Інсталяція і налаштування веб-сервера Apache

HTTP-сервер Apache - самий широко використовуваний веб-сервер в світі. Він має безліч потужних функцій, включаючи динамічно завантажувані модулі, надійну підтримку різних медиаформатов і інтеграцію з іншим популярним програмним забезпеченням.

Крок 1 – Інсталяція Apache

Apache доступний в репозиторіях програмного забезпечення Ubuntu за замовчуванням, і тому його можна встановити за допомогою стандартних інструментів управління пакетами.

Для початку вивантажимо покажчик локальних пакетів, щоб відобразити останні зміни на попередніх рівнях:



Потім встановимо пакет apache2:



Після підтвердження установки apt встановить Apache і всіх необхідних залежностей.

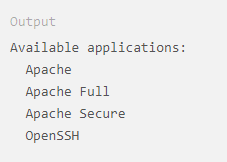
Крок 2 - Налаштування брандмауера

Перш ніж тестувати Apache, необхідно змінити налаштування брандмауера, щоб дозволити доступ до веб-портам за замовчуванням. Якщо ви виконали попередні вказівки, у вас повинен бути встановлений брандмауер UFW, налаштований для обмеження доступу до вашого сервера.

Під час установки Apache реєструється в UFW, надаючи кілька профілів додатків, які можна використовувати для включення або відключення доступу до Apache через брандмауер.

Виведіть список профілів додатків ufw, ввівши таку команду:





Як показав висновок, є три профілі, доступних для Apache:

* Apache: цей профіль відкриває тільки порт 80 (нормальний веб-трафік без шифрування)
* Apache Full: цей профіль відкриває порт 80 (нормальний веб-трафік без шифрування) і порт 443 (трафік з шифруванням TLS / SSL)
* Apache Secure: цей профіль відкриває тільки порт 443 (трафік з шифруванням TLS / SSL)

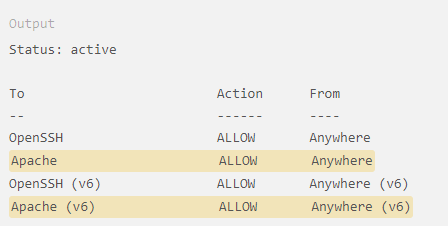
Рекомендується застосовувати самий обмежує профіль, який буде дозволяти заданий трафік. Оскільки в цьому модулі ми ще не налаштували SSL для нашого сервера, нам потрібно буде тільки дозволити трафік на порту 80.



Для перевірки змін введіть:



У висновку буде представлений список дозволеного трафіку HTTP:



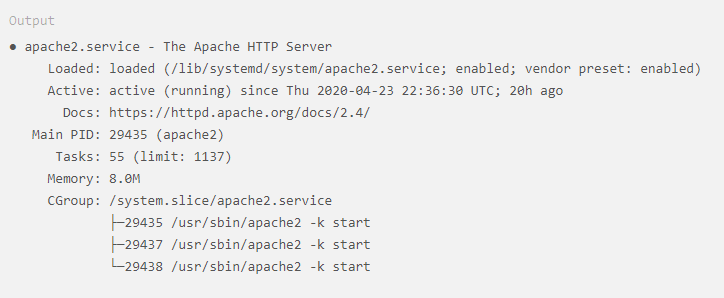
Як показав висновок, профіль був активований для доступу до веб-сервера Apache.

Крок 3 - Перевірка веб-сервера

В кінці процесу установки Ubuntu 20.04 запускає Apache. Веб-сервер уже повинен бути запущений і працювати.

Використовуйте команду ініціалізації systemd, щоб перевірити роботу служби:





Висновок підтвердив, що служба успішно запущена. Однак найкраще протестувати її запуск за допомогою запиту сторінки з Apache.

Відкрийте сторінку Apache за замовчуванням, щоб підтвердити роботу програмного забезпечення через ваш IP-адреса: Якщо ви не знаєте IP-адресу вашого сервера, є кілька способів дізнатися його за допомогою командного рядка.

Спробуйте ввести в командному рядку сервера наступну команду:



Ви отримаєте декілька адрес, розділених пробілами. Ви можете спробувати кожен з них в браузері, щоб переконатися в їх працездатності.

