**Module: WEB APPLICATION DEPLOYMENT**

**Code: SFDWD501**

**LU 1: Introduce software deployment concepts**

**Learning Outcome 1.1: Introduce web application deployment**

**Introduction to Web Deploy**

**What is Web Deploy?**

****

**Web Deploy** is an extensible client-server tool for syncing content and configuration to IIS. Web Deploy is used primarily in two scenarios:

Developers use it to sync (aka ‘publish') a compiled web applications (ASP .Net, PHP etc) from developer tools (Visual Studio, WebMatrix, etc) to IIS

IT professionals use it to migrate websites & applications from an operating system running an older version of IIS such as IIS6 to an operating system running a newer version of IIS such as IIS 7.5.

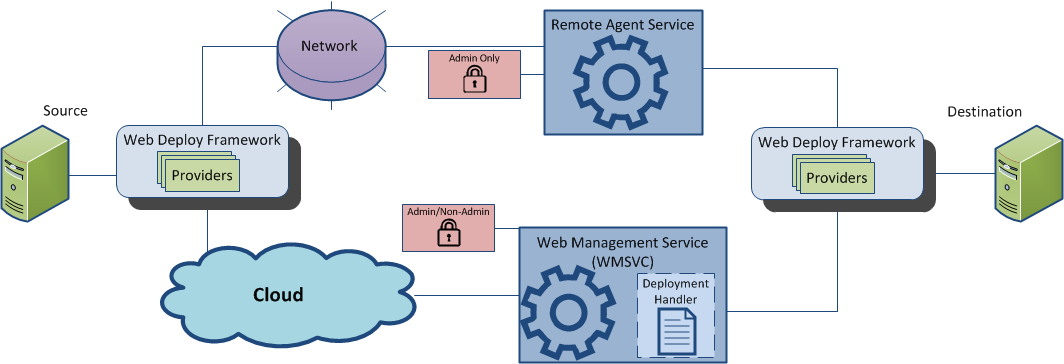
**How does Web Deploy compare to FTP?**

Web Deploy is often compared to technologies like FTP, XCOPY or RoboCopy. While these technologies are useful, Web Deploy offers several benefits.

**Here is a comparison of Web Deploy to FTP:**

* **Web Deploy is faster than FTP**. Web Deploy does not issue a different command for each operation. Instead, it does a comparison at the start of the sync and only transfers changes.
* **Web Deploy is secure**. Web Deploy supports transfer over HTTPS. Note that variants of FTP such as SFTP and FTPS are also secure.
* **Web Deploy can set security descriptors (ACLs) on destination files and directories**. For example, you can use Web Deploy to give a Windows user specific access to your application's ‘Downloads' folder during deployment.
* **Web Deploy can publish databases.** Web Deploy has out-of-box support for scripting out SQL Server, MySQL Server, Sqlite and SQL Server Compact databases and applying the resulting script during the sync. This can be very handy if your app needs a database to function.
* **Web Deploy can apply transforms during deployment.** You can use Web Deploy to change a connection string or application setting during a sync. Web Deploy supports a large range of transforms, including transforming IIS settings.
* **Web Deploy integrates with Visual Studio 2010 and Web Matrix.**
* **Web Deploy is extensible.** Web Deploy has a rich, publicly-accessible extensibility model which lets you author new scenarios.

**How does it work?**

****

Most of the Web Deploy operations are modeled around sync operation between a source and a destination. Sync operation is orchestrated by Web Deploy framework using one or more Web deploys providers as described below.

**Framework**

Web Deploy consists of a framework which manages connection state and orchestrates the sync between source & destination. For example, the framework is responsible for skipping syncing certain types of content, running rules and transforming files.

The framework is accessible via a public API as well as via a command-line executable, msdeploy.exe.

