Software requirements

Application name:

"Intelligent incremental backup system for web servers"

Ondrej Makši, Filip Lukáč,

Marek Kukučka, Ivan Martynenko

Table of contents

Introduction

1.1. Purpose of requirements document

This is a formal document created with the standard IEEE/ANSI 830-1998 (Recommended Practice for Software Requirements Specifications). The main purpose of this document is to specify all the requirements for the system, which was created as a project in the subject "Development of Information Systems" in the Faculty of Mathematics Physics and Informatics (FMFI UK). The requirements document is dedicated to all stakeholders. It can also work as an agreement between the developers and the client.

1.2. Scope of the product

The main goal is to create a system that can create backup copies of websites used in the Faculty of Mathematics Physics and Informatics (UK BA) with the ability to store them on a local disk as well as remotely and restore them to their original state when needed.

1.3. Definitions, acronyms and abbreviations

Stakeholder - is anyone who influences or interferes with product development

Full backup - this is the most complete type of backup, which clones all the selected data.

Incremental backup - will only store changes that were made since the previous backup was done.

CLI (command-line interface) - is a text-based user interface used to communicate with the system

Configuration file - A file that defines the parameters, options, settings and preferences applied to computer programs

1.4. References

- Github project repository
- Subject webpage

1.5. Overview of the remainder of the document

The second chapter describes the product, its features, and users who interact with the product.

The third chapter focuses on the exact specification of the individual functions of the system.

General description

2.1. Product perspective

The application will be a console program which will be distributed as a compiled executable file, but with open-source code so that the user can compile it himself.

The application will be able to backup and restore sites on the server where it was installed according to user-configured requirements with the options of a full or incremental backup, and remote storage.

2.2. Product functions

The main purpose of the application is to backup and store all data (diverse file types) from the remote server.

There will exist two types of backup:

- Full backup - makes a full backup
- Incremental backup - makes a copy only of files/folders that were modified/created/removed since last backup

There will exist two different backup storages:

- local - data will be stored on the local machine
- remote - stored on a remote server (remote server is only a remote storage location, commands are not started from there, only from local machines)

The application administrator will be able to configure options (per site) within a configuration file. See section 3 for more details. The application could be configured only by a user with root privileges and will run on a linux system.

2.3. User characteristics

Administrator - A user with root privileges that can modify the config file and run backups on selected sites. This is also the only user that can configure the application and make changes in the configuration file.

2.4. General constraints

For suitable and comfortable usage of the program root(sudo) privileges are needed as well as having the configuration file correctly set up.
There is a restriction of having a minimum of 1 full backup stored.

2.5. Assumptions and dependencies

Software will be developed as a CLI application in Java programming language.

Specific requirements

3.1. Functional requirements

3.1.1 The application must be a console program

3.1.2 User interaction with the program will be based on the CLI

3.1.3 The CLI of the application must have the following parameters and features

3.1.3.1 Output the list of configured sites with the date of the last backup

3.1.3.2 Output a list of backups for each site

3.1.3.3 Can modify local configuration file. After changing a local configuration file, the remote server should receive a notification from the local machine that the file has been changed

3.1.3.4 Be able to delete local configuration file. After deleting a local configuration file the remote server should be notified by the local machine that the file was delete

3.1.3.5 Be able to manually run a full or incremental backup on the local machine

3.1.3.6 Be able to restore a specific selected backup

3.1.4 Application will be distributed as a compiled executable file

3.1.5 The application must be open source

3.1.6 The application must be configurable using configuration files

3.1.7 The configuration file must have the following parameters

3.1.7.1 Id (site name)

3.1.7.2 List of included files

These files would be included in a backup. can contain folders or files, or files with wildcards

3.1.7.3 List of excluded files

These files will not be included in the backup and will be ignored by the application

3.1.7.4 Pre_backup script - optional, if anything is needed to prepare the website for backup it should be included in this script which will be run before each backup. For example, stopping the instance or dumping the db.

3.1.7.5 Post_backup script - optional, counterpart of 3.1.7.4 - if anything is needed to be executed after the backup, for example starting the instance, or the database, or cleaning up the db dump, it should be included in this script which will be run after each backup.

3.1.7.6 Pre_restore script - same as 3.1.7.4 but run before each restore.

3.1.7.7 Post_restore script - same as 3.1.7.5 but run after each restore.

3.1.7.8 List of categories of previous full backups to keep on remote storage server

These categories are the age of the backup set in days. The number of categories (intervals) depends on the user requirements and can be variable.

Example:

- Keep one 180 days old
- Keep one 30 days old
- Keep one 7 days old

That means to keep one backup that is *n* days old and one that is *n* days younger.

3.1.8 The application will be run with root rights

3.1.9 Backups will always be saved to a local drive and copied to a remote server

3.1.10 Backups must be able to be restored to their original location

3.1.11 It is sufficient to have 1 full copy on the local drive if required by the configuration

3.2. Non-functional requirements

3.2.1 The application must be simple to use

3.2.2 The application should run smoothly and efficiently, without any significant time loss

3.2.3 The application must be reasonably safe from malicious attacks

3.2.4 The application must be able to check the authorization and authentication of its users

3.2.5 The application should not save more backups than necessary

3.2.6 The application has to be developed exactly for the linux platform