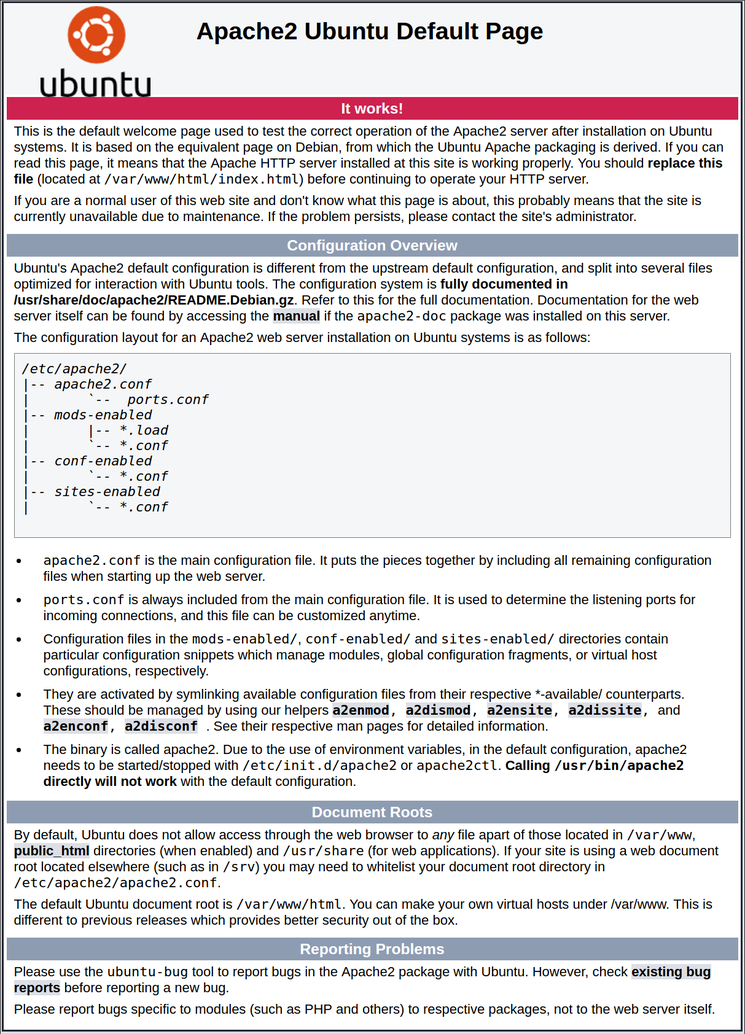
Ще один варіант - використовувати інструмент Icanhazip, який повинен видати вам ваш публічний IP-адресу, видимий в іншому розташуванні в Інтернеті:



Коли ви дізнаєтеся IP-адреса вашого сервера, введіть його в адресний рядок вашого сервера:



Ви побачите веб-сторінку Ubuntu 20.04 Apache за замовчуванням:



Крок 4 - Налаштування віртуальних хостів

При використанні веб-сервера Apache ви можете використовувати віртуальні хости (аналогічні серверним блокам в Nginx) для інкапсуляції даних конфігурації та розміщення на одному сервері декількох доменів. Ми створимо домен your\_domain, але ви повинні замінити це ім'я власним доменним ім'ям. Про те, як налаштувати доменне ім'я за допомогою DigitalOcean, можна дізнатися з нашої мережевої документації.

В Apache в Ubuntu 20.04 за замовчуванням включений один серверний блок, налаштований на обслуговування документів з директорії / var / www / html. Хоча це добре працює для окремого сайту, при хостингу декількох сайтів це незручно. Замість зміни / var / www / html ми створимо в / var / www структуру директорій для нашого сайту your\_domain, залишивши / var / www / html як директорію за замовчуванням для виведення в разі, якщо запит клієнта не відповідають ніякі інші сайти.