**Providers**

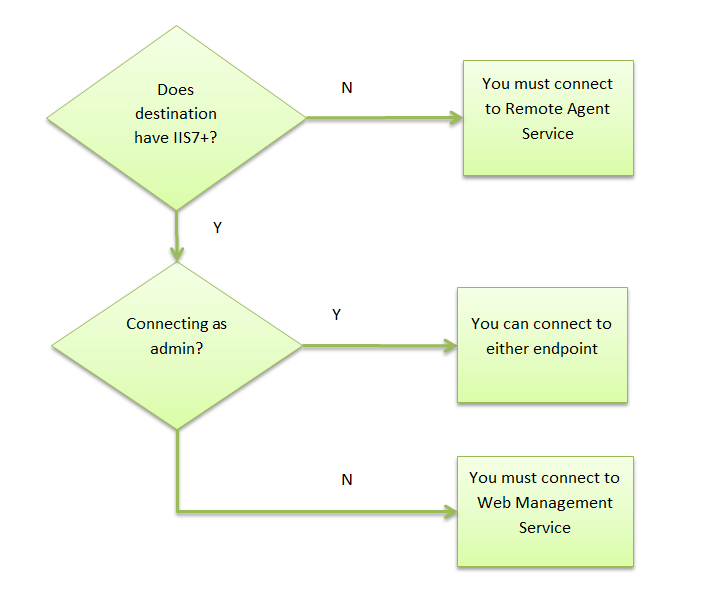
Web Deploy can synchronize any kind of data between two locations through the use of its built-in providers which plug into the framework. Providers know how to synchronize a particular type of data between two sources, or retrieve useful information about the data source.

For example, Web Deploy has a provider that works with SQL databases (dbFullSql) that can synchronize an entire SQL Server database to another server. Other built-in providers synchronize MySQL databases (dbMySql), IIS 7 configuration (appHostConfig), GAC assemblies (gacAssembly), or COM objects (comObject32/comObject64). A more complete listing can be found on TechNet.

The provider model is extensible and lets developers write their own providers for Web Deploy if they need to synchronize custom data.

**Under the Hood**

When a source initiates an action through Web Deploy, the Web Deploy Framework establishes a connection with the destination. Web Deploy supports two connection end-points, and the one you use depends on several factors:



* If the destination is Windows Server 2003 running IIS6, you can connect to its Web Deployment Agent Service (also known as the Remote Agent Service) endpoint. The Remote Agent Service provides security for this connection, and is only accessible to administrator users.
* If the destination is Windows Server 2008 running IIS7 or Windows Server 2008 R2 running IIS7.5 AND you are connecting with non-administrator credentials, you must connect to the Web Management Service endpoint. Web Deploy installs a handler in Web Management Service, which allows non-administrators to deploy after authenticating and authorizing with the Web Management Service.
* If the destination is Windows Server 2008 running IIS7 or Windows Server 2008 R2 running IIS7.5 AND you are connecting with administrator credentials, you can connect to either endpoint.

**Learning Outcome 1.1: Introduce web application deployment**

1. **Description of Deployment** 
   1. **Definition:** Web Deploy (Web Deployment Tool) is a Microsoft tool that significantly simplifies the migration, management, and deployment of IIS web servers, web applications, and websites. You can install and configure Web Deploy as a third-party component on your managed server.
   2. **Reasons to have Web Deploy on your server**
2. ***Simple applications publishing:*** Web developers who write code in ***Visual Studio* (IDE)** and ***Web Matrix*** (development tool) can use Web Deploy to publish their applications to a production server.
3. ***New market for your hosting plans:*** Web Matrix helps its users find a suitable hosting plan in the ***Microsoft Web Hosting Gallery***, a catalog in which hosting providers advertise hosting offers.
   1. **Benefits of Cloud Hosting**

****

Cloud hosting has become increasingly popular in recent years, and for good reason. Considering it’s recent rise, you probably have a lot of questions. It’s a flexible, scalable, and high performing hosting solution that offers your business a unique set of features.

**1. Flexible Pricing Structure**

In most other styles of hosting you pay a monthly rate regardless of whether to not you utilize all of the server resources. With cloud hosting, you only pay for what you use.

**2. Very Easy to Scale Server Resources**

With cloud hosting scaling your server resources couldn’t be easier. For most cloud servers you’ll have access to an intuitive site management dashboard that lets you view your site’s performance in real time.

