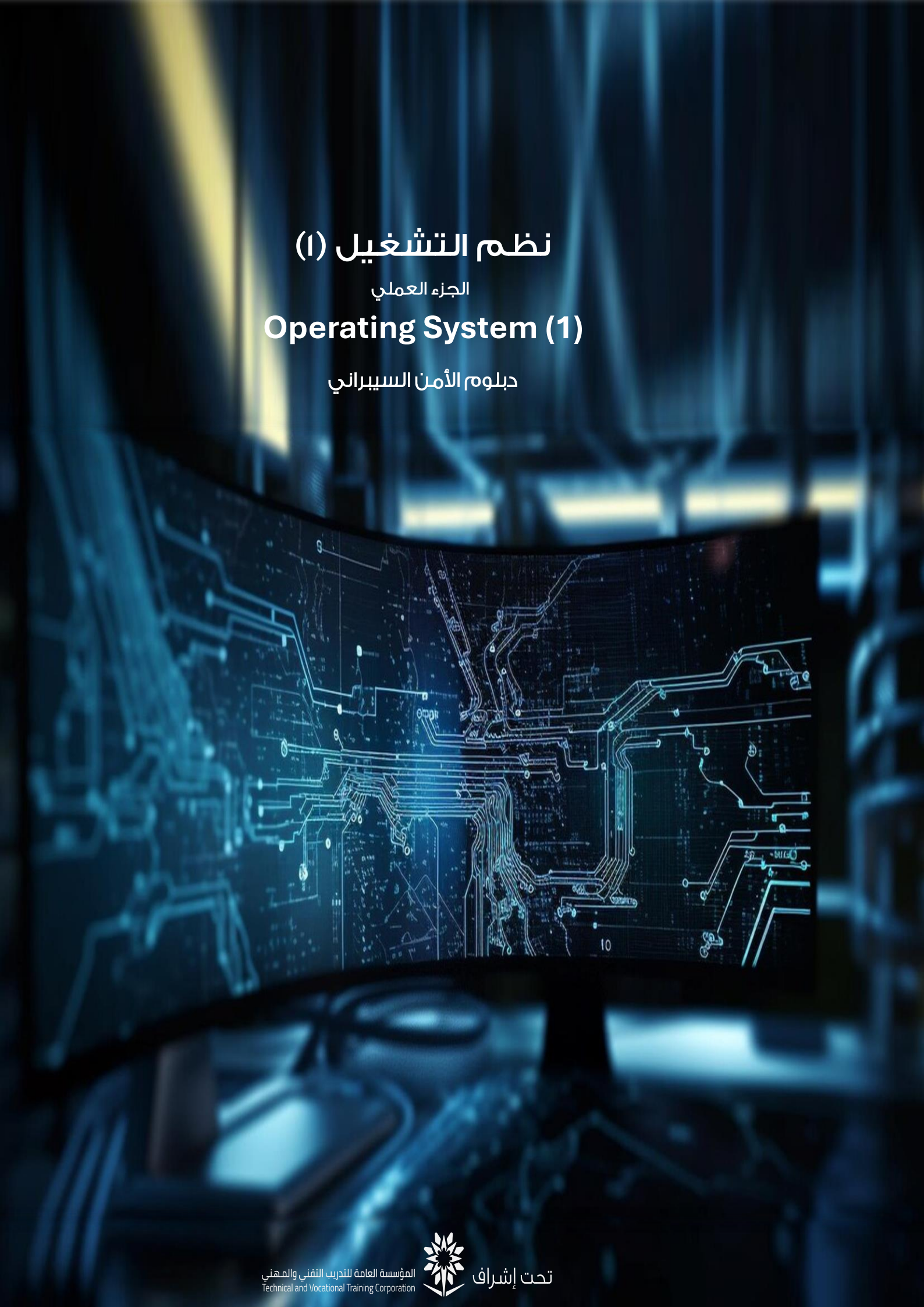
نظم التشغيل (١)

الجزء العملي

Operating System (1)

دبلوم الأمن السيبراني



تحت إشراف
المؤسسة العامة للتدريب التقني والمهني
Technical and Vocational Training Corporation

❖ الأهداف التفصيلية للمقرر

- بنهاية هذا المقرر سيكون المتدرب قادراً وبكفاءة على أن:
- ينصب نظام تشغيل الشبكات Windows Server 2012 R2
- ينصب المتحكم بالمجال Controller Domain
- ينشئ حسابات المستخدمين بالمجموعات والحواسيب
- يدير حسابات المستخدمين بالمجموعات والحواسيب
- يكون الإعدادات اللازمة على IPv4
- يتعامل مع كل من موجه الأوامر و PowerShell لإدارة وتنفيذ العمليات في خدمات المجال للدليل النشط AD DS
- ينشئ التكوينات وإدارتها في بروتوكول التكوين الديناميكي للمضيف DHCP وعمل التأمينات اللازمة
- ينصب نظام أسماء النطاقات DNS
- يدير نظام أسماء النطاقات DNS
- يطبق على نظام العنونة IPv6
- يطبق تكنولوجيا التخزين المحلي
- يؤمن مشاركة الملفات والمجلدات
- ينشئ ويدير نهج المجموعة
- يكون إعدادات زيادة الأمان على مصادر الخادم وتكوين الجدار الناري
- يستخدم Hyper-v ويدير الشبكات والتخزين في الأنظمة الوهمية
- يعرف البرمجة ومستوياتها
- يذكر تعريف لغة C++ واستخداماتها ومميزاتها وعناصرها
- يذكر تعريف برنامج Visual studio ومميزاته
- يتعرف على طريقة تشغيل برنامج Visual studio c++
- يستوعب الهيكلة العامة لبرنامج بلغة c++
- يلم بمتغيرات c++ الثابتة وأنواعها وشروط تسميتها والاعلام عنها
- يتعامل مع المتغيرات الثابتة بواسطة لغة c++
- يستكشف عملية إسناد القيم للمتغيرات بلغة c++
- يتعرف على دوال الإخراج بلغة c++
- يتعرف على دوال الادخال بلغة c++
- يتعامل مع العمليات الحسابية وطرق تمثيلها وأولوياتها
- يتعرف على عبارة if الشرطية الاعتيادية
- يتعامل على عبارة if-else الشرطية
- يستوعب العبارات الشرطية
- يتعامل مع عبارة if-else if الشرطية
- يبني برنامج بواسطة عبارة الاختيار المتعدد Switch
- يوضح ماهي for-loop التكرارية الاعتيادية
- يعمل على for-loop التكرارية المتداخلة
- يتعرف على عبارة break
- يعمل على عبارة while التكرارية
- يتعرف على عبارة do – while التكرارية
- يفرق بين دوال التكرار for / while/do-while

الفهرس

الفصل الأول :نظام تشغيل الشبكاتwindows server2012r

خطوات تثبيت نسخة السيرفر ٢٠١٢ ..... ٢

الفصل الثاني :خدمات المجال للدليل النشط

تنصيب المتحكم بالمجال domain control..... ١١

إعداد خدمة DNS..... ١٥

الفصل الثالث :إدارة خدمات المجال للدليل النشط

إدارة حسابات الحواسيب ١٩

إدارة تفويض الصلاحيات..... ٣٠

الفصل الرابع: كيفية جعل خدمات المجال للدليل النشط تعمل بشكل الي

استخدام موجه الأوامر لأداره خدمات المجال للدليل النشط ٣٧

استخدام PowerShellلإدارة خدمات المجال للدليل النشط ٤٣

اجراء العمليات باستخدام PowerShell..... ٤٨

الفصل الخامس :بروتوكولات التكوين الديناميكي للمضيف DHCP

لأنشاء وتكوين النطاق..... ٥٣

ادارة قاعدة البيانات DHCP..... ٥٧

تأمين ومتابعة DHCP..... ٥٩

الفصل السادس: نظام أسماء النطاقات DHCP

تنصيب وأداره خادم DNS..... ٦٤

أدارة مناطق DNS..... ٦٥

الفصل السابع :تكنولوجيا التخزين المحلي

أدارة الأقراص وأحجامها ٧١

التحكم في مساحة التخزين..... ٧٣

الفصل الثامن :خدمات الطباعة وتأمين الملفات

التشفير ٨١

حماية الملفات والمجلدات المشتركة بواسطة SHADOW COPIES..... ٨٦

تكوين خدمة مجلدات العمل..... ٨٨

تكوين الطباعة الشبكية ٩٢

الفصل التاسع: نهج المجموعة

معالجة نهج المجموعة ٩٩
تكوين المتجر المركزي لنهج المجموعة ١٠٥
الفصل العاشر :تأمين نظام تشغيل الشبكات Windows Server2012r
تعطيل تثبيت التطبيقات ١١٠
تكوين الجدار الناري في Windowsبأعدادات متقدمة ١١١
الفصل الحادي عشر : الأنظمة الوهمية و Hyper-v
تطبيق Hyper-v..... ١١٦
إدارة التخزين في الأنظمة الوهمية ١٢١
إدارة الشبكات الوهمية ١٢٦
الفصل الثاني عشر: مقدمة عن لغة ++c
تعريف البرمجة ومستوياتها:..... ١٣٠
تعريف لغة ++C:..... ١٣٠
مميزات لغة ++C:..... ١٣٢
استخدامات لغة ++C:..... ١٣٣
تعريف برنامج Visual Studio code ومميزاته:..... ١٣٣
خطوات تشغيل برنامج Visual Studio code:..... ١٣٤
الفصل الثالث عشر: التعرف على أساسيات لغة ++c
الهيكلة العامة للبرنامج:..... ١٣٧
التعليقات:..... ١٣٧
المتغيرات وشروط تسميتها:..... ١٣٨
انواع المتغيرات:..... ١٤٠
امر الاخراج Cout في لغة ++C:..... ١٤٤
امر الادخال Cin في لغة ++C:..... ١٤٧
تمثيل العلميات الرياضية :..... ١٥١
الفصل الرابع عشر: الجمل الشرطية والعبارات الشرطية (if – Switch)
الجمل الشرطية والعبارات الشرطية:..... ١٥٩
انواع جمل الشرط :..... ١٥٩
جملة الشرط if:..... ١٦٠

جملة الشرط else:.....١٦٢
جملة الشرط else-if:.....١٦٣
العوامل التي تستخدم في المقارنات:.....١٦٤
تعريف الجملة Switch:.....١٧٣
الفصل الخامس عشر: عبارات أو جمل التكرار
التكرار وحلقات التكرار:.....١٧٨
اسلوب التكرار باستعمال حلقة For:.....١٧٨
اسلوب التكرار باستعمال حلقة while:.....١٧٩
الفرق بين دوال التكرار(For , while , while-do):.....١٨٢
الفصل السادس عشر: تحديد الأخطاء الشائعة في البرمجة وتصحيحها
مفهوم معالجة الأخطاء:.....١٨٥
أمثلة على أنواع الاخطاء:.....١٨٧
المراجع.....١٨٩

مقدمة:

ويندوز سيرفر ٢٠١٢ R2 بالإنجليزيةWindows Server 2012 R2 : هو الإصدار السادس من نظام تشغيل خادم Windows Server من قبل مايكروسوفت، كجزء من عائلة أنظمة التشغيل ويندوز إن تي .تم الكشف عنه في ٣ يونيو ٢٠١٣ في TechEd أمريكا الشمالية، وتم إصداره في ١٨ أكتوبر ٢٠١٣، حيث انه برنامج سهل الاستخدام ومتعدد المهام.

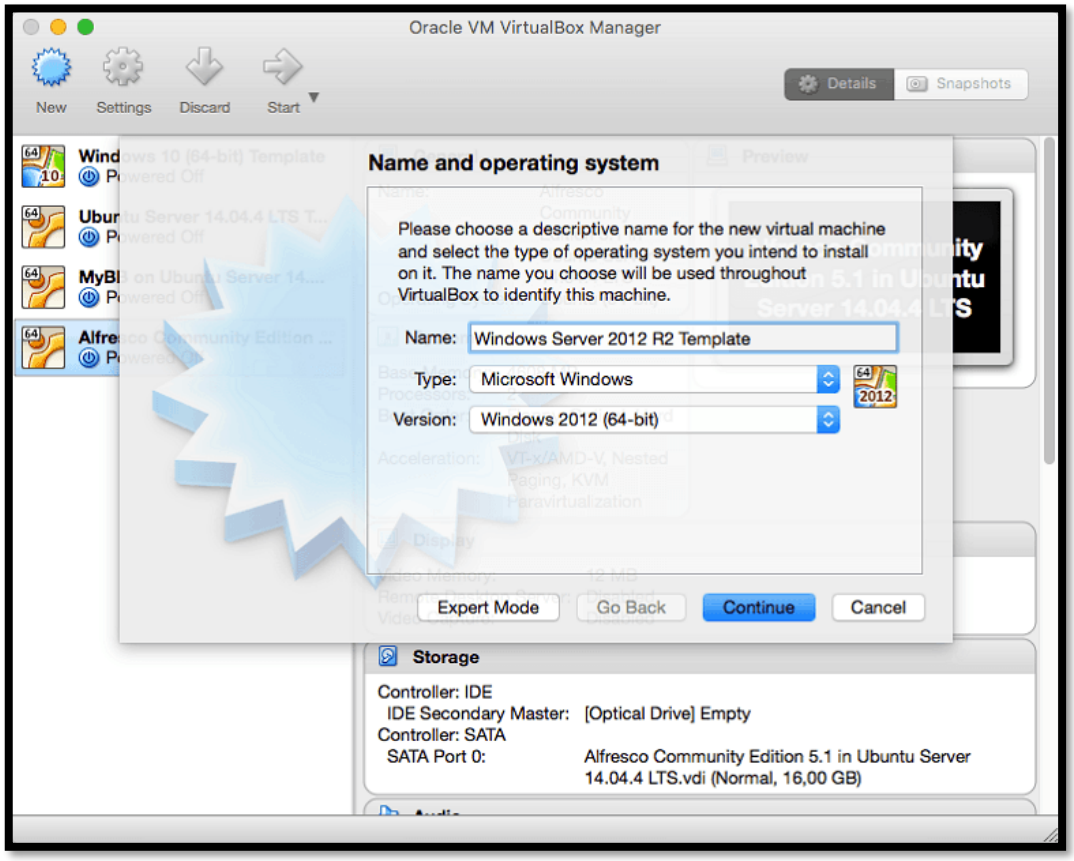
وايضا سوف نتحدث في هذه الملزمة عن احد لغات البرمجة العالية المستوى والاكثر استخدام في وقتنا الحالي وهي لغة C++ هي لغة سهلة الاستخدام وقريبة جدا من لغة الانسان وتمتاز ايضا بالسرعة والاداء العالي والتعدد الكائني.

الفصل الأوّل: نظام تشغيل الشبكات Windows Server 2012R

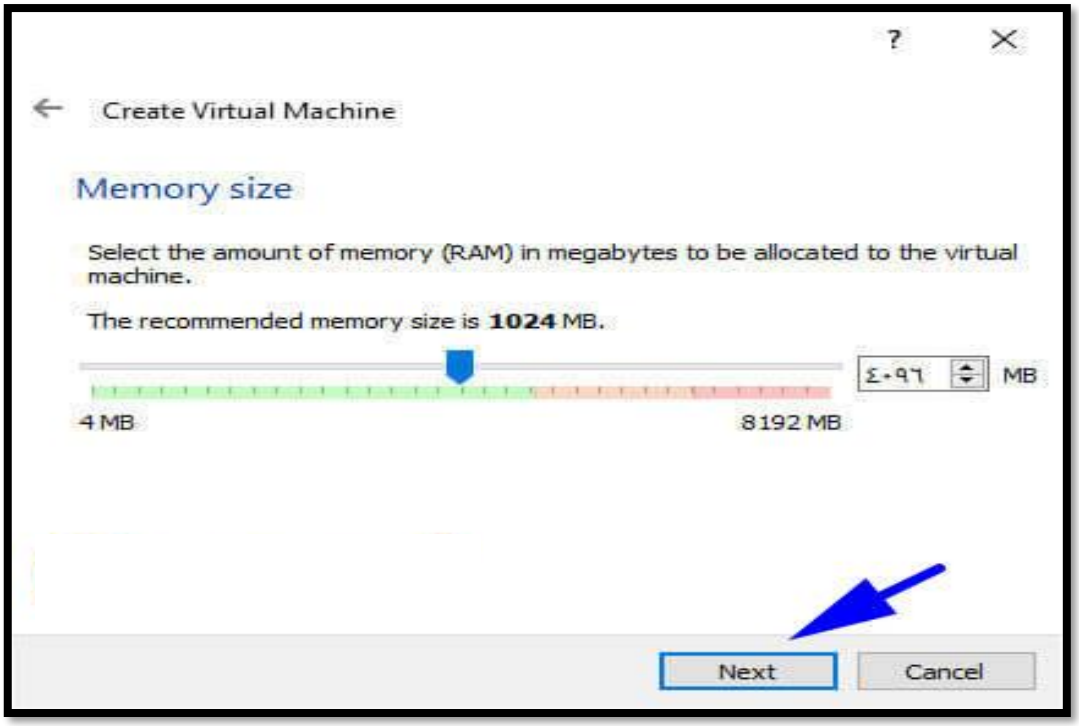
- **خطوات تثبيت نسخة السيرفر ٢٠١٢ على Oracle VM VirtualBox:**

١. افتح برنامج virtualbox وانقر على New

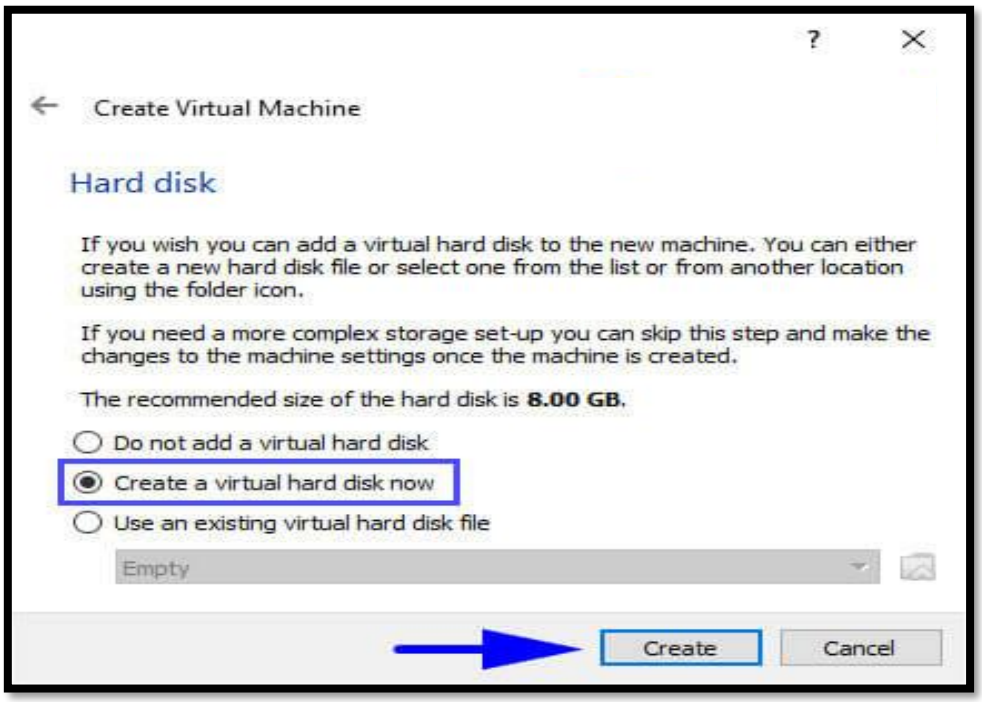
٢. الخانة الأولى أكتب اسم للنظام الخانة الثانية والثالثة اختار نوع النسخة



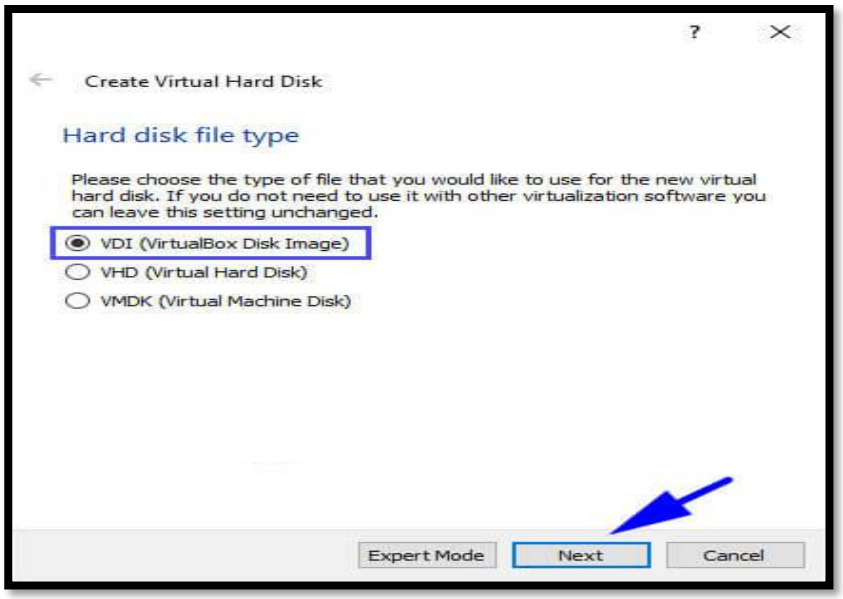
٣. اعطاء مساحة من الرام لاستخدامها على نظام لينكس، اذا كان حاسوبك يحتوي على رام 8000MB اختار 5000MB اما اذا كان لديك الرام 4000MB اختار 3000MB هذه المساحة تُستخدم اثناء تشغيل البرنامج فقط، عند اغلاقه ستُعاد المساحة المحددة للجهاز.



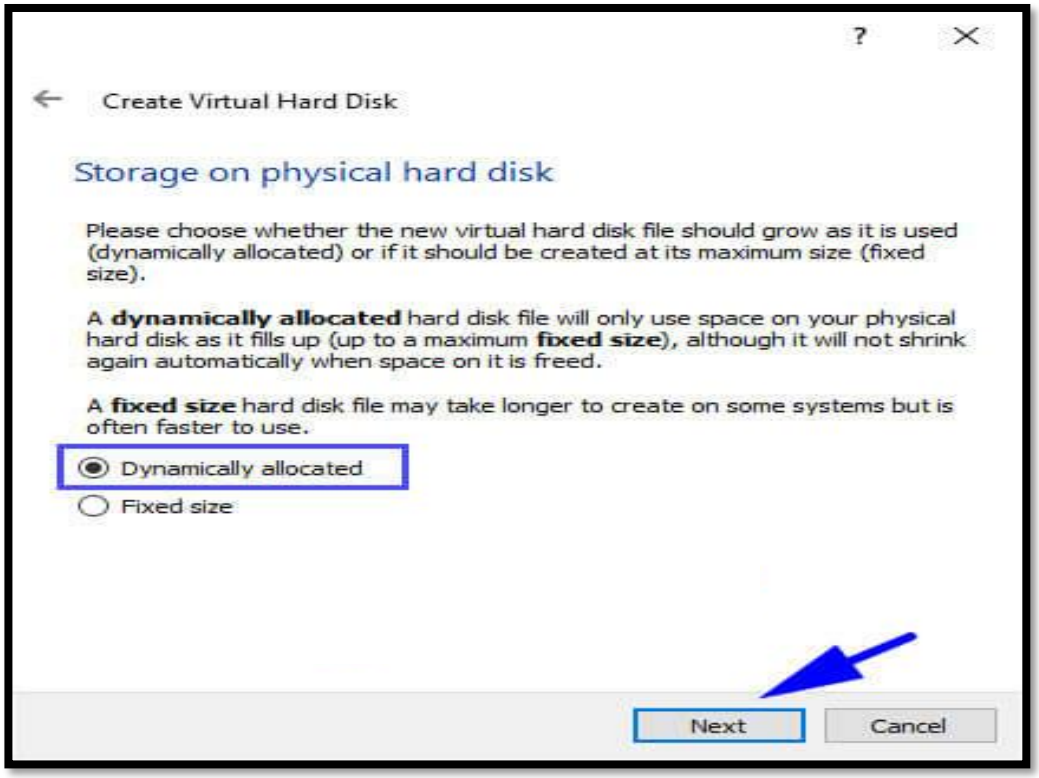
٤. تحديد على Create a virtual hard disk now



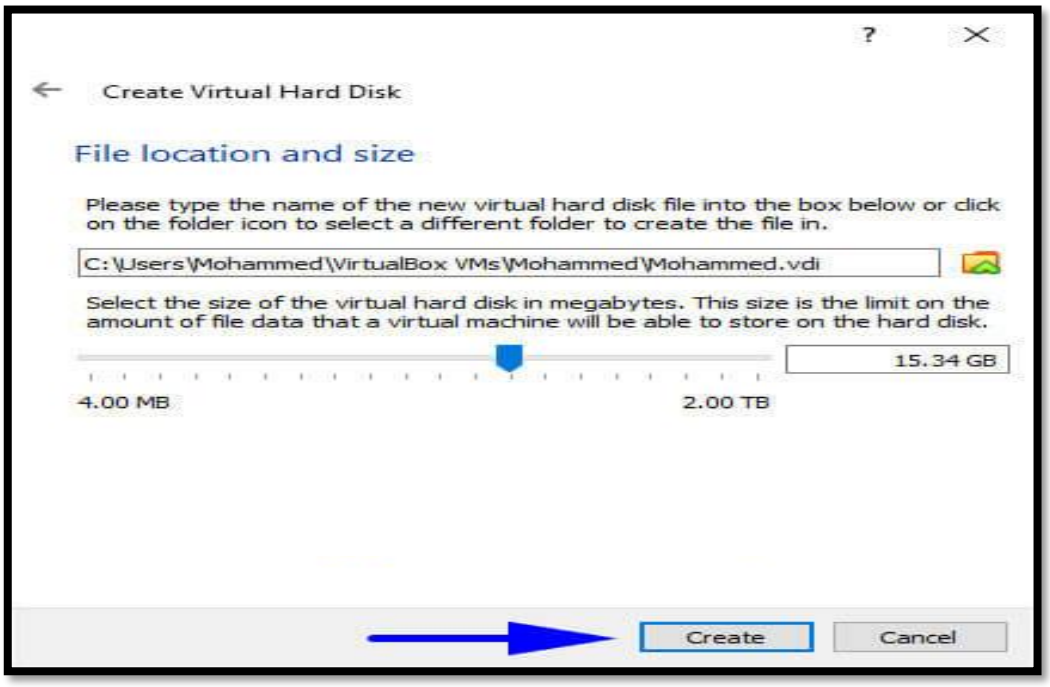
٥. حدد على الخيار VDI (VirtualBox Disk Image)



٦. اختار Dynamically allocated

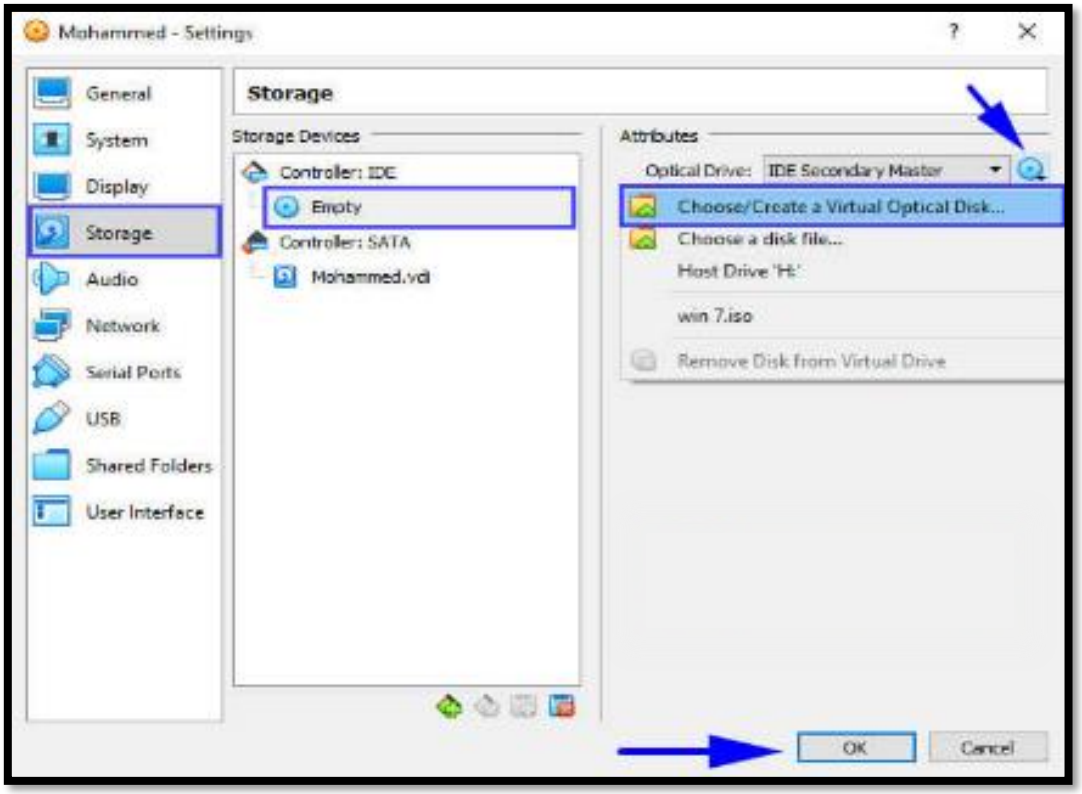


٧. اعطاء مساحة من الهارديسك لاستخدامها على نظام الويندوز سيرفر



٨. انقر بالزر الأيمن من الماوس على النظام الجديد ثم Settings

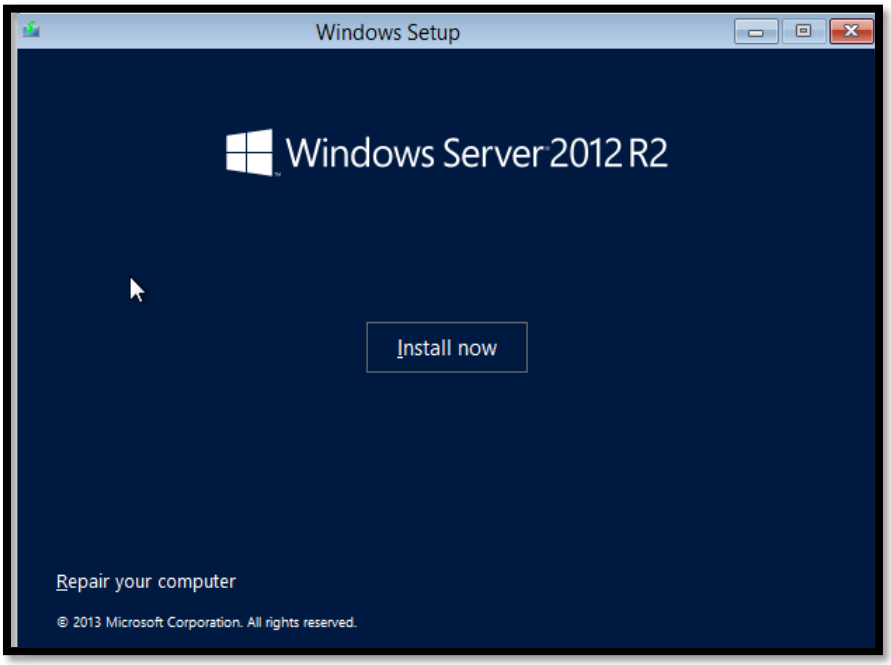
٩. توجه الى Storage ثم Empty ومن ثم Create a Virtual Optical Disk



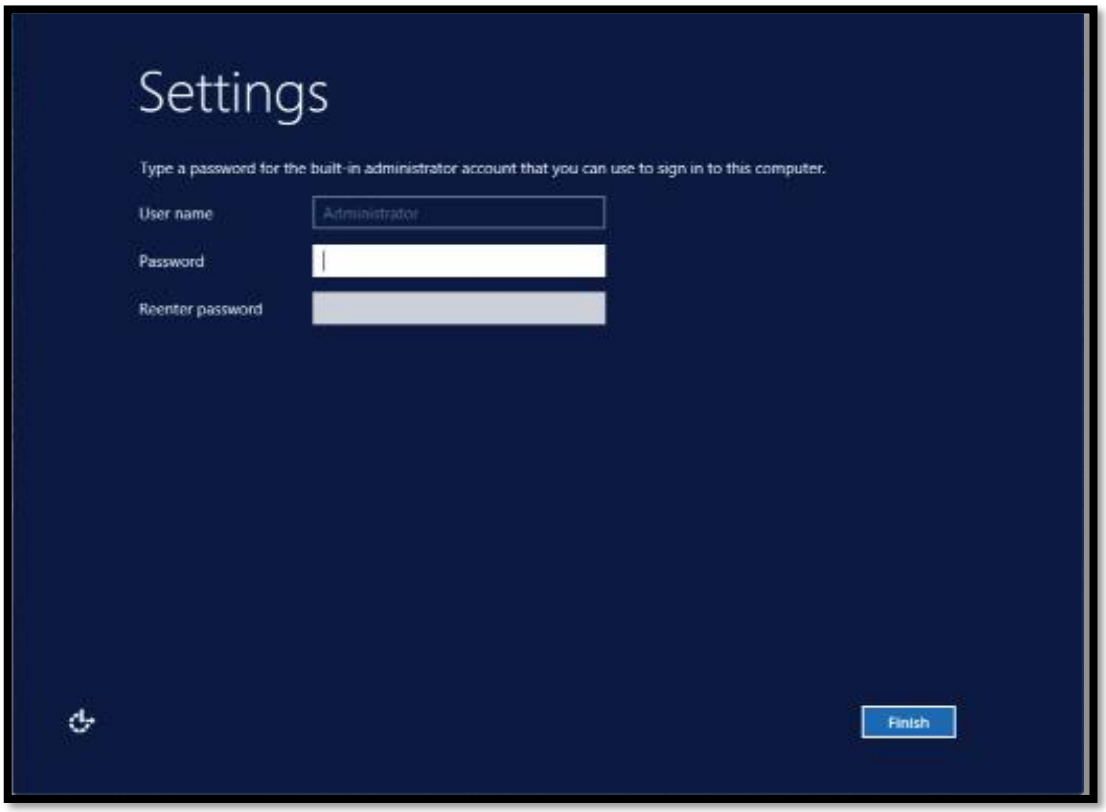
١٠. انقر على Add واختار نظام الويندوز المراد تثبيته.

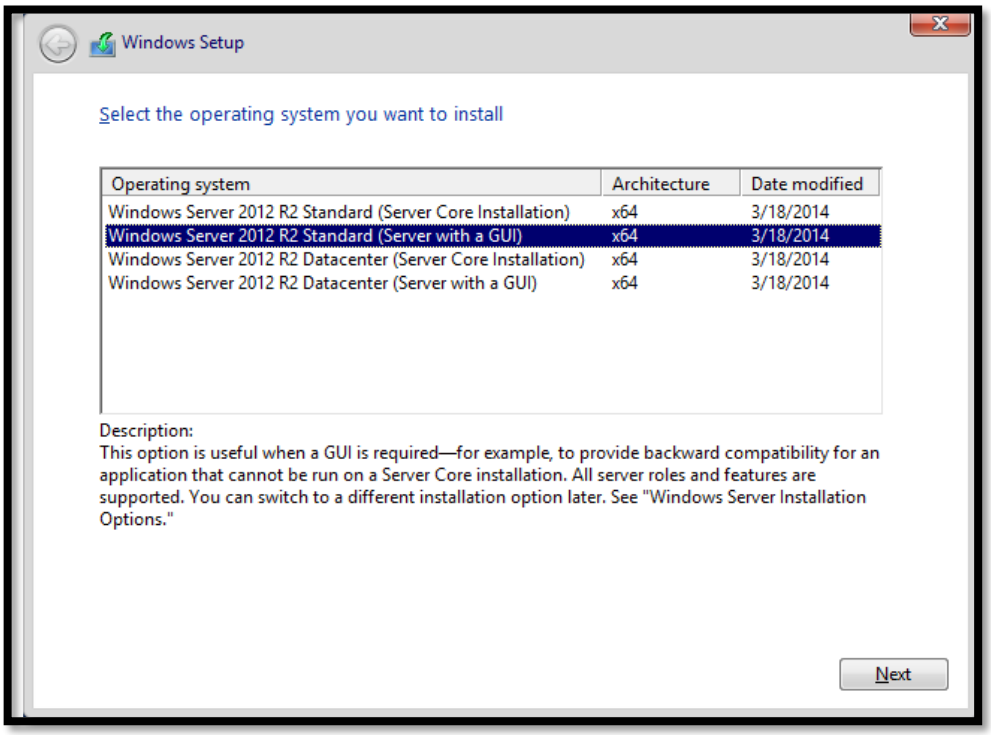
١١. انقر على Start

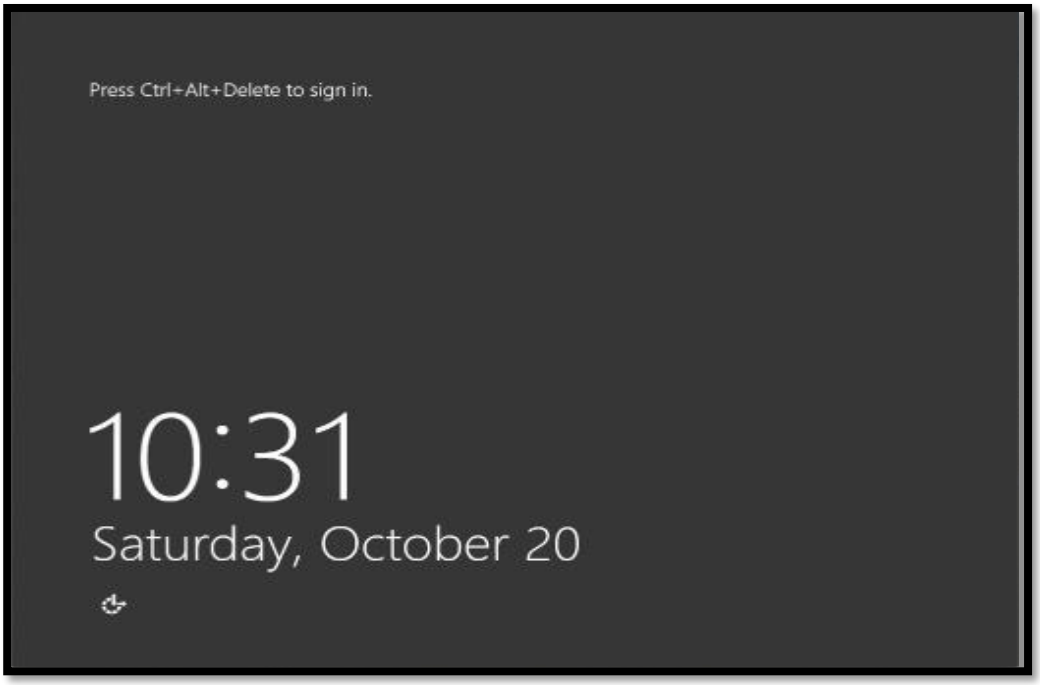
تم الانتهاء من إنشاء البيئة الوهمية، الآن نقوم بوضع نسخة الـ ISO في القرص، ليتم تنصيبه عل المكنة الوهمية، ونقوم بتشغيله لبدء التنصيب، قم باتباع الخطوات في الشاشات التالية لتنصيب النظام :



Windows Setup
Windows Server 2012 R2
Language to install: English (United States)
Time and currency format: English (United States)
Keyboard or input method: US
Enter your language and other preferences and click "Next" to continue.
© 2013 Microsoft Corporation. All rights reserved.
Next


Settings
Type a password for the built-in administrator account that you can use to sign in to this computer.
User name
Administrator
Password
Reenter password
Finish


Windows Setup
Select the operating system you want to install
Operating system
Architecture
Date modified
Windows Server 2012 R2 Standard (Server Core Installation)
x64
3/18/2014
Windows Server 2012 R2 Standard (Server with a GUI)
x64
3/18/2014
Windows Server 2012 R2 Datacenter (Server Core Installation)
x64
3/18/2014
Windows Server 2012 R2 Datacenter (Server with a GUI)
x64
3/18/2014
Description:
This option is useful when a GUI is required—for example, to provide backward compatibility for an application that cannot be run on a Server Core installation. All server roles and features are supported. You can switch to a different installation option later. See "Windows Server Installation Options."
Next


Press Ctrl+Alt+Delete to sign in.
10:31
Saturday, October 20

تمرين:

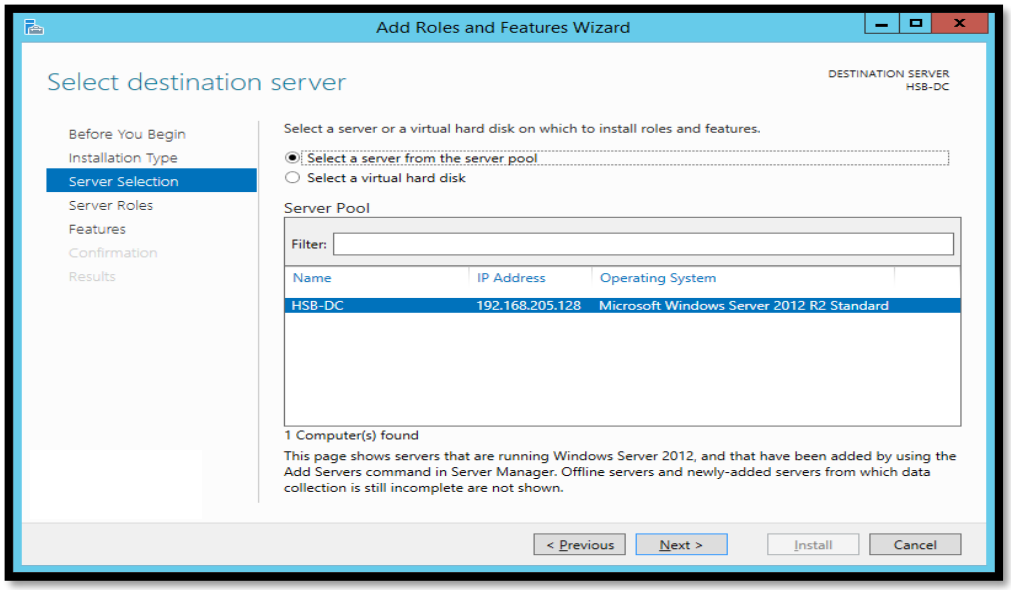
قم بتثبيت نسخة السيرفر ٢٠١٢ على Oracle VM VirtualBox متبع الخطوات التي قمت بدراستها؟

الفصل الثاني: خدمات المجال للدليل النشط AD DS

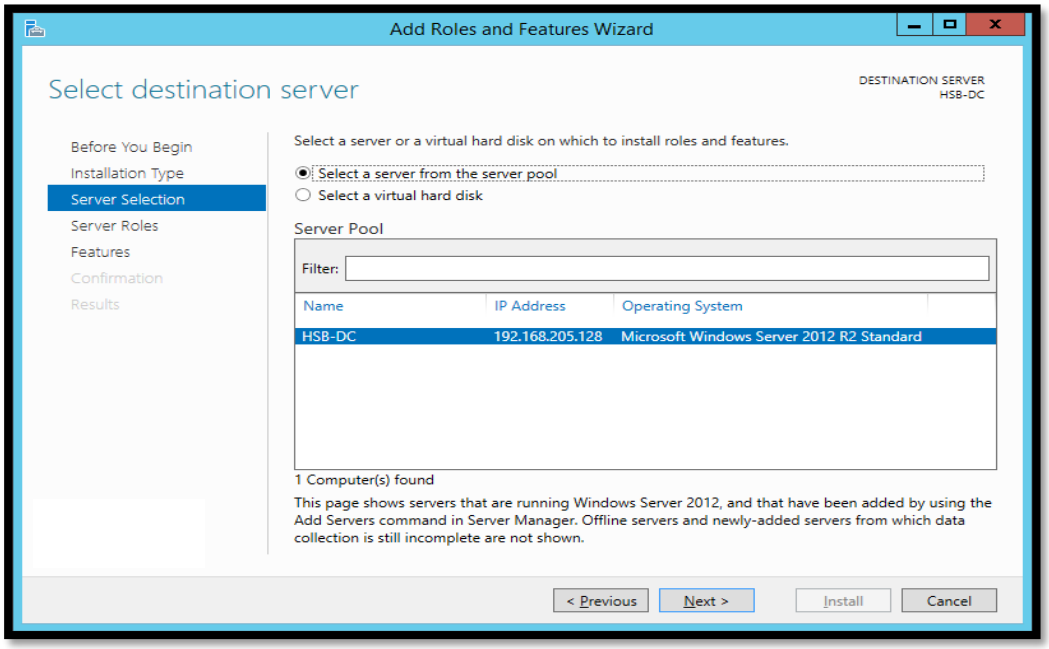
❖ تنصيب المتحكم بالمجال Domain Controller

شغّل برنامج مدير الخادم Server Manager من صفحة البداية، حيث تجده في القسم الأيسر منها، أو يمكنك أن تلاحظ وجوده بسهولة بجوار زر إبدا مباشرةً:

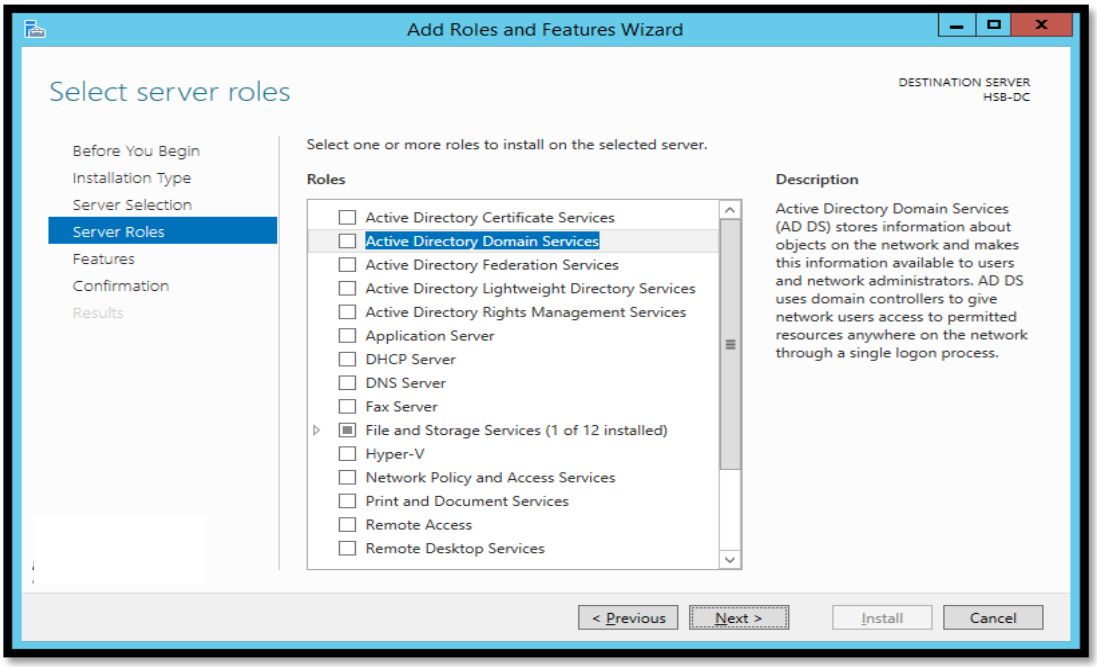
بعد أن تظهر نافذة برنامج مدير الخادم، انقر على القائمة Manage في الأعلى واختر Add Roles and Features لإضافة مزايا جديدة إلى ويندوز، ستظهر النافذة التالية:



انقر زر Next لتنتقل إلى نافذة أخرى تسمح لنا باختيار نمط التنصيب Select installation type سنترك الإعدادات الافتراضيّة كما هي، انقر Next لتظهر النافذة التي تحدّد الخادم الذي نريد التنصيب عليه:



يمكننا في الواقع اختيار خادم آخر لإضافة المزايا المطلوبة إليه، في حالتنا هذه لايوجد سوى خادمنا الحالي لذلك فانقر Next ستظهر نافذة تحتوي على جميع الأدوار Roles المتاحة التي من الممكن يأخذها هذا الخادم :

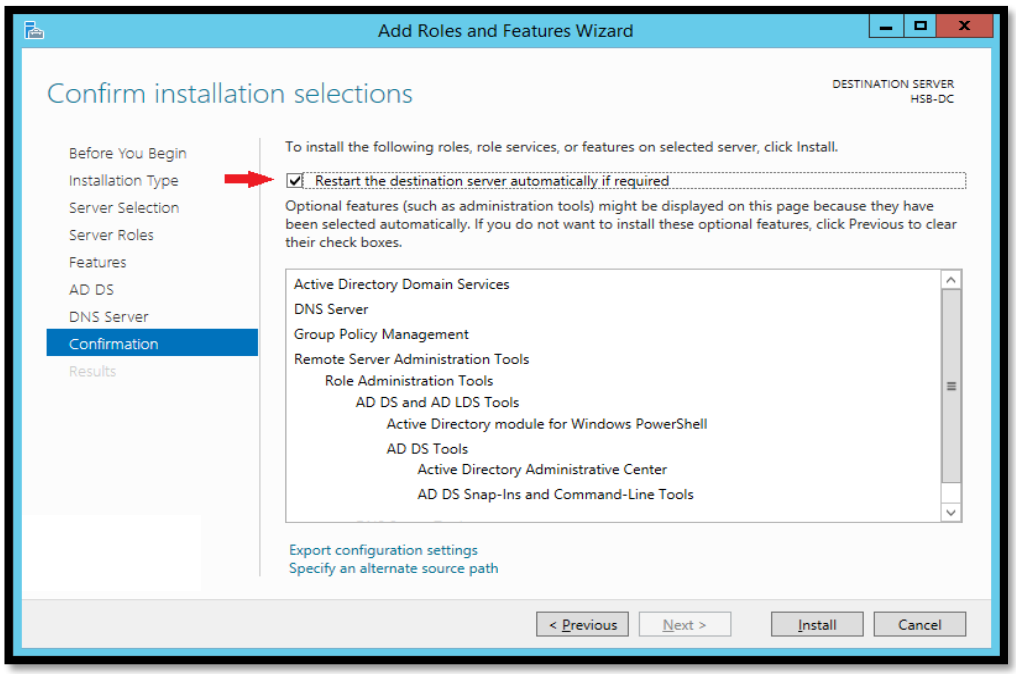


اختر من هذه الأدوار ما يلي:

Active Directory Domain Services

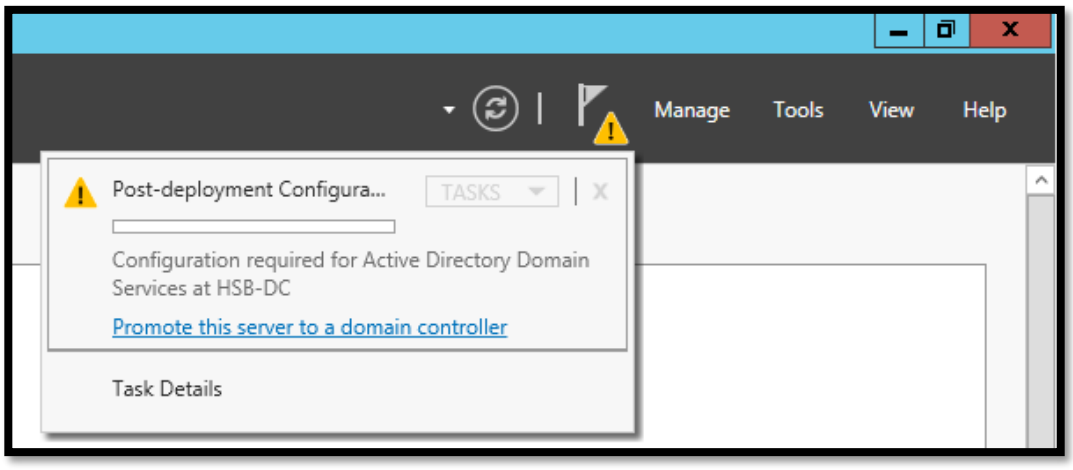
DNS Server

انقر Next لتظهر النافذة الأخيرة التي تسبق عمليّة التنصيب:

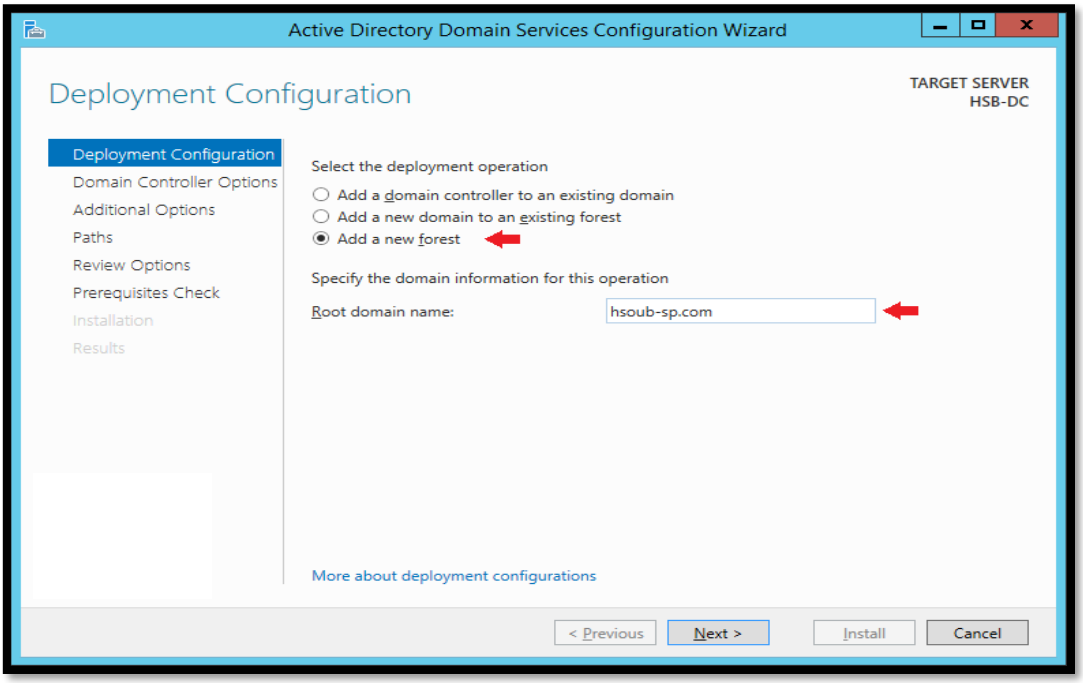


اختر صندوق الاختيار Restart the destination server automatically if required واختر Yesفي حال ظهرت رسالة تأكيد، يسمح هذا الاختيار بإعادة تشغيل الخادم بشكل تلقائي إذا تطلّب الأمر ذلك.

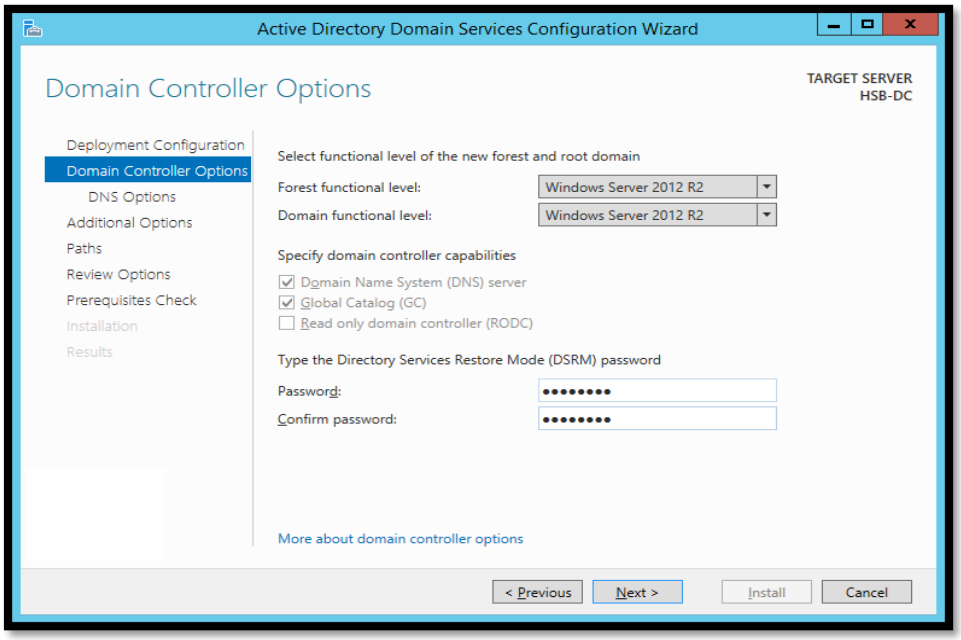
نقر زر Install لتبدأ عمليّة التنصيب، هذه المرّة ستلاحظ وجود إشارة تعجّب ضمن مثلث أصفر صغير، انقر فوقها لتظهر قائمة صغيرة كما في الشكل التالي:



انقر فوق Promote this server to a domain controller لترقية هذا الخادم وتحويله إلى متحكّم بالمجال اختر الخيار الأخير، ثمّ اكتب الاسم hsoub-sp.com لاسم المجال Root domain name كما يظهر في الشكل التالي تماماً، ثمّ انقر بعد ذلك الزر Next.

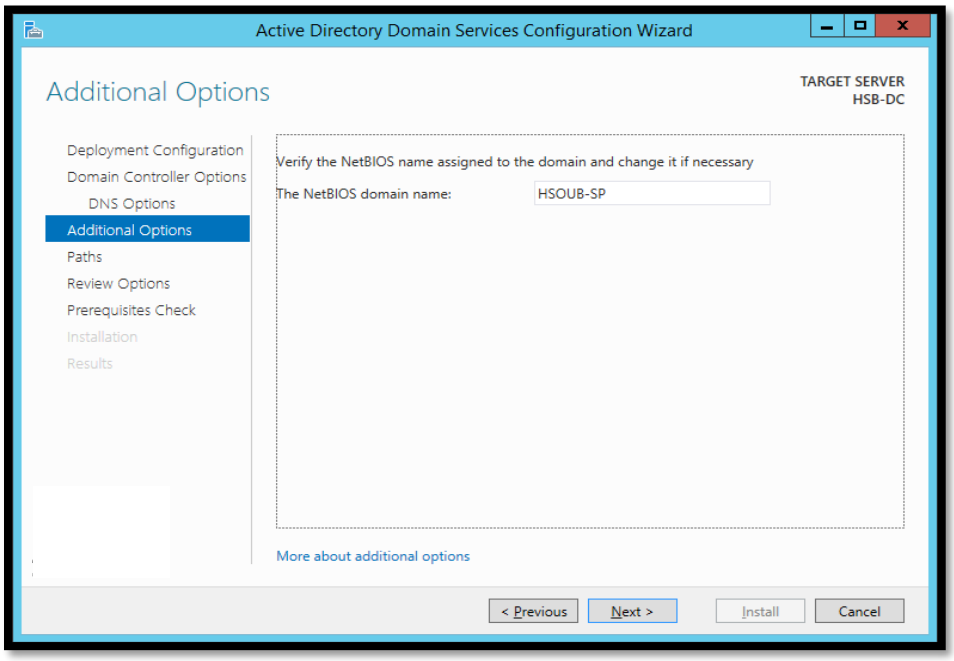


بعد ذلك ستظهر النافذة التالية، التي تسمح لنا بوضع كلمة مرور خاصّة بطور الاستعادة restore mode اكتب كلمة المرور التي ترغبها ثم انقر Next

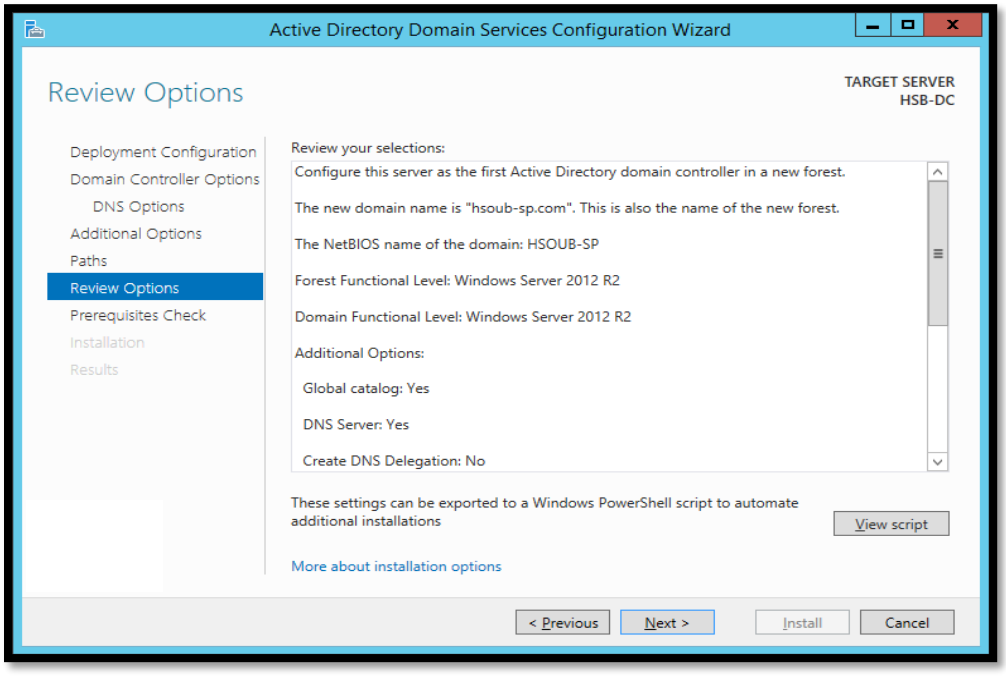


- **إعداد خدمة DNS**

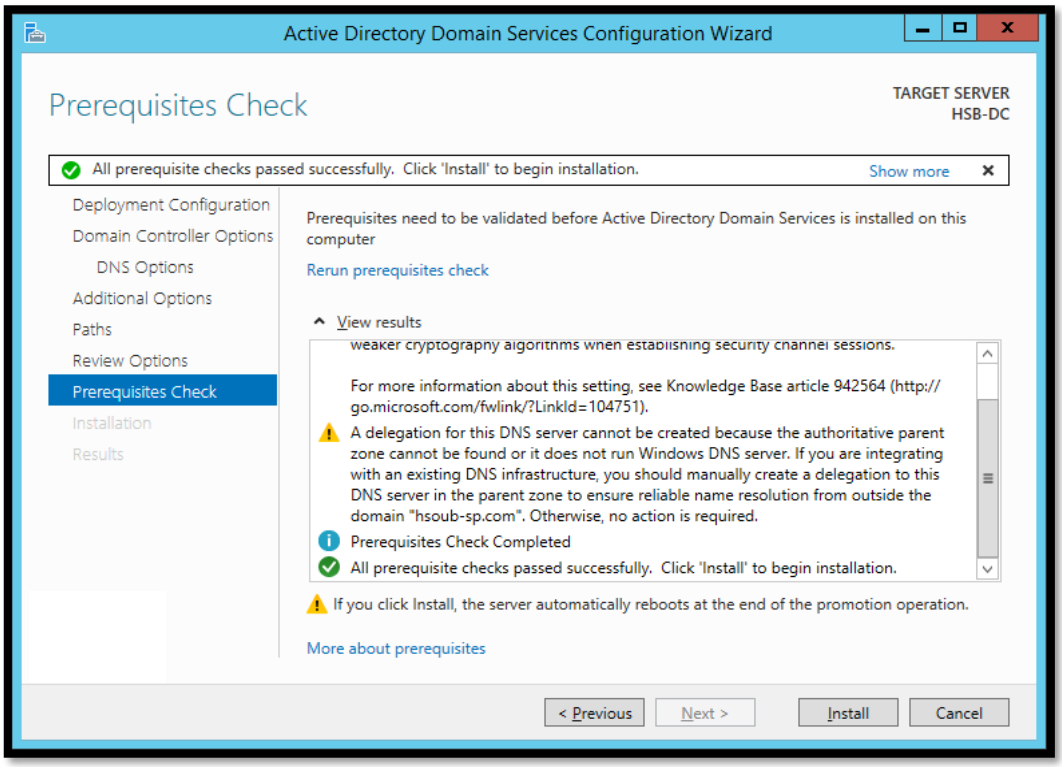
ستظهر بعد ذلك النافذة الخاصة بخيارات DNS والتي يظهر في أعلاها شريط تنبيه أصفر صغير، ولا بأس في ذلك لأنّنا نعمل على شبكة داخلية، لانريد إجراء أي تغيير هنا لذلك انقر زر Next فحسب، لتحصل على نافذة الخيارات الإضافيّة والتي تضم اسم NetBIOS الخاص بهذا المجال لأغراض التوافقيّة:



انقر زر Next لتظهر النافذة الخاصّة بتحديد المسارات، لانريد أيضاً إجراء أي تغييرات هنا، لذلك انقر الزر Next دون إجراء أيّ تعديل، ستحصل على النافذة التالية التي تلخّص المهام التي سيعمل على تنفيذها برنامج الإعداد.



