Architecture Design

# Movielens

|  |  |
| --- | --- |
| **Written By** | Abhishek |
| **Document Version** | ` |
| **`Revised Date** | 12.08.2022 |

**DOCUMENT CONTROL**

## Change Record:

|  |  |  |  |
| --- | --- | --- | --- |
| **VERSION** | **DATE** | **AUTHOR** | **COMMENTS** |
| 0.1 | 01.06.2022 | Abhishek | Introduction and architecture defined |
| 0.2 | 06.06.2022 | ` | Architecture & Architecture description appended and  Updated. |
|  |  |  |  |
|  |  |  |  |

**Reviews:**

|  |  |  |  |
| --- | --- | --- | --- |
| **VERSION** | **DATE** | **REVIEWER** | **COMMENTS** |
|  |  |  |  |

**Approval Status:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **VERSION** | **REVIEW**  **DATE** | **REVIEWED BY** |  | **APPROVED BY** | **COMMENTS** |
|  |  |  |  |  |  |

# Contents

1. [Introduction 04](#_TOC_250005)
   1. [What is Architecture Design Document? 04](#_TOC_250004)
   2. [Scope 04](#_TOC_250003)
2. [Architecture 05](#_TOC_250002)
   1. Jupyter Architecture 05
   2. Jupyter Server Architecture 05
   3. Gateway/Load Balancer 06
   4. Application Server 06
   5. DB Server 07
   6. Data Engine 07
   7. Backgrounder 07
   8. Data Server 07
   9. Mechanism Flow 07
3. Deployment 08
   1. [Deployment Options in Heroku 09](#_TOC_250001)
   2. [Single Node Architecture 10](#_TOC_250000)
   3. Three Node Architecture 11
   4. Five Node Architecture 12

# Introduction

## 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.

## 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.

# Architecture



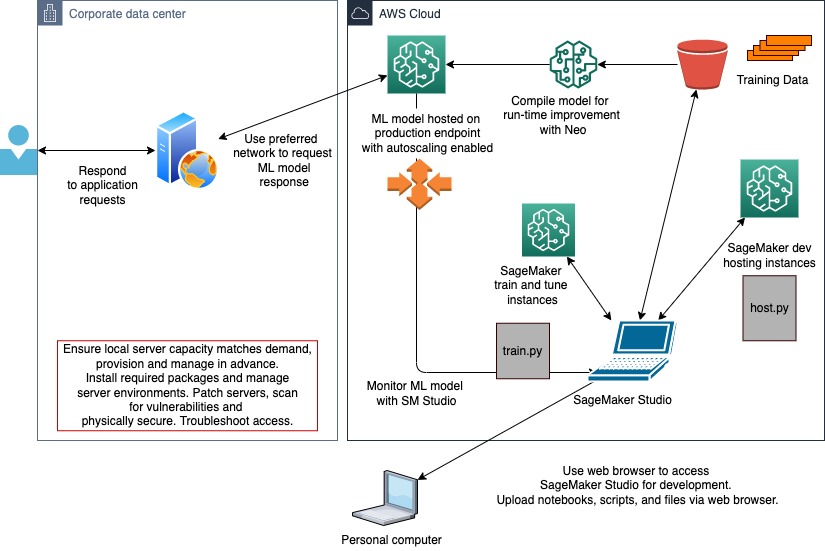
**` Architecture**

Scala ide has a highly scalable, architecture that serves creating models for clients, web clients and desktop-installed software. Scala eclipse architecture supports fast and flexible deployments in various platforms

### ARCHITECTURE DESIGN

**6**

The following diagram shows several Servers’ architecture:



Jupyter is internally managed by the various kernel processes for modeling.

**1. Gateway/Load Balancer**

It acts as an Entry gate to the ` Server and also balances the load to the Server if multiple Processes or kernel are configured.

**2) Application Server:-**

Application Server processes (wgserver.exe) handle browsing and permissions for the specific Server web and mobile interfaces. When a user opens a view in a client device, that user starts a session on T` Server. This means that an Application Server thread starts and checks the permissions for that user and that view.

1. **Repository:-**

Here Server Repository is a `data strax database that stores server data. This data includes information about various Server users, groups and group assignments, permissions, projects, data sources, and extract metadata and refresh information.

1. **Intra Server:-**

Once a view is opened, the client sends a request to the process `. The DB process then sends queries directly to the data source, returning a result set that is rendered as objects and presented to the user. Each Server has its own cache that can be shared across multiple users

1. **Data Engine:-**

It Stores data extracts and answers queries.