Створіть наступну директорію для your\_domain:



Потім призначте володіння Директорією за допомогою змінної середовища $ USER:



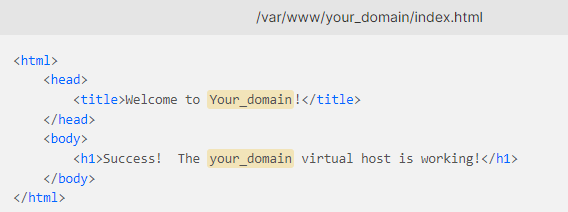
Дозволи кореневих директорій веб-сервера повинні бути правильними, якщо ви не змінювали значення umask, яке встановлює дозволу файлу за замовчуванням. Щоб переконатися, що дозволу коректні, дозволити власнику читати, писати і запускати файли, а групам і іншим користувачам дозволити тільки читати і запускати файли, ви можете ввести наступну команду:



Потім створіть в якості прикладу сторінку index.html, використовуючи nano або свій улюблений редактор:



Додайте в сторінку наступний зразок коду HTML:

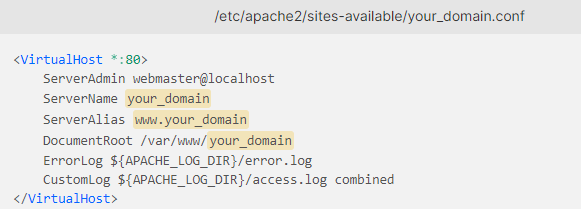


Збережіть файл і закрийте його після завершення.

Для обслуговування цього контенту Apache необхідно створити файл віртуального хоста з правильними директивами. Замість зміни файлу конфігурації за замовчуванням, розташованого в /etc/apache2/sites-available/000-default.conf, ми створимо новий файл в /etc/apache2/sites-available/your\_domain.conf:



Введіть наступний блок конфігурації, який схожий на заданий за замовчуванням, але оновлений з урахуванням нової директорії і доменного імені:



Зверніть увагу, що ми змінили DocumentRoot на нову директорію, а ServerAdmin - на адресу електронної пошти, доступний адміністратору сайту your\_domain. Також ми додали дві директиви: директиву ServerName, яка встановлює базовий домен і повинна відповідати визначенню віртуального хоста, і директиву ServerAlias, яка задає додаткові імена, які повинні давати збіг, як якщо б вони були базовими іменами.

Збережіть файл і закрийте його після завершення.

Активуємо файл за допомогою інструменту a2ensite:



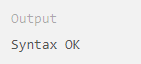
Вимкніть сайт за замовчуванням, визначених у 000-default.conf:



Потім перевіримо помилки конфігурації:



Ви повинні отримати наступний результат:



Перезапустити Apache для внесення змін:



Тепер Apache повинен обслуговувати ваше доменне ім'я. Ви можете перевірити це, відкривши в браузері адресу http://example.com, після чого повинні побачити приблизно наступне:

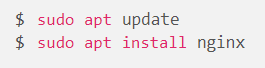


Інсталяція та налаштування Nginx

Крок 1 - Установка Nginx

Оскільки Nginx доступний в репозиторіях Ubuntu за замовчуванням, його можна встановити з цих репозиторіїв за допомогою системи пакетів apt.

Так як це перша наша взаємодія з системою пакетів aptв цьому сеансі, ми також оновимо індекс локальних пакетів, щоб отримати доступ до актуальних списками пакетів. Потім ми можемо виконати установку nginx:



Після прийняття процедури aptвиконає установку Nginx та інших необхідних залежностей на ваш сервер.

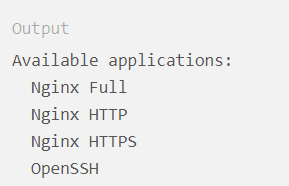
Крок 2 - Налаштування брандмауера

Перед тестуванням Nginx необхідно виконати настройку програмного забезпечення брандмауера, щоб дозволити доступ до служби. Nginx реєструє себе як службу за допомогою ufwпісля установки, спрощуючи доступ до Nginx.