انقر على الزر Next لتظهر النافذة الأخيرة التي تفحص متطلّبات التنصيب والتي ستعطيك تقريراً في نهاية عملها حول إمكانيّة بدء عمليّة التنصيب، ستحصل على نافذة شبيهة بما يلي:



انقر زر التنصيب Install ستبدأ عمليّة التنصيب وتأخذ بعض الوقت، وعندما تنتهي سيتم إعادة تشغيل الخادم بشكل تلقائي. بعد إعادة التشغيل والوصول إلى نافذة تسجيل الدخول ستلاحظ وجود بعض التغييرات، نجد أن اسم المستخدم administrator مسبوق باسم المجال الذي أنشأناه وهذه بالطبع إشارة جيّدة إلى نجاح العمليّة، أدخل كلمة المرور ثم سجّل الدخول.

تمرين :

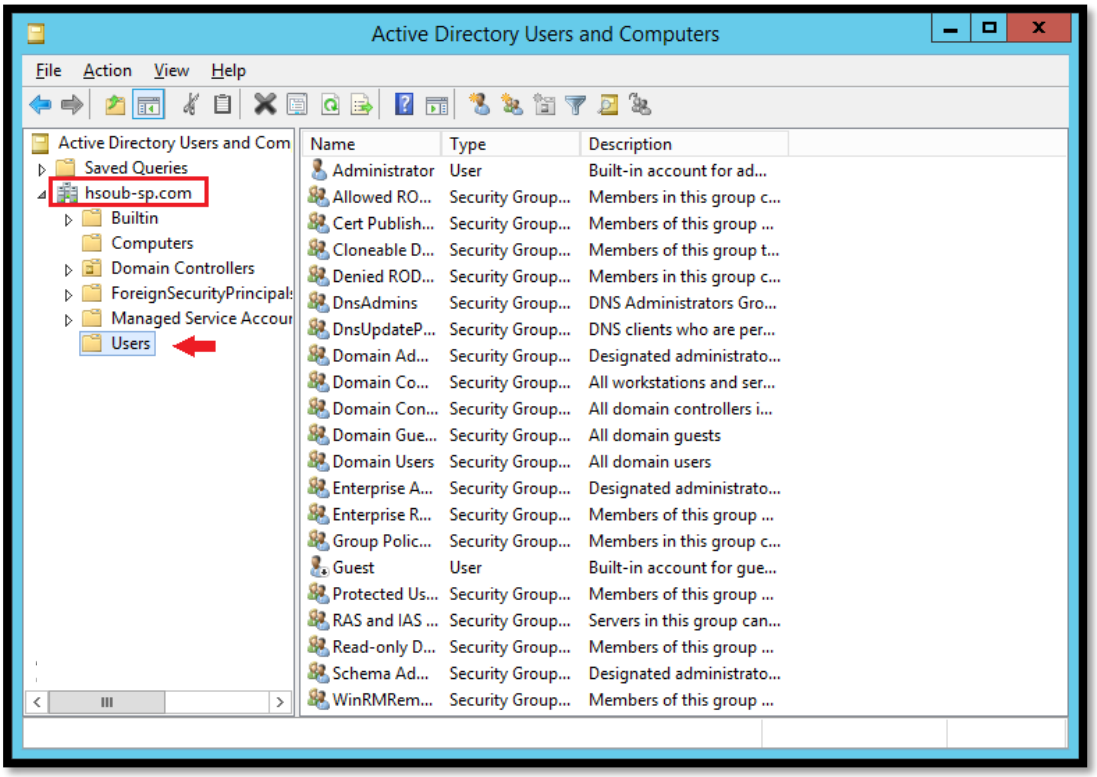
قم بتنصيب المتحكم بالمجال**Domain controller** متبع الخطوات التي قمت بدراستها ؟

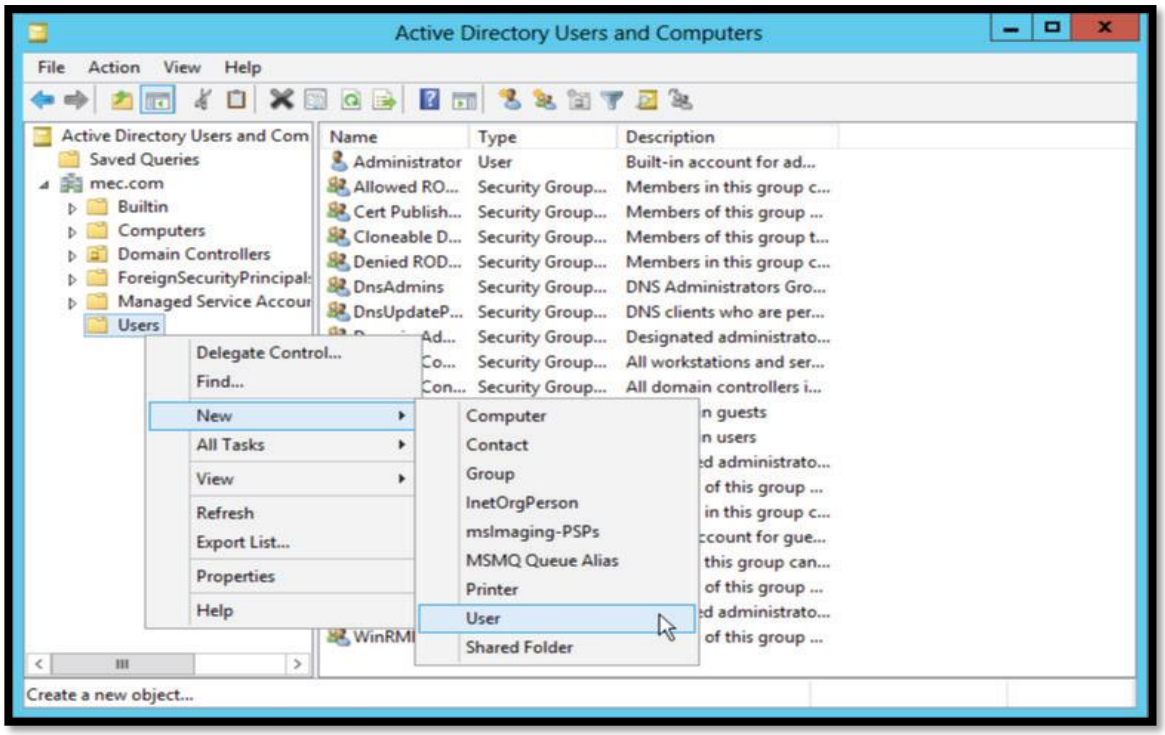
الفصل الثالث : إدارة خدمات المجال للدليل النشط AD DS

❖ إدارة حسابات الحواسيب

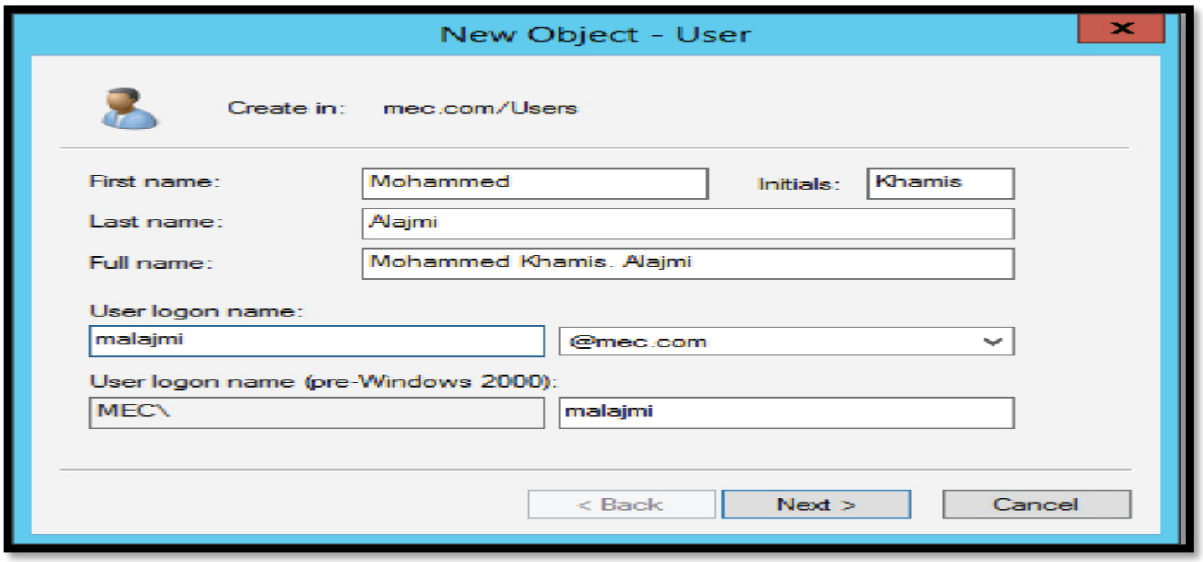
الخطوات :

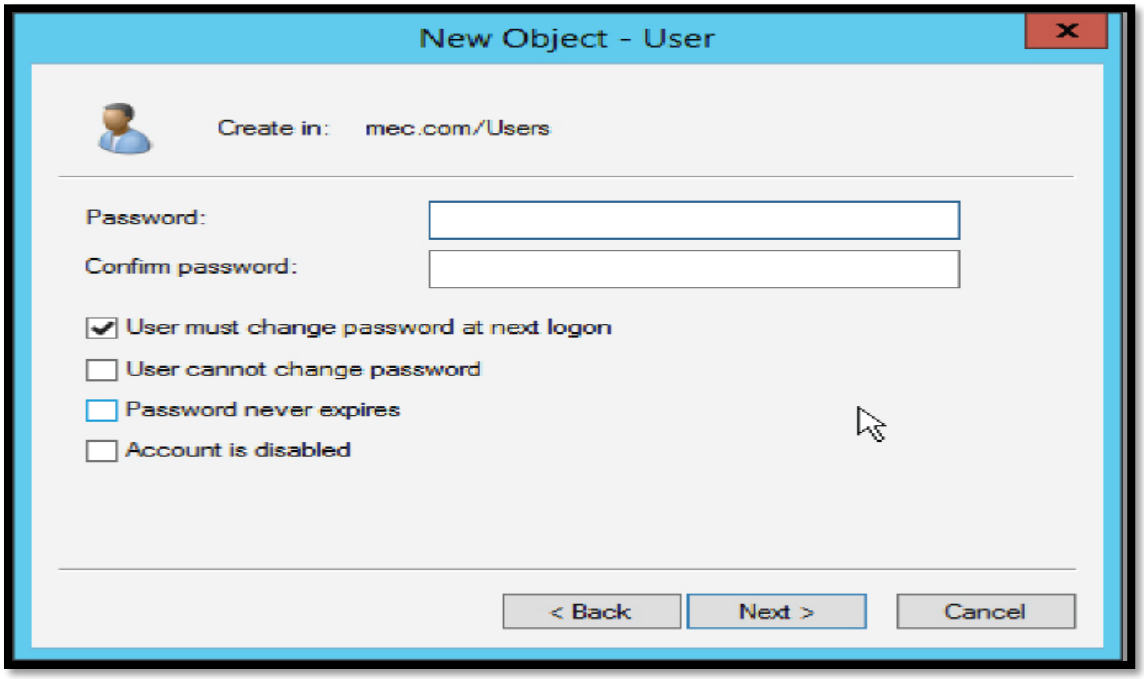
نذهب للـ Active Directory Users and Computers كما في الشكل





نحدد بالزر الأيمن ونختار مستخدم جديد





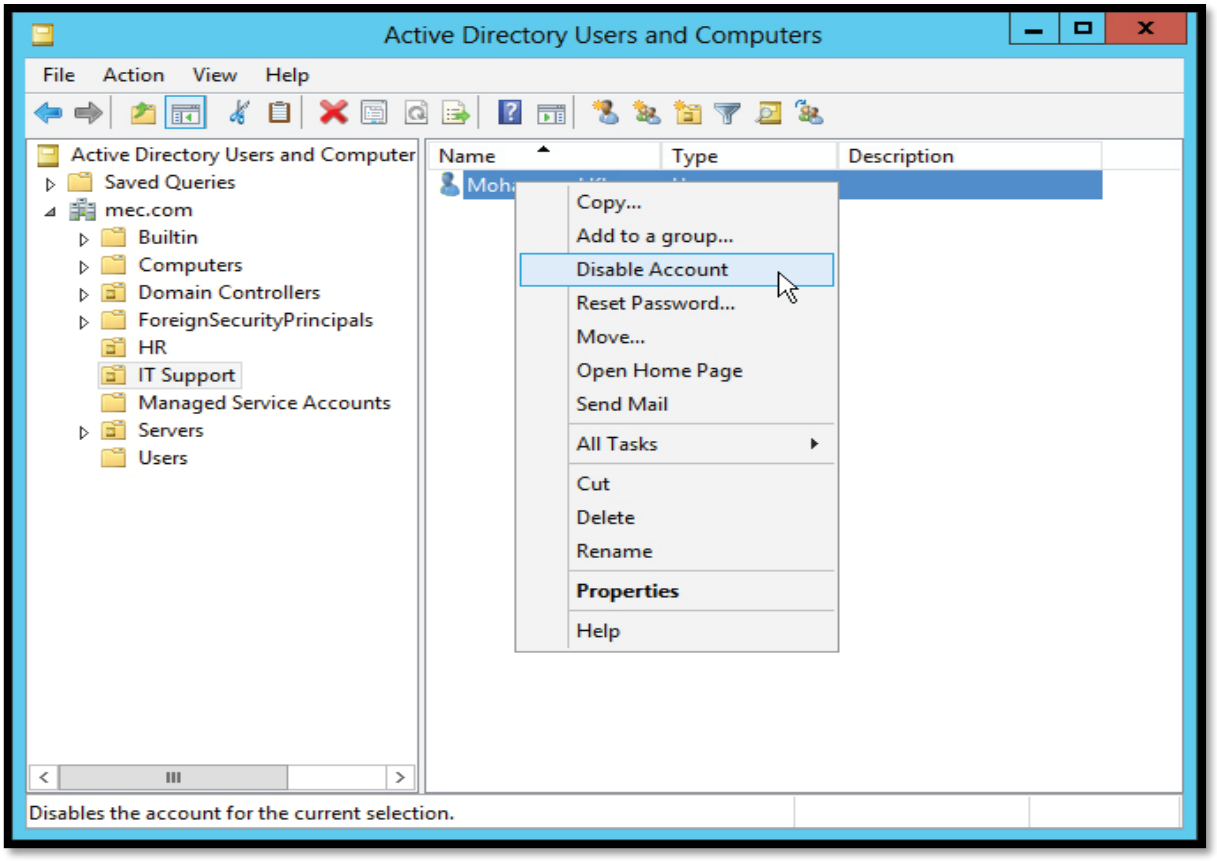
يجب أن تكون المعلومات المسجلة للمستخدم صحيحة ودقيقة، حتى نستطيع التعرف على هوية صاحب الحساب.

User logon name في هذه الخانة نحدد اسم المستخدم الذي سوف يستخدم لعملية تسجيل الدخول.

ندخل كلمة المرور الخاصة، وكما نلاحظ ظهور خطأ، في حالة اختيار كلمة مرور تحتوي على حروف أو أرقام فقط.

يجب أن تكون كلمة المرور ضمن الشروط والقوانين التي يحددها نظام التشغيل.

ومن ضمن الخيارات التي تظهر عند إنشاء مستخدم جديد بعد كتابة بياناته واختيار كلمة المرور

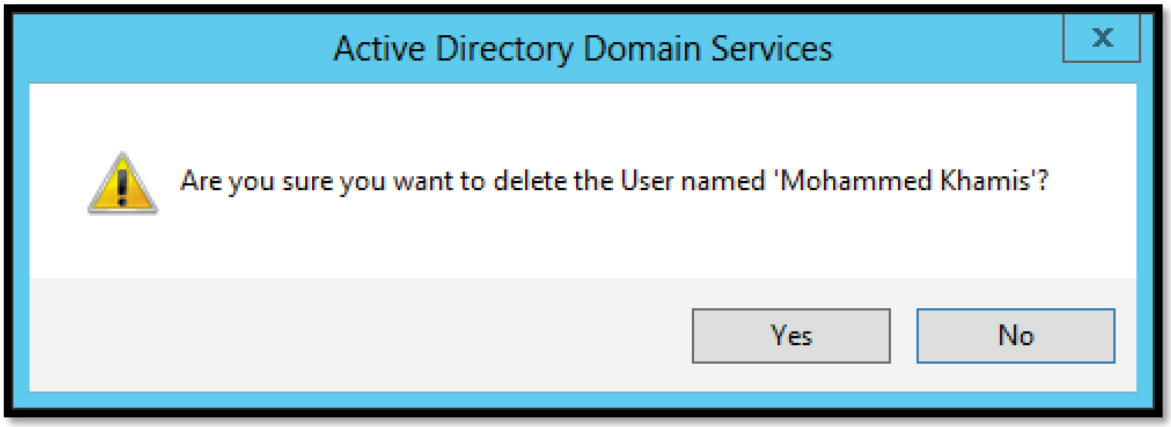


- **إلغاء وتفعيل حساب المستخدم**

ويمكن إلغاء وتفعيل الحساب عن طريق إيجاد اسم المستخدم، ونختار بالزر الأيمن Disable Account كما في الشكل التالي وإذا أردنا تفعيل الحساب، نضغط على Enable.

تأكيد حذف الحساب : ويمكن حذف الحساب عن طريق إيجاد اسم المستخدم، ونختار بالزر الأيمن Delete

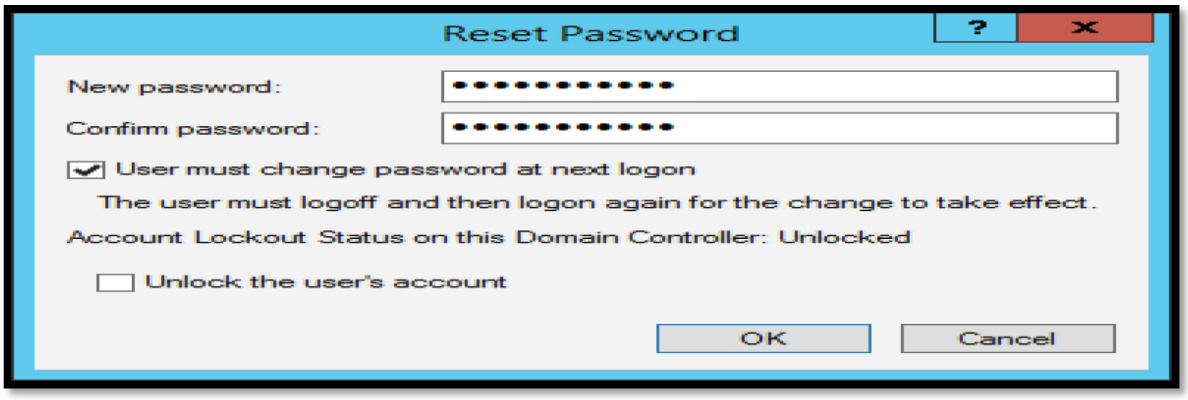
سوف تظهر رسالة لتأكيد الحذف :



- **إعادة تعيين كلمة المرور**

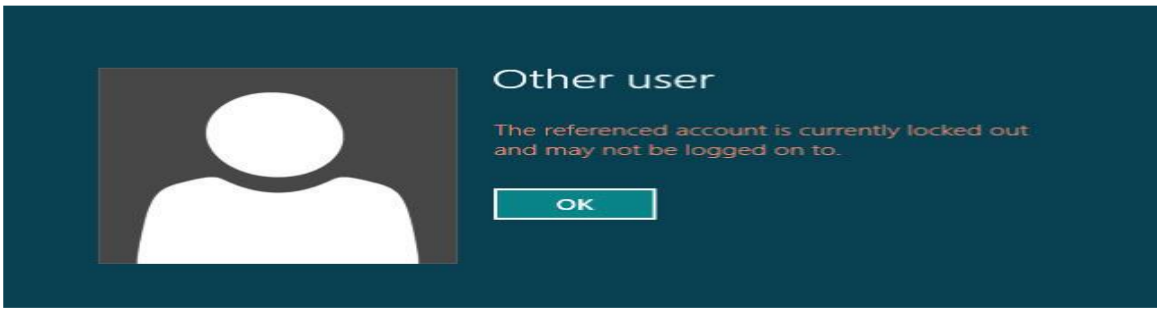
لإعادة تعيين كلمة المرور، اتبع الخطوات التالية :

- نذهب للـ Active Directory Users and Computers.
- نحدد اسم المستخدم بالزر الأيمن.
- Reset Password .
- تعيين كلمة مرور جديدة

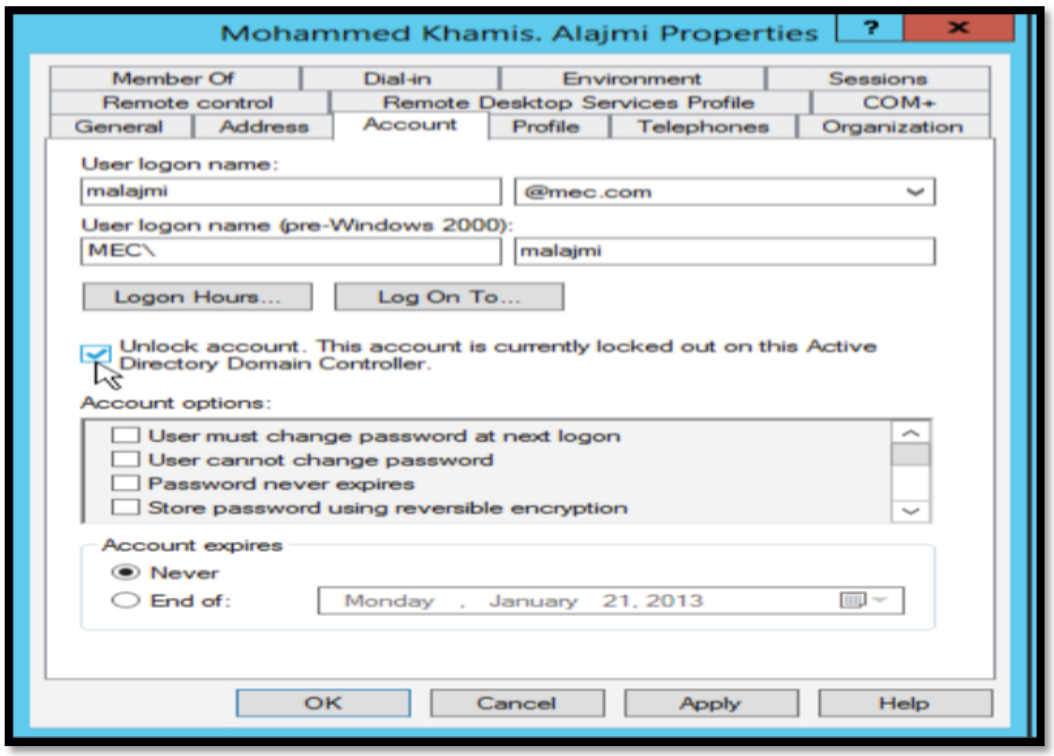


تفعيل خاصية Account Lockout

وهي خاصية يتم تفعيلها عن طريق سياسات وقوانين يفرضها نظام المؤسسة ويمكن لمدير الشبكة أن يفعل الحساب من جديد عن طريق إعادة تعيين كلمة المرور "في حالة أن المستخدم قد نسي كلمة المرور" لاحظ في الشكل كيف يتم إلغاء الحساب في حالة ادخال كلمة المرور بشكل خاطئ لعدة مرات

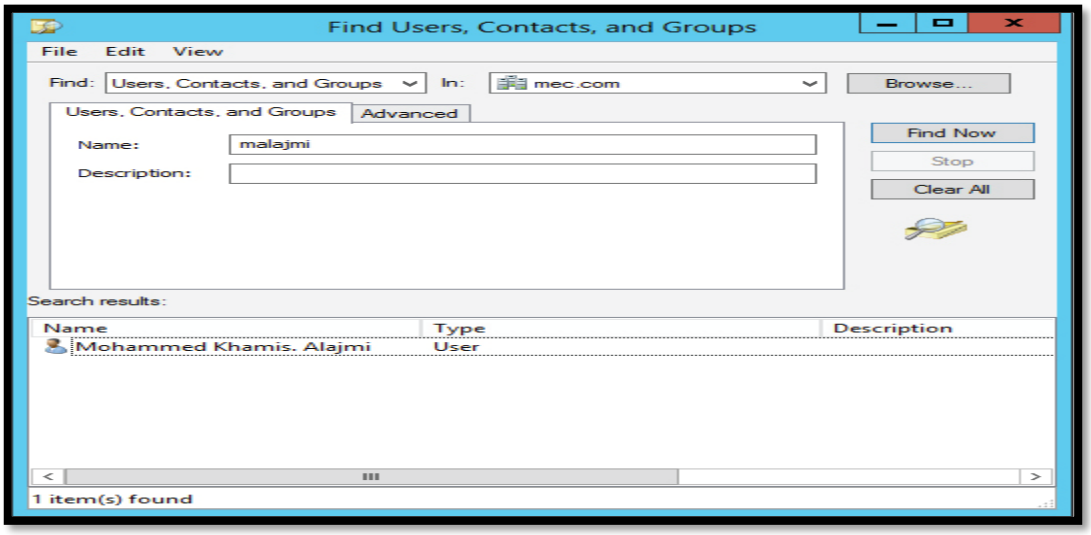


كما يمكن لمدير الشبكة أيضاً إلغاء قفل الحساب عن طريق خصائص المستخدم ثم Account وبعد ذلك نحدد خيار Unlock account،

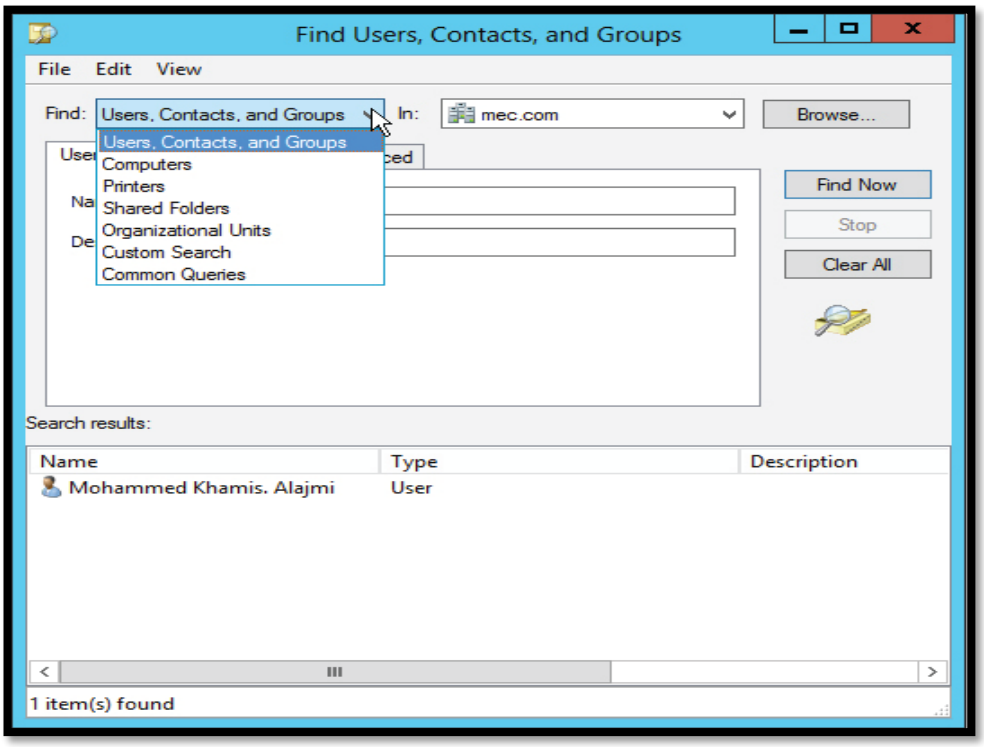


للبحث عن حساب المستخدم، نذهب للـ Active Directory Users and Computers **ونضغط على زر البحث:**

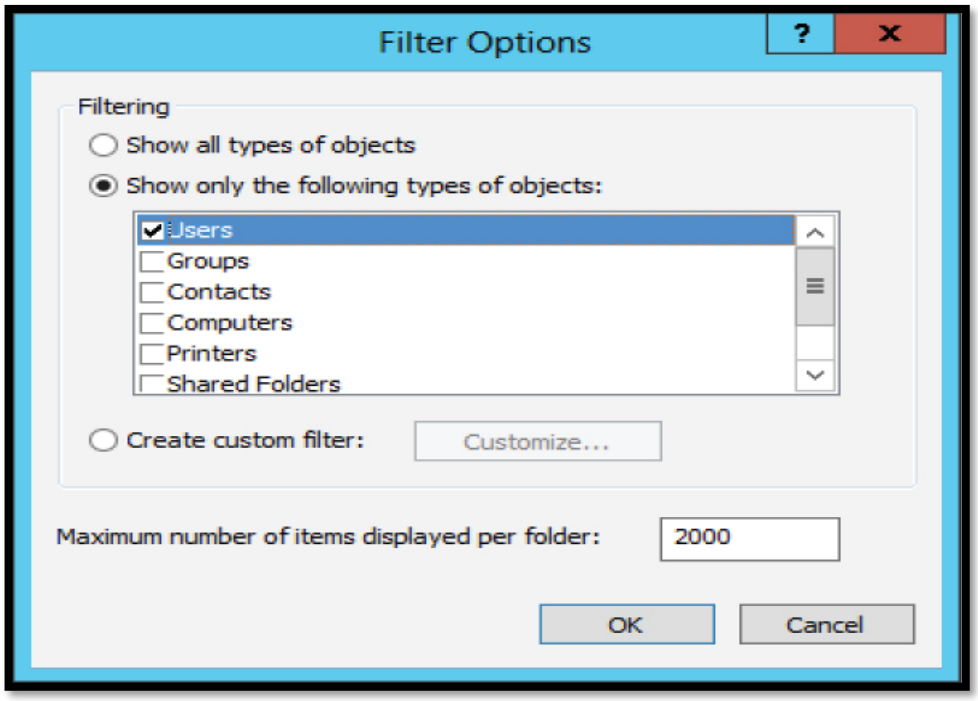
نقوم بكتابة اسم الحساب المراد البحث عنه في " Name "ونضغط على " Find Now "كما في الشكل



-كما نلاحظ في الشكل التالي ، فإنه يمكن تغيير خصائص البحث بحيث يمكننا البحث عن الأجهزة، الطابعات وغيرها من الخصائص التي تسهل عملية البحث.

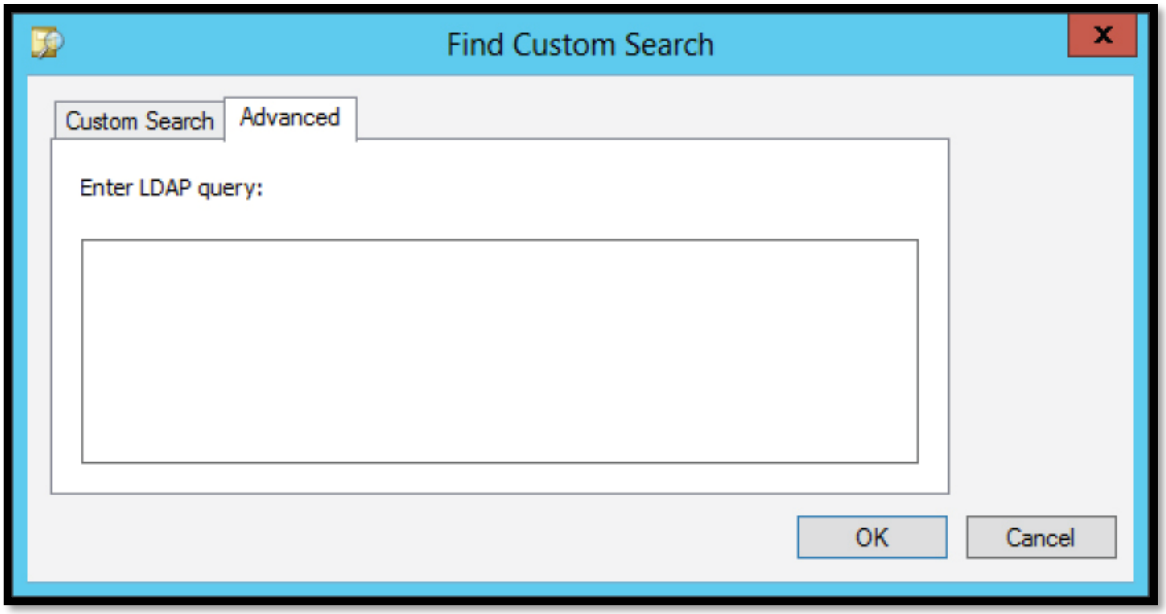


يمكننا أيضاً القيام بعملية التصفية (Filter) وذلك لعرض بعض الأجهزة أو حسابات المستخدمين، فمثلاً يمكننا عرض حسابات المستخدمين فقط على الـ Active Directory Users and Computers دون عرض الأجهزة كما نلاحظ في الشكل



يتيح لنا الـ AD خاصية أخرى لعرض الحسابات، فعند اختيار " Create custom filter " ، يمكننا البحث عن الحسابات مستخدمين، أجهزة، وغيرها، التي تبدأ بحرف معين أو اسم معين وهكذا.

-كما يمكن استخدام خاصية LDAB Query ، والتي تتيح لنا خيارات متقدمة لعرض الحسابات، فمثلاً يمكننا عرض حسابات المستخدمين التي تم إلغائها، أو الحسابات غير النشطة منذ فترة زمنية محددة.



- **إنشاء المجموعات**

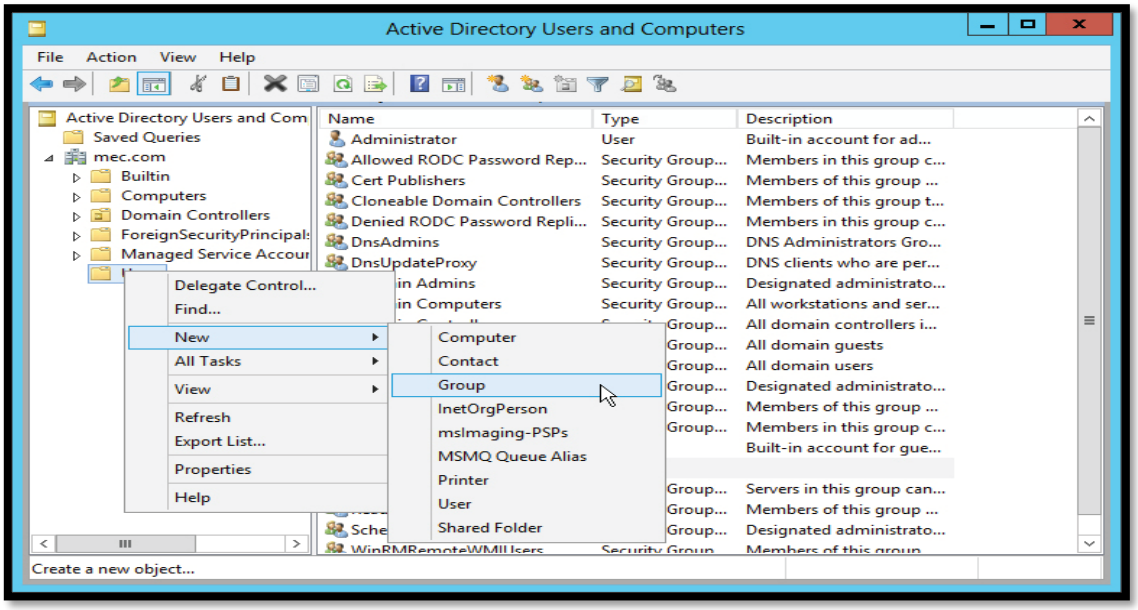
بعد أن تعرفنا على أنواع المجموعات وماهي فائدتها، سوف نتعرف على كيفية إنشاء مجموعة جديدة وعلى الخصائص المحددة لكل نوع من أنواع المجموعات.

لإنشاء مجموعة جديدة إتبع الخطوات التالية :

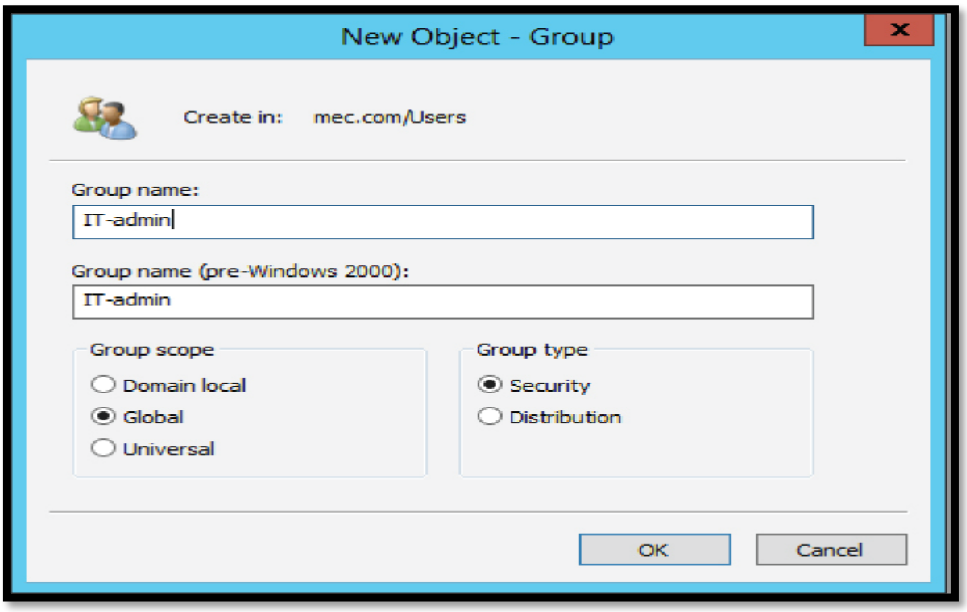
-نذهب للـActive Directory.

- Users.

-اتبع الخطوات كما في الشكل



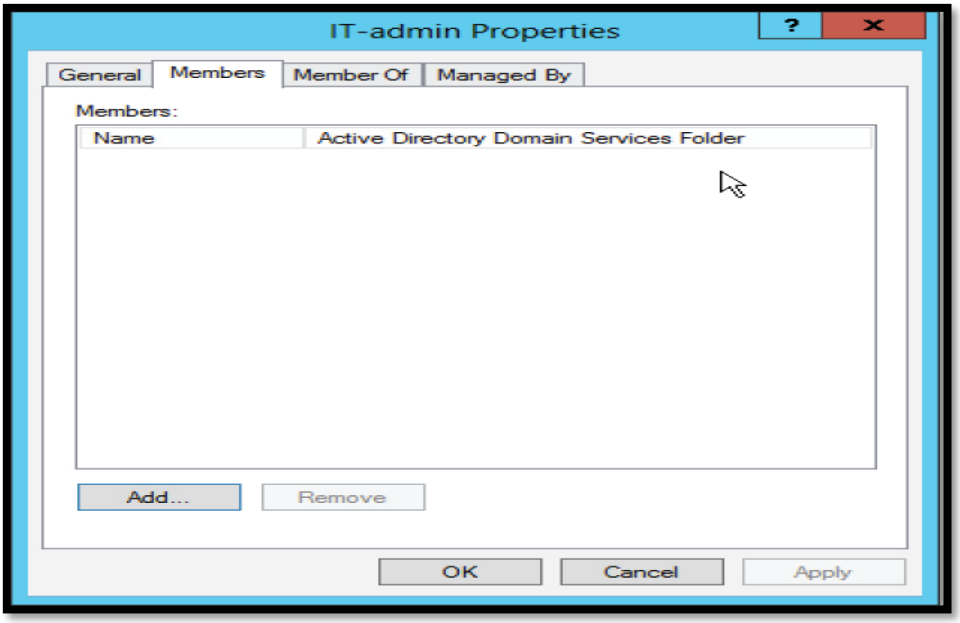
نختار اسما مناسباً للمجموعة ثم نقوم بتحديد نوع المجموعة.



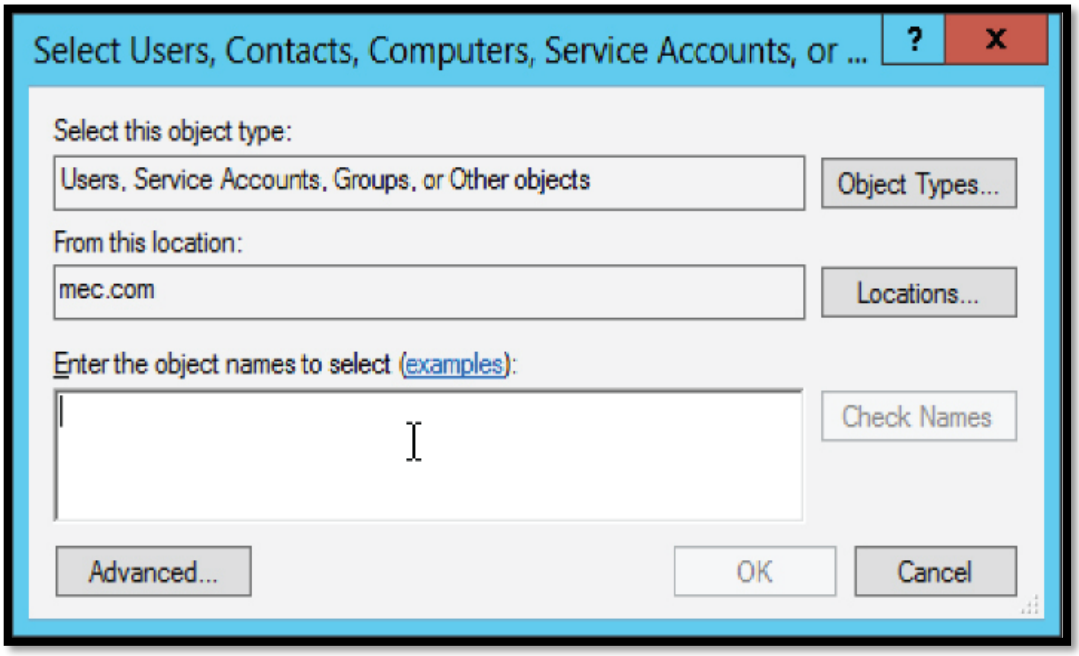
- **إضافة المستخدمين للمجموعات**

نقوم بإضافة المستخدمين للمجموعة كما في الشكل بالضغط على Add في الخيار "Members" والبحث عن اسم المستخدم الذي نريد إضافته، وبعدها يمكننا أن نقوم بإعطاء الصلاحيات لهذه المجموعة.

-ولو أردنا إضافة مجموعة لمجموعة أخرى، يمكننا ذلك عن طريق Member.



ويمكننا أيضــاً إضــافة أجهزة الشــبكة للمجموعات عن طريق تغيير ال Object Types ، وإضــافة الأجهزة من ضمن قائمة البحث.



- **حذف المجموعات**

لحذف أي مجموعة من المجموعات إتبع الخطوات التالية:

- نذهب للـActive Directory Users and Computers.
- نحدد مكان وجود المجموعة، نضغط بالزر الأيمن ثم نقوم بحذفها.

- **إنشاء بعض حسابات الكمبيوتر**

هناك طريقتان رئيسيتان للتعامل مع إنشاء حسابات الكمبيوتر:

- قم بإنشاء الحساب ثم ضم الكمبيوتر إلى المجال.
- قم بإنشاء الحساب عند ضم الكمبيوتر إلى المجال.

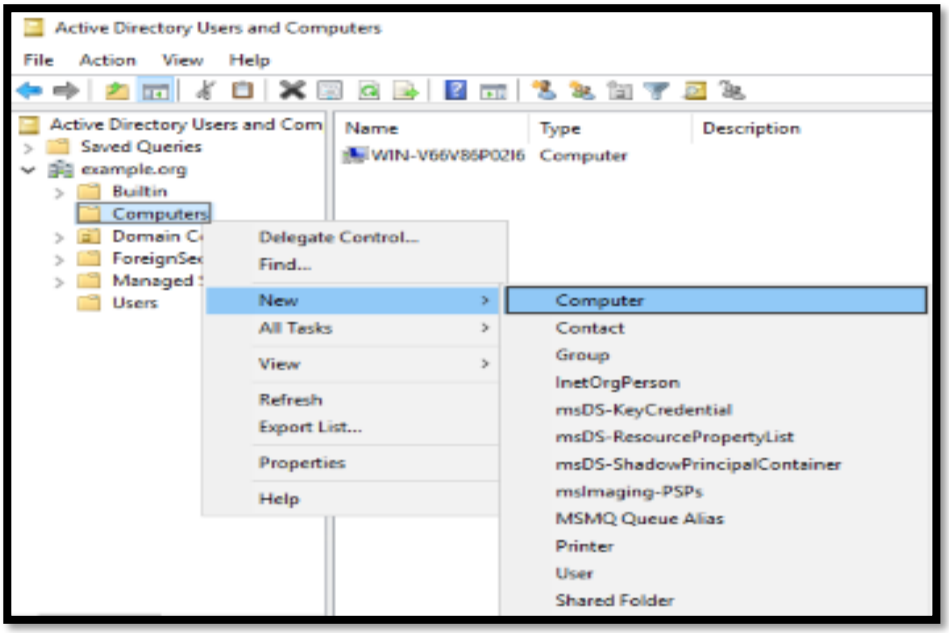
ما هي الطريقة الأفضل؟ يعتمد ذلك على كيفية إنشاء أجهزة الكمبيوتر الخاصة بك، إذا كنت تستخدم نظاماً آلياً مثل خدمات نشر Windows ، فإنه يتوقع وجود حساب كمبيوتر، من ناحية أخرى ، إذا قمت بإنشاء أجهزتك من صورة أو نموذج ، فحينئذ ستنشئ الحساب عند ضم الجهاز إلى المجال.

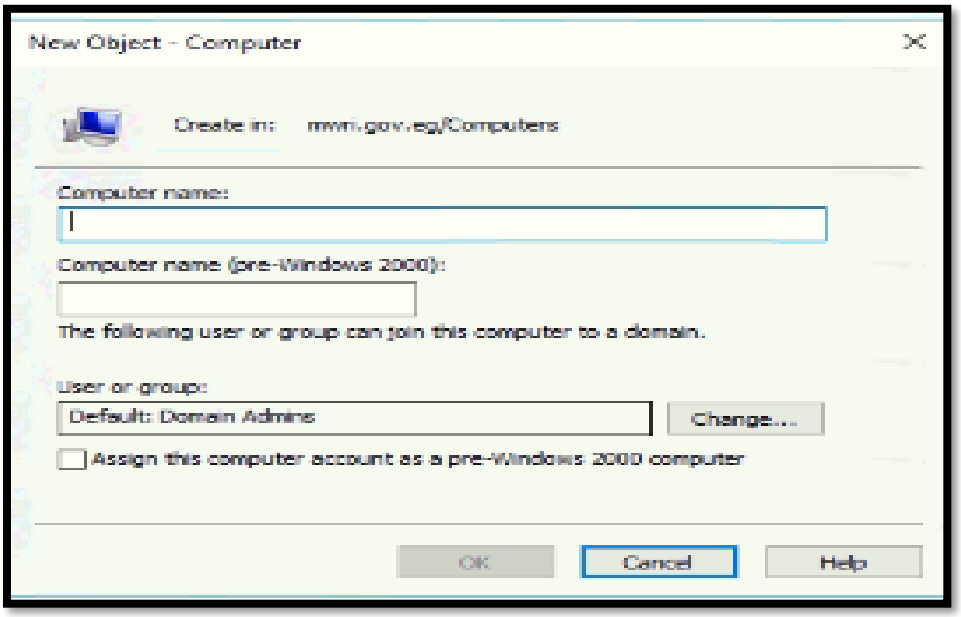
ميزة أخرى لإنشاء الحساب في Active Directory ، والمعروفة باسم التدريج المسبق ، هي أن الحساب موجود في الوحدة التنظيمية الصحيحة ، لذلك سيتلقى الجهاز كائنات نهج المجموعة الصحيحة بمجرد انضمامه إلى المجال، أيضاً ، يمكنك التحكم في من يمكنه ضم الجهاز إلى المجال.

- **إنشاء حساب كمبيوتر AD**

اختر New من قائمة المهام.

اختر Computer





- أضف اسم الكمبيوتر (سيتم ملء اسم ما قبل Windows 2000 تلقائياً).
- انقر فوق "تغيير" لتعديل المستخدم أو المجموعة التي يمكنها إضافة الجهاز إلى المجال.
- انقر فوق التالي.

❖ إدارة تفويض الصلاحيات

في المثال التالي، سنقوم بإنشاء OU ،ونسميها" IT Support " ، وسوف نقوم أيضاً بإنشاء مجموعة من المستخدمين في "IT Support" كفريق عمل مساعد لمدير الشبكة، وإعطائهم صلاحيات مختلفة عن بعضهم البعض حتى يتم فهم الموضوع بشكل أفضل كما هو موضح في الجدول:

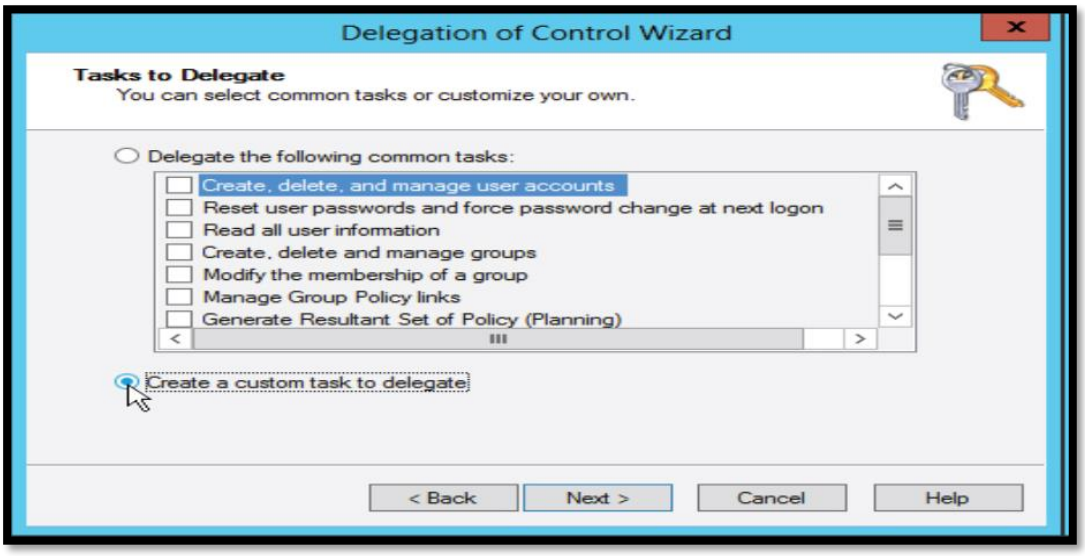
المستخدم	الصلاحيات
M-admin	إنشاء المستخدمين
Ali-admin	حذف المستخدمين
Said-admin	إعادة تعيين كلمة المرور

كما نلاحظ من الجدول السابق، فإن لكل مستخدم صلاحيات تختلف عن غيره من المستخدمين , وسوف نقوم بتطبيق هذه الصلاحيات لكل المستخدمين المساعدين لمدير الشبكة.

لإعطاء الصلاحيات بواسطة Delegate Control إتبع الخطوات التالية:

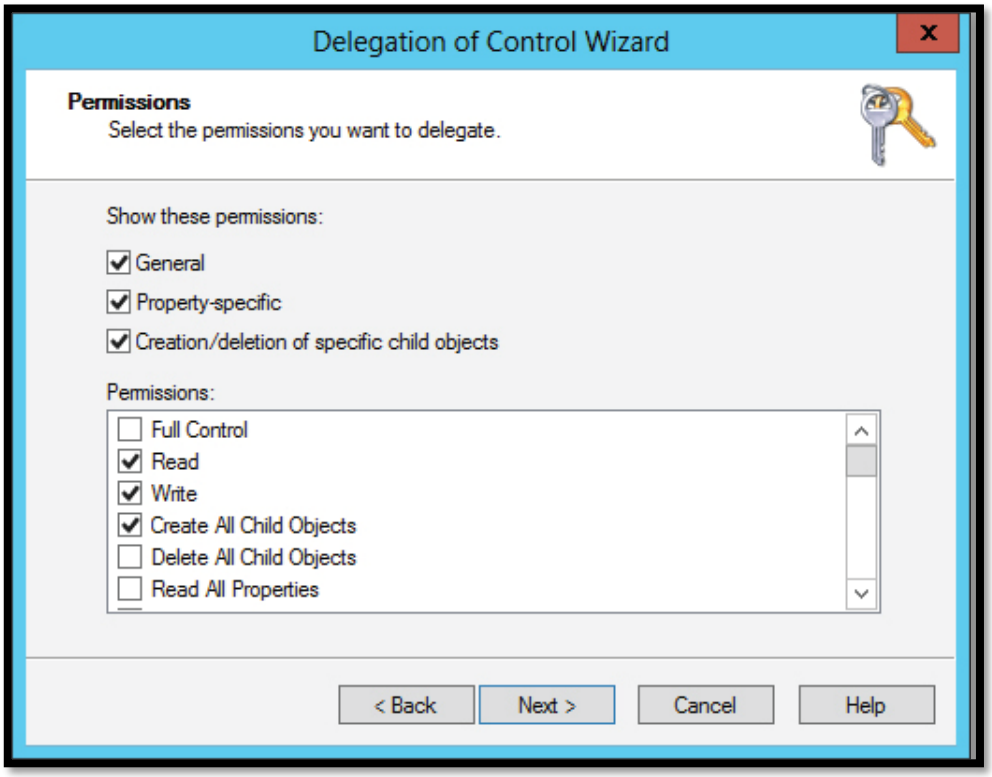
- نذهب لـActive Directory Users and Computer.
- نضغط بالزر الأيمن على IT Support.
- نختار Delegate Control.

- نقوم بإختيار المستخدم لإعطائه الصلاحيات بالضغط على Add.
- نقوم بإختيار Create a custom task to delegate.



ملاحظة :

لا يمكننا استخدام Create, delete, and manage user accounts، وذلك لأن هذا الخيار لا يتيح لنا إنشاء مستخدم فقط، بل يعطي الصلاحيات بحذف المستخدمين كذلك.

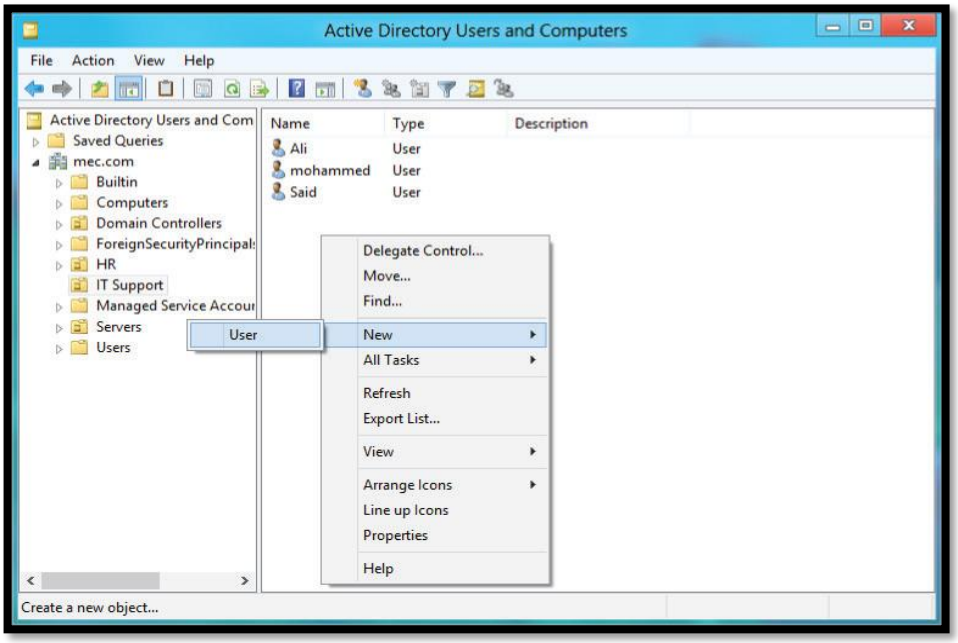


- **تطبيق الصلاحيات الـ delegations على المستخدمين :**

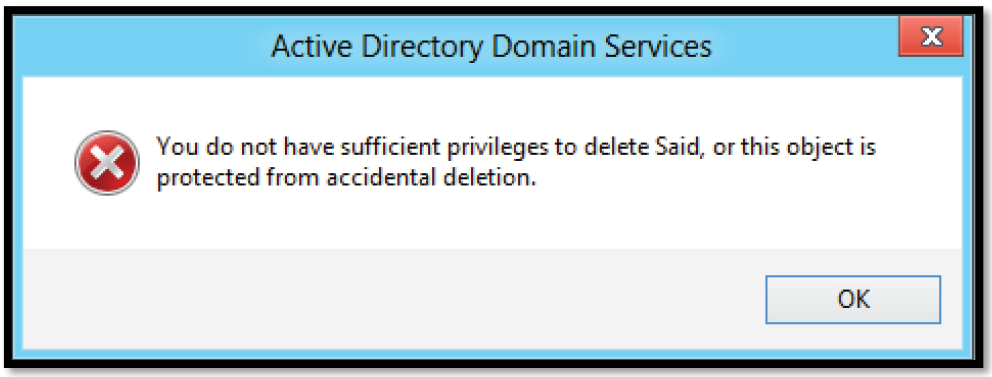
١ - نقوم بتسجيل الدخول بـ M-admin.

. Active Directory Users and Computers

نضغط بالزر الأيمن على IT Support ، ولاحظ أنه لا يوجد أي خيار إلا إنشاء مستخدم جديد كما في الشكل التالي:



وعند محاولة حذف أي مستخدم من IT Support فإن المستخدم لا يملك أي صلاحيات لذلك كما نرى في الشكل

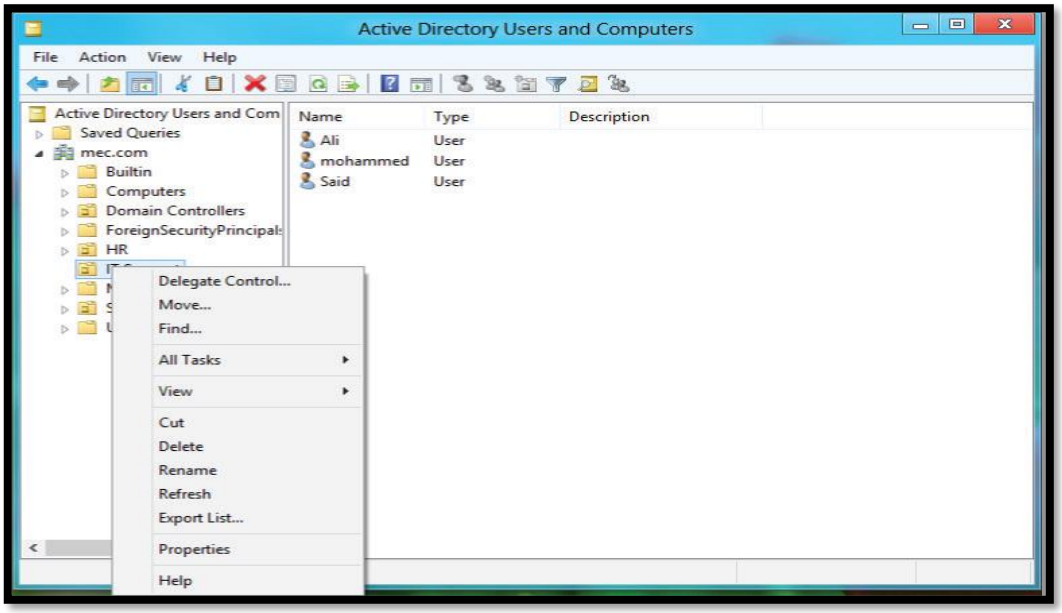


١- نقوم بتسجيل الدخول بـ Ali-admin

Active Directory Users and Computers .

٢- نحدد أحد المستخدمين الموجودين فـ IT Support وبالزر الأيمن نحاول حذف المستخدم

وعند محالة المستخدم " Ali-admin " إنشاء حساب حساب جديد أو إعادة كلمة المرور لأي مستخدم، لا يستطيع إنشاء مستخدم جديد أو إعادة تعيين كلمة المرور لأي مستخدم، بينما يمكنه حذف أي مستخدم ينتمي للـ"IT Support".

