

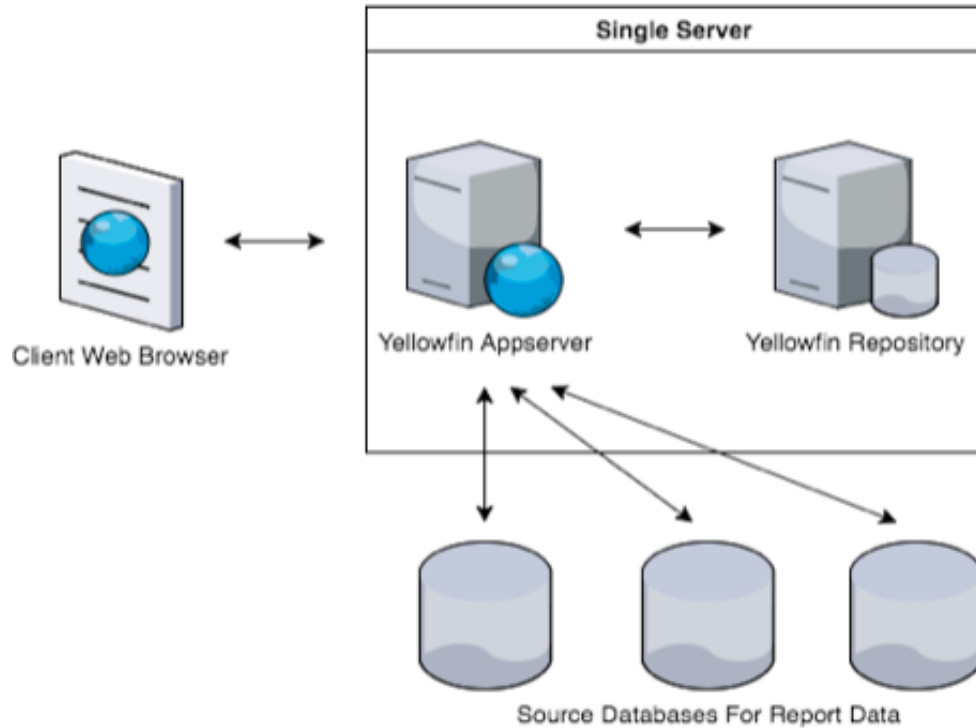
# Server Configuration

Yellowfin can be installed in different configurations based on the specific needs and projected user's volumes. Listed below are three possible configurations and the scenarios in which you may choose to use them. These three examples are:

- Single Server Installation
- Separated Application and Database Server
- Multi Server Clustered Installation

## Single Server Installation

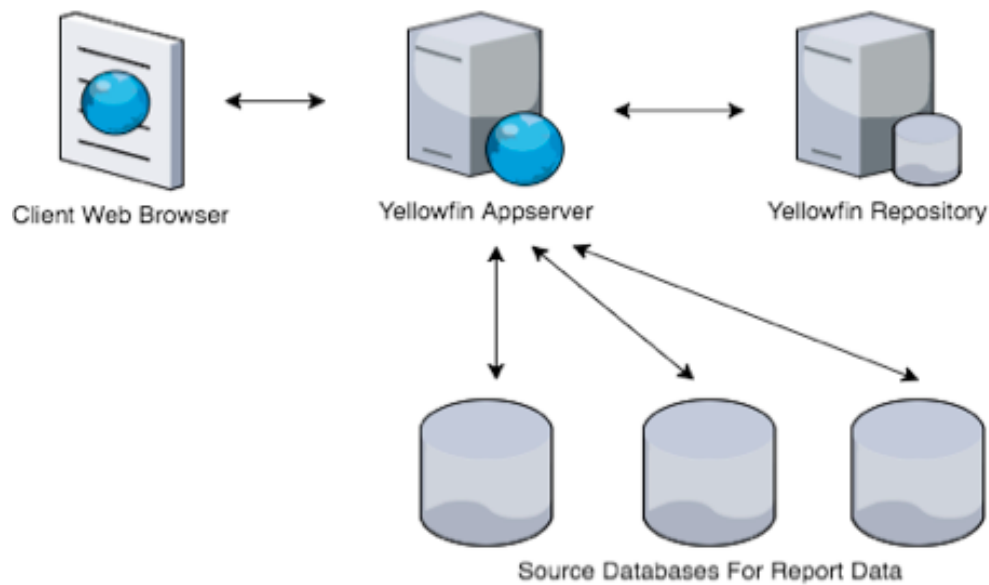
This layout has the Yellowfin application server and the repository database co-existing on the same server. The reporting source may also reside in the same DBMS as the repository.



This is an ideal implementation for a small number of concurrent users. It minimizes hardware cost and installation effort to allow a rapid implementation of Yellowfin. This configuration removes any network latency between the Yellowfin application and the database server, but both systems share system resources.

