Software Requirements Specification (SRS) for Cloudifi - Cloud Cost Optimizer

**1. Problem Analysis**

**1.1 Aim**

The goal of the Cloudifi project is to create a platform for optimizing cloud costs by helping you easily scale out/in your virtual instances manually as well automatically. It is specifically designed to help businesses drive down the contract value of cloud infrastructures while making sure applications perform at maximum performance levels. Cloudifi offers deep insights into costs, allows you to customize your scaling policies and provides an intelligent real-time algorithm for reducing cloud waste.

**1.2 Scope**

Cloudifi is a one size fits all solution and supports the following user roles:

**Developer, Admin, and Executive.**

**For Developers:**

Provisioning and de-provisioning applications on Virtual Machines

Requesting changes to application resource limits.

Updating Application-Specific Scale Policies

Detailed breakdown of costs vs. savings of applications

**For Admins:**

Limit resources and applications for a user from doing any desired action.

Updating instance types.

Managing user accounts, creation and deletion of the users.

Detailed Application Overviews

**For Executives:**

Developers getting quick, high-level glimpses of app costs and savings

It allows to scale numerous things on your platform and do so at a daily, weekly or monthly basis . These operations are like scaling up/down of instances, terminating instances or changing the types of instances. Smartifi algorithm - Automated scaling based on real-time utilization metrics for most efficient auto-scaling to up-scale (or down-scale) necessary resources to meet demand without losing performance or cost benefits.

**1.3 Overview**  
Cloudifi addresses the critical need for cloud resource efficiency and cost efficiency in today’s cloud-deployment environment. The main features of the platform are:  
  
Application Management: Allows developers to create, delete, and manage applications in virtual instances.  
Cost Monitoring: Provides real-time insight into actual costs and potential savings calculated for each application.  
Scheduling Functionality: Enables users to define and apply scaling schedules on a daily, weekly or monthly basis, specifying start times and actions such as scale up/down scaling and standing by instances  
Scaling Operations: Provides manual and systematic scaling operations, including:  
All instances stop.  
Keep track of all the times.  
Reduce the instances.  
Increase the instances.  
Scale down the instance type.  
Scale up the instances.  
Smartifi algorithm: Monitors usage metrics every minute and automatically scales instances based on a predefined threshold (scale up if usage is less than 50%, scale up if greater than 80%).  
By integrating these features, Cloudifi ensures more efficient use of resources, significant cost savings, and improved efficiency for cloud-based applications

**1.4 Target audience**  
Cloudifi’s key target audiences include:  
  
Developers: These users are responsible for developing and maintaining applications in virtual instances. They need tools to create, delete, and modify applications and systems, and track costs and savings.

Administrators: These users are responsible for the overall management of services and resources. They handle application and user restrictions, update instance types, and manage user accounts.

Entrepreneurs: These users are focused on finding higher levels of spending and savings. Access to application metrics is needed to make informed investment decisions and develop resource allocation plans.

**1.5 Definitions (Acronyms and abbreviations) .**  
Cloudifi: The name of the cloud cost optimization platform.

Developer: The user role responsible for creating, deleting, and maintaining applications in virtual instances.

Admin: User role that manages application and user restrictions, manages instances, and manages user accounts.

Staff: Implementation focused on identifying spending and high levels of savings insight.

Scheduling: A set of rules governing how and when scaling operations are performed.

Scaling: The process of changing the number or type of virtual instances based on resource usage and application needs.

Smartifi algorithm: An intelligent algorithm that makes scaling decisions based on real-time usage data for efficiency and cost-effectiveness.

USP (Unique Selling Proposition): What sets the platform apart from competitors.

**1.7 References**  
The following articles provide additional information and background information on the Cloudifi project:  
  
Amazon Web Services (AWS) Auto Scaling Documentation: This resource provides detailed information about AWS auto-scaling features, providing insight into industry best practices for scaling cloud resources AWS auto-scaling documentation

Cloud Cost Management Best Practices : This article from Google Cloud outlines options for optimizing cloud costs, providing valuable tips and tricks on the Cloudifi platform Cloud Cost Management Best Practices

Managing Virtual Instances on Azure : This documentation from Microsoft Azure includes managing virtual machine instances, virtual instance management that provides insight into managing cloud resources efficiently

Scaling Policy Strategies by Red Hat: This resource discusses effective strategies for scaling policy, helping to identify Cloudifi’s policy management features Scaling Policy Strategies

By leveraging these references, the Cloudifi platform can be developed with a solid foundation of industry knowledge and best practices, ensuring it caters to its target audience needs and optimize its cloud cost-effectiveness objectives.

**Software Requirements Specification (SRS) for Cloudifi - Cloud Cost Optimizer**

**2. Software requirements analysis and planning**

**2.1 Major system components/connections:**

Cloudifi is organized around several key features and interfaces: •

**User Interface (UI):**

This feature provides the basic interface for users (developers, admins, executives). It includes dashboards and documentation for monitoring applications, monitoring costs, and setting up scaling plans.

