IIS INTRODUCTION:

Hello World… In today’s fast-paced digital world, businesses rely on robust web servers to host applications, websites, and services efficiently. Imagine you’re setting up a company website, and you need a reliable way to host it on a Windows server. That’s where **Internet Information Services (IIS)** comes in! IIS is a powerful web server that allows businesses to host websites, APIs, and applications with ease. From small business websites to large-scale enterprise applications, IIS offers flexibility, security, and performance optimization. Let’s go….

**What is IIS???**

* Internet Information Services
* A web server runs on a computer (physical or virtual) and processes the web traffic.
* Some other common web servers are:  
  - IIS (Windows-based)  
  - Apache HTTP Server (Cross Platform)  
  - Nginx (Cross Platform, High-performance web server)
* IIS is a web server software developed by Microsoft that is used to host websites, web applications, and services on Windows servers.
* It allows users to serve HTML pages, run .Net applications, and host various types of content like images, videos, and web services.
* IIS is a Windows component.
* It is responsible for hosting both Web and FTP sites on the server.

**Role of a Web Server:**

1. A user (client) requests a website by entering a URL (www.sample.com) in a web browser.
2. The request is sent to a web server like IIS.
3. The web server processes the request & sends the appropriate web page back to the browser.
4. The browser displays the web page to the user.

A computer screen shot of a computer

AI-generated content may be incorrect.

Role of a Web Server

**Why IIS???**

* IIS comes with a user-friendly management interface (IIS Manager) and integrates well with Windows tools.
* Built-in features like SSL/TLS support, authentication mechanisms, and request filtering help keep your web applications secure.
* IIS can handle everything from small personal websites to large enterprise-level applications with high traffic.
* Features like dynamic compression, caching, and HTTP/2 support ensure fast and efficient content delivery.
* IIS can be customized with modules and extensions to meet specific requirements.

IIS Installation:

Hello World… In today’s cloud-first world, hosting web applications on scalable infrastructure is essential. **Amazon EC2 (Elastic Compute Cloud)** allows businesses and developers to run Windows Server instances without managing physical hardware. With AWS EC2, you get flexibility, scalability, and cost-effectiveness while maintaining full control over your Windows Server environment.

A computer with a symbol and a sign

AI-generated content may be incorrect.

IIS is a powerful and feature-rich web server that enables seamless web hosting on Windows-based systems. In this guide, we’ll walk you through launching a Windows EC2 instance, installing IIS, and verifying the installation to ensure it’s running correctly.

**Overview:**

* Launching a Windows EC2 Server
* Connect to Windows Server
* Installing IIS on the server
* Verifying the Installation

**Step1: Launching a Windows EC2 Server**

1. Go to the **AWS Management Console**.
2. Navigate to the **EC2** dashboard under the **Compute** section.
3. Launch an instance.
4. Give a name to the instance, such as My Server — IIS.
5. Choose a **Windows Server AMI**(Ex: Windows\_Server-2025-English-Full-Base)
6. Choose an **instance type** like t3.medium or so.
7. Select a **key pair** for secure access.  
   If a key pair isn’t available, we need to create one.
8. **Security Group** — Allow **RDP (port 3389)** for remote access and **HTTP (port 80)** for web traffic.
9. And then the **Launch Instance** option.

**Step2: Connect to Windows Server**

* Once the above EC2 is launched and running fine, we need to connect to the server.
* Select the **Instance** and then the **Connect** option.

A screenshot of a computer

AI-generated content may be incorrect.

* Choose the **RDP client** option.
* Download the **remote desktop file**.
* Then, Get the **password** option to login to the server.

A screenshot of a computer

AI-generated content may be incorrect.

* For the password, we need to upload the private key (the key pair which associated with the respective server)
* **Decrypt** the password option and copy the password.

A screenshot of a computer

AI-generated content may be incorrect.

* Open the above downloaded RDP file.
* Provide the credentials (Username, and password) to log in.

A blue wavy surface with a blue background

AI-generated content may be incorrect.

**Step3: Installing IIS on the Server**

* **Start**→**Control Panel**→**Programs** →**Turn Windows features on or off**

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

* Once the **Server Manager** is opened, select the **Add Roles and Features** option from the**Manage** section.

A screenshot of a computer

AI-generated content may be incorrect.

* Below are the screenshots for the complete process of installation:

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Step4: Verify the IIS Installation**

* Once the IIS is installed, you can verify it by accessing the “**http://localhost**” or “**http://localhost:80**” or “**http://ec2-server-public-ip**” URLs.
* If the installation is successful, the Welcome/Home page of the IIS will be displayed.
* Another way to verify:

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A computer screen shot of a computer

AI-generated content may be incorrect.

IIS HOSTING the First Simple Web APP:

A computer and a few other icons

AI-generated content may be incorrect.

Hosting Web App by Sai Manasa

Hello World… IIS provides a stable and scalable platform for hosting websites and applications on Windows Server, making it an ideal choice for developers and businesses alike.

In this guide, we will walk you through **deploying a simple web application** (HTML) on IIS inside your **AWS EC2 Windows Server**. This includes configuring IIS, uploading your website files, and verifying that your web app is accessible. So, let’s get started…

* Steps for launching an EC2 Windows server and Installing IIS on the server: [Blog](https://medium.com/@saimanasak/iis-installation-2c63d6e424de)

**Step1: Create a simple web page**

* Below is the HTML code snippet for a simple web page.
* Make sure the file name is index.html.
* Because: IIS is configured to look for default files like index.html, index.htm, or default.aspx in a directory when no specific file is requested. If no such file exists, IIS may return a **403 Forbidden** or **404 Not Found** error.

**Step2: Hosting the Web App**

* Initially, we need to add a website.

A screenshot of a computer

AI-generated content may be incorrect.

* Fill out the fields.
* **Site Name:** Sample
* **Physical Path:** Browse the location of the web application folder.
* We need to store the web app files in the C:\inetpub\wwwroot directory. Because:  
  IIS is pre-configured to serve files from wwwroot, so placing files here ensures they are accessible without extra configuration.  
  The wwwroot folder has built-in security permissions that restrict unauthorized access while allowing IIS to serve files securely.  
  wwwroot is the standard directory for web hosting on Windows servers, making it easier for developers and administrators to manage and maintain web applications.
* We can also store the files outside this directory, but we need to configure IIS accordingly.
* **Binding:**  
  Type: HTTP  
  IP Address: All Unassigned  
  Port: 81 (The IIS is already hosted on port 80 if the IIS default web server is removed we can then use the 80 OR we need to configure the hosts when we want to access the new app on 80 though the default IIS browser is present)

A screenshot of a computer

AI-generated content may be incorrect.

* We can browse our website in the below ways or using the “**http://localhost:81**” URL.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

* Finally, the app is hosted 🙌

A screenshot of a computer

AI-generated content may be incorrect.

**Which I tried:**

**Default Page:**

A computer screen shot of a computer

AI-generated content may be incorrect.

**My simple webpage:**

A screenshot of a computer

AI-generated content may be incorrect.

**The simple webpage which is Showed below:**

A screenshot of a computer

AI-generated content may be incorrect.