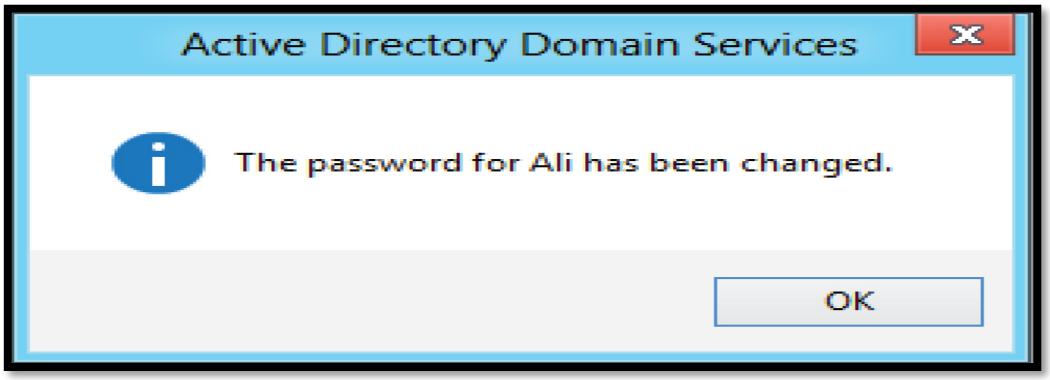


٣- نقوم بتسجيل الدخول بـ Said-admin.

Active Directory Users and Computers

نحدد بالزر الأيمن على أحد المستخدمين في الـ IT Support، إعادة تعيين كلمة المرور.

نلاحظ بأن Said-admin لديه الصلاحيات الخاصة بإعادة تعيين كلمة المرور



- **إضافة وتغيير الصلاحيات عن طريق Delegation Con**

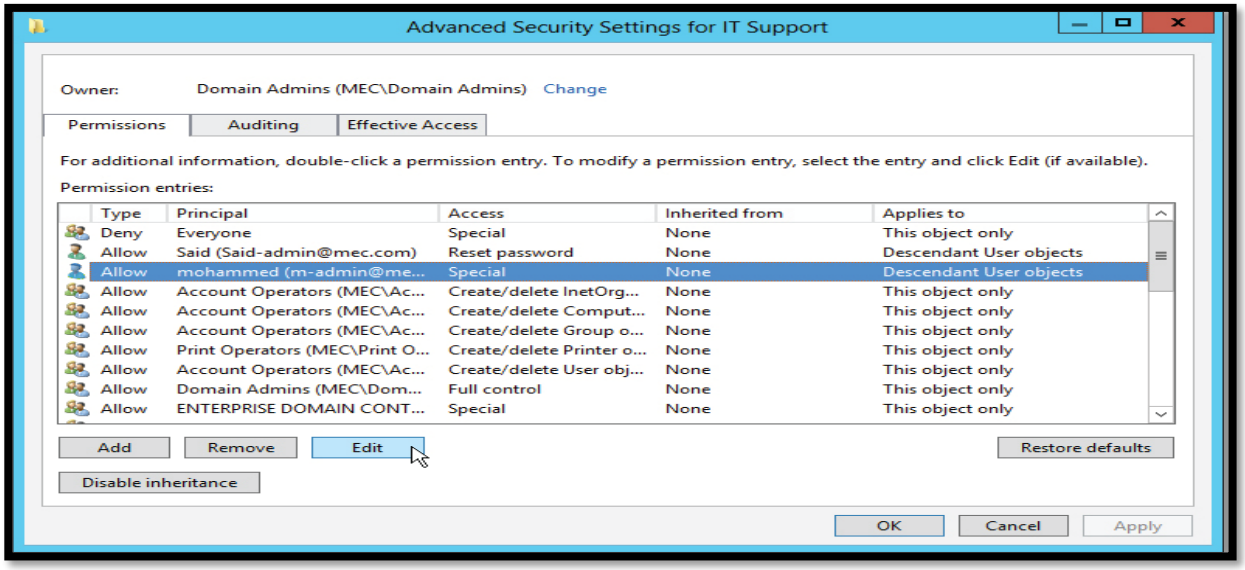
يمكن لمدير الشبكة القيام بتعديل الصلاحيات للفريق المساعد له،

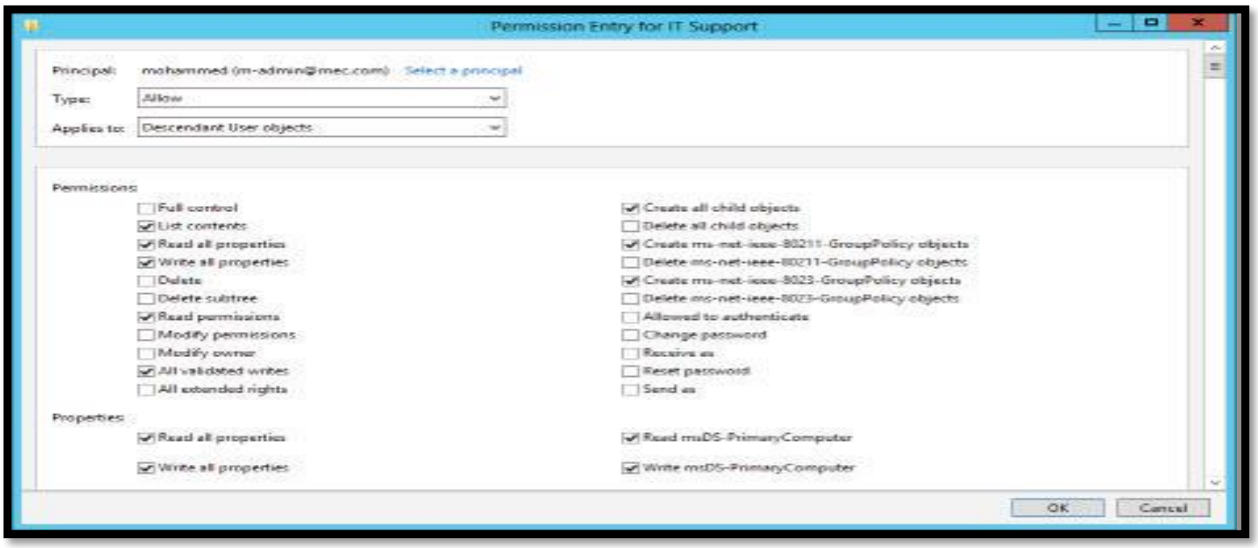
لتعديل الصلاحيات، إتبع الخطوات التالية:

- View Advanced Features .
- نذهب لخصائص IT Support.
- Advanced.

نحدد اسم المستخدم m-admin ونختار Edit كما في الشكل

بعد ذلك نقوم بحذف \ إضافة الصلاحيات التي نريد تطبيقها على المستخدم





تمرين :

إنشاء مجموعة جديدة متبعاً الخطوات التي تمت دراستها ..

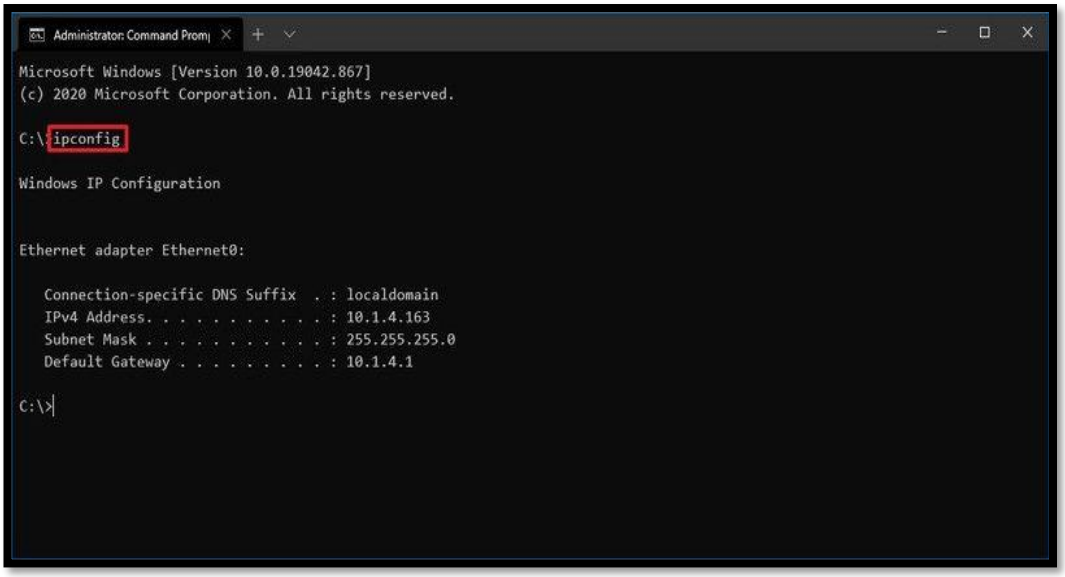
الفصل الرابع: كيفية جعل خدمات المجال للدليل النشط AD DS تعمل بشكل آلي

❖ استخدام موجه الأوامر لإدارة خدمات المجال للدليل النشط AD DS

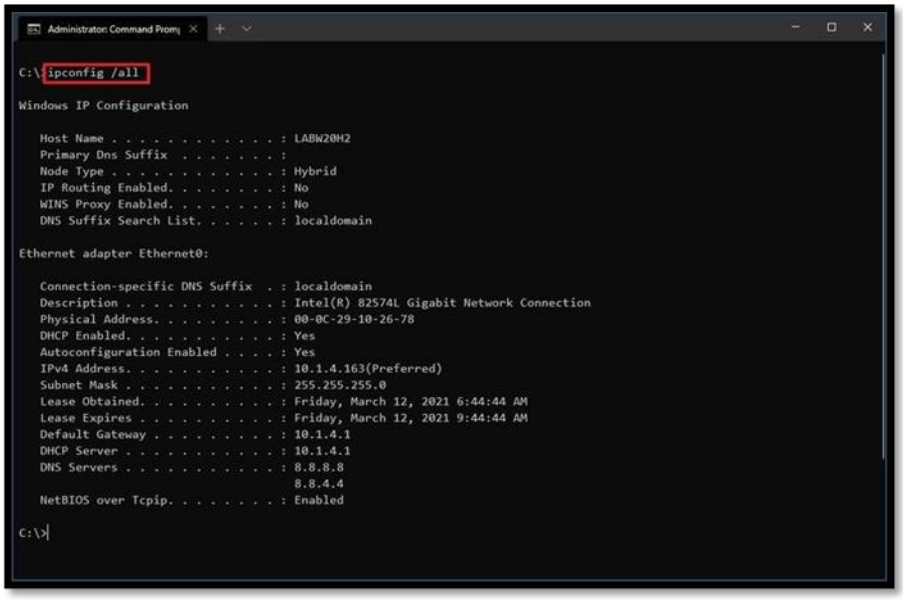
سنسلط الضوء على ثماني أدوات لموجه الأوامر والتي من المفترض أن تساعدك في إدارة واستكشاف مشكلات الشبكات

IPConfig

ipconfig يعد (تكوين بروتوكول الإنترنت) من بين أكثر أدوات الشـــبكات شـــيوعاً التي تتيح لك الاستعلام عن تكوين شبكة يتضمن الأمر أيضاً خيارات لتنفيذ إجراءات مختلفة ، مثل تحديث إعدادات بروتوكول التحكم في الإرســـال / بروتوكول الإنترنت (TCP / IP) لتكوين الديناميكي للمضـــيف (DHCP) ونظام اسم المجال (DNS)



Cls أمر لمسح الشاشة بعد أن لم تعد بحاجة إلى المعلومات لمتابعة تشغيل الأوامر بدون فوضى لعرض كل تكوين شبكة ipconfig /all



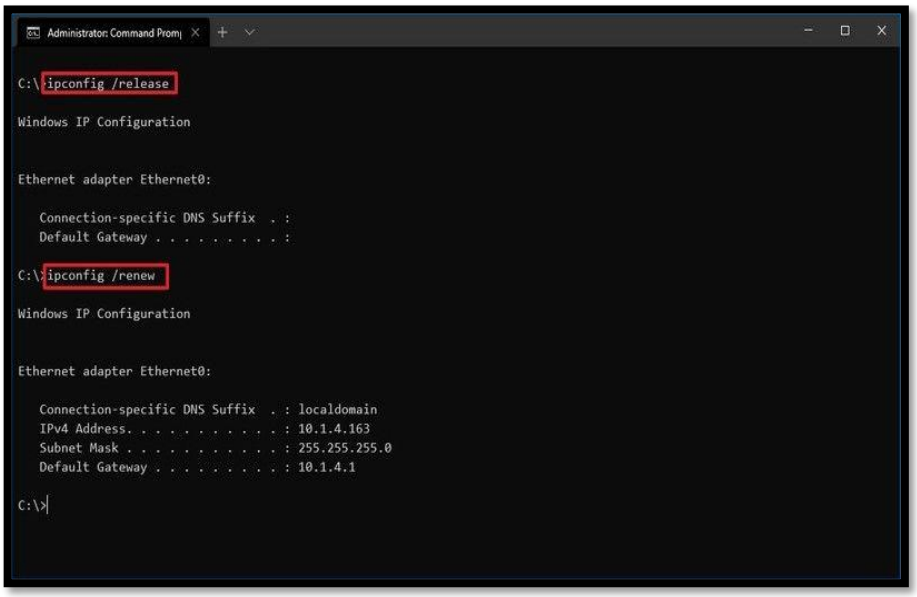
-قم بتحديث إعدادات الشبكة

-كتب الأمر التالي لإزالة تكوين الشبكة الحالي واضغط **يدخل**

ipconfig /release

-اكتب الأمر التالي لإعادة تكوين تكوين الشبكة واضغط **يدخل**

ipconfig /renew

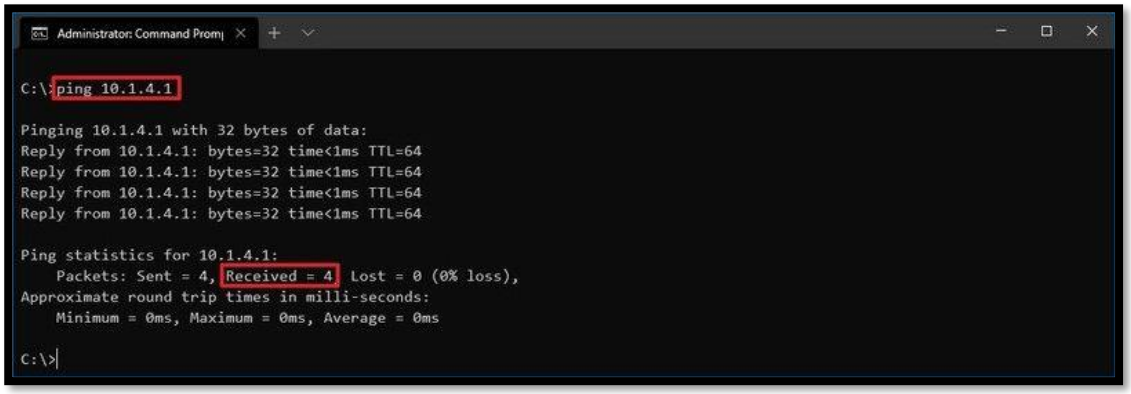


بعد إكمال الخطوات، سـيقوم الأمر الأول بمسـح التكوين الحالي، وسـيقوم الأمر الثاني بجلب الإعدادات الجديدة من خادم DHCP لحل مشـكلات الاتصـال، إذا لم نته صـلاحية الإعدادات المعينة ديناميكياً في الخادم، فمن الشائع رؤية نفس عنوان IP المعاد تكوينه على الجهاز.

- **اختبر اتصال الجهاز ping IP-OR-DOMAIN**

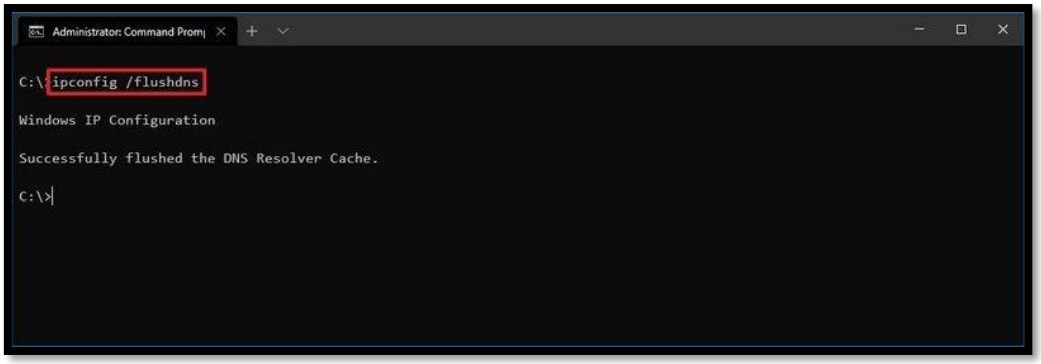
اسـتبدل **IP-OR-DOMAIN** باسـتخدام عنوان IP الفعلي أو اسـم المجال للكمبيوتر أو الخدمة التي تريد اختبارها.

على سبيل المثال، يختبر هذا الأمر الاتصال بين الجهاز المحلي وجهاز التوجيه



-قم بتحديث إعدادات DNS

ipconfig /flushdns



معلومات سريعة :إذا كنت تستخدم a-الخيار على سبيل المثال (ping -a 10.1.4.1) سيقوم الأمر أيضاً بتحليل العنوان إلى اسم مضيف لاختبار مكدس شبكات الكمبيوتر المحلي

ping 127.0.0.1 أو ping loopback

ملاحظة سريعة 127.0.0.1 :هو عنوان معروف جيداً ويشار إليه باسم عنوان الاسترجاع، عند تشغيل الأمر ، إذا تلقيت رداً ، فهذا يعني أن مكدس الشبكات على الجهاز باستخدام عنوان الشبكة الخاص هذا هو نفس الأمر ping

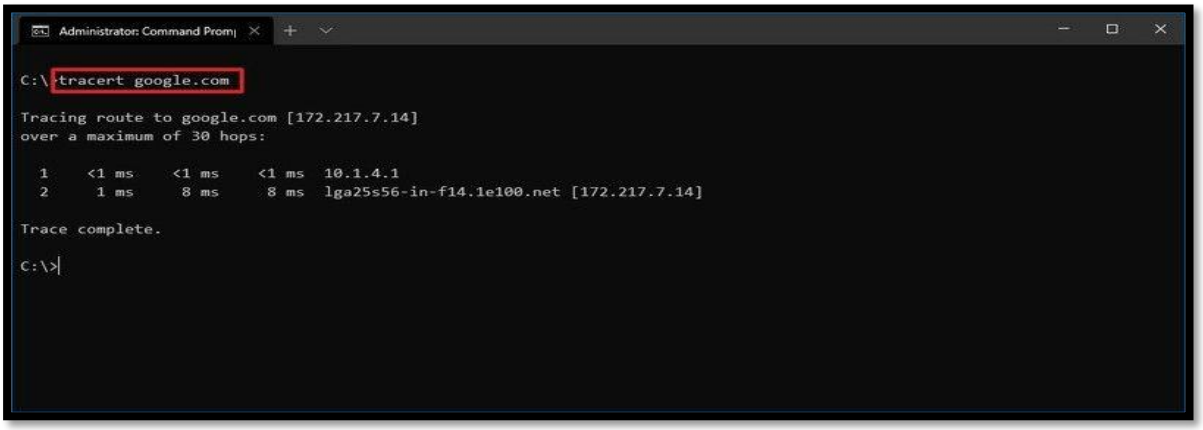
بمجرد إكمال الخطوات، إذا تلقيت أربعة ردود صدى ناجحة من الوجهة، فهذا يعني أن الجهاز يمكنه التحدث مع المضيف البعيد، إذا انتهت مهلة الطلب ، فهناك مشكلة يمكن أن تسببها العديد من الأسباب ، إذا كنت تتعامل مع مشكلات الاتصال ، فابدأ في تثبيت الكمبيوتر المحلي للتأكد من عمل مكدس الشبكة، ثم اختبر الاتصال بالموجه للتأكد من أن المشكلة ليست في الشبكة المحلية، ثم حاول اختبار اتصال أحد مواقع الويب لمعرفة ما إذا كانت هناك مشكلة في اتصال الإنترنت أو المضيف البعيد يجب أن تدرك أيضاً أنه إذا قام الجهاز البعيد أو الخدمة بحظر بروتوكول ICMP فستنتهي مهلة أمر ping.

- **تشخيص نشاط فقدان الحزم**

ping IP-OR-DOMAIN -t يختبر هذا الأمر الاتصال بين الجهاز المحلي وجهاز التوجيه لمدة ٦٠ ثانية

Tracert

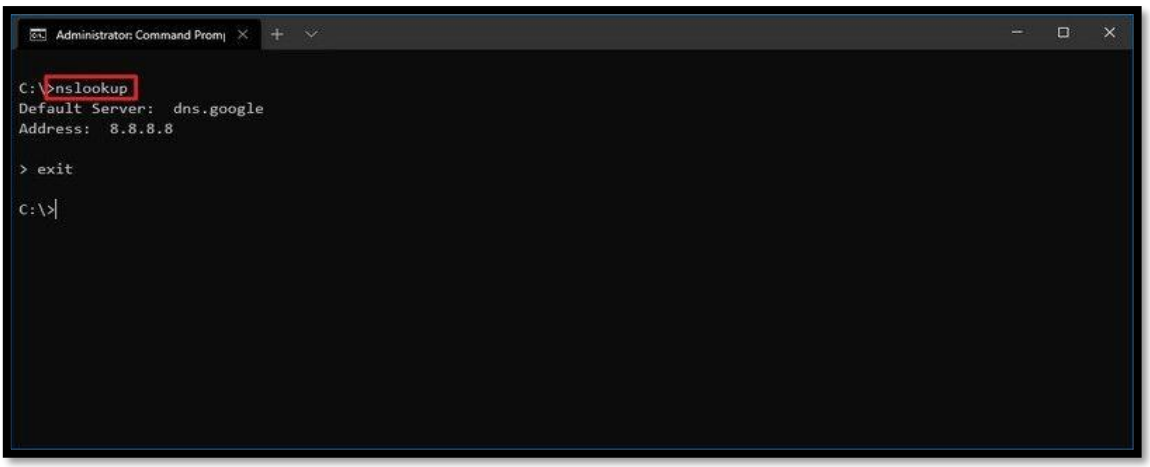
تتضمن Windows أيضاً ملفات **tracert** (تتبع المسار) ، أداة تشخيص لتحديد مسار الشبكة إلى وجهة باستخدام سلسلة من طلبات ارتداد ICMP التي تزيد بمقدار واحد في كل مرة ، مما يسمح بعرض قائمة بالمسار الذي سلكته الطلبات والمدة لك، بخلاف الأمر ping ، يتضمن كل طلب قيمة (TTL) مدة البقاء
tracert IP-OR-DOMAIN



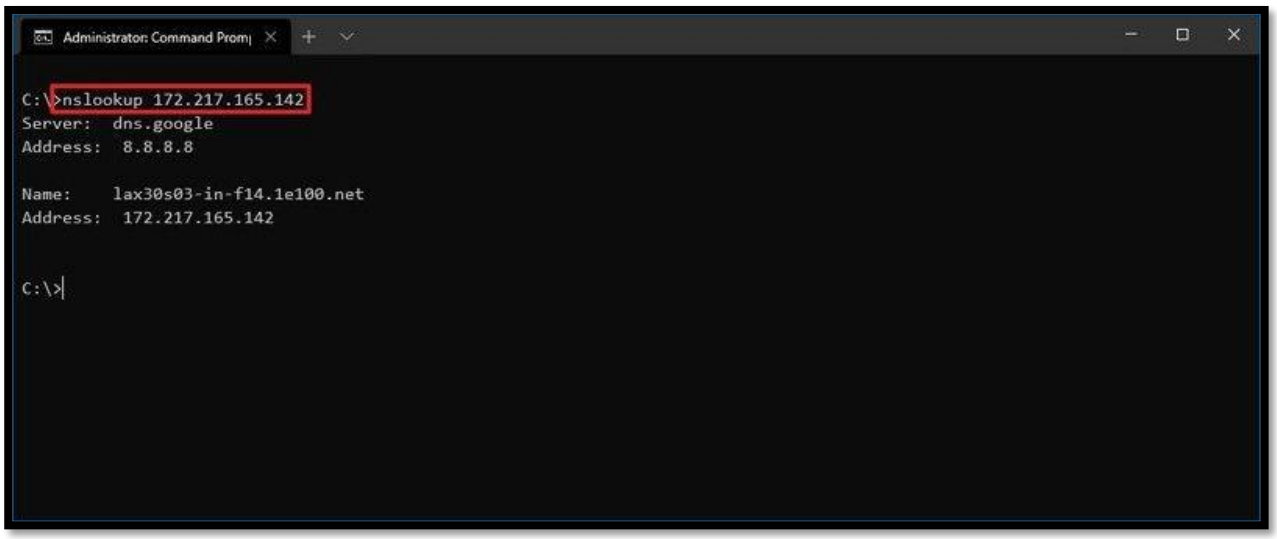
tracert -h HOP-COUNT IP-OR-DOMAIN لتحديد عدد القفزات

NSLookup

تعرض تفاصيل قيمة لاستكشاف المشكلات المتعلقة بـ(Name Server Lookup)

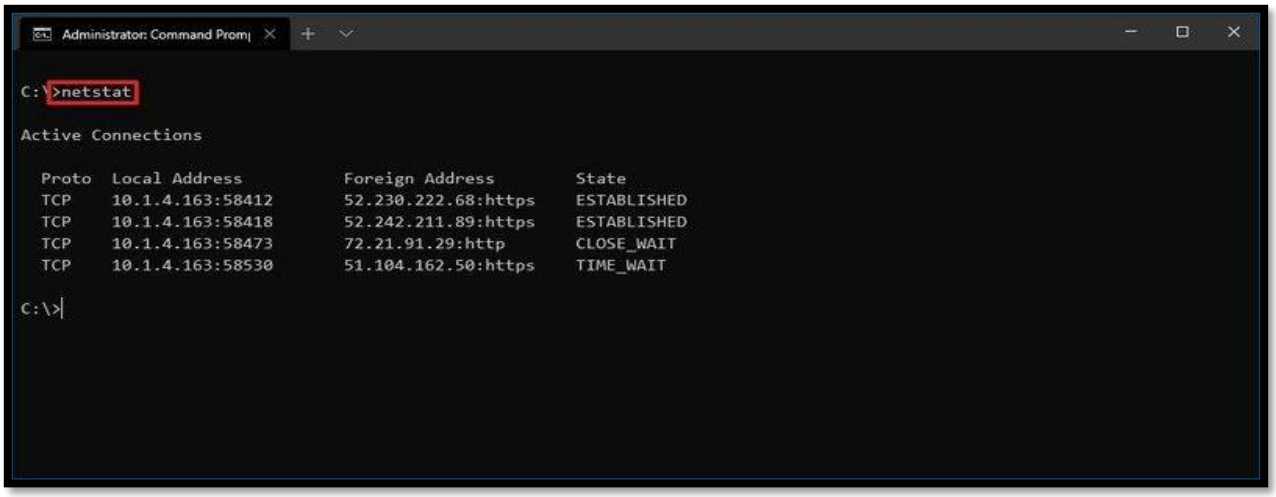


لتحديد اسم وعنوان خادم معين



NetStat (-a)

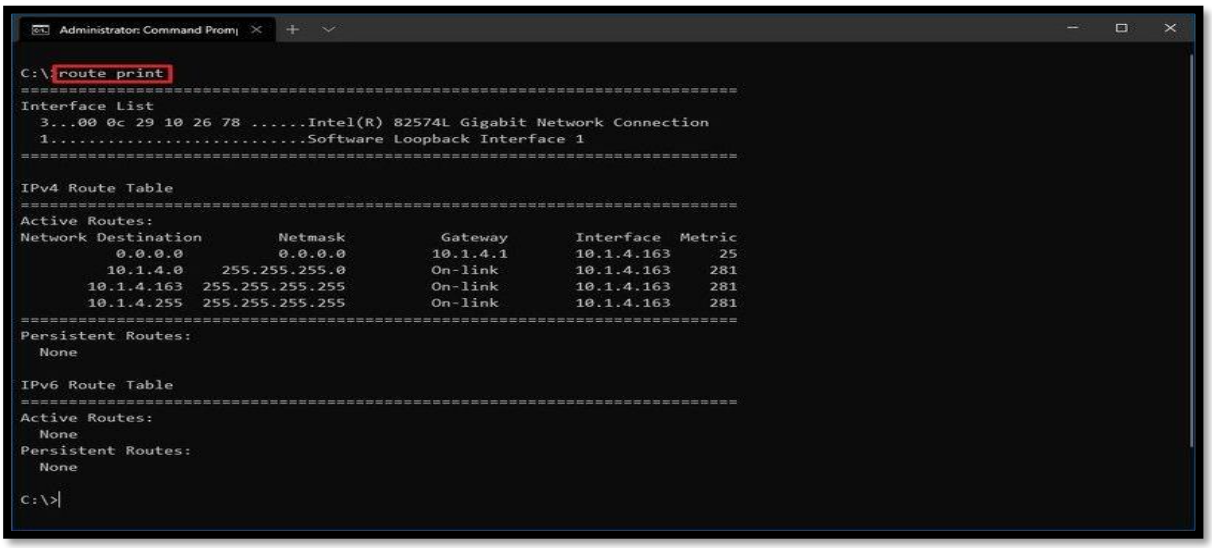
تعرض أداة (إحصائيات الشبكة) إحصائيات لجميع اتصالات الشبكة، يتيح لك فهم المنافذ المفتوحة والمتصلة لمراقبة واستكشاف مشكلات شبكات لنظام التشغيل Windows 10 والتطبيقات وإصلاحها



ARP -a- تحتفظ Windows 10 بملف **ARP** جدول (بروتوكول تحليل العنوان) ، الذي يخزن IP لإدخالات التحكم في الوصول إلى الوسائط (MAC) التي قام النظام بحلها، تتيح لك أداة arp عرض الجدول بالكامل وتعديل الإدخالات واستخدامها لتحديد عنوان MAC الخاص بجهاز الكمبيوتر البعيد.

الطريق "route print"

تعرض الأداة جدول التوجيه الذي يسمح لنظام التشغيل بفهم الشبكة والتواصل مع الأجهزة والخدمات الأخرى.



❖ استخدام PowerShell لإدارة خدمات المجال للدليل النشط AD DS

إنشاء كائن الكمبيوتر

لإنشاء كائن كمبيوتر جديد في وحدة تنظيمية محددة (OU) ،أدخل الأمر cmdlet:

```
New-ADComputer -Name "ComputerName" -SamAccountName "ComputerName" -Path "OU=Computers,DC=Domain,DC=com"
```

تحديد اسم الكائن لكل من المعلمة -Name و –SamAccountName ، لاتنس تضمين الاسم المميز (DN) في علامات الاقتباس بعد -Path ، والتي تحدد مكان إنشاء الكائن.

يكون هذا مفيداً بشكل خاص عند إدارة مواقع أو مجالات متعددة، لاسيما عند كتابة تغييرات نصية على AD مثل إنشاء حسابات جديدة، والتي قد تكون مرتبطة بالتعيينات الجديدة عندما تكون على متنها الموارد البشرية.

- **إنشاء مجموعات أمان جديدة**

```
New-ADGroup -Name "Security Group Name" -SamAccountName "SecurityGroupName" -GroupCategory Security -GroupScope Global -DisplayName "Security Group Name" -Path "CN=Groups,DC=Domain,DC=com" -Description "Brief description of the what security group is used for"
```

يساعد هذا الأمر cmdlet في إنشاء مثل هذه المجموعات، مما يخفف النفقات الإدارية بشكل كبير، أصبحت إدارة الحسابات وأجهزة الكمبيوتر أسهل بكثير عند تطبيق إعدادات الإدارة المستندة إلى الأدوار على مجموعات الأمان مقابل إجراء تغييرات على كل حساب وكل كائن على حدة.

- **إنشاء حساب مستخدم جديد :**

```
New-ADUser -Name "User Account Name" -SamAccountName "UserAccountName" -AccountPassword (ConvertTo-SecureString "password" -AsPlainText -Force) -DisplayName "User Name" -Enabled $True -GivenName "FirstName" -Path "CN=Users,,DC=Domain,DC=com" -Server "controller.domain.com" -Surname "LastName" -UserPrincipalName username@domain.com
```

يسير إنشاء حسابات المستخدمين وإدارتها جنباً إلى جنب عندما تعمل AD كواجهة خلفية للبنية التحتية المركزية للمؤسسة، يمكن أن ينمو إنشاء حسابات مستخدمين جديدة ليكون مهمة شاقة، نظراً لجميع المعلمات والحقول النصية العديدة الموجودة، وعلى الرغم من أنه ليس كل شيء مطلوباً، إلا أن العديد منها مفيد بالتأكيد عند ملؤه، مثل البريد الإلكتروني والعنوان على سبيل المثال لا الحصر.

يمكن تكوين مضيف كامل لهذه السمات عن طريق إضافة معلمة -OtherAttributes وإقرانها بالسمات التي ترغب في تكوينها، مثل السابق: @ {"title = "job_title"؛ mail = "username@domain.com""} مثلا.

- **إنشاء OU جديد:**

```
New-ADOrganizationalUnit -Name "OU Name" -Path "DC=Domain,DC=com"
```

تعتبر OUs ضرورية لإدارة حسابات المستخدمين وكائنات الكمبيوتر في النهاية الخلفية للشبكة، بعبارة أخرى ، سيشمل Active Directory الذي تم إعداده بشكل صحيح مجموعة مصممة جيداً من OUs ، والتي يمكن من خلالها تقسيم جميع الكائنات التي تديرها AD مركزياً، مما يؤدي إلى إنشاء دليل منظم ومنظم ودليل يفسح المجال بسهولة لتقوية الشبكة مع الحفاظ على سهولة إدارتها.

تشير أسطر Cmdlets التي تبدأ بـ "جديد" إلى إنشاء هذا الكائن، عندما تتطلب المهمة إزالة كائن ، استخدم بادئة "إزالة" بدلاً من "جديد" بالإضافة إلى ذلك ، يعمل هذا عندما تحتاج إلى تعديل كائن ، بدلاً من استخدام بادئة "Set" بدلاً من "جديد" يتبع بناء جملة أوامر cmdlets هذه تنسيقاً مشابهاً لتنسيق أوامر cmdlets ذات البادئة الجديدة المذكورة أعلاه.

- **إضافة / إزالة المستخدمين أو كائنات الكمبيوتر إلى / من المجموعات**

Add-ADGroupMember SecurityGroupName -Members Username01 -Server "controller.domain.com" Remove-ADGroupMember SecurityGroupName -Members Username01 -Server "controller.domain.com"

بمجرد إنشاء حسابات المستخدمين وكائنات الكمبيوتر، يمكن للمسؤولين اختيار إضافتها إلى مجموعات الأمان الموجودة لزيادة سهولة الإدارة، بتنفيذ الأمر cmdlet هذا ، ستحصل المعلمة -Members التي تسرد الحسابات المراد إضافتها على عضوية في مجموعة الأمان، عند إضافة حسابات متعددة ، يمكن فصلها بفاصلة في سطر واحد، بالإضافة إلى ذلك ، فإن المعلمة -Server على الرغم من كونها اختيارية ، ستحدد وحدة تحكم المجال التي يجب الاتصال بها لمعالجة التغييرات، لإزالة أعضاء من مجموعة ، استخدم الأمر Remove-ADGroupMember الذي يتبع نفس الصيغة.

احصل على كلمة المرور المخزنة محلياً من كائن الكمبيوتر

Get-AdmPwdPassword -ComputerName "computer.domain.net"

بالنسبة للمؤسسات التي طبقت حل كلمة مرور المسؤول المحلي (LAPS) الممتاز (والمجاني) من Microsoft ، هناك أمر cmdlet يمكنه الحصول على كلمة المرور المخزنة في كائن الكمبيوتر داخل AD وعرضها في خطوة واحدة سهلة الاسترداد مقابل البحث في قاعدة البيانات لكل كائن لتعقب السمة ms-mcs-AdmPwd

- **ربط جهاز كمبيوتر بمجال**

-Add-Computer -DomainName "domain.com" -Credential

Domain\Username -Restart Force

يعد ضم جهاز كمبيوتر إلى المجال أحد أكثر المهام شيوعاً التي يؤديها محترفو تكنولوجيا المعلومات، عشرات الآلاف من المرات في حياتهم المهنية، وعلى الرغم من أن المهمة تستغرق بضع دقائق فقط لكل جهاز، إلا أنه يمكن تشغيلها من PS إلى التبسيط ومع بعض التعديلات، قد تكون العملية بأكملها مؤتمتة بالكامل، فقط تذكر أن الأمر cmdlet - كما هو مفصل أعلاه - سيطلب من المسؤول كلمة المرور الخاصة به لإكمال الانضمام إلى المجال قبل إعادة تشغيل الجهاز.

تمكين / تعطيل المستخدمين أو أجهزة الكمبيوتر أو حسابات الخدمة

Enable-ADAccount -Identity "ComputerName"

Disable-ADAccount -Identity "Username"

يمكّن (أو يعطل) حساب مستخدم أو كائن كمبيوتر أو حساب خدمة تديره AD للسماح (أو منع) المستخدم أو حساب الكمبيوتر من المصادقة مع أو على الشبكة.

- **تفعيل (فتح) حسابات المستخدمين**

Unlock-ADAccount -Identity "Username"

غالباً ما يُلاحظ من خلال التحليل الإحصائي أن قسم تكنولوجيا المعلومات يقضي وقتاً منظماً في حل مكالمات مكتب المساعدة المتعلقة بإلغاء تأمين حسابات المستخدمين وإعادة تعيين كلمات المرور الخاصة بهم، تمت تغطية الأخير أعلاه باستخدام الأمر cmdlet لـ Set-ADUser ، ومع ذلك ، فإن الأول هو أمر cmdlet المحدد هذا، ينطبق هذا بشكل أساسي على الحسابات التي تجاوز فيها المستخدمون عدادات إعادة محاولة كلمة المرور الخاصة بهم وقاموا بإغلاق حسابهم بشكل فعال من الاستخدام، بمجرد إلغاء القفل ، تتاح وظيفة AD (بما في ذلك مصادقة الحساب) مرة أخرى.

- **البحث عن اسم الكمبيوتر أو حسابات المستخدمين المعطلة**

```
Search-ADAccount -AccountDisabled | FT Name,ObjectClass
```

يمكن استخدام الأمر Search-ADAccount cmdlet لأكثر من مجرد البحث عن حسابات معطلة، هناك العديد من المعلمات التي يمكن استدعاؤها للبحث الفعال في الإعلان والإبلاغ عن نتائج المطابقة ومع ذلك، لتسليط الضوء على وظائفها، قررت استخدام الحسابات المعطلة لأنها أمر cmdlet مهم ومفيد يجب معرفته.

❖ إجراء عمليات باستخدام PowerShell

هناك ثلاث عمليات يتم إجراؤها في بيئة Active Directory: إنشـــاء وتعديل وحذف، يمكن لأي مسؤول Active Directory لديه أذونات كافية تنفيذ عمليات الإنشاء والتعديل والحذف.

يمكن إجراء العمليات على كائنات مثل المستخدمين وأجهزة الكمبيوتر وخصائص المستخدم والكمبيوتر وجهات الاتصال والكائنات الأخرى باستثناء كائنات Active Directory الهامة، بشكل افتراضي ، لايملك المستخدمون (بما في ذلك مسؤولي المجال) أذونات لإجراء أي عمليات على كائنات Active Directory الهامة.

في بيئة Active Directory حيث توجد ملايين العناصـــر، يصـــعب أحياناً مراقبة التغييرات التي تم إجراؤها، قد يتم إجراء آلاف العمليات يومياً.

على سبيل المثال، قد يقوم شخص ما بإنشاء حسابات مستخدمين ، بينما قد يقوم شخص آخر بحذف حسابات المستخدمين، وبالمثل ، قد يكون لديك شخص يعدل خصائص المستخدم،

نتيجة لذلك ، يصبح من الضروري تنفيذ آلية تدقيق تساعدك على تتبع تغييرات Active Directory وتساعدك على تحديد المستخدمين الذين أجروا التغييرات.

كان Windows Server 2008 وأنظمة التشـــغيل الســـابقة يفتقران إلى القدرة على الإبلاغ عن تفاصيل كافية عن التغييرات التي تم إجراؤها في Active Directory

أمثلة :

Example 1: Set properties for a user

Set-ADUser -Identity ChewDavid -HomePage 'http://fabrikam.com/employees/ChewDavid' -LogonWorkstations 'ChewDavid-DSKTOP,ChewDavid-LPTOP'

Example 2: Set properties for multiple users

Get-ADUser -Filter 'Name -like "*"' -SearchBase 'OU=HumanResources,OU=UserAccounts,DC=FABRIKAM,DC=COM' -Properties DisplayName | % {Set-ADUser $_ -DisplayName ($_.Surname + ' ' + $_.GivenName)}

Example 3: Set properties

Set-ADUser -Identity GlenJohn -Replace @{title="director";mail="glenjohn@fabrikam.com"}

Example 4: Modify a user otherMailbox property

Set-ADUser -Identity GlenJohn -Remove @{otherMailbox="glen.john"} -Add @{url="fabrikam.com"} -Replace @{title="manager"} -Clear description

Example 5: Set user properties to a local instance

$User = Get-ADUser -Identity GlenJohn -Properties mail,department

$User.mail = "glen@fabrikam.com"

$User.department = "Accounting"

Set-ADUser -Instance $User

Example 6: Set attributes for a user

```
$Hours = New-Object byte[] 21

$Hours[5] = 255; $Hours[8] = 255; $Hours[11] = 255; $Hours[14] = 255; $Hours[17] = 255;

$Hours[6] = 1; $Hours[9] = 1; $Hours[12] = 1; $Hours[15] = 1; $Hours[18] = 1;

$ReplaceHashTable = New-Object HashTable

$ReplaceHashTable.Add("logonHours", $Hours)

$ReplaceHashTable.Add("description", "Sarah Davis can only logon from Monday through Friday from 8:00 AM to 5:00 PM")

Set-ADUser -Identity "SarahDavis" -Replace $ReplaceHashTable
```

يعيّن هذا المثال ساعات تسجيل دخول المستخدم إلى الاثنين إلى الجمعة من ٨:٠٠ صباحاً إلى ٥:٠٠ مساءً ويضيف وصفاً، يقوم بتحديث سمة logonHours مع صفيف البايت المحدد وسمة الوصف بالسلسلة المحددة.

Example 7: Set a property for a user

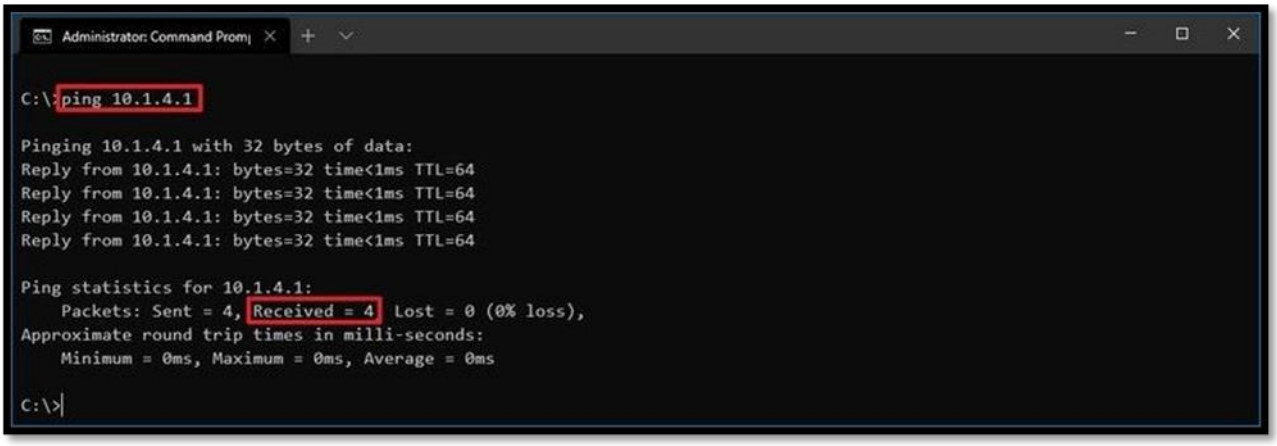
```
$Manager = Get-ADUser -Identity GlenJohn -Server Corp-DC01

Set-ADUser -Identity ChewDavid -Manager $Manager -Server Branch-DC02
```

Example 8: Get a user and set a property

```
Get-ADUser -Identity "DavidChew" | Set-ADUser -Manager "ElisaDaugherty"
```

تمرين :

أختبر هذا الأمرping IP-OR-DOMAIN للاتصال بين الجهاز المحلي وجهاز التوجيه مستخدمة محرر الأوامر PowerShell؟



الفصل الخامس: بروتوكولات التكوين الديناميكي للمضيف DHCP

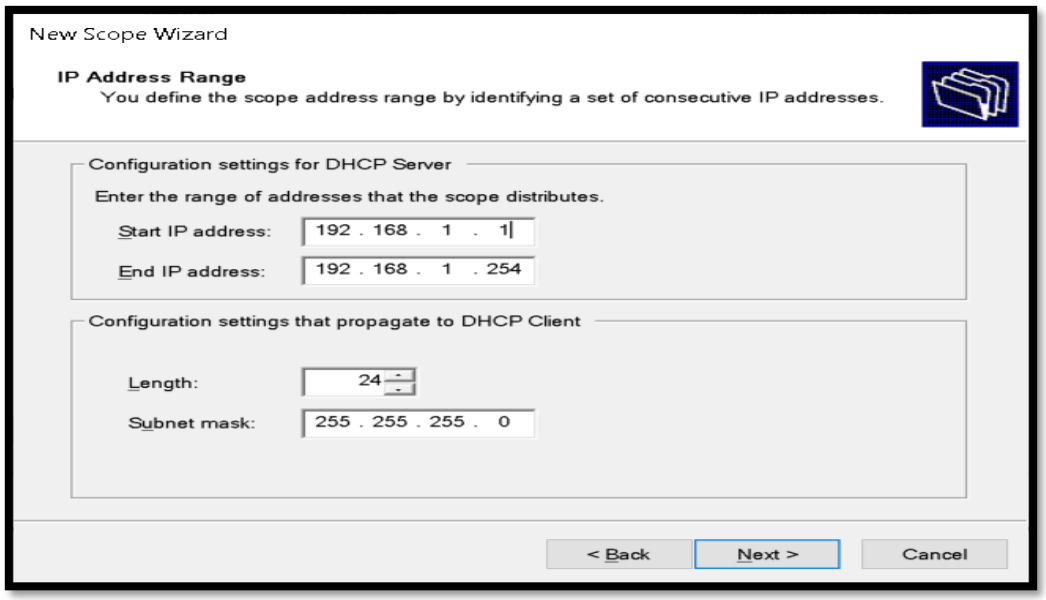
❖ **لإنشاء وتكوين نطاق، يجب عليك تعريف الخصائص التالية:**

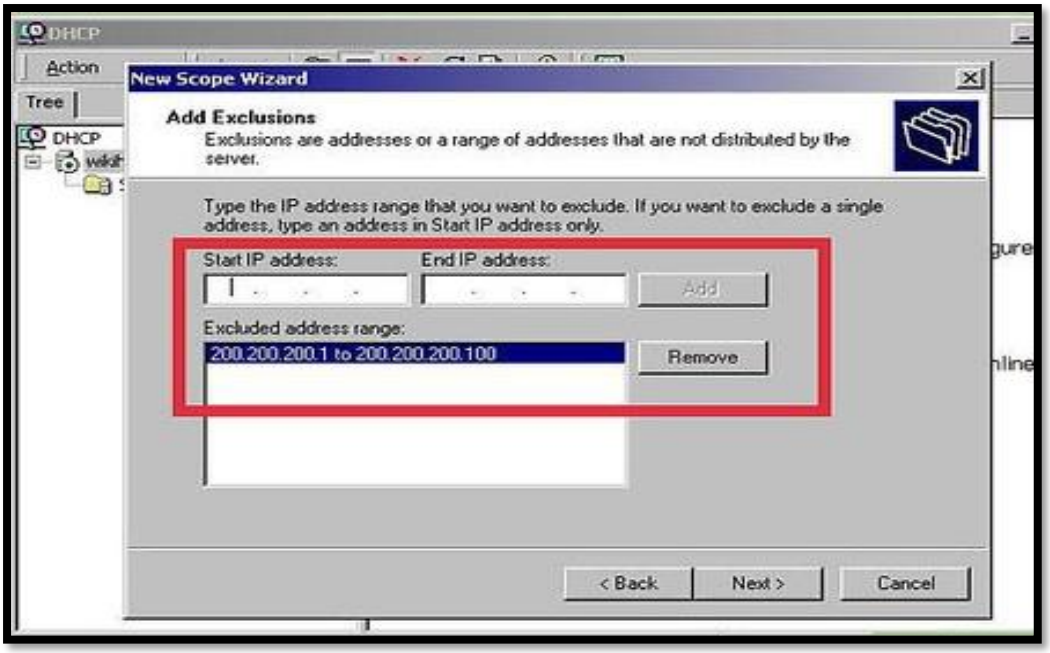
- الاسم والوصف، تعرّف هذه الخاصية النطاق، الاسم إلزامي.
- نطاق عنوان IP تسرد هذه الخاصية نطاق العناوين المتوفرة للإيجار، هذه الخاصية إلزامية.
- قناع الشبكة الفرعية تستخدم أجهزة الكمبيوتر العميلة هذه الخاصية لتحديد موقعها في البنية الأساسية لشبكة المؤسسة، هذه الخاصية إلزامية.
- الاستثناءات، تسرد هذه الخاصية عناوين مفردة أو كتلاً من العناوين الموجودة ضمن نطاق عنوان IP، ولكن لن تكون متوفرة للإيجار، هذه الخاصية اختيارية.
- تأخير تشير هذه الخاصية إلى مقدار الوقت الذي يجب تأخيره قبل إرسال DHCPOFFER الإعداد الافتراضي هو صفر مللي ثانية.
- مدة التأجير تسرد هذه الخاصية مدة التأجير، استخدم مُدد أقصر للنطاقات التي لها عناوين IP محدودة واستخدم مُدد أطول للشبكات أكثر ثباتاً.
- الخيارات يمكنك تكوين العديد من الخصائص الاختيارية على نطاق،

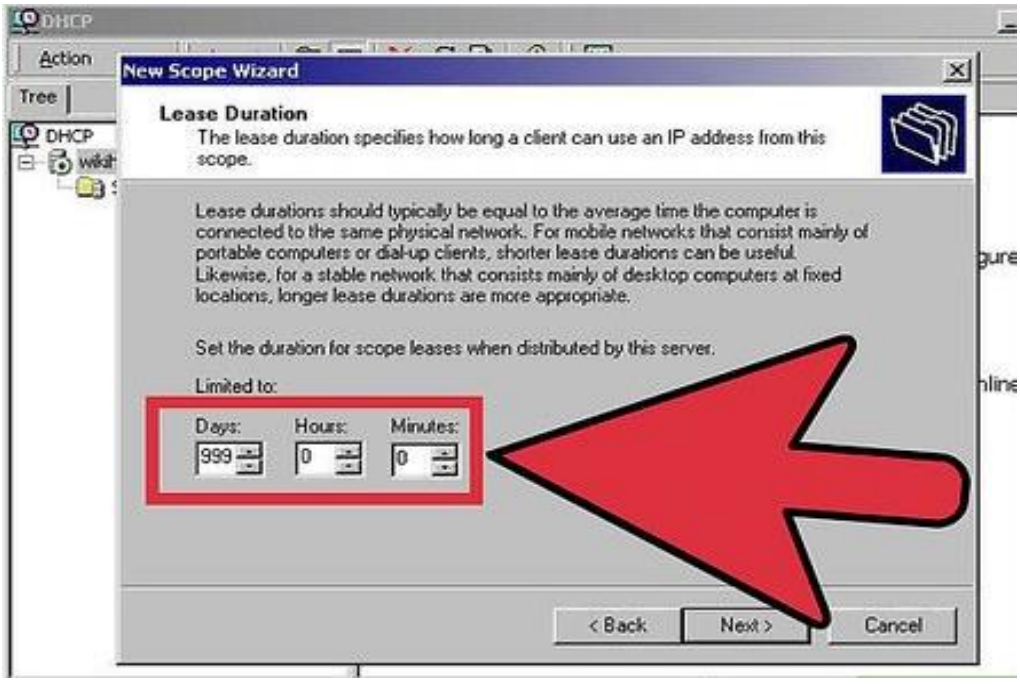
ولكن عادةً ما تقوم بتكوين الخصائص التالية:

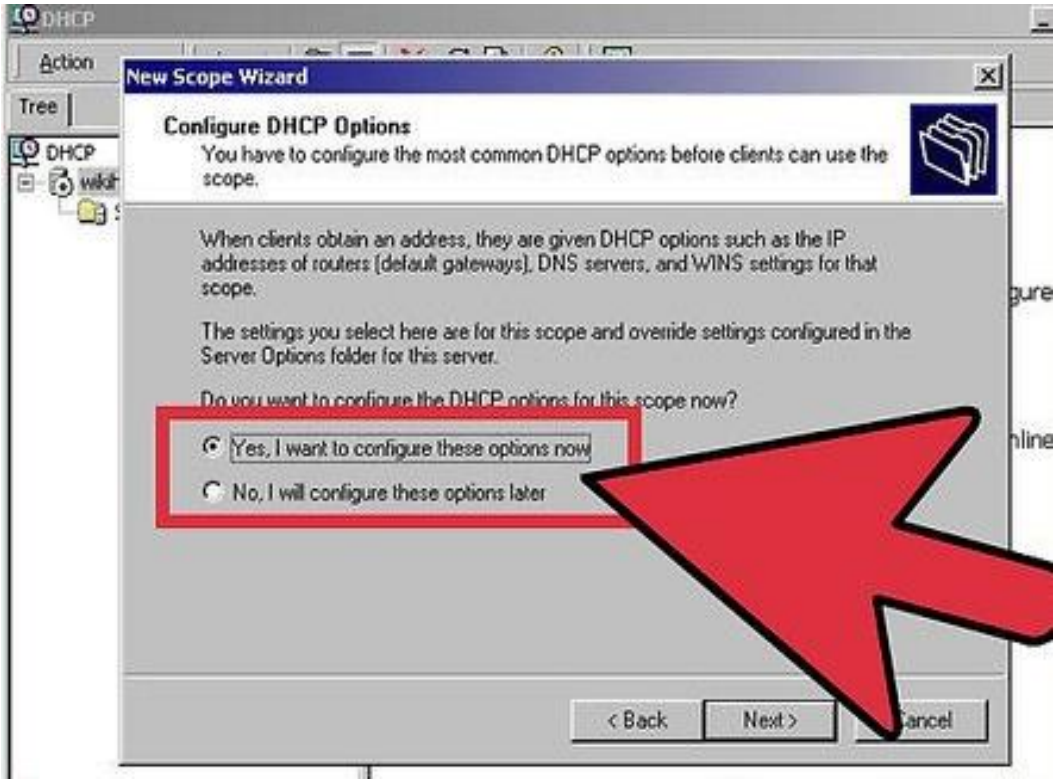
- الخيار ٠٠٣ جهاز التوجيه (البوابة الافتراضية للشبكة الفرعية)
- الخيار ٠٠٦ خوادم DNS
- الخيار ٠١٥ لاحقة DNS

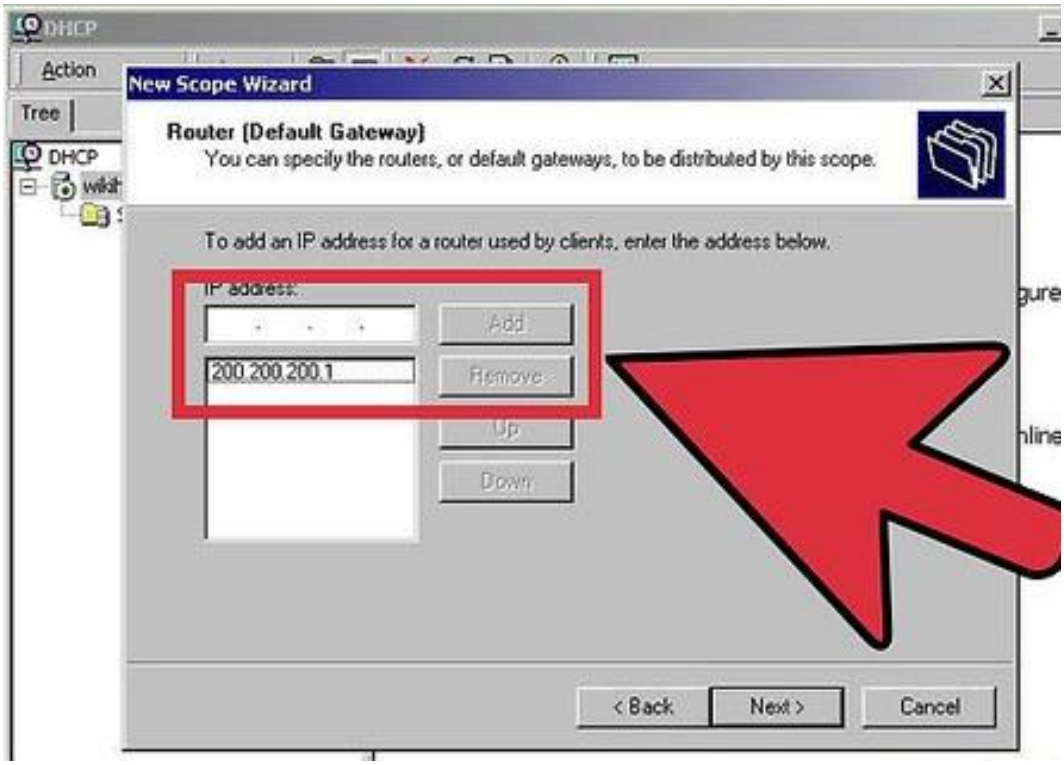
- التنشيط يجب تنشيط النطاق قبل أن تتمكن من تأجير عناوين IP .

