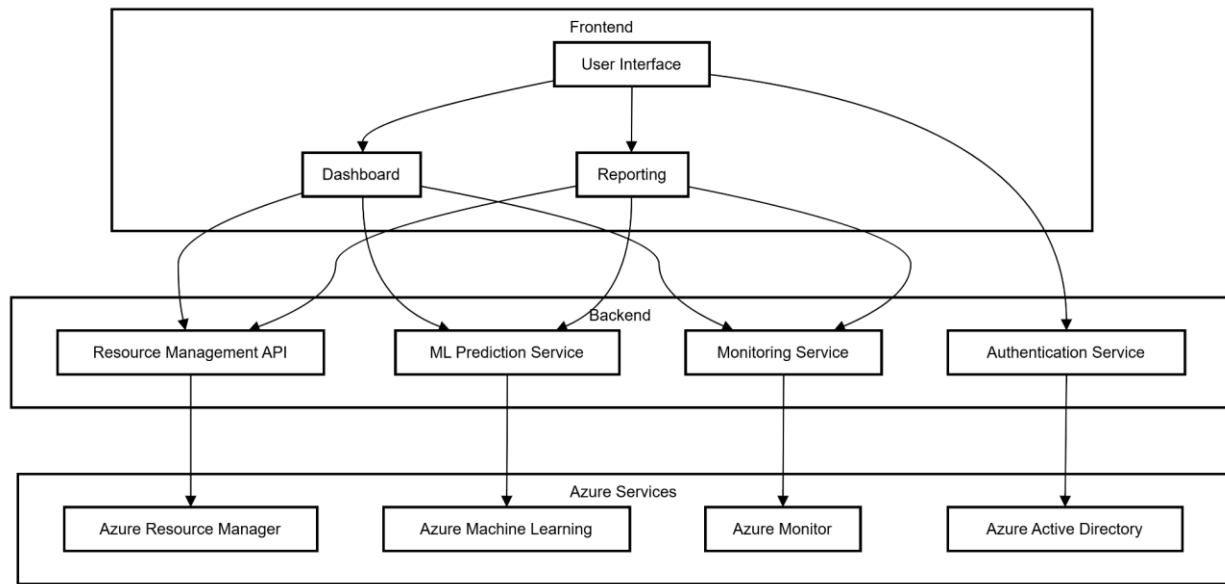
Azure Active Directory (AAD): Manages identities and access for users and applications, ensuring secure authentication and single sign-on (SSO) capabilities across the Azure ecosystem.

User Database: Stores user profiles and permissions managed through Azure Active Directory, facilitating centralized user management and authentication.

BUSINESS ARCHITECTURE DIAGRAM

EXP.NO: 7

DATE: 19.4.2024



Frontend Modules:

1. User Interface (UI):
 - Main entry point for users to interact with the system
 - Provides access to all features and functions
 - Ensures a user-friendly experience for managing cloud resources
2. Dashboard:
 - Displays real-time information about cloud resources
 - Allows users to view and manage Azure resources
 - Presents key performance indicators and resource utilization metrics
3. Reporting:
 - Generates detailed reports on resource usage, costs, and performance
 - Provides customizable analytics and data visualization tools
 - Allows exporting of reports in various formats

Backend Modules:

4. Resource Management API:

- Handles requests for provisioning, scaling, and deprovisioning Azure resources
- Implements logic for optimal resource allocation
- Interfaces with Azure Resource Manager to execute resource management tasks

5. ML Prediction Service:

- Runs machine learning models for predictive analytics
- Forecasts resource needs and potential issues
- Provides recommendations for optimizing resource usage

6. Monitoring Service:

- Collects real-time data on resource performance and utilization
- Analyzes collected data to identify trends and anomalies
- Triggers alerts based on predefined thresholds

7. Authentication Service:

- Manages user authentication and authorization
- Integrates with Azure Active Directory for secure access control
- Ensures that users can only access resources they're authorized to use

Azure Services:

8. Azure Resource Manager:

- Central service for deploying and managing Azure resources
- Provides a consistent management layer for all Azure resources
- Enables applying access control to all services

9. Azure Machine Learning:

- Platform for developing, training, and deploying machine learning models
- Provides tools and services to build and manage ML workflows
- Enables scaling of ML operations for large-scale predictions

