Design and Architecture Document

Server Perfromance Monitoring And Improvement Tools

SPMIT – V1.0

Sachin Patil

Sachin\_patil.1982@yahoo.com

Table of Contents

[1. Document Information & Review History - 2 -](#_Toc32187447)

[2. Introduction - 3 -](#_Toc32187448)

[3. Specification - 3 -](#_Toc32187449)

[4. Architecture - 4 -](#_Toc32187450)

[1) Engine - 4 -](#_Toc32187451)

[2) UI - 5 -](#_Toc32187452)

[3) Reports - 5 -](#_Toc32187453)

# Document Information & Review History

|  |  |  |
| --- | --- | --- |
| Document Information | |  |
| *File Name* | SPMIT-Workflow | |
| *Author* | Sachin Patil | |
| *Post Date* | 09/02/2020 | |

|  |  |  |  |
| --- | --- | --- | --- |
| Revision History | |  |  |
| ***Author/s*** | **Date** | | **Notes** |
| Sachin Patil | 09/02/2020 | | First Draft |
|  |  | |  |

# Introduction

This project is an automation to Monitor and Improve the performance of the server based on the events capture on the server.

As per this automation, it captures the event, analyse it based on predefine criteria, notify and generate reports through which engineering team can automate further to improve the performance

# Specification

Infrastructure:-

* EC2 Instance 🡪 Windows Server 2016 Standard

Database:-

* SQL Server 2019 Enterprise Edition

Coding Platform:-

* Python
* Django
* HTML
* Celery

# Architecture

This automation contains 3 sections

1. Engine
2. UI
3. Reports

## Engine

It’s a heart of the automation, which will pull the data from Server, analyse, notify and generate reports. Here is the workflow diagram as follows.

* This is Daily Job with Python Code
* It pulls the data from System Events
* Save these events in txt format at common accessible location
* Code will further push this log in SQL DB Table “ServerLogs\_Step1\_Raw” for further processing
* Code will further apply cleansing mechanism to get the correct data and put into “ServerLogs\_Step2\_Clean” SQL Table
* Code will further filter records which contains give set of words and put into “ServerLogs\_Step2\_Final”

Save into .txt file

Pull data from System Events

Filter records which contains “users”, “assets”, uploads”, “admin”, “dashboard”

* Tokenize
* Probability
* Stop-words
* Remove Duplication
* Stemming

Push into SQL DB

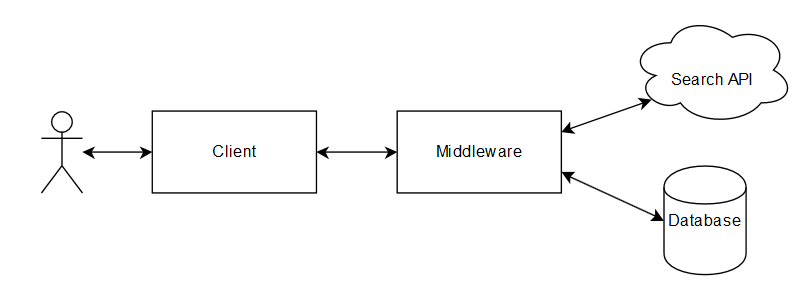
## UI

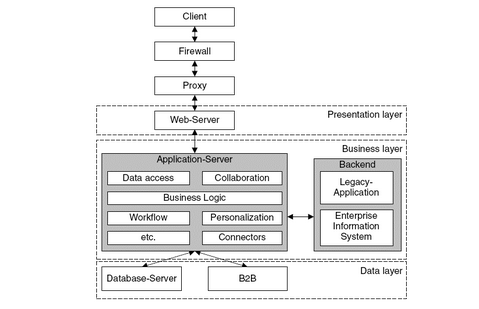
UI of this application can get the visualise effect of above workflow and user can trigger the engine work flow manually by uploading log files.

## Reports

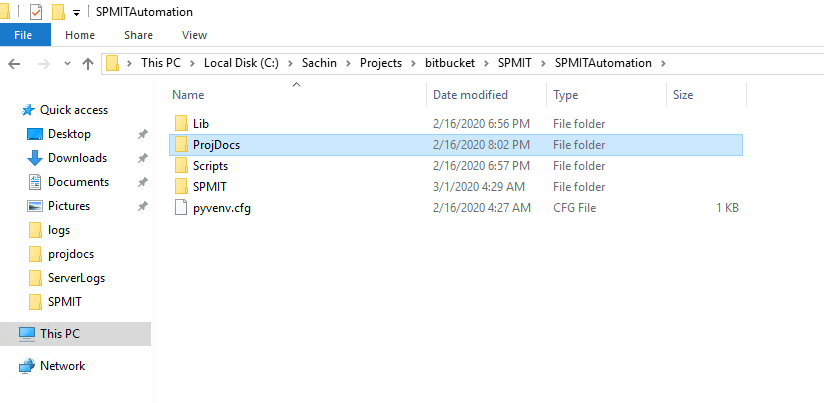
Reports of this application has been planned with SQL Reporting / Tableau depends on compatibility of these software’s with EC2 instance.

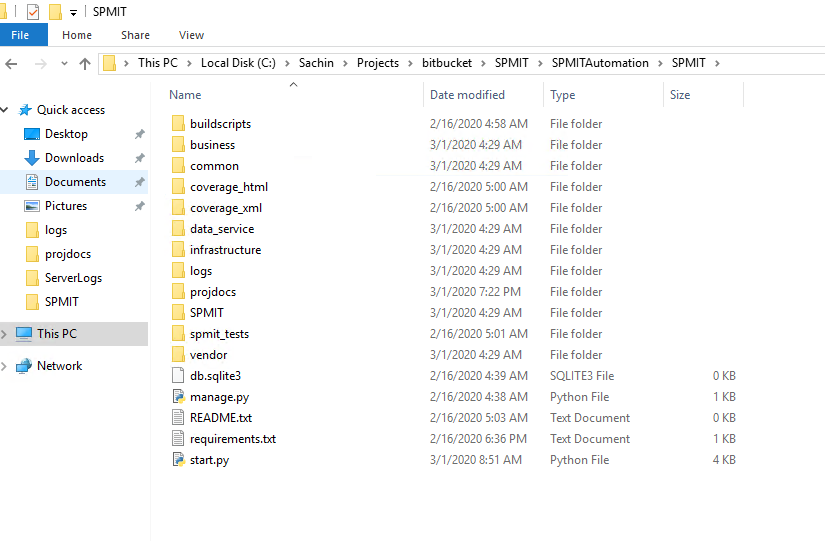
## Code Architecture





## Code Folder Design





## Database Design

## 