IIS Web Servers (Web Apps and Web Services)

IIS Setup

The IIS best practices documentation at Microsoft should be used to configure and maintain the IIS web servers.

[https://techcommunity.microsoft.com/t5/core-infrastructure-and-security/iis-best-practices/ba-p/1241577]

Web servers require Windows IIS 10.x to be installed, and ASP.NET to be hosted under IIS. The installation of the full .NET Framework 4.8 is therefore a prerequisite. Refer to Microsoft's documentation for .NET Framework and IIS installation instructions.

IIS running on Windows 10 is limited to 20 concurrent connections, which is fine for development computers. However, Windows Server Standard Edition or better should be used for the Test, QA and Prod environments (even for relatively small systems).

DGP mainly uses the default IIS installation options:

- Web Management Tools
    - IIS Management Console
- World Wide Web Services
    - Application Development Features
        - .NET Extensibility 4.8
        - ASP.NET 4.8
        - ISAPI Extensions
        - ISAPI Filters
    - Common HTTP Features
        - Default Document
        - Directory Browsing
        - HTTP Errors
        - Static Content
    - Health and Diagnostics

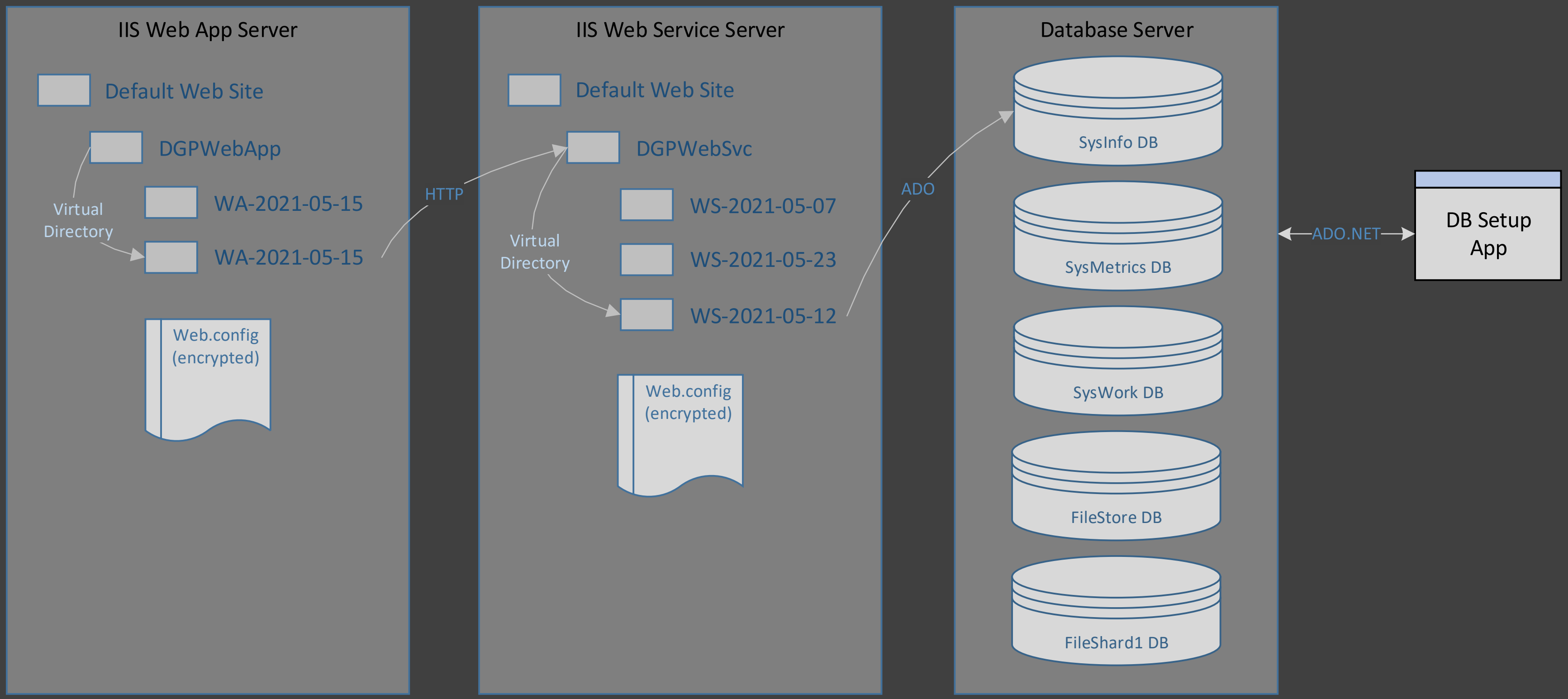
        - HTTP Logging
    - Performance Features
        - Dynamic Content Compression
        - Static Content Compression
    - Security
        - Request Filtering

Additional options can be selected as necessary, but these are the basics for the IIS web server.

The default web site created by IIS is fine to host DGP web apps. However, in the Test, QA and Prod environments, each application should be configured to use its own separate app domain. This can help to insure that problems or restarts for one web app do not spread to other web apps and web services, etc. Logging is the other main issue. In general, logging should be turned off most of the time, and only enabled as needed. Forgetting to disable logging can quickly fill up storage with enormous log files for web apps that see heavy traffic from many concurrent users.