**• Application Management Module:**

Manages the activities associated with applications running virtual instances. This module allows users to create new applications, delete existing ones, and manage specific settings such as resource limits and scaling settings

**. • Policy management module:**

This module allows users to define and apply scaling policies. It includes options for daily, weekly, or monthly schedules, specifying start times and actions (e.g., scale instances up/down, to finish instances).

**• Resource Monitoring Module**:

Monitors real-time usage metrics of applications, providing insight into actual costs and potential savings. It sends data to the Smartifi Algorithm for scaling decisions.

**• Smartifi algorithm module:**

uses real-time usage metrics from the resource monitoring module to automate scaling decisions. It changes the number or quality of samples based on a predefined threshold (e.g., reduce if utilization is less than 50%, reduce if it is more than 80%).

**• The user management module**

handles user authentication, authorization, and account management functions. Ensures security of access and usage-based authorization across the platform.

**2.2 Description of Individual Phases/Modules**

**2.2.1 User Characteristics**

* **Developers:** Proficient in application development and cloud infrastructure management. They require tools for creating, deleting, and configuring applications, as well as monitoring their resource usage and costs.
* **Admins:** Experienced in resource allocation and management. Admins oversee the overall application portfolio, manage user permissions, update instance types, and ensure adherence to organizational policies.
* **Executives:** Focus on high-level financial insights and strategic decision-making. They rely on summarized cost and savings reports to optimize resource allocation and budgeting.

**2.2.2 General Constraints - Design Operating Environment**

Cloudifi works within specific restrictions and environmental considerations:

**• Cloud environment:**

The platform runs in a cloud service provider such as AWS, Azure, or Google Cloud. It uses their APIs for instance management and cost tracking. •

**Web Accessibility:**

Users access Cloudifi through a common web browser (e.g., Chrome, Firefox, Edge) on a variety of devices (desktop, tablet, mobile phone). •

**Scalability requirements:**

Accommodate the growing number of users and applications on the platform, scaling horizontally (adding more instances) and vertically (upgraded hardware resources). •

**Performance expectations:**

Cloudifi expects to efficiently handle real-time data processing and scaling processes for performance metrics, ensuring minimal latency and high responsiveness •

**Security Standards:**

Adherence to industry standards for data security (data encryption in transit and at rest), user authentication, licensing, and secure API integration. •

**Integration with external systems:**

Seamless integration with third-party APIs (e.g., cloud service providers, identity management systems) for improved performance and data synchronization

**2.2.3 Assumptions**

**• Practical Knowledge:**

Users are assumed to have basic knowledge of cloud computing concepts, including virtual instances, scaling operations, and cost management •

**Internet connectivity:**

Users moving to Cloudifi need a stable Internet connection to ensure continuous access to the platform and real-time data updates. •

**Platform reliability:**

The platform assumes reliable operation under normal operating conditions, with mechanisms in place to quickly monitor and address potential downtime or service interruptions.

**2.2.4 Functional Requirements Input/Output Description**

**Input Requirements:1. Application Creation and Management:**o **Create Application:** **Input:** Developers input application details such as application name, description, required instance type, initial resource allocation (CPU, memory, storage), and deployment configuration.

**UI Interaction:** Form-based input on the application management page. **Validation:** Ensure that the application name is unique, and required fields are completed correctly.o **Delete Application:** **Input:** Developers select an existing application to delete. **UI Interaction**: Selection from a list of applications with a confirmation dialog. **Validation:** Confirmation prompt to avoid accidental deletions.

o **Update Application:** **Input**: Developers can update application details and resource allocation. **UI** **Interaction**: Editable fields within the application management page. **Validation**: Check for valid and permissible resource limits.2. **Threshold** **Management**:o **Request** **Threshold** **Change**: **Input:** Developers input new resource thresholds for CPU, memory, or storage. **UI Interaction**: Form-based input or slider adjustments on the application settings page. **Validation**: Validate against predefined organizational limits.3. **Scaling Policy Definition:**o **Define New Policy**: **Input**: Developers input policy details such as policy name, scaling actions (scale up/down, standby, terminate), schedule (daily, weekly, monthly), and specific start times. **UI Interaction**: Policy creation wizard or form on the policy management page. **Validation:** Ensure that policy parameters are within permissible ranges and schedules do not conflict with existing policies.

