



SOFTWARE DESIGN SPECIFICATION

<Cloud Computing>  
  
<Automation of MongoDB database backup>

|  |  |  |  |
| --- | --- | --- | --- |
| **Created By:** | Yogesh.D.Khairnar | **Approved By:** | <Domain Lead Name> |
| **Created On:** | 21-11-2023 | **Approved On:** | DD-MMM-YYYY |

Page left blank intentionally

**INDEX**

[**1** **PURPOSE** 2](#_Toc142418236)

[**2** **PROJECT SCOPE** 2](#_Toc142418237)

[**3** **SYSTEM OVERVIEW** 2](#_Toc142418238)

[**4** **DESIGN CONSIDERATIONS** 2](#_Toc142418239)

[4.1 Requirements 3](#_Toc142418240)

[4.2 Assumptions 3](#_Toc142418241)

[4.3 Dependencies 3](#_Toc142418242)

[**5** **SYSTEM ARCHITECTURE** 3](#_Toc142418243)

[5.1 Architectural Strategies 4](#_Toc142418244)

[5.2 Structure & Relationships 4](#_Toc142418245)

[**6** **DETAILED DESCRIPTION OF COMPONENTS** 4](#_Toc142418246)

[**7** **INTEGRATION** 5](#_Toc142418247)

[**8** **APPENDICES** 1](#_Toc142418248)

[8.1 Appendix A – Detailed Description of Components 1](#_Toc142418249)

**General Instructions for using the Live Project POC Document**

* This template and the subsequent document created using this template is a confidential document and is the intellectual property of Cloud Counselage Pvt. Ltd. Circulating it outside of the organisation without the consent of Cloud Counselage Pvt. Ltd. is the breach of company policies and will lead to legal actions
* The Design Specification of a software forms the basis of development of software
* The **text between inequality (< >) is to be replaced** by relevant text
* Please **remove the yellow highlight on the Text** between the inequality (< >). This is done to help you notice the text to be changed/replaced
* The text in *italics* highlighted in grey is just for reference and should be removed after adding the relevant text

# **PURPOSE**

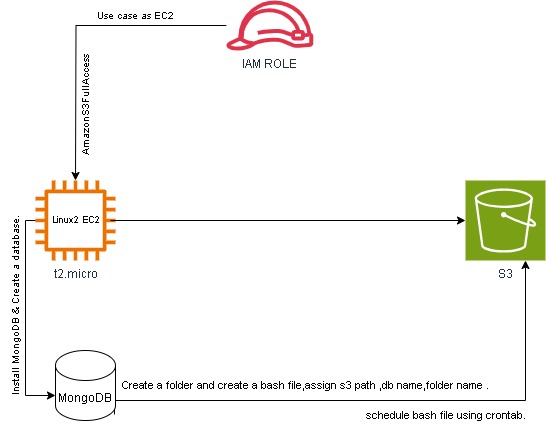
This document is created based on the requirement specification document. The purpose of this Software Design Specification (SDS) Document is to break down the project into components to describe in detail what the purpose of each component is and how it will be implemented. The SDS will also serve as a tool for verification and validation of the final product.

# **PROJECT SCOPE**

The scope of the Automation of MongoDB database backup includes its distinct features, its benefits, and its limitations. The system's distinct features allow it to automate the backup process for MongoDB database and store backup in AWS S3.The system enables the user to utilize the remaining time in other work as the whole process is based on automation so its less time consuming.

# **SYSTEM OVERVIEW**

This section will provide an outline of the various components and subsystems of MongoDB database backup automation.



# **DESIGN CONSIDERATIONS**

This section describes requirements, assumptions and dependencies to be addressed to devise a complete design solution.

## Requirements

Ec2 instance,IAM Role,S3 bucket,MongoDB,Backup script. The list of components

## Assumptions

EC2 Iinstance,Backup script-Stable connectivity between EC2 and MongoDB .The list of components

## Dependencies

Successful configuration of EC2,IAM Role configured with appropriateS3 permissionThe list of components.

# **SYSTEM ARCHITECTURE**

The software system architecture refers to the logical organization of a distributed system into software components. It defines how components of a software system are assembled, their relationship and communication between them. It serves as a blueprint for software application and development basis for developer team. An effective architecture serves as the conceptual glue that holds every phase of the project together for all of its stakeholders, enabling agility, time and cost savings, and early identification of design risks.

The Software architecture:

* Defines structure of a system
* Defines behaviour of a system
* Defines component relationship
* Defines communication structure
* Balances stakeholder’s needs
* Influences team structure
* Focuses on significant elements
* Captures early design decisions

Below some important characteristics which are commonly considered are explained.