**3. Redundant Server Environment**

With most types of hosting your site lives on a single server. If something were to happen to that server, then your site would go offline and you wouldn’t be able to get back online until that server is fixed.

**4. High Uptime and Availability**

If you’re using any kind of traditional hosting your site’s uptime depends upon the physical server environment.

**5. Speedy Server Setup Process**

You can deploy a cloud hosting server in record time. Unless you’re signing up for a beginner shared hosting package, it might take some time to deploy your web server.

**6. Safety From Server Hardware Issues**

With cloud hosting your site is isolated from any physical server issues including hacking, hardware failure, or system overload.

**7. Faster Website Speed and Performance**

Most cloud servers can offer you blazing fast speeds. In addition to being able to effortlessly increase your site’s capacity, cloud hosting also allows for easier load balancing between multiple server environments, which can help to put much less strain on a single server’s resources.

**8. Works Great for Sites With Scalable Traffic**

If your traffic levels are up and down, then it’s going to be hard to find a host that’ll give you the best value for your money. With cloud hosting, you have the ability to easily scale your server resources up and down on an as-needed basis

**9. Simple Server Management Dashboard**

In the past cloud servers were a little difficult to manage and it required technical expertise to manage and scale your server effectively.

**10. Decrease Your Overall Environmental Impact**

If you’re a company or individual who’s concerned about their environmental impact, but need a hosting solution that goes beyond traditional shared hosting, then cloud hosting could be exactly what you’re looking for.

**Fundamental Steps to Deploying a Website**

**Step 1: Preparation**

There are a few things to consider when you are finalizing a website, and they all depend on what type of deployment you will be completing.

The three general scenarios of a website deployment is:

1. The client has **nothing** (i.e. this is their first website)
2. The client **already has hosting** and you will be deploying the site on their server
3. The client **already has hosting** but you will be moving to a new server

**Step 2: Set Up DNS Records**

If you are going to be setting up the website on a new host and you have access to the DNS management administration, then that’s great.

Create yourself an A record (the address record that maps a domain name to the IP address of the server) or subdomain record for a live development site such as dev.domainname.com. Point this subdomain to the IP address of the new server.

**Step 3: Set Up a Live Testing Site**

A practice I recommend doing is setting up a subdomain URL prior to officially deploying the site. Something like dev.domainname.com which will eventually be on domainname.com.

**Step 4: Set Up Email Accounts**

Developers deploying a website often overlook email, but it will be a priority to the client. Does your client have mail hosted on their old server? Are you moving their email?

If their email is currently in the same hosting account as the old website, then you will probably be moving mail to the new server.

If so, collect all email account addresses and set up the exact same accounts on the new server. In most cases, you then won’t need to change anything, it will just transition to the new mail server at the same time the website does.

### Step 5: Backup and Go Live

Even if you are hosting on a new server, take a full backup including any databases of the old website, as you never know when you might need something.

**Website Deployment Checklist**

1. Have access to DNS record management or know the people to contact
2. Set up the DNS records and make sure that all the settings are correct
3. Set up and test the website on the production server (where it will live)
4. Set up email
5. Back up the old site (if applicable) and deploy the new one
6. **Exploration of deployment process / activities**
7. **Release:** Release is to make the product built available for testing before deployment. Usually sent from the development team.

The release activity follows from the completed development process and is sometimes classified as part of the development process rather than deployment process. Includes all operation to prepare the system for assembly and transfer to the computer system on which it will be run on the production.

1. **Installation:** The installation includesconfiguration, running, testing, and making necessary changes. The word implementation in sometimes used to mean the same thing.

The installation involves establishing some form of command, short cut script or service for executing the software (manually or automatically). For complex system it may involve configuration of the system.