10. Azure Monitor:

- Comprehensive solution for collecting, analyzing, and acting on telemetry from Azure resources
- Provides full observability into applications, infrastructure, and network
- Enables creating dashboards and setting up alerts

11. Azure Active Directory (AD):

- Cloud-based identity and access management service

- Enables employees to access external resources
- Helps secure access to Azure resources and applications

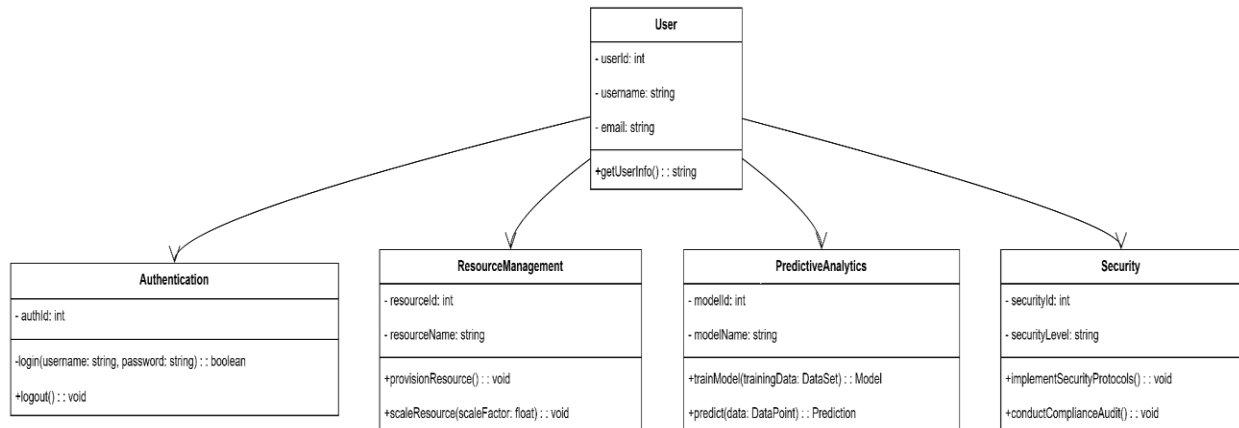
These modules work together to provide a comprehensive cloud optimization solution:

- The frontend modules offer an intuitive interface for users to interact with the system.
- The backend modules handle the core logic, data processing, and integration with Azure services.
- The Azure services provide the underlying infrastructure and platforms necessary for managing cloud resources, running ML models, monitoring performance, and handling authentication.

CLASS DIAGRAM

EXP.NO: 8

DATE: 30.4.2024



1. User Class:

- Represents a user in the system with basic information and methods to retrieve user details.

2. Authentication Class:

- **Explanation:** Manages user authentication and login/logout functionality.

3. ResourceManagement Class:

- **Explanation:** Handles the provisioning and scaling of resources within the system.

4. PredictiveAnalytics Class:

- **Explanation:** Implements predictive analytics functionalities including model training and prediction.

5. Security Class:

- **Explanation:** Manages security protocols and conducts compliance audits within the system.

● Relationships:

- **Authentication:** Each user can authenticate with the system.
- **ResourceManagement:** Users can manage and utilize system resources.
- **PredictiveAnalytics:** Users can utilize predictive analytics capabilities.
- **Security:** Users interact with security features and compliance audits.

SEQUENCE DIAGRAM

EXP.NO: 9

DATE: 10.5.2024

User Interface (UI):

- Initiates API requests to start processes, interacts directly with the system.

API Gateway:

- Receives and manages incoming requests from UI and other clients, including authentication and routing to appropriate services.

Authentication Service:

- Handles user authentication and authorization processes, ensuring secure access to system resources.

Resource Management Service:

- Manages Azure resources such as virtual machines, databases, and storage accounts. Handles provisioning, configuration, and deployment based application requirements.

ML Prediction Service:

- Executes machine learning models deployed on Azure Machine Learning. Receives input data, processes it through ML models, and provides predictions or insights.

Monitoring Service:

- Collects real-time telemetry data from various Azure resources (e.g., VMs, databases, containers). Monitors performance metrics, availability, and health status.

Azure Resource Manager (ARM):

- Manages Azure resources lifecycle, including creation, update, and deletion. Provides a unified management layer for organizing and controlling resources.

Azure Machine Learning (AML):

- Executes machine learning workflows, including model training, evaluation, and deployment. Integrates with ML Prediction Service to provide predictive capabilities.