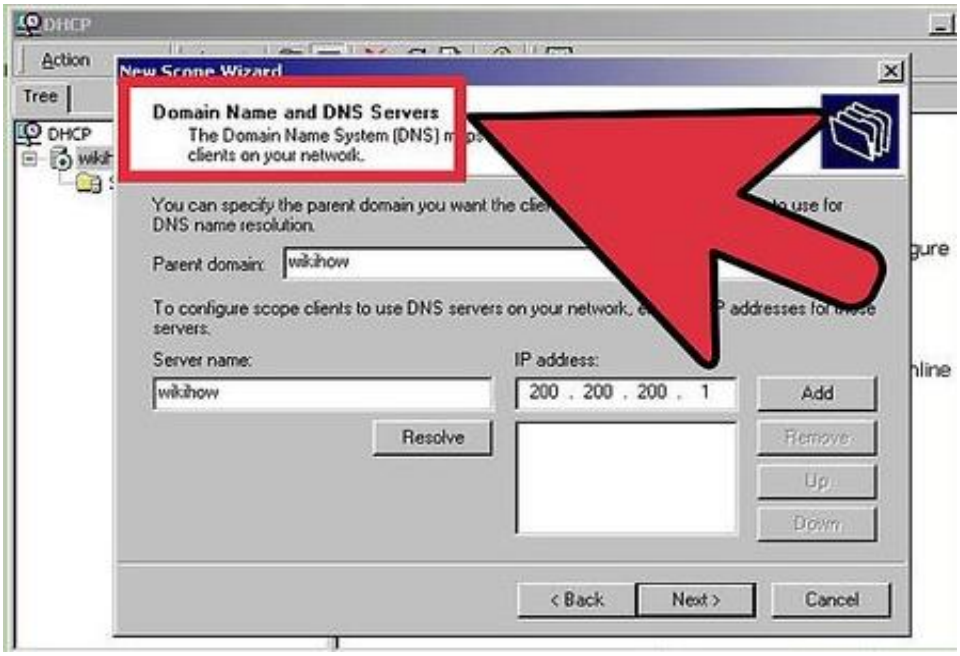

New Scope Wizard
IP Address Range
You define the scope address range by identifying a set of consecutive IP addresses.
Configuration settings for DHCP Server
Enter the range of addresses that the scope distributes.
Start IP address: 192 . 168 . 1 . 1
End IP address: 192 . 168 . 1 . 254
Configuration settings that propagate to DHCP Client
Length: 24
Subnet mask: 255 . 255 . 255 . 0
< Back
Next >
Cancel

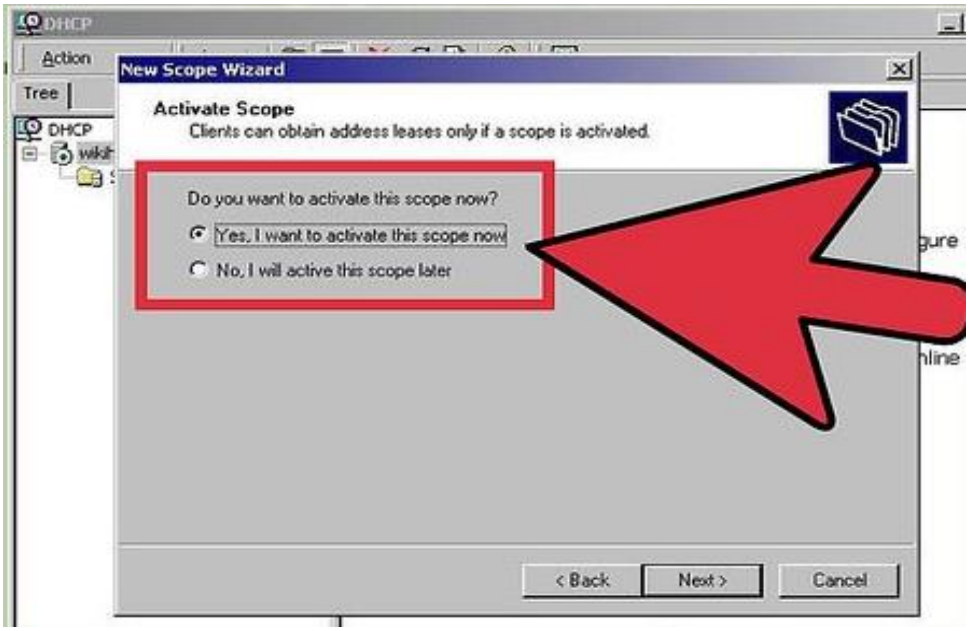

DHCP
Action
Tree
New Scope Wizard
Add Exclusions
Exclusions are addresses or a range of addresses that are not distributed by the server.
Type the IP address range that you want to exclude. If you want to exclude a single address, type an address in Start IP address only.
Start IP address:
End IP address:
Add
Excluded address range:
200.200.200.1 to 200.200.200.100
Remove
< Back
Next >
Cancel

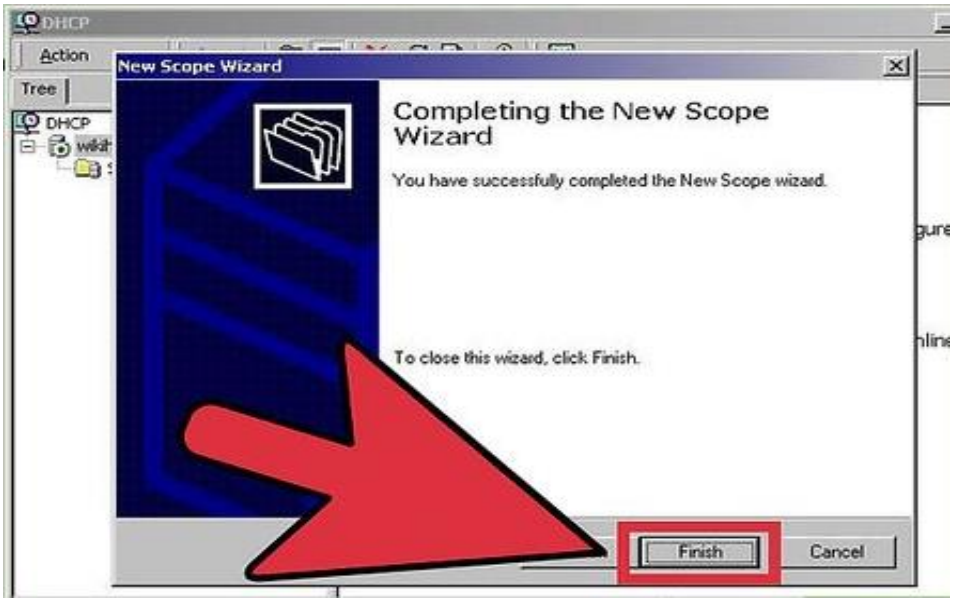

DHCP
Action
Tree
New Scope Wizard
Lease Duration
The lease duration specifies how long a client can use an IP address from this scope.
Lease durations should typically be equal to the average time the computer is connected to the same physical network. For mobile networks that consist mainly of portable computers or dial-up clients, shorter lease durations can be useful. Likewise, for a stable network that consists mainly of desktop computers at fixed locations, longer lease durations are more appropriate.
Set the duration for scope leases when distributed by this server.
Limited to:
Days: 999
Hours: 0
Minutes: 0
< Back
Next >
Cancel


DHCP
Action
Tree
DHCP
New Scope Wizard
Configure DHCP Options
You have to configure the most common DHCP options before clients can use the scope.
When clients obtain an address, they are given DHCP options such as the IP addresses of routers (default gateways), DNS servers, and WINS settings for that scope.
The settings you select here are for this scope and override settings configured in the Server Options folder for this server.
Do you want to configure the DHCP options for this scope now?
Yes, I want to configure these options now
No, I will configure these options later
< Back
Next >
Cancel


DHCP
Action
Tree
DHCP
New Scope Wizard
Router (Default Gateway)
You can specify the routers, or default gateways, to be distributed by this scope.
To add an IP address for a router used by clients, enter the address below.
IP address:
200.200.200.1
Add
Remove
Up
Down
< Back
Next >
Cancel

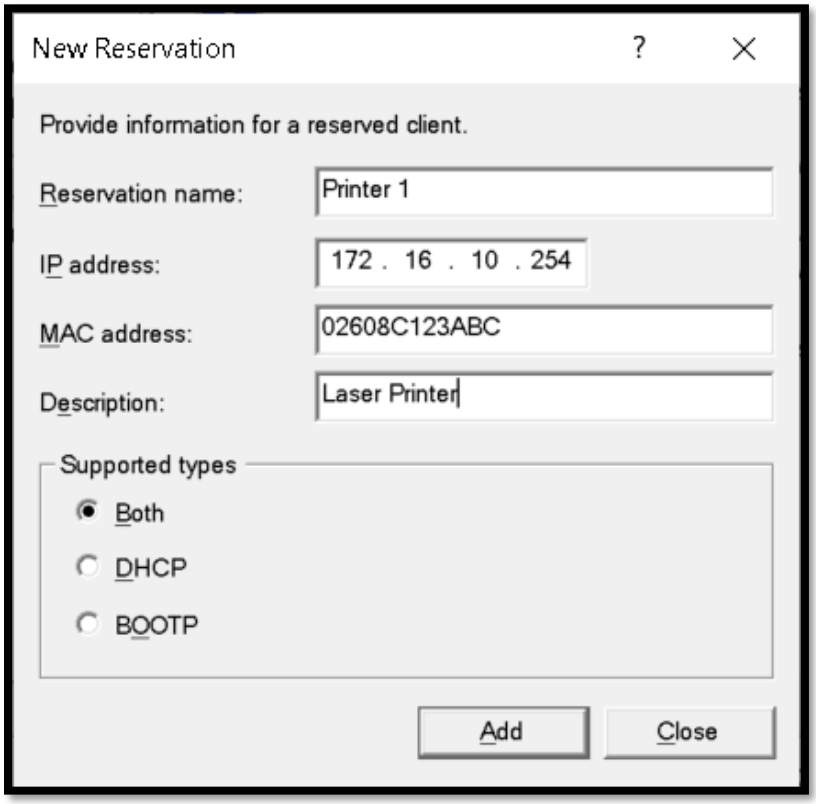

DHCP
Action
Tree
New Scope Wizard
Domain Name and DNS Servers
The Domain Name System (DNS) m
clients on your network.
You can specify the parent domain you want the clie
DNS name resolution.
Parent domain:
wikihow
To configure scope clients to use DNS servers on your network, e
servers.
Server name:
IP address:
wikihow
200 . 200 . 200 . 1
Add
Resolve
Remove
Up
Down
< Back
Next >
Cancel


DHCP
Action
Tree
New Scope Wizard
Activate Scope
Clients can obtain address leases only if a scope is activated.
Do you want to activate this scope now?
Yes, I want to activate this scope now
No, I will active this scope later
< Back
Next >
Cancel


DHCP
Action
Tree
New Scope Wizard
Completing the New Scope Wizard
You have successfully completed the New Scope wizard.
To close this wizard, click Finish.
Finish
Cancel

❖ إدارة قاعدة البيانات DHCP

إذا كنت تريد كمبيوتر أو جهازاً للحصـــول على عنوان معين من مدى النطاق، يمكنك حجز هذا العنوان بشـكل دائم لتعيينه إلى هذا الجهاز في DHCP والحجوزات مفيدة لتتبع عناوين IP التي تم تعيينها إلى أجهزة مثل الطابعات، لإنشـــاء حجز، حدد النطاق في وحدة تحكم **DHCP** ومن القائمة **Action** حدد **New Reservation**



يجب توفير المعلومات التالية لإنشاء الحجز في مربع الحوار New Reservation :

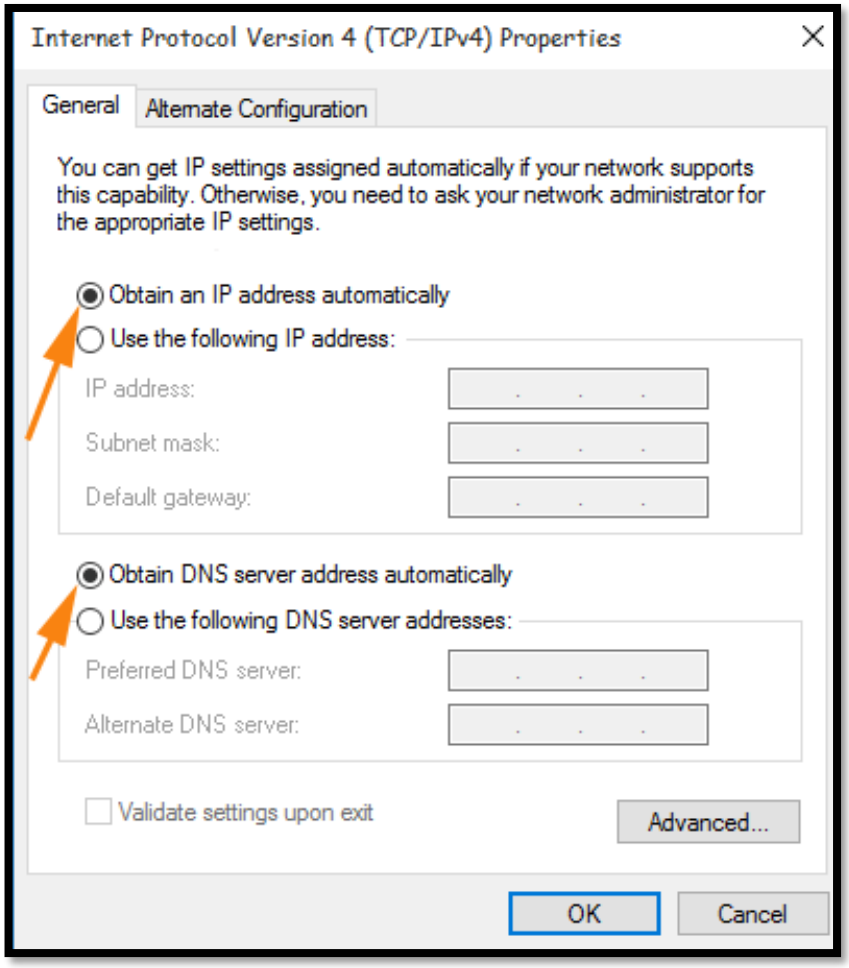
- اسم الحجز، اسم مألوف للإشارة إلى الحجز.
- عنوان IP عنوان IP من النطاق الذي تريد تعيينه إلى الجهاز.
- عنوان MAC عنوان MAC للواجهة التي تريد تعيين العنوان لها.
- الوصف حقل اختياري يمكنك فيه تقديم تعليق حول الحجز.

ملحوظة :

إذا حصل عميل بالفعل على عنوان IP من خادم DHCP ، يمكنك تحويل التأجير الموجودة إلى حجز في وحدة تحكم DHCP

إضافة عنوان الشبكة للأجهزة عن طريق الـ DHCP :

كيفية تعيين عنوان الشبكة بشكل تلقائي للأجهزة، وهذا هو الاختيار الافتراضي دائماً كما بالشكل التالي



❖ تأمين ومتابعة DHCP

نظراً لعدم وجود آلية مصـــادقة بين خوادم DHCP وعملاء DHCP ، فبمجرد تكوين خادم DHCP حديثًا على الشــبكة ، يمكنه تخصــيص عناوين IP ومعلمات الشــبكة الأخرى لعملاء DHCP حتى لو كانت مزيفة، يتم توصيل خادم DHCP الزائف بجهاز WLAN التجميعي عبر شبكة من الطبقة الثانية، عندما يطبق عميل متصـــل بجهاز WLAN على عنوان IP من خلال DHCP ، ويســـتجيب خادم DHCP الزائف قبل الخوادم الأخرى ويعين عنوان IP للعميل ، ســـيحدث تعارض في عنوان IP على الشبكة ، مما يؤثر على خدمات الشبكة.

للدفاع ضـــد الهجمات الســـابقة، قم بتمكين تطفل DHCP على جهاز WLAN وقم بتكوين الواجهة المتصلة بخادم DHCP الصالح كواجهة موثوقة لتصفية خوادم DHCP المارقة.

قم بتكوين فحص صلاحية الحزمة بناءً على تطفل DHCP

Configure the interface connected to the valid DHCP server as a trusted interface.

<HUAWEI> system-view

[HUAWEI] dhcp enable

[HUAWEI] dhcp snooping enable

[HUAWEI] interface gigabitethernet 0/0/1

[HUAWEI-GigabitEthernet0/0/1] dhcp snooping enable

[HUAWEI-GigabitEthernet0/0/1] dhcp snooping trusted

[HUAWEI-GigabitEthernet0/0/1] quit

قم بتمكين تطفل DHCP على واجهة مستخدم أخرى أو في شبكة محلية ظاهرية

Enable DHCP snooping on another user-side interface or in a VLAN.

[HUAWEI] interface gigabitethernet 0/0/2

[HUAWEI-GigabitEthernet0/0/2] dhcp snooping enable

- **الدفاع ضد هجمات فيضانات DHCP**

عندما يتلقى جهاز WLAN يعمل كخادم DHCP أو وكيل ترحيل عددًا كبيراً من حزم DHCP المرسلة من قبل مستخدم ضار ، لايمكن لجهاز WLAN معالجة حزم DHCP الصالحة لأنه يركز قدرة معالجة DHCP الخاصة به على هذه الحزم الضارة، نتيجة لذلك ، لايمكن للعملاء الحصول على عناوين IP أو تجديدها .

للدفاع ضد الهجمات السابقة، قم بتكوين سياسات الأمان التالية على جهاز WLAN:

- **DHCP port-level protection**

يراقب جهاز WLAN معدل حزم DHCP بناءً على المنافذ، عندما يتجاوز معدل حزم DHCP المرسلة إلى مستوى التحكم من منفذ واحد العتبة المحددة ، يرسل الجهاز حزم DHCP هذه إلى مستوى التحكم عبر قناة مستقلة، هذه الوظيفة تتجنب تأثير الهجوم على حزم DHCP الصالحة.

<HUAWEI> system-view

[HUAWEI] dhcp enable

[HUAWEI] dhcp snooping enable

[HUAWEI] interface gigabitethernet 0/0/1

[HUAWEI-GigabitEthernet0/0/1] dhcp snooping enable

[HUAWEI-GigabitEthernet0/0/1] dhcp snooping check dhcp-rate enable

[HUAWEI-GigabitEthernet0/0/1] dhcp snooping check dhcp-rate 30

[HUAWEI-GigabitEthernet0/0/1] quit

- **DHCP user-level protection**

يراقب جهاز WLAN معدل حزم DHCP المرسلة إلى مستوى التحكم بناءً على المستخدمين (عناوين MAC أو IP) عندما يتجاوز معدل حزم DHCP من مســـتخدم الحد المعين، يتجاهل الجهاز حزم DHCP الخاصة بهذا المستخدم لفترة زمنية معينة.

<HUAWEI> system-view

[HUAWEI] cpu-defend policy antiatk

[HUAWEI-cpu-defend-policy-antiatk] auto-defend enable

[HUAWEI-cpu-defend-policy-antiatk] auto-defend threshold 30

[HUAWEI-cpu-defend-policy-antiatk] undo auto-defend trace-type source-portvlan

[HUAWEI-cpu-defend-policy-antiatk] undo auto-defend protocol tcp telnet ttl-expired igmp icmp dhcpv6 mld nd

[HUAWEI-cpu-defend-policy-antiatk] auto-defend action deny timer 300

[HUAWEI-cpu-defend-policy-antiatk] quit

[HUAWEI] cpu-defend-policy antiatk

تمرين :

اكتب صح امام العبارة الصحيحة وخطأ امام العبارة الخاطئة

١. في إنشاء وتكوين النطاق يجب ان تعرف الاسم والوصف، تعرّف هذه الخاصية النطاق(صح)

٢. من الخصائص لإنشاء وتكوين نطاق نطاق عنوان IP تسرد هذه الخاصية نطاق العناوين المتوفرة للإيجار، هذه الخاصية غيرإلزامية (خطأ)

٣. في إنشاء وتكوين النطاق ان قناع الشبكة الفرعية تستخدم أجهزة الكمبيوتر العميلة هذه الخاصية لتحديد موقعها في البنية الأساسية لشبكة المؤسسة، هذه الخاصية إلزامية (صح)

٤. في إنشاء وتكوين النطاق ان الاستثناءات، تسرد هذه الخاصية عناوين مفردة أو كتلاً من العناوين الموجودة ضمن نطاق عنوان IP، ولكن لن تكون متوفرة للإيجار، هذه الخاصية الزامية(خطأ)

٥. في إنشاء وتكوين النطاق ان التأخير يشير في هذه الخاصية إلى مقدار الوقت الذي يجب تأخيره قبل إرسال DHCPOFFER الإعداد الافتراضي هو صفر مللي ثانية(صح)

٦. في إنشاء وتكوين النطاق ان مدة التأجير تسرد هذه الخاصية مدة التأجير، استخدم مُدد أقصر للنطاقات التي لها عناوين IP محدودة واستخدم مُدد أطول للشبكات أكثر مرونه(خطأ)

٧. في إنشاء وتكوين نطاق ان الخيارات يمكنك من تكوين العديد من الخصائص الاختيارية على نطاق(صح)

٨. في إدارة قاعدة البيانات DHCPإذا كنت تريد كمبيوتر أو جهازاً للحصول على عنوان معين من مدى النطاق، يمكنك حجز هذا العنوان بشكل دائم لتعيينه إلى هذا الجهاز فيDHC (صح)

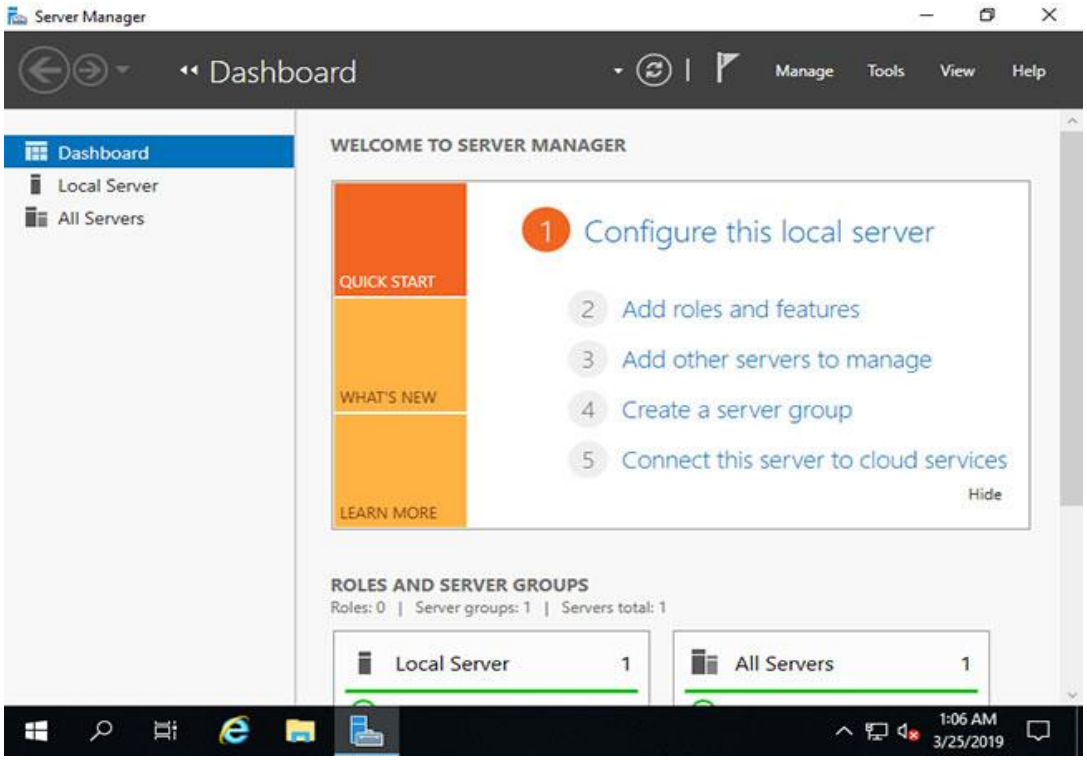
٩. في تأمين ومتابعة DHCP اذا حدث تعارض في عنوان IP على الشبكة ، سوف يؤثر على خدمات الشبكة(صح)

١٠. إذا حصل عميل بالفعل على عنوان IP من خادم DHCP ، فأنه لا يمكنك تحويل التأجير الموجودة إلى حجز في وحدة تحكم DHCP (خطأ)

الفصل السادس : نظام أسماء النطاقات DNS

تنصيب وإدارة خادم DNS

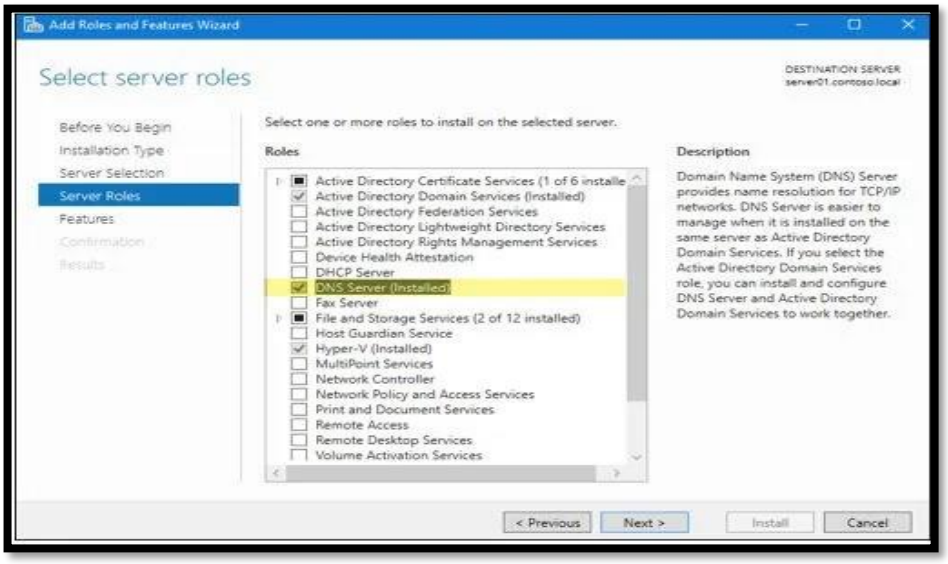
شغّل برنامج مدير الخادم Server Manager من صفحة البداية، حيث تجده في القسم الأيسر منها.



نضغط على Add roles and features

Server selection

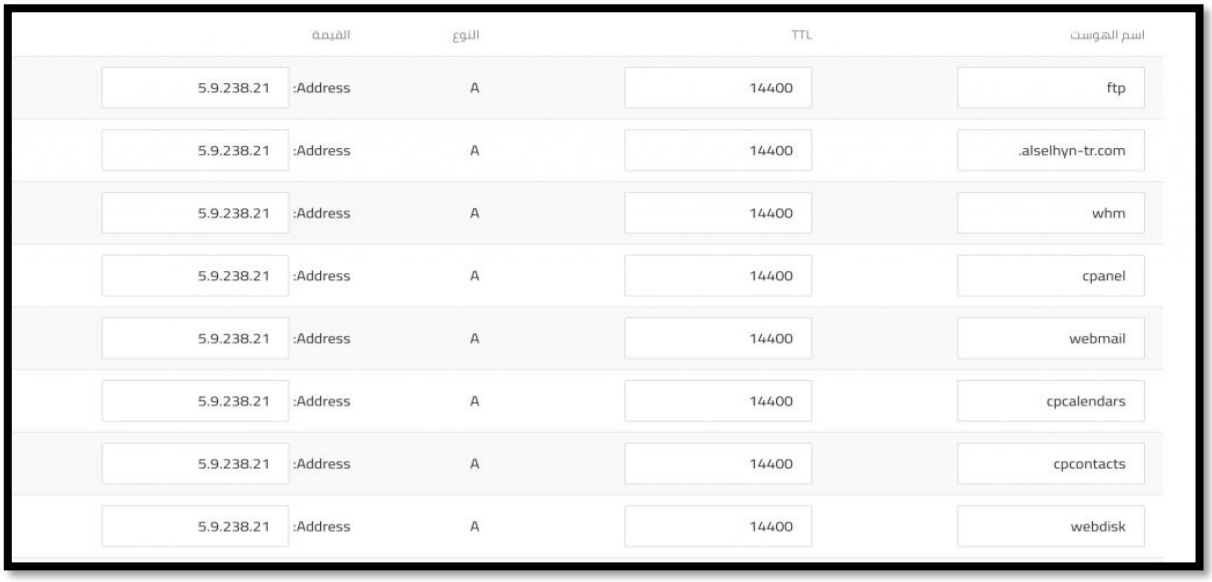
نختار اسم الخادم الذي سيتم التثبيت عليه وبالضغط على Next تظهر الشاشة التالية ثم نختار DNS server



❖ إدارة مناطق DNS

تساعد سجلات الـ DNS على تعريف ملكية الدومين وهوية المرسل في الإيميلات وغيره , وتتنوع سجلات الـ DNS وهي كثيرة وأكثر أنواع سجلات الـ DNS استخداماً في المواقع الإلكترونية هي:

1-سجل :A Record

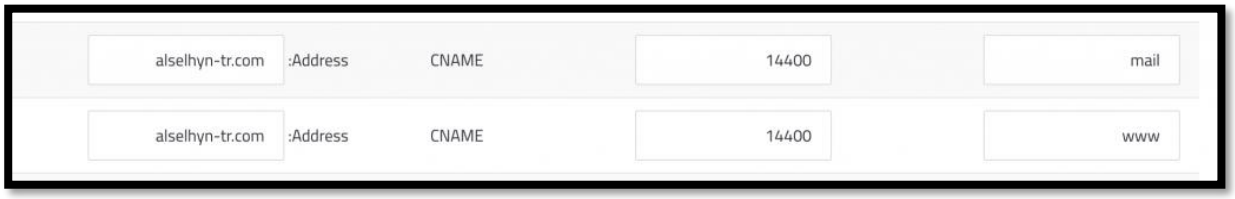


اسم الهوست	TTL	النوع	القيمة	
ftp	14400	A	:Address	5.9.238.21
.alselhyn-tr.com	14400	A	:Address	5.9.238.21
whm	14400	A	:Address	5.9.238.21
cpanel	14400	A	:Address	5.9.238.21
webmail	14400	A	:Address	5.9.238.21
cpcalendars	14400	A	:Address	5.9.238.21
cpcontacts	14400	A	:Address	5.9.238.21
webdisk	14400	A	:Address	5.9.238.21

من أشهر أنواع السجلات وأكثرها استخداماً وهو المسؤول عن اتصال الدومين بـ IP السيرفر أو الموقع بشكل مباشر والمسؤول عن انشاء دومينات فرعية Sub Domain

كما يسمح لك بربط عدة دومينات فرعية من الدومين الرئيسي بعدد مختلف من IP السيرفرات إذن هو ببساطة المسؤول عن ربط الدومين والـIP وهو الذي يسهل على الناس كتابة اسم الدومين للدخول الى الموقع بدل كتابة الـ IP الصعب حفظه بشكل مباشر.

2-سجل :CNAME



يستعمل سجل CNAME في توجيه دومين أو دومين فرعي الى مجال اخر

ويختلف عن سجل A Record بأنه مسؤول عن توجيه المجالات والدومينات الى دومين او مجال اخر وليس الى IP

يتم استخدامه ايضاً في www للدومين كما نلاحظ في الصورة

بحيث اذا ذهبنا مثلاً الى www.domain.com سيفتح موقع domain.com دون أي تأثير وهذه وظيفة سجل الـ CName

3-سجل :MX Record

ويعني سجل Mail Exchanger (MX) أي مبادل البريد أو المسؤول عن توجيه الايميلات لهذا الدومين الى خادم البريد

بحيث يمكن توجيه الايميلات الى أكثر من خادم بريد لاستقبال وارسال الإيميلات

ويختلف MX عن سجلات A و Cname بأنه يحوي خيار الأولوية بحيث يمكن ادخال اكثر من خادم بريد واحد وتوجيه الرسائل إليهم حسب الأولوية

بحال عدم قدرة خادم على معالجة طلبات ارسال واستقبال البريد الإلكتروني يتم توجيه إلى الخادم الثاني مباشرة لإكمال العملية عادة يكون الخادم الافتراضي للايميلات في لوحة تحكم Cpanel هو اسم الدومين نفسه كما نلاحظ في الصورة .

4 -سجل :TXT

سجل TXT المسؤول عن إدخال بيانات نصية ضمن سجلات الدومين وعادة يستخدم في تعريف ملكية الدومين مثل إثبات الدومين في احصائيات جوجل

وإثبات الدومين في استخدام ايميلات من الخادم

كما نلاحظ في الصورة ايضاً اثبات نص IP الخادم ضمن السجل اللنصي لاستخدام وإثبات الايميلات

5-سجل :NS

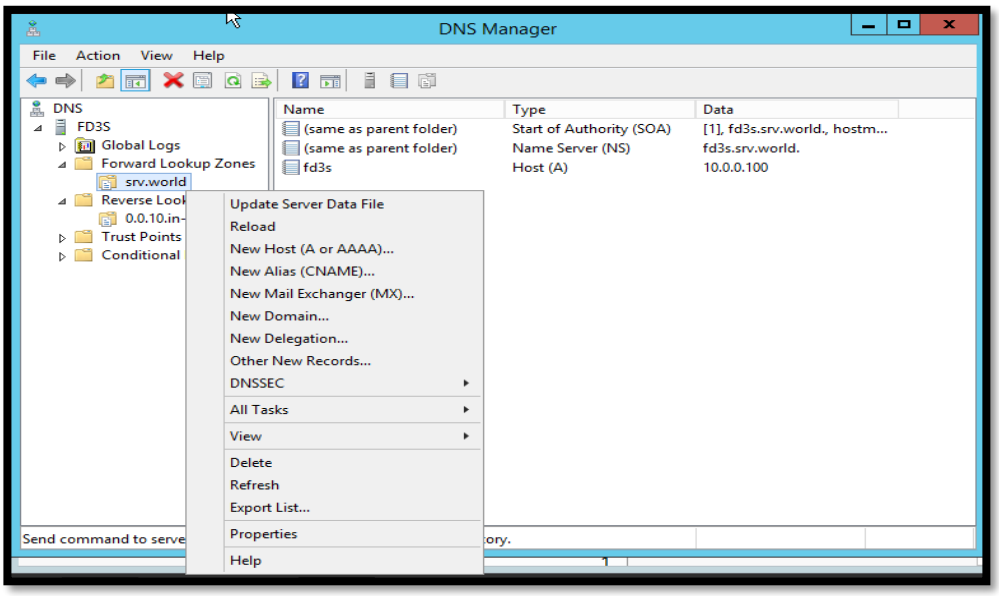
سجل NS المسؤول عن تعيين النيم سيرفر Nameserver للدومين وهو المسؤول عن ربط الدومين بالسيرفر لإستعراض ملفات الموقع وعمل الإيميلات وغيره

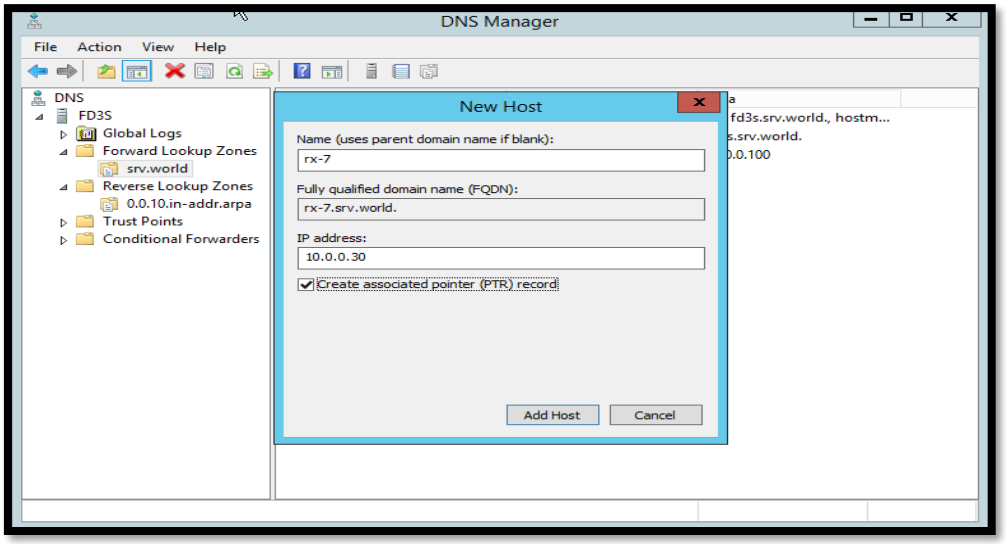
وعادة مانراه في اثناء تسجيلنا ضمن استضافة جديدة مثلاً في استضافة كواليتي

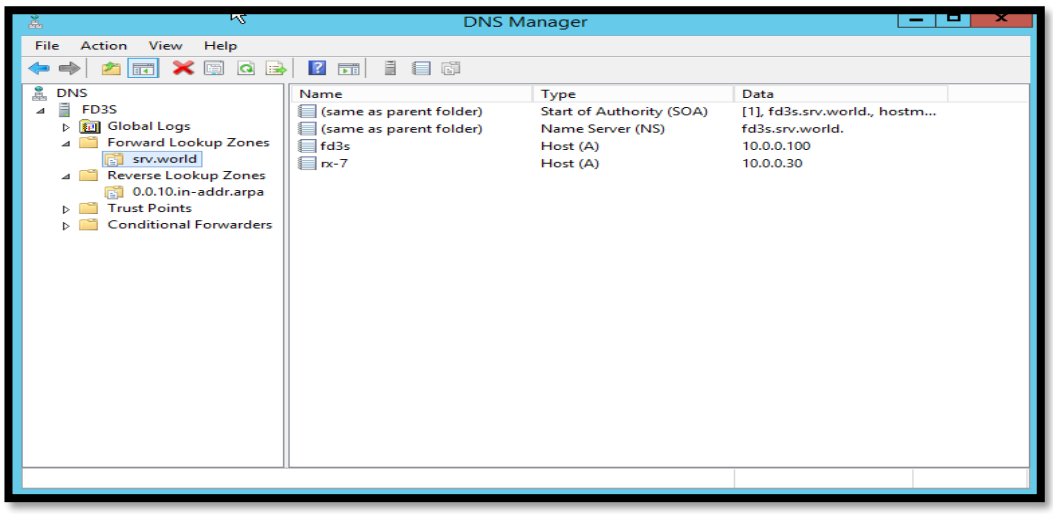
ns1.qualiteyserver.com

ns2.qualiteyserver.com

كيفية إضافة Record







تمرين:

قم بتنصيب خادم DNS متبعاً الخطوات التي قمت بدراستها ؟

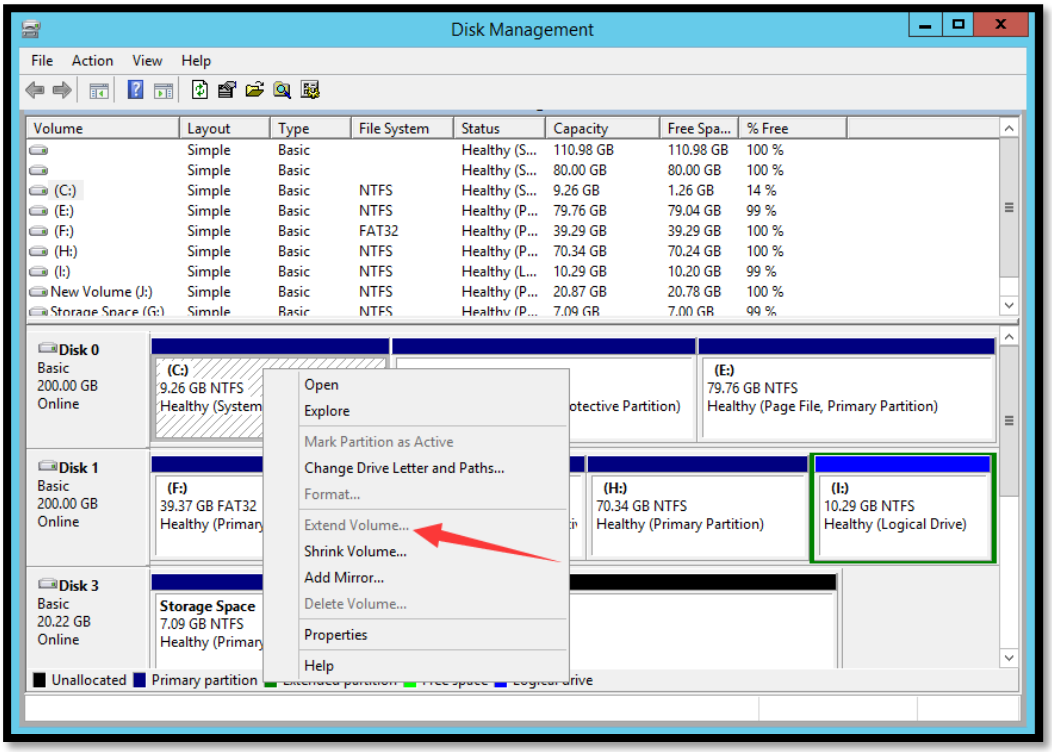
الفصل السابع: تكنولوجيا التخزين المحلي

❖ إدارة الأقراص وأحجامها

يمكننا إدارة مساحة القرص الصلب من الأداة Disk management

والتي يتم الوصول عليها بالضغط على علامة windows (Right Click) ثم اختيار

Disk management



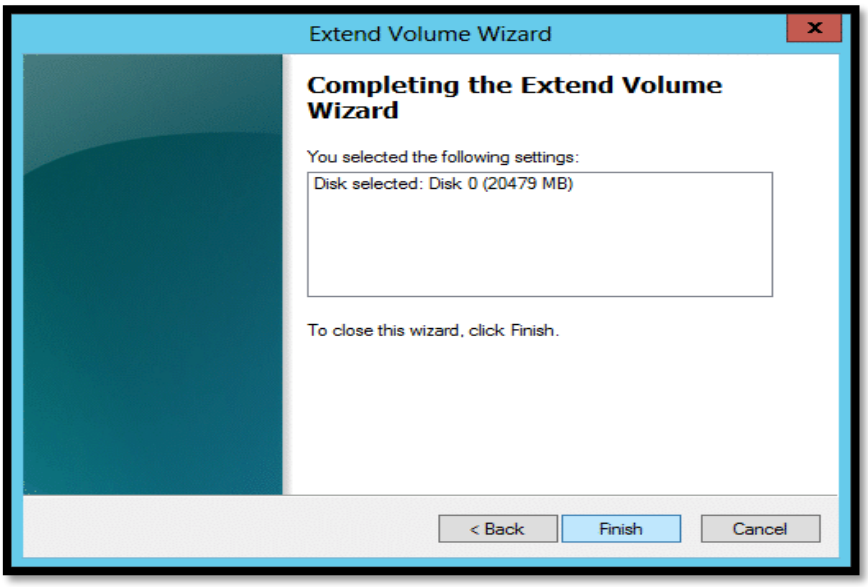
يتم تخصـــيص حجم محرك الأقراص C: أثناء تثبيت نظام التشـــغيل ، ولكن من المحتمل جداً أن تنفد المســـاحة بعد فترة من الوقت، كثير من الناس يســـألون ما إذا كان من الممكن إعادة تقســـيم محركات الأقراص الثابتة Server 2012 دون البدء من جديد، نعم طبعا.

خطوات إعادة تقسيم القرص الصلب باستخدام Server 2012 Disk Management:

- **لتوسيع القرص الصلب:**

انقر بزر الماوس الأيمن على القسم ذي المساحة غير المخصصة المتجاورة وحدد توسيع نطاق التداول.

ببساطة انقر التالي إلى نهاية في نافذة توسيع معالج الصوت.

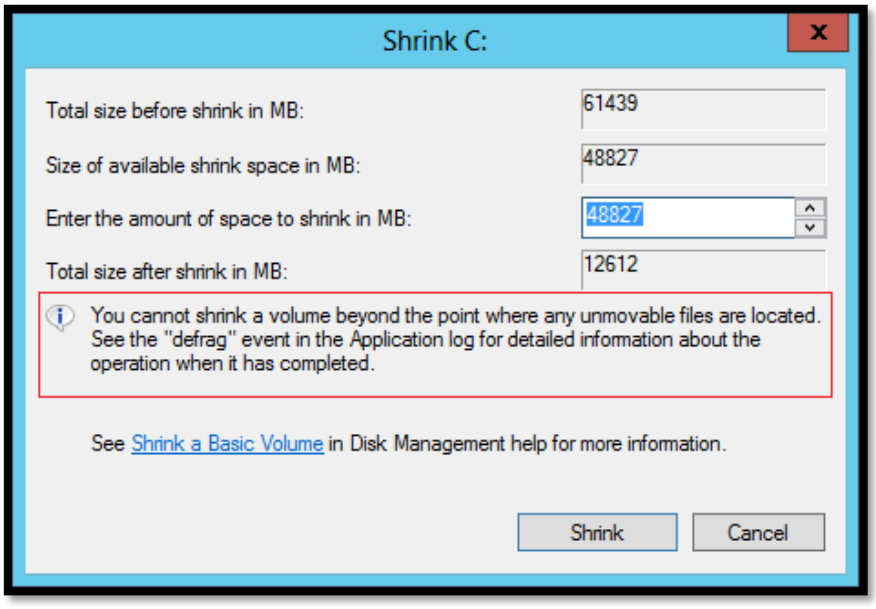


لتقليص القرص الصلب:

صحافة و على لوحة المفاتيح، اكتب diskmgmt.msc والصحافة أدخل لفتح إدارة القرص،

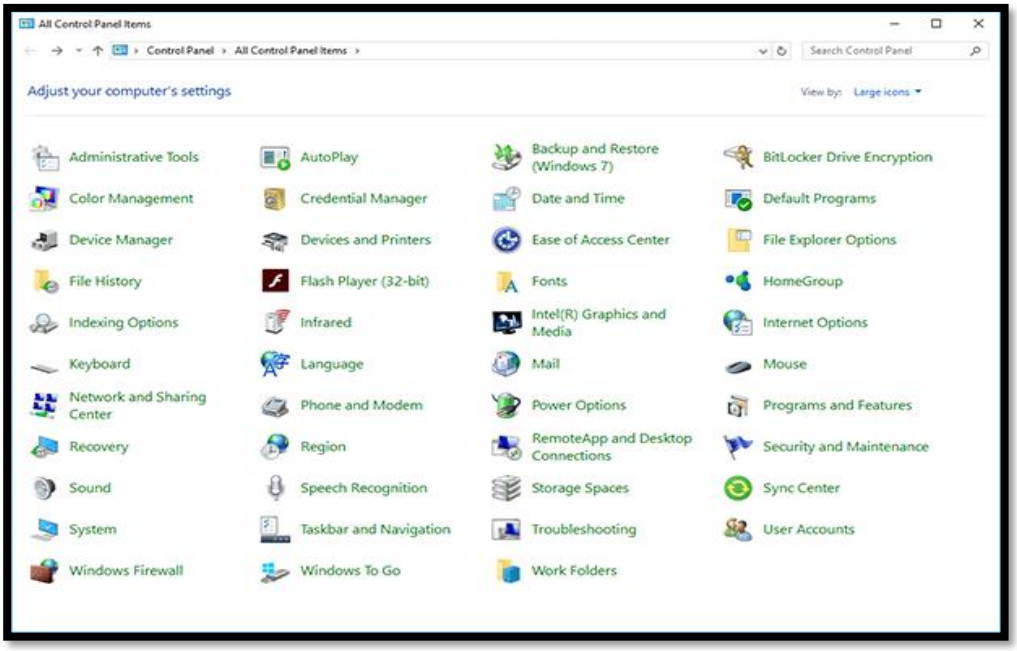
انقر بزر الماوس الأيمن على محرك الأقراص وحدد تقليص الحجم

أدخل مقدار المساحة وانقر انكماش للبدء



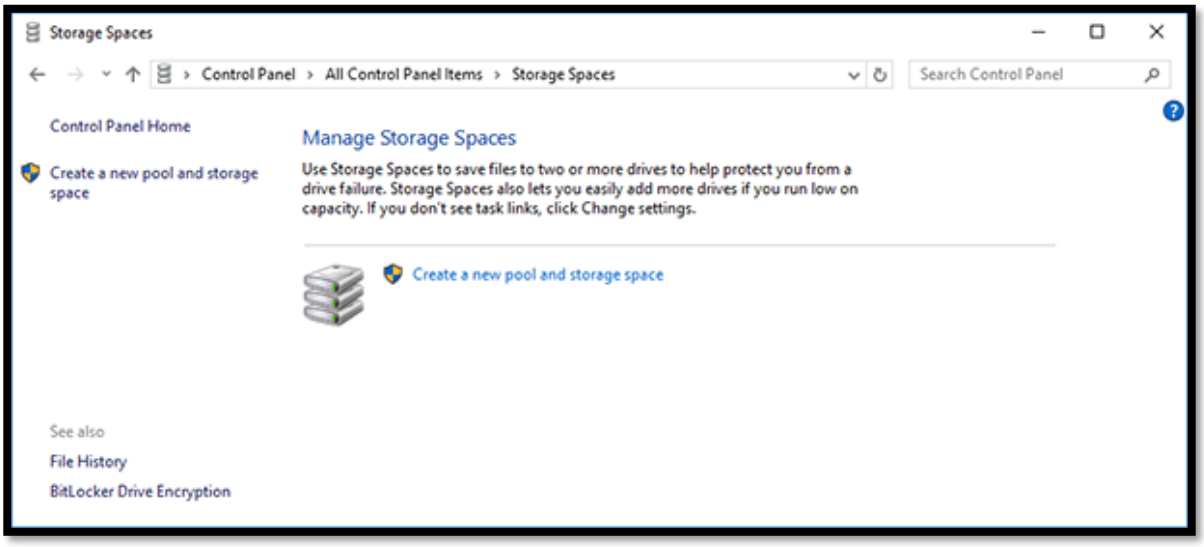
❖ التحكم في مساحات التخزين

يمكنك إنشاء مساحة تخزين من خلال القائمة الخاصة في لوحة التحكم.



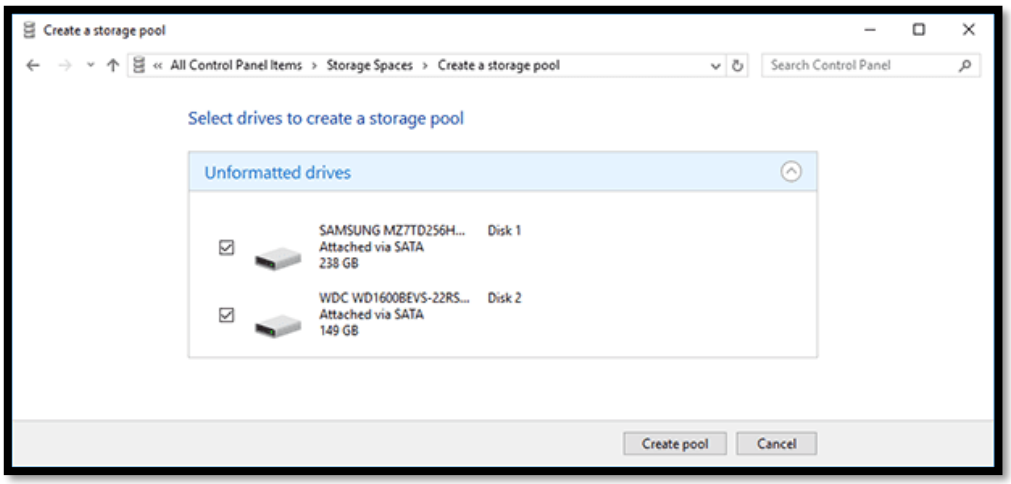
بتوصيل جميع الأقراص الصلبة بجهاز الكمبيوتر الخاص بك والتي ستستخدمها لإنشاء مساحة تخزين.

بعد ذلك انتقل إلى لوحة التحكم، وحدد القائمة مساحات التخزين /إنشاء مجموعة جديدة ومساحة تخزين.



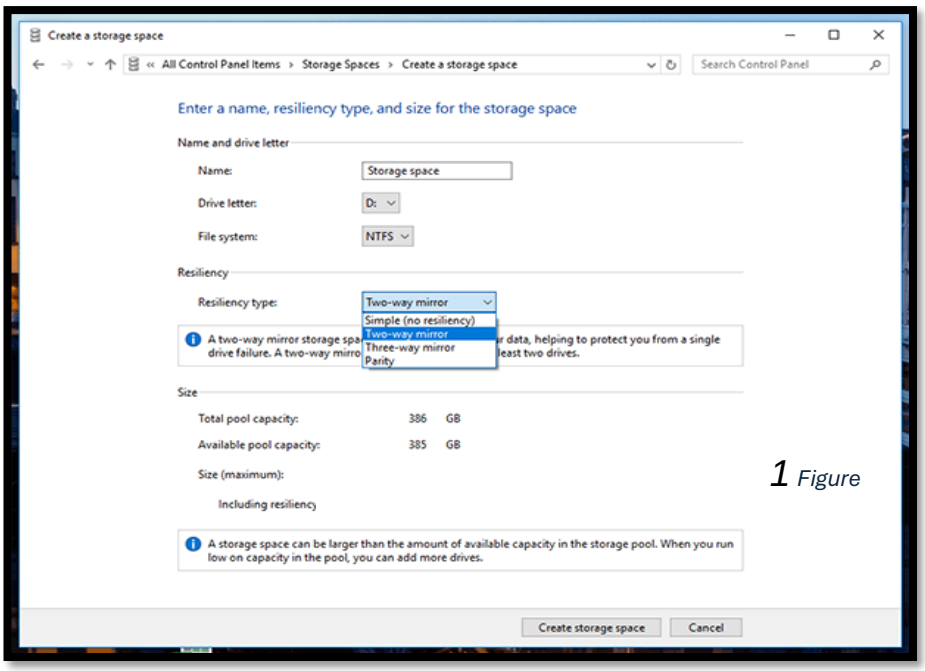
حدد الأقراص التي تريد إضافتها إلى المجموعة ثم إنقر على "إنشاء مجموعة"

يرجى تذكر أنه سيتم حذف جميع البيانات من محركات الأقراص المحددة لمجموعة التخزين.



بعد إنشـــاء المجموعة، يجب عليك ضـــبط مســـاحة التخزين الجديدة – تعيين إســـم وتحديد حرف محرك أقراص. سيتم تمثيل هذا الحرف والإسم في نظام التشغيل Windows

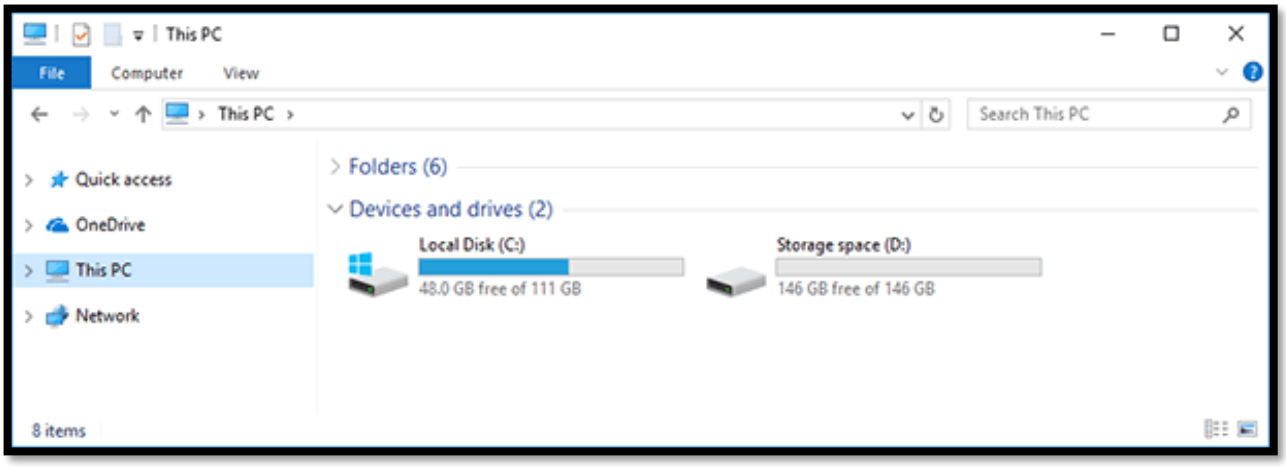
يمكنك أيضـاً تحديد نظام ملفات Windows الأسـاسـي NTFS – ، أو نوع النظام الجديد ReFS إذا كنت ســتنشــئ مســاحة تخزين متطابقة أو متكافئة تهدف إلى حمايتك من فقدان البيانات، فمن الأفضــل اختيار نوع نظام الملفات المرن ReFS .



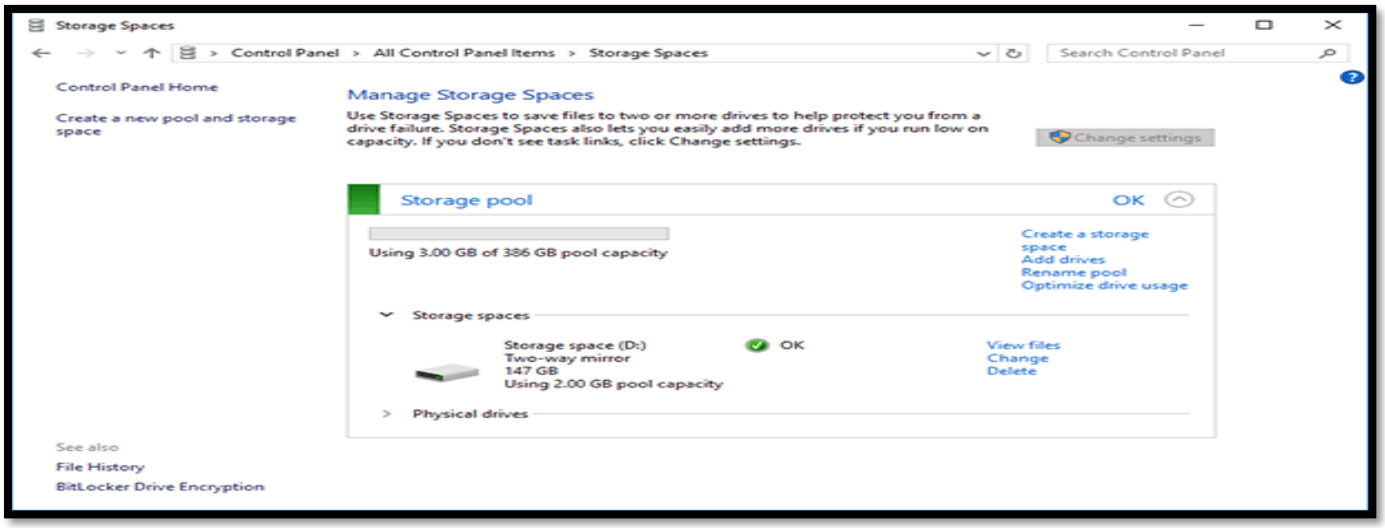
1 Figure

كيفية إستخدام مساحات التخزين

سيتم عرض مساحة التخزين التي قمت بإنشائها كمحرك أقراص آخر في المجلد هذا الكمبيوتر، سيحمل محرك الأقراص هذا الإسم والحرف الذي قمت بتعيينه له، وسيبدو مشابهاً لمحركات الأقراص الأخرى .

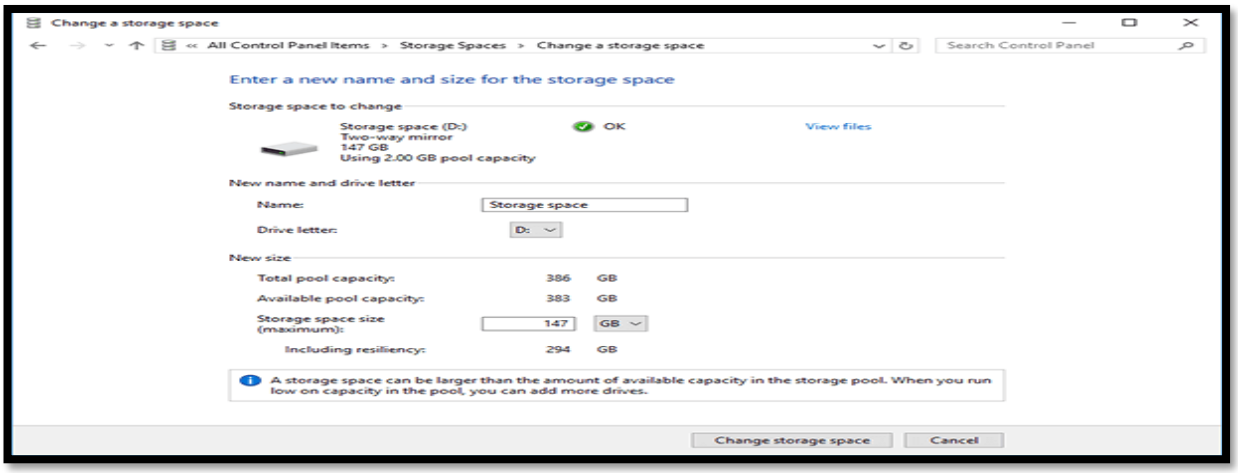


بعد إنشاء وحدة تخزين على القرص، يمكنك الإنتقال إلى لوحة التحكم لإدارتها أو تغيير الإعدادات.

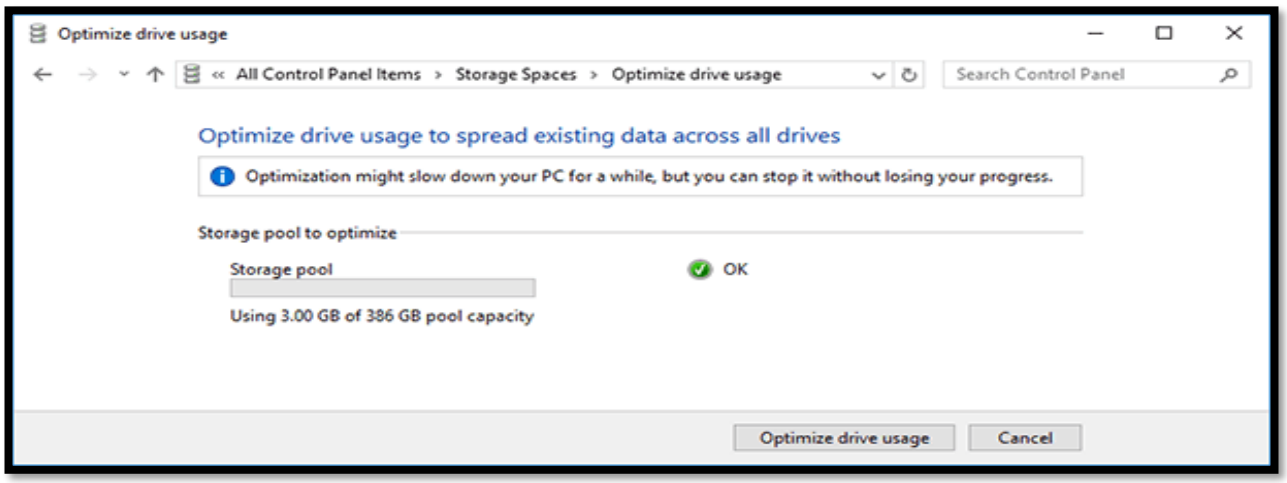


يمكنك إنشاء مساحة تخزين أخرى، عددهم محدود فقط بعدد الأقراص المادية المتصلة بالكمبيوتر، يمكنك إضافة محركات أو إعادة تسمية مجموعة التخزين.

يمكنك تغيير اسم أو حرف مساحة التخزين من خلال النقر على "تغيير الإعدادات" في قسم مساحات التخزين.

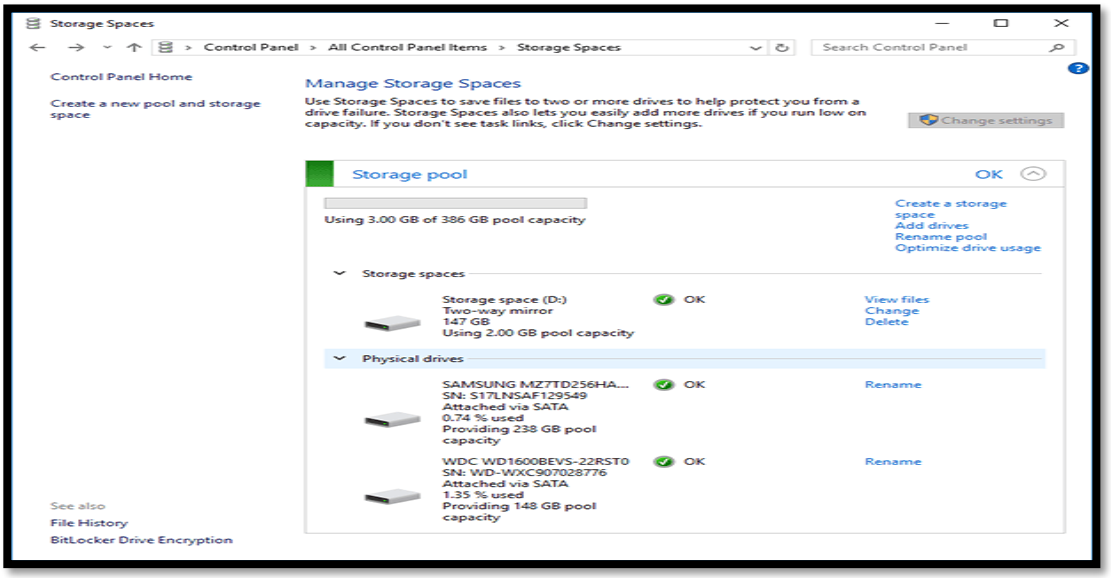


لإضافة أقراص إلى مساحة التخزين الحالية، إختر "إضافة محركات أقراص" وقم بتحديد الأقراص التي تريد إضافتها، عند إختيار القائمة "تحسين إستخدام محرك الأقراص"، سيقوم نظام التشغيل Windows بنشر البيانات الموجودة بالتساوي عبر الأقراص داخل مساحة التخزين.

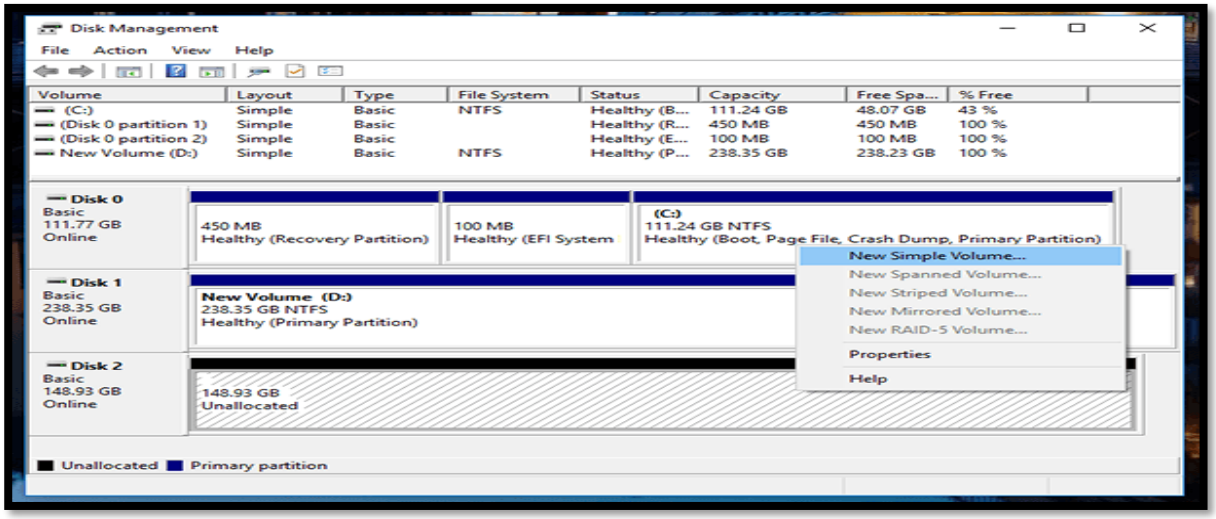


إذا كانت مساحة التخزين الحالية تتكون من ثلاثة أقراص فعلية أو أكثر، فيمكن إزالة أحدها، للقيام بذلك، إفتح قائمة لأقراص المادية ثم إنقر على رابط "التحضير للإزالة" بجوار القرص الذي تريد إزالته، في حالتنا، لايوجد إرتباط لأن مساحة التخزين تتكون من قرصين فقط.