Для виведення списку конфігурацій додатків, які відомі ufw, необхідно ввести наступну команду:



Необхідно отримати список профілів додатків:



Як показав висновок, є три профілі, доступних для Nginx:

* Nginx Full : цей профіль відкриває порт 80 (звичайний веб-трафік без шифрування) і порт 443 (трафік з шифруванням TLS / SSL)
* Nginx HTTP : цей профіль відкриває тільки порт 80 (звичайний веб-трафік без шифрування)
* Nginx HTTPS : цей профіль відкриває тільки порт 443 (трафік з шифруванням TLS / SSL)

Рекомендується застосовувати самий обмежує профіль, який буде дозволяти заданий трафік. Зараз нам потрібно буде вирішити трафік на порту 80.

Для активації можна ввести наступну команду:



Для перевірки змін введіть:



Висновок вкаже, який трафік HTTP дозволений:

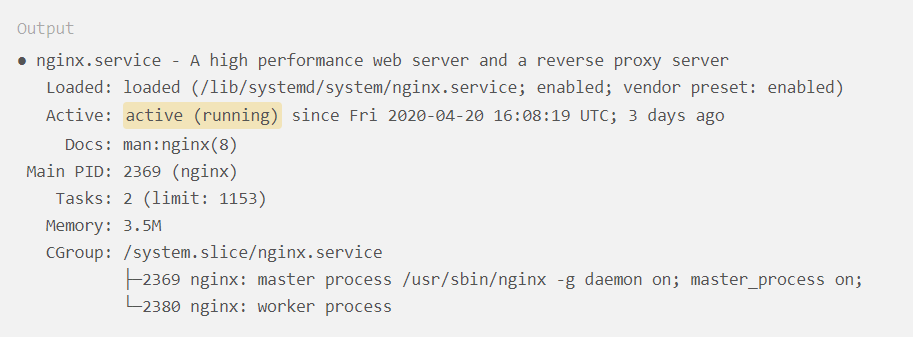


Крок 3 - Перевірка веб-сервера

В кінці процесу установки Ubuntu 20.04 запускає Nginx. Веб-сервер уже повинен бути запущений і працювати.

Використовуйте команду systemdinit system, щоб перевірити роботу служби:





Висновок підтвердили, що служба успішно запущена. Однак найкраще протестувати її запуск за допомогою запиту сторінки з Nginx.

Відкрийте сторінку Nginx за замовчуванням, щоб підтвердити роботу програмного забезпечення через IP-адресу вашого сервера. Якщо ви не знаєте IP-адресу вашого сервера, ви можете знайти його за допомогою інструменту icanhazip.com, який видасть вам ваш публічний IP-адресу, отриманий від іншого розташування в Інтернеті:



Коли ви дізнаєтеся IP-адреса вашого сервера, введіть його в адресний рядок браузера:



Ви повинні отримати початкову сторінку Nginx за замовчуванням:



Якщо ви перебуваєте на цій сторінці, ваш сервер працює коректно і готовий до управління.

Крок 4 - Налаштування блоків сервера

При використанні веб-сервера Nginx ви можете використовувати блоки сервера (аналогічні віртуальним хостам в Apache) для інкапсуляції даних конфігурації та розміщення на одному сервері декількох доменів. Ми створимо домен your\_domain , але ви повинні замінити це ім'я власним доменним ім'ям . Додаткову інформацію про налаштування доменного імені за допомогою платформи DigitalOcean можна знайти в нашій статті Введення в DigitalOcean DNS .

У Nginx на Ubuntu 20.04 за замовчуванням включений один серверний блок, налаштований для виведення документів з директорії /var/www/html. Хоча це добре працює для окремого сайту, при хостингу декількох сайтів це незручно. Замість зміни /var/www/htmlми створимо в /var/wwwструктуру директорій для нашого сайту your\_domain , залишивши /var/www/htmlяк директорію за замовчуванням для виведення в разі, якщо запит клієнта не відповідають ніякі інші сайти.