1. **Backgrounder:-**

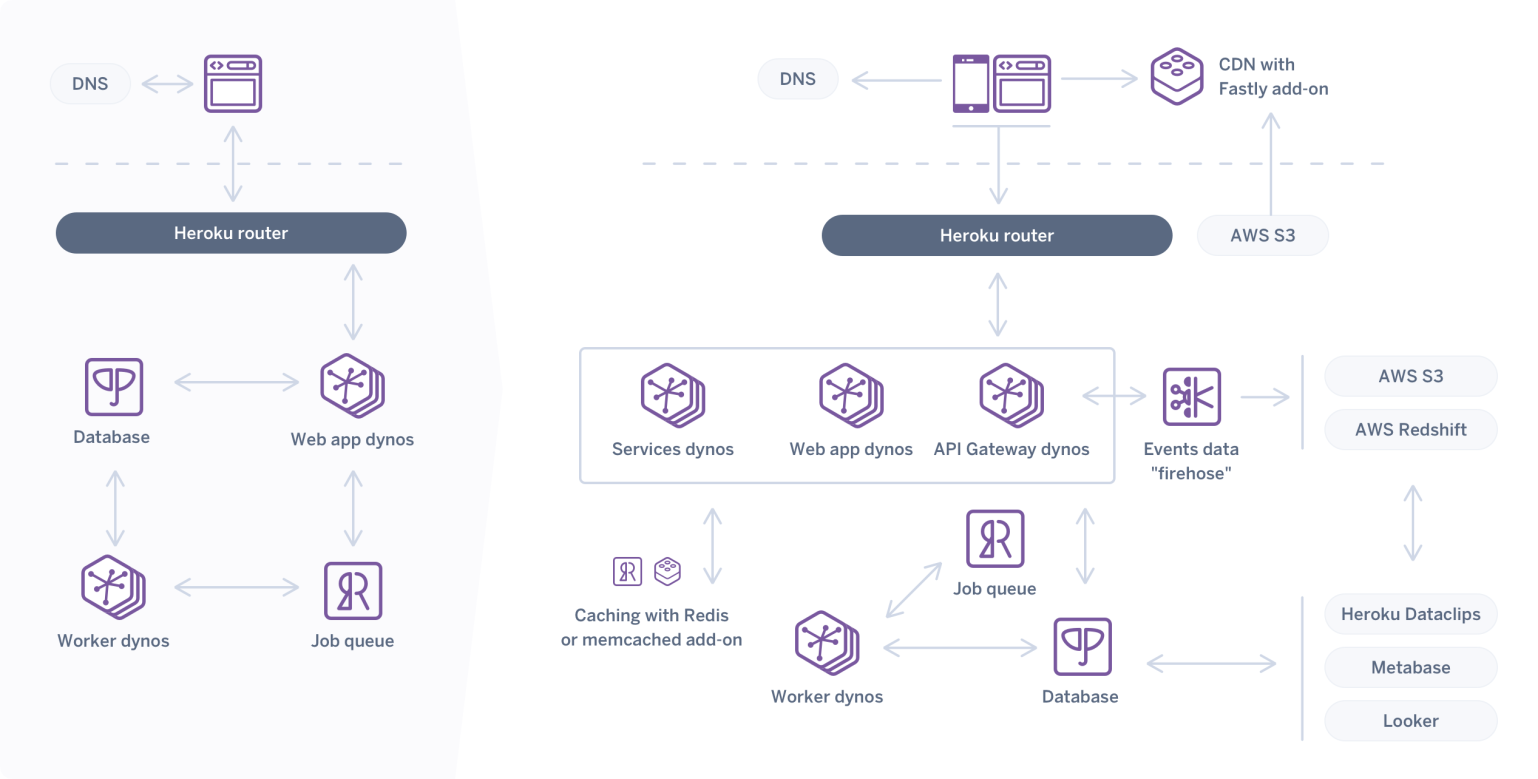
The backgrounder Executes server task which includes refreshes scheduled extracts, tasks initiated from bases and manages other background tasks.

1. **Data Server:-**

Data Server Manages connections to several Server data sources

It also maintains metadata from ` Desktop, such as measurements, definitions, and groups.

