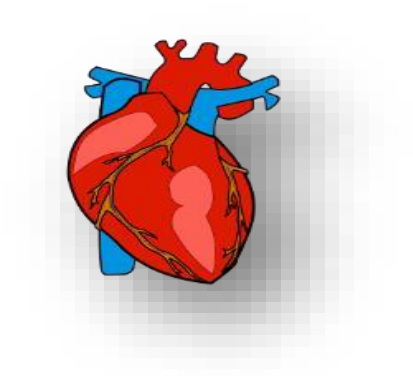
Architecture Design

Heart Disease Diagnostic Analysis



Revision Number - 1.1

Last Date of Revision - 25/01/2022

SULTAN SHAIKH

Document Control

Date	Version	Description	Author
23/01/2022	1.0	Introduction, Architecture, Deployment	Sultan Shaikh
25/01/2022	1.1	Final Revision	Sultan Shaikh

Contents

1. Introduction

1.1 What is Architecture Design Document?

Any software needs the architectural design to represent the design of the 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 help the designer to understand the overall properties of the system.

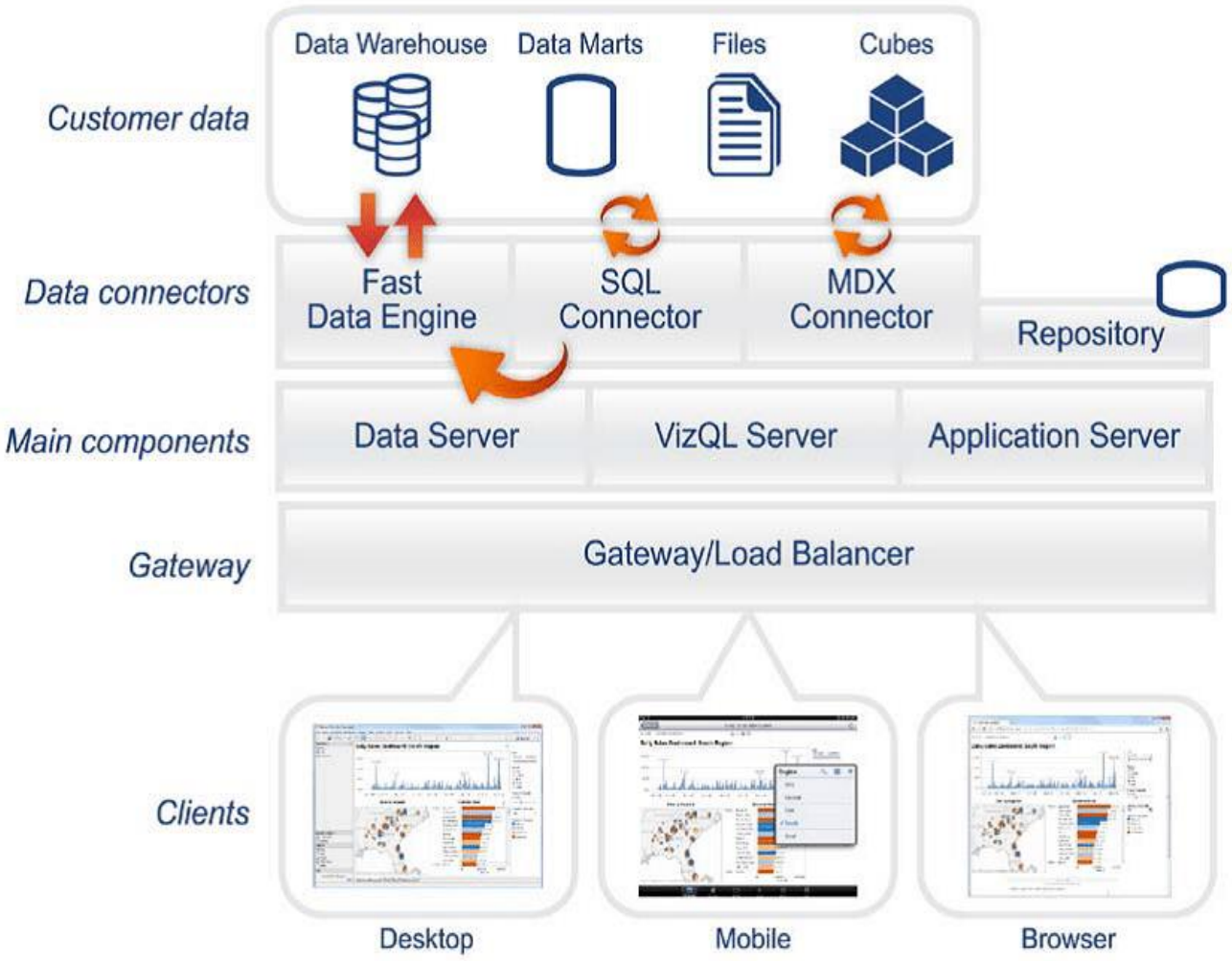
1.2 What is Scope?