* + **Modify Existing Policy:**
    - **Input:** Developers select an existing policy to modify and update the parameters.
    - **UI Interaction:** Editable fields in the policy management interface.
    - **Validation:** Validate updates against current application state and other policies.
  + **2. User Account Management:**
  + o **Create a user account:** **Input:** Administrators enter additional user information such as username, email, role (Developer, Admin, Executive), and initial password. **UI interaction**: text-based forms on the user management page. **Verification:** Ensure unique username, valid email format, and assigned roles.
  + o **Delete the user account:** **Input:** Administrators specify a user account to delete.** UI interaction**: select from a username in a confirmation dialog. **Verify**: Ensure that it is not dependent on critical performance by issuing a confirmation letter.
  + o **Update the user account:** **Input:** Admins enter updated user information or change user roles.** UI Interface**: Fields that can be edited between user programs. **Validation**: Validation against organizational policies and operational boundaries.
  + 3. **Real-time performance metrics**:
  + o **Input:** The system receives continuous data from the cloud provider's monitoring services, including metrics such as CPU usage, memory usage, disk I/O, and network bandwidth**o Source:** Cloud Service Provider API.o **Validation**: Verify data accuracy and precision in real time.
  + **Requirements:**
  + **1. Project overview:o Expenditure Statement: .** Results: A detailed breakdown of the actual costs incurred by each application, including compute, storage, and network processing costs.** UI Interaction**: The design of the application is displayed in the dashboard with visuals (charts, graphs).**o Potential savings:** Results: Potential savings were calculated based on current management plans and opportunities for efficiency. UI Interface: Banking opportunities are highlighted in the application overview dashboard.**o Performance status**: Output: Real-time status of applications, including current resource usage, instance status (running, standby, finished), and health checks.
  + **UI Interaction:** Status indicators and detailed reports in the application overview section.

1. **Scaling Operations Feedback:**
   * **Manual Operations:**

Output:

Confirmation and status updates for manual scaling operations performed by users (e.g., instances scaled up/down, put on standby, completed). **UI interaction**: instant messaging, activity logging, new workflow status.o Automation using Smartifi algorithm: **Results:** Detailed reports on automated scaling operations based on real-time usage metrics. This includes actions taken (scaled up/down), instances affected, and time. **UI interface:** Report structure and detailed information accessible through a system management dashboard.

1. **Alerts and Notifications:**

**Scaling Warning: .** **Output:** Warning for critical scaling events, such as instances scaled up/down due to meeting usage thresholds. **UI interaction:** in-app notifications, email alerts, and log entries.o System Performance Reporting:** Results**: A report on the implementation of the scaling plan, detailing the actions taken and the results.** UI Interface**: Available through system management interface with options for downloading advanced reports.

1. **User Activity Logs:**

**o Output**: Records of user activity, including application creation/deletion, configuration changes, and scaling operations.**o UI Interface**: Admin interface for reviewing user activities, with filtering and search capabilities.

1. **Real-time dashboard metrics:**o **Production**: real-time display of critical metrics such as total resources, ongoing costs, and utilization percentage.o **Interactive UI:** A dynamic dashboard with interactive charts and graphs, allowing users to explore specific data points.

**2.2.5 Non-Functional Requirements**

**• Performance**: Perform scaling operations and data updates in real time without compromising system performance, ensuring minimal latency for critical tasks**• Functionality**: The UI should be intuitive, helping to facilitate navigation and interaction for all users (developers, admins, executives).**• Security**: Strong methods of authentication, data encryption, and role-based control (RBAC) to protect sensitive information and ensure compliance with data privacy regulations . . . .• **Reliability:** High availability with minimal downtime, supported by automated backup and recovery processes to protect data integrity and service continuity.• **Scalability:** Ability to scale horizontally and vertically to better meet the demands of a growing user base and increasing workload.

**2.3 Identify individual module deliverables**• **UI Modules**: Deliverables include wireframes, mockups, and UI design specifications to ensure a smooth and accessible user interface.• **Application Management Module**: The deliverable contains functionality for creating, deleting, and managing applications, as well as interfaces for configuring scaling policies and viewing application information• **Policy Management Module**: Includes tools for defining and applying scaling policies to deliverables, developing policy plans, and monitoring policy quality• **Features Monitoring Module**: Features to be delivered include real-time monitoring of application costs, usage metrics, and automatic data feeds to the Smartifi algorithm.• **Smartifi Algorithm Module**: Deliverables include real-time performance metrics and algorithmic logic for automatic scaling decisions based on preset thresholds• **User management module**: Features to be delivered include authentication mechanisms, user management interfaces, and account management services to ensure secure and controlled access to the platform.

**2.4 External Interface Requirements**

• User Interface: Responsive UI that can be accessed via a standard web browser (Chrome, Firefox, Edge) on a variety of devices (desktop, tablet, mobile phone).• API: Interfaces with APIs from cloud service providers (e.g., AWS, Azure) for instance management, cost tracking, and data manipulation.• Secure Connectivity: Provide integration with identity management systems (e.g., OAuth, LDAP) for user authentication and authorization, ensuring secure access to platform resources

**2.5 Logical Database Requirements**

* **Database System**: A relational database management system (RDBMS) such as PostgreSQL or MySQL is used to store application data, user profiles, scaling policies, and usage metrics.• **Database schema:** A set of schema structures including tables, relationships, and indexes for efficient data retrieval, storage, and processing.• Data Integrity: Applying restrictions (e.g., unique restrictions, special foreign connections) to maintain data integrity and accuracy.**• Backup and Recovery**: Planned backups, transaction logging, and disaster recovery processes to protect against data loss and ensure database reliability..