بعد ذلك، سيأخذ النظام البيانات من القرص الذي يتم حذفه إلى القرصين الفعليين المتبقيين (أو أكثر) من مساحة التخزين، نتيجة لذلك، سيتغير الرابط "التحضير للإزالة" إلى "إزالة".

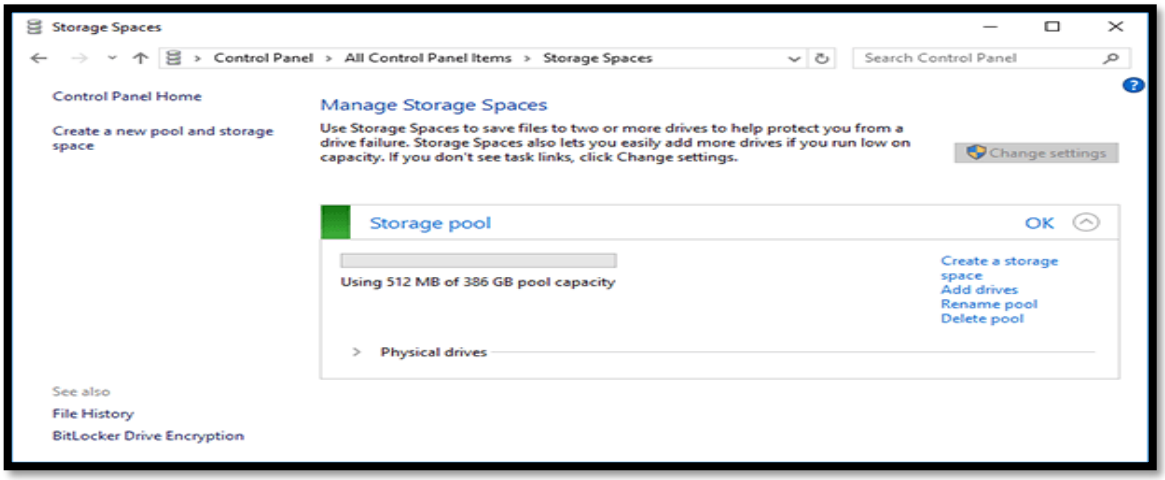


سيصبح القرص الفعلي المحذوف من مساحة التخزين متاحاً في نافذة إدارة الأقراص، لتمكين المزيد من العمل معها، قد تضطر إلى إنشاء قسم جديد هناك وتهيئته.



من بين خيارات إدارة مساحة التخزين، هناك خيار لحذف مساحة التخزين (يوجد على يمين إسم مساحة التخزين في قسم مساحات التخزين (يرجى تذكر أنه سيتم مسح جميع البيانات من مجموعة التخزين المحذوفة).

إذا كنت تريد حذف مساحة التخزين، فستصبح قائمة "حذف المجموعة "نشطة؛ عند إختيارها، يمكنك حذف مجموعة التخزين بالكامل.



تمرين :

قلص القرص الصلب باستخدام Server 2012 Disk Management متبعاً الخطوات التي تمت دراستها ؟

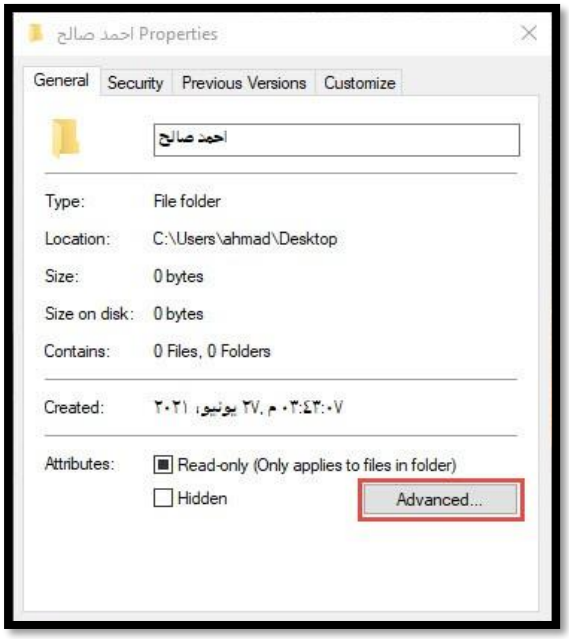
الفصل الثامن: خدمات الطباعة وتأمين الملفات

التشفير:

١- ابحث عن الملف او المجلد الذي تريد حمايتها

٢- انقر بزر الماوس الأيمن على الملف أو المجلد ثم انقر على Properties أسفل قائمة السياق.

٣- انقر على زر ...Advanced في قسم Attributes في النافذة.

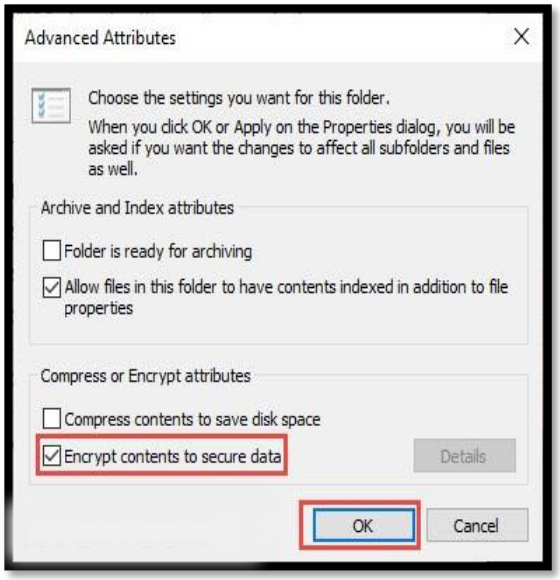


٤- حدد مربع الاختيار Encrypt contents to secure data

٥- انقر على الزر Ok للعودة إلى نافذة الخصائص الرئيسية.

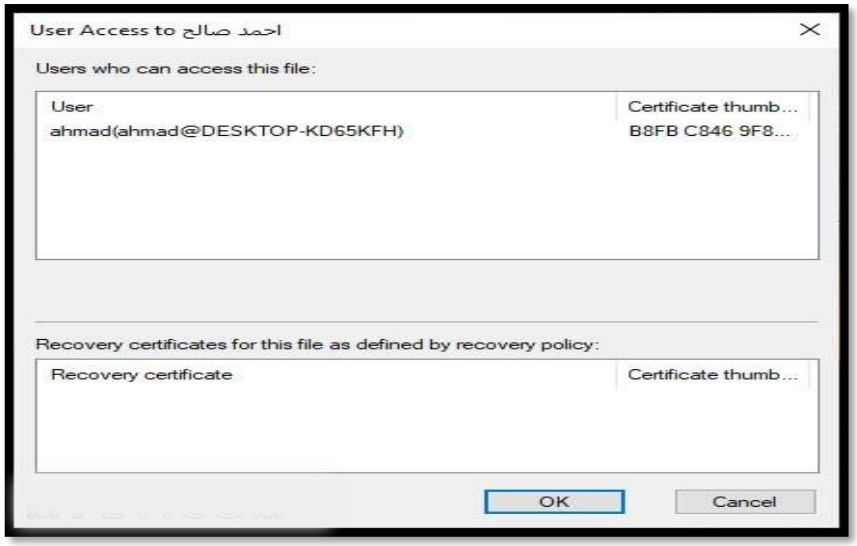
٦- الآن اضغط على Apply وسيرسل لك ويندوز اشعارًا لبدأ حماية الملفات انقر عليه.

٧- انقر على Back up now لبدأ النسخ الاحتياطي، وتابع الخطوات التي ستظهر على الشاشة



بعد الإنتهاء من التكوين سـيكون لديك ملف احتياطي صـغير، سـتحتاج إلى هذه المعلومات إذا فقدت الوصول إلى ملفاتك المحمية.

بعد إجراء النسـخ الاحتياطي، سـيتم الآن حماية الملفات،يتم تشـفير هذه الملفات بمفتاح مرتبط بحسـاب مسـتخدم ويندوز الخاص بك، يُمكنك التحقق منه بالنقر على Details بجوار زر الاختيار Encrypt contents to secure data.



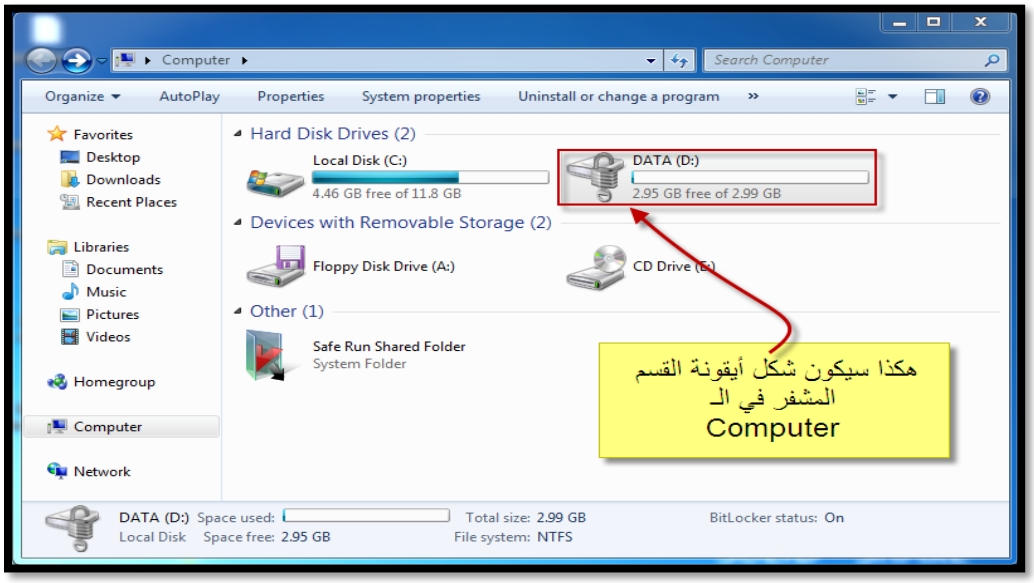
إذا حاول أي شخص آخر الوصول إلى الملف – سواء من حساب مستخدم آخر، أو عن طريق إزالة محرك الأقراص الثابتة فعلياً – فستظهر المحتويات على أنها نص مشوه لا معنى له.

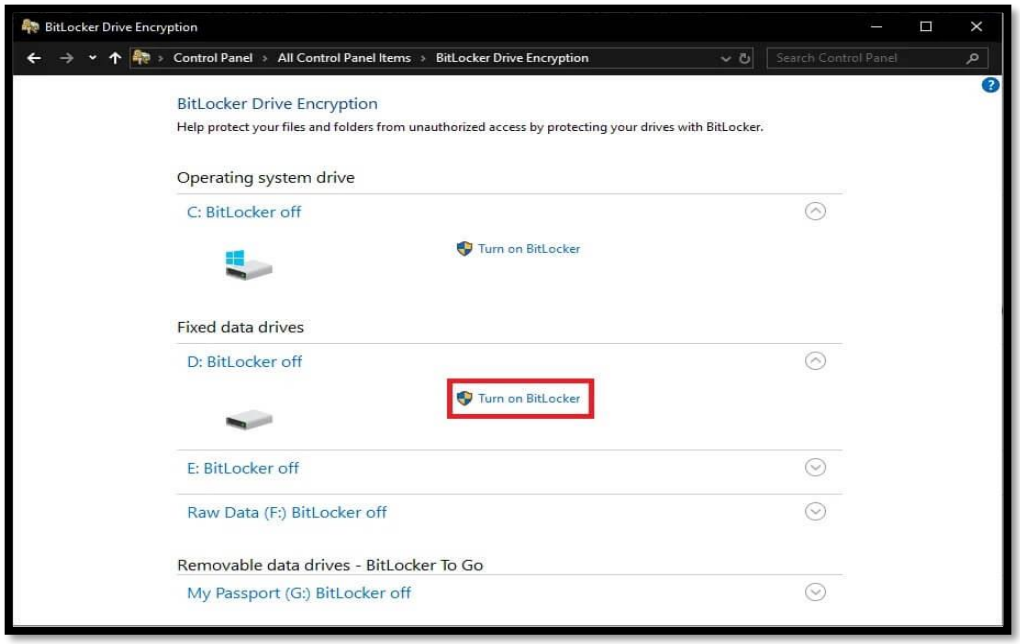
يمكنك فك حماية الملفات في أي وقت بالعودة إلى نافذة Properties وفتح Advanced مرة أخرى.

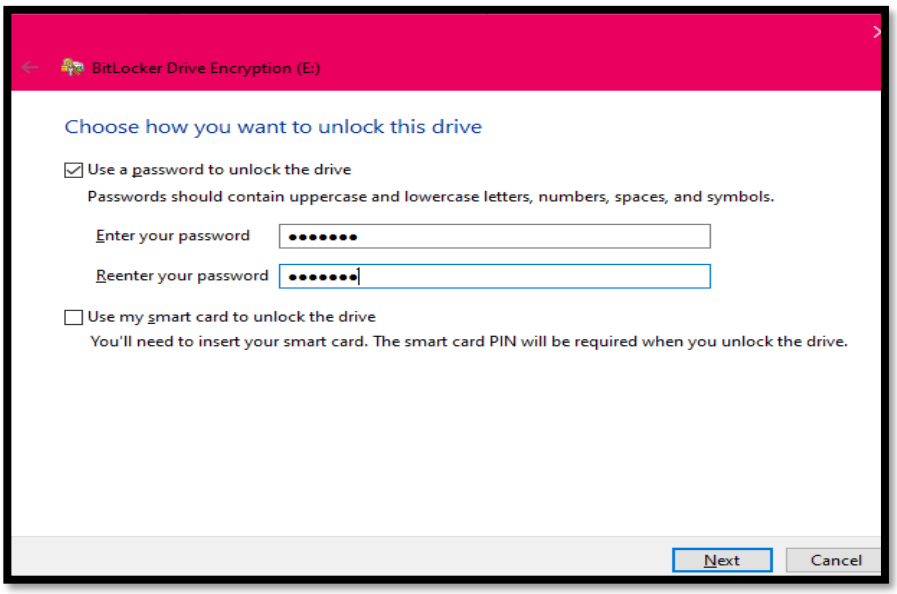
فقط قم بإلغاء تحديد مربع الاختيار Encrypt contents to secure data وانقر فوق Ok لإغلاق النوافذ.

خطوات التشفير باستخدام Bitlocker

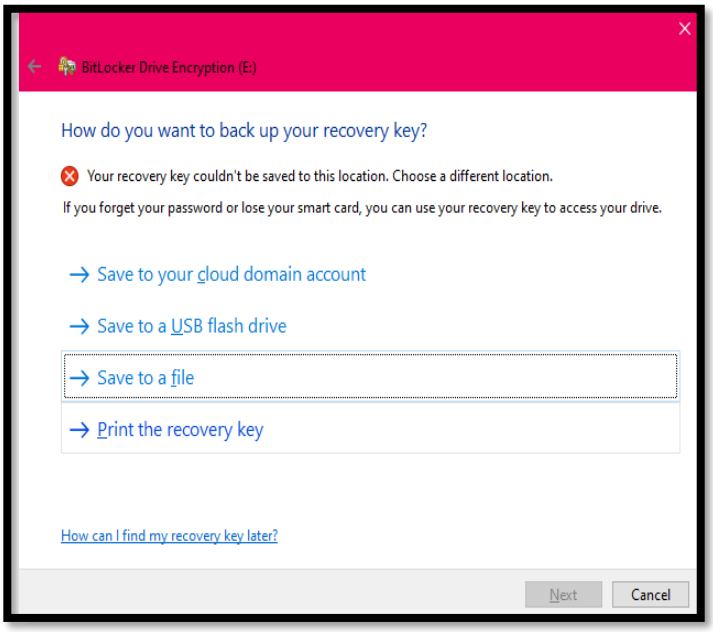
١. اذهب الى control panel
٢. يتم اختيار Bitlocker Drive Encryption
٣. يتم اختيار البارتيشن المطلوب تشفيره ثم الضغط على Turn on Bitlocker
٤. يتم إدخال باسورد لفك التشفير

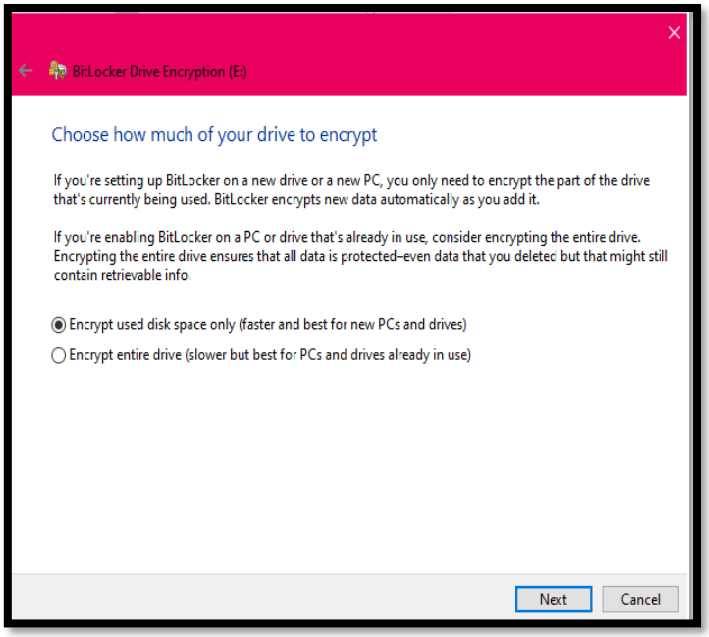




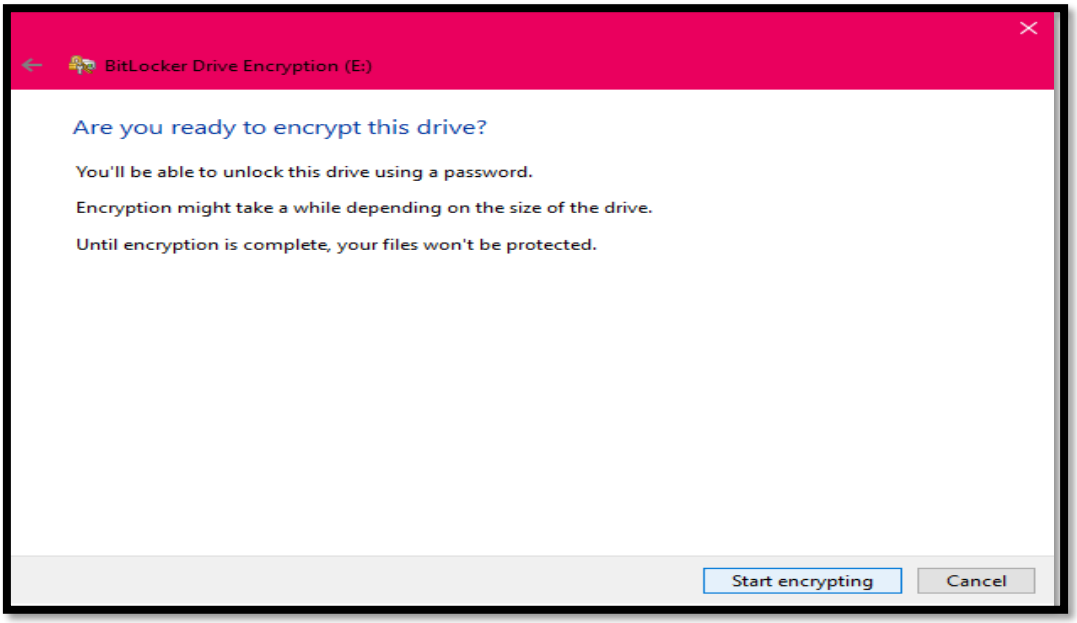


يتم طلب اختيار حفظ مفتاح التشفير لاستخدامه عند فك التشفير كما في الشكل التالي





بمجرد الضغط على Start Encrypting تبدأ عملية التشفير

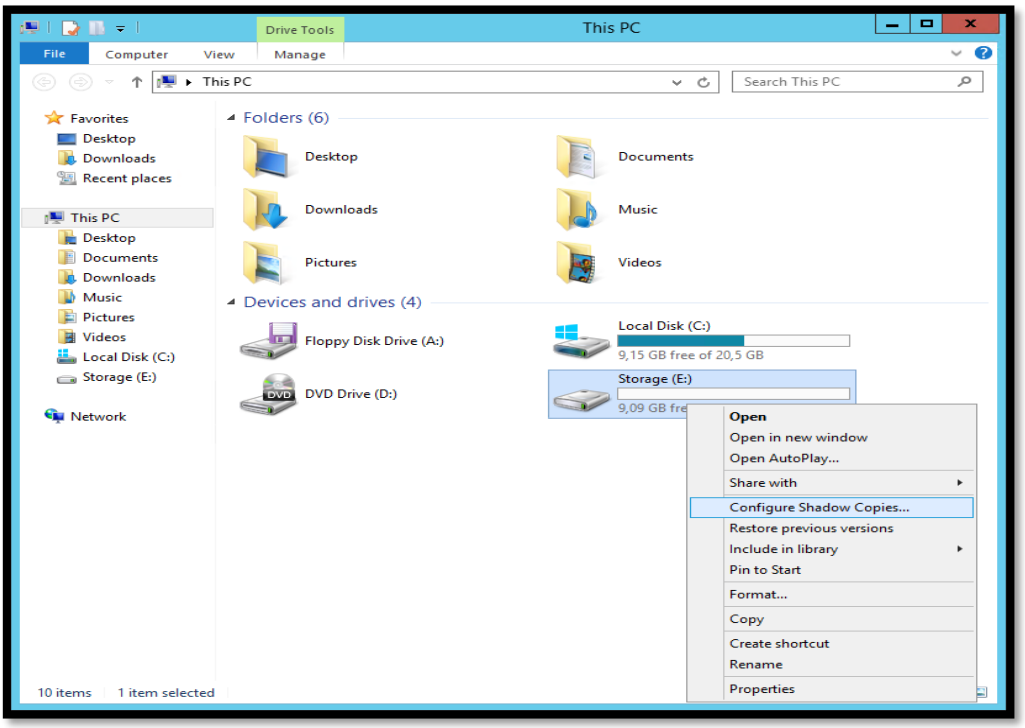


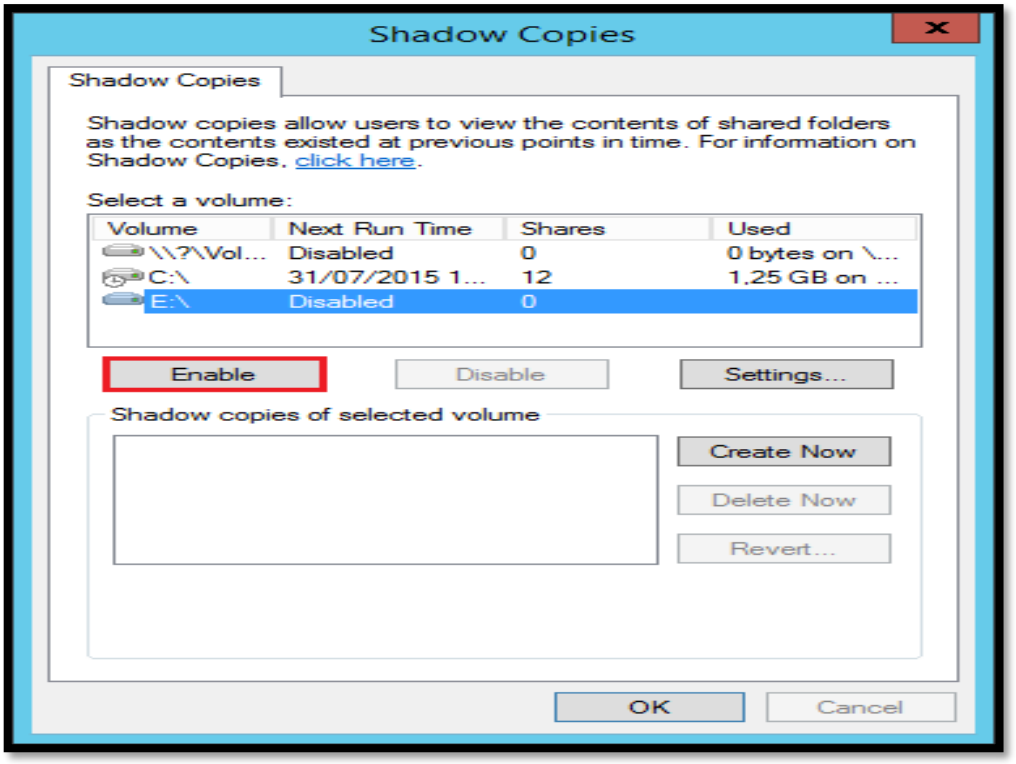
❖ حماية الملفات والمجلدات المشتركة بواسطة Shadow Copies

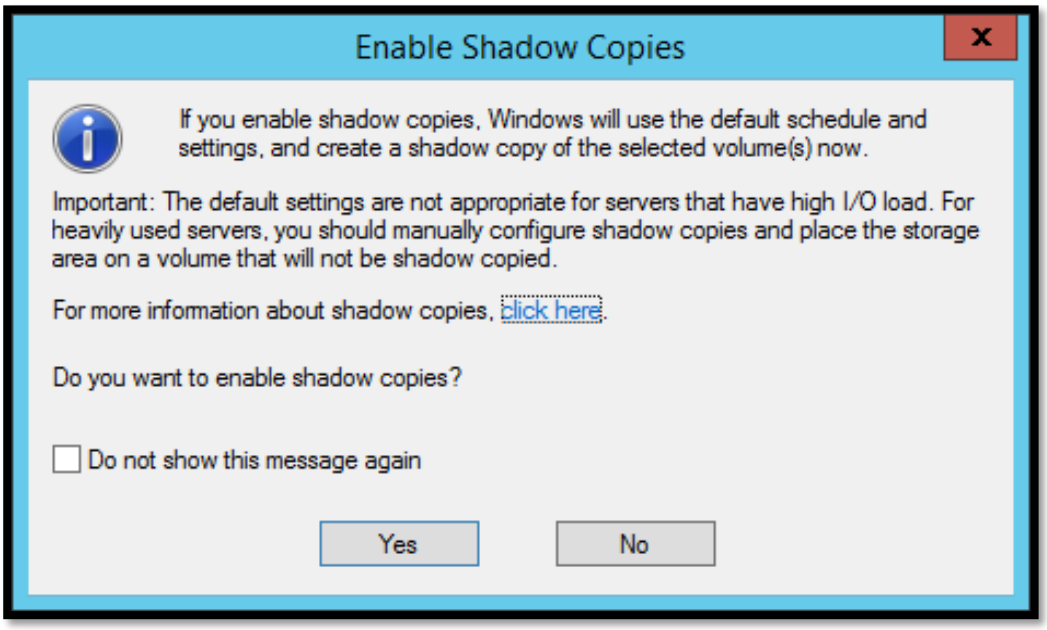
تقنية النسخة المطابقة shadow copy هي نوع من النسخ الاحتياطي لأخذ لقطات قابلة للاستعادة من وحدة تخزين.

ملحوظة : shadow copies تقنية تستخدم للسماح باستعادة الحالات السابقة لوحدة التخزين بالكامل ، لا يمكنك استعادة الحالات السابقة للملفات و / أو المجلدات الفردية.

تتم تفعيلها كما بالصور :







سيتم إنشاء لقطة أولى، بالإعدادات الافتراضية على النحو التالي:

سيتم تخزين نسخ الظل لوحدة التخزين في نفس المجلد

سوف تأخذ نسخ الظل لوحدة التخزين أقصى قدر من ١٠٪ من مساحة القرص المحلي

يحتفظ النظام بحد أدنى ٣٠٠ ميغا بايت من مساحة القرص للنسخ الاحتياطية

يقوم النظام بجدولة نسختين ظاهريتين يومياً (٧,٠٠ ص و ١٢,٠٠ م)

❖ تكوين خدمة مجلدات العمل

عند عمل مجموعة من المجلدات، يجب أن نراعي كيفية وصول المستخدمين لهذه المجلدات، ومن لديه الصلاحيات للدخول لهذه المجلدات.

فعند إنشاء مجلدات للمستخدمين على الـ File Server، يجب أن تكون Shared Folders، بمعنى أنه يمكن للمستخدمين الوصول لهذه المجلدات عن طريق الشبكة.

كما في الشكل التالي ، كيفية عمل Sharing Folder، بعد إنشاء مجلد جديد.

نضغط بالزر الأيمن، ثم نختار خصائص المجلد.

. Sharing

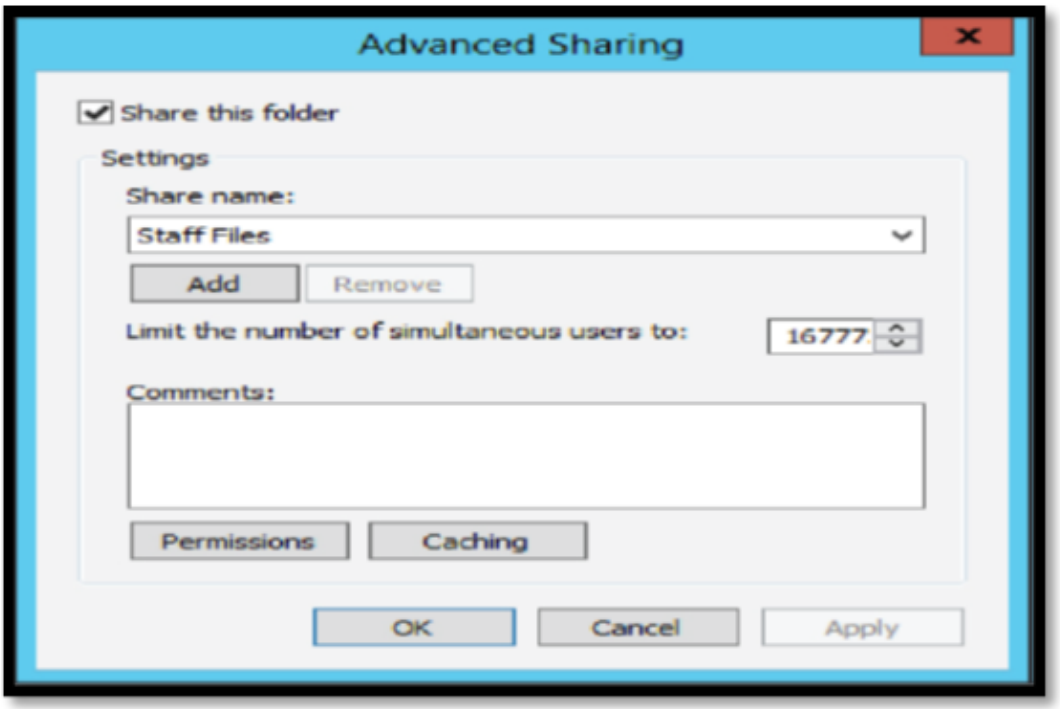
. Advanced Sharing

. Share this folder

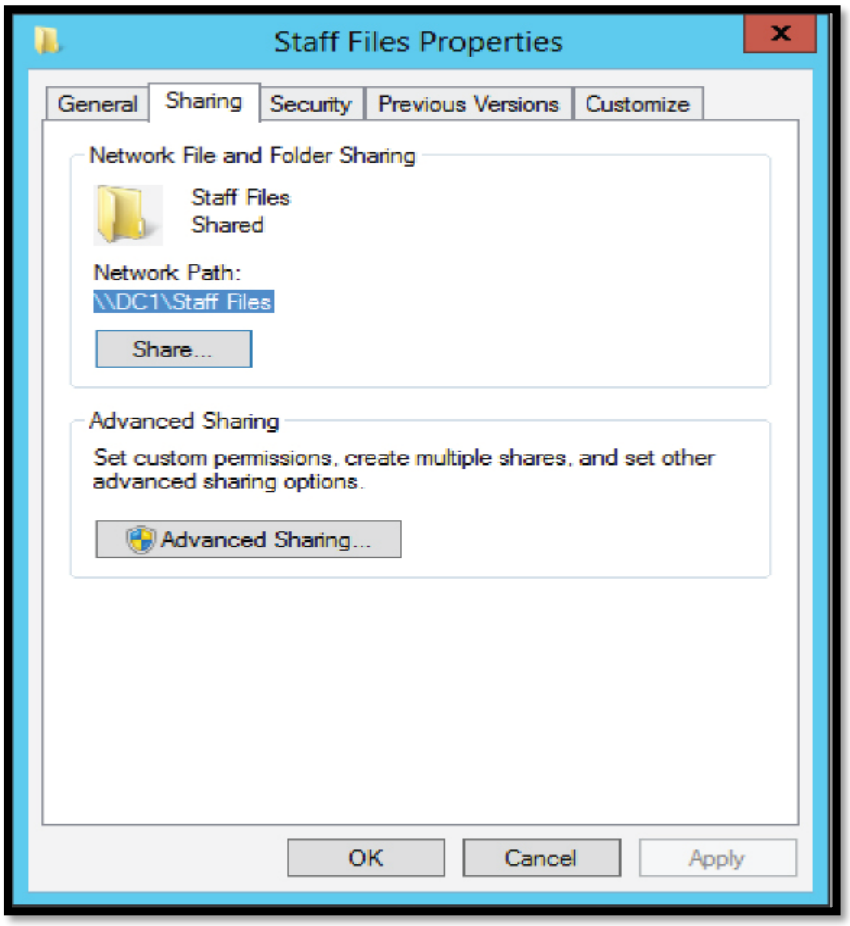
. Permissions

بعد ذلك نقوم بتحديد الأشخاص لإعطائهم الصلاحيات للوصول لهذا المجلد

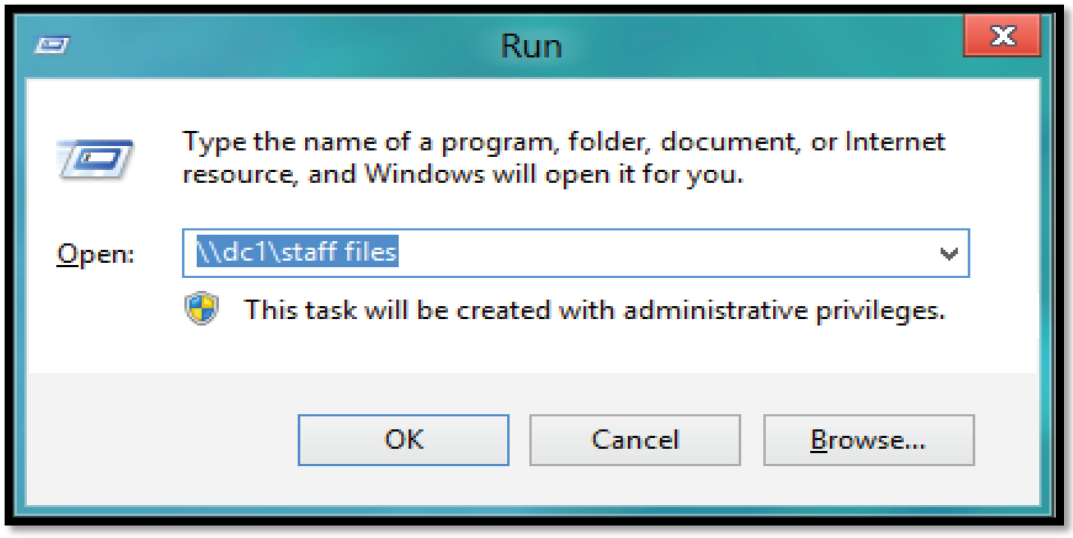
كيفية الوصول للـShared Folder



يمكن للمستخدمين الوصول للمجلدات التي تمت مشاركتها في الشبكة بعدة طرق، وسوف نتعرف على هذه الطرق بشكل مفصل، ومن هذه الطرق، تحديد مكان المجلد، كما نشاهد في الشكل التالي .

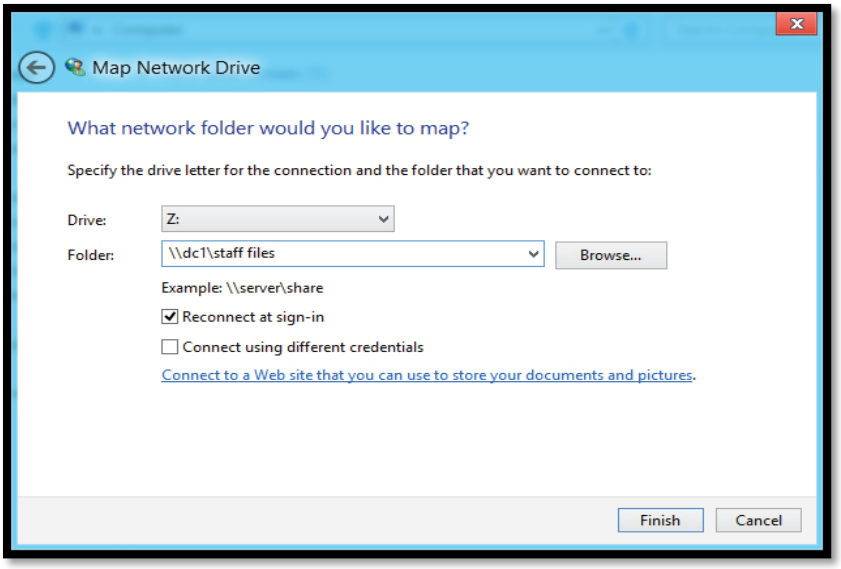


بعد معرفة مكان المجلد الذي تمت مشاركته مع المستخدمين، سوف نقوم بتجربة الدخول للمجلد عن طريق أحد الأجهزة نقوم بتسجيل الدخول ونذهب للـ Run ونكتب مسار المجلد

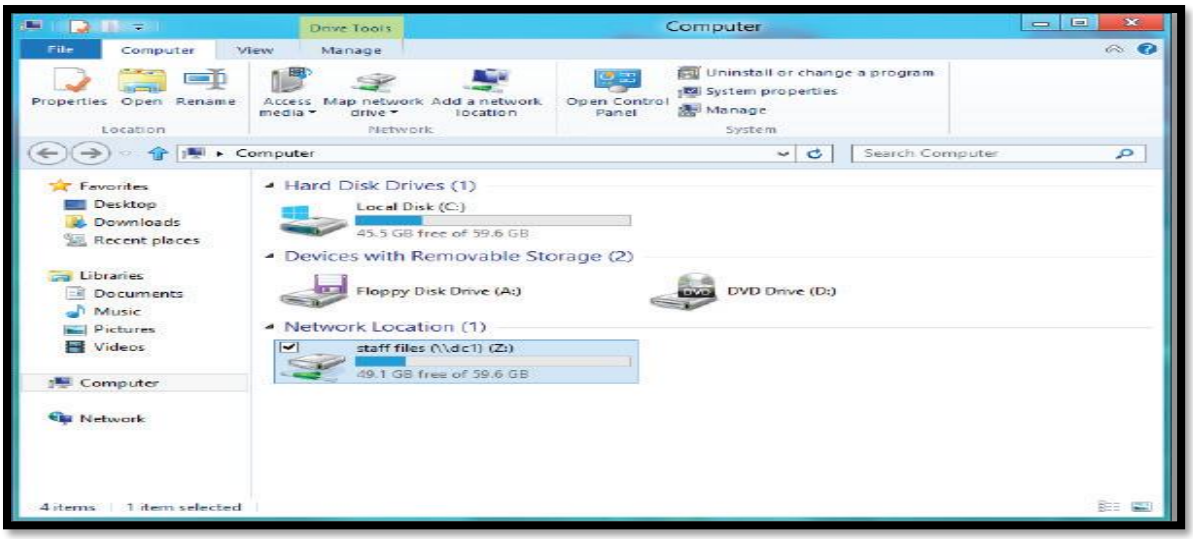


يمكننا استخدام Map network Drive عن طريق الخطوات التالية:

- نقوم بتسجيل الدخول لأحد أجهزة المستخدمين.
- My Computer .
- Computer .
- Map network drive .
- ونحدد مكان الملجد الذي تمت مشاركته



بعد ذلك سوف يستطيع المستخدم الدخول للمجلد الذي تمت مشاركته



سوف نتعرف في المثال التالي على كيفية تطبيق الـ Quota ، وكيفية تعيين المساحات التخزينة المختلفة لكل مستخدم.

نضغط بالزر الأيمن على القرص الصلب C drive أو غيره من الـPartitions.

. Quota-

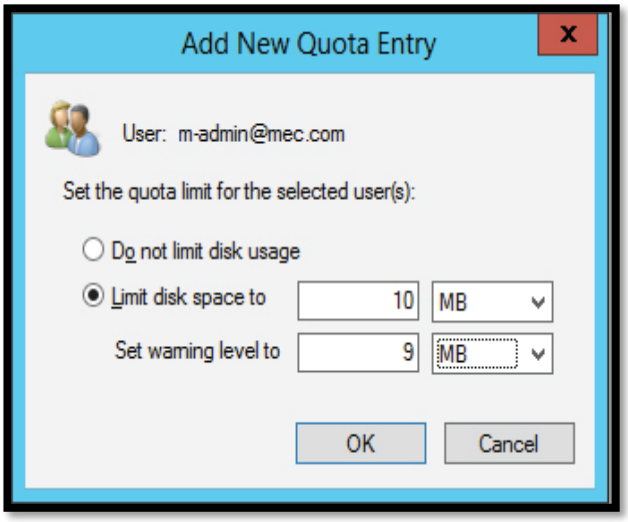
. Enable quota management-

. Deny disk space to users exceeding quota limit-

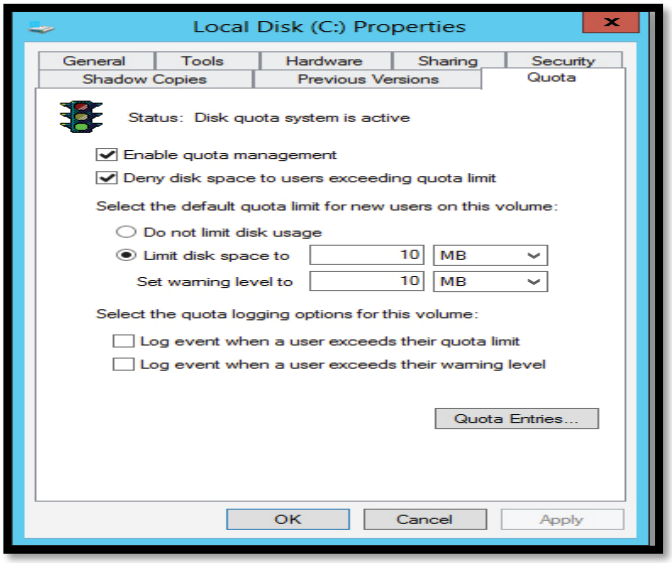
Quota Entries-، ويستخدم هذا الخيار لتحديد المساحات التخزينية لكل مستخدم.

نختار-New Quota Entry

نحدد إسم المستخدم، ونختار المساحة التخزينية الخاصة به،-



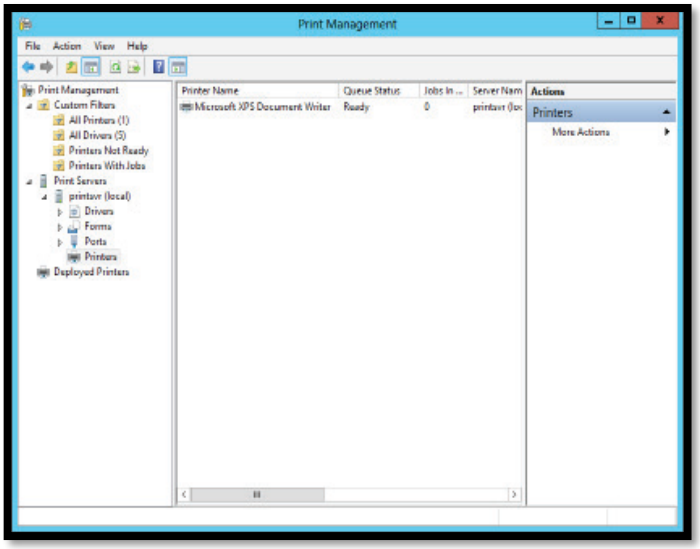
ويجب أن نلاحظ بأن باقي المستخدمين سوف يستخدمون المساحة التخزينية بشكل كامل، ولذلك يجب أن نحدد مساحة تخزينية لباقي المستخدمين



❖ تكوين الطباعة الشبكية

لتثبيت Print Server إتبع الخطوات التالية :

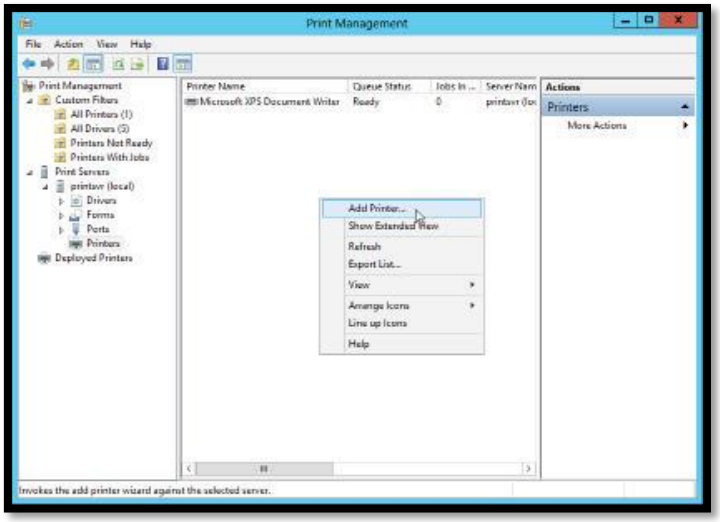
- نقوم بتثبيت Windows Server 2012 وإضافته للنطاق، وليكن إسمه"printsvr". "
- نذهب لـServer Manager.
- Add Roles and Features .
- نضيف Print and Document Services.
- نقوم بإنهاء الإعدادات بالضغط على Next.
- يمكننا بعد ذلك التحكم بطابعات الشبكة كما في الشكل التالي:



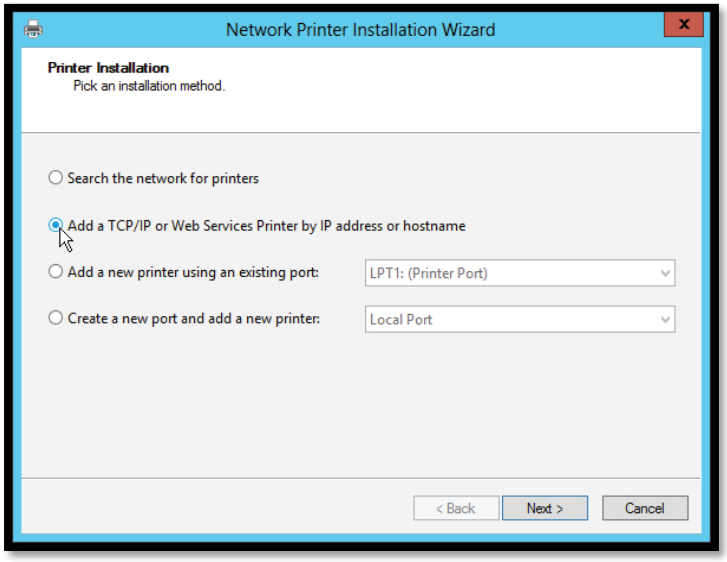
إنشاء Network Printer

لإنشاء طابعة جديدة ومشاركتها عبر الشبكة، اتبع الخطوات التالية:

- نذهب لـPrint Management.
- Print Servers, printsvr .
- Printers .
- نضغط بالزر الأيمن ونختار Add Print



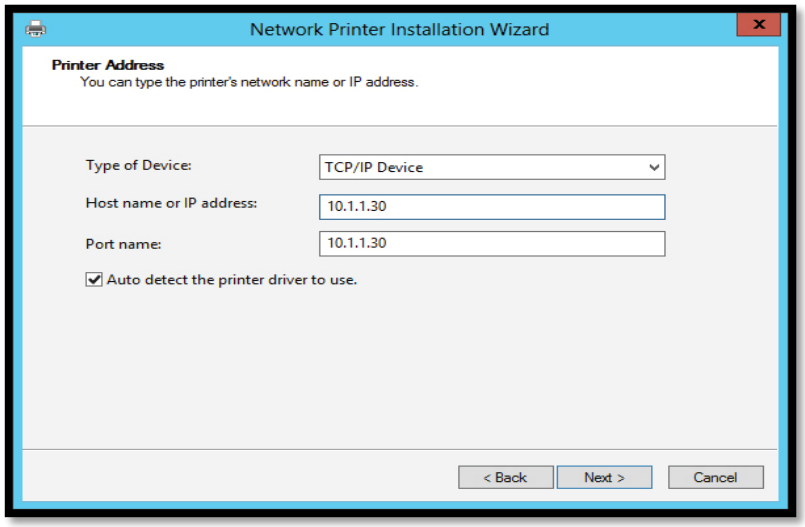
بعد ذلك سوف نقوم بإعداد الطابعة على النحو التالي

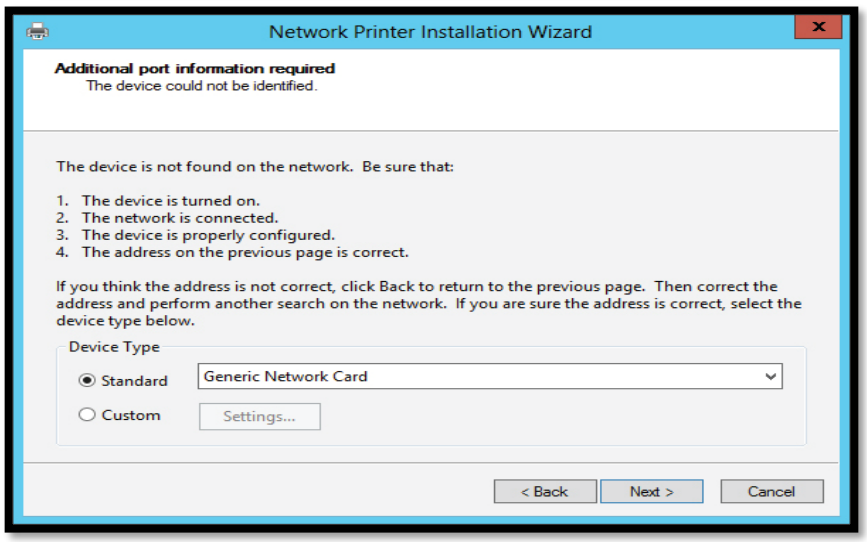


نحدد عنوان الشبكة الخاص بالطابعة، ويجب أن نقوم بذلك على الطابعة أولا.

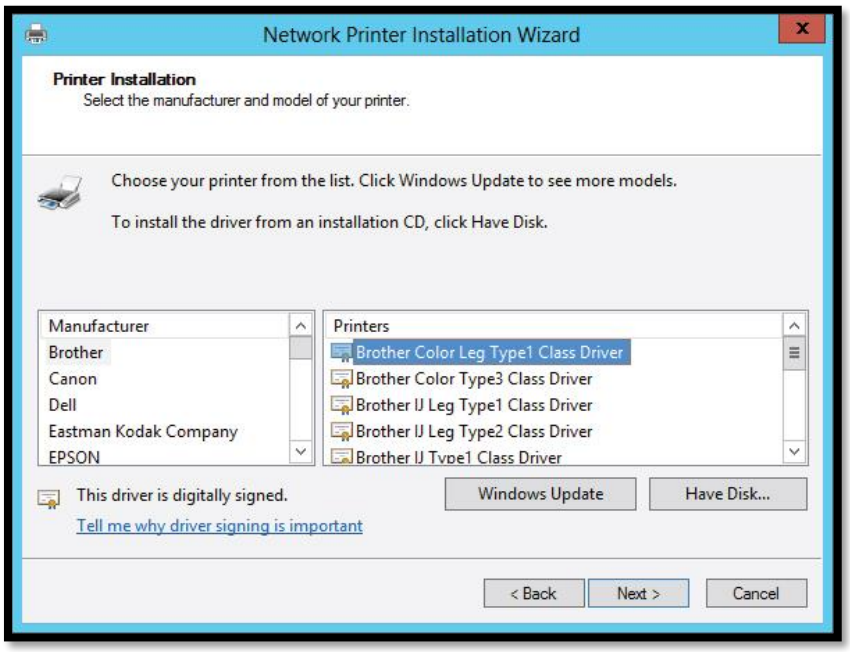
سوف يقوم المعالج بالبحث عن منفذ عنوان الشبكة، وقد يستغرق هذا الأمر اقل من دقيقة.

نحدد الخيارات الإفتراضية كما في الشكل ثم نضغط Next

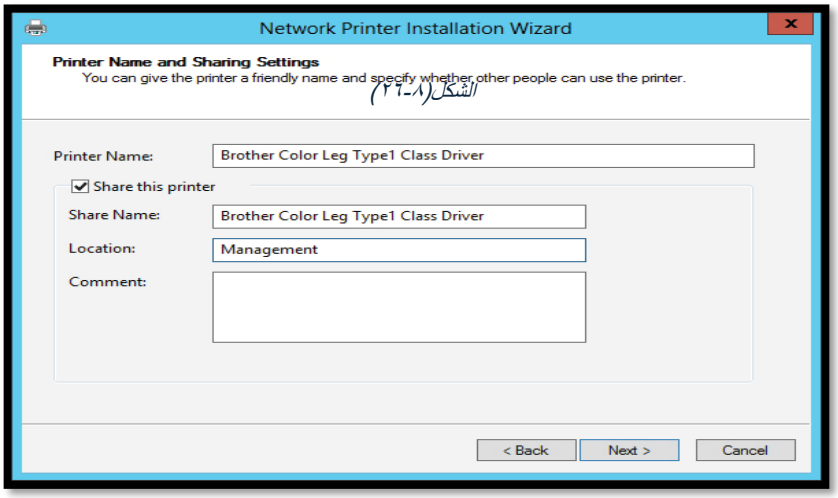




إختيار برنامج الطابعة، ودائما ما يكون مرفقا معها، أو نقوم بتحميله من موقع الشركة المصنعة للطابعة ولكن لأننا نعمل في النظام الإفتراضـــي، فســوف نقوم بتثبيت طابعات إفتراضــية أيضــاً وتثبيت البرامج الخاصة بها، التي يتيحها Windows Server 2012كما نلاحظ في الشكل



بعد ذلك نحدد إسم الطابعات



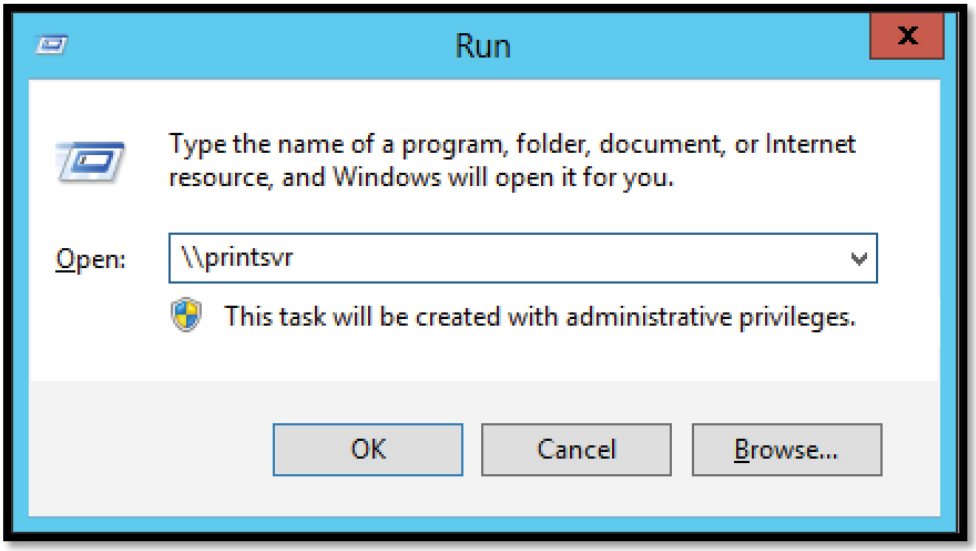
الشكل(٨-٢٦)

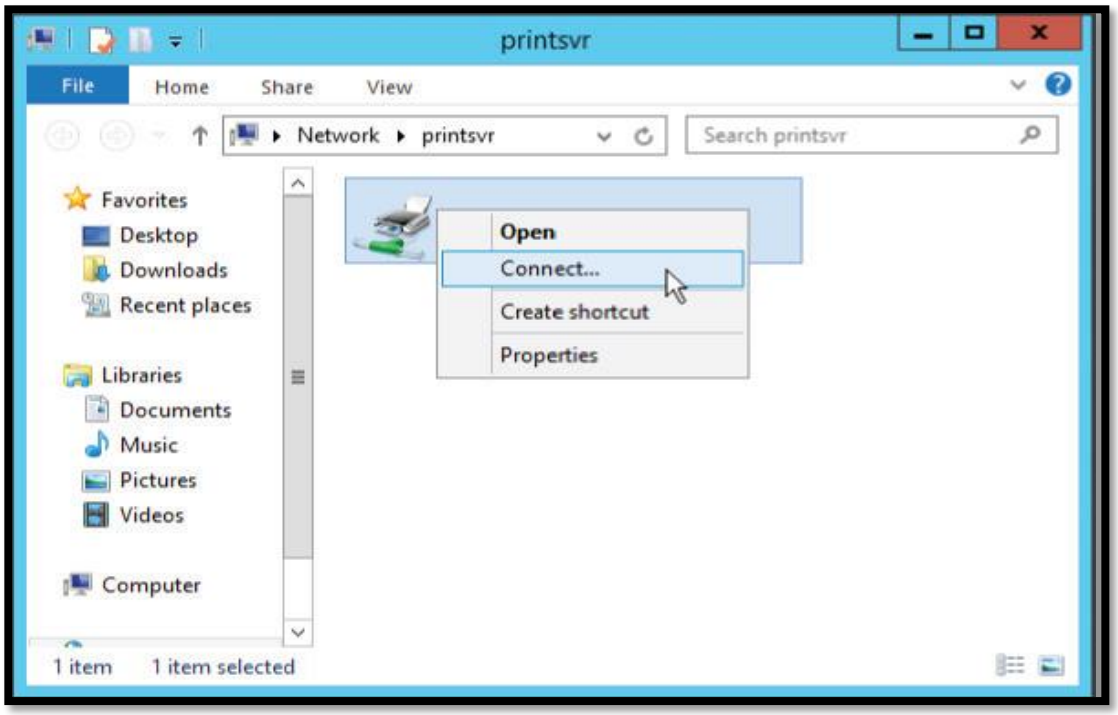
بعد القيام بتثبيت الطابعة ومشاركتها عبر الشبكة، يستطيع المستخدم الوصول إليها، وإضافتها لجهازه، عن طريق الـRun". "

-ثم نقوم بكتابة \\printsvr". "

-سوف نلاحظ وجود الطابعة التي قمنا بإضافتها ومشاركتها عبر الشبكة.

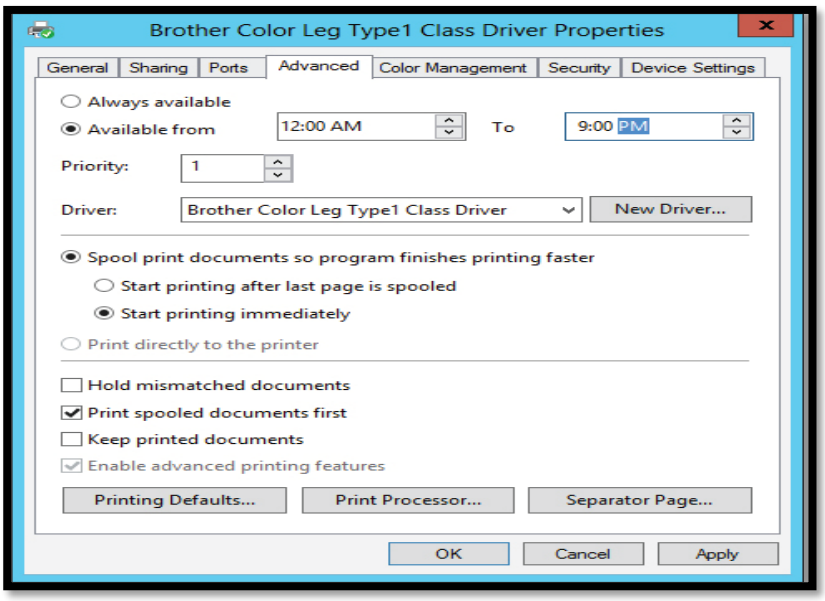
-نضغط بالزر الأيمن على الطابعة ونختار Connect .





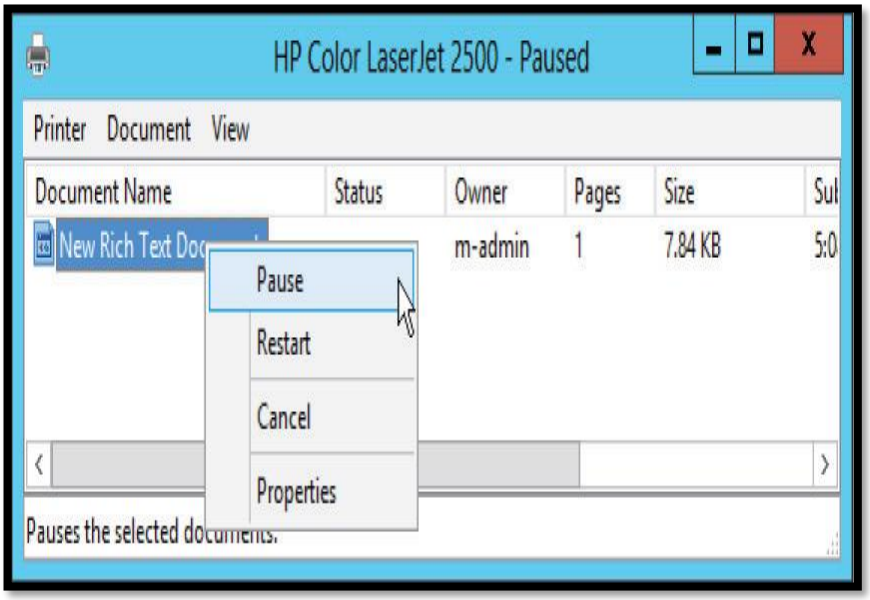
صلاحيات الطباعة

كما يمكن لمدير الشبكة أيضاً تحديد أوقات توفر الطابعة والطباعة عليها، ففي بعض المؤسسات يخصص وقت معين للطباعة، ولا يمكن للمستخدمين الطباعة إلا في الأوقات المحددة



الإيقاف المؤقت وإعادة أوامر الطباعة

يمكن لمدير الشبكة إيقاف الأوامر التي تم إرسالها للطابعة وذلك عن طريق تحديد الطابعة بالزر الأيمن ثم إختيار Open Printer Queue ، كما نلاحظ في الشكل التالي ، ثم Pause ، ويمكن إعادة أوامر الطباعة بالضغط على Restart .