Створіть директорію для your\_domain наступним чином, використовуючи прапор -pдля створення необхідних батьківських директорій:



Потім призначте володіння Директорією за допомогою змінної середовища $USER:



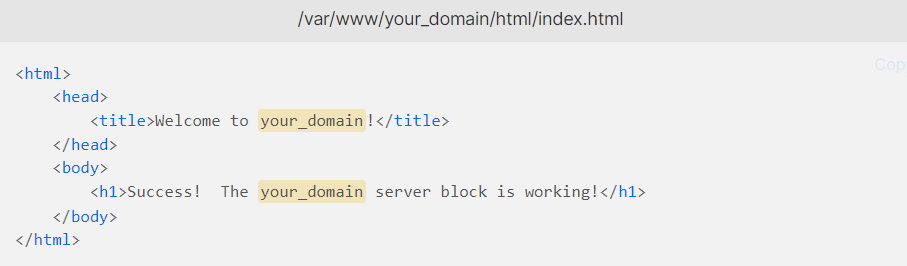
Дозволи кореневих директорій веб-сервера повинні бути правильними, якщо ви не змінювали значення unmask, яке встановлює дозволу файлу за замовчуванням. Щоб переконатися, що дозволу коректні, дозволити власнику читати, писати і запускати файли, а групам і іншим користувачам дозволити тільки читати і запускати файли, ви можете ввести наступну команду:



Потім створіть в якості прикладу сторінку index.html, використовуючи nanoабо свій улюблений редактор:



Додайте в сторінку наступний зразок коду HTML:



Збережіть і закрийте файл за допомогою введення CTRLі X, а потім Yі ENTERпісля завершення роботи.

Для обслуговування цього контенту Nginx необхідно створити серверний блок з правильними директивами. Замість того щоб змінювати файл конфігурації за замовчуванням безпосередньо, ми створимо новий файл в директорії :/etc/nginx/sites-available/your\_domain



Введіть наступний блок конфігурації, який схожий на заданий за замовчуванням, але оновлений з урахуванням нової директорії і доменного імені:



Ми оновили конфігурацію rootіз зазначенням новий каталог і замінили server\_nameна ім'я нашого домену.

Тепер ми активуємо файл, створивши посилання в директорії sites-enabled, який Nginx зчитує при запуску:



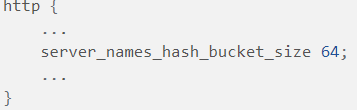
Два серверних блоку активовані і налаштовані для реагування на запити на основі директив listenі server\_name(додаткову інформацію про обробку Nginx цих директив можна знайти тут ):

* your\_domain: Буде відповідати на запити your\_domainі www.your\_domain.
* default: Буде відповідати на будь-які запити порту 80, що не відповідають двом іншим блокам.

Щоб уникнути можливої проблеми з хешем пам'яті при додаванні додаткових імен серверів, необхідно змінити одне значення у файлі /etc/nginx/nginx.conf. Відкрийте файл:



Знайдіть директиву server\_names\_hash\_bucket\_sizeі видаліть символ #, щоб розкоментувати рядок. Якщо ви використовуєте nano, ви можете швидко знайти слова в файлі, натиснувши CTRLі w.



Збережіть файл і закрийте його після завершення.

Проведіть тестування, щоб переконатися у відсутності помилок синтаксису в файлах Nginx:



Якщо проблеми відсутні, перезапустіть Nginx, щоб активувати зміни:



Тепер Nginx повинен обслуговувати ваше доменне ім'я. Ви можете перевірити це, відкривши в браузері адресу http://example.com, після чого повинні побачити приблизно наступне:



Встановлення та налаштовувати IIS

У більшості випадків в реальних виробничих проектах один або кілька серверів будуть використовуватися для обслуговування клієнтських запитів веб-сайту. Ці сервери можуть належати і управлятися безпосередньо вами, спеціалізованої командою або ж сторонньою компанією, що надає послуги хостингу. У будь-якому випадку рано чи пізно настає момент, коли написання коду і його тестування завершено, і робота повинна бути представлена ​​широкій публіці - в цьому і полягає розгортання веб-сайту.

У цій та наступних статтях ми розглянемо різні варіанти розгортання. Однак у всіх випадках основні передумови залишаються незмінними. На робочій станції є готовий веб-сайт, який потрібно розгорнути на сервері, щоб він був доступний клієнтам. Для ASP.NET таким сервером є IIS (Internet Information Services - інформаційні служби Інтернету) , а його поточною версією - IIS 8. Коли він був вперше реалізований, IIS був базовий веб-сервер. З роками IIS розвинувся в складний сервер додатків, що надає широке безліч функціональних засобів, найбільш важливим з яких є підтримка хостингу додатків ASP.NET.