Architecture Design Document (ADD) is an architectural design process that follows a step-by-step refinement process. The process can be used for designingdata 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

2.1 Tableau Server Architecture

Tableau has a highly scalable, n-tier client-server architecture that serves mobile clients, Web clients and desktop-installed software. Tableau Server architecture supports fast and flexible deployments.



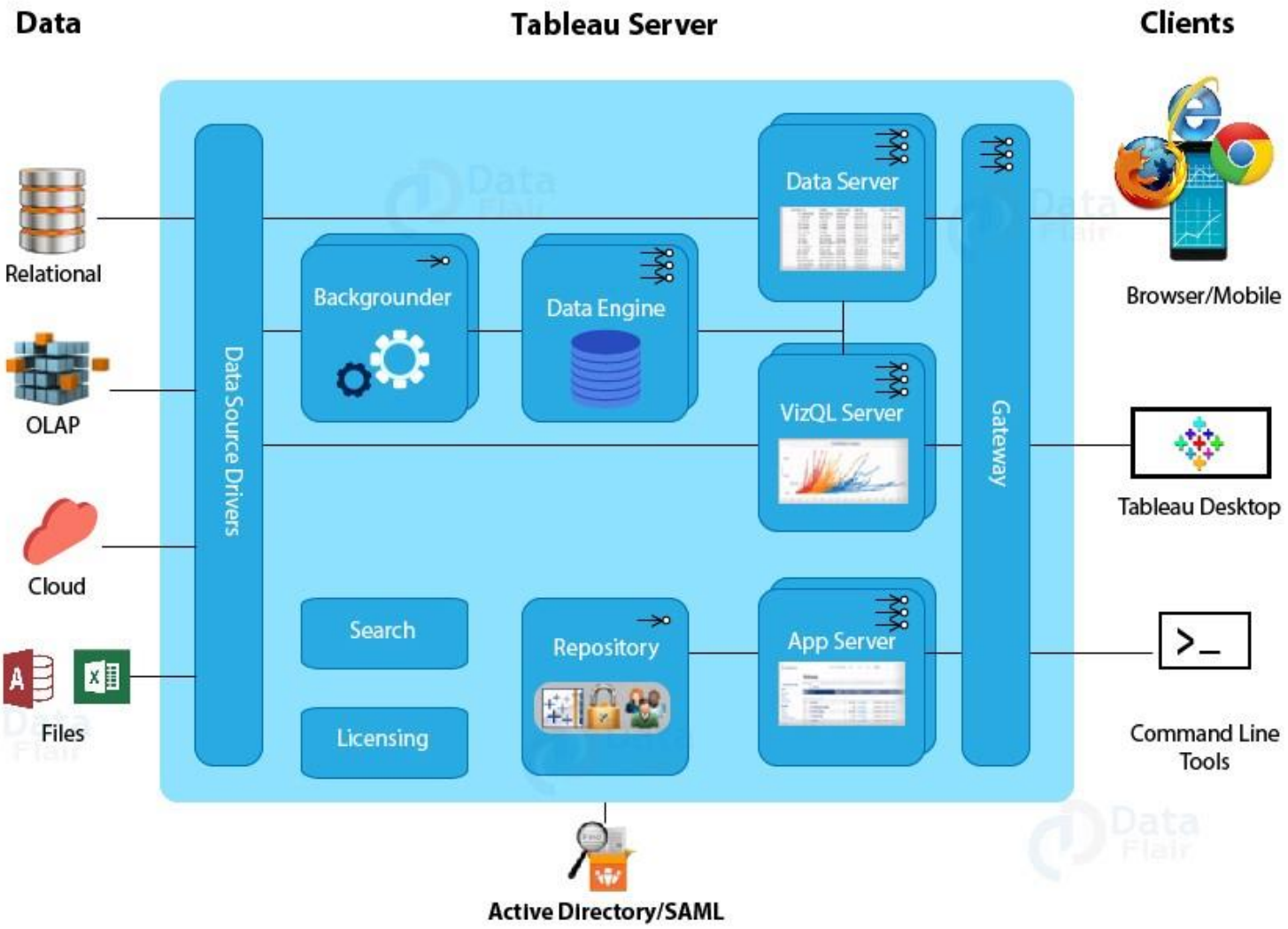


Tableau Server is internally managed by multiple server processes.
1. Gateway/Load Balancer
It acts as an Entry gate to the Tableau Server and also balances the load to the Server If multiple processes are configured.

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

3. Repository
