Architecture Design

Wine Review Analysis

Written By	Author 1
Document Version	0.1
Last Revised Date	

DOCUMENT CONTROL

Change Record:

VERSION	DATE	AUTHOR	COMMENTS
0.1	04- Apr - 2022	Author 1	

Reviews:

VERSION	DATE	REVIEWER	COMMENTS

Approval Status:

VERSION	REVIEW DATE	REVIEWED BY		APPROVED BY	COMMENTS

Contents

1. Introduction 4
1.2 Scope 4
2. Architecture 5
3. Deployment Description 6
3.1 Introduction 6
3.2 Prerequisites 6
3.3 Triggering your deployment 6
3.4 Using environments 6
3.5 Using concurrency 6
3.6 Viewing deployment history 6
3.7 Monitoring workflow runs 7

1. Introduction

1.1 What is Architecture design document?

Any software needs the architectural design to represents the design of software. IEEE defines architectural design as “the process of defining a collection of hardware and software components and their interfaces to establish the framework for the development of a computer system.” The software that is built for computer-based systems can exhibit one of these many architectures.

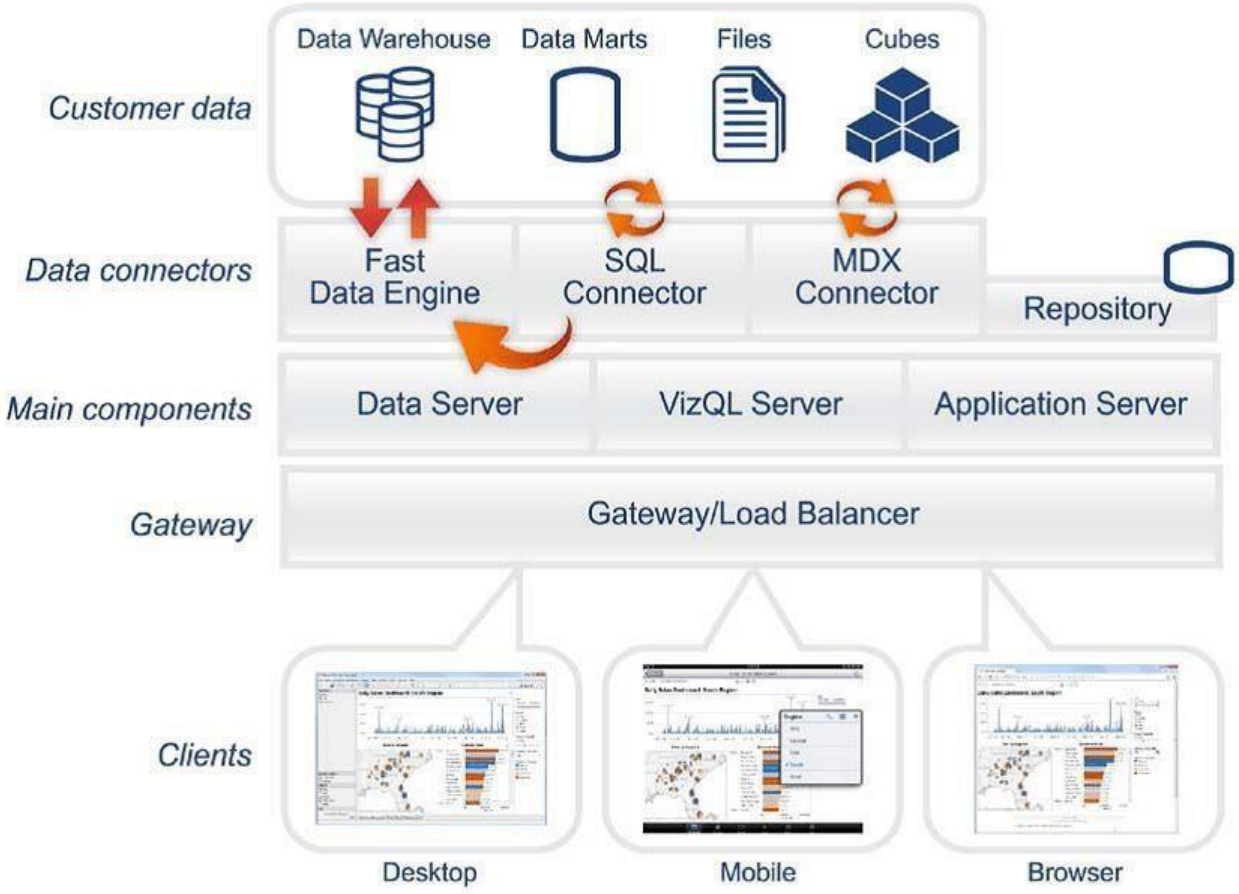
Each style will describe a system category that consists of :

- A set of components (eg: a database, computational modules) that will perform a function required by the system.
- The set of connectors will help in coordination, communication, and cooperation between the components.
- Conditions that how components can be integrated to form the system.
- Semantic models that help the designer to understand the overall properties of the system.

1.2 Scope

Architecture Design Document (ADD) is an architecture design process that follows a step-by-step refinement process. The process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall, the design principles may be defined during requirement analysis and then refined during architectural design work.

2. Architecture



3. Deployment Description

3.1 Introduction

GitHub Actions offers features that let you control deployments. You can:

- Trigger workflows with a variety of events.
- Configure environments to set rules before a job can proceed and to limit access to secrets.
- Use concurrency to control the number of deployments running at a time.

For more information about continuous deployment, see "About continuous deployment."

3.2 Prerequisites

You should be familiar with the syntax for GitHub Actions. For more information, see "Learn GitHub Actions."

3.3 Triggering your deployment

You can use a variety of events to trigger your deployment workflow. Some of the most common are: pull_request, push, and workflow_dispatch.
For example, a workflow with the following triggers runs whenever:

- There is a push to the main branch.
- A pull request targeting the main branch is opened, synchronized, or reopened.
- Someone manually triggers it.

```
on: push: branches: -
main pull_request:
branches: - main
workflow_dispatch:
```

For more information, see "Events that trigger workflows."

3.4 Using environments

Environments are used to describe a general deployment target like production, staging, or development. When a GitHub Actions workflow deploys to an environment, the environment is displayed on the main page of the repository. You can use environments to require approval for a job to proceed, restrict which branches can trigger a workflow, or limit access to secrets. For more information about creating environments, see "Using environments for deployment."

3.5 Using concurrency

Concurrency ensures that only a single job or workflow using the same concurrency group will run at a time. You can use concurrency so that an environment has a maximum of one deployment in progress and one deployment pending at a time.

3.6 Viewing deployment history

When a GitHub Actions workflow deploys to an environment, the environment is displayed on the main page of the repository. For more information about viewing deployments to environments, see "Viewing deployment history."

3.7 Monitoring workflow runs

Every workflow run generates a real-time graph that illustrates the run progress. You can use this graph to monitor and debug deployments. For more information see, "Using the visualization graph."

You can also view the logs of each workflow run and the history of workflow runs. For more information, see "Viewing workflow run history."