Production Deployment

This section of the documentation covers the key concepts Ed-Fi Dashboard implementations should understand before deploying to a production environment, along with details about specific configurations.

Content

Production System Components

The Ed-Fi Dashboard websites and multiple databases work together to provide primary and supporting functions for a production instance. An overview follows:

Websites

- **Ed-Fi Dashboards.** The main Ed-Fi Dashboards site. Access to the configuration section is role based.
- **Security Token Service (STS)**. Handles authentication for the Ed-Fi Dashboards website.
- **Metric Metadata Configuration Utility**. Assists in generating configuration data for any custom metrics you want the ETL to calculate. Optional.

Databases

- **App.** A database that stores configuration data for the Ed-Fi Dashboards.
- **Dashboards.** A database containing metric calculation data for the Dashboards.
- **DW.** A database containing historical snapshot of dashboard metrics.
- **ETL Log.** A database containing exception details and logging information.

Configuration

This section describes essential configuration settings.

Database Connection Strings

Database connection strings will need to point to the correct databases in Web.config for all three applications: App (EdFi.Dashboards.Presentation.Web), STS (EdFi.Dashboards.SecurityTokenService.Web), and Metric Metadata Configuration Utility (MetricsMetadataUtility.Web) websites. For detailed information see Dashboard Connection Strings.

App

Configuration of authentication and authorization is the most important part of Dashboard UI configuration, followed by caching and machine key settings (useful in load-balanced scenarios).

STS

Configuration settings in the STS website will allow the addition of one or more LDAP directories for authentication. Additionally, SSL certificates used for signing and encrypting data can be configured here. For detailed information see UI Developers' Guide - Security Overview.

Planning for a Secure Production Deployment

The security of student information is a primary concern on any production system. This section outlines considerations for platform hosts when planning Ed-Fi Dashboard deployments.

The information in this section can be used as a checklist or as input into a threat modeling exercise during the deployment planning cycle.

Value Assets

A list of high-value assets and a brief description of each follows:

Asset	Description
Ed-Fi Operational Data Store	The operational data store contains student, parent, and staff personally identifiable information data, along with potentially sensitive financial information (e.g., employee payroll, budgets). The ODS is the source of the data for your Dashboard deployment.

Ed-Fi Dashboard Data Store	The Dashboard Data Store contains the results of metric calculations run on your ODS. Like the ODS, it also contains student, parent, and staff personally identifiable information.
Ed-Fi Dashboard Data Warehouse	The Dashboard Data Warehouse capture historical data from the Ed-Fi Dashboard Data Store, and there contains the same personally identifiable information.
Ed-Fi Dashboard Web Services	This includes both the Dashboard UI and STS websites, and (optionally) the Metric Metadata Configuration Utility. Staff, Administrators and other end-users of these systems expect them to be available.
ETL log data	Usually logged to the EtlLogDb (though configurable to log elsewhere, e.g. to files), this data is useful in troubleshooting problems during metric calculation. This data *may* contain information about server configuration, schools, students, etc.
Personally Identifiable Information	Information such as student names, demographic information, academic performance, and disciplinary records can be valuable to attackers – and disclosure of this information is regulated by laws such as the Family Educational Rights and Privacy Act (FERPA).

Security Recommendations for Production Deployments

The following are recommendations and precautions for implementers to consider when planning a production deployment.

An obvious disclaimer applies: these are just general guidelines offered in summary form. Dashboard hosts should include trained security professionals in their deployment planning and conduct security audits prior to deployment — and periodically thereafter.

Recommendations for Dashboard Databases

- Ensure applications have the least privilege access to the underlying databases.
- Encrypt database storage.
- Ensure databases do not accept external connections.

Recommendations for the Dashboard UI

- Restrict access to the website at the network / IP level.
- Encrypt connection strings in configuration files.
- Allow only HTTPS connections.
- Set “Persist Security Info” to false in the database connection string.
- Configure reasonably short session timeouts.
- Ensure cookies require SSL.

Recommendations for the Security Token Service Website

- Restrict access to the website at the network / IP level.
- Ensure applications have the least privilege access to the underlying databases.
- Encrypt connection strings in configuration files.
- Allow only HTTPS connections.
- Set “Persist Security Info” to false in the database connection string.
- Configure reasonably short token timeouts.

Recommendations for the Metric Configuration Utility

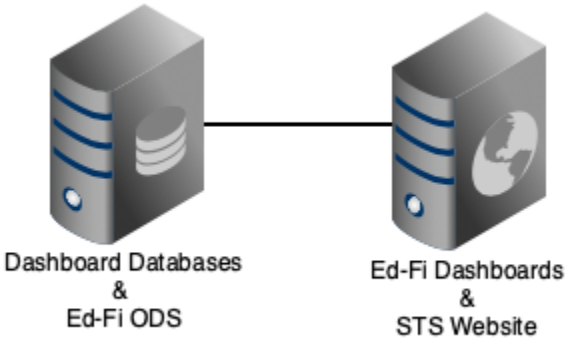
- Restrict access to the website at the network / IP level.

Reference Models for Production Deployment

The following sections provide a number of archetypical deployment models. These reference models are not intended to be a complete prescription for any given installation, but rather a starting point to weigh options and plan a deployment that serves your organization’s needs.