Tableau Server Repository is a PostgreSQL database that stores server data. This data Includes information about Tableau Server users, groups and group assignments, permissions, projects,
data sources, and extract metadata and refresh information.

4. VIZQL Server:-
Once a view is opened, the client sends a request to the VizQL process. The VizQL process

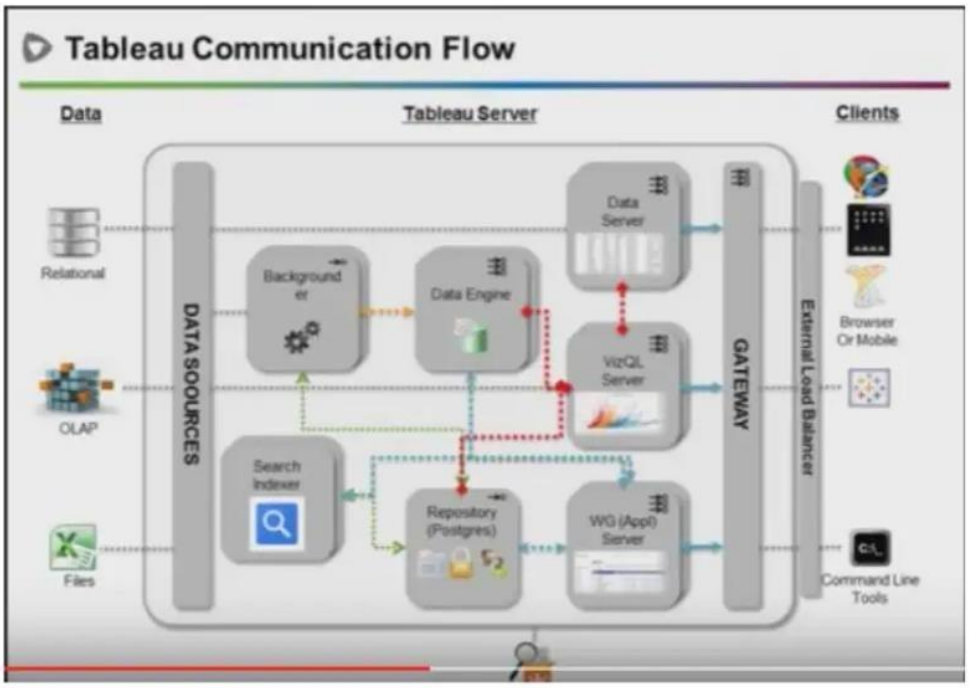
then sends queries directly to the data source, returning a result set that is rendered as images and presented to the user. Each VizQL Server has its own cache that can be shared across multiple users

5. Data Engine:-
It Stores data extracts and answers queries.

6. Backgrounder:-
The backgrounder Executes server tasks which includes refreshes scheduled extracts, tasks initiated from tabcmd and manages other background tasks.