1. **Activation:** Is the activity of starting up the executable component of software for the first time(not to be configured with the common use of the term activation concerning a software license which is a function of digital right management system)
2. **Deactivation:** Is the inverse of activation and refers to shutting down any already executing components of a system. Deactivation is often required to perform other deployment activities.
3. **Uninstall:** Uninstall is the inverse of installation. It is the removal of a system. It may also involve some reconfigurations of others software system in order to remove the installed system dependency.
4. **Update:** Update process replaces an earlier version of all part of a software system with a newer release. It commonly consists of deactivation followed by installation.
5. **Built-in update:** Mechanism for installing updates are built into some software systems (or in the case of some operating systems such as Linux, Android and IOS into the operating system itself).

Automation of this update process ranges from folly automatic to user installed and controlled.

1. **Version tracking:** Version tracking systems help the user find and install updates to software systems. For example: Software catalog stories version and other information for each software package installed on a localsystem.
2. **Adaptation:** Refers a process where an interactive system(Adaptive system) adapts its behavior to individuals users based on information acquired about its users and its environment.

**Learning Outcome 1.2: Explore the deployment environment**

* 1. **Description of 5-tier System Architecture**

In software engineering, multitier architecture (often referred to as n-tier architecture) or multilayered architecture is client–server architecture in which presentation, application processing, and data management functions are physically separated.

* 1. **Persistence:** Is a group of files which is used to communicate between the application and DB.
  2. **Back-end services:** Back end is the scripts and database interaction that actually process data. Client-side is contrasted with server-side.
  3. **Virtualization:** Is a conceptual model specifying the arrangement and interrelationships of the particular components involved in delivering a virtual rather than physical version of something, such as an operating system (OS), a server, a storage device or network resources.
  4. **Application logic:** Also called the middle tier, logic tier, business logic or logic tier, this tier is pulled from the presentation tier. It controls application functionality by performing detailed processing.
  5. **Presentation layer:** Provides external access to the solution (specifically the Business Logic, and by extension the underlying data). This includes User Interfaces and can include System Interfaces such as Http Handlers, JSON Services and so on.

1. **Exploration of Virtualization technology** 
   1. **Description of virtualization**

Virtualization refers to the creation of a virtual resource such as a server, desktop, operating system, file, storage or network.

The main goal of virtualization is to manage workloads by radically transforming traditional computing to make it more scalable. Virtualization has been a part of the IT landscape for decades now, and today it can be applied to a wide range of system layers, including operating system-level virtualization, hardware-level virtualization and server virtualization.

* 1. **Properties and benefits of Virtual machine**

**Virtual Machines Benefits**

In general, VMware virtual machines possess four key characteristics that benefit the user:

* **Compatibility:** Virtual machines are compatible with all standard x86 computers
* **Isolation:** Virtual machines are isolated from each other as if physically separated
* **Encapsulation:** Virtual machines encapsulate a complete computing environment
* **Hardware independence:** Virtual machines run independently of underlying hardware

**Virtual Machine Properties**

The Virtual Machine Properties tab shows details about a client computer that is deployed as a virtual machine (VM).

**Virtualization Client Name**

Name of the client computer that provides a management access point for the hypervisor where the virtual machine is deployed.

**Virtual Machine Name**

Name of the client computer for the virtual machine

**Vendor**

Hypervisor for the virtual machine

**Uuid**

Universally unique ID (UUID) for the virtual machine.

**Hardware Version**

The hardware version of the virtual machine on the host computer, which determines the virtual hardware functions supported for the VM.

**Host**

IP address for the host where the virtual machine is located

**Host Version**

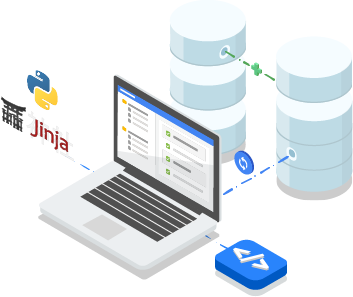
Version and build for the hypervisor software on the host

* 1. **Hyper-visor technology**

Hypervisor technology is software on which multiple virtual machines can run, with the hypervisor layer controlling the hardware and allocating resources to each VM operating system.

**Learning Outcome 1.3: Identify deployment tools, equipment in accordance with application requirements.**

1. **Description of software deployment Tools** 
   1. **Jenkins:** Is an open source automation server which enables developers around the world to reliably build, test, and deploy their software
   2. **Google Cloud deployment Manager:** is an infrastructure management service that makes it simple to create, deploy, and manage Google Cloud Platform resources.