Azure Kubernetes Service (AKS):

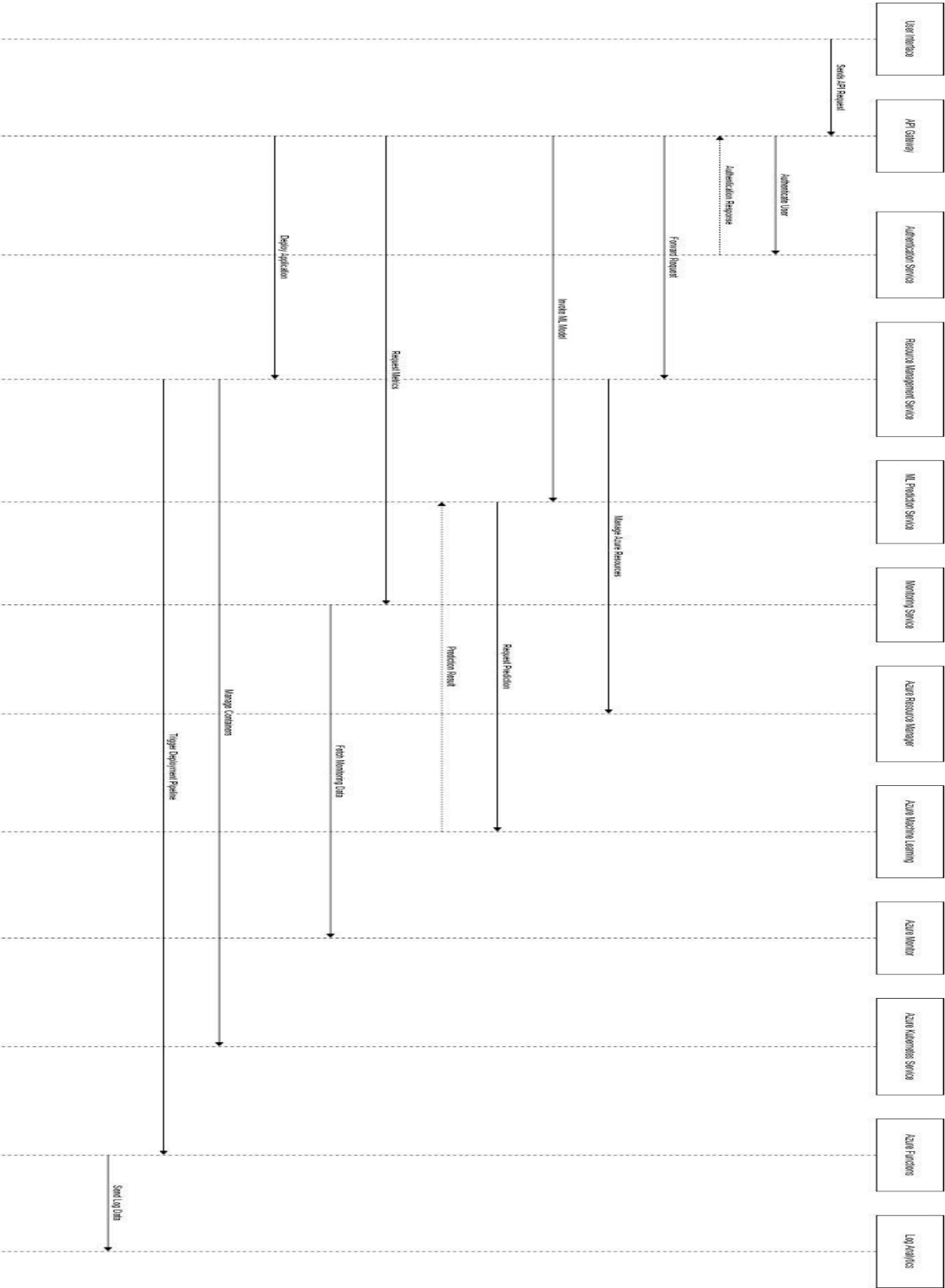
- Manages containerized applications using Kubernetes orchestration. Scales applications automatically based on demand, ensuring optimal performance and resource utilization.

Azure Functions:

- Executes serverless compute logic in response to events or triggers. Supports deployment pipelines, automation tasks, and background processing without managing infrastructure.