7. Data Server:-
Data Server Manages connections to Tableau Server data sources.It also maintains metadata from Tableau Desktop, such as calculations, definitions, and groups.

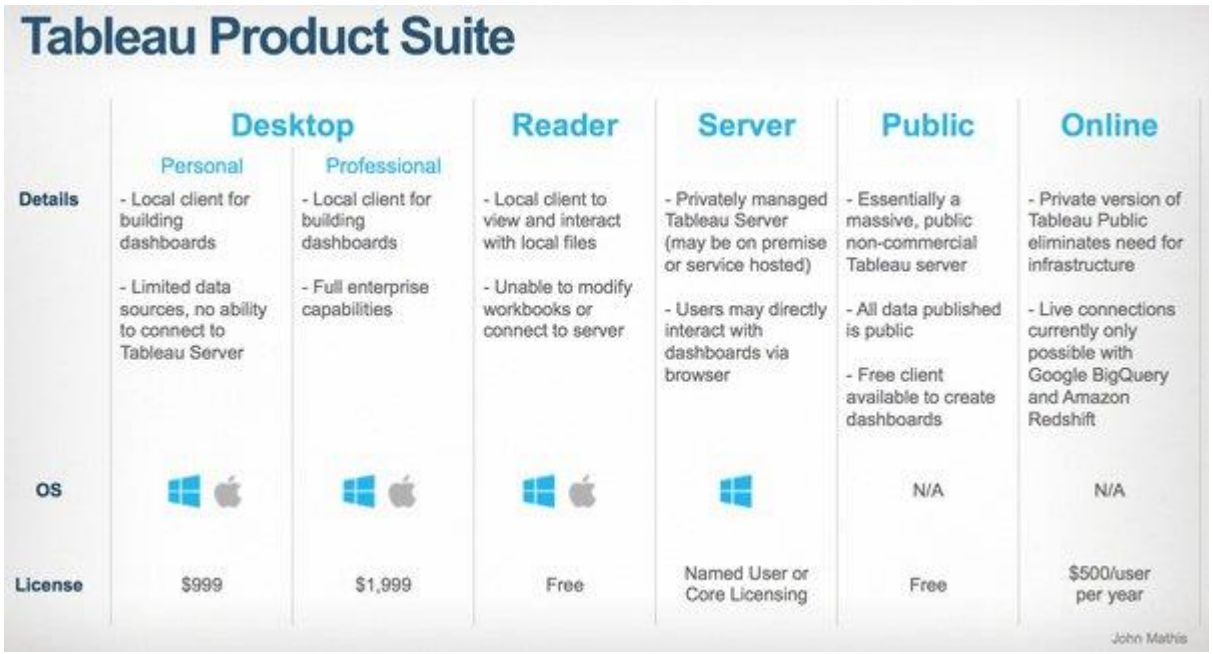
8. Tableau Communication Flow



3. Deployment

3.1 Deployment Options

The deployment process lets you clone content from one stage in the pipeline to another, typically from development to test, and from test to production. Tableau's analytics platform offers three different deployment options depending on your environment and needs. The below graphic shows each option at a glance