إلغاء الملفات المطبوعة

كما يمكن أيضاً لمدير الشبكة إلغاء أوامر الطباعة التي تم إرسالها للطابعة كما نلاحظ في الشكل السابق، وذلك عن طريق اختيار Cancel.

تمرين :

تشفير البيانات على القرص الصلب لمنع الوصول غير المصرح به متبعاً الخطوات التي تمت دراستها ؟

الفصل التاسع: نهج المجموعة

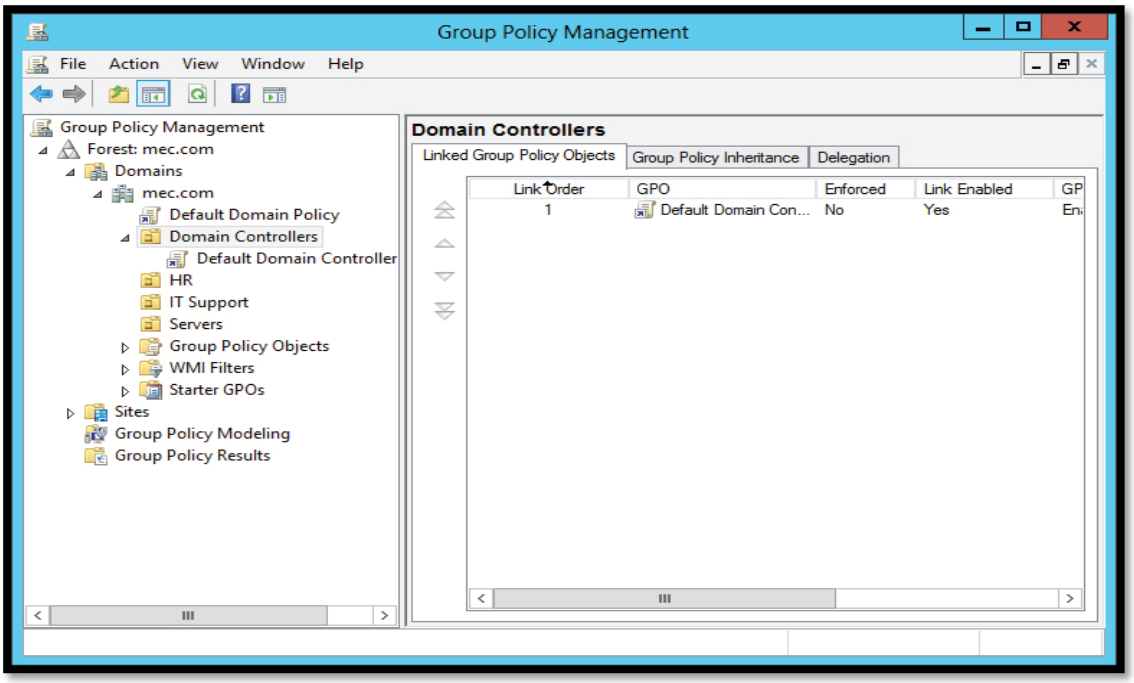
❖معالجة نهج المجموعة

لاحظ في الشكل التالي ستبأن جميع الـ OUs التي قمنا بإنشائها، تكون موجودة في الـ Group Policy الذي يتيح لنا إضافة صلاحيات محددة لكل OU.

كما نلاحظ هناك مجموعة من السياسات والقوانين التي تكون موجودة بشكل إفتراضي:

Default Domain Policy وهي القوانين الافتراضية الخاصة بكل الأجهزة والمستخدمين المضافين للنطاق.

Default Domain Controller Policy وهي القوانين الخاصة بالـDomain Controllers



إضافة Group Policy Object

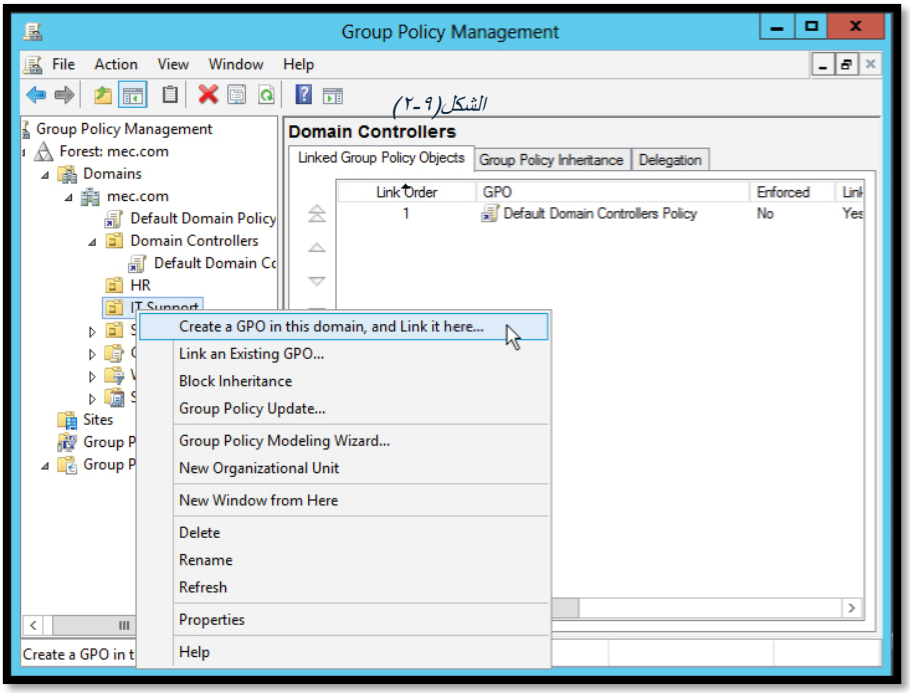
سوف نقوم بإضافة سياسات وقوانين لـ IT Support ،وكما أشرنا سابقاً، فإن هذه القوانين سوف تطبق فقط على المستخدمين الذين تمت إضافتهم للـ IT Support ، أما باقي المستخدمين فقد تكون لهم سياسات مختلفة تماماً، فكل OU لها قوانين معينه يحددها مدير الشبكة وطبيعة العمل في المؤسسة.

لإضافة GPO إتبع الخطوات التالية:

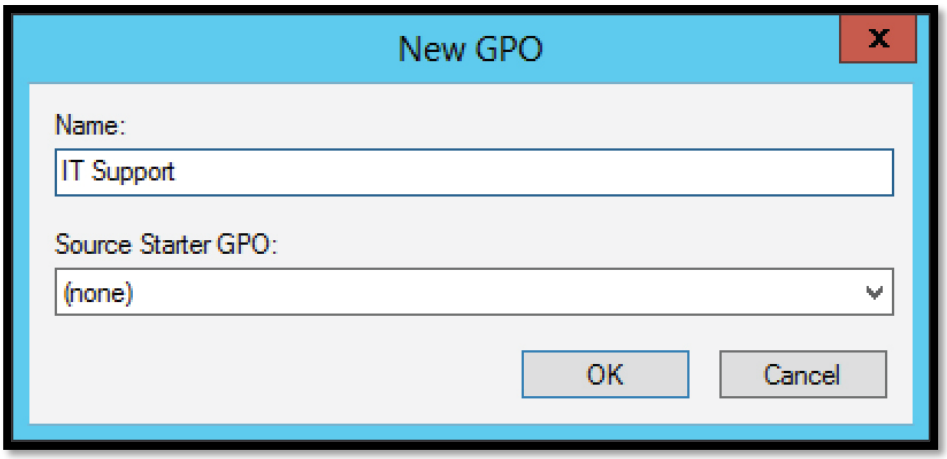
نذهب لقائمة البرامج ثم نضغط على Group Policy Management .

نحدد بالزر الأيمن على IT Support.

Great a GPO in this domain and link it here كما في الشكل التالي



الشكل(٩-٢)



نقوم بتسمية الـ GPO التي قمنا بإنشائها كما نرى في الشكل، بعد ذلك نضغط بالزر الأيمن ونختار Edit.

سوف نلاحظ وجود سياسات وقوانين خاصة بالأجهزة وأخرى للمستخدمين. ويجب أن نلاحظ أن هنالك فرق كبير بينهم، فالقوانين المتعلقة بالأجهزة هي تنطبق فقط على الأجهزة، وليس للمستخدم أي علاقة بها، وكذلك بالنسبة للقوانين المتعلقة بالمستخدم.

حذف Group Policy Object

إن حذف أي GPO قد يتسبب بحذف كل القوانين والسياسات المفروضة على المستخدمين، أو الأجهزة، لذلك يجب التأكد تماماً من أن الـ GPO لم يعد لها أي حاجة ويمكن الإستغناء عنها.

لحذف GPO إتبع الخطوت التالية:

نحدد ال GPO المراد حذفها بالزر الأيمن ثم نختار Delete ونأكد الحذف.

Group Policy Objects

وتحتوي على كل GPOs التي تم إنشـائها، ويمكن لمدير الشـبكة الرجوع إليها مرة أخرى، عند حذف أي GPO عن طريق الخطأ، كما يمكن لمدير الشبكة إنشاء GPO جديدة وإضافتها لأي OU تم إنشائها مسبقاً، وذلك عن طريق إختيار Link an Existing GPO.

Filters WMI هي عبارة عن مجموعة من الإعدادات لتصفية السياسات والقوانين المُطبَّقة سواءً كانت على الأجهزة أو المسـتخدمين، فعلى سـبيل المثال؛ لو أردنا تطبيق مجموعة من القوانين على الأجهزة، فبامكاننا تحديد أي نوع من الأنظمة نريد لتطبيق هذه القوانين، فيمكن تطبيق القوانين على الأجهزة التي تستخدم نظام Windows 7 دون سواها، وهكذا

يمكننا تطبيق WMI Filters عن طريق الخطوات التالية:

نضغط بالزر الأيمن على WMI Filters ، ونختار New.

نحدد إسماً للـFilters IWM.

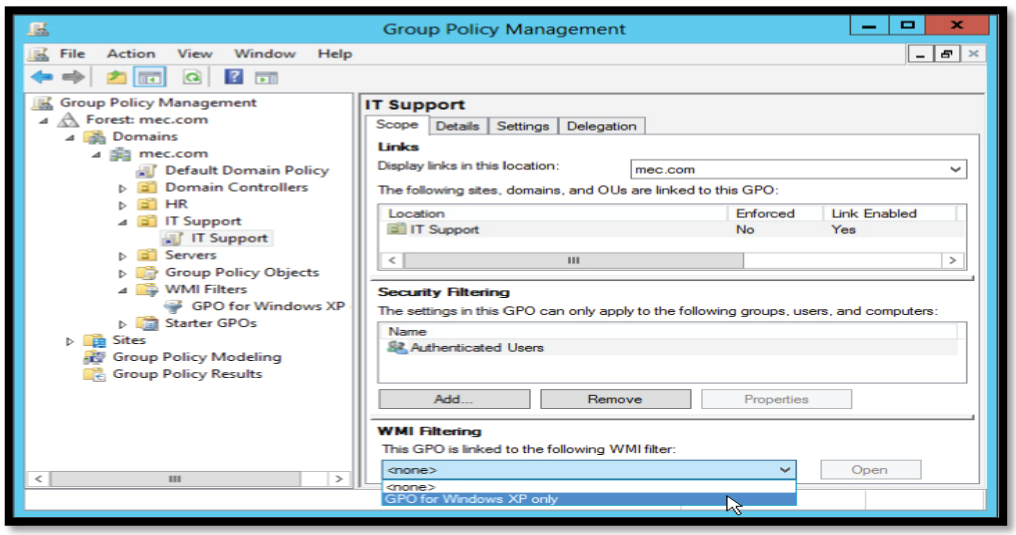
- Add .

ثم نظيف Query ،

ويمكننا الحصول على بعض الأمثلة لهذه الإعدادات من الرابط التالي:

http://technet.microsoft.com/en-us/library/cc779036(v=ws.10).aspx

بعد ذلك نحدد GPO التي تم إنشاؤها كما نلاحظ في الشكل التالي

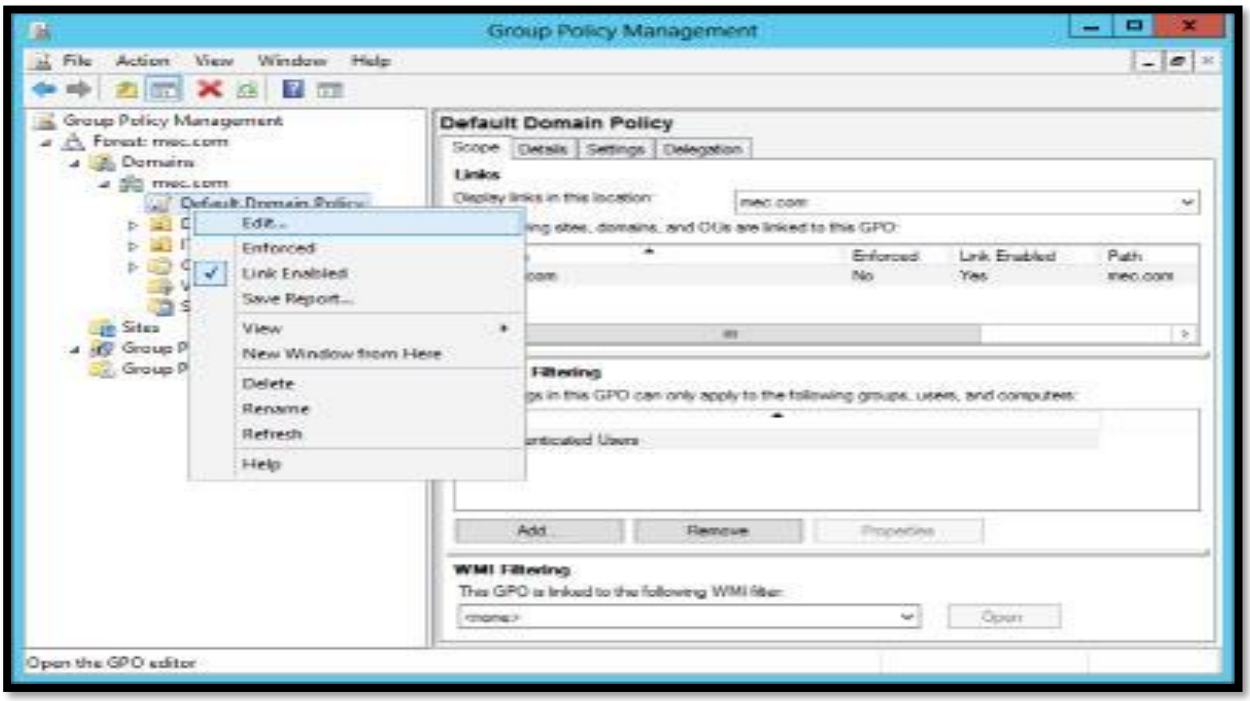


لإضافة وتعديل السياسات الخاصة بكلمة المرور، إتبع الخطوات التالية :

-نذهب لـGroup Policy Management .

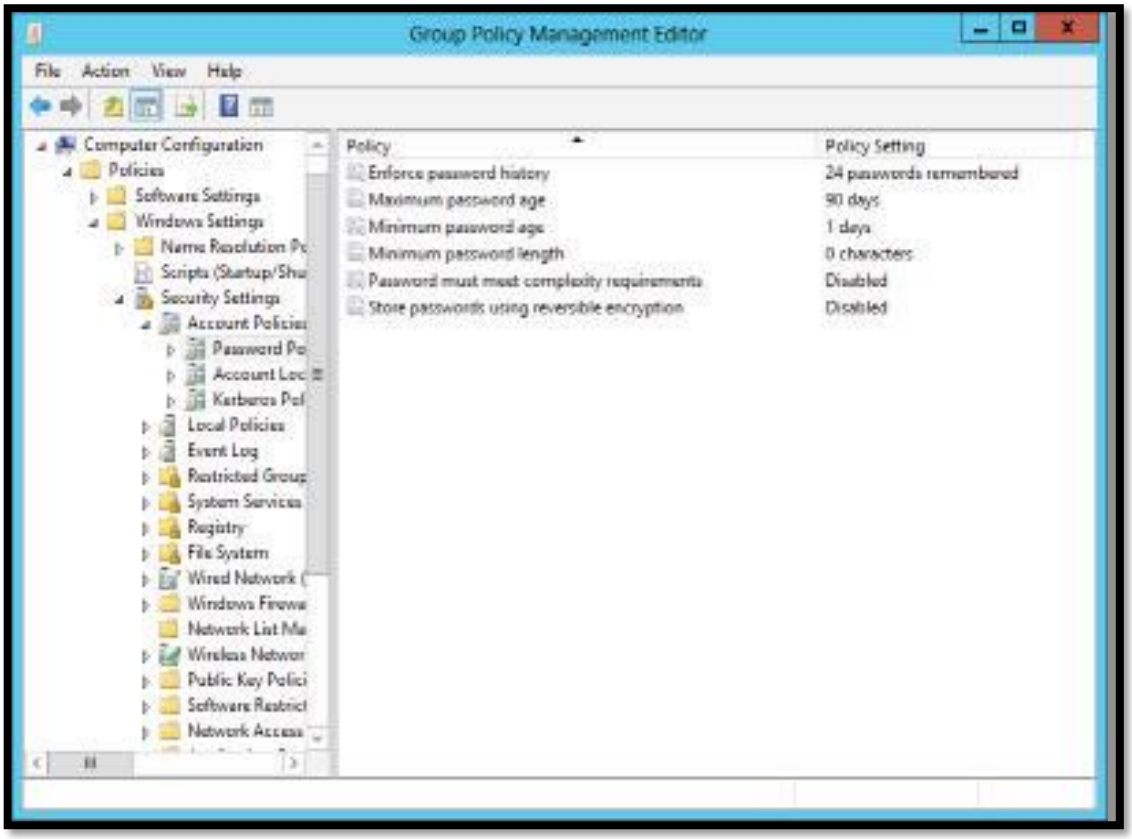
-نضغط بالزر الأيمن على Default Group Policy ونختار Edit .

Chapter 1 : Installing and Configuring Windows Server 2012 Group Policy Management
Computer Configuration -
. Policies -
. Windows Settings -
. Security Settings –
. Account Policies -
. Password Policy -



ثم نقوم بتغيير الإعدادات كما في الشكل التالي ،وبذلك يمكن للمستخدمين تعيين كلمات مرور بسيطة حروف أو أرقام فقط.

لا ينصـح أبداً بإسـتخدام كلمات المرور التي تتكون من أرقام فقط أو حروف، وذلك لأن مسـتوى الحماية في أي مؤسسـة يجب أن يكون عالي جداً، حتى لايتم تخمين كلمات المرور أو سـرقتها من قبل أشـخاص غير مخولين بإستخدام شبكة المؤسسة.



لإعداد الرسالة الترحيبية إتبع الخطوات التالية :

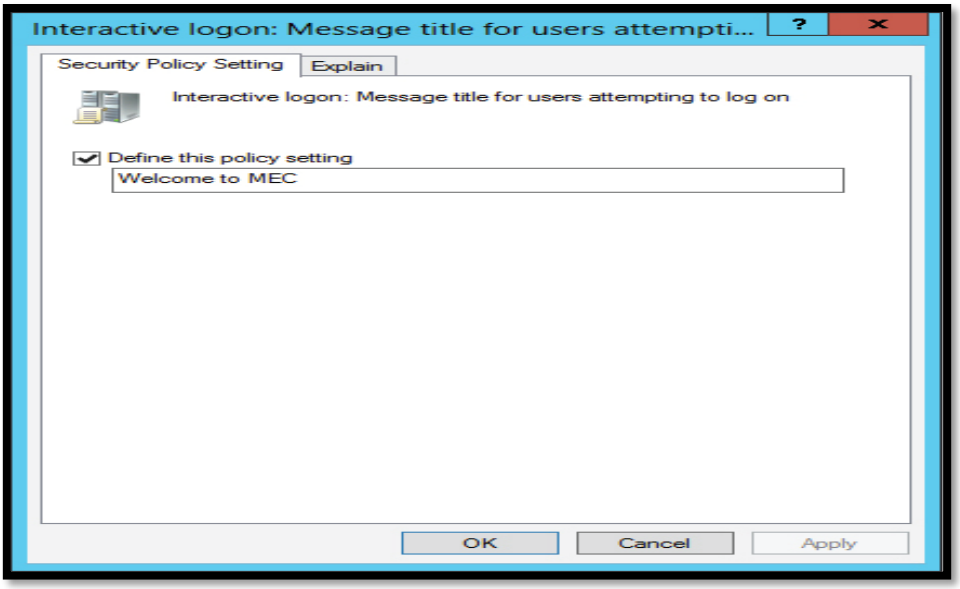
-نذهب لـGroup Policy Management.

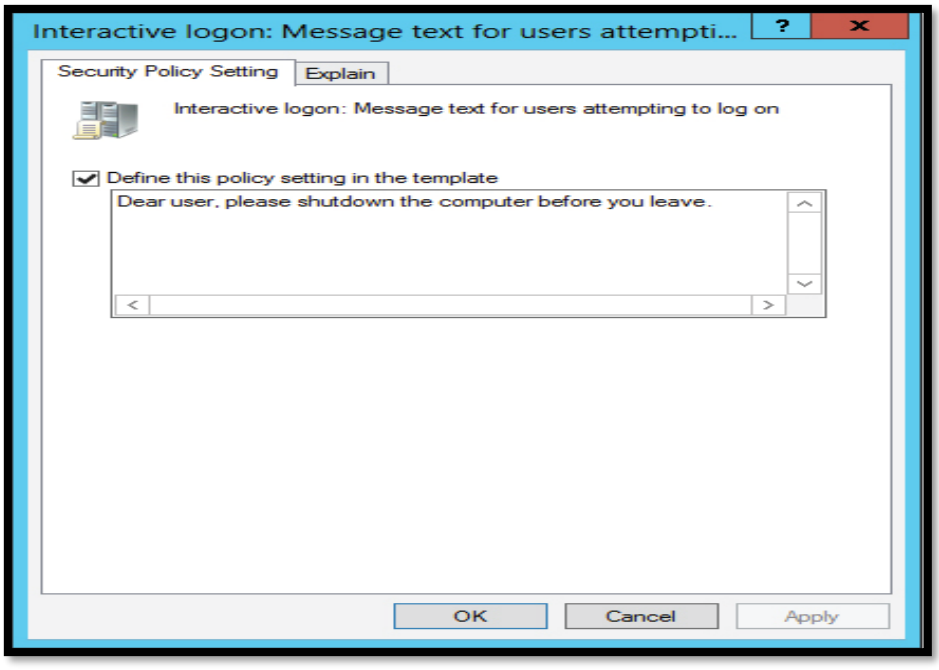
- Computer Configuration .
- Windows Settings .
- Security Settings .
- Local Policies .
- Security Options .

ثم بعد ذلك نبحث

١- Interactive logon: Message title for users attempting to log on عنوان الرسالة

٢- Interactive logon: Message text for users attempting to log on نص الرسالة.

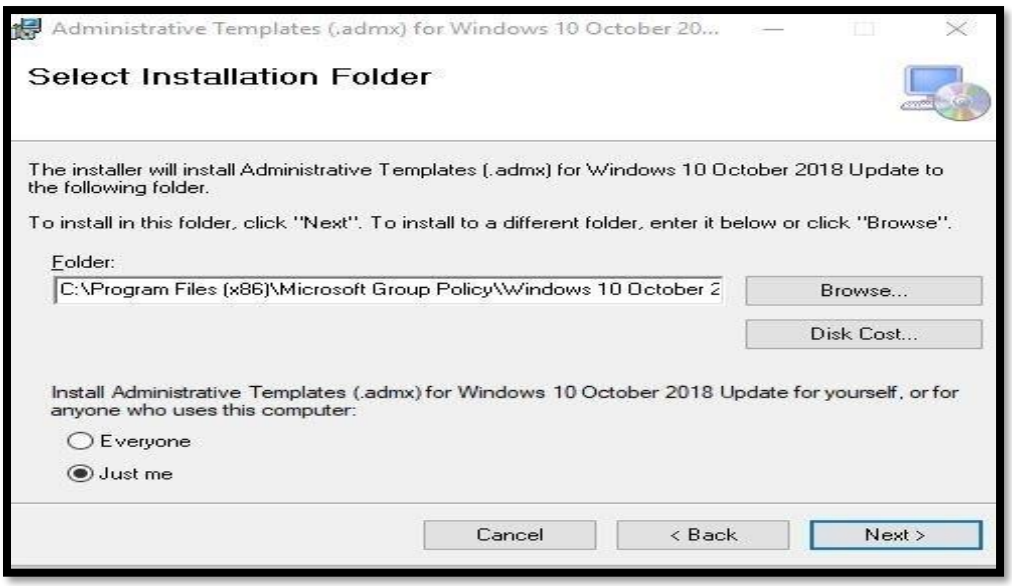




❖ تكوين المتجر المركزي لنهج المجموعة

قم بتنزيل قوالب نهج المجموعة الجديدة، عندما تقوم Microsoft بإصـــدار إصـــدارات جديدة من Windows ، فإنها تقوم أيضاً بإصدار قوالب نهج مجموعة جديدة.

قم بتنزيل واستخراج القوالب على جهاز الكمبيوتر الخاص بك.

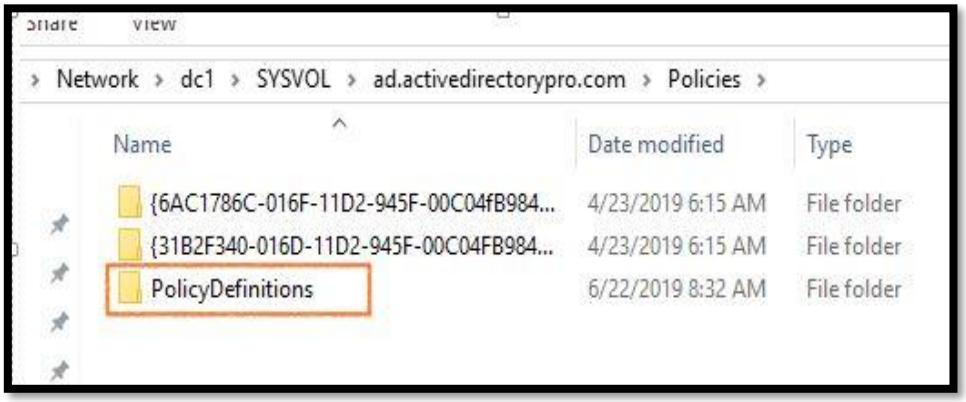


إنشاء مجلد تعريفات السياسة

الخطوة التالية هي إنشاء مجلد PolicyDefinitions في دليل SYSVOL

المسار الدقيق هو SYSVOL\Yourdomain\Policies

انقر بزر الماوس الأيمن ، وقم بإنشاء مجلد جديد وق بتسميته PolicyDefinitions



يمكنك أيضــاً العثور على هذا المســار عن طريق تســجيل الدخول إلى وحدة تحكم المجال الخاصــة بك والتصفح إلى هذا المسار

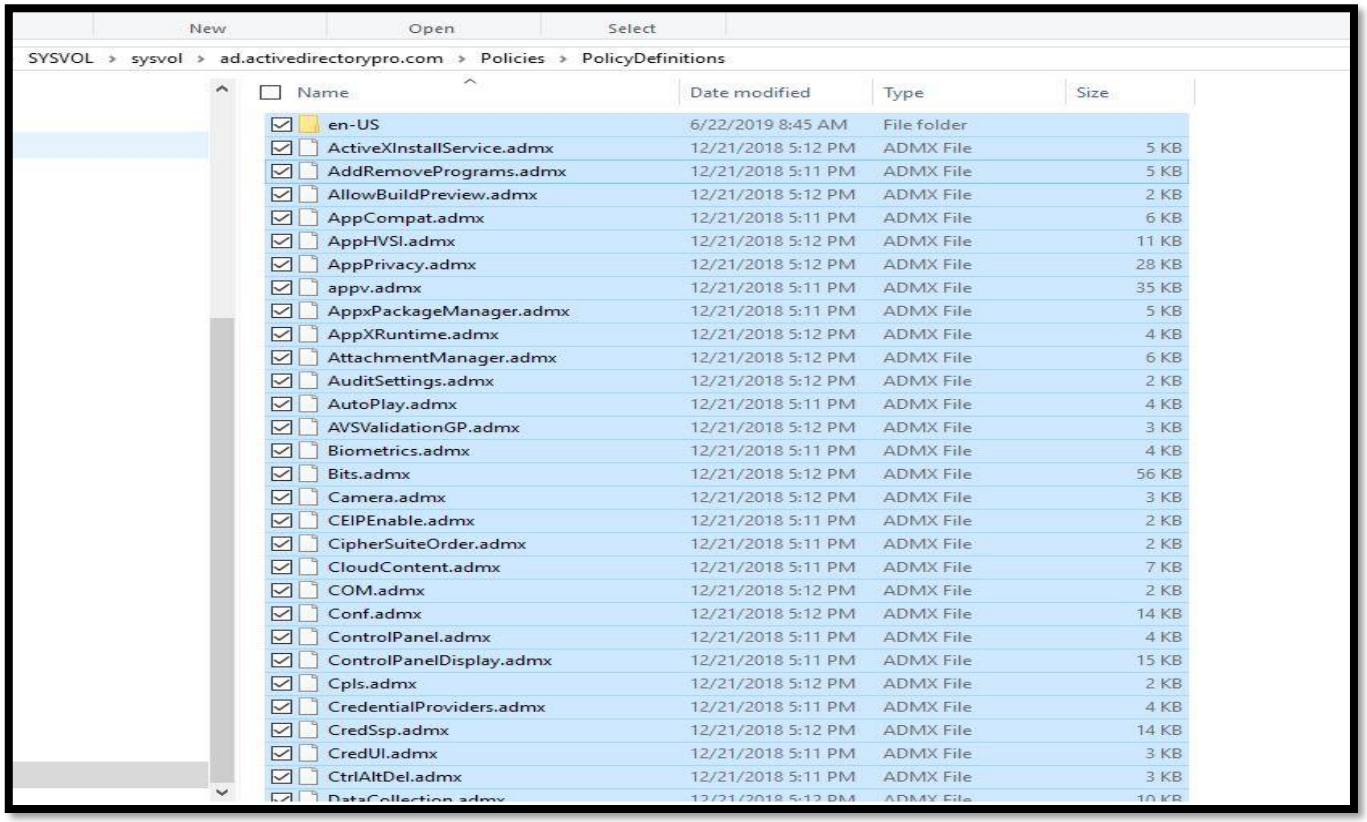
C:\Windows\SYSVOL\sysvol\YOURDOMAIN\Policies

قم بنسخ الملفات المستخرجة في مجلد PolicyDefinitions

ستلاحظ وجود مجلدات للغات مختلفة في المجلد المستخرج.

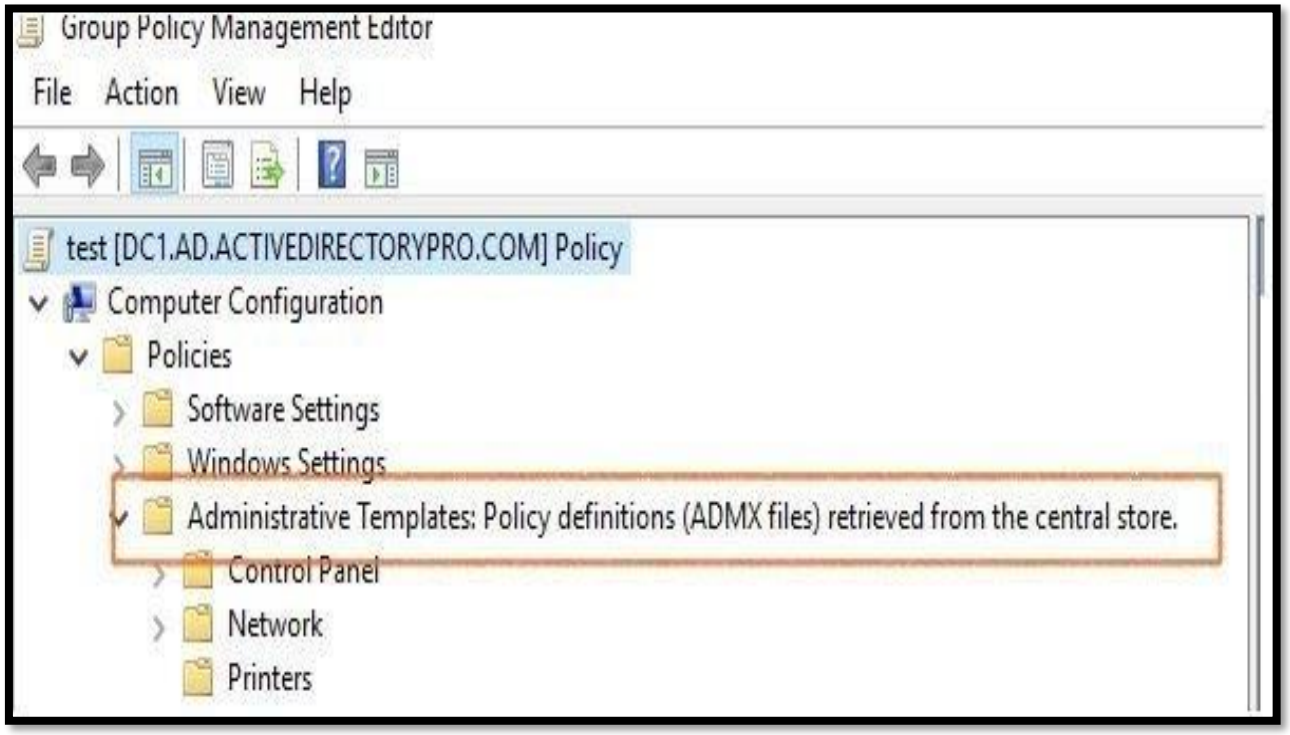
يمكنك نســخها جميعاً إلى المتجر المركزي ولكن اختر فقط نســخ اللغة التي أحتاجها، ليســت هناك حاجة لنسخ القوالب للغات أخرى، ولكن مرة أخرى، لن تكون هذه مشكلة كبيرة إذا قمت بذلك.

يمكنك أن ترى أدناه لقد قمت بنسخ القوالب المستخرجة إلى مجلد PolicyDefinitions الذي تم إنشاؤه في الخطوة 3



تحقق من المتجر المركزي لنهج المجموعة

للتحقق من عمل المتجر المركزي، قم بتحرير GPO وانتقل إلى تكوين الكمبيوتر \ السياسات \ القوالب الإدارية، يجب أن يقول الآن أنه تم استرداد التعريفات من المتجر المركزي مثل لقطة الشاشة أدناه



تمرين:

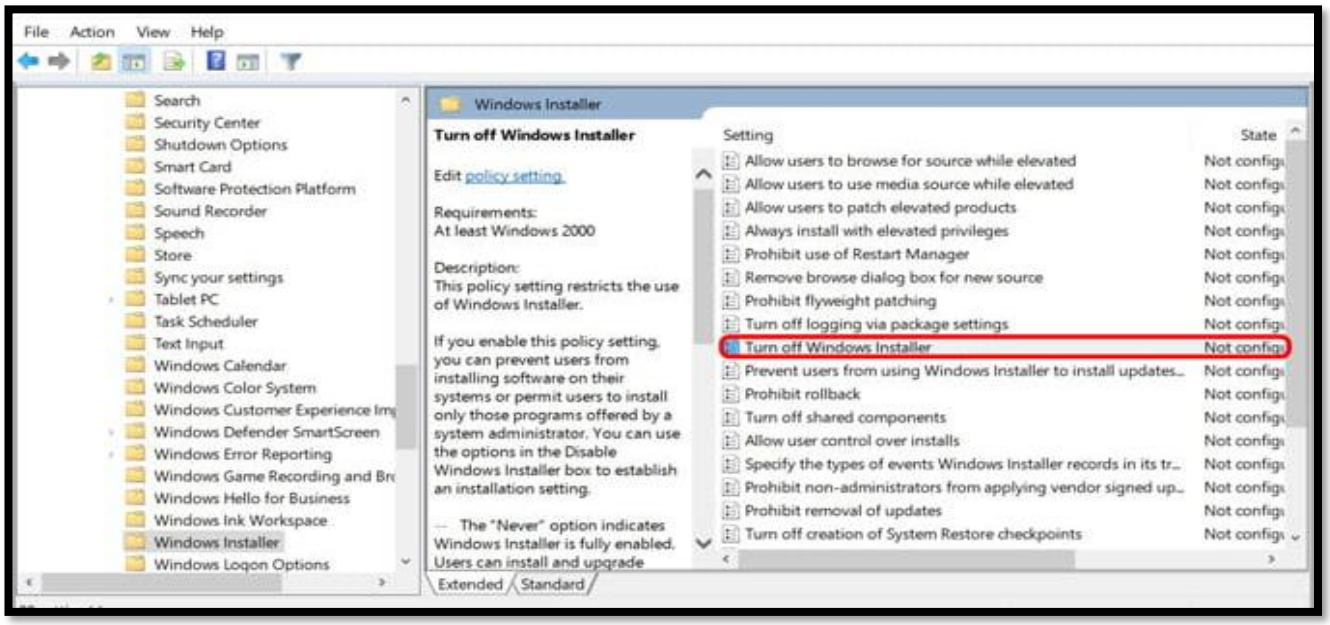
قم بتنزيل واستخراج القوالب على جهاز الكمبيوتر الخاص بك متبعاً الخطوات التي تمت دراستها ؟

الفصل العاشر: تأمين نظام تشغيل الشبكات Windows Server 2012R2

أنواع القواعد هي كما يلي:

تعطيل تثبيت التطبيقات

من خلال عدم السـماح للمسـتخدمين بتثبيت تطبيقات مُختلفة، يمكنك تقليل مقدار الصـيانة والتنظيف المطلوب عند تثبيت شيء سيء حيث أنه أيضاً أحد الأسباب المحتملة لوصول البرامج الضارة، هذا مفيد أكثر خاصةً في المدارس ، أين تُريد من الطلاب الوصول إلى ما هو مطلوب استخدامه فقط.

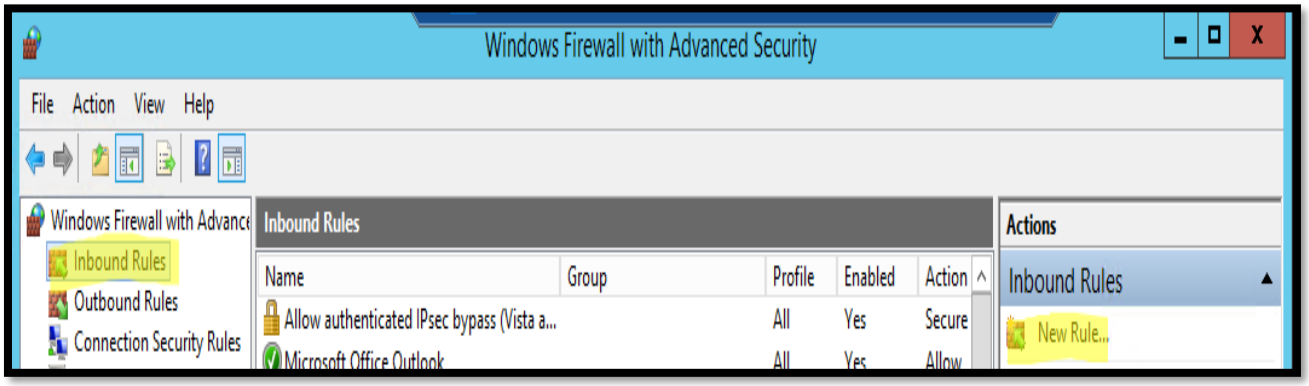


أذا كنت ترغب في تقييد المستخدمين من تثبيت التطبيقات أو تشغيلها ،يُمكنك تعيين ذلك عن طريق فتح سياسة المجموعة والإنتقال إلى Computer Configurations > Administrative Templates > Windows Components > Windows Installer

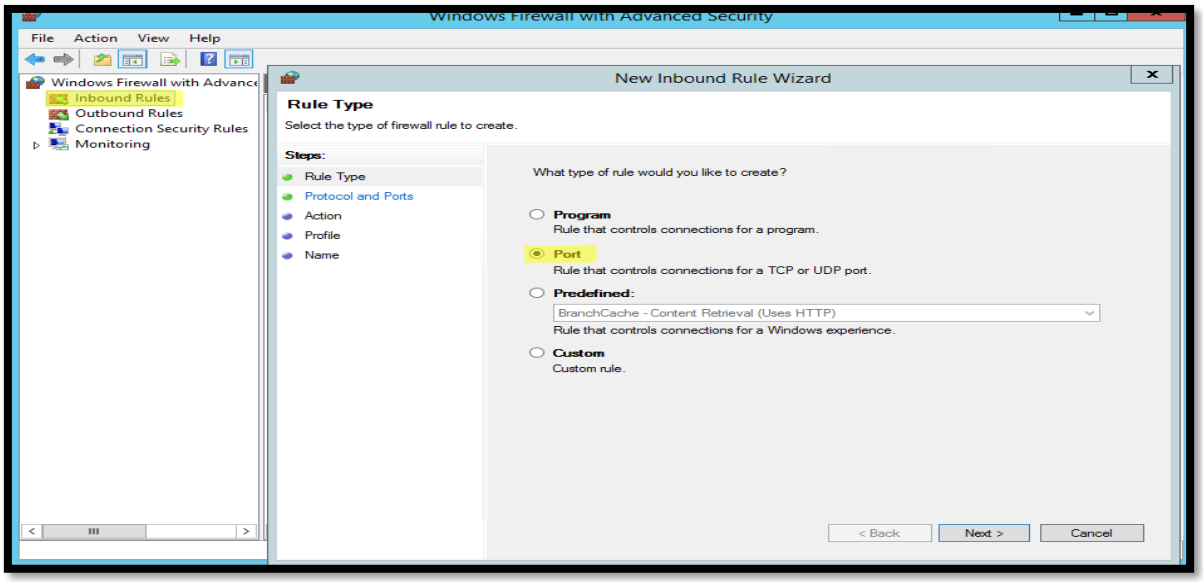
وعليك النقر نقراً مزدوجاً فوق خيار Turn off Windows Installer قم بتغيير الإعداد إلى تمكين وتأكد من أن الخيار يُشير إلى "For Non-managed applications only"، حتى يتمكن المُستخدم من تثبيت جميع التطبيقات التي تسمح بها الإدارة، انقر الآن على تطبيق وقم بإعادة تشغيل جهاز الكمبيوتر لإجراء التغييرات.

❖تكوين الجدار الناري في windows بإعدادات متقدمة

افتح " جدار حماية Windows " على النظام المضيف المثيل الافتراضي ملقم SQL وانقر فوق قاعدة جديدة ضمن القواعد الواردة .



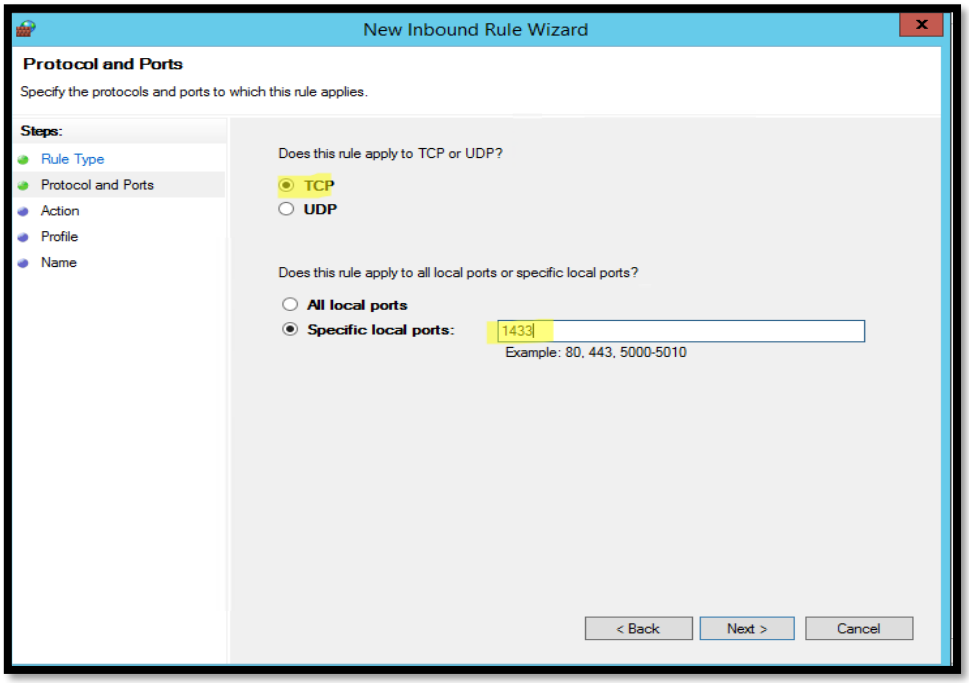
حدد الخيار منفذ وانقر فوق التالي.



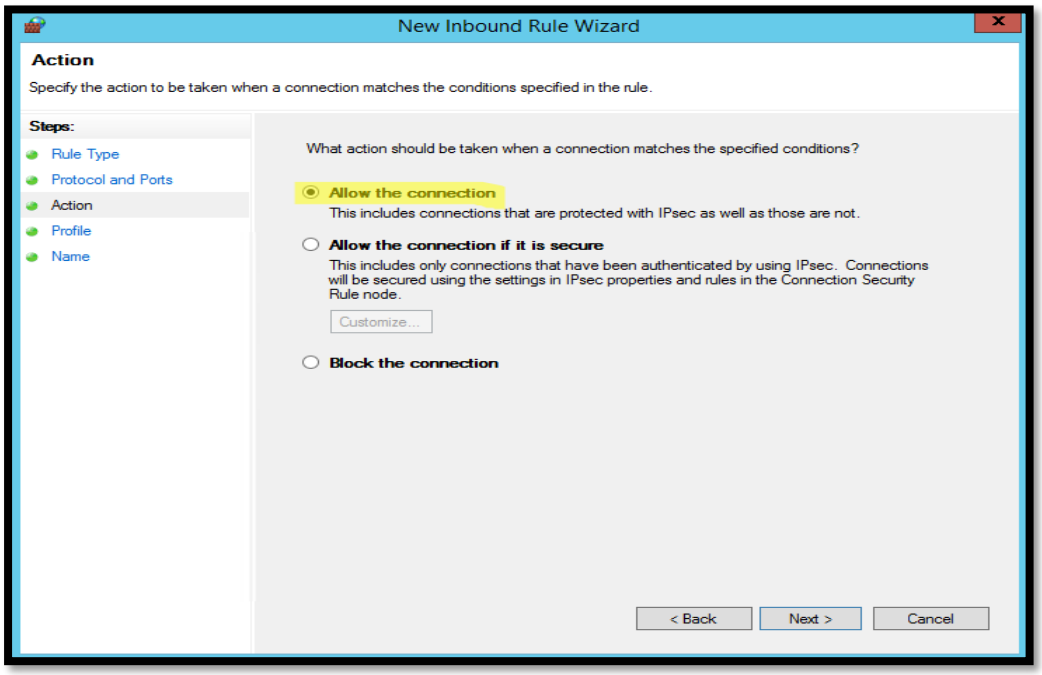
في الشاشة التالية:

TCPكالبروتوكول.

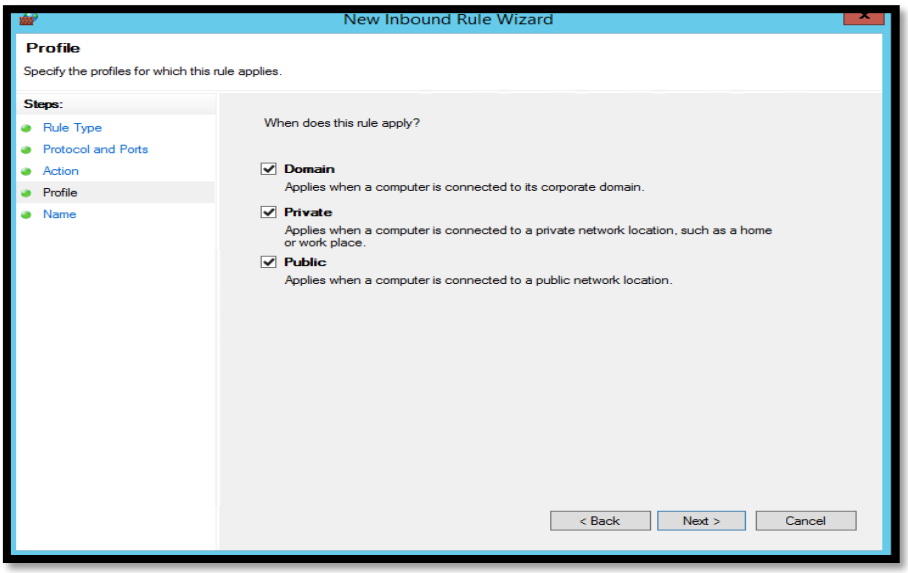
حدد المنافذ المحلية المحددة وتحديد القيمة ك ١٤٣٣(بورت خاص بالـ (SQL server) ثم انقر فوق التالي.



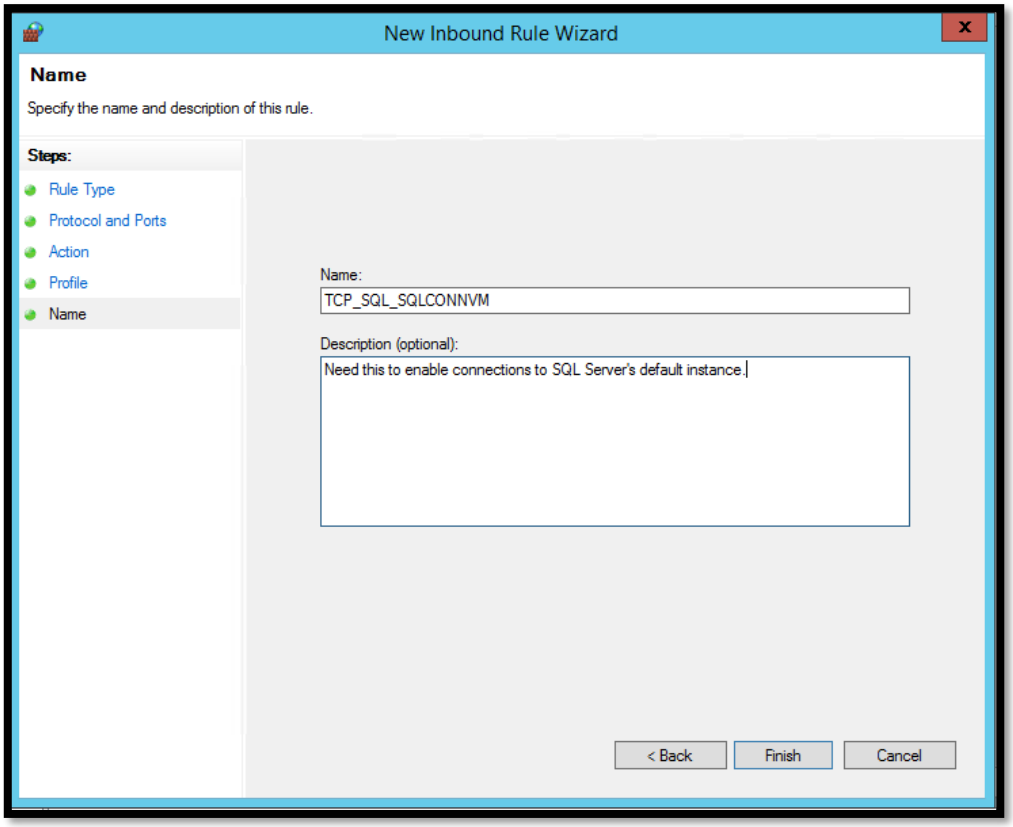
في الشاشة التالية تحديد السماح الاتصال ثم انقر فوق التالي .



في الشاشة التالية حدد الخيار الأكثر ملاءمة للبيئة الخاصة بك ثم انقر فوق التالي.



إعطاء الاسم للقاعدة الخاصة بك ووصفاً واضحا للرجوع إليها مستقبلا في التالي الشاشة وانقر فوق إنهاء.



بمجرد الانتهاء، يجب أن تشاهد إنشاء تلك القاعدة وتمكينها بشكل افتراضي.

تمرين :

كون الجدار الناري مستخدمً الاعدادات المتقدمة ؟

الفصل الحادي عشر : الأنظمة الوهمية و Hyper-v

تطبيق Hyper-V

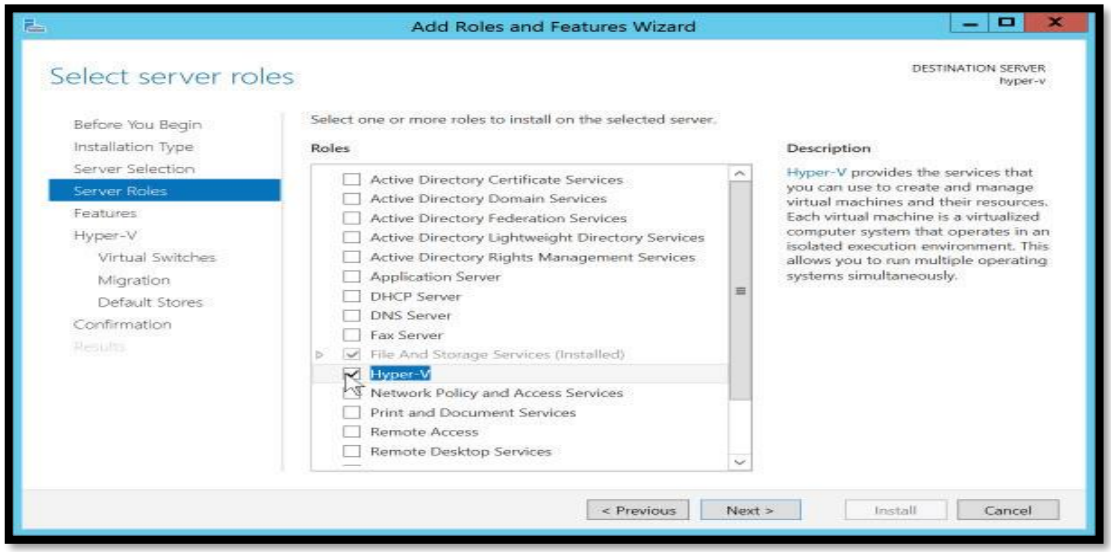
ومن هذه الأدوات المستخدمة لتثبيت أنظمة التشغيل الإفتراضية هو " Hyper-V" وهو عبارة عن " Role" يتم تثبيتها في Windows Server 2012 كما سوف نرى لتثبيت " Hyper-V " **إتبع الخطوات التالية:**

-نذهب لـ Server Manage .

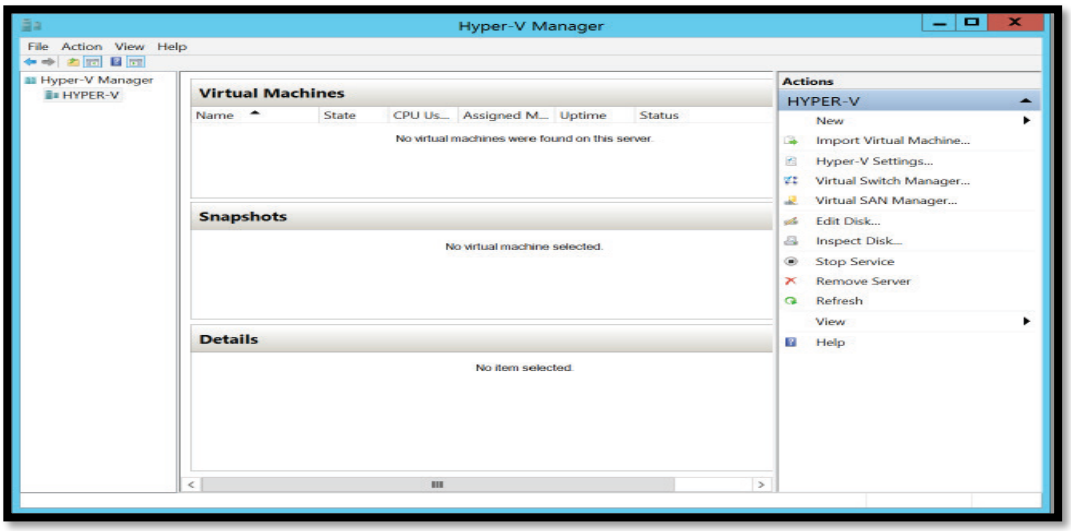
- Add Roles and Features .

-نحدد على " Hyper-V " كما نلاحظ في الشكل التالي.

-نضغط على Next دون تغير الإعدادت، ثم نقوم بإعادة تشغيل الجهاز.

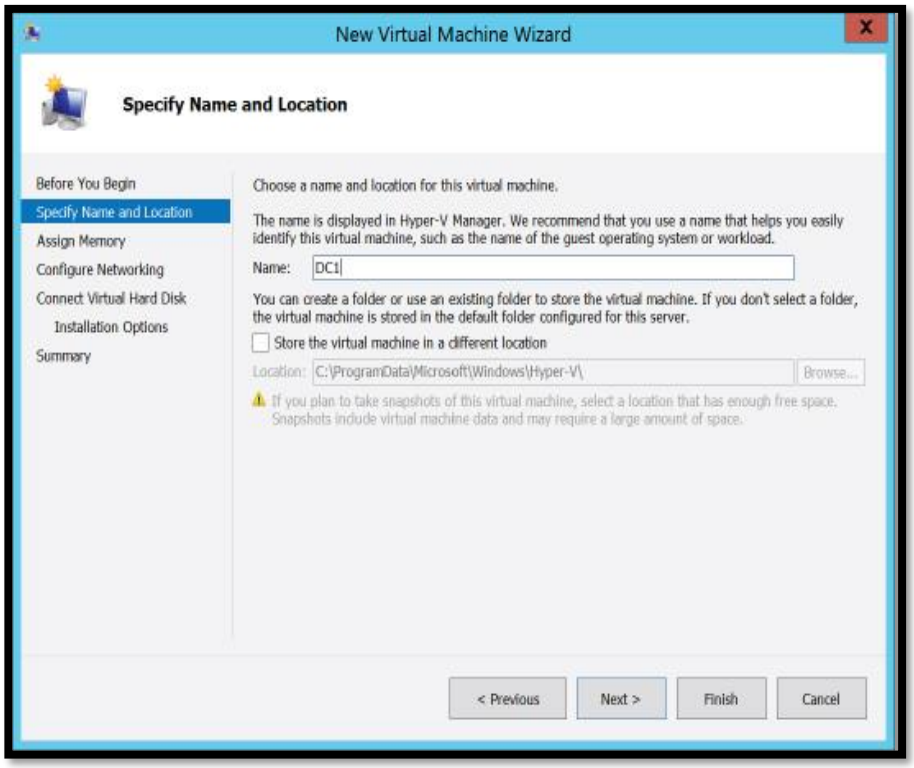


يحتوي الـ " Hyper-V " على مجموعة من الخصائص كما نلاحظ في الشكل التالي ،وسوف نتعرف على بعض هذه الخصائص كما سوف نرى -سوف نقوم بإنشاء نظام وهمي جديد وذلك بالضغط على " New "ثم "New Virtual Machine".



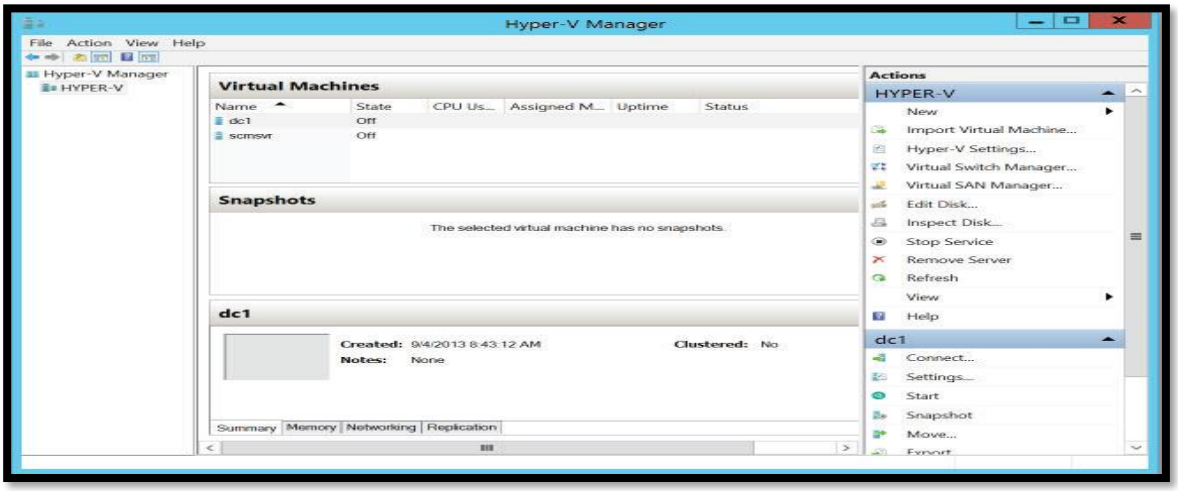
كما نلاحظ في الشكل التالي سوف يقوم المعالج بإضافة بيانات وخصائص نظام التشغيل الإفتراضي نحدد الـ " RAM " التي نريد إستخدامها ثم نحدد المساحة التخزينية HDD لنظام التشغيل.