**Operational Architecture Characteristics:**

* Availability
* Performance
* Reliability
* Low fault tolerance
* Scalability

**Structural Architecture Characteristics:**

* Configurability
* Extensibility
* Supportability
* Portability
* Maintainability

**Cross-Cutting Architecture Characteristics:**

* Accessibility
* Security
* Usability
* Privacy
* Feasibility

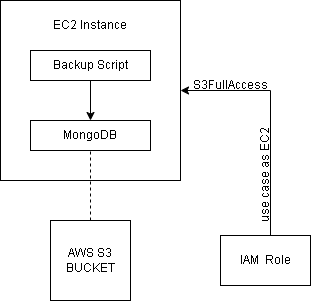
## Architectural Strategies

Components:-1) EC2 Instance-Launch an instance,Installation of MongoDB setup,host the backup.sh file.

2) AWS S3 Bucket- Storage for backup file.

3) AWS IAM Role- Grant permission for ec2 instance to access S3

## Structure & Relationships



# **DETAILED DESCRIPTION OF COMPONENTS**

For detailed description of the components, please refer **Appendix A – Detailed Description of Components**

The below template will be used to specify the details of all the components

**Table 1: Detailed Design Specification Template**

|  |  |
| --- | --- |
| **Identification** | The unique name for the component and the location of the component in the system. |
| **Type** | A module, a subprogram, a form, a data file, a control procedure, a class, etc. |
| **Purpose** | Function and performance requirements implemented by the design component, including derived requirements. Derived requirements are not explicitly stated in the SRS - but are implied or adjunct to formally stated SDS requirements. |
| **Subordinates** | The internal structure of the component, the constituents of the component, and the functional requirements satisfied by each part. |
| **Dependencies** | How the component’s function and performance relate to other components. How this component is used by other components. The other components that use this component. Interaction details such as timing, interaction conditions (such as order of execution and data sharing), and responsibility for creation, duplication, use, storage, and elimination of components. |
| **Interfaces** | Detailed description of all external or internal interfaces as well as of any mechanism for communicating through messages, parameters, or common data areas. All error messages and error codes should be identified. All screen formats, interactive messages, and other user interface components (originally defined in the SRS) should be given here. |
| **Resources** | A complete description of all resources (hardware or software) external to the component but required to carry out its functions. |
| **Processing** | A full description of the functions presented in the Function subsection. Pseudocode can be used to document algorithms, equations, and logic. |
| **Data** | For the data internal to the component, describes the representation method, initial values, use, semantics, and format. |

# **INTEGRATIONS**

1. **EC2 Instance Setup.**
   1. **Launch EC2 Instance- Launch EC2 Instance with AMI as Amazon Linux 2.**
   2. **Install MongoDB Tools-Connect to EC2 with direct connect,Install MongoDB.**
2. **AWS S3 Configuration.**

**2.1) Create S3 Bucket-Create a bucket and a folder where the backup will be stored.**

**2.2) IAM Role for EC2-Create IAM Role with use case as EC2,then give Amazon S3 FullAccess permission.**

1. **Backup Script Integration.**

**3.1) Script Development-Create backup file(.sh file),Assign the folder name,database name and S3 path.**

**3.2) Execution and Scheduling**

**-Execute the file,if it works properly then schedule it by using cron command.After scheduling reexecute the file.**

# **APPENDICES**

## Appendix A – Detailed Description of Components

|  |  |
| --- | --- |
| **Identification** | **EC2.** |
| **Type** | Computing Resources. |
| **Purpose** | Host virtual server,Execute backup script,Interact with other AWS Services. |
| **Subordinates** | Backup Script |
| **Dependencies** | IAM Role |
| **Interfaces** | AWS Management console,CLI |
| **Resources** | t2.micro,AMI- Amazon Linux2 |
| **Processing** | Runs MongoDB backup script. |
| **Data** | Store temporary backup file |

|  |  |
| --- | --- |
| **Identification** | **Backup Script** |
| **Type** | Bash script. |
| **Purpose** | Automate mongodb backup and upload to s3. |
| **Subordinates** | Mongodb tools. |
| **Dependencies** | EC2 instance setup. |
| **Interfaces** | AWS CLI,Mongo terminal. |
| **Resources** | Script file,mongo tools. |
| **Processing** | Backup,s3 upload. |
| **Data** | Temporary storage on EC2. |

|  |  |
| --- | --- |
| **Identification** | **AWS S3 Bucket** |
| **Type** | Storage resource. |
| **Purpose** | Stores mongodb backup. |
| **Subordinates** | Backup files. |
| **Dependencies** | IAM Role permission. |
| **Interfaces** | AWS Management Console. |
| **Resources** | S3 Bucket. |
| **Processing** | Storage. |
| **Data** | Mongodb backup files. |

|  |  |
| --- | --- |
| **Identification** | **IAM Role** |
| **Type** | Access Management |
| **Purpose** | Provide EC2 instance with permission for s3 access. |
| **Subordinates** | EC2 instance. |
| **Dependencies** | S3 bucket configuration. |
| **Interfaces** | AWS IAM console |
| **Resources** | IAM role. |
| **Processing** | Give S3 access to EC2. |
| **Data** |  |