Web Application Deployment

The deployment and maintenance of all of the tiers of a DGP Lattice system is both standardized and greatly simplified using two concepts. The first is the XCOPY deployment option for .NET applications which is used to create a folder that contains the application assemblies and associated files, and the second is the immutable append-only pattern used so frequently in all parts of DGP systems. All parts of DGP Lattice have been built with the full .NET Framework 4.8, and follow this same XCOPY immutable append-only deployment pattern.



The process for ASP.NET web apps and web services is identical. A parent folder is created on the web server below the appropriate web site. Web apps and web services are "published" in Visual Studio or a build server into a local folder. The publish process in Visual Studio allows for several pre-compilation (Ahead of Time) options.

This folder is then copied and renamed using the current date as its version number. Deployment is done by copying the version folder under the parent folder of the web app. The config file in the version folder is edited with values for the given environment. The path of the virtual directory for the web app/service is then edited to point to the release folder.

New releases repeat this process, copying a new release folder under the parent folder of the web application. The .config file is edited in the new release folder. Frequently, the config file is simply copied from a previous version folder into the new version folder. Then the path to the virtual directory for the web app/service is then edited to point to the new version folder. To rollback a new version, the path to the virtual directory is pointed back to the prior version folder.

This is among the simplest mechanisms possible for ease of deployments and rollbacks. Each native app, web app and web service follow the immutable append-only pattern internally, which means that each new version contains all of the functionality of previous versions, plus some new functionality and/or bug fixes. This enables continuous system evolution and the CI/CD/CT processes without ever breaking backward compatibility for client applications and integrated systems.

Native client applications are configured to connect to the URL of either a reverse proxy in the web app tier, or directly to the URL of a web service in the web service tier. The endpoint URL's used by native applications are stored in app.config files and in system list files. Web applications store the endpoint URL of web services in their web.config file. Web services store ADO.NET connection strings, system account info, encryption key info and file storage paths in their web.config files. Many of these values were created using the DB Setup utility when a system was first set up, and saved securely afterward.

The content of the web service web.config files can be encrypted using the aspnet_regiis.exe utility in the locations of selected environments. This utility can also be used to encrypt the app.config files of native apps whenever appropriate.

Web App and Web Service Configuration

DGPWebApp Web.config Keys

Web.Config Key	Sample Value	Description
LocState	ONLINE	Current state of the web app reverse proxy, which can be used to effectively disable a web service for new applications calling the Login method
SvcURL	http://localost/DGPWebSvc	The URL of the DGPWebSvc used by the web applications and reverse proxy pages (if applicable)

DGPWebSvc Web.config Keys

Web.Config Key	Sample Value	Description
SvcKeyVersion	SvcKeyV1	The label of the current encryption key
SvcKeyV1	(32-byte encryption key)	The value of the specified encryption key (maintains a list of current and all previous encryption keys)
LocState	ONLINE	Current state of the web service, which can be used to effectively disable a web service for new applications calling the Login method
System	DGP	The name of the system that owns the web service
Environment	Dev	The environment that owns the web service
Location	Win10Dev	The location that owns the web service
WebSvcName	DGPWebSvc	The name of the web service to be returned by the Login method
WebSvcVersion	2020-11-22	The date string of the web service version to be returned by the Login method
EventSource	.NET Runtime	The name of the event source to be used to log entries to the Event Viewer (the default value works if no custom event source has been created
EventID	1000	The event ID that works with the event source, when an event ID value is needed
TTLCheckFlag	ON	Turns TTL check on or off in the message processing pipeline
TTLMS	10000	The maximum MS allowed for the TTL check
UserCacheFlag	ON	Turns caching of the UserInfo object on or off in the message processing pipeline
UserCacheSec	600	How long the UserInfo object can be cached until it becomes obsolete and is removed
RateLimitFlag	OFF	Turns the rate limit check on or off
MaxMethBatch	10	Sets the maximum number of methods in a single API request message allowed by the message processing pipeline
MaxReqMsgKB	64	Sets the maximum size of an API request message allowed by the message processing pipeline

MaxRespMsgKB	64	Sets the maximum size of a response message allowed by the message processing pipeline
MaxFailedLogin	5	Sets the maximum number of failed authentications allowed by the message processing pipeline
PasswordLength	8	Sets the minimum password length allowed for password resets
ExpireDays	90	Sets the number of days until a password expires
MaxClaimBatch	5	Sets the maximum number of records that can be claimed by the AutoWork test harness
MaxFileSize	10000000	Sets the maximum size of a file in bytes that is allowed to be stored in Lattice
MaxSegSize	45000	Sets the maximum size of a file segment when uploading and downloading a file in Lattice
MaxFavorites	100	Sets the maximum number of favorites that are allowed per user in Lattice
SysInfo	ADO.NET connection string	The ADO.NET connection string for the SysInfo database
SysWork	ADO.NET connection string	The ADO.NET connection string for the SysWork database
SysMetrics	ADO.NET connection string	The ADO.NET connection string for the SysMetrics database
FileStore	ADO.NET connection string	The ADO.NET connection string for the FileStore database
FileShard1	ADO.NET connection string	The ADO.NET connection string for the FileShard1 database
TestDB1	ADO.NET connection string	The ADO.NET connection string for the TestDB1 database
TestDB2	ADO.NET connection string	The ADO.NET connection string for the TestDB2 database

Configuration files are used to store system and application "secrets" needed to initialize and run a DGP system. The aspnet_regiis.exe utility is used to encrypt to AppSettings section of the various .config files in the locations of environments whenever it is necessary for security. This type of encryption is used for DGP web apps and web services, but can also be used for client applications (like the AutoWork service, for example) when they contain potentially sensitive data.