Log Analytics:

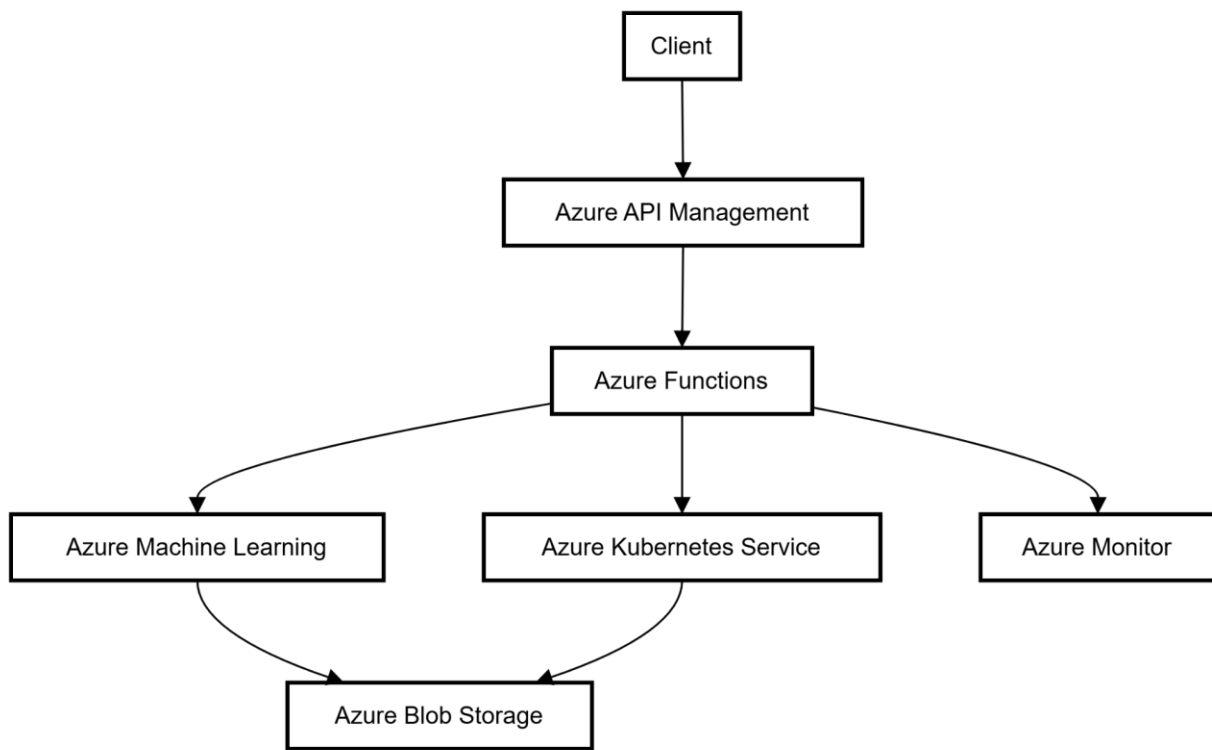
- Collects, analyzes, and visualizes log data and telemetry from Azure Functions and other services. Provides insights into application and system performance, troubleshooting, and security auditing.



ARCHITECTURAL PATTERN

EXP.NO: 10

DATE: 17.5.2024



Module Explanation

1. Client :

- Represents the external client or application interacting with the Azure services.
- Initiates requests and consumes APIs exposed by Azure API Management.

2. Azure API Management:

- Provides a gateway for managing and securing APIs.
- Handles incoming requests from the Client, performs API policies, and forwards requests to downstream services.

3. Azure Functions :

- Serverless compute service that runs code on demand.
- Receives requests from Azure API Management, executes specific functions or logic, and integrates with other Azure services.

4. Azure Machine Learning:

- Cloud-based platform for building, training, and deploying machine learning models.
- Azure Functions can invoke machine learning models deployed on Azure Machine Learning for predictions or data processing tasks.

5. Azure Kubernetes Service:

- Managed Kubernetes service for container orchestration.
- Azure Functions can interact with Azure Kubernetes Service to manage containerized applications or microservices.

6. Azure Blob Storage :

- Object storage service for storing large amounts of unstructured data.
- Azure Machine Learning can store and retrieve datasets or model artifacts in Azure Blob Storage.

7. Azure Monitor :

- Platform service for monitoring Azure resources and applications.
- Azure Functions and Azure Kubernetes Service can send telemetry data and logs to Azure Monitor for monitoring, analysis, and alerting purposes.