* 1. **CircleCI:** CircleCI is a cloud-based system — no dedicated server required, and you do not need to administrate it. However, it also offers an on-prem solution that allows you to run it in your private cloud or data center.
  2. **Octopus Deploy:** Is an automated deployment and release management server. It is designed to simplify deployment of ASP.NET applications, Windows Services and databases.
  3. **AWS Code Deploy:** Is a service that automates code deployments to Elastic Compute Cloud (EC2) and on-premises servers. Accelerating how fast a developer can release code allows him to release new features for an application faster and avoid deployment errors in complex applications.

1. **Description of hardware deployment equipment** 
   1. **Server:** A server is a type of computer or device on a network that manages network resources. Servers are often dedicated, meaning that they perform no other tasks besides their server tasks.

A server in this case could refer to the program that is managing resources rather than the entire computer

* 1. **Computer:** Is an electronic device for storing and processing data, typically in binary form, according to instructions given to it in a variable program.
  2. **Operating system:** Isthe low-level software that supports a computer's basic functions, such as scheduling tasks and controlling peripherals.
  3. **Antivirus:** Is software designed to detect and destroy computer viruses

**LU2: Analyze web application deployment requirements**

**Learning Outcome 2.1: Explore web application deployment server in line with application requirements.**

1. **Analysis of application requirements**

* Customer Requirements
* Functional Requirements
* Non-functional Requirements
* Performance Requirements
* Design Requirements
* Delivery Requirements
* Allocation Requirements

1. **Identification of suitable deployment server**

* Backup Services
* Customer Service
* Adequate Scalability
* Guaranteed Up-time
* Reputation

**Learning Outcome 2.2: Interpret hardware and software deployment requirements.**

1. **Identification of hardware deployment requirements**

* Server processor
* Server RAM
* Disk storage

1. **Identification of Software deployment requirements**

* Operating system
* Third party software’s compatibility
* Firewalls settings

1. **Production of analysis report**

* **Server**
* **Deployment package:** A deployment package is a ZIP archive that contains your function code and dependencies. You need to create a deployment package if you use the Lambda API to manage functions, or if you need to include libraries and dependencies other than the AWS SDK.

**Learning Outcome 2.3: Identify technologies used and network compatibility.**

1. **Exploration of Network compatibility**

* **Protocols:** A protocol is a set of rules and guidelines for communicating data. Rules are defined for each step and process during communication between two or more computers. Networks have to follow these rules to successfully transmit data
* **Server accessibility**
* **Server visibility**

1. **Identification of web technology used for web development**

* PHP Version
* MySQL version
* Apache server

**LU3: Distribute the application**

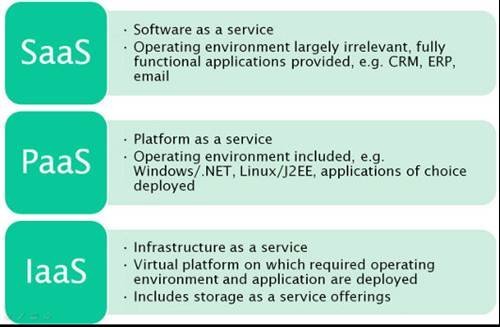
**Learning Outcome 3.1: Inspect web application deployment model.**

1. **Analysis of web application deployment models** 
   1. **Platform as a service (PaaS):** Platform as a Service (PaaS) or Application Platform as a Service (aPaaS) or platform-based service is a category of cloud computing services that provides a platform allowing customers to develop, run, and manage applications without the complexity of building and maintaining the infrastructure typically associated with developing and launching an app
   2. **Infrastructure as a service (IaaS)**: Infrastructure as a service (IaaS) is a service model that delivers computer infrastructure on an outsourced basis to support enterprise operations.

Typically, IaaS provides hardware, storage, servers and data center space or network components; it may also include software.