On-Premises, Two-Server Deployment Model

Small deployments can function on a simple, two-server model. One server hosts the Dashboard and STS websites, and is responsible for running the ETL process. The other hosts the SQL Server platform, ODS data store and Dashboard-related databases.



This configuration is excellent for small organizations because it is inexpensive, easy to maintain, and leverages common technologies that are usually supportable by in-house staff. It allows the database server to remain in the internal network while the web server is placed in the DMZ. When properly configured, this approach ensures that even in the event that a server in the DMZ is compromised, that the student data would remain secure. This model may also be used for a cloud-based or hybrid deployment.

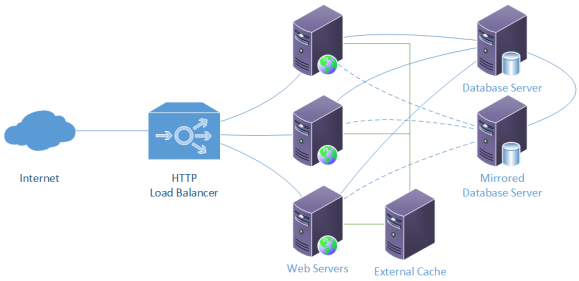
Load-Balanced Deployment Model

The Ed-Fi Dashboard is built to be part of a load-balanced solution. The API does not use server state information, so multiple physical servers may be installed as part of a cloud-based or on-premises load-balanced solution.

A load-balanced deployment, whether on-site or in the cloud includes an HTTP Load Balancer which analyzes incoming network traffic, multiple web servers, an external cache, and mirrored database servers. In this example, a basic load balancing scenario is described. More extreme examples of fault tolerance such as geographic redundancy may be useful but are not described herein.

The load balancer does exactly what its name implies, by routing incoming requests as efficiently as possible to the individual web servers, providing additional capacity as well as redundancy.

Mirrored database servers do not provide any performance or scalability improvements, but they do provide redundancy.



Multiple web servers with identical configurations are deployed and registered with the load balancer. Database configurations and connection strings are altered to implement automatic failover (see, e.g., https://technet.microsoft.com/en-us/library/cc917713.aspx#ECAA for details).

Azure Cloud-Based Deployment Model

Azure deployments include a redundant web server for failover, although only one is active at a time. Deploying to the backup server and then swapping the primary and backup allow for higher availability and automatic failover. Performance is limited to the capabilities of one server.

Load balancing is an optional feature that is enabled when multiple servers are configured using the Microsoft Azure management console. Azure provides a load balancer as part of their platform under a multiple-server configuration.

AWS Cloud-Based Deployment Model

Amazon Web Services, while providing their own platform as a service offerings, has not been targeted specifically. However, automatically deploying identical servers and load balancing among them is a valuable feature that requires additional hardware and configuration in an on-site deployment configuration.

As noted above, these deployment reference models are simply starting points for planning. The section that follows discusses several techniques that may be somewhere between useful and necessary in your production environment.

Environmental Considerations

Every production deployment has its own environmental considerations that are unique. The following items, while not unique to the Ed-Fi Dashboard, should be considered when planning for a production deployment.

Windows Domain and Service Accounts

In any nontrivial deployment (i.e., more than a few servers), it is recommended that the IIS Servers be members of a DMZ domain. One of the key benefits of having a DMZ domain is that domain service accounts can be used for all credentials between services in the domain. Windows Authentication is more secure, and less brittle than storing usernames and passwords using clear text or encrypted configuration files.

IIS Load Balancing

Multiple IIS servers are needed for load balancing and horizontal scaling. Provisioning multiple servers should be performed in conjunction with automated deployment scripts to minimize the potential for differences in configuration.

Microsoft SQL Server

It is highly recommended that **Windows Authentication** be used instead of SQL authentication, and that SQL Server authentication be disabled in SQL Server. Running each IIS application domain as a windows service account, and providing the appropriate permissions to those service accounts in SQL Server, allows the connection strings to contain no user names and passwords. It is recommended that SQL Servers be mirrored and that connection strings settings include failover settings. Best practices always include good database backup policies.

Pre-Deployment Tasks

The Dashboard is designed to be customized. That being the case, the “out of the box” configuration of the Dashboard is suitable for a developer machine and sandbox deployment, not a production deployment. For example, the solution ships with developer-friendly implementations that demonstrate basic functionality but may not represent functionality required for a specific installation.

Configuration considerations that your organization should evaluate include:

- What authentication system will be plugged into the STS.
- Custom metrics the organization may wish to have calculated.

Directory Systems

The primary method of allowing the STS website (and therefore Dashboard) to authenticate users is to configure the STS to integrate with an existing Active Directory or LDAP server in the target environment. For more information on Active directory, see 22709065.

Custom Metrics

The Dashboard has the ability to allow an organization to define their own metrics that can be calculated by the ETL process and displayed in the Dashboard UI. Further documentation can be found here.

Conclusion

Every organization has its own requirements and resources so no two deployments will be exactly alike. This document covers several technical options, but when evaluating your options, you should, of course, remember to take into account factors like cost, support expertise in your organization, security and privacy requirements, and so on.

For more information about the Ed-Fi Dashboard solution, visit techdocs.ed-fi.org. While the Ed-Fi Alliance doesn’t provide implementation services, it can connect organizations with vendors experienced in Ed-Fi technology deployments. Contact info@ed-fi.org for details.