У цій статті основну увагу приділяється IIS 8. Хоча машина, на якій запущений IIS 8, тут називається сервером, IIS можна запускати під керуванням версій Windows як для робочої станції, так і для сервера. На робочих станціях доступні не всі, але більшість функціональних можливостей, що дозволяє розміщувати складні веб-сайти. По можливості ми рекомендуємо використовувати Windows Server, однак недорогою альтернативою можуть послужити Windows 7 або Windows 8.

У Microsoft прив'язують випуски IIS з випусками Windows. До складу Windows Server 2008 і Windows Vista входить версія IIS 7.0, до складу Windows Server 2008 R2 і Windows 7 - версія IIS 7.5, а до складу Windows Server 2012 і Windows 8 - IIS 8. Версії - 7.0 і 7.5 - в Microsoft узагальнено називають IIS 7, що може вносити плутанину. Версію IIS, підтримувану операційною системою, змінити не можна - Windows Server 2008 буде використовувати тільки IIS 7.0. Наприклад, модернізувати її до версії IIS 7.5, використовуваної в Windows Server 2008 R2, не вийде.

установка IIS

Компонент IIS включений як частина установки Windows (як для сервера, так і для робочих станцій) і вимагає активізації і конфігурації. Далі представлені три способи активації IIS для різних операційних систем.

Установка IIS на настільних версіях Windows (Windows Vista, Windows 7 і Windows 8)

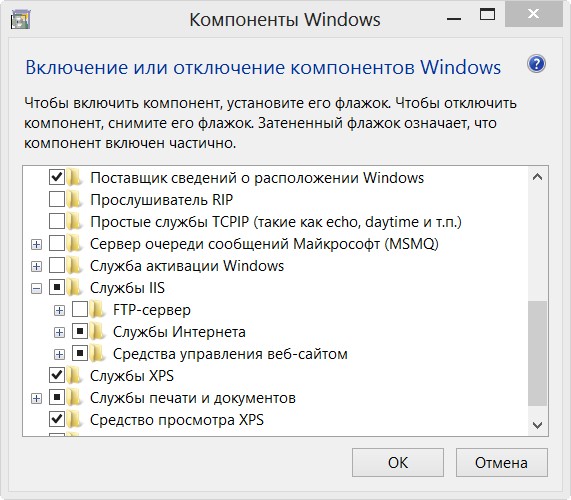
Кожна версія операційної системи Windows пропонує свою версію IIS - IIS 8 (в Windows 8), IIS 7.5 (в Windows 7) або IIS 7 (в Windows Vista). У всіх цих версіях Windows, IIS включений, але спочатку не встановлено. Щоб встановити його, необхідно виконати наступні дії:

Відкрийте панель управління.

Виберіть "Програми".

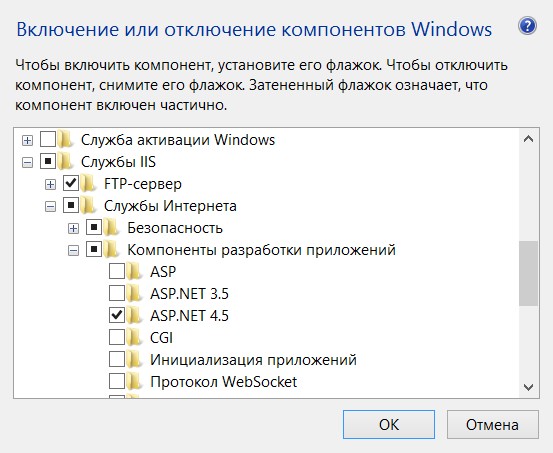
Натисніть кнопку "Включення або відключення компонентів Windows". Тепер вам потрібно почекати, поки Windows досліджує вашу систему.