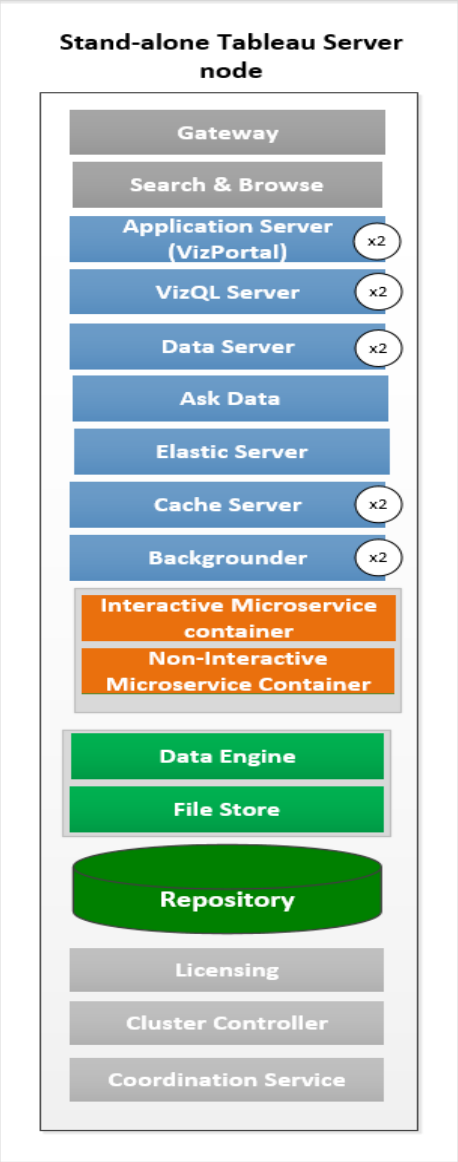


1. Tableau Online Get up and running quickly with no hardware required. Tableau Online is fully hosted by Tableau so all upgrades and maintenance are automatically managed for you.

2. Tableau 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 Tableau Server (on your choice of Windows or Linux). Bring your own license or purchase on your preferred marketplace.

3. Tableau Server deployed on-premises: Manage and scale your own hardware and software (whether Windows or Linux) as needed. Customize your deployment as you see fit.

3.2 Single Node Architecture



This architecture is a single node architecture. This is the simplest deployment topology. This type of installation is reasonable for testing, running trials, and for environmentsthat can handle occasional downtime and system availability due to lack of redundancy. All server processes are running on a single machine. There is less redundancy and fewersafeguards in the event of a problem with one of the server processes. You also need to make sure the computer you install Tableau Server on has adequate resources to handle the processes and the demands of users and data.

3.3 Two Node Architecture

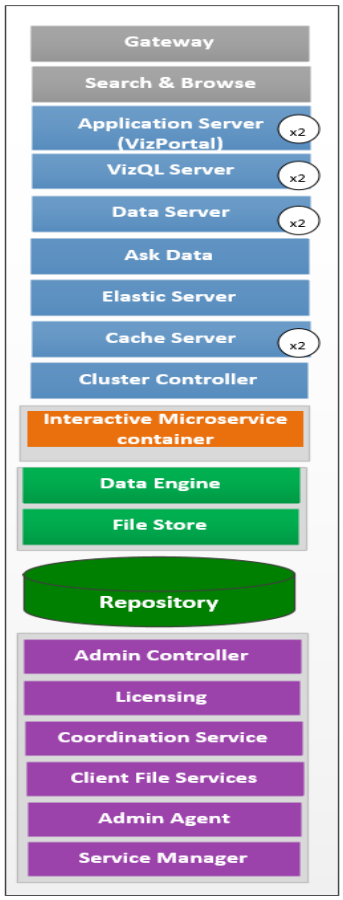
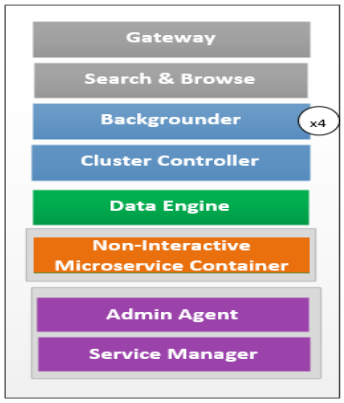