**8) DB Flow**

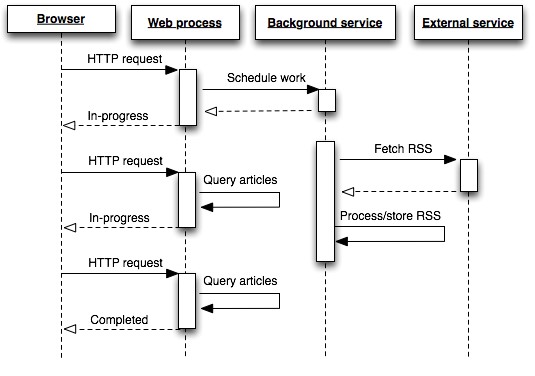


# Deployment Description

## Deployment options in Heroku

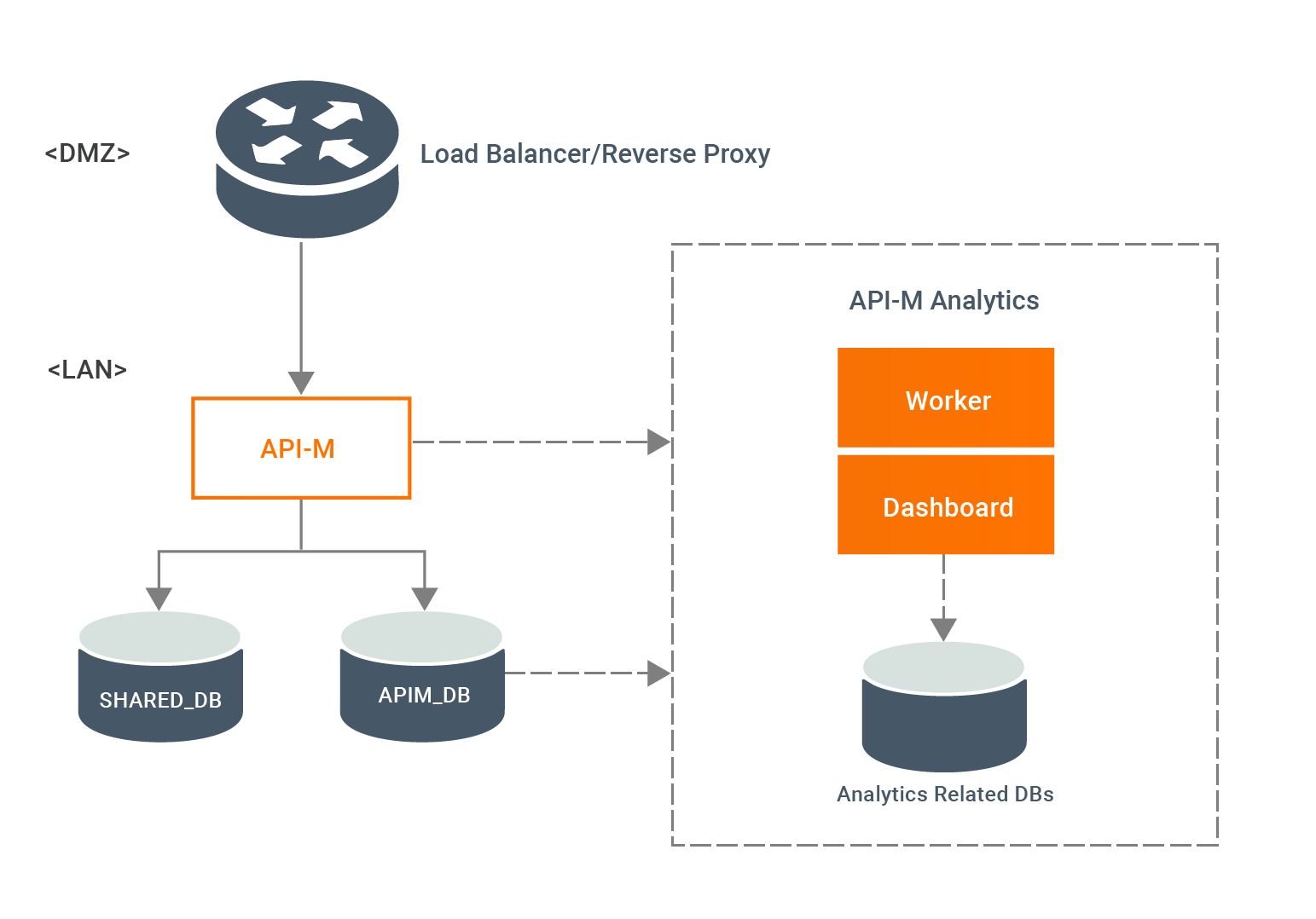
Cloud analytics platform offers several different deployment options depending on ours

environment and needs. The below graphic shows each option at a glance:



1. **Heroku** up and running quickly with no hardware required`, fully hosted by these so all upgrades and maintenance are automatically managed for you.
2. **Deployment Server** deployed on public cloud: Leverage the flexibility and scalability of cloud infrastructure without giving up control. Deploy to Amazon Web Services, Google Cloud Platform, or Microsoft Azure infrastructure to quickly get started with these Server (on ours choice of Windows or Linux). Bring our own license or purchase on ours preferred marketplace.
3. **` Server deployed on-premises**: Manage and scale our own hardware and software (whether Windows or Linux) as needed. Customize our deployment as we see fit.

## Single Node Architecture



This architecture is a single node architecture. This is the most simple deployment topology.

## 3 Node Architecture



This architecture is a 3 Node Architecture which is more capable to handle concurrent requests.

If we need failover or high availability, or want a second instance of the repository, we must install specific Server on a cluster of at several ` computers. In a cluster that includes at various ` nodes, we can configure these instances of the repository, which gives our cluster failover capability.

## 5 Node Architecture



When we install ` Server on a Five-node cluster, we can install server processes on one or both nodes. A five-node cluster can improve the performance of this Server, because the work is spread across multiple machines.

Note the following about five-node clusters:

* + - A five-node cluster does not provide failover or support for high availability.
    - We can't install more than one instance of the repository on a two-node cluster, and the repository must be on the initial node.