-بعد ذلك نحدد الطريقة المستخدمة للتثبيت DVD, ISO ويمكننا تجاهل عملية التثبيت، وسوف يقوم المعالج بوضع الإعدادات فقط



Virtual Switch Manager

بعد الإنتهاء من الإعدادات الخاصة بنظام التشغيل، نضغط على " Virtual Switch Manager " كما نلاحظ في الشكل التالي:



سوف نلاحظ بأن هناك ثلاثة أنواع من الـ " Network Adapter " ولكل نوع خصائص مختلفة عن الآخر سوق نبينها في الجدول التالي:

External	Internal	Private
يسمح هذا الخيار بالإتصال بالشبكات الخارجية"Internet".	يمكن من خلال هذا الخيار عمل إتصال ما بين الجهاز الحقيقي والأنظمة إفتراضية.	يمكن من خلال هذا الخيار عمل إتصال مابين الأنظمة الإفتراضية فقط.

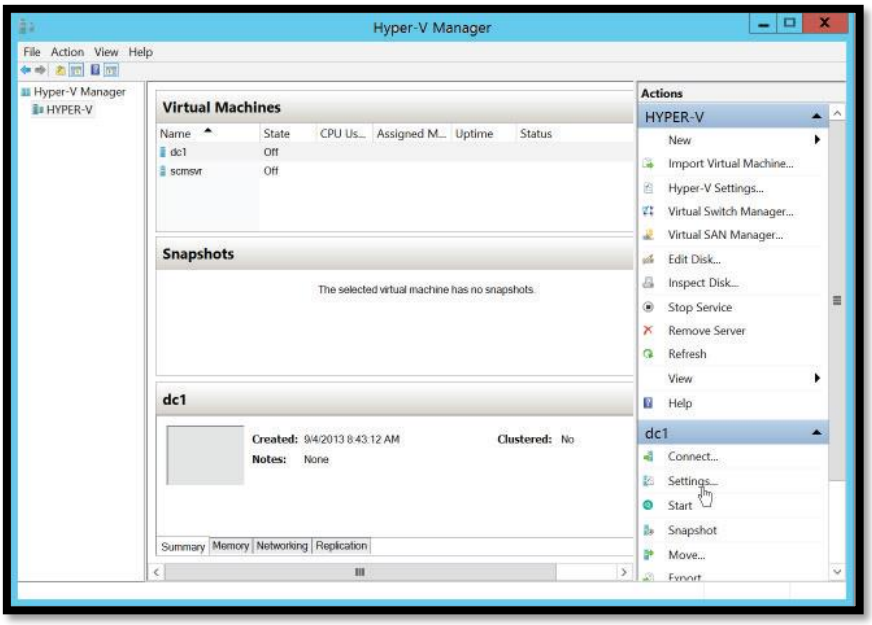
لإضافة " Network Adapter " نحدد على النوع الذي نريده، ثم نضغط "Add".

-سوف نقوم في المثال الحالي بإنشاء الـ " Network Adapters " التالية Internal, Private

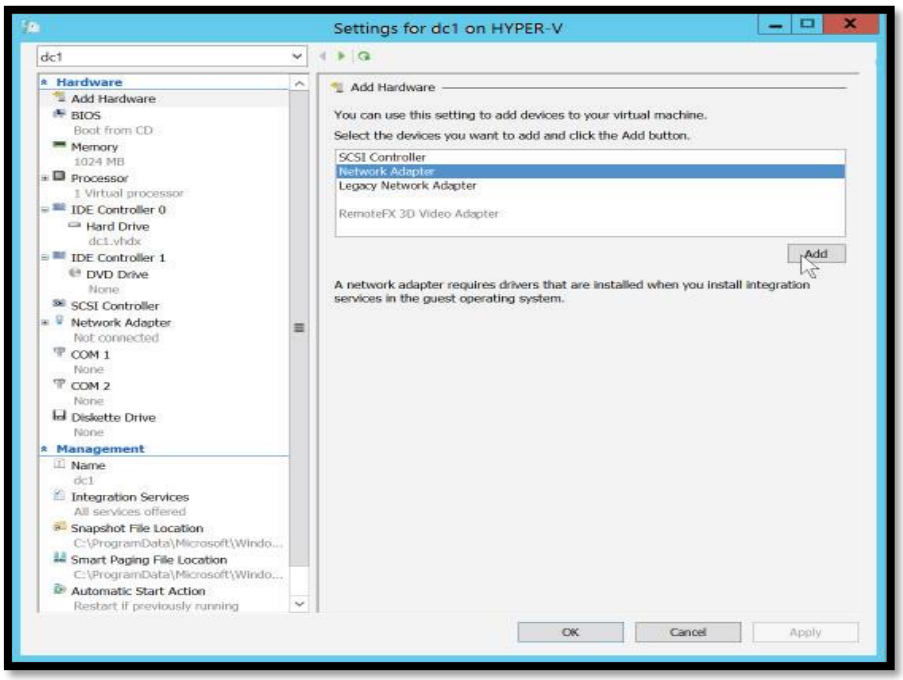
-بعد ذلك نضغط على OK لإنهاء الإعدادات.

-نحدد على نظام التشغيل الإفتراضي كما نلاحظ في الشكل التالي ثم نضغط " Settings " يجب أن نتأكد بأن نظام التشغيل لا يعمل، وذلك

لأنه لا يمكن تغيير الإعدادت أثناء عمل النظام.



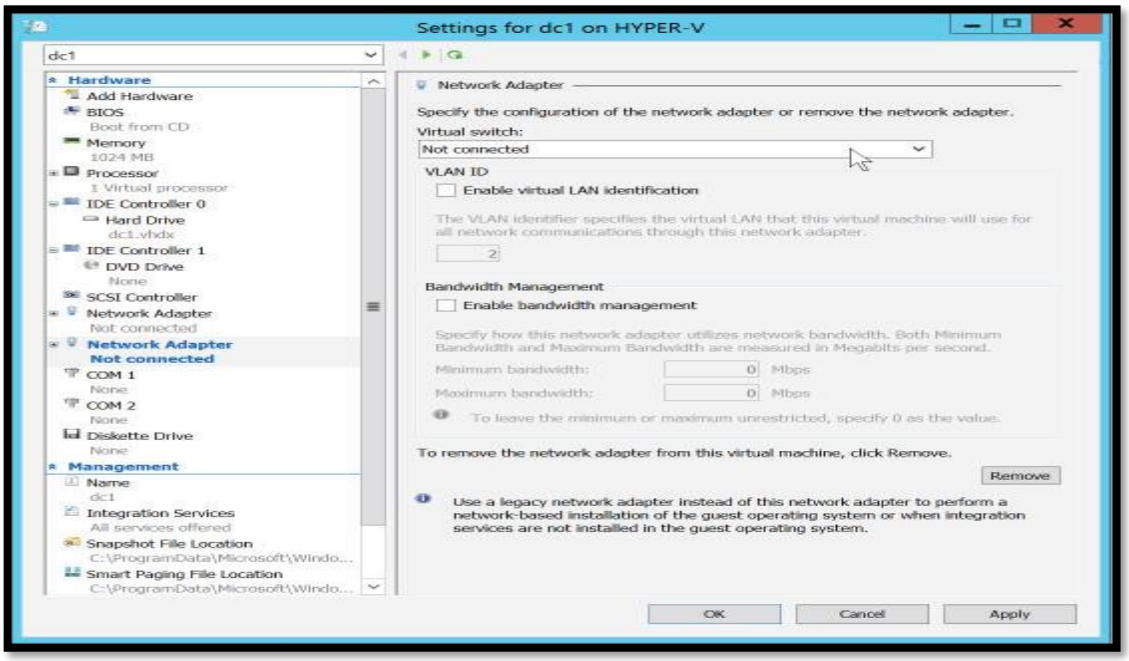
- نضغط على" Add Hardware " ، ثم " Network Adapter " ثم نضغط على "Add"



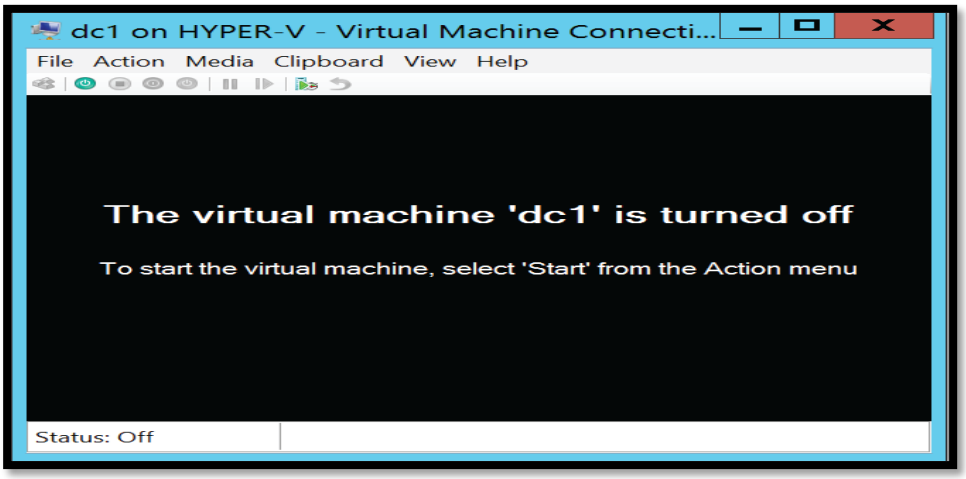
-من قائمة الـ " Virtual Switch " نختار" Internal " ، ثم نكرر العملية لإضافة "Private".

-بعد ذلك نضغط على OK لإنهاء الإعدادات.

-بعد ذلك نضغط بالزر الأيمن على نظام التشغيل ونختار "Connect".



كما نلاحظ في الشكل التالي هناك مجموعة من الخيارات التي يمكن من خلالها التحكم بنظام التشغيل الإفتراضي، ولكل خيار خاصية معينة يقوم بها كما هو موضح في الشكل.

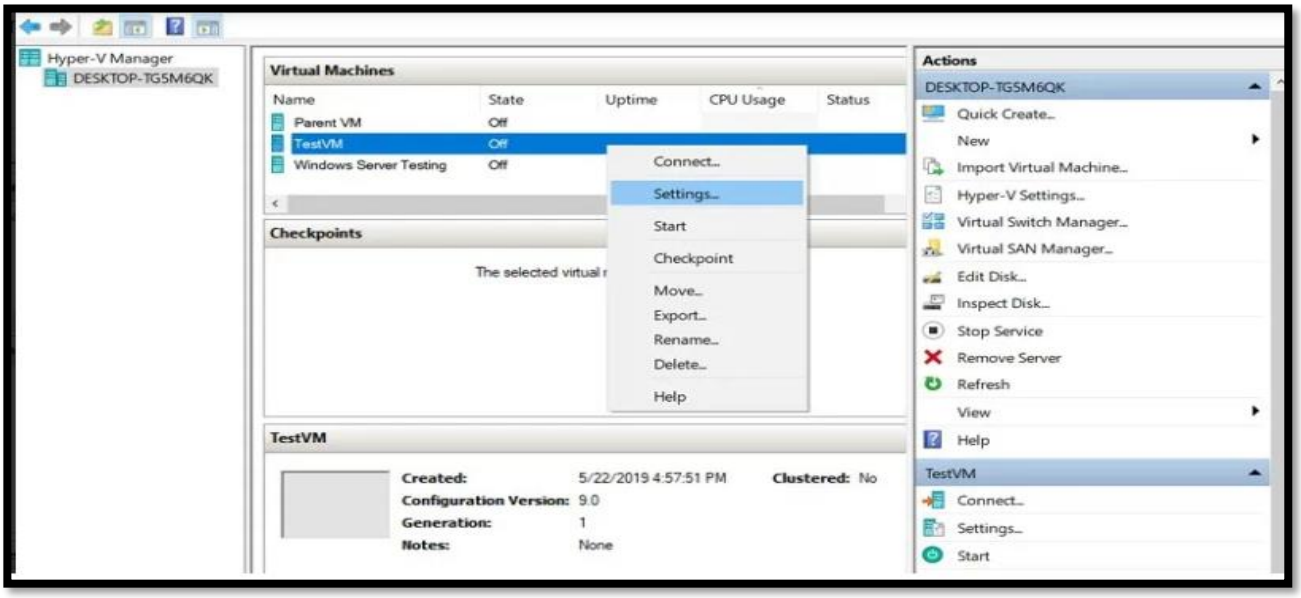


بعد أن نقوم بتشغيل نظام التشغيل، يجب أن نتأكد من تغيير عناوين الشبكة حتى يتم الإتصال ما بين الأنظمة الإفتراضية والجهاز الحقيقي

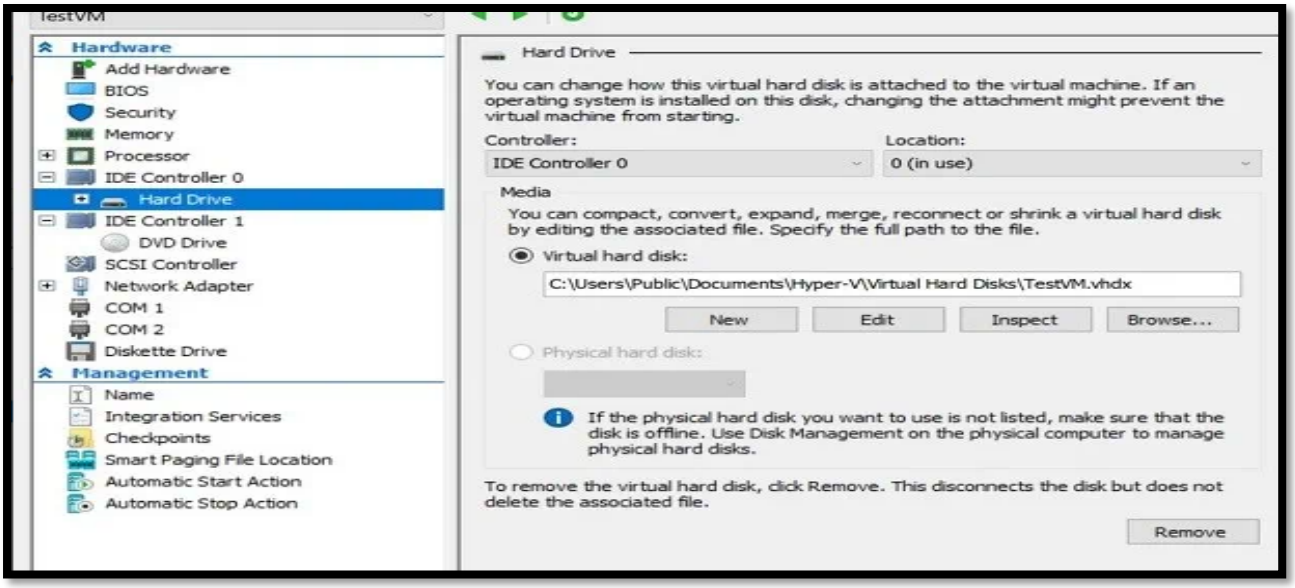
❖ إدارة التخزين في الأنظمة الوهمية

لزيادة المساحة التخزينية للمكنة الافتراضية التي تم إنشاوها **نتبع الخطوات التالية**:

-نقوم باختيار VM ثم نضغط بالزر الأيمن ونختار settings

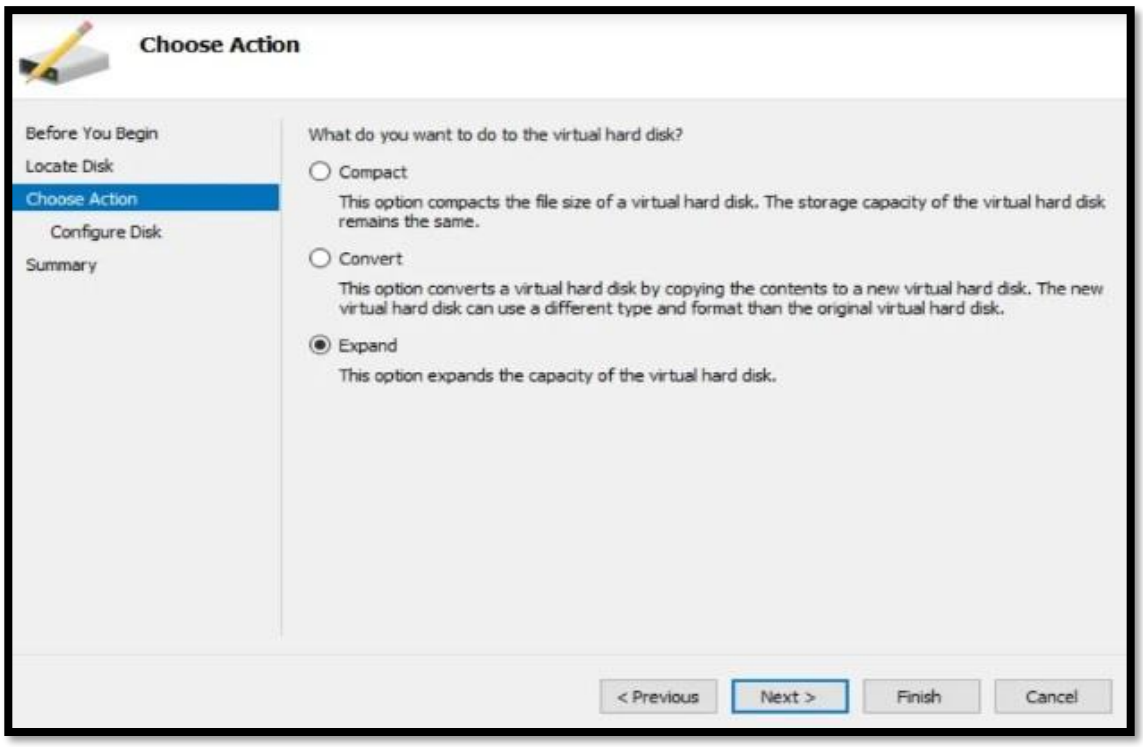


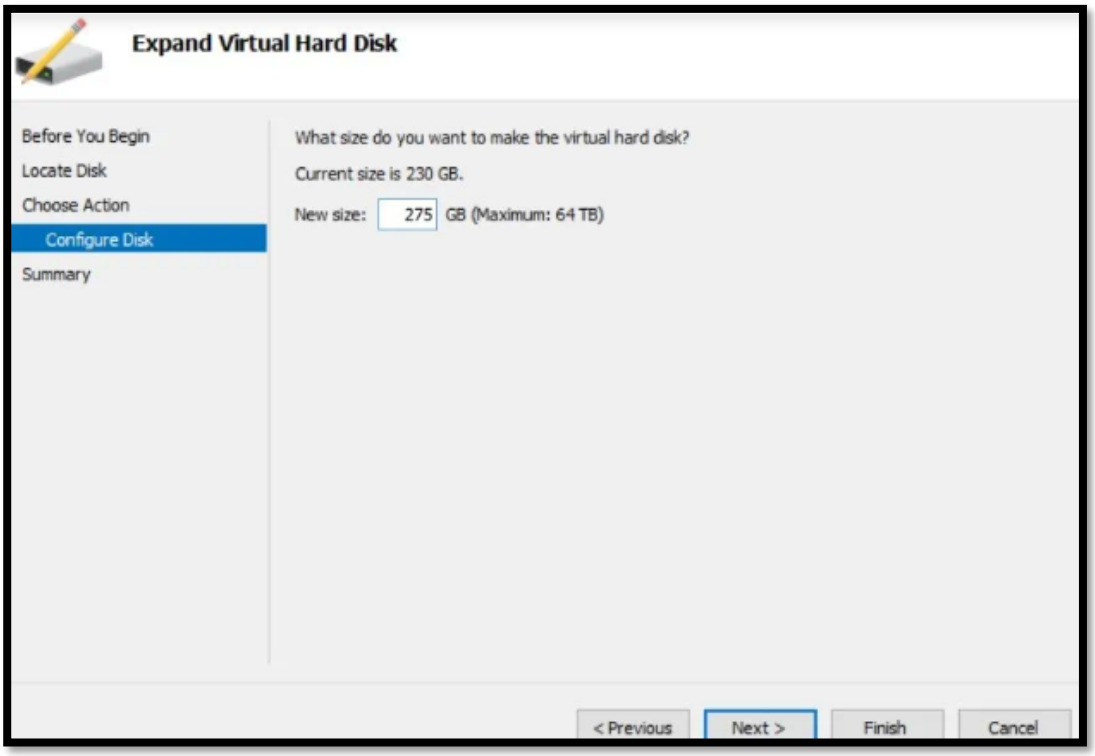
-نختار Hard Drive

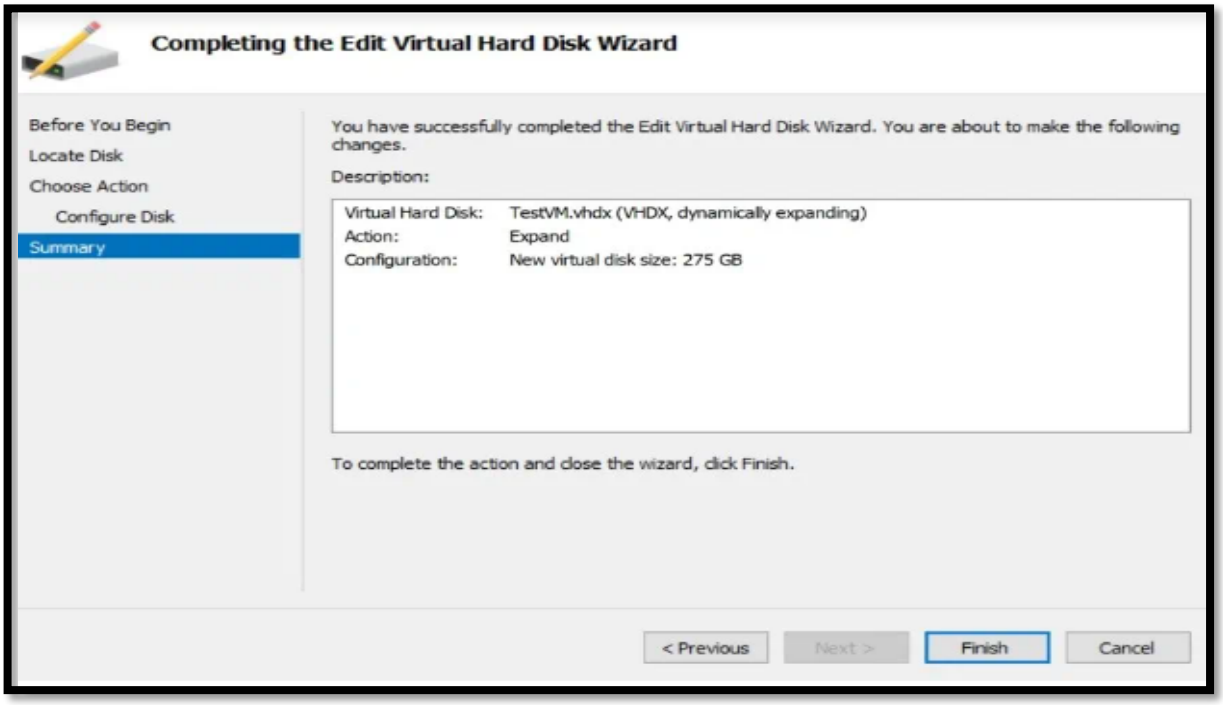


-نضغط على زر Edit

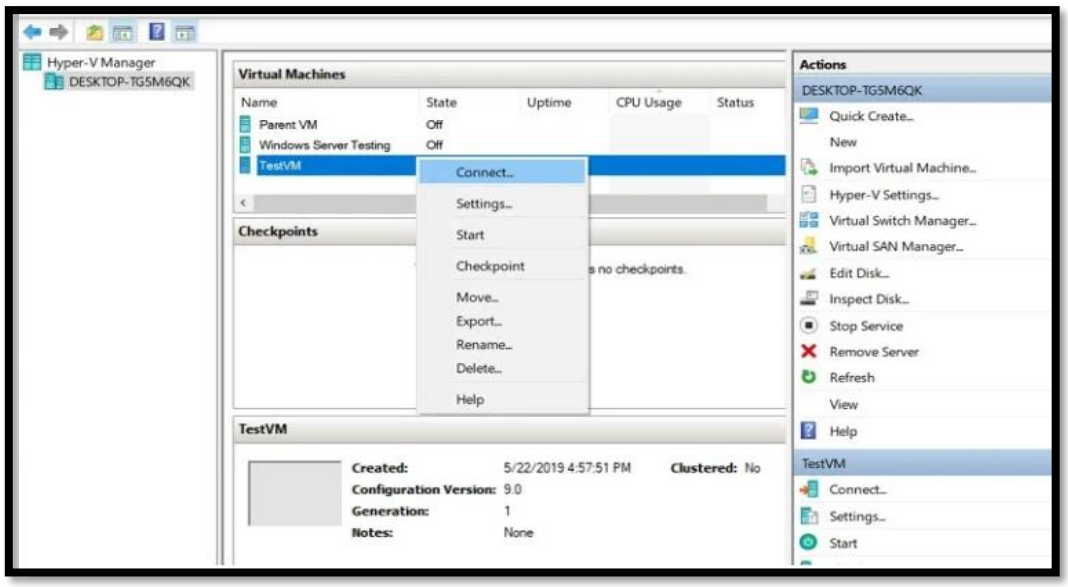
-نقوم باختيار Expand لزيادة مساحة الهارد

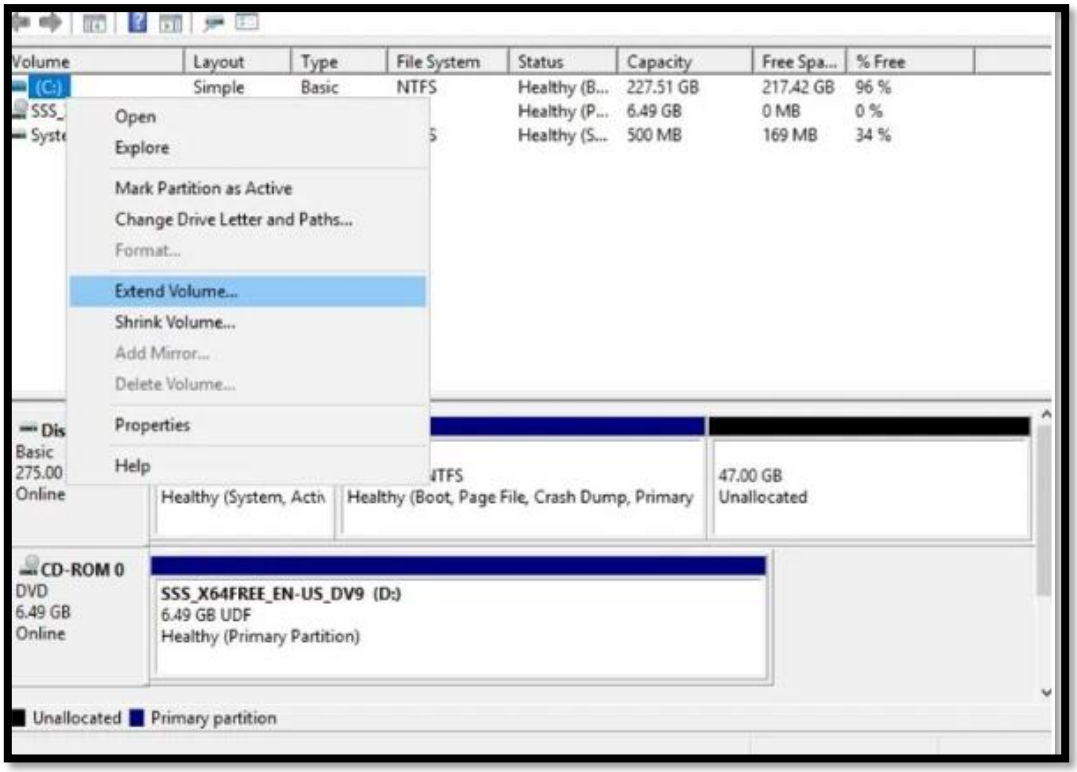


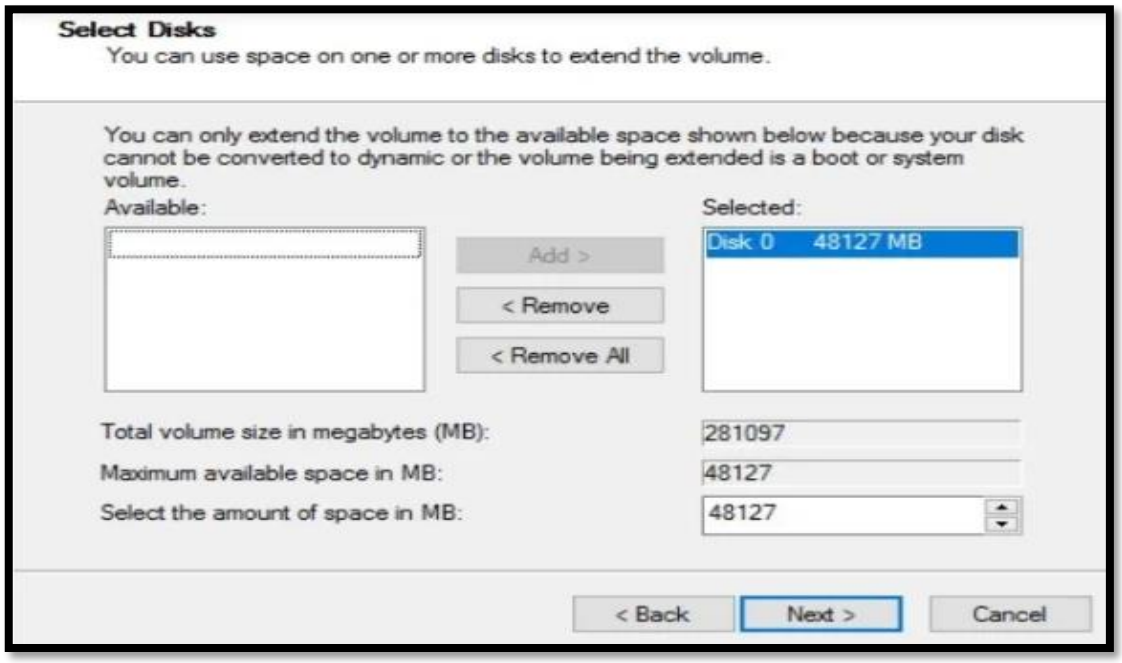

Expand Virtual Hard Disk
Before You Begin
Locate Disk
Choose Action
Configure Disk
Summary
What size do you want to make the virtual hard disk?
Current size is 230 GB.
New size: 275 GB (Maximum: 64 TB)
< Previous
Next >
Finish
Cancel

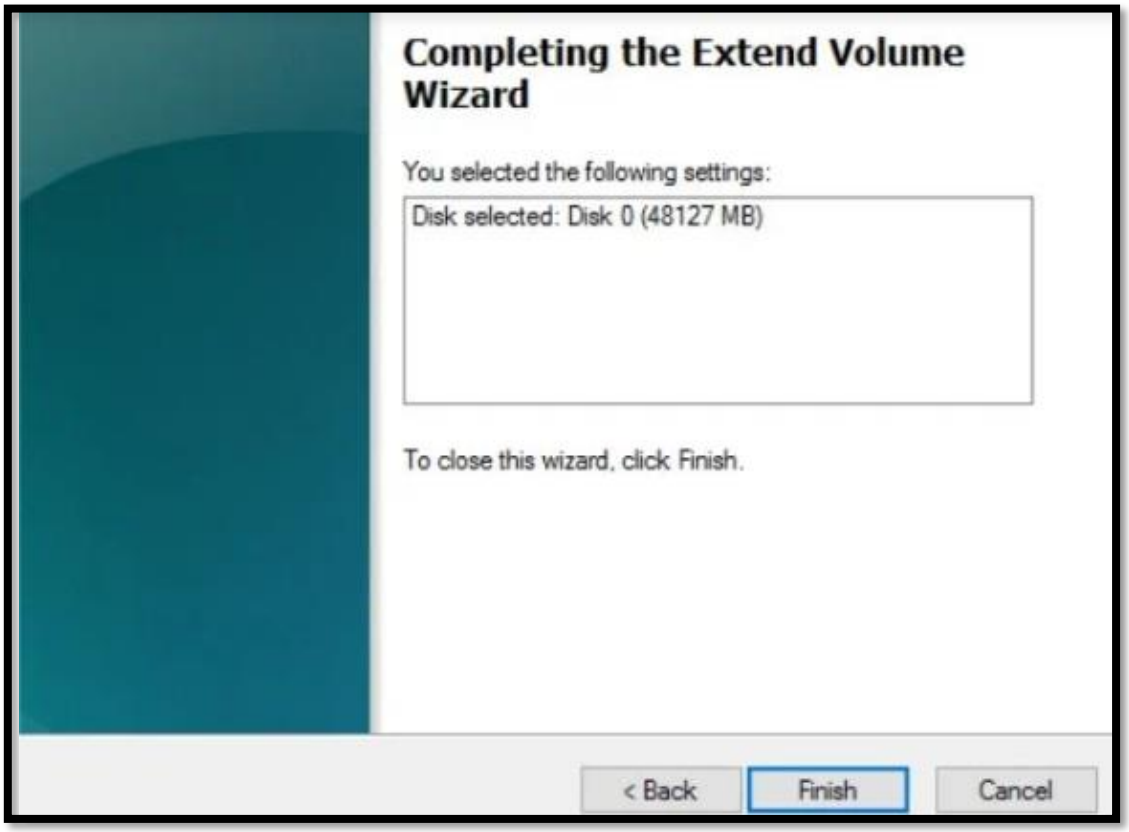

Completing the Edit Virtual Hard Disk Wizard
Before You Begin
Locate Disk
Choose Action
Configure Disk
Summary
You have successfully completed the Edit Virtual Hard Disk Wizard. You are about to make the following changes.
Description:
Virtual Hard Disk: TestVM.vhdx (VHDX, dynamically expanding)
Action: Expand
Configuration: New virtual disk size: 275 GB
To complete the action and close the wizard, click Finish.
< Previous
Next >
Finish
Cancel

-يتم بعد ذلك زيادة القرص الصلب من داخل المكنه الإفتراضية:






Select Disks
You can use space on one or more disks to extend the volume.
You can only extend the volume to the available space shown below because your disk cannot be converted to dynamic or the volume being extended is a boot or system volume.
Available:
Add >
< Remove
< Remove All
Selected:
Disk 0 48127 MB
Total volume size in megabytes (MB): 281097
Maximum available space in MB: 48127
Select the amount of space in MB: 48127
< Back
Next >
Cancel

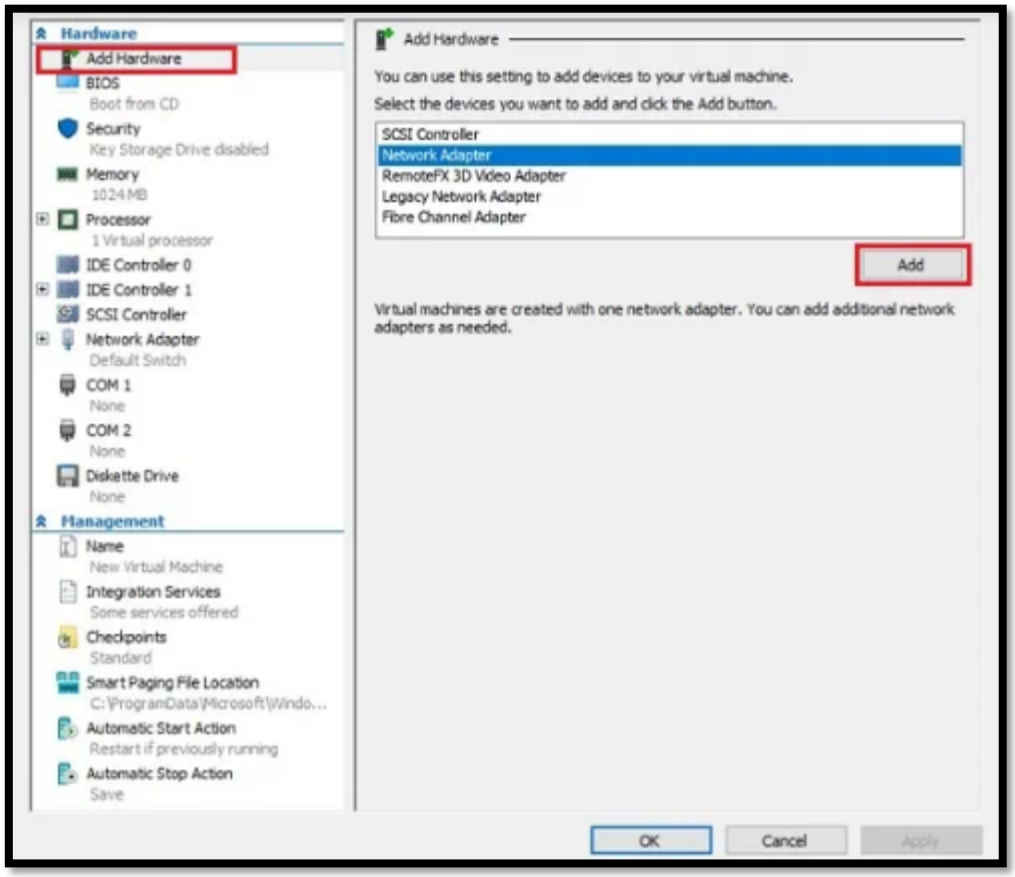

Completing the Extend Volume Wizard
You selected the following settings:
Disk selected: Disk 0 (48127 MB)
To close this wizard, click Finish.
< Back
Finish
Cancel

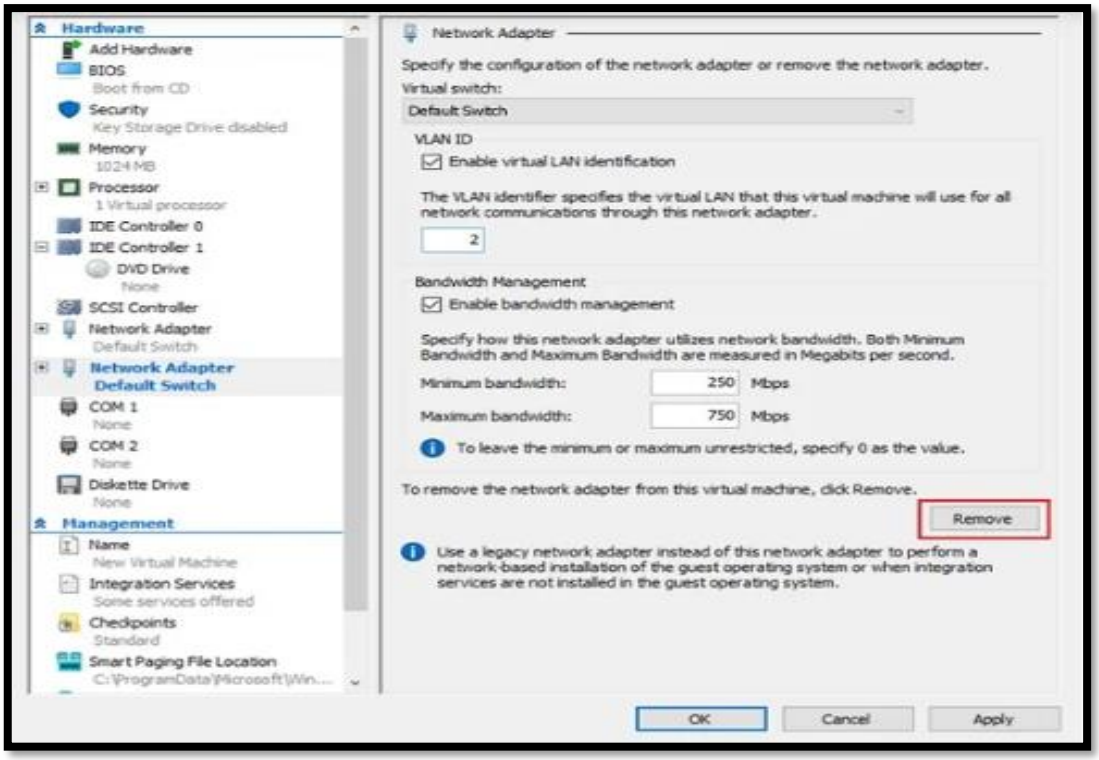
❖ إدارة الشبكات الوهمية

نستطيع تكوين محول شبكة Hyper-V في Hyper-V Manager باستخدام واجهة المستخدم الرسومية (GUI) ، أو في PowerShell عن طريق تشغيل برامج PowerShell النصية، سنناقشهم بالتفصيل الآن ، **لنبدأ بالطريقة السابقة:**

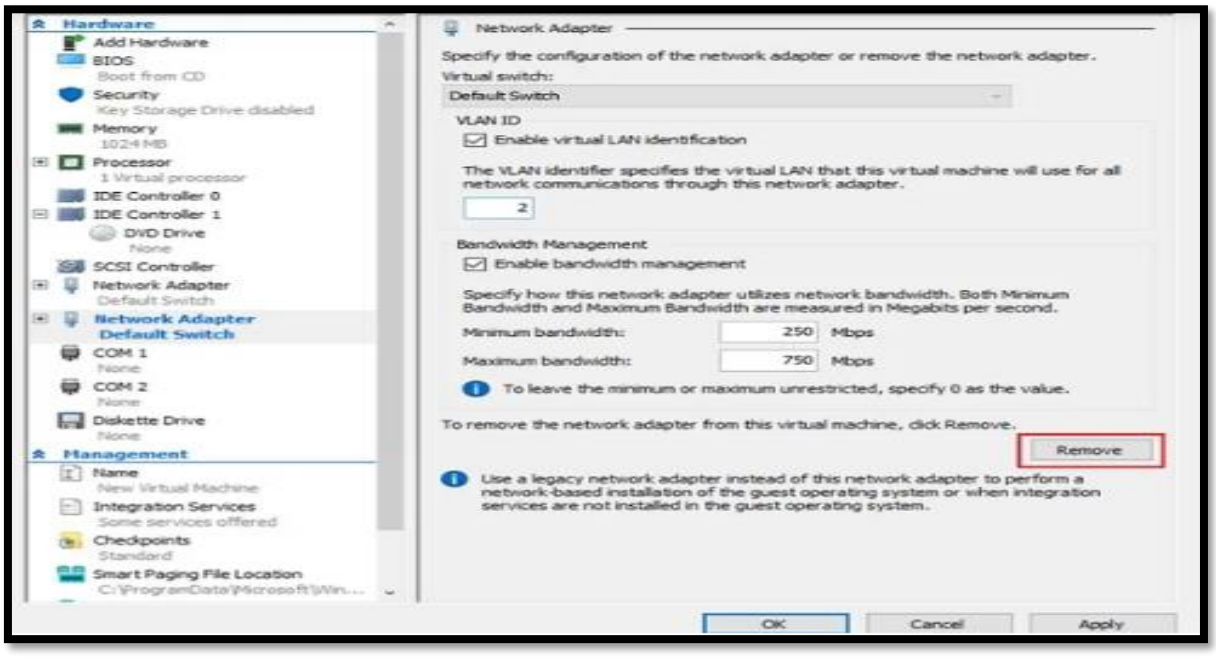
-نقوم بفتح Hyper-v manager

نقوم باختيار VM ثم نضغط بالزر الأيمن ونختار settings





ستتم إعادة توجيهك إلى قسم محول الشبكة حيث يمكنك تكوين محول شبكة Hyper-V جديد عن طريق اختيار محول افتراضي للاتصال به ، عن طريق تمكين تعريف VLAN وتحديد معرف VLAN ، وعن طريق تحديد الحد الأقصى والحد الأدنى لمقدار استخدام النطاق الترددي، يمكنك أيضاً إزالة محول شبكة Hyper-V بالنقر فوق إزالة كما في الشكل التالي



تمرين :

زيادة المساحة التخزينية للمكنة الافتراضية التي تم إنشاوها متبعاً الخطوات التي تمت دراستها ؟

الفصل الثاني عشر: مقدمة عن لغة c++

تعريف البرمجة:

البرمجة هي عملية كتابة تعليمات وتوجيه أوامر لجهاز الحاسوب أو أي جهاز آخر مثل قارئات أقراص الدي في دي أو أجهزة استقبال الصوت والصورة في نظم الاتصالات الحديثة، لتوجيه هذا الجهاز وإعلامه بكيفية التعامل مع البيانات أو كيفية تنفيذ سلسلة من الأعمال المطلوبة تسمى خوارزمية.

مستويات لغات البرمجة:

يقصد بمستوى لغة البرمجة مقدار تفاصيل الرموز البرمجية التي تستخدمها هذه اللغة، فكلما قلت التفاصيل ارتفع مستوى لغة البرمجة وزادت سهولتها والقدرة على فهم تعليماتها من قبل البشر، وكلما زادات التفاصيل الدقيقة كلما انخفض مستوى اللغة واقتربت أكثر من فهم الآلة.

توجد المئات من لغات البرمجة في العالم، وتصنف لغات البرمجة بناء على درجة تجريدها إلى ثلاثة مستويات هي:

- لغات البرمجة منخفضة المستوىLow level Programming Languages
- لغات البرمجة متوسطة المستوىMedium Level Programming Language
- لغات البرمجة عالية المستوىHigh level Programming Languages

١- لغات البرمجة منخفضة المستوىLow level Programming Languages

لغات البرمجة منخفضة المستوى هي أحد مستويات لغات البرمجة المكتوبة بطريقة يسهل على أجهزة الحواسيب فهمها وتفسيرها، ويمكن من خلالها التخاطب مباشرة مع أجهزة الحاسوب، وإنجاز العمليات على مستوى وحدة المعالجة المركزية CPU والوصول المباشر إلى ذاكرة الحاسوب وإدارتها والتحكم في عمليات الإدخال والإخراج.

تعد كل من لغة الآلة Machine language التي تستخدم بيانات رقمية مكونة من ٠ و ١ فقط، ولغة التجميع Assembly language أبرز الأمثلة على لغات البرمجة منخفضة المستوى.

تتميز اللغات منخفضة المستوى بسرعة التنفيذ وأداء أفضل من اللغات عالية المستوى لأن الكود البرمجي المكتوب بها لا يحتاج إلى تفسير أو ترجمة لينفذ ويمكن للحاسوب فهمه وتنفيذه مباشرة، كما أنها تنفذ بأقل قدر ممكن من مساحة الذاكرة.

تستخدم اللغات منخفضة المستوى عادة في كتابة نواة نظام التشغيل أو برامج تعريف العتاد وتشغيله device driver أو التطبيقات المخصصة التي تتطلب التعامل المباشر مع عتاد الحاسوب، ولا تستخدم من قبل المطورين بشكل مباشر لأغراض البرمجة المعروفة مثل تصميم مواقع الويب أو تطبيقات الجوال أو تطوير الألعاب وما شابه.

- لغة الآلةMachine Language

لغة الآلة هي أدنى مستوى من مستويات لغات البرمجة وهي تكتب بتنسيق مكون من سلسلة من البتات أو من الأصفار والواحدات التي تمثل ترميزًا للعمليات التي يجب أن تتم في وحدة المعالجة المركزية وهو تنسيق يصعب على البشر قراءته وكتابته مباشرة.

يمكن تفسير لغة الآلة وتنفيذها مباشرة من قبل الحاسوب أو بعبارة أدق من قبل وحدة المعالجة المركزية للحاسوب، وفي بداية اختراع الحاسوب كانت هذه اللغة هي الوسيلة الوحيدة للاتصال بين البشر والحواسيب وكانت تُكتب على بطاقات مثقبة تمثل الأصفار والواحدات، ونظرًا لأن لغة الآلة هي لغة تعطي تعليمات مباشرة إلى المعالج فيمكنك من خلال هذه اللغة استخدام جميع الوظائف التي يمتلكها المعالج.

- لغة التجميعAssembly Language

لغة التجميع هي لغة برمجة منخفضة المستوى تم تطويرها للتقليل من صعوبة فهم لغة الآلة، فهي تعتمد في ترميزها على استخدام تعليمات وأوامر مكونة من كلمات قصيرة ومفهومة بدلًا من استخدام التعليمات الرقمية التي يصعب على البشر حفظها وكتابة البرامج مباشرة بها.

في هذه اللغة يتم ربط كل رقم بسلسلة من الأحرف لسهولة تذكره ومعرفة معناه واستخدامه في كتابة البرنامج، حيث تتضمن لغة التجميع أوامر بسيطة مثل ADDلإضافة قيمة إلى قيمة، والأمر SUBلطرح قيمة من قيمة، والأمر MOVلنقل قيمة من موضع لآخر ...إلخ. ومن خلال هذه الأوامر البسيطة يمكنك تنفيذ أوامر أكثر تقدمًا كالعمليات الحسابية والرياضية المختلفة.

٢- لغات البرمجة عالية المستوى.High level Programming Languages

تعد لغات البرمجة عالية المستوى أكثر لغات البرمجة استخدامًا وشيوعًا ومن أهم مستويات لغات البرمجة، فقد تم تصميمها لتكون مفهومة من قبل البشر وهي تحاكي اللغة الإنجليزية إلى حد ما وتستخدم صياغة واضحة وكلمات محددة ومفهومة ورموزًا رياضية شائعة مما يمكن المبرمجين من تطوير برامجهم بسرعة أكبر ويقلل احتمالية حدوث الأخطاء.

هناك المئات من لغات البرمجة عالية المستوى المتفاوتة في درجة صعوبتها لكنها بالمجمل تبقى سهلة التعلم والتنفيذ ومن أشهر الأمثلة على اللغات عالية المستوى جافاسكربت وبايثون وروبي و C++ و PHP

يعد هذا النوع من لغات البرمجة صديقًا للمبرمجين لكونها سهلة الفهم والاستخدام والصيانة، وهي تصلح لتطوير مختلف أنواع التطبيقات والبرامج كتطبيقات سطح المكتب ومواقع الويب والألعاب وتطبيقات الذكاء الاصطناعي.

٣- لغات البرمجة متوسطة المستوىMedium Level Programming Language

نظرًا للتطور الهائل الذي شهدته لغات البرمجة فقد ظهرت في الآونة الأخيرة لغات برمجة مرنة ومختصرة الصياغة وأسهل في الفهم وفي طريقة التنفيذ كلغة بايثون أو روبي ...إلخ. أتاحت هذه اللغات للمبرمجين الاستغناء عن كتابة الكثير من التفاصيل عند تطوير البرامج لذا تم تصنيفها من مستويات لغات البرمجة على أنها هي اللغات العالية المستوى وتمت إضافة تصنيف جديد للغات البرمجة هو لغات البرمجة متوسطة المستوى لتصنيف لغات البرمجة الأقدم التي ظهرت في البدايات مثل C ولغة Ada ولغة جافا بأنها لغات متوسطة المستوى.

يمكن القول أن لغات البرمجة متوسطة المستوى تجمع بين أفضل إمكانيات اللغات عالية المستوى وإمكانيات اللغات منخفضة المستوى مثل لغة التجميع، فهي تسمح بإدارة الذاكرة يدويًا ومعالجة البيانات والتعامل مع البتات والبايتات والمؤشرات مباشرة لذا تعد لغات البرمجة الأنسب للتخاطب مع التجهيزات وبرمجة الأنظمة.

لغة C++:

تعد لغة C++ من لغات البرمجة العالية الأداء والمتطورة، فقد تطويرها عن لغة C مع إضافة الصفوف والمعاملات الحسابية والأكواد البرمجية وغيرها من الإضافات التي جعلتها لغة قوية جدا ومتعددة الخصائص والاستخدامات في عديد من المجالات فلغة برمجة C++ من لغات البرمجة الكائنية المصرفة، كما أنها متعددة أنماط البرمجة ،ومن اللغات الأكثر شيوعا واستخدامه حاليا.

مميزاتها :

تمتاز لغة برمجة السي بلس بلس بأنها تجمع بين خصائص اللغات العالية المستوى وكذلك اللغات منخفضة المستوى، فلغة برمجة C++ قريبة من لغة C عالية المستوى، وأيضا قريبة من لغة التجميع والتي يعد مستواها منخفضا.

تمتاز لغة C++ بالتعدد الكائني فقد تم تطوير التعامل مع برامج الإجراءات إلى بناء البرنامج بشكل يعتمد على الكائنات بواسطة لغة السي بلس بلس حيث يوجد لكل كائن أدواته الخاصة التي يمكن العمل عليها.

١. لغة C++ من اللغات القوية جدا التي تمتاز بالسرعة والأداء العالي مما يجعلها تستخدم في إنشاء البرامج المعقدة بالإضافة لتطوير تطبيقات سطح المكتب و. web browsers
٢. تستطيع ربط لغة C++ بقواعد البيانات والعمل ضمن إطار ذلك .
٣. لغة C++ هي لغة مرنة فيمكنك كتابة التعبيرات الرياضية باستخدام متغيرات من أعداد حقيقية وأعداد صحيحة بسهولة تامة.

استخداماتها :

١- تطوير أنظمة التشغيل فنواة نظام التشغيل يونكس UNIX ولينكس Linux وويندوز Windows ونواة أنظمة Apple OS X كلها مكتوبة بلغة سيC
٢- برمجة التطبيقات المساعدة لنظام التشغيل فمعظم متصفحات الويب على سبيل المثال مبرمجة بلغة Cكما أن معظم إضافات المتصفحات مكتوبة بها.
٣- برمجة التطبيقات المضمنة وتطبيقات انترنت الأشياء IoTوالأنظمة المدمجة وبرمجة المتحكمات الدقيقة المستخدمة في برمجة الأجهزة الذكية كالسيارات ذاتية القيادة أو الروبوتات الذكية لما توفره من تحكم مباشر بالعتاد.
٤- بناء قواعد البيانات فالعديد من أنظمة إدارة قواعد البيانات المعروفة مثل أوراكل Oracle و MySQL و PostgreSQL تستخدم لغة.C
٥- تعد لغة C كذلك لغة مثالية لبرمجة تطبيقات سطح المكتب فعالة وعالية الأداء، فهي تمكن المطور من الوصول بسهولة لموارد الحاسوب كالمعالج والذاكرة والتحكم في أداء التطبيق بمرونة كبيرة.
٦- برمجة الالعاب الألكترونية وتطوير محركات الألعاب Game Engines ومن أشهر الألعاب المطورة باستخدام لغة سي C لعبة Quake وDoom
٧- تطوير خوادم الويب الكبيرة التي تحتاج إلى التعامل مع عدد كبير من الطلبات في وقت واحد ومعالجة البيانات بسرعة مثل خادم أباتشي Apacheأو إنجن إكس Nginx
٨- وبالعموم تصلح لغة سي بشكل عام لتطوير كافة التطبيقات عالية الأداء التي تحتاج لإنجاز الكثير من العمليات أو تتطلب معالجة كمية ضخمة من البيانات بسرعة.

تعريف برنامج Visual Studio code ومميزاته

يعتبر **visual studio code**من البرامج المجانية والمفتوحة المصدر، والتي يمكن تنزيلها على مختلف أنظمة التشغيل بحجم لا يتجاوز ١٠٠ .MBكما يمكن أن تجده باسم **VSCode**وهو بيئة عمل و محرر أكواد **(IDE AND Code Editor)**تم تصميمه بوساطة شركة **Microsoft.**

كما تم إصدار النسخة الأولى منه في ٢٩ أبريل من عام ٢٠١٥، وقد تم ترخيصه بوساطة شركة **MIT.** وهي شركة تراخيص برمجية عالمية. إضافة لذلك تم وضع الكود الخاص بالبرنامج على موقع **GitHub** ويعتبر الإصدار الأكثر استقراراً هو **1.65.2**الذي تم إطلاقه في ١٠ مارس ٢٠٢٠.

على الرغم من حجمه الصغير، إلا أنه يعتبر من أهم محررات النصوص البرمجية المستخدمة على الإطلاق. وقد ظهر هذا البرنامج بعد العديد من نسخ محررات الأكواد البرمجية، التي أثبتت عدم قدرتها على مواكبة لغات البرمجة المختلفة والحديثة مثل VI Editor, Sublime Text Editor.

كما ويقدم هذا البرنامج القدرة على توفير القواعد اللازمة لكتابة مختلف لغات البرمجة الحديثة , من خلال تنزيل وتضمين المكاتب والحزم المتعلقة باللغة المطلوبة. وبمجرد تنزيلها فإنك ستجد هذا البرنامج يعالج الكود المكتوب داخله بما يوافق قواعد اللغة التي تم تنزيلها .**(syntax)**

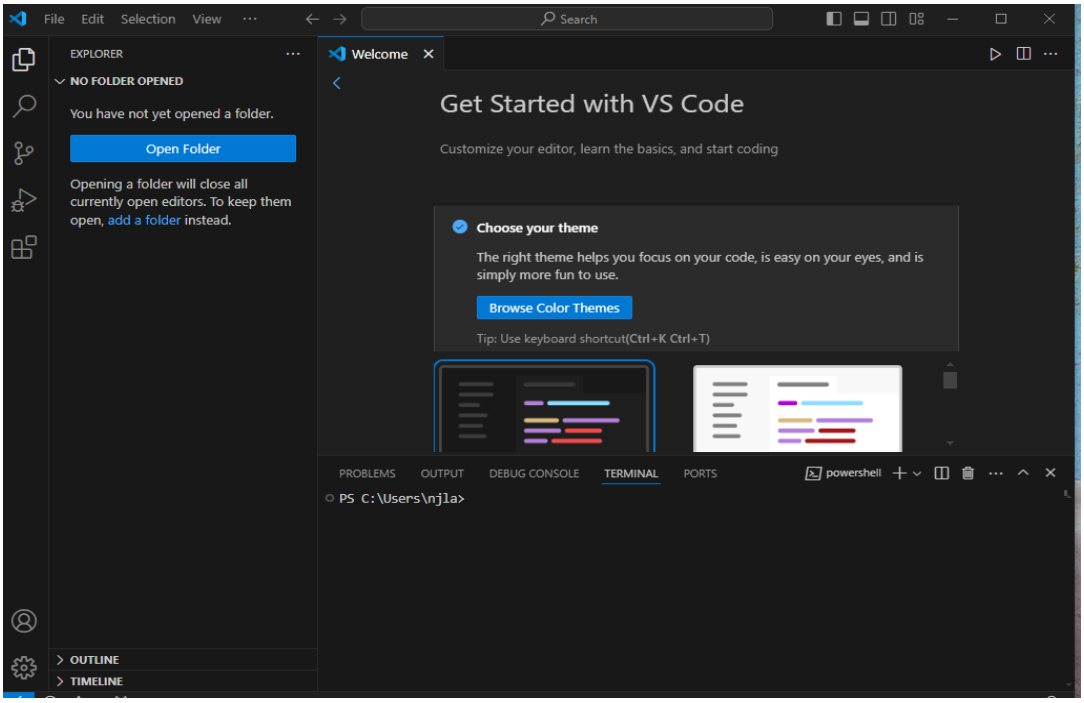
مميزاته :

- حجم صغير.
- خالي من التكاليف اللاحقة المترتبة على تنزيل الحزم.
- التسلسل الهرمي لملفاته.
- دعم التعامل مع قواعد البيانات من خلاله.
- إضافة التعليقات.
- الحفظ التلقائي للتعليمات والأسطر البرمجية.
- سهل الاستخدام بالنسبة للمبتدئين في البرمجة.

تشغيل برنامج Visual Studio C++:

من مربع البحث ← نقوم بكتابة Visual Studio code ← ستظهر لنا الواجهة كما في الشكل (١٢,١).

ومن ثم الانتقال الى File ← New file ← ثم نضع اسم الملف على سبيل المثال text.cpp لابد من كتابة الامتداد cpp حتى يتم التعرف على الملف بانه يدعم لغة c++



تمرين:

- أمام كل عبارة اجيب ما اذا كانت العبارة صحيحة ام خاطئة:

م	العبارة	العلامة
١	من مميزات برنامج Visual Studio code انه صغير الحجم	صح
٢	لغة C++ من اللغات القوية جدا التي تمتاز بالسرعة والأداء العالي	صح
٣	لغات منخفضة المستوى تعد من النوع من لغات البرمجة صديقًا للمبرمجين لكونها سهلة الفهم والاستخدام والصيانة	خطا
٤	تستخدم لغة C++ في بناء قواعد بيانات	صح
٥	تعد كل من لغة الآلة التي تستخدم بيانات رقمية مكونة من ٠ و ١ فقط	صح

- اختار الأجابة الصحيحة:

٦- يمكن القول أن لغات البرمجة ..... المستوى تجمع بين أفضل إمكانيات اللغات عالية المستوى وإمكانيات اللغات منخفضة المستوى مثل لغة التجميع

أ- منخفضة
ج- متوسطة
ب- عالية
د- الجيل الرابع

٧- لغة ..... هي أدنى مستوى من مستويات لغات البرمجة وهي تكتب بتنسيق مكون من سلسلة من البتات

أ- PHP
ج- التجميع
ب- C++
د- الالة

٨- ..... هي عملية كتابة تعليمات وتوجيه أوامر لجهاز الحاسوب أو أي جهاز آخر

أ- البرمجة
ج- التجميع
ب- C++
د- الالة

٩- عدد مستويات لغات البرمجة هي :

أ- مستويان
ج- خمس مستويات
ب-ثلاث مستويات
د- مستوى واحد فقط

١٠- لغة برمجة ..... من لغات البرمجة الكائنية المصرفة

أ- PHP
ج- التجميع
ب- الالة
د- C++

الفصل الثالث عشر: التعرف على أساسيات لغة c++

الهيكلة العامة للبرنامج:

الكود بلغة C++

```
منطقة التعريفات العامة واستدعاء مكاتب للبرنامج
# include<iostream.h>
الدالة التي تكتب بداخلها اكواد البرنامج
Main()
{
ساحة الاكواد أو منطقة كتابة اكواد البرنامج والقراءة والطباعة
}
```

الشكل (١٣,١)

منطقة التعريفات العامة واستدعاء مكاتب للبرنامج:

في هذا المكان يتم كتابة جميع المكاتب التي سنحتاج إليها داخل البرنامج وكذالك المتغيرات التي تعرف بشكل عام لكل البرنامج والسجلات والدوال على سبيل المثال دالة القراءة (printf, scanf)في لغة C تقع ضمن المكتبة <h.iostream>

- دالة (main) :اول دالة تنفذ عنما يعمل البرنامج لذالك بداية البرنامج تكون من هذه الدالة.
- ساحة كتبة الأكواد: هي المنطقة التي يتم بداخلها كتابة الأكواد البرمجية وتعريفات وغيرها
- يجب وضع فاصلة منقوطة في نهاية أي تعبير مبرمج من قبل المستخدم للدلالة على أن التعبير انتهى.

لماذا أجد الأمر using namespace std; موضوع ضمن كود المشروع إذا لم يكن وجوده إجبارياً؟

في البداية هذا السطر معناه أنك تريد تضمين الأشياء الموجودة بداخل المكتبة std بشكل مباشر في المشروع.
في حال أردت إزالة هذا السطر ستكون مجبر على تحديد إسم المكتبة التي تستخدم منها الأمر cout و cin و غيرها من الأوامر التي سنتعلمها في دروس لاحقة.