* 1. **Software as a service (SaaS):** Is a software distribution model in which a third-party provider hosts applications and makes them available to customers over the Internet.

SaaS is one of three main categories of cloud computing, alongside infrastructure as a service (IaaS) and platform as a service (PaaS).



**Learning Outcome 3.2: Allocate the web application package**

1. **Allocation of web application deployment package.**

* **Application files:** An "application file" is also the term used to describe a file that a program puts on a computer after it gets installed**.**
* **Database file:** DB files store data information, usually stored in a series of tables, table fields, and field data values.

The information is then stored and organized according to the data model, the most common model structure being the relational model.

1. **Application Code/logic configuration**

* **Client-side code:** This party requests pages from the Server, and displays them to the user. In most cases, the client is a web browser
* **Server-side code:** This party is responsible for serving pages.

## Server-side Programming

Server-side programming is the general name for the kinds of programs which are run on the **Server**.

### Uses

* Process user input.
* Compiles pages.
* Structure web applications.
* Interact with permanent storage (SQL, files).

### Example Languages

* PHP
* Python
* ASP.Net in C#, C++, or Visual Basic.
* Nearly any language (C++, C#, Java). These were not designed specifically for the task, but are now often used for application-level web services.

## Client-side programming

Much like the server-side, Client-side programming is the name for all of the programs which are run on the **Client**.

### Uses

* Make interactive webpages.
* Make stuff happen dynamically on the web page.
* Interact with temporary storage, and local storage (Cookies, localStorage).
* Send requests to the server, and retrieve data from it.
* Provide a remote service for client-side applications, such as software registration, content delivery, or remote multi-player gaming.

### Example languages

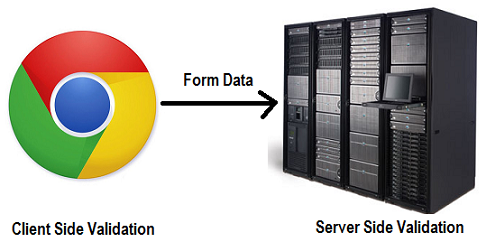
* JavaScript (primarily)
* HTML
* CSS
* Any language running on a client device that interacts with a remote service is a client-side language.

1. **Configuration of database server**

* **Server configuration:** You must customize the database server properties and features by setting configuration parameters, create storage spaces, and configure connectivity. You can automate startup.

You customize the database server properties by setting or modifying configuration parameters in the on config file.

* **Server validation:** In the Server Side Validation, the input submitted by the user is being sent to the server and validated using one of server side scripting languages such as ASP.Net, PHP etc.



**Learning Outcome 3.3: Experiment and launch application in line with customer needs.**

* 1. **Collection process of web application files.**
* Index file
* Resources folder
* Includes folder
* Classes
* Database
  1. **Deployment in staging environment:** A staging environment is a part of a workflow in most development projects and the deployment process.

It is a place where you can find a preview of work that was done by developers to review and, once satisfied, approve to be moved to production for the public to see

* **Staging server**: is a type of server that is used to test a software, website or service in a production-similar environment before being set live.

It is part of a staging environment or staging site, where it serves as a temporary hosting and testing server for any new software or websites.

* **Staging Database:** is a temporary table that is used to stage the data for temporary purpose just before loading it to the Target table from the Source Table.
* **Staging application files:** A Staging application offers an isolated environment that allows you to push and pull changes between Live and Staging applications without impacting the Live application.
  1. **Testing phase and validation process**
* **Smoke test:** Is a type of software testing that comprises of a non-exhaustive set of tests that aim at ensuring that the most important functions work. The result of this testing is used to decide if a build is stable enough to proceed with further testing.
* **User acceptance:** User acceptance testing (UAT) is the last phase of the software testing process.

During UAT, actual software users test the software to make sure it can handle required tasks in real-world scenarios, according to specifications. UAT is one of the final and critical software project procedures that must occur before newly developed software is rolled out to the market.

* **Validity:** The state of being legally or officially binding or acceptable.

1. **Deployment in production environment.**

* Deployment server
* Database
* Application files
* Final delivery

**Thank you!**