Знайдіть елемент Internet Information Services (Служби IIS) у верхній частині списку і натисніть на галочку щоб включити його:



Зверніть увагу, що Windows дозволяє включити безліч компонентів IIS: підтримка FTP-сервера, додаткові інструменти управління, служби забезпечення сумісності з IIS 6 і т.д.

Переконайтеся, що ви вибрали підтримку ASP.NET. Для цього розкрийте вузол Служби Інтернету -> Компоненти розробки додатків -> ASP.NET (Internet Information Services -> World Wide Web Services -> Application Development Features -> ASP.NET):



Якщо ви хочете використовувати підтримку IIS в Visual Studio, яка дозволяє вам створювати віртуальні каталоги IIS безпосередньо в діалоговому вікні New Web Site, вам потрібно вибрати пункт «Сумісність управління IIS 6» в розділі «Засоби управління веб-сайтом» (Web Management Tools - > IIS 6 Management Compatibility).

Як тільки ви вибрали потрібні параметри IIS, натисніть кнопку OK для завершення установки.

Установка IIS в Windows Server 2012

Процес установки IIS в Windows Server 2012, по суті, такий же, як і в Windows Server 2008. Основна відмінність полягає в тому, що призначений для користувача інтерфейс дещо відрізняється. Детальний опис ви можете знайти перейшовши за посиланням Installing IIS 8 on Windows Server 2012 .

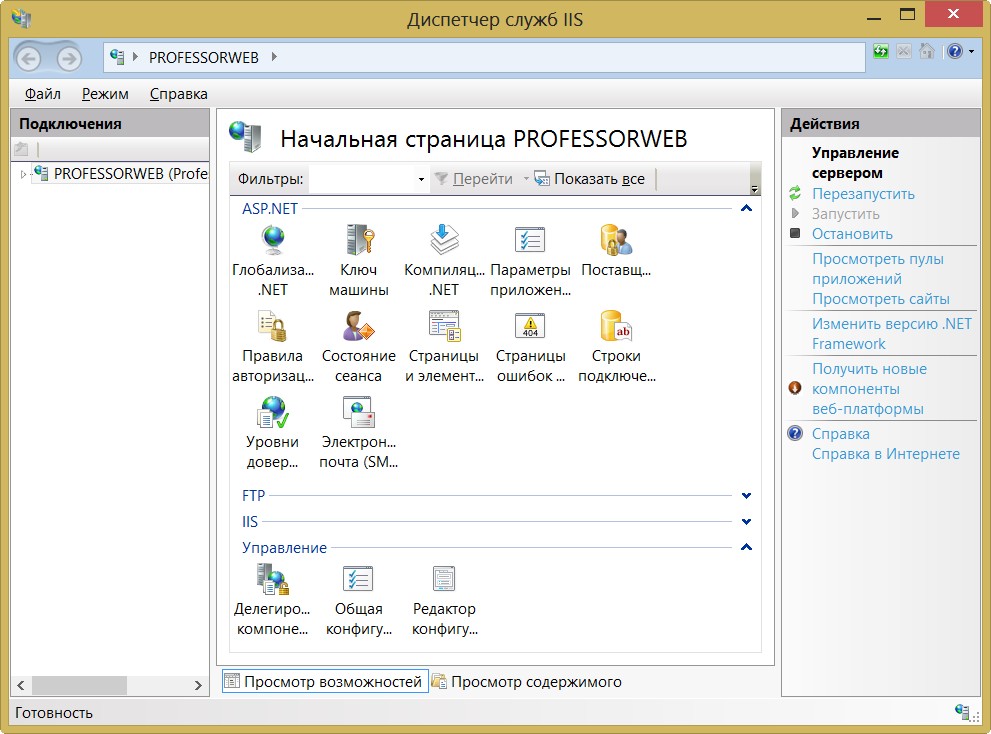
управління IIS

При установці IIS, він автоматично створює каталог з ім'ям C: \ inetpub \ wwwroot, який представляє ваш веб-сайт. Всі файли в цьому каталозі буде відображатися, як ніби вони знаходяться в кореневому каталозі вашого веб-сервера.