## Separate Application and Database Server Installation

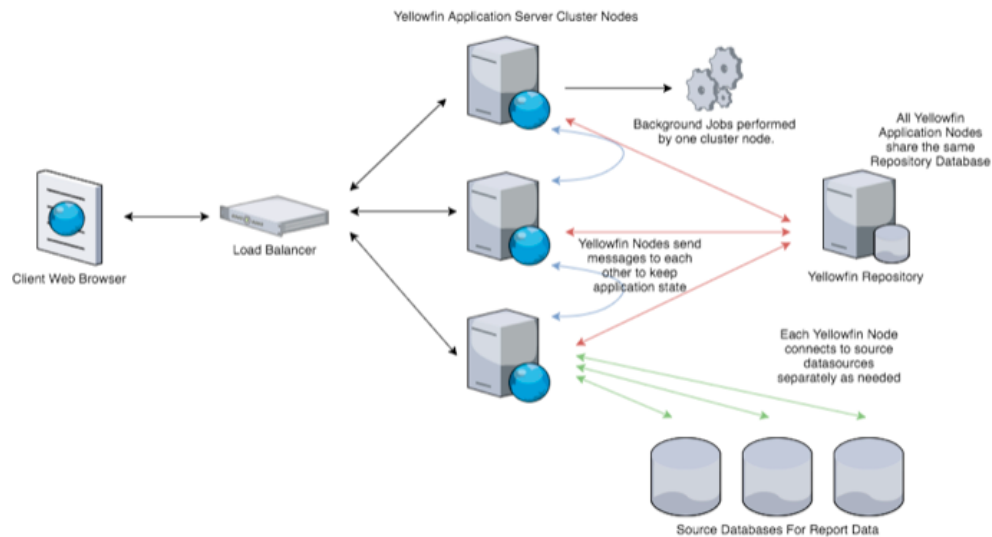
Separating the Yellowfin configuration database and the Yellowfin application server will provide a more scalable option to support larger user numbers. It also provides better capability to tune the reporting server and database management independently. This is the recommended configuration for single node Yellowfin configurations.



This configuration does introduce network latency between the Yellowfin application server and repository server, but does allow for an entire system to be dedicated to each component. Clustering from this configuration is also simple. Additional application server nodes can be connected to the repository for greater capacity.

## Cluster Installation

For large numbers of concurrent users and high availability, Yellowfin can be deployed as a cluster.



Each application server node added to the cluster increases capacity for supporting additional users. Processing load is distributed across available nodes.

A Load Balancer is responsible for distributing incoming requests to Yellowfin nodes. This could be a hardware appliance or software application (such as HAProxy, Apache Webserver, or IIS).

Yellowfin nodes are “cluster aware” and will send messages to each other to keep the application synchronized. This functionality needs to be turned on. Configuration options for enabling clustering is documented in the [Yellowfin Clustering Guide](#).

Yellowfin nodes share a single repository database, and thus this should also be scaled to support the load of the cluster. It is recommended that a clustered database system also be used to provide additional capacity for both the Yellowfin repository and reporting data sources.

## Server Placement Considerations

The geographic spread of users and the resulting placement of the various Yellowfin servers for optimal performance are of particular interest during deployments. This section discusses key considerations with respect to the placement of the various Yellowfin components and interaction with the end user client.

## Application & Yellowfin Database

The frequent interactions between the application server and the database server (metadata repository) mandate that they have close proximity to each other. A dedicated high speed, high bandwidth network greatly increases performance due to reduced network latency.

Depending on the capabilities on the database server, it may be possible to run a distributed repository database across multiple regions. However, it is important that this database retains transactional integrity.

## Application Server and Data Sources

The same reasons cited for the relative placement of application server and the database server apply to the placement of the organizational data sources (transactional databases or data warehouse). While the application server and data sources can be spread out geographically, a dedicated high speed, high bandwidth network greatly increases performance.

Generally, Yellowfin data sources are read-only, and it is possible to run queries on read-only replicas. Yellowfin has no built-in capabilities for distributing reporting queries across multiple data source nodes, however, many JDBC drivers have clustering features built in.

## Application Server and Web Server

The application server takes the role of a web server in most deployment scenarios. The web server is the component of the software that delivers static content. A web server can be configured to deliver static content from Yellowfin, which would free up resources on the application server.

When a web server is used, it is usually configured to deliver static content, while requests for dynamic content are passed through to the application server. For this reason, it is recommended that the web server and application server(s) exist on the same high-speed network.

## Web Servers Relative to Web Users

In general, the network bandwidth between the web server and web browser is the most unreliable and least controllable by the organization. For this reason, when users are highly distributed and high bandwidth networks do not exist, it is common to place web servers closer to remote users. However, once the static content is initially cached by a user's browser, most requests generated by a user will be dynamic requests for the application server.

**Previous topic:** [Estimating capacity requirements](#)

**Next topic:** [Scalability testing](#)