Tableau Server Initial Node

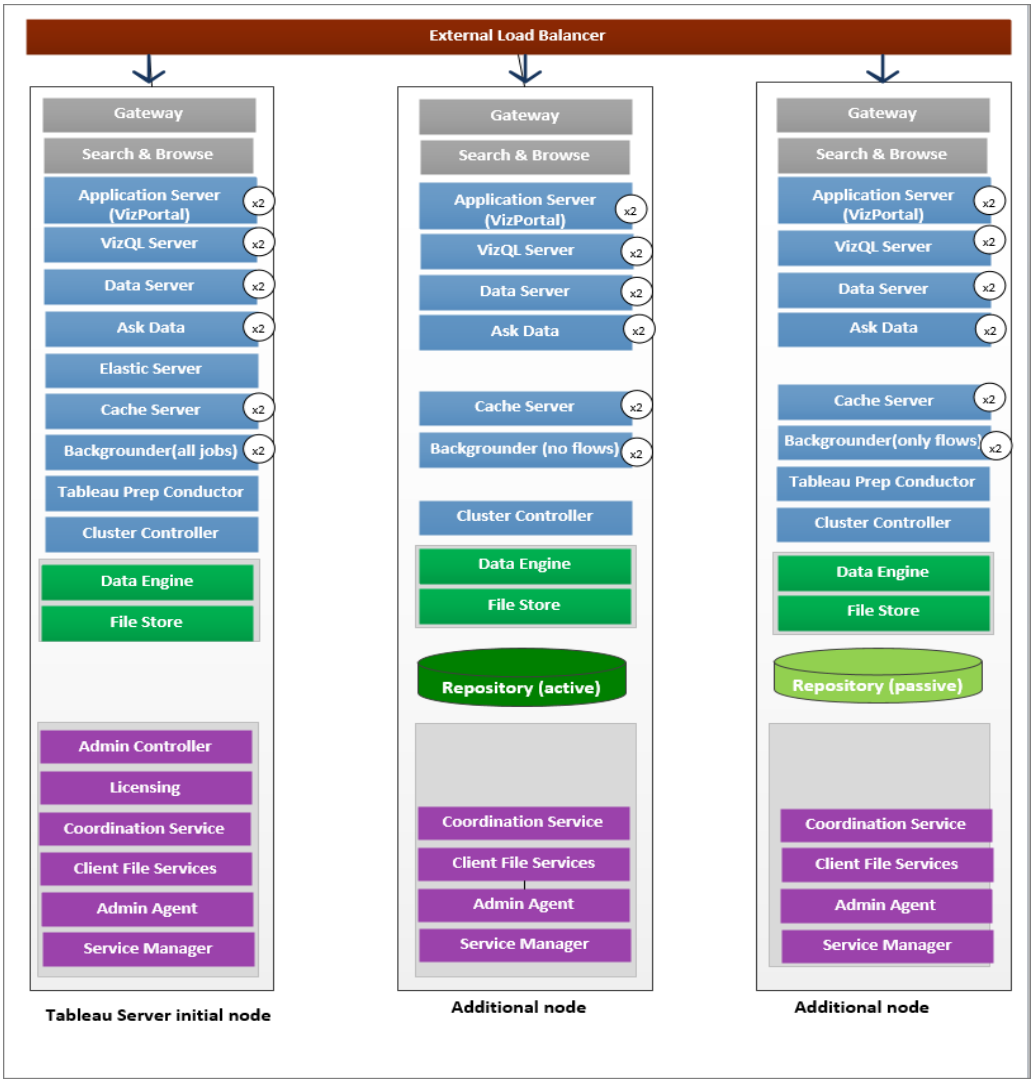


Additional Node

If we need failover or high availability or want a second instance of the repository, install Tableau Server on a cluster of at least three computers. In a cluster that includes at leastthree nodes, you can configure two instances of the repository, which gives our cluster failovercapability.

- Extract heavy environment
- Frequent extract refreshes

3.4 Five Node Architecture (Highly Available)



An HA installation of Tableau Server is a special type of multi-node installation with a minimum of three nodes and multiple instances of key processes (the Repository, FileStore/Data Engine (Hyper), Coordination Service, and Client File Service) on different computers. With an HA installation, there is built-in redundancy of those key processes, including multiple File Stores, and automatic Repository failover. The goal is to minimizesystem downtime by eliminating single points of failure and enabling detection of failureswith failover where possible.

Downtime is still possible, in the event of an initial node failure. Dashboards and views may load more slowly than expected, and timeouts are possible, depending on how your the system isconfigured and being used.