Щоб додати додаткові сторінки на ваш веб-сервер, можна скопіювати файли HTML, ASP або ASP.NET безпосередньо в каталог C: \ Inetpub \ wwwroot. Наприклад якщо додати файл TestFile.html в цей каталог, ви можете запитати його в браузері через URL-адресу http: //localhost/TestFile.html. Ви навіть можете створювати вкладені папки для групування пов'язаних ресурсів. Наприклад, ви можете отримати доступ до C: \ inetpub \ wwwroot \ MySite \ MyFile.html через браузер, використовуючи URL-адресу http: //localhost/MySite/MyFile.html.

Каталог wwwroot зручний для запуску простих прикладів і статичних сторінок. Для правильного використання ASP.NET ви повинні зробити свій власний віртуальний каталог для кожного веб-додатки, яке ви створюєте. Наприклад, ви можете створити папку з будь-яким ім'ям на будь-якому диску вашого комп'ютера і помістити її в віртуальний каталог IIS ніби вона розташована в каталозі C: ​​\ inetpub \ wwwroot.

Перш ніж почати роботу, вам потрібно запустити диспетчер служб IIS. Його можна знайти в меню Start (Пуск). Конкретне розташування може залежати від використовуваної версії Windows (IIS -> Диспетчер служб IIS). Ярлик програми буде розташовуватися в розділі Programs (Програми) або Administrative Tools (Адміністрування). Початкова сторінка IIS Manager показана на малюнку нижче:



Тепер потрібно ознайомитися з рядом термінів, які використовуються в IIS. У лівій частині вікна IIS Manager відображається запис з ім'ям використовуваного сервера. Наш сервер має ім'я PROFESSORWEB, сгенерированное за замовчуванням Windows 8, яка буде використовуватися в більшості прикладів. У центральній області відображається уявлення сервера. Це уявлення відображає набір значків, які дозволяють конфігурувати параметри сервера. У правій частині екрана розташований список доступних дій. Наприклад, в цьому поданні можна запускати, зупиняти і перезапускати сервер.

Якщо розгорнути елемент сервера в деревовидному поданні в лівій частині екрана, відобразиться елемент Sites (Сайти), що містить єдину запис Default Web Site (Веб-сайт за замовчуванням). Сайт - це колекція файлів і каталогів, що утворюють веб-сайт. На одному сервері IIS може підтримувати кілька сайтів, як правило, на різних портах TCP / IP (за замовчуванням використовується порт 80). Поєднання імені сервера і порту сайту утворює першу частину URL-адреси. Наприклад, при використанні сервера mywebserver з сайтом, підключеним до порту 80, URL-адресу виглядає наступним чином:



Кожен сайт може містити безліч файлів і каталогів. Кожен з них утворює частина URL-адреси. Так, URL-адресу статичної сторінки mypage.html, розташованої в каталозі myfiles, буде наступним:



У деяких ситуаціях ім'я, під яким сервер відомий вам, і ім'я, яке клієнти використовують для отримання вмісту, будуть відрізнятися. Ми залишимо цей нюанс без уваги, але адміністратор сервера або компанія, що надає послуги хостингу, нададуть необхідні відомості, якщо це важливо для конкретного сервера.

Щоб перевірити працездатність IIS виберіть Default Web Site і в правій області диспетчера служб IIS виберіть пункт "Запустити". Після цього натисніть кнопку "Огляд \* .80 (http)" щоб відкрити сторінку сайту в браузері:



Як бачите, в моєму випадку я поміняв порт буде використовуватися під (з 80 на 8080). Я зробив це, тому що на 80-му у мене запущений локальний Apache-сервер. Якщо у вас виникає така ж проблема, то змінити порт можна клацнувши правою кнопкою миші по сайту (Default Web Site) і вибравши в контекстному меню "Змінити прив'язки" (Bindings). Після цього в діалоговому вікні можна змінити порт, який використовується за умовчанням.

Отже, кожен сервер може підтримувати безліч сайтів, кожен з яких працює на іншому порту або з іншим IP-адресою. Кожен сайт може мати безліч файлів і каталогів, і поєднання цих елементів надає інформацію про URL-адресу. Ми повернемося до URL-адресами та використання IIS Manager при розгляді кожного з підходів до розгортання.