التعليقات

نستخدم التعليقات لنضع ملاحظات حول الكود الذي كتبناه فقط, لكي لا ننسى كيف برمجنا الكود في حال أردنا مراجعته أو التعديل عليه بعد وقت طويل.
التعليقات لا تؤثر إطلاقاً على الكود المكتوب, و في لغة C++ يوجد أسلوبين لوضع التعليقات:

- تعليق على سطر واحد.
- تعليق على عدة أسطر.

أمثلة

في حال كان التعليق يتألف من سطر واحد, نضع الرمز // قبل كتابة التعليق.

// this is a one line comment تعليق سطر واحد

في حال كان التعليق يتألف من عدة أسطر, نبدأ التعليق بالرمز /* و ننهيه بالرمز */.

```
/*
This is multlines comment
هذا تعليق
يتألف من
عدة سطور
*/
```

تذكر: أنت لست مجبراً على وضع تعليقات في برامجك. و لكننا ننصحك بوضع تعليقات دائماً حتى تساعدك في فهم الكود الذي كتبته.

المتغيرات:

عبارة عن أماكن يتم حجزها في الذاكرة بهدف تخزين بيانات فيها أثناء تشغيل البرنامج.
النوع الذي نعطيه للمتغير يجعل نظام التشغيل يحدد نوع البيانات الذي يمكن تخزينه في المساحة المحجوزة لهذا المتغير في الذاكرة.
على سبيل المثال, إذا أردت تخزين رقم في متغير, يجب أن يكون نوع المتغير في الأساس مخصص لتخزين الأرقام حتى يسمح لك بذلك.
وإذا أردت تخزين نص في المتغير يجب أن يكون نوع المتغير في الأساس مخصص لتخزين النصوص وهكذا.

شروط تسمية المتغيرات:

١- لا يمكن أن يبدأ برقم
٢- لا يمكن ان يحتوي على مسافة
٣- لا يحتوي على أي رمز خاص مثل * أو & وغيرها...ماعدا الخط التحتي " _ "
٤- لا يمكن أن يكون احدى الكلمات المحجوزة مثل (DO, main)
٥- أن يكون الاسم مكتوبا من سلسلة متصلة من الحروف أو الارقام بشرط أن يبدا بحرف او خط تحتي "_"

بعض الأمثلة الصحيحة على اسماء المتغيرات:

- B6
- X_ray
- Matrix
- Ok_
- A
- Soft_fine
- Door12
- _new

وبعض الامثلة على أسماء المتغيرات الغير مقبولة للأسباب المبينة:

7-up	لانه بدأ برقم وليس حرف
B6.1	لاستعمالة الرمز الخاص (.)
Salim!	لاستعمالة الرمز الخاص (!)
No#1	لاستعمالة علامة (#)
Ъ2	لا يجوز استعمال حروف غير إنجليزية

ومن الجدير بالذكر ، أن لغة C++ تفرق بين الحروف الأبجدية الصغيرة والكبيرة ، فمثلا الأسماء SYSTEM , system ,System تعامل كأسماء مختلفة عن بعضها البعض بسبب اختلاف معاملة المترجم للحروف الصغيرة والكبيرة.

الكلمات المحجوزة

وهي كلمات قياسية معروفة مسبقا لمترجم C++ ، وتكتب عادة بحروف صغيرة ، ولها معان خاصة بها تؤديها في برنامج C++ ، وهذه الكلمات المحجوزة حسب الترتيب الأبجدي هي:

near	Static	asm	Double	long	Sizeof
do	int	While	new	auto	else
For	This	Void	Delete	Goto	if
const	Entry	char	Class	Public	Case
Continue	Extern	struct	inline	float	Private
Virtual	Volatile	Frinde	enum	near	Static
cdecl	Default	inline	Overload	Unsigned	Typedef
Signed	Pascal	Operator	Switch	Template	Union
Register	Protected	far	Catch	char	Const
				break	Return

وينبغي التنبيه إلى أن هذه الكلمات المحجوزة ، لا يجوز إعادة تعريفها أو استعمالها لغير ما خصصت له, وكما تلاحظ من قائمة الكلمات المحجوزة ، أن لغة C++ تعد لغة صغيرة إذ تتكون من عدد قليل من الكلمات المحجوزة تقريبا ٥٢ كلمة محجوزة فقط.

أنواع المتغيرات:

تعريف المتغير	استخدامه
Int var;	يستخدم لتعريف المتغير من نوع integer أي رقمي مثلا (**int x=5**)
Float var;	يستخدم لتعريف المتغير من نوع كسري مثلا (Float var=5.4;)
Char var;	يستخدم لتعريف المتغير من نوع حرفي مثلا (Char var='a';)
Double var;	يستخدم لتعريف المتغير من نوع Double أي حقيقي
Void var;	يستخدم لتعريف المتغير من نوع Void أي لا يرجع أي قيمة

اسناد قيم للمتغيرات:

تكون طريقة إسناد إي قيمة أو تعبير رياضي إلى متغير بشكل التالي

متغير أو تعبير رياضي=اسم المتغير

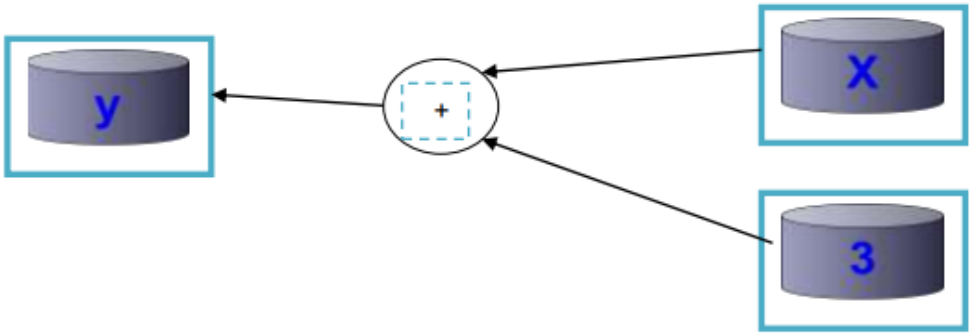
- في الطرف الأيمن من المساواة يوجد فقط اسم المتغير الذي نريد إسناد قيم إليه
- في الطرف الأيسر من المساواة نستطيع كتابة إي تعبير رياضي أو متغير أو قيمة معينة

مثال : إسناد قيمة تعبير رياضي إلى متغير (بطريقة صحيحة)

الكود

```
1.int y, x=5;
2.y=x+3;
```

- في الخطوة رقم (2) أسندنا القيمة الناتجة من تعبير رياضي وهو (x+3) إلى المتغير (y)



- لو تلاحظ أن الطرف الأيمن مكون من متغير فقط والطرف الأيسر مكون من تعبير رياضي

- يمكن أن يسند المتغير بقيمة مباشرة إثناء وقت التعريف هكذا

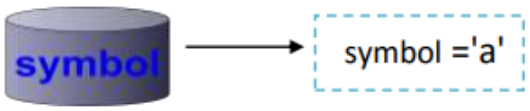
الكود

```
int x=5;
```

- قد يسند المتغير بقيمة معينة بعد التعريف في أي مكان في البرنامج.

الكود

```
char symbol;
symbol ='a';
```



- المتغير(symbol) تم تخزين الحرف a في داخله.

- الحروف عندما تخزن تضع بين عالمة تنصيصية واحد داللة على انها حرف واحد فقط

- قد نسند قيمة متغير إلى متغير أخر

الكو د
```
int first ,second;
first =51;
second= first ;
```

- أصبح المتغير second يحوي نفس قيمة المتغير first في الخطوة رقم (٣)
- أو قد يسند المتغير بقيمة في وقت اإلدخال من لوحة المفاتيح (سنتطرق عليه لاحقا)

مثال:

```
#include <iostream>
using namespace std;
int main()
{
   int a = 10;          // هنا قمنا بتعريف متغير إسمه a واسند له قيمته تساوي ١٠
   cout << "a = " << a;    // هنا قمنا بطباعة قيمة المتغير a
     return 0;
}
```

سنحصل على النتيجة التالية عند التشغيل.

```
a = 10
```

أساليب تعريف المتغيرات في C++

في البداية أنواع المتغيرات هي نفسها أنواع البيانات الأولية

char - bool - double - float - int

الآن, من خلال الأمثلة سنتعرف على عدة أساليب يمكننا إتباعها لتعريف المتغيرات و تصغير حجم الكود.

المثال الأول

في حال كنا نريد تعريف عدة متغيرات من نفس النوع و بدون إعطائهم قيم أولية.

نستطيع تعريفهم هكذا.

```
int a;

int b;

int c;
```

و نستطيع إختصار الكود كالتالي.

```
int a, b, c;
```

المثال الثاني

في حال كنا نريد تعريف عدة متغيرات من نفس النوع و إعطائهم قيم مباشرةً عند إنشائهم.

نستطيع تعريفهم هكذا.

```
int a = 10;

int b = 20;

int c = 30;
```

و نستطيع إختصار الكود كالتالي.

```
int a=10, b=20, c=30;
```

المتغيرات الثابتة:

الثابت (**Constant**)عبارة عن أي شيء يتم تعريفه بشكل لا يمكن إعادة تغيير قيمته.
لتعريف المتغير أو الكائن كثابت نستخدم الكلمة **const** و عندها نصبح غير قادرين على تغيير قيمته.

```
const float pi=3.14;
```

أمر الاخراج Cout في C++

هي دوال تستخدم لعرض نواتج العمليات التي نريدها أمام المستخدم في شاشة التنفيذ والدوال.

مبادئ الطباعة الأساسية

عليك مراعاة المبادئ التالية عند استخدام الأمر cout.

- لعرض رقم, ضعه كما هو.
- لعرض قيمة متغير, ضعه كما هو.
- لعرض حرف يجب وضعه بين ' '.
- لعرض كلمة أو نص, يجب وضعه بين " ".

مثال

```
#include <iostream>
using namespace std;
int main()
{
   cout << "C++";
   return 0;
}
```

سنحصل على النتيجة التالية عند التشغيل.

```
C++
```

- طباعة عدة أشياء دفعة واحدة بواسطة الأمر cout في C++

لدمج عدة أشياء مختلفة و عرضها باستخدام الأمر cout نضع الرمز << بين كل شيئين نريد طباعتهما.

- في المثال التالي, قمنا بإنشاء برنامج يحتوي على متغيرين name و year.
بعدها قمنا بعرضهما كجملة واحدة.

مثال

```
#include <iostream>

using namespace std;
int main()

{

    char name = "Ali";

    int year   = 1996;

    cout << "Hello my name is " << name << ", I born in " << year;

    return 0;

}
```

سنحصل على النتيجة التالية عند التشغيل.

```
Hello my name is Ali, I born in 1996
```

عرض المحتوى على سطر جديد C++

عند عرض المحتوى بواسطة الأمر cout يمكنك تحديد نهاية السطر من خلال وضع العامل endl و عندها سيظهر أي شيء تطبعه بعده على سطر جديد.

معلومة: endl إختصار لجملة End Line.

مثال

```
#include <iostream>

using namespace std;
int main()

{

    cout << "Hello" << endl << "C++ course";

    return 0;

}
```

سنحصل على النتيجة التالية عند التشغيل.

```
Hello
C++ course
```

النزول على سطر جديد بواسطة الرمز \n في C++

إذا أردت النزول على سطر جديد بدون الحاجة لكتابة endl << فيمكنك تمرير \n كنص.

ملاحظة: أغلب المبرمجين يفضلون تمرير \n بدل كتابة endl << لأنها مختصرة أكثر.

معلومة: \n إختصار لجملة New Line.

في المثال التالي قمنا بتمرير \n كنص للأمر cout من أجل النزول على سطر جديد.

مثال

```
#include <iostream>
using namespace std;
int main() {
    cout << "Hello \nC++ course";
    return 0;
}
```

سنحصل على النتيجة التالية عند التشغيل.

```
Hello
C++ course
```

هنا قمنا بإعادة المثال السابق تماماً مع كتابة أمرين cout هذه المرة.

ملاحظة: هذا الأسلوب أسهل من ناحية القراءة و التعديل و لهذا ننصح باعتماده بدلاً من الأسلوب السابق.

مثال

```
#include <iostream>
using namespace std;
int main() {
    cout << "hello\n";
    cout << "C++ course";
    return 0; }
```

سنحصل على النتيجة التالية عند التشغيل.

```
Hello
C++ course
```

وضع عدة مسافات فارغة بواسطة الرمز \t في C++

إذا أردت وضع عدة مسافات فارغة دفعة واحدة و بشكل مرتب فالأفضل أن تستخدم الرمز \t بدل وضع المسافات بنفسك.

من الأشياء التي قد تجعلك تستخدم \t هي أنه عند طباعة عدة أسطر تجده يحاول عرضهم لك بشكل متناسق لإظهار المحتوى بشكل أسهل للقراءة.

معلومة: \t إختصار لجملة Tab Space

في المثال التالي قمنا بوضع الرمز \t بين الكلمات من أجل وضع مسافة تلقائية بينهم بشكل مرتب.

مثال

```
#include <iostream>

using namespace std;

int main() {

    cout << "Name:\tAli Rabeei\n";

    cout << "Job:\tProgrammer";

    return 0; }
```

سنحصل على النتيجة التالية عند التشغيل.

```
Name:   Ali Rabeei
Job:    Programmer
```

أمر الإدخال cin في C++

لجعل المستخدم قادر على إدخال بيانات في البرنامج أثناء اشتغاله نستخدم أمر الإدخال cin >>.

في كل مرة تقوم فيها باستدعاء هذا الأمر يقوم المترجم بانتظارك لإدخال عدد أو كلمة واحدة من لوحة المفاتيح (Keyboard).

بعد الإنتهاء من الإدخال و النقر على الزر Enter سيتم إرجاع الشيء الذي قمت بإدخاله في المكان الذي تم منه إستدعاء الأمر cin >>.

مثال

```
#include <iostream>
using namespace std;
int main() {
    // قمنا بتجهيز المتغيرين name و age لتخزين المعلومات التي سيدخلها المستخدم لاحقاً
    string name;
    int age;
    // هنا سيطلب من المستخدم إدخال إسمه و من ثم سيتم تخزين الإسم الذي يدخله في المتغير name
    cout << "What's your name? ";
    cin >> name;
    // هنا سيطلب من المستخدم إدخال عمره و من ثم سيتم تخزين الرقم الذي يدخله في المتغير age
    cout << "How old are you? ";
    cin >> age;
    // في الأخير قمنا بعرض المعلومات التي يدخلها المستخدم بشكل مرتب
    cout << "Your name is " << name << ", and your age is " << age << " years.";
    return 0; }
```

سنحصل على النتيجة التالية عند التشغيل مع الإشارة إلى أننا قمنا بتعليم البيانات التي إنتظرنا البرنامج لإدخالها من لوحة المفاتيح باللون الأصفر.

```
What's your name? Ali
How old are you? 25

Your name is Ali, and your age is 25 years.
```

الفرق بين الأمر cinو الأمر cout

في البداية معرفة سبب تسمية الأوامر بهذه الكلمات سيسهل عليك تذكر الفرق بينهما, لذا تذكر دائماً أصلهما:

- cinيقصد بها كلمة See Inو التي بدورها تستخدم لإدخال قيمة في المتغير.
- coutيقصد بها كلمة See Outو التي بدورها تستخدم للحصول على قيمة المتغير.

الأمر cin نضع بعده إسم المتغير الذي سنخزن القيمة التي يدخلها المستخدم فيها و تكون الأسهم باتجاه اليمين كالتالي.

```
// x قم بتخزين ما سيدخله المستخدم في المتغير
cin >> x;
```

الأمر **cout** نضع بعده أي شيء نريد عرضه أو أي متغير نريد عرض قيمته و تكون الأسهم باتجاه اليسار كالتالي.

```
// x قم بعرض القيمة المدخلة في المتغير
cout << "x = " << x;
```

المشاكل التي يسببها إدخال مسافات فارغة

تذكر دائماً أن الأمر **cin >>** يستخدم لإدخال رقم أو كلمة واحدة فقط بمعنى أنه لا يجب أن يحتوي ما تدخله على مسافات فارغة.
في حال كان ما أدخلته يحتوي على مسافات فارغة, سيسبب ذلك مشاكل في الكود لأن المترجم سيعتبر أنك تحاول إدخال مجموعة قيم و ليس قيمة واحدة.

عندما يجدك المترجم قمت بإدخال مجموعة قيم في ذات الوقت الذي تحاول فيه أن تخزن قيمة واحدة فقط في متغير, سيعتبر أنك تريد إدخال القيم الأخرى في المرات القادمة التي تطلب فيها من المستخدم أن يدخل قيم.

في المثال التالي, قمنا بإنشاء برنامج يطلب من المستخدم إدخال إسمه و بريده الإلكتروني, بعدها يعرض له ما قام بإدخاله.

عند تشغيل البرنامج سنتعمد إدخال إسم المستخدم الكامل (إسمه و إسم عائلته) مع وضع مسافة فارغة بينهما حتى ترى الخطأ الذي سيحدث.

مثال

```
#include <iostream>
using namespace std;
int main() {
    // قمنا بتجهيز المتغيرين name و email لتخزين المعلومات التي سيدخلها المستخدم لاحقاً
    string name;
    string email;
    // name هنا سيطلب من المستخدم إدخال إسمه و من ثم سيتم تخزين الإسم الذي يدخله في المتغير
    cout << "Enter your name: ";
    cin >> name;
    // email هنا سيطلب من المستخدم إدخال بريده الإلكتروني و من ثم سيتم تخزين البريد الذي يدخله في المتغير
    cout << "Enter your email: ";
    cin >> email;
    // في الأخير قمنا بعرض المعلومات التي يدخلها المستخدم بشكل مرتب
    cout << "\n\n----------";
    cout << "\nName = " << name;
    cout << "\nEmail = " << email;
    return 0; }
```

سنحصل على النتيجة التالية عند التشغيل مع الإشارة إلى أننا قمنا بتعليم البيانات التي إنتظرنا البرنامج لإدخالها من لوحة المفاتيح باللون الأصفر.

```
Enter your name: ali ahmed
Enter your email:                <- لاحظ أنه لم ينتظرنا لندخل البريد الإلكتروني بل إعتبر أنه تم إدخاله

----------
Name = Ali                <- لاحظ أنه قام بوضع Ali فقط في المتغير name
Email = ahmed             <- لاحظ أنه قام بوضع ahmed في المتغير email بشكل تلقائي
```

تمثيل العمليات الرياضية:

تمثل العمليات الرياضية برمجيا بطريقة مشابه لطريقة تمثيلها رياضيا مع تغير طفيف بالرموز الرياضية لما يكافئها من الرموز البرمجية والحظ هذا الجدول التوضيحي للعمليات وتمثيلها رياضيا وبرمجيا

لعوامل التي تستخدم في العمليات الحسابية (Arithmetic Operators)

إسم العامل	رمزه	مثال	شرح الكود
Assignment	=	a = b	أعطي a قيمة b
Addition	+	a + b	أضف قيمة b على قيمة a
Subtraction	-	a - b	إطرح قيمة b من قيمة a
Multiplication	*	a * b	أضرب قيمة a بقيمة b
Division	/	a / b	أقسم قيمة a على قيمة b
Modulo	%	a % b	للحصول على آخر رقم يبقى عندما نقسم قيمة a على قيمة b
Increment	++	a++	لإضافة 1 على قيمة a و تستخدم في الحلقات
Decrement	--	a--	لإنقاص 1 من قيمة a و تستخدم في الحلقات

العامل = (Assignment Operator)

يستخدم لإعطاء قيمة للمتغير.العامل=

مثال

```
#include <iostream>
using namespace std;
int main()
{
    int a = 5;     // 5 القيمة a أعطينا المتغير
    int b = a;     // a قيمة المتغير b أعطينا المتغير
    cout << "a = " << a << endl;
    cout << "b = " << b;
    return 0;
}
```

سنحصل على النتيجة التالية عند التشغيل.

```
a = 5
b = 5
```

العامل + (Addition Operator)

العامل **+** يستخدم لإضافة قيمة على قيمة, أي في عمليات الجمع.

مثال

```
#include <iostream>
using namespace std;
int main() {
    int a = 3;
    int b = 4;
    int c = a + b;   // c = 3 + 4 = 7
    cout << "c = " << c;
    return 0; }
```

سنحصل على النتيجة التالية عند التشغيل.

```
c = 7
```

العامل - (Subtraction Operator)

العامل - يستخدم لإنقاص قيمة من قيمة, أي في عمليات الطرح.

مثال

```
#include <iostream>
using namespace std;
int main() {
    int a = 3;
    int b = 4;
int c = a - b;   // c = 3 - 4 = -1
    cout << "c = " << c;
    return 0; }
```

سنحصل على النتيجة التالية عند التشغيل.

```
c = -1
```

العامل * (Multiplication Operator)

العامل * يستخدم لضرب قيمة بقيمة, أي في عمليات الضرب.

مثال

```
#include <iostream>
using namespace std;
int main() {
    int a = 6;
    int b = 5;
    int c = a * b;     // c = 6 * 5 = 30
    cout << "c = " << c;
    return 0;
}
```

سنحصل على النتيجة التالية عند التشغيل.

```
c = 30
```

العامل / (Division Operator)

العامل **/** يستخدم لقسمة قيمة على قيمة, أي في عمليات القسمة.

مثال

```
#include <iostream>

using namespace std;

int main() {

    int a = 8;

    int b = 5;

    int c = a / b;     // c = 8 / 5 = 1

    cout << "c = " << c;

    return 0; }
```

سنحصل على النتيجة التالية عند التشغيل.

```
c = 1
```

ملاحظة: سبب عدم ظهور أي أرقام بعد الفاصلة هو أننا عرفنا المتغيرات كأعداد صحيحة **int**.

العامل % (Modulo Operator)

العامل % يقال له الـ Modulo و يسمى Remainder في الرياضيات و هو آخر رقم يبقى من عملية القسمة.

إذاً نستخدم الـ Modulo للحصول على آخر رقم يبقى من عملية القسمة.

و له فوائد كثيرة, فمثلاً يمكننا إستخدامه لمعرفة ما إذا كان الرقم مفرد أو مزدوج (أي Even or Odd) و هذا شرحناه بتفصيل في مادة الخوارزميات.

في هذا المثال سنقوم بتخزين الرقم الذي يبقى من القسمة في المتغير c.

مثال

```
#include <iostream>
using namespace std;
int main() {
    int a = 8;
    int b = 5;
    int c = a % b;     // c = 8 % 5 = 3
    cout << "c = " << c;
    return 0; }
```

سنحصل على النتيجة التالية عند التشغيل.

```
c = 3
```

العامل ++ (Increment Operator)

العامل **++** يستخدم لزيادة قيمة المتغير واحداً, و هذا الأسلوب يستخدم كثيراً في الحلقات لزيادة قيمة العداد واحداً في كل دورة بكود أقل.

مثال

```
#include <iostream>
using namespace std;
int main() {
    int a = 5;
    a++;     // a = 5 + 1 = 6
    cout << "a = " << a;
    return 0; }
```

سنحصل على النتيجة التالية عند التشغيل.

```
a = 6
```

العامل (Decrement Operator) --

العامل -- يستخدم لإنقاص قيمة المتغير واحداً, و هذا الأسلوب يستخدم كثيراً في الحلقات لإنقاص قيمة العداد واحداً في كل دورة بكود أقل.

مثال

```
#include <iostream>
using namespace std;
int main() {
    int a = 5;
    a--;     // a = 5 - 1 = 4
    cout << "a = " << a;
    return 0; }
```

سنحصل على النتيجة التالية عند التشغيل.

```
a = 4
```

تمرين عملي:

اكتب برنامجا لإدخال عدد ما وليكن العدد ٧ ومن ثم طباعة جدول الضرب له؟

الفصل الرابع عشر: الجمل الشرطية والعبارات الشرطية
(if – Switch)

الجمل الشرطية والعبارات الشرطية(Structures Control)

الشروط (**conditions**) تستخدم لتحديد طريقة عمل البرنامج نسبةً للمتغيرات التي تطرأ على الكود.

كمثال بسيط, يمكنك بناء برنامج لمشاهدة الأفلام, عند الدخول إليه يطلب من المستخدم في البداية أن يدخل عمره لكي يقوم بعرض أفلام تناسب عمره.

يمكنك وضع العدد الذي تريده من الشروط في البرنامج الواحد, و تستطيع وضع الشروط بداخل بعضها البعض أيضاً.

أنواع جمل الشرط

إسم الجملة	دواعي الإستخدام
if - else - else if statements	نستخدمهم إذا كنا نريد تنفيذ كود معين في حال تحقق الشرط أو مجموعة من الشروط التي وضعناها.
switch statement	نستخدمها إذا كنا نريد إختبار قيمة متغير معين مع لائحة من الإحتمالات نقوم نحن بوضعها, و إذا تساوت هذه القيمة مع أي إحتمال وضعناه ستتنفذ الأوامر التي وضعناها في هذا الإحتمال فقط.

جمل الشرط بشكل عام

الشكل العام لوضع الشروط هو التالي.

```
if ( condition )
{
   // إذا كان الشرط صحيحاً نفذ هذا الكود
}

else if ( condition )
{
   // إذا كان الشرط صحيحاً نفذ هذا الكود
}

else
{
   // نفذ هذا الكود في حال لم يتم التعرف على الكود في أي شرط
}
```

لست بحاجة إلى إستخدام الجمل الثلاثة في كل شرط تضعه في البرنامج, و لكنك مجبر على إستخدام جملة الشرط **if** مع أي شرط.

جملة الشرط if

if في اللغة العربية تعني " إذا ", و هي تستخدم فقط في حال كنت تريد تنفيذ كود معين حسب شرط معين.

المثال الأول

إذا كانت قيمة المتغير S أكبر من 5 سيتم طباعة الجملة: S is bigger than 5.

```
#include <iostream>

using namespace std;

int main() {

    int S = 0;

    if( S > 5 ) {

        cout << "S is bigger than 5";    }

    return 0; }
```

سنحصل على النتيجة التالية عند التشغيل.

فارغة

هنا سأل نفسه التالي: هل قيمة المتغير S أكبر من 5؟
فكان جواب الشرط كلا (false) , لذلك لم ينفذ أمر الطباعة الموجود في جملة الشرط.

المثال الثاني

إذا كانت قيمة المتغير S أكبر من 5 سيتم طباعة الجملة: S is bigger than 5.

```
#include <iostream>

using namespace std;

int main() {

    int S = 30;

    if( S > 5 )    {

        cout << "S is bigger than 5";    }

    return 0; }
```

سنحصل على النتيجة التالية عند التشغيل.

```
S is bigger than 5
```

هنا سأل نفسه التالي: هل قيمة المتغير **S** أكبر من **5**؟
فكان جواب الشرط نعم (**true**), لذلك نفذ أمر الطباعة الموجود في جملة الشرط.

جملة الشرط else

else في اللغة العربية تعني " أي شيء آخر ", و هي تستخدم فقط في حال كنا نريد تنفيذ كود معين في حال كانت نتيجة جميع الشروط التي قبلها تساوي false.

يجب وضعها دائماً في الأخير, لأنها تستخدم في حال لم يتم تنفيذ أي جملة شرطية قبلها.

إذا نفذ البرنامج الجملة if أو else if فإنه سيتجاهل الجملة else.
و إذا لم ينفذ أي جملة من الجمل if و else if فإنه سينفذ الجملة else.

المثال الأول

إذا كانت قيمة المتغير S تساوي 5 سيتم طباعة الجملة: S is equal 5.
إذا كانت قيمة المتغير S لا تساوي 5 سيتم طباعة الجملة: S is not equal 5.

```
#include <iostream>

using namespace std;

int main() {
    int S = 5;
    if( S == 5 )  {
        cout << "S is equal 5";  }
    else  {
        cout << "S is not equal 5";  }
    return 0; }
```

سنحصل على النتيجة التالية عند التشغيل.

```
S is equal 5
```

هنا سأل نفسه التالي: هل قيمة المتغير S تساوي 5؟
فكان جواب الشرط نعم (true) , لذلك نفذ أمر الطباعة الموجود في الجملة if.

جملة الشرط else if

جملة else if تستخدم إذا كنت تريد وضع أكثر من إحتمال (أي أكثر من شرط).
جملة أو جمل الـ else if يوضعون في الوسط, أي بين الجملتين if و else.

مثال

إذا كانت قيمة المتغير number تساوي 1 سيتم طباعة الكلمة: one.
•إذا كانت قيمة المتغير number تساوي 2 سيتم طباعة الكلمة: two.
•إذا كانت قيمة المتغير number تساوي 3 سيتم طباعة الكلمة: three.
•إذا كانت قيمة المتغير number أكبر أو تساوي 4 سيتم طباعة الجملة: four or greater.
•إذا كانت قيمة المتغير number أصغر من 0 سيتم طباعة الجملة: negative number.

```
#include <iostream>
using namespace std;
int main() {
   int number = 3;
   if( number == 1 ) {
      cout << "one";  }
   else if( number == 2 )  {
      cout << "two";  }
   else if( number == 3 )   {
      cout << "three";   }
   else if( number >= 4 )  {
      cout << "four or greater";    }
   else  {
      cout << "negative number";   }
   return 0; }
```

سنحصل على النتيجة التالية عند التشغيل.

```
three
```

هنا سأل نفسه التالي: هل قيمة المتغير number تساوي 1؟
فكان جواب الشرط كلا (false), فانتقل إلى الشرط الذي يليه.
ثم سأل نفسه التالي: هل قيمة المتغير number تساوي 2؟
فكان جواب الشرط كلا (false), فانتقل إلى الشرط الذي يليه.
ثم سأل نفسه التالي: هل قيمة المتغير number تساوي 3؟
فكان جواب الشرط هذه المرة نعم (true), فقام بتنفيذ أمر الطباعة الموجود في جملة الشرط الثالثة, ثم تجاوز جميع جمل الشرط التي أتت بعده.

العوامل التي تستخدم في المقارنات (Comparison Operators)

إسم العامل	رمزه	مثال	شرح الكود
Equal to	==	(a == b)	هل قيمة a تساوي قيمة b ؟ إذا كان الجواب نعم فإنها ترجع true
Not equal to	!=	(a != b)	هل قيمة a لا تساوي قيمة b ؟ إذا كان الجواب نعم فإنها ترجع true
Greater than	>	(a > b)	هل قيمة a أكبر من قيمة b ؟ إذا كان الجواب نعم فإنها ترجع true
Less than	<	(a < b)	هل قيمة a أصغر من قيمة b ؟ إذا كان الجواب نعم فإنها ترجع true
Greater than or Equal to	>=	(a >= b)	هل قيمة a أكبر أو تساوي قيمة b ؟ إذا كان الجواب نعم فإنها ترجع true
Less than or Equal to	<=	(a <= b)	هل قيمة a أصغر أو تساوي قيمة b ؟ إذا كان الجواب نعم فإنها ترجع true

العامل == (Equal To Operator)

العامل == يستخدم لتنفيذ كود معين إذا كانت قيمة المتغير الأول تساوي قيمة المتغير الثاني.
عندما نضعه في الشرط فإنه يعني هل قيمة المتغير الأول تساوي قيمة المتغير الثاني؟

- إذا كانت تساويها سيكون الجواب true و بالتالي سينفذ الكود.
- أما إذا لم تكن تساويها سيكون الجواب false و بالتالي لن ينفذ الكود.

مثال

```
#include <iostream>
using namespace std;
int main() {
    int a = 10;
    int b = 10;
    int c = 20;
    // الشرط التالي يعني أنه إذا كانت قيمة المتغير a تساوي قيمة المتغير b سيتم تنفيذ أمر الطباعة
    if( a == b ) {
        cout << "a = b";  }
    // الشرط التالي يعني أنه إذا كانت قيمة المتغير a تساوي قيمة المتغير c سيتم تنفيذ أمر الطباعة
    if( a == c ) {
        cout << "a = c";    }
    return 0;}
```

سنحصل على النتيجة التالية عند التشغيل.

```
a = b
```

نلاحظ أنه نفذ أمر الطباعة الموضوع في الشرط الأول لأن جواب الشرط كان true.
•لم ينفذ أمر الطباعة الموضوع في الشرط الثاني لأن جواب الشرط كان false.

العامل (Not Equal To Operator) !=

العامل != يستخدم لتنفيذ كود معين إذا كانت قيمة المتغير الأول لا تساوي قيمة المتغير الثاني.
عندما نضعه في الشرط فإنه يعني هل قيمة المتغير الأول لا تساوي قيمة المتغير الثاني؟

- إذا كانت لا تساويها سيكون الجواب true و بالتالي سينفذ الكود.
- أما إذا كانت تساويها سيكون الجواب false و بالتالي لن ينفذ الكود.

مثال

```
#include <iostream>
using namespace std;
int main() {
    int a = 10;
    int b = 10;
    int c = 20;
    // الشرط التالي يعني أنه إذا كانت قيمة المتغير a لا تساوي قيمة المتغير b سيتم تنفيذ أمر الطباعة
    if( a != b )  {
        cout << "a != b";  }
    // الشرط التالي يعني أنه إذا كانت قيمة المتغير a لا تساوي قيمة المتغير c سيتم تنفيذ أمر الطباعة
    if( a != c )  {
        cout << "a != c"; }
    return 0; }
```

سنحصل على النتيجة التالية عند التشغيل.

```
a != c
```

نلاحظ أنه نفذ أمر الطباعة الموضوع في الشرط الثاني لأن جواب الشرط كان true.
•لم ينفذ أمر الطباعة الموضوع في الشرط الأول لأن جواب الشرط كان false.

العامل > (Greater Than Operator)

العامل > يستخدم لتنفيذ كود معين إذا كانت قيمة المتغير الأول أكبر من قيمة المتغير الثاني.
عندما نضعه في الشرط فإنه يعني هل قيمة المتغير الأول أكبر من قيمة المتغير الثاني؟

- إذا كانت أكبر منها سيكون الجواب true و بالتالي سينفذ الكود.
- إذا لم تكن أكبر منها سيكون الجواب false و بالتالي لن ينفذ الكود.

مثال

```
#include <iostream>
using namespace std;
int main() {
    int a = 10;
    int b = 5;
    // الشرط التالي يعني أنه إذا كانت قيمة المتغير a أكبر من قيمة المتغير b سيتم تنفيذ أمر الطباعة
    if( a > b ) {
        cout << "a > b";  }
    // الشرط التالي يعني أنه إذا كانت قيمة المتغير b أكبر من قيمة المتغير a سيتم تنفيذ أمر الطباعة
    if( b > a ) {
        cout << "b > a"; }
    return 0;}
```

سنحصل على النتيجة التالية عند التشغيل.

```
a > b
```

نلاحظ أنه نفذ أمر الطباعة الموضوع في الشرط الأول لأن جواب الشرط كان true.
•لم ينفذ أمر الطباعة الموضوع في الشرط الثاني لأن جواب الشرط كان false.

العامل < (Less Than Operator)

العامل < يستخدم لتنفيذ كود معين إذا كانت قيمة المتغير الأول أصغر من قيمة المتغير الثاني.
عندما نضعه في الشرط فإنه يعني هل قيمة المتغير الأول أصغر من قيمة المتغير الثاني؟

- إذا كانت أصغر منها سيكون الجواب true و بالتالي سينفذ الكود.
- إذا لم تكن أصغر منها سيكون الجواب false و بالتالي لن ينفذ الكود.

مثال

```
#include <iostream>
using namespace std;
int main() {
    int a = 10;
    int b = 5;
    // الشرط التالي يعني أنه إذا كانت قيمة المتغير a أصغر من قيمة المتغير b سيتم تنفيذ أمر الطباعة
    if( a < b )  {
        cout << "a < b";  }
    // الشرط التالي يعني أنه إذا كانت قيمة المتغير b أصغر من قيمة المتغير a سيتم تنفيذ أمر الطباعة
    if( b < a ) {
        cout << "b < a";  }
    return 0; }
```

سنحصل على النتيجة التالية عند التشغيل.

```
b < a
```

نلاحظ أنه نفذ أمر الطباعة الموضوع في الشرط الثاني لأن جواب الشرط كان true.
•لم ينفذ أمر الطباعة الموضوع في الشرط الأول لأن جواب الشرط كان false.

العامل >= (Greater Than or Equal To Operator)

العامل >= يستخدم لتنفيذ كود معين إذا كانت قيمة المتغير الأول أكبر أو تساوي قيمة المتغير الثاني.
عندما نضعه في الشرط فإنه يعني هل قيمة المتغير الأول أكبر أو تساوي قيمة المتغير الثاني؟

- إذا كانت أكبر منها أو تساويها سيكون الجواب true و بالتالي سينفذ الكود.
- إذا لم تكن أكبر منها أو تساويها سيكون الجواب false و بالتالي لن ينفذ الكود.

مثال

```
#include <iostream>
using namespace std;
int main() {
   int a = 5;
   int b = 5;
   int c = 10;
   // الشرط التالي يعني أنه إذا كانت قيمة a أكبر أو تساوي قيمة المتغير b سيتم تنفيذ أمر الطباعة المتغير
   if( a >= b ) {
      cout << "a >= b" << endl; }
   // الشرط التالي يعني أنه إذا كانت قيمة a أكبر أو تساوي قيمة المتغير c سيتم تنفيذ أمر الطباعة المتغير
   if( a >= c )  {
      cout << "a >= c" << endl; }
   // الشرط التالي يعني أنه إذا كانت قيمة c أكبر أو تساوي قيمة المتغير a سيتم تنفيذ أمر الطباعة المتغير
   if( c >= a )  {
      cout << "c >= a" << endl;  }
   return 0;}
```

سنحصل على النتيجة التالية عند التشغيل.

```
a >= b
c >= a
```

نلاحظ أنه نفذ أمر الطباعة الموضوع في الشرط الأول و الثالث لأن جواب الشرط كان true.
•لم ينفذ أمر الطباعة الموضوع في الشرط الثاني لأن جواب الشرط كان false.

العامل <= (Less Than or Equal To Operator)

العامل <= يستخدم لتنفيذ كود معين إذا كانت قيمة المتغير الأول أصغر أو تساوي قيمة المتغير الثاني.
عندما نضعه في الشرط فإنه يعني هل قيمة المتغير الأول أصغر أو تساوي قيمة المتغير الثاني؟

- إذا كانت أصغر منها أو تساويها سيكون الجواب true و بالتالي سينفذ الكود.
- إذا لم تكن أصغر منها أو تساويها سيكون الجواب false و بالتالي لن ينفذ الكود.

مثال

```
#include <iostream>
using namespace std;
int main() {
   int a = 5;
   int b = 5;
   int c = 10;
   // الشرط التالي يعني أنه إذا كانت قيمة a أصغر أو تساوي قيمة المتغير b سيتم تنفيذ أمر الطباعة المتغير
   if( a <= b )  {
      cout << "a <= b" << endl; }
   // الشرط التالي يعني أنه إذا كانت قيمة a أصغر أو تساوي قيمة المتغير c سيتم تنفيذ أمر الطباعة المتغير
   if( a <= c )   {
      cout << "a <= c" << endl;    }
   // الشرط التالي يعني أنه إذا كانت قيمة c أصغر أو تساوي قيمة المتغير a سيتم تنفيذ أمر الطباعة المتغير
   if( c <= a ) {
      cout << "c <= a" << endl;    }
   return 0; }
```

سنحصل على النتيجة التالية عند التشغيل.

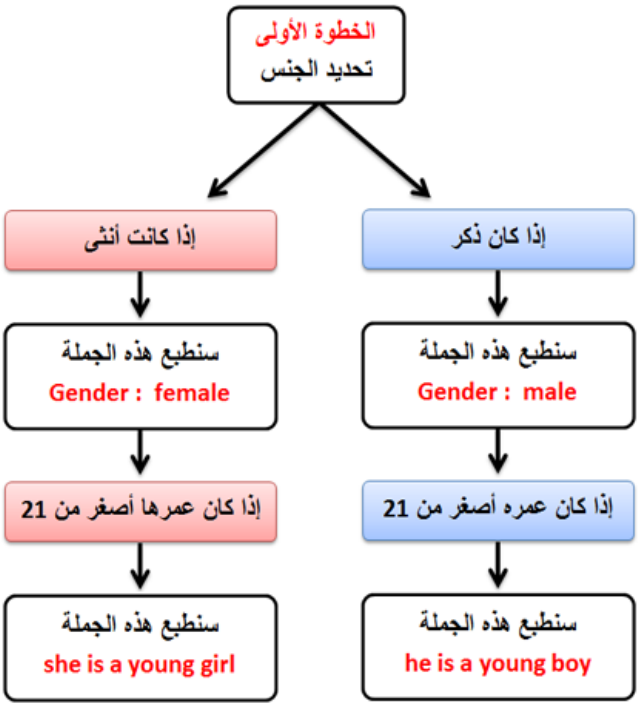
```
a <= b
a <= c
```

نلاحظ أنه نفذ أمر الطباعة الموضوع في الشرط الأول و الثاني لأن جواب الشرط كان true.
•لم ينفذ أمر الطباعة الموضوع في الشرط الثالث لأن جواب الشرط كان false.

مثال حول شرط بداخل شرط

نستطيع وضع الشروط بداخل بعضها, و يمكننا وضع العدد الذي نريده من الشروط.
في البرامج الفعلية يقوم المبرمج بوضع العديد من الشروط بداخل بعضها, و ذلك حسب الفكرة التي يريد تنفيذها في برنامجه.

مثال بسيط, لنفترض أننا نريد تحويل هذه الصورة إلى برنامج.



مثال

هنا قمنا بتجهيز متغير يمثل نوع الجنس إسمه isMale و قلنا إذا كانت قيمته true فهذا يعني أن الشخص هو ذكر.
و إذا كانت تساوي false فهذا يعني أن الشخص هو أنثى.
بالإضافة لمتغير يمثل العمر إسمه age.

```
#include <iostream>
using namespace std;
int main(){
    bool isMale = false;
    int age = 14;
    if( isMale == true )   {
        cout << "Gender: male";
        if( age <= 21 )     {
            cout << "\nHe is a young boy";  }   }
    else    {
        cout << "Gender: female";
        if( age <= 21 )        {
            cout << "\nShe is a young girl";     }    }
    return 0; }
```

سنحصل على النتيجة التالية عند التشغيل.

```
Gender: female
She is a young girl
```

هنا سأل نفسه التالي: هل قيمة المتغير isMale تساوي true؟
فكان جواب الشرط false , فانتقل إلى جملة الشرط الإفتراضية else لينفذ ما بداخلها.
أولاً, قام بتنفيذ دالة الطباعة الموجودة فيها, فطبع الجملة Gender: female.
ثم وجد شرطاً آخر بداخل الشرط الإفتراضي, فسأل نفسه التالي: هل قيمة المتغير age أصغر أو تساوي 21؟
فكان جواب الشرط true, فقام بتنفيذ دالة الطباعة الموجودة فيها, فطبع الجملة She is a young girl.

تعريف الجملة switch

switch نستخدمها إذا كنا نريد إختبار قيمة متغير معين مع لائحة من الإحتمالات نقوم نحن بوضعها, و إذا تساوت هذه القيمة مع أي إحتمال وضعناه ستتنفذ الأوامر التي وضعناها في هذا الإحتمال فقط.
كل إحتمال نضعه يسمى case.
أنواع المتغيرات التي يمكن إختبار قيمتها باستخدام هذه الجملة هي:
int - short - char - byte - enum.

طريقة تعريفها

يمكننا تعريفها بعدة أشكال, الشكل الأساسي هو التالي:

```
switch(expression) {
   case value:
      // Statements
      break;
   case value:
      // Statements
      break;
   default:
      // Statements
      break;
}
```

- switch تعني إختبر قيمة المتغير الموضوع بين قوسين.
- expression هنا يقصد بها المتغير الذي نريد إختبار قيمته.
  نوع المتغير الذي يسمح لنا بإختباره. int - byte - short - char - String - enum
- case تعني حالة value, تعني قيمة, و Statements تعني أوامر.
  و يقصد من هذا كله, أنه في حال كانت قيمة الـ expression تساوي هذه القيمة سيقوم بتنفيذ الأوامر الموضوعة بعد النقطتين :.
  الآن بعد تنفيذ جميع الأوامر الموضوعة بعد النقطتين, يجب وضع break لكي يخرج من الجملة switch مباشرةً بدل أن ينتقل للـ case التالية الموجودة في الجملة switch.
  نستطيع وضع العدد الذي نريده من الـ case بداخل الجملة switch.
  إنتبه: الـ expression و الـ value يجب أن يكونا من نفس النوع.
- default تعني إفتراضياً و هي نفس فكرة الجملة else, و يمكننا أن لا نضعها أيضاً.
  هذه الجملة تنفذ فقط في حال لم تنفذ أي case موجودة في الجملة switch و لذلك نضعها بالآخر.

لا حاجة لوضع break للحالة الأخيرة لأن البرنامج سيخرج من الجملة switch في جميع الأحوال.
ستفهم فكرة الجملة switch من خلال الأمثلة و سنريك كيف يمكنك أن تنفذ نفس الأوامر لأكثر من حالة, من خلال الأمثلة.

المثال الأول

سنقوم باختبار قيمة المتغير x و الذي نوعه int.

سنضع عدة حالات و كل حالة تطبع شيء معين.

```
#include <iostream>
using namespace std;
int main() {
    int x = 2;
    switch( x ) {   // إختبر قيمة المتغير x
        case 1:   // في حال كانت تساوي ١ سيتم تنفيذ أمر الطباعة الموضوع فيها
            cout << "x contain 1";
            break;
        case 2:   // في حال كانت تساوي ٢ سيتم تنفيذ أمر الطباعة الموضوع فيها
            cout << "x contain 2";
            break;
        case 3:   // في حال كانت تساوي ٣ سيتم تنفيذ أمر الطباعة الموضوع فيها
            cout << "x contain 3";
            break;
        default:   //  في حال كانت لا تساوي أي قيمة من القيم الموضوعة سيتم تنفيذ أمر الطباعة الموضوع فيها
            cout << "x contain a different value";    }
    return 0; }
```

سنحصل على النتيجة التالية عند التشغيل.

```
x contain 2
```

وضع نفس الأوامر لأكثر من حالة

إذا أردت وضع نفس الأوامر لأكثر من حالة, عليك وضع الحالات تحت بعضها, ثم كتابة الأوامر, ثم وضع break.

مثال

سنقوم باختبار قيمة المتغير x و الذي نوعه int.

سنضع ثلاث حالات ينفذون نفس الأوامر.

```
#include <iostream>
using namespace std;
int main() {
   int x = 3;
   switch( x )  {    // إختبر قيمة المتغير x
      case 1:    // في حال كانت تساوي ١ أو ٢ أو ٣ سيتم تنفيذ أمر الطباعة
      case 2:
      case 3:
         cout << "x contain 1 or 2 or 3";
         break;
      default:    // في حال كانت لا تساوي أي قيمة من القيم الموضوعة سيتم تنفيذ أمر الطباعة الموضوع فيها
         cout << "x contain a different value";  }
   return 0;}
```

سنحصل على النتيجة التالية عند التشغيل.

```
x contain 1 or 2 or 3
```

تمرين عملي:

قم بإنشاء برنامج لإدخال علامة طالب فإذا كانت العلامة اكبر أو تساوي ٩٠ فالتقدير (A) أما إذا كانت اكبر أو تساوي ٨٠ فالتقدير (B)أما إذا كانت اكبر أو تساوي 70فال تقدير (C) أما إذا كانت اكبر أو تساوي ٦٠ فالتقدير (D) أما إذا كانت اكبر أو تساوي ٥٠ فالتقدير (E) ما عدا ذلك فالتقدير(F)؟

الحل:

main.cpp

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5
6  int mark;
7  char grade;
8  cin>>mark;
9  if (mark>=90)
10 grade='a';
11 else
12 if (mark>=80)
13 grade='b';
14 else
15 if (mark>=70)
16 grade='c';
17 else
18 if (mark>=60)
19 grade='d';
20 else
21 if (mark>=50)
22 grade='e';
23 else
24 grade='f';
25 cout<<grade;
26 return 0; }
```

input

```
90
a

...Program finished with exit code 0
Press ENTER to exit console.
```

الشكل (١٤,٢)

الفصل الخامس عشر: عبارات أو جمل التكرار

التكرار وحلقات التكرار Loops and Repetition

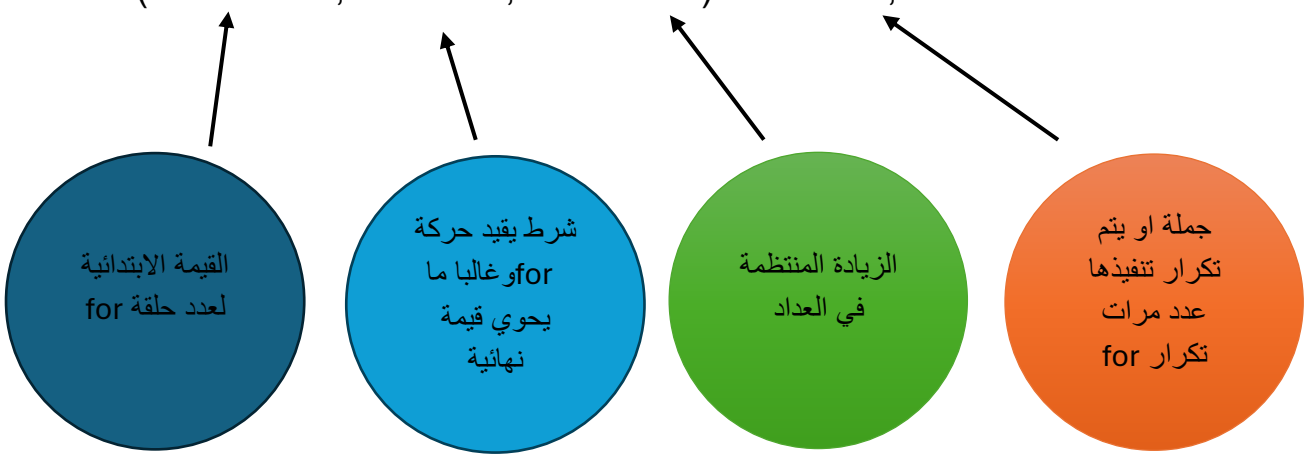
توفر لغة C++ ، كسائر لغات البرمجة ، عددا من أساليب التكرار المشروط ، وغير المشروط ومن هذه الأساليب:

أسلوب التكرار باستعمال حلقة For

يمتلك أسلوب التكرار باستعمال for قوة ومرونة ، لا تتوفران في غيرها من اللغات.

الصيغة العامة الأولى

```
For (initial-value; condition; increment ) statement;
```



تقوم جملة For هنا مبتدئة بقيمة العداد الابتدائية بتنفيذ الجملة statement١ أول مرة ، وفي المرة التالية تزداد القيمة الابتدائية للعداد بمقدار الزيادة ثم تنفذ جملة statement١ مرة ثانية.. وهكذا حتى يستكمل الشرطcondition أمر إنهاء عمليات التكرار والخروج من حلقة For .

الصيغة العامة الثانية

```
For ( initial-value; condition; increment )
{
statement;
}
```

جملة أو اكثر

امثلة :

في المثال التالي قمنا بتعريف حلقة تطبع جميع الأرقام من **1** إلى **10**.

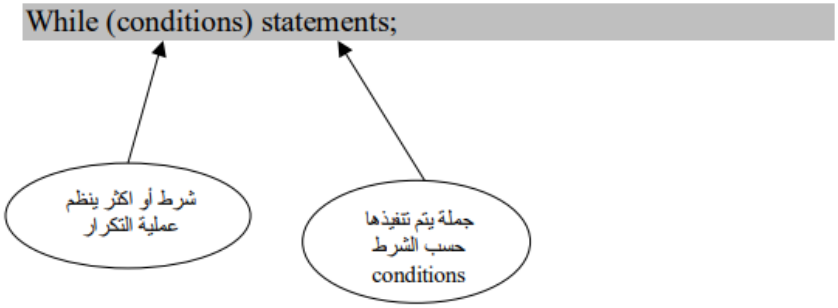
```cpp
#include <iostream>

using namespace std;

int main()
{
    // هنا قمنا for تتألف من 10 دورات. في كل دورة تطبع قيمة العداد المستخدم فيها
        بإنشاء حلقة
    for( int i=1; i<=10; i++ )
    {
        cout << i << endl;
    }

    return 0;
}
```

الناتج:

```
/tmp/OaQLOp9QSP.o
1
2
3
4
5
6
7
8
9
10

=== Code Execution Successful ===
```

أسلوب التكرار باستعمال حلقة While

أسلوب التكرار باستعمال حلقة while أسلوب آخر يماثل أسلوب حلقة for ، مع بعض الاختلافات البسيطة ، وهو أسلوب يثرى لغة C++ ، ويزدها قوة ومرونة ، والشكل العام لهذا الأسلوب:



ومعنى حلقة التكرار while هو الأتي:

آي ما دام الشرط (أو الشروط) متحققا (وجوابه true) فيتم تكرار تنفيذ الجملة أو الجمل (statements)، وينتقل تسلسل تنفيذ البرنامج إلى الجملة التي تلي حلقة while.

مثال: في المثال التالي قمنا بتعريف حلقة تطبع جميع الأرقام من ١ إلى ١٠.

```cpp
#include <iostream>

using namespace std;

int main()
{
    // هنا قمنا بتعريف المتغير الذي استخدمناه كعداد في الحلقة
    int i=1;

    // تظل تنفذ الأوامر الموضوعة فيها طالما أن قيمة العدد لا تزال أصغر أو تساوي
        10 while هنا أنشأنا حلقة
    while( i<=10 )
    {
        // في كل دورة سيتم طباعة قيمة العداد ثم إضافة 1 عليها
        cout << i << endl;
        i++;
    }

    return 0;
}
```

الناتج:

```
/tmp/aRcPo3FbGR.o
1
2
3
4
5
6
7
8
9
10

=== Code Execution Successful ===
```

جملة الإيقاف Break

من الاسم نستطيع أن نلاحظ أن وظيفة Break هي إيقاف بنية أو حلقة تكرار عند تحقق شرط أو شروط معينة ، وعند تنفيذها يتم القفز إلى سلسلة الجمل التالية للبنية أو حلقة التكرار ، وتستعمل Break أيضا في إيقاف حلقة التكرار لانهائي ، أوالخروج منها إلى الجمل التي تليها وكما في المثال التالي:

مثال : في المثال التالي قمنا بتعريف حلقة كانت ستطبع جميع الأرقام من ١ إلى ١٠ لولا أننا استخدمنا الجملة break لجعل الحلقة تتوقف عندما تصبح قيمة العداد i تساوي ٦.

```cpp
#include <iostream>

using namespace std;

int main()
{
    // هنا قمنا for تتألف من 10 دورات. في كل دورة تطبع قيمة العداد المستخدم فيها بإنشاء حلقة
    for( int i=1; i<=10; i++ )
    {
        // في كل دورة سيتم فحص قيمة العداد و بمجرد أن تصبح تساوي 6 سيتم إيقاف الحلقة نهائياً
        if( i == 6 ) {
            break;
        }

        cout << i << endl;
    }

    return 0;
}
```

الناتج:

```
/tmp/lj8nljNoiH.o
1
2
3
4
5


=== Code Execution Successful ===
```

الفرق بين دوال التكرار (while-do,while-for)

- في جملة do while و while ، يجب أن تقوم بتعريف المتغيّر وإعطائه قيمة ابتدائية، قبل الشروع في كتابة الجملة.
- كما أنه في جمل while و do while، يتوجب عمل جملة الزيادة أو النقصان داخل الحاصرتين.
- لكن فقط في جملتي for و while عند تشغيل البرنامج يتم أولًا فحص الشرط قبل الدخول إلى عملية تنفيذ الجمل داخل loop، بعكس do while التي تنفذ مرة واحدة على الأقل، بعد ذلك يتم فحص الشرط.
- أيضا يجب وضع فاصلة_منقوطة(;) بعد الشرط لجملة do while.

الفرق الجوهري بين for وبقية جمل التكرار : هي أن بداية انطلاقة for معروفة مسبقًا من تهيئة المتغير، أما في while و do while فقد تكون غير معروفة بعد.

تمرين :

قم بتعريف حلقة تطبع جميع الأرقام من 1 إلى 50.

الحل:

```cpp
#include <iostream>

using namespace std;

int main()
{
    // هنا قمنا بتعريف المتغير الذي استخدمناه كعداد في الحلقة
    int i=1;

    // تظل تنفذ الأوامر الموضوعة فيها طالما أن قيمة العدد لا تزال أصغر أو تساوي
        10 while هنا أنشأنا حلقة
    while( i<=20 )
    {
        // في كل دورة سيتم طباعة قيمة العداد ثم إضافة 1 عليها
        cout << i << endl;
        i++;
    }

    return 0;
}
```

الناتج :

```
/tmp/37TcJe8Jqb.o
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20

=== Code Execution Successful ===
```

الفصل السادس عشر: تحديد الأخطاء الشائعة في البرمجة وتصحيحها

مفهوم معالجة الأخطاء

معالجة الأخطاء (Exceptions Handling) يقصد منها كتابة الكود الذي قد يسبب أي مشكلة في البرنامج بطريقة تضمن أنه إذا حدث الخطأ المتوقع أو أي خطأ آخر فإن البرنامج لن يعلّق أو يتم إغلاقه بشكل فجائي.

ظهور خطأ في البرنامج بشكل مفاجئ هو أمر سيئ جداً لأنه يؤدي إلى نفور عدد كبير من المستخدمين و عدم رغبتهم في العودة إلى استخدام هذا البرنامج مجدداً.

أنواع الأخطاء

- أخطاء لغوية (Syntax Errors) و يقصد بها أن تخالف مبادئ اللغة مثل أن تعرّف شيء بطريقة خاطئة أو تنسى وضع فاصلة منقوطة.
- أخطاء تحدث أثناء تشغيل البرنامج يقال لها إستثناءات (Exceptions) مما يؤدي إلى تعليقه و إيقافه بشكل غير طبيعي.
- أخطاء منطقية (Logical Errors) و يقصد منها أن الكود يعمل بدون أي مشاكل و لكن نتيجة تشغيل هذا الكود غير صحيحة.

إذاً, أي خطأ برمجي يحدث معك أثناء تشغيل البرنامج يقال له إستثناء (Exception) حتى إن كان اسم الخطأ يحتوي على كلمة Errorبمعنى آخر, أي Error يظهر لك أثناء تشغيل البرنامج يعتبر Exception.

في هذا الدرس ستتعلم كيف تتجنب حدوث أخطاء في البرامج التي تكتبها, و فعلياً ستتعلم كيف تجهز البرنامج للتعامل مع الأخطاء التي قد تحدث أثناء تشغيله لجعل البرنامج شغال دائماً في نظر المستخدم و لا يظهر له أي أخطاء.

بعض الأسباب التي تسبب حدوث إستثناء

- في حال كان البرنامج يتصل بالشبكة و فجأةً إنقطع الإتصال.
- في حال كان البرنامج يحاول قراءة معلومات من ملف نصي, و كان هذا الملف غير موجود.
- في حال كان البرنامج يحاول إنشاء أو حذف ملف و لكنه لا يملك صلاحية لفعل ذلك.

معلومة تقنية

في حال كان الكود الذي كتبته يحتوي على أخطاء لغوية (Syntax Errors) لا بد من أن تصلحها كلها حتى يستطيع المترجم تحويل الكود الذي كتبته لكود يفهمه الحاسوب و من ثم ينفذه لك. أي لا يمكنك حماية البرنامج من أخطاء موجودة في الكود نفسه بل يمكنك حمايته من المشاكل التي قد تحدث وقت عمل هذا الكود.

في هذا الدرس ستتعلم كيف أن المكتبات الجاهزة التي قد تستخدمها مستقبلاً في مشاريعك يتم جعلها تظهر أخطاء إذا لم تستخدمها كما يجب.

معرفة هذا الأمر مهمة جداً لك أيضاً, لأنك ستتمكن من مشاركة الكود الذي تقوم بإعداده مع مبرمجين آخرين و إظهار أخطاء فيه إن لم يقوموا باستخدامه بشكل صحيح.

أمثلة على أنواع الأخطاء

في المثال التالي لم نضع فاصلة منقوطة في آخر الأمر cout مما سيؤدي لحدوث مشكلة عندما يحاول المترجم تشغيل البرنامج.

إذاً الكود التالي يحتوي على خطأ لغوي (Syntax Error).

```cpp
#include <iostream>

using namespace std;

int main()
{
    int x = 10;

    cout << x

    return 0;
}
```

سيظهر الخطأ التالي عند التشغيل والذي يعني أن المترجم يتوقع أن تضع له فاصلة منقوطة في آخر السطر التاسع.

```
ERROR!
/tmp/jeEHa3JZLr.cpp: In function 'int main()':
/tmp/jeEHa3JZLr.cpp:9:14: error: expected ';' before 'return'
    9 |     cout << x
      |              ^
      |              ;
   10 |
   11 |     return 0;
      |     ~~~~~~


=== Code Exited With Errors ===
```

في المثال التالي قمنا بإنشاء برنامج يطبع للطالب ما إذا كان ناجحاً أو راسباً بناءاً على معدله النهائي.

من المفترض أنه يتم اعتبار الطالب راسب في حال كان معدله بين ٠ و ٩,٩ , و يتم اعتباره ناجح في حال كان معدله ١٠ و ٢٠.

هنا تعمدنا وضع خطأ منطقي (Logical Error) حيث أننا عند طباعة نتيجة الطالب لم نتأكد ما إذا كان المعدل بين ٠ كحد أدنى و ٢٠ كحد أقصى.

```cpp
#include <iostream>

using namespace std;

int main()
{
    float average = 25;

    if (average < 10)
    {
        cout << "The student failed the exam";
    }
    else if (average >= 10)
    {
        cout << "The student passed the exam";
    }

    return 0;
}
```

سنحصل على النتيجة التالية عند التشغيل ونلاحظ أنه لا يوجد مشكلة برمجية سببت إيقاف الكود لكننا نعلم أن هناك مشكلة منطقية في الكود لأن المعدل الذي تم على أساسه طباعة جملة النجاح هو معدل مستحيل أن يكون حقيقي

```
/tmp/V30i6gmNnr.o
The student passed the exam

=== Code Execution Successful ===
```

المراجع

#	اسم المرجع
١	Course 20410D: Installing and Configuring Windows Server 2012 -1
٢	Installing and Configuring Windows Server® 2012 R2 Exam 70-410 by Craig Zacker – 2014
٣	Introducing Windows Server 2012 R2 Technical Overview Mitch Tulloch with the Windows Server team
٤	شرح عملي باللغة العربية Guide to Microsoft Server – محمد خميس العجمي
٥	موسوعة البرمجة بلغة C++للمؤلف / مجلاد مشاري السبيعي
٦	كتابة خطوة بخطوة لتعلم لغة c ,c++ للكاتب حسين احمد طالب
٧	https://www.youtube.com/watch?v=_EGk-5ozWR0 قناة